1/29/2025

Software-Requirement Specification

Patient Management System

University of Chakwal

DEPARTMENT OF COMPUTER SCIENCE

SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT

Project Name: Patient Management System

By: Sara Arif

Date: 29th January ,2025

Appendix

REQUIREMENTS ELICITATION TECHNIQUES

INTERVIEWS:

Conduct one-on-one interviews with various stakeholders (doctors, nurses, admin staff, patients) to understand their needs and expectations. Interviews provide detailed, qualitative insights and allow direct communication with end-users to clarify their specific requirements for the system.

1. SIMULATED INTERVIEWS

Interviews are conducted with stakeholders like doctors, nurses, admin staff, and patients to gather in-depth insights.

INTERVIEW WITH A DOCTOR

Q1: WHAT FEATURES WOULD YOU LIKE TO SEE IN THE PATIENT MANAGEMENT SYSTEM?
A: "I would like to access patients' complete medical history before consultations. It's essential to have quick and easy access to notes, lab reports, and past prescriptions."

Q2: DO YOU FACE ANY CHALLENGES WITH THE CURRENT SYSTEM?
A: "Yes, it's time-consuming to manually search through records. Also, the lack of reminders for follow-ups often leads to delays in patient care."

Q3: WOULD YOU PREFER ANY SPECIFIC INTERFACE DESIGN FOR THIS SYSTEM?
A: "It should be simple to navigate and work seamlessly on tablets and desktop computers. A search function for patient records would be very helpful."

INTERVIEW WITH A NURSE

Q1: WHAT WOULD MAKE PATIENT MANAGEMENT EASIER FOR YOU?
A: "An efficient way to add new patient information and update records in real time would save us a lot of time."

Q2: HOW DO YOU CURRENTLY MANAGE PATIENT DATA?
A: "Right now, we rely on a combination of paper records and basic software, which often causes duplication errors."

Q3: WHAT LEVEL OF ACCESS DO YOU NEED IN THE SYSTEM?
A: "I only need to update patient information and add new data. I don't need access to billing or appointment management."

INTERVIEW WITH AN ADMIN STAFF MEMBER

Q1: WHAT ARE YOUR KEY RESPONSIBILITIES REGARDING PATIENT DATA?
A: "I handle appointment scheduling, patient registration, and billing."

Q2: WHAT CHALLENGES DO YOU FACE IN YOUR ROLE?
A: "Generating accurate invoices is time-consuming because the current system doesn't integrate billing with patient records."

Q3: WHAT SYSTEM FEATURES WOULD IMPROVE YOUR WORKFLOW?
A: "Automated invoice generation and an appointment reminder system would be a game-changer for us."

INTERVIEW WITH A PATIENT

Q1: HOW DO YOU CURRENTLY BOOK APPOINTMENTS?
A: "I usually call the clinic, but sometimes it's hard to get through, and I wish there was an online option."

Q2: WOULD YOU USE A SYSTEM TO VIEW YOUR MEDICAL RECORDS?
A: "Yes, especially for downloading reports or viewing prescriptions."

Q3: WHAT REMINDERS WOULD YOU FIND HELPFUL?
A: "Appointment reminders via SMS or email would be great to avoid missed appointments."

SURVEYS:

Distribute questionnaires to a larger group of stakeholders to gather quantitative data. This method can help capture feedback from many users, especially regarding general usability preferences and common system features.

2. SIMULATED SURVEYS

Surveys are distributed to a larger group of stakeholders to gather quantitative data.

SAMPLE SURVEY FOR DOCTORS

1. **How often do you access patient records?**
    - ☐ Rarely
    - ☐ Occasionally
    - ☐ Frequently
    - ☑ Always
2. **What is the most critical feature for you in a PMS?**
    - ☑ Access to medical history
    - ☐ Appointment scheduling
    - ☐ Prescription management
3. **Do you prefer a desktop or tablet interface?**
    - ☑ Tablet
    - ☐ Desktop
    - ☐ Both

SAMPLE SURVEY FOR NURSES

1. **How often do you add or update patient records?**
    - ☐ Rarely
    - ☐ Occasionally
    - ☐ Frequently
    - ☑ Always
2. **What features are most important for your role?**
    - ☑ Adding new patient records
    - ☐ Managing billing
    - ☐ Scheduling appointments
3. **Would you benefit from training sessions on the system?**
    - ☐ Yes
    - ☑ No

SAMPLE SURVEY FOR PATIENTS

1. **Would you use an online portal to book appointments?**
    - ☑ Yes
    - ☐ No
2. **How would you like to receive appointment reminders?**
    - ☐ SMS
    - ☑ Email
    - ☐ Both
3. **Would you like access to your medical records?**
    - ☑ Yes
    - ☐ No

User Stories

- As a doctor, I want to access patient medical histories so that I can review relevant information
- As a doctor, I want to access patient medical histories so that I can review relevant information before consultations.
- As a patient, I want to book an appointment online so that I can schedule visits without calling the clinic.
- As a nurse, I want to add new patients to the system so that doctors have access to updated patient information.
- As an admin staff member, I want to generate invoices for patients so that billing is simplified and accurate.
- As a patient, I want to receive reminders about my appointments so that I don't miss them.

Patient Registration:

The system shall allow the receptionist or administrator to register a new patient by entering personal details, such as name, age, gender, contact information, and medical history.

- The system shall assign a unique patient ID to each registered patient for easy tracking of their medical records.
- The system shall validate the entered information to ensure completeness and accuracy.
- The system shall allow patients to update their personal information when necessary.

TABLE OF CONTENTS

1. **Requirement elicitation techniques**
   - Interviews
   - surveys
2. **User stories**
3. **Patient Registration**
4. **Introduction**
   - Product scope
   - Product value
   - Intended audience
   - Intended use
   - General description
5. **User characteristics**
6. **Product functions**
7. **Functional requirements**
8. **External interface requirements**
   - User interface requirements
   - Hardware interface requirements
   - Software interface requirements
   - Communication interface requirement
9. **Non-functional requirements**
   - Security
   - Capacity
   - Compatibility
   - Reliability
   - Scalability
   - Maintainability
   - Usability
   - Other non-functional requirements
10. **MOSCOW method**
    - **Software design**
    - **Class diagram**
    - **Sequence diagram**
    - **Collaboration diagram**
    - **Use case diagram**
11. **Operating environment**
12. **Definitions and acronyms**
13. **Literature Review**
14. **References**

1 Introduction

Describe the purpose of the document.

The purpose of this Software Requirements Specification (SRS) document is to clearly define the functional and non-functional requirements of the Patient Management System (PMS). It serves as a detailed reference for developers, stakeholders, and project managers, ensuring a shared understanding of the system's capabilities and constraints. The document outlines the system's scope, user needs, interface requirements, security considerations, and performance expectations, providing a foundation for design, implementation, and validation of the system.

1.1 Product scope

List the benefits, objectives, and goals of the product.

The Patient Management System (PMS) is designed to streamline patient record management, appointment scheduling, and data storage for healthcare providers. The goal is to enhance patient care efficiency by providing healthcare staff with reliable and quick access to relevant patient information.

1.2 Product value

Describe how the audience will find value in the product.

PMS simplifies the healthcare process by reducing paperwork, increasing data accuracy, and facilitating secure and easy access to patient records. It benefits medical staff by saving time, improving patient experience, and enhancing overall service quality.

1.3 Intended audience

Write who the product is intended to serve.

- Doctors
- Nurses and other medical staff
- Admin staff handling billing and patient records
- Patients interacting with the system to view records or manage appointments

1.4 Intended use

Describe how will the intended audience use this product.

This system is intended to:

- Enable healthcare staff to access, update, and manage patient records.
- Allow patients to book appointments and view their health information.
- Ensure secure handling and storage of sensitive medical data.

1.5 General description

Give a summary of the functions the software would perform and the features to be included.

