



2022 – 2023

ESKİŞEHİR TECHNICAL UNIVERSITY

COMPUTER ENGINEERING

DATABASE MANAGEMENT SYSTEMS

PROJECT: BLOOD DONATION SYSTEM

Developers: [Adar Sönmez](#),

[Yusuf Mert Ülgen](#)

Starting Date: 25.02.2022

Phase 1 End Date: 03.03.2022

Phase 2 End Date: 01.05.2022

Phase 3 End Date: 29.05.2022

GitHub Repository: <https://github.com/adarSonmez/blood-donation-system>

Table of Contents

PHASE 1	3
Requirements Analysis	3
Entity-Relationship (E/R) Diagram.....	4
PHASE 2	5
Changes We Made to Our ER diagram.....	5
Relational Schema Design and Normalization	6
Our “CREATE TABLE” Commands.....	7
PHASE 3 FINAL PHASE.....	8
Detailed description of our project through Phase1 to Phase3.....	8
Functionalities of Our Program.....	9
1. Pages that can be viewed without authorization	9
2. Login.....	10
3. Donate Blood	11
4. Order Blood.....	12
5. Manager Orders.....	13
6. Manage Account.....	14

PHASE 1

Requirements Analysis

Our site offers different login options for different user types. The users of our system are differentiated as receptionist, hospital, and system manager. All three user types have in common the properties of a unique user id, name, address, phone number, email, password, and user type. The Receptionist is responsible for registering the donors who come to donate blood to the blood bank. Each donor has a unique donor id, name, blood type, number of donations, and date of birth. For each donated blood, blood id, date of blood donation and date of deterioration of blood are stored. Blood can be stored in various blood banks. Each blood bank has a unique bank id, capacity, and address. The system manager is responsible for blood banks. Hospitals that need blood log in to the system and create blood orders. The desired blood type, status, amount of blood, order date and order id for these orders are stored. It is also the responsibility of the system managers to approve these orders.

More Functional Requirements

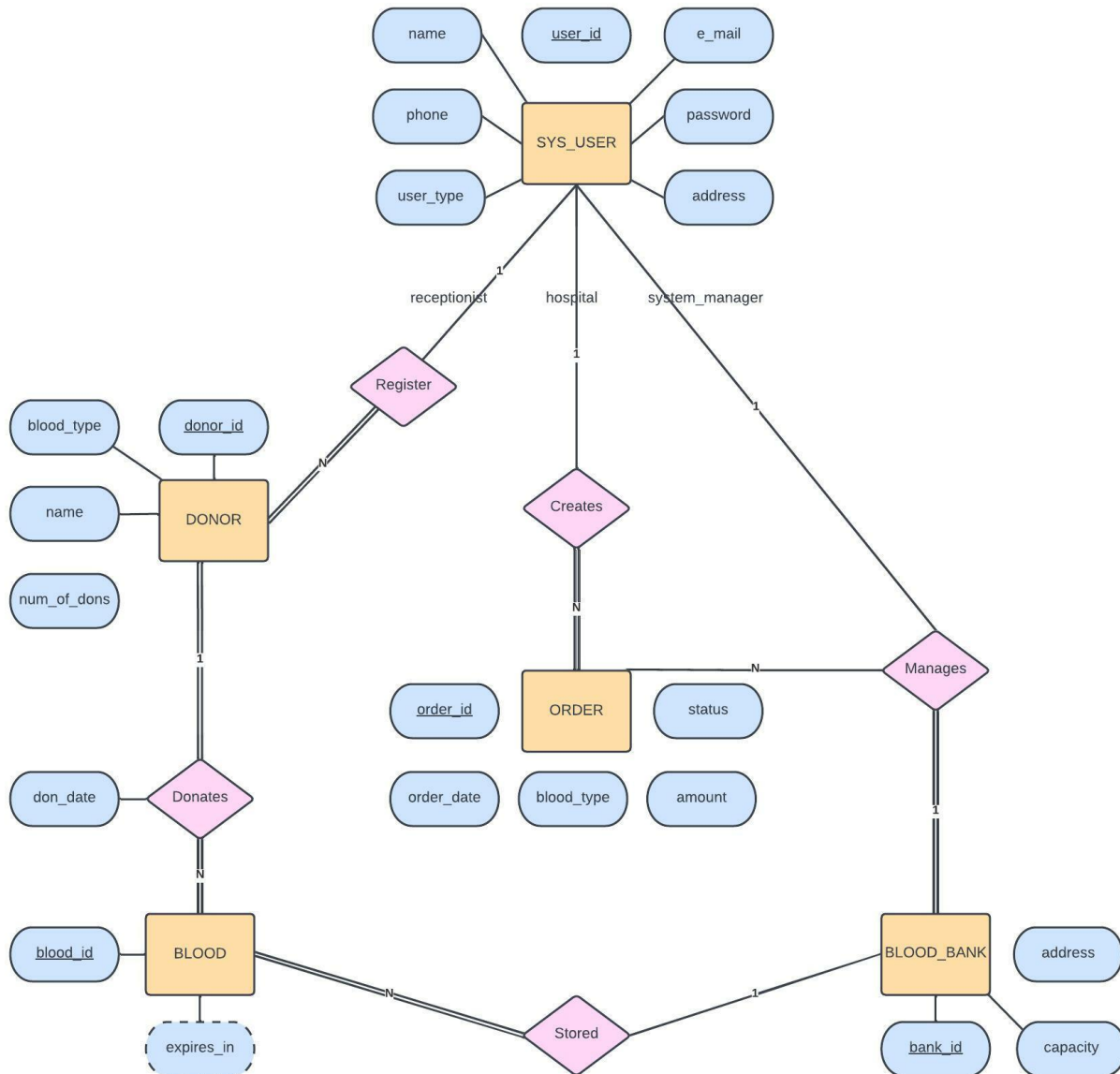
- When someone wants to donate blood, it should be checked (in program by query) whether there is enough room in the blood bank, and if there is, the type and amount of blood specified in the blood bank should be increased. If there is not enough room in the blood bank, blood should not be taken from the person.
- If a person donates a blood, then number of donations increases.
- When the system administrator approves a blood order if the specified amount and type of blood is available in the blood bank, status becomes approved, and the amount of this blood should be reduced. If there is not enough blood, the approval process should be canceled automatically, and status becomes rejected. By default, status is waiting.
- The blood closest to the deterioration date is given to the hospitals primarily.

