Project Design Phase
Solution Architecture

Date	15 February 2025
Team ID	LTVIP2026TMIDS88461
Project Name	DocSpot — Seamless Appointment Booking for Health
Maximum Marks	4 Marks

Solution Architecture:

Overview

Solution architecture is a structured process that bridges the gap between business problems and technology implementation. It translates identified customer pain points and business requirements into a scalable, secure, and efficient technical solution.

For the DocSpot – Seamless Appointment Booking System, the solution architecture defines how the platform components interact to deliver a reliable healthcare appointment management system.

Goals of the Solution Architecture

The primary objectives of the solution architecture are to:

● Identify the most appropriate technology stack to solve healthcare appointment management challenges efficiently.

● Define the overall system structure, including frontend, backend, database, and third-party integrations.

● Describe system characteristics, such as scalability, performance, reliability, and security.

● Translate business requirements into technical components and development modules.

● Define system behavior, including data flow, authentication, booking logic, and notification workflows.

● Establish development phases and implementation roadmap aligned with project milestones.

● Ensure compliance and data protection standards suitable for healthcare-related information.

● Provide clear technical specifications that guide development, testing, deployment, and maintenance.

Key Architectural Objectives for DocSpot

● Build a secure role-based system for Patients, Doctors, Admins, and Customer Care Executives.
● Enable real-time appointment booking with slot validation to prevent double bookings.

● Implement scalable backend services capable of handling 1000+ concurrent users.
● Integrate email and SMS notification services for appointment confirmations and reminders.
● Maintain data integrity and audit logging for compliance and traceability.
● Support future enhancements such as payment integration and analytics expansion.

Architecture Principles Followed

● Modular Design – Separation of concerns (Frontend → API → Services → Database).
● Service Layer Pattern – Controllers delegate logic to services for maintainability.
● RESTful API Standards – Clean, scalable endpoint design.
● Security First Approach – JWT authentication, bcrypt hashing, role-based authorization.
● Scalability & Performance Optimization – Redis caching, asynchronous job queues.
● Cloud-Ready Deployment – Designed for horizontal scaling and containerization.

High-Level Architecture Components

1. Frontend Layer
    - React-based Web Application
    - Role-based UI rendering
    - Responsive design for mobile and desktop
2. Backend Layer
    - Node.js with Express.js
    - RESTful APIs
    - Authentication & Authorization (JWT)
    - Business Logic Services
3. Database Layer
    - MongoDB with Mongoose
    - Indexed collections for performance
    - Audit logs and activity tracking
4. Queue & Notification Layer
    - Redis + Bull for background job processing
    - Email service integration (SMTP)
    - SMS service integration
5. Security Layer
    - HTTPS encryption
    - Input validation & sanitization & Role-Based Access Control (RBAC)

Solution Architecture Diagram: