

Syrian Private University
Faculty of Computer & Information
Engineering.
Department of Software & Information
Systems Engineering.



Title

**Radiography
Management System**

إدارة مخبر أشعة

Prepare By

فاطمة الدرويش
صفا الذهري
ملاك برمود

Supervised by

المهندسة: رانيه رجب

Academic Year 2026\2025

Supervisor Certification

I certify that the preparation of the project entitled

**.....
prepared by**

**.....
was made under my supervision at Department of Computer and
Information Engineering – Faculty of Computer & Communication
Engineering in partial fulfilment of the Requirements for the
Degree**

**of Bachelor of Software and information System Engineering.
Academic Rank:..... Name:.....**

Signature:

Date:

Abstract

This intelligent cloud-based radiology system represents a qualitative leap in laboratory and diagnostic services by integrating advanced artificial intelligence with secure cloud storage. The platform significantly enhances the overall performance and quality of healthcare services, ensures high-level data security and instant accessibility to sensitive patient information and images from anywhere, and accelerates consultation processes through real-time sharing among specialists and consultants. The AI component provides ultra-fast and highly accurate analysis of radiological images, offers early diagnostic insights to physicians, reduces human errors, and supports faster and more accurate clinical decision-making. Ultimately, the system improves patient outcomes, shortens waiting times, and delivers a smoother, more precise, and patient-centred healthcare experience.

الملخص

يمثل هذا النظام الذكي السحابي للأشعة فزعة نوعية في خدمات المختبرات والتشخيص من خلال دمج الذكاء الاصطناعي المتقدم مع التخزين السحابي الآمن. يعمل النظام على تحسين جودة وكفاءة الخدمات الصحية بشكل كبير، وضمان أمان البيانات الحساسة وسهولة الوصول الفوري إلى معلومات وصور المرضى من أي مكان، وتسريع عمليات الاستشارة والتشاور بين الأخصائيين والأطباء الاستشاريين. يقوم الذكاء الاصطناعي بتحليل الصور الشعاعية بسرعة ودقة فائقة، ويقدم رؤى تشخيصية مبكرة للأطباء، ويساهم في تقليل الأخطاء البشرية وتسريع اتخاذ القرارات العلاجية الصحيحة. في النهاية، يؤدي النظام إلى تحسين نتائج المرضي، تقليل أوقات الانتظار، وتقديم تجربة رعاية صحية أكثر سلاسة ودقة وتركيزًا على المريض.

Table of Contents

Chapter / Section	Title / Subtitle	Page
Chapter 1	Introduction and Definition of the Project	1
1	Solve The Problem	2
2	Aim of the Project	2
3	Documentation Description	3
4	Project's Management	3
4.1	Human Resource	3
4.2	Programming Resource	3
4.2.1	Bootstrap 5 (Front-End-Framework)	3
4.2.2	XAMPP Local Host	3
4.3	Technologies Used	4
4.3.1	HTML5	4
4.3.2	CSS	4
4.3.3	JavaScript	4
4.3.4	SMTP	4
4.3.5	PHP	4
4.4	Major Task (Incremental Model)	4
Chapter 2	Theoretical Study	6
	Introduction	7
1	What is an Intelligent Cloud-Based Radiology System?	7
2	Key Features of Our Intelligent Radiology System	7
2.1	Cloud-Based Image Storage	7
2.2	Real-Time Collaborative Consultation	7
2.3	AI-Powered Image Analysis	7
2.4	Early Diagnostic Support	7
3	Benefits of Intelligent Cloud-Based Radiology Platforms	8

3.1	Enhanced Diagnostic Speed & Efficiency	8
3.2	Improved Accuracy & Reduced Errors	8
3.3	Better Collaboration & Knowledge Sharing	8
3.4	Patient-Centered Outcomes & Accessibility	8
	Similar Systems	9
1.1	Legacy / On-Premises Radiology Systems	9
1.2	Electronic Health Record / Electronic Medical Record Systems (EHR/EMR)	9
1.3	Radiology Information System (RIS)	9
	In Summary — Why Our System Stands Out	10
	Comparison of similar Systems	11
Chapter 3	Analytical and Design Study	12
	Introduction	13
1	Requirements Gathering Approach	13
2	Project Requirements	13
2.1	Functional Requirements	13
2.2	Non-Functional Requirements	14
	Requirements Tracing Matrix	16
3	Analytical Chart of the Project	19
3.1	Use-Case Diagram	19
3.2	Use-case Narrative	21
3.3	Class Diagram	30
3.4	ERD-Diagram	31
3.5	Site Maps of The System	32
3.6	Gantt Chart	32
3.7	Activity Diagrams	33
3.8	Sequence Diagrams	35
4	Database	42
Chapter 4	Implementation and Testing of Project	43
1	Log-In	44

2	(Admin) Dashboard	44
3	(Employee) Appointment	45
4	(Doctor) My appointment	45
	Testing	49
Chapter 5	Conclusion and Next Stages	51
	Conclusion	52
Chapter 6	References and Appendices	54
6	References	55
Appendix A	PHP Programming Language	56
Appendix B	MySQL Database Management System	56

Table of figure

Figure No.	Title / Description	Page
Figure 1	Use-Case Diagram high level	19
Figure 2	Use-Case Diagram low level	20
Figure 3	Class Diagram	30
Figure 4	ERD-Diagram	31
Figure 5	Site Map-Diagram	32
Figure 6	Gantt Chart	32
Figure 7	Activity Diagram (Actor 1)	33
Figure 8	Activity Diagram (Actor 2)	33
Figure 9	Activity Diagram (Actor 3)	34
Figure 10	Activity Diagram (Actor 4)	34
Figure 11	Sequence Diagram (use cases 1,2)	35
Figure 12	Sequence Diagram (use cases 3,4)	35
Figure 13	Sequence Diagram (use cases 5,6)	36
Figure 14	Sequence Diagram (use cases 7,8)	36
Figure 15	Sequence Diagram (use cases 9,10)	37
Figure 16	Sequence Diagram (use cases 11,12)	37
Figure 17	Sequence Diagram (use cases 13,14)	38
Figure 18	Sequence Diagram (use cases 15,16)	38
Figure 19	Sequence Diagram (use cases 17,18)	39
Figure 20	Sequence Diagram (use cases 19,20)	39
Figure 21	Sequence Diagram (use cases 21,22)	40
Figure 22	Sequence Diagram (use cases 23,24)	40
Figure 23	Sequence Diagram (use cases 25,26)	41
Figure 24	Sequence Diagram (use cases 27,28)	41
Figure 25	Database	42
Figure 26	Log-In	44
Figure 27	Admin Dashboard	44

Figure 28	Appointment	45
Figure 29	my Appointments	45
Figure 30	levels.php	46
Figure 31	Unauthorized Access Page	46
Figure 32	Error Page	47
Figure 33	autoload.php	47
Figure 34	conf.php	48
Figure 35	db.class.php	48

List of tables

Table No.	Title / Description	Page
Table 1	Comparison of Similar Systems	11
Table 2	Requirements Tracing Matrix (RTM)	16-17-18
Table 3–22	Use Case Narratives (each use case in separate table)	21–29
Table 23	Test Cases	49-50

Chapter1

Introduction and Definition of the Project

1) Solve The Problem:

In light of the rapid advancements in medical imaging technologies and the growing demand for efficient, accurate, and accessible diagnostic services worldwide, many traditional radiology laboratories still rely heavily on local servers, manual image processing, paper-based reports, and face-to-face consultations. This creates several critical challenges: long turnaround times for image analysis, limited real-time collaboration between radiologists and specialists, difficulty in providing fast second opinions, vulnerability to data loss or restricted access in remote or underserved areas, higher risk of human error in interpretation, and delays in patient diagnosis and treatment.

The problem we aim to solve is to provide a modern, intelligent, cloud-based radiology system that integrates advanced artificial intelligence with secure, accessible cloud infrastructure. This system offers a fast, accurate, and collaborative diagnostic platform that radiologists, specialists, physicians, and healthcare administrators can use seamlessly from any location, eliminating the limitations of traditional methods and significantly improving the speed, precision, and quality of radiological services.

2) Aim of the Project:

The project aims to deliver a comprehensive, intelligent, cloud-based radiology platform that revolutionizes diagnostic imaging services through the integration of artificial intelligence and secure cloud technology.

The platform is designed to achieve the following key objectives:

Constant availability and accessibility — allowing radiologists, consultants, physicians, and administrators to access patient data, radiological images, and AI-assisted analysis anytime and anywhere using computers, tablets, or smartphones, without geographical or time restrictions.

Exceptional ease of use and flexibility — through:

Intuitive, professional, and well-organized interfaces designed for medical professionals

Secure, user-friendly dashboards tailored for radiologists, specialists, lab administrators, and referring physicians

Integration of artificial intelligence for advanced diagnostics — using AI algorithms to provide ultra-fast and highly accurate analysis of radiological images (X-rays, CT, MRI, etc.), early diagnostic insights, anomaly detection, and preliminary reports to support radiologists and reduce interpretation time.

Effortless and continuous collaboration — enabling real-time sharing of images and reports among specialists and consultants worldwide, accelerating second opinions, multidisciplinary consultations, and peer review without delays.

Comprehensive data security, storage, and monitoring — providing cloud-based storage with high-level encryption, role-based access control, audit trails, and detailed performance reports (processing time, accuracy rates, case volumes), along with automated notifications to support effective decision-making and quality assurance.

Ultimately, the project seeks to build a complete, smart, secure, sustainable, and scalable digital radiology ecosystem that transforms diagnostic imaging into a faster, more accurate, and truly patient-centered service — regardless of location, resource availability, or traditional infrastructure limitations. By reducing human error, shortening waiting times, and improving diagnostic outcomes, the system enhances overall patient care quality and healthcare efficiency.

3) Documentation Description:

Project Study: Defining project objectives and development method.

Theoretical study of the project: The project was studied in theory and then applied to the practical side of the implementation.

Analytical and Design Study of the Project: Includes a detailed description of the project components and how to analyze, design, investigate, and test these components.

4) Project's Management:

4.1 Human Resource: The working group includes students/researchers from the Faculty of Computer and Information Engineering – Department of Software & Information Systems Engineering (or relevant department) at Syrian Private University.

4.2 Programming Resource:

4.2.1 Bootstrap 5 (Front-End Framework): It is one of the most popular and powerful open-source libraries that combines HTML, CSS, and JavaScript to build responsive pages and applications. It is especially useful for creating websites compatible with various screen sizes, including mobile devices.

We relied on this framework in our project because:

Great flexibility and ease of use.

Designed to be compatible with all devices.

Compatible with all modern browsers.

Offers stable and reusable design components.

Can be learned easily and quickly.

Excellent scalability using JavaScript.

Support for jQuery (when needed).

Comprehensive JavaScript API support.

Can be used in any editor or programming environment.

4.2.2 XAMPP Local Host: The project was developed and tested using the local server environment (Apache Web Server) with the following features:

Free and open source — widely adopted with a large community, regular updates, and fast support.

Supports embedded services including:

X: Multi-platform support.

A: Apache HTTP Server.

M: MariaDB (formerly MySQL-compatible).

P: PHP programming language.

P: Perl programming language (optional).

Multi-platform: Works on Windows, Linux, macOS.

Easy to install and use — comes as a single package with automatic setup, no manual configuration required.

4.3 Technologies Used:

4.3.1 HTML5: Hypertext Markup Language (HTML5) allows the creation and structuring of web content using elements, tags, and attributes. While not a programming language, it forms the foundation for building structured, semantic web pages.

4.3.2 CSS: Cascading Style Sheets (CSS) is a design language used to control the presentation of web pages — including colors, fonts, spacing, layouts, backgrounds, and responsive behavior across devices. CSS works in harmony with HTML to create visually appealing and user-friendly interfaces.

4.3.3 JavaScript: JavaScript is a versatile, text-based programming language used for both client-side and server-side development. It adds interactivity to web pages, enabling dynamic features such as real-time image viewing, notifications, dashboard updates, and AI result rendering.

4.3.4 SMTP (Email Service): Simple Mail Transfer Protocol (SMTP) is used to send automated notifications and reports (e.g., critical findings alerts, consultation requests, or result delivery confirmations) via email to physicians, specialists, or administrators.

4.3.5 PHP: PHP (Hypertext Preprocessor) is a widely used open-source server-side scripting language ideal for building dynamic web applications. It powers the backend of our platform — handling secure data processing, database interactions, user authentication, AI result integration, report generation, and real-time communication.

4.4 Major Task: The project was developed using the Incremental Model, where the system is built and delivered in small, functional increments (iterations). Each increment adds new features and improvements, allowing early delivery of working parts, continuous feedback, and progressive risk reduction.

The project was structured through several incremental phases, with the following key activities repeated and refined in each iteration:

Planning Phase (for each increment):

Define objectives and scope of the current increment

Prioritize features (e.g., core image upload → AI analysis → secure sharing → reporting)

Estimate time, effort, and resources

Plan feedback sessions with stakeholders (radiologists, specialists, hospital IT)

Requirements Analysis & Specification (for each increment):

Gather and refine detailed requirements

Create prototypes/mockups (especially for diagnostic interfaces and dashboards)

Validate with radiologists, physicians, and potential users

Design & Development (Engineering):

Design architecture and user interfaces for the increment

Develop features (e.g., image upload, AI integration, cloud storage, real-time alerts)

Write clean, modular, testable code

Perform unit and integration testing

Testing & Integration:

Conduct functional, usability, performance, and security testing

Integrate new features with previous increments

Perform regression testing

Evaluation & Feedback:

Deliver working increment to stakeholders (radiologists, specialists)

Collect feedback on accuracy, speed, usability, and diagnostic support

Identify risks (e.g., performance, data security)

Adjust plan for next increment

Deployment & Release (after selected increments):

Deploy stable increments (e.g., initial image storage → full AI analysis → collaboration features)

Provide early access for validation in clinical settings

Main Advantages Realized by Using the Incremental Model:

Early delivery of usable components (e.g., secure image upload first)

Continuous feedback from radiologists and specialists → better alignment with real diagnostic needs

Reduced risk through small, manageable increments

Easier adaptation to changing requirements (new AI models, regulations, or clinical protocols)

Higher stakeholder satisfaction due to visible progress throughout development

This incremental approach was particularly suitable for an AI-powered, cloud-based radiology system, as it allowed step-by-step validation of diagnostic accuracy, speed, security, and usability with real medical professionals from the very beginning.

Chapter 2

Theoretical Study

Introduction

Gone are the days when radiologists and physicians had to rely solely on slow, manual analysis of X-rays, CT scans, and MRIs, often leading to long waiting times, diagnostic delays, and increased risk of human error. With the rapid advancement of artificial intelligence and cloud computing in healthcare, radiology services have become far more efficient, accurate, and accessible. Intelligent cloud-based radiology platforms represent one of the most important innovations in modern diagnostic medicine, transforming image interpretation into a faster, more precise, and collaborative process.

1) What is an Intelligent Cloud-Based Radiology System?

An intelligent cloud-based radiology system is a digital platform that integrates advanced artificial intelligence with secure cloud storage and real-time collaboration tools. The goal is to make radiological image analysis feel seamless and assisted rather than time-consuming and error-prone — thereby boosting diagnostic speed, accuracy, and collaboration among healthcare professionals while improving patient outcomes and reducing workload stress.

2) Key Features of Our Intelligent Radiology System:

Our platform, designed for modern laboratories, hospitals, and radiology centres, incorporates several powerful AI-driven and cloud-based elements:

2.1 Cloud-Based Image Storage & Instant Access Secure, encrypted cloud storage of all patient images and reports, allowing radiologists, consultants, and specialists to access sensitive data instantly from any location or device.

2.2 Real-Time Collaborative Consultation Immediate sharing of cases with consulting physicians and specialists worldwide, enabling fast second opinions, multidisciplinary discussions, and faster treatment decisions.

2.3 AI-Powered Image Analysis Advanced artificial intelligence algorithms that automatically detect abnormalities, highlight regions of interest, classify findings (e.g., fractures, tumors, infections), and provide quantitative measurements with high speed and precision.

2.4 Early Diagnostic Support & Error Reduction The AI acts as a smart assistant, offering preliminary insights, flagging potential issues early, and helping reduce human oversight while supporting faster, more confident clinical decisions.

3) Benefits of Intelligent Cloud-Based Radiology Platforms

The advantages of this approach are numerous, especially for healthcare providers and patients. Here are the key benefits aligned with our system's focus:

3.1 Enhanced Diagnostic Speed & Efficiency

Dramatically reduces analysis time from hours to minutes

Accelerates consultation and second-opinion processes through instant cloud access

3.2 Improved Accuracy & Reduced Errors

AI provides objective, consistent analysis, minimizing human fatigue-related mistakes

Early detection of subtle findings improves diagnostic confidence and precision

3.3 Better Collaboration & Knowledge Sharing

Enables real-time expert input from anywhere, supporting remote and underserved areas

Facilitates multidisciplinary teamwork and continuous learning among radiologists

3.4 Patient-Centered Outcomes & Accessibility

Significantly shortens patient waiting times for results and treatment planning

Improves overall quality of care, patient safety, and satisfaction through faster, more accurate diagnoses

Makes high-level radiological expertise accessible regardless of geographic location

In a Nutshell: Don't think of investing in AI-powered cloud radiology systems as an unnecessary expense or complex technology burden. Instead, view them as powerful tools that deliver real clinical value through speed, precision, and collaboration. Choose an intelligent platform that matches your laboratory's needs — it can help reduce diagnostic delays, minimize errors, improve patient outcomes, and elevate the standard of care without compromising safety or efficiency. In the era of modern medicine, AI-assisted cloud radiology is an investment in faster, smarter, and more compassionate healthcare!

Similar Systems / Competing Platforms:

Here is the corrected and adapted section on similar systems, rewritten to fit the context of your **intelligent cloud-based radiology system** (the AI-powered, cloud radiology platform). The comparison now highlights your system's unique strengths — such as specialized AI image analysis, full cloud accessibility, real-time collaboration, advanced diagnostic support, and centralized radiology center management — compared to legacy/on-premise systems, traditional Electronic Health Record (EHR/EMR) systems, and standard Radiology Information Systems (RIS).

1.1 Legacy / On-Premises Radiology Systems These are traditional, locally installed radiology software solutions (often older hospital information systems or basic PACS/RIS deployments running on local servers).

Key observations:

- Limited to on-site access only (no cloud mobility — doctors/specialists must be physically present or use VPN).
- No built-in advanced AI tools for automated image analysis or early diagnostic support.
- Heavy reliance on paper-based workflows or manual data entry in many cases.
- Difficult and expensive to scale, update, or integrate with modern multi-device access.
- No real-time collaboration features for remote consultants.

1.2 Electronic Health Record / Electronic Medical Record

Systems (EHR/EMR) Popular EHR/EMR platforms (e.g., Epic, Cerner, Meditech, or local systems) are designed for overall patient record management, including some integration with radiology results.

Key features:

- Strong patient data management and integration with other hospital systems.
- Good tracking of patient history and follow-up.
- Usually include basic viewing of radiology reports/images.

Limitations compared to our system:

- No specialized, dedicated radiology center management dashboard.
- Limited or no built-in advanced AI image analysis (mostly manual interpretation).
- Collaboration is slow (no instant cloud-based specialist sharing).
- Often require additional modules or third-party PACS/RIS for full radiology functionality.
- Not optimized for ultra-fast, AI-assisted diagnostic workflows.

1.3 Radiology Information System (RIS) Standard RIS platforms (e.g., Merge RIS, MedInformatix, or local RIS solutions) focus mainly on radiology workflow: scheduling, reporting, billing, and basic image management.

Key features:

- Efficient handling of radiology scheduling, reporting, and billing.
- Integration with PACS for image storage/viewing.

- Good reduction of paperwork in radiology departments.
Limitations compared to our system:
 - No advanced AI diagnostic support or automated image analysis.
 - Often lack full cloud-native mobility and multi-device instant access.
 - Limited real-time collaboration with external consultants.
 - No centralized, intelligent radiology center management with AI insights.
 - Less emphasis on early diagnostic alerts and error reduction via AI.
- In Summary — Why Our System Stands Out:** Our intelligent cloud-based radiology platform combines the best aspects of legacy systems (structured workflow), EHR/EMR (patient data integration), and standard RIS (radiology-specific operations) while delivering unique, game-changing advantages:
- **Specialized radiology center management dashboard** — a complete, centralized solution tailored for radiology labs and departments.
 - **Full cloud-native architecture** — secure, instant access from any device, anywhere, with real-time collaboration among specialists.
 - **Advanced AI diagnostic tools** — ultra-fast, accurate image analysis, early detection alerts, and significant reduction in human error.
 - **Patient-centred improvements** — dramatically shorter waiting times, better follow-up, and higher diagnostic confidence.
 - **Modern accessibility & scalability** — multi-device support, no heavy on-premise infrastructure, and sustainable growth without constant hardware upgrades.
- This positions our platform as a next-generation, AI-powered radiology solution that truly bridges the gap between traditional systems and the future of smart, collaborative, and precise diagnostic healthcare.

Comparison of Similar Systems:

Our System	(Legacy/On-premise)	نظام السجلات الطبية الالكترونية (HER/EMR)	نظام معلومات الأشعة (RIS)	System Feature
✓		✓	✓	مخصص لإدارة مركز أشعة
✓		✓		يتتكامل مع أنظمة الأرشيف
✓	✓	✓	✓	يقلل استخدام الورق
✓		✓	✓	يحسن تتبع المريض
✓				يقدم أدوات تشخيص متقدمة
✓		✓	✓	سيطرة على كافة البيانات
✓	✓		✓	واجهة مستخدم بسيطة وسهلة
✓				القدرة على الوصول من أجهزة متعددة

Table (1) Comparison of System

Chapter 3

Analytical and Design Study

Introduction

This section details the functional and non-functional requirements of the intelligent cloud-based radiology system. These requirements form the foundation of the entire project. Any errors or omissions at this stage may lead to significant issues in subsequent development, testing, and deployment phases.

The requirements were defined based on:

- Analysis of the target users' needs (administrators, physicians, reception staff, and patients)
- Review of modern radiology and teleradiology platforms
- Best practices in AI-assisted diagnostics, cloud security, and healthcare workflow optimization
- Continuous stakeholder feedback (radiologists, lab managers, and hospital administration) during the incremental development process

1. Requirements Gathering

- Approach Requirements were collected through:
- Study of existing radiology and diagnostic imaging systems
- Review of current healthcare standards, AI diagnostic guidelines, and cloud security protocols
- Iterative refinement during each development increment

2. Project Requirements

2.1 Functional Requirements

The system supports multiple user roles with clearly defined responsibilities and privileges.

Actor: System Administrator

REQ-1: Add employee (full name, username, password, email, phone, department, role – e.g., reception staff)

REQ-2: Edit employee data

REQ-3: Delete employee (with confirmation)

REQ-4: Add department (full name and other details)

REQ-5: Edit department

REQ-6: Delete department (with confirmation)

REQ-7: Add physician (full name, username, password, email, phone, specialty/department, work schedule)

REQ-8: Edit physician data

REQ-9: Delete physician (with confirmation)

REQ-10: View comprehensive statistical reports (number of patients, bookings, revenues, department occupancy rates, most requested departments, etc.)

Actor: doctor

REQ-11: Upload radiological image (X-Ray, MRI, CT, etc.) and permanently link it to the patient's file

REQ-12: Add medical diagnosis, suggested treatment, and clinical notes after each visit

REQ-13: Search for any patient by name or file number and view complete medical history (bookings, diagnoses, images, lab results)

REQ-14: Request AI assistance — upload image or describe case and receive AI-powered analysis, detection, or diagnostic suggestions

Actor: Reception Staff

REQ-15: Register new patient (complete personal data, guardian phone if minor)

REQ-16: Edit patient personal data

REQ-17: Check-out patient (end visit, issue invoice if applicable)

REQ-18: Create new appointment (select physician, date, time, considering physician schedule)

REQ-19: Modify existing appointment (change physician, date, or time)

REQ-20: Cancel appointment (with confirmation and optional patient notification)

REQ-21: View daily reports (number of bookings, new patients, cancelled appointments, etc.)

Actor: Patient

REQ-22: View and download all uploaded radiological images

REQ-23: Upload new image (or select existing) and request instant AI analysis

REQ-24: Book a new appointment independently (select specialty, physician, and available time)

REQ-25: Modify own appointment (change date/time) if before the scheduled time

REQ-26: Cancel own appointment (with sufficient notice)

REQ-27: Complete personal and medical information on first login (address, date of birth, blood type, chronic diseases, etc.)

REQ-28: Edit personal and medical information at any time

2.2 Non-Functional Requirements

2.2.1 Usability

- All interfaces must be clean, intuitive, and user-friendly for medical staff and patients
- Minimal clicks for critical actions (e.g., image upload, AI request, report viewing)
- Dashboards for administrators and physicians must be professional, organized, and visually clear

2.2.2 Performance

- Response time: ≤ 2 seconds for most interactions (image upload, AI analysis request, report loading)
- Support for high concurrency (multiple simultaneous users in busy labs/hospitals)
- Ultra-fast AI image processing with high accuracy

2.2.3 Availability

- The system must be accessible 24/7 via web and mobile devices
- Target uptime: $\geq 99.9\%$ (critical for healthcare)

2.2.4 Reliability

- Graceful handling of unexpected inputs (no crashes during image upload or AI processing)
- All critical data (images, diagnoses, patient records) must be preserved and recoverable

2.2.5 Implementation Constraints

- Frontend: Responsive design (modern framework e.g., React / Angular)
- Backend: RESTful API with secure cloud integration
- Database: Relational (MySQL/PostgreSQL) + cloud storage for large image files

2.2.6 Testing

- Unit testing for backend services and AI integration
- Integration testing for image upload, AI analysis, and cloud sync
- Usability testing with real radiologists and patients
- Security testing (OWASP Top 10) including HIPAA/GDPR compliance checks

2.2.7 Security

- All passwords stored using strong hashing (MD5)
- Role-based access control (RBAC) to restrict actions per user type
- Protection against SQL injection using prepared statements and security headers
- HTTPS enforced across the platform
- End-to-end encryption for patient images and medical data
- Secure file uploads with size/type validation, virus scanning, and DICOM compliance

2.2.8 Maintainability & Scalability

- Modular architecture allowing easy addition of new AI models and departments
- Automated cloud backups and disaster recovery mechanisms
- Scalable infrastructure to handle growing image volumes and user base

Requirements tracing matrix

Requirement ID	Actor	Full Functional Description	Priority	Test Case
REQ-1	Administrator	The Administrator must be able to add a new employee to the system by specifying full name, username, password, email, phone number, department, and role (e.g., reception staff).	1	TC-1
REQ-2	Administrator	The Administrator must be able to edit all data of any existing employee in the system.	1	TC-2
REQ-3	Administrator	The Administrator must be able to permanently delete any employee from the system, with a deletion confirmation prompt.	1	TC-3
REQ-4	Administrator	The Administrator must be able to add a new department to the system by specifying the full name and other relevant information.	1	TC-4
REQ-5	Administrator	The Administrator must be able to edit all data of any existing department in the system.	1	TC-5
REQ-6	Administrator	The Administrator must be able to permanently delete any department from the system, with a deletion confirmation prompt.	1	TC-6
REQ-7	Administrator	The Administrator must be able to add a new physician by specifying full name, username, password, email, phone number, specialty (department), and work schedule.	1	TC-7
REQ-8	Administrator	The Administrator must be able to edit all existing data of any physician in the system.	1	TC-8
REQ-9	Administrator	The Administrator must be able to permanently delete any physician from the system, with a deletion confirmation prompt.	1	TC-9
REQ-10	Administrator	The Administrator must have access to a dashboard containing comprehensive statistical reports (number of patients, bookings, revenues, department occupancy rates, most requested departments, etc.).	4	TC-10
REQ-11	Physician	The physician must be able to upload a radiological image (X-Ray, MRI, CT, etc.) for a specific patient and permanently link it to the patient's file.	3	TC-11

REQ-12	Physician	The physician must be able to write the medical diagnosis, suggested treatment, and clinical notes in the patient's file after each visit.	3	TC-12
REQ-13	Physician	The physician must be able to search for any patient by name or file number and view the complete medical history (bookings, diagnoses, radiological images, lab results).	3	TC-13
REQ-14	Physician	The physician must be able to upload a radiological image or describe the case and request analysis or diagnostic suggestions from the integrated medical AI.	-	TC-14
REQ-15	Reception Staff	The reception staff must be able to register a new patient in the system by entering complete personal data and guardian phone number (if minor).	2	TC-15
REQ-16	Reception Staff	The reception staff must be able to edit any personal data of an existing patient.	2	TC-16
REQ-17	Reception Staff	The reception staff must be able to complete the patient visit (check-out), register the exit, and issue an invoice if applicable.	2	TC-17
REQ-18	Reception Staff	The reception staff must be able to book a new appointment for a patient with a specific physician on a specific date and time, considering the physician's schedule.	2	TC-18
REQ-19	Reception Staff	The reception staff must be able to modify an existing appointment (change physician, date, or time).	2	TC-19
REQ-20	Reception Staff	The reception staff must be able to cancel an appointment with a confirmation prompt and optional patient notification (if possible).	2	TC-20
REQ-21	Reception Staff	The reception staff must be able to view daily reports (number of bookings, new patients, canceled appointments, etc.).	4	TC-21
REQ-22	Patient	The patient must be able to log in to their account and view all uploaded radiological images with the ability to download them.	-	TC-22
REQ-23	Patient	The patient must be able to upload a new radiological image (or select an existing one) and request instant AI analysis.	-	TC-23
REQ-24	Patient	The patient must be able to book a new appointment independently by selecting the specialty, physician, and available time.	-	TC-24

REQ-25	Patient	The patient must be able to modify their own appointment (change date or time) as long as the appointment time has not arrived.	-	TC-25
REQ-26	Patient	The patient must be able to cancel their own appointment at any time with sufficient notice before the scheduled time.	-	TC-26
REQ-27	Patient	Upon first login, the patient must be prompted to complete their personal information (address, date of birth, blood type, chronic diseases, etc.).	-	TC-27
REQ-28	Patient	The patient must be able to edit their personal and medical information at any time.	-	TC-28

Table (2) Requirements Tracing Matrix (RTM)

3.Analytical Chart of the Project:

3.1 Use-Case Diagram:

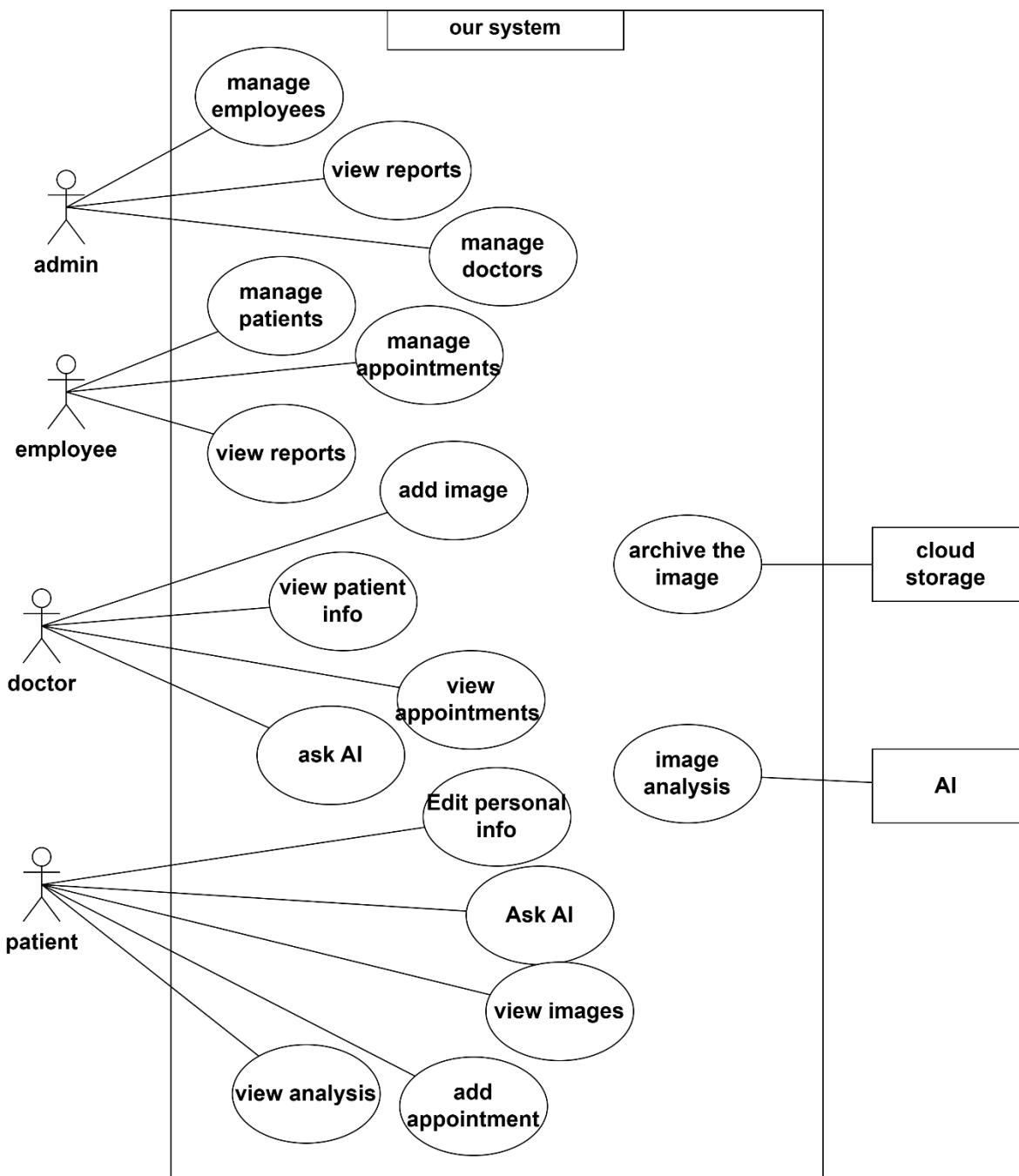


Figure (1) Use-Case Diagram high level

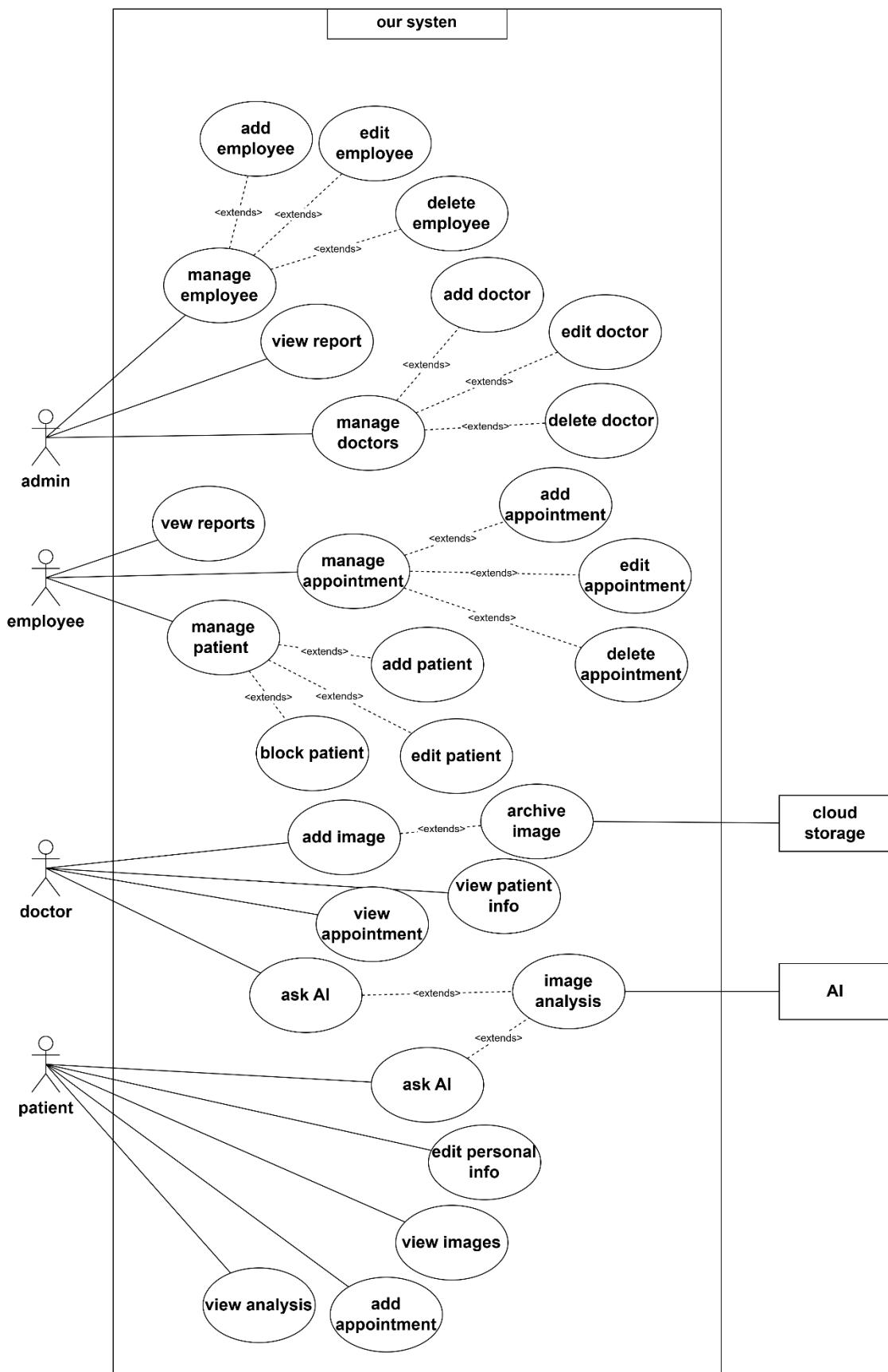


Figure (2) Use-Case Diagram low level

3.2 Use-case Narrative:

Field	Content
Use Case Name	Add Employee
Use Case ID	UC-01
Actor	Administrator (Admin)
Requirement ID	REQ-1
Description	The Administrator adds a new employee to the system by specifying their basic information and department.
Preconditions	The Administrator must be logged in with permission to manage employees.
Postconditions	Employee data is saved in the database and an account is created.
Main Flow	1. Administrator enters the Add Employee page 2. Fills in the fields (name, email, password, department...) 3. Clicks "Save" 4. System validates the data 5. Employee is saved and success confirmation is displayed
Alternative Flow	If data is invalid or duplicated → error message appears and re-entry is requested

Table (3) Use case 1

Field	Content
Use Case Name	Edit Employee
Use Case ID	UC-02
Actor	Administrator
Requirement ID	REQ-2
Description	The Administrator edits the data of an existing employee.
Preconditions	Administrator is logged in and the employee already exists.
Postconditions	Employee data is updated in the database.
Main Flow	1. Administrator selects the employee from the list 2. Clicks "Edit" 3. Modifies required fields 4. Clicks "Save" 5. System validates and updates the data
Alternative Flow	If data is invalid → error message and retry requested

Table (4) Use case 2

Field	Content
Use Case Name	Delete Employee
Use Case ID	UC-03
Actor	Administrator
Requirement ID	REQ-3

Description	The Administrator deletes an employee from the system.
Preconditions	Administrator is logged in and the employee exists.
Postconditions	Employee account and data are permanently deleted from the database.
Main Flow	1. Administrator selects the employee 2. Clicks "Delete" 3. Deletion confirmation appears 4. Confirms deletion 5. Deletion occurs and success message is shown
Alternative Flow	If confirmation is rejected → operation is canceled

Table (5) Use case 3

Field	Content
Use Case Name	Add Department
Use Case ID	UC-04
Actor	Administrator (Admin)
Requirement ID	REQ-4
Description	The Administrator adds a new department to the system by specifying its basic information.
Preconditions	Administrator is logged in with permission to manage departments.
Postconditions	Department data is saved in the database.
Main Flow	1. Administrator enters the Add Department page 2. Fills in the fields (name...) 3. Clicks "Save" 4. System validates the data 5. Department is saved and success confirmation is displayed
Alternative Flow	If data is invalid or duplicated → error message and re-entry requested

Table (6) Use case 4

Field	Content
Use Case Name	Edit Department
Use Case ID	UC-05
Actor	Administrator
Requirement ID	REQ-5
Description	The Administrator edits the data of an existing department.
Preconditions	Administrator is logged in and the department already exists.
Postconditions	Department data is updated in the database.
Main Flow	1. Administrator selects the department from the list 2. Clicks "Edit" 3. Modifies required fields 4. Clicks "Save" 5. System validates and updates the data
Alternative Flow	If data is invalid → error message and retry requested

Table (7) Use case 5

Field	Content
Use Case Name	Delete Department
Use Case ID	UC-06
Actor	Administrator
Requirement ID	REQ-6
Description	The Administrator deletes a department from the system.
Preconditions	Administrator is logged in and the department exists.
Postconditions	Department account and data are permanently deleted from the database.
Main Flow	1. Administrator selects the department 2. Clicks "Delete" 3. Deletion confirmation appears 4. Confirms deletion 5. Deletion occurs and success message is shown
Alternative Flow	If confirmation is rejected → operation is canceled

Table (8) Use case 6

Field	Content
Use Case Name	Add doctor
Use Case ID	UC-07
Actor	Administrator (Admin)
Requirement ID	REQ-7
Description	The Administrator adds a new doctor to the system by specifying their basic information.
Preconditions	Administrator is logged in with permission to manage doctors.
Postconditions	doctor data is saved in the database.
Main Flow	1. Administrator enters the Add doctor page 2. Fills in the fields (name, password...) 3. Clicks "Save" 4. System validates the data 5. doctor is saved and success confirmation is displayed
Alternative Flow	If data is invalid or duplicated → error message and re-entry requested

Table (9) Use case 7

Field	Content
Use Case Name	Edit doctor
Use Case ID	UC-08
Actor	Administrator
Requirement ID	REQ-8
Description	The Administrator edits the data of an existing doctor.
Preconditions	Administrator is logged in and the doctor already exists.
Postconditions	doctor data is updated in the database.
Main Flow	1. Administrator selects the doctor from the list 2. Clicks "Edit" 3. Modifies required fields 4. Clicks "Save" 5. System validates and updates the data

Alternative Flow	If data is invalid → error message and retry requested
-------------------------	--

Table (10) Use case 8

Field	Content
Use Case Name	Delete doctor
Use Case ID	UC-09
Actor	Administrator
Requirement ID	REQ-9
Description	The Administrator deletes a doctor from the system.
Preconditions	Administrator is logged in and the doctor exists.
Postconditions	doctor account and data are permanently deleted from the database.
Main Flow	1. Administrator selects the doctor 2. Clicks "Delete" 3. Deletion confirmation appears 4. Confirms deletion 5. Deletion occurs and success message is shown
Alternative Flow	If confirmation is rejected → operation is canceled

Table (11) Use case 9

Field	Content
Use Case Name	View Reports
Use Case ID	UC-10
Actor	Administrator
Requirement ID	REQ-10
Description	Display statistical reports about center usage (bookings, patients, departments...).
Preconditions	Administrator is logged in.
Postconditions	No changes – view only.
Main Flow	1. Enter the reports page 2. Select report type or period 3. Display charts and numbers
Alternative Flow	No data → Message "No data available"

Table (12) Use case 10

Field	Content
Use Case Name	Add Radiological Image
Use Case ID	UC-11
Actor	doctor
Requirement ID	REQ-11
Description	The doctor uploads a radiological image for a specific patient.
Preconditions	doctor is logged in, patient exists in the system.
Postconditions	Image is saved and permanently linked to the patient.
Main Flow	1. Select the patient 2. Upload the file 3. Add notes (optional) 4. Save
Alternative Flow	Unsupported file or too large → error message

Table (13) Use case 11

Field	Content
Use Case Name	Add Diagnosis
Use Case ID	UC-12
Actor	doctor
Requirement ID	REQ-12
Description	The doctor adds a medical diagnosis for a specific patient after examination.
Preconditions	doctor is logged in, patient exists, current booking or visit exists.
Postconditions	Diagnosis is saved in the patient's medical record.
Main Flow	1. doctor selects the patient 2. Enters diagnosis page 3. Writes diagnosis and notes 4. Clicks Save 5. Addition confirmation is shown
Alternative Flow	Incomplete data → error message and request to complete fields

Table (14) Use case 12

Field	Content
Use Case Name	Query Patient
Use Case ID	UC-13
Actor	doctor
Requirement ID	REQ-13
Description	The doctor searches for patient data and views their medical record.
Preconditions	doctor is logged in, patient exists in the database.
Postconditions	Patient data and medical record are displayed (view only).
Main Flow	1. doctor enters patient name or file number in search 2. Clicks Search 3. System displays patient data and record (bookings, images, diagnoses)
Alternative Flow	No patient found → message "No patient with this information"

Table (15) Use case 13

Field	Content
Use Case Name	Request AI Assistance
Use Case ID	UC-14
Actor	doctor
Requirement ID	REQ-14
Description	The doctor requests suggestions or analysis from the AI system based on patient data.
Preconditions	doctor is logged in, AI service is available.
Postconditions	AI suggestions are displayed to the doctor.

Main Flow	1. doctor selects patient or image 2. Clicks "Request Assistance" 3. System sends request 4. Result (possible diagnosis, suggestions) is displayed
Alternative Flow	AI connection failure → message "Service currently unavailable"

Table (16) Use case 14

Field	Content
Use Case Name	Add Patient
Use Case ID	UC-15
Actor	employee
Requirement ID	REQ-15
Description	The employee registers a new patient in the system.
Preconditions	employee is logged in with sufficient permissions.
Postconditions	Patient data is saved in the database.
Main Flow	1. Enter Add Patient page 2. Fill in personal information 3. Save 4. Addition confirmation
Alternative Flow	Duplicate or invalid data → error message

Table (17) Use case 15

Field	Content
Use Case Name	Edit Patient
Use Case ID	UC-16
Actor	employee
Requirement ID	REQ-16
Description	Edit data of an existing patient.
Preconditions	Patient exists, employee is logged in.
Postconditions	Patient data is updated.
Main Flow	1. Search for the patient 2. Edit fields 3. Save changes
Alternative Flow	Input error → retry

Table (18) Use case 16

Field	Content
Use Case Name	Discharge Patient
Use Case ID	UC-17
Actor	employee
Requirement ID	REQ-17
Description	End a patient's stay or treatment and record discharge.
Preconditions	Patient exists as inpatient or under treatment.
Postconditions	Patient status changed to "Discharged" and record closed.
Main Flow	1. Select patient 2. Enter discharge date and notes 3. Confirm 4. Save
Alternative Flow	Cancel operation → no change

Table (19) Use case 17

Field	Content
Use Case Name	Add Appointment

Use Case ID	UC-18
Actor	employee
Requirement ID	REQ-18
Description	Create a new appointment for a patient with a physician and department.
Preconditions	Patient and physician exist.
Postconditions	Appointment is saved and physician/reception notified (optional).
Main Flow	1. Select patient, physician, date/time 2. Save 3. Confirm appointment
Alternative Flow	Schedule conflict → suggest alternative times

Table (20) Use case 18

Field	Content
Use Case Name	Edit Appointment
Use Case ID	UC-19
Actor	employee
Requirement ID	REQ-19
Description	Change details of an existing appointment.
Preconditions	Appointment exists.
Postconditions	Appointment is updated in the system.
Main Flow	1. Select appointment 2. Modify date/time/physician 3. Save
Alternative Flow	New conflict → error message

Table (21) Use case 19

Field	Content
Use Case Name	Delete Appointment
Use Case ID	UC-20
Actor	employee
Requirement ID	REQ-20
Description	Cancel an existing appointment.
Preconditions	Appointment exists.
Postconditions	Appointment is removed from the system.
Main Flow	1. Select appointment 2. Confirm deletion 3. Execute deletion
Alternative Flow	Cancel confirmation → no deletion

Table (22) Use case 20

Field	Content
Use Case Name	View Reports
Use Case ID	UC-21
Actor	employee
Requirement ID	REQ-21
Description	Display limited reports related to bookings and patients.
Preconditions	employee is logged in.
Postconditions	View only, no changes.
Main Flow	1. Enter reports page 2. Select report 3. Display data
Alternative Flow	No data → empty message

Table (23) Use case 21

Field	Content
Use Case Name	View Radiological Images
Use Case ID	UC-22
Actor	Patient
Requirement ID	REQ-22
Description	The patient views their radiological images.
Preconditions	Patient is logged in, images are linked to their account.
Postconditions	Images are displayed (view only).
Main Flow	1. Enter images page 2. Display list of images 3. Select and view an image
Alternative Flow	No images → message "No images available"

Table (24) Use case 22

Field	Content
Use Case Name	Request AI Diagnosis
Use Case ID	UC-23
Actor	Patient
Requirement ID	REQ-23
Description	The patient requests AI analysis of their radiological image.
Preconditions	Image exists, AI service is available.
Postconditions	AI analysis result is displayed to the patient.
Main Flow	1. Select image 2. Request analysis 3. Display result
Alternative Flow	Service failure → error message

Table (25) Use case 23

Field	Content
Use Case Name	Add Appointment
Use Case ID	UC-24
Actor	Patient
Requirement ID	REQ-24
Description	The patient books an appointment themselves.
Preconditions	Patient is logged in.
Postconditions	Appointment is saved and physician/reception notified.
Main Flow	1. Select department/physician and date/time 2. Confirm booking
Alternative Flow	No available times → suggest alternatives

Table (26) Use case 24

Field	Content
Use Case Name	Edit Appointment
Use Case ID	UC-25
Actor	Patient
Requirement ID	REQ-25
Description	The patient modifies their own appointment.
Preconditions	Appointment exists for the patient.

Postconditions	Appointment is updated.
Main Flow	1. View My Appointments 2. Select appointment and edit 3. Save
Alternative Flow	Conflict → error message

Table (27) Use case 25

Field	Content
Use Case Name	Delete Appointment
Use Case ID	UC-26
Actor	Patient
Requirement ID	REQ-26
Description	The patient cancels their own appointment.
Preconditions	Appointment exists.
Postconditions	Appointment is removed.
Main Flow	1. Select appointment 2. Confirm cancellation 3. Execute cancellation
Alternative Flow	Cancel confirmation → no deletion

Table (28) Use case 26

Field	Content
Use Case Name	Add Personal Information
Use Case ID	UC-27
Actor	Patient
Requirement ID	REQ-27
Description	Complete or add personal information on first login.
Preconditions	Patient is logging in for the first time.
Postconditions	Complete personal information is saved.
Main Flow	1. Display information form 2. Fill in fields 3. Save
Alternative Flow	Incomplete data → request completion

Table (29) Use case 27

Field	Content
Use Case Name	Edit Personal Information
Use Case ID	UC-28
Actor	Patient
Requirement ID	REQ-28
Description	Modify personal information such as address or phone number.
Preconditions	Patient is logged in.
Postconditions	Personal information is updated in the profile.
Main Flow	1. Enter profile page 2. Edit fields 3. Save
Alternative Flow	Invalid data → error message

Table (30) Use case 28

3.3 Class Diagram:

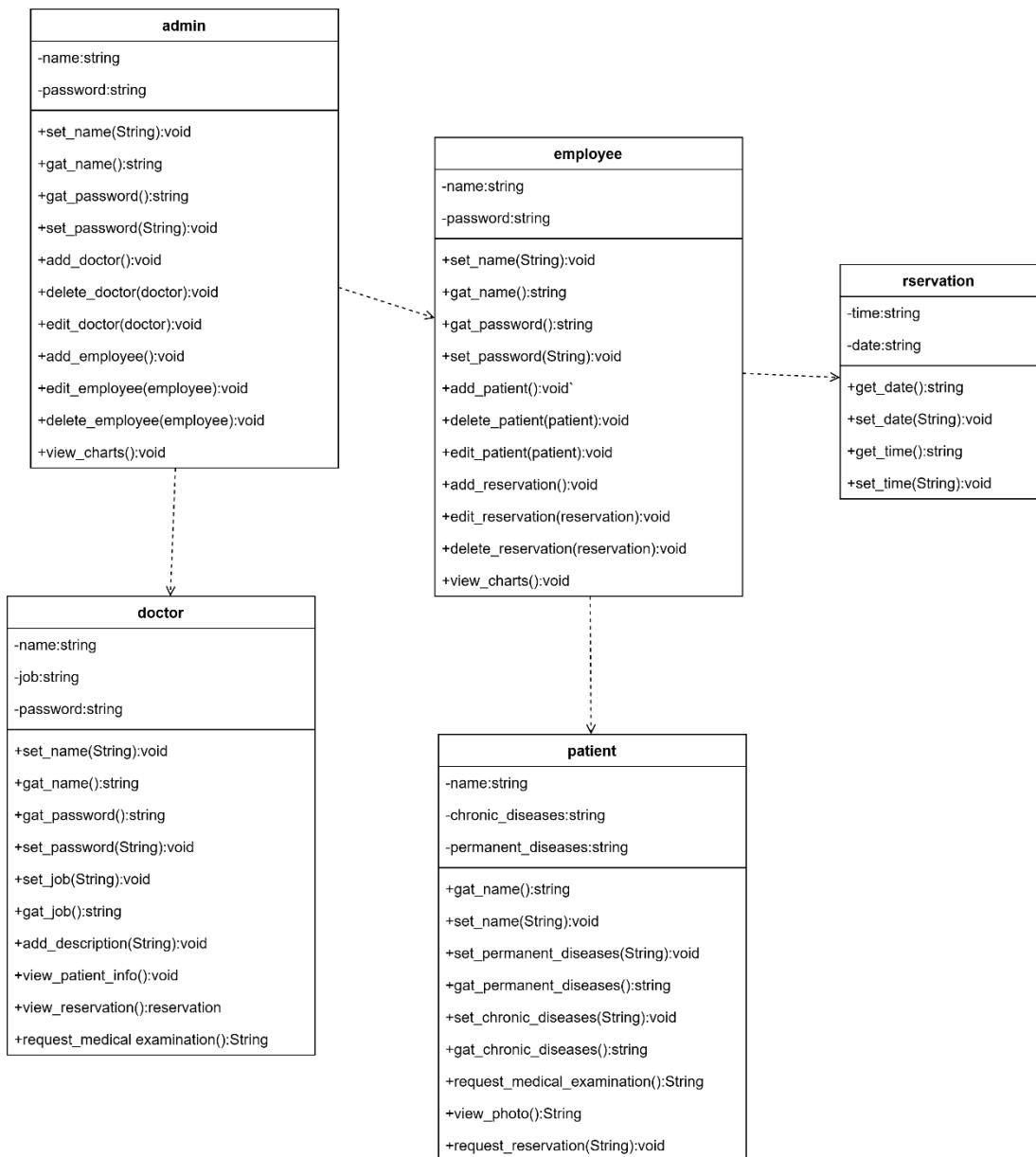


Figure (3) Class-Diagram

3.4 ERD-Diagram:

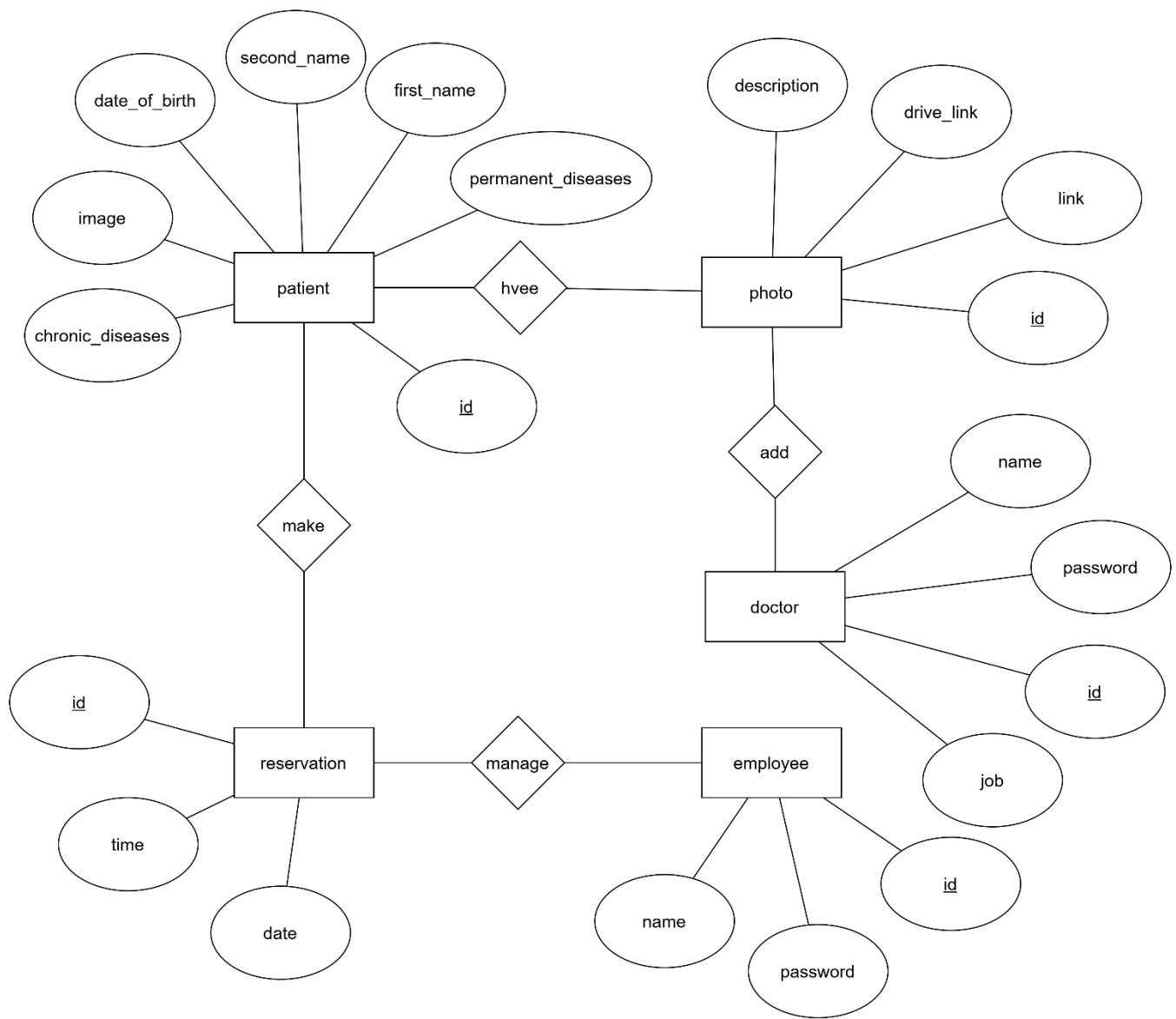


Figure (4) ERD-Diagram

3.5 Site Maps of The System:

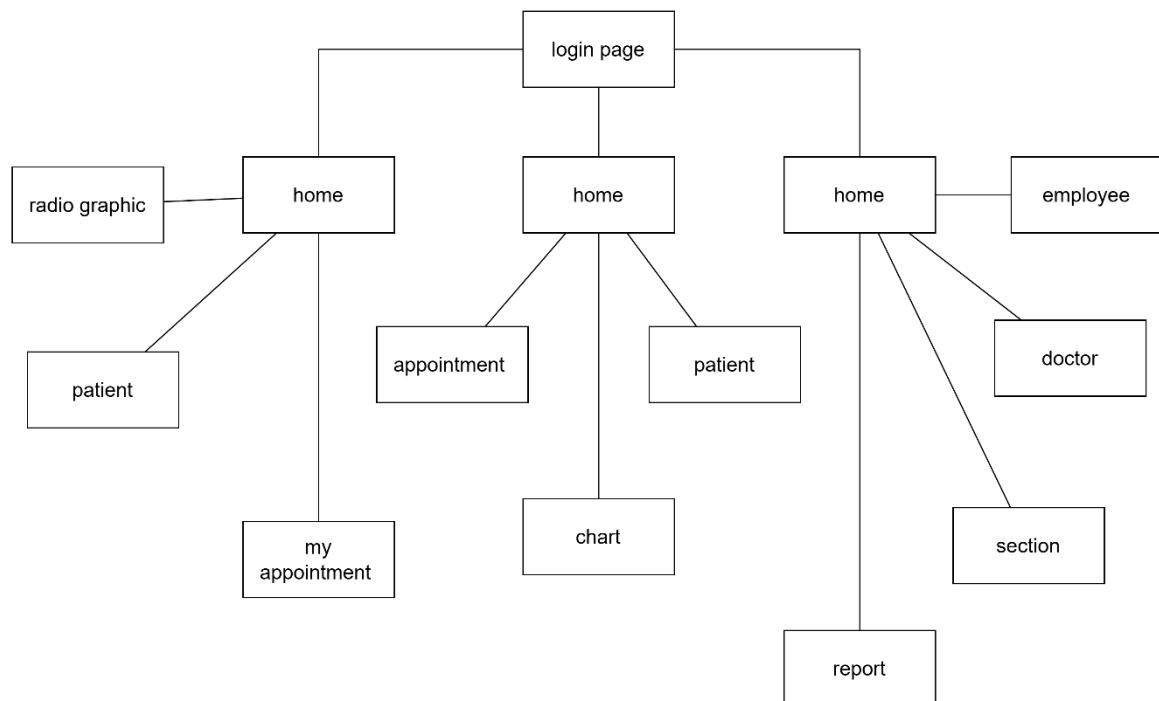


Figure (5) Site Map-Diagram

3.6 Gant Chart:

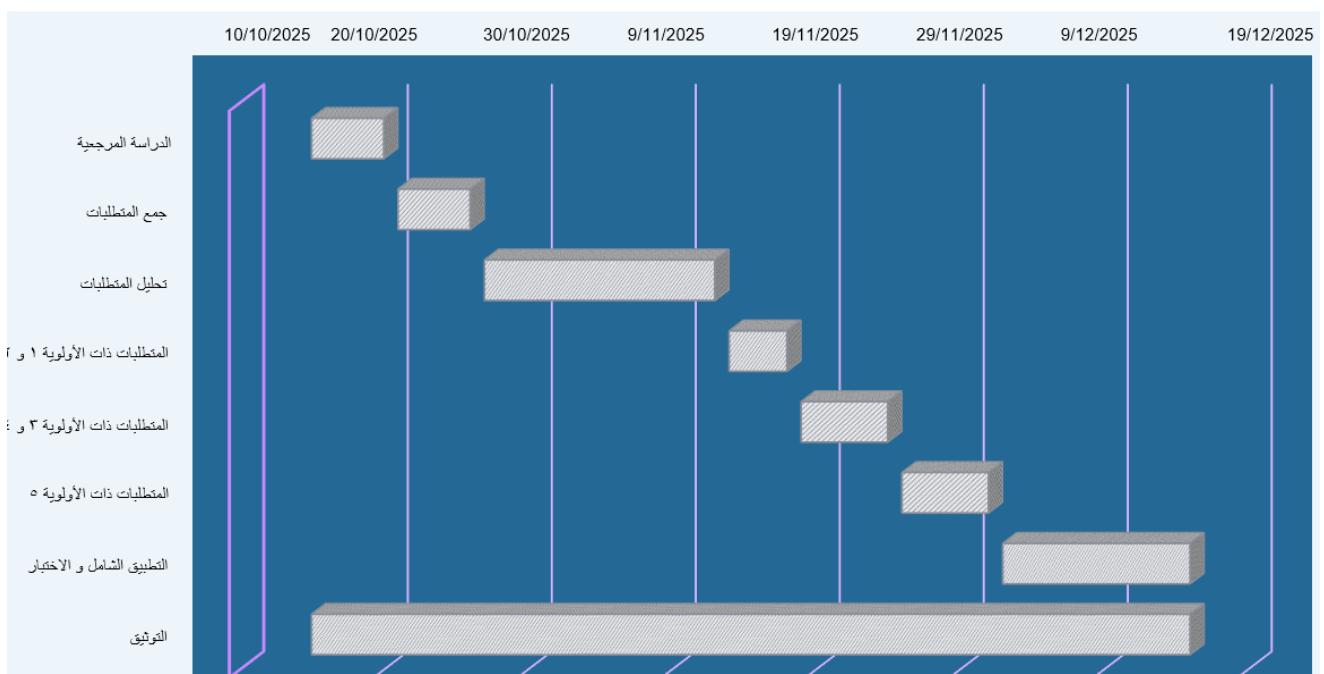


Figure (6) Gant Chart

3.7 Activity-Diagram:

3.7.1 Employee activity

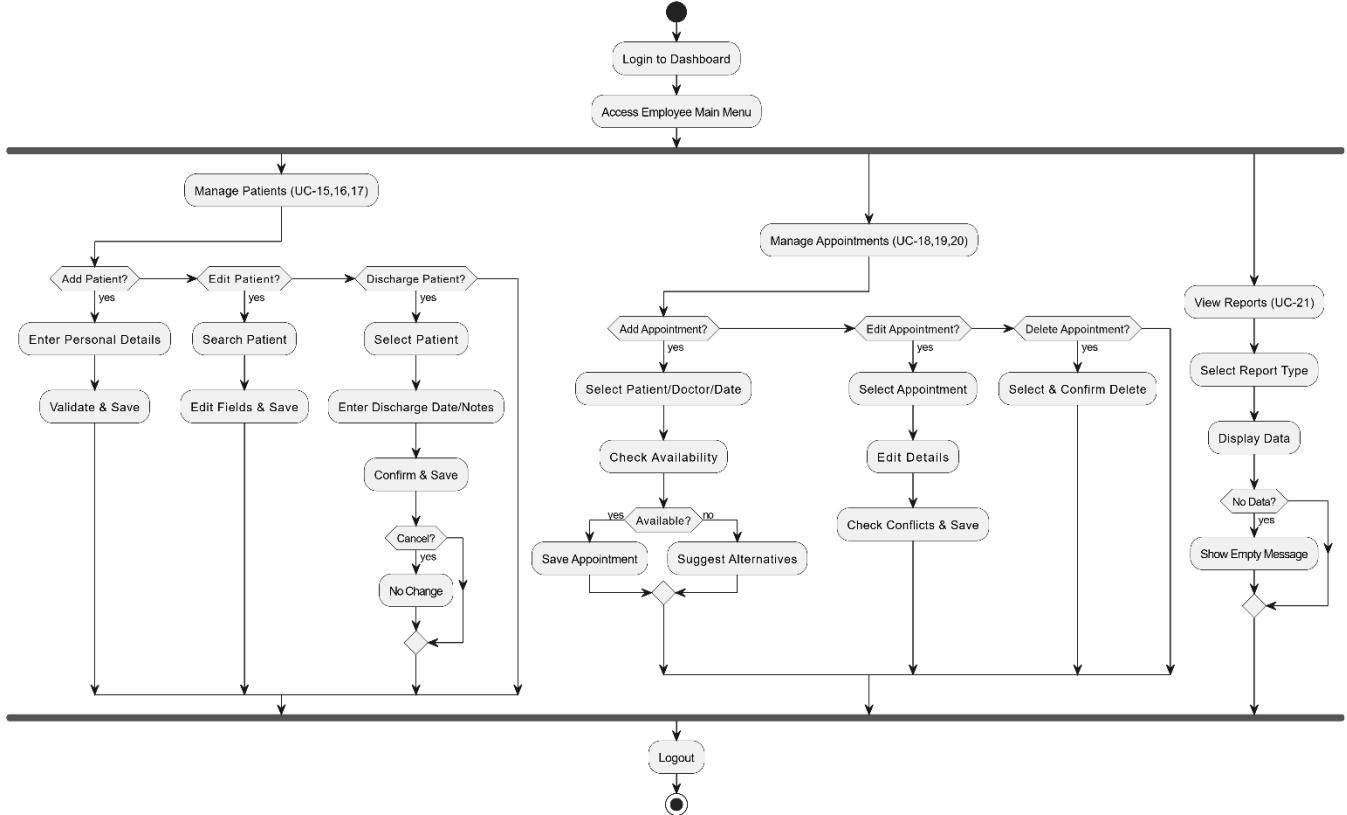


Figure (7) Activity -Diagram (employee)

3.7.2 Admin activity

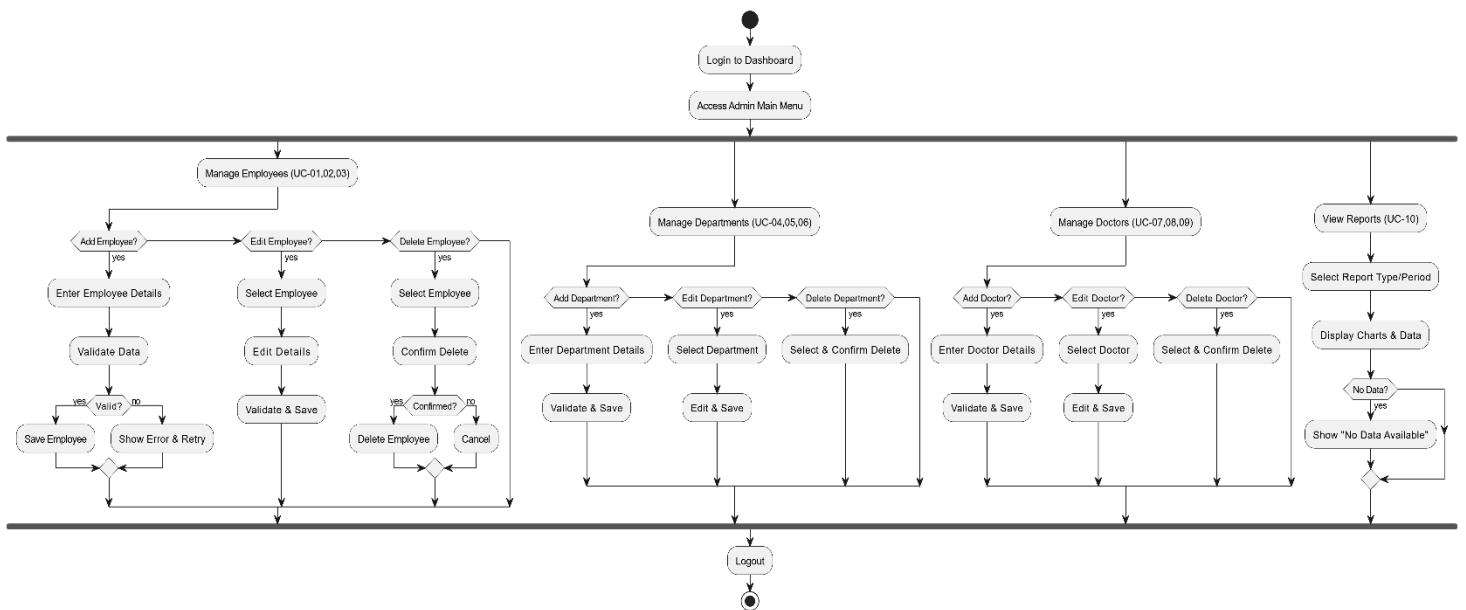


Figure (8) Activity -Diagram (admin)

3.7.3 Patient activity

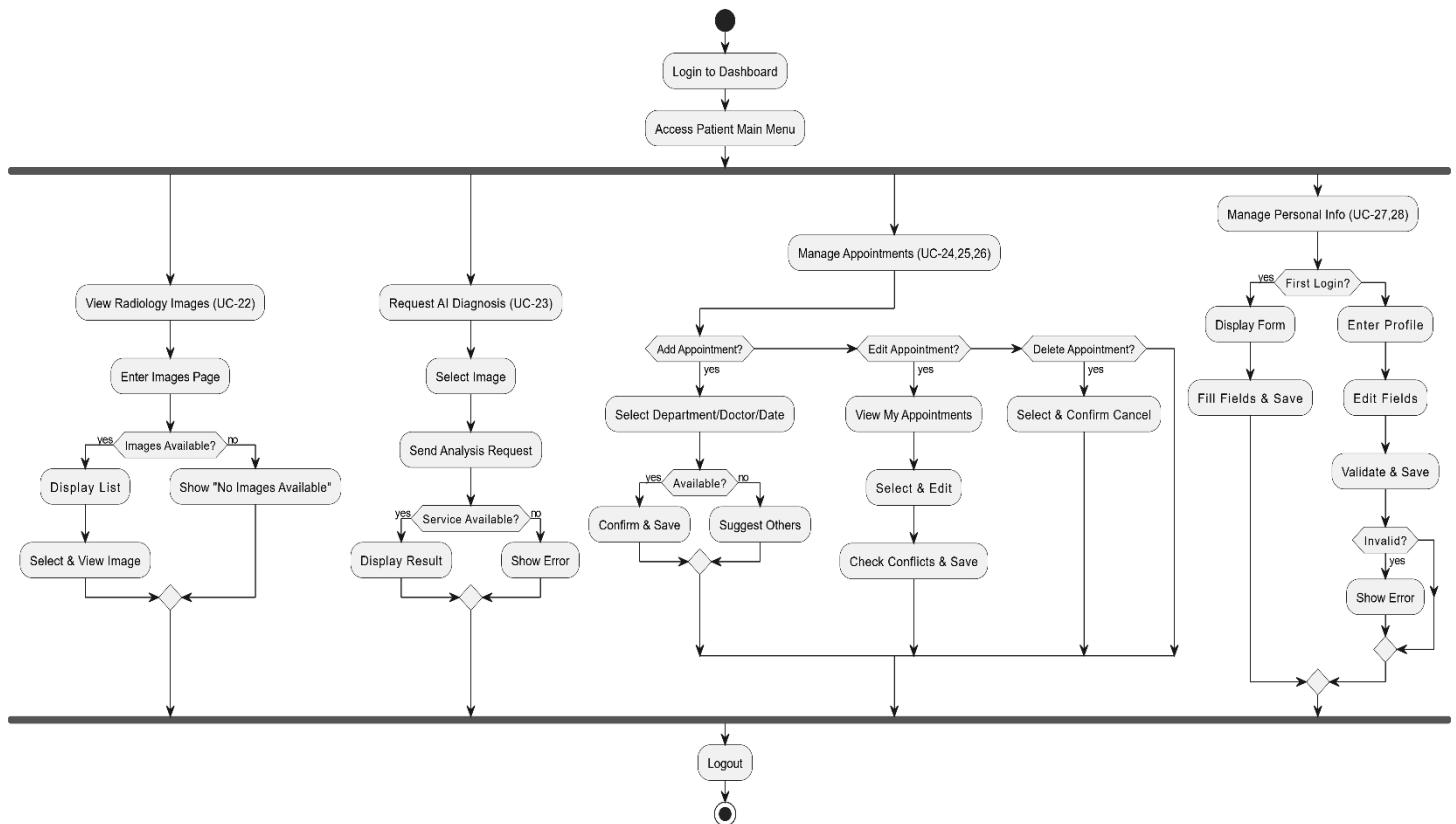


Figure (9) Activity -Diagram (patient)

3.7.4 Doctor activity

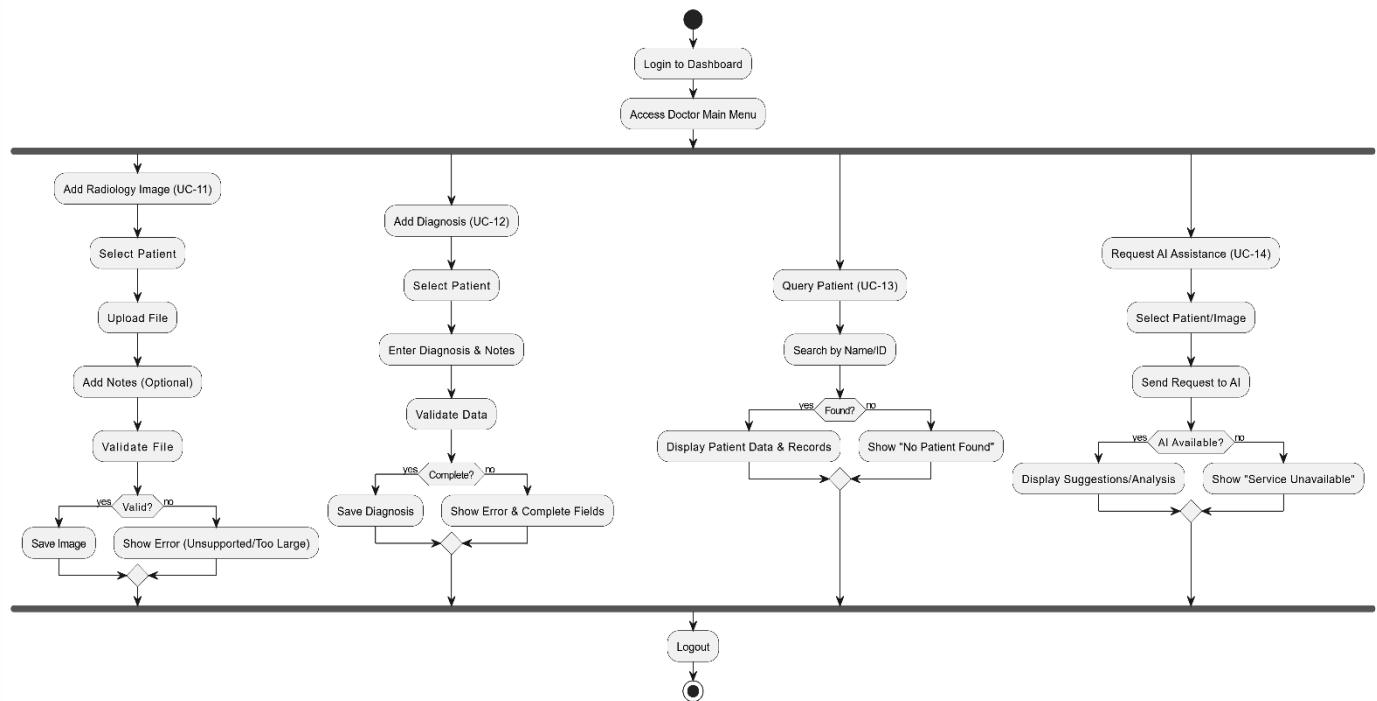


Figure (10) Activity-Diagram (doctor)

3.8 Sequence Diagram:

3.8.1 use cases 1,2

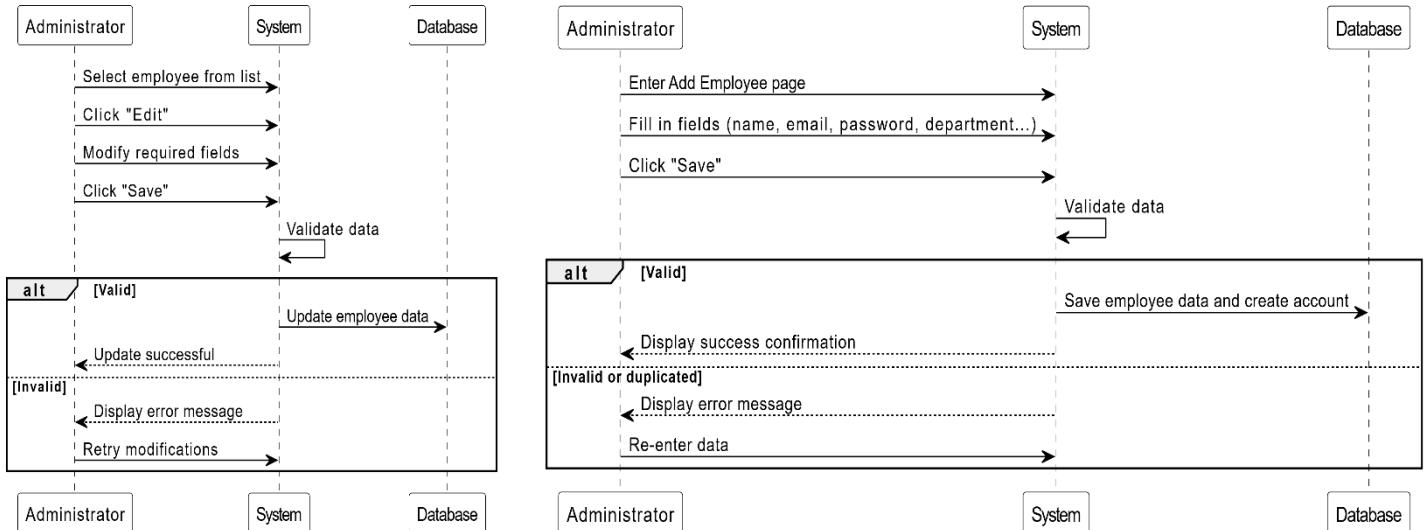


Figure (11) use cases 1,2

3.8.2 use cases 3,4

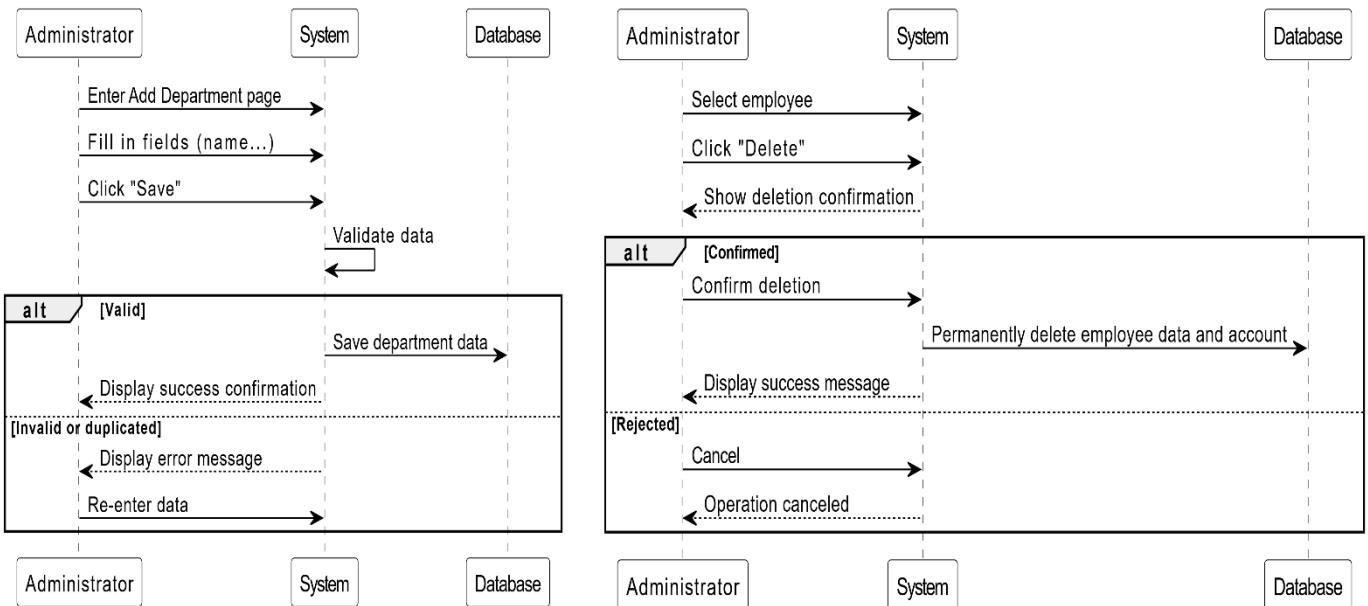


Figure (12) use cases 3,4

3.8.3 use cases 5,6

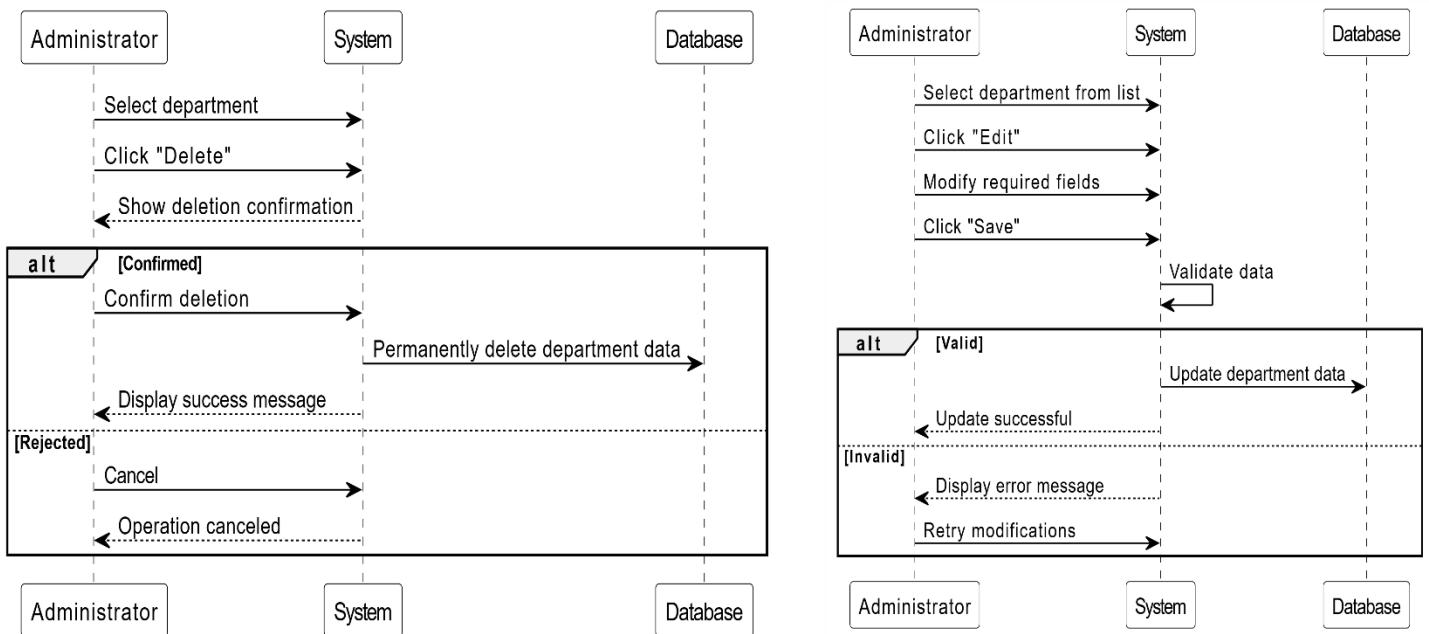


Figure (13) use cases 5,6

3.8.4 use cases 7,8

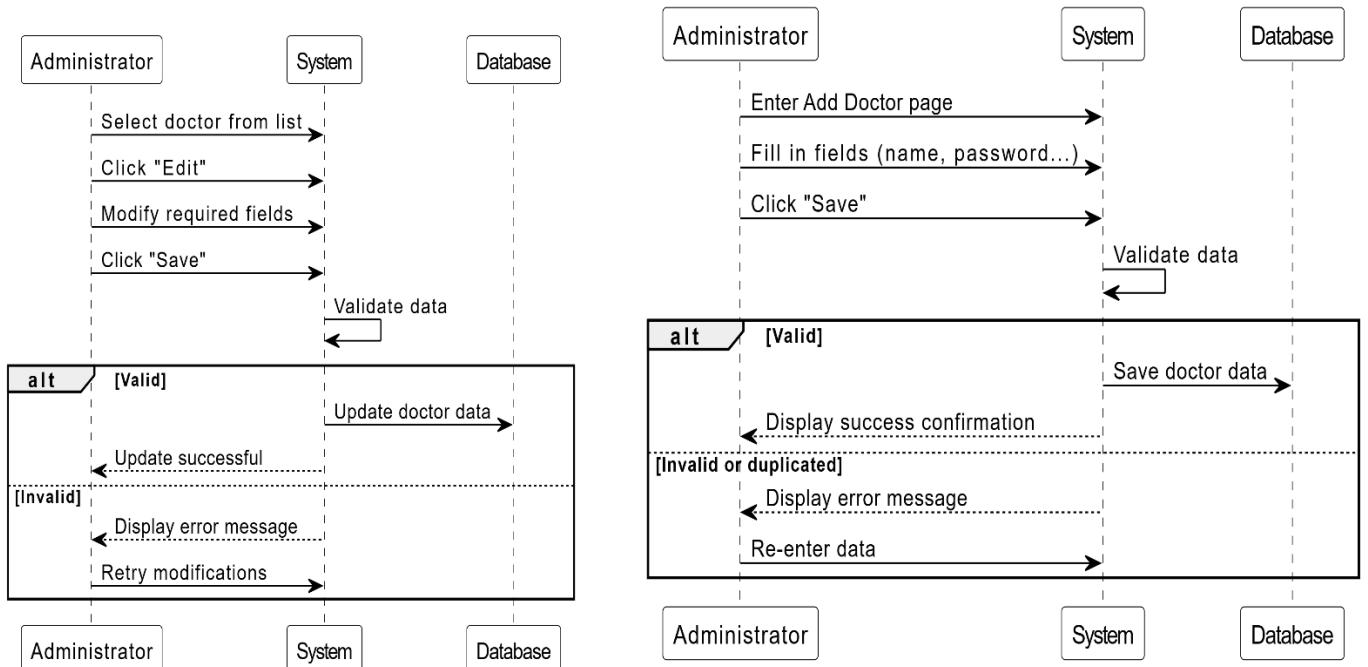


Figure (14) use cases 7,8

3.8.5 use cases 9,10

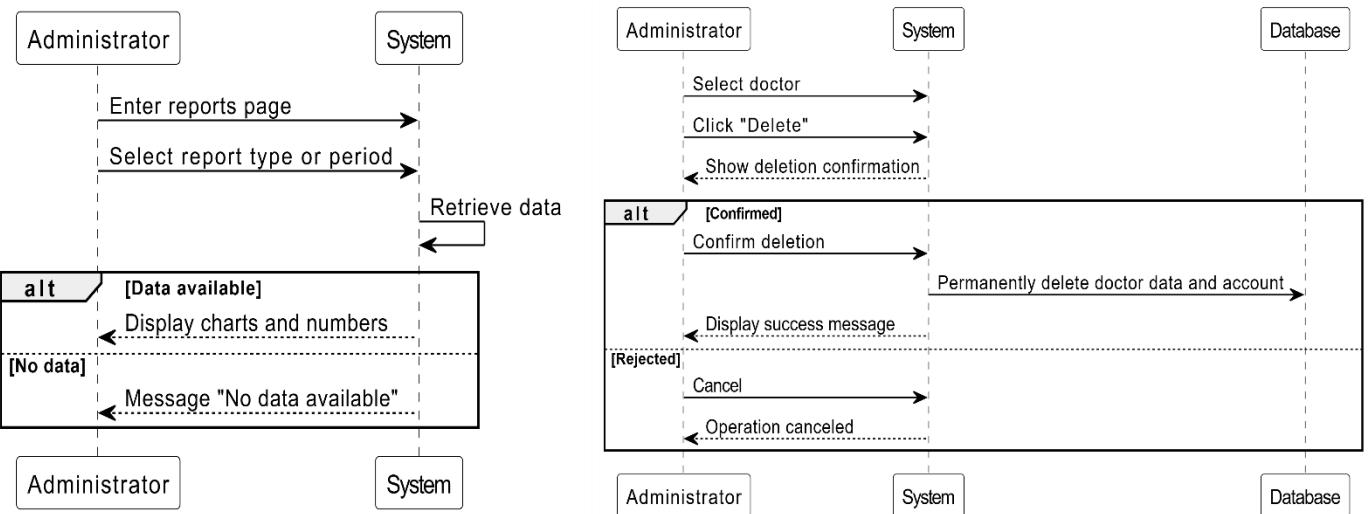


Figure (15) use cases 9,10

3.8.6 use cases 11,12

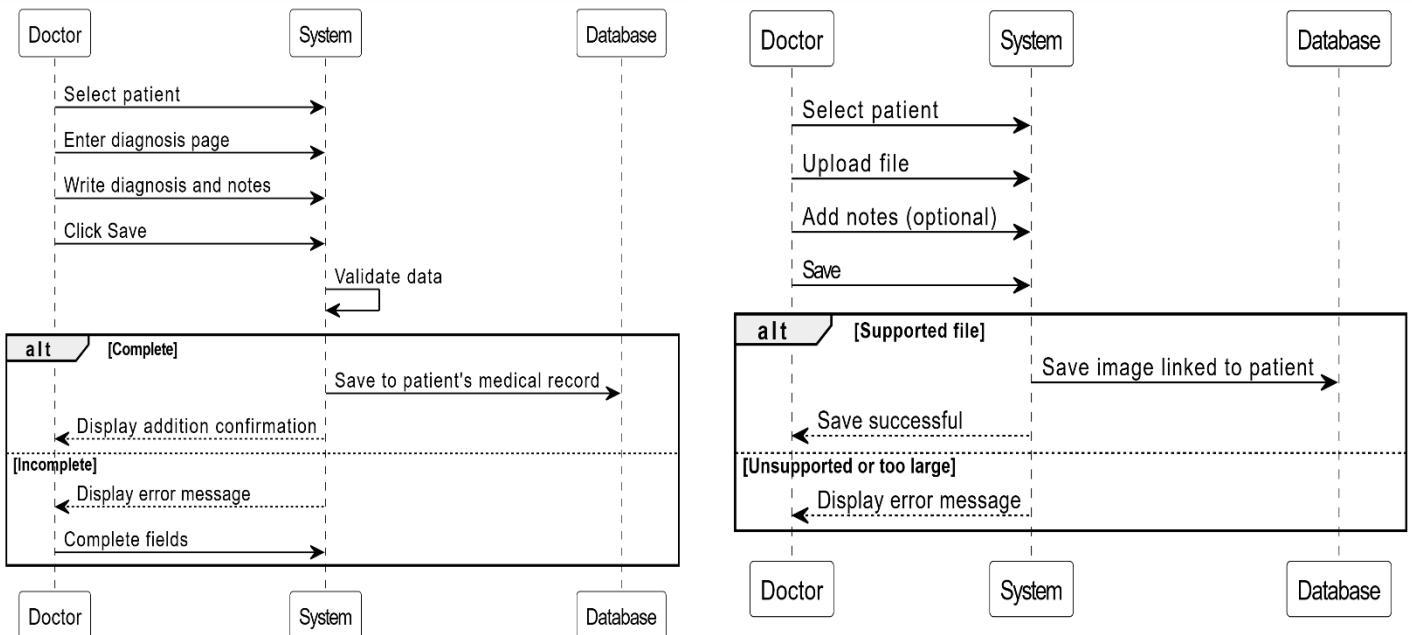


Figure (16) use cases 11,12

3.8.7 use cases 13,14

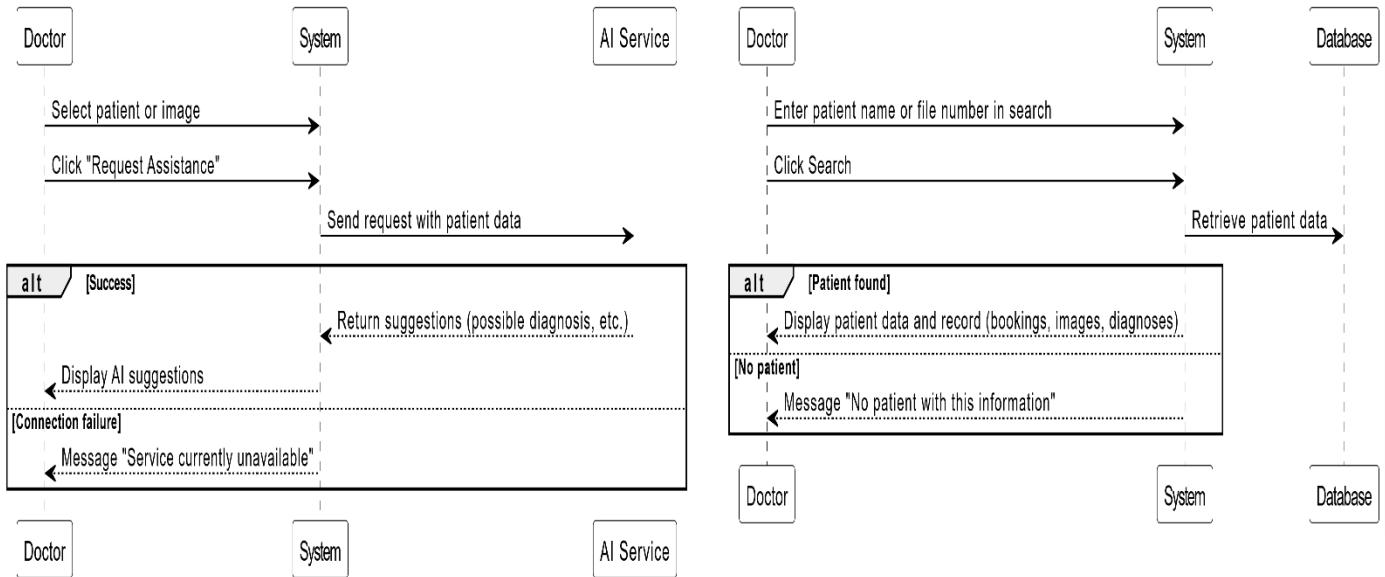


Figure (17) use cases 13,14

3.8.8 use cases 15,16

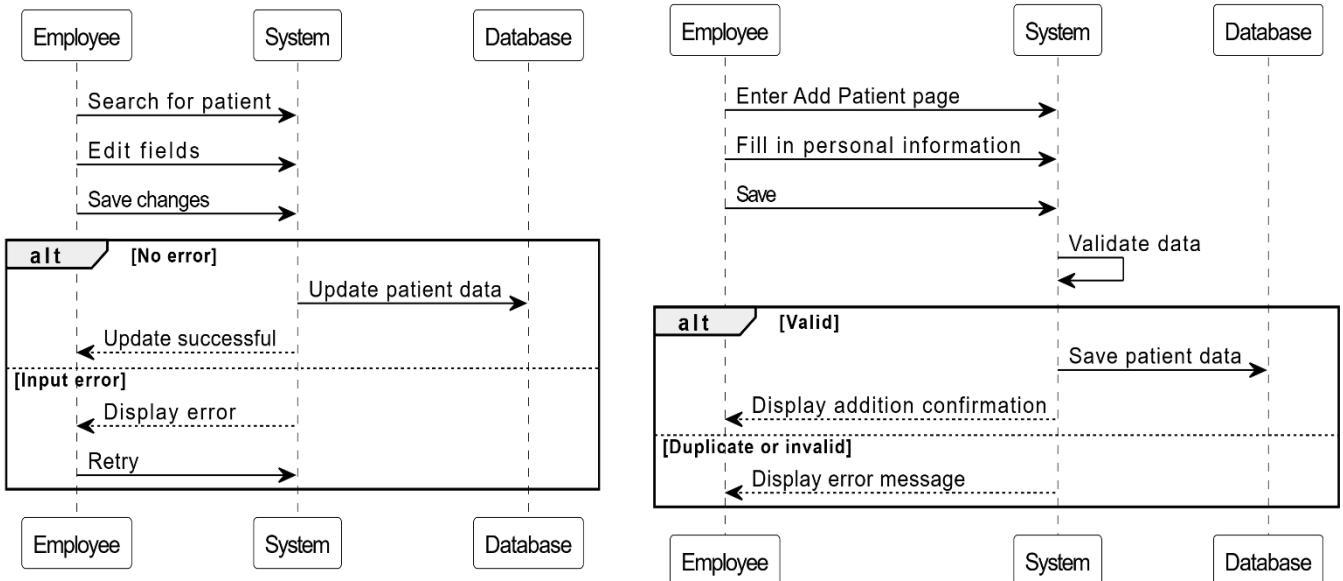


Figure (18) use cases 15,16

3.8.9 use cases 17,18

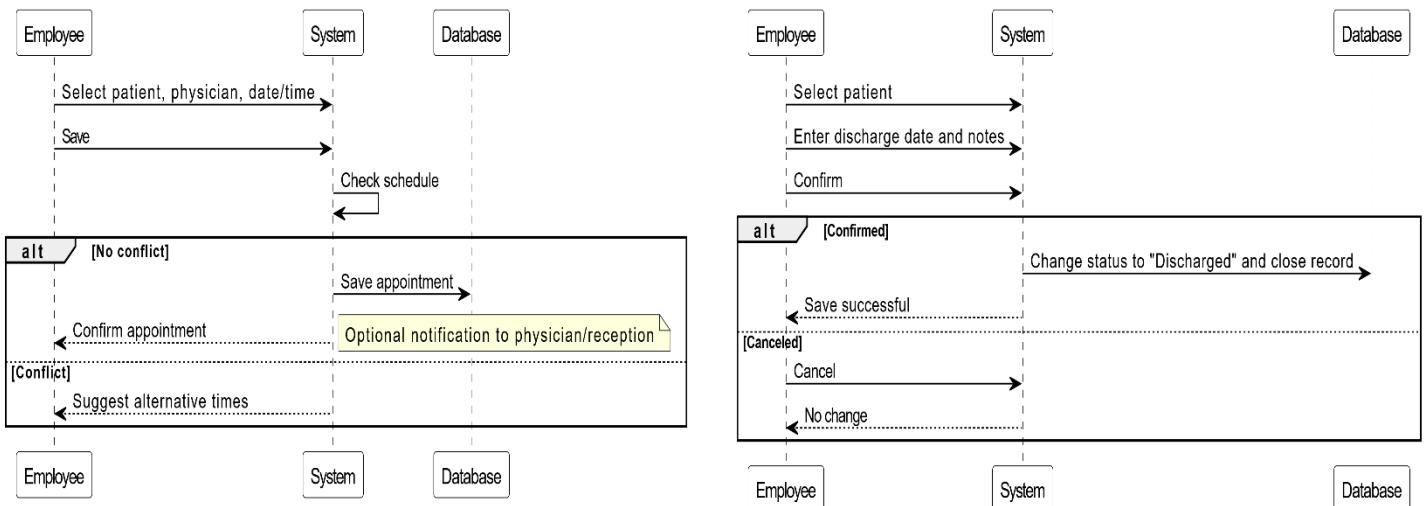


Figure (19) use cases 17,18

3.8.10 use cases 19,20

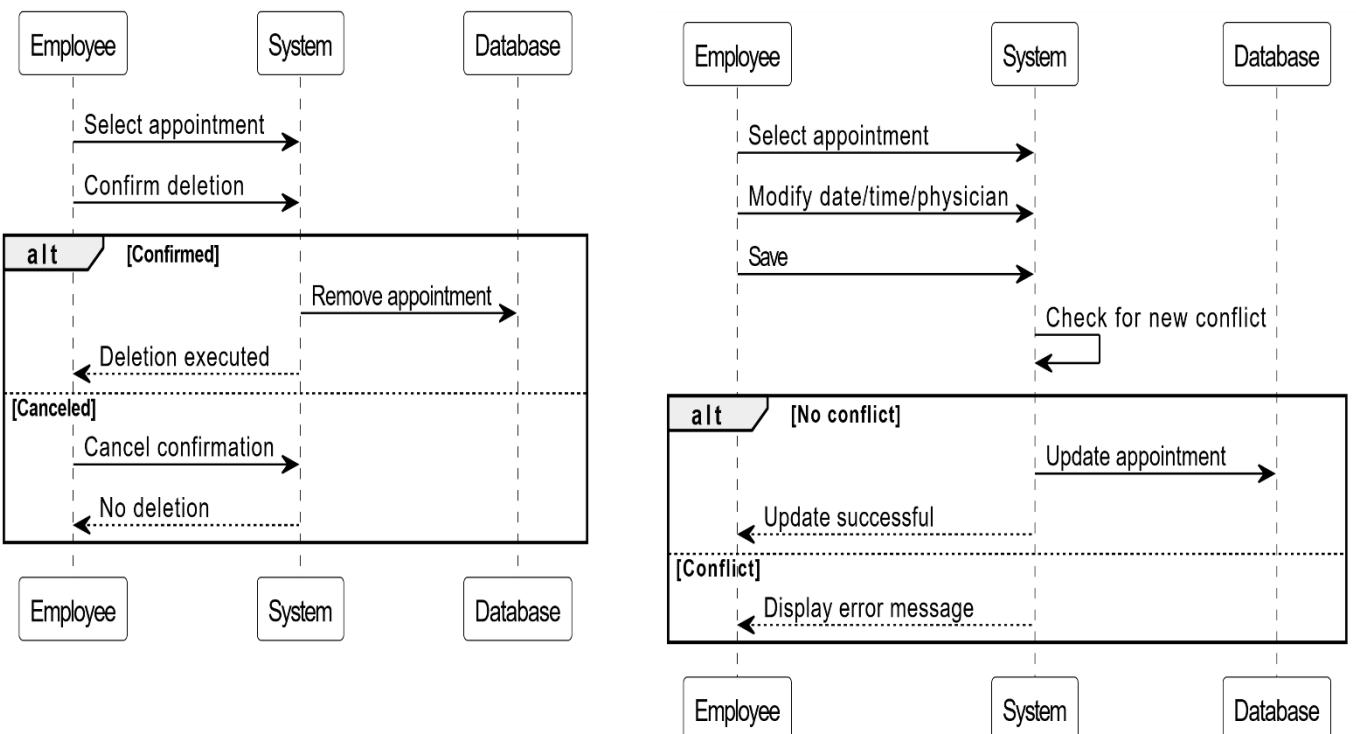


Figure (20) use cases 19,20

3.8.11 use cases 21,22

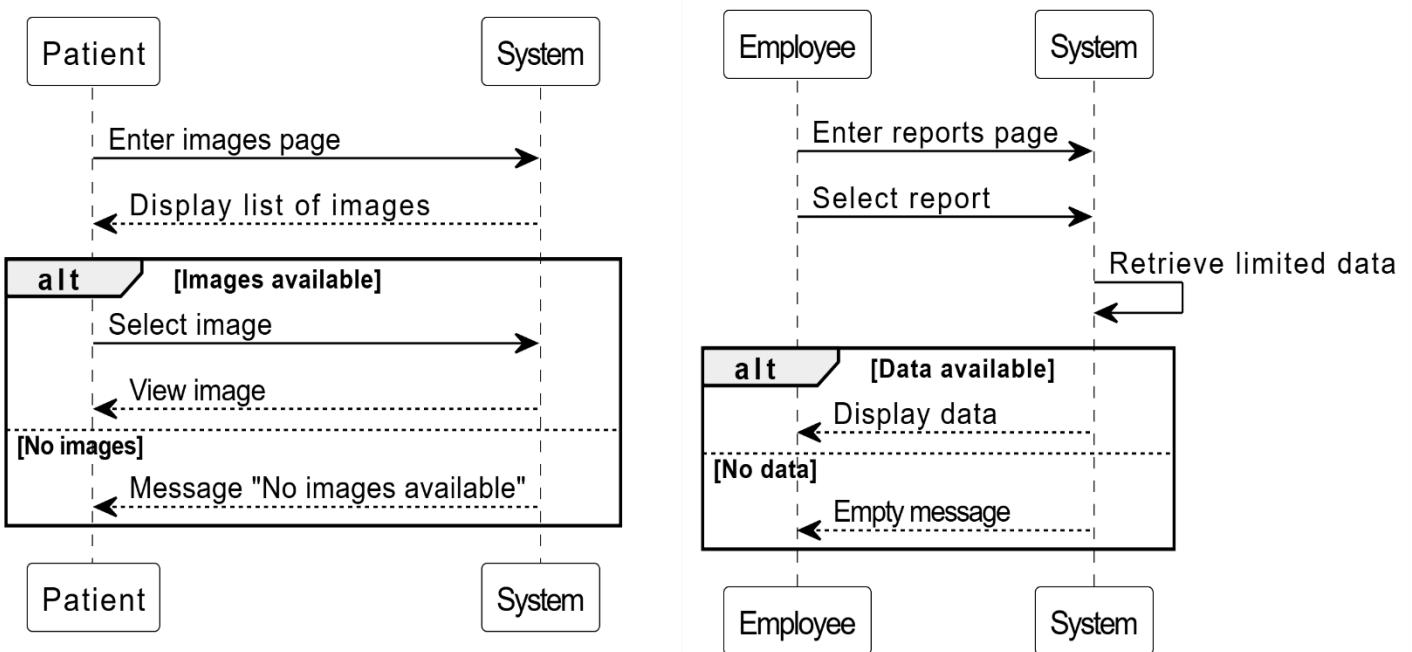


Figure (21) use cases 21,22

3.8.12 use cases 23,24

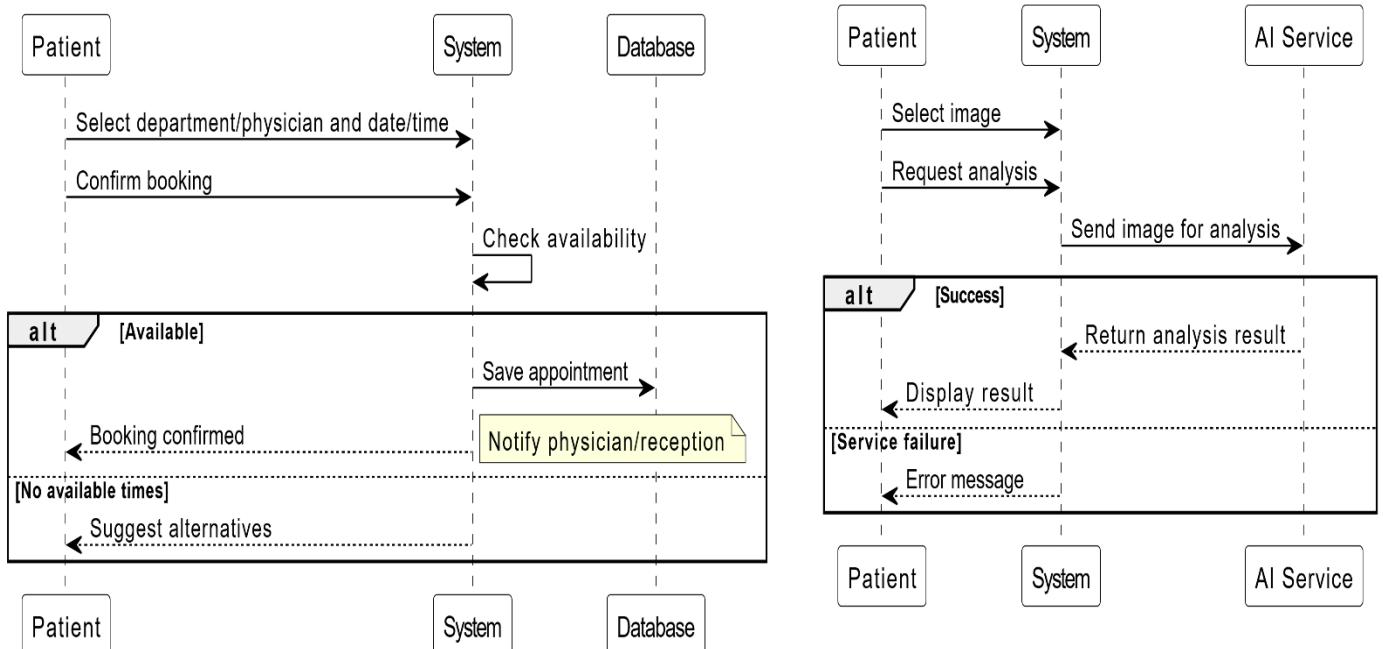


Figure (22) use cases 23,24

3.8.13 use cases 25,26

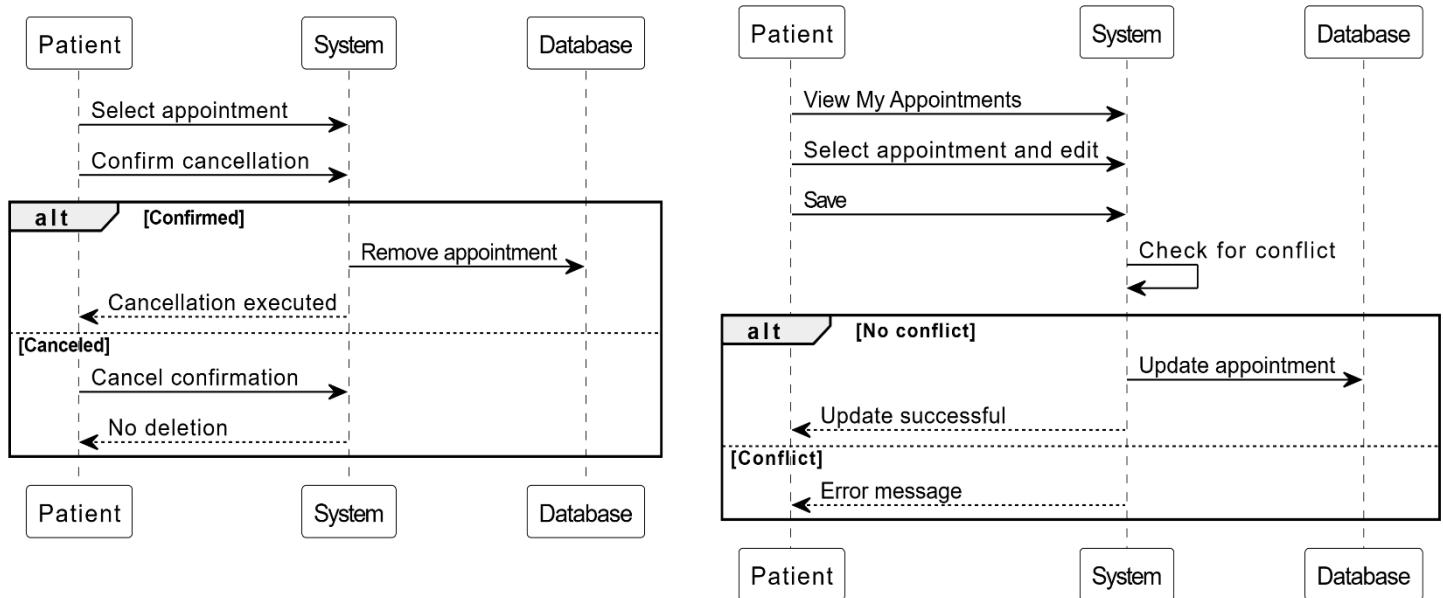


Figure (23) use cases 25,26

3.8.14 use cases 27,28

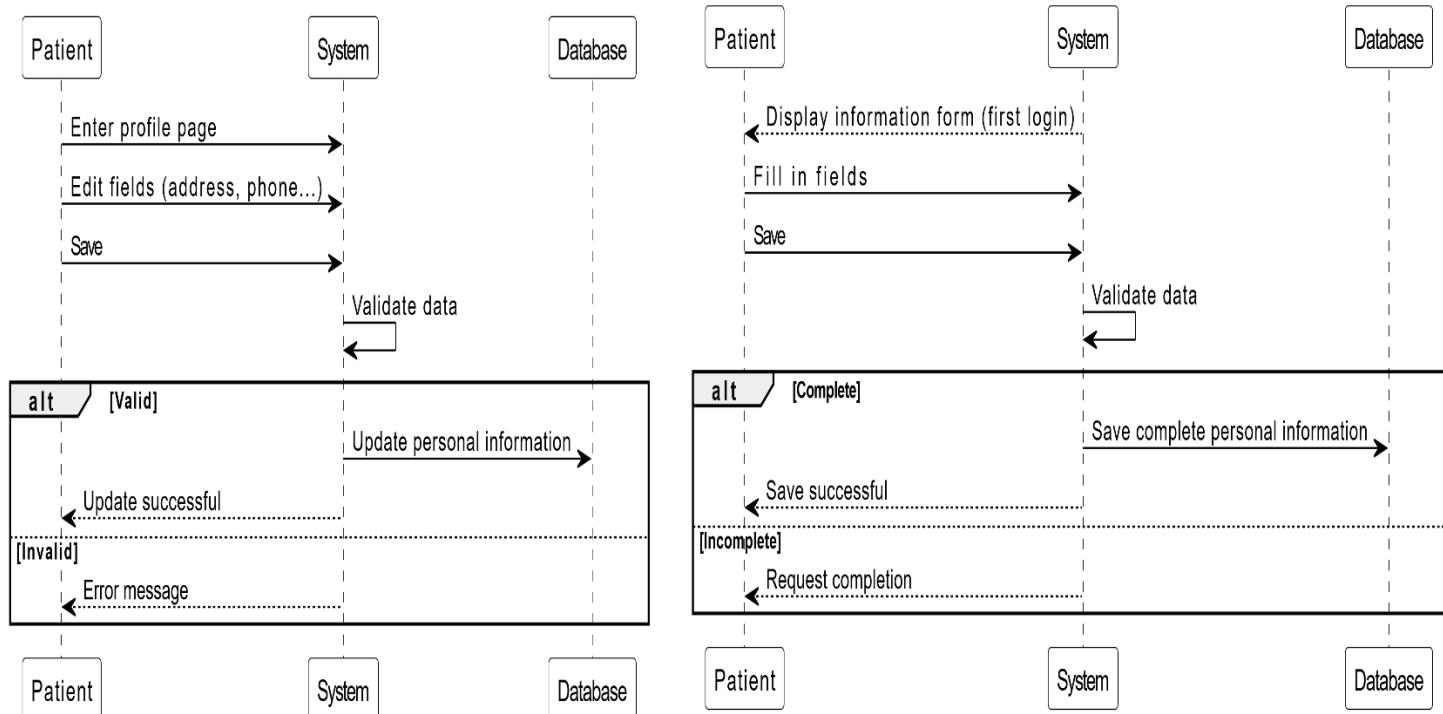


Figure (24) use cases 27,28

4. Database:

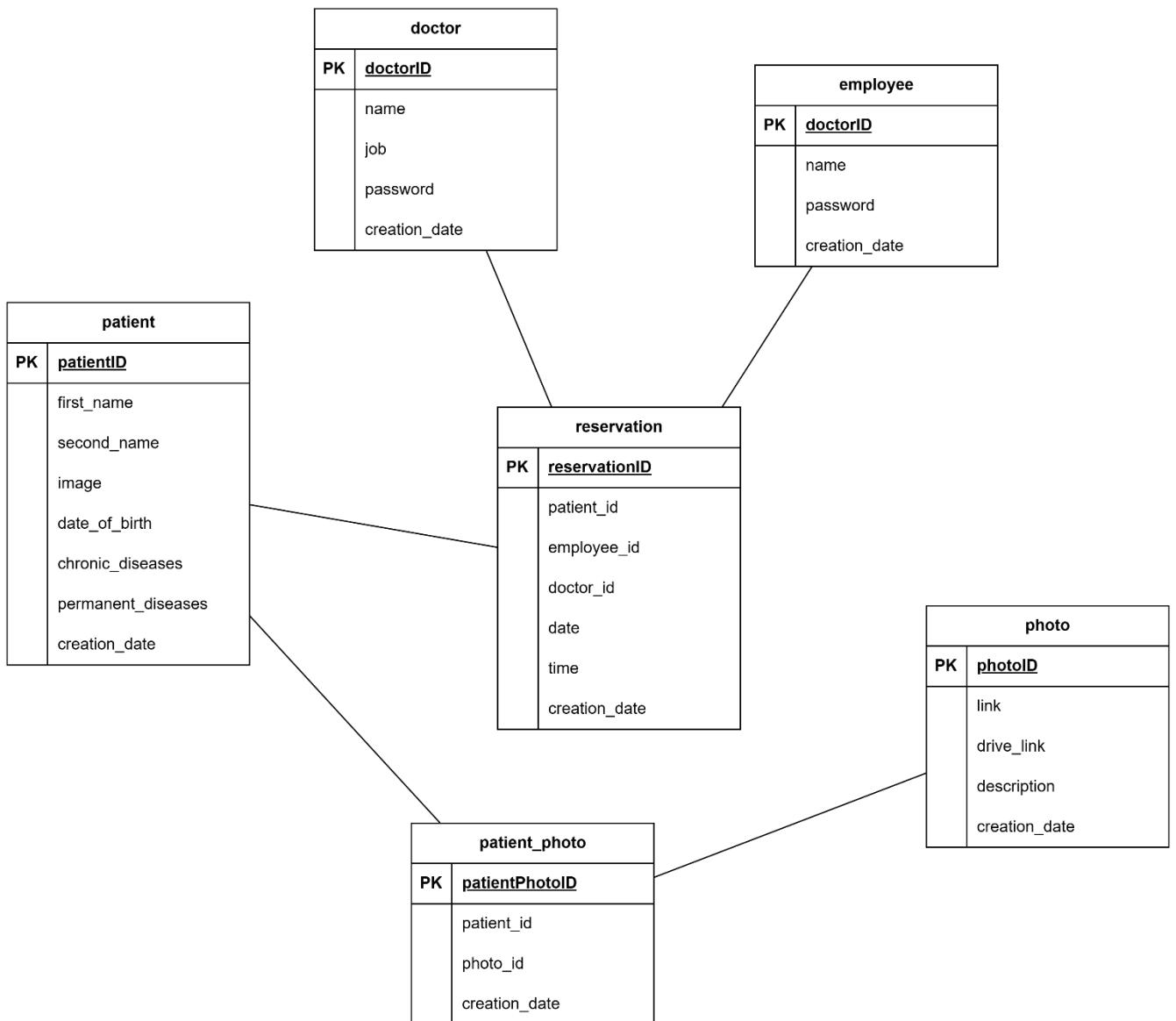


Figure (25) database

Chapter4

Implementation and Testing of Project

1.Log-In:



Figure (26) Log-In

On this page, the Admin and doctor and the employee enters his user's name and password, and then the site verifies the entered user name and password, and if they are correct, then it will move it to the main interface, but if the user's name or password is wrong, then it will show a message that there is no one with this information entered.

2.(Admin) Dashboard:

A screenshot of the Admin dashboard. On the left is a dark sidebar with navigation links: "RMS" (selected), "Employees" (highlighted in blue), "Doctors", and "Section". The main content area has a header "EMPLOYEE" with a "Home" link and a "Add New" button. Below is a table titled "All Employees" with two rows of data. The table columns are "Id", "Username", "Email", and "Operations".

ID	Username	Email	Operations
7	ammar	ammar_ali@gmail.com	<button>Edit</button> <button>Delete</button>
9	malek	malek_ali@gmail.com	<button>Edit</button> <button>Delete</button>

Copyright © 2026HMS All rights reserved.

Figure (27) Admin dashboard

On this page, the site displays all the information about the system, which include Employee Doctors and sections and from any of them he can show details.

3.(Employee) Appointment:

The screenshot shows the RMS application's appointment management page. The left sidebar has 'Patient', 'Appointment' (highlighted in blue), and 'Reports'. The main header is 'APPOINTMENT'. A 'Add New' button is at the top left. Below it is a table titled 'All Appointment' with columns: Id, Patient Name, Doctor Name, Date, Time, Section, Status, Creation Date, and Operations (Edit and Delete buttons). The data in the table is as follows:

Id	Patient Name	Doctor Name	Date	Time	Section	Status	Creation Date	Operations
1	khaled	ali	2025-12-15	13:27	Radiography	accepted	1970-08-20 14:33:19	<button>Edit</button> <button>Delete</button>
2	karlos		2025-12-15	13:46	Radiography		2603-10-11 14:33:19	<button>Edit</button> <button>Delete</button>
3	karlos		2025-12-15	13:42	Radiography		2025-12-15 13:42:30	<button>Edit</button> <button>Delete</button>

Figure (28) appointment

On this page, the employee manages the appointments and can edit their time and date and can delete the appointment

4.(Doctor) My Appointment

The screenshot shows the RMS application's 'My Appointment' page. The left sidebar has 'Radiographic', 'My Appointment' (highlighted in blue), and 'Reports'. The main header is 'MY_APPOINTMENT'. A 'All Appointment' button is at the top left. Below it is a table with the same structure as Figure 28, showing the same three appointments. The data is as follows:

Id	Patient Name	Doctor Name	Date	Time	Section	Status	Creation Date	Operations
1	khaled	ali	2025-12-15	13:27	Radiography	accepted	1970-08-20 14:33:19	<button>Edit</button>
2	karlos		2025-12-15	13:46	Radiography		2603-10-11 14:33:19	<button>Edit</button>
3	karlos		2025-12-15	13:42	Radiography		2025-12-15 13:42:30	<button>Edit</button>

At the bottom, there is a copyright notice: 'Copyright © 2026HMS All rights reserved.'

Figure (29) My appointment

On this page, the doctor will view its appointment so he can accept them and enrol with the patient

The Dashboard was created based on the previously explained interface diagrams, and the two types of users who will use the panel were implemented. A file was built containing a permissions determination matrix that determines what pages the user is allowed to access and prevents him from accessing pages that are outside his permissions.

This way we ensure that no one has access to data outside their authority

```
<?php
$tabs=array("employee">1,"radiographic">1,"doctor">1,"patient">1,"reports">1,"section">1,"";
if($admin_arrays[0]['admin_level']==1){
    $tabs['doctor']=1;$tabs['reports']=0;$tabs['radiographic']=0;$tabs['employee']=1;$tabs['patient']=0;
}
else if($admin_arrays[0]['admin_level']==2){
    $tabs['doctor']=0;$tabs['reports']=1;$tabs['radiographic']=0;$tabs['employee']=0;$tabs['patient']=1;
}
else if($admin_arrays[0]['admin_level']==3){
    $tabs['doctor']=0;$tabs['reports']=1;$tabs['radiographic']=1;$tabs['employee']=0;$tabs['patient']=1;
}
function out(){
    print '<meta http-equiv="refresh" content="0;URL=index.php?m=home&a=out" />';
    die;
}
if(isset($_GET['m']) && $_GET['m']=='employee'){
    if($admin_arrays[0]['admin_level']==2)
    {
        out();
    }
}
```

Figure (30) levels.php

If the user accesses a page that is outside his scope of authority, he will be directed to the unauthorized access page

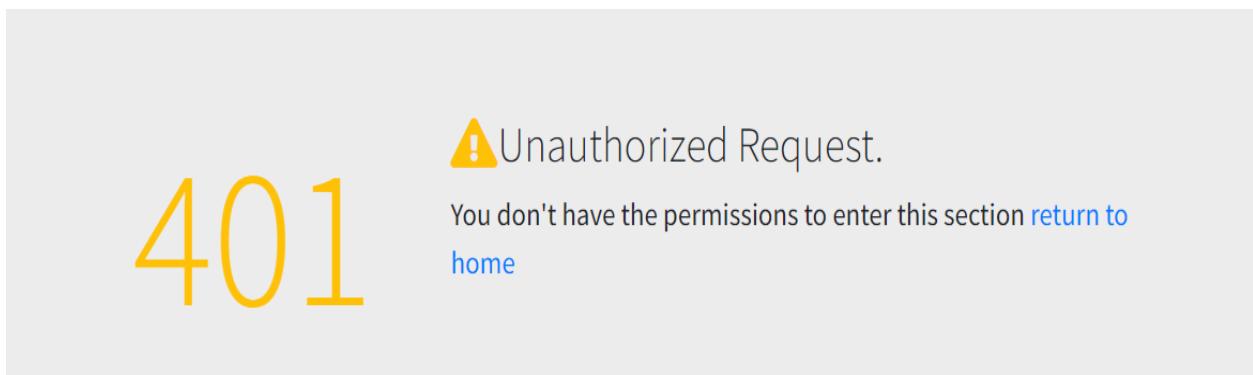


Figure (31) unauthorized

or he will be directed to the error page if he accesses any page in a wrong way

404 ERROR PAGE

⚠ Oops! Page not found.

404

We could not find the page you were looking for. Meanwhile,
you may [return to home](#)

Figure (32) error

The autoload.php file has been added, which compares the user's level with his permissions within the level permissions matrix in the level.php file. It also downloads the JavaScript files related to the loaded page.

```
require('functions/levels.php');
if(isset($_GET['m']) && isset($_GET['a'])){
    if($_GET['m'] != '' && $_GET['a'] != ''){
        $m = $_GET['m'];
        $a = $_GET['a'];
        if(file_exists('modules/' . $m)){
            if(file_exists('modules/' . $m . '/' . $a . '.php')){
                $html_require = 'modules/' . $m . '/' . $a . '.php';
                $html_header = strtoupper($m);
                if(file_exists('modules/' . $m . '/script.js')){
                    $html_require_script = 'modules/' . $m . '/'
                        script.js';|
```

Figure (33) autoload.php

The conf.php file was created and most of the variables used in most of the files were installed in order to achieve the principle of reuse in order to reduce the amount of code written and achieve the same result.

```
<?php
ini_set('memory_limit', '500M');
error_reporting();
date_default_timezone_set("Asia/Damascus");
require("db.class.php");
$full_site_url = '';
$user = 'root';
$password = '';
$server = 'localhost';
$dbname = 'hms_db';
$db = new db;
$db->connect($dbname , $server , $user , $password);
?>
```

Figure (34) conf.php

Then we created the functions that will allow us to connect to the database to bring the data from it in a class which will allow us to combine the front end and the back end together while using the conf.php information to link to the right data base

```
1 <?php
2 class db{
3     var $dbConnect;
4     function connect($db , $server = "localhost" , $user ="root" ,
5                     $pass ="" ){-
6         }
7         function select($query){-
8             }
9             function select_where($query , $where = 1){-
10                }
11                function update($table , $values , $where = 1){-
12                    }
13                    function insert($table , $values ){-
14                        }
15                        function delete($table , $where){-
16                            }
```

Figure (35) db.class.php

Testing:

Number	Requirement Number	Title	Test Case	Type	Expected Result
TC-1	REQ-1	Add Employee	The manager enters all correct data → Employee is added Leaving a required field empty → Error message	Positive/Negative	Addition success / Prevent addition with error message
TC-2	REQ-2	Edit Employee	Edit existing employee's data → Save is done	Positive	Data update / Reject edit
TC-3	REQ-3	Delete Employee	Confirm deletion → Deletion is done Cancel deletion → No deletion	Positive/Negative	Permanent deletion / Record remains
TC-4	REQ-4	Add Department	The manager enters all correct data → Department is added Leaving a required field empty → Error message	Positive/Negative	Addition success / Prevent addition with error message
TC-5	REQ-5	Edit Department	Edit existing department's data → Save is done	Positive	Data update
TC-6	REQ-6	Delete Department	Confirm deletion → Deletion is done Cancel deletion → No deletion	Positive/Negative	Permanent deletion / Record remains
TC-7	REQ-7	Add Doctor	Enter complete and correct data → Doctor is added Not adding a required field → Reject	Positive/Negative	Success / Message "Username exists"
TC-8	REQ-8	Edit Doctor	Edit specialty or schedule → Update is done	Positive	Data update success
TC-9	REQ-9	Delete Doctor	Confirm doctor deletion → Success Reject doctor deletion → Reject	Positive/Negative	Deletion / Reject with message
TC-10	REQ-10	View Reports (Admin)	Display dashboard → Shows correct statistics (number of patients, bookings, etc.)	Positive	Display data accurately
TC-11	REQ-11	Add Radiological Image	Upload valid image file → Upload and link to patient Upload unsupported file → Reject	Positive/Negative	Upload success / Error message
TC-12	REQ-12	Add Diagnosis	Write diagnosis and treatment → Saved with patient's image	Positive	Save diagnosis with date
TC-13	REQ-13	Query Patient (Doctor)	Search by name or number → Display full medical history	Positive	Display all related records
TC-14	REQ-14	Request AI Assistance	Upload image and request analysis → AI returns result Without image → Error message	Positive/Negative	Display analysis / Message "Please upload image"
TC-15	REQ-15	Add New Patient	Enter complete data → Create new patient file	Positive	Registration success and new file number

TC-16	REQ-16	Edit Patient Data	Edit phone number or address → Update is done	Positive	Data update
TC-17	REQ-17	Discharge Patient	End visit and issue invoice → Change patient status to "Discharged"	Positive	Update status and issue invoice
TC-18	REQ-18	Add Booking (Employee)	Book appointment → Success and save appointment	Positive	Booking confirmation
TC-19	REQ-19	Edit Booking (Employee)	Change time or doctor → Update booking	Positive	Successful update
TC-20	REQ-20	Delete Booking (Employee)	Cancel booking → Delete with patient notification if possible	Positive	Delete booking and update schedule
TC-21	REQ-21	View Reports (Employee)	Display daily reports → Shows correct numbers	Positive	Display daily statistics accurately
TC-22	REQ-22	View Radiological Images (Patient)	Patient login → Display all images with download option	Positive	Display and download list
TC-23	REQ-23	Request AI Diagnosis (Patient)	Upload or select image → Returns analysis Without image → Error message	Positive/Negative	Display result / Error message
TC-24	REQ-24	Add Booking (Patient)	Select specialty and available time → Successful booking	Positive	Booking confirmation and notification
TC-25	REQ-25	Edit Booking (Patient)	Edit before appointment → Success After appointment → Reject	Positive/Negative	Update / Message "Cannot edit"
TC-26	REQ-26	Delete Booking (Patient)	Cancel with sufficient time → Success At the last moment → Reject or fine	Positive/Negative	Successful cancel / Reject
TC-27	REQ-27	Add Personal Information (Patient)	First login → Mandatory form → Save data	Positive	Complete personal profile
TC-28	REQ-28	Edit Personal Information (Patient)	Edit personal information → Save changes	Positive	Data update success

Table (23) test cases

Chapter 5

Conclusion and Next Stages

Conclusion

In today's world, technology has become an essential part of healthcare management, particularly in medical centers and radiology departments. Medical staff, doctors, employees, and patients increasingly seek modern, efficient, accurate, and secure ways to manage patient data, appointments, radiological images, diagnoses, and treatment follow-up — ways that reduce errors, save time, and improve the quality of care.

Traditional manual methods of managing patient records, appointments, and radiological imaging often lead to errors, delays, lost information, duplicate records, and conflicts between staff and patients.

This project was designed to meet these evolving needs by creating a comprehensive, integrated digital management system for radiology centers and diagnostic imaging departments. Through our system:

1. Doctors can easily add and link radiological images to patients, write accurate diagnoses with dates, view complete patient medical histories, and request AI-assisted analysis for faster and more precise interpretations.
2. Employees (reception & administrative staff) benefit from powerful tools to register new patients, edit their information, manage appointments (add/edit/cancel), discharge patients, and generate daily and statistical reports.
3. Administrators have full control through an easy-to-use dashboard that allows them to add, edit, and delete employees, doctors, and departments, as well as view comprehensive center performance reports (number of patients, bookings, imaging procedures, etc.).
4. Patients gain direct access to their radiological images with download options, can request AI-based preliminary analysis, book/modify/cancel their own appointments (within allowed time limits), and complete/update their personal information securely.

The platform ensures accessibility anytime and anywhere across devices, reduces administrative burdens, minimizes errors, enhances communication between all parties, and improves overall patient experience and care quality. We believe this project has successfully achieved its core objectives: transforming the management of radiology centers into a modern, efficient, secure, and patient-centered digital experience that increases accuracy, accelerates processes, reduces waiting times, and builds trust among patients, doctors, and administrative staff.

Future Enhancements (Senior Project 2 / Next Phase):

1. Integrate advanced AI models with higher accuracy for automatic detection of common radiological findings (with final doctor approval).
2. Develop mobile applications (Android & iOS) for doctors, employees, and patients to improve accessibility and receive real-time notifications about appointments, results, and discharges.
3. Add electronic payment integration for appointment fees and imaging services.
4. Implement a multilingual interface (Arabic + English) with right-to-left support for better usability in Arabic-speaking countries.

5. Add features for sharing radiological images and reports securely with other hospitals or referring physicians.
6. Develop advanced analytics and predictive reports (e.g., peak booking times, most requested imaging types) to support administrative decision-making.
With this strong foundation in place, we are confident that the Radiology Center Management System has the potential to become a leading digital solution in the field of diagnostic imaging management in Syria and the region. The journey continues — toward even more intelligent, integrated, and patient-focused healthcare experiences

Chapter6

References and Appendices:

References

- [1] Alan Dennis, Systems Analysis and Design,2012
- [2] Joel Murach, Murach's PHP and MySQL,2010.
- [3] Luke Welling, PHP and MySQL Web Development, Second Edition,2003.
- [4] Alan Forbes, The Joy of PHP: A Beginner's Guide to Programming Interactive Web Applications with PHP and MySQL,2012.
- [5] Jake Spurlock, Bootstrap: Responsive Web Development ,2013.
- [6] Matt Lambert., Learning Bootstrap, 2014.
- [7] Marijn Haverbeke, Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming, 2018.

Appendices:

Appendix A: PHP Programming Language

PHP (which stands for **PHP: Hypertext Preprocessor**) is a widely used, open-source, general-purpose scripting language especially suited for **web development**. It was originally created by Rasmus Lerdorf in 1994 as a set of Perl scripts called "Personal Home Pages" (PHP/FI). Over time, it evolved into a full-fledged programming language.

Key milestones in its history:

- In 1995, the first public release appeared.
- PHP 3 (1998) introduced a major rewrite, added object-oriented programming basics, and redefined PHP as "Hypertext Preprocessor" — a recursive acronym.
- Subsequent versions (PHP 4, 5, 7, and 8 series) brought significant improvements in performance (Zend Engine), security, modern syntax (e.g., JIT compiler in PHP 8), typed properties, attributes, enums, and more.
- As of January 2026, the latest stable versions are from the **PHP 8.5** series (released late 2025), with PHP 8.6 under active development for release by the end of 2026.

Main Advantages of PHP (relevant to our educational platform):

1. **Open-source and free** — full access to source code, allowing unlimited modifications and community-driven improvements.
2. **Easy to learn and simple syntax** — ideal for rapid development of interactive web applications like our platform.
3. **Cross-platform** — runs seamlessly on Windows, Linux, macOS, and various web servers (Apache, Nginx, etc.).
4. **Native HTML integration** — PHP code can be embedded directly into HTML, making it perfect for creating dynamic interfaces.
5. **High flexibility and security** — strong support for modern security practices, sessions, authentication, and data validation — crucial for protecting users' progress data.
6. **Extensive ecosystem** — vast libraries, frameworks (Laravel, Symfony), and tools for building scalable web apps, including our management dashboards.
7. **Strong community and documentation** — official PHP.net site and global forums provide excellent support.

In our project, PHP serves as the backend language powering the API, user authentication, appointments management, and progress tracking— ensuring a fast, reliable, and maintainable system.

Appendix B: MySQL Database Management System

MySQL is one of the world's most popular open-source **relational database management systems (RDBMS)**. It uses **Structured Query Language (SQL)** for managing data and organizes information in tables with defined relationships.

History and Ownership:

- Created in 1995 by Michael "Monty" Widenius (named after his daughter "My").

- Initially developed by MySQL AB (Sweden).
- Acquired by Sun Microsystems in 2008, then by Oracle Corporation in 2010 (current owner).
- MySQL remains open-source under the GNU General Public License (GPL), with optional commercial editions from Oracle.
- As of January 2026, MySQL follows a dual-track release model: **Innovation releases** (new features) and **Long-Term Support (LTS)** releases (stable, long-lived bug/security fixes). MySQL 8.0 reaches End-of-Life in April 2026; recommended production versions are **MySQL 8.4 LTS** and newer **Innovation releases** (e.g., 9.x series).

Key Features of MySQL:

- Extremely fast performance — multi-threaded server design enables quick queries, even with large datasets.
- Easy integration — works seamlessly with PHP (via PDO or mysqli), making it the ideal database for our platform's backend.
- Robust security features — user authentication, encryption, role-based access control, and protection against SQL injection.
- Support for multiple storage engines (e.g., InnoDB for transactions and ACID compliance).
- Graphical user interfaces (GUI) — free tools like **phpMyAdmin**, **MySQL Workbench**, and **Adminer** allow visual management of databases, tables, and queries — very useful for administrators maintaining our educational content.
- Scalability and reliability — handles high concurrency, replication, and backups — perfect for growing user bases of admin, employees, and doctors. In our platform, MySQL stores all critical data: user accounts, Appointments, reports, and analytics — providing a secure, efficient, and scalable foundation for the entire system.

These appendices provide technical background on the core technologies used in the development of the platform, highlighting why PHP and MySQL were chosen for their proven reliability, ease of use, performance, and strong community support in building modern web applications.