



1

BLOOD BANK MANAGEMENT SYSTEM

OVERVIEW

Aim

The aim of Blood Bank management system is to create a bridge between the demand and supply of blood. To develop a **Blood Bank Management System** where authorized members can insert and retrieve the blood, plasma and stem cells data from the blood inventory. Record an activity when a donor wills to donate and grant to the recipients when needed.

Phases

- Conceptual Design
- Relational Schema
- Implementation
- Document & Demo

The right side of the slide contains two icons. The first is a graphic of three red blood drops with blood types A+, A-, AB-, B+, B-, and O- arranged around them. The second is a diagram of a database system architecture showing components like management software, database, system, DBMS, data, technology, and software.

2

CONCEPTUAL DESIGN

ASSUMPTIONS AND INSIGHTS

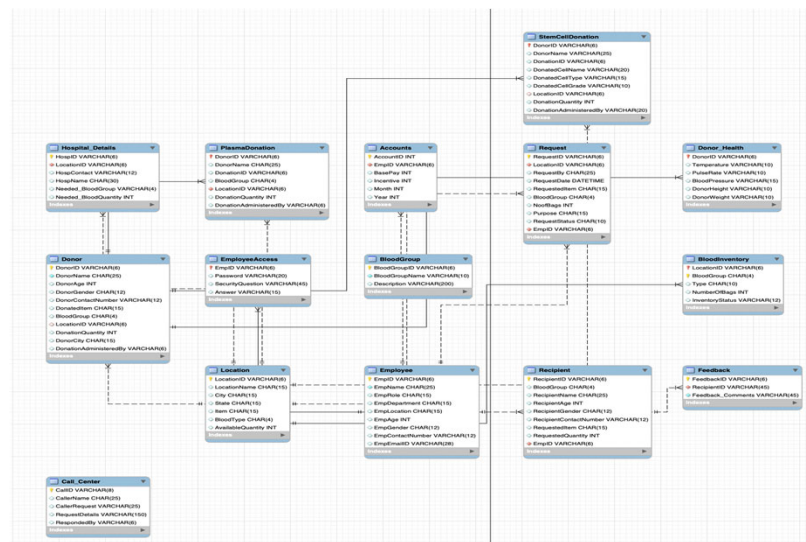
- Users on the website can be categorized into three, **DONORS** who donate blood, blood plasma and stem cells according to their possibility, **RECIPIENTS** are the patients or the customers who obtain the item according to their requirement, and **EMPLOYEES** who manages the donors and recipients.
- There are eight main blood types, but some are rarer than others. The list below shows the percentage of donors with each blood type: O positive: 35%, O negative: 13%, A positive: 30%, A negative: 8%, B positive: 8%, B negative: 2%, AB positive: 2%, AB negative: 1%
- Assigning a Blood Badge to a Donor according to the quantity of blood donated.
- Based on the filters like Age or Gender, Feedback is retrieved from Recipients.
- A duplicate record is not allowed for data redundancy.



3

RELATIONAL SCHEMA

EER DIAGRAM



4

IMPLEMENTATION

PROCESS OF BUILDING BLOOD BANK DATABASE

Implementation is done in MySQL using SQL statements in the following parts

1. Relational Schema and setting up the table relationships.
2. Creation of Database, Tables, Structures, Data Types and Format
3. Creation of complex queries to perform more accurate searches on a database.
4. Creation of triggers, stored procedures and functions corresponding to the distinct requirements



5

IMPLEMENTATION

COMPLEX QUERY

Tables used:
Feedback
Recipient

Fetches feedback from Male recipients.

```
Select FeedbackID, Feedback_Comments
From Feedback
Join Recipient On Feedback.RecipientID = Recipient.RecipientID
Where Recipient. RecipientGender = 'M';
```

FeedbackID	Feedback_Comments
202	Genuine Service
204	Sincere Response
206	Great experience
207	Personalized and specialized Care
208	Slow response



