

Darshan University

A Project Report on

**“Blood Bank Management”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – V

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Akshat Patel | Enrollment No.: 22010101472 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2101CS503)** for entitled **“Blood Bank Management System”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of **Rajkumar B. Gondaliya** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Akshat Patel

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Blood Bank Management System” has** been satisfactorily prepared by **Akshat Patel** (**22010101472**) under my guidance in the fulfillment of the course **Software Engineering (2101CS503)** work during the academic year 2023-2024.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof., R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. **R. B. Gondaliya** and all the faculty members for helping me through my project by giving me the necessary suggestions and advice along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Akshat Patel**

**ABSTRACT**

A Blood Bank Management System (BBMS) is an integral component of modern healthcare infrastructure, designed to streamline and enhance the collection, storage, and distribution of blood and its components. This system aims to ensure the availability and safety of blood for transfusions, address the challenges of blood shortages, and minimize wastage. The BBMS employs advanced database management technologies to maintain comprehensive records of donors, recipients, blood inventory, and transfusion history.

The core functionalities of a BBMS include donor registration, blood donation scheduling, blood type and component testing, inventory management, and recipient matching. Automated notifications and alerts are generated to manage inventory levels, notify donors about upcoming donation schedules, and track blood expiration dates. The system also integrates with laboratory equipment for real-time updates on blood test results, ensuring compliance with safety and quality standards.

Additionally, the BBMS provides an interface for healthcare professionals to access and request blood units, enhancing the coordination between blood banks and hospitals. This system also supports data analytics to identify trends in blood donation and usage, aiding in strategic planning and public health initiatives.

**Table of Contents**

[List of Figures I](#_Toc177642405)

[List of Tables II](#_Toc177642406)

[1 Introduction 1](#_Toc177642407)

[1.1 Product perspective 1](#_Toc177642408)

[1.2 Product features 1](#_Toc177642409)

[1.2.1 There are three different users who will be using this product: 1](#_Toc177642410)

[1.2.2 The features that are required for the Administrator are: 1](#_Toc177642411)

[1.2.3 The features that are required for the Donor are: 1](#_Toc177642412)

[1.2.4 The features that are required for the Recipient are: 1](#_Toc177642413)

[1.3 Functional Requirement 1](#_Toc177642414)

[1.3.1 Donor 1](#_Toc177642415)

[1.3.2 Receiver (Hospital/Clinic Staff) 1](#_Toc177642416)

[1.3.3 Blood Bank Staff 2](#_Toc177642417)

[1.3.4 Admin 2](#_Toc177642418)

[1.4 Non-Functional Requirement 2](#_Toc177642419)

[1.4.1 Usability 2](#_Toc177642420)

[1.4.2 Accuracy 2](#_Toc177642421)

[1.4.3 Availability 2](#_Toc177642422)

[1.4.4 Maintainability 2](#_Toc177642423)

[1.4.5 Security 3](#_Toc177642424)

[1.4.6 Scalability 3](#_Toc177642425)

[1.4.7 Performance 3](#_Toc177642426)

[2 Design and Implementation Constraints 4](#_Toc177642427)

[2.1 Use case diagram 4](#_Toc177642428)

[2.2 Activity diagram and Swimlane diagram 5](#_Toc177642429)

[2.3 Sequence diagram 9](#_Toc177642430)

[2.4 State diagram 10](#_Toc177642431)

[2.5 Class diagram 11](#_Toc177642432)

[2.6 Data flow diagram 12](#_Toc177642433)

[2.6.1 DFD level 0 12](#_Toc177642434)

[2.6.2 DFD Level 1 13](#_Toc177642435)

[3 External interface requirement (Screens) 14](#_Toc177642436)

[3.1 Screen-1: Blood Request Form 14](#_Toc177642437)

[3.2 Screen-2: Login Form 15](#_Toc177642438)

[3.3 Screen-3: Donor Registration 16](#_Toc177642439)

[3.4 Screen-4: Search Donor Availability 18](#_Toc177642440)

[3.5 Screen-5: Enquiry/Contact 19](#_Toc177642441)

[4 Database design 20](#_Toc177642442)

[4.1 List of Tables 20](#_Toc177642443)

[5 Stories and Scenario 21](#_Toc177642444)

[5.1 Story-1: Add New Blood Donor to the Blood Bank Database 21](#_Toc177642445)

[5.1.1 Scenario# S1.1 21](#_Toc177642446)

[5.1.2 Scenario# S1.2 21](#_Toc177642447)

[5.1.3 Scenario# S1.3 21](#_Toc177642448)

[5.2 Story-2: Update Blood Inventory Levels 21](#_Toc177642449)

[5.2.1 Scenario# S2.1 21](#_Toc177642450)

[5.2.2 Scenario# S2.2 22](#_Toc177642451)

[5.2.3 Scenario# S2.3 22](#_Toc177642452)

[6 Test cases 23](#_Toc177642453)

[7 References 27](#_Toc177642454)

# List of Figures

[Figure 2.1‑1 Use Case diagram for Blood Bank Management System 4](#_Toc177540507)

[Figure 2.2‑1 Activity diagram for Blood Allocation 5](#_Toc177540508)

[Figure 2.3‑1 Sequence diagram for Blood Allocation 5](#_Toc177540508)

[Figure 2.4‑1 State diagram of Blood 10](#_Toc177540509)

[Figure 2.4‑2 State diagram for Donor 10](#_Toc177540510)

[Figure 2.5‑1 Class diagram for Blood bank management system 11](#_Toc177540511)

[Figure 2.6‑1 DFD Level 0 diagram for Blood bank management system 12](#_Toc177540512)

[Figure 2.6‑2 DFD level-1 for Blood bank management system 13](#_Toc177540513)

[Figure 3.1‑1 Screen-1: Blood Request Form 14](#_Toc177540514)

[Figure 3.2‑1 Screen-2: Login Form 15](#_Toc177540515)

[Figure 3.3‑1 Screen-3: Donor Registration 16](#_Toc177540516)

Figure 3.4-1 Screen-4: Search Donor Availability………………………………………………………………………………………...17

[Figure 3.5-1 Screen-5: Enquiry/Contact 18](#_Toc177540517)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 14](#_Toc177632355)

[Table 3.2‑1 Screen element of Login form 15](#_Toc177632356)

