Capstone Project

SRS Document of AI-Powered X-Ray Detection with Explainable AI

Internal Advisor:

Mr. Farooq Javed

Project Manager:

Dr. Muhammad Saad Razzaq

Project Team:

Mohad Imran(TL) – BSCS51F22S095
Hamza Amin(TM) – BSCS51F22S052
Umer Aslam(TM) – BSCS51F22S084

Submission Date:

11 November, 2025

Project Manager’s Signature

Table of Contents

1. Introduction

1.1 Purpose of Document

The purpose of the **Software Requirements Specification (SRS)** is to clearly define all the functional and non-functional requirements of the **AI-Powered X-Ray Detection System with Explainability (XAI)**. It serves as a comprehensive guide describing what the system will do, the conditions under which it will operate, and how it will perform. This document ensures that every stakeholder—developers, testers, project supervisors, and end users—has a shared understanding of the system's goals, features, and limitations.

The SRS also acts as a foundation for system design, implementation, and validation. It provides measurable criteria for evaluating whether the final product meets the intended objectives, such as disease detection accuracy, explainability through visual outputs (like Grad-CAM), and ease of use for healthcare professionals.

1.2Project Overview

The AI-Powered X-Ray Detection with Explainability project aims to assist healthcare professionals in diagnosing chest diseases such as pneumonia and tuberculosis using deep learning models (CNN, ResNet, DenseNet) integrated with Explainable AI (XAI) techniques like Grad-CAM and LIME. The system not only predicts disease presence but also highlights the affected regions, improving transparency and medical trust.

Developed as a web-based platform, it enables doctors to upload X-ray images, view AI-generated diagnostic reports, and manage patient records efficiently. By combining automation with explainability, the project ensures accurate, interpretable, and quick medical assistance—especially for rural areas of Pakistan, where radiology resources are limited.

The following are the project's goals and benefits:

1) Develop an AI system to detect pneumonia and tuberculosis from X-rays.
2) Use Explainable AI (XAI) to show highlighted disease areas.
3) Create a web-based platform for X-ray upload and patient record management.
4) Train models on local Pakistani datasets for better accuracy.
5) Ensure transparency and trust in AI-based diagnosis.
6) Faster and more accurate disease detection.
7) Visual explanations for better medical trust.
8) Accessible diagnostic support in rural areas.
9) Reduced workload for healthcare professionals.
10) Scalable for future medical applications.

1.3Scope

The AI-Powered X-Ray Detection with Explainability system aims to assist medical professionals in detecting chest-related diseases using deep learning and interpretable AI. The project focuses on pneumonia and tuberculosis detection from chest X-ray images, integrating explainable visual outputs to build clinician trust and support diagnostic decision-making.

In Scope (System Will Do):

1) **Image Processing & Upload:**
   Enable users to upload anonymized chest X-ray images (JPEG, PNG, or DICOM) through a web interface for automated analysis.
2) **Disease Detection:**
   Utilize deep convolutional neural networks (CNNs) such as ResNet and DenseNet to identify pneumonia and tuberculosis, providing results with confidence scores.
3) **Explainability:**
   Implement visual explanation tools like Grad-CAM and LIME, highlighting key image regions influencing the model's predictions.
4) **Dashboard & Record Management:**
   Offer a simple web-based dashboard to view results, visualize heatmaps, and maintain patient record summaries.
5) **Model Training & Evaluation:**
   Train, validate, and test models using both local (Pakistani) and public datasets, ensuring accuracy and transparency through detailed performance reporting.
6) **Documentation & Reporting:**
   Provide technical documentation, a user guide, and ethical guidelines clarifying that the system supports rather than replaces clinical judgment.

Out of Scope (System Will Not Do):

1) The system will not detect non-chest diseases or interpret CT, MRI, or other imaging modalities.
2) No integration with hospital information systems (HIS) or PACS during this phase.
3) The project will not involve direct control of imaging hardware or real-time data acquisition.
4) Regulatory approval, clinical deployment, or live hospital integration are beyond current project scope.

