CRUOR COLLECTION SYSTEM

A MINI PROJECT REPORT

Submitted by

POOJA B	**[711719104063]**
SASI KEERTHANA R	**[711719104083]**
SASI KUMAR P	**[711719104084]**
THARINI M	**[711719104101]**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KGiSL INSTITUTE OF TECHNOLOGY, SARAVANAMPATTI

ANNA UNIVERSITY :: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"CRUOR COLLECTION SYSTEM**" is the bonafide work of **"POOJA B, SASI KEERTHANA R, SASI KUMAR P, THARINI M"** who carried out the project work under my supervision.

SIGNATURE

Dr. RAMKUMAR R P

Head Of The Department,

Computer Science and Engineering,

KGiSL Institute of Technology,

Saravanampatti,

Coimbatore-35

SIGNATURE

Ms. RAMYA S R

Supervisor,

Assistant Professor

Computer Science and Engineering,

KGiSL Institute of Technology,

Saravanampatti,

Coimbatore-35

Submitted for the Anna University Viva-Voce examination held on _____

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude to our beloved Chairman, **Padma Shri Dr. G. BAKTHAVATHSALAM,** KGiSL Educational Institutions, for having provided the facilities during the course of our study in the college.

We express our heartfelt gratitude to our venerated Managing Director, **Dr. ASHOK BAKTHAVATHSALAM**, KGiSL Educational Institutions, who gave the opportunity to frame the project to the full satisfaction.

We are very grateful to **Mr. R. ARAVIND KUMAR, CEO, Dr. M SELVAM, M.E., Ph.D.,** Principal and **Dr. S. SURESH KUMAR, M.E., Ph.D.,**Vice Principal for their valuable guidance and blessings.

We would like to thank **Dr. RAMKUMAR R P, M.E., Ph.D.,** Head of the Department, Department of Computer Science and Engineering for his unwaveringsupport during the entire course of our project work and who modelled us both technically and morally for achieving greater success in this project work.

We express our sincere thanks to our faculty guide **Ms. RAMYA S R**, **M.E.,** Assistant Professor, Department of Computer Science and Engineering.

We also thank all the **faculty members** of our department for their help in making this project a successful one.

Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all they meant to us during the crucial times of the completion of our project.

ABSTRACT

A cruor collection system is a web-based system for blood banks to manage information about the donors and blood stock. The hospital can send message alert to all the people who are all near to the hospital. The hospital can check availability of required blood from volunteers and send the message alert to all the donors. In some places only government hospital that handles blood bank currently is using a standalone system. Since most blood banks are still in paper based system, various disadvantages are experienced by various stakeholders, which endanger the lives of patients and deter healthcare system. This web-based system allows hospital to check the availability of the blood bags anytime. Threats on improper blood donor documentation or misplaced records will be totally eradicated. The process involving record about the blood donors, blood bag collection, storage and inventory will be systematized and organized. Through this system, any person who is interested in donating the blood can register himself in the same way if any organization wants to register itself with this site they can also register.

Admin and donor have the main authority who can do addition and modification if required. The requirement of the blood has to be requested and we supply the information of the donor. The donors can update their status whether they are available or not. After the implementation of the project, the blood searching process is expected to be faster, easier, user friendly and reliable.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HTML	Hypertext Markup Language
CSS	Cascading style sheet
JS	JavaScript
SQL	Structured Query Language
UDT	User Defined Technique

CHAPTER 1

INTRODUCTION

Cruor collection system is a process of collecting blood and managing blood stocks, approving blood requests and updating available blood types. Cruor collection system provides a function to send a text message to the donor for their user account and the hospital, the availability of blood request. Cruor collection is to maintain records of blood donors, blood donation information and blood stocks in a centralized database system.

1.1 PROBLEM DEFINITION

- Cruor collection system is web-based system that sends a text message.
- This system helps to register the details of the donors, blood collection details as well as blood issued reports.
- Scarcity of rare blood groups and unavailability of blood during emergency.

1.2 OBJECTIVE OF THE PROJECT

The objective of this project is to contribute to develop a web application for blood banks to manage information about their donors and blood stock. The main objectives of the website development can be defined as follows:

- To ensure hospital to have good supply or inventories of blood bags.
- To check the availability of blood bags anytime.
- To manage the information of its blood donor.
- Function to check if the person donate blood for the last 3 months.
- To allow good documentation about the donor and its blood donation activities.
- Support fast searching to find match blood bags for the right person.

1.2.1 Scope

This research study covers the three basic operations of blood banks, namely donor registration, monitoring of blood bags or products' inventories, and monitoring of blood bags or products' issuance. Also, due to time-constraint, respondents will be from

hospitals from North Region in the Oman, though the research study talks about blood banks in the Sultanate of Oman. In addition, the study considers three possible users of the system, namely hospital administrator, doctors, and blood receptionists.

1.2.2 Limitations

This research study does not cover the actual blood collection activity, and actual blood transfusion operation. Blood donors and patients or recipients of blood donation are not system users, their registration or information will be encoded by the blood bank receptionists.

1.3 ASSUMPTIONS AND HYPOTHESIS

The researchers assume the following assumptions:

- Internet connectivity is needed for the online blood management system. Internet speed may affect the perception of the systems users with regards to the system effectiveness and efficiency.
- Blood transfusion should be performed by medical or professional doctors only. The over-all safety depends on the success of the medical operation.

The researchers identify the following hypotheses:

- There is a significant difference in the level of blood transfusion safety between manual-based and online blood bank systems.
- There is an increased level of blood transfusion safety in using online blood bank management systems while there is an increased risk when using manual-based one.

1.4 SIGNIFICANCE OF PROBLEM

The findings of this study will benefit blood banks in managing blood donation donors, activities, and blood bags. This will allow the hospital to take decision if a particular type of blood is needed and currently unavailable in the hospital, however, available in another nearby hospitals. Furthermore, managing the blood bags in the blood bank will be much easier because each blood bag has an information about the donor,

donation activity details, and the expiration date. Also, doctor can use this system to serve blood bags to their patient and monitor the details of the donor.

The main advantages of the system are:

- Blood bank staff can find and manage the donor details on the system easily.
- The expiration date of blood bags can be viewed in the system.
- Hospital can be alerted about issued blood bags and its availability.
- The system is systematized, and organized in managing blood donor records and blood donation activities.

CHAPTER 2

LITERATURE REVIEW

The purpose of Cruor collection system is to send a text message and used for storing donor information. According to Teena C A, Sankar K and Kannan S. in their study entitled "A Study on Blood Bank Management", they defined Blood Bank Information System as an information management system that contributes to the management of donor records and blood bank. Their system allowed an authorized blood bank administrator to sign in with a password to manage easily the records of donors and patients who need blood. The system provided many features including the central database, quick access to the system content through the login, includes the search code to find donors on a given basis, and the ease of adding and updating donor data. The main aim of the system was to complete0the process of the blood bank. This system was designed to suit all types of blood banks.

Once successful in the implementation of the application, it can be applied and rolled out in several blood banks. This application contains User Login Screen, Blood Management, Menu Form, Blood Stock, Donor Management, Donor Registration, Blood Reservation, Donor Blood Test, Recipient Management and Blood Reservation. In similar manner, the researchers planned in their application to have hospital administrator, doctors, and blood bank receptionists as users. The authors did not mention the research method they used, and failed to provide screenshots of the system prototypes, making difficult for the researchers to visualize their application. No discussion also for their respondents, samples and sampling techniques used. Subsequently, the researchers planned to provide figures to explain the system, screenshots of system prototypes, and other diagrams that can help other researchers to visualize the development of web-based blood bank management system. Also, the researchers will explicitly discuss its research methods, sampling procedures, and statistical treatment to be used for analyzing the gathered data.

