SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Multimodal Lung Diagnosis AI System

---

1. INTRODUCTION

1.1 Purpose

This document explains the requirements and working of the Multimodal Lung Diagnosis AI System. The system helps in detecting lung diseases by analyzing chest X-ray images and clinical text reports together. It also provides a symptom checking feature and allows patients to send reports to doctors through an internal communication system.

1.2 Scope

The project is designed as an AI-based healthcare assistant that supports both patients and doctors. Patients can check symptoms, upload reports, and receive simplified results. Doctors can view patient submissions, access detailed analysis, and use AI insights to support diagnosis.

Main goals:

- Improve disease detection using multimodal AI.
- Provide early guidance through symptom checking.
- Enable patient–doctor report sharing.
- Offer explainable AI results.

1.3 Intended Users

- Patients who want early health guidance.
- Doctors who need detailed medical insights.
- Developers and reviewers evaluating the system.

---

2. OVERALL SYSTEM DESCRIPTION

2.1 Product Perspective

The system is a web-based platform consisting of two main interfaces:

- Patient Interface
- Doctor Interface

Both interfaces connect to a backend server that runs AI models and manages data storage.

2.2 Product Functions

The system will:

- Allow patients to enter symptoms.
- Allow patients to upload X-ray images and clinical text.
- Run AI models to analyze medical data.
- Generate predictions and explanations.
- Save patient reports in a database.
- Allow doctors to view stored reports and analysis.

2.3 User Roles

Patient

- Enters symptoms.
- Uploads reports.
- Views simplified results.

Doctor

- Accesses stored patient reports.
- Views detailed AI analysis.
- Uses visual explanations for decision support.

2.4 Operating Environment

- Operating System: Windows 10/11
- Programming Language: Python
- Framework: FastAPI or Flask
- Libraries: PyTorch/TensorFlow, OpenCV, NumPy, Pandas
- Database: Any supported database (SQL or NoSQL)

---

3. SYSTEM WORKFLOW REQUIREMENTS

3.1 Patient Interface Workflow

3.1.1 Symptom Checker

The system shall allow patients to enter symptoms.
The backend shall analyze symptoms using simple logic or NLP and provide advice:

- Monitor condition.
- Visit doctor.

3.1.2 Multimodal Diagnosis

The system shall allow patients to upload:

- Chest X-ray image.
- Clinical text report.

The system shall:

1. Preprocess image and text data.
2. Extract features using separate AI models.
3. Combine features using multimodal fusion.
4. Predict disease type and severity level.
5. Generate heatmaps and explanation text.

3.1.3 Data Storage

The system shall save:

- Patient inputs.
- Prediction results.
- Date and time.

3.1.4 Patient Output

The patient interface shall display:

- Simple diagnosis summary.
- Easy explanation.
- Guidance message.

3.2 Doctor Interface Workflow

3.2.1 Doctor Dashboard

The system shall provide a dashboard where doctors can view submitted patient reports.

3.2.2 Report Viewing

The backend shall retrieve stored data from the database and display:

- Patient inputs.
- AI predictions.
- Heatmaps.
- Severity level.

3.2.3 Detailed Analysis

Doctor Mode shall include:

- Probability scores.
- Visual explanations.
- Full AI reasoning output.

---

4. FUNCTIONAL REQUIREMENTS

FR1 — System shall accept symptom text input.
FR2 — System shall accept chest X-ray image upload.
FR3 — System shall accept clinical text reports.
FR4 — System shall preprocess image and text data.
FR5 — System shall run image and text models separately.
FR6 — System shall combine features using multimodal fusion.
FR7 — System shall predict lung disease type.
FR8 — System shall estimate severity level.
FR9 — System shall generate heatmaps and explanations.
FR10 — System shall store reports in database.
FR11 — System shall allow doctors to view saved reports.

---

5. NON-FUNCTIONAL REQUIREMENTS

Performance

- Results should be generated within a few seconds.

Usability

- Patient interface must be simple and easy to understand.
- Doctor interface must clearly present technical details.

Reliability

- System should work consistently for different inputs.

Security

- Patient data must be stored securely.

Scalability

- System should support future expansion and cloud deployment.

---

6. SYSTEM REQUIREMENTS

Hardware

- Minimum 8 GB RAM.
- Standard PC with internet access.

Software

- Python environment.
- Deep learning frameworks.
- Image processing libraries.
- Database system.

---

7. EXTERNAL INTERFACE REQUIREMENTS

Patient Interface

- Symptom input form.
- Upload button for X-ray and text.
- Result display page.

Doctor Interface

- Dashboard listing patient reports.
- Detailed analysis view.

---

8. FUTURE ENHANCEMENTS

- Real-time chat between patient and doctor.
- Cloud-based deployment.
- Advanced transformer-based multimodal models.
- Integration with hospital systems.
