

|  |  |
| --- | --- |
| System  Requirement  For developer team |  |
| While designing and developing the system bear in mind that system must satisfy all requirements mentioned in this documents.  By Foyaj Ahmmad Farabi, AKMU, 17 May 2025 |  |

**Blood Donation Management System (BDMS)**

Appendix D: Detail System Requirements

**Section 1: User-Facing Functional Modules**

**1.1 User Registration Module**

**Requirement 1.1 (Single-Step Registration Form):**

The system shall implement a progressive disclosure one-step registration form.

**Requirement 1.1.1 (Form Auto-Save and Progress Tracking):**

The form shall automatically save data entered in each step.

**Requirement 1.1.2 (Real-Time Field Validation):**

The system shall validate each field in real-time, displaying suggestions if incorrect data is entered. If incorrect data is submitted, inline guidance will explain the correction.

On any form failure, display a toast message and scroll to the first invalid input.

**Requirement 1.1.3 (Eligibility Criteria Validation):**

* Age ≥ 18 years (based on DOB)
* Weight > 50 kg
* Hemoglobin > 13 g/dL (males), > 12 g/dL (females)
* Last donation ≥ 120 days ago

**Requirement 1.1.4 (Unique Donor ID Generation):**

Upon successful registration, the system shall auto-generate a unique donor ID in the format: BDMS-YYYY-XXXX.

**Requirement 1.1.5 (Success Confirmation Modal):**

A confirmation pop-up shall display: "Your account has been created successfully. Your Donor ID is BDMS-YYYY-XXXX.”

**Requirement 1.1.6 (Phone Verification Trigger):**

The system shall trigger the OTP verification (see Requirement 1.3) upon successful form submission.

**Requirement 1.1.7 (Field Details and Validations):**

* **First and Last Name**: First letter must be capitalized
* **Gender**: Dropdown (Male, Female, Other)
* **Birth Date**: Date picker (YYYY-MM-DD format); must not allow future dates or age < 18
* **Phone Number**: Default country code +880; 11-digit validation
* **Email**: Advanced validation (MX check, disposable detection, role-based check)
* **Username**: Must be unique; 3 validations: availability, character limit (200), character types (Letter and number only)
* **Password**: Min 8 chars, Max 64, Must include 1 uppercase, 1 lowercase, 1 number, 1 special character.
* **Address**: Street, searchable dropdowns for upozilla and zilla (Auto address suggestions from google map while typing)
* **Zip Code**: 4-digit numeric input with validation
* **Blood Group**: Dropdown (A+, A-, B+, B-, O+, O-, AB+, AB-)
* **Weight**: Must be numeric, >50kg
* **Hemoglobin**: Numeric input; allowed range 13-17 g/dL
* **Last Donation Date**: Must not be future date; selectable via calendar
* **Donation Count**: Max value capped at 250 (displaying suggestions to input correct data if value is inserted over 250)

**Requirement 1.1.8 (Form Submission):**

After filling up the required field on the registration form user shall submit the form by clicking “Submit” button.

* If the all-user input is in order, then after hitting Submit button system shall show a pop up with a confirmation message that will say “Your account has been created successfully” Your Donor ID ID: BDMS-YYYY-XXXX. – the system shall generate unique donor ID.
* On the pop up there will be an option to log in. After tapping the “Login” button for the first time, a newly registered user shall be redirected to the phone number verification page. This verification step will occur only once.
* The verification page shall contain a single data field labeled “OTP” and a “Verify” button.
* Upon successful form submission during registration, the system shall automatically send a six-digit OTP to the user’s registered phone number.
* The user must enter the received OTP into the OTP field on the verification page in order to verify their account.
* The OTP shall remain valid for 15 minutes from the time of generation.
* Once the user enters a valid OTP, they shall tap the “Verify” button.
* If the OTP is valid, the system shall display a confirmation pop-up stating:
* “Your account verification is successful. You will now be automatically redirected to your profile. Welcome to the donor club.”
* After displaying the message, the system shall automatically redirect the user to their profile/dashboard.
* If an invalid OTP is entered, the system shall show a warning message: “Invalid OTP” and display a “Resend OTP” button.
* By tapping the “Resend OTP” button, the user shall receive a new OTP on their registered phone number.
* In case of connectivity issues, the user shall have the option to receive the OTP via their registered email address.
* Once the OTP is successfully verified, the same success message and automatic redirection to the user profile shall apply.
* If the user input is wrong on any data field form will show error and it will give suggestions to the user to fix the error. After fixing the error the user will be able to submit the form.

**Requirement 1.1.9 (Accessibility and Responsiveness):**

* Must support and comply with WCAG 2.1 AA Compliance
* The platform must be fully responsive across all devices (desktop, tablet, mobile).

**Requirement 1.1.10 (Security Requirements):**

* Form must operate over TLS 1.3
* AES-256 encryption for data at rest
* CAPTCHA enforcement after 3 failed attempts

Note: The system shall allow all users to register for blood donation regardless of their donation history or hemoglobin levels. If a user doesn't meet the required hemoglobin threshold or hasn't waited for the mandatory four-month interval between donations, they will still be registered but automatically marked as "Not Eligible to Donate." In such cases, the system will immediately send notification and email to the user explaining their ineligibility status and clearly stating the specific reason - whether it's due to insufficient hemoglobin levels or not meeting the required waiting period since their last donation. This approach ensures that all potential donors can register in the system while maintaining proper eligibility tracking that aligns with blood donation safety standards. Apart from hemoglobin point and last donation date all data field of the form will be mandatory field.

**1.2 Login Module**

**Requirement 1.2 (Universal Login Support):**

The login form shall allow registered users to authenticate using any one of the three credentials provided at the time of registration: username, email address, or phone number.

* **Username Field:** The system shall accept a valid username, email address, or phone number as input for user identification. Any of these credentials, if previously registered, shall be valid for login.
* **Password Field:** The user shall enter the password that was created during the registration process.
* **Login Button:** The login form shall include a “Login” button to initiate the authentication process.
* Upon entering valid credentials (username/email/phone number and matching password), the system shall grant access to the user's profile/dashboard.
* If any credential is invalid or does not match the system records, the login attempt shall be rejected, and an appropriate error message shall be displayed, prompting the user to correct the input.

**Password Reset Option:**

* The login form shall include a “Forgot Password?” link or button.
* Upon clicking “Forgot Password?”, the user shall be prompted to enter their registered email address or phone number.
* The system shall send a password reset OTP to the provided email and phone number.
* The user shall be able to set a new password by entering the valid OTP, followed by a new password.
* Once the password is successfully reset, the user shall be able to log in using the updated credentials.

**Requirement 1.2.1 (Session Control and Security):**

* Limit login attempts to 5/hour per IP
* Apply CAPTCHA after 3 failed attempts
* Session expires after 15 minutes of inactivity
* No simultaneous sessions per user account

**Requirement 1.2.2 (Successful Login Redirection):**

* First-time logins shall redirect to the OTP verification option. Future logins shall redirect to the user dashboard.
* After re-setting the password system shall redirect user to the login page.

