



**Kombolcha Institute of Technology
College of Informatics
Department of Software Engineering**

We here by Submitted a project in Hakim AI: AI-Powered Symptom Checker & Online Doctor
Consultation Platform
in partial fulfilment of bachelor's Degree in Software Engineering
Submitted to the departments of Software Engineering

Group Members:

No.	Name	ID Number
1.	Nuredin Fentaw.....	WOUR/4330/14
2.	Fuad Abdela	WOUR/4345/14

Advisor's Name: _____ **Signature** _____

Discriminations

We hereby declare that our project titled Hakim AI: AI-Powered Symptom Checker & Online Doctor Consultation Platform is original and not submitted/Published by any individual/ Organization.

Group members

S.N. Name List ID Number date ----/----/2025 sign:

- 1.
- 2.
- 3.
- 4.

Advisor's Name: Date: -----/----/2025 Sign

Acknowledgement

We express our sincere gratitude to everyone who contributed to the successful completion of Hakim AI Version 1.0.

We are deeply thankful to our supervisors and instructors in the Software Engineering Internship Program for their invaluable guidance, constructive feedback, and consistent support throughout the project. Their expertise greatly influenced the direction and quality of this work.

Special thanks go to our mentors and peers for their participation in informal consultations, prototype testing, and usability feedback. Their practical insights into Ethiopia's healthcare challenges helped refine the platform's relevance and user experience.

We also appreciate the open-source community and the providers of essential tools and frameworks, Next.js, scikit-learn, Chapa, and others—whose resources enabled rapid development under internship constraints.

Finally, we thank our families and friends for their patience, understanding, and moral support during this intensive period.

This project would not have been possible without the collective efforts of all these individuals and organizations.

Nuredin Fentaw Fuad Abdela

December ___, 2025

Table of contents

Catalog

Acknowledgement.....	II
Table of contents	III
List of Tables	V
List of Figures/Illustrations	V
List of Abbreviations, Symbols, and Specialized Nomenclature	VI
.....	VIII
Abstract	VIII
Chapter One: Introduction	1
1.1 Introduction	1
1.1.1 Background of the Organization	1
1.1.2 Existing System	2
1.1.3 Statement of the Problem	2
1.1.4 Proposed System	3
1.2 Objectives	3
1.2.1 General Objective	3
1.2.2 Specific Objectives	4
1.3 Scope and Limitations	4
1.3.1 Scope	4
1.3.2 Limitations	5
1.4 Methodology	6
1.4.1 Development Approach	6
1.4.2 Data Collection and Fact-Finding Techniques	6
1.4.3 System Analysis and Design Approach	7
1.4.4 Technology Requirements	7
1.5 Feasibility Study	9
1.5.1 Technical Feasibility	9
1.5.2 Operational Feasibility	9
1.5.3 Economic Feasibility	9
1.5.4 Legal Feasibility	10
1.5.5 Political Feasibility	10
1.6 Risk Assessment Strategy	11
1.7 Significance of the Project	12
1.7.1 Beneficiaries of the Project	12
1.8 Project Schedule	13
1.9 Project Budget Breakdown and Cost Analysis	14
1.10 Team Composition	15
Chapter Two: System Requirement Specification	16
2.1 Background (Overview)	16
2.1.1 Scope	17
2.1.2 Purpose	18
2.1.3 Document Convention	18
2.1.4 Intended Audience and Suggested Readings	19
2.1.5 List of Acronyms, Abbreviations and Definitions	20
2.2 Overall Description of Software Requirements	20

2.2.1 Product Perspectives	20
2.2.2 Product Features/Functions	21
2.2.3 User Characteristics	21
2.3 General Constraints	22
2.3.1 Software Constraints	23
2.3.2 Hardware Constraints	23
2.3.3 Assumptions and Dependencies	23
2.3.4 User Documentation	24
2.4 Specific Requirements	24
2.4.1 User Requirements	25
2.4.2 Functional Requirements	25
2.4.3 Non-Functional Requirements	26
2.5 External Interface Requirements	27
2.5.1 User Interfaces	27
2.5.2 Hardware Interfaces	31
2.5.3 Software Interfaces	31
2.5.4 Communication Interfaces	32
2.6 System Requirements Modeling	32
2.7 Essential Use Case Diagrams	32
Chapter Three: Requirement Analysis Modeling	33
3.1 Overview of Analysis Model	33
3.2 System Use Case Diagram	33
3.2.1 Use Case Descriptions	34
3.3 Sequence Diagram	40
3.4 Activity Diagrams	44
3.5 Conceptual Class Diagram	46
Chapter Four: System Design	47
4.1 Overview	47

List of Tables

- Table 1.1 – Project Timeline and Phases
- Table 1.2 – Project Budget Breakdown
- Table 1.3 – Risk Assessment Register
- Table 2.1 – Functional Requirements
- Table 2.2 – Non-Functional Requirements
- Table 4.1 – Subsystem Decomposition
- Table 4.2 – Database Schema (Main Tables)
- Table 6.1 – Test Execution Summary
- Table 6.2 – Sample Test Cases

List of Figures/Illustrations

- Figure 3.1 – Use Case Diagram
- Figure 3.2 – Activity Diagram: Perform Symptom Check
- Figure 3.3 – Activity Diagram: Book Appointment
- Figure 3.4 – Sequence Diagram: Perform Symptom Check
- Figure 3.5 – Sequence Diagram: Book and Conduct Consultation
- Figure 3.6 – Conceptual Class Diagram
- Figure 4.1 – High-Level Component Diagram
- Figure 4.2 – Deployment Diagram
- Figure 4.3 – Detailed Class Diagram
- Figure 4.4 – Entity-Relationship (ER) Diagram
- Figure 5.1 – Symptom Input Form
- Figure 5.2 – AI Preliminary Assessment Report
- Figure 5.3 – Doctor Search and Booking Interface
- Figure 5.4 – Real-Time Video Consultation Room
- Figure 5.5 – Patient Medical History Timeline

List of Abbreviations, Symbols, and Specialized Nomenclature

This list includes key abbreviations, acronyms, and specialized terms used throughout the report for Hakim AI Version 1.0. All scientific and technical nomenclature follows standard conventions, with measurements aligned to the International System of Units (SI) where applicable.

Table. Abbreviations and Acronyms

Abbreviation	Full Form / Meaning
AI	Artificial Intelligence
API	Application Programming interface
FMOH	Federal Ministry of Health(Ethiopia)
FR	Functional Requirement
FSM	Finite State Machine
MVP	Minimum Viable Product
NFR	Non Functional Requirement
OOD	Object Oriented Design
PWA	Progressive Web App
RBAC	Role-Based Access Control
REST	Representational State Transfer
SRS	System Requirement Specification
UI	User Interfaces
UML	Unified Modeling Language
UX	User Experiences

WebRTC	Web Real-Time Communication
--------	-----------------------------

Table.Symbols and Specialized Nomenclature

Symbol/Term	Definition / Meaning	Context / Units (SI where applicable)
id	Unique identifier for database entities	Integer(primary Key)
JSONB	Binary JSON data type in PostgreSQL	Flexible structured storage
JWT	JSON Web Token	Authentication Token
TIMESTAMP WITH TIME ZONE	Date-time with timezone support	ISO 8601 format (e.g,2025-12-22 T1 4:30:00+03:00)
DECIMAL(m,n)	Precise decimal number	Monetary values(e.g,DECIMAL(10,2) for Ethiopian Birr)
<<include>>	UML stereotype for mandatory inclusion relationship	Use Case Diagram
<<extend>>	UML stereotype for conditional extension relationship	Use Case Diagram
SeverityLevel	Enumeration:mild,moderate,sereve	AI assessment output

Hakim AI: AI-Powered Symptom Checker & Online Doctor Consultation Platform

" " -

Abstract

1

This project presents Hakim AI Version 1.0, an innovative AI-powered symptom checker and online doctor consultation platform developed to address key healthcare access challenges in Ethiopia. The primary problems tackled include geographical barriers (long travel distances), overcrowded facilities, specialist shortages, and risks of misinformation from unstructured self-diagnosis. The platform offers a responsive web-based Progressive Web App (PWA) where users input symptoms via structured forms and receive instant AI-generated preliminary reports (top probable conditions with probabilities, severity levels—mild/moderate/severe—and recommendations), accompanied by mandatory safety disclaimers. For moderate or severe cases, users can seamlessly book paid video consultations with verified doctors, featuring real-time communication (WebRTC primary, Agora fallback), digital prescriptions, and medical history storage.

Implemented using modern technologies—Next.js (frontend), Node.js/FastAPI (backend), PostgreSQL (database), scikit-learn Random Forest (AI), and Chapa (payments)—the MVP was completed and deployed on Vercel within an 8-week internship timeline. Key findings include high usability (SUS score >80), reliable performance on low-bandwidth networks, AI accuracy >75% on validation data, and robust concurrency handling for bookings. The monetization model (10–20% commissions, premium tiers) supports sustainability.

Hakim AI significantly contributes to Ethiopia's digital health transformation (aligned with FMOH Digital Health Blueprint 2021–2030) by empowering rural and underserved patients with timely guidance, reducing facility congestion, and fostering proactive care. It demonstrates that focused software engineering can yield impactful, affordable health technology solutions in resource-constrained settings.

Chapter One: Introduction

1.1 Introduction

This report documents the development lifecycle of Hakim AI Version 1.0, an AI-powered symptom checker and online doctor consultation platform designed to improve healthcare accessibility in underserved regions of Ethiopia.

Hakim AI is a full-stack Progressive Web App that allows users to input symptoms via structured forms and receive instant AI-generated preliminary assessments, including probable conditions, severity levels, and recommendations. For moderate or severe cases, users can book paid video consultations with verified doctors, featuring real-time video/chat, digital prescriptions, and medical history storage.

The platform tackles key barriers: geographical distance, overcrowded facilities, specialist shortages, and risks of unreliable self-diagnosis. By combining AI triage with telemedicine and a sustainable monetization model, it delivers timely, affordable guidance, especially for rural communities, while reducing hospital congestion and supporting proactive care.

Developed during a software engineering internship, Hakim AI uses modern technologies (Next.js, Node.js/FastAPI, PostgreSQL, scikit-learn, Chapa) to create a scalable digital health solution aligned with national priorities.

1.1.1 Background of the Organization

Hakim AI aligns with the strategic goals of Ethiopia's Federal Ministry of Health (FMOH), the primary body overseeing the national health sector. FMOH Vision: "To see healthy, productive, and prosperous Ethiopians." Mission: "To promote health and well-being through equitable, high-quality health services."

Key frameworks include the Health Sector Transformation Plan (HSTP), Information Revolution initiative, and Digital Health Blueprint (2021–2030), which aims for "quality, affordable, equitable, and technologically enabled health service delivery by 2030" via effective, secure digital systems.

Hakim AI supports these by providing AI-based preliminary assessments and remote consultations, addressing rural access gaps, facility overload, and specialist shortages. It promotes early detection, reduces unnecessary visits, and ensures equitable care through secure data handling, AI disclaimers, and potential interoperability.

1.1.2 Existing System

Ethiopia's current system relies mainly on in-person visits to health facilities, with manual triage by nurses or community workers. Oversight is by FMOH, which is introducing digital tools via the Digital Health Blueprint (2021–2030) and related strategies.

Merits:

- Reliable in-person examinations and interventions.
- Subsidized/free basic services for low-income groups.
- Established facility network with grassroots data support (e.g., eCHIS).
- Strong regulatory compliance.

Demerits (vs. Hakim AI):

- Limited rural access due to travel distances and overcrowding.
- Inefficiency from manual processes and long waits.
- High misinformation risk from unstructured self-diagnosis.
- Uneven scalability and lack of advanced digital features nationwide.
- No built-in incentives/monetization for providers.

