BREATHEWELL-LUNG DISEASE DETECTION AND CARE ASSISTANCE

Zulqarnain Hassan (SP21379)

Muhammad Younus(SP21452)

Mumtaz Ali(SP21386)

Supervised By

Dr. Maryam Imtiaz Malik

Submitted for the partial Fulfillment of BS Software Engineering degree to the Faculty of Engineering and Computing

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

December, 2025

ABSTRACT

Respiratory are widespread and often go undiagnosed due to the lack of accessible early detection tools. Early symptoms of these conditions are often subtle and easily overlooked, leading to delayed diagnosis and treatment. Limited access to specialized healthcare facilities, lack of affordable screening tools, and low public awareness contribute to the underdiagnosis and progression of respiratory illnesses. Many individuals find it difficult to monitor their lung health, identify worsening symptoms, and receive care recommendations without visiting a healthcare provider.

To address this challenge, this project introduces BreatheWell an application designed to detect, monitor, and manage respiratory conditions, specifically asthma, COPD, bronchitis, and pneumonia. With this application users can report symptoms to upload and record breathing and cough sounds, which are analyzed using machine learning to identify patterns such as wheezing, dry or wet coughs, and irregular breathing. By combining audio analysis with symptom input, BreatheWell delivers accurate assessments and helps users understand their respiratory status. Additionally, it provides features for tracking symptoms over time, generating visual timelines and detailed health reports. It also offers personalized care plans, home remedy suggestions, and alerts for worsening conditions, while integrating real-time air quality data to warn users of environmental risks. Moreover, an AI chatbot provides real-time support, and users can maintain a comprehensive medical history with downloadable reports for ongoing respiratory health management.

BreatheWell employs a modern full-stack architecture to deliver its AI-powered respiratory health solutions. The frontend is built with Reac.js to create a dynamic and responsive user interface. Backend services are powered by Python or Node.js, chosen for its extensive libraries in data science and machine learning, with frameworks such as Framework of python (Django or Flask) managing API development and server-side logic. utilizing PostgreSQL and MongoDB for efficient data management and firebase for authentication and real-time user-system communication.

CERTIFICATE

Date:_____________

Final Approval

It is certified that the project report titled **BreatheWell-lung disease detection and care assistance** submitted by Zulqarnain Hassan, Muhammad Younus and Mumtaz Ali for the partial fulfillment of the requirement of "Bachelor's Degree in Software Engineering" is approved.

COMMITTEE

Dr. Muhammad Noman Malik
DEAN FEC Signature: ____________________

Dr. Sumaira Nazir
HOD Software Engineering Signature: ____________________

Dr. Naveed Ahmed
Head Project Committee Signature: ____________________

Dr. Maryam Imtiaz Malik
Supervisor Signature: ____________________

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade and it may result in withdrawal of our Bachelor’s degree.

Group Members	**Signature**
1. Zulqarnain Hassan	____________________
2. Muhammad Younus	____________________
3. Mumtaz Ali	____________________

TABLE OF CONTENTS

CHAPTER **PAGE**

CHEPTER: 1 INTRODUCTION-- 12
1.1 Introduction -- 13
1.2 Motivation-- 13
1.2.1 Mild Early Symptoms and Late Diagnosis -- 13
1.2.2 Limited Access to Specialists --- 14
1.2.3 Expensive and Inconvenient Testing -- 14
1.2.4 Low Awareness and Poor Self-Tracking -- 14
1.2.5 Environmental Risks --- 14
1.2.6 Need for Ongoing and Personal Care --- 14
1.3 Problem Statement --- 14
1.4 Proposed System --- 15
1.4.1 Symptom Submission --- 15
1.4.2 AI-Powered Cough & Breathing Analysis -- 15
1.4.3 Smart Diagnosis System --- 15
1.4.4 Symptom Tracker & Dashboard -- 15
1.4.5 Personalized Care Plan --- 15
1.4.6 Environmental Alert System --- 15
1.4.7 AI Chatbot -- 15
1.4.8 Exercise Recommendations -- 16
1.5 Goals And Objectives -- 16
1.5.1 Goal --- 16
1.5.2 Objectives-- 16
1.6 Developed System Features -- 16
1.6.1 Symptom Submission --- 16
1.6.2 AI-Powered Cough and Breathing Analysis -- 16
1.6.3 Smart Diagnosis System --- 17
1.6.4 Symptom Tracker and Dashboard --- 17
1.6.5 Personalized Care Plan --- 17
1.6.6 Environmental Alert System --- 17

1.6.8 Patient History Tracking ---------- 17
1.6.9 Exercise Recommendations ---------- 17
1.6.10 Report Generation ---------- 18
1.6.11 User Authentication (Sign Up and Login) ---------- 18
1.7 Scope of the Study ---------- 18
1.8 Tools & Technology ---------- 18
1.8.1 Software ---------- 18
1.8.2 Hardware ---------- 18
1.9 Expertise of the Team Members ---------- 19
1.10 Process Model ---------- 20
1.11 Nature of the Project ---------- 21
1.12 Process Flow ---------- 21
1.13 Milestone ---------- 26
1.14 Overview ---------- 26
CHEPTER: 2 BACKGROUND AND EXISTING WORK ---------- 27
2.1 Introduction ---------- 28
2.2 Key Components of the BreatheWell System ---------- 28
2.2.1 Frontend – React ---------- 28
2.2.2 Backend ---------- 28
2.2.3 Artificial Intelligence – Machine Learning and Deep Learning ---------- 28
2.2.4 Database ---------- 29
2.3 Existing Systems ---------- 29
2.4 Limitations of Existing Systems ---------- 29
2.4.1 Ada Health ---------- 29
2.4.2 AsthmaMD ---------- 29
2.4.3 ResApp Health ---------- 30
2.4.4 Hyfe App ---------- 30
2.5 Comparison Table of Existing Systems vs. BreatheWell ---------- 30
2.6 Summary ---------- 31
CHEPTER: 3 REQUIREMENTS SPECIFICATION ---------- 32
3.1 Introduction ---------- 33

3.2 Interface Requirements -- 33
3.2.1 Hardware Requirements -- 33
3.2.2 Software Requirements -- 33
3.3 Functional Requirements -- 34
3.3.1 Signup -- 34
3.3.2 Login -- 34
3.3.3 Symptom Submission -- 34
3.3.4 Audio Recording/Upload -- 34
3.3.5 AI Audio Analysis -- 35
3.3.6 Smart Diagnosis -- 35
3.3.7 Personalized Care Plan Generation -- 35
3.3.8 Dashboard -- 36
3.3.9 Environmental Alerts -- 36
3.3.10 AI Chatbot Support -- 36
3.3.11 Patient History Tracking -- 36
3.3.12 Report Generation -- 37
3.3.13 Exercise Recommendation -- 37
3.3.14 Schedule Doctor appointment -- 37
3.3.15 See Appointment Request -- 37
3.4 Non-Functional Requirements -- 38
3.5 Use Case Diagram -- 38
3.6 Use Cases Description -- 40
3.6.1 Perform Smart Diagnosis -- 40
3.6.2 View History -- 41
3.6.3 Check Environmental Alert -- 41
3.6.4 Account Register -- 42
3.6.5 Consult AI Assistant -- 43
3.6.6 Generate Report -- 44
3.6.7 Submit Symptoms -- 45
3.6.8 Record Cough -- 46
3.6.9 View Care Plan -- 46

3.6.10 Schedule Doctor Appointment-- 47
3.6.11 Exercise Recommendation --- 48
3.6.12 See Appointment Request-- 49
3.7 Summary --- 49
CHEPTER: 4 SYSTEM MODELING-- 50
4.1 Introduction --- 51
4.2 System Design --- 51
4.2.1 Architectural Overview (Three-Tier Model) --- 51
4.2.2 Specialized Services and Interoperability -- 51
4.3 Interface Design --- 52
4.4 The 4+1 View Model of Architecture --- 52
4.4.1 Logical View --- 53
4.4.2 Process View -- 55
4.4.3 Development View -- 66
4.4.4 Physical View -- 67
4.5 App Design-- 69
4.6 Summary --- 69
REFERENCES -- 70

List of Figures

Figures **Page**

Figure 1.1 Waterfall Model Flow Diagram 21
Figure 1.2 Overall system's user flow 22
Figure 1.3 Flowchart for symptom submission 23
Figure 1.4 Flowchart for cough and breath record 24
Figure 1.5 Flowchart for smart diagnosis 25
Figure 1.6 Gantt chat 26
Figure 3.1 use case diagram 39
Figure 4.1 4+1 view model 53
Figure 4.2 class diagram 54
Figure 4.3 ERD for BreatheWell 55
Figure 4.4 Activity diagram 56
Figure 4.5 Sequence Diagram for Smart Diagnosis Check-up 57
Figure 4.6 sequence diagram for Track Health History 58
Figure 4.7 sequence diagram for Receive Environmental Alert 59
Figure 4.8 sequence diagram for Interact with AI Chatbot 60
Figure 4.9 sequence diagram for Health Report 61
Figure 4.10 sequence diagram for account registration 62
Figure 4.11 sequence diagram for care plan 63
Figure 4.12 sequence diagram for exercise recommendation 64
Figure 4.13 sequence diagram for appointment schedule 65
Figure 4.14 sequence diagram for see appointment request for doctor 66
Figure 4.15 Component Diagram for BreatheWell System 67
Figure 4.16 Deployment Diagram 68

List of Tables

Tables **Page**

Table 1.1 Tool & Technology 19
Table 1.2 Expertise of Members 19
Table 2.1 Existing Systems vs. BreatheWell 30
Table 3.1 Signup 34
Table 3.2 Login 34
Table 3.3 Symptom Submission 34
Table 3.4 : Audio Recording/Upload 35
Table 3.5 AI Audio Analysis 35
Table 3.6 Smart Diagnosis 35
Table 3.7 Personalized Care Plan Generation 35
Table 3.8 Dashboard 36
Table 3.9 : Environmental Alerts 36
Table 3.10 AI Chatbot Support 36
Table 3.11 Patient History Tracking 36
Table 3.12 Report Generation 37
Table 3.13: Exercise Recommendation 37
Table 3.14 Schedule Doctor appointment 37
Table 3.15 see appointment request 37
Table 3.16: Non-functional requirements 38
Table 3.17: Perform Smart Diagnosis 40
Table 3.18: View History 41
Table 3.19: Check Environmental Alert 42
Table 3.20: Account Register 42
Table 3.21 Consult AI Assistant 43
Table 3.22: Generate Report 44
Table 3.23: Submit Symptoms 45
Table 3.24: Record Cough 46
Table 3.25: View Care Plan 46

Table 3.26: Schedule Doctor Appointment 47
Table 3.27: Exercise Recommendation 48
Table 3.28: See Appointment Request.. 49

CHEPTER 1

INTRODUCTION

1.1 Introduction

BreatheWell is an innovative, AI-powered respiratory health monitoring system designed to assess and predict respiratory conditions, such as Asthma, Chronic Obstructive Pulmonary Disease (COPD), and Pneumonia, by analyzing cough and breathing sounds. Respiratory diseases pose significant health risks globally, often requiring timely diagnosis and continuous monitoring. Approximately half a billion people live with asthma and COPD, and together these two conditions cause almost 4 million deaths every year. More than 1 million of these deaths occur "prematurely" in people aged under 70 years [1]. Despite COPD's prevalence, access to affordable care remains limited, especially in low- and middle-income countries, where 85% of COPD deaths occur [2]. Traditional diagnostic methods, such as spirometry and peak flow meters, are effective but typically rely on clinical settings, making them less accessible for many individuals.

To bridge this gap, BreatheWell uses machine learning and sensor-based data collection to provide a user-friendly solution for proactive respiratory care. By leveraging advanced signal processing and deep learning algorithms, the system analyzes and classifies various respiratory sounds, including dry/wet coughs, wheezing, and abnormal breathing patterns. This allows for the early detection of respiratory conditions, enabling real-time monitoring and prediction of lung health status.

The goal of BreatheWell is to enhance remote health monitoring, reduce diagnostic delays, and empower individuals to take charge of their respiratory well-being. The system not only serves as a tool for individuals to monitor their condition from the comfort of their homes but also has the potential to assist healthcare providers in diagnosing respiratory diseases more efficiently and at an earlier stage. By providing users with valuable insights and real-time analysis, BreatheWell aims to significantly improve respiratory care and prevent complications associated with undiagnosed or poorly managed respiratory conditions.

BreatheWell will not commonly found in other health monitoring apps or websites. While tools like Hyfe and ResApp Health provide breath analysis and symptom tracking, BreatheWell goes a step further with its AI-driven predictions based on audio analysis. Additionally, it incorporates an environment alert system and doctor connect/chatbot triage, providing a comprehensive solution for respiratory health. With a focus on conditions like bronchitis, BreatheWell offers a versatile tool for managing and preventing various respiratory diseases, setting it apart from existing solutions.

1.2 Motivation

The idea for BreatheWell came from noticing several important problems in today's respiratory healthcare:

1.2.1 Mild Early Symptoms and Late Diagnosis

Many lung diseases start with very small symptoms that people often ignore. Because of this, they delay visiting a doctor, which can lead to serious lung damage that could have been prevented.

1.2.2 Limited Access to Specialists

In many rural or less developed areas, it's hard to find lung specialists or good respiratory care. This lack of access causes even more delays in getting the right treatment.

1.2.3 Expensive and Inconvenient Testing

Easy and affordable tools for early lung health screening are not available to everyone. Most traditional tests require going to hospitals or clinics, which can take time and cost a lot.

1.2.4 Low Awareness and Poor Self-Tracking

Many people don't realize how important it is to keep an eye on their breathing health. They might not notice patterns of worsening symptoms or know when to seek medical help.

1.2.5 Environmental Risks

Things like air pollution and pollen can badly affect people's lungs, especially if they already have breathing problems. Getting real-time updates about these risks can help people take action early.

1.2.6 Need for Ongoing and Personal Care

Chronic lung diseases need regular monitoring and care plans that fit each person's needs. Having a system that offers continuous support and customized advice can make a big difference in people's health and daily life.

1.3 Problem Statement

Worldwide, the prevalence of respiratory conditions like asthma, COPD, pneumonia, and others is rising, which poses significant health risks. To reduce complications and improve patient outcomes, early diagnosis and ongoing monitoring are essential. Real-time data for continuous monitoring is frequently unavailable to patients outside of healthcare settings, and traditional diagnostic tools frequently necessitate clinic visits, limiting accessibility. This leads to delayed diagnoses, poor management of symptoms, and ultimately worsened health conditions.

Without access to continuous monitoring, many patients with chronic respiratory diseases struggle to receive timely care or modify their treatment plans. Moreover, there is a lack of personalized, at-home solutions that integrate real-time data and medical guidance.

By providing an AI-driven platform that analyzes cough sounds and breathing patterns to detect early signs of respiratory conditions like asthma, COPD, and pneumonia, BreatheWell hopes to fill in these gaps. The system empowers users to actively monitor their respiratory health from home, get personalized insights, and make informed decisions about their treatment. By integrating environmental alerts, symptom tracking, and doctor connect/chatbot triage, BreatheWell provides a comprehensive tool for proactive respiratory care, improving quality of life, and reducing the burden on healthcare systems.

1.4 Proposed System

BreatheWell is suggested as a user-friendly web application that uses artificial intelligence to help in the detection, tracking, and management of health conditions related to the lungs. Through a combination of self-reported symptoms and AI-driven analysis of their cough and breathing sounds, the core of the system enables users to actively participate in monitoring their respiratory health.

The following essential features will be incorporated into the proposed system:

1.4.1 Symptom Submission

A user-friendly interface that allows them to record a variety of respiratory symptoms.

1.4.2 AI-Powered Cough & Breathing Analysis

Utilizing machine learning models to analyze user-recorded or uploaded audio samples to identify patterns indicative of different respiratory conditions.

1.4.3 Smart Diagnosis System

An intelligent module that combines submitted symptom data with the results of the audio analysis to provide a more accurate prediction of potential lung conditions.

1.4.4 Symptom Tracker & Dashboard

A visual platform for users to track their symptoms over time, view trends, and gain insights into their respiratory health.

1.4.5 Personalized Care Plan

Generation of tailored recommendations, including home remedies, lifestyle adjustments, and guidance on when to seek professional medical advice, based on the system's analysis.

1.4.6 Environmental Alert System

Preventive advice and notifications in real time about air quality in the user's area.

1.4.7 AI Chatbot

An interactive virtual assistant that provides immediate support, and answer common questions.

1.4.8 Exercise Recommendations

Based on the user's condition, suggestions for appropriate light breathing exercises and physical activities.

BreatheWell aims to bridge the gap between individuals and proactive respiratory health management by combining these features into a unified and accessible web application.

1.5 Goals And Objectives

1.5.1 Goal

The goal of BreatheWell is to develop an AI-powered, non-invasive respiratory health monitoring system that enables early detection and continuous tracking of respiratory conditions like asthma, COPD, pneumonia, and bronchitis, providing users with real-time insights and personalized care.

1.5.2 Objectives

To provide detection respiratory conditions like asthma, COPD, and pneumonia through AI analysis of cough sounds and breathing patterns.

To offer a non-invasive solution for ongoing monitoring of respiratory health at home.

To provide customized symptom tracking and treatment recommendations based on individual health data.

