

Blood Bank Management System: Progression Report

Project Title

Blood Bank Management System (BBMS)

Objectives

The primary objective of the BBMS is to automate and streamline the processes involved in blood donation and transfusion, ensuring efficient management of blood stocks, donor and recipient information, and administrative tasks. It aims to enhance operational efficiency, provide a user-friendly interface, and ensure the safety and reliability of blood transfusions.

Previously Proposed Features/Modules/Properties

- Admin Panel: User management, inventory management, report generation, notifications.
- Donor Panel: Registration, profile management, appointment scheduling, donation history, notifications.
- Recipient Panel: Blood request submission, request tracking, profile management, notifications.
- Lab Technician Panel: Blood testing, result management, equipment management, report generation.
- Inventory Manager Panel: Stock monitoring, storage management, expiry management.
- Hospital/Clinic Representative Panel: Blood request management, communication, notifications.
- ER Diagram: Entities (Admin, Donor, Recipient, Lab Technician, Inventory Manager, Blood Unit, Hospital/Clinic Representative) and their relationships.

Accomplished Features/Modules/Properties

- Admin Panel: User management, inventory management.
- Recipient Panel: Blood request submission, request tracking.
- Inventory Manager Panel: Stock monitoring, storage management, expiry management.
- Hospital/Clinic Representative Panel: Blood request management, communication.
- ER Diagram: Entities (Users, Locations, BloodUnits, Donations, BloodRequests, BloodTestResults, Inventory, HospitalRepresentativeInfo, LabTechnicianInfo) and their relationships based on the latest database schema.

Database: Developed and optimized a relational database schema to support efficient data management across all modules, including:

- ❖ **Users:** Manages user accounts with roles and permissions.
- ❖ **Locations:** Stores information about labs and hospitals.
- ❖ **BloodUnits:** Tracks blood donations and availability.
- ❖ **Donations:** Records donor activities and blood donations.
- ❖ **BloodRequests:** Manages requests for blood by recipients.
- ❖ **BloodTestResults:** Stores results from blood tests performed by lab technicians.
- ❖ **Inventory:** Tracks blood inventory and storage details.
- ❖ **HospitalRepresentativeInfo** and **LabTechnicianInfo:** Stores additional information for hospital representatives and lab technicians.

Future Works

- **Donor Section Implementation:** Develop the donor section with functionalities such as donor registration, profile management, appointment scheduling, donation history tracking, and eligibility status updates.
- **Lab Section Development:** Implement the lab section to manage blood testing processes, including recording and updating test results, managing testing equipment, and generating test outcome reports.
- **Enhancements to Admin Section:** Improve the admin section with advanced user management capabilities, inventory oversight features, and comprehensive reporting tools for better operational control.
- **Enhancements to Inventory Manager Section:** Extend the inventory manager section with enhanced stock monitoring functionalities, optimized storage management tools, and tools for managing blood unit expiry dates and disposal.
- **User Interface Refinements:** Conduct user interface refinements across all sections to enhance usability, streamline workflows, and ensure consistency in design elements.

Actions and Responsibilities of Each PHP File

Admin section:

add_user.php

- **Responsibility:** Handles the functionality to add a new user to the system.
- **Actions:** Validates input data, processes user registration, and updates the database with new user information.

cancel_donation.php

- **Responsibility:** Manages the cancellation of blood donation appointments.
- **Actions:** Retrieves donation appointments, allows administrators to cancel appointments, and updates the database accordingly.

dashboard.php

- **Responsibility:** Displays the main dashboard interface for administrators.
- **Actions:** Retrieves and displays critical system statistics, user information summaries, and provides navigation to other administrative tasks.

delete_user.php

- **Responsibility:** Handles the deletion of user accounts from the system.
- **Actions:** Verifies permissions, deletes user records from the database, and manages associated data integrity concerns.

edit_user.php

- **Responsibility:** Manages the editing and updating of user information.
- **Actions:** Retrieves current user details, allows administrators to modify user data such as roles, permissions, and personal information, and updates the database accordingly.

schedule_donation.php

- **Responsibility:** Facilitates the scheduling of blood donation appointments.
- **Actions:** Validates donor eligibility, manages appointment scheduling, and updates the database with appointment details.

Hospital section:

cancel_request.php

- **Responsibility:** Manages the cancellation of blood requests submitted by hospitals/clinics.
- **Actions:** Retrieves active blood requests, allows hospital representatives to cancel specific requests, and updates the database accordingly.

dashboard.php

- **Responsibility:** Displays the main dashboard interface for hospital representatives.
- **Actions:** Retrieves and displays information related to blood requests, request statuses, and provides navigation to other functionalities such as request submission and request cancellation.

edit_request.php

- **Responsibility:** Handles the editing of existing blood requests.
- **Actions:** Retrieves current request details, allows hospital representatives to modify request parameters such as blood type and volume, and updates the database with the edited request information.

submit_request.php

- **Responsibility:** Facilitates the submission of new blood requests by hospitals/clinics.
- **Actions:** Validates request parameters, manages submission of new requests into the system, and updates the database with the new request details.

Inventory section:

dashboard.php

- **Responsibility:** Displays the main dashboard interface for inventory managers.
- **Actions:** Retrieves and displays critical inventory information such as blood stock levels, expiration dates, and storage details. It provides an overview of the current status of blood units in the inventory.

Recipient section:

count_request_pending.php

- **Responsibility:** Counts the number of pending blood requests made by recipients.
- **Actions:** Retrieves pending blood request data from the database and counts the total number of requests awaiting fulfillment.

dashboard.php

- **Responsibility:** Displays the main dashboard interface for recipient representatives.
- **Actions:** Retrieves and displays information related to blood requests, request statuses, and provides navigation to other functionalities such as requesting blood and managing requests.

fetch_blood_requests.php

- **Responsibility:** Fetches details of all blood requests made by recipients.
- **Actions:** Retrieves comprehensive information about blood requests from the database, including request status, blood type, volume, and request date.

navbar.php

- **Responsibility:** Provides the navigation bar interface used across multiple pages in the recipient representative section.
- **Actions:** Defines the navigation structure and links for easy access to various sections and functionalities within the recipient representative dashboard.

request_for_blood.php

- **Responsibility:** Handles the process of submitting a new blood request by recipients.
- **Actions:** Validates recipient input for blood request parameters such as blood type and volume, manages the submission process into the system, and updates the database with the new request details.

Auth section:

login.php

- **Responsibility:** *Manages user authentication and login functionality.*
- **Actions:** *Validates user credentials (username/email and password), authenticates users against the database, and grants access to the system upon successful login. It also handles session management to maintain user authentication across pages.*

logout.php

- **Responsibility:** *Handles user logout functionality.*
- **Actions:** *Terminates the user session, effectively logging the user out of the system and redirecting them to the login page or a specified destination.*

register.php

- **Responsibility:** *Manages user registration process.*
- **Actions:** *Validates user input for registration data such as username, email, password, and other required fields. It creates a new user account in the database upon successful validation and provides feedback to the user about the registration status.*

Latest ER Diagram