The PMS application will provide a central database for storing patient information, an interface for scheduling appointments, and various functionalities tailored to each user role to enable efficient and secure healthcare service delivery.

USER CHARACTERISTICS FOR PATIENT MANAGEMENT SYSTEM (PMS)

The three main types of users for the PMS are:

- **Doctor**
- **Nurse**
- **Patient**
- **Admin Staff**

The following table describes general user characteristics that will affect the functionality of the PMS.

Type of User	User Characteristic	User Technical Expertise	How the user characteristic and technical expertise affect PMS functionality
Doctor	- Needs quick access to patient records for efficient decision-making. - Responsible for reviewing medical histories and updating treatment information.	- Moderate to high technical proficiency - Experience with EMR (Electronic Medical Records) systems or other digital tools in healthcare	- Requires an interface with rapid access to patient data. - Should have an intuitive layout for reviewing and updating records.
Nurse	- Focuses on recording patient vitals, administering medications, and assisting with registration. - Needs to quickly input new patient data.	- Moderate technical proficiency - Typically familiar with EMR systems	- Interface should support fast data entry and easy navigation between patient records. - Input forms need to be optimized for quick information capture.
Patient	- Uses the system mainly for booking appointments and viewing medical information. - Older patients may have limited technical skills, while younger patients may adapt to online interfaces easily.	- Varies by age group: - Younger users: high familiarity with web/mobile applications - Older users: limited experience, may need support	- User interface should be straightforward, with minimal steps to book appointments or view records. - Help features and prompts should be available, especially for older users.
Admin Staff	- Manages billing, patient invoicing, and appointment schedules. - Focuses on data accuracy and financial documentation.	- Moderate technical proficiency - Familiar with office and billing software	- Billing and scheduling modules should have straightforward workflows. - System should support secure data handling, with easy access to patient and billing information.

PRODUCT FUNCTIONS

The high-level summary of functions in Patient Management system (PMS) is described in the following concept map.

Detailed functional requirements will be described after this.

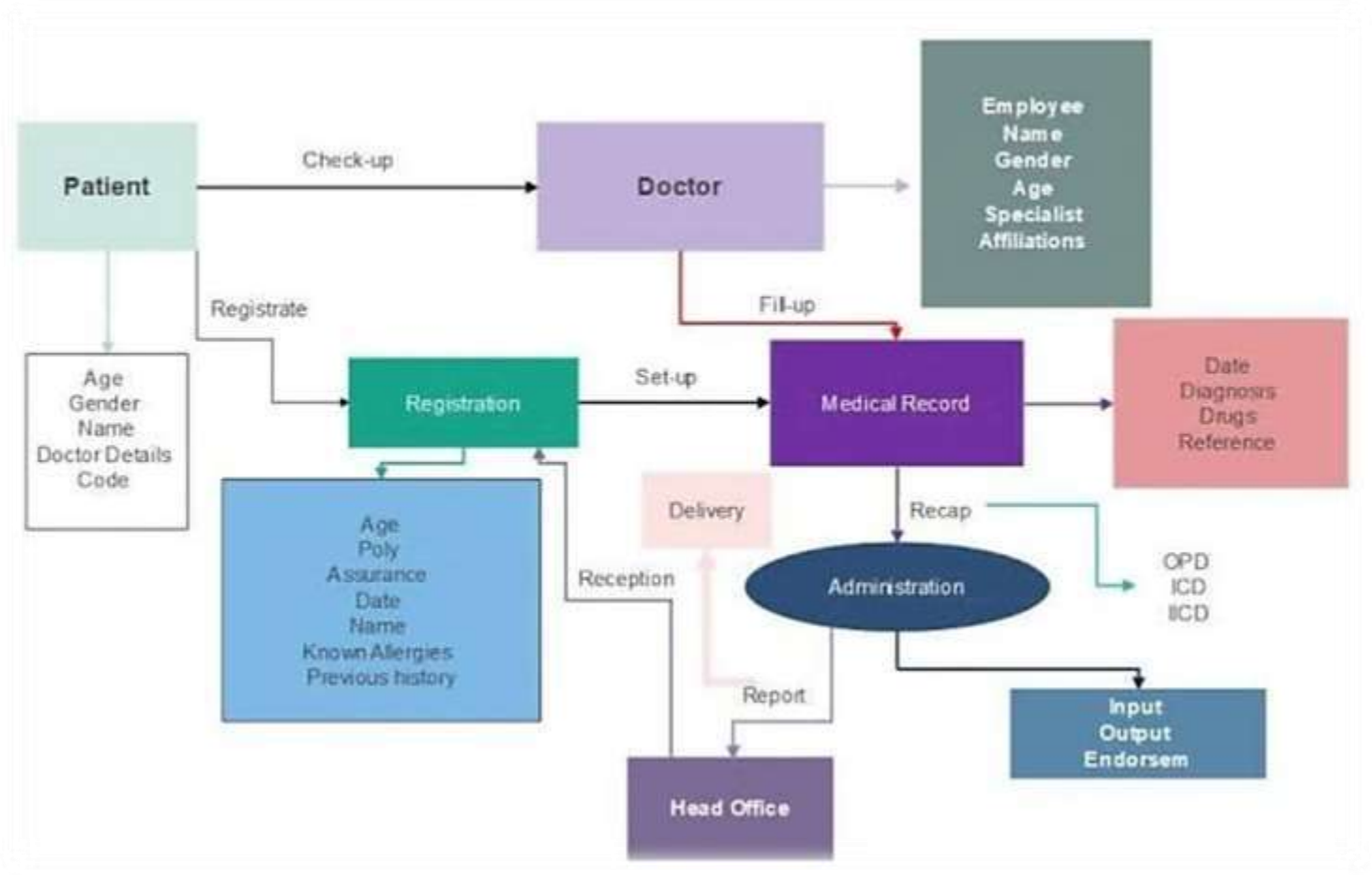


Figure 1: PATIENT MANAGEMENT SYSTEM

2 FUNCTIONAL REQUIREMENTS

List the design requirements, graphics requirements, operating system requirements, and constraints of the product.

Appointment Management	The system should allow patients to book, view, and cancel appointments. • **Use Case**: A patient books an appointment online by selecting a date, time, and doctor. The system confirms the appointment and sends a reminder to the patient. • **Priority**: MUST-HAVE • **Traceability**: This requirement aligns with the non-functional performance requirement that records should load within 2 seconds to ensure a smooth booking experience.
Medical Record Access:	Doctors can view and update patients' medical histories. • **Use Case:** A doctor accesses a patient's medical history via a search function. The patient's diagnosis, prescriptions, and visit history are displayed for the doctor's review. • **Priority:** MUST-HAVE • **Traceability:** Encrypted data transmission will secure patient medical histories
Patient Registration:	Admin staff can add new patient profiles with relevant details. • **Use Case:** A nurse registers a new patient by entering personal and medical details into the system. The system generates a unique patient ID and stores the information securely. • **Priority:** MUST-HAVE
Billing Management:	Admin staff can create and manage invoices for each patient. • **Use Case:** Admin staff generates an invoice based on the patient's medical visits, and the patient is notified of the total bill through email or SMS. • **Priority:** SHOULD-HAVE • **Traceability:** Integration with secure payment gateways for billing.

Document Storage:	The system can upload and store medical documents related to patient records. • **Use Case:** Doctors can upload documents such as X-rays, lab reports, or prescriptions into the system for secure storage. • **Priority:** COULD-HAVE
Notifications:	Patients receive reminders for upcoming appointments. • **Use Case:** The system sends automated reminders to patients about upcoming appointments via SMS or email. • **Priority:** SHOULD-HAVE
Doctor Dashboard:	Doctors can view daily appointments on a dashboard. • **Use Case**: A doctor's dashboard displays the day's appointments, ongoing treatments, and alerts for patients requiring immediate attention. • **Priority:** MUST-HAVE
Activity Logging:	User actions are logged to ensure accountability and security. • **Use Case**: The system logs every user interaction (e.g., access to medical records, billing actions) for accountability and auditing purposes. • **Priority**: COULD-HAVE

3 External interface requirements

3.1	**User interface requirements**	Describe the logic behind the interactions between the users and the software (screen layouts, style guides, etc.).

