

**ADDIS ABABA UNIVERSITY**



**College of Natural and Computational Sciences**

**SCHOOL OF INFORMATION SCIENCE**

**FUNDAMENTALS OF DATA BASE**

**Group Members ID**

1. **Bezawit Lulekal ………………………………….UGR/3818/15**
2. **Hewan Zewdie…………………………………..UGR/3869/15**

**3. Mariamawit Getachew ……………………..UGR/6508/15**

**4. Mihret Daniel ……………………………………UGR/ 6495/15**

**5. Ribka Muluye ……………………………………UGR/9838/15**

**4. Setota Girma …………………………………….UGR/0524/15**

**SUBMITTED TO: - Teacher Adey**

**SUBMITTED DATE: - 1/25/2024**

**TABLE OF CONTENTS**

[Acknowledgment 2](#_Toc128649726)

[1. Introduction 3](#_Toc128649727)

[1.1. Background 3](#_Toc128649728)

1.2 Purpuse of the system....................................................................................................... 4

[1.3. Objectives of the system...........................................................................................................4](#_Toc128649729)

[1.4. Problems in the existing system..............................6](#_Toc128649730)

[1.5. Scope of the system..............................................................................................7](#_Toc128649732)

1.6 Database Develompment Methodology........................................................................8.. 1.7 Deliverables of the project........................................................................................10.....

1.8 Development Tools, Platforms and Technologies.....................................................11......

1.9 project time plan.......................................................................................................11.....

2. Requirments Specification...................................................................................................12...

2.1 Data requirements of the system..................................................................................12..

2.2 Transaction requirnments..............................................................................................13....

[3. Database design.........................................................................................................................1](#_Toc128649733)4

[3.1. Conceptual database design...............................................................................................1](#_Toc128649734)4

[3.1.1. Entity identification and description........................................................................1](#_Toc128649735)4

[3.1.2. Attributes identification and description...............................................................1](#_Toc128649736)5

[3.1.3. Relationship identification and description............................................................1](#_Toc128649737)5

[3.1.4. Entity relationship (ER) diagram..........................................................................1](#_Toc128649738)8

[3.2. Logical database design...............................................................................................1](#_Toc128649739)9

[3.2.1. ER to table mapping.](#_Toc128649740)..........................................................................................19.

[3.2.2. Validate model using normalization................................................................2](#_Toc128649741)0

[3.2.3. Relational Schema with Referential Integrity after normalization...................2](#_Toc128649742)3

[3.3. Physical database design of the new system.................................................................2](#_Toc128649743)5

[3.3.1. Physical design strategy......................................................................................2](#_Toc128649744)5

[3.3.2. Hardware implementation...................................................................................2](#_Toc128649745)7

[3.3.3. Front end design outline.....................................................................................2](#_Toc128649746)8

[4. Implementation............................................................................................................2](#_Toc128649747)9

**[ANNEXES............................................................................................................................3](#_Toc128649748)**3

[Group collaboration statement........................................................................................3](#_Toc128649749)3

# **Acknowledgment**

We would first want to express our gratitude to Teacher Adey, our instructor, for assigning a project that allowed us to explore our favorite topic. She provides us with a good insight into what we are going to do in our project. We also appreciate our parents' material and mental assistance. They are giving us the material tools we need to complete this project and providing the moral support that keeps us going every day. We also appreciate the hard work and dedication of each group member, without which this project would not have been completed.

**We are most grateful to God, who bestowed upon us all wisdom and the ability to accomplish this Project.**

**Chapter 1 Introduction**

Blood is a vital fluid that flows through the body carrying vital components and carrying out vital tasks that are required for life. It is essential for delivering waste products, nutrition, hormones, and oxygen to different organs and tissues.

Blood is made up of platelets, white blood cells, red blood cells, and plasma. Blood has several uses. It can be transfused to other persons when there are some conditions like giving birth or overflow of blood from our body. This transfusion will help to save one’s life. The information about this key life saver should be kept carefully in order to maintain the health of the society. Therefore, we initiated this project to help improve data monitoring in blood banks known as “BLOOD DONATION MANAGEMENT SYSTEM.”

**1.1 Back Ground**

Blood banks are essential to the life-saving transfusion of blood from donors to recipients all over the world. However, the limitations of manual and file-based systems make it difficult for many blood banks to manage and handle blood effectively. These challenges may result in problems with data integrity, duplication, inflexibility, and erroneous decision-making.

We suggest creating an extensive **"Blood Donation Management System"** in order to solve these issues and enhance blood management. A strong and effective database will be used by this system to optimize the flow of blood from donors at blood banks to hospitals and, eventually, to the patients in need.

Databases offer benefits like minimizing duplication, promoting flexible data flow, enabling timely access, and preserving data integrity. These advantages enable blood banks and hospitals to make informed decisions based on accurate data. Customizing databases to meet specific blood bank requirements optimizes daily operations and ensures access to superior data.

**1.2. Purpose of the system**

The purpose of the Blood Donation Management System is to streamline and optimize the process of blood donation, collection, distribution, and management. It aims to connect blood donors, recipients, and blood banks efficiently to ensure timely availability of blood units, improve patient care, and ultimately save lives.

**1.3 Objective of the system**

Our database system's goal is to give blood banks the greatest data management solution possible. The important choices made by blood banks are therefore supported by this data management system. The accomplishment of this overall success will come from the cooperative application of several alternate goals.

Let’s look some objectives:-

**1.3.1. General Objectives : -**

* Enhance Data Management: - Provide a solid framework for managing data so that blood banks may streamline their operations and make well-informed decisions.
* Strengthen Cooperation: - Provide simultaneous access to data for several users, encouraging efficient teamwork and real-time updates.
* Guarantee Data Security: - To guarantee data security and uphold the accuracy and consistency of information, establish user authorization levels.
* Optimize Resource Allocation: - To help managers make fast and informed decisions about the distribution of blood units, provide precise and current information on blood availability.
* Motivate Making decisions based on evidence: - Facilitate the process of extracting valuable insights from long-term data and bolster evidence-based policies, plans, and strategic planning.
* Improve Efficiency: - In blood bank operations, streamline processes to cut down on duplication, decrease errors, and save time and energy.