[Table 3.3‑1 Screen element of Add borrower book 17](#_Toc177632357)

[Table 3.4‑1 Screen element of Add borrower book 17](#_Toc177632358)

[Table 3.5‑1 Screen Enquiry/Contact 18](#_Toc177632359)

Table 4.1-1 Table: Donor…………………………………………………………………………………………………………………………….19

[Table 4.1‑2 Table: Recipient 19](#_Toc177632360)

[Table 4.1‑3 Table: Blood Request 19](#_Toc177632361)

[Table 4.1‑4 Table: Blood 20](#_Toc177632362)

[Table 4.1‑5 Table: Blood](#_Toc177632362)  Bank Staff…………………………………………………………………………………………………………….21

# Introduction

## Product perspective

In the healthcare sector, the availability and proper management of blood and its components are critical for patient care, especially in emergency and surgical situations. The Blood Bank Management System (BBMS) is a specialized software solution designed to streamline the complex processes involved in blood collection, storage, and distribution. By integrating advanced database management and automation technologies, the BBMS ensures efficient tracking and management of blood donations, inventories, and transfusions. This system not only enhances the operational efficiency of blood banks but also improves the safety and reliability of blood supplies, ultimately playing a vital role in saving lives and supporting public health initiatives.

## Product features

### There are three different users who will be using this product:

* Blood bank manager who will be acting as the administrator**.**
* Donor individuals donating blood.
* Recipient/Requestor which are hospitals or individuals requesting blood.

### The features that are required for the Administrator are:

* Track blood donations, categorize by blood type, and monitor stock levels.
* Maintain records of donors, including medical history, donation frequency, and blood type.
* Approve or reject blood requests from hospitals or individuals based on availability.
* Quickly find donors or blood units based on blood type, donor ID, or expiration dates.
* Set and manage appointments for blood donation drives or individual donations.

### The features that are required for the Donor are:

* Check personal donation records, including dates and blood types donated.
* Book appointments for upcoming blood donation drives or personal donations.
* Get reminders for eligible donation dates, upcoming appointments, and special blood drive events.
* Edit personal details such as contact information and medical history.
* Request and download a certificate of donation for personal records.

### The features that are required for the Recipient are:

* View available blood units based on blood type and location.
* Submit requests for specific blood types and units based on medical needs.
* Monitor the status of blood requests (approved, pending, or rejected).
* Get updates on the status of submitted requests and availability of needed blood units.
* Manage and update personal or hospital account details.

## Functional Requirement

### Donor

* Registration: Complete and submit personal details and medical history.
* Appointment Management: Schedule, reschedule or cancel donation appointments.
* History: View donation history.
* Eligibility: Check eligibility status for the next donation.
* Incentives: Receive information about donor incentives (e.g., certificates, rewards).
* Feedback: Provide feedback on the donation process.

### Receiver (Hospital/Clinic Staff)

* Registration: Complete and submit personal details and medical history.
* Blood Request Management: Submit blood requests specifying blood type and quantity, track the status of blood requests (pending, approved, dispatched) ,update or cancel blood requests as needed.
* Blood Allocation: Allocate blood units for specific patients.
* Blood Reservation: Reserve blood units for scheduled surgeries or emergencies.
* Inventory: Track the inventory of blood units received.
* Reporting and Documentation: Generate reports on blood usage and request history also document patient transfusion details and outcomes.

### Blood Bank Staff

* Registration: Complete and submit personal details and medical history.
* Donor Management:Register new donors and update existing donor information. Verify donor eligibility and manage donor deferrals, send reminders and notifications to donors regarding their eligibility and upcoming donation dates.
* Blood Inventory Management: Monitor and update the status of blood units (e.g., available, reserved, used, expired).
* Blood Testing and Quality Control: Record and manage any incidents or issues related to blood quality.
* Reporting and Analytics: Generate reports on donor demographics, blood inventory status, and usage trends.

### Admin

* User Management: Add, update, or remove user accounts (Donors, Receivers, Blood Bank Staff).
* System Configuration: Manage system notifications and alerts.
* Support and Maintenance: Provide technical support and assistance to users also perform regular system maintenance and updates.

## Non-Functional Requirement

### Usability

* The user interface should be intuitive and easy to navigate, enabling users to access and retrieve necessary information without specialized training. Multilingual support should be available to cater to diverse user requirements.

### Accuracy

* All data related to blood donors, donations, inventory, and recipients must be accurate, consistent, and dependable to ensure proper functioning of the system and safety of patients.

### Availability

* The system must be operational during all working hours of the blood bank, with a recovery time of under one hour in case of failure. System responses to user requests should be processed in under two seconds.

### Maintainability

* The system should be designed for ease of maintenance, allowing quick and simple updates or modifications. It should also be portable to different environments with minimal adjustments.

### Security

* The system should implement strong security protocols to protect sensitive data related to donors and patients. Role-based access control should be enforced to prevent unauthorized access, with encryption for sensitive data storage and transmission.

### Scalability

* The system should be capable of handling increased loads, such as a growing number of users, donor registrations, and blood inventory, without significant performance degradation.

### Performance

* The system must ensure fast and efficient processing of transactions, including data retrieval, donor registration, and blood inventory updates, with minimal delay, even during peak usage times.