1.6 Developed System Features

The BreatheWell system has a lot of features that help with the problems that were found and accomplish the goals of the project. Each feature is carefully designed to enhance early detection, continuous monitoring, and personalized respiratory care. The main features of the system are outlined below:

1.6.1 Symptom Submission

Users can easily submit information about their current respiratory symptoms through a simple and interactive form. This makes it easier to collect structured data about symptoms, which is important for early analysis and diagnosis.

1.6.2 AI-Powered Cough and Breathing Analysis

BreatheWell analyzes audio recordings of coughs and breathing sounds using machine learning models. The system can assist in the early detection of potential respiratory issues by detecting patterns and anomalies in the audio signals.

1.6.3 Smart Diagnosis System

The system combines user-submitted symptoms and the results from audio analysis to provide AI-driven predictions about possible lung conditions. This smart diagnostic feature aims to guide users toward early medical intervention.

1.6.4 Symptom Tracker and Dashboard

A user-friendly dashboard visually displays the user's symptom history, trends, and health progress over time. Users can keep track of their respiratory health with this feature and see if there are any patterns of improvement or deterioration.

1.6.5 Personalized Care Plan

Based on the user's health data and diagnosis results, the system offers customized recommendations. These include home remedies, lifestyle changes, medication reminders, and guidance on when to seek professional medical help.

1.6.6 Environmental Alert System

BreatheWell monitors real-time air quality. Users receive timely alerts and health advice to help them avoid exposure to harmful environmental conditions that could worsen their respiratory issues.

1.6.7 AI Chatbot Support

An integrated AI chatbot provides 24/7 assistance for common queries related to symptoms and care plans. This ensures continuous support and enhances user engagement.

1.6.8 Patient History Tracking

BreatheWell maintains a detailed record of each user's respiratory health journey, including submitted symptoms, diagnostic results, and care plans. This history aids in better understanding long-term health patterns.

1.6.9 Exercise Recommendations

The system suggests appropriate breathing exercises and physical activities tailored to the user's specific respiratory condition. This encourages active participation in improving lung function and overall health.

1.6.10 Report Generation

Users can generate detailed health reports summarizing their symptoms, diagnostic results, and care plans. These reports can be downloaded for personal use or during consultations.

1.6.11 User Authentication (Sign Up and Login)

To ensure secure access to personal health information, BreatheWell includes a complete authentication system. New users can create an account through a Sign-Up process, while returning users can access their personalized dashboard through a secure Login. User data privacy is safeguarded and a personalized healthcare experience is made possible by authentication.

1.7 Scope of the Study

The scope of this project focuses on the development and initial testing of the BreatheWell web application's core features for the detection and management of common lung conditions, specifically asthma, bronchitis, COPD (Chronic Obstructive Pulmonary Disease), and pneumonia. This includes the user interface and backend infrastructure, the integration of AI/ML models for audio analysis to identify patterns indicative of these conditions and the smart diagnosis system to predict the likelihood of their occurrence based on submitted symptoms and audio analysis. The project also covers the symptom tracking dashboard, personalized care plan generation tailored to these conditions, a real-time environmental alert system (using public APIs), a basic AI chatbot to provide information related to these diseases, a doctor appointment request system, and functionalities for patient history tracking and report generation pertaining to these specific lung conditions.

1.8 Tools & Technology

Some of the tools and technologies that can be used in our system are listed below:

1.8.1 Software

1. React
2. Node.js/Django/flask
3. Firebase
4. MongoDB
5. Figma

1.8.2 Hardware

1. Laptop/PC core i3-4th Generation or above for better performance
2. 4GB RAM or above

Table 1.1 show the detailed tool and technologies use to build BretheWell

Table 1.1 Tool & Technology

Tool & Technology	Explanation
React(frontend)	React is a powerful and widely-used JavaScript library for building modern, responsive, and dynamic user interfaces.
Python(Flask)	Node.js is a fast, lightweight JavaScript runtime used for building scalable server-side applications, APIs, and real-time services.
Firebase	We will use Firebase as the back-end language to implement OTP and messaging.
Machine learning	Machine Learning is a technology where computers use algorithms to recognize patterns, make decisions, and improve from experience without manual programming.
Deep Learning	Deep Learning is a subset of machine learning where algorithms are structured in layers to create an artificial neural network that can learn and make intelligent decisions.
MongoDB	MongoDB is a popular NoSQL database that stores data in flexible, JSON-like documents instead of traditional tables.

1.9 Expertise of the Team Members

Each team member is pre-equipped with the basic knowledge needed to complete this project.

Table 1.2 show the expertise of our team members.

Table 1.2 Expertise of Members

Group Members	Expertise
Zulqarnain Hussain	Programming (C++, Python, JavaScript),MongoDB, UML Modelling, Web Development, Machine & Deep Learning, flask, react, node.js
Muhammad Younus	Programming (C++, Java, Python, C#),Web Development, Documentation & MongoDB
Mumtaz Ali	Programming (C++, Java, Python, C#), Web Development, Designing, Testing

1.10 Process Model

The Waterfall process model will be used to develop BreatheWell. This is a linear and sequential model where each phase of the software development lifecycle is completed in its entirety before the next phase begins.

The phases in the Waterfall model typically include:

1.10.1 Requirements Gathering and Analysis

A thorough understanding of all the functional and non-functional requirements of the BreatheWell system is documented. This phase aims to record a comprehensive and unambiguous description of the system's requirements.

1.10.2 System Design

Based on the detailed requirements, a comprehensive system design is created. This includes the overall architecture of the application, database design, user interface design, and specifications for hardware and software components.

1.10.3 Implementation

The actual coding and development of the BreatheWell web application take place in this phase, following the design specifications. Each module and component of the system is built and integrated.

1.10.4 Testing

Once the implementation is complete, the entire system undergoes rigorous testing to identify and fix any defects or bugs. Different levels of testing, such as unit testing, integration testing, and system testing, are performed.

1.10.5 Deployment

After successful testing, the BreatheWell application is deployed to the target environment, making it accessible to users.

1.10.6 Maintenance

The system enters the maintenance phase following deployment, which entails addressing any issues that arise, providing updates, and possibly adding new features in response to user feedback or changing requirements.

1.10.7 Predictability and Milestones

Figure 1.1 illustrates the Waterfall model, a classic linear-sequential software development life cycle (SDLC) approach. This methodology requires the strict completion of one phase before proceeding to the next, following a downward flow through requirement analysis, system design, implementation, testing, and maintenance.

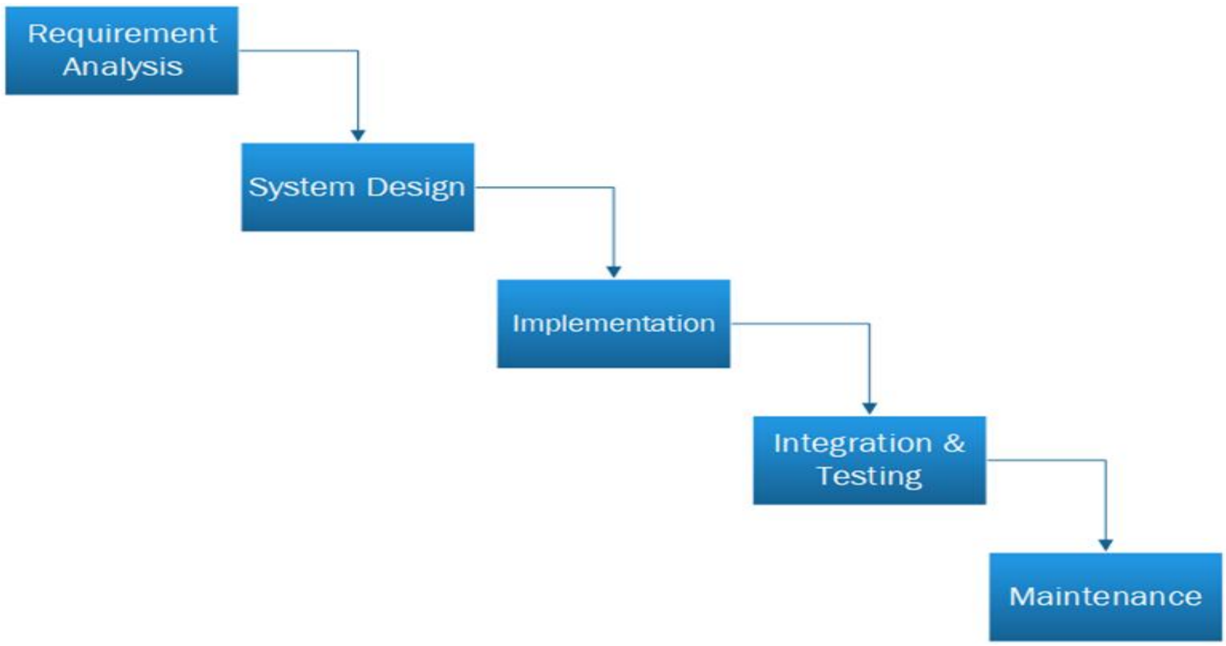

Figure 1.1 Waterfall Model Flow Diagram

1.11 Nature of the Project

This project is primarily a Research and Development (R&D) endeavor focused on exploring the application of artificial intelligence and web technologies to address challenges in respiratory health management. The initial phase involves extensive research into the viability and efficacy of AI-driven audio analysis and smart diagnosis in this field, despite the ultimate objective of creating a useful tool for individuals. Because it involves the design, development, and testing of a complex web application with a variety of interconnected features, the project also incorporates elements of software development. Furthermore, it touches upon Healthcare Technology, aiming to provide a digital solution that can potentially improve health outcomes and empower individuals in managing their well-being.

The interdisciplinary nature of the project, combining expertise in software development, artificial intelligence, and an understanding of healthcare needs, makes it a multifaceted undertaking.

1.12 Process Flow

We will describe the flow of our project through a flowchart

1.12.1 Overall System's user Flow

Figure 1.2 shows the overall system's user flow.

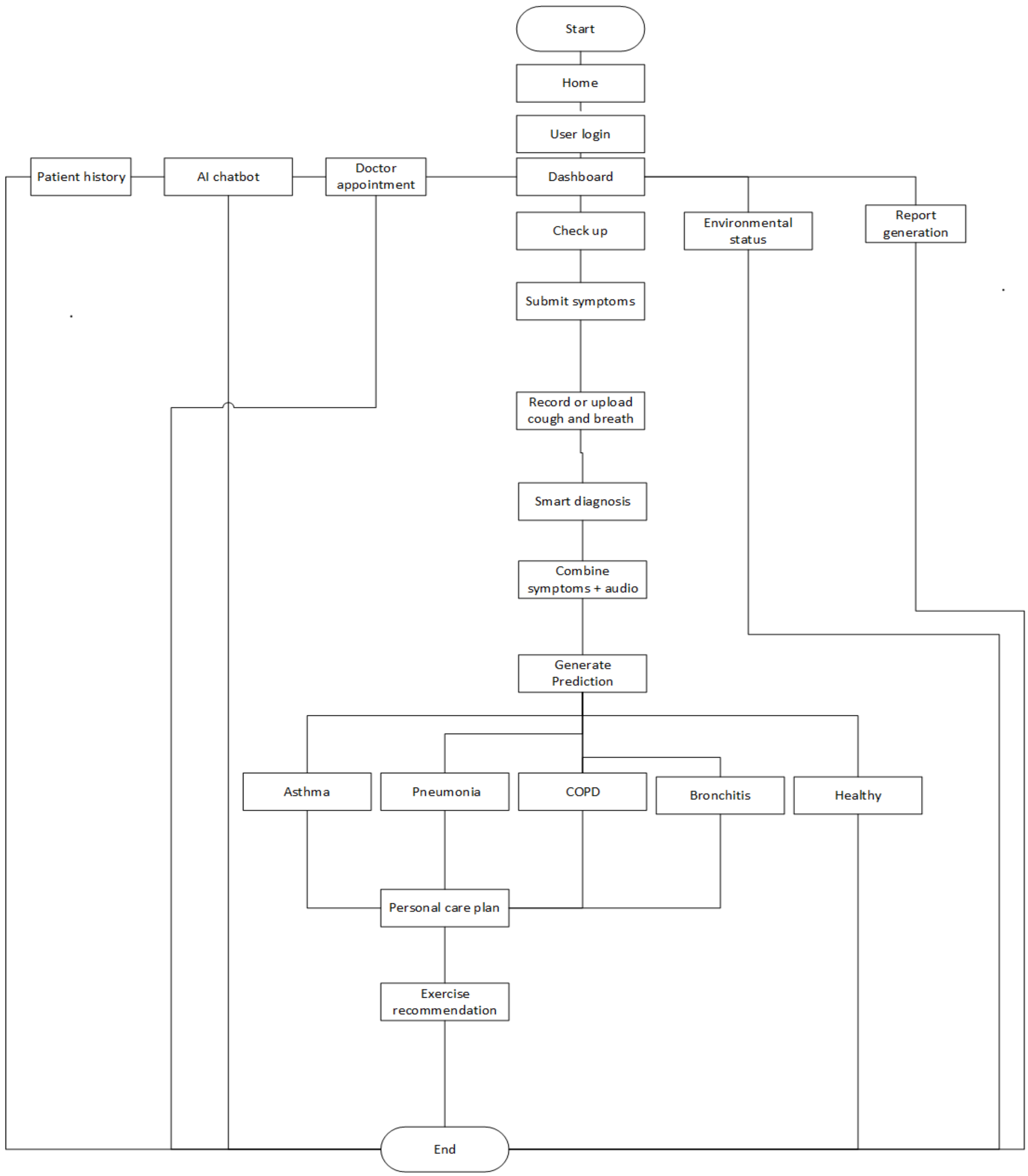

Figure 1.2 Overall system's user flow

1.12.2 Flowchart for symptom submission

Figure 1.3 show the flow of symptom submission.

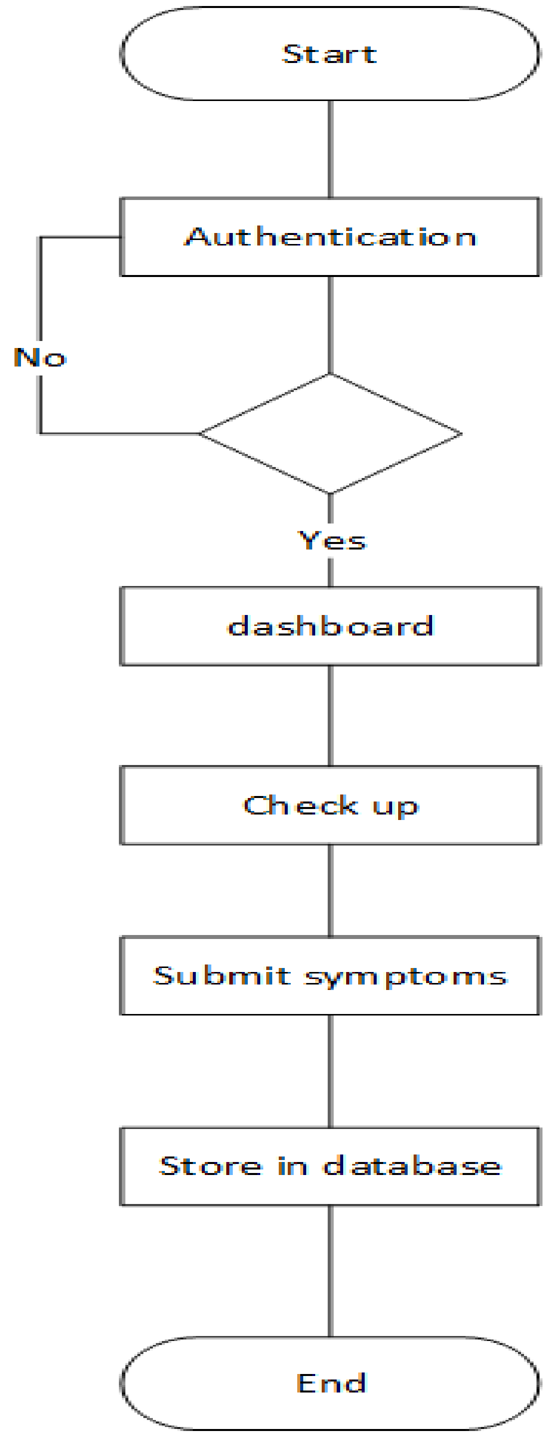

Figure 1.3 Flowchart for symptom submission

1.12.3 Flowchart for cough and breath record

Figure 1.4 show the flow of cough recording

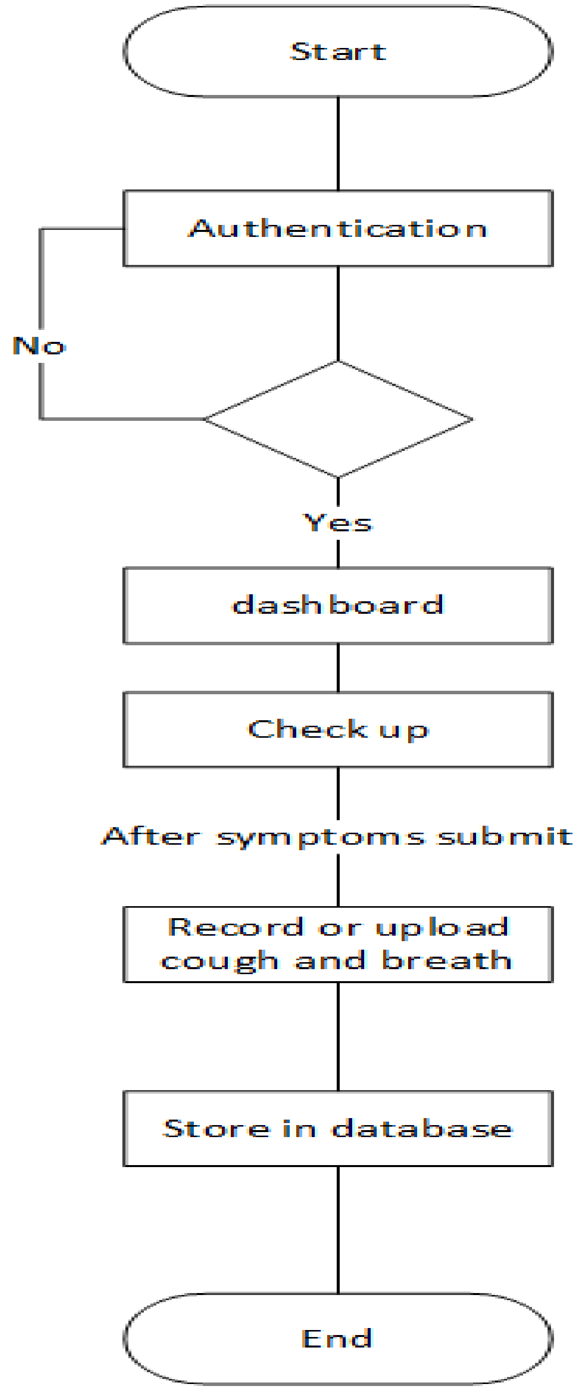

Figure 1.4 Flowchart for cough and breath record

1.12.4 Flowchart for smart diagnosis

Figure 1.5 shows how smart diagnosis combine symptoms and cough to predict diseases

