 **BLOOD BANK DATABASE SYSTEM**

**FINAL REPORT**

**Abstract**

This paper contains a list of system specifications, a database architecture, a basic definition of the methods, and an overview of how to incorporate a blood donation management system. The aim of this project is to develop a database management framework capable of enabling the stress-free management and coordination of a blood donor database that can be quickly incorporated in NGOs.

**Made By**

Diya Sachdev Ayush Varma

1910110140 1910110108

Database Management Systems

Shiv Nadar University

Computer Science, B. Tech

**SUMMARY**

The importance and need of blood at the correct time cannot be overstated at such a period (Covid-19). The basic aim of this project was to make a database with capabilities to be useful in such times. It revolves around three types of users- donor, recipient and blood bank administrators.

**GENERAL DESCRIPTION**

The blood donation system we aim to build with the implementation of our database would aim to facilitate the easy and efficient connection of blood donors and recipients to blood-banks. The system would register users in three groups namely, a donor, recipient and a blood bank. Our users will be classified into persons (donor and recipient) and organizations (blood-banks). We would record static locations of all users and aim to provide the most appropriate solution, taking into account distance. Persons will have their medical history and blood group recorded and made available to aid the efficiency of the system. Organizations will have their activity history recorded for accountability and references.

**REQUIREMENT ANALYSIS**

An efficient blood bank database would aim to easily connect potential donors with blood-banks, blood-banks with potential recipients and vice versa. The potential donors would be connected with their nearest blood bank. The data of potential donors will be stored according to their blood types, disease history and proximity to nearest blood-bank in the database.

Another Objective of the database is to register and store blood-banks by their location in order for them to be connected to potential donors and recipient with ease. The blood banks aim to receive and give out blood from people in need (recipients) closest to them. The blood banks record the instances of blood received according to the blood group received, location and past illnesses associated with donor. The banks also record the instances of blood given out according to similar criteria, replacing past illnesses with reason for requesting blood.

The blood recipient would need to be connected to a bank which possesses his blood in the closest proximity to him. The recipient would provide details of the blood type requested, reason for request and location.

**Entities:**

1. User
2. Blood\_Bank
3. Person
4. Donor
5. Recipient
6. Medical\_Info
7. History\_Of\_Donations
8. Blood\_Type

The **User** table is a super class for **Blood\_Bank** and **Person**. They share the same basic attributes with **Name** attribute and **First\_Name, Last\_Name and Blood\_Id** attributes being added to the **Blood\_Bank** and **Person** respectively.

**Donor** and **Recipient** are both sub-classes of **person** that share the same attributes and taking the **User\_Id** as their Donor or Recipient Ids from the **User** where Ids will be generated automatically by the system according to the status of the User; Donor, Recipient or Blood bank.

**Medical\_Info** table will be made to control and approve whether the donor will be able to donate blood or not according to his medical case. Therefore, Medical\_Info table will store the medical reports of the donors and the date of them.

**History\_Of\_Donations** table will store the previous donations of donors so that the system will not accept them to donate any blood earlier than **56 days** from the previous donation.

**Blood\_Type** table is made to monitor the blood types with some attributes to help automate the system for donation to make it easier and faster. The attributes will show which type can donate to which and vice versa. **Blood\_Id** being the primary key of this table, it will be a critical attribute in the other tables to manage the process of donation and make it simpler.

**Database Schema:**

1. **ADDRESS:** This table where the main addresses are saved so that only the Address\_Id is used in the user table to avoid redundancy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Address\_Id** | **City** | **District** | **Neighborhood** |
| **TYPE** | numeric | varchar(20) | varchar(20) | varchar(60) |
| **KEY** | PK |  |  |  |
| **EXAMPLE** | 1000000 | Dwarka | South Delhi | Janakpuri |

*Table 1: ADDRESS TABLE*

1. **STATUS:** This table defines the users and categorizes them into the main 3 categories; blood donor, blood recipient and blood bank.

|  |  |  |
| --- | --- | --- |
|  | **Status\_Id** | **Status** |
| **TYPE** | numeric | varchar(20) |
| **KEY** | PK |  |
| **EXAMPLE** | 2 | Donor |

***Table 2:*** *STATUS TABLE*

1. **USER:** This table saves the user’s personal data in the database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **User\_Id** | **Status\_Id** | **Address\_Id** | **Phone\_No** | **Password** |
| **TYPE** | numeric | numeric | numeric | Varchar(10) | Varchar(45) |
| **KEY** | PK | FK | FK |  |  |
| **EXAMPLE** | 4 | 2 | 1000000 | 5539190967 | 123 |

***Table 3:*** *USER TABLE*

1. **BLOOD\_TYPE:** This table defines and saves all the blood groups with their donation and receipt features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Blood\_Id** | **Blood\_Code** | **Donates\_to** | **Receives\_from** |
| **TYPE** | numeric | Varchar(5) | Varchar(45) | Varchar(45) |
| **KEY** | PK |  |  |  |
| **EXAMPLE** | 4 | O | All | O |

