



Palestine Technical University - Kadoorie
Faculty of Engineering and Technology
Department of Computer Systems Engineering



Software Engineering Requirements Document
Medical Care System

Prepared By:

Raghad Hanon	202110023	3
Abeer Mahmoud	202110119	4
Hanan Issa	202110057	2
Nada Asad	202112529	4

Supervisor:

Dr. Osama Hamed

Abstract

As technology continues to advance, its impact on society becomes increasingly evident, touching various aspects of our lives. Nowhere is this more pronounced than in the field of medicine. To keep pace with these advancements, it has become essential to embrace technology that not only saves time and effort for users but also enhances the delivery of healthcare services.

One of the remarkable achievements of modern medicine is the development of applications and websites that leverage technological progress to offer convenience and efficiency to both healthcare providers and patients. Our project aims to provide an archiving and healthcare system suitable for use in clinics as well as for personal use by patients. It is designed to streamline access to patients' medical histories, offering several advantages over traditional methods where doctors obtain patient information directly from the patient. Additionally, it allows patients to contribute to their own medical history.

In our healthcare system, we offer the following features:

1. Medical information: This can be written or reviewed on a patient's profile by healthcare providers after a patient has requested to contact them, provided a description of their medical conditions, and then undergone a medical examination. This information includes:
 - 1.1 Allergies: Information regarding any known allergies the patient may have to medications, foods, or other substances.
 - 1.2 Chronic diseases: Details about any chronic conditions or ongoing health issues the patient has, such as diabetes, hypertension, asthma, or heart disease, are recorded in their medical records.
 - 1.3 Medicines: Details about any medicines that the patient takes.
 - 1.4 Vital signs of the patient: Records of the patient's vital signs, including blood sugar levels, blood pressure, which may be recorded during routine check-ups, hospital visits, or other medical encounters.
2. Personal information for users: Basic information about the user, such as name, gender, contact information and address, other medical information such as blood type ,weight and height for the patient , CV and clinic information for the doctor.
3. List of doctors categorized based on their medical specialization: Patients can choose a doctor that matches their needs and request treatment accordingly.
4. List of patients for each doctor: Each doctor has a list of patients they have treated, allowing for efficient management of patient-doctor relationships.

By centralizing this information in a digital format, our system aims to enhance the efficiency and effectiveness of healthcare delivery, benefiting both patients and healthcare providers alike.

- Frontend GitHub repo: <https://github.com/RaghadHanon/MedicoPortal>
- Backend GitHub repo: <https://github.com/RaghadHanon/MedicoPortalBackend>
- Video introducing our website: <https://drive.google.com/file/d/1QIMdOefKHsPtD-sCjVeVMpC4JDylG5B5/view>

مع استمرار تقدم التكنولوجيا، يصبح تأثيرها على المجتمع أكثر وضوحًا، حيث تلامس جوانب مختلفة من حياتنا. ولا يوجد مكان يتجلى فيه هذا التأثير أكثر من مجال الطب. لمواكبة هذه التطورات، أصبح من الضروري تبني التكنولوجيا التي لا توفر الوقت والجهد للمستخدمين فحسب، بل تعزز أيضًا تقديم خدمات الرعاية الصحية.

واحدة من الإنجازات الرائعة للطب الحديث هي تطوير التطبيقات والمواقع الإلكترونية التي تستفيد من التقدم التكنولوجي لتقديم الراحة والكفاءة لكل من الأطباء والمرضى. يهدف مشروعنا إلى توفير نظام أرشفة ورعاية صحية مناسب للاستخدام في العيادات وكذلك للاستخدام الشخصي من قبل المرضى، وهو مصمم لتبسيط الوصول إلى السجلات الطبية للمرضى، مقدماً عدة مزايا مقارنة بالطرق التقليدية حيث يحصل الأطباء على معلومات المريض مباشرة منه، بالإضافة إلى ذلك، يسمح للمرضى بالمساهمة في تاريخهم الطبي.

في نظام الرعاية الصحية الخاص بنا، نقدم الميزات التالية:

1. المعلومات الطبية: يمكن إضافتها أو مراجعتها في ملف المريض من قبل الأطباء بعد أن يقوم المريض بالتواصل معهم موفراً وصفاً لحالته الطبية، وعند الحاجة القيام بزيارة للعيادة لكي يخضع لفحص طبي.

تتضمن هذه المعلومات:

- تشخيص الحساسيات : معلومات حول أي حساسية معروفة لدى المريض تجاه الأدوية أو الأطعمة أو المواد الأخرى.
- الأمراض المزمنة: تفاصيل حول أي حالات مزمنة أو مشاكل صحية مستمرة لدى المريض، مثل السكري، ارتفاع ضغط الدم، الربو، أو أمراض القلب، مسجلة في سجلاتهم الطبية.
- الأدوية: تفاصيل حول أي أدوية يتناولها المريض.
- العلامات الحيوية للمريض: سجلات العلامات الحيوية للمريض، بما في ذلك مستويات السكر في الدم، ضغط الدم، والتي قد يتم تسجيلها أثناء الفحوصات الروتينية أو زيارات المستشفى أو غيرها من اللقاءات الطبية.

2. المعلومات الشخصية للمستخدمين: معلومات أساسية حول المستخدم، مثل الاسم، الجنس، معلومات الاتصال، والعنوان، بالإضافة إلى المعلومات الطبية الأخرى مثل فصيلة الدم والوزن والطول للمريض، والسيرة الذاتية ومعلومات العيادة الطبية للطبيب.

3. قائمة الأطباء المصنفة بناءً على تخصصاتهم الطبية: يمكن للمرضى اختيار طبيب يتناسب مع احتياجاتهم وطلب العلاج بناءً على ذلك.

4. قائمة المرضى لكل طبيب: لكل طبيب قائمة بالمرضى الذين قاموا بمعالجتهم، مما يسمح بإدارة فعالة لعلاقات المريض بالطبيب ومراجعة المعلومات الطبية التي تم اضافتها لكل منهم.

من خلال توثيق هذه المعلومات بشكل رقمي، يهدف نظامنا إلى تعزيز كفاءة وفعالية تقديم الرعاية الصحية، مما يعود بالفائدة على كل من المرضى ومقدمي الرعاية الصحية على حد سواء.

- رابط الجيت هب للفرONT : <https://github.com/RaghadHanou/MedicoPortal>
- رابط الجيت هب للباك : <https://github.com/RaghadHanou/MedicoPortalBackend>
- فيديو مرفق لتوضيح تفاصيل العمل : <https://drive.google.com/file/d/1QIMdOefKHsPtDhthttps://drive.google.com/file/d/1QIMdOefKHsPtDht/view>

* ترجمة يدوية

Name	Role
Raghad Hanon	In the Medico Portal Project, my responsibilities included implementing the back-end part using Visual Studio and designing the database on SQL Server. Additionally, I played a role in designing and implementing the overall structure of the react project, home page , features page , doctor registration page, and doctor profile page. I also contributed to writing the Software Requirements Specification (SRS) and digitizing doctor sequential diagrams using EdrawMax.
Abeer Mahmoud	In the Medico Portal Project, my responsibilities included participation in writing and designing the Software Requirements Specification (SRS) and digitizing diagrams using Microsoft Visio. Additionally, I contributed to design the doctors list page, doctor card page, and response page using React.
Hanan Issa	In the Medico Portal Project, my responsibilities included participation in writing and designing the Software Requirements Specification (SRS) and digitizing diagrams. Additionally I built the login page, response form page ,and part of the doctor profile page, and a loader page in the front-end.
Nada Asaad	In the Medico Portal Project, I contributed to designing the patient profile, medical specifications page, and patient registration page using React. Additionally, I played a role in writing and designing the Software Requirements Specification and digitizing sequential diagrams using Microsoft Visio.

Table of contents

1. CHAPTER 1: Introduction	7
1.1 Purpose	7
1.2 Document Conventions	7
1.3 Intended Audience and Reading Suggestions	8
1.3.1 Developer	8
1.3.2 End users	8
1.3.2.1 Healthcare Professionals	8
1.3.2.2 Patients	9
1.3.2.3 General Users	9
1.4 Project Scope	9
1.5 References	9
2 CHAPTER 2: Overall Description	9
2.1 Product Perspective	9
2.2 Product Features	10
2.3 User Classes and Characteristics	11
2.3.1 Patients	11
2.3.1 Healthcare Providers:	11
2.3.2 System Administrators:	11
2.3.3 Other Stakeholders:	11
2.4 Operating Environment	12
2.5 Design and Implementation Constraints	12
2.5.1 User Interface (UI) and User Experience (UX):	12
2.5.2 Interoperability:	12
2.5.3 Scalability:	12
2.5.4 Data Security and Privacy:	12
2.5.1 Performance	13
2.6 Assumptions and Dependencies	13
3 CHAPTER 3: System Features	14
3.1 Patient-Doctor Relationship Management	14
3.1.1 Description and Priority:	14
3.1.2 Stimulus/Response Sequences:	14
3.1.3 Functional Requirements:	14
3.2 Doctor's Directory	15
3.2.1 Description and Priority:	15
3.2.2 Stimulus/Response Sequences:	15
3.2.3 Functional Requirements:	15
3.3 Patient Health Record	15
3.3.1 Description and Priority:	15
3.3.2 Stimulus/Response Sequences:	15
3.3.3 Functional Requirements:	16
3.4 Account Management	17
3.4.1 Description and Priority:	17
3.4.2 Stimulus/Response Sequences:	17
3.4.3 Functional Requirements:	17
4 CHAPTER 4: External Interface Requirements	17
4.1 User Interfaces:	17
4.2 Hardware Interfaces	18
4.3 Software Interfaces	18

4.4	Communications Interfaces	19
5	CHAPTER 5: Other Nonfunctional Requirements	19
5.1	Performance Requirements	19
5.2	Safety Requirements	19
5.3	Security Requirements	19
5.4	Software Quality Attributes	19
6	CHAPTER 6: Systems Models	20
6.1	Use Case Diagrams	20
6.1.1	User Use Case	20
6.1.2	Patient Use Case	20
6.1.3	Doctor Use Case	21
6.2	Sequence Diagrams	22
6.2.1	View Medical Specifications Sequence Diagram	22
6.2.2	View & Search Doctors Sequence Diagram	23
6.2.3	Registration Sequence Diagram	24
6.2.4	Login Sequence Diagram	25
6.2.5	Patient Send Request Sequence Diagram	26
6.2.6	view Request Sequence Diagram	27
6.2.7	View & Update Daily Checkup Sequence Diagram	28
6.2.8	View & Update Medical Information Sequence Diagram	29
6.2.9	View & Update Clinic Information Sequence Diagram	30
6.2.10	View & update Doctor About Information Sequence Diagram	31
6.2.11	Request Response Process Sequence Diagram	32
6.3	Class Diagrams	33
6.4	E-R DIGRAM	34

Table of figures

Figure	Description	Page
1	ER-Diagram of MedicoPortal website	11
2	Use Case Diagram for User	20
3	Use Case Diagram for Patient	20
4	Use Case Diagram for Doctor	21
5	View Medical Specifications Sequence Diagram	22
6	View & Search Doctors Sequence Diagram	23
7	Registration Sequence Diagram	24
8	Login Sequence Diagram	25
9	Patient Send Request Sequence Diagram	26
10	View Requests Sequence Diagram	27
11	View & Update Daily Checkup Sequence Diagram	28
12	View & Update Medical Information Sequence Diagram	29
13	View & Update Clinic Sequence	30
14	View & Update Doctor about Information Sequence Diagram	31
15	Request Response Process Sequence Diagram	32
16	Class Diagram	33
17	E-R Diagram	34

Software Requirements Specification

1. CHAPTER 1: Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline functional and non-functional requirements of the development of the MedicoPortal website and to serve as a guide for the development team to ensure that the product meets the perspective.

