

1) Introduction:

Overview of the Project-

CRUDCare is a comprehensive Blood Bank Management System designed to streamline and manage the entire life cycle of blood donations, requests, and inventory. It allows multiple user roles—admin, hospital representatives, donors, recipients, lab technicians, and inventory managers—to collaborate efficiently within a secure and user-friendly environment. The system ensures that blood donations are tracked, requests are fulfilled promptly, and inventory levels are properly maintained, all while providing real-time data access and management for different stakeholders.

Objectives:

- * To provide a centralized platform for managing blood donations, requests, and inventory.
- * To enable efficient user role management with secure access control for various stakeholders.
- * To track and manage blood units from donation to testing and storage.
- * To ensure accurate and timely fulfillment of blood requests.
- * To maintain transparency and accountability across all blood bank operations.
- * To implement an intuitive and user-friendly interface for streamlined operations and real-time data access.

2) System Design:

Database Design

The database for CRUDCare is designed to efficiently handle the operations related to blood donations, requests, inventory management, and user roles, with a well-structured relational schema. Each table is dedicated to storing relevant data for the different modules of the system, ensuring integrity, consistency, and scalability.

Table Breakdown

Users Table: Central repository for user information and roles. Admins use this to manage system-wide operations without a dedicated admin table.

Blood Requests Table: Manages requests for blood, tracking the type, volume, and status of each request.

Blood Test Results Table: Records test results for blood units to ensure they meet safety standards.

Blood Units Table: Details each blood unit, including its donor, type, and current status.

Donations Table: Tracks donations, linking each one to the donor, blood unit, and location.

Hospital Representative Info Table: Contains information about representatives from hospitals and their associated locations.

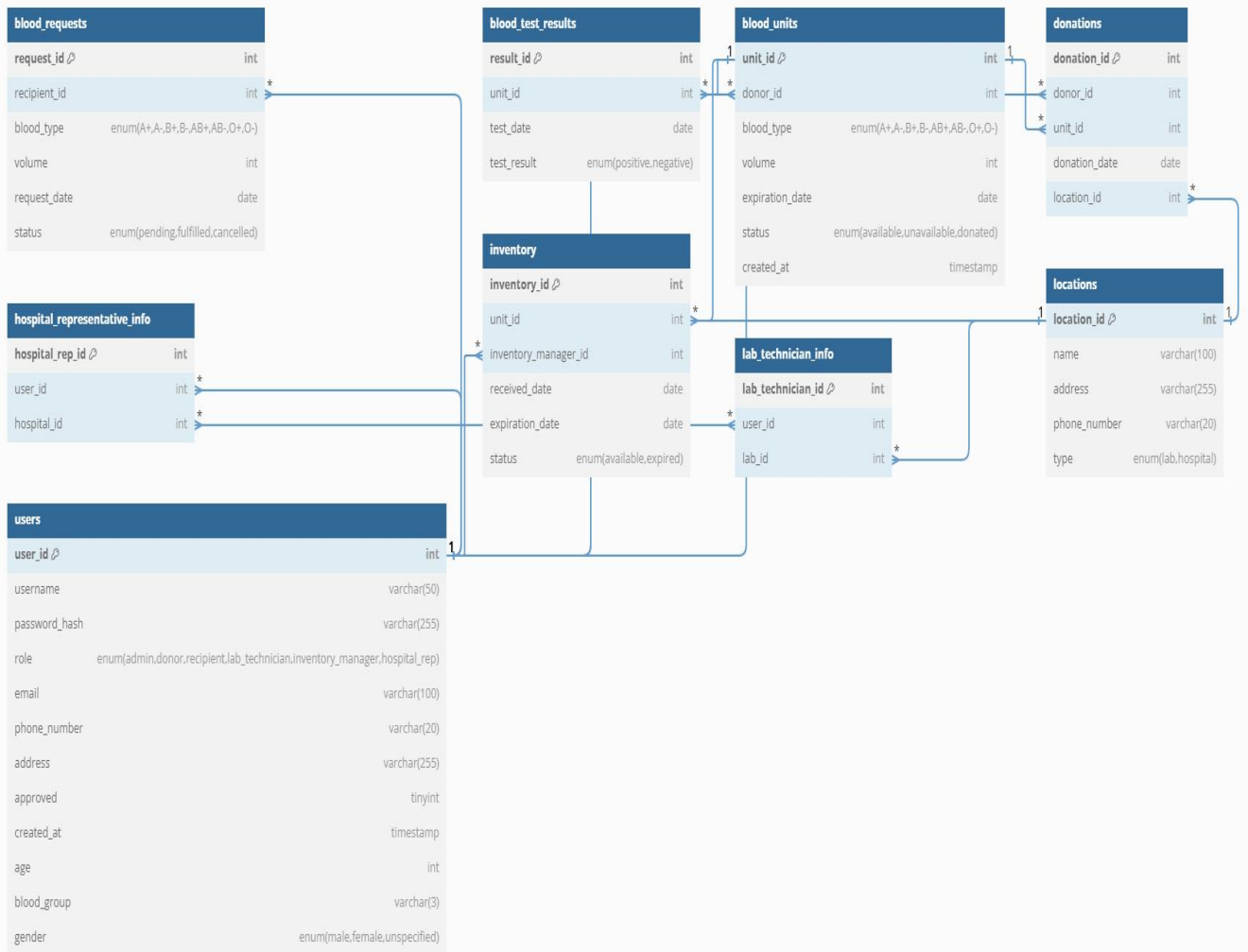
Inventory Table: Monitors blood units in inventory, including their status and expiration dates.

