

**Medical Imaging AI Integration**

**Team Members:**

**Andi Shyti  
Simeon Flamuraj  
Erina Gjikaj  
Tea Begaj**

1. **Introduction**  
     1.1 A Visionary Journey  
     1.2 Bridging Gaps in Diagnostics  
     1.3 Navigating the Narrative
2. **Medical Imaging AI Overview**  
     2.1 AI-Assisted Diagnosis Workflow  
     2.2 AI-Powered Image Analysis  
     2.3 Clinician Review and Interaction  
     2.4 Role-Based Access and Security  
     2.5 Key Features  
       2.5.1 Image Upload and Preprocessing  
       2.5.2 CAM Visualization and Confidence Scoring  
       2.5.3 Patient Records and History Tracking  
       2.5.4 Report Generation and Export  
       2.5.5 EMR Integration and Follow-Up Tools
3. **Scope**  
     3.1 Pneumonia Detection via Chest X-Rays  
       3.1.1 AI-Based Diagnostic Support  
       3.1.2 Clinical Efficiency and Accuracy  
       3.1.3 Visual Explainability for Clinicians  
       3.1.4 Integration with Hospital Systems  
       3.1.5 Use in Low-Resource Settings  
       3.1.6 Support for Desktop and Web
4. **Requirements**  
     4.1 Functional Requirements  
       4.1.1 Image Upload and AI Analysis  
       4.1.2 User Access and Reporting  
     4.2 Non-Functional Requirements  
     4.3 External Requirements
5. **Stakeholders**
6. **Scenarios**
7. **Use Cases**  
     7.1 User Registration  
     7.2 Profile and Role Management  
     7.3 X-Ray Upload and AI Diagnosis  
     7.4 AI Output Review and Editing  
     7.5 Report Approval and Sharing  
     7.6 Appointment Scheduling  
     7.7 Hospital Administration  
     7.8 Clinical Experts  
     7.9 IT and System Maintenance  
     7.10 Compliance and Security Officers
8. **BPMN Diagram**
9. **Activity Diagram**
10. **State Diagram**
11. **Collaboration Diagram**
12. **Sequence Diagram**
13. **Data Flow Diagram (DFD)**
14. **Class Diagram**  
      14.1 User  
      14.2 MedicalImage  
      14.3 AIModel  
      14.4 DiagnosisReport  
      14.5 CAMOverlay  
      14.6 Database  
      14.7 Appointment  
      14.8 EMRIntegration
15. **Factory Method Design Pattern**
16. **Abstract Factory Design Pattern**
17. **Proxy Design Pattern**
18. **Object Diagram**
19. **Package Diagram**
20. **Component Architecture**
21. **Deployment Diagram**

**1. Introduction**

Welcome to **Medical Imaging AI Integration** – an innovative solution built to enhance medical diagnostics through the power of artificial intelligence. In the evolving landscape of digital health, this application serves as a bridge between cutting-edge technology and clinical expertise, enabling faster, more accurate, and more efficient patient care.

**1.1 A Visionary Journey**

Step into a future where medical professionals are empowered with AI-driven insights. Imagine a diagnostic process where doctors, radiologists, and technicians can upload medical scans and instantly receive detailed, AI-assisted analyses. This application is more than a tool – it's a partner in transforming how medical imaging is interpreted.

**1.2 Bridging Gaps in Diagnostics**

Medical imaging is crucial in diagnosis, yet it often faces delays, subjective interpretations, and resource constraints. This project aims to fill these gaps by introducing an intelligent assistant that supports medical staff in identifying anomalies such as tumors, fractures, and infections. Through AI-powered diagnostics, it elevates both the speed and accuracy of medical evaluations.

**1.3 Navigating the Narrative**

This documentation will guide you through the core functionalities, technical foundation, and user-centric design of the application. You’ll explore how AI integrates with medical workflows to deliver annotated images, probability-based diagnoses, and structured reports—paving the way for smarter, data-informed healthcare decisions.

### 2. Medical Imaging AI Integration Overview

**Medical Imaging AI Integration** is a powerful application designed to assist medical professionals in analyzing diagnostic images using advanced artificial intelligence. It streamlines the workflow of radiologists, doctors, and technicians by providing fast, accurate, and explainable diagnostic insights directly from medical scans.

#### 2.1 AI-Assisted Diagnostic Support

* Medical professionals can upload imaging data—such as X-rays, MRIs, or CT scans—directly into the system.
* The AI model analyzes the image to detect possible abnormalities (e.g., tumors, fractures, infections).
* It presents results with highlighted regions of concern and confidence scores, helping professionals make informed decisions quickly.

#### 2.2 Intelligent Communication and Visualization

* The application displays visual annotations on the medical images (e.g., bounding boxes, heatmaps) to indicate areas of interest.
* Each detection is accompanied by diagnostic suggestions and a probability score, offering transparency and aiding in diagnosis validation.
* Professionals can compare current scans with previous results for the same patient, facilitating longitudinal analysis.

#### 2.3 User Roles and Secure Access

* Multi-user role support allows controlled access based on user type—doctors, radiologists, or administrators.
* Users log in through secure hospital credentials to ensure compliance with data privacy standards.
* Admins manage role assignments and oversee system usage, maintaining operational integrity.

#### 2.4 Patient History and Reporting

* Each patient’s imaging history is securely stored and organized for easy retrieval and comparison.
* Doctors can download or share AI-generated diagnostic reports, with the ability to append personal notes.
* Reports can be exported as PDF files or sent via secure email channels to other professionals for collaboration.

#### 2.5 Key Features

**2.5.1 Image Upload & Management**  
Users can upload images in JPEG, PNG, or DICOM format, view scan history, and filter by patient or date.

**2.5.2** **AI Analysis & Confidence Scores**  
Advanced AI models process images to detect anomalies and assign confidence levels to possible diagnoses.

**2.5.3 Visual Marking & Annotation**  
Abnormalities are clearly marked on images to guide medical professionals visually through the findings.

**2.5.4 Secure Role-Based Access**  
Supports secure login, encrypted data storage, and administrator-controlled access levels.

**2.5.5 Reporting & Export**  
AI-generated reports can be reviewed, edited with manual notes, and exported as PDFs or emailed.

**2.5.6 Feedback Loop**  
Doctors can validate AI predictions as correct or incorrect to help improve future model accuracy.

### 3. Scope

#### 3.1 Scope of the Medical Imaging AI Integration Project

The **Medical Imaging AI Integration** project is designed to support and enhance medical diagnostics through AI-powered imaging analysis. It addresses key challenges in medical imaging interpretation, streamlines clinical workflows, and promotes diagnostic accuracy.

**3.1.1 AI-Powered Medical Imaging Diagnostics**

* The application focuses on analyzing medical images (X-rays, MRIs, CT scans) using trained AI models to detect conditions such as tumors, fractures, or infections.
* It supports radiologists and doctors by providing second-opinion suggestions and visual indicators.

**3.1.2 Multi-Format Image Handling**

* Users can upload images in standard formats such as JPEG, PNG, and DICOM.
* The system provides tools for viewing, managing, and filtering uploaded images for each patient.

**3.1.3 Intelligent Analysis and Confidence Scoring**

* The AI engine processes medical images and returns results with confidence scores for each detected anomaly.
* Visual markers (e.g., bounding boxes, heatmaps) highlight regions of concern on the image.

**3.1.4 Secure Communication and Reporting**

* The application supports secure report generation and sharing, enabling collaboration between doctors and specialists.
* Reports include both AI-generated content and manual notes added by medical professionals.

**3.1.5 Role-Based User Management**

* The system is built to accommodate multiple user roles, including doctors, radiologists, and administrators.
* User permissions and data access are managed securely and in compliance with healthcare data regulations.

**3.1.6 Patient History and Longitudinal Monitoring**

* A core part of the scope includes maintaining a historical record of each patient’s imaging data and diagnoses.
* Users can compare past and present scans to track health progression over time.

#### 3.2 Usability for Medical Professionals and Healthcare Institutions

* The application is tailored primarily for hospital use, enabling medical staff to streamline diagnostics and reporting.
* It is also adaptable for broader clinical environments such as imaging centers, research facilities, or telemedicine platforms.

**4. Requirements**

**Functional Requirements**

**User Requirements**

**User Registration and Authentication**

* Users (doctors, radiologists, technicians) must be able to register securely on the platform.
* Registration must include essential details such as name, role, department, and contact information.
* Implement secure login with support for two-factor authentication (2FA).

**Profile Management**

* Users should have access to a personalized dashboard.
* Dashboard includes user role, access to patient data, and image history.
* Admins can manage user roles and permissions.

**Image Upload and Management**

* Users can upload medical images in JPEG, PNG, and DICOM formats.
* Each image must be linked to a patient ID and metadata (scan date, modality, notes).
* Users should be able to view, search, and filter image history.

**AI-Powered Image Analysis**

* Users can request AI analysis on uploaded images.
* The AI will detect abnormalities and generate confidence scores.
* Results are shown with visual aids (e.g., bounding boxes, heatmaps).

**Reporting and Documentation**

* AI analysis results can be exported as PDF reports.
* Users can add manual comments or notes to reports.
* Reports can be shared securely with other specialists.

**Comparison and History Tracking**

* Allow users to compare current scan results with previous images of the same patient.
* Visual and textual differences should be highlighted.

**Communication and Feedback**

* Users can mark AI suggestions as "Correct" or "Incorrect."
* Include feedback forms to assess AI performance.
* Doctors can leave notes about diagnostic decisions based on AI output.

**Appointment Scheduling (Optional)**

* Allow doctors to schedule follow-up diagnostics or consultations.
* Notify users of scheduled events via email or in-app alerts.

**System Requirements**

**Security and Privacy**

* Encrypt all sensitive data, including medical images and user details.
* Comply with GDPR and HIPAA for healthcare data handling.
* Perform regular penetration testing and vulnerability scanning.

**Scalability**

* Use cloud-native architecture for horizontal scalability.
* Support growing user and data volume, including heavy image data loads.
* Implement load balancing and optimized database queries.

**Cross-Platform Compatibility**

* Ensure responsive UI for desktops and tablets used in clinical settings.
* Build using a cross-platform framework like React, Angular, or Electron for desktop.
* Future-proof for mobile expansion using Flutter or React Native.

