A project report submitted in partial fulfilment for the Award

of

BACHELOR OF TECHNOLOGY

IN

Computer Science & Engineering

By

PREETI SINGH(12115175)

To

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB,

INDIA

BLOOD BANK ADMINISTRATION MANAGEMENT SYSTEM

ABSTRACT

This project aims to develop a Blood Bank Management System. A Blood Bank Management System can be used in any clinic, hospital, labs or any emergency situation which requires

blood units for survival. Our system can be used to find required type of blood in emergency situations from either blood bank or even blood donors.

Current system uses a grapevine communication for finding blood in cases of emergency, may it be by a donor or blood bank. The intentions of proposing such a system is to abolish the panic caused during an emergency due to unavailability of blood.

INTRODUCTION

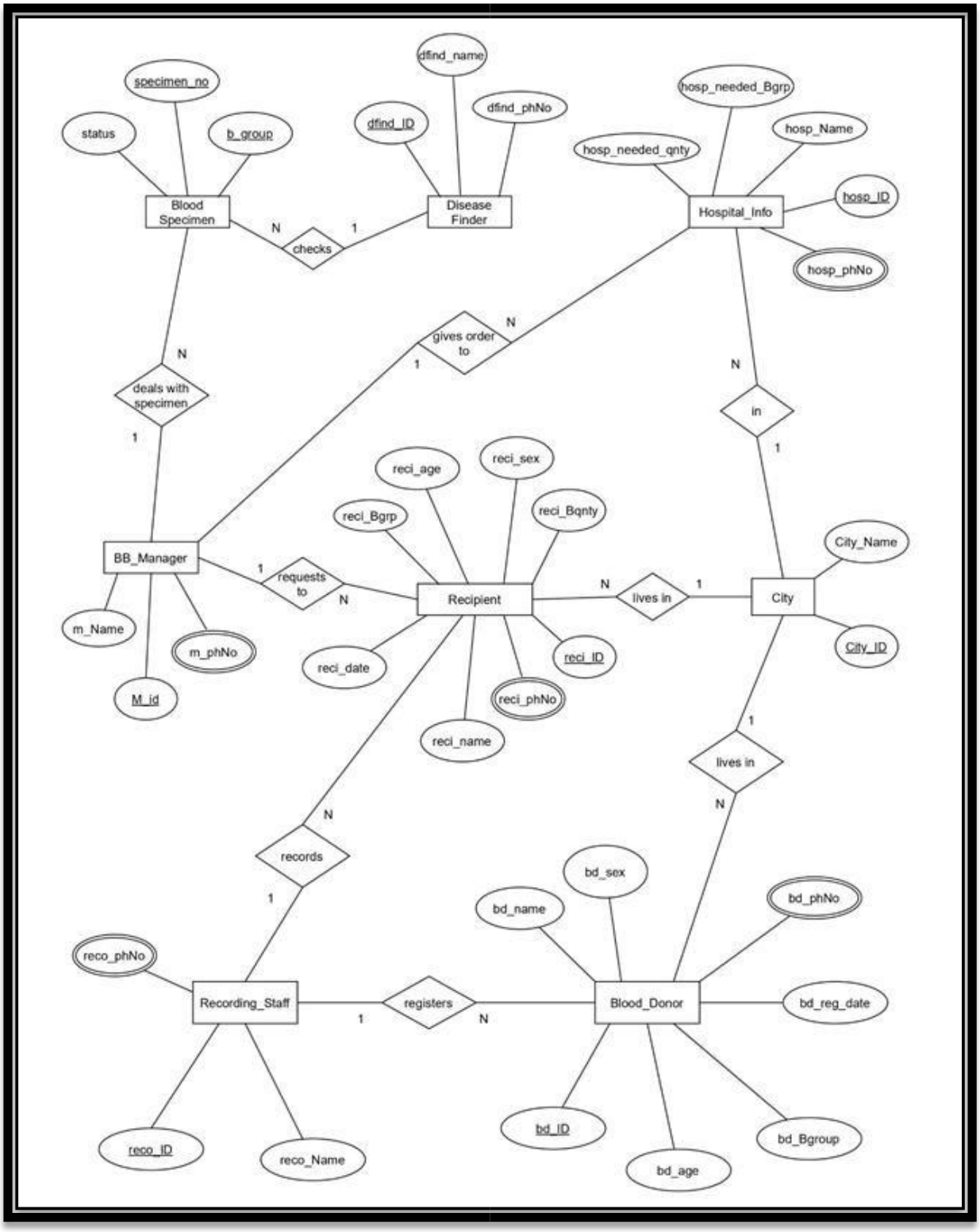
Blood banks collect, store and provide collected blood to the patients who are in need of blood. The people who donate blood are called 'donors'. The banks then group the blood which they receive according to the blood groups. They also make sure that the blood is not contaminated. The main mission of the blood bank is to provide the blood to the hospitals and health care systems which saves the patient's life. No hospital can maintain the health care system without pure and adequate blood.

