**ADBMS**

# BLOOD BANK MANAGEMENT SYSTEM


## Bachelor of Technology (Computer Science)

**(MAY 2021)**

**Submitted By:**

**Guneet Kohli ( 1805172 ) D3 CSE A**

**Jashanpreet Kaur ( 1805188 ) D3 CSE A**

**Mayank Thakur ( 1805201 ) D3 CSE A**

UNDER THE GUIDANCE OF ER. PRIYANKA ARORA

GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA

# ACKNOWLEDGEMENT

We express our sincere regard and indebtedness to our project guide Er. Priyanka Arora, for his valuable time, guidance, encouragement, support and cooperation throughout the duration of our project. We would sincerely like to thank the CSE Department for giving the opportunity to work on enhancing our technical skills while undergoing this project. This project was done under the guidance of Er. Priyanka Arora. This project helped in understanding the various parameters which are involved in database management systems.

We would like to thank Dr. Parminder Singh, Head of Department and whole department for their support.

Guneet Kohli ( 1805172 )

Jashanpreet Kaur ( 1805188 )

Mayank Thakur ( 1805201 )

# ABSTRACT

The purpose of this study was to develop a blood management information system to assist in the management of blood donor records and ease/or control the distribution of blood in various parts of the country based on the hospital's demands. Without quick and timely access to donor records creating market strategies for blood donation, lobbying and sensitization of blood donors becomes very di cult.The blood management information system or functionalities to quick access to donor records collected from various parts of the country. It enables monitoring of the results and performance of the blood donation activity such that relevant and measurable objectives of the organization can be checked.The reports generated by the system give answers to most of the challenges management faces as far as blood donor records are concerned.

The proposed Blood Bank App helps the people 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. They don’t need to go anywhere to search for blood when they need to. Our life is so busy so we don’t have time to spend going here and there, we can use technical way to search the blood by using the Blood Bank software we can and thousands of people who are donating the blood and also get the detail the of that person that in which city he belongs to and what is the Blood group of that person.

# INDEX

# CHAPTER 1 : INTRODUCTION

## 1.1 INTRODUCTION

The software system is an online blood bank management system that helps in managing various blood bank operations effectively. The project consists of a central repository containing various blood deposits available along with associated details. These details include blood type, storage area and date of storage. These details help in maintaining and monitoring the blood deposits. The project is an online system that allows to check weather required blood deposits of a particular group are available in the blood bank. Moreover the system also has added features such as patient name and contacts, blood booking and even need for certain blood group is posted on the website to find available donors for a blood emergency. This online system is supported by an SQL database to store blood and user specific details.

## 1.2 AIM

The main aim of developing this software is to provide blood to the people who are in need of blood. The numbers of persons who are in need of blood are increasing in large numbers day by day. Using this system, the user can search the blood group available in the city and he can also get the contact number of the donor who has the same blood group. Inorder to help people who are in need of blood, this Online Blood Bank software can be used effectively for getting the details of available blood groups and user can also get contact number of the blood donors having the same blood group and within the same city.

## 1.3 EXISTING SYSTEM

There are a quite good number of software packages that exist for BLOOD BANK Inventory control. I found that the existing system is limited only to those particular bloodbank.At the present there is no software to keep any records in blood banks. 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.Difficult to manage and maintain the files.Chance of damage of files, if the data is stored in the files for duration of time. Privacy is difficult. Time consuming is retervieng, storing and updating the data. It is difficult to keep track of the record about the donor & receiver he has donated or recieved the blood at the last time.

## 1.4 PROPOSED SYSTEM

The proposed system (Blood Bank Management System) is designed to help the Blood Bank administrator to meet the demand of Blood by sending and/or serving the request for Blood as and when required.The proposed system gives the procedural approach of how to bridge the gap between Recipient, Donor, and Blood Banks. This Application will provide a common ground for all the three parties (i.e. Recipient, Donor, and Blood Banks) and will ensure the fulfillment of demand for Blood requested by Recipient and/or Blood Bank.The features of proposed system are ease of data

entry , system should provide user friendly interfaces , no need to maintain any manual register and form , immediate data retrieval and so on. The new system covers all the aspects of the existing system as well as enhanced features for the existing system e.g. Bill Provision, etc.

## 1.5 FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts to the feasibility study.

1) Operational Feasibility

2) Technical Feasibility

3) Economical Feasibility

### 1.5.1 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed 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 system development.The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design.

### 1.5.2 TECHNICAL FEASIBILITY

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a

high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the needs of the proposed system.

### 1.5.3 ECONOMICAL FEASIBILITY

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected.

## 1.6 ORGANISATION OF THE REPORT

### 1.6.1 INTRODUCTION

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

### 1.6.2 SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

### 1.6.3 DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

### 1.6.4 IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Back-end development of the application.

### 1.6.5 RESULTS AND DISCUSSION

This section has screenshots of all the implementations and their description.

# CHAPTER 2: SOFTWARE REQUIREMENTS AND SPECIFICATION

## 2.1 Hardware Requirements

| Number | Description |
|---|---|
| 1 | PC with 512GB or more Hard disk. |
| 2 | PC with 2 GB RAM. |

## 2.2 Software Requirements

| Number | Description | Type |
|---|---|---|
| 1 | Operating System | Windows 10 |
| 2 | Language | PL/SQL |
| 3 | Database | MySQL |
| 4 | IDE | Oracle Live, APEX ORACLE |
| 5 | Browser | Google Chrome |