***Table 4:*** *BLOOD\_GROUP TABLE*

1. **DONOR:** This table saves the data of donors who registered to the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Donor\_Id** | **First\_Name** | **Last\_Name** | **Blood\_Id** |
| **TYPE** | numeric | Varchar(5) | Varchar(45) | Varchar(45) |
| **KEY** | PK,FK |  |  | FK |
| **EXAMPLE** | 4 | Ayush | Varma | 4 |

***Table 5:*** *DONOR TABLE*

1. **MEDICAL\_INFO:** This table saves the medical reports that state whether donors can donate blood or not.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Report\_Id** | **Donor\_Id** | **Date** | **Result** |
| **TYPE** | numeric | numeric | date | Varchar(45) |
| **KEY** | PK | FK |  |  |
| **EXAMPLE** | 150100 | 4 | 2020-05-02 | No disease |

***Table 6:*** *MEDICAL\_INFO TABLE*

1. **RECIPIENT:** This table saves the data of those recipients who registered to the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Recipient\_Id** | **First\_Name** | **Last\_Name** | **Blood\_Id** |
| **TYPE** | numeric | Varchar(5) | Varchar(45) | Varchar(45) |
| **KEY** | PK,FK |  |  | FK |
| **EXAMPLE** | 3 | Diya | Sachdev | 1 |

***Table 7:*** *RECIPIENT TABLE*

1. **BLOOD\_BANK:** This table saves the data of blood banks which are registered to the system.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Bank\_Id** | **Name** | **Capacity** |
| **TYPE** | numeric | Varchar(5) | numeric |
| **KEY** | PK,FK |  |  |
| **EXAMPLE** | 1 | White Cross, Pitampura | 1000 |

***Table 8:*** *BLOOD\_BANK TABLE*

1. **GIVES\_TO:** This table saves the blood donations information of donors in blood banks.

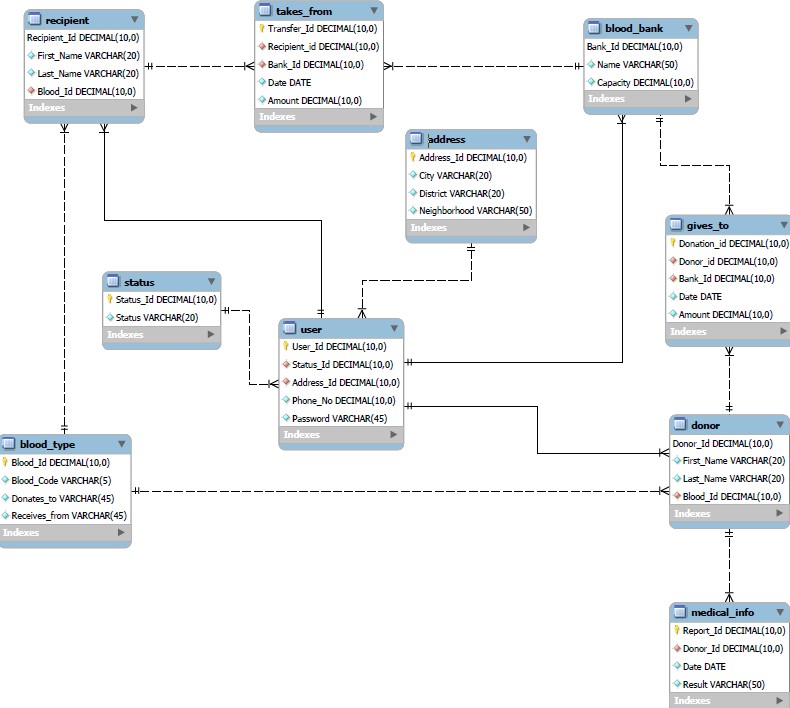
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Donation\_Id** | **Donor\_Id** | **Bank\_Id** | **Date** | **Amount** |
| **TYPE** | numeric | numeric | numeric | date | numeric |
| **KEY** | PK | FK | FK |  |  |
| **EXAMPLE** | 990001 | 4 | 1 | 2/19/2020 | 700 |

