

Lab Experiment 2

Aim: Develop DFD Model (Level 0, Level 1 DFD and data dictionary) of the sample problem.

AP22110011192

K. Navaneeth Kumar

DFD Model: (Data Flow Diagram)

- In Software engineering DFD(data flow diagram) can be drawn to represent the system of different levels of abstraction.
- A data flow diagram (DFD) shows how data moves through an information system but does not show program logic or processing steps.
- Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see mainly 3 levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.
- DFD is a graphical representation of data flow in any system. It is capable of illustrating incoming data flow, outgoing data flow and store data.
- DFD describes anything about how data flows through the system.
- The choice of DFD level depends on the complexity of the system and the level of detail required to understand the system. Higher levels of DFD provide a broad overview of the system, while lower levels provide more detail about the system's processes, data flows, and data stores.

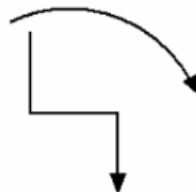
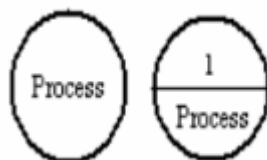
Four basics symbols of DFDs:

1. Process

A process transforms incoming data flow into outgoing data flow.

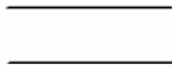
2. Dataflow

Dataflows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.



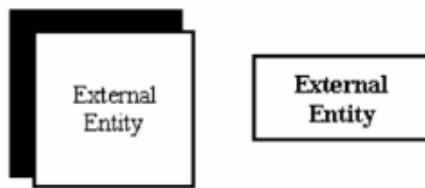
3. Data Store

Datastores are repositories of data in the system. They are sometimes also referred to as files.



4. External Entity

External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.



1. Preliminary Context Diagram

Context Diagram: This diagram shows the system as a single process with inputs and outputs connected to external entities. Identify major inputs and outputs for your system and represent them graphically.

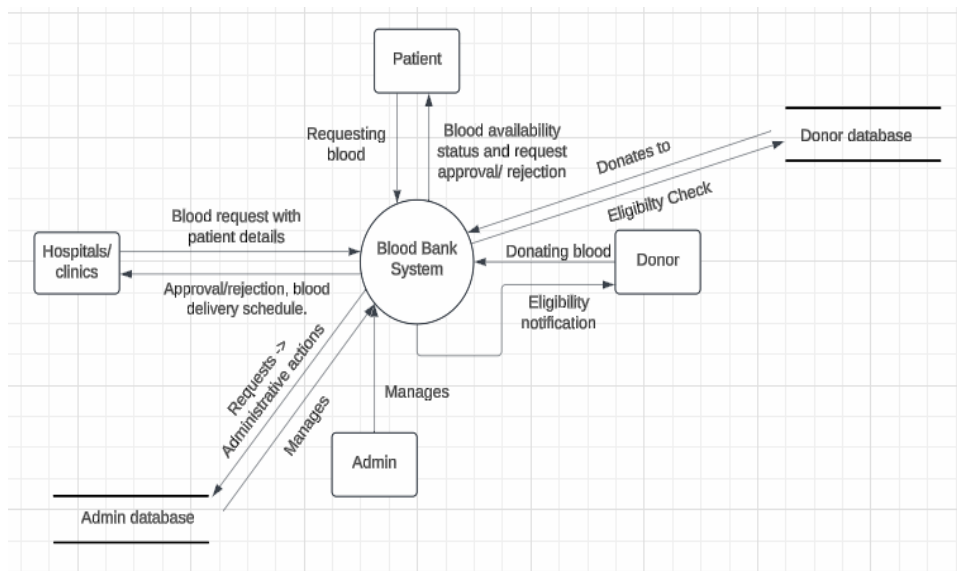
System: Blood Bank

Entities:

- **Patient:** Requests blood and receives status updates.
- **Donor:** Donates blood and receives eligibility notifications.
- **Hospital/Clinic:** Sends blood requests with patient details and receives approval/rejection along with delivery schedules.
- **Admin:** Manages the system's operations.