6

IMPLEMENTATION

STORED FUNCTION

Tables used:
Donor

Assign a BLOOD BADGE to a donor based on the quantity of blood donated overtime.

```
USE BloodBankManagementSystems;
DELIMITER $$
CREATE FUNCTION BloodBadge(
DonationQuantity INT
)
RETURNS VARCHAR(30)
DETERMINISTIC
BEGIN
DECLARE bloodBadge VARCHAR(30);
IF DonationQuantity = 1 THEN
SET bloodBadge = 'BRONZE';
ELSEIF (DonationQuantity >= 1 AND
DonationQuantity <= 2) THEN
SET bloodBadge = 'GOLD';
ELSEIF (DonationQuantity >= 3) THEN
SET bloodBadge = 'PLATINUM';
END IF;
-- return the customer level
RETURN (bloodBadge);
END$$
DELIMITER ;
SHOW FUNCTION STATUS
WHERE db = 'BloodBankManagementSystems';
-- SELECT * FROM Donor;
SELECT DonorName, BloodBadge(DonationQuantity) FROM Donor ORDER BY DonorName;
```

DonorName	BloodBadge(DonationQuant...
Chandler	BRONZE
Clint	GOLD
Davidson	BRONZE
Flue	PLATINUM
Joey	BRONZE
Kate	PLATINUM
Iouie	PLATINUM
Michael	BRONZE
Mimmi	PLATINUM
Momu	PLATINUM
Monica	BRONZE
Natasha	BRONZE



7

IMPLEMENTATION

STORED PROCEDURE

Tables used:
Donor

Fetches the Donor details according to the Donor ID.

```
DELIMITER //
CREATE PROCEDURE GetPatientByID(IN DonorID varchar(6))
BEGIN
select * from Donor
where DonorID = donor.DonorID;
END //
DELIMITER ;
use BloodBankManagementSystems;
call GetPatientByID(324351);
```

DonorID	DonorName	DonorAge	DonorGender	DonorContactNum...	DonatedItem	BloodGroup	LocationID	DonationQuant...	DonorCity	DonationAdministered...
324351	Davidson	35	M	8887770000	StemCell	B+	5Y3S	1	Michigan	Tom



8

IMPLEMENTATION

TRIGGER

Tables used:
EmployeeAccess

The trigger doesn't allow duplicate records.

```
use BloodBankManagementSystems;

SELECT FeedbackID, Feedback_Comments
FROM Feedback
JOIN Recipient ON Feedback.RecipientID = Recipient.RecipientID
WHERE Recipient.RecipientGender = 'M';

DELIMITER $$
CREATE TRIGGER insert_trigger_employee
BEFORE INSERT
ON Employee FOR EACH ROW
Begin
    if exists (select * from Employee where EmpID = new.EmpID) then
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Your record already exists';
    end if;
end $$
DELIMITER ;

INSERT INTO 'BloodBankManagementSystems'.Employee (EmpID, EmpName, EmpRole, EmpDepartment, EmpLocation, EmpAge, EmpGender, EmpContactNumber, EmpE
VALUES ('111111', 'Rim', 'Surgeon', 'Critical Care', 'Los Angeles', '34', 'M', '9873457892', 'rim123@gmail.com');
```

Error Code: 1644. Your record already exists



9

FUTURE SCOPE AND CONCLUSION

FUTURE SCOPE:

- The future implementation will be online help for the customers and chatting with administrator.
- Enhancing the project adding marketing leads and strategies.
- In our project we have created a stored function which determines the blood badge of a user. In future, we want to classify even the rare blood donors who doesn't satisfy the blood badge condition of 2 + units' donation as gold.
- Avoid accumulating blood from individuals who might also be unsuitable due to the threat of health factors that might harm the patients.

CONCLUSION:

The main purpose is to bring the donor and the patient in a common platform. This should encourage new donors and retain old donors to donate blood.



10

BLOOD DONOR. LIFE SAVER.