On the other hand, study entitled "Blood Bank Management System" done by [3]Kumar, R., Singh, S. and Ragavi, V A, the researchers developed a web-based blood

management which assists the blood donor records management, and provides ease of control in the distribution of blood products in various parts of the country considering demands of hospitals. The developed system was scalable and adaptable to meet the complex needs usually of a blood bank. Based on this study, since entering the details about the blood donors and related records were done manually, thus, tracking of blood donation activities was difficult and complicated, and even led to erroneous information. Subsequently, the researchers mentioned that manual-based system can be waste of time, lead to the error-prone results, consumes a lot of manpower, lacks data security, data retrieval requires a lot of time, reports consume a long time to produce, and there is less precise accuracy on the results. As such, by developing and implementing a web-based blood management information system, there was a quick and timely access to donor records, and the system provided management timely, confidential and secured medical reports. There were three users in the system, namely: Administrator, Donor, and Acceptor. Each user has been given user ID and password to identify their identity. The said application was developed using ASP.NET, C#.NET, and using Sql Server 2000/2005 for the database. The research paper failed to mention the methods of research used.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is a problem-solving technique that decay a system into component pieces of purpose of studying how well those component parts work and interact to accomplish their purpose. The following chapter provides a detailed description of the existing system. It also provides an overview of the proposed system and feasibility of insurance bot.

3.1 EXISTING SYSTEM

With the growing population and the advancement in medical science the demand blood has also increased. There are a quite good number of software packages that exist for BLOOD BANK MANAGEMENT or BLOOD INVENTORY SYSTEM. But, existing system is limited only to those particular blood bank not to hospitals. There is no software to keep any records in blood bank. It becomes difficult to provide any record immediately at times of emergency. Required more human efforts in maintaining the branch related information. Manually to keep the accounts is also tedious & risky job & to maintain those accounts in ledgers for a long period is also very difficult.

3.1.1 DISADVANTAGES

- Blood request are sent through the website or mail id. It causes the user to view the message only he/she is online
- Difficult to manage and maintain the file and Chance of damage of files, if the data is stored in the files for duration of time.
- Time consuming is high to, storing and updating the data.
- It is difficult to keep track the record about the donor & receiver he has donated or received the blood at the last time.

3.2 PROPOSED SYSTEM

The proposed Cruor collection system helps the hospital who are in need of a blood by giving them all details of blood group availability or regarding the donors with the same blood group through blood request by text message through website. This system, overcomes the drawbacks of the present system. The main aim of this project is to develop a web based system to store the blood donor's record safely.

For any blood bank system, we can see message notifications when we are in online but in cruor collection system, we can see message notifications when we are in offline. Cruor collection system saves a lot of time to the blood donor. To overcome the problem in all blood bank management system we have proposed Cruor collection system.

3.2.1 ADVANTAGES

- The Proposed System saves time as he can search donors online without going anywhere and work is reduced very much which prevails in the present system.
- Using this system donor can get text message in time and can save his relative or friend life.
- The main benefit of this system is the information of available blood group and inventory of the blood group.
- The hospital in need of blood can search for the donors on online by giving their details and city name.
- It is very flexible and user friendly and Helpful.
- The people are not limited to receive or provide services in working hours of the branch only; he is serviced 24 hours a day, 7 days of week and 365 days of the year.

3.3 FEASIBILITY STUDY

The feasibility study is performed to determine whether the proposed system is viable considering the Technical, Operational and Economical factors. After going through feasibility study, we can have a clear-cut view of the system's benefits and drawbacks.

3.3.1 Tests of Feasibility

Feasibility study is conducted once the problem is clearly understood. Feasibility study is necessary to determine that the proposed system in cruor collection system is feasible by considering the technical, operational, and economical factors. By having a detailed feasibility study the management in the will have a clear-cut view of the proposed system of the cruor collection system. Feasibility study encompasses the following things:

- Technical Feasibility
- Economical Feasibility
- Operational feasibility

3.3.1.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

The proposed system is developed using Active Server Page, VB Script and HTML as front-end tool. The Proposed system needs a personal web server to serve the requests submitted by the users. The Web browser is used to view the web page that is available within the Windows operating system itself. The proposed system will run under Win9x, NT, and win2000 environment. As windows is user friendly and GUI OS it is very easy to use. All the required hardware and software are readily available in the market. Hence the system is technically feasible.

3.3.1.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system in cruor collection system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of insurance bot development.

The essential questions that help in testing the operational feasibility of a system include the following:

- The customer is benefited more as most of his time is saved. The customer is serviced at his place of work.
- The cost of the proposed system is almost negligible when compared to the benefits gained.

3.3.1.3 Economical Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system of the cruor collection system. A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. 13 The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 Message

- The system should allow hospital to send a message request to the registered donors.
- The system should allow user to receive the blood request.
- The system again send message to remaining donors as “BLOOD RECEIVED”

4.1.2 Searching

- The system should allow hospital to search for information about donor’s login detail and blood available details.
- The system should allow users to search for information about beneficiary, terms and conditions
- The system should allow users to search for information about the login.

4.1.3 Login

- The Login page allows the donor and hospital to login and see their information.
- The login page is applicable for only registered hospital and donors.
- The donor and hospital have a specific username and password
- The donor can login anytime to see their progress and change their profile
- If the donor or hospital forgot their password, they can reset their password only if they are registered.

4.1.4 Administrative System

- Information management: The administrator should be able to add, update and delete user accounts.
- Login management: The administrator and donor should be able to view and delete logs.
- Database management: The administrator should be able to manage the database.

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 User Interface

- The system shall maintain an easy to use interface across all functionality and for all users
- The client's user interface should be compatible with all commonly used browsers, such as Internet explorer, Firefox, Google chrome and Safari.

4.2.2 Scalability

The system shall be able to scale based on the number of users accounts using the system.

4.2.3 Security

- The administrative system should be protected from unauthorized access.
- The database should be protected from attacks and unauthorized access.
- The interface should be protected from attacks.
- All passwords should be stored as a secure hash of the administrator password.

4.2.4 Portability

- The system should run on a variety of operating systems that support the Java language.
- The system should run on a variety of hardware.

4.2.5 Maintainability

- The system should be easy to maintain.
- There should be a clear separation between the interface and the client.
- There should be a clear separation between the data access objects that map the database and the business logic code.

4.2.6 Exception Handling

Exceptions should be reported effectively to the user if they occur.

4.2.7 Ethics

The system shall not store or process any information about its users.

4.3 HARDWARE REQUIREMENTS

Processor	:	Dual core processor
RAM	:	2 GB
Hard Disk	:	250 GB
Monitor	:	16’’ Color Monitor
Keyboard	:	Standard 110 keys
Pointing Device	:	Mouse
Smart Phone	:	Any type

4.4 SOFTWARE REQUIREMENTS

Programming Language	:	Python
Operating System	:	Windows/Linux/Visual studio
Front End	:	HTML, CSS
Back End	:	SQL lite
Web Browser	:	Mozilla Firefox, Google Chrome

CHAPTER 5
SOFTWARE DESCRIPTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between donors and hospital on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

5.1 FRONT END

The front end is designed using framework which includes a collaborative platform for hospital and the donor. It is the collaborative end to end platform made by developers. Here all the web tools are integrated and it allows interaction between the client and the web page. It uses python for backend functions and Django framework for designing. The web application access easier because of framework.

5.1.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is used for web development (server side), software development and system scripting.

5.1.1.1 Features

- Simple: It is very simple and easy to use, compare to other scripting language it is very simple and easy, this is widely used all over the world.
- Interpreted: It is an interpreted language, i.e. there is no need for compilation.

- Open Source: Open source means you no need to pay for use python, you can free download and use.
- Platform Independent: Python code will be run on every platform, Linux and Windows.
- Case Sensitive: Python is case sensitive scripting language at time of variable declaration. In Python, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.