Our website aims to offer a versatile platform suitable for both medical clinics doctors and personal use, enabling users to access various medical services. These services include retrieving medical records, summarizing medical history, and securely storing it within a personal account. This setup facilitates healthcare professionals in receiving personalized health information through the platform, ensuring patients receive appropriate medical care.

The system's objectives are to enhance access to healthcare services, streamline communication between patients and healthcare providers, and ultimately improve the overall patient care experience.

1.2 Document Conventions

Acronym	Term	Definition
SRS	Software Requirements Specification	Comprehensive document that outlines the functional and non-functional requirements for a software system.
	Medico Portal	The name of the medical care software website.
ER	Entity Relationship model	The Entity-Relationship model is a conceptual framework used to describe the data and relationships within a database system.
DI	Drug interaction	Refers to the effects that occur when two or more drugs are administered together, influencing the efficacy or toxicity of one or more of the drugs involved.
HTML	Hypertext Markup Language	The standard markup language for documents designed to be displayed in a web browser.
CSS	Cascading Style Sheets	style sheet language used for specifying the presentation and styling of a document written in a markup language
RAM	Random-access memory	a form of electronic computer memory that can be read and changed in any order
HL7	Health Level Seven	Abbreviated to HL7, is a range of global standards for the transfer of clinical and administrative health data between applications with the aim to improve patient outcomes and health system performance.

FHIR	Fast Healthcare Interoperability Resources	A set of rules and specifications for exchanging electronic health care data.
RBAC	Role-based access control	Refers to the idea of assigning permissions to users based on their role within an organization.
	React	Is a free and open-source front-end JavaScript library for building user interfaces based on components.
API	Application Programming Interface	
iOS		is a mobile operating system developed by Apple Inc.
MacOS		Is an operating system developed and marketed by Apple since 2001.
SQL	Structured Query Language	Structured Query Language is a domain-specific language used to manage data, especially in a relational database management system.
ASP.NET		Is a server-side web-application framework designed for web development to produce dynamic web pages.
JSON		Is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays.
HTTP	The Hypertext Transfer Protocol	Is the foundation of the World Wide Web, and is used to load webpages using hypertext links.
HTTPS	Hypertext Transfer Protocol Secure	is an extension of the Hypertext Transfer Protocol
HIPAA	The Health Insurance Portability and Accountability Act	

1.3 *Intended Audience and Reading Suggestions*

This SRS document is beneficial for:

1.3.1 **Developer**

The document will serve the development team as a guide to make it easier to move from one development phase to another, ensuring that the final product meets the specified needs and requirements of the system.

1.3.2 **End users**

1.3.2.1 **Healthcare Professionals**

Guidelines on utilizing the website for patient management, accessing patients' medical records efficiently, and the ability to provide the patients with medical reports.

1.3.2.2 Patients

Instructions on how to use the website to consult with healthcare providers, and access personalized health information, with the ability to add a daily checkup.

1.3.2.3 General Users

Overview of the website, list of doctors and medical specifications, features and how it can improve access to healthcare services.

1.4 Project Scope

This website was built to help the doctor diagnose the patient more accurately, through the doctor's access to the patient's complete record of diseases, allergies, medications, previous medical reports, and personal information. Therefore:

1. It enables the patient to obtain an overview of his general medical condition and archive it.
2. It enables the doctor to review the patient's medical condition very accurately.
3. It increases the efficiency and accuracy of the diagnosis given by the doctor.
4. Preventing many possible mistakes, such as:
 - DI (Drug interaction).
 - Giving medication that is dangerous to his health condition.
 - Unintentional neglect for the patient's allergies or chronic diseases.

The purpose of the online medical care system is to archive patients' medical records and to create a convenient and easy-to-use application for patients to get the best medical care and doctors to get all the information to do their work in the best way. The system relies on a relational database that includes all medical information about the patient, whether personal or written by former doctors.

Our Medico Portal website has some features which make it special to use, such as:

1. Data Security and Privacy.
2. User-Friendly Interface.
3. Mobile Accessibility.
4. Continuous Improvement.
5. Collaboration with Healthcare Providers.

Above all, we hope to provide a comfortable user experience along with the best prices available.

1.5 References

- http://users.encs.concordia.ca/~eshihab/teaching/slides/srs_template_sep14.pdf
- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2 CHAPTER 2: Overall Description

2.1 Product Perspective

the healthcare system described is initially designed as a standalone product focused on managing patient health records and facilitating care within healthcare facilities. However, it is anticipated that in the future,

it will integrate with clinics management systems, laboratory information systems (LIS), and hospital management system (HMS). This integration will enhance interoperability, streamline workflows, and improve care coordination across different healthcare settings. The system's modular components, such as patient health records and user management, will remain central, while interoperability interfaces will be established to enable seamless data exchange with other healthcare IT systems. Continuous evolution and upgrades will ensure alignment with evolving healthcare needs and technological advancements, maximizing the system's value and impact on patient care delivery.

2.2 *Product Features*

The healthcare system offers the following major features:

2.2.1 Medical Reports Management:

Ability to store and manage medical reports and other relevant documentation authored by health care providers.

2.2.2 Allergy Information:

Capability to record and manage information about patient allergies to medications, foods, or other substances.

2.2.3 Chronic Disease Tracking:

Functionality to document and track details about chronic conditions or ongoing health issues experienced by patients.

2.2.4 Medication Management:

Ability to document and manage information about medications prescribed to patients.

2.2.5 Vital Signs Monitoring:

Feature for recording and monitoring patients' vital signs, such as blood sugar levels and blood pressure, during medical encounters.

2.2.6 User Profile Management:

Capability to maintain and update basic information about users, including name, age, gender, contact details, address, and medical information.

2.2.7 Doctor Directory:

Functionality to categorize and list doctors based on their medical specialization.

2.2.8 Patient-Doctor Relationship Management:

Ability to assign patients to doctors and manage patient-doctor relationships efficiently.

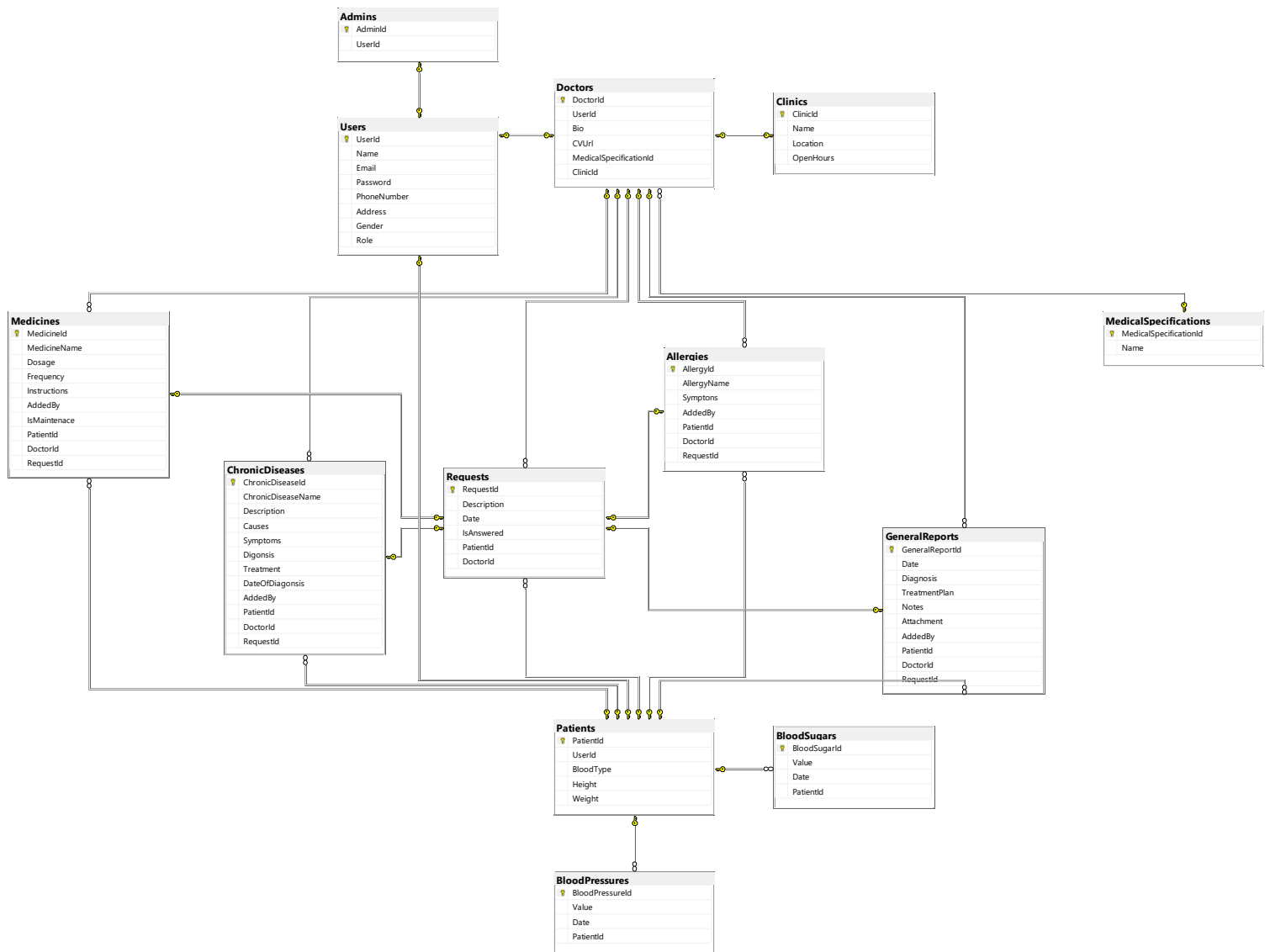


Figure 1: ER-Diagram of MedicoPortal website

2.3 User Classes and Characteristics

2.3.1 Patients:

Primary users accessing the system for viewing records, updating personal information and contact the doctors. Varied technical proficiency and healthcare literacy levels.

2.3.1 Healthcare Providers:

Use the system for accessing patient records, updating treatment plans, inputting medical data. Generally proficient with digital tools, with varying familiarity with the system.

2.3.2 System Administrators:

Responsible for maintaining, configuring, and securing the system. High technical proficiency and expertise in system administration.

2.3.3 Other Stakeholders:

IT support and others with varied technical expertise and responsibilities relevant to system operation and compliance.

2.4 *Operating Environment*

- The software is designed to run on standard hardware commonly found in healthcare facilities, including desktop computers, laptops.
- Minimum hardware specifications may include processors with adequate processing power, sufficient RAM for multitasking, and storage space for data storage and software installation.
- The product will be operating in any famous environment like: windows, Linux. computer hardware store is a website and shall operate in all famous browsers, The only requirement to use this online product would be the internet connection. The basic input devices required are keyboard, mouse and output devices are monitor etc.