**1.2.2. Specific Objectives : -**

* Real-time Blood Availability: - Provide a mechanism that enables managers to ascertain the kind and quantity of blood that is available for distribution with speed and accuracy.
* Longer Data Lifespan and Insights: - Build the system so that data may be analyzed and stored for a longer time, giving blood banks the opportunity to draw important conclusions.
* Well-Informed Decision Making: - Give managers access to accurate and up-to-date information so they can better allocate resources and improve blood bank operations as a whole.
* Levels of User Authorization: - Establish distinct user permission levels to guarantee data security and quality by giving users the proper access privileges.
* Smooth Information Flow: - Establish a system that makes it easier for hospitals and blood banks to exchange information, promoting efficient teamwork and quick access to critical data.
* Data Accuracy and Consistency: Reduce duplicate data entry, eliminate unnecessary information, and guarantee data accuracy and consistency across the system.
* Superior Quality Data for Examination: - Assist evidence-based policy and strategy planning by enabling the system to supply high-quality data for a variety of analysis and interpretations.
* Time and Energy Efficiency: - Provide a system that minimizes the amount of time and energy needed to process data pertaining to blood, hence increasing operational efficiency.
* Error Mitigation using Restraints: In order to reduce errors and guarantee data accuracy, dependability, and reliability, implement limitations in data entry operations.

The blood donation system seeks to accomplish these broad and targeted goals in order to strengthen the capacity of blood banks to save lives in an efficient and effective manner, as well as to optimize decision-making procedures and blood resource administration.

**1.4 Problem Solving**

Since the main goal of our project is to create a database that will enable blood banks to provide high-quality data flow, we assumed that the current system was either manual or file-based. Blood banks' databases logically address the following issues, which were the main ones caused by the previous systems. Let’s see the Problems: -

* Inefficient Data Management
* Disclosure of recipients’ medical records to unauthorized bodies
* Human-made errors in collection, processing and presentation of data
* Complexity to update data and Ineffective Data Processing
* Security issues such as illegal blood transfusions
* High risk of blood data loss and Limited Access
* Time delay in processing large amount of blood data
* Massive amount of accumulated papers in laboratories
* Time delay in record retrieval with some particular property and generating reports.

**1.5 Scope of this system**

The following important blood bank-related activities are included in our project: -

1. **Donor Registration and Information Management:**

* Creating a system for the registration and updating of donor data, such as contact details, medical history, and eligibility requirements.
* Putting in place features for viewing and searching donor data for convenience of access and retrieval.

1. **Summarized Data: -**  Creating tools to offer condensed data, like blood type distribution, donor trends, and donor statistics, to support analysis and decision-making.
2. **Blood Inventory Management:**

* Developing a module to monitor and control the different kinds of blood units that are kept on hand at the blood bank.
* Enabling features to keep an eye on blood unit quantities, storage locations, and expiration dates.

1. **Donation and Transfusion Recording:**

* Establishing a system to document and monitor each blood transfusion, including recipient details, blood type compatibility, and transfusion specifics.
* implementing features that record donor information, donation type, and donation history, and register and monitor each blood donation.

1. **Blood Distribution to Hospitals:**

* Creating a module to oversee the delivery of blood units to medical centers or hospitals in accordance with their compatibility and requirements.

1. **Information Surveillance and Upgrades:**

* Including tools that make it simple for the manager of the blood bank to keep an eye on how information is moving through the system.
* giving the manager the ability to alter donor data, the blood inventory, and other pertinent information as needed.

The blood bank system's fundamental functions and activities, such as donor management, blood inventory tracking, transfusion and donation recording, blood distribution, and information monitoring, are covered by the scope that was previously described.

**1.6 Database Development Methodology**

**1.6.1 Data Sources & Collection Methods**

**Requirements Gathering and Analysis**

* **Stakeholder Engagement: -** Involves blood banks, hospitals, donors, and healthcare professionals to understand their basic needs and data requirements.

Data sources for the Blood Donation Management System may include:

* Registration forms, including personal details, medical history, and blood type.
* Blood bank inventory data, including available blood units, expiration dates, and storage conditions.
* Recipient information, such as medical history, blood type, and transfusion requirements.
* Collection methods may involve online registration forms, manual data entry, integration with health records systems, and periodic updates from blood banks.
* **Defining the Data: -** Defines critical data elements including donor information, blood type details, inventory levels, and donation history, establishing standardized data formats and terminology.
* **Mapping the User Journey: -** Identifies how various user groups will interact with the database, analyzing their specific needs and data access levels required for each role.

**1.6.2 DB Analysis and Design Methods**

**1. Database Design**

* Conceptual Design: Creates a broad data model showcasing entities such as donors, blood banks, and hospitals, alongside their connections and characteristics using tools like Entity-Relationship Diagrams (ERDs).
* Logical Design: Converts the conceptual model into a precise database structure compatible with the selected DBMS (e.g., MySQL), specifying tables, attributes, data types, relationships (primary/foreign keys), and constraints.
* Physical Design: Fine-tunes the structure for maximum performance and storage effectiveness, considering strategies like normalization to simplify data retrieval and reduce duplication.

**2. Implementation**

* Building the Environment: Establishes a development environment equipped with the chosen DBMS, development tools, and testing frameworks.
* Coding and Scripting: Develops SQL statements and database objects to construct the database schema and implement functionalities.
* Data Migration and Integration: Migrates existing data from legacy systems or other sources, ensuring data integrity and consistency.

**3. Testing and Deployment**

* Unit Testing: Tests individual functionalities within the database logic.
* Integration Testing: Ensures the database interacts seamlessly with other components of the BDMS.
* Performance Testing: Evaluates database performance under load to identify and address bottlenecks.
* Deployment: Deploys the tested database to a production environment with proper security measures and access control.

1. **Continuous Evolution**: Maintenance and Growth

* Monitoring and Performance Tuning: Monitors database performance and resource usage, making adjustments as needed.
* Security Updates: Applies regular security patches and updates to the DBMS and its components.
* Data Archiving and Backups: Implements robust data backup and archiving strategies to ensure data safety and recovery.
* Evolution and Adaptability: Adapts the database as user needs and regulations evolve, utilizing Agile methodologies for flexible adjustments and iterative development.

