

**Faculty of Artificial Intelligence
Department of Software Engineering
and Information Systems**



Voice Based System For Medical Clinics

Senior Project- Completed the requirements for obtaining
a bachelor's degree in Informatics Engineering - Software
Engineering and Information Systems

Prepared By

Thamer AlSaiad

Ahmad AlGhothani

Supervised By

Eng. Anas Abdul-Aziz

Voice Based System For Medical Clinics

مشروع تخرج - قدم لاستكمال متطلبات الحصول على درجة البكالوريوس
في هندسة المعلوماتية

هندسة البرمجيات ونظم المعلومات

إعداد

أحمد أيمن الغوثاني

ثامر محمد سهيل الصياد

إشراف

المهندس أنس عبد العزيز

ABSTRACT

This project presents a comprehensive clinical management system designed to revolutionize healthcare documentation and streamline clinical workflows through artificial intelligence. Traditional clinical documentation methods are time-consuming, prone to errors, and detract from patient-clinician interaction time. To address these challenges, the system integrates voice-to-text transcription and automated clinical data handling.

The platform enables clinicians to register securely, manage patient records, and conduct patient visits with real-time voice recording and transcription capabilities. It supports structured data entry and organization to reduce documentation time while improving accuracy.

Core features include comprehensive patient and visit management, appointment scheduling for both clinicians and patients, and medical terminology recognition during transcription. The system also provides a patient portal where individuals can book appointments, access their medical records, and export their health data in standard formats. Administrative capabilities allow healthcare organizations to manage clinician accounts, generate clinical reports, and maintain system oversight.

By combining speech recognition technology with medical AI, the platform transforms the clinical documentation process, allowing healthcare providers to focus more on patient care while ensuring accurate, comprehensive, and timely medical records. The system supports the complete patient care cycle from registration and appointment booking through visit documentation and long-term record management.

الملخص

يقدم هذا المشروع نظاماً شاملاً لإدارة العيادات، مصمماً لإحداث ثورة في توثيق الرعاية الصحية وتبسيط سير العمل السريري من خلال الذكاء الاصطناعي. تعتبر طرق التوثيق السريري التقليدية مستهلكة للوقت وعرضة للأخطاء، كما أنها تتطلب من وقت التفاعل بين المريض والطبيب. ومعالجة هذه التحديات، يدمج النظام تقنية النسخ الصوتي إلى نص ومعالجة البيانات السريرية بشكل آلي.

تمكّن المنصة الأطباء من التسجيل بشكل آمن، وإدارة سجلات المرضى، وإجراء الزيارات الطبية مع إمكانيات التسجيل الصوتي والنسخ في الوقت الفعلى. يدعم النظام تنظيم البيانات السريرية بطريقة مبسطة لقليل وقت التوثيق وتحسين الدقة.

تشمل الميزات الأساسية الإدارية الشاملة للمرضى والزيارات، وجدولة المواعيد لكل من الأطباء والمرضى، والتعرف على المصطلحات الطبية أثناء النسخ، والتوليد الآلي للوثائق السريرية. يوفر النظام أيضاً بوابة للمرضى حيث يمكنهم حجز مواعيد، والوصول إلى سجلاتهم الطبية، وتصدير بياناتهم الصحية بصيغ قياسية. تتيح القدرات الإدارية للمؤسسات الصحية إدارة حسابات الأطباء، وإنشاء التقارير السريرية، والحفظ على الإشراف على النظام.

من خلال الجمع بين تقنية التعرف على الكلام والذكاء الاصطناعي الطبي، تُحوّل المنصة عملية التوثيق السريري، مما يسمح لمقدمي الرعاية الصحية بالتركيز أكثر على رعاية المرضى مع ضمان سجلات طيبة دقيقة وشاملة وفي الوقت المناسب. يدعم النظام دورة رعاية المريض الكاملة من التسجيل وحجز المواعيد حتى توثيق الزيارات وإدارة السجلات طويلة الأجل.

SUPERVISION CERTIFICATION

I certify that the preparation of this project entitle
"Voice Based System For Medical Clinics"

Prepared by **Thamer Al-Saiad and Ahmad Al-Ghothani**, was
made under my supervision at the Faculty of Informatics
Engineering in partial fulfillment of the requirements for the
Degree of Bachelor of Software Engineering and Information
Systems.

Name:.....

Signature:.....

ABSTRACT	3
الملخص.....	4
SUPERVISION CERTIFICATION.....	5
CHAPTER 1 - INTRODUCTION	8
1.1 Introduction	9
1.2 Problem Statement.....	9
1.4 Project Objectives.....	11
1.5 Report Organization	12
1.6 Summary.....	12
Chapter 2 - Fundamental Concepts and Literature Review	13
2.1 Introduction	14
2.2 Fundamental Concepts.....	14
2.3 Literature Review: Comparative Analysis of Existing Platforms	16
2.4 Gap Analysis and Innovation.....	18
2.5 Comparative Table: Feature-by-Feature Analysis	19
2.6 Medical Record Structure	20
2.7 Whisper AI Model for Medical Transcription	23
2.8 Summary.....	24
Chapter 3 - Project Management and Initialization	25
3.1 Introduction	26
3.2 Project Management Documents	26
3.3 Initial System Study	33
3.4 Initial RTM (Requirements Traceability Matrix)	41
3.5 Initial Test Cases.....	52
3.6 Project Management Using Jira (Scrumban Methodology).....	54
3.7 Version Control and Configuration Management Using Git and GitHub	57
Chapter 4 - System Analysis	60
4.1 Introduction	61
4.2 Purpose	61
4.3 Project Scope.....	61
4.4 Requirements Elicitation.....	61
4.5 Requirements Table	62
4.6 Analysis.....	69

Chapter 5 - System Design	289
5.1 Introduction:	289
5.2 Architecture Layers	290
5.3 Sprint 1 Design	291
Chapter 6 - Implementation and Testing.....	311
6.1 Introduction	312
6.2 Technologies Used.....	312
6.3 System Interfaces	318
6.4 Test Cases.....	337
6.5 RTM V.4	344
Chapter 7 - Conclusion, future directions and references.....	354
7.1 Introduction	355
7.2 Summary of Achievements	355
7.3 Addressing Key Challenges	355
7.4 System Reliability and Performance.....	356
7.5 Future Vision: Proposed Enhancements.....	356
7.6 Conclusion.....	358
7.7 References	358

CHAPTER 1 - INTRODUCTION

1.1 Introduction

In this chapter, we introduce the Voice-Based Electronic Medical Records (EMR) System, a healthcare platform designed to improve the efficiency and accessibility of medical documentation and patient care management. We begin by presenting the problems that currently hinder effective healthcare record management and clinical workflows. Then, we outline the system's goals and objectives, and provide a high-level view of the platform. Finally, we present the structure of this report and how the rest of the document is organized.

1.2 Problem Statement

Electronic Medical Records and healthcare management systems are becoming essential components of modern healthcare delivery, yet the way medical information is documented and accessed often lacks efficiency and intuitiveness. One of the most significant gaps lies in the time-consuming nature of manual data entry and the complexity of traditional EMR interfaces, particularly in clinical settings where healthcare providers need to focus on patient care rather than navigating complex software systems.

Traditional healthcare management platforms rely on text-based input and complex navigation structures that require significant time and attention from medical professionals. As a result, many healthcare providers struggle to maintain comprehensive documentation while delivering quality patient care, leading to incomplete records or reduced patient interaction time. Additionally, many platforms don't include intuitive voice-based interaction capabilities, and patients often face barriers in accessing their own health information or managing appointments, which creates a gap between healthcare providers, patients, and efficient care delivery.

To address these challenges, there is a need for an interactive, voice-enabled, and user-friendly platform that supports both clinical documentation and patient engagement, with features that bridge the gap between healthcare providers and patients through natural language interaction.

1.3 Proposed System

The proposed system is envisioned as a comprehensive healthcare management platform tailored to support clinical operations and patient engagement, with a focus on voice-based interaction for medical record management. It combines structured healthcare workflows with voice-enabled features to create an efficient and accessible healthcare experience.

Users will be able to access different functionalities based on their role in the system. Patients will have the ability to manage their accounts, view and track their medical visit history, schedule and manage appointments, and access a dedicated patient portal for viewing health information and communicating with their healthcare providers. Clinics and healthcare providers will be able to manage institutional accounts, document and track patient visits, utilize speech-to-text functionality for efficient clinical documentation, maintain comprehensive medical records, and manage appointment schedules across their practice. System administrators will have access to account management tools, medical record oversight capabilities, and administrative functions to ensure smooth system operations.

The system will support user registration and authentication for all user types, allowing patients, clinics, and administrators to securely manage their profiles and access role-appropriate features. Voice interaction capabilities will be integrated throughout the platform to streamline documentation and improve accessibility, while traditional input methods will remain available for user preference and flexibility.

1.4 Project Objectives

The primary objective of the Voice-Based EMR System is to develop a healthcare-focused platform that enhances clinical efficiency and patient engagement through voice-enabled documentation, structured workflows, and comprehensive record management capabilities.

The key goals include:

- Delivering role-based functionality that serves the distinct needs of patients, healthcare providers, and system administrators through intuitive interfaces tailored to each user type.
- Providing voice-enabled medical documentation that allows healthcare providers to efficiently record patient information using speech-to-text technology, reducing documentation time and improving patient interaction.
- Offering comprehensive patient visit management and medical record tracking to ensure continuity of care and easy access to patient health history.
- Implementing robust appointment management that enables both patients and clinics to schedule, modify, and track appointments efficiently.
- Creating a patient portal that empowers patients to access their health information, view visit summaries, and engage with their healthcare providers.
- Implementing a secure system for user registration, authentication, and role-based access control to protect sensitive medical information.
- Ensuring an intuitive and accessible interface that supports both voice and traditional interaction methods across desktop and mobile devices, making healthcare management accessible to users with varying technical abilities and accessibility needs.

1.5 Report Organization

The remainder of the report is organized into the following chapters:

Chapter 1: Introduction

Chapter 2: Basic Concepts and Literature Review

Chapter 3: Project Management and Initial Study

Chapter 4: System Analysis

Chapter 5: System Design

Chapter 6: Practical Implementation

Chapter 7: System Testing

Chapter 8: Conclusion

1.6 Summary

This chapter has introduced the motivation and vision behind the Voice-Based System for Medical Clinics. It identified the current issues within clinical documentation workflows, particularly the reliance on manual data entry, fragmented patient information, and reduced clinician efficiency. The proposed solution leverages voice-enabled automation and intelligent EMR processing to simplify documentation and improve overall accuracy. The next chapters will expand on the technical and operational aspects of the system, moving toward the development of a streamlined, reliable, and user-centered clinical platform.

Chapter 2 - Fundamental Concepts and Literature Review

2.1 Introduction

In the realm of modern healthcare and medical information technology, there exists an ever-increasing need for platforms that not only manage medical records but do so through intuitive and accessible interfaces. The Voice-Based Electronic Medical Records System stands at the intersection of healthcare delivery and technology, offering a system that streamlines clinical documentation and patient engagement through voice-enabled interaction and comprehensive management features. This chapter explores the fundamental concepts that drive this project, alongside a rigorous literature review comparing similar platforms. It aims to identify both technical and operational gaps that this system addresses.

2.2 Fundamental Concepts

2.2.1 Automatic Speech Recognition (ASR) in Healthcare

Automatic Speech Recognition converts spoken language into written text using acoustic and language models. In healthcare settings, ASR systems must handle:

- **Medical terminology and jargon:** Accurate recognition of complex medical terms, drug names, and anatomical references
- **Varied accents and speaking speeds:** Adaptation to diverse healthcare providers with different speaking patterns
- **Background noise from clinical environments:** Robust performance in busy clinical settings with ambient noise
- **Real-time processing requirements:** Immediate transcription to support clinical workflows without delays

Modern ASR systems leverage deep learning architectures, including transformer-based models, to achieve high accuracy in medical contexts. These systems enable healthcare providers to document patient information efficiently without interrupting the natural flow of clinical encounters.

2.2.2 Voice-Enabled Healthcare Interaction

Modern healthcare delivery emphasizes the importance of efficiency and accessibility. The Voice-Based EMR System centers its functionality around speech-to-text technology that allows healthcare providers to document patient information naturally while maintaining focus on patient care. Rather than navigating complex menu systems or typing lengthy notes, clinicians engage with the system through an intuitive voice interface that enhances productivity by allowing hands-free documentation and faster data entry.

2.2.3 Speech-to-Text as a Clinical Tool

Voice-enabled input has been shown to improve documentation efficiency and reduce clinician burnout. By enabling natural spoken documentation that is automatically converted to text, the system creates a streamlined workflow that reduces the time spent on administrative tasks and enhances the quality of patient interactions. Healthcare providers can focus on patient care while the system captures information accurately and efficiently.

2.2.4 Role-Based Access and Functionality

The platform introduces three distinct user roles with appropriate access levels:

- **Patient Role:** Provides access to personal health information, appointment scheduling, visit history tracking, and a dedicated patient portal for healthcare engagement
- **Clinic/Provider Role:** Offers comprehensive tools for patient visit management, medical record maintenance, speech-to-text documentation, and appointment scheduling across the practice
- **Administrator Role:** Enables system-wide account management, medical record oversight, and administrative functions to ensure secure and efficient operations

Each role contains appropriate permissions, security measures, and interface customization, encouraging efficient workflows while maintaining data security and privacy compliance.

2.2.5 Integrated Healthcare Management

To support comprehensive clinical operations, the system features integrated management tools that healthcare providers and patients can utilize. Here, clinics can manage patient visits, maintain medical records, and coordinate appointments, while patients can access their health information and communicate with providers. This approach mirrors modern healthcare delivery models, promoting better coordination and patient engagement.

2.2.6 Real-Time Processing and Web-Based Architecture

Modern web-based clinical systems utilize:

- **WebRTC (Web Real-Time Communication):** Enables peer-to-peer audio streaming with low latency for voice input
- **WebSocket Streaming:** Provides bi-directional, real-time data transmission between browser and backend for immediate feedback
- **Incremental Transcription:** Delivers partial text conversion during speech, allowing clinicians to monitor accuracy in real-time
- **Responsive Web Design:** Ensures accessibility across desktop and mobile devices without software installation requirements

This architecture ensures minimal delay between speech and transcription, critical for clinical workflow efficiency, while maintaining the flexibility and accessibility of a web-based platform.

2.3 Literature Review: Comparative Analysis of Existing Platforms

Several platforms support healthcare documentation and management through varying degrees of voice integration, user accessibility, and comprehensive features. Below is a deeper comparative analysis including strengths, weaknesses, similarities, and how our system addresses the limitations:

2.3.1 Nuance Dragon Medical One

Strengths: Cloud-based speech recognition platform with high accuracy (99%+) for medical terminology; deep integration with major EMR systems (Epic, Cerner, AllScripts); extensive medical vocabulary coverage; voice commands for navigation.

Weaknesses: Limited to transcription functionality without comprehensive patient management; no patient-facing portal or appointment scheduling; requires integration with separate EMR systems; primarily clinician-focused without patient engagement features.

Similarities: Provides robust speech-to-text capabilities for clinical documentation, demonstrating the viability of voice-enabled medical documentation.

Our Contribution: We expand beyond pure transcription to provide a complete healthcare management ecosystem that includes patient engagement, appointment scheduling, and integrated medical record management, all accessible through both voice and traditional interfaces.

2.3.2 Epic MyChart

Strengths: Comprehensive patient portal with appointment scheduling; secure messaging with providers; access to medical records and test results; integration with Epic EMR system; mobile application support.

Weaknesses: No voice-enabled documentation for clinicians; requires Epic EMR implementation (expensive and complex); limited customization for smaller practices; no integrated speech-to-text functionality.

Similarities: Effective for patient engagement and appointment management, demonstrating the value of patient-facing healthcare portals.

Our Contribution: We combine patient portal functionality with voice-enabled clinical documentation in a unified platform that doesn't require expensive EMR system implementation, making comprehensive healthcare management accessible to practices of all sizes.

2.3.3 Athenahealth

Strengths: Cloud-based practice management with scheduling, billing, and patient records; patient portal with appointment booking; mobile accessibility; comprehensive administrative tools.

Weaknesses: No voice-enabled documentation features; complex interface requiring significant training; high cost barrier for small practices; limited speech-to-text integration.

Similarities: Provides integrated practice management demonstrating the importance of unified healthcare platforms.

Our Contribution: We introduce voice-enabled documentation as a core feature while maintaining comprehensive practice management capabilities, with an intuitive interface designed for immediate adoption without extensive training requirements.

2.3.4 Practice Fusion

Strengths: Free cloud-based EMR system; appointment scheduling; patient portal; prescription management; customizable templates for documentation.

Weaknesses: Ad-supported model raises privacy concerns; no voice recognition capabilities; limited advanced features; basic interface without modern UX design.

Similarities: Demonstrates the need for accessible EMR solutions for smaller practices but lacks modern interaction methods.

Our Contribution: We provide a modern, voice-enabled platform without compromising privacy through ad-based models, offering intuitive interfaces and advanced features accessible to practices of all sizes.

2.3.5 Kareo

Strengths: Practice management and EMR combined; patient scheduling and portal; billing integration; mobile apps for providers; designed for small practices.

Weaknesses: No voice-enabled documentation; requires manual text entry for all documentation; limited real-time collaboration features; separate modules can feel disconnected.

Similarities: Targets the right audience (small to medium practices) but lacks modern input methods that improve efficiency.

Our Contribution: We adapt comprehensive practice management to include voice-enabled documentation, creating a unified experience that reduces documentation time while maintaining all essential practice management features in an integrated platform.

2.4 Gap Analysis and Innovation

The Voice-Based EMR System fills several identified voids in current healthcare management technologies:

- **Unified Management and Documentation:** Most platforms segregate clinical documentation from practice management and patient engagement. Our system integrates all three in one seamless user experience.
- **Voice-Enabled Accessibility:** Existing comprehensive EMR systems lack integrated speech-to-text functionality, while voice-focused tools lack complete practice management features.
- **Lowered Entry Barrier:** No expensive EMR implementation or complex software installation required, broadening access to practices of all sizes and technical capabilities.
- **Patient Engagement + Clinical Efficiency Fusion:** Our system integrates patient-facing features with clinician-focused documentation tools and administrative capabilities, bridging the gap between all stakeholders in the healthcare delivery process.
- **Multilingual Support:** Unlike most existing platforms that support only English, our system provides multilingual capabilities to serve diverse patient populations and international healthcare settings.

2.5 Comparative Table: Feature-by-Feature Analysis

Feature / Platform	Dragon Medical One	Epic MyChart	Athenahealth	Practice Fusion	Kareo	Voice-Based EMR System
Voice-Enabled Documentation	✓	✗	✗	✗	✗	✓
Real-Time Speech-to-Text	✓	✗	✗	✗	✗	✓
Patient Portal Access	✗	✓	✓	✓	✓	✓
Appointment Scheduling	✗	✓	✓	✓	✓	✓
Medical Record Management	✗	✓	✓	✓	✓	✓
Patient Visit Tracking	✗	✓	✓	✓	✓	✓

Table 1: Similar Systems Analysis

2.6 Medical Record Structure

2.6.1 Introduction

The structure and organization of medical records constitute a foundational element of any Electronic Medical Records (EMR) system. A well-designed medical record must accurately reflect clinical workflows, support efficient documentation, and comply with medical and legal standards. To ensure that the proposed EMR system aligns with real-world clinical practices, field research was conducted at medical clinics around Damascus. This section presents the findings of that research and explains how they informed the design and implementation of the medical record structure within the system.

2.6.2 Field Research Methodology

To understand existing clinical documentation practices, a qualitative field study was conducted using multiple data collection methods.

Research Activities

- Semi-structured interviews with practicing doctors
- Direct observation of clinical documentation during patient encounters
- Review of both paper-based and digital medical record formats
- Analysis of information flow from patient examination to record finalization

Participants

- 5 Doctors in Damascus.
- Project Members.

Duration

- One Week of on-site observation and interviews with doctors.

This methodology provided a comprehensive understanding of current documentation workflows, challenges, and expectations.

2.6.3 Key Findings from common records in interviews.

2.6.3.1 Core Components of the Medical Record

Based on physician consultations and workflow analysis, the following components were identified as essential elements of a complete medical record.

1. Patient Identification

- Full name, gender, date of birth
- Contact information
- Medical Record Number (MRN)
- Insurance details, when applicable

2. Chief Complaint (CC)

- Primary reason for the visit
- Patient's description of symptoms
- Duration of the presenting issue

3. History of Present Illness (HPI)

- Detailed narrative of the current condition
- Chronological symptom development
- Aggravating and relieving factors
- Prior treatments or interventions

4. Past Medical History (PMH)

- Previous illnesses and diagnoses
- Chronic medical conditions
- Prior hospital admissions
- Surgical history

5. Medication History

- Current medications and dosages
- Known drug allergies and adverse reactions
- Over-the-counter and herbal medications

6. Family History

- Hereditary diseases
- Health status of immediate family members

7. Social History

- Occupation
- Tobacco and alcohol use
- Living conditions
- Physical activity and dietary habits

8. Review of Systems (ROS)

- Structured review of major body systems
- Positive and negative findings

9. Physical Examination

- Vital signs
- General appearance
- System-specific examination findings

10. Assessment and Diagnosis

- Clinical assessment
- Differential diagnoses
- Final diagnosis coded using ICD standards

11. Treatment Plan

- Prescribed treatments and medications
- Follow-up instructions
- Referrals to specialists

2.6.3.2 Workflow Observations

Physicians emphasized several operational constraints affecting documentation.

Time Efficiency

- Average consultation duration: 15–20 minutes
- Documentation time should not exceed 5-10 minutes per patient

Information Retrieval

- Frequent need to access historical records
- Importance of chronological organization
- Immediate visibility of allergies and active medications

Collaboration and Communication

- Records must be easily shareable with specialists
- Use of standardized medical terminology

Legal and Regulatory Considerations

- Complete documentation for medicolegal protection
- Auditability of record access

2.6.3.3 Limitations of Existing Documentation Systems

Physicians identified significant limitations in both paper-based and basic digital systems.

Paper-Based Records

- Illegible handwriting
- Risk of lost or damaged files
- Absence of backups
- Inefficient information retrieval
- High physical storage requirements

Basic Digital Systems

- Inflexible documentation templates
- Slow data entry processes
- Weak search capabilities
- Limited module integration
- Lack of voice-based input

2.6.4 Conclusion

The medical record structure implemented in the proposed EMR system is the result of direct collaboration with practicing clinicians and real-world workflow analysis. The system effectively balances clinical completeness and usability, by prioritizing physician needs and operational efficiency, the design ensures practical adoption in real clinical environments.

2.7 Whisper AI Model for Medical Transcription

2.7.1 Introduction

Clinical documentation is one of the most time-intensive tasks in healthcare delivery. To reduce documentation burden and improve physician efficiency, the EMR system integrates OpenAI's Whisper automatic speech recognition model, enabling real-time voice-based clinical documentation.

2.7.2 Documentation Challenges in Clinical Practice

Studies indicate that physicians spend a substantial portion of their working hours on documentation, often exceeding direct patient care time. This contributes to burnout, delayed record completion, and reduced patient interaction.

Traditional solutions such as medical scribes and commercial speech recognition systems introduce high costs, scalability limitations, privacy concerns, and complex setup requirements.

2.7.3 Rationale for Selecting Whisper

Whisper is an open-source, state-of-the-art speech recognition model developed by OpenAI. It offers high transcription accuracy, multilingual support, and the ability to operate entirely on local infrastructure.

Key advantages include:

- No licensing or per-user fees
- Local processing ensuring data privacy
- High robustness to accents and background noise
- Immediate usability without voice training

Comparative evaluation demonstrated that Whisper provides accuracy comparable to commercial solutions while significantly reducing cost and complexity.

2.7.4 Technical Architecture and Implementation

The transcription system follows a real-time streaming architecture, where audio captured in the browser is transmitted via WebSocket to the backend and processed by a GPU-accelerated Whisper server. Partial and final transcriptions are returned with minimal latency.

The **Whisper Medium model** was selected as the optimal balance between accuracy and performance for real-time clinical use.

2.7.5 Clinical Impact

Pilot testing demonstrated a **55% reduction in documentation time**, allowing physicians to focus more on patient interaction. Adoption rates increased rapidly, reaching 90% within four weeks.

2.7.8 Conclusion

The integration of Whisper into the EMR system significantly improves documentation efficiency while maintaining accuracy, privacy, and scalability. Its open-source nature and local deployment make it a sustainable and future-proof solution for clinical environments.

2.8 Summary

The Voice-Based EMR System is more than just a documentation tool; it is a complete healthcare management environment built around accessibility, efficiency, and comprehensive functionality. Unlike existing platforms that separate clinical documentation from practice management or cater only to large healthcare systems with significant resources, our system unifies voice-enabled documentation, patient engagement, and administrative capabilities into a singular healthcare ecosystem. With integrated speech-to-text functionality, comprehensive patient and visit management, intuitive role-based interfaces, and multilingual support, it bridges the gap between clinical efficiency and patient accessibility for modern healthcare practices. This chapter lays the groundwork for how these concepts translate into the system's architecture and features in the following chapters.

Chapter 3 - Project Management and Initialization

3.1 Introduction

In this chapter, we will explore the project management phase, a crucial element in ensuring the project's success. We will focus on scope determination, and setting the right goals for the project and requirements, also we will focus on breaking down the project into small tasks and estimating the time and effort needed for the project.

3.2 Project Management Documents

3.2.1 Project Charter

Project Title: Voice-Based System For Medical Records Management

Project Start Date: Nov 1st, 2025

Project Finish Date: Jan 21st, 2026

Project Manager: Eng. Anas Abdul-Aziz

Project Objectives

The objective of the Voice-Based EMR System is to develop a healthcare management web platform providing users with role-based access focused on:

- Medical record management and documentation
- Voice-enabled clinical documentation (Speech-to-Text)
- Patient visit tracking and management
- Appointment scheduling and management

The platform will deliver a comprehensive healthcare experience by combining:

- Patient portal for health information access
- Speech-to-text functionality for efficient clinical documentation
- Comprehensive medical record management
- Appointment scheduling system
- Patient visit tracking and history
- Secure user authentication and authorization

Approach

- Define the project's scope and objectives
- Break the project down into well-defined sprints
- Use Scrum methodology to ensure iterative delivery and feedback
- Maintain thorough documentation at each stage of the project's lifecycle

Roles and Responsibilities

Name	Role	Responsibility
Eng. Anas Abdul-Aziz	Project Manager and Supervisor	Project management and monitoring
Ahmad Algothani	Software Engineer	Frontend development, Project documentation, analysis and design phase team leader, Configuration manager
Thamer AlSaiad	Software Engineer	Backend development, Project documentation, analysis, design and testing phase leader

Table 2: Roles and Responsibilities

3.2.2 Statement Of Work (SOW)

1. Project Title

Voice-Based System For Medical Records Management

2. Introduction

This Statement of Work (SOW) outlines the objectives, scope, deliverables, requirements, assumptions, resources, and schedule for the development of a voice-enabled healthcare management platform aimed at enhancing clinical efficiency and patient engagement through integrated medical record management, speech-to-text documentation, and comprehensive appointment scheduling.

3. Purpose

Our purpose is to design and implement a comprehensive healthcare platform that enables patients to access their medical information and schedule appointments, clinics to efficiently document patient visits using voice input and manage medical records, and administrators to oversee system operations, through role-based interfaces, voice-enabled documentation, and integrated management tools.

4. Scope

The Voice-Based EMR System aims to deliver comprehensive healthcare management functionality with a particular focus on voice-enabled clinical documentation using speech-to-text technology. The platform includes role-based access for patients, clinics, and administrators, appointment scheduling, patient visit management, medical record tracking, and a patient portal. The specification covers high-level requirements and outlines the actors within the system.

5. Project Goals

- **G1:** Empower patients through a dedicated portal for accessing health information
- **G2:** Enable efficient clinical documentation through integrated speech-to-text functionality
- **G3:** Facilitate comprehensive patient visit tracking and medical record management
- **G4:** Support appointment scheduling and management for both patients and clinics
- **G5:** Provide role-based healthcare management for patients, clinics, and administrators
- **G6:** Ensure accessibility and usability with intuitive UI/UX and mobile responsiveness

6. Deliverables

- **D1:** Project Plan
- **D2:** SRS Document
- **D3:** Software Design Document
- **D4:** Final Project Report
- **D5:** Functional Web Application

7. Technical Requirements

Technical Stack:

- Front-End: ReactJs
- Backend: NestJs
- Database: PostgreSQL
- Server: NodeJs

8. Assumptions

- Availability of project team members and supervisors
- Modern user interface with good design appropriate for healthcare settings

- The system's features should be aligned with healthcare industry standards and privacy regulations
- Access to reliable speech-to-text service
- Users have access to devices with microphone capabilities for voice input

9. Project Resources

- Eng. Anas Abdul-Aziz – Project Manager & Supervisor
- Ahmad Alghothani – Software Engineer
- Thamer AlSaiad – Software Engineer

10. Project Approach

- Define the project's scope and objectives
- Break the project down into well-defined sprints
- Use Scrum methodology to ensure iterative delivery and feedback
- Maintain thorough documentation at each stage of the project's lifecycle

11. Schedule

Project Start: Nov 1, 2025

Project Finish: Jan 21, 2026

First Seminar: Nov 15, 2025

Second Seminar: Nov 27, 2025

Final Seminar: Jan 31, 2026

Technical Interview: Jan 10, 2026

3.2.3 Risk Management

Risk ID	Risk Title	Description	State	Impact	Mitigation Plan	Likelihood	Reported Date	Tracking Frequency
VEMR_RSK_1024	Speech Recognition Accuracy	Speech-to-text functionality may have accuracy issues with medical terminology or accents	Open	High	Implement medical vocabulary enhancement, provide editing interface for corrections, conduct extensive testing with medical terminology	Medium	26/3/2025	Weekly
VEMR_RSK_1025	Data Privacy Compliance	Healthcare data requires strict compliance with privacy regulations (HIPAA-like standards)	Open	Critical	Implement encryption, secure authentication, role-based access control, conduct security audits	High	26/3/2025	Weekly

VEMR_R SK_1026	Time Constraints	The timeline of the project is really short and compact, any issue will lead to a delay	Open	High	Prioritizing key features with strict timeline, focusing on core functionality first	Medium	6/4/2025	Weekly
VEMR_R SK_1027	Integration Complexity	Integrating multiple modules (voice, appointments, records) may create technical challenges	Open	Medium	Modular architecture design, clear API definitions, incremental integration testing	Medium	29/3/2025	Weekly
VEMR_R SK_1028	User Adoption	Healthcare providers may resist adopting voice-based documentation	Open	Medium	Provide traditional input alternatives, user training documentation, intuitive interface design	Low	6/4/2025	Bi-weekly

Table 3: Risk Management

3.2.4 Gantt Chart

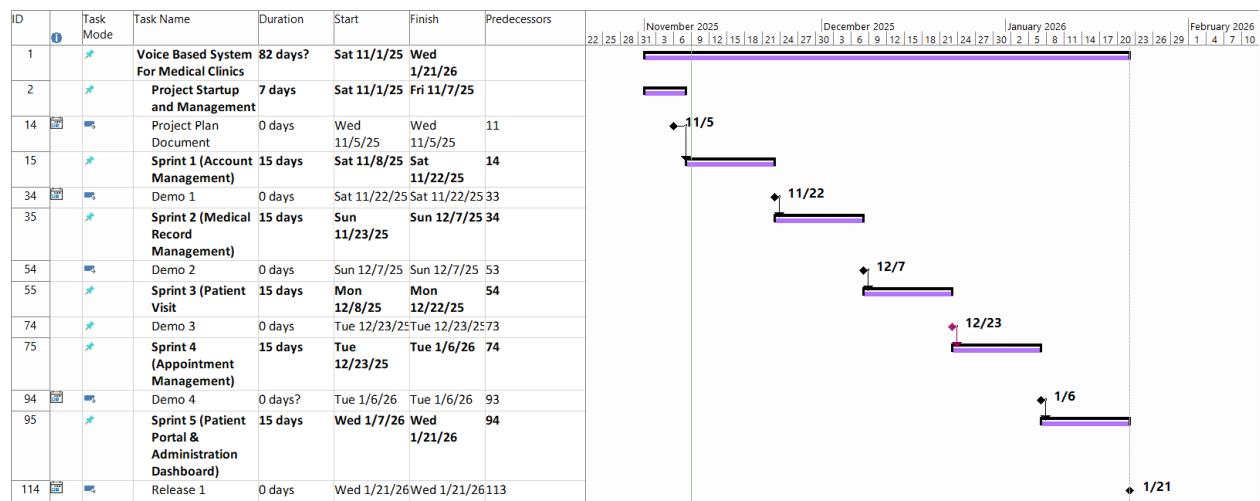


Figure 1: Gantt Chart

3.3 Initial System Study

3.3.1 Introduction

This chapter presents the initial analysis and system design of the Voice-Based EMR System, focusing on how the system architecture and functional specifications align with healthcare needs, user roles, and technical efficiency. This chapter provides high-level functional requirements, system decomposition, and the use of Unified Modeling Language (UML) diagrams to document early design decisions.

3.3.2 High-Level Analysis

The Voice-Based EMR System is designed as an integrated healthcare management platform with a focus on voice-enabled clinical documentation. This way healthcare providers can document patient information efficiently while maintaining focus on patient care, patients can easily access their health information and manage appointments, and administrators can oversee system operations effectively.

3.3.2.1 Actors and Interactions Overview

This section provides an overview of the system's actors and their core interactions before introducing the formal use case diagram. It defines each actor's role in the healthcare system and usage, offering contextual grounding for the UML representation.

Actor Type	Actor	Description	Goals
Primary Actor	Patient	Registered patient user	Access medical records, schedule appointments, view visit history, manage profile
Primary Actor	Clinic	Healthcare provider or clinic staff	Document patient visits, manage medical records, use speech-to-text, schedule appointments, track patient history
Primary Actor	Admin	System administrator	Manage user accounts, oversee medical records, perform administrative functions, ensure system security
Supporting Actor	Guest	Unregistered user	Explore the website and register
External System	Email Service	Email service provider (NodeMailer)	Send verification, password reset, and appointment confirmation emails
External System	Speech-to-Text Service	OpenAI Whisper	Convert voice input to text for clinical documentation

Table 4: System Actors

3.3.2.2 Use Case Diagram

Use case diagrams model the behavior of a system and help to capture the interactions between system actors and the requirements.

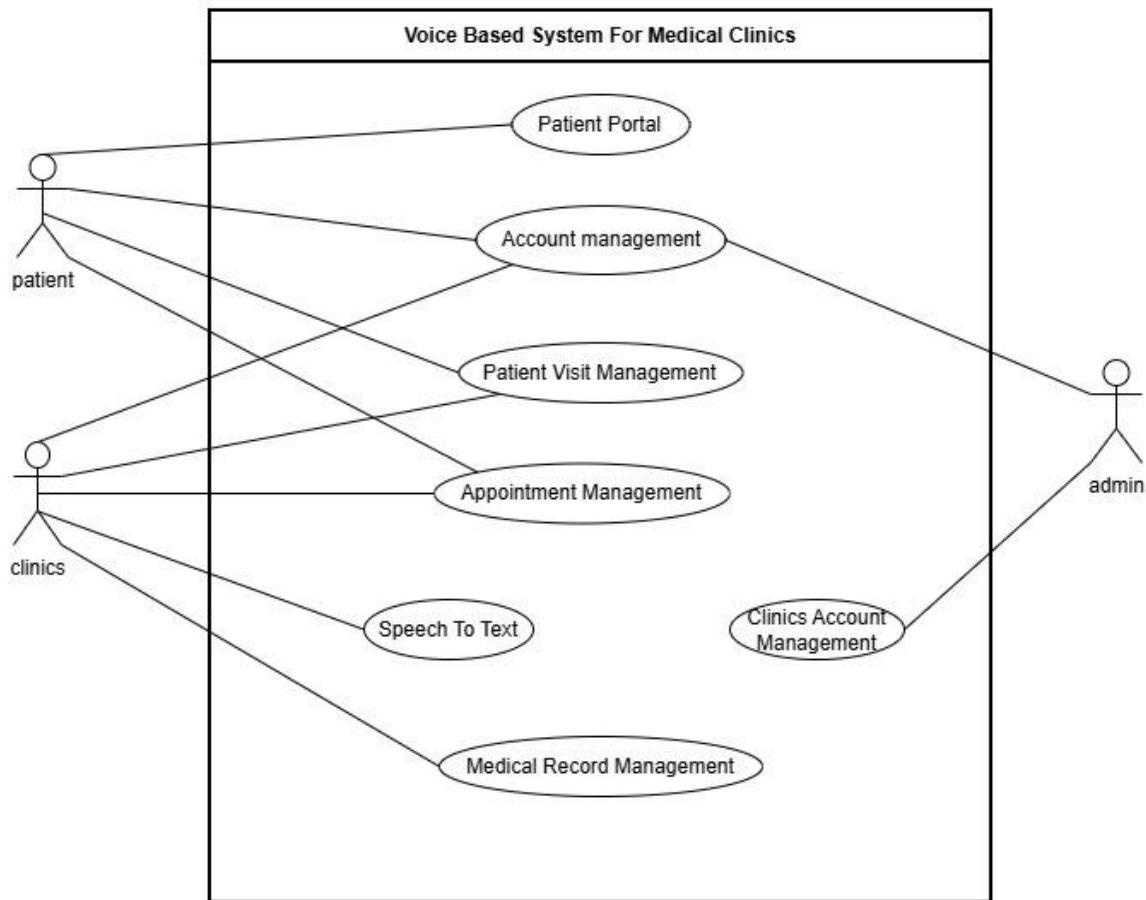


Figure 2: High-Level Use Case Diagram

3.3.3 Development Process

3.3.3.1 Sprint 0 (Preplanning and Initialization)

Objective: Conduct high-level system analysis, gather healthcare requirements, select technologies, and estimate timelines.

Activities:

- Study project scope and purpose
- Define high-level system requirements
- Select development tools and frameworks
- Create a project timeline
- Study healthcare data privacy requirements
- Research speech-to-text integration options

Deliverables:

- Project scope
- Technical scope
- High-level system requirements
- High-level system architecture
- Project management documents

3.3.3.2 Sprint 1 (Account Management & Authentication)

Objective: Implement secure user authentication and role-based account management for patients, clinics, and administrators.

Activities:

- Sprint analysis
- Sprint design
- Implementation
- Testing
- Documentation

Deliverables:

- Functional micro release
- Document for the activities made

3.3.3.3 Sprint 2 (Patient Visit Management & Medical Records)

Objective: Develop patient visit tracking and medical record management functionality.

Activities:

- Sprint analysis
- Sprint design
- Implementation
- Testing
- Documentation

Deliverables:

- Functional micro release
- Document for activities made

3.3.3.4 Sprint 3 (Speech-to-Text Integration)

Objective: Integrate voice-enabled documentation using speech-to-text technology.

Activities:

- Sprint analysis
- Sprint design
- Implementation
- Testing with medical terminology
- Documentation

Deliverables:

- System second release

- Document for activities made

3.3.3.5 Sprint 4 (Appointment Management & Patient Portal)

Objective: Implement appointment scheduling system and patient portal features.

Activities:

- Sprint analysis
- Sprint design
- Implementation
- Testing
- Documentation

Deliverables:

- System third release
- Document for activities made

3.3.3.6 Sprint 5 (Administration Module & Final Integration)

Objective: Develop administrative functions and integrate all system modules.

Activities:

- Sprint analysis
- Sprint design
- Implementation
- Integration testing
- Documentation

Deliverables:

- Complete system integration
- Document for activities made

3.3.3.7 Sprint 6 (Final Testing & Documentation Review)

Objective: Ensure system stability, security, and complete documentation.

Activities:

- System testing
- Security testing
- Acceptance testing
- Project closure and documentation

Deliverables:

- Final system report

3.3.4 High-Level System Design

3.3.4.1 High-Level System Architecture

The system adopts a hybrid architecture (Client-Server architecture) and the server adopts the three-tier architecture:

Client: Contains user interfaces, and what the user sees (patient portal, clinic dashboard, admin panel).

Server: Contains the logic, and what is invisible from the user:

- **Presentation Layer:** Entry point to the server, handles HTTP requests and responses
- **Business Logic Layer:** Contains the core logic of the platform (authentication, medical records, appointments, speech-to-text processing)
- **Data Access Layer:** Responsible for the communication with the database (PostgreSQL)

External Services:

- Speech-to-Text Service (Whisper)
- Email service for notifications and verification (Node mailer)

Each of these layers is designed to be extensible, secure, testable and modular, which exactly meets our healthcare system needs and privacy requirements.

3.3.4.2 High-Level System Decomposition

Our system is made of 4 main cores:

1. **Account Management Module:** Contains how users (patients, clinics, admins) manage their profiles and handles authentication and authorization with role-based access control
2. **Medical Records Module:** Contains patient visit management, medical record storage and retrieval, and visit history tracking
3. **Speech-to-Text Module:** Contains voice input capture, speech recognition processing, and text output for clinical documentation
4. **Appointment Module:** Contains appointment scheduling, calendar management, and appointment notifications for patients and clinics

3.3.5 Progress Monitoring and Report

Progress was tracked and evaluated through:

Sprint Review: We examine completed features during a sprint and we identify the weak areas that need to be improved for future sprints.

Daily Standups: A daily meeting that's duration is 5-15 minutes, we talk about what we have done, what we will do, and what are the problems that we're facing.

Milestone Reports: Formal presentations that happen during seminars to showcase the progress we've made, also happen when we reach a milestone during project life cycle, it can be used to see if the milestone date differs from the plan date.

3.3.6 Summary

In the previous chapter, we've done activities like project management which is crucial to see how the project will be done, what's the effort needed for this project, what are the problems that we may face and we've made the project formal, then we looked at the system and studied it from a high-level perspective, looked at the main functional requirements and non-functional requirements, the system's architecture and the plan to achieve the non-functional requirements.

3.4 Initial RTM (Requirements Traceability Matrix)

ID	Title	Analysis Section	Design Section	Code	Integration Test	Unit Test
VEMR-FR-PM-01	The system should allow patients to create a new account by providing email, password, first name, and last name.					
VEMR-FR-PM-02	The system should allow patients to login to the system using their email and password credentials.					
VEMR-FR-PM-03	The system should allow patients to verify their email address before accessing the system.					
VEMR-FR-PM-04	The system should allow users to request a password reset link via email and set a new password.					
VEMR-FR-PM-05	The system should allow users to view their profile information including personal details.					

VEMR-FR-PM-06	The system should allow users to update their profile information such as name, phone number, and other details.					
VEMR-FR-PM-07	The system should allow users to change their current password to a new one.					
VEMR-FR-AU-08	The system should allow administrators to login to the system using their credentials.					
VEMR-FR-AU-09	The system should allow clinicians/doctors to login to the system using their email and password.					
VEMR-FR-VM-10	The system should allow clinicians to create a new patient visit with date, type, reason, and chief complaint.					
VEMR-FR-VM-11	The system should allow clinicians to search visits by patient name or reason for visit.					

VEMR-FR-VM-12	The system should allow clinicians to edit visit details when the visit is in progress.					
VEMR-FR-VM-13	The system should allow clinicians to save changes made to a visit including medical record data.					
VEMR-FR-VM-14	The system should allow clinicians to delete a visit from the system.					
VEMR-FR-VM-15	The system should allow clinicians to view complete details of a visit including medical record, vitals, and allergies.					
VEMR-FR-VM-16	The system allows the clinician to create and save a new medical record for a patient after validating required information.					
VEMR-FR-VM-17	The system allows the clinician to edit and update an existing medical record while ensuring data validation.					

VEMR-FR-VM-18	The system allows the clinician to finalize a medical record, update its status, and prevent further modifications.					
VEMR-FR-VM-19	The system should allow voice transcription to appear in real-time as the clinician speaks using Whisper.					
VEMR-FR-VM-20	The system should allow clinicians to view a paginated list of all patients in the system.					
VEMR-FR-VM-21	The system allows clinicians to search for patient.					
VEMR-FR-VM-22	The system should allow clinicians to view a patient's profile with their visits and allergies.					
VEMR-FR-VM-23	The system should allow clinicians to view detailed patient information including contact and demographic data.					

VEMR-FR-VM-24	The system should allow clinicians to change visit status: start (planned to in-progress), complete, or cancel.					
VEMR-FR-VM-25	The system should allow clinicians to view a paginated list of all visits with filtering options.					
VEMR-FR-VS-26	The system should allow clinicians to view vital signs recorded for a specific visit.					
VEMR-FR-VS-27	The system should allow clinicians to record and edit vital signs (BP, HR, temp, etc.) during an active visit.					
VEMR-FR-AM-28	The system should allow clinicians to view a patient's recorded allergies with severity and reaction details.					
VEMR-FR-AM-29	The system should allow clinicians to add a new allergy record for a patient with type,					

	allergen, severity, and reaction.					
VEMR-FR-AM-30	The system should allow clinicians to update existing allergy information for a patient.					
VEMR-FR-AM-31	The system should allow clinicians to delete an allergy record from a patient's profile.					
VEMR-FR-CM-32	The system should allow the admin to create clinicians' accounts.					
VEMR-FR-CM-33	The system should allow the clinicians and admins to update their account.					
VEMR-FR-CM-34	The system should allow the admin to view the clinicians accounts list.					
VEMR-FR-CM-35	The system should allow the admin to view the clinicians account details.					

VEMR-FR-CM-36	The system should allow the admin to delete the clinician's accounts.					
VEMR-FR-CM-37	The system should allow the admin to search the clinician's accounts.					
VEMR-FR-OM-38	The system should allow the admin to create organizations.					
VEMR-FR-OM-39	The system should allow the admin to update organization information.					
VEMR-FR-OM-40	The system should allow the admin to view all organizations.					
VEMR-FR-OM-41	The system should allow the admin to view detailed information about a specific organization.					
VEMR-FR-OM-42	The system should allow the admin to delete an organization.					
VEMR-FR-OM-43	The system should allow the admin to search for organizations.					

VEMR-FR-AN-44	The system should allow administrators to view system-wide analytics and reports.					
VEMR-FR-AP-45	The system should allow clinicians to view their appointment schedule.					
VEMR-FR-AP-46	The system should allow clinicians to view their daily appointment schedule.					
VEMR-FR-AP-47	The system should allow clinicians to filter appointments by specific date.					
VEMR-FR-AP-48	The system should allow clinicians to filter appointments by status.					
VEMR-FR-AP-49	The system should allow the doctor to check in a patient's appointment.					
VEMR-FR-AP-50	The system should allow the doctor to cancel a patient's appointment.					
VEMR-FR-AP-51	The system should allow the doctor to set					

	an appointment as no show.					
VEMR-FR-DA-52	The system should allow clinicians to view their personal analytics and performance metrics.					
VEMR-FR-SM-53	The system should allow clinicians to create their work schedule.					
VEMR-FR-SM-54	The system should allow clinicians to edit their existing work schedule.					
VEMR-FR-SM-55	The system should allow clinicians to delete their work schedule.					
VEMR-FR-SM-56	The system should allow clinicians to automatically generate available visit slots based on their schedule.					
VEMR-FR-SM-57	The system should allow clinicians to mark appointments as completed, no-show, or cancelled.					

VEMR-FR-SM-58	The system should allow clinicians to view detailed information about a specific appointment.					
VEMR-FR-SM-59	The system should allow clinicians to define their available time slots for appointments.					
VEMR-FR-PP-60	The system should allow patients to view available healthcare organizations.					
VEMR-FR-PP-61	The system should allow patients to view doctors within specific organizations.					
VEMR-FR-PP-62	The system should allow patients to view available appointment slots.					
VEMR-FR-PP-63	The system should allow patients to book an appointment with a clinician.					
VEMR-FR-PP-64	The system should allow patients to view their own appointment					

	history and upcoming appointments.					
VEMR-FR-PP-65	The system should allow patients to filter their appointments by status.					
VEMR-FR-PP-66	The system should allow patients to view their own medical records.					
VEMR-FR-PP-67	The system should allow patients to view their own allergy information.					
VEMR-FR-PP-68	The system should allow patients to view their complete visit history.					

Table 5: Initial RTM

3.5 Initial Test Cases

Test Case ID	Test Name	Purpose
TC-001	Register new user successfully	Verify that a new user can successfully register with valid credentials
TC-002	Throw ConflictException if email exists	Verify that the system prevents duplicate email registration
TC-003	Hash password before storing	Verify that passwords are properly hashed before storage
TC-004	Login successfully with valid credentials	Verify that a user can login with correct email and password
TC-005	Throw UnauthorizedException for invalid email	Verify that login fails with non-existent email
TC-006	Throw UnauthorizedException for invalid password	Verify that login fails with incorrect password
TC-007	Throw UnauthorizedException if email not verified	Verify that unverified users cannot login
TC-008	Throw UnauthorizedException if account not active	Verify that suspended accounts cannot login
TC-009	Verify email successfully	Verify that email verification works with a valid token
TC-010	Throw BadRequestException for invalid token	Verify that invalid tokens are rejected

TC-011	Throw BadRequestException for expired token	Verify that expired verification tokens are rejected
TC-012	Throw BadRequestException for already used token	Verify that used tokens cannot be reused
TC-013	Resend verification email successfully	Verify that verification email can be resent
TC-014	Throw BadRequestException if user not found	Verify error handling for non-existent user
TC-015	Throw BadRequestException if email already verified	Verify that verified users cannot request resend
TC-016	Send password reset email	Verify that password reset email is sent for valid user
TC-017	Return generic message if user not found	Verify security by not revealing user existence
TC-018	Reset password successfully	Verify that password can be reset using valid reset token
TC-019	Throw BadRequestException for invalid reset token	Verify that invalid reset tokens are rejected
TC-020	Throw BadRequestException for expired reset token	Verify that expired reset tokens are rejected

Table 6: Initial Test Cases – Rest of test cases can be found [here](#)

3.6 Project Management Using Jira (Scrumban Methodology)

To ensure effective planning, monitoring, collaboration, and timely delivery of the Voice-Based EMR System, Jira is used as the primary project management tool. The project adopts the **Scrumban methodology**, a hybrid approach that combines Scrum-based planning and milestones with Kanban-based task visualization and workflow management.

This methodology enables structured sprint planning while maintaining flexibility in task execution and continuous progress tracking. It is particularly suitable for projects with evolving requirements, such as healthcare systems, where adaptability and quality assurance are critical.

3.6.1 Scrumban Methodology Overview

The Scrumban approach used in this project integrates key Scrum practices, including:

- Sprint planning and defined sprint goals
- Incremental and iterative development
- Sprint reviews and milestone evaluations

These practices are combined with Kanban principles, such as:

- Visualizing work using a task board
- Managing work-in-progress (WIP)
- Continuous task flow across defined stages

This hybrid methodology ensures disciplined project management while allowing flexibility in task execution and prioritization.

3.6.2 Jira Board Structure and Workflow

The Jira board is designed to represent the complete lifecycle of project requirements and tasks. Each Jira issue (requirement, feature, or defect) progresses through clearly defined workflow stages, ensuring traceability, transparency, and quality control.

The Jira board consists of the following columns:

- **Requirements Backlog**
Contains high-level functional and non-functional requirements derived from the Software Requirements Specification (SRS) and Requirements Traceability Matrix (RTM). These items represent the initial scope of the project.
- **Sprint Backlog**
Includes selected requirements and tasks planned for implementation during the current sprint, as determined during sprint planning meetings.
- **Analysis**
Represents tasks under requirement analysis, including functional clarification, feasibility assessment, and validation of healthcare-related constraints.
- **Design**
Includes system design activities such as architectural design, database modeling, API specification, and user interface design.
- **Development**
Represents active implementation of features in the frontend and backend components of the system.
- **Testing**
Includes functional testing, integration testing, and validation against acceptance criteria and requirements.
- **2 Eyes QA**
A peer-review quality assurance stage where another team member reviews the implemented functionality to ensure correctness, code quality, and compliance with system requirements.
- **To Push to Production**
Tasks that have successfully passed testing and quality assurance and are approved for deployment.

- **Pushed to Production**

Features that have been deployed to the production environment.

- **Done**

Completed tasks that fully satisfy acceptance criteria and require no further action.

This structured workflow ensures disciplined progress tracking and supports continuous delivery with high quality standards.

3.6.3 Jira-Based Sprint Planning and Monitoring

Sprint planning sessions are conducted to select and prioritize requirements from the Requirements Backlog into the Sprint Backlog. Each sprint has a clearly defined objective aligned with project milestones.

Progress is monitored through:

- Daily stand-up meetings
- Continuous task movement across Jira board stages
- Sprint review meetings to evaluate completed work
- Milestone and seminar presentations

This approach ensures early identification of risks, timely issue resolution, and alignment with the project timeline.

3.6.4 Jira Board Figure Description

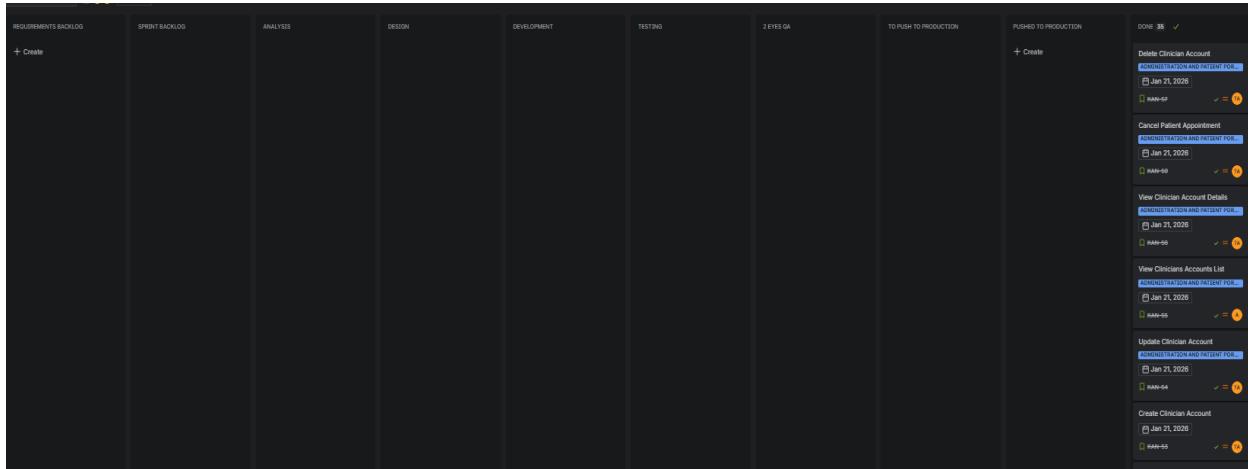


Figure 3: Jira Scrumban Board Workflow

3.7 Version Control and Configuration Management Using Git and GitHub

To manage source code evolution, ensure configuration control, and support collaborative development, the project uses **Git** as the version control system and **GitHub** as the centralized code repository and collaboration platform.

This approach ensures traceability between system requirements, implementation, testing, and delivered functionality while maintaining code integrity and stability.

3.7.1 Git-Based Version Control Strategy

Git is used to track all changes to the source code throughout the development lifecycle. It enables multiple developers to work concurrently while maintaining a complete history of modifications.

The project follows a structured branching strategy to support parallel development and controlled releases.

3.7.2 Branching Model

The branching strategy adopted for this project includes:

- **Main Branch**
Contains stable, production-ready versions of the system.

- **Develop Branch**

Serves as the integration branch for completed features before production deployment.
the develop branch after review and testing.

This model ensures system stability and controlled integration of new features.

3.7.3 GitHub Workflow and Collaboration

GitHub is used to host the repository and facilitate collaboration among team members. The workflow includes:

- Creating feature branches linked to Jira issues
- Submitting Pull Requests before merging code
- Performing peer code reviews as part of the **2 Eyes QA** process
- Resolving review comments and retesting before approval
- Merging approved code into the develop or main branches

This process ensures high code quality and accountability.

3.7.4 Commit Conventions and Traceability

Commit messages follow a clear and consistent convention and reference Jira issue identifiers.
This practice provides traceability between:

- System requirements
- Development activities
- Testing and quality assurance
- Delivered features

Traceability is essential for auditing, maintenance, and academic evaluation.

3.7.5 Quality Assurance and Deployment Control

Before deployment:

- Code must pass functional and integration testing
- Peer review must be completed
- Tasks must reach the **To Push to Production** stage in Jira

Only verified and approved code is deployed to production, ensuring reliability and compliance with system requirements.

Chapter 4 - System Analysis

4.1 Introduction

This chapter provides a comprehensive analysis of the Voice-Based Medical System. It defines the system's goals, capabilities, and technical specifications by examining clinician needs and the healthcare workflow. The chapter outlines both functional and non-functional requirements and presents the key actors and their interaction processes. Together, these elements form the foundation for designing and developing a modern, intelligent clinical platform powered by voice-enabled documentation and automated patient management.

4.2 Purpose

Our purpose is to design and implement a structured educational platform that enables learners to gain foundational and advanced knowledge in AI algorithms, particularly in clustering and exploratory data analytics, through guided lessons, interactive visualizations, and providing structured projects.

4.3 Project Scope

The Voice-Based Medical System aims to streamline clinical workflows by providing voice-enabled documentation, intelligent patient management, and automated appointment handling. The system focuses on enhancing clinical efficiency through AI-powered features that support tasks such as generating medical notes, retrieving patient information, and organizing clinical data. The specification defines the high-level requirements of the platform and identifies the main actors interacting with the system.

4.4 Requirements Elicitation

While analyzing existing healthcare systems and conducting our feasibility study, we examined the strengths and weaknesses of other voice-enabled and EMR solutions. Our goal was to extract the best requirements from all evaluated systems in order to build a comprehensive, efficient, and user-friendly clinical platform. This ensures that the system supports clinicians at different experience levels and makes clinical documentation faster, smoother, and more intuitive through voice-driven interaction.

4.5 Requirements Table

Req_ID	Requirement Title	Category	Type
VEMR-FR-PM-01	The system should allow patients to create a new account by providing email, password, first name, and last name.	Patient Management	Functional
VEMR-FR-PM-02	The system should allow patients to login to the system using their email and password credentials.	Patient Management	Functional
VEMR-FR-PM-03	The system should allow patients to verify their email address before accessing the system.	Patient Management	Functional
VEMR-FR-PM-04	The system should allow users to request a password reset link via email and set a new password.	Patient Management	Functional
VEMR-FR-PM-05	The system should allow users to view their profile information including personal details.	Patient Management	Functional
VEMR-FR-PM-06	The system should allow users to update their profile information such as name, phone number, and other details	Patient Management	Functional
VEMR-FR-PM-07	The system should allow users to change their current password to a new one.	Patient Management	Functional
VEMR-FR-AU-08	The system should allow administrators to login to the system using their credentials.	Authentication	Functional
VEMR-FR-AU-09	The system should allow clinicians/doctors to login to the system using their email and password.	Authentication	Functional

Chapter 4 - System Analysis

VEMR-FR-VM-10	The system should allow clinicians to create a new patient visit with date, type, reason, and chief complaint.	Visit Management	Functional
VEMR-FR-VM-11	The system should allow clinicians to search visits by patient name or reason for visit.	Visit Management	Functional
VEMR-FR-VM-12	The system should allow clinicians to edit visit details when the visit is in progress.	Visit Management	Functional
VEMR-FR-VM-13	The system should allow clinicians to save changes made to a visit including medical record data.	Visit Management	Functional
VEMR-FR-VM-14	The system should allow clinicians to delete a visit from the system.	Visit Management	Functional
VEMR-FR-VM-15	The system should allow clinicians to view complete details of a visit including medical record, vitals, and allergies.	Visit Management	Functional
VEMR-FR-VM-16	The system allows the clinician to create and save a new medical record for a patient after validating required information.	Visit Management	Functional
VEMR-FR-VM-17	The system allows the clinician to edit and update an existing medical record while ensuring data validation.	Visit Management	Functional
VEMR-FR-VM-18	The system allows the clinician to finalize a medical record, update its status, and prevent further modifications.	Visit Management	Functional

Chapter 4 - System Analysis

VEMR-FR-VM-19	The system should allow voice transcription to appear in real-time as the clinician speaks using Whisper.	Visit Management	Functional
VEMR-FR-VM-20	The system should allow clinicians to view a paginated list of all patients in the system	Visit Management	Functional
VEMR-FR-VM-21	The system allows clinicians to search for patient	Visit Management	Functional
VEMR-FR-VM-22	The system should allow clinicians to view a patient's profile with their visits and allergies.	Visit Management	Functional
VEMR-FR-VM-23	The system should allow clinicians to view detailed patient information including contact and demographic data.	Visit Management	Functional
VEMR-FR-VM-24	The system should allow clinicians to change visit status: start (planned to in-progress), complete, or cancel.	Visit Management	Functional
VEMR-FR-VM-25	The system should allow clinicians to view a paginated list of all visits with filtering options.	Visit Management	Functional
VEMR-FR-VS-26	The system should allow clinicians to view vital signs recorded for a specific visit.	Vital Signs Management	Functional
VEMR-FR-VS-27	The system should allow clinicians to record and edit vital signs (BP, HR, temp, etc.) during an active visit.	Vital Signs Management	Functional
VEMR-FR-AM-28	The system should allow clinicians to view a patient's recorded allergies with severity and reaction details.	Allergy Management	Functional

Chapter 4 - System Analysis

VEMR-FR-AM-29	The system should allow clinicians to add a new allergy record for a patient with type, allergen, severity, and reaction.	Allergy Management	Functional
VEMR-FR-AM-30	The system should allow clinicians to update existing allergy information for a patient.	Allergy Management	Functional
VEMR-FR-AM-31	The system should allow clinicians to delete an allergy record from a patient's profile	Allergy Management	Functional
VEMR-FR-CM-32	The system should allow the admin to create clinicians' accounts.	Clinician Management	Functional
VEMR-FR-CM-33	The system should allow the clinicians and admins to update their account.	Clinician Management	Functional
VEMR-FR-CM-34	The system should allow the admin to view the clinicians accounts list	Clinician Management	Functional
VEMR-FR-CM-35	The system should allow the admin to view the clinicians account details.	Clinician Management	Functional
VEMR-FR-CM-36	The system should allow the admin to delete the clinician's accounts.	Clinician Management	Functional
VEMR-FR-CM-37	The system should allow the admin to search the clinician's accounts.	Clinician Management	Functional
VEMR-FR-OM-38	The system should allow the admin to create organizations.	Organization Management	Functional
VEMR-FR-OM-39	The system should allow the admin to update organization information.	Organization Management	Functional

Chapter 4 - System Analysis

VEMR-FR-OM-40	The system should allow the admin to view all organizations.	Organization Management	Functional
VEMR-FR-OM-41	The system should allow the admin to view detailed information about a specific organization.	Clinician Management	Functional
VEMR-FR-OM-42	The system should allow the admin to delete an organization	Clinician Management	Functional
VEMR-FR-OM-43	The system should allow the admin to search for organizations.	Clinician Management	Functional
VEMR-FR-AN-44	The system should allow administrators to view system-wide analytics and reports.	Analytics	Functional
VEMR-FR-AP-45	The system should allow clinicians to view their appointment schedule.	Appointment Management	Functional
VEMR-FR-AP-46	The system should allow clinicians to view their daily appointment schedule.	Appointment Management	Functional
VEMR-FR-AP-47	The system should allow clinicians to filter appointments by specific date.	Appointment Management	Functional
VEMR-FR-AP-48	The system should allow clinicians to filter appointments by status.	Appointment Management	Functional
VEMR-FR-AP-49	The system should allow the doctor to check in a patient's appointment	Appointment Management	Functional
VEMR-FR-AP-50	The system should allow the doctor to cancel a patient's appointment	Appointment Management	Functional

Chapter 4 - System Analysis

VEMR-FR-AP-51	The system should allow the doctor to set an appointment as no show	Appointment Management	Functional
VEMR-FR-DA-52	The system should allow clinicians to view their personal analytics and performance metrics.	Doctor Analytics	Functional
VEMR-FR-SM-53	The system should allow clinicians to create their work schedule.	Schedule Management	Functional
VEMR-FR-SM-54	The system should allow clinicians to edit their existing work schedule.	Schedule Management	Functional
VEMR-FR-SM-55	The system should allow clinicians to delete their work schedule.	Schedule Management	Functional
VEMR-FR-SM-56	The system should allow clinicians to automatically generate available visit slots based on their schedule.	Schedule Management	Functional
VEMR-FR-SM-57	The system should allow clinicians to mark appointments as completed, no-show, or cancelled.	Schedule Management	Functional
VEMR-FR-SM-58	The system should allow clinicians to view detailed information about a specific appointment.	Schedule Management	Functional
VEMR-FR-SM-59	The system should allow clinicians to define their available time slots for appointments.	Schedule Management	Functional
VEMR-FR-PP-60	The system should allow patients to view available healthcare organizations.	Patient Portal	Functional

Chapter 4 - System Analysis

VEMR-FR-PP-61	The system should allow patients to view doctors within specific organizations	Patient Portal	Functional
VEMR-FR-PP-62	The system should allow patients to view available appointment slots.	Patient Portal	Functional
VEMR-FR-PP-63	The system should allow patients to book an appointment with a clinician.	Patient Portal	Functional
VEMR-FR-PP-64	The system should allow patients to view their own appointment history and upcoming appointments.	Patient Portal	Functional
VEMR-FR-PP-65	The system should allow patients to filter their appointments by status.	Patient Portal	Functional
VEMR-FR-PP-66	The system should allow patients to view their own medical records.	Patient Portal	Functional
VEMR-FR-PP-67	The system should allow patients to view their own allergy information.	Patient Portal	Functional
VEMR-FR-PP-68	The system should allow patients to view their complete visit history.	Patient Portal	Functional

Table 7, Requirements Table

4.6 Analysis

In this Section, we will introduce the analytical study for the first sprint (Account Management and User Authentication). The analysis includes identifying and modeling the primary functional requirements through Unified Modeling Language (UML) diagrams, such as use case, activity and sequence diagrams, to visualize system behavior and interactions. This structured approach ensures alignment between user needs, functional goals, and software components, providing a clear roadmap for implementation and testing during this sprint.

- The list of requirements that we will complete for this sprint:

- ❖ **VEMR-FR-AM-01:** The system should allow clinicians to register for an account using their email and password
- ❖ **VEMR-FR-AM-02:** The system should allow clinicians to log in to their accounts using their credentials
- ❖ **VEMR-FR-AM-03:** The system should allow users to verify their accounts via email verification link
- ❖ **VEMR-FR-AM-04:** The system should allow users to reset their password using the emailed link
- ❖ **VEMR-FR-AM-05:** The system should allow users to view their profile data
- ❖ **VEMR-FR-AM-06:** The system should allow users to update their profile information
- ❖ **VEMR-FR-AM-07:** The system should allow users to change their

4.6.1 Requirements Modeling

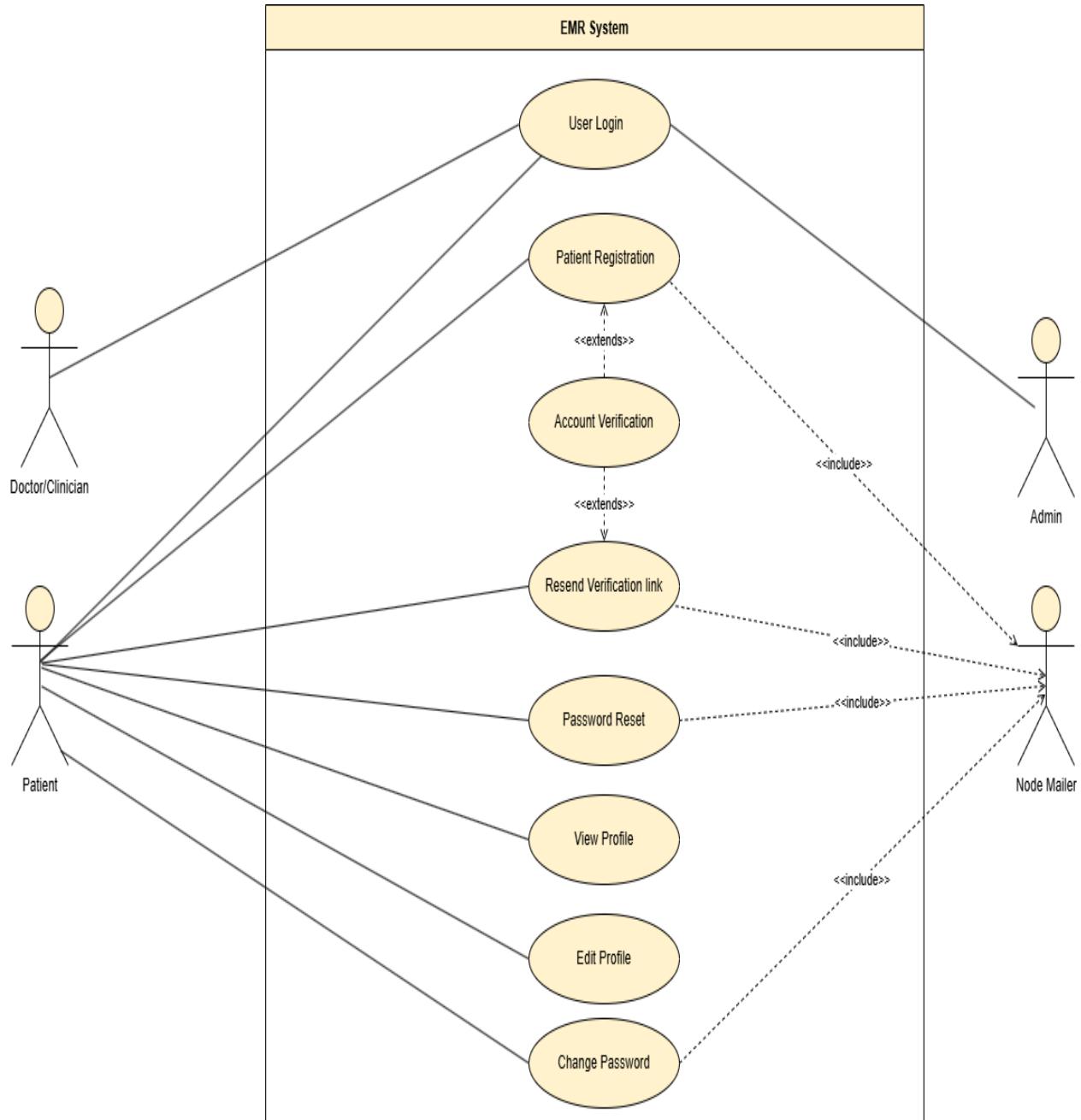


Figure 5, Sprint 1 Use Case Diagram

Chapter 4 - System Analysis

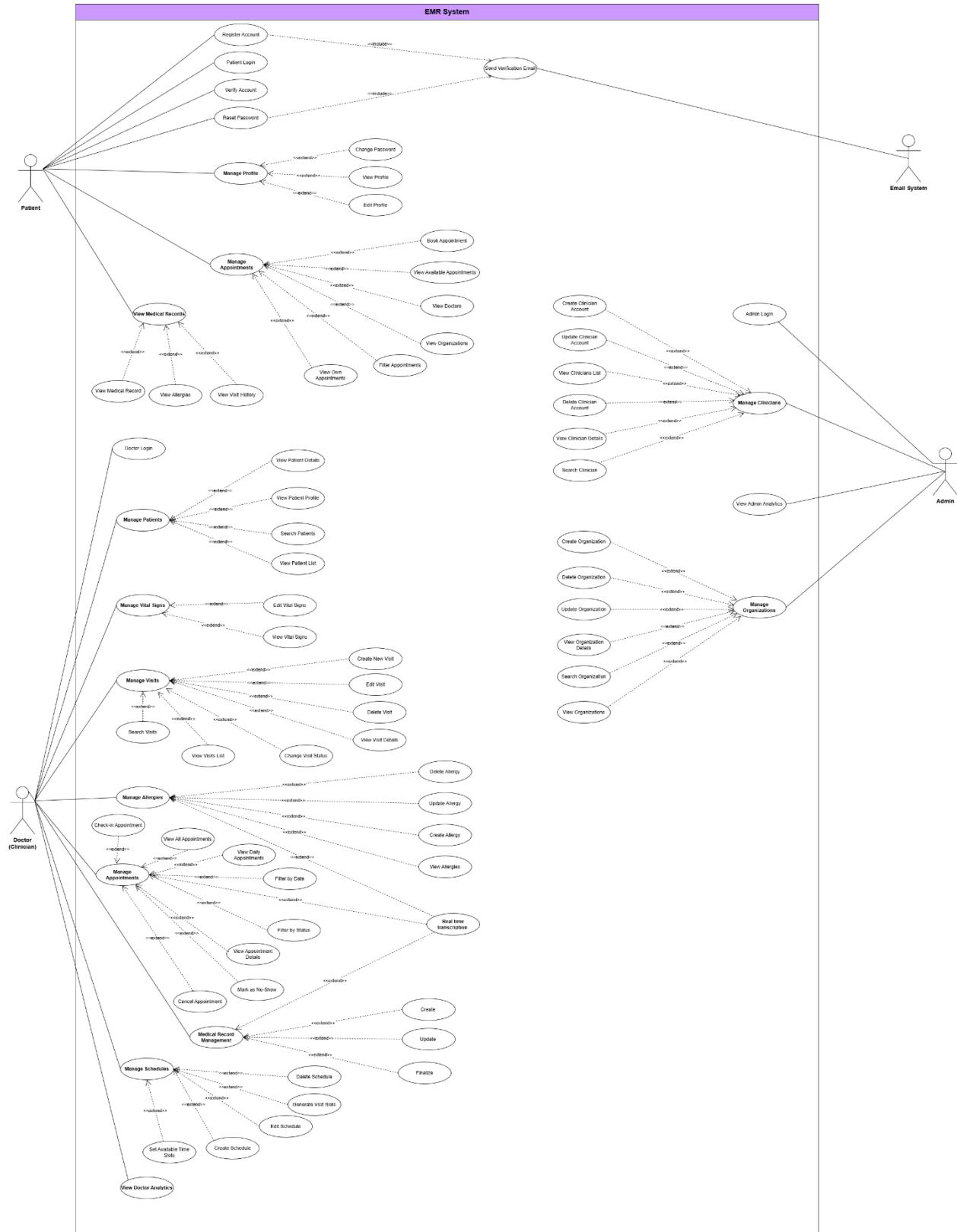


Figure 4: Low-Level Use Case Diagram

- **Sign up:**

Use case ID	VEMR-FR-PM-01
Use case name	Patient Registration
Description	The system allow patient to register for an account using their email and password
Actor	Patient
Pre-conditions	Patient does not have an exist
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to the registration page 2. Patient enters registration information: (First name, Last name, Email, Password) 3. System validates the input data 4. System checks if email is already registered 5. System creates patient account 6. System generates verification token 7. System send verification email to patient 8. System return success message
Alternative scenario	<p>A1: Email Already Registered</p> <ul style="list-style-type: none"> - At step 4, if email already exists - System returns Error." Email already registered" <p>A2: Invalid Input Data</p> <ul style="list-style-type: none"> - At step 3, if validation fails - System return validation error <p>A3: Email Sending Failure</p> <ul style="list-style-type: none"> - At step 8, if email fails to send - System logs the error - Account is still created - Patient can request resend verification email later
Post condition	Patient is authenticated

Table 8 , Use Case Specification (Sign up)

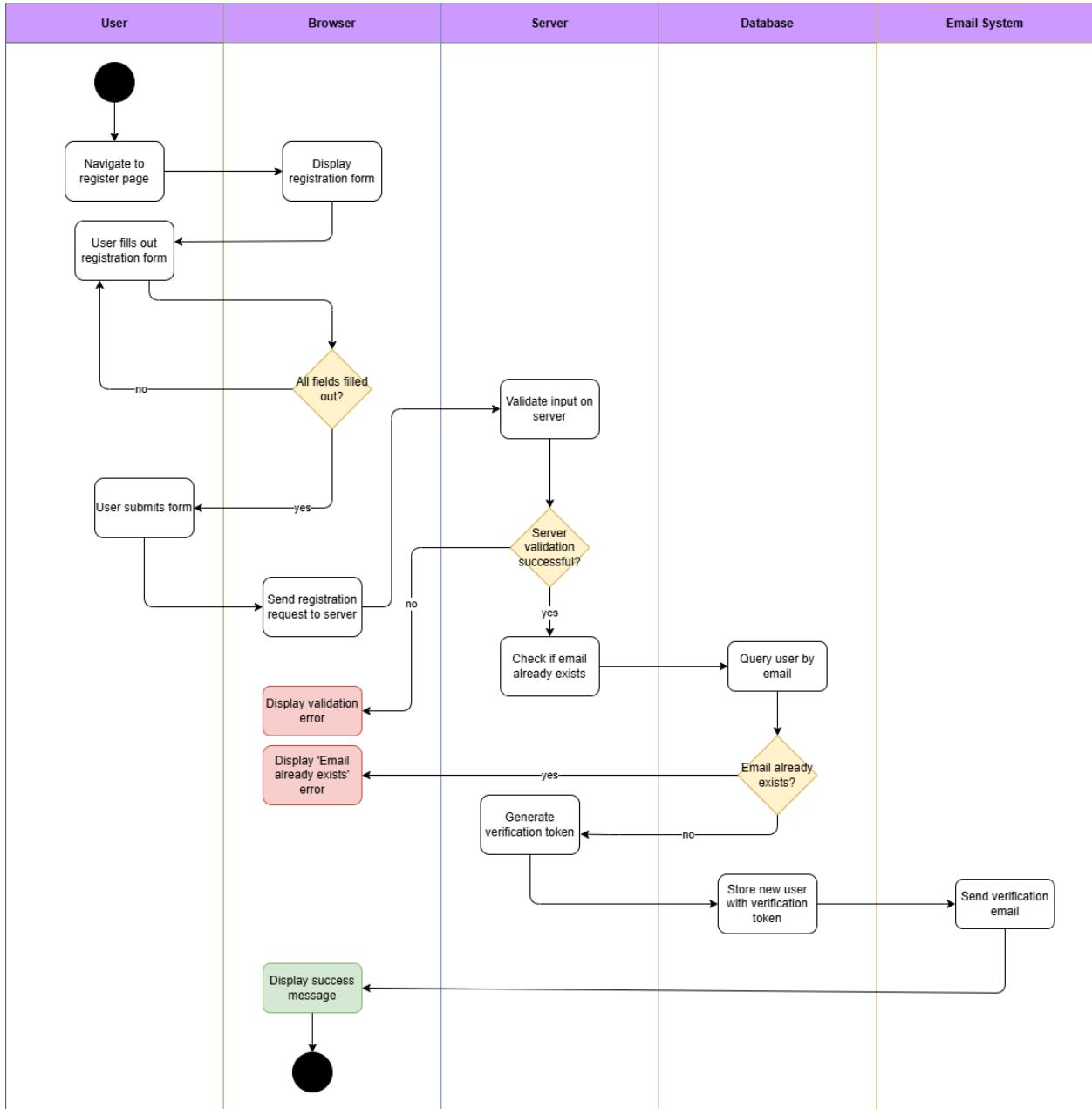


Figure 6, Activity Diagram (Sign up)

Chapter 4 - System Analysis

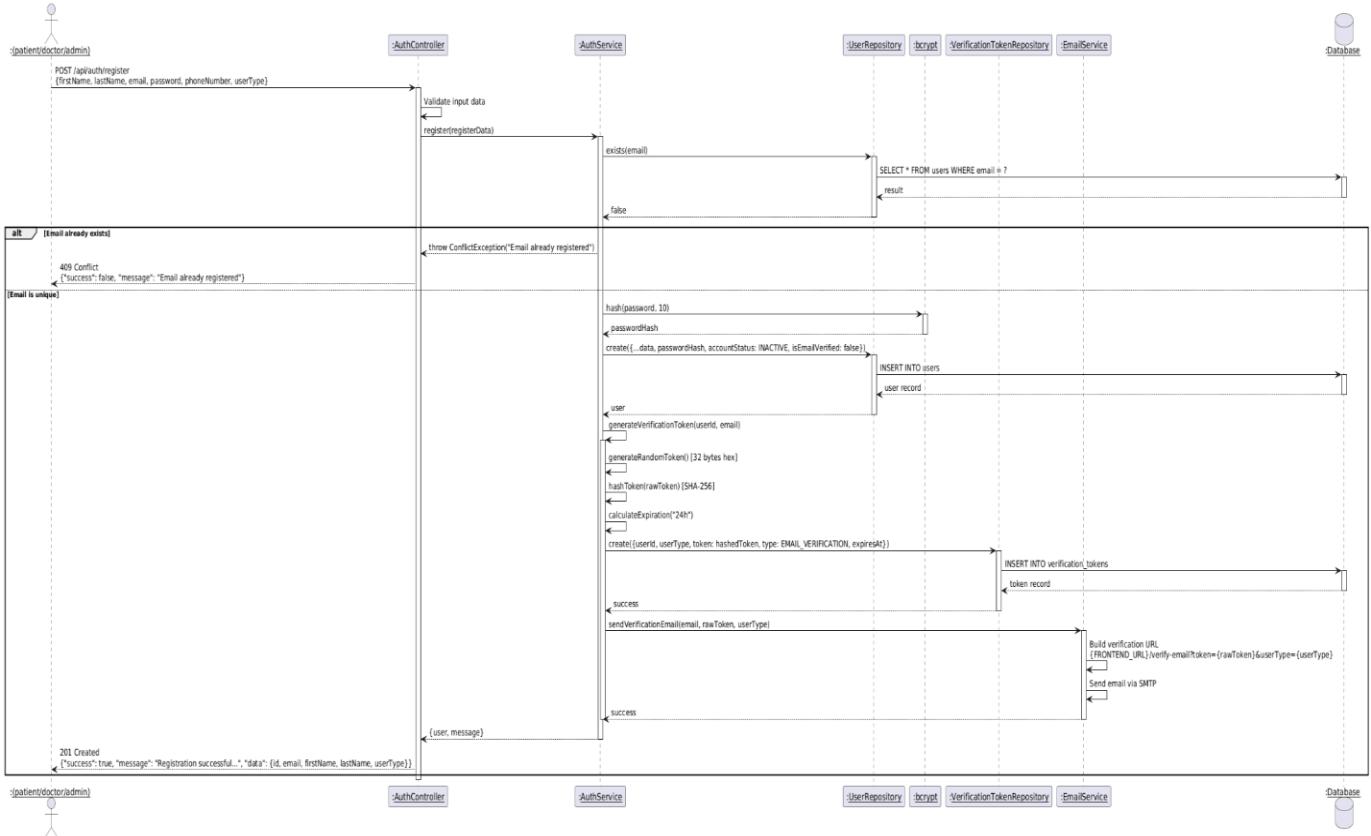


Figure 7, Sequence Diagram (Sign up)

- **Log in:**

Use case ID	VEMR-FR-PM-02
Use case name	Patient Login
Description	This use case allows the Patient to login into his account
Actor	Patient
Pre-conditions	<ul style="list-style-type: none"> - User has Registered account -User email is verified
Main scenario	<ol style="list-style-type: none"> 1. User navigate to login page 2. User enters credentials (Email, Password) 3. System validates the input data 4. System retrieves User record by email 5. System compares password with stored hash 6. System verifies email is verified 7. System checks account status is ACTIVE 8. System updates last login timestamp 9. System generate JWT access token and refresh token 10. System returns tokens and User information
Alternative scenario	<p>A1: Invalid Credentials</p> <ul style="list-style-type: none"> - At step 5, if email not found or password doesn't match - System returns error: "Invalid credentials" <p>A2: Email not verified</p> <ul style="list-style-type: none"> - At step 5, if email is not verified - System return error: "Please verify your email before login" <p>A3: Account Not active</p> <ul style="list-style-type: none"> - At step 7, if account status is not ACTIVE - System return error: "Account is not active"
Post condition	User is authenticated

Table 9 , Use Case Specification (Log in)

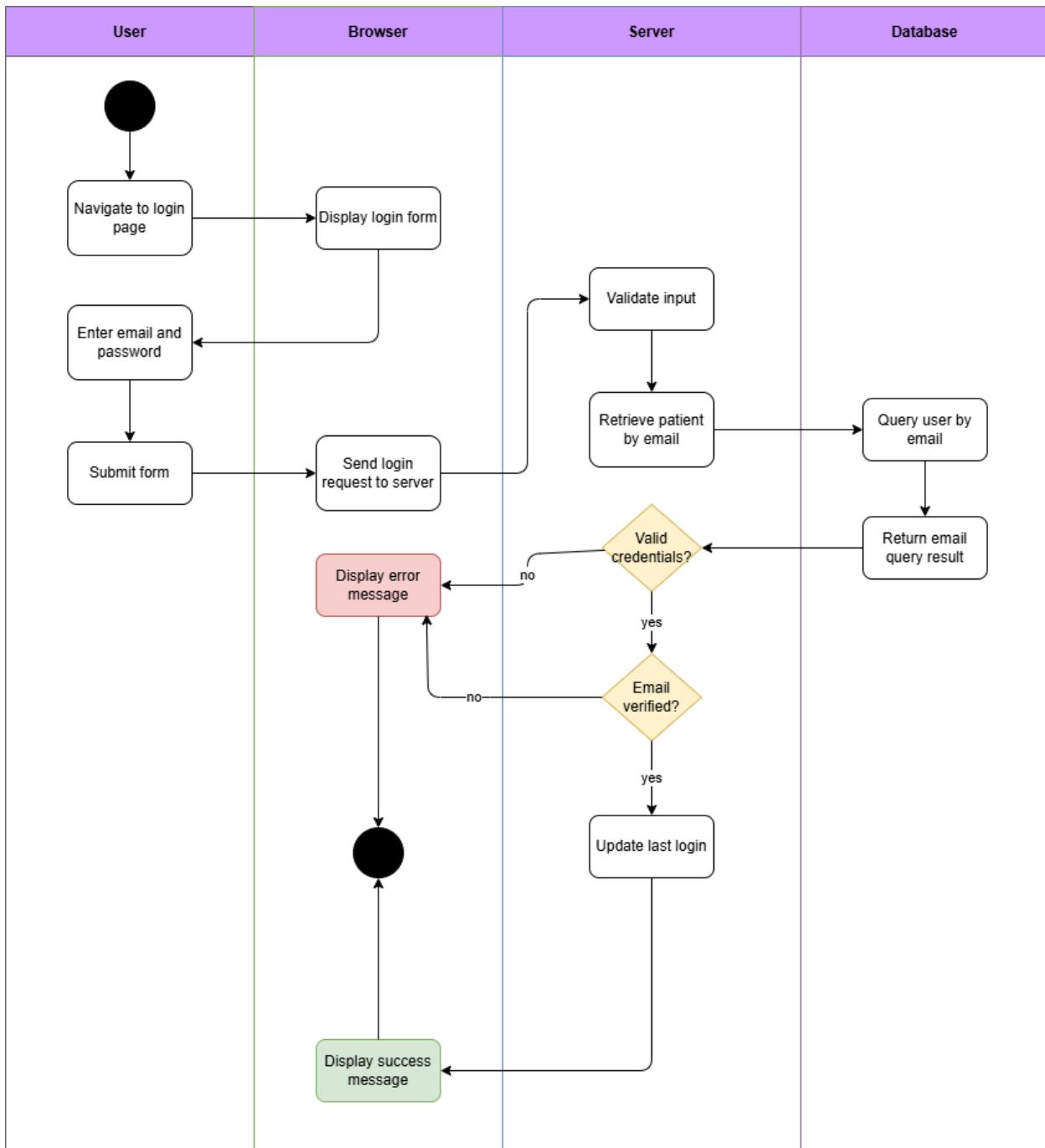


Figure 8, Activity Diagram (Log in)

Chapter 4 - System Analysis

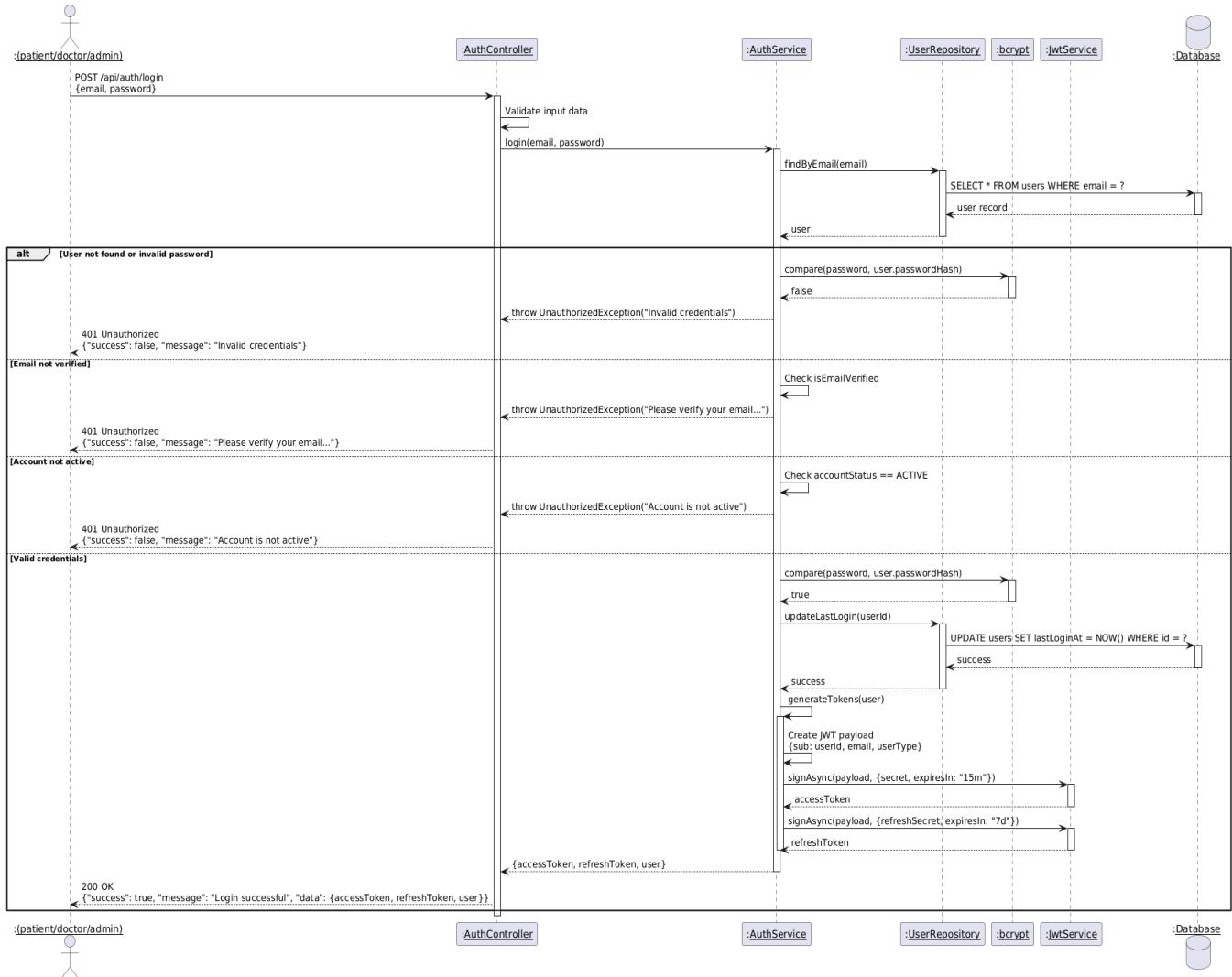


Figure 9, Sequence Diagram (Log in)

● Account Verification:

Use case ID	VEMR-FR-PM-03
Use case name	Account Verification
Description	The system allows users to verify their account via email verification link
Actor	Patient
Pre-conditions	Verification email has been sent
Main scenario	<ol style="list-style-type: none"> 1. User receives a verification email 2. User clicks the verification link 3. System validates the token 4. System verifies token type, user type, expiration, and usage 5. System verifies the user's email 6. System activates the account 7. System returns a success message
Alternative scenario	<p>A1: invalid Token</p> <ul style="list-style-type: none"> - At step 5, if the token is not found in the database - System returns error: "Invalid or expired verification token" <p>A2: Token already used</p> <ul style="list-style-type: none"> - At step 9, if the token is already marked used - System return error: "Verification token has already been used"
Post condition	User account status is changed to ACTIVE

Table 10, Use Case Specification (Account Verification)

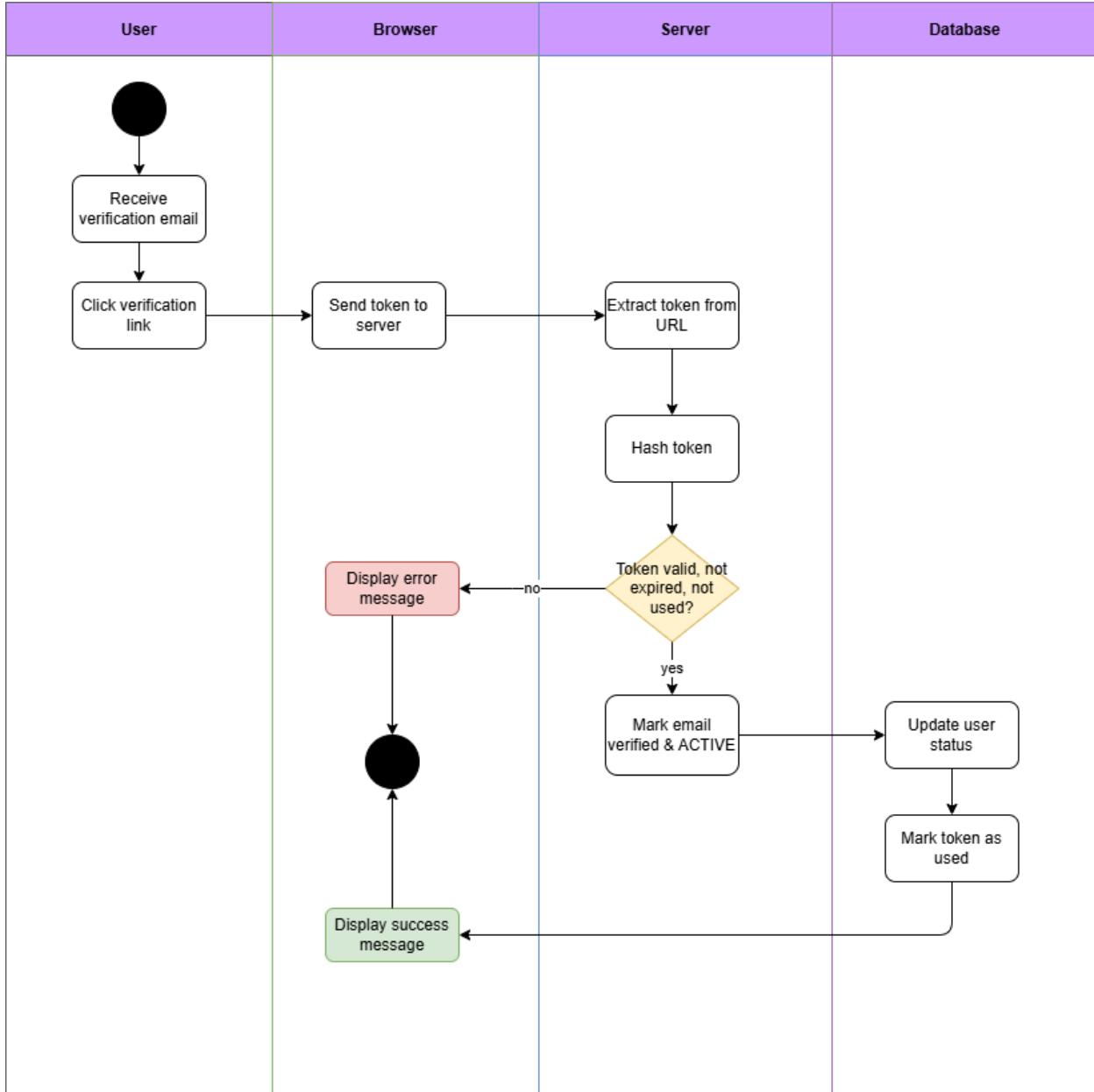


Figure 10, Activity diagram (Account Verification)

Chapter 4 - System Analysis

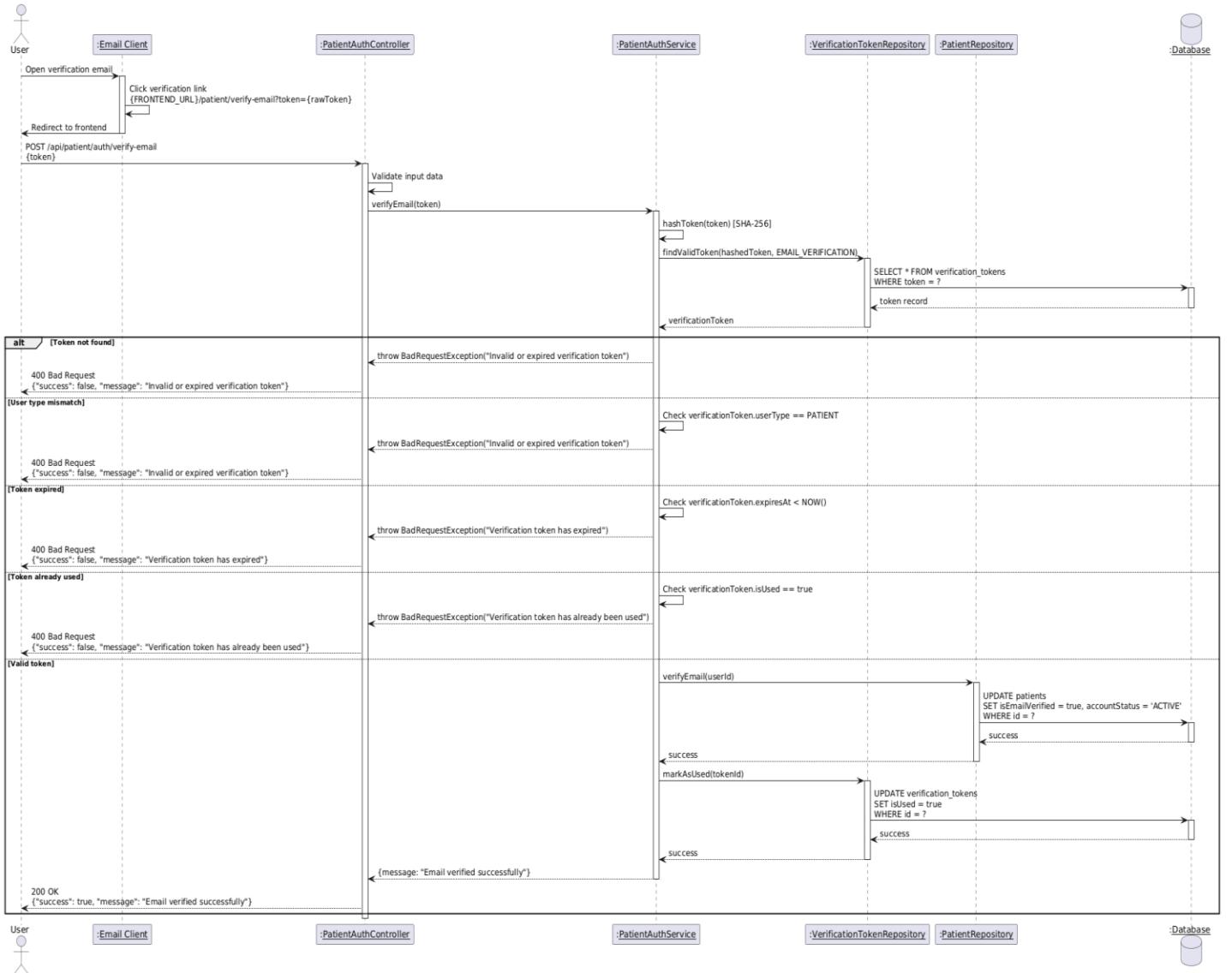


Figure 11, Sequence Diagram (Account Verification)

● Reset Password:

Use case ID	VEMR-FR-PM-04
Use case name	Reset password
Description	The system allows users to reset their password using an email link
Actor	Patient
Pre-conditions	Users has a registered account
Main scenario	<ol style="list-style-type: none"> 1. User opens the Forgot Password page 2. Users enter their email address 3. System sends a password reset link to the email 4. User opens the reset link 5. User enters a new password 6. System saves the new password and shows a success message
Alternative scenario	<p>A1:Email Not Found</p> <ul style="list-style-type: none"> - At step 3, if the email is not found in the database - System return error: " Invalid or expired verification token" <p>A2:Token Already used</p> <ul style="list-style-type: none"> - At step 5, if the token is already marked as used - System return error: " Verification token has already been used"
Post condition	User password is updated

Table 11, Use Case Specification (Reset Password)

