Blood Bank Management System

Use Case Specification

Project Name: Blood Bank Management System

Project Description: The Blood Bank Management System (BBMS) aims to manage the donation, storage, and distribution of blood. The system will handle donor registrations, blood unit tracking, recipient registrations, transaction logs, and employee management.

Primary Actors:

- Donor
- Recipient
- Employee (Admin/Staff)

Use Cases:

1. **Register Donor**

    - Description: This use case allows a new donor to register. Both donors and employees can initiate this process. Upon completion, the donor's information, including personal details and blood type, is stored in the database.
    - Actor: Donor, Employee
    - Pre-condition: None
    - Post-condition: Donor information is stored in the database.

2. **Register Recipient**
    - Description: This use case allows a new recipient to register. Recipients or employees can initiate this process. Upon completion, the recipient's information, including personal details and required blood type, is stored in the database.

- Actor: Recipient, Employee
- Pre-condition: None
- Post-condition: Recipient information is stored in the database.

3. **Log Donation**

- Description: This use case allows employees to log a new blood donation. It requires the donor to be registered in the system. Once logged, the blood unit information is stored in the database, and the donor's last donation date is updated to the current date.
- Actor: Employee
- Pre-condition: Donor must be registered.
- Post-condition: Blood unit information is stored, and the donor's last donation date is updated.

4. **Log Transaction**

- Description: This use case allows employees to log a new blood transaction. It requires the recipient to be registered in the system and the blood unit to be available. Upon logging the transaction, the blood unit's status is updated to 'Used', and a record of the transaction is stored in the database.
- Actor: Employee
- Pre-condition: Recipient must be registered, and blood unit must be available.
- Post-condition: Blood unit status is updated to 'Used' and the transaction is logged.

5. **Manage Employees**

- Description: This use case allows the admin to add, update, or remove employee information. This process can only be initiated by an admin after logging in. Upon completion, the employee information is updated in the database, reflecting the changes made by the admin.
- Actor: Admin

- Pre-condition: Admin login
- Post-condition: Employee information is updated in the database.

2. Create DDL (Data Definition Language)

Create Donors Table

```
CREATE TABLE Donors (
   DonorID INT IDENTITY(1,1) PRIMARY KEY,
   FirstName VARCHAR(50) NOT NULL,
   LastName VARCHAR(50) NOT NULL,
   Gender VARCHAR(10),
   BirthDate DATE,
   BloodType VARCHAR(3) NOT NULL,
   ContactNumber VARCHAR(15),
   Email VARCHAR(50),
   Address VARCHAR(100),
   LastDonationDate DATE
);
```

Output:-

SQLQuery1.sql - PT...YODA\tapan.k (70))*

```
ADD EmployeeID INT;

ALTER TABLE Recipients
ADD CONSTRAINT FK_Recipients_Employees
FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID);

INSERT INTO Donors (FirstName, LastName, Gender, BirthDate, BloodType, ContactNumber, Email, Address, LastDonationDate)
VALUES
('John', 'Doe', 'Male', '1980-01-15', 'O+', '1234567890', 'john.doe@example.com', '123 Elm Street', '2023-06-01'),
('Jane', 'Smith', 'Female', '1990-02-20', 'A-', '2345678901', 'jane.smith@example.com', '456 Oak Street', '2023-05-15'),
('Mike', 'Johnson', 'Male', '1985-03-25', 'B+', '3456789012', 'mike.johnson@example.com', '789 Pine Street', '2023-07-10'),
('Sara', 'Williams', 'Female', '1995-04-30', 'AB-', '4567890123', 'sara.williams@example.com', '321 Maple Street', '2023-06-20'),
('Tom', 'Brown', 'Male', '1975-05-10', 'O-', '5678901234', 'tom.brown@example.com', '654 Birch Street', '2023-07-01');

select * from Donors

INSERT INTO BloodUnits (BloodType, CollectionDate, ExpiryDate, DonorID, Status)
```

100 %

Results | Messages

	DonorID	FirstName	LastName	Gender	BirthDate	BloodType	ContactNumber	Email	Address	LastDonationDate
1	1	John	Doe	Male	1980-01-15	O+	1234567890	john.doe@example.com	123 Elm Street	2023-06-01
2	2	Jane	Smith	Female	1990-02-20	A-	2345678901	jane.smith@example.com	456 Oak Street	2023-05-15
3	3	Mike	Johnson	Male	1985-03-25	B+	3456789012	mike.johnson@example.com	789 Pine Street	2023-07-10
4	4	Sara	Williams	Female	1995-04-30	AB-	4567890123	sara.williams@example.com	321 Maple Street	2023-06-20
5	5	Tom	Brown	Male	1975-05-10	O-	5678901234	tom.brown@example.com	654 Birch Street	2023-07-01