**1.3 Phone Verification Module**

**Requirement 1.3 (OTP-Based Phone Verification):**

* OTP: 6-digit numeric code, expires in 15 minutes
* Delivery: SMS first, fallback to email
* Max retries: 3. If exceeded, temporary block (10 minutes)
* Verified users shall be redirected to their dashboard
* Invalid OTP attempts should trigger warning and display "Resend OTP" option

**Requirement 1.4: User Profile Features**

The user profile shall contain the following core functionalities:

**1.4.1 Profile Photo & Cover Photo Management**

* **Profile Photo:**
  + Upload size: 180x180px (desktop), 128x128px (mobile).
  + Displayed on the left side of the profile to avoid covering the cover image.
* **Cover Photo:**
  + Display size: 820x312px (desktop), 640x360px (mobile).
  + The image will stretch to fit the full cover area.
* **Functional Requirements:**
  + Users can **add**, **update**, or **remove** both profile and cover photos at any time.
  + Only one photo per type (profile and cover) is allowed.
  + Both images are visible to other users and System Administrators.

**1.4.2 Bio Section**

* A text area shall allow the user to write a short bio.
* **Character limit**: 250 characters.
* Positioned below the profile picture.

**1.4.3 Modular Info Sections (Display + Edit)**

Modular layout sections shall include the following:

* **Work History**
* **Education**
* **Testimonials**
* **Blood Donation History**
* **Social Links**
* **Current City**
* **Hometown**

Each section will include:

* A display area visible to other users and admins.
* An **“Edit Details”** button that opens a modal popup for real-time data updates. (Design Reference Facebook Profile)

**Requirement 1.4.4: Edit Modal Functionality**

**Work Section**

* Users may add up to **three most recent work histories**.
* Includes:
  1. Company
  2. Position
  3. City/Town
  4. Description (Max 1000 characters)
  5. Time Period (From - To) with checkbox: *"I currently work here"*
* “Add Workspace” button to add new work entries.

**Education Section**

* Allows entry of **University/College** and **High School** details.
* Fields include:
  1. School/College/University Name
  2. Time Period (From – To)
  3. Graduated checkbox
  4. Course
  5. Description (Max 1000 characters)
  6. Type of Institution (High School / College / University via checkbox)
* “Add Education” button to add up to two more educational entries.

**Location Fields**

* **Current Town/City** and **Hometown** must be validated against the backends’ saved city list.

**Blood Donation History (LinkedIn-style chain)**

* Entry fields:
  1. Hospital Name *(autocomplete suggestion from database, e.g., Ibn Sina Hospital)*
  2. Hospital Location (e.g., Uttara, Dhaka)
  3. Date of Donation
  4. Type of Donation (Blood / Platelet / Others via circle checkbox)
  5. Upload Donation Picture *(optional)*

**Testimonial Section**

* Other users may submit testimonials with:
  + Character limit: **5000**
  + Media support: JPEG, PNG, MP4 (max size: **500 MB**)
  + Star rating (1–5 stars) **required** before submission
* Users **cannot delete** received testimonials.
* Users **can report** a testimonial.
* System Administrators and Managers **can remove** reported testimonials after review.

**Social Links**

* Acceptable platforms: **Facebook, Instagram, Twitter, GitHub, LinkedIn, Portfolio**
* Links redirect to external profiles in a new tab.

**Requirement 1.4.5: Profile Verification Workflow**

* **“Verify” button** next to “Edit Details”.
* Redirects user to a page explaining benefits of verification.
* Clicking “Get Verified” redirects user to the **Share ID** identity verification portal.
* Upon successful verification:
  + User receives a **red badge** next to their name.

**Requirement 1.4.6: Emergency Blood Request System**

**Trigger:**

* Users can launch **"Emergency Blood Request"** from their profile.

**Pop-Up Questionnaire:**

* **Hospital Info**:
  + Name, Street Address, City/Upazila, District, ZIP Code
* **Patient Info**:
  + Name, Relationship, Health Issue, Hemoglobin level, Platelet count
* **Donation Request Info**:
  + Blood Type, Quantity, Duty Doctor’s Name & Phone Number
* **Guardian Info**:
  + Name & Contact Number
* **Required Date/Time**
* **Document Upload** *(JPEG, PNG, PDF – mandatory)*
* **Automatic Classification:**
  + If the request is **within 3 days**, it's marked as *emergency*.
  + If **more than 3 days**, moved to *general request* queue.
* **Submission Conditions**:
  + All fields are **mandatory**.
  + Upon submission, emergency requests will be automatically listed on **admin dashboards**.

**Requirement 1.4.7 (Admin and Donor Sync):**

* Emergency requests must appear on Admin Panel
* Eligible donors (by location + blood type) must be notified
* Full audit trail must log all submissions

**Requirement 1.4.8: “Search for Donor” (Donor Search Tool)**

* **Search Interface (Airbnb-style)** with filters:
  + **Blood Group**, **District, City,** **Donor Type** (Eligible / All)
* **Instant Search** – results fetched and displayed from the backend database. On the search results all donor profile will show up as contact card/card style.

**1.4.8.1: Unified Donor Search Interface**

* The system shall provide a **single search form interface** accessible to both:
  + Registered users (logged-in)
  + General visitors (not logged-in)
* The form shall support the following filters:
  + **Blood Group**
  + **District**
  + **Eligibility Type** (Eligible Donors Only / All Donors)
  + **Required Date**

**1.4.8.2: Search Results Display**

* Upon clicking the **Search** button, the system shall retrieve donor profiles from the backend based on filter matches.
* The results will be displayed as **contact cards**, showing the following donor information:
  + Donor Name
  + Age
  + Total Times Donated
  + Average Rating (stars)
* Each contact card shall include a **CTA (Call-To-Action) button**, labeled: “**Get Donor Details**”

**1.4.8.3: Guest Visitor Behavior (Unauthenticated Access)**

When a **non-logged-in (guest) user** clicks on the CTA button:

* A **modal popup** shall appear with the following message:

“To get donor contact details, you must create an account or log in.”

* The modal will include a prominent **“Create an Account”** button.
* The modal will remain on-screen for **60 seconds**.
  + After 60 seconds, the user will be automatically **redirected to the registration form**.

**1.4.8.3.1 (Guest Search Limitations):**

Unregistered users may view donor name, age, donation count, and ratings. Clicking "Get Contact Details" prompts user registration.

**Requirement 1.6.2 (Pagination and Real-Time Refresh):**

* Default: 10 results/page
* Profile updates instantly affect search visibility

**1.4.8.4: Registered User Behavior (Authenticated Access)**

If a **logged-in user** clicks the CTA button:

* The system shall immediately present a **blood request form** (Form will Include data field: Patient Name, Patient Health Problem, Blood Group, Hemoglobin point, Why blood needed, how many blood unit needed, Hospital Location [Street address, City, District, Post Code).
* Upon submitting the form:
  + A secure message shall be sent to the donor.
  + Donor contact information becomes **partially visible** (e.g., phone number masked initially, fully visible after the donor response).