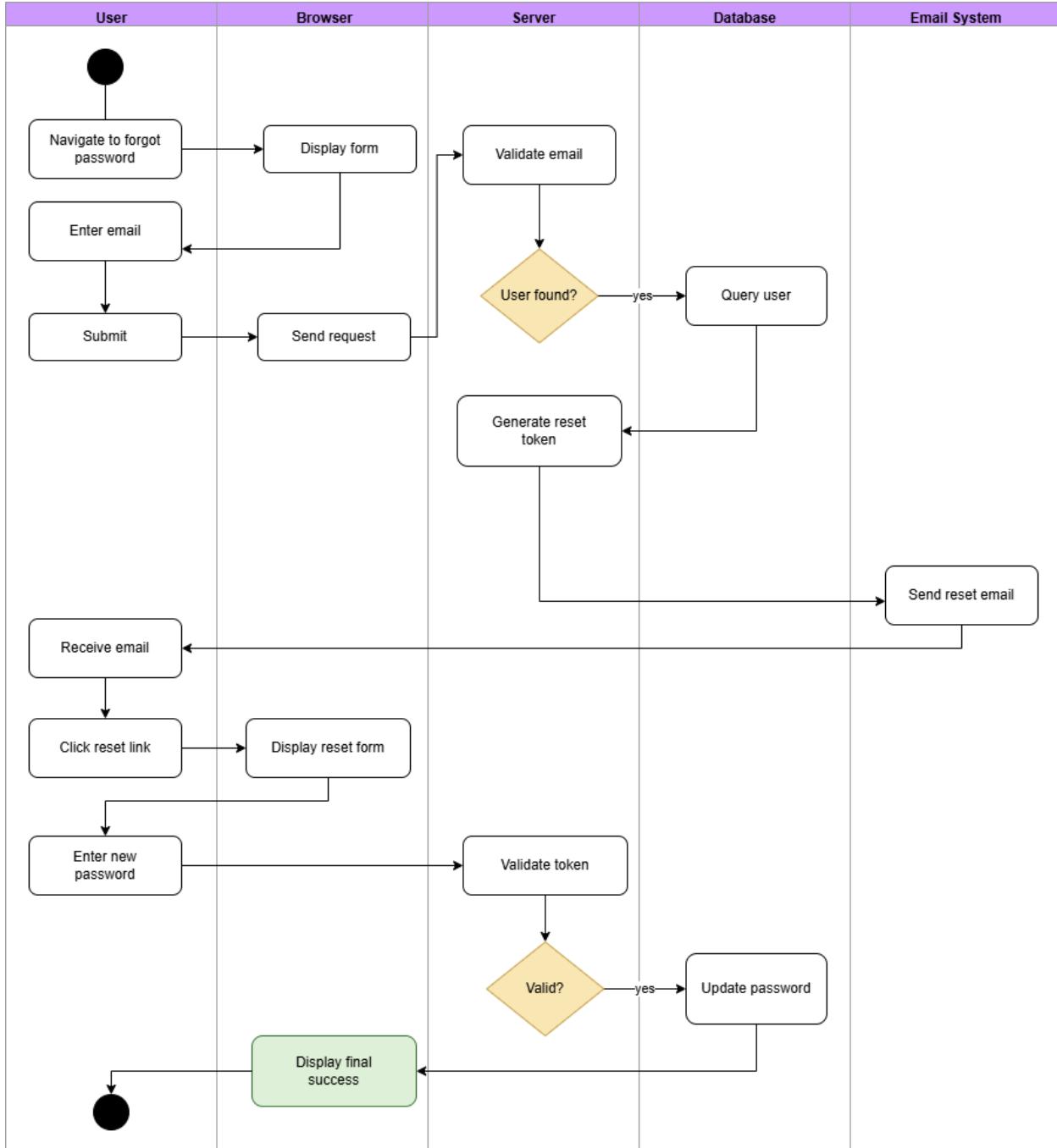


Figure 12, Activity diagram (Reset Password)

Chapter 4 - System Analysis

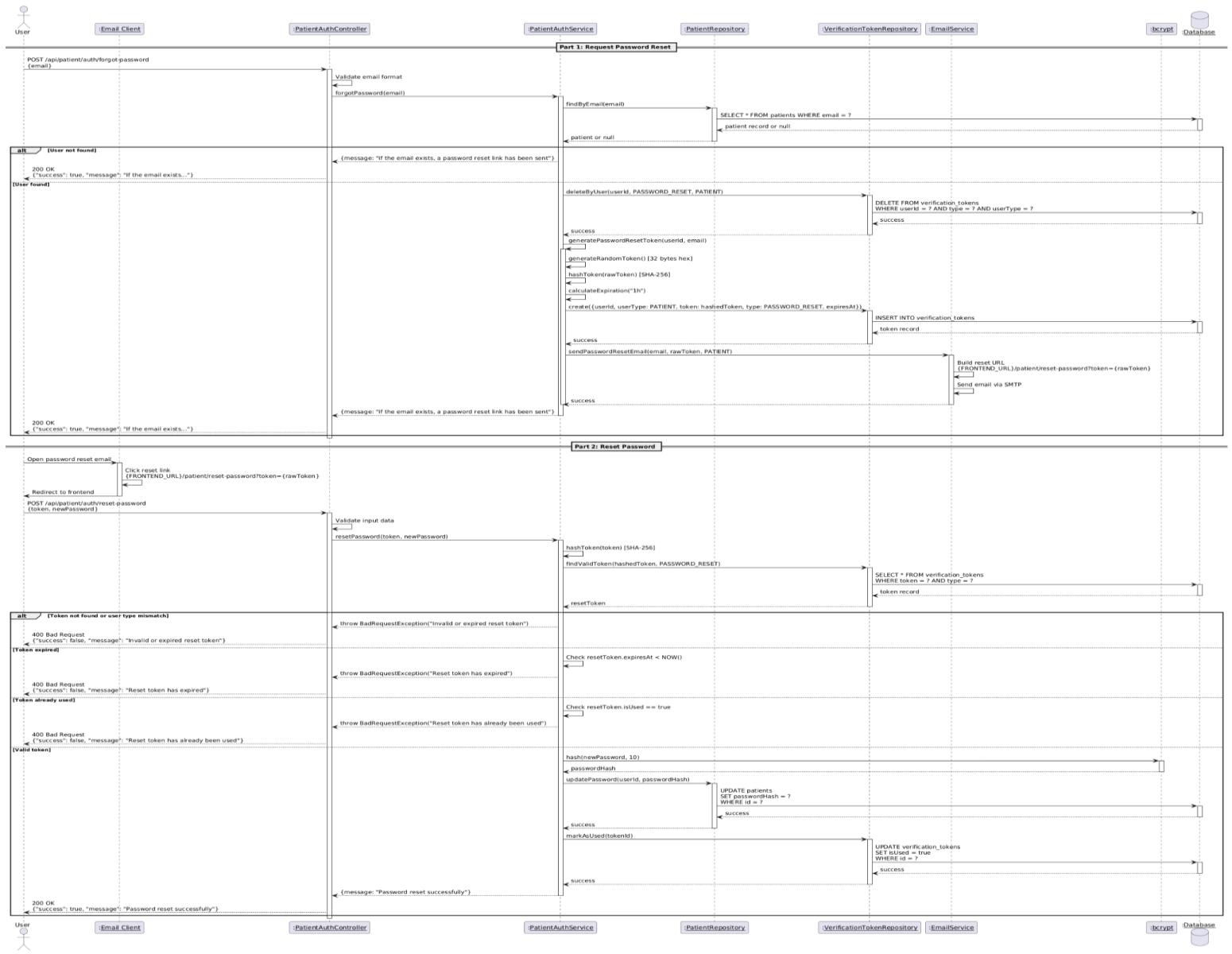


Figure 13, Sequence Diagram (Reset Password)

- **View Profile:**

Use case ID	VEMR-FR-PM-05
Use case name	View profile
Description	The system allows users to view profile data.
Actor	Patient
Pre-conditions	User is logged in
Main scenario	<ol style="list-style-type: none"> 1. User navigates to profile page 2. System extracts JWT token from request header. 3. System validate JWT token 4. System extract user ID from token payload 5. System retrieves user profile from database by ID 6. System returns user profile information
Alternative scenario	<p>A1: Invalid token</p> <ul style="list-style-type: none"> - At step 3, if token is invalid or expired - System returns 401 Unauthorized error <p>A2: User not found</p> <ul style="list-style-type: none"> - At step 5, if user ID does not exist - System returns 404 Not Found error
Post condition	Users profile information is displayed

Table 12, Use Case Specification (View Profile)

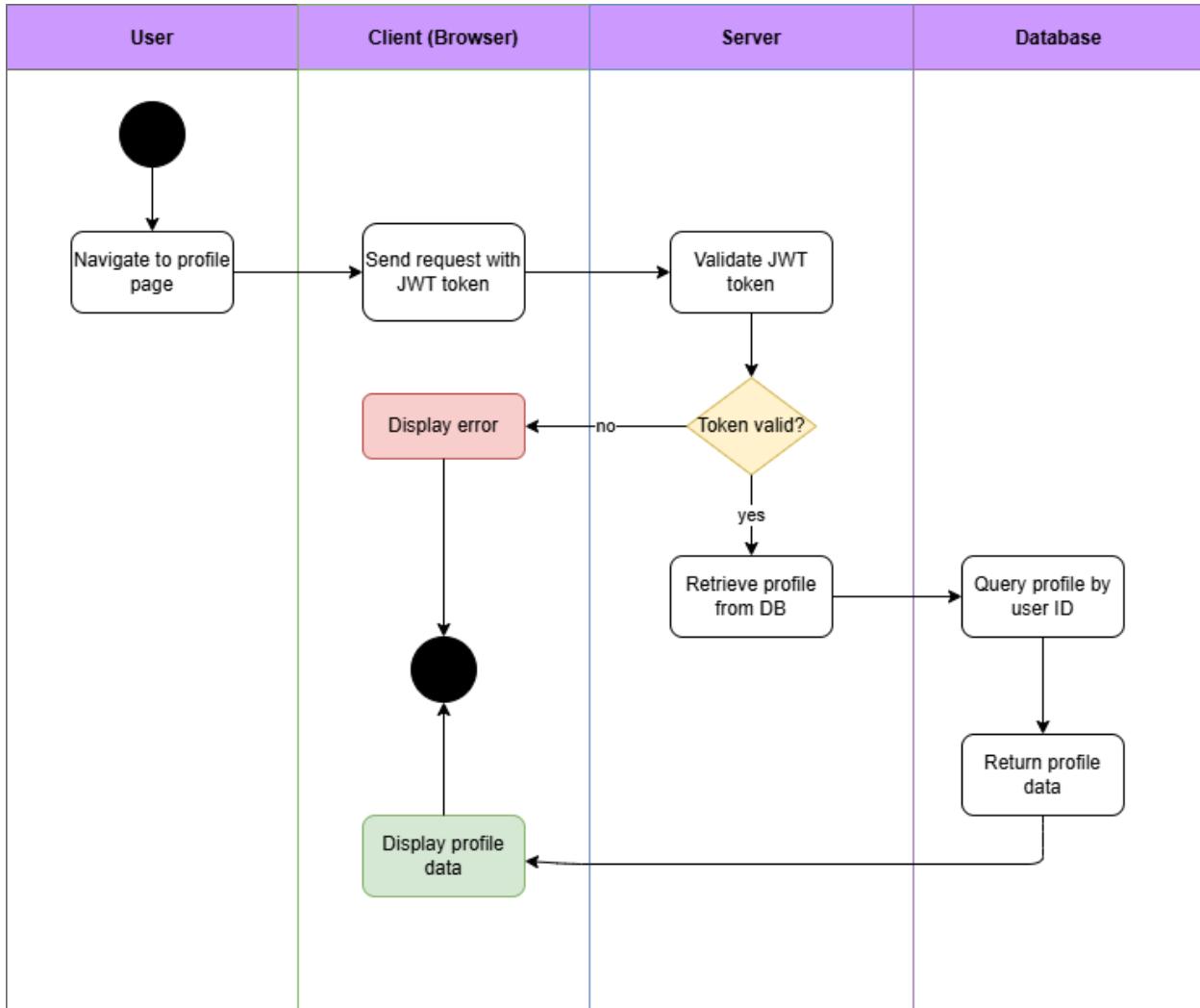


Figure 14, Activity Diagram (View Profile)

Chapter 4 - System Analysis

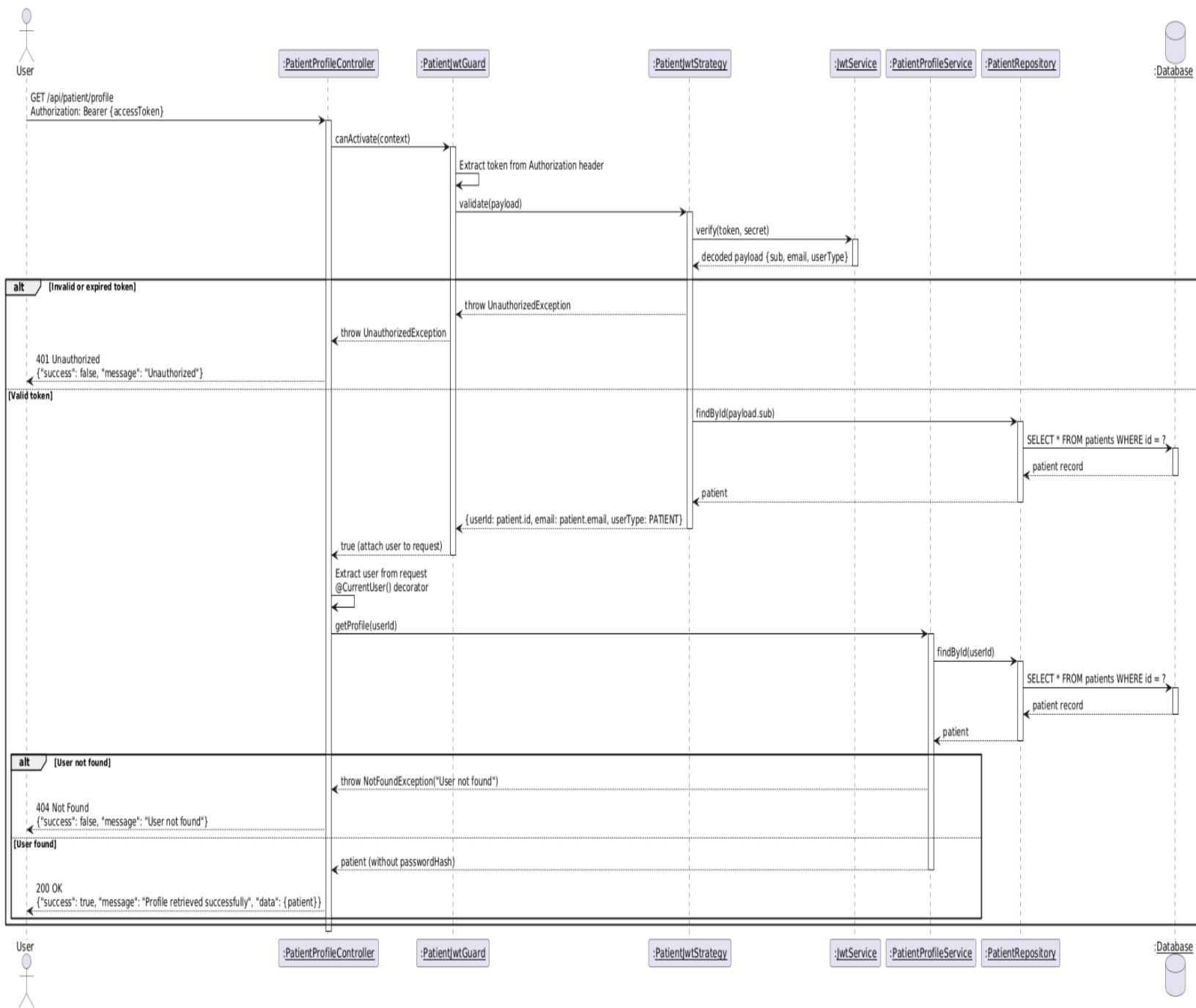


Figure 15, Sequence Diagram (View Profile)

● Edit Profile

Use case ID	VEMR-FR-PM-06
Use case name	Edit profile
Description	The system allows users to update their profile information
Actor	Patient
Pre-conditions	User us logged in
Main scenario	<ol style="list-style-type: none"> 1. User navigates to edit profile page 2. User views current profile information 3. User modifies desired fields 4. User submits the update form 5. System extracts JWT token from request header 6. System validates JWT token 7. System extracts user ID from token payload 8. System validates the input data 9. System updates user profile in database 10. System retrieves updated profile 11. System return updated profile information
Alternative scenario	<p>A1: Invalid Token</p> <ul style="list-style-type: none"> - At step 6, if token is invalid or expired - System return 401 Unauthorized error <p>A2: Invalid Input Data</p> <ul style="list-style-type: none"> - At step 8, if validation fails - System return validation error <p>A3: User Not Found</p> <ul style="list-style-type: none"> - At step 9, if user ID does not exist - System return 404 Not Found error
Post condition	User profile information is updated in the database

Table 13, Use Case Specification (Edit Profile)

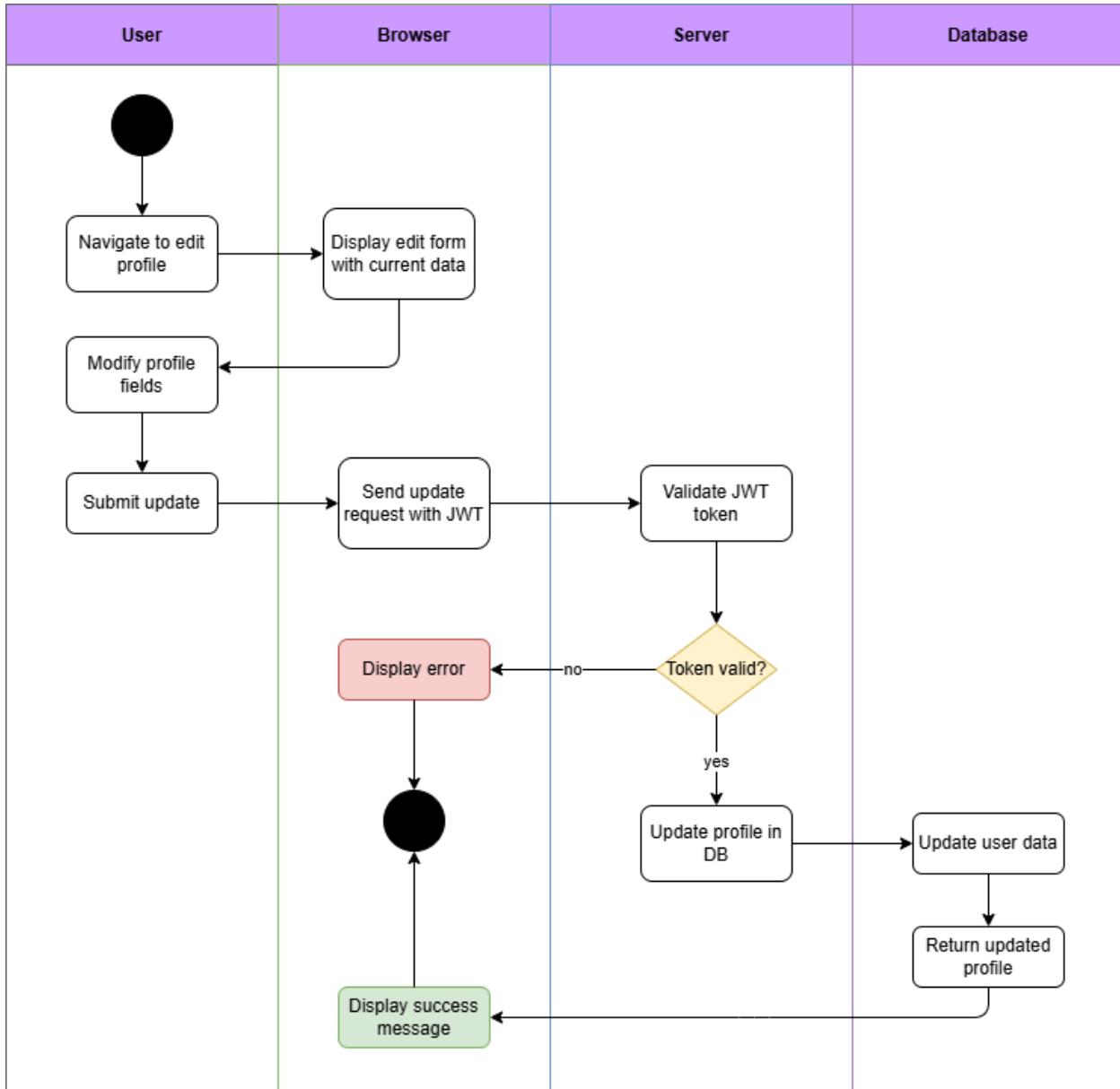


Figure 16, Activity Diagram (Edit Profile)

Chapter 4 - System Analysis

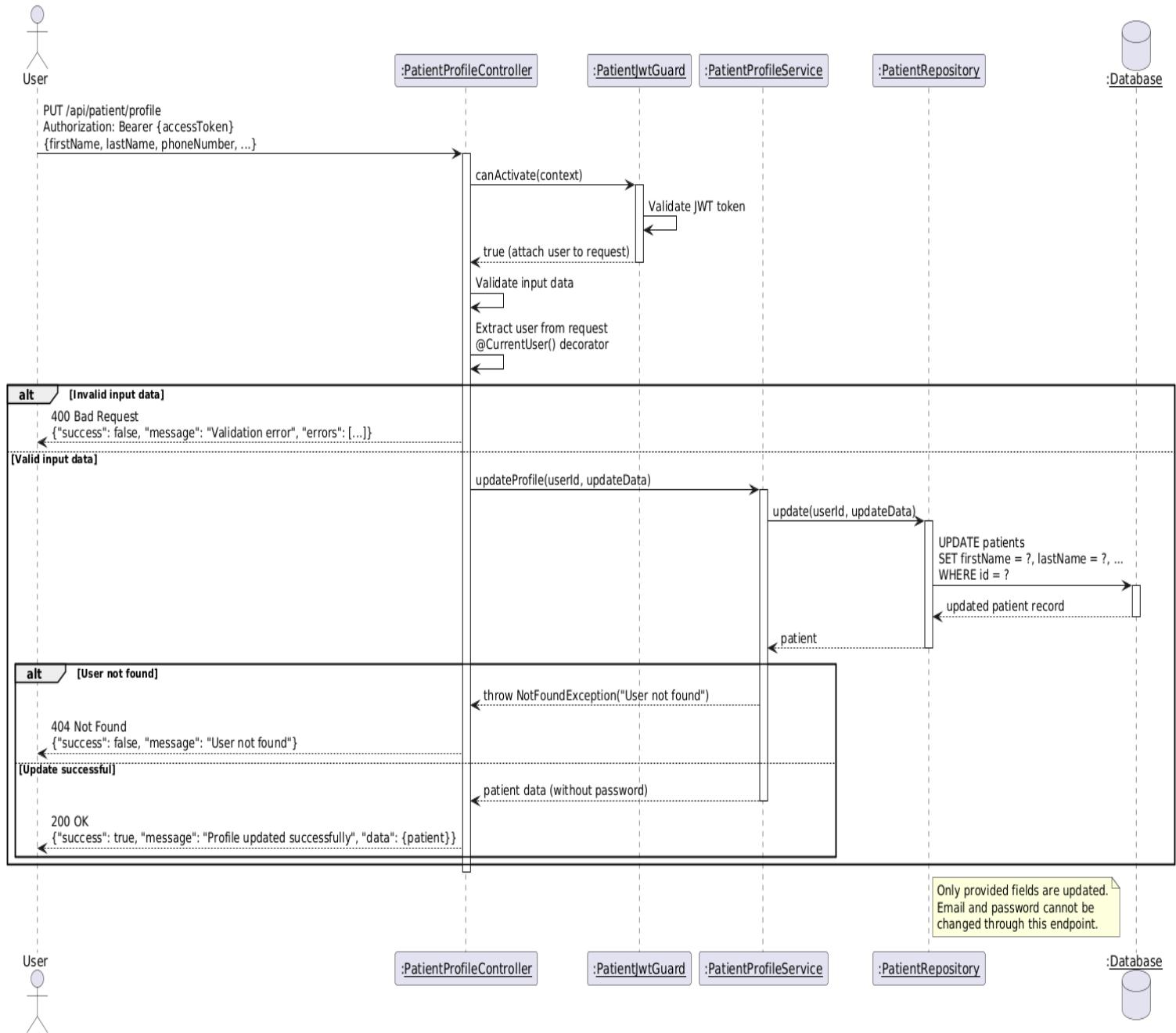


Figure 17, Sequence Diagram (Edit Profile)

● Change Password:

Use case ID	VEMR-FR-PM-07
Use case name	Change Password
Description	The system allows users to change their password
From	Patient
Pre-conditions	User is logged in
Main scenario	<ol style="list-style-type: none"> 1. User navigates to change password page 2. User enters current password then new password 3. User confirms new password 4. User submits the form 5. System extracts JWT token from request header and validate it 6. System extracts user ID from token payload 7. System validates input data 8. System retrieves user profile from data 9. System verifies current password 10. System hashes new password 11. System updates password in database 12. System returns success message
Alternative scenario	<p>A1: Invalid Token</p> <ul style="list-style-type: none"> - At step 5, if token is invalid or expired - System returns 401 Unauthorized error <p>A2: Invalid Input Data</p> <ul style="list-style-type: none"> - At step 7, if validation fails - System returns validation errors <p>A3: Incorrect Current Password</p> <ul style="list-style-type: none"> - At step 9, if current password does not match - System returns error: "Current password is incorrect"
Post condition	User password is updated in the database

Table 14, Use Case Specification (Change Password)

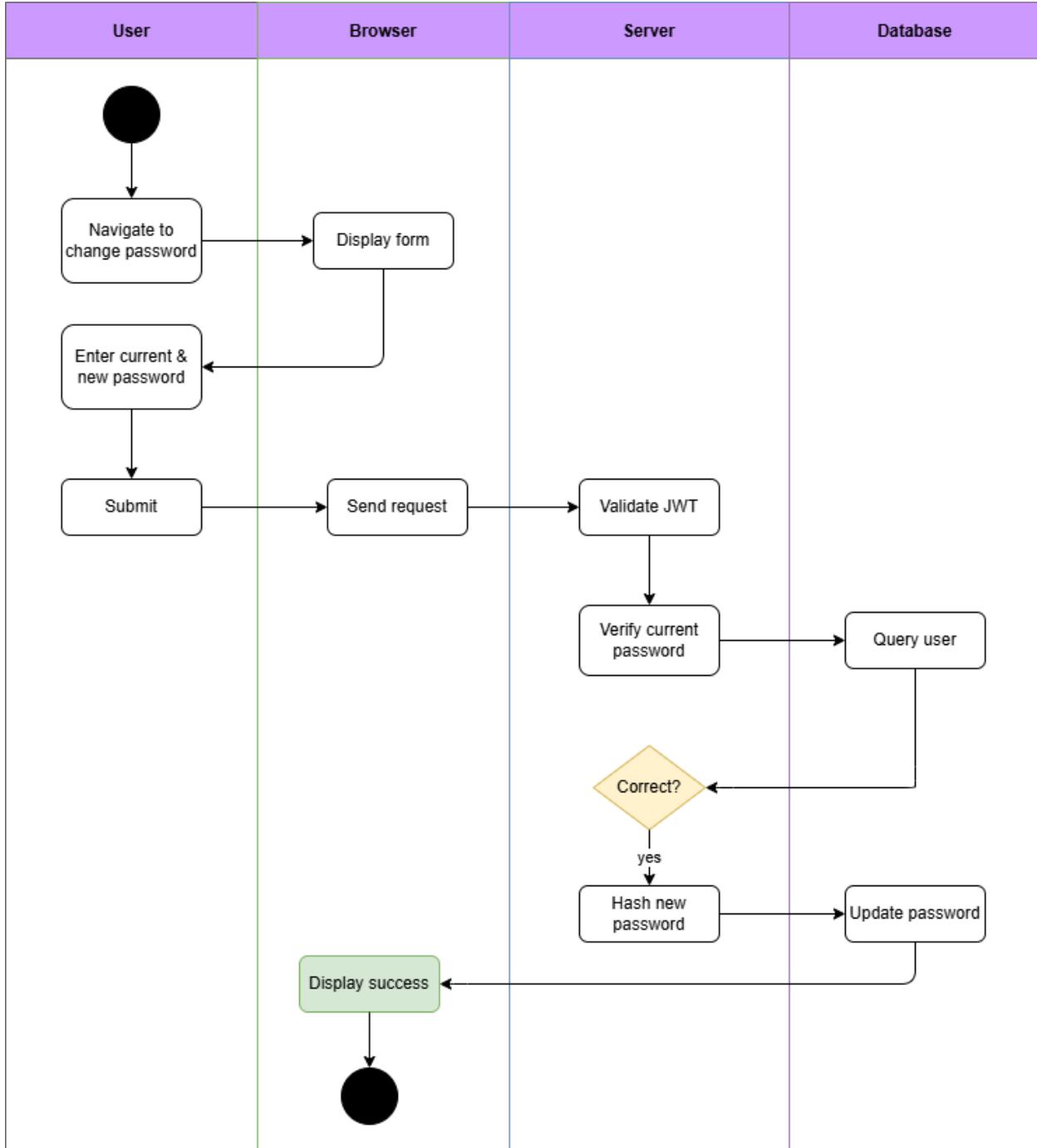


Diagram 18, Activity Diagram (Change Password)

Chapter 4 - System Analysis

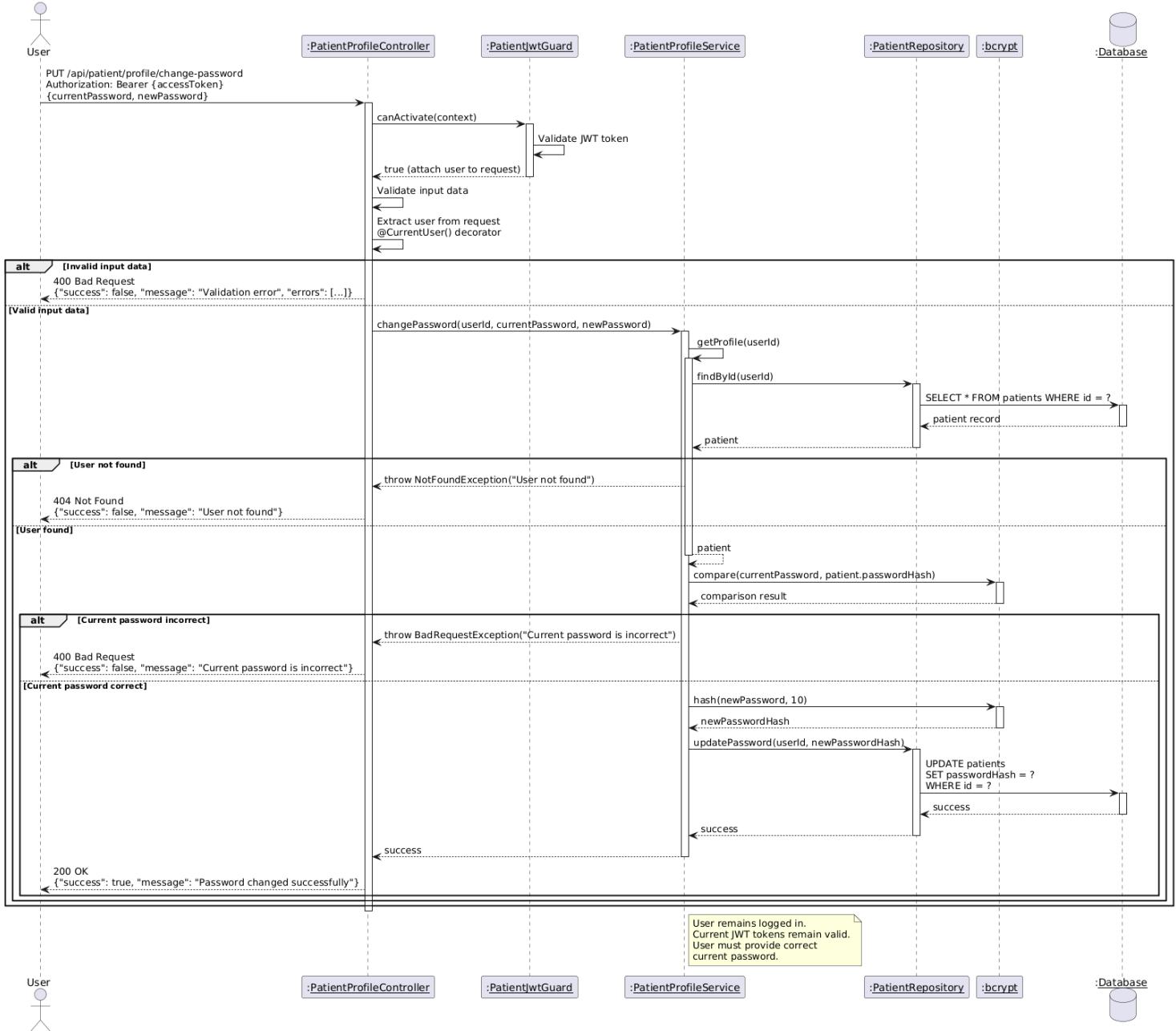


Figure 19, Sequence Diagram (Change Password)

● Admin Login:

Use case ID	VEMR-FR-AU-08
Use case name	Admin Login
Description	The system allows an administrator to log in using an email and password to access the admin dashboard.
Actor	Admin
Pre-conditions	Administrator account exists
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to the admin login page. 2. Administrator enters email and password. 3. Administrator clicks the Login button. 4. System validates the input data. 5. System verifies the credentials and generates a JWT token. 6. System stores the JWT token and redirects the administrator to the admin dashboard.
Alternative scenario	<p>A1: Invalid Credentials</p> <ul style="list-style-type: none"> - At step 5 , If the email or password is incorrect -System displays "Invalid email or password" <p>A2: Validation Error</p> <ul style="list-style-type: none"> - At step 4, If the email format is invalid or the password is empty -System displays a validation error message <p>A3: Already Logged In</p> <ul style="list-style-type: none"> - At step 1 ,If a valid JWT token already exists -System redirects the administrator directly to the admin dashboard
Post condition	Administrator is authenticated

Table 15, Use Case Specification (Admin Login)

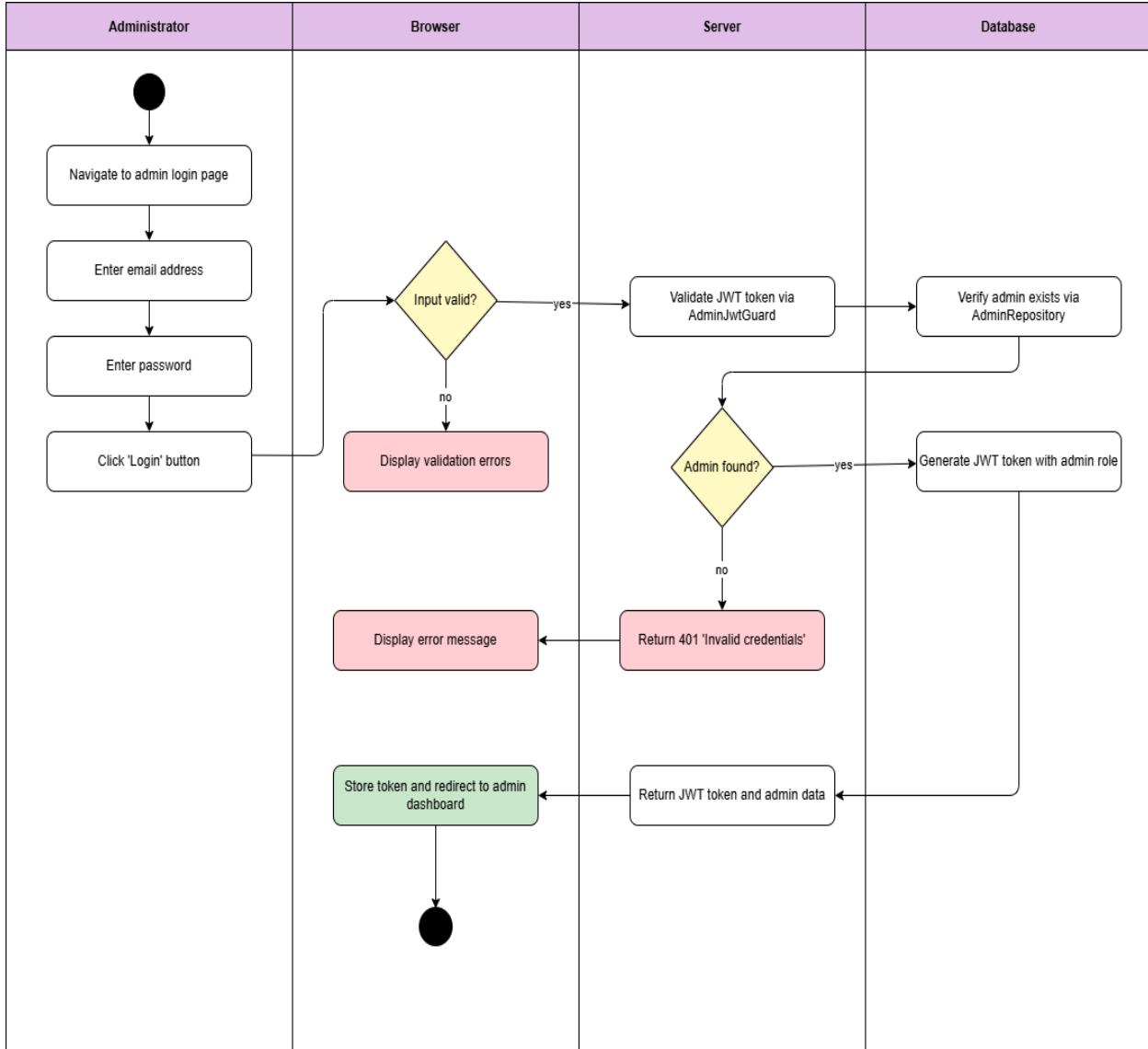


Diagram 20, Activity Diagram (Admin Login)

Chapter 4 - System Analysis

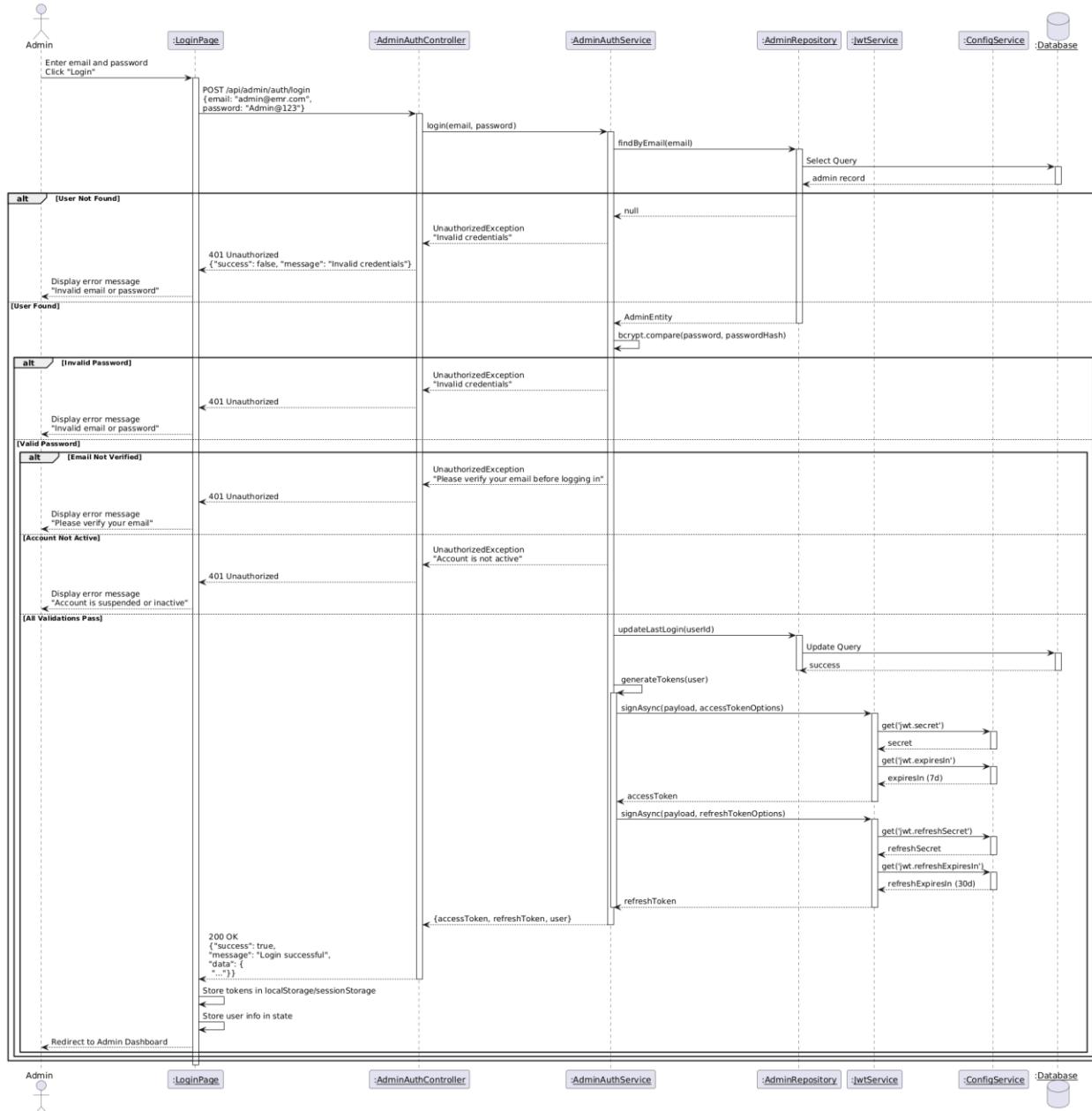


Diagram 21, Sequence Diagram (Admin Login)

● Clinician Login :

Use case ID	VEMR-FR-AU-09
Use case name	Clinician Login
Description	The system allows an administrator to log in using an email and password to access the Clinician dashboard.
Actor	Clinician
Pre-conditions	Clinician account exists in the system
Main scenario	<p>1.Clinician navigates to clinician login page 2.Clinician enters email address and password. 3. Clinician clicks "Login" button. 4. System validates the input data. 5.System verifies the credentials and generates a JWT token. 6.System stores the JWT token and redirects the administrator to the Clinician dashboard.</p>
Alternative scenario	<p>A1: Invalid Input - At step 4 , if the email format is invalid or the password is empty -System displays a validation error message</p> <p>A2: Account Not Found - At step 5, if the email or password is incorrect, or the account does not exist -System displays "Invalid email or password"</p> <p>A3: Already Logged In - At step 1 ,If a valid JWT token already exists -System redirects the clinician directly to the clinician dashboard</p>
Post condition	Administrator is authenticated

Table 16, Use Case Specification (Clinician Login)

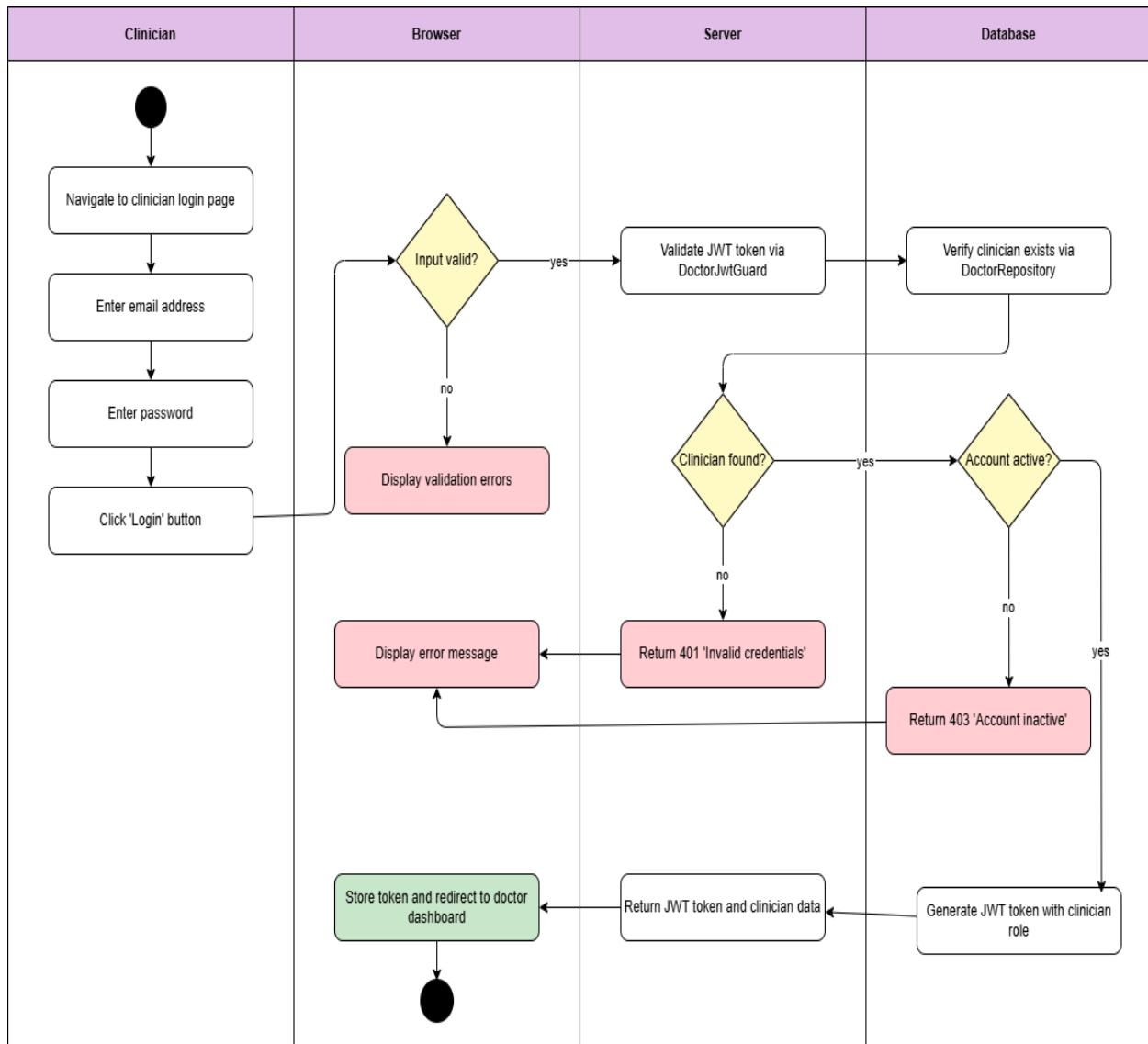


Diagram 22, Activity Diagram (Clinician Login)

Chapter 4 - System Analysis

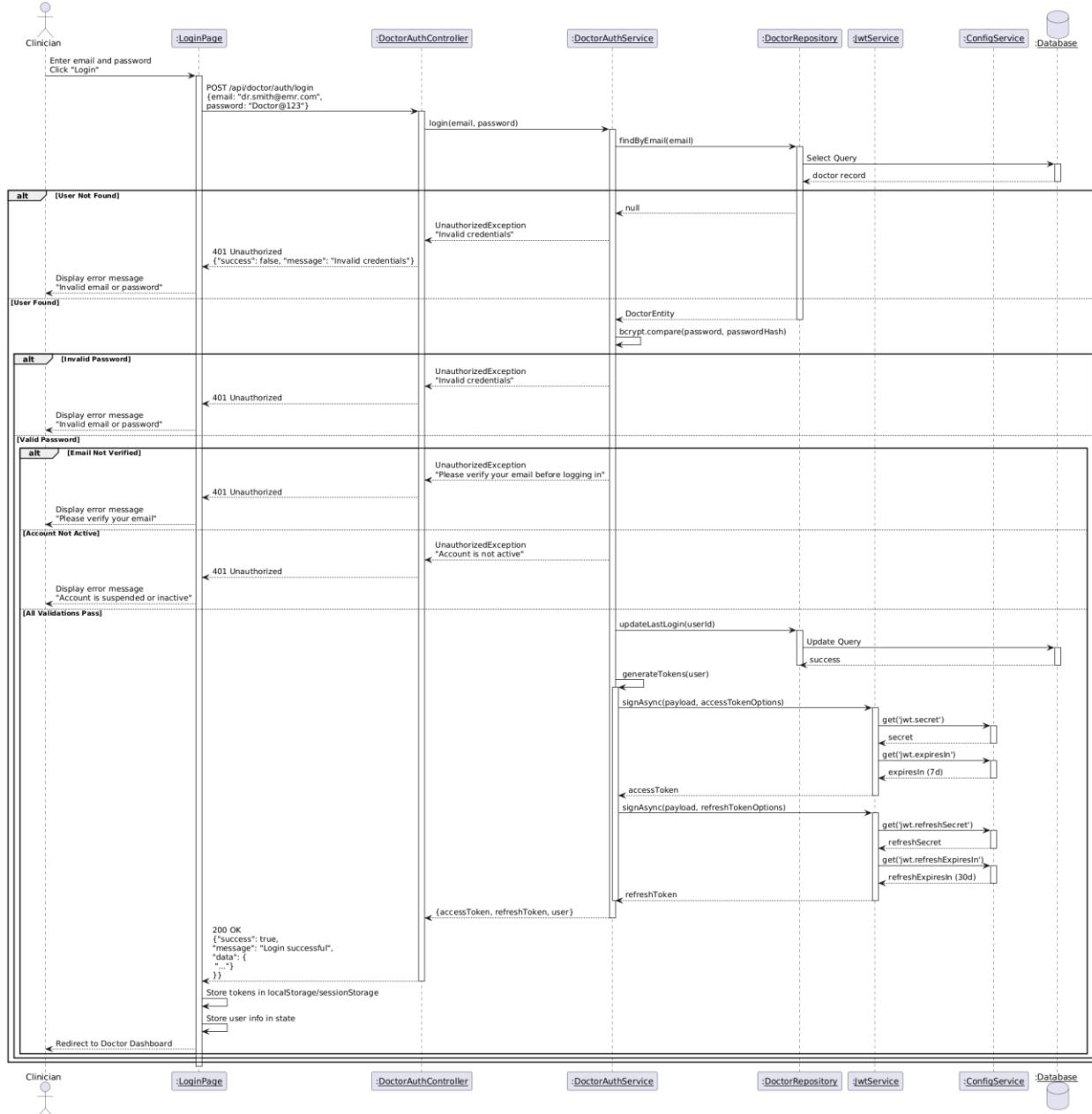


Diagram 23, Sequence Diagram (Clinician Login)

- **Create New Visit:**

Use case ID	VEMR-FR-VM-10
Use case name	Create New Visit
Description	The system allows clinicians to create a new visit
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.clinician click “new visit” button from visit page 2.System displays Encounter Form model 3.Clinician selects patient and fills required field 4.Clinican optionally enters reason for visit, chief complaint, location, and notes (support voice input). 5.Clinician submits the form 6.System validates input, authenticates user, and verifies patient exists. 7.System creates encounter with status “planned” 8.System Displays success message and closes modal</p>
Alternative scenario	<p>A1:Validation Error</p> <ul style="list-style-type: none"> - At step 6, if required fields are missing or patient not found -System displays error message on form <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 7, if JWT token is invalid or expired -System returns existing encounter without creating duplicate
Post condition	New visit is created with status “planned”

Table 17, Use Case Specification (Create New Visit)

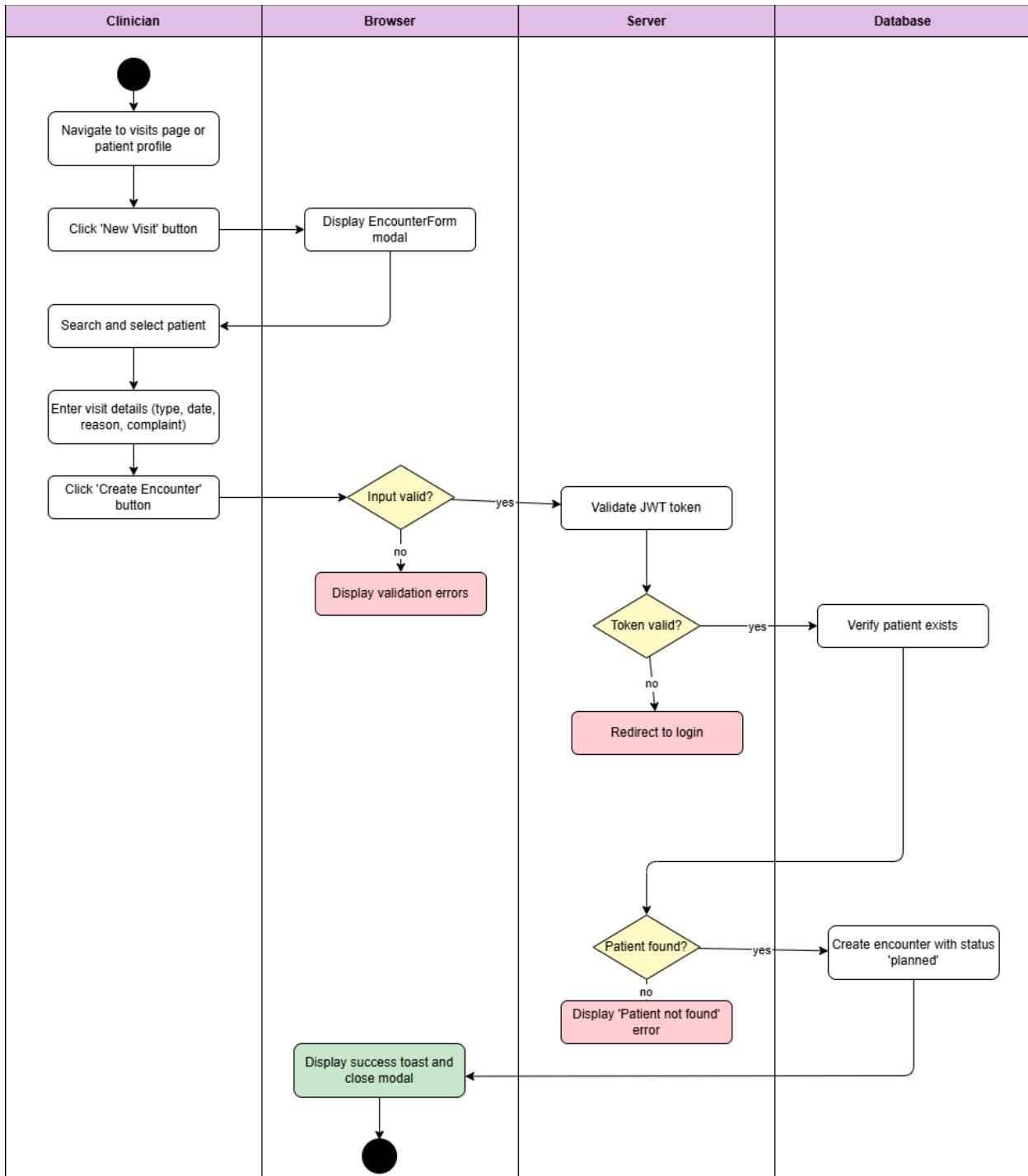


Diagram 24, Activity Diagram (Create New Visit)

Chapter 4 - System Analysis

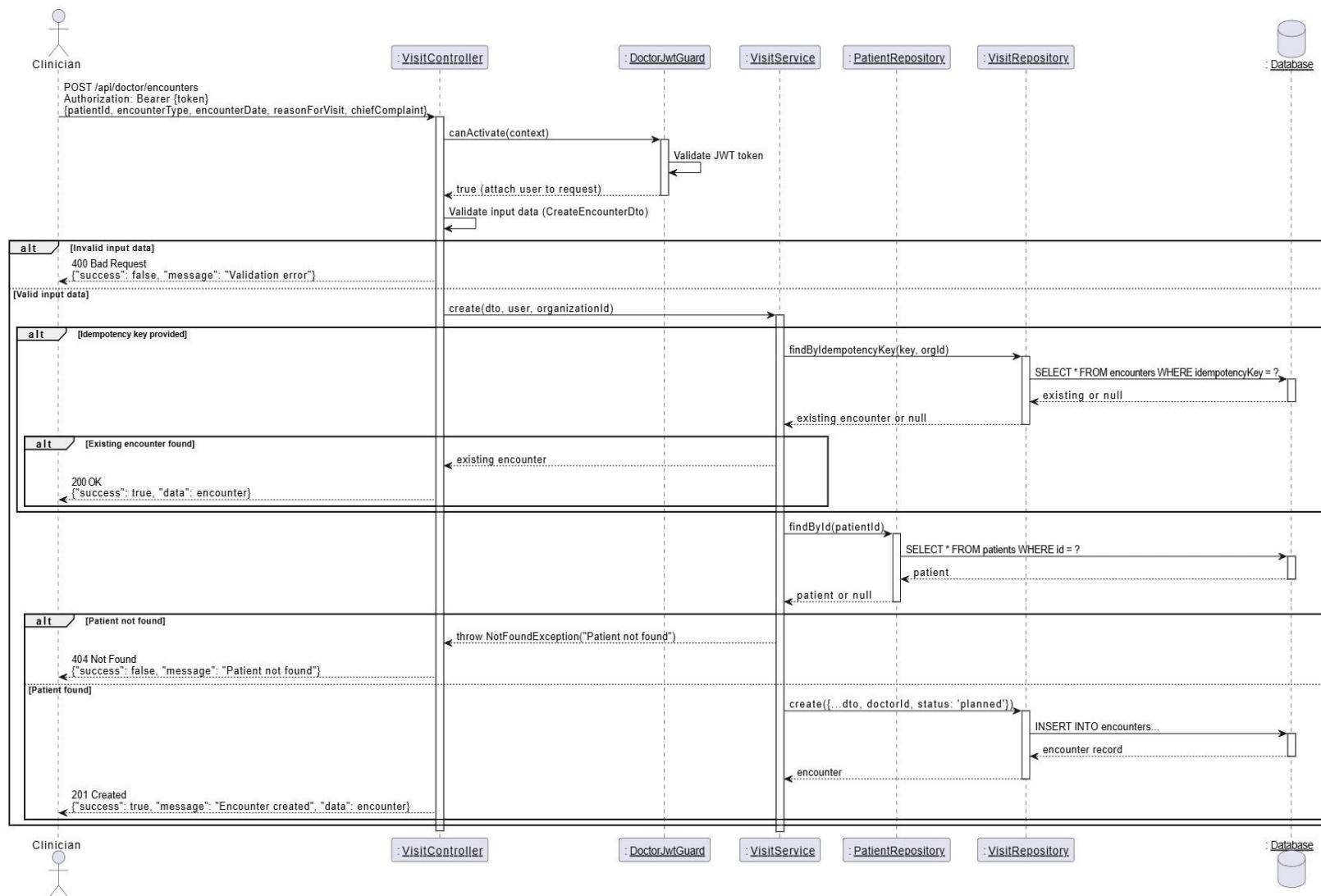


Diagram 25, Sequence Diagram (Create New Visit)

● Search Visit:

Use case ID	VEMR-FR-VM-11
Use case name	Search visit
Description	The system allows clinicians to search for visit
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician enters search query in search input field on visits page.</p> <p>2.System authenticated user and searches visits by patient name</p> <p>3.System returns paginated results with matching visits</p> <p>4.System displays search results in visits table</p>
Alternative scenario	<p>A1:No Result Found</p> <ul style="list-style-type: none"> - At step 3, if no visit matches the query -System displays “No visits found” message <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 7, if JWT token is invalid or expired -System redirects to login page
Post condition	Matching visits are displayed in the table

Table 18, Use Case Specification (Search Visit)

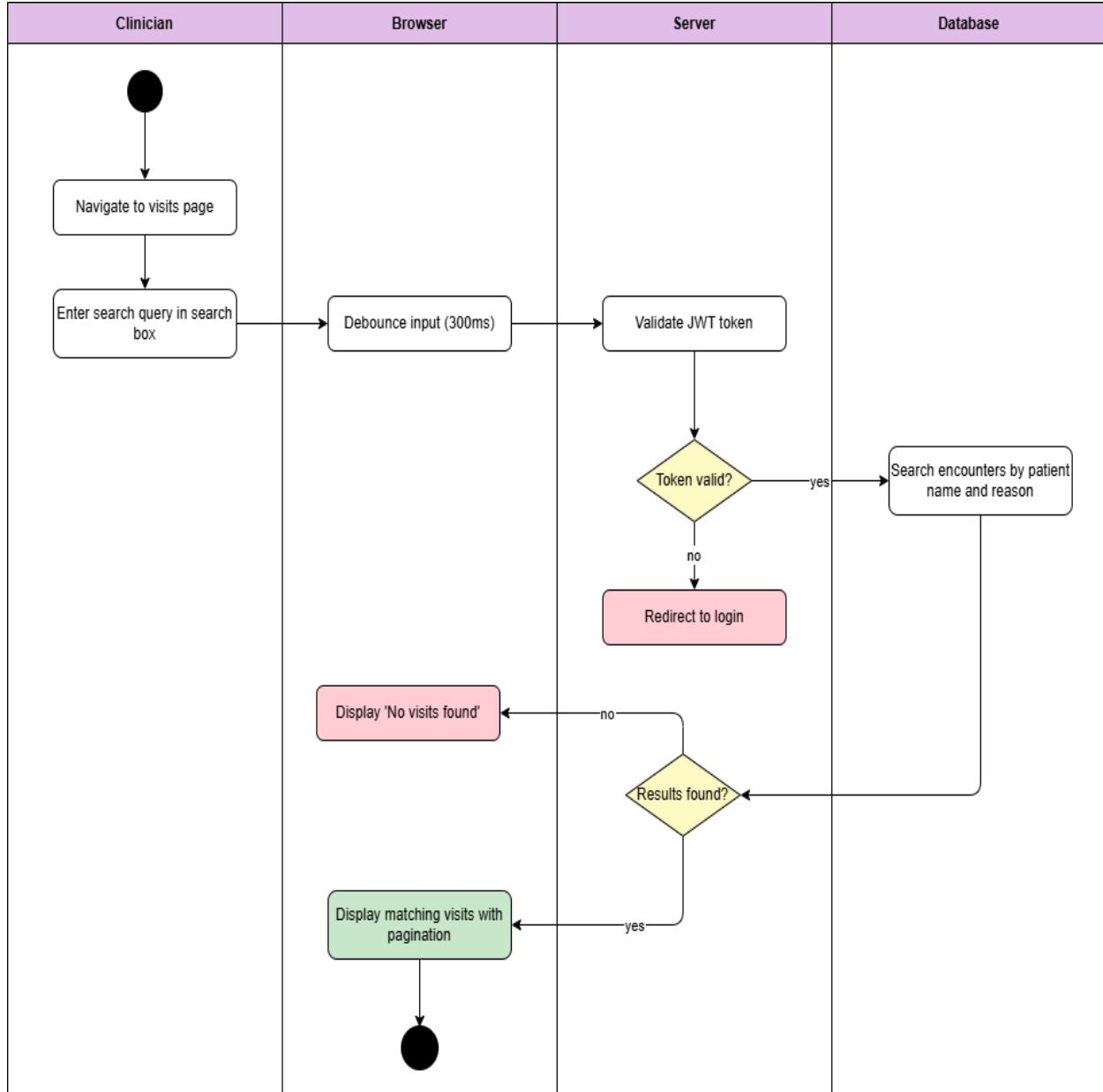


Diagram 26, Activity Diagram (Search Visit)

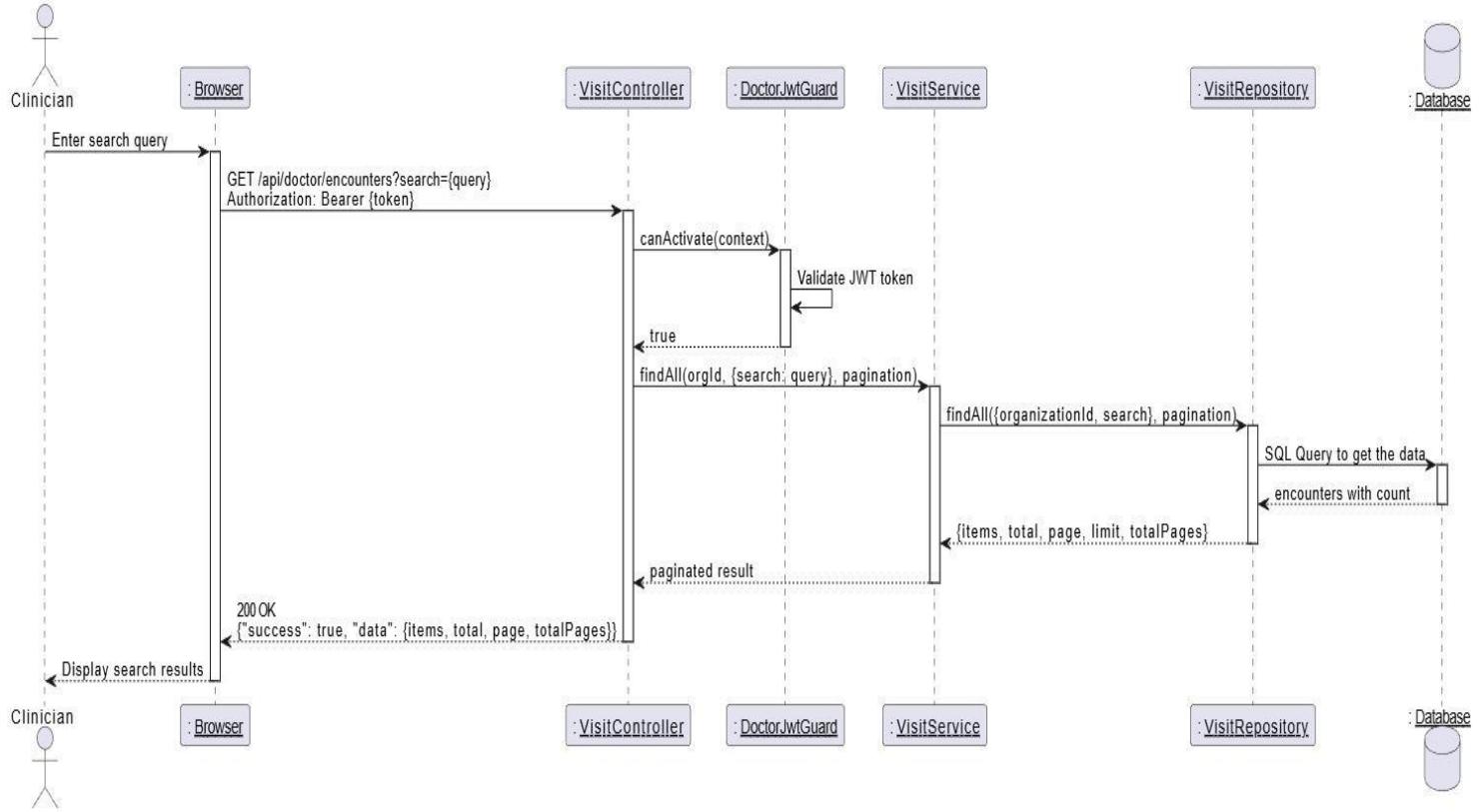


Diagram 27, Sequence Diagram (Search Visit)

- **Edit Visit:**

Use case ID	VEMR-FR-VM-12
Use case name	Edit visit
Description	The system allows clinicians to edit visit details.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician open visit detail page and clicks edit button</p> <p>2.System updates visit information (type , reason , chief , complaint , notes , location)</p> <p>3.Clinician submits changes</p> <p>4.System validates input , authenticates user , and verifies visit is in progress</p> <p>5.System verifies clinician is the assigned doctor</p> <p>6.System updates encounter in database</p> <p>7.System displays success message</p>
Alternative scenario	<p>A1: Visit Not Editable</p> <ul style="list-style-type: none"> - At step 4, if visit is not “in progress” or does not exist -System displays error message <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired -System redirects to login page
Post condition	Visit is updated with new information

Table 19, Use Case Specification (Edit Visit)

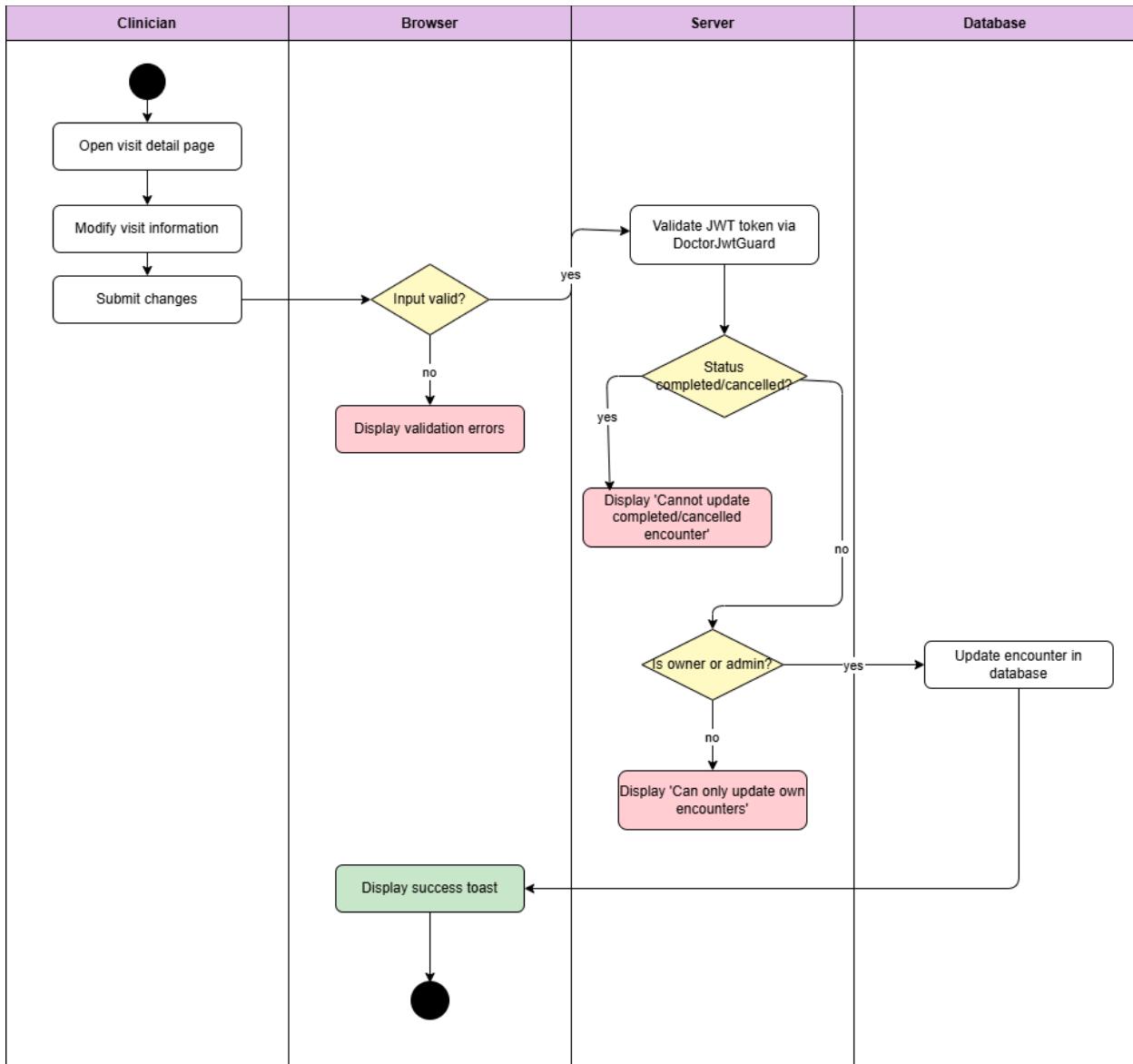


Diagram 28, Activity Diagram (Edit Visit)

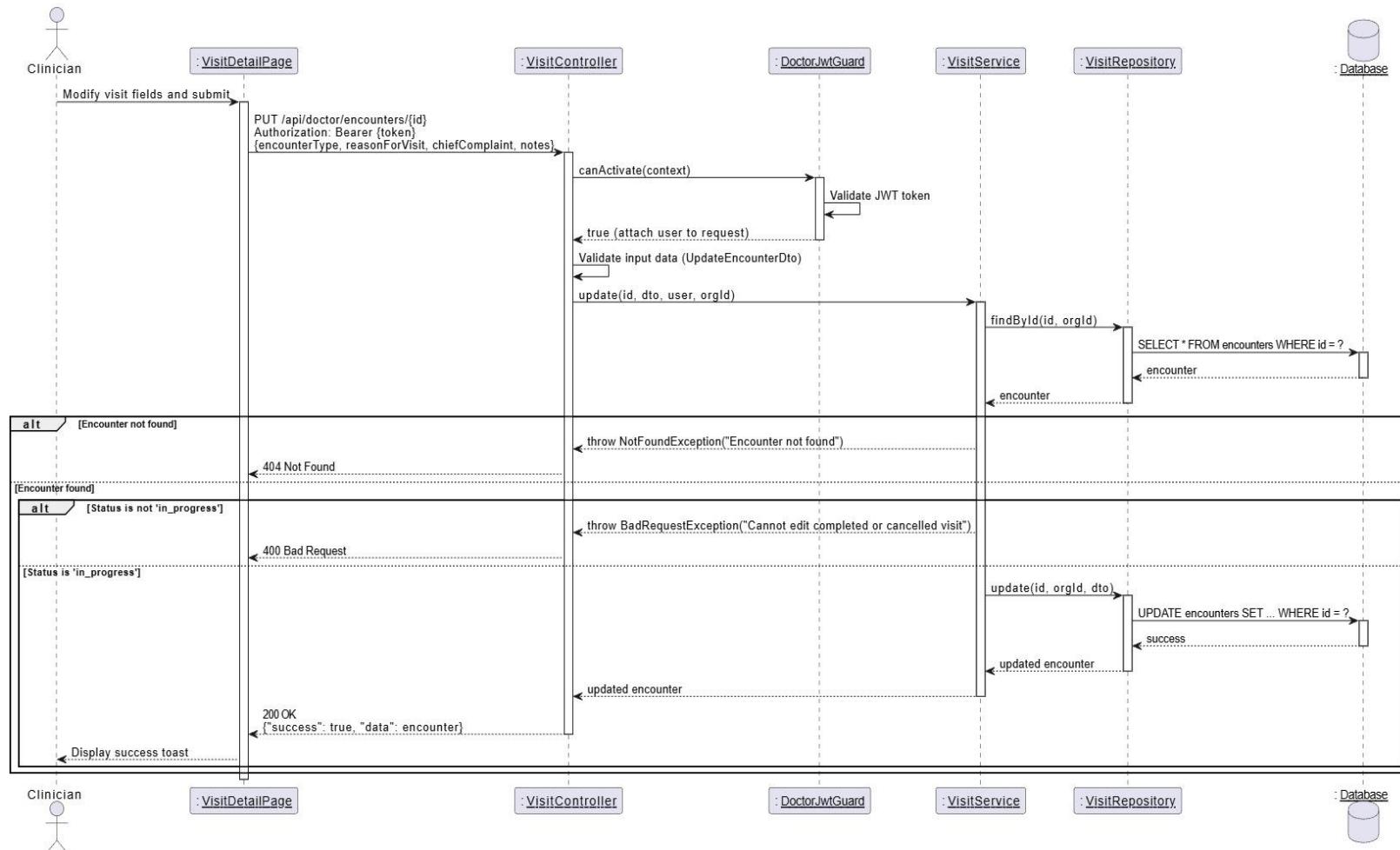


Diagram 29, Sequence Diagram (Edit Visit)

- **Save Visit:**

Use case ID	VEMR-FR-VM-13
Use case name	Save visit
Description	The system allows clinicians to save visit details.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician makes changes to visit or medical record fields 2.Clinician click save button 3.System validates input, authenticates user. 4.System verifies visit clinician is in progress and not finalized 5.System updates encounter/medical record in database 6.System displays success message.</p>
Alternative scenario	<p>A1: Validation or Status Error</p> <ul style="list-style-type: none"> - At step 3-4, if validation fails, visit not in progress , or record finalized -System redirects to login page <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired -System redirects to login page
Post condition	Visit / medical record changes are persisted

Table 20, Use Case Specification (Save Visit)

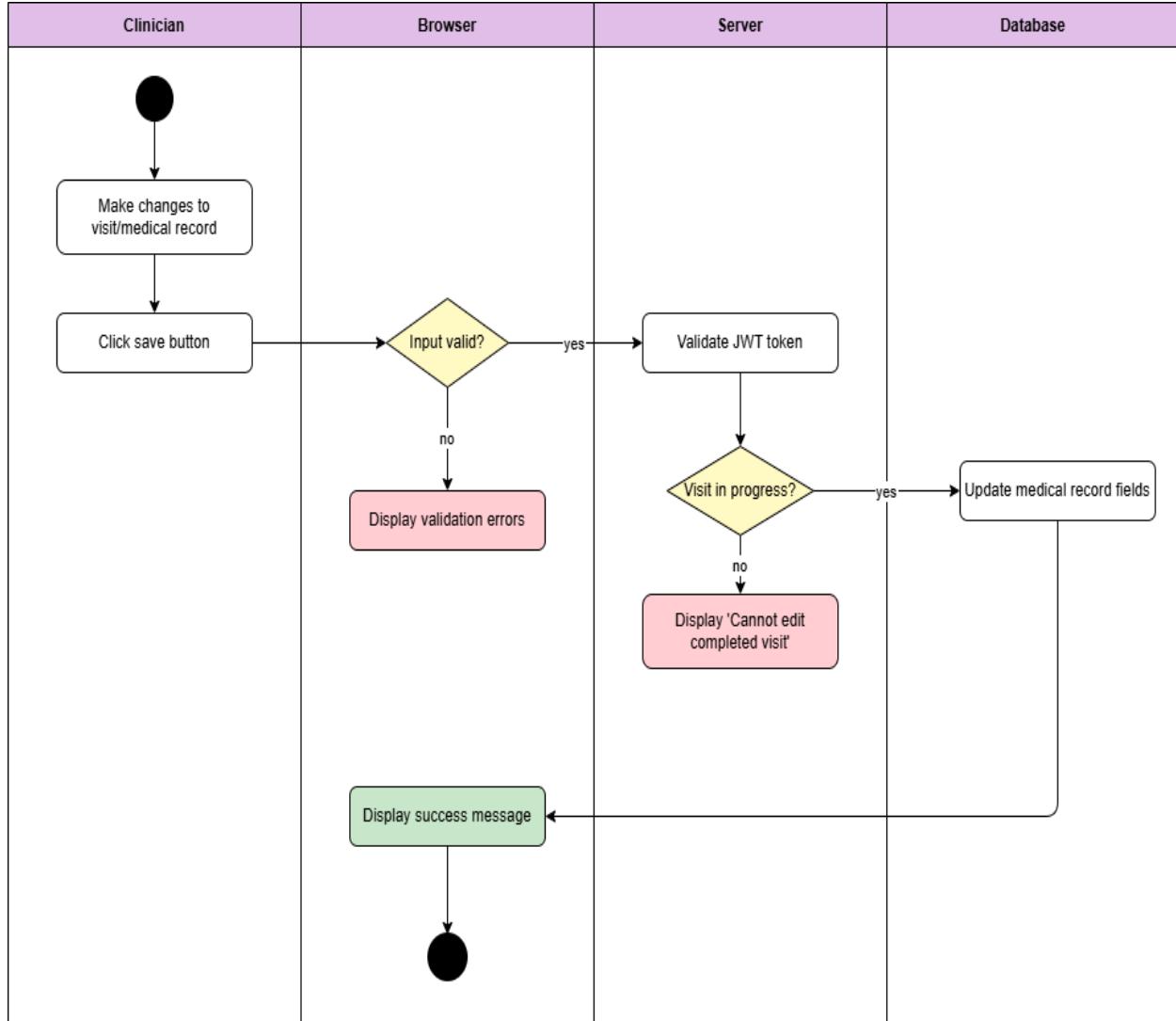


Diagram 30, Activity Diagram (Save Visit)

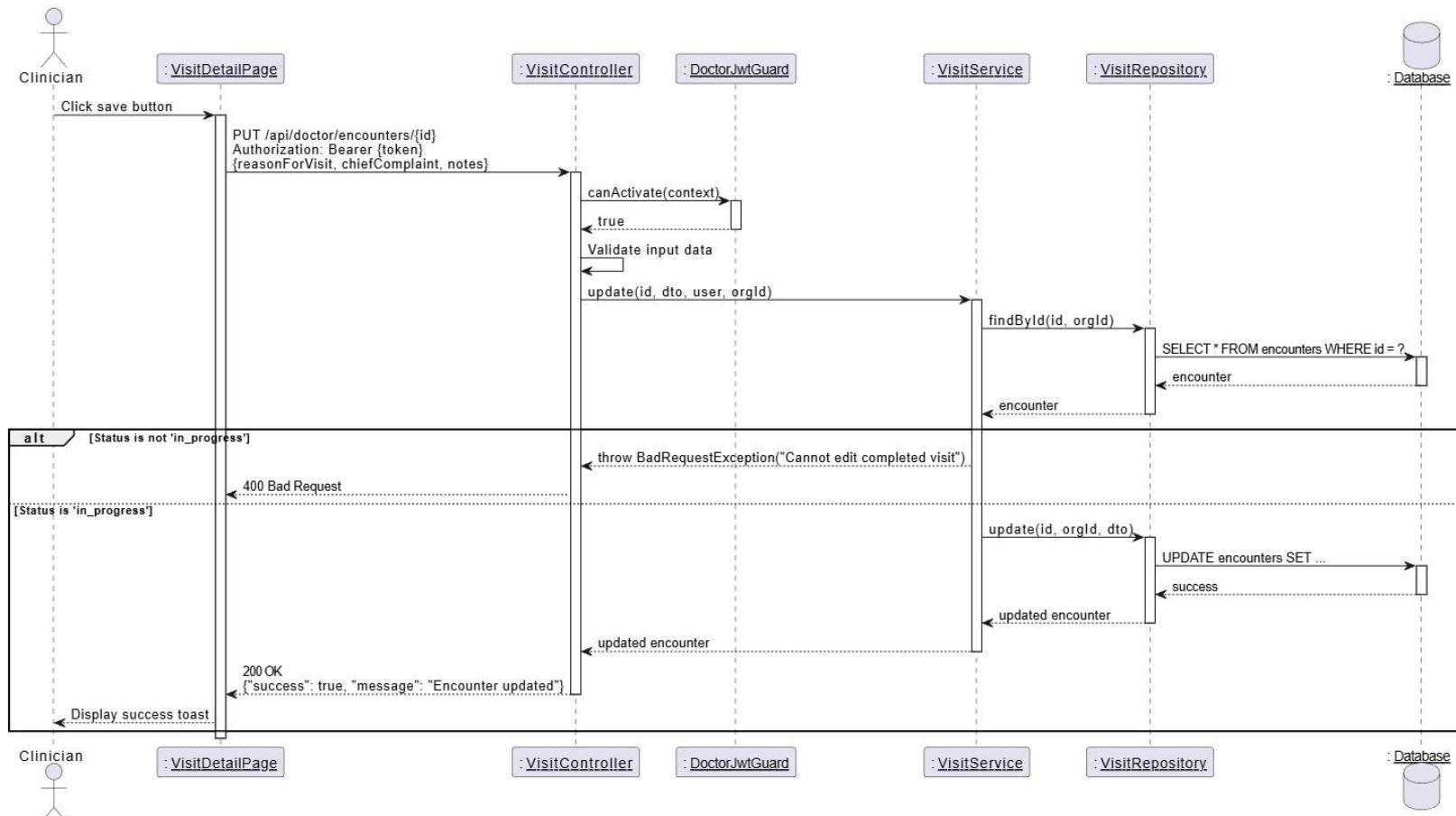


Diagram 31, Sequence Diagram (Save Visit)

- **Delete Visit:**

Use case ID	VEMR-FR-VM-14
Use case name	Delete Visit
Description	The system allows clinicians to delete a visit.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician selects visit to delete from visits list or details page</p> <p>2.System prompts for confirmation dialog</p> <p>3.Clinician confirms deletion</p> <p>4.System authenticates user and verifies clinician is the assigned doctor</p> <p>5.System soft-deletes the encounter (sets deleted at timestamp)</p> <p>6.System displays success message and removes visit from list</p>
Alternative scenario	<p>A1:User Cancels</p> <ul style="list-style-type: none"> - At step 3, if clinician cancels confirmation -No action is taken <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired -System redirects to login page
Post condition	Visit is deleted.

Table 21, Use Case Specification (Delete Visit)

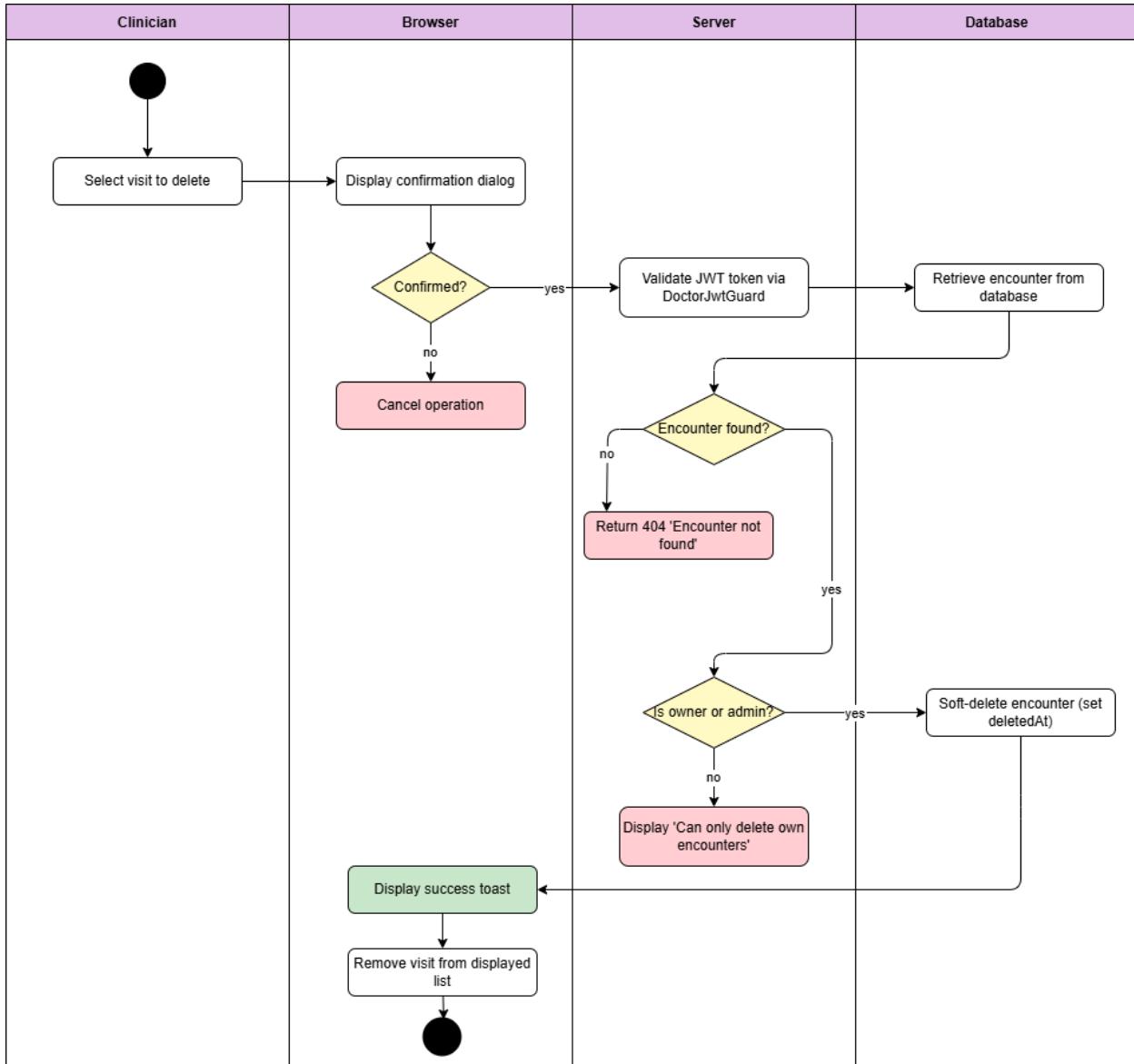


Diagram 32, Activity Diagram (Delete Visit)

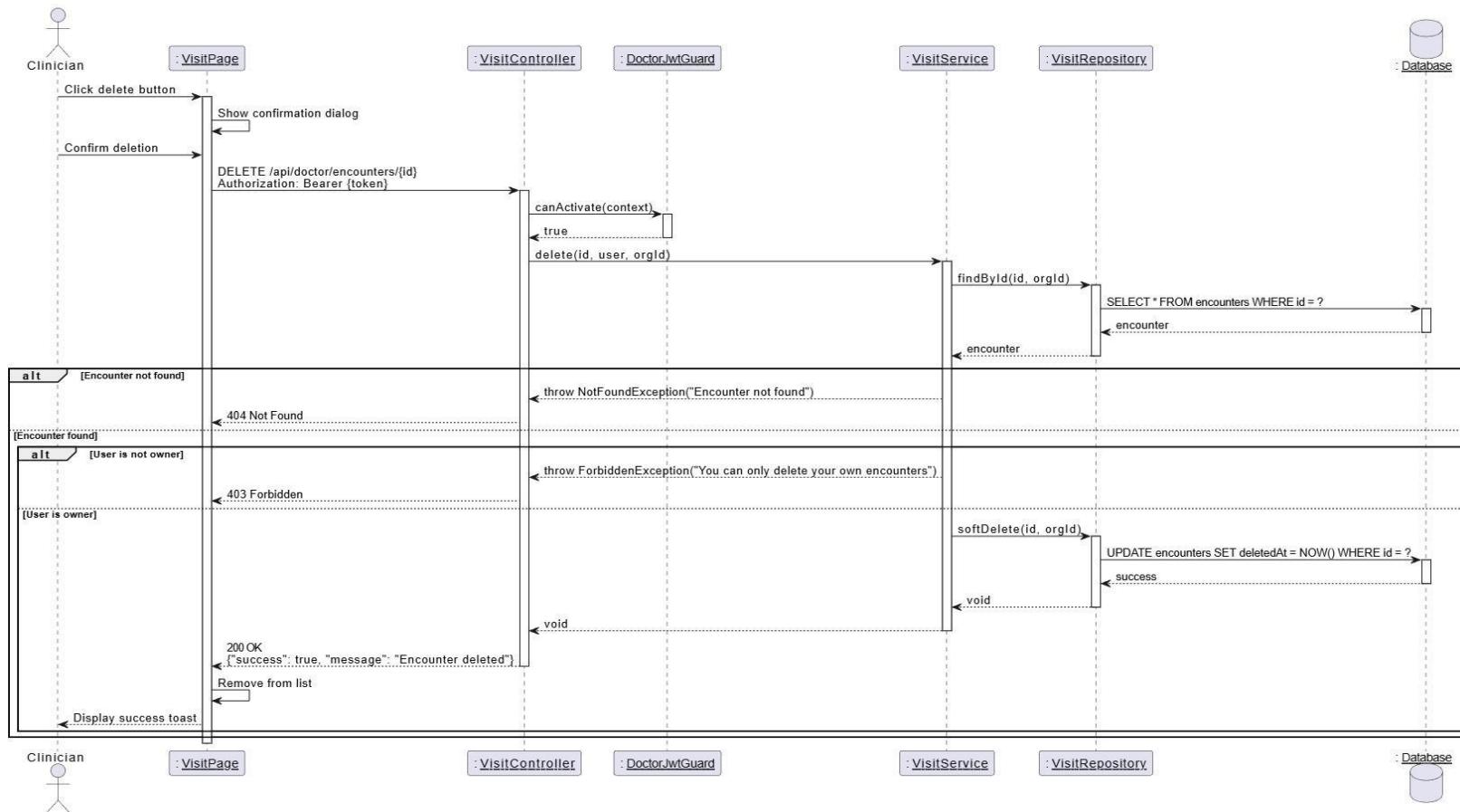


Diagram 33, Sequence Diagram (Delete Visit)

- **View Visit Details:**

Use case ID	VEMR-FR-VM-15
Use case name	View visit details
Description	The system allows clinicians to view complete details of a visit
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician click on a visit form the visits list 2.System navigates to visit detail page 3.System authenticates user and retrieves encounter details 4.System retrieves associated medical record, vital signs, and patient allergies 5.System displays visit details page with tabs (details ,vitals, Allergies)</p>
Alternative scenario	<p>A1: Visit Not Found</p> <ul style="list-style-type: none"> - At step 3, if visit does not exist -System displays error page or redirects <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired -System redirects to login page
Post condition	Visit details are displayed with all associated data

Table 22, Use Case Specification (View Visit Details)

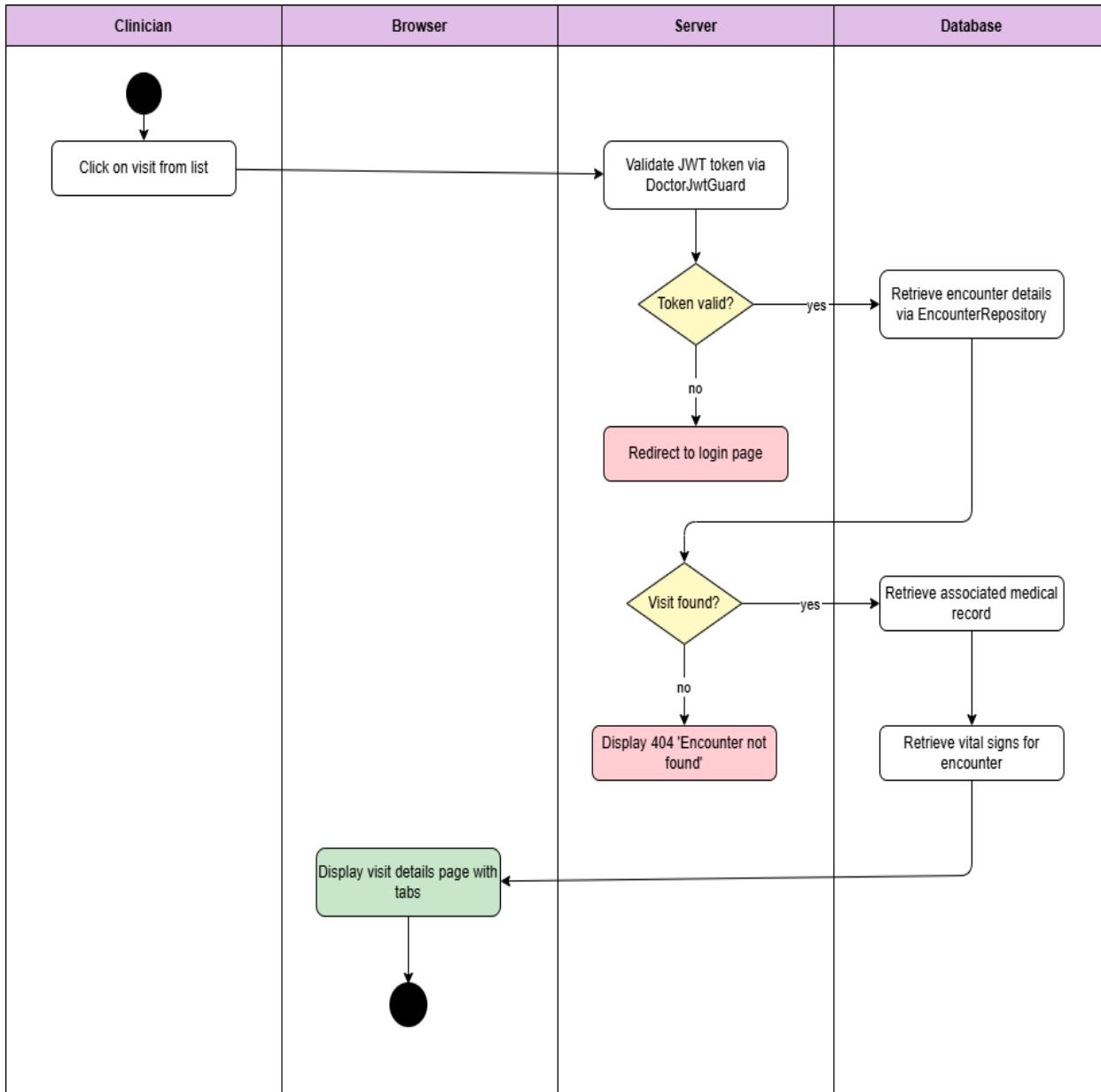


Diagram 34, Activity Diagram (View Visit Details)

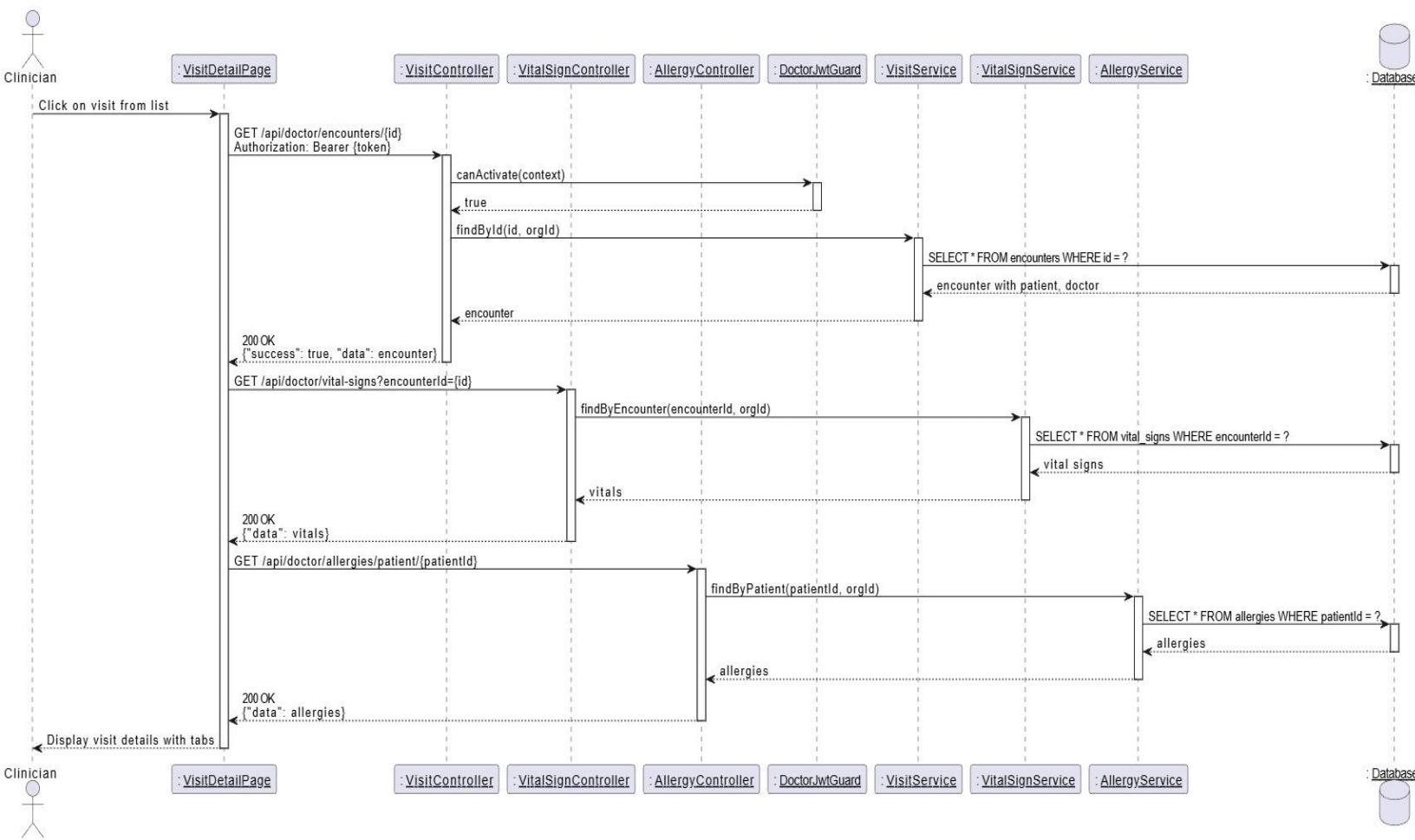


Diagram 35, Sequence Diagram (View Visit Details)

● Create Medical Record:

Use case ID	VEMR-FR-VM-16
Use case name	Create Medical Record
Description	The system allows clinicians to create a new medical record for a patient during an active visit.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to visit details page 2. Clinician clicks "Create Medical Record" button 3. System displays medical record form 4. Clinician enters medical information 5. Clinician optionally uses voice transcription (Whisper) for data entry 6. Clinician clicks "Save Draft" button 7. System validates JWT token 8. System validates required fields 9. System creates medical record 10. System sets status to "Draft" 11. System displays success message with created record
Alternative scenario	<p>A1: Medical Record Already Exists</p> <ul style="list-style-type: none"> - At step 3, if medical record already exists for this visit - System displays error message <p>A2: Voice Transcription Used</p> <ul style="list-style-type: none"> - At step 5, clinician clicks microphone icon - System activates Whisper transcription service - Clinician speaks medical notes - Clinician reviews and corrects if needed
Post condition	Medical record is created in database

Table 23, Use Case Specification (Create Medical Record)