Lab Technician Info Table: Stores information about lab technicians and the labs where they work.

Locations Table: Keeps track of lab and hospital locations, including contact details.

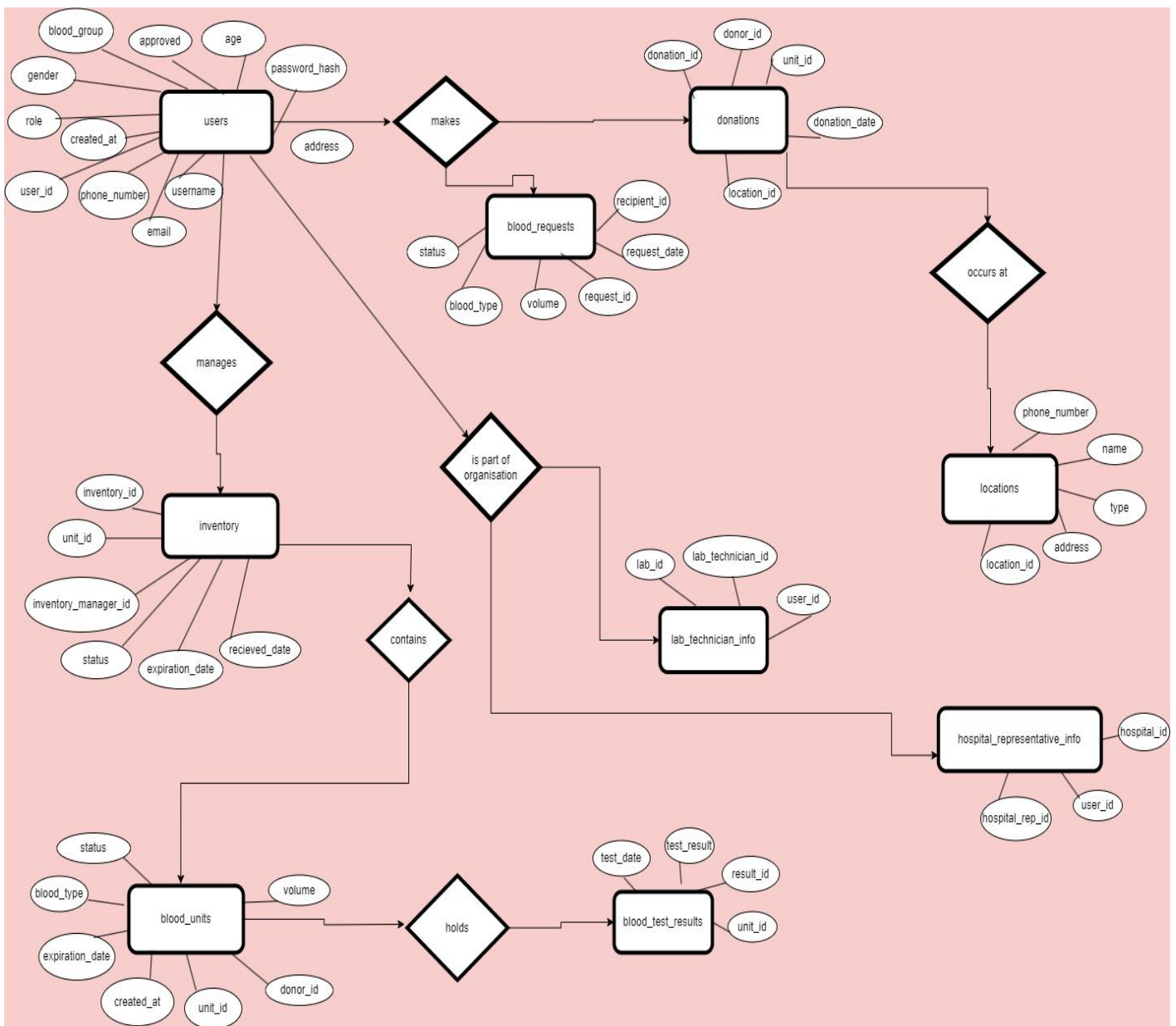
Admin Role: Admins manage the system's overall operations, including user management and overseeing donations and inventory, using their elevated privileges through the users table without a separate admin table.

Relationship Diagram Between the Tables



ER Diagram

The ER diagram for CRUDCare illustrates how the system's entities are connected. It shows users, donations, blood units, requests, and test results as distinct entities, represented by rectangles. Relationships like donations, blood requests, and blood tests are depicted through connecting lines, highlighting how users donate blood, how blood units are linked to test results, and how inventory is managed. Each entity's key attributes are represented, ensuring a clear view of how data flows between different parts of the system.



System Architecture

The system follows a three-tier architecture to efficiently manage and operate the CRUDCare blood bank management system:

Front-End:

Technologies Used: HTML, CSS, JavaScript, and Bootstrap.

Purpose: This tier is responsible for creating the user interfaces for different roles within the system. It ensures a user-friendly experience by presenting data and functionality in a visually appealing and interactive manner.

Back-End:

Technology Used: PHP.

Purpose: PHP handles the business logic of the application, processes user requests, and interacts with the database. It manages the functionality behind operations such as adding donations, managing user accounts, and processing blood requests.

Database:

Technology Used: MySQL.

Purpose: MySQL stores and manages all data related to users, blood requests, blood units, and other system components. It ensures data integrity and provides efficient data retrieval for the front-end and back-end components.

