## engineering College – CoMputer sCIENCE DEPARTMENT

|  |  |  |
| --- | --- | --- |
| **INSTRUCTORS NAME: Dr. Akila Sarirete** | | |
| **COURSE NAME: Software Engineering** | **COURSE NUMBER: CS3151** | **SECTION: 1, 2** |

Software Requirements Specification (SRS) Guidelines

**Group Members:** *Leen Sharab – Sarah Alshumayri – Lujain Almarri – Reema Abdullah – Ameera Attiah*

# 1. Introduction

## 1.1 Purpose

* **Define the main objectives and purposes of this document.**

The main objectives and purposes of this document are to gather the stakeholders' and the system requirements in order to deliver and implement an efficient and complete software.

* **Identify the intended audience.**

1. **Patients**
2. **Physicians**

## 1.2 Scope

* **Clearly delineate the boundaries of the software application.**

The boundaries of our software are patient record system, admission system, management system, medical imaging system, pharmacy system, radiology information system, laboratory information system, and doctor record system.

* **Describe the software's main capabilities.**

Our software’s main capabilities are inserting personal information (name, dob, address, etc.), view own medical information, verify the information inserted by the doctor, view doctor's information (name, specialty, phone number, email), log in to the patient's account with a special id, edit the medical information of the patient, insert required medications for the patient, upload documents of medical reports and imaging information, provide authentication and verification of patient accounts, ensure system security, store all patient record information, including patient information, medications, laboratory, system radiology/non-radiology info, maintain a record of the last edited name (doctor's name) when changes are made to the records.

## 1.3 Definitions, Acronyms, and Abbreviations

* **Define technical and industry-specific terms for clarity.**
* **List acronyms and abbreviations with their expanded forms.**

|  |  |
| --- | --- |
| Definitions |  |
|  | ***Medical Memory Software:*** The software system designed for storing, managing, and providing access to patient medical records and related information. It includes features for personal information management, medication tracking, medical report storage, and more. |
|  | ***Patient Information:*** Data related to an individual's health and medical history, such as name, date of birth (DOB), address, and contact details. |
| Acronyms |  |
|  | CT Scan - Computed Tomography Scan |
|  | BMI – Body Mass Index |
|  | MRI - Magnetic Resonance Imaging |
|  | BCG - Bacillus Calmette-Guérin |
|  | DTP: Diphtheria - Tetanus - Pertussis vaccine |
|  | DOB - Date of Birth |
|  | PHR - Personal Health Record |
| Abbreviations |  |
|  | Rx - Prescription |
|  | Lab - Laboratory |
|  | Dr – Doctor (Physician) |
|  | Mg – milligram |

## 1.4 References

* **Cite external documents or sources referenced in the SRS.**
* **Provide version and publication details where relevant.**

1. Software Engineering, 10th edition, ISBN 978-0-13-394303-0, by Ian Sommerville, published by Pearson Education © 2016.
2. <http://chat.openai.com>
3. <https://asana.com/resources/software-requirement-document-template>
4. <https://www.apple.com/sa/ios/health/>

## 1.5 Overview

* **Summarize the content structure of the SRS.**

The Software Requirements Specification (SRS) serves as the cornerstone of software development projects, following a structured format. It begins with an introduction that outlines the project's scope and purpose and includes definitions and abbreviations. The overall description section offers a comprehensive view, addressing product perspective, functions, user characteristics, and constraints. Specific requirements delve into intricate details, covering external interfaces, functional and performance criteria, and design constraints. System features are detailed with input-output specifics and processing logic. External interface requirements include user interfaces, hardware and software connections, and communication protocols. Non-functional requirements define aspects like performance, safety, security, and quality attributes. Additional specifications and legal considerations are addressed in the 'Other Requirements' section. Supplementary materials are housed in the appendices, ensuring that the SRS effectively conveys the software's complexities to all stakeholders, fostering a shared understanding among developers, testers, and project managers.

# 2. General Description

## 2.1 Product Perspective

* **Explain how the software fits into the broader system or environment.**

In the context of medical memory software, the broader system typically includes the healthcare ecosystem in which the software operates. This ecosystem involves various stakeholders and systems, including healthcare providers, patients' information record, Admission System, and other healthcare facilities. The software acts as a crucial component within this ecosystem, providing a platform for managing and accessing patient medical records and information securely.

1. **Integration with Healthcare Providers:** The software serves as a bridge between patients and healthcare providers. It allows healthcare professionals (physicians, nurses, etc.) to access and update patient records, prescribe medications, and upload medical reports.
2. **Patient-Centric Approach:** The software is patient-centered, enabling individuals to access and manage their own medical information. Patients can view their medical records, verify information provided by doctors, and maintain their medication history.
3. **Regulatory Compliance:** The software aligns with healthcare regulations and standards, ensuring compliance with data protection laws and healthcare data exchange standards.This is crucial for maintaining data privacy and interoperability within the healthcare environment.
4. **Data Exchange:** The software may support data exchange with external systems or facilities, enabling the sharing of critical patient information. For example, it can export medical records or imaging data in a standardized format for referrals or consultations with external specialists or healthcare organizations.
5. **Patient Portal:** The software may offer a patient portal where individuals can access their health information from various providers, thus integrating data from different sources into one unified view.
6. **Security Measures:** It plays a critical role in maintaining the security and integrity of patient records. Robust security measures protect patient data from unauthorized access or breaches, aligning with broader healthcare cybersecurity standards.

By providing these functionalities and adhering to healthcare standards and regulations, the software becomes an integral part of the broader healthcare system. It enhances communication and data sharing among stakeholders, ultimately contributing to improved patient care and data management within the healthcare environment.

* **Discuss any predecessor or successor systems.**

It is essential to consider any predecessor or successor systems, as they can provide valuable insights into the software's evolution and integration within the broader healthcare environment.

**Predecessor Systems:**

1. **Manual Paper-Based Medical Records:** Many healthcare facilities and clinics still maintain paper-based medical records for their patients. These records typically include handwritten notes, test results, prescriptions, and other important health-related documents.
2. **Physical Media for Medical Imaging:** In the absence of digital systems, patients often receive their medical imaging results on CDs, which they can physically carry from one healthcare provider to another for consultation or treatment. These CDs contain images from radiology procedures like X-rays, MRIs, CT scans, and ultrasounds.

**Successor Systems:**

1. **Future Versions of the Software:** Successor systems could be future versions or updates of our medical memory software. These successors may introduce new features, improved user experiences, and enhanced security.
2. **Emerging Healthcare Technologies:** Future healthcare technologies, such as artificial intelligence (AI) and machine learning-driven diagnostic tools, may work in conjunction with our software to enhance patient care and data analysis. our software may evolve to be an integral component of these advanced healthcare solutions.
3. **Blockchain-Based Health Data Platforms:** Successor systems could explore blockchain technology for enhanced data security and patient data ownership. Our software could be adapted to integrate with blockchain-based health data platforms, ensuring even more robust data protection.

## 2.2 Product Functions

* **Summarize key functionalities the software will offer.**

1. Patient data management, including personal information, medical records, and imaging data.
2. Secure data storage and retrieval.
3. Medication management, including prescription and administration.
4. User roles and permissions, such as administrators, physician, and patients.
5. User interface features for easy navigation and data entry.
6. User authentication and authorization mechanisms.

* **Highlight the value or benefits for the end-users.**

The benefits that the end-user will gain from our software is that their personal information and medical files will all be in one place, making it easier for the end-user to change between hospitals, accessing their medical information easily.