# Design and Implementation Constraints

## Use case diagram

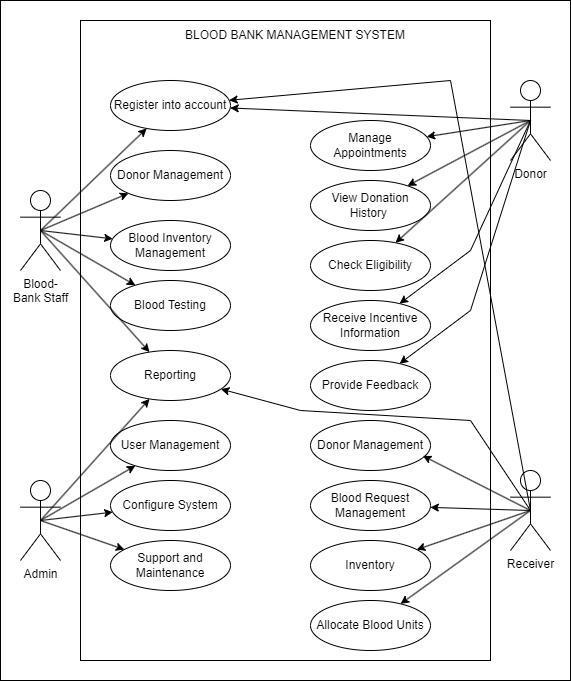


Figure 2.1‑ Use Case diagram for Blood Bank Management System

## Activity diagram and Swimlane diagram

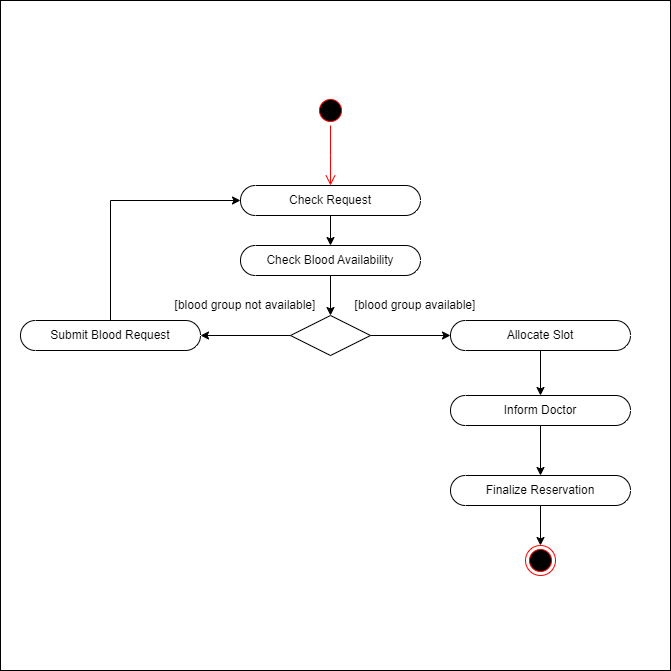


Figure ‑ Activity diagram for Blood Allocation

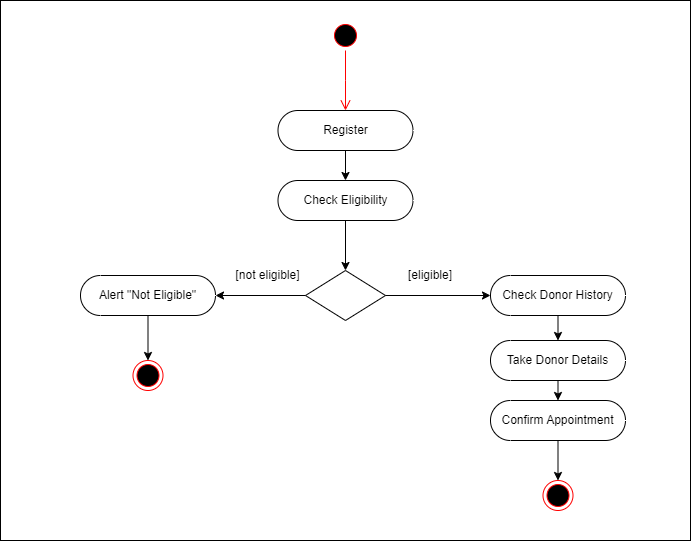


Figure 2.2‑2 Activity diagram for Donor Appointment Scheduling

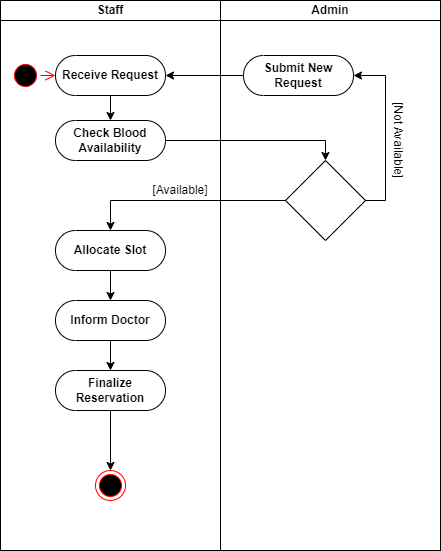


Figure 2.2-3 Swimlane diagram for Blood Allocation

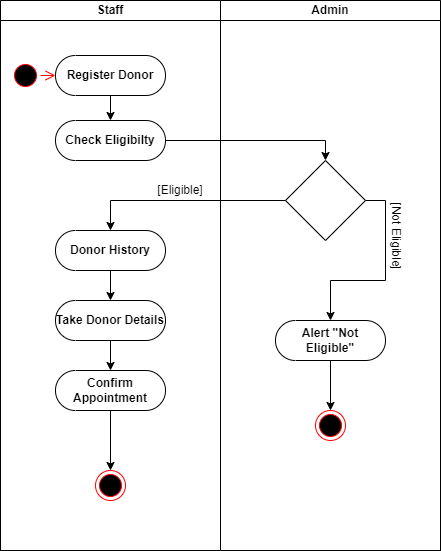


Figure ‑4 Swimlane diagram for Donor Registration

## Sequence diagram

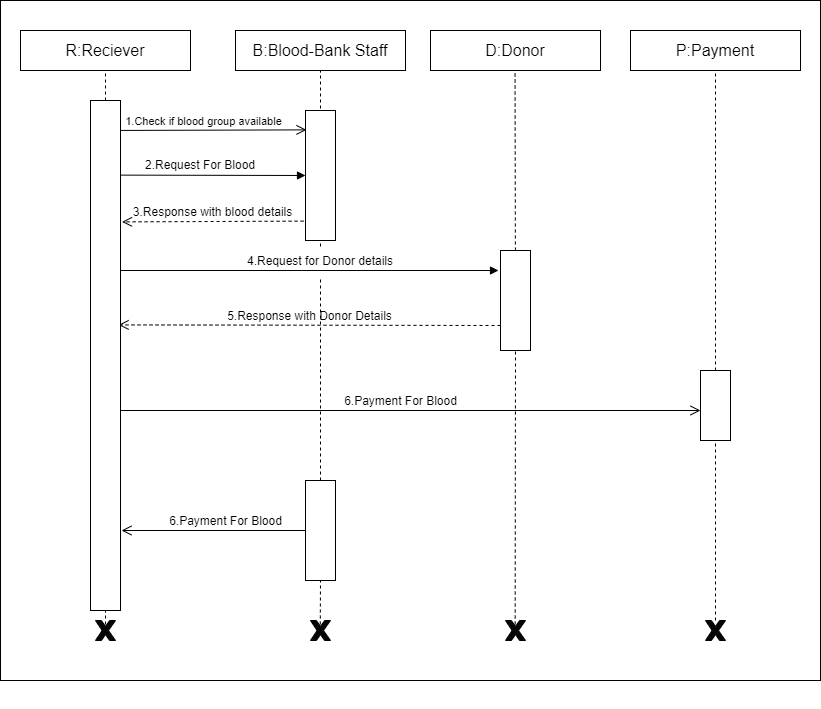
‑

Figure 2.3-1 Sequence diagram for Blood Allocation

## State diagram

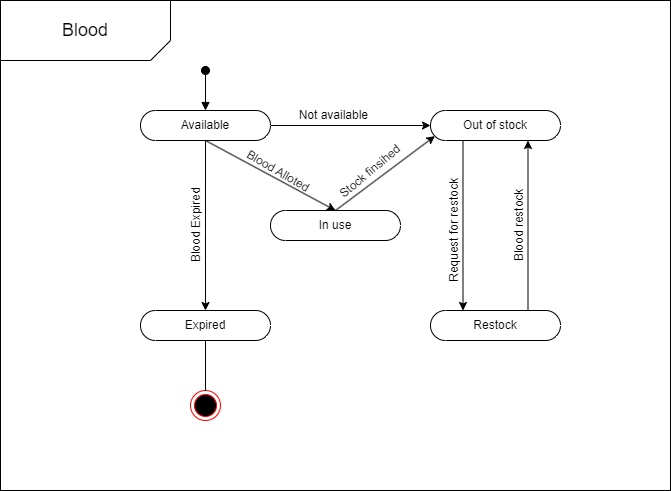


