

**<<BLOOD DONATION SUPPORT SYSTEM>>**

**Software Requirement Specification**

– Ho Chi Minh City, June 2025 –

# Table of Contents

## [I. Overview3](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.1ftpm9by3528)

[1. Introduction](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.tzys91xrcbc9)

[2. Business Main Flows](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.57s16wj63zy2)

[3. Business Rules](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=)

[4. System Functions](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.qn7075tm7c8)

[5. Logical Entity Relationship Diagram](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.7u3mu08fvf5x)

## [II. Functional Requirements](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.bmep67o44sd9)

## [1.Blood Donation](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.cojy1s54z2u7)

1. [Submit donation request](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.asjbp2s823ys)
2. Approve/reject request
3. Perform donation process
4. Generate donation history

## [2. Blood Transfusion](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.gwu2c6t7y0o6)

1. [Submit transfusion request](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.asjbp2s823ys)
2. Approve/reject request
3. Perform transfusion process
4. Generate transfusion history

## [3. Blood](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.gwu2c6t7y0o6) Transfusion Emergency

1. [Submit transfusion emergency request](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.asjbp2s823ys)
2. Approve/reject emergency request
3. Perform transfusion emergency process
4. Generate transfusion emergency history

## [4. Authentication](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.gwu2c6t7y0o6)

1. Login
2. Register
3. Forgotten Password

**I. Overview**

**1. Introduction**

The Blood Donation Support System is designed to digitize and optimize the management, organization, and coordination of blood donation and transfusion activities at medical facilities. The primary goal of the system is to provide an online platform that allows individuals to register as blood donors, submit emergency blood requests, and easily track their donation history. At the same time, the system assists medical staff in approving requests, managing blood supply logistics, and updating blood inventory accurately and efficiently.

The application will be deployed in a web environment accessible via modern web browsers on desktop computers. The system supports multiple types of users, including:

* **Guest:** Can submit emergency blood requests without registration.
* **Member:** Can register for blood donations, manage personal information, and track donation history.
* **Staff:** Responsible for approving requests, scheduling donations, and updating donation processes.
* **Admin:** Manages user accounts, system configurations, and oversees overall operations.

**Key assumptions and constraints include:**

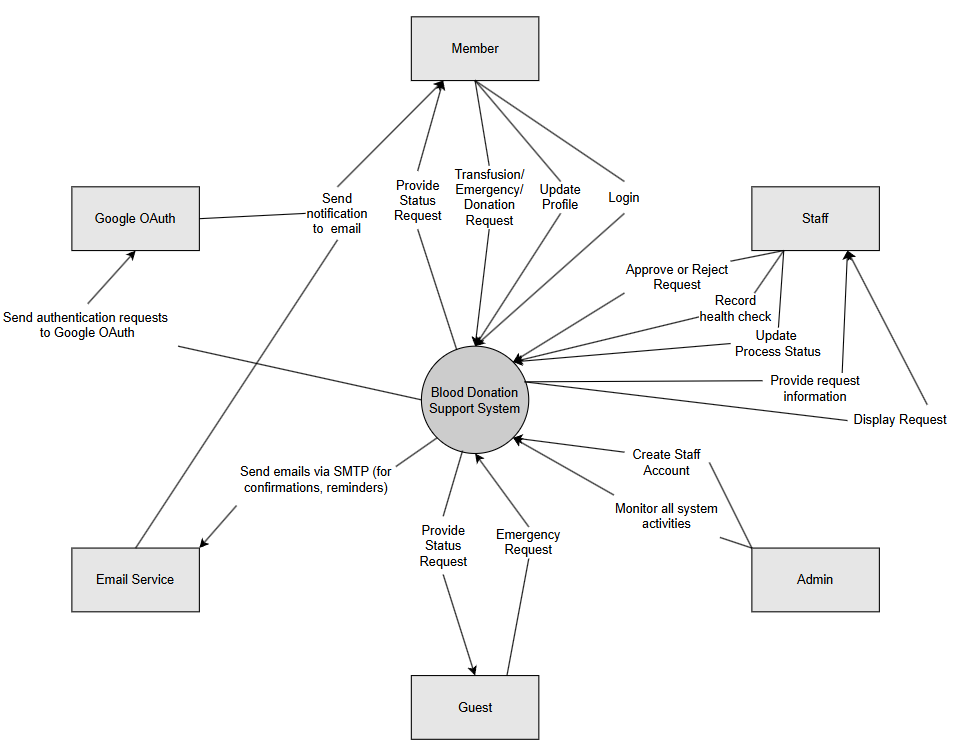
* The backend is implemented using Java with the Spring Boot framework, and the frontend is developed using React.
* Data is stored using SQL Server as the database management system.
* User management features rely on authentication via email and password.
* Data security and integrity are critical requirements for all system functionalities.
* Business processes such as request approvals and result updates are expected to be handled in real-time or near real-time.
* The system will initially be deployed in selected internal medical facilities before being scaled for broader use.

This document provides a detailed specification of the functional and non-functional requirements of the software, serving as the foundation for system design, development, and testing.