Our Interesting Queries

- How many blood donations have been made so far?
- What is the ratio of the most donated blood type to the total donated blood type?
- What is the ratio of the least donated blood type to the total donated blood type?
- Who are the top 10 blood donors in 2022?
- Who is the receptionist who has registered the most donors so far?
- Which hospital orders the most blood?

Entity-Relationship (E/R) Diagram

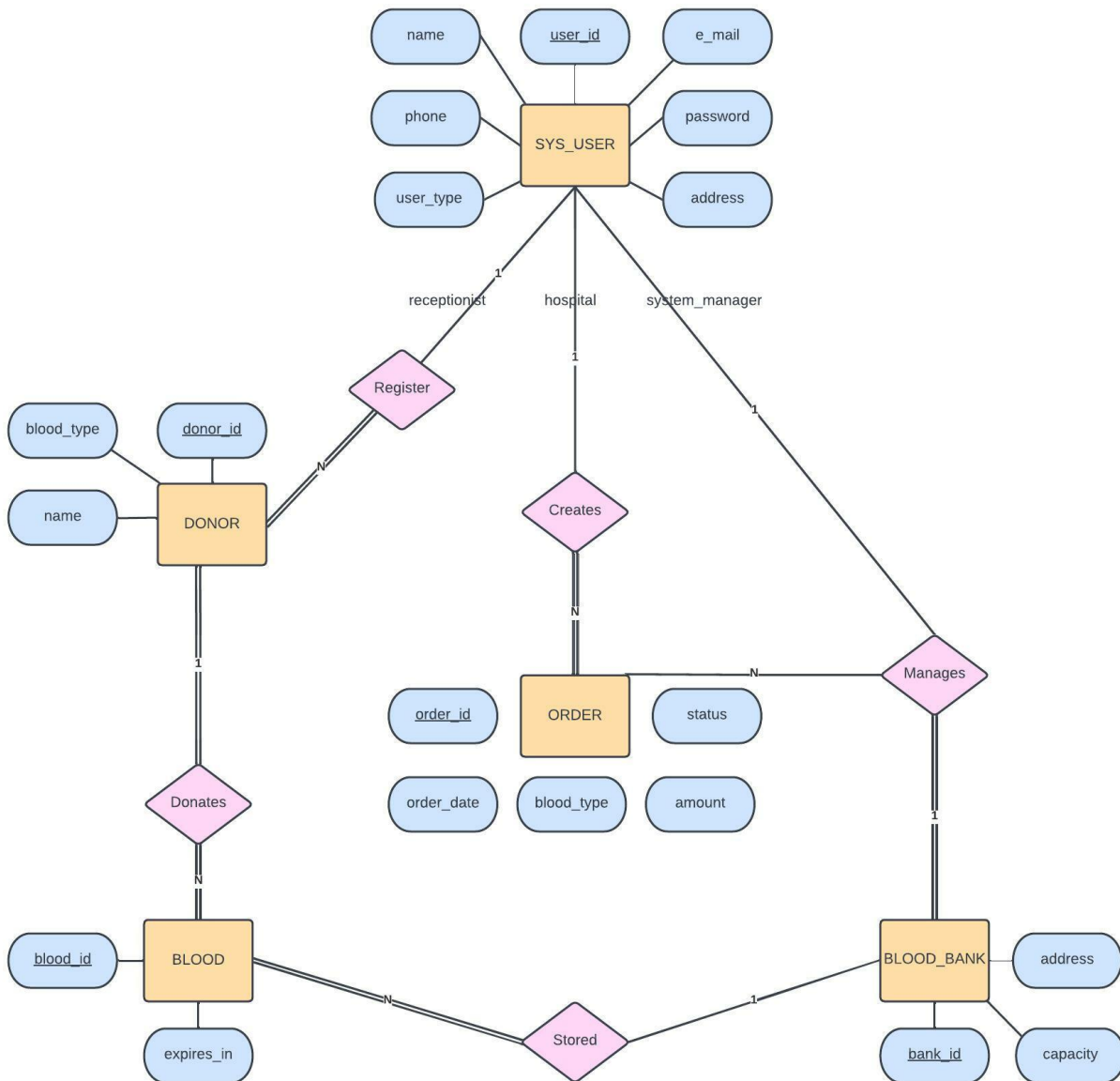
Instead of creating separate entity sets for different users (system manager, hospital, receptionist) in our E/R diagram, we combined them in our sys_user entity and provided differentiation with the user_type attribute. The pink diamonds in our diagram represent relations. Since most of our relationships are one-to-many, our preference would be to show them as columns of related tables rather than as separate tables in the relational database.



