

# Blood Bank database

## *Project's features*

The database system will manage donation, storage and distribution of blood between hospitals and blood banks as well as storage information about donors, recipients, hospitals and blood banks.

- Blood bank management – tracking available stock for each blood type and keeping basic information like capacity, name, contact information, etc.
- Donor management – storing information about donors including blood type, contact number, etc.
- Donation records- history of donations including donor, quantity of blood, expiration date, status, etc.
- Recipient management – information about recipients, what hospital they are in and their blood request.
- Blood request management – blood requests are linked to recipient, hospital and blood bank with available stock
- Hospital information – details about different hospitals working with blood banks
- Blood bank stock management- updating stock for each blood type after any operation as request, donation or reaching date of expiration

## *Requirements table*

Requirement ID	Requirement name	Description	Priority	Feature name
1	Tracking blood bank stock availability	Tracking blood availability for every blood type	High	Blood bank management
2	Storing donor information	Storing donors information along with eligibility	High	Donor management
3	Tracking history of donations	Tracking donations and keeping records with date, quantity etc.	Medium	Donation history

4	Storing recipient and hospital information	Information of recipient and the hospital he is in	High	Recipient management
5	Managing blood requests	Managing blood request tied to specific patient and blood bank	High	Blood request management
6	Updating blood stock	Updating blood stock after each donation/request or if date of expiration has been reached	High	Blood Bank stock management

## USER STORIES:

Targets: Donor, Hospital Staff, Blood Bank Staff, Recipient

- As a Donor, I want to register my blood type and contact information so I can be contacted for future donations.
- As a Hospital Staff, I want to request blood for a patient so they can receive timely treatment.
- As a Blood Bank Staff, I want to update the blood stock after a donation or request so the inventory is accurate.
- As a Recipient, I want my hospital to receive the blood I need based on my blood type.
- As a Blood Bank Staff, I want to track which blood types are available so I can manage incoming requests efficiently.
- As a Hospital Staff, I want to view and manage recipients and their blood needs.

## USE CASES:

### 1. REGISTER DONOR

Actor: Donations worker

Goal: Register a new donor with relevant information

Description: The donor provides their personal and blood information for registration.

Preconditions: 1) Donor must be eligible for donation

Standard scenario: Donor is registered and data is stored

Alternate scenario: Donor is found ineligible and not registered

Trigger: Donor submits registration form

## **2. REQUEST BLOOD**

Actor: Hospital worker

Goal: Request a specific blood type for a patient

Description: A hospital staff member requests blood for a recipient from a blood bank

Preconditions: 1) Recipient is already registered

2) Blood bank has sufficient stock

Standard scenario: Blood is assigned and request is recorded

Alternate scenario: Blood bank has no matching stock

Trigger: Blood request is submitted by hospital

## **3. UPDATE BLOOD STOCK**

Actor: Blood Bank worker

Goal: Keep the stock up to date after a donation or request

Description: Blood stock is updated manually or automatically based on operations

Preconditions: 1) Donation or request event occurred

Standard scenario: Stock is increased or decreased accordingly

Alternate scenario: Stock becomes negative or invalid

Trigger: Donation or request operation is completed

## **4. TRACK DONATION HISTORY**

Actor: Blood Bank worker

Goal: Review the donation history of a donor

Description: List all past donations including quantity and date

Preconditions: 1) Donor is registered

Standard scenario: Donation history is displayed

Alternate scenario: No history found

Trigger: Staff selects donor record

## **5. CHECK BLOOD AVAILABILITY**

Actor: Blood Bank worker

Goal: View current stock for each blood type

Description: Staff checks blood availability before fulfilling a request

Precondition: 1) Stock data is up to date

Standard scenario: Blood type availability is shown

Alternate scenario: Stock is outdated

Trigger: Staff initiates stock query

## **6. REGISTER BLOOD DONATION**

Actor: Blood Bank worker

Goal: Log a new donation into the system

Description: A donation is recorded with donor, date, quantity, and expiration

Precondition: 1) Donor is registered and eligible

Standard scenario: Donation is logged and stock updated

Alternate scenario: Data entry error

Trigger: Donation process is completed

## **7. REGISTER DONOR**

Actor: Donations worker

Goal: Register a new donor with relevant information

Description: The donor provides their personal and blood information for registration.