2. Overall System Description

The AI-Powered X-Ray Detection System with Explainability (XAI) will be developed in a web-based environment, allowing access from computers and mobile devices connected to the internet. The backend will be built using Python frameworks (Flask/Django) integrated with deep learning models (CNN, ResNet, DenseNet) trained in Google Colab or similar environments. The frontend will be designed using React.js or Bootstrap, ensuring an interactive and user-friendly interface for healthcare professionals.

Radiologists, medical staff, and healthcare workers, especially in rural and under-resourced areas of Pakistan, where access to expert radiologists is limited, will primarily use the system. Users will be able to upload a chest X-ray image, view the automated disease prediction (such as pneumonia or tuberculosis), and see a visual explanation (using Grad-CAM or LIME) highlighting the affected lung region. A dashboard will also allow viewing, storing, and managing patient records.

2.1User Characteristics

1) Primary / Critical Users

i. Radiologists / Specialists

- Use the system for accurate, explainable second opinions on chest X-rays.
- Moderate technical skills; daily hospital use.
- Full access to uploads, overlays, reports.
- Need high accuracy, traceable explainability, and easy visualization tools.

ii. General Practitioners / Rural Clinicians

- Use for quick triage when specialists aren't available.
- Basic computer skills; frequent use in rural areas.
- Access to uploads, predictions, and referral suggestions.
- Need simple UI, clear summaries, and offline support.

iii. Technicians / Record Keepers

- Handle image uploads and patient data management.
- Moderate technical skills; daily users.
- Access for upload/delete, record updates.
- Need batch uploads, validation, and error prompts.

2) Secondary / Important Users

i. Hospital Administrators

- Monitor usage, compliance, and performance.
- Low–moderate skill; weekly or monthly use.
- Access to dashboards and reports.
- Need KPIs, user logs, and exportable data.

ii. Data Scientists / ML Engineers

- Handle model training, validation, and monitoring.
- High technical skills; periodic use.
- Access to datasets, metrics, and logs.

- Need reproducible pipelines and tracking tools.

iii. IT / DevOps Staff

- Maintain system uptime, security, and backups.
- High technical skills; ongoing role.
- Admin access for deployments and logs.
- Need RBAC, secure storage, and alerts.

3) Tertiary / Occasional Users

i. Medical Students / Trainees

- Learn from AI explanations and case studies.
- Variable skill; occasional use.
- Read-only access to anonymized cases.
- Need tutorials and interactive learning.

ii. Public Health Officials / Researchers

- Analyze aggregated health data.
- Moderate–high skill; periodic use.
- Access to de-identified statistics.
- Need export and filtering options.

iii. Patients / Caregivers

- View understandable medical reports.
- Low skill; occasional use.
- Access to personal reports with consent.
- Need plain explanations and privacy controls.

iv. NGOs / Outreach Workers

- Use during health camps and screenings.
- Low–moderate skill; campaign-based use.
- Limited upload/report access.
- Need mobile-friendly and offline support.

4) Cross-Cutting Considerations

- **Usability:** Simple, responsive UI for all roles.
- **Privacy:** Enforce RBAC, anonymization, and audit trails.
- **Explainability:** Layered insights for both experts and non-experts.
- **Training:** Quick onboarding guides and tooltips per user type.

2.2 Operating environment

The AI-Powered X-Ray Detection System will operate in a web-based environment, accessible through modern browsers and compatible with standard computing devices. The system will primarily run on cloud or local servers capable of handling deep learning computations and data storage.

1) **Hardware Platform:**
    - Server with GPU support (e.g., NVIDIA Tesla/RTX) for model training and inference
    - Client systems: Standard desktop/laptop or tablet with minimum 8GB RAM
    - Internet connectivity for web-based access
2) **Operating System:**
    - Server: Linux (Ubuntu 20.04 or later) or Windows Server
    - Client: Windows 10/11, macOS, or Linux
3) **Software Components:**
    - Backend: Python (Flask/Django), TensorFlow, PyTorch, OpenCV, Scikit-learn
    - Frontend: React.js, Bootstrap, HTML, CSS, JavaScript
    - Database: MySQL or Firebase for patient record storage
    - Visualization Tools: Matplotlib, Plotly, Seaborn for explainability visualization (Grad-CAM, LIME)
    - Development and Testing: Google Colab / VS Code for training and experimentation
4) **Integration & Coexistence:**
    - The system will coexist with standard web technologies and may integrate with cloud storage services (e.g., Firebase Storage or Google Drive).
    - It does not require integration with hospital hardware or hospital information systems at this stage.