The major concern each blood bank has is to monitor the quality of the blood and monitor the people who donates the blood, that is 'donors'. But this a tough job. The existing system will not satisfy the need of maintaining quality blood and keep track of donors. To overcome all these limitations we introduced a new system called 'Blood Donation Management System'.

The 'Blood Bank Management System' allows us to keep track of quality of blood and also keeps track of available blood when requested by the acceptor. The existing systems are Manual systems which are time consuming and not so effective. 'Blood Bank Management system' automates the distribution of blood. This database consists of thousands of records of each blood bank.

By using this system searching the available blood becomes easy and saves lot of time than the manual system. It will hoard, operate, recover and analyse information concerned with the administrative and inventory management within a blood bank. This system is developed in a manner that it is manageable, time effective, cost effective, flexible and much man power is not required.

ER DIAGRAM



INFORMATION OF ENTITIES

In total we have eight entities and information of each entity is mentioned below:-

1. **Blood_Donor:**
   (Attributes – bd_ID, bd_name, bd_sex, bd_age, bd_Bgroup, bd_reg_date, bd_phNo)

The donor is the person who donates blood, on donation a donor id (bd_ID) is generated and used as primary key to identify the donor information. Other than that name, age , sex , blood group, phone number and registration dates will be stored in database under Blood_Donor entity.

2. **Recipient:**
   (Attributes – reci_ID, reci_name, reci_age, reci_Bgrp, reci_Bqnty , reci_sex, reci_reg_date, reci_phNo)

The Recipient is the person who receives blood from blood bank, when blood is given to a recipient a recipient ID (reci_ID) is generated and used as primary key for the recipient entity to identify blood recipients information. Along with it name ,age, sex, blood group (needed), blood quantity(needed) , phone number, and registration dates are also stored in the data base under recipient entity.

3. **BB_Manager:**
   (Attributes – m_ID, m_Name, m_phNo)

The blood bank manager is the person who takes care of the available blood samples in the blood bank, he is also responsible for handling blood requests from recipients and hospitals. Blood manager has a unique identification number (m_ID) used as primary key along with name and phone number of blood bank manager will be stored in data base under BB_Manager entity.

4. **Recording_Staff :**
   (Attributes – reco_ID, reco_Name, reco_phNo)

The recording staff is a person who registers the blood donor and recipients and the Recording_Staff enitity has reco_ID which is primary key along with recorder's name and recorder's phone number will also be stored in the data base under Recording_Staff entity.

5. **BloodSpecimen :**
   (Attributes – specimen_number, b_group , status)

In data base, under Blood Specimen entity we will store the information of blood samples which are available in the blood bank. In this entity specimen_number and b_group together will be primary key along with status attribute which will show if the blood is contaminated on not.

6. **DiseaseFinder :**
   (Attributes - dfind_ID, dfind_name, dfind_PhNo)

In data base , under DiseaseFinder entity we will store the information of the doctor who checks the blood for any kind of contaminations. To store that information we have unique identification number (dfind_ID) as primary key. Along with name and phone number of the doctor will also be stored under same entity.

7. **Hospital_Info :**
   (Attributes – hosp_ID, hosp_name, hosp_needed_Bgrp, hosp_needed_Bqnty)

In the data base, under Hospital_Info entity we will store the information of hospitals. In this hosp_ID and hosp_needed_Bgrp together makes the primary key. We will store hospital name and the blood quantity required at the hospital.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when, 1.
It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

Normalization of Blood Bank Database:

1. Blood_Donor *(bd_Id, bd_name, bd_phNo bd_sex, bd_age, bd_reg_date, bd_Bgroup, reco_ID, City_ID)*

{bd_Id} = > {bd_name} (functional dependency exists, because two different bd_name do not correspond to the same bd_Id).
{bd_ID} = > {bd_sex} (functional dependency exists).
{bd_ID} = > {bd_age} (functional dependency exists).
{bd_ID} = > {bd_reg_date} date (functional dependency exists).
{bd_ID} = > {reco_id} (functional dependency exists).
{bd_ID} = > {city_id} (functional dependency exists).
{bd_ID} = > {bd_Bgroup} (functional dependency exists).

As the attributes of this table does not have sub attributes, it is in first normal form. Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form. Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

2. City *(city_id , city_name)*

{city_id}= > {city_name}

The table is in first normal form.
The table is in second normal form.
The table is in third normal form.

The table is in first normal form.
The table is in second normal form.
The table is in third normal form.

6. Disease_finder *(dfind_id, dfind_name, dfind_PhNo)*

{ dfind_id } = > { dfind_name }
{ dfind_id } = > { dfind_PhNo } (functional dependency exists).

The table is in first normal form.
The table is in second normal form.
The table is in third normal form.

7. manager *(M_id, m_name, m_phNo)*

{M_id} = >{m_name}
{M_id} = > {m_phNo} (functional dependency exists)

The table is in first normal form.
The table is in second normal form.
The table is in third normal form.

8. Hospital_Info *(hosp_Id, hosp_Name, hosp_phNo, hosp_needed_Bgrp, hosp_needed_qty, city_id, m_id)*

{hosp_Id}= > {hosp_Name, hosp_phNo, city_id, m_id}
{hosp_Id, hosp_needed_Bgrp } = > hosp_needed_qty (functional dependency exists)

The table is in first normal form.
Since every non-primary key attribute is not fully functionally dependent on the primary key of the table, this table is not in second normal form. Hence we have to split the table.

Hospital_1 (hosp_Id, hosp_phNo, hosp_Name, city_id, m_id).

ADVANTAGES OF BLOOD BANK MANAGEMENT SYSTEM:

A unique blood bank management system with multiple benefits.A unique blood bank management system with multiple benef

Ideal Blood Bank Inventory Management Solution.

Improved patient safety through human error reduction.

Simultaneous track & trace of multiple blood bags.

Instant localization & validation with LED tags.

Reconciliation of data with physical reality.

DISADVANTAGES:

Disadvantages : Manual document and data entry.

Only web based system is available no mobile based system avai

Data base Handling.

File Handling.

Server interactions.

Security Encryption / Decryption.

Data protection on Device.