5.1.1.2 Advantanges

- Open source and community development
- Extensive support libraries (NumPy for numerical calculations, Pandas for data analytics etc.)
- Versatile, Easy to read, learn and write
- User-friendly data structures
- High-level language
- Dynamically typed language (No need to mention data type based on the value assigned, it takes data type)
- Object-oriented language
- Portable and Interactive
- Ideal for prototypes – provide more functionality with less coding
- Highly Efficient (Python's clean object-oriented design provides enhanced process control, and the language is equipped with excellent text processing and integration capabilities, as well as its own unit testing framework, which makes it more efficient).

5.2 BACK END

The back end is designed using MySQL, whose primary function is to store data securely and retrieve it later, as requested by other software applications.

5.2.1 MySQL

MySQL is an open-source relational database management system (RDBMS). The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal.

5.2.2 Features of MySQL

- A broad subset of ANSI SQL 99, as well as extensions.
- Cross-platform support.
- Stored procedures, using a procedural language that closely adheres to SQL/PSM.
- Triggers and Cursors.
- Updatable views.
- Online DDL when using the InnoDB Storage Engine.
- Information schema.

5.2.3 Advantages of MySQL

- Data Security
- On-Demand Scalability
- High Performance
- Round-the-Clock Uptime
- Comprehensive Transactional Support

CHAPTER 6

PROJECT DESCRIPTION

Cruor collection system automates are used in most of the hospitals. The proposed Cruor collection system helps the hospital who are in need of a blood by giving them all details of blood group availability or regarding the donors with the same blood group through blood request by text message through website. This system, overcomes the drawbacks of the present system. The main aim of this project is to develop a web-based system to store the blood donor's record safely.

6.1 OVERVIEW OF THE PROJECT

The cruor collection provides a complete set of methods in order to send message notifications. Cruor collection system stores the donor's information in the database.

6.2 MODULE DESCRIPTION

6.2.1 Login Page

- The Login page allows the donor and hospital to login and see their information.
- The login page is applicable for only registered hospital and donors.
- The donor and hospital have a specific username and password
- The donor can login anytime to see their progress and change their profile
- If the donor or hospital forget their password, they can reset their password only if they are registered.

6.2.2 Donors Information

Donors' information is stored in database using MySQL. Database having basic information of donors like name, age, blood group and phone number. The Donors information are stored safe and secured. The donor can modify their information. Hospital can maintain the database.

6.2.3 Blood Request

When the blood is needed, the hospital can send a blood request to the donors those who are already registered. Blood request is sent through text message to the registered donors. The database is maintained by the administrative system.

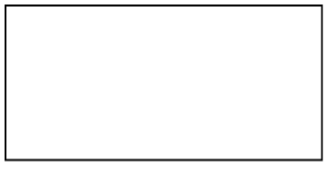
6.2.4 Blood Received Message Alert

When the blood request received to the donor, they can donate the blood. The donors who ready to donate blood contact to the hospital. After receiving blood remaining donors can get the message as “BLOOD RECEIVED”.

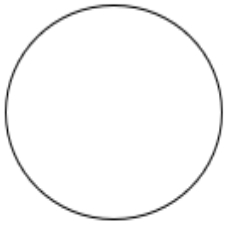
6.3 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identifies how the information flows through the processes. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs. The data flow diagram also depicts the flow of the process and it has various levels. The initial level is context level which describes the entire system functionality and the next level describes each and every sub module in the main system as a separate process or describes all the process involved in the system separately.

Data flow diagram are made up of number of symbols,



Square presenting external entities, which are sources or destinations of data

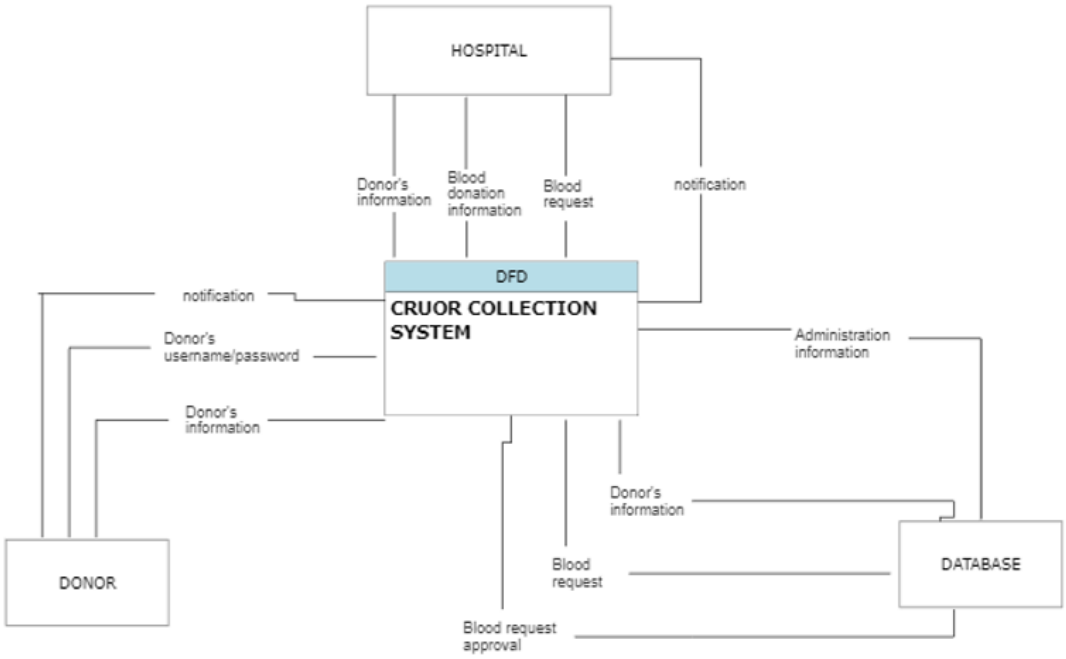


Circle representing processes, which take data as input, do something to it & output to it

Arrows representing the data flows, which can either, be electronic data or physical items.

Parallel lines representing data stores, including electronic stores such as databases or XML files and physical stores

6.3.1 DFD LEVEL 0



6.4 DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can

then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

TABLE 6.4.1 User login detail

S.NO	FIELD NAME	DATATYPE
1	user name	Varchar
2	Use mobile number	Long int
3	password	Varchar

TABLE 6.4.2 Admin login details

S.NO	FIELD NAME	DATATYPE
1	Admin name	varchar
2	password	Varchar

TABLE 6.4.3 User's Details

S.NO	FIELD NAME	FIELD TYPE
1	User's name	Varchar

2	mobile number	Double
3	Password	Varchar

6.5 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

6.6 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output.

CHAPTER 7

SYSTEM TESTING

System Testing is a level of the software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. By definition of ISTQB system testing is the process of testing an integrated system to verify that it meets specified.

7.1 TESTING METHODS

Software Testing Type is a classification of different testing activities into categories, each having, a defined test objective, test strategy, and test deliverables. The goal of having a testing type is to validate the Application under Test for the defined Test Objective.

For instance, the goal of Accessibility testing is to validate the AUT to be accessible by disabled people. So, if your Software solution must be disabled friendly, you check it against Accessibility Test Cases.

7.2 TYPES OF TESTING

7.2.1 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, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this cruor collection system, every unit of code is been tested and the correctness of every module is been ensured.

7.2.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration

testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

In this cruor collection system, the units are been tested as a whole and the testing was successful.

7.2.3 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing usually describes what the system does. Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system.

Functional testing has many types:

- Smoke testing
- Sanity testing
- Regression testing
- Usability testing

7.2.4 Stress Testing

Stress testing a Non-Functional testing technique that is performed as part of performance testing. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.

Reasons can include:

- to determine breaking points or safe usage limits
- to confirm mathematical model is accurate enough in predicting breaking points or safe usage limits
- to confirm intended specifications are being met

- to determine modes of failure (how exactly a system fails)
- to test stable operation of a part or system outside standard usage

In this cruor collection system, whole of the modules are been tested and it has the safe usage measures.