Figure ‑ State diagram of Blood

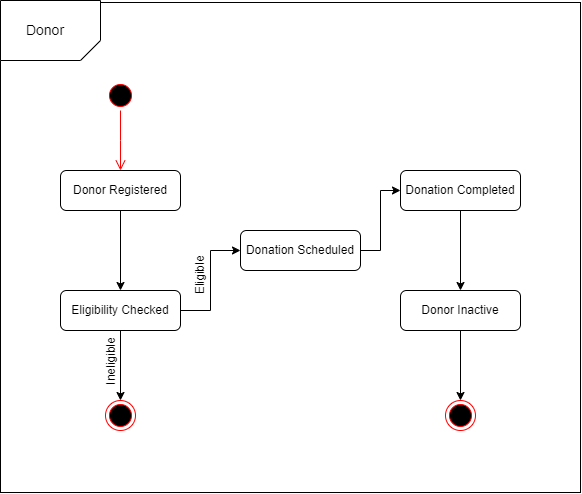


Figure ‑ State diagram for Donor

## Class diagram

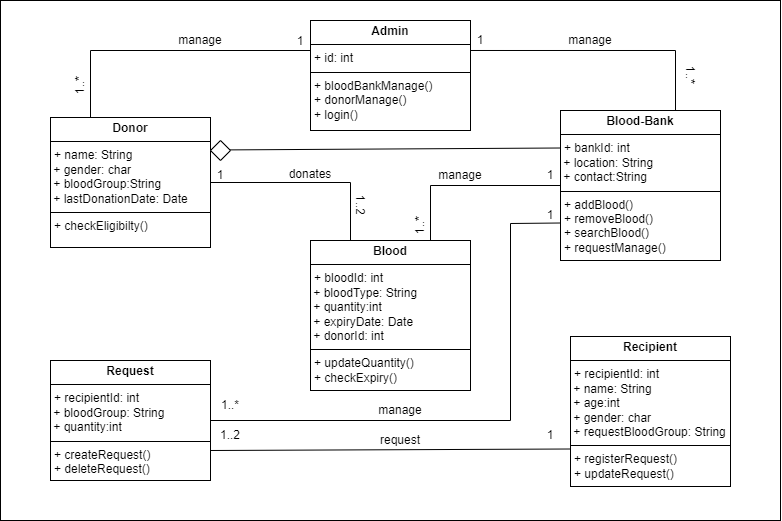


Figure ‑ Class diagram for Blood bank management system

## Data flow diagram

### DFD level 0

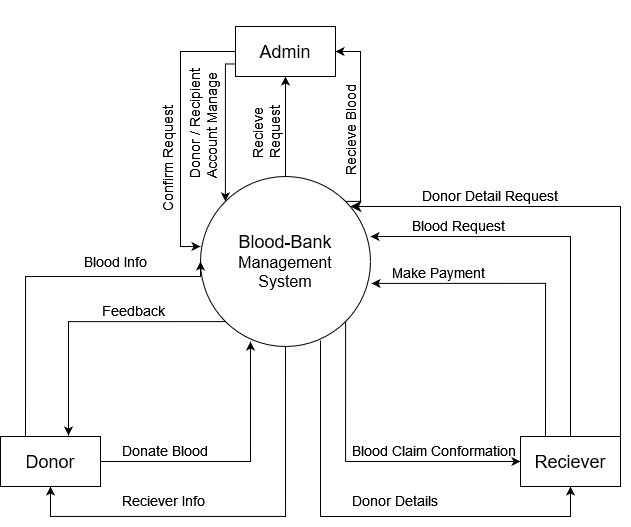


Figure ‑ DFD Level 0 diagram for Blood bank management system

### DFD Level 1

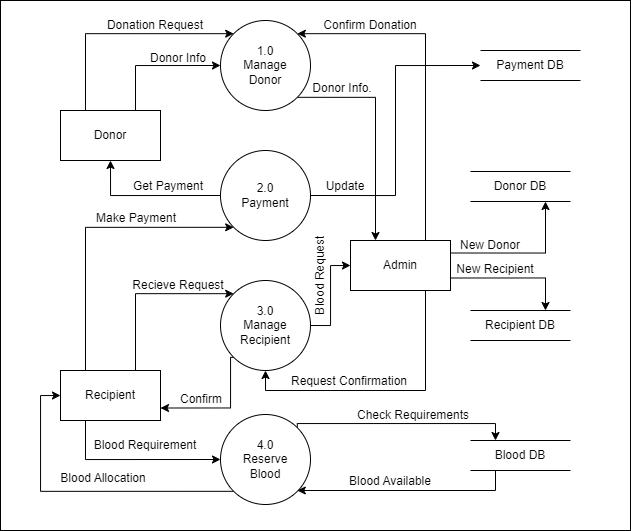


Figure ‑ DFD level-1 for Blood bank management system

# External interface requirement (Screens)

## Screen-1: Blood Request Form

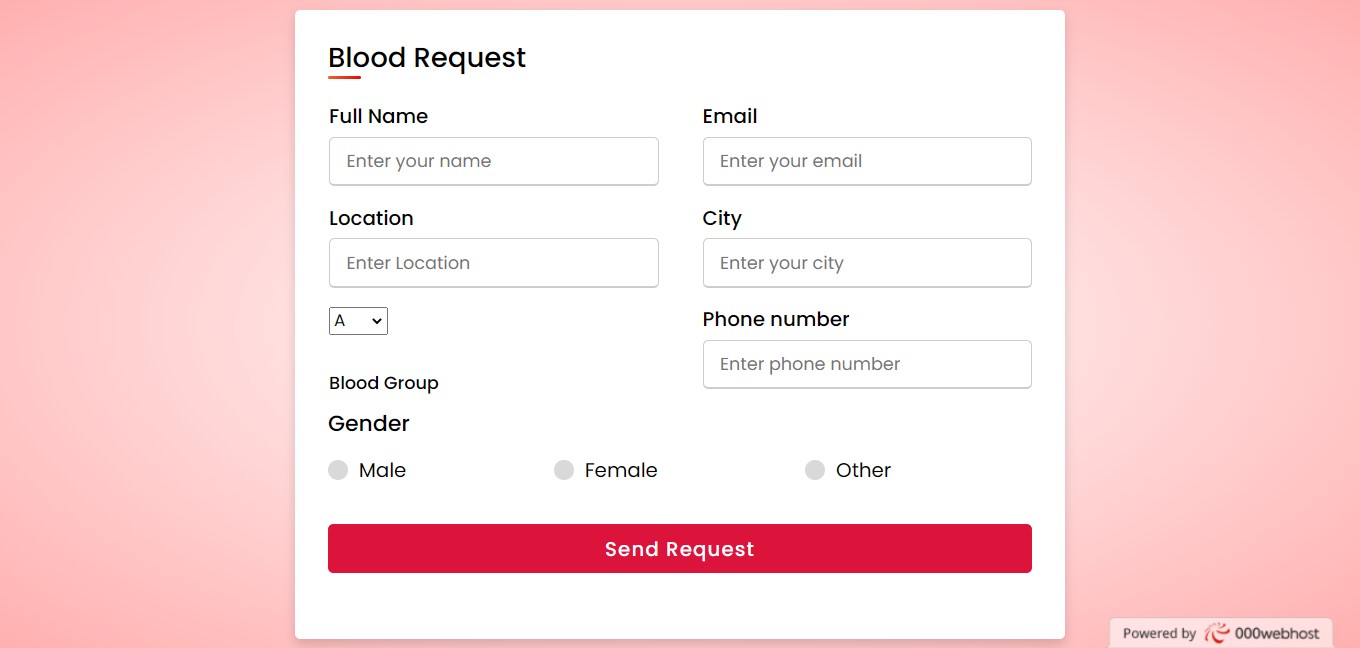


Figure ‑ Screen-1: Blood Request Form

**Purpose:** This form will allow the target end-users to register a blood-request in the system. To register, the following information will be encoded in the system.

Table ‑ Screen element of Registration form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Full name | Textbox | M | 1 | Full name field should be editable and accept the Full name. |
| 2 | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 3 | Location | Textbox | M | 1 | Location field should be editable. |
| 4 | City | Textbox | M | 1 | City field should be editable. |
| 5 | Blood Group | Dropdown | M | 1 | Blood Group options should be viable and updated timely. |
| 6 | Phone Number | Textbox | M | 1 | Phone Number should be editable and be exactly of 10 digits. |
| 7 | Gender | Radio | ---- | ---- | Gender can be selected from given options. |
| 8 | Send Request | Button | ---- | ---- | Checks the request details and registers it into database. |