Create BloodUnits Table

```
CREATE TABLE BloodUnits (
    UnitID INT IDENTITY(1,1) PRIMARY KEY,
    BloodType VARCHAR(3) NOT NULL,
    CollectionDate DATE NOT NULL,
    ExpiryDate DATE NOT NULL,
    DonorID INT,
    Status VARCHAR(20) DEFAULT 'Available',
    FOREIGN KEY (DonorID) REFERENCES Donors(DonorID)
);
```

Output:-

SQLQuery1.sql - PT...YODA\tapan.k (70))*

```
('Tom', 'Brown', 'Male', '1975-05-10', 'O-', '5678901234', 'tom.brown@example.co

select * from Donors

INSERT INTO BloodUnits (BloodType, CollectionDate, ExpiryDate, DonorID, Status)
VALUES
('O+', '2023-06-01', '2023-09-01', 1, 'Available'),
('A-', '2023-05-15', '2023-08-15', 2, 'Available'),
('B+', '2023-07-10', '2023-10-10', 3, 'Available'),
('AB-', '2023-06-20', '2023-09-20', 4, 'Available'),
('O-', '2023-07-01', '2023-10-01', 5, 'Available');

select * from BloodUnits
```

100 %

Results | Messages

	UnitID	BloodType	CollectionDate	ExpiryDate	DonorID	Status
1	1	O+	2023-06-01	2023-09-01	1	Available
2	2	A-	2023-05-15	2023-08-15	2	Available
3	3	B+	2023-07-10	2023-10-10	3	Available
4	4	AB-	2023-06-20	2023-09-20	4	Available
5	5	O-	2023-07-01	2023-10-01	5	Available

Create Recipients Table

```
CREATE TABLE Recipients (
    RecipientID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Gender VARCHAR(10),
    BirthDate DATE,
    BloodType VARCHAR(3) NOT NULL,
    ContactNumber VARCHAR(15),
    Email VARCHAR(50),
    Address VARCHAR(100)
);
```

Output:-

SQLQuery1.sql - PT...YODA\tapan.k (70))*

```
select * from BloodUnits

INSERT INTO Recipients (FirstName, LastName, Gender, BirthDate, BloodType, ContactNumber, Email, Address, EmployeeID)
VALUES
('Tapan', 'Patel', 'Male', '1990-05-15', 'A+', '1234567890', 'tapan.patel@example.com', '123 Lakeview Street', 1),
('Sam', 'Johnson', 'Male', '1985-08-20', 'O+', '2345678901', 'sam.johnson@example.com', '456 Riverside Avenue', 2),
('Ram', 'Singh', 'Male', '1992-12-10', 'B+', '3456789012', 'ram.singh@example.com', '789 Highland Road', 3),
('John', 'Doe', 'Male', '1988-03-05', 'AB-', '4567890123', 'john.doe@example.com', '321 Mountain Drive', 4),
('Eva', 'Smith', 'Female', '1995-07-25', 'O-', '5678901234', 'eva.smith@example.com', '654 Forest Lane', 5);
Select *from Recipients
```

100 %

Results | Messages

	RecipientID	FirstName	LastName	Gender	BirthDate	BloodType	ContactNumber	Email	Address	EmployeeID
1	2	Tapan	Patel	Male	1990-05-15	A+	1234567890	tapan.patel@example.com	123 Lakeview Street	1
2	3	Sam	Johnson	Male	1985-08-20	O+	2345678901	sam.johnson@example.com	456 Riverside Avenue	2
3	4	Ram	Singh	Male	1992-12-10	B+	3456789012	ram.singh@example.com	789 Highland Road	3
4	5	John	Doe	Male	1988-03-05	AB-	4567890123	john.doe@example.com	321 Mountain Drive	4
5	6	Eva	Smith	Female	1995-07-25	O-	5678901234	eva.smith@example.com	654 Forest Lane	5

Create Transactions Table

```
CREATE TABLE Transactions (
    TransactionID INT IDENTITY(1,1) PRIMARY KEY,
    UnitID INT NOT NULL,
    DonorID INT,
    RecipientID INT,
    TransactionType VARCHAR(20) NOT NULL, -- 'Donation' or 'Transfusion'
```

```
    TransactionDate DATE NOT NULL,
    FOREIGN KEY (UnitID) REFERENCES BloodUnits(UnitID),
    FOREIGN KEY (DonorID) REFERENCES Donors(DonorID),
    FOREIGN KEY (RecipientID) REFERENCES Recipients(RecipientID)
);
```