PHASE 2

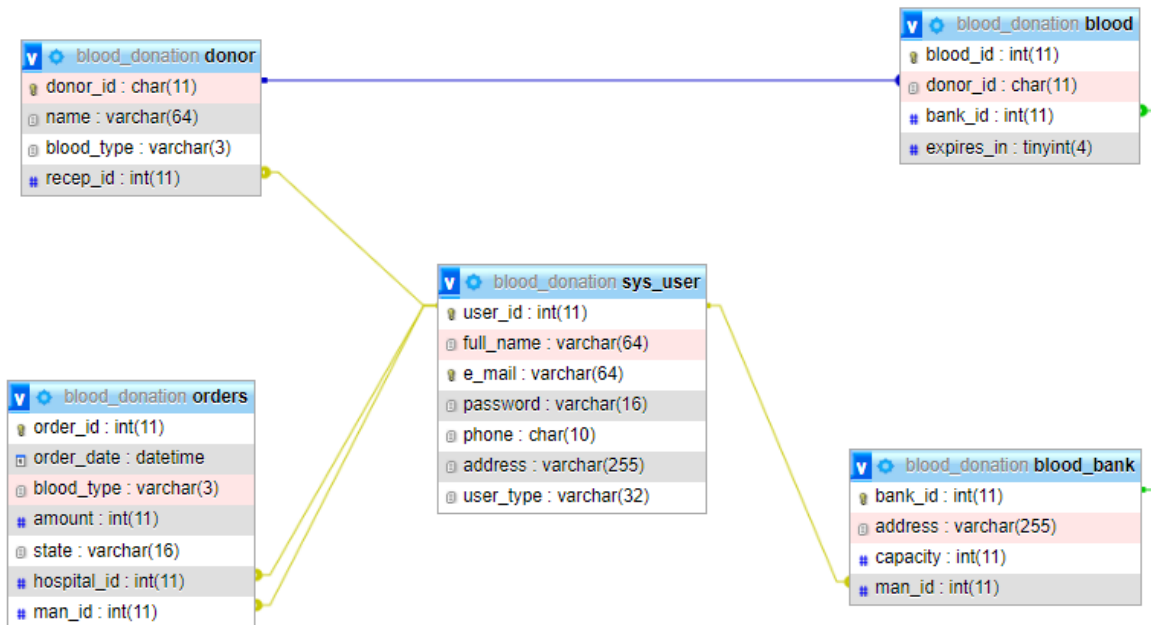
Changes We Made to Our ER diagram.

1. Removed num_of_dons attribute from DONOR entity set. Because it has been noticed that this value can be obtained from the BLOOD entity set with a SQL query.
2. The don_date attribute has been deleted from the DONATES relationship. Because we decided that the decay time of the blood is more important to us than the date of donation, and we kept expires_in attribute in the BLOOD entity set. By doing this, we have reduced the number of tables.



Relational Schema Design and Normalization

- 1. NF
 - a. Every table have a primary key
 - b. All our attributes have atomic data types
 - c. Each column keeps only one type of data there are no repeated columns. (User can have at most one phone number and address, so we do not break this rule.)
- 2. NF
 - a. There is no partial dependency in our tables. So, we are not violating the 2nd normal form.
- 3. NF
 - a. Since e-mail is unique, the e-mail column is dependent on all other user columns. In this case, we have a transitive dependency in our Sys_user table. So, we violate 3rd normal form. But we have done this on purpose. Because we wanted to collect all the information of the user in a single table. We could have made e-mail the primary key and made our table suitable for the 3rd normal form, but since e-mail is a data that can be modified, we did not want to make it a primary key.



Our “CREATE TABLE” Commands

```
CREATE TABLE IF NOT EXISTS sys_user (  
    user_id INT NOT NULL AUTO_INCREMENT,  
    full_name VARCHAR(64) NOT NULL,  
    e_mail VARCHAR(64) NOT NULL,  
    password VARCHAR(16) NOT NULL,  
    phone CHAR(10),  
    address VARCHAR(255),  
    user_type VARCHAR(32) NOT NULL,  
    PRIMARY KEY (user_id),  
    UNIQUE e_mail (e_mail)  
);  
  
CREATE TABLE IF NOT EXISTS donor (  
    donor_id CHAR(11) NOT NULL,  
    name VARCHAR(64) NOT NULL,  
    blood_type VARCHAR(3) NOT NULL,  
    recep_id INT NOT NULL,  
    FOREIGN KEY (recep_id) REFERENCES sys_user(user_id) ON DELETE CASCADE,  
    PRIMARY KEY (donor_id)  
);  
  
CREATE TABLE IF NOT EXISTS orders (  
    order_id INT NOT NULL AUTO_INCREMENT,  
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
    blood_type VARCHAR(3) NOT NULL,  
    amount INT NOT NULL,  
    state VARCHAR(16) DEFAULT "waiting",  
    hospital_id INT NOT NULL,  
    man_id INT NOT NULL,  
    FOREIGN KEY(man_id) REFERENCES sys_user(user_id) ON DELETE CASCADE,  
    FOREIGN KEY(hospital_id) REFERENCES sys_user(user_id) ON DELETE CASCADE,  
    PRIMARY KEY(order_id)  
);  
  
CREATE TABLE IF NOT EXISTS blood_bank (  
    bank_id INT NOT NULL AUTO_INCREMENT,  
    address VARCHAR(255) NOT NULL,  
    capacity INT DEFAULT 0,  
    man_id INT NOT NULL,  
    FOREIGN KEY(man_id) REFERENCES sys_user(user_id) ON DELETE CASCADE,  
    PRIMARY KEY(bank_id)  
);  
  
CREATE TABLE IF NOT EXISTS blood (  
    blood_id INT NOT NULL AUTO_INCREMENT,  
    donor_id CHAR(11) NOT NULL,  
    bank_id INT NOT NULL,  
    used TINYINT DEFAULT 0,  
    donated_date DATE DEFAULT CURRENT_DATE(),  
    FOREIGN KEY(bank_id) REFERENCES blood_bank(bank_id) ON DELETE CASCADE,  
    FOREIGN KEY(donor_id) REFERENCES donor(donor_id) ON DELETE CASCADE,  
    PRIMARY KEY(blood_id)  
);  
  
--
```