## Screen-2: Login Form

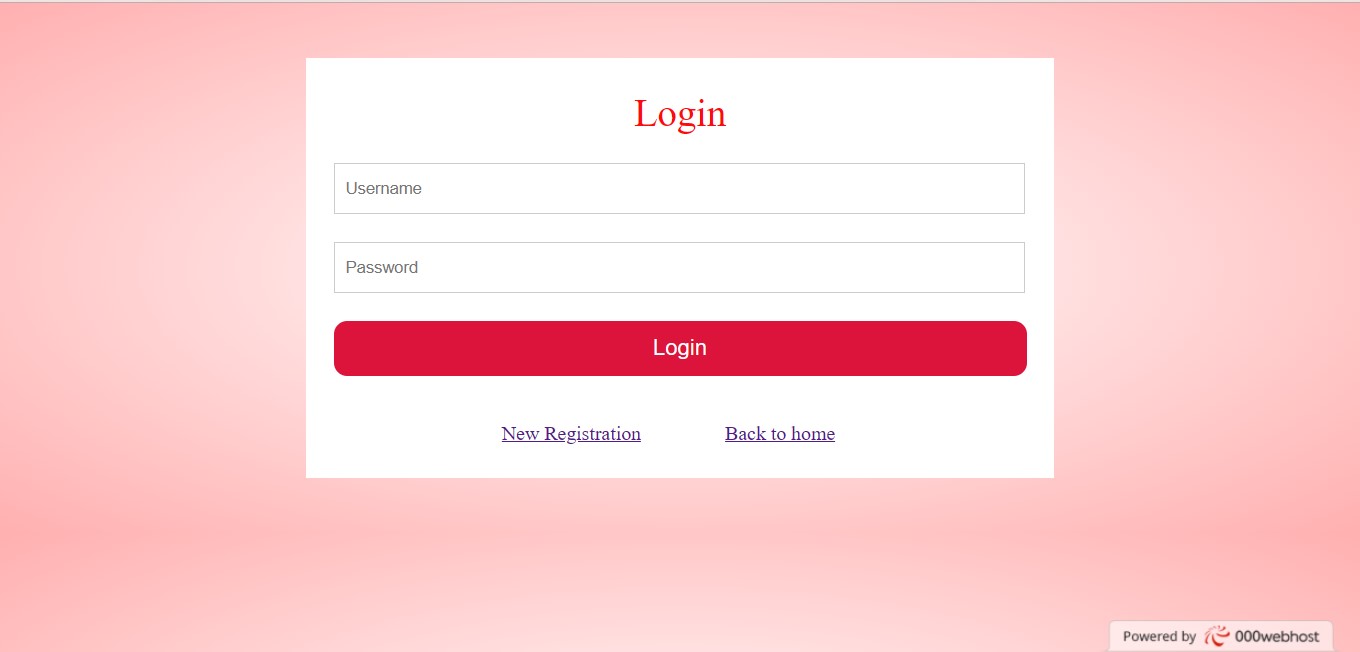


Figure ‑ Screen-2: Login Form

**Purpose:** This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

Table ‑ Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 3 | Login | Button | ------ | ------ | Login button navigates to another page if login credentials are valid. |
| 4 | New Registration | Link | ------ | ------ | Link for navigate to membership registration. |
| 5 | Back to home | Link | ------ | ------ | Link for navigate to homepage. |

## Screen-3: Donor Registration

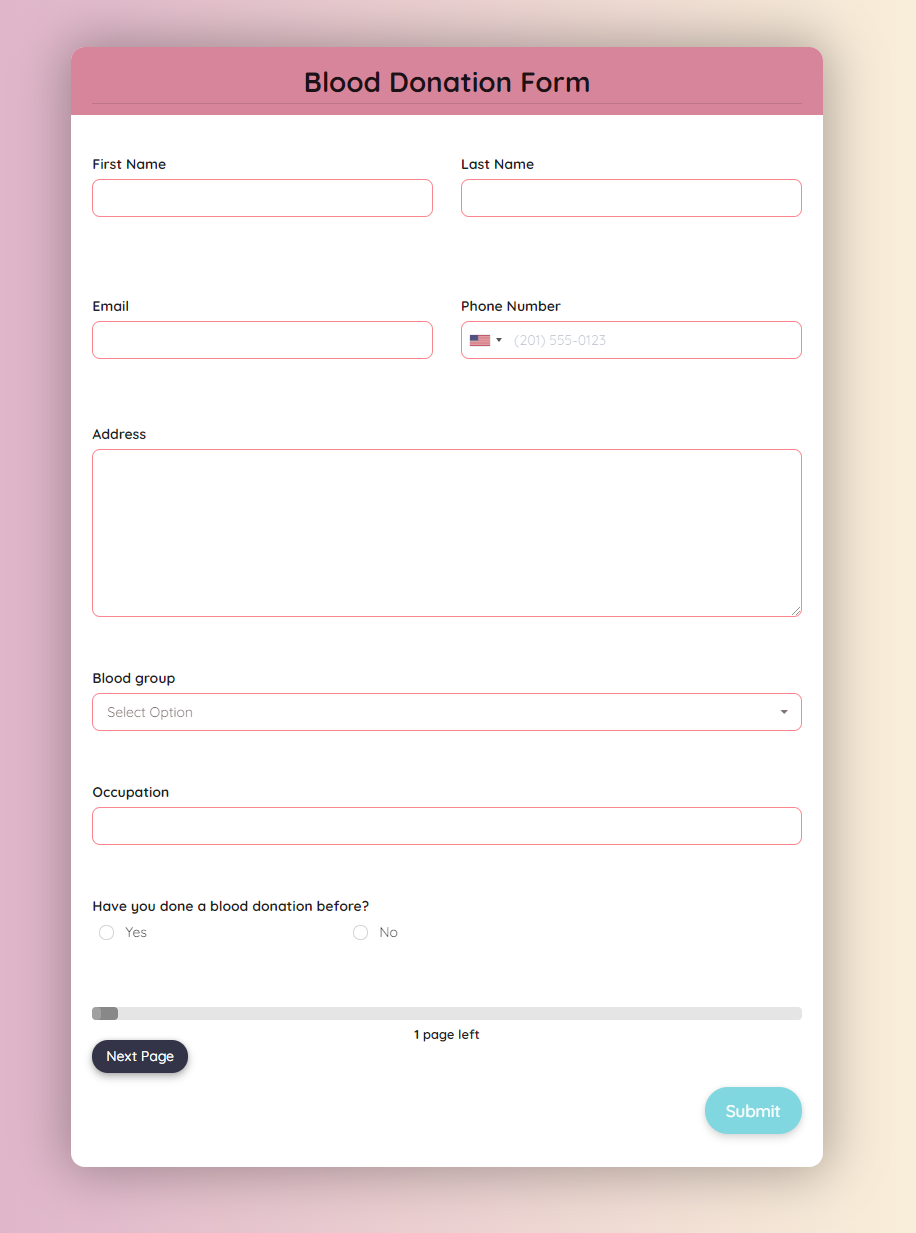


Figure ‑ Screen-3: Donor Registration

**Purpose:** This form will allow the target end-users to register a donation in the system. To register, the following information will be encoded in the system.

Table ‑ Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | First Name | Textbox | M | 1 | First Name field should be editable and accept the First Name. |
| 2 | Last Name | Textbox | M | 1 | Last Name field should be editable and accept the Last Name. |
| 3 | Email Address | Textbox | M | 1 | Email Address field should be editable and accept the email with proper format. |
| 4 | Address | Textbox | M | 1 | Address field should be editable. |
| 5 | Phone Number | Textbox | M | 1 | Phone Number field should be editable and accept only 10-digit phone number. |
| 6 | Blood Group | Dropdown | M | 1 | List of Blood Groups that can be donated. |
| 7 | History | Radio | ---- | ---- | Checks if donor has donated blood in the past, answers in Yes/No. |
| 8 | Next Page | Button | ---- | ---- | Traverses to the next page. |
| 9 | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |

## Screen-4: Search Donor Availability

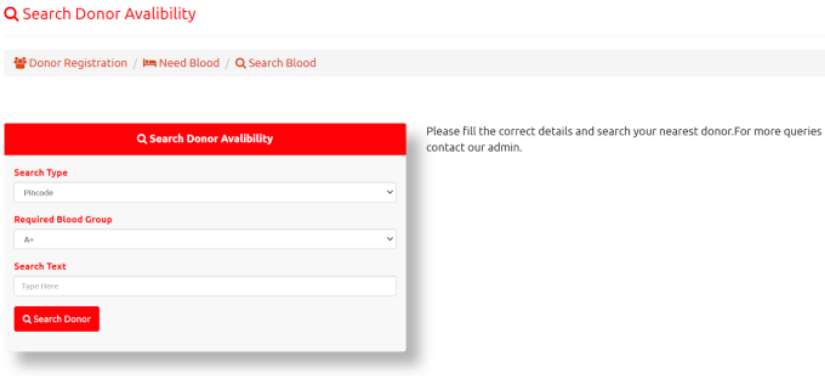


Figure 3.4-1 Screen-4: Search Donor Availability

**Purpose:** The registered user can search for an active donor directly by entering the proper blood type and location. The Active Donors Database is frequently updated by the Blood Bank Admin.