7.2.5 Acceptance Testing

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

In this cruor collection system, the donor’s acceptance is been monitored and it is been put into usage.

7.2.6 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box are testing. It is one of two parts of the "box testing" approach of software testing. Its counter-part, black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing.

The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing"

symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

In this cruor collection system, all the inner functionality is been tested and it is been correctly implemented.

7.2.7 Black Box Testing

Black box testing is a software testing technique in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In this cruor collection system, the implementation part is been checked for its correctness.

7.2.7.1 Methods of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones

- Functional testing - This black box testing type is related to functional requirements of a system; it is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.

7.3 TESTING STRATEGY

Test Strategy is also known as test approach defines how testing would be carried out. Test approach has two techniques:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- Reactive - An approach in which the testing is not started until after design and coding are completed.

- Test strategy calls for implementing two entirely different methodologies for testing this project. The cruor collection system includes a fair amount of manual UI-based testing.

In this cruor collection system, proactive approach is been used for testing. Since proactive approach is efficient it is been used in this cruor.

CHAPTER 8

SYSTEM IMPLEMENTATION

This section discusses the implementation details declaring each stage to fulfill the main scenario listed above in system description. Next sub-section clarifies algorithms implemented in the project based on use-cases declared earlier. Then, the following sub-section explains user interfaces.

8.1 REGISTER

It is handy to create an account in the website by giving the basic details in which it prompts for a simple login with the mail id or Mobile Number and the password credentials, on verification of the mail id, it allows the donor to see their progress and updates.

8.2 BLOOD DONATION

The system wishes to launch on the internet, so that the donors are able to view their blood donation records online and administrators can create, update, delete and query record conveniently.

8.3 REQUIREMENTS

The requirements tab allows the user to perform the relevant tasks that are been pertained based on the condition given on the trigger tab. Generally, the requirements are been specified in special cases if not required then it can be omitted.

8.4 ACTIONS

The actions are been generally performed with the creation of entities. The entities are the data bucket that is used to store the values. The entities are been used to map the keywords with the intent and it is been made to give the response for the user's queries. In this actions tab, the way by which the donor responds to the query will be given. It can be added with text, buttons, images, animated GIFs, quick replies, cards, links and fallbacks.

8.5 DATABASE CONNECTING

Database connecting allows the developer to connect the hospital and the donor. Storing the information in the database is a secure manner. It provides the authentication. Whenever the information is required, it fetch from the database and give it to the user.

8.6 DATABASE TESTING

This step is the process of testing whether the programming code will work correctly with the conditions in our system or not. In this phase, we will fix bugs in order to produce a system with maximum performance.

CHAPTER 9

CONCLUSION & FUTURE ENHANCEMENTS

9.1 CONCLUSION

Based on results, this report concluded that cruor collection system is much better than the existing system. The findings showed that respondents prefer to use cruor collection system rather than the existing blood collection system because it offers many advantages and benefits that lead to its effectiveness and efficiency. Because of the increased confidence on the users of the system, it can be concluded that the cruor collection system enhances blood transfusion safety because it provides better ways of handling the various processes in blood bank. Cruor collection system offers donor to enter the data through simple and interactive forms. This is very helpful for the hospital to enter the desired information through so much simplicity.

9.2 FUTURE ENHANCEMENT

This project is focused on maintain the donors record Support of various regional languages for better reach. The future enhancements can be done by developing the database backup and database maintenance activities and also developing portability of all operating systems.

CHAPTER 10

APPENDIX

10.1. SOURCE CODE

Index.html

```
<!DOCTYPE html>
<html>
<head>
<title>CRUOR COLLECTION SYSTEM</title>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: left;
}
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align: right;
}
```

```
li a:hover {
  background-color: #111;
}
marquee{
   font-size:25px;
  font-style: oblique;
  font-weight: bold;
}
p{
  font-size:25px;
  font-style: oblique;
  font-weight: bold;
}
h1 {
  color: red ;
  font-style: oblique;
  font-size:43px;
}
h2 {
  color: blue;
  font-style: oblique;
  font-size:33px;
}
</style>
</head>
<body>
<ul>
      <li><a href="index.html">HOME</a></li>
      <li><a href="register.html">REGISTRATION</a></li>
      <li><a href="conditions.html">CONDITIONS</a></li>
```

```
        <li><a href="login.html">LOGIN</a></li>
        <li><a href="request.html">MESSAGES</a></li>
</ul>
</head>
<body style="background-image:url('first.jfif');background-repeat:no-
repeat;background-size:cover;">
<h1 align="center">Welcome To Give The Gift Of Life</h1>
<style>
* {box-sizing: border-box}
body {font-family: Verdana, sans-serif; margin:0}
.mySlides {display: none}
img {vertical-align: middle;}
/* Slideshow container */
.slideshow-container {
  max-width: 500px;
  position: relative;
  margin: auto;
}
/* Next & previous buttons */
.prev, .next {
  cursor: pointer;
  position: absolute;
  top: 50%;
  width: auto;
  padding: 16px;
  margin-top: -22px;
  color: white;
  font-weight: bold;
  font-size: 18px;
  transition: 0.6s ease;
```

```
  border-radius: 0 3px;
  user-select: none;
}
/* Position the "next button" to the right */
.next {
  right: 0;
  border-radius: 3px 0 0 3px;
}
/* On hover, add a black background color with a little bit see-through */
.prev: hover, .next:hover {
  background-color: rgba(0,0,0,0.8);
}
/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* Number text (1/3 etc) */
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}
```

```
/* The dots/bullets/indicators */
.dot {
  cursor: pointer;
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}
.active, .dot:hover {
  background-color: #717171;
}
/* Fading animation */
.fade {
  animation-name: fade;
  animation-duration: 1.5s;
}
@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}
/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .prev, .next,.text {font-size: 11px}
}
body {
  font-family: Arial, Helvetica, sans-serif;
  margin: 0;
```

```
}
html {
  box-sizing: border-box;
}
*, *:before, *:after {
  box-sizing: inherit;
}
.column {
  float: left;
  width: 33.3%;
  margin-bottom: 16px;
  padding: 0 8px;
}
.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  margin: 8px;
}
.about-section {
  padding: 50px;
  text-align: center;
  background-color: #474e5d;
  color: white;
}
.container {
  padding: 0 16px;
}
.container::after, .row::after {
  content: "";
  clear: both;
  display: table;
```

```
}
.title {
  color: grey;
}
.button {
  border: none;
  outline: 0;
  display: inline-block;
  padding: 8px;
  color: white;
  background-color: #000;
  text-align: center;
  cursor: pointer;
  width: 100%;
}
.button:hover {
  background-color: #555;
}
@media screen and (max-width: 650px) {
  .column {
    width: 100%;
    display: block;
  }
}
</style>
</head>
<body>
<div class="slideshow-container">
<div class="mySlides fade">
  <div class="numbertext">1 / 3</div>
```

```
  <img src="blood.jfif" style="width:100%">
  <div class="text">Caption Text</div>
</div>
<div class="mySlides fade">
  <div class="numbertext">2 / 3</div>
  <img src="okpat.jfif" style="width:100%">
  <div class="text">Caption Two</div>
</div>
<div class="mySlides fade">
  <div class="numbertext">3 / 3</div>
  <img src="doluu.jfif" style="width:100%">
  <div class="text">Caption Three</div>
</div>

<a class="prev" onclick="plusSlides(-1)">❮</a>
<a class="next" onclick="plusSlides(1)">❯</a>

</div>
<br>

<div style="text-align:center">
  <span class="dot" onclick="currentSlide(1)"></span>
  <span class="dot" onclick="currentSlide(2)"></span>
  <span class="dot" onclick="currentSlide(3)"></span>
</div>

<script>
let slideIndex = 1;
showSlides(slideIndex);
```

```
function plusSlides(n) {
  showSlides(slideIndex += n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

function showSlides(n) {
  let i;
  let slides = document.getElementsByClassName("mySlides");
  let dots = document.getElementsByClassName("dot");
  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = slides.length}
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
}
</script>
<marquee><b>Every drop of blood is like a breath for some one!Donate
blood</b></marquee>
<h2 align="left">OUR SERVICES</h1>
<p align="left">Cruor collection system always available to serve blood needed
peoples 24/7. <br>Due to the shortage of blood,we developed this service.<br>We
have the donors information of blood availability.<br>Wastage of blood will be
```

```
avoided.</p>
<div class="about-section">
  <p>Contact us : cruorcollection06@gmail.com</p>
  <p>Anybody can give Blood.</p>
  <p>Tears of A Mother cannot Save Her Child But Your Blood Can Save
Them.</p>
</div>
</body>
</html>
```