- **Interface Design:** Intuitive and accessible for various user roles, including healthcare staff and patients.
- **Accessibility:** Basic tutorials should be available, and the design should accommodate both desktop and mobile interfaces.

3.2	**Hardware interface requirements**	List the supported devices the software is intended to run on, the network requirements, and the communication protocols to be used.

- The system will require standard PCs or tablets in healthcare facilities for medical and admin staff.
- Patients can access their information on smartphones, tablets, or personal computers.

3.3	**Software interface requirements**	Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.

- Compatible with Windows, MacOS, and Linux operating systems.
- Requires integration with secure payment gateways (for billing).
- The system should interface with SMS or email services for sending notifications.

3.4	**Communication interface requirements**	List any requirements for the communication programs your product will use, like emails or embedded forms.

- **Database Connection:** Secure and encrypted connection to the central database.
- **External APIs:** For SMS/email notifications and payment processing.

4 NON-FUNCTIONAL REQUIREMENTS

4.1 **Security**	Include any privacy and data protection regulations that should be adhered to.

- All patient data must be encrypted in transit and at rest.
- Role-based access control to limit data access based on user type.

4.2 **Capacity**	Describe the current and future storage needs of your software.

- The system should handle up to 1,000 simultaneous users.

4.3 **Compatibility**	List the minimum hardware requirements for your software.

- Compatible with major operating systems and browsers.

4.4 **Reliability**	Calculate what the critical failure time of your product would be under normal usage.

- 99.9% uptime with regular data backups.

4.5 **Scalability**	Calculate the highest workloads under which your software will still perform as expected.

- Should support growth in the number of users, patient records, and healthcare facilities over time

4.6 **Maintainability**	Describe how continuous integration should be used to deploy features and bug fixes quickly.

- The system should be easy to maintain, with documentation for updates and a modular design for easy scaling.

4.7 **Usability**	Describe how easy it should be for end-users to use your software.

- User-friendly interface with clear navigation and accessible help features

4.8 **Other**	List any additional non-functional requirements.

- Performance: Records should load within 2 seconds for user convenience.

PRIORITIZE REQUIREMENTS USING THE MOSCOW METHOD

1. MUST-HAVE:

- Patient booking and viewing of appointments.
- Doctors' access to patient medical histories.
- Secure login and role-based access.

2. SHOULD-HAVE:

- Notifications for upcoming appointments.
- Admin management of billing and invoices.

3. COULD-HAVE:

- Document upload and storage for patient files.
- User activity logging for accountability.

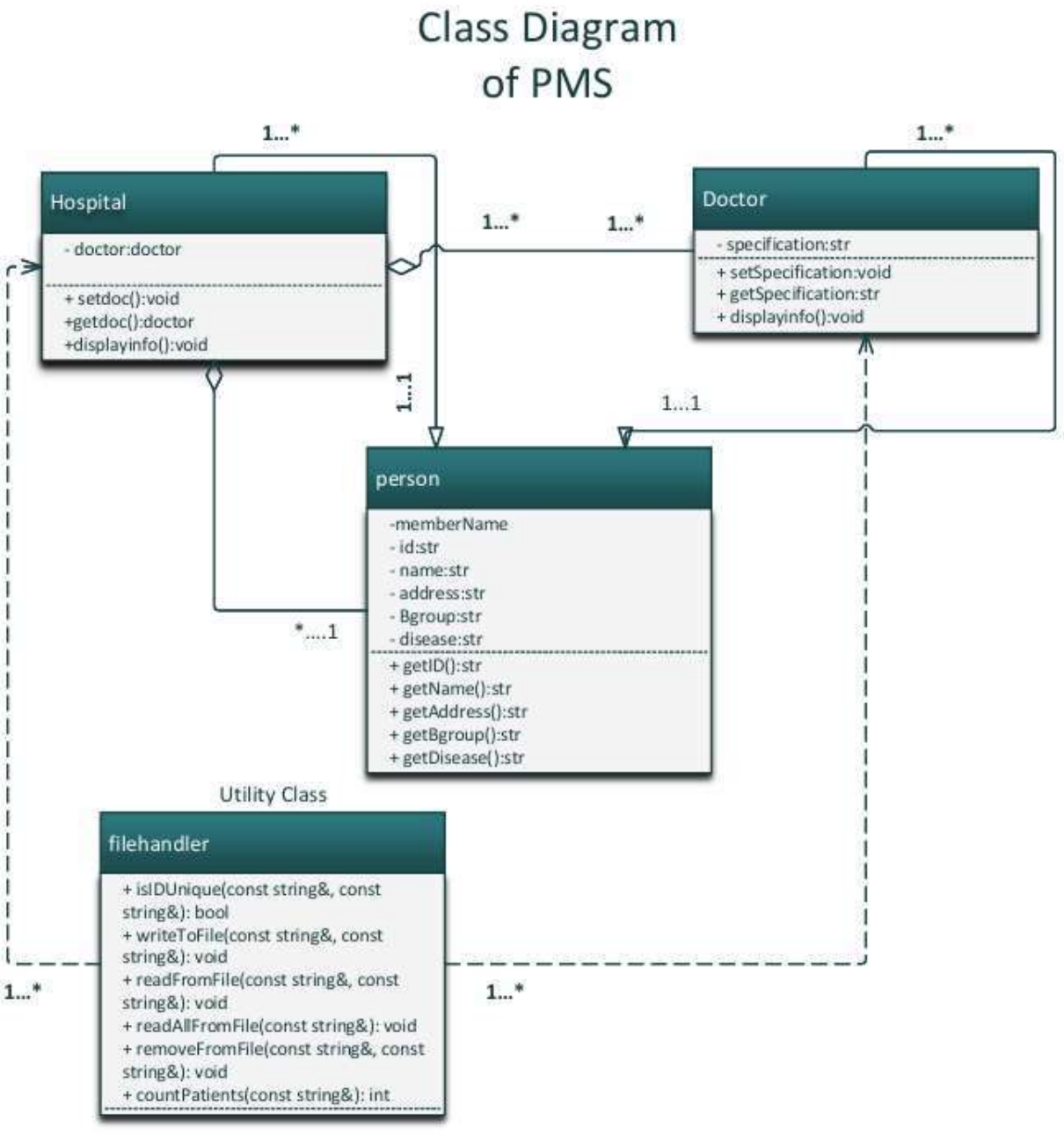
4. WON'T-HAVE (FOR THIS INITIAL VERSION):

- Advanced analytics on patient data (e.g., trends or health statistics).