***Table 9:*** *GIVES\_TO TABLE*

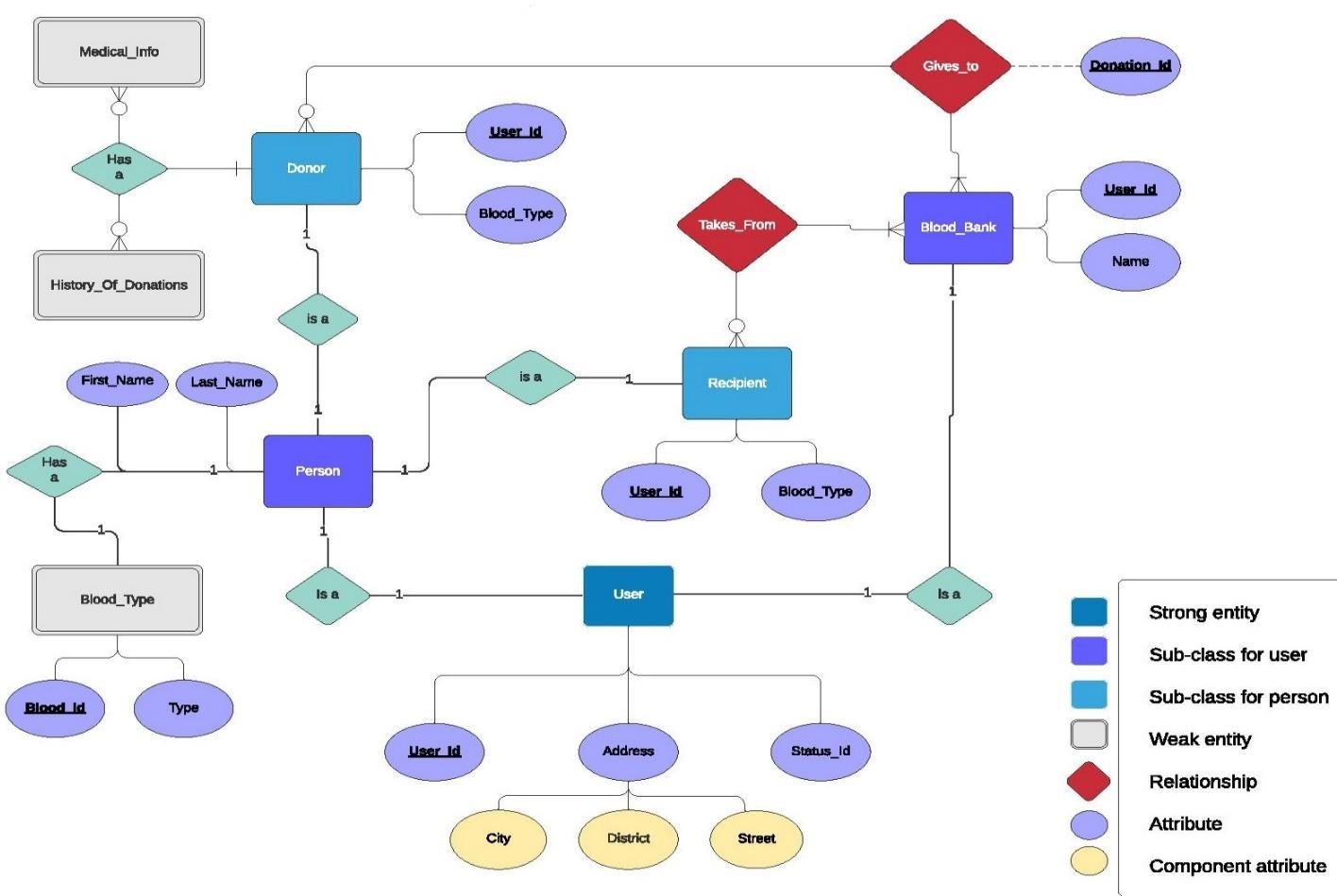
1. **TAKES\_FROM:** This table saves the blood donations information of donors in blood banks.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Transfer\_Id** | **Recipient\_Id** | **Bank\_Id** | **Date** | **Amount** |
| **TYPE** | numeric | numeric | numeric | date | numeric |
| **KEY** | PK | FK | FK |  |  |
| **EXAMPLE** | 770001 | 3 | 1 | 3/12/2020 | 450 |

***Table 10:*** *TAKES\_FROM TABLE*



**E-R DIAGRAM**



***Figure 3:*** *Entity-Relationship diagram*

**Features Implemented:**

We have implemented the functionality of Multi-User Environment support. It is divided into Admin, Donor and Recipient respectively. Using Referential Integrity and its various concept helps us maintain the data consistency of the database. Implementation of DELETE ON CASCADE and many more helps in reducing the redundancy of codes as well as the easy maintenance of code.

The database is made to work irrespective of the platform and therefore it is hosted on a server rather than it being on the local devices of user for flexibility purposes. For this we have used REMOTE mysql servers.

**Limitations:**

We wanted to execute the category-wise amount of blood left, in each bank. This would give a better statistical information to the blood banks as well as the recipients arriving for the blood beforehand.

We had earlier planned to use triggers and cursors but couldn’t due to the time constraint. Also, the program is at a niche stage and we would love to scale the program for a better interface and support so that it can be usable for organisations with minimal efforts.

**Future Ideas:**

We will be implementing the medical history of the donors as well as the recipients so that any possible mishappening can be ruled out. Also, we will be formulating the full blood flow path in which details of the donor(takes\_from) and the recipient(gives\_to) will be implemented.