**Requirement 1.4.8.5: Messaging System (Template-Based)**

**Requirement 1. (Structured Message Templates):**

* Templates grouped by code (e.g., REQ\_INIT, DON\_AVAIL, DON\_NO\_SHOW)
* Blood seeker must complete short form before contacting donor
* Template messages shall include context placeholders: [Blood Group], [Location], [Time]

**Requirement 2 (Message Limits and Logging):**

* All messages are logged with metadata (sender, receiver, timestamp)
* Premade responses available for follow-up, cancellation, confirmation
* Abuse reporting available for each message

Each message is tagged with a **code** and grouped into categories:

| **Category** | **Message Code** | **Example Template** |
| --- | --- | --- |
| Initial Request | REQ\_INIT | “Dear donor, this is a request for [Blood Group] blood at [Location] on [Date] at [Time]. Please respond if you’re available.” |
| Donor Available | DON\_AVAIL | “Yes, I am available to donate at [Time] on [Date].” |
| Donor Unavailable | DON\_UNAVAIL | “Sorry, I’m unable to donate at this time.” |
| Blood Seeker Confirmation | SEEK\_CONF | “Thanks! Please reach [Hospital Name] by [Time, Date].” |
| Follow-Up Before Donation | FOLLOW\_UP | “Are we still on for the donation today?” |
| Donor No-show | DON\_NO\_SHOW | “Donor didn’t show up at the promised time.” + manual note field (Max 2000 characters) |
| Donation Confirmation | DON\_SUCCESS | “Blood was donated as planned.” *(Prompts testimonial pop-up)* |
| Cancel Request | REQ\_CANCEL | “No longer need a donor, thank you.” |

**1.4.8.6: Search Pagination System**

The search result view shall include pagination support with the following requirements:

* **Pagination Controls**:
  + First Page
  + Previous Page
  + Next Page
  + Last Page
* **Results Count Display**:
  + Format: "1–10 of 1248 results"
* **Page Size Selector**:
  + Dropdown allowing the user to choose number of donor cards per page:
    - Options: 10, 25, 50, 100
* **URL Parameters**:
  + The current search filters and page shall be **preserved in URL parameters** for sharing and reloading consistency.

**1.4.8.7: System Constraints**

* Maximum allowed search result: 5000 entries (to protect performance).
* Throttling to be implemented for **unauthenticated user searches** (e.g., 10 searches per IP/hour).
* All search results are subject to **real-time availability and eligibility filter logic** on backend.

**Requirement 1.4.8: Privacy & Security**

* Users shall have access to a **Settings** section to:
  + Manage privacy
  + Enable/disable 2FA (Two-Factor Authentication)
  + Change Password
  + Set blood donation availability (Available / Not Available)
* A **logout** option must be available at all times on the user profile.
* Backend session management shall be implemented to prevent unauthorized access.

**Requirement 1.4.9 (Real-Time Sync):**

Any profile updates shall reflect instantly on:

* Designated section of the profile
* Search listings
* Admin dashboard
* Blood request match engine

**Section 2: System Architecture and Non-Functional Requirements**

**2.1 System Architecture Overview**

**Requirement 2.1 (Architecture Style):**

The system should be built using a microservices-ready architecture that supports scalability, modularity, and fault isolation. It shall expose public RESTful APIs that allow for interaction between modules and external systems.

**Recommended Technology Stack**

- Frontend: Html, CSS, JavaScript, React.js

- Backend: Node.js (Express) or Django

- Database: SQL

- Auth: JWT + OAuth2

- Hosting: AWS/Azure

**Requirement 2.1.1 (Service Types):**

* **User Service**: Handles authentication, registration, profile updates
* **Donation Service**: Manages donation records and eligibility
* **Notification Service**: Sends email, SMS, and in-app alerts
* **Search Service**: Powers donor lookup with real-time indexing
* **Admin Dashboard Service**: Processes analytics and moderation tools
* **Media Service**: Handles document and image uploads

**Requirement 2.1.2 (Data Flow):**

All services shall communicate over HTTPS and expose secure REST APIs. Real-time updates between profile, search, and admin dashboard shall be handled via a publishing event model (e.g., Kafka or Redis Streams).

**Requirement 2.1.3 (API Gateway):**

An API gateway shall manage routing, rate limiting, and authentication between client and backend services.

**2.2 Security and Privacy**

**Requirement 2.2 (Encryption):**

* All PII and health data at rest shall be encrypted using AES-256
* All in-transit communication shall be used TLS 1.3

**Requirement 2.2.1 (Authentication & Authorization):**

* OAuth 2.0 or JWT for secure session tokens
* Role-based access control (RBAC) must be enforced across all APIs
* Admin users must enable two-factor authentication (2FA)

**Requirement 2.2.2 (Data Privacy Compliance):**

* The system shall comply with GDPR and local data regulations
* All users shall have the right to request deletion or anonymization of their data
* Consent shall be collected explicitly for data processing

**2.3 Performance and Reliability**

**Requirement 2.3 (Response Time and Load Handling):**

* 95% of API requests must respond within 300ms
* The system shall handle a minimum of 500 concurrent users with horizontal scalability

**Requirement 2.3.1 (Auto-Scaling):**

All critical services shall auto-scale based on CPU/memory usage or API request rate using Kubernetes or equivalent orchestration.

**Requirement 2.3.2 (Failover Mechanism):**

* Services must support active-passive failover for high availability
* Health checks and automatic restart policies shall be in place for all containers

**2.4 Logging, Auditing, and Monitoring**

**Requirement 2.4 (Centralized Logging):**

* All system logs (auth, requests, errors) must be pushed to a centralized logging service (e.g., ELK Stack)

**Requirement 2.4.1 (Audit Trail):**

* Every critical action (e.g., donor record update, emergency request, message sent) shall generate an audit entry with user ID, timestamp, and IP

**Requirement 2.4.2 (Monitoring & Alerts):**

* The system shall support uptime monitoring, performance metrics, and alerting for API latencies or service failure (e.g., Prometheus + Grafana)

**2.5 Accessibility and Compliance**

**Requirement 2.5 (Accessibility Standards):**

* The frontend must meet WCAG 2.1 AA standards
* All buttons, forms, and modals must be keyboard accessible and screen reader friendly

**Requirement 2.5.1 (Multi-Language Readiness):**

* The UI shall support localization and dynamic language switching for at least English and Bangla

**2.6 Deployment and DevOps**

**Requirement 2.6 (CI/CD Pipeline):**

* The system must be deployed using an automated CI/CD pipeline supporting staging and production environments

**Requirement 2.6.1 (Source Control):**

* All code must be maintained in a secure Git repository (e.g., GitHub or GitLab)
* Pull request reviews and automated test runs shall be mandatory before merge.

