HealthSync

"Intelligent Telehealth & Remote Patient Monitoring Platform"

Group Members

Muhammad Moosa Khalil (01-131222-035)

Tayyab Aamir Ali (01-131222-048)

Supervisor: Engineer Sulman Zafar

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad, in partial fulfillment of the requirements for the award of the degree of Bachelor of Software Engineering.

May 2026

FYP COMPLETION CERTIFICATE

Student Name: Muhammad Moosa Khalil Enrolment No:01-131222-035

Student Name: Tayyab Aamir Ali Enrolment No:01-131222-048

Programme of Study: Bachelor of Software Engineering

Project Title: Health Sync Intelligent Telehealth & Remote Patient Monitoring Platform

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at ___________ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Name: Engineer Sulman Zafar

Supervisor's Signature: ___________________________

Date: __________________________ Name: _____________________________

Certificate of Originality

This is certified that the intellectual contents of the project Health Sync Intelligent Telehealth & Remote Patient Monitoring Platform are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: Muhammad Moosa Khalil

Signature: _______________________ Date: _________________

Name of the Student: Tayyab Aamir Ali

Signature: ________________ Date: _________________

SUSTAINABLE DEVELOPMENT GOALS

HealthSync (Remote Patient Monitoring & Intelligent Telehealth Platform)

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	SDG 9	Industry, Innovation, and Infrastructure ✓
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities ✓
SDG 3	Good Health and Well Being ✓	SDG 11	Sustainable Cities and Communities
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Range of Complex Problem Solving

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	✓
4	Familiarity of issues	Involve infrequently encountered issues	
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	✓
7	Consequences	Have significant consequences in a range of contexts.	✓
8	Interdependence	Are high level problems including many component parts or sub-problems	✓
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	✓
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	✓
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	✓
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Abstract

HealthSync was built with Next.js 15, React 19, PostgreSQL and Prisma ORM. The system has been designed in such a way that they have four distinct user roles such as Super Administrator, Clinic Administrator, Healthcare Provider, and Patient. The roles offer a tailored dashboard display and distinct workflows that are user specific. Moreover, the system also enforces strict control over access and authorization using Role-Based Access Control (RBAC) and JWT Authentication.
To achieve these two, HealthSync will be in a position to integrate directly with the Fitbits of patients through OAuth 2.0 to get the real-time and continuous vital signs information. It is through this connection that we are able to get minute-by-minute heart rate, total daily steps taken, sleep patterns, peripheral oxygen saturation (SpO2), Heart Rate Variability (HRV), and respiratory rate. After collection of the data, it is processed by a three-phase AI Engine. Phase 1 uses rule-based constraints depending on the age and medical history of the patient and the medication he/she has been prescribed now. Phase 2 performs a statistical analysis to determine the trend of the resting heart rate identification, determines abnormal data changes by calculating Z-score, and traces the data by relying on the calculation of a linear regression. Phase 3 runs the Cerebras Llama 3.3-70B Large Language Model (LLM) to generate clear and concise medical summaries, assess the risk to health of the identified problem(s), and provide actionable next-steps.

To get access to continuous and real-time vital sign readings of patients, HealthSync can directly interoperate with the Fitbits of patients via OAuth 2.0. It is through this relationship that we are able to gather minute by minute heart rate, total daily steps taken, sleep patterns. After collecting the data, it is processed by a three-phase AI Engine. Phase 3 applies the Cerebras Llama 3.3-70B Large Language Model (LLM) to generate clear and concise medical summaries, assess the risk to health presented by the detected problem(s), and suggests actionable next steps.

Keywords: Healthcare Management System, Remote Patient Monitoring, AI Clinical Decision Support, Wearable IoT Integration, Telehealth, WebRTC, Role-Based Access Control, Next.js, Large Language Model

Dedication

This dedication is devoted to our parents, whose love, sacrifices, and continuous encouragement made this achievement possible. Without them, we would not have been able to complete this work.We also wish to acknowledge our teachers and mentors for their guidance and support throughout this journey. They helped us understand our objectives and taught us to approach our work with critical thinking and dedication. Above all, this project is dedicated to everyone who deserves access to better-quality healthcare. In the end, we created this for the benefit of the patients.

Acknowledgments

We are deeply grateful to our supervisor who has been our guide all along, provided tremendous feedback and helped us throughout the process of creating HealthSync.

Their profound knowledge about software engineering and health-care technologies was key in achieving success in this venture.

Moreover, our gratitude is extended to the teachers from the Software Engineering faculty for their support and providing us with necessary knowledge and skills needed to develop this project.

Another important source of help for us has been the open-source community. Without wonderful development tools such as Next.js, React, Prisma, PeerJS, and Cerebras AI Platform, we would not be able to create this platform.

Lastly, we express our sincere thanks to all of our friends and family members for supporting us throughout the process.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The healthcare industry is undergoing a significant digital transformation driven by the need for improved patient outcomes, operational efficiency, and accessibility. Traditional clinic management systems often operate in silos, lacking integration between appointment scheduling, medical records, prescription management, and patient communication. Furthermore, the emergence of lot wearable devices and AI-powered analytics presents an unprecedented opportunity to shift from reactive to proactive healthcare delivery through continuous remote patient monitoring.

HealthSync addresses these challenges by providing a unified, full-stack web platform that integrates clinic management workflows with real-time wearable device data and AI-driven clinical decision support, enabling healthcare providers to deliver more informed, timely, and personalized care.

1.1 Motivation

HealthSync was created because of a number of problems inherent to the current state of technology in the sphere of medicine:

- **Separate Clinic Workflow Platforms:** Nowadays, clinics make use of totally different platforms for appointment scheduling, management of electronic medical record systems, prescription making, and even patient communication. As a result, useful data is being stored separately, making the process inefficient while forcing healthcare professionals to continuously shift between different programs, which is both stressful and error prone.

- **Inability to Use Data Generated by Consumer Health Technologies:** People can use Fitbit or any other device to monitor their health status and obtain up-to-date information regarding their heart rate, sleep quality, SpO2, etc. Still, this precious information is not available to doctors and cannot be used for enhancing patients' health condition. Even though people constantly wear those devices, it does not help medical professionals benefit from this data.

- **No Use of AI in Clinics:** While AI technologies keep developing actively nowadays, there are no recent innovations based on artificial intelligence, including LLMs, incorporated into the clinic workflow.

- **Failure to Optimize Telehealth Solutions:** Undoubtedly, the pandemic of the virus has provided us with several takeaways regarding telehealth advantages. However, despite all that happened, many current health IT solutions see telehealth as an auxiliary function rather than incorporating it in the core functions.

- **Accessibility:** Quite often, people residing in rural or other remote regions may face obstacles when trying to visit medical institutions for examinations. With the help of our system and its features such as remote monitoring and communication via real-time calls, access can be greatly increased.

1.2 Objectives

Some of the specific objectives that need to be attained during this project include:

1. Creating an integrated system for the management of clinical practices through integration of appointment scheduling, patient recordkeeping, prescription capabilities, patient-doctor chat services, and RBAC-based user authentication.

2. Attaining the capability to perform real-time remote monitoring of patients by linking up the patients' Fitbits using OAuth 2.0, which will help achieve synchronization and display of patients' vital signs at hourly and minute levels.

3. Creation of a sophisticated multi-tier AI solution by making use of personalized rule-based thresholds, statistics-based techniques (trend and z-score anomaly detection) and Cerebras Llama 3.3-70b Large Language Model for creating accurate clinical reports.

4. Creation of video conferencing capability with WebRTC technology (PeerJS).

5. Provide a secure platform with five distinct user roles. In this case, it is vital to consider the full isolation of the two distinct Super Admin and Clinic Admin roles, together with their distinct responsibilities, and the other roles of Providers and Patients. All users shall have exclusive rights, data segregation, and dashboards.

6. Create an e-prescribing platform that works seamlessly with the existing database that has 8,557 medications, including all steps starting from prescription writing to its handling.

7. Add a secure instant messaging service for both patients and providers. The messaging system would include triaging messages, editing, deleting, and reading receipt features.

8. Deploy the application on the Vercel cloud platform. Further, we will use automated cron jobs for scheduling AI visits, cleansing data, and notifying patients.

1.3 Main contributions

These are just some of the innovative features that HealthSync introduces into the sphere of digital health technologies:

1. A Three-Phase Personalized AI Health Engine

Although the majority of alert systems employ a fairly standard methodology for analyzing the patients' state, the HealthSync system uses a personalized AI analysis of each patient. In particular, heart problems are detected through an analysis of rule-based personalized thresholds, multiple statistics (z-score, linear regression), and Cerebras Llama 3.3-70B LLM writing a clinical report based on the analysis of all indicators. This is the first possibility of conducting an artificial intelligence assessment of the patient's heart condition in the clinic environment without the use of expensive equipment typical for hospital settings.

2. Real-Time Wearable Integration at Clinical Granularity

In fact, this technology collects data provided by the client's personal devices (e.g., Fitbit). More specifically, the data about heart rate, SpO2, HRV, sleep patterns, breathing, and more is recorded and displayed to physicians during their work with the clients.

3. Pre-Consultation AI Briefing

A brief about the patient's disease is generated by AI in the last ten minutes before the consultation begins. This brief is created on the basis of the patient's symptoms, medical history, current medications, and lab results. This information is instantly shared with the physician who sees the patient. Therefore, less time is spent on preparing the diagnosis, which is an option that can't be found in any other open-source clinic management system.

4. End-to-End Integrated Clinical Platform

HealthSync brings all elements of clinic operation together through a single application considering different user profiles. HealthSync enables scheduling consultations, conducting video consultations via WebRTC protocol, managing the patient's medical history, issuing prescriptions among 8,557 available medicines, sending messages instantly, and measuring vital signs. Earlier, these operations could be carried out only through several applications.

5. Accessible Deployment for Resource-Constrained Clinics

This web app is being deployed using Vercel approach with automatic maintenance and open APIs using cron. This shows that it is indeed feasible to develop a high-quality AI-based healthcare web app with zero cost server infrastructure. With such features, such applications are highly appropriate for clinics in developing regions owing to affordability and deployment.

1.4 Report organisation

The following are the approaches that will be used in organizing the report:

- In chapter two, there is a discussion about the background studies and the literature review. There is a discussion on health care management systems, patient monitoring techniques, AI-based applications in healthcare and telehealth. Furthermore, there is a discussion on the particular problem statement being solved by HealthSync.

- In chapter three, there are system requirements including use case diagrams, functional requirements, non-functional requirements and interface requirements. There is a further description on database requirement and project feasibility.

- In chapter four adescribes the system design. Here the design of the system architecture, database design, and API design will be described using 28+ Prisma models and 60+ REST endpoints correspondingly. Also, the design of the components and the UI will be provided.

- In chapter five focuses on the system implementation. Here the tools and technologies used for creating the system will be described, and the steps of development will be given.

- In chapter six provides information about testing and evaluating the system. Tests will include unit tests using Jest and end-to-end tests using Playwright, and the results will be given using examples of the test case.

- In chapter seven is the conclusion. Here we give the conclusions about the contributions made by us, as well as suggestions for further studies.

Chapter 2
Background Study/Literature Review

This chapter looks at the latest technology used in healthcare platforms, remote patient monitoring, AI-driven clinical decision support, and telehealth systems. It sets the stage for exactly why we built HealthSync. We will also point out the missing features in current systems that inspired us to start this project.

2.1 Key Concepts

2.1.1 Clinic Management Systems(CMS)

Clinic Management Systems are software platforms built to manage the daily administrative and clinical tasks inside healthcare facilities. These systems usually handle core jobs like registering patients, booking appointments, managing medical records, billing, and creating reports. Today, modern CMS platforms are quickly shifting toward cloud-based architectures. This switch makes them easier to access, highly scalable, and much cheaper to run since clinics do not need to buy heavy server infrastructure.

2.1.2 Electronic Health Records (EHR)

Electronic Health Records are digital versions of patients' paper charts that provide real-time, patient-centered records accessible to authorized users. EHR systems contain the medical and treatment history of patients including diagnoses, medications, treatment plans, immunization dates, allergies, radiology images, and laboratory results. The SOAP (Subjective, Objective, Assessment, Plan) note format is the standard for clinical documentation.

2.1.3 Remote Patient Monitoring (RPM)

Remote Patient Monitoring involves the use of digital technologies to collect medical data from individuals in one location and electronically transmit it to healthcare providers in a different location for assessment and recommendations. RPM devices include wearable sensors (Fitbit, Apple Watch), blood pressure monitors, pulse oximeters, and glucose monitors. The global RPM market is projected to reach $117.1 billion by 2025.

2.1.4 Role-Based Access Control (RBAC)

RBAC controls access to data based on each individual's role within the organization. This concept is essential in creating data access guidelines in the healthcare sector because various individuals, such as the patient, clinician, or admin, will need access to different sets of data. Least privilege access ensures that only the required data is accessible by each user.

2.1.5 WebRTC and Telehealth

Web Real-Time Communication (WebRTC) refers to the standard technology that enables peer-to-peer communication through the use of web browsers without the need for any other applications. WebRTC forms the backbone of the current telemedicine services, which facilitate HIPAA compliant doctor-patient video consultations.

2.1.6 AI and Large Language Models in Healthcare

The application of AI technologies in healthcare is getting more common nowadays in terms of making clinical decisions, diagnoses, and forecasts on health trends. For instance, such systems as GPT-4, Llama, and Gemini can easily process complex information from the medical environment and create clinical insights and predictions as well as offer recommendations. The downside of this kind of technology lies in the inability to maintain complete accuracy of the obtained results (avoiding AI hallucination).

2.2 Existing Healthcare Management Systems

2.2.1 Existing Epic Systems

Epic is one of the largest vendors of electronic health records in the US market that provides various functionalities in terms of different workflows and patients' access to their own medical information (MyChart). However, there are several significant downsides of this system. First, the price of its implementation is extremely high. Moreover, it has been created specifically for hospitals, not for private clinics.

2.2.2 Cerner (Oracle Health)

Moving forward, there is another option, which is Cerner. The system allows healthcare organizations to control all the processes regarding clinical practices, financial issues, and others. Just like Epic, it was created for deployment in large medical organizations and hospitals. The implementation process takes plenty of time and money.

2.2.3 Practice Fusion

Another software is Practice Fusion, which is a cloud-based platform developed specifically for small clinics. This product can assist doctors in managing the progress of their patients by writing notes, making appointments, and managing prescriptions. Nevertheless, it does not include any technological innovations. Practice Fusion cannot be integrated with wearables, telemedicine tools, and AI-based analysis of the patient's health.

2.2.4 Athenahealth

The next one is Athenahealth, which can assist its users in storing all the necessary medical data, managing the whole process of revenue cycle, and engaging patients. Even though it includes telehealth services, the platform is not innovative enough technologically. It does not offer integration with wearables or AI-based analysis.

2.2.5 Comparison Summary

The table below shows a comparison of the products:

Feature	**Epic**	**Cerner**	**Practice Fusion**	**Athenahealth**	**HealthSync**
EHR / Medical Records	✓	✓	✓	✓	✓
Appointment Scheduling	✓	✓	✓	✓	✓
E-Prescriptions	✓	✓	✓	✓	✓

Video Consultation	✓	✓	✗	✓	✓
Patient Portal	✓	✓	✓	✓	✓
Wearable Device Integration	Limited	Limited	✗	✗	✓ (Fitbit)
AI-Powered Analytics	Limited	Limited	✗	✗ (3-phase)	✓
Real-time Chat	✓	✓	✗	✓	✓
Multi-Role RBAC	✓	✓	Basic	Basic (5 roles)	✓
Cost	Very High	Very High	Free/Low	Medium	Open Source
Target	Hospitals	Hospitals	Small Clinics	Medium Practices	Clinics

Table 1: Comparison Summary

2.3 Remote Patient Monitoring Technologies

2.3.1 Fitbit Web API

The Fitbit REST API is highly beneficial since it allows applications to communicate with a user's health data after accessing it using OAuth 2.0. This API offers data both on a daily summary basis and in time-series form throughout the day. Metrics like heart rate, steps taken by the user, quality of sleep, SpO2, HRV, and respiration rate are all securely stored. Should a user need extremely accurate physical information, one can get their heart rate every minute using Fitbit.

2.3.2 Apple Healthkit

The Apple HealthKit is where users can store their health and fitness information on their iOS-based mobile phones. These data are safely gathered and then distributed to any application that receives consent from the user.

2.3.3 Google Health Connect

Google Health Connect (formerly Google Fit) provides an API for accessing health data from Android devices and connected wearables. It supports a wide range of data types but has undergone significant API changes and lacks the stability of Fitbit's established web API.

2.4 AI in Healthcare

2.4.1 Rule-Based Clinical Decision Support Systems

The Classical CDSS is founded on simple "if-then" rules that generate warnings if vital signs cross certain limits. It might be used to send alerts if the heart rate goes beyond 100 beats per minute (bpm) or falls below 60 bpm (tachycardia and bradycardia correspondingly). Such systems are straightforward and easy to comprehend. On the contrary, they ignore the overall situation of each patient.

2.4.2 Statistical and Machine Learning Approaches

By switching from simple rules to advanced statistical procedures, one can identify subtle trends and anomalies in the patient's data. Linear regression methods are utilized for trend detection while anomalies are discovered via z-scores calculation. Moreover, there are particular methods developed for analyzing time series.

2.4.3 LLM-Based Clinical Narrative Generation

With the advent of novel advances in large language models (LLMs), there have emerged new possibilities to generate clinically relevant narratives based on raw patient data using artificial intelligence. LLMs, such as Llama 3.3-70b, incorporate all patient-specific data regarding their physiological patterns, medical history, lifestyles, medications, etc. On this basis, the algorithm generates accurate predictions and gives useful suggestions. In this case, one of the main challenges is to design adequate prompts to avoid hallucinations and guarantee the accuracy of medical information.

2.5 Telehealth & Video Consultation

As part of a response to the ongoing COVID-19 pandemic across the globe, governments have accelerated their adaptation to the telehealth systems significantly faster than anticipated. The contemporary telehealth applications demand a variety of audiovisual communication capabilities, including secure and fast connections between the provider and patient, scheduling appointments, checking the patient's medical record, and prescribing medications. In terms of its distributed nature, WebRTC has become the main technology for integrating remote video consultations into web browsers.The use of PeerJS significantly simplifies the usage of WebRTC because less work is done in preparing the technology for use.

2.6 Gaps in existing Solutions

After analyzing the advanced technologies that are currently available on the market, it was found out that our product should solve these existing weaknesses:

- **Fragmentation:** All the programs that are currently offered do not have a comprehensive functionality, since the specialization of some software may be in one area, such as EHR, scheduling system, telehealth system. It means that all these systems need to be combined into one.

- **No Real-time Data Analysis from Wearable Devices:** The software for clinics does not allow collecting the necessary data on other devices, for example, FitBit. It prevents continuous monitoring by patients.

- **No Complex AI Analysis:** Simple alarm notification systems are commonly used now. However, there is no such AI that could analyze the information using rule-based systems, statistics, and LLMs.