2.3 System constraints

Following are some constraints related to our project:

1) **Software Constraints**
    - The system must be compatible with Python frameworks (TensorFlow, PyTorch, Keras) and web technologies (Flask/Django, React.js).
    - The application will require significant GPU resources for model training and testing, which limits deployment on low-performance systems.
    - The database (MySQL/Firebase) must ensure secure and efficient data storage for patient records and images.
    - The web dashboard must run smoothly on common browsers (Chrome, Firefox, Edge).
2) **Hardware Constraints**
    - The AI model training requires high-performance hardware with GPU (e.g., NVIDIA CUDA-enabled GPU).
    - For deployment, a system with at least 8 GB RAM and sufficient storage (minimum 50 GB) is required.
    - Mobile or low-end devices may face slower performance for real-time predictions.
3) **Cultural / Language Constraints**
    - The interface language will be primarily **English**, which may pose challenges for some healthcare workers in rural areas of Pakistan.
    - Future versions may include **Urdu language support** for accessibility.
4) **Legal Constraints**

- The system must comply with **data privacy and medical ethics** standards, ensuring that all X-ray data is anonymized before training or processing.
- No patient-identifiable information should be stored or shared without consent.
- Open-source datasets must be used under their respective licenses (e.g., NIH ChestX-ray14).

5) Environmental Constraints

- The web application will be used in hospitals or rural clinics, where **internet connectivity may be unstable**, affecting cloud-based model performance.
- The system design should not rely on audio cues since hospital environments may restrict noise.

6) User Constraints

- The users (doctors, technicians) may have **limited technical expertise**, so the interface should be simple, visual, and easy to use.
- Since the system targets rural clinics, it must require minimal setup and run efficiently on mid-level computers.
- The explainability visuals (heatmaps, Grad-CAM overlays) must be clear and interpretable even to non-technical users.

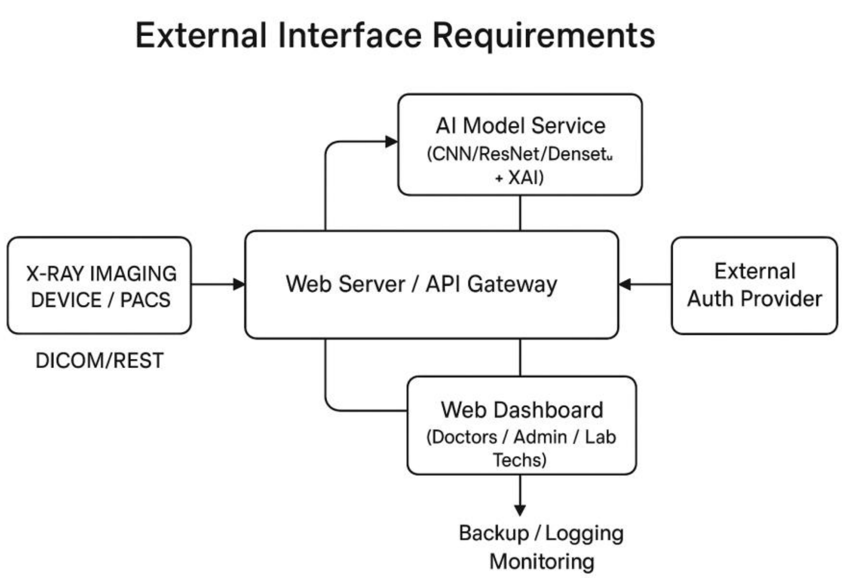
7) Off-the-Shelf Component Constraints

- Dependence on pre-trained models (ResNet, DenseNet) limits full customization.
- Cloud platforms like Google Colab have session time limits, which may interrupt long training processes.
- Third-party libraries and APIs must be compatible with the selected development framework versions.