**Requirement 2.6.2 (Backups):**

* Encrypted backups of databases shall occur daily and be stored for 30 days
* Backup integrity checks shall run weekly

**Section 3: Admin Dashboard Requirements**

**Requirement (Dashboard Structure):**

The admin dashboard shall present a modular interface organized into 8 main sections:

1. Dashboard (Analytics Overview)
2. Donor Management
3. Inventory
4. Scheduled Blood Drive
5. Emergency Blood Request
6. Analytics
7. Reactivation Request
8. Verification Request
9. Settings
10. Logout

**3.1 Dashboard Analytics Widgets**

**Requirement 3.1.1 (Total Donors):**

* **Display**: Numerical count of total registered donors (lifetime)
* **Optional**: Trend indicator (e.g., % increase from last month)
* **Click Action**: By clicking on the widgets, it will Opens detailed donor list with filters/sorting

**Requirement 3.1.2 (Total Blood Donations):**

* **Display**: Total number of blood donations recorded (lifetime)
* **Optional**: Monthly/yearly trend comparison
* **Click Action**: By clicking on the widgets, it will Opens full donation record view with filtering by date, type

**Requirement 3.1.3 (Critical Blood Levels):**

* **Display**: List of blood groups below 50 eligible donors
* **Threshold Logic**: Any blood group <50 active, eligible donors is marked critical
* **Visual**: Color-coded indicator (Red = Critical, Yellow = Low, Green = Healthy)
* **Click Action**: By clicking on the widgets, it will Navigates to blood type report with drill-down statistics

### **Requirement 3.1.4 (Donors Ready to Donate):**

* Numerical count of registered donors who are **currently marked as available to donate blood and donors who donated blood 4 months ago**
* This count is based on donors who have toggled their donation status to **“Available”** in their privacy/settings section and their last donation date
* Include a **trend indicator** (e.g., "% increase/decrease from the last 30 days.
* **Click Action**:
  + When clicked, the widget opens a **filtered donor list** view (On donor management option) showing only:
    - Donors who are currently eligible and marked available
    - Donors who have donated blood 4 months ago
    - Filtering options: Blood group, location, gender, last donation date
    - Sorting options: Alphabetical, most active, last donation date
* **Tooltip (on hover)**:
  + “Shows total number of donors who are currently ready and eligible to donate blood.”

**Requirement 3.1.5 (Pending Emergency Requests):**

* **Display**: Count of unfulfilled emergency blood requests
* **Click Action**: Opens emergency request queue with request details

**Requirement 3.1.6 (Scheduled Blood Donations):**

* **Display**: Total number of upcoming scheduled donation appointments
* **Click Action**: Opens schedule management and contact overview

**Requirement 3.1.7 (Duplicate Data Alert):**

* **Display**: Count of flagged duplicate or suspicious donor profiles
* **Logic**: Matched by name, DOB, blood group, hemoglobin, or location (2+ criteria)
* **Click Action**: Opens review queue in Donor Management section

**Requirement 3.1.8 Graph and Charts**

The system shall present demographic statistics visually through dynamic charts to help administrators analyze donor population trends effectively. The following charts will be implemented:

**Requirement A: Blood Group Distribution – Pie Chart**

* The system shall display the percentage distribution of registered donors by blood group in a pie chart format.
* Each slice represents a different blood group (e.g., A+, A−, B+, AB+, O−, etc.).
* The chart shall display percentage labels and a legend for clear identification.
* Admin shall be able to export the chart as PNG, PDF, and CSV (tabular breakdown).  
  Data shall update in real time based on the current donor database.

**Requirement B: Gender Distribution – Bar Chart**

* The system shall display a vertical bar chart comparing the number of male, female, and other gender identities among registered donors.
* Bars shall be labeled clearly and optionally colored distinctly.
* The chart shall include tooltips showing exact count and percentage.
* A toggle shall be available to filter by blood group or city.
* Export functionality (PNG, PDF, CSV) shall be available.

**Requirement C: Population Pyramid – Age & Gender Distribution**

* The system shall display a population pyramid to visualize the age distribution of donors by gender.
* The chart shall split genders into two sides:
* Left (Male)
* Right (Female)  
  Age groups (e.g., 18–24, 25–34, etc.) will be represented on the Y-axis, and donor count on the X-axis.
* The chart shall support filters by location (district, city), blood group, and availability status.
* Tooltip interaction and export to PNG/PDF/CSV shall be supported.

**Requirement D: Age-Sex Distribution – Line/Grouped Bar Chart**

* The system shall provide an alternate Age-Sex comparison chart using:
* Grouped bar chart (Male vs Female counts per age group), or
* Line chart (overlayed gender lines across ages).
* The X-axis shall represent age groups or single-year ages, and the Y-axis shall show donor count.
* This chart is intended for detailed tracking and comparison of age-gender demographics over time.
* Filters and export options will be identical to other charts.

**Requirement E: Chart Filters and Interactivity**

* All charts shall allow filtering by:
* Date range (e.g., registered this year)
* Location (District, City)
* Blood group
* Verified/unverified donors
* Available/unavailable donors
* Charts shall refresh automatically based on selected filters without reloading the page.
* Tooltips, legends, and hover effects shall enhance usability.

**3.2 Donor Management Requirements**

**Requirement 3.2.1 (Donor Registration)**

Donors shall be creating an account and registered through the registration form. For details see User Registration Module

**Requirement 3.2.2 (Data Collection):**

1. The system shall allow administrators to register new donors individually or in bulk via spreadsheet upload. All required fields from the public registration form must be present. System-generated Donor IDs shall follow the format MEDON-YYYY-XXXX.

2. Donor’s data will also be collected through registration form. Upon successful completion of registration form donor data shall be added to the database as well as on the donor list of admin dashboards. Donor registration data shall be updated real time on the database and on the all preferred sections of the admin dashboard.

**Requirement 3.2.3: Manually Entered Donor Profile Handling**

**Requirement A: Inclusion in Search Results**

* The system shall include **manually entered donor profiles** in search results alongside user-created profiles.
* These profiles shall be filterable by standard criteria (e.g., blood group, district, availability).
* Displayed in a profile card format, showing:
  + Donor Name
  + Age
  + Blood Group
  + Number of Donations
  + Rating (if available)
  + Label/tag: **“System Upload”**

Note: For system verified donor profile system should show a disclaimer that contains the source of these data, and these data may outdated and these are collected from different sources. So, if there is any wrong information system owners are not responsible for any type of incident.

**Requirement B: Unique Donor ID Format for Manual Entries**

* Manually entered donors shall be assigned a system-generated unique Donor ID using the format:
  + **MEDON-YYYY-XXXX**
    - YYYY: 4-digit year of entry
    - XXXX: 4-digit sequence number
* Example: MEDON-2025-0017
* System shall validate uniqueness and auto-increment the ID sequence.

**Requirement C: Profile Access Restrictions**

* Manually entered donor profiles:
  + **Cannot log in** or access the dashboard by default
  + **Cannot update information themselves**
  + Are **read-only to the public and editable only by Admin**
  + Will be marked internally as System-Created for audit purposes

**Requirement D: Claim This Profile Functionality**

**Claim Prompt**

* Each manually created donor profile card shall include a **“Claim this Profile”** button, visible to general users and registered users.