Hakim AI builds on this foundation by adding automated AI triage and telemedicine to improve accessibility, efficiency, and sustainability.

1.1.3 Statement of the Problem

Ethiopia's healthcare faces three major interconnected challenges:

1. **Limited Access in Rural Areas** (>80% of population): Long travel distances delay diagnosis and worsen preventable conditions. Doctor-patient ratio is ~1:10,000 (vs. WHO's 1:1,000), exacerbated by geography, conflicts, disasters, and urban-rural resource imbalance. Metrics: high mortality from treatable diseases, urban-rural disparities.

2. **Urban Facility Overcrowding:** Influx of rural patients causes extended waits (4–6+ hours), staff overload, and higher infection risks. Driven by population growth (projected 130 million by 2025), worker shortages, and displacement crises.
3. **Risky Self-Diagnosis:** Access barriers lead to reliance on unreliable internet searches or traditional remedies (>40% self-medication). Low health literacy amplifies misinformation, especially during outbreaks.

Hakim AI addresses these via instant AI assessments, remote consultations, and digital triage, promoting early detection and reduced travel/congestion. Potential risks include AI inaccuracies, privacy concerns, and digital divide—mitigated by disclaimers, secure design, and future complementary measures.

1.1.4 Proposed System

Hakim AI is a complementary AI-powered platform integrating symptom checking with telemedicine.

Core Features:

- Structured symptom input → Random Forest AI report (probable conditions, probabilities, severity, recommendations) with safety disclaimers.
- Seamless booking of paid video consultations with verified doctors (license-checked).
- Real-time video/chat , digital prescriptions, and history storage.
- Tiered access: Free (limited assessments), Premium (unlimited + priority).
- Monetization: Doctor-set fees with 15% platform commission.

Deployed as a responsive Next.js PWA on Vercel, it extends reach via smartphones, supports FMOH digital goals, acts as first-line triage, and fosters a hybrid ecosystem for greater equity and efficiency without replacing physical care.

1.2 Objectives

1.2.1 General Objective

The general objective of Hakim AI is to design, develop, and deploy a fully functional AI-powered symptom checker and online doctor consultation platform that improves access to timely, reliable healthcare services in Ethiopia, especially in rural and underserved areas. It aims to reduce the burden on physical facilities, promote early detection, and support informed health decisions.

By integrating AI for preliminary assessments with secure telemedicine, the platform bridges patients and certified doctors, fosters a sustainable digital health ecosystem via affordable monetization, and contributes to FMOH goals of equitable care and digital transformation. It provides an efficient, user-friendly, scalable solution that empowers patients and offers doctors additional income and remote practice opportunities.

1.2.2 Specific Objectives

The specific objectives directly address the identified healthcare challenges:

- Enable rural/remote users to receive instant AI preliminary assessments without travel, reducing diagnostic delays and improving access for underserved populations.
- Implement AI severity classification and triage to guide mild cases toward self-management, decreasing unnecessary urban facility visits, alleviating congestion, and prioritizing critical care.
- Deliver structured AI analysis with evidence-based recommendations and professional disclaimers to minimize misinformation risks from unstructured self-diagnosis and promote safer health decisions.
- Facilitate seamless, secure online consultations (video, chat, digital prescriptions) with verified doctors, providing an affordable remote option that bridges geographical barriers and complements in-person care.

1.3 Scope and Limitations

1.3.1 Scope

Hakim AI is scoped as digital health platform focused on preliminary symptom assessment and remote consultations, targeted at the Ethiopian context with future scalability potential.

In Scope:

- Responsive web/PWA for symptom input and AI reports (probable conditions, severity, recommendations).
- Random Forest ML model integration via secure API.
- Telemedicine: doctor registration/verification, scheduling, real-time video/chat , prescriptions, history storage.

- Roles: Patient (free/premium tiers), Doctor, Admin (verification, analytics, payouts).
- Monetization: consultation fees (doctor-set + 15% commission), premium subscriptions.
- Authentication, Chapa payments, data privacy, Vercel deployment.
- Basic admin analytics.
- End-to-end suitable for pilot testing in areas with internet access.

Out of Scope:

- Native mobile apps (iOS/Android).
- Physical exams, labs, imaging, or emergency services.
- Regulatory certification or integration with national systems (DHIS2, eCHIS, EMRs).
- Advanced AI (image diagnosis).
- Specialized consultations beyond general practice.
- Large-scale clinical AI validation.
- Full multilingual support beyond basic English/Amharic.
- Pharmacy/delivery or insurance integration.

The project focuses on software/AI development and UX as a complementary tool to physical healthcare.

1.3.2 Limitations

As an internship MVP Hakim AI has the following constraints:

User Perspective:

- Requires reliable internet, limiting access in remote low-coverage areas.
- Limited multilingual support (basic English).
- No offline mode.
- Potential barriers for low digital literacy users (no voice/simplified interfaces).

Technological Perspective:

- Web/PWA only (no native apps; limited push notifications/device integration).
- Basic Random Forest model on public datasets (may miss Ethiopia-specific patterns).
- Structured inputs only (no image/NLP support).

System Perspective:

- No integration with national health systems.
- No pharmacy/insurance linkage.
- Untested for very high concurrency.
- Security lacks 2FA, penetration testing, or full regulatory auditing.

These stem from time/resource constraints and MVP focus, not design flaws. Future work can address them via partnerships, local data training, and extended development.

1.4 Methodology

Hakim AI was developed using a structured, flexible methodology tailored to the 10–13 week internship constraints, emphasizing rapid prototyping, iterative improvement, and delivery of a functional MVP.

1.4.1 Development Approach

Agile methodology was adopted for its flexibility in short-term projects involving evolving requirements and emerging technologies (AI, real-time communication).

Key Agile Practices Applied:

- **Iterative Development:** Short 1–2 week sprints covering requirements, backend setup, AI development, frontend, integrations, testing, and deployment.
- **Incremental Delivery:** Each sprint delivered working increments (e.g., functional AI API by sprint 3; full symptom-to-consultation flow later).
- **Continuous Feedback:** End-of-sprint reviews enabled adjustments, bug fixes, and reprioritization.
- **Collaboration:** Pair programming on critical components (AI-backend integration, payments) for knowledge sharing and quality.

Agile was chosen over Waterfall due to the need for experimentation and adaptation in innovative health-tech projects.

1.4.2 Data Collection and Fact-Finding Techniques

Data was gathered from secondary (primary source) and primary (supplementary) methods for requirements, design, AI training, and problem validation.

Secondary Sources:

- Public datasets (Kaggle) for AI training (symptom-disease mappings).
- Benchmarking of platforms (Ada Health, Babylon Health, Practo, Teladoc) via documentation and reviews.
- Technical references for tools (Next.js, scikit-learn, WebRTC, Chapa).

Primary Sources:

- Informal discussions with mentors, peers, and healthcare professionals to validate problems and refine features.
- Direct simulation of patient/doctor journeys for common scenarios.

Secondary sources dominated due to ethical/time constraints prohibiting real patient data collection.

1.4.3 System Analysis and Design Approach

An object-oriented analysis and design (OOAD) approach was combined with agile iterative prototyping.

Analysis Phase:

- Use case diagrams for actors (Patient, Doctor, Admin) and interactions.
- Sequence diagrams for flows (e.g., symptom submission → AI report → booking).
- Class diagrams for entities and PostgreSQL relationships.

Design Phase:

- Component-based Next.js UI (reusable elements like SymptomForm, VideoRoom).
- RESTful API design.
- Progressed from low-fidelity Figma wireframes to high-fidelity mocks and functional code.
- Sprint-end demos for early flaw detection (e.g., symptom input adjustments).
- MoSCoW method: Must-have (AI assessment, video consultation); Should-have (payments); Could-have (premium features), Won't-have: Native apps.

1.4.4 Technology Requirements

Software Requirements

Development & Runtime:

- Node.js v18+ or Python 3.10+ (FastAPI parts)
- Next.js 14+
- PostgreSQL 15+
- Git

Key Libraries/Frameworks:

- Frontend: React (Next.js), Zustand, Tailwind CSS, ZeGoCloud
- Backend: Express.js/FastAPI, Better-Auth
- AI: scikit-learn, pandas, numpy
- Payment: Chapa SDK
- Testing: Jest, React Testing Library, Cypress
- Deployment: Vercel CLI

Third-Party Services:

- Chapa (payments)
- Vercel (hosting)

Browser Support: Modern browsers (Chrome, Firefox, Safari, Edge) v100+

Hardware Requirements

For Development (Minimum):

- Processor: Intel i5 (8th gen) or equivalent
- RAM: 8 GB (16 GB recommended)
- Storage: 256 GB SSD
- Display: 1366×768
- Internet: 10 Mbps+

Recommended: i7+, 16 GB+ RAM (no GPU needed)

For Production: Vercel serverless + external PostgreSQL (e.g., Supabase); equivalent to 2 vCPU, 4 GB RAM for prototype scale.

These ensured accessibility on standard hardware and scalable cloud deployment within internship limits.

1.5 Feasibility Study

The feasibility study evaluates Hakim AI's viability as an MVP across technical, operational, economic, legal, and political dimensions. It confirms high feasibility for short-term development/deployment and strong long-term scalability/impact in Ethiopia's digital health landscape.

1.5.1 Technical Feasibility

Hakim AI uses mature, cost-effective, open-source technologies aligned with industry standards.

Stack: Next.js (frontend), Node.js/FastAPI (backend), PostgreSQL (storage), scikit-learn (AI), ZeGoCloud (video), Chapa (payments). These are well-documented, community-supported, and proven in similar platforms.

Random Forest model needs minimal resources (trainable on laptops/cloud). Vercel deployment provides auto-scaling, SSL, and easy management.

Short-term: Full MVP in 8–10 weeks with end-to-end features.

Long-term: Modular design supports upgrades (advanced AI, native apps, wearables, analytics) and horizontal scaling.

Technical risks are low; highly feasible now and in the future.

1.5.2 Operational Feasibility

Hakim AI complements existing workflows with minimal disruption.

Users: Familiar interfaces (like e-commerce apps) ease patient adoption. Doctors: Standard devices for registration, scheduling, and remote sessions. Admin: Simple dashboard for oversight.

High acceptance due to user-friendly design and stakeholder benefits.

1.5.3 Economic Feasibility

Strong feasibility via low costs and sustainable monetization.

Costs: Minimal (intern time, free/open-source tools, ~\$50–100/month cloud hosting). No licenses/hardware needed.

Revenue: 15% commission on consultations, premium subscriptions (49ETB/month).

Clear savings and revenue potential for all stakeholders.

1.5.4 Legal Feasibility

Operates within current Ethiopian frameworks with built-in safeguards.

Compliance: Secure data handling aligns with Personal Data Protection Proclamation (No. 1321/2023). Doctor verification (license + admin approval) ensures certified providers. Prominent disclaimers limit liability; digital prescriptions require pharmacy verification.

Short-term: Functions as private telemedicine facilitator . **Long-term:** Adaptable to evolving regulations (e.g., e-prescriptions, record sharing).

Minor risks (e.g., cross-regional practice) mitigated by geo-restrictions and terms of service.

1.5.5 Political Feasibility

High feasibility due to alignment with national priorities.

Supports FMOH Digital Health Blueprint (2021–2030), Information Revolution, and Digital Ethiopia 2025—focusing on equitable care, telemedicine, and data-driven systems. Addresses rural-urban disparities and universal coverage goals.

Short-term: Demonstrates contribution to digital transformation; potential support from authorities/partners (e.g., USAID). Long-term: Enables public-private partnerships and integration with eCHIS/DHIS2 for policy insights.

No anticipated opposition; empowers doctors, eases system strain, advances development agendas.