PHASE 3 | FINAL PHASE

Detailed description of our project through Phase1 to Phase3

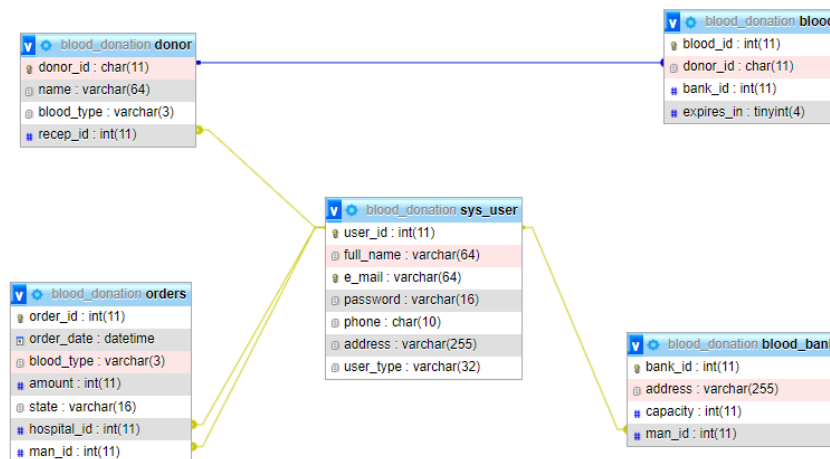
In Phase 1, we made the requirements list of our program. We have fulfilled all these requirements in the final phase. The requirements we fulfill are listed below.

- System administrators will be able to register new users to the system. These users can register as a hospital, receptionist, or system administrator.
- A different interface will be presented for each different user type.
- Hospitals can order blood.
- System administrators can have the authority to approve and reject blood orders of hospitals.
- Receptionists can register new donors to the system and register the information of donated blood to the system.

The interesting queries we identified in Phase 1 are shown below. All these queries and many more were successfully run in our final application.

- How many blood donations have been made so far?
- What is the ratio of the most donated blood type to the total donated blood type?
- What is the ratio of the least donated blood type to the total donated blood type?
- Who are the top 10 blood donors in 2022?
- Who is the receptionist who has registered the most donors so far?
- Which hospital orders the most blood?

Our database that we built in Phase 2 is shown below. There are 2 changes we made for phase 3 in this database. The first change we made was to add the donated_date attribute instead of the expires_in attribute in the blood chart. The reason for this is that we wanted to give the detection of blood deterioration to the application side, not to the database. The second change was to add the “used” attribute in the blood table, which determines whether the relevant blood is sent to a hospital.



Functionalities of Our Program

1. Pages that can be viewed without authorization

There are 4 routes that users can access without logging into our application. Our home page (Figure 1), which is the first of these, contains statistics about our blood donation system. When the "How to Donate" button is clicked on the homepage, a page listing the addresses of blood banks appears (Figure 2). When the "See Top Tens" button is clicked on the homepage, the tables with the first 10 donors who donated the most blood, the first 10 hospitals with the most blood donations, and the first 10 receptionists who registered the most donors will be shown on a new page. The last route we can access without logging in is our Login page. This page is explained on the next page in details.