Preconditions: 1) Donor must be eligible for donation

Standard scenario: Donor is registered and data is stored

Alternate scenario: Donor is found ineligible and not registered

Trigger: Donor submits registration form

## **8. REQUEST BLOOD**

Actor: Hospital worker

Goal: Request a specific blood type for a patient

Description: A hospital staff member requests blood for a recipient from a blood bank

Preconditions: 1) Recipient is already registered 2) Blood bank has sufficient stock

Standard scenario: Blood is assigned and request is recorded

Alternate scenario: Blood bank has no matching stock

Trigger: Blood request is submitted by hospital

## **9. UPDATE BLOOD STOCK**

Actor: Blood Bank worker

Goal: Keep the stock up to date after a donation or request

Description: Blood stock is updated manually or automatically based on operations

Preconditions: 1) Donation or request event occurred

Standard scenario: Stock is increased or decreased accordingly

Alternate scenario: Stock becomes negative or invalid

Trigger: Donation or request operation is completed

## **10. TRACK DONATION HISTORY**

Actor: Blood Bank worker

Goal: Review the donation history of a donor

Description: List all past donations including quantity and date

Preconditions: 1) Donor is registered

Standard scenario: Donation history is displayed

Alternate scenario: No history found

Trigger: Staff selects donor record

### **11. CHECK BLOOD AVAILABILITY**

Actor: Blood Bank worker

Goal: View current stock for each blood type

Description: Staff checks blood availability before fulfilling a request

Precondition: 1) Stock data is up to date

Standard scenario: Blood type availability is shown

Alternate scenario: Stock is outdated

Trigger: Staff initiates stock query

### **12. REGISTER BLOOD DONATION**

Actor: Blood Bank worker

Goal: Log a new donation into the system

Description: A donation is recorded with donor, date, quantity, and expiration

Precondition: 1) Donor is registered and eligible

Standard scenario: Donation is logged and stock updated

Alternate scenario: Data entry error

Trigger: Donation process is completed

### **13. CREATE BLOOD BANK**

Actor: Blood Bank Manager

Goal: Register a new blood bank in the system

Description: A new blood bank is created with relevant details (name, location, capacity)

Preconditions: User has authorization

Standard scenario: Blood bank is successfully registered

Alternate scenario: Data entry is incomplete or duplicate

Trigger: Manager submits creation form

### **14. DELETE BLOOD BANK**

Actor: Blood Bank Manager

Goal: Remove an existing blood bank from the system

Description: A blood bank and its records are deleted

Preconditions: Blood bank exists and has no pending transactions

Standard scenario: Bank is removed from the system

Alternate scenario: Bank has active data or cannot be deleted

Trigger: Manager initiates deletion

### **15. UPDATE BLOOD BANK**

Actor: Blood Bank Manager

Goal: Modify existing blood bank information

Description: Location, contact, or stock-related details are updated

Preconditions: Blood bank exists

Standard scenario: Data is successfully updated

Alternate scenario: Data validation error

Trigger: Manager edits record

## **16. CREATE RECIPIENT**

Actor: Request Manager

Goal: Register a recipient in need of blood

Description: Recipient details and blood type are entered

Preconditions: None

Standard scenario: Recipient is added to the system

Alternate scenario: Recipient already exists

Trigger: Manager adds new recipient

## **17. DELETE RECIPIENT**

Actor: Request Manager

Goal: Remove a recipient's data

Description: A recipient's record is deleted from the system

Preconditions: Recipient exists

Standard scenario: Record is deleted

Alternate scenario: Deletion blocked due to active requests

Trigger: Manager initiates deletion

## **18. CREATE BLOOD REQUEST**

Actor: Request Manager

Goal: Submit a new blood request for a recipient

Description: A blood request is created based on recipient's need

Preconditions: Recipient is registered

Standard scenario: Request is logged in system

Alternate scenario: Stock unavailable or request invalid

Trigger: Manager submits request