2.5 *Design and Implementation Constraints*

2.5.1 User Interface (UI) and User Experience (UX):

- Accessibility: The system must be accessible to users with disabilities.
- Intuitive Design: The interface must be user-friendly, allowing both patients and doctors to navigate easily without extensive training.
- Responsiveness: The design should be responsive to work across various devices (desktops, tablets, smartphones).

We were able to accomplish these constraints in our system.

2.5.2 Interoperability:

- Standards Compliance: Must comply with healthcare data standards such as HL7 and FHIR for data exchange with other medical systems.
- Integration: Should support integration with other healthcare providers' systems, pharmacies, and insurance companies.

We were not able to accomplish these constraints in our system.

2.5.3 Scalability:

- Modular Architecture: The system should be designed to easily add new features and scale with an increasing number of users and data volume.
- Cloud Compatibility: Support for deployment on cloud platforms to leverage scalable infrastructure.

We were not able to accomplish these constraints in our system.

2.5.4 Data Security and Privacy:

- Encryption: patient data must be encrypted in transit and at rest using strong encryption standards.
- Access Control: Implement robust role-based access control (RBAC) to ensure that only authorized personnel can access sensitive information.

We were able to accomplish these constraints in our system.

2.5.1 Performance:

- **Low Latency:** The system must ensure low latency in data retrieval and processing to provide a seamless experience.
- **High Availability:** Design for high availability to ensure the system is operational 24/7 with minimal downtime.

We were able to accomplish these constraints partially in our system.

2.6 Assumptions and Dependencies

Assumptions:

1. User Adoption:

- **Patients and Doctors:** It is assumed that both patients and doctors will be willing to adopt and regularly use the system.
- **Technical Proficiency:** Users have basic technical proficiency to interact with the system without extensive training.

2. Data Accuracy:

- **Historical Data:** The data inputted by patients and historical medical records provided by previous doctors are accurate and complete.
- **Ongoing Updates:** Doctors will accurately update patient records after each consultation.

3. Technical Infrastructure:

- **Internet Access:** Users have reliable internet access to use the online system.
- **Hardware Requirements:** Users have access to necessary hardware (computers, tablets, smartphones) that meets the minimum system requirements.

4. Security:

- **Data Protection:** The system's security measures will be sufficient to prevent unauthorized access and data breaches.
- **Compliance:** The encryption and security protocols will meet industry standards and regulatory requirements.

Dependencies:

1. Technical Dependencies:

- **Software and Tools:** The system relies on specific software tools and frameworks for development (e.g., React).
- **APIs:** External APIs for integration with other healthcare systems, pharmacies, and insurance companies must be reliable and accessible.
- **Cloud Services:** Dependence on cloud service providers for hosting, scalability, and data storage.

2. Regulatory Bodies:

- **Approval and Certification:** The system must be approved and certified by relevant healthcare regulatory bodies before it can be deployed.
- **Ongoing Compliance:** Continuous updates and modifications may be required to maintain compliance with evolving regulations.

3. Data Sources:

- Medical Records: Access to existing medical records from previous healthcare providers.
- Patient Input: Accurate input from patients regarding their medical history, allergies, and medications.

4. Training and Support:

- User Training: Effective training programs for users (patients and doctors) to ensure they can use the system efficiently.
- Technical Support: Ongoing technical support to address issues and provide assistance.

3 CHAPTER 3: System Features

3.1 *Patient-Doctor Relationship Management*

3.1.1 Description and Priority:

This feature enables collaborative health record management between doctors and patients, allowing doctors to access and update patient health records based on patient requests after they have undergone a medical examination. It is of High priority as it ensures that patient health records are accurate and up-to-date.

3.1.2 Stimulus/Response Sequences:

- Stimulus: Patients sends a request to the doctor to update medical information such as reports, chronic diseases, allergies, and vital signs after they have undergone a medical examination.
- Response: The doctor receives the request and accesses the patient's profile containing his medical records.
- Subsequent Stimulus: Doctor reviews the requests and adds or updates medical information as necessary.
- Subsequent Response: The system confirms the successful addition or update of medical information.

3.1.3 Functional Requirements:

- Enable patients to send requests to their doctors to update their health information, including medical reports, chronic diseases, allergies.
- Allow doctors to securely access and update patients' health records in response to patient requests.
- Provide an intuitive and user-friendly interface for doctors to review patient requests and interact with health records collaboratively.
- Enable patients to review the updated health information after it has been added or updated by the doctor.

3.2 *Doctor's Directory*

3.2.1 Description and Priority:

This feature provides a comprehensive directory of doctors categorized based on their medical specialization, allowing patients to search and choose a doctor that matches their needs. It is of High priority as it plays a crucial role in facilitating patient access to appropriate healthcare providers within the healthcare system.

3.2.2 Stimulus/Response Sequences:

- Stimulus: User accesses the doctor's directory and selects a medical specialization.
 - Response: System displays a list of doctors within the selected medical specialization.
- Stimulus: User selects a doctor from the list.
 - Response: System displays the profile of the selected doctor, including qualifications, availability, contact details, clinic details and the ability to send a request allowing the doctor to view patient medical profile.

3.2.3 Functional Requirements:

- Maintain a database of doctors with information on their medical specialization, qualifications, and contact details.
- Categorize doctors based on their medical specialization for easy navigation and selection by patients.
- Display detailed profiles of doctors, including their clinic affiliations.
- Allow users to access a complete list of doctors without applying any filters for browsing purposes.
- Implement search functionality to enable users to search for a specific doctor by name, medical specialization.

3.3 *Patient Health Record*

3.3.1 Description and Priority:

This feature involves managing various aspects of a patient's medical history, including medical reports, allergies, chronic diseases, medicines, vital signs, and personal information. It is of High priority as it serves as a comprehensive repository for patient health information, essential for providing quality healthcare services.

3.3.2 Stimulus/Response Sequences:

Patient Actions:

- Stimulus: Patient selects the option to view and update general data (e.g., Blood type, height, weight).
 - Response: System displays a form for update general data and confirms successful submission after validation.

- Stimulus: Patient selects the option to view allergies.
 - Response: System displays a list of the patient's recorded allergies.
- Stimulus: Patient selects the option to view chronic diseases.
 - Response: System displays a list of the patient's chronic diseases.
- Stimulus: Patient selects the option to view medical reports authored by doctors.
 - Response: System displays the list of reports with options to view each report in detail.
- Stimulus: Patient selects the option to add a daily check-up record (e.g., blood sugar level, blood pressure).
 - Response: System displays a form for entering daily check-up data and confirms successful submission after validation.
- Stimulus: Patient selects the option to view daily check-up history (e.g., blood sugar level, blood pressure).
 - Response: System displays a list of daily check-up data (value and date).

Doctor Actions:

- Stimulus: Doctor views a request with the option to access patient medical profile.
 - Response: System displays patient profile containing his medical history.
- Stimulus: The doctor chooses to reply to an unanswered request.
 - Response: System displays a form to add medical information such as reports, chronic diseases, allergies, medicines with detailed fields.
 - Subsequent Response: The system confirms the successful addition or update of medical.
 - Subsequent Stimulus: The doctor chooses to view a summary of his response to the patient.
 - Subsequent Response: The system displays a summary of the specified requests.

3.3.3 Functional Requirements:

Patient Functional Requirements:

- The system provides an interface for patients to view their recorded allergies.
- The system allows patients to view their chronic diseases.
- The system enables patients to view medical reports authored by doctors.
- The system allows patients to view their medicines.
- The system displays a form for patients to enter daily check-up data such as blood sugar levels and blood pressure.
- The system displays a list of daily check-up history data such as blood sugar levels and blood pressure.

Doctor Functional Requirements:

- The system allows doctors to view a patient's recorded allergies and provide options to edit or add new allergy information.
- The system enables doctors to view a patient's chronic diseases and provide options to edit or add new chronic disease information.

- The system enables doctors to view a patient's medicines and provide options to edit or add new medicines information.
- The system provides an interface for doctors to view daily check-up records submitted by the patient.
- The system provides a form for doctors to add general medical reports to the patient's health record.

3.4 Account Management

3.4.1 Description and Priority:

The Account Management feature is essential for enabling users to register, log in, and log out of the healthcare system. During registration, users can sign up as either a patient or a doctor, which determines their subsequent interactions and access within the system. This feature is crucial for establishing user identities and controlling access within the system. Due to its critical role in system security and user access, this feature has a high priority.

3.4.2 Stimulus/Response Sequences:

- Stimulus: User selects "Register" from the main page.
 - Response: System displays an option for the user to choose whether to register as a doctor or a patient.
- Stimulus: User selects either "Register as Doctor" or "Register as Patient".
 - Response: System displays the corresponding registration form.
- Stimulus: User fills out the registration form with personal details and.
 - Response: System validates input, creates a new account, and stores user information.
- Stimulus: User submits the registration form.
 - Response: System confirms successful registration and redirects user to the appropriate dashboard (patient or doctor).

3.4.3 Functional Requirements:

- The system shall provide a logout option accessible from the user's dashboard.
- The system shall terminate the user session upon logout.
- The system shall redirect the user to the main page after logout.

4 CHAPTER 4: External Interface Requirements

4.1 User Interfaces:

The user interfaces for the healthcare system will be developed using modern web technologies to ensure compatibility across different devices and browsers. The front-end software will utilize HTML, CSS, JavaScript and React to create interactive and responsive user interfaces. Here's a breakdown of the logical characteristics:

- Layout: The layout will be designed to be intuitive and user-friendly, with clear navigation menus and organized content sections.
- Visual Design: The interfaces will follow a clean and professional design aesthetic, with visually appealing elements and appropriate use of colors and typography.

- **Interactivity:** Users will be able to interact with various elements on the interface, such as buttons, forms, and dropdown menus, to perform tasks such as searching for doctors, viewing medical records, updating information's.
- **Error Handling:** Informative error messages will be displayed to users in case of incorrect inputs or system errors, guiding them on how to proceed.

4.2 Hardware Interfaces

The healthcare system will be compatible with standard hardware components commonly used by users, including:

Devices	Any device with a web browser, such as desktop computers, laptops, tablets, and smartphones, can access the system.
Operating Systems	The system will be compatible with various operating systems, including Windows, macOS, iOS, and Android.

4.3 Software Interfaces

The healthcare system will integrate with various software components to support its functionality:

Device	Any device with a web browser, such as desktop computers, laptops, tablets, and smartphones, can access the system.
Operating Systems	The system will be compatible with various operating systems, including Windows, macOS, iOS, and Android
Database	Patient data, doctor information, and other system records will be stored and managed using a relational database management system (e.g., SQL Server).
Web Browser	Users will require a modern web browser that supports HTML, CSS, JavaScript and React to access the system's interfaces.
ASP.NET	To implement the backend, we have chosen ASP.NET framework for its interactive support and ease of use.
HTML5, CSS3, and JavaScript	To implement the frontend, these technologies are used for structuring and presenting content on the web, designing the layout, and providing interactivity and dynamic content updates on the client side.
React.js	A JavaScript library used for building user interfaces, particularly single-page applications. It allows for the creation of reusable UI components.
Bootstrap	A CSS framework directed at responsive and mobile-first web development, ensuring that the web application is visually appealing and works well on different devices.
JWT (JSON Web Tokens)	Used for securely transmitting information between parties as a JSON object. JWT is used for user authentication and session management.