## 2.3 User Characteristics

* **Detail user demographics, skills, or training levels.**

|  |  |  |  |
| --- | --- | --- | --- |
| User Type | Age Groups | Training/Education Level | English Level |
| Patients | +18 | Varied Education Levels | Intermediate |
| Healthcare Providers | Adults | Extensive Medical Training | Varying Technical Profiency |
| Administrative Staff | Adults | Training for System Training | Intermediate |
| Technical Support Personnel | Adults | Deep Technical Expertise | Intermediate |

* **Discuss any user-specific requirements or constraints.**

1. **User Training:** Develop user-friendly interfaces and offer training resources, especially for healthcare providers and administrative staff.
2. **Privacy and Security:** Implement robust data security measures and educate users on data protection practices.
3. **User Feedback Mechanism:** Include a feature for user feedback to enhance the software based on user experiences and suggestions.
4. **Mobile Responsiveness:** Ensure the software is accessible on various devices, including smartphones and tablets.
5. **Technical Support:** Provide responsive and knowledgeable technical support for addressing user queries and issues promptly.

These user-specific requirements emphasize the importance of accessibility, user support, data security, and user feedback for creating a user-friendly and effective medical memory software.

## 2.4 General Constraints

* **List constraints related to technology, regulations, or business rules.**

**Technology Constraints:**

* + **Data storage limitations:** Capacity constraints for storing large volumes of patient records, imaging data, and medical reports.
  + **Integration challenges:** Difficulty integrating the various systems (admission, management, imaging, pharmacy, etc.) to ensure seamless data flow and communication.
  + **Security vulnerabilities:** Risk of data breaches or unauthorized access to sensitive patient information due to potential loopholes in the security system.
  + **Compatibility issues:** Ensuring that the software is compatible with various operating systems, browsers, and devices to provide a seamless user experience for both patients and healthcare providers.

**Regulatory Constraints:**

* + **Data privacy and security regulations:** Data protection laws to protect patient data privacy and ensure data security.
  + **Medical data standards:** Compliance with industry-specific standards for storing, managing, and exchanging medical data.
  + **Authentication and access control:** Ensuring that the software adheres to regulations governing user authentication and access control to safeguard patient data from unauthorized access.

**Business Rules Constraints:**

* + **Data retention policies:** Compliance with data retention policies that dictate how long patient records and medical data must be stored and maintained.
  + **Patient consent and permissions:** Adherence to rules governing patient consent for data sharing and medical procedures, ensuring that the software complies with these consent requirements.

# 3. System Context

## 3.1 System Interfaces

* **Detail interactions with other software systems.**

Our medicomemroy system will establish interactions with multiple external systems to meet our system's requirements. Here's how it will work:

**1. Admission system:** Users will begin by signing in or signing up through the admission system to access their personal accounts.

**2. Patient Records:** The patient record system will contain comprehensive medical information about the patient, including their medical history.

**3. Physician Records:** The physician record system will store all essential details about the patient's personal physician.

**4. Account and Record Management:** A management system will be in place to manage user accounts, patient records, and physician records effectively.

**5. Pharmacy Services:** The pharmacy system will provide information about prescribed medications, including dosage and instructions.

**6. Laboratory Information Systems:** Physicians and lab technicians will utilize laboratory information systems to coordinate various medical tests, both inpatient and outpatient, spanning hematology, chemistry, immunology, and microbiology.

**7. Medical Imaging Information:** The medical imaging information system will offer data about radiology and non-radiology imaging, such as ultrasounds, x-rays, CT scans, MRIs, and more, for patients.

This collaborative integration with external systems enhances the functionality of our system and ensures comprehensive healthcare support for our users.

* **Describe data exchange or integration points.**

Data Exchange and Integration Points for the Patient Medical Record System:

**1. User Authentication and Access:**

- The patient medical record system integrates with a user authentication system to allow authorized users, such as healthcare providers and patients, to log in securely.

- It communicates user access rights and credentials for personalized data retrieval.

**2. Integration with Physician Records:**

- To ensure comprehensive patient care, the system exchanges data with the physician record system.

- This includes sharing patient information, appointment scheduling, and treatment plans, fostering a collaborative healthcare environment.

**3. Pharmacy Services Integration:**

- The system collaborates with the pharmacy system to exchange prescription information, medication lists, and dosage instructions, ensuring patients receive the right medications in a timely manner.

**4. Laboratory Information Systems:**

- The patient medical record system connects with laboratory information systems to facilitate the ordering of medical tests and the retrieval of test results.

- This ensures seamless coordination for inpatient and outpatient testing, including hematology, chemistry, immunology, and microbiology.

**5. Medical Imaging Information Integration:**

- For comprehensive patient care, the system integrates with a medical imaging information system.

- It shares information about radiology and non-radiology imaging, including data on ultrasounds, x-rays, CT scans, MRIs, and more.

**6. Account and Record Management:**

- The system exchanges data with the account and record management system to maintain accurate and up-to-date patient records. This includes personal information updates and record access controls.

**7. Data Backup and Recovery Integration:**

- To ensure data integrity, the system integrates with a data backup and recovery system. It regularly exchanges data for secure backup and recovery procedures.

**9. Security Measures:**

- Robust security measures are implemented to protect patient data during exchange, including encryption, access controls, and authentication mechanisms.

**10. Performance Requirements:**

- The system sets performance standards for data exchange, including response times for accessing patient records and handling simultaneous user requests.

## 3.2 User Interfaces

* **Describe GUI layout, navigation, or user interaction points.**

**Home Page:** Contain a navbar that has MedicoMemory Logo (Sign-Up, Log-In, About Us, Contact Us). Overview about MedicoMemory System, Get started button. Achievements, List of hospitals/clinics collaborates with our system   
  
**Sign-Up Page:** That includes textboxes for Full Name, Date of Birth, Email Address, Username, Password, Confirm Password, Security Questions (optional), Contact Information (if needed), Insurance Information (if relevant) and a Submit Button with clear error messages for validation.

**Log-In Page:** That includes textboxes for Username and Password, a "Forgot Password" or "Password Reset" link, and a Login button.

**About US Page:** An "About Us" page should include a brief description of our website, core values, vision, leadership team, privacy and security commitment, contact information, images and videos, and links to social media profiles.

**Contact Us Page:** The "Contact Us" page features input fields for users to enter their email, subject, and a message, along with a submission button.

Upon login or registration, a user profile page will be displayed, initially showing personal information. If no data is provided, textboxes will appear for the user to fill in; once all boxes are filled, they will disappear, leaving the organized information with an edit button at the top of the page. Additionally, a side menu will feature buttons for systematic profile, laboratory, radiology, non-radiology, medications, vaccinations, and medical reports. Every page will include relevant text fields for data input; for instance, the systematic profile page will permit the selection of specific body parts, leading to dedicated pages for detailed health conditions for the selected body part. Another example is the medication, the medication page will include drug name, dosage, frequency, start date, end date, any reactions, prescriber, and purpose.

* **Specify any standards or guidelines to be followed.**

**Ensuring the Security of Sensitive Patient Data:** In our website, the security and privacy of sensitive patient data stand as our foremost concern. We are committed to implementing the highest standards and guidelines to maintain the confidentiality and integrity of patient information. Compliance with these measures is not just a requirement; it's our ethical and legal obligation to safeguard the trust and confidence of our valued users.

## 3.3 Hardware Interfaces

* **List hardware components the software will interact with.**

Our medicomemroy system is a website-based software application that is accessible on various devices, the software will interact with a range of hardware components to ensure smooth functionality and optimal user experience. These hardware components include:

**Central Processing Unit (CPU):** The software will interact with the CPU of the user's device to process and execute instructions, ensuring efficient handling of various tasks performed on the website.

**Random Access Memory (RAM):** The software will utilize the device's RAM to temporarily store data and program instructions, enabling swift access to website elements and enhancing overall performance during user interactions.

**Storage Drives:** The software will interact with the storage drives of the user's device, including hard disk drives (HDDs) or solid-state drives (SSDs), to store and retrieve website data, user preferences, and any cached information necessary for a seamless browsing experience.

**Display and Graphics Processing Unit (GPU):** The software will engage with the device's display and GPU to render visual elements, graphics, and user interface components, ensuring that the website is visually appealing and easily navigable across different screen sizes and resolutions.