**Claim Form Requirements**

* When a user clicks “Claim this Profile,” a modal or new page shall open with a verification form that requires:
  + Full Name
  + Phone Number
  + Email Address
  + Date of Birth (must match admin record)
  + Optional: Blood Donation Record Upload (PDF/JPEG/PNG, max 5MB)
  + One Government-Issued ID (NID/Passport/Driving License – JPEG/PDF)

**Verification Workflow**

* Submitted claims shall be reviewed by an admin or verification team.
* Upon Submitted claim the claim shall listed real time on the Request Verification section of the admin dashboard for admin review.
* Upon submission of any **verification claim** (e.g., **Profile Claim Verification** or **Red Badge Verification**), the claim shall:
* Be **instantly listed** in the **"Request Verification"** section of the **Admin Dashboard**.
* Appear in real-time **without page reload**, using live data binding or websocket-based sync.
* Show a **notification badge/alert** to the admin when a new verification request is received.
* **Each Claim Entry Must Include:**
* System generated Unique Claim ID formatted as: CLAIM-MEDON-YYYY-Profile ID
* User’s Full Name and User ID
* Type of Verification Request:
* PROFILE\_CLAIM
* RED\_BADGE\_VERIFICATION
* Submission Timestamp (Date & Time)
* Request Status (Pending, Approved, Declined)
* Action Buttons: View Details, Approve, Decline
* Quick Link to User Profile
* **Filtering & Sorting:**

Admin must be able to:

* Filter requests by:
* Verification Type (PROFILE\_CLAIM, RED\_BADGE\_VERIFICATION)
* Status (Pending, Approved, Declined)
* Date Range (Lifetime, Yearly, Monthly, Weekly)
* Sort by:
* Submission Date (Newest/Oldest)
* User Name (A-Z/Z-A)
* Verification Type
* **Admin Actions:**
* **Upon admin Approval of the Request:**
* Instantly mark the request as **approved**.
* If RED\_BADGE\_VERIFICATION, automatically display **red verified badge** next to user’s name on their profile.
* Send **generic congratulatory email** to the user:
* *"Congratulations! Your verification request has been approved, and your profile now carries verified status."*
* **Decline Request:**
* Admin must **enter a mandatory explanation note** before declining.
* Status changes to **declined**.
* User receives an email with subject:

*"Verification Request Declined – Explanation from Admin"*

* Email body will include:
* Timestamp
* Decline reason (entered by admin)
* **Email Behavior:**

**All verification emails** must include:

* Username
* Reference to the type of verification request
* Direct contact/support link
* System-generated Claim ID
* **Upon approval:**
  + The user will be linked to the claimed donor profile
  + The temporary login credentials will be emailed to the user by the system
  + User shall be able to login by using the temporary login credentials
  + User shall be able to reset their temporary login credentials
  + The profile status will be updated to **“Claimed & Verified”**
  + The Donor ID (MEDON format) will be retained
* If rejected:
  + The user will receive a system message with reasons and retry instructions

**Notification System**

* Both the claimant and admin shall receive system alerts/emails:
  + On form submission
  + Upon approval or rejection
  + If additional verification documents are needed

Note: After verification, the user can update personal info, excluding Name, Date of Birth, Blood Group via the Profile Edit modal.

**Requirement 3.2.4: Admin Controls**

* Admins shall have a dedicated interface to:
  + Add/Edit/Delete manual donor entries
  + View all claim requests and their statuses
  + Approve or reject claims
  + Merge a claimed profile with an existing user account (if a duplicate exists)
  + Mark profiles as "Inactive" or "Deceased" if needed

**Requirement 3.2.5 (Data Validation):**

Upon manual or bulk entry, the system shall validate:

* Email format and uniqueness. Upon missing email [noemail@bdms.com](mailto:noemail@bdms.com) will be acceptable
* Date fields (DOB, last donation)
* Presence of all mandatory fields
* Blood group validity
* Phone number structure and uniqueness

**Requirement 3.2.6 (Eligibility Check):**

The system shall calculate donor eligibility based on:

* Age ≥ 18
* Weight > 50kg
* Hemoglobin within accepted range
* ≥ 120 days since last donation

If ineligible, the donor shall be marked as "Not Eligible" with reason provided.

**Requirement 3.2.7 (Data Storage):**

All donor data must be encrypted at rest, stored in a secure, GDPR-compliant manner.

**Requirement 3.2.8(Acknowledgement):**

System shall confirm successful entry with a success message and donor ID.

**Requirement 3.2.9 (Donor Listing and Overview)**

**Requirement 3.2.9.1 (Tabular View with proper filtering option):**

The admin view shall show donors in tabular format including filtering/sorting and search option for following:

* Donor ID
* Name
* Blood Type
* Email & Phone
* Last Donation Date
* Status (ACTIVE / DEFERRED / INACTIVE)
* Action Buttons: View/Edit, Send Message, Send Email

**Requirement 3.2.9.2 (Pagination):**

* Admin can select how many entries to view per page (default: 10)
* Show page number and total pages (e.g., Page 2 of 18)
* Navigation: First, Previous, Next, Last buttons

**Requirement 3.2.9.3 (Selection & Bulk Action):**

Admins shall be able to select multiple donor rows for bulk operations (see 3.3.6).

**Requirement 3.2.9.4 (Donor Actions)**

* **View/Edit Donor:** Admin shall be able to open a detailed modal to update any donor field with real-time validation.
* **Contact Donor:** Admin shall contact donors via email, phone, or in-system messaging. Contact options must adhere to user privacy preferences.
* **Deactivate/Delete Donor:** Admin can deactivate a donor instead of deleting. Deleted records remain archived for audit. Deactivated users will be notified via email and must contact admin for reactivation.

**Requirement 3.2.9.5 (Data Export)**

**Export Format Support:**

Admins shall export full or filtered donor data in:

* CSV
* Excel (XLSX)
* PDF

**Export Filters:**

All filters applied to the list (e.g., by blood type) shall carry over to exports.

**Requirement 3.2.9.6 (Sorting & Filtering)**

**Sortable Columns:**

Admin shall sort by:

* Donor ID
* Name
* Blood Type
* Last Donation Date
* Status

**Filtering Criteria:**

Admin shall be able to filter by:

* Blood Type
* Donation Status
* Last Donation (range)
* Donor Name

**Requirement 3.2.9.7 (Bulk Operations)**

**Available Actions:**

When donors are selected, bulk actions shall include:

* Change status
* Send mass email
* Export selected donors
* Deactivate selected profiles
* Delete selected profile

**User Deactivation Handling:**

* Upon deactivation of a user profile, the system shall automatically send an email to the affected user.
* The email shall include:
  + The reason for deactivation (selected/admin-entered at time of deactivation).
  + A button labeled "Request to Reactivate".
  + Clear instructions on how the user can proceed to get their account reactivated.

**Requirement 3.2.9.8 Reactivation Request Flow:**

* Request to Reactivate button shall redirect the user to a secure reactivation request form.
* The Reactivation Request Form shall include the following fields:
  1. Full Name (auto-filled if logged in)
  2. Registered Email Address (auto-filled)
  3. Phone Number (optional, editable)
  4. Why was your account inactive? (textarea, 1000 characters max)
  5. Why should we reactivate your account? (textarea, 1000 characters max)
  6. Optional: Upload Supporting Document (PDF, JPEG, PNG – Max size: 10MB)
* **Upon submission:**
  1. The request shall be stored in the database.
  2. The request will be visible to admins in the “Reactivation Requests” section of the admin dashboard.