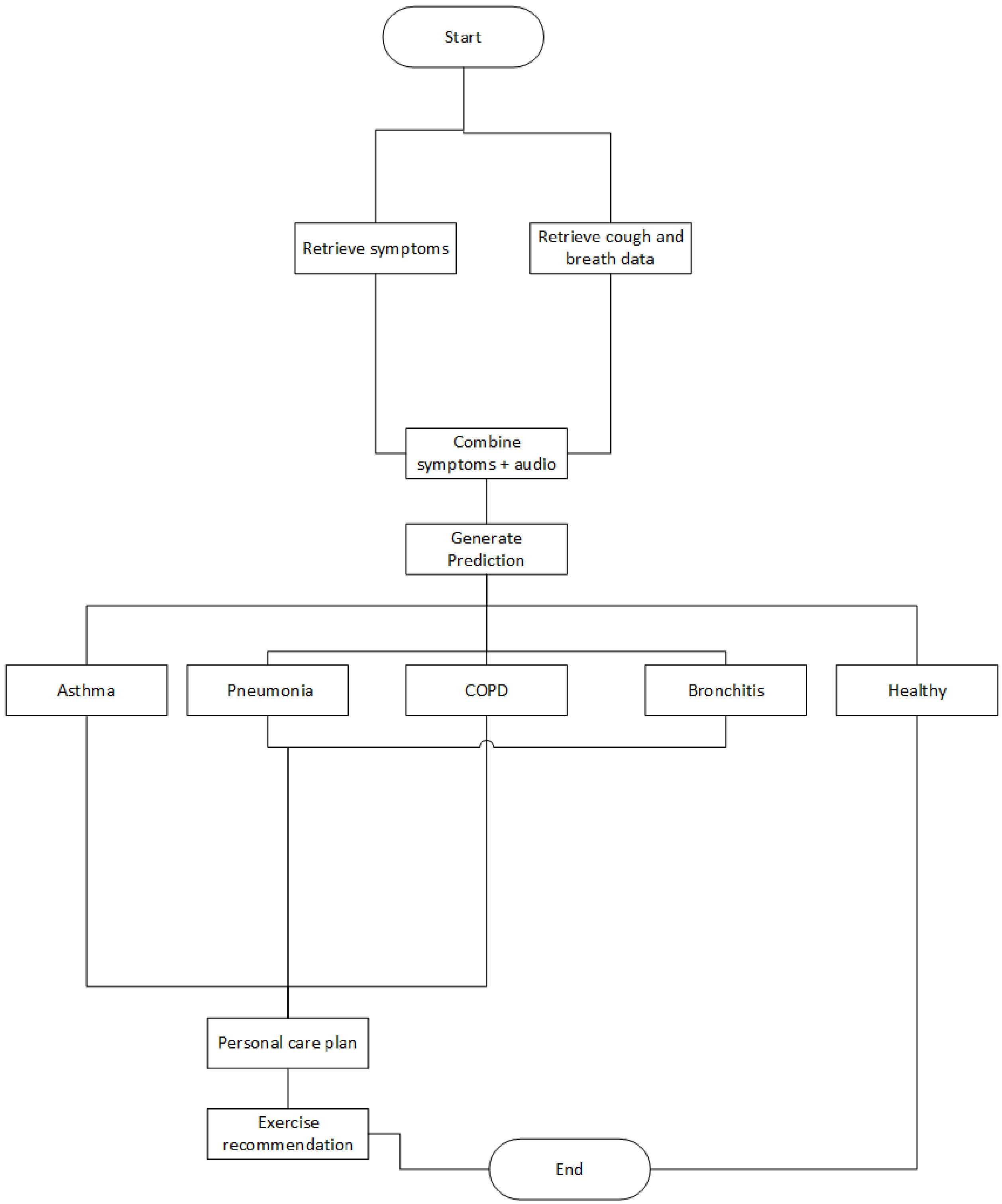

Figure 1.5 Flowchart for smart diagnosis

1.13 Milestone

Figure 1.6 shows the overall milestone of the project.

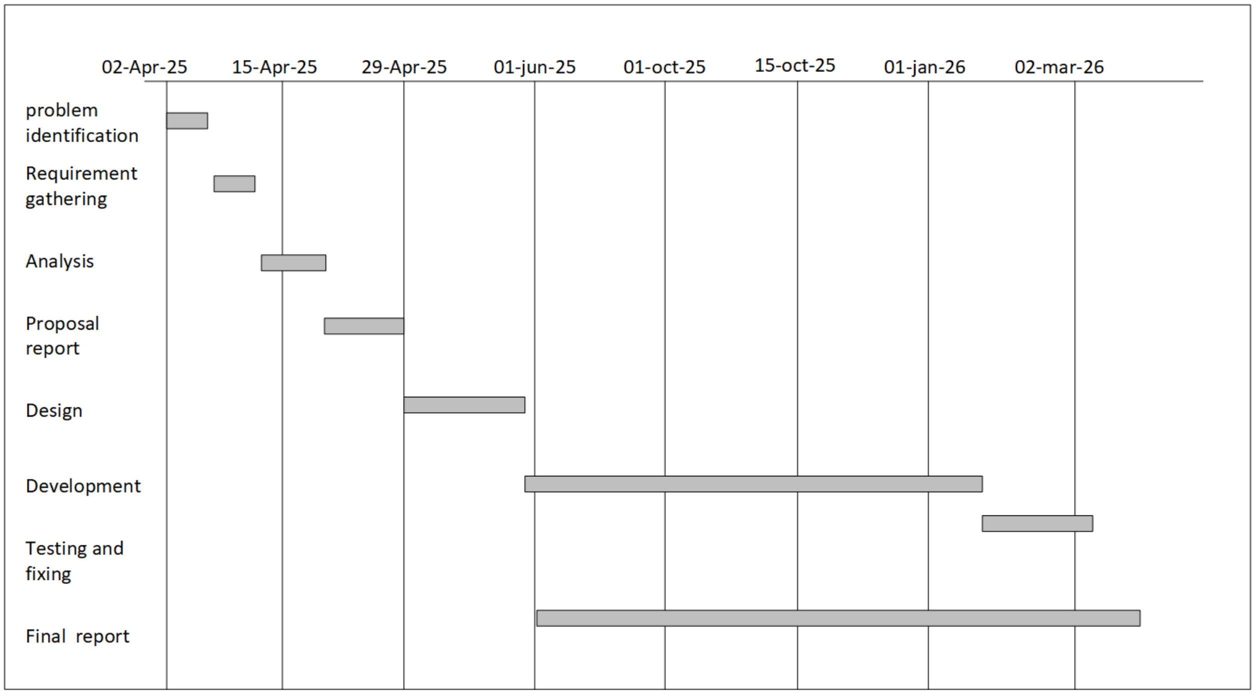

Figure 1.6 Gantt chat

1.14 Overview

Chapter 1, "Introduction," establishes the critical context for the BreatheWell project by highlighting the global challenge of respiratory diseases and the limitations in current detection and management. It outlines the motivations for developing an AI-powered web application, clearly defines the problems BreatheWell aims to solve, and introduces the proposed system with its core features. The chapter then sets the project's direction by detailing its goals and objectives, delineates the scope of the current study, and provides an overview of the technologies, team expertise, and the Incremental development process. Finally, it offers a glimpse into the project's flow, key milestones, and a roadmap for the subsequent chapters, providing a comprehensive foundation for understanding the BreatheWell initiative.

CHEPTER 2

BACKGROUND AND EXISTING WORK

2.1 Introduction

In this chapter, we will explore the background of the BreatheWell software project. We'll look at some existing systems for respiratory health monitoring and highlight their features and limitations. By comparing these systems with BreatheWell, we will show how it offers a more complete solution.

2.2 Key Components of the BreatheWell System

BreatheWell has several important components that make it work:

2.2.1 Frontend – React

React is a popular JavaScript library used to build interactive user interfaces. It's chosen for BreatheWell because:

- It allows us to create reusable UI components.
- It is fast and updates the page efficiently with its Virtual DOM.
- It has a large community and many useful libraries.
- It easily connects with the backend to show real-time health data.

2.2.2 Backend

The backend of BreatheWell is built using either Node.js or Django, each offering unique advantages:

Flask

- Provides security and efficiency for handling large health data.

- Comes with an admin panel to manage user data and health records easily.

2.2.3 Artificial Intelligence – Machine Learning and Deep Learning

Cough and Breathing Sound Analysis (Deep Learning)

- Deep learning models (like CNNs, RNNs, or specialized audio classification models) are used to analyze cough and breathing sounds recorded by the user.
- The system learns to detect patterns or abnormalities in audio that may be linked to diseases like asthma, bronchitis, pneumonia.
- Instead of manually analyzing sounds, the deep learning model automatically predicts if the sound is normal or abnormal.

Smart Diagnosis System (Machine Learning)

- Machine learning models (such as Decision Trees, Random Forest, SVM, or Ensemble Methods) are used to combine the symptom data (from user input) and audio analysis results.
- The ML model predicts the most likely respiratory condition based on the combined data.
- This helps in early and automated detection without waiting for a doctor's manual evaluation.

2.2.4 Database

BreatheWell uses MongoDB databases to store and manage health data. These databases offer:
Scalability and flexibility to handle large amounts of data.
Secure and efficient data storage with real-time updates.

2.3 Existing Systems

Several healthcare apps exist to monitor general health or lung conditions. Notable examples include:

- **Ada Health** [3]: A general health app with AI-based symptom checking.
- **AsthmaMD** [4]: An app designed specifically for managing asthma.
- **ResApp Health** [5]: Uses cough sound analysis to diagnose lung conditions.
- **Hyfe App** [6]: Focuses on cough analysis for lung health.

These apps are useful but have limitations that BreatheWell aims to address.

2.4 Limitations of Existing Systems

2.4.1 Ada Health

- **Symptoms**: Tracks general health but not specifically respiratory conditions.
- **Limitations**:
    - No cough or breath analysis.
    - No real-time lung health monitoring.
    - No personalized treatment plans or environmental alerts.

2.4.2 AsthmaMD

- **Symptoms**: Designed for asthma management, tracking asthma symptoms.
- **Limitations**
    - Only focuses on asthma, not other conditions like COPD or pneumonia.
    - No cough or breath sound analysis.

    - No environmental or personalized care alerts.

2.4.3 ResApp Health

- **Symptoms**: Diagnoses conditions like asthma using cough sound analysis.
- **Limitations**
    - No continuous symptom tracking.
    - Lacks personalized treatment or environmental alerts.

2.4.4 Hyfe App

- **Symptoms**: Focuses on analyzing cough sounds.
- **Limitations**
    - Only tracks cough, not other respiratory symptoms.

    - No environmental or personalized care alerts.

2.5 Comparison Table of Existing Systems vs. BreatheWell

Table 2.1 shows the comparison between existing system and the proposed system (BreatheWell).

Table 2.1 Existing Systems vs. BreatheWell

Feature	Ada Health	AsthmaMD	ResApp Health	Hyfe App	Proposed System
Cough & Breath Analysis	No	No	Yes	Yes	Yes
Symptom Tracking	Yes	Yes	No	Yes	Yes
Personalized Treatment Plans	Yes	Yes	No	No	Yes
Environmental Alert System	No	No	No	No	Yes
Chatbot assistant	No	No	No	No	Yes
Doctor appointment	No	No	No	No	Yes

History tracking	No	No	No	No	Yes
Exercise recommendation	No	No	No	No	Yes

2.6 Summary

This chapter has reviewed existing healthcare systems that monitor respiratory health. While platforms like Ada Health, AsthmaMD, ResApp Health, and Hyfe App provide useful features, none of them offer a comprehensive solution for lung health. BreatheWell overcomes these gaps by combining cough and breath analysis, symptom tracking, personalized care plans, environmental alerts, and chatbot triage.

CHEPTER 3
REQUIREMENTS SPECIFICATION

3.1 Introduction

This chapter formally outlines the detailed requirements for the BreatheWell AI-powered web application, an innovative respiratory health monitoring system. It defines the necessary resources and specifications for its development and operation, categorizing the system's needs into functional and non-functional requirements. These requirements ensure that the system meets its goals of providing non-invasive, AI-driven detection, tracking, and personalized care for respiratory conditions such as asthma, COPD, pneumonia, and bronchitis. Finally, the chapter visualizes the primary user-system interactions through a Use Case Diagram and provides detailed descriptions of the key use cases.

3.2 Interface Requirements

The successful development and deployment of BreatheWell require hardware, software and human expertise resources.

3.2.1 Hardware Requirements

The minimum hardware specifications for development and better performance are:

- **Processor:** Laptop/PC with a Core i3-4th Generation or above.
- **RAM:** 4GB or above.
- **User Devices:** Devices (smartphones, PCs) must have a reliable microphone for recording cough and breathing sounds.

3.2.2 Software Requirements

This covers the operating systems, development tools, frameworks, and databases required.

- **Frontend: React.js** is used to create a dynamic and responsive user interface.
- **Backend & Logic: Python** is chosen for its extensive libraries in data science and machine learning, with frameworks such as Django or Flask managing API development and server-side logic.

  Node.js is also considered for the backend.
- **Artificial Intelligence:** Machine Learning and Deep Learning algorithms are utilized for analyzing respiratory sounds and smart diagnosis.
- **Database: MongoDB** (a popular NoSQL database) are utilized for efficient data management.

- **Authentication/Real-time:** Firebase is used for authentication, real-time user-system communication, and implementation of OTP and messaging.

- **Design & Modelling:** Figma is used for designing and modeling.

3.3 Functional Requirements

Functional requirements define the specific actions or services the BreatheWell system must perform for the user.

3.3.1 Signup

Table 3.1 shows the user registration functional requirement

Table 3.1 Signup

FR-01	Signup
Description	The system shall enable the user to register their account.
Priority	High

3.3.2 Login

Table 3.2 shows the user Login functional requirement

Table 3.2 Login

FR-02	Login
Description	The system shall provide a login to the user.
Priority	High

3.3.3 Symptom Submission

Table 3.3 shows the Symptom Submission functional requirement.

Table 3.3 Symptom Submission

FR-03	Symptom Submission
Description	The system must provide a user-friendly interface for users to easily record and submit information about their current respiratory symptoms
Priority	High

3.3.4 Audio Recording/Upload

Table 3.4 shows the Audio Recording/Upload functional requirement.

Table 3.4 : Audio Recording/Upload

FR-04	Audio Recording/Upload
Description	The system must allow users to record or upload cough and breathing sounds through an intuitive web interface.
Priority	High

3.3.5 AI Audio Analysis

Table 3.5 shows the AI Audio Analysis functional requirement.

Table 3.5 AI Audio Analysis

FR-05	AI Audio Analysis
Description	The system must leverage deep learning to analyze uploaded/recorded audio samples to identify respiratory patterns, such as dry/wet coughs, wheezing, and irregular breathing.
Priority	High

3.3.6 Smart Diagnosis

Table 3.6 shows the Smart Diagnosis functional requirement.

Table 3.6 Smart Diagnosis

FR-06	Smart Diagnosis
Description	The system must combine submitted symptom data with the results of the audio analysis to provide an AI-driven prediction of potential lung conditions.
Priority	High

3.3.7 Personalized Care Plan Generation

Table 3.7 shows the Personalized Care Plan Generation functional requirement.

Table 3.7 Personalized Care Plan Generation

FR-07	Personalized Care Plan Generation
Description	The system must generate tailored recommendations, including home remedies, lifestyle adjustments, and professional guidance, based on the system's analysis and diagnosis.
Priority	High

3.3.8 Dashboard

Table 3.8 shows the Dashboard functional requirement.