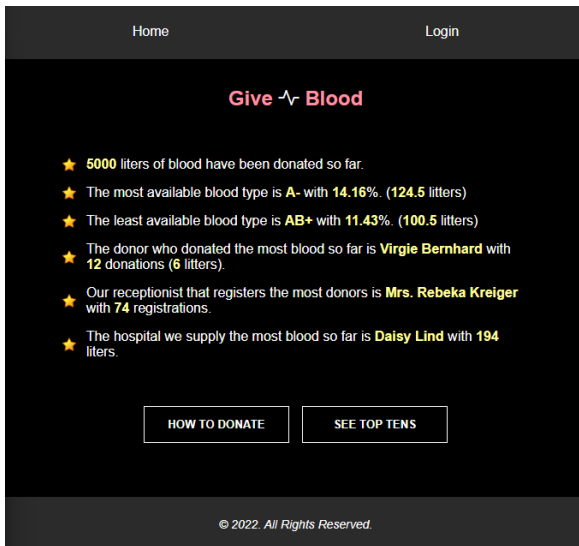


Figure 1

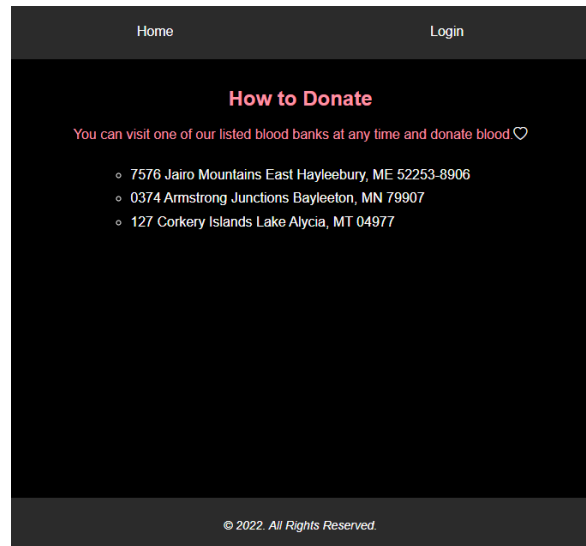


Figure 2

The 'Top Tens' page features a dark blue header with 'Home' and 'Login' links. The title 'Top Tens' is in red. Below it, three tables are displayed side-by-side. The footer contains the copyright notice '© 2022. All Rights Reserved.'.

Top Donors			Top Hospitals			Top Receptionist		
No.	Name	Donations	No.	Name	Orders (L)	No.	Name	Registrations
1	Virgie Bernhard	12	1	Daisy Lind	194	1	Mrs. Rebeka Kreiger	74
2	Hannah Welch	12	2	Zakary Jakubowski	168.5	2	Miss Brenda Cole	73
3	Webster Ruecker	11	3	Carmelo Gerhold	159	3	Orpha Koch	73
4	Garland Kuhn	10	4	Marco Cummings	146.5	4	Crstel Nikolaus	72
5	Shyann McClure	10	5	Lavon Kuhic	146	5	Kallie Jenkins	71
6	Floyd Dickens	10	6	Angeline Paucek Sr.	146	6	Prof. London Brown	71
7	Miss Trudie Cassin DDS	10	7	Lysanne Adams	131	7	Jennings Friesen V	71
8	Mr. Gaston Thiel DDS	10	8	Prof. Ethan Streich	128.5	8	Nicholas Lebsack II	70
9	Ms. Norene Goyette DVM	10	9	Hellen Shanahan	128	9	Dr. Lonnie Wiza	68
10	Hilario Conroy	10	10	Eugene Kerluke	126	10	Dr. Bianca Kris IV	68

Figure 3

2. Login

Only users who are registered to the system as a system manager and hospital or receptionist can log in to the system. The system manager does these registrations (it will be explained in detail later). Users who are not registered in the system can click on "contact us" and request a random system manager to register themselves via e-mail (Figure 5). Users registered in the system can log in to the system by entering their email and passwords (Figure 4). After login, the username of the user logged into the system is written in the upper middle part of the screen (Figure 6).

For convenience, some system users' modified login information is given below.

Manager Login -> mail: manager@example.com password: 123456

Hospital Login -> mail: hospital@example.com password: 123456

Receptionist Login -> mail: recep@example.com password: 123456

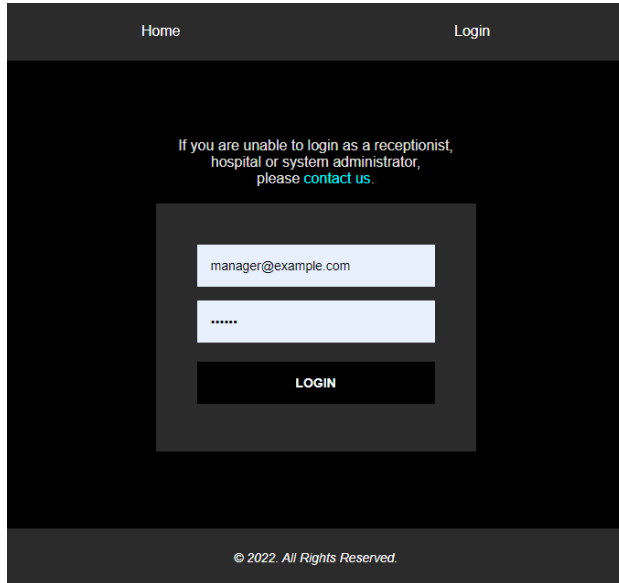


Figure 4

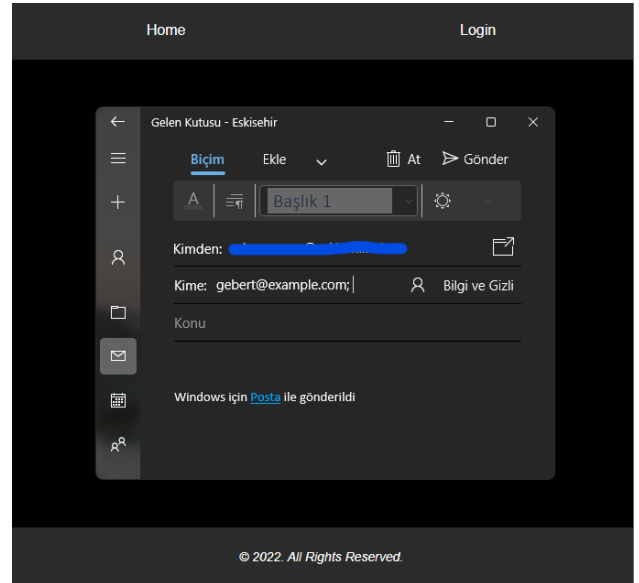


Figure 5

3. Donate Blood

The 2 routes that can be accessed by the user who logs into the system as a receptionist are shown in the upper navigation menu (Figure 6). While all user types logged into the system can access the "Manage Account" page, only the receptionist can access "Donate blood" page (Figure 7). On the Donate blood page, firstly, the ssn of the person to whom you want to donate blood is entered. If this person is registered in the system as a donor, the person's information is brought from database (Figure 8). If the donor is not registered in the system, the input fields where the donor's name and blood type will be entered become active to be filled. (Figure 9) After filling all the fields, the "Donate Blood" button is clicked, and the blood is donated.