3. External Interface Requirements

1) Purpose / scope

1) Specify the interfaces required so the system can accept X-ray images, call the AI engine, show results and XAI visualizations, store/retrieve patient records, and interact with external systems (imaging devices, cloud storage, optional HIS/PACS). This aligns with the project goals in the proposal (dashboard, model, XAI).
2) Context Diagram:



3) External entities (short descriptions)

1) X-Ray Imaging Device / PACS — supplies chest X-ray images (DICOM preferred).
2) Web Dashboard (Clinicians / Admin / Lab Techs) — browser-based UI used to upload images, view predictions, XAI heatmaps, and patient history.
3) AI Model Service — separate backend service that accepts images and returns diagnosis + XAI overlays.
4) Database — stores patient metadata, results, and audit logs (MySQL or Firebase as listed).
5) Authentication / Authorization Provider — local (JWT) or external (OAuth2 / hospital SSO).
6) Backup & Monitoring Services — for logs, model performance telemetry, and backups.
7) Optional: Hospital Information System (HIS) — if integrated in future (out of scope now but considered at interface level).

4) Interface Requirements (Functional & Technical)

i. Image Input

- **Formats:** DICOM (preferred), JPEG, PNG
- **Transfer:** DICOM (C-STORE) or HTTPS uploads
- **Limit:** Max 10 MB; larger files rejected with clear error
- **Metadata:** Patient ID, Study Date (required for non-DICOM)
- **Validation:** Server checks format & metadata before processing

ii. Web API

- **Protocol:** HTTPS (REST JSON, TLS 1.2+)
- **Auth:** JWT tokens
- **Key Endpoints:**
  - POST /images → upload image
  - GET /images/{id}/result → get diagnosis + XAI overlay
  - POST /predict → direct inference
- **Errors:** 400, 401, 415, 500

iii. AI Model Service

- **Input:** Image + optional metadata
- **Latency:** < 2 s on GPU (avg)
- **Retry:** Up to 2 times on failure

iv. Database Interface

- **Connection:** TLS-secured (DB/Firebase)
- **Tables:** Patient, Study/Image, Result, AuditLog

- **Backup:** Retention + CSV/JSON export support

v. Web Dashboard

- **Compatible Browsers:** Chrome, Firefox, Edge
- **Functions:** Upload, view results + XAI, search, export reports
- **Accessibility:** WCAG compliance
- **Communication:** REST API + WebSocket updates

vi. Authentication & Authorization

- **Roles:** Clinician, Radiologist, Admin, Technician
- **Methods:** Email + password (bcrypt), OAuth2/SAML SSO
- **Sessions:** JWT + refresh tokens
- **Audit:** All patient data access logged

vii. Logging & Monitoring

- **Logs:** Requests, timestamps, user IDs
- **Explainability:** Versioned and stored with results
- **Monitoring:** Health check and model status endpoints

5) Non-functional interface constraints

1) Security & Privacy: encrypt data in transit (TLS) and at rest (AES-256). Anonymize images where required. Follow local regulatory requirements for patient data (state compliance requirements in SRS).
2) Availability: target 99% uptime for core services during working hours; define off-hours maintenance windows.
3) Throughput / Performance: support N concurrent inference requests (document target; e.g., 10 concurrent GPU inferences). Provide queuing/backpressure behavior in SRS.
4) Scalability: AI service should be deployable as containerized microservice (Kubernetes) to scale horizontally.
5) Interoperability: DICOM compatibility for imaging devices; JSON REST for integrations.
6) Localization: date/time zones and language (English + Urdu optional) support in UI.

6) Data formats & schemas (summary)

1) DICOM — primary imaging format; keep header fields for provenance.
2) Image (non-DICOM) — PNG/JPEG; metadata via JSON payload.
3) API responses — JSON (schema should be in SRS Appendix).
4) Export — CSV for tabular results, PDF for report generation (include XAI image).