The Blood Donation Support System is a newly developed software designed to replace current manual or fragmented processes for blood donation registration, emergency transfusion request handling, and internal blood inventory management at medical facilities. The system is intended for local deployment within individual healthcare units, helping them manage blood-related processes and data more efficiently. In the future, the system may be extended to integrate with external services such as user authentication via Google.

**Context Diagram (textual description):**

* **Guest User:** Submits emergency blood transfusion requests.
* **Member User:** Submits blood donation registration forms, updates personal information, and tracks donation history.
* **Medical Staff:** Approves donation and transfusion requests, records health check results, and updates donation processes.
* **Administrator:** Manages user accounts, roles, and system configurations.
* **Email Service:** Sends confirmation messages, reminders, and system notifications to users.
* **Google OAuth:** Handles user login via Google accounts (planned for future integration).

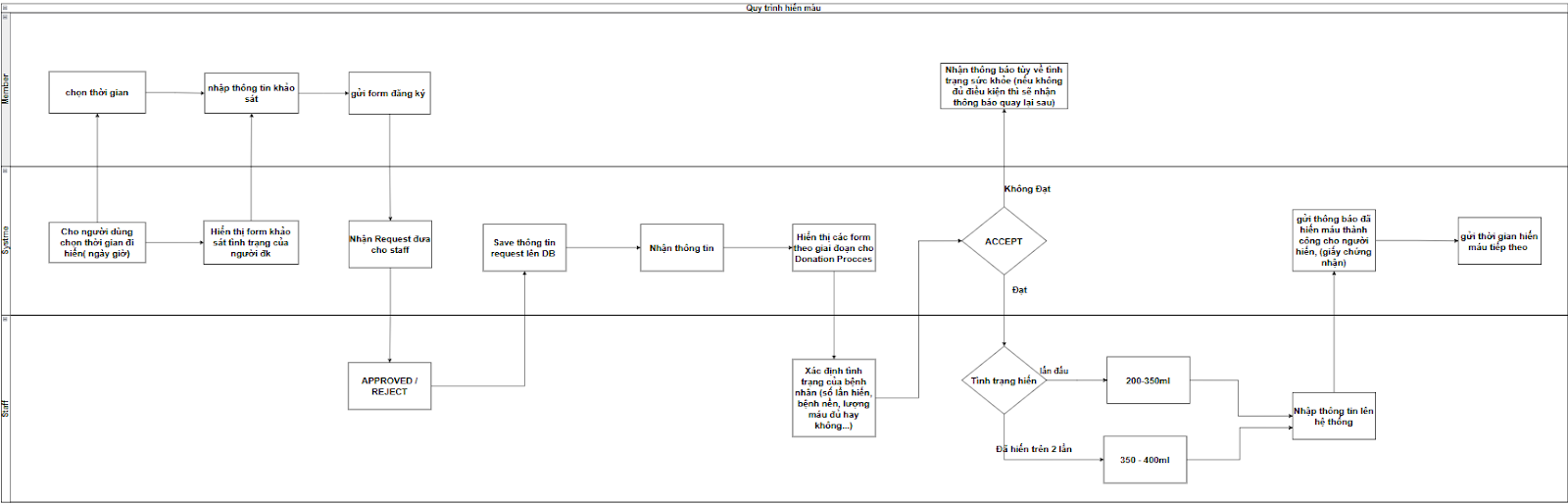


**2. Business Main Flows**

*This section outlines the key business flows of the system, with swimlane diagrams showing how different actors interact to complete each process.*

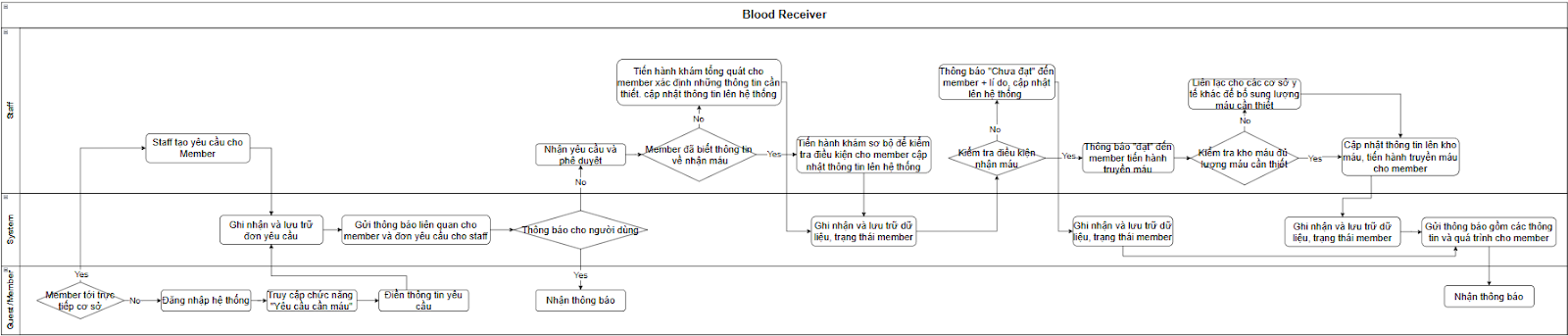
***2.1. Main-flow 01***

*This main flow describes the complete blood donation process from submitting a request to successfully recording the donation history.*