Conclusion: Hakim AI is highly feasible across all dimensions, delivering immediate access/efficiency gains and a solid foundation for long-term impact in Ethiopia's digital health transformation.

1.6 Risk Assessment Strategy

A systematic risk assessment was conducted to identify threats to the 10-13 week timeline. Risks were evaluated by likelihood, impact, and priority, with proactive mitigations and contingencies defined, especially for external factors.

NO.	Risk Description	Category	Likelihood	Impact	Priority	Mitigation Strategy
1	AI accuracy below expectations (public datasets not Ethiopia-specific)	Technical/AI	Medium	high	high	Use multiple datasets; early metrics (>75% accuracy); UI disclaimers
2	Third-party service issues (Chapa downtime/API changes)	external	medium	high	high	Early integration testing (week 6); sandbox use;
3	Team availability/coordination problems (illness, academics)	Operational	medium	medium	medium	Weekly meetings; shared GitHub issues; handover documentation
4	Data privacy/security concerns (sensitive data exposure)	Legal/Ethical	Low	high	medium	Anonymized test data; early authentication; no real patient data; secure practices
5	Scope creep (unplanned features e.g., multilingual support)	project	medium	high	high	Strict MoSCoW adherence; log new requests as future enhancements

Table 1 Risk Assesmnet

Risks were identified at project start, reviewed weekly, and updated in a shared register. High-priority risks received dedicated buffers (e.g., 2–3 days in integration/testing).

Proactive measures—especially for integrations and dependencies—enabled on-time MVP completion. Contingencies ensured resilience, keeping focus on core objectives.

1.7 Significance of the Project

The successful development and deployment of Hakim AI marks a significant advancement in health technology for resource-constrained settings like Ethiopia. It delivers a practical, AI-integrated telemedicine platform that addresses gaps in accessibility, efficiency, and equity.

By combining preliminary symptom assessment with affordable remote consultations, Hakim AI supports digital health transformation, early detection, and reduced burden on physical infrastructure. It aligns with FMOH's Digital Health Blueprint and demonstrates software engineering's potential to bridge geographical barriers and promote proactive care.

Long-term, the platform can evolve into a widely adopted tool that saves lives, lowers costs, and fosters digital wellness.

1.7.1 Beneficiaries of the Project

Hakim AI provides tangible benefits across stakeholders.

Federal Ministry of Health and Health Sector Stakeholders:

- Aligns with HSTP and Digital Health Blueprint (2021–2030) by advancing telemedicine and AI-enabled care.
- Cost-effective extension of reach to rural/underserved areas without major infrastructure investment.
- Generates anonymized data (usage, symptoms, trends) for policy, resource allocation, and interventions.
- Enhances reputation as digital health innovator; attracts partnerships (donors, NGOs, private sector).
- Reduces long-term expenditure via early detection and prevention of escalations.

System Users:

- **Patients:** 24/7 instant AI guidance without travel; saves time/money; reduces self-misdiagnosis risks. Telemedicine offers affordable consultations, digital prescriptions, and history tracking for better outcomes—especially benefiting rural, low-income, and urban users.

- **Doctors:** Flexible remote practice expansion; additional income (self-set fees/schedules); wider patient reach for health equity without relocating.
- **Administrators/Facilities:** Reduced overcrowding as mild cases shift digitally; frees resources for emergencies/complex care.

Developers (Nuredin Fentaw and Fuad Abdela):

- Hands-on full-stack experience with AI (scikit-learn), real-time communication , payments (Chapa), and frameworks (Next.js/FastAPI).
- Expertise in health-tech challenges: ethical AI, privacy, user-centered design.
- Skills in agile management, risk assessment, collaboration (GitHub, pair programming), and end-to-end delivery.
- Stronger employability in health-tech sector via real-world portfolio.
- Personal fulfillment from impactful contribution to national healthcare.

Overall, Hakim AI fosters a reinforcing ecosystem: empowered patients increase demand, engaged doctors ensure quality, authorities gain insights, and developers build expertise—creating a sustainable model for digital health innovation.

1.8 Project Schedule

Hakim AI was developed within an 8-week internship timeline (November 1 – December 20, 2025), following an Agile approach with iterative sprints and overlaps for parallel work.

Detailed Project Schedule

Phase	Task Description	Responsible	Duration	Start Date	End Date	Deliverables
1	Requirements gathering & research	Fuad	3 weeks	Nov 1 , 2025	Nov 21, 2025	Requirements document, wireframes, dataset selection
2	Database & backend setup (PostgreSQL schema, authentication, core REST APIs, admin basics)	Nuredin (lead), Fuad (support)	1 weeks	Nov 22 , 2025	Nov 29 , 2025	Functional backend, seeded database, API documentation (Postman)
3	AI model development (preprocessing, Random	Nuredin	2 weeks	Nov 30, 2025	Dec 14, 2025	Trained model ,assessment

	Forest training/evaluation, FastAPI exposure)					endpoint, accuracy report (>75%)
4	Frontend development (Next.js patient/doctor UIs, responsive design, Zustand state management)	Nuredin (lead), Fuad (support)	1 week	Dec 15, 2025	Dec 28 , 2025	Complete web interfaces, interactive prototype
5	Testing, bug fixing & deployment (unit/integration/end-to-end testing, security checks, Vercel deployment, documentation)	both	1 week	Dec 28, 2025	Jan 5, 2026	Deployed application, test reports, final documentation

Table 2 8 weeks (with overlaps and buffers to remain within 10-week maximum).

This schedule incorporated 2–3 days of buffer time in later phases to accommodate minor delays (e.g., integration issues with Chapa sandbox approval). Weekly progress reviews ensured adherence to milestones, and the overlapping phases maximized efficiency through parallel development—Nuredin focusing on AI/backend while Fuad led frontend/integrations.

The structured yet flexible scheduling enabled successful on-time completion of all core components, resulting in a fully deployed Hakim AI platform ready for pilot use by the internship deadline of December 20, 2025.

1.9 Project Budget Breakdown and Cost Analysis

Hakim AI was developed with a focus on cost-effectiveness during the 8-week internship (November 1 – Jan 5, 2026). Most tools were open-source or free-tier, keeping expenses minimal. Costs are in USD (\approx 150 ETB in 2025). The analysis covers internship development costs and post-launch operational costs for a deploying organization (e.g., FMOH or partner).

Budget Breakdown

Input (hardware or software used)	cost	quantity	Total cost	units
Pen for writing	25	2	50	ETB

Note book	100	2	200	ETB
mouse	450	2	900	ETB
internet	2,600	2	5,400	ETB
Expense (food, water, sanitation materials)	9,000	2	18,000	ETB
Time spent for the project	60,000	2	120,000	ETB
Effort	50000	2	100,000	ETB
Total			244,550	ETB

Table 3 budget breakdown

1.10 Team Composition

Hakim AI was developed by a two-member team of software engineering interns: Nuredin Fentaw and Fuad Abdela. The team combined complementary skills in full-stack development, AI, and integrations. Both members maintained comprehensive knowledge of the project, participating in joint reviews, alternative evaluations, and decision rationalization (e.g., technology choices like Zustand over Redux, FastAPI for AI, WebRTC primary over full Agora).

Team Members and Roles

- Nuredin Fentaw (AI and Backend Specialist) Background: Experienced in machine learning, data processing, and API development. Focal Responsibilities:
 - Led database & backend setup (Phase 2): PostgreSQL schema, core APIs, authentication.
 - Led AI model development (Phase 3): Data preprocessing, Random Forest training/evaluation, FastAPI exposure.
 - Supported frontend development and final testing/deployment.
 - Co-led requirements gathering and final documentation. Key Contributions: Proposed and evaluated AI algorithms (selected Random Forest for accuracy and interpretability); rationalized relational database choice; conducted code reviews for security and maintainability.
- Fuad Abdela (Frontend and Integration Specialist) Background: Skilled in UI/UX design, web applications, and real-time/third-party integrations. Focal Responsibilities:
 - Led requirements gathering & research (Phase 1): Requirements document, wireframes, dataset selection.
 - Led frontend development (Phase 4): Next.js patient/doctor portals, responsive design, Zustand state management.

- Supported backend/AI development and final testing/deployment.
- Co-led requirements and final documentation. Key Contributions: Evaluated UI frameworks (chose Next.js for PWA capabilities); optimized real-time integrations; improved usability through prototype testing and debugging.

Chapter Two: System Requirement Specification

2.1 Background (Overview)

Ethiopia's healthcare system faces major barriers: limited rural access (>80% population), long travel distances, urban overcrowding, specialist shortages (doctor-patient ratio ~1:10,000 vs. WHO's 1:1,000), and misinformation from unstructured self-diagnosis. These drive delayed diagnoses and inequities.

FMOH addresses this via Digital Health Blueprint (2021–2030) and Health Sector Transformation Plan, promoting telemedicine, AI, and data-driven care for universal health coverage. Recent efforts include hospital connectivity and telemedicine expansion to 200 facilities by 2025. Private platforms like TenaFirst offer consultations/pharmacy links, but lack integrated AI triage.

Globally, AI symptom checkers (Ada Health: ~70% top-3 accuracy, high urgency safety; Babylon, Buoy) improve triage and reduce visits but struggle with localized diseases, rare conditions, and low-literacy contexts—relevant to Ethiopia's limited internet and cultural factors.

Hakim AI fills this gap with a hybrid platform: AI preliminary assessments (structured inputs, severity classification, >75% severe-case recall target) plus verified telemedicine, using accessible web/PWA tech. It complements national initiatives with disclaimers and local alignment.

Sources: FMOH reports, WHO strategies, Ethiopian telemedicine studies, global benchmark evaluations (e.g., BMJ Open 2020).

2.1.1 Scope

Hakim AI's scope defines the boundaries for delivering a functional MVP AI-powered symptom checker and online doctor consultation platform within internship constraints.

Project Goals:

- Develop and deploy a web-based AI symptom assessments and remote consultations.
- Integrate sustainable monetization (fees, premium tiers).
- Improve accessibility, reduce misinformation, and complement physical facilities in Ethiopia.

Key Deliverables:

- Deployed Vercel web/PWA application.
- AI symptom checker (probable conditions, severity, recommendations).
- Telemedicine: doctor verification, booking, video/chat, prescriptions, history storage.
- Admin dashboard (approval, analytics, payouts).
- Payment integration and tiered access.
- Full documentation, code repository, presentation.

Major Tasks:

- Requirements, design, prototyping.
- Backend (APIs, database, authentication).
- AI training/integration.
- Frontend UI/UX.
- Integrations (video, payments).
- Testing and deployment.
- Documentation/risk management.

Boundaries:

- **In-Scope:** Web/PWA, basic Random Forest AI (structured inputs), core integrations.
- **Out-of-Scope:** Native apps, offline mode, advanced AI (NLP/image), national system/pharmacy/insurance integration, clinical trials, certification, marketing.

This focused scope ensures feasible MVP delivery with verifiable goals.

2.1.2 Purpose

This System Requirement Specification (SRS) defines software requirements for Hakim AI Version 1.0

The product is a standalone full-stack web application providing:

- Instant AI preliminary assessments from symptom inputs.
- Secure paid remote consultations with verified Ethiopian doctors (video/chat, prescriptions).
- Appointment scheduling, doctor verification, monetization (fees + premium subscriptions).

Covered Components:

- Patient features (symptom input, assessments, booking, consultations).
- Doctor portal (registration, schedules, sessions).
- Admin dashboard (verification, analytics, payouts).
- Backend (APIs, database, authentication).
- AI integration.
- Third-party services (video, payments).

No separate subsystems or larger systems are included; future enhancements (native apps, advanced AI, national integrations, pharmacy links) are out of scope.