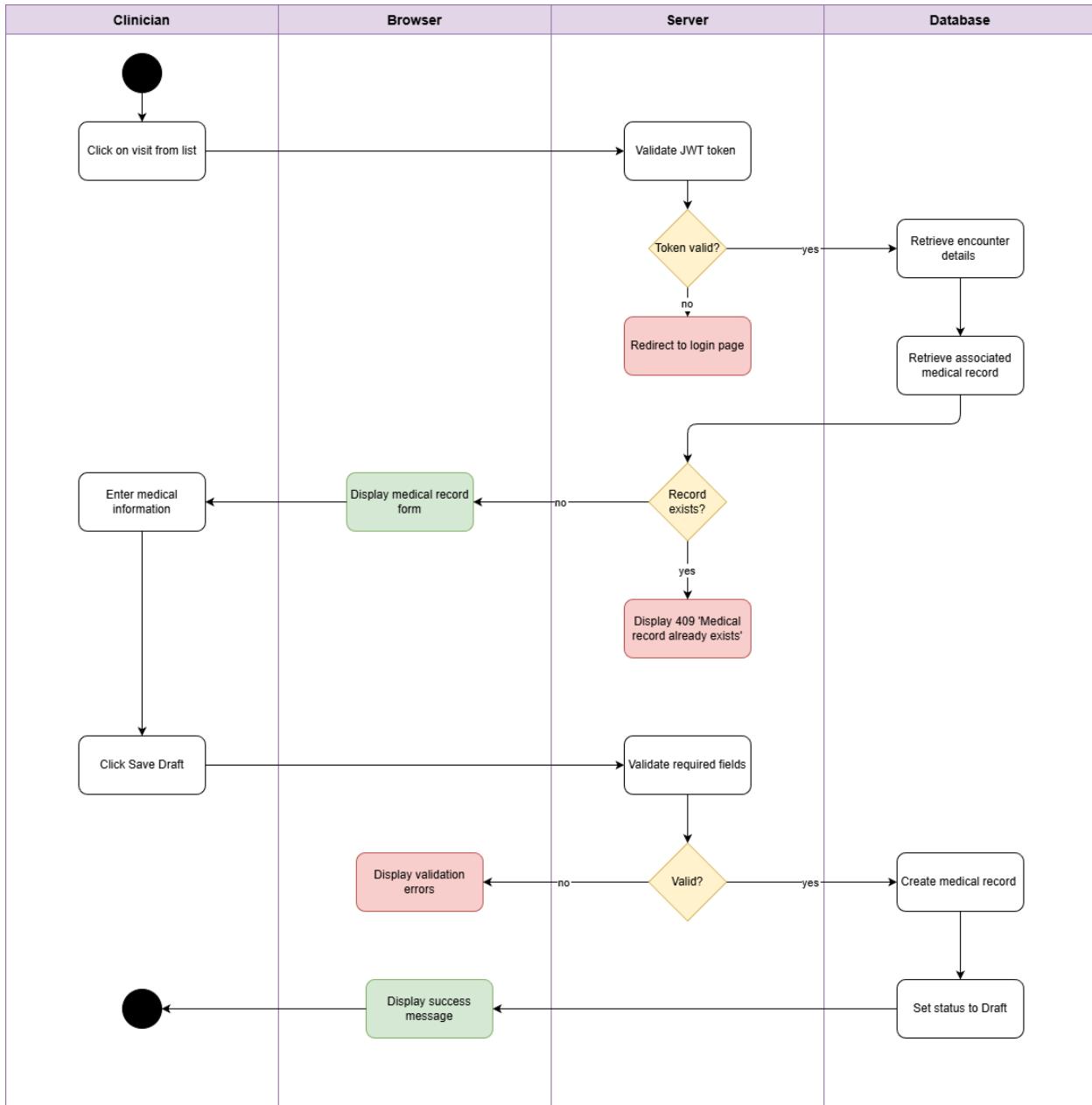


Diagram 36, Activity Diagram (Create Medical Record)

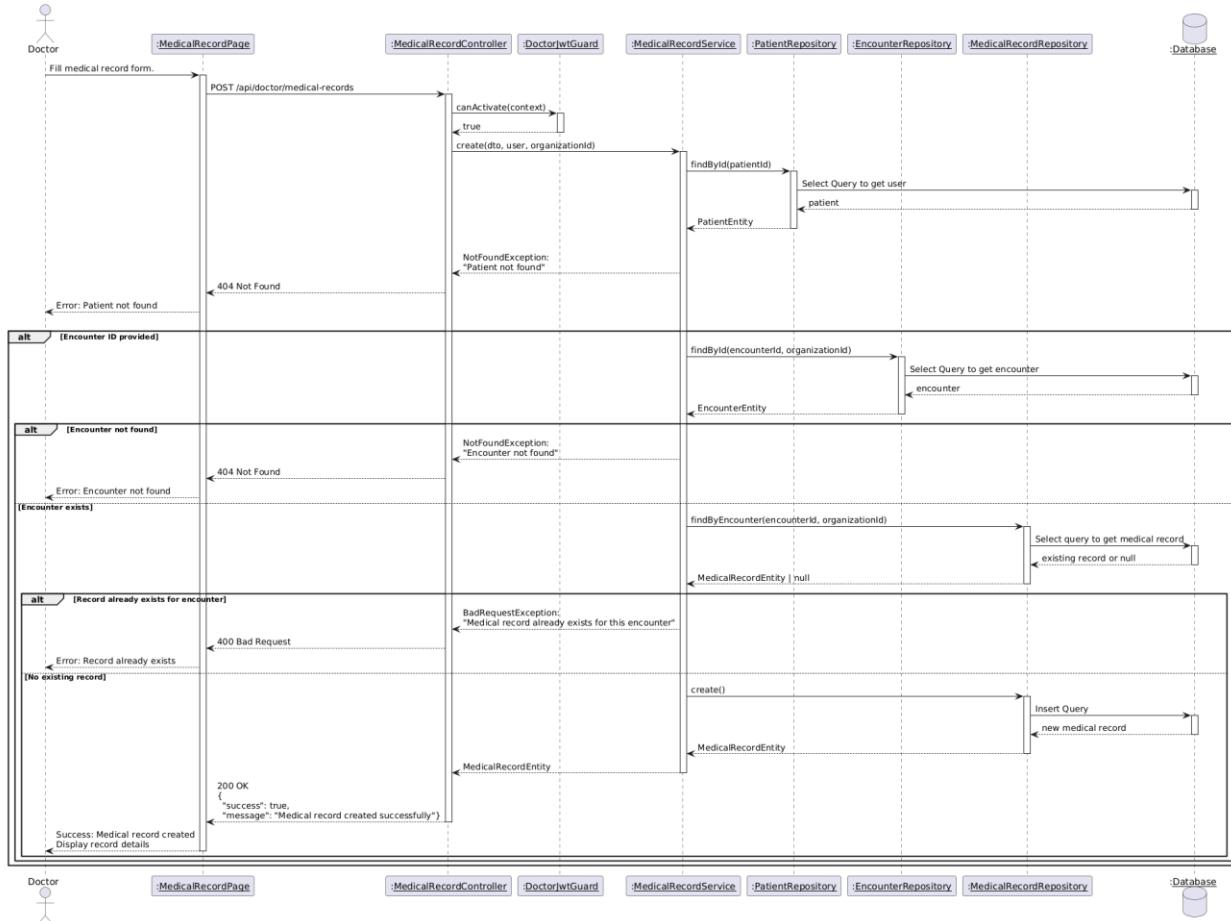


Diagram 37, Sequence Diagram (View Visit Details)

● Update Medical Record:

Use case ID	VEMR-FR-VM-17
Use case name	Update Medical Record
Description	The system allows clinicians to modify existing medical record information with audit trail tracking
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to visit details page 2. Clinician clicks "Edit Medical Record" button 3. System validates JWT token 4. System checks medical record status 5. System displays medical record in edit mode 6. Clinician modifies one or more sections 7. Clinician optionally uses voice transcription for updates 8. Clinician clicks "Save Changes" button 9. System validates modified fields 10. System creates audit log entry 11. System updates medical record 12. System displays success message
Alternative scenario	<p>A1: Record is Finalized</p> <ul style="list-style-type: none"> - At step 4, if medical record status is "Finalized" - System displays error message <p>A2: Voice Transcription Used</p> <ul style="list-style-type: none"> - At step 7, clinician clicks microphone icon - System activates Whisper transcription service - Clinician speaks updates - System appends transcribed text
Post condition	Medical record is updated in database

Table 24, Use Case Specification (Update Medical Record)

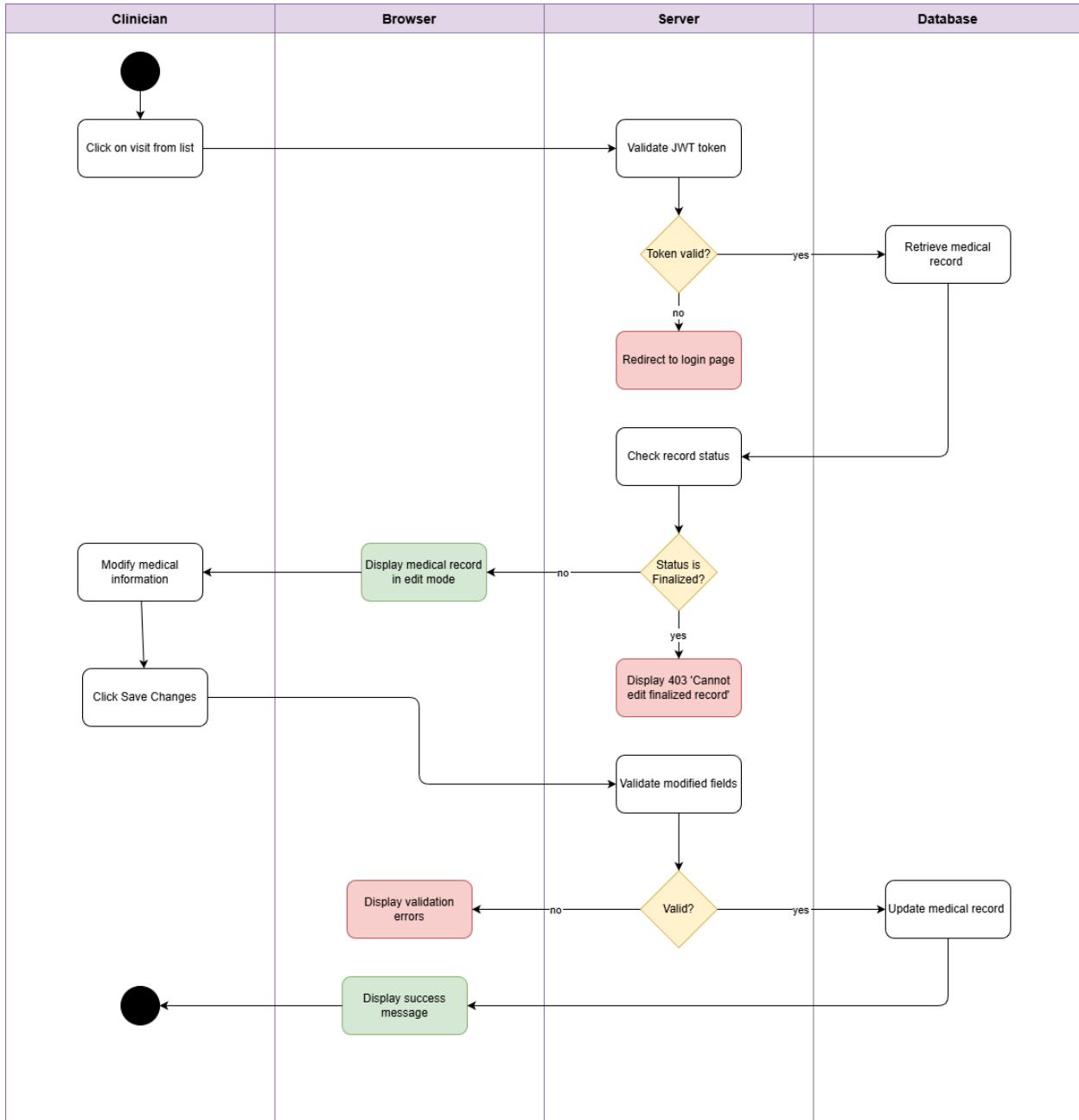


Diagram 38, Activity Diagram (Update Medical Record)

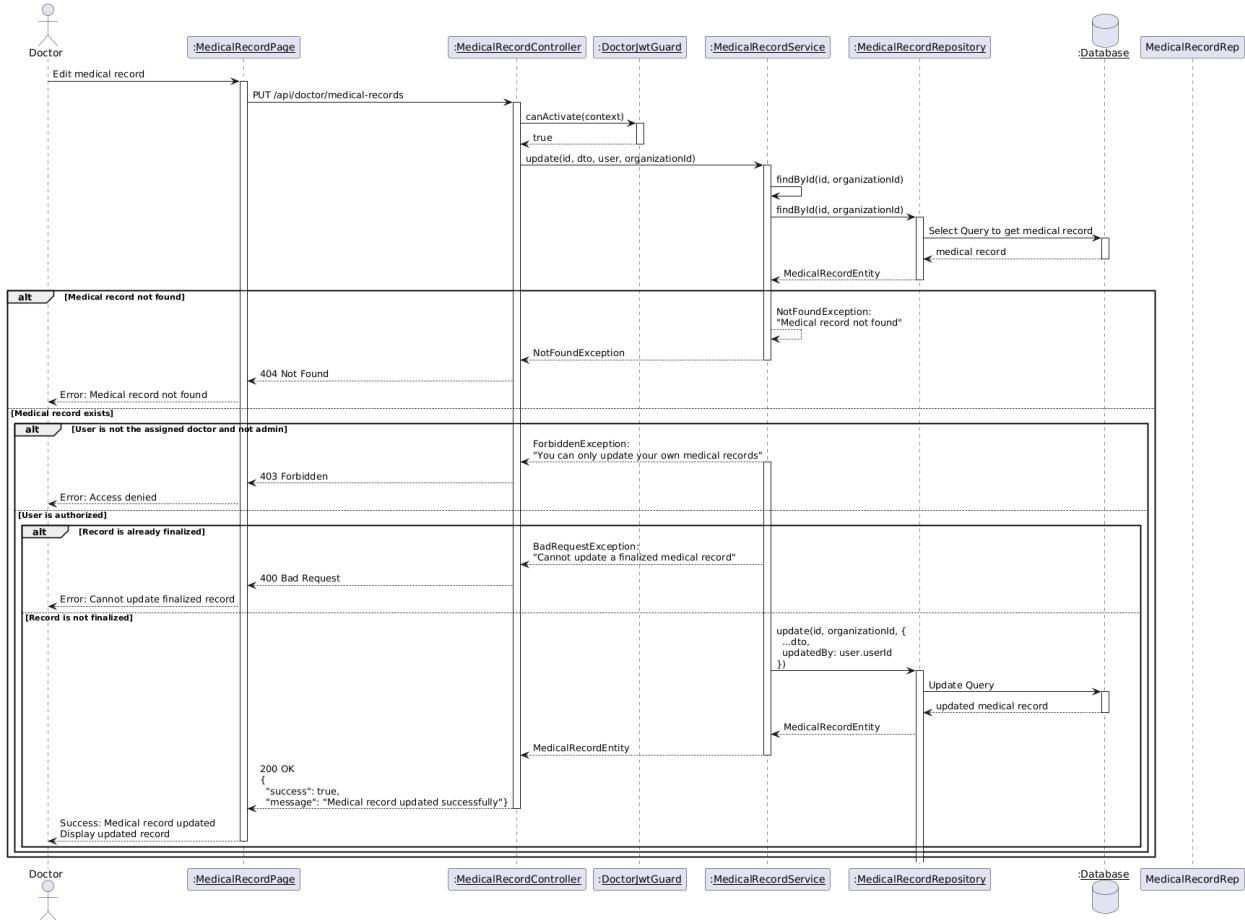


Diagram 39, Sequence Diagram (View Visit Details)

● Finalize Medical Record:

Use case ID	VEMR-FR-VM-18
Use case name	Finalize Medical Record
Description	The system allows clinicians to finalize and lock a medical record, making it read-only and available to patients.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to visit details page 2. Clinician reviews medical record for completeness 3. Clinician clicks "Finalize Medical Record" button 4. System validates JWT token 5. System validates all required fields are completed 6. System displays confirmation dialog with summary 7. Clinician confirms finalization 8. System validates clinician credentials (re-authentication) 9. System changes medical record status to "Finalized" 10. System adds digital signature with timestamp 11. System changes visit status to "Completed" 12. System sends notification to patient 13. System displays success message
Alternative scenario	<p>A1: : Incomplete Required Fields</p> <ul style="list-style-type: none"> - At step 5, if required fields are missing <ul style="list-style-type: none"> - system displays error message listing missing fields <p>A2: Clinician Cancels Finalization</p> <ul style="list-style-type: none"> - At step 7, clinician clicks microphone icon <ul style="list-style-type: none"> - Medical record remains in "Draft" status - System closes confirmation dialog
Post condition	Medical record status is "Finalized"

Table 25, Use Case Specification (Finalize Medical Record)

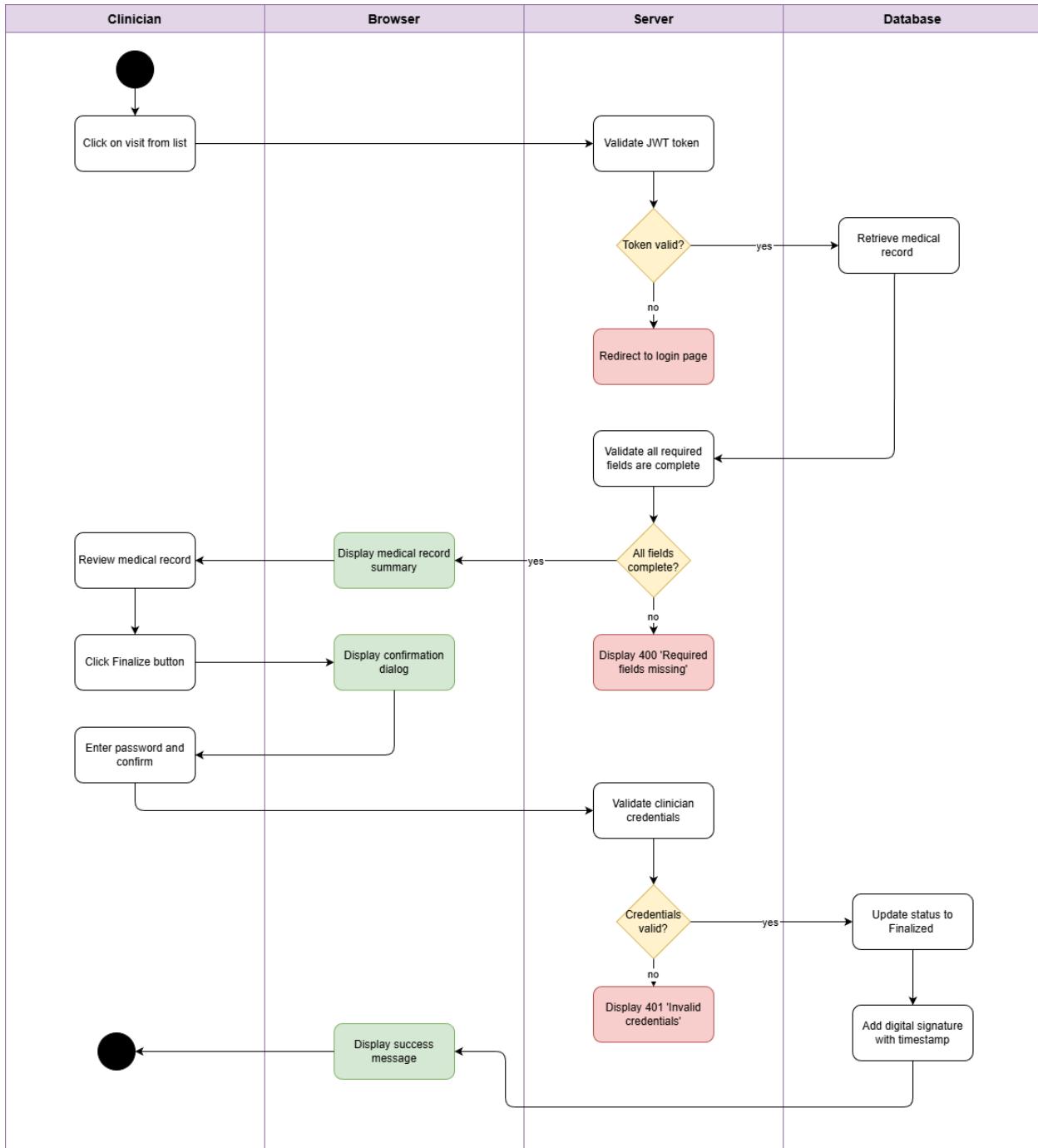


Diagram 40, Activity Diagram (Finalize Medical Record)

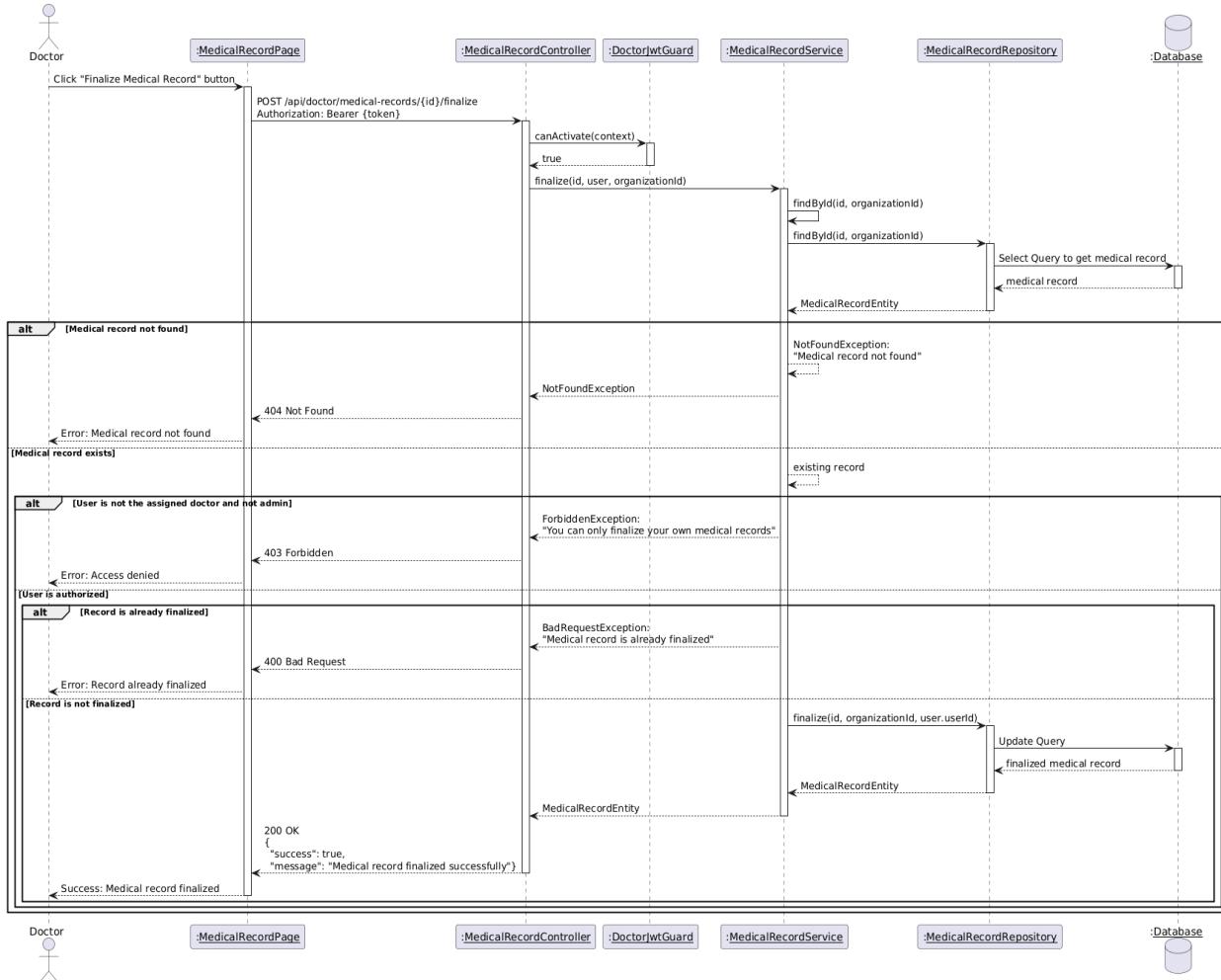


Diagram 41, Sequence Diagram (View Visit Details)

● Real-time Transcription Display:

Use case ID	VEMR-FR-VM-19
Use case name	Real-time Transcription Display
Description	Voice transcription appears in real-time as the clinician speaks using whisper
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinicians clicks microphone button on form field 2.Browser requests and user grants microphone permission 3.System establishes WebSocket connection to whisper server and starts recording 4.System sends audio chunks every 3 seconds to server 5.Whisper server transcribes audio and send partial text back 6.System displays transcription in real-time in the form fields 7.Clinician click stop button 8.System sends final audio chunk and receives final transcription 9.System closes connection and stop recording</p>
Alternative scenario	<p>A1: Microphone Access Denied</p> <ul style="list-style-type: none"> - At step 2, if user denies permission -System displays error and recording remains inactive <p>A2: Server Connection Failed</p> <ul style="list-style-type: none"> - At step 3, if WebSocket connection fails or server not running -System display error message
Post condition	<ul style="list-style-type: none"> - Transcribed text is added to the form field - Audio recording is stopped - WebSocket connection is closed

Table 26, Use Case Specification (Real-time Transcription Display)

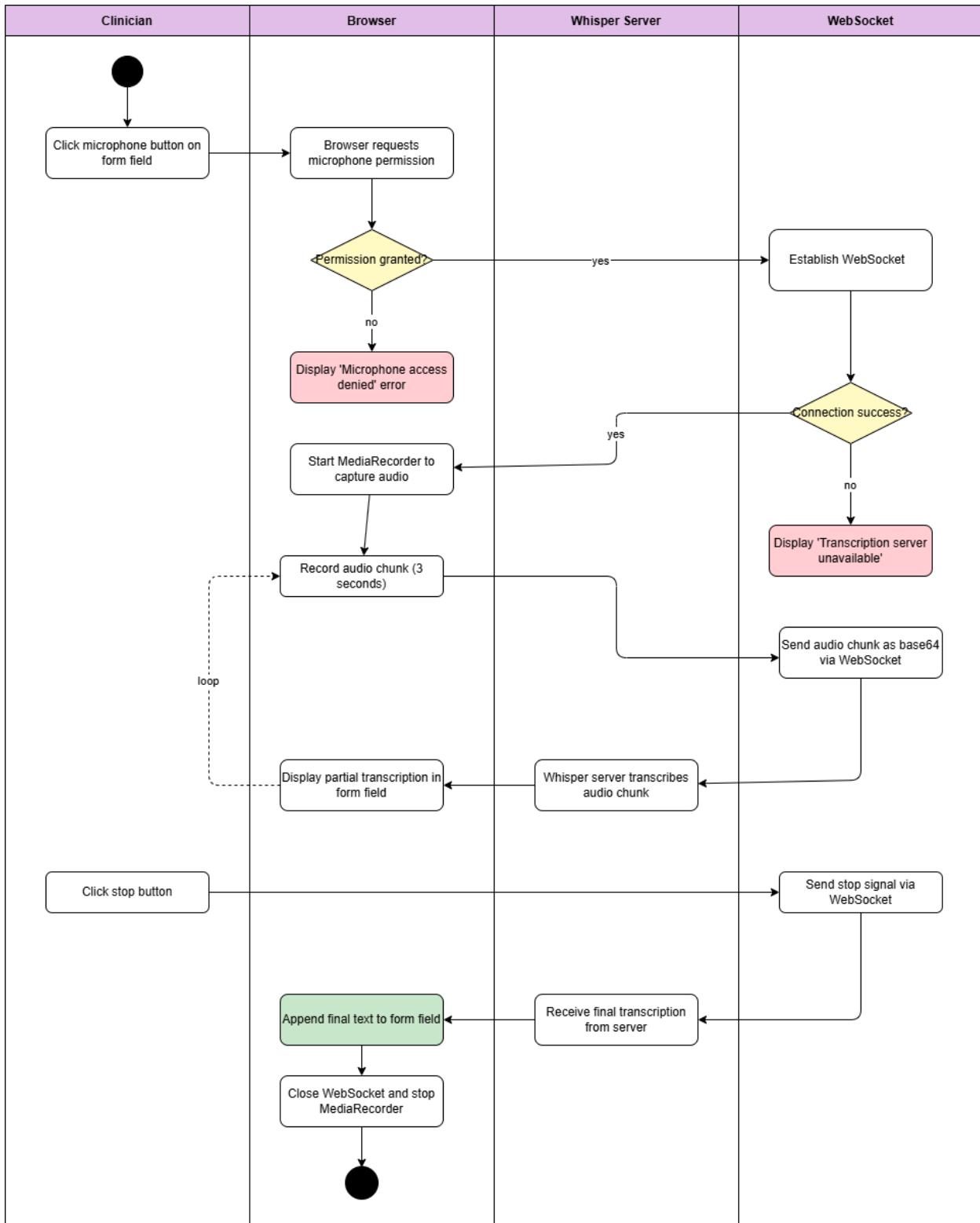


Diagram 42, Activity Diagram (Real-time Transcription Display)

Chapter 4 - System Analysis

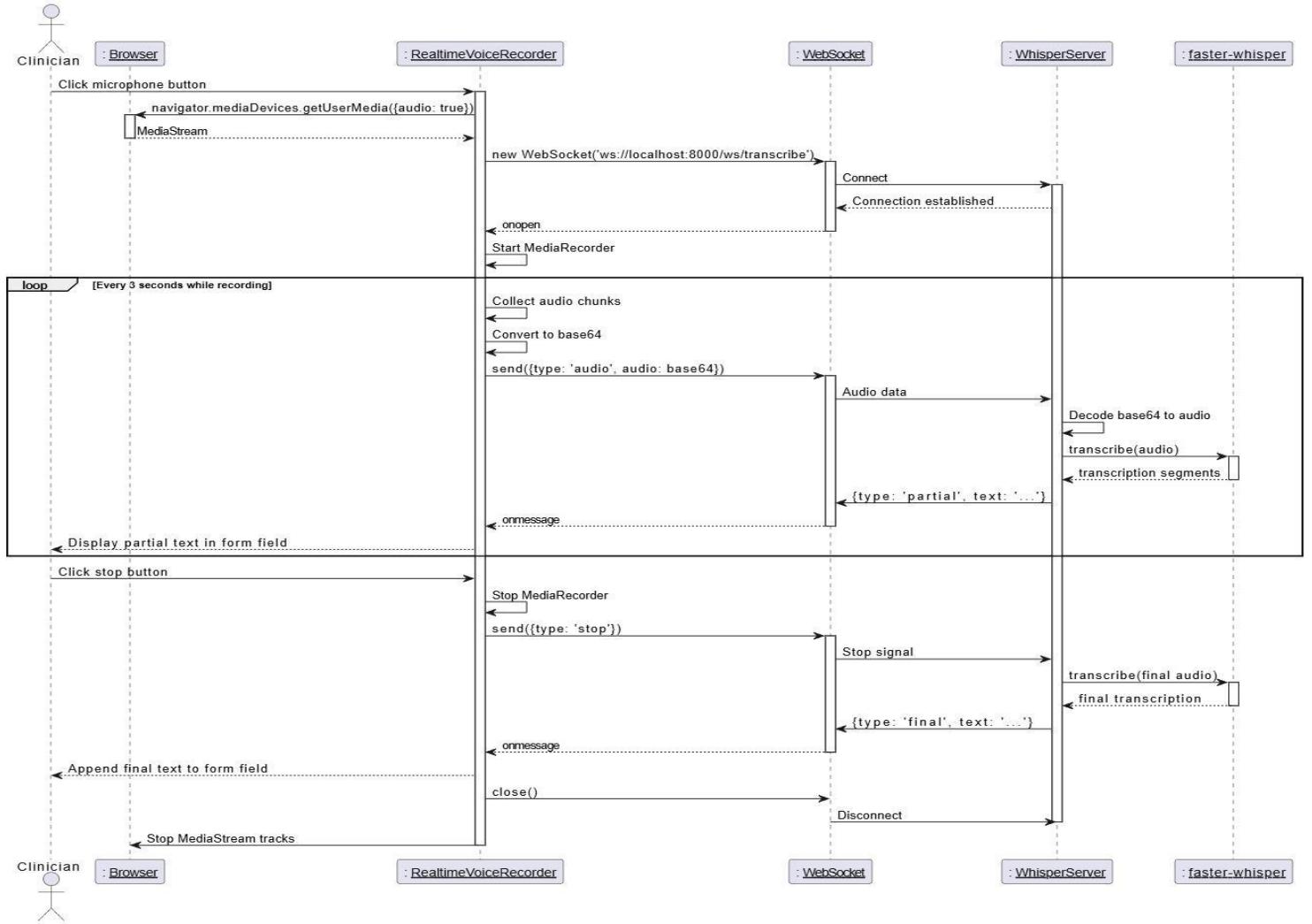


Diagram 43, Sequence Diagram (Real-time Transcription Display)

- **View Patient List:**

Use case ID	VEMR-FR-VM-20
Use case name	View patient list
Description	The system allows clinicians to view a paginated list of all patients
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician navigates to the patient's page.</p> <p>2.System authenticates user and retrieves paginated patient list</p> <p>3.System displays patients in a table with basic info (name, email, phone, date of birth).</p> <p>4.Clinician can navigate between pages using pagination controls</p>
Alternative scenario	<p>A1: No Patients</p> <ul style="list-style-type: none"> - At step 2, No patient exists -System displays “No patients found message” <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired -System return 401 Unauthorized
Post condition	Patient list is displayed with pagination

Table 27, Use Case Specification (View Patient List)

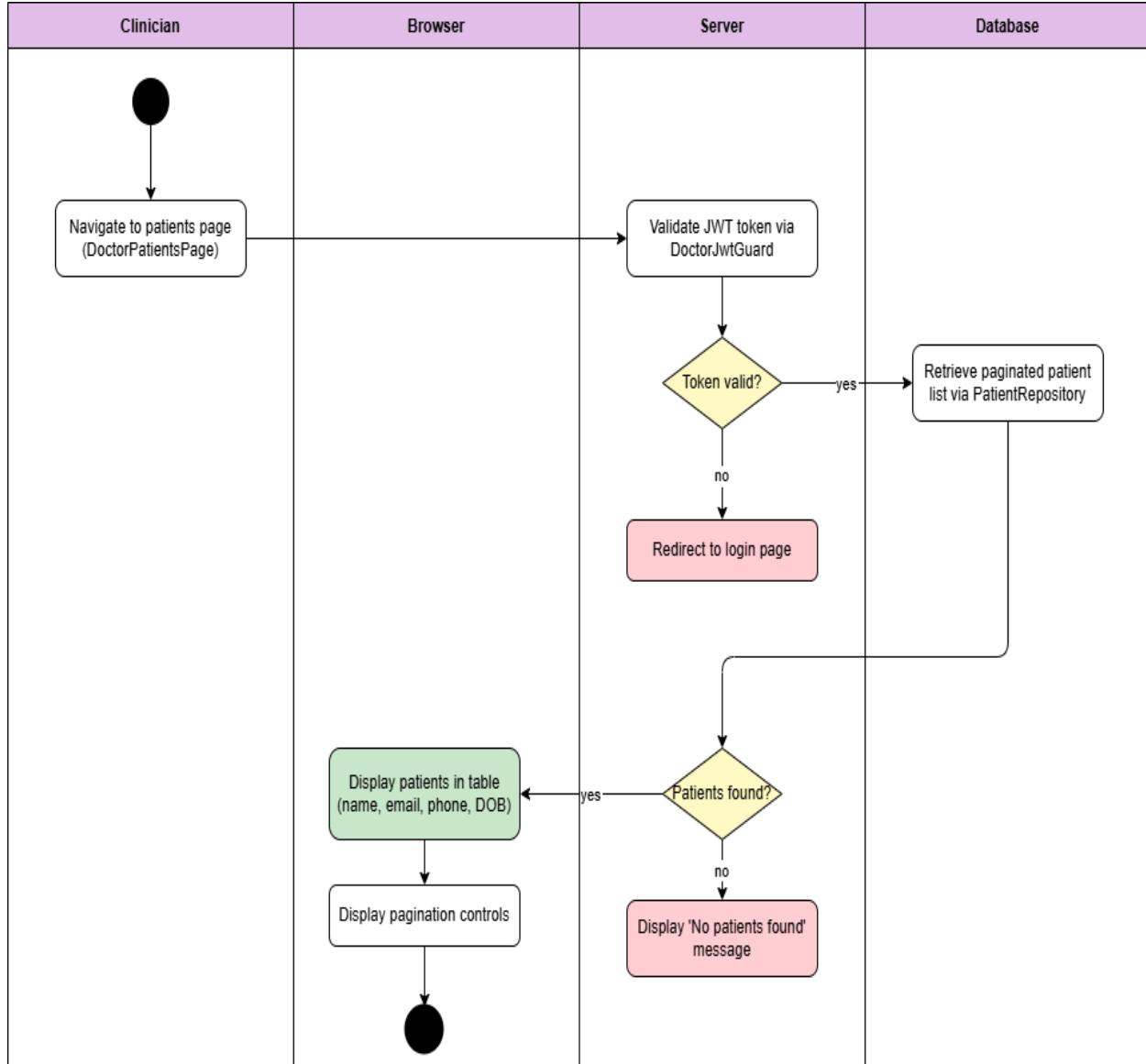


Diagram 44, Activity Diagram (View Patient List)

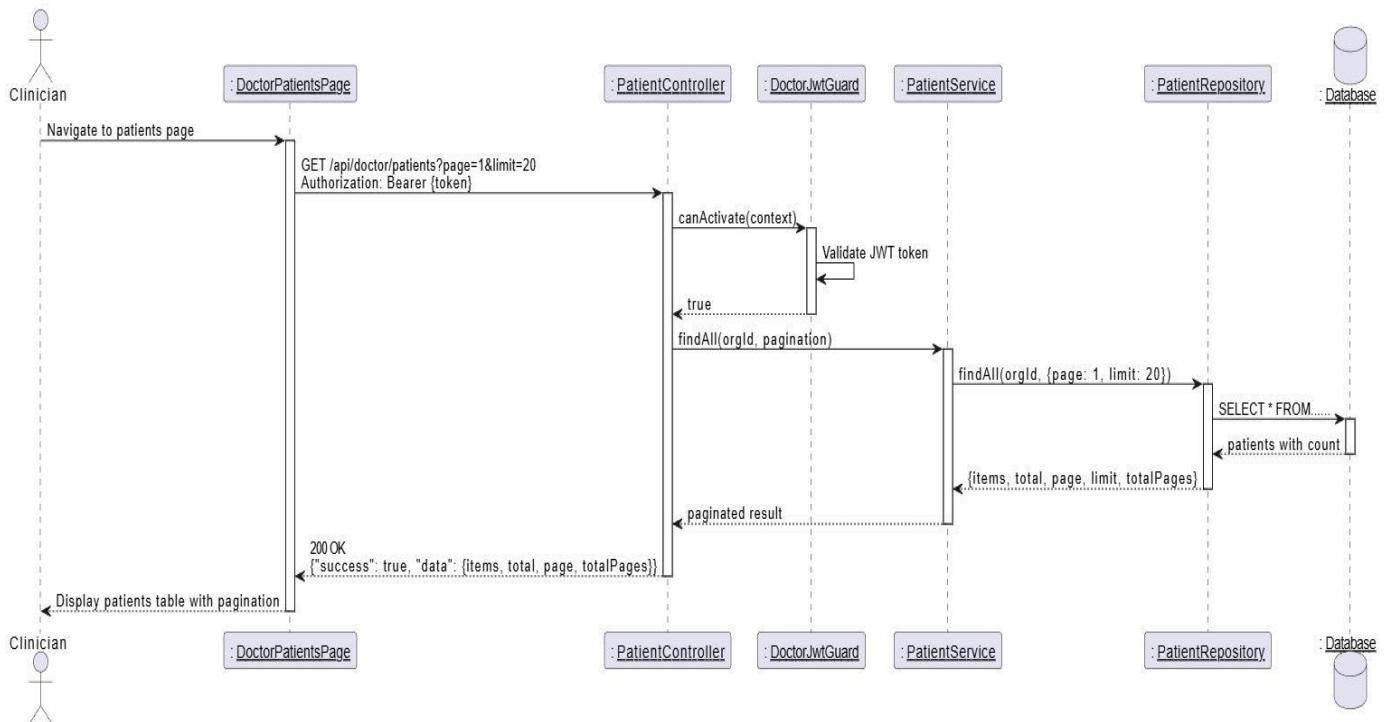


Diagram 45, Sequence Diagram (View Patient List)

- **Search Patient:**

Use case ID	VEMR-FR-VM-21
Use case name	Search patient
Description	The system allows clinicians to search for patient
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician enters search query in search input field 2.System authenticates user and searches patients by name 3.System returns paginated results with matching patients 4.System displays search results in patients table</p>
Alternative scenario	<p>A1: No Results Found</p> <ul style="list-style-type: none"> - At step 3, if no patients match the query <p>-System displays "No patients found "message</p> <p>A2: Empty Search Query</p> <ul style="list-style-type: none"> - At step 2, if search query is cleared <p>-System returns all patients with pagination</p> <p>A3: Authentication Error</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired <p>-System redirects to login page</p>
Post condition	Matching patient are displayed in the table

Table 28, Use Case Specification (Search Patient)

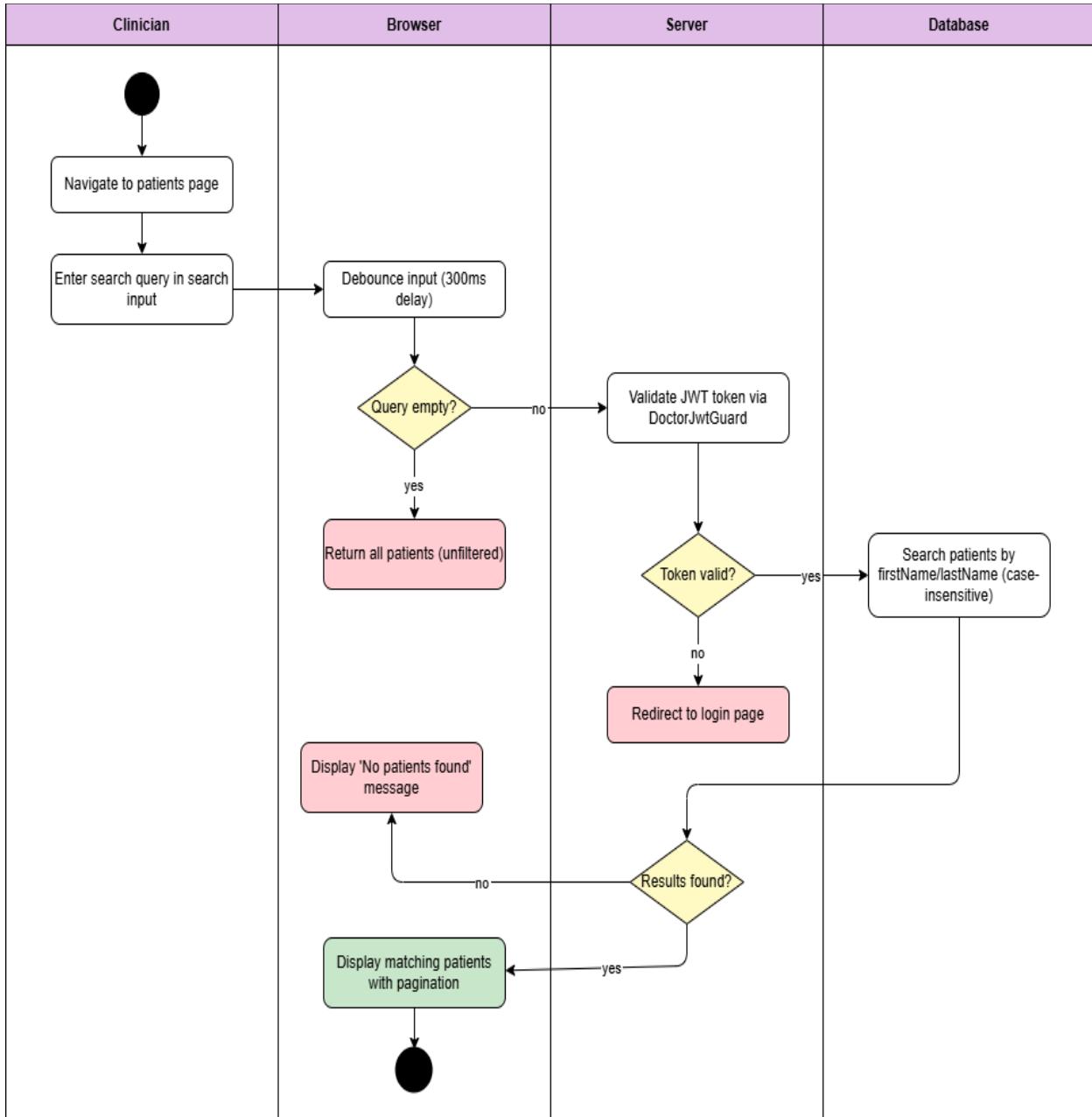


Diagram 46, Activity Diagram (Search Patient)

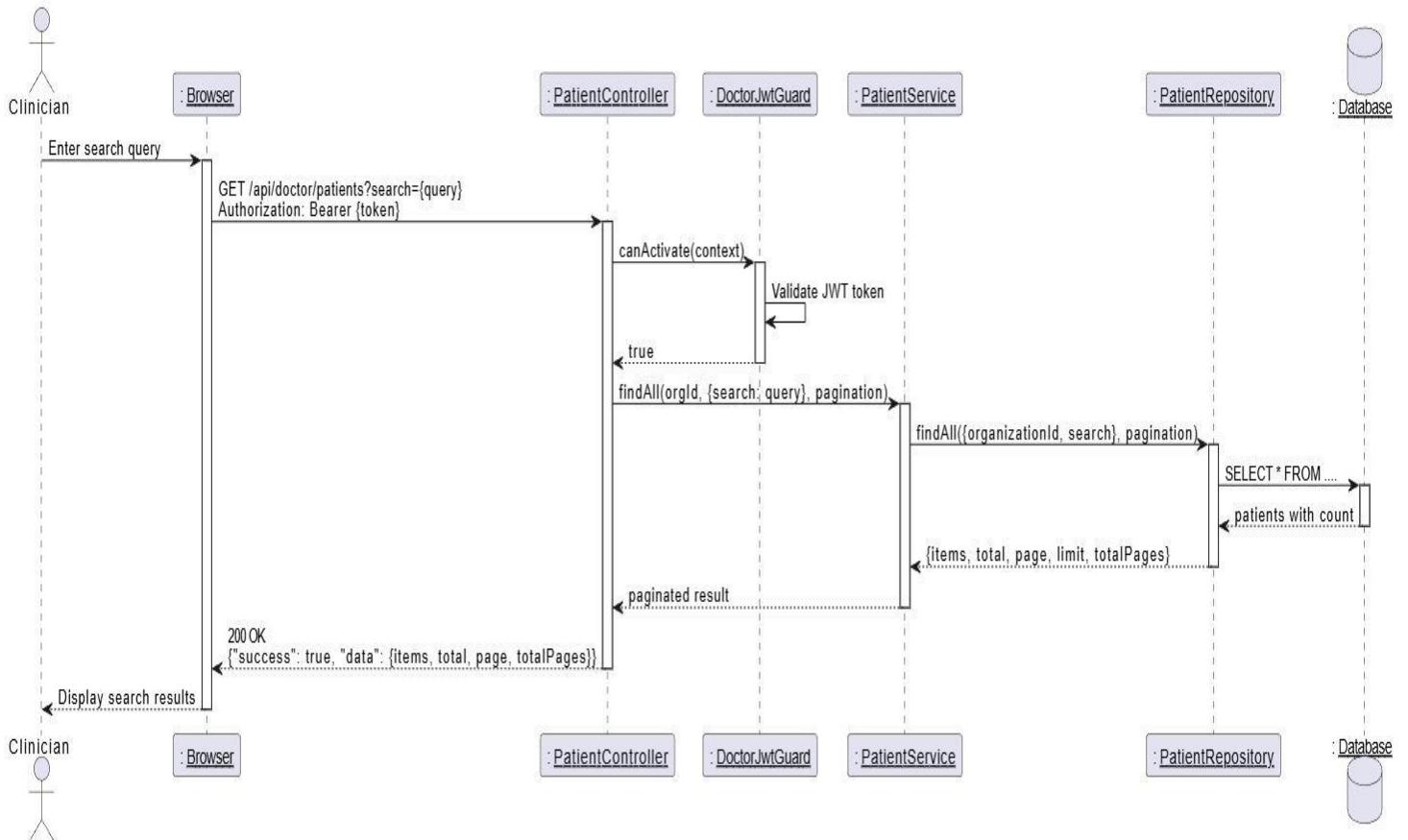


Diagram 47, Sequence Diagram (Search Patient)

- **View Patient Profile:**

Use case ID	VEMR-FR-VM-22
Use case name	View patient Profile
Description	The system allows clinicians to view a patient's profile with their visits and allergies.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to patients list page 2.Clinician searches for patient by name 3. Clinician clicks on patient name to view profile 4.System validates JWT token via DoctorJwtGuard 5.System retrieves patient data via Patient Repository 6.System retrieves patient visits via Encounter Repository 7. System retrieves patient allergies via Allergy Repository 8.System displays patient profile with personal details, visits history, and allergies list
Alternative scenario	<p>A1: Patient Not Found</p> <ul style="list-style-type: none"> - At step 5, if patient ID does not exist -System displays error message <p>A2: No Visits or Allergies</p> <ul style="list-style-type: none"> - At step 6 or 7, if patient has no visits or allergies -System displays empty state message
Post condition	Patient details are displayed

Table 29, Use Case Specification (View Patient Profile)

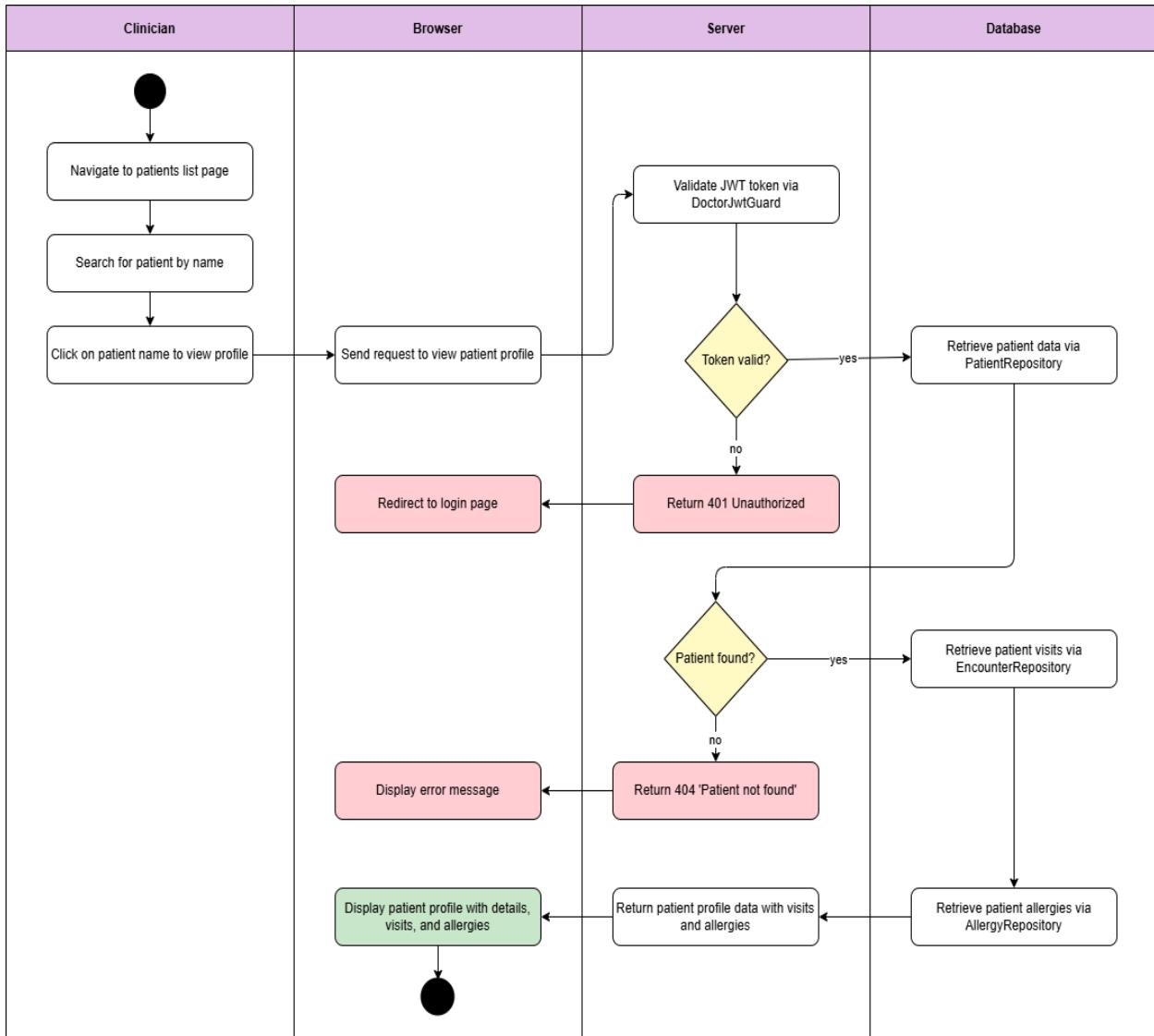


Diagram 48, Activity Diagram (View Patient Profile)

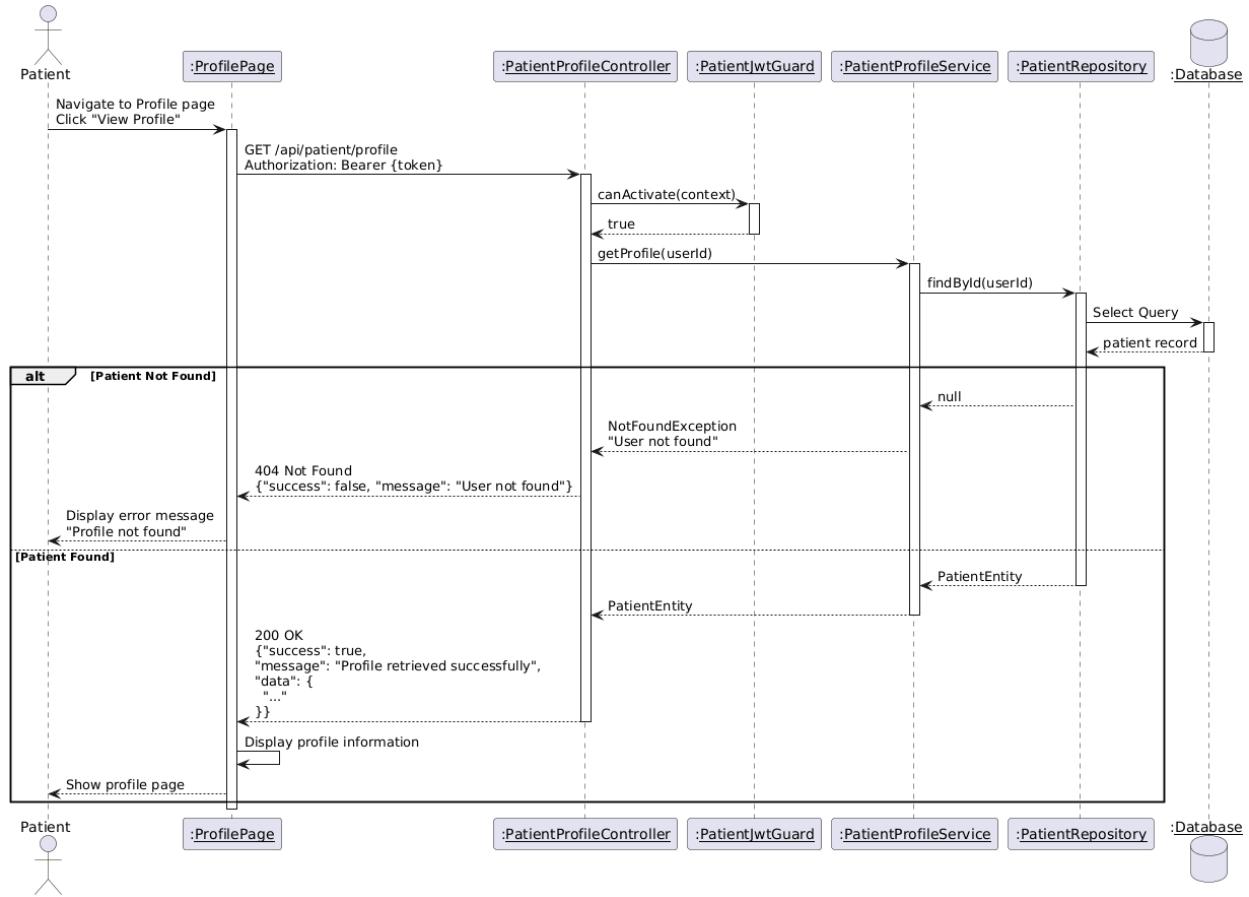


Diagram 49, Sequence Diagram (View Patient Profile)

- **View Patient Details:**

Use case ID	VEMR-FR-VM-23
Use case name	View patient details
Description	The system allows clinicians to view detailed patient information
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician click on a patient from the patients list 2.System navigates to patient detail page 3.System authenticates user and retrieves patient details 4.System retrieves patient encounters and allergies 5.System displays patient profile with tabs (Visits, Allergies)</p>
Alternative scenario	<p>A1: Patient Not Found</p> <ul style="list-style-type: none"> - At step 3, if patient does not exist -System displays error page or redirects <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired -System redirects to login page
Post condition	Patient details are displayed

Table 30, Use Case Specification (View Patient Details)

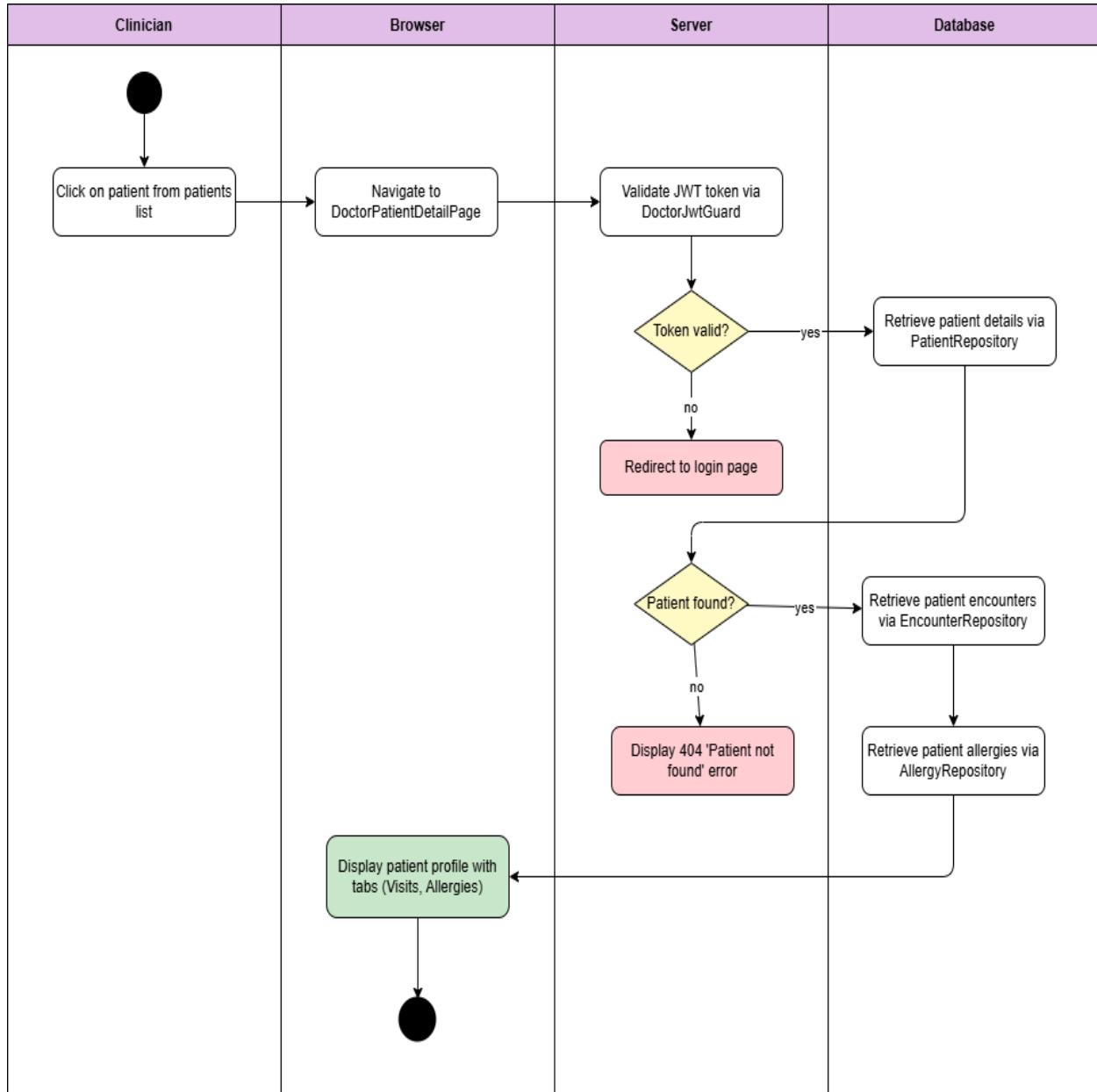


Diagram 50, Activity Diagram (View Patient Details)

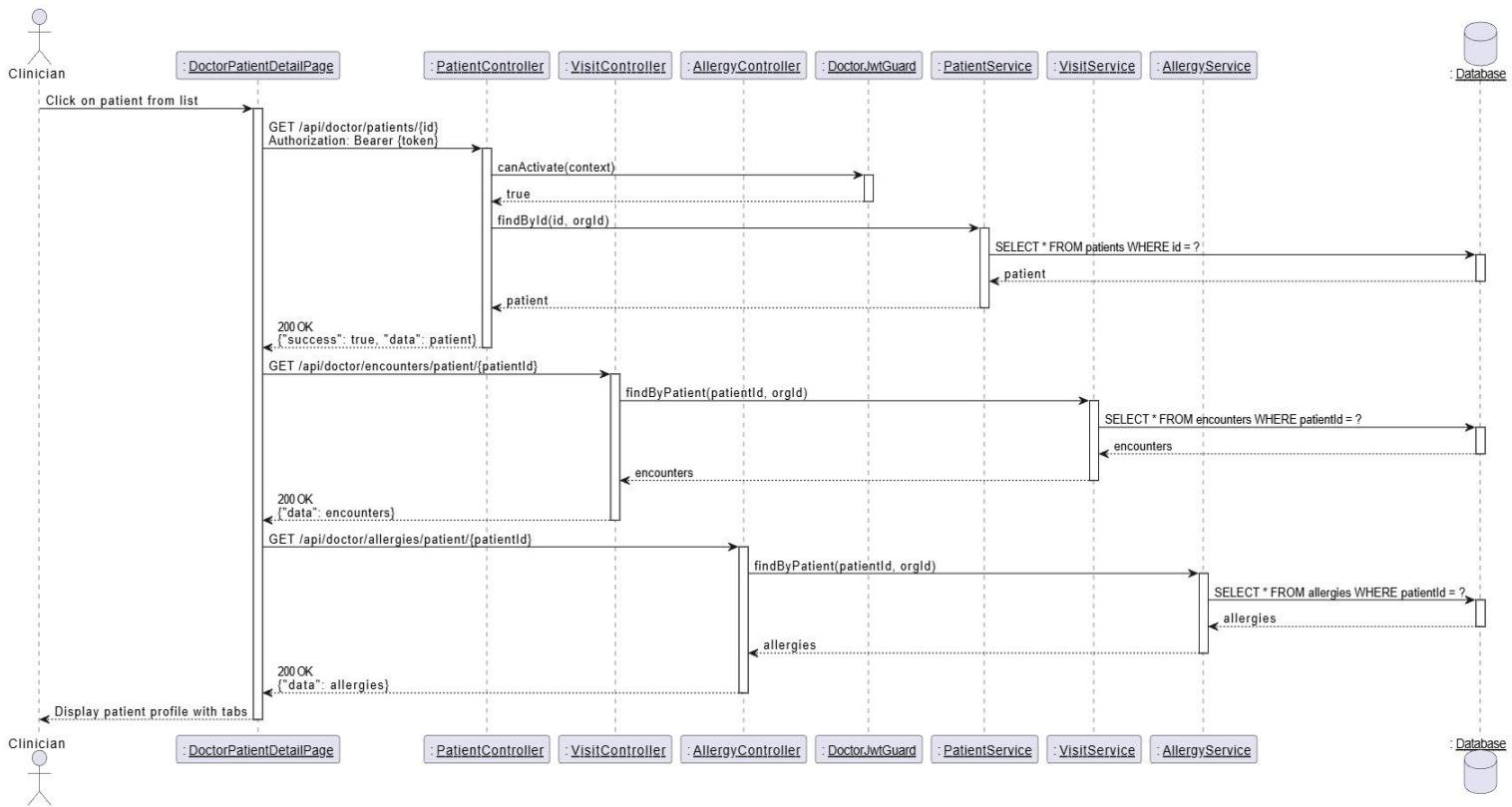


Diagram 51, Sequence Diagram (View Patient Details)

● Change Visit Status:

Use case ID	VEMR-FR-VM-24
Use case name	Change visit status
Description	The system allows clinicians to change visit status (Start - Complete - Cancel).
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>Start Visit:</p> <ol style="list-style-type: none"> 1.Clinician clicks “Start visit” button on a planned visit. 2.System authenticates user and validates visit status is a “planned” 3.System updates status to “InProgress” and sets start Time 4.System displays success message. <p>Complete Visit:</p> <ol style="list-style-type: none"> 1.Clinician clicks “complete visit” button on an in-progress visit. 2.System authenticates user and validates visit status is a “InProgress” 3.System updates status to “Complete” and sets end Time 4.System displays success message.
Alternative scenario	<p>A1: Invalid Status Transition</p> <ul style="list-style-type: none"> - At step 2, if visit status does not allow the transition -System displays error message <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired -System redirects to login page
Post condition	

Table 31, Use Case Specification (Change Visit Status)

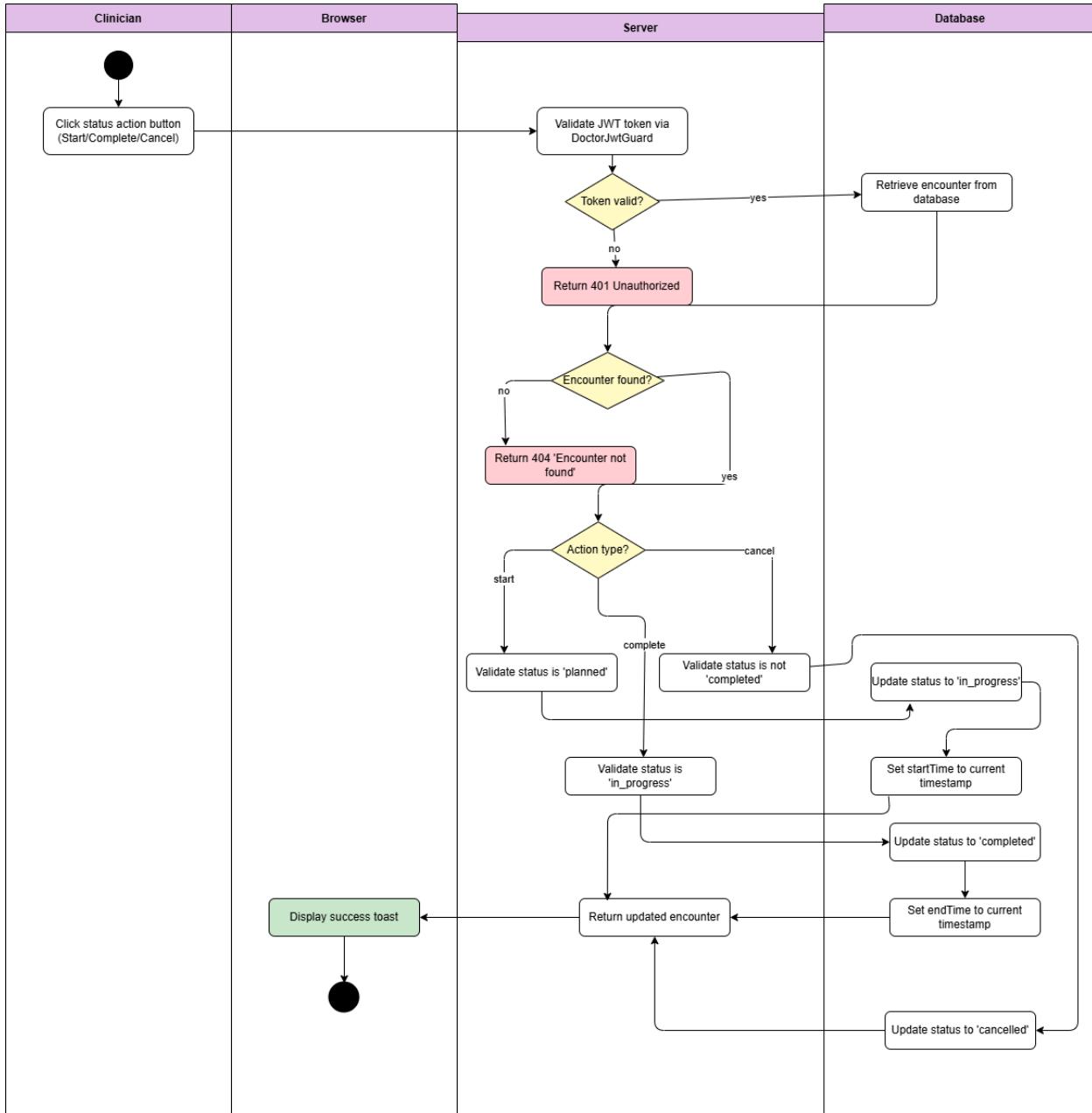


Diagram 52, Activity Diagram (Change Visit Status)

Chapter 4 - System Analysis

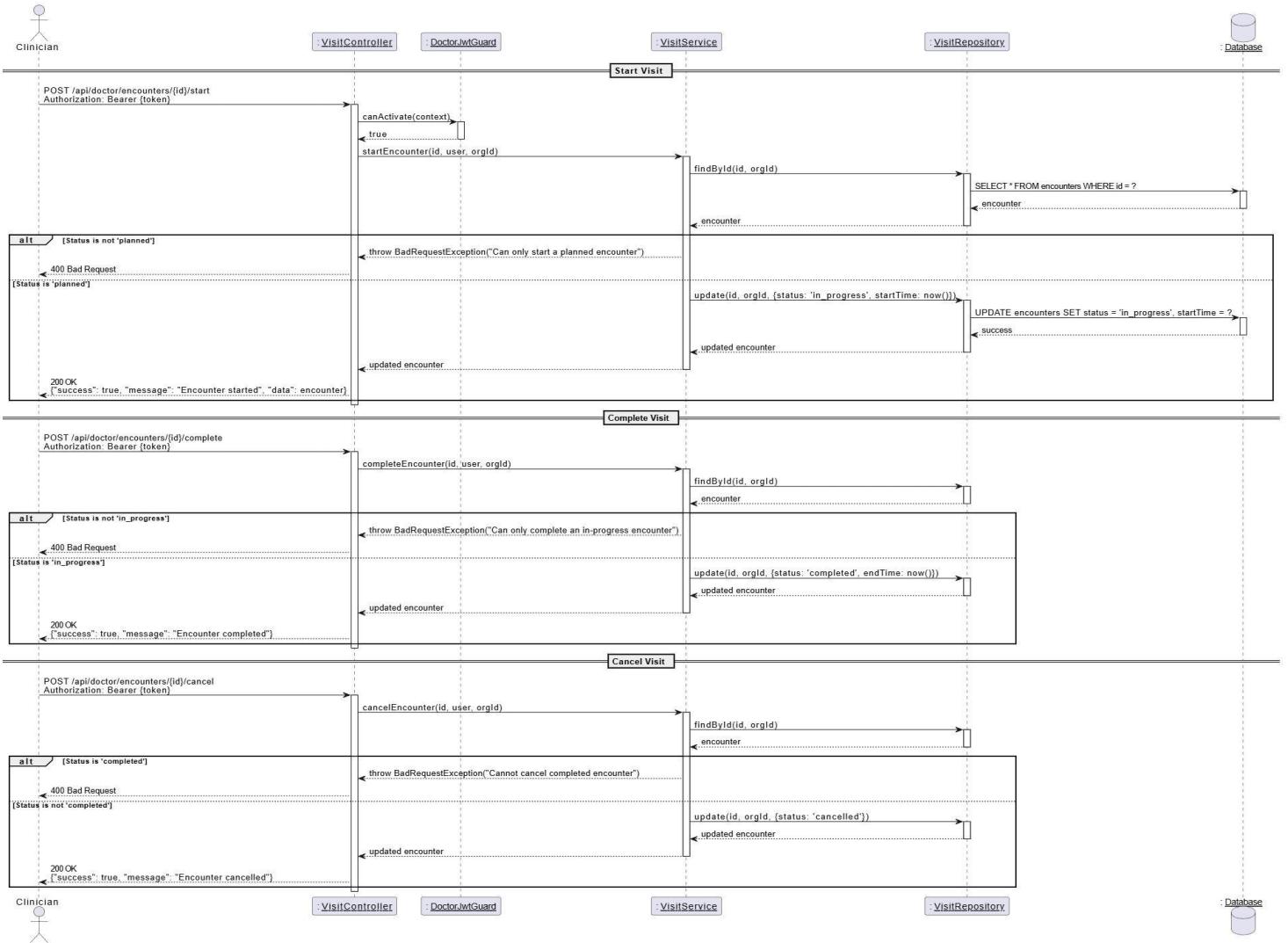


Diagram 53, Sequence Diagram (Change Visit Status)

- **View Visit List:**

Use case ID	VEMR-FR-VM-25
Use case name	View visit list
Description	The system allows clinicians to view a paginated list
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician navigated to visits page 2.System authenticates user and retrieves paginated encounter 3.System displays visits in a table with patient name, status, date, type. 4.Clinicians can filter by status, search, or paginate</p>
Alternative scenario	<p>A1: No visit</p> <ul style="list-style-type: none"> - At step 2, if no encounters exist -System displays “No visits found” message <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired -System redirects to login page
Post condition	Visits list is displayed with pagination

Table 32, Use Case Specification (View Visit List)

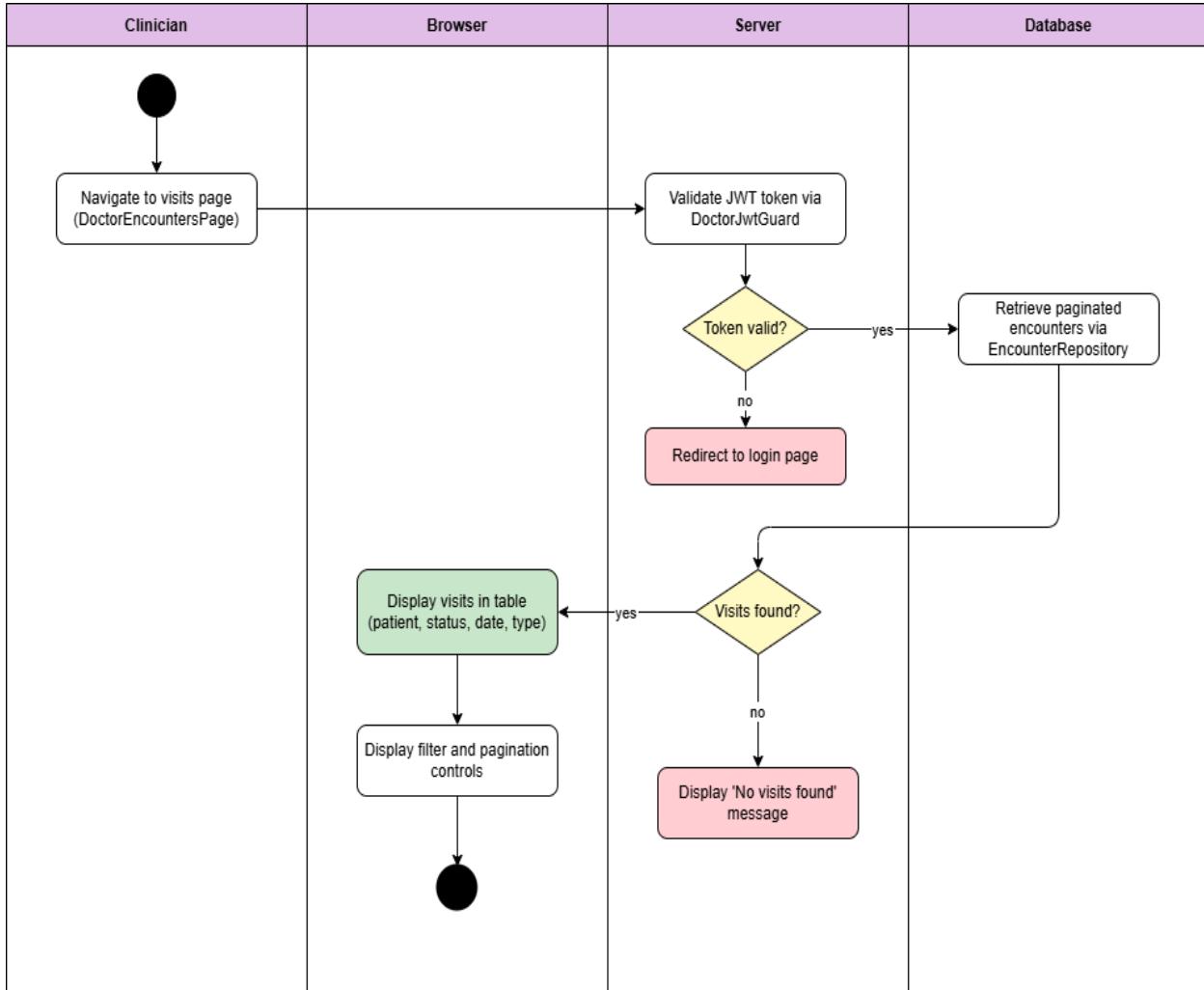


Diagram 54, Activity Diagram (View Visit List)

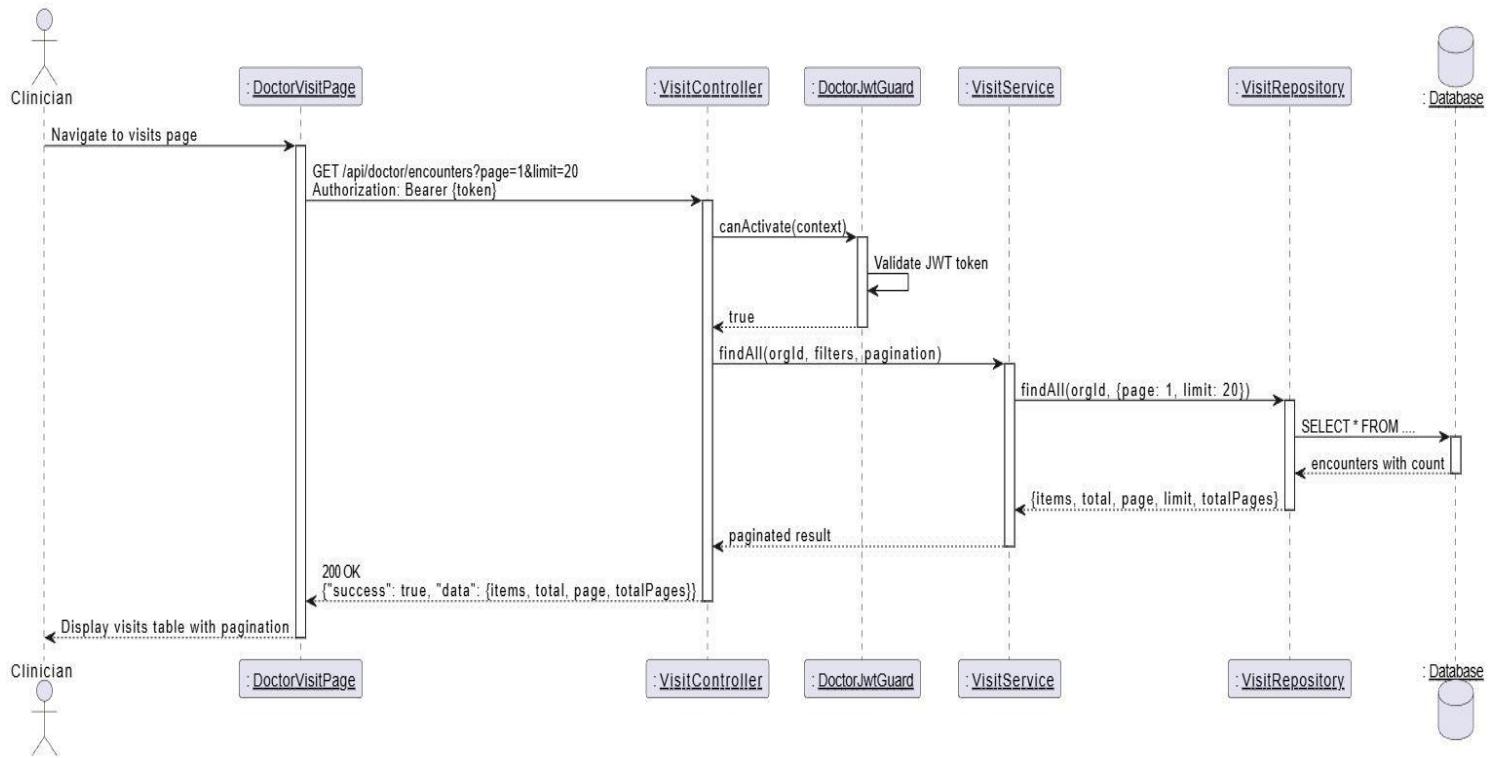


Diagram 55, Sequence Diagram (View Visit List)

● View Patient Vital Signs:

Use case ID	VEMR-FR-VS-26
Use case name	View Patient Vital Signs
Description	The system allows clinicians to view vital signs recorded
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician opens visit detail page and clicks “Vital signs” tab 2.System authenticates user and retrieves vital signs for this encounter 3.System displays vital signs grid (BP - HR - Temp - RR - Spo2 - Weight -Height - BMI).</p>
Alternative scenario	<p>A1: No Vital Record</p> <ul style="list-style-type: none"> - At step 2,if no vital exist for this visit . -System displays empty state with “- - “values <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 2,if JWT token is invalid or expired -System redirects to login page
Post condition	Visits list

Table 33, Use Case Specification (View Patient Vital Signs)

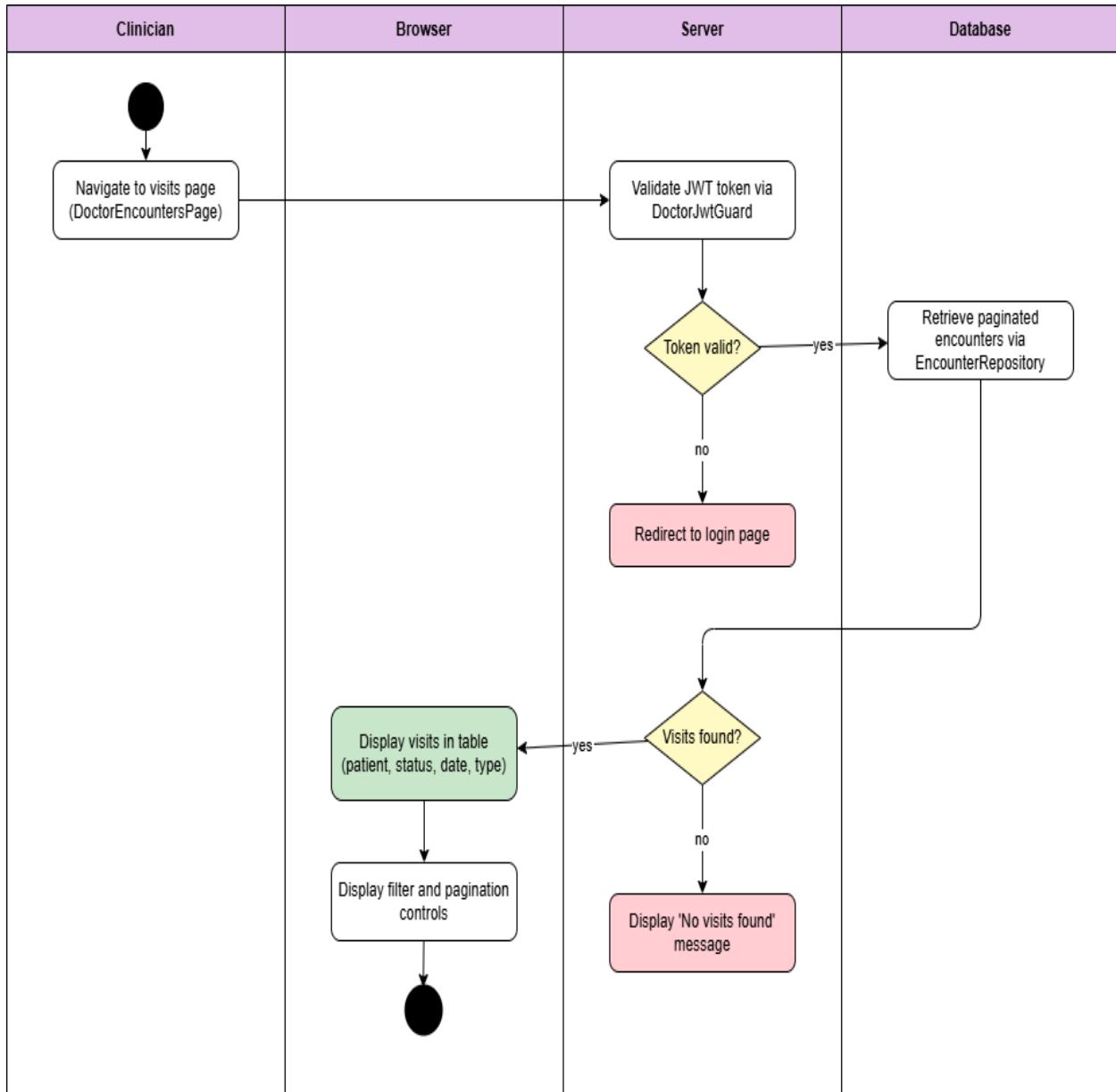


Diagram 56, Activity Diagram (View Patient Vital Signs)

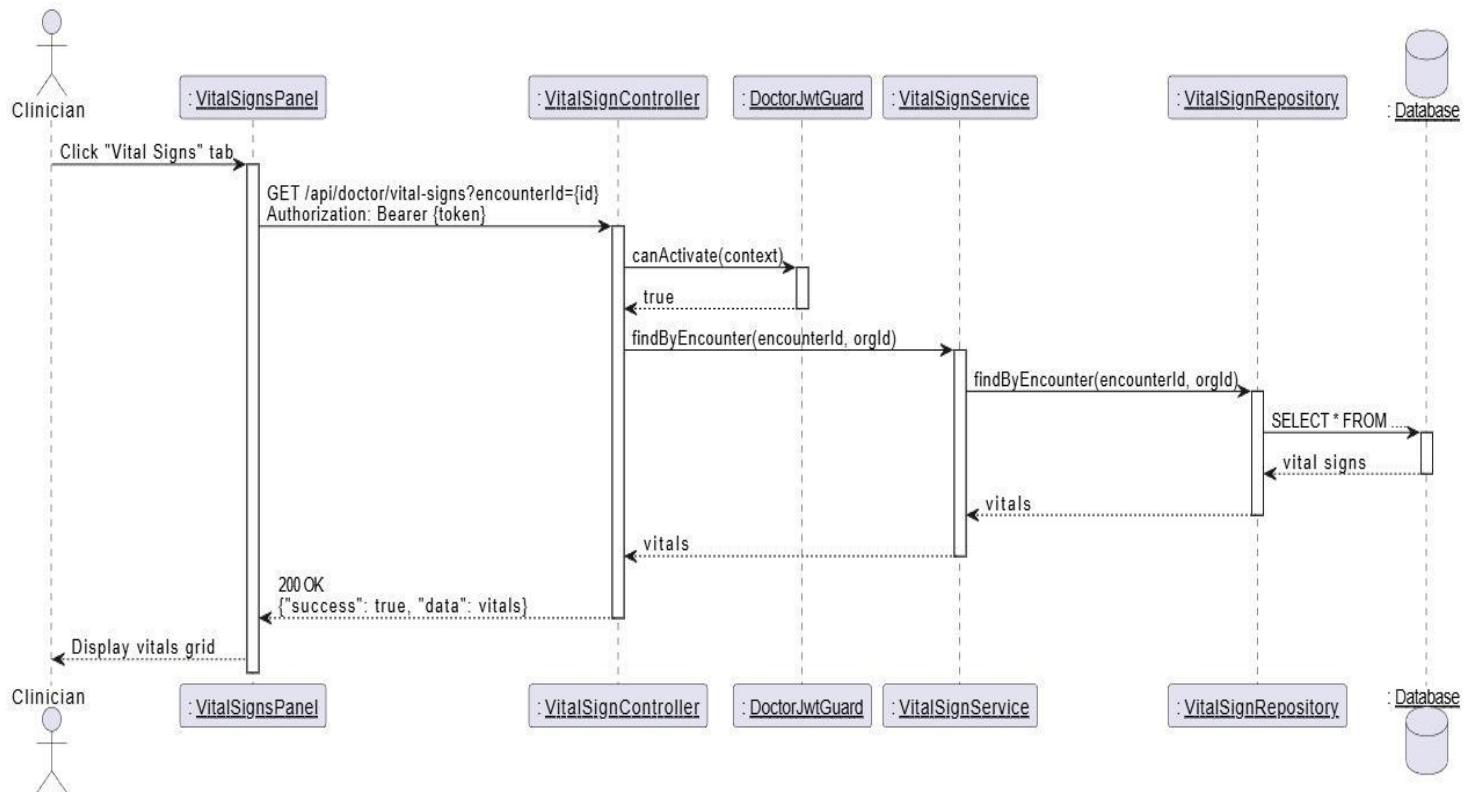


Diagram 57, Sequence Diagram (View Patient Vital Signs)

● Edit Patient Vital Signs:

Use case ID	VEMR-FR-VS-27
Use case name	Edit Patient Vital Signs
Description	The system allows clinicians to record and edit vital signs
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician clicks “Record Vitals ” button on vital signs panel</p> <p>2.System displays vital sign form modal</p> <p>3.Clinicians selects vital sign type and enters value and unit</p> <p>4.Clinicians submits form</p> <p>5.System validates input and authenticates user</p> <p>6.System creates vital sign record linked to encounter</p> <p>7.System displays updated vitals grid</p>
Alternative scenario	<p>A1: Visit Not in Progress</p> <ul style="list-style-type: none"> - if visit is not in progress. -Record button is hidden or disabled <p>A2: Invalid Value</p> <ul style="list-style-type: none"> - At step 5, if value is invalid (out of range -wrong format) -System displays validation error <p>A3: Authentication Error</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired -System redirects to login page
Post condition	Visits sign is recorded for this visit

Table 34, Use Case Specification (Edit Patient Vital Signs)

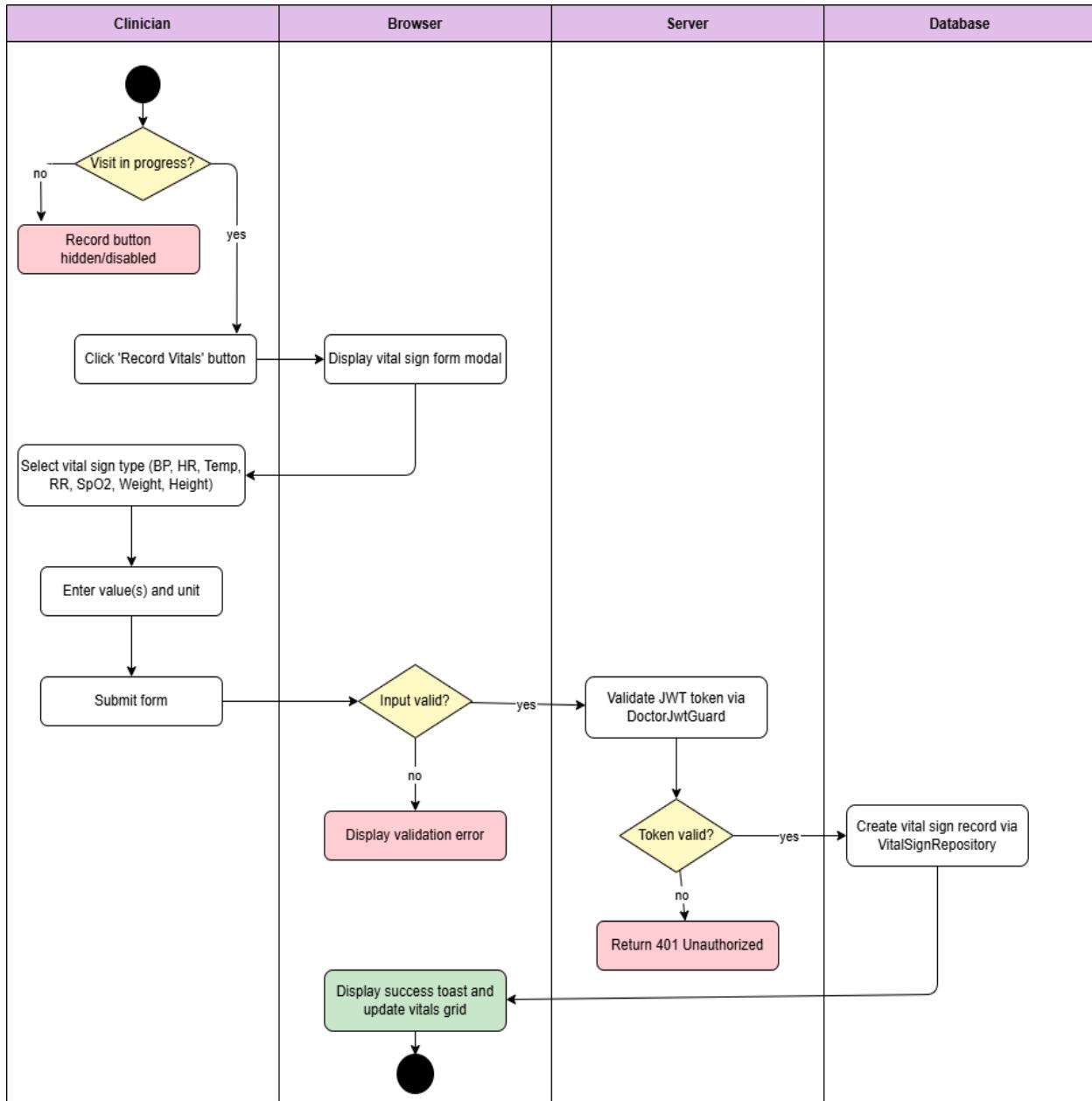


Diagram 58, Activity Diagram (Edit Patient Vital Signs)

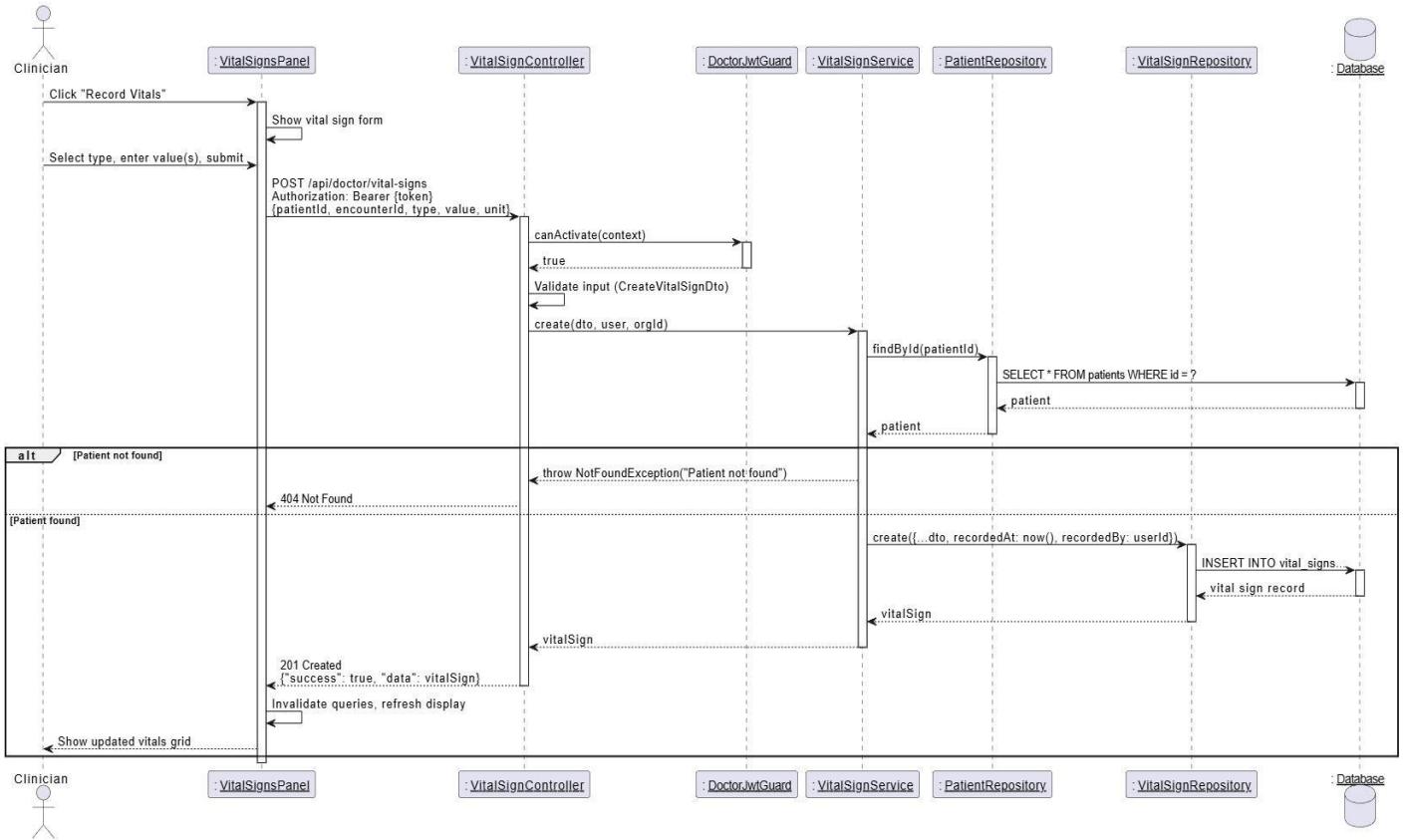


Diagram 59, Sequence Diagram (Edit Patient Vital Signs)

- **View Allergies:**

Use case ID	VEMR-FR-AM-28
Use case name	View Allergies
Description	The system allows clinicians to view a patient recorded allergies with severity
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician opens visit details or patient profile page and clicks “Allergies” tab.</p> <p>2.System authenticates user and retrieves patient allergies</p> <p>3.System displays allergies list with allergies, type, severity, reaction</p>
Alternative scenario	<p>A1: No Allergies</p> <ul style="list-style-type: none"> - At step 2, if no allergies recorded for patient -System displays “No allergies recorded message” <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired -System redirects to login page
Post condition	Allergies are displayed in a list format

Table 35, Use Case Specification (View Allergies)

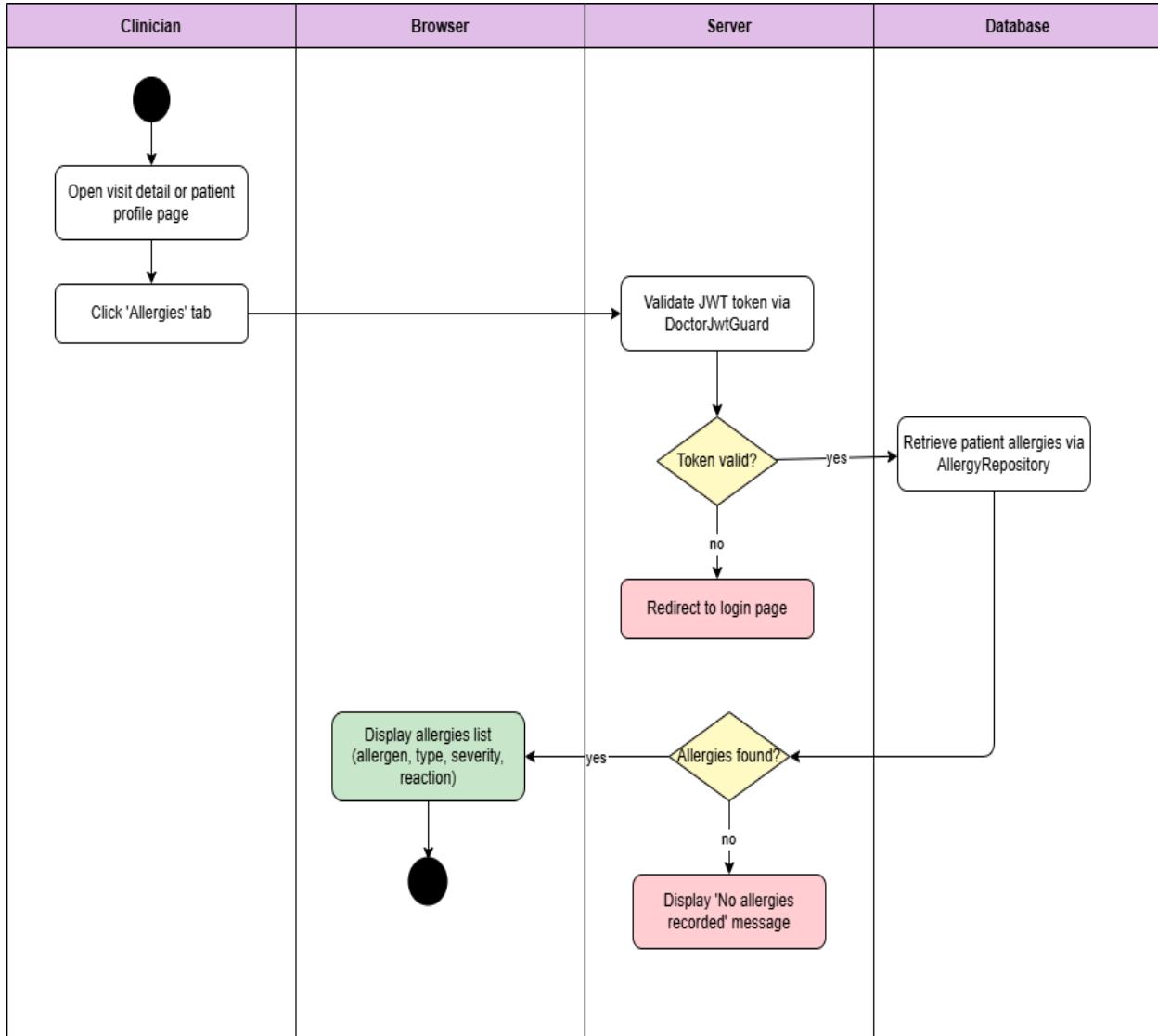


Diagram 60, Activity Diagram (View Allergies)