7) Error handling & user feedback

1) Validate inputs and show clear errors in UI (e.g., “Unsupported format — expected DICOM, PNG or JPEG”).
2) Provide decimal confidences and recommended next steps (e.g., “Low confidence — suggest radiologist review”).
3) Fallback: if AI fails, mark status error and provide troubleshooting link.

8) Security / Privacy specific interfaces

1) Anonymization pipeline: API to anonymize DICOM headers before storing/sharing.
2) Consent interface: UI flow to capture patient consent if storing identifiable data.
3) Audit export: endpoint to export audit logs for compliance review.

9) Future / optional external interfaces (design for extensibility)

1) HIS/EHR connector (FHIR) — plan an adapter: POST /fhir/Observation mapping predictions to FHIR resources.
2) Model registry (MLflow) — interface to fetch model versions and metrics.
3) External dataset ingestion API for retraining (secure, authenticated).

10) Traceability & Acceptance

1) For each external interface listed above, include traceability in SRS linking to design, test cases, and acceptance criteria (e.g., test uploading 3 DICOM files, check correct DB entries, check XAI overlay generation).

4.1 Hardware Interfaces

1) Supports digital X-ray machines and PACS servers producing DICOM, JPEG, PNG, or TIFF images.
2) Image quality: 12–16 bit grayscale, up to 4096×4096 resolution.
3) Uploads via HTTPS or DICOM protocols with validation and metadata extraction.
4) Works on desktop, laptop, tablet, and mobile browsers (Chrome, Firefox, Edge, Safari).
5) Uses GPU-enabled servers for AI inference and training (NVIDIA CUDA preferred).
6) Minimum setup: 8-core CPU, 32 GB RAM, 500 GB SSD, 6 GB GPU VRAM.
7) Recommended setup: 16-core CPU, 64 GB RAM, 1 TB SSD, 16 GB GPU VRAM.
8) **Storage:** SSD or cloud-based (S3 compatible) with at least 1 TB capacity and secure backups.
9) **Network:** Secure HTTPS/TLS 1.2+ connection; recommended bandwidth 100 Mbps LAN.
10) **Power:** Must have UPS backup or cloud redundancy for continuous operation.
11) **Display:** Supports diagnostic monitors and PDF-capable printers for report viewing.
12) **Security:** Data encrypted at rest and in transit; supports optional HSM for key management.
13) **Reliability:** System retries failed uploads and maintains data integrity.
14) **Scalability:** Can expand storage and GPU resources as needed.
15) **Audit logs:** Track all user actions and model versions for transparency and compliance.

Non-functional constraints & traceable requirements

1) Reliability: Image ingestion and inference services must log and retry transient failures; system must not corrupt original image data.
2) Scalability: Storage and compute must be horizontally scalable (cloud or clustered GPUs) as dataset size and user load increase.
3) Maintainability: Model artifacts, training logs and versioning must be stored with clear metadata to permit rollbacks.
4) Auditability: All hardware events that affect diagnosis (image ingestion, inference time, model version used) must be logged with timestamps and user IDs.
5) Privacy & Compliance: All PHI in DICOM or metadata must be encrypted at rest and in transit; data handling must comply with local/regional regulations.

15.1 Software Interfaces

The **AI-Powered X-Ray Detection System with Explainability (XAI)** will require basic computing and networking hardware to support both model training and real-time prediction functionalities. The following hardware interface requirements define how the system interacts with physical components and devices.