Product Purpose:

- Enable early symptom evaluation to reduce misinformation/delays.
- Bridge geographical barriers for rural/underserved patients.
- Alleviate facility overcrowding via digital triage.
- Provide doctors flexible income.
- Support FMOH digital health transformation for equity and efficiency.

Meeting these requirements delivers a scalable foundation for improved healthcare access in Ethiopia.

2.1.3 Document Convention

Indicators	Most common and recommended
Alignment	Justified
Margin	Left=3cm, right=2.5, top=2.5, bottm=2.5

Title font size /heading 1 e.g Chapter One	16pt ,bold
Sub Title font size /heading 2 e.g 1.1Introduction	14pt,bold
Sub Title font size /heading 3 e.g 1.1.1 Introduction	12pt
Whole document / Normal text Font size	12
Font style	Regular
Font type	Time New Roman
Page Color	White
Language	English
Line between	1.5
Correction with fluid	Not allowed
Typing machine	Computer
Crossing out words	Not allowed
Printing quality	Laser or later quality
Font color	Black

Table 4 Document Convention

This SRS for Hakim AI Version 1.0 is intended for readers involved in evaluation, development, adoption, or extension of the project, accommodating varying technical and domain expertise.

2.1.4 Intended Audience and Suggested Readings

- Project Assessors/Examiners: Academic supervisors, internship coordinators, and examiners evaluating quality, standards compliance, and healthcare alignment.
- Developers/Future Maintainers: Engineers extending or maintaining the system (e.g., adding native apps, advanced AI, integrations).
- Project Managers/Stakeholders: FMOH representatives, health-tech organizations, or sponsors assessing scope, feasibility, and national strategy alignment.
- Testers/QA Personnel: Individuals verifying functional/non-functional requirements via testing.
- Health Domain Experts/Policy Makers: Healthcare professionals, FMOH officials, or NGOs evaluating clinical relevance, ethics, and impact.
- Documentation Writers: Those creating user manuals, API docs, or training materials from requirements.

Document Organization: The SRS (Chapter 2) covers background, scope, purpose, elicitation, and functional/non-functional requirements. Subsequent chapters address design, implementation, testing, impact/limitations, and future work.

2.1.5 List of Acronyms, Abbreviations and Definitions

2.2 Overall Description of Software Requirements

Hakim AI Version 1.0 requirements are tailored to Ethiopia's 2025 healthcare context: limited rural access, low doctor-patient ratios, >50% smartphone penetration, variable 3G/4G connectivity, and FMOH digital health priorities. They emphasize practicality, affordability, and rapid deployment over feature excess.

Requirements deliver competitive advantages:

- Cost: Free/open-source tools (Vercel, scikit-learn, Chapa); development <\$1,200; operational costs revenue-scaled (15% commissions).
- Quality: Measurable targets (>75% AI accuracy, disclaimers, verification, encryption) for safety/trust.
- Timing: MVP-focused for 13-week Agile delivery and immediate pilot.

Structured inputs (not NLP) ensure low-bandwidth performance (<5s assessments); minimizes SDK costs. Requirements directly address user needs: instant guidance for patients, flexible income for doctors, oversight for admins.

This lean approach positions Hakim AI as an accessible, affordable entrant—faster/cheaper than global platforms, aligned with national goals, and scalable.

2.2.1 Product Perspectives

Hakim AI Version 1.0 is a new, standalone web-based platform developed for Ethiopia's healthcare gaps, not a replacement or extension of existing systems.

It addresses geographical barriers, overcrowding, specialist shortages, and misinformation risks, supporting FMOH Digital Health Blueprint (2021–2030).

No interfaces with national systems (DHIS2, eCHIS, EMRs). External interfaces:

- Users via browsers/mobile PWA.
- Chapa (payments).
- Vercel (hosting).

2.2.2 Product Features/Functions

Patients:

- Structured symptom input → instant AI report (top-3 conditions/probabilities, severity, recommendations) with disclaimers.
- Search/book verified doctors; instant scheduling.
- Secure video/chat consultations; digital prescriptions.
- View/store medical history.
- Free tier (limited assessments); premium (unlimited + priority).

Doctors:

- Registration with license upload/verification.
- Manage availability/schedules; receive booking notifications.
- Conduct video/chat sessions; access patient history/AI reports; issue prescriptions.
- Set fees; receive payouts (minus 15% commission).

Administrators:

- Verify doctors; manage users.
- Monitor analytics (usage, revenue, trends).
- Handle payouts and system oversight.

These MVP features deliver end-to-end value: accessibility for patients, flexibility for doctors, governance for admins.

2.2.3 User Characteristics

Three user classes, with patients most critical for impact.

1. Patients (Primary – Most Important)

- Frequency: High/moderate.
- Functions: Symptom assessment, booking, consultations, history.
- Expertise: Low–moderate (basic smartphone literacy).
- Education: Varied (low rural literacy to urban higher).
- Privileges: Own data only.
- Characteristics: rural/urban; cost/language-sensitive. Prioritized: intuitive UI, low-data performance, disclaimers.

2. Doctors (Highly Important)

- Frequency: Moderate.
- Functions: Registration, scheduling, consultations, prescriptions, revenue.
- Expertise: Moderate–high (familiar with digital tools).
- Education: High (licensed professionals).
- Privileges: Patient data for own consultations.
- Characteristics: Urban-based physicians seeking remote income/flexibility. Prioritized: reliable video, easy scheduling, secure payouts.

3. Administrators (Important – Less Frequent)

- Frequency: Low.
- Functions: Verification, user management, analytics, payouts.
- Expertise: Moderate (dashboard experience).
- Education: Moderate–high.
- Privileges: Full system access.
- Characteristics: Small team (1–3) for governance/compliance. Prioritized: comprehensive dashboards.

Requirements are role-specific (e.g., simple forms/disclaimers for patients; verification/payouts for doctors) to maximize adoption and effectiveness across classes.

2.3 General Constraints

Hakim AI Version 1.0 is constrained by internship timeline/resources, Ethiopia's variable connectivity/diverse devices, and need for cost-effective sustainability. These ensure lightweight, accessible design with ethical focus.

2.3.1 Software Constraints

Operating Environment:

- **Client-Side:** Modern browsers (Chrome/Firefox/Safari/Edge, last 5 years, ES6+ support); mobile-responsive PWA. Minimum: 2 GB RAM devices, Android 8+/iOS 12+, 3G/4G/Wi-Fi (low-bandwidth optimized).
- **Server-Side:** Vercel serverless; Node.js/FastAPI APIs; PostgreSQL (e.g., Supabase).

Design/Implementation Constraints:

- Open-source/free-tier only (Next.js, Node.js/FastAPI, scikit-learn, PostgreSQL, Better-Auth, ZeGoCloud, Chapa).
- AI: Lightweight Random Forest on public datasets.
- Integrations: Only Chapa and ZeGoCloud
- Language: English primary

2.3.2 Hardware Constraints

Hakim AI requires minimal hardware for accessibility in Ethiopia.

Development:

- Standard laptops (i5/equivalent, 8 GB RAM, SSD); personal devices used.
- No GPUs/specialized hardware (CPU-based AI).
- Stable internet assumed for testing/deployment.

Deployment/User:

- **Server:** Vercel serverless (no dedicated hardware).
- **Client:** Low-end smartphones/computers (2 GB RAM, webcam/mic); video optimized for 512 kbps (chat fallback).

2.3.3 Assumptions and Dependencies

Assumptions:

- Users have smartphones/browsers + \geq 3G connectivity (core features functional on low bandwidth).
- AI achieves $>75\%$ accuracy + $>90\%$ severe recall; users heed disclaimers.

- Sufficient doctor participation via incentives.
- Current laws allow private telemedicine with verification/disclaimers.
- Basic digital literacy; accurate symptom reporting.
- Premium uptake supports sustainability.

Dependencies:

- **Third-Party:** Vercel (hosting/limits), Chapa (payments/reliability), open-source libraries (stability).
- Public datasets for AI.
- Stable intern internet/GitHub/Vercel access.
- Post-deployment: Organization oversight/funding if tiers exceeded.

2.3.4 User Documentation

Documentation is simple, accessible (English) for varying literacy.

Types:

- Patient Guide: Account setup, symptom input, AI reports/disclaimers, booking, consultations, history, premium.
- Doctor Guide: Registration/license, scheduling, consultations, prescriptions, earnings.
- Admin Guide: Verification, management, analytics, payouts.
- FAQ/Troubleshooting.
- Disclaimers/Terms of Use.

Distribution:

- In-app: Tooltips, guided tours, searchable Help/FAQ.
- Downloadable PDFs from dashboard/website.
- Automated announcement emails with links.

2.4 Specific Requirements

Specific requirements provide actionable guidance for development, categorized into user, functional, and non-functional. They ensure comprehensive coverage, traceability to Ethiopian healthcare needs, and verifiability.

2.4.1 User Requirements

User requirements capture high-level needs from end-users and stakeholders, elicited via research, consultations, and benchmarking. Phrased from user perspective; prioritized as Essential/Desirable.

ID	User Requirements Statement	Stakeholder	Priority	Rationale/Need
UR1	As a patient, I need an easy way to describe symptoms and get quick preliminary guidance without traveling.	Patients	Essential	Addresses travel barriers, delays, misinformation.
UR2	As a patient, I need clear information that AI advice is not a final diagnosis.	Patients	Essential	Ensures safe use; ethical requirement.
UR3	As a patient, I need affordable remote doctor access via video/chat for advice/prescriptions.	Patients	Essential	Reduces barriers/costs; decongests facilities.
UR4	As a patient, I need free basic features and optional paid unlimited/priority access.	Patients	Essential	Promotes equity and reach.
UR5	As a patient, I need secure storage/review of past assessments/consultations.	Patients	Essential	Improves continuity/trust.
UR6	As a doctor, I need simple registration, verification, and availability setting.	Doctors	Essential	Encourages participation.
UR7	As a doctor, I need fair income with transparent fees/timely payouts.	Doctors	Essential	Motivates regular use; sustainability.
UR8	As a doctor, I need reliable video/chat and patient history access.	Doctors	Essential	Enables quality remote care.
UR9	As an administrator, I need oversight for doctor verification, usage monitoring, revenue.	System Owners	Essential	Ensures quality/sustainability.

Table 5 User Requirements

2.4.2 Functional Requirements

ID	Functional Requirements	Priority	Verification Criteria	Error Handling/ Invalid Inputs

FR-1	Structured symptom input form (multiple symptoms, pain level, duration, body area).	High	≥50 symptoms; valid submission to AI.	Inline errors; prevent incomplete submission.
FR-2	Generate/display AI report: top-3 conditions/probabilities, severity, recommendations.	High	<5s generation; logical output.	Error message on failure; prompt doctor consult.
FR-3	Mandatory prominent disclaimer on reports with user acknowledgment.	High	Visible 100%; checkbox required.	Block view until acknowledged.
FR-4	Search/book doctors by specialty/availability; instant slot reservation.	High	<3s search; confirmed booking.	"Slot unavailable" on conflict; concurrency lock.
FR-5	Secure video/chat sessions, encrypted.	High	<10s connect; functional media.	Auto chat fallback; reconnect on drop.
FR-6	Doctors issue/save digital prescriptions (medication, dosage, instructions).	High	Saved/viewable post-session.	Validate fields; prevent incomplete save.
FR-7	Doctor registration with license, admin approval workflow.	High	Inactive until approved; secure storage.	Reject invalid files; notify rejection.
FR-8	Manage doctor schedules	High	Instant reflection in booking view.	Warn/prevent overlaps.
FR-9	Process payments via Chapa; calculate/deduct 15% commission.	High	Success confirmation; accurate records.	"Payment failed" retry; no charge on error.
FR-10	Tiered access: free (≤ 10 reports/month); premium (unlimited + priority).	High	Accurate counters; features unlocked post-pay.	Prompt upgrade on limit; block excess reports.
FR-11	Secure storage/display of patient history (reports, notes, prescriptions).	High	<5s load; authorized access only.	"Unavailable" on issue; ensure backups.
FR-12	Admin tools: verify doctors, manage users, analytics, payouts.	High	Logged actions; real-time accurate data.	Role checks; confirm destructive actions.
FR-13	Notifications (email/SMS) for bookings, reminders, receipts, decisions.	High	<1 min delivery.	Log/retry failures; in-app fallback.