**1.7 Deliverables of the Project**

* SQL code used to build the database: a SQL statement-based script to build the database that will store the data for the blood donation system.
* Table, view, and index creation SQL scripts: These scripts generate the required database tables, views, and indexes in accordance with the logical database architecture.
* Requirements for Data Entry: Create and execute user forms or user interfaces that ensure correct and efficient data entry of blood donation information into the system.
* The implementation of queries and reports to extract particular data from the database is necessary for data retrieval. This will enable users to view and examine blood donation data as needed.
* Development of features for updating and changing already-existing data in the database, enabling adjustments and corrections to be made as needed, is required for data update.
* Data Removal Requirement: To ensure appropriate data management and compliance with data protection legislation, processes must be implemented to delete or archive data that is no longer relevant or needed.
* The process of putting the conceptual database design into practice involves creating an entity-relationship (E-R) diagram and designing the entities, characteristics, and relationships that make up the design.
* Logical database design is the process of translating an idea-based database design into a logical database design, guaranteeing normalization and getting rid of redundant information and odd data.
* Validating the Data Model with Normalization: Verifying that the database architecture complies with the first, second, and third normal forms (1NF, 2NF, and 3NF) of normalization.
* Relational Schema with Referential Integrity: Using the logical design as a basis, create a relational schema and include restrictions on referential integrity to preserve data integrity.
* Physical Database Design: Outlining the approach for the physical database design, which includes choosing the best partitioning, indexing, and storage configurations.
* The documentation and deployment plan for installing and configuring the database in the production environment, along with considerations for backup and recovery procedures, are the details of the database deployment.
* Testing: Complete testing protocols, such as unit, integration, and user acceptability testing, are carried out to confirm the blood donation system's functionality, performance, and dependability.

Together, these deliverables guarantee the blood donation system's successful testing and deployment, offering a dependable and efficient database solution for blood bank data management.

**1.8 Development Tools, Platforms and Technologies**

Database management System : - Microsoft SQL for data storage and management.

**1.9 Project Time Plan**

* Requirements Gathering and Analysis : - 4-7 days
* System Design and Database Design : - 1-2 weeks
* Development and Implementation : - 8-10 days
* Project Documentation : - 6-10 days

**Chapter 2** **Requirement Specification**

**2.1 Data requirements**

These data requirements actually enable for the safe and efficient management of blood transfusion procedures, donor involvement, inventory control, and blood donation processes. The requirements of this system include: -

* **Donor Data:** - Gathering and keeping track of pertinent donor data, including name, phone number, age, gender, blood type, medical background, qualifying conditions, and donation preferences.
* **Blood Units and Components: -** Monitoring and organizing data pertaining to blood units and components, such as blood type, volume, location of storage, dates of expiration, and testing and safety status.
* **Donation History: -** Keeping track of a donor's prior donations, including the dates, kinds of donations (platelets, plasma, whole blood, etc.), and any negative outcomes or deferrals.
* **Transfusion Records: -** Documenting specifics about blood transfusions, such as recipient details, compatibility of blood types, date of transfusion, and any complications or adverse reactions.
* **Inventory management: -** Includes tracking stock levels, storage conditions, and utilization rates in real-time as well as keeping an eye on the availability and status of blood units and components.
* **Testing and Screening Results: -** Recording and data storage of the outcomes of tests and screenings—such as blood typing, compatibility testing, and infectious disease screening—conducted on donated blood.
* **Donor Consent and Privacy: -**  Preserving documentation of blood donors' consent and making sure privacy laws are followed to safeguard private donor data.
* **Communication and Notifications: -** Maintaining a record of communications, setting up appointments, and notifying donors of updates, reminders, and appreciation.
* **Reporting and Analytics: -** Producing reports and analytics to offer information on trends in blood donation, the characteristics of donors, the rates at which blood is used, and other important performance metrics for quality control and decision-making.

**2.2 Transaction Requirements**

**2.2.1 Data Entry Requirements**

- Donor Registration: - Allow users to register as blood donors, entering personal information such as name, contact details, age, gender, blood type, and medical history.

- Blood Donation Records: - Enable recording of blood donation events, including donor details, donation date, blood type, and quantity donated.

- Recipient Registration: - Capture recipient information, including name, contact details, medical history, blood type, and transfusion requirements.

- Blood Bank Inventory Management: - Facilitate the entry of blood units into the inventory, including details such as blood type, donation date, expiry date, and storage location.

**2.2.2 Data Retrieval Requirements**

- Donor Search: Allow searching for donors based on criteria such as blood type, location, and availability.

- Blood Bank Inventory Lookup: Enable querying the blood bank inventory to retrieve available blood units based on criteria such as blood type, quantity, and expiry date.

- Recipient Matching: Retrieve suitable blood donors for specific recipients based on compatibility factors such as blood type and location.

**2.2.3 Data Updating Requirements**

- Donor Profile Updates: Allow donors to update their personal information and medical history as needed.

- Blood Donation Records Updates: Enable editing of donation records to correct errors or update donation details.

- Blood Bank Inventory Updates: Support adjustments to the inventory, including adding new blood units, updating quantities, and marking units as expired or used.

**2.2.4 Data Removal Requirements:**

- Donor Deletion: Allow for the removal of donor records upon request or in case of duplicate entries or inaccuracies.

- Blood Donation Records Deletion: Enable the deletion of donation records that are no longer relevant or accurate.

- Blood Bank Inventory Management: Support the removal of expired or unusable blood units from the inventory to maintain data accuracy.

These transaction requirements ensure that the Blood Donation Management System can effectively handle data entry, retrieval, updating, and removal processes to support the operations of blood donation centers, blood banks, and healthcare facilities.

**Chapter 3 Database des****ign**

* 1. **Conceptual database design**

**3.1.1 Entity identification and description**

* **Hospital: -** Represents a health-care facility or receives blood units from the blood-bank for transfusions.
* **Donor: -** an Individual who voluntarily donates blood.
* **Blood unit: -** a unit of donated blood that is collected, processed and stored in the blood bank.
* **Transfusion: -** the process of transferring blood from a donor to a recipient for medical treatment.
* **Blood bank: -** An organization responsible for collecting , storing, and distributing blood.
* **Staff: -** Represents the employees and personnel working at the blood bank.
* **Recipient: -** An individual who receives a blood transfusion at a health care facility.
* **Donation: -** A specific instance of a donor donating blood.

**3.1.2 Attributes identification and description**

* **Hospital: - Hospital ID, Name, Phone number, Location. HospitalID and phone number are together composite primary key.**
* **Donor: - donor id, First name, last name, gender, date of birth, phone number, address, blood group, and Age. Donor ID, phone number and Address are together composite primary key.**
* **Blood quantity: - blood bank id, blood group, amount, Collection date and expiration date.composite key :- Blood Bank ID and Blood Group.**
* **Transfusion: donor id, blood bank id, recipient id, hospital id, date, time, and amount transferred.composite primary key:- donor id, blood bank id, recipient id, hospital id, date.**
* **Blood bank: - Blood bank ID, Name, Location, Phone number.Blood bank ID and phone number are together composite primary key.**
* **Staff: - staff id, first name, last name, position, DOB, gender, phone number, salary and address. Staff ID, phone number and Address are together composite primary key.**
* **Recipient: - Recipient ID, FirstName, Last name, age, Phone number, Gender, Blood group, Adress, Disease.Recipient ID, phone number, address and disease are together composite primary key.**
* **Donation: - donor id, blood bank id, staff id, date, amount donated and donation type. Composit key:- DonorID, blood bankID and StaffID, date.**

**3.1.3 Relationship identification and description**

1. **Donors – donations**:

* Relationship: Their relationship is 'donor donates blood.' This is established by posting the donor's primary key in the donation entity.
* Cardinality: Every donor must have at least one donation recorded, but each donation record is only for one donor.