**Data Storage and Retrieval**

* Store images securely in an encrypted database or cloud bucket.
* Use indexing and metadata tagging for fast image retrieval.
* Archive old image data to optimize performance.

**AI Integration**

* Ensure real-time, API-based communication with the AI model.
* The model should be containerized (e.g., via Docker) for modular deployment.
* Allow updates to AI logic without disrupting system functionality.

**Nonfunctional Requirements**

**Performance**

* Application response time must remain under 2 seconds.
* Capable of handling 10,000+ concurrent image processing requests.

**Reliability**

* Target uptime of 99.9% with auto-recovery mechanisms.
* Use health checks and monitoring tools for continuous availability.

**Security**

* Enforce role-based access control (RBAC).
* Use HTTPS, data encryption at rest and in transit, and strong password policies.

**Usability**

* UI must be clean, intuitive, and accessible for users with varying levels of tech proficiency.
* Follow accessibility standards (WCAG 2.1).

**Scalability**

* System should scale to accommodate 20% growth in users and image volume per month.

**Compatibility**

* Support major browsers (Chrome, Firefox, Edge).
* Ensure compatibility with hospital IT systems.

**Maintainability**

* Follow modular, microservices-based architecture.
* Maintain clean, well-documented code and clear API documentation.

**Regulatory Compliance**

* Ensure compliance with healthcare-specific regulations (HIPAA, GDPR).
* Perform regular audits and ensure traceable logging.

**Interoperability**

* Enable future integration with EMR/EHR systems via standard HL7 or FHIR APIs.
* Allow export/import of reports in standard formats (PDF, JSON, HL7).

**Collaboration**

* Maintain feedback loops with medical professionals during testing.
* Involve radiologists and doctors in model validation and usability reviews.

### External Requirements

**Collaboration with Healthcare Professionals**

* Involve radiologists, doctors, and imaging specialists throughout the development lifecycle.
* Use their expertise to validate AI-generated diagnostic suggestions and visual outputs.
* Ensure all AI decisions reflect current clinical standards and diagnostic protocols.
* Schedule regular review sessions with professionals to evaluate the AI’s performance on real-world data.

**Collaboration with End-Users (Medical Staff)**

* Engage medical imaging technicians, junior doctors, and radiology department staff for usability testing.
* Collect feedback on ease of image upload, analysis flow, and clarity of AI results.
* Adapt interface and reporting tools based on feedback to improve usability and reduce diagnostic time.
* Ensure that the system meets daily workflow needs without disrupting existing processes.

### 5. Stakeholders

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Stakeholder Name** | **Role/Responsibility** | **Importance** | **Influence** | **Positive Impacts/Interests** | **Concerns** | **Communication Needs** |
| **Hospital Administrators** | Oversee operations and approve tech adoption | High | High | Improved diagnostic accuracy, streamlined imaging workflows | Data privacy, cost of implementation, legal liability | Project milestones, cost-benefit analysis, performance summaries |
| **Radiologists** | Primary users analyzing AI results | High | High | Faster, AI-assisted diagnoses, visual annotations | Misdiagnoses, false positives/negatives | Real-time AI output, confidence scores, image annotations |
| **Doctors** | Use AI results in treatment decisions | High | Medium | Quicker decision-making, better treatment planning | Clarity of AI results, ease of use | Summary reports, PDF exports, patient scan comparisons |
| **Patients** | Indirect users whose scans are analyzed | High | Low/Medium | More accurate, faster diagnostics | Misdiagnosis, privacy of medical images | Clear consent forms, transparency about AI involvement |
| **IT & Development Team** | Develop, secure, and maintain the platform | Medium | High | Technical implementation success, scalability | Technical debt, system bugs | Sprint reviews, system logs, error reports, AI model feedback |
| **Hospital Admin Staff** | Coordinate appointments and patient data | Medium | Medium | Streamlined workflows, automated documentation | Workflow disruptions, system learning curve | Integration documentation, training sessions |
| **Regulatory Authorities** | Ensure legal and ethical compliance | High | High | Patient safety, data governance | Compliance violations, lack of audit trails | Regular audits, policy updates, compliance certification documents |
| **Medical Imaging Technicians** | Upload scans and manage imaging data | Medium | Medium | Simplified upload and storage, faster feedback loop | Complexity of UI, technical issues | Upload procedures, training on supported formats (JPEG, PNG, DICOM) |
| **Admin (System Admins)** | Manage user roles, access, and system health | Medium | Medium | Controlled access, smooth performance | Role mismanagement, downtimes | Logs, access reports, performance monitoring tools |
| **End Users (External Clinics)** | Use platform for AI support without full integration | Low | Low | Access to AI diagnostics, low-cost second opinions | Trust in AI, data ownership | Promotional materials, trial feedback, usage reports |

### 6. SCENARIOS

#### ****Scenario 1: AI-Assisted Diagnosis****

**User**: Dr. Laura, a radiologist reviewing chest X-rays.

1. **Scan Upload**:  
   • Dr. Laura logs into the platform and uploads a patient's chest X-ray.
2. **AI Analysis**:  
   • The system automatically analyzes the image and highlights suspected anomalies (e.g., nodules, fluid buildup).  
   • AI provides a confidence score and possible diagnosis (e.g., pneumonia, tumor).
3. **Decision Support**:  
   • Dr. Laura uses the AI's annotations to cross-check her analysis.  
   • She finalizes the diagnosis, using the AI as a second opinion.

#### ****Scenario 2: Patient Record Review with Imaging****

**User**: Dr. Mark, a physician preparing for a patient consultation.

1. **Patient Profile Access**:  
   • Dr. Mark opens the patient’s profile on the hospital system integrated with the Medical Imaging AI.
2. **Imaging Insights**:  
   • He reviews AI-analyzed CT scans and MRI images, which include flagged regions and clinical notes.
3. **Treatment Planning**:  
   • Based on AI suggestions and his own expertise, he prepares a personalized treatment plan.

#### ****Scenario 3: Technician Upload Workflow****

**User**: Mia, a medical imaging technician.

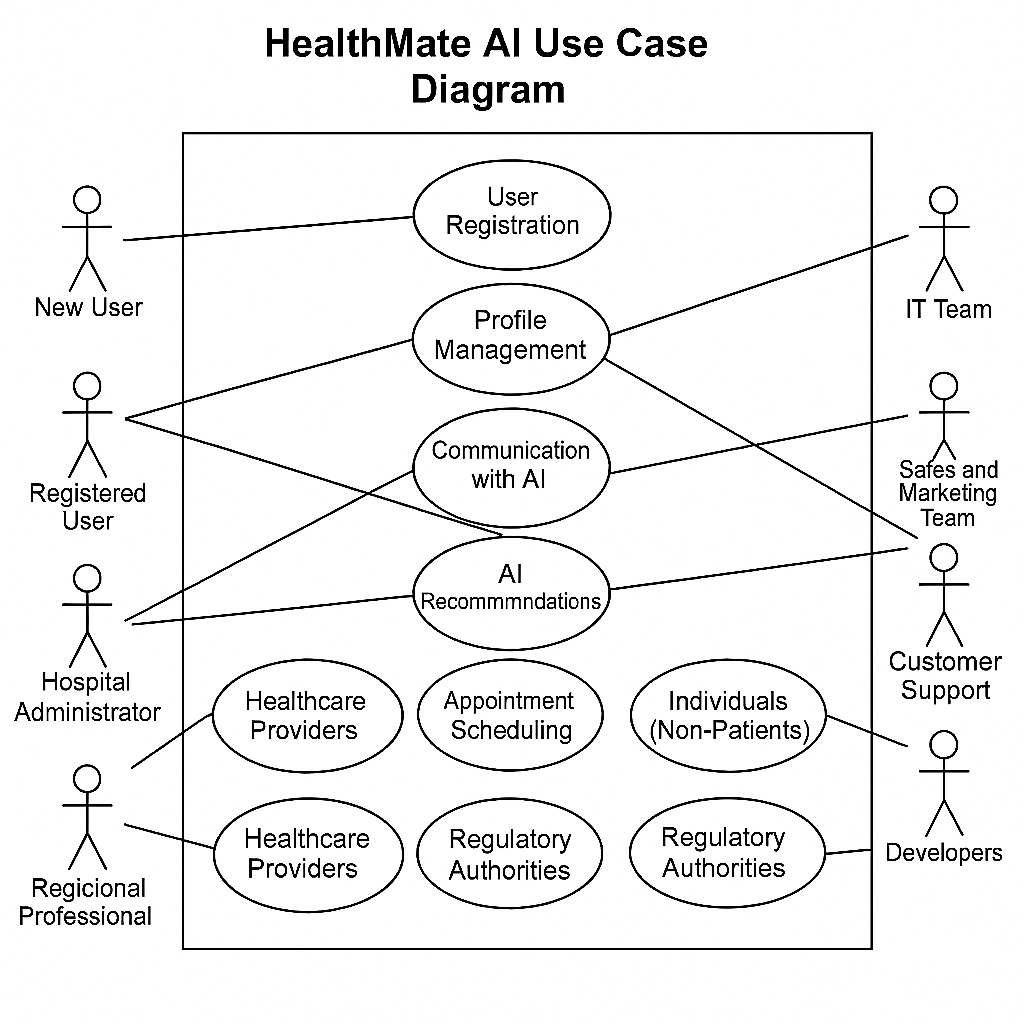
1. **Scan Handling**:  
   • Mia completes an MRI scan and uploads the DICOM file through the AI platform.
2. **AI Processing**:  
   • The system automatically converts and analyzes the scan in the background.  
   • Within seconds, it provides visual overlays and metadata.
3. **Technician Review**:  
   • Mia verifies the AI upload success and tags the case for radiologist review.

#### ****Scenario 4: Patient-Facing Explanation****

**User**: Emma, a patient receiving her results.

1. **AI Summary Request**:  
   • Emma receives a notification about her recent scan results.
2. **Simplified Output**:  
   • The system presents a patient-friendly summary of findings (e.g., “No signs of fracture, normal alignment”).
3. **Doctor Follow-Up**:  
   • She is scheduled for a follow-up to discuss results in more detail with her physician.

**7. Use Case**



This section presents an in-depth overview of how different user types interact with the Medical Imaging AI Integration platform. The use cases reflect real-world functionality aimed at enhancing diagnostic efficiency, patient engagement, and healthcare delivery through artificial intelligence and medical imaging technologies.

### ****7.1 User Registration****

**Actors:** New User  
**Description:**  
New users can initiate the registration process by providing essential information such as full name, date of birth, contact details (email, phone number), and preferred methods of communication (e.g., push notifications, SMS, email). The system includes data validation, captcha verification, and encryption protocols to ensure secure and authenticated registration. Once verified, the system creates a unique user ID and stores the information in compliance with medical data privacy laws (e.g., HIPAA, GDPR). This registration serves as the first step in enabling access to personalized AI-powered healthcare services.

### ****7.2 Profile Management****

**Actors:** Registered User  
**Description:**  
After successful registration, users gain access to a personal profile dashboard where they can manage and update their information at any time. This includes health-related data such as allergies, existing medical conditions, prior diagnoses, surgical history, and emergency contacts. The profile also includes configurable settings such as preferred language, notification frequency, and interface themes. The system ensures data integrity and confidentiality while allowing the AI system to better tailor its feedback and recommendations to the user's health background.

### ****7.3 Health Monitoring****

**Actors:** Registered User  
**Description:**  
Users can manually input or sync health-related data such as blood pressure, glucose levels, medication adherence, and symptoms experienced post-surgery or during treatment. The interface is designed to be user-friendly, supporting data entry via forms, wearable device integration, and voice input. The system processes this data and displays graphical trends, comparisons over time, and alerts for abnormal readings. This module enhances early detection of complications and improves the user's ability to actively engage with their health status.

### ****7.4 Communication with AI****

**Actors:** Registered User  
**Description:**  
Users are empowered to communicate with the AI system through a natural language interface. They can describe symptoms, ask health-related questions, request explanations of medical terms, or even upload medical images such as X-rays, MRIs, or CT scans. The AI leverages NLP (Natural Language Processing) and computer vision to interpret input in real time and respond with human-like, medically-informed replies. The interface supports multilingual interaction and allows follow-up questions for clarification, simulating a conversational experience similar to speaking with a healthcare professional.

### ****7.5 AI Recommendations****

**Actors:** Registered User  
**Description:**  
The AI engine analyzes the user’s profile data, health monitoring inputs, and submitted medical images to provide personalized health recommendations. These may include advice on medication adherence, warning signs for potential complications, suggestions for lifestyle changes, or alerts to seek professional medical attention. The system emphasizes transparency by showing the reasoning behind each recommendation, referencing medical data or guidelines used in decision-making. Users can review and save these recommendations or share them with healthcare professionals.

### ****7.6 Appointment Scheduling****

**Actors:** Registered User, Healthcare Professional  
**Description:**  
Users can browse available time slots and schedule appointments with doctors, radiologists, or other healthcare specialists. The calendar-based interface includes filters for specialty, availability, and language preferences. The system sends confirmation emails and reminders and supports cancellations or rescheduling with just a few clicks. Healthcare professionals also have access to appointment management tools to organize their schedules efficiently. The system maintains a full appointment history for both patients and providers.

### ****7.7 Hospital Administration****

**Actors:** Hospital Administrator  
**Description:**  
Administrators manage the deployment and usage of the AI system within a hospital environment. They can oversee user accounts, assign roles to staff, configure access permissions, and ensure that hospital policies are implemented correctly. Administrators also monitor usage metrics, system performance, compliance logs, and incident reports. Their dashboard includes tools to generate insights about how the system supports clinical workflows, supports audits, and flags areas for improvement or retraining of the AI module.

### ****7.8 Medical Imaging Upload****

**Actors:** Registered User  
**Description:**  
Users can securely upload medical imaging files (e.g., DICOM, JPEG, PNG) either manually or through integrations with hospital imaging systems (PACS). Upon upload, the images are pre-processed and analyzed using advanced deep learning algorithms trained to detect abnormalities such as tumors, fractures, or lesions. The system flags anomalies and highlights regions of interest, offering preliminary assessments which can be reviewed by healthcare professionals. Uploads are stored securely and can be reused for follow-ups or second opinions.

### ****7.9 Healthcare Providers****

**Actors:** Doctors, Radiologists, Nurses  
**Description:**  
Healthcare providers use the platform to access and review patient-submitted health data and medical images. They can verify AI-generated insights, annotate images, prescribe medications, and update patient records. The system enables asynchronous consultations, where providers can respond to patient messages or review cases at their convenience. The collaboration tools also allow secure sharing of cases among specialists, promoting multi-disciplinary decision-making and improving diagnostic accuracy.

### ****7.10 IT Team****

**Actors:** IT Staff  
**Description:**  
The IT team is responsible for maintaining the system’s infrastructure, including backend services, databases, and network configurations. Their responsibilities include deploying software updates, managing user access controls, monitoring server uptime, and implementing cybersecurity protocols. The IT team works closely with developers and hospital administrators to ensure that the system meets performance standards and that patient data remains confidential and secure.

### ****7.11 Sales and Marketing Team****

**Actors:** Marketing and Sales Representatives  
**Description:**  
This team promotes the application across healthcare institutions, conferences, and digital platforms. They gather market insights, track user acquisition metrics, and work on branding and communication strategies. Their feedback is crucial in refining the system’s user experience and addressing unmet market needs. They also handle demos, onboarding campaigns, and engage with stakeholders to grow the system’s adoption.

### ****7.12 Customer Support****

**Actors:** Support Agents  
**Description:**  
Customer support agents assist users facing technical issues or usability challenges. They manage tickets, guide users through processes like image uploading or appointment booking, and troubleshoot system errors. Support agents also document recurring problems and collaborate with the development and IT teams to implement long-term solutions. Their role is key in maintaining high user satisfaction and engagement.

### ****7.13 Individuals (Non-Patients)****

**Actors:** General Public  
**Description:**  
Individuals not currently receiving treatment can still use the platform for informational purposes. They may inquire about health topics, check symptoms, or upload scans for general assessments without a formal diagnosis. This promotes early awareness and preventive healthcare, particularly for users considering medical consultation but seeking initial guidance.

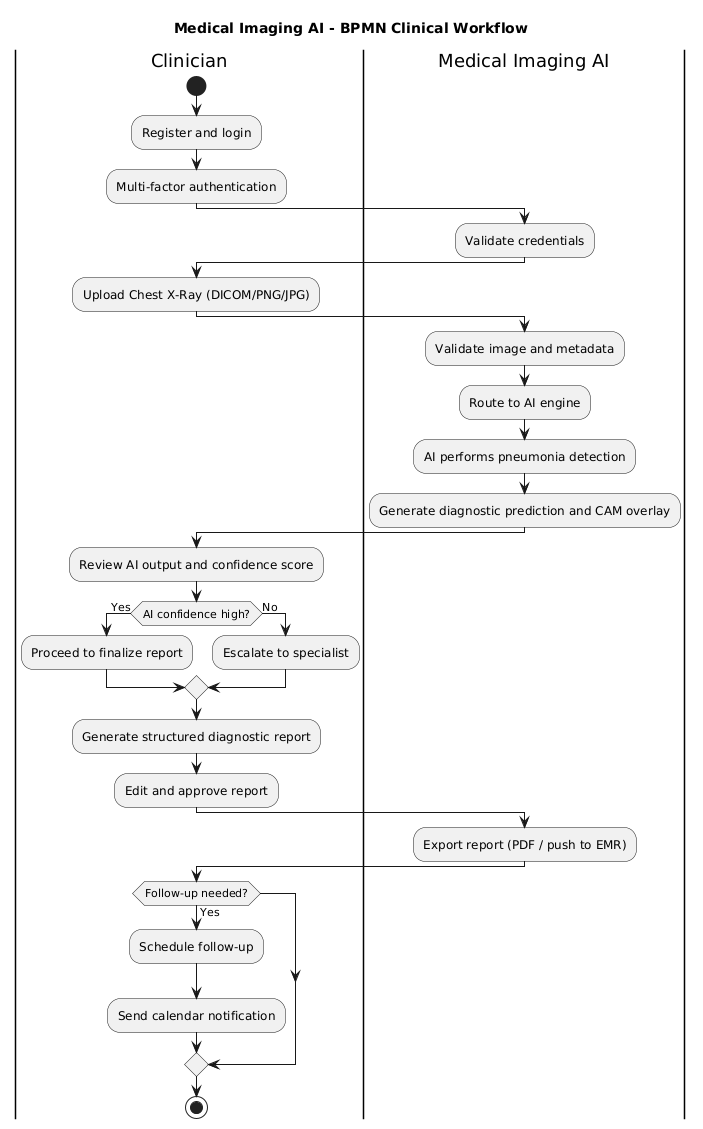
### ****7.14 Regulatory Authorities****

**Actors:** Compliance and Regulatory Bodies  
**Description:**  
Regulatory bodies ensure that the system complies with national and international standards for medical software. Their oversight includes reviewing data handling practices, evaluating algorithm fairness, ensuring transparency in AI decision-making, and auditing logs to confirm ethical use. These authorities may periodically assess system components or require formal certifications for continued operation.

### ****7.15 Developers****

**Actors:** Software Developers  
**Description:**  
Developers design, implement, and refine the AI algorithms and platform architecture. They work closely with medical professionals to integrate diagnostic models and ensure clinical accuracy. Developers also implement user interface features, bug fixes, and system optimizations. Their ongoing work is crucial for system scalability, reliability, and the adoption of cutting-edge AI techniques in medical imaging.

**8. BPMN Diagram**

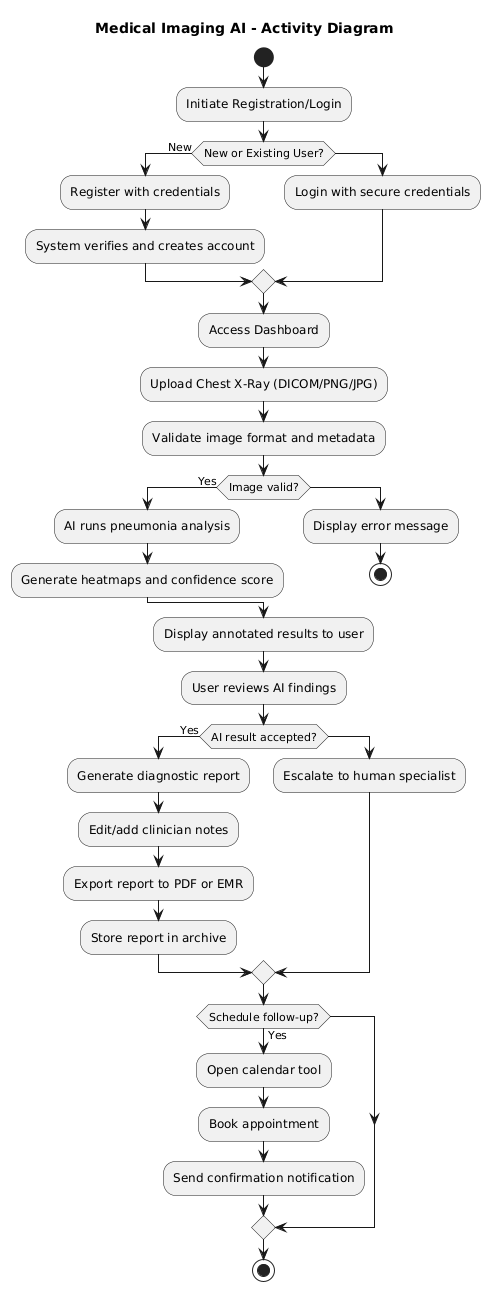
****

The BPMN diagram outlines the high-level workflow of a clinician interacting with the Medical Imaging AI system from registration to diagnosis and report export. It represents the sequential and decision-based flow in a clinical setting.

1. **User Registration and Login**: The clinician initiates the process by registering and securely logging into the platform. Multi-factor authentication ensures that only authorized medical personnel can proceed.
2. **Image Upload**: Once authenticated, users access the dashboard and upload a chest X-ray using the intuitive interface. Supported formats include DICOM, PNG, and JPG. The system validates file types and metadata.
3. **AI Analysis**: The uploaded image is routed to the AI engine, which performs pneumonia detection using a trained convolutional neural network. The system generates diagnostic predictions along with annotated overlays such as CAMs (Class Activation Maps).
4. **Diagnostic Review**: The clinician reviews the AI’s output, which includes visual explanations and confidence scores. If the AI’s confidence is high, the user may proceed to finalize the report. If confidence is low or the case is complex, the system allows escalation to a human specialist.
5. **Report Generation**: A structured diagnostic report is generated that includes findings, annotations, and physician commentary. Users can edit, approve, and export the report to PDF or push it to the EMR system.
6. **Follow-Up Scheduling**: If necessary, the clinician schedules a follow-up or second opinion. Integration with calendar tools and notification systems ensures timely patient care.

This diagram demonstrates how Medical Imaging AI fits seamlessly into a clinical workflow without replacing the physician, instead acting as a highly skilled assistant.

**9. Activity diagram**



This activity diagram represents the dynamic workflow of how users interact with the Medical Imaging AI platform. It outlines a comprehensive diagnostic journey, from user registration to AI-based image analysis and report generation. The system integrates artificial intelligence, medical imaging validation, and human-in-the-loop decision-making to ensure diagnostic accuracy, usability, and compliance.

**8.1 Start Point**

* **Symbol**: Filled black circle.
* **Meaning**: The user initiates interaction with the system, either to register or log in.

**8.2 User Registration/Login**

* **Actors**: New or Returning Users.
* **Actions**:
  + New users proceed to registration.
  + Existing users proceed to login.

**8.3 Register with Credentials (New Users)**

* **Input**: Full name, email, password, medical credentials (if applicable).
* **Security**: May include multi-factor authentication.
* **Outcome**: Account is created and stored securely.

**8.4 Login with Secure Credentials (Existing Users)**

* **Input**: Email and password.
* **Security**: Enforced via encryption and secure authentication protocols.

**8.5 System Verifies and Creates Account**

* **Action**: Backend system validates credentials and either creates a new account or grants access to an existing one.

**8.6 Access Dashboard**

* **User Interface**: Personalized dashboard becomes available.
* **Features**: Upload images, view history, access reports.

**8.7 Upload Chest X-ray (DICOM/JPG/PNG)**

* **Action**: User uploads a medical image.
* **Formats Supported**: DICOM, JPG, PNG.
* **Metadata**: Includes patient ID, scan date, modality.

**8.8 Validate Image Format and Metadata**

* **System Task**: Checks image format, resolution, and metadata completeness.
* **Outcome**: Determines whether the image is suitable for AI analysis.

**8.9 Valid Image?**

* **Decision Point**:
  + If invalid: Display error message and prompt re-upload.
  + If valid: Proceed to AI analysis.

**8.10 AI Runs Pneumonia Analysis**

* **Component**: AI Inference Engine (e.g., FastAI ResNet model).
* **Task**: Detects signs of pneumonia in the uploaded image.

**8.11 Generate Heatmaps and Confidence Score**

* **Output**:
  + Heatmaps (e.g., Grad-CAM) highlighting affected areas.
  + Confidence score (e.g., 95.6%) indicating AI certainty.

**8.12 Display Annotated Results or Error**

* **If successful**: Annotated image and AI findings are shown to the user.
* **If failed**: Error message is displayed with troubleshooting options.

**8.13 User Reviews AI Findings**

* **Action**: Radiologist or doctor reviews the AI-generated results.
* **Options**:
  + Accept the findings.
  + Escalate to a human specialist for further review.

**8.14 AI Results Accepted?**

* **Decision Point**:
  + If accepted: Proceed to report generation.
  + If not accepted: Escalate to human specialist.

**8.15 Generate Diagnostic Report**

* **Component**: Report Generator.
* **Content**: Includes diagnosis, confidence score, annotated image, and physician notes.

**8.16 Edit/Annotate Reports**

* **User Action**: Clinician can edit the report, add comments, or highlight specific findings.

**8.17 Escalate to Human Specialist (if needed)**

* **Action**: Case is forwarded to a senior radiologist or external consultant for second opinion.

**8.18 Send Confirmation Email**

* **System Task**: Sends a confirmation or notification email to the user or patient with the report or next steps.

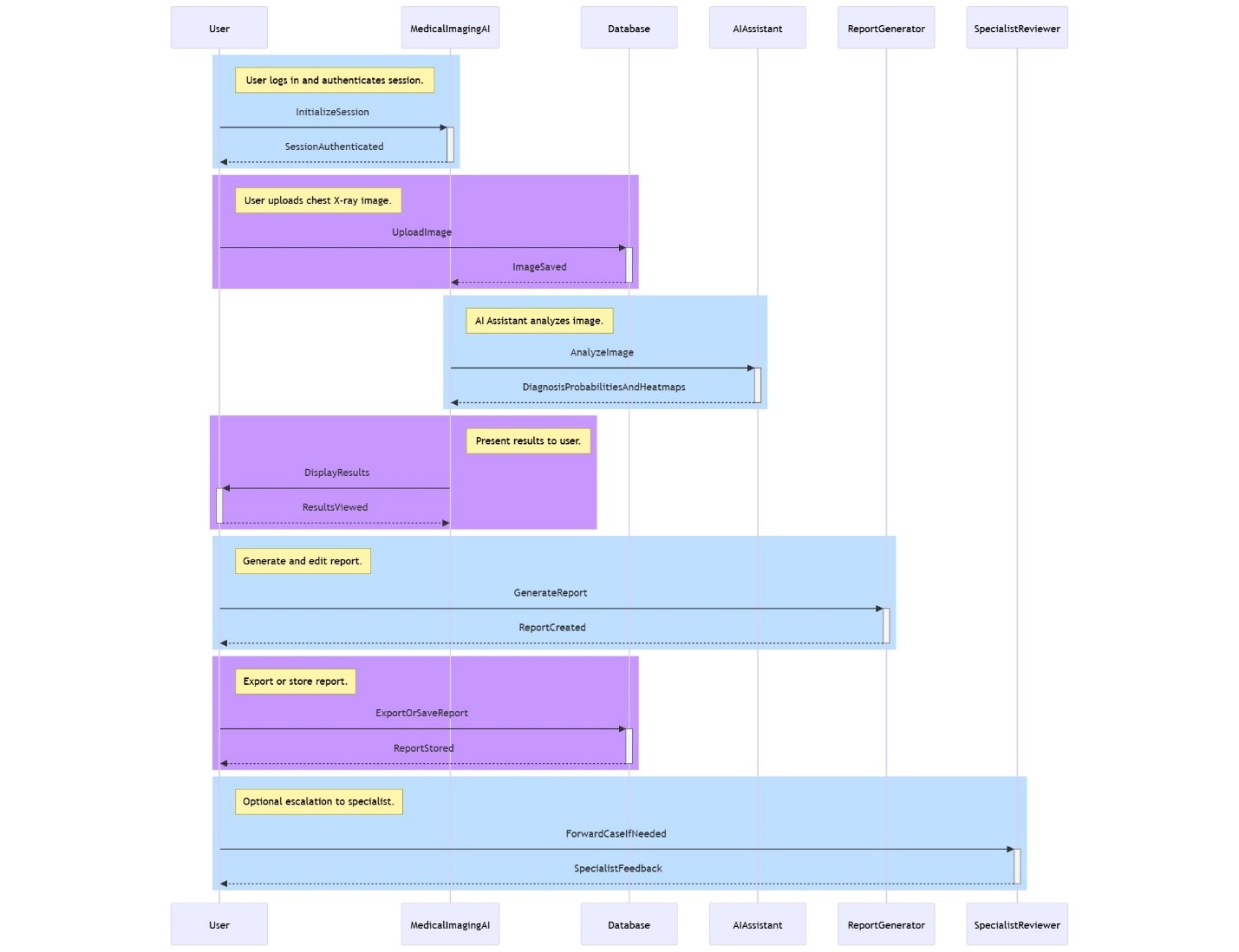
**8.19 End Point**

* **Symbol**: Black circle with red outline.
* **Meaning**: The workflow concludes with either a finalized report or an escalated case under review.

**System Highlights**

* **AI-Human Collaboration**: AI supports but does not replace clinical judgment.
* **Security & Compliance**: Encrypted data handling, HIPAA/GDPR aligned.
* **Scalability**: Designed for hospitals, clinics, and mobile diagnostic units.
* **User-Centric**: Intuitive interface for both technical and non-technical users.

**10. Sequence diagram**

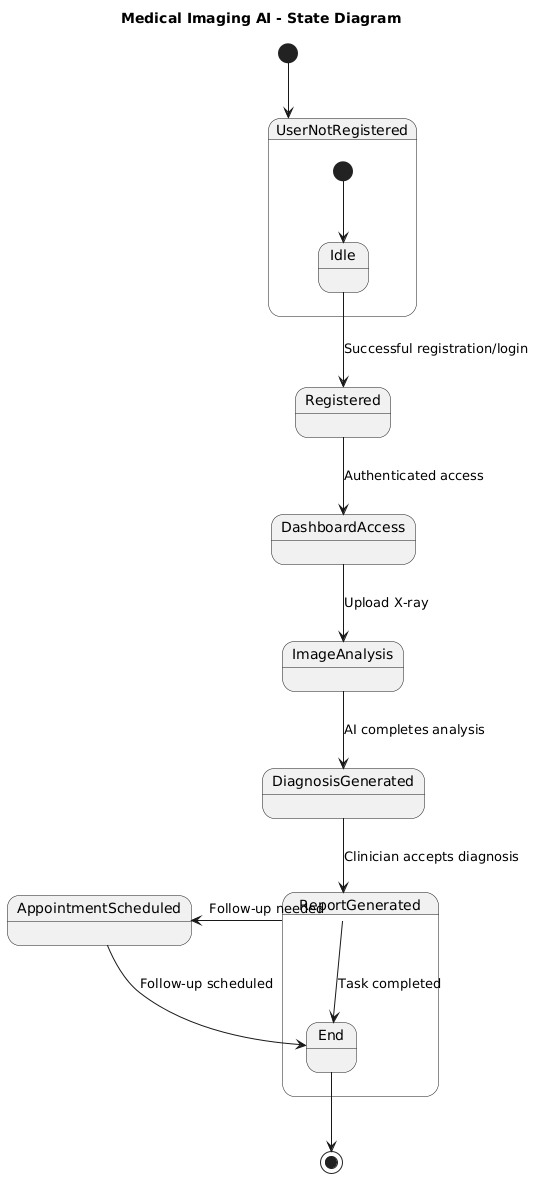
  
The following sequence diagram illustrates the step-by-step interaction between different system components during a typical pneumonia diagnosis process using the Medical Imaging AI system.

**Participants:**

* **User**: Typically a radiologist or clinician.
* **MedicalImagingAI**: Main application logic.
* **Database**: Stores images and reports.
* **AI Assistant**: AI model performing image analysis.
* **Report Generator**: Creates and formats diagnostic reports.
* **Specialist Reviewer**: A medical expert for optional case escalation.

**Workflow Steps:**

1. **Session Initialization**
   * The user logs in and the system authenticates the session to ensure secure access.
2. **Image Upload**
   * The user uploads a chest X-ray image.
   * The image is saved in the database for processing.
3. **AI Analysis**
   * The AI Assistant analyzes the image.
   * It returns diagnostic probabilities and visual heatmaps indicating areas of concern.
4. **Results Presentation**
   * The MedicalImagingAI system displays the AI-generated results to the user for review.
5. **Report Generation**
   * A report is generated based on the AI’s findings.
   * The user can view, edit, or add manual notes to the report.
6. **Export or Storage**
   * The report is either exported to the Electronic Medical Record (EMR) system or saved for future reference.
7. **Specialist Review (Optional)**
   * If further analysis is required, the case is forwarded to a human specialist.
   * The specialist provides feedback for final diagnosis.

**10. State Diagram  
**

The state diagram models the different states of the Medical Imaging AI system as it transitions through various user actions and system processes. It provides a conceptual view of how the application behaves at runtime.

**UserNotRegistered**: This is the default entry point. The system is in an idle state waiting for new user registration.

**Registered**: After successful authentication, the user transitions to the “Registered” state, gaining access to all functionalities.

**Dashboard Access**: The user can now navigate to the dashboard to manage patient cases, view prior diagnoses, or initiate a new diagnostic process.

**ImageAnalysis**: Once an image is uploaded, the system transitions into a state of active AI processing. The user waits for the analysis to complete.

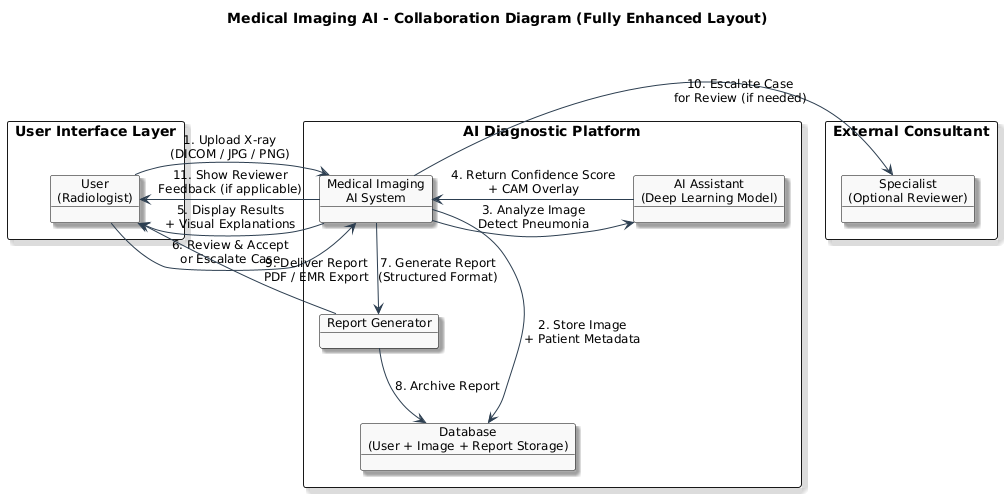
**DiagnosisGenerated**: The system displays the AI results, including diagnostic probability and annotated overlays.

**ReportGenerated**: When the clinician accepts the diagnosis, a formal report is generated. The user can edit or approve the report.

**AppointmentScheduled**: If follow-up is needed, the user enters the scheduling state and selects available dates.

**End**: Upon logout or task completion, the user returns to the idle state.

This state model demonstrates how users and data flow through various phases of interaction, reflecting a controlled and predictable system lifecycle.

**11. Collaboration Diagram**  


The collaboration diagram illustrates how system components interact to deliver core services, focusing on relationships between entities rather than temporal order. It highlights the real-time collaboration between user-facing components and backend AI services.

**Entities Involved:**

**User**: Initiates image upload, interacts with results, and triggers report generation.

**Medical Imaging AI System**: Coordinates input processing, AI inference, and report management.

**AI Assistant**: Performs pneumonia detection using deep learning models. It analyzes X-rays, generates confidence scores, and produces visual explanations.

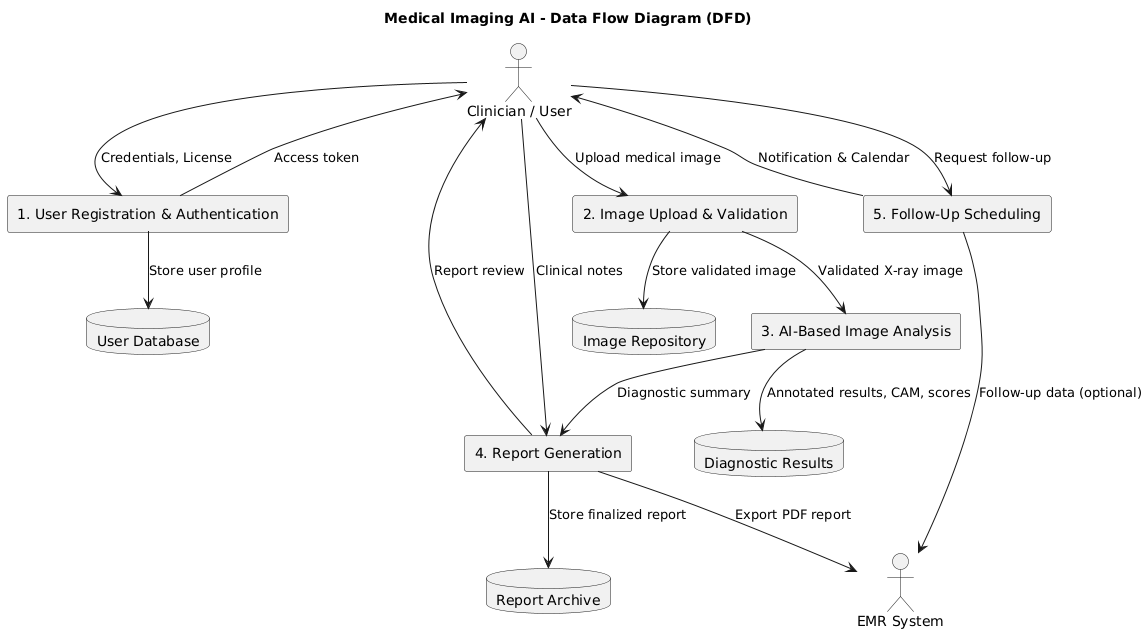
**Database**: Stores user profiles, medical images, reports, and interaction history securely.

**Report Generator**: Converts AI output and user annotations into a structured clinical report.

**Specialist (Optional)**: Accesses cases escalated by users for manual review or second opinion.

**Interaction Flow:**

1. The user sends an image to the Medical Imaging AI platform.
2. The image is stored in the database.
3. The AI Assistant processes the image and returns predictions with annotations.
4. The platform displays results and enables user review.
5. The Report Generator compiles a complete clinical document.
6. If needed, the case is shared with a specialist for additional input.

**12. DFD Diagram  
**

The Data Flow Diagram (DFD) for Medical Imaging AI provides a comprehensive view of how data moves through the system, from user input to AI-driven diagnostics and report output. It outlines the interaction between users, AI modules, data storage, and external systems like EMRs.

**System Context Overview**

The DFD illustrates the end-to-end workflow that enables pneumonia detection in chest X-rays. It includes data sources (users, medical images), processes (AI analysis, report generation), data stores (images, patient profiles), and outputs (diagnostic results and reports).

**External Entities**

* **Clinician/User**: Radiologists and physicians who input data and interact with the system.
* **EMR System**: External hospital systems that consume the AI-generated reports and patient data.

**Processes**

1. **User Registration & Authentication**
   * Input: User credentials and medical license.
   * Output: Secure access token and user profile creation.
2. **Image Upload & Validation**
   * Input: Medical image files (DICOM, JPG, PNG).
   * Output: Verified and preprocessed image stored in the system.
3. **AI-Based Image Analysis**
   * Input: Chest X-ray.
   * Processing: Deep learning model performs pneumonia detection using convolutional neural networks.
   * Output: Annotated image, confidence score, diagnostic summary.
4. **Report Generation**
   * Input: AI diagnostic output and physician notes.
   * Output: Editable report in structured format, exportable to PDF/EMR.
5. **Follow-Up Scheduling**
   * Input: Case status, physician preferences.
   * Output: Calendar event and notification to user and patient (optional integration).

**Data Stores**

* **User Database**: Contains profiles, credentials, and session data.
* **Image Repository**: Securely stores uploaded images with patient references.
* **Diagnostic Results**: Stores AI predictions, annotations, and visualizations.
* **Report Archive**: Keeps finalized reports for legal, clinical, and audit purposes.

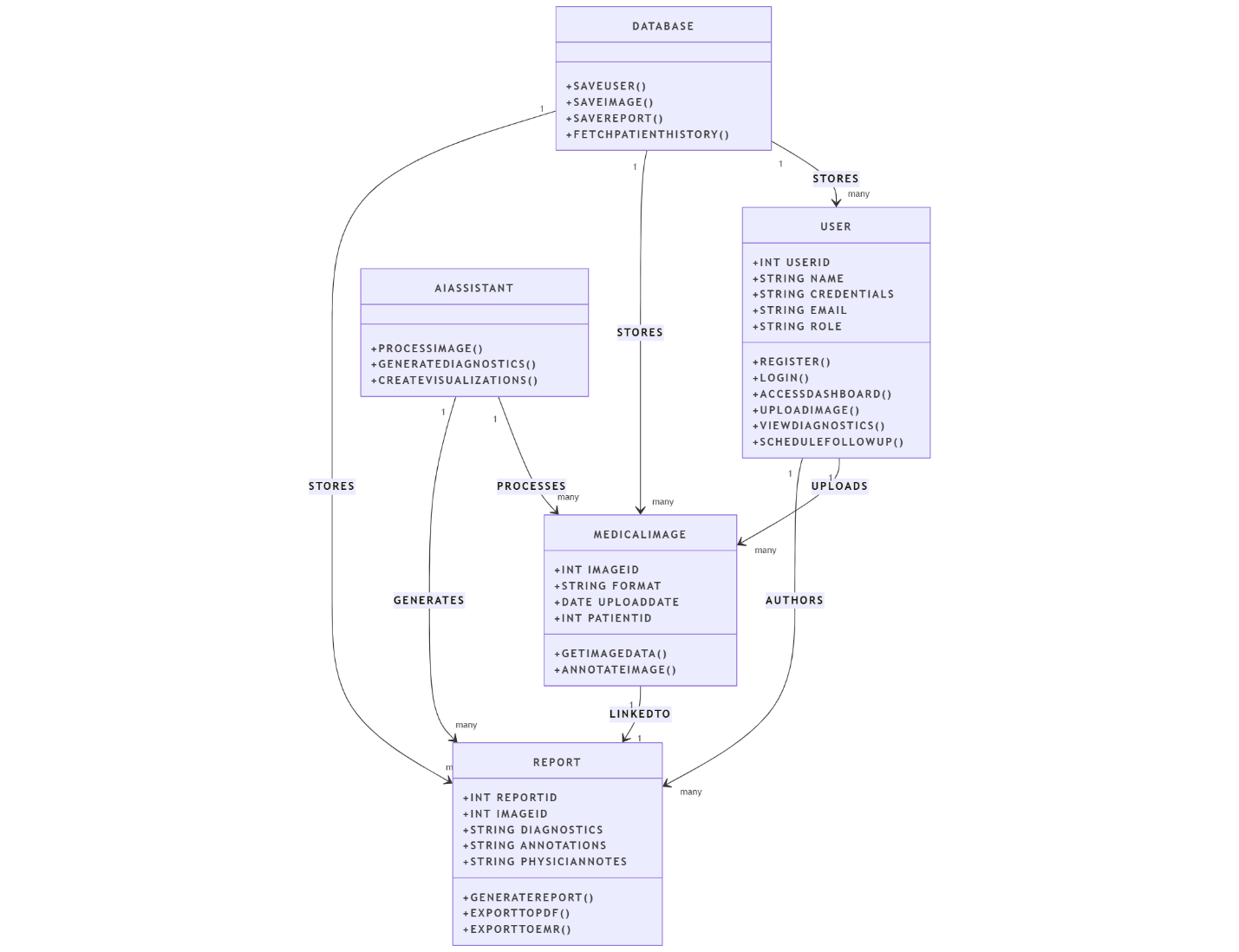
**Data Flows**

* **From User to System**: Image uploads, input notes, scheduling requests.
* **From AI Module to Report Generator**: Diagnostic outputs, CAM heatmaps.
* **From System to EMR**: Finalized clinical reports and follow-up records.
* **From Report Generator to Archive**: Stored reports for future retrieval and analysis.

**How It Works**

1. The clinician logs into the system and uploads a chest X-ray.
2. The image is validated, stored, and passed to the AI module.
3. The AI analyzes the image, detects pneumonia patterns, and returns annotated results.
4. A report is generated using this output and clinician input.
5. The report is exported to the EMR or saved for auditing.
6. Optionally, a follow-up review is scheduled with calendar integration.

**13. Class Diagram**



**1. User**

Represents the person interacting with the system (e.g., radiologist, doctor).

**Attributes**:

* userID, name, credentials, email, role

**Methods**:

* register(): Create a new user account.
* login(): Authenticate the user.
* accessDashboard(): Open the main interface.
* uploadImage(): Upload medical images (e.g., chest X-rays).
* viewDiagnostics(): View analysis results.
* scheduleFollowUp(): Schedule a follow-up session if needed.

**Relations**:

* **Uploads** many MedicalImages
* **Authors** many Reports
* Is **stored** in the Database

**2. MedicalImage**

Stores data about uploaded medical images.

**Attributes**:

* imageID, format, uploadDate, patientID

**Methods**:

* getImageData(): Retrieve image content.
* annotateImage(): Add notes/diagnostic marks.

**Relations**:

* **Uploaded by** a User
* **Processed by** the AIAssistant
* **Stored in** the Database
* **Linked to** one or more Reports

**3. AIAssistant**

The AI engine that analyzes medical images.

**Methods**:

* processImage(): Run image through AI model.
* generateDiagnostics(): Produce diagnostic results (e.g., probability of pneumonia).
* createVisualizations(): Generate heatmaps or overlays.

**Relations**:

* **Processes** MedicalImages
* **Generates** Reports

**4. Report**

Stores the output of the AI analysis (and potentially edited by a doctor).

**Attributes**:

* reportID, imageID, diagnostics, annotations, physicianNotes

**Methods**:

* generateReport(): Create a new report from image diagnostics.
* exportToPDF(): Export report to PDF format.
* exportToEMR(): Send report to electronic medical records system.

**Relations**:

* **Linked to** one MedicalImage
* **Authored by** a User
* **Stored in** the Database

**5.Database**

Handles system-wide data storage.

**Methods**:

* saveUser(): Store new user details.
* saveImage(): Store uploaded images.
* saveReport(): Store generated reports.
* fetchPatientHistory(): Retrieve a patient's diagnostic history.

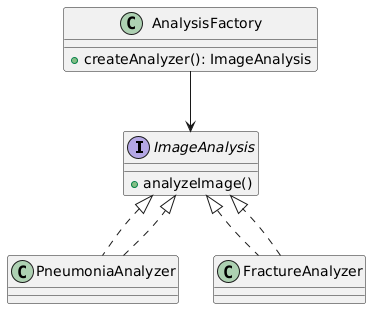
**Relations**:

* **Stores** many User, MedicalImage, and Report entries

**Summary of System Flow:**

1. **User registers/logs in**.
2. **Uploads a medical image**, which is saved in the **Database**.
3. The **AI Assistant** **processes the image** and generates diagnostics.
4. The output is turned into a **Report**, **linked** to the image and user.
5. The **Report** is stored in the **Database** and can be exported or reviewed.

**14. Factory Method Design Pattern**



**1. AnalysisFactory Class**

* **Purpose**: This is a **factory class** used to create instances of image analysis tools.
* **Method**:
  + createAnalyzer(): ImageAnalysis  
    This method returns an object of type ImageAnalysis. It decides which specific analyzer (like Pneumonia or Fracture) to create.

**2. ImageAnalysis Class**

* **Purpose**: This is an **abstract base class** or **interface** that defines a common method for all image analyzers.
* **Method**:
  + analyzeImage()  
    This is the method that all subclasses must implement to perform their specific type of image analysis.

**3. PneumoniaAnalyzer Class**

* **Inherits from**: ImageAnalysis
* **Purpose**: This class specializes in analyzing images for signs of **pneumonia**.
* **Behavior**: It implements the analyzeImage() method with logic specific to detecting pneumonia in medical images.

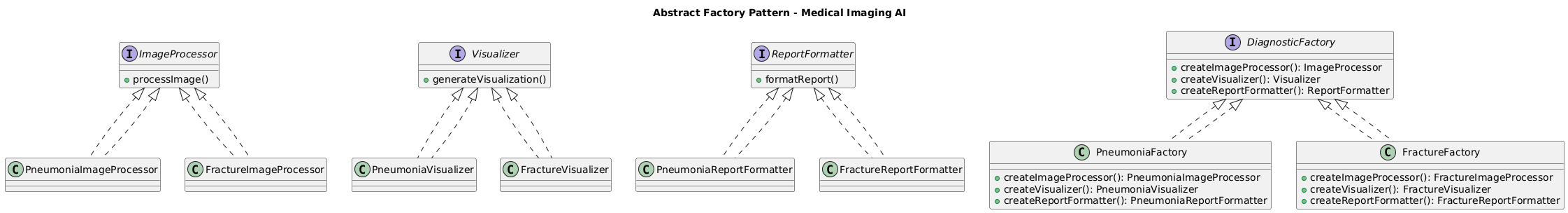
**4. FractureAnalyzer Class**

* **Inherits from**: ImageAnalysis
* **Purpose**: This class is designed to detect **bone fractures** in medical images.
* **Behavior**: It also implements the analyzeImage() method, but with logic tailored to identifying fractures.

**How It All Works Together**

1. A client (like a doctor using the system) requests an analysis.
2. The AnalysisFactory decides which analyzer to use (e.g., PneumoniaAnalyzer or FractureAnalyzer).
3. The factory returns an instance of the appropriate subclass.
4. The client calls analyzeImage() on that instance to perform the analysis.

**15. Abstract Factory Pattern Overview**

****

**1. Abstract Factory Pattern Overview**

The **Abstract Factory Pattern** is used to create families of related or dependent objects without specifying their concrete classes. It’s especially useful when your system needs to be independent of how its objects are created.

**2. Components in the Diagram**

**A. ProductProcessor**

These are abstract processors for different product types:

* **ProductAProcessor**
* **ProductBProcessor**

They define how each product type should be processed.

**B. Productor**

These are the actual product types:

* **ProductA**
* **ProductB**

They represent the core items being created and processed.

**C. RegionalProcessor**

These are concrete implementations of ProductProcessor for different regions:

* **RegionalAProcessor**
  + Handles ProductA and ProductB with methods like understandingProductA, ProcessedProductA, etc.
* **RegionalBProcessor**
  + Similar to RegionalA but tailored for another region.

Each regional processor knows how to handle products in a region-specific way.

**D. RegionalFactory**

These are the abstract factories that produce region-specific processors:

* **RegionalFactory1**
  + Can produce:
    - understandingRegionalA, ProcessedRegionalA
    - understandingRegionalB, ProcessedRegionalB

This factory encapsulates the logic for creating region-specific product processors.

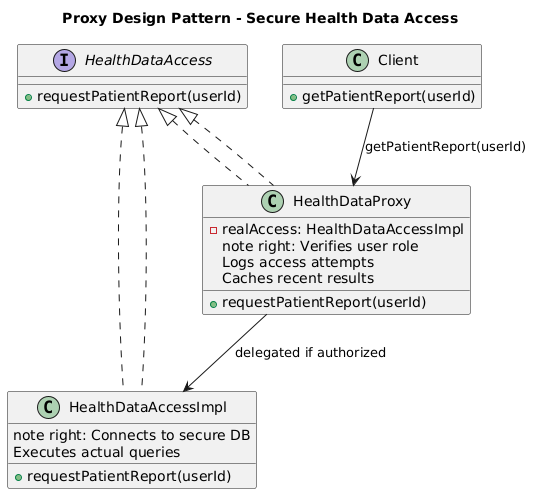
**3. How It All Works Together**

1. A client requests a product processor from a RegionalFactory.
2. The factory returns a RegionalProcessor (e.g., RegionalAProcessor).
3. The processor then handles specific products (e.g., ProductA, ProductB) using region-specific logic.

**Use Case Example**

Imagine a medical imaging AI system that needs to process scans differently in Europe and Asia due to regulatory or clinical differences:

* RegionalAProcessor might follow EU standards.
* RegionalBProcessor might follow Asian standards.
* The RegionalFactory ensures the correct processor is used based on the region.

**16. Proxy Design Pattern **

**1. HealthDataAccess (Interface)**

* This is the **common interface** that both the proxy and the real data access class implement.
* It defines a method:
  + requestPatientReport(userId)
* This ensures that the client can interact with either the proxy or the real implementation in the same way.

**2. Client**

* The **Client** is the user or system component that needs access to patient reports.
* It calls:
  + getPatientReport(userId) on the **HealthDataProxy**.
* The client is unaware of whether it’s talking to the proxy or the real data source—this is the essence of the proxy pattern.

**3. HealthDataProxy (Proxy Class)**

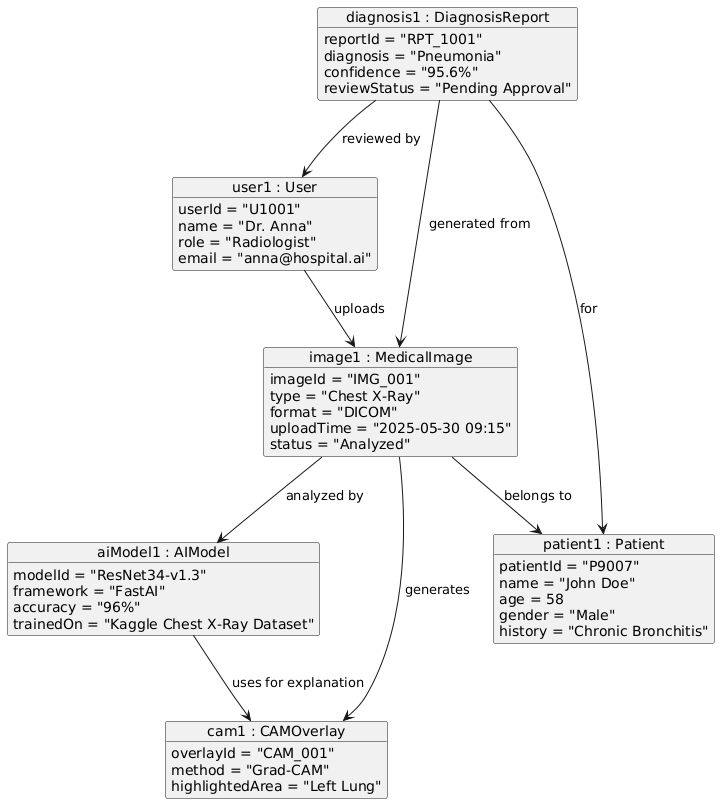
* This is the **proxy** that controls access to the real data source.
* It performs several important functions:
  + **Verifies user role**: Ensures the user has permission to access the requested data.
  + **Logs access attempts**: Keeps a record of who accessed what and when (important for auditing and compliance).
  + **Caches recent results**: Improves performance by avoiding repeated queries for the same data.
* If the user is authorized, it delegates the request to:
  + HealthDataAccessImpl

**4. HealthDataAccessImpl (Real Subject)**

* This is the **actual implementation** that connects to the secure database.
* It executes the real query to fetch the patient report.
* It implements:
  + requestPatientReport(userId)
* This class is only accessed through the proxy, ensuring that all access is controlled and logged.

**How It All Works Together**

1. The **Client** requests a patient report.
2. The **HealthDataProxy** intercepts the request.
3. The proxy:
   * Checks if the user is authorized.
   * Logs the request.
   * Checks the cache.
4. If everything is valid, it forwards the request to **HealthDataAccessImpl**.
5. The real class fetches the data from the secure database and returns it.

**17. Object Diagram  
**

This diagram represents the **data flow and relationships** between different components in a medical imaging AI system.

**1. User**

* **Who**: Dr. Anna (Radiologist)
* **Attributes**:
  + userid = "U1001"
  + name = "Dr. Anna"
  + role = "Radiologist"
  + email = "anna@hospital.ai"
* **Role**: Uploads medical images and reviews AI-generated diagnosis reports.

**2. Patient**

* **Who**: John Doe
* **Attributes**:
  + patientid = "P\_0001"
  + name = "John Doe"
  + age = 54
  + gender = Male
  + history = Chronic Bronchitis
* **Role**: The subject of the medical imaging and diagnosis.

**3. MedicalImage**

* **What**: A chest X-ray image uploaded for diagnosis.
* **Attributes**:
  + imageid = "IMG\_001"
  + type = "Chest X-Ray"
  + format = "DICOM"
  + uploadTime = "2025-05-30 @9:15"
* **Role**: The input image analyzed by the AI model.

**4. AIModel**

* **What**: The deep learning model used for diagnosis.
* **Attributes**:
  + modelid = "ResNet34-V4.3"
  + version = "4.3"
  + accuracy = "96%"
  + trainedOn = "Kaggle Chest X-Ray Dataset"
* **Role**: Processes the medical image and generates a diagnosis.

**5. CAMOverlay**

* **What**: A visual explanation of the AI’s decision.
* **Attributes**:
  + overlayId = "CAM\_001"
  + method = "Grad-CAM"
  + highlightedArea = "Left Lung"
* **Role**: Highlights the region of the image that influenced the AI’s diagnosis (explainability).

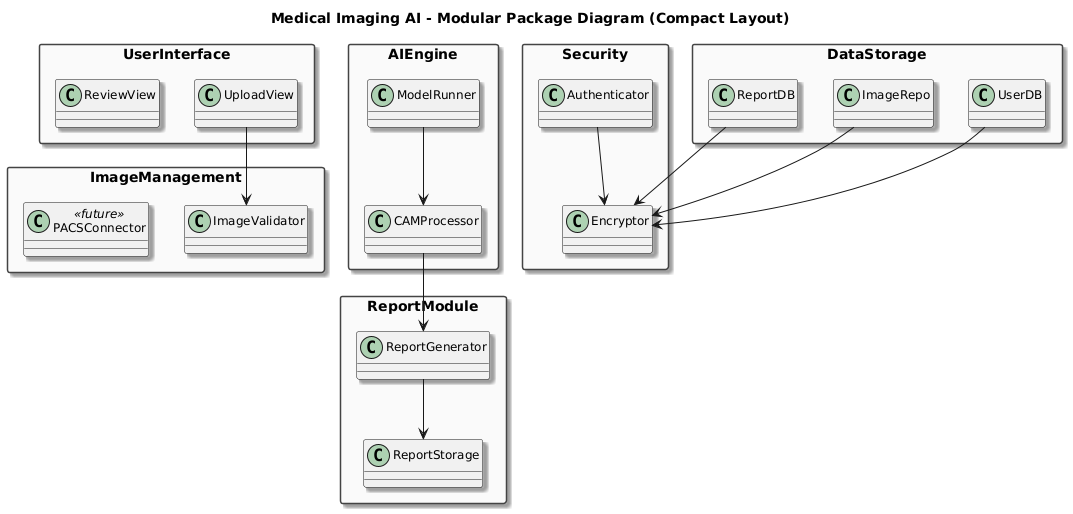
**6. DiagnosisReport**

* **What**: The final output of the AI analysis.
* **Attributes**:
  + reportid = "RPT\_1001"
  + diagnosis = "Pneumonia"
  + confidence = "95.6%"
  + reviewStatus = "Pending Approval"
* **Role**: Summarizes the AI’s findings and is reviewed by the radiologist.

**Workflow Summary**

1. **Dr. Anna** uploads a **Chest X-Ray** for **John Doe**.
2. The **AIModel** (ResNet34) analyzes the image.
3. It generates a **DiagnosisReport** indicating pneumonia with 95.6% confidence.
4. A **CAMOverlay** highlights the left lung to explain the AI’s decision.
5. The report is marked as **Pending Approval** for Dr. Anna to review.

**18. Package Diagram**



This image shows a **Package Diagram** for a **Medical Imaging AI system**, which organizes the system into logical modules.

**1. UserInterface Module**

* **Components**:
  + UploadView: Interface for uploading medical images.
  + ReviewView: Interface for reviewing AI-generated results and reports.
* **Purpose**: This is what doctors and technicians interact with directly.

**2. ImageManagement Module**

* **Components**:
  + PACSConnector: Connects to hospital imaging systems (PACS).
  + ImageValidator: Checks image format, quality, and metadata.
* **Purpose**: Ensures that uploaded images are valid and compatible with the system.

**3. AIEgine Module**

* **Components**:
  + ModelRunner: Runs the AI model (e.g., ResNet) on the uploaded image.
  + CAMProcessor: Generates Class Activation Maps (CAMs) for visual explainability.
* **Purpose**: Performs the core AI analysis and produces interpretable results.

**4. ReportModule**

* **Components**:
  + ReportGenerator: Creates diagnostic reports based on AI output and user input.
  + ReportStorage: Saves reports for future access or EMR integration.
* **Purpose**: Handles the creation and storage of diagnostic reports.

**5. Security Module**

* **Components**:
  + Authenticator: Manages user login and role-based access.
  + Encryptor: Encrypts sensitive data (e.g., patient info, reports).
* **Purpose**: Ensures data privacy and system security (HIPAA/GDPR compliance).

**6. DataStorage Module**

* **Components**:
  + UserDB: Stores user profiles and credentials.
  + ImageRepo: Stores uploaded medical images.
  + ReportDB: Stores diagnostic reports.
* **Purpose**: Centralized storage for all system data.

**How It All Works Together**

1. A user uploads an image via UploadView.
2. ImageValidator checks the image.
3. ModelRunner analyzes it and CAMProcessor generates visual overlays.
4. ReportGenerator creates a report, which is stored in ReportStorage.
5. Authenticator ensures only authorized users access the system.
6. All data is securely stored in UserDB, ImageRepo, and ReportDB.

**19. Components**

**1. User Interface (UI)**

* **Description**: The UI provides a clean, intuitive interface for doctors, radiologists, and hospital staff to interact with the system efficiently.
* **Responsibilities**:
  + Allow image upload (X-ray, MRI, CT) in JPEG, PNG, or DICOM formats.
  + Display AI-detected anomalies with visual annotations (bounding boxes, heatmaps).
  + Enable navigation across patient history, report generation, and role-specific dashboards.

**2. AI Analysis Engine**

* **Description**: The AI engine is the backbone of the diagnostic process, analyzing medical images using deep learning models.
* **Responsibilities**:
  + Run inference on uploaded scans to detect abnormalities (e.g., tumors, fractures).
  + Generate confidence scores and annotated images.
  + Learn from user feedback to improve accuracy over time.

**3. User Management & Authentication**

* **Description**: This module handles secure registration, login, and role-based access for doctors, radiologists, and administrators.
* **Responsibilities**:
  + Authenticate users using hospital credentials.
  + Assign and manage user roles.
  + Maintain compliance with medical data access policies.

**4. Patient Health Record Module**

* **Description**: Maintains and organizes each patient’s imaging history, diagnostic reports, and comparison data.
* **Responsibilities**:
  + Store and retrieve past scans and reports securely.
  + Facilitate longitudinal health tracking and comparison of results.
  + Support filtering by patient, date, or diagnosis.

**5. Diagnostic Reporting Module**

* **Description**: Enables the generation, review, and export of AI-assisted diagnostic reports.
* **Responsibilities**:
  + Provide editable, downloadable PDF reports.
  + Allow addition of manual doctor notes.
  + Support secure report sharing via email or internal systems.

**6. Feedback Loop**

* **Description**: Enhances the AI’s diagnostic precision by incorporating real-time feedback from medical professionals.
* **Responsibilities**:
  + Allow marking AI predictions as correct or incorrect.
  + Feed labeled outcomes back into the learning model.
  + Track model performance improvements over time.

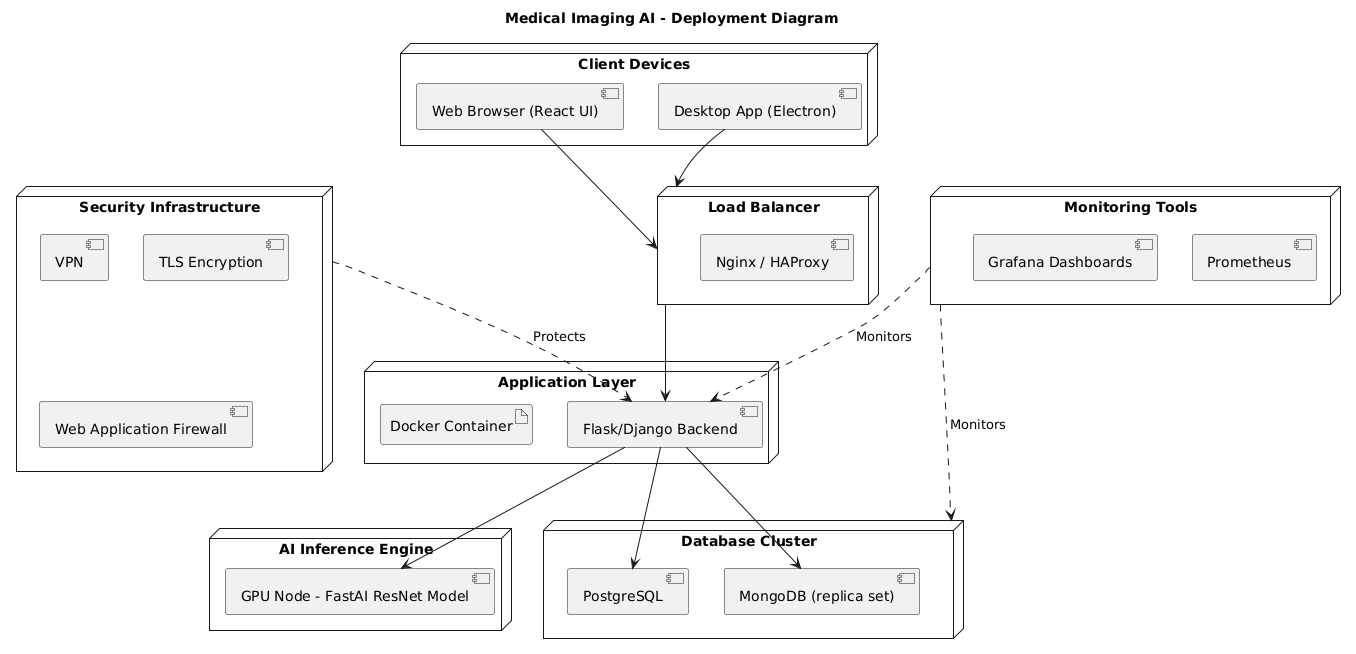
**7. Hospital Communication Interface**

* **Description**: Facilitates consultation and collaboration between hospital staff and external specialists.
* **Responsibilities**:
  + Enable escalated communication for complex cases.
  + Coordinate hospital visit recommendations.
  + Help streamline patient referral processes.

**8. Security & Compliance Module**

* **Description**: Ensures full adherence to medical data privacy laws such as HIPAA or GDPR.
* **Responsibilities**:
  + Encrypt sensitive data at rest and in transit.
  + Monitor access logs and perform routine audits.
  + Implement strict access control and session management.

**20. Deployment Diagram**

****

This is a **Deployment Diagram** for a **Medical Imaging AI system**, showing how the system is structured and deployed across different environments. Let’s go through it step by step:

**1. Client Devices**

* **Web Browser (React UI)**: Used by doctors or technicians to access the system via a web interface.
* **Desktop App (Electron)**: An alternative desktop-based interface for the same users.
* **Purpose**: These are the entry points where users upload images, view results, and interact with the system.

**2. Load Balancer**

* **Tools**: Nginx or HAProxy
* **Purpose**: Distributes incoming traffic from client devices across multiple backend servers to ensure high availability and performance.

**3. Monitoring Tools**

* **Grafana Dashboards**: Visualize system metrics and performance.
* **Prometheus**: Collects and stores time-series data for monitoring.
* **Purpose**: These tools monitor the health and performance of the entire system.

**4. Database Cluster**

* **PostgreSQL**: Stores structured data like user profiles, reports, and logs.
* **MongoDB (Replica Set)**: Stores unstructured or semi-structured data like image metadata or AI results.
* **Purpose**: Ensures data persistence, redundancy, and fast access.

**5. AI Inference Engine**

* **GPU Node**: A server with a GPU that runs the AI model.
* **Model**: FastAI ResNet (used for analyzing medical images).
* **Purpose**: Performs the actual image analysis and returns diagnostic results.

**6. Application Layer**

* **Docker Containers**: Encapsulate the backend services for easy deployment and scaling.
* **Flask/Django Backend**: Handles business logic, API requests, and communication between the UI and AI engine.
* **Purpose**: Core logic and orchestration of the system.

**7. Security Infrastructure**

* **VPN**: Secures remote access to the system.
* **TLS Encryption**: Encrypts data in transit.
* **Web Application Firewall (WAF)**: Protects against malicious traffic and attacks.
* **Purpose**: Ensures the system is secure and compliant with healthcare regulations (e.g., HIPAA, GDPR).

**How It All Works Together**

1. A user accesses the system via a **web browser or desktop app**.
2. The **load balancer** routes the request to the appropriate backend container.
3. The **backend** processes the request and interacts with:
   * The **AI engine** for image analysis.
   * The **database** for storing/retrieving data.
4. **Monitoring tools** track system health.
5. **Security infrastructure** ensures all communication and access are protected.