Table 3.4‑ Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Pin code | Dropdown | M | 1 | Phone Number field should be selectable and must have valid values. |
| 2 | Blood Group | Dropdown | M | 1 | List of Blood Groups that are available for donation. |
| 3 | Search Text | Textbox | M | 1 | User may search for details of a registered donor. (Ex. Donor Name) |
| 4 | Search | Button | ----- | ----- | Search is a button that may search for a registered donor based on the entered details. |

## Screen-5: Enquiry/Contact

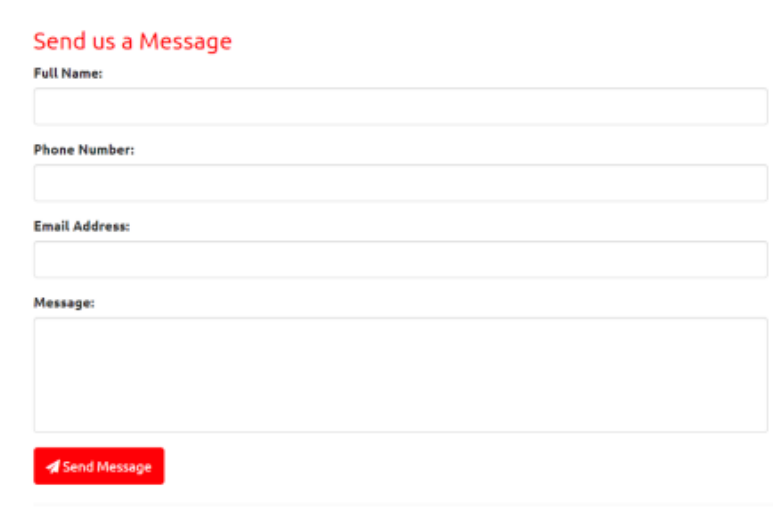


Figure 3.5-1 Screen-5: Enquiry/Contact

**Purpose:** The registered user can directly message the admin for any queries. The user can also contact via details from the main contact section.

Table 3.5‑ Screen Enquiry/Contact

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Full name | Textbox | M | 1 | Full name field should be editable and accept the Full name. |
| 2 | Phone Number | Textbox | M | 1 | Phone Number field should be editable and accept only 10-digit mobile number. |
| 3 | Email Address | Textbox | M | 1 | Email Address field should be editable and accept the email with proper format. |
| 4 | Message | Textbox | M | 1 | Address field should be editable and may contain longer string. |
| 5 | Send Message | Button | ----- | ----- | Send Message is a button for storing the entered data into database. |

# Database design

## List of Tables

* Donor
* Recipient
* Blood Request
* Blood
* Blood Bank Staff

Table 4.1-1 Table: Donor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constraints | Default Value & Description |
| DonorID | int | NN | PK (Auto Increment) | Unique identifier for each donor |
| DonorName | varchar(100) | NN |  | Full name of the donor |
| Gender | varchar(10) | AN |  | Gender of the donor |
| BloodGroup | varchar(10) | NN |  | Blood group (e.g., A+, O-) |
| Age | int | AN |  | Age of the donor |
| RecipientID | int | AN | FK | Unique identifier for each recipient |

Table ‑ Table: Recipient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constraints | Default Value & Description |
| RecipientID | int | NN | PK (Auto Increment) | Unique identifier for each recipient |
| RecipientName | varchar(100) | NN |  | Full name of the recipient |
| Gender | varchar(10) | AN |  | Gender of the recipient |
| BloodGroup | varchar(10) | NN |  | Blood group needed by the recipient |
| Age | int | AN |  | Age of the donor |
| DonorID | int | AN | FK | Unique identifier for each donor |

Table ‑ Table: Blood Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constraints | Default Value & Description |
| RequestID | int | NN | PK (Auto Increment) | Unique identifier for each request |
| RecipientID | int | NN | FK | Reference to **Recipient** table |
| BloodID | int | NN | FK | Reference to **Blood** table |
| Quantity | int | AN |  | Quantity of the blood requested |

Table ‑ Table: Blood

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| BloodID | int | NN | PK (Auto Increment) | Unique identifier for each blood group |
| DonorID | int | AN | FK | Unique identifier for each donor |
| BloodGroup | varchar(10) | NN |  | Blood group of the blood available |
| Quantity | int | AN |  | Quantity of the blood available |
| ExpiryDate | DateTime | AN |  | Expiry date of the blood |

Table ‑5 Table: Blood Bank Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constraints | Default Value & Description |
| StaffID | int | NN | PK (Auto Increment) | Unique identifier for each staff member |
| StaffName | varchar(100) | NN |  | Full name of the staff member |
| Designation | varchar(100) | AN |  | Job title or role (e.g., Admin, Nurse) |
| Contact | varchar(15) | AN |  | Phone number of the staff member |
| Gender | varchar(10) | AN |  | Gender of the employee |
| Salary | int | NN |  | Salary of the employee |