Table 6 Functional Requirements

2.4.3 Non-Functional Requirements

ID	Non-Functional Requirements	Priority	Verification Criteria	Error Handling/ Invalid Inputs
NFR-1	Security: Encryption (rest/transit), JWT	High	No critical vulnerabilities; audit	Health data protection;

	auth, RBAC; sensitive data restricted.		logs.	legal/ethical.
NFR -2	Performance: AI <5s, pages <3s, APIs <1s (95th percentile).	High	Simulated 3G testing.	Usability on variable Ethiopian networks.
NFR -3	Usability: Intuitive UI; English	High	>90% task completion.	Diverse literacy levels.
NFR -4	Reliability: >99% availability; graceful failure recovery.	High	Uptime monitoring; reconnect attempts.	Critical health tool.
NFR -5	Scalability: ≥1,000 concurrent users.	High	Vercel stress testing.	Growth preparation.
NFR -6	Maintainability: Modular, documented, testable code.	MEDIU M	>70% coverage; complete API docs.	Long-term evolution.
NFR -7	Portability: Consistent across major browsers; PWA on mobile.	High	Cross-browser testing; installable PWA.	Broad device reach.
NFR -8	Safety: Mandatory disclaimers, escalation prompts, no emergency overrides.	High	100% enforcement; severe prompts booking.	Prevent misinterpretation/harm.

Table 7 Non-Functional requirements

2.5 External Interface Requirements

2.5.1 User Interfaces

The Healthcare AI Platform (Hakim AI) will be a web-based application providing a comprehensive, user-friendly interface designed for intuitive navigation across multiple healthcare stakeholders. The interface is built using modern web technologies (Next.js, React, Tailwind CSS) ensuring responsive design and minimal learning curve for users.

The system provides role-based personalized views for three distinct user types: Administrators, Doctors, and Patients. Each role has access to specialized features tailored to their workflow, including AI-powered symptom analysis, appointment scheduling, real-time video consultations, prescription management, and healthcare analytics.

Each page maintains a consistent layout with a persistent sidebar navigation that includes links to dashboard, appointments, consultations, AI chatbot, prescriptions, notifications, and user profile settings. All interfaces implement real-time validation for user input with contextual error messages and success notifications using toast alerts. The platform supports both light and dark themes with smooth transitions and modern glassmorphism design patterns.

Patient Portal Interface

The patient portal provides comprehensive healthcare management tools with the following key interfaces:

- Patient Dashboard Interface: Displays an overview of upcoming appointments, recent AI symptom analyses, active prescriptions, available doctors by specialty, and health notifications. Features quick-access cards for booking consultations, accessing the AI chatbot, and viewing medical history.
- AI Symptom Analysis Interface: Interactive symptom checker allowing patients to input multiple symptoms through a multi-select interface. The system processes symptoms through machine learning models and provides AI-generated medical explanations in Amharic language, including potential conditions, warning signs, self-care recommendations, and when to seek urgent care.
- Doctor Consultation Interface: Browse and filter doctors by specialty category, view doctor profiles with ratings, experience, and availability. Supports two consultation types: video calls and in-hospital visits. Displays doctor status indicators (Online, Offline, Busy) in real-time.
- Appointment Management Interface: Calendar-based appointment booking system showing doctor availability, appointment history with status tracking (Pending, Approved, Cancelled, Completed), and automated email reminders (24 hours, 1 hour, and 10 minutes before appointments).
- Video Conference Interface: Integrated ZegoCloud video calling platform for real-time doctor consultations with screen sharing, chat, and recording capabilities.
- Real-time Chat Interface: Direct messaging system with doctors, displaying conversation history, unread message indicators, and typing status indicators.
- Prescription Management Interface: View all prescriptions issued by doctors including symptoms, diagnosis, prescribed medications, usage instructions, and prescription date.
- AI Chatbot Interface: Conversational AI assistant providing 24/7 health information, answering medical questions, and guiding patients through the platform features.
- Hospital Locator Interface: Interactive map (Leaflet) showing nearby hospitals and healthcare facilities with location markers and directions.
- Subscription Packages Interface: Display available healthcare packages with pricing, features comparison, and subscription management.

Doctor Portal Interface

The doctor portal provides professional healthcare delivery tools with the following interfaces

- Doctor Dashboard Interface: Comprehensive analytics showing total consultations, earnings overview, appointment statistics, patient ratings, and upcoming schedule. Features revenue charts, appointment trends, and performance metrics.
- Profile Completion Interface: Multi-step form for doctors to complete their professional profile including bio, license number, specialization, years of experience, category selection, profile picture upload, and consultation pricing. Mandatory completion before accessing other portal features.
- Appointment Management Interface: View all appointments (pending, approved, completed) with patient details, appointment type (video/hospital), date/time, and action buttons for approval/cancellation. Includes appointment link generation for video consultations.
- Patient Consultation Interface: Access patient medical history, previous symptom logs, and AI analysis results during consultations. Create and issue digital prescriptions with symptoms, diagnosis, medication, and usage instructions.
- Real-time Chat Interface: Communicate with patients through secure messaging with conversation threading and message history.
- Blog Management Interface: Create and publish medical articles/blogs with rich text editor, title, content, and publication date tracking.
- Earnings & Withdrawal Interface: View consultation earnings, revenue breakdown by appointment type, and submit withdrawal requests with bank account details (bank name, account number, account name). Track withdrawal status (Pending, Approved, Rejected).
- Status Management Interface: Toggle availability status (Online, Offline, Busy) to control patient visibility and appointment bookings.

Admin Panel Interface

Administrators have access to comprehensive platform management tools:

- Admin Dashboard Interface: System-wide analytics displaying total users (patients + doctors), approved vs. pending doctor applications, total appointments, global revenue, monthly growth charts (line graphs), and recent user registrations with role badges.
- User Management Interface: Data table with sorting, filtering, and pagination showing all users with details (name, email, role, status, registration date). Actions include viewing user details, status updates (Active/Inactive), and user deletion.

- Doctor Approval Interface: Review pending doctor applications with profile details, license verification, and approve/reject actions. Toggle doctor approval status affecting their visibility to patients.
- Doctor Management Interface: Comprehensive table of all doctors with specialization, experience, approval status, ratings, and management actions.
- Withdrawal Management Interface: Review doctor withdrawal requests showing amount, bank details, request date, and approval/rejection actions with status tracking.
- Analytics & Reporting Interface: Advanced analytics with customizable date ranges, export functionality, user growth trends, appointment statistics, revenue reports, and platform performance metrics.
- Category Management Interface: Create, edit, and delete medical specialization categories used for doctor classification and patient search filters.
- Blog Moderation Interface: Review, approve, and manage doctor-submitted blogs before publication.
- Notification Broadcasting Interface: Send system-wide notifications or targeted messages to specific user roles.

Notification System Interface

All user roles have access to a unified notification interface:

- Real-time notification dropdown with unread count badges
- Notification types include: appointment confirmations, reminders, doctor approvals, prescription updates, chat messages, and system announcements
- Mark as read/unread functionality
- Notification history with timestamps and categorization

Design & Accessibility Standards

All interfaces adhere to the following standards:

- Responsive Design: Mobile-first approach with breakpoints for tablets and desktops
- Accessibility: WCAG 2.1 Level AA compliance with keyboard navigation, screen reader support, and proper ARIA labels
- Loading States: Skeleton screens and loading indicators for asynchronous operations
- Error Handling: User-friendly error messages with recovery suggestions

- Validation: Real-time form validation with inline error messages
- Consistency: Unified color scheme (indigo primary, emerald success, red error), typography (Geist Sans), and component library (shadcn/ui + Radix UI)
- Performance: Optimized images, lazy loading, and code splitting for fast page loads
- Internationalization: Support for Amharic language in AI-generated content

2.5.2 Hardware Interfaces

Hakim AI is purely web-based with no specialized hardware dependencies.

Client-Side:

- Display, touchscreen/keyboard/mouse for UI.
- Camera/microphone for video (via browser getUserMedia; permission required).
- Speakers for audio.
- Fallback to chat if hardware denied/unavailable.

Network: Standard internet (Wi-Fi/mobile data).

Server-Side: Vercel cloud (no managed hardware).

Minimum: Mid-range smartphones/computers with camera/mic. Hardware-agnostic design maximizes accessibility in Ethiopia.

2.5.3 Software Interfaces

The system will rely on several software components to provide a seamless user experience. These components include the backend, frontend, and third-party integrations.

- **Backend Framework:** Express.js and FastApi for server-side operations.
- **Frontend Framework:** Next.js for dynamic, interactions.
- **Database:** PostgreSQL to store all project and task data.
- **Web Browser Support:** The system will be compatible with modern browsers like Google Chrome, Mozilla Firefox, Brave, and Microsoft Edge.

The communication between the client and the server will be asynchronous, ensuring smooth operation even under high user load.

- **Hosting Service:** The Hakim Ai will be hosted on a cloud service that
- supports the necessary technologies(Vercel),

2.5.4 Communication Interfaces

The system will communicate over HTTPS to ensure secure data transmission between users and the server. The primary communication protocol will be HTTP/HTTPS for all web interactions, with real time updates being pushed through Web Sockets using Pusher ..

2.6 System Requirements Modeling

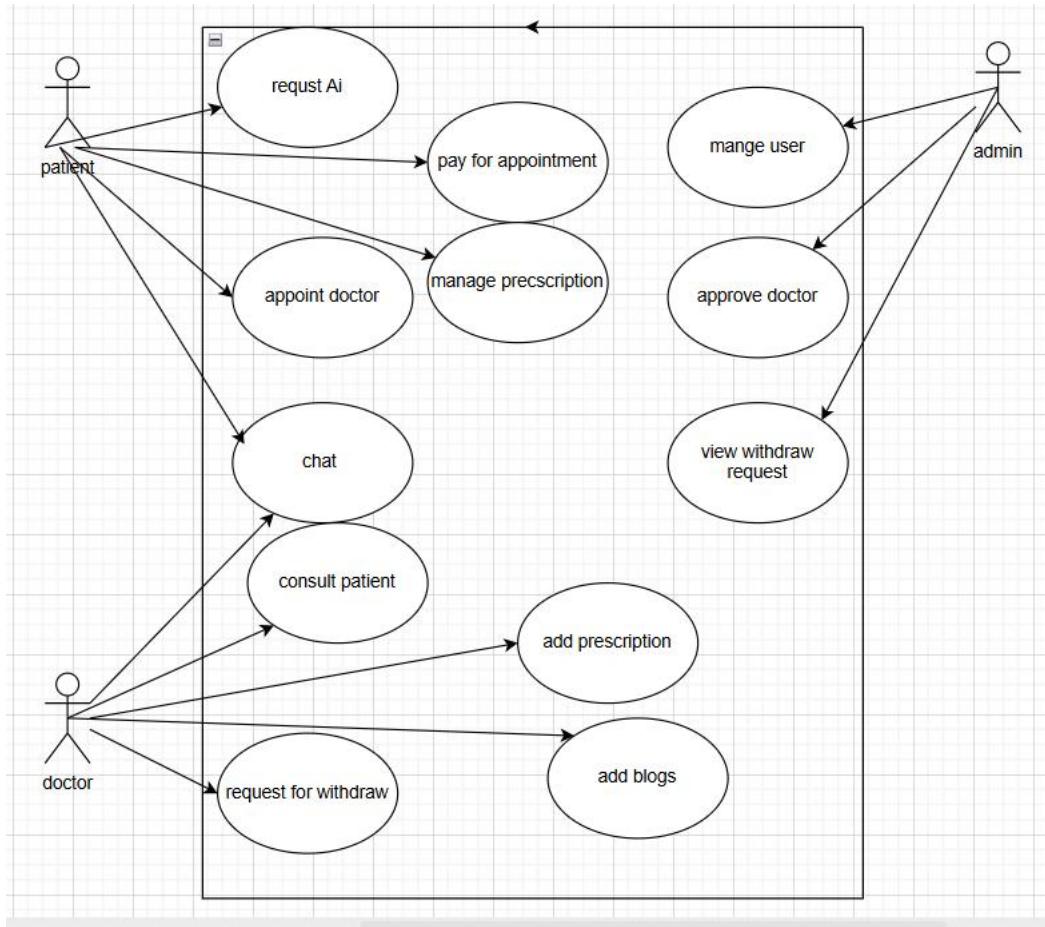
2.7 Essential Use Case Diagrams

Essential use cases for **Hakim AI Version 1.0** provide a simplified, abstract, technology-free description of core user-system interactions, expressed in the language of the application domain (healthcare access and telemedicine). These use cases focus on user intentions and system responsibilities, omitting implementation details (e.g., specific UI elements, APIs, or databases). They represent generalized, meaningful tasks that are complete and well-defined, emphasizing goals such as obtaining preliminary health guidance, connecting with doctors, and managing remote care.