2. **Donors – transfusions**:

* Relationship: Their relationship is 'donor gives blood for transfusion.' This is established by posting the donor's primary key in the transfusion entity.
* Cardinality: Each donor can give zero, one, or more blood units for transfusion. Every transfusion has only one donor.

**3**. **Recipients - transfusions**:

* Relationship: Their relationship is 'recipient needs blood transfusion.' This is established by posting the recipient’s primary key in the transfusion entity.
* Cardinality: Every transfusion record has only one recipient.

4. **Blood banks – donations**:

* Relationship: Their relationship is 'blood bank stores donated blood.' This is established by posting the blood bank's primary key in the donation entity.
* Cardinality: Each blood bank must store at least one donation, but each donation is only recorded in one blood bank.

5. **Blood banks – transfusions**:

* Relationship: Their relationship is 'blood bank supplies blood for transfusion.' This is established by posting the blood bank's primary key in the transfusion entity.
* Cardinality: Each blood bank can contribute to zero, one, or more transfusions. Every transfusion record has only one blood bank associated with it.

6. **Blood banks - staff**:

They have two types of relationships

* Relationship: First, 'staff works at a blood bank.' This is established by posting the blood bank's primary key in the staff entity.
* Cardinality: Every blood bank must have at least one worker, but each staff member only works at one blood bank.
* Relationship: Second, 'staff manages a blood bank.' This is established by posting the staff's primary key in the blood bank entity.
* Cardinality: Each staff member might or might not be a manager at a specific blood bank, but each blood bank has only one manager.

7. **Blood banks – blood quantity**:

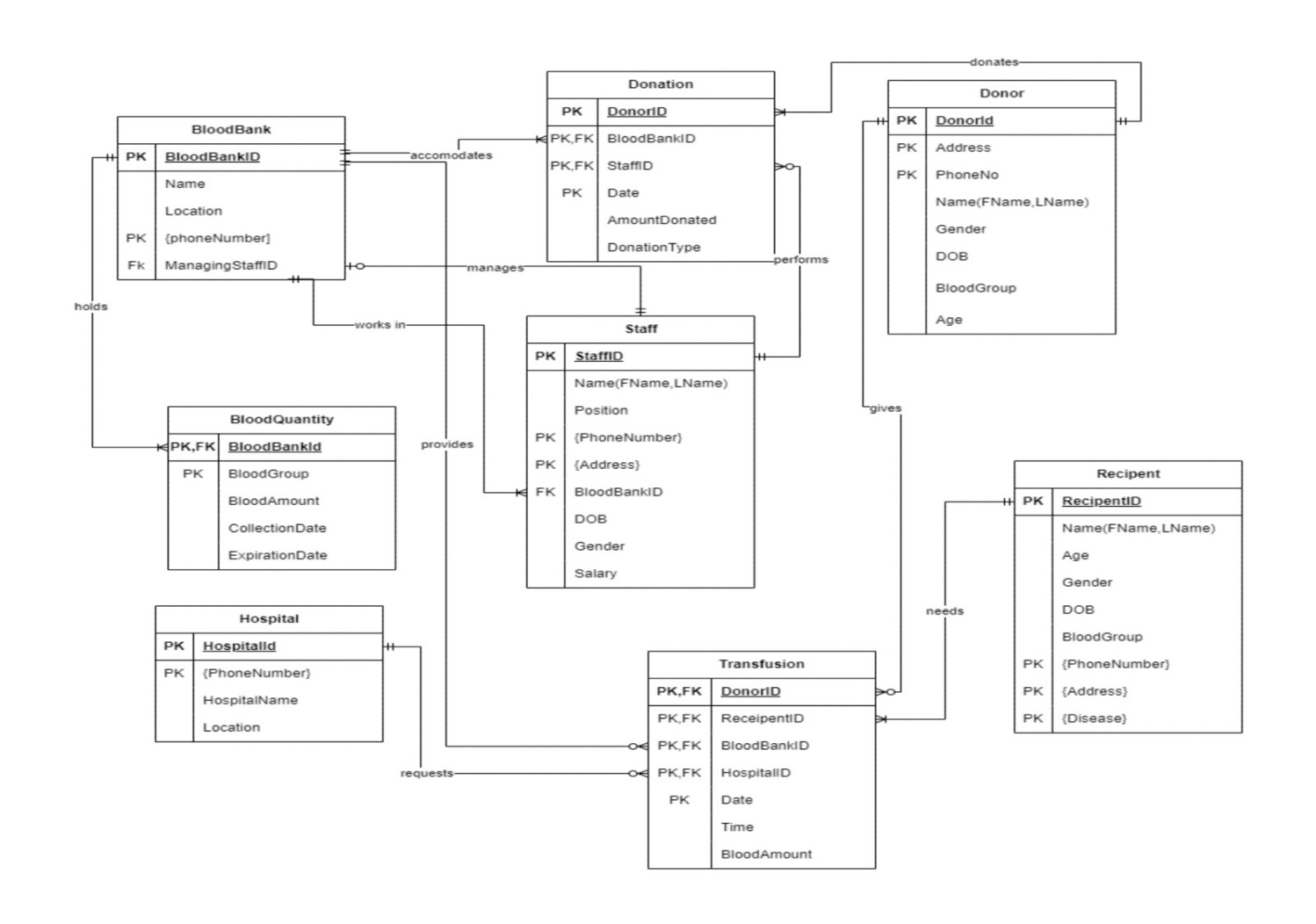
* Relationship: Their relationship is 'blood bank stores blood.' This is established by posting the blood bank's primary key in the blood entity.
* Cardinality: Each blood bank can store one or more records of blood quantities, but each record belongs to only one blood bank.

8. **Staff -donations**:

* Relationship: Their relationship is 'staff member handles blood donations.' This is established by posting the staff member's primary key in the donation entity.
* Cardinality: Each staff member can handle zero, one, or more donations, but each donation is only handled by one staff member.