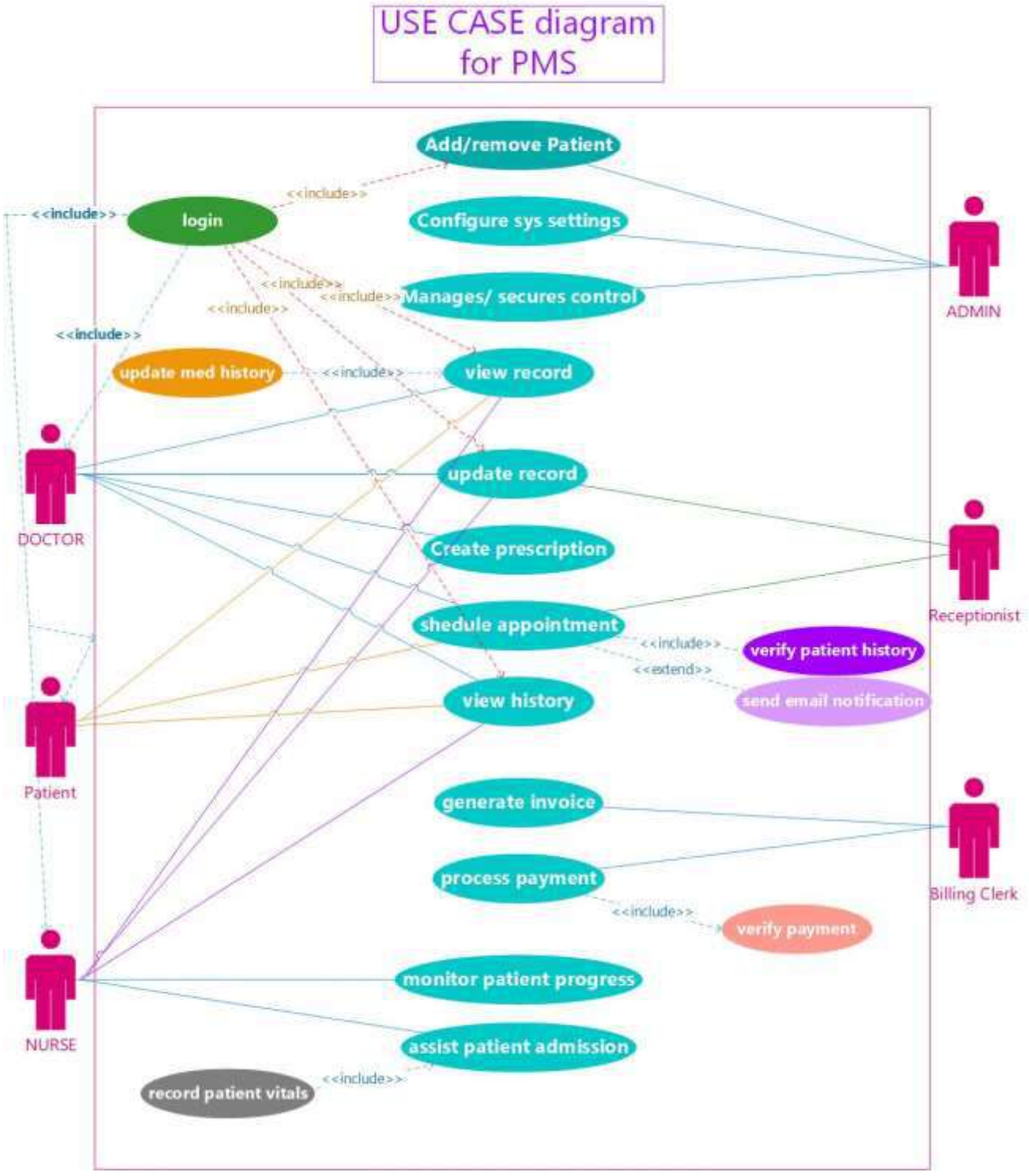
SOFTWARE DESIGN

1. Class diagram
2. Use case Diagram
3. Sequence Diagrams
4. Collaboration Diagram

CLASS DIAGRAM



USE CASE DIAGRAM



SEQUENCE DIAGRAMS

USE CASE NAME: ADMIN MANAGEMENT

Actor: Admin (Hospital Administrator or Super User)

Description:

The Admin is responsible for managing system settings, user roles (doctors, receptionists, patients), and other administrative functions within the Patient Management System.

Stakeholders and Interests:

Admin: Wants to manage user accounts, roles, and system settings effectively.

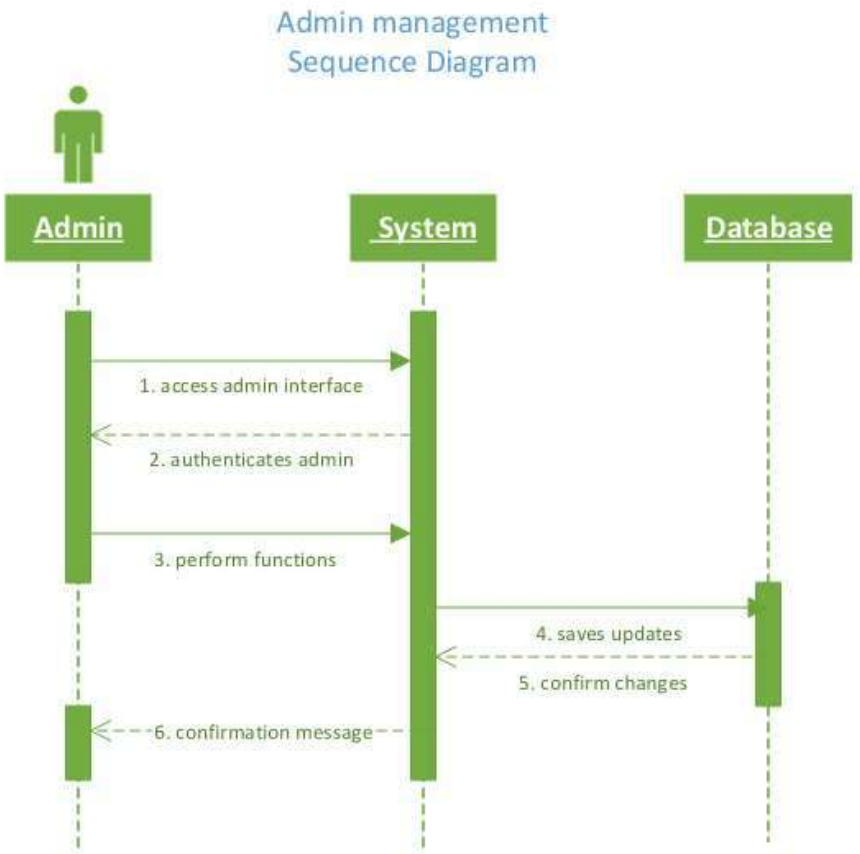
System: Ensures the proper update and management of user roles and system settings.

PRECONDITIONS:

Admin has login credentials and sufficient access rights to perform administrative tasks.

POSTCONDITIONS

User roles (doctors, receptionists, patients) and system settings are updated as needed.



USE CASE NAME : DOCTOR CONSULTATION

Actor(s): Doctor, Patient

Stakeholders and Interests:

Patient: Seeks consultation for diagnosis and treatment.

Doctor: Provides medical consultation based on the patient's symptoms and history.

PRECONDITIONS:

The patient has a scheduled appointment.

POSTCONDITIONS:

The doctor's consultation notes, diagnosis, and any prescribed treatments are recorded.

MAIN FLOW:

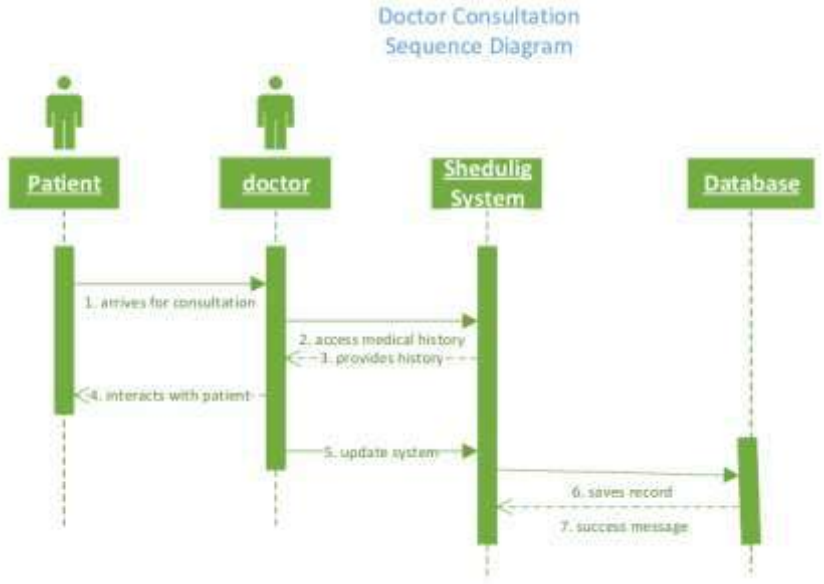
- Patient arrives for the consultation.
- Doctor conducts the consultation, asking questions about the patient’s symptoms and reviewing their medical history.
- Doctor provides a diagnosis and possible treatment options.
- System records the consultation details in the patient's medical record.

ALTERNATE FLOWS:

If the patient requires further testing, the doctor will order tests and update the medical record.

EXTENSIONS:

The doctor can prescribe medications or recommend further treatments.



USE CASE NAME: PATIENT LOGIN/AUTHENTICATION

Actor(s): Patient

Stakeholders and Interests:

Patient: Wants secure access to their profile and appointment information.

System: Ensures the security of patient data.

PRECONDITIONS:

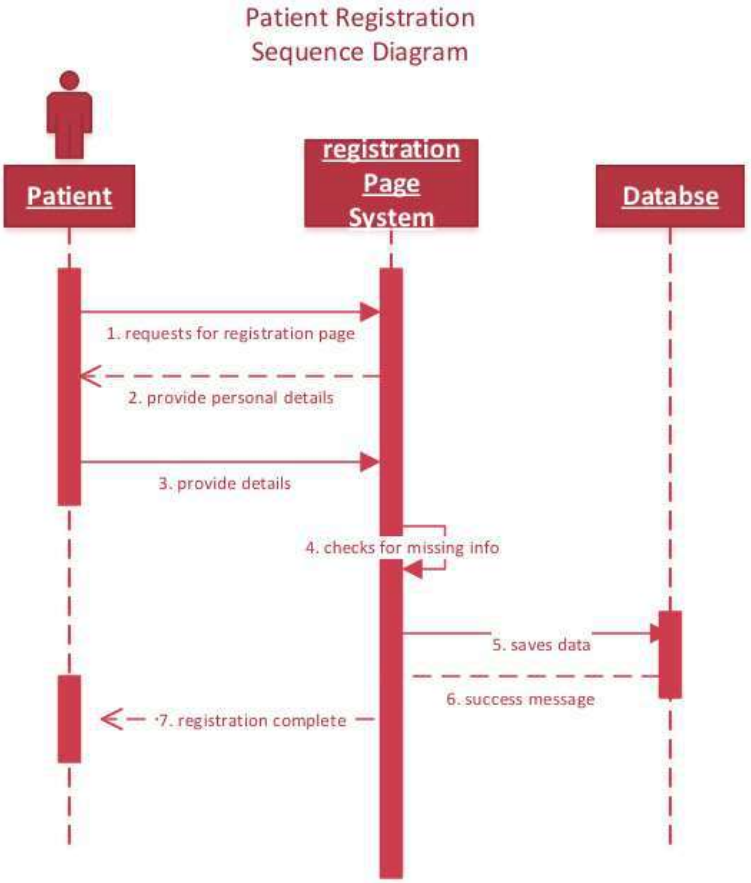
The patient is already registered in the system.

POSTCONDITIONS:

The patient is logged in and granted access to their profile.

MAIN FLOW:

- Patient enters login credentials (username/password).
- System validates credentials.
- If credentials are valid, System grants access to the patient's profile.
- System displays the dashboard with options like scheduling appointments, viewing medical history, etc.



USE CASE NAME: VIEW MEDICAL HISTORY

Actor(s): Patient, Doctor

Stakeholders and Interests:

Patient: Wants to access their past medical records.

Doctor: Needs access to patient history to make informed decisions during consultations.

PRECONDITIONS:

The patient has a stored medical history in the system.

POSTCONDITIONS:

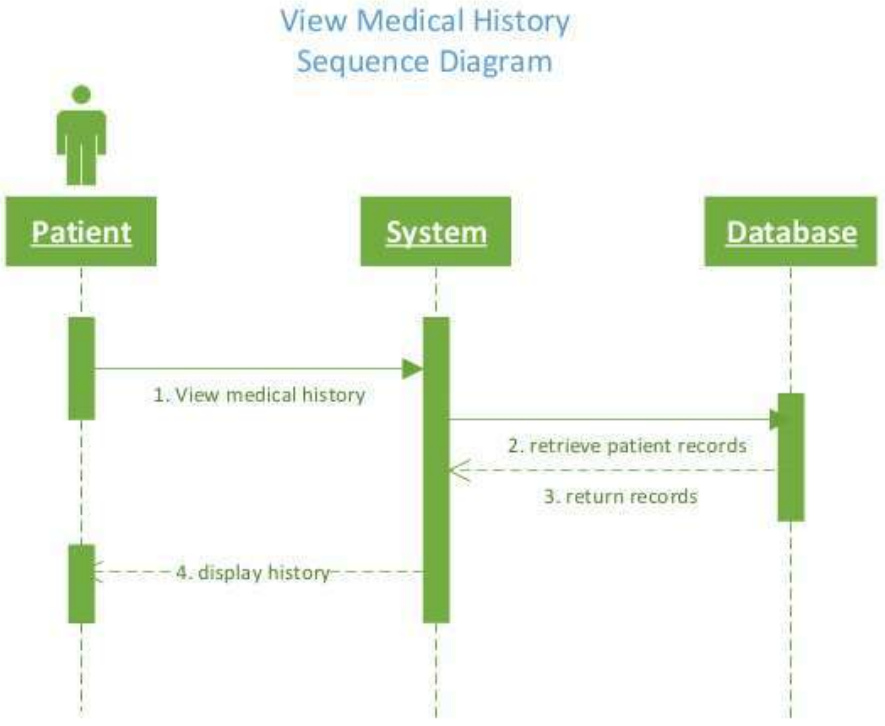
The system displays the patient's medical history, including past treatments, diagnoses, and prescriptions.

MAIN FLOW:

- Patient/Doctor selects the "View Medical History" option.
- System retrieves the patient's historical medical records from the database.
- System displays the patient's medical history, including dates of treatments, doctor names, and notes.

ALTENATE FLOWS:

If no medical history is found, the system informs the user.



USE CASE NAME: PROCESS PAYMENT

Actor(s): Billing Clerk, Patient

Stakeholders and Interests:

Patient: Wants to pay bills easily and securely.

Billing Clerk: Needs to ensure accurate billing and payment records.

PRECONDITIONS:

- The patient has received a bill for services.
- The billing system is functional and accessible.

POSTCONDITIONS:

- Payment is recorded in the system.
- A receipt is generated and provided to the patient.

MAIN FLOW:

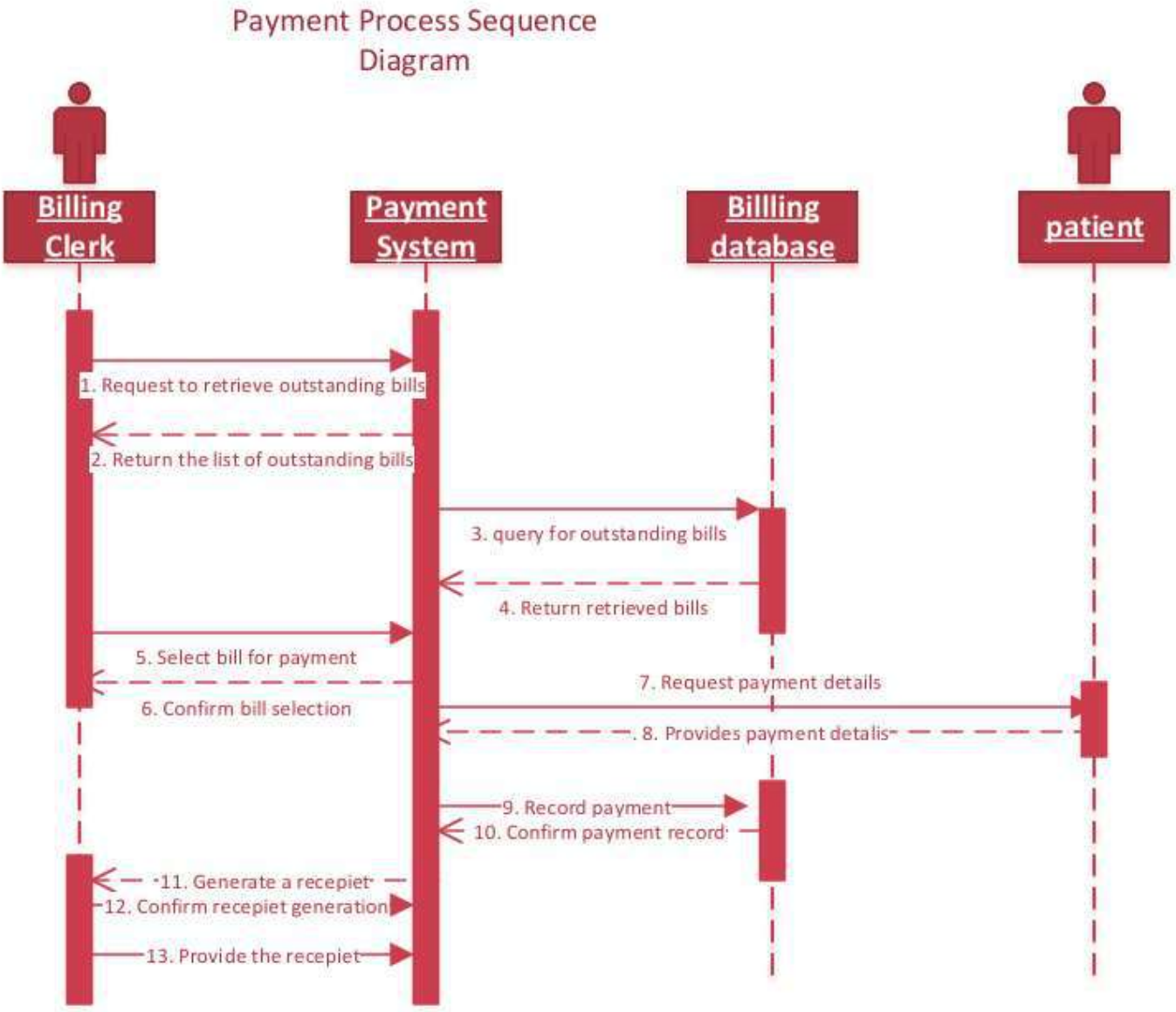
- Billing Clerk accesses the "Process Payment" interface.
- The system retrieves the patient's outstanding bills.
- Billing Clerk selects the bill(s) for payment.
- The system displays payment options (e.g., credit card, cash, insurance).
- Patient provides payment details.
- The system validates the payment method.
- The system processes the payment.
- A confirmation message is displayed, and a receipt is generated.
- Billing Clerk provides the receipt to the Patient.

ALTERNATE FLOW:

- If payment validation fails (e.g., declined credit card), the system prompts the Billing Clerk to re-enter the payment details or select an alternative payment method.

SPECIAL REQUIREMENTS:

- The system must process payments securely and comply with relevant regulations.
- Payment confirmation must be instant.



USE CASE NAME: SCHEDULE APPOINTMENT

Actor(s): Patient, Receptionist

Stakeholders and Interests:

Patient: Wants to schedule an appointment with a doctor at a convenient time.

Receptionist: Manages the scheduling of appointments and ensures accurate records.

PRECONDITIONS:

- The patient is logged into the system (for online scheduling).
- The receptionist has access to the scheduling system.
- Postconditions:
- An appointment is successfully scheduled and confirmed in the system.
- The patient receives a confirmation (either via email or SMS).
- The doctor's schedule is updated.

MAIN FLOW:

- Patient (or Receptionist, if the patient is unable to do it themselves) accesses the "Schedule Appointment" interface.
- The system requests the patient's details (name, contact info, etc.), or the receptionist enters them manually.
- The system displays available appointment slots for doctors.
- The Patient selects the preferred date and time from the available options.
- The Receptionist verifies the details and confirms the appointment (if the patient is scheduling in person).
- The system schedules the appointment and updates the doctor's calendar.
- The system sends a confirmation message to the Patient via email/SMS.
- The system updates the appointment records in the database.

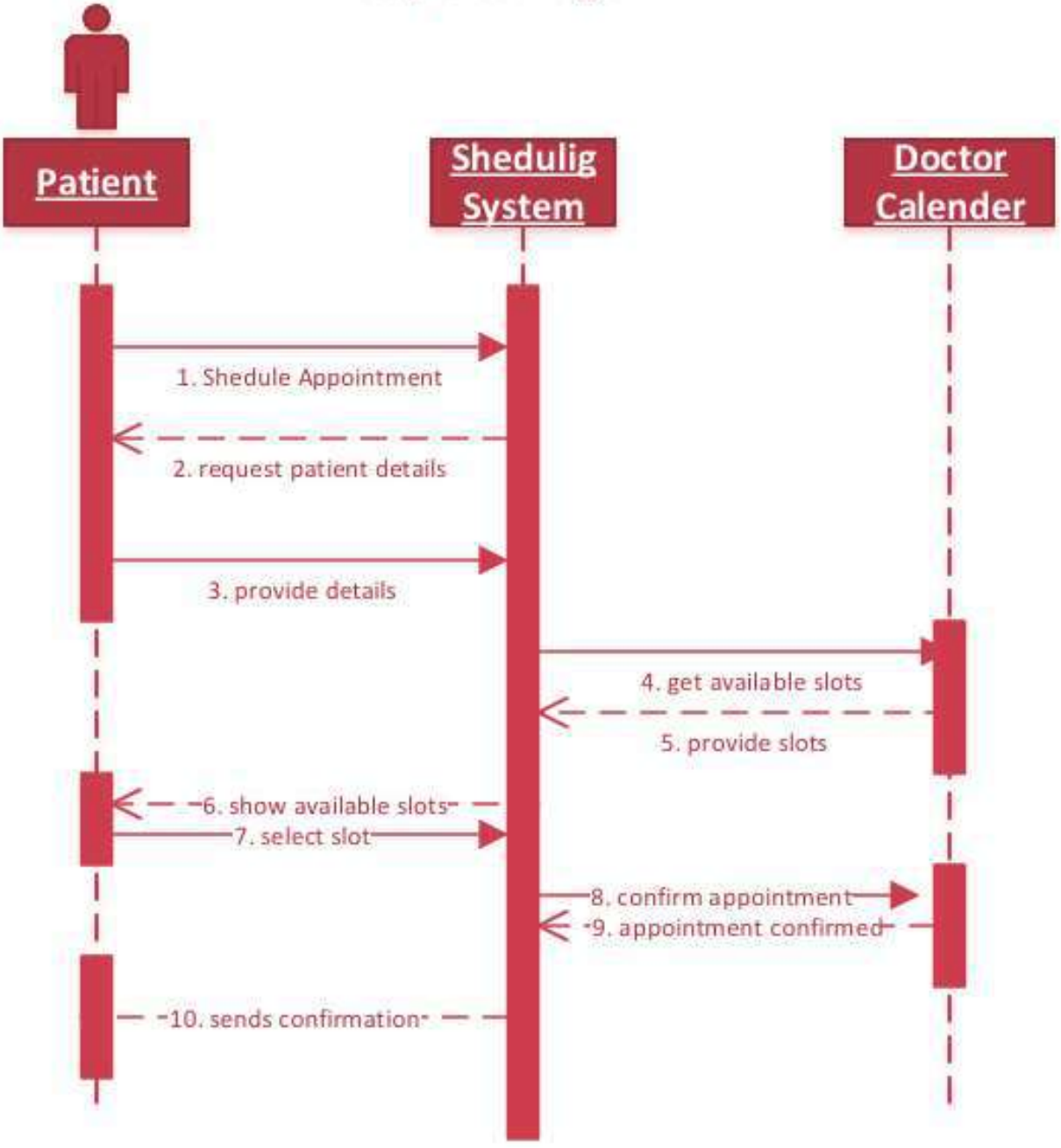
ALTERNATE FLOWS:

If no appointment slots are available, the system will suggest the next available date and time.