## **19. FULFILL BLOOD REQUEST**

Actor: Request Manager

Goal: Match a request with available blood and fulfill it

Description: The system allocates blood to fulfill a request

Preconditions: Request and stock exist

Standard scenario: Request is fulfilled and stock updated

Alternate scenario: Stock is insufficient

Trigger: Manager initiates fulfillment

## **20. SEARCH BLOOD REQUEST**

Actor: Request Manager

Goal: View details of past or current blood requests

Description: Filter/search requests by recipient, date, or blood type

Preconditions: Request data exists

Standard scenario: Relevant requests are displayed

Alternate scenario: No match found

Trigger: Search query initiated

## **21. UPDATE BLOOD REQUEST**

Actor: Request Manager

Goal: Modify an existing blood request

Description: Change in quantity, urgency, or recipient details

Preconditions: Request exists

Standard scenario: Request is successfully updated

Alternate scenario: Request already fulfilled or locked

Trigger: Manager edits request

## **22. CREATE HOSPITAL**

Actor: Request Manager

Goal: Register a new hospital in the system

Description: Enter hospital's name, contact, and credentials

Preconditions: User has permission

Standard scenario: Hospital is added

Alternate scenario: Hospital already exists

Trigger: Form submission

## **23. DELETE HOSPITAL**

Actor: Request Manager

Goal: Remove a hospital's record

Description: Delete inactive or incorrect hospital entries

Preconditions: Hospital has no pending requests

Standard scenario: Record is deleted

Alternate scenario: Constraints prevent deletion

Trigger: Manager deletes record

## **24. SEARCH HOSPITAL**

Actor: Request Manager

Goal: Look up hospital records

Description: Query by name, location, or ID

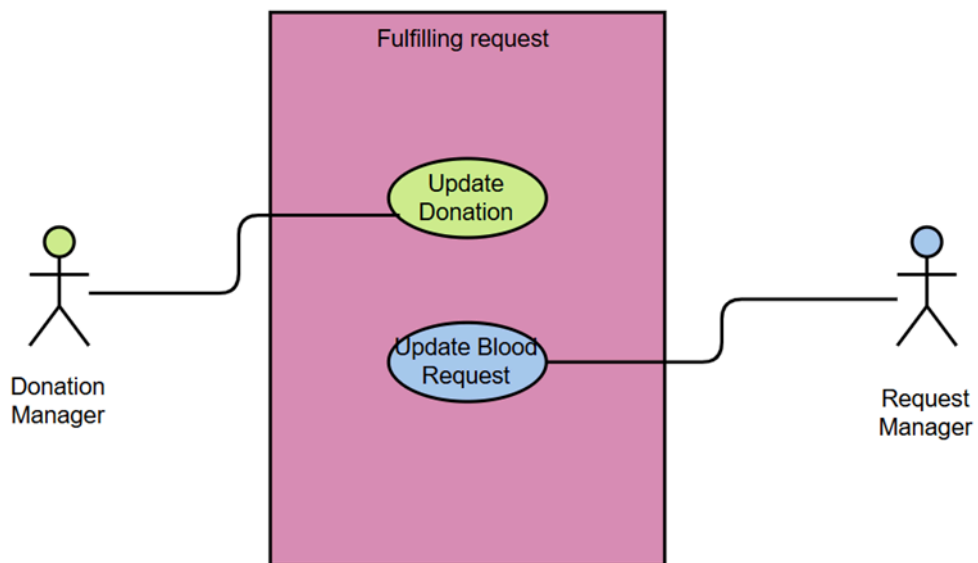
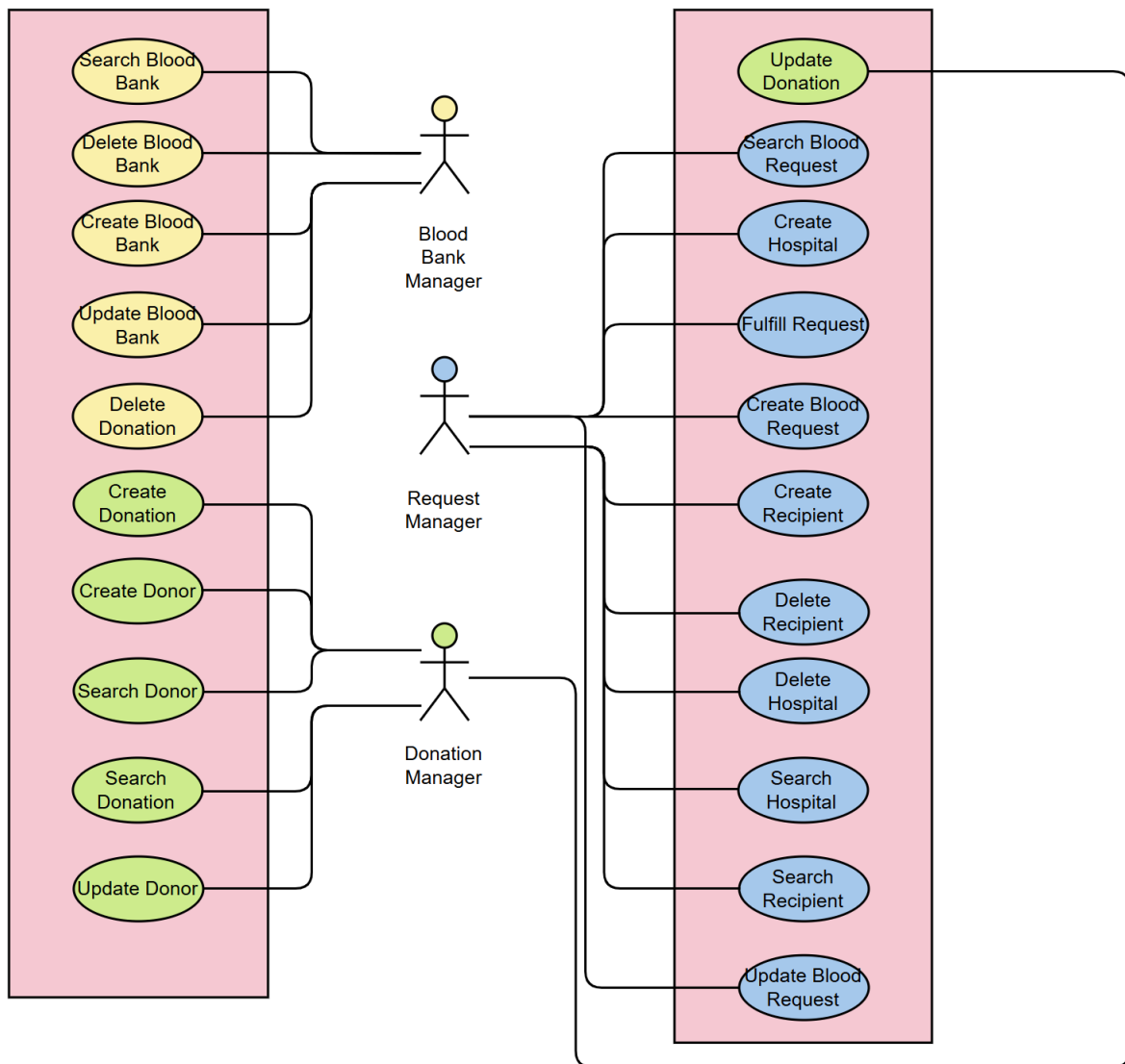
Preconditions: Hospital exists in system