****

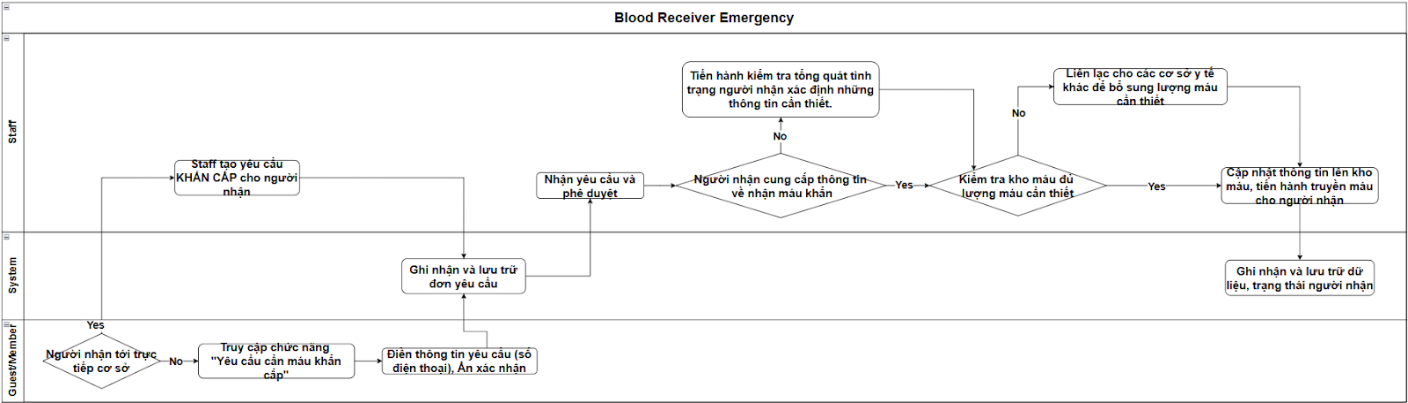
***2.2. Main-flow 02***

*This main flow covers the steps from creating a blood request to receiving and recording the transfusion result.*

******

***2.3. Main-flow 03***

*This main flow handles emergency blood request creation, prioritization, and fulfillment to ensure urgent cases are addressed quickly.*

******

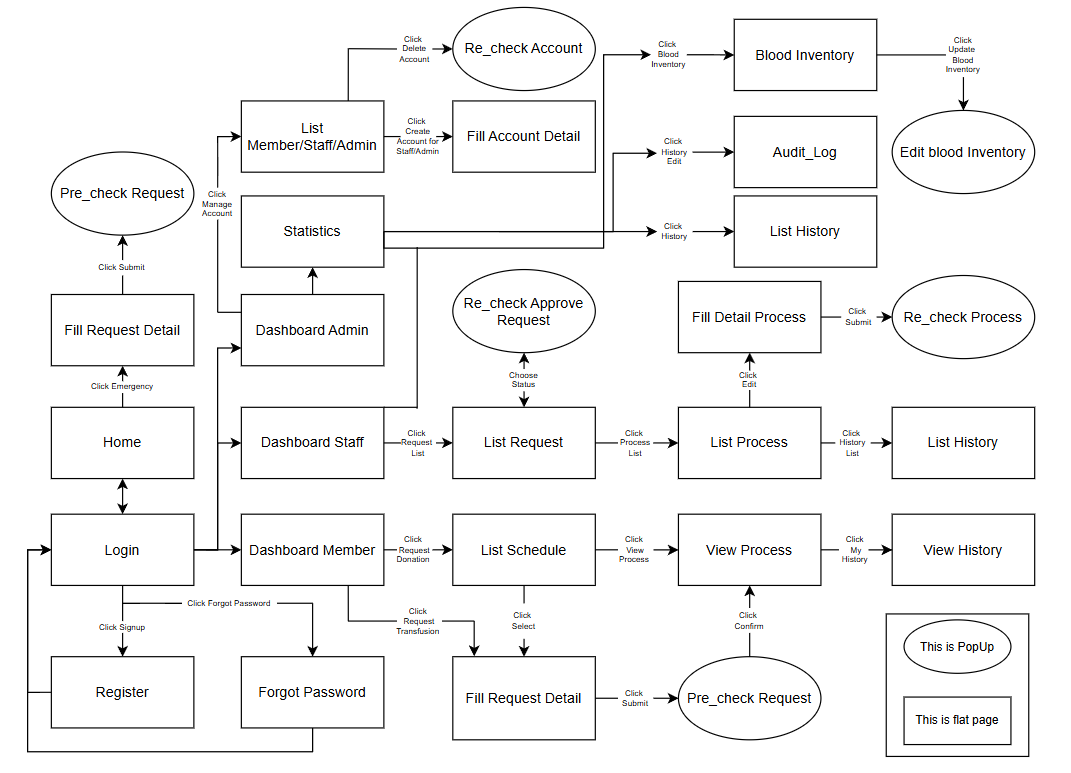
**3. Business Rules**

*This section lists all business rules applied to ensure safety, correctness, and policy compliance across system features.*

