

BloodCare CRM

Problem Statement

During medical emergencies, accidents, and critical healthcare situations, hospitals and patients often face difficulties in accessing the required blood on time. Many blood banks still rely on manual records, phone calls, or disconnected systems to manage donor details, blood availability, and blood requests. This leads to several operational challenges such as:

- Delays in identifying suitable blood donors
- Poor tracking of blood requests and their approval status
- Inefficient coordination between hospitals and blood banks
- Limited visibility of blood inventory and request data
- Difficulty in monitoring emergency blood requirements

These challenges can result in delayed medical treatment and reduced efficiency in emergency healthcare services.

To address these issues, this project proposes a **Salesforce-based BloodCare CRM** that centralizes blood bank, donor, and blood request information into a single platform. The system enables:

- Centralized management of blood banks and donor information
- Efficient handling and tracking of blood requests
- Automatic approval of emergency blood requests using automation
- Real-time monitoring through reports and dashboards

This solution improves coordination between blood banks and hospitals, reduces response time during emergencies, and supports data-driven decision-making in blood management operations.

Project Objectives

The main objectives of the Blood Care CRM are:

- To digitally manage blood banks, donors, and blood requests
 - To maintain structured and accurate records of blood donors
 - To reduce manual effort and delays in blood request processing
 - To automatically prioritize emergency blood requests
 - To provide real-time insights using Salesforce reports and dashboards
-

Phase 1: Problem Understanding & Industry Analysis

1. Requirement Gathering

Based on real-world healthcare and emergency blood management scenarios, the following functional requirements were identified for the BloodCare CRM system:

Blood Bank Management

- Blood Bank Name
- Location and contact details
- Association with donors and blood requests

Donor Management

- Donor Name and contact information
- Blood group details
- Last donation date
- Association with a blood bank

Blood Request Management

- Blood group required
- Quantity of blood needed
- Urgency level (Normal or Emergency)
- Request status (New, Approved, Fulfilled)
- Hospital name
- Association with a blood bank

Reporting & Monitoring

- Blood requests grouped by status
- Blood requests by blood group
- Donor distribution across blood banks
- Dashboard-based visualization for quick insights

Details

Description

API Name **Blood_Bank_c** Custom Singular Label **Blood Bank** Plural Label **Blood Banks**

Enable Reports Track Activities

Track Field History Deployment Status **Deployed** Help Settings Standard salesforce.com Help Window

Edit **Delete**

Details

Description

API Name **Blood_Request_c** Custom Singular Label **Blood Request** Plural Label **Blood Requests**

Enable Reports Track Activities

Track Field History Deployment Status **Deployed** Help Settings Standard salesforce.com Help Window

Edit **Delete**

2. Stakeholder Analysis

Stakeholder	Responsibilities
Salesforce Admin	Configuration of objects, fields, relationships, automation, reports, and dashboards

Stakeholder	Responsibilities
Blood Bank Staff	Manage donor records and handle blood requests
Hospitals	Raise blood requests and track request status
Donors	Provide blood and update donation details
Management	Monitor system data using reports and dashboards

3. Business Process Mapping

The overall business process flow of the BloodCare CRM system is as follows:

Blood Requirement Identified → Blood Request Raised → Urgency Evaluation → Emergency Auto-Approval → Donor & Blood Bank Coordination → Fulfillment → Reporting & Monitoring

This structured workflow ensures faster response during emergencies and smooth coordination between hospitals and blood banks.

4. Industry-Specific Use Case Analysis

- Healthcare Emergency Environment: Requires quick handling of urgent blood requests
- Time Sensitivity: Emergency cases demand immediate approval and processing
- Data Accuracy: Correct donor and blood group data is critical
- Transparency Needs: Reports and dashboards help management monitor requests

Salesforce CRM offers a secure, cloud-based, and scalable platform that supports efficient blood management operations.

5. AppExchange Exploration

Several existing solutions were analyzed:

- Healthcare Management Systems: Too complex for focused blood management
- Hospital ERP Tools: Designed for complete hospital operations
- Custom Salesforce CRM: Lightweight, configurable, and scalable

Hence, a custom Salesforce CRM solution was selected as the most suitable approach for this project.

Why This Project is Simple Yet Impactful

- Uses core Salesforce Admin concepts
- Focuses on real-world healthcare needs
- Automation implemented without complex coding
- Easy to demonstrate and understand
- Scalable for future enhancements such as inventory tracking and notifications

Phase 2: Org Setup & Configuration

This phase focuses on preparing the Salesforce environment to support the Blood Care CRM application. Proper org setup and configuration ensure that the system is secure, scalable, and ready for customization and data management.