High-Level Data Flows:

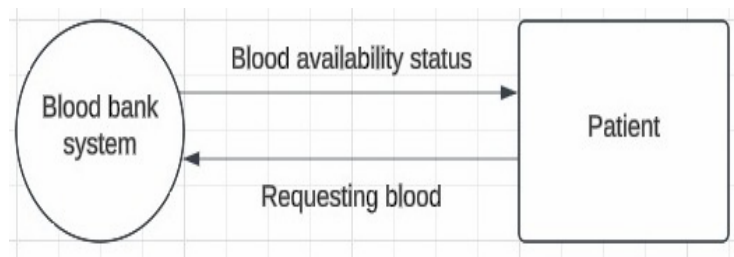
- Patient → Requests blood → Blood Bank System.
- Blood Bank System → Status update/Approval → Patient.
- Donor → Donates blood → Blood Bank System.
- Blood Bank System → Eligibility notification → Donor.
- Hospital → Sends blood request → Blood Bank System.
- Blood Bank System → Delivery schedule/Approval → Hospital.
- Admin → Manages → Blood Bank System.



2. Identify Use Cases

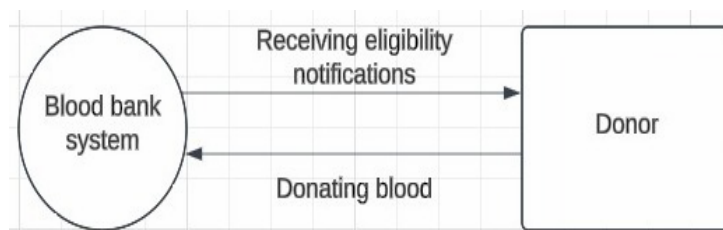
- **Patient Use Cases:**

- Requesting blood.
- Checking blood availability.



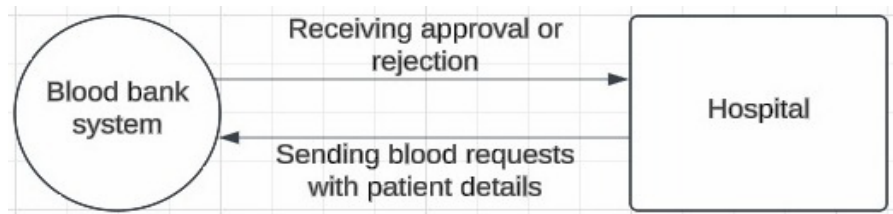
- **Donor Use Cases:**

- Donating blood.
- Receiving eligibility notifications.



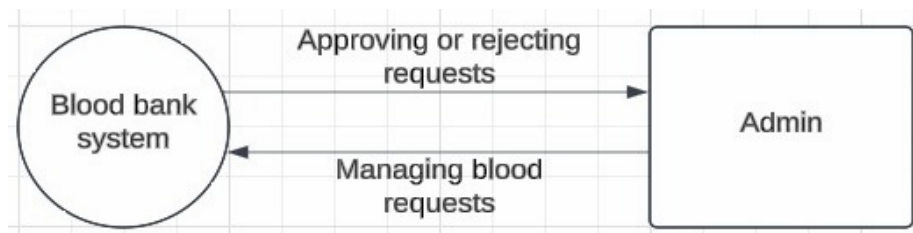
- **Hospital Use Cases:**

- Sending blood requests.
- Receiving approval or rejection.



- **Admin Use Cases:**

- Managing blood records.
- Approving or rejecting requests.



Level 0 Data Flow Diagram (DFD):

Purpose of Level 0 DFD:

- The Level 0 DFD provides an expanded view of the context diagram for the Blood Bank System.
- It represents the primary processes, external entities, and the flow of data between them and the system.
- It illustrates how key functionalities interact within the system while maintaining a top-level abstraction.

Key Components

1. External Entities:

- **Donor:** Contributes blood donations to the system and receives notifications about their eligibility.
- **Patient:** Requests blood and receives validation responses from the system.
- **Hospitals/Clinics:** Submit blood requests and receive blood delivery schedules or request validation statuses.
- **Admin:** Oversees the blood bank system's management, ensuring proper operations and monitoring activities.