|  |  |
| --- | --- |
| **Business Rule ID** | **Business Rule Describe** |
| BR-0 | The system only accepts blood donation requests from individuals aged 18 or older (based on date of birth or ID/passport). If under 18, the request is rejected with the message: "Not old enough to donate blood." |
| BR-02 | The system only accepts blood donation requests from females weighing at least 45kg and males at least 50kg. If below the required weight, the request is rejected with the message: "Not enough weight to donate blood." |
| BR-03 | The system only accepts whole blood donation requests if at least 12 weeks (84 days) have passed since the donor's last donation. Otherwise, the request is rejected with the message: "Not enough recovery time between donations." |
| BR-04 | Each donor can be registered with only one blood type. Once confirmed and saved, the blood type cannot be changed unless verified by a medical facility. |
| BR-05 | The system only allows blood distribution between medically compatible blood types. Incompatible matches are blocked to ensure transfusion safety. |
| BR-06 | Users can only search for blood donation events within the same city. Results outside the allowed range are excluded. |
| BR-07 | Emergency blood requests must be displayed at the top of the blood request list to prioritize urgent medical needs. |
| BR-08 | The system automatically sends a recovery reminder via email and dashboard notification 12 weeks after a member's last whole blood donation. |
| BR-09 | A member is considered a potential donor only if their profile status is set to "Ready to Donate." Only such members are included in donor search and matching results. |
| BR-10 | Each patient can have only one active emergency blood request at a time. New requests are blocked if an active one already exists, unless resolved or canceled. |
| BR-11 | All blood inventory updates created or edited by staff must be approved by an Admin before becoming official. Changes are stored as “pending” until approved. |

**4. System Functions**

***4.1. Screen Flow***

*The diagram shows the navigation flow between main screens for each user role in the system.*

***4.2. Screen Details***

*This section describes the purpose and function of each screen shown in the system's screen flow.*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Authentication | Login | Login screen for all users (Member, Staff, Admin) to access the system. |
| 2 | Authentication | Register | Registration form for creating a new user account (typically for Members). |
| 3 | Authentication | Forgot Password | Allows users to request password reset via email. |
| 4 | Home & Navigation | Home | Main landing page after login (for Members), showing blood request options. |
| 5 | Blood Donation/transfusion Request | Fill Request Detail | Form for submitting blood donation or transfusion requests (for Members or Guests). |
| 6 | Blood Donation/transfusion Request | Pre Check Request | Popup for validating eligibility (age, weight, interval, etc.) before submission. |
| 7 | Blood Donation/transfusion Request | List Schedule | Displays available donation schedules for Members to register. |
| 8 | Blood Donation/transfusion Request | View Process | Shows progress of a submitted donation request. |
| 9 | Emergency Request | Fill Request Detail | Form for submitting blood donation or transfusion requests (for Members or Guests). |
| 10 | Emergency Request | Pre Check Request | Popup for validating eligibility (age, weight, interval, etc.) before submission. |
| 11 | Donation/ Transfusion Management (Staff) | Dashboard Staff | Main dashboard for staff users to manage requests and processes. |
| 12 | Donation/ Transfusion Management (Staff) | List Request | Lists pending blood requests awaiting staff approval. |
| 13 | Donation/ Transfusion Management (Staff) | Re\_check Approve Request | Popup to confirm approval decision before processing a request. |
| 14 | Donation/ Transfusion Management (Staff) | List Process | Lists approved donation requests waiting for process recording. |
| 15 | Donation/Transfusion Process Recording (Staff) | Fill Detail Process | Form for staff to enter donation process details (e.g., blood pressure, quantity). |
| 16 | Donation/Transfusion Process Recording (Staff) | Recheck Process | Popup to confirm donation process data before final submission. |
| 17 | Admin Management | Dashboard Admin | Admin dashboard to manage users, view reports, and oversee system status. |
| 18 | Admin Management | List Member/Staff/Admin | Interface to manage and view all users by their roles. |
| 19 | Admin Management | Fill Account Detail | Form for creating new Staff/Admin accounts. |
| 20 | Admin Management | Recheck Account | Popup to confirm user creation or deletion actions. |
| 21 | Blood Inventory | Blood Inventory | Displays current blood stock information. |
| 22 | Blood Inventory | Edit Blood Inventory | Form to update blood inventory levels (requires admin approval). |
| 23 | History & Logs | List History | Shows donation and transfusion history records. |
| 24 | History & Logs | Audit\_Log | Displays system activity logs for tracking user actions. |
| 25 | Statistics | Statistics | Provides statistical summaries of donation requests, blood types, and other data. |
| 26 | Blood Usage Tracking | Blood Usage History | Displays the usage status of donated blood by the user. Users can view where, when, and how their blood was used (if available). |