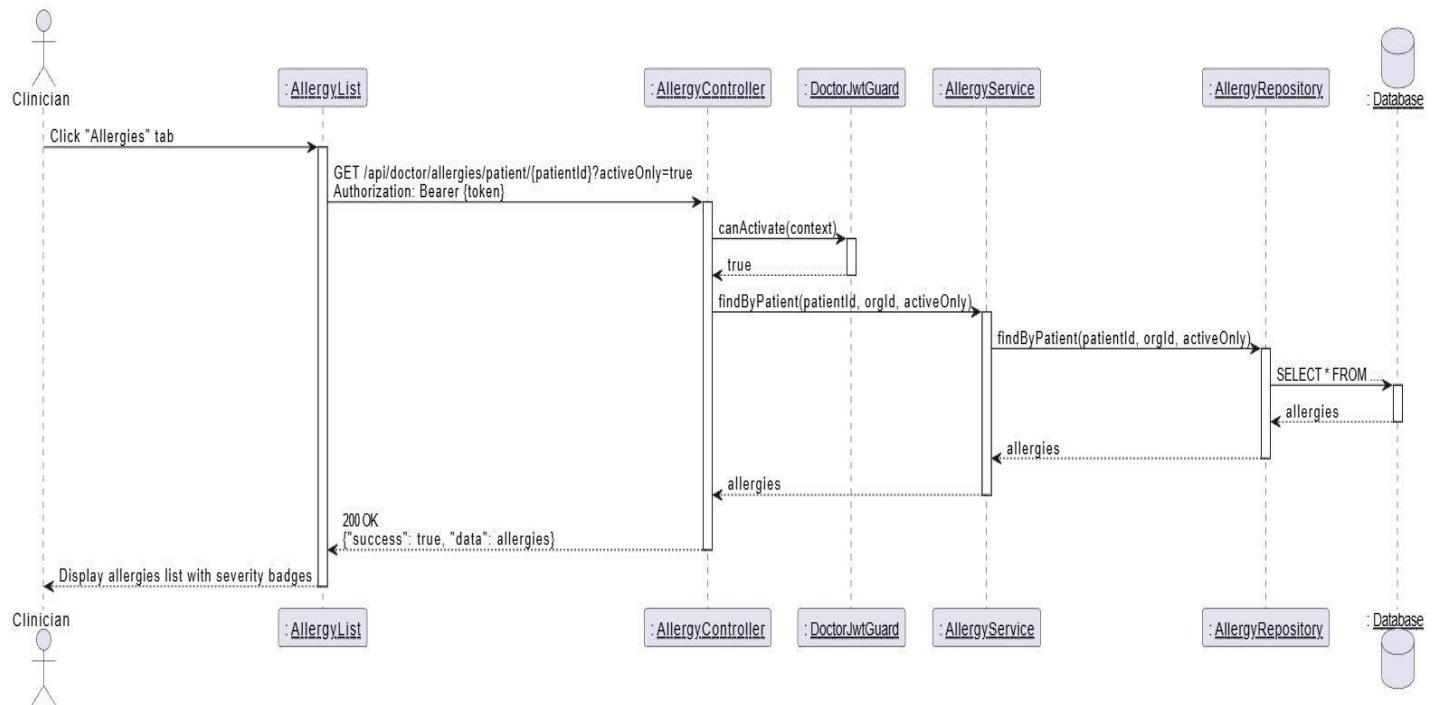


Diagram 61, Sequence Diagram (View Allergies)

● Create Allergies:

Use case ID	VEMR-FR-AM-29
Use case name	Create Allergies
Description	The system allows clinicians to add a new allergy record for a patient
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician clicks “Add Allergy” button on allergies panel 2.System displays allergy form modal 3.Clinicians enters allergy form details (type, allergen, severity, reaction, onset date, notes). 4.Clinicians submits form 5.System validates input, authenticates user, and verifies patient exist. 6.System checks for duplicate allergen and creates allergy record 7.System closes modal and refreshed allergy list</p>
Alternative scenario	<p>A1: Duplicate or Not Found Error</p> <ul style="list-style-type: none"> - At step 5 - 6, if allergen already exists or patient not found -System displays error message. <p>A2: Visit Not In Progress</p> <ul style="list-style-type: none"> - If visit is not in progress -Add button is hidden or disabled <p>A3: Authentication Error</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired -System redirects to login page
Post condition	Allergies is created for the patient

Table 36, Use Case Specification (Create Allergies)

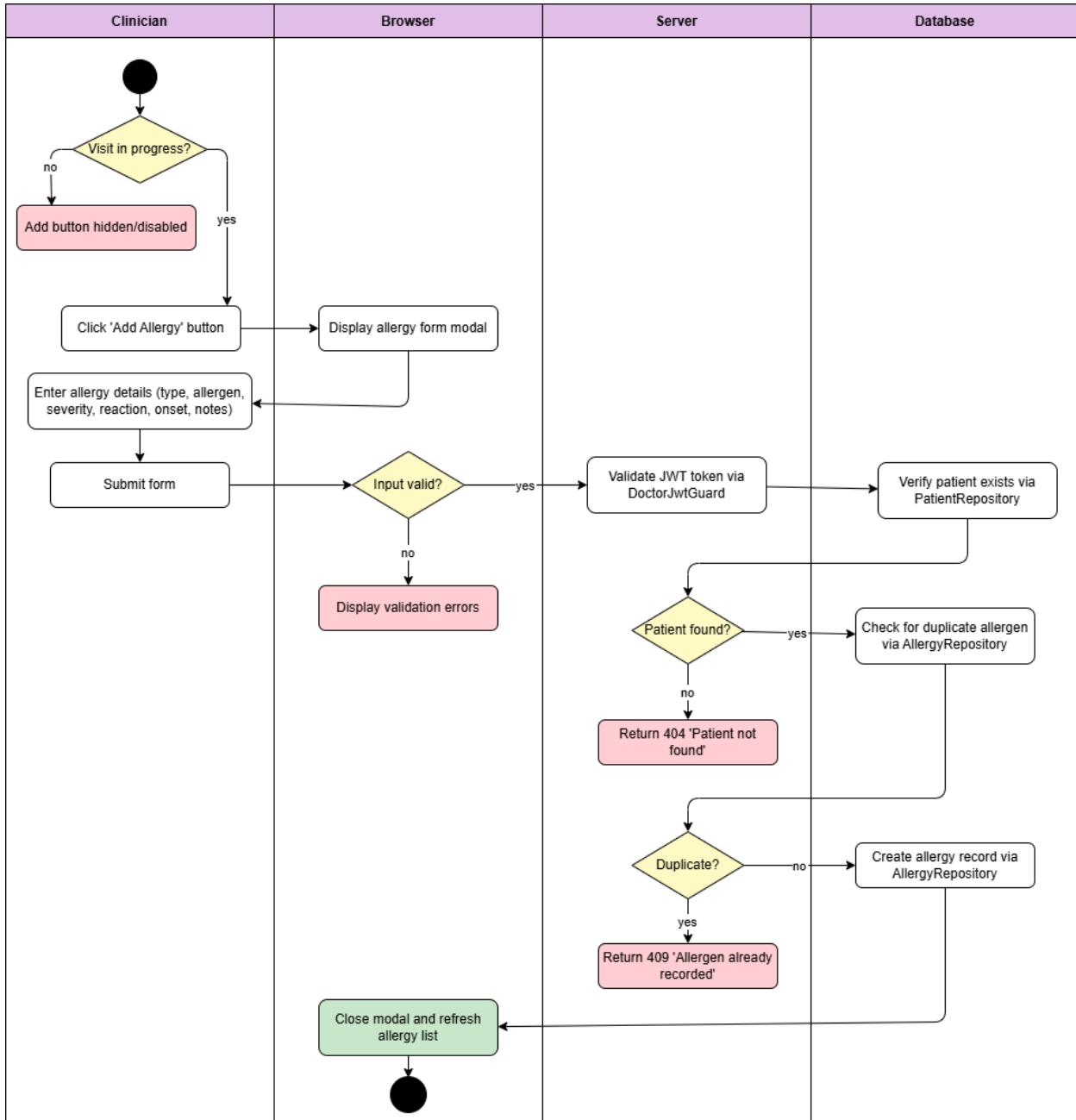


Diagram 62, Activity Diagram (Create Allergies)

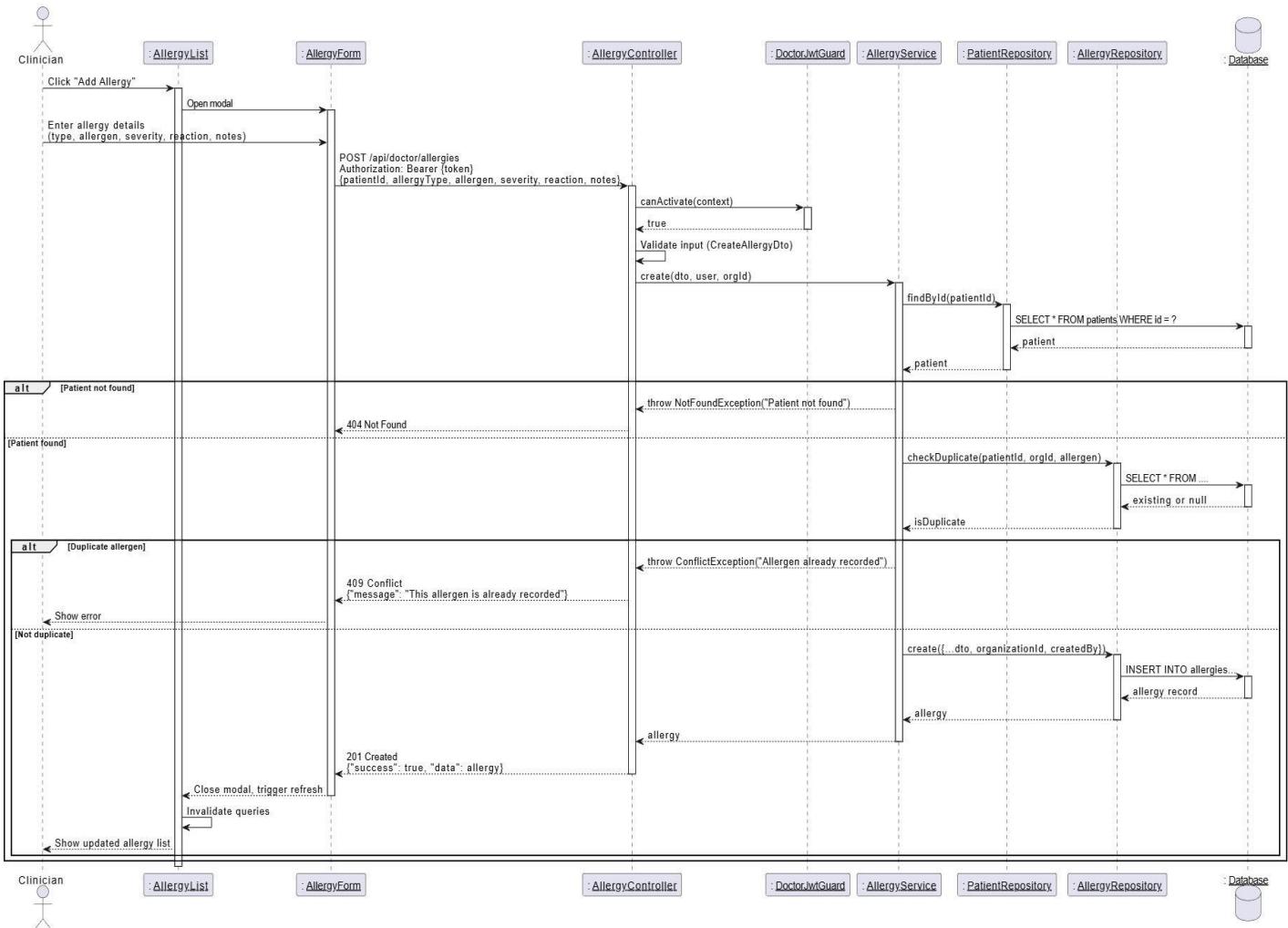


Diagram 63, Sequence Diagram (Create Allergies)

● Update Allergies:

Use case ID	VEMR-FR-AM-30
Use case name	Update Allergies
Description	The system allows clinicians to update existing allergy information for a patient
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician clicks “Edit” button on allergy in the list 2.System displays allergy from modal with existing data 3.Clinicians modifies allergy details and submit form 4.System validates input, authenticates user, and retrieves existing allergy 5.System checks for duplicate if allergen changed and updates allergy record 6.System closes modal and refreshed allergy list</p>
Alternative scenario	<p>A1: Duplicate or Not Found Error</p> <ul style="list-style-type: none"> - At step 4 - 5, if new allergen already exists or allergy not found -System displays error message. <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired -System redirects to login page
Post condition	Allergies is updated

Table 37, Use Case Specification (Update Allergies)

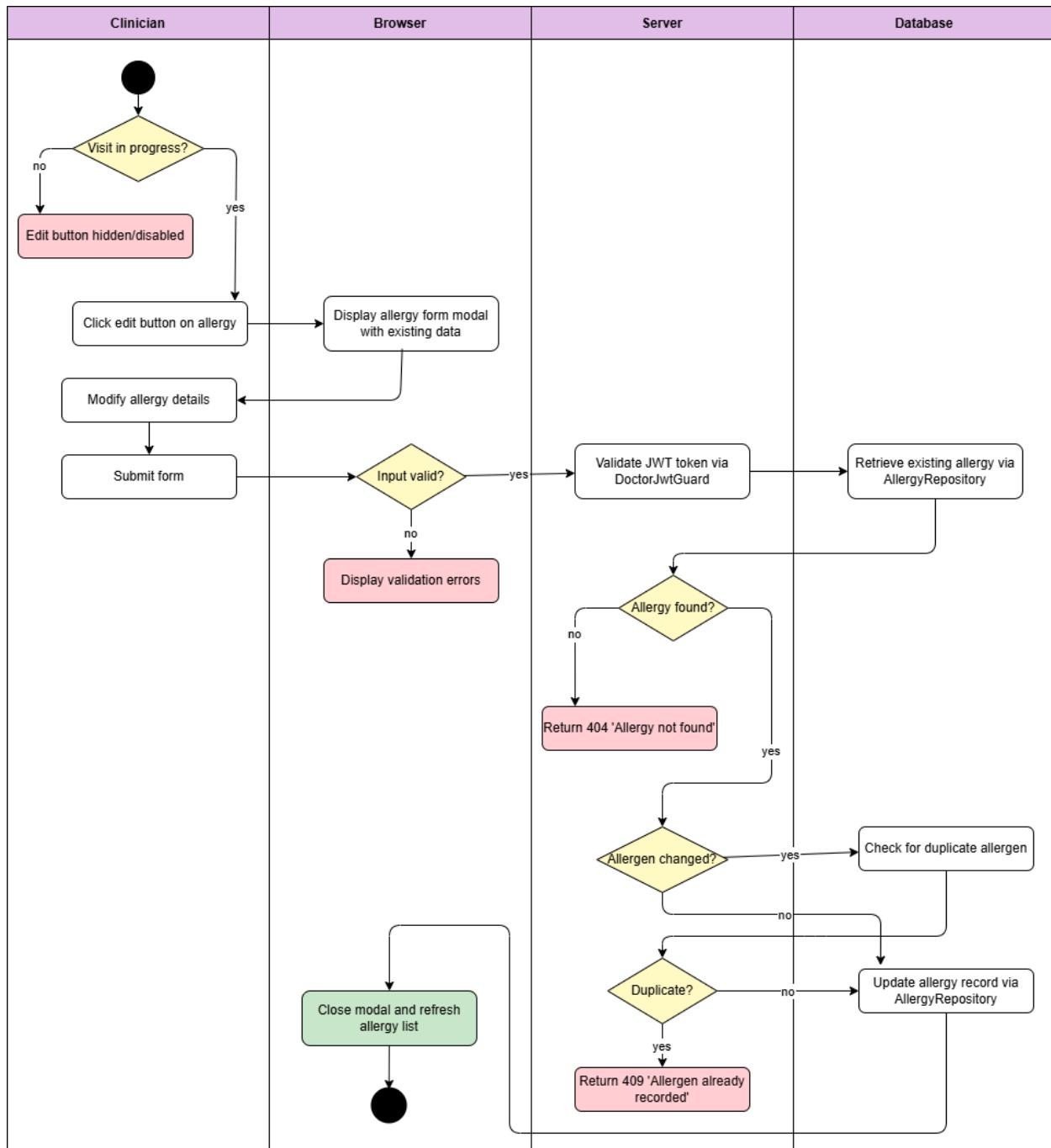


Diagram 64, Activity Diagram (Update Allergies)

Chapter 4 - System Analysis

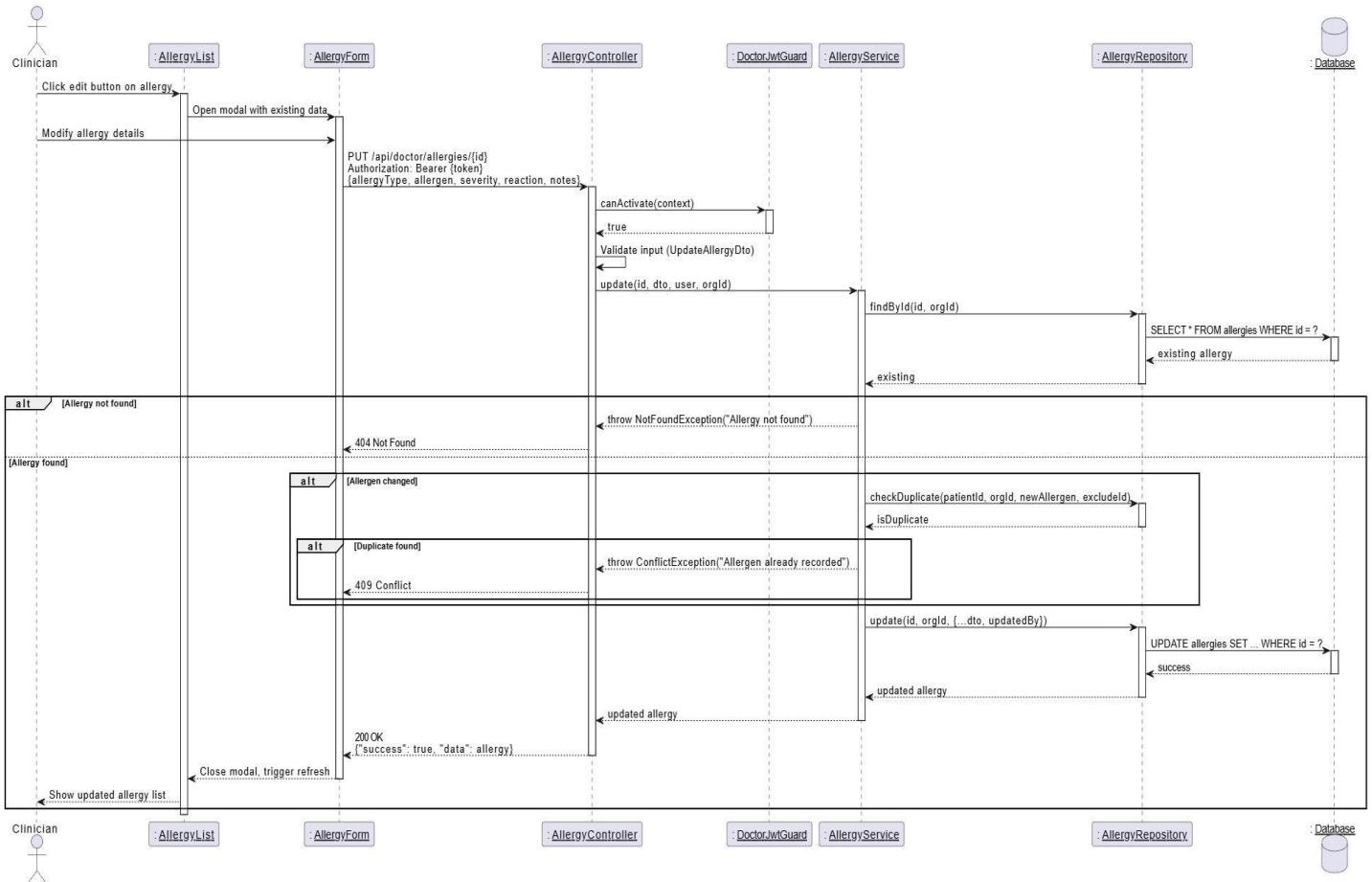


Diagram 65, Sequence Diagram (Update Allergies)

● Delete Allergies:

Use case ID	VEMR-FR-AM-31
Use case name	Delete Allergies
Description	The system allows clinicians to delete an allergy record from a patient
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<p>1.Clinician clicks “Delete” button on allergy in the list 2.System prompts for configuration dialog 3.Clinicians confirms deletion 4.System authenticates user and retrieves allergy 5.System soft-deletes allergy record (set delete At , delete By) 6.System displays success message and refreshes allergy list</p>
Alternative scenario	<p>A1: Allergy Not Found Error</p> <ul style="list-style-type: none"> - At step 4, if allergy does noy exist -System displays error message. <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired -System redirects to login page <p>A3: User Cancel</p> <ul style="list-style-type: none"> - At step 3, if clinician cancels confirmation -No action is taken
Post condition	Allergy is deleted (deleted at is set)

Table 38, Use Case Specification (Delete Allergies)

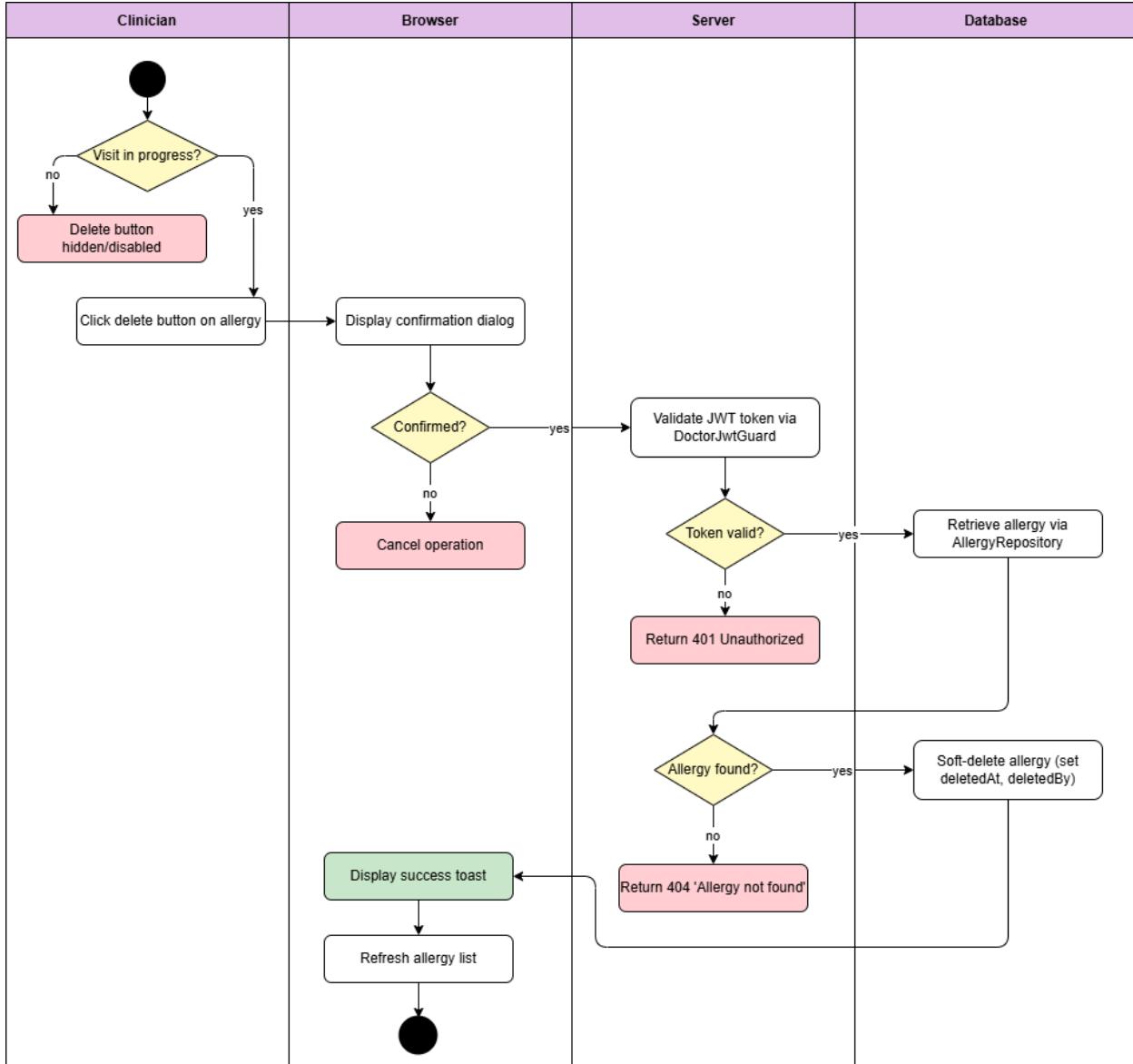


Diagram 66, Activity Diagram (Delete Allergies)

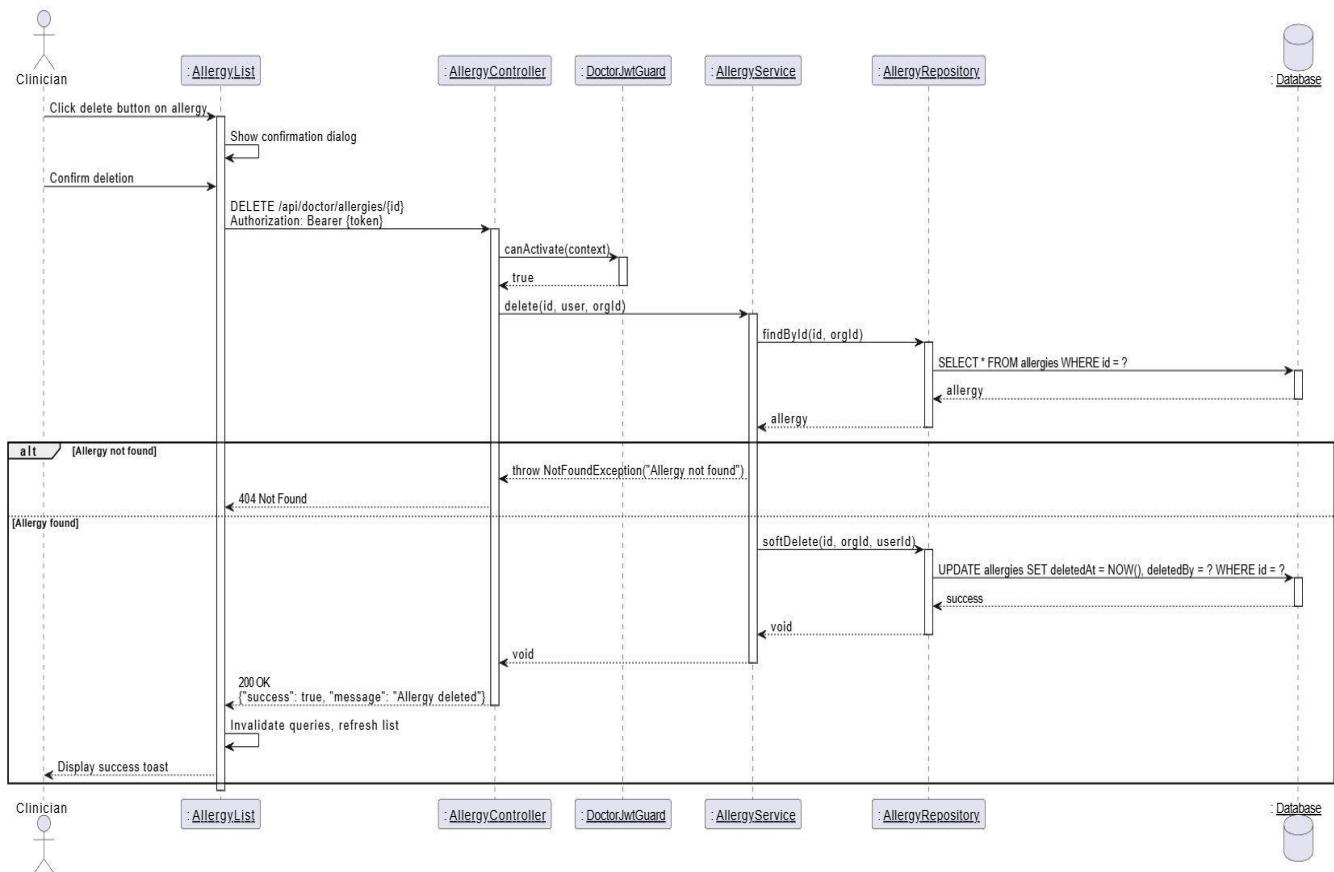


Diagram 67, Sequence Diagram (Delete Allergies)

● Create Clinicians Account:

Use case ID	VEMR-FR-CM-32
Use case name	Create clinicians account
Description	The system allows administrators to create new clinicians account
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<p>1.Admin navigates to doctor management page 2.Admin click “Add Doctor” button 3.System displays Create Doctor Form modal 4.Admin enters doctor details 5.Admin submits the form 6.System validates input and check email uniqueness 7.System hashes password and creates doctor account with status “inactive” 8.System displays success message</p>
Alternative scenario	<p>A1: Validation Error</p> <ul style="list-style-type: none"> - At step 6, if required fields are missing or email format invalid -System displays error message on form <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 6 ,if JWT token is invalid or expired -System redirects to login page <p>A3: Email Already Exist</p> <ul style="list-style-type: none"> - At step 6, if email is already registered -System displays “Email already exist” error
Post condition	New doctor account is created with status

Table 39, Use Case Specification (Create Clinicians Account)

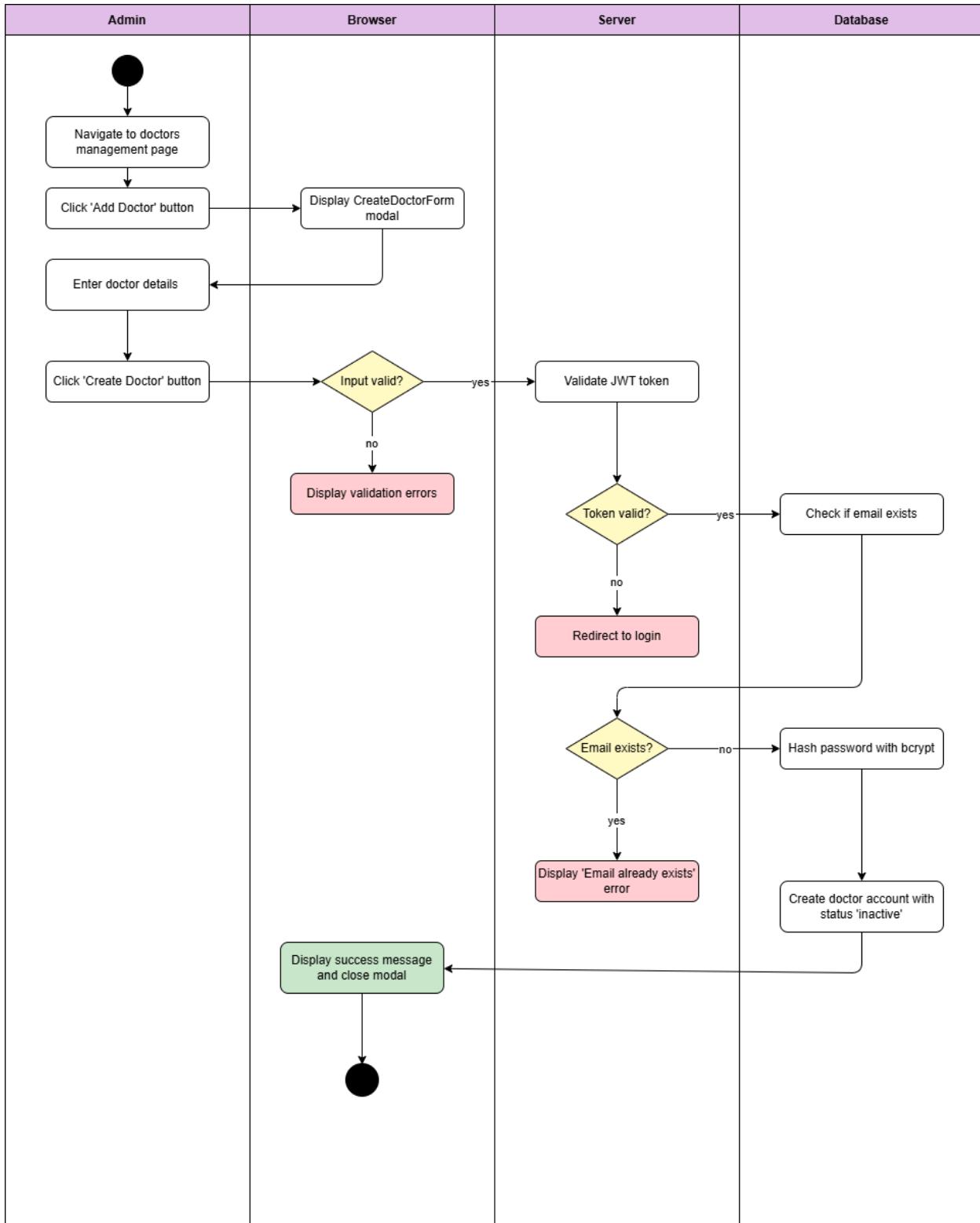


Diagram 68, Activity Diagram (Create Clinicians Account)

Chapter 4 - System Analysis

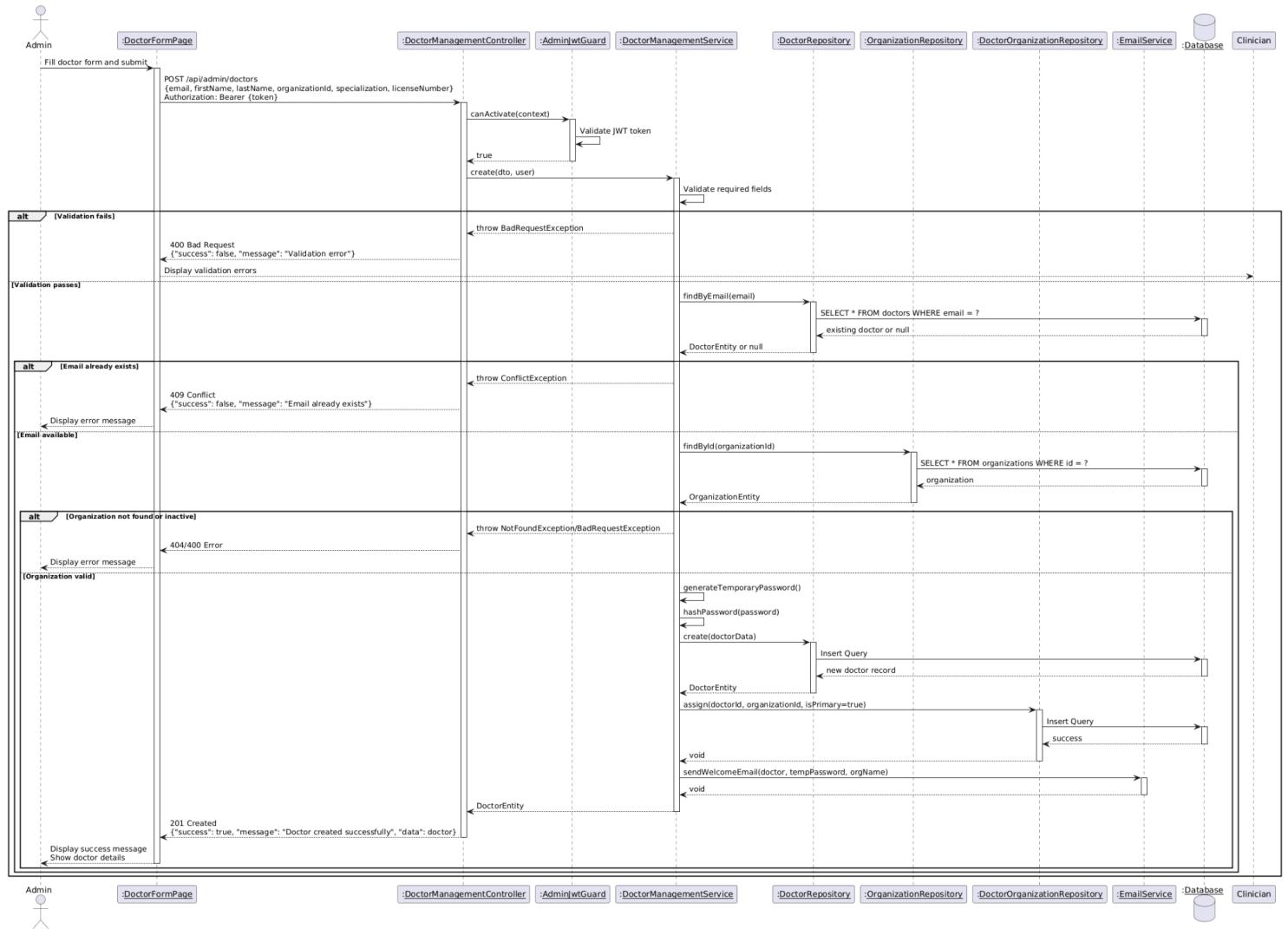


Diagram 69, Sequence Diagram (Create Clinicians Account)

● Update Clinicians Account:

Use case ID	VEMR-FR-CM-33
Use case name	Update clinicians account
Description	The system allows administrators to update existing clinicians account
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<p>1.Admin navigates to doctor management page 2.Admin selects a doctor from the list 3.Admin click "Edit" button 4.System displays Edit Doctor Form modal with current doctor information 5.Admin modifies doctor details 6.Admin submits the form 7.System validates input 8.System updates doctor account information 9.System displays success message and refreshes doctor list</p>
Alternative scenario	<p>A1: Validation Error</p> <ul style="list-style-type: none"> - At step 7 ,if required fields are missing or email format invalid -System displays error message on form <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 7 ,if JWT token is invalid or expired -System redirects to login page <p>A3: Doctor Not Found</p> <ul style="list-style-type: none"> - At step 4, if doctor ID is invalid -System displays “Doctor Not Found” error
Post condition	Doctor account information is updated

Table 40, Use Case Specification (Update Clinicians Account)

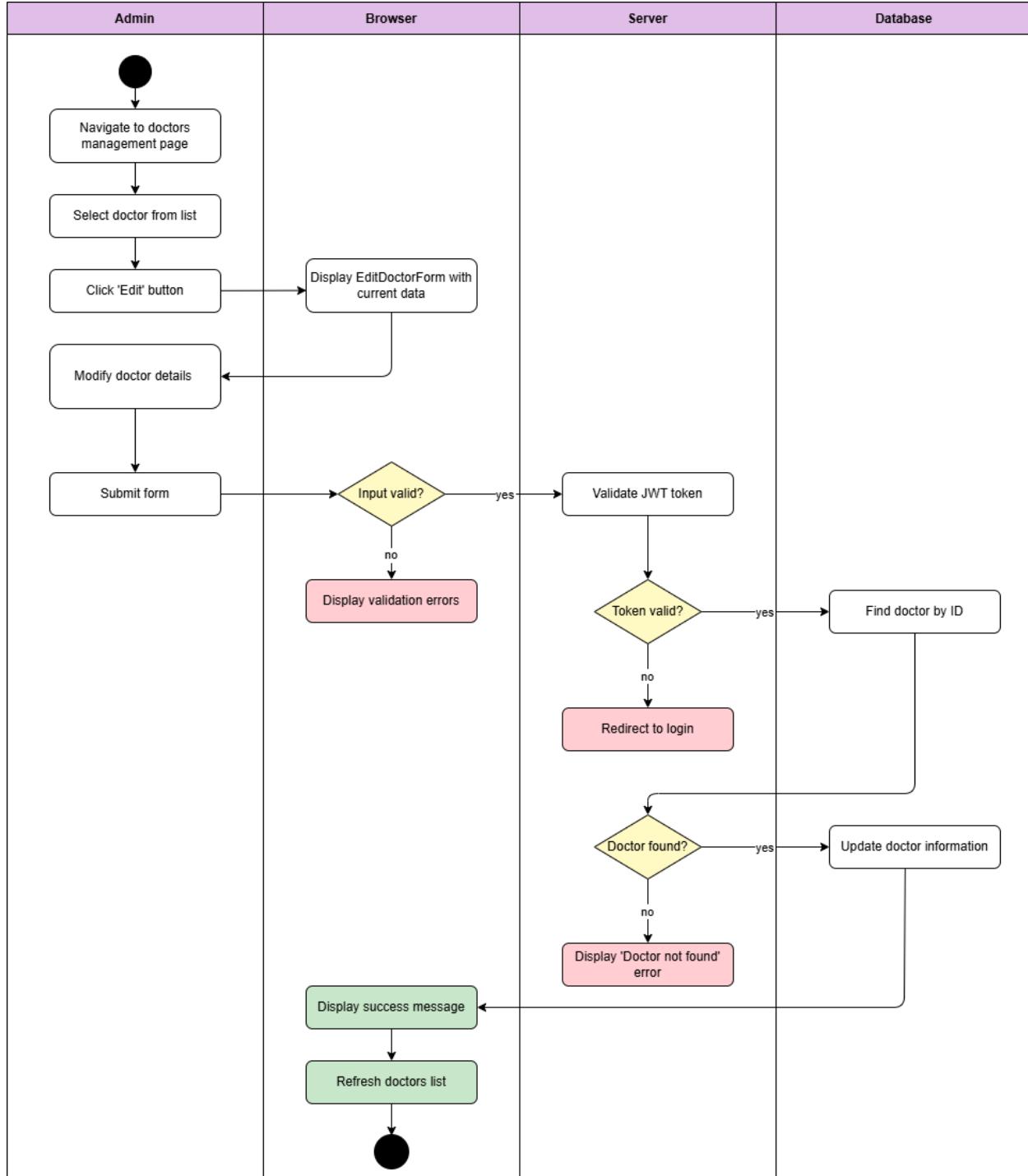


Diagram 70, Activity Diagram (Update Clinicians Account)

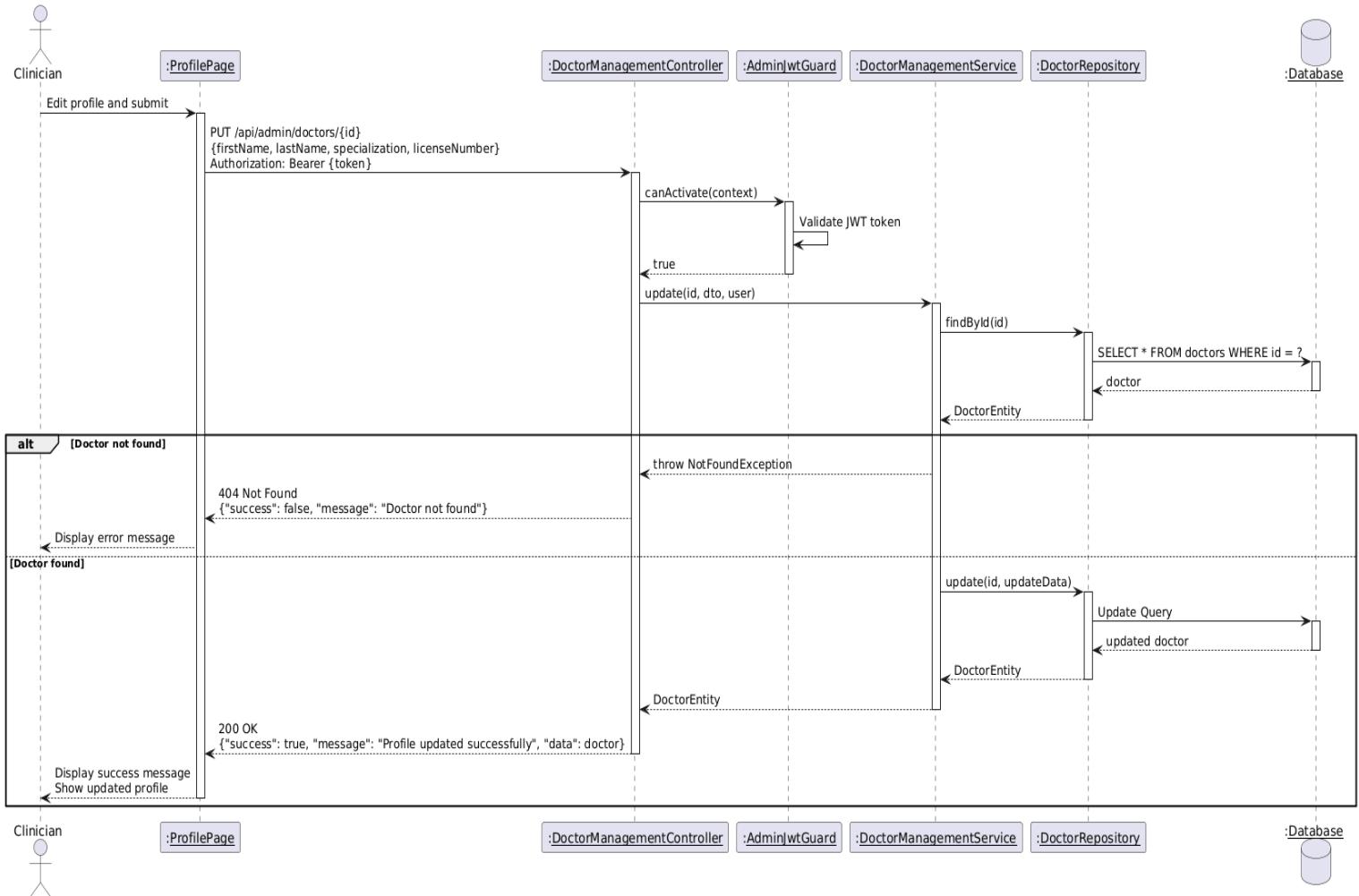


Diagram 71, Sequence Diagram (Update Clinicians Account)

- **View Clinicians Account List:**

Use case ID	VEMR-FR-CM-34
Use case name	View clinicians account List
Description	The system allows administrators to view a paginated list of all clinicians Account with search and filter capabilities
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<p>1.Admin navigates to doctor management page 2.System displays list of all doctor with: (Name - Email - Specialization - Account Status - Organization) 3.Admin can:</p> <ul style="list-style-type: none"> - Search by name, email, or specialization. - Filter by account status (active, inactive, suspended) - Filter by organization - Sort by any column - Navigate through pages
Alternative scenario	<p>A1: No Doctors Found</p> <ul style="list-style-type: none"> - At step 2, if no doctors match search/filter criteria -System displays “No doctor found” message <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 7, if JWT token is invalid or expired -System redirects to login page
Post condition	Doctor list is displayed with current data

Table 41, Use Case Specification (View Clinicians Account List)

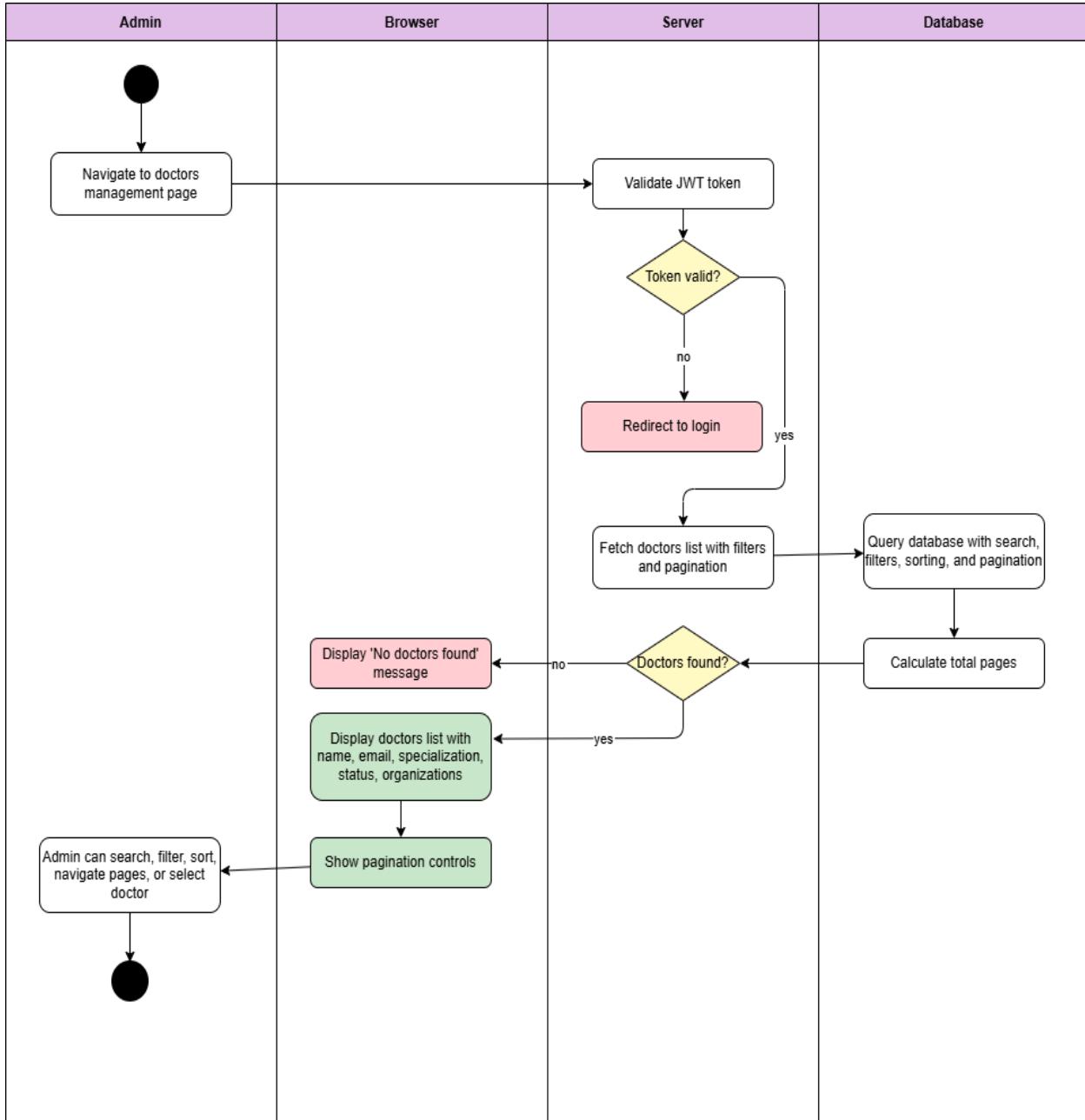


Diagram 72, Activity Diagram (View Clinicians Account List)

Chapter 4 - System Analysis

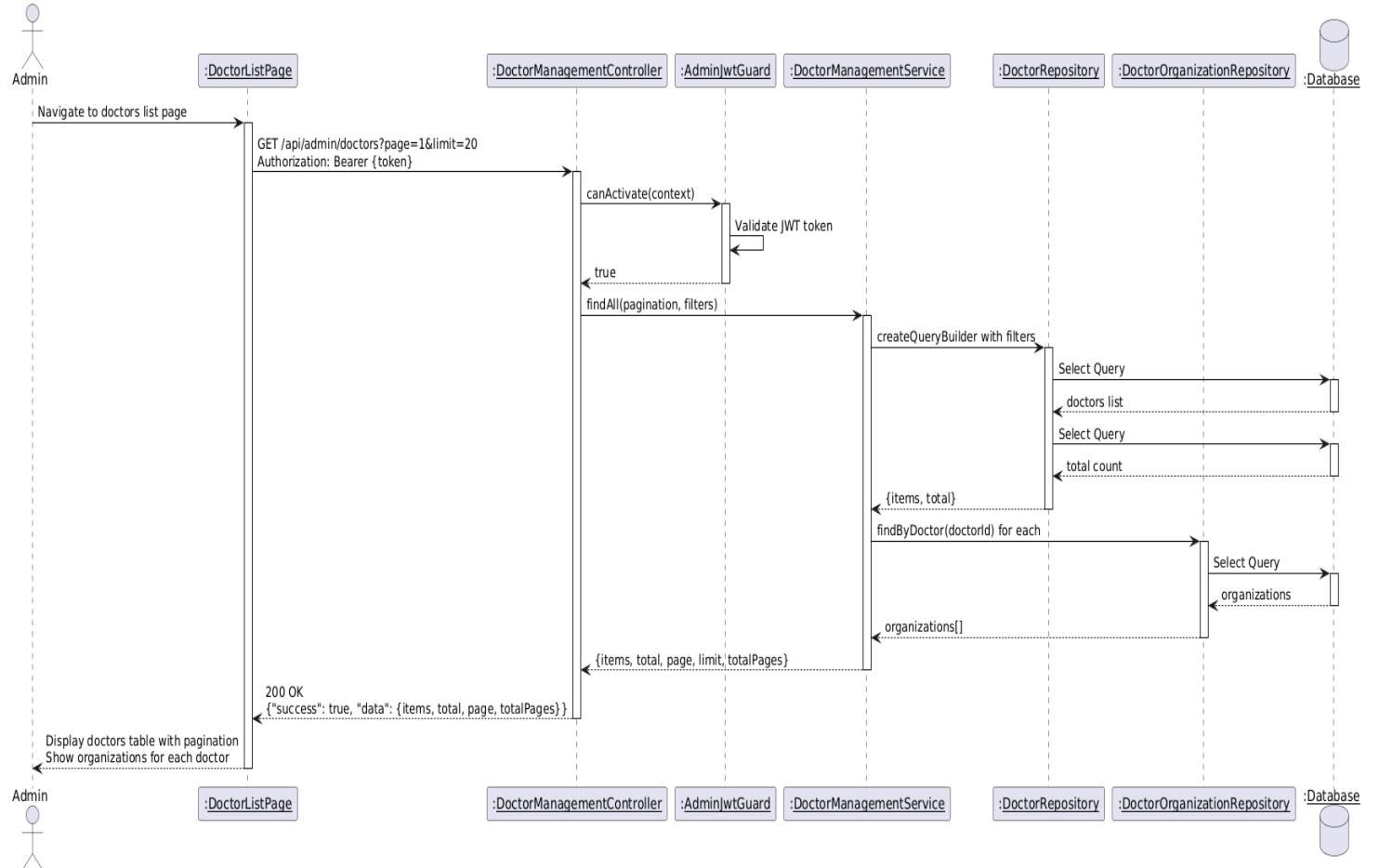


Diagram 73, Sequence Diagram (View Clinicians Account List)

- **View Clinicians Account Details:**

Use case ID	VEMR-FR-CM-35
Use case name	View clinicians account Details
Description	The system allows administrators to view details information about a specific clinician account
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<p>1.Admin navigates to doctor management page 2.Admin select a doctor from the list 3.Admin click “View Details “button or click on doctor name 4.System displays doctor details page with:</p> <ul style="list-style-type: none"> - Personal information - Account status - Organization - Account activity - Statistics
Alternative scenario	<p>A1: Doctor Not Found</p> <ul style="list-style-type: none"> - At step 4, if doctor ID is invalid -System displays “Doctor Not Found” error and redirects to list <p>A2: Authentication Error</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired -System redirects to login page
Post condition	Doctor details are displayed

Table 42, Use Case Specification (View Clinicians Account Details)

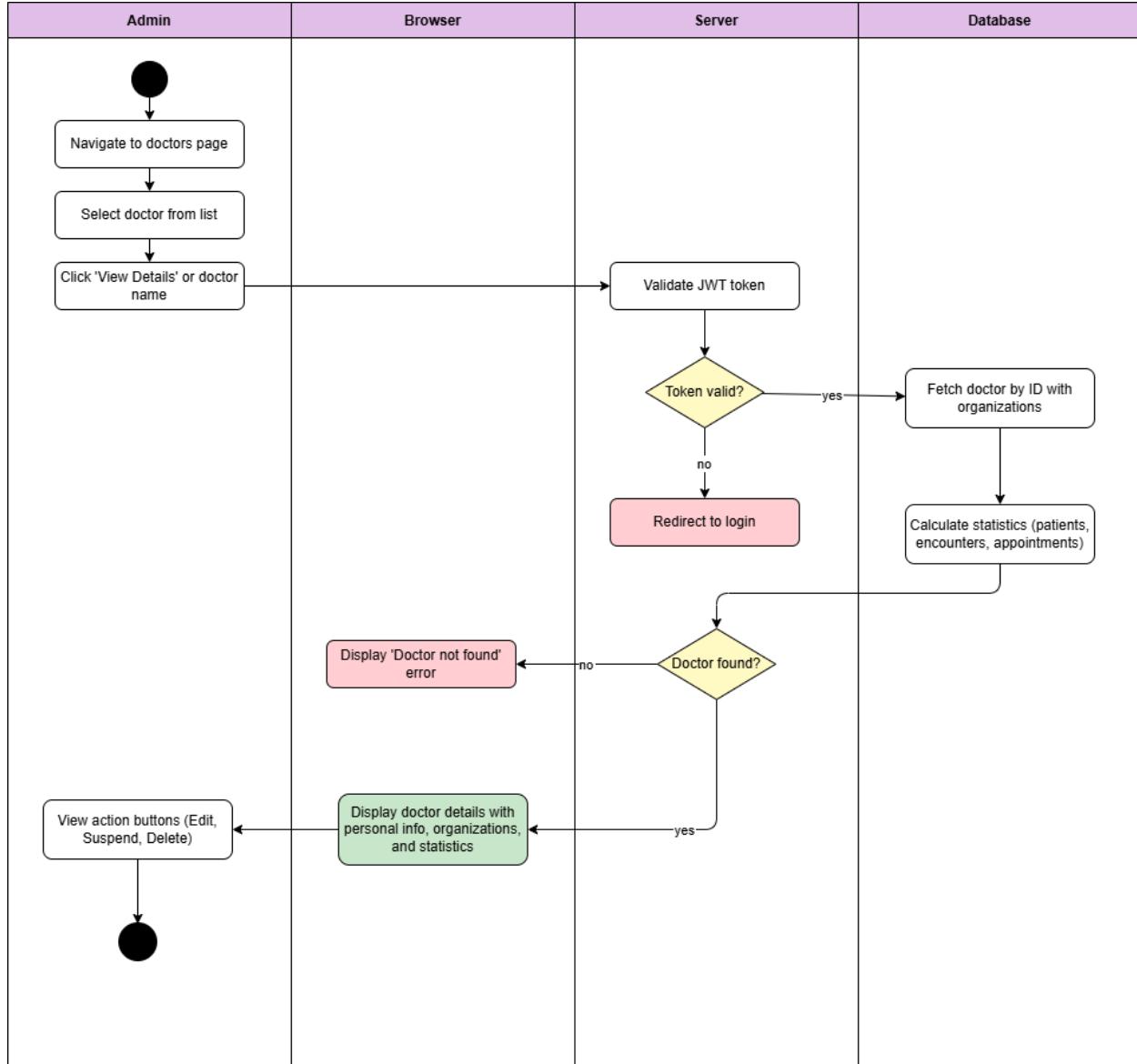


Diagram 74, Activity Diagram (View Clinicians Account Details)

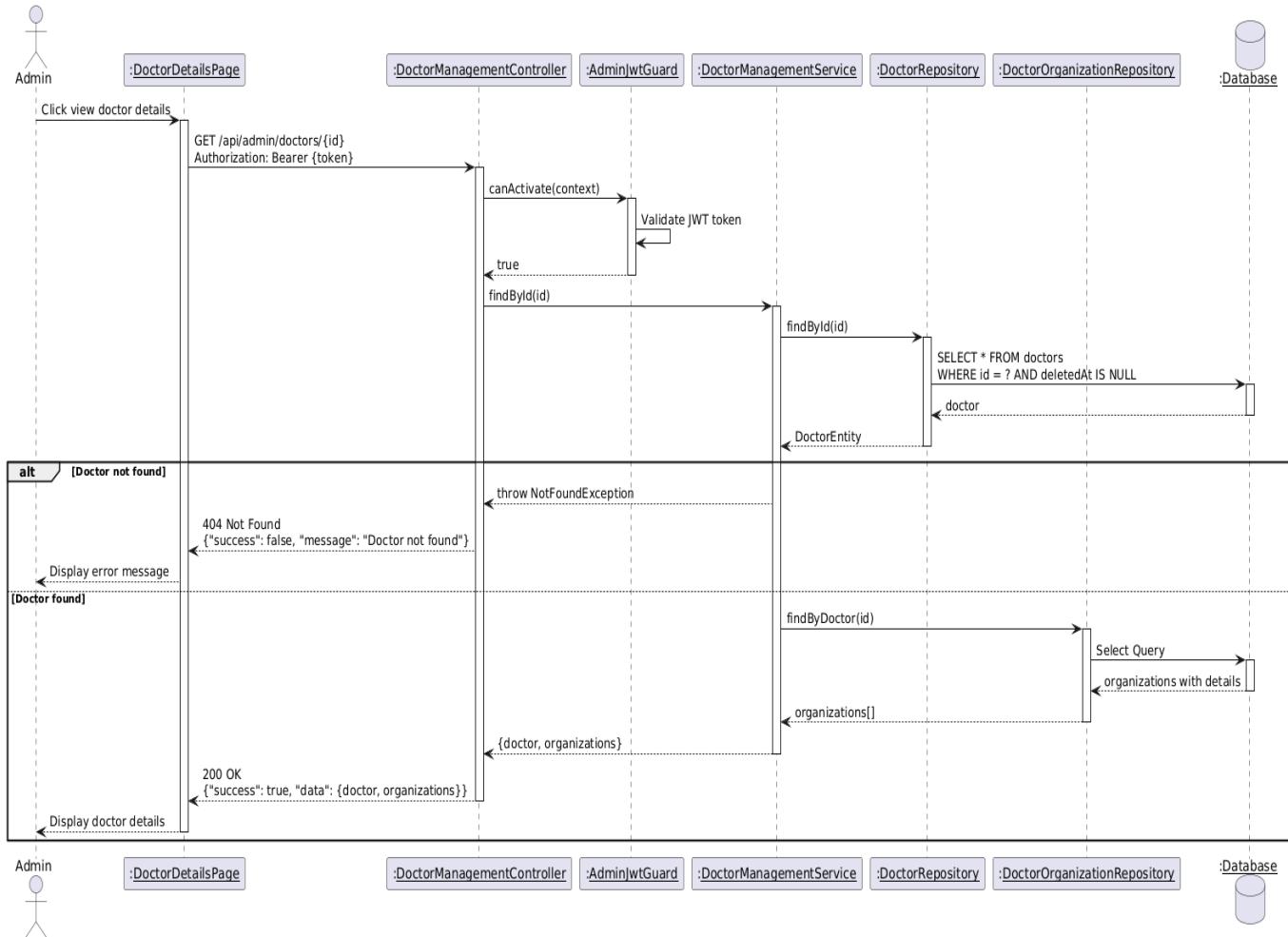


Diagram 75, Sequence Diagram (View Clinicians Account List)

● Delete Clinicians Account:

Use case ID	VEMR-FR-CM-36
Use case name	Delete clinicians account
Description	The system allows administrators to delete a clinician account
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<p>1.Admin navigates to doctor management page 2.Admin selects a doctor from the list 3.Admin click "Delete" button 4.System displays confirmation dialog with warning message 5.Admin confirm deletion 6.System validates that doctor has no active appointments 7.System performs soft delete (Set deleted at timestamp) 8.System displays success message 9.Doctor is removed from the list</p>
Alternative scenario	<p>A1: Doctor Has Active Appointments</p> <ul style="list-style-type: none"> - At step 6, if doctor has upcoming or in-progress appointments -System displays error: "Cannot delete doctor with active appointments" <p>A2: Doctor Not Found</p> <ul style="list-style-type: none"> - At step 6, if Doctor ID is invalid -System displays "Doctor Not Found error" <p>A3: Authentication Error</p> <ul style="list-style-type: none"> - At step 6, if JWT token is invalid or expired -System redirects to login page
Post condition	Doctor account is soft deleted

Table 43, Use Case Specification (Delete Clinicians Account)

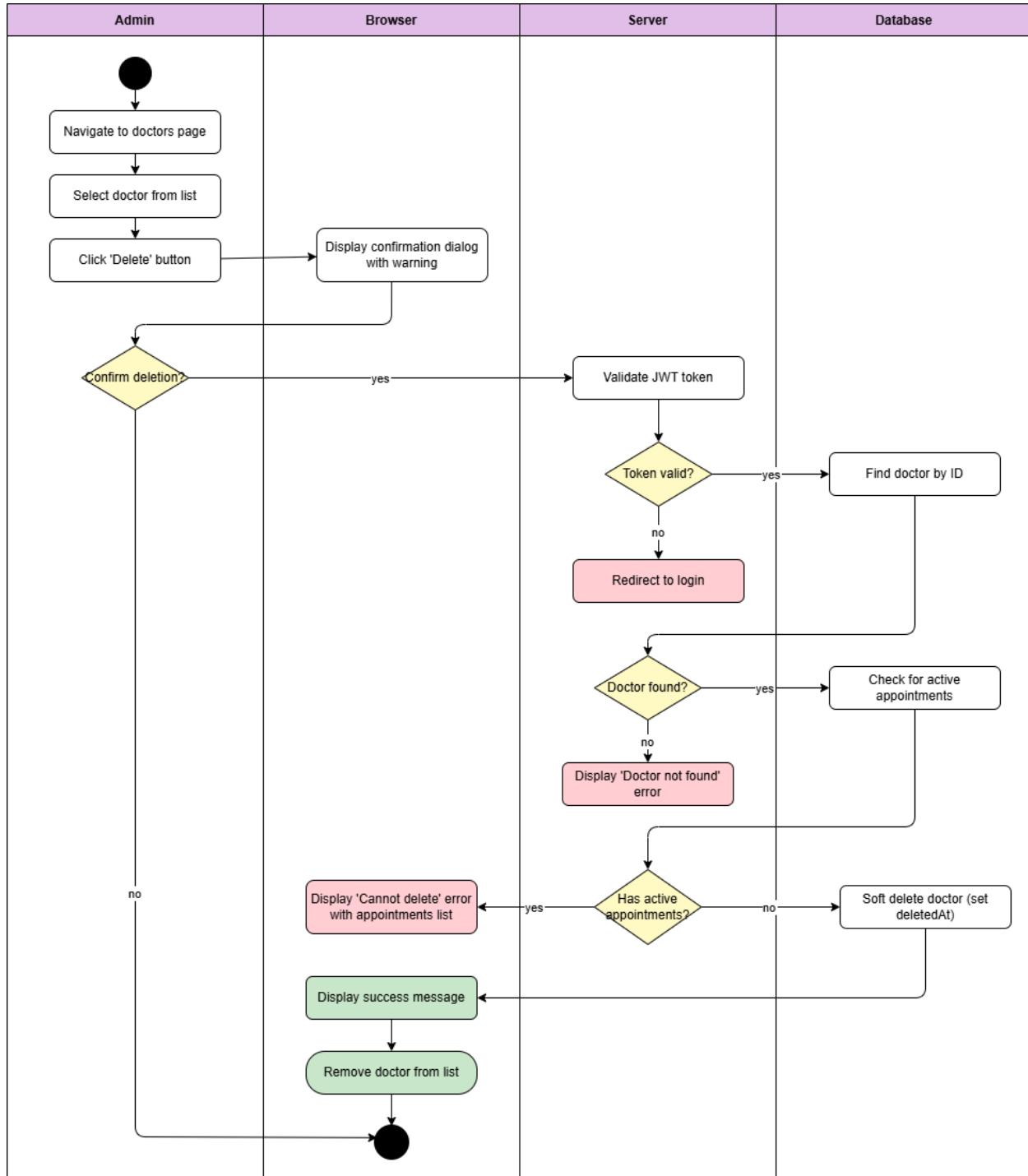


Diagram 76, Activity Diagram (Delete Clinicians Account)

Chapter 4 - System Analysis

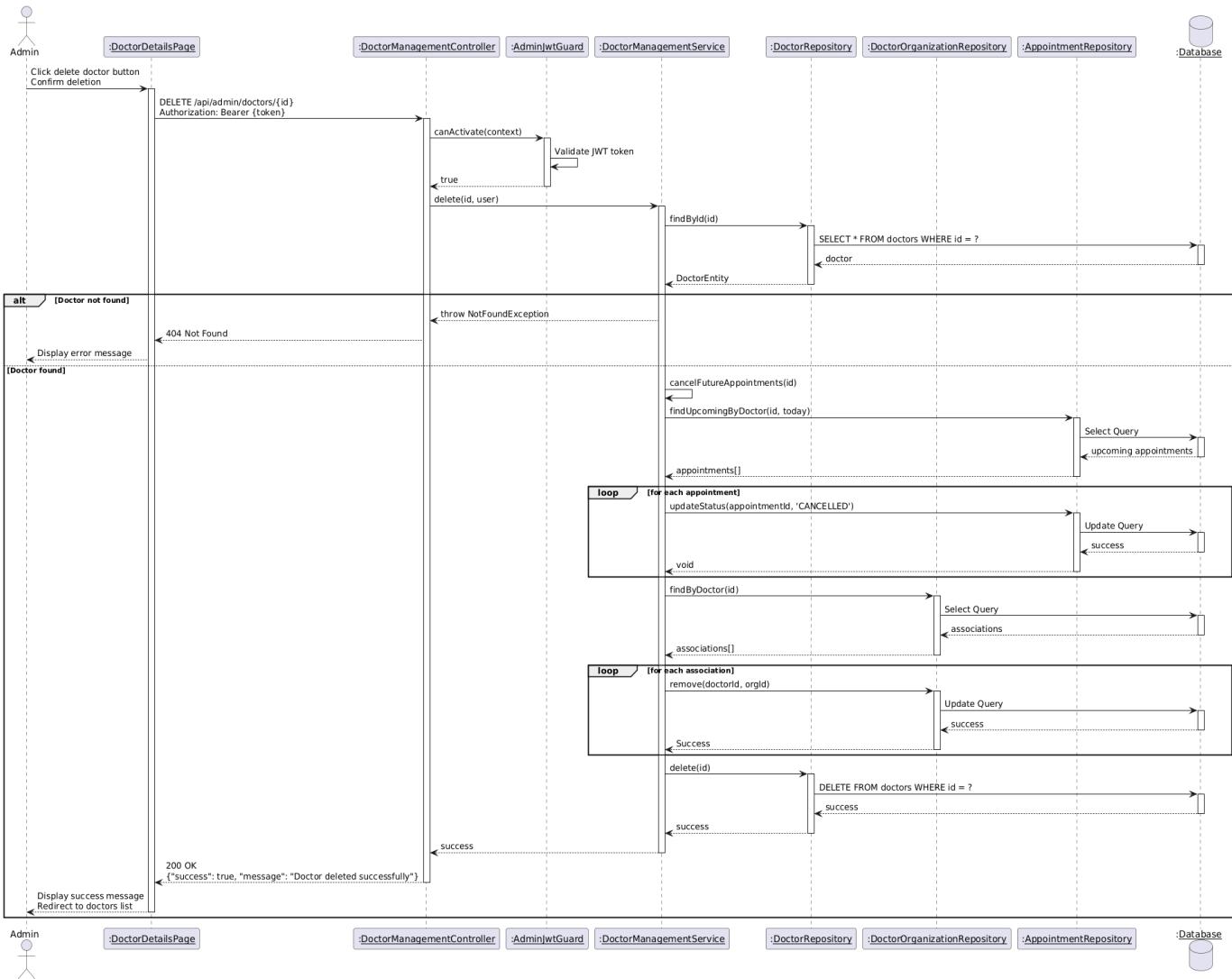


Diagram 77, Sequence Diagram (View Clinicians Account List)

● Search Clinicians Account:

Use case ID	VEMR-FR-CM-37
Use case name	Search clinicians account
Description	The system allows administrators to search for clinician accounts by name or email.
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to clinicians' management page 2. Administrator enters search query in search box 3. System debounces input (300ms delay) 4. System validates JWT token via AdminJwtGuard 5. System searches clinicians by first name, last name, or email via Doctor Repository 6. System returns paginated search results 7. System displays matching clinician accounts in table
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired -System returns 401 Unauthorized <p>A2: No Results Found</p> <ul style="list-style-type: none"> - At step 5, if no clinicians match the search query -System displays "No clinicians found" message <p>A3: Empty Search Query</p> <ul style="list-style-type: none"> - At step 3, if search query is empty or cleared -System returns all clinicians with pagination
Post condition	Administrator can view search results with pagination

Table 44, Use Case Specification (Search Clinicians Account)

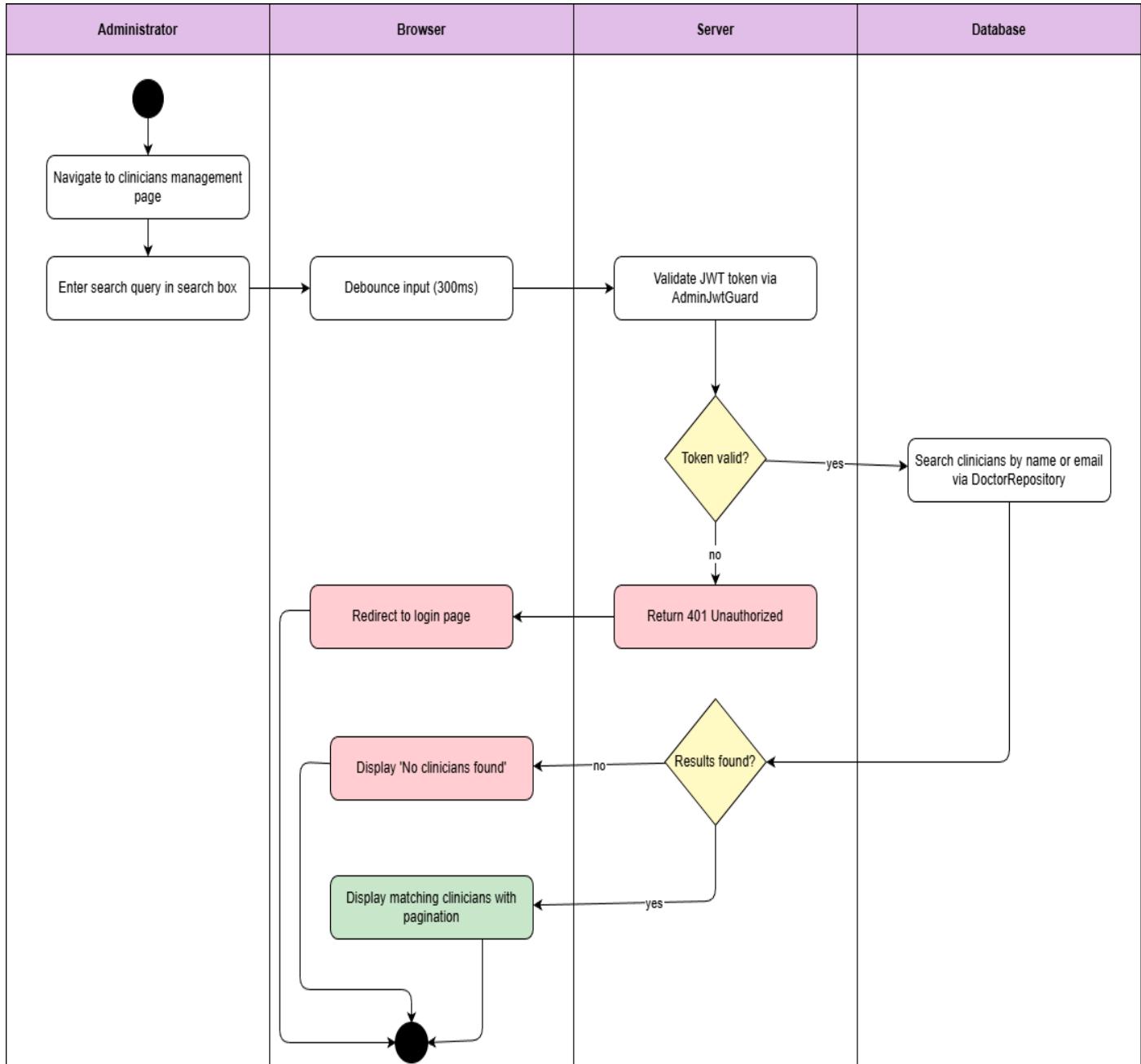


Diagram 78, Activity Diagram (Search Clinicians Account List)

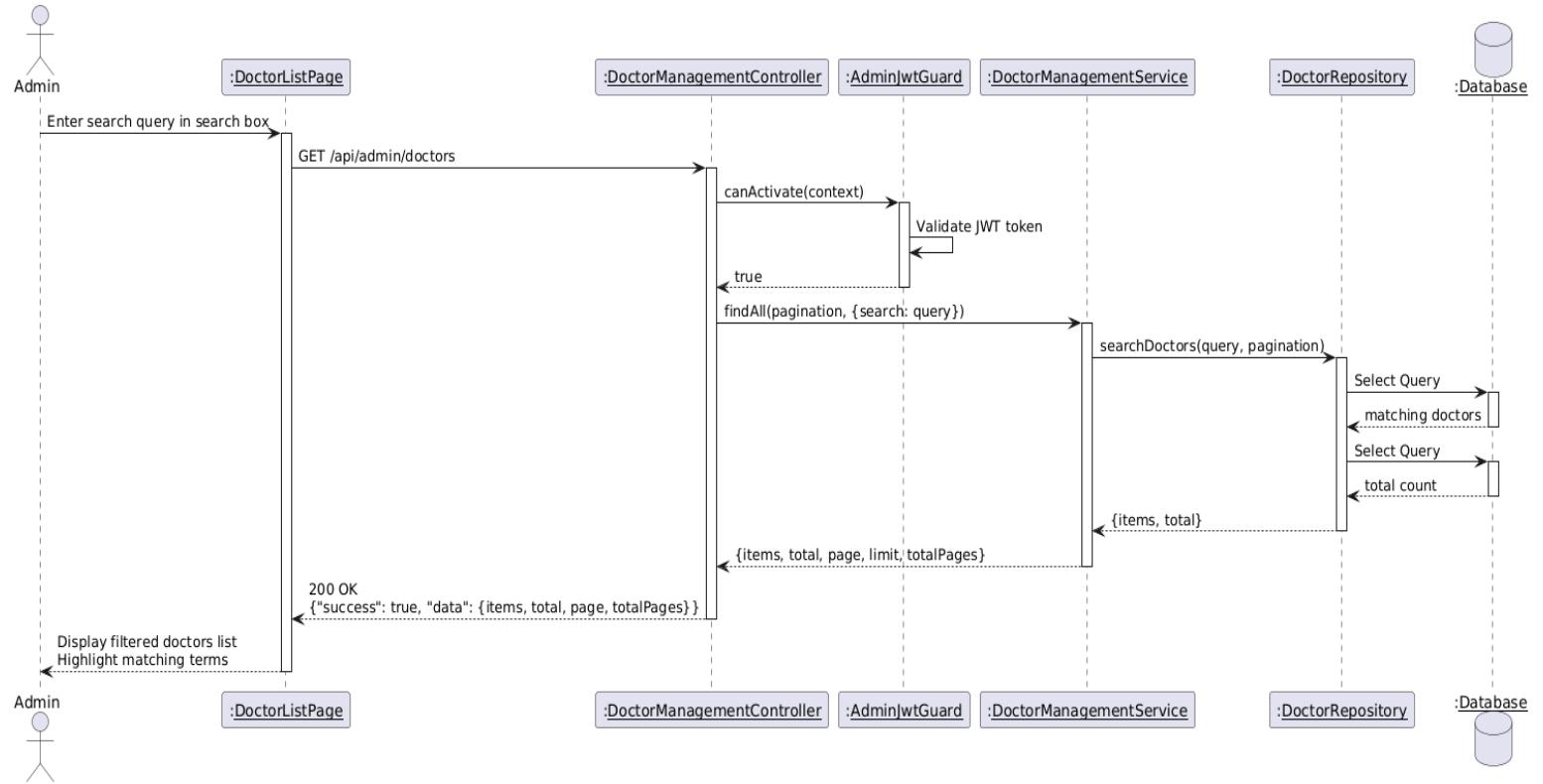


Diagram 79, Sequence Diagram (Search Clinicians Account List)

● Create Organization:

Use case ID	VEMR-FR-OM-38
Use case name	Create organization
Description	The system allows administrators to create a new healthcare organization.
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to organizations management page 2. Administrator clicks "Create Organization" button 3. System displays organization creation form 4. Administrator enters organization details (name, address, phone, email) 5. Administrator submits form 6. System validates input fields 7. System validates JWT token 8. System checks for duplicate organization name 9. System creates new organization record 10. System displays success message and refreshes organization list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 7, if JWT token is invalid or expired <p>System returns 401 Unauthorized</p> <p>A2: Validation Error</p> <ul style="list-style-type: none"> - At step 6, if required fields are empty or invalid <p>-System displays validation error messages</p> <p>A3: Duplicate Organization</p> <ul style="list-style-type: none"> - At step 8, if organization name already exists <p>-System displays error message</p>
Post condition	New organization is created in the system

Table 45, Use Case Specification (Create Organization)

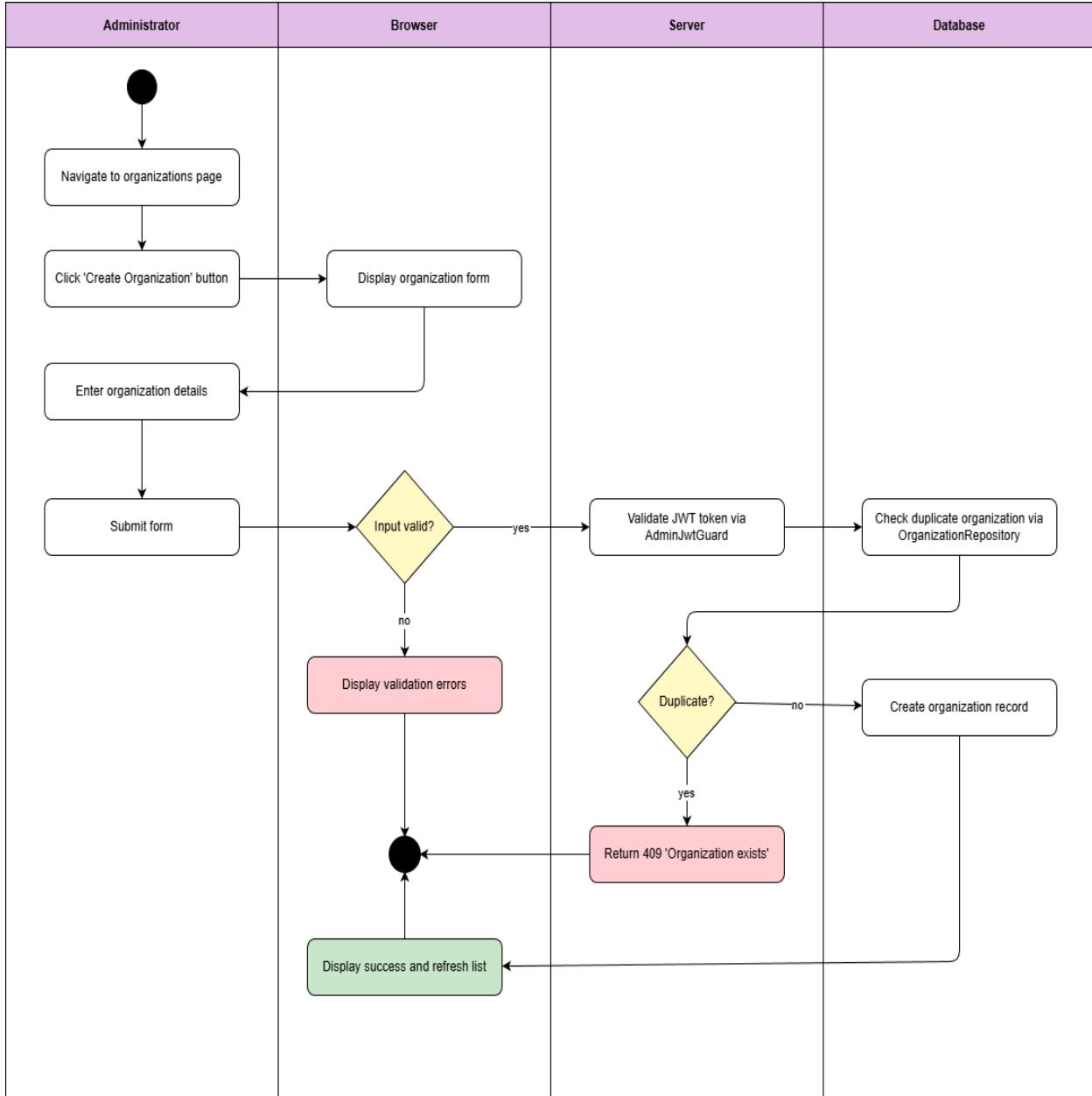


Diagram 80, Activity Diagram (Create Organization)

Chapter 4 - System Analysis

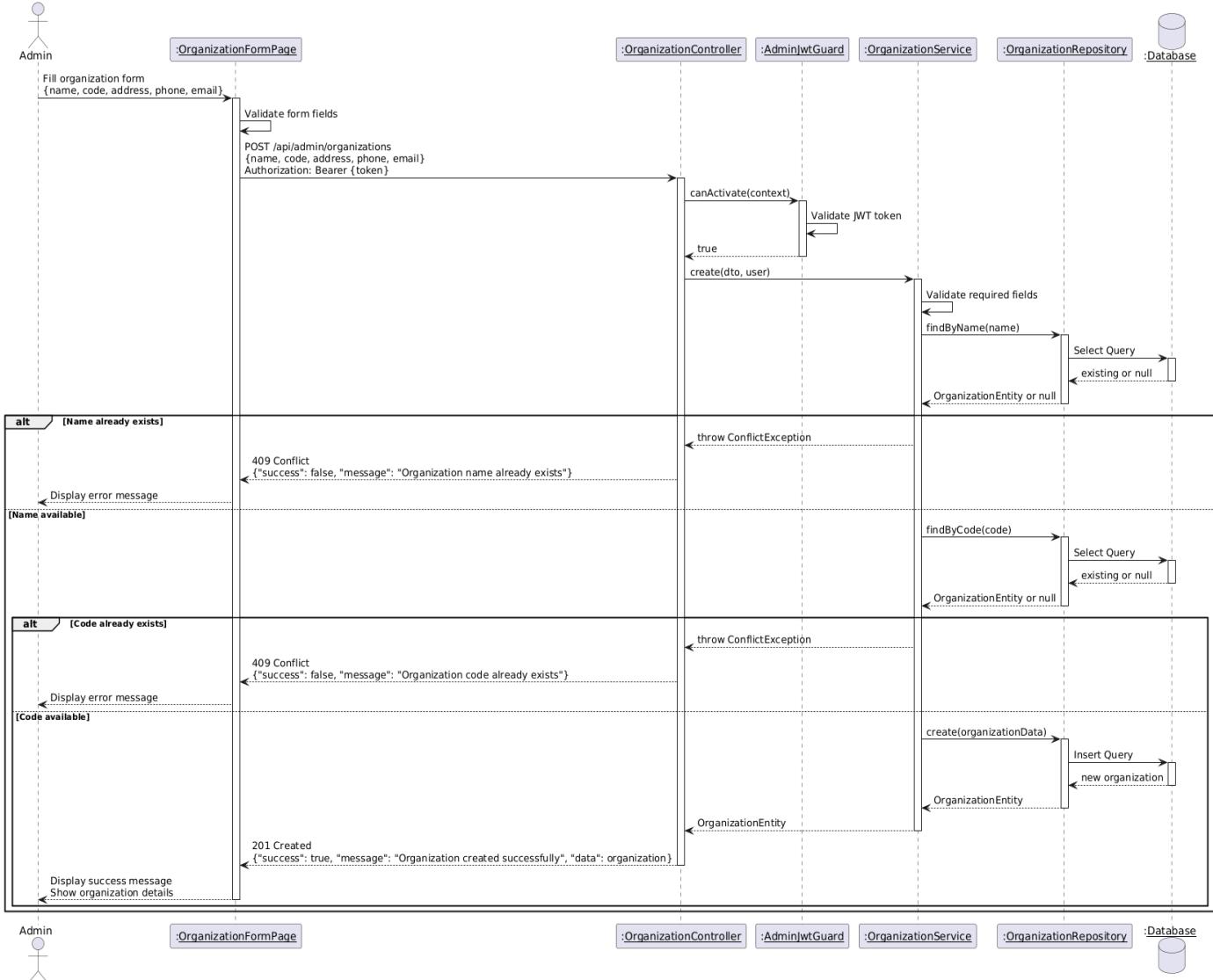


Diagram 81, Sequence Diagram (Create Organization)

● Update Organization:

Use case ID	VEMR-FR-OM-39
Use case name	Update organization
Description	The system allows administrators to update organization information.
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to organizations list page 2. Administrator clicks "Edit" button on organization row 3. System displays organization edit form with current data 4. Administrator modifies organization details 5. Administrator submits form 6. System validates input fields 7. System validates JWT token 8. System updates organization record 9. System displays success message and refreshes organization list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 7, if JWT token is invalid or expired <p>System returns 401 Unauthorized</p> <p>A2: Validation Error</p> <ul style="list-style-type: none"> - At step 6, if required fields are empty or invalid <p>-System displays validation error messages</p> <p>A3: Organization Not Found</p> <ul style="list-style-type: none"> - At step 8, if organization ID does not exist <p>-System displays error message</p>
Post condition	Organization information is updated in the system

Table 46, Use Case Specification (Update Organization)

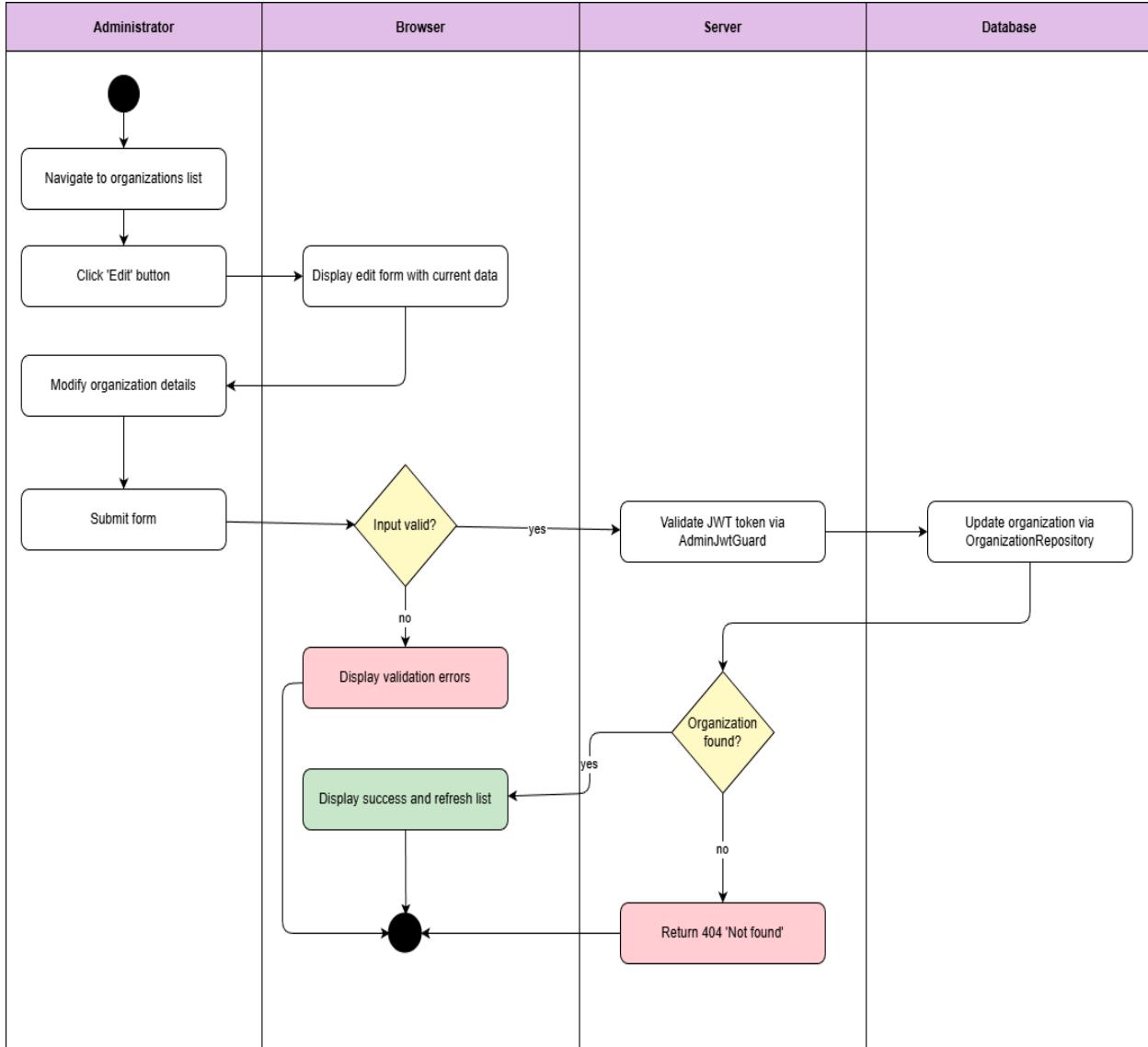


Diagram 82, Activity Diagram (Update Organization)

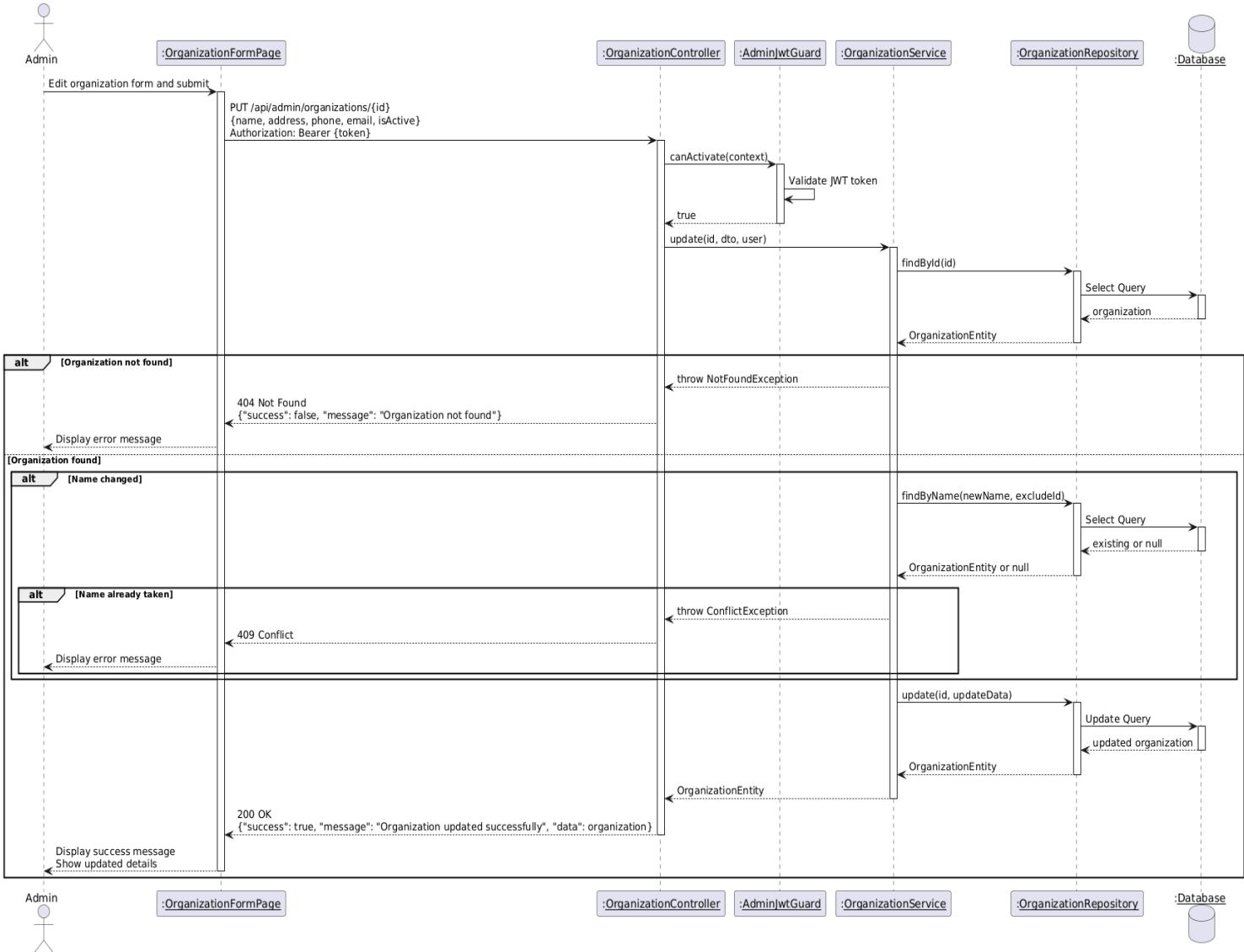


Diagram 83, Sequence Diagram (Update Organization)

● View Organization:

Use case ID	VEMR-FR-OM-40
Use case name	View organization
Description	The system allows administrators to view all healthcare organizations with pagination
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to organizations management page 2. System validates JWT token 3. System retrieves organizations list 4. System returns paginated organizations with total count 5. System displays organizations in table with pagination controls
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired System returns 401 Unauthorized <p>A2: No Organizations Found</p> <ul style="list-style-type: none"> - At step 3, if no organizations exist in the system -System displays "No organizations found" message
Post condition	Organization's list is displayed with pagination

Table 47, Use Case Specification (View Organization)

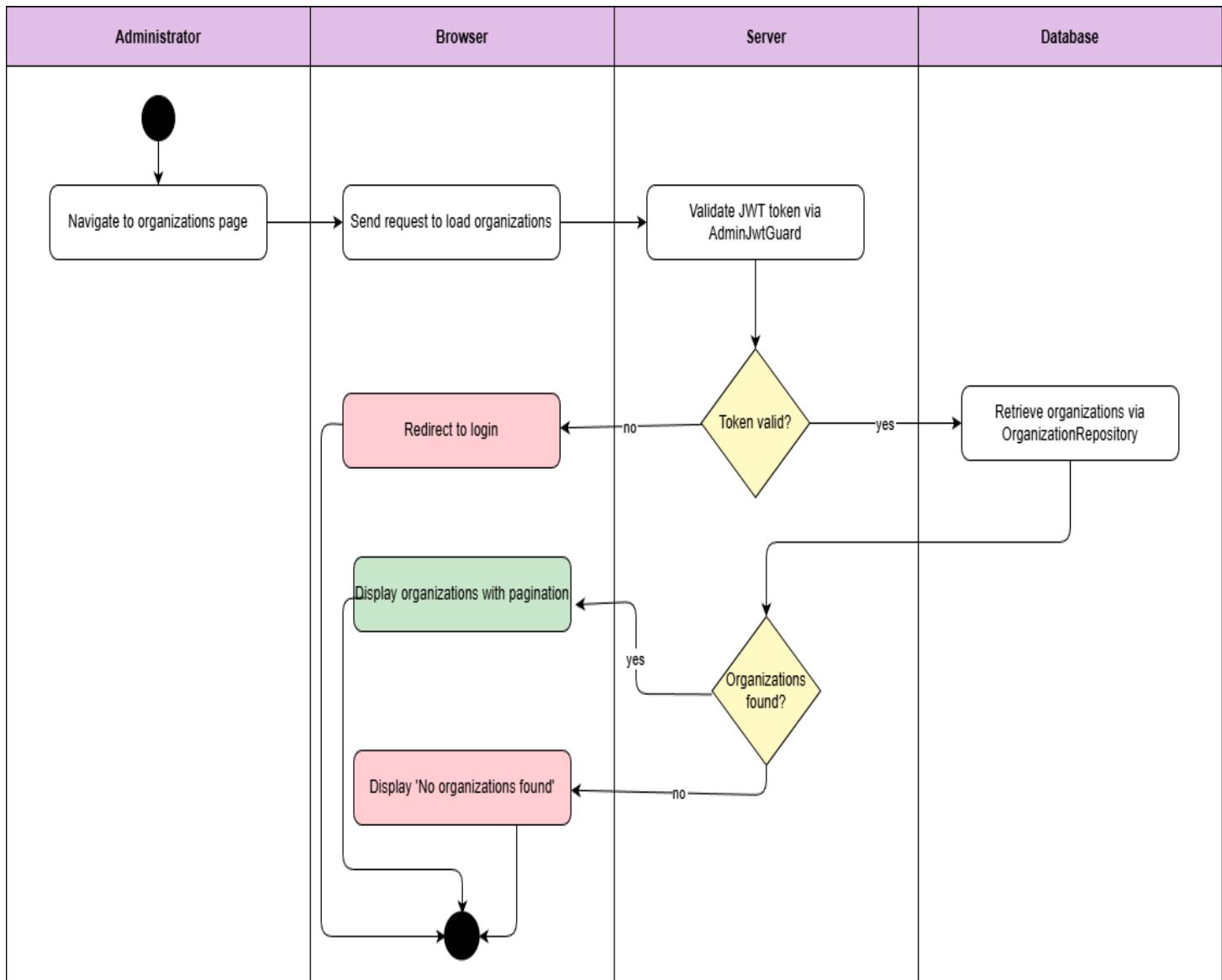


Diagram 84, Activity Diagram (View Organization)

