Community Health Checkup and Telemedicine Portal

CSE 341: Advanced Programming

S U B M I T T E D B Y

Name	ID	Intake
Pranta Kumer Pandit	**20234103390**	**52**
Mehedi Hasan Alif	**20234103384**	**52**
Juwel Rana	**20234103363**	**52**

S U P E R V I S E D B Y :

Mst. Maharin Afroj
(lecturer)

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)

12th, January, 2018

Abstract

The *Community Health Check and Telemedicine System* is a technology-driven healthcare platform designed to provide accessible, affordable, and timely medical services to individuals within a community. The primary objective of the system is to bridge the gap between patients and healthcare professionals, especially in areas where traditional medical facilities are limited. Through this platform, users can perform basic health screenings—such as blood pressure, glucose level, BMI, and general vitals—while receiving remote medical consultations via secure video or audio communication.

The system integrates Electronic Health Records (EHR), enabling patients to store, track, and manage their medical history in a centralized and secure environment. It also offers an efficient appointment scheduling module, real-time notification alerts, and digital prescriptions issued directly by registered doctors. For healthcare providers, the platform ensures easy access to patient histories, diagnostic data, and consultation management tools, making virtual care more effective and accurate.

This project follows a structured methodology consisting of requirement analysis, system design, development, and testing to ensure accuracy, usability, and system reliability. By leveraging modern technologies such as web interfaces, online communication tools, and a database-driven backend, the system offers a scalable and user-friendly model suitable for both rural and urban communities.

Overall, the Community Health Check and Telemedicine System aims to reduce travel time, healthcare costs, and the burden on physical facilities while improving health awareness, early diagnosis, and continuity of care. This platform demonstrates how digital health solutions can significantly impact community wellbeing by making healthcare more accessible, organized, and patient-centered.

The system also emphasizes community-level health improvement by enabling continuous monitoring and data-driven assessment. Aggregated health data collected from users can help identify emerging health trends, common risk factors, and seasonal disease patterns within a community. This information can support policymakers, NGOs, and local healthcare authorities in planning targeted awareness campaigns, vaccination drives, and preventive healthcare programs. By creating a digital ecosystem that connects patients, doctors, and administrators, the project strengthens the overall healthcare infrastructure and promotes a collaborative approach to public health management.

In terms of long-term sustainability, the platform is designed to be scalable and adaptable, allowing new features and modules to be integrated as healthcare needs evolve. The system can be expanded to support specialist consultations, AI-based health predictions, multilingual interfaces, and integration with wearable health devices. Its lightweight architecture and cloud-ready design ensure that it can support large user bases without compromising performance.
Ultimately, this project not only addresses immediate healthcare challenges but also lays the groundwork for future digital health innovations that can empower communities and enhance the quality of life on a broader scale.

List of Figures

Table of Contents

Chapter 1
INTRODUCTION

1.1. Problem Specification

In many communities, especially in rural and semi-urban areas, people struggle to access timely and reliable healthcare services due to doctor shortages, long travel distances, and overcrowded hospitals. A memorable incident that inspired this project occurred when several elderly patients in a local community had to wait hours for basic check-ups, and some even skipped treatment because they could not afford transportation to the nearest clinic. This highlighted a critical gap in accessible healthcare and motivated the need for a system that brings medical support directly to people's homes.

1.2. Objectives

The main objectives of the *Community Health Check and Telemedicine System* are to ensure accessible, efficient, and technology-driven healthcare services for community members. The project aims to achieve the following goals:

1. **To provide an easy-to-use digital platform** where community members can perform basic health checks and record their vital health data.
2. **To enable remote consultations** through video or audio communication so that patients can receive timely medical advice without traveling long distances.
3. **To maintain a secure Electronic Health Record (EHR)** for storing and managing patients' medical histories, prescriptions, and diagnostic reports.
4. **To simplify appointment scheduling** by offering an automated system that connects patients with available doctors efficiently.
5. **To reduce healthcare burden** on physical clinics by offering digital alternatives for routine check-ups and follow-ups.
6. **To increase health awareness** within the community by providing continuous monitoring and easy access to medical guidance.
7. **To build a scalable and user-friendly system** that can be expanded with more modules and features in the future.

1.3. Scope

The scope of the *Community Health Check and Telemedicine System* defines the boundaries and overall direction of the project, outlining what the system will deliver and how it will function. This project focuses on developing a digital healthcare platform that connects patients, doctors, and administrators within a unified environment. The system includes features such as health monitoring, online consultations, appointment management, and electronic health record storage.

1.4. Organization Of Project Report

This project report is organized into several chapters, each presenting a specific part of the system development process in a clear and structured manner. **Chapter 1** introduces the background of the project, outlines the core problems, states the objectives, highlights the

project scope, and explains how the entire report is structured. This chapter provides the foundation for understanding why the system is needed and what it aims to achieve.

Chapter 2 focuses on the theoretical background and related works. It explains existing telemedicine systems, community health monitoring approaches, and the technologies relevant to this project. This chapter helps to justify the innovation by comparing current solutions and identifying gaps.

Chapter 2
BACKGROUND

2.1. Existing System Analysis

Before developing the *Community Health Check and Telemedicine System*, it is important to analyze the currently available systems—both manual and digital—that provide similar healthcare services. Understanding how these systems work, along with their strengths and weaknesses, helps identify the improvements needed in the proposed solution.

1. Manual Community Health Service System

Many rural and semi-urban communities still rely on traditional manual healthcare processes. Patients visit local clinics or health camps for basic check-ups, medical advice, and treatment.

Pros:

- Simple and easy for less tech-savvy people.
- Direct physical interaction with health workers.
- Low initial cost to set up basic health camps.

Cons:

- Requires long waiting times due to patient overload.
- Limited availability of doctors and diagnostic devices.
- No organized record-keeping; patient histories often get lost.
- Difficult for elderly or remote-area residents to travel frequently.
- Emergency cases often remain untreated due to delays.

2. Digital Telemedicine Platforms (Existing Apps/Websites)

Some digital telemedicine services exist in Bangladesh and globally, such as **Doctorola, Maya App, PRAAVA telehealth,** and international services like **Teladoc**. These platforms allow users to consult doctors online and access basic health support.

Pros:

- Enables remote consultation via video/audio call.
- Reduces travel time and cost for patients.
- Provides access to specialist doctors nationwide.
- Some platforms support digital prescriptions and chat-based advice.

Cons:

- Most platforms do *not* include community-level health check tools (BP, glucose, BMI tracking).
- Many services are expensive for rural or low-income users.
- Limited integration with Electronic Health Records (EHR).