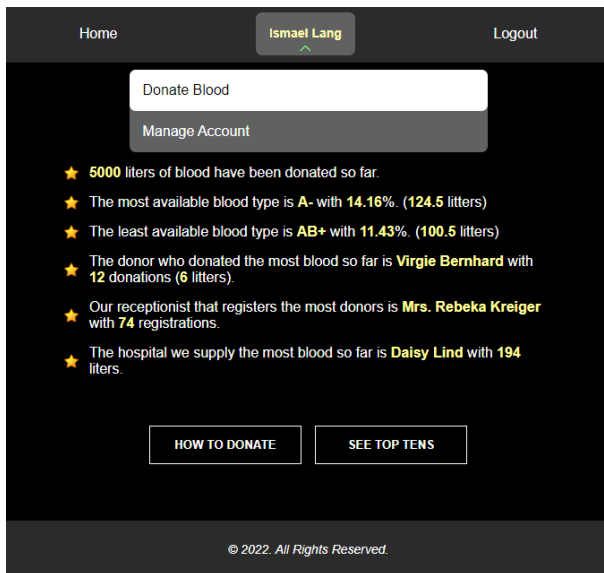


Figure 6

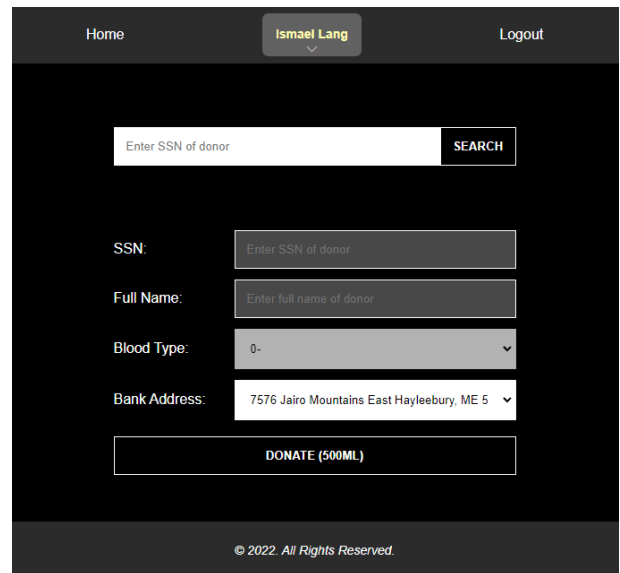


Figure 7

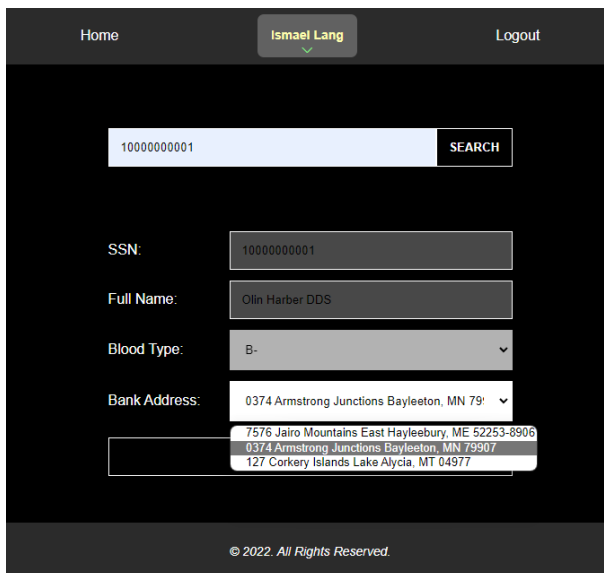


Figure 8

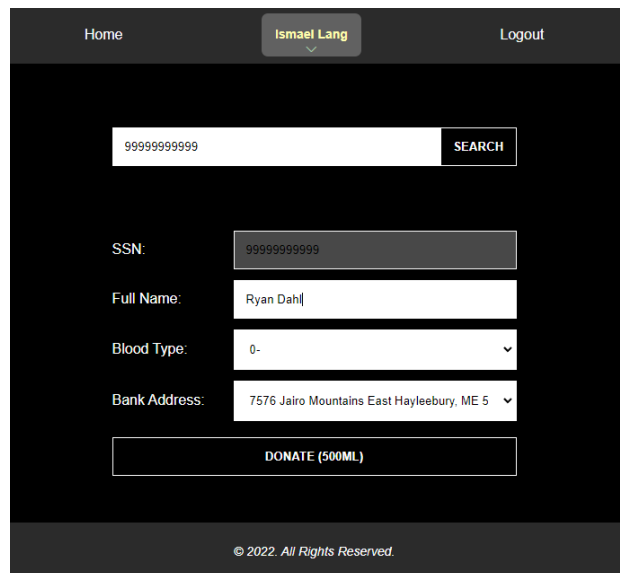


Figure 9

4. Order Blood

The upper navigation menu of the users logged in as a hospital is shown in Figure 10. Hospitals can view their past blood orders on the "My Orders" page (Figure 11). They can order new blood on the "Order blood" page (Figure 12). After the blood order is placed, the hospital is redirected to the "My Orders" page, where they can cancel their pending orders (Figure 13).