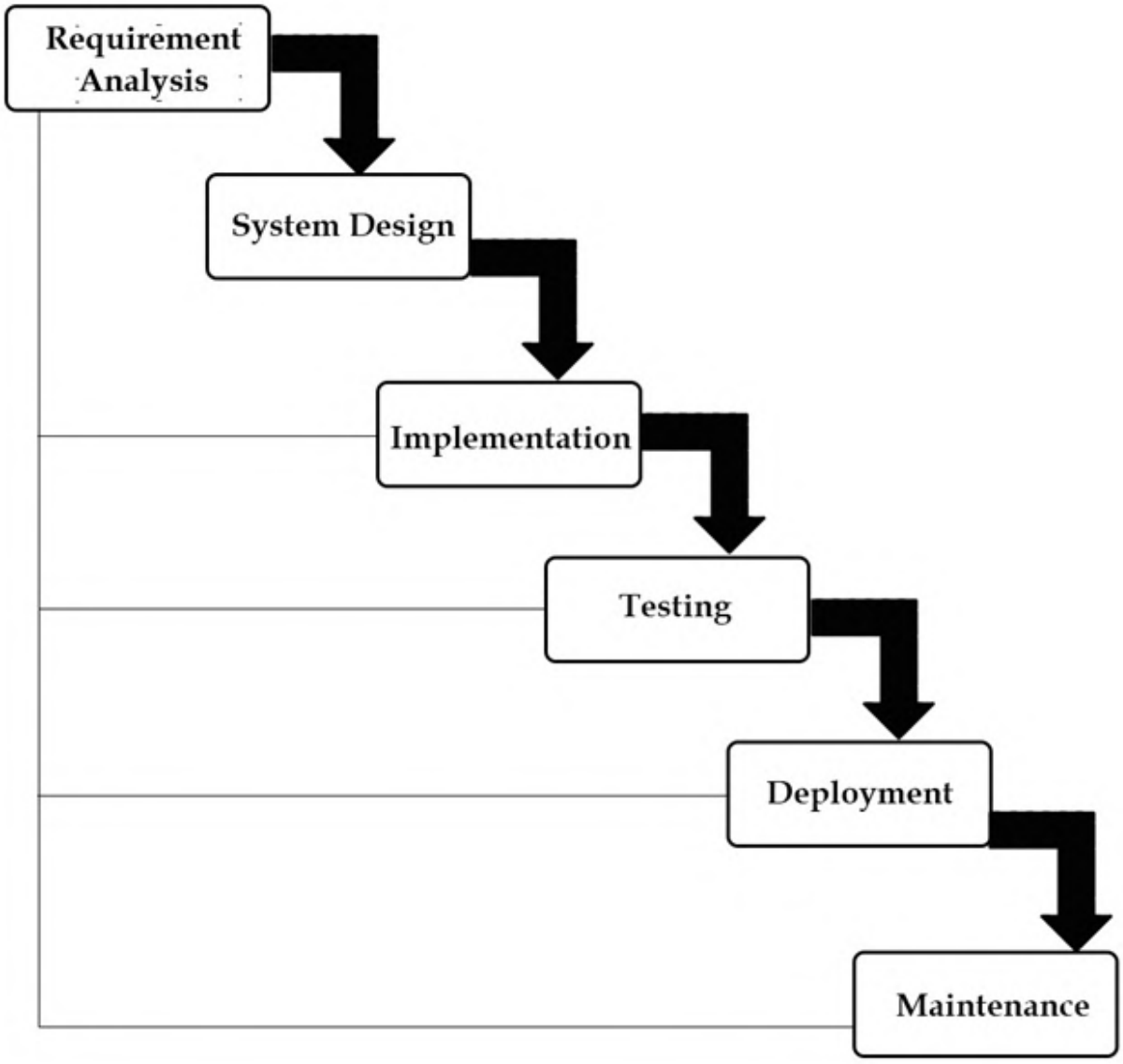
# CHAPTER 3 : DESIGN & PLANNING

## 3.1 Software Development Life Cycle Model

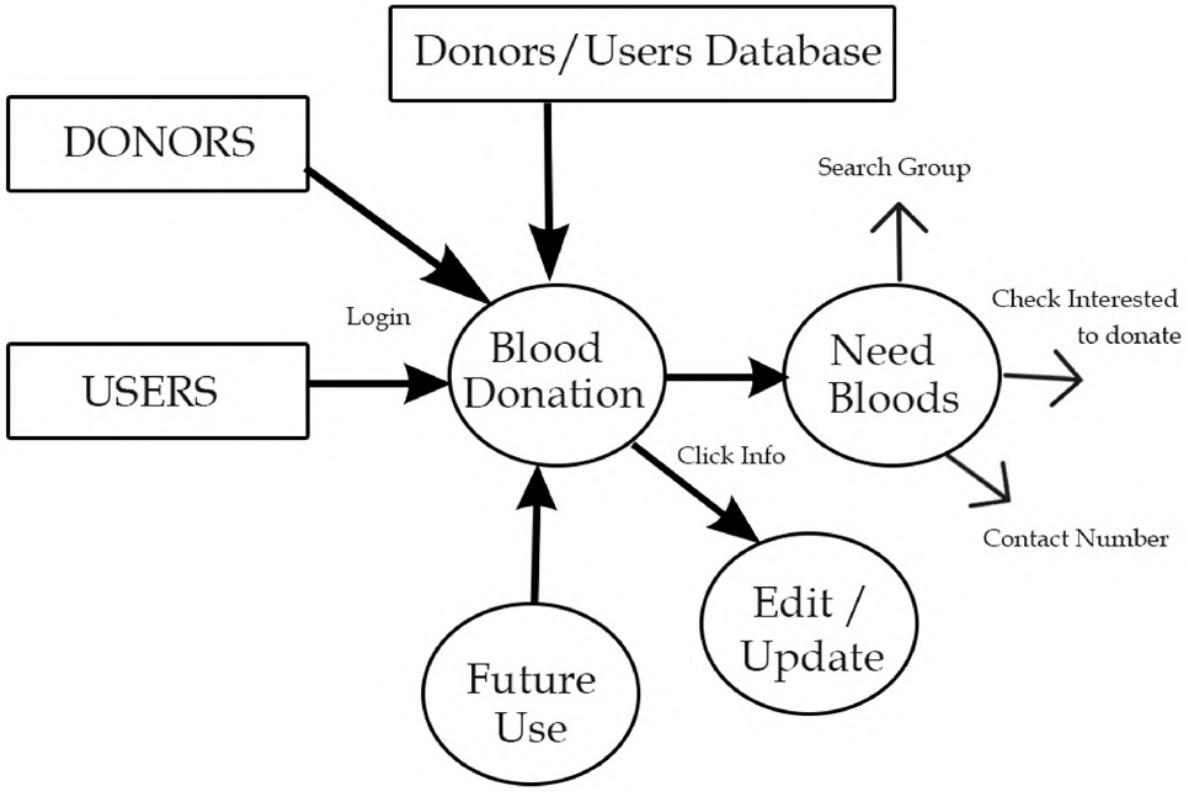
3.1.1 WATERFALL MODEL

The waterfall model was selected as the SDLC model due to the following reasons:

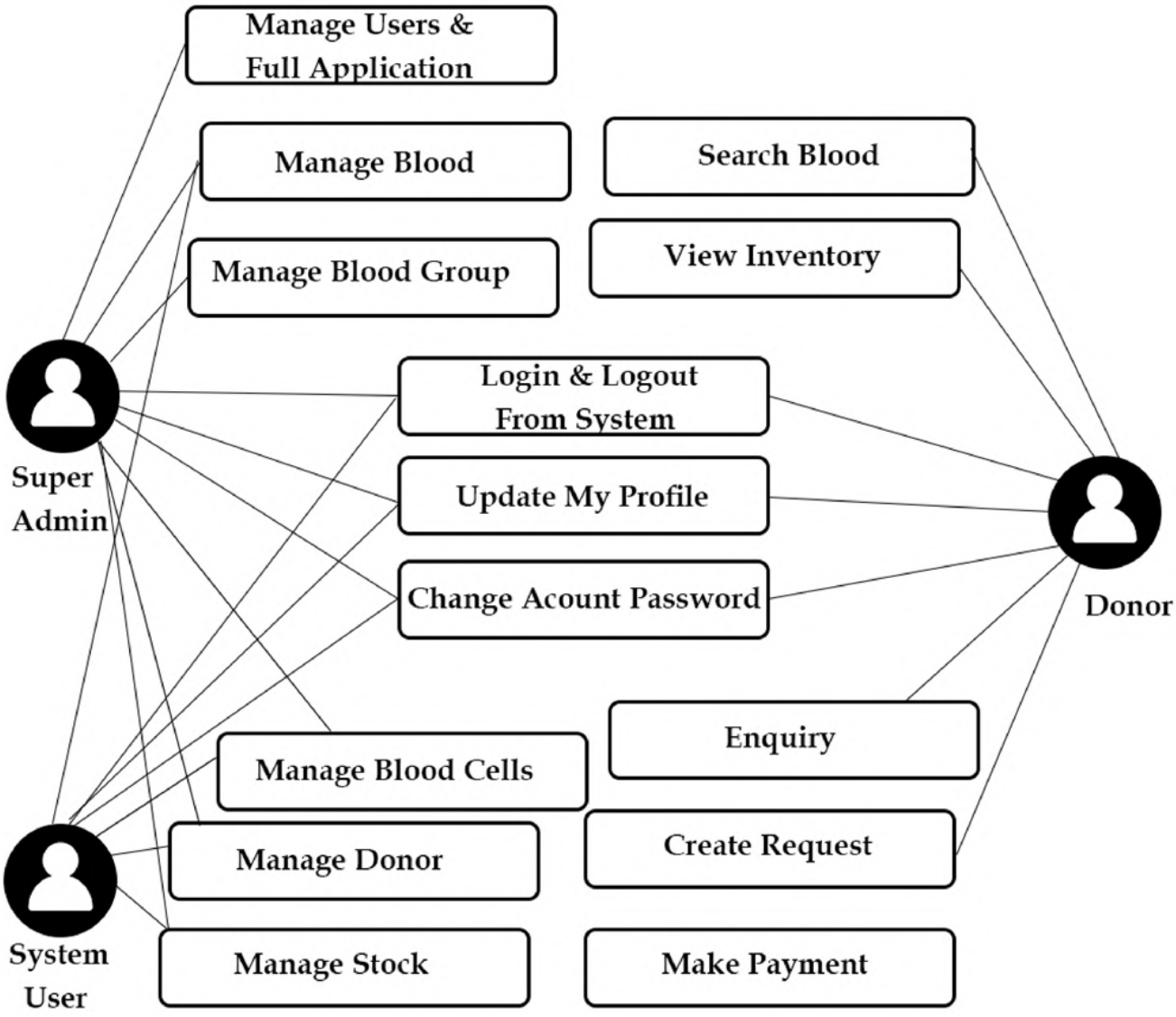
- Requirements were very well documented, clear and fixed.
- Simple and easy to understand and use.
- There were no ambiguous requirements.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Clearly defined stages.
- Well understood milestones.Easy to arrange tasks.


Requirement Analysis
System Design
Implementation
Testing
Deployment
Maintenance

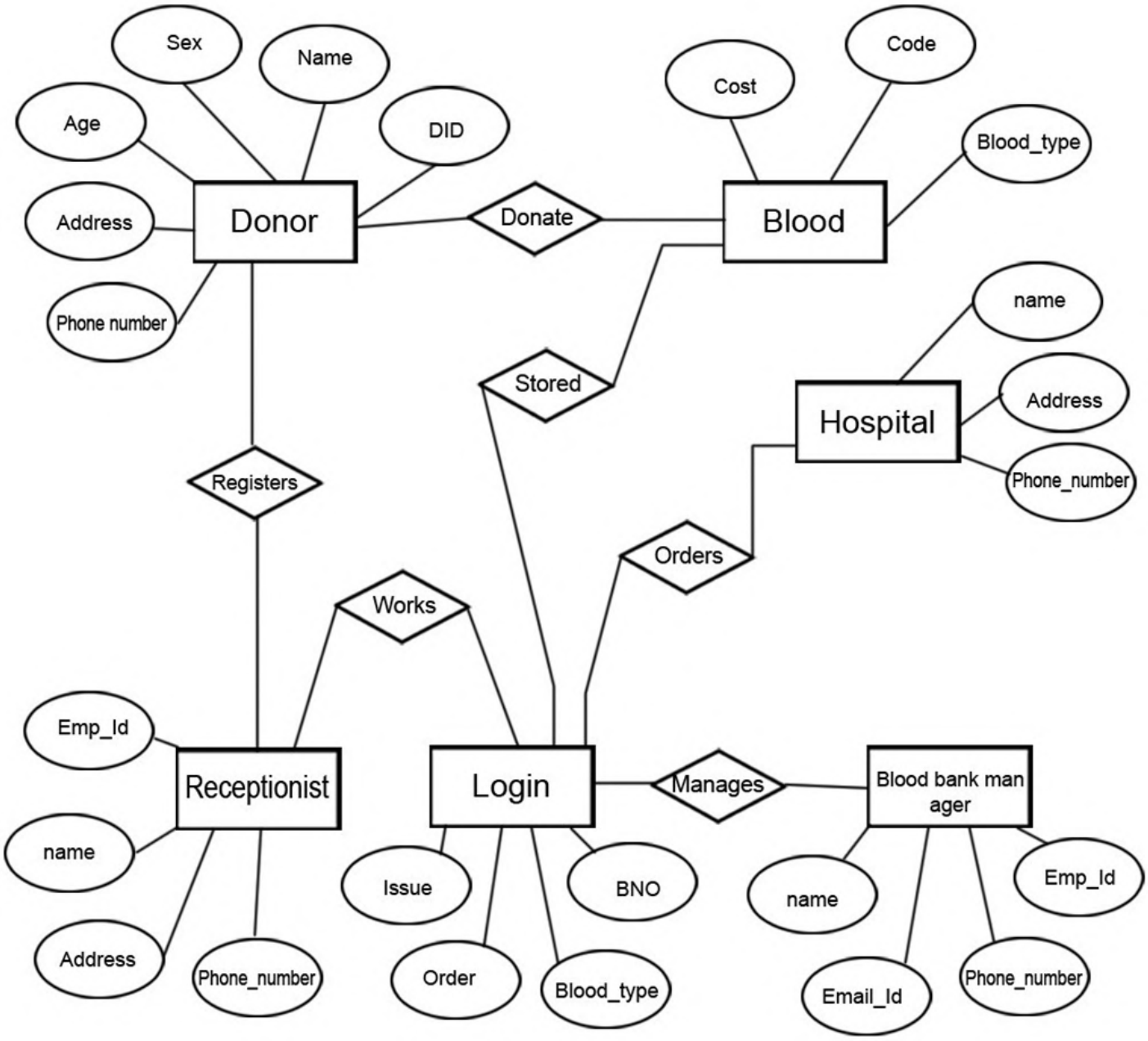
## 3.2 GENERAL OVERVIEW



## 3.3 USE CASE DIAGRAM

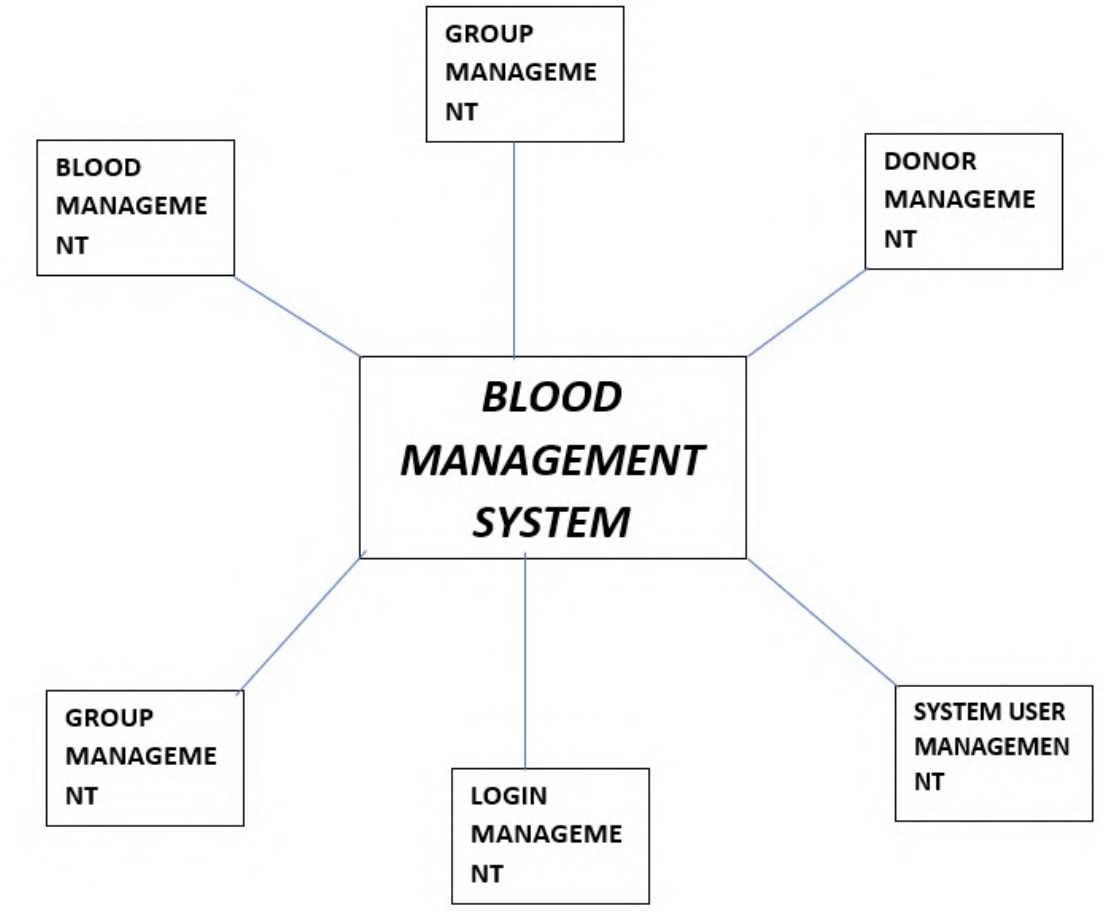


## 3.4 ER Diagram



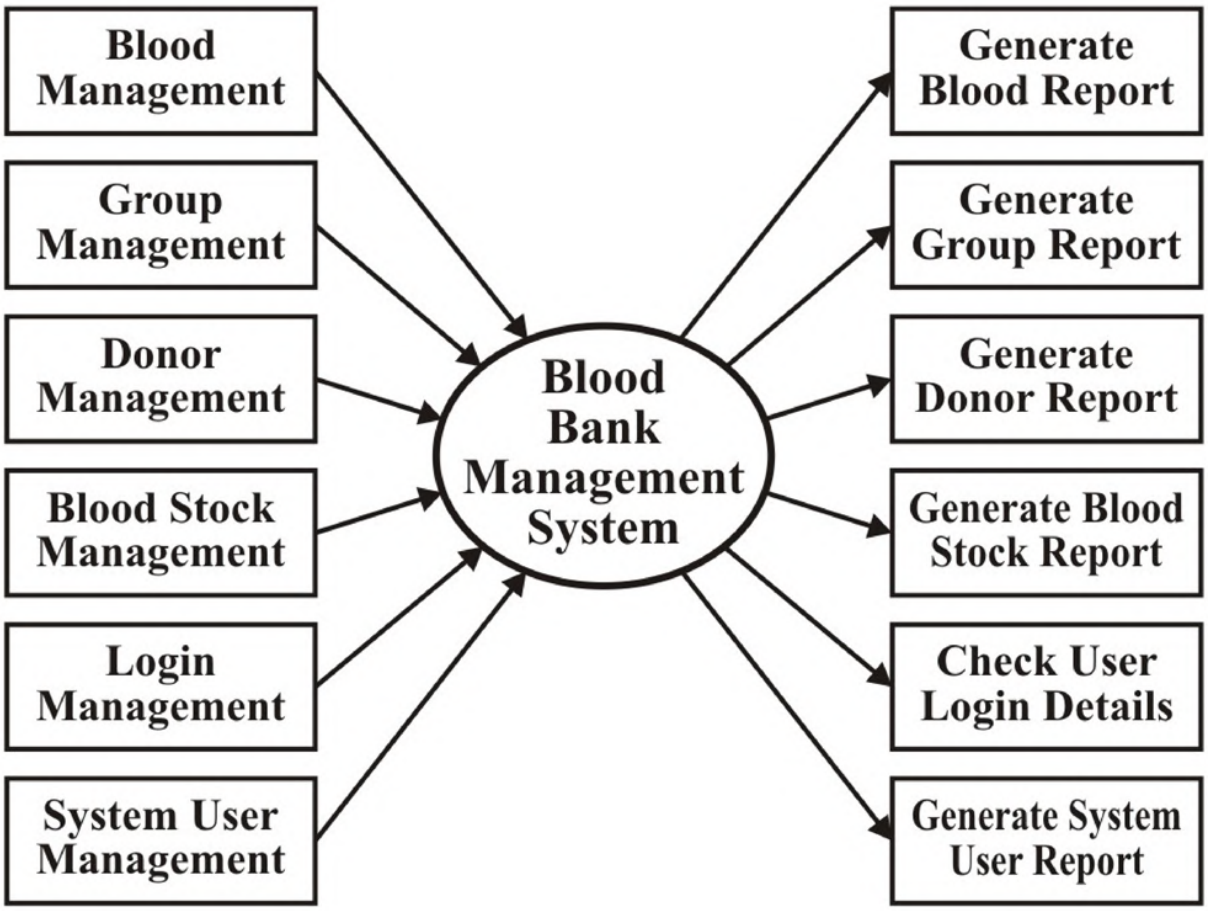
## 3.5 DFD Diagram

### 3.5.1 Zero-Level DFD Diagram

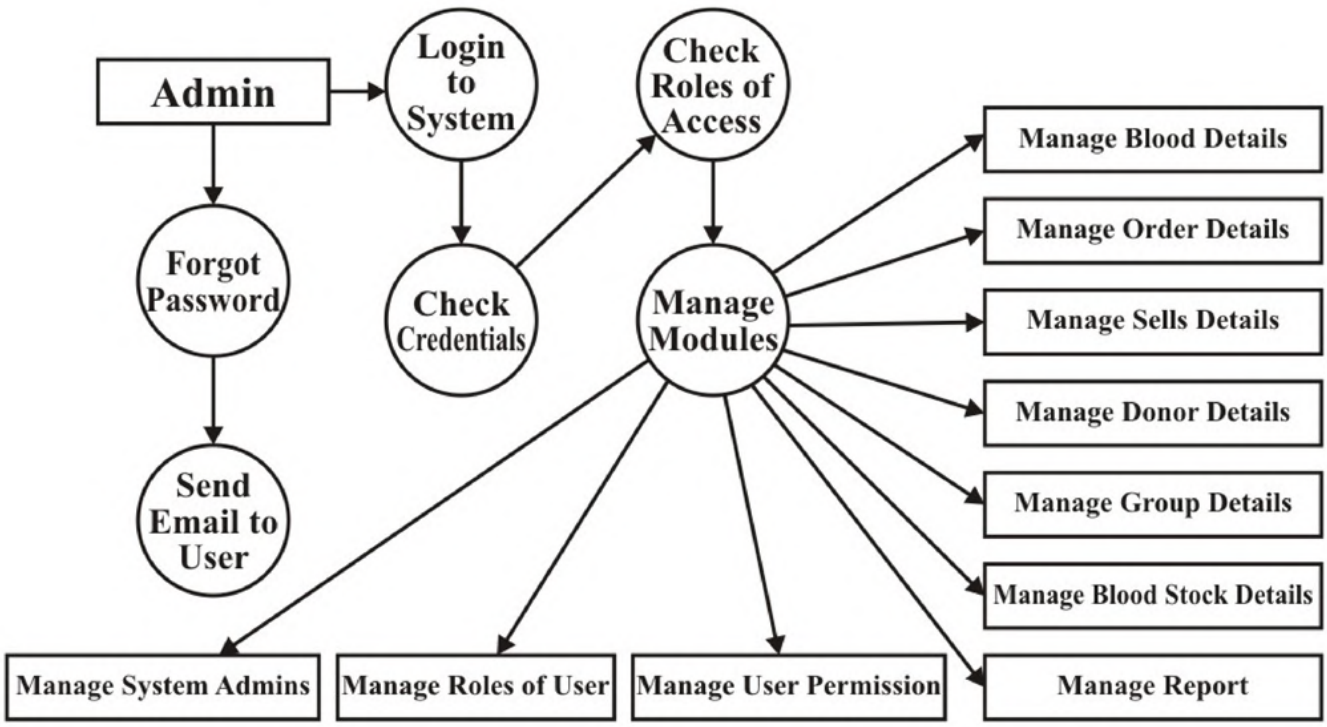



**ZERO LEVEL DFD- BLOOD BANK MANAGEMENT SYSTEM**

3.5.2 First-Level DFD Diagram



### 3.5.3 Second-Level DFD Diagram




**Second Level DFD - Blood Bank Management System**

# CHAPTER 4 : IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project.

## 4.1 Apex Oracle

Oracle Application Express (APEX) is a low-code development platform that enables you to build scalable, secure enterprise apps, with world-class features, that can be deployed anywhere. Using APEX, developers can quickly develop and deploy compelling apps that solve real problems and provide immediate value. You won't need to be an expert in a vast array of technologies to deliver sophisticated solutions. Focus on solving the problem and let APEX take care of the rest. Oracle APEX minimizes the complexity involved with multi-faceted applications and provides developers with the features they need to solve business problems without needing to become experts in vast web technologies. Explore how Oracle APEX helps you build better apps by taking care of these six facets of application development:

(i)Data (ii)User Interface (iii) Security (iv)Accessibility (iv)Monitoring (v)Globalization

## 4.2 ORACLE LIVE

Oracle recently introduced a new generation online SQL tool, called Live SQL. Live SQL is a kind of search engine for database developers but also a great free learning tool for those of you who want to learn and code SQL on an Oracle Database.

# CHAPTER 5 : TESTING

## 5.1 : UNIT TESTING

### 5.1.1 Introduction

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. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