- **Enterprise Versus Cheap Solution:** While there are great solutions on the market, such as Epic and Cerner, they are rather expensive and cannot be used in small clinics. There are cheap alternatives, but their qualities are far from perfect.

Chapter 3

System Requirements

This chapter presents the system requirements for HealthSync, including use case diagrams,functional requirements, non-functional requirements, interface requirements, database requirements, project feasibility, and analysis models.

3.1 Use Case Diagram

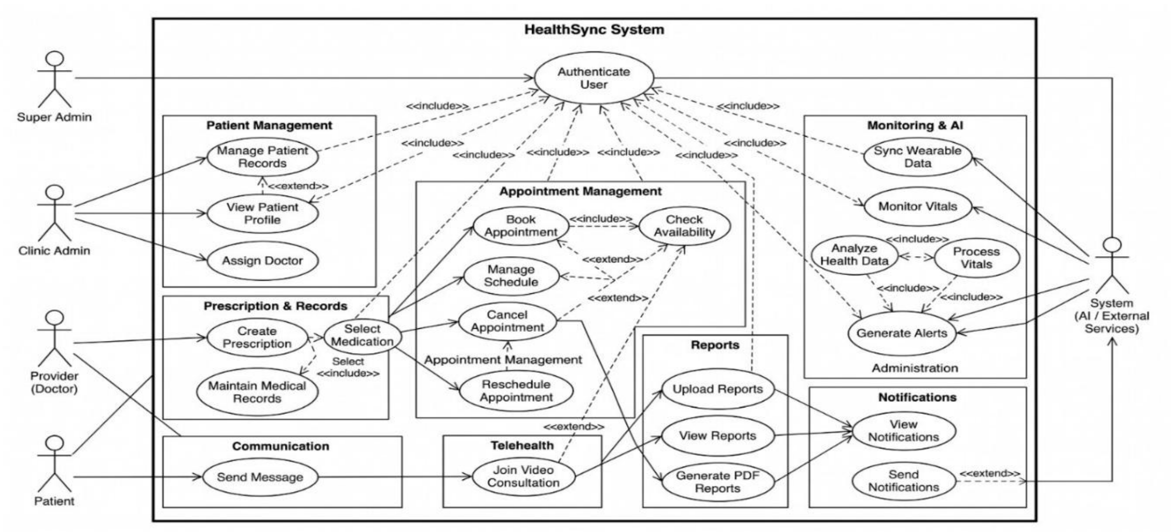


Figure 1 Use Case Diagram

3.2 Functional Requirements

3.2.1 FR-01: Authentication and Authorization

Use Case ID	**UC1**
Use Case Name	Authentication and Authorization
Actor(s)	Super Admin, Clinic Admin, Provider, Patient
Pre-Conditions	User must have registered account or invitation link
Priority	High
Basic Flow	System authenticates users using JWT and assigns role-based access with dashboard redirection
Actor Actions	**System Response**
1. User enters email and password	2. System validates credentials
3. User submits login	4. System generates JWT session
5. System identifies role	6. System applies RBAC permissions
7. User logs in	8. System redirects to dashboard
9. Admin sends invitation	10. System generates secure token link
11. Doctor registers	12. System auto-verifies doctor
Alternative Course	
3.a Invalid credentials	3.a.1 System shows error
9.a Invalid/expired token	9.a.1 System rejects request

Table 2: Authentication & Authorization Use Case

3.2.2 FR-02: Patient Management

Use Case ID	**UC2**
Use Case Name	Patient Management
Actor(s)	Admin, Doctor
Pre-Conditions	User must be authenticated
Priority	High

Basic Flow	System manages patient registration, profile, and doctor assignment
Actor Actions	**System Response**
1. Open patient form	2. System displays form
3. Enter patient details	4. System generates MRN
5. Add health profile	6. System stores profile
7. Link patient account	8. System links user account
9. Assign primary doctor	10. System updates mapping
11. View profile completeness	12. System shows percentage
Alternative Course	
3.a Missing data	3.a.1 System prompts completion
7.a Invalid linking	7.a.1 System shows error

Table 3: Patient Management Use Case

3.2.3 FR-03: Appointment Management

Use Case ID	**UC3**
Use Case Name	Appointment Management
Actor(s)	Patient, Doctor, Admin
Pre-Conditions	User must be logged in and doctor availability configured
Priority	High
Basic Flow	System handles appointment booking, scheduling, and lifecycle
Actor Actions	**System Response**
1. Start booking wizard	2. System shows steps
3. Select doctor	4. System shows time slots
5. Choose date/time	6. System validates slot
7. Confirm booking	8. System creates appointment
9. Doctor updates status	10. System updates lifecycle

11.Request cancel/reschedule	12. System enforces policy
Alternative Course	
5.a Slot unavailable	5.a.1 Ask to select another
11.a Late cancellation	11.a.1 Apply restriction

Table 4: Appointment Management Use Case

3.2.4 FR-04: Prescription Management

Use Case ID	**UC4**
Use Case Name	Prescription Management
Actor(s)	Provider
Pre-Conditions	Provider must be authenticated and authorized
Priority	High
Basic Flow	System allows providers to create and manage digital prescriptions with multiple medications
Actor Actions	**System Response**
1. Provider opens prescription module	2. System displays prescription form
3. Provider searches medication	4. System fetches drug from database
5. Provider adds medication details	6. System records drug info (dose, route, etc.)
7. Provider adds multiple items	8. System stores all prescription items
9. Provider saves prescription	10. System sets status (DRAFT/ACTIVE)
11. Provider updates/discontinues	12. System updates lifecycle status
Alternative Course	
3.a Drug not found	3.a.1 System shows error
9.a Missing required fields	9.a.1 System prevents saving

Table 5: Prescription Management Use Case

3.2.5 FR-05: Medical Records

Use Case ID	**UC5**
Use Case Name	Medical Records
Actor(s)	Provider
Pre-Conditions	Patient must exist and consultation must be active
Priority	High
Basic Flow	System allows providers to create SOAP-based medical records linked to consultations
Actor Actions	**System Response**
1. Provider opens consultation	2. System loads patient record
3. Provider enters Subjective data	4. System stores patient symptoms
5. Provider enters Objective data	6. System stores exam results
7. Provider enters Assessment	8. System saves diagnosis
9. Provider enters Plan	10. System stores treatment plan
11. Provider saves record	12. System links record to consultation
Alternative Course	
3.a Incomplete SOAP fields	3.a.1 System prompts completion
11.a Save failure	11.a.1 System shows error

Table 6: Medical Records Use Case

3.2.6 FR-06: Chat and Messaging

Use Case ID	**UC6**
Use Case Name	Chat and Messaging
Actor(s)	Patient, Provider
Pre-Conditions	Users must be authenticated
Priority	Medium
Basic Flow	System enables real-time messaging between patients and providers with workflow tracking
Actor Actions	**System Response**
1. User opens chat	2. System loads conversation
3. User sends message	4. System delivers message in real-time
5. Provider reviews message	6. System updates status (IN_REVIEW)
7. Conversation assigned	8. System assigns doctor
9. Provider replies	10. System updates thread
11. Conversation resolved	12. System marks as RESOLVED
Alternative Course	
3.a Message not sent	3.a.1 System retries or shows error
7.aNodoctor available	7.a.1 System keeps in queue

Table 7: Chat and Messaging Use Case

3.2.7 FR-07: Video Consultation

Use Case ID	**UC7**
Use Case Name	Video Consultation
Actor(s)	Patient, Provider
Pre-Conditions	Appointment must be scheduled
Priority	High
Basic Flow	System enables WebRTC-based video consultation linked with appointments
Actor Actions	**System Response**
1. User joins consultation	2. System initializes video session
3. Patient and doctor connect	4. System establishes WebRTC connection
5. Participants interact	6. System maintains session
7. Provider takes notes	8. System stores session notes
9. Session ends	10. System updates status to COMPLETED
Alternative Course	
2.a Connection failure	2.a.1 System retries or shows error
3.a Participant disconnects	3.a.1 System logs event and allows rejoin

Table 8: Video Consultation Use Case

3.2.8 FR-08: Fitbit Integration and Remote Monitoring

Use Case ID	**UC8**
Use Case Name	Fitbit Integration and Remote Monitoring
Actor(s)	Patient
Pre-Conditions	Patient must connect Fitbit account
Priority	High
Basic Flow	System integrates Fitbit data using OAuth and syncs health metrics
Actor Actions	**System Response**

1. Patient connects Fitbit account	2. System initiates OAuth flow
3. User grants permission	4. System stores access token
5. System fetches daily data	6. System stores vitals data
7. System performs intraday sync	8. System updates real-time data
9. Token expires	10. System refreshes token automatically
Alternative Course	
3.a Permission denied	3.a.1 System cancels integration
9.a Token refresh fails	9.a.1 System asks user to reconnect

Table 9: Fitbit Integration and Remote Monitoring Use Case

3.2.9 FR-09: AI Heart Health Analysis

Use Case ID	**UC9**
Use Case Name	**AI Heart Health Analysis**
Actor(s)	System, Provider
Pre-Conditions	Patient vitals data must be available
Priority	High
Basic Flow	System analyzes heart data and generates AI-based insights and alerts
Actor Actions	**System Response**
1. System collects vitals data	2. System processes data
3. System calculates health score	4. System generates score (0–100)
5. System detects anomalies	6. System identifies cardiac events
7. AI generates prediction	8. System outputs risk and recommendations
9. Alert generated	10. System creates alert workflow
Alternative Course	
3.a Insufficient data	3.a.1 System skips analysis

7.a AI failure	7.a.1 System logs error

Table 10: AI Heart Health Analysis Use Case

3.2.10 FR-10: Reports and Documents

Use Case ID	**UC10**
Use Case Name	Reports and Documents
Actor(s)	Patient, Provider
Pre-Conditions	User must be authenticated
Priority	Medium
Basic Flow	System manages report uploads, viewing, and PDF generation
Actor Actions	**System Response**
1. User uploads report	2. System stores file
3. Provider views report	4. System retrieves document
5. Provider adds annotation	6. System saves notes
7. System generates PDF	8. System provides downloadable file
9. User sets visibility	10. System enforces access control
Alternative Course	
1.a Invalid file format	1.a.1 System rejects upload
7.a PDF generation fails	7.a.1 System shows error

Table 11: Reports and Documents Use Case

3.2.11 FR-11: Notifications

Use Case ID	**UC11**
Use Case Name	Notifications
Actor(s)	System, User
Pre-Conditions	User must be logged in
Priority	Medium

Basic Flow	System delivers notifications via in-app and email
Actor Actions	**System Response**
1. Event triggers notification	2. System generates notification
3. System sends notification	4. User receives notification
5. User opens notification	6. System marks as read
7. User marks all as read	8. System updates all statuses
Alternative Course	
3.a Email delivery fails	3.a.1 System retries sending
5.a Notification not opened	5.a.1 Remains unread

Table 12: Notifications Use Case

3.2.12 FR-12: Administration

Use Case ID	**UC12**
Use Case Name	Administration
Actor(s)	Super Admin, Clinic Admin
Pre-Conditions	Admin must be authenticated
Priority	High
Basic Flow	System allows admins to manage clinics, users, roles, and settings
Actor Actions	**System Response**
1. Admin creates clinic	2. System stores clinic data
3. Admin invites staff	4. System sends invitation
5. Admin assigns roles	6. System updates permissions
7. Admin manages schedules	8. System updates availability
9. Admin updates settings	10. System applies configurations
Alternative Course	
3.a Invalid email	3.a.1 System shows error
5.a Unauthorized role change	5.a.1 System blocks action

Table 13: Administration Use Case

3.3 Interface Requirements

3.3.1 User Interface Requirements

ID	Requirement
IR-U1	The system shall have adaptive web design which works perfectly fine in the case of the desktop, tablet, and mobile without any requirement of installation from application stores.
IR-U2	Every type of user role gets their unique dashboard where the custom menu is displayed on the sidebar based on their task or responsibility.
IR-U3	Users shall switch between dark and light mode based on their preferences inside the interface.
IR-U4	The digital form needs to validate the errors of the user in real-time while entering data and give feedback on it.
IR-U5	There must be a notification bell icon at the top navigation bar which tells the number of unread notifications of the users through badges.
IR-U6	Appointments are made by patients with the help of four steps wizard that tells users about their progress on each step.
IR-U7	Vital signs need to be shown by using charts inside the health monitoring dashboard with different time frame choices like an hour, twelve hours, twenty-four hours, and seven days.
IR-U8	While searching for medication, the Prescription module will auto complete with suggestions while allowing browsing through these by using the keyboard and pressing the Enter key.
IR-U9	It will check whether the user's profile is complete or not; new users will get a welcoming screen while those with less than 70 percent completion will have a continuous banner reminding them.
IR-U10	Each form that requires any kind of document upload will allow the user to perform a drag-and-drop operation along with instant notifications about their file types.

Table 14: User Interface Requirements

3.3.2 Physical Interface Requirements

ID	Requirement
IR-P1	The ID requirements are that the system must be integrated with the Fitbit wearable device through the Fitbit Web API on an HTTPS connection.
IR-P2	The ID requirements are that the OAuth 2.0 Login process must redirect users from the Fitbit app to a predefined callback URL.
IR-P3	The system should obtain the user's health data using the Fitbit REST APIs in the finest granularity possible, even up to the minute level.

Table 15: Physical Interface Requirements

3.3.4 Software / Component Interface Requirements

ID	Requirement
IR-S1	In communicating with the backend, the frontend will employ RESTful Next.js API Routes only over HTTPS.
IR-S2	In order to facilitate session management, secure HTTP-only cookies will be employed for access and refresh tokens.
IR-S3	PeerJS JavaScript library will be employed in managing WebRTC media streams for live video chats
IR-S4	File uploads and downloads such as Medical Reports, PDF documents, user profile pictures will be performed via Supabase storage and vercel blob storage.
IR-S5	AI will create Health reports after connecting with the Cerebras API (Llama 3.3-70B).
IR-S6	SMTP services will perform email notifications after setting up the environment variables.
IR-S7	AI health checks, cleaning, and reminders may be performed via Vercel Cron Jobs.
IR-S8	Prisma ORM will facilitate all database activities while the application cannot make raw SQL queries.

Table 16: Software / Component Interface Requirement

3.4 Database Requirements

3.4.1 Database Requirements

ID	Requirement
DR-1	The application shall make use of PostgreSQL as the principal relational database for storing data.
DR-2	All actions related to databases shall be done through the Prisma ORM, and all migrations should be carried out according to the schema
DR-3	To isolate patient data from being accessed illegally, each table with patient data shall have a clinicId attribute
DR-4	To guarantee data integrity is maintained in the system, foreign keys shall be used to keep a connection between tables and delete data by cascading in some cases
DR-5	To facilitate quick performance of the application, indexes will be created on patientId, clinicId, status, and date attributes.
DR-6	All sensitive data such as passwords will be stored in hashed form only.

Table 17: Database Requirements

3.4.2 Core Data Entities

Table	Purpose
User	Stores all user accounts, credentials, profile picture, and completeness score
Role / UserRole	Defines roles and maps users to roles with optional clinic scope
AuthEffectiveGrant	Stores resolved permission grants per user per clinic for RBAC enforcement
Clinic	Represents a registered clinic; root scope for all clinical data
DoctorProfile	Stores professional details, specialization, verification status per doctor
PatientRecord	Master patient identity including demographics, MRN, conditions, allergies

Consultation	Links doctor-patient encounters to a clinic
MedicalRecord	Stores SOAP-based medical history entries
Appointment	Stores booking records, status, and scheduling details
DoctorAvailability	Weekly recurring schedule per doctor
DoctorSlot	Individual time slots with availability tracking
Prescription / PrescriptionItem	Stores prescriptions and medication details

Table 18: Core Data Entities

3.4.3 Data Retention & Constraints

ID	Requirement
DR-7	Intraday vital sign data shall be retained for a minimum of 90 days
DR-8	Daily vital sign snapshots shall be retained for a minimum of 2 years
DR-9	Patient MRN shall be unique within each clinic enforced via (clinicId, mrn) constraint
DR-10	User invite tokens shall have an expiry timestamp and be invalidated upon consumption
DR-11	Deleted parent records shall cascade-delete all associated child records
DR-12	PDF and medical report files shall be stored externally; only references stored in DB

Table 19: Data Retention & Constraints

3.5 Non-Functional Requirements

NFR ID	Category	Requirement
NFR-01	Security	JWT tokens with HS256 signing, 2-hour access token expiry, 7-day refresh token expiry. Password hashing using bcryptjs. Middleware-enforced API authentication with public route whitelist. Clinic-scoped data isolation preventing cross-clinic access.
NFR-02	Performance	Database connection pooling (up to 20 concurrent connections). Permission cache with 120-second TTL to reduce database round trips. Indexed database queries on critical access paths.
NFR-03	Usability	The system has user-friendly dashboards designed for easy navigations. Also, there is a simple 4-step wizard to schedule appointments and autocomplete feature in searching medications.
NFR-04	Reliability	The system provides reliability because of its auto reliability that will automate daily analysis of health data, weekly clean-ups, and reminders daily. The tokens from Fitbit will get renewed automatically to prevent sync issues. In addition, a retry mechanism will be used in case of syncing errors.
NFR-05	Scalability	The system uses serverless infrastructure from Vercel and includes automatic scalability. Prisma ORM, together with connection pooling, is used to enable scalability of the database. The application architecture is clinic-centric, hence making the system capable of handling multiple clinics concurrently.
NFR-06	Maintainability	The system is developed using the TypeScript language in strict mode so that there are no type errors within the codebase. The system follows a modularity approach for separating concerns.

Table 20: Non Functional Requirements

3.6 Project Feasibility

3.6.1 Technical Feasibility

It can be said with certainty that this application is feasible from a technical point of view. The following technologies can be mentioned in particular: Next.js 15, React 19 which offer functionalities such as SSR (Server Side Rendering), App Router, Server Actions. It was decided to use PostgreSQL as a database since it is known to be a reliable open source database which works well with JSON and Prisma ORM. The source of health data would be the Fitbit REST API along with OAuth 2.0. To ensure instant analysis of patients' health state the Cerebras Llama 3.3-70b Inference was chosen. Finally, the use of PeerJS would allow using WebRTC much easier.

3.6.2 Operational Feasibility

This application has been developed for small and medium sized clinics using some manual paperwork or other means of digital processing. The system's role-based design ensures that each user type encounters only relevant features, reducing training overhead. The invitation-based onboarding workflow ensures controlled user access

3.6.3 Legal & Ethical Feasibility

• The system is designed with HIPAA-awareness including encrypted tokens, access
logging (audit trails), role-based data access, and clinic-scoped isolation.
• Patient consent is obtained for Fitbit data access through the OAuth 2.0 flow.
• AI-generated recommendations include explicit disclaimers and are designed to support (not replace) clinical judgment.
• Data rights and privacy policies are available on specific pages of the platform, including /data rights, /privacy, and /terms).

3.7 Analysis Models

3.7.1 Context-Level Data Flow Diagram (Level 0)

The Level 0 DFD illustrates HealthSync as one central system that connects with all external entities.