Standard scenario: Matching hospital(s) shown

Alternate scenario: No results

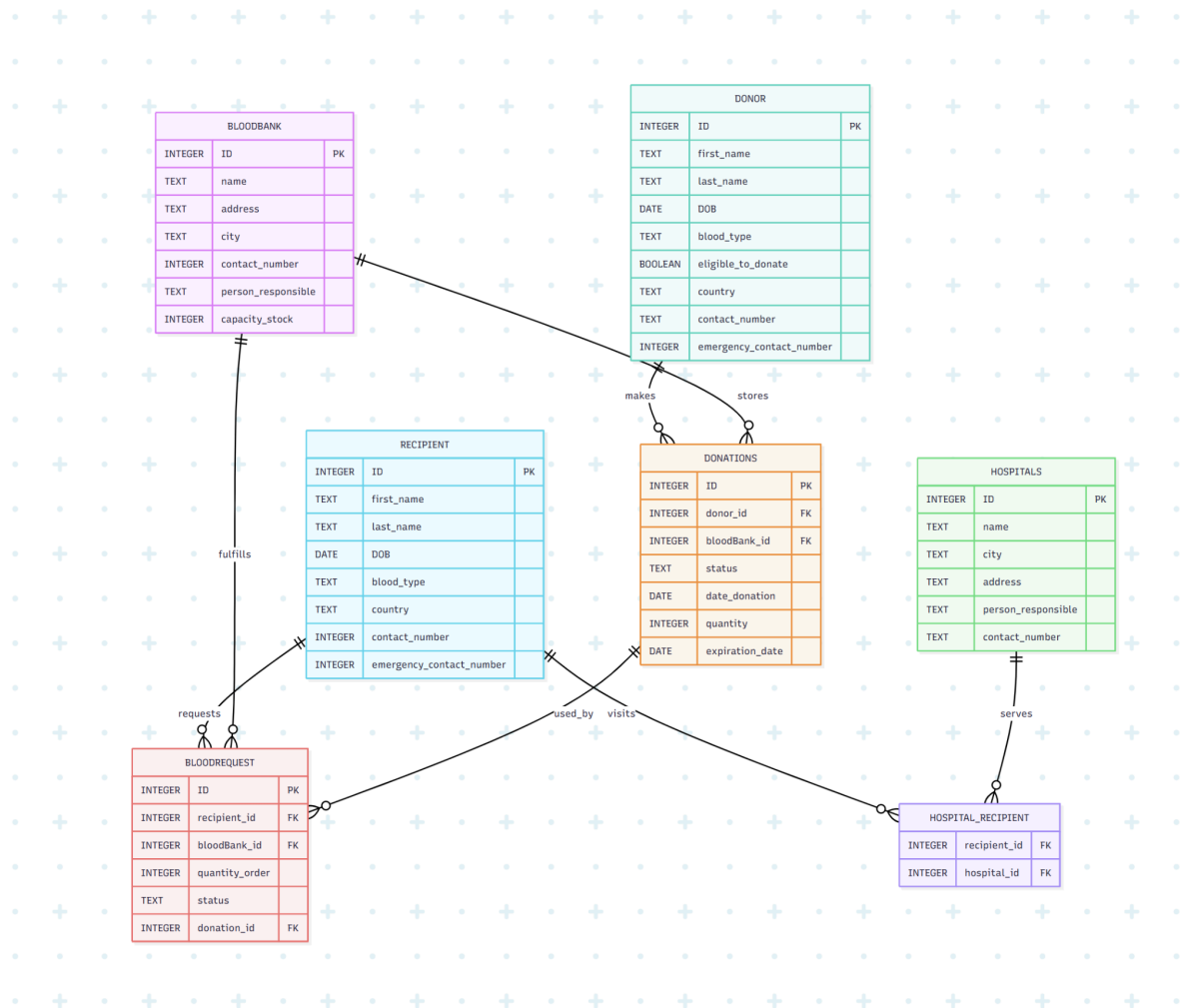
Trigger: Search input entered