**Input Devices:** The software will interact with various input devices, such as keyboards, mice, touchscreens, or styluses, to enable users to interact with the website, input data, and navigate through different sections and features seamlessly.

**Network Interface Cards (NIC) and Connectivity Components:** The software will communicate with the device's network interface cards and other connectivity components, such as Wi-Fi or Ethernet adapters, to establish reliable internet connections, enabling users to access the website through various network infrastructures.

**Battery (for mobile devices):** In the case of mobile devices, the software will interact with the device's battery to optimize power consumption and ensure efficient operation, allowing users to access the website without excessive drain on the device's battery life.

* **Detail communication protocols or hardware specifications.**

**Hypertext Transfer Protocol (HTTP) or HTTPS:** These communication protocols enable the secure transfer of data between a web server and a user's web browser, ensuring the safe and efficient delivery of web pages and resources over the internet.

**Transmission Control Protocol (TCP) and Internet Protocol (IP):** TCP/IP protocols facilitate the transmission of data packets across interconnected networks, ensuring reliable and orderly communication between devices and systems, thereby enabling the seamless access and transfer of information over the internet.

**Ethernet or Wi-Fi Standards:** The software leverages Ethernet or Wi-Fi standards, such as IEEE 802.11, to establish reliable and secure local area network (LAN) or wireless network connections, allowing users to access the website through various devices and network infrastructures.

**Secure Socket Layer (SSL) and Transport Layer Security (TLS):** SSL and TLS protocols provide secure and encrypted communication channels, ensuring the confidentiality and integrity of data transmitted between the web server and the user's browser, thereby safeguarding sensitive information from unauthorized access and cyber threats.

**Internet Protocol Addressing (IPv4 or IPv6):** The software accommodates both IPv4 and IPv6 addressing formats, enabling the identification and location of devices connected to the internet, thereby ensuring seamless connectivity and data transfer between the website and users' devices, irrespective of their network configurations.

# 4. System Features & Requirements

## 4.1 System Features

For each feature, provide:

* Description and priority level
* Expected user interactions or inputs
* Expected system responses or outputs

**Feature #1 : Sign up**

* ***Description and Priority Level:*** The sign-up feature enables new users, including patients and doctors, to create accounts within the system, facilitating their access to the platform. This feature holds a high priority as it serves as the first step for users to engage with the system and access its functionalities.
* ***Expected User Interactions or Inputs:*** New users are required to provide essential information such as their full name, email address, date of birth, contact details, and a secure password. For doctors, additional information such as their medical license number and specialty might be required.
* ***Expected System Responses or Outputs:*** After the user submits the required information, the system should verify the data for completeness and validity. Upon successful validation, the system should create a unique user account for the individual, assigning a secure username and generating appropriate access credentials. The system should then display a confirmation message or notification, indicating that the sign-up process was successful. In case of any issues during the sign-up process, the system should provide clear error messages to guide the user on how to rectify the inputted information.

**Feature #2 : Log in to the System**

* ***Description and Priority Level:*** This feature allows users and doctors to securely access the system using their unique credentials, ensuring data security and confidentiality. It holds high priority as it serves as the primary gateway for all user interactions with the system.
* ***Expected User Interactions or Inputs:*** Users and doctors would be required to input their respective usernames and passwords or any other required authentication factors, such as special IDs for doctors, to access the system.
* ***Expected System Responses or Outputs:*** Upon successful authentication, the system should grant access to the user, allowing them to view the relevant information or perform the necessary actions based on their roles (user or doctor). In case of unsuccessful login attempts, the system should display appropriate error messages, guiding the user to retry or recover their credentials through a secure process.

**Feature #3 : View and Edit Profile (Patient)**

* ***Description and Priority Level:*** This feature allows patients to access and modify their own profile information within the system, including personal details, and other relevant data. It holds a medium to high priority, as it empowers patients to actively participate in managing their personal information.
* ***Expected User Interactions or Inputs:*** Patients would interact with the system by logging in and navigating to their profile section, where they can view their personal details, medical history, and other relevant information. And only edit their personal information profile, they would input the updated information through designated form fields.
* ***Expected System Responses or Outputs:*** The system should display the patient's profile information in a clear and organized manner, allowing them to easily comprehend their medical data. When the patient initiates an edit, the system should validate the changes and update the information accordingly. After successful editing, the system should display a confirmation message, informing the patient that the changes were saved successfully. In case of any issues during the editing process, the system should provide clear error messages to guide the patient on how to rectify the changes.

**Feature: View and Edit Profile (Physician)**

* ***Description and Priority Level:*** This feature enables physicians to access and modify patient profiles, including medical records, laboratory results, radiology information, medication details, and medical imaging data. It holds a high priority, as it is essential for physicians to have comprehensive and up-to-date patient information for accurate diagnosis and treatment.
* ***Expected User Interactions or Inputs:*** Physicians would interact with the system by logging in and navigating to the patient's profile section, where they can view all relevant medical data. To make changes, physicians would input updated information, such as diagnosis, treatment plans, or prescription details, through the designated forms or fields.
* ***Expected System Responses or Outputs:*** The system should present the patient's complete medical information in an organized and easily accessible format for the physician's review. When the physician initiates an edit, the system should validate the changes and update the information accordingly. After successful editing, the system should display a confirmation message, informing the physician that the changes were saved successfully. In case of any issues during the editing process, the system should provide clear error messages to guide the physician on how to rectify any inaccuracies.

## 4.2 External Interface Requirements

* **Detail specifications for software, hardware, and user interfaces.**

1. **Software Interfaces:**

- The software should integrate with a secure database system to store and retrieve patient records, including personal information, medical data, medications, laboratory results, and radiology/non-radiology information.

- The software should provide integration with Google and email services on the sign-up and sign-in pages, enabling users to conveniently use their Google accounts or personal email addresses for direct sign-up and sign-in.

1. **Hardware Interfaces:**

- The software should be compatible with common hardware components, such as personal computers, laptops, and mobile devices, to ensure accessibility for healthcare providers and patients.

1. **User Interfaces:**

The software's graphical user interface (GUI) should be designed to facilitate user interactions. It should include:

- Input forms for inserting personal information (name, date of birth, address, etc.).

- Access to view a patient's own medical information.

- Access to view doctor's information, including name, specialty, phone number, and email.

- A secure login interface for patients, accessible with a special ID.

- Options for healthcare providers to edit a patient's medical information.

- An interface for inserting required medications for the patient.

- A feature for uploading documents, such as medical reports and imaging information.

- Authentication and verification mechanisms for patient accounts.

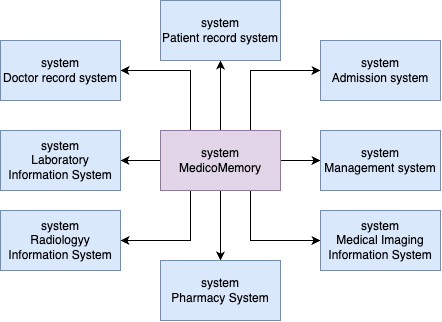
- Security features to ensure the integrity and confidentiality of patient data.

- A user-friendly layout for viewing and editing patient records.