Data Flow

The data flow in the CRUDCare system follows a structured process involving the front-end interfaces, back-end logic, and database. Here's a step-by-step explanation:

User Input: A user interacts with the front-end interface (e.g., a hospital representative submits a blood request via a web form).

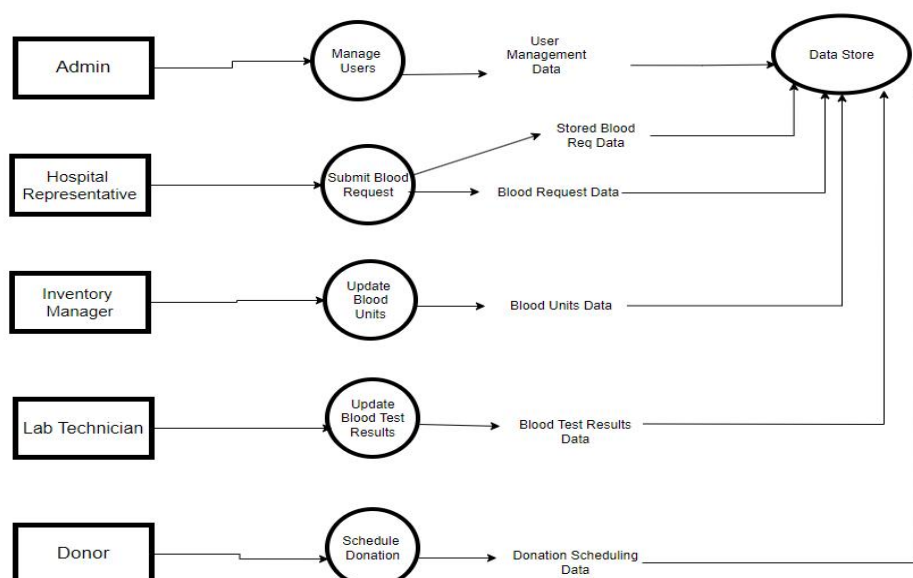
Front-End to Back-End: The data entered in the form is sent to the server using HTTP requests (e.g., via POST method). This request is handled by PHP scripts on the back-end.

Back-End Processing: The PHP back-end receives the data, performs validation checks, and processes the request. This might include sanitizing input to prevent security issues and ensuring all required fields are completed.

Database Interaction: After validation, the processed data is sent to the database. For instance, the PHP script executes an SQL INSERT statement to store the blood request in the blood_requests table.

Response: Once the data is successfully stored, the system might provide feedback to the user, such as a confirmation message or a redirect to another page.

Data Retrieval: For dynamic content, the process can involve fetching data from the database. For example, when a user views a list of blood requests, the PHP back-end executes an SQL SELECT query to retrieve the relevant records and then sends them to the front-end for display.



DBMS Features Used

Data Models:

The CRUDCare system uses a relational data model as it is built on MySQL, which organizes data into structured tables with relationships defined by foreign keys. Each entity in the system, such as users, blood requests, donations, and blood units, is represented as a table. This model ensures data integrity and allows for efficient data retrieval using SQL queries.

SQL Queries:

Here are examples of the key SQL queries used in the CRUDCare system:

SELECT: To retrieve data about available blood units.

Sql

```
SELECT * FROM blood_units WHERE status = 'available';
```

JOIN: To retrieve blood requests along with recipient details.

Sql

```
SELECT br.request_id, u.username, br.blood_type, br.volume, br.status  
FROM blood_requests br  
JOIN users u ON br.recipient_id = u.user_id;
```

UPDATE: To update the status of a blood request to 'fulfilled'.

Sql

```
UPDATE blood_requests SET status = 'fulfilled' WHERE request_id = 1;
```

DELETE: To delete a specific blood unit that has expired.

Sql

```
DELETE FROM blood_units WHERE expiration_date < CURDATE();
```

Transactions:

In CRUDCare, transactions are used to ensure the ACID properties (Atomicity, Consistency, Isolation, Durability) when performing critical operations like blood unit updates or request fulfillment. For instance, when updating the blood request and the associated inventory in a single operation, a transaction ensures that either both updates succeed, or neither does, to maintain consistency.

Indexes and Optimization:

To improve performance, indexes are used on key columns such as `user_id`, `request_id`, and `unit_id`. Indexing these columns speeds up data retrieval for common queries involving joins or lookups based on primary or foreign keys.

For example:

Sql

```
CREATE INDEX idx_user_id ON users(user_id);  
CREATE INDEX idx_request_id ON blood_requests(request_id);
```

These indexing techniques help optimize query performance, particularly in large datasets like user and blood request tables.

System Implementation

Development Tools:

The CRUDCare Blood Bank Management System was implemented using the following tools and technologies:

Programming Languages

PHP: For back-end logic and server-side scripting.

HTML, CSS, JavaScript: For building the front-end user interfaces.

Frameworks

Bootstrap: For responsive design and styling of the user interface.

Database Management System (DBMS)

MySQL: Used to store and manage all data, including users, blood units, requests, and donations.


Homepage

[Home](#)
[Services](#)
[Looking for Blood](#)
[Want to Donate Blood](#)
[Learn About Donation](#)
[Meet the Developers](#)

Save Lives by Donating Blood

Your blood can give someone a new lease on life

[Login to Account](#)




Services

Our services ensure safe and effective blood transfusion processes

Blood Data Management
Empowerment and empowerment for blood donations, storage, and distribution.

Tracking and Reporting
Real-time tracking and detailed reporting of blood inventory and donor activities.

Emergency Blood Supply
Special response to emergency blood requests.



HOW TO FIND BLOOD

Use our system to quickly and easily find the blood you need.

- Register or log in to your account
- Search for the required blood type
- Contact the donor or blood bank

Our system ensures a fast and reliable way to get the blood you need.

Want to Donate Blood?

Join our cause and donate blood to save lives.

Check your eligibility to donate.
Find a blood donation center near you.
Prepare yourself for donation day.

How to Donate Blood

Follow these simple steps to donate blood and save lives.

1. Register yourself at a blood donation center.
2. Fill out a donor history questionnaire.
3. Undergo a medical physical to check your eligibility.
4. Donate blood and receive refreshments afterwards.

Your contribution can save lives. Be a hero, donate blood.

[Register Now](#)
[Activate the system](#)

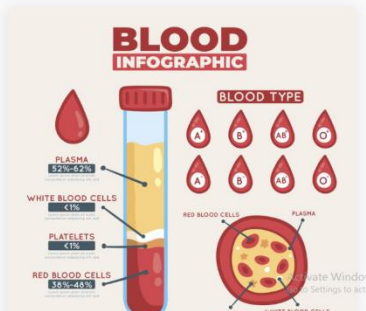
[Learn About Donation](#)

Learn About Donation

Understand the process and compatibility of blood donation

Blood Group Compatibility


Blood Type	Can Donate To	Can Receive From
A+	A+, AB+	A+, A-, O+, O-
O+	O+, A+, B+, AB+	O+, O-
B+	B+, AB+	B+, B-, O+, O-
AB+	AB+	All Blood Types
A-	A-, A+, AB-, AB+	A-, O-
O-	All Blood Types	O-
B-	B-, B+, AB-, AB+	B-, O-
AB-	AB-, AB+	AB-, A-, B-, O-




[Meet the Developers](#)

Meet the Developers


Our talented team of developers who make it all possible




Shezan
Lead Developer




Sakib
UI/UX Designer



Shadman
Quality Assurance



Sharif
Database Specialist



Rii
Backend Developer

CRUDCare
 Your partner in saving lives
 123 Street Name
 City, State, ZIP Code
 Country
 Phone: +1 234 567 890
 Email: info@example.com


Useful Links
[Home](#)
[Services](#)
[Looking for Blood](#)
[Want to Donate Blood](#)
[Learn About Donation](#)
[Meet the Developers](#)

Our Services
[Blood Data Management](#)
[Tracking and Reporting](#)
[Emergency Blood Supply](#)

© Copyright CRUDCare Blood Bank. All Rights Reserved

Front-End Interface

Admin Dashboard


Manage Users
Manage Donations
Manage Requests

Admin Dashboard


Logout

Manage Users

Add User

User ID	Username	Email	Role	Approved	Actions
1	admin	admin1@example.com	admin	1	<button>Delete</button>
2	donor	donor1@example.com	donor	1	<button>Delete</button>
3	recipient	recipient1@example.com	recipient	1	<button>Delete</button>
4	lab	labtech1@example.com	lab_technician	1	<button>Delete</button>
5	inventory	invmanager1@example.com	inventory_manager	1	<button>Delete</button>
6	hospital	hosprep1@example.com	hospital_rep	1	<button>Delete</button>
7	hospital2	hosrep2@example.com	hospital_rep	0	<button>Delete</button>
8	recipient2	recipient2@example.com	recipient	1	<button>Delete</button>

Hospital Representative Dashboard


Submit Blood Request
Track Blood Requests
Blood Availability

Hospital Representative Dashboard

Logout

Hospital: North Hospital
Location: 456 Hospital St, Hospital City
Phone: 2222222222
Representative: hospital
Representative Phone: 6789012345


Submit Blood Request

Select Blood Type Blood Type

Volume (in mL) Volume (in mL)

Submit Request

Recipient Dashboard


Dashboard
Request for Blood
Blood Requests History
Available Blood

Pending Blood Requests

You have 2 pending blood requests.

Available blood

Inventory ID	Unit id	Inventory Manager Id	Received Date	Expiration Date	Status
1	1	5	2024-06-03	2024-12-31	available
2	2	5	2024-06-17	2024-11-30	available

Donor Dashboard

Donor Dashboard

Logout

Profile Information

Username:

donor

Email:

donor1@example.com

Phone Number:

2345678901

Address:

456 Donor St, Donor City

Blood Donation

Schedule a Donation

Preferred Donation Date

mm/dd/yyyy

Preferred Time Slot

08:00 - 10:00

Preferred Location

Enter donation center or hospital

Schedule Donation

Lab Technician Dashboard

CRUDCare

Blood Units

Blood Test Results

Lab Technician Dashboard

Logout

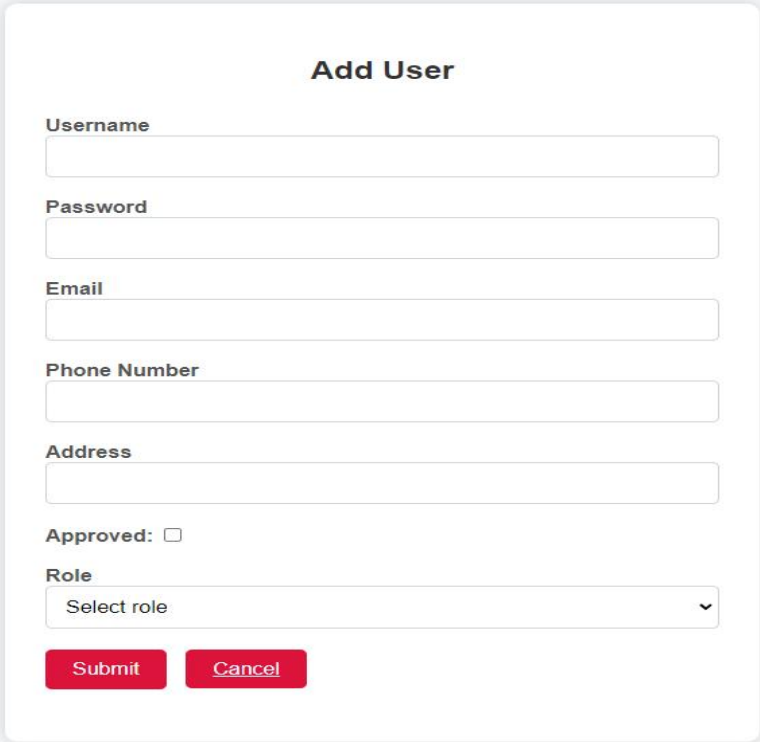
Blood Units

Add Blood Units

Unit ID	Blood Type	Volume	Expiration Date	Status	Test Result	Actions
1	A+	500	2024-12-31	available	negative	<div>Add Blood Test</div> <div>Update</div> <div>Remove</div>
2	O-	450	2024-11-30	available	negative	<div>Add Blood Test</div> <div>Update</div> <div>Remove</div>

Back-End Components

Add User: A form used by administrators to add new users to the system, including assigning roles and setting initial information.



The 'Add User' form is a white rectangular box with a light gray border and a subtle drop shadow. It features a title 'Add User' at the top center. Below the title are five text input fields: 'Username', 'Password', 'Email', 'Phone Number', and 'Address'. Each field is followed by a red 'Submit' button and a red 'Cancel' button. Below the 'Address' field is an 'Approved' checkbox and a 'Role' dropdown menu with the placeholder text 'Select role'. The form is set against a light gray background.

Add User

Username

Password

Email

Phone Number

Address

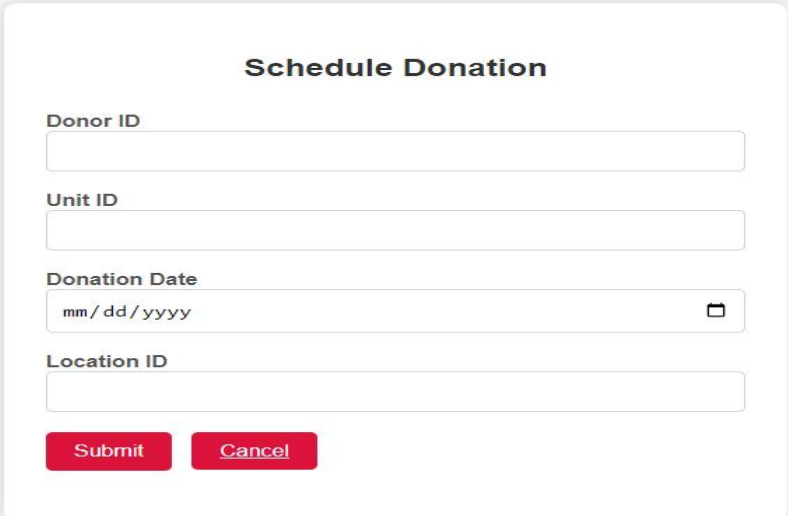
Approved: ☐

Role

Select role

Submit Cancel

Schedule Donation: Allows admins and users to schedule blood donations, capturing details like the donor, donation date, and location.



The 'Schedule Donation' form is a white rectangular box with a light gray border and a subtle drop shadow. It features a title 'Schedule Donation' at the top center. Below the title are four text input fields: 'Donor ID', 'Unit ID', 'Donation Date', and 'Location ID'. The 'Donation Date' field includes a date picker icon. Each field is followed by a red 'Submit' button and a red 'Cancel' button. The form is set against a light gray background.

Schedule Donation

Donor ID

Unit ID

Donation Date

mm/dd/yyyy

Location ID

Submit Cancel

Edit Blood Request: Provides functionality for modifying existing blood requests, including changes to request volume, status, and blood type.

Edit Blood Request

Blood Type

Volume (in mL)

[Go Back to Dashboard](#)

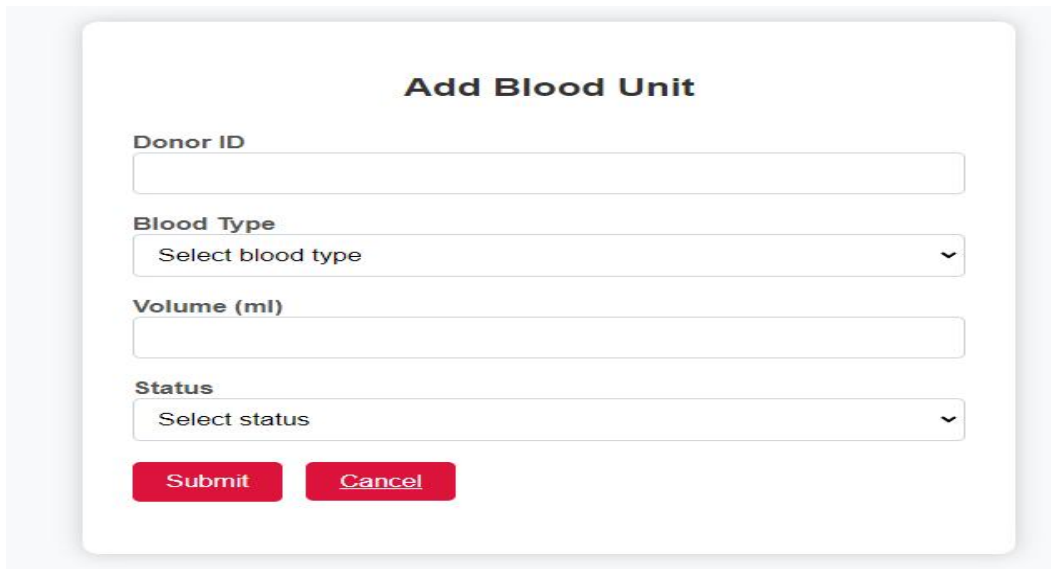
Update Blood Request: Allows for changes to existing blood requests, including adjustments to volume and blood type.

Update Blood Request

Blood Type

Volume (ml)

Add Blood Unit: Used to add new blood units to the system, including details such as the donor, blood type, volume, and expiration date.



The 'Add Blood Unit' form is a white card with rounded corners and a subtle shadow. It features a title 'Add Blood Unit' in bold black text at the top center. Below the title are four input fields: 'Donor ID' (a text box), 'Blood Type' (a dropdown menu with 'Select blood type' as the placeholder), 'Volume (ml)' (a text box), and 'Status' (a dropdown menu with 'Select status' as the placeholder). At the bottom of the form are two red buttons: 'Submit' and 'Cancel'.

Add Blood Unit

Donor ID

Blood Type

Select blood type

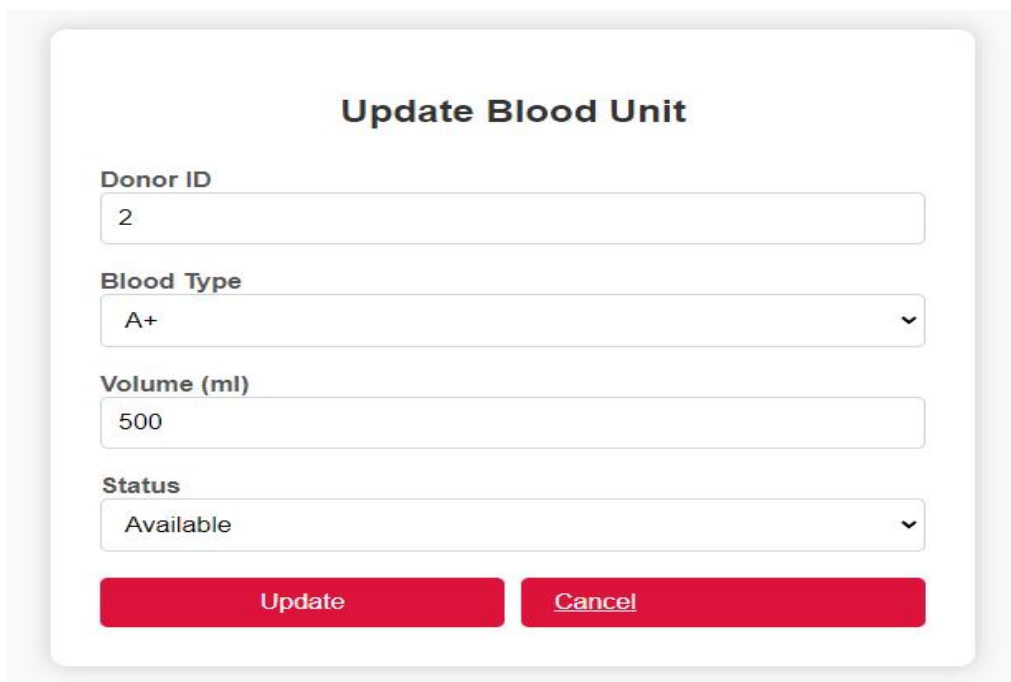
Volume (ml)

Status

Select status

Submit Cancel

Update Blood Unit: Allows for updates to existing blood units, including adjustments to volume, expiration date, and status.



The 'Update Blood Unit' form is a white card with rounded corners and a subtle shadow. It features a title 'Update Blood Unit' in bold black text at the top center. Below the title are four input fields: 'Donor ID' (a text box containing the value '2'), 'Blood Type' (a dropdown menu with 'A+' as the selected value), 'Volume (ml)' (a text box containing the value '500'), and 'Status' (a dropdown menu with 'Available' as the selected value). At the bottom of the form are two red buttons: 'Update' and 'Cancel'.

Update Blood Unit

Donor ID

2

Blood Type

A+

Volume (ml)

500

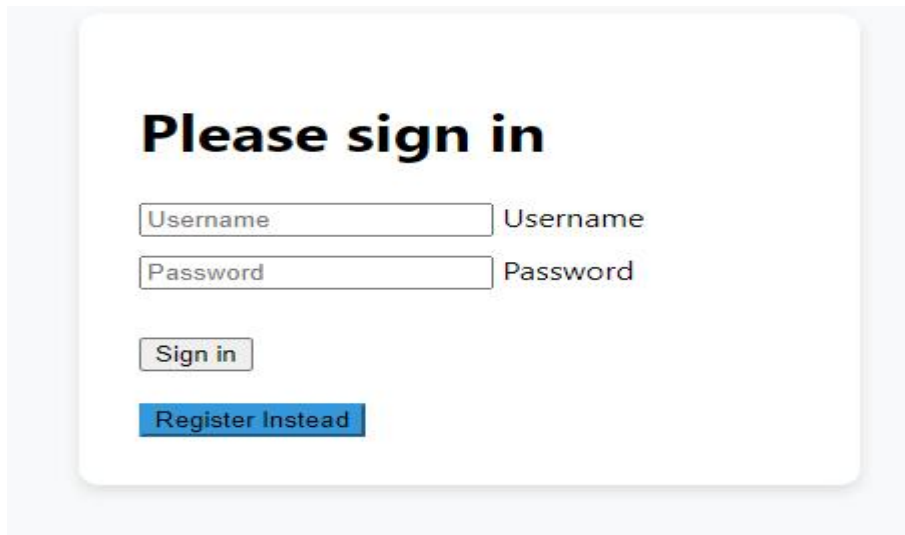
Status

Available

Update Cancel

User Authentication & Security

Login: Users enter their credentials (username and password) to access the system. Authentication is handled by verifying the provided credentials against stored hashed values in the database. Successful login grants access to user-specific features and data based on their role.



Please sign in

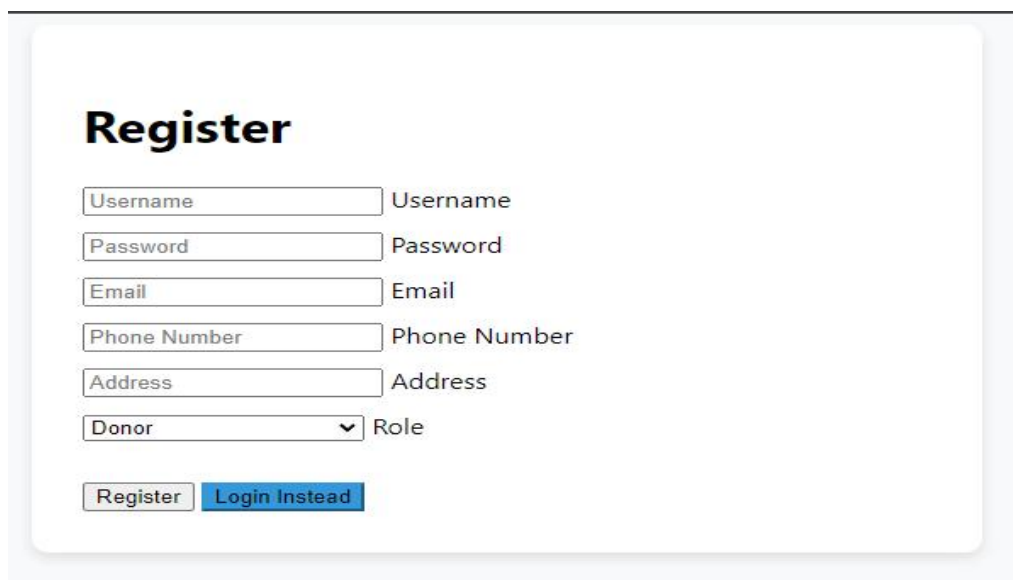
Username

Password

Sign in

Register Instead

Register: New users can create an account by providing necessary details such as username, email, password, and role. Registration involves validating the input data, hashing the password, and storing the new user information in the database. Newly registered users must be approved (if applicable) before they can fully access the system.



Register

Username

Password

Email

Phone Number

Address

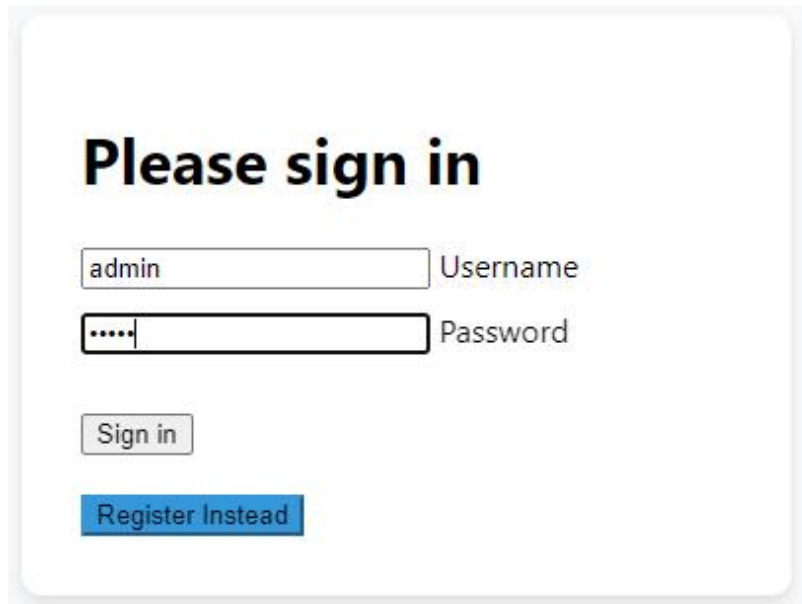
Donor Role

Register Login Instead

Testing

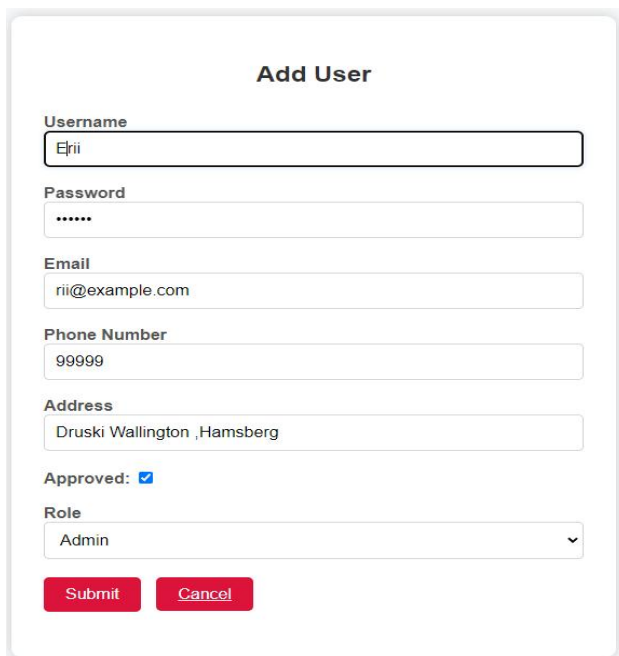
Key test cases were designed to ensure the proper functionality of CRUDCare:

User Login: Testing for correct username and password combinations, and appropriate error messages for incorrect login attempts.



A sign-in form titled "Please sign in". It contains two input fields: "Username" with the value "admin" and "Password" with masked characters ".....". Below the fields are two buttons: "Sign in" and "Register Instead".

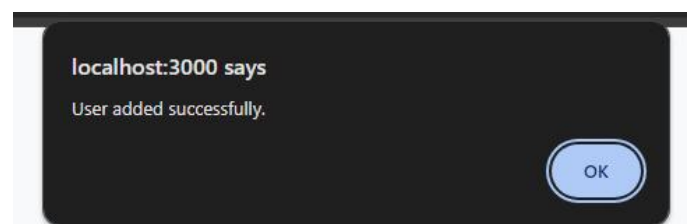
User Management: Testing adding, editing, and deleting user accounts to confirm proper handling of user roles and information.



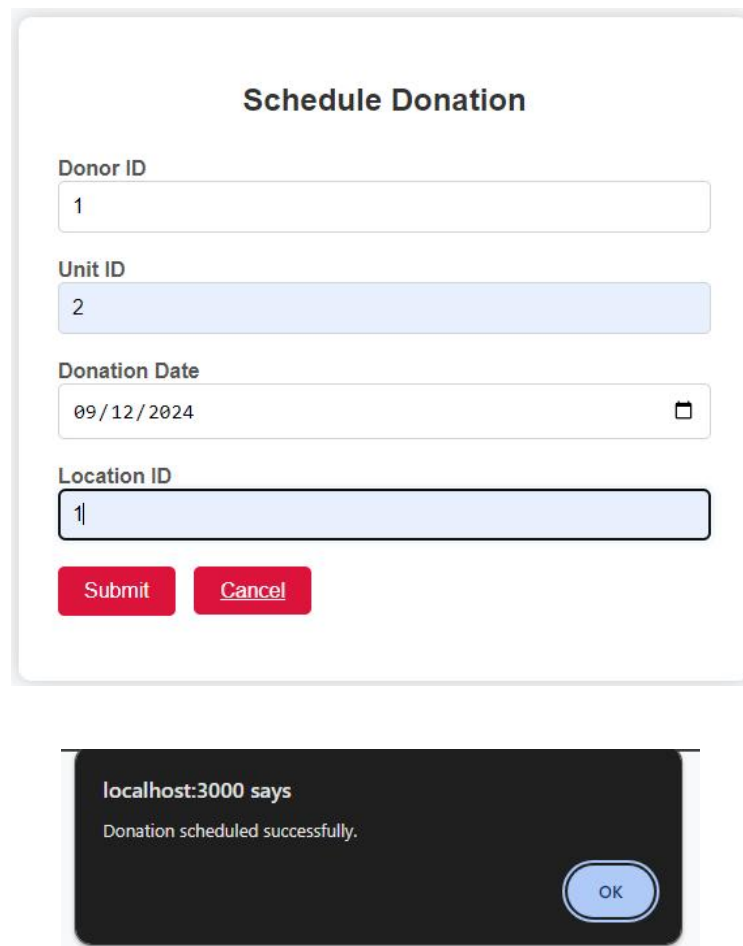
An "Add User" form with the following fields and values:

- Username: Erii
- Password:
- Email: rii@example.com
- Phone Number: 99999
- Address: Druski Wallington ,Hamsberg
- Approved: ☒
- Role: Admin (dropdown menu)

At the bottom are two buttons: "Submit" and "Cancel".



Blood Donation Scheduling: Verifying that donation schedules are saved correctly and associated with the donor's information.



The image shows a web form titled "Schedule Donation" with the following fields: "Donor ID" (value: 1), "Unit ID" (value: 2), "Donation Date" (value: 09/12/2024), and "Location ID" (value: 1). Below the fields are "Submit" and "Cancel" buttons. Below the form is a dark notification box with the text "localhost:3000 says Donation scheduled successfully." and an "OK" button.

Error Handling:

The system includes robust error handling mechanisms:

Invalid Inputs: User input validation is performed on both the client-side (using JavaScript) and server-side (PHP) to prevent invalid data from being submitted.

Database Errors: For issues like database connection failures or query errors, custom error messages are shown, and critical errors are logged for further investigation.

Performance Testing:

Query Execution Time: Database queries were tested for performance using SQL EXPLAIN to ensure that JOIN operations and SELECT queries retrieve data efficiently.

System Load Handling: Simulated multiple concurrent users to test the system's ability to handle data loads, ensuring it maintains performance under high user traffic.