1. Salesforce Edition Selection

For this project, a **Salesforce Developer Edition** was used. The Developer Edition provides full access to Salesforce CRM features, making it suitable for learning, development, and project demonstrations without licensing cost.

Use Case:

Using Developer Edition allows administrators to create custom objects, fields, relationships, validation rules, flows, reports, and dashboards required for managing blood banks, donors, and blood requests.

2. Company Profile Setup

The organization's basic profile was configured to represent a healthcare support organization responsible for managing blood donation and distribution.

Use Case:

Company profile setup helps align the CRM system with the organization managing blood bank operations.

Configured details include:

- Organization name
 - Default language and locale
 - Time zone
-

3. Business Hours & Holidays

Business hours define the operational working time of the organization.

Use Case:

Although blood requests may occur at any time, standard business hours were configured to support reporting, scheduling, and future automation such as alerts and notifications.

- Default Salesforce business hours were used
 - Holidays can be configured in the future if required for reporting purposes
-

4. Fiscal Year Settings

The standard fiscal year provided by Salesforce was used for this project.

Use Case:

Fiscal year settings support time-based reporting and dashboard analysis of blood requests and donor activities.

5. User Setup & Licenses

The project uses the default system administrator user available in the Salesforce Developer Edition org.

Use Case:

A single admin user is sufficient for development and demonstration purposes. In real-world deployment, multiple users such as blood bank staff, hospital users, and managers can be created with appropriate roles and permissions.

6. Login Access Policies

Default Salesforce login and security policies were retained.

Use Case:

Salesforce provides built-in security features including password policies, session timeouts, and login restrictions, ensuring secure access to the BloodCare CRM system.

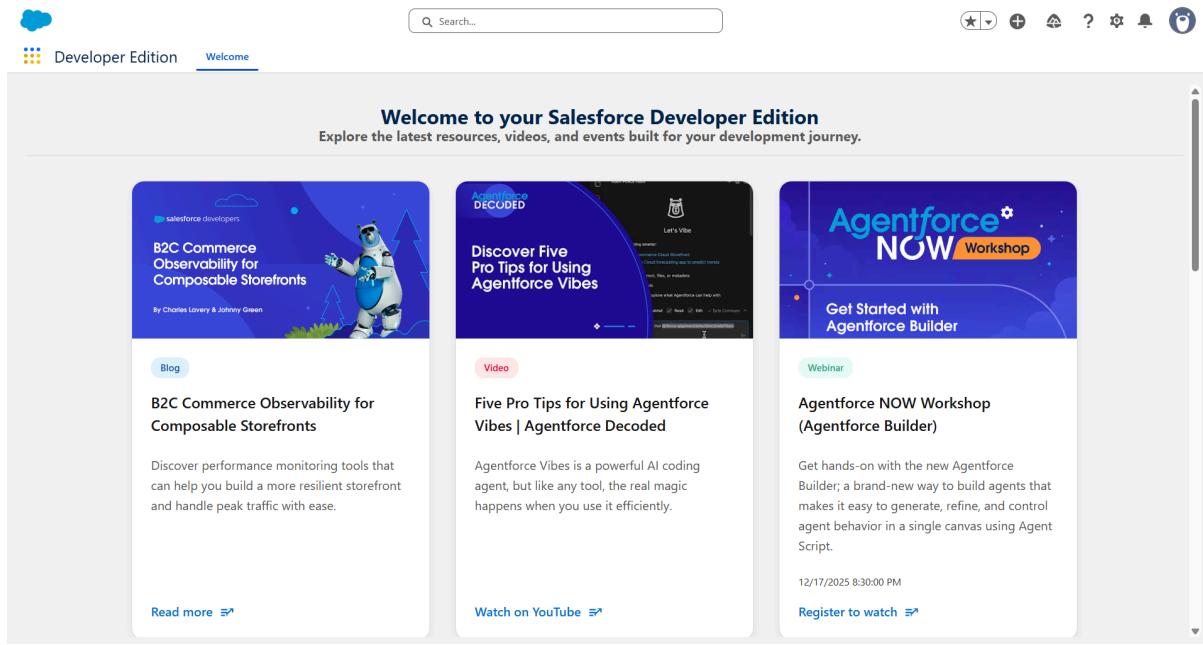
7. Development Org Setup

A dedicated Salesforce Developer Org was used exclusively for building the BloodCare CRM project.

Use Case:

Using a development org ensures:

- Safe experimentation
- No impact on production data
- Easy reset and modification during learning



8. Sandbox Usage

No sandbox was used for this project.