Register.html

```
<!Doctype html>
<html>
<head>
<title>register form</title>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
li {
  float: left;
}
li a {
  display: block;
  color: white;
```

```
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align: center;
  button align: center;
}
li a: hover {
  background-color: #111;
}
button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}
</style>
</head>
<bodystyle="background-image:url('poki.jfif');background-repeat:no-
repeat;background-size:cover;">
<ul>
      <li><a href="index.html">HOME</a></li>
      <li><a href="register.html">REGISTRATION</a></li>
      <li><a href="conditions.html">CONDITIONS</a></li>
```

```
<li><a href="login.html">LOGIN</a></li>
<li><a href="request.html">MESSAGES</a></li></ul>
<form action="thank.html" onsubmit="myFunction()">
<h1><center>REGISTRATION</center></h1><br>
<p align="center">
<lable>First Name:</lable>
<input type="variable"name="firstname" required><br><br>
<lable>Last Name:</lable>
<input type="variable"name="lastname" required><br><br>
<lable>GENDER:</lable><br>
<lable>MALE<Input type="radio"name="gender"><br>
<lable>FEMALE<Input type="radio"name="gender"><br>
<lable>OTHER<Input type="radio"name="gender"><br>
<lable>age:</lable>
<Input type="Number"required/><br><br>
<lable>Mobile Number:</lable>
<Input type="Number"required/><br><br>
<lable>email ID:</lable>
<inputtype="email"name="email"pattern=".+@gmail.com"size="30"
required><br><br>
<label>Blood group:</lable>
<select>
<option name="blood"/>O+
<option name="blood"/>A+
<option name="radio"name="blood"/>B+
<option name="checkbox"name="blood"/>AB+
<option name="checkbox"name="blood"/>O-
<option name="checkbox"name="blood"/>A-
<option name="checkbox"name="blood"/>AB-
```

```
<option name="checkbox"name="blood"/>B-
</select>
<br><br>
<label>Address</label><br>
<textarea colm="10" row="5">
</textarea><br><br>
<button type="submit">SUBMIT</button>
</p>
<script>
function myFunction() {
  alert("The form was submitted");}
</script>
</body></html>
```

Login.html

```
<!Doctype html>
<html>
<head>
<title>Login form</title>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
li {
  float: left;
}
```

```
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align:right;
}
li a:hover {
  background-color: #111;
}
input[type=submit]{
background-color:#04AA6D;
padding:16px 32px;
margin:4px 2px;
cursor:pointer;
align:center;
}
</style>
</head>
<bodystyle="background-image:url('vivosa.jfif');background-repeat:no-repeat;>
<ul>
      <li><a href="index.html">HOME</a></li>
      <li><a href="register.html">REGISTRATION</a></li>
      <li><a href="conditions.html">CONDITIONS</a></li>
      <li><a href="login.html">LOGIN</a></li>
      <li><a href="request.html">MESSAGES</a></li>
</ul>
<form action="thank.html" onsubmit="myFunction()">
```

```
<h1 align="center">LOGIN FORM</h1>
<p align="center">
<lable>USER NAME:</lable>
<Input type="First Name"required><br><br>
<lable>Password:</lable>
<Input type="password"/><br><br>
<input type ="submit"/>
</form>
<script>
function myFunction() {
  alert("The form was submitted");
}
</script></p>
</body>
</html>
```

Information.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>List of donors</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
<linkrel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.
   min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <script
   src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<style>
ul {
```

```
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
li {
  float: left;
}
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align:right;
}
li a:hover {
  background-color: #111;
}
button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
 text-decoration: none;
  display: inline-block;
  font-size: 16px;
```

```
 margin: 4px 2px;
 cursor: pointer;
}
</style>
</head>
<bodystyle="background-image:url('tharu.jfif');background-repeat:no-repeat;>
<ul>
      <li><a href="index.html">HOME</a></li>
      <li><a href="register.html">REGISTRATION</a></li>
      <li><a href="conditions.html">CONDITIONS</a></li>
      <li><a href="login.html">LOGIN</a></li>
      <li><a href="request.html">MESSAGES</a></li>
</ul>
<h1>DONORS INFORMATION</h1>
<div class="container">
 <table class="table">
   <thead>
     <tr>
       <th>Donar ID</th>
       <th>Donarname</th>
       <th>Age</th>
       <th>BloodGroup</th>
       <th>PhoneNo</th>
       <th>Email</th>
     </tr>
   </thead>
   <tbody>
     <tr class="0111">
       <td>0111</td>
```

```
  <td>Pooja</td>
  <td>20</td>
  <td>AB+</td>
  <td>985363544</td>
  <td>pooja2@gmail.com</td>
</tr>
<tr class="0112">
  <td>0112</td>
  <td>SasiKumar</td>
  <td>22</td>
  <td>A+</td>
  <td>982854557</td>
  <td>sasikumar3@gmail.com</td>
</tr>
<tr class="0113">
  <td>0113</td>
  <td>SasiKeerthana</td>
  <td>21</td>
  <td>B+</td>
  <td>8465697856</td>
  <td>sasi5@gmail.com</td>
</tr>
<tr class="0114">
  <td>0115</td>
  <td>Tharini</td>
  <td>24</td>
  <td>O+</td>
  <td>953251857</td>
  <td>tharini2@gmail.com</td>
```

```
      </tr>
      <tr class="0115">
        <td>0115</td>
        <td>Nivi</td>
        <td>26</td>
        <td>AB-</td>
        <td>9321145673</td>
        <td>nivi7@gmail.com</td>
      </tr>
    </tbody>
  </table>
</div>
<p align="center">
<button type="button1"><a href="Send Request.html">Send Request</a></button>
<button type="button2"><a href="respond.html">Send Request</a></button>
</p>
</body>
</html>
```

Reply.html

```
<!DOCTYPE html>
<head>
  <title>Send Reply</title>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
```

```
  background-color: #333;
}
li {
  float: left;
}
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align:right;
}
li a:hover {
  background-color: #111;
}
button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
 text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
.block {
  display: block;
```

```
  width: 100%;
  border: none;
  background-color: #04AA6D;
  color: white;
  padding: 14px 28px;
  font-size: 16px;
  cursor: pointer;
  text-align: center;
}
.block:hover {
  background-color: #ddd;
  color: black;
</style>
</head>
<bodystyle="background-image:url('mesge.jfif');background-repeat:no-repeat;>
<ul>
      <li><a href="index.html">HOME</a></li>
      <li><a href="register.html">REGISTRATION</a></li>
      <li><a href="conditions.html">CONDITIONS</a></li>
      <li><a href="login.html">LOGIN</a></li>
      <li><a href="request.html">MESSAGES</a></li>
</ul>
<p align="center">
<label>Donor:</label>
<Input type="First Name"/><br><br>
<label>Phone Number:</label>
<Input type="number"/><br><br>
<label>Enter Message</label><br>
<textarea rows="4" cols="50" name="comment" form="usrform">
```