SPECIAL REQUIREMENTS:

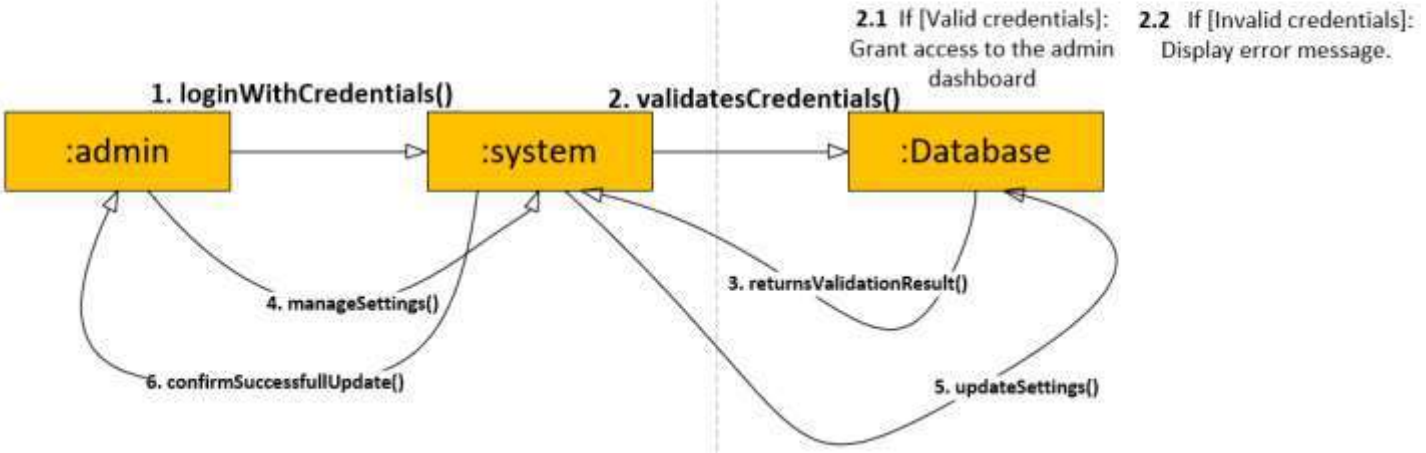
The appointment confirmation should be sent immediately after the appointment is confirmed.

Schedule Appointment
Sequence Diagram

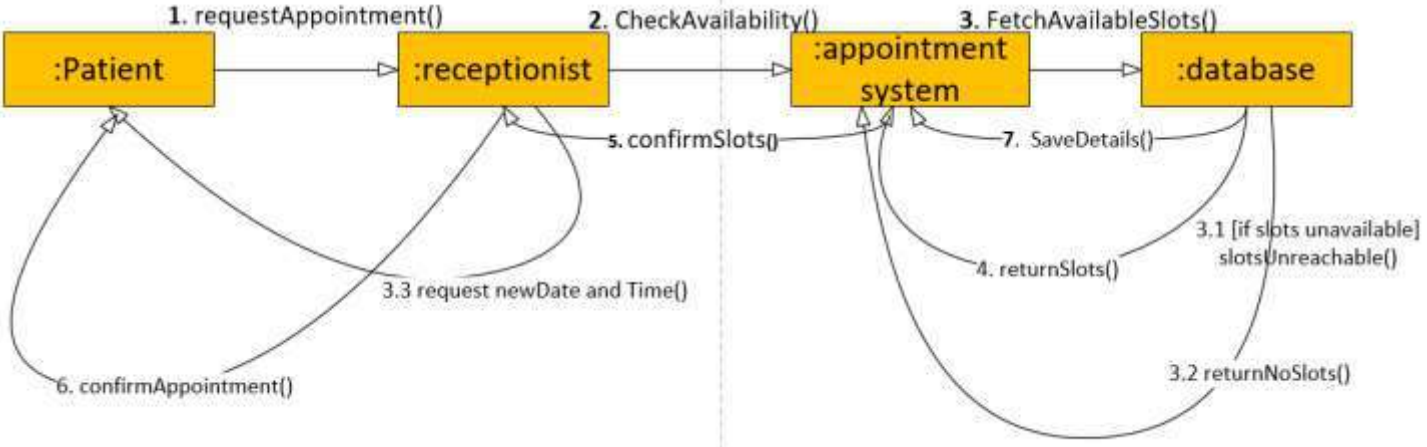


COLLABORATION DIAGRAMS

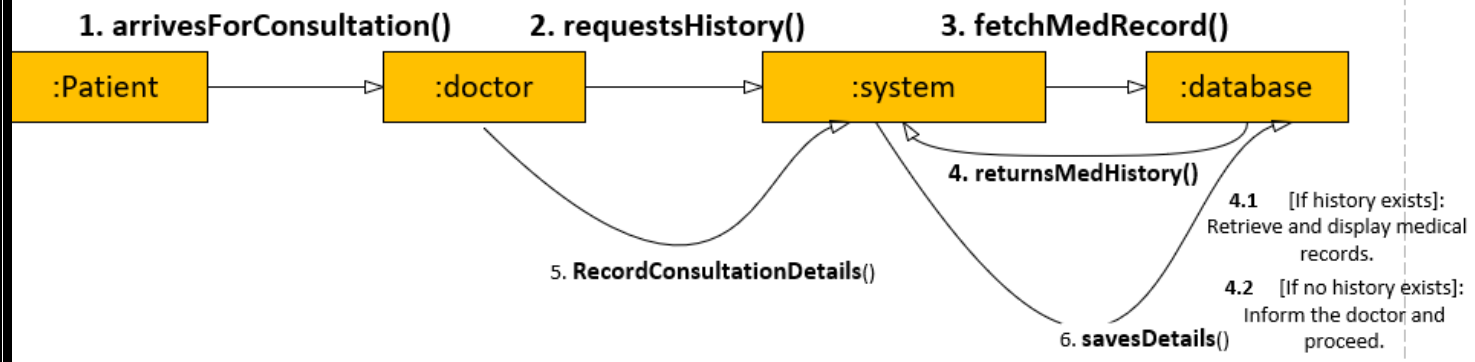
Collaboration Diagram for Admin Management()



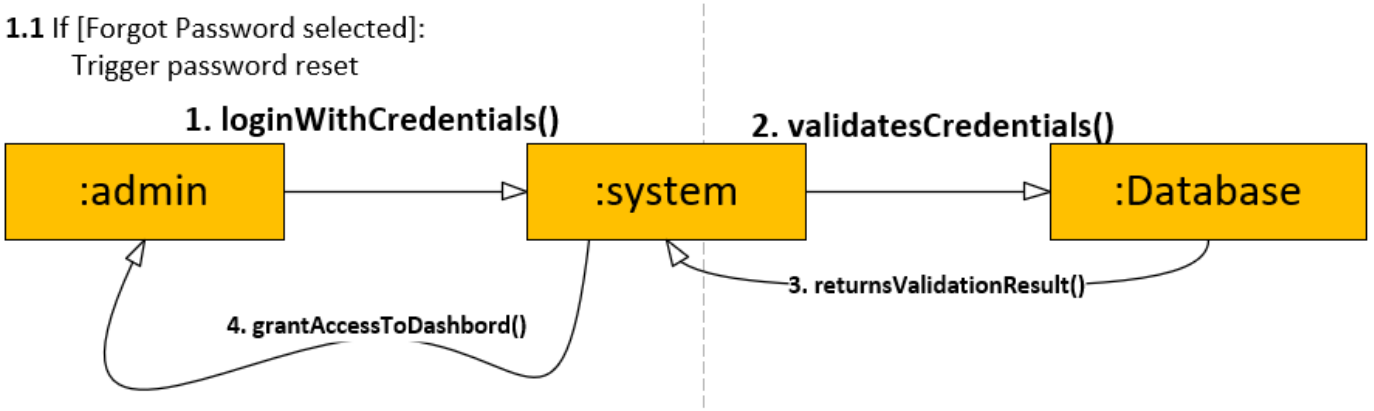
Collaboration Diagram for Appointment Sheduling