2. Processes:

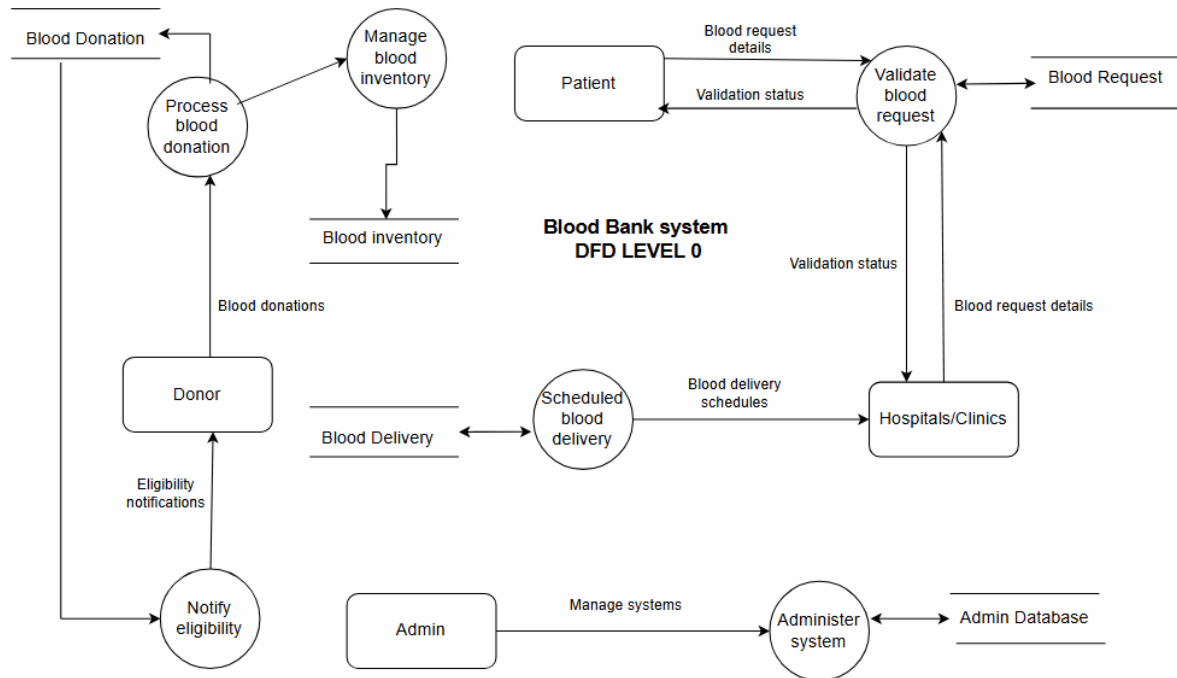
- **Validate Blood Request:** Verifies requests from patients and hospitals against blood inventory to determine whether they can be fulfilled.
- **Notify Eligibility:** Sends notifications to donors about their eligibility to donate based on specific criteria (e.g., health conditions or time since their last donation).
- **Process Blood Donation:** Manages and records the details of incoming blood donations, ensuring proper handling and storage.
- **Scheduled Blood Delivery:** Schedules deliveries of approved blood requests for patients or hospitals and ensures timely dispatch.
- **Administer System:** Allows the admin to manage operations, update system settings, and oversee processes.
- **Manage Blood Inventory:** Tracks and updates the available stock of blood types, ensuring that data is current and accessible for validation and delivery.

3. Data Stores:

- **Blood Inventory:** Contains records of available blood, including types and quantities.
- **Donor Records:** Maintains information about registered donors, including eligibility status and donation history.
- **Request Records:** Logs blood requests from patients and hospitals to ensure tracking and validation.

4. Data Flows:

- **Blood Donations:** Flow of blood from donors to the system.
- **Eligibility Notifications:** Messages sent to donors about their donation eligibility.
- **Blood Request Details:** Information submitted by patients or hospitals about their blood needs.
- **Validation Status:** Response provided to patients or hospitals indicating whether their request can be fulfilled.
- **Blood Delivery Schedules:** Details provided to hospitals and clinics about when and how blood will be delivered.
- **Manage Systems:** Data and actions performed by the admin to maintain system functionality.



Level 1 Data Flow Diagram (DFD):

Purpose of Level 1 DFD:

- The Level 1 DFD is a detailed breakdown of the **Blood Bank System**'s major process into its sub-processes.
- It focuses on illustrating the internal flows and interactions of processes within the system.

Key Components

Sub-processes:

1. Verify Donor Eligibility:

- Validates the donor's details and medical history to ensure eligibility for blood donation.
- Sub-processes include:
 - Collecting donor information.
 - Checking eligibility criteria.
 - Notifying eligibility status to the donor.

2. Process Blood Donation:

- Manages the collection, processing, and storage of blood from donors.
- Sub-processes include:
 - Verifying donor details.
 - Processing and storing the blood.

3. Manage Blood Inventory:

- Tracks blood units in the inventory, including monitoring expiry and updating stock levels.
- Sub-processes include:
 - Monitoring blood expiry.
 - Alerting low stock.
 - Tracking inventory levels.
 - Updating blood inventory.

