Telemedicine Platform – Project Report

College: Lovely Professional University

Course: B.Tech – CSE

Semester: 3

1. Introduction

The Telemedicine Platform allows patients and doctors to connect remotely using secure video consultations, chat messaging, and e-prescriptions.
The goal is to make healthcare accessible anytime, anywhere, especially for rural areas and emergency cases.
This system integrates appointment scheduling, patient records, and real-time teleconsultation services in one platform.

2. Stakeholders

• Patients – Book appointments, video consult, receive e-prescriptions.

• Doctors – Provide consultations, issue prescriptions, view patient history.

• Admin – Manage users, track system health, handle escalations.

• Pharmacy – Process digital prescriptions.

• Technical Support – Ensure uptime, resolve technical issues.

3. User Stories

• As a patient, I want to create an account to access telemedicine services.

• As a patient, I want to schedule appointments easily.

• As a doctor, I want to access patient history before consultations.

• As a patient, I want to receive prescriptions digitally.

• As an admin, I want dashboards to monitor system usage.

4. Context Diagram

A context diagram shows the high-level interactions of the telemedicine system with external entities.

The system communicates with:

Patients → booking, consultation, records

Doctors → schedules, consultations, prescriptions

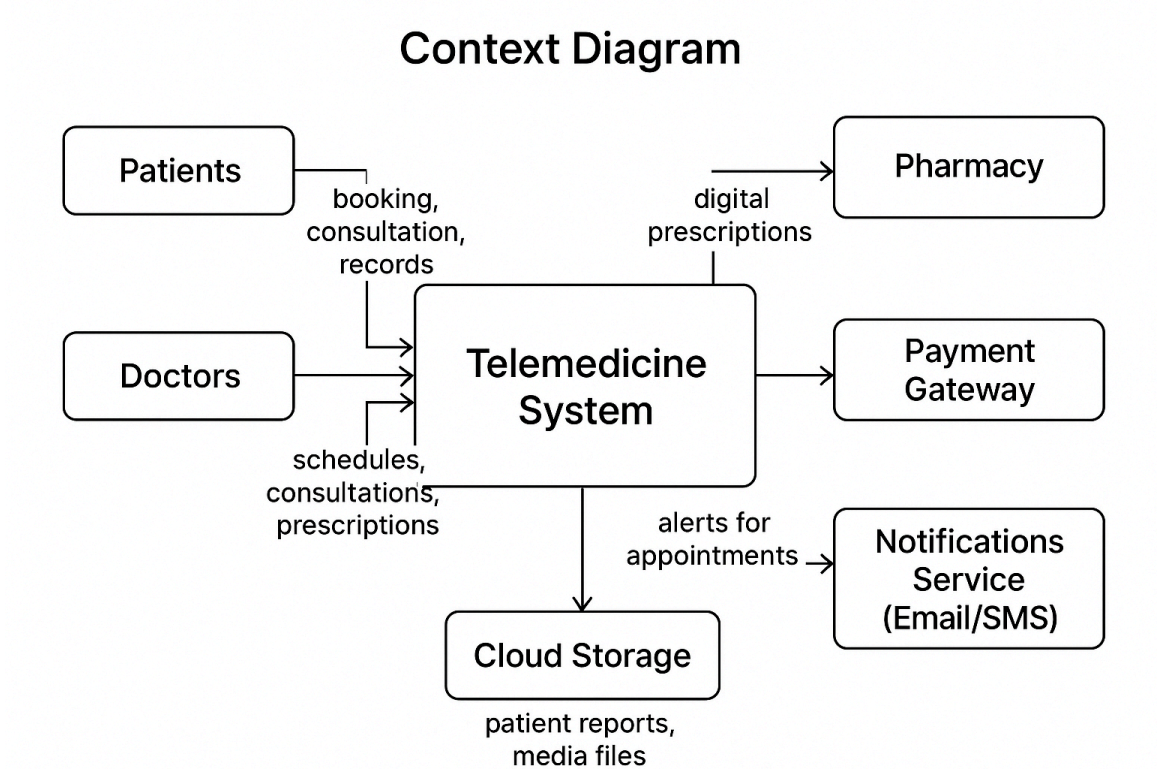
Pharmacy → digital prescriptions

Payment Gateway → online payments for consultation

Notifications Service (Email/SMS) → alerts for appointments

Cloud Storage → patient reports, media files

This diagram clearly defines what is inside the system and what interacts with it from outside.