Table 3.8 Dashboard

FR-08	Dashboard
Description	The system must provide a visual platform (dynamic dashboard) from where users can to other functionalities.
Priority	High

3.3.9 Environmental Alerts

Table 3.9 shows the Environmental Alerts functional requirement.

Table 3.9 : Environmental Alerts

FR-09	Environmental Alerts
Description	The system must monitor real-time air quality in the user's area to issue timely alerts and preventive health advice that may impact respiratory health.
Priority	High

3.3.10 AI Chatbot Support

Table 3.10 shows the AI Chatbot Support functional requirement.

Table 3.10 AI Chatbot Support

FR-10	AI Chatbot Support
Description	The system must include an integrated AI chatbot for real-time assistance, providing immediate support and answering common queries.
Priority	High

3.3.11 Patient History Tracking

Table 3.11 shows the Patient History Tracking functional requirement

Table 3.11 Patient History Tracking

FR-11	Patient History Tracking
Description	The system must maintain a detailed, comprehensive record of each user's past symptoms, diagnoses, AI predictions, and care plans.
Priority	High

3.3.12 Report Generation

Table 3.12 shows the Report Generation functional requirement

Table 3.12 Report Generation

FR-12	Report Generation
Description	The system must allow users to generate and download detailed health reports summarizing their symptoms, diagnostic results, and care plans.
Priority	High

3.3.13 Exercise Recommendation

Table 3.13 shows the Exercise Recommendation functional requirement

Table 3.13: Exercise Recommendation

FR-13	Exercise Recommendation
Description	The system must suggest appropriate breathing exercises and physical activities tailored to the user's specific respiratory condition to improve lung function.
Priority	High

3.3.14 Schedule Doctor appointment

Table 3.14 shows the doctor appointment functional requirement

Table 3.14 Schedule Doctor appointment

FR-14	Schedule Doctor appointment
Description	The system must allow user to request doctors for appointment
Priority	High

3.3.15 See Appointment Request

Table 3.15 shows the see appointment request functional requirement

Table 3.15 see appointment request

FR-15	See Appointment Request
Description	The system must allow doctor to see appointment requests
Priority	High

3.4 Non-Functional Requirements

Table 3.16 shows non-functional requirements specify the criteria for judging the quality of the system's operation.

Table 3.16: Non-functional requirements

ID	Non-Functional Requirement	Description
NFR-1	AI Processing Speed	The AI-powered analysis and Smart Diagnosis System must process and return a prediction in a timely manner to provide real-time insights
NFR-2	Scalability	The full-stack architecture must support scalability to efficiently manage a potentially large volume of user health data
NFR-3	Data Privacy	The system must safeguard user data privacy to ensure secure access to personal health information.
NFR-4	Authentication Security	The complete authentication system must be secure to prevent unauthorized access to personal health records.
NFR-5	User Interface	The interface must be dynamic, responsive, and intuitive to make it easy for users to submit symptoms and upload or record sounds.
NFR-6	Accessibility	The web application must be accessible, bridging the gap for individuals who have limited access to specialized healthcare facilities or affordable screening tools.
NFR-7	System Integrity	The system must reliably analyze respiratory patterns and cross-reference symptom data for comprehensive and accurate insights.

3.5 Use Case Diagram

A Use Case Diagram visually represents the functionality of the BreatheWell system in terms of actors (users) and use cases (functions). The primary actor is the Registered User.

Figure 3.1 shows the use case diagram for our system.

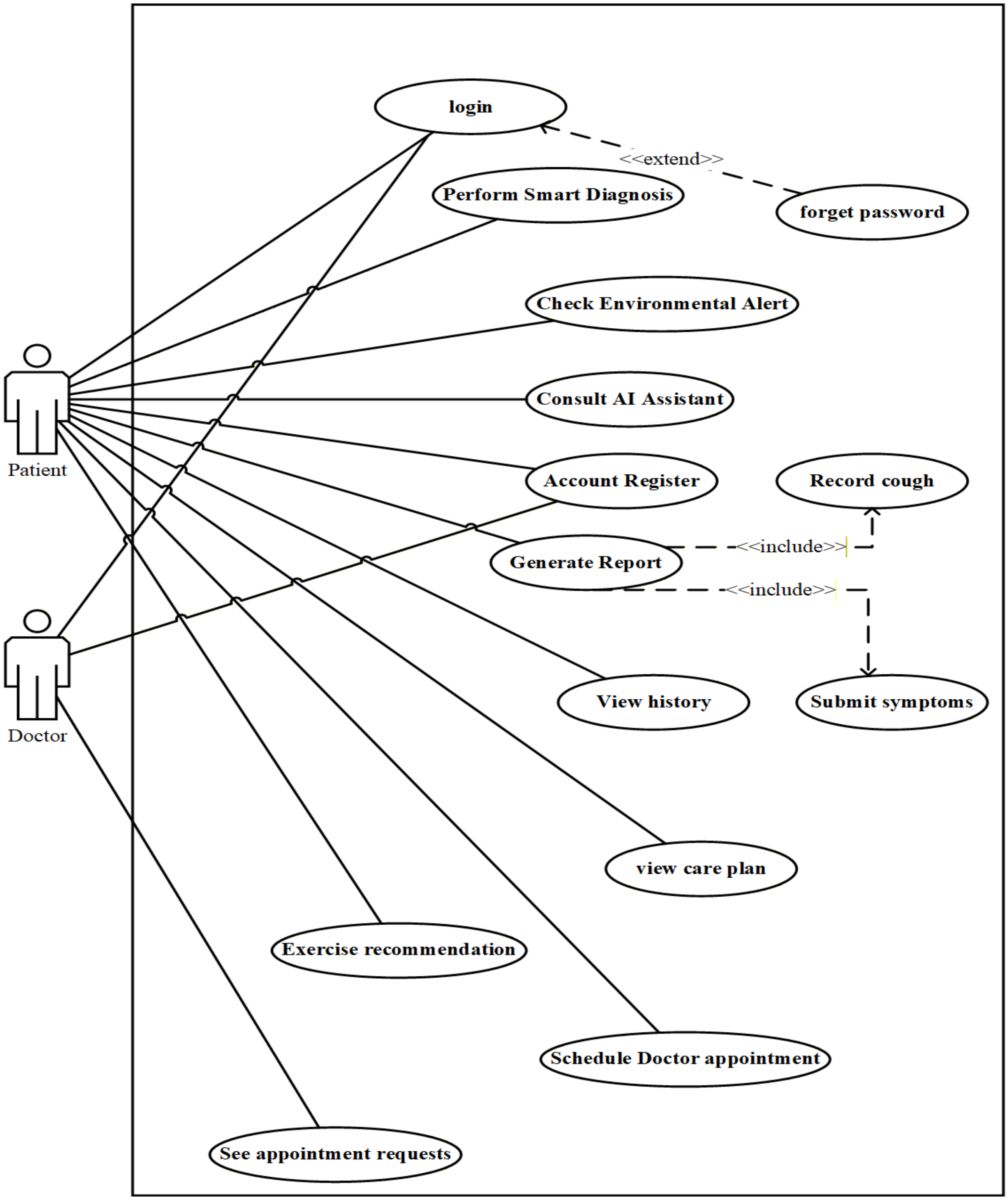

Figure 3.1 use case diagram

3.6 Use Cases Description

Detailed descriptions for use cases that define the functionality of BreatheWell.

3.6.1 Perform Smart Diagnosis

The system must combine submitted symptom data with the results of the audio analysis to provide an AI-driven prediction of potential lung conditions

Table 3.17: Perform Smart Diagnosis

Use Case ID	UC-01
Use Case Name	Perform Smart Diagnosis
Primary Actor	Registered User
Goal	To obtain an AI-driven prediction of a potential respiratory condition and receive a personalized care plan.
Preconditions	The User must be logged into the system .
Flow of Events	1. User navigates to the "Check-up" section. 2. User submits their current respiratory symptoms . 3. System stores symptom data. 4. User records or uploads a sample of their cough/breathing sounds . 5. System stores the audio file. 6. System's AI model analyzes the audio for patterns like wheezing or irregular breathing. 7. System's Smart Diagnosis module combines symptom data and audio analysis results. 8. System generates a predicted diagnosis. 9. System displays the diagnosis and a personalized care plan.
Postconditions	A new record of the check-up, including symptoms, audio analysis, and diagnosis, is saved in the Patient History .

Cross reference	FR-06

3.6.2 View History

The system must provide a visual platform (dynamic dashboard) where users can monitor their symptoms over time, track trends, and view detailed health reports.

Table 3.18: View History

Use Case ID	UC-02
Use Case Name	View History
Primary Actor	Registered User
Goal	To review historical health data and observe symptom trends over time.
Preconditions	The User must be logged in and have prior check-up data recorded (FR-10).
Flow of Events	1. User navigates to the "Dashboard" section. 2. Select patient history 3. System retrieves the user's detailed health history, including submitted symptoms, diagnostic results, and care plans . 3. System visually displays the data, showing trends and progress over time .
Postconditions	The user has reviewed their respiratory health journey, and a downloadable health report may be generated.
Alternative flow	3a. If user is new then show no record found
Cross reference	FR-11

3.6.3 Check Environmental Alert

The system must monitor real-time air quality in the user's area to issue timely alerts and preventive health advice that may impact respiratory health.

Table 3.19: Check Environmental Alert

Use Case ID	UC-03
Use Case Name	Check Environmental Alert
Primary Actor	Registered User
Goal	To be proactively warned about local air quality conditions that may affect respiratory health.
Preconditions	The User is logged in and the system has access to the user's location or pre-set location.
Flow of Events	1. System continuously integrates real-time air quality data (using public APIs). 2. System analyzes the data against thresholds that may impact respiratory health. 3. If a risk is detected, the system issues a timely alert to the user . 4. The alert includes preventive advice to help the user avoid exposure to harmful environmental conditions. 5. User views the notification on the dashboard or in an alert system area.
Postconditions	The user is informed of the environmental risk and advised on precautionary actions.
Cross reference	FR-10

3.6.4 Account Register

The system shall enable the user to register their account. The system shall provide a login to the user.

Table 3.20: Account Register

Use Case ID	UC-04
Use Case Name	Account Register

Primary Actor	User (patient or doctor)
Goal	To securely access or register for the BreatheWell system.
Preconditions	The User must have an active internet connection.
Flow of Events	1. User selects "Sign Up" option. 2. User provides required details (name, email, secure password). 3. System hashes and salts the password before storage . 4. System creates a new user account and directs the user to the Dashboard.
Postconditions	The User is either securely logged into the system or a new, registered account has been created .
Alternative flow	3a. If user already exist show user already exist, please login
Cross reference	FR-01

3.6.5 Consult AI Assistant

The system must include an integrated AI chatbot for real-time assistance, providing immediate support and answering common queries.

Table 3.21 Consult AI Assistant

Use Case ID	UC-05
Use Case Name	Consult AI Assistant
Primary Actor	Registered User
Goal	To receive immediate, real-time answers or guidance on general respiratory health queries.

Preconditions	The User must be logged into the system.
Flow of Events	1. User selects the "AI Chatbot" icon or section. 2. System loads the Chatbot interface . 3. User types a query related to symptoms, conditions, or general health. 4. The Chatbot processes the natural language input. 5. Chatbot retrieves the most relevant information or suggested next steps (e.g., "submit a check-up"). 6. Chatbot displays the response to the user in real-time. 7. User continues the conversation or closes the chatbot window.
Postconditions	The User has received information or assistance, and the interaction is logged for future model training/improvement.
Cross reference	FR-05

3.6.6 Generate Report

The system must allow users to generate and download detailed health reports summarizing their symptoms, diagnostic results, and care plans.

Table 3.22: Generate Report

Use Case ID	UC-06
Use Case Name	Generate Report
Primary Actor	Registered User
Goal	To produce a printable/downloadable summary of the User's health history for sharing with a healthcare provider.
Preconditions	The User must be logged in and have prior health data recorded
Flow of Events	1. User navigates to the "Patient History" or "Dashboard" section.

	2. User selects the "Generate Report" option. 3. System generate the report based on the symptoms and cough diagnosis result 4. System formats the data into a structured, professional PDF document. 5. System presents the User with the option to view, print, or download the generated report.
Postconditions	A comprehensive, structured health report is generated and made available to the User .
Alternative flow	3a. if user did not perform AI diagnosis at least once, Prompt for symptoms and cough diagnosis before generating report
Cross reference	FR-12

3.6.7 Submit Symptoms

The system must allow users to submit their symptoms to predict possible respiratory condition.

Table 3.23: Submit Symptoms

Use Case ID	UC-07
Use Case Name	Submit Symptoms
Primary Actor	Registered User
Goal	To submit symptoms for AI-driven prediction
Preconditions	The User must be logged into the system.
Flow of Events	1. User navigates to the "Check-up" section. 2. User submits their current respiratory symptoms. 3. System stores symptom data.
Postconditions	A new record of the symptoms is saved in the Patient History.

Cross reference	FR-03

3.6.8 Record Cough

The system must allow users to submit their symptoms to predict possible respiratory condition.

Table 3.24: Record Cough

Use Case ID	UC-08
Use Case Name	Record Cough
Primary Actor	Registered User
Goal	To record cough for AI-driven prediction
Preconditions	The User must be logged into the system (FR-1).
Flow of Events	1. User navigates to the "Check-up" section. 2. User submits their current respiratory symptoms. 3. System stores symptom data. 4. User records or uploads a sample of their cough sounds. 5. System stores the audio file.
Postconditions	A new record of the cough is saved in the Patient History.
Cross reference	FR-04

3.6.9 View Care Plan

The system must allow users to see detailed health care plan summarizing their symptoms, diagnostic results.

Table 3.25: View Care Plan

Use Case ID	UC-09
Use Case Name	View Care Plan

Primary Actor	Registered User
Goal	To see user’s care plan based on the symptoms and result.
Preconditions	The User must be logged in and have prior health data recorded
Flow of Events	1. User navigates to the "Dashboard" section. 2. User selects the "view care plan" option . 3. System securely retrieves the care plan. 4. System display the detail care plan.
Postconditions	A comprehensive, structured health care plan is generated and made available to the User .
Alternative flow	3a. If user is new system shows record not found
Cross reference	FR-07

3.6.10 Schedule Doctor Appointment

The system must allow users to get appointment from a doctor.

Table 3.26: Schedule Doctor Appointment

Use Case ID	UC-10
Use Case Name	Schedule Doctor Appointment
Primary Actor	Registered User
Goal	To get an appointment from a doctor.
Preconditions	The User must be logged in and have prior health data recorded
Flow of Events	1. User navigates to the "Dashboard" section. 2. User selects the "doctor appointment" option . 3. System show available doctors

	4. System securely retrieves the user’s details like symptoms and allow user to send to the doctor.
Postconditions	System show an appointment detail with date and location.
Alternative flow	4. Show patient report not found, if user try to request appointment without AI diagnosis.
Cross reference	FR-14

3.6.11 Exercise Recommendation

The system allow users to see detailed recommended exercise based on the patient’s condition.

Table 3.27: Exercise Recommendation

Use Case ID	UC-11
Use Case Name	Exercise Recommendation
Primary Actor	Registered User
Goal	To see recommended exercise.
Preconditions	The User must be logged in and have prior health data recorded
Flow of Events	1. User navigates to the "Dashboard" section. 2. User selects the "exercise" option . 3. System securely retrieves the specific recommended exercise. 4. System display the detail exercise steps.
Alternative flow	3a. If user is new and no any patient report then show no recommendation yet.
Postconditions	The recommended exercise details are successfully displayed to the User.
Cross reference	FR-13

3.6.12 See Appointment Request

The system allow doctor to see appointment request with patient details.

Table 3.28: See Appointment Request

Use Case ID	UC-12
Use Case Name	See Appointment Request
Primary Actor	Registered doctor
Goal	To see appointment requests from patient and schedule an appointment.
Preconditions	The User must be logged in
Flow of Events	1. User navigates to the "Dashboard" section. 2. User selects the " appointment requests " option . 3. System list the appointment requests.
Postconditions	The request details are successfully displayed to the doctor.
Alternative flow	3a. No request yet at the movement if request did not send from any patient.
Cross reference	FR-15

3.7 Summary

Chapter 3 has established the foundational requirements specification for the BreatheWell system. It defined the necessary hardware and software resources and clearly detailed the expected functional requirements, covering all core features from secure authentication to the AI-driven Smart Diagnosis. Crucially, non-functional requirements were established to ensure the system is secure, performs well, and is reliable. Finally, a Use Case Diagram and Use Case Descriptions provided a clear, structural view of how the user will interact with the system's key functionalities, paving the way for the detailed system design and architecture in the subsequent chapter.

CHEPTER 4
SYSTEM MODELING

4.1 Introduction

This chapter focuses on the architectural design and modeling of the BreatheWell system, translating the defined functional and non-functional requirements (from Chapter 3) into a clear technical structure. It details the overall system architecture, defines the user interface design principles, and utilizes the 4+1 View Model to present a multi-faceted view of the software architecture. This modeling effort lays the foundation for the successful implementation of the AI-powered respiratory health monitoring application.

4.2 System Design

The BreatheWell system adopts a modern Three-Tier Architecture integrated with specialized services for Artificial Intelligence and external data retrieval. This approach ensures separation of concerns, scalability, and maintainability.