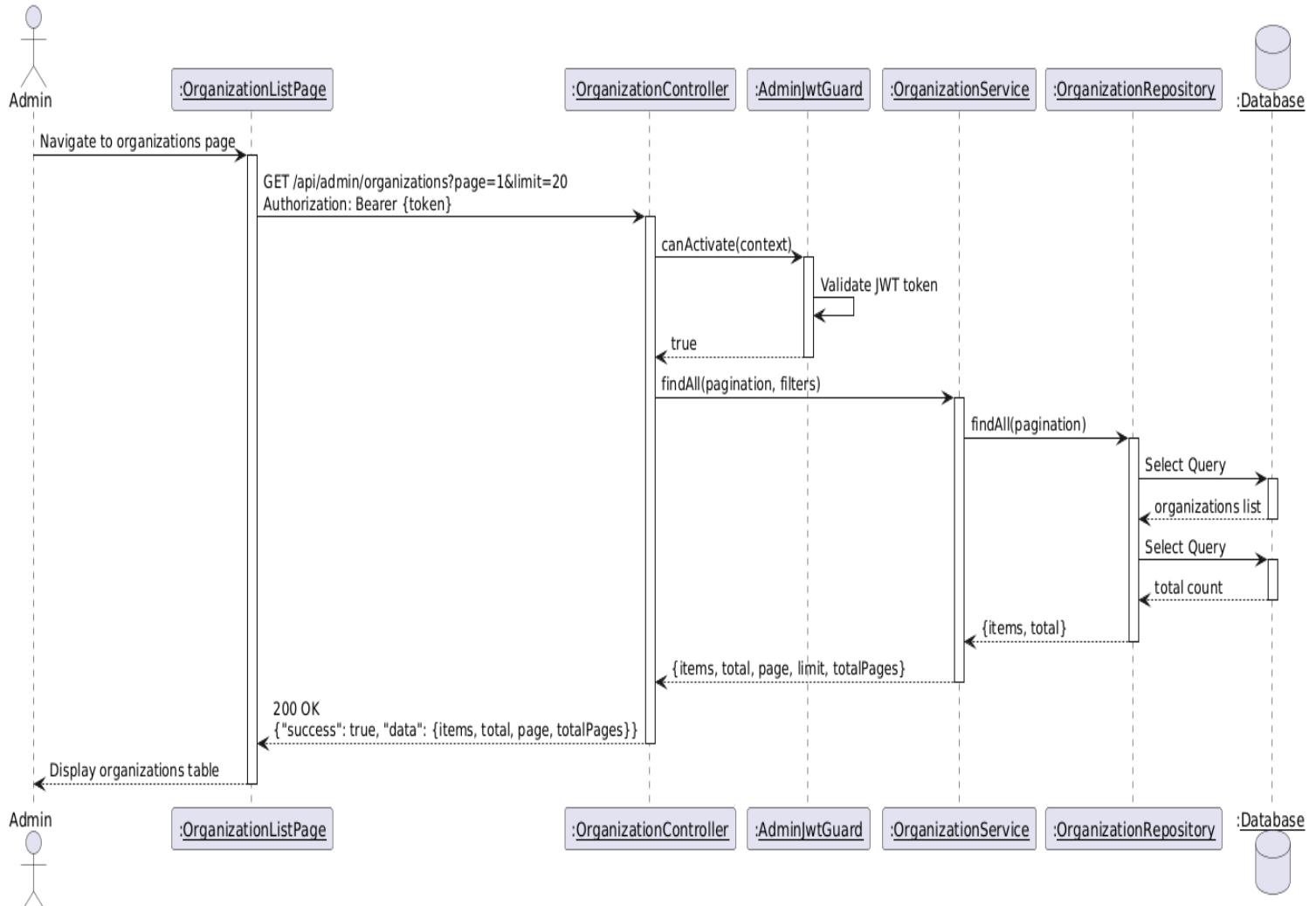


Diagram 85, Sequence Diagram (View Organization)

- **View Organization Details:**

Use case ID	VEMR-FR-OM-41
Use case name	View organization Details
Description	The system allows administrators to view detailed information about a specific organization.
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to organizations list page 2. Administrator clicks on organization name to view details 3. System validates JWT token 4. System retrieves organization data 5. System retrieves associated clinicians via Doctor Repository 6. System displays organization details with clinicians list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired System returns 401 Unauthorized <p>A2: No Organizations Found</p> <ul style="list-style-type: none"> - At step 4, if organization ID does not exist -System displays error message <p>A3: No Clinicians</p> <ul style="list-style-type: none"> - System displays empty state message -System displays "No organizations found" message
Post condition	Organization details are displayed with complete information

Table 48, Use Case Specification (View Organization Details)

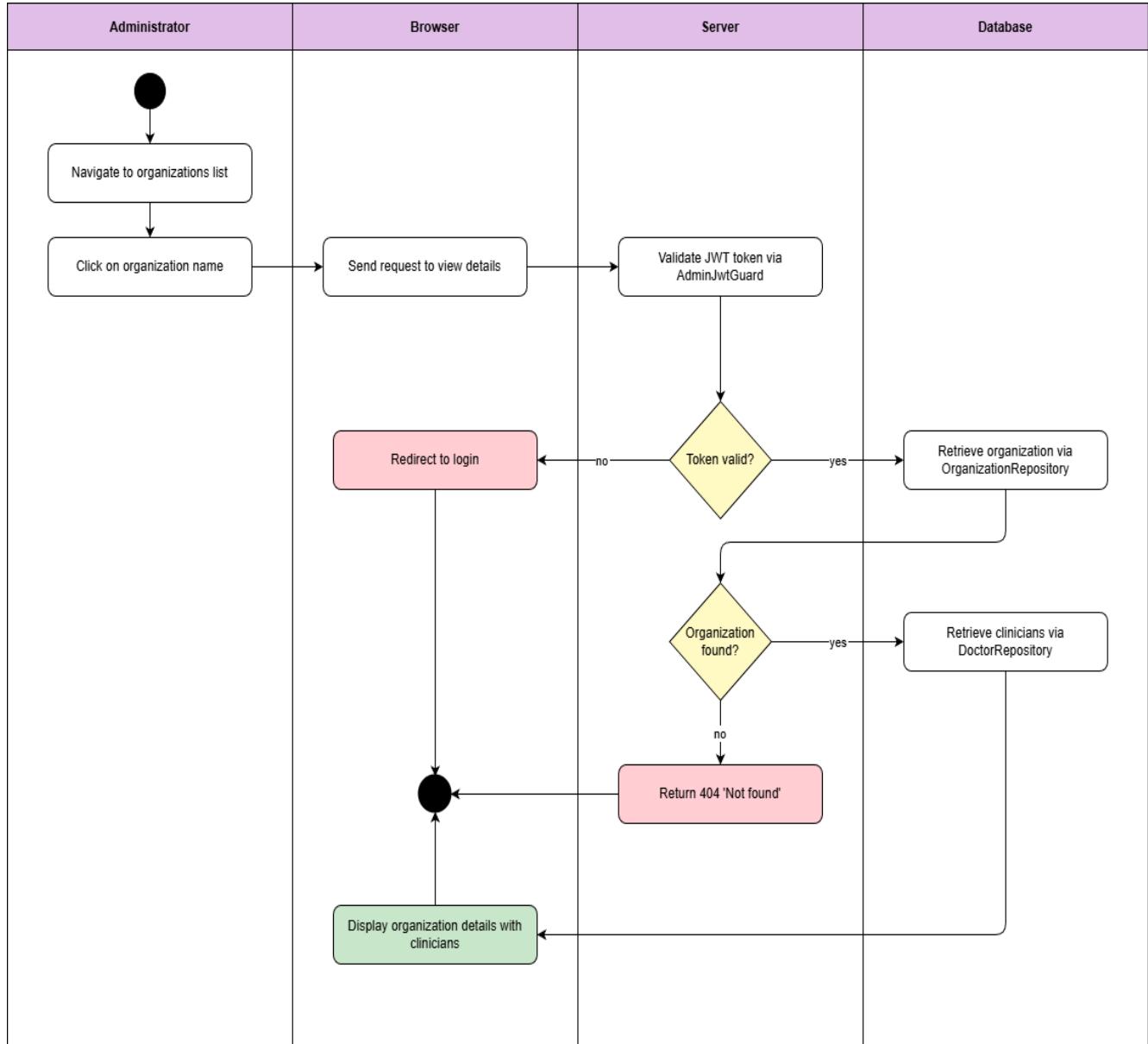


Diagram 86, Activity Diagram (View Organization Details)

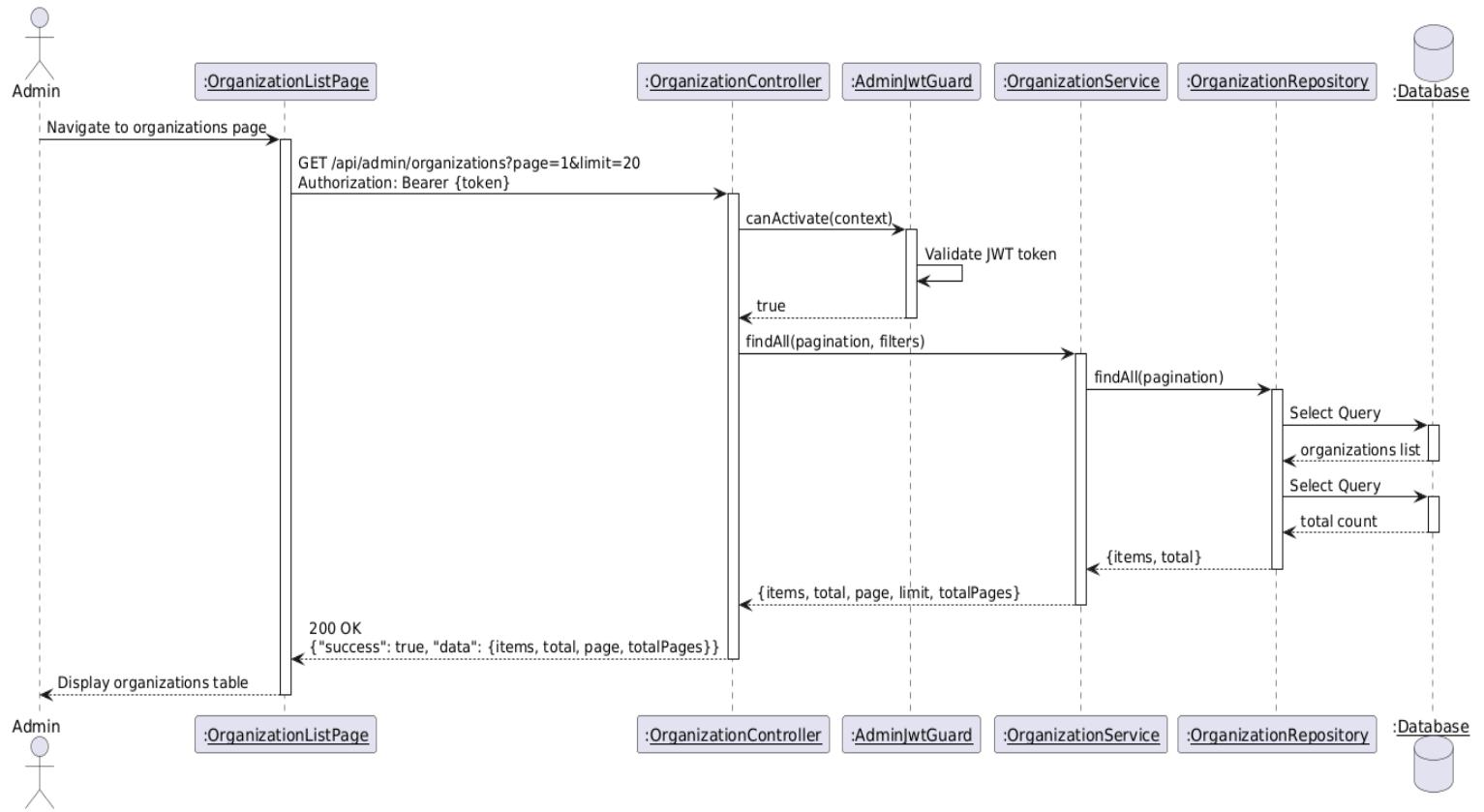


Diagram 87, Sequence Diagram (View Organization Details)

- **Delete Organization:**

Use case ID	VEMR-FR-OM-42
Use case name	Delete Organization
Description	The system allows administrators to delete a healthcare organization from the system
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to organizations list page 2. Administrator clicks "Delete" button on organization row 3. System displays confirmation dialog 4. Administrator confirms deletion 5. System validates JWT token 6. System checks for associated clinicians 7. System deletes organization record 8. System displays success message and refreshes organization list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 5, if JWT token is invalid or expired - System returns 401 Unauthorized <p>A2: Organization Not Found</p> <ul style="list-style-type: none"> - At step 7, if organization ID does not exist - System displays error message <p>A3: Organization Has Clinicians</p> <ul style="list-style-type: none"> - At step 6, if organization has associated clinicians - System displays error message
Post condition	Organization is removed from the system

Table 49, Use Case Specification (Delete Organization)

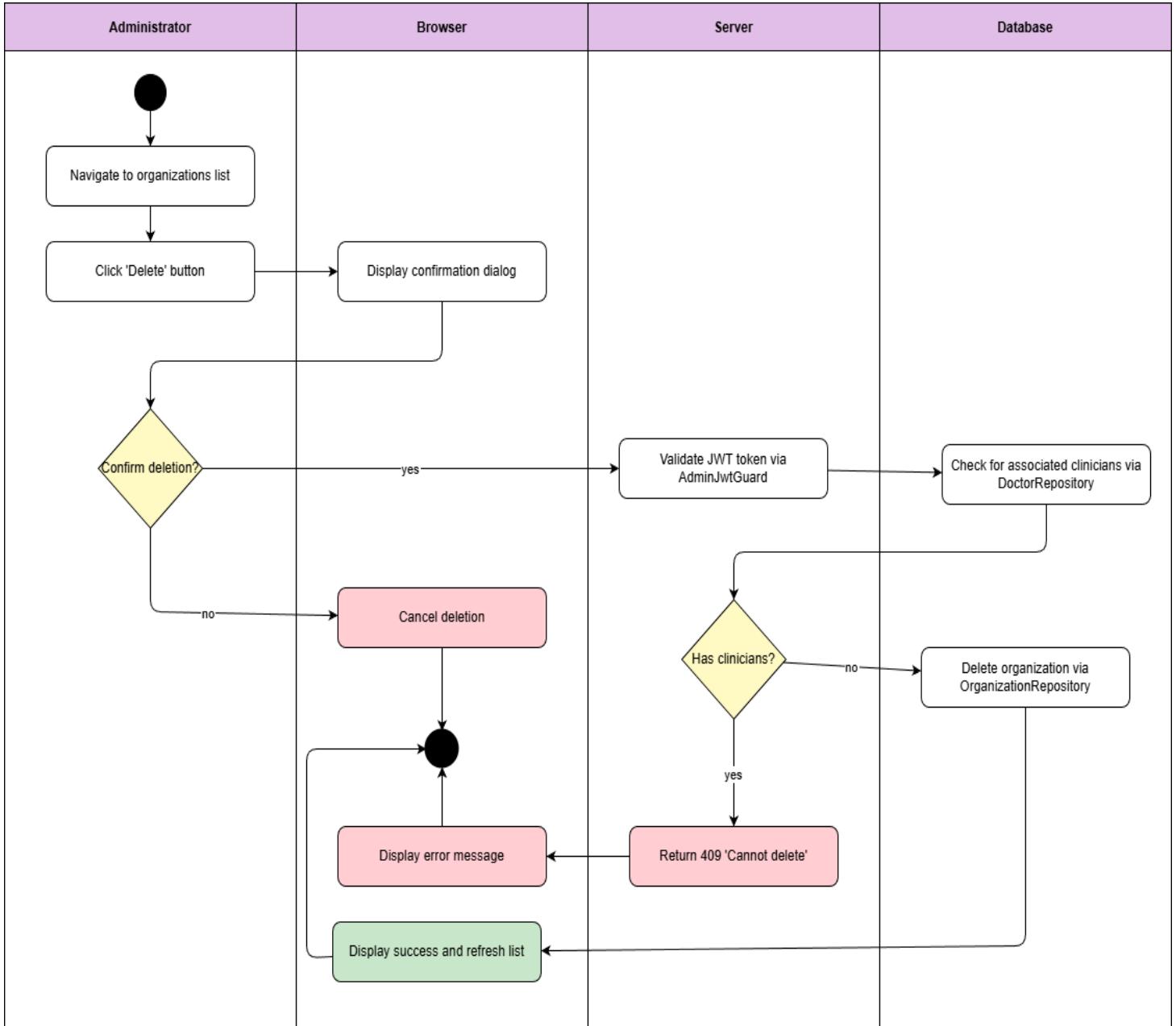


Diagram 88, Activity Diagram (Delete Organization)

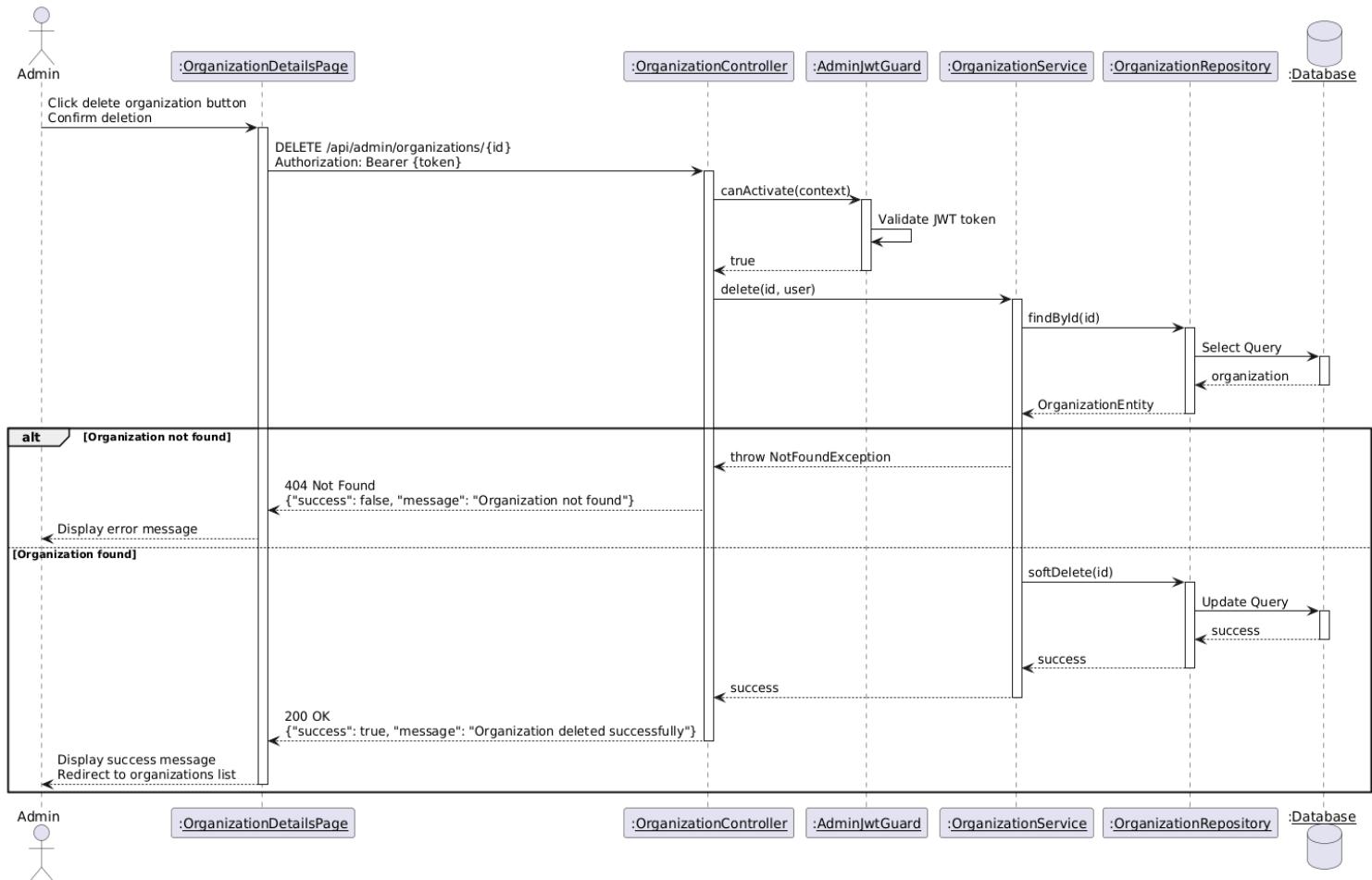


Diagram 89, Sequence Diagram (Delete Organization)

● Search Organization:

Use case ID	VEMR-FR-OM-43
Use case name	Search Organization
Description	The system allows administrators to search for organizations by name or location.
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to organizations management page 2. Administrator enters search query in search box 3. System debounces input 4. System validates JWT token 5. System searches organizations by name or address 6. System returns paginated search results 7. System displays matching organizations in table
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired <p>System returns 401 Unauthorized</p> <p>A2: No Results Found</p> <ul style="list-style-type: none"> - At step 5, if no organizations match the search query <p>-System returns empty list</p> <p>A3: Empty Search Query</p> <ul style="list-style-type: none"> - At step 3, if search query is empty or cleared <p>- System returns all organizations with pagination</p>
Post condition	Administrator can view search results with pagination

Table 50, Use Case Specification (Search Organization)

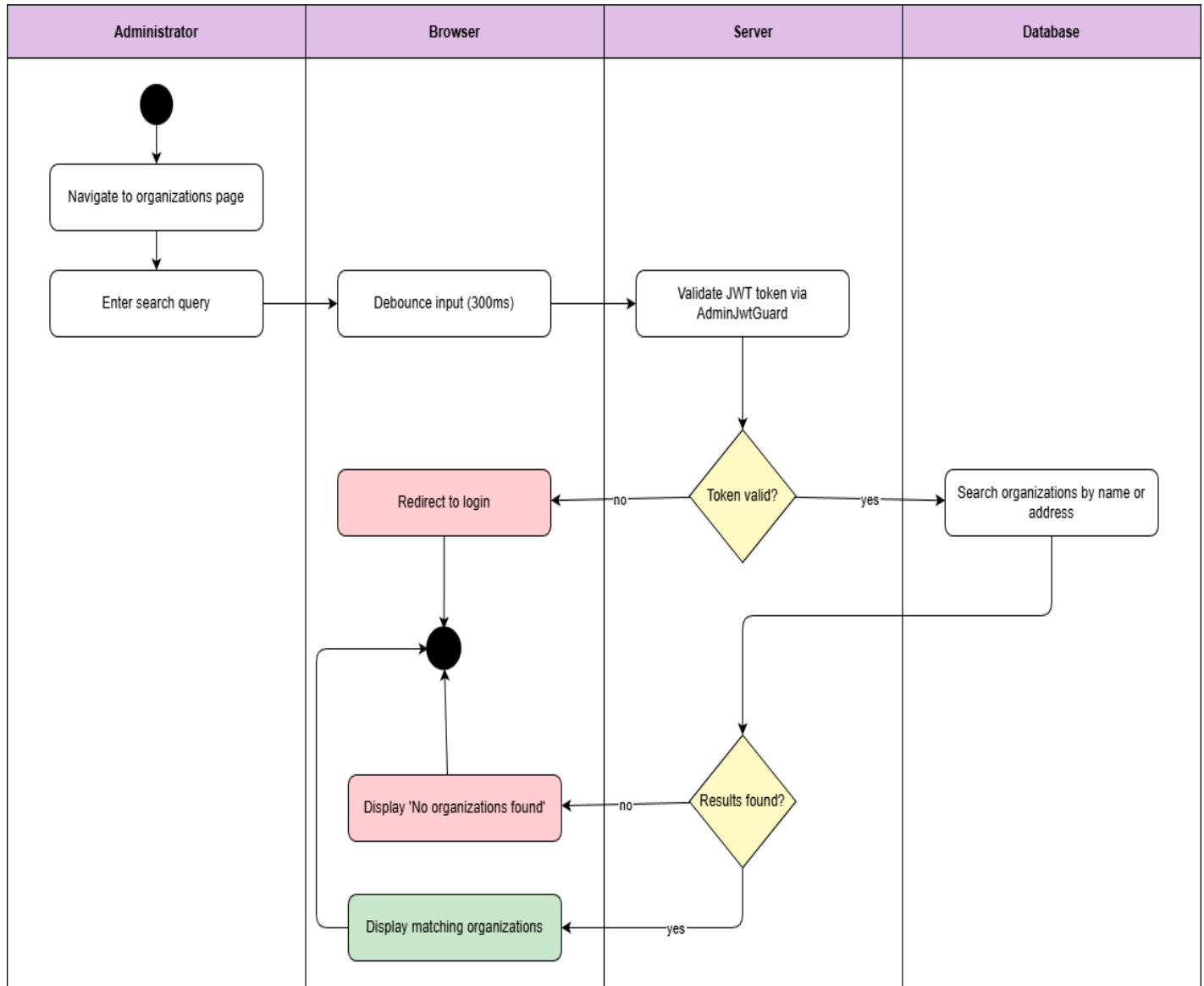


Diagram 90, Activity Diagram (Search Organization)

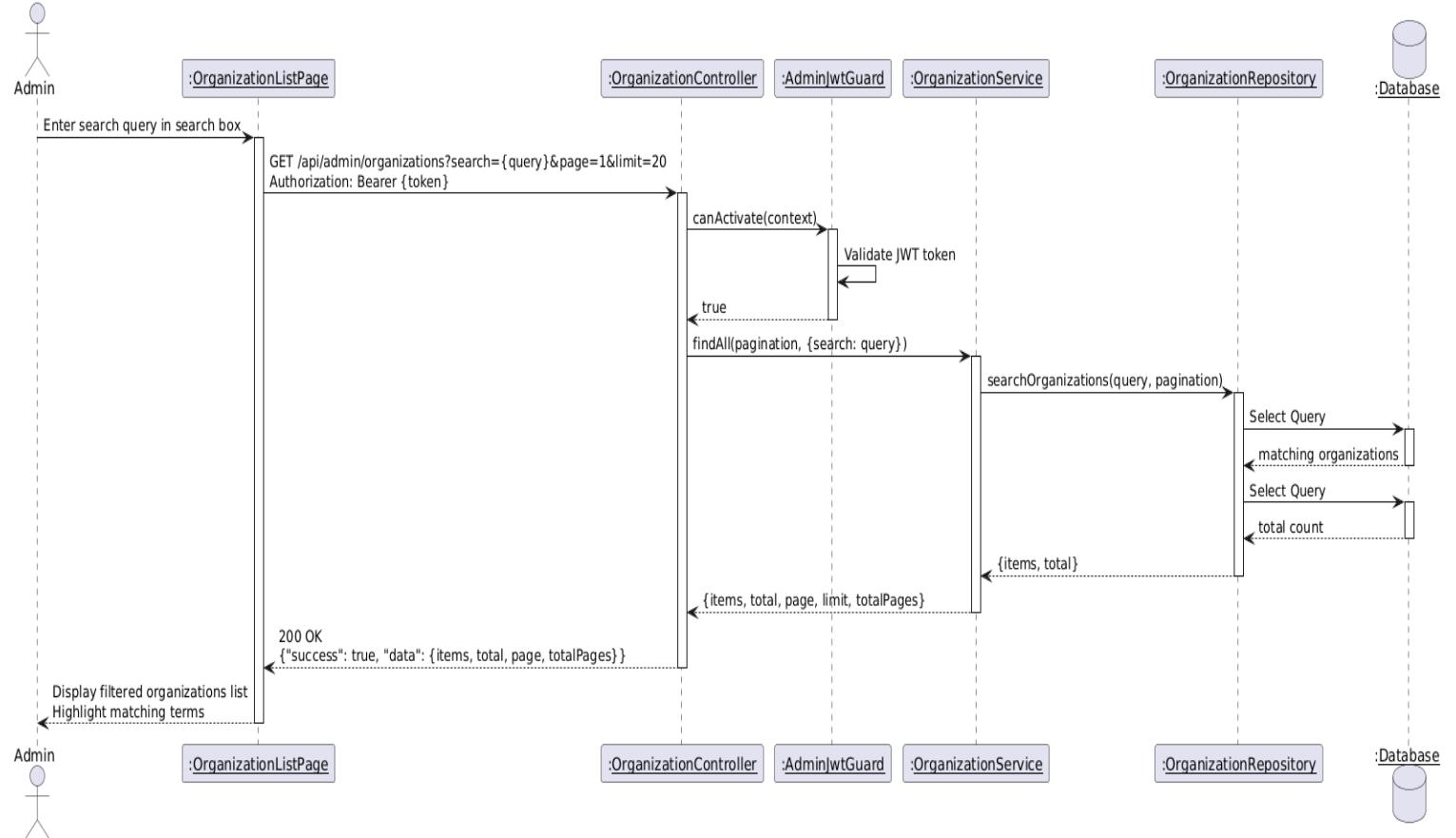


Diagram 91, Sequence Diagram (Search Organization)

- **View Admin Analytics:**

Use case ID	VEMR-FR-AN-44
Use case name	View admin analytics
Description	The system allows administrators to view system-wide analytics and statistics.
From	Admin
Pre-conditions	Admin is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Administrator navigates to admin dashboard 2. System validates JWT token 3. System retrieves total patients count 4. System retrieves total clinicians count 5. System retrieves total organizations count 6. System retrieves total visits count 7. System calculates visit statistics by status 8. System displays analytics dashboard with charts and statistics
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired System returns 401 Unauthorized <p>A2: No Data Available</p> <ul style="list-style-type: none"> - At steps 3-6, if no data exists in the system - System shows empty state message
Post condition	Analytics dashboard is displayed with system statistics

Table 51, Use Case Specification (View Admin Analytics)

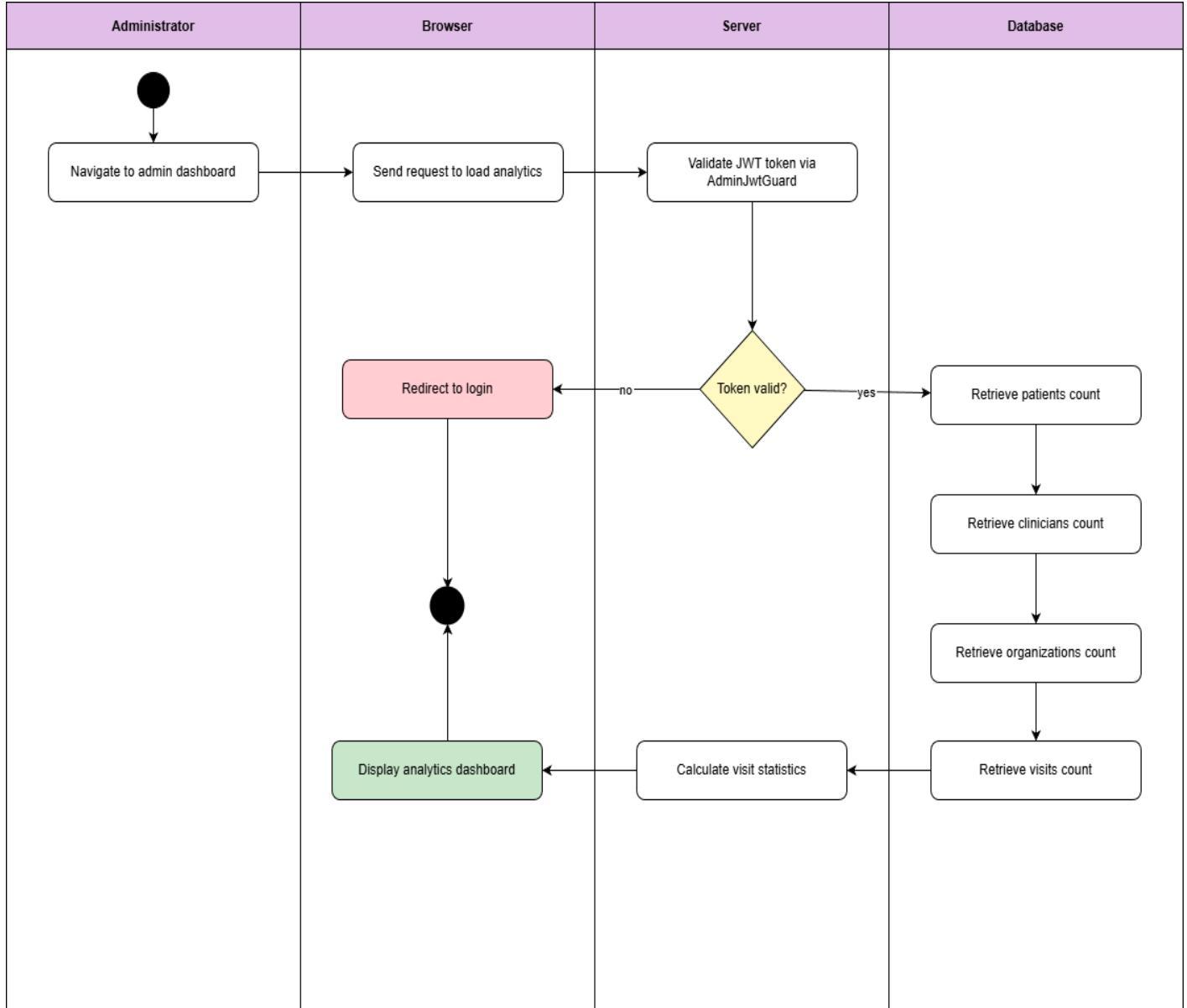


Diagram 92, Activity Diagram (View Admin Analytics)

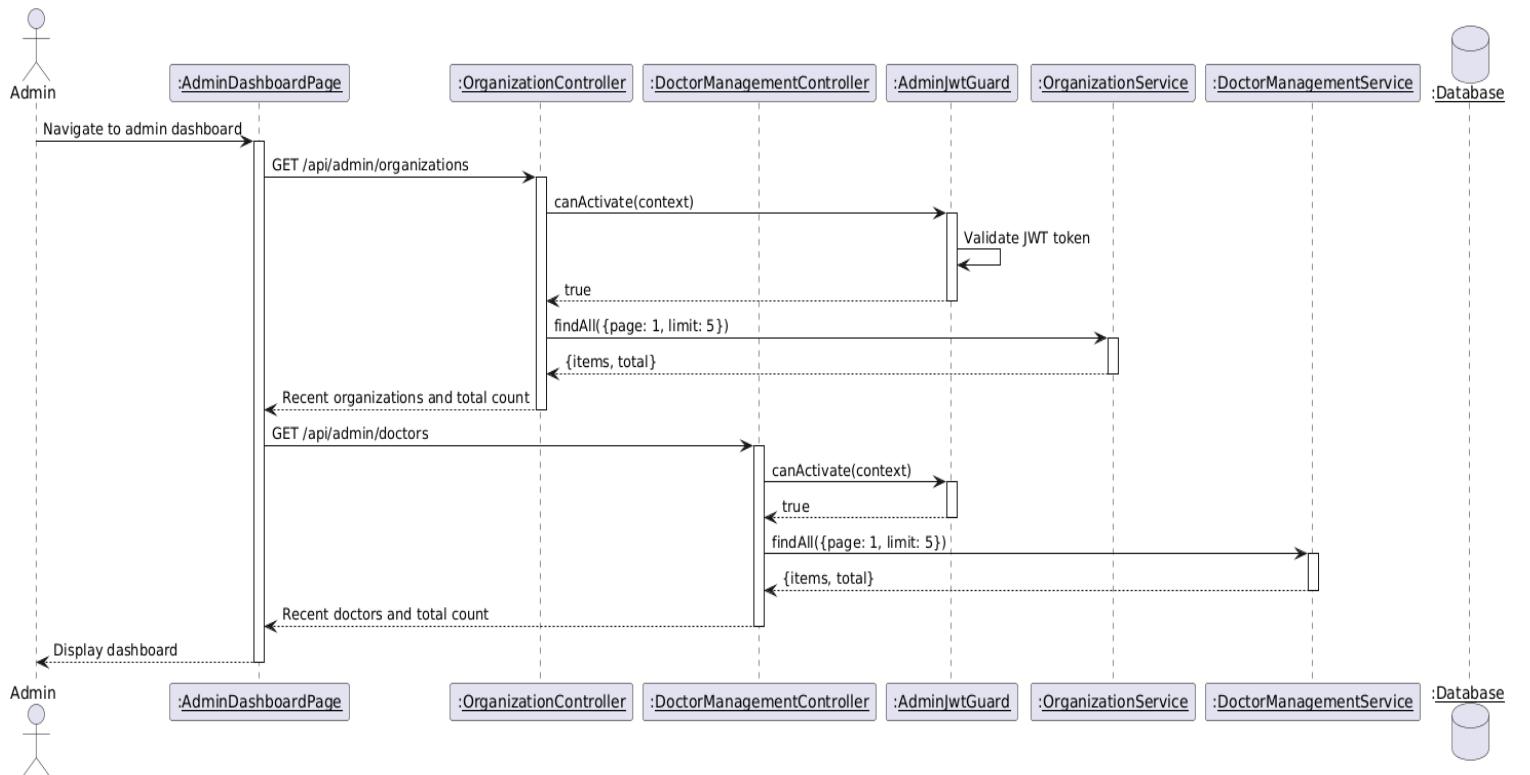


Diagram 93, Sequence Diagram (View Admin Analytics)

- **View All Appointment:**

Use case ID	VEMR-FR-AP-45
Use case name	View All Appointment
Description	The system allows clinicians to view their complete appointment schedule
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to appointments page 2. System validates JWT token 3. System retrieves all appointments for clinician 4. System returns paginated appointments with patient details 5. System displays appointments list with date, time, patient name, and status
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired - System redirects to login page <p>A2: No Appointments Found</p> <ul style="list-style-type: none"> - At step 3, if clinician has no appointments - System returns empty list
Post condition	Appointment's list is displayed with pagination

Table 52, Use Case Specification (View All Appointment)

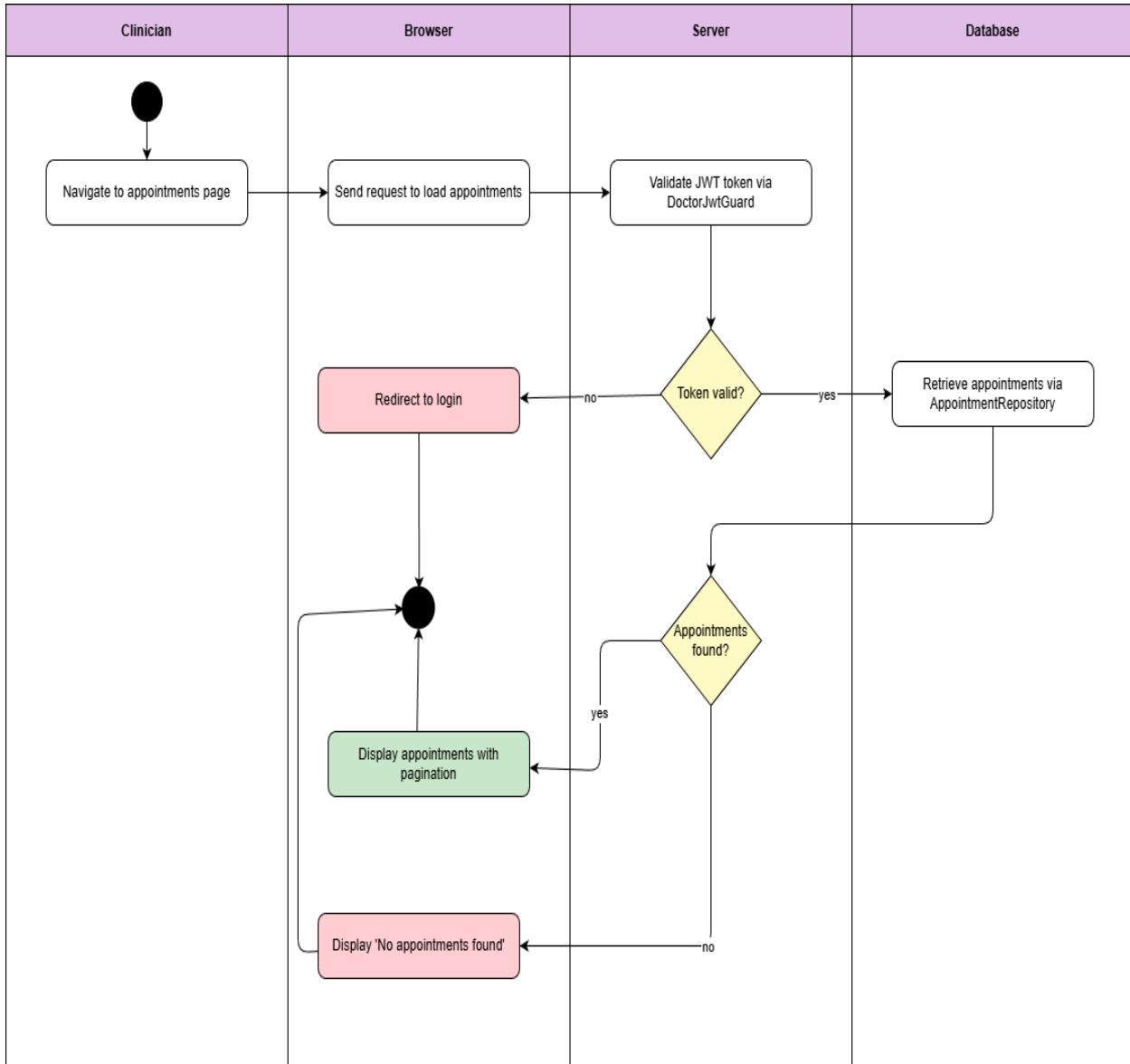


Diagram 94, Activity Diagram (View All Appointment)

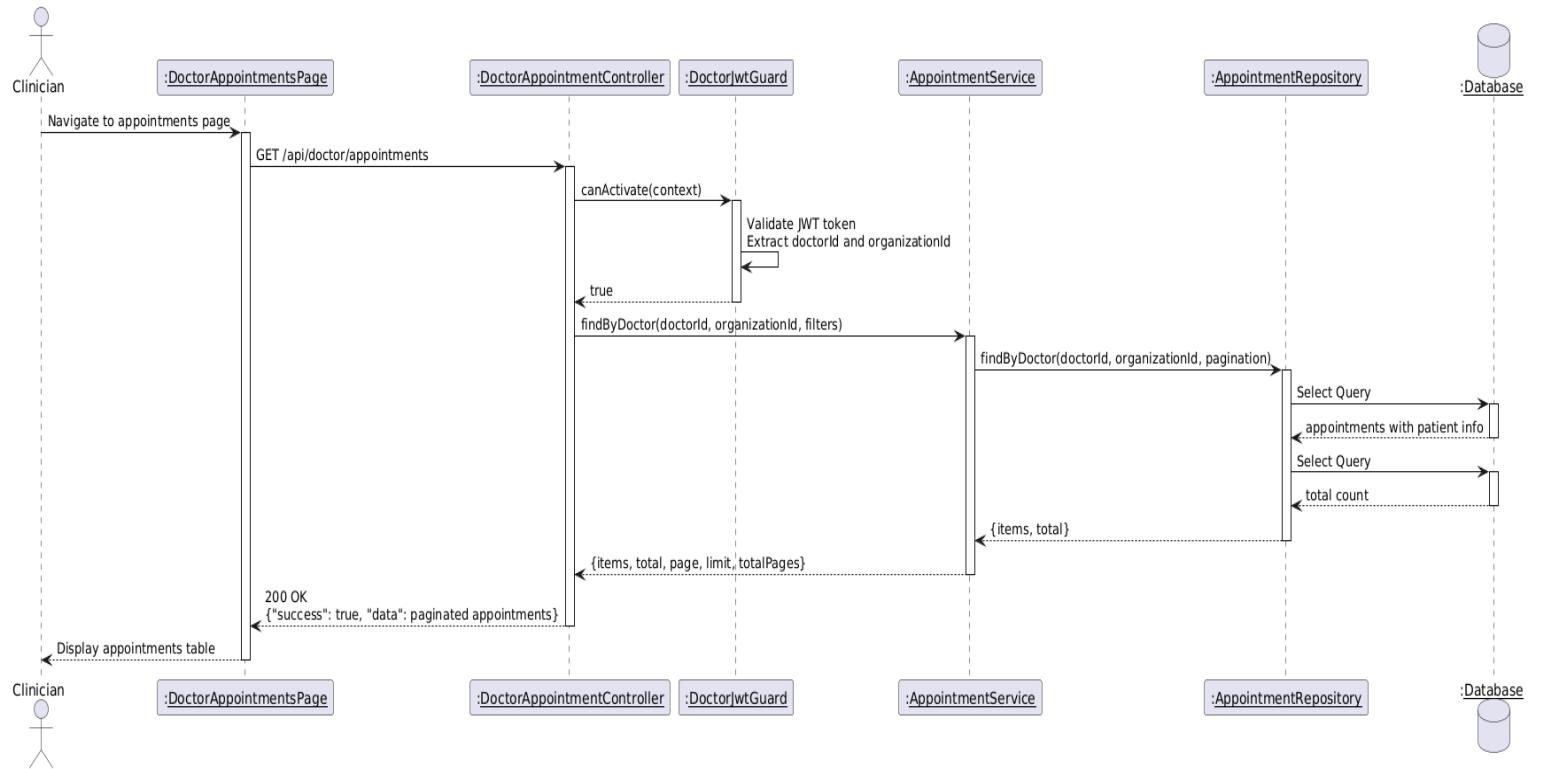


Diagram 95, Sequence Diagram (View All Appointment)

- **View Daily Appointments:**

Use case ID	VEMR-FR-AP-46
Use case name	View Daily Appointments
Description	The system allows clinicians to view their daily appointment schedule.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to daily appointments view 2. System validates JWT token 3. System retrieves today's appointments for clinician 4. System filters appointments by current date 5. System displays daily appointments with time slots and patient details
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired - System redirects to login page <p>A2: No Appointments Today</p> <ul style="list-style-type: none"> - At step 3, if clinician has no appointments for today - System returns empty list
Post condition	Daily appointments are displayed in chronological order

Table 534, Use Case Specification (View Daily Appointments)

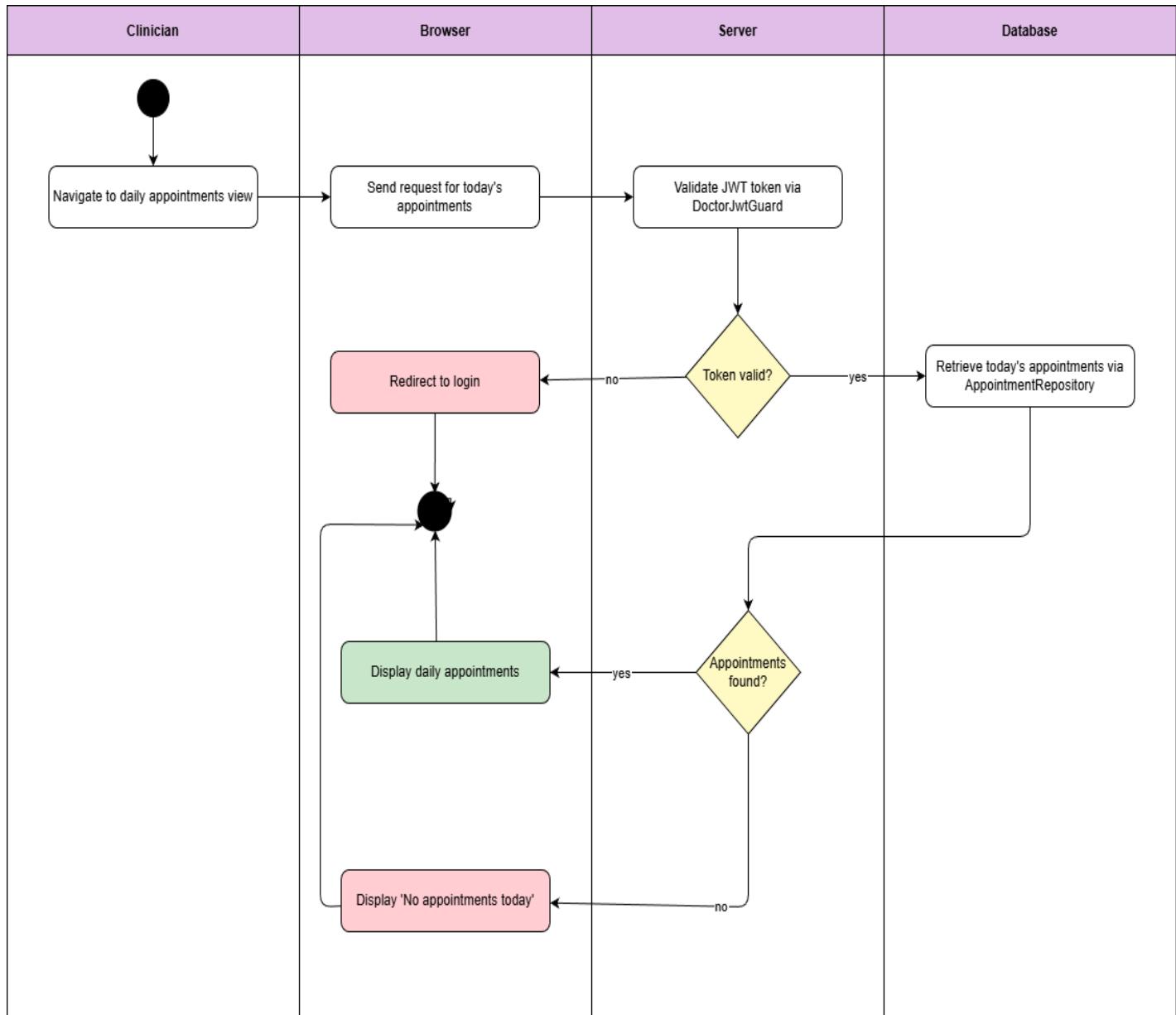


Diagram 96, Activity Diagram (View Daily Appointment)

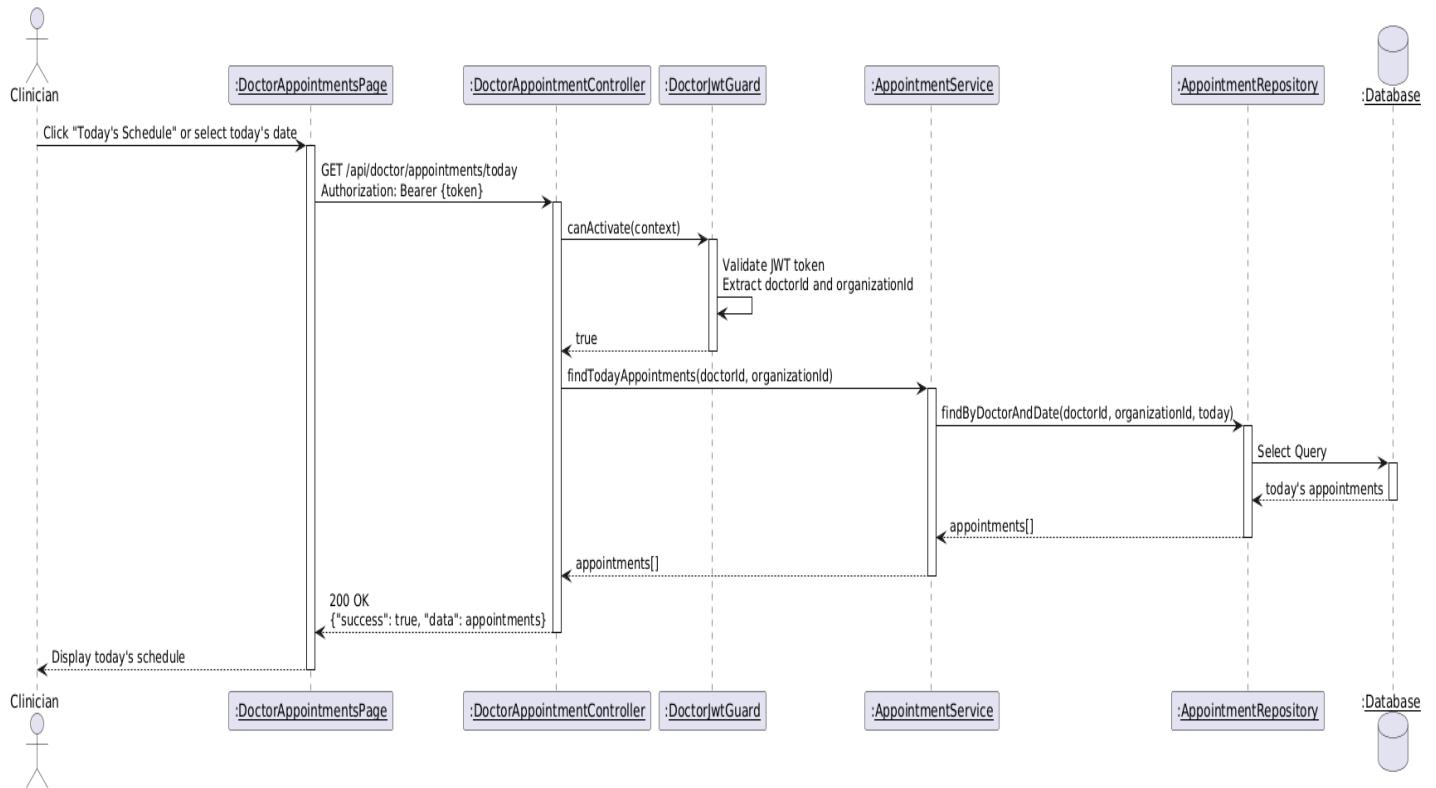


Diagram 97, Sequence Diagram (View Daily Appointment)

- **Filter Appointments Per Date:**

Use case ID	VEMR-FR-AP-47
Use case name	Filter Appointments Per Date
Description	The system allows clinicians to filter appointments by specific date
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to appointments page 2. Clinician selects date from date picker 3. System validates JWT token via DoctorJwtGuard 4. System retrieves appointments for selected date via Appointment Repository 5. System displays filtered appointments for the selected date
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: No Appointments on Selected Date</p> <ul style="list-style-type: none"> - At step 4, if no appointments exist for selected date - System displays "No appointments found for this date" message
Post condition	Clinician can view appointments for specific date

Table 54, Use Case Specification (Filter Appointments Per Date)

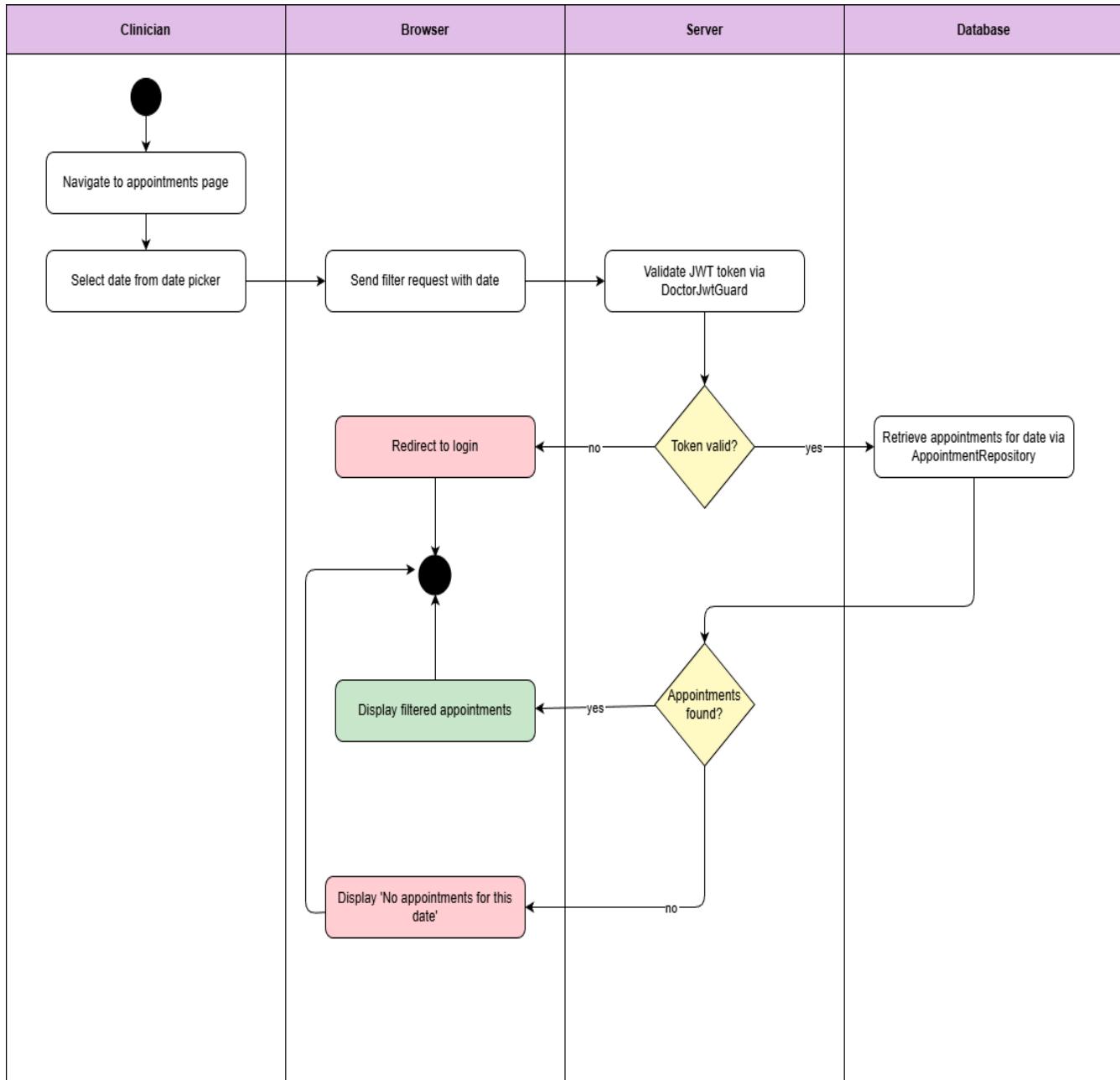


Diagram 98, Activity Diagram (Filter Appointments Per Date)

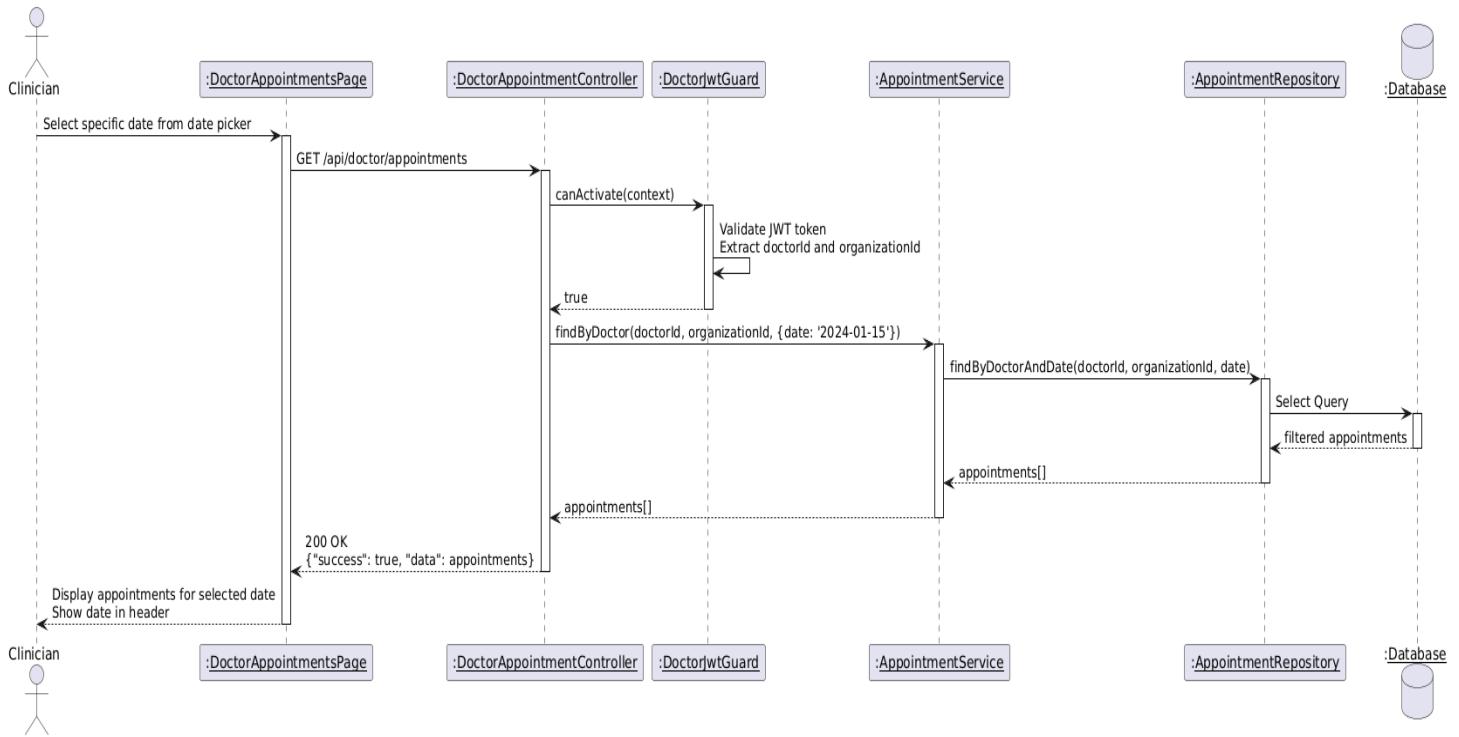


Diagram 99, Sequence Diagram (Filter Appointments Per Date)

- **Filter Appointments Per Status:**

Use case ID	VEMR-FR-AP-48
Use case name	Filter Appointments Per Status
Description	The system allows clinicians to filter appointments by status (scheduled, completed, cancelled, no-show).
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to appointments page 2. Clinician selects status from filter dropdown 3. System validates JWT token via DoctorJwtGuard 4. System retrieves appointments with selected status via Appointment Repository 5. System displays filtered appointments by status
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: No Appointments on Selected Status</p> <ul style="list-style-type: none"> - At step 4, if no appointments exist for selected status - System displays "No appointments found for this status" message
Post condition	Clinician can view appointments by status

Table 55, Use Case Specification (Filter Appointments Per Status)

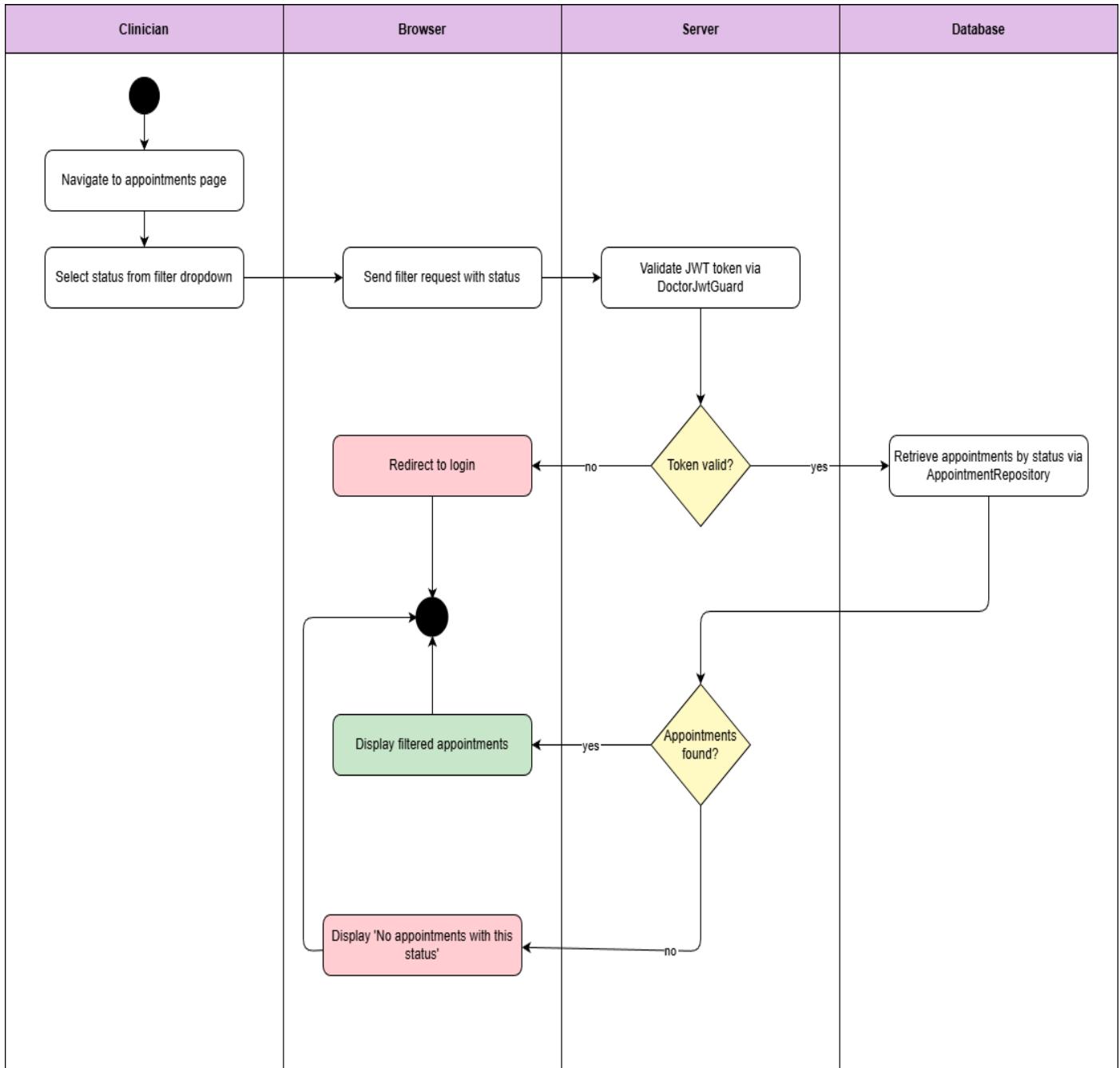


Diagram 100, Activity Diagram (Filter Appointments Per Status)

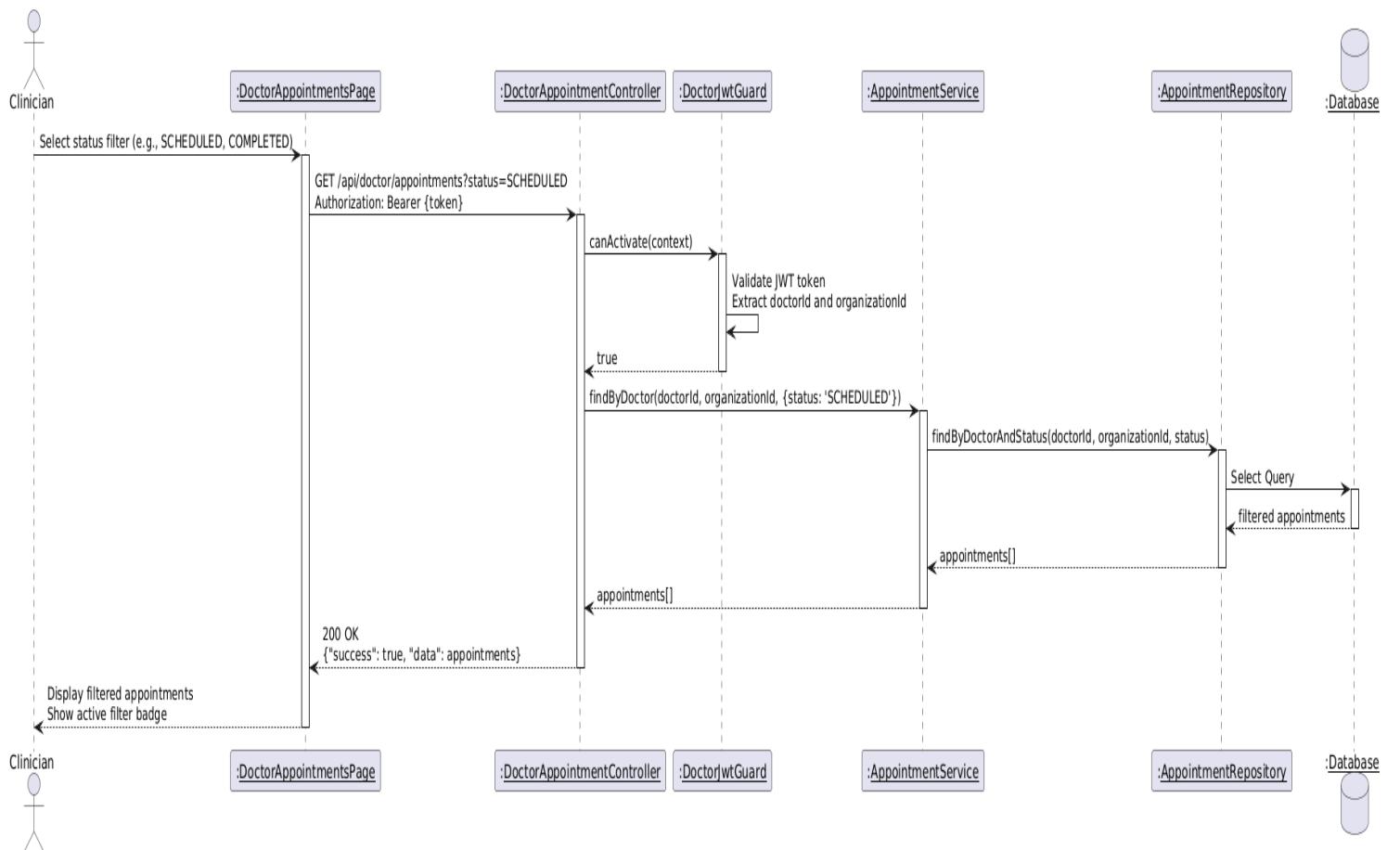


Diagram 101, Sequence Diagram (Filter Appointments Per Status)

- **View Appointment Details:**

Use case ID	VEMR-FR-AP-49
Use case name	View Appointment Details
Description	The system allows clinicians to view detailed information about a specific appointment
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to appointments list 2. Clinician clicks on appointment to view details 3. System validates JWT token via DoctorJwtGuard 4. System retrieves appointment details via Appointment Repository 5. System retrieves patient information via Patient Repository 6. System displays appointment details with patient info, date, time, status, and notes
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: Appointment Not Found</p> <ul style="list-style-type: none"> - At step 4, if appointment ID does not exist - System displays error message
Post condition	Clinician can view patient and appointment information

Table 56, Use Case Specification (View Appointment Details)

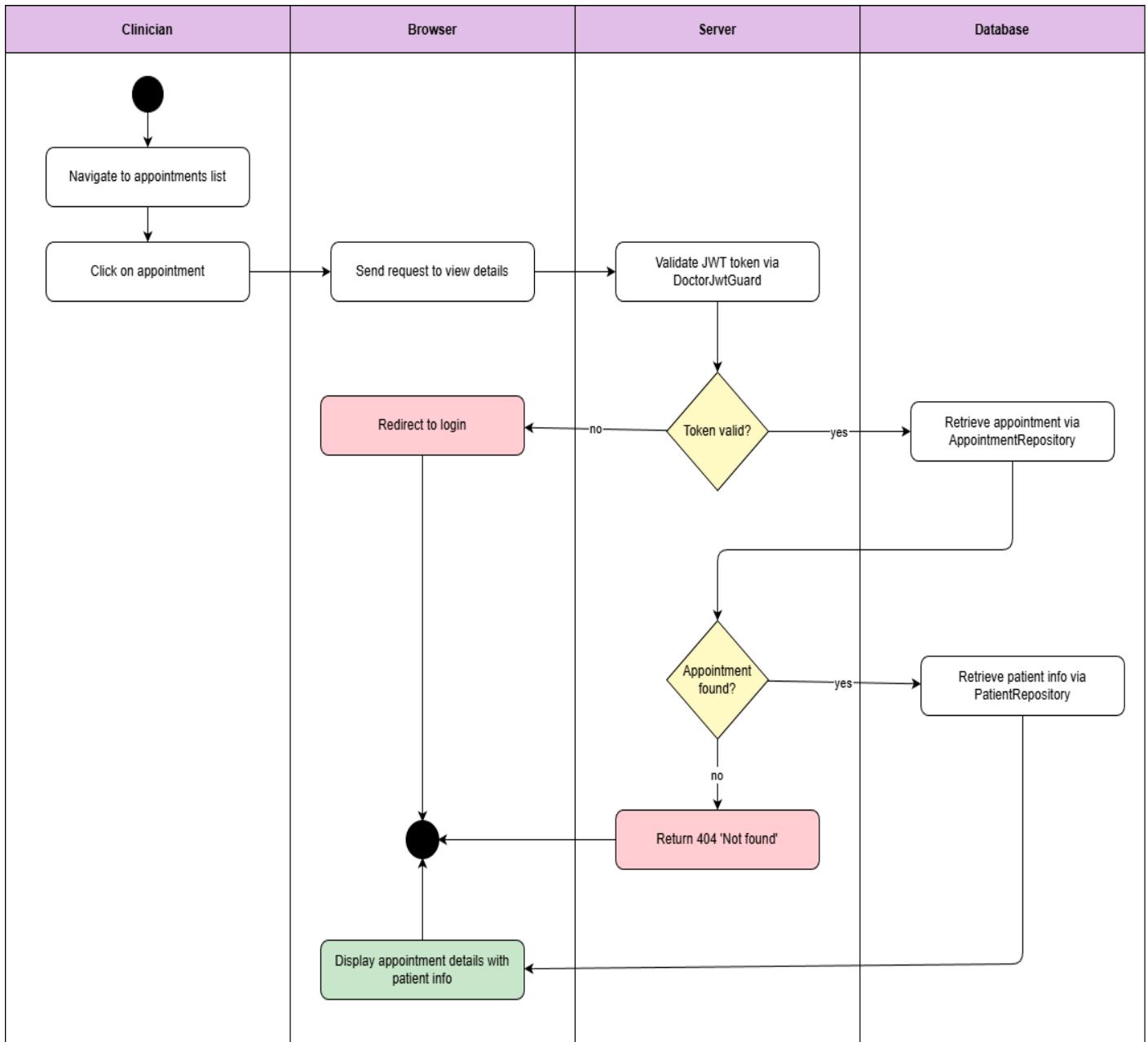


Diagram 102, Activity Diagram (View Appointment Details)

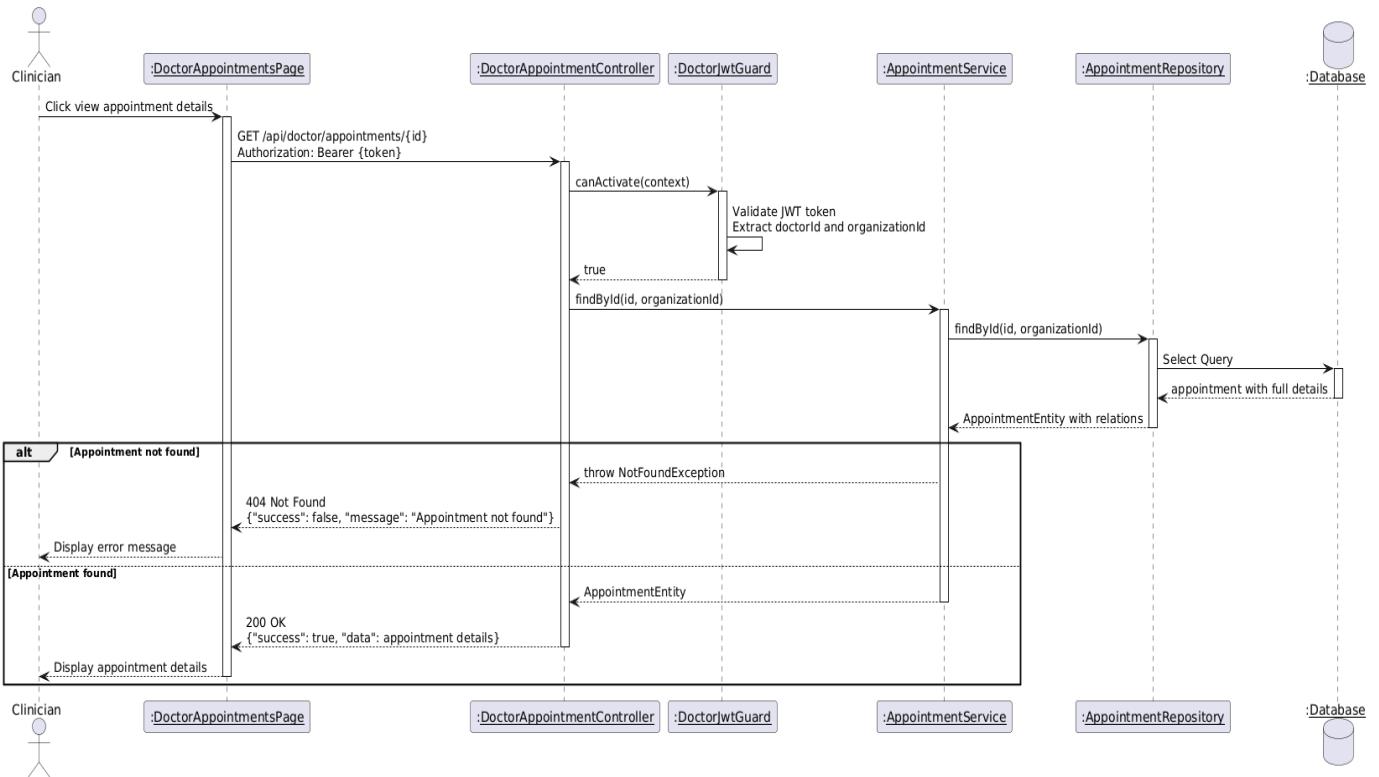


Diagram 103, Sequence Diagram (View Appointment Details)

- **Check In an Appointment:**

Use case ID	VEMR-FR-AP-50
Use case name	Check In an Appointment
Description	The system allows clinicians to check in a patient for their appointment.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to appointments list 2. Clinician selects appointment 3. Clinician clicks "Check In" button 4. System validates JWT token via DoctorJwtGuard 5. System verifies appointment status is scheduled 6. System updates appointment status to "checked-in" via Appointment Repository 7. System displays success message and refreshes appointments list.
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired - System redirects to login page <p>A2: Appointment Not Found</p> <ul style="list-style-type: none"> - At step 5, if appointment ID does not exist - System displays error message <p>A3: Invalid Status</p> <ul style="list-style-type: none"> - At step 5, if appointment is not in scheduled status - System displays error message
Post condition	Appointment status is updated to checked-in

Table 57, Use Case Specification (Check in an Appointment)

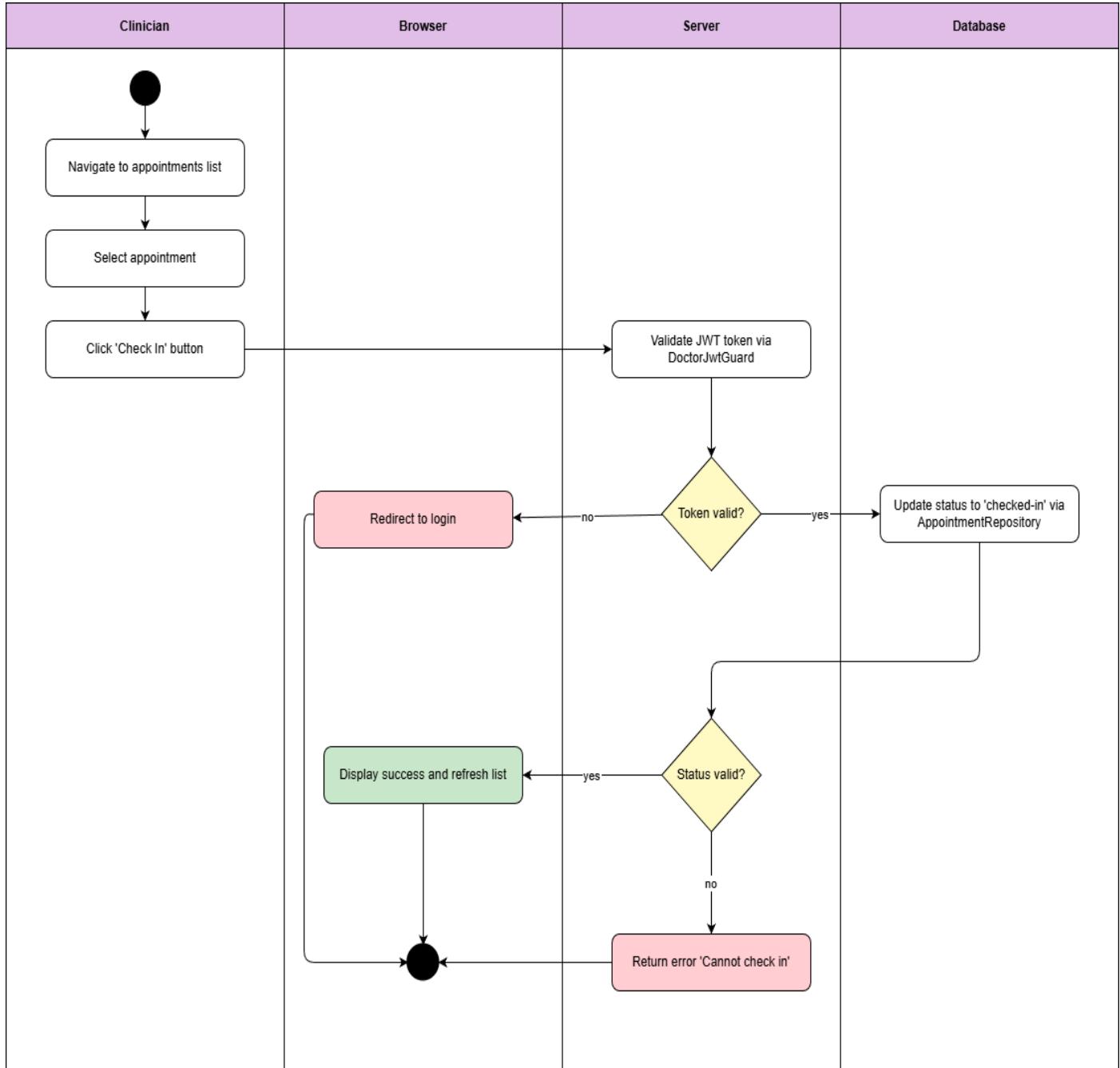


Diagram 104, Activity Diagram (Check in an Appointment)

Chapter 4 - System Analysis

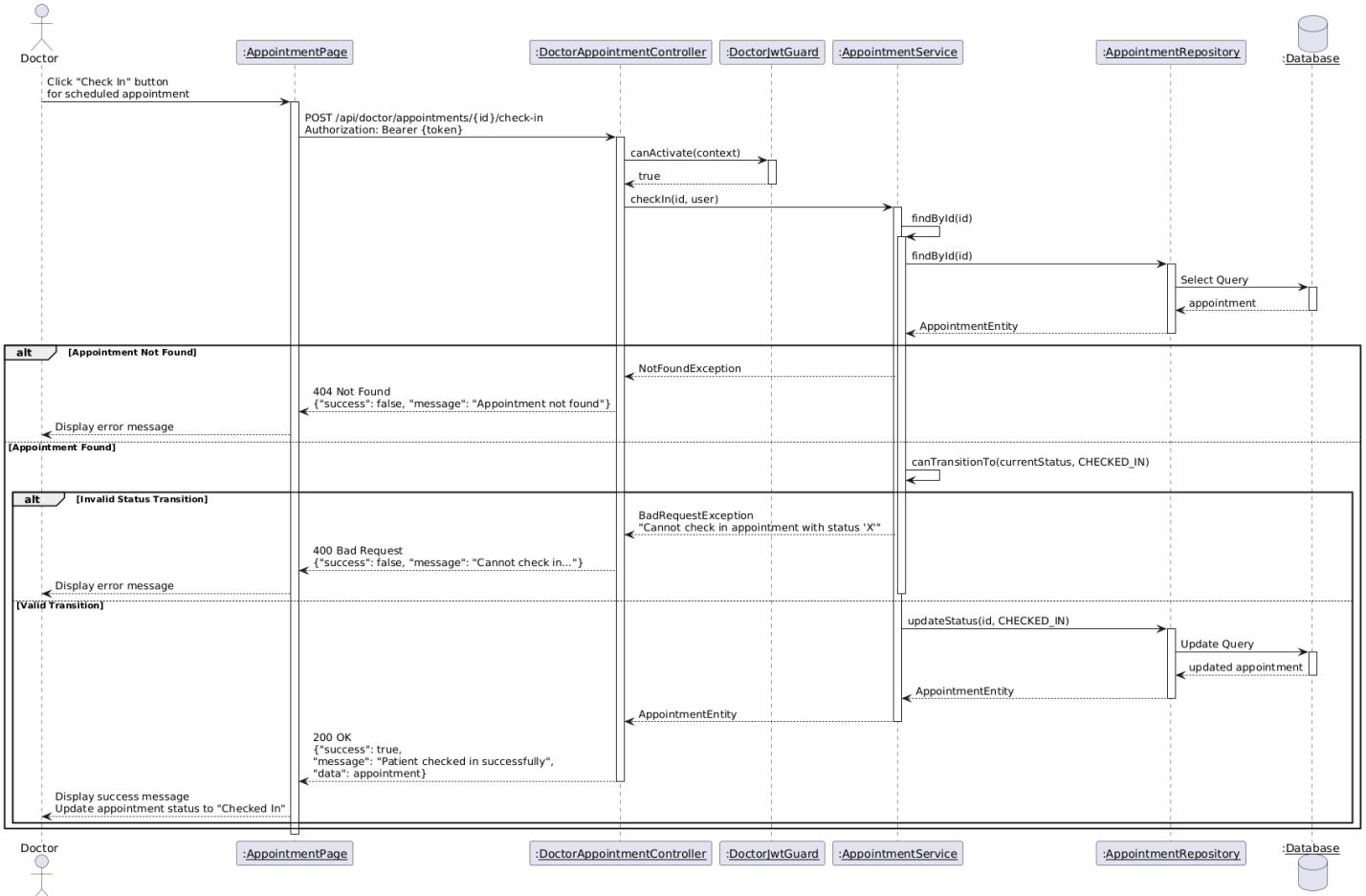


Diagram 105, Sequence Diagram (Check in an Appointment)

● Cancel an Appointment:

Use case ID	VEMR-FR-AP-51
Use case name	Cancel an Appointment
Description	The system allows clinicians or patients to cancel an appointment
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. User navigates to appointments list 2. User selects appointment 3. User clicks "Cancel" button 4. System displays cancellation confirmation dialog 5. User confirms cancellation 6. System validates JWT token via JwtGuard 7. System verifies appointment can be cancelled 8. System updates appointment status to "cancelled" via Appointment Repository 9. System releases appointment slot 10. System displays success message and refreshes appointments list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired - System redirects to login page <p>A2: Appointment Not Found</p> <ul style="list-style-type: none"> - At step 7, if appointment ID does not exist - System displays error message <p>A3: Cannot Cancel</p> <ul style="list-style-type: none"> - At step 7, if appointment is already completed - System displays error message
Post condition	Appointment status is updated to checked-Appointment status is updated to cancelled

Table 58, Use Case Specification (Cancel an Appointment)

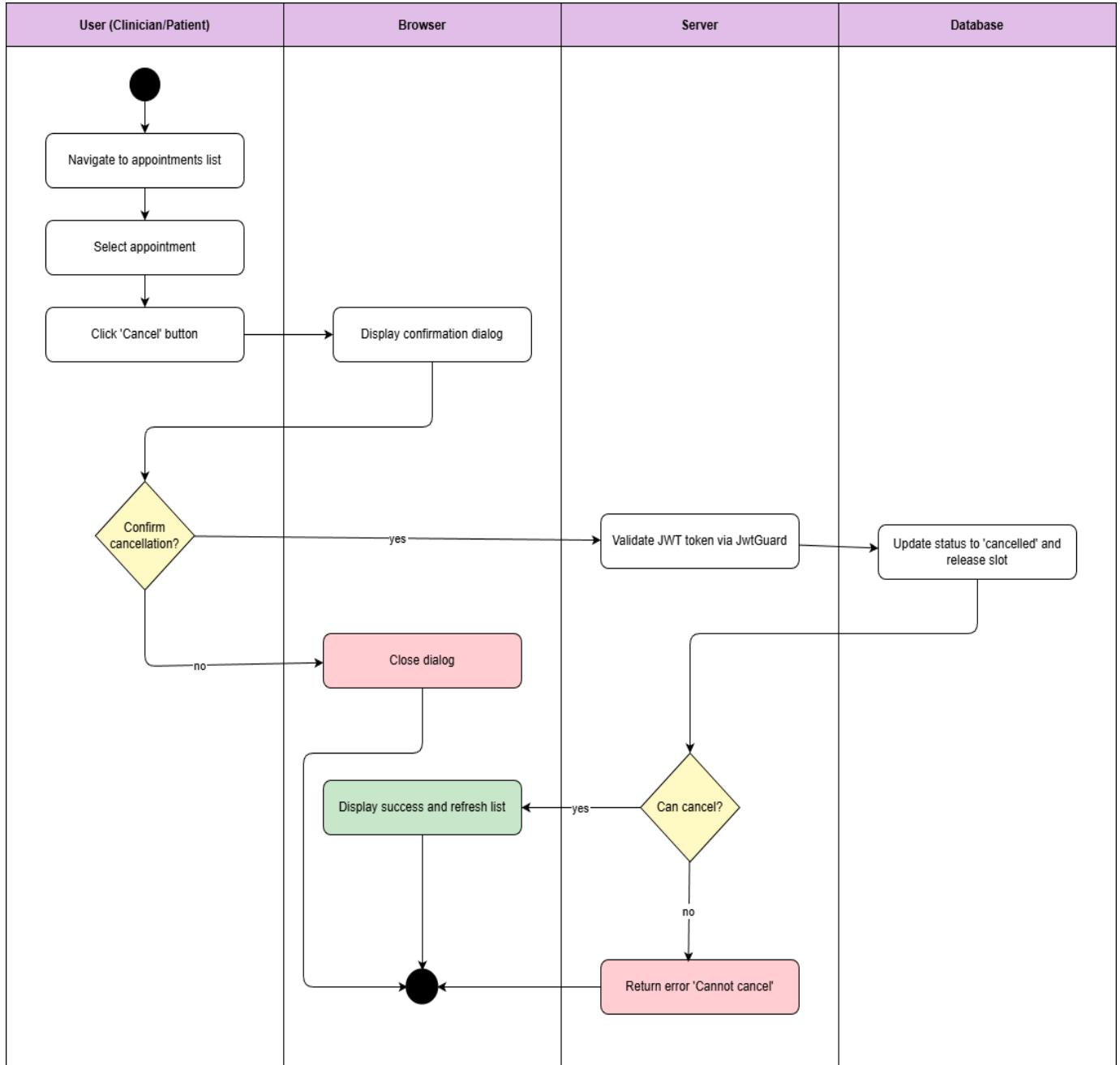


Diagram 106, Activity Diagram (Cancel an Appointment)

Chapter 4 - System Analysis

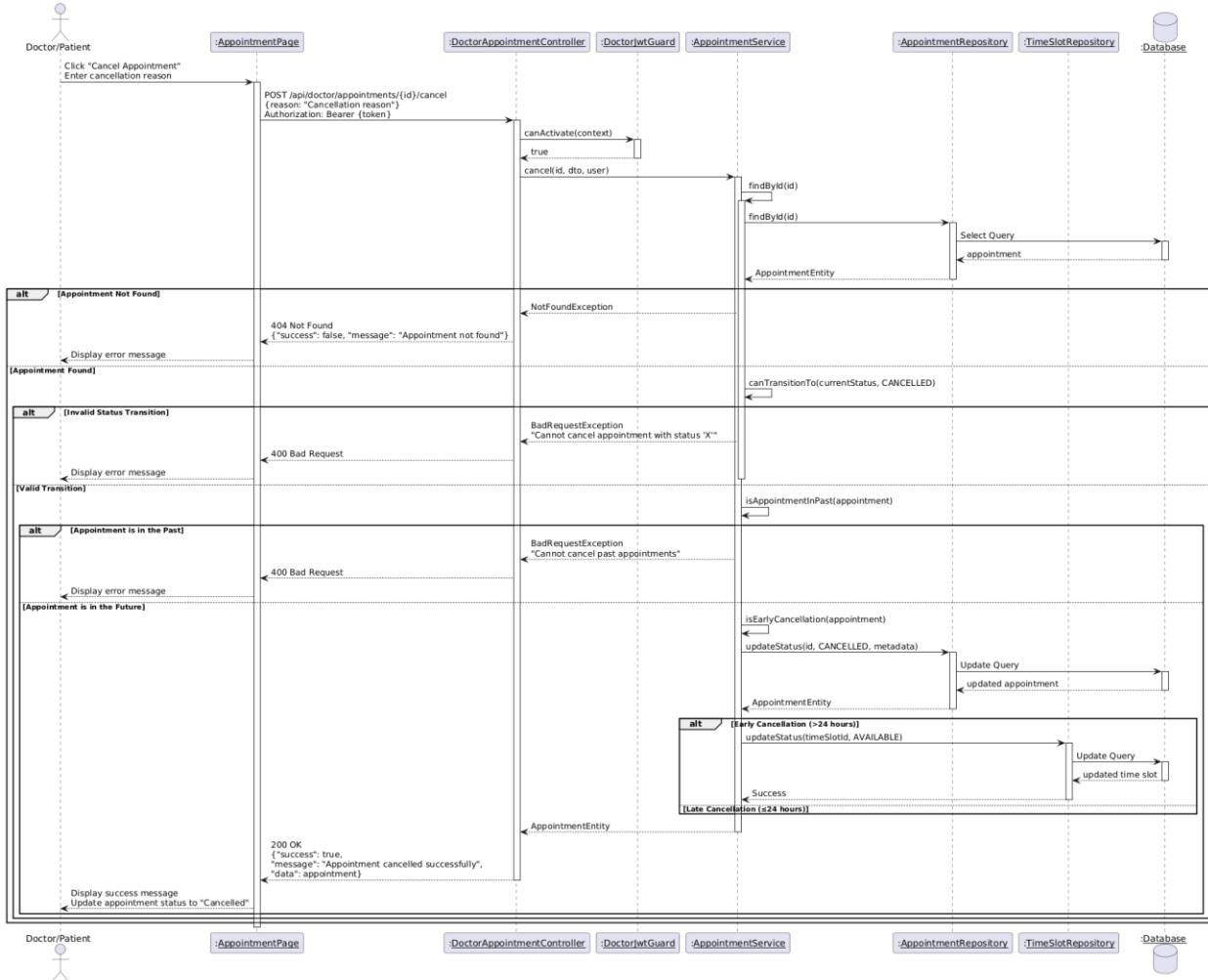


Diagram 107, Sequence Diagram (Cancel an Appointment)

- **Set an Appointment as No Show:**

Use case ID	VEMR-FR-DA-52
Use case name	Set an Appointment as No Show
Description	The system allows clinicians to mark an appointment as no-show when patient doesn't arrive.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to appointments list 2. Clinician selects appointment 3. Clinician clicks "Mark as No Show" button 4. System validates JWT token via DoctorJwtGuard 5. System verifies appointment status 6. System updates appointment status to "no-show" via Appointment Repository 7. System releases appointment slot 8. System displays success message and refreshes appointments list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 4, if JWT token is invalid or expired - System redirects to login page <p>A2: Appointment Not Found</p> <ul style="list-style-type: none"> - At step 5, if appointment ID does not exist - System displays error message <p>A3: Invalid Status</p> <ul style="list-style-type: none"> - At step 5, if appointment is already completed or cancelled - System displays error message
Post condition	-Appointment status is updated to no-show

Table 59, Use Case Specification (Set an Appointment as No Show)

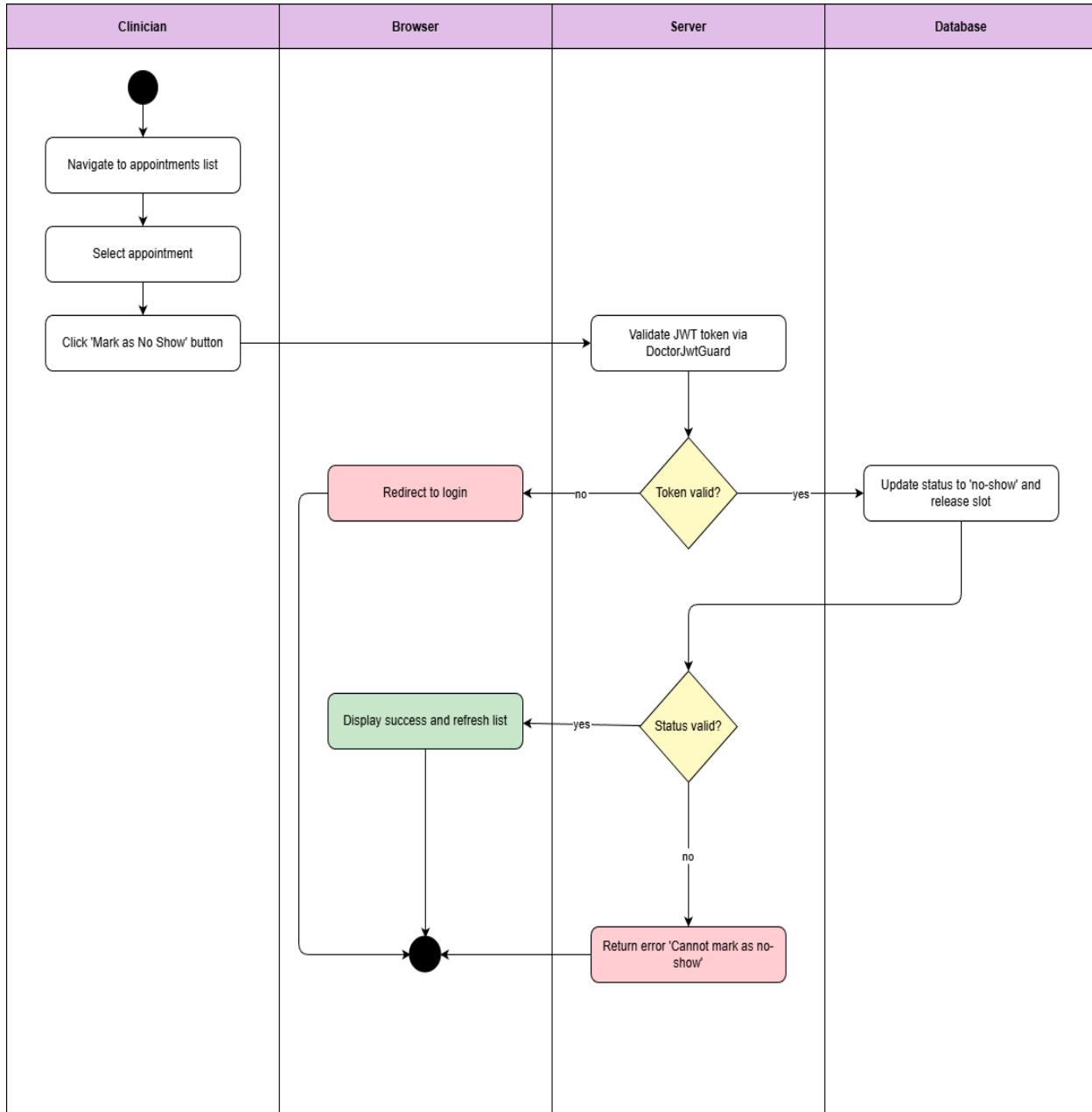


Diagram 108, Activity Diagram (Set an Appointment as No Show)