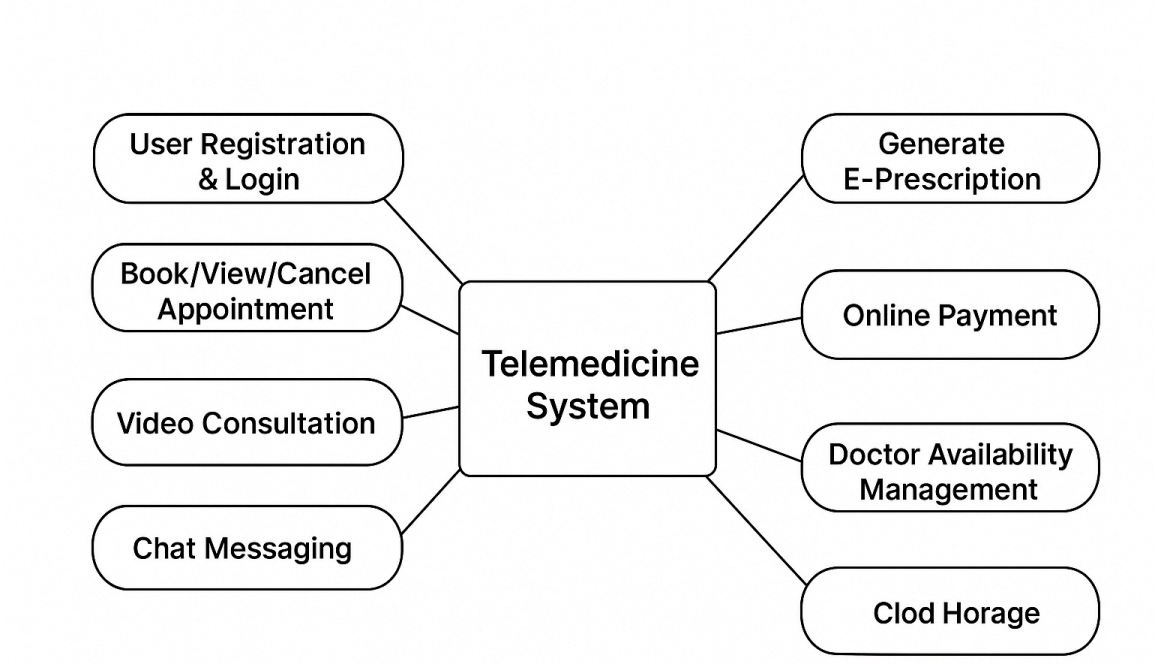
5. Use Case Diagram

Major use cases included in the Telemedicine System:

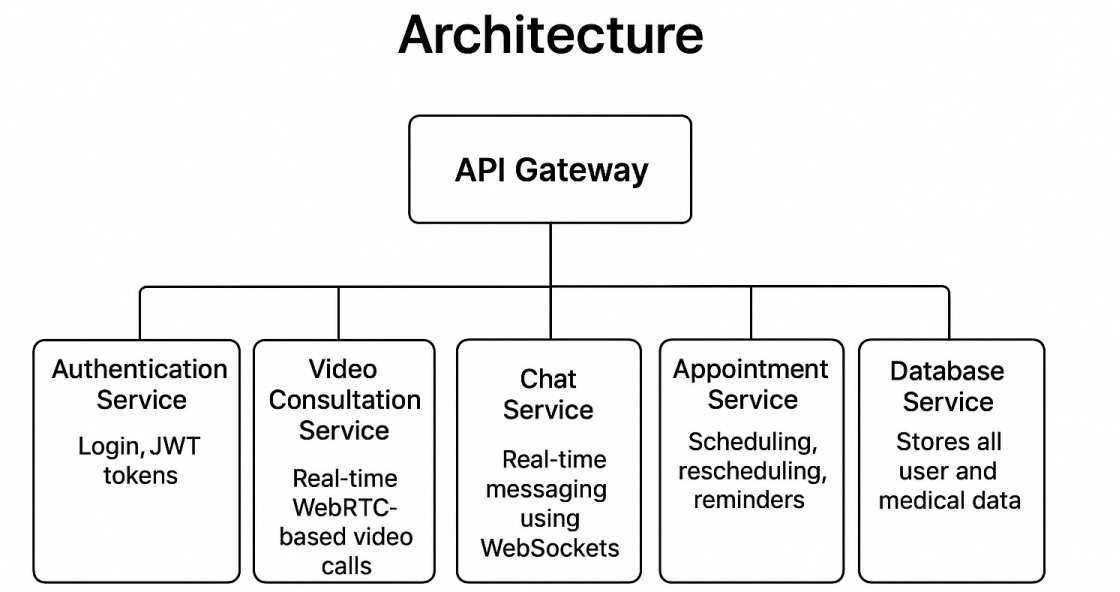
- User Registration & Login
- Book/View/Cancel Appointment

- Video Consultation
- Chat Messaging
- Upload Medical Reports
- Generate E-Prescription
- Online Payment
- Doctor Availability Management

These use cases provide a complete functional overview of the platform.



6. Architecture



The platform uses Microservices Architecture for better scalability and modularity.

Core Components

1. **API Gateway** – Single entry point for all clients
2. **Authentication Service** – Handles login, JWT tokens
3. **Video Consultation Service** – Real-time WebRTC-based video calls
4. **Chat Service** – Real-time messaging using WebSockets
5. **Appointment Service** – Scheduling, rescheduling, reminders
6. **Prescription Service** – Secure e-prescription generation

7. **Database Service** – Stores all user and medical data

Why Microservices?

- Services can scale individually (video service may need more resources)
- Easier maintenance
- Independent deployment

7. Trade-Off Analysis

Monolith vs Microservices vs Serverless:
• Monolith – Simple to develop but not scalable.

• Microservices – Highly scalable, independent deployments.

• Serverless – Low cost but limited control over long-running processes.

Decision: Microservices for flexibility and scalability in healthcare workloads.

8. Design Patterns

Builder Pattern

Used for generating structured e-prescriptions containing medicines, instructions, dosage, doctor details.

Strategy Pattern

Used for appointment scheduling where multiple rules (doctor availability, time slots, urgency) may apply.