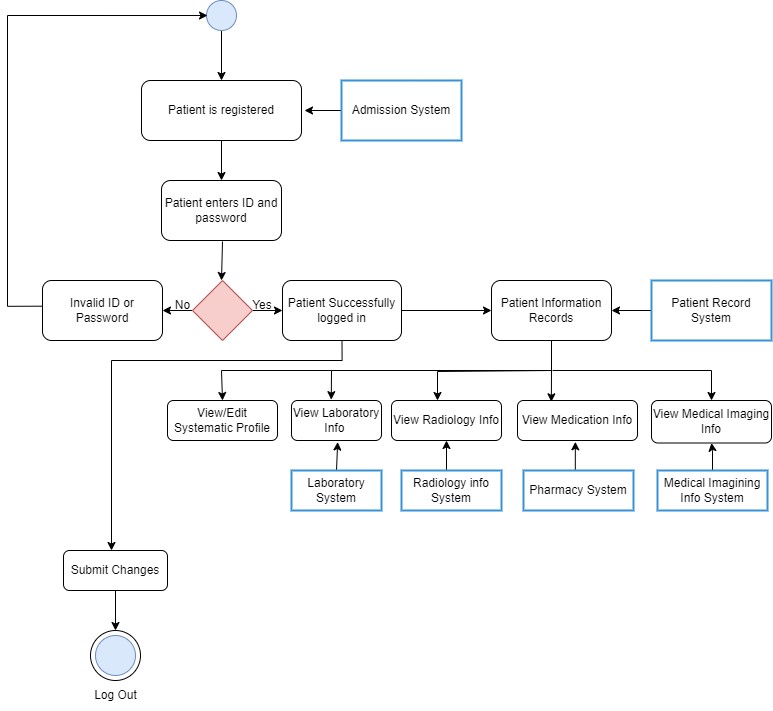
- A system for recording the last edited name (doctor's name) when changes are made to the records.

## 4.3 System Models

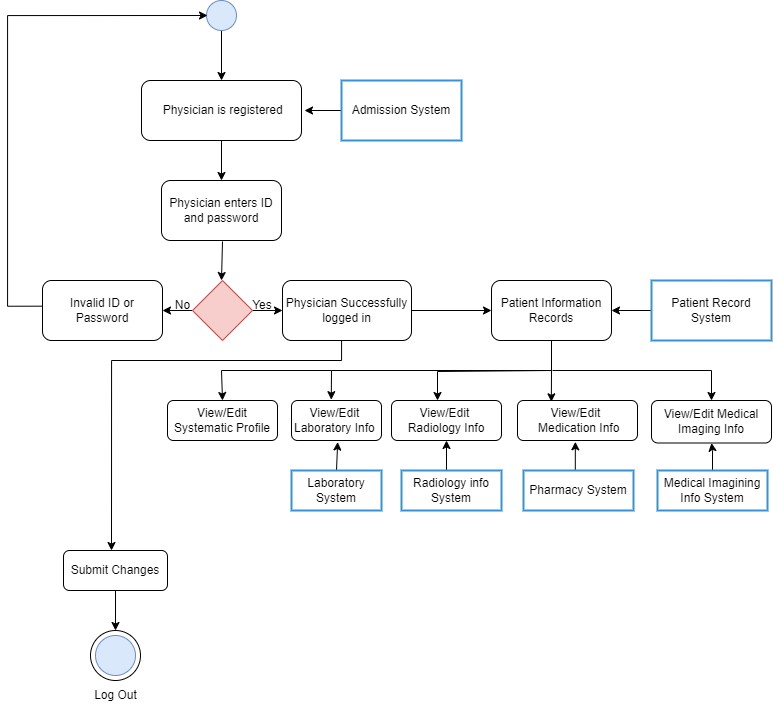
* **Context Model**

****

**Description:** The diagram depicts the integration of various medical information systems centered around "MedicoMemory." This core system interfaces with several others, including the Doctor record system, Patient record system, Laboratory Information System, Radiology Information System, Admission system, Management system, Medical Imaging Information System, and Pharmacy System. These connections indicate a centralized approach where "MedicoMemory" acts as a hub for data exchange and coordination among the various healthcare-related systems.

* **Activity Diagrams**

**Description:** The diagram outlines a patient's interaction with our healthcare information system. Initially, the patient's registration is verified by the Admission System. Following this, the patient logs in by entering their ID and password. If the credentials are correct, the patient gains access to the Patient Information Records within the Patient Record System. From here, the patient can view or edit their systematic profile, access laboratory data from the Laboratory System, review radiology information from the Radiology info System, check medication details from the Pharmacy System, and view medical imaging data from the Medical Imaging Info System.



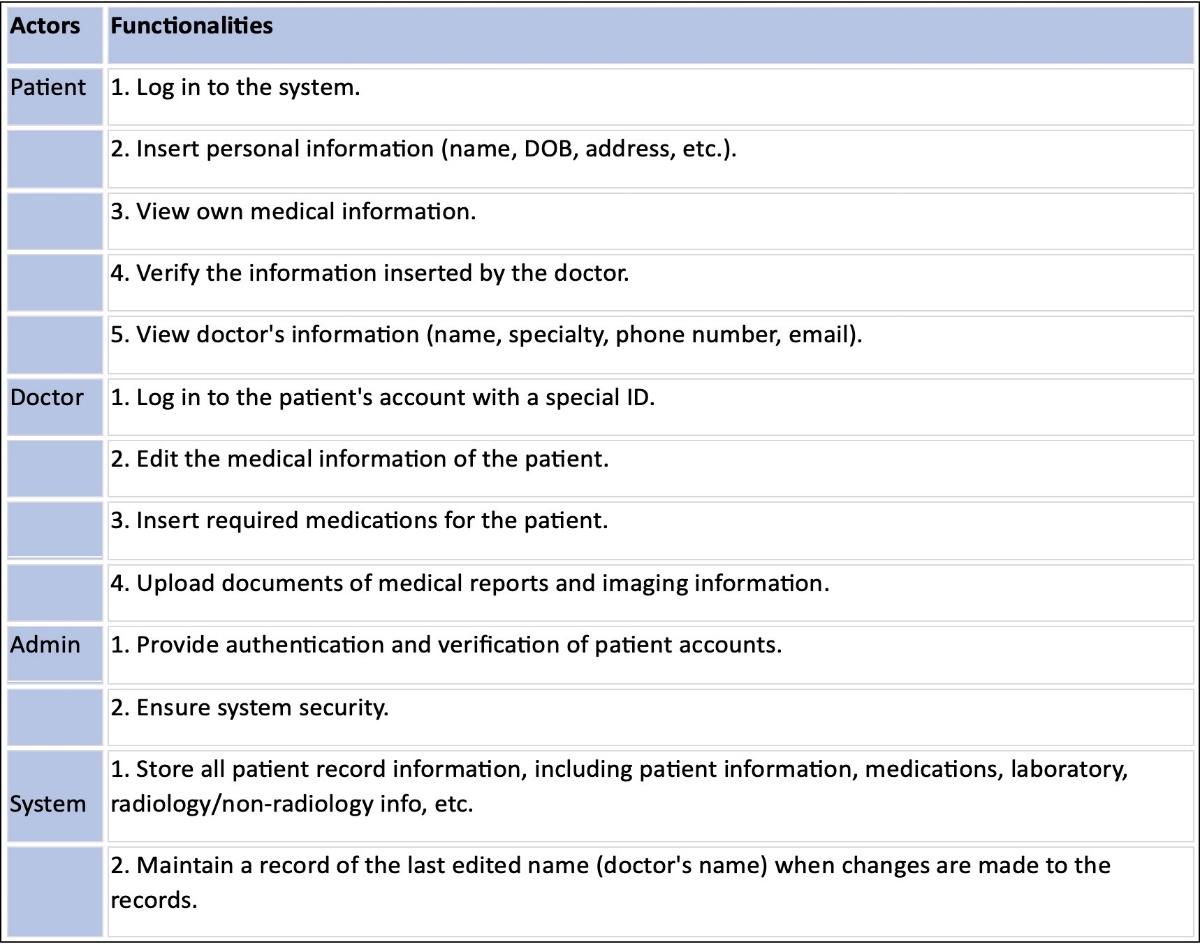
**Description:** The diagram showcases a physician's interaction flow with a healthcare information system. Upon verification of the physician's registration by the Admission System, they can log in using their ID and password. Successful authentication provides the physician access to the Patient Information Records within the Patient Record System. From this point, the physician has the capability to view and edit various data sections: the systematic profile, laboratory details via the Laboratory System, radiology information from the Radiology info System, medication particulars from the Pharmacy System, and medical imaging data from the Medical Imaging Info System. Any modifications made can be finalized by submitting changes, and the physician can subsequently log out when done.