SQL IMPLEMENTATION

The implementation on SQL Server is given below :-

1:

```
CREATE TABLE Manager
( M_id int NOT NULL PRIMARY KEY,
mName varchar(100) NOT NULL,
m_phNo int
);
INSERT into Manager
VALUES(101,'shivank', 9693959671)

INSERT into Manager
VALUES(102,'shwetanshu', 9693959672)

INSERT into Manager
VALUES(103,'singh', 9693959673)

INSERT into Manager
VALUES(104,'yusuf', 9693959674)

INSERT into Manager
VALUES(105,'jackson', 9693959675)
INSERT into Manager
VALUES(106,'akhil', 9693959676)

INSERT into Manager
VALUES(107,'jojo', 9693959677)
```

```
INSERT into Manager

VALUES(108,'stella', 9693959678)


INSERT into Manager

VALUES(109,'monika', 9693959679)


INSERT into Manager

VALUES(110,'himanshi', 9693959680);

select * from Manager;

----------------------
 2: CREATE TABLE Blood_Donor

      ( bd_ID int NOT NULL PRIMARY KEY,

      bd_name varchar(100) NOT NULL,

      bd_age varchar(100),

      bd_sex varchar(100),

      bd_Bgroup varchar(10),

      bd_reg_date date,

      reco_ID int NOT NULL,

      City_ID int NOT NULL,

      FOREIGN KEY(reco_ID) REFERENCES Recording_Staff(reco_ID),

      FOREIGN KEY(City_ID) REFERENCES City(City_ID)


      INSERT into Blood_Donor

      VALUES(150011,'Mark',25,'M','O+','2015-07-19',101412,1100)

      INSERT into Blood_Donor

      VALUES(150012,'Abdul',35,'M','A-','2015-12-24',101412,1100)

      INSERT into Blood_Donor

      VALUES(150013,'Shivank',22,'M','AB+','2015-08-28',101212,1200)

      INSERT into Blood_Donor
```

```
VALUES(150014,'shweta',29,'M','B+','2015-12-17',101212,1300)

INSERT into Blood_Donor

VALUES(150015,'Shyam',42,'M','A+','2016-11-22',101212,1300)

INSERT into Blood_Donor

VALUES(150016,'Daniel',44,'F','AB-','2016-02-06',101212,1200)

INSERT into Blood_Donor

VALUES(150017,'Mike tyson',33,'M','B-','2016-10-15',101312,1400)

INSERT into Blood_Donor

VALUES(150018,'Elisa',31,'F','O+','2016-01-04',101312,1200)

INSERT into Blood_Donor

VALUES(150020,'shivansh',29,'M','O-','2016-12-17',101212,1200);

select * from Blood_Donor;

----------
```

3:
```
CREATE TABLE BloodSpecimen

( specimen_number int NOT NULL,

b_group varchar(10) NOT NULL,

status int,

dfind_ID int NOT NULL,

M_id int NOT NULL,

primary key (specimen_number,b_group),

FOREIGN KEY(M_id) REFERENCES Manager(M_id),

FOREIGN KEY(dfind_ID) REFERENCES DiseaseFinder(dfind_ID)

);

INSERT into BloodSpecimen

VALUES(1001, 'B+', 1,11,101)

INSERT into BloodSpecimen

VALUES(1002, 'O+', 1,12,102)
```

```
INSERT into BloodSpecimen
VALUES(1003, 'AB+', 1,11,102)
INSERT into BloodSpecimen
VALUES(1004, 'O-', 1,13,103)
INSERT into BloodSpecimen
VALUES(1005, 'A+', 0,14,101)
INSERT into BloodSpecimen
VALUES(1006, 'A-', 1,13,104)
INSERT into BloodSpecimen
VALUES(1007, 'AB-', 1,15,104)
INSERT into BloodSpecimen
VALUES(1008, 'AB-', 0,11,105)
INSERT into BloodSpecimen
VALUES(1009, 'B+', 1,13,105)
INSERT into BloodSpecimen
VALUES(1010, 'O+', 0,12,105)
INSERT into BloodSpecimen
VALUES(1011, 'O+', 1,13,103)
INSERT into BloodSpecimen
VALUES(1012, 'O-', 1,14,102)
INSERT into BloodSpecimen
VALUES(1013, 'B-', 1,14,102)
INSERT into BloodSpecimen
VALUES(1014, 'AB+', 0,15,101);
Select * from BloodSpecimen;
---------------------------------
```

4: CREATE TABLE City

```
( City_ID int NOT NULL PRIMARY KEY,

City_name varchar(100) NOT NULL

);

INSERT into City

VALUES(1100,'Dallas')

INSERT into City

VALUES(1200,'Austin')

INSERT into City

VALUES(1300,'Irving')

INSERT into City

VALUES(1400,'Houston')

INSERT into City

VALUES(1500,'Richardson')

INSERT into City

VALUES(1600,'Plano')

INSERT into City

VALUES(1700,'Frisco')

INSERT into City

VALUES(1800,'Arlington')

INSERT into City

VALUES(1900,'San Antonio')

INSERT into City

VALUES(2000,'Tyler');

select * from City;
```