```
Enter text here...</textarea>
</textarea><br><br>
<button class="block">SEND ME</button>
</body>
</head>
```

Request.html

```
<!DOCTYPE html>
<head>
  <title>Send Request</title>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
li {
  float: left;
}
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align:right;
}
li a:hover {
```

```
  background-color: #111;
}
button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
 text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
.block {
  display: block;
  width: 100%;
  border: none;
  background-color: #04AA6D;
  color: white;
  padding: 14px 28px;
  font-size: 16px;
  cursor: pointer;
  text-align: center;
}
.block:hover {
  background-color: #ddd;
  color: black;
</style>
</head>
```

```
<bodystyle="background-image:url('mesge.jfif');background-repeat:no-repeat;">
<ul>
    <li><a href="index.html">HOME</a></li>
    <li><a href="register.html">REGISTRATION</a></li>
    <li><a href="conditions.html">CONDITIONS</a></li>
    <li><a href="login.html">LOGIN</a></li>
    <li><a href="request.html">MESSAGES</a></li>
</ul>
<p align="center">
<lable>Donar:</lable>
<Input type="First Name"/><br><br>
<lable>Phone Number:</lable>
<Input type="number"/><br><br>
<lable>Enter Message</lable><br>
<textarea rows="4" cols="50" name="comment" form="usrform">
Enter text here...</textarea>
</textarea><br><br>
<button class="block">SEND ME</button>
</body>
</head>
```

Submit.html

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
```

```
  overflow: hidden;
  background-color: #333;}
li {
  float: left;}
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align:center;
  button align:center;
}
li a:hover {
  background-color: #111;
}
</style>
</head>
<bodystyle="background-image:url('than.png');background-repeat:no-repeat;background-size:cover;">
<ul>
    <li><a href="index.html">HOME</a></li>
    <li><a href="register.html">REGISTRATION</a></li>
    <li><a href="conditions.html">CONDITIONS</a></li>
    <li><a href="login.html">LOGIN</a></li>
    <li><a href="request.html">MESSAGES</a></li></ul>
<h1 align="center">INFORMATION STORED<h1>
```

Every blood donor is an asset to the nation who saves people’s lives, and you’re one of them. Thank you for your regular contribution to our blood donation drive. We appreciate your efforts. Thank you. </body>

</html>

Condition.html

```
<!Doctype html>
<html>
<head>
<title>register form</title>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
li {
  float: left;
}
li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  align:right;}
li a:hover {
  background-color: #111;}
```

```
input[type=submit]{
background-color:#04AA6D;
padding:16px 32px;
margin:4px 2px;
cursor:pointer;
align:center;
}
p{
  font-size:23px;
  font-style: oblique;
  font-weight: bold;
}
h4 {
  color: red;
  font-weight: bold;}
h1 {
  color:green ;
  font-style: oblique;
  font-size:43px;}
</style>
</head>
<bodystyle="background-image:url('sear.jfif');background-repeat:no-repeat;">
<ul>
      <li><a href="index.html">HOME</a></li>
      <li><a href="register.html">REGISTRATION</a></li>
      <li><a href="conditions.html">CONDITIONS</a></li>
      <li><a href="login.html">LOGIN</a></li>
      <li><a href="request.html">MESSAGES</a></li>
</ul>
```

<h1 align="center">BLOOD DONATION</h1>

<h4>AGE</h4>

You are aged between 18 and 65.

* In some countries national legislation permits 16–17 year-olds to donate provided that they fulfil the physical and hematological criteria required and that appropriate consent is obtained.

* In some countries, regular donors over the age of 65 may be accepted at the discretion of the responsible physician. The upper age limit in some countries are 60.

<h4>WEIGHT</h4>

you weigh at least 50 kg.

* In some countries, donors of whole blood donations should weigh at least 45 kg to donate 350 ml ± 10% .

<h4>HEALTH</h4>

You must be in good health at the time you donate.

You cannot donate if you have a cold, flu, sore throat, cold sore, stomach bug or any other infection. If you have recently had a tattoo or body piercing you cannot donate for 6 months from the date of the procedure. If the body piercing was performed by a registered health professional and any inflammation has settled completely, you can donate blood after 12 hour. If you have visited the dentist for a minor procedure you must wait 24 hours before donating; for major work wait a month. You must not donate blood If you do not meet the minimum hemoglobin level for blood donation

* A test will be administered at the donation site. In many countries, a hemoglobin level of not less than 12.0 g/dl for females and not less than 13.0 g/dl for males as the threshold.

<h4>TRAVEL</h4>

Travel to areas where mosquito-borne infections are endemic, e.g. malaria, dengue and Zika virus infections, may result in a temporary deferral.

Many countries also implemented the policy to defer blood donors with a history of travel or residence for defined cumulative exposure periods in specified countries or areas, as a measure to reduce the risk of transmitting variant Creutzfeldt-Jakob Disease (vCJD) by blood transfusion.

<h4>PREGNANCY</h4>

Following pregnancy, the deferral period should last as many months as the duration of the pregnancy.

It is not advisable to donate blood while breast-feeding. Following childbirth, the deferral period is at least 9 months (as for pregnancy) and until 3 months after your baby is significantly weaned (i.e. getting most of his/her nutrition from solids or bottle feeding).</h4>

<h4>BLOOD DONATION REQUIREMENTS</h4>

To donate blood, Lifeblood donors must:

Be healthy and not suffering from a cold, flu or other illness at the time of donation.

Be aged between 18 and 75 years (other rules may apply if you are a current donor).

Weigh at least 50kg.

Have normal temperature and blood pressure.

Additional screening measures are in place due to the coronavirus (COVID-19) pandemic.

Meet guidelines designed to protect both the donor and the people who will receive the blood.