**Requirement 3.2.9.9 Admin Dashboard - Reactivation Requests Section:**

* Admins shall view all reactivation requests in a tabular format:
  + Columns: Request ID, User Name, Email, Date Submitted, Status (Pending, Approved, Declined), Admin Note (if any), Action Buttons (Approve, Decline)
  + Ability to view full request details (including uploaded files and reasons)
* Admin Actions:
  + Approve:
    - Account is reactivated instantly
    - System sends a congratulatory email to the user (template-based)
  + Decline:
    - Admin must provide a reason (textarea, required)
    - System sends an email to the user with the admin’s reason for declining

**Requirement 3.2.9.10 Email Templates:**

* Reactivation Request Submitted:
  + Confirmation to user that their request has been received
  + Estimated response time (e.g., 2–5 business days)
* Reactivation Approved:
  + Subject: "Your Blood Donation Profile Has Been Reactivated!"
  + Body: Congratulations message and encouragement to update profile/donation status
* Reactivation Declined:
  + Subject: "Your Reactivation Request Was Declined"
  + Body: Admin explanation + option to contact support

**Requirement 3.2.9.11 Admin Filters and Sorting Features:**

* Admin shall be able to:
  + Filter requests by:
    - Time Range: Lifetime, Yearly, Monthly, Weekly
    - Status: Approved, Declined, Pending
  + Sort by:
    - Date of request
    - User name/email
    - Approval/Decline date

**3.3 Inventory Management Requirements**

**Requirement 3.3.1 (Blood Donation Recording):**

The system shall allow administrators to record details of each donation as part of the inventory process.

* **Fields Required**: Donor ID, Date of Donation, Time of Donation, Hospital/Location, Volume Donated, Donation Type (Whole Blood, Platelet, Plasma)
* **Validation**: Date and time fields must reflect actual timestamp; donor ID must exist and be eligible
* **Auto-Linking**: Each recorded donation shall automatically update the donor’s profile history and mark the donor as ineligible for 90 days

**Requirement 3.3.2 (Blood Unit Tracking):**

The system shall track the availability of blood units per blood group across time.

* **Real-Time Counter**: Dashboard shall show real-time available units per group (e.g., A+, B-, AB-)
* **Data Sync**: Every donation recorded shall increase the available count for that blood type; every issuance shall decrease it
* **Critical Alerts**: If stock for any group falls below defined threshold (e.g., <10 units), mark as critical and notify admin

**Requirement 3.3.3 (Blood Issuance Recording):**

The system shall provide functionality to record issuance of blood units.

* **Fields Required**: Date of Issue, Time, Recipient (Patient ID or Name), Blood Group, Units Issued, Hospital Name, Purpose/Notes
* **Validation**: Units issued must not exceed available stock
* **Audit Logging**: Every issuance shall generate a log entry with admin ID, timestamp, and change in stock

**Requirement 3.3.4 (Blood Request Status Tracking):**

The system shall track all internal blood requests from the emergency or scheduled donation modules.

* **Statuses**: Pending, Approved, In Progress, Fulfilled, Rejected
* **Tracking**: Admin dashboard shall show a list of active requests with status updates
* **Search & Filter**: Admin can filter requests by status, requestor name, hospital, or date

**3.4 Scheduled Blood Drive Requirements**

**Requirement 3.4.1 (Blood Drive Creation and Tracking):**

The system shall allow administrators to schedule and manage both one-time and recurring blood donation drives.

* **Fields Required**: Drive Title, Date, Time, Venue (with address), Organizer Contact, Targeted Blood Groups (optional), Description, Registration Deadline, Donation Goal (units or donor count)
* **Validation**: Time and date must not be in the past; location must be searchable via integrated map if enabled
* **Auto-Population**: The system shall prefill donor info if a drive is tied to previously approved users

**Requirement 3.4.2 (Donor Group Listings):**

* If multiple donors are scheduled for a specific drive:
  + Group them by date/time and drive title
  + Provide overview cards with names, blood types, and status (e.g., confirmed, pending)
* Individual donor listings for one-off entries should remain as single-row records

**Requirement 3.4.3 (Drive Modification and Cancellation):**

* Admins shall edit details of any scheduled blood drive unless it is within 24 hours of start time
* Admins can cancel drives with notification to all affected donors
* Notification channels: Email, SMS, In-System Alert

**Requirement 3.4.4 (Fraud and Abuse Tracking):**

The system shall keep a record of users involved in multiple or frequent group bookings.

* **Track Fields**: Donor ID, Number of Drives Booked, Units Requested
* **Flag Logic**: Auto-flag users with 5+ bookings/month or 3+ requests with no follow-through

**Requirement 3.4.5 (Drive Metrics and Reporting):**

Each drive shall include:

* Total number of registered donors
* Number of fulfilled donations
* Units collected
* Units issued from that drive (if applicable)
* Option to export report in PDF or Excel

**3.5 Emergency Blood Request Requirements**

**Requirement 3.5.1 (Request Submission):**

The system shall provide a structured emergency blood request form accessible from both user profiles and admin dashboard.

* **Fields Required**: Hospital Name, Hospital Address (Street, Upazila, Zilla, Zip Code), Patient Name, Relation to Patient, Health Condition, Hemoglobin Level, Required Blood Type and Quantity, Duty Doctor Name & Contact, Patient Guardian Name & Contact, Required Date & Time, Medical Documents (JPEG, PNG, PDF)
* **Validation**: All fields mandatory. Date must not be in the past. Files must not exceed 5 MB. Document upload required.
* **Auto-Classification**: Requests needing blood within 3 days are flagged as EMERGENCY and auto-routed for urgent handling.

**Requirement 3.5.2 (Verification and Fraud Prevention):**

* The system shall validate uploaded documents using MIME type and file size.
* The system shall log IP addresses, device info, and user ID at time of request.
* Requests shall be flagged if the same user submits more than 3 emergency requests within 7 days.

**Requirement 3.5.3 (Request Routing & Notification):**

* Emergency requests shall trigger instant notifications to:
  + Admins (via dashboard and email)
  + Nearby eligible donors (via SMS and in-app alert)
* Donor matching logic includes:
  + Blood group compatibility
  + Distance from hospital location
  + Donation eligibility (time since last donation)

**Requirement 3.5.4 (Admin View and Status Management):**

Admins shall view all emergency requests in a sortable and filterable table.

* **Statuses**: New, Verified, In Progress, Fulfilled, Cancelled, Rejected
* **Filters**: Status, Request Date, Hospital, User ID, Blood Type
* **Bulk Actions**: Approve, Reject, Assign Donor

**Requirement 3.5.5 (Real-Time Updates):**

* As status changes, updates must be instantly reflected in:
  + Admin dashboard
  + Requestor’s profile history
  + Notification center for all assigned parties

**Requirement 3.5.6 (Audit and Reporting):**

* Every emergency request shall be logged with:
  + Timestamp
  + User and admin action history
  + Final resolution status
* Admins can export emergency request logs as PDF or CSV for auditing

**3.6 Analytics Requirements**

**3.6.1 Donor Analytics**

* **3.6.1.1 Demographics Analysis**  
  The system shall generate donor demographic reports including:
  + Age distribution presented via histograms or age brackets (e.g., 18–25, 26–35)
  + Gender breakdown will be shown on bar graphs
  + Geographic distribution mapped by administrative boundaries (district, upazila) using interactive maps

**Data Refresh:** Reports shall update in near real-time with a maximum delay of 5 minutes after new data entry.  
**Filtering:** Admins shall filter demographic reports by date ranges, donor status, and blood group.  
**Data Security:** Demographic reports shall exclude personally identifiable information (PII) to comply with GDPR and privacy policies.