# 

# Stories and Scenario

## Story-1: Add New Blood Donor to the Blood Bank Database

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Blood Bank Administrator,  I want to add a new blood donor to the blood bank database  So that the donor's information is available for future blood donation and management |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | Adding a new donor is crucial for maintaining an up-to-date donor database and ensuring that the blood bank can contact donors for future blood drives. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Donor with Valid Information |
| Prerequisite | **:** | The administrator is logged into the Blood Bank Management System. |
| Acceptance Criteria | **:** | **Given:**  The administrator is navigated to the donor management page. Valid donor information, including name, contact details, blood type, and medical history, is available.  **When**: The administrator selects the "Add New Donor" option  **And**: The administrator enters valid donor details  **And**: The administrator clicks the "Save" button to add the donor to the database.  **Then**: The system successfully adds the donor to the database and the administrator receives a confirmation message with the donor's unique identification number. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New Donor with Invalid Information. |
| Prerequisite | **:** | The administrator is logged into the Blood Bank Management System. |
| Acceptance Criteria | **:** | **Given:** The administrator is on the donor management page  **When**: The administrator selects the "Add New Donor" option  **And**: The administrator enters incomplete or incorrect donor details  **And**: The administrator clicks the "Save" button to add the donor to the database.  **Then**: The system displays error messages for the incorrect or missing information, and the donor is not added to the database. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Attempting to Add a Duplicate Donor |
| Prerequisite | **:** | The administrator is logged into the Blood Bank Management System and is on the donor management page. |
| Acceptance Criteria | **:** | **Given**: The donor information, including name and contact details, is already present in the database.  **When**: The administrator selects the "Add New Donor" option.  **And**: The administrator enters details that match an existing donor in the database  **And**: The administrator clicks the "Save" button to add the donor.  **Then**: The system detects the duplicate and prompts the administrator with a warning message, and the donor is not added again. |

## Story-2: Update Blood Inventory Levels

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Blood Bank Technician,  I want to update the blood inventory levels  So that the blood bank can keep track of available blood types and quantities accurately. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Accurate tracking of blood inventory is essential for ensuring that sufficient blood supplies are available for medical needs and for managing inventory effectively. |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Updating Blood Inventory with Accurate Information |
| Prerequisite | **:** | The technician is logged into the Blood Bank Management System. |
| Acceptance Criteria | **:** | **Given**: The technician is on the blood inventory management page. Accurate blood inventory information, including blood type, quantity, and expiration date, is available.  **When**: The technician selects the "Update Inventory" option  **And**: The technician enters valid inventory details  **And**: The technician clicks the "Save" button to update the inventory.  **Then**: The system successfully updates the blood inventory levels, and the technician receives a confirmation message indicating the updated inventory details. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Updating Blood Inventory with Inaccurate Information |
| Prerequisite | **:** | The technician is logged into the Blood Bank Management System. |
| Acceptance Criteria | **:** | **Given**: The technician is on the blood inventory management page  **When**: The technician selects the "Update Inventory" option  **And**: The technician enters inaccurate or incomplete inventory details  **And**: The technician clicks the "Save" button to update the inventory.  **Then**: The system displays error messages for the inaccurate or incomplete information, and the inventory is not updated. |

### Scenario# S2.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Attempting to Update Inventory with Expired Blood |
| Prerequisite | **:** | The technician is logged into the Blood Bank Management System and is on the blood inventory management page. |
| Acceptance Criteria | **:** | **Given**: The blood inventory contains details of blood that is past its expiration date  **When**: The technician attempts to update the inventory with quantities of expired blood  **And**: The technician clicks the "Save" button.  **Then**: The system detects the expired blood and prompts the technician with a warning message, indicating that expired blood cannot be updated in the inventory, and the inventory is not updated with those quantities. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Blood Bank Management System | Test Designed by: | Akshat Patel |
| Module Name: | **User Registration** | **Test Designed date:** | 23-08-2024 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 23-08-2024 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: Web application should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Register with valid details | Functional | Register a new user with valid details in the blood bank system | TC\_001 |
| TC\_002 | Register with invalid email format | Functional | Register a new user with an invalid email format | TC\_002 |
| TC\_003 | Verify registration page elements | GUI | Verify that all elements are present on the registration page | TC\_003 |
| TC\_004 | Register with missing required fields | Functional | Verify that all fields required are filled on the registration page | TC\_004 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site should launch properly | Site launched successfully | Pass |  | <https://example.com/register> |  |
| 2 | Enter valid Username in the username field | Username field should be editable and accept input | Username input accepted | Pass |  | Username: [john.doe@example.com](mailto:john.doe@example.com) |  |
| 3 | Enter valid Email in the email field | Email field should be editable and accept input | Email input accepted | Pass |  | Email: [john.doe@example.com](mailto:john.doe@example.com) |  |
| 4 | Enter valid Password in the password field | Password field should be editable and accept input | Password input accepted | Pass |  | Password: SecurePass123 |  |
| 5 | Click on the Register button | User should be registered and redirected to a confirmation page | User registered successfully and redirected | Pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Register with invalid email format |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Attempt registration with an invalid email format | Error message should be displayed for invalid email format | Error message displayed | Pass |  | Email: john.doe@com |  |
| 2 | Verify error message for missing domain part in email | Display an error message indicating missing domain | Error message displayed | Pass |  | Email: john.doe@ |  |
| 3 | Verify error message for missing "@" in email | Display an error message for missing "@" in email | Error message displayed | Pass |  | Email: john.doeexample.com |  |

|  |  |
| --- | --- |
| **Test Case Title** | Verify registration page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given URL | The site should launch properly | Site launched successfully | Pass |  | <https://example.com/register> |  |
| 2 | Verify that the registration screen contains elements such as Username, Email, Password, and Register button | All listed controls displayed properly on the page | Registration page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on page load | Cursor should be focused in Username textbox | Cursor focused in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly | When tab is pressed, cursor should move to next control | Cursor moved to next control | Pass |  |  |  |
| 5 | Verify that all fields such as Username, Email have a valid placeholder | All text fields should have proper placeholders | All text fields have proper placeholders | Pass |  |  |  |
| 6 | Verify that labels float upward when the text field is in focus or filled (In case of floating label) | Labels should float above filled fields | Labels floated above filled fields | Pass |  |  |  |
| 7 | Verify that Register button is functioning correctly | Register button should submit the form | Register button functional. | Pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Register with missing required fields |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Attempt registration with missing Username field | Error message should be displayed for missing Username | Error message displayed | Pass |  | <https://example.com/register> |  |
| 2 | Attempt registration with missing Email field | Error message should be displayed for missing Email | Error message displayed | Pass |  |  |  |
| 3 | Attempt registration with missing Password field | Error message should be displayed for missing Password | Error message displayed | Pass |  |  |  |
| 4 | Verify that validation messages are properly shown for each missing field | Validation messages should be accurate and visible | Validation messages accurate | Pass |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml