Hazard Analysis
Scanalyze AI

Team 16, Ace
Harrison Chiu
Hamza Issa
Ahmad Hamadi
Jared Paul
Gurnoor Bal

Contents

Table 1: Revision History

Date	Developer(s)	Change
25 October 2024	Harrison	Section 5, 6
25 October 2024	Hamza	Section 6
25 October 2024	Jared	Section 1, 3
25 October 2024	Gurnoor	Section 2
25 October 2024	Ahmad	Section 7
2 April 2025	Harrison, Hamza, Jared, Gurnoor, Ahmad	Revised all sections to fit with final product

1 Introduction

This document outlines the potential hazards and corresponding controls for the "AI for Chest X-ray" project, focusing on convolutional neural networks (CNNs) to automatically detect lung and heart conditions. A hazard, in this context, refers to any condition, event, or malfunction that could compromise system accuracy, reliability, security, or patient safety. Conducting a thorough hazard analysis is crucial to mitigate risks, ensure legal compliance (e.g., HIPAA), and improve the system's safety and reliability for clinical decision support, while safeguarding patient data integrity and confidentiality.

In developing machine learning systems for medical diagnostics, hazard analysis is essential to identify potential risks, whether from system failures, unintended consequences, or improper functionality. Unlike traditional software systems, machine learning models function as "black boxes," making it difficult or impossible to formally verify correctness or understand the internal logic of their decisions. This inherent uncertainty increases the importance of comprehensive hazard analysis to anticipate failure modes and mitigate risks. These risks can lead to harm, mislead users with erroneous outputs, or negatively impact the system's performance. For this project, hazards encompass both software issues and interactions between the AI system and users, such as radiologists and medical institutions. The goal of this hazard analysis is to assess these risks, evaluate their potential impacts, and implement measures to ensure the system functions safely and effectively in a real-world medical environment.

2 Scope and Purpose of Hazard Analysis

The purpose of this hazard analysis is to identify and mitigate risks associated with the "AI for Chest X-ray" system, particularly in clinical environments where decisions informed by AI carry significant consequences. Hazards may arise from various sources, including image misclassification (false positives/negatives), compromised data privacy, system downtime, and failures in the user interface or backend infrastructure.

The scope of this hazard analysis includes the full lifecycle of the system, from data acquisition and preprocessing, through CNN inference and diagnostic report generation, to user interaction and system maintenance. This includes the frontend web application used by radiologists, the backend infrastructure hosting the CNN model, and the data pipelines processing public datasets (e.g., CheXpert, MIMIC).

While the analysis focuses on components developed and maintained by the project team, it excludes external infrastructure such as hospital-managed servers, third-party PACS systems, or internet service reliability, unless they directly impact the behavior of the AI system.

By analyzing hazards within this defined scope, the document seeks to safeguard diagnostic accuracy, ensure regulatory compliance (e.g., HIPAA, PIPEDA), and maintain clinical trust in the system.

3 System Boundaries and Components

The system under analysis is designed to facilitate the detection of lung and heart conditions from chest X-ray images using convolutional neural networks (CNNs). It consists of several integrated components that together enable real-time diagnostic support for medical professionals in clinical and research settings.

The AI system interacts with both human users (e.g., radiologists, clinicians) and digital services (e.g., hospital information systems) through a secure and user-friendly web-based platform. It enables clinicians to upload chest X-ray images, receive automated multi-label disease classifications, and visualize important diagnostic regions using Grad-CAM heatmaps.

Frontend

1. A web interface for user interaction with secure login and session control.
2. Functionalities for uploading X-ray images, receiving diagnosis results, and viewing Grad-CAM heatmaps.
3. Role-based access to diagnostic notes, audit logs, and diagnostic history.

Backend

1. A CNN-based classification server that accepts images and returns disease predictions with confidence scores.
2. Secure databases for storing processed X-ray images, diagnostic predictions, and user logs.
3. RESTful API endpoints for handling model inference, image storage, and diagnostic result delivery.

Training Pipeline (External to deployment scope)

1. An offline model training environment utilizing publicly available datasets (e.g., CheXpert, MIMIC-CXR).
2. Preprocessing modules to normalize, resize, and label training images.
3. Evaluation procedures to monitor model performance, bias, and generalization.

The system boundary includes only components deployed and maintained by the development team. These include the user-facing web application, backend AI services, and associated internal databases. Excluded from the scope are external hospital IT networks, PACS servers, and unmanaged end-user devices. Clearly defining this boundary ensures that hazard identification focuses only on parts of the system where the team can directly implement mitigations or safety measures.

4 Critical Assumptions

In conducting the hazard analysis for the CNN-based chest X-ray analysis system, several critical assumptions are made regarding the functionality, environment, and usage of the system. These assumptions are necessary to define the scope of potential hazards and their mitigation strategies. While assumptions help simplify the analysis, they also highlight areas where potential risks may need to be revisited as the system evolves.

1. **Assumption 1: Model Performance**
   It is assumed that the convolutional neural network (CNN) model will perform within the expected accuracy levels (60%) as defined by the system requirements. This implies that the model has been properly validated on representative data. However, any deviation from this performance (e.g., degradation in unseen settings) could lead to unsafe predictions and misdiagnoses.