9. **Hospitals - transfusions**:

* Relationship: Their relationship is 'hospital requests blood transfusion.' This is established by posting the hospital's primary key in the transfusion entity.
* Cardinality: Each hospital can request zero, one, or more transfusions, but each transfusion record is only requested by one hospital.

**3.1.4. Entity Relationship (ER) Diagram**

### **3.2 Logical Database Desighn**

### **3.2.1.** **ER to table mapping**

**DONOR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DonorID | FName | LName | Gender | DOB | ContactNumber | Address | BloodGroup | Age |

**DONATION**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DonorID | BloodBankID | StaffID | Date | AmountDonated | DonationType |

**BLOODBANK**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BloodBankID | Name | Location | PhoneNumber | MangingStaffID |

**BLOODQUANTITY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BloodBankID | BloodGroup | BloodAmount | CollectionDate | ExpirationDate |

**TRANSFUSION**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DonorID | BloodBankID | RecipientID | HospitalID | Date | Time | BloodAmount |

**RECIEPENT**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RecipentID | FName | LName | Gender | DOB | BloodGroup | PhoneNumber | Address |
| Disease |

**HOSPITAL**

|  |  |  |  |
| --- | --- | --- | --- |
| HospitalID | HospitalName | PhoneNumber | Location |

**STAFF**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| StaffID | | FName | | LName | | Position | PhoneNumber | Address | BloodBankID |
| Gender | DOB | | Salary | |

### **3.2.2. Validate model using normalization**

#### **3.2.2.1. First normal form (1NF)**

Luckily, all the tables adhere to the first normal form, meaning they satisfy the following conditions: \_

* Each table possesses a distinct name.
* Each attribute value is indivisible or atomic.
* Every row is distinct or unique.
* The attributes (columns) in tables have distinct names.
* Rows and columns are not ordered.

**3.2.2.2. Second normal form (2NF)**

Partial functional dependency causes several tables to be non-compliant with the second normal form. These are listed below:-

1. **DONOR table:**

* DonorID as a part of primary key:-

DonorID→FName, Lname, Gender, DOB, BloodGroup

* PhoneNumber as a part of primary key:-

PhoneNumber→FName, Lname, Gender, DOB, BloodGroup

* Address as a part of primary key doesn’t determine anything.

Normalized table:

DONOR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DonorID | FName | LName | Gender | DOB | BloodGroup |

DONORPHONE

|  |  |
| --- | --- |
| DonorID | PhoneNumber |

DONORADDRESS

|  |  |
| --- | --- |
| DonorID | Address |

1. **BLOODBANK table:**

* BloodBankID as a part of primary key:

BloodBankID→Name, Location, StaffID

* TelephoneNumber as a part of primary key:

TelephoneNumber→Name, Location, StaffID

Normalized table:

BLOODBANK

|  |  |  |  |
| --- | --- | --- | --- |
| BloodBankID | Name | Location | MangingStaffID |

BLOODBANKTEL

|  |  |
| --- | --- |
| BloodBankID | PhoneNumber |

1. **RECIPIENT table:**

* RecipientID as a part of primary key:

RecipientID→ FName, Lname, Gender, DOB, BloodGroup

* PhoneNumber as a part of primary key:-

PhoneNumber→FName, Lname, Gender, DOB, BloodGroup

* Address and Disease as a part of primary key doesn’t determine anything.

Normalized table:

RECIPIENT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RecipientID | FName | LName | Gender | DOB | BloodGroup |

RECIPIENTPHONE

|  |  |
| --- | --- |
| RecipientID | PhoneNumber |

  RECIPIENTADDRESS

|  |  |
| --- | --- |
| RecipientID | Address |

RECIPIENTMEDICALCONDITION

|  |  |
| --- | --- |
| RecipientID | Disease |

1. **HOSPITAL table:**

* HospitalID as a part of primay key:

HospitalID→HospitalName, Location

* PhoneNumber as a part of a primary key:

PhoneNumber →Name, Location

Normalized table:

HOSPITAL

|  |  |  |
| --- | --- | --- |
| HospitalID | HospitalName | Location |

HOSPITALTELEPHONE

|  |  |
| --- | --- |
| HospitalID | PhoneNumber |

1. **STAFF table:**

* StaffID as a part of primary key:

StaffID→ FName, Lname, Gender, DOB, Position, BloodBankID, Salary

* PhoneNumber as a part of primary key:

PhoneNumber→ FName, Lname, Gender, DOB, Position, BloodBankID, Salary

* Address as a part of primary key determines nothing.

Normalized form:

STAFF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| StaffID | FName | LName | Gender | DOB | Role | BloodBankID |
| Salary | |  |  |  |  |  |

STAFFPHONE

|  |  |
| --- | --- |
| StaffID | PhoneNumber |

STAFFADDRESS

|  |  |
| --- | --- |
| StaffID | Address |

**3.2.2.3  Third normal form (3NF)**

Our tables conform to the third normal form as there are no transitive dependencies between the primary key attribute and non-key attributes, and there are no partial functional dependencies among the non-key attributes. The table is in third normal form because:

* The table satisfies the second normal form (2NF) and
* Does not contain any transitive dependencies.

**3.2.3. Relational Schema with Referential Integrity after Normalization**

The normalized relation schema with all the foreign keys is:-

**DONOR**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DonorID | FName | LName | Gender | DOB | BloodGroup | Age |

**DONORPHONE**

|  |  |
| --- | --- |
| DonorID | PhoneNumber |

**DONORADDRESS**

|  |  |
| --- | --- |
| DonorID | Address |

**DONATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DonorID | BloodBankID | StaffID | Date | AmountDonated |

**BLOODBANK**

|  |  |  |  |
| --- | --- | --- | --- |
| BloodBankID | Name | Location | MangingStaffID |

**BLOODBANKTEL**

|  |  |
| --- | --- |
| BloodBankID | PhoneNumber |

**BLOODQUANTITY**

|  |  |  |
| --- | --- | --- |
| BloodBankID | BloodGroup | BloodAmount |

**TRANSFUSION**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DonorID | BloodBankID | RecipientID | HospitalID | Date | Time | AmountTransferred |

**RECIPIENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RecipientID | FName | LName | Gender | DOB | BloodGroup |

**RECIPIENTPHONE**

|  |  |
| --- | --- |
| RecipientID | PhoneNumber |

**RECIPIENTADDRESS**

|  |  |
| --- | --- |
| RecipientID | Address |

RECIPIENTMEDICALCONDITION

|  |  |
| --- | --- |
| RecipientID | Disease |

HOSPITAL

|  |  |  |
| --- | --- | --- |
| HospitalID | HospitalName | Location |

HOSPITALTELEPHONE

|  |  |
| --- | --- |
| HospitalID | PhoneNumber |

STAFF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| StaffID | FName | LName | Gender | DOB | Position | BloodBankID |
| Salary | |  |  |  |  |  |

STAFFPHONE

|  |  |
| --- | --- |
| StaffID | PhoneNumber |

STAFFADDRESS

|  |  |
| --- | --- |
| StaffID | Address |

### 

### **3.3Physical database design of the new** **system**

### **3.3.1Physical design strategy**

In this section, there are data type and size specifications for our logical database design. These will be used later in implementation section.