***4.3. User Authorization***

*This section outlines which user roles are authorized to access and perform actions on each system feature or screen.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Role1** | **Role2** | **Role3** | **Role4** |
| **Home** | X | X | X | X |
| **Login** | X | X | X | X |
| **Register** | X | X |  |  |
| **Forgot Password** | X | X | X | X |
| **Fill Request Detail** | X | X |  |  |
| Submit Request | X | X |  |  |
| **List Schedule** |  | X |  |  |
| View Schedule |  | X |  |  |
| **View Process** |  | X |  |  |
| **Dashboard Member** |  | X |  |  |
| **Dashboard Staff** |  |  | X |  |
| **Dashboard Admin** |  |  |  | X |
| **List Request** |  |  | X |  |
| Approval Request |  |  | X |  |
| **List Process** |  |  | X |  |
| Add Process Detail |  |  | X |  |
| Submit Process |  |  | X |  |
| **Blood Inventory** |  |  | X | X |
| Update Inventory |  |  | X | X |
| **List History** |  | X | X | X |
| View Own History |  | X |  |  |
| View All History |  |  | X | X |
| **List Member/Staff/Admin** |  |  |  | X |
| Add New Account |  |  |  | X |
| Delete Account |  |  |  | X |
| **Audit Log** |  |  |  | X |
| **Statistics** |  |  |  | X |

In which:

* Role1: Guest
* Role2: Member
* Role3: Staff
* Role4: Admin

***4.4. Non-Screen Functions***

*This section covers automated and backend system functions such as services, scheduled jobs, and internal logic that support system features.*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | Blood Donation Validation | validateAgeForDonation() | Automatically checks if donor is at least 18 years old before accepting request. |
| 2 | Blood Donation Validation | validateWeightByGender() | Validates that female donors weigh ≥45kg and males ≥50kg before submission. |
| 3 | Donation Interval Enforcement | checkDonationInterval() | Checks if at least 12 weeks have passed since the donor’s last whole blood donation. |
| 4 | User Profile Management | lockConfirmedBloodType() | Prevents users from changing confirmed blood type unless verified by a medical source. |
| 5 | Blood Compatibility Matching | verifyBloodTypeCompatibility() | Ensures that blood distribution occurs only between compatible blood types. |
| 6 | Event Search | filterEventsByCity() | Limits search results to blood donation events within the user’s city only. |
| 7 | Emergency Prioritization | sortEmergencyRequestsFirst() | Automatically ranks emergency requests to appear at the top of blood request lists. |
| 8 | Donation Reminder | sendRecoveryReminderJob() | Sends email and dashboard notification 12 weeks after member's last donation. |
| 9 | Donor Availability | filterDonorsByReadyStatus() | Includes only members with "Ready to Donate" status in donor matching results. |
| 10 | Emergency Request Restriction | validateSingleActiveEmergency() | Blocks new emergency requests for a patient if an active one already exists. |
| 11 | Inventory Approval Workflow | markInventoryChangePending() | Flags updates to blood inventory as “pending” until approved by Admin. |
| 12 | Blood Usage Tracking | trackAndNotifyBloodUsage() | When a donated blood unit is used, the system links it to the donor and optionally sends a notification or updates the donor’s usage history screen. |

**5. Logical Entity Relationship Diagram**

*This section provides a logical view of system entities and their relationships to support database design and data integrity.*

**

**II. Functional Requirements**

**1. Blood Donation**

**a. Submit Donation Request**

* **Function Trigger:**
  + This function is triggered when a Member logs into the system and navigates to:
  + Dashboard Member → Blood Donation Service → List Schedule → Submit Donation Request
* **Function Description:**
  + Actor: Member
  + Purpose: To allow a member to submit a new blood donation request for an available donation schedule.
  + Interface: Web form screen, containing personal details (pre-filled), blood type (read-only), schedule selection, and health info.
  + Data Processing:
    - Save request to donation\_request table with status = PENDING
    - Notify responsible staff for approval.
* **Screen Layout:**
* **Function Details:**
  + **Data Validation**:
    - The schedule must be selected.
    - Members must be ≥ 18 years old (check from date of birth).
    - Can not submit if there is a **pending** or **approved** donation request not yet completed.
  + **Business Logic**:
    - One request per schedule per member.
    - Auto-calculate eligibility based on last donation date (must be ≥ 12 weeks ago).
  + **Normal Case**:
    - Member submits request → system saves to DB → system displays confirmation message.
  + **Abnormal Case**:
    - If a member is under 18 → show error: "You are not eligible to donate blood."
    - If it already has an active request → show error: "You already have a pending donation request."