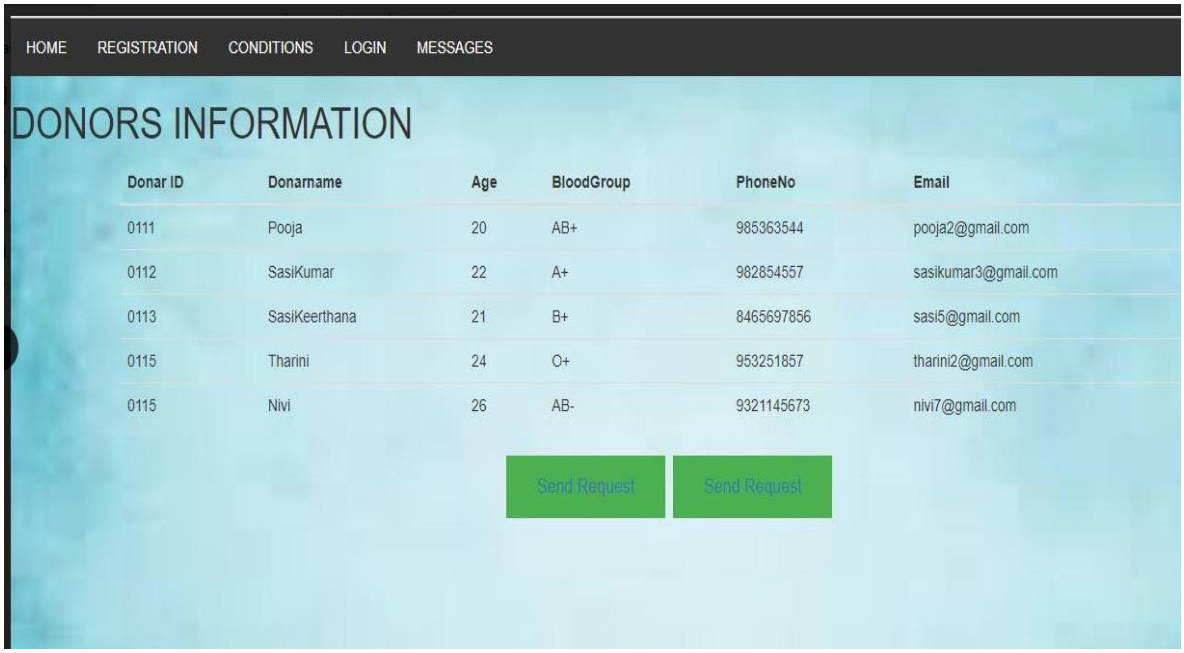
</body>

</html>

10.2 SCREENSHOTS

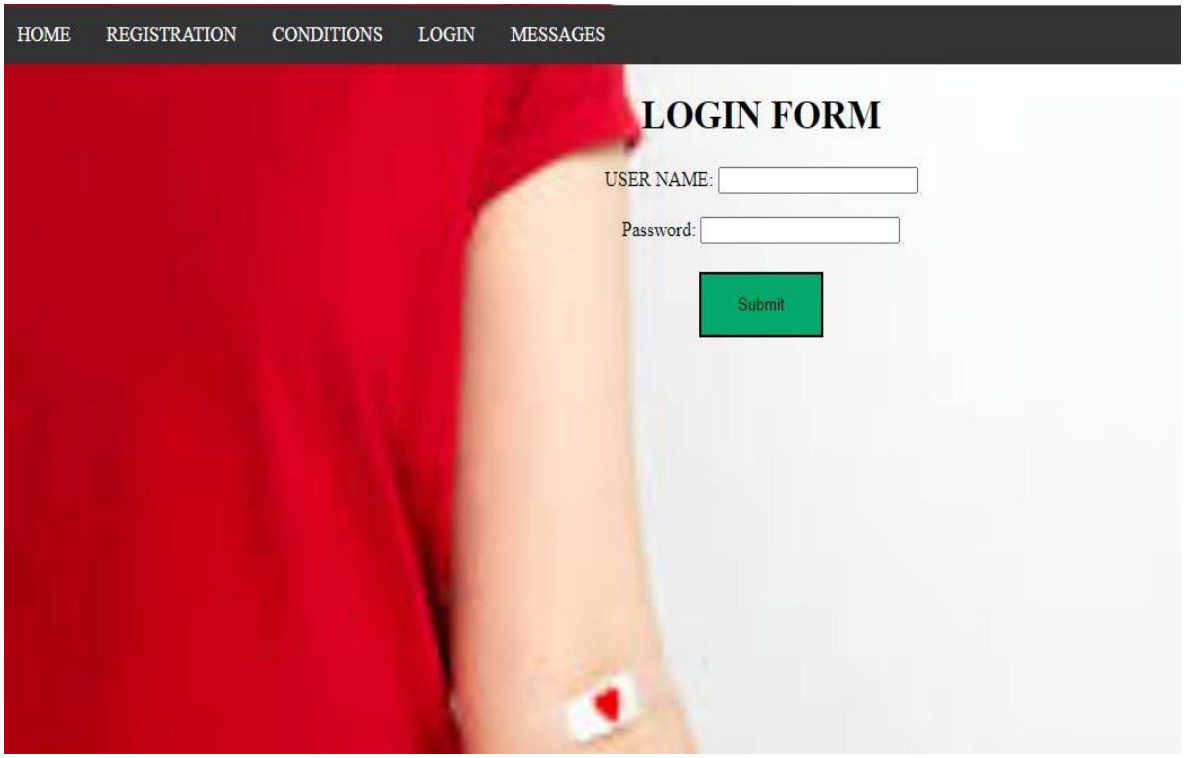
10.2.1 Home screen

10.2.2 Donor information

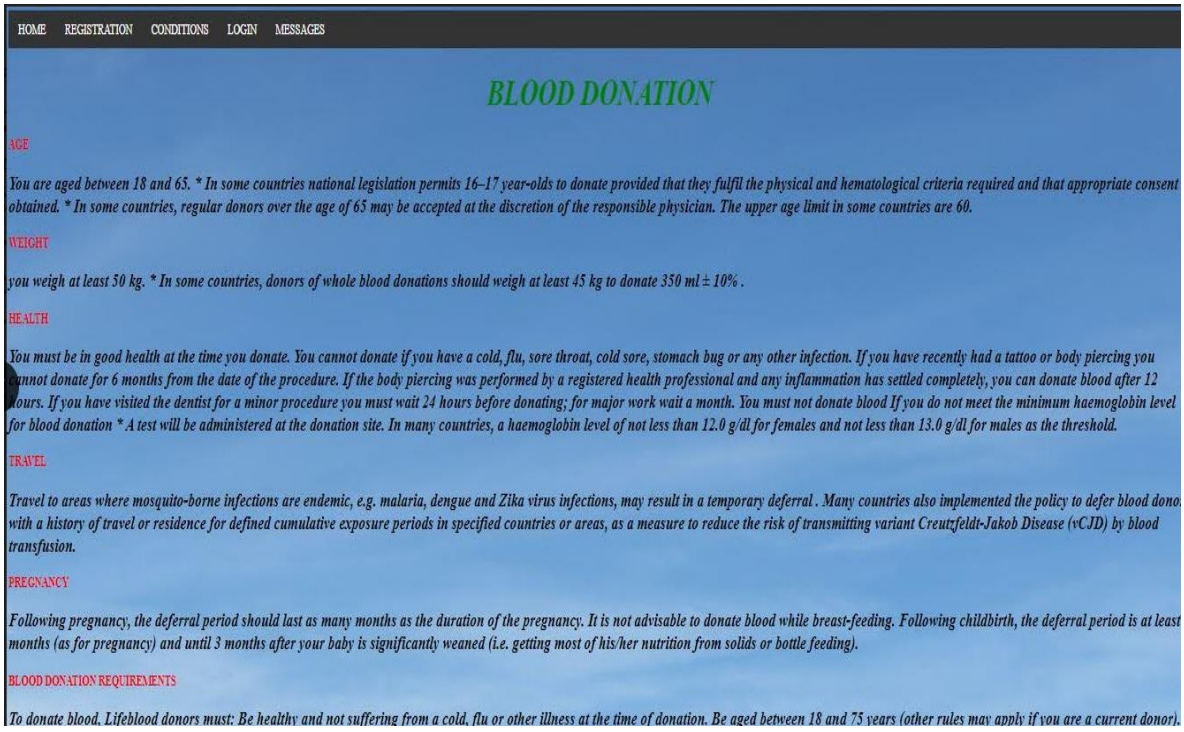


10.2.3 Registration

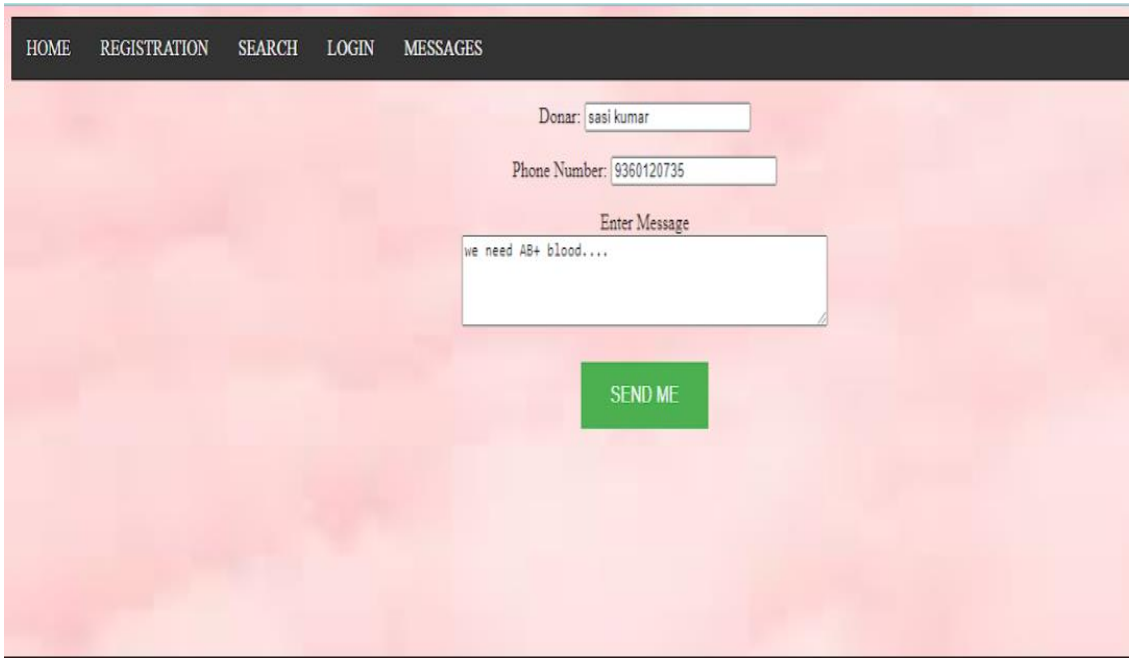
10.2.4 Login form

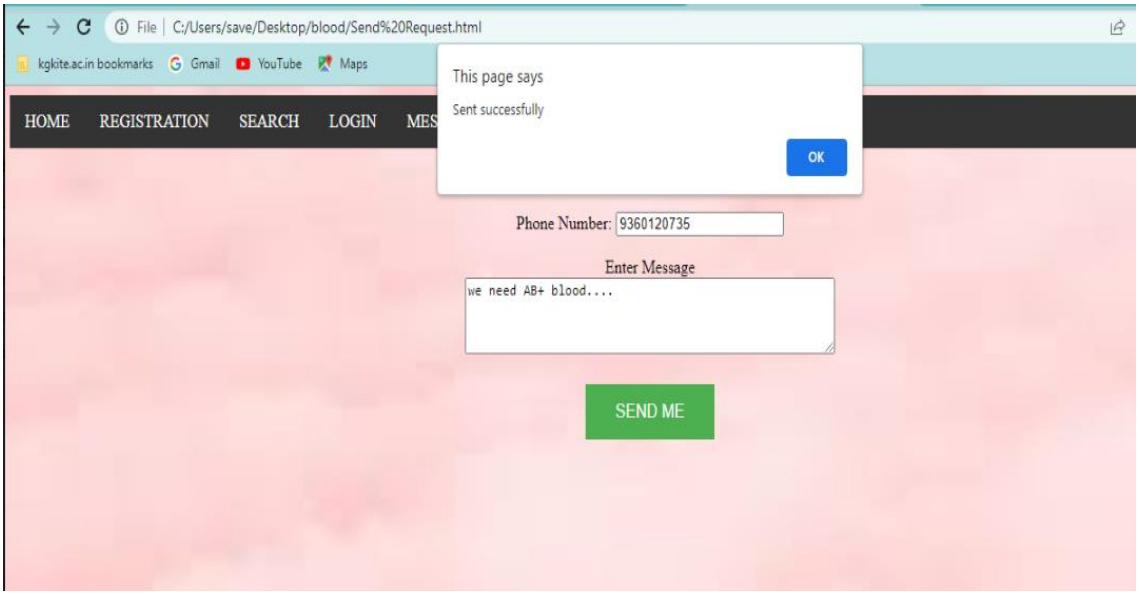


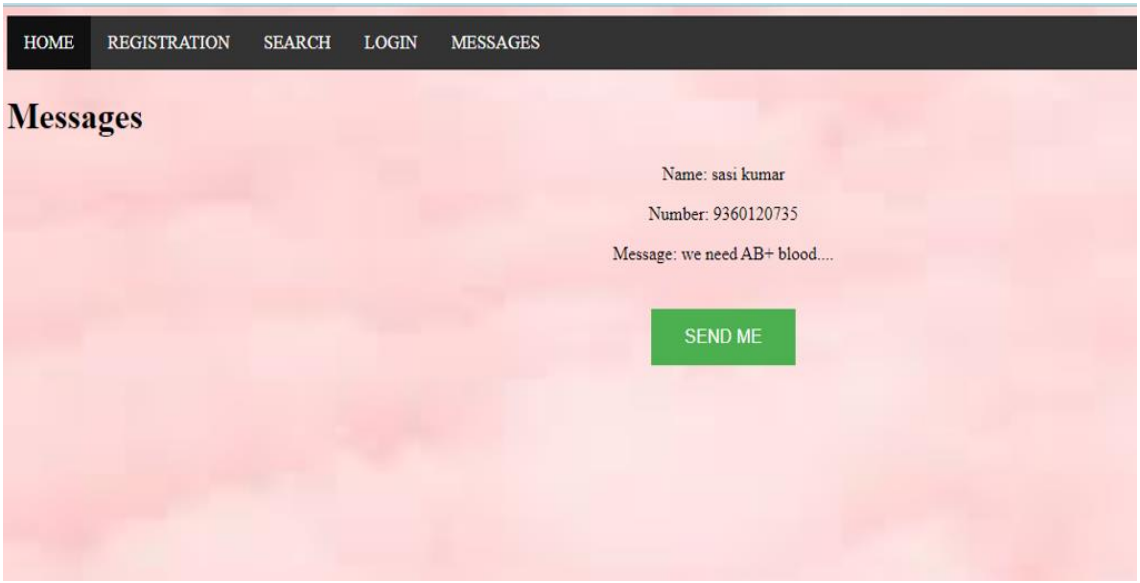
10.2.5 Conditions



10.2.6 Message Alert




File | C:/Users/save/Desktop/blood/Send%20Request.html
kgkite.ac.in bookmarks
Gmail
YouTube
Maps
This page says
Sent successfully
OK
HOME
REGISTRATION
SEARCH
LOGIN
Phone Number: 9360120735
Enter Message
we need AB+ blood....
SEND ME


HOME
REGISTRATION
SEARCH
LOGIN
MESSAGES
Messages
Name: sasi kumar
Number: 9360120735
Message: we need AB+ blood....
SEND ME

CHAPTER 11

REFERENCES

[1] Michael Chau, Eddie Cheng and Chi Wai Chan, Data Analysis for Healthcare: A Case Study in Blood Donation Center Analysis. Proceedings of Sixteenth Americas Conference on Information Systems (AMICS), 2019.

[2] Shyam Sundaram and T Santhanam, Classification of Blood Donors using Data Mining. Proceedings of the Semantic E-Business and Enterprise Computing, pp, 2019.

[3] Li B. N., & Dong, M. C. Banking on blood, Computing and Control Engineering (August–September), 22–25, 2020.