Chapter 4 - System Analysis

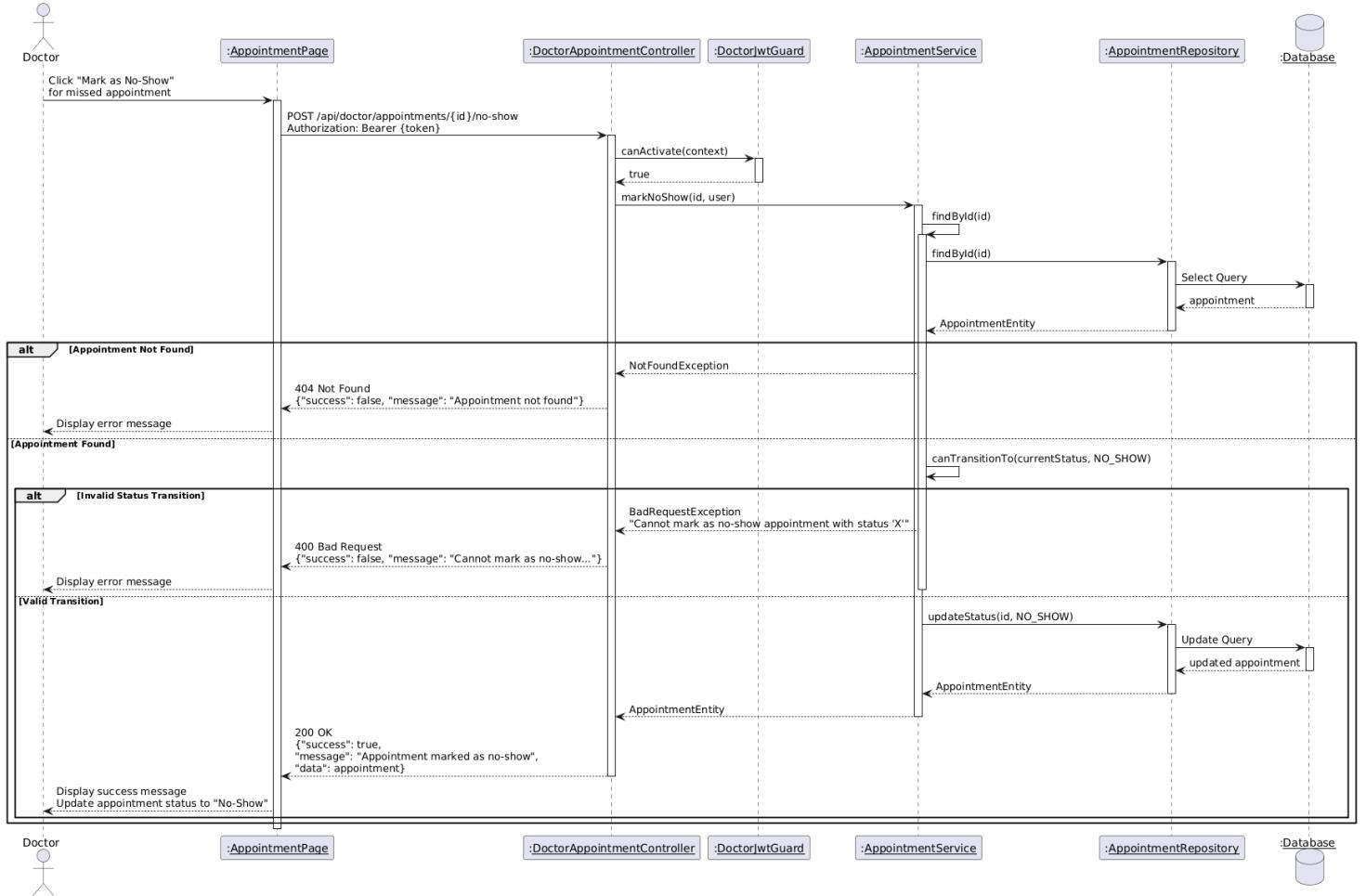


Diagram 109, Sequence Diagram (Set an Appointment as No Show)

- **View Doctor Analytics:**

Use case ID	VEMR-FR-SM-53
Use case name	View Doctor Analytics
Description	The system allows clinicians to view their personal analytics and performance metrics.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to analytics dashboard 2. System validates JWT token via DoctorJwtGuard 3. System retrieves total appointments count via Appointment Repository 4. System retrieves completed visits count via Encounter Repository 5. System calculates appointment statistics by status 6. System displays analytics dashboard with charts and metrics
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired - System redirects to login page <p>A2: No Data Available</p> <ul style="list-style-type: none"> - At steps 3-4, if no data exists - System shows empty state message
Post condition	Clinician can view personal statistics

Table 60, Use Case Specification (View Doctor Analytics)

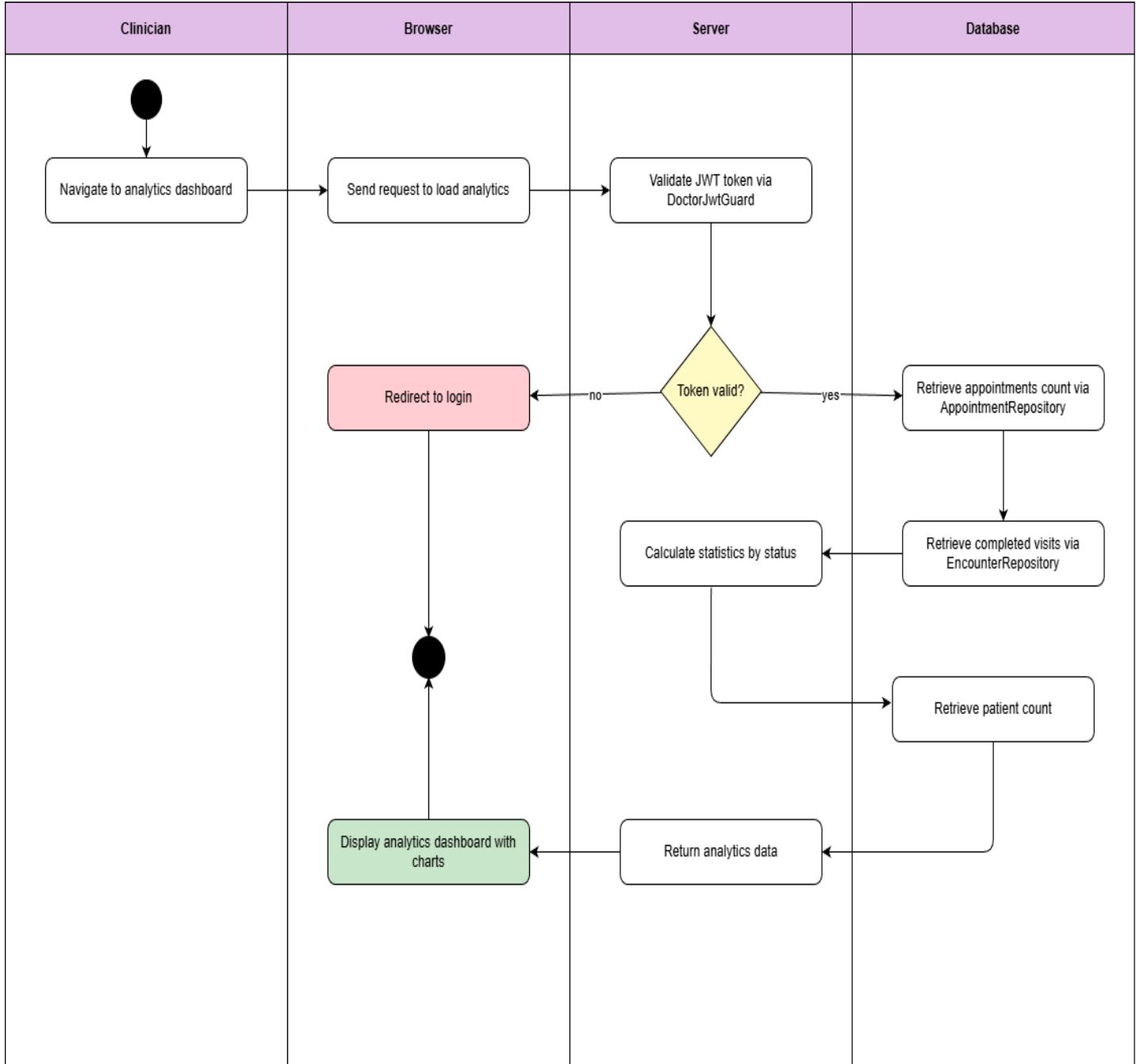


Diagram 110, Activity Diagram (View Doctor Analytics)

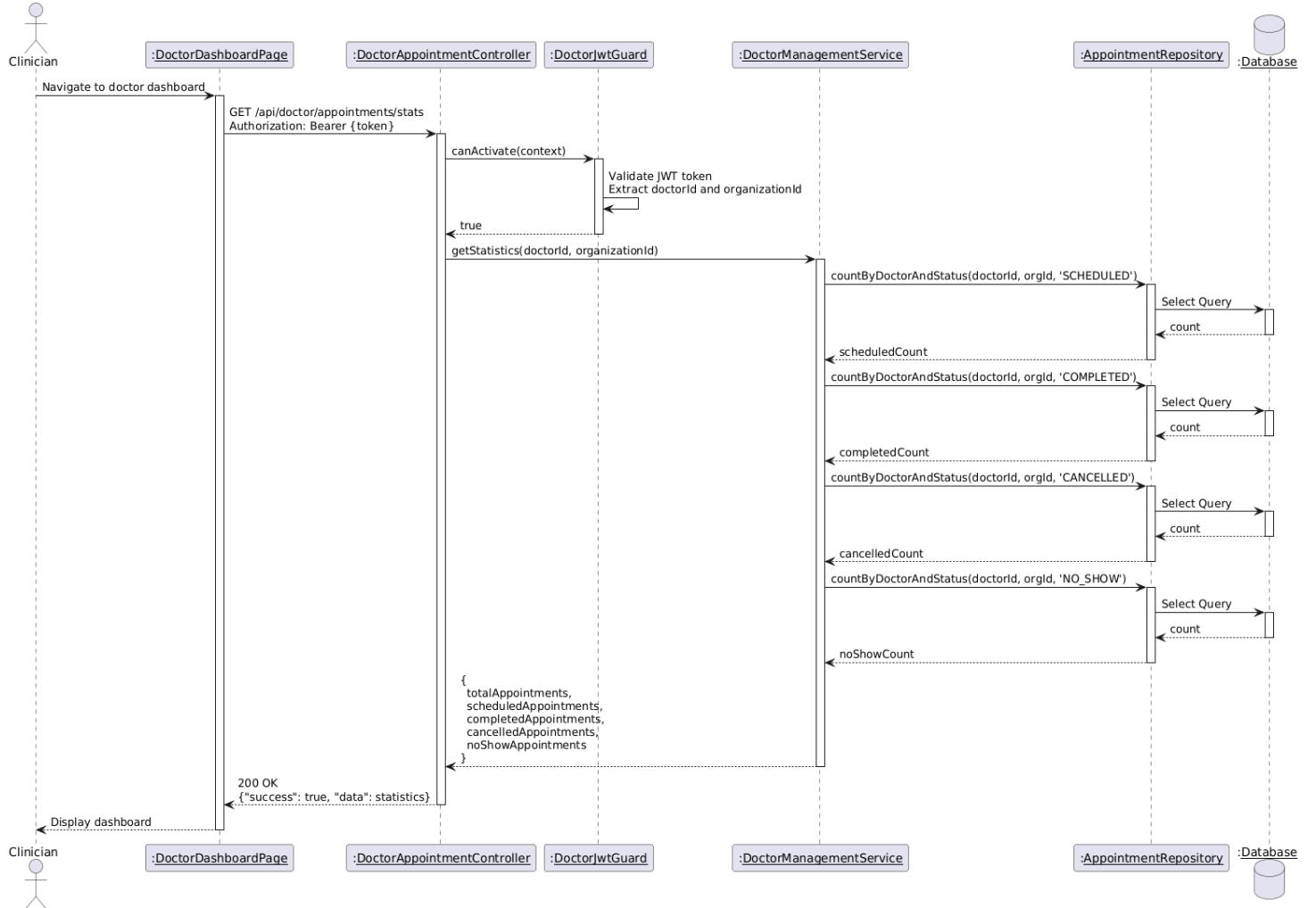


Diagram 111, Sequence Diagram (View Doctor Analytics)

● Create Schedule:

Use case ID	VEMR-FR-SM-54
Use case name	Create Schedule
Description	The system allows clinicians to create their work schedule
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to schedule management page 2. Clinician clicks "Create Schedule" button 3. System displays schedule creation form 4. Clinician enters schedule details (day, start time, end time) 5. Clinician submits form 6. System validates input fields 7. System validates JWT token via DoctorJwtGuard 8. System creates schedule record via Schedule Repository 9. System displays success message and refreshes schedule list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 7, if JWT token is invalid or expired - System redirects to login page <p>A2: Validation Error</p> <ul style="list-style-type: none"> - At step 6, if required fields are empty or invalid - System displays validation error messages
Post condition	New schedule is created in the system

Table 61, Use Case Specification (Create Schedule)

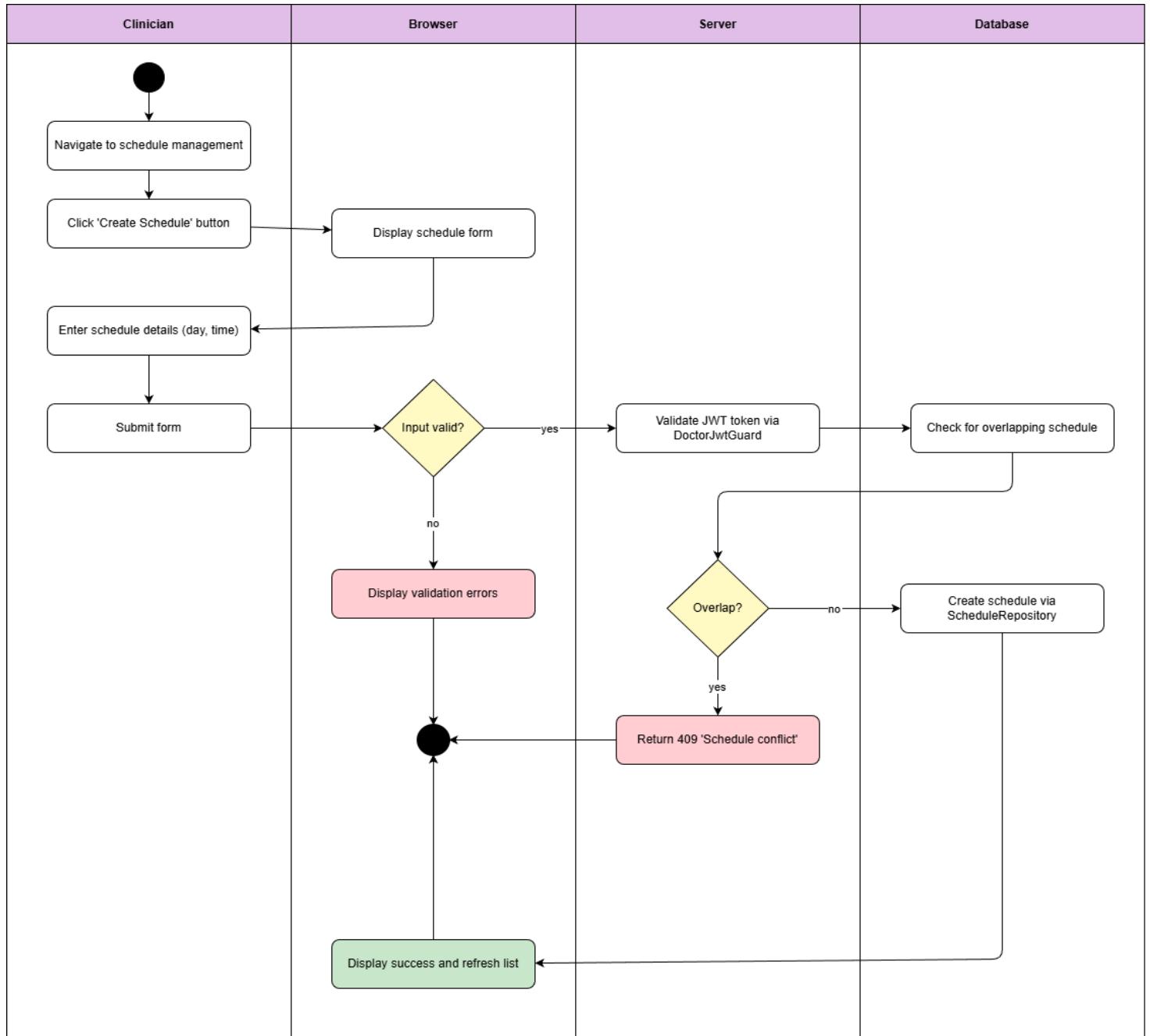


Diagram 112, Activity Diagram (Create Schedule)

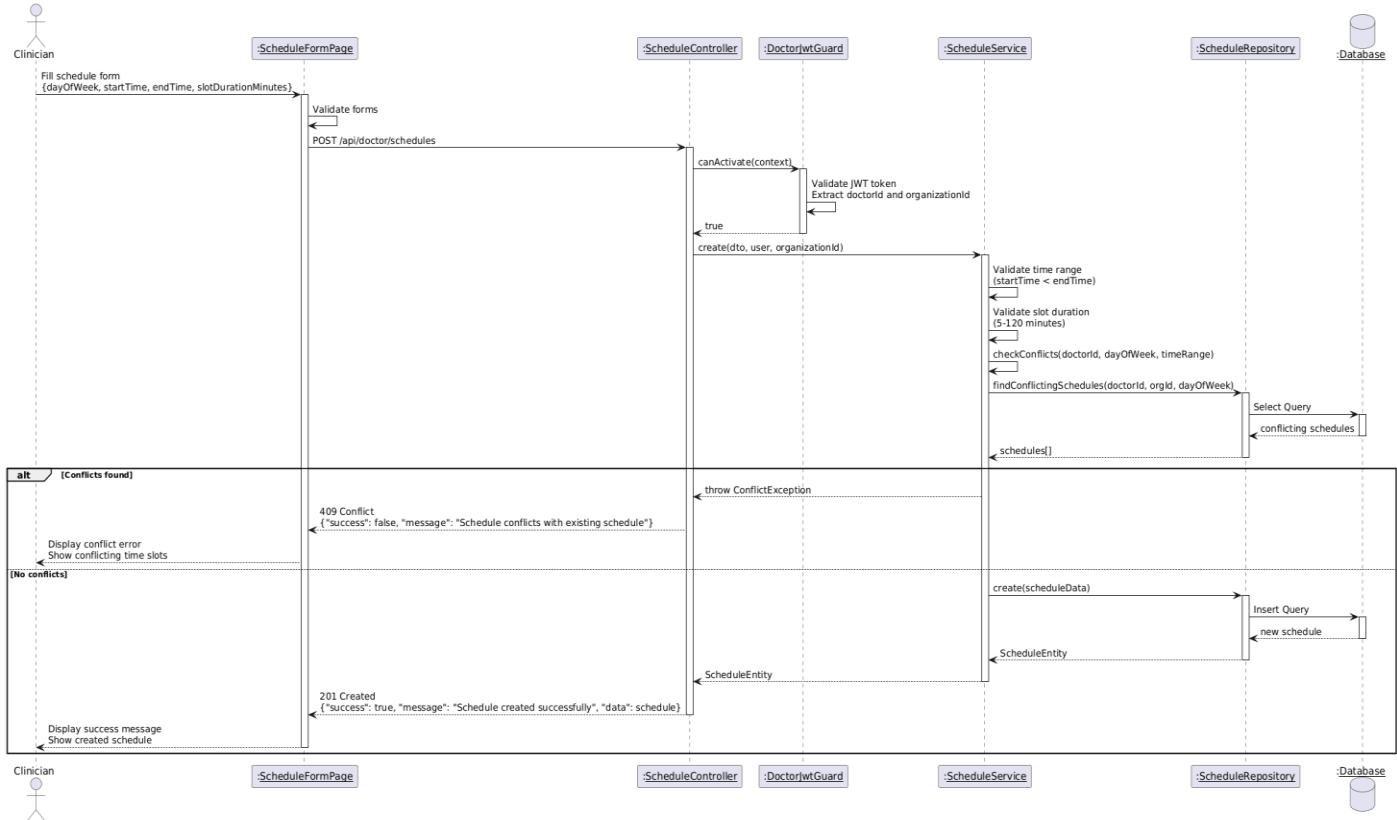


Diagram 113, Sequence Diagram (Create Schedule)

● Edit Schedule:

Use case ID	VEMR-FR-SM-55
Use case name	Edit Schedule
Description	The system allows clinicians to edit their existing work schedule.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to schedule list 2. Clinician clicks "Edit" button on schedule 3. System displays edit form with current data 4. Clinician modifies schedule details 5. Clinician submits form 6. System validates input fields 7. System validates JWT token via DoctorJwtGuard 8. System updates schedule via Schedule Repository 9. System displays success message and refreshes schedule list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 7, if JWT token is invalid or expired - System redirects to login page <p>A2: Validation Error</p> <ul style="list-style-type: none"> - At step 6, if required fields are empty or invalid - System displays validation error messages <p>A3: Schedule Not Found</p> <ul style="list-style-type: none"> - At step 8, if schedule ID does not exist - System displays error message
Post condition	Schedule information is updated in the system

Table 62, Use Case Specification (Edit Schedule)

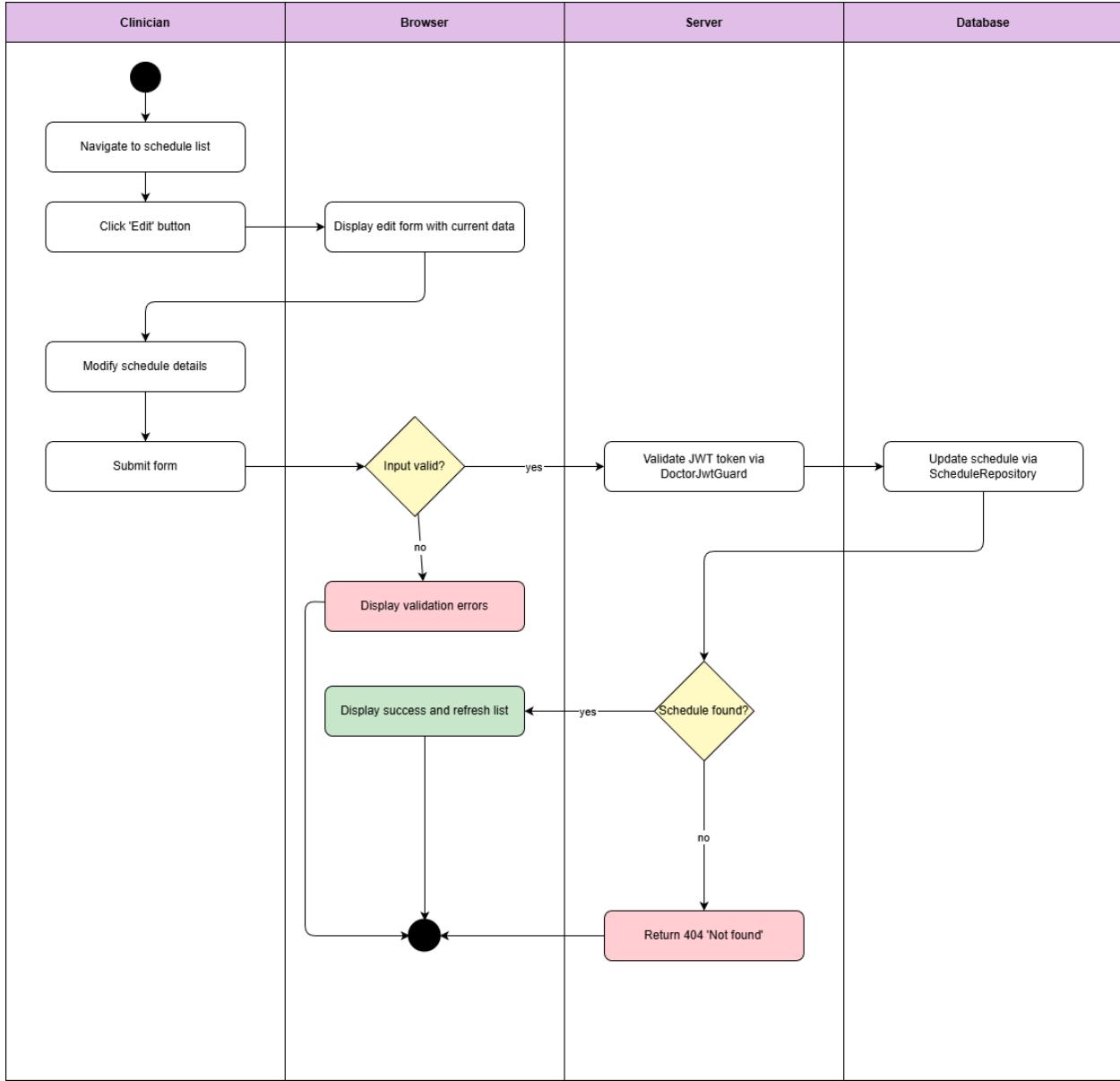


Diagram 114, Activity Diagram (Edit Schedule)

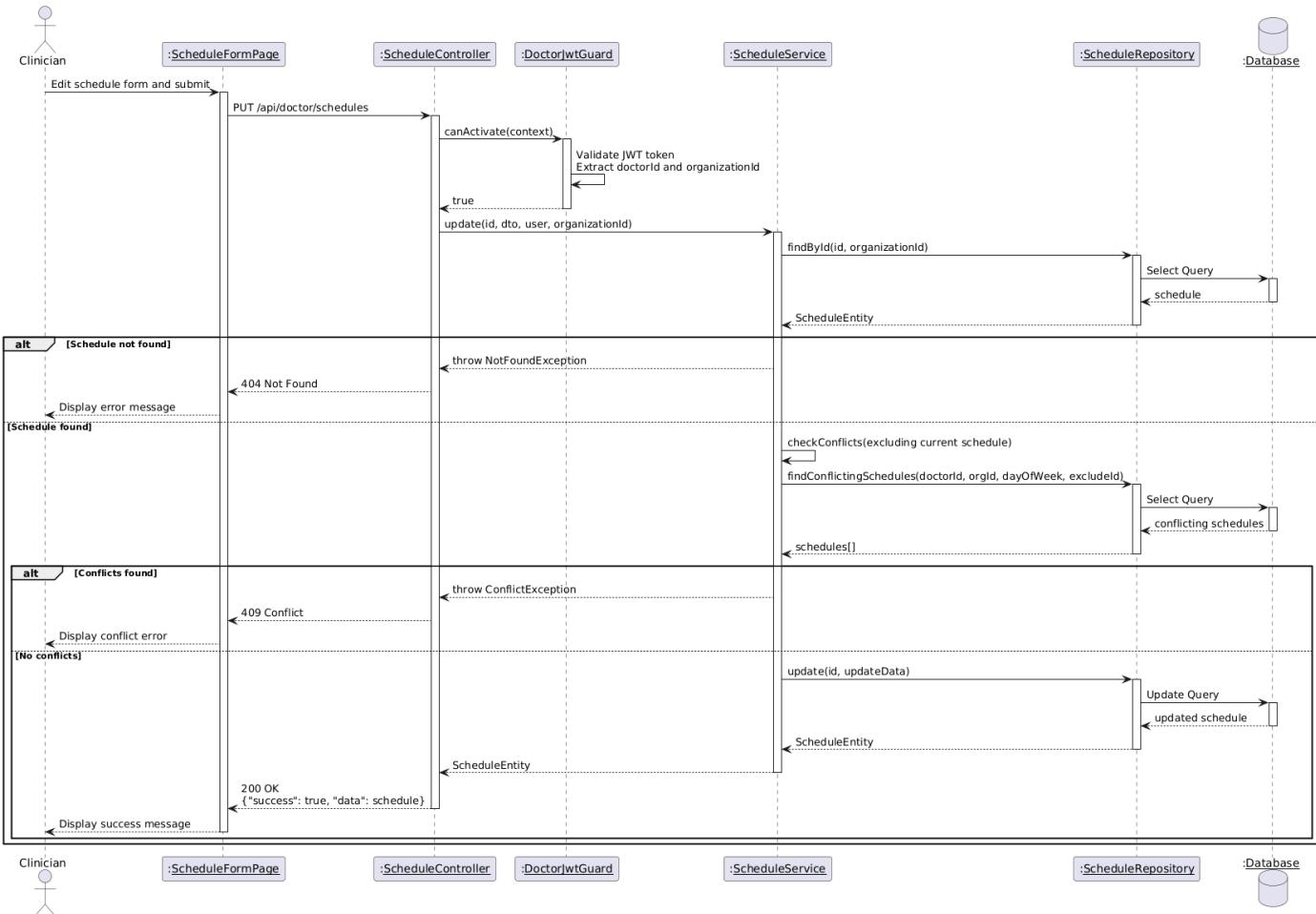


Diagram 115, Sequence Diagram (Edit Schedule)

● Delete Schedule:

Use case ID	VEMR-FR-SM-56
Use case name	Delete Schedule
Description	The system allows clinicians to delete their work schedule.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to schedule list 2. Clinician clicks "Delete" button on schedule 3. System displays confirmation dialog 4. Clinician confirms deletion 5. System validates JWT token via DoctorJwtGuard 6. System deletes schedule via Schedule Repository 7. System displays success message and refreshes schedule list
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 5, if JWT token is invalid or expired - System redirects to login page <p>A2: Schedule Not Found</p> <ul style="list-style-type: none"> - At step 6, if schedule ID does not exist - System displays error message <p>A3: Deletion Cancelled</p> <ul style="list-style-type: none"> - At step 4, if clinician cancels confirmation - No deletion occurs
Post condition	Schedule is removed from the system

Table 63, Use Case Specification (Delete Schedule)

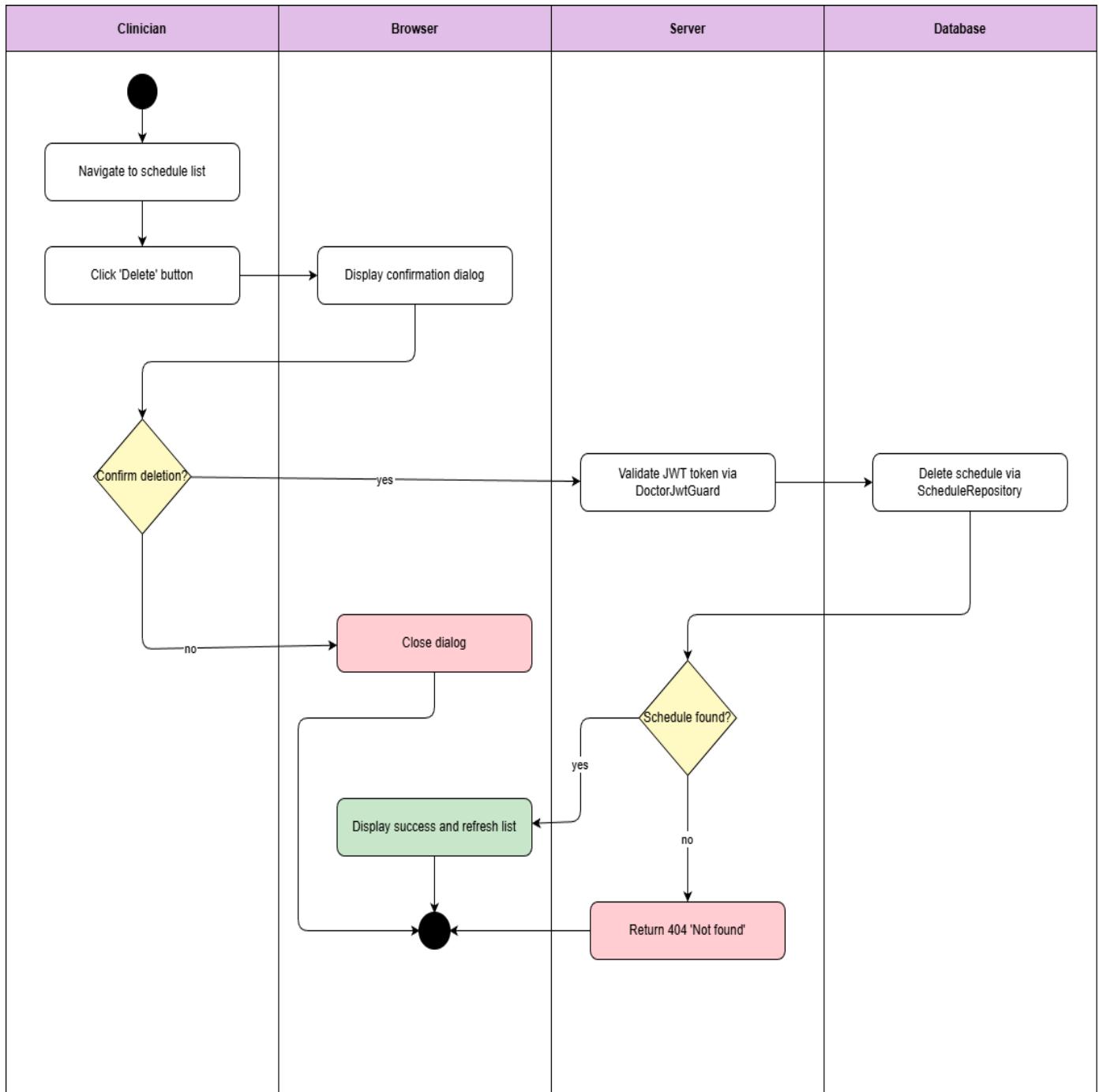


Diagram 116, Activity Diagram (Delete Schedule)

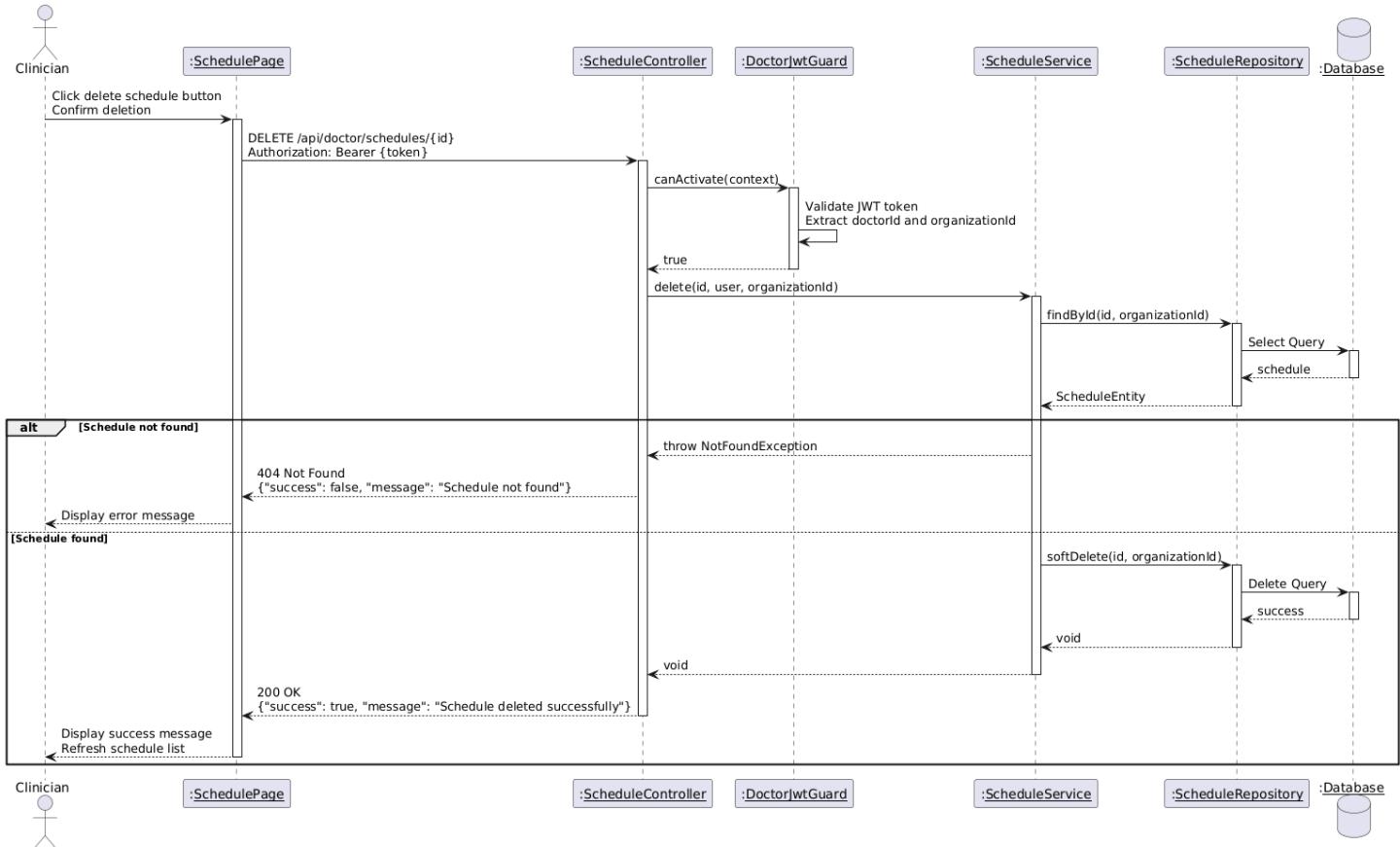


Diagram 117, Activity Diagram (Delete Schedule)

- **Generate Visit Slots:**

Use case ID	VEMR-FR-SM-57
Use case name	Generate Visit Slots
Description	The system allows clinicians to automatically generate available visit slots based on their schedule.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to schedule management page 2. Clinician clicks "Generate Slots" button 3. System validates JWT token via DoctorJwtGuard 4. System retrieves clinician schedules via Schedule Repository 5. System generates time slots based on schedule and slot duration 6. System creates available slots via Slot Repository 7. System displays success message with generated slots count
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: No Schedule Found</p> <ul style="list-style-type: none"> - At step 4, if schedule ID does not exist - System displays "Please create schedule first" message
Post condition	Available visit slots are generated

Table 64, Use Case Specification (Generate Visit Slots)

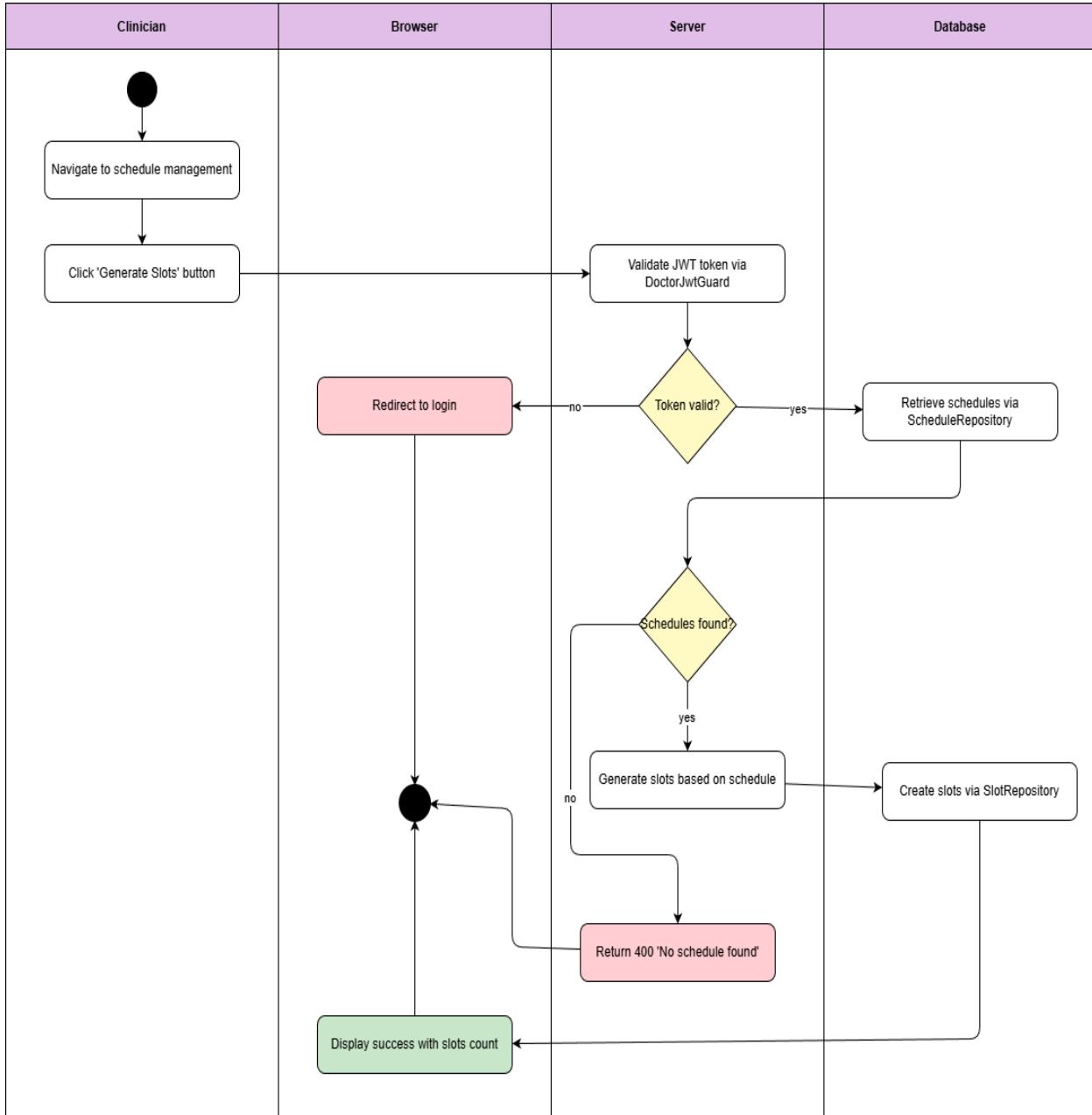


Diagram 118, Activity Diagram (Generate Visit Slots)

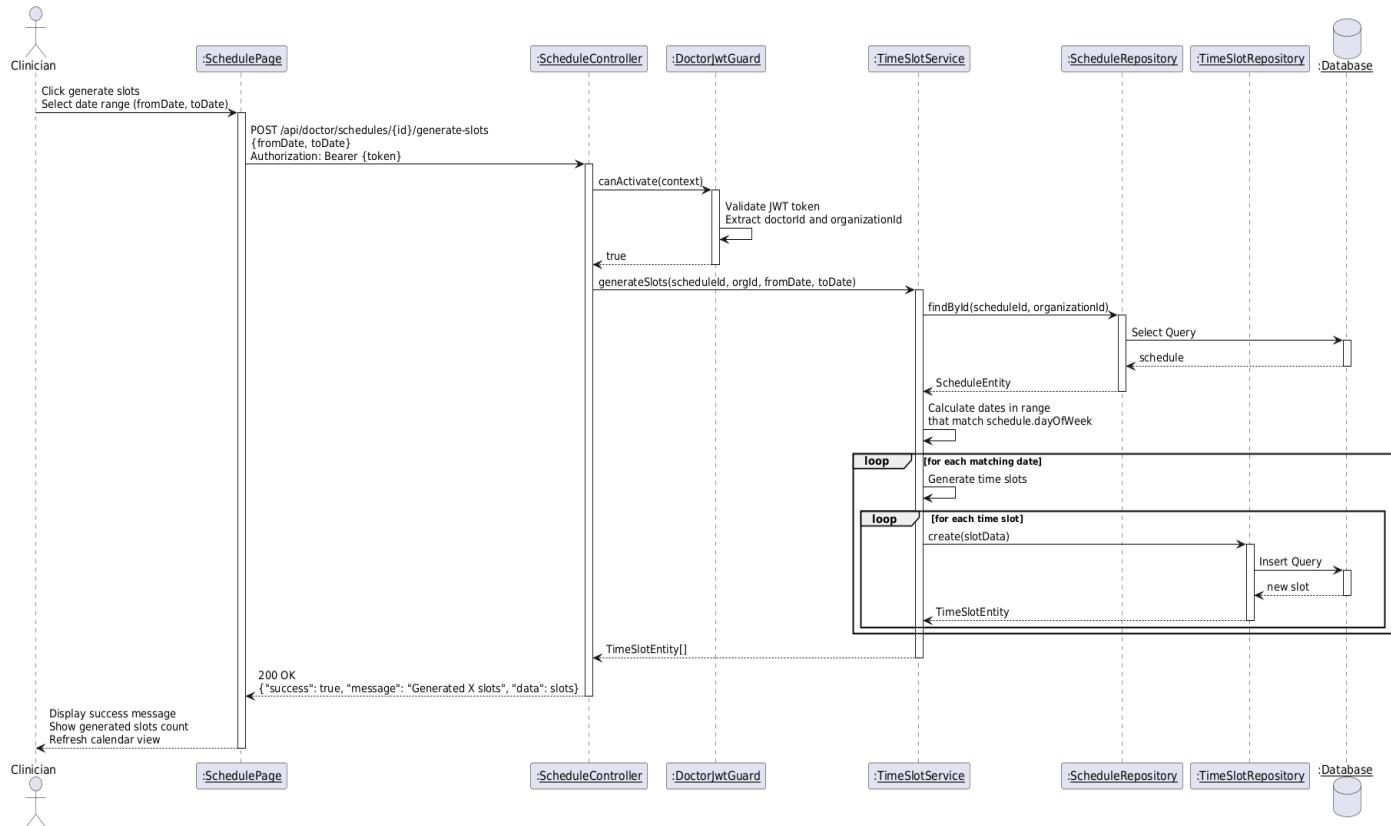


Diagram 119, Sequence Diagram (Generate Visit Slots)

- **Set Appointment Status:**

Use case ID	VEMR-FR-SM-58
Use case name	Set Appointment Status
Description	The system allows clinicians to automatically generate available visit slots based on their schedule.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to schedule management page 2. Clinician clicks "Generate Slots" button 3. System validates JWT token via DoctorJwtGuard 4. System retrieves clinician schedules via Schedule Repository 5. System generates time slots based on schedule and slot duration 6. System creates available slots via Slot Repository 7. System displays success message with generated slots count
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: No Schedule Found</p> <ul style="list-style-type: none"> - At step 4, if schedule ID does not exist - System displays "Please create schedule first" message
Post condition	Available visit slots are generated

Table 65, Use Case Specification (Set Appointment Status)

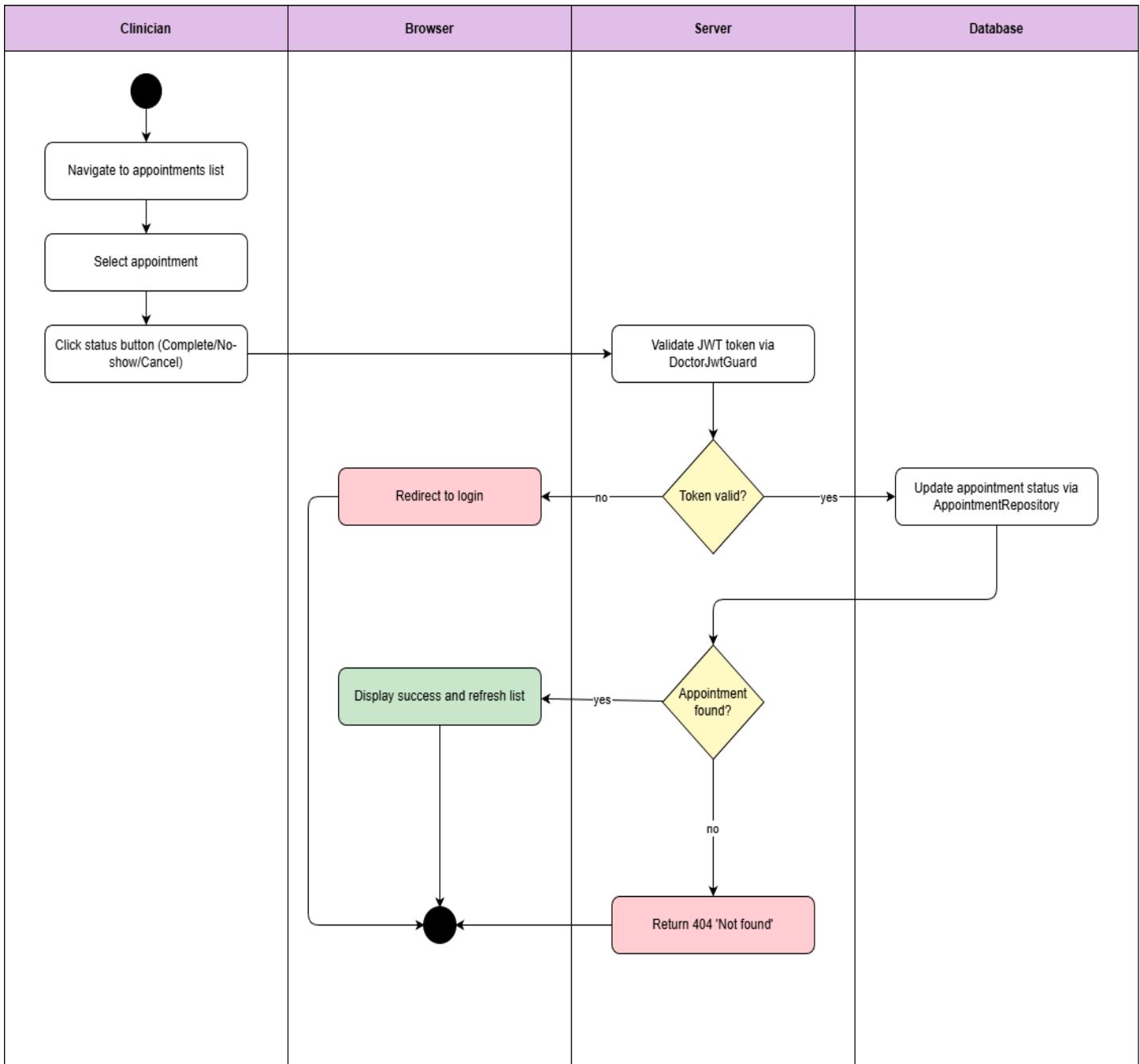


Diagram 120, Activity Diagram (Set Appointment Status)

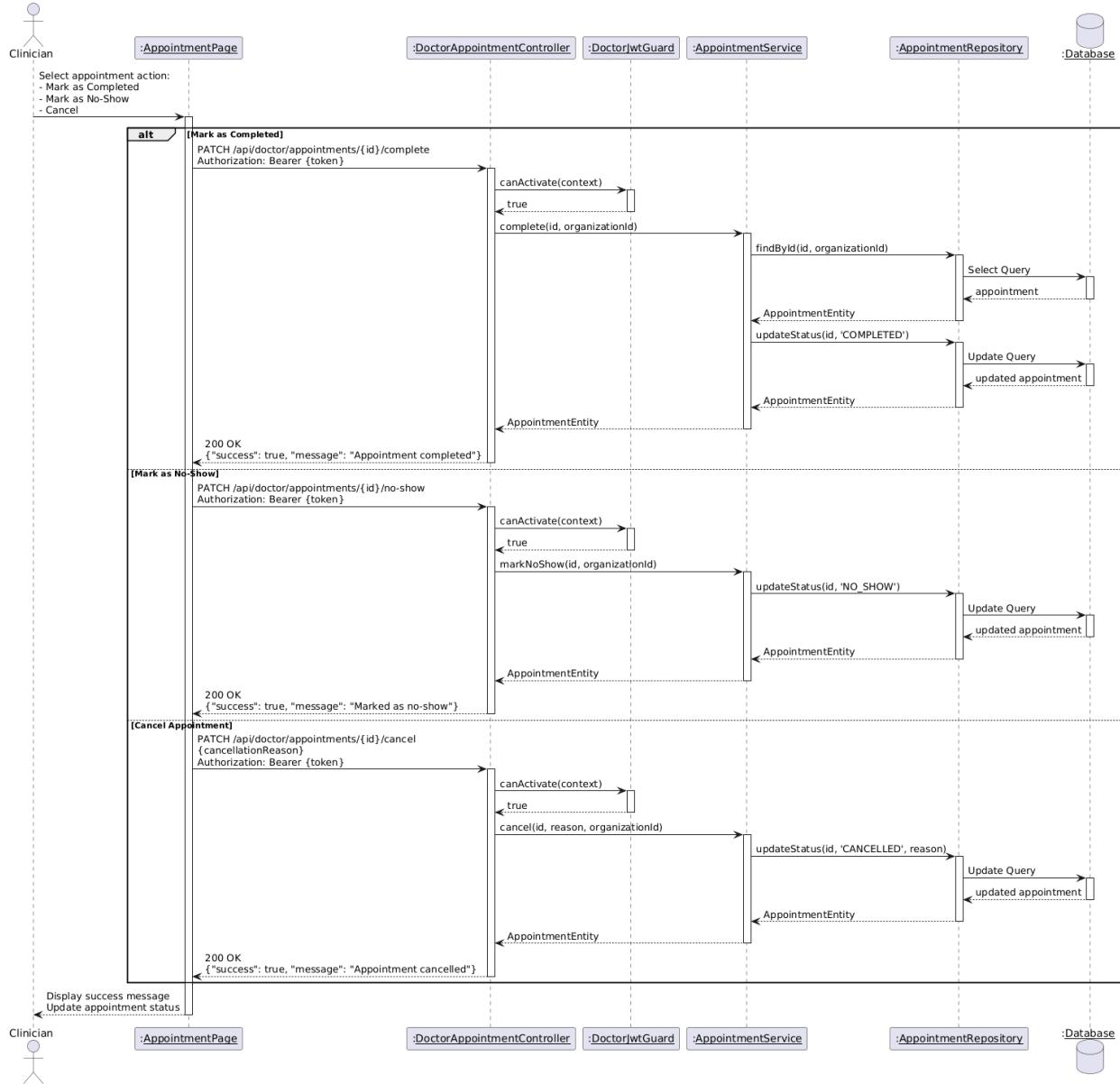


Diagram 121, Sequence Diagram (Set Appointment Status)

- **Set Available Time Slots:**

Use case ID	VEMR-FR-SM-59
Use case name	Set Available Time Slots
Description	The system allows clinicians to define their available time slots for appointments.
From	Clinician
Pre-conditions	Clinician is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Clinician navigates to availability management page 2. Clinician selects date and time range 3. Clinician sets slot duration 4. Clinician submits availability form 5. System validates JWT token via DoctorJwtGuard 6. System creates time slots via Slot Repository 7. System displays success message with created slots
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 5, if JWT token is invalid or expired - System redirects to login page <p>A2: Validation Error</p> <ul style="list-style-type: none"> - At step 4, if required fields are empty or invalid - System displays validation error messages <p>A3: Overlapping Slots</p> <ul style="list-style-type: none"> - At step 6, if slots overlap with existing slots - System displays error message
Post condition	Available time slots are generated

Table 66, Use Case Specification (Set Available Time Slots)

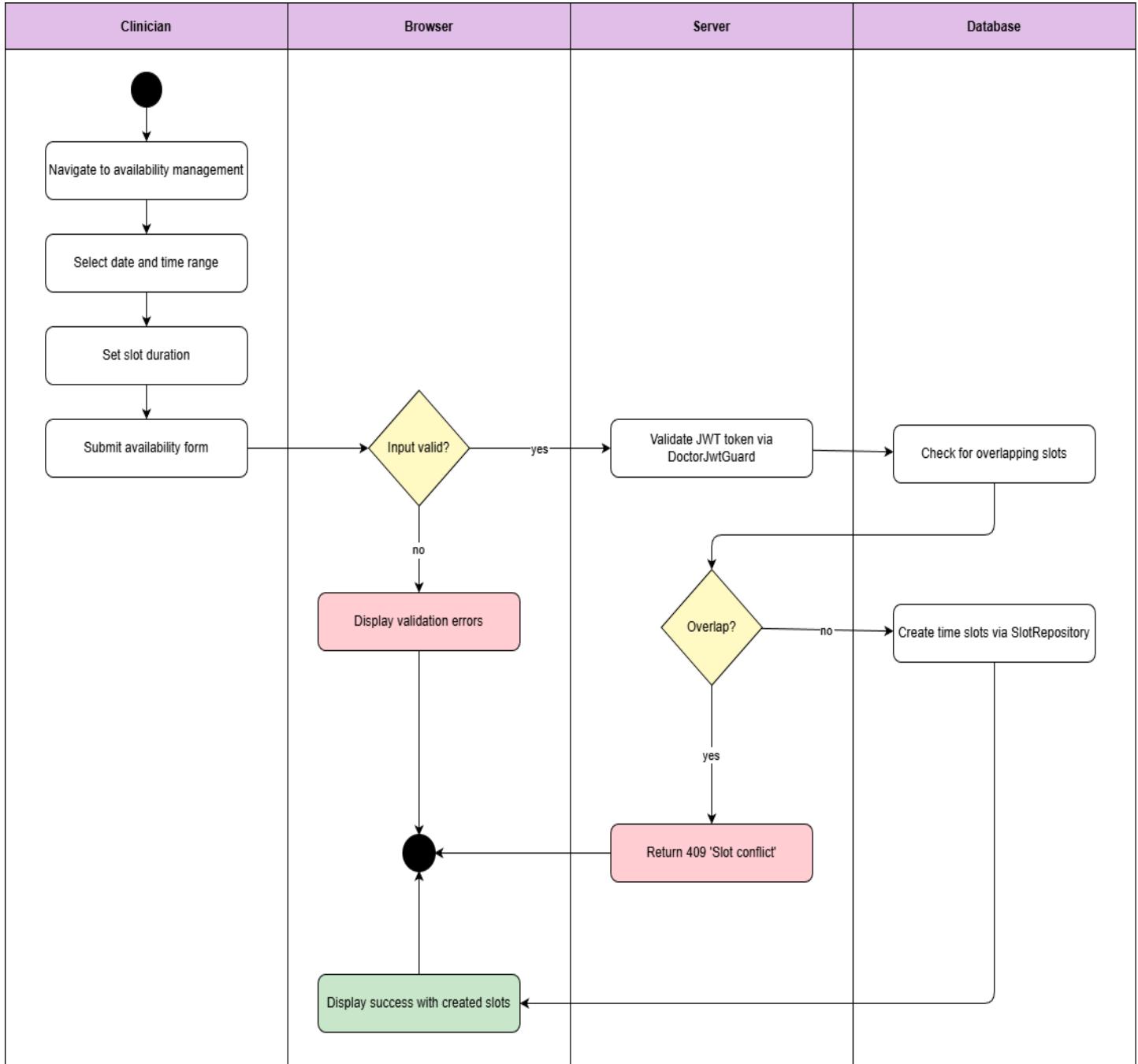


Diagram 122, Activity Diagram (Set Available Time Slot)

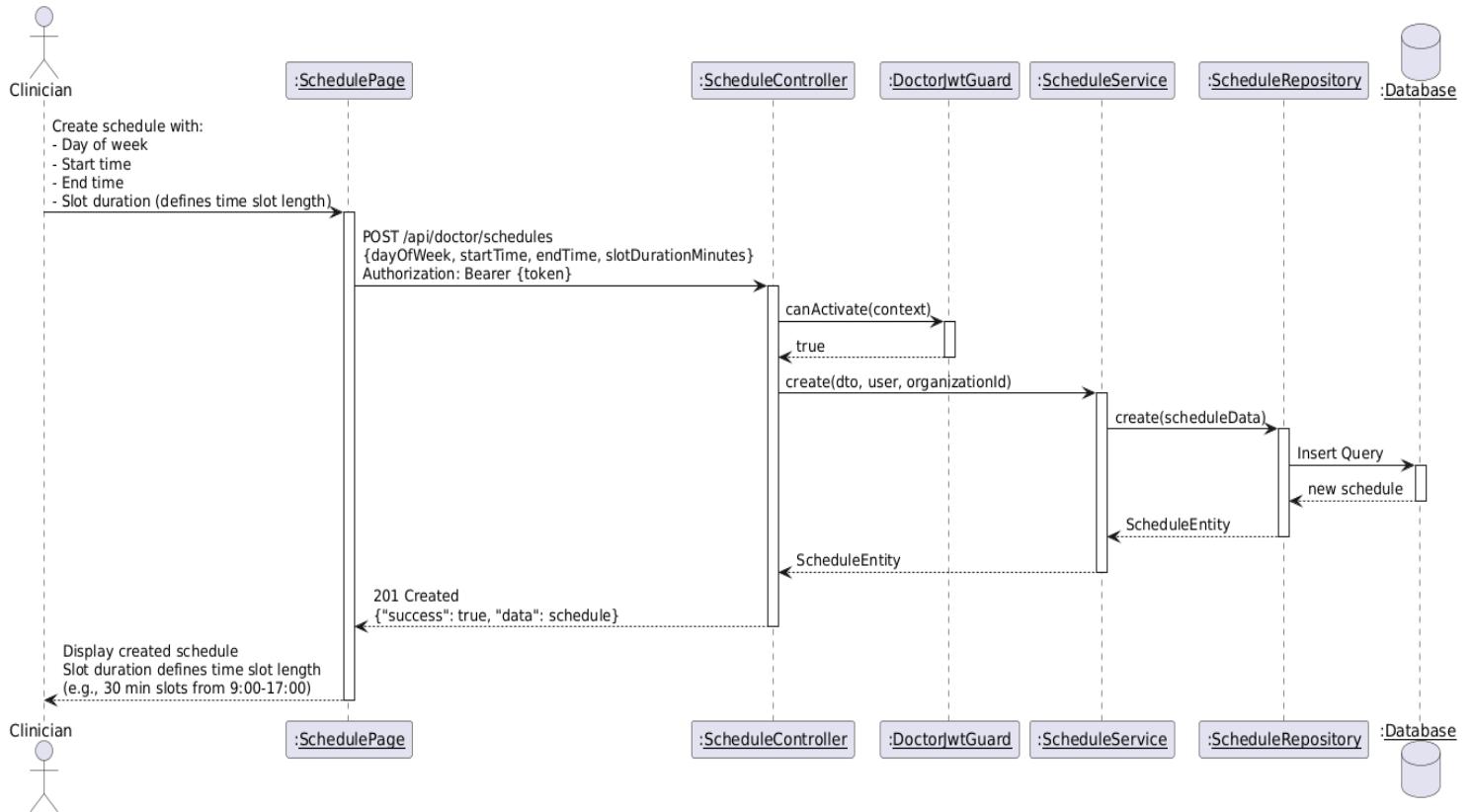


Diagram 123, Sequence Diagram (Set Available Time Slots)

- **View Organizations:**

Use case ID	VEMR-FR-PP-60
Use case name	View Organizations
Description	The system allows patients to view available healthcare organizations
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to organizations page 2. System validates JWT token via PatientJwtGuard 3. System retrieves organizations list via Organization Repository 4. System displays organizations with details
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired - System redirects to login page <p>A2: No Organizations Found</p> <ul style="list-style-type: none"> - At step 3, if no organizations exist - System displays "No organizations available" message <p>A3: Overlapping Slots</p> <ul style="list-style-type: none"> - At step 6, if slots overlap with existing slots - System displays "No organizations available" message
Post condition	Patient can view available healthcare organizations

Table 67, Use Case Specification (View Organizations)

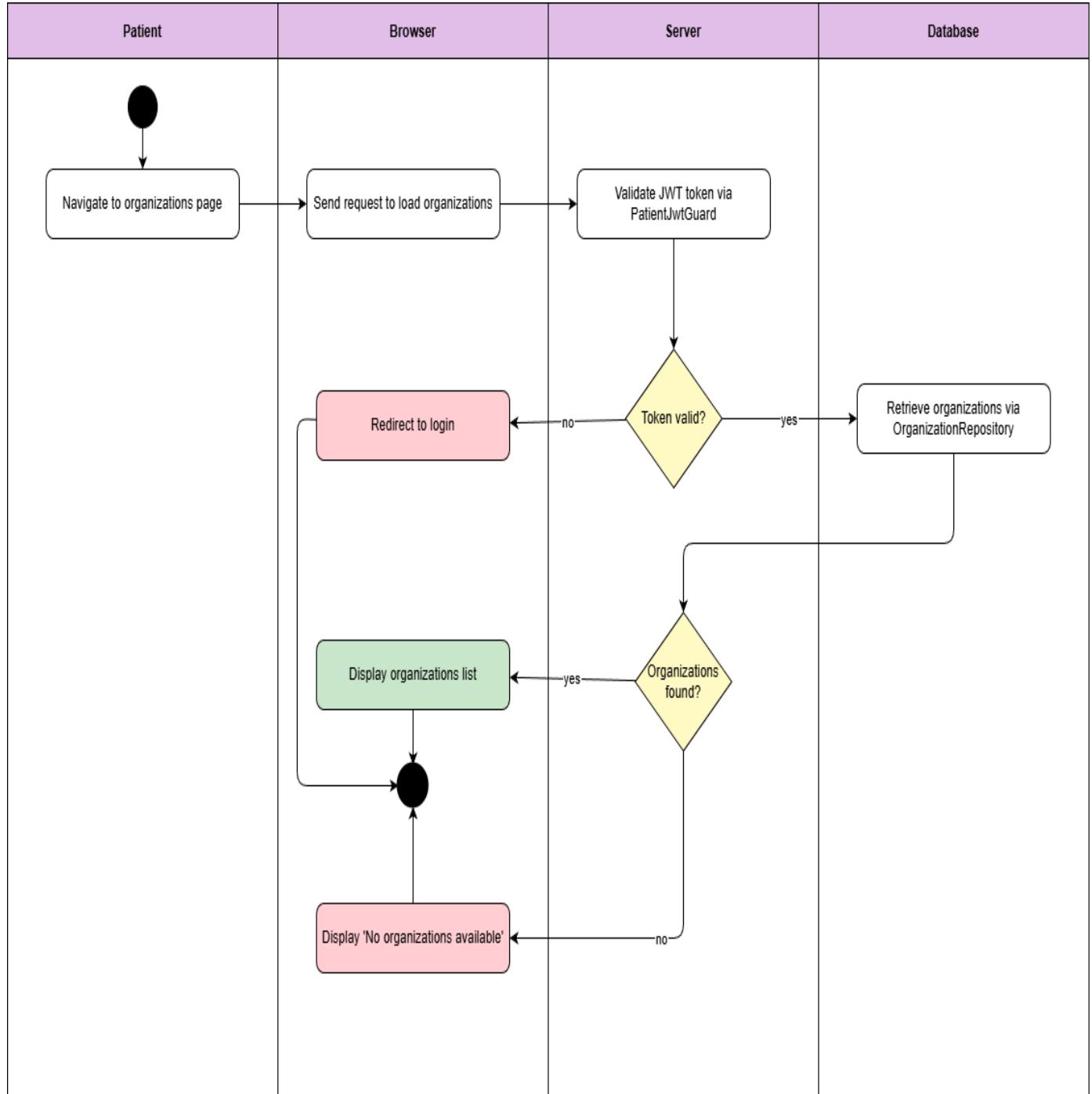


Diagram 124, Activity Diagram (View Organizations)

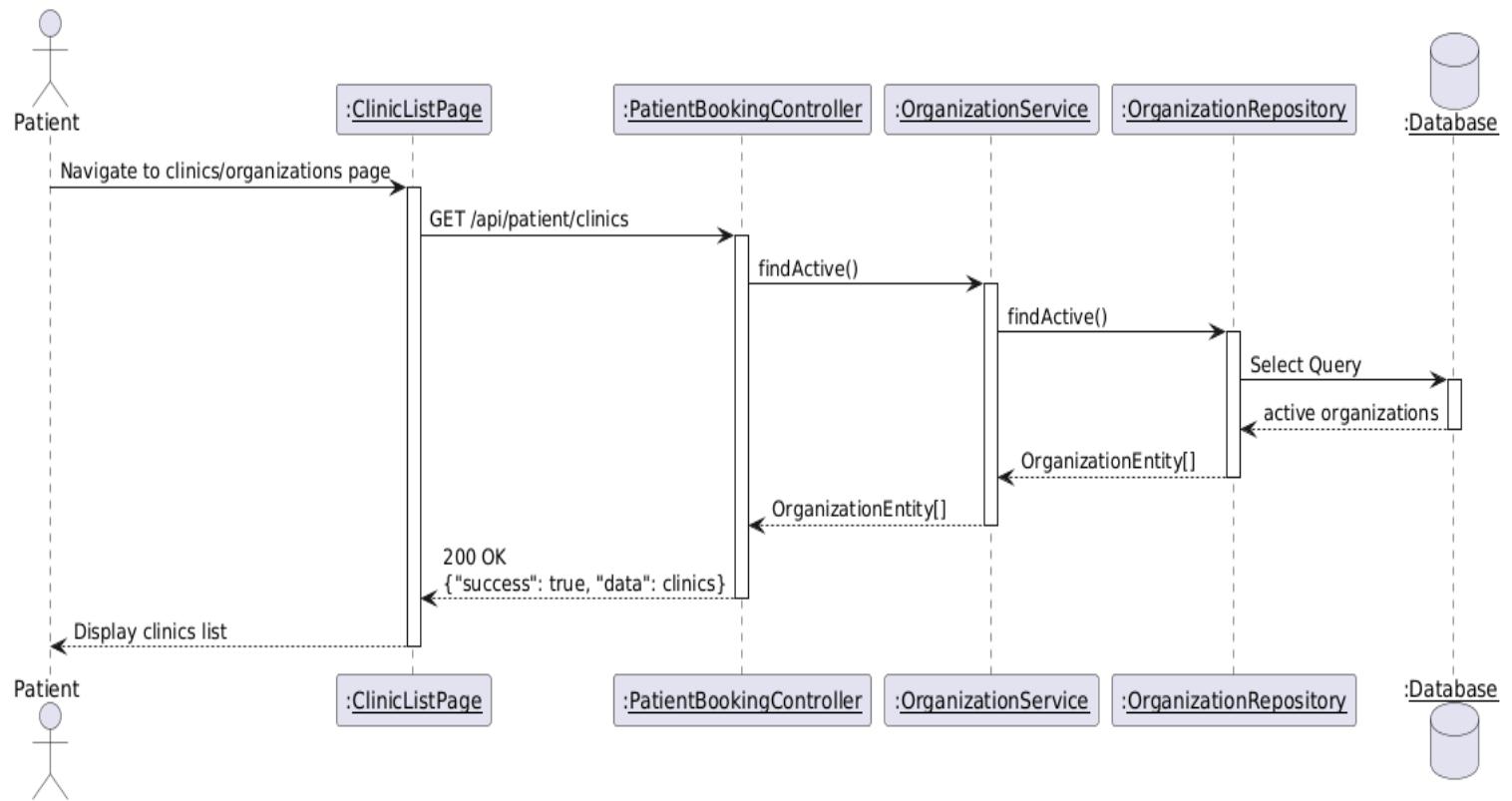


Diagram 125, Sequence Diagram (View Organizations)

- **View Doctor In Organizations:**

Use case ID	VEMR-FR-PP-61
Use case name	View Doctors in Organizations
Description	The system allows patients to view doctors within specific organizations
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to organizations page 2. System validates JWT token via PatientJwtGuard 3. System retrieves organizations list via Organization Repository 4. System displays organizations with details
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: No Doctors Found</p> <ul style="list-style-type: none"> - At step 4, if organization has no doctors - System displays "No doctors available in this organization" message
Post condition	Patient can view available doctors

Table 68, Use Case Specification (View Doctors in Organizations)

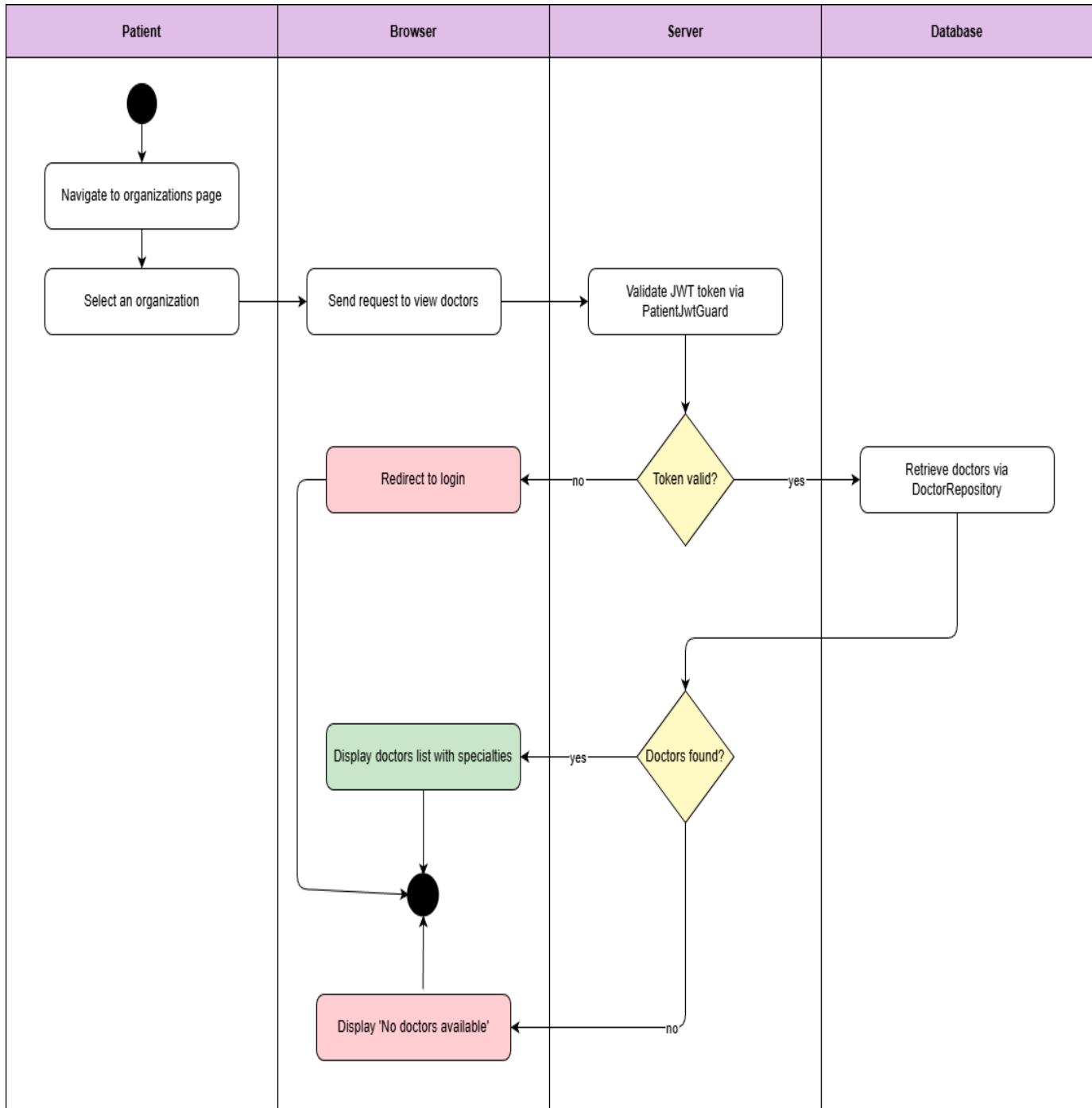


Diagram 126, Activity Diagram (View Doctors in Organizations)

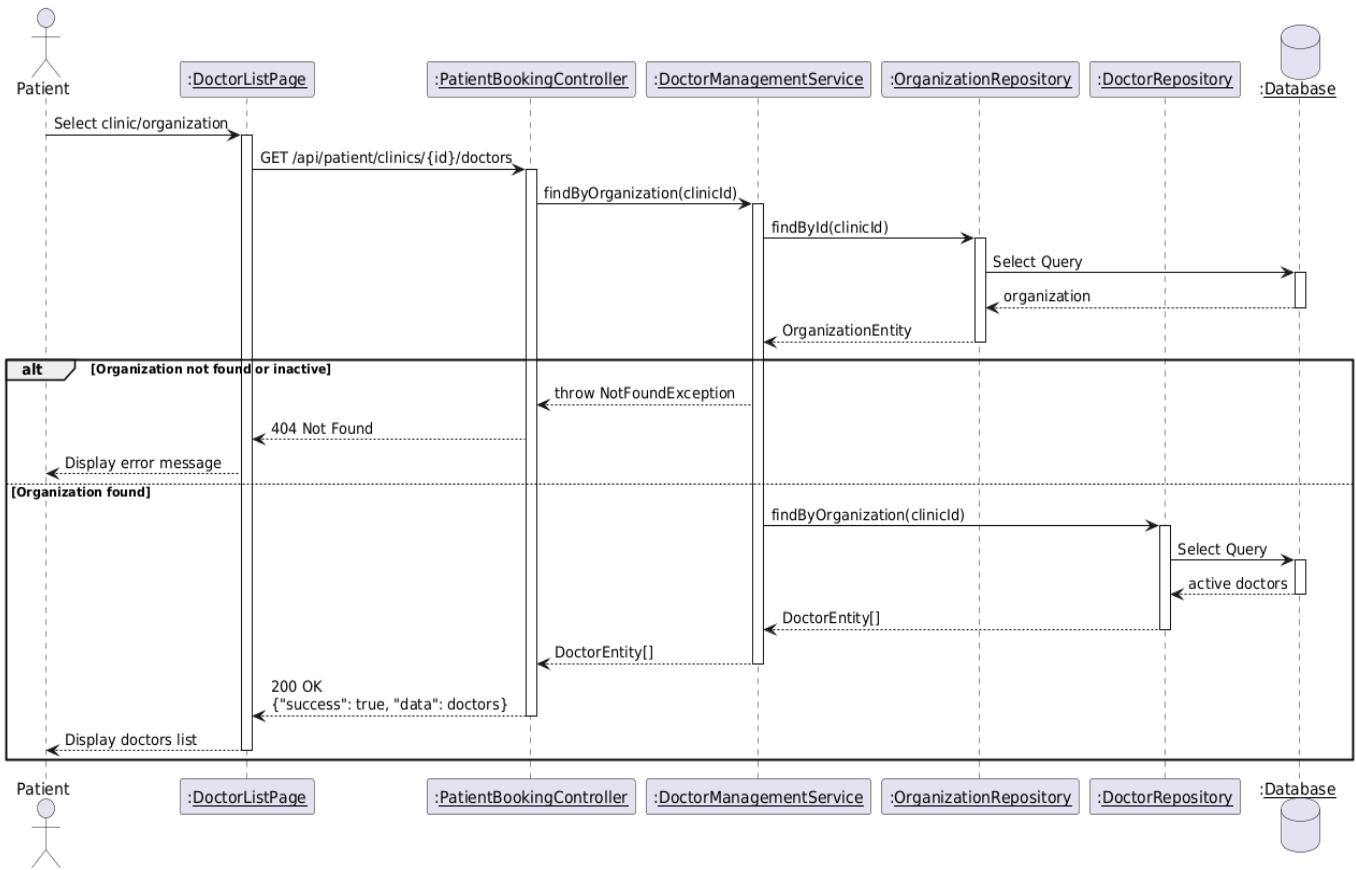


Diagram 127, Activity Diagram (View Doctors in Organizations)

- **View Available Appointment:**

Use case ID	VEMR-FR-PP-62
Use case name	View Available Appointments
Description	The system allows patients to view available appointment slots
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to doctor profile 2. Patient clicks "View Available Slots" button 3. System validates JWT token via PatientJwtGuard 4. System retrieves available slots via Slot Repository 5. System displays available appointment slots by date and time
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: No Slots Available</p> <ul style="list-style-type: none"> - At step 4, if doctor has no available slots - System displays "No available slots" message
Post condition	Available appointment slots are displayed

Table 69, Use Case Specification (View Available Appointment)

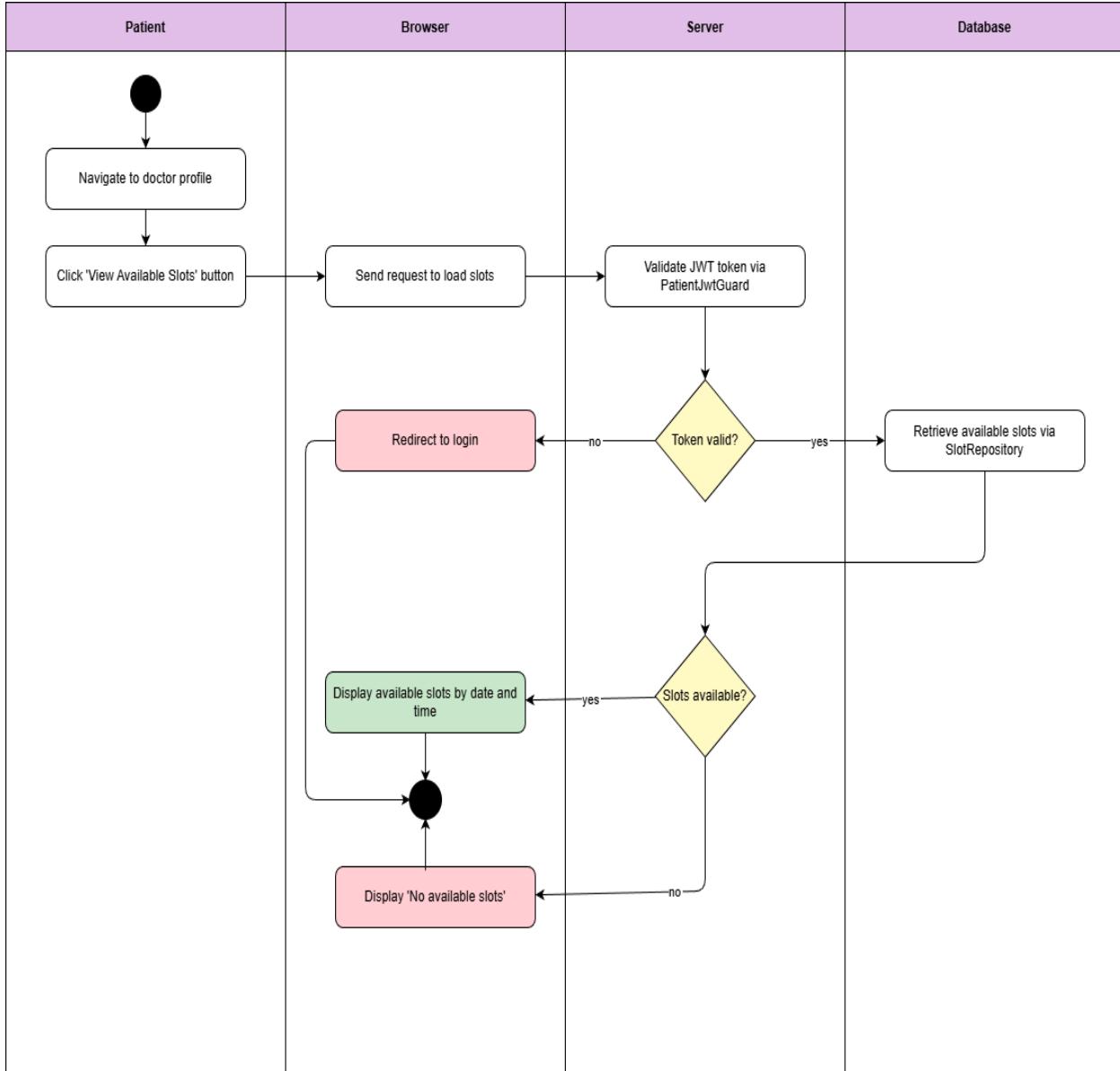


Diagram 128, Activity Diagram (View Available Appointment)

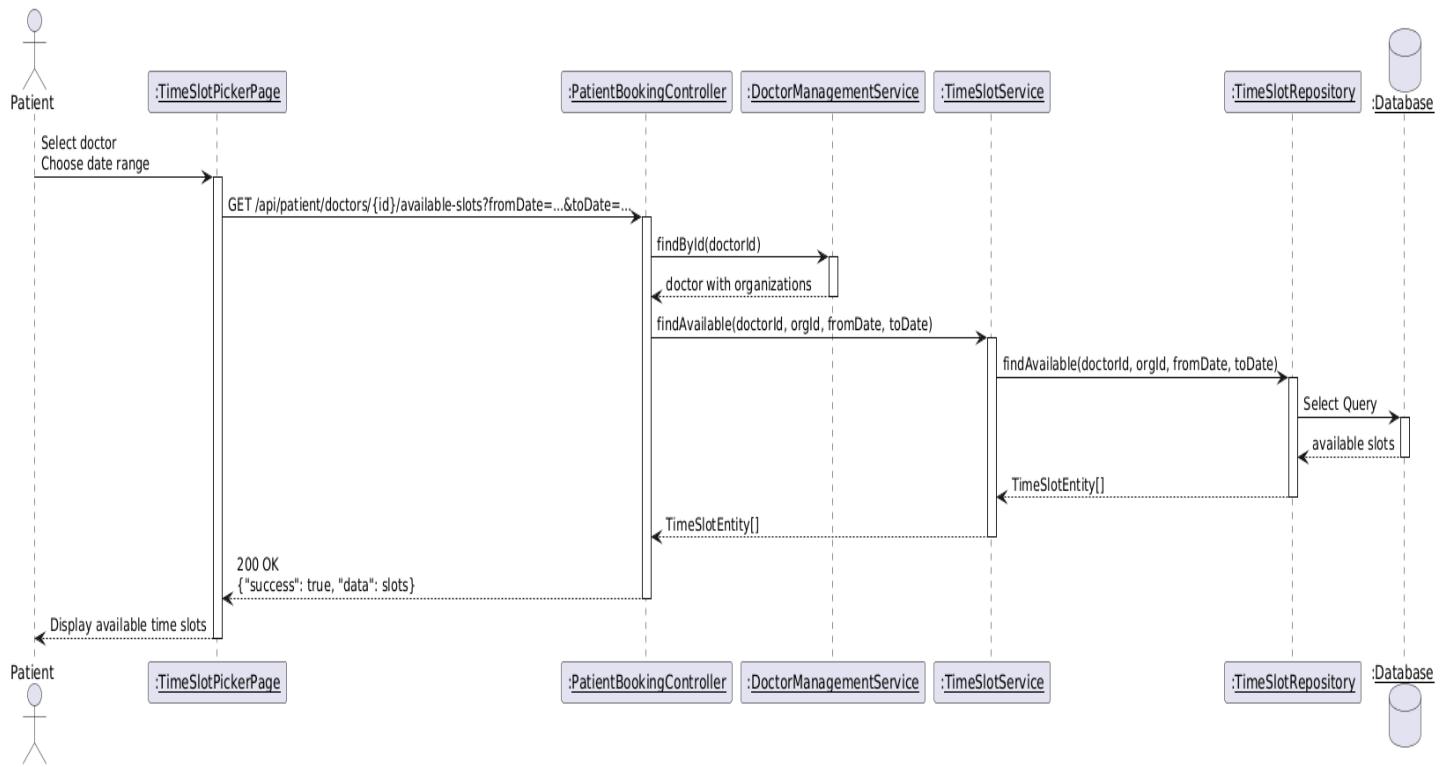


Diagram 129, Sequence Diagram (View Available Appointment)

● Book an Appointment:

Use case ID	VEMR-FR-PP-63
Use case name	Book an Appointment
Description	The system allows patients to book an appointment with a clinician.
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient views available appointment slots 2. Patient selects desired slot 3. Patient enters appointment reason 4. Patient confirms booking 5. System validates JWT token via PatientJwtGuard 6. System checks slot availability via Slot Repository 7. System creates appointment via Appointment Repository 8. System marks slot as booked 9. System displays success message with appointment details
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 5 , if JWT token is invalid or expired -System redirects to login page <p>A2: Slot No Longer Available</p> <ul style="list-style-type: none"> - At step 6, if slot is already booked - System displays error message <p>A3: Validation Error</p> <ul style="list-style-type: none"> - At step 3, if appointment reason is empty - System displays validation error
Post condition	Slot is marked as booked

Table 70, Use Case Specification (Book an Appointment)

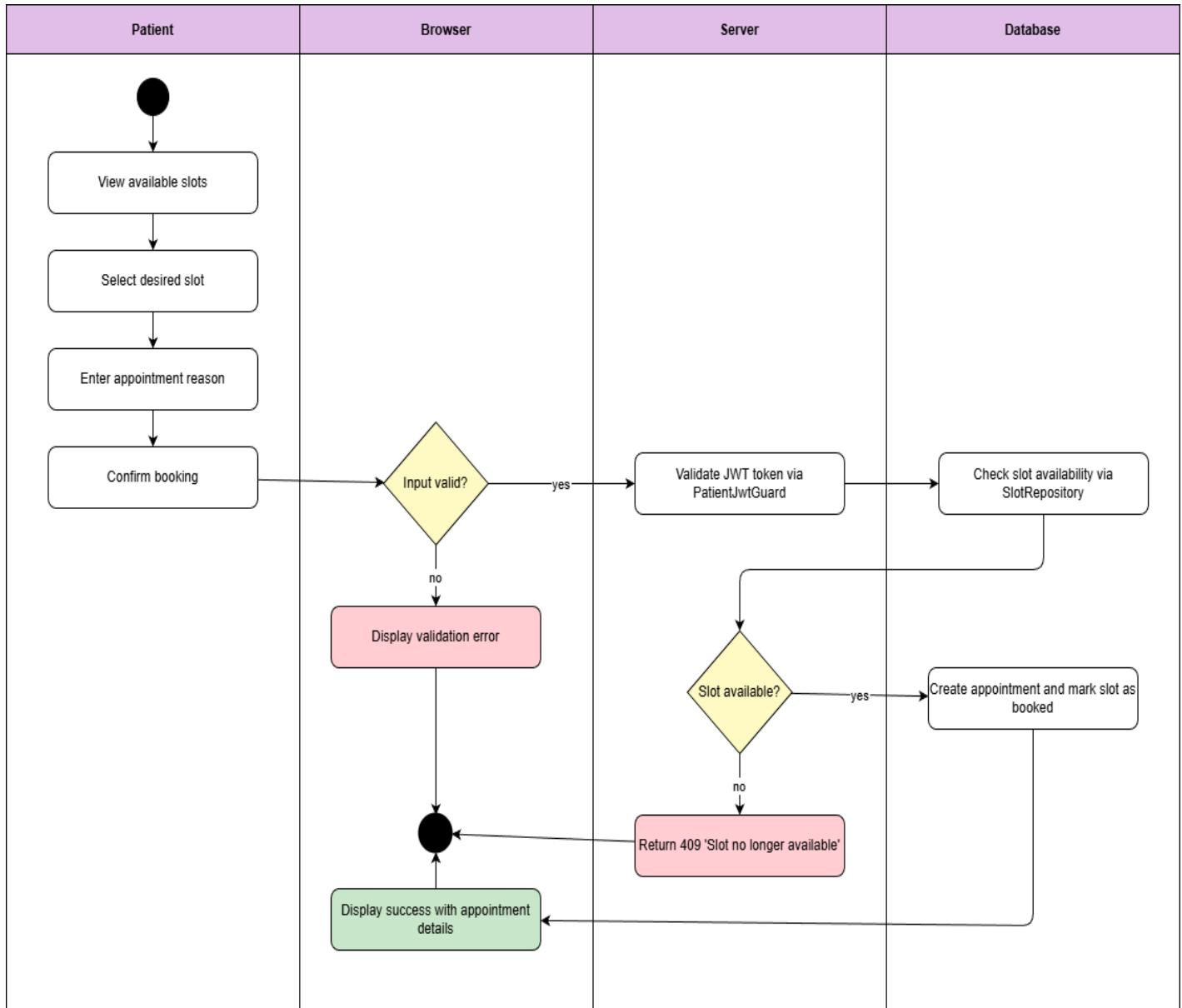


Diagram 130, Activity Diagram (Book an Appointment)

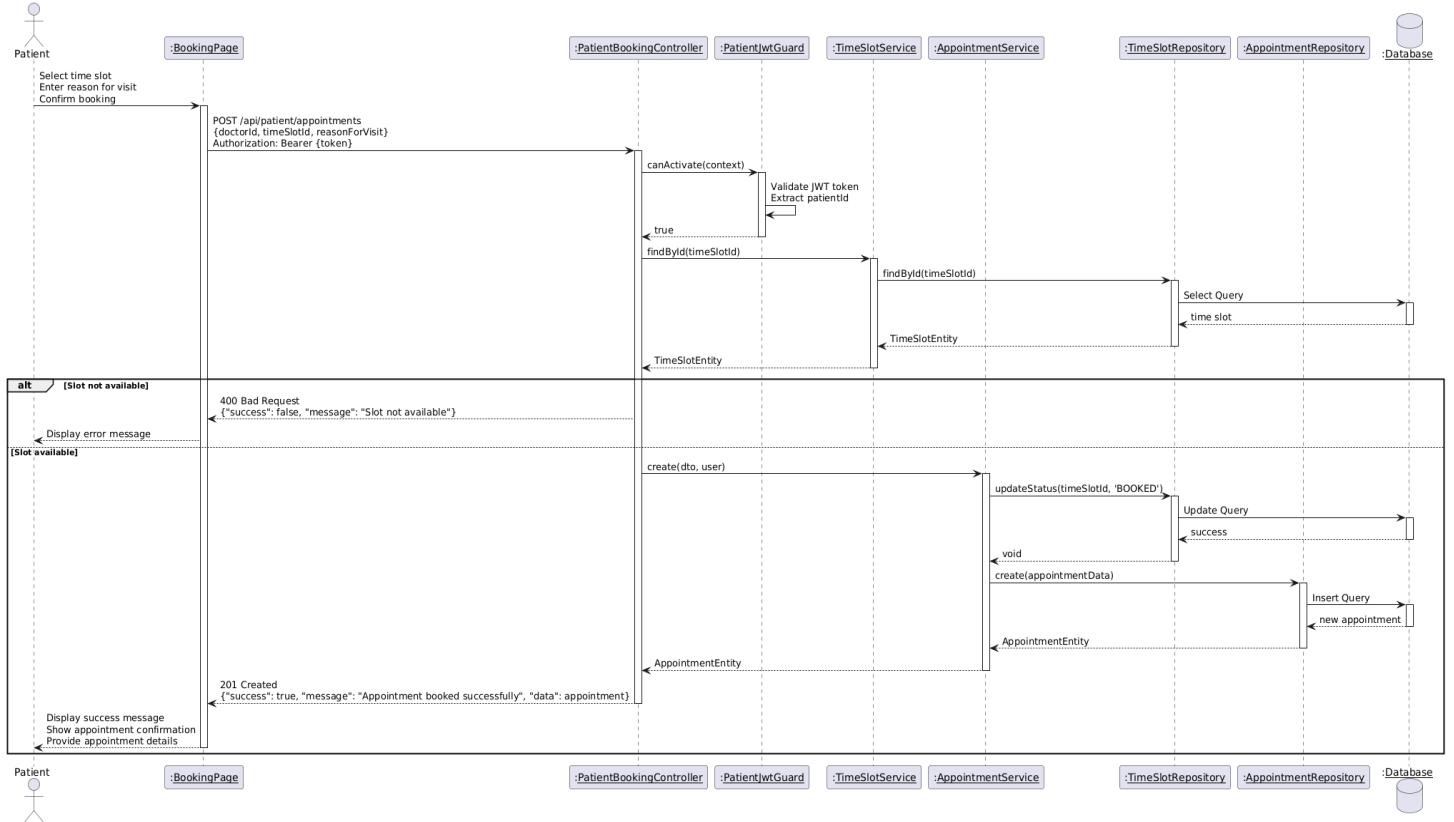


Diagram 131, Sequence Diagram (Book an Appointment)

- **View Own Appointments:**

Use case ID	VEMR-FR-PP-64
Use case name	View Own Appointments
Description	The system allows patients to view their own appointment history and upcoming appointments.
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to my appointments page 2. System validates JWT token via PatientJwtGuard 3. System retrieves patient appointments via Appointment Repository 4. System displays appointments with doctor name, date, time, and status
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired - System redirects to login page <p>A2: No Appointments Found</p> <ul style="list-style-type: none"> - At step 3, if patient has no appointments - System returns empty list
Post condition	Patient can view appointment history

Table 71, Use Case Specification (View Own Appointments)

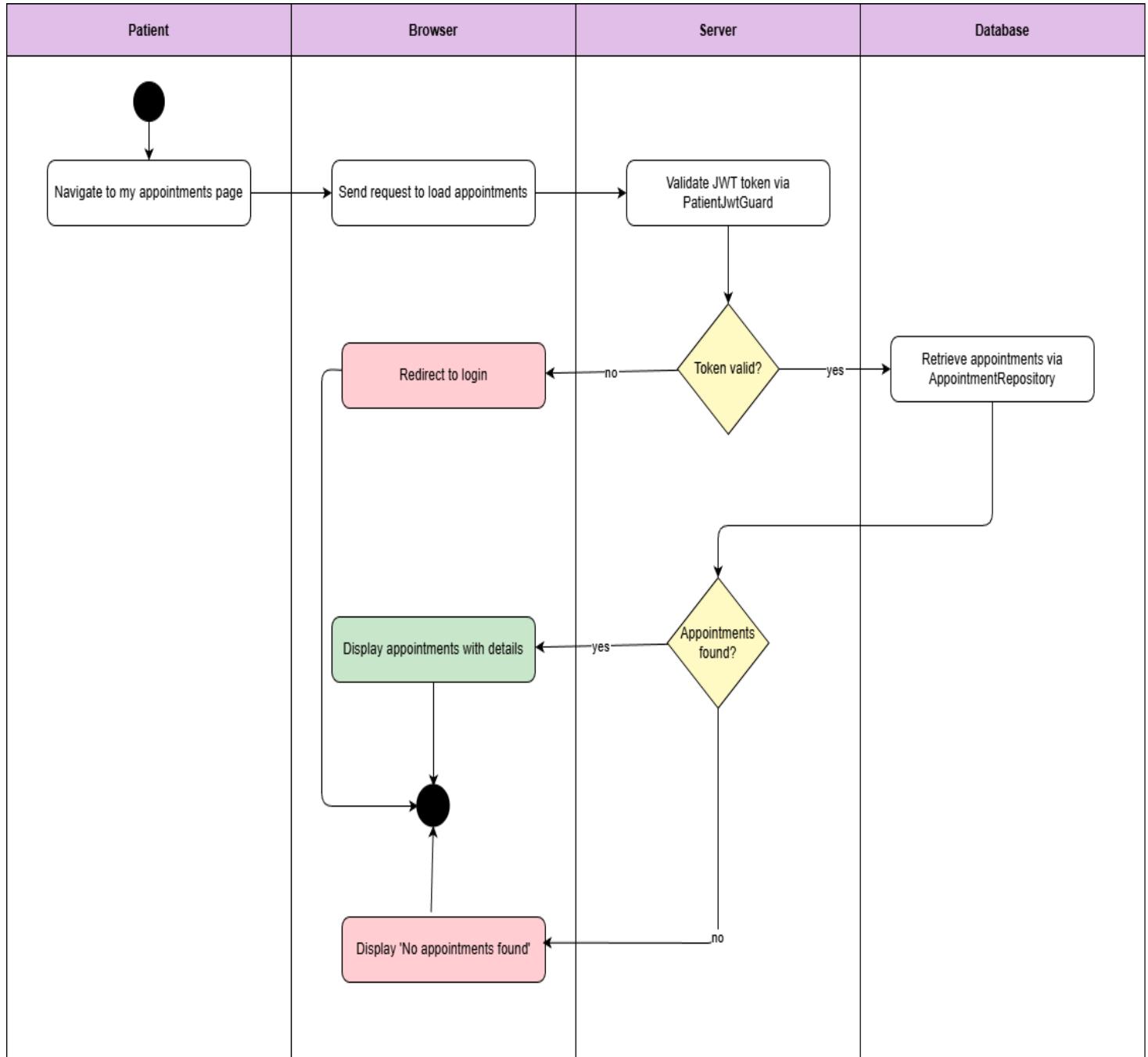


Diagram 132, Activity Diagram (View Own Appointments)

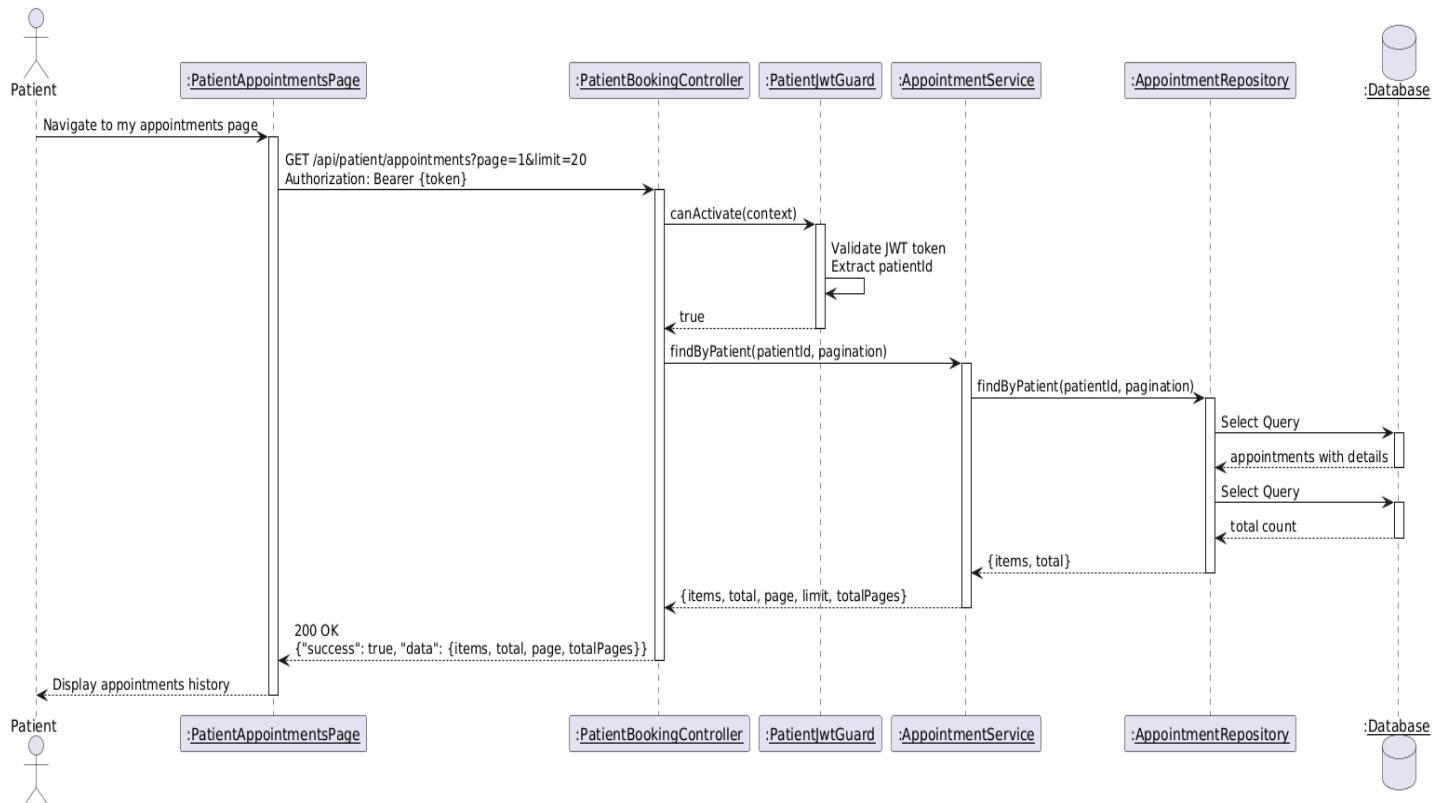


Diagram 133, Sequence Diagram (View Own Appointments)