**b. Approve/reject request**

* **Function Trigger:**
  + This function is triggered when a **Staff** logs into the system and navigates to:
  + Dashboard Staff → Donation Request Management → List Request →  Pending Request
  + From there, the staff selects a pending request and chooses to either **approve** or **reject** it.
* **Function Description:**
  + Actor: Staff
  + Purpose: Review and approve or reject donation requests submitted by members.
  + Interface: List of pending donation requests with filter/search options. Clicking on a request opens a detailed view with member information, selected schedule, and health info.
  + Data Processing:
    - If approved:
      * Set status = APPROVED
      * Set approver\_id = [Staff ID], approved\_time = [current timestamp]
    - If rejected:
      * Set status = REJECTED
      * Include rejection note (required)
      * Notify member via email/dashboard.
* **Screen Layout:**
* **Function Details:**
  + **Data Validation**:
    - A request must be in PENDING status to be approved or rejected.
    - Rejection must include a reason (non-empty note field).
  + **Business Logic**:
    - A member can only have one **approved** request per active schedule.
    - Cannot approve a request if the selected schedule is already full.
    - Upon approval or rejection, a notification is sent to the member.
  + **Normal Case**:
    - Staff approve a valid request → system updates status, logs staff info.
    - Staff rejects with reason → system updates status, sends rejection note.
  + **Abnormal Case**:
    - Attempt to approve already approved request → show error: *"This request has already been processed."*
    - Missing rejection reason → prevent submission with message: *"Rejection reason is required."*

**c. Perform donation process**

* **Function Trigger:**
  + This function is triggered when a **Staff** accesses:
  + Dashboard → Donation Management → Approved Requests,
  + then selects a request and clicks **"Start Donation Process"** to record the actual donation.
* **Function Description:**
  + **Actor(s)**: Staff (typically medical personnel)
  + **Purpose**: To document the actual blood donation session for an approved request, including medical check results, donation data, and blood unit creation.
  + **Interface**: A structured form capturing donation date, vital signs, blood quantity, type of donation, and optional notes.
  + **Data Processing**:
    - Insert data into donation\_process table.
    - Validate medical criteria before allowing completion.
    - On success, create a new record in blood\_unit and optionally update blood\_inventory.
* **Screen Layout:**
* **Function Details:**
  + **Data Validation**:
    - startTime < endTime, both required.
    - Blood pressure must match format NNN/NN (e.g., 120/80).
    - Quantity between 100–500 ml.
    - Hemoglobin > minimum safe threshold (e.g., 12.5 g/dL for women, 13.0 for men).
    - Donation Type must be selected.
    - If marked as “Completed”, health check must pass.
  + **Business Logic**:
    - Only staff can perform this function.
    - One donation process per approved request.
    - On completion:
      * Update request status = COMPLETED
      * Insert a blood unit record (linked to donation\_history)
      * Inventory may be updated automatically or pending approval depending on policy.
    - On failure:
      * Update process as FAILED
      * Add note with reason (e.g., failed health check)
  + **Normal Case**:
    - Staff enters complete valid data → donation is recorded → confirmation shown → data saved.
  + **Abnormal Case**:
    - Missing required fields → validation errors.
    - Blood pressure in wrong format → error message.
    - Trying to complete an already completed/failed process → blocked with message.

**d. Generate donation history**

* **Function Trigger:**
  + This function is automatically triggered **after a donation process is marked as COMPLETED** by staff.
* **Function Description:**
  + **Actor(s)**: System (auto-triggered), Staff (view only).
  + **Purpose**: To log the completed donation session into the donor’s history for future reference, reporting, and eligibility tracking.
  + **Interface**: No manual screen — history is recorded in the backend and viewable in the "Donation History" screen for both **Member** and **Staff/Admin**.
  + **Data Processing**:
    - Extract relevant data from:
      * donation\_process (staff, date, health info)
      * donation\_request (schedule, donor)
      * donation\_schedule (facility info)
    - Save data into donation\_history table.
    - Associate with new blood unit ID (if created).
    - Update donor’s last donation timestamp.
* **Screen Layout:**
* **Function Details:**
  + **Data Validation**:
    - Only COMPLETED donation processes are recorded.
    - Ensure mapping to a valid blood unit and donor.
  + **Business Logic**:
    - Automatically triggered without staff action.
    - Donor’s profile is updated with last\_donation\_date = current donation date.
    - May be used to calculate eligibility for next donation (≥ 12 weeks later).
    - Links to reporting modules and blood inventory tracking.
  + **Normal Case**:
    - Once donation is marked complete → history is created → visible to donor.
  + **Abnormal Case**:
    - If related data (e.g., donation\_request or schedule) is missing → system logs error for admin review (but does not block user flow).
    - If history already exists for the same request → skip duplicate creation.

**2. Blood Transfusion**

**a. Submit Transfusion Request**

* **Function Trigger:** This function is triggered when a user (Member or Guest) navigates to:
  + Dashboard → Emergency Blood Request → Submit Request
* **Function Description:**
  + **· Actor:** Member or Guest
  + **· Purpose:** To allow a user to request blood transfusion in emergency situations.
  + **· Interface:** Web form screen, including fields: full name, contact info, blood type (select), quantity needed, reason, and optional notes.
* **Data Processing:**
  + · Save request to emergency\_request table with status = PENDING
  + · Notify assigned staff for review and action
* **Screen Layout:** A simple web form with fields for personal and medical info, and submit button
* **Function Details:**
  + **· Data Validation:**
    - o Required fields: full name, phone, blood type, quantity
    - o Quantity must be in range 1–500 ml
  + **· Business Logic:**
    - o Guests can submit without login
    - o Requests from the same user must be completed before creating new ones
* **Normal Case:**User submits request → system saves to DB → confirmation message displayed
* **Abnormal Case:**
  + If missing required fields → show error: “Please fill in all required information.”
  + If quantity invalid → show error: “Quantity must be between 1 and 500 ml.”