4.4 Communications Interfaces

Communication within the healthcare system will primarily occur over standard web protocols and technologies:

Web Protocols	The system will utilize HTTP and HTTPS protocols for communication between clients and the web server, ensuring secure data transmission over the internet.
Web Browsers	The system will be accessible through any modern web browser, allowing users to access its interfaces from a wide range of devices and platforms.

5 CHAPTER 5: Other Nonfunctional Requirements

5.1 Performance Requirements

- Response Time: The system should respond to user interactions within 3 second under normal load conditions to ensure a smooth and responsive user experience.
- Scalability: The system should be capable of handling a large number of concurrent users without significant degradation in performance. It should scale horizontally by adding more server resources as the user base grows.
- Database Performance: Database queries should execute within 1 second to retrieve patient records, doctor information, and other data needed for system operations.

5.2 Safety Requirements

- Data Integrity: The system must ensure the integrity of patient health records and other sensitive information to prevent unauthorized access, modification, or deletion.
- Compliance with Healthcare Regulations: The system should comply with relevant healthcare regulations such as HIPAA to safeguard patient privacy and confidentiality.

5.3 Security Requirements

- User Authentication: Users should be required to authenticate themselves using secure credentials (e.g., username and password) before accessing the system. Multi-factor authentication may be implemented for added security.
- Encryption: Patient health records and other sensitive data should be encrypted both in transit and at rest to protect against unauthorized access.

5.4 Software Quality Attributes

- Usability: Our healthcare system prioritizes user experience by providing intuitive friendly interfaces and straightforward navigation.
- Reliability: Our healthcare system is built to be dependable and stable, ensuring continuous availability of healthcare services with minimal downtime.
- Maintainability: The architecture of our healthcare system is designed for ease of maintenance, featuring modular components and well-documented code.