Output:-

SQLQuery1.sql - PT...YODA\tapan.k (70))*

```
INSERT INTO Transactions (UnitID, DonorID, RecipientID, TransactionType, TransactionDate)
VALUES
(1, 1, NULL, 'Donation', '2023-06-01'),
(2, 2, NULL, 'Donation', '2023-05-15'),
(3, 3, NULL, 'Donation', '2023-07-10'),
(4, 4, NULL, 'Donation', '2023-06-20'),
(5, 5, NULL, 'Donation', '2023-07-01');

INSERT INTO Transactions (UnitID, DonorID, RecipientID, TransactionType, TransactionDate)
VALUES
(1, NULL, 2, 'Transfusion', '2023-06-05'),
(2, NULL, 3, 'Transfusion', '2023-05-20'),
(3, NULL, 4, 'Transfusion', '2023-07-15'),
(4, NULL, 5, 'Transfusion', '2023-06-25'),
(5, NULL, 6, 'Transfusion', '2023-07-05');

select * from  Transactions
```

100 %

Results | Messages

	TransactionID	UnitID	DonorID	RecipientID	TransactionType	TransactionDate
1	19	1	1	NULL	Donation	2023-06-01
2	20	2	2	NULL	Donation	2023-05-15
3	21	3	3	NULL	Donation	2023-07-10
4	22	4	4	NULL	Donation	2023-06-20
5	23	5	5	NULL	Donation	2023-07-01
6	26	1	NULL	2	Transfusion	2023-06-05
7	27	2	NULL	3	Transfusion	2023-05-20
8	28	3	NULL	4	Transfusion	2023-07-15
9	29	4	NULL	5	Transfusion	2023-06-25
10	30	5	NULL	6	Transfusion	2023-07-05

Create Employees Table

```
CREATE TABLE Employees (
```

```
    EmployeeID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Gender VARCHAR(10),
    BirthDate DATE,
    ContactNumber VARCHAR(15),
    Email VARCHAR(50),
    Address VARCHAR(100),
    Position VARCHAR(50),
    HireDate DATE
);
```

Output:-

SQLQuery1.sql - PT...YODA\tapan.k (70))*

```
(4, NULL, 5, 'Transfusion', '2023-06-25'),
(5, NULL, 6, 'Transfusion', '2023-07-05');

select * from  Transactions

INSERT INTO Employees (FirstName, LastName, Gender, BirthDate, ContactNumber, Email, Address, Position, HireDate)
VALUES
('Emily', 'Taylor', 'Female', '1985-01-15', '1231231234', 'emily.taylor@example.com', '1234 Oak Avenue', 'Admin', '2015-05-01'),
('James', 'Anderson', 'Male', '1979-02-20', '2342342345', 'james.anderson@example.com', '5678 Maple Avenue', 'Staff', '2016-06-10'),
('Olivia', 'Thomas', 'Female', '1990-03-25', '3453453456', 'olivia.thomas@example.com', '9101 Birch Avenue', 'Staff', '2017-07-20'),
('William', 'Moore', 'Male', '1983-04-30', '4564564567', 'william.moore@example.com', '1121 Pine Avenue', 'Staff', '2018-08-15'),
('Sophia', 'Jackson', 'Female', '1995-05-10', '5675675678', 'sophia.jackson@example.com', '3141 Cedar Avenue', 'Admin', '2019-09-25');
use DBProject_tapan

select * from Employees
```

100 %

Results | Messages

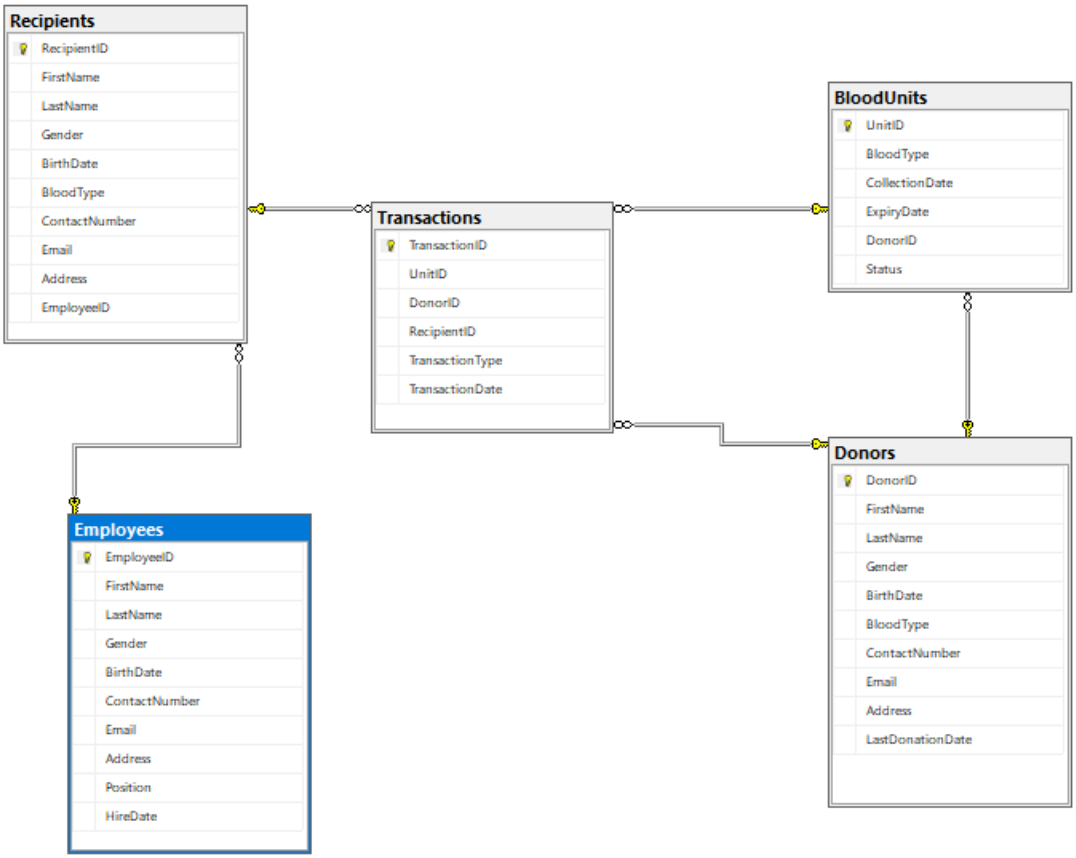
	EmployeeID	FirstName	LastName	Gender	BirthDate	ContactNumber	Email	Address	Position	HireDate
1	1	Emily	Taylor	Female	1985-01-15	1231231234	emily.taylor@example.com	1234 Oak Avenue	Admin	2015-05-01
2	2	James	Anderson	Male	1979-02-20	2342342345	james.anderson@example.com	5678 Maple Avenue	Staff	2016-06-10
3	3	Olivia	Thomas	Female	1990-03-25	3453453456	olivia.thomas@example.com	9101 Birch Avenue	Staff	2017-07-20
4	4	William	Moore	Male	1983-04-30	4564564567	william.moore@example.com	1121 Pine Avenue	Staff	2018-08-15
5	5	Sophia	Jackson	Female	1995-05-10	5675675678	sophia.jackson@example.com	3141 Cedar Avenue	Admin	2019-09-25

3. ER Diagram

The Entity-Relationship (ER) diagram visually represents the relationships between these tables. Below is a textual description of the ER diagram:

- **Donors** (DonorID) 1-to-Many **BloodUnits** (DonorID)
- **BloodUnits** (UnitID) Many-to-1 **Transactions** (UnitID)

- **Donors** (DonorID) Many-to-1 **Transactions** (DonorID)
- **Recipients** (RecipientID) Many-to-1 **Transactions** (RecipientID)
- **Employees** (EmployeeID) One to One **Recipients** (RecipientID)



4. Example Queries for Data Extraction

1.Retrieve All Records from a Table

```
SELECT * FROM Donors;
```

Results | Messages

	DonorID	FirstName	LastName	Gender	BirthDate	BloodType	ContactNumber	Email	Address	LastDonationDate
1	1	John	Doe	Male	1980-01-15	O+	1234567890	john.doe@example.com	123 Elm Street	2023-06-01
2	2	Jane	Smith	Female	1990-02-20	A-	2345678901	jane.smith@example.com	456 Oak Street	2023-05-15
3	3	Mike	Johnson	Male	1985-03-25	B+	3456789012	mike.johnson@example.com	789 Pine Street	2023-07-10
4	4	Sara	Williams	Female	1995-04-30	AB-	4567890123	sara.williams@example.com	321 Maple Street	2023-06-20
5	5	Tom	Brown	Male	1975-05-10	O-	5678901234	tom.brown@example.com	654 Birch Street	2023-07-01