**b. Approve/Reject Request**

* **Function Trigger:** Triggered when a Staff logs in and navigates to:
  + Dashboard Staff → Emergency Requests → Pending List → View Details → Approve / Reject
* **Function Description:**
  + **· Actor:** Staff
  + **· Purpose:** To allow staff to evaluate and respond to emergency blood requests
  + **· Interface:** Detail view screen with all request information, input for note, and action buttons
* **Data Processing:**
  + · Update emergency\_request status to APPROVED or REJECTED
  + · Save approver ID and decision time
  + · Optionally, notify the requester
* **Screen Layout:** Detail card with request info, note field, approve/reject buttons
* **Function Details:**
  + **· Data Validation:**
    - o Staff must be logged in
    - o Note field required when rejecting
  + **· Business Logic:**
    - o Cannot approve if insufficient blood in stock
    - o Once approved, blood allocation process begins
* **Normal Case:** Staff approves → system updates record → confirmation shown
* **Abnormal Case:** If no blood available → show error: “Not enough stock to approve this request.”

**c. Perform Transfusion Process**

* **Function Trigger:**
  + Triggered when staff accesses:
  + Dashboard Staff → Approved Requests → Start Transfusion Process
* **Function Description:**
  + **· Actor:** Staff
  + **· Purpose:** To track the actual transfusion activity in the system
  + **· Interface:** Input screen with actual transfusion date, time, blood component used, quantity, and transfusion notes
* **Data Processing:**
  + · Save transfusion info to transfusion\_process table
  + · Deduct used blood from inventory
  + · Link process with related emergency request
* **Screen Layout:** Form with date/time pickers, dropdown for blood component, quantity input, and text area for notes
* **Function Details:**
  + **· Data Validation:**
    - All fields are mandatory
    - Quantity must match approved amount
  + **· Business Logic:**
    - One transfusion per approved request
    - Only staff can perform this function
* **Normal Case:** Staff submits process → system saves to DB and updates stock
* **Abnormal Case:** If quantity mismatch → show error: “Used quantity must match approved request.”

**d. Generate Transfusion History**

* **Function Trigger:** Triggered automatically once transfusion is marked completed
* **Function Description:**
  + **· Actor:** System (auto-generated)
  + **· Purpose:** To store a permanent record of completed transfusion activities
  + **· Interface:** No user input; viewed in:  
    Dashboard Admin → Transfusion History
* **Data Processing:**
  + · Insert new record into transfusion\_history table
  + · Data includes: request info, transfusion data, staff info, and timestamp
* **Screen Layout:**List view with filters by date, blood type, patient, staff, etc.
* **Function Details:**
  + **· Business Logic:**
    - History record created only if transfusion is successful
    - Immutable record — no edits allowed after creation
* **Normal Case:**Transfusion complete → history record saved → visible in admin dashboard
* **Abnormal Case:**If data incomplete → system skips history creation and logs error internally

[**3. Blood**](https://docs.google.com/document/d/1qXpkUvF9x37Xjz-HwgoKbiDzwAQ-GGyPuK_2KgZPGuM/edit?tab=t.0#heading=h.gwu2c6t7y0o6) **Transfusion Emergency**

**a. Submit Transfusion Emergency Request**

* **Function Trigger:** This function is triggered when a user (Member or Guest) navigates to:
  + Dashboard → Emergency Transfusion → Submit Emergency Request
* **Function Description:**
  + **· Actor:** Member or Guest
  + **· Purpose:** To allow a user to submit an urgent blood transfusion request without scheduling
  + **· Interface:** Emergency request form including: full name, phone number, blood type (select), blood component (select), quantity (ml), emergency reason, and optional notes
* **Data Processing:**
  + · Save to emergency\_request table with status = PENDING
  + · Notify staff for immediate review
* **Screen Layout:**Simple emergency form with all required fields and Submit button
* **Function Details:**
  + **· Data Validation:**
    - o Required: full name, phone, blood type, component, quantity
    - o Quantity must be between 1 and 500 ml
  + **· Business Logic:**
    - o No login required
    - o Multiple emergency requests allowed, but staff must approve manually
    - o Guest accounts do not save history unless email provided
* **Normal Case:** User submits → system saves to DB → confirmation message displayed
* **Abnormal Case:**
  + If missing fields → error: “Please complete all required information.”
  + If invalid quantity → error: “Quantity must be between 1 and 500 ml.”

**b. Approve/Reject Emergency Request**

* **Function Trigger:** Triggered when a Staff navigates to:Dashboard Staff → Emergency Transfusion Requests → Pending → View Details → Approve / Reject
* **Function Description:**
  + **· Actor:** Staff
  + **· Purpose:** To evaluate emergency requests and determine feasibility
  + **· Interface:** Detail view of request with option to add note, and action buttons
* **Data Processing:**
  + · Update emergency\_request status to APPROVED or REJECTED
  + · Save approver ID, timestamp, and note
  + · Notify requester (if contact provided)
* **Screen Layout:**
* **Function Details:**
  + **· Data Validation:**
    - o Reject requires note
  + **· Business Logic:**
    - o Only approve if matching blood is available in stock
    - o Request marked REJECTED if no match found
* **Normal Case:** Staff approves → system updates status → inventory is checked
* **Abnormal Case:** If no matching blood available → error: “No compatible blood component available for this request.”