The essential use case diagram abstracts the system's key functionalities and actors, highlighting dependencies (e.g., <>include<> for mandatory sub-tasks, <>extend<> for optional variations).

Actors (in Domain Language):

- **Patient:** An individual seeking health guidance or professional consultation.
- **Doctor:** A certified healthcare provider offering remote advice.
- **Administrator:** A system overseer ensuring quality and operations.

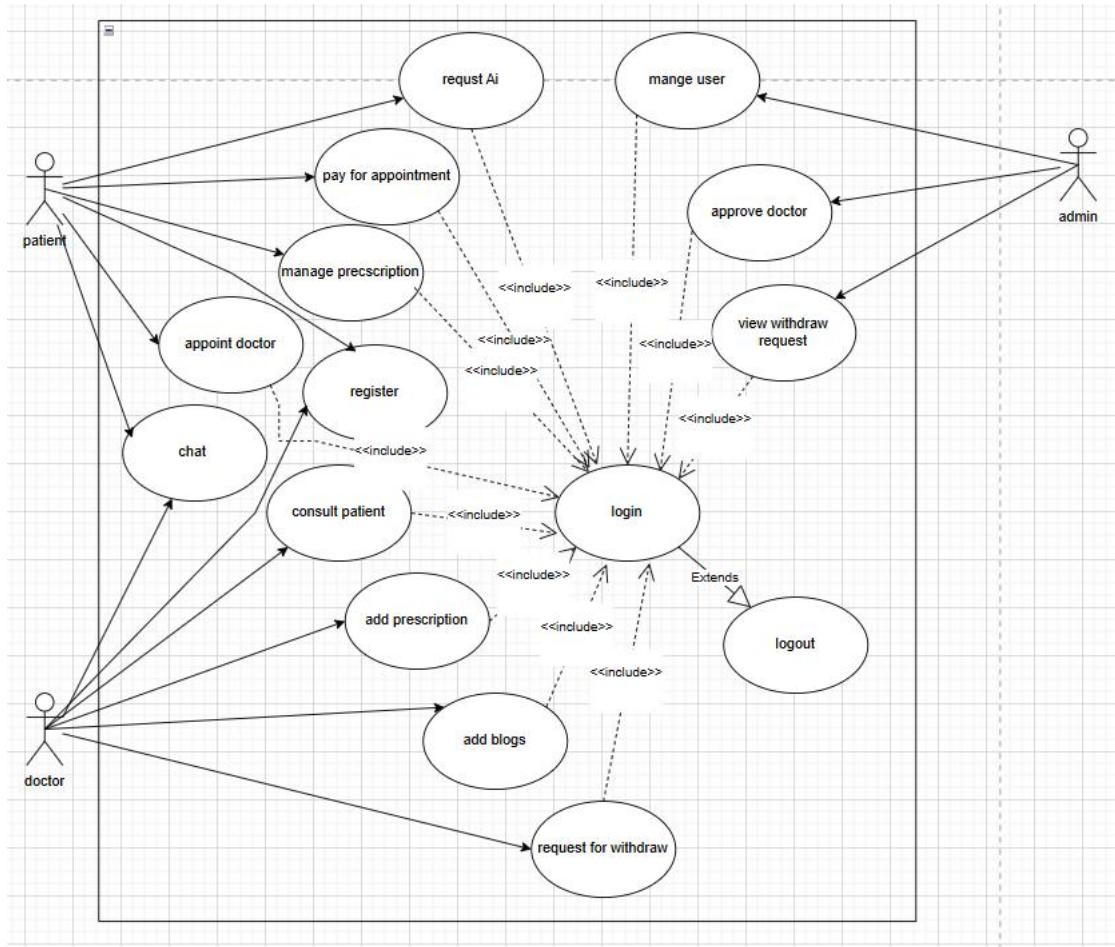


Chapter Three: Requirement Analysis Modeling

3.1 Overview of Analysis Model

The requirements analysis phase of the Gizebet Project Management Platform elaborates on the basic requirements identified during the requirements elicitation. This phase forms the first technical representation of the system and utilizes various modeling approaches to ensure that the system's requirements are fully understood and documented. The models created during this phase lay the foundation for the design and implementation phases.

3.2 System Use Case Diagram



3.2.1 Use Case Descriptions

Authentication Use Cases

Use Case 1: Register

Attribute	Details
Use Case Name	Register
Use Case ID	UC-001
Actor	Patient, Doctor
Description	New users can create an account to access the Healthcare AI Platform with role-based access.
Precondition	User does not have an existing account with the provided email address.
Post Condition	User account is created in the system. Doctor users are redirected to profile completion. Patient users can access the patient portal immediately.

Basic Course of action

User Action

1. User navigates to the platform homepage.
2. User clicks on the "Register" link.
3. User fills in registration details (Full Name, Email, Password, Phone, Role selection: Patient/Doctor).
4. User clicks the "Register" button.
5. Use case ends.

System Response

- 1.1 The system displays the landing page with navigation options.
- 2.1 The system displays the registration form.
- 3.1 The system validates input fields in real-time (email format, password strength, required fields).
- 4.1 The system verifies email uniqueness.
- 4.2 The system hashes the password using bcrypt.
- 4.3 The system creates a User record in the database.
- 4.4 The system creates a corresponding Patient or Doctor profile based on role selection.
- 4.5 The system creates a session token.
- 4.6 For Doctors: System redirects to profile completion page.
- 4.7 For Patients: System redirects to patient dashboard.
- 5.1 Success notification is displayed.

Use case 2: Login

Attribute Details

Use Case Name Login

Use Case ID UC-002

Actor Patient, Doctor, Admin

Description Users with valid credentials can log in to access the system based on their assigned role.

Precondition User must have a registered account in the system.

Post Condition User is authenticated and redirected to their role-specific dashboard. Session is created.

Basic course of action

User Action

1. User navigates to the platform homepage.
2. User clicks on the "Login" link in the navigation.
3. User enters their email and password.
4. User clicks the "Login" button.

System Response

- 1.1 The system displays the landing page.
- 2.1 The system displays the login page with email and password fields.
- 3.1 The system validates input format.
- 4.1 The system retrieves user record by email.

User Action	System Response
	4.2 The system verifies password using bcrypt comparison.
	4.3 The system creates a session token and stores it in the database.
	4.4 The system sets session cookie in the browser.
	4.5 Based on user role, system redirects:
	- Admin → Admin Dashboard
	- Doctor → Doctor Dashboard (if profile complete) or Profile Completion
	- Patient → Patient Dashboard
5. Use case ends.	5.1 Success notification is displayed.

Patient Use cases

Use case 3: Request Ai Symptom analysis

Attribute	Details
Use Case Name	Request AI
Use Case ID	UC-003
Actor	Patient
Description	Patients can input their symptoms and receive AI-powered disease predictions with detailed medical explanations in Amharic.
Precondition	Patient must be logged in.
Post Condition	AI analysis is generated and saved to patient's symptom log history.

User Action	System Response
1. Patient navigates to AI Chatbot section.	1.1 The system displays the AI symptom checker interface.
2. Patient selects multiple symptoms from the symptom list (multi-select interface).	2.1 The system displays selected symptoms with visual indicators.
3. Patient clicks "Analyze Symptoms" button.	3.1 The system sends symptoms to FastAPI service . 3.2 FastAPI service concatenates symptoms into a single string. 3.3 The ML model (scikit-learn) vectorizes symptoms using TF-IDF. 3.4 The model predicts the most likely disease. 3.5 The system sends prediction to Google Gemini API with a prompt requesting Amharic explanation. 3.6 Gemini generates detailed medical advice including: - Acknowledgment of symptoms

- Explanation of predicted condition
 - Warning signs to monitor
 - When to seek urgent care
 - Self-care recommendations
- 3.7 The system saves the symptom log to the database (SymptomLog table).
- 3.8 The system displays the AI-generated response in a chat bubble format.
- 4.1 The system provides options to book a doctor consultation or save the report.
4. Patient reads the AI explanation.
5. Use case ends.

Use case 4 : Appoint Doctor

Attribute	Details
Use Case Name	Appoint Doctor
Use Case ID	UC-004
Actor	Patient
Description	Patients can browse doctors by specialty, view profiles, and book appointments for video consultations or hospital visits.
Precondition	Patient must be logged in. At least one approved doctor must exist in the system.
Post Condition	Appointment is created with PENDING status. Doctor receives notification. Automated reminder emails are scheduled.

User Action	System Response
<ol style="list-style-type: none"> 1. Patient navigates to "Consult Doctor" section. 2. Patient filters doctors by specialty category. 3. Patient clicks on a doctor's profile card. 	<ol style="list-style-type: none"> 1.1 The system displays a list of approved doctors with filters (specialty, rating, availability). 2.1 The system queries doctors by categoryId and displays filtered results. 3.1 The system displays detailed doctor profile including: <ul style="list-style-type: none"> - Bio, specialization, experience - License number - Average rating and total reviews - Consultation price - Current status (Online/Offline/Busy)

Attribute	Details
4. Patient clicks "Book Appointment" button.	4.1 The system displays appointment booking form.
5. Patient selects appointment type (Video Call or Hospital Visit).	5.1 The system updates form fields based on selection.
6. Patient selects date and time.	6.1 The system validates doctor availability for selected time slot.
7. Patient clicks "Confirm Appointment" button.	7.1 The system creates Appointment record with status PENDING. 7.2 The system calculates appointment cost based on doctor's price. 7.3 For video appointments: System generates unique appointment link (ZegoCloud). 7.4 The system schedules cron job reminders (24h, 1h, 10min before appointment). 7.5 The system sends notification to the doctor. 7.6 The system redirects patient to appointment confirmation page.
8. Use case ends.	8.1 Success notification: "Appointment request sent. Waiting for doctor approval."

Use case 5: Pay for Appointment

Attribute	Details
Use Case Name	Pay for Appointment
Use Case ID	UC-005
Actor	Patient
Description	Patients can make payments for approved appointments to confirm their consultation.
Precondition	Patient must have an appointment with APPROVED status.
Post Condition	Payment is recorded. Appointment is confirmed. Doctor receives payment notification.