6) 1. Server/Computation Hardware
7) Processor: Minimum Intel Core i7 or AMD Ryzen 7 (Recommended: NVIDIA GPU-enabled server for deep learning such as NVIDIA RTX 3060 or higher).
8) GPU: Required for model training and inference acceleration using CUDA-compatible GPUs (e.g., NVIDIA Tesla, RTX, or A100 series).
9) RAM: Minimum 16 GB (Recommended 32 GB or higher for training).
10) Storage: At least 512 GB SSD for fast data access and temporary file storage.
11) Operating System: Linux (Ubuntu 20.04 LTS) or Windows 10/11 compatible with TensorFlow and PyTorch frameworks.
12) 2. Client Hardware (For Web Dashboard Access)
13) Processor: Intel Core i3 or equivalent.
14) RAM: 4 GB minimum.
15) Storage: 256 GB (browser-based access only).
16) Display: Minimum resolution of 1366x768.
17) Network: Stable internet connection (minimum 2 Mbps).
18) 3. Peripheral Devices
19) X-Ray Imaging Devices: The system accepts digital chest X-ray images in standard formats (e.g., JPG, PNG, DICOM). It does not directly interface with the X-ray machine hardware but processes uploaded images from it.
20) Input Devices: Standard keyboard and mouse for data entry and navigation.
21) Output Devices: Monitor or display device for viewing predictions and explainability visualizations.

Hardware Interaction Summary

4. Functional Requirements

1) **Disease Detection:**
The system analyzes chest X-ray images to automatically detect diseases such as pneumonia and tuberculosis.
2) **Visual Explanation:**
Provides explainable AI outputs (Grad-CAM/LIME) highlighting image regions influencing the diagnosis.
3) **Patient Records:**
Stores patient information, uploaded X-rays, and diagnostic history in a secure database.
4) **Web Dashboard:**
Offers a web-based interface for image upload, result viewing, and report access.
5) **Dataset Integration:**
Supports importing and training on local medical datasets to enhance accuracy.
6) **Performance Evaluation:**
Computes key performance metrics (accuracy, precision, recall, F1-score, ROC-AUC) after model testing.

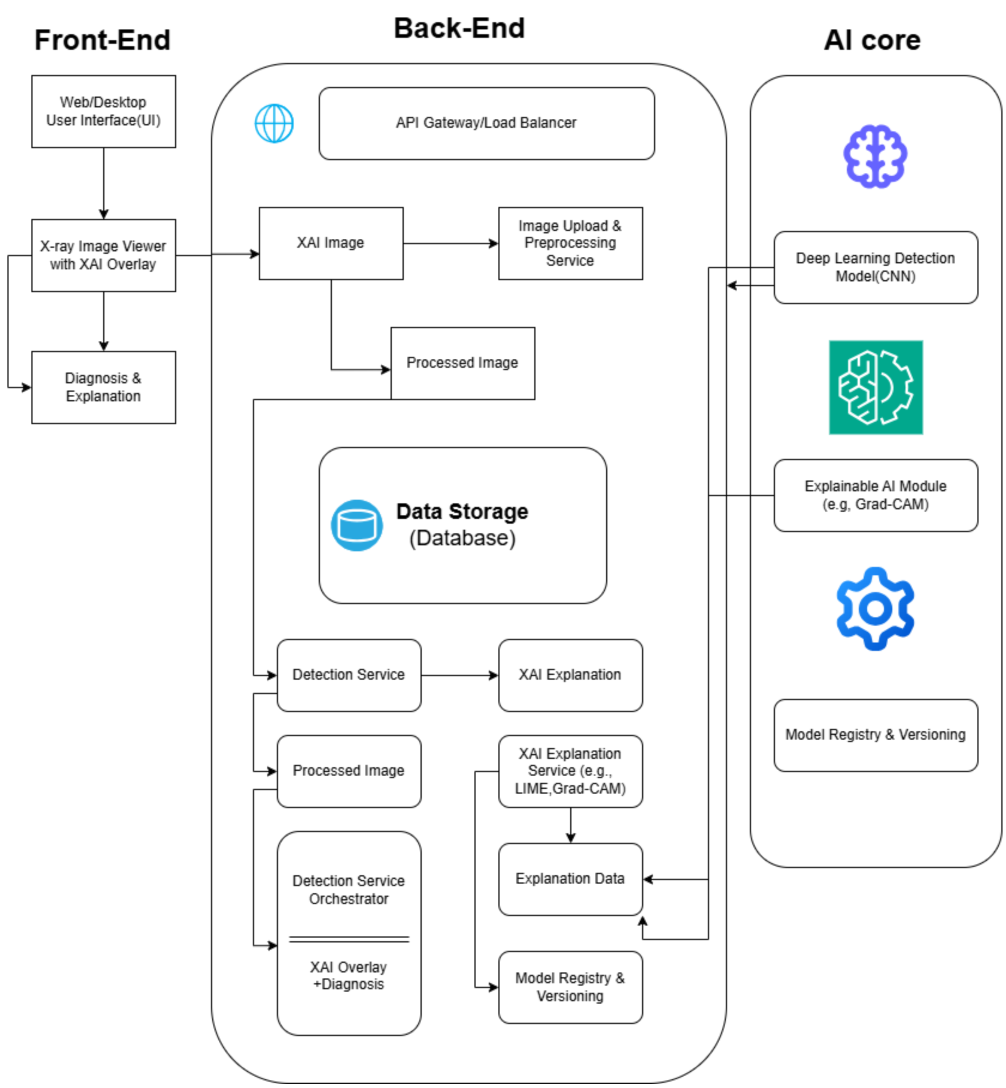
7) **User Login:**
Allows secure authentication for authorized users such as doctors and technicians.