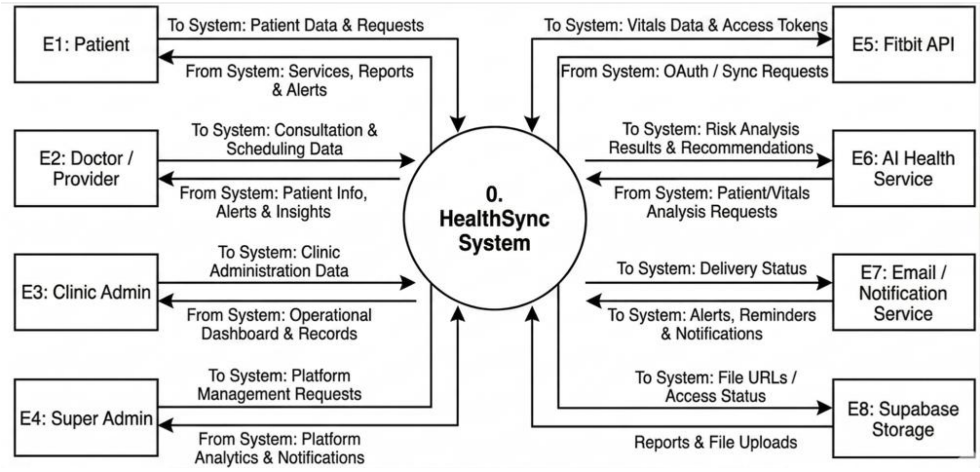


Figure 2: Context-Level Data Flow Diagram (Level 0)

3.7.2 Level 1 Data Flow Diagram

Decomposes the system into 12 major subsystem processes and 12 data stores.

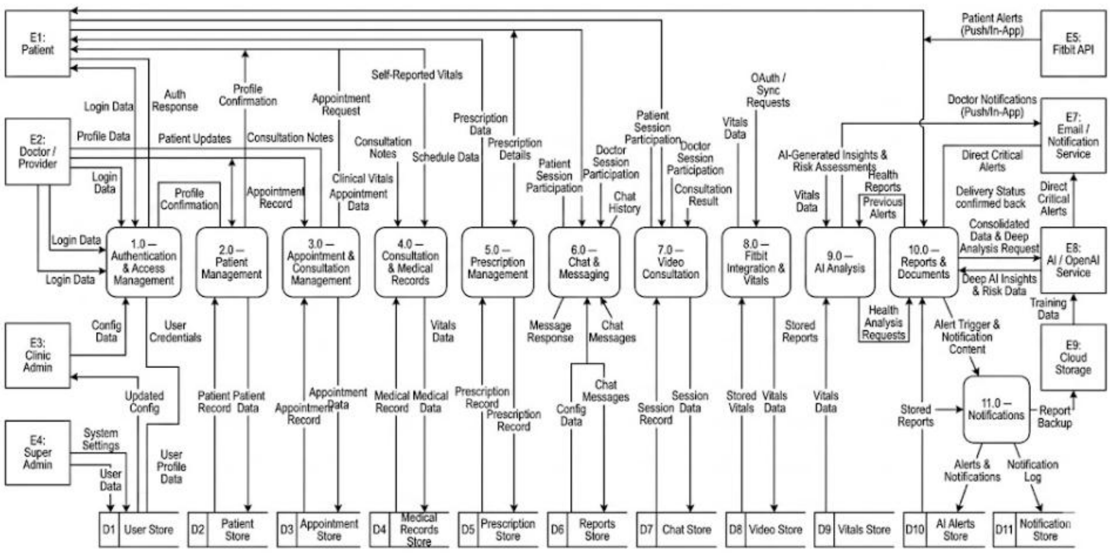


Figure 3: Level 1 Data Flow Diagram

3.7.3 Appointment Booking Flow (Activity Diagram)

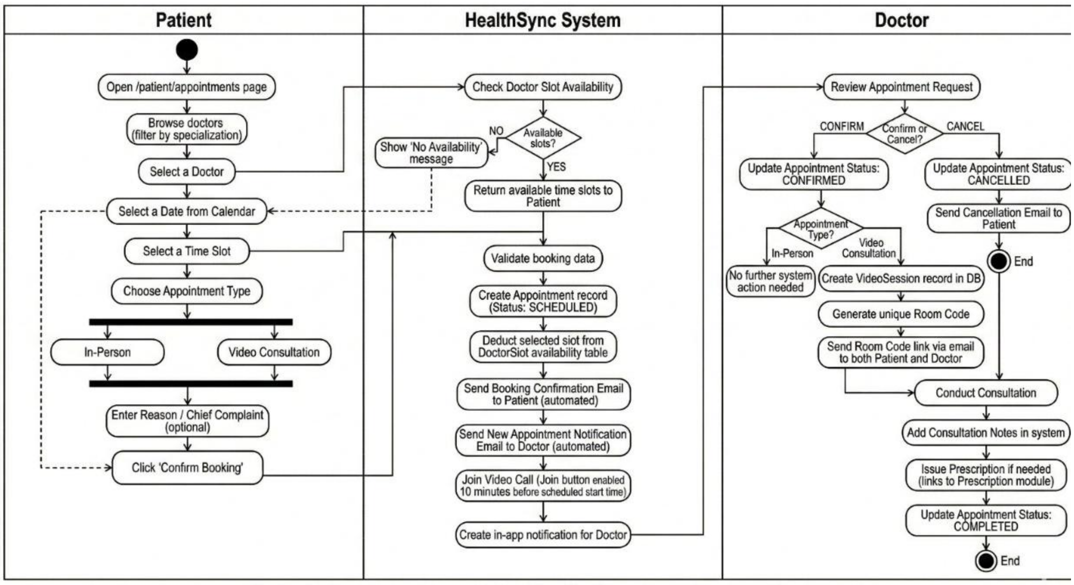


Figure 4: Appointment Booking Flow (Activity Diagram)

3.7.4 AI Health Analysis Pipeline (Activity Diagram)

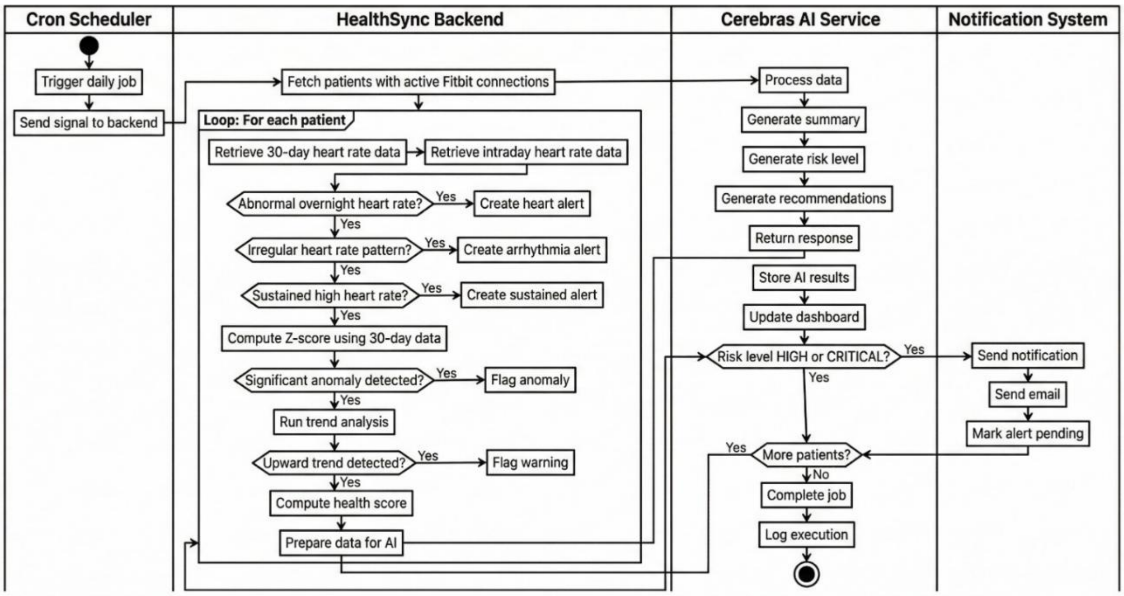


Figure 5: AI Health Analysis Pipeline (Activity Diagram)

3.7.5 User Authentication and Role based Access (Sequence diagram)

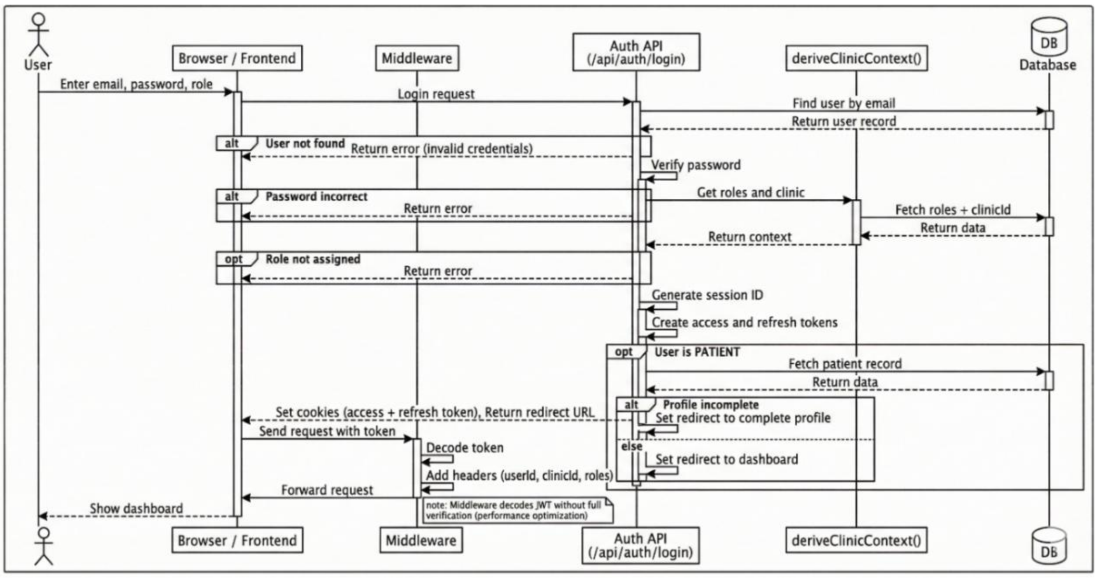


Figure 6: User Authentication and Role based Access (Sequence diagram)

3.7.6 Fitbit OAuth2 Connection and Daily Vitals Sync (Sequence Diagram)

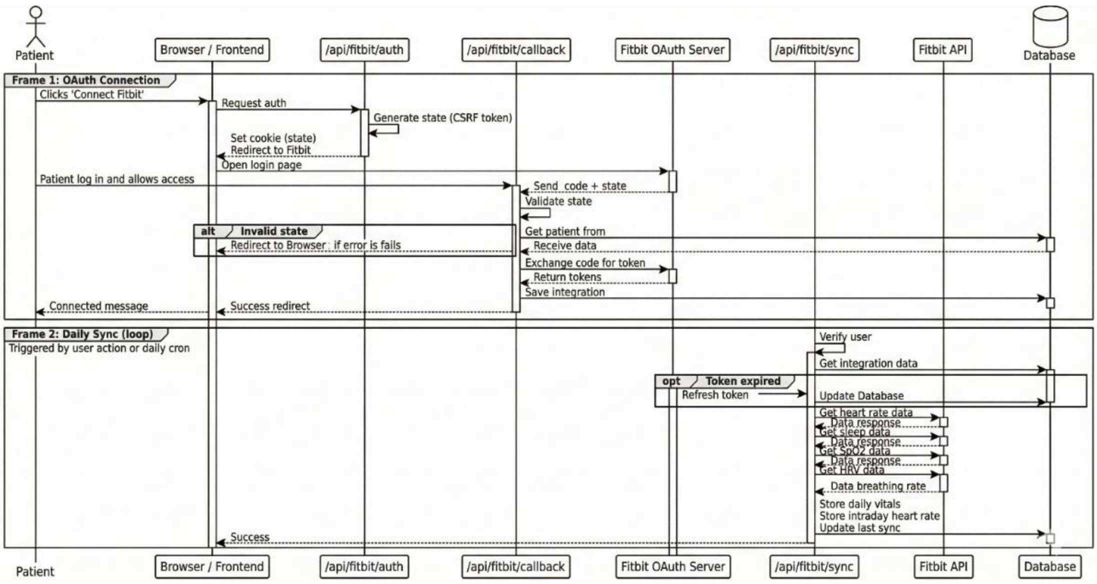


Figure 7: Fitbit OAuth2 Connection and Daily Vitals Sync (Sequence Diagram)

3.8 Conclusion

This part is dedicated to the detailed description of functionality that is expected to be implemented in HealthSync starting from the core functionalities to database requirements. Functional requirements refer to the basic set of functionalities of the software including such aspects as user login, patients' data, scheduling appointments, writing prescriptions, real-time chat, and AI health check. Concerning non-functional requirements, then HealthSync will be reliable, scalable, fast and easy to maintain. Furthermore, the interface requirements will help create connections between the software and its users and third-party software like Fitbit wearable devices and AI applications. Finally, in terms of the data layer, data storage security and scalability will be provided with the use of the developed database architecture based on PostgreSQL and Prisma ORM. According to the feasibility study conducted, it can be stated that the project is technologically, legally and operationally feasible due to the use of up-to-date technologies and compliance with all the existing regulations.

Also, we should not forget about the analysis models created by us, which include data flow diagrams and sequence diagrams.

Chapter 4
System Design

The design of the HealthSync application has been covered in this chapter and covers the following components such as design approach, architecture, logical design, dynamic view, component design, data modeling, and user interface design.

4.1 Design Approach

Design approach considerations are taken into account while creating HealthSync application:

- **Component-Based Architecture:**
  React Component design has been done with an emphasis on separation of concern for better maintenance and reuse.

- **Server Side Rendering (SSR)**
  The Server Side Rendering (SSR) approach was implemented using Next.js App Router to enhance SEO capabilities and quick page loading times.

- **API-First Approach**
  All the functionalities were covered under the RESTful API Routes approach.

- **Schema First Approach for Database Design**
  A schema first approach for database design is used to create databases through Prisma.

- **Role-Based Segregation of Functionalities**
  Different routes, layouts, and API routes were developed based on the role through the role-based segregation approach.

- **Modular AI Pipeline:**
  The design approach of modular AI Pipeline has been used which includes rule-based, statistical, and LLM-based designs.

4.2 Design Constraints

There are going to be several technical problems that we will have to overcome while designing our application:

• Restrictions on Fitbit API Request:

Our API requests per hour for each user cannot exceed 150 requests. That is why it is imperative that we design a suitable caching and synchronizing mechanism.

• **Constraints with Respect to Function Execution within Vercel's Server:**

There will be a timeout if the execution of the function exceeds 15 seconds. Further, the size of the body of the request should not exceed 15MB.

• **Incompatibility of WebRTC Technology:**

Since our video calling feature uses the WebRTC technology, it is necessary that we use up-to-date browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

• Need for Third-Party Services:

As our algorithm depends on some third-party services, there can be a possibility that the AI algorithm might not function correctly because of network problems.

• **Need for Suitable Node.js Versions:**

In order to become fully compliant with Next.js 15, it is imperative to use Node.js versions from 18 to 22.

4.3 System Architecture

We are planning to employ the traditional three-tier architecture provided by Vercel as each tier performs a specific purpose

4.3.1 Presentation Tier (Frontend)

- **Framework:** Next.js 15 App Router with React 19
- **Styling:** Tailwind CSS 4 with Radix UI components
- **State Management:** React Context API with custom hooks

- **Data Visualization:** Recharts for vital signs and analytics
- **Video Communication:** PeerJS for WebRTC-based consultations
- **Icons:** Lucide React icon library

4.3.2 Application Tier (Backend)

- **Runtime:** Next.js API Routes (serverless functions on Vercel)
- **Authentication:** JWT (HS256) with middleware enforcement
- **Authorization:** Role-Based Access Control (RBAC) with permission caching (120-second TTL)
- **Email Services:** Nodemailer with Gmail SMTP and connection pooling
- **File Storage:** Vercel Blob and Supabase Storage
- **AI Engine:**

    1. Rule-Based Module
    2. Statistical Analysis Module
    3. LLM-Based Module (Cerebras)

- **Scheduling:** Vercel Cron Jobs for automated background tasks

4.3.3 Data Tier (Database)

- **Database Management System:** PostgreSQL
- **ORM:** Prisma (v6.17.1)
- **Schema Design:**

    1. 28+ data models
    2. 25+ enums
    3. 50+ indexes

- **Connection Handling:** Database connection pooling (maximum 20 concurrent connections)