Justification:

Since this is a learning and demonstration project developed in a Developer Edition org, all configurations were done directly in the same environment.

9. Deployment Basics

Deployment was not required for this project as it was developed and demonstrated within a single Salesforce org.

Phase 3: Data Modeling & Relationships

This phase focuses on designing the data structure of the **BloodCare CRM** system. Proper data modeling ensures efficient storage, retrieval, and relationship management between blood banks, donors, and blood requests.

1. Standard & Custom Objects

Salesforce standard objects were not sufficient to represent blood donation and blood request management processes. Therefore, custom objects were created to meet the project requirements.

Custom Objects Created:

1. Blood Bank

- o Represents blood banks that store and manage blood units.

2. Donor

- o Stores information about blood donors associated with a blood bank.

3. Blood Request

- o Stores blood requirement details raised by hospitals or medical centers.

Use Case:

Custom objects allow the CRM to manage blood-bank-specific data that is not supported by standard Salesforce objects.

2. Fields

Each custom object includes relevant fields to capture required information.

Blood Bank Fields

- Blood Bank Name (Text – Standard Name field)
- Location (Text)
- Contact Number (Phone)

Donor Fields

- Donor Name (Text – Standard Name field)
- Blood Group (Picklist)
- Phone (Phone)

- Email (Email)
- Last Donation Date (Date)
- Blood Bank (Lookup to Blood Bank)

Blood Request Fields

- Blood Group Required (Picklist)
- Quantity (Number)
- Urgency Level (Picklist – Normal, Emergency)
- Status (Picklist – New, Approved, Fulfilled)
- Hospital Name (Text)
- Blood Bank (Lookup to Blood Bank)

Use Case:

These fields capture structured data required for blood donor management, emergency request handling, and reporting.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Blood Bank	Blood_Bank__c	Lookup(Blood Bank)		✓
Blood Group Required	Blood_Group_Required__c	Picklist		
Blood Request Name	Name	Text(80)		✓
Created By	CreatedById	Lookup(User)		
Hospital Name	Hospital_Name__c	Text(100)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

Blood Bank

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Blood Bank Name	Name	Text(80)		✓
Contact Number	Contact_Number__c	Phone		▼
Created By	CreatedById	Lookup(User)		▼
Last Modified By	LastModifiedById	Lookup(User)		▼
Location	Location__c	Text(100)		▼
Owner	OwnerId	Lookup(User,Group)		✓

Donor

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Blood Bank	Blood_Bank__c	Lookup(Blood Bank)		✓
Blood Group	Blood_Group__c	Picklist		▼
Created By	CreatedById	Lookup(User)		▼
Donor Name	Name	Text(80)		✓
Email	Email__c	Email		▼
Last Donation Date	Last_Donation_Date__c	Date		▼
Last Modified By	LastModifiedById	Lookup(User)		▼

3. Record Types

Record Types were not implemented in this phase.

Justification:

The current scope of the project does not require multiple business processes within the same object. A single record type is sufficient for managing bloodcare camp data.

4. Page Layouts

Custom page layouts were configured for each object to improve usability.

Blood Request Page Layout

- Blood request details section
- Urgency level and status fields
- Blood bank and hospital details