4.2.1 Architectural Overview (Three-Tier Model)

The system is divided into three logical layers:

1. **Presentation Tier (Frontend)**
   - **Technology:** Developed using React.js.
   - **Function:** Handles the User Interface (UI), user experience (UX), and client-side interactions. This layer runs on the user's browser or mobile device, allowing users to submit symptoms, record/upload audio, view dashboards, and interact with the AI chatbot.
2. **Application/Logic Tier (Backend API)**
   - **Technology:** Developed using Python (Flask/Django) or Node.js.
   - **Function:** Acts as the intermediary between the frontend and the data/AI services. It executes all business logic, including user authentication, processing form submissions, managing sessions, calling the AI Analysis Service, generating personalized care plans, and serving data to the dashboard.
3. **Data Tier (Databases & Storage)**
   - **Technology: MongoDB** (for potentially unstructured or rapidly changing data, like chatbot logs). Dedicated storage for user-uploaded audio files.
   - **Function:** Securely stores and manages all application data, including user profiles, symptom history, diagnostic results, and configuration settings.

4.2.2 Specialized Services and Interoperability

The core functionality relies on two main services integrated with the Logic Tier:

1. **AI Analysis Service:** A dedicated microservice, likely implemented in Python with TensorFlow/PyTorch, responsible solely for running the trained Machine Learning models (FR-4). It receives raw audio data from the Logic Tier and returns the analyzed respiratory patterns and smart diagnosis prediction (FR-5).
2. **External API Gateway:** Handles communication with third-party services, primarily the Air Quality Index (AQI) API to fetch real-time environmental data necessary for Environmental Alerts.

4.3 Interface Design

Interface design focuses on creating a user-friendly, accessible, and intuitive experience. The design adheres to modern web standards, prioritizing responsiveness for optimal use on both mobile and desktop devices.

4.3.1 User Interface (UI) Components

- **Dashboard:** A central hub showing the user's current health status, recent diagnosis, and key historical trends. It utilizes clear visual aids like charts and graphs.
- **Check-up Module:** A multi-step, clean interface guiding the user through symptom submission and audio recording/upload. Visual feedback is crucial here to confirm successful recording.
- **Alerts Panel:** A dedicated area for prominently displaying Environmental Alerts and appointment notifications.
- **Navigation:** A simple, persistent navigation structure (sidebar or bottom tab bar on mobile) provides quick access to main sections:

4.3.2 Design Principles

1. **Clarity and Simplicity:** Minimalist design to reduce cognitive load, focusing user attention on health data and next steps.
2. **Visual Feedback:** Immediate feedback for critical actions, such as confirming audio upload or diagnosis calculation.
3. **Accessibility:** Use of high contrast color palettes and proper font sizes to ensure readability for users of all abilities.

4.4 The 4+1 View Model of Architecture

The 4+1 View Model, introduced by Philippe Kruchten in 1995, provides a comprehensive structure for documenting software architecture by separating it into five concurrent views. This separation ensures that the unique concerns of all stakeholders from end-users to system engineers are effectively addressed.

The model consists of:

Logical View: Captures the system's functionality and key abstractions.

Process View: Deals with the dynamic aspects of runtime behavior.

Development View: Focuses on the static organization of the software in its development environment.

Physical View: Illustrates the system's deployment onto hardware nodes.

Scenarios (+1): The unifying view that validates the architecture through practical use cases.

Figure 4.1 represents the 4 + 1 view model of Architecture.

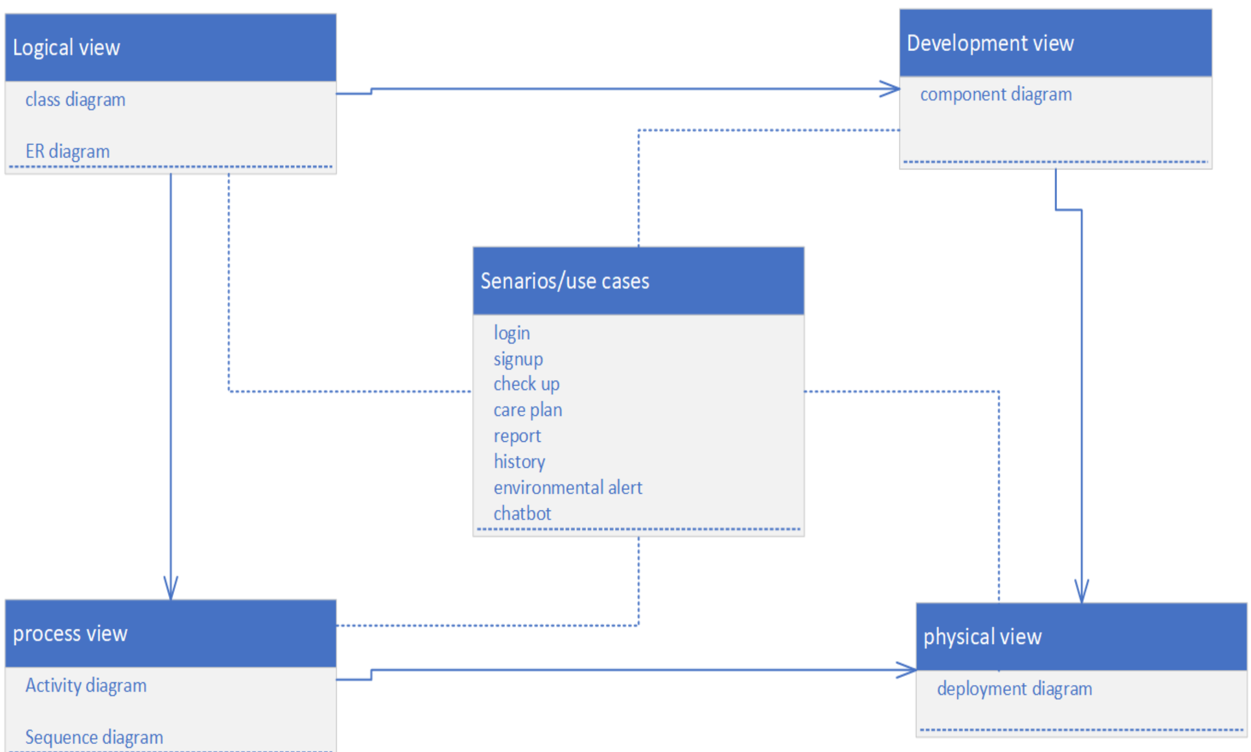

Figure 4.1 4+1 view model

4.4.1 Logical View

This view focuses on the system's functional requirements describing the design object model and the system's class and module structure.

4.4.1.1 UML Class Diagram

The Class Diagram is used to show the static, object-oriented structure of the system's business concepts. It is essential for the software developer stakeholder.

- **Focus:** Core system concepts, their attributes, methods (functions), and relationships (inheritance, association, aggregation).
- **Purpose in BreatheWell:** To model the classes in the application logic layer.

Figure 4.2 shows the class diagram of our system (BreatheWell).

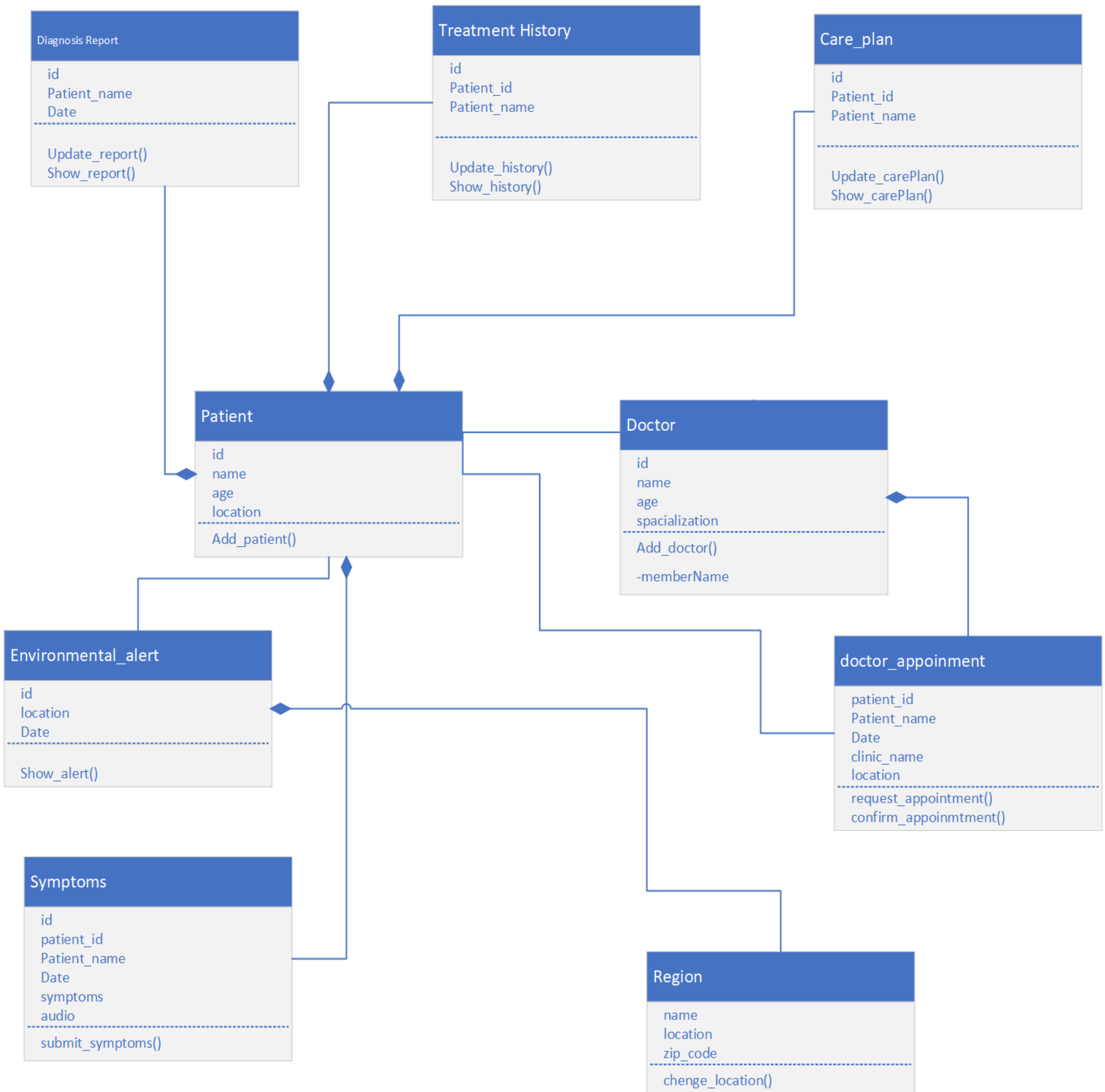

Figure 4.2 class diagram

4.4.1.2 Entity-Relationship Diagram (ERD)

The ERD is used to model the structure of the data stored in the database. It is essential for the database designer and helps validate the data-centric requirements.

- **Focus:** Entities (tables), attributes (columns), and the relationships between entities (one-to-one, one-to-many, many-to-many).
- **Purpose in BreatheWell:** To design the robust schema for the Data Tier (MongoDB). It ensures that all patient history and diagnostic data are securely and efficiently stored.

Figure 4.3 represent the Entity Relationship diagram of our system.

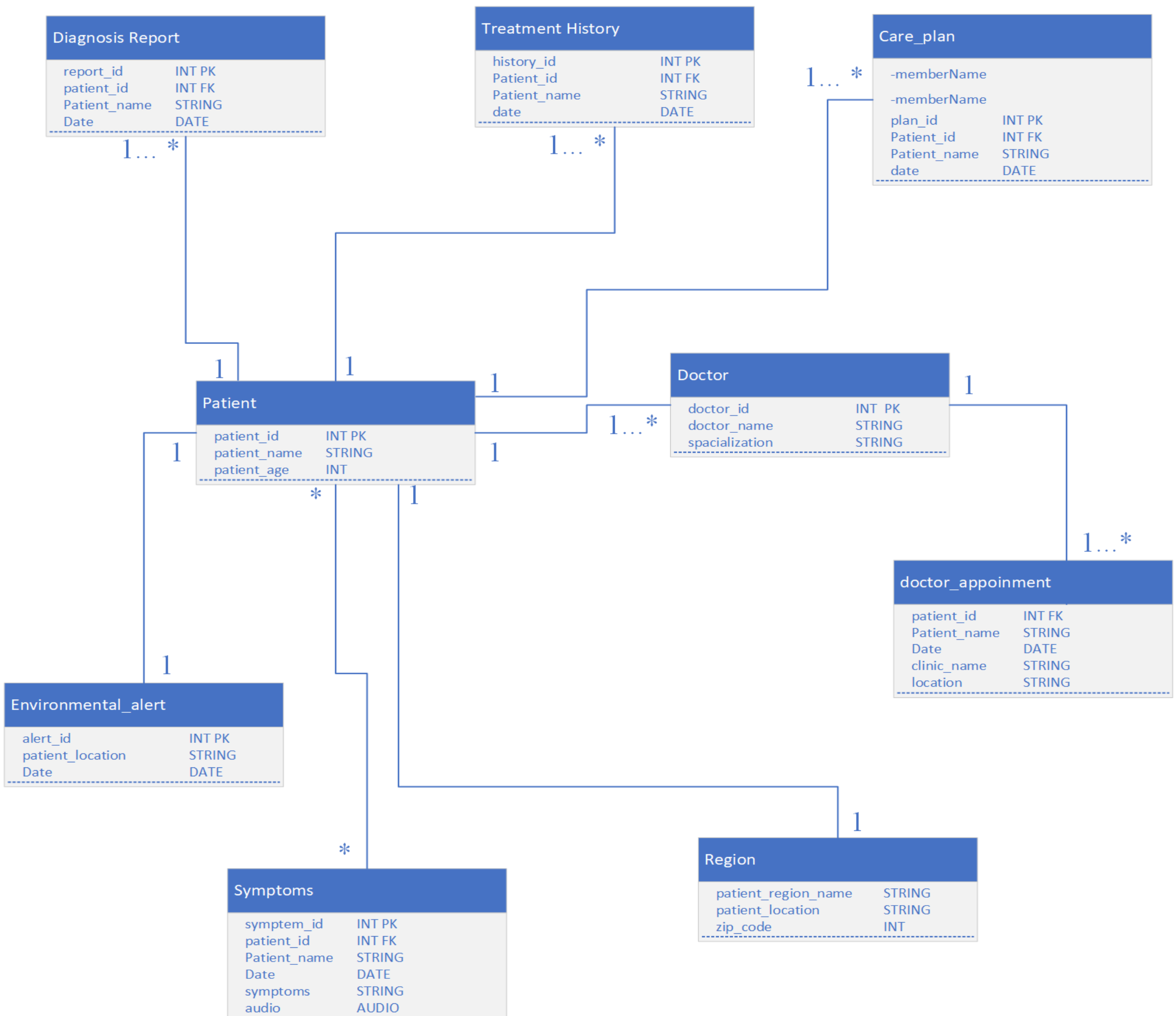

Figure 4.3 ERD for BreatheWell

4.4.2 Process View

The Process View addresses the system's dynamic aspects, focusing on the execution of tasks. It shows how the system will perform in different scenarios.

4.4.2.1 Activity Diagram

The Activity Diagram models the flow of control within the system, focusing on the step-by-step actions required to complete a business process. It is used to show the workflow from one activity to the next, often demonstrating parallel activities.

Figure 4.4 shows the overall activity diagram of our system.

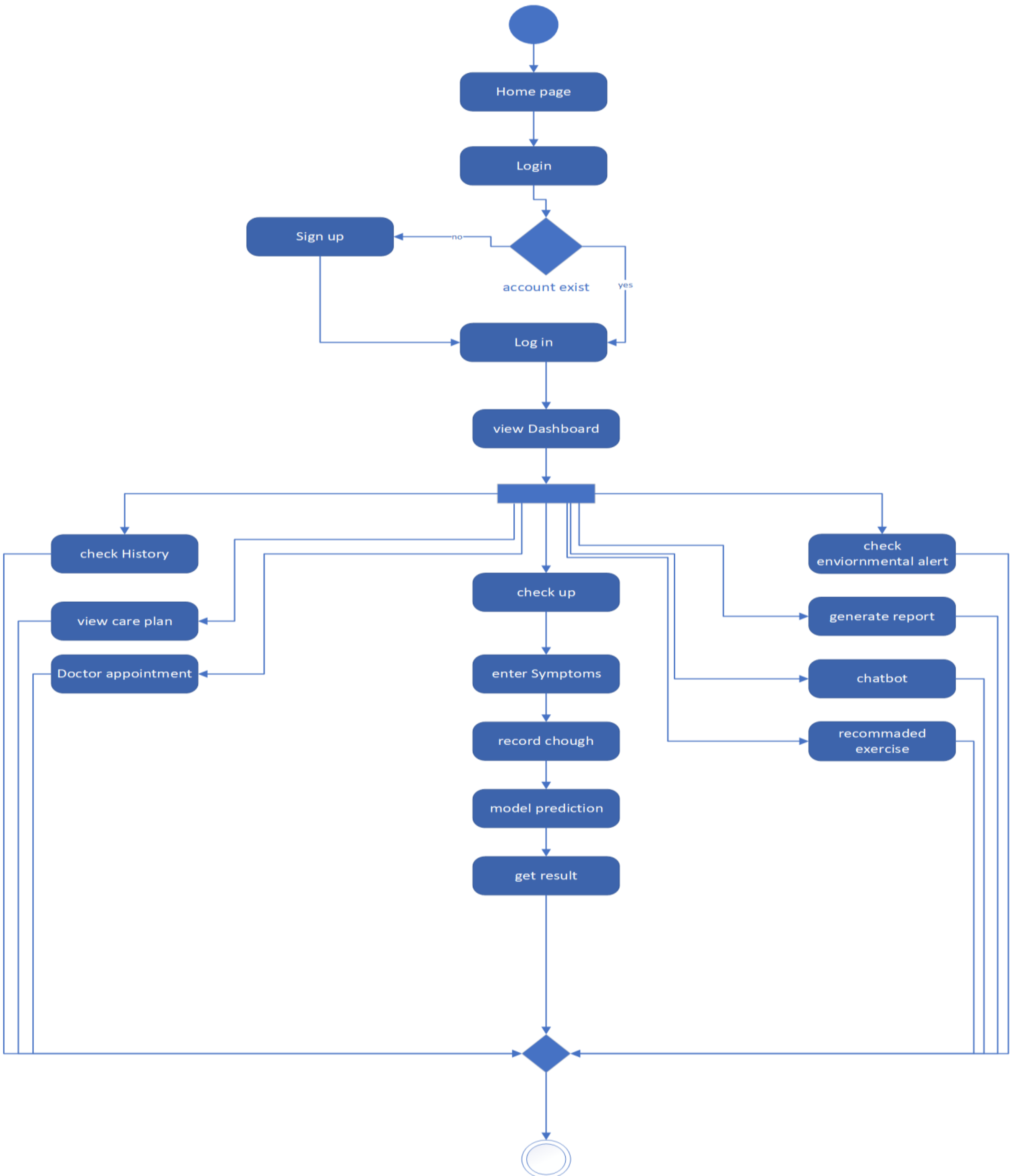

Figure 4.4 Activity diagram

4.4.2.2 Sequence Diagram

The Sequence Diagram models the time-ordered interactions between objects or components to complete a specific task.

