Hospital Management System (HMS) Project Task Assignment

From: Project Manager
To: Developer Team
Date: December 13, 2025
Project: Hospital Management System using Spring Boot
Deadline: 3-6 months (iterative delivery preferred)

Dear Team,

You are assigned to develop a **Hospital Management System (HMS)** using **Spring Boot** as the primary backend framework. The goal is to create a secure, scalable, and efficient web application that automates key hospital operations. We will follow an agile approach with sprints (2-4 weeks each) for incremental delivery.

Project Overview

- **Technology Stack:** Spring Boot 3.x, Java 17+, Spring Data JPA, Spring Security (JWT), MySQL/PostgreSQL, Lombok, Maven/Gradle.
- **Architecture:** Layered (Controllers, Services, Repositories, DTOs); start monolithic, prepare for microservices if needed.
- **Core Features:** Role-based access (Admin, Doctor, Receptionist/Patient optional).
- **Non-Functional:** Secure authentication, data validation, error handling, API documentation (Swagger), basic reporting.

Phased Tasks (Suggested Sprints)

Sprint 1: Setup and Authentication (2 weeks)

- Initialize project using Spring Initializr (add dependencies: Web, JPA, Security, Lombok, MySQL Driver, DevTools).
- Configure database connection in `application.yml`.
- Implement user registration and JWT-based login/logout.
- Set up role-based security (Admin, Doctor, Receptionist).
- Add global exception handling and basic logging.

Sprint 2: Patient and Doctor Management (2-3 weeks)

- Design entities: Patient (id, name, dob, contact, address), Doctor (id, name, specialization, contact), User (for roles).
- Create JPA repositories and services for CRUD operations.
- Build REST APIs:

  - `/api/patients` (GET, POST, PUT, DELETE)
  - `/api/doctors` (GET, POST, PUT, DELETE – Admin only)
- Implement validation (@Valid) and DTOs for requests/responses.

Sprint 3: Appointment Scheduling (2 weeks)

- Add Appointment entity (id, patientId, doctorId, dateTime, status: booked/completed/cancelled).
- APIs for booking, viewing, updating, and cancelling appointments.
- Add logic for checking doctor availability (no overlapping appointments).
- Role restrictions: Patients/Receptionists can book; Doctors view their schedule.

Sprint 4: Billing and Basic Reporting (2 weeks)

- Add Billing entity linked to appointments/patients.
- Simple billing calculation (consultation fee + medicines).
- Generate basic reports (e.g., daily appointments, patient list) – export to PDF if possible (use JasperReports or iText).
- Admin dashboard APIs for summaries.

Sprint 5: Enhancements and Testing (2-3 weeks)

- Add pharmacy/inventory basics if time allows (Medicine entity, stock tracking).
- Implement pagination and search for lists (Spring Data Pageable).
- API documentation with Springdoc OpenAPI/Swagger.
- Write unit/integration tests (JUnit, Mockito, @SpringBootTest).
- Security testing (role enforcement).

Final Sprint: Deployment and Polish

- Dockerize the application.
- Prepare for deployment (JAR build, basic Actuator for monitoring).
- Fix bugs from UAT feedback.
- Document the project (README with setup instructions, API endpoints).

Deliverables

- Fully functional backend with REST APIs.
- Source code on GitHub (with branches for features).
- Postman collection for API testing.
- Basic frontend integration ready (if we add React/Angular later).
- Weekly progress updates and demo at sprint end.

Best Practices to Follow

- Clean code: Use Lombok, proper naming, comments.
- Git: Commit often, use feature branches.
- Testing: Aim for 70%+ coverage on core modules.

- Security: Encrypt sensitive data, use HTTPS in production.

If you encounter blockers, raise them immediately. Let's aim for an MVP by end of Sprint 3.

Good luck!

Project Manager

Note: To convert this to a PDF, simply open this HTML file in your browser and use "Print → Save as PDF" or use an online HTML-to-PDF converter.