5 :

```
CREATE TABLE DiseaseFinder

( dfind_ID int NOT NULL PRIMARY KEY,

dfind_name varchar(100) NOT NULL,

dfind_PhNo bigint

);
```

```
INSERT into DiseaseFinder
VALUES(11,'Peter',9693959681)
INSERT into DiseaseFinder
VALUES(12,'Park',9693959682)
INSERT into DiseaseFinder
VALUES(13,'Jerry',9693959683)
INSERT into DiseaseFinder
VALUES(14,'shivam',9693959672)
INSERT into DiseaseFinder
VALUES(15,'Monika',9693959679)
INSERT into DiseaseFinder
VALUES(16,'Ram',9693959684)
INSERT into DiseaseFinder
VALUES(17,'Swathi',9693959685)
INSERT into DiseaseFinder
VALUES(18,'Gautham',9693959686)
INSERT into DiseaseFinder
VALUES(19,'Ashwin',9693959687)
INSERT into DiseaseFinder
VALUES(20,'Yash',9693959688);
select * from DiseaseFinder;
```

6:

```
CREATE TABLE Hospital_Info_1
( hosp_ID int NOT NULL,
hosp_name varchar(100) NOT NULL,
City_ID int NOT NULL,
M_id int NOT NULL,
primary key(hosp_ID),
FOREIGN KEY(M_id) REFERENCES Manager(M_id),
FOREIGN KEY(City_ID) REFERENCES City(City_ID)
);
INSERT into Hospital_Info_1
```

```
VALUES(1,'MayoClinic',1100,101),

INSERT into Hospital_Info_1

VALUES(2,'CleavelandClinic',1200,103),

INSERT into Hospital_Info_1

VALUES(3,'NYU',1300,103);

INSERT into Hospital_Info_1

VALUES(4,'Baylor',1400,104),

INSERT into Hospital_Info_1

VALUES(5,'Charlton',1800,103),

INSERT into Hospital_Info_1

VALUES(6,'Greenoaks',1300,106),

INSERT into Hospital_Info_1

VALUES(7,'Forestpark',1300,102),

INSERT into Hospital_Info_1

VALUES(8,'Parkland',1200,106),

INSERT into Hospital_Info_1

VALUES(9,'Pinecreek',1500,109),

INSERT into Hospital_Info_1

VALUES(10,'WalnutHill',1700,105);

select * from Hospital_Info_1;

--------------------
```

7 :

```
CREATE TABLE Hospital_Info_2

( hosp_ID int NOT NULL,

hosp_name varchar(100) NOT NULL,

hosp_needed_Bgrp varchar(10),

hosp_needed_qnty int,

primary key(hosp_ID,hosp_needed_Bgrp)

);

INSERT into Hospital_Info_2

VALUES(1,'MayoClinic','A+',20)
```

```
INSERT into Hospital_Info_2

VALUES(1,'MayoClinic','A-',0)

INSERT into Hospital_Info_2

VALUES(1,'MayoClinic','AB+',40)

INSERT into Hospital_Info_2

VALUES(1,'MayoClinic','AB-',10)

INSERT into Hospital_Info_2

VALUES(1,'MayoClinic','B-',20)

INSERT into Hospital_Info_2

VALUES(2,'CleavelandClinic','A+',40)

INSERT into Hospital_Info_2

VALUES(2,'CleavelandClinic','AB+',20)

INSERT into Hospital_Info_2

VALUES(2,'CleavelandClinic','A-',10)

INSERT into Hospital_Info_2

VALUES(2,'CleavelandClinic','B-',30)

INSERT into Hospital_Info_2

VALUES(2,'CleavelandClinic','B+',0)

INSERT into Hospital_Info_2

VALUES(2,'CleavelandClinic','AB-',10)

INSERT into Hospital_Info_2

VALUES(3,'NYU','A+',0)

INSERT into Hospital_Info_2

VALUES(3,'NYU','AB+',0)

INSERT into Hospital_Info_2

VALUES(3,'NYU','A-',0)

INSERT into Hospital_Info_2

VALUES(3,'NYU','B-',20)

INSERT into Hospital_Info_2

VALUES(3,'NYU','B+',10)

INSERT into Hospital_Info_2
```

```
VALUES(3,'NYU','AB-',0)

INSERT into Hospital_Info_2

VALUES(4,'Baylor','A+',10)

INSERT into Hospital_Info_2

VALUES(4,'Baylor','A-',40)

INSERT into Hospital_Info_2

VALUES(7,'Forestpark','B-',40)

INSERT into Hospital_Info_2

VALUES(8,'Parkland','B+',10)

INSERT into Hospital_Info_2

VALUES(9,'Pinecreek','AB-',20);

select * from Hospital_Info_2;


----------------------------------------
8 :     CREATE TABLE Recipient

        ( reci_ID int NOT NULL PRIMARY kEY,

        reci_name varchar(100) NOT NULL,

        reci_age varchar(10),

        reci_Brgp varchar(100),

        reci_Bqnty float,

        reco_ID int NOT NULL,

        City_ID int NOT NULL,

        M_id int NOT NULL,

        FOREIGN KEY(M_id) REFERENCES Manager(M_id),

        FOREIGN KEY(City_ID) REFERENCES City(City_ID)

        );

        Alter table Recipient

        ADD reci_sex varchar(100);

        Alter table Recipient

        ADD reci_reg_date date;

        INSERT into Recipient
```

```
VALUES(10001,'Peter',25,'B+',1.5,101212,1100,101,'M','2015-12-17')

INSERT into Recipient

VALUES(10002,'shivank',60,'A+',1,101312,1100,102,'M','2015-12-16')

INSERT into Recipient

VALUES(10003,'akhil',35,'AB+',0.5,101312,1200,102,'M','2015-10-17')

INSERT into Recipient

VALUES(10004,'Parker',66,'B+',1,101212,1300,104,'M','2016-11-17')

INSERT into Recipient

VALUES(10005,'jojo',53,'B-',1,101412,1400,105,'M','2015-04-17')

INSERT into Recipient

VALUES(10006,'Preetham',45,'O+',1.5,101512,1500,105,'M','2015-12-17')

INSERT into Recipient

VALUES(10007,'Swetha',22,'AB-',1,101212,1500,101,'F','2015-05-17')

INSERT into Recipient

VALUES(10008,'Swathi',25,'B+',2,101412,1300,103,'F','2015-12-14')

INSERT into Recipient

VALUES(10009,'Lance',30,'A+',1.5,101312,1100,104,'M','2015-02-16')

INSERT into Recipient

VALUES(10010,'Marsh',25,'AB+',3.5,101212,1200,107,'M','2016-10-17')

select * from Recipient;

----------
```

IMPLEMENTATION IMAGES :

M_id	mName	m_phNo
101	shivank	9693959671
102	shwetanshu	9693959672
103	singh	9693959673
104	yusuf	9693959674
105	jackson	9693959675
106	akhil	9693959676
107	jojo	9693959677
108	stella	9693959678
109	monika	9693959679
110	himanshi	9693959680

bd_ID	bd_name	bd_age	bd_sex	bd_Bgroup	bd_reg_date	reco_ID	City_ID
150011	Mark	25	M	O+	2015-07-19	101412	1100
150012	Abdul	35	M	A-	2015-12-24	101412	1100
150013	Shivank	22	M	AB+	2015-08-28	101212	1200
150014	shweta	29	M	B+	2015-12-17	101212	1300
150015	Shyam	42	M	A+	2016-11-22	101212	1300
150016	Dan	44	F	AB-	2016-02-06	101212	1200
150017	Mike	33	M	B-	2016-10-15	101312	1400
150018	Elisa	31	F	O+	2016-01-04	101312	1200
150019	Carrol	24	F	AB+	2016-09-10	101312	1500
150020	shivansh	29	M	O-	2016-12-17	101212	1200

specimen_number	b_group	status	dfind_ID	M_id
1001	B+	1	11	101
1002	O+	1	12	102
1003	AB+	1	11	102
1004	O-	1	13	103
1005	A+	0	14	101
1006	A-	1	13	104
1007	AB-	1	15	104
1008	AB-	0	11	105
1009	B+	1	13	105
1010	O+	0	12	105
1011	O+	1	13	103
1012	O-	1	14	102
1013	B-	1	14	102
1014	AB+	0	15	101