6 CHAPTER 6: Systems Models

6.1 Use Case Diagrams

6.1.1 User Use Case

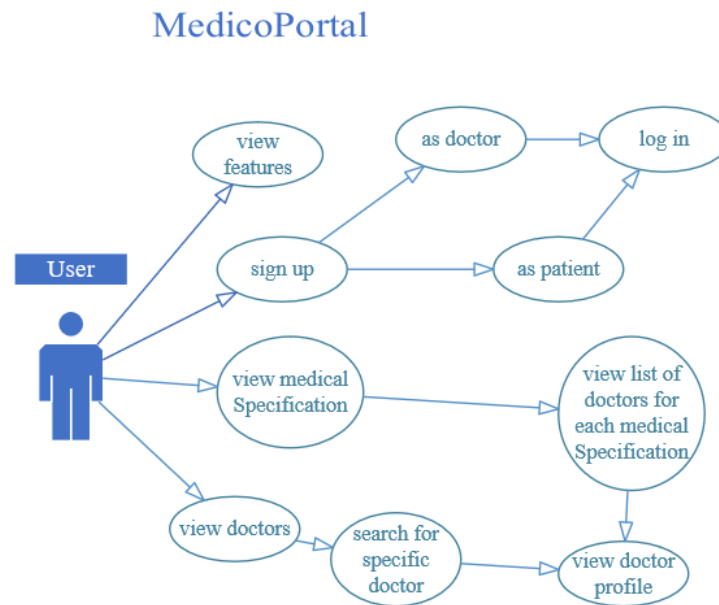


Figure 2: Use Case Diagram for User

6.1.2 Patient Use Case

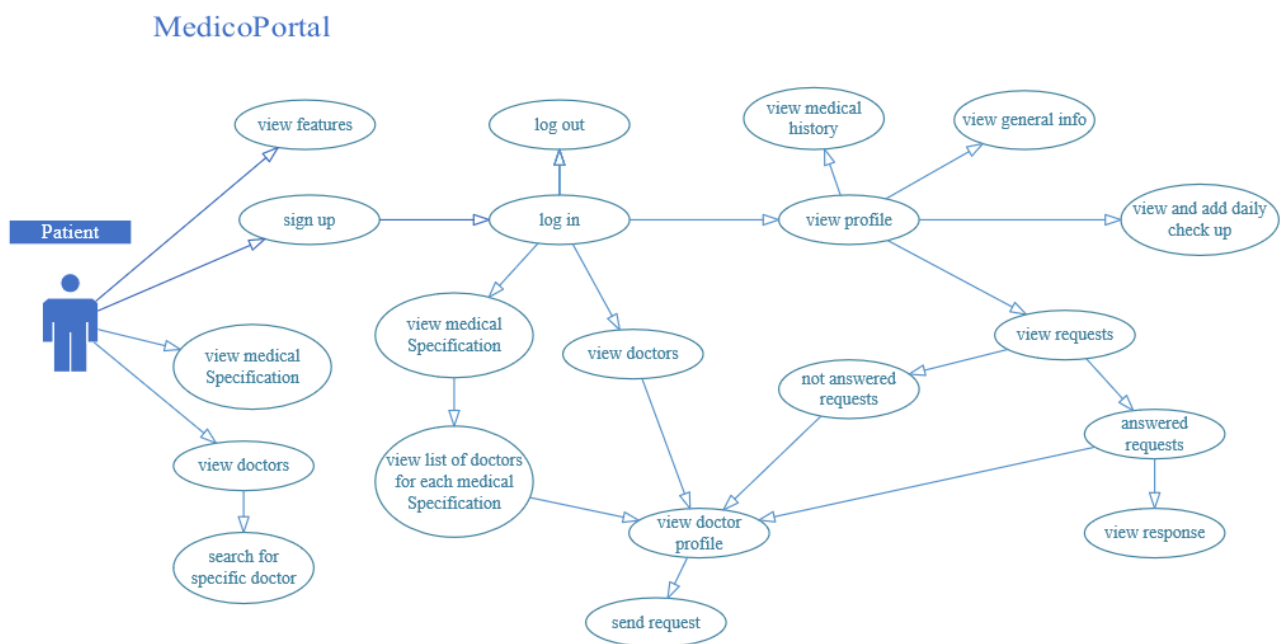


Figure 3: Use Case Diagram for Patient

6.1.3 Doctor Use Case

MedicoPortal

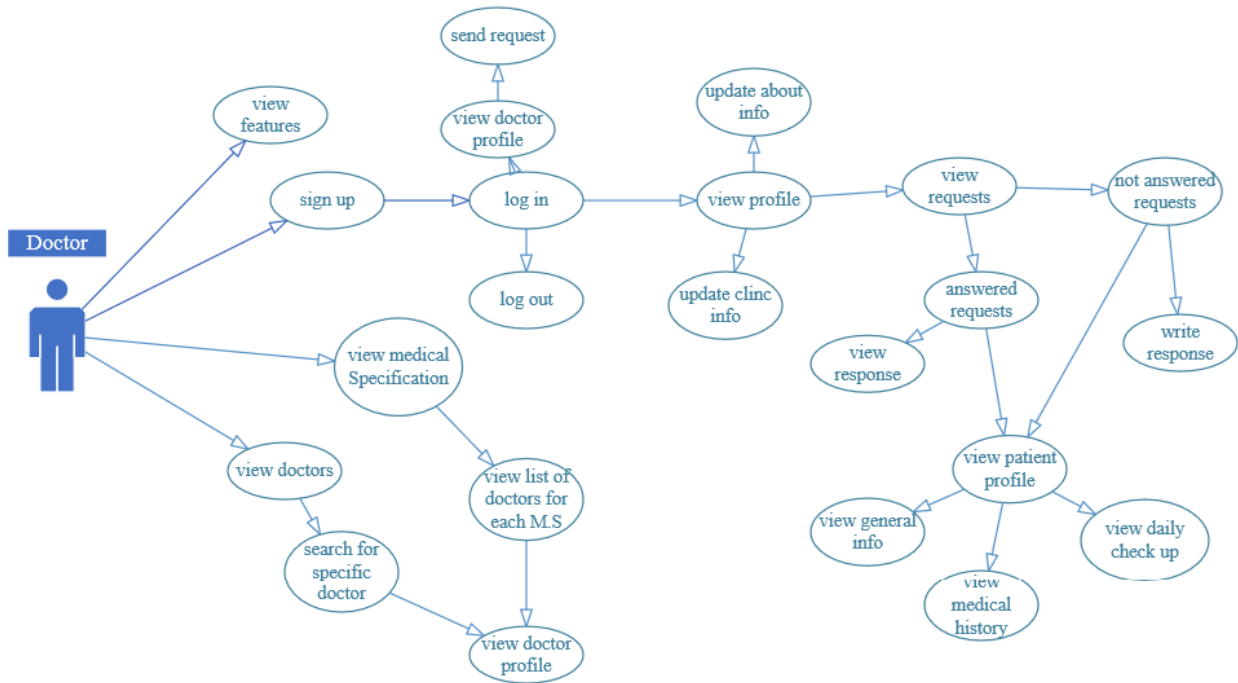


Figure 4: Use Case Diagram for Doctor

6.2 Sequence Diagrams

6.2.1 View Medical Specifications Sequence Diagram

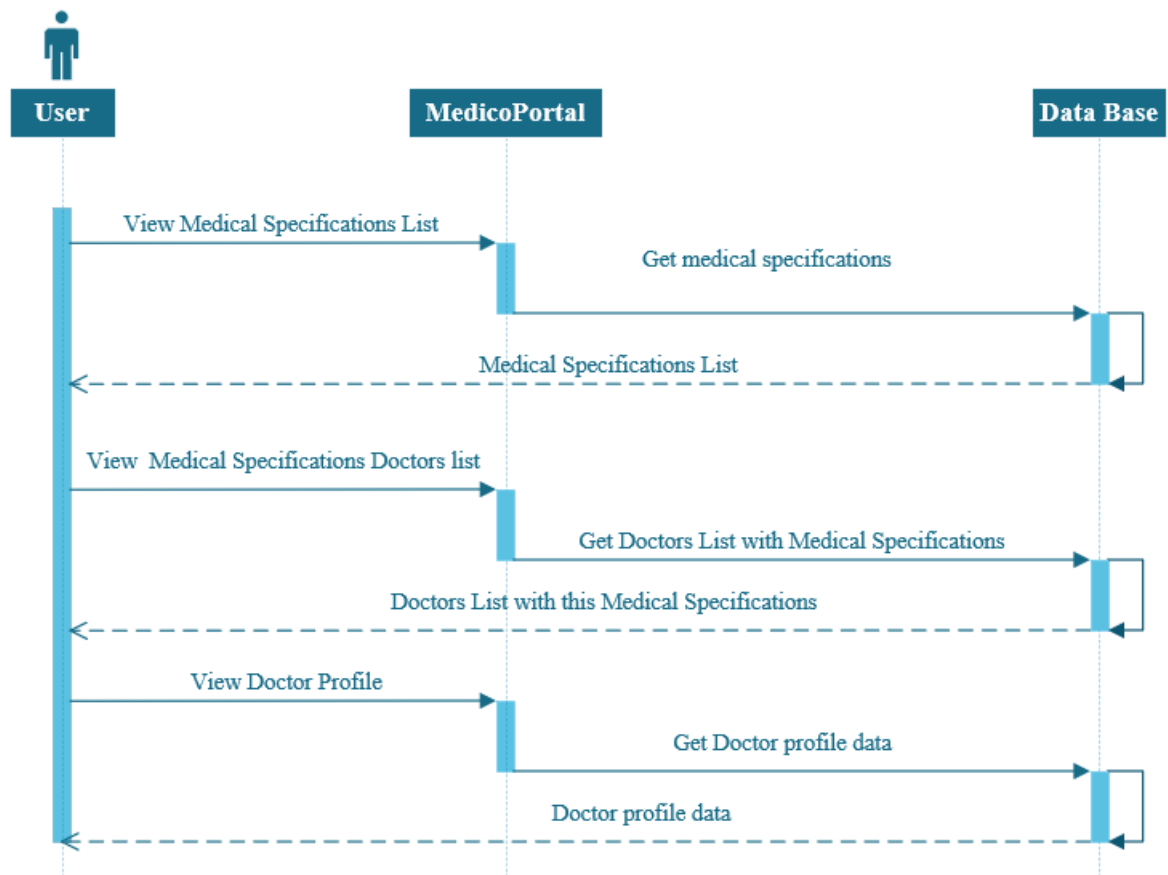


Figure 5: View Medical Specifications Sequence Diagram