### 5.1.2 Benefits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

1) **Find problems early** : Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

2) **Facilitates Change** : Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

3) **Simplifies Integration** : Unit testing may reduce uncertainty in the units themselves and can be used in a

bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier,

4) **Documentation** : Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviours that are to be trapped by the unit.

## 5.2 : 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.

### 5.2.1 Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using

black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatic complexity of the software and software architecture, reusability of modules and life-cycle and versioning management.Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high frequency integration.

#### 5.2.1.1 Big Bang

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration

testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment.

#### 5.2.1.2 Top-down And Bottom-up

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also

helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage. Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module. Sandwich testing is an approach to combine top down testing with bottom up testing.

## 5.3 : SOFTWARE VERIFICATION AND VALIDATION

### 5.3.1 Introduction

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfils its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review. Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

- Validation : Are we building the right product?
- Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that “you built it right”. Software validation ensures that “you built the right thing”.

### 5.3.2 Classification of Methods

In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-mission critical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are used.

### 5.3.3 Test Cases

A test case is a tool used in the process. Test cases may be prepared for software verification and validation.

# CHAPTER 6: RESULTS

## 6.1 : DATABASE

6.1.1 ADMIN

```
CREATE TABLE admin (
    A_id varchar(50) NOT NULL,
    username varchar(50) NOT NULL,
    A_password varchar(50) NOT NULL,
    PRIMARY KEY (A_id)
    )
```

| A_ID | USERNAME | A_PASSWORD |
|---|---|---|
| 1 | adminGUNEET | admin&^123 |
| 2 | adminMAYANK | admin#$123 |
| 3 | adminJASHAN | admin9023&123 |
| 4 | adminPriyankaMam | admin&123123 |

6.1.2 BLOOD

```
CREATE TABLE blood (
    username varchar(50) NOT NULL,
    B_password varchar(50) NOT NULL
)
```

| USERNAME | B_PASSWORD |
|---|---|
| chandler | 1234 |
| pheobe | 123456 |
| monica | 123 |

6.1.3 CAMPS

```
CREATE TABLE camps (
  c_name varchar(50) NOT NULL,
  venue varchar(50) NOT NULL,
  c_date date NOT NULL,
  c_time varchar(50) NOT NULL
)
```

| C_NAME | VENUE | C_DATE | C_TIME |
|---|---|---|---|
| Ludhiana | BRS Nagar | 13-FEB-21 | 18:00 |
| Ludhiana | Sarabha Nagar | 10-AUG-20 | 12:40 |
| Ludhiana | DUGRI | 10-MAY-21 | 18:00 |

6.1.4 DETAILS

```
CREATE TABLE details (
  sno VARCHAR(20) NOT NULL,
  city varchar(40) NOT NULL,
  area varchar(40) NOT NULL,
  D_date DATE NOT NULL,
  D_time varchar(23) NOT NULL,
  D_name varchar(20) NOT NULL,
  PRIMARY KEY (sno) )
```

| SNO | CITY | AREA | D_DATE | D_TIME | D_NAME |
|---|---|---|---|---|---|
| 1 | ludhiana | model town | 23-OCT-20 | 09:40:00 | Guneet |
| 2 | patiala | leela bhavan | 17-OCT-20 | 09:30:00 | Jashanpreet |
| 3 | chandigarh | sector 17 | 02-MAY-21 | 07:30:00 | Mayank |
| 4 | DNN | Model Town | 02-MAY-21 | 09:30:00 | Devansh |
| 5 | Jalandhar | Urban Estate | 02-MAY-21 | 09:30:00 | Abhishek |
| 6 | Ludhiana | Model Town | 02-JUL-20 | 11:30:00 | Palak |