2. **Assumption 2: Dataset Quality and Relevance**
   The datasets used for training (CheXpert, MIMIC-CXR, NIH) are assumed to be representative of real-world chest X-ray images and correctly annotated. If these assumptions fail (e.g., due to dataset bias, poor labeling quality, or underrepresentation of minority groups), the model may generate inaccurate, biased, or non-generalizable results.

3. **Assumption 3: Web Application Stability**
   It is assumed that the web application operates reliably across standard devices and browsers with typical internet connectivity. Failures such as latency, browser incompatibility, or server outages could interrupt image uploads or result delivery, delaying diagnoses in time-sensitive settings.

4. **Assumption 4: Data Privacy and Security**
   The system is assumed to follow appropriate data privacy policies and not collect any personally identifiable information (PII). If this assumption fails due to unintentional logging or insecure storage, users and patients may be exposed to regulatory or ethical risks (e.g., HIPAA/PIPEDA violations).

5. **Assumption 5: User Competency**
   It is assumed that users (e.g., radiologists, clinicians) possess adequate domain knowledge to interpret results responsibly. However, given the model's black-box nature, a lack of training or experience could lead to misuse or overreliance on predictions.

6. **Assumption 6: Secure Integration with Hospital Systems**
   It is assumed that integration with hospital IT infrastructure and PACS is secure, encrypted, and compliant with healthcare standards. If this assumption

is invalid, data transfers could become a point of vulnerability.

7. **Assumption 7: Backend and Infrastructure Availability**
   It is assumed that backend components (APIs, servers, databases) will be accessible during clinical use, except during maintenance. Any unexpected downtime or failure may lead to incomplete workflows or delays in clinical decision-making.

5 Failure Mode and Effect Analysis

This section presents the hazards identified in the Chest Scan system using a Failure Mode and Effect Analysis (FMEA) approach. The hazards include technical failures, misclassifications, data access issues, and user interface risks. These risks are evaluated by component, potential failure mode, cause, effect, and corresponding mitigation strategies. Tables 2 and 3 summarize these failure scenarios.

Table 2: FMEA Worksheet Part 1: Frontend, Backend, and Model

Component	Failure Modes	Effects of Failure	Causes of Failure	Detection	Controls	Risk	Recommended Action	Req.	Ref.
User Access (Web App)	Fail to authenticate authorized user	Radiologist or technician cannot access system to view results	Incorrect login credentials or broken auth module	Manual testing, log reviews	Alternative login flow, session timeout handler	Low	Add backup login method and authentication retry	AR1, SR2	H1.1
	Unauthorized access by outsider	Breach of sensitive data or results	Weak password policy or missing rate limits	Penetration tests, log analysis	Password hashing, MFA, CAPTCHA	Medium	Enforce password policies, limit IP login attempts, use MFA	AR0, SR3	H1.2
Image Upload (Frontend to Backend)	Failure to upload CXR	AI model receives no input for classification	Slow connection, bad file format, JS failure	Manual upload test, server logs	File validator, size checker	Low	Retry upload on failure, enforce accepted formats	–	H2
Image Fetch (Backend to Frontend)	Failure to load result image	Radiologist sees broken preview, no heatmap	Timeout or backend fetch failure	User reports, frontend log	Retry fetch or preload fallback image	Low	Add async retry with loading spinner UI	–	H3
Result Display	Report not shown or corrupted	Clinician can't interpret findings	Incomplete JSON, broken rendering logic	Console errors, schema mismatch test	Input sanitization, response schema validator	Medium	Add UI fallback and log corrupted entries for QA review	–	H4.1
	Wrong report shown for wrong image	Misdiagnosis, legal liability	Race condition, output mislabeling	Manual test with mismatched upload, audit logs	Record-ID binding	High	Tag outputs with unique hash, verify via audit trail	SR0	H4.2
CNN Model	False positive (predicts condition not present)	Unnecessary treatment, patient stress	Model overfit on noisy training samples	ROC curve and test set misclass check	Use calibrated thresholds, ensemble validation	High	Adjust decision thresholds and retrain on more negatives	SR0, SR1	H5.1
	False negative (misses actual disease)	Missed diagnosis, worsened outcomes	Lack of data for rare class or subtle features	Confusion matrix on critical class	Data augmentation for underrepresented class	High	Use focal loss and boost rare class sampling	SR0, SR1	H5.2
Model Training Process	Overfitting during training	Model fails in real-world unseen cases	Too many epochs, small train set, high variance	Gap in train/validation accuracy, generalization test	Early stopping, dropout, K-fold validation	High	Use regularization and mix training datasets	SR1	H6

Table 3: FMEA Worksheet Part 2: Security, Infrastructure, and Data Handling

Component	**Failure Modes**	**Effects of Failure**	**Causes of Failure**	**Detection**	**Controls**	**Risk**	**Recommended Action**	**Req.**	**Ref.**
Data Security	Unauthorized access to DB	PHI breach, HIPAA violation	Weak auth layer, no encryption at rest	Intrusion detection tools, access logs	AES-256 encryption, RBAC	High	Add IP whitelist, rotate keys regularly	AR2, SR3	H7.1
	Broad internal access	Insider threat risk	Dev access to full dataset in prod	Manual audit, test access scoping	