- No unified system combining **health screening + EHR + telemedicine + community monitoring**.
- Requires stable internet connection, which is a challenge in remote areas.

---

3. Local Health Camps and NGO-Based Health Services

Organizations sometimes arrange temporary health camps or mobile clinics within communities.

Pros:

- Brings basic health services directly to people.
- Useful for awareness campaigns, vaccinations, or initial screening.
- Provides short-term support during emergencies.

Cons:

- Not available year-round; only temporary.
- No digital record system for follow-up care.
- Limited doctor availability and tools.
- Cannot provide continuous monitoring or remote consultation.

2.2. Supporting Literatures

The development of the *Community Health Check and Telemedicine System* is supported by various theoretical concepts, technological tools, and methodological approaches that ensure an effective, scalable, and user-friendly solution.

1. Theoretical Knowledge

- **Health Informatics:** The project applies principles of health informatics to collect, store, and manage patient data electronically, ensuring accuracy, security, and easy retrieval.
- **Telemedicine Theory:** Remote consultation methods and digital healthcare delivery principles guide the design of video/audio consultations, appointment scheduling, and patient-doctor communication.
- **Systems Analysis and Design:** Structured system analysis techniques, such as requirement gathering, use case modeling, activity diagrams, and data flow diagrams, were applied to create an organized workflow.

Methodological Knowledge

- **Software Development Life Cycle (SDLC):** The project followed SDLC phases including requirement analysis, system design, implementation, testing, and deployment.
- **Agile Methodology:** Agile principles were applied to iteratively develop the system, allowing testing, feedback, and refinement during the development process.

3. Tools and Technologies

- **Frontend Development:** ReactJS was chosen for building a dynamic, responsive, and interactive user interface.
- **Backend Development:** Node.js/Python (Flask/Django) was used to handle server-side logic, data processing, and API integration.
- **Database:** MySQL was used for managing structured health data, ensuring reliability and easy querying.
- **Telecommunication Tools:** WebRTC or similar APIs were used to implement secure video and audio consultations.
- **UI/UX Design Tools:** Figma or Adobe XD helped design intuitive interfaces for patients and doctors.
- **Security Tools:** Encryption protocols (SSL/TLS) and authentication frameworks were applied to protect sensitive patient data.

4. Reason for Choosing These Tools

These tools and techniques were selected to provide a **scalable, secure, and efficient system**. ReactJS ensures a smooth and responsive frontend, while Node.js/Python and MySQL provide a reliable backend for storing and retrieving large amounts of medical data. The integration of telecommunication APIs enables real-time consultations, which is a core functionality. Security measures were prioritized to comply with ethical standards and protect patient confidentiality.

By combining these theoretical, methodological, and technological approaches, the system ensures **effective health monitoring, easy access to telemedicine, and organized patient record management**, fulfilling the project objectives.

5. Technological Knowledge

- **Cloud Computing & Web Hosting:** The system leverages cloud infrastructure to ensure scalability, reliability, and remote access from anywhere. Cloud-based deployment allows multiple users to access the system simultaneously without performance issues.
- **Data Analytics & Reporting:** Basic analytics and reporting techniques are applied to monitor community health trends, generate insights, and assist healthcare providers in decision-making.

6. Integration and Interoperability

- **API Integration:** APIs are used to integrate telemedicine services, third-party diagnostic tools, and health monitoring devices into the system.
- **Interoperability Standards:** HL7/FHIR or similar healthcare data standards are considered to ensure seamless exchange of medical information between different modules and future systems.

Chapter 3
SYSTEM ANALYSIS & DESIGN

3.1. Technology & Tools

This project uses a combination of modern development technologies, AI frameworks, and development tools to ensure a secure, scalable, and intelligent platform. The chosen technologies align with current industry standards and fully support the functional requirements of Community Health Checkup & Telemedicine Portal, including real-time communication, authentication, automated moderation, saving data of patient and doctor , providing real analysis data.