4. Validate Blood Request:

- Checks and validates requests from patients or hospitals for blood.
- Sub-processes include:
 - Matching requests with available inventory.
 - Approving or rejecting requests.
 - Notifying patients or hospitals about request status.

5. Schedule Blood Delivery:

- Plans and coordinates the delivery of blood to hospitals or clinics.
- Sub-processes include:
 - Planning delivery routes.
 - Notifying delivery personnel.
 - Confirming delivery to hospitals.
 - Updating delivery status.

6. Administer System:

- Manages administrative tasks like user permissions, system rules, and report generation.
- Sub-processes include:
 - Managing users and permissions.
 - Updating system rules.
 - Generating reports.

Data Flows:

- **From Donor:**
 - Donor information flows into processes for verification and eligibility analysis.
 - Eligibility notifications are sent back to the donor.
- **From Patient or Hospital:**
 - Blood request details flow into the validation process.
 - Request validation status flows back to patients or hospitals.
- **To and From Inventory:**
 - Blood donation data updates the inventory.
 - Blood inventory data is used to validate requests and track stock levels.
- **Delivery Data:**
 - Delivery schedules flow to delivery personnel and hospitals.
 - Confirmation of delivery updates system records.

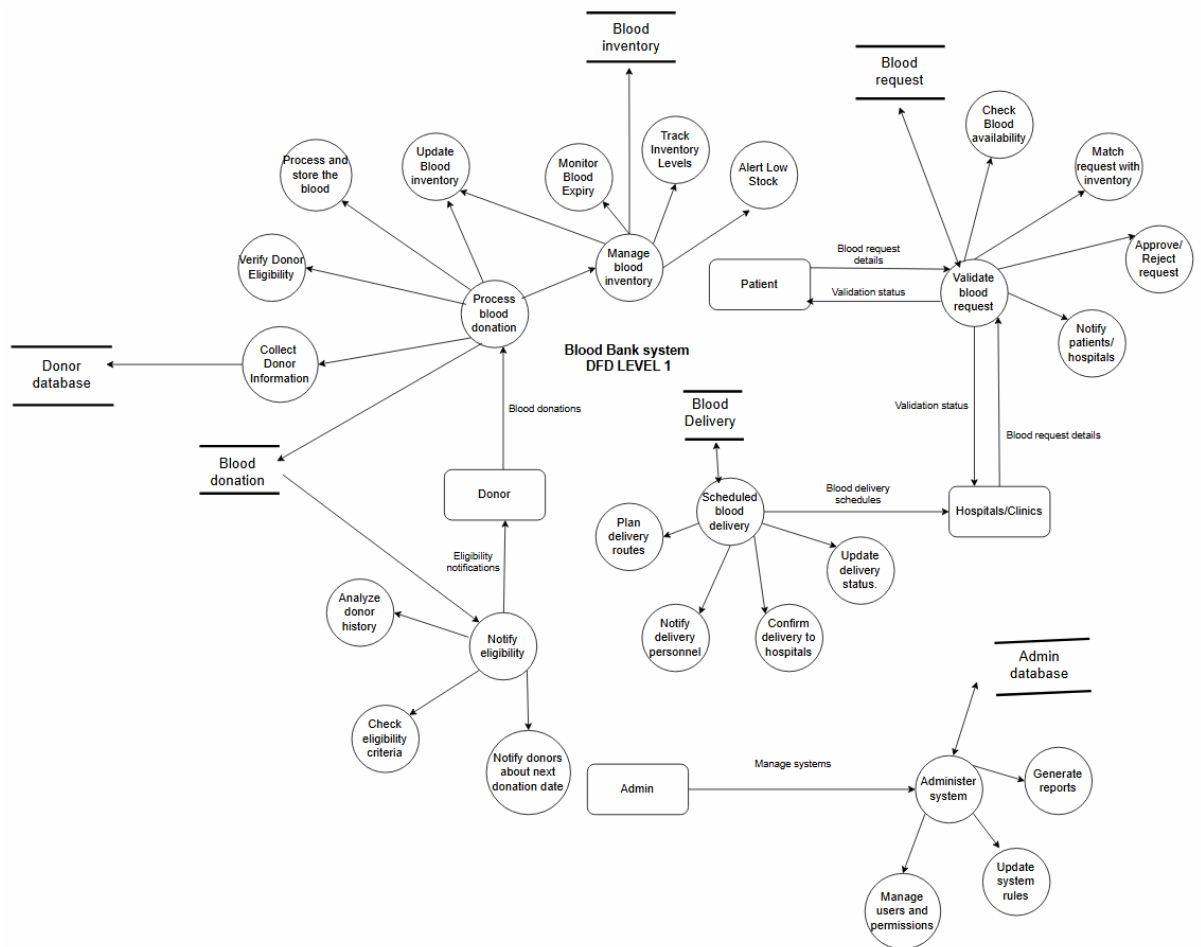
External Entities:

1. **Donor:**
 - Provides information and blood for donation.
 - Receives notifications about eligibility and donation status.
2. **Patient:**
 - Submits blood requests.

- Receives request validation status.
- 3. **Hospitals/Clinics:**
 - Submit bulk blood requests.
 - Receive deliveries of blood and delivery confirmations.
- 4. **Admin:**
 - Manages the system's users, permissions, and settings.

Data Stores

1. **Donor Database:**
 - Stores information about donors and their donation history.
2. **Blood Inventory Database:**
 - Tracks blood units, their types, and expiry status.
3. **Request Database:**
 - Contains details of patient or hospital requests.
4. **Delivery Database:**
 - Logs scheduled and completed deliveries.
5. **Admin Data Store:**
 - Maintains records of users, permissions, and system rules.



Data Dictionary for Blood Bank Management System

- The data dictionary is a reference work of metadata (data about data) that organizes and defines the structure, meaning, and relationships of data used in the Blood Bank Management System.
- As a document, the data dictionary provides clarity and consistency across the system, ensuring that all stakeholders, including donors, patients, hospitals, and administrators, have a shared understanding of the terms and processes.
- The data dictionary enables system analysts, developers, and users to navigate through the design and functionality of the system effectively.

Data Dictionary Notation

Notation	Meaning
=	is composed of
+	and

[]	either-or
{ }	n repetitions of
(...)	optional data
* ... text ... *	delimits a comment

Key Components

Donor

- Represents individuals who provide blood donations.
- **Attributes:**
 - Donor_ID = Unique identifier for a donor.
 - Name = Donor's full name.
 - Age = Donor's age.
 - Blood_Type = [A+, A-, B+, B-, AB+, AB-, O+, O-].
 - Contact_Details = {Phone_Number, Email_Address}.
 - Address = Residential address of the donor.
 - Eligibility_Status = *Indicates if the donor is currently eligible.*
 - Donation_History = {n records of previous donations}.

Blood Inventory

- Tracks the stock and management of blood units in the system.
- **Attributes:**
 - Inventory_ID = Unique identifier for each blood unit entry.
 - Blood_Type = [A+, A-, B+, B-, AB+, AB-, O+, O-].
 - Quantity = Amount of available blood units.
 - Expiry_Date = *Date after which blood is no longer usable.*
 - Status = [In_Stock, Low_Stock, Expired].

Blood Donation

- Records the process of blood donation and storage.
- **Attributes:**

- Donation_ID = Unique identifier for a donation event.
- Donor_ID = Links donation to a specific donor.
- Donation_Date = *Date of the donation event.*
- Blood_Type = Type of blood donated.
- Volume = Amount of blood donated (in mL).
- Processing_Status = [Processed, Stored, Disposed].

Blood Request

- Manages blood requests from hospitals or patients.
- **Attributes:**
 - Request_ID = Unique identifier for a blood request.
 - Requester_Name = [Hospital_Name | Patient_Name].
 - Blood_Type = Type of blood requested.
 - Quantity = Number of units required.
 - Request_Date = *Date of the request submission.*
 - Approval_Status = [Pending, Approved, Rejected].
 - Fulfillment_Status = [In-Progress, Delivered].

Blood Delivery

- Tracks the delivery of blood to requesting entities.
- **Attributes:**
 - Delivery_ID = Unique identifier for a delivery entry.
 - Request_ID = Links delivery to a specific blood request.
 - Delivery_Date = *Date when blood is delivered to the requester.*
 - Delivery_Status = [Scheduled, Delivered, Failed].
 - Route_Details = Information about the delivery route.

Admin

- System administrator who manages the system and generates reports.
- **Attributes:**
 - Admin_ID = Unique identifier for an admin user.

- Username = Login username for the admin.
- Role = [Super_Admin, Manager].
- Permissions = {Access_Rights for system functions}.