4.4 Logical Design

4.4.1 Class Diagram

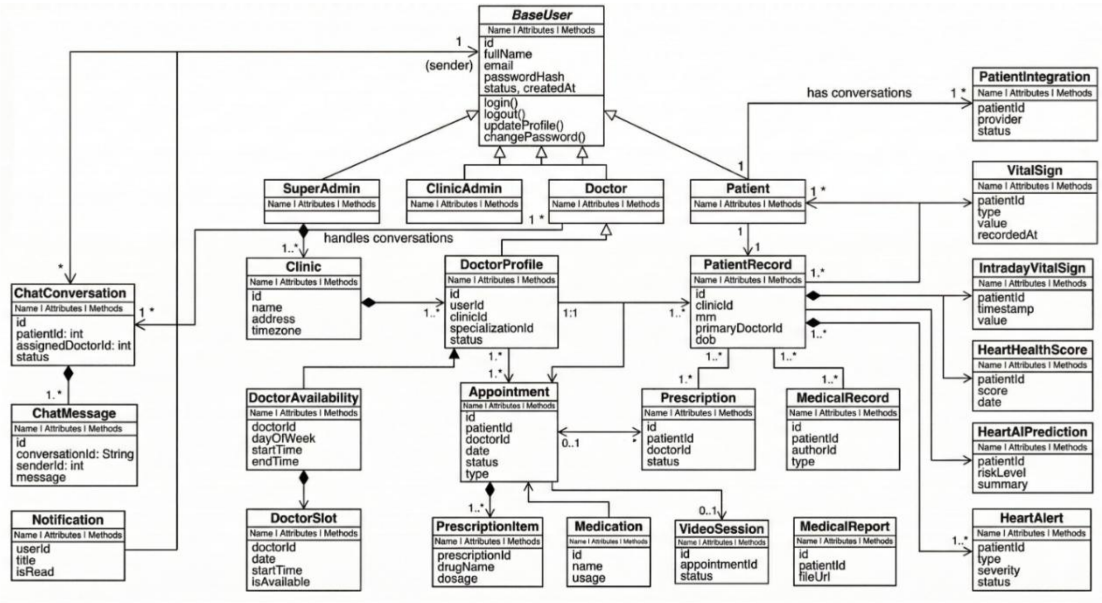


Figure 8: Class Diagram

4.5 Dynamic View

4.5.1 Sequence Diagrams

4.5.1.1 Sequence Diagram JWT Token Refresh Flow

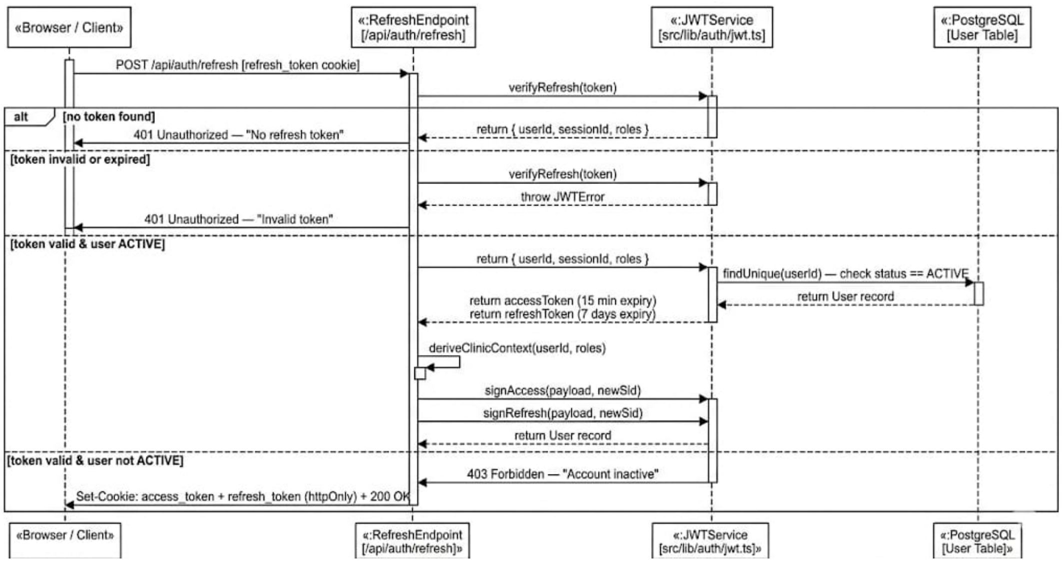


Figure 9: Sequence Diagram JWT Token Refresh Flow

4.5.1.2 Sequence Diagram Secure Messaging Flow

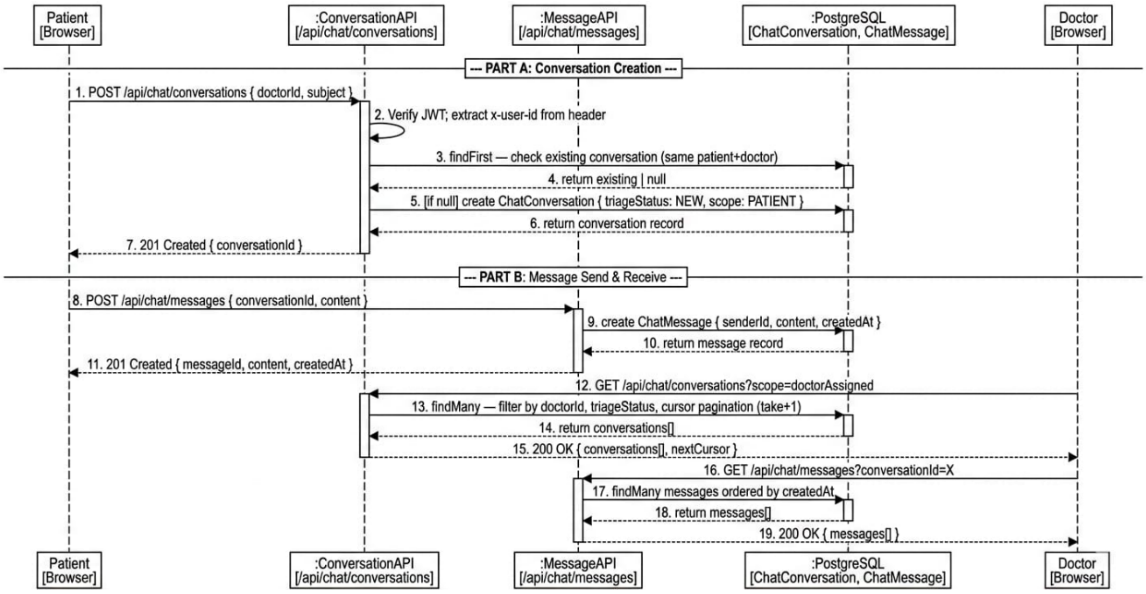


Figure 10: Sequence Diagram Secure Messaging Flow

4.5.1.3 Clinic Admin Manage Appointments

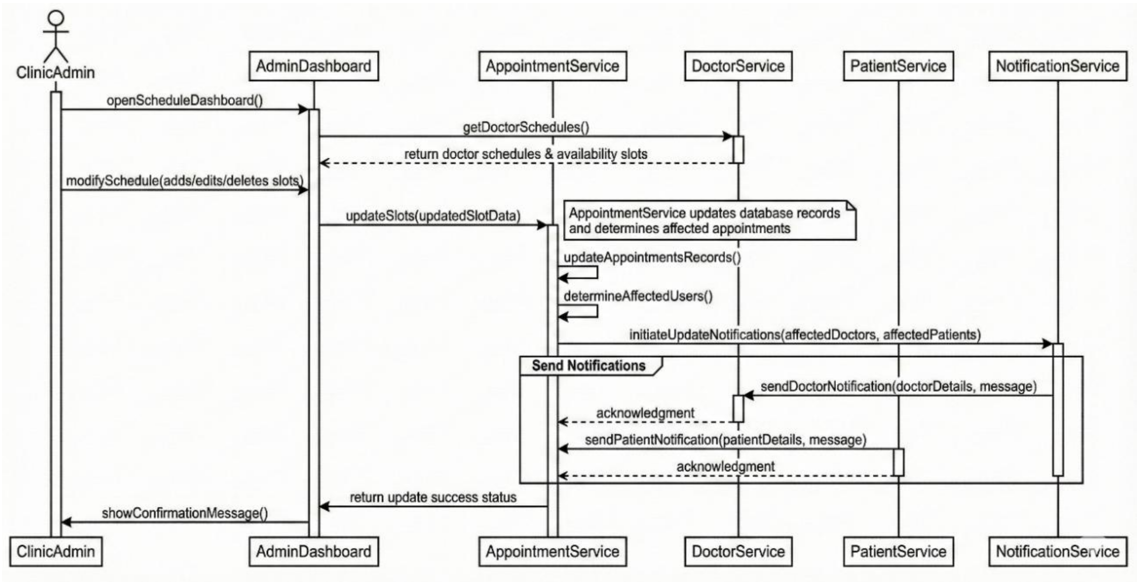


Figure 11: Clinic Admin Manage Appointments

4.5.1.4 Doctor E-Prescription Creation & Signing Flow

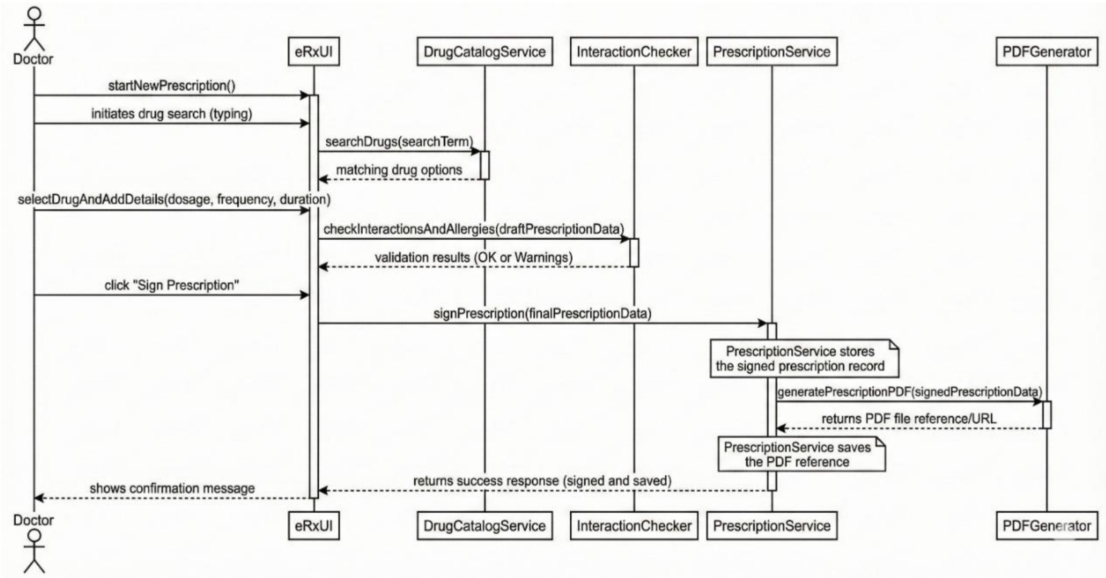


Figure 12:Doctor E-Prescription Creation & Signing Flow

4.5.1.5 Doctor RPM Alert Review & Acknowledgement Flow

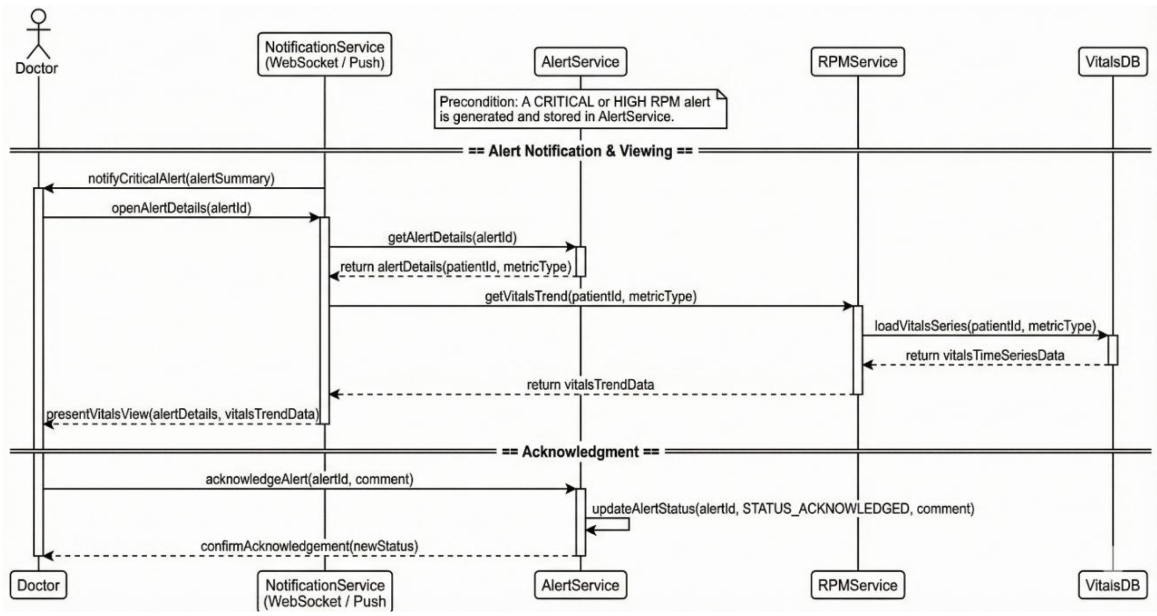


Figure 13: Doctor RPM Alert Review & Acknowledgement Flow

4.5.1.6 Patient Device Connect & Vitals Sync Flow

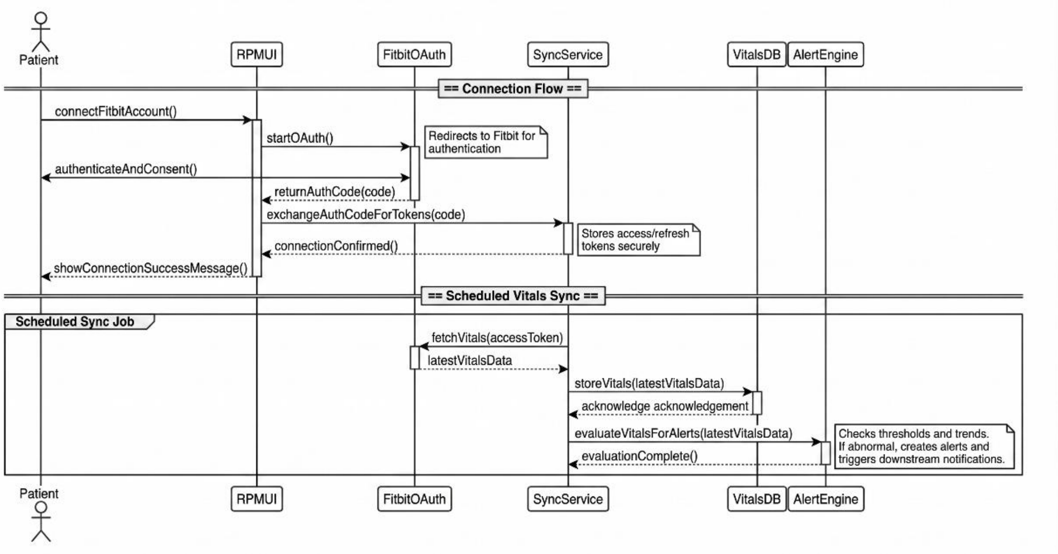


Figure 14: Patient Device Connect & Vitals Sync Flow

4.5.2 Activity Diagrams

4.5.2.1 Activity Diagram Medical Report Upload

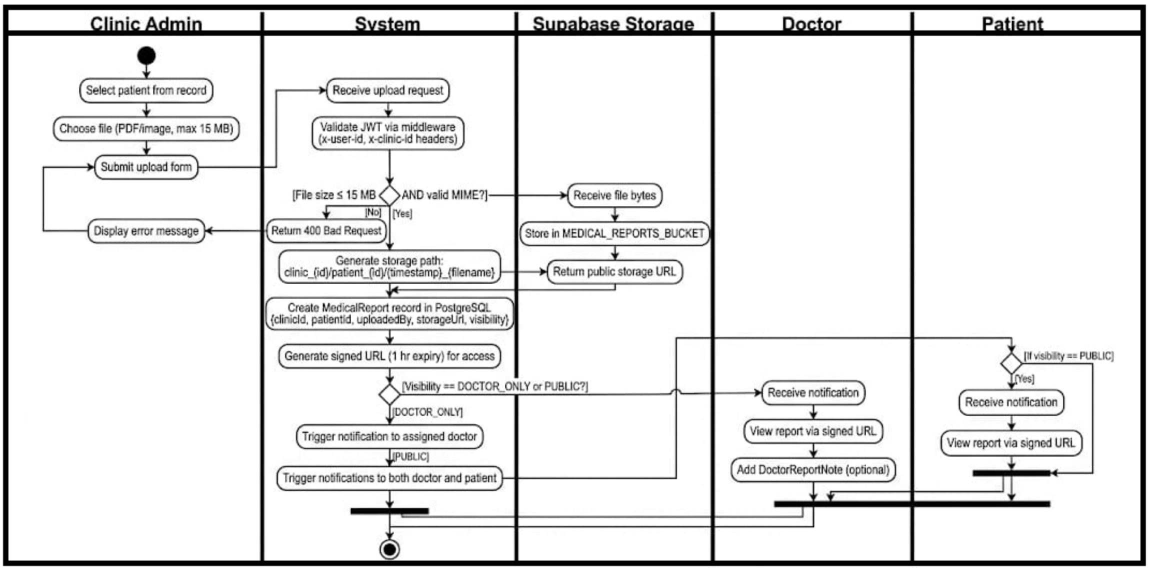


Figure 15: Activity Diagram Medical Report Upload

4.5.2.2 Activity Diagram Chat Conversation Triage

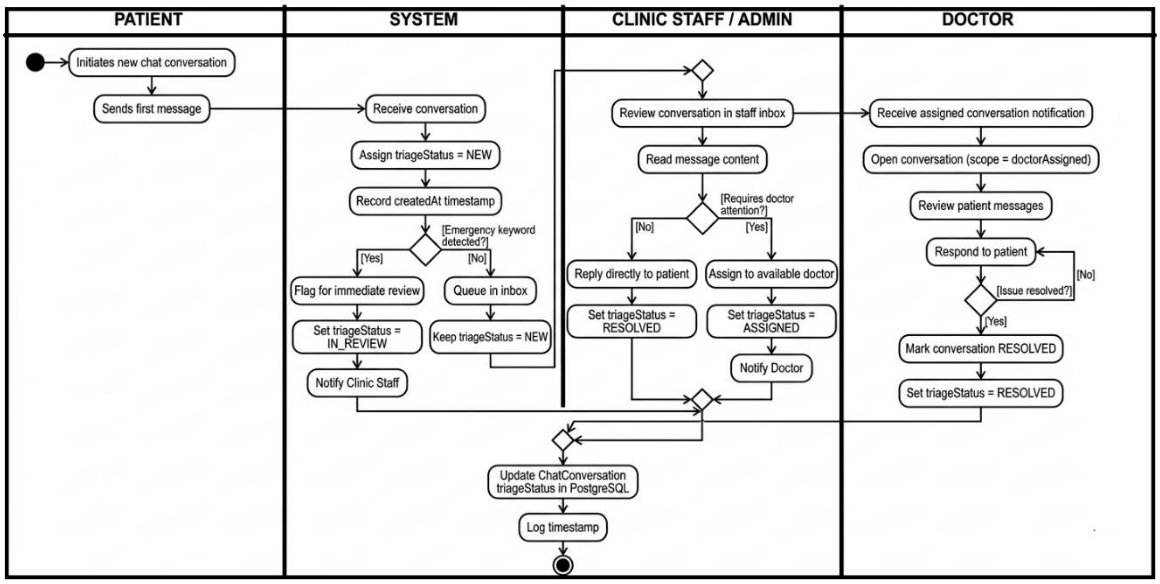


Figure 16: Activity Diagram Chat Conversation Triage

4.5.2.3 Activity Diagram Clinic Admin (Manage Appointments)

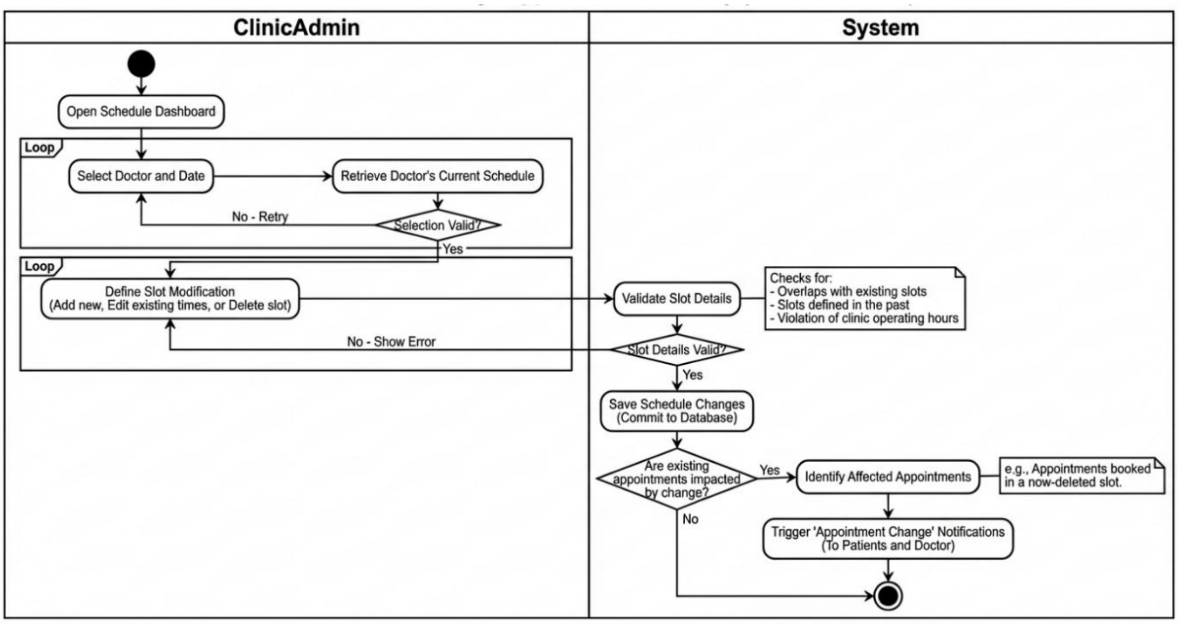


Figure 17: Activity Diagram Clinic Admin (Manage Appointments)