1. **DONOR table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_DonorID | Fixed length character string length 7 |
| domain\_FName | Variable length character string maximum length 30 |
| domain\_LName | Variable length character string maximum length 30 |
| domain\_Gender | Fixed length character string length 1 |
| domain\_DOB | Date composed of day, month and year |
| domain\_Age | Variable length character maximum length 3 |
| domain\_BloodGroup | Variable length character string maximum length 3 |

1. **DONORPHONE table:**

|  |  |
| --- | --- |
| Attribute name | Data Type and Size |
| Domain\_ DonorID | Fixed length character string length 7 |
| domain\_PhoneNumber | Variable length character string maximum length 15 |

1. **DONORADDRESS table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| Domain\_DonorID | Fixed length character string length 7 |
| domain\_Address | Variable length character string maximum length 100 |

1. **DONATION table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_ BloodBankID | Fixed length character string length 3 |
| domain\_ StaffID | Fixed length character string length 3 |
| domain\_ Date | Date composed of day, month and year |
| domain\_DonorId | Fixed length character string length 7 |
| Domain\_AmountDonated | Floating point number |

1. **BLOODBANK table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_ BloodBankID | Fixed length character string length 3 |
| domain\_ Name | Fixed length character string length 60 |
| domain\_ Location | Variable length character string maximum length 100 |
| domain\_ManagingStaffID | Fixed length character string length 3 |

1. **BLOODBANKPHONE table**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_BloodBankID | Fixed length character string length 3 |
| domain\_PhoneNumber | Variable length character string maximum length 15 |

1. **BLOODQUANTITY table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_ BloodBankID | Fixed length character string length 3 |
| domain\_BloodGroup | Fixed length character string length 3 |
| domain\_ BloodAmount | Floating point number |
| domain\_Date(Collection and Expiration) | Date composed of day, month and year |

1. **TRANSFUSION table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_DonorID | Fixed length character string length 7 |
| domain\_ BloodBankID | Fixed length character string length 3 |
| domain\_ RecipientID | Fixed length character string length 7 |
| domain\_ HospitalID | Fixed length character string length 5 |
| domain\_ Date | Date composed of day, month and year |
| domain\_BloodAmount | Floating point number |
| Domain\_ Time | Time composed of smaller time units |

1. **RECIPIENT table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_ RecipientID | Fixed length character string length 7 |
| domain\_FirstName | Variable length character string maximum length 30 |
| domain\_LName | Variable length character string maximum length 30 |
| domain\_Gender | Fixed length character string length 1 |
| domain\_BloodGroup | Fixed length character string length 3 |
| domain\_Date | Date composed of day, month and year |

1. **RECIPIENTPHONE table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| Domain\_RecipientID | Fixed length character string length 7 |
| domain\_ PhoneNumber | Variable length character maximum length 15 |

1. **RECIPIENTADRESS table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| Domain\_RecipientID | Fixed length character string length 7 |
| Domain\_ Address | Variable length character string maximum length 100 |

1. **RECIPIENTMEDICALCONDITION table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_RecipientID | Fixed length character string length 7 |
| domain\_Disease | Variable length character string maximum length 50 |

1. **.HOSPITAL table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_ HospitalID | Fixed length character string length 5 |
| domain\_ HospitalName | Fixed length character string length 60 |
| domain\_ Location | Variable length character string maximum length 100 |

1. **HOSPITALPHONE table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_ HospitalID | Fixed length character string length 5 |
| domain\_ PhoneNumber | Fixed length character string length 13 |

1. **STAFF table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_ StaffID | Fixed length character string length 3 |
| domain\_FName | Variable length character string maximum length 30 |
| domain\_LName | Variable length character string maximum length 30 |
| domain\_Position | Variable length character string maximum length 30 |
| domain\_Gender | Fixed length character string length 1 |
| domain\_DOB | Date composed of day, month and year |
| domain\_ Address | Variable length character string maximum length100 |
| domain\_ BloodBankID | Fixed length character string length 3 |
| domain\_Salary | Money like floating point number |

1. **STAFFPHONE table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_StaffID | Fixed length character string length 3 |
| domain\_PhoneNumber | Fixed length character string length 13 |

1. **STAFFADDRESS table:**

|  |  |
| --- | --- |
| Attribute name | Data type and size |
| domain\_StaffID | Fixed length character string length 3 |
| domain\_Address | Variable length character string maximum length 100 |

### **3.3.2 Hardware implementation**

Provided that the administration server and SQL server are on different devices, when the network includes 100,000 devices, the following tables show the minimum hardware prerequisites: -

**Configuration of the device that has administration server installed:**

|  |  |
| --- | --- |
| Hardware | Value |
| CPU | 8cores, 2.13GHz |
| RAM | 8GB |
| Hard drive | 1TB, with RAID |
| Network adapter | 1Gbit |

**Configuration of the device with SQL server installed:**

|  |  |
| --- | --- |
| Hardware | Value |
| CPU | 8cores, 2.53GHz |
| RAM | 26GB |
| Hard drive | 500GB, SATA RAID |
| Network adapter | 1Gbit |

### **3.3.3 Front end design outline**

The database has multiple users, and each user will have a distinct degree of authorization.

* The highest privilege to access a variety of information about the blood bank, including details about donors, recipients, the hospital, the blood bank, and staff, will be granted to the managers. In accordance with company rules, he or she may change certain attributes.
* Depending on their position within the blood bank, staff members will have the ability to insert and update data about themselves. Additionally, records of the donors and transfusions should be kept.
* The hospital manager will be able to request blood transfusions and insert and update hospital-related data.