User Action	System Response
1. Patient navigates to "My Appointments" section.	1.1 The system displays list of appointments with status indicators.
2. Patient selects an approved appointment.	2.1 The system displays appointment details and payment button.
3. Patient clicks "Pay Now" button.	3.1 The system displays payment form with amount and payment methods.
4. Patient selects payment method and enters payment details.	4.1 The system validates payment information.
5. Patient confirms payment.	5.1 The system processes payment through payment gateway.

- 5.2 The system creates Payment record with status PAID.
- 5.3 The system updates appointment status to CONFIRMED.
- 5.4 The system updates doctor's earnings balance.
- 5.5 The system sends confirmation email to patient with appointment details.
- 5.6 The system sends payment notification to doctor.
- 6.1 Success notification: "Payment successful. Your appointment is confirmed."
6. Use case ends.

Use case 6: Consult Patient (Video call)

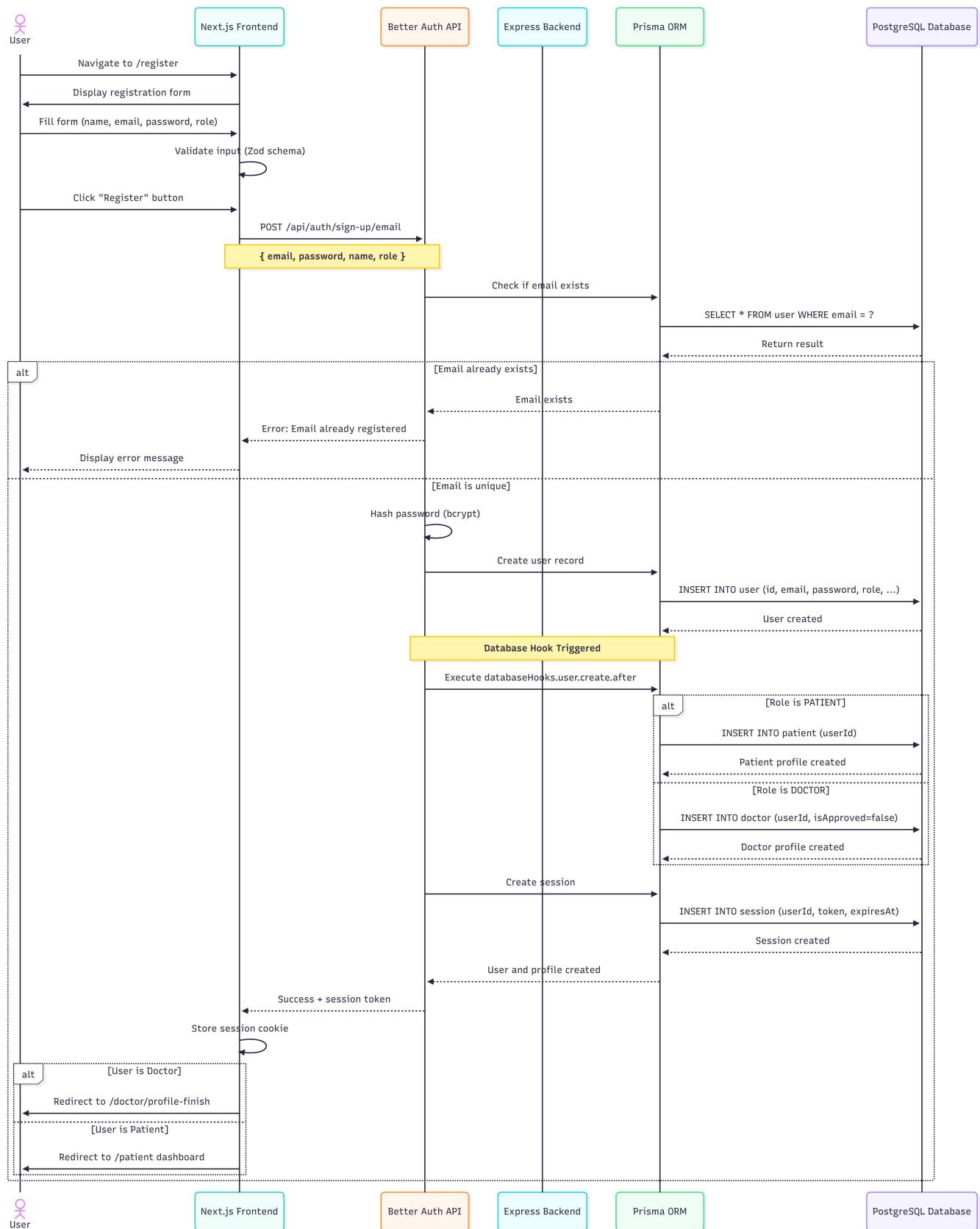
Attribute	Details
Use Case Name	Consult Patient
Use Case ID	UC-006
Actor	Patient, Doctor
Description	Patients and doctors can conduct real-time video consultations using integrated ZegoCloud video conferencing.
Precondition	Appointment must be APPROVED and PAID. Appointment time must be within valid window. Both parties must be logged in.
Post Condition	Video consultation is completed. Appointment status is updated to COMPLETED.

User Action	System Response
1. Patient navigates to "My Appointments" at scheduled time.	1.1 The system displays appointment with "Join Video Call" button.
2. Patient clicks "Join Video Call" button.	2.1 The system validates appointment time (within 15 minutes of scheduled time). 2.2 The system initializes ZegoCloud SDK with appointment link. 2.3 The system requests camera and microphone permissions. 3.1 The system connects to ZegoCloud video room. 3.2 The system displays video interface with patient's camera feed. 3.3 The system shows "Waiting for doctor..." message.
3. Patient grants permissions.	
4. Doctor joins the video call from their portal.	4.1 The system establishes peer-to-peer video connection. 4.2 The system displays both video feeds (patient and doctor).

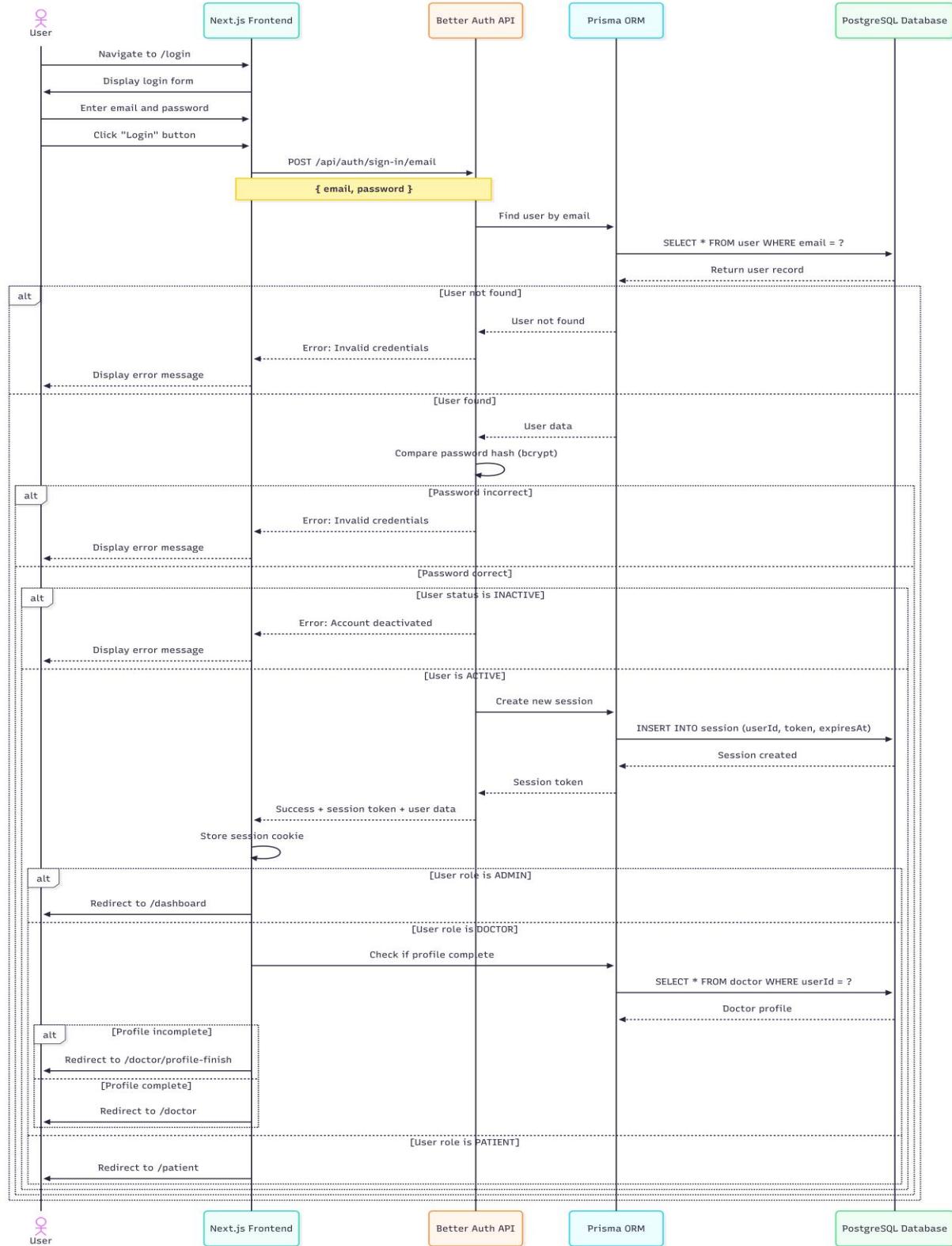
User Action	System Response
5. Patient and doctor conduct consultation.	4.3 The system enables chat, screen sharing, and recording controls.
6. Either party clicks "End Call" button.	5.1 The system maintains stable video/audio connection. 5.2 The system tracks call duration.
7. Use case ends.	6.1 The system terminates video connection. 6.2 The system updates Appointment status to COMPLETED. 6.3 The system prompts patient to rate the doctor. 7.1 Success notification: "Consultation completed."