City_ID	City_name
1100	Dallas
1200	Austin
1300	Irving
1400	Houston
1500	Richardson
1600	Plano
1700	Frisco
1800	Arlington
1900	San Antonio
2000	Tyler

dfind_ID	dfind_name	dfind_PhNo
11	Peter	9693959681
12	Park	9693959682
13	Jerry	9693959683
14	shivam	9693959672
15	Monika	9693959679
16	Ram	9693959684
17	Swathi	9693959685
18	Gautham	9693959686
19	Ashwin	9693959687
20	Yash	9693959688

hosp_ID	hosp_name	City_ID	M_id
1	MayoClinic	1100	101
2	CleavelandClinic	1200	103
3	NYU	1300	103
4	Baylor	1400	104
5	Charlton	1800	103
6	Greenoaks	1300	106
7	Forestpark	1300	102
8	Parkland	1200	106
9	Pinecreek	1500	109
10	WalnutHill	1700	105

hosp_ID	hosp_name	hosp_needed_Bgrp	hosp_needed_qnty
1	MayoClinic	A+	20
1	MayoClinic	A-	0
1	MayoClinic	AB+	40
1	MayoClinic	AB-	10
1	MayoClinic	B-	20
2	CleavelandClinic	A+	40
2	CleavelandClinic	AB+	20
2	CleavelandClinic	A-	10
2	CleavelandClinic	B-	30
2	CleavelandClinic	B+	0
2	CleavelandClinic	AB-	10
3	NYU	A+	0
3	NYU	AB+	0
3	NYU	A-	0
3	NYU	B-	20
3	NYU	B+	10
3	NYU	AB-	0
4	Baylor	A+	10

reci_ID	reci_name	reci_age	reci_Brgp	reci_Bqnty	reco_ID	City_ID	M_id	reci_sex	reci_reg_date
10001	Peter	25	B+	1.5	101212	1100	101	M	2015-12-17
10002	shivank	60	A+	1	101312	1100	102	M	2015-12-16
10003	akhil	35	AB+	0.5	101312	1200	102	M	2015-10-17
10004	Parker	66	B+	1	101212	1300	104	M	2016-11-17
10005	jojo	53	B-	1	101412	1400	105	M	2015-04-17
10006	Preetham	45	O+	1.5	101512	1500	105	M	2015-12-17
10007	Swetha	22	AB-	1	101212	1500	101	F	2015-05-17
10008	Swathi	25	B+	2	101412	1300	103	F	2015-12-14
10009	Lance	30	A+	1.5	101312	1100	104	M	2015-02-16
10010	Marsh	25	AB+	3.5	101212	1200	107	M	2016-10-17

CONCLUSION

Prior to this project, a general study of blood bank management system was conducted from recent researches of various authors and facts were gathered in which helped to uncover the misfits that the system was facing.

After proper analysation of these problems, a solution was then developed in order to meet up the needs of a more advanced system. This system is known as the centralized blood bank repository which helped in eliminating all the problems that the previous systems were facing. With this system, Blood banks/

Centers, Hospitals, Patients and Blood donors will be brought together to enjoy a large number of functionalities and access a vast amount of information, thereby making blood donation and reception a lot easier and faster.

Before implementing the database, in the design phase, We have explored various features, operations of a blood bank to figure out required entities, attributes and the relationship among entities to make an efficient Entity Relationship Diagram(ERD). After analyzing all the requirements, I have created our ERD and then converted the ERD to relational model and normalized the tables.

Using SQL Server I have created the tables for my database and inserted some sample values in the tables. Finally, I have executed sample queries on the database to check its performance to retrieve useful information accurately and speedily.