**Chapter 4 Implementation and Testing**

**Implementation**

CREATE DATABASE BLOOD\_DONATION\_MANAGEMENT\_SYSTEM

USE BLOOD\_DONATION\_MANAGEMENT\_SYSTEM

--CREATING TABLES

CREATE TABLE DONOR(

DONORID CHAR(7) PRIMARY KEY,

FNAME VARCHAR(30) NOT NULL,

LNAME VARCHAR(30) NOT NULL,

GENDER CHAR(1) NOT NULL,

DOB DATE NOT NULL,

AGE varchar(3) NOT NULL,

BLOODGROUP CHAR(3) NOT NULL

)

CREATE TABLE DONORPHONE(

DONORID CHAR(7) NOT NULL FOREIGN KEY REFERENCES DONOR(DONORID),

PHONENUMBER CHAR(15) NOT NULL,

PRIMARY KEY(DONORID,PHONENUMBER))

CREATE TABLE DONORADDRESS(

DONORID CHAR(7) NOT NULL FOREIGN KEY REFERENCES DONOR(DONORID),

ADDRESS VARCHAR(100) NOT NULL,

PRIMARY KEY(DONORID, ADDRESS))

CREATE TABLE RECIPIENT(

RECIPIENTID CHAR(7) PRIMARY KEY,

FNAME VARCHAR(30) NOT NULL,

LNAME VARCHAR(30) NOT NULL,

GENDER CHAR(1) NOT NULL,

DOB DATE NOT NULL,

BLOODGROUP CHAR(3) NOT NULL

)

CREATE TABLE HOSPITAL(

HOSPITALID CHAR(5) PRIMARY KEY,

NAME CHAR(60) NOT NULL,

LOCATION VARCHAR(100) NOT NULL)

CREATE TABLE BLOODBANK(

BLOODBANKID CHAR(3) PRIMARY KEY,

NAME CHAR(60) NOT NULL,

LOCATION VARCHAR(100) NOT NULL

)

CREATE TABLE STAFF(

STAFFID CHAR(3) PRIMARY KEY,

FNAME VARCHAR(30) NOT NULL,

LNAME VARCHAR(30) NOT NULL,

POSITION VARCHAR(30) NOT NULL,

GENDER CHAR(1) NOT NULL,

DOB DATE NOT NULL,

ROLE CHAR(50) NOT NULL,

SALARY MONEY NOT NULL,

BLOODBANKID CHAR(3) FOREIGN KEY REFERENCES BLOODBANK(BLOODBANKID))

ALTER TABLE BLOODBANK ADD MANGSTAFFID CHAR(3)

ALTER TABLE BLOODBANK ADD CONSTRAINT FK1 FOREIGN KEY(MANGSTAFFID) REFERENCES STAFF(STAFFID)

CREATE TABLE DONATION(

DONORID CHAR(7) NOT NULL FOREIGN KEY REFERENCES DONOR(DONORID),

BLOODBANKID CHAR(3) NOT NULL FOREIGN KEY REFERENCES BLOODBANK(BLOODBANKID),

STAFFID CHAR(3) NOT NULL FOREIGN KEY REFERENCES STAFF(STAFFID),

DATE DATE NOT NULL,

AMOUNTDONATED FLOAT NOT NULL,

PRIMARY KEY(DONORID,BLOODBANKID,STAFFID,DATE,AMOUNTDONATED)

)

CREATE TABLE BLOODBANKPHONE(

BLOODBANKID CHAR(3) NOT NULL FOREIGN KEY REFERENCES BLOODBANK(BLOODBANKID),

PHONENUMBER CHAR(15) NOT NULL,

PRIMARY KEY(BLOODBANKID,PHONENUMBER))

CREATE TABLE BLOODQUANTITY(

BLOODBANKID CHAR(3) NOT NULL FOREIGN KEY REFERENCES BLOODBANK(BLOODBANKID),

BLOODGROUP CHAR(3) NOT NULL,

BLOADAMOUNT FLOAT NOT NULL,

DATE DATE NOT NULL,

PRIMARY KEY(BLOODBANKID,BLOADAMOUNT,DATE))

ALTER TABLE BLOODQUANTITY

ADD expire\_date date ;

ALTER TABLE BlOODQUANTITY

ADD collection\_date date;

CREATE TABLE TRANSFUSION (

DONORID CHAR(7) NOT NULL FOREIGN KEY REFERENCES DONOR(DONORID),

BLOODBANKID CHAR(3) NOT NULL FOREIGN KEY REFERENCES BLOODBANK(BLOODBANKID),

RECIPIENTID CHAR(7) NOT NULL FOREIGN KEY REFERENCES RECIPIENT(RECIPIENTID),

HOSPITALID CHAR(5) NOT NULL FOREIGN KEY REFERENCES HOSPITAL(HOSPITALID),

DATE DATE NOT NULL,

TIME TIME NOT NULL,

AMOUNTTRANSFERRED FLOAT NOT NULL,

PRIMARY KEY(DONORID,BLOODBANKID,RECIPIENTID,HOSPITALID,DATE,TIME))

CREATE TABLE RECIPIENTPHONE(

RECIPIENTID CHAR(7) NOT NULL FOREIGN KEY REFERENCES RECIPIENT(RECIPIENTID),

PHONENUMBER CHAR(15) NOT NULL,

PRIMARY KEY(RECIPIENTID,PHONENUMBER))

CREATE TABLE RECIPIENTADDRESS(

RECIPIENTID CHAR(7) NOT NULL FOREIGN KEY REFERENCES RECIPIENT(RECIPIENTID),

ADDRESS VARCHAR(100) NOT NULL,

PRIMARY KEY(RECIPIENTID, ADDRESS))

CREATE TABLE RECIPIENTMEDICALCONDITION(

RECIPIENTID CHAR(7) NOT NULL FOREIGN KEY REFERENCES RECIPIENT(RECIPIENTID),

MEDICALHISTORY VARCHAR(50) NOT NULL,

PRIMARY KEY(RECIPIENTID,MEDICALHISTORY))

CREATE TABLE HOSPITALTELEPHONE(

HOSPITALID CHAR(5) NOT NULL FOREIGN KEY REFERENCES HOSPITAL(HOSPITALID),

PHONENUMBER CHAR(13) NOT NULL,

PRIMARY KEY(HOSPITALID,PHONENUMBER))

CREATE TABLE STAFFPHONE(

STAFFID CHAR(3) NOT NULL FOREIGN KEY REFERENCES STAFF(STAFFID),

PHONENUMBER CHAR(13) NOT NULL,

PRIMARY KEY(STAFFID,PHONENUMBER))