Blood orders are randomly distributed to system managers to be managed (approved or rejected). How system managers manage these orders is explained on the next page.

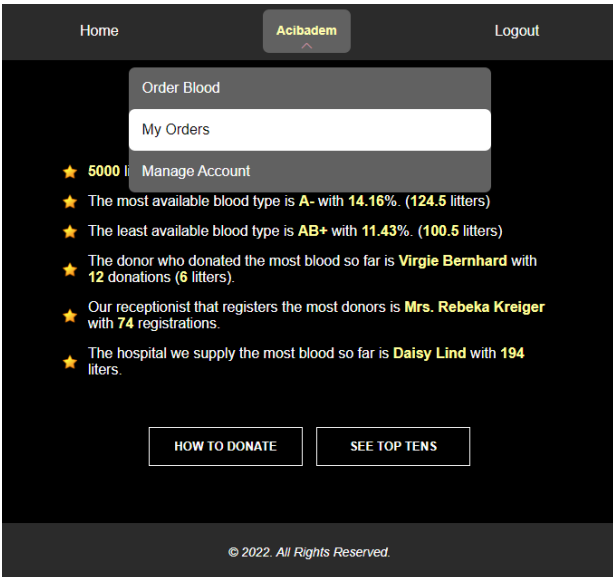


Figure 10

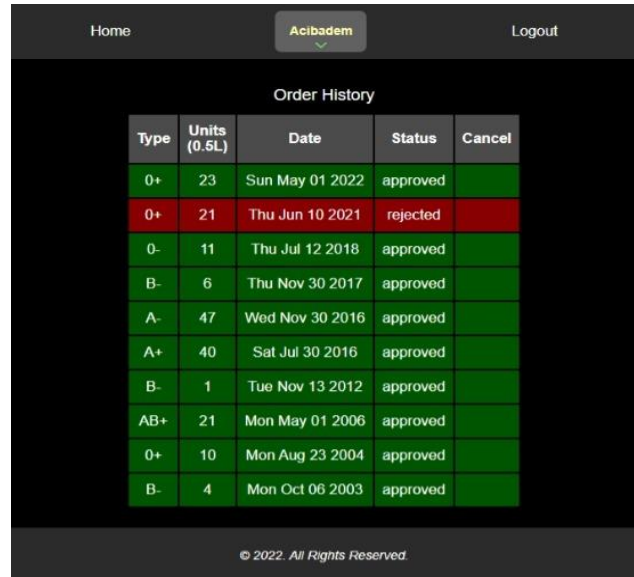


Figure 11

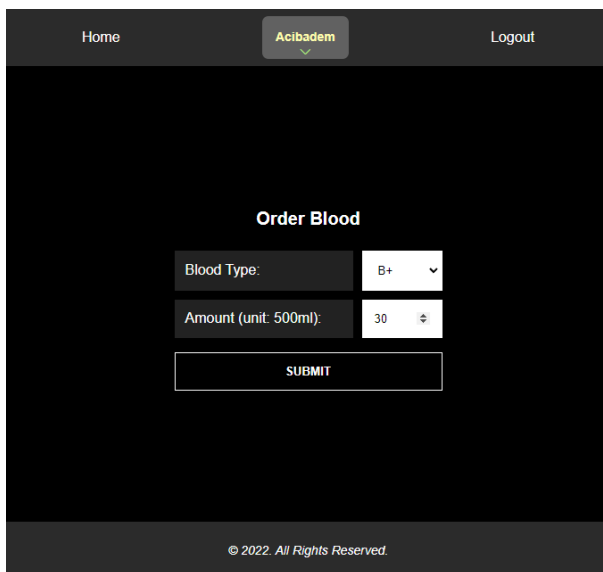


Figure 12

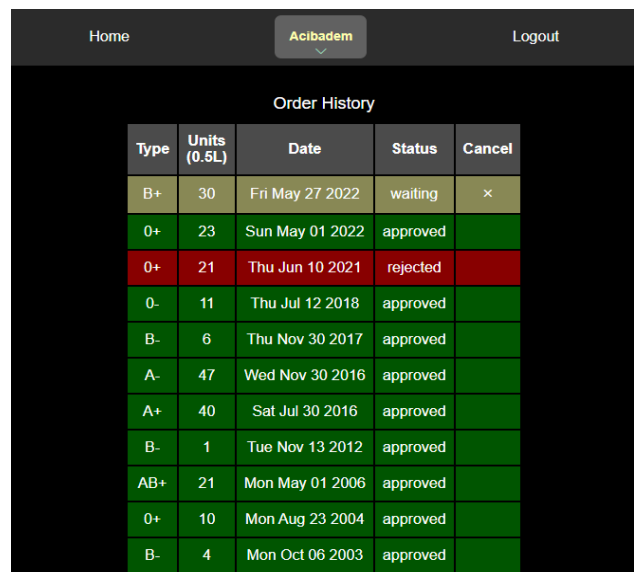


Figure 13

5. Manager Orders

The upper navigation menu of the users logged in as a system manager is shown in Figure 14. For security reasons, only system managers register users. The system user registration menu is shown in Figure 15. The management page of blood orders given by hospitals is shown in figure 16. When a system manager approves an order, the quantities of blood in the blood banks are reduced simultaneously in the Type/Units table on the same page (Figure 16).