2.Filter Records Based on a Condition

- Query Task: Select all transactions from the transactions table where the transaction date is after January 1, 2023.

```
SELECT * FROM Transactions
WHERE TransactionDate > '2023-07-01';
```

OUTPUT:-

Results | Messages

	TransactionID	UnitID	DonorID	RecipientID	TransactionType	TransactionDate
1	21	3	3	NULL	Donation	2023-07-10
2	28	3	NULL	4	Transfusion	2023-07-15
3	30	5	NULL	6	Transfusion	2023-07-05

3.Join Two Tables

- Query Task: Retrieve the names of donors along with their transaction IDs from the donors and transactions tables.

```
SELECT Donors.FirstName, Donors.LastName, Transactions.TransactionID
FROM Donors
INNER JOIN Transactions ON Donors.DonorID = Transactions.DonorID;
```

OUTPUT:-

Results | Messages

	FirstName	LastName	TransactionID
1	John	Doe	19
2	Jane	Smith	20
3	Mike	Johnson	21
4	Sara	Williams	22
5	Tom	Brown	23

4.Aggregate Data Using Group By

- Query Task: Find the total number of transactions for each donor.

```
SELECT DonorID, COUNT(*) AS TotalTransactions
FROM Transactions
GROUP BY DonorID;
```

OUTPUT:-

Results | Messages

	FirstName	LastName	TransactionID
1	John	Doe	19
2	Jane	Smith	20
3	Mike	Johnson	21
4	Sara	Williams	22
5	Tom	Brown	23

5.Filter Groups Using HAVING

- Query Task: Retrieve the donor IDs and their total number of transactions, but only for donors who have less than 5 transactions.

```
SELECT DonorID, COUNT(*) AS TotalTransactions
```

```
FROM Transactions
GROUP BY DonorID
HAVING COUNT(*) < 5;
```

OUTPUT:-

Results | Messages

	DonorID	TotalTransactions
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1

6.Order Results Using ORDER BY

- Query Task: Select all blood units from the blood units table and order them by collection date in descending order.

```
SELECT * FROM BloodUnits
ORDER BY CollectionDate DESC;
```

OUTPUT:-

Results | Messages

	UnitID	BloodType	CollectionDate	ExpiryDate	DonorID	Status
1	3	B+	2023-07-10	2023-10-10	3	Available
2	5	O-	2023-07-01	2023-10-01	5	Available
3	4	AB-	2023-06-20	2023-09-20	4	Available
4	1	O+	2023-06-01	2023-09-01	1	Available
5	2	A-	2023-05-15	2023-08-15	2	Available

7.Retrieve Data with a Subquery

Query Task: Find the names of donors who have made donations with less than 2 units.

```
SELECT FirstName, LastName
FROM Donors
WHERE DonorID IN (
   SELECT DonorID
   FROM Transactions
   WHERE TransactionType = 'Donation'
   GROUP BY DonorID
   HAVING COUNT(UnitID)  <2
);
```

OUTPUT:-

Results | Messages

	FirstName	LastName
1	John	Doe
2	Jane	Smith
3	Mike	Johnson
4	Sara	Williams
5	Tom	Brown

8.Use CASE Statements

- Query Task: Retrieve transaction details along with a column that indicates if the transaction type is 'Donation' or 'Transfusion'.

```
SELECT TransactionID, UnitID, DonorID, RecipientID, TransactionDate,
      CASE
         WHEN TransactionType = 'Donation' THEN 'Donated'
         WHEN TransactionType = 'Transfusion' THEN 'Transfused'
         ELSE 'Other'
      END AS TransactionStatus

         FROM Transactions;
```

OUTPUT:

Results | Messages

	TransactionID	UnitID	DonorID	RecipientID	TransactionDate	TransactionStatus
1	19	1	1	NULL	2023-06-01	Donated
2	20	2	2	NULL	2023-05-15	Donated
3	21	3	3	NULL	2023-07-10	Donated
4	22	4	4	NULL	2023-06-20	Donated
5	23	5	5	NULL	2023-07-01	Donated
6	26	1	NULL	2	2023-06-05	Transfered
7	27	2	NULL	3	2023-05-20	Transfered
8	28	3	NULL	4	2023-07-15	Transfered
9	29	4	NULL	5	2023-06-25	Transfered
10	30	5	NULL	6	2023-07-05	Transfered