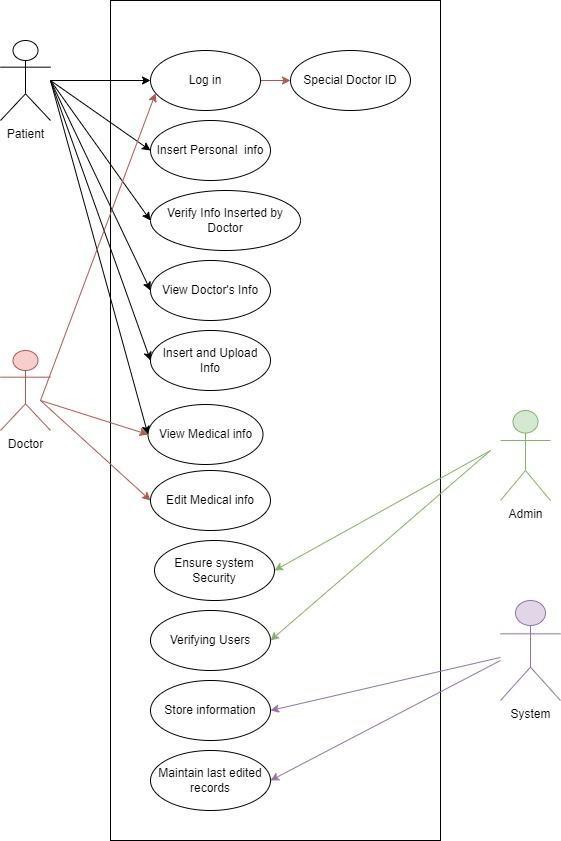
A flowchart of a computer program

Description automatically generated

**Description:** The diagram illustrates the user flow for website navigation and authentication through the Admission System. When a user navigates to the website, they're presented with a query to determine if they already have an account. Users without an account are directed to a "Create Account" form, where they input data, including email and password, which gets saved. Conversely, existing account holders are prompted for their username and password. These credentials undergo a validation process. If the credentials are validated successfully, users are directed to the dashboard; otherwise, they're prompted about the invalid input and asked to re-enter their details. From the dashboard, users have the option to log out.

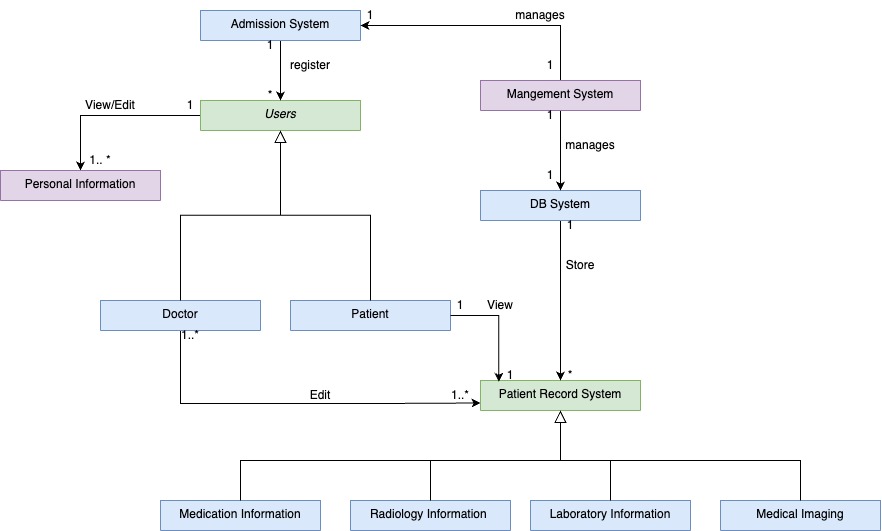
* **Use Case Diagram**





**Description:** The use case diagram delineates the interactions among Patients, Doctors, Admin, and a System within a healthcare platform. Patients have functionalities such as logging in (with a special doctor ID option), inputting personal information, verifying the information provided by the doctor, and viewing the doctor's details. Conversely, Doctors can insert, upload, view, and edit medical information. The Admin role emphasizes system security, verifying users, and overseeing information storage. Lastly, the System is responsible for securely storing data and maintaining records of the most recent edits. Each role's actions intersect, showcasing the interdependence and collaborative nature of the platform.

* **Class Diagram**



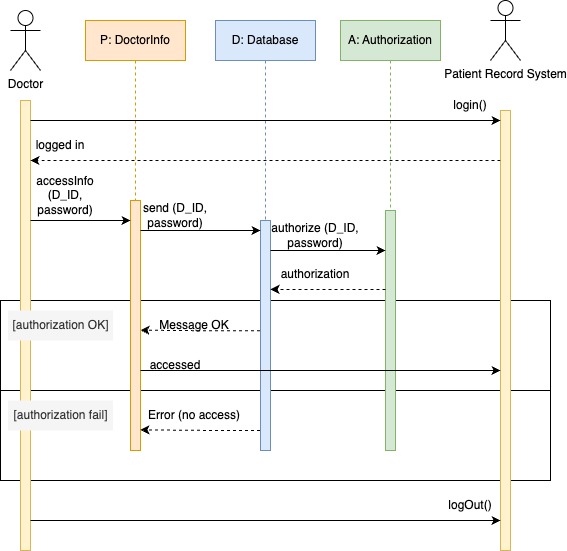
**Description:** The class diagram portrays the structure and relationships within our healthcare system. Central to the diagram is the "Users" class, which can register through the "Admission System." These users are further categorized as "Doctor" or "Patient." Doctors and Patients can view and edit personal information. A "Patient Record System" is in place for viewing patient records, which includes information like medication, radiology, laboratory details, and medical imaging. Furthermore, the "Management System" oversees the "DB System," responsible for data storage. This comprehensive layout signifies the intricacies of managing patient data, user access, and system control within a medical facility.

* **Sequence Diagrams**

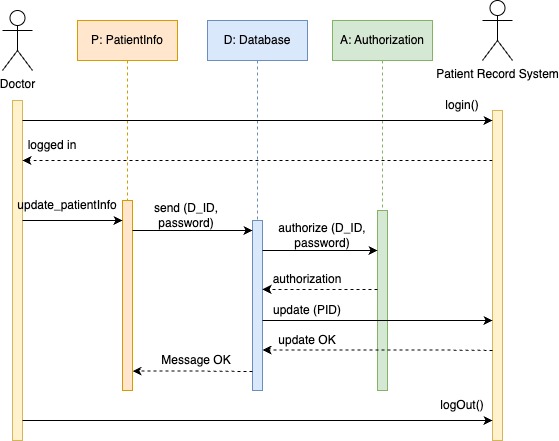
A diagram of a patient login

Description automatically generated

**Description:** The sequence diagram illustrates the login authorization process for a patient trying to access their records in the Patient Record System. The patient, represented on the left, initiates the process by sending their personal identification (PID) and password to the "PatientInfo" system. This information is then transmitted to the "Database," which communicates with the "Authorization" module to validate the credentials. If authorization is successful, the patient receives a "Message OK" indicating successful access to their records, and they are logged into the Patient Record System. Conversely, if the authorization fails, the patient receives an "Error (no access)" message. The sequence concludes with the patient logging out of the system.



**Description:** The sequence diagram showcases the process a doctor undergoes to access the Patient Record System. The doctor, already logged into the system, attempts to retrieve doctor-specific information by submitting their unique identification (D\_ID) and password to the "DoctorInfo" module. This module relays the credentials to the "Database" for validation. In tandem, the "Database" communicates with the "Authorization" module to confirm the doctor's identity. If the authorization is successful, a "Message OK" signal indicates the doctor has been granted access; however, if there's a failure in authorization, an "Error (no access)" message is relayed. The process concludes with the doctor logging out of the system.