6.2.2 View & Search Doctors Sequence Diagram

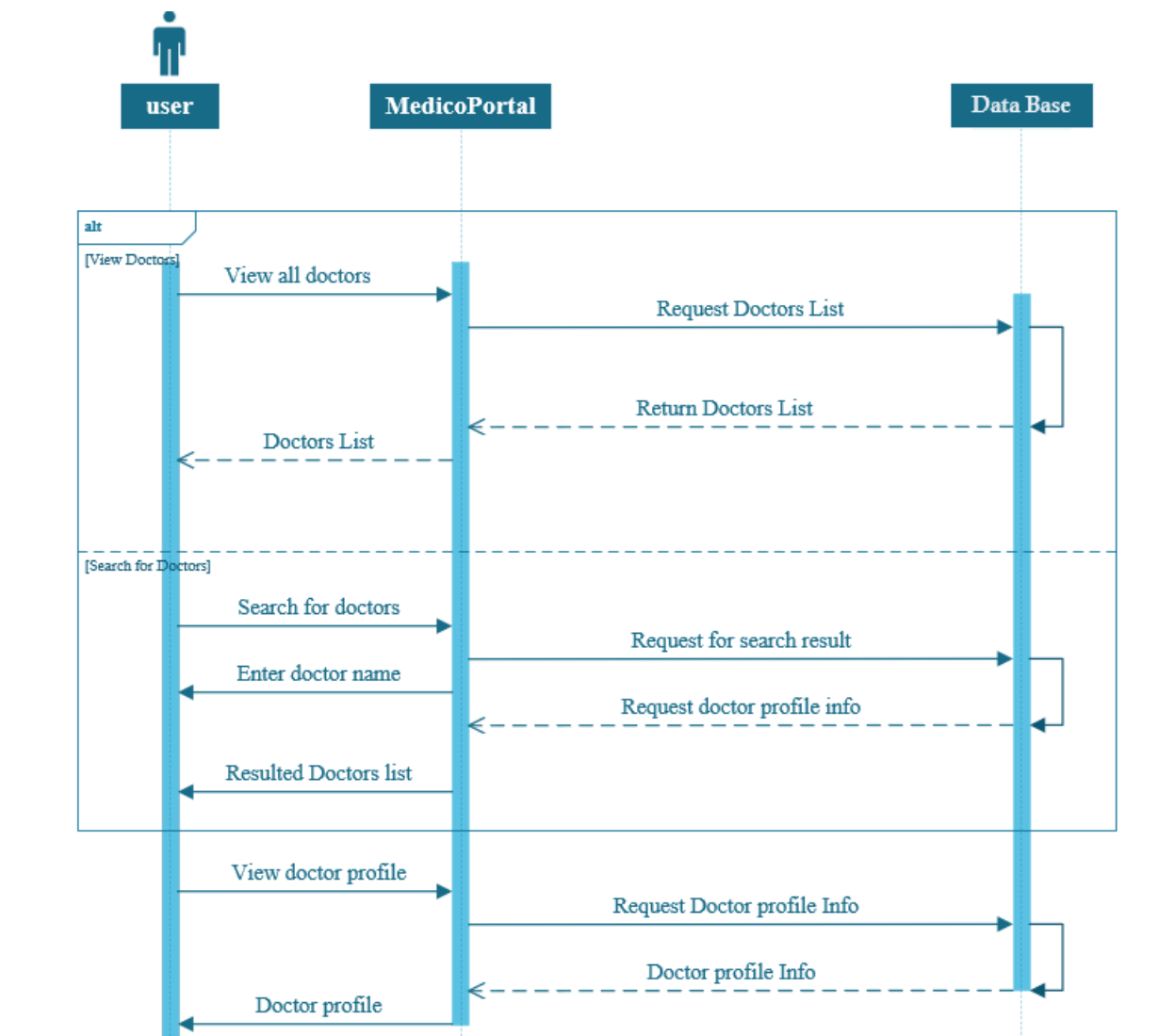


Figure 6: View & Search Doctors Sequence Diagram

6.2.3 Registration Sequence Diagram

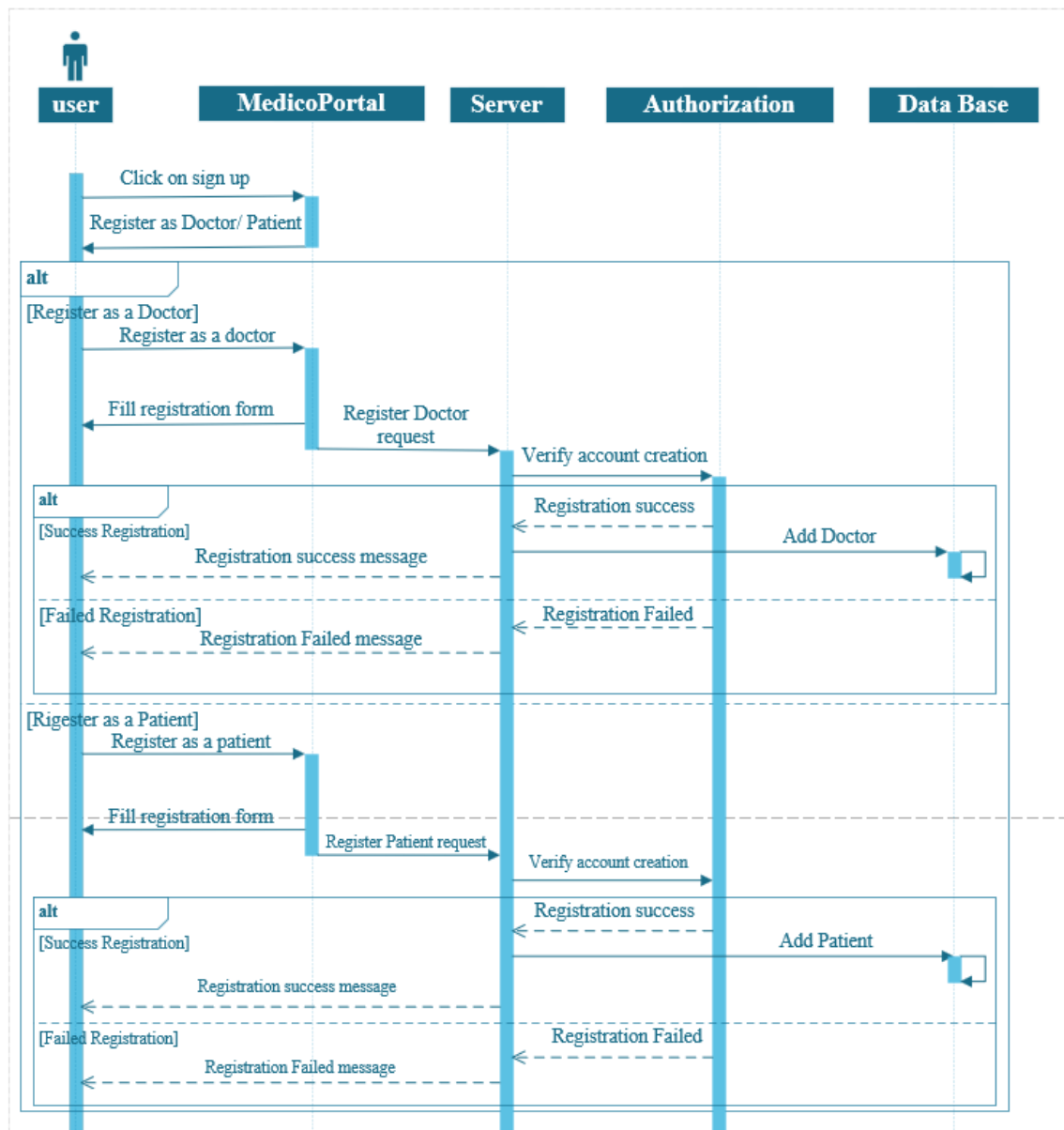


Figure 7: Registration Sequence Diagram

6.2.4 Login Sequence Diagram

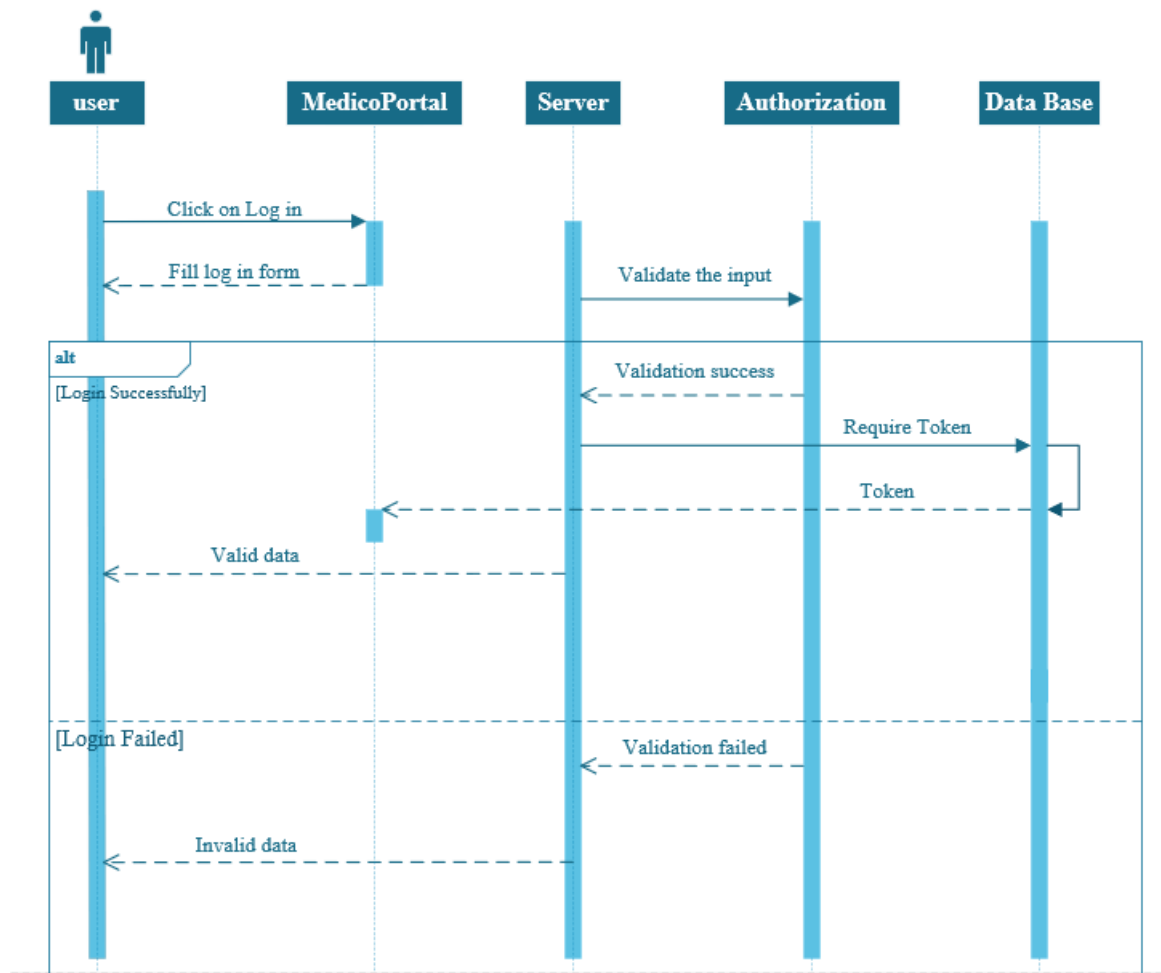


Figure 8: Login Sequence Diagram

6.2.5 Patient Send Request Sequence Diagram

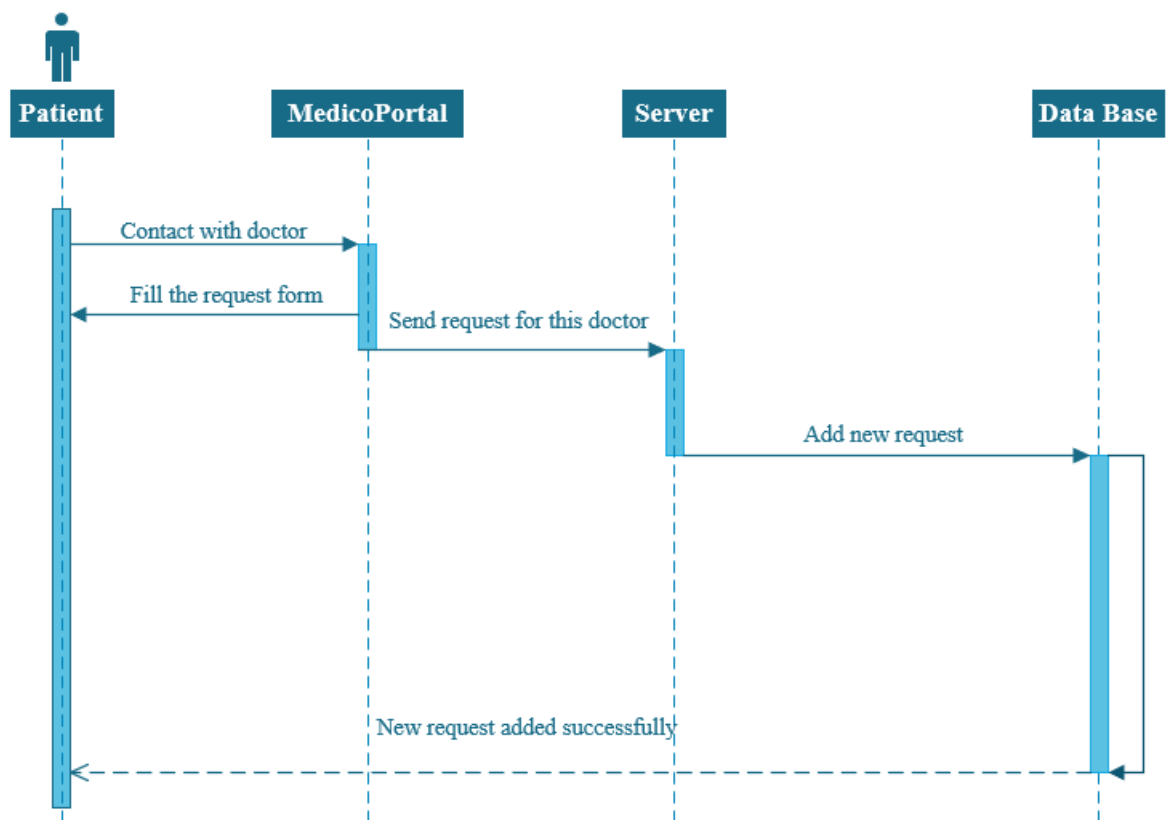


Figure 9: Patient Send Request Sequence Diagram