3.3 Sequence Diagram

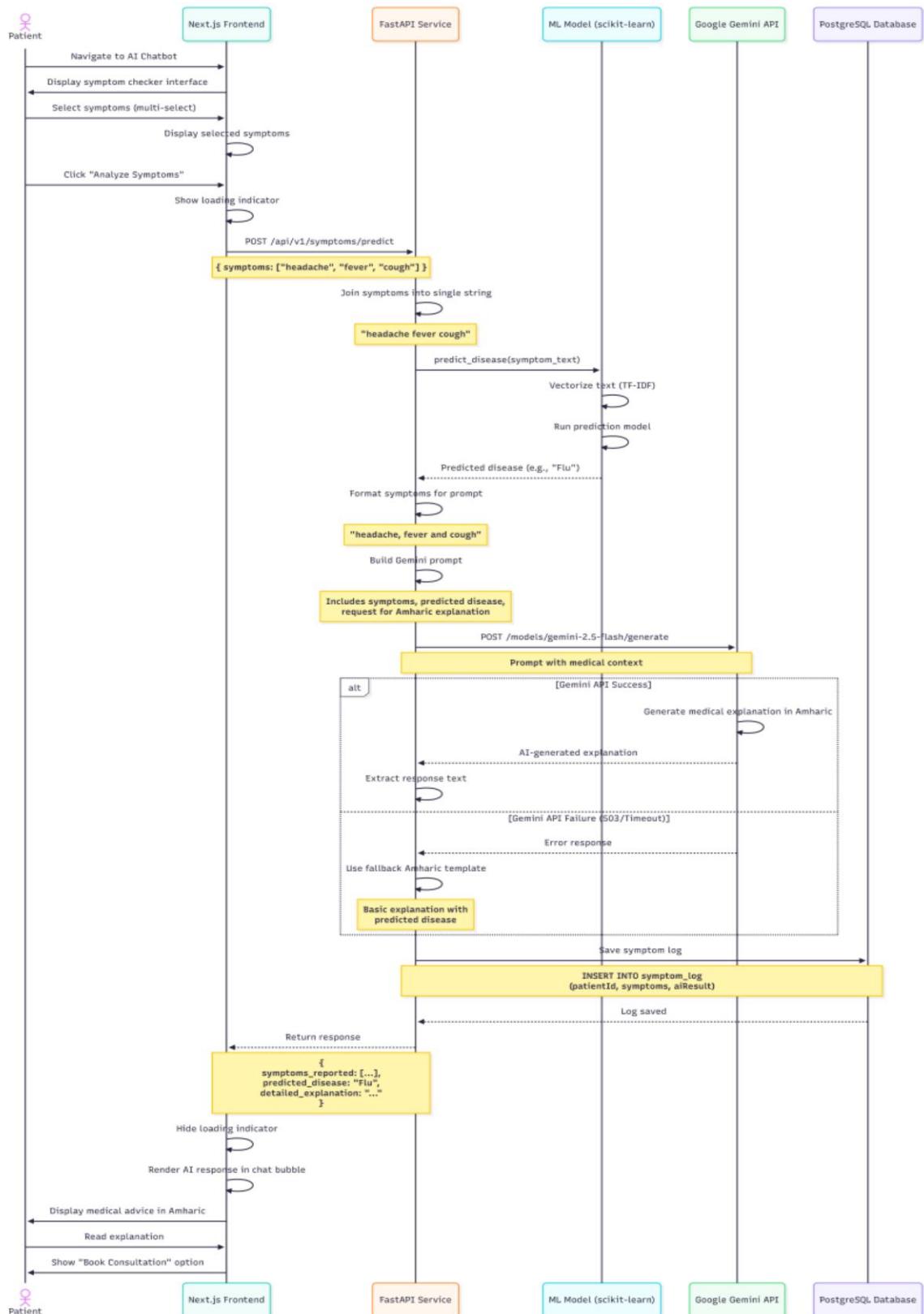
1. user registration sequence diagram



2. user login sequence diagram

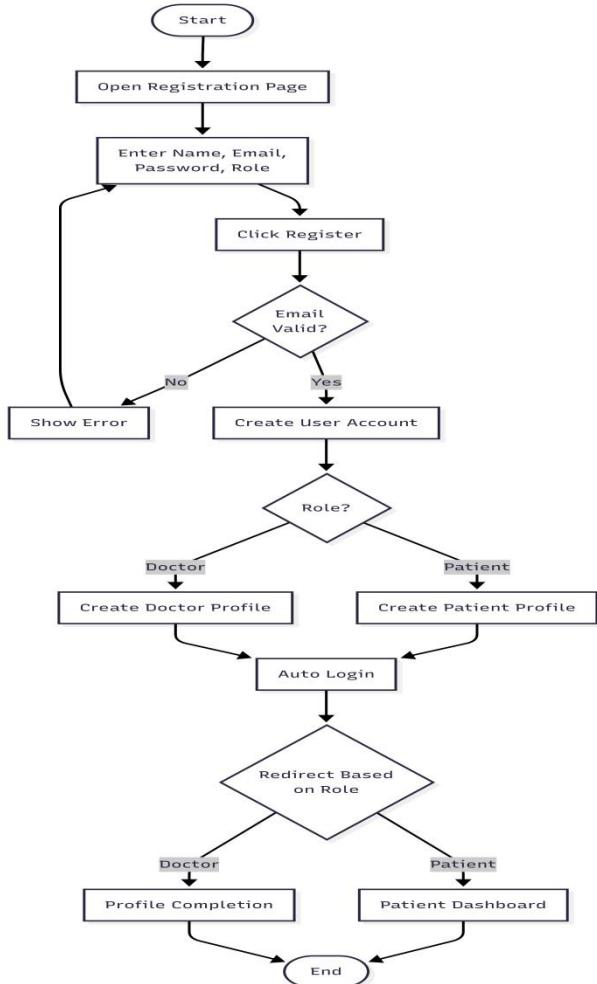


3. AI symptom analysis

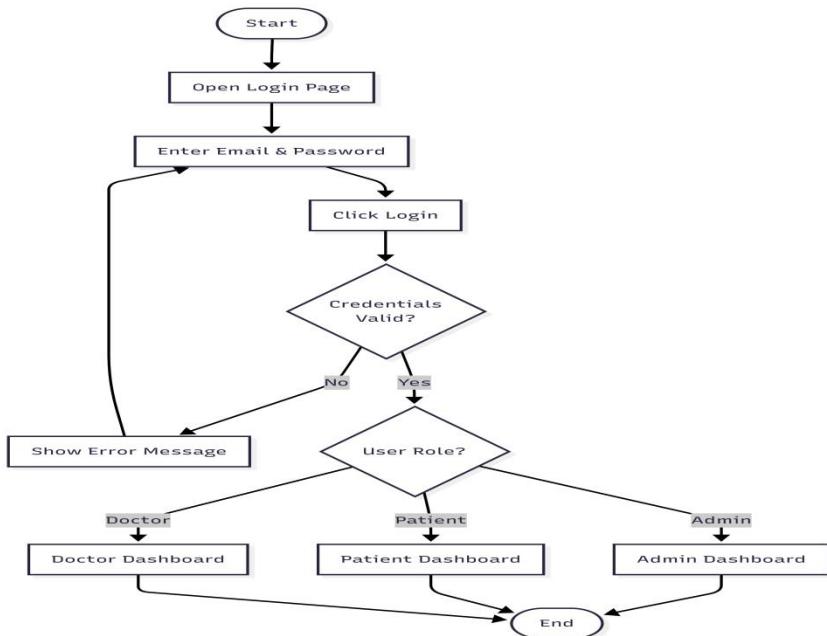


3.4 Activity Diagrams

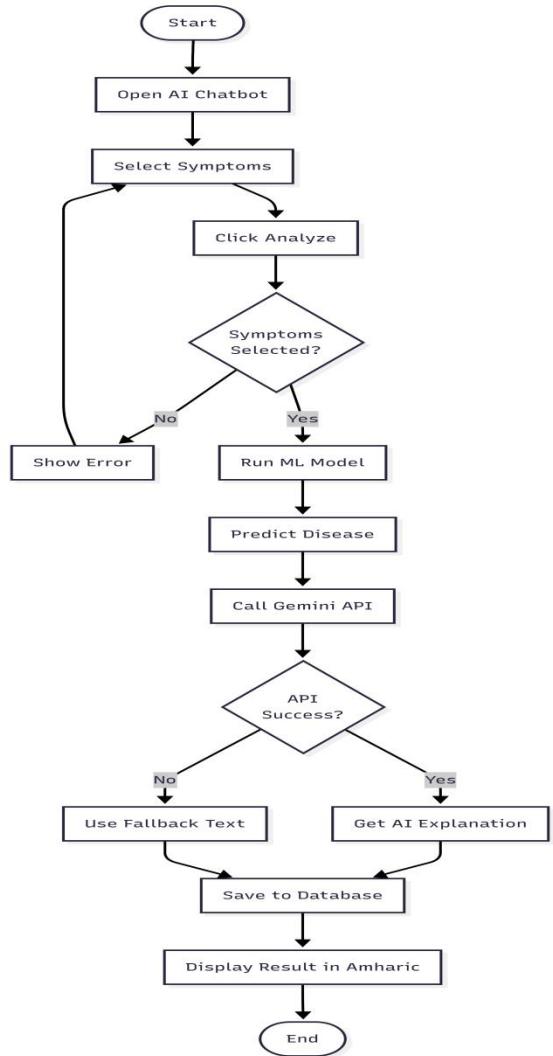
1. user registration activity diagram



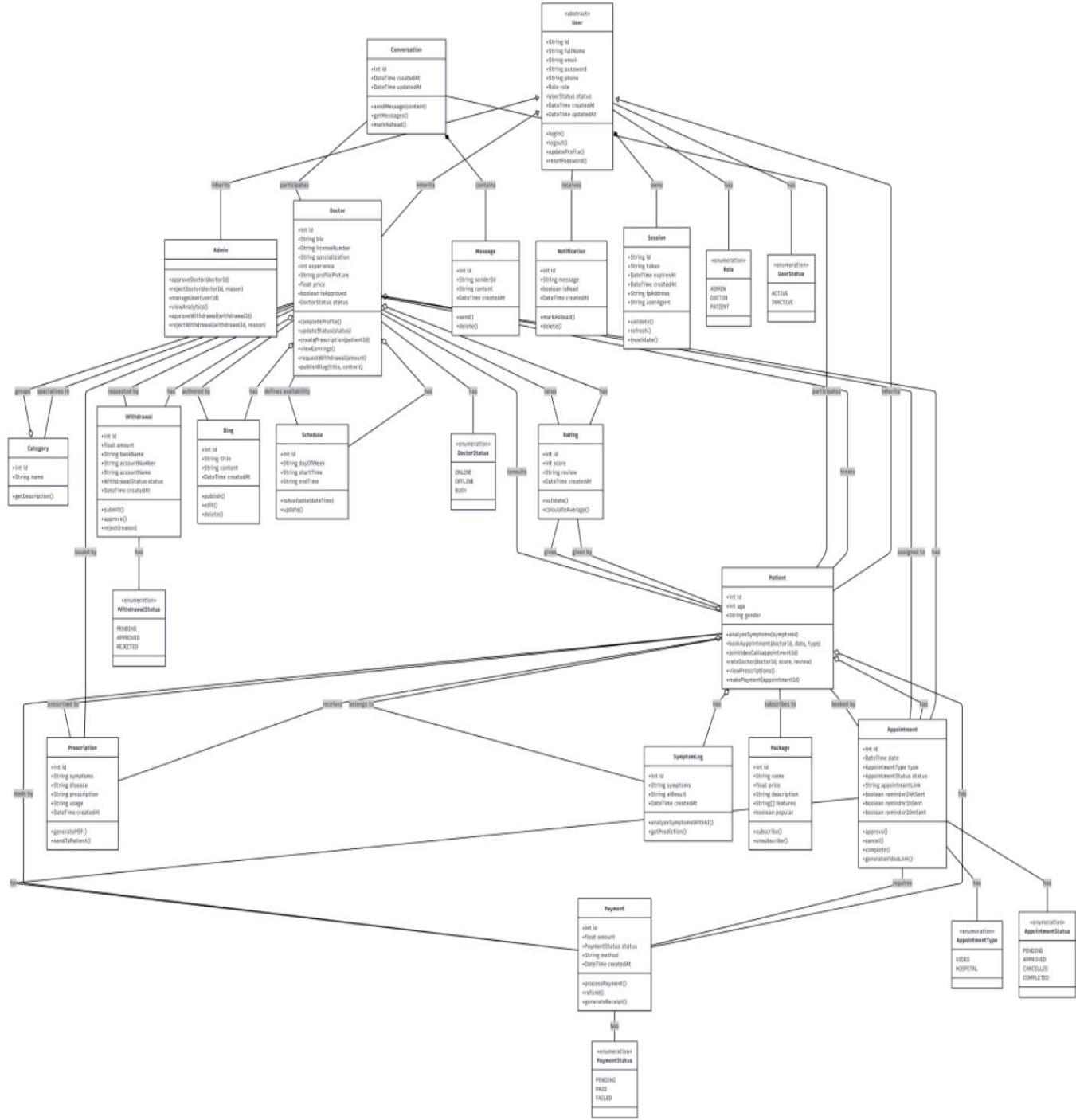
2 . user login activity diagram



3 . Ai syphom analysis diagram



3.5 Conceptual Class Diagram



Chapter Four: System Design

4.1 Overview