4.5.2.4 Activity Diagram Doctor (Conduct Telehealth Consultation)

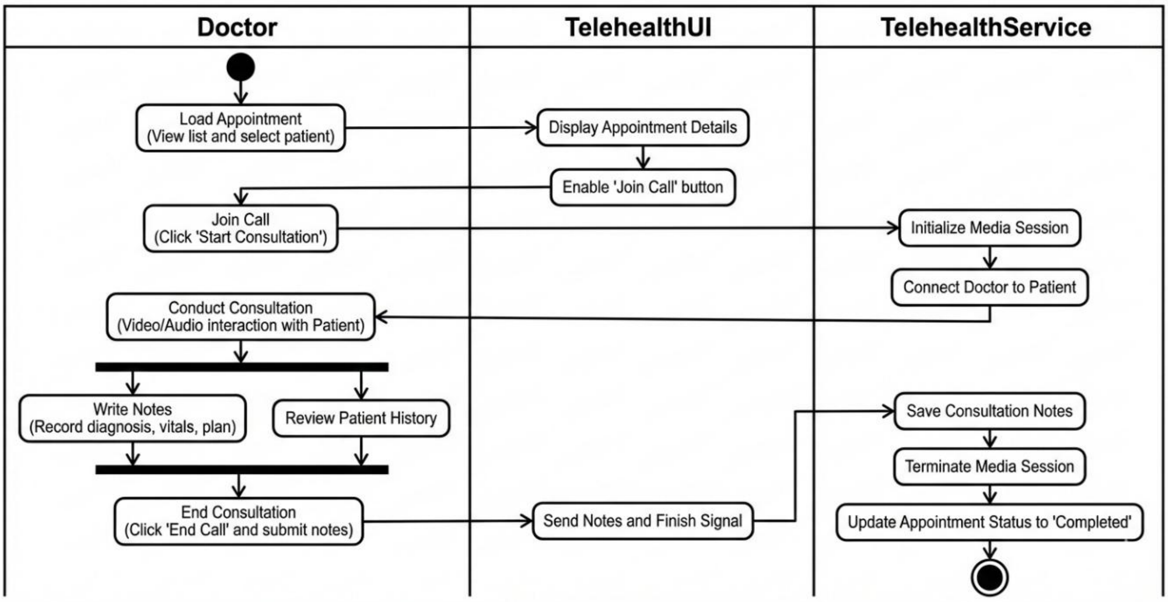


Figure 18: Activity Diagram Doctor (Conduct Telehealth Consultation)