**Description:** The sequence diagram depicts a doctor's interaction with the Patient Record System to update patient information. Initially, the doctor is logged into the system. When the doctor decides to update a patient's information, they interact with the "PatientInfo" module, which then forwards the doctor's identification (D\_ID) and password to the "Database" for verification. The "Database" collaborates with the "Authorization" module to authenticate the doctor's credentials. Upon successful authorization, the doctor is permitted to update the patient's record, identified by their personal identification (PID). The update is confirmed with an "update OK" message, and the sequence culminates with the doctor logging out of the system.

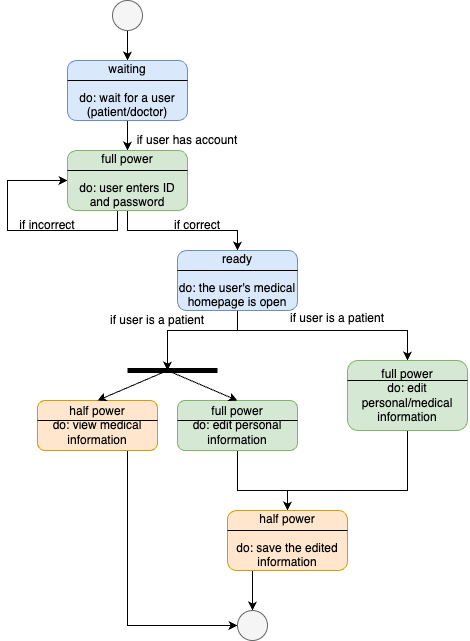
* **State Diagrams**

A diagram of a user account

Description automatically generated

|  |  |
| --- | --- |
| States | Description |
| Waiting/ Ready | Represent the state where the software is idle and isn’t performing any operation |
| Full Power | Represents the state where the software is performing an operation. |
| Half Power | Represent the state where the software is in a transitional phase of the operation. |

**Description:** The State diagram illustrates the authentication process for a user, whether they are a patient or a doctor, accessing a medical system. Initially, the system is in a waiting state, anticipating a user's interaction. When a user approaches, the system checks whether the user already possesses an account. If they do, they're prompted to input their ID and password. In cases where the provided credentials are accurate, the user is directed to their medical homepage. Conversely, if the credentials are incorrect, the process loops back to the user entering their details. If the user doesn't have an existing account, they are provided with full permissions to register, necessitating the input of personal details like ID, full name, and date of birth to create a new account.



**Description:** The system remains in a waiting state for user input. Once a user, either a patient or doctor, initiates interaction and possesses an account, they are prompted to input their ID and password. Upon successful authentication, they access their medical homepage. For users identified as patients, there are differentiated access levels: they can modify personal details but only view medical data. After making potential updates, patients can save the altered information.

# 5. Non-functional Requirements

## 5.1 Performance Requirements

* Specify expected response times, throughput, or scalability metrics.

1. **Response Time:** We should aim for pages to load swiftly, ideally within 2-3 seconds, ensuring a smooth and responsive user experience.
2. **Throughput:** We need to determine the maximum number of users our website can handle simultaneously without performance degradation. Let's target, for example, the capability to support 500 concurrent users.
3. **Scalability Metrics:**

* *Scaling for More Users:* It's important to plan for the possibility that our website becomes really popular, with more and more people using it. We should make sure it can handle the increased traffic smoothly. One way to achieve this is by adding more servers or computers when the need arises.
* *Managing Growing Medical Data*: As we continue to gather patient records and medical information, the amount of data can become quite substantial. To ensure our website performs well, we should think about strategies to handle this growth effectively. This might involve splitting the data into smaller parts (sharding), creating duplicate copies (replication), or using specialized databases like NoSQL that can manage large volumes of data efficiently. These measures will be essential for maintaining the website's speed and reliability as it expands.

## 5.2 Safety Requirements

* Detail measures to prevent harm or loss in system operation.
* **Regular Data Backups**: Schedule automated, frequent backups of all patient data to prevent data loss due to various types of failures or security incidents.
* **Failover Mechanisms**: Set up failover mechanisms to ensure uninterrupted service in the event of server or infrastructure failures.
* **Emergency Shutdown Procedures**: Develop and document procedures for emergency system shutdown in case of security breaches or other critical incidents to prevent unauthorized access or data corruption.

## 5.3 Security Requirements

* List security protocols, encryption methods, or access controls.
* Detail any compliance or industry standards.
* **Transport Layer Security (TLS)**: Implement TLS to secure data in transit, ensuring that all communication between users and the system is encrypted.
* **User Authentication**: Use user authentication methods, including password policies, to control access to the system.
* **Access Control Lists (ACLs)**: Apply access control lists to restrict data access to authorized users only, based on their roles and responsibilities. { admins, patients, and physcians }
* **HIPAA Compliance**: Adhere to the Health Insurance Portability and Accountability Act (HIPAA) standards and regulations to ensure the protection of patient health information.

## 5.4 Software Quality Attributes

* Specify requirements for maintainability, portability, or reliability.
* Detail any testing or quality assurance procedures.
* **Maintainability:** our codebase will adhere to the best coding practices and will follow established coding standards. This will include creating clear, well-documented code to make future maintenance and updates more straightforward.
* **Portability:** We will ensure that the website becomes compatible with multiple web browsers and mobile devices, broadening our audience's reach and improving accessibility.
* **Reliability:** we will ensure that our website will achieve a high uptime percentage, and will be capable of handling sudden surges in traffic without any service disruptions.
* **Testing and Quality Assurance:** We will implement a robust testing strategy, which will encompass unit testing, integration testing, and user acceptance testing. We will regularly conduct security assessments, vulnerability scans, and penetration testing to identify and rectify any weaknesses.
* **Quality Assurance Procedures:** We will adhere to a strict quality assurance process to verify that the website meets specified security and performance requirements. This will involve code reviews, static code analysis, and continuous integration practices to maintain high standards.

# Other Requirements

Our comprehensive software requirements report encapsulates all essential aspects, leaving no room for additional requirements. Through precise analysis and thorough consideration of various factors, we've attempted to encompass every aspect of the project's needs. This report serves as a comprehensive guide, detailing functional requirements, non-functional specifications, security measures, compatibility criteria, regulatory compliance, documentation necessities, testing parameters, and maintenance expectations. The exhaustive nature of this document ensures that all vital components have been addressed, providing a robust foundation for the successful development and implementation of the software. As such, we confidently assert that no further requirements are necessary for the successful realization of the project's objectives.

# 7. Agile Project Management

## 7.1 Team Composition and Roles

* **List all team members and define their roles (e.g., Product Owner, Scrum Master, Developer, Tester).**

**Product owner:** Leen Sharab

**Scrum master:** Reema Abdullah

**Developers:** Sarah Alshumayri, Leen Sharab, Reema Abdullah, Lujain Almarri, Ameera Attiah

**Testers:** Sarah Alshumayri, Leen Sharab, Reema Abdullah, Lujain Almarri, Ameera Attiah

* **Identify the designated team leader and their responsibilities for this phase (note that the team leader should be changed for the next phase)**

Our team leader for this phase is **Reema Abdullah**, and her responsibilities for this phase are to make sure that everyone is doing their part properly, helping each team member when they’re struggling, and make sure that we’ll finish our work on time.

**Next Team Leader:** Ameera Attiah

**Following her:** Lujain Almarri

## 7.2 Collaboration and Communication

* **Describe the tools and platforms used for team communication (e.g., Slack, Microsoft Teams).**

The tools and platforms used for communicating between team members are Whatsapp and Zoom.

* **Explain how the team will handle decision-making and conflict resolution.**

Our team manages decision-making and conflict resolution through a structured process that involves each team member expressing their opinions, gathering these opinions, and subsequently conducting a team-wide vote.

## 7.3 Iterative Development

* **Detail the sprint or iteration length (e.g., 2 weeks).**

In our project, we have adopted a 1-week sprint duration. We found that this length allows us to strike a balance between delivering valuable increments of our product and maintaining a sustainable pace for the development team.

* **Describe the approach for backlog refinement, sprint planning, daily stand-ups, and sprint retrospectives.**

**Backlog Refinement:** We conduct regular backlog refinement sessions as part of our ongoing process. These sessions typically occur every week, allowing us to add, remove, or clarify items in the product backlog. It is a collaborative effort involving the Product Owner, Scrum Master, and development team.

**Sprint Planning:** Our sprint planning meetings are well-structured. We start with a sprint goal defined by the Product Owner. The team reviews the prioritized backlog items and commits to a specific set of user stories for the sprint.

**Daily Stand-ups:** Our daily stand-up meetings are held each morning. Team members provide brief updates on their work, focusing on what they completed the previous day, what they plan to work on today, and any obstacles they're facing. The Scrum Master ensures that these meetings stay concise and productive.

**Sprint Retrospectives:** Sprint retrospectives are a crucial part of our process. We hold these meetings at the end of each sprint to reflect on our performance. We identify what went well and what could be improved. The team collectively defines action items for continuous improvement, and the Scrum Master follows up on their implementation.

* **Discuss any adaptations or customizations made to standard Agile methodologies.**

We have made a few adaptations to standard Agile methodologies to better suit our project. For instance, we have tailored our definition of "Done" to include specific quality and testing criteria that align with our industry standards. We have also introduced a mid-sprint check-in to address any emerging issues or adjustments needed to meet the sprint goal. These adaptations have helped us maintain a high level of quality and adaptability in our project.

## 7.4 Deliverables and Milestones

* **List the expected deliverables for each sprint or iteration.**

**Sprint 1:** In the beginning of our project, we took the first steps. We organized our workspace and made sure all our tools were ready. We made a list of all the important tasks we need to do, ranking them from most to least important. We also filled out the SRS document to collect all the requirements for the software we're making for our client. This got us well-prepared to start working on our project.

**Sprint 2:** the primary deliverables involved the implementation of core system functionalities. This includes the development of user registration and authentication features, allowing users to sign up and log in to their accounts.

**Sprint 3:** the primary deliverable is the development of the patient profile management features. This includes allowing users to update and manage their personal and medical information. This sprint also includes the development of the physician’s profile with all the access control protocols.

**Sprint 4:** this sprint involves the development of prescription management features, allowing healthcare providers to prescribe medications, specify dosages, and track prescription history. It also involves the integration with external systems, specifically laboratory information systems for test orders and results and medical imaging systems that enables users to access and store radiology and non-radiology images.

**Sprint 5:** our main focus was on strengthening the security of our software. We enhanced access controls, added data encryption, improved user authentication, documented security measures, and conducted thorough security testing. These efforts ensured that our system is well-protected and prepared to defend against potential security threats, safeguarding user data and maintaining trust and confidence.

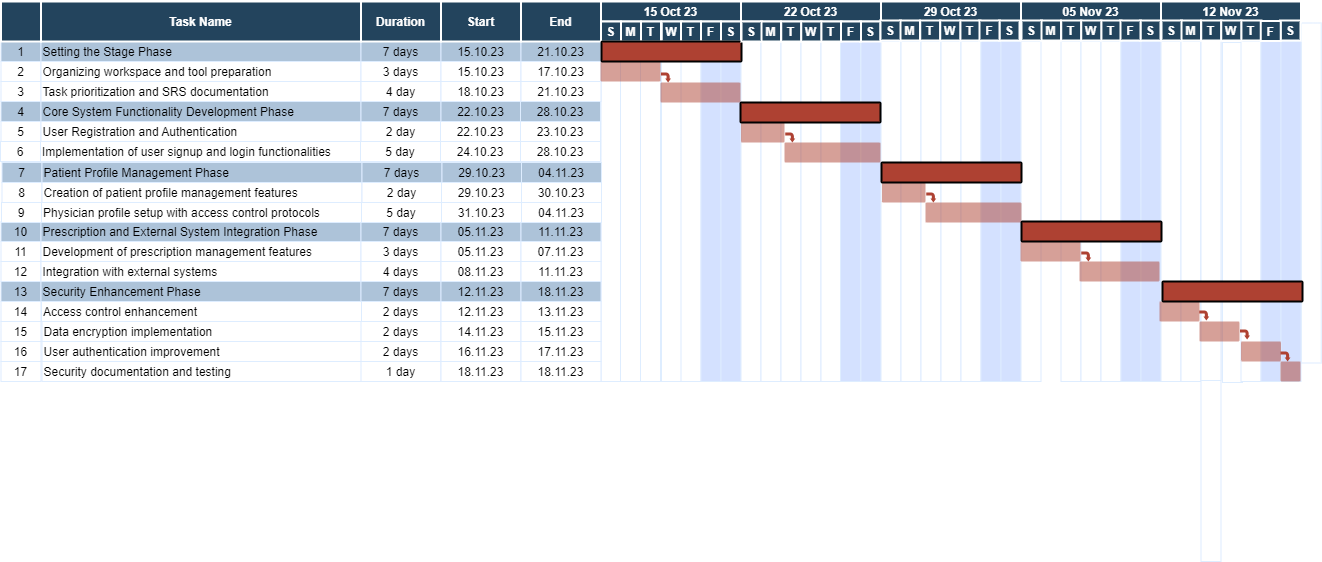
After completing each sprint, we will conduct testing and validation with our client to ensure that everything is proceeding as intended and to address any client feedback or requirements. This ongoing validation process will help us maintain alignment with the client's needs and expectations throughout the project.

* **Identify major project milestones (e.g., alpha release, beta testing).**

The major project milestones include the Alpha Release, which signifies the completion of core functionalities, patient and physician profiles, and prescription management features, initiating internal testing and validation to ensure software functionality and stability before further development.

# 8. Project Timeline (Gantt Chart)

* Provide a visual representation of the project timeline, showing tasks, durations, dependencies, and milestones.
* Alternatively, describe the major phases and their estimated completion dates.



# Appendix A: Glossary

* Define terms and abbreviations.