6.2.6view Request Sequence Diagram

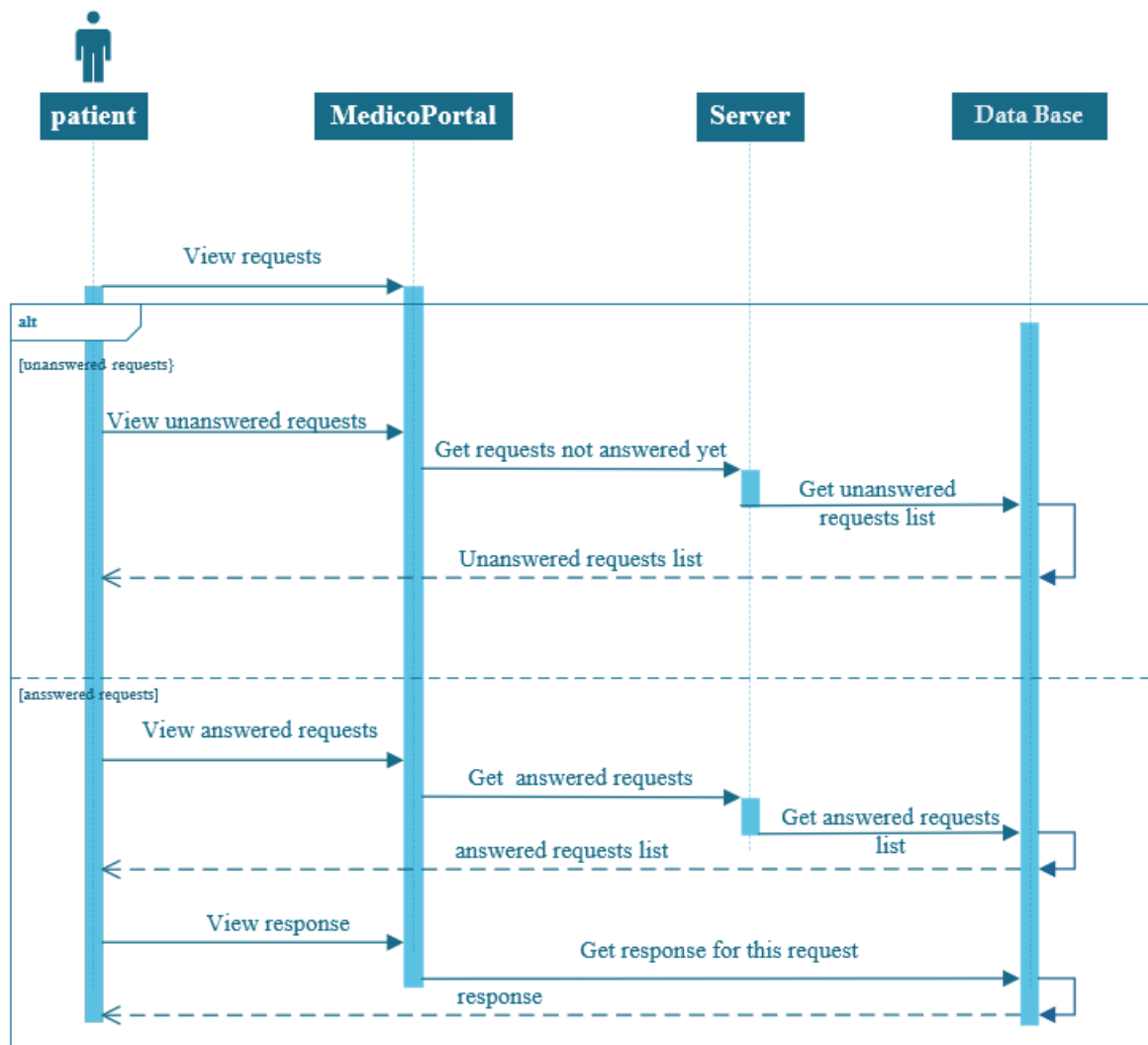


Figure 10: View Requests Sequence Diagram

6.2.7View & Update Daily Checkup Sequence Diagram

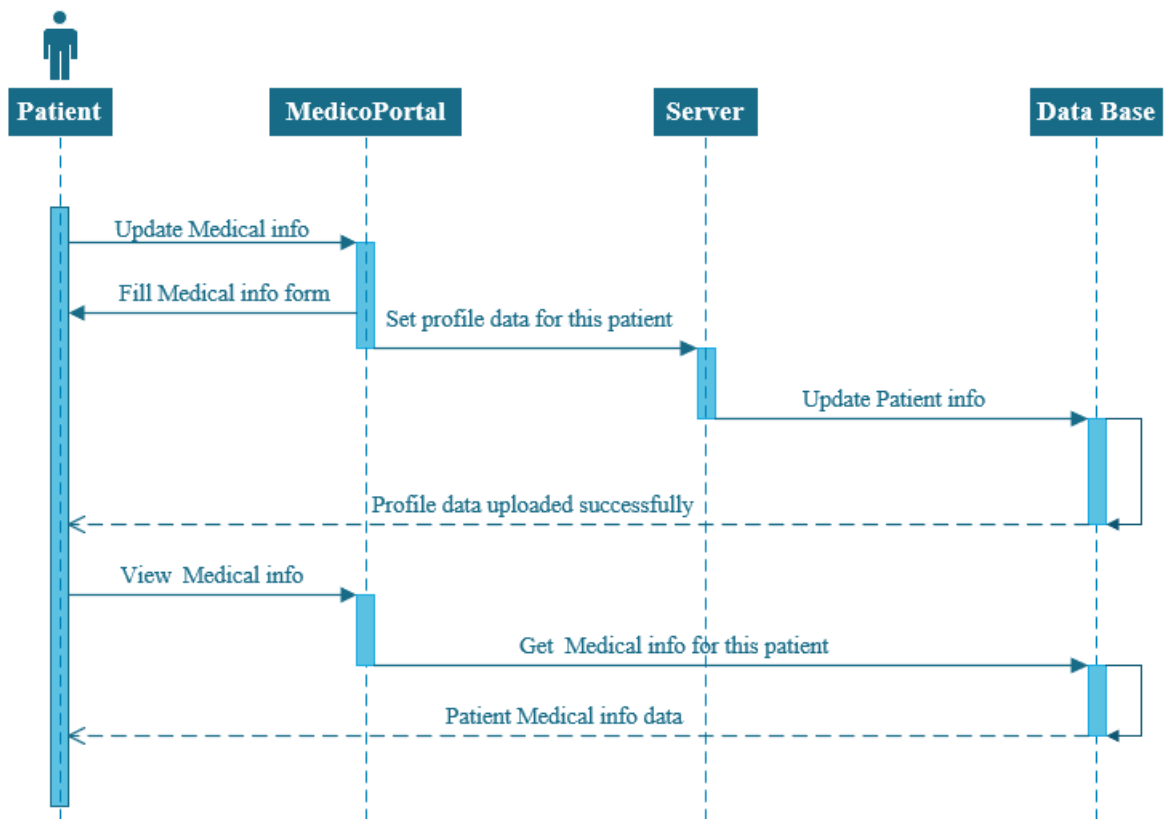


Figure 11: View & Update Daily Checkup Sequence Diagram

6.2.8 View & Update Medical Information Sequence Diagram

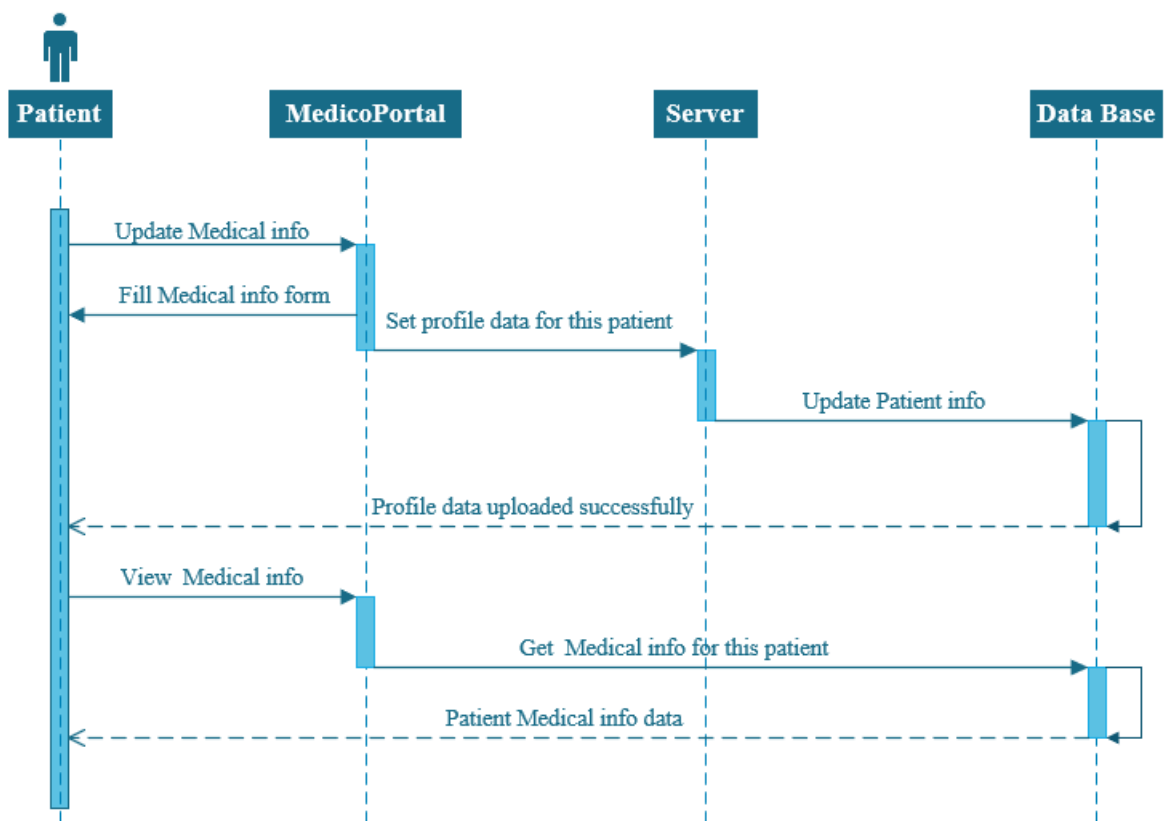


Figure 12: View & Update Medical Information Sequence Diagram

6.2.9 View & Update Clinic Information Sequence Diagram

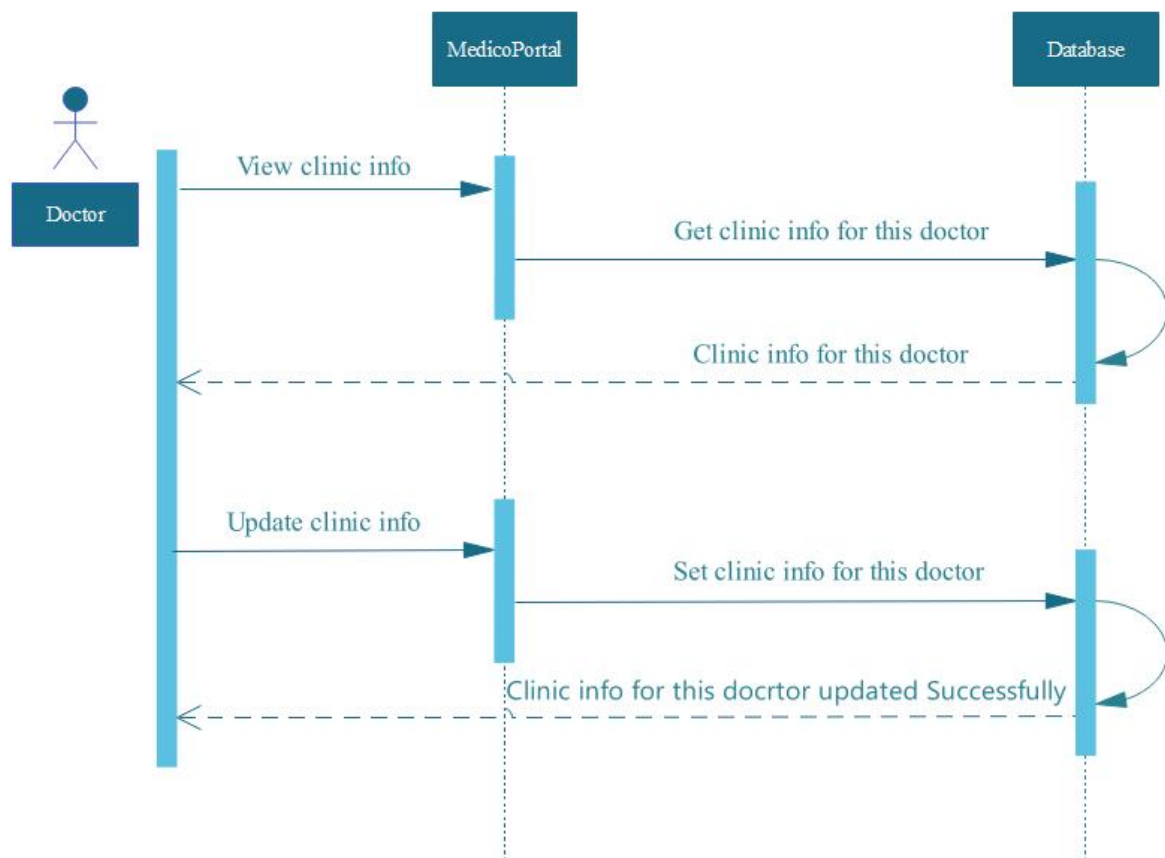