4.5.2.5 Activity Diagram Doctor Create & Sign Prescription

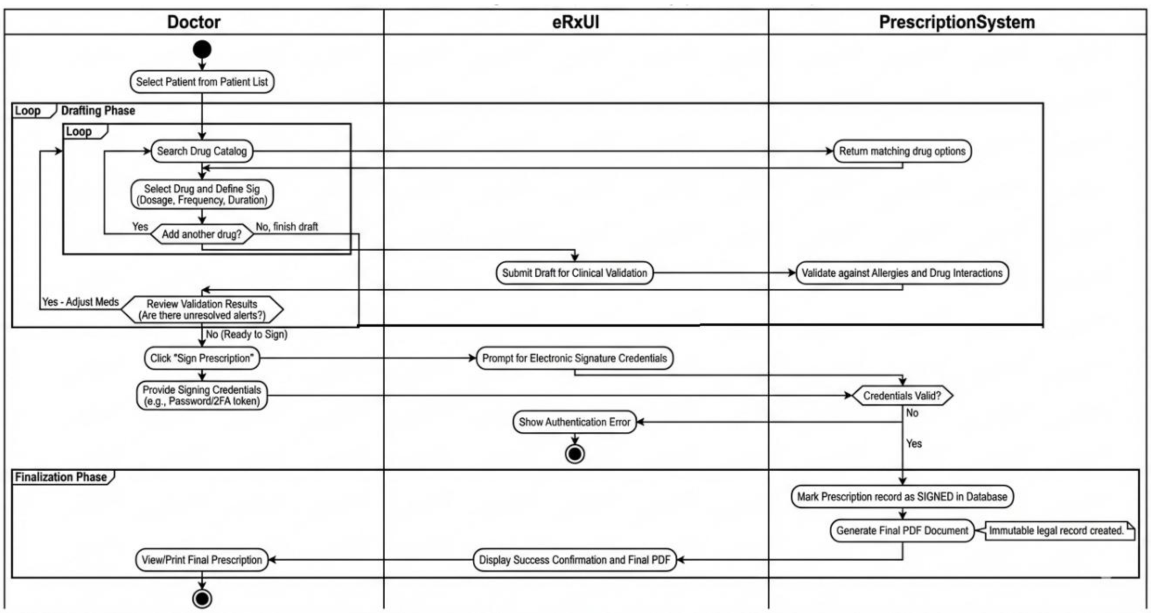


Figure 19: Activity Diagram Doctor Create & Sign Prescription

4.5.2.6 Activity Diagram Patient Device Connect & Sync Vitals Activity

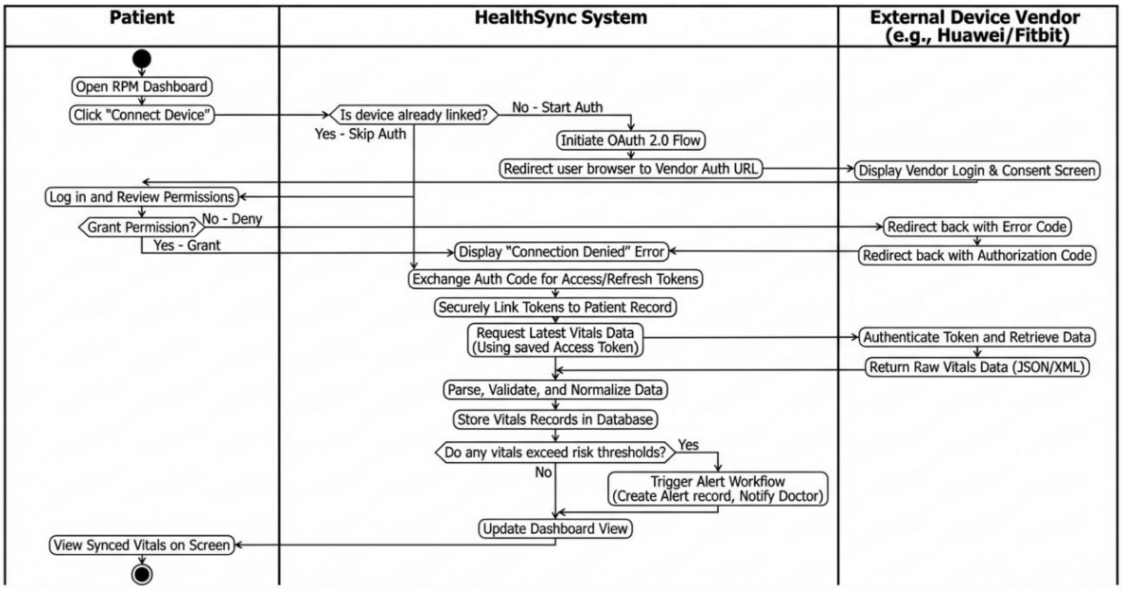


Figure 20: Activity Diagram Patient Device Connect & Sync Vitals Activity

Component Design

4.6.1 Package Diagram

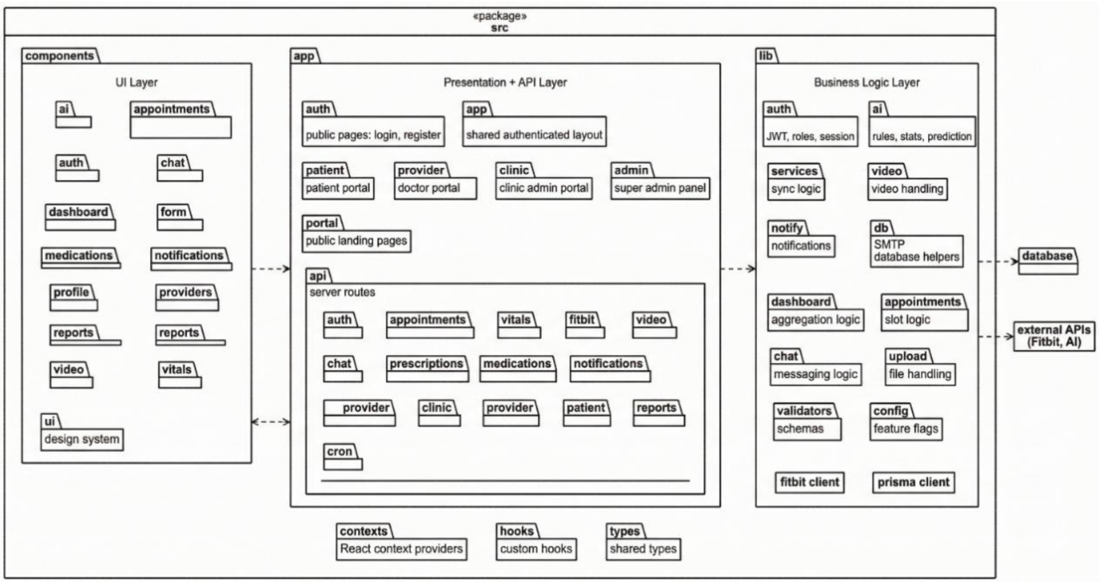


Figure 21: Package Diagram

4.6.2 Component Diagram

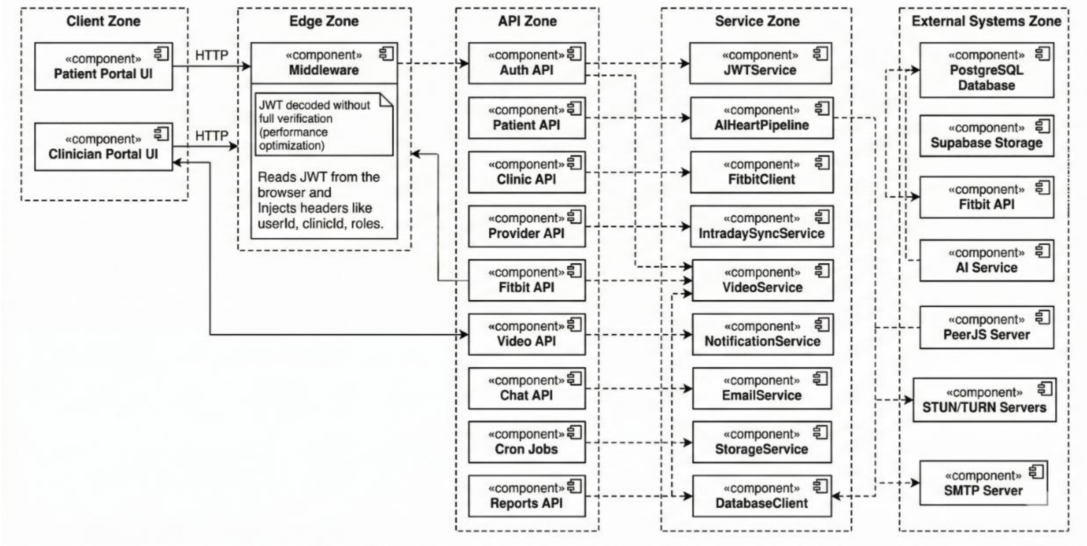


Figure 22: Component Diagram

4.6.3 Deployment Diagram

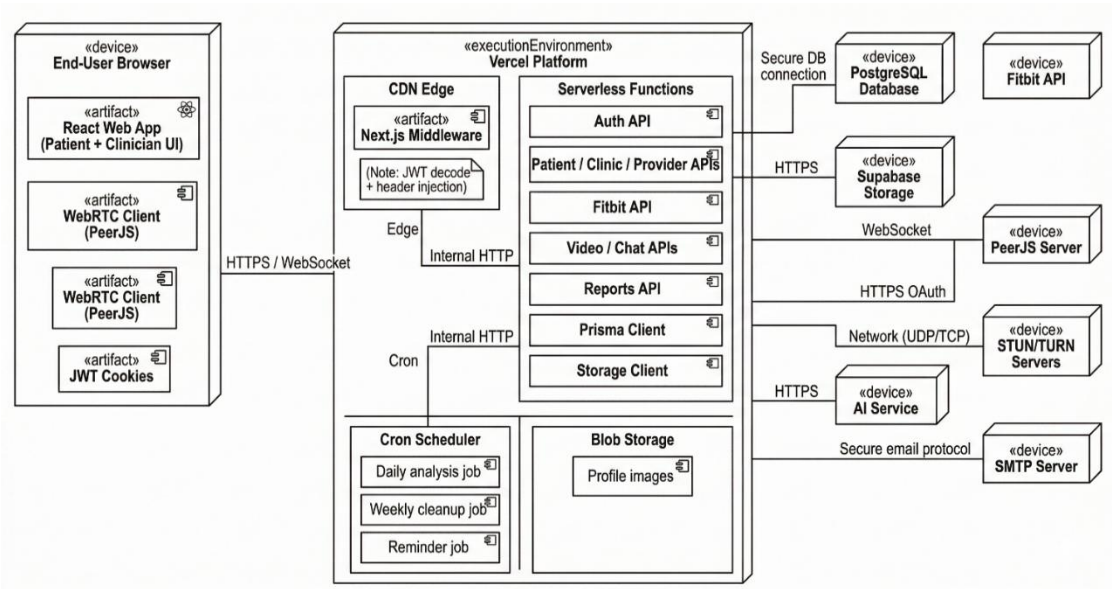


Figure 23: Deployment Diagram

4.7 Data Models

4.7.1. ERD Diagram

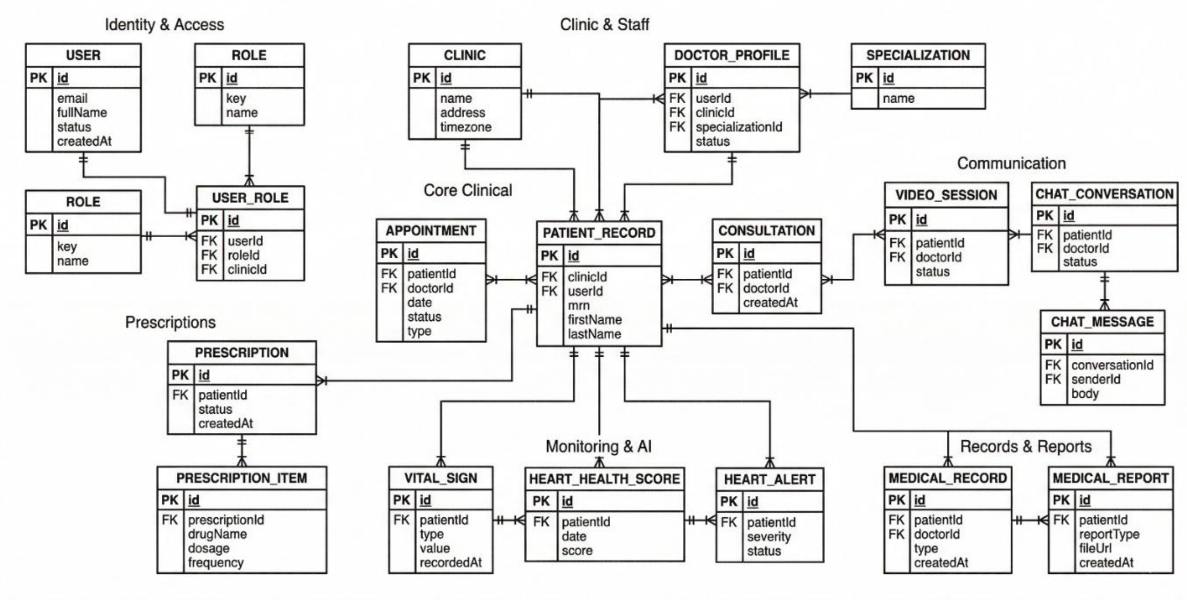


Figure 24: ERD Diagram

4.8 User Interface Design

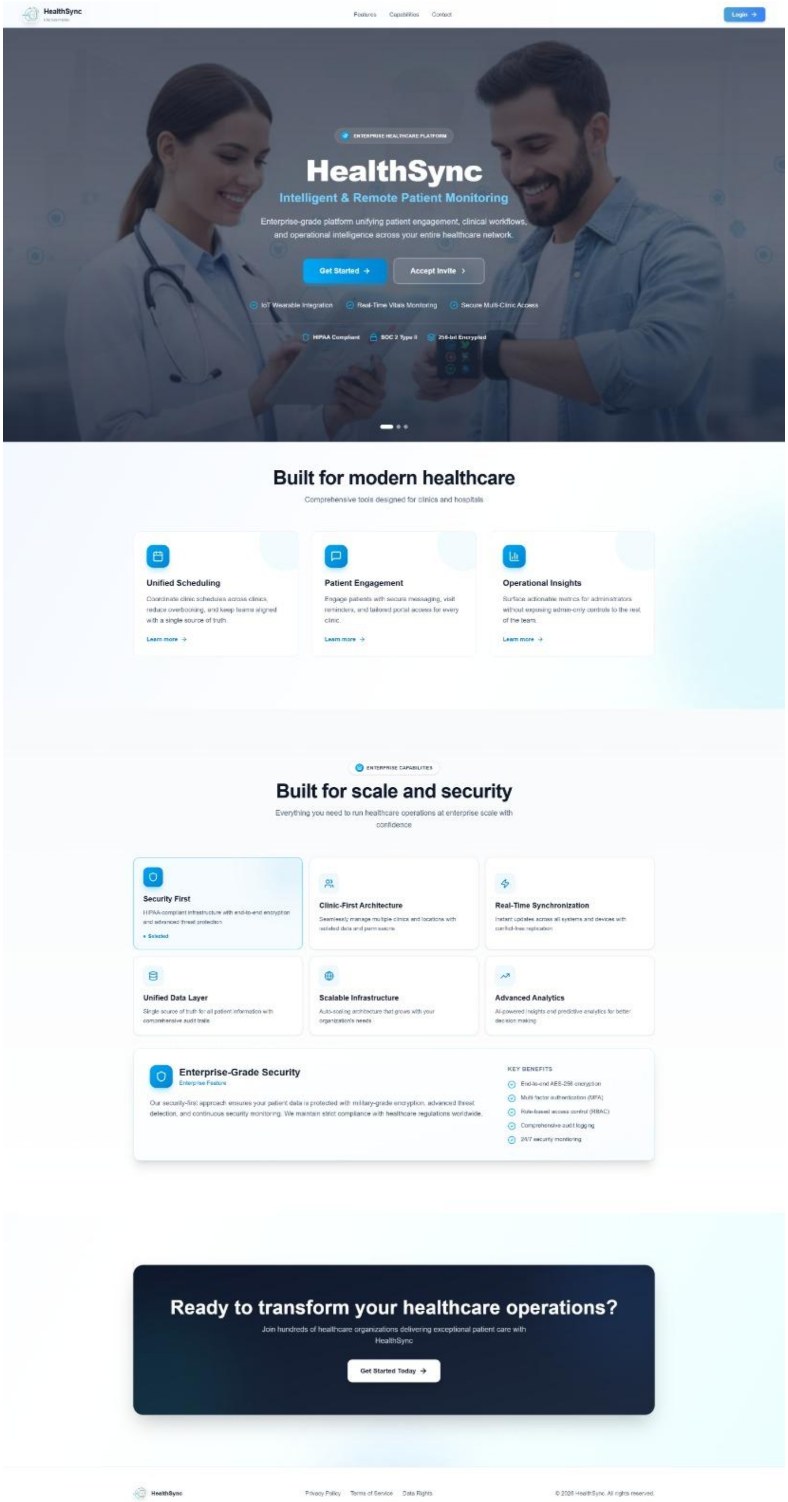


Figure 25: Homepage of the HealthSync

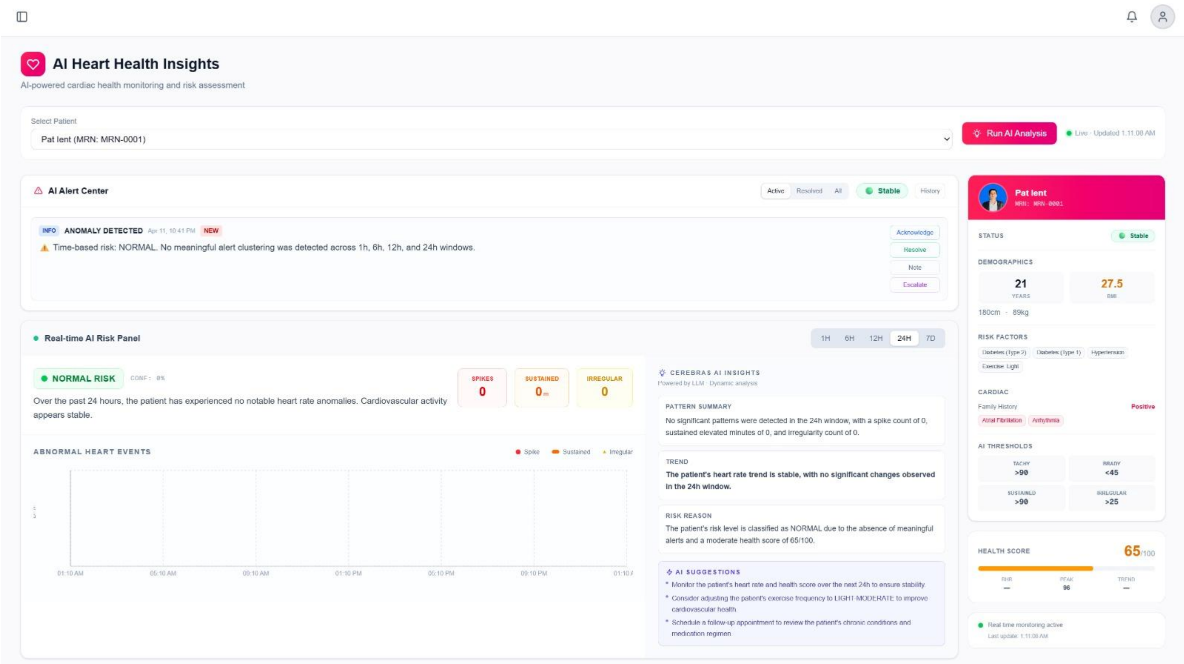


Figure 26: AI Insights Page in the Doctor Dashboard

Figure 27: Monitoring Page in the Doctor Dashboard

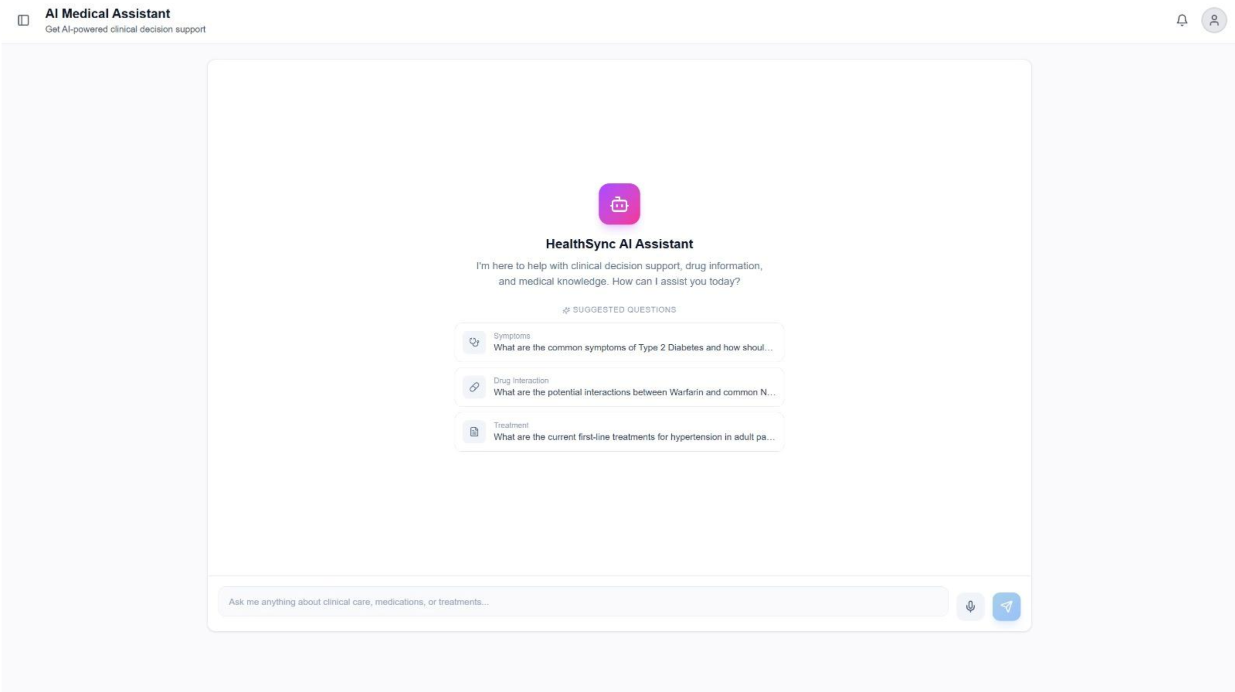


Figure 28: AI Assistant Page in the Doctor Dashboard

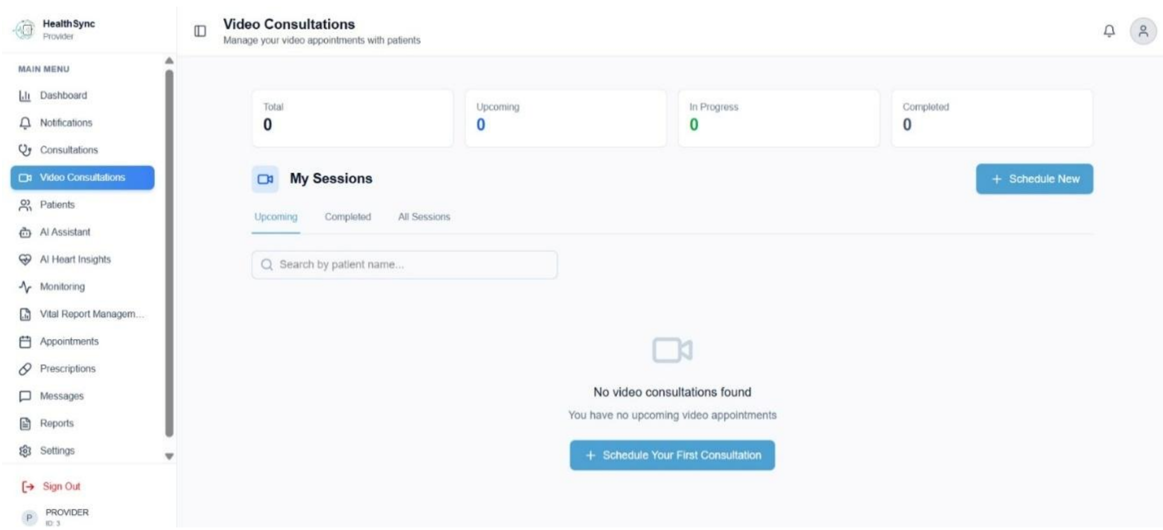


Figure 29: Video Consultation Page in the Doctor Dashboard

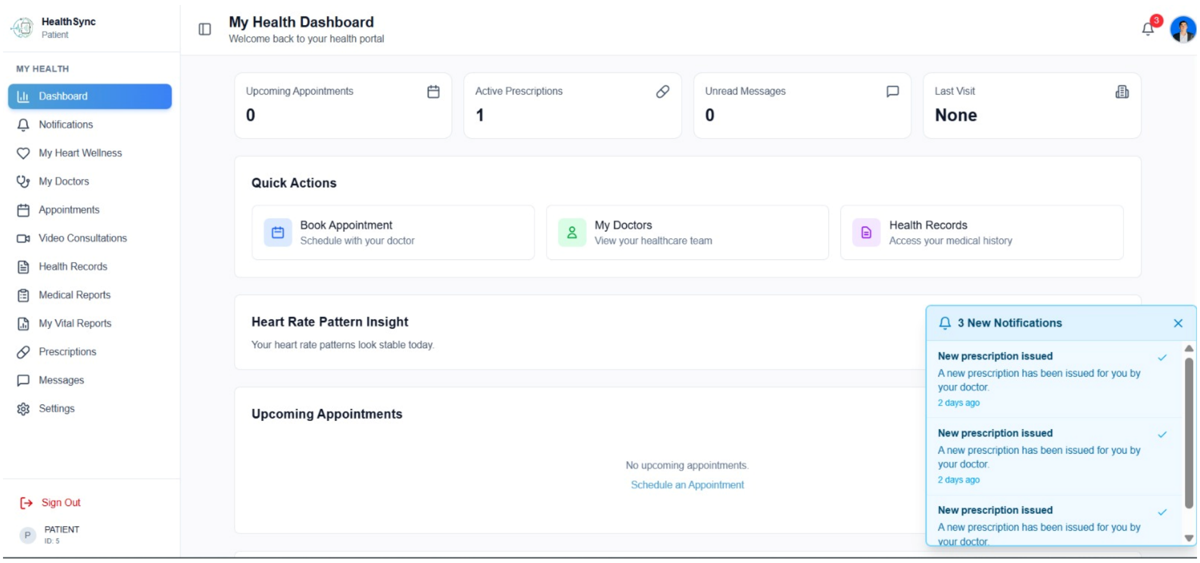


Figure 30: Patient Dashboard Page

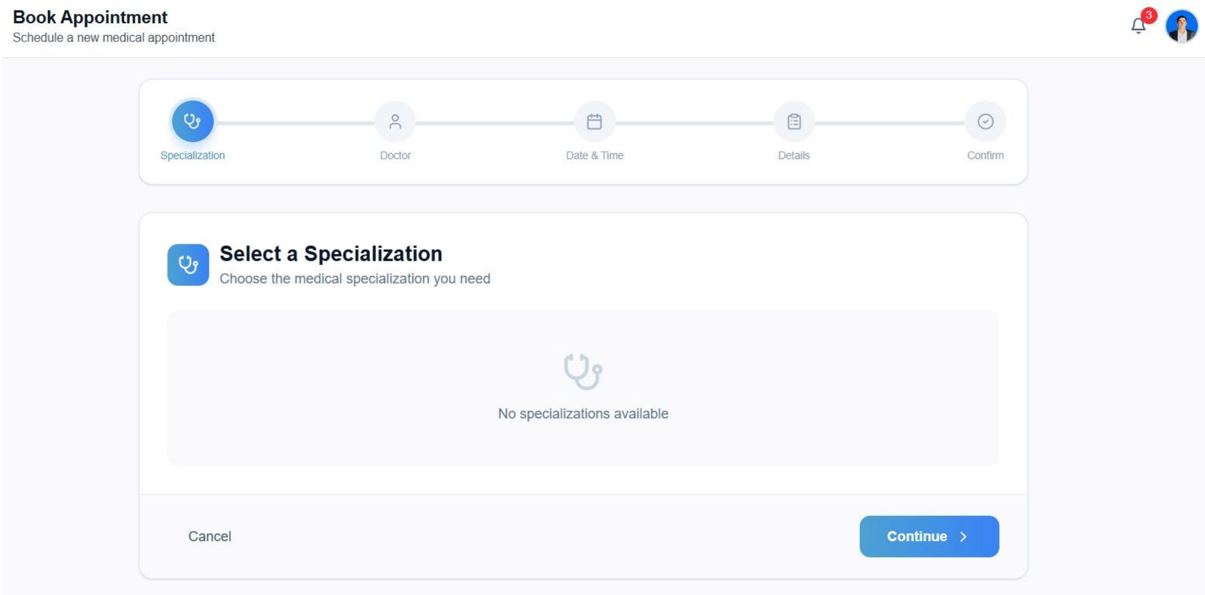


Figure 31: Appointment Booking Page in the Patient Dashboard

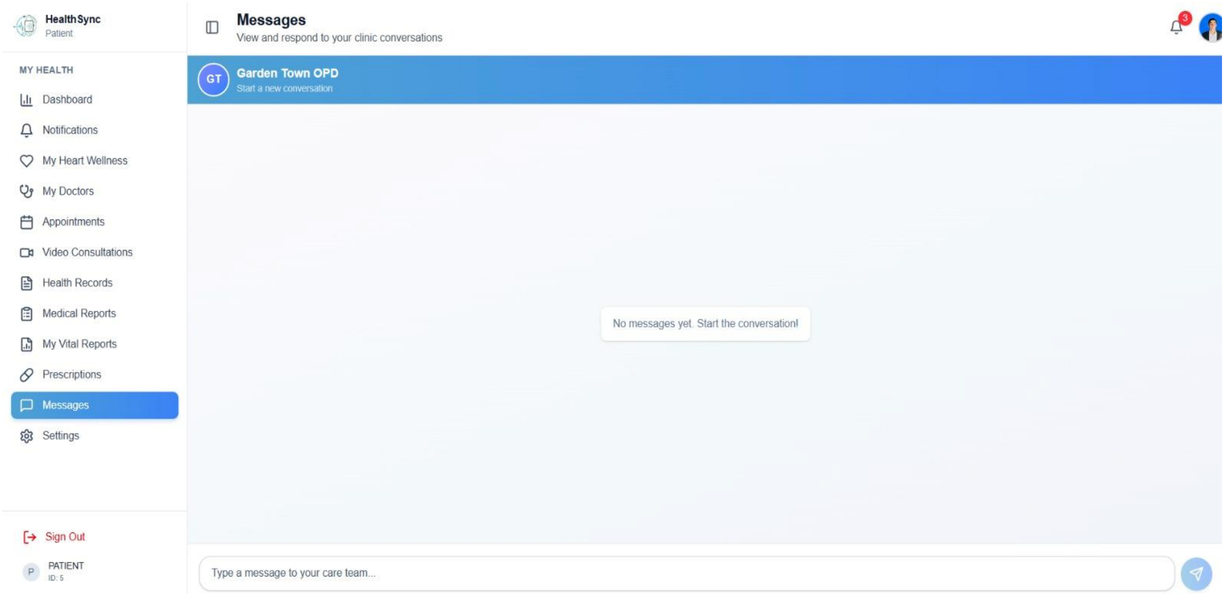


Figure 32: Messaging Page in the Patient Dashboard

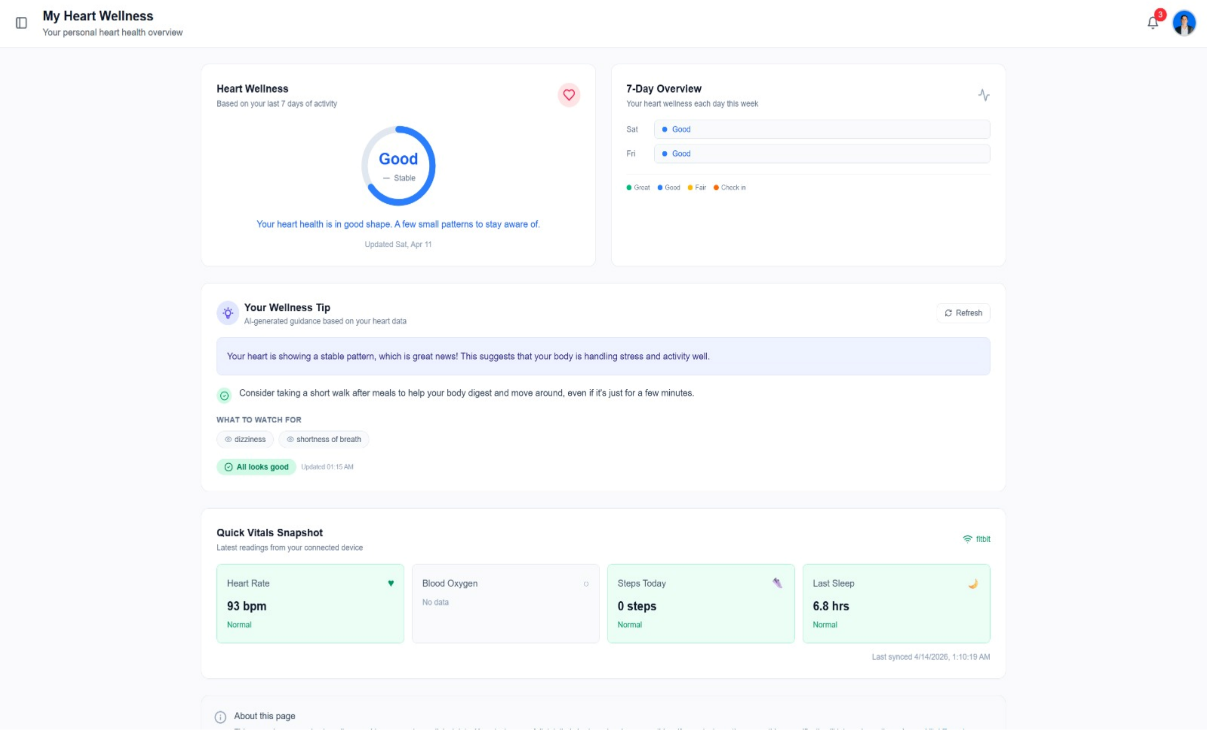


Figure 33: Heart Wellness Monitoring Page in the Patient Dashboard

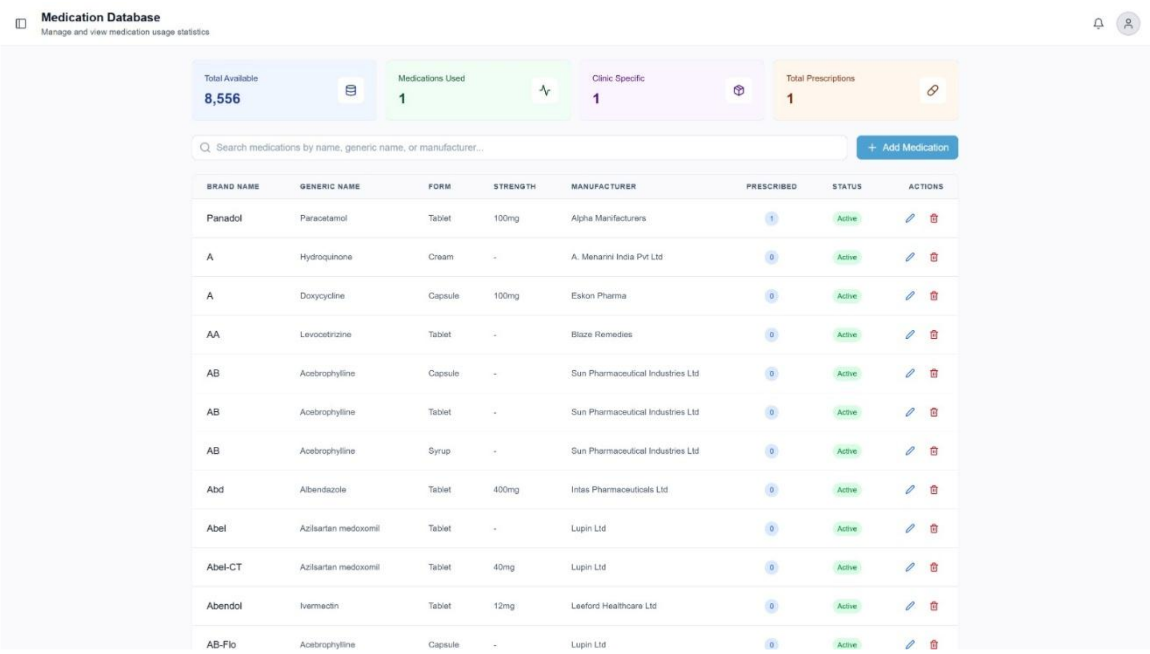


Figure 34: Medication Management Page in the Clinic Dashboard

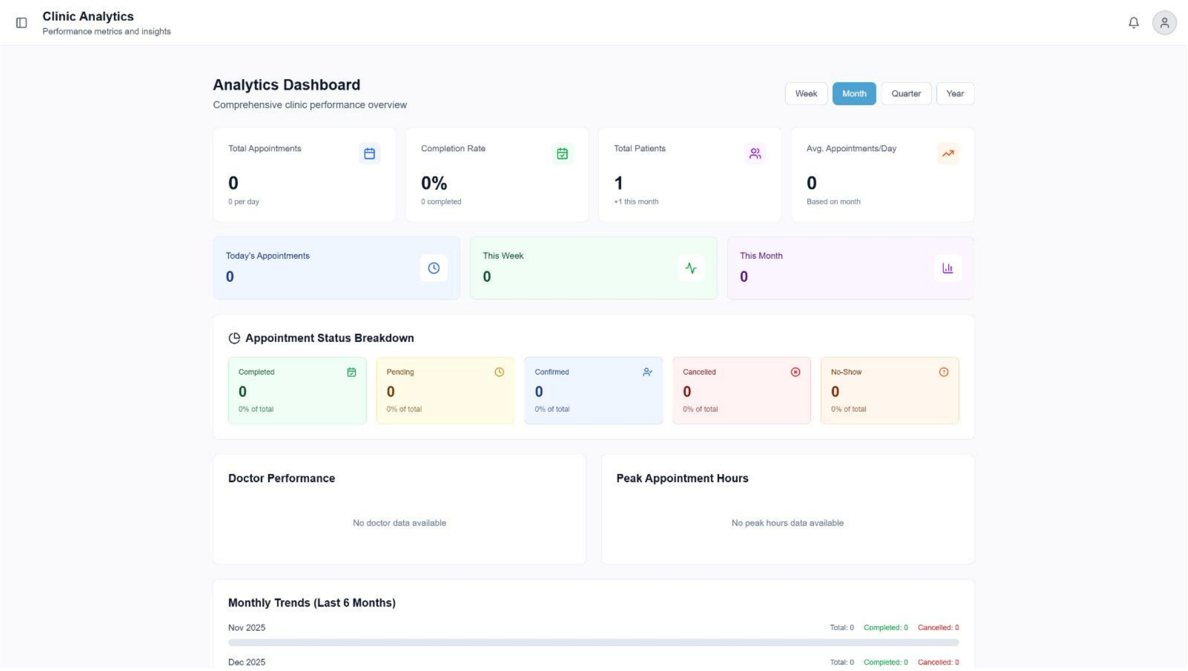


Figure 35: Clinical Analytics Page in the Super Admin Dashboard

4.9 System Prototype

HealthSync app is a fully functional web app built based on Next.js, PostgreSQL database, and actual API access. We have an application that uses a multi-tenant portal where the user has their own portal if he or she is a patient, provider, clinic admin, or super admin. We used Tailwind CSS in developing the front end part of our application, wherein this framework includes component, Lucide icons, and Recharts for charts and graphs. Moreover, the methods utilized for logging in are via JSON Web Token and cookie-based mechanism. Other registration methods include email invitation and password reset. Lastly, no prior design was done before moving to the development phase.

Patient Portal lets the users log into the dashboard, book appointments, keep tabs on the medication, observe their heart, send messages, view test results, and participate in video consultation sessions. The Provider Portal lets the physicians manage their patients, manage their schedule, analyze data using AI-driven analytics, keep track of health parameters, prescribe medication, and perform teleconsultations. There is total isolation of the administration accounts, where the Clinic Administrators take care of normal clinical activities such as booking appointments, physicians' management, and personal data of the patients, whereas the Super Administrator account is responsible for overall management of the application. Users of the application are provided with the same and unified interface with navigation, responsiveness, real notifications, and clear confirmations throughout the application. Features such as real-time notifications, live chat functionality, and video conferencing powered by WebRTC technology makes the application extremely easy to use. The application uses an AI engine which suggests potential health threats and makes recommendations for clinicians.

4.10 Conclusion

Firstly, this chapter has elaborated upon all the aspects involved in the implementation of HealthSync architecture. For instance, we have employed an object-oriented design pattern using Next.js and Prisma in order to ensure modularity and responsibility of each module. Having considered the security policies along with the constraints pertaining to the scalability of the project and resource management, we have identified the best-fit technologies, namely, JWT-based authentication, Neon PostgreSQL-based database management, and PeerJS for connecting to the application. With a view to increasing the efficiency and scalability of our project, a three-tier serverless architecture consisting of the client layer, gateway layer, and application layer has been designed. In addition to this, the logical architecture of the system has been provided by means of ER diagrams and development of a relational model comprising 26 tables.

Designing the structure of the database involved going through all the necessary stages, starting from conceptual up to physical one. To ensure that all data manipulations take place in real time, we had used PostgreSQL capabilities of working with JSONB, proper indexing, and keys. Lastly, we managed to demonstrate that everything will work out by making a working prototype of our project available on Vercel. Every particular interface, including Patient Dashboard, Provider Tools, Clinic Admin Management Interface, and even Super Admin Control Panel, has been developed for real-life performance. All the above design considerations fit our functional requirements perfectly well.

Chapter 5

System Implementation

This chapter covers the deployment of the HealthSync application. The deployment, the technology used, and other specific details of the development process for the application's significant functions shall be elaborated in this section.

5.1 Technology Stack

Component	Technology	Version
Framework	Next.js (App Router with Turbopack)	15.5.7
UI Library	React	19.1.0
Language	TypeScript (strict mode)	5.x
Styling	Tailwind CSS	4.x
Component Library	Radix UI (Dialog, Tabs, Avatar)	Latest
Icons	Lucide React	0.446.0
ORM	Prisma Client	6.17.1
Database	PostgreSQL	15+
Authentication	JWT (HS256, custom implementation)	—
Password Hashing	bcryptjs	2.4.3
Validation	Zod	3.23.8
Forms	react-hook-form	7.67.0
Charts	Recharts	3.7.0
Date Utilities	date-fns	4.1.0
WebRTC	PeerJS	1.5.5
Email	Nodemailer	7.0.10
PDF Generation	jsPDF + pdf-lib	4.2.0 / 1.17.1
File Storage	Vercel Blob + Supabase	2.0.0 / 2.87.1
AI/LLM	Cerebras API (Llama 3.3-70b)	—
Unit Testing	Jest + ts-jest	30.2.0
E2E Testing	Playwright	1.56.1
Linting	ESLint 9	9.x
Deployment	Vercel	—

Table 21: Technology Stack

5.2 Authentication And Authorization Implemntation

5.2.1 JWT Token System

We designed our custom-made JWT authentication system, without incorporating any third party tools like NextAuth for HealthSync. There are two tokens used in this system. First is the access-token that expires after 2 hours and secondly is the refresh token whose lifespan is 7 days. Access-token holds important user information such as userId, clinicId, roleKeys[], account status, and sessionId (sid). Refresh-token on the other hand stores only the userId and sid. Both access-token and refresh-token hold information in an HS256 signed format. To prevent XSS attacks, both the tokens are used solely in HTTP only cookies.

5.2.2 RBAC Permission System

A mechanism for controlling access has been introduced in our security model known as the Role-Based Access Control (RBAC). As per this security model, permissions have been allocated to every role individually. One can adjust the permission mechanism according to his/her own choice by granting or refusing Auth effective grant permissions based on the availability of ALLOWED or DENIED permissions. In situations where there is any discrepancy regarding permissions, permission of DENY prevails over permission of ALLOW. Permission verification has been implemented through catching within a period.

5.3 Appointment Management Implementation

The appointment manager module is implemented in such a way that the patient will just have to follow the steps in the form of a simple wizard to accomplish the task within four simple steps. In the first step, the patient is supposed to select the preferred doctor based on the name and specialization. The second step is the step where the patient views the calendar and selects the preferred date and time slot. In the third step, he/she will insert information regarding why he/she wishes to visit the doctor. In the final step, he/she will be viewing the summary.

5.4 Fitbit Integration and Remote Monitoring

5.4.1 OAuth 2.0 Flow

In this case, the authentication process was done using the OAuth 2.0 flow. With this mechanism, we can monitor many health metrics like physical activity, heartbeat, sleep metrics, users' data, oxygen level, and breathing rate. Here, the security of tokens is handled independently, and they are updated every time we make an API call.

5.4.2 Data Sync Architecture

In respect of our data sync solution, there are two kinds of health metrics collected in VitalSign table: daily (mean/ resting heart rate, number of steps, calories burned, distance covered, sleep hours/ periods, SpO2, HRV, respiratory rate, and Active Zone Minutes) and minute-by-minute metrics (heart rate). The latter metrics are obtained from IntradayVitalSign table. Therefore, it means that we need to generate up to 1,440 metrics daily in order to get information on metrics like heart rate.

5.5 AI Heart Health Analysis Engine

5.5.1 Phase 1: Rule-Based Alerts

The rule-based alert setting process is the first stage of working of our AI Engine. It consists of six types of alerts in accordance with patients' ages. Afterwards, personal thresholds for these metrics are defined considering patients' particularities. If patients are athletes, then the threshold value for bradycardia is lowered by 10 bpm. When beta-blockers are used, the threshold for tachycardia is lowered by 10 bpm. In case of stimulants use, the threshold value should be increased by 5 bpm. There is no bradycardia threshold when pacemaker is used.

5.5.2 Phase 2: Statistical Analysis

The second stage involves the calculation of statistics based on all gathered data. The average resting heart rate over the entire night period, the highest heart rate value, and the analysis of the obtained data by calculating the linear regression curve for seven days are evaluated. In addition to this, in the second stage, data anomalies are sought by applying the Z-Score formula, wherein only the data values having $|Z| > 3.0$ are considered as abnormal values. This enables the device to identify if there are any heart rate anomalies, along with their duration. Based on the above parameters, the HeartHealth value from 0 to 100 will be obtained. A rating of 40 points will be awarded for maintaining a normal heart resting rate, while 30 points will be allotted for no heart rate anomalies and abnormally high heart rate values.

5.5.3 Phase 3: LLM Narrative Generation

In the final stage, structured data of patients will be transferred to the LLM on Cerebras Llama 3.3-70b. Thanks to the processing, a narrative will be generated in order to make an appropriate summary of the obtained information. As variables of input, it should be highlighted that the demographic characteristics of the patients, their history, way of life, indicators of cardiovascular health, and alerts concerning the actual status of each patient are included. In addition to this, a correctly structured JSON file will be created in the following format: a brief narrative about the case in question, the corresponding diagnosis, suggestions regarding treatment. At the same time, the level of risk is assessed by the algorithm as either LOW, MODERATE, HIGH, or CRITICAL. Some prompts are provided in the system, which implies that there are some particular situations that the AI considers. They include such factors as side effects from beta-blockers, baseline values specific for patients who are athletes, co-risk, arrhythmia, and others.

5.6 Video Consultation Implementation

To facilitate video consultation development we utilize PeerJS framework. In particular, once the doctor starts the video call, a VideoSession object is instantiated. The code needed to join the conference room is then generated. In other words, both the physician and the patient become connected to the PeerJS signaling server responsible for exchanging ICE candidates and configuring the network using the STUN/TURN protocol. After the establishment of the session, the media stream is directly delivered from one peer to another. The video call is displayed through the VideoCallRoom.tsx component.

5.7 Prescription and Medication Management

The drug database that has been incorporated in the application contains more than 8,557 drugs. The database consists of comprehensive information such as drug name, generic name, strength, formulations, composition, indications, adverse reactions, and manufacturer. It is possible for the users of the application to obtain the required information through the full-text search engine. Search will be performed on brandName, genericName, and searchText among others. Our system utilizes database indexing which ensures very quick searching. Prescriptions transition from draft stage to active prescription. The prescriber is free to input details pertaining to the prescription such as dosage strength, route of administration, sig, frequency, and duration.

5.8 Deployment and DevOps

The deployment of HealthSync to Vercel is done through the automated DevOps process to ensure that there are no manual processes involved. The automatic execution of three cron jobs is ensured. First, a daily job to analyze hearts occurs at 6:00 AM. Second, the execution of vitals cleaning process takes place at 3:00 AM on Sundays. Finally, the daily reminder service process occurs at 8:00 AM. Whenever there is a change in the system, the management of Prisma client code generation, database migration, and production build occurs automatically.

5.9 Conclusion

In the next section of this paper, the implementation of HealthSync is going to be discussed extensively, putting a focus on all of the significant aspects of its development. To start with, the role of JWT Token authentication mechanism and RBAC permissions in ensuring secure storage of user information and access permissions will be discussed in more detail so as to understand better the measures taken to ensure the security of user data. Next, it will be shown how the module responsible for scheduling patient appointments contributes to the ease of use of the app in creating such appointments. Lastly, it will be explained how the application integrates with the Fitbit health tracker.

The structure and working principles of the artificial intelligence system used to detect heart diseases will be elaborated upon in this part of the paper. The algorithm combines the implementation of three different methods: rule checks, statistical analysis, and generation of narratives by a large language model. It is designed to offer useful suggestions on the patients' condition that can be utilized by doctors during the diagnosis process. Using WebRTC and PeerJS protocols, the developers implemented an integrated live video consultation system that allows to deliver remote medical services through the application.

Chapter 6

System Testing & Evaluation

The development and testing processes for implementing the proposed healthsync software are discussed in this chapter. The software undergoes various stages of testing to establish its reliability, security, and efficiency. As far as testing process goes, all modules of the proposed application will be included.

6.1 Test Strategy

The Testing methodology follows a multi-layered process whereby individual modules are tested, and at other times, the whole system is tested. The following is how the application is going to be tested.

- **Unit Testing (Jest):**
  This allows testing individual functions and modules in the application and its business logic.

- **Component Testing:**
  It entails testing individual components of the application to see whether they work as per the set requirements. The component tests entail authentication, RBAC, and notifications services tests.

- **Integration Testing:**
  In this regard, testing of interaction between various modules takes place. Tests for JWT authentication, middleware, and RBAC flow will take place in this layer.

- **End-to-End Testing (Playwright):**
  This is testing which involves simulation of actions performed by users. Appointment and consultations tests fall under this layer of testing.

- **Manual Testing:**
  This involves manual testing of visually presented modules of the application. The visual modules include the dashboard, fitbit integration, and video consultation.

6.2 Component Testing

Component testing was done to check the functionality of the main modules within HealthSync. These are as follows:

- The authentication module (JWT management and login)
- RBAC
- The notification module
- The configuration module
- The multi-tenant database

All the above components were tested individually to ensure their perfect functionality. integration.

6.3 Unit Testing

Jest testing tool was used to perform the unit testing to check if the backend and the API are functioning.**Authentication Tests**

- Login validation with correct/incorrect credentials
- JWT token generation and verification
- Role-based access validation

Permission Tests

- Permission of the roles SUPERADMIN, PROVIDER, and PATIENT was tested.
- Access permission was performed, and it was ensured that DENY overrides ALLOW rules.
- Caching validation

Configuration Tests

- Default vs clinic-level configuration overrides
- Key filtering and locking mechanisms

Invite Tests

- User invitation flow
- Password hashing validation

Notification Tests

- Quiet hours suppression
- Critical alert bypass

Database Context Tests

- Tenant-based isolation
- Cross-tenant access restriction
- Unit tests for all were performed, and it was observed that the percentage of passing was 100%, making it very strong.

6.4 Integrated Testing

With the help of integration testing, it could be determined how the integration of various components took place and whether they worked well together or not. The reason behind the use of integration testing was that when two things are combined, we should get the desired output.

- JWT authentication with middleware pipeline
- RBAC system with permission flow
- Configuration hierarchy resolution

These tests ensured that interconnected modules behave correctly when combined, especially in multi-tenant scenarios.

6.5 System Testing

Playwright testing tool was used as the E2E testing framework for system testing. Some scenarios are shown below.

Onboarding Workflow Testing

- Clinic creation
- Admin onboarding
- Doctor invitation and verification
- Patient registration
- Appointment booking
- Data scoping and cross-clinic restriction

This validated complete real-world workflows from start to finish.