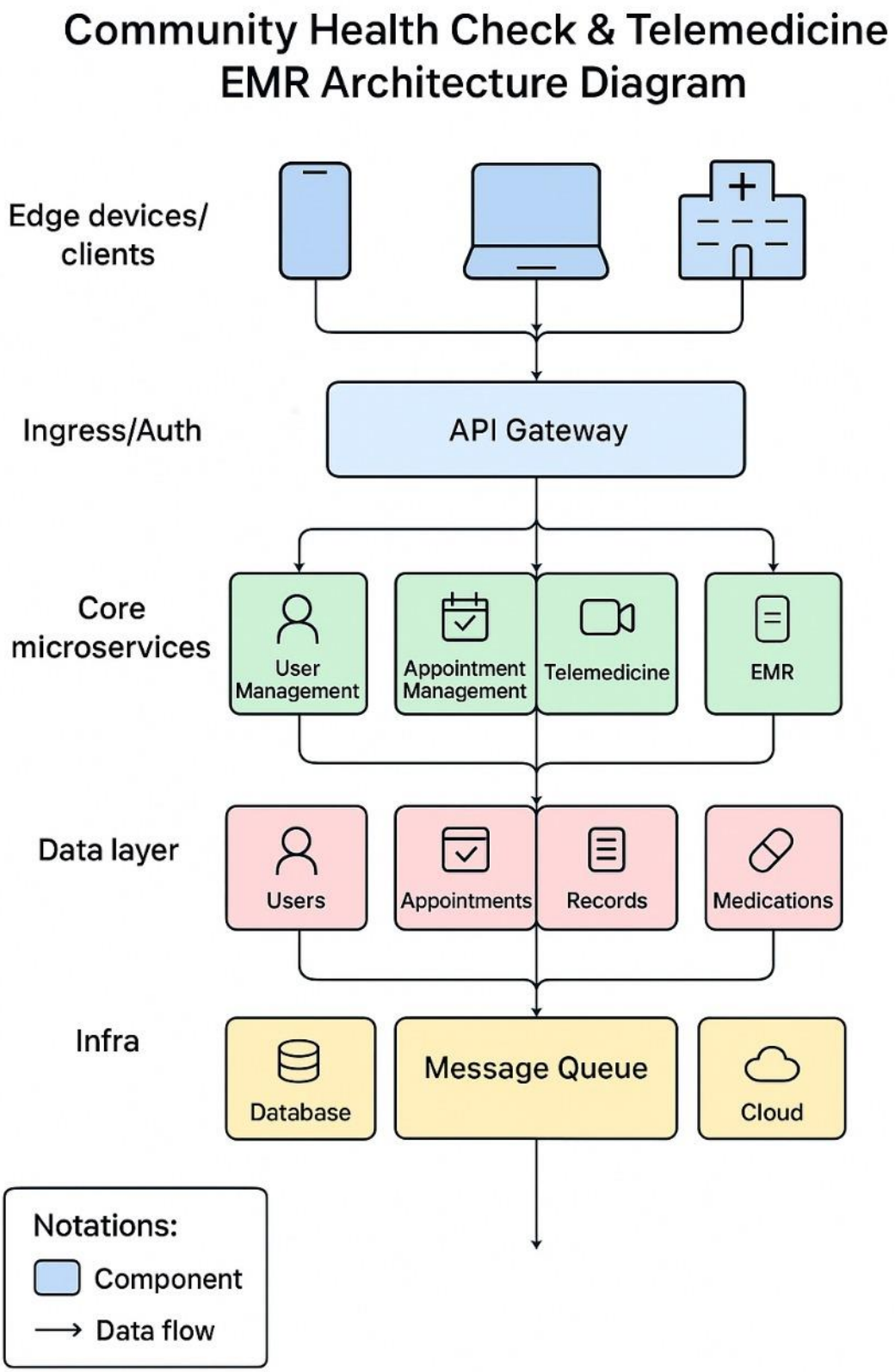
3.2. Model & Diagram

This Section describe the logical design and visual modeling of the Community Health Checkup website.This model include Use-Case diagrams, System Architecture ,E-R dia Gram,DFD,contxt level DFD, flowchart which will give Understanding os system workflows, user interactions, and backend processes. These diagrams help ensure the system architecture is well-organized, consistent, and aligned with the project's overall functional requirments

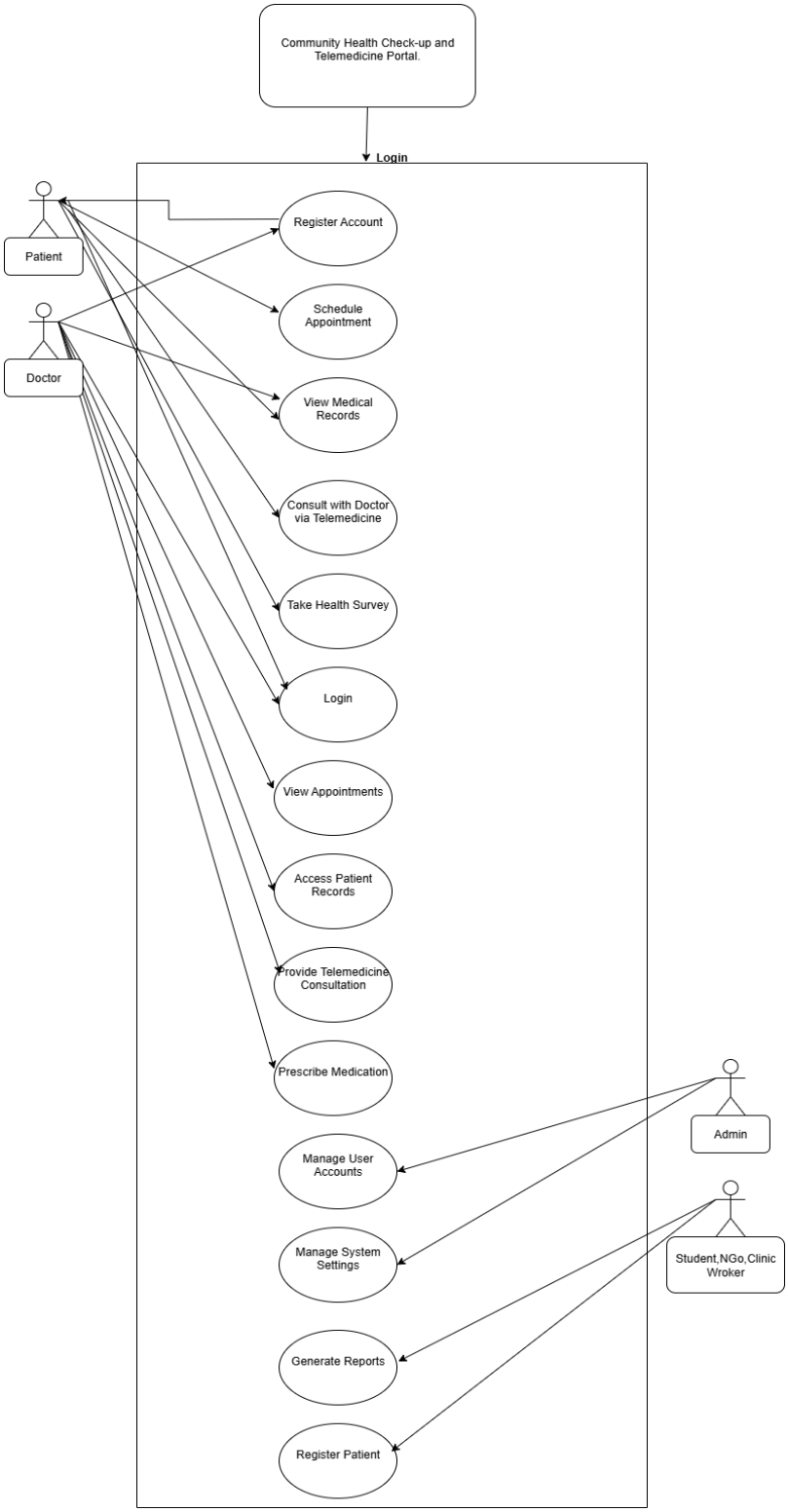
3.2.1. Model (SDLC/Agile/Waterfall/OOM)

This project follows an Agile and iterative development model, allowing the system to evolve through continuous planning, development, testing, and improvement. Instead of building the entire system at once, the features were developed in small, manageable iterations, enabling regular feedback from users and stakeholders. This approach helped refine requirements, improve system quality, and quickly adapt to changes throughout the project lifecycle. The iterative nature ensured that each module of the Community Health Checkup & Telemedicine Portal was tested, evaluated, and enhanced before proceeding to the next phase, resulting in a more flexible, scalable, and user-focused system.

