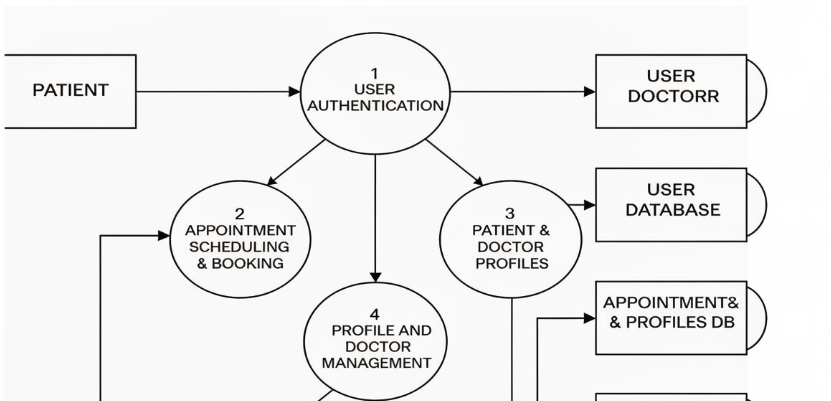
**Project Design Phase-II**

**Data Flow Diagram & User Stories**

|  |  |
| --- | --- |
| Date | 26-11-2025 |
| Team ID |  |
| Project Name | Docspot |
| Maximum Marks | 4 Marks |

DocSpot is a full-stack web application that allows patients to register, browse doctors, book appointments, upload medical documents, and manage their health interactions online, while doctors and admins manage availability, approvals, and system monitoring. The system uses a MERN-style stack with MongoDB for data persistence, Express/Node.js for APIs, React with Bootstrap for the UI, and JWT-based security for authentication and role-based access control.

**Level 1 Data Flow Diagram:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| Patient | Appointment Booking | As a patient, I can book an appointment with a doctor for a specific date and time. | Appointment appears in my dashboard and in the doctor’s upcoming appointments list. | High | Sprint-1 |
| Patient | Document Upload | As a patient, I can upload a medical report for a booked appointment. | File is validated and visible in the appointment details for me and my doctor. | Medium | Sprint-2 |
| Doctor | Schedule Management | As a doctor, I can set my available slots so patients can book based on my schedule. | Patients can only book within defined availability and see real-time slot status. | High | Sprint-1 |
| Admin | Doctor Verification | As an admin, I can approve or reject doctor accounts after reviewing their documents. | Only approved doctors appear in patient search and can accept appointments. | High | Sprint-1 |