|  |  |  |
| --- | --- | --- |
| Word | Definition | Page number |
| Account and Record Management | System in place to manage user accounts, patient records, and physician records effectively. | 6 |
| Activity Diagrams | Depict the flow of activities within the system. | 13 |
| Admission System | A system allowing users to sign in or sign up, serving as the entry point to access personal accounts. | 6 |
| Alpha Release | Completion of core functionalities, patient and physician profiles, and prescription management features. | 24 |
| Backlog Refinement | Weekly sessions for adding, removing, or clarifying items in the product backlog. | 23 |
| Blockchain Technology | A decentralized and secure digital ledger system that can be used to record transactions across many computers, enhancing the security and transparency of data management in various industries, including healthcare. | 4 |
| Capacity Constraints | Restrictions on the amount of data that can be stored, particularly concerning patient records, imaging data, and medical reports. | 5 |
| Class Diagram | Represents the structure of the system in terms of classes and their relationships. | 17 |
| Context Model | Illustrates the system's interactions with external entities. | 12 |
| Daily Stand-ups | Daily meetings for brief updates on work progress. | 23 |
| Data Breach Risk | The potential for unauthorized access or disclosure of sensitive patient information due to vulnerabilities in the security infrastructure. | 6 |
| Data Exchange | The seamless transfer of patient information and medical records between the software and external systems or facilities, ensuring efficient communication and collaboration within the healthcare network. | 3 |
| Demographic Details | In-depth information about the characteristics and traits of users, encompassing age, education levels, and technical proficiency. | 5 |
| Designated Team Leader | The Designated Team Leader's responsibilities include overseeing team progress, assisting team members, and ensuring timely completion of tasks. | 22 |
| Developers | Developers are team members responsible for implementing software features based on the project requirements. | 22 |
| Display and Graphics Processing Unit (GPU) | Renders visual elements, graphics, and user interface components. | 8 |
| Educational Proficiency | The extent of knowledge and skills acquired through formal training or education, especially relevant to healthcare providers and administrative staff. | 5 |
| Ethernet or Wi-Fi Standards | Establishes reliable and secure local area or wireless network connections. | 10 |
| Hardware Interfaces | * + 1. Compatible with personal computers, laptops, and mobile devices for accessibility. | 11 |
| HTTP or HTTPS | Enables secure data transfer between web servers and browsers. | 9 |
| Industry-Specific Compliance | Ensuring that the software aligns with established standards for the storage, management, and exchange of medical data within the healthcare industry. | 6 |
| Integration with Healthcare Providers | The process of linking the software with various healthcare professionals and institutions, facilitating the exchange of patient data and medical information within the broader healthcare ecosystem. | 3 |
| Laboratory Information Systems | Coordinates various medical tests, both inpatient and outpatient, spanning hematology, chemistry, immunology, and microbiology. | 7 |
| Medical Imaging Information System: | Offers data about radiology and non-radiology imaging, such as ultrasounds, x-rays, CT scans, MRIs, etc. | 7 |
| Patient Data Management | The comprehensive process of organizing, storing, and accessing patient-related information, encompassing personal details, medical records, test results, and imaging data within the software. | 4 |
| Patient Portal | An online platform that grants patients access to their health information, allowing them to view medical records, test results, and other relevant data from various healthcare providers in one centralized location | 3 |
| Patient Records System | Stores comprehensive medical information about patients, including their medical history. | 6 |
| Patient-Centric Approach | A methodology focused on putting the patient at the center of the healthcare experience, allowing them to actively participate in managing their medical information and treatment processes. | 3 |
| Performance Requirements | Sets standards for response times and handling simultaneous user requests during data exchange. | 8 |
| Pharmacy Services System | Provides information about prescribed medications, including dosage and instructions. | 6 |
| Physician Records System | Contains essential details about the patient's personal physician. | 6 |
| Predecessor Systems | Previous manual or physical methods of managing medical information and data, such as paper-based records and physical media for medical imaging, were used before the introduction of the current software. | 4 |
| Product Owner | The Product Owner is responsible for defining and prioritizing product features, ensuring alignment with client requirements. | 22 |
| Regulatory Compliance | The adherence to established laws, regulations, and industry standards governing the management, protection, and exchange of healthcare data and information. | 3 |
| Response Time | The time it takes for pages to load, ideally within 2-3 seconds, ensuring a smooth and responsive user experience. | 20 |
| Robust Security Protocols | Strong and comprehensive measures implemented to safeguard data from unauthorized access, ensuring user privacy and compliance with privacy regulations. | 5 |
| Scrum Master | The Scrum Master facilitates the Scrum process, removes impediments, and ensures the team adheres to Agile principles. | 22 |
| Security Measures | Robust protocols and technologies implemented to safeguard sensitive patient data from unauthorized access, ensuring data integrity and privacy within the healthcare environment. | 3 |
| Security Measures | Implements encryption, access controls, and authentication mechanisms for secure data exchange. | 8 |
| Security of Sensitive Patient Data | Adheres to the highest standards and guidelines for maintaining the confidentiality and integrity of patient information, ensuring ethical and legal obligations are met. | 8 |
| Sequence Diagrams | Display interactions between system components over time. | 17 |
| Software Interfaces | * + 1. Integrates with a secure database system for storing and retrieving patient records. | 11 |
| Sprint Planning | Well-structured meetings starting with a sprint goal defined by the Product Owner. | 23 |
| Stakeholders | Stakeholders are individuals or groups with a vested interest or concern in a project, organization, or system, whose involvement and input can influence its outcomes and decisions. | 1 |
| State Diagrams | Illustrate the different states a system can exist in and transitions between them. | 19 |
| Successor Systems | Future iterations or advancements of the software, as well as emerging healthcare technologies like blockchain-based health data platforms, designed to build upon the current software's capabilities and provide enhanced functionalities and data security. | 4 |
| Tailored Needs | Specific demands and necessities unique to each category of users, influencing the design and functionality of the software. | 5 |
| TCP and IP | Facilitates transmission of data packets across interconnected networks. | 9 |
| Technical Expertise | The depth of knowledge and skill in handling technical aspects, particularly pertaining to healthcare providers and technical support personnel. | 5 |
| Testers | Testers are team members responsible for testing and validating software functionality to ensure quality and identify defects. | 22 |
| Throughput | The maximum number of users the website can handle simultaneously without performance degradation, targeting the capability to support 500 concurrent users. | 20 |
| Transport Layer Security (TLS) | Implementation of TLS to encrypt data in transit, ensuring secure communication. | 21 |
| Use Case Diagram | Illustrates system functionality from a user's perspective. | 15 |
| User Authentication | Utilization of user authentication methods, including password policies, to control system access. | 21 |
| User Authentication and Access | Integrates with a user authentication system for secure logins and personalized data retrieval. | 7 |
| User Authentication and Authorization Mechanisms | The implementation of security protocols and procedures that verify user identities and grant appropriate access privileges based on predefined roles and permissions, ensuring data protection and privacy for all users. | 4 |

Software Requirements Specification (SRS) Assignment Rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Criteria (Points)** | **Exemplary** | **Proficient** | **Needs Improvement** |
| **Introduction** | Purpose (2 pts) | Objectives and purposes clearly defined | Vague or incomplete definition | Missing or irrelevant definition |
| **Introduction** | Scope (3 pts) | Comprehensive delineation of software boundaries and capabilities | Partially described or some missing details | Vague or missing description |
| **Introduction** | Definitions, Acronyms, Abbreviations (2 pts) | Comprehensive list with clear definitions | Some terms missing or inadequately defined | No list or definitions provided |
| **Introduction** | References (2 pts) | Relevant references cited with proper format | Some references missing or improperly cited | No references provided |
| **Introduction** | Overview (1 pts) | Clear summary of the SRS content structure | Vague or missing overview |  |
| **General Description** | Product Perspective, Functions, User Characteristics, and Constraints (10 pts) | Comprehensive and clear description of all sub-points | Minor details missing or inadequately described | Major details missing or vaguely described |
| **System Context** | System, User, and Hardware Interfaces (15 pts) | Detailed descriptions with relevant examples or scenarios | Minor details missing or inadequately described | Major details missing or vaguely described |
| **System Features & Requirements** | System Features, External Interface Requirements (15 pts) | Comprehensive descriptions with clear diagrams and examples | Minor details or diagrams missing or inadequately described | Major details or diagrams missing or vaguely described |
| **System Features & Requirements** | System Models (25 pts) | Clear and comprehensive context, activity, use case, and sequence diagrams | Minor details or diagrams missing or inadequately described | Major details or diagrams missing or vaguely described |
| **Non-functional Requirements** | Performance, Safety, Security, Software Quality Attributes (20 pts) | Comprehensive and relevant descriptions | Minor details missing or inadequately described | Major details missing or vaguely described |
| **Other Requirements & Appendices** | Other Requirements, Glossary, Use Case Details (10 pts) | Comprehensive and relevant descriptions | Minor details missing or inadequately described | Major details missing or vaguely described |
| **Agile Project Management** | Team Composition, Collaboration, Iterative Development, Deliverables (20 pts) | Comprehensive descriptions with clear roles and processes | Minor details missing or inadequately described | Major details missing or vaguely described |
| **Project Timeline** | Gantt Chart or Timeline Description (10 pts) | Clear visual representation or description with milestones | Missing some tasks or milestones | Vague or missing major details |