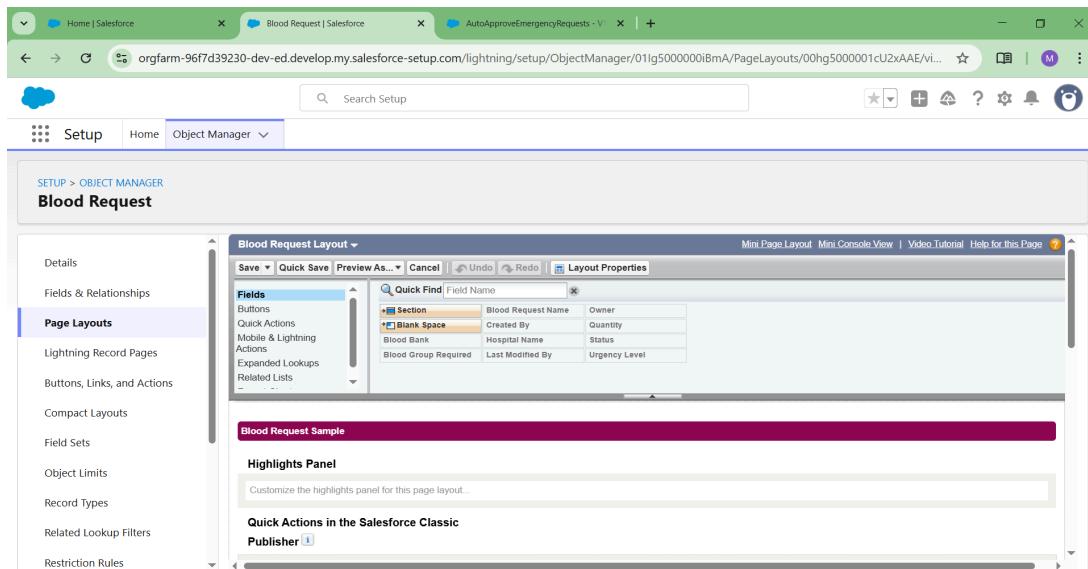
Donor Page Layout

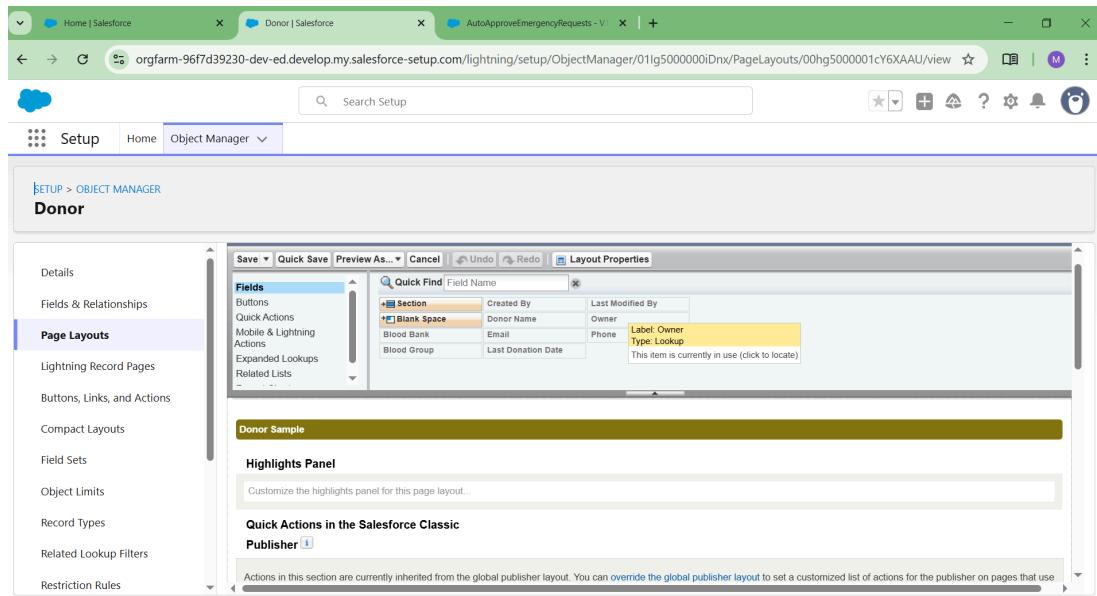
- Donor personal information
- Blood group and last donation date
- Associated blood bank

Blood Bank Page Layout

- Blood bank information
- Related donors
- Related blood requests

Use Case: Page layouts improve data entry efficiency and provide a clear view of related record





5. Compact Layouts

Compact layouts were not customized.

Justification:

Default compact layouts are sufficient for this project and support record highlights in Lightning Experience.

6. Schema Builder

Schema Builder was used to visually verify object relationships.

Use Case:

Schema Builder provides a graphical representation of how Blood Bank, Donor, and Blood Request objects are connected, helping validate the data model.

7. Relationships

Lookup Relationships Implemented

1. Donor → Blood Bank

- o Each donor is associated with one blood bank.

2. Blood Request → Blood Bank

- o Each blood request is linked to a blood bank responsible for fulfilling it.

Use Case:

Lookup relationships provide flexibility, allowing records to exist independently if needed.

8. Lookup vs Master-Detail vs Hierarchical Relationships

- **Lookup Relationship** was chosen over Master-Detail.
- Hierarchical relationships were not required.

Justification:

Lookup relationships allow independent record management and avoid automatic record deletion if a related record is removed.

9. Junction Objects

Junction objects were not implemented.

10. External Objects

External objects were not used.

Phase 4: Process Automation (Admin)

This phase focuses on automating key business processes in the **BloodCare CRM** system to reduce manual effort, ensure data accuracy, and improve efficiency in handling blood requests, especially during emergency situations.

1. Validation Rules

Validation rules were implemented to ensure that critical data is correctly entered before saving records.

Validation Rule Use Cases

Blood Request Quantity Validation

- o A validation rule was created on the Blood Request object to ensure that the quantity of blood requested is always greater than zero.

Use Case:

This prevents invalid or incorrect blood requests from being saved and ensures data accuracy, which is critical in emergency healthcare scenarios.

The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** Home | Salesforce, Blood Request | Salesforce, AutoApproveEmergencyRequests - V
- Search Bar:** Search Setup
- Navigation Bar:** Setup, Home, Object Manager
- Current Path:** SETUP > OBJECT MANAGER Blood Request
- Section:** Validation Rule Detail
- Rule Name:** Quantity_Must_Be_Positive
- Error Condition Formula:** Quantity__c <= 0
- Error Message:** Quantity must be greater than zero
- Description:** (empty)
- Created By:** Lakshmisri.Maddi, 12/19/2025, 2:11 AM
- Active:** checked
- Error Location:** Blood Bank
- Modified By:** Lakshmisri.Maddi, 12/19/2025, 2:11 AM
- Buttons:** Edit, Clone

2. Automation Using Salesforce Flow

Salesforce Flow was used to automate approval of emergency blood requests.

Flow Use Case: Auto-Approval of Emergency Requests

- A record-triggered flow was created on the Blood Request object
- When a blood request is created with **Urgency Level = Emergency**
- The system automatically updates the **Status** field to **Approved**

Workflow Rules, Process Builder, Approval Process, Email Alerts, Custom Notifications were not used in this project.

Phase 5: Apex Programming (Developer)

This phase focuses on server-side programming using Apex to handle complex business logic, asynchronous processing, and advanced automation.

However, **Apex programming was not implemented** in the current version of the **BloodCare CRM** project.

Although Apex was not implemented in the current phase, it can be added in future versions for:

- Automatic matching of donors based on blood group availability
- Bulk processing of blood requests during large-scale emergencies
- Scheduled data cleanup and reporting tasks
- Integration with external hospital or healthcare systems

Phase 6: User Interface Development

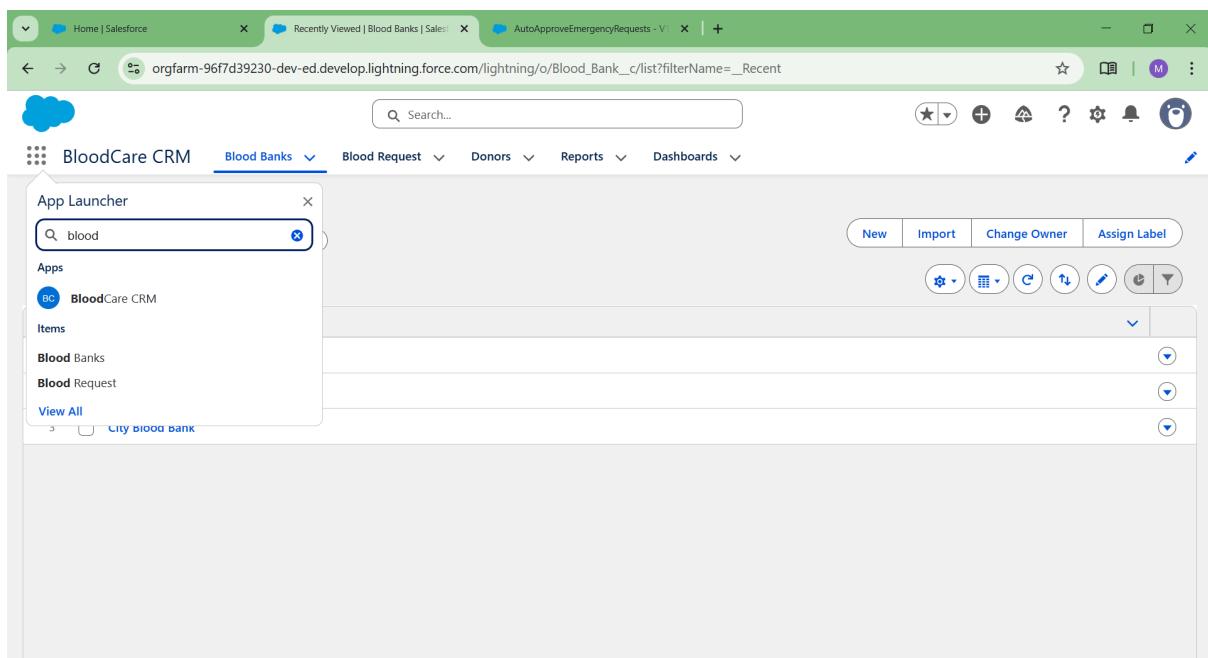
This phase focuses on designing a user-friendly interface using **Salesforce Lightning Experience**. The objective is to ensure that blood bank staff, hospital users, and administrators can easily navigate and use the **BloodCare CRM** system.

1. Lightning App Builder

A custom Lightning App named “**BloodCare CRM**” was created using Salesforce App Manager..

Use Case:

The custom app groups all relevant objects such as **Blood Banks, Donors, Blood Requests, Reports, and Dashboards** into a single workspace, improving accessibility and user productivity



2. Record Pages

Standard Lightning Record Pages were used for all custom objects.

Objects Covered:

- Blood Bank
- Donor
- Blood Request

Use Case:

Record pages display important information such as donor details, blood request urgency, status, and associated blood bank in a structured and easy-to-understand format.

3. Tabs

Custom tabs were created and added to the Lightning App for all custom objects.

Tabs Created:

- Blood Banks

- Donors
- Blood Requests

Use Case:

Tabs allow users to quickly access and manage records without navigating through Setup or Object Manager.

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'Custom Object Tabs' and lists three tabs: 'Blood Banks' (Bank style), 'Blood Request' (Red Cross style), and 'Donors' (Apple style). Below this are sections for 'Web Tabs' (No Web Tabs have been defined), 'Visualforce Tabs' (No Visualforce Tabs have been defined), and 'Lightning Component Tabs' (One tab: 'Get Started with Agentforce' in Heart style). A search bar at the top and a sidebar with 'User Interface' and 'Tabs' options are also visible.

4. Home Page Layouts

The default Salesforce Home page was used.

Use Case:

The Home page provides quick access to recent records, tasks, and reports.

5. Navigation Service

Default Salesforce navigation was used.

Use Case:

Users can seamlessly navigate between Blood Banks, Donors, Blood Requests, Reports, and Dashboards within the BloodCare CRM application.

6. Utility Bar, Lightning Web Components (LWC), Apex with LWC, Events in LWC, Wire Adapters, Imperative Apex Calls were not implemented.

Phase 7: Integration & External Access

This phase focuses on enabling communication between Salesforce and external systems using APIs, authentication mechanisms, and integration tools. Such integrations are commonly used to exchange data with external healthcare systems, hospital databases, or logistics platforms involved in blood supply management.

For the **BloodCare CRM** project, **external integrations were not implemented** in the current project scope.

Phase 8: Data Management & Deployment

This phase focuses on managing data efficiently within Salesforce and understanding basic deployment concepts. Proper data management ensures data accuracy, avoids duplication, and supports reporting and analytics within the **BloodCare CRM** system.

1. Data Import Wizard

The Salesforce **Data Import Wizard** was used to create and manage records in the system.

Use Case:

Blood Bank, Donor, and Blood Request records were primarily added manually during development and testing. The Data Import Wizard can also be used to import records using CSV files for bulk data entry in real-world scenarios.

2. Data Loader, Duplicate Rules, Data Export & Backup, Change Sets, Unmanaged vs Managed Packages, ANT Migration Tool, VS Code & Salesforce DX were not used in this project.

Phase 9: Reporting, Dashboards & Security Review

This phase focuses on analyzing data using Salesforce reports and dashboards and reviewing security configurations to ensure controlled access to blood donor and blood request information within the **BloodCare CRM** system.

1. Reports

Salesforce reports were created to analyze data captured in the BloodCare CRM application.