6.6 Test Cases

Below are sample test cases following the required table format:

6.6.1 Authentication and Authorization

Test Scenario ID	**TS-001**	**Test Case ID**	**TC-001**
Test Case Description	Verify login, JWT authentication, and RBAC		
Test Priority	High		
Prerequisite	User registered or invited		
Post-Requisite	User redirected to role dashboard		
S. No	Action	Inputs	Expected Outcome
1	Enter credentials	Email, Password	Credentials validated
2	Submit login	Valid data	JWT generated
3	Role identified	User role	RBAC applied
4	Login success	-	Redirect to dashboard

Table 22: Authentication and Authorization Test Case

6.6.2 Patient Management

Test Scenario ID	**TS-002**	**Test Case ID**	**TC-002**
Test Case Description	Verify patient registration and profile		
Test Priority	High		
Prerequisite	Admin logged in		
Post-Requisite	Patient record created		
S. No	Action	Inputs	Expected Outcome

1	Open form	-	Form displayed
2	Enter details	Patient data	MRN generated
3	Add profile	Health info	Profile saved
4	Assign doctor	Doctor ID	Mapping updated

Table 23: Patient Management Test Case

6.6.3 Appointment Management

Test Scenario ID	**TS-003**	**Test Case ID**	**TC-003**
Test Case Description	Verify appointment booking and lifecycle		
Test Priority	High		
Prerequisite	User logged in		
Post-Requisite	Appointment created		
S. No	Action	Inputs	Expected Outcome
1	Start booking	-	Wizard shown
2	Select doctor	Doctor	Slots displayed
3	Choose time	Date/Time	Slot validated
4	Confirm	Valid data	Appointment created

Table 24: Appointment Management Test Case

6.6.4 Prescription Management

Test Scenario ID	**TS-004**	**Test Case ID**	**TC-004**
Test Case Description	Verify prescription creation		
Test Priority	High		
Prerequisite	Provider logged in		

Post-Requisite	Prescription saved		
S. No	Action	Inputs	Expected Outcome
1	Open module	-	Form displayed
2	Search drug	Drug name	Drug fetched
3	Add details	Dose, route	Data stored
4	Save	Valid data	Status set

Table 25: Prescription Management Test Case

6.6.5 Medical Records

Test Scenario ID	**TS-005**	**Test Case ID**	**TC-005**
Test Case Description	Verify SOAP medical record creation		
Test Priority	High		
Prerequisite	Consultation active		
Post-Requisite	Record saved		
S. No	Action	Inputs	Expected Outcome
1	Open consultation	-	Record loaded
2	Enter subjective	Symptoms	Stored
3	Enter objective	Results	Stored
4	Save record	Valid data	Linked to consultation

Table 26: Medical Records Test Case

6.6.6 Chat and Messaging

Test Scenario ID	TS-006	Test Case ID	TC-006
Test Case Description	Verify real-time chat		
Test Priority	Medium		
Prerequisite	User logged in		
Post-Requisite	Message delivered		
S. No	Action	Inputs	Expected Outcome
1	Open chat	-	Conversation loaded
2	Send message	Text	Delivered
3	Assign doctor	-	Doctor assigned
4	Resolve chat	-	Status updated

Table 27: Chat And Messaging Test Case

6.6.7 Video Consultation

Test Scenario ID	TS-007	Test Case ID	TC-007
Test Case Description	Verify video consultation		
Test Priority	High		
Prerequisite	Appointment scheduled		
Post-Requisite	Session completed		
S. No	Action	Inputs	Expected Outcome
1	Join session	-	Session started
2	Connect	User	WebRTC connected
3	Interaction	Video/audio	Stable session
4	End session	-	Status updated

Table 28: Video Consultation Test Case

6.6.8 Fitbit Integration

Test Scenario ID	**TS-008**	**Test Case ID**	**TC-008**
Test Case Description	Verify Fitbit data sync		
Test Priority	High		
Prerequisite	Fitbit connected		
Post-Requisite	Data stored		
S. No	Action	Inputs	Expected Outcome
1	Connect Fitbit	OAuth	Token stored
2	Fetch data	API	Data retrieved
3	Sync data	-	Stored
4	Refresh token	Expired	Auto refreshed

Table 29: Fitbit Integration Test Case

6.6.9 AI Heart Health Analysis

Test Scenario ID	**TS-009**	**Test Case ID**	**TC-009**
Test Case Description	Verify AI health analysis		
Test Priority	High		
Prerequisite	Vitals available		
Post-Requisite	Insights generated		
S. No	Action	Inputs	Expected Outcome
1	Collect data	Vitals	Processed
2	Analyze	Data	Score generated
3	Detect anomaly	-	Events detected
4	Generate alert	-	Alert created

Table 30: AI Heart Health Analysis Test Case

6.6.10 Reports and Documents

Test Scenario ID	**TS-010**	**Test Case ID**	**TC-010**
Test Case Description	Verify report handling		
Test Priority	Medium		
Prerequisite	User logged in		
Post-Requisite	File available		
S. No	Action	Inputs	Expected Outcome
1	Upload file	Report	Stored
2	View file	-	Retrieved
3	Annotate	Notes	Saved
4	Generate PDF	-	Downloadable

Table 31: Reports And Documents Test Case

6.6.11 Notifications

Test Scenario ID	**TS-011**	**Test Case ID**	**TC-011**
Test Case Description	Verify notifications		
Test Priority	Medium		
Prerequisite	User logged in		
Post-Requisite	Notification received		
S. No	Action	Inputs	Expected Outcome
1	Trigger event	-	Notification created
2	Send	-	Delivered
3	Open	-	Marked read
4	Mark all read	-	Updated

Table 32: Notifications Test Case

6.6.12 Administration

Test Scenario ID	TS-012	Test Case ID	TC-012
Test Case Description	Verify admin controls		
Test Priority	High		
Prerequisite	Admin logged in		
Post-Requisite	System updated		
S. No	**Action**	**Inputs**	**Expected Outcome**
1	Create clinic	Data	Stored
2	Invite staff	Email	Sent
3	Assign roles	Role	Updated
4	Update settings	Config	Applied

Table 33: Administration Test Case

6.7 Results & Evaluation

Testing the system involved a comprehensive approach through unit testing, integration testing, and system testing. Emphasis was mainly placed on ensuring that all functional requirements were fully implemented to perfection. All modules were subjected to rigorous testing to evaluate their effectiveness.

It is evident from the results that all critical operations have been completed to perfection under normal circumstances. They involve the login and security, management of patients, appointments, and prescriptions. It has also been possible to confirm the implementation of medical history check-up, chatting and messaging, video calling, synchronization of the Fitbit application, and analysis of the heart's well-being using artificial intelligence.

All of these essential components have been efficiently tested and no major problems have been identified when using them. Small problems pertaining to managing the input form as well as some exceptions have been handled with the help of software enhancements.

Our solution has been proved efficient based on the performance of our solution under different kinds of loads. The role-based access control strategy that we used became quite useful in ensuring that there is no illegal access to our data. Data was accessed only by individuals having appropriate clearance. Third-party applications like the Fitbit API and artificial intelligence solutions were instrumental in addressing exceptions and disconnections. To summarize, all of our SRS requirements have been satisfied.

6.8 Conclusion

Testing of the system has been carried out thoroughly throughout this chapter by verifying and validating the system against all objectives established.

It is clear from the results obtained that the system under consideration is highly reliable and safe enough for use in medical practice and capable of dealing with different situations that arise in reality. The system does an excellent job by fulfilling its main objectives, and everything works smoothly.

Having finished with the process of testing, it is high time to put the system into use by implementing it in a medical environment. Future improvements may be done to improve the performance of the system and its AI capabilities with many users.

Chapter 7
Conclusion

It can be concluded that HealthSync managed to prove its success through showing that it is possible to develop a scalable AI-based solution for healthcare that would combine aspects of patient management, remote monitoring, and smart analytics. All set goals have been met, and this platform itself serves as a great foundation for further developments in healthcare.

Now, HealthSync is completely operational after all testing and contains all necessary functions for all five types of users. Through making use of the next technologies including Next.js 15, React 19, PostgreSQL, Fitbit wearables, and AI models provided by Cerebras Llama 3.3-70B, we managed to prove the possibility of implementing this kind of solution.

7.1 Contributions

There is a variety of valuable contributions that HealthSync can make within the domain of health IT:

1. **Unified Healthcare Platform**
   It combines a number of healthcare solutions into one product which guarantees absence of data scattering and the ability to get access to various healthcare services, including appointment bookings, data records management, medications control and video-calling.

2. **AI Heart Health Engine**
   The presented three-stage model not only provides some basic information about the functioning of one's heart but also incorporates personal health alerts, statistic trends, and even a clinical narrative created through the use of the Llama 3.3-70B model.

3. Wearable Integration

A complete data chain for Fitbit devices has been developed making it possible to monitor patients' heartbeats every minute.

4. Role-Based Access Control

Our reliable solution uses management of numerous roles with fine-grained permissions. Using ALLOW/DENY principle, permission cache and clinic data isolation, we succeeded in reaching a certain security level that is necessary for the medical professional environment.

5. Scalable Database Design

The database designed by us contains more than 28 tables. The architecture is optimized for workflows, analysis and machine learning in the healthcare field. For performance and proper indexing of data in the database, we implemented appropriate indexes, enum types, and cascading deletes.

7.2 Reflections

Strengths

- Modern full-stack architecture using Next.js 15 and React 19
- Strong type safety with TypeScript (strict mode)
- Efficient ORM using Prisma for type-safe database access
- Modular AI design allowing independent improvement of each phase
- Production deployment with automated cron jobs on Vercel

Challenges

- Timing and retries had to be carefully planned to bypass the constraints imposed by Fitbit’s API rate limiting policy.
- Streaming real-time audio and video communications was exceptionally difficult while using WebRTC.
- Developing the correct prompts for the AI to create accurate responses without creating false data was tough.

7.3 Future work

- It is necessary to optimize the user interface to make it mobile-friendly.
- The ability to predict certain patient populations that are vulnerable to illnesses through predictive models is possible.
- Clinical training of the AI will enhance its precision.
- Implement a native mobile application (iOS/Android)
- Expand wearable integrations (Apple Watch, Samsung Health, Garmin)
- Add billing and insurance management modules
- Implement HL7 FHIR compliance for healthcare interoperability

References

- https://dev.fitbit.com/login
- **https://nextjs.org/**
- **https://railway.com/**
- **https://www.hhs.gov/hipaa/index.html**
- **https://www.cerebras.ai/inference**
- **K. He et al., "Deep Learning in Healthcare,"** ***IEEE Journal*****, 2022.**
- **J. Smith et al., "Telemedicine Systems and Applications,"** ***Springer*****, 2021.**
- **A. Brown, "IoT-Based Remote Patient Monitoring Systems,"** ***Elsevier*****, 2023.**
- **M. Johnson, "AI in Clinical Decision Support Systems,"** ***IEEE Access*****, 2022.**
- **https://supabase.com/dashboard/project/wrnxoqwkjxgkhwsvhclg/storage/files**
- **https://nodejs.org/en**

APPENDIX A

TURNITIN AI DETECTION REPORT

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

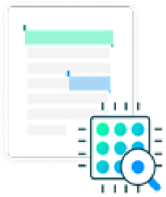
Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.



False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

turnitin Page 2 of 92 - AI Writing Overview Submission ID trn:oid:::3618:135393339

APPENDIX B

TURNITIN PLAGIRISM REPORT

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- **131** Not Cited or Quoted 14%
  Matches with neither in-text citation nor quotation marks
- **2** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
3% Publications
13% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **131** Not Cited or Quoted 14%
  Matches with neither in-text citation nor quotation marks
- **2** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
3%	Publications
13%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.bahria.edu.pk	2%
2	Student papers	Higher Education Commission Pakistan on 2026-04-08	1%
3	Student papers	Higher Education Commission Pakistan on 2020-07-22	<1%
4	Internet	www.theibfr.com	<1%
5	Student papers	Islington College,Nepal on 2025-12-29	<1%
6	Internet	prr.hec.gov.pk	<1%
7	Internet	www.coursehero.com	<1%
8	Student papers	University of Greenwich on 2023-04-18	<1%
9	Student papers	University of Wales Institute, Cardiff on 2025-09-20	<1%
10	Student papers	Islington College,Nepal on 2025-12-29	<1%

11	Student papers	
Kingston University on 2025-04-28		<1%
12	Student papers	
Ain Shams University on 2020-06-24		<1%
13	Student papers	
Higher Education Commission Pakistan on 2021-02-03		<1%
14	Internet	
github.com		<1%
15	Student papers	
Bannari Amman Institute of Technology on 2026-04-10		<1%
16	Student papers	
Asia Pacific Instutute of Information Technology on 2026-02-04		<1%
17	Student papers	
Higher Education Commission Pakistan on 2025-07-15		<1%
18	Student papers	
Asia Pacific University College of Technology and Innovation (UCTI) on 2025-09-03		<1%
19	Student papers	
University of Ulster on 2013-04-25		<1%
20	Internet	
dict.mizoram.gov.in		<1%
21	Student papers	
Bindura University of Science Education on 2024-05-30		<1%
22	Student papers	
Middlesex University on 2026-02-06		<1%
23	Publication	
Phenikaa University		<1%
24	Student papers	
University of Leicester on 2025-09-05		<1%

25 Student papers

Feng Chia University on 2020-03-09 <1%

26 Student papers

American Public University System on 2026-03-01 <1%

27 Student papers

Asia Pacific Instutute of Information Technology on 2021-11-15 <1%

28 Student papers

Islington College,Nepal on 2025-12-30 <1%

29 Student papers

Islington College,Nepal on 2025-12-30 <1%

30 Student papers

Universiti Sains Islam Malaysia on 2017-01-09 <1%

31 Student papers

CTI Education Group on 2008-11-19 <1%

32 Student papers

Los Rios Community College District on 2011-04-12 <1%

33 Student papers

University of Arizona Global Campus (Canvas LTI 1.1) on 2025-12-25 <1%

34 Student papers

University of Cape Town on 2017-01-08 <1%

35 Student papers

University of Greenwich on 2017-02-01 <1%

36 Student papers

Monash University on 2025-10-08 <1%

37 Student papers

University of Northampton on 2025-08-25 <1%

38 Student papers

University of Wales Institute, Cardiff on 2020-09-30 <1%

39 Student papers

King's Own Institute on 2022-01-30 <1%

40 Internet

stuartpiltch.org <1%

41 Student papers

Curtin University LTI on 2026-04-02 <1%

42 Internet

storm.prohosting.com <1%

43 Student papers

Islington College,Nepal on 2025-12-29 <1%

44 Student papers

University of Bedfordshire on 2025-08-30 <1%

45 Student papers

University of Greenwich on 2021-11-29 <1%

46 Student papers

University of Westminster on 2023-02-14 <1%

47 Internet

repository.president.ac.id <1%

48 Student papers

CSU, San Jose State University on 2018-05-10 <1%

49 Student papers

Higher Education Commission Pakistan on 2020-12-17 <1%

50 Internet

afribary.com <1%

51 Internet

fdocuments.us <1%

52 Internet

www.accurayresearch.com <1%

53	Internet	
www.slideshare.net		<1%
54	Student papers	
President University on 2019-12-19		<1%
55	Publication	
Sunil Gupta, Monit Kapoor, Samiya Khan. "Handbook on Augmenting Telehealth ...		<1%
56	Student papers	
University of Galway Canvas on 2026-04-05		<1%
57	Student papers	
University of Greenwich on 2009-12-15		<1%
58	Student papers	
University of Luton on 2011-05-27		<1%
59	Student papers	
University of Macau on 2024-05-13		<1%
60	Student papers	
University of Newcastle on 2018-12-04		<1%
61	Student papers	
University of Newcastle upon Tyne on 2019-09-09		<1%
62	Student papers	
Waterford Institute of Technology on 2025-04-30		<1%
63	Internet	
jicrcr.com		<1%
64	Student papers	
American University of Iraq on 2025-12-08		<1%
65	Student papers	
American University of the Middle East on 2025-04-14		<1%
66	Student papers	
BB9.1 PROD on 2025-04-22		<1%

67 Student papers
BB9.1 PROD on 2025-04-22 <1%

68 Student papers
City University of Hong Kong on 2015-07-13 <1%

69 Student papers
Heriot-Watt University on 2022-08-15 <1%

70 Student papers
Informatics Education Limited on 2010-02-04 <1%

71 Student papers
Islington College,Nepal on 2025-12-29 <1%

72 Student papers
Kingston University on 2021-05-21 <1%

73 Student papers
Koc University on 2025-11-03 <1%

74 Student papers
Midlands State University on 2018-05-10 <1%

75 Student papers
Open University on 2025-07-17 <1%

76 Student papers
Pukyong National University on 2016-05-24 <1%

77 Student papers
University of Greenwich on 2022-12-08 <1%

78 Student papers
University of Mauritius on 2015-03-30 <1%

79 Student papers
University of Westminster on 2025-12-14 <1%

80 Student papers
aaup on 2025-06-22 <1%

81 Internet
caldito.me <1%

82 Internet
theses.lib.polyu.edu.hk <1%

83 Internet
tryflowdrive.com <1%

84 Student papers
Adventist University of Central Africa on 2025-07-11 <1%

85 Student papers
CSU, San Jose State University on 2015-08-07 <1%

86 Student papers
Arab Open University on 2026-03-31 <1%

87 Student papers
Higher Education Commission Pakistan on 2019-04-05 <1%

88 Student papers
Higher Education Commission Pakistan on 2025-05-14 <1%

89 Student papers
Higher Education Commission Pakistan on 2026-03-28 <1%

90 Publication
Rakesh Kumar, Meenu Gupta. "Innovation in HealthTech - A Roadmap for Empow... <1%

91 Student papers
University of Greenwich on 2025-11-29 <1%

92 Student papers
University of Westminster on 2025-12-17 <1%

93 Student papers
University of the Philippines Baguio on 2026-02-04 <1%

APPENDIX C

API ENDPOINTS REFERENCE

1. **Authentication Endpoints**
   - **POST /api/auth/register**
     Used for user registration.
   - **POST /api/auth/login**
     Used for user login.
   - **POST /api/auth/logout**
     Used for user logout.
   - **POST /api/auth/refresh-token**
     Used to refresh the JWT authentication token.
2. **Patient Endpoints**
   - **GET /api/patients/:id**
     Retrieves the patient profile.
   - **PUT /api/patients/:id**
     Updates the patient profile.
   - **GET /api/patients/:id/medical-records**
     Retrieves the patient's medical records.
   - **GET /api/patients/:id/vitals**
     Retrieves patient vitals (integrated with FitBit).
3. **Doctor Endpoints**
   - **GET /api/doctors/:id**
     Retrieves the doctor profile.
   - **GET /api/doctors/:id/appointments**
     Retrieves the doctor's appointments.
   - **PUT /api/doctors/:id/profile**
     Updates the doctor profile.
4. **Appointments Endpoints**
   - **GET /api/appointments**
     Lists all appointments (filtered based on user).
   - **POST /api/appointments**
     Creates a new appointment.

- **PUT /api/appointments/:id**

  Updates an existing appointment.

- **DELETE /api/appointments/:id**

  Cancels an appointment.

5. Messages Endpoints

- **GET /api/messages**

  Retrieves user messages.

- **POST /api/messages**

  Sends a new message.

- **PUT /api/messages/:id/read**

  Marks a message as read.