4.4.2.2.1 Sequence Diagram for Smart Diagnosis Check-up

Figure 4.5 illustrates the interaction flow when a registered user performs a Smart Diagnosis Check-up using the BreatheWell system.

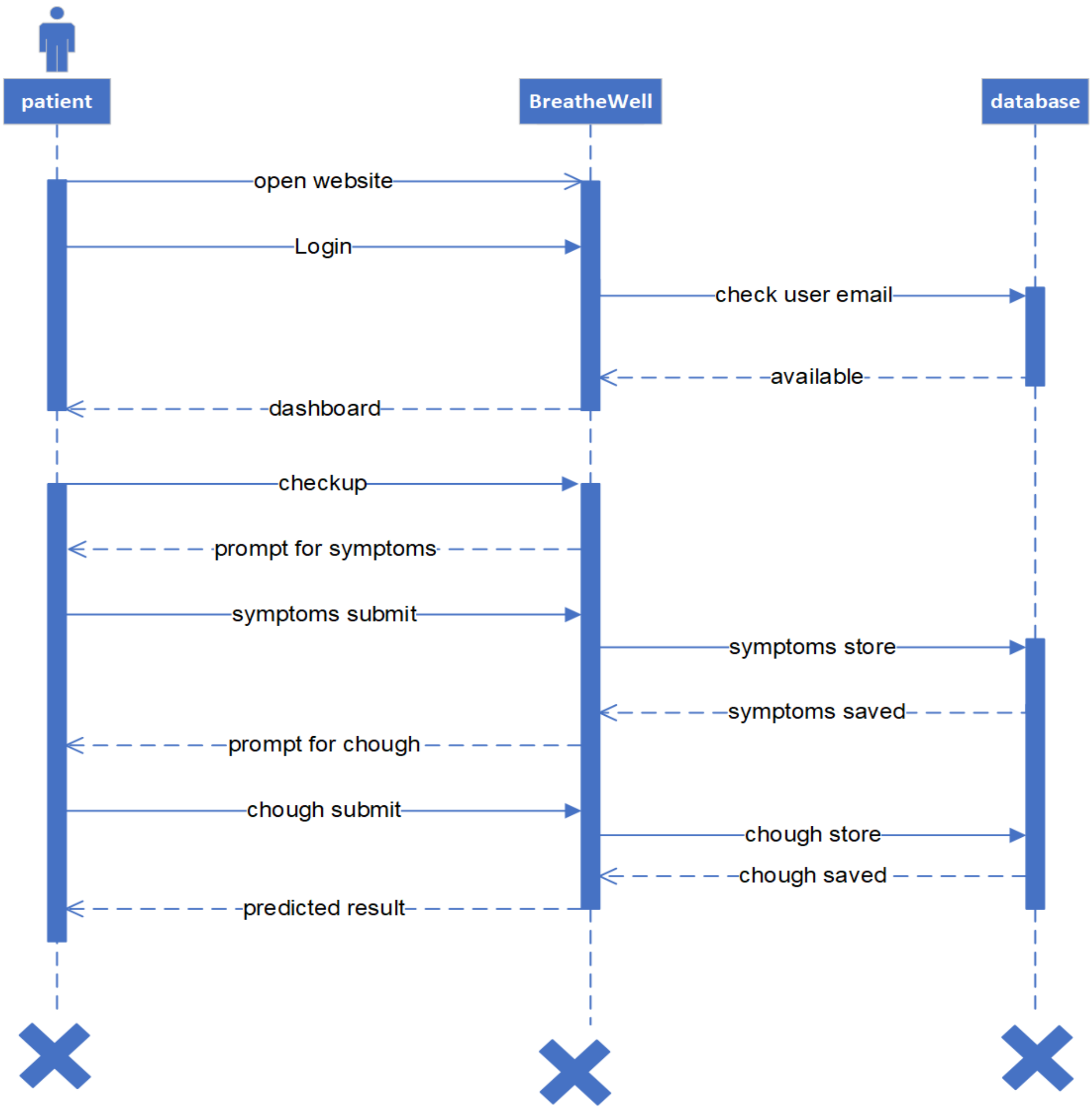

Figure 4.5 Sequence Diagram for Smart Diagnosis Check-up

4.4.2.2.2 Track Health History

Figure 4.6 models the flow when a user accesses their health history, symptom logs, and past diagnosis records.

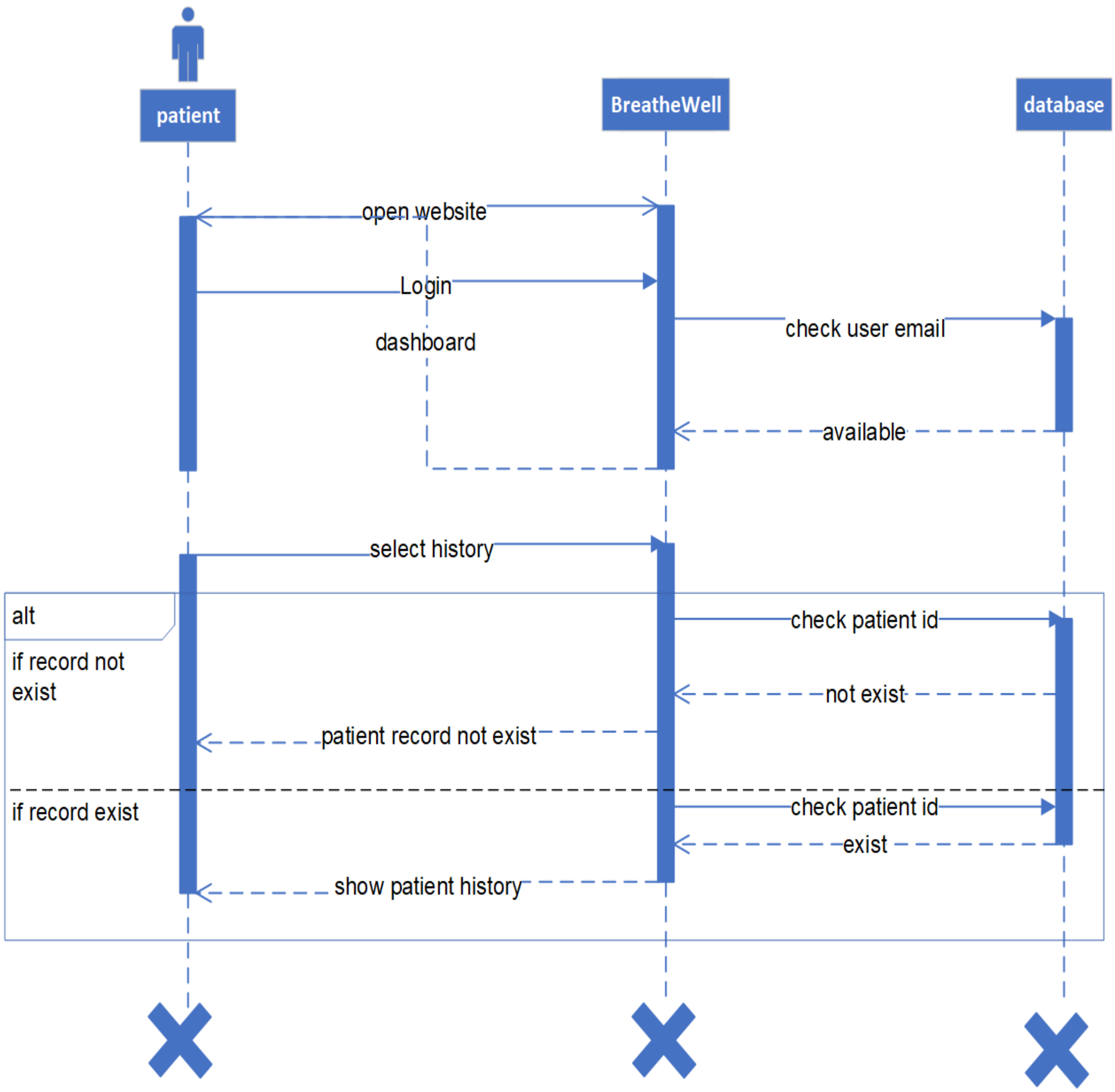

Figure 4.6 sequence diagram for Track Health History

4.4.2.2.3 Receive Environmental Alert

Figure 4.7 models the background process of checking Air Quality Index (AQI) and notifying the user .

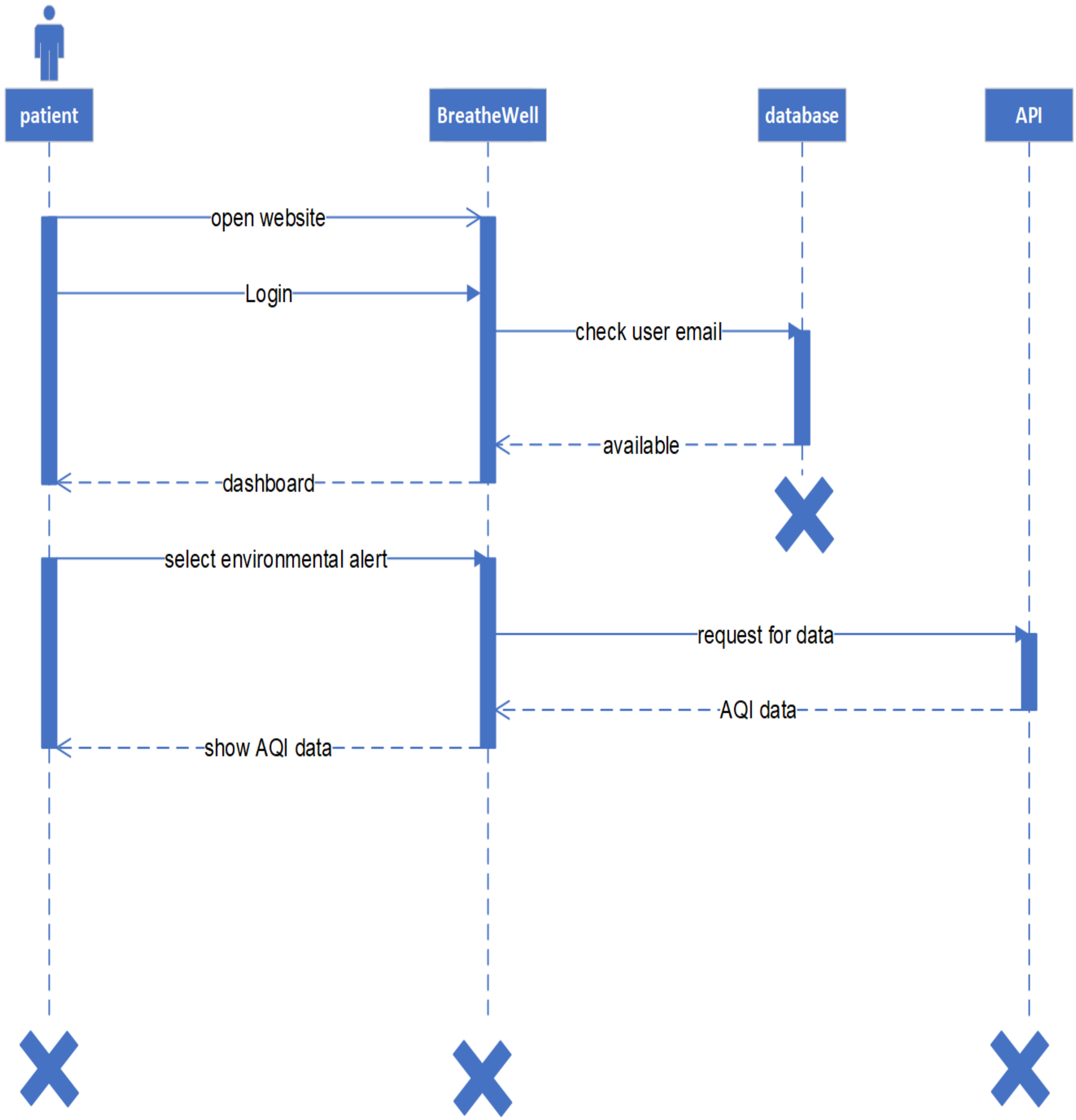

Figure 4.7 sequence diagram for Receive Environmental Alert

4.4.2.2.4 Interact with AI Chatbot

Figure 4.8 models a real-time, interactive session with the AI Chatbot

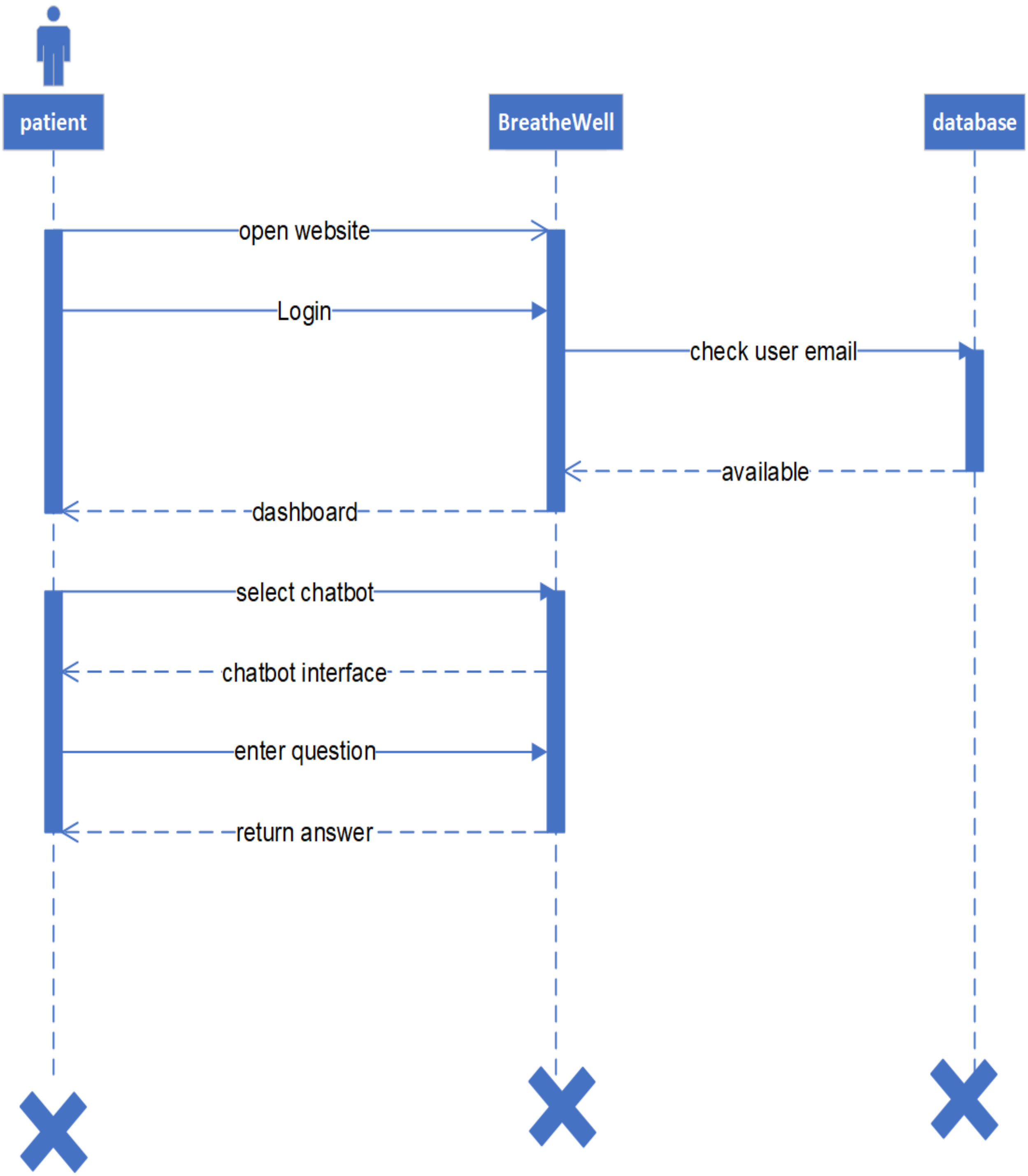

Figure 4.8 sequence diagram for Interact with AI Chatbot

4.4.2.2.5 Generate Report

Figure 4.9 models the process of compiling user data into a single, structured, and downloadable document.