Report Types Used

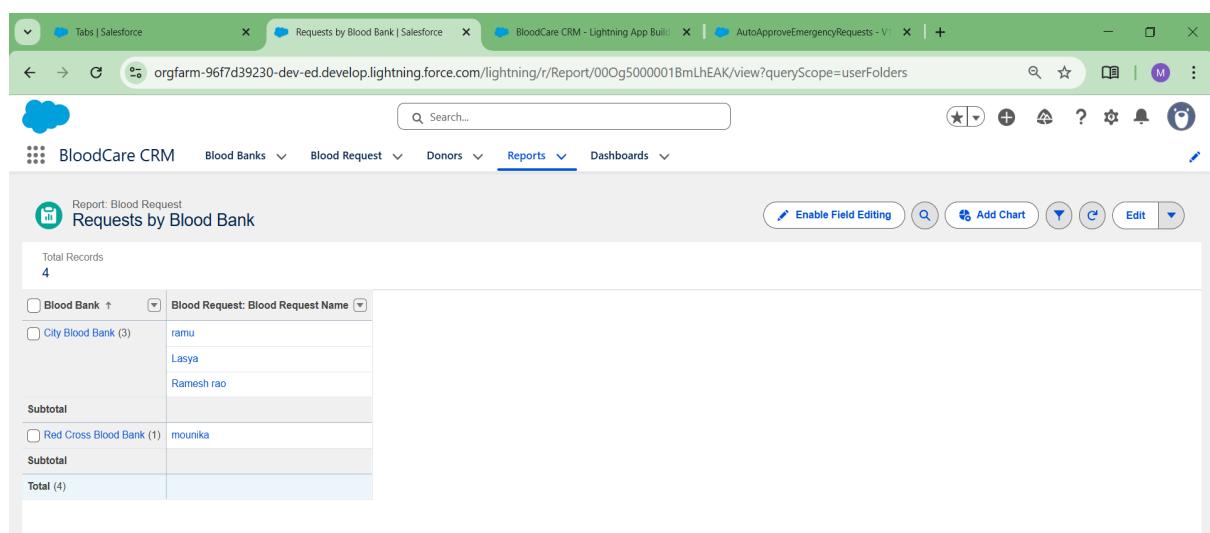
- Custom Object Reports

Sample Reports Created

1. **Blood Requests by Status**
2. **Blood Requests by Blood Group**
3. **Donor Distribution by Blood Bank**

Use Case:

Reports help administrators and management monitor blood request processing, donor availability, and blood bank performance.



The screenshot shows the Salesforce Lightning interface with multiple tabs open at the top. The active tab is 'Requests by Blood Bank | Salesforce'. Below the tabs, the page header includes the BloodCare CRM logo and navigation links for 'Blood Banks', 'Blood Request', 'Donors', 'Reports' (which is currently selected), and 'Dashboards'. The main content area displays a report titled 'Report: Blood Request Requests by Blood Bank'. The report summary shows 'Total Records 4'. A table lists the data, with the first three records under 'City Blood Bank' and one record under 'Red Cross Blood Bank'. The table columns are 'Blood Bank' and 'Blood Request: Blood Request Name'. The data is as follows:

Blood Bank	Blood Request: Blood Request Name
City Blood Bank (3)	ramu
	Lasya
	Ramesh rao
Subtotal	
Red Cross Blood Bank (1)	mounika
Subtotal	
Total (4)	

2. Report Formats

The following report format was used:

- **Tabular Report**
- **Summary Reports (where grouping was applied)**

Justification:

Tabular reports are simple and effective for quick viewing and exporting data.

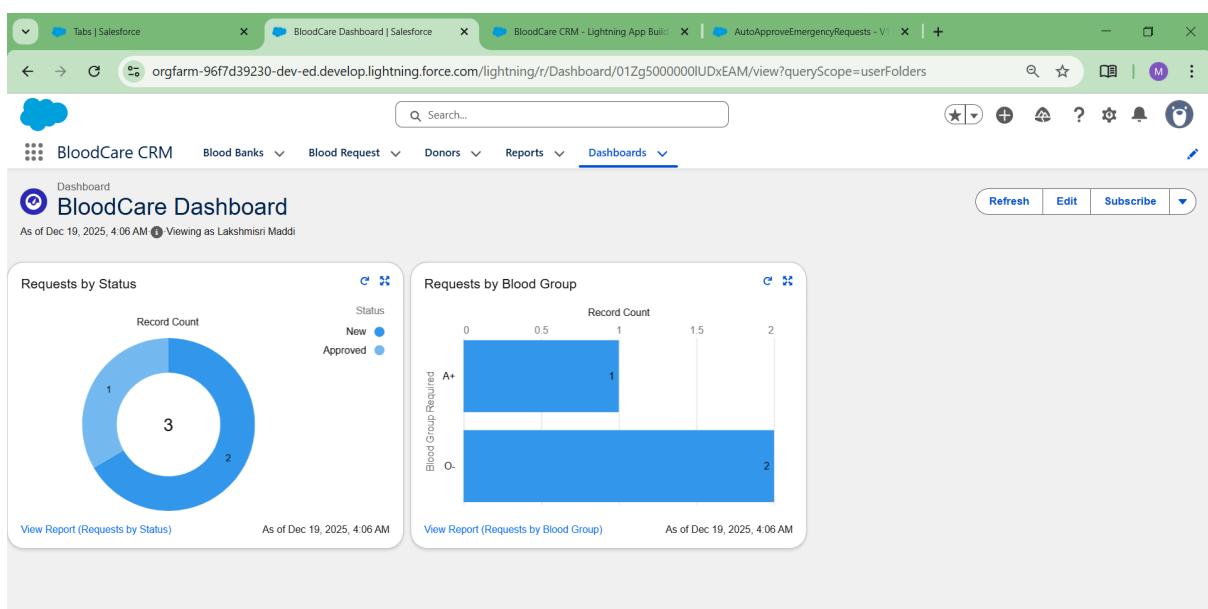
3. Dashboards

A basic dashboard was created using the above reports.