Collaboration Diagram for Doctor Consultation



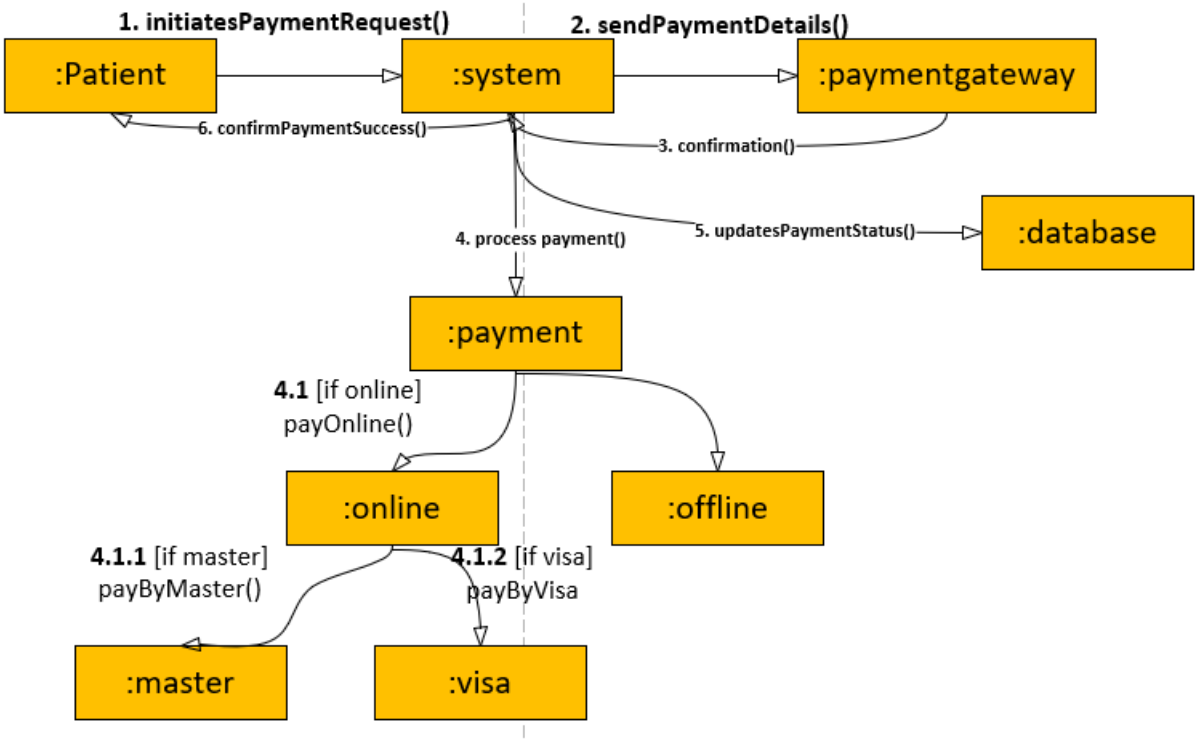
Collaboration Diagram for Patient Login



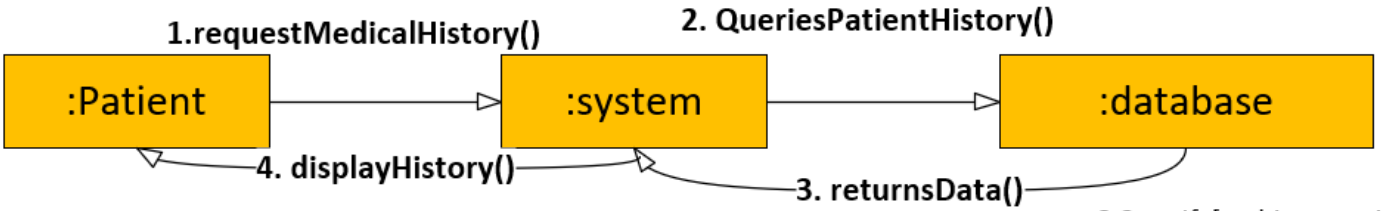
4.1 If [Valid credentials]: Grant access to the dashboard.

4.2 If [Invalid credentials]: Display error message.

Collaboration Diagram for Process Payment



Collaboration Diagram for View medical history



3.1 If [history exists]:
Display the retrieved history.

3.2 If [no history exists]:
Inform the patient with an error message.

OPERATING ENVIRONMENT

The PMS will operate in a hospital setting with the following specifications:

- **Hardware**:
    - Servers with Intel Core i5, 8 GB RAM, 500 GB SSD, and a minimum 10 Mbps internet connection.
- **Operating Systems**:
    - **Server**: Windows Server 2019/2022 or Linux (Ubuntu 20.04+)
    - **Client**: Web browsers (Chrome, Firefox, Edge), accessible on Windows 10/11, macOS, iOS, and Android.
- **Software Components**:
    - **Database**: MySQL or PostgreSQL
    - **Web Server**: Apache or Nginx
    - **Backend**: Java (JDK 11+) or Python (3.8+)
    - **Frontend**: ReactJS or Angular
    - **Security**: SSL/TLS encryption, OAuth 2.0 or JWT for authentication.
- **External Integrations**:
    - Payment gateways (e.g., PayPal, Stripe), hospital database, and email servers.

6 *Definitions and acronyms*

PMS:	Patient Management System
API:	Application Programming Interface
Encryption:	The process of encoding data to protect it from unauthorized access
Role-Based Access Control (RBAC):	A security protocol limiting access to certain data based on user roles

LITERATURE REVIEW FOR PATIENT MANAGEMENT SYSTEM (PMS)

Section	Details	✔/✘
Introduction	Identifies the purpose of analyzing existing systems for patient management.	✔
	Establishes the need for a modern, scalable, and secure PMS.	✔
	Guides system development based on gaps in existing solutions.	✔
Background and Related Work	**Hospital Management Systems (HMS):** Centralized data storage, but lacks real-time updates and multi-user support.	✘
	Electronic Health Record (EHR) Systems: Focus on digitization, but limited role-based access control for security.	✘
	Appointment Scheduling Applications: Simplifies bookings, but lacks integration with patient medical records.	✘
	Compliance and Privacy Systems: Ensures regulatory adherence (e.g., HIPAA), but operates in silos without system-wide integration.	✘
Gaps and Challenges	**Lack of integration:** Disjointed modules for scheduling, records, and billing management.	✘
	Scalability issues: Ineffective in supporting large healthcare networks.	✘
	Complex interfaces: Non-intuitive design for non-technical users.	✘
	Security concerns: Insufficient encryption and access control measures.	✘
	Limited interoperability: Challenges in data sharing across systems and departments.	✘
Relevance to Our Project	Addresses the need for an all-in-one system with integrated features.	✔
	Focuses on scalability for multi-branch hospitals.	✔
	Prioritizes robust security with role-based access and encryption.	✔
	Offers a user-friendly, responsive interface for ease of use.	✔
	Enhances interoperability across departments and systems.	✔
Conclusion	Highlights existing advancements and persistent gaps.	✔
	Demonstrates how the proposed PMS will bridge these gaps.	✔
	Justifies the need for a scalable, secure, and user-friendly solution tailored for modern healthcare needs.	✔

REFRENCES

https://dspmuranchi.ac.in/pdf/Blog/srs_template-ieee.pdf
https://www.scribd.com/document/468033245/srs-template-ieee
https://www.slideshare.net/slideshow/srs-for-hospital-management-system/20978192
https://asana.com/resources/software-requirement-document-template
Preparing Research Reports and Integrative Literature Reviews - IEEE Professional Communication Society