Figure 13: View & Update Clinic Sequence

6.2.10 View & update Doctor About Information Sequence Diagram

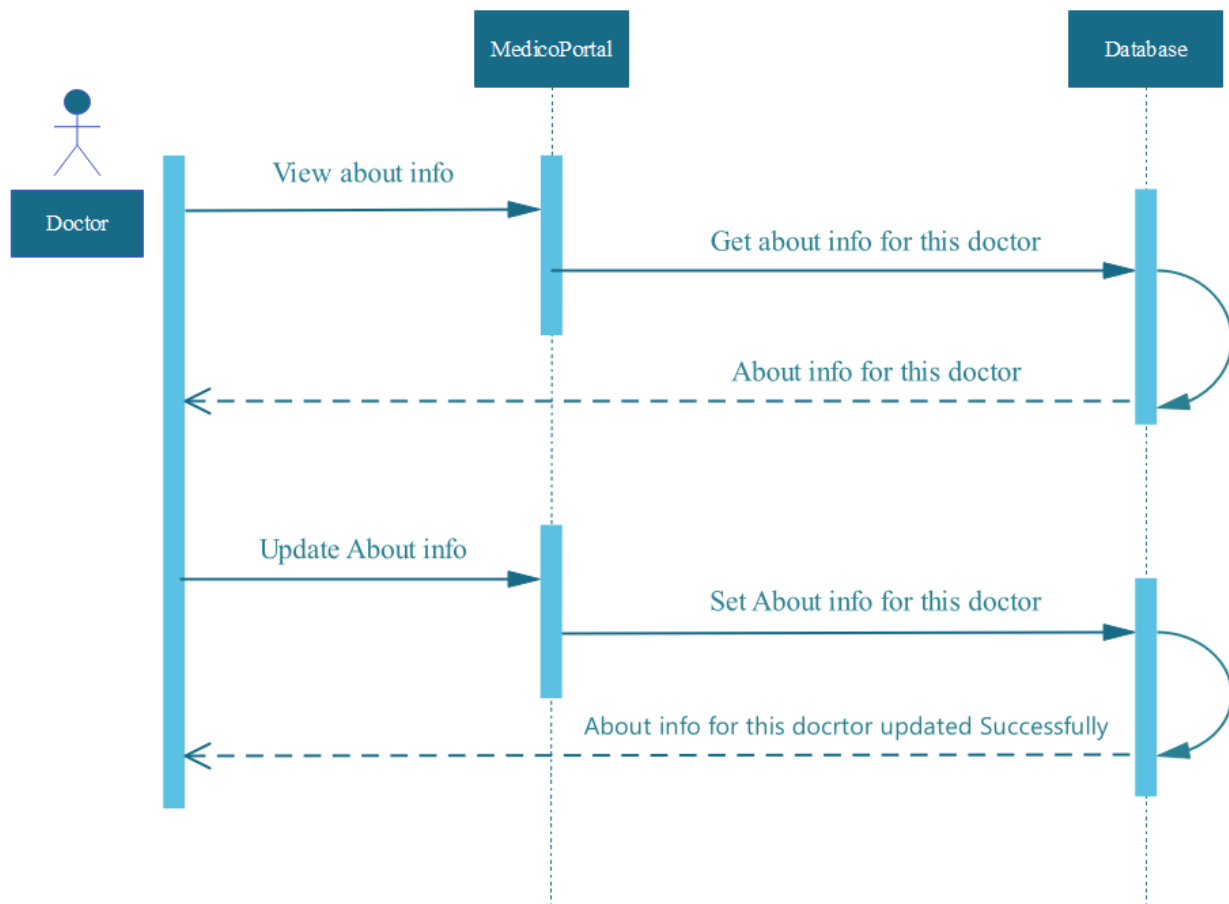


Figure 14: View & Update Doctor about Information Sequence Diagram

6.2.11 Request Response Process Sequence Diagram

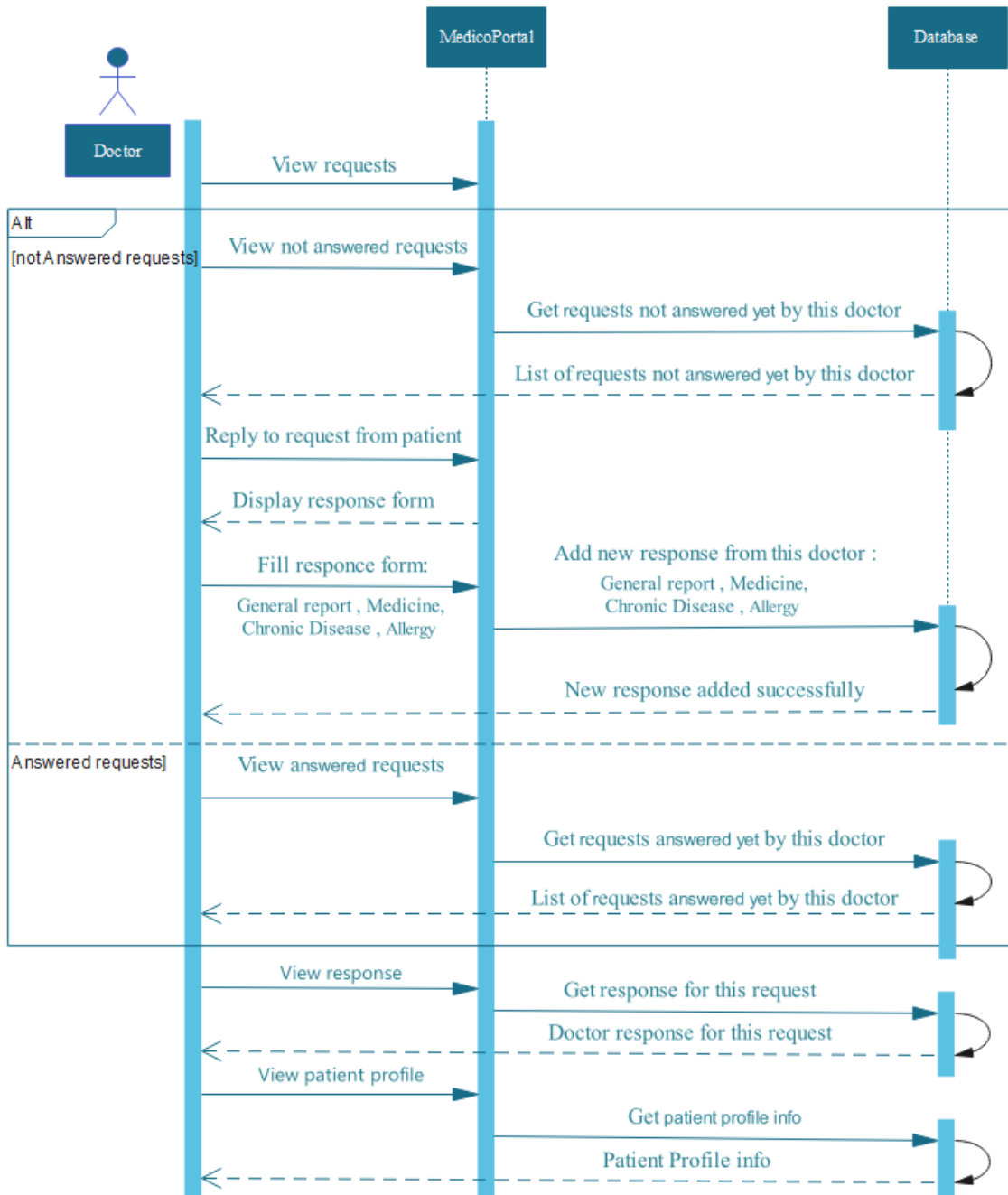


Figure 15: Request Response Process Sequence Diagram

6.3 Class Diagrams

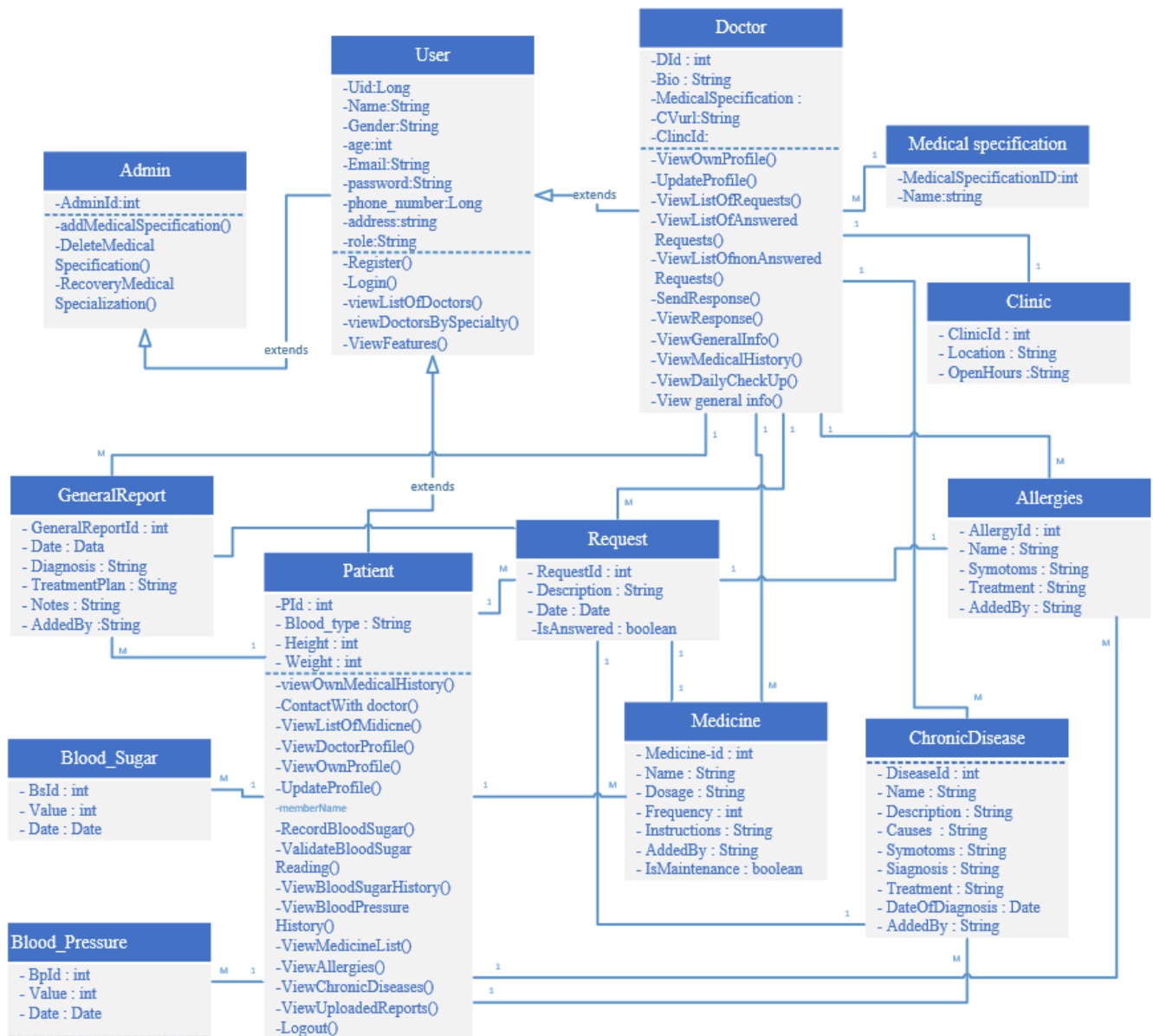


Figure 16: Class Diagram

6.4 E-R Diagram

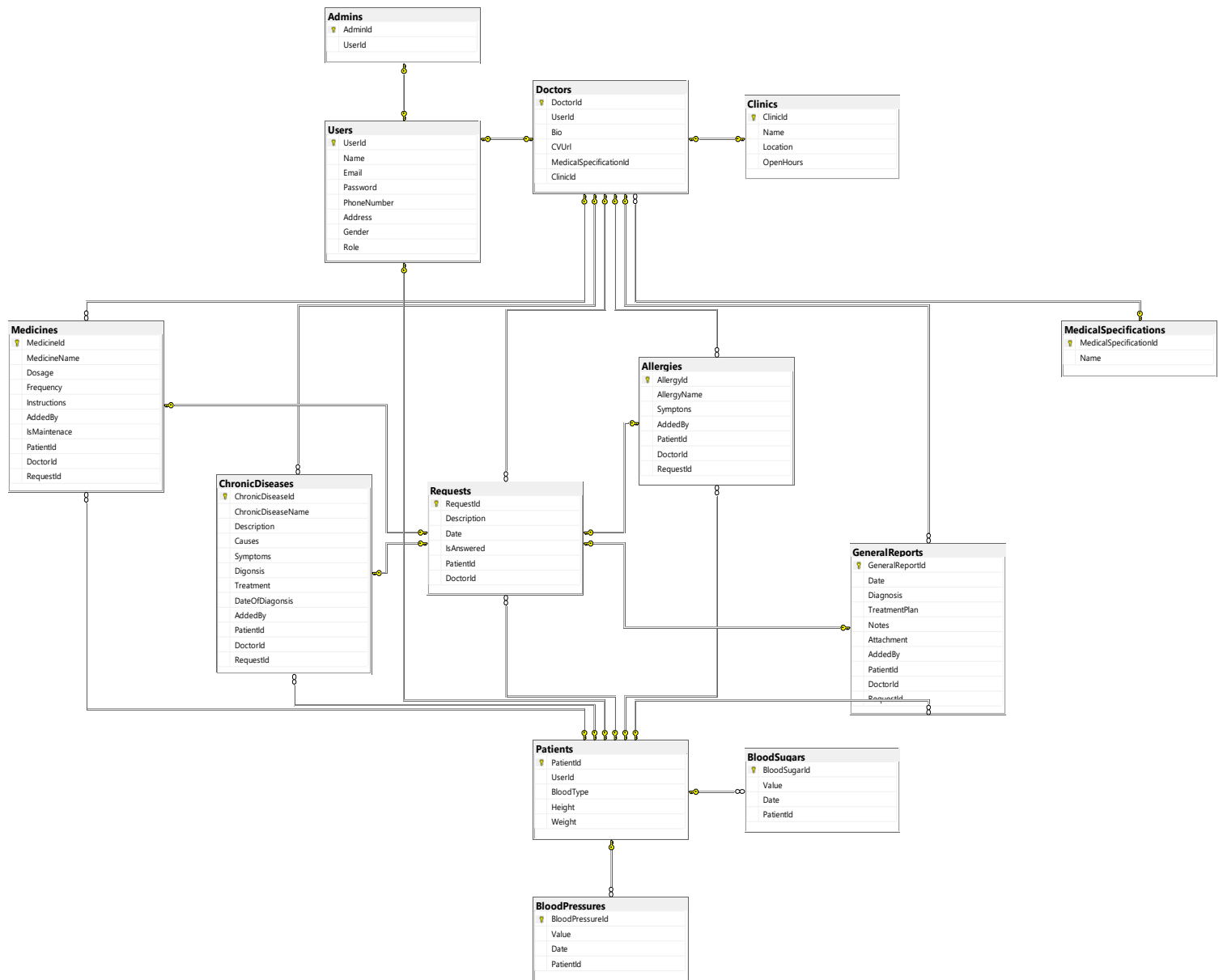


Figure 17: E-R Diagram