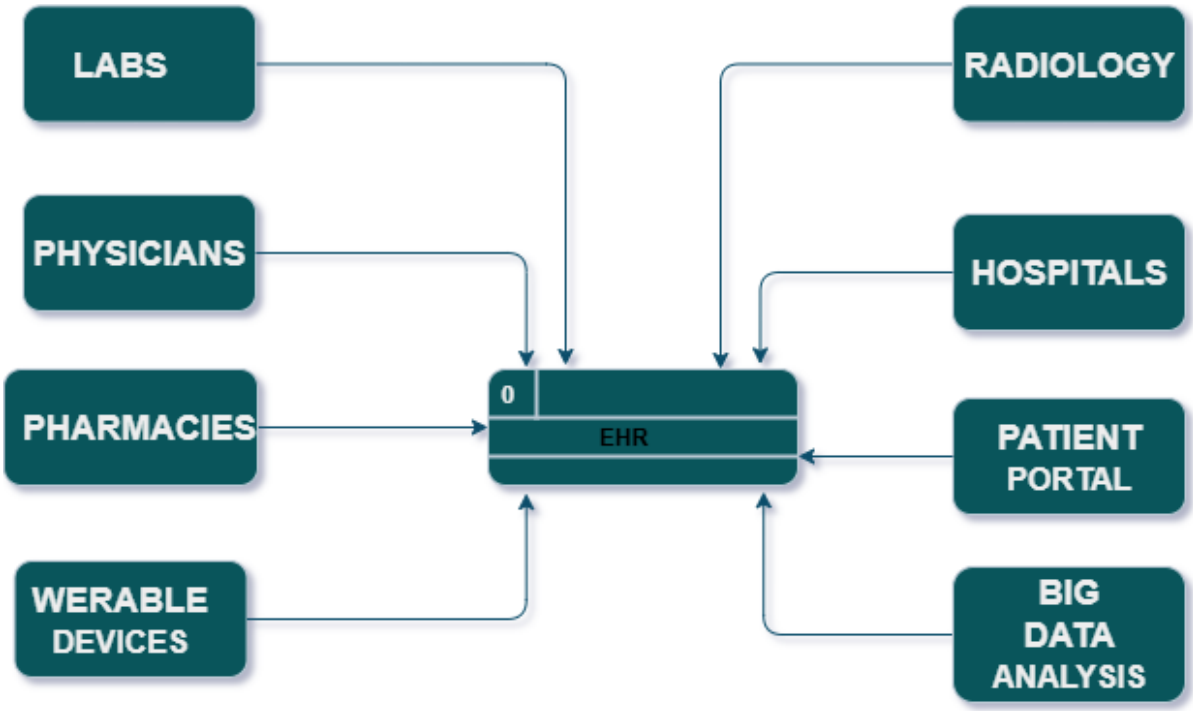
3.2.2. System Architecture



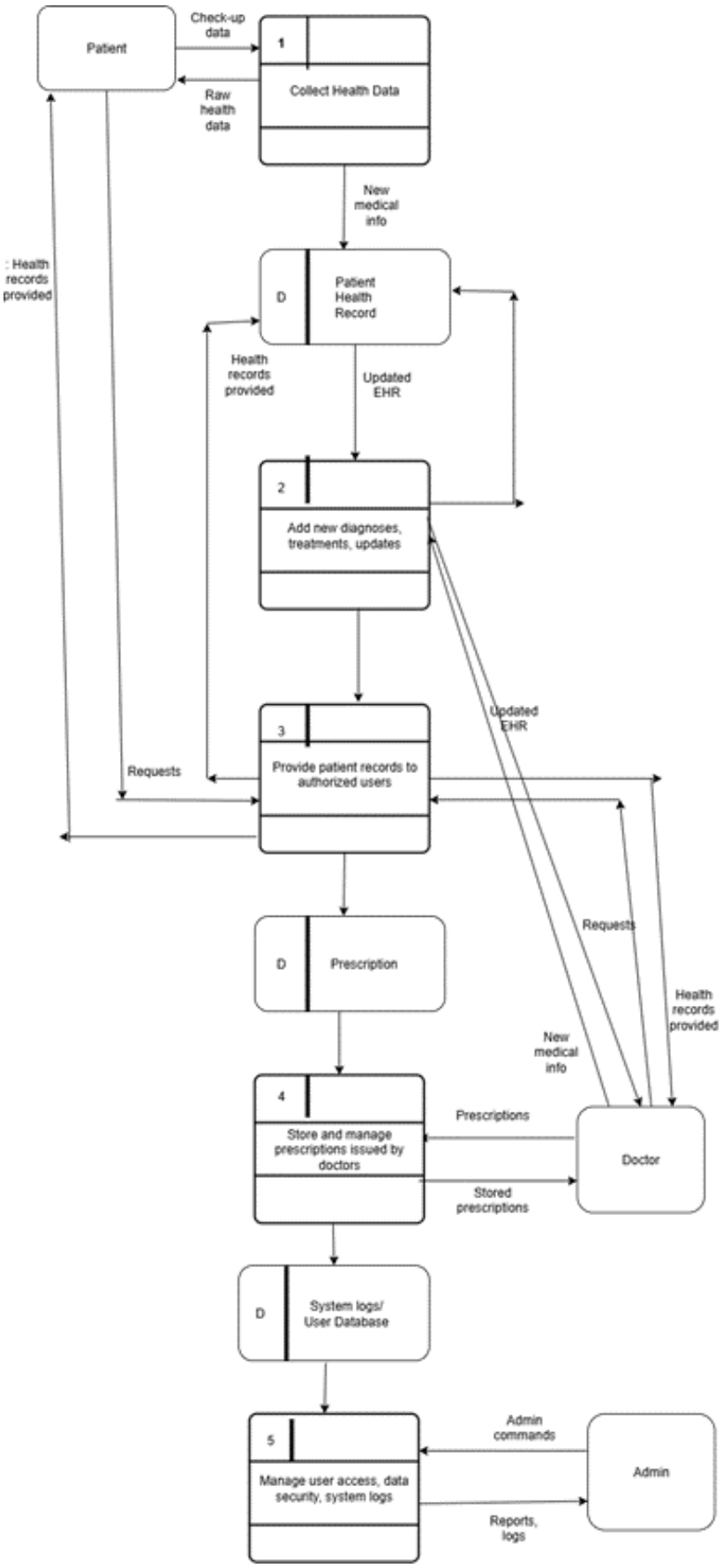
3.2.3. Use Case Diagram



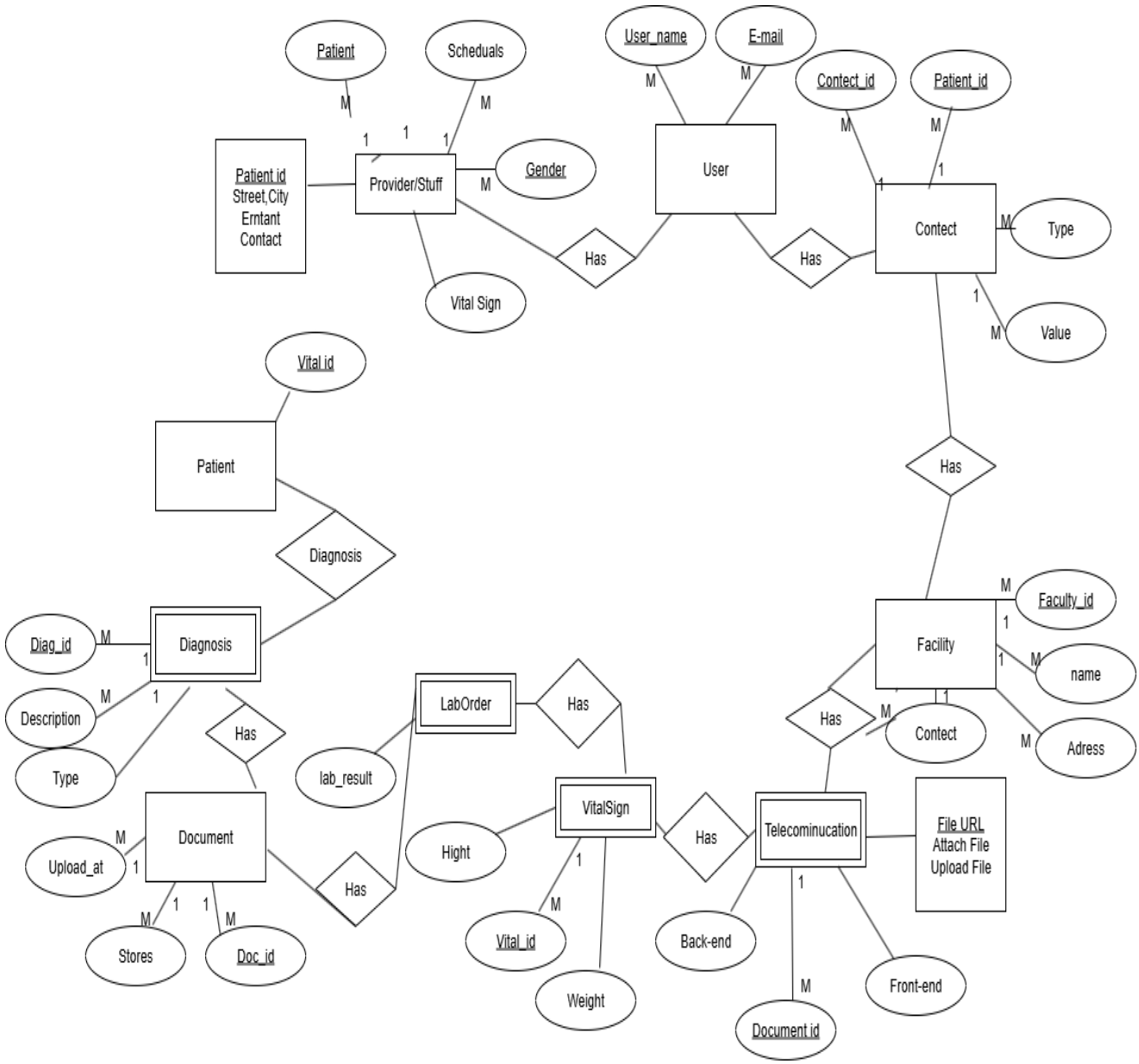
3.2.4. Context Level Diagram



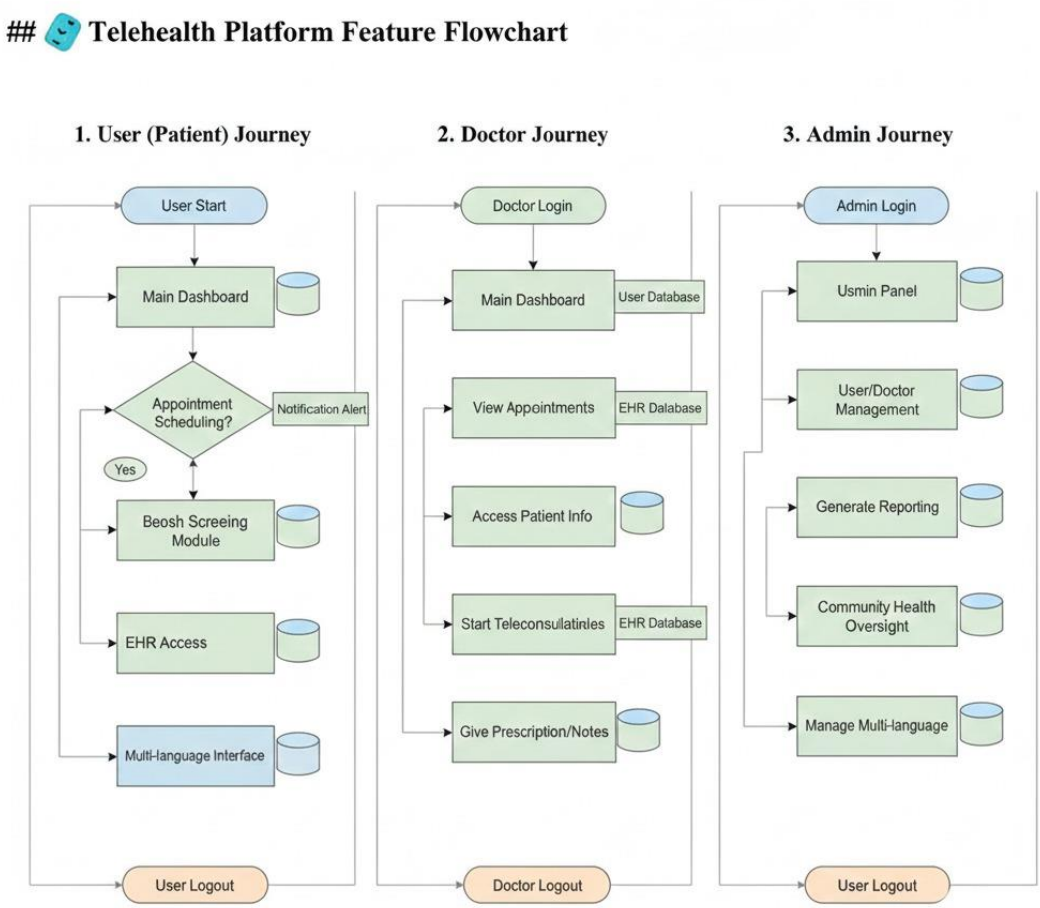
3.2.5. Data Flow Diagram



3.2.6. Database Schema



3.2.7. Algorithms/Flowchart



Chapter 4
IMPLEMENTATION

4.1. Front-End Design

The front-end interface of the Community Health Checkup & Telemedicine Portal was designed with a focus on clarity, accessibility, and a modern healthcare-themed UI/UX approach. The visual design emphasizes smooth navigation and an intuitive experience for patients, doctors, and administrators.