- **Filter Appointment by Status:**

Use case ID	VEMR-FR-PP-65
Use case name	Filter Appointment by Status
Description	The system allows patients to view their own appointment history and upcoming appointments.
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to my appointments page 2. Patient selects status from filter dropdown 3. System validates JWT token via PatientJwtGuard 4. System retrieves appointments with selected status via Appointment Repository 5. System displays filtered appointments
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 3, if JWT token is invalid or expired - System redirects to login page <p>A2: No Appointments with Selected Status</p> <ul style="list-style-type: none"> - At step 4, if no appointments match the selected status - System displays "No appointments found with this status" message
Post condition	Appointments are filtered by selected status

Table 72, Use Case Specification (Filter Appointment by Status)

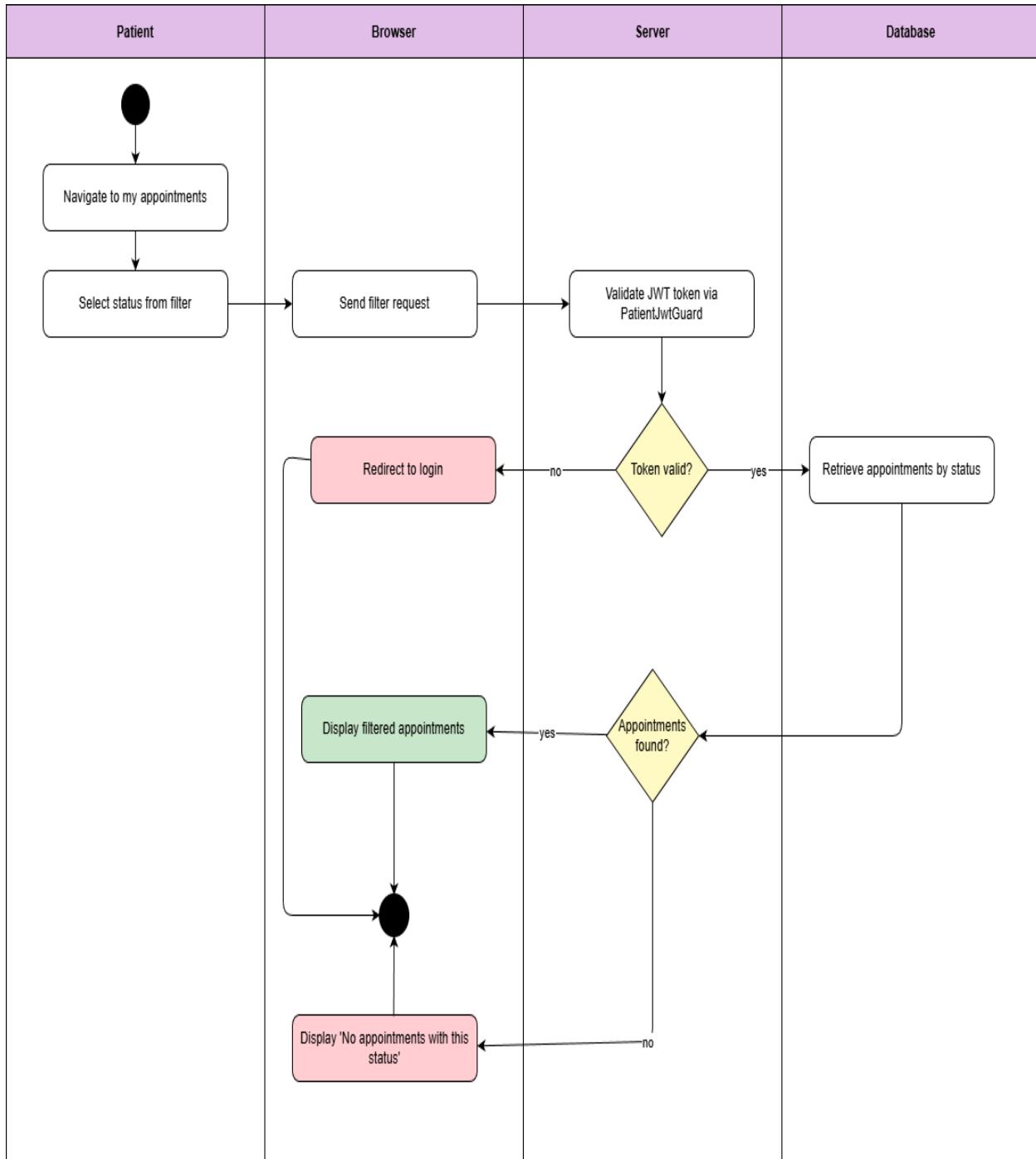


Diagram 134, Activity Diagram (Filter Appointment by Status)

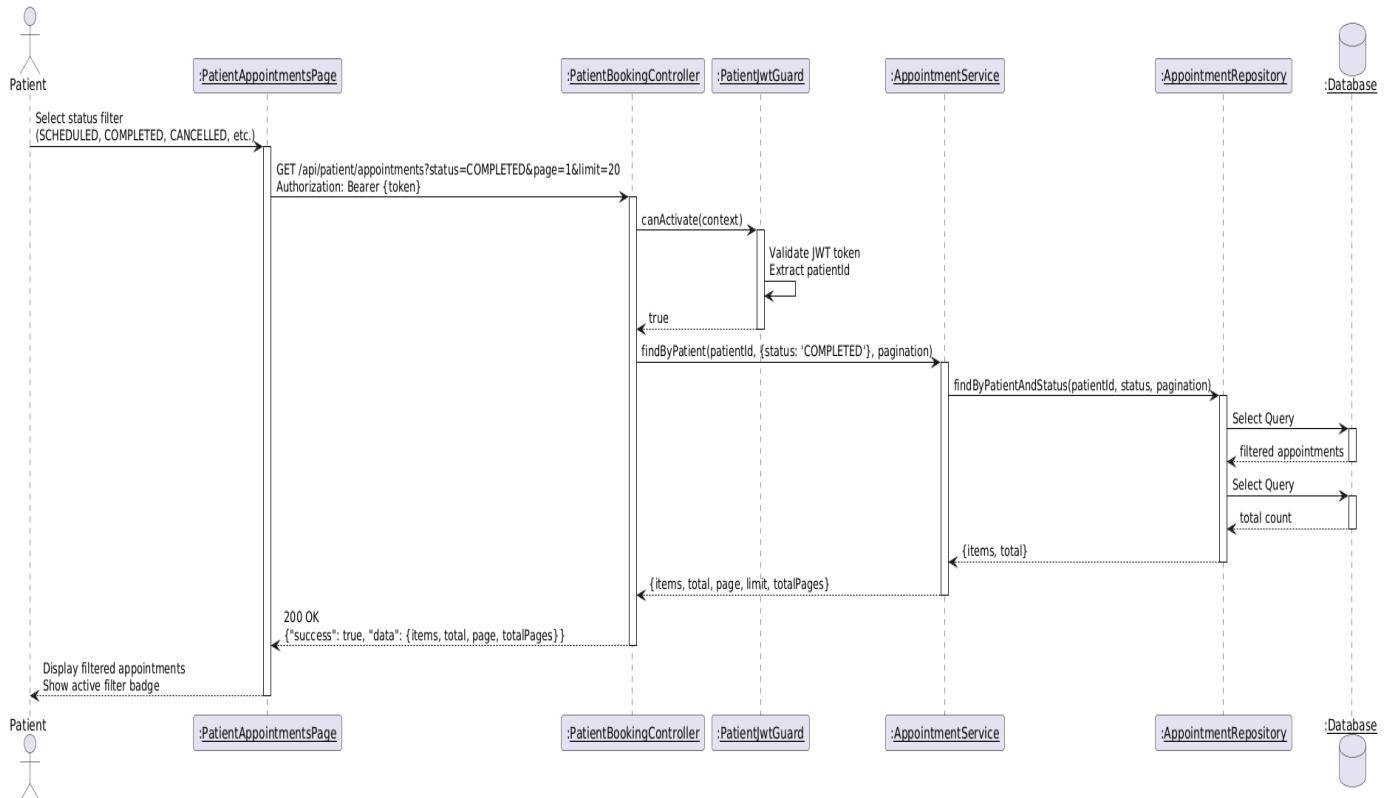


Diagram 135, Sequence Diagram (Filter Appointment by Status)

- **View Own Medical Record:**

Use case ID	VEMR-FR-PP-66
Use case name	View Own Medical Record
Description	The system allows patients to view their own medical records.
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to medical records page 2. System validates JWT token 3. System retrieves patient medical records 4. System displays medical records with diagnoses and treatments
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2 , if JWT token is invalid or expired -System redirects to login page <p>A2: No Medical Records Found</p> <ul style="list-style-type: none"> - At step 3, if patient has no medical records - System displays "No medical records found" message
Post condition	Patient can view their health information

Table 73, Use Case Specification (View Own Medical Records)

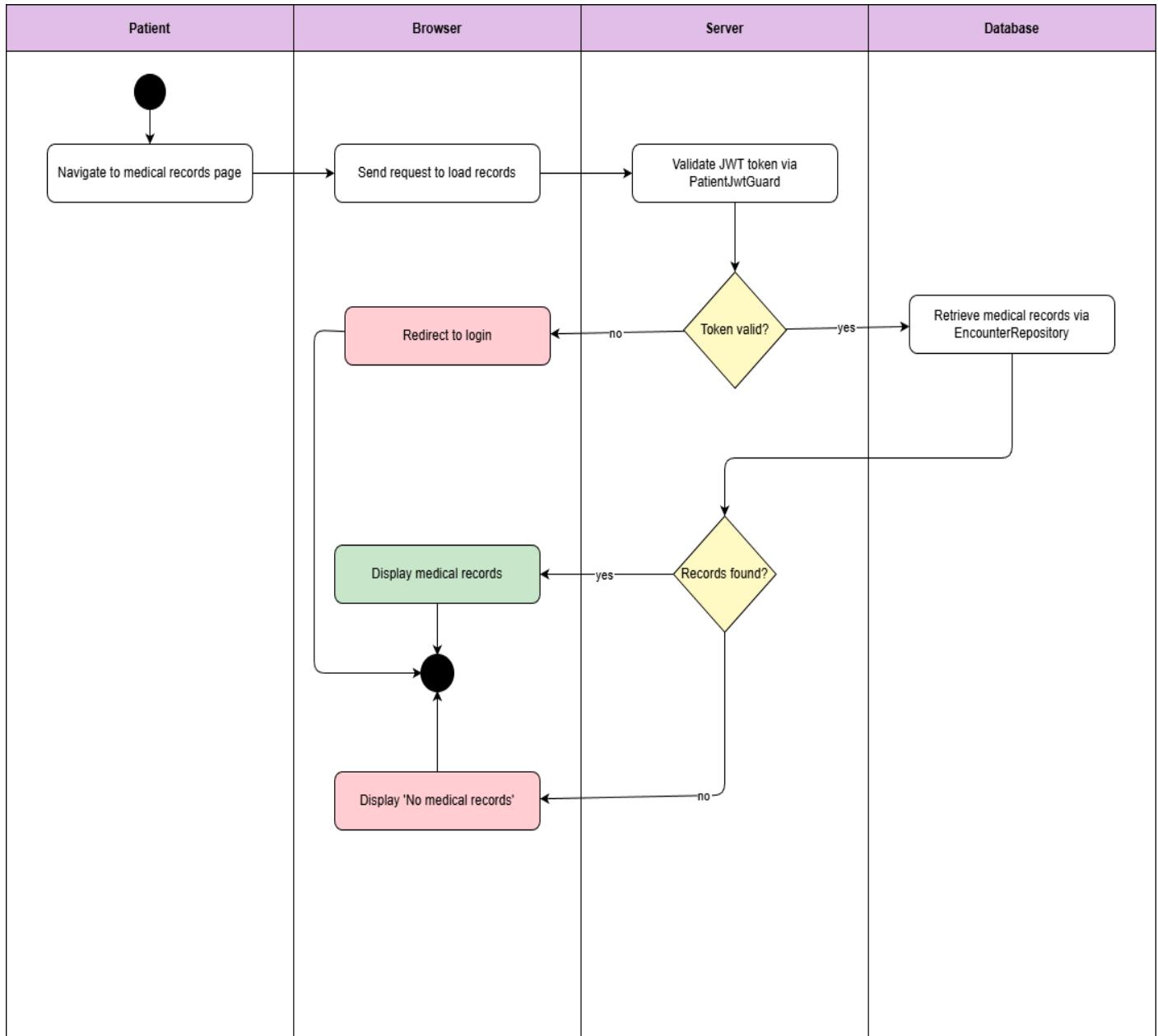


Diagram 136, Activity Diagram (View Own Medical Records)

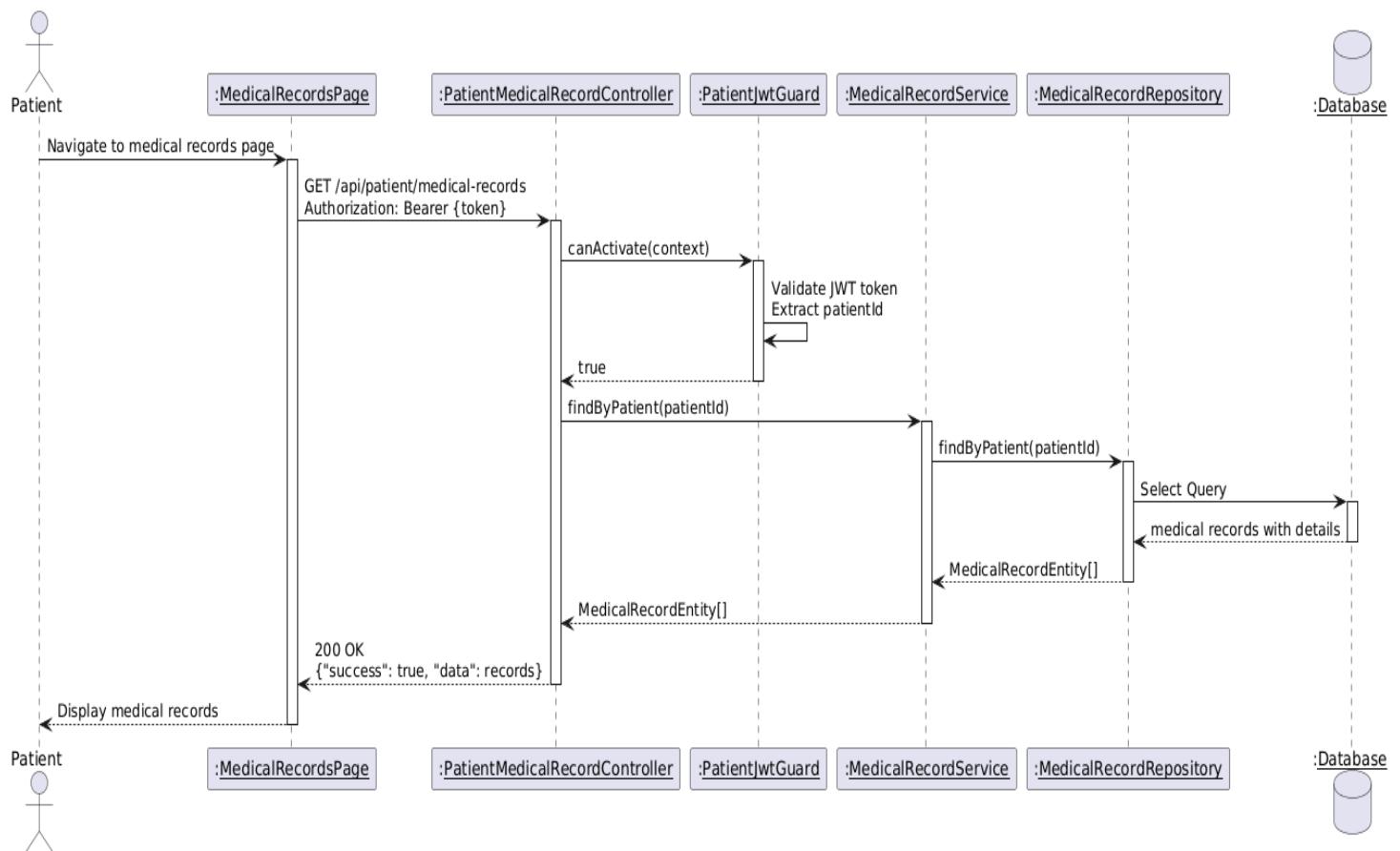


Diagram 137, Sequence Diagram (View Own Medical Records)

- **View Own Allergies:**

Use case ID	VEMR-FR-PP-67
Use case name	View Own Allergies
Description	The system allows patients to view their own allergy information.
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to allergies page 2. System validates JWT token 3. System retrieves patient allergies 4. System displays allergies with type, allergen, severity, and reaction
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2 , if JWT token is invalid or expired -System redirects to login page <p>A2: No Allergies Found</p> <ul style="list-style-type: none"> - At step 3, if patient has no recorded allergies - System displays "No allergies recorded" message
Post condition	Patient can view their allergy information

Table 74, Use Case Specification (View Own Allergies)

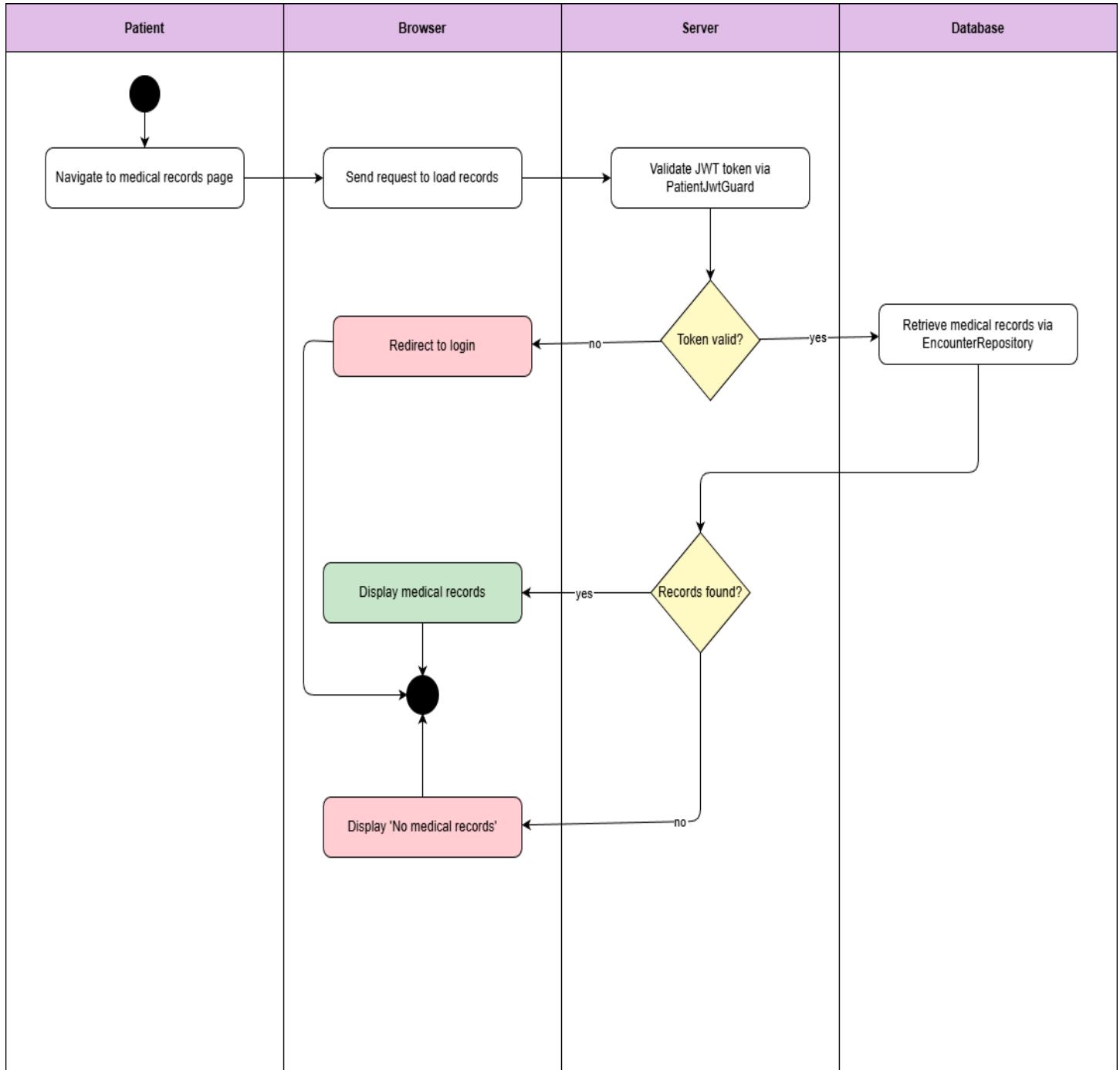


Diagram 138, Activity Diagram (View Own Allergies)

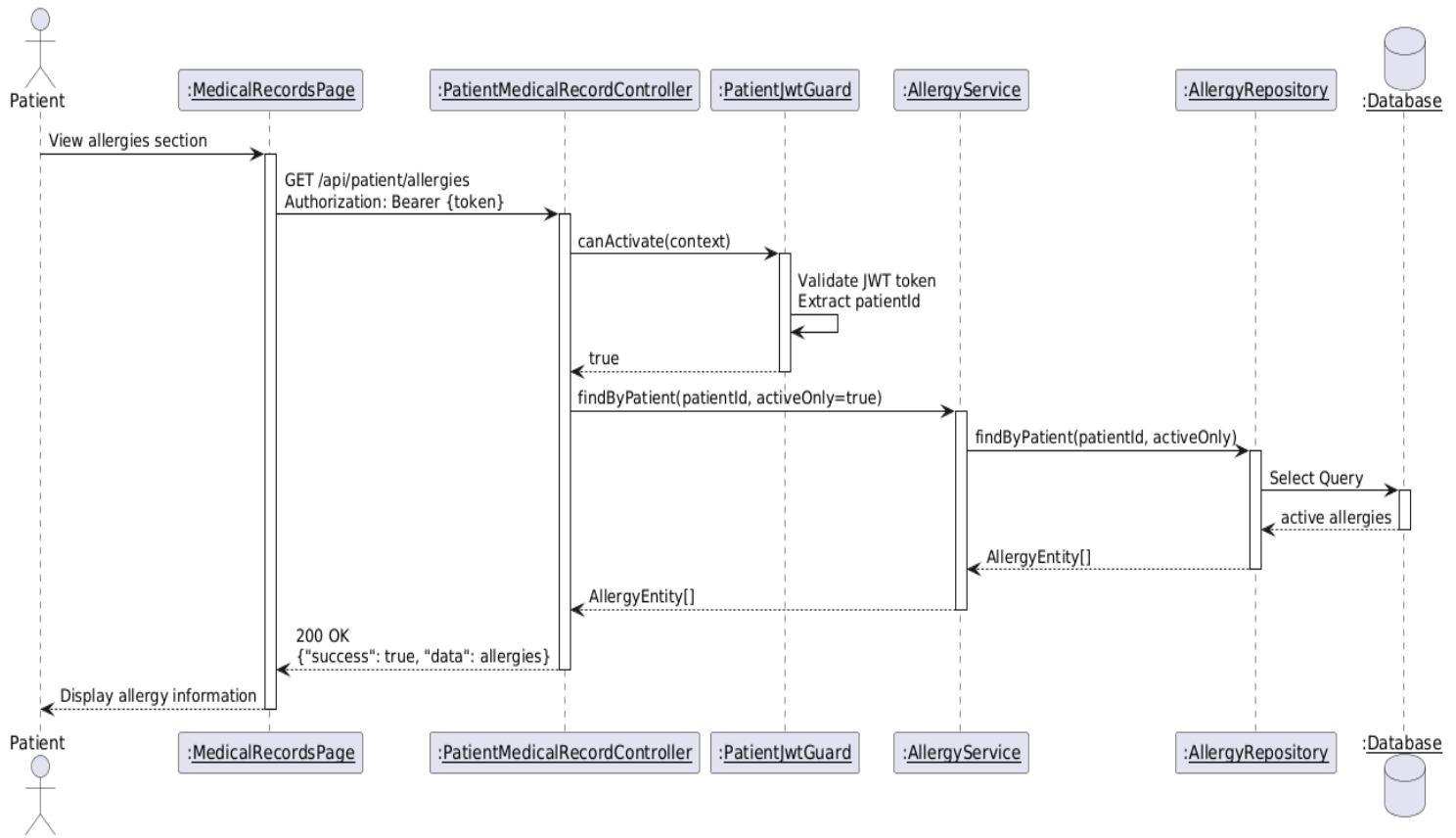


Diagram 139, Sequence Diagram (View Own Allergies)

- **View Own Visits History:**

Use case ID	VEMR-FR-PP -68
Use case name	View Own Visits History
Description	The system allows patients to view their complete visit history.
From	Patient
Pre-conditions	Patient is authenticated
Main scenario	<ol style="list-style-type: none"> 1. Patient navigates to visits history page 2. System validates JWT token 3. System retrieves patient visits 4. System displays visits with date, doctor, reason, and status
Alternative scenario	<p>A1: Unauthorized Access</p> <ul style="list-style-type: none"> - At step 2, if JWT token is invalid or expired - System redirects to login page <p>A2: No Visits Found</p> <ul style="list-style-type: none"> - At step 3, if patient has no visit history - System displays "No visits found" message
Post condition	Patient can view their complete medical visit history

Table 75, Use Case Specification (View Own Visits History)

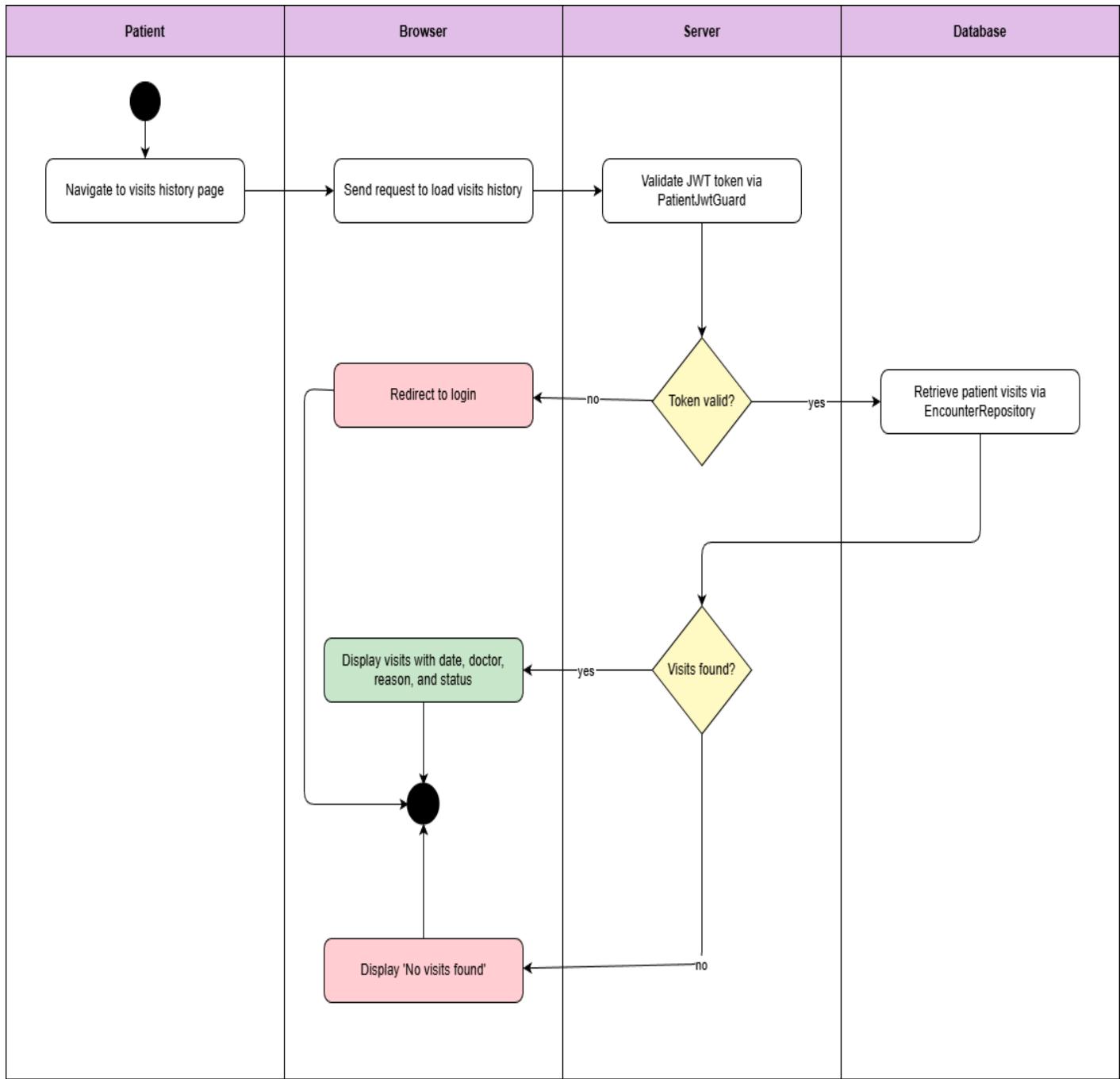


Diagram 140, Activity Diagram (View Own Visits History)

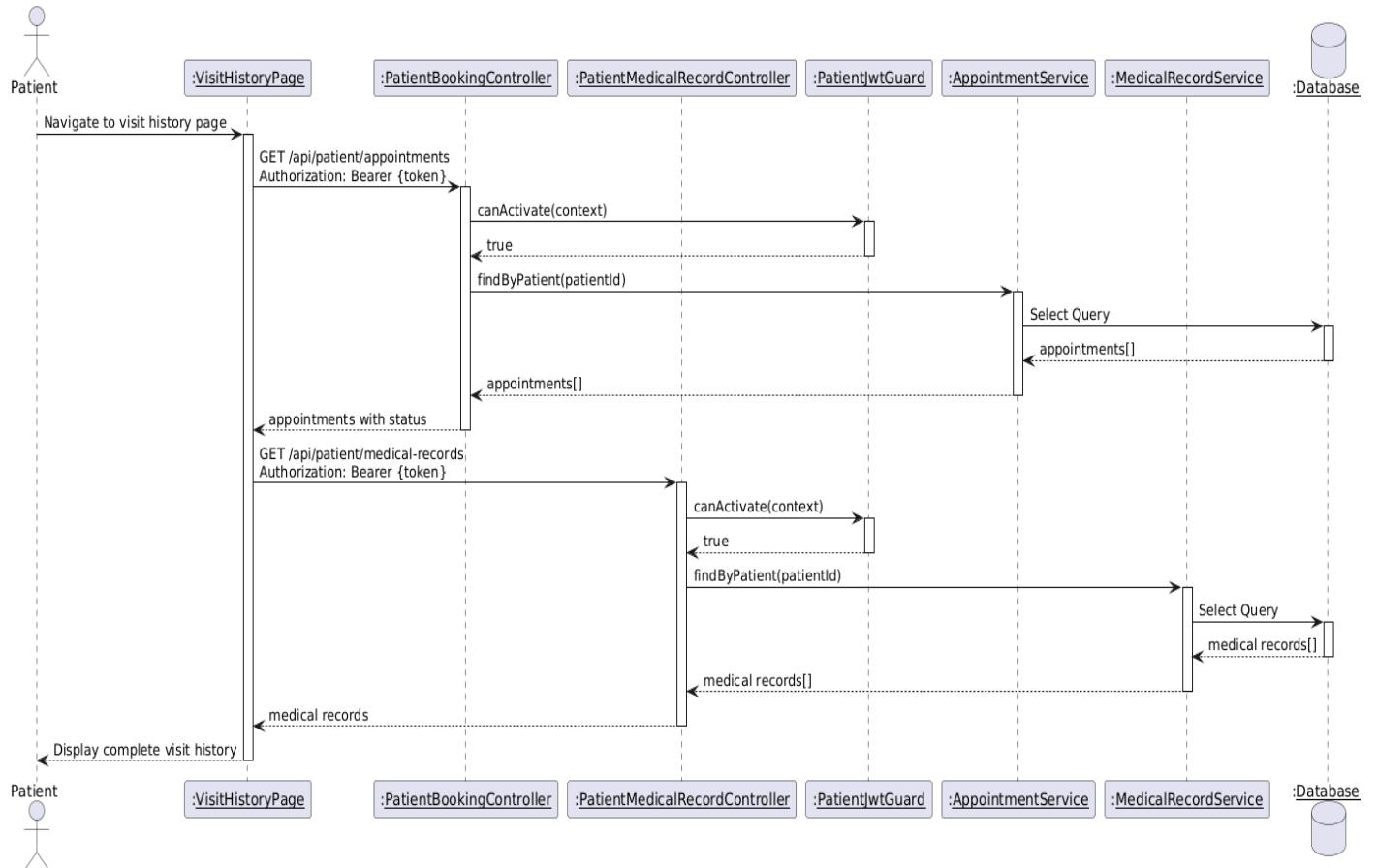


Diagram 141, Sequence Diagram (View Own Visits History)

4.7 RTM V.2

Chapter 4 - System Analysis

ID	Title	Analysis Section	Design Section	Code	Integration Test	Unit Test
VEMR-FR-PM-01	The system should allow patients to create a new account by providing email, password, first name, and last name.	Specification and Diagrams				
VEMR-FR-PM-02	The system should allow patients to login to the system using their email and password credentials.	Specification and Diagrams				
VEMR-FR-PM-03	The system should allow patients to verify their email address before accessing the system.	Specification and Diagrams				
VEMR-FR-PM-04	The system should allow users to request a password reset link via email and set a new password.	Specification and Diagrams				
VEMR-FR-PM-05	The system should allow users to view their profile information including personal details.	Specification and Diagrams				

VEMR-FR-PM-06	The system should allow users to update their profile information such as name, phone number, and other details.	Specification and Diagrams				
VEMR-FR-PM-07	The system should allow users to change their current password to a new one.	Specification and Diagrams				
VEMR-FR-AU-08	The system should allow administrators to login to the system using their credentials.	Specification and Diagrams				
VEMR-FR-AU-09	The system should allow clinicians/doctors to login to the system using their email and password.	Specification and Diagrams				
VEMR-FR-VM-10	The system should allow clinicians to create a new patient visit with date, type, reason, and chief complaint.	Specification and Diagrams				
VEMR-FR-VM-11	The system should allow clinicians to	Specification and Diagrams				

	search visits by patient name or reason for visit.					
VEMR-FR-VM-12	The system should allow clinicians to edit visit details when the visit is in progress.	Specification and Diagrams				
VEMR-FR-VM-13	The system should allow clinicians to save changes made to a visit including medical record data.	Specification and Diagrams				
VEMR-FR-VM-14	The system should allow clinicians to delete a visit from the system.	Specification and Diagrams				
VEMR-FR-VM-15	The system should allow clinicians to view complete details of a visit including medical record, vitals, and allergies.	Specification and Diagrams				
VEMR-FR-VM-16	The system allows the clinician to create and save a new medical record for a patient after validating required information.	Specification and Diagrams				

VEMR-FR-VM-17	The system allows the clinician to edit and update an existing medical record while ensuring data validation.	Specification and Diagrams				
VEMR-FR-VM-18	The system allows the clinician to finalize a medical record, update its status, and prevent further modifications.	Specification and Diagrams				
VEMR-FR-VM-19	The system should allow voice transcription to appear in real-time as the clinician speaks using Whisper.	Specification and Diagrams				
VEMR-FR-VM-20	The system should allow clinicians to view a paginated list of all patients in the system.	Specification and Diagrams				
VEMR-FR-VM-21	The system allows clinicians to search for patient.	Specification and Diagrams				
VEMR-FR-VM-22	The system should allow clinicians to view a patient's profile with	Specification and Diagrams				

	their visits and allergies.					
VEMR-FR-VM-23	The system should allow clinicians to view detailed patient information including contact and demographic data.	Specification and Diagrams				
VEMR-FR-VM-24	The system should allow clinicians to change visit status: start (planned to in-progress), complete, or cancel.	Specification and Diagrams				
VEMR-FR-VM-25	The system should allow clinicians to view a paginated list of all visits with filtering options.	Specification and Diagrams				
VEMR-FR-VS-26	The system should allow clinicians to view vital signs recorded for a specific visit.	Specification and Diagrams				
VEMR-FR-VS-27	The system should allow clinicians to record and edit vital signs (BP, HR, temp,	Specification and Diagrams				

	etc.) during an active visit.					
VEMR-FR-AM-28	The system should allow clinicians to view a patient's recorded allergies with severity and reaction details.	Specification and Diagrams				
VEMR-FR-AM-29	The system should allow clinicians to add a new allergy record for a patient with type, allergen, severity, and reaction.	Specification and Diagrams				
VEMR-FR-AM-30	The system should allow clinicians to update existing allergy information for a patient.	Specification and Diagrams				
VEMR-FR-AM-31	The system should allow clinicians to delete an allergy record from a patient's profile.	Specification and Diagrams				
VEMR-FR-CM-32	The system should allow the admin to create clinicians' accounts.	Specification and Diagrams				

VEMR-FR-CM-33	The system should allow the clinicians and admins to update their account.	Specification and Diagrams				
VEMR-FR-CM-34	The system should allow the admin to view the clinicians accounts list.	Specification and Diagrams				
VEMR-FR-CM-35	The system should allow the admin to view the clinicians account details.	Specification and Diagrams				
VEMR-FR-CM-36	The system should allow the admin to delete the clinician's accounts.	Specification and Diagrams				
VEMR-FR-CM-37	The system should allow the admin to search the clinician's accounts.	Specification and Diagrams				
VEMR-FR-OM-38	The system should allow the admin to create organizations.	Specification and Diagrams				
VEMR-FR-OM-39	The system should allow the admin to	Specification and Diagrams				

	update organization information.					
VEMR-FR-OM-40	The system should allow the admin to view all organizations.	Specification and Diagrams				
VEMR-FR-OM-41	The system should allow the admin to view detailed information about a specific organization.	Specification and Diagrams				
VEMR-FR-OM-42	The system should allow the admin to delete an organization.	Specification and Diagrams				
VEMR-FR-OM-43	The system should allow the admin to search for organizations.	Specification and Diagrams				
VEMR-FR-AN-44	The system should allow administrators to view system-wide analytics and reports.	Specification and Diagrams				
VEMR-FR-AP-45	The system should allow clinicians to view their appointment schedule.	Specification and Diagrams				

VEMR-FR-AP-46	The system should allow clinicians to view their daily appointment schedule.	Specification and Diagrams				
VEMR-FR-AP-47	The system should allow clinicians to filter appointments by specific date.	Specification and Diagrams				
VEMR-FR-AP-48	The system should allow clinicians to filter appointments by status.	Specification and Diagrams				
VEMR-FR-AP-49	The system should allow the doctor to check in a patient's appointment.	Specification and Diagrams				
VEMR-FR-AP-50	The system should allow the doctor to cancel a patient's appointment.	Specification and Diagrams				
VEMR-FR-AP-51	The system should allow the doctor to set an appointment as no show.	Specification and Diagrams				
VEMR-FR-DA-52	The system should allow clinicians to view their personal analytics	Specification and Diagrams				

	and performance metrics.					
VEMR-FR-SM-53	The system should allow clinicians to create their work schedule.	Specification and Diagrams				
VEMR-FR-SM-54	The system should allow clinicians to edit their existing work schedule.	Specification and Diagrams				
VEMR-FR-SM-55	The system should allow clinicians to delete their work schedule.	Specification and Diagrams				
VEMR-FR-SM-56	The system should allow clinicians to automatically generate available visit slots based on their schedule.	Specification and Diagrams				
VEMR-FR-SM-57	The system should allow clinicians to mark appointments as completed, no-show, or cancelled.	Specification and Diagrams				
VEMR-FR-SM-58	The system should allow clinicians to view	Specification and Diagrams				

	detailed information about a specific appointment.					
VEMR-FR-SM-59	The system should allow clinicians to define their available time slots for appointments.	Specification and Diagrams				
VEMR-FR-PP-60	The system should allow patients to view available healthcare organizations.	Specification and Diagrams				
VEMR-FR-PP-61	The system should allow patients to view doctors within specific organizations.	Specification and Diagrams				
VEMR-FR-PP-62	The system should allow patients to view available appointment slots.	Specification and Diagrams				
VEMR-FR-PP-63	The system should allow patients to book an appointment with a clinician.	Specification and Diagrams				
VEMR-FR-PP-64	The system should allow patients to view	Specification and Diagrams				

	their own appointment history and upcoming appointments.					
VEMR-FR-PP-65	The system should allow patients to filter their appointments by status.	<u>Specification and Diagrams</u>				
VEMR-FR-PP-66	The system should allow patients to view their own medical records.	<u>Specification and Diagrams</u>				
VEMR-FR-PP-67	The system should allow patients to view their own allergy information.	<u>Specification and Diagrams</u>				
VEMR-FR-PP-68	The system should allow patients to view their complete visit history.	<u>Specification and Diagrams</u>				

Table 76 - RTM V.2

Chapter 5 - System Design

5.1 Introduction:

The system employs a layered architecture combined with a client-server architecture pattern. This approach provides clear separation of concerns, enhances maintainability, and enables scalable development. The architecture is designed to ensure efficient data flow, interaction, and integration between components.

5.2 Architecture Layers

5.2.1 Client Side

This is the place where users interact with the system through a web-based user interface.

5.2.2 Server-Side

Presentation Layer: The API Gateway represents the entry point for User requests. This layer handles communication between the User and the server.

Business Logic Layer: Contains the core functionality of the system, including Account management module, Patient visit management module, Speech to Text Module, Medical record Module, Administration Module, Appointment Management Module.

Data Access Layer: Provides an abstraction to interact with the database using the Repository pattern, responsible for data access operations.

5.2.3 Database

The database stores persistent data and is accessed via repositories in the Data Access Layer. The system uses PostgreSQL as his database.

5.3 Sprint 1 Design

5.3.1 System Architecture & Component Diagram

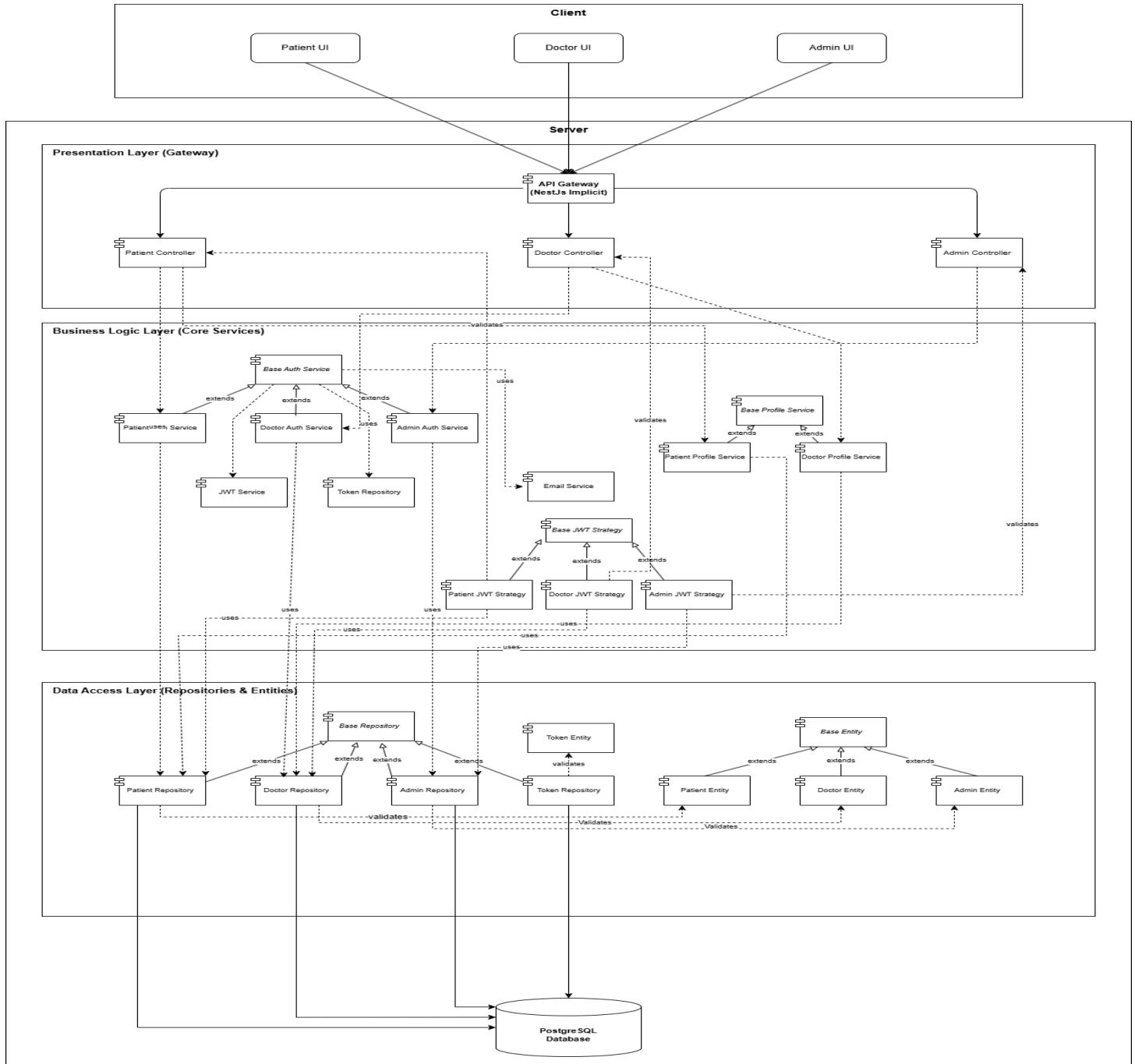


Figure 142, Sprint 1 Component Diagram

Chapter 5 - System Design

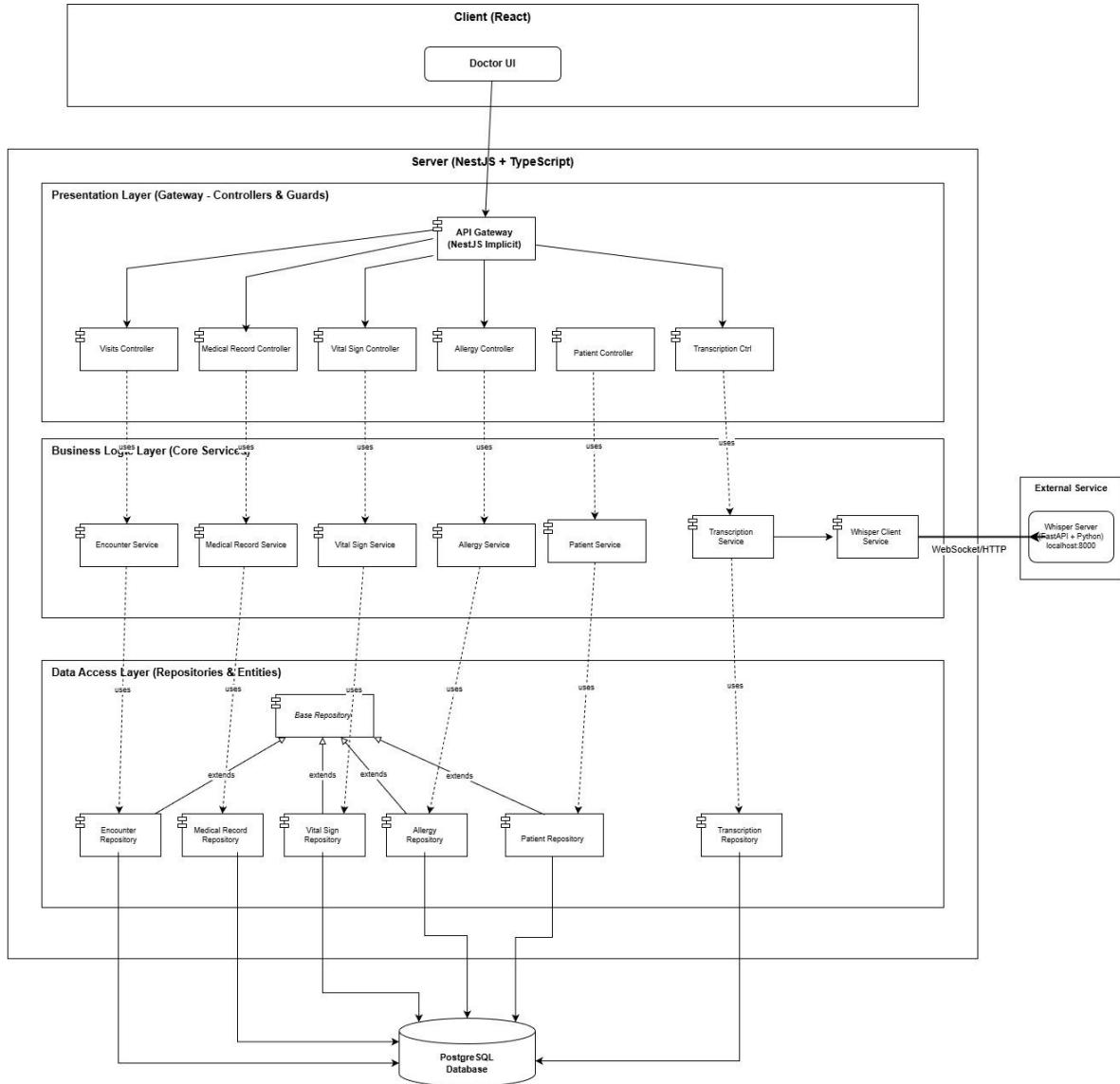


Figure 143, Sprint 2 + 3 Component Diagram

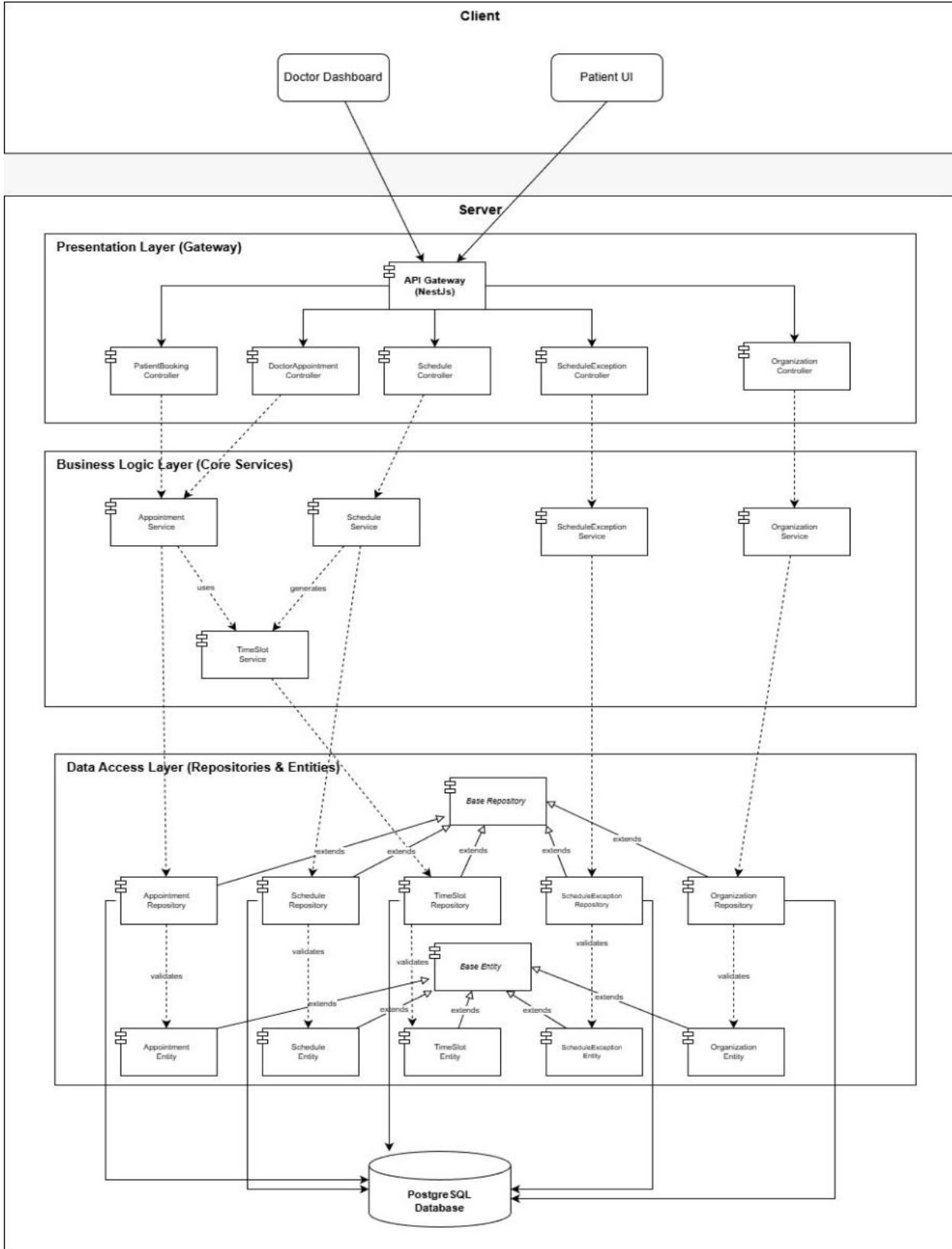


Figure 144, Sprint 4 Component diagram

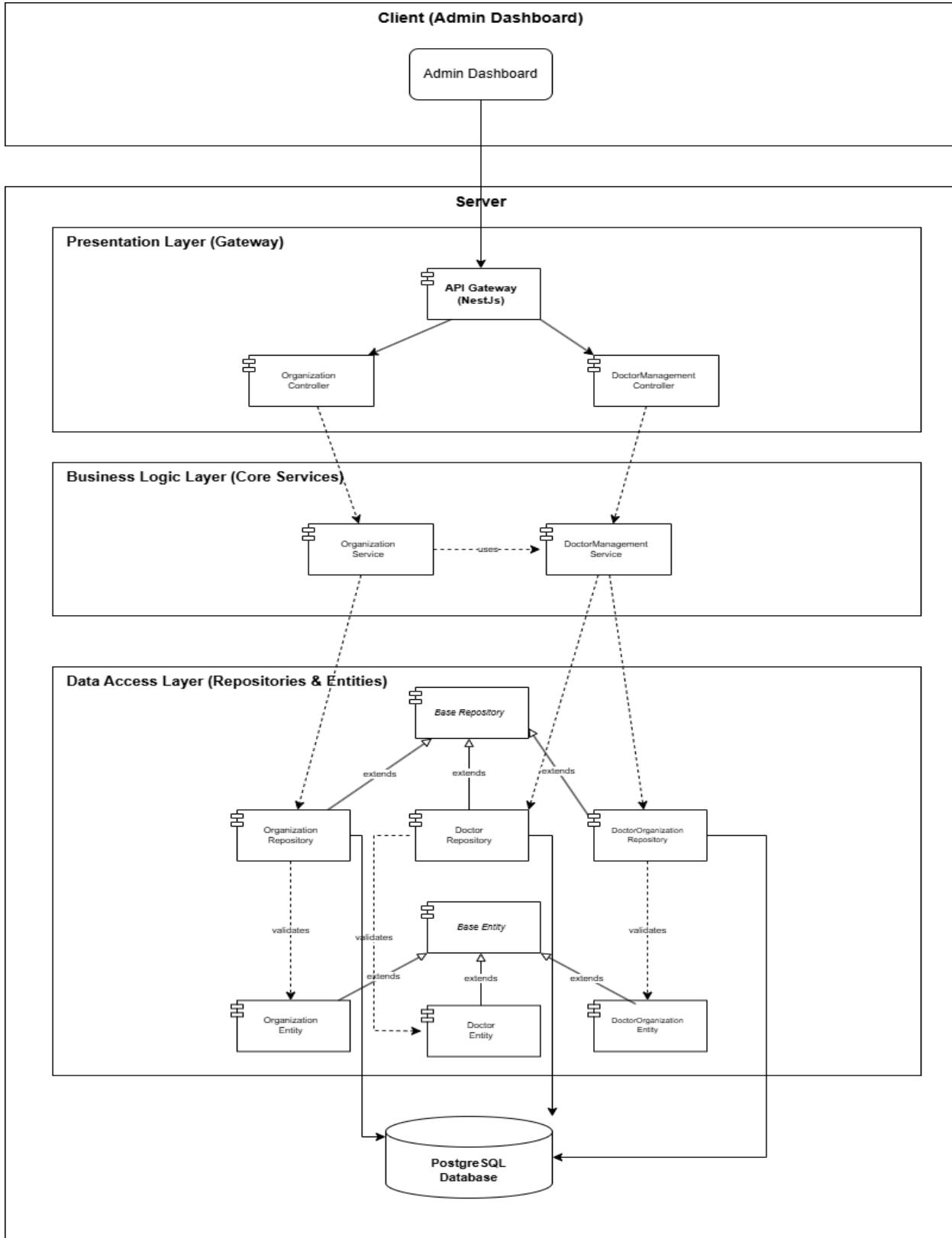


Figure 145, Sprint 5 Component Diagram

5.3.2 Class Diagram

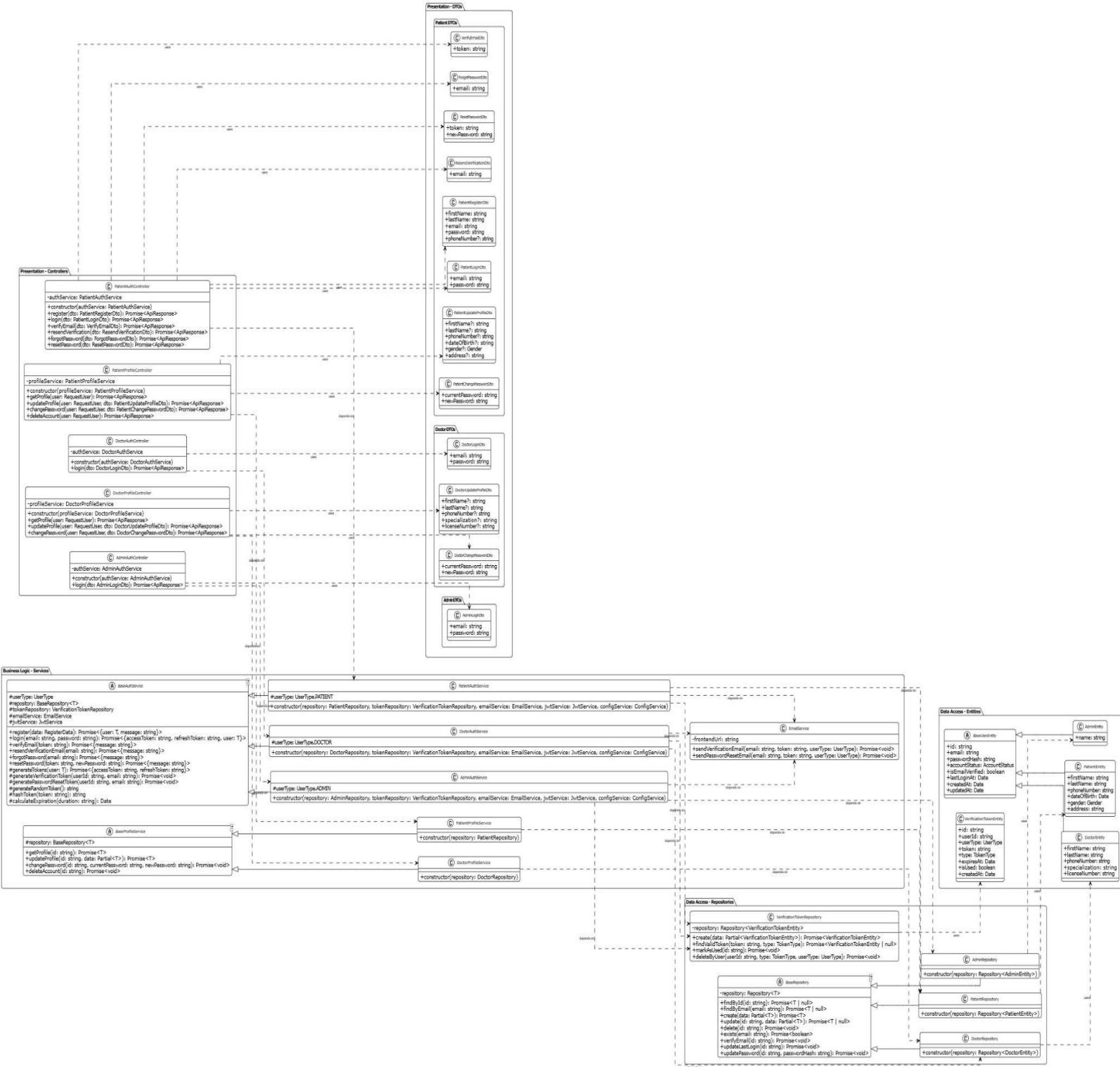


Figure 146, Sprint 1 Class Diagram

Chapter 5 - System Design

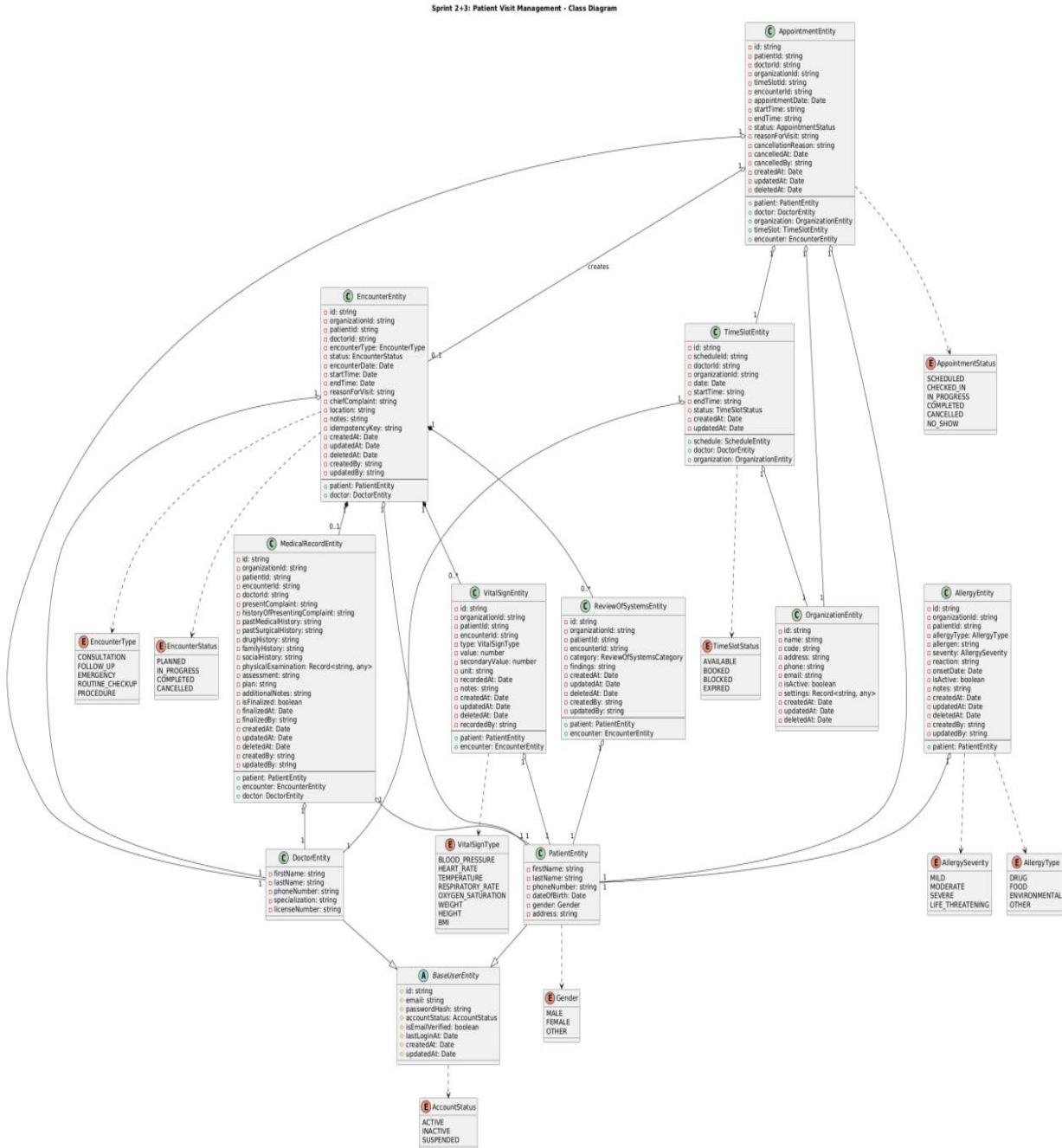


Figure 147, Sprint 2+3 Component Diagram

Chapter 5 - System Design

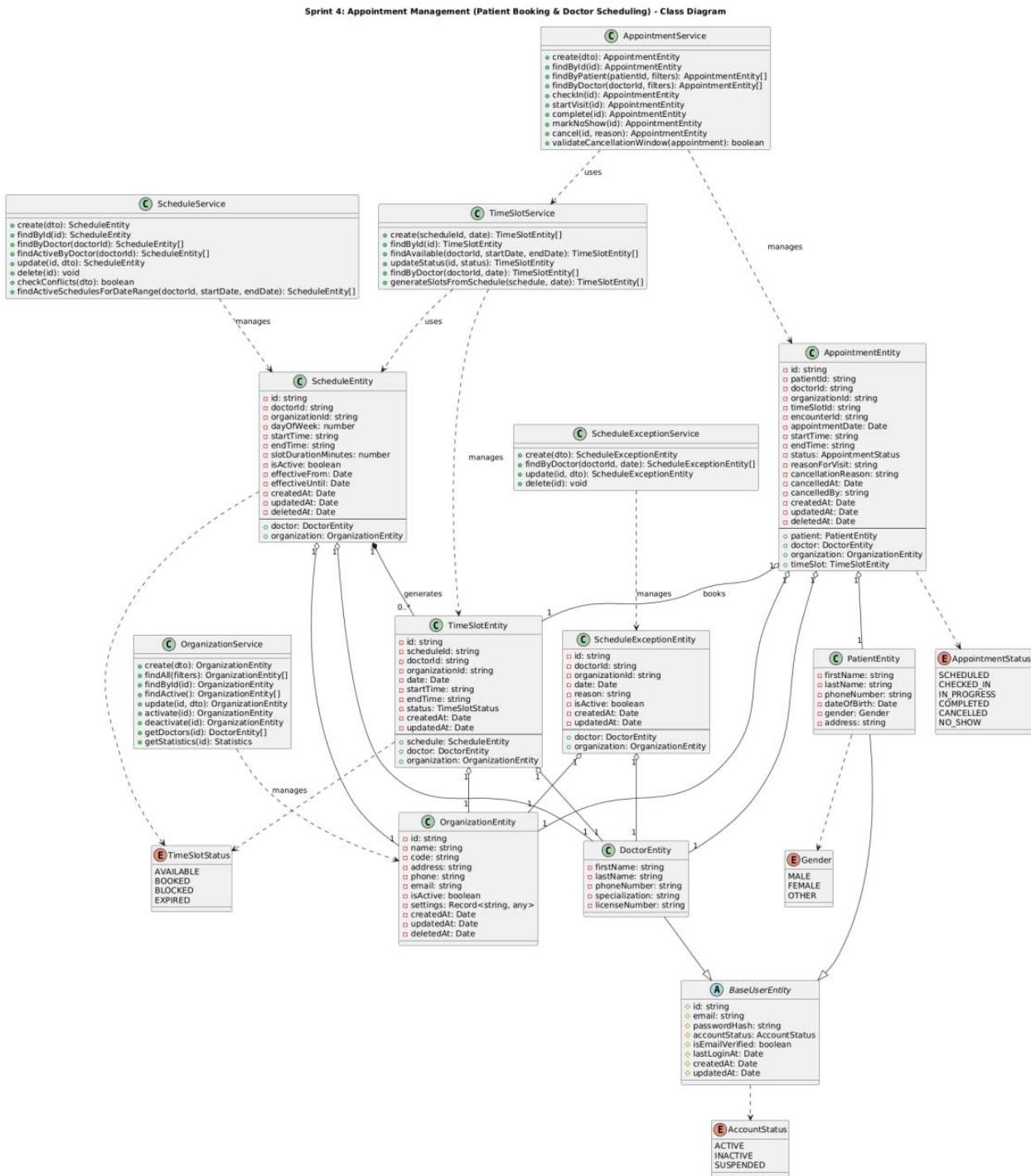


Figure 148, Sprint 4 Component Diagram

Chapter 5 - System Design

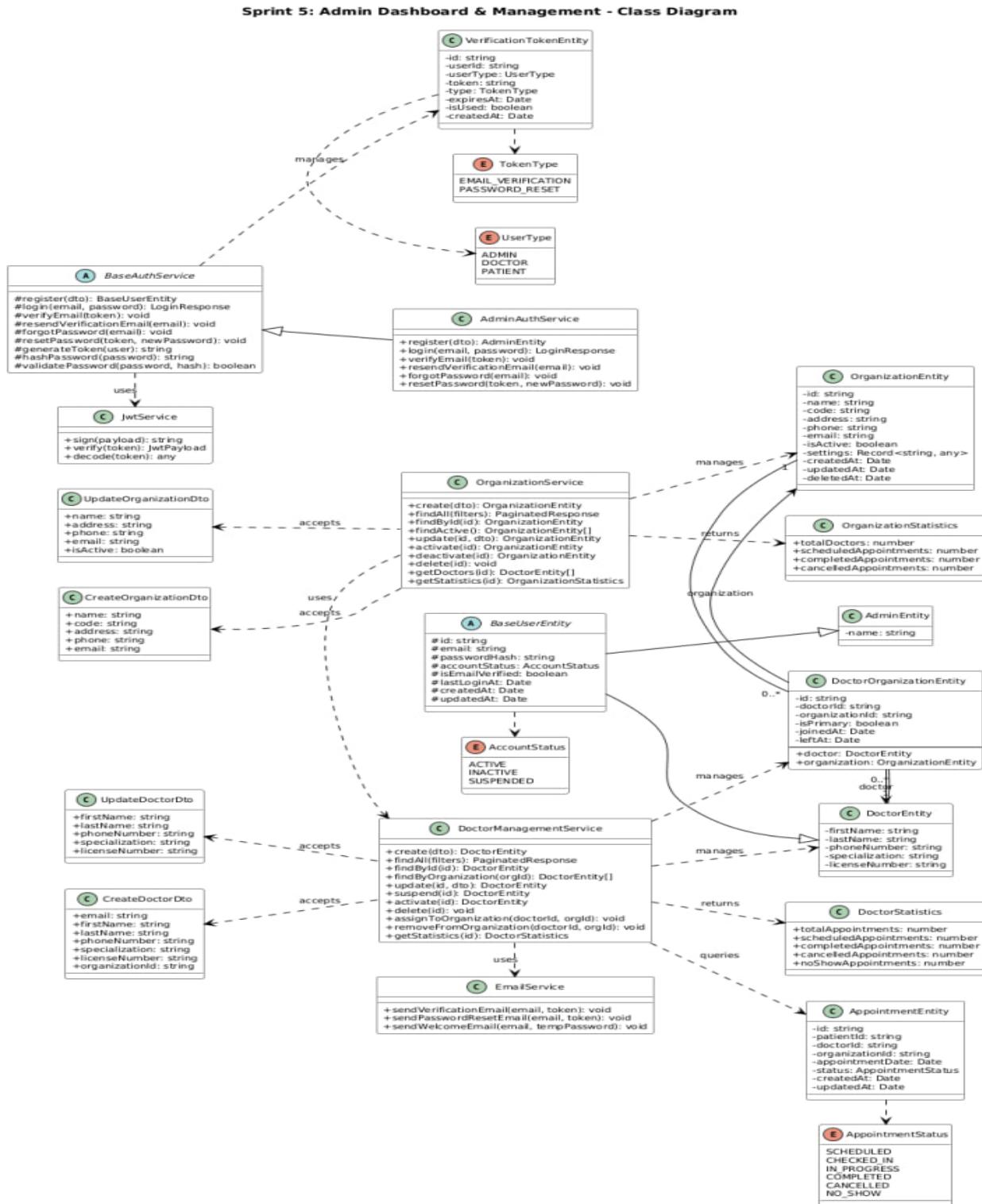


Figure 149, Sprint 5 Class Diagram

Entity-Relationship Diagram

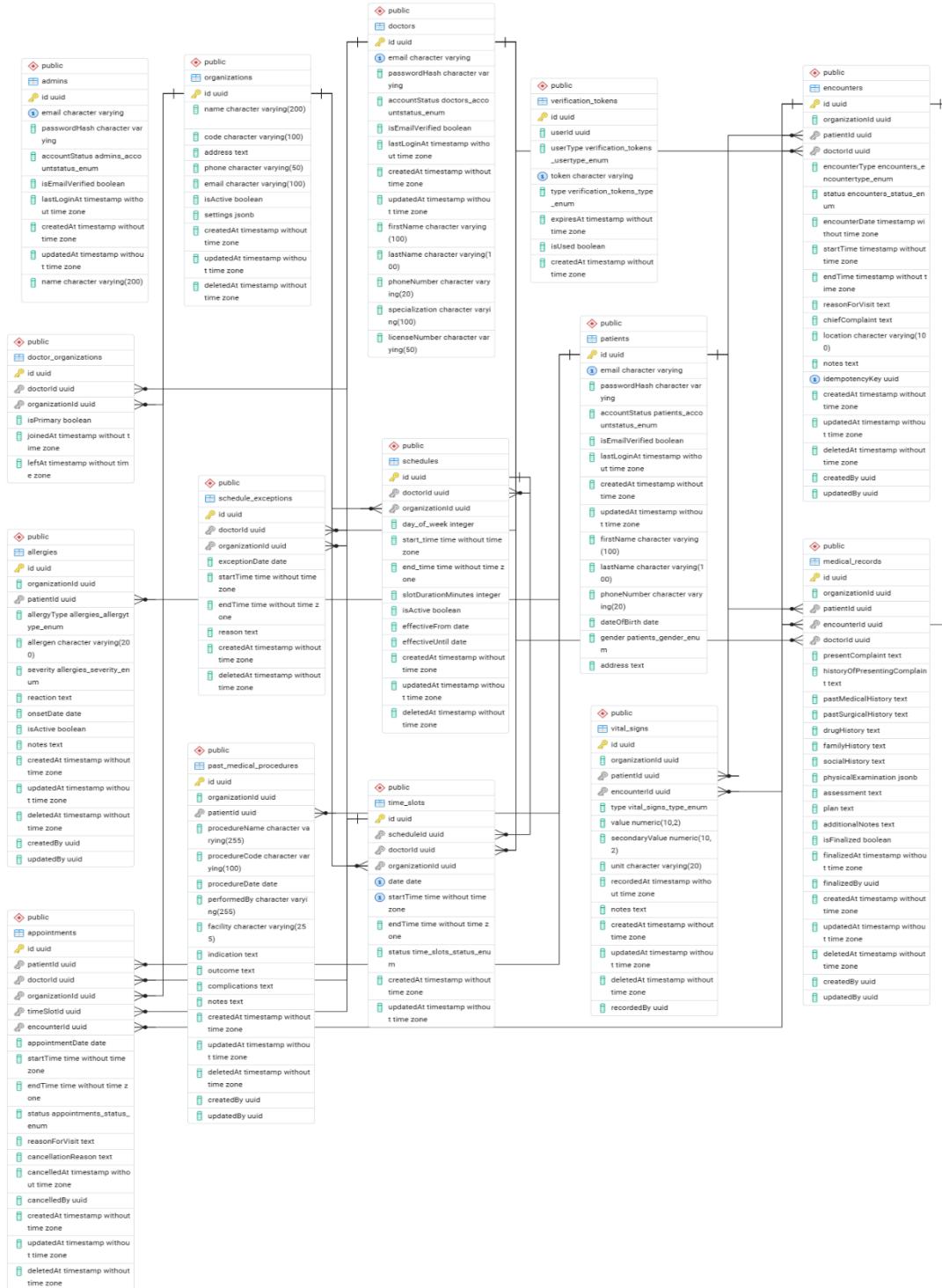


Figure 150, ERD
5.5 RTM V.3

ID	Title	Analysis Section	Design Section	Code	Integration Test	Unit Test
VEMR-FR-PM-01	The system should allow patients to create a new account by providing email, password, first name, and last name.	Specification and Diagrams	Component Diagram			
VEMR-FR-PM-02	The system should allow patients to login to the system using their email and password credentials.	Specification and Diagrams	Component Diagram			
VEMR-FR-PM-03	The system should allow patients to verify their email address before accessing the system.	Specification and Diagrams	Component Diagram			
VEMR-FR-PM-04	The system should allow users to request a password reset link via email and set a new password.	Specification and Diagrams	Component Diagram			
VEMR-FR-PM-05	The system should allow users to view their profile information including personal details.	Specification and Diagrams	Component Diagram			
VEMR-FR-PM-06	The system should allow users to update their profile information such as	Specification and Diagrams	Component Diagram			

	name, phone number, and other details.					
VEMR-FR-PM-07	The system should allow users to change their current password to a new one.	Specification and Diagrams	Component Diagram			
VEMR-FR-AU-08	The system should allow administrators to login to the system using their credentials.	Specification and Diagrams	Component Diagram			
VEMR-FR-AU-09	The system should allow clinicians/doctors to login to the system using their email and password.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-10	The system should allow clinicians to create a new patient visit with date, type, reason, and chief complaint.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-11	The system should allow clinicians to search visits by patient name or reason for visit.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-12	The system should allow clinicians to edit visit	Specification and Diagrams	Component Diagram			

	details when the visit is in progress.					
VEMR-FR-VM-13	The system should allow clinicians to save changes made to a visit including medical record data.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-14	The system should allow clinicians to delete a visit from the system.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-15	The system should allow clinicians to view complete details of a visit including medical record, vitals, and allergies.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-16	The system allows the clinician to create and save a new medical record for a patient after validating required information.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-17	The system allows the clinician to edit and update an existing medical record while ensuring data validation.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-18	The system allows the clinician to finalize a	Specification and Diagrams	Component Diagram			

	medical record, update its status, and prevent further modifications.					
VEMR-FR-VM-19	The system should allow voice transcription to appear in real-time as the clinician speaks using Whisper.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-20	The system should allow clinicians to view a paginated list of all patients in the system.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-21	The system allows clinicians to search for patient.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-22	The system should allow clinicians to view a patient's profile with their visits and allergies.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-23	The system should allow clinicians to view detailed patient information including contact and demographic data.	Specification and Diagrams	Component Diagram			
VEMR-FR-VM-24	The system should allow clinicians to change visit	Specification and Diagrams	Component Diagram			

Chapter 5 - System Design

	status: start (planned to in-progress), complete, or cancel.					
VEMR-FR-VM-25	The system should allow clinicians to view a paginated list of all visits with filtering options.	Specification and Diagrams	Component Diagram			
VEMR-FR-VS-26	The system should allow clinicians to view vital signs recorded for a specific visit.	Specification and Diagrams	Component Diagram			
VEMR-FR-VS-27	The system should allow clinicians to record and edit vital signs (BP, HR, temp, etc.) during an active visit.	Specification and Diagrams	Component Diagram			
VEMR-FR-AM-28	The system should allow clinicians to view a patient's recorded allergies with severity and reaction details.	Specification and Diagrams	Component Diagram			
VEMR-FR-AM-29	The system should allow clinicians to add a new allergy record for a patient with type, allergen, severity, and reaction.	Specification and Diagrams	Component Diagram			

VEMR-FR-AM-30	The system should allow clinicians to update existing allergy information for a patient.	Specification and Diagrams	Component Diagram			
VEMR-FR-AM-31	The system should allow clinicians to delete an allergy record from a patient's profile.	Specification and Diagrams	Component Diagram			
VEMR-FR-CM-32	The system should allow the admin to create clinicians' accounts.	Specification and Diagrams	Component Diagram			
VEMR-FR-CM-33	The system should allow the clinicians and admins to update their account.	Specification and Diagrams	Component Diagram			
VEMR-FR-CM-34	The system should allow the admin to view the clinicians accounts list.	Specification and Diagrams	Component Diagram			
VEMR-FR-CM-35	The system should allow the admin to view the clinicians account details.	Specification and Diagrams	Component Diagram			
VEMR-FR-CM-36	The system should allow the admin to delete the clinician's accounts.	Specification and Diagrams	Component Diagram			

VEMR-FR-CM-37	The system should allow the admin to search the clinician's accounts.	Specification and Diagrams	Component Diagram			
VEMR-FR-OM-38	The system should allow the admin to create organizations.	Specification and Diagrams	Component Diagram			
VEMR-FR-OM-39	The system should allow the admin to update organization information.	Specification and Diagrams	Component Diagram			
VEMR-FR-OM-40	The system should allow the admin to view all organizations.	Specification and Diagrams	Component Diagram			
VEMR-FR-OM-41	The system should allow the admin to view detailed information about a specific organization.	Specification and Diagrams	Component Diagram			
VEMR-FR-OM-42	The system should allow the admin to delete an organization.	Specification and Diagrams	Component Diagram			
VEMR-FR-OM-43	The system should allow the admin to search for organizations.	Specification and Diagrams	Component Diagram			

VEMR-FR-AN-44	The system should allow administrators to view system-wide analytics and reports.	Specification and Diagrams	Component Diagram			
VEMR-FR-AP-45	The system should allow clinicians to view their appointment schedule.	Specification and Diagrams	Component Diagram			
VEMR-FR-AP-46	The system should allow clinicians to view their daily appointment schedule.	Specification and Diagrams	Component Diagram			
VEMR-FR-AP-47	The system should allow clinicians to filter appointments by specific date.	Specification and Diagrams	Component Diagram			
VEMR-FR-AP-48	The system should allow clinicians to filter appointments by status.	Specification and Diagrams	Component Diagram			
VEMR-FR-AP-49	The system should allow the doctor to check in a patient's appointment.	Specification and Diagrams	Component Diagram			
VEMR-FR-AP-50	The system should allow the doctor to cancel a patient's appointment.	Specification and Diagrams	Component Diagram			

VEMR-FR-AP-51	The system should allow the doctor to set an appointment as no show.	Specification and Diagrams	Component Diagram			
VEMR-FR-DA-52	The system should allow clinicians to view their personal analytics and performance metrics.	Specification and Diagrams	Component Diagram			
VEMR-FR-SM-53	The system should allow clinicians to create their work schedule.	Specification and Diagrams	Component Diagram			
VEMR-FR-SM-54	The system should allow clinicians to edit their existing work schedule.	Specification and Diagrams	Component Diagram			
VEMR-FR-SM-55	The system should allow clinicians to delete their work schedule.	Specification and Diagrams	Component Diagram			
VEMR-FR-SM-56	The system should allow clinicians to automatically generate available visit slots based on their schedule.	Specification and Diagrams	Component Diagram			
VEMR-FR-SM-57	The system should allow clinicians to mark appointments as	Specification and Diagrams	Component Diagram			

	completed, no-show, or cancelled.				
VEMR-FR-SM-58	The system should allow clinicians to view detailed information about a specific appointment.	Specification and Diagrams	Component Diagram		
VEMR-FR-SM-59	The system should allow clinicians to define their available time slots for appointments.	Specification and Diagrams	Component Diagram		
VEMR-FR-PP-60	The system should allow patients to view available healthcare organizations.	Specification and Diagrams	Component Diagram		
VEMR-FR-PP-61	The system should allow patients to view doctors within specific organizations.	Specification and Diagrams	Component Diagram		
VEMR-FR-PP-62	The system should allow patients to view available appointment slots.	Specification and Diagrams	Component Diagram		
VEMR-FR-PP-63	The system should allow patients to book an appointment with a clinician.	Specification and Diagrams	Component Diagram		

VEMR-FR-PP-64	The system should allow patients to view their own appointment history and upcoming appointments.	Specification and Diagrams	Component Diagram			
VEMR-FR-PP-65	The system should allow patients to filter their appointments by status.	Specification and Diagrams	Component Diagram			
VEMR-FR-PP-66	The system should allow patients to view their own medical records.	Specification and Diagrams	Component Diagram			
VEMR-FR-PP-67	The system should allow patients to view their own allergy information.	Specification and Diagrams	Component Diagram			
VEMR-FR-PP-68	The system should allow patients to view their complete visit history.	Specification and Diagrams	Component Diagram			

Table 77- RTM V.3

Chapter 6 - Implementation and Testing

6.1 Introduction

This chapter details the implementation and testing phase of the Voice-Based Electronic Medical Records (EMR) System, focusing on how each system component and interface was developed, integrated, and validated. The goal is to document how the requirements defined in earlier chapters were realized into a working application and how each module was verified through testing to ensure correctness, usability, performance, security, and compliance with healthcare standards. The implementation follows industry best practices for medical software development.

6.2 Technologies Used

1. Node.js

Node.js is a powerful JavaScript runtime built on Chrome's V8 JavaScript engine that allows developers to execute JavaScript code on the server-side. It enables building scalable and high-performance applications using an event-driven, non-blocking I/O model. Node.js is particularly well-suited for building microservices and API-driven applications, making it ideal for our EMR system's backend architecture.

It's Role in Our System:

- Powers the backend API server handling all clinical, administrative, and scheduling operations
- Manages concurrent requests from multiple users (doctors, patients, administrators) efficiently
- Provides real-time capabilities for appointments and system updates
- Enables seamless integration with external services such as email notifications and transcription services
- Supports asynchronous processing of medical records and clinical documentation

2. Nest.Js

NestJS is a progressive Node.js framework for building efficient, reliable, and scalable server-side applications. It uses TypeScript by default and combines elements of Object-Oriented Programming (OOP), Functional Programming (FP), and Functional Reactive Programming (FRP). NestJS provides an out-of-the-box application architecture that allows developers to create highly testable, scalable, and maintainable applications.

It's Role in Our System:

- Serves as the primary backend framework organizing the entire application architecture
- Implements dependency injection for better code organization and testability
- Provides modular structure separating concerns (authentication, clinical services, scheduling)
- Offers built-in support for validation, guards, and interceptors for security
- Facilitates the creation of RESTful APIs and WebSocket gateways for real-time features

3. TypeScript

TypeScript is a strongly typed programming language that builds on JavaScript by adding static type definitions. It enhances code quality, readability, and maintainability by catching errors during development rather than at runtime.

TypeScript's interfaces, classes, and generics have been instrumental in creating a robust architecture for our EMR system, ensuring type safety across the entire codebase.

It's Role in Our System:

- Defines strict types for all medical entities (patients, doctors, appointments, medical records)
- Ensures data integrity through compile-time type checking
- Provides IntelliSense and autocomplete for improved developer productivity
- Creates clear contracts between different system modules through interfaces
- Reduces runtime errors in critical healthcare operations

4. React

React is a popular JavaScript library developed by Meta for building user interfaces, primarily for web applications. It is based on a component-driven architecture, allowing developers to create reusable UI elements and manage complex application states efficiently. React employs a virtual DOM for optimized rendering, ensuring high performance and responsiveness. It's particularly suited for building dynamic, single-page applications (SPAs) and offers flexibility to integrate with other libraries or frameworks for managing functionality like routing or state management.

It's Role In Our System:

- Powers the frontend user interfaces for patients, doctors, and administrators
- Manages complex UI states for appointment scheduling and medical record viewing
- Provides responsive and intuitive interfaces for clinical documentation
- Enables real-time updates for appointment status changes
- Implements role-based UI components for different user types

5. PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation for reliability, feature robustness, and performance. It supports advanced data types, complex queries, and ACID (Atomicity, Consistency, Isolation, Durability) compliance, making it ideal for applications requiring data integrity and consistency.

It's Role In Our System:

- Stores all critical medical data including patient records, appointments, and clinical documentation
- Maintains referential integrity between related entities (patients, doctors, encounters, allergies)
- Supports complex queries for medical record retrieval and reporting
- Provides transaction support for ensuring data consistency in multi-step operations
- Enables efficient indexing for fast retrieval of patient information

6. Jest

Jest is a delightful JavaScript testing framework with a focus on simplicity and support for TypeScript. It provides a complete testing solution with built-in test runners, assertion libraries, and mocking capabilities. Our project leverages Jest for comprehensive unit and integration testing of all system components.

It's Role In Our System:

- Tests all authentication flows (registration, login, password reset)
- Validates clinical service operations (allergies, encounters, medical records)
- Verifies scheduling logic (appointments, time slots, conflicts)
- Tests administrative functions (doctor management, organization setup)
- Ensures business rules are correctly implemented (cancellation policies, access control)

6.3 Testing Report

6.3.1 Introduction

This report provides a comprehensive analysis of the test coverage and quality assurance measures implemented for the Electronic Medical Records (EMR) System backend. The testing suite demonstrates exceptional coverage across all critical system components, with 324 tests across 21 test suites for unit testing.

And all API Endpoints tested for integration testing using postman

6.3.2 Unit Testing Metrics

```
===== Coverage summary =====
Statements : 95.96% ( 381/397 )
Branches   : 93.04% ( 107/115 )
Functions   : 92.68% ( 76/82 )
Lines       : 95.68% ( 355/371 )
=====
Test Suites: 21 passed, 21 total
Tests:      324 passed, 324 total
Snapshots:  0 total
Time:       14.001 s
Ran all test suites.
```

Figure 141, Testing metrics picture

6.3.3 Integration Testing

We did integration testing using postman and we put all the test results on the Jira Tickets

Chapter 6 - Implementation and Testing

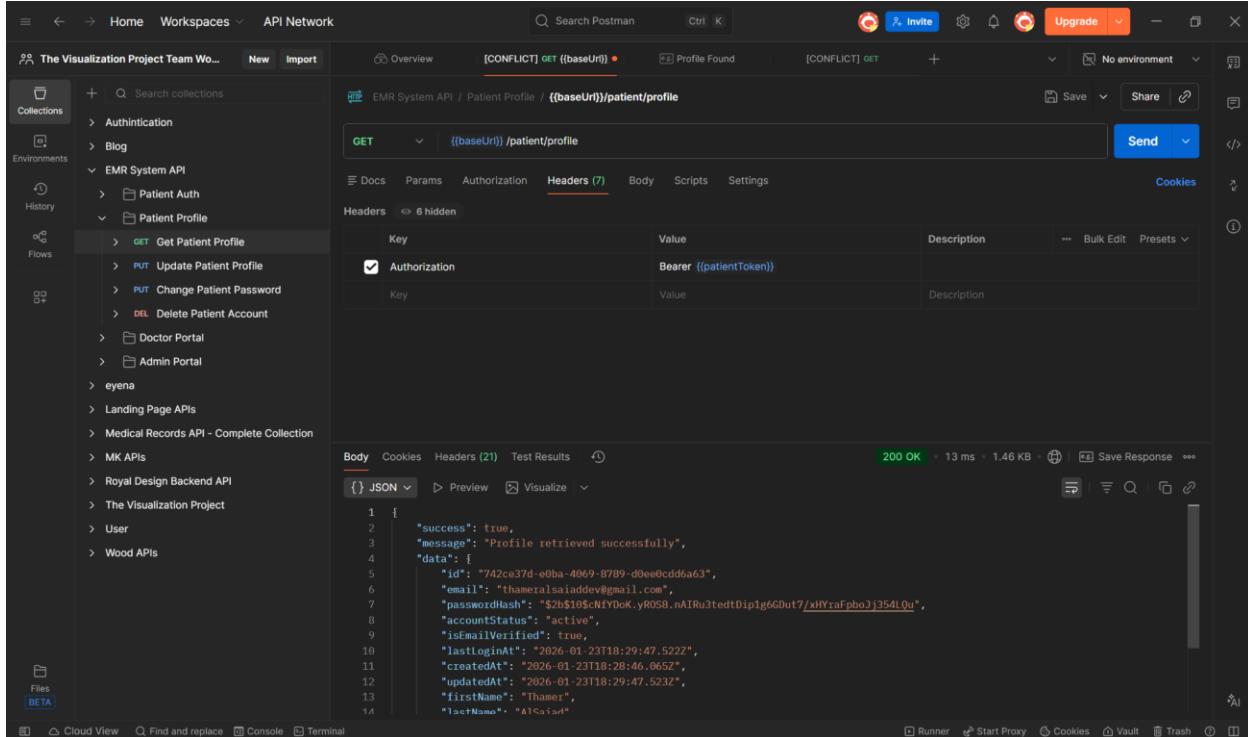


Figure 151, Postman Integration Testing

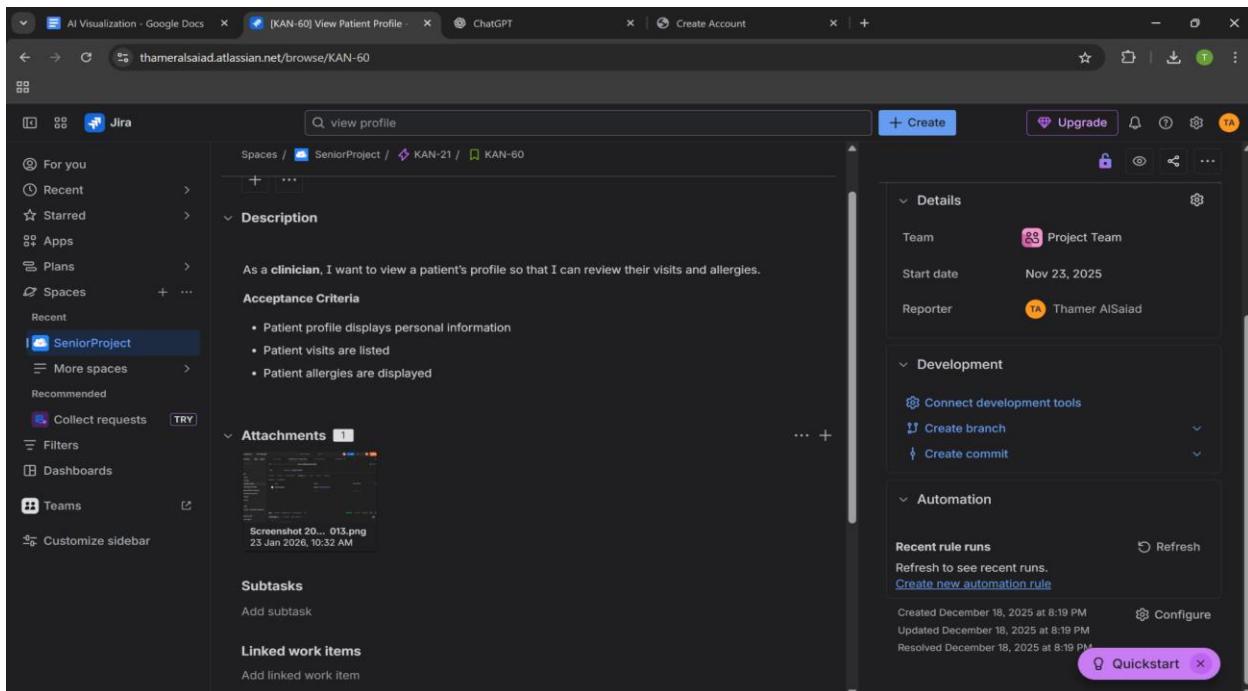


Figure 152, Jira Ticket with postman test attached

6.3 System Interfaces

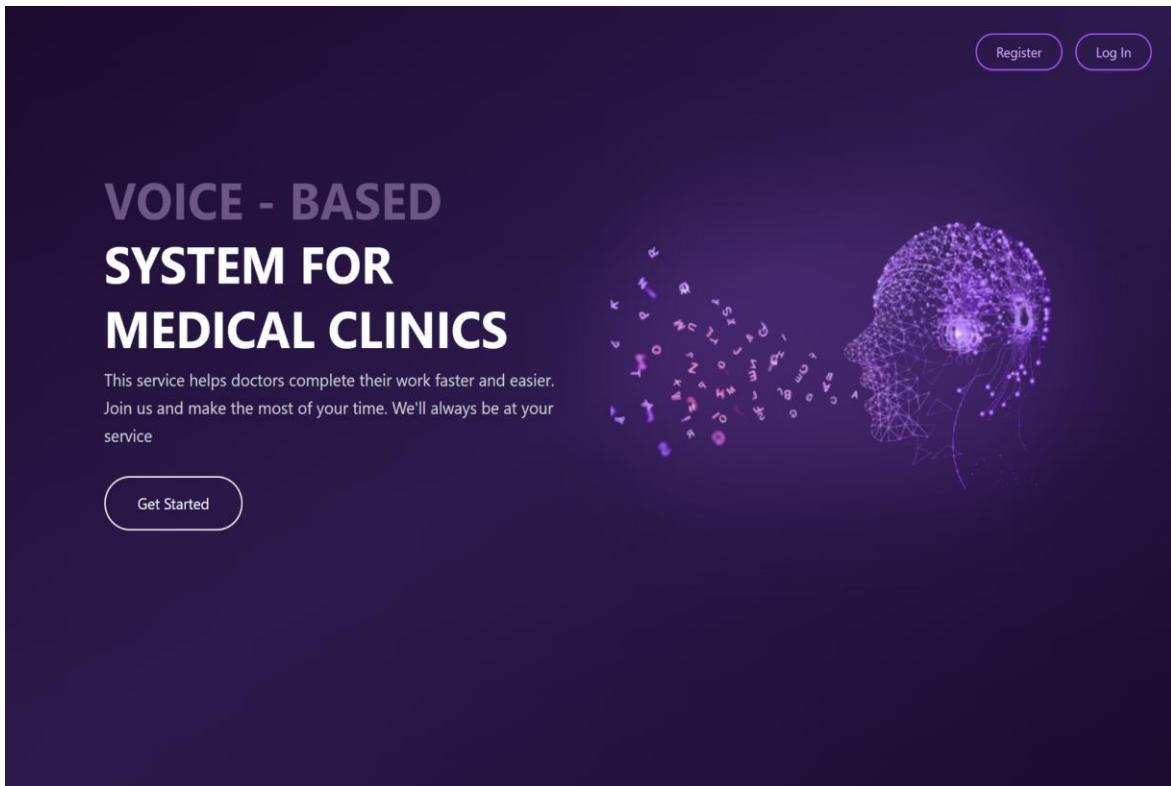


Figure 153, Main user interface

The image shows a 'Create An Account' form. It includes fields for First name and Last name, both with placeholder text 'John Doe'. There is also a field for Email Address and two fields for Password and Confirm your password. Below the password fields is a note: 'Use 8 or more characters with letters, numbers & symbols' and a checkbox for 'Show password'. At the bottom is a 'Create an account' button. To the right of the form is a circular icon containing three medical professionals (a male doctor, a female doctor, and a nurse) standing next to various medical icons like a heart rate monitor, test tubes, and a stethoscope.