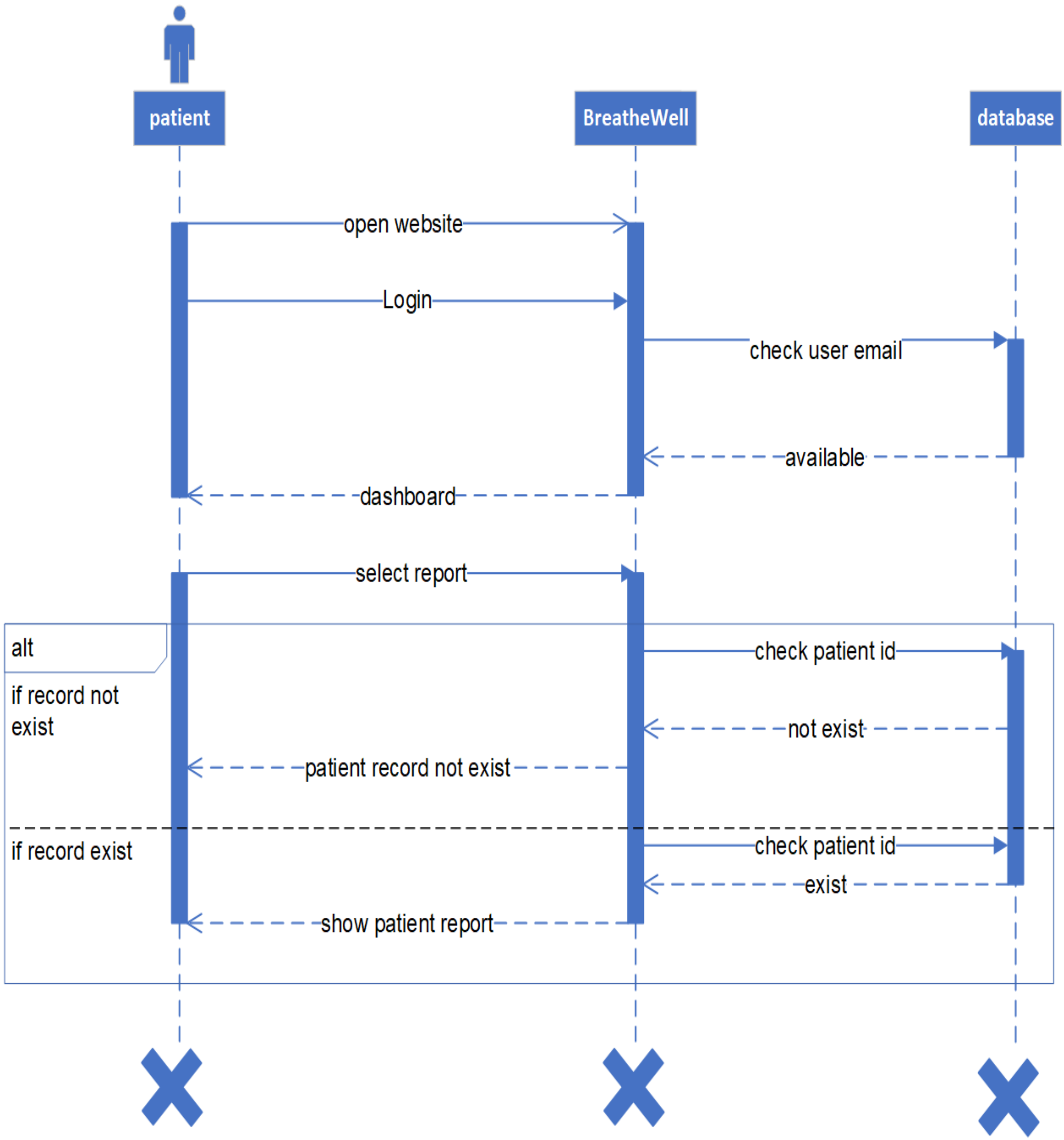

Figure 4.9 sequence diagram for Health Report

4.4.2.2.6 Account Register

Figure 4.10 models the process of creating a new user's account on the system.

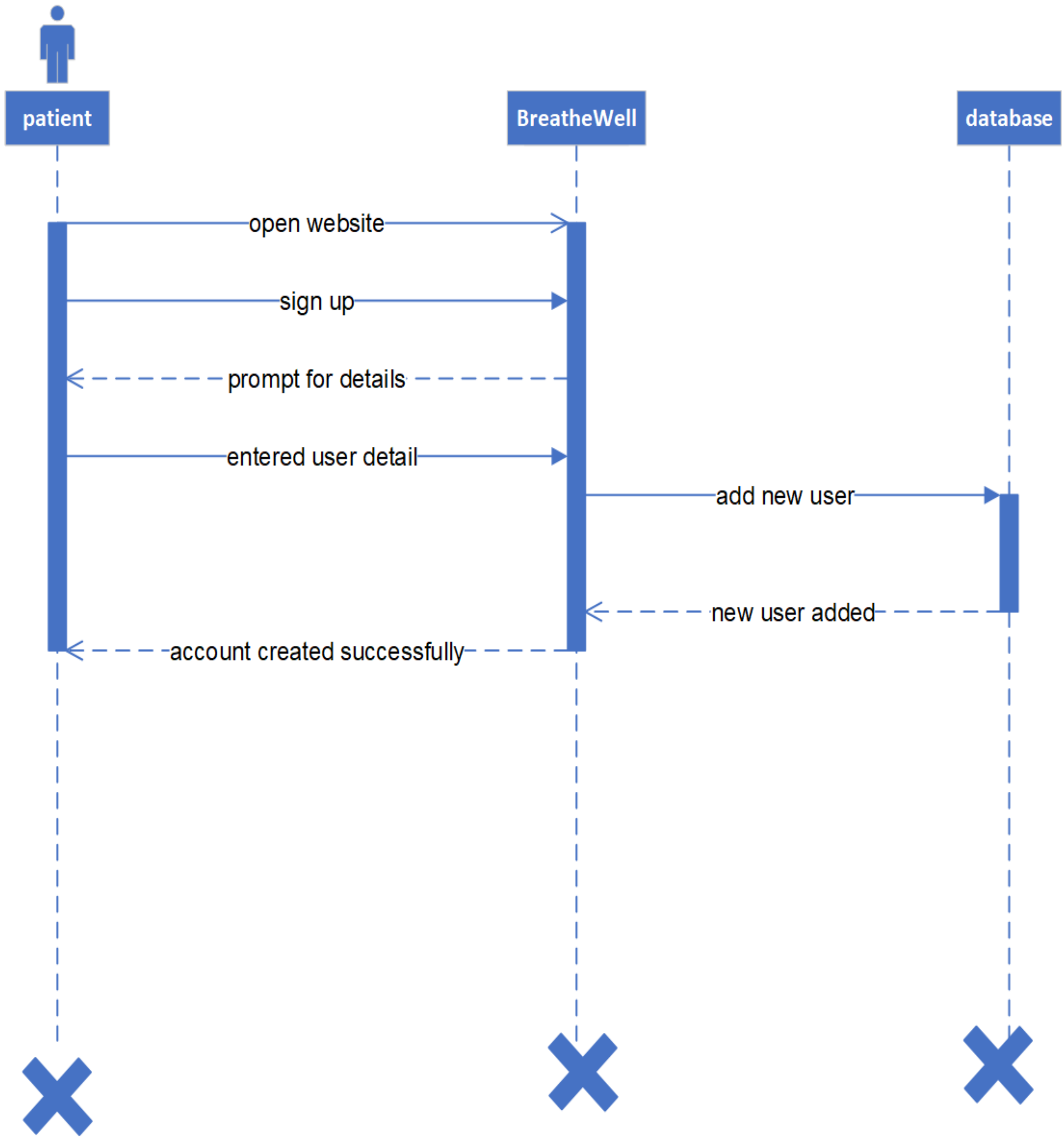

Figure 4.10 sequence diagram for account registration

4.4.2.2.7 Care plan

Figure 4.11 models the process to see the patient’s care plan

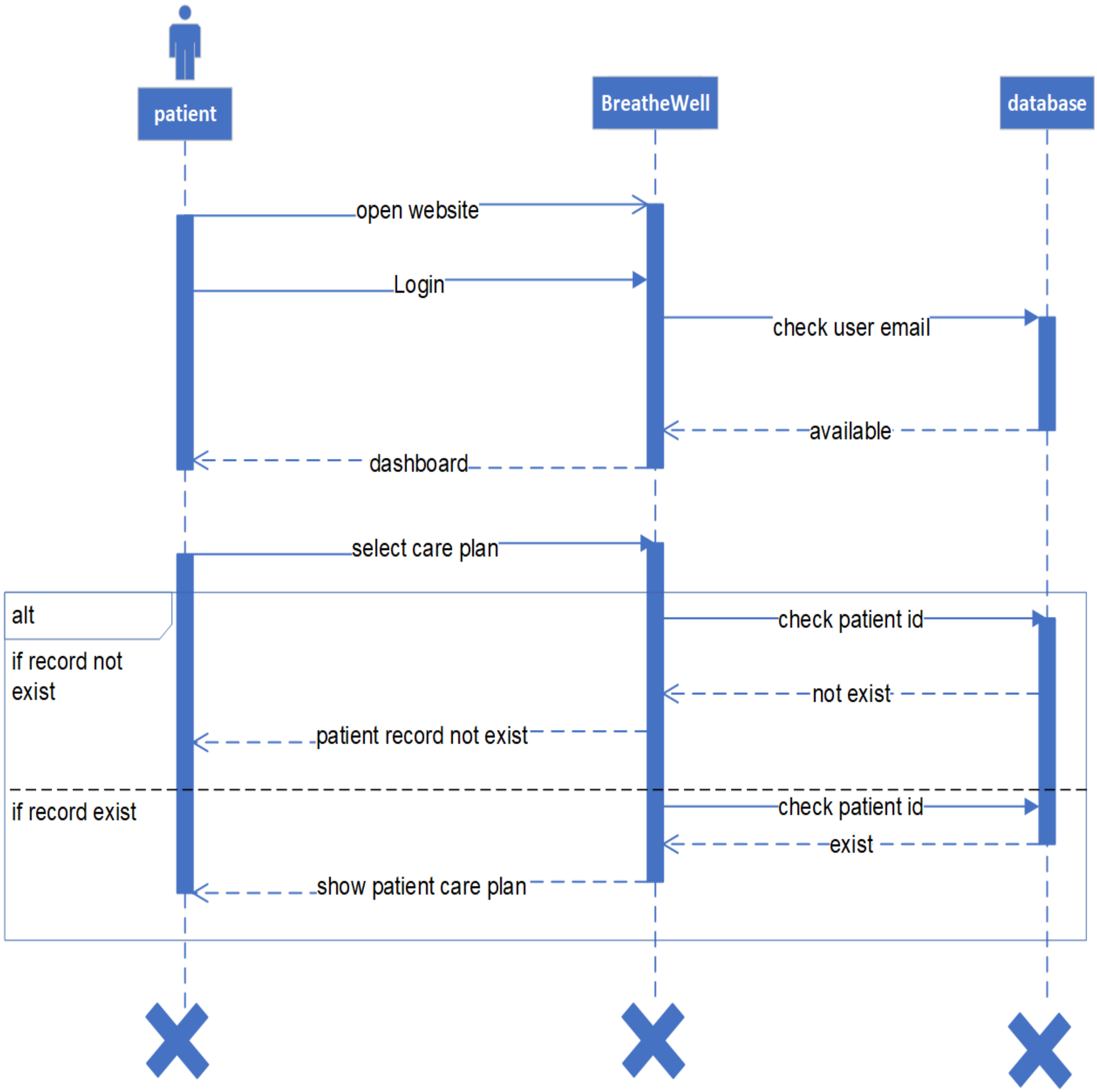

Figure 4.11 sequence diagram for care plan

4.4.2.2.8 Exercise Recommendation

Figure 4.12 models the process to see exercise recommendation

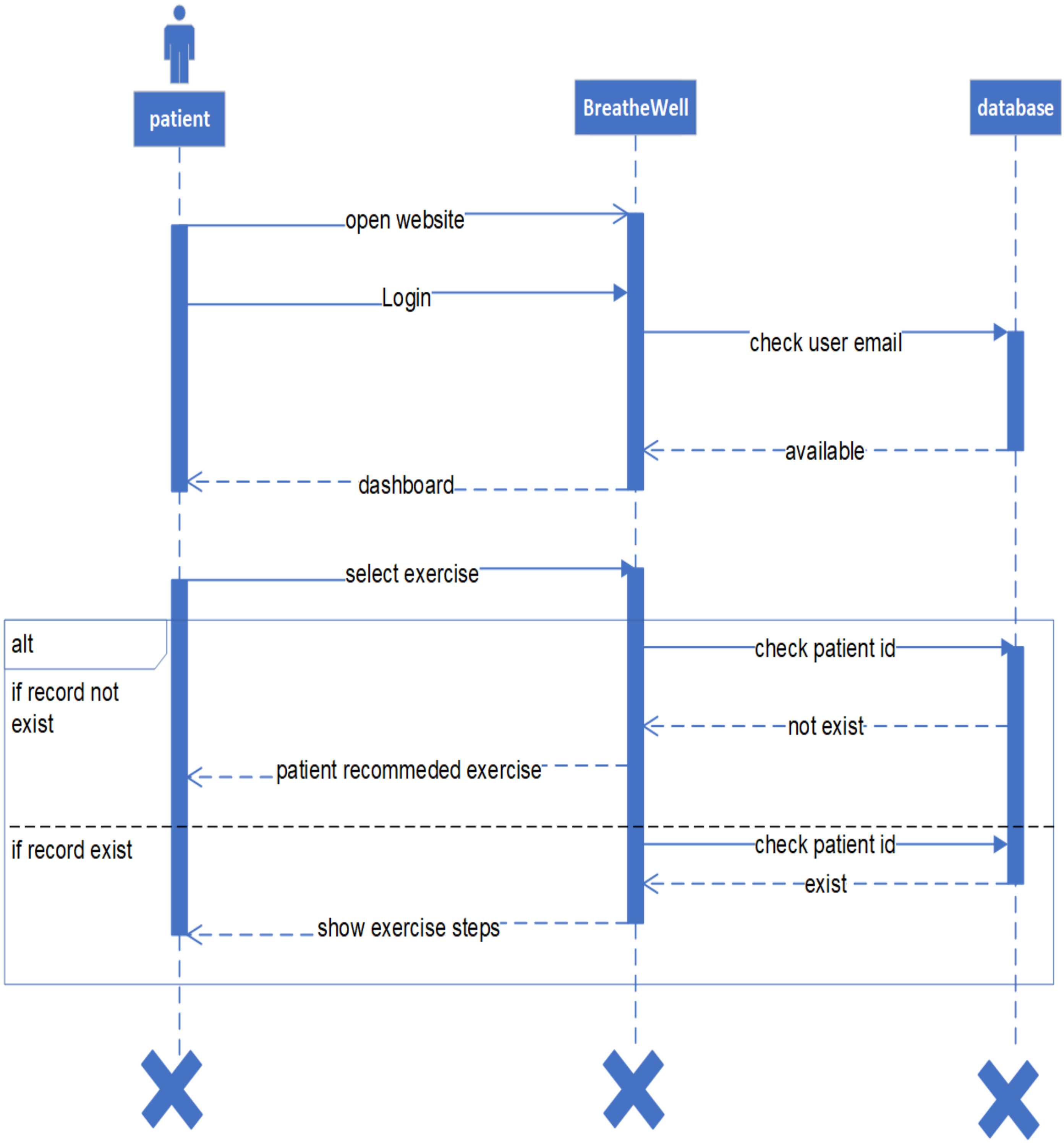

Figure 4.12 sequence diagram for exercise recommendation

4.4.2.2.9 Appointment Schedule

Figure 4.13 models the process to make a successful appointment.