The frontend was developed using **javascript**, supported by **TailwindCSS** for consistent and responsive styling and DaisyUI for responsive.Taliwind cart-based structure allowed the team to build reusable UI blocks such as dashboards, appointment cards, health record tiles, and teleconsultation interfaces. This significantly improved development speed and maintained a uniform look throughout the system.

Key interface screens include:

- **Landing Page UI** featuring registration/login options, a hero section, and public health information browsing.
- **Patient Dashboard** displaying health records, prescriptions, appointments, notifications, and screening options.
- **Booking Module UI** for selecting doctors, choosing date/time, and confirming consultations.
- **EHR & Prescription UI** enabling patients to view their medical history, screening data, and doctor-issued prescriptions.
- **Live Teleconsultation Interface** with a placeholder space for integrated video/audio calls (Jitsi).
- **Profile & Settings UI** for updating user information and preferences.

TailwindCSS ensured strong responsiveness across desktops, tablets, and mobile screens, while maintaining a clean, healthcare-friendly theme with dark-mode elements, soft shadows, and smooth animations. The final design prioritizes readability and patient comfort, aligning with modern digital health platforms.

4.2. Back-End Design

The backend of the Community Health Checkup & Telemedicine Portal was implemented using **Spring Boot (Java 17+)**, following a well-structured layered architecture:

Controller → Service → Repository → Database

This structure ensured clean separation of concerns, reduced duplicated logic, and allowed the project to scale with ease.

Core Backend Functionalities Developed

1. Authentication & Authorization

- User login, registration, logout, and role assignment (Patient, Doctor, Admin).
- JWT-based authentication ensures secure access to resources.
- Password hashing and token validation protect sensitive healthcare data.
- Role-based access control (RBAC) restricts features:
    - Patients access EHR, screening, appointments.
    - Doctors access teleconsultation, prescriptions, patient records.
    - Admins manage system-level data.

---

2. CRUD Operations

The system includes full CRUD functionalities for all essential healthcare modules.

Module	CRUD Operations Implemented
Users	Create account, update profile, change password
Health Screening	Add BP/Glucose/BMI data, view history, update entries
Appointments	Book, update status, cancel, view history
Prescriptions	Doctor creates prescriptions, patient views them
Health Records (EHR)	Store, view, update medical information
Teleconsultation	Manage consultation sessions and status

CRUD operations were implemented using **Spring Data JPA + MySQL**, ensuring fast, secure, and efficient database handling with minimal boilerplate.

4.3. Modules / Features

Below is the complete list of modules and features implemented in the Community Health Checkup & Telemedicine Portal:

---

1. User Authentication Module

- Registration & Login
- JWT-secured authentication
- Role-based access control (Patient, Doctor, Admin)
- Profile update & password management

---

2. Patient Dashboard Module

- Displays appointments, notifications, prescriptions, EHR, and screening summary
- Personalized greeting and quick-access cards
- Real-time health overview

---

3. Appointment Booking Module

- Patients select doctor, date, and time
- Doctors view pending and upcoming appointments
- Appointment status updates (Confirmed / Pending / Cancelled)
- Automated notification display

---

4. Health Screening Module

- Input fields for BP, Glucose, BMI, and health metrics
- Stores data in database
- Patients can track their screening history
- Used by doctors during consultations

---

5. Electronic Health Records (EHR) Module

- Stores patient medical history
- Accessible to both patients and doctors
- Integrated with prescriptions and screening data

---

6. Prescription Module

- Doctors issue digital prescriptions
- Patients view and download prescriptions
- Includes medication list, dosage, and notes

---

7. Teleconsultation Module

- Live video/audio session window (Agora/Jitsi placeholder integrated)
- Patients and doctors can conduct consultations online
- Session ends with follow-up actions (Prescription, screening request)

---

8. Notifications Module

- Appointment confirmation
- New prescription alerts
- Screening reminder notifications
- Consultation reminder

---

9. Profile & Settings Module

- Update profile information
- Change account details
- Multilanguage option (English + others if needed)

10. Admin Management Module

- Manage users (doctor/patient accounts)
- Manage system-level data
- Monitor appointments and screening logs

Chapter 5
USER MANUAL

5.1. System Requirements

5.1.1. Hardware Requirements
Minimum Recommended Hardware Configuration:

- **Processor:** Intel Core i3 (7th Gen) or AMD equivalent
- **RAM:** 4 GB (Recommended: 8 GB)
- **Storage:** Minimum 10 GB free space
- **Display:** 1280×720 resolution or higher
- **Internet:** Stable broadband connection (required for telemedicine video sessions)
- **Peripherals:** Webcam, microphone, and speakers/headphones

---

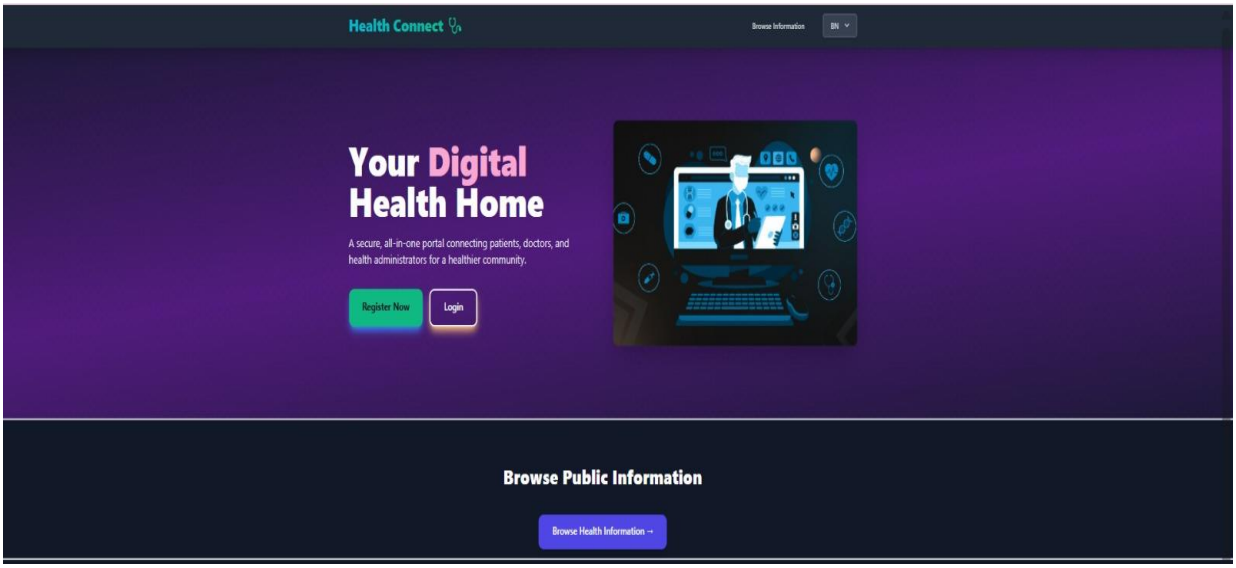
5.1.2. Software Requirements
Minimum Recommended Software Configuration (Java Backend):

