Hospital Management System

Content Overview

Hospital management system

Figma file Link (scan this link)

https://www.figma.com/design/AnjMOisy7MhxYCFhrRwTem/Hospital-management-system--Community-?node-id=1518-1935&p=f&t=j00eXWCuYTKQGJil-0

Note:-

If you find this project useful, you can use it in your own projects and add any additional features you need. If I’ve missed something, feel free to extend or modify it as required

📌 1. Problem Statement

Hospitals require an integrated digital system to manage patients, doctors, appointments, billing, and records. The system should:

- Reduce manual paperwork
- Improve patient/doctor efficiency
- Ensure security of medical records
- Handle large-scale concurrent operations

📌 2. Workflows (Core Modules)

🔹 Patient Workflow

1. Patient registers → Authentication with JWT
2. Books appointment with doctor
3. Receives notifications (SMS/Email)
4. Pays bill online/offline
5. Accesses medical history & prescriptions

🔹 Doctor Workflow

1. Doctor login → Dashboard
2. View assigned appointments
3. Update patient diagnosis, prescriptions
4. Track patient medical history

🔹 Admin Workflow

1. Manage doctors, departments, schedules
2. Manage patients, room allocation
3. Generate bills & insurance claims
4. Audit logs, reporting

📌 3. Technical Details & Best Practices

📱 **Frontend**

•**Angular** + **Bootstrap** → Responsive UI for patients, doctors, admins

⚙️ **Backend**

•**Spring Boot** (Microservices):

- Patient Service
- Doctor Service
- Appointment Service
- Billing Service
- Notification Service
- Audit/Logs Service

🗄 **Databases**

•**MySQL** → Structured (patients, doctors, appointments, billing)

•**MongoDB** → Unstructured (medical records, prescriptions, logs)

🔀 **Messaging**

•**Kafka** → Async notifications, appointment updates, billing events

🔒 **Security**

•**JWT / OAuth2** for authentication & role-based access (Doctor, Patient, Admin)

🛠 **Best Practices**

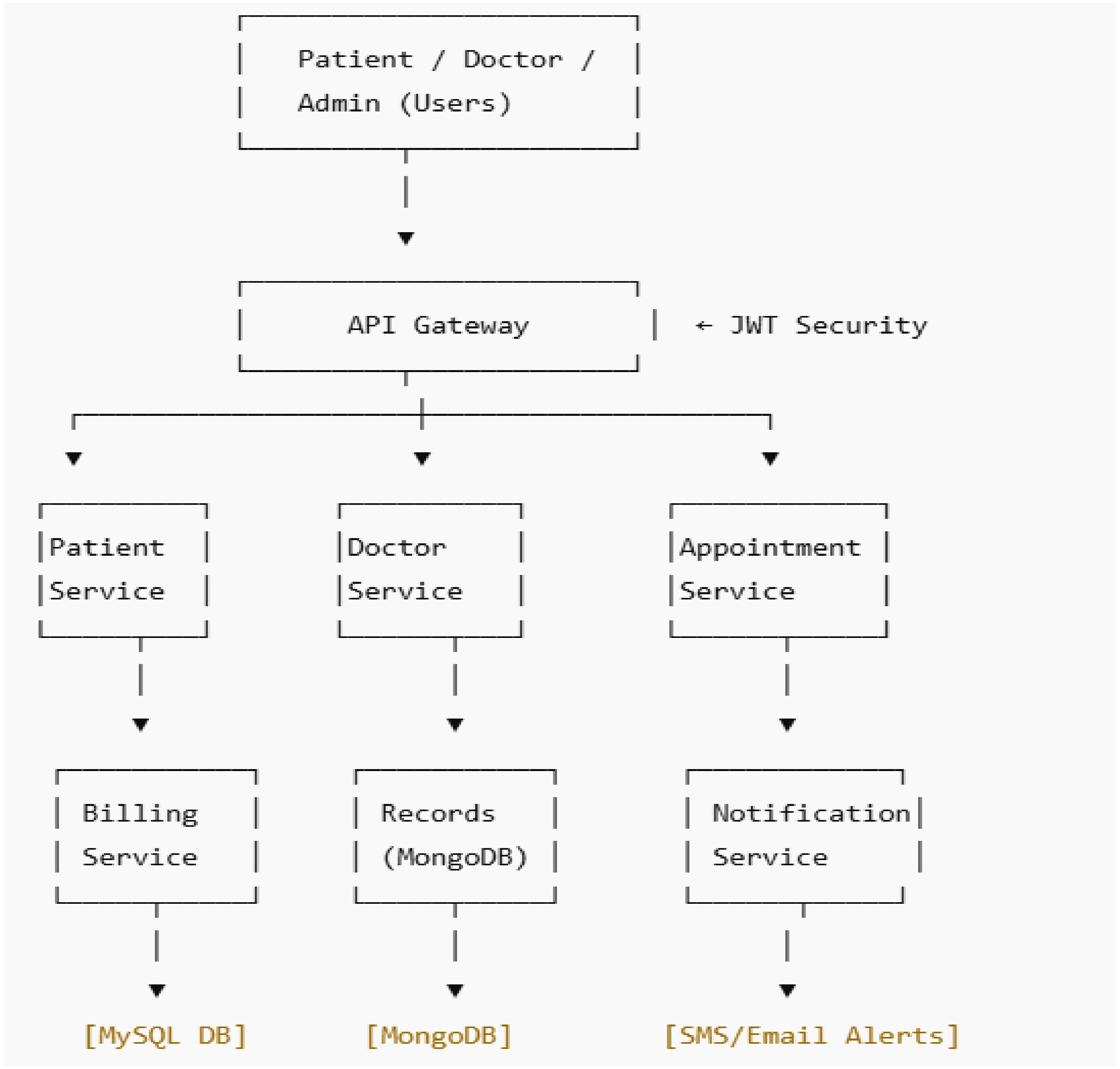
•**Pagination** for patient records, billing history

•**AOP Logging** for sensitive actions (medical updates, payments)

•**Event-driven design** with Kafka

•**Auditing** (HIPAA/GDPR readiness)

📌 4. Overall Flow Diagram



📌 5. Outcome of This Project

By the end, you will have a **production-ready HMS** with:

- Secure login for doctors, patients, admins
- Online appointment booking
- Billing & payment system (with pagination & history)
- Medical records stored securely
- Notifications for patients & doctors
- Admin dashboards with reports & audit logs

📌 6. Tech Stack with Icons

- **Spring Boot** (backend microservices)
- **Angular** (frontend)
- **MySQL** (transactional DB)
- **MongoDB** (medical records/logs)
- **Apache Kafka** (event-driven communication)
- **Bootstrap** (responsive UI)
- **JWT** (authentication & authorization)

Documentation Using API Doc's and Swagger UI

(refer this for swagger UI & Api doc's

---->https://github.com/SuryaBhaskarG/rest-semi

----->https://github.com/thevipulvats/journalApp