**c. Perform Transfusion Emergency Process**

* **Function Trigger:** Triggered by staff from: Dashboard Staff → Emergency Transfusion → Approved List → Perform Process
* **Function Description:**
  + **· Actor:** Staff
  + **· Purpose:** To log the actual emergency transfusion that was performed
  + **· Interface:** Input screen for transfusion data: date, start/end time, component used, quantity, and medical notes
* **Data Processing:**
  + · Save to emergency\_process table
  + · Deduct blood from inventory
  + · Link to emergency\_request for history tracking
* **Screen Layout:** Form with date/time pickers, dropdowns for component, numeric field for quantity, and text area
* **Function Details:**
  + **· Data Validation:**
    - o Quantity must not exceed approved amount
    - o Time must be valid (end > start)
  + **· Business Logic:**
    - o One emergency process per approved request
    - o Only staff can execute process
* **Normal Case:**
* Staff records process → system updates inventory → marks process as COMPLETED
* **Abnormal Case:**
* l If quantity > approved → error: “Used quantity exceeds approved limit.”
* l If invalid times → error: “End time must be after start time.”

**d. Generate Transfusion Emergency History**

* **Function Trigger:** Automatically triggered after the emergency transfusion process is marked COMPLETED
* **Function Description:**
  + **· Actor:** System
  + **· Purpose:** To create an immutable log of the emergency transfusion performed
  + **· Interface:** History view in:  
    Dashboard Admin → Emergency Transfusion History
* **Data Processing:**
  + · Insert new record into emergency\_history table
  + · Include details: patient info, blood type, component, quantity, transfusion time, staff info
* **Screen Layout:** List of past emergency transfusions with filters (by blood type, date, staff, etc.)
* **Function Details:**
  + **· Business Logic:**
    - o Record created only if emergency process is marked COMPLETED
    - o History is read-only
* **Normal Case:** System creates history record → available in admin dashboard
* **Abnormal Case:** If data is incomplete or missing → system logs internal error, skips history creation

**4. Authentication**

**a. Login**

* **Function Trigger**:  
   Triggered when a user navigates to the Login screen from the homepage or system header.
* **Function Description**:
  + **Actors**: Member, Staff, Admin
  + **Purpose**: Allow users to access the system using email and password.
  + **Validation**:
    - email: required, must not be blank
    - password: required, must not be blank
  + **Business Logic**:
    - Authenticate credentials using authenticationManager.
    - If incorrect: return error "Invalid username or password".
    - On success: generate and return authentication token.
* **Error Handling**:
  + Empty fields → display specific validation messages.
  + Wrong credentials → generic error message to avoid security leaks.

**b. Register**

* **Function Trigger**:  
   Triggered when a guest user clicks the “Register” button from the login page or homepage.
* **Function Description**:
  + **Actors**: Guest
  + **Purpose**: Allow guest users to create a Member account.
  + **Validation**:
    - email: required, valid format, must be unique
    - password:
      * Requirement:
        + Required
        + min 6 characters
        + Must contain at least:

One uppercase letter (A–Z)

One lowercase letter (a–z)

One digit (0–9)

One special character (e.g., @,$,!,%,\*,?,&)

* + - * Error Messages (if invalid):
        + "Password is required."
        + "Password must be at least 6 characters."
        + "Password must contain at least one uppercase letter, one lowercase letter, one number, and one special character."
    - fullName: required
    - cccd: required, exactly 12 digits, must be unique
    - gender: required (enum)
    - bloodTypeId: required
    - dateOfBirth: required, must be in the past
    - phone: required, must start with 0, exactly 10 digits, must be unique
    - address: required
  + **Business Logic**:
    - Check for uniqueness: email, phone, CCCD
    - On conflict: throw descriptive IllegalArgumentException
    - Create account with default role: MEMBER
    - Save to database and return confirmation.
* **Error Handling**:
  + Duplicate email/phone/cccd → show specific error:
    - "Email already in use" / "Phone number already in use" / "CCCD already in use"
* Invalid field formats → show validation messages.

**c. Forgotten Password**

* **Function Trigger**:  
   Triggered when a user clicks “Forgot Password” on the login screen.
* **Function Description**:
  + **Actors**: All roles
  + **Purpose**: Allow users to recover access by receiving a password reset link.
  + **Process Overview**:
    - User inputs registered email
    - System verifies email exists
    - Send reset link via SMTP (or token) to email
    - User clicks link → redirected to reset form
    - Enter new password → update securely in DB
* **Validation**:
  + Email must exist in system
  + New password must meet security criteria (e.g., min 6 chars)
* **Error Handling**:
  + Email not found → error message: "Email not registered"
  + Expired or invalid token → "Reset link is invalid or expired"