**Requirement Analysis**

**Data Flow diagram**

|  |  |
| --- | --- |
| Date | 27-06-2025 |
| Team ID | LTVIP2025TMID46688 |
| Project Name | DocSpot |
| Maximum Marks | 4 Marks |

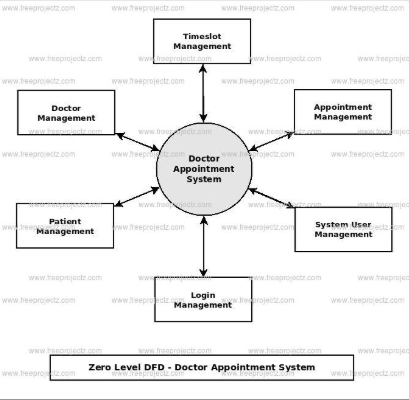
**Data Flow Diagram – DocSpot**

The **Data Flow Diagram (DFD)** is a crucial part of system analysis and design. It visually represents how data moves through the DocSpot system — from input to processing and output — across different user roles such as patients, doctors, and administrators.

The DFD provides a clear picture of the **flow of information**, the **interactions between system components**, and the **boundaries of the system**. It helps both developers and stakeholders understand the system's architecture at various abstraction levels.

**Level 0 DFD – Context Diagram**

The **Level 0 DFD** shows the overall system as a single process and how external entities interact with it.



**External Entities:**

* **User**: Registers, logs in, books appointments
* **Doctor**: Applies for approval, manages appointments
* **Admin**: Reviews doctor applications, approves/rejects them

**Central Process:**

* **DocSpot System**: The core application handling all requests and data storage

**Data Stores:**

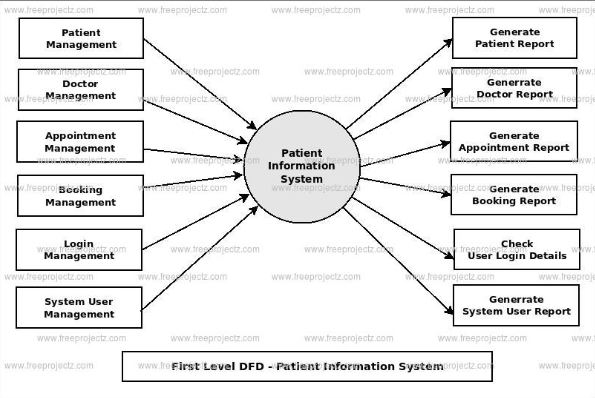
* **User Database**
* **Doctor Database**
* **Appointment Database**

**Flow Overview:**

* Users send data (registration, login, booking) → System processes → Saves to database → Returns confirmation
* Doctors apply → System routes to Admin → Admin updates status → System notifies doctor
* All activities are logged in the database

**Level 1 DFD – Detailed Process Flow**

The **Level 1 DFD** expands the system into **individual processes and data flows**.



**Processes:**

1. **User Authentication**: Validates login/register details
2. **Appointment Booking**: Matches available doctors and schedules
3. **Doctor Application**: Submits data to admin for review
4. **Admin Review**: Approves/rejects doctor applications
5. **Notification System**: Sends status updates to users and doctors

**Data Flows:**

* Input from forms (login, apply, book) → Backend APIs → Database
* Admin actions → Update doctor status → Notify users
* Appointment confirmations → Shown in dashboards

**Key Components in DFD:**

| **Component** | **Description** |
| --- | --- |
| **User** | Registers, books appointments, views history |
| **Doctor** | Applies for approval, views appointments |
| **Admin** | Manages doctor requests and monitors system activity |
| **Frontend (React)** | Sends requests to backend via Axios |
| **Backend (Node.js/Express)** | Processes data, performs validations, routes logic |
| **Database (MongoDB)** | Stores all records in collections (users, doctors, bookings) |

**Purpose and Outcome**

* Ensures **clear understanding** of data flow before development
* Helps identify **system boundaries**, **data dependencies**, and **interactions**
* Assists in efficient database structuring and API planning
* Acts as a reference for debugging and future feature expansion