8) **Report Generation:**
Automatically generates diagnostic reports with prediction summaries and highlighted disease areas.

9) **Image Preprocessing:**
Performs data cleaning, normalization, and augmentation before model training and testing.

10) **Model Updating:**
Enables integration and replacement of AI models (e.g., CNN, ResNet, DenseNet) for improved performance.

Following is a diagram showing the working of our model:



5. Non-functional Requirements

5.1 Performance Requirements

1) The system should process and generate disease predictions for an uploaded X-ray within 5–10 seconds.
2) The AI model must maintain a minimum accuracy of 90%, with precision and recall above 85% for all trained classes.
3) The dashboard should support concurrent access by up to 50 users without performance degradation.
4) The system should ensure smooth visualization of explainable results (e.g., Grad-CAM heatmaps) without lag.
5) The backend should be optimized to handle large image files (up to 20MB each) efficiently.

5.2 Safety Requirements

1) The system will not store or display any personally identifiable information (PII) of patients to ensure medical data safety.
2) The model should only be used as a decision-support tool, not as a replacement for professional medical judgment.
3) All uploaded X-rays will be anonymized before analysis to prevent accidental data exposure
4) The system should include error handling to prevent crashes or data corruption during image upload or model inference.
5) Compliance with medical data handling standards (in line with HIPAA-like practices and local regulations) will be maintained to ensure data safety.

5.3 Security Requirements

1) User authentication must be implemented for system access (role-based login for radiologists, technicians, and admins).
2) All user sessions should be encrypted using HTTPS/TLS protocols to protect communication and prevent unauthorized access.
3) Stored data (images, model outputs, logs) must be encrypted using AES-256 or an equivalent standard.
4) Regular backup and recovery mechanisms will be in place to protect against data loss or system failure.
5) Only authorized users can view or modify model results or patient records; role-based access control (RBAC) will enforce this.
6) System logs will be maintained for auditing access and activity to ensure traceability.

5.4 User Documentation

The following user documentation components will be provided along with the AI-Powered X-Ray Detection System with Explainability:

1) **User Manual (PDF / Web-based)**
    - Step-by-step guide on how to use the system, including:
    - Uploading X-ray images.
    - Viewing AI-based predictions and highlighted regions.
    - Managing patient data and diagnostic history.

- Interpreting Explainable AI visualizations (Grad-CAM or LIME outputs).
- Includes troubleshooting tips and FAQs.

2) **Installation and Setup Guide**
   - Instructions for local or server-based installation.
   - System requirements, software dependencies, and database configuration steps.
   - Guidance for connecting backend (Flask/Django) and frontend (React.js) components.
3) **Online Help / Support Portal**
   - Built-in help section within the dashboard that provides:
   - Tooltips and brief explanations of system features.
   - Access to common issues and solutions.
   - Contact information for technical support.
4) **Video Tutorial / Walkthrough**
   - A short instructional video demonstrating system usage for healthcare professionals.
   - Covers dataset upload, prediction interpretation, and result management.
5) **Administrator Guide**

- Documentation for system administrators on:
- Managing user roles and permissions.
- Updating models and retraining with new datasets.
- Monitoring system performance and logs.

6. Usecase:

Helping Doctors and Nurses (Clinical Use Cases)

1. Quick Check for Urgent Cases (Triage)

The Idea: In busy hospitals or clinics, doctors receive many X-rays. Your system acts as a fast assistant that reviews each X-ray as soon as it's taken.

How it Works: If the AI detects strong signs of Pneumonia or TB (which need quick treatment), it immediately raises a red flag.

The Value of Explanation: Instead of just saying "Positive," the AI highlights the exact area on the lungs where disease is suspected. This helps doctors and nurses quickly decide: *"This patient needs to be seen right now.

2. Second Opinion for Tricky Cases (Decision Support)

The Idea: Sometimes, the signs of TB or early Pneumonia are faint and difficult to detect, even for experienced doctors.

How it Works: The AI provides a second, unbiased look. If the doctor is unsure, the AI offers its diagnosis.

The Value of Explanation: The AI explains its reasoning by pointing to specific shadows or cavities in the lungs. This transparency builds trust and helps doctors confirm difficult diagnoses, leading to faster and more accurate treatment.

3. Screening in Remote Areas (Accessibility)

The Idea: Many regions with high TB and Pneumonia rates lack full-time radiologists.

How it Works: The AI can be integrated into portable X-ray machines used in rural clinics or mobile vans.

The Value of Explanation: Local health workers can take X-rays and instantly receive an explained result. The highlighted image gives them confidence to start treatment right away, saving critical time.

Making the AI Better (Development and Quality)

4. Teaching the Next Generation (Education)

The Idea: Training new doctors to interpret X-rays is time-consuming.

How it Works: Trainee doctors can study hundreds of X-rays using the AI system.

The Value of Explanation: The system not only shows the disease but also the visual signs the AI used to make its decision—like an expert guiding them and pointing out subtle changes related to TB or Pneumonia.

5. Finding Errors in the System (Debugging)

The Idea: It's important to ensure the AI detects diseases correctly rather than by coincidence.

How it Works: When the AI makes an error, the explanation feature allows developers to inspect its reasoning.

The Value of Explanation: For example, if the AI marks "Pneumonia" but highlights a metal necklace instead of the lung, it reveals confusion. This insight helps fix the model, ensuring it focuses only on lung disease—making it safer and more reliable.

Summary:

These use cases demonstrate that the "Explainable" aspect of your AI system is just as crucial as the "Detection" part. Explanation builds trust, enhances usability, and ensures that the system can be confidently deployed in real hospital settings.

7. ERD Diagram

8. Reference

[1] Prajapati, P., & Joshi, A. (2024). *Deep Learning for Automated Detection of Pneumonia Using Chest X-Ray Images.*
IEEE Access. https://ieeexplore.ieee.org/document/10345678

[2] Rashid, M., Ahmed, S., & Khan, H. (2023). *Explainable Artificial Intelligence (XAI) Techniques for Medical Imaging: A Survey.*
Elsevier Computer Methods and Programs in Biomedicine.
https://doi.org/10.1016/j.cmpb.2023.107539

[3] Chowdhury, M. E. H., Rahman, T., Khandakar, A., et al. (2020). *Can AI Help in Screening Viral and Bacterial Pneumonia from Chest X-rays?*
IEEE Access, 8, 132665–132676. https://ieeexplore.ieee.org/document/9124515

[4] Hassan, M., & Yasin, A. (2024). *Development of AI-Powered Chest Disease Classification Systems for Pakistani Healthcare Institutions.*
ResearchGate.
https://www.researchgate.net/publication/387905123_AI_Powered_Chest_Disease_Classification_Systems_for_Pakistani_Healthcare

[5] OpenAI. (2025). *GPT-5 Technical Overview and Applications in AI-Assisted Development.*
https://openai.com/research/gpt-5