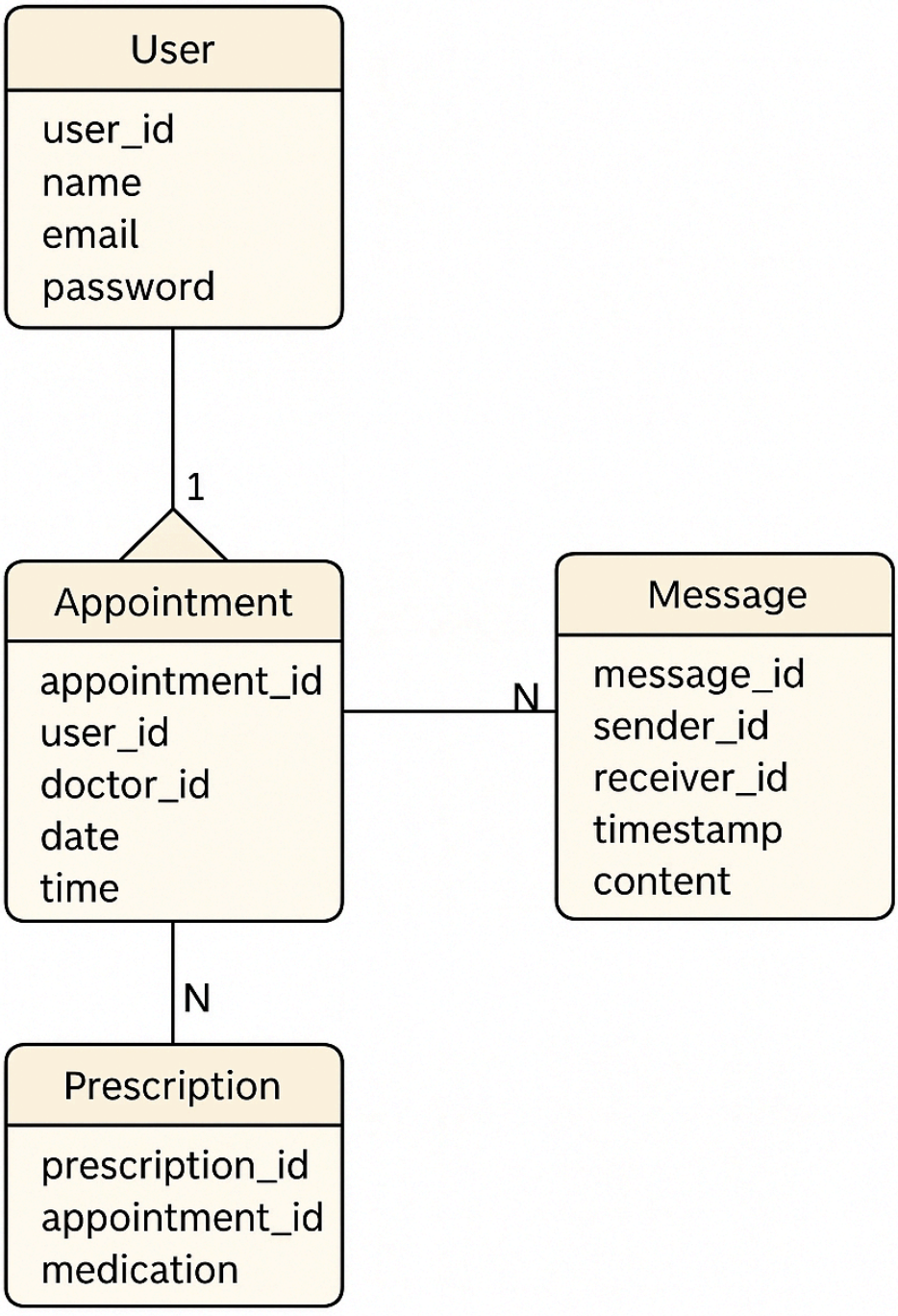
Observer Pattern

Used for sending notifications when:

- Appointment booked
- Appointment reminder
- Prescription generated

These patterns help keep code modular and maintainable.

9. Database Design + ERD


User
user_id
name
email
password
1
Appointment
appointment_id
user_id
doctor_id
date
time
N
Message
message_id
sender_id
receiver_id
timestamp
content
N
Prescription
prescription_id
appointment_id
medication

The ERD (Entity Relationship Diagram) includes these key entities:

Patient

PatientID, Name, Age, Gender, History, Reports

Doctor

DoctorID, Name, Specialization, Experience, Schedule

Appointment

AppointmentID, PatientID, DoctorID, Date, Time, Status

Prescription

PrescriptionID, AppointmentID, MedicineList, Dosage, Notes

ChatMessage

MessageID, SenderID, ReceiverID, Timestamp, MessageText

The relationships ensure smooth functioning of appointments, consultations, and prescriptions.

10. Performance

To maintain smooth performance even with thousands of users:

Caching

- Redis cache to store frequently accessed doctor profiles and schedules

CDN (Content Delivery Network)

- Serves medical images, reports faster worldwide

Load Balancer (Nginx)

- Distributes traffic across multiple servers to prevent overload

Database Replication

- Improves read performance and reliability

11. Security

Security is extremely important because medical data is sensitive.

Key Security Measures

- JWT Authentication
- Role-Based Access Control (RBAC)
- Encryption (HTTPS, AES-256)
- Secure APIs with rate limiting
- Protection against OWASP Top 10 threats like XSS, SQL Injection
- Logs & monitoring

12. API Specification

Examples:

POST /auth/login – User login

GET /appointments – Fetch upcoming appointments

POST /appointments/create – Schedule appointment

POST /video/start – Start call session

POST /prescription/create – Issue e-prescription

Errors: 400 (Bad Request), 401 (Unauthorized), 500 (Server Error).

13. Observability

To maintain system reliability:

Logs

Track errors, user actions, and system events

Metrics

CPU usage, number of active video calls, response time

Distributed Tracing

Helps debug microservices interactions

SLO (Service Level Objective)

Goal: **99.9% uptime**

14. Tech Stack Justification

Frontend: HTML, CSS, JavaScript

Backend: Python Flask

Database: SQL (MySQL / PostgreSQL)

Why this stack?

• HTML/CSS/JS provide a clean, responsive user interface.

• Flask is lightweight, fast, and perfect for building REST APIs.

• SQL databases ensure structured data management and ACID compliance.

• Easy deployment and great community support make this stack highly practical for real-world telemedicine applications.

15. Conclusion

The Telemedicine Platform is a modern solution that transforms healthcare by making it **digital, accessible, and efficient**. With features like online consultations, scheduling, digital records, and secure communication, the system reduces patient effort, saves time for doctors, and improves overall healthcare quality.

The use of microservices, strong security practices, and scalable architecture makes it suitable for real-world deployment and future growth.