6.1.5 DONATE

```
CREATE TABLE donate (
  email varchar(1000) NOT NULL,
```

```
  username varchar(100) NOT NULL,
  age VARCHAR(100) NOT NULL,
  gender varchar(1000) NOT NULL,
  weight varchar(100) NOT NULL,
  lastdonated varchar(100) NOT NULL,
  dateOfsubmission date NOT NULL
)
```

| EMAIL | USERNAME | AGE | GENDER | WEIGHT | LASTDONATED | DATEOFSUBMISSION |
|---|---|---|---|---|---|---|
| guneetkohli@gmail.com | guneet | 20 | Female | 42 | 02-FEB-16 | 17-JAN-20 |
| jashan@gmail.com | jashan | 21 | Female | 52 | 02-MAY-18 | 12-MAR-20 |
| mayankthakur@gmail.com | mayank | 21 | Male | 60 | 22-JUN-19 | 28-OCT-20 |
| palak@gmail.com | palak | 21 | Female | 49 | 22-OCT-17 | 28-MAR-21 |
| devansh12@gmail.com | devansh | 21 | Male | 65 | 19-JUN-09 | 13-MAY-21 |

6.1.6 USER

```
CREATE TABLE userr(
  user_id varchar(100) NOT NULL,
  username varchar(100) NOT NULL,
  user_password varchar(100) NOT NULL,
  blood_group varchar(100) NOT NULL,
  city varchar(100) NOT NULL,
  phone varchar(100) NOT NULL,
```

```
  PRIMARY KEY (user_id)
)
```

| USER_ID | USERNAME | USER_PASSWORD | BLOOD_GROUP | CITY | PHONE |
|---|---|---|---|---|---|
| 1 | Guneet | 7123 | A+ | ludhiana | 9435462121 |
| 2 | sanchit | 17234 | B+ | khanna | 2147483647 |
| 3 | Jashan | 71234 | B+ | ludhiana | 2147483647 |
| 4 | Sonali | 123784 | A- | patiala | 9587985 |
| 5 | Mayank | 459 | A+ | khanaa | 2147483647 |
| 6 | Palak | 5374 | O+ | ludhiana | 894589358 |
| 7 | Saloni | 63374 | O+ | ludhiana | 2147483647 |
| 8 | Sunaina | 72334 | O+ | ludhiana | 2147483647 |
| 9 | Rupali | 8253 | A+ | ludhiana | 2147483647 |
| 10 | Kartik | 7253 | AB+ | ludhiana | 2147483647 |
| 11 | Abhishek | 92345 | O+ | ludhiana | 2147483647 |

6.1.7 USERS

```
CREATE TABLE user_s(
  u_id varchar(20) NOT NULL,
  username varchar(20) NOT NULL,
  u_password varchar(20) NOT NULL,
  PRIMARY KEY (u_id)
)
```

| U_ID | USERNAME | U_PASSWORD |
|---|---|---|
| 1 | guneet | 1713 |
| 2 | jashan | 1903 |
| 3 | mayank | 1378 |

6.1.8 PATIENT

```
create table Patient(
    patient_id integer,
    patient_name varchar(20) not null,
    blood_group varchar(4),
    disease varchar(20),
    patient_address varchar(20),
    patient_contact_no varchar(20),
    primary  key (patient_id)
)
```

| PATIENT_ID | PATIENT_NAME | BLOOD_GROUP | DISEASE | PATIENT_ADDRESS | PATIENT_CONTACT_NO |
|---|---|---|---|---|---|
| 1001 | PA | A+ | N/A | Dhaka | 01234 |
| 1002 | PB | B+ | N/A | Khulna | 04234 |
| 1003 | PC | A+ | N/A | Khulna | 01634 |
| 1004 | PD | B+ | N/A | Dhaka | 01294 |
| 1005 | PE | A+ | N/A | Dhaka | 01434 |

Download CSV

5 rows selected.

6.1.9 DONOR

```
create table Donar(
    donar_id integer,
    donar_name varchar(20) not null,
    blood_group varchar(4),
    donar_contact_no varchar(20),
    donar_address varchar(20),
    disease varchar(20),
    primary key (donar_id)
);
```

| DONAR_ID | DONAR_NAME | BLOOD_GROUP | DONAR_CONTACT_NO | DONAR_ADDRESS | DISEASE |
|---|---|---|---|---|---|
| 101 | DA | A+ | 0134 | Dhaka | - |
| 102 | DB | A+ | 0134 | Dhaka | - |
| 103 | DC | B+ | 0134 | Khulna | - |
| 104 | DD | A+ | 0134 | Khulna | - |
| 105 | DE | B+ | 0134 | Dhaka | - |
| 106 | DF | A+ | 0134 | Chittagong | - |
| 107 | DG | A+ | 0134 | Dhaka | - |
| 108 | DH | B+ | 0134 | Khulna | - |

Download CSV

8 rows selected.

6.1.10 DONOR

```
create table BloodBank(
    blood_bank_id integer,
    blood_bank_name varchar(20),
    blood_bank_address varchar(20),
    blood_bank_contact_no varchar(20),
    primary key (blood_bank_id)
);
```

| BLOOD_BANK_ID | BLOOD_BANK_NAME | BLOOD_BANK_ADDRESS | BLOOD_BANK_CONTACT_NO |
|---|---|---|---|
| 1 | BBA | Dhaka | 911 |
| 2 | BBB | Khulna | 912 |
| 3 | BBC | Dhaka | 913 |
| 4 | BBD | Khulna | 914 |
| 5 | BBE | Dhaka | 915 |
| 6 | BBF | Khulna | 916 |
| 7 | BBG | Dhaka | 917 |
| 8 | BBH | Khulna | 918 |
| 9 | BBI | Dhaka | 919 |
| 10 | BBJ | Khulna | 920 |

### 6.1.10 DONATES

```
create table Donates(
    donate_id integer,
    donar_id integer,
    date_of_donation date,
    blood_bank_id integer,
    flag integer default 1,
    primary key(donate_id),
    foreign key (donar_id) references Donar(donar_id),
    foreign key (blood_bank_id) references BloodBank(blood_bank_id)
);
```

| DONATE_ID | DONAR_ID | DATE_OF_DONATION | BLOOD_BANK_ID | FLAG |
|---|---|---|---|---|
| 10001 | 101 | 28-FEB-18 | 1 | 1 |
| 10002 | 104 | 27-FEB-18 | 2 | 1 |
| 10003 | 102 | 26-FEB-18 | 1 | 1 |
| 10004 | 106 | 25-APR-18 | 2 | 1 |
| 10005 | 105 | 24-FEB-18 | 1 | 1 |
| 10006 | 106 | 28-FEB-18 | 1 | 1 |
| 10007 | 107 | 27-FEB-18 | 2 | 1 |

# 6.2: CURSOR TO FETCH AVAILABLE DONORS IN A CITY

```
DECLARE
c_id details.sno%type;
c_city details.city%type;
c_name details.d_name%type;
cursor c_details is
select sno,city ,d_name from details;
BEGIN
open c_details;
loop
fetch c_details into c_id,c_city ,c_name;
c_city:='ludhiana';
dbms_output.put_line(c_id||' '||c_city||' '||c_name);
exit;
end loop;
close c_details;
end;
/
```

| SNO | CITY | AREA | D_DATE | D_TIME | D_NAME |
|---|---|---|---|---|---|
| 1 | ludhiana | model town | 23-OCT-20 | 09:40:00 | Guneet |
| 2 | patiala | leela bhavan | 17-OCT-20 | 09:30:00 | Jashanpreet |
| 3 | chandigarh | sector 17 | 02-MAY-21 | 07:30:00 | Mayank |
| 4 | DNN | Model Town | 02-MAY-21 | 09:30:00 | Devansh |
| 5 | Jalandhar | Urban Estate | 02-MAY-21 | 09:30:00 | Abhishek |

Download CSV
5 rows selected.

Statement processed.
1 ludhiana Guneet

## 6.3: FUNCTION TO COUNT TOTAL DONORS

```
CREATE OR REPLACE FUNCTION totalDonar
RETURN number IS
  total integer := 0;
BEGIN
  SELECT count(*) into total
  FROM Donar;

  RETURN total;
END;
```

## 6.4: TRIGGERS

### 6.4.1 Donor Disease Checker

(If the disease is fatal, donor's information won't be inserted)

```
create or replace trigger desease_trg
before insert on Donar
for each row
```

```
when(NEW.donar_id>0)
begin
   if :NEW.disease = 'HepB' then
      dbms_output.put_line('donar with fatal disease!');
      delete from donar where donar_id = :NEW.donar_id;
   end if;
end;
```

```
Trigger created.
```

```
1 row(s) inserted.
donar with fatal disease!
```

**6.4.2 Donor Location Change**

(If there is an update in donor's location)

```
create or replace trigger loc_change
before update on Donar
for each row
when(NEW.donar_address!=OLD.donar_address)
declare

begin
   dbms_output.put_line('Old location : '|| :OLD.donar_address);
   dbms_output.put_line('New location : '|| :NEW.donar_address);
End;
```

```
Trigger created.
```

update Donar set donar_address = 'LUDHIANA' where donar_id = 101 ;
insert into Donar values(111,'DX','B+','0134','Dhaka','HepB');

```
1 row(s) updated.
Old location : Dhaka
New location : LUDHIANA
```

## 6.5: PROCEDURE

```
PROCEDURE no_of_donar_available(p_blood IN
Patient.blood_group%type,numm OUT integer) IS
BEGIN
  numm := 0;
  cnt := 101;
  while cnt<=110
  loop
   select blood_group into d_blood
   from Donar where donar_id = cnt;

      if (d_blood = p_blood) then
         numm := numm + 1;
      end if;
   cnt:=cnt+1;
  end loop;
END;
```

# CHAPTER 7 : ADVANTAGES

- Helps Blood Banks to automate blood donors and depositories online.It is a faster way to keep records, hence it saves time and is more efficient.
- Encourages blood donors to donate.
- Helps people find blood donors in times of need.
- Donors can view the blood donation camp organised at the different places.
- Store, process, retrieve and analyse information concerned with the administrative and inventory management within a blood bank.
- All the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way would be available here.
- It is a faster way to keep records of customers and staff, hence it saves time and is more efficient.
- Security of the patient and donor is of the utmost importance. By keeping track of the login activity through the application, any unwanted personnel in the camp can be found in time.
- Donor’s data is checked, if any deadly disease is found in the database, then details of the donor are not shared with the patient as the safety and well being of both the parties is required.

# CHAPTER 8 : CONCLUSION

In the world of information technology, the whole world is becoming a global village, where end users can get the information just sitting at home with one click. In fact, the government has taken a step in order to transform the system. Management information systems help to make the system paper less. Now the end user has to enroll himself and his job is done. All the money transactions are made possible because of the management information system. Researchers believe that by developing the management information system for the blood bank make revolutionary changes in the system. It is a small contribution of the researcher in order to serve mankind.

In this project we developed a complete back end software in which one can search for blood donation camps nearby.
From this application we can get an update on whether a particular blood group is available or not and count of donors as well, as it'll be easy for the patient to search online thus saving a lot of time.

In addition to this it can also help one in knowing if the donor is having any deadly disease, if so the database will generate a warning and the details of the donor won't be shared with the patient.
From this program we can also keep a track of various blood donors in a particular city, it also takes the last donation date so that the donor doesn't get any weakness. Another feature is that of change of location, if the donor moves to some other place, a feature is provided which will change the address of donor.

# REFERENCES

[1]Kumar, R., Singh, S. and Ragavi, V.A. (2017). ). Blood Bank Management System. Retrieved from http://ijariie.com/AdminUploadPdf/Blood_Bank_Management_System_ijariie6874.pdf2. Liyana, F.(2017).

[2] Blood Bank Management System Using Rule-Based Method. Retrievedfrom http://greenskill.net/suhailan/fyp/report/038077.pdf

[3]. Ministry of health. Article source: https://www.moh.gov.om/

[4] Teena, C.A, Sankar, K. and Kannan, S. (2014). A Study on Blood Bank Management.Retrieved from https://www.idosi.org/mejsr/mejsr19(8)14/21.pdf12

[5] Blood Bank Management System. Retrieved from https://www.tribuneindia.com/news/archive/bathinda/now-online-blood-bank-system-to-benefit-public-21283