[4] Glynn S A, Kleinman S H, Schreiber G B, Zuck T, McCombs S, Bethel J, Motivations to donate blood: demographic comparisons, Transfusion, 2021.

[5] Pawar R, Thigale P, Walekar G, Thakar and D. Joshi, 2019-Optimal Facility for Location Tracking of Blood Bank and Donor. International Research Journal of Engineering and Technology 2020.

[6] Parikh S, P. Kathiria, Y. Vaghela, H. Shah and D. Dholakiya, 2018. AGeo-Location based Mobile Service that Dynamically Locates and Notifies the nearest Blood Donors for Blood Donation during Medical Emergencies. International Journal of Computer Applications 2018.

[7] Guide to the preparation, use and quality assurance of blood components, 16th edition. Strasbourg: Council of Europe 2019.

[8] Boulton F. Evidence-based criteria for the care and selection of blood donors, with some comments on the relationship to blood supply, and emphasis on the management of donation-induced iron depletion. Transfusion Medicine 2021.

[9] Newman B. Iron depletion by whole-blood donation harms menstruating females: the current whole-blood-collection paradigm needs to be changed. Transfusion 2018.

[10] Hillyer C Comparable safety of blood donation in high-risk autologous donors versus non-high-risk donors and directed donors in a hospital setting. American Journal of Clinical Pathology 2020