CREATE TABLE STAFFADDRESS(

STAFFID CHAR(3) NOT NULL FOREIGN KEY REFERENCES STAFF(STAFFID),

ADDRESS VARCHAR(100) NOT NULL,

PRIMARY KEY(STAFFID, ADDRESS))

--INSERT STATEMENTS

INSERT INTO DONOR

VALUES('D000001','Dawit','Girma','M','2002-10-25','54','O+')

INSERT INTO BLOODBANK(BLOODBANKID,NAME,LOCATION)

VALUES('B01','BLOODBANK1','Addis Ababa')

INSERT INTO STAFF

VALUES('S01','Adoniyas','Bekele','Assistant','M','2002-08-20','BLOOD BANK MEDICAL LAB ASSISTANT','20,000','B01')

UPDATE BLOODBANKPHONE

SET PHONENUMBER='0944375911' WHERE BLOODBANKID='B01'

INSERT INTO BLOODBANK

VALUES('B01','+251119080808')

INSERT INTO BLOODUNTITY

VALUES ('B01','O+',1000.00,'2023-03-01')

INSERT INTO DONATION

VALUES('D000001','B01','S01','2023-03-01',500.00)

INSERT INTO DONORADDRESS

VALUES('D000001','Addis Ababa')

INSERT INTO DONORPHONE

VALUES('D000001','+251970707070')

INSERT INTO HOSPITAL

VALUES('H0001','TIKUR ANBESA HOSPITAL','Addis Ababa')

INSERT INTO HOSPITALTELEPHONE

VALUES('H0001','+25131401281')

INSERT INTO RECIPIENT

VALUES('R000001','Fasika', 'Ewnetu', 'F', '2001-08-12','AB+')

select \* from recipient

INSERT INTO RECIPIENTADDRESS

VALUES('R000001','Kolfe Keraniyo')

INSERT INTO RECIPIENTMEDICALCONDITION

VALUES('R000001','Hemophilia')

INSERT INTO RECIPIENTPHONE

VALUES('R000001','+2519080899')

INSERT INTO STAFFADDRESS

VALUES('S01','Kirkos')

INSERT INTO STAFFPHONE

VALUES('S01','+251901010101')

INSERT INTO TRANSFUSION

VALUES('D000001','B01','R000001','H0001','2023-03-01','02:20',500)

UPDATE DONOR

SET FNAME='Temesgen', LNAME=' Gonfa', GENDER='M', DOB='2001-02-27', BLOODGROUP='AB-'

WHERE DONORID='D000001'

--VIEW AND PRIVILEGE STATEMENTS

CREATE VIEW DONOR\_INFORMATION AS

SELECT DONOR.DONORID,FNAME,LNAME,BLOODGROUP,DOB,PHONENUMBER

FROM DONOR INNER JOIN DONORPHONE

ON DONOR.DONORID=DONORPHONE.DONORID

SELECT \*FROM DONOR\_INFORMATION

CREATE VIEW RECIPIENT\_INFORMATION AS

SELECT RECIPIENT.RECIPIENTID,FNAME,LNAME,BLOODGROUP,DOB

FROM RECIPIENT INNER JOIN RECIPIENTMEDICALCONDITION

ON RECIPIENT.RECIPIENTID=RECIPIENTMEDICALCONDITION.RECIPIENTID

SELECT\*FROM RECIPIENT\_INFORMATION

CREATE VIEW HOSPITAL\_INFORMATION AS

SELECT HOSPITAL.HOSPITALID,NAME,LOCATION,PHONENUMBER

FROM HOSPITAL INNER JOIN HOSPITALTELEPHONE

ON HOSPITAL.HOSPITALID=HOSPITALTELEPHONE.HOSPITALID

SELECT\*FROM HOSPITAL\_INFORMATION

CREATE VIEW STAFF\_INFORMATION AS

SELECT STAFF.STAFFID,FNAME,LNAME,ROLE,DOB,BLOODBANKID,SALARY,PHONENUMBER

FROM STAFF INNER JOIN STAFFPHONE

ON STAFF.STAFFID=STAFFPHONE.STAFFID

SELECT\*FROM STAFF\_INFORMATION

CREATE VIEW MANG\_PRIVILEGE AS

SELECT DONOR\_INFORMATION.DONORID,FNAME,LNAME,BLOODGROUP,DOB,PHONENUMBER,ADDRESS

FROM DONOR\_INFORMATION INNER JOIN DONORADDRESS

ON DONOR\_INFORMATION.DONORID=DONORADDRESS.DONORID

SELECT \* FROM MANG\_PRIVILEGE

--MANAGER'S PRIVILEGE ON UPDATE

UPDATE STAFF

SET SALARY='10,000'

WHERE STAFFID='S01'

CREATE VIEW MANG\_PRIVILEGE2 AS

SELECT STAFF\_INFORMATION.STAFFID,FNAME,LNAME,ROLE,DOB,BLOODBANKID,SALARY,PHONENUMBER,ADDRESS

FROM STAFF\_INFORMATION INNER JOIN STAFFADDRESS

ON STAFF\_INFORMATION.STAFFID=STAFFADDRESS.STAFFID

SELECT \*FROM MANG\_PRIVILEGE2

ALTER TABLE donor

ADD CONSTRAINT check\_age CHECK (age > 18);

# **ANNEXES**

As for the SQL scripts, they are all available in the implementation section. The group collaboration statement is presented below.

## Group collaboration statement (**Bezawit Lulekal, Hewan Zewdie, Mariamawit Getachew, Mihret Daniel, Ribka Muluye, Setota Girma)** have been working together towards the accomplishment of this project. Starting from selection of title on consensus, Ribka Muluye(and also the ER table to mapping which is the logical database design) worked towards identification and screening of entities, attributes and relationships which are essential for the project. The ER diagramming was delegated primarily for Hewan Zewdie(and also the database development methodology). After the ER diagramming, it has been reviewed by all members to see all the important components have not been forgotten. The normalization part was left to Mihret Daniel and Bezawit Lulekal who have been checking and modifying the mapped design to minimize data duplication using various normal forms. The second normal form was the one required in this project since it was a relation and no transitive dependency existed. The physical desighn strategy was done by Stota Girma and Mariamawit Getachew who worked the front end outline design and some parts of the implementation. The final and implementation section was done by a combined effort of Mihret Daniel who had been writing the code in SSMS to try if the system works. The system worked as expected and this was the story behind the success of the project.

**References**

Google,Reference book Data base systems a practical approach to desighn, implementation, and management.