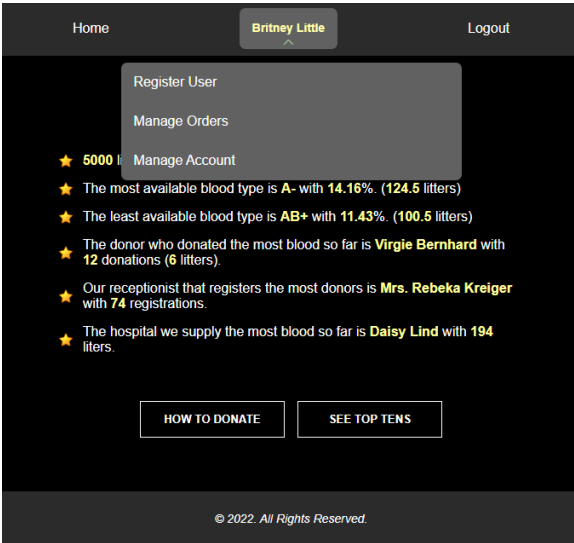


Figure 14

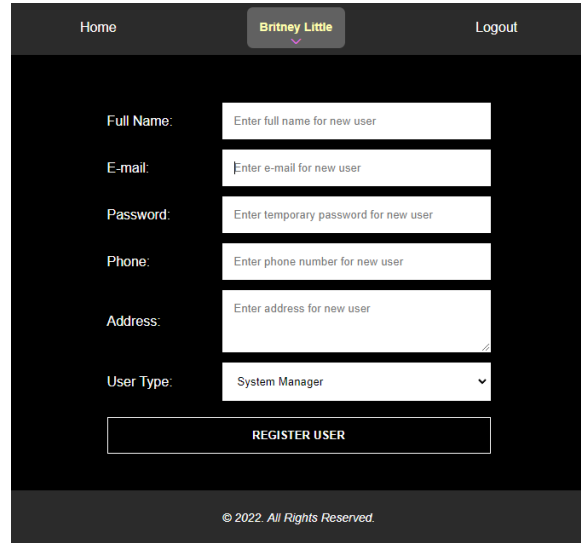


Figure 15

Home

Britney Little

Logout

Please manage orders!

Hospital	Type	Units	Order Date	Approve	Reject
Allene Reichel	O+	16	5/28/2022	✓	✕
Fay Luettgen PhD	A-	6	5/12/2022	✓	✕
Katherine Kilback V	A-	3	5/4/2022	✓	✕
Mr. Gianni Miller	B+	47	4/7/2022	✓	✕
Angeline Paucek Sr.	A+	1	12/23/2021	✓	✕
Hellen Shanahan	A+	40	12/16/2021	✓	✕
Allene Reichel	AB+	19	10/30/2021	✓	✕
Prof. Ethan Streich	A-	29	7/8/2021	✓	✕
Mabel Christiansen	B+	12	5/3/2021	✓	✕

Type	Units (0.5L)
B-	233
O+	228
O-	213
AB-	212
A-	207
AB+	201
B+	172
A+	134

© 2022. All Rights Reserved.

Figure 16

6. Manage Account

Every logged in user can access the manage account page. On this page, the user can edit some of their information or delete their account (Figure 17). The user must mark the check box of the information he/she wants to edit (Figure 18). When the user clicks the Delete account button, the user is asked whether he/she is sure about this (Figure 19). The result of the user's attempt to log in to the system, who has deleted his/her account, is shown in figure 20.

The screenshot shows the 'Manage Account' page for a user named 'Britney Little'. At the top, there are links for 'Home', the user's name 'Britney Little' with a dropdown arrow, and a 'Logout' link. The main content area contains three sections: 'Update Password' with an unchecked checkbox and a password input field; 'Update Phone Number' with an unchecked checkbox and a phone number input field containing '839-919-57'; and 'Update Address' with an unchecked checkbox and an address input field containing '1807 Kuhic Summit, East Amanda, NC 06908-1584'. Below these is a large 'UPDATE USER' button. At the bottom, there is a red 'Delete Account' link and a footer with '© 2022. All Rights Reserved.'

Figure 17

This screenshot is identical to Figure 17, but the checkboxes for 'Update Phone Number' and 'Update Address' are now checked. The input fields for these fields are highlighted in white, indicating they are active or selected.

Figure 18

The screenshot shows a confirmation dialog box overlaid on the 'Manage Account' page. The dialog has a title bar that says 'localhost:3000 says' and contains the text 'Are you sure you want to delete your account?'. It has two buttons: 'OK' and 'Cancel'. The background page is dimmed but the form elements from Figure 17 are still visible.

Figure 19

The screenshot shows the login page after the user's account has been deleted. A message at the top says 'localhost:3000 says' followed by 'User is not found!'. Below this, a text box contains the message: 'If you are unable to login as a receptionist, hospital or system administrator, please [contact us](#).' The login form itself has an email input field with 'manager@example.com', a password input field with masked characters, and a yellow 'LOGIN' button. The footer shows '© 2022. All Rights Reserved.'

Figure 20