* **3.6.1.2 Donor Behavior Analysis**  
  The system shall track and visualize:
  + Donation frequency per donor over customizable periods
  + Registration trends (daily, monthly, yearly) with comparison to previous periods
  + Donor retention metrics including re-donation rates within recommended intervals (e.g., 90 days)  
    **Visualizations:** Line charts, heatmaps, and cohort analysis dashboards.  
    **Export:** Reports can be exported in CSV, PDF, and Excel formats.  
    **API Access:** Analytics data shall be available via secure REST API endpoints for external BI tools.

**3.6.2 Donation Analytics**

* **3.6.2.1 Volume Tracking**  
  The system shall provide aggregate donation volume statistics over selectable timeframes (daily, weekly, monthly, yearly).  
  **Real-Time Dashboard:** Key volume metrics shall refresh every 1 minute on the admin dashboard.
* **3.6.2.2 Distribution Breakdown**  
  Detailed donation counts shall be available by:
  + Blood group (A+, O-, etc.)
  + Location (district/upazila)  
    Reports shall highlight high-demand blood types and regions with shortages.  
    **Alerts:** Automatic notifications shall trigger when donations for a blood group fall below defined thresholds.
* **3.6.2.3 Trend Analysis**  
  The system shall detect seasonal trends, spikes, or drops in donation activity using time series analysis.  
  **Machine Learning Integration (Optional):** System may integrate predictive analytics to forecast donation needs and optimize campaigns.

**3.6.3 Blood Inventory Analytics**

* **3.6.3.1 Inventory Level Trends**  
  Historical inventory data shall be plotted over time to identify stock fluctuations.  
  **Retention:** Inventory data shall be retained for at least 5 years for compliance and auditing.
* **3.6.3.2 Usage Patterns**  
  The system shall correlate blood issuance and expiry rates to optimize inventory management.

**3.6.4 Key Metrics Dashboard**

* **3.6.4.1 Total Donors Widget**  
  Displays total donor count with percentage change versus previous period (configurable: day, week, month, year).  
  **Drill-Down:** Clickable to open detailed donor lists with filtering capabilities.  
  **Performance:** Widget shall load data within 2 seconds even with 100,000+ donor records.
* **3.6.4.2 Total Donations Widget**  
  Displays total donations count, supporting dynamic filtering by time and blood group.
* **3.6.4.3 Critical Stock Alert Widget**  
  Color-coded indicator for blood groups with inventory below thresholds:
  + Red: Critical (<10 units)
  + Yellow: Low (10–20 units)
  + Green: Sufficient (>20 units)
  + Widget refresh rate: update data real time

**3.6.5 Blood Inventory Status Details**

* **3.6.5.1 Blood Type Distribution Chart**  
  Interactive pie or bar charts showing percentage share of each blood group in current inventory. Filterable by date/time/blood group/location range.
* **3.6.5.2 Per-Type Status Indicators**  
  Each blood group shall have a status badge: Critical, Low, Good, or Rare (for rare types).  
  Alerts must be push-notified to administrators when critical or rare statuses appear.
* **3.6.5.3 Inventory Count & History**  
  Detailed list view of units available per blood group with timestamps of last update.  
  Include audit trail for stock changes (donation, issuance, expiry).

Note: The analytics section of the admin dashboard shall include required bar chart, Pie Chart, Population Pyramid, Age-Sex chart and charts for demographics these chart shall do real time synchronization with the all chart on the admin dashboard.

Also the analytics section should have the feature and functionality to export data in Excel, PDF, PNG, and JPEG format.

The analytics section should provide insights about the following:

1. The total registered user ([including eligible, not eligible, active and de-active donor]Lifetime, Yearly, Monthly, Weekly, Daily and custom period of date)

2. Total Eligible donor (Defined by applying eligibility criteria 4 month interval from last donation, Weight 50kg, Hemoglobin point meets the eligibility criteria.)

3. Total Not eligible donor (Defined by eligibility criteria)

4. Total active donor (People who visits their profile at least once a month, information is updated)

5. Total deactivated donors (Doesn’t visit their profile, de activated by the system Admin due to some reason)

6. Total available donor (Active and Eligible)

7. Total blood unit available ( 1 Unit per user. So it will count all eligible and not eligible user)

8. Analytics should include registration data (Total registration completed Lifetime, Yearly, Monthly, Weekly, Daily and custom period of date)

9. Analytics should also include data about the people who has deleted their profile.

10. Analytics should also be include statistics about the visitor on the site (Lifetime, Yearly, Monthly, Weekly, Daily and custom period of date)

11. Analytics should also include data about the total number of blood has been donated from the system. This data should be filterable by Age-sex, Demographics and data should be represented in tabular format and graph chart style.

12. Analytics should include data about scheduled donation (Allow admin to detail view of all scheduled donation)

13. Analytics should also include the number of failed schedules like blood seeker scheduled donation, but the donor didn’t show up or donation cancelled due to some other reason. The system should include detailed report, notes and analytics.

14. These data should be displayed on the Admin Dashboard as proper graph chart as widgets. Any changes or update on the system should be updated on all required field or sections immediately

**Security & Compliance Notes:**

* All analytics data transmissions shall use TLS 1.3 encryption.
* Data access is restricted via Role-Based Access Control (RBAC) to admin users.
* User interface components shall meet WCAG 2.1 AA accessibility standards.
* Logs of analytics report views and exports shall be maintained for auditing.

**3.7 Reactivation Request**

* Admins shall view all reactivation requests in a tabular format under Reactivation Request section of the admin dashboard:
  + Columns: Request ID, User Name, Email, Date Submitted, Status (Pending, Approved, Declined), Admin Note (if any), Action Buttons (Approve, Decline)
  + Ability to view full request details (including uploaded files and reasons)
* Admin Actions:
  + Approve:
    - Account is reactivated instantly
    - System sends a congratulatory email to the user (template-based)
  + Decline:
    - Admin must provide a reason (textarea, required)
    - System sends an email to the user with the admin’s reason for declining

For details see Requirement 3.2.9.9 Admin Dashboard - Reactivation Requests Section:

**3.8 Verification Request**

All Profile claim requests and Red badge requests should be listed on this designated section. See verification Workflow for more information.

**3.9 Settings Requirements**

**3.9.1 Profile Management**

* **3.9.1.1 View/Edit Profile:**  
  Administrators shall be able to view and update their profile information, including:
  + Full name
  + Email address (with format and uniqueness validation)
  + Phone number (validated per country format)
  + Profile picture upload (JPEG/PNG, max 5MB, cropped to 180x180 pixels)  
    **Real-Time Validation:** All fields must validate on input with immediate feedback.  
    **Data Security:** Profile updates shall be transmitted over TLS 1.3 and stored with AES-256 encryption at rest.

**3.9.2 Password Management**

* **3.9.2.1 Change Password:**  
  Admins can change their password, requiring:
  + Entry of current password
  + New password meeting complexity: minimum 12 characters, mix of uppercase, lowercase, digits, and special characters
  + Confirmation of new password  
    Password strength meter shall guide admins.  
    **Password Storage:** Passwords must be securely hashed using bcrypt or Argon2.  
    **Failure Handling:** After 5 failed attempts, account locks for 15 minutes.