USE- CASE diagram



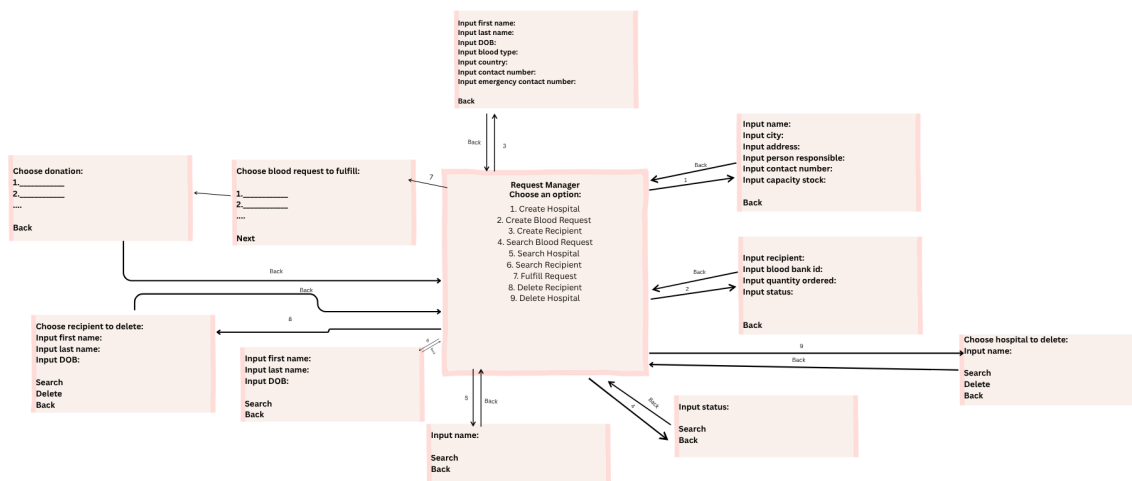


## ER DIAGRAM:



## UI mockups:

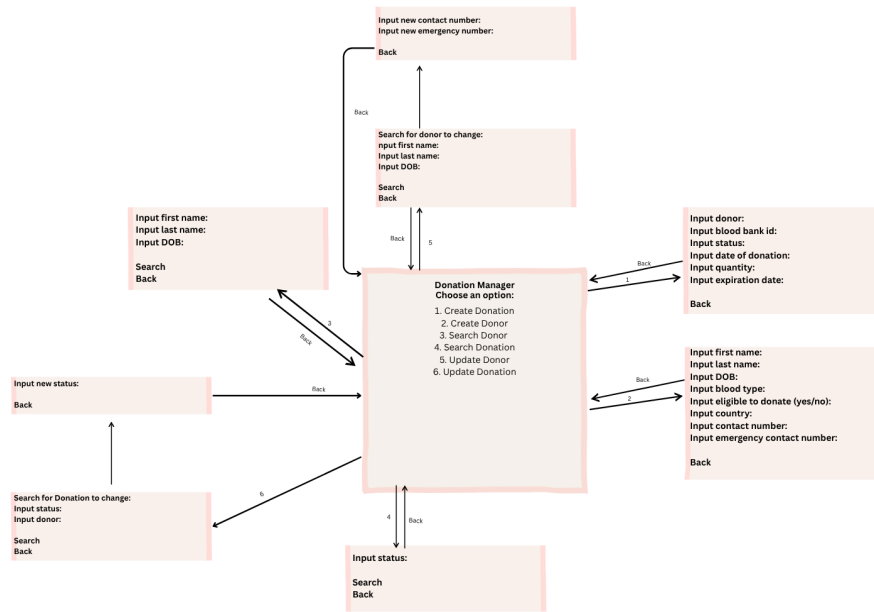
- REQUEST MANAGER:



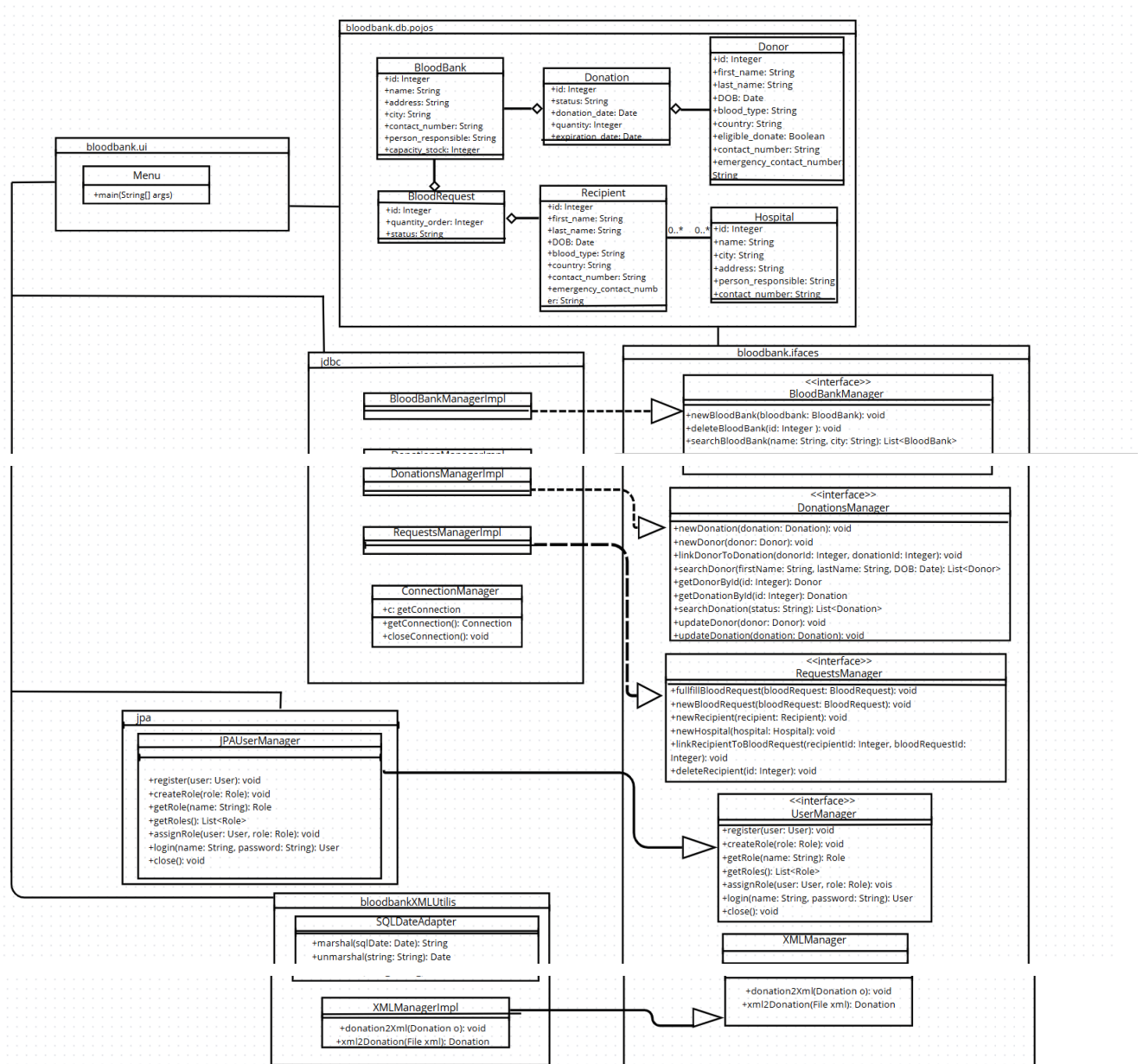
## ● BLOODBANK MANAGER:



## ● DONATIONS MANAGER:



## UML class diagram of the complete system:



All in one:

