Software Engineering Project

Blood Donation System

**Clarifying the project requirements:**

The project involves developing an application for managing blood donations in a hospital or clinic, with users falling into three main roles: Donor, Medical Staff (Assistant), and System Administrator. Each role will have access to specific functionalities, detailed below.

**General Objective:**

This application will allow users to manage the blood donation process, from scheduling a donation, checking eligibility, to processing collections and managing blood stocks.

**Roles and functionalities:**

**1.User (Donor)**

The donor is responsible for donating blood. Key functionalities include:

* **View personal data**: the user can view their personal information (name, address, CNP, etc.)
* **Schedule a blood donation**: the user can schedule a future donation
* **Fill in eligibility formular**: the donor can fill out a self-assessment form indicating health status and blood type. Based on the form, the system confirms eligibility
* **View donation history**: the user can see all past donations
* **Cancel or reschedule a donation**: the donor has the option to cancel or reschedule a previously set donation
* **Access to statistics and rewards**: the donor can see statistics about how many donations they’ve made and any rewards offered for their contribution

**2.Assistant (Medical Staff)**

The medical assistant is responsible for managing the donations in the system. Their functionalities include:

* **Process a donation**: the assistant records and processes blood donations
* **Access donor profiles**: the assistant can view the personal details and donation history for each donor
* **Generate and export reports**: reports on donations, blood stocks, eligible donors etc.
* **View statistics**: detailed statistics on donations and collections within the medical facility

**3.System Administrator**

The system administrator is responsible for managing the platform and monitoring all activity. Their functionalities include:

* **Manage users and medical staff:** Perform CRUD operations (create, read, update, delete) on users (donors, assistants). Admins can add, modify or delete accounts.
* **Monitor activities:** The admin can see who has donated, who processed the donations and other activity logs
* **Manage blood stocks:** Centralized control over blood stock levels

**4.Spectator**

The spectator role allows users to access the application without the need for account creation or login

* **View blood donation information** (faq about the donation process, benefits of donating blood, eligibility criteria for donors)
* **View educational resources**: access to educational materials about blood donation and its impact on the community
* **Access statistics** (blood type distribution, number of donors, number of medical staff, total donations made etc.)

**Technical Requirements**

**1.User Management**

* Sign-up and log-in functionalities for users
* Each user will have a specific role (Donor, Medical Staff, Administrator) and access to functionalities based on their role

**2.Role-based permissions**

* Access control for functionalities based on user roles
* Users can only access functionalities intended for them

**3.Interaction between Backend and Frontend**

* **RESTful API**: Django REST Framework to create an API for communication between the frontend and backend
* **Communication protocol**: ensure a clear method of communication

**4. Database Management**

* **Django ORM:** Django’s ORM for database interactions
* **Data security:** implement validation and sanitization for sensitive data, using Django Forms

**5.Creating a Desktop Interface**

* **Choosing a framework:** selecting an appropriate framework for the user interface
* **UI/UX Design:** create an intuitive and user-friendly interface, following UI/UX design principles

**6.Error Handling and User Feedback**

* **Error messages**: ensure the application properly handles communication and backend errors, providing feedback to users

**7.Security and Production Configurations**

* **Encryption of Sensitive Data**: if handling sensitive data, ensure it is properly encrypted and managed

9.**Documentation**

* Create clear documentation for users explaining how to use the application
* Include a technical description to facilitate understanding of the code by other developers