Figure 154, User registration page

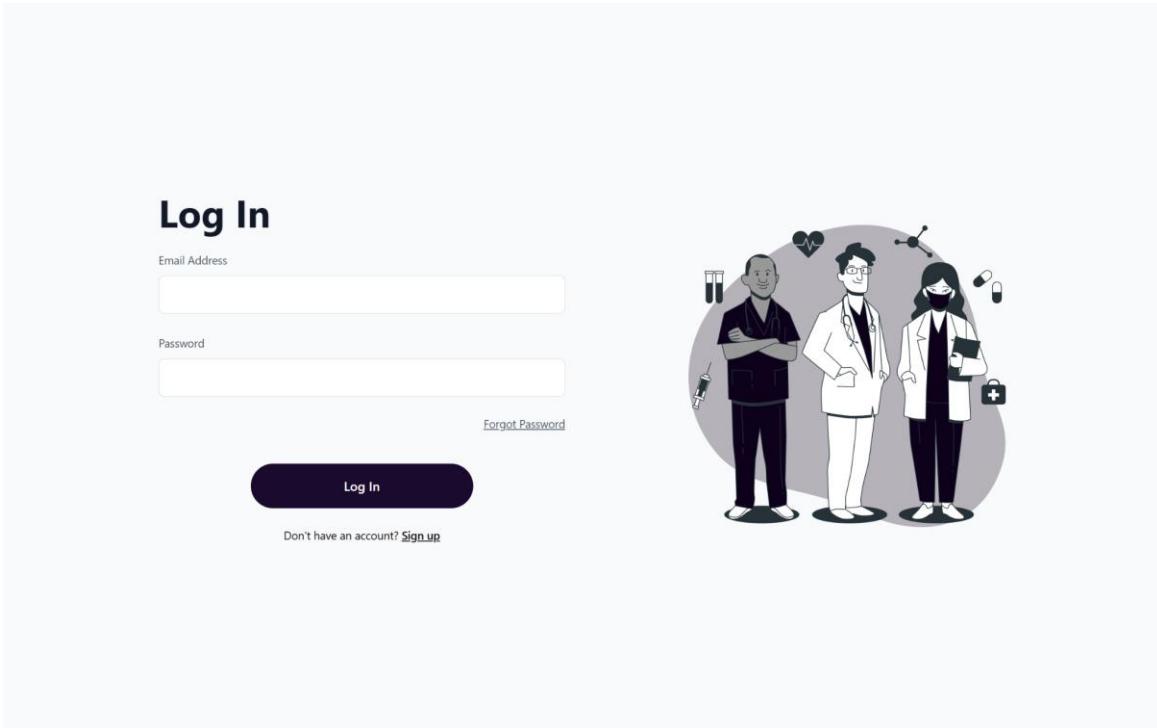


Figure 155, Login page

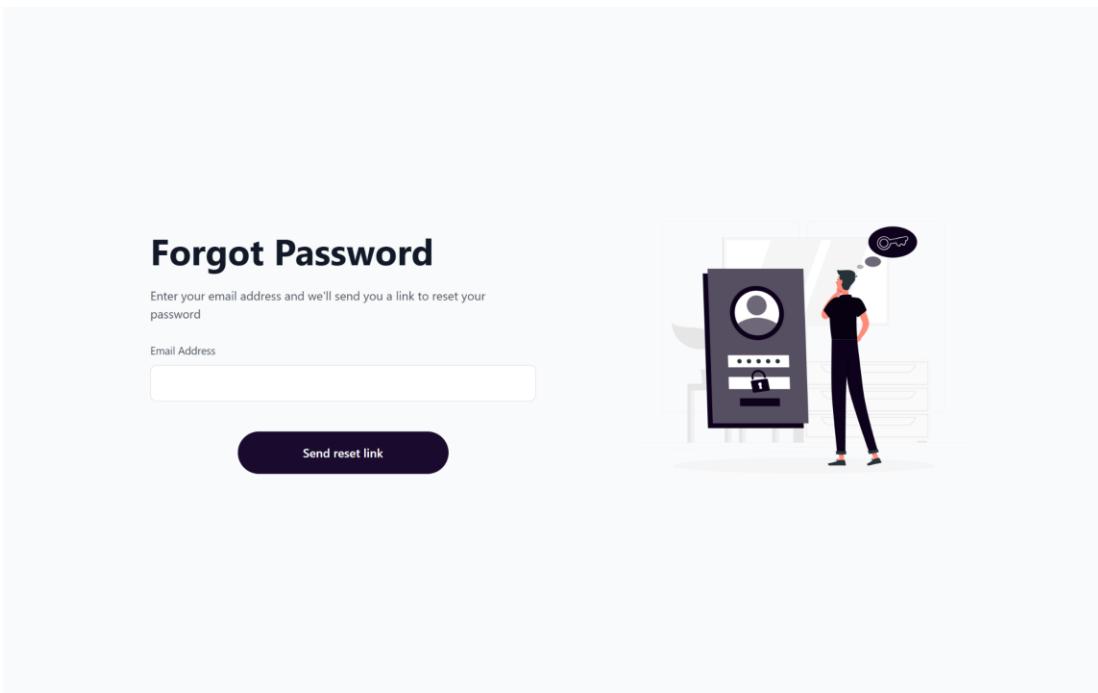


Figure 156, Forgot password page

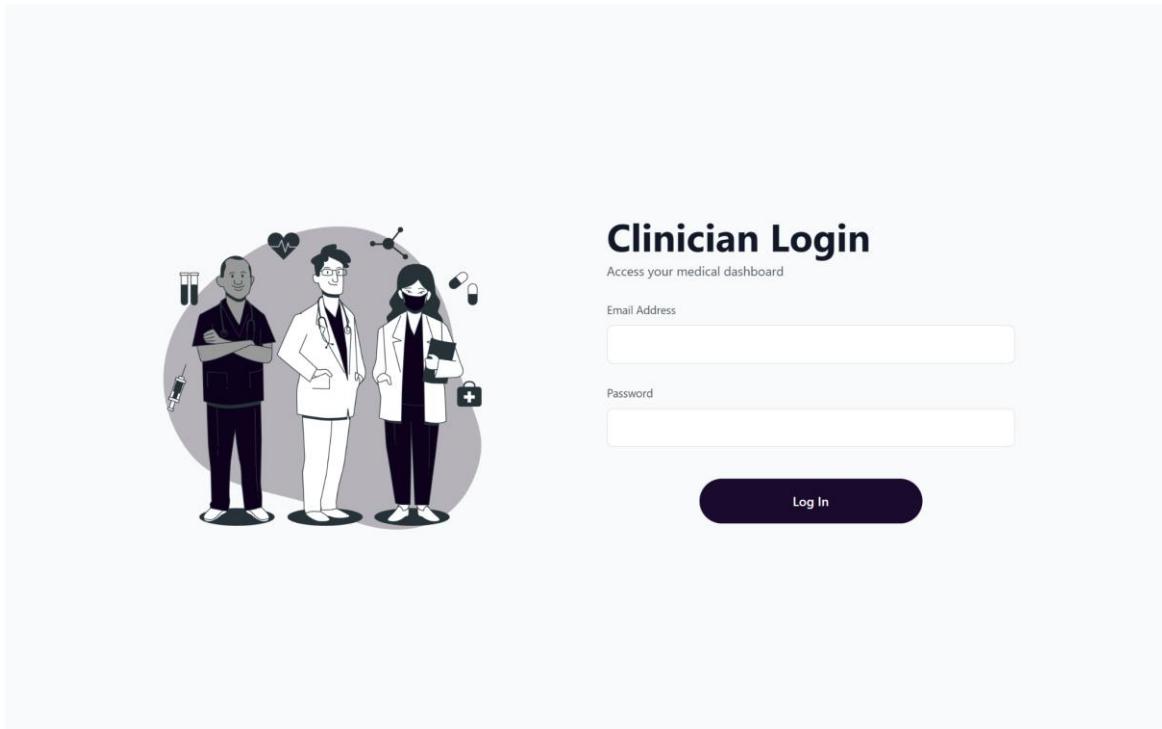


Figure 157, Clinician Login page

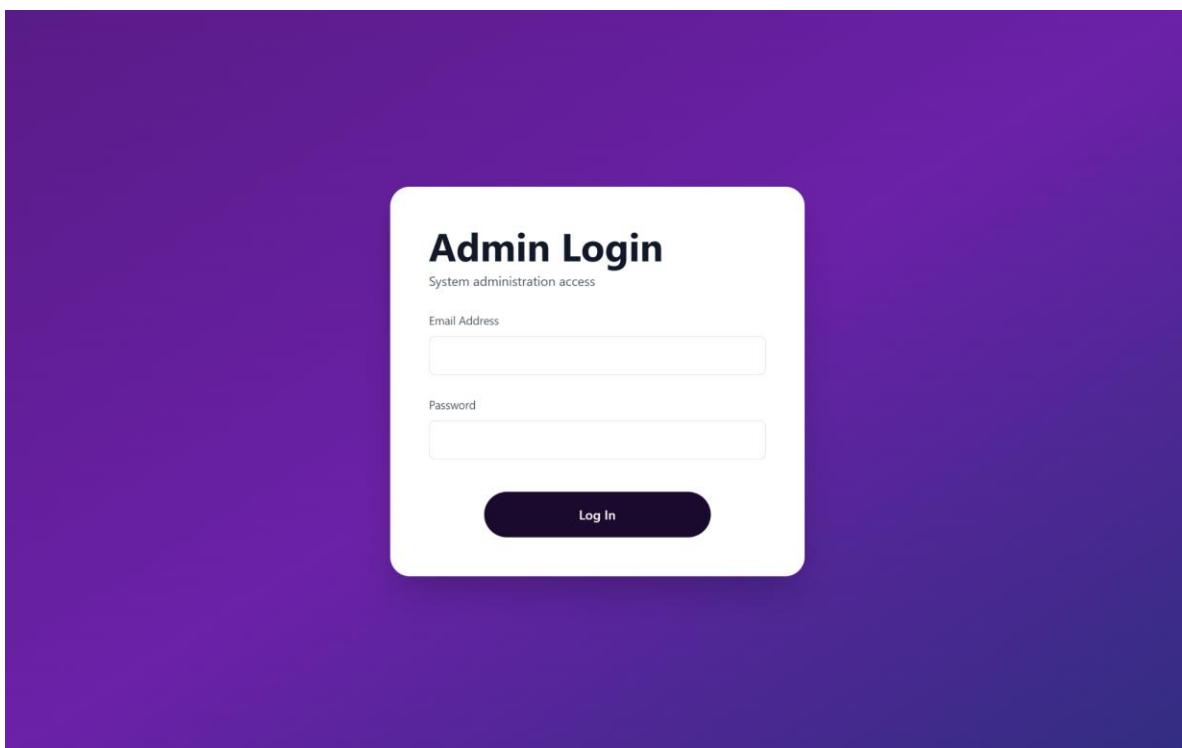


Figure 158, Admin Login page

Chapter 6 - Implementation and Testing

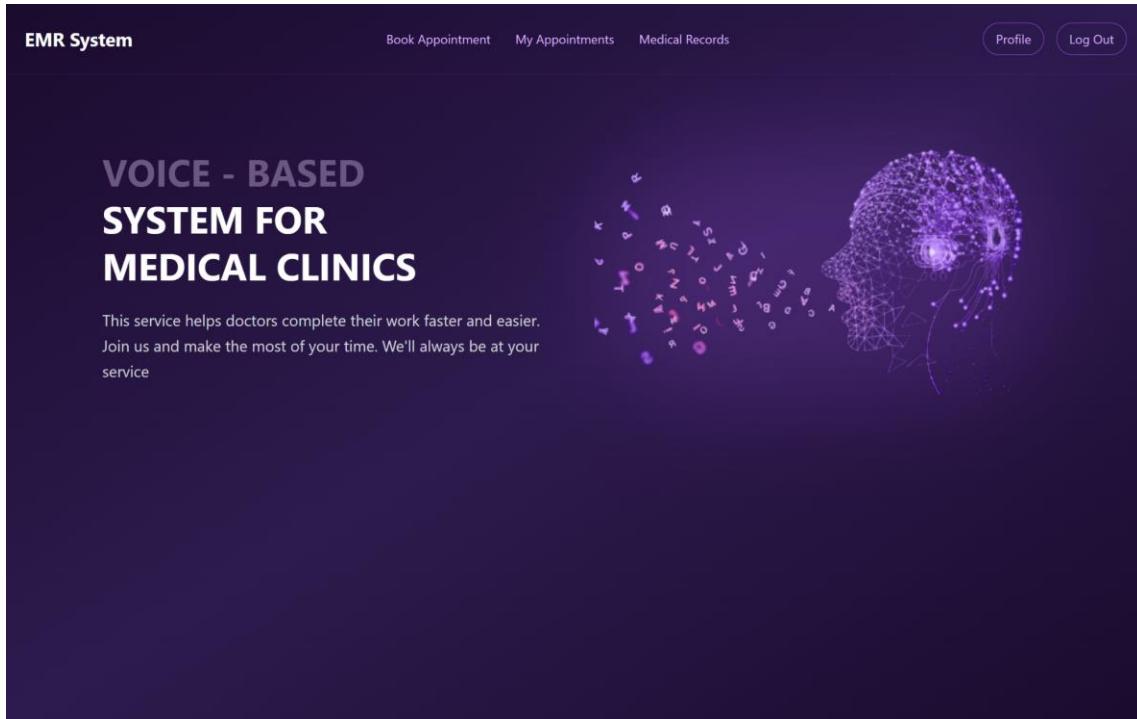


Figure 159, User home page

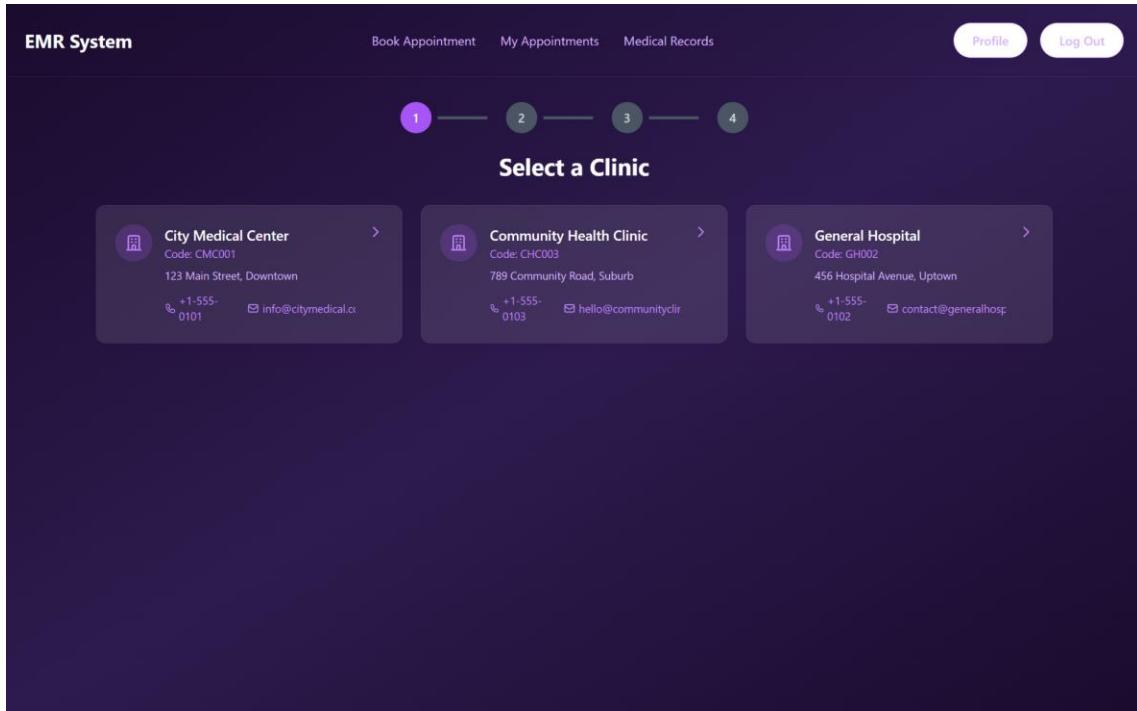


Figure 160, Page for selecting the organization to book an appointment

Chapter 6 - Implementation and Testing

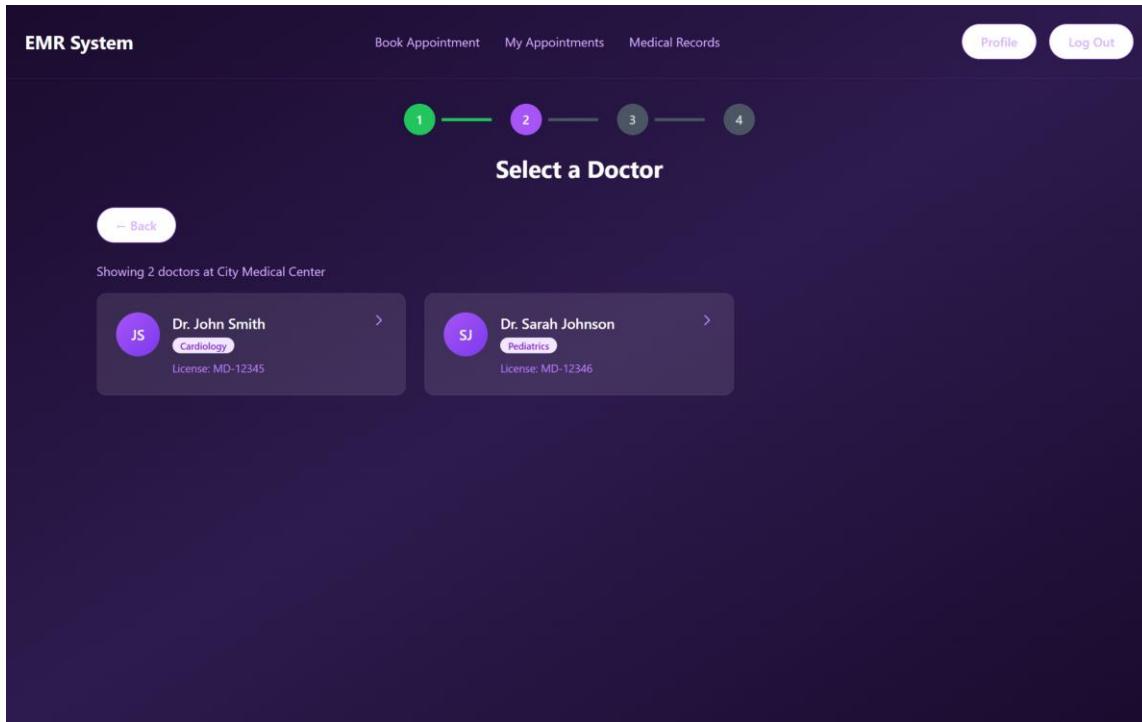


Figure 161, Page for selecting the clinician to book an appointment

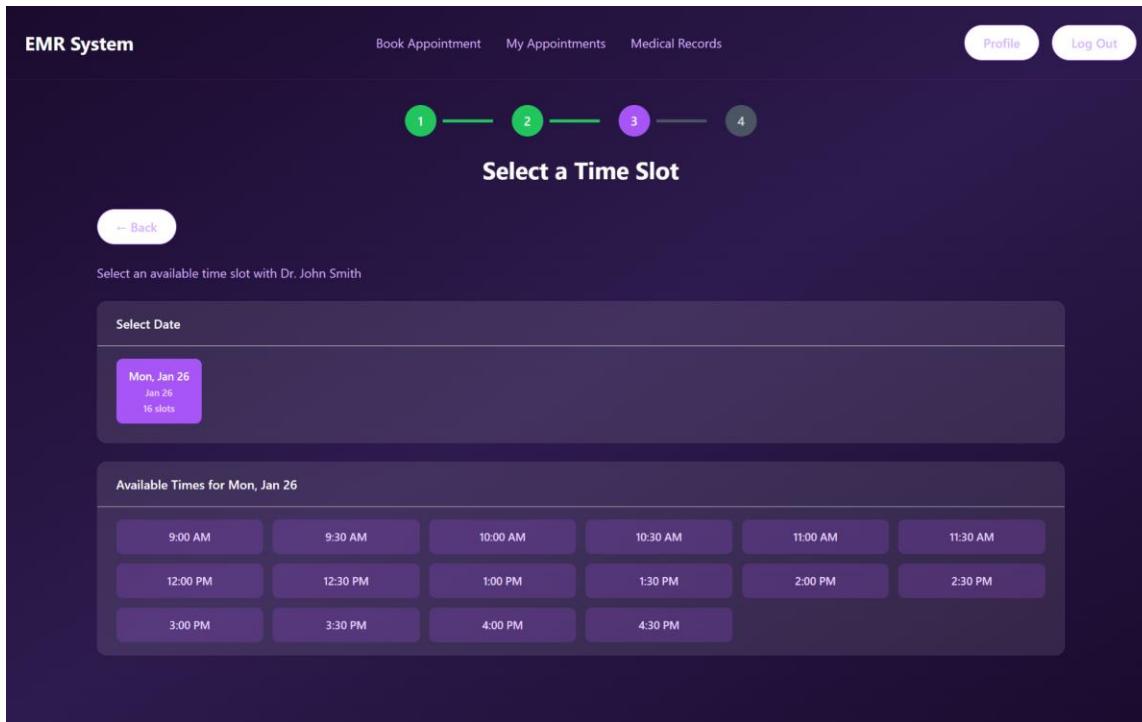


Figure 162, Page for selecting a time slot to book an appointment

Chapter 6 - Implementation and Testing

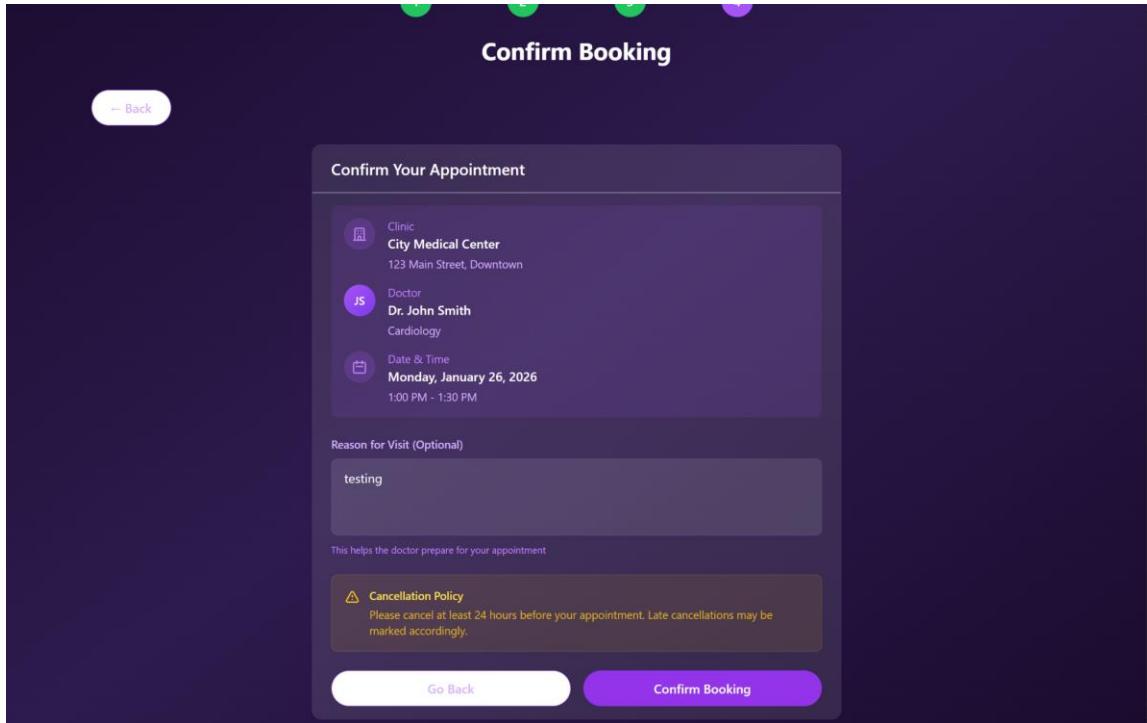


Figure 163, Booking confirmation page

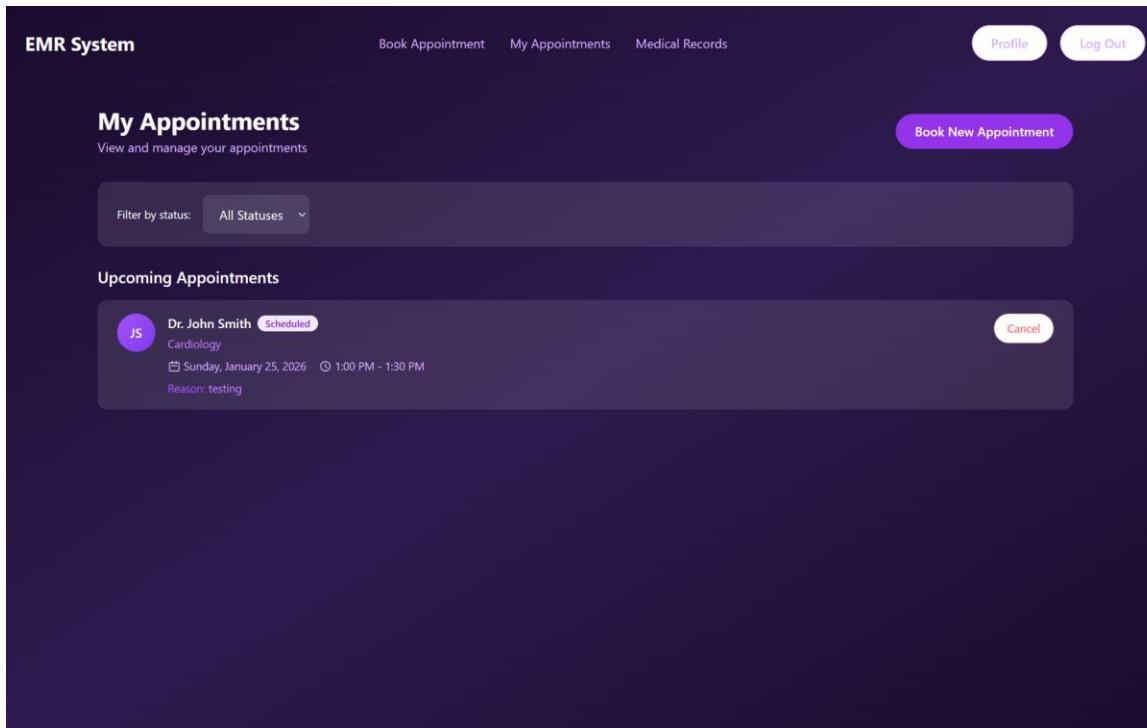


Figure 164, My Bookings View Page

Chapter 6 - Implementation and Testing

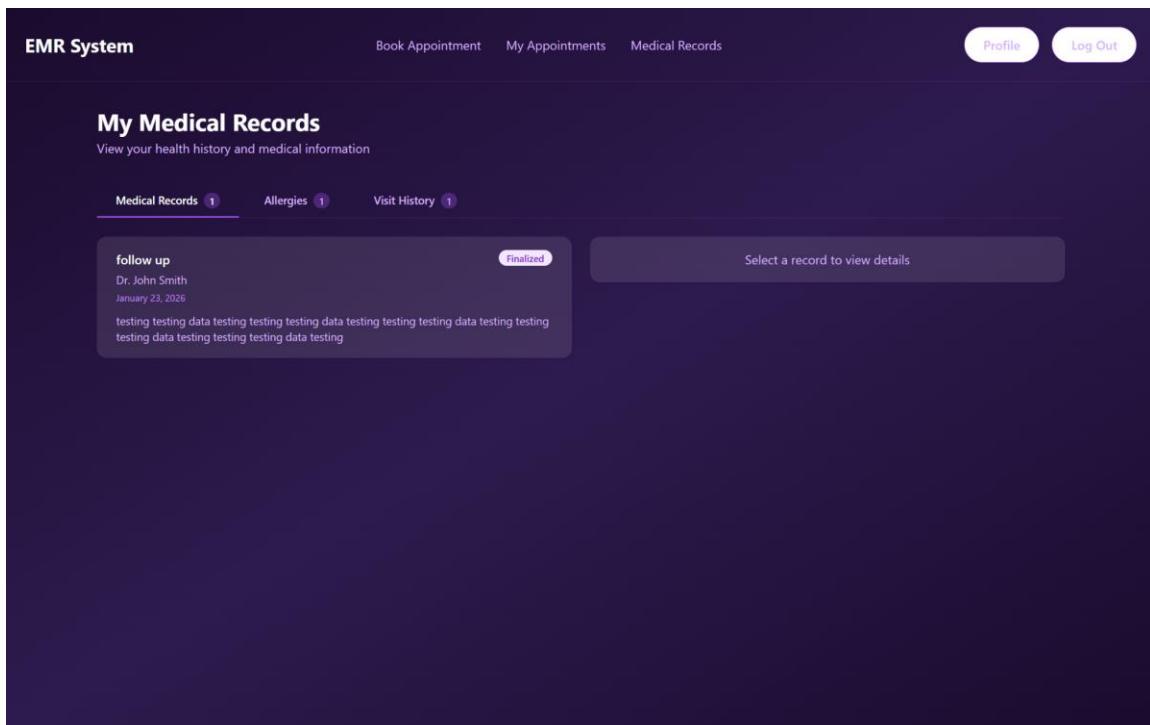


Figure 165, View my medical record page

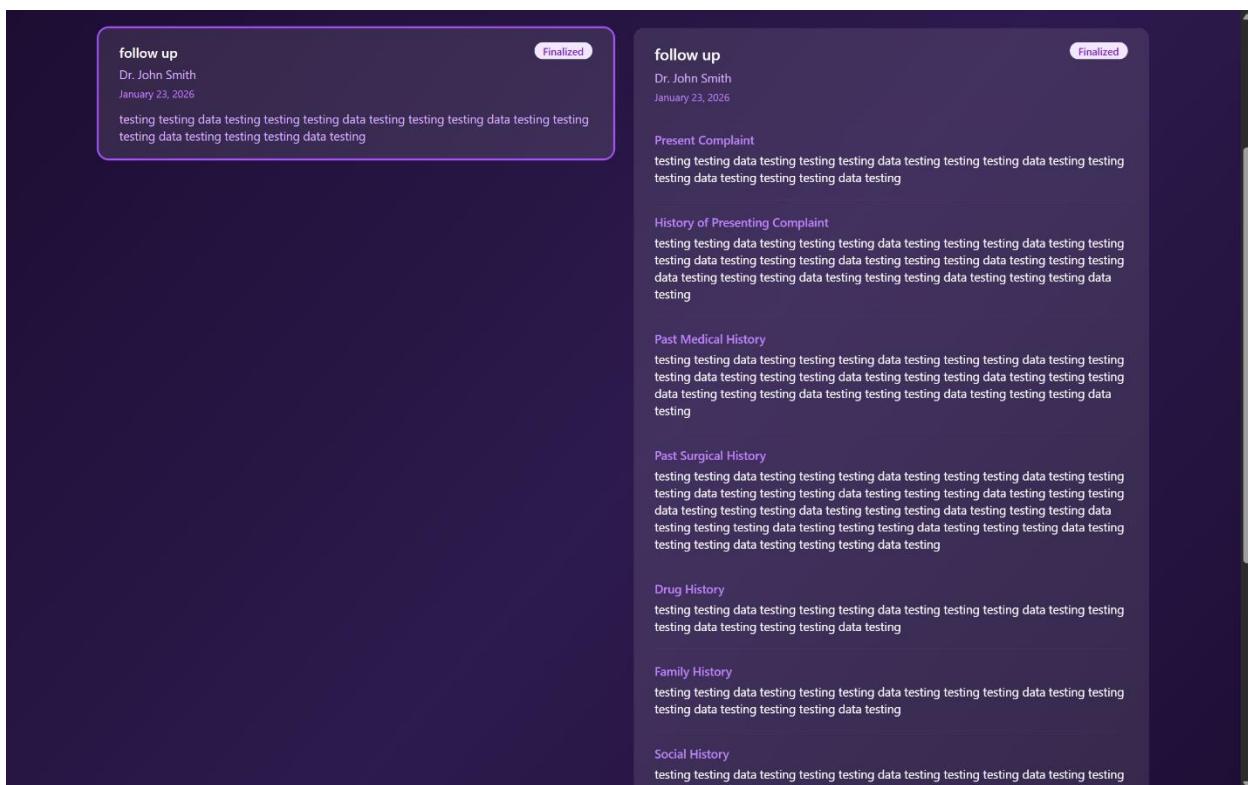


Figure 166, View own Medical Record Details

Chapter 6 - Implementation and Testing

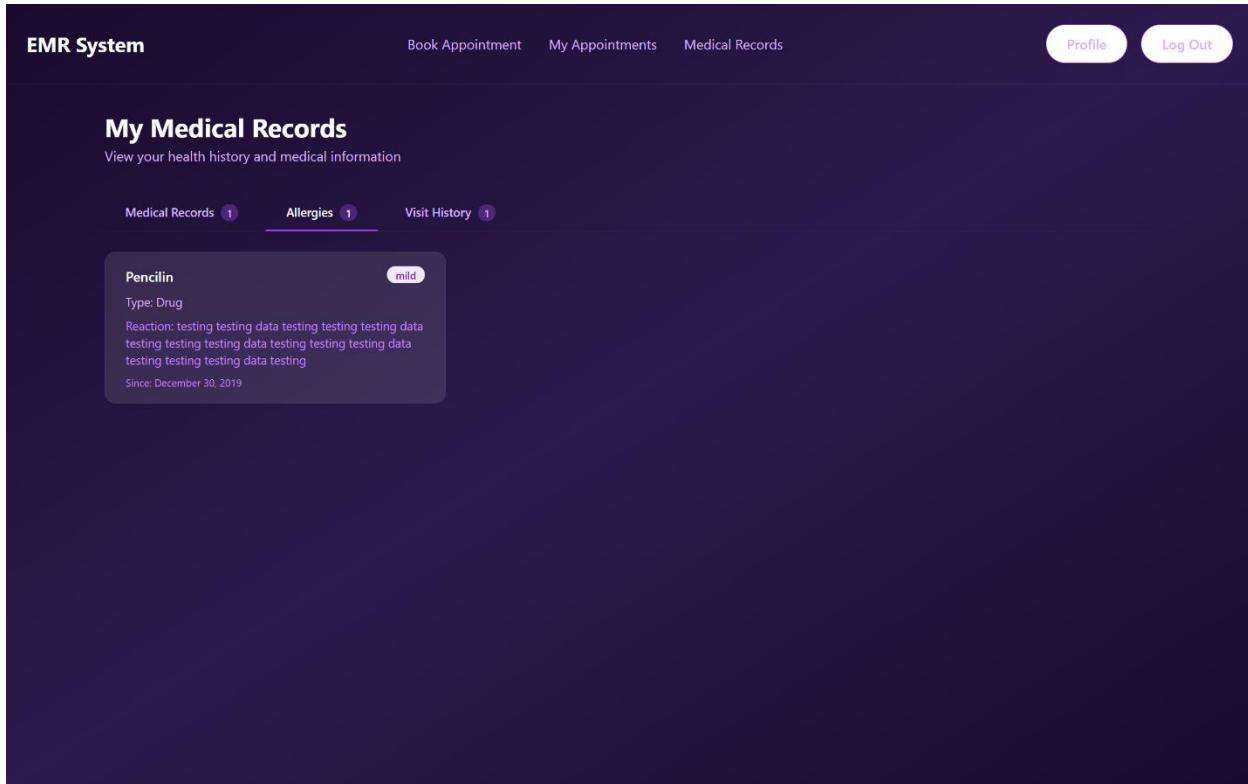


Figure 167, View own allergies

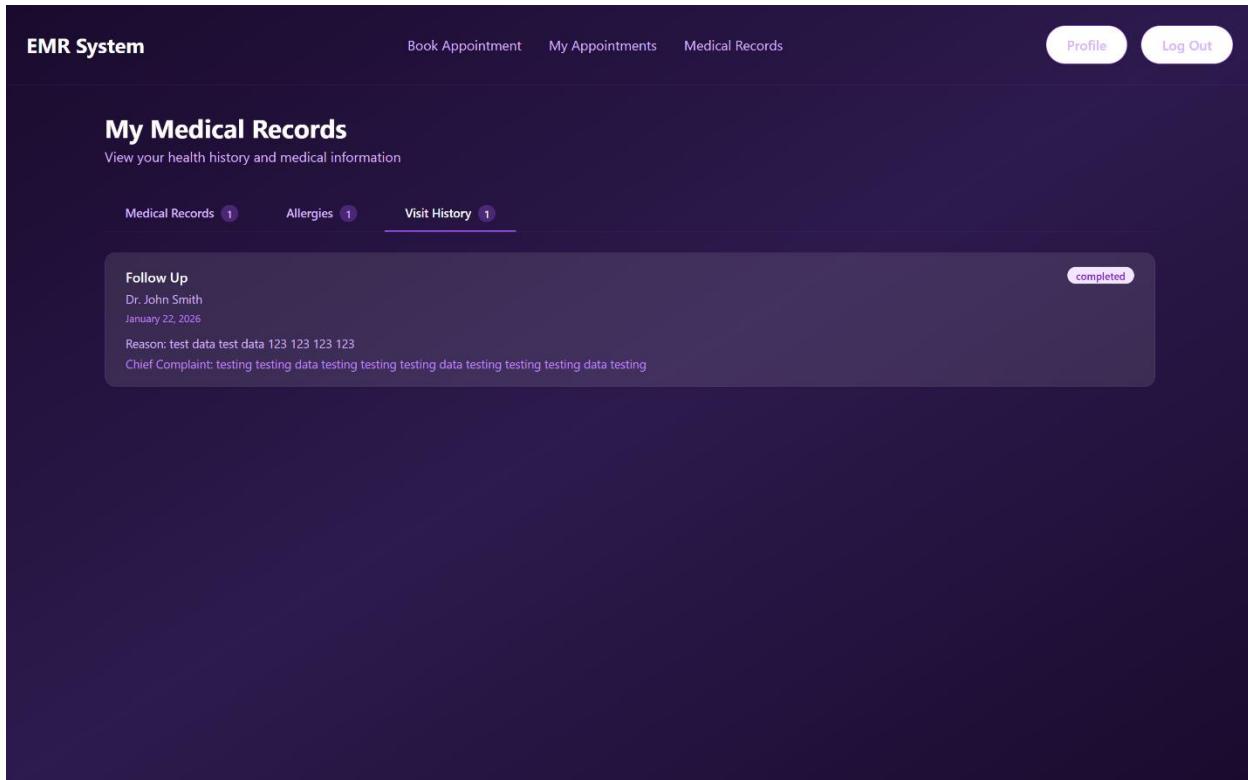


Figure 168, View Own Visit History Details

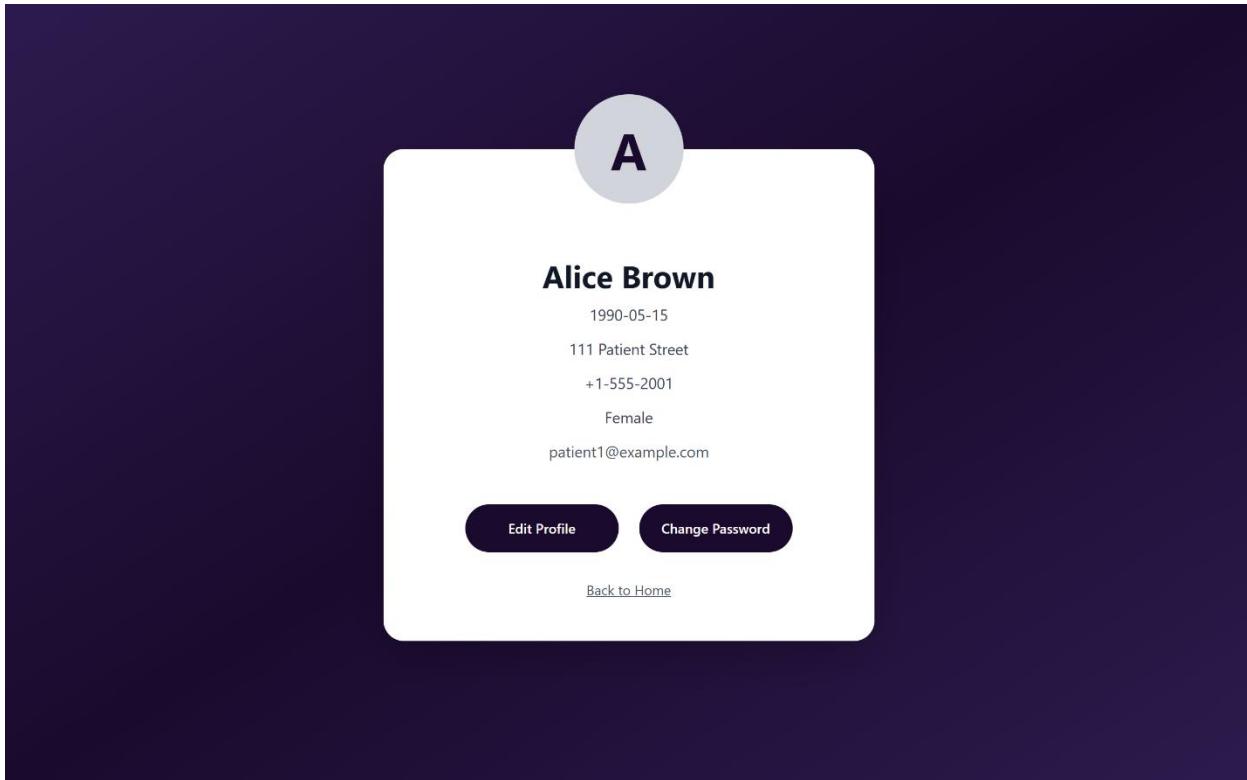


Figure 169, View Own Profile

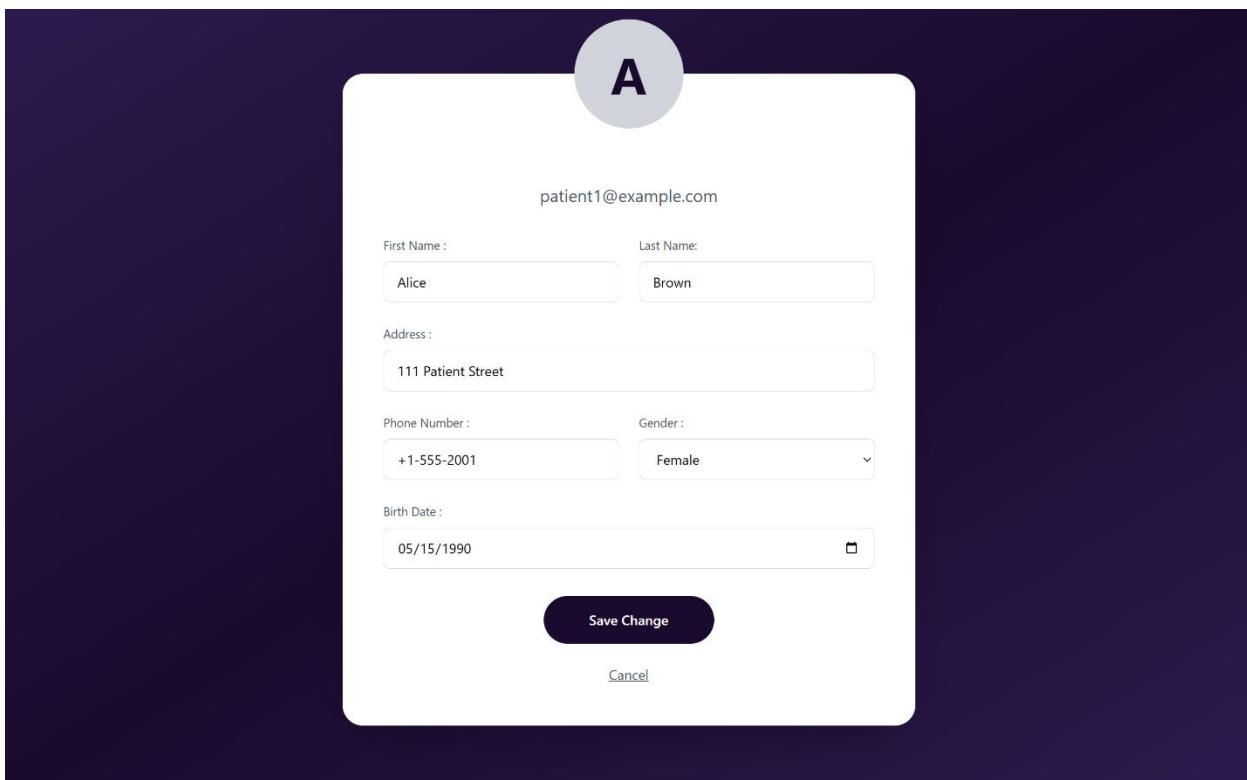


Figure 170, Edit Own Profile

Chapter 6 - Implementation and Testing

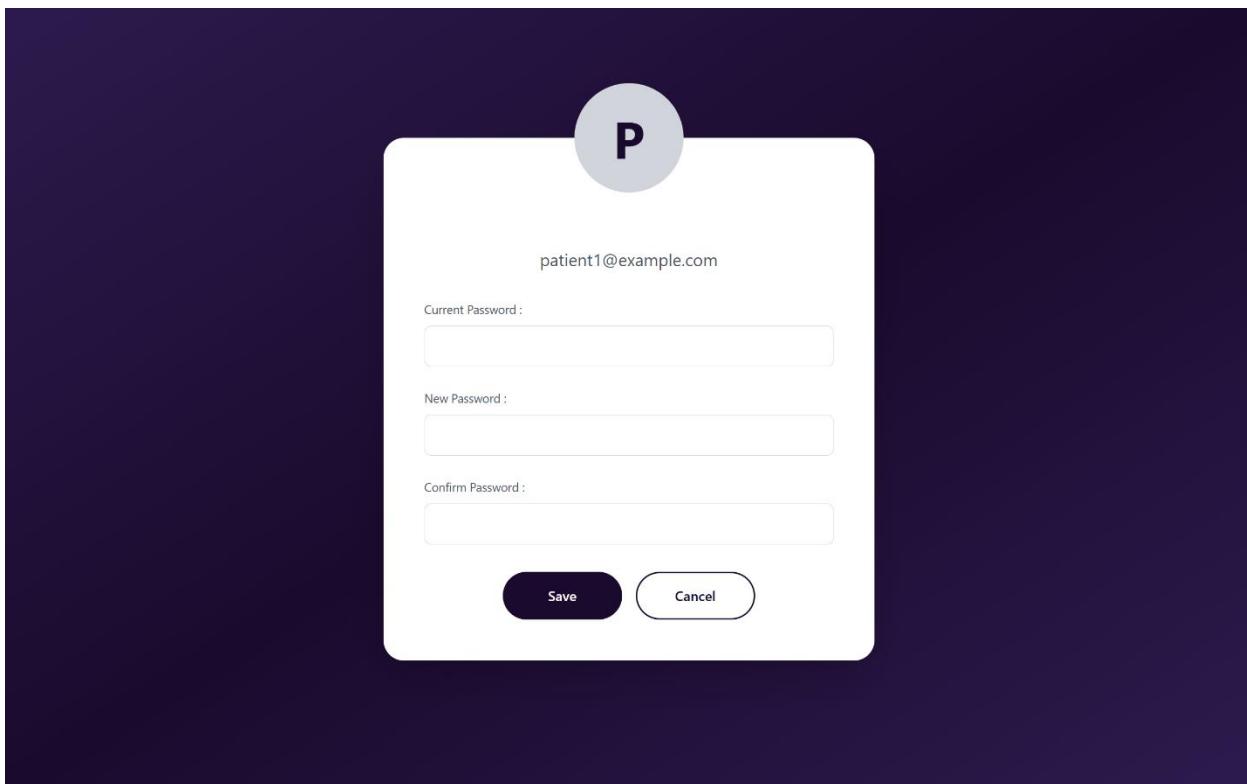


Figure 171, Change Password

A screenshot of the Doctor Dashboard. On the left is a dark sidebar with the title 'EMR Portal' at the top. Below it are five menu items: 'Dashboard' (selected), 'Schedule', 'Appointments', 'Visits', and 'Patients'. At the bottom of the sidebar are 'Dr. Alice Brown' and a 'Logout' link. The main area is titled 'Dashboard' and includes a welcome message 'Welcome back, Doctor'. It features three purple summary cards: 'Active Visits' (0), 'Total Visits' (1), and a 'Quick Action' button labeled 'New Visit' with a plus sign. Below these are two tables: 'Active Visits' (No active visits) and 'Recent Visits' (listing 'Alice Brown' with a 'follow up' note and a 'completed' status).

Active Visits		Recent Visits	
0	View All	Alice Brown	View All
No active visits		follow up • 1/22/2026	
		completed	

Figure 172, View Doctor Dashboard and Analytics

Chapter 6 - Implementation and Testing

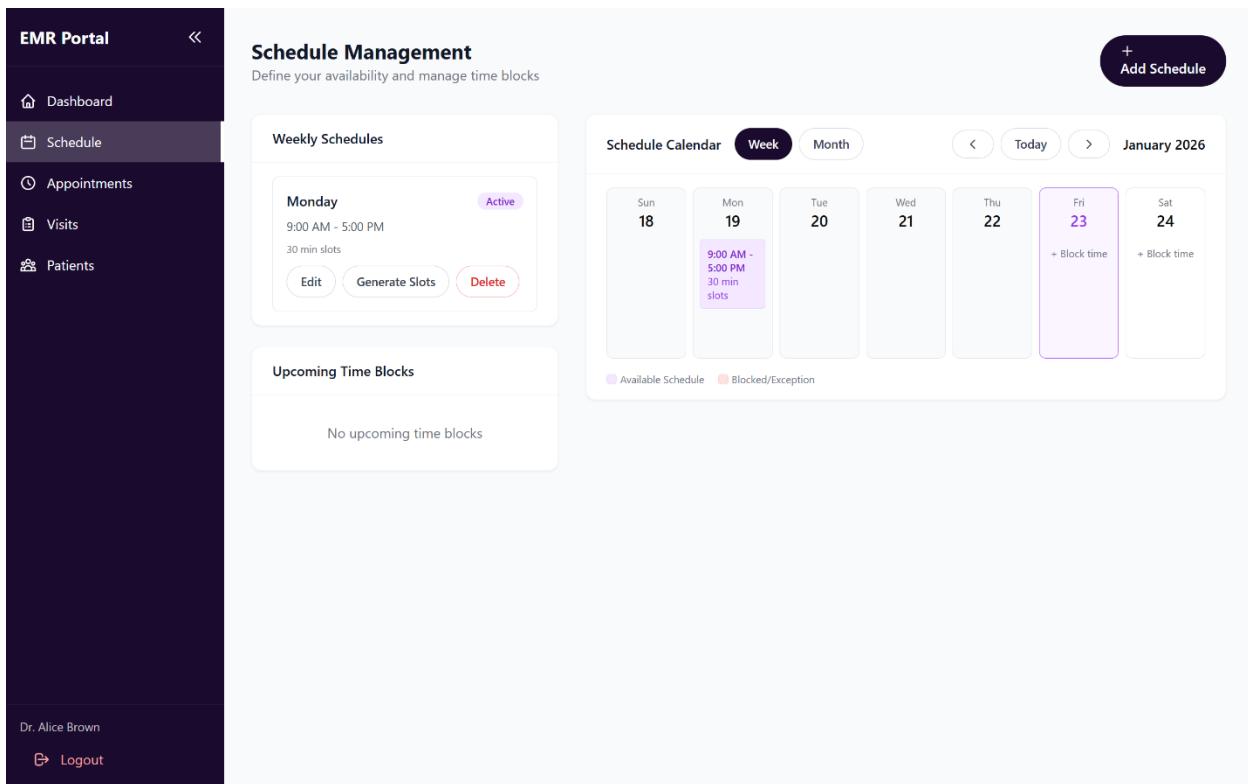


Figure 173, View Own Schedules

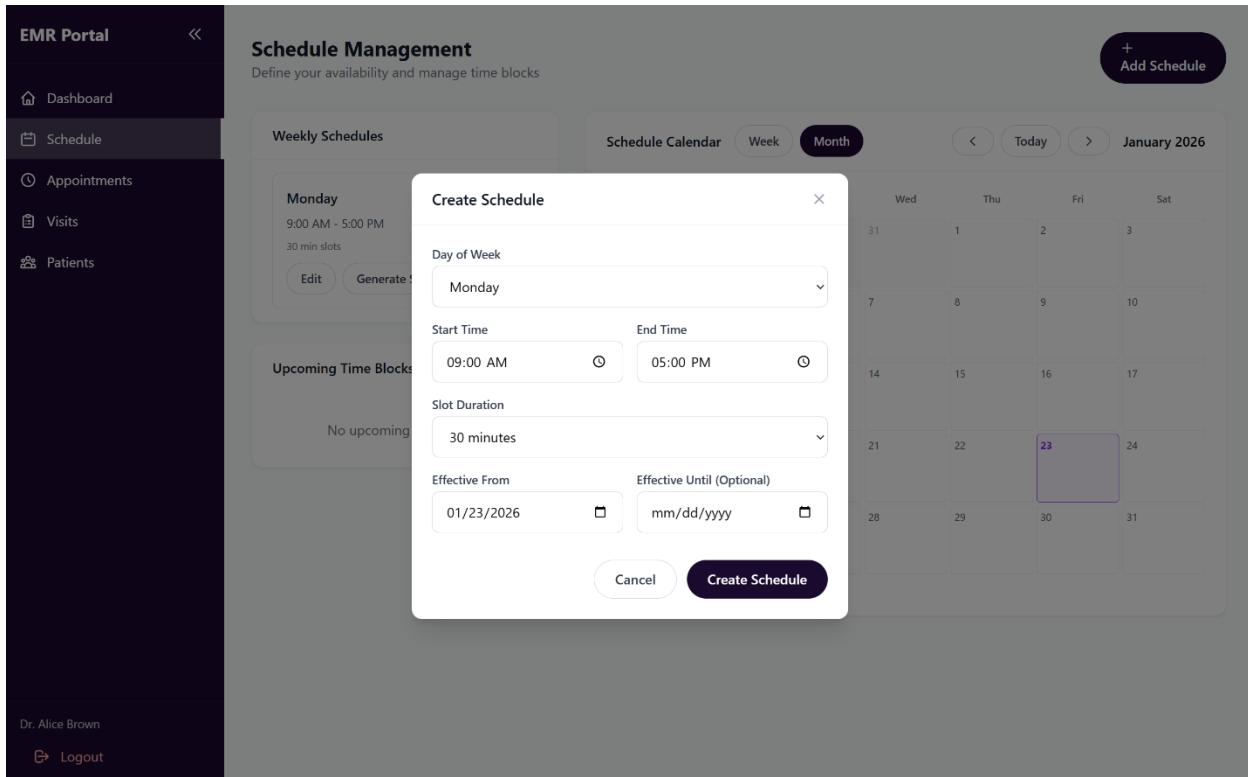


Figure 174, Create Schedule

Chapter 6 - Implementation and Testing

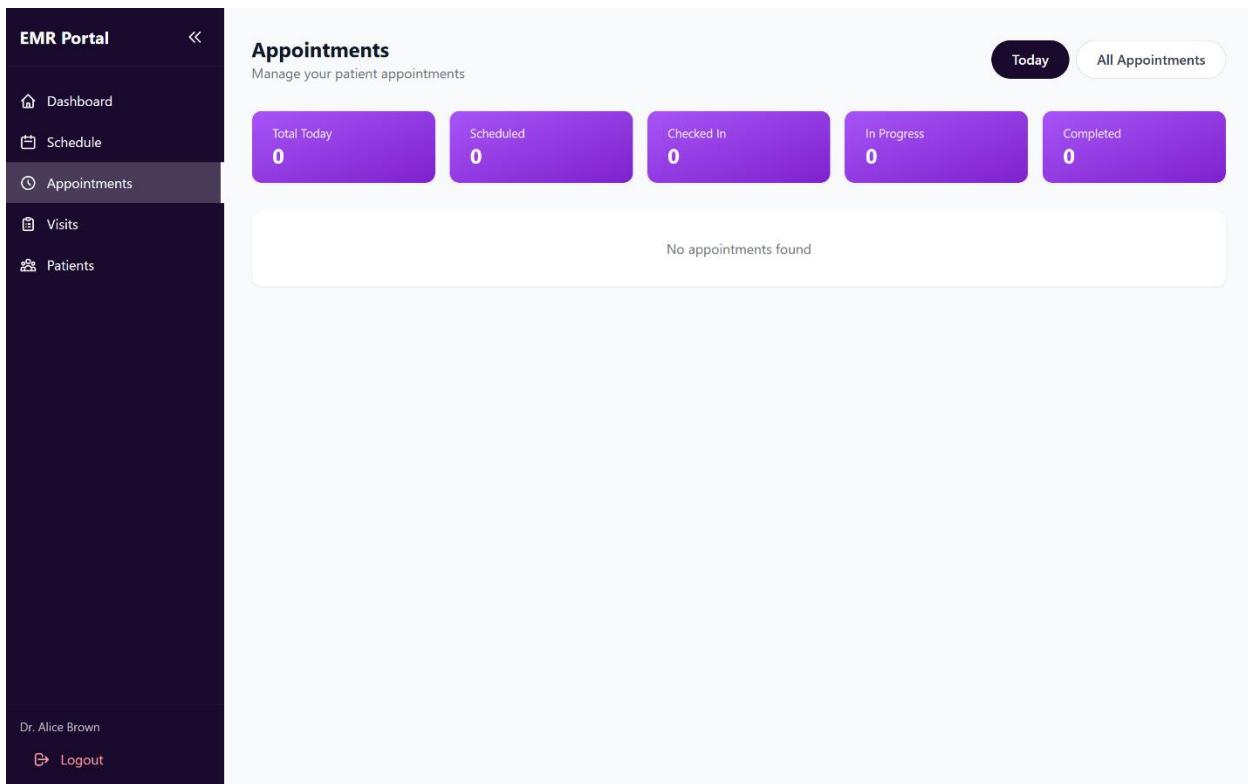


Figure 175, View Own Appointments

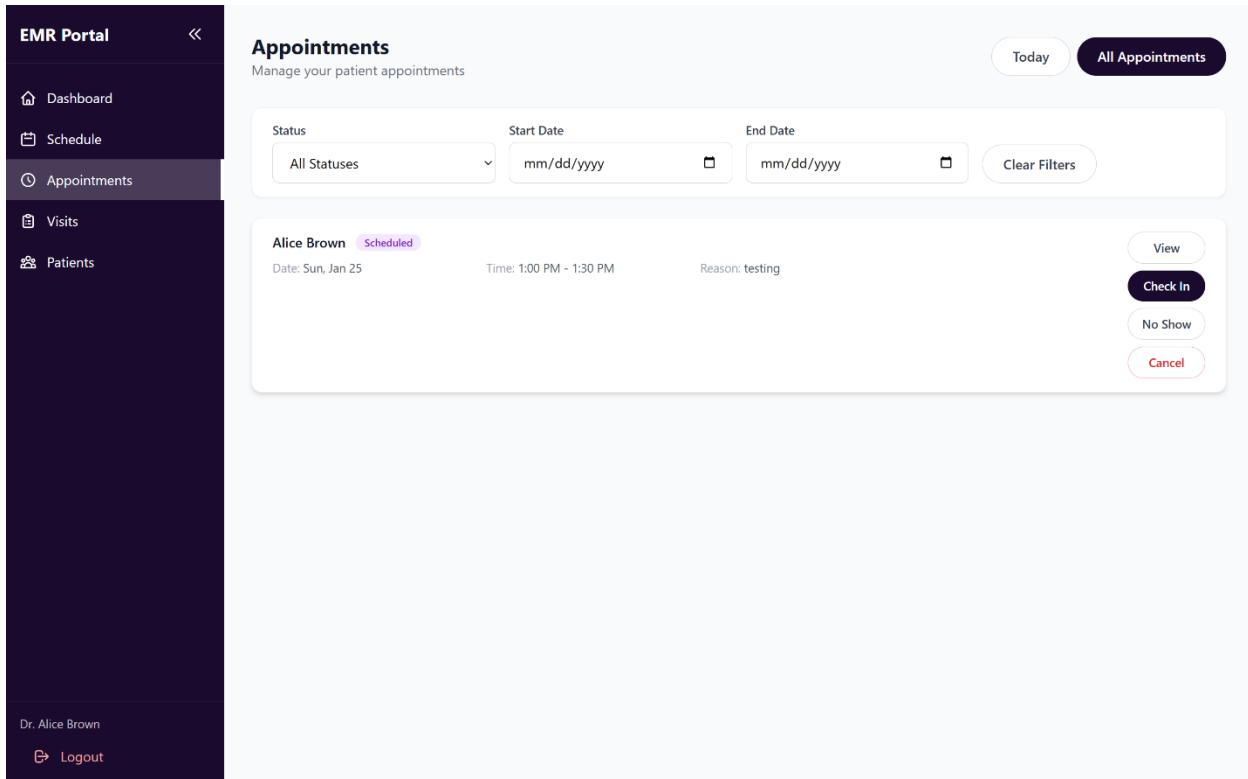


Figure 176, View All Appointments

Chapter 6 - Implementation and Testing

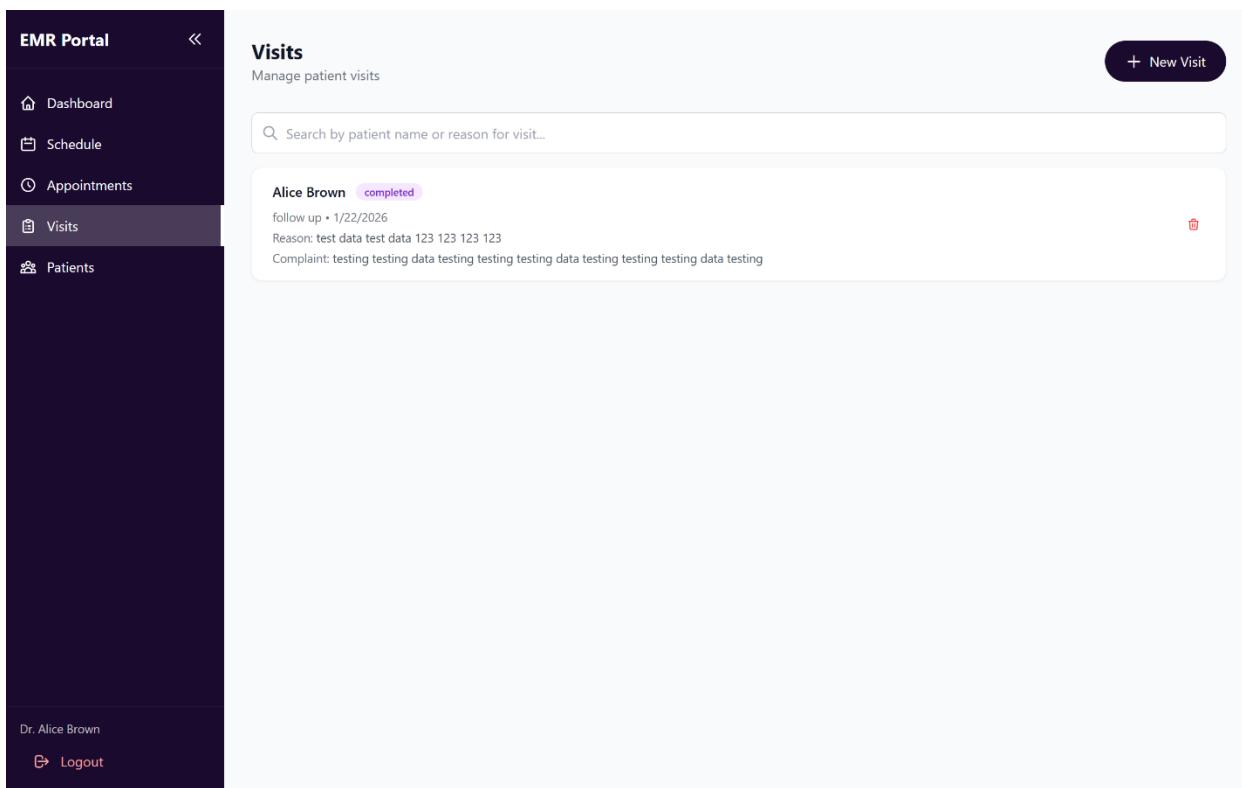


Figure 177, View Visit History

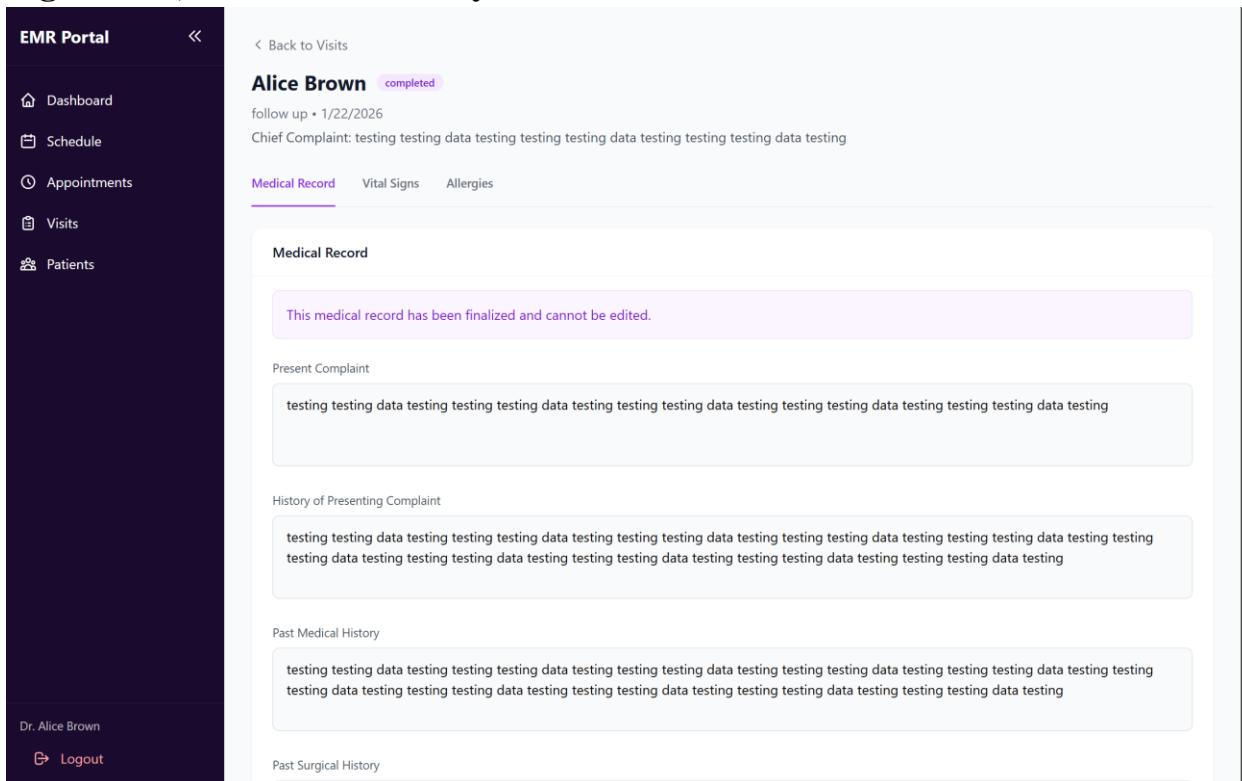


Figure 178, View Medical Record

Chapter 6 - Implementation and Testing

The screenshot shows the EMR Portal interface for a completed visit for patient Alice Brown. The left sidebar includes links for Dashboard, Schedule, Appointments, Visits, and Patients. The main content area shows a summary of the visit, including follow-up information (follow up • 1/22/2026) and chief complaints. Below this, tabs for Medical Record, Vital Signs, and Allergies are present, with Vital Signs being the active tab. A section titled "Vital Signs for this Visit" displays current vital signs: Blood Pressure (50.00 mmHg), Heart Rate (90.00 bpm), Temperature (38.00 °C), Respiratory Rate (120.00 /min), SpO2 (30.00 %), Weight (70.00 kg), Height (170.00 cm), and BMI (100.00 kg/m²). A "Vitals History" section lists previous measurements with their respective times: BMI (100.00 kg/m²) at 12:18:00 PM, Height (170.00 cm) at 12:17:55 PM, Weight (70.00 kg) at 12:17:51 PM, SpO2 (30.00 %) at 12:17:45 PM, Respiratory Rate (120.00 /min) at 12:17:40 PM, and Temperature (38.00 °C) at 12:17:35 PM. The bottom of the page shows a footer with Dr. Alice Brown's name and a Logout link.

Figure 179, View Visit's Vital Signs.

The screenshot shows the EMR Portal interface for a completed visit for patient Alice Brown. The left sidebar includes links for Dashboard, Schedule, Appointments, Visits, and Patients. The main content area shows a summary of the visit, including follow-up information (follow up • 1/22/2026) and chief complaints. Below this, tabs for Medical Record, Vital Signs, and Allergies are present, with Allergies being the active tab. A section titled "Allergies" displays a single entry: Pencillin (mild drug). A note below states: "Reaction: testing testing data testing". The bottom of the page shows a footer with Dr. Alice Brown's name and a Logout link.

Figure 180, View Visit's Allergies.

Chapter 6 - Implementation and Testing

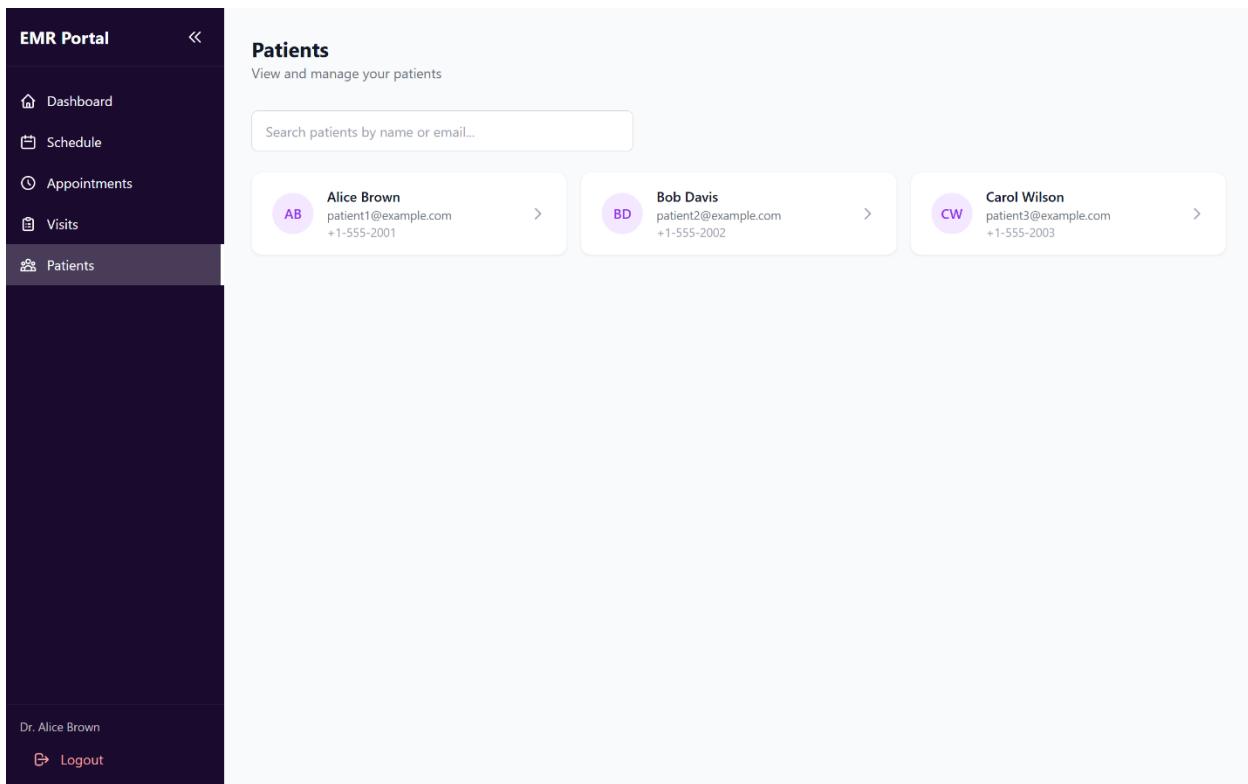


Figure 181, View Patients Lists.

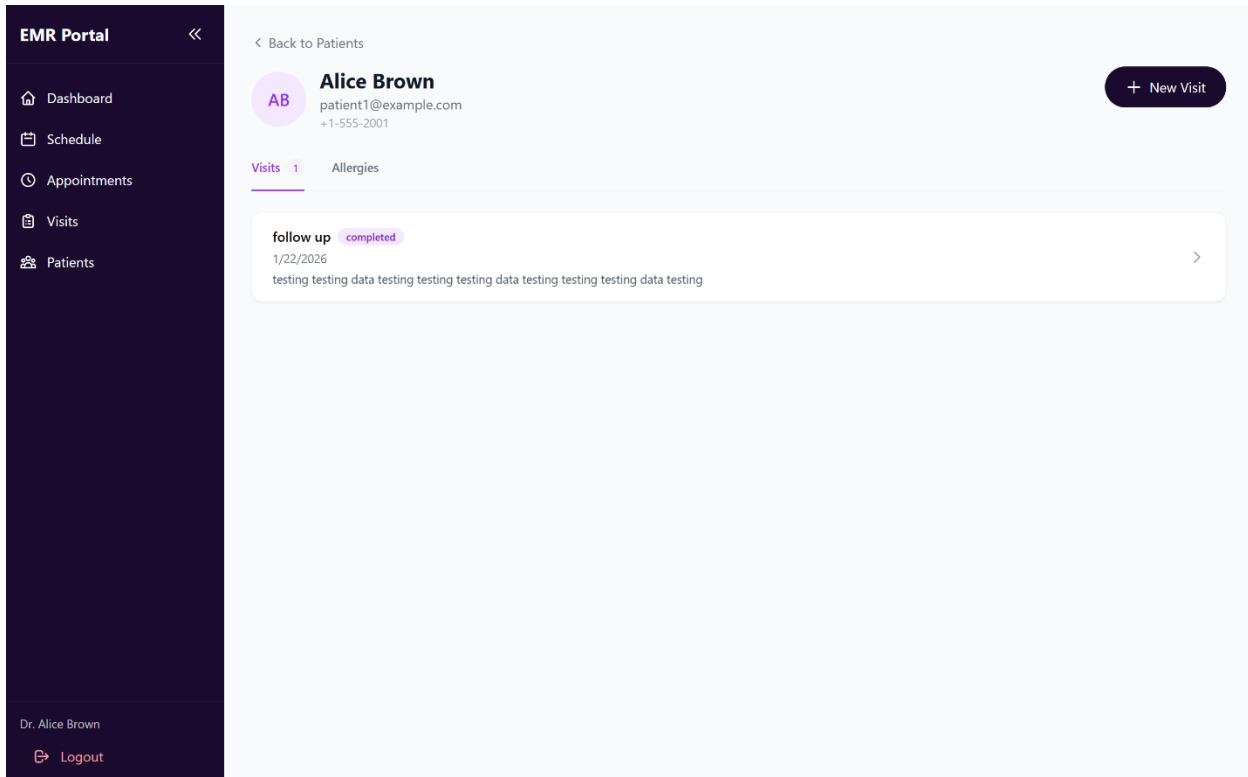


Figure 182, View Patient's Record.

Chapter 6 - Implementation and Testing

The screenshot shows the EMR Portal interface. On the left, a dark sidebar menu includes 'Dashboard', 'Schedule', 'Appointments', 'Visits' (which is selected), and 'Patients'. Below the menu are 'Dr. Alice Brown' and 'Logout' buttons. The main content area displays a patient profile for 'Alice Brown' (patient1@example.com, +1-555-2001). A purple circular icon with 'AB' is next to her name. The 'Visits' tab shows 1 visit. The 'Allergies' tab is selected, showing an entry for 'Pencillin' (mild drug) with the note 'Reaction: testing testing data testing testing testing data testing testing testing data testing testing testing data testing'.

Figure 183, View Patient's Allergies.

The screenshot shows the 'Visits' section of the EMR Portal. A modal window titled 'New Visit' is open. It contains fields for 'Patient' (search bar with 'Alice Brown'), 'Encounter Type' (dropdown set to 'Consultation'), 'Encounter Date' (date input set to '01/23/2026'), 'Reason for Visit' (text input placeholder 'Brief reason for the visit'), 'Chief Complaint' (text input placeholder 'Patient's main complaint'), 'Location' (text input placeholder 'Clinic, Room number, etc.'), and 'Notes' (text input placeholder 'Notes'). There are microphone icons next to each text input field. The background shows the sidebar with 'Visits' selected and a dark overlay of the main content area.

Figure 184, Create New Visit.

Chapter 6 - Implementation and Testing

The screenshot shows the Admin Dashboard of a medical clinic system. The left sidebar includes links for Admin Portal, Dashboard, Organizations, Doctors, and Logout. The main dashboard displays the following information:

- Total Organizations:** 3 (3 active)
- Total Doctors:** 3 (3 active)
- Quick Actions:** Add Organization (Create a new clinic), Add Doctor (Register a new doctor), Manage All (View all records).
- Recent Organizations:**
 - City Medical Center (CMC001) - Active
 - Community Health Clinic (CHC003) - Active
 - General Hospital (GH002) - Active
- Recent Doctors:**
 - Dr. Sarah Johnson (Pediatrics • dr.johnson@emr.com) - Active
 - Dr. John Smith (Cardiology • dr.smith@emr.com) - Active
 - Dr. Michael Williams (General Practice • dr.williams@emr.com) - Active

Figure 185, View Admin Dashboard And Analytics.

The screenshot shows the Organizations page of the Admin Portal. The left sidebar includes links for Admin Portal, Dashboard, Organizations, Doctors, and Logout. The main content area displays the following:

Organizations
Manage clinics and medical facilities

New Organization

Search by name or code...

Name	Code	Email	Phone	Actions
City Medical Center	CMC001	info@citymedical.com	+1-555-0101	
Community Health Clinic	CHC003	hello@communityclinic.com	+1-555-0103	
General Hospital	GH002	contact@generalhospital.com	+1-555-0102	

Figure 186, View Organization List.

Chapter 6 - Implementation and Testing

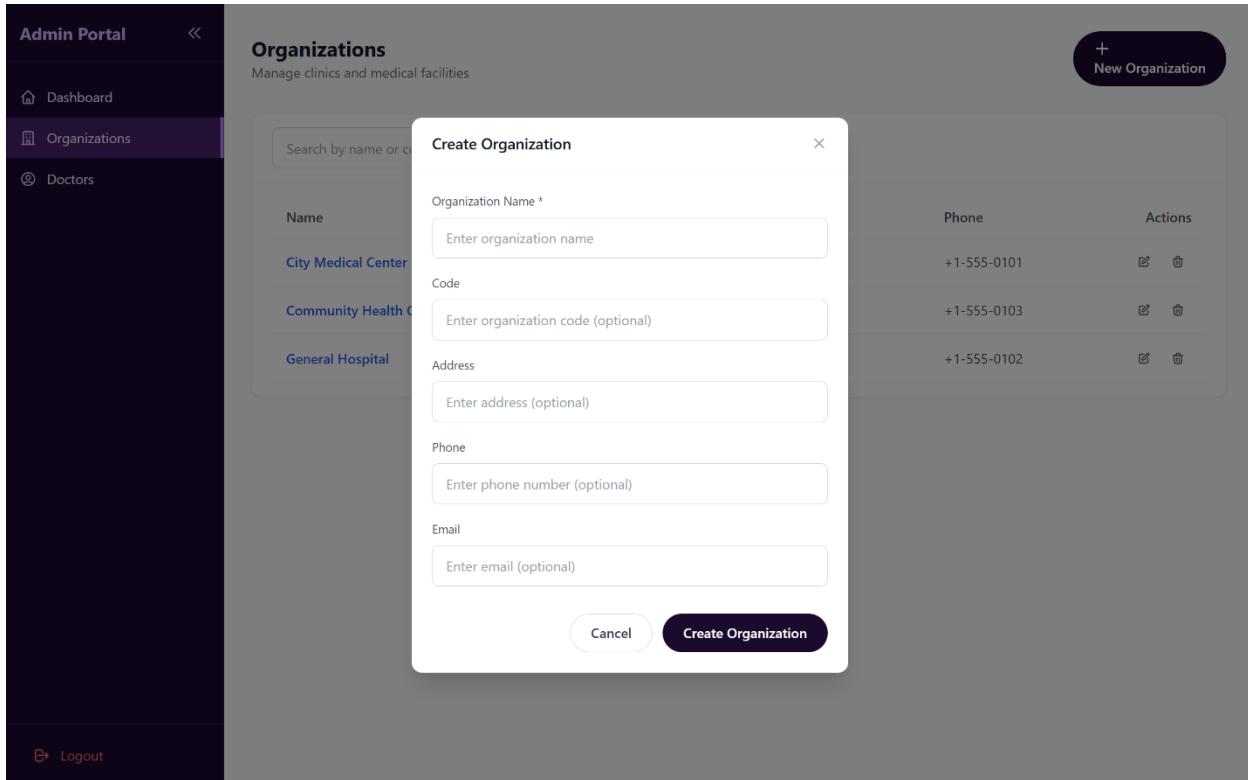


Figure 187, Create New Organization.

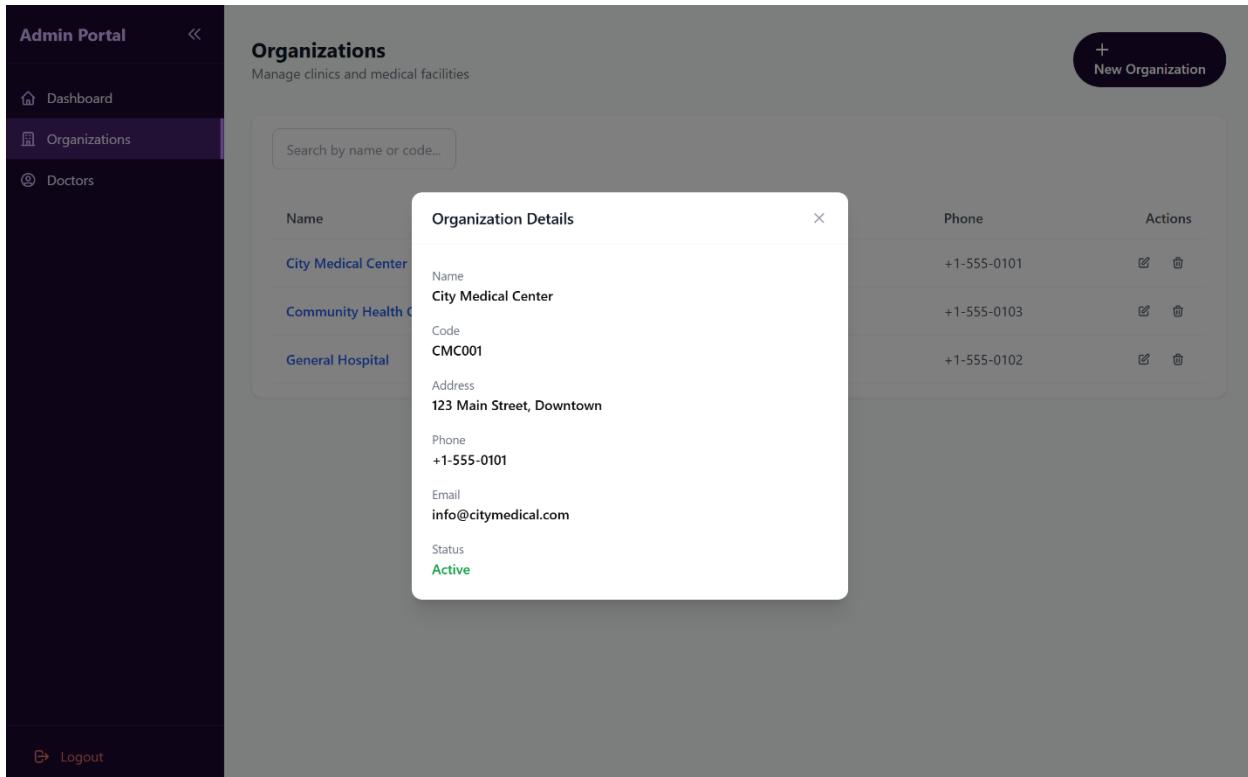


Figure 188, View Organization Details.

Chapter 6 - Implementation and Testing

The screenshot shows the Admin Portal interface. On the left is a dark sidebar with navigation links: Dashboard, Organizations, and Doctors (which is highlighted). At the bottom of the sidebar is a Logout button. The main content area has a header "Doctors" with a subtitle "Manage doctor accounts and assignments". A search bar says "Search by name, email, or". Below is a table with columns: Name, Email, Specialization, Organization, and Actions. The table contains three rows of data:

Name	Email	Specialization	Organization	Actions
Dr. Sarah Johnson	dr.johnson@emr.com	Pediatrics	City Medical Center	
Dr. John Smith	dr.smith@emr.com	Cardiology	City Medical Center	
Dr. Michael Williams	dr.williams@emr.com	General Practice	Community Health Clinic	

In the top right corner of the main content area, there is a button labeled "+ New Doctor".

Figure 189, View Doctors List.

This screenshot shows the "Create Doctor" dialog box overlaid on the Doctors list. The dialog has fields for Email*, Password*, Name (First Name and Last Name), Phone Number, Specialization, License Number, and Organization. At the bottom are "Cancel" and "Create Doctor" buttons. The background Doctors list shows the same three doctors as Figure 189.

Figure 190, Create New Doctor.

Chapter 6 - Implementation and Testing

The screenshot shows the Admin Portal interface. On the left, a sidebar menu includes 'Dashboard', 'Organizations', and 'Doctors' (which is selected). The main content area is titled 'Doctors' with the subtitle 'Manage doctor accounts and assignments'. A search bar at the top right says 'Search by name, email, or phone'. Below it, a table lists three doctors: Dr. Sarah Johnson, Dr. John Smith, and Dr. Michael Williams. A modal window titled 'Doctor Details' is open for Dr. John Smith. It contains the following information:

Name	Email	Specialization	Organization	Actions
Dr. Sarah Johnson			City Medical Center	
Dr. John Smith	dr.john.smith@emr.com	Cardiology	City Medical Center	
Dr. Michael Williams	dr.michael.williams@emr.com		Community Health Clinic	

Modal Content (for Dr. John Smith):

- Name: Dr. John Smith
- Email: dr.smith@emr.com
- Phone: +1-555-1001
- Specialization: Cardiology
- License Number: MD-12345

Logout button is visible at the bottom left of the sidebar.

Figure 191, View Doctors Details.

6.4 Test Cases

Chapter 6 - Implementation and Testing

Test Case ID	Test Name	Purpose	Test Data	Steps	Expected Result	Actual Result
TC-001	Register new user successfully	Verify user registration with email, password hashing, and verification token generation	Email: "john@example.com" Password: "Password123!" First Name: "John" Last Name: "Doe"	1. Check if email exists2. Hash password3. Create user record4. Generate verification token5. Send verification email	User account created successfully with hashed password, verification token generated, and email sent	PASSED
TC-002	Throw ConflictException if email exists	Verify duplicate email prevention	Email: "existing@example.com"	1. Check if email exists2. Throw ConflictException	ConflictException thrown with message "Email already exists"	PASSED
TC-003	Hash password before storing	Verify passwords are securely hashed using bcrypt	Email: "test@example.com" Password: "Password123!"	1. Check email availability2. Hash password using bcrypt3. Verify hash differs from plain password4. Create user with hashed password	Password is hashed and stored securely, hash differs from plain text	PASSED

Chapter 6 - Implementation and Testing

TC-004	Login successfully with valid credentials	Verify login flow with JWT token generation	Email: "test@example.com" Password: "password123"	1. Find user by email 2. Compare password hash 3. Generate JWT tokens 4. Update last login 5. Return tokens and user data	JWT access and refresh tokens returned with user profile (password excluded)	PASSED
TC-005	Throw UnauthorizedException for invalid email	Verify login fails with non-existent email	Email: "wrong@example.com" Password: "password"	1. Find user by email 2. User not found 3. Throw UnauthorizedException	UnauthorizedException thrown with message "Invalid credentials"	PASSED
TC-006	Throw UnauthorizedException for invalid password	Verify login fails with incorrect password	Email: "test@example.com" Password: "wrongpassword"	1. Find user by email 2. Compare password hash 3. Hash comparison fails 4. Throw UnauthorizedException	UnauthorizedException thrown with message "Invalid credentials"	PASSED

Chapter 6 - Implementation and Testing

TC-007	Throw UnauthorizedException if email not verified	Verify unverified users cannot login	Email: "test@example.com" isEmailVerified: false	1. Find user by email2. Check email verification status3. Throw UnauthorizedException	UnauthorizedException thrown with message "Please verify your email"	PASSED
TC-008	Throw UnauthorizedException if account not active	Verify suspended accounts cannot login	Email: "test@example.com" accountStatus: "SUSPENDED"	1. Find user by email2. Check account status3. Throw UnauthorizedException	UnauthorizedException thrown with message "Account is not active"	PASSED
TC-009	Verify email successfully	Verify email verification with valid token	Token: "valid-token-123"	1. Find verification token2. Check token expiration3. Update user email verification status4. Mark token as used	User email marked as verified, token marked as used	PASSED

Chapter 6 - Implementation and Testing

TC-010	Throw BadRequestException for invalid token	Verify invalid tokens are rejected	Token: "invalid-token"	1. Find verification token2. Token not found3. Throw BadRequestException	BadRequestException thrown with message "Invalid or expired verification token"	PASSED
TC-011	Throw BadRequestException for expired token	Verify expired verification tokens are rejected	Token: "expired-token" expiresAt: 25 hours ago	1. Find verification token2. Check token expiration3. Token is expired4. Throw BadRequestException	BadRequestException thrown with message "Verification token has expired"	PASSED
TC-012	Throw BadRequestException for already used token	Verify used tokens cannot be reused	Token: "used-token" isUsed: true	1. Find verification token2. Check if token is used3. Throw BadRequestException	BadRequestException thrown with message "Token has already been used"	PASSED

Chapter 6 - Implementation and Testing

TC-013	Resend verification email successfully	Verify verification email can be resent	Email: "test@example.com" isEmailVerified: false	1. Find user by email2. Delete old tokens3. Generate new token4. Send verification email	New verification token generated and email sent successfully	PASSED
TC-014	Throw BadRequestException if user not found	Verify error handling for non-existent user	Email: "notfound@example.com"	1. Find user by email2. User not found3. Throw BadRequestException	BadRequestException thrown with message "User not found"	PASSED
TC-015	Throw BadRequestException if email already verified	Verify verified users cannot request resend	Email: "test@example.com" isEmailVerified: true	1. Find user by email2. Check verification status3. Throw BadRequestException	BadRequestException thrown with message "Email is already verified"	PASSED

Chapter 6 - Implementation and Testing

TC-016	Send password reset email	Verify password reset email is sent for valid user	Email: "test@example.com"	1. Find user by email2. Delete old reset tokens3. Generate reset token4. Send reset email	Reset token generated and email sent with reset link	PASSED
TC-017	Return generic message if user not found	Verify security by not revealing user existence	Email: "notfound@example.com"	1. Find user by email2. User not found3. Return generic success message	Generic message returned without revealing user doesn't exist	PASSED
TC-018	Reset password successfully	Verify password can be reset using valid reset token	Token: "reset-token-123"New Password: "NewPassword123!"	1. Find reset token2. Check token expiration3. Hash new password4. Update user password5. Mark token as used	Password updated successfully, token marked as used	PASSED

Chapter 6 - Implementation and Testing

TC-019	Throw BadRequestException for invalid reset token	Verify invalid reset tokens are rejected	Token: "invalid-token"	1. Find reset token2. Token not found3. Throw BadRequestException	BadRequestException thrown with message "Invalid or expired reset token"	PASSED
TC-020	Throw BadRequestException for expired reset token	Verify expired reset tokens are rejected	Token: "expired-token" expiresAt: 2 hours ago	1. Find reset token2. Check token expiration3. Token is expired4. Throw BadRequestException	BadRequestException thrown with message "Reset token has expired"	PASSED

Table 78, Test Cases Table - The Rest of Test Cases can be found [here](#)

6.5 RTM V.4

ID	Title	Analysis Section	Design Section	Code	Integration Test	Unit Test
VEMR-FR-PM-01	The system should allow patients to create a new account by providing email, password, first name, and last name.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PM-02	The system should allow patients to login to the system using their email and password credentials.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PM-03	The system should allow patients to verify their email address before accessing the system.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PM-04	The system should allow users to request a password reset link via email and set a new password.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PM-05	The system should allow users to view their profile information including personal details.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PM-06	The system should allow users to update their profile information such as name, phone number, and other details.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PM-07	The system should allow users to change their current password to a new one.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AU-08	The system should allow administrators to login to the system using their credentials.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-AU-09	The system should allow clinicians/doctors to login to the system using their email and password.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-10	The system should allow clinicians to create a new patient visit with date, type, reason, and chief complaint.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-11	The system should allow clinicians to search visits by patient name or reason for visit.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-12	The system should allow clinicians to edit visit details when the visit is in progress.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-13	The system should allow clinicians to save changes made to a visit including medical record data.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-14	The system should allow clinicians to delete a visit from the system.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-15	The system should allow clinicians to view complete details of a visit including medical record, vitals, and allergies.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-VM-16	The system allows the clinician to create and save a new medical record for a patient after validating required information.		Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-17	The system allows the clinician to edit and update an existing medical record while ensuring data validation.		Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-18	The system allows the clinician to finalize a medical record, update its status, and prevent further modifications.		Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-19	The system should allow voice transcription to appear in real-time as the clinician speaks using Whisper.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-20	The system should allow clinicians to view a paginated list of all patients in the system.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-21	The system allows clinicians to search for patient.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-22	The system should allow clinicians to view a patient's profile with their visits and allergies.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-VM-23	The system should allow clinicians to view detailed patient information including contact and demographic data.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-24	The system should allow clinicians to change visit status: start (planned to in-progress), complete, or cancel.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VM-25	The system should allow clinicians to view a paginated list of all visits with filtering options.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VS-26	The system should allow clinicians to view vital signs recorded for a specific visit.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-VS-27	The system should allow clinicians to record and edit vital signs (BP, HR, temp, etc.) during an active visit.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AM-28	The system should allow clinicians to view a patient's recorded allergies with severity and reaction details.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AM-29	The system should allow clinicians to add a new allergy record for a patient with type, allergen, severity, and reaction.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-AM-30	The system should allow clinicians to update existing allergy information for a patient.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AM-31	The system should allow clinicians to delete an allergy record from a patient's profile.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-CM-32	The system should allow the admin to create clinicians' accounts.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-CM-33	The system should allow the clinicians and admins to update their account.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-CM-34	The system should allow the admin to view the clinicians accounts list.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-CM-35	The system should allow the admin to view the clinicians account details.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-CM-36	The system should allow the admin to delete the clinician's accounts.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-CM-37	The system should allow the admin to search the clinician's accounts.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-OM-38	The system should allow the admin to create organizations.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-OM-39	The system should allow the admin to update organization information.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-OM-40	The system should allow the admin to view all organizations.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-OM-41	The system should allow the admin to view detailed information about a specific organization.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-OM-42	The system should allow the admin to delete an organization.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-OM-43	The system should allow the admin to search for organizations.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AN-44	The system should allow administrators to view system-wide analytics and reports.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AP-45	The system should allow clinicians to view their appointment schedule.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-AP-46	The system should allow clinicians to view their daily appointment schedule.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AP-47	The system should allow clinicians to filter appointments by specific date.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AP-48	The system should allow clinicians to filter appointments by status.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AP-49	The system should allow the doctor to check in a patient's appointment.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AP-50	The system should allow the doctor to cancel a patient's appointment.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-AP-51	The system should allow the doctor to set an appointment as no show.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-DA-52	The system should allow clinicians to view their personal analytics and performance metrics.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-SM-53	The system should allow clinicians to create their work schedule.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-SM-54	The system should allow clinicians to edit their existing work schedule.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-SM-55	The system should allow clinicians to delete their work schedule.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-SM-56	The system should allow clinicians to automatically generate available visit slots based on their schedule.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-SM-57	The system should allow clinicians to mark appointments as completed, no-show, or cancelled.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-SM-58	The system should allow clinicians to view detailed information about a specific appointment.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-SM-59	The system should allow clinicians to define their available time slots for appointments.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-60	The system should allow patients to view available healthcare organizations.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-61	The system should allow patients to view doctors within specific organizations.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Chapter 6 - Implementation and Testing

VEMR-FR-PP-62	The system should allow patients to view available appointment slots.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-63	The system should allow patients to book an appointment with a clinician.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-64	The system should allow patients to view their own appointment history and upcoming appointments.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-65	The system should allow patients to filter their appointments by status.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-66	The system should allow patients to view their own medical records.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-67	The system should allow patients to view their own allergy information.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub
VEMR-FR-PP-68	The system should allow patients to view their complete visit history.	Specification and Diagrams	Component Diagram	GitHub	Jira	GitHub

Table 79- RTM V.4

Chapter 7 - Conclusion, future directions and references

7.1 Introduction

The Electronic Medical Records (EMR) System has successfully established a comprehensive healthcare management platform aimed at revolutionizing the way medical practices manage patient care, clinical documentation, and appointment scheduling. This chapter provides a comprehensive summary of the project's achievements, reiterates how the system addresses identified healthcare management challenges, and outlines a future vision for continuous improvement and expansion.

7.2 Summary of Achievements

The development of the EMR System has yielded a robust healthcare management platform with several key achievements:

- **Integrating structured medical record management:** with support for allergies, vital signs, past medical procedures, and review of systems, the platform enables healthcare providers to maintain complete and accurate patient records, significantly improving the quality of clinical documentation and patient care continuity.
- **The system delivers a sophisticated scheduling system with features including:**
 - Real-time availability checking
 - Automated time slot management
 - Smart cancellation policies (early vs. late cancellation)
 - Multiple appointment status tracking (scheduled, checked-in, in-progress, completed, cancelled, no-show)
 - Conflict detection and prevention
- **A comprehensive security framework implementing:**
 - JWT-based authentication with access and refresh tokens
 - Bcrypt password hashing with 10 salt rounds
 - Email verification system
 - Account status management (active, inactive, suspended)
 - Role-specific authorization guards
- **Comprehensive User Management:** A secure system for user registration, authentication, and progress tracking ensures a personalized and continuous learning journey.
- **Intuitive User Interface:** The platform features a user-friendly and responsive UI, designed for seamless navigation across various devices.

7.3 Addressing Key Challenges

The EMR System directly addresses the limitations prevalent in traditional healthcare management:

- **Manual, paper-based medical records are prone to loss, damage, and are difficult to access.**
- **Phone-based scheduling leads to double bookings, missed appointments, and administrative overhead.**
- **Patients have minimal access to their own health information and limited control over appointments.**
- **Incomplete or inconsistent medical records across different visits and providers.**

7.4 System Reliability and Performance

The technical foundation of the EMR System ensures a reliable and performant healthcare environment:

- **Scalability: Architecture:** Modular NestJS backend with TypeORM for efficient database operations
 - **Database:** PostgreSQL with optimized indexes for fast queries
 - **Design:** Designed to handle multiple organizations, hundreds of doctors, and thousands of patients simultaneously
- **Extensibility:**
 - **Layers Structure:** Clean separation of concerns (Gateway → Service → Repository layers)
 - **Easy Integration:** New features can be added without disrupting existing functionality
 - **API-First Design:** RESTful APIs enable future mobile app development
- **Security:**
 - **Authentication:** JWT tokens with 7-day access and 30-day refresh token expiration
 - **Authorization:** Role-specific guards (AdminJwtGuard, DoctorJwtGuard, PatientJwtGuard)
- **Availability:** Aims for high uptime to ensure continuous access to the platform.

7.5 Future Vision: Proposed Enhancements

To further solidify the EMR System's position as a leading healthcare management platform, the following future enhancements are envisioned:

- **7.5.1 Advanced Clinical Features**
 - **E-Prescribing: Electronic prescription management with drug interaction checking**

- **Lab Integration:** Direct integration with laboratory systems for test ordering and results
 - **Imaging Integration:** PACS integration for viewing radiology images
 - **Clinical Decision Support:** AI-powered alerts for drug interactions, allergies, and clinical guidelines
 - **Telemedicine:** Video consultation capabilities for remote patient care
- **7.5.2 Enhanced Scheduling Features**
 - **Recurring Appointments:** Support for follow-up appointment series
 - **Waitlist Management:** Automatic notification when earlier slots become available
 - **Multi-Provider Scheduling:** Book appointments requiring multiple providers
 - **Resource Scheduling:** Schedule rooms, equipment, and other clinic resources
 - **SMS/Email Reminders:** Automated appointment reminders to reduce no-shows
- **7.5.3 Patient Portal Enhancements**
 - **Secure Messaging:** Direct communication between patients and providers
 - **Document Upload:** Patients can upload insurance cards, referrals, and other documents
 - **Family Account Management:** Parents managing children's appointments and records
 - **Health Tracking:** Patient-entered vital signs, symptoms, and medication adherence
 - **Bill Payment:** Online payment for medical services
- **7.5.4 Reporting and Analytics**
 - **Clinical Analytics Dashboard:** Track patient outcomes, quality metrics, and clinical indicators
 - **Financial Reports:** Revenue tracking, billing reports, and insurance claim analytics
 - **Operational Metrics:** Appointment utilization, wait times, and provider productivity
 - **Compliance Reports:** HIPAA audit logs, access reports, and security monitoring
 - **Custom Report Builder:** Allow administrators to create custom reports
- **7.5.5 Mobile Applications**
 - **Native Mobile Apps:** iOS and Android apps for patients and providers
 - **Offline Capability:** Allow doctors to document visits without internet connectivity
 - **Push Notifications:** Real-time alerts for appointments, messages, and lab results
 - **Biometric Authentication:** Fingerprint and face recognition for secure mobile access
- **7.5.6 Interoperability**
 - **HL7/FHIR Integration:** Standard healthcare data exchange protocols
 - **Health Information Exchange (HIE):** Share patient data with other health care systems
 - **Insurance Verification:** Real-time insurance eligibility checking
 - **Pharmacy Integration:** Electronic prescription transmission to pharmacies
- **7.5.7 AI and Machine Learning**
 - **Predictive Analytics:** Predict no-show probability and optimize scheduling
 - **Natural Language Processing:** Auto-populate clinical notes from voice dictation
 - **Risk Stratification:** Identify high-risk patients requiring proactive care

- **Appointment Optimization:** AI-driven scheduling recommendations
- **7.5.8 Administrative Enhancements**
 - **Billing and Claims Management:** Integrated medical billing system
 - **Inventory Management:** Track medical supplies and equipment
 - **Staff Scheduling:** Manage provider schedules and time-off requests
 - **Compliance Management:** Track certifications, licenses, and training requirements
 - **Multi-Location Support:** Enhanced features for healthcare systems with multiple clinics

7.6 Conclusion

The EMR System has successfully created an innovative and effective platform for healthcare management, addressing critical needs in clinical documentation, appointment scheduling, and patient engagement. By implementing modern technologies, robust security measures, and user-centric design, the system empowers healthcare providers to deliver better patient care while improving operational efficiency.

7.7 References

Backend Technologies

- **Node.js**
Node.js Documentation: <https://nodejs.org/docs/>
- **NestJS**
NestJS Documentation: <https://docs.nestjs.com/>
- **TypeScript**
TypeScript Documentation: <https://www.typescriptlang.org/docs/>
- **TypeORM**
TypeORM Documentation: <https://typeorm.io/>
- **PostgreSQL**
PostgreSQL Documentation: <https://www.postgresql.org/docs/>
- **JWT (JSON Web Tokens)**
JWT Introduction: <https://jwt.io/introduction>
- **Bcrypt**
Bcrypt npm Package: <https://www.npmjs.com/package/bcrypt>
- **Jest**
Jest Documentation: <https://jestjs.io/docs/getting-started>

Frontend Technologies

- **React**
React Documentation: <https://react.dev/>
- **TypeScript**
TypeScript Documentation: <https://www.typescriptlang.org/docs/>
- **Tailwind CSS**
Tailwind CSS Documentation: <https://tailwindcss.com/docs>
- **Vite**
Vite Documentation: <https://vitejs.dev/guide/>
- **React Router**
React Router Documentation: <https://reactrouter.com/>

Healthcare Standards

- **HL7 FHIR**
FHIR Documentation: <https://www.hl7.org/fhir/>
- **HIPAA Compliance**
HIPAA Guidelines: <https://www.hhs.gov/hipaa/>

Development Resources

- **MDN Web Docs**
MDN Web Docs: <https://developer.mozilla.org/>
- **Stack Overflow**
Stack Overflow: <https://stackoverflow.com/>
- **GitHub**
GitHub: <https://github.com/>

Testing and Quality

- **Jest Testing Framework**
Jest Documentation: <https://jestjs.io/>
- **ESLint**
ESLint Documentation: <https://eslint.org/docs/>