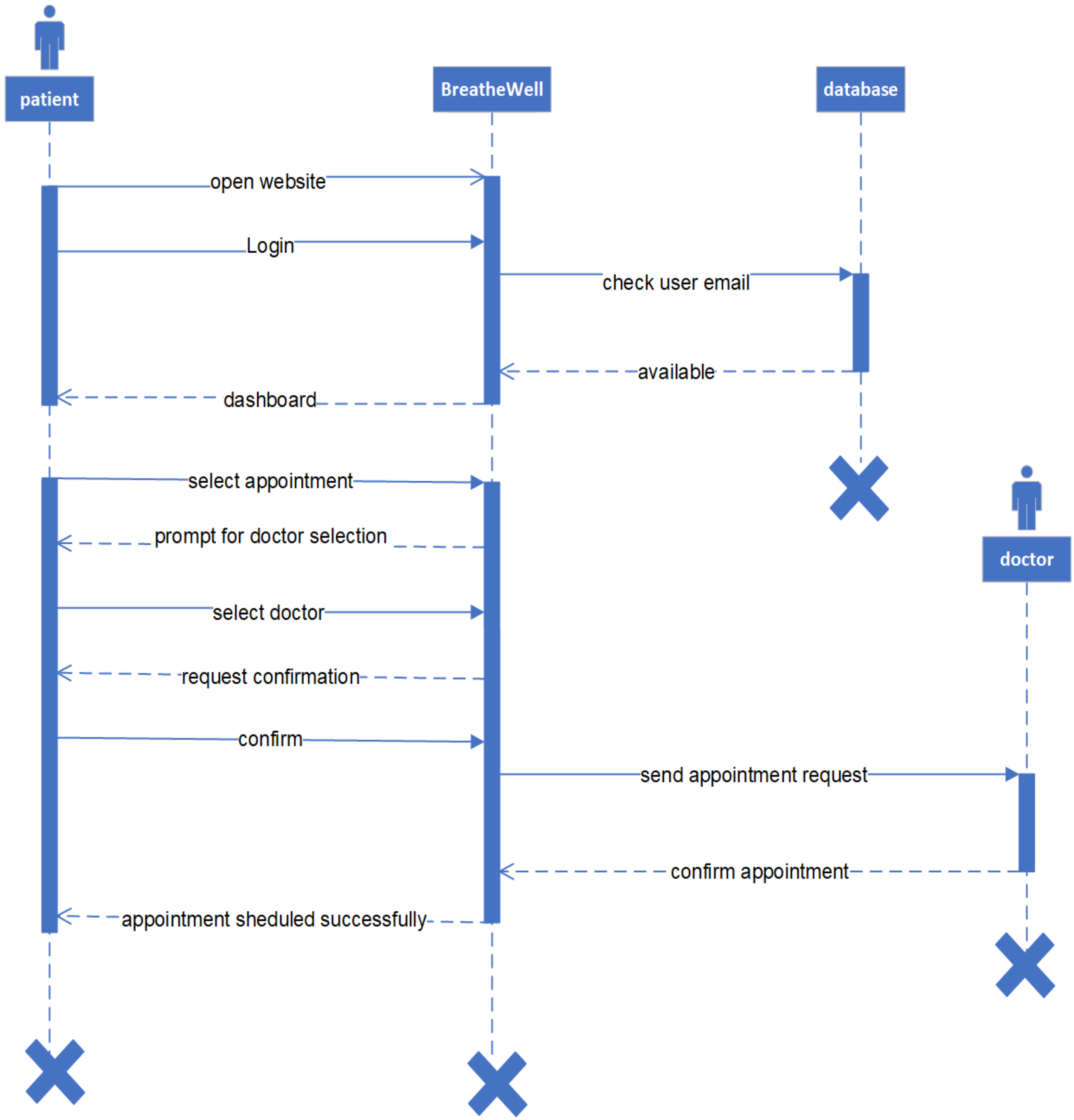

Figure 4.13 sequence diagram for appointment schedule

4.4.2.2.10 See Appointment

Figure 4.14 models the process of checking the appointment requests for a doctor.

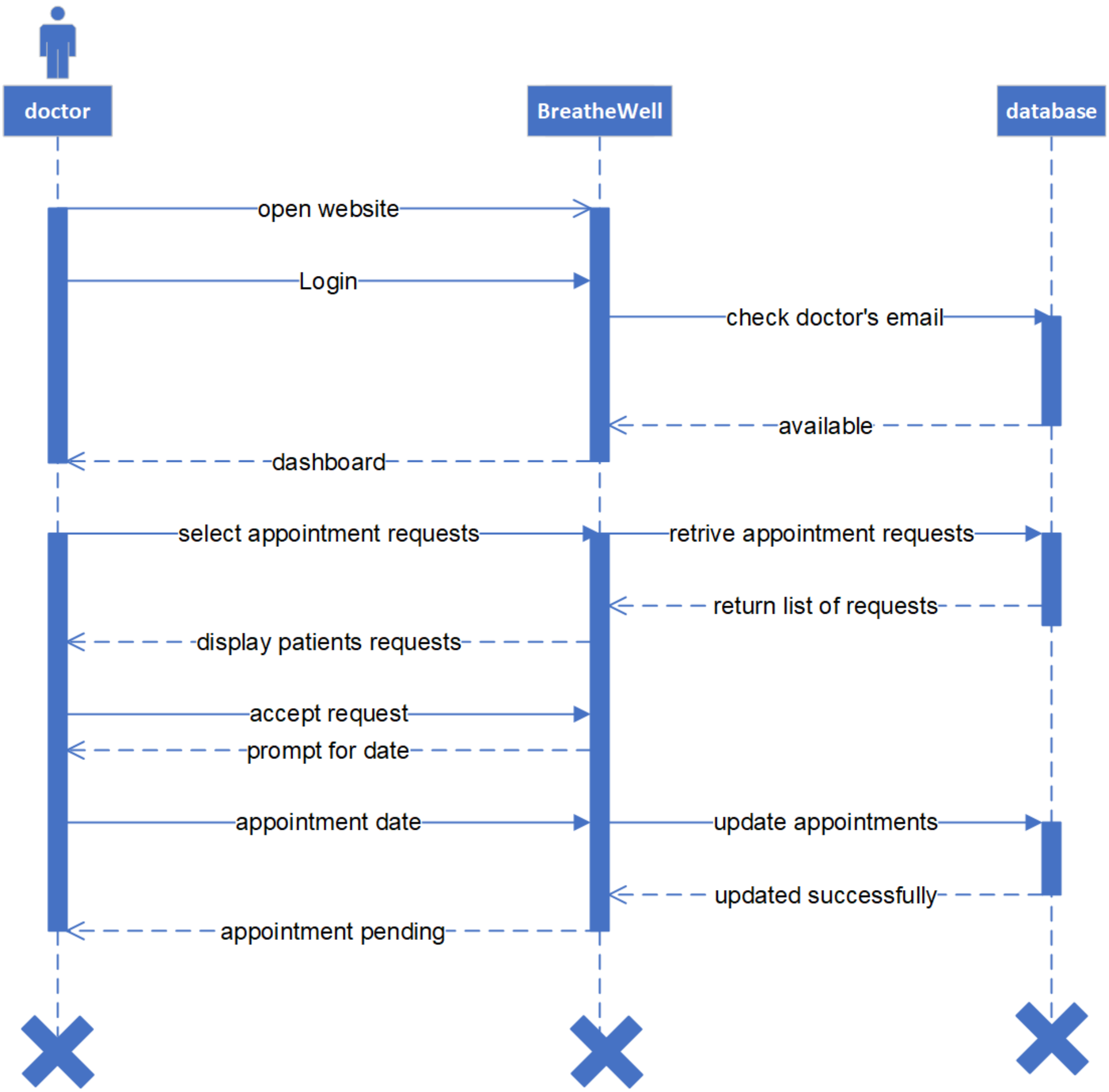

Figure 4.14 sequence diagram for see appointment request for doctor

4.4.3 Development View

This view describes the actual software implementation structure, focusing on the software organization, modules, sub-systems, and deployment strategy for developers.

4.4.3.1 Component Diagram for BreatheWell System.

The diagram illustrates how the major software components are structured, linked, and depend on each other to deliver the system's functionality. The system is divided into four main physical components: the Client Application, the API, the Logic Server, the AI/ML Service, and the Data Storage Components. Figure 4.15 shows the component diagram for our system

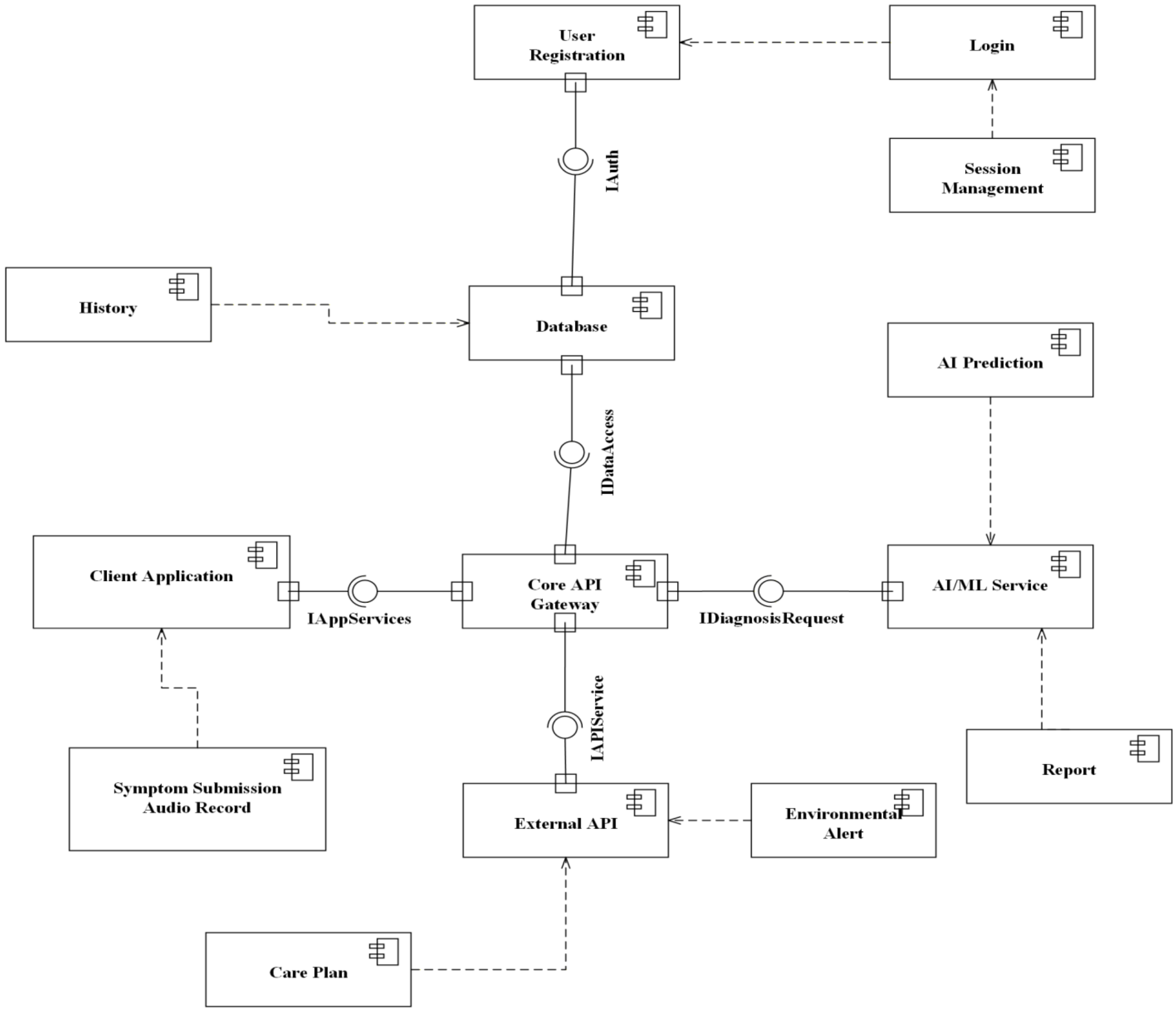

Figure 4.15 Component Diagram for BreatheWell System

4.4.4 Physical View

The Physical View of the BreatheWell system illustrates how the software components are deployed across various hardware nodes and cloud environments. It represents the system's runtime configuration, highlighting the physical distribution of software artifacts, their hosting infrastructure, and inter-node communication.

4.4.4.1 Deployment Diagram

1. The Deployment Diagram illustrates the run-time environment by detailing the nodes (hardware/virtual servers) and the artifacts (software components) deployed within them, connected by their communication pathways. Figure 4.16 shows the dyployment diagram of our system.

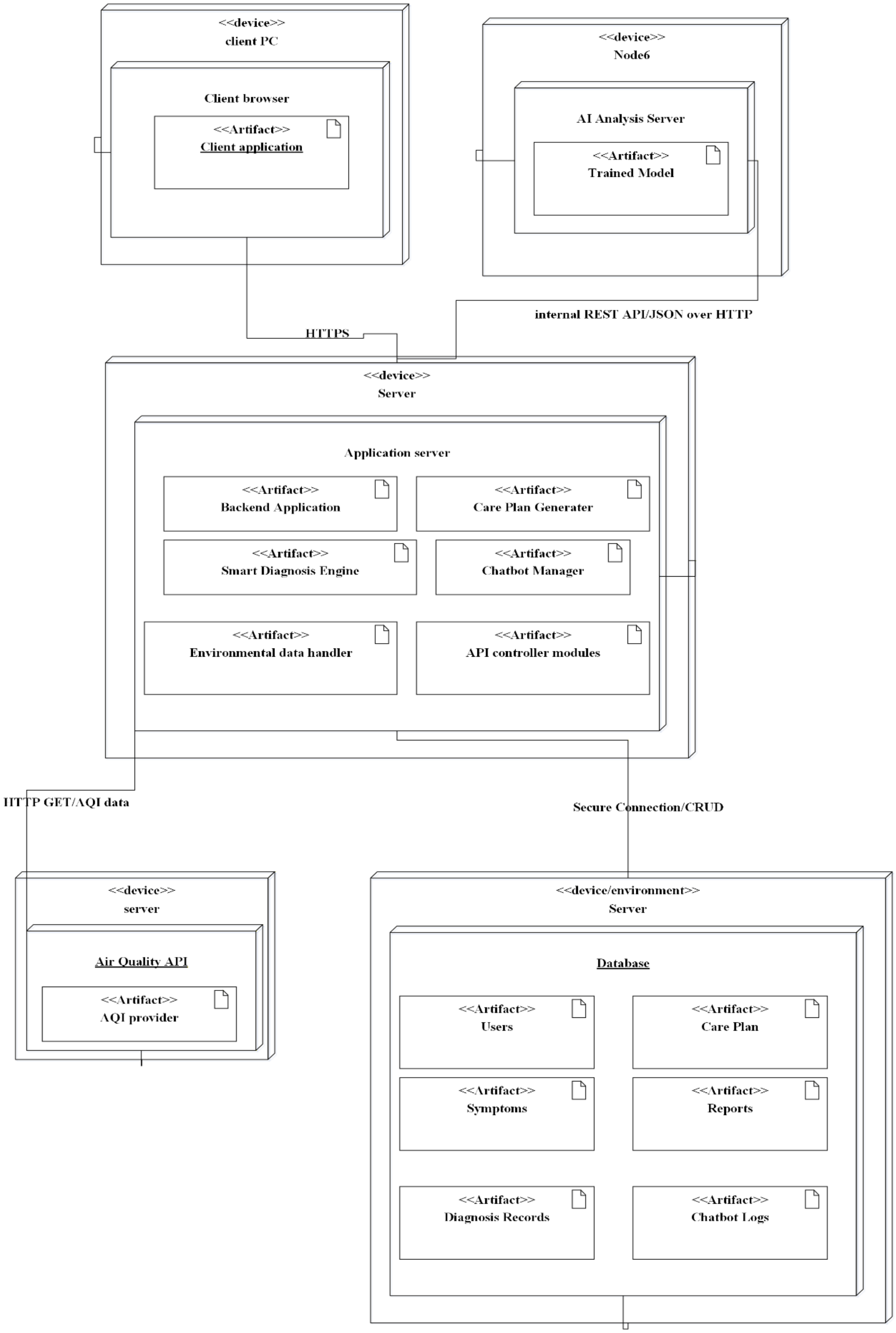

Figure 4.16 Deployment Diagram

4.4.5 Scenario View (+1 View)

This view, often represented by Use Case Diagrams and Sequence Diagrams, describes the interactions between the system components to fulfill a requirement. It validates the architectural design against the requirements.

4.5 App Design

Pending….

4.6 Summary

Chapter 4 provided the comprehensive System Modeling and Design for the BreatheWell application. The Three-Tier Architecture was established to ensure scalability and maintainability, separating the Presentation, Logic, and Data tiers. The importance of an intuitive Interface Design was highlighted, focusing on clarity and accessibility. Finally, the 4+1 View Model was used to structure the technical documentation, providing the necessary Logical, Process, Development, and Physical Views to guide the implementation phase effectively. The next chapter will focus on the detailed implementation and coding of these designed components.

References

[1] W. W. M. K. H. Labaki,, "Chronic respiratory diseases: a global view," vol. 8.6, pp. 531-533, 2020.

[2] H. A. E. Holland,, "How to adapt the pulmonary rehabilitation programme to patients with chronic respiratory disease other than COPD," *European Respiratory,* vol. 22., 2013.

[3] (. Ada Health GmbH, "Ada Health: About Us," [Online]. Available: https://about.ada.com/. [Accessed 07 December 2025].

[4] . I. AsthmaMD,, "AsthmaMD: About," 2010. [Online]. Available: https://www.asthmamd.org/about/. [Accessed 07 December 2025].

[5] U. News, "UQ startup acquired by Pfizer.," The University of Queensland, 27 September 2022. [Online]. Available: https://news.uq.edu.au/2022-09-27-uq-startup-acquired-pfizer. [Accessed 07 December 2025].

[6] Hyfe Inc. (n.d.)., "Hyfe: Detect & Quantify Cough," [Online]. Available: https://www.hyfe.com/. [Accessed 07 December 2025].