Use Case:

Dashboards provide a visual summary of:

- Blood requests by status
- Blood group-wise demand
- Donor distribution across blood banks



4. Dynamic Dashboards

Dynamic dashboards were not implemented.

Justification:

Since the project was developed in a single-user environment, role-based dynamic dashboards were not required.

5. Profiles

Standard Salesforce profiles were used.

Profiles Used:

- System Administrator
- Standard User (conceptually)

Use Case:

Profiles control object-level permissions and access within the BloodCare CRM system.

6. Roles

Role hierarchy was not configured.

Justification:

The project uses a single-user setup. Role hierarchy can be added in real-world deployments.

7. Users

Only one active user (Admin) was used.

Use Case:

Sufficient for development and demo purposes.

8. Permission Sets

Permission Sets were not created.

9. Organization-Wide Defaults (OWD)

Default organization-wide sharing settings were used.

Use Case:

OWD ensures baseline data visibility across records within the organization.

10. Sharing Rules

Sharing rules were not configured.

11. Field-Level Security

Field-level security was managed through default profile settings.

Use Case:

Controls visibility of sensitive donor and blood request information.

12. Session Settings

Default session settings were retained.

Use Case:

Ensures secure and stable user sessions.

13. Login IP Ranges

Login IP ranges were not restricted.

Phase 10: Quality Assurance Testing

This phase ensures that all implemented features of the **BloodCare CRM** system are tested thoroughly and presented effectively. Proper testing validates that the system functions as expected and demonstrates its business value in managing blood banks, donors, and blood requests.

1. Quality Assurance Testing

Testing was performed manually to verify the correct functioning of all implemented Salesforce features.

Testing Approach

- Manual testing using sample data
 - Each feature was tested by creating, updating, and viewing records
 - Validation rules and automation were tested using different input scenarios
 - Screenshots were captured for both input and output results
-

2. Test Cases

Test Case 1: Blood Bank Record Creation

- **Use Case:** Create a new Blood Bank
- **Test Steps (Input):**
 - Enter Blood Bank Name, Location, and Contact Number
 - Save the record
- **Expected Result:**
 - Blood Bank record is successfully created
- **Actual Result:**
 - Blood Bank record created successfully

Test Case 2: Donor Assignment to Blood Bank

- **Use Case:** Assign Donor to Blood Bank
 - **Test Steps (Input):**
 - Create a Donor record
 - Select Blood Bank using lookup field
 - **Expected Result:**
 - Donor is linked to the selected Blood Bank
 - **Actual Result:**
 - Donor successfully assigned to Blood Bank
-

Test Case 3: Blood Request Creation

- **Use Case:** Create a new Blood Request
 - **Test Steps (Input):**
 - Enter Blood Group Required, Quantity, Urgency Level
 - Select associated Blood Bank
 - Save the record
 - **Expected Result:**
 - Blood Request record saved successfully
 - **Actual Result:**
 - Blood Request record created successfully
-

Test Case 4: Validation Rule Execution

- **Use Case:** Prevent invalid blood request quantity
 - **Test Steps (Input):**
 - Try saving Blood Request with Quantity = 0
 - **Expected Result:**
 - Error message displayed
 - **Actual Result:**
 - Validation error shown as expected
-

Test Case 5: Flow Automation Execution

- **Use Case:** Auto-approve emergency blood requests
 - **Test Steps (Input):**
 - Create a Blood Request with Urgency Level = Emergency
 - **Expected Result:**
 - Status is automatically updated to Approved
 - **Actual Result:**
 - Status updated to Approved successfully
-

Test Case 6: Reports Generation

- **Use Case:** View Blood Requests by Status report
- **Test Steps (Input):**
 - Open reportOpen the report from Reports tab
- **Expected Result:**
 - Report displays accurate blood request data

- **Actual Result:**

- Report displayed successfully

The **BloodCare CRM** project successfully addresses key challenges related to blood management during medical emergencies. By leveraging Salesforce's declarative capabilities, the system enables efficient donor management, automated emergency request handling, and real-time reporting through dashboards.