**3.9.3 Two-Factor Authentication (2FA)**

* **3.9.3.1 Enable/Disable 2FA:**  
  Admins can enable or disable 2FA via:
  + Authenticator apps (TOTP, e.g., Google Authenticator)
  + SMS-based OTP (as backup)  
    Enabling 2FA requires confirmation via OTP.  
    **Session Policy:** Once enabled, 2FA is required for all future logins.  
    **Recovery:** Admins can generate backup codes.

**3.9.4 Notification Preferences**

* **3.9.4.1 Manage Notifications:**  
  Admins can configure preferences for receiving notifications via:
  + Email
  + SMS
  + In-app push notifications  
    Notification categories include:
  + Emergency blood requests
  + Scheduled blood drives reminders
  + System alerts and critical inventory levels

**3.9.5 Dashboard Layout Customization**

* **3.9.5.1 Widget Arrangement:**  
  Admins shall be able to rearrange dashboard widgets by drag-and-drop interface.
  + Widget visibility toggles to show/hide specific widgets
  + Save and reset layout options  
    **Persistence:** Layout settings shall be saved per user in the database.

**3.9.6 Default Start Page**

* **3.9.6.1 Set Home Page:**  
  Admins can select which dashboard section appears after login (e.g., Dashboard, Donor Management, Inventory).

**3.9.7 Login History and Audit Logs**

* **3.9.7.1 View Login History:**  
  The system shall display admin’s recent login attempts, showing:
  + Date and time
  + IP address
  + Device/browser info
  + Login success/failure status
  + Geolocation approximate data (country/region)
* **3.9.7.2 Audit Logging:**  
  All profile changes, password changes, and 2FA status updates must be logged with timestamps and admin ID.

**3.10 Logout Requirements**

**3.10 .1 Secure Logout**

* **3.10 .1.1 Logout Action:**  
  Users and administrators shall have a clearly visible logout button in the UI.  
  Clicking it shall:
  + Terminate the user session on server and client side
  + Clear authentication cookies and tokens securely

**3.10 .2 Session Termination and Management**

* **3.10 .2.1 Single Session Enforcement:**  
  The system shall prevent simultaneous sessions for the same user/admin account.  
  Logging in from a new device/browser shall invalidate existing sessions.
* **3.10 .2.2 Session Timeout:**  
  Sessions shall expire after 15 minutes of inactivity.  
  Users shall receive a warning 2 minutes before expiration with option to extend.

**3.10 .3 Logout Confirmation**

* **3.10 .3.1 Feedback:**  
  Upon logout, a confirmation message shall appear:  
  “You have successfully logged out.”
* **3.10 .3.2 Redirect:**  
  The system shall redirect users to the login page automatically after logout.

**Security & Compliance Notes:**

* All logout and session management operations shall be protected against CSRF and session fixation attacks.
* HTTPS/TLS 1.3 mandatory on all logout requests.
* User privacy shall be maintained throughout logout and session expiration flows.

**Section 4: API Specifications**

**4.1 Architectural Style**

* The BDMS backend shall adopt a RESTful API design to support microservices architecture and enable scalability.
* APIs shall communicate over HTTPS using TLS 1.3 to ensure secure data transfer.

**4.2 Authentication and Authorization**

* All API endpoints shall require authentication using OAuth 2.0 or JWT tokens.
* Role-Based Access Control (RBAC) shall restrict access to endpoints based on user roles (Donor, Admin, Super Admin).
* Two-factor authentication (2FA) tokens shall be required for admin-level endpoints.

**4.3 Common API Endpoints**

**4.3.1 User Registration & Profile Management**

* POST /api/users/register — Creates a new donor user with multi-step form data.
* GET /api/users/{userId} — Fetch user profile details.
* PUT /api/users/{userId} — Update profile data with validation.
* Password and 2FA management APIs under /api/auth.

**4.3.2 Login & Verification**

* POST /api/auth/login — Login via username/email/phone + password.
* POST /api/auth/verify-otp — Verify OTP sent via SMS/email.
* Session token management with refresh token support.

**4.3.3 Donor Management (Admin)**

* GET /api/admin/donors — Paginated list with filters (blood group, status).
* POST /api/admin/donors — Create new donor manually or via bulk upload.
* PUT /api/admin/donors/{donorId} — Edit donor details.
* POST /api/admin/donors/export — Export donor data in CSV/Excel/PDF.

**4.3.4 Blood Inventory & Donations**

* POST /api/inventory/donations — Record new blood donation.
* GET /api/inventory/stock — Current blood stock per blood group.
* POST /api/inventory/issuance — Record blood issuance to patients.

**4.3.5 Blood Drives & Emergency Requests**

* POST /api/blood-drives — Schedule new blood drives.
* GET /api/blood-drives — List scheduled drives.
* POST /api/emergency-requests — Submit emergency blood request.
* GET /api/emergency-requests — Admin access for managing requests.

**4.4 API Security & Rate Limiting**

* APIs shall implement rate limiting to prevent abuse (e.g., 100 requests/min/user).
* All sensitive data fields (PII, health info) must be encrypted or masked.
* Audit logging for all critical API calls.

**Section 5: Integration Notes**

**5.1 OTP Verification**

* Integration with third-party SMS gateway (e.g., Twilio, Nexmo) for OTP delivery.
* Email OTP fallback via services like SendGrid or Mailgun.
* OTPs are 6-digit, valid for 15 minutes, with resend limits to prevent abuse.

**5.2 Email Services**

* Bulk and transactional emails shall be managed via SMTP or APIs from SendGrid/Mailgun.
* Support for email verification, password resets, notifications, and marketing.

**5.3 SMS Messaging**

* Predefined message templates with dynamic placeholders (blood group, date, location).
* SMS sending via gateway with delivery status callbacks.

**5.4 Map and Location Services**

* Use Google Maps API or OpenStreetMap for location lookup and geocoding (hospital, donor addresses).
* Autocomplete address fields with location validation.

**5.5 Payment Gateway (Optional)**

* Integration option for donations or payment processing (if required in future).

**Section 6: Deployment Guidelines**

**6.1 System Architecture**

* Backend deployed as containerized microservices (Docker/Kubernetes).
* Services include User Management, Donor Management, Inventory, Notification, Analytics.
* Database: Use PostgreSQL or MySQL with encrypted storage for PII.
* Caching with Redis or Memcached for fast retrieval of frequent queries.

**6.2 Security Considerations**

* TLS 1.3 mandatory for all client-server and inter-service communication.
* AES-256 encryption for PII in databases and backups.
* WAF and IDS/IPS for perimeter security.
* Regular vulnerability scanning and patching.

**6.3 Scalability and Performance**

* Autoscaling based on CPU/memory metrics and request rate.
* Load balancing with HAProxy or cloud provider native load balancer.
* Use CDN for static content and media (profile pictures, cover photos).

**6.4 Failover and Backup**

* Daily full database backups with incremental backups every hour.
* Geo-redundant storage to protect against data center failures.
* Health checks and automated failover configured for critical services.

**6.5 Monitoring and Logging**

* Centralized logging with ELK stack (Elasticsearch, Logstash, Kibana).
* Monitoring via Prometheus and Grafana dashboards.
* Alerting for service failures, low inventory, or suspicious activity.