- **Operating System:**
  - Windows 10 / 11
  - Ubuntu 20.04+
  - macOS 10.15+
- **Backend Environment:**
  - Java JDK 17+ (LTS Preferred)
  - Spring Boot 3.x Framework
  - Apache Maven / Gradle
- **Database:**
  - MySQL 8+ / PostgreSQL
  - Hibernate / JPA
- **Frontend Requirements:**
  - HTML5, CSS3, JavaScript
  - (Optional) React / Vue / Angular
- **Testing & Development Tools:**
  - Postman or Swagger UI
  - VS Code / IntelliJ IDEA / Eclipse
- **Application Server:**
  - Spring Boot Embedded Tomcat
  - (Optional) External Apache Tomcat 9/10
- **Browser Support:**
  - Google Chrome (Latest)
  - Mozilla Firefox (Latest)
  - Microsoft Edge
- **Additional Libraries/Dependencies:**
  - Spring Security
  - Lombok
  - Jackson (JSON Processor)
  - WebRTC / Jitsi Integration (for video consultation)

5.2. User Interfaces

Below are the key modules and interface descriptions of the **Health Check & Telemedicine System.**

First UI Look

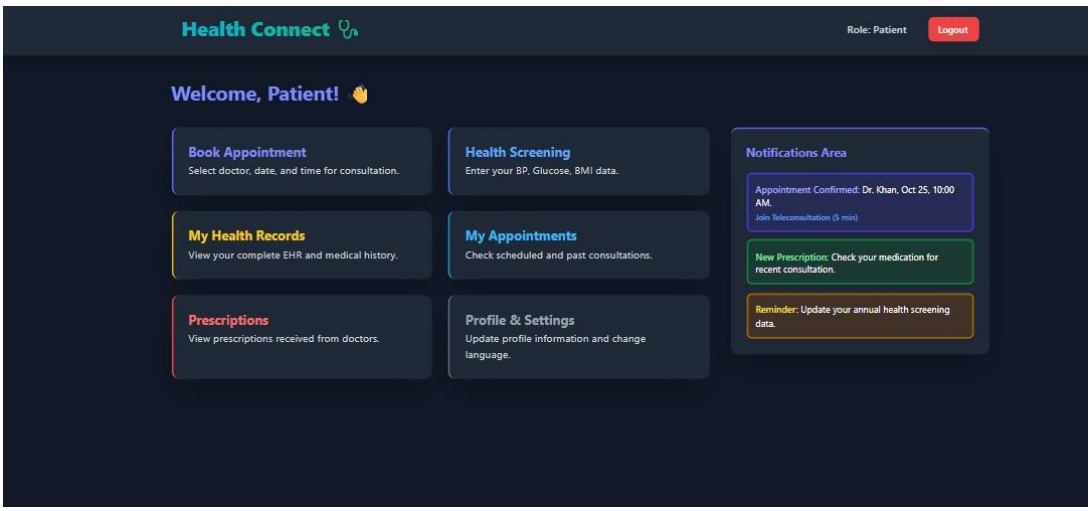


5.2.1. Patient Panel (Panel A)

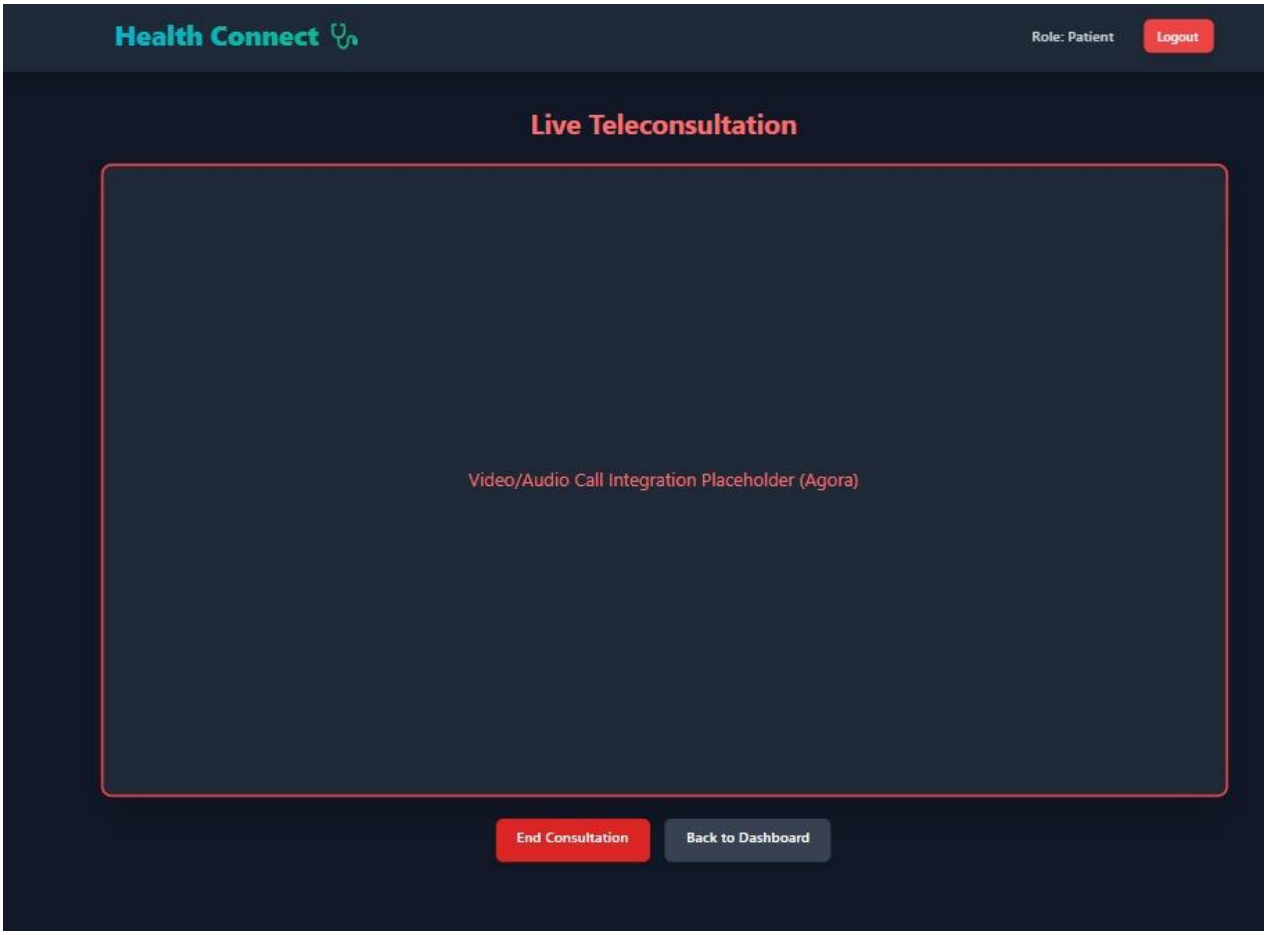
Features:

- Patient Dashboard
- View & Update Profile
- Add Symptoms / Submit Health Information
- Download Health Reports
- Book Telemedicine Appointments
- Start Video Consultation

Feature List Look



Video Call Feature Look



5.2.2. Doctor Panel (Panel B)

Features:

- Doctor Dashboard
- View Assigned Patients
- Approve or Reject Appointments
- Start Online Video Consultation
- Upload Prescriptions
- Review Patient Health History

5.2.3. Admin Panel (Panel C)

Features:

- Manage Doctor Accounts
- Manage Patient Accounts
- View System Logs
- Generate Analytical Reports
- Monitor Telemedicine Sessions
- Update System Settings

5.2.4. Additional Interface Modules

- Authentication Panel (Login/Registration)
- Health Monitoring Panel (IoT/Sensor-based if included)
- Report Upload Panel

- Appointment Management Interface

5.2.5. Login Credentials (Demo Only)

Below are sample login credentials for testing and demonstration:

Patient Login

- **Username:** patient01
- **Password:** Patient@123

Doctor Login

- **Username:** doctor01
- **Password:** Doctor@123

Admin Login

- **Username:** admin
- **Password:** Admin@12345

Chapter 6
CONCLUSION

6.1 Conclusion

The **Health Check & Telemedicine System** has been developed with the intention of bringing healthcare services closer to users through digital innovation. This project successfully integrates essential features such as patient registration, doctor management, appointment scheduling, health report handling, and real-time telemedicine consultations in a unified platform.

Throughout the development process, we gained valuable hands-on experience with **Java (Spring Boot)**, **RESTful API design**, **relational database management**, and **secure user authentication**. We also enhanced our understanding of system architecture, user interface design, and effective communication between frontend and backend modules.

This project demonstrated how technology can significantly bridge the gap between patients and healthcare providers—reducing waiting times, improving accessibility, and making remote healthcare a practical reality. Overall, the system reflects our learning, collaborative effort, and commitment to building a meaningful solution for modern healthcare needs.

6.2 Limitations

Although the system performs its core functionalities effectively, certain limitations remain due to time, resources, and project scope:

1. **Video calling performance depends on network stability**, which may affect telemedicine sessions.
2. **No AI-powered medical prediction or diagnosis support** has been implemented.
3. The system currently lacks **online payment processing** for consultation fees.
4. **Real-time IoT-based health monitoring** (e.g., heart rate, blood pressure sensors) is not fully integrated.
5. Doctor scheduling is **manual** and could benefit from automation.
6. The user interface, while functional, can be further optimized for a more commercial-grade experience.
7. The system does not support **multi-language accessibility** for different user groups.
8. **Scalability challenges** may arise if the number of users increases rapidly, requiring server load balancing and optimization.
9. It lacks **role hierarchy** beyond basic Admin/Doctor/Patient roles, making permission control less flexible.

These limitations highlight areas where enhancements can greatly improve user experience and system performance.

6.3 Future Works

The project has significant potential for further development and can be transformed into a complete digital healthcare ecosystem. Future enhancements may include:

1. **AI-driven symptom analysis** and predictive health analytics.
2. **Full IoT integration** for real-time vital monitoring through wearable sensors.
3. **Secure online payment system** for appointment fees and premium medical services.
4. **Advanced doctor scheduling system** with automated availability and notifications.
5. Dedicated **Android and iOS mobile applications** for better accessibility.
6. **E-pharmacy module** for online medicine ordering linked to digital prescriptions.
7. Integration with **Electronic Health Record (EHR)** systems for nationwide medical data access.
8. **Machine Learning models** for disease risk assessment and early detection.
9. **Digital signature & QR-based verification** for prescriptions and medical documents.
10. **Multilingual support** to make the platform accessible to diverse users.

Implementing these future improvements will transform the current project into a robust, scalable, and intelligent telemedicine platform capable of serving users at a national level.

References

[1] “Data Flow Diagram?” Draw.io. [Online]. Available: www.Draw.io.com/data-flow-diagram
. [Accessed: 24-Nov-2025].

[2] “Telemedicine: A Guide to Virtual Healthcare,” World Health Organization (WHO). [Online]. Available: www.who.int/health-topics/telemedicine
. [Accessed: 24-Nov-2025].

[3] “Unified Modeling Language (UML) Diagrams,” Visual Paradigm. [Online]. Available: www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-diagrams/
. [Accessed: 24-Nov-2025].

[4] “Spring Boot Documentation,” Spring.io. [Online]. Available: spring.io/projects/spring-boot. [Accessed: 24-Nov-2025].

[5] “Java Platform Standard Edition Documentation,” Oracle. [Online]. Available: docs.oracle.com/javase/. [Accessed: 24-Nov-2025].

[6] “MySQL 8.0 Reference Manual,” MySQL Documentation. [Online]. Available: dev.mysql.com/doc/. [Accessed: 24-Nov-2025].

[7] “RESTful Web Services,” Mozilla Developer Network (MDN). [Online]. Available: developer.mozilla.org/en-US/docs/Web/Services/REST. [Accessed: 24-Nov-2025].

[8] “What Is Telemedicine?” HealthIT.gov. [Online]. Available: www.healthit.gov/faq/what-telehealth-what-telemedicine
. [Accessed: 24-Nov-2025].

[9] “WebRTC API,” Mozilla Developer Network (MDN). [Online]. Available: developer.mozilla.org/en-US/docs/Web/API/WebRTC_API. [Accessed: 24-Nov-2025].

[11]”Jitsi Video call open source”.[Online].Available: https://jitsi.org/api/ [Accessed: 24-Nov-2025].

[11] “Software Engineering Documentation Standards,” IEEE. [Online]. Available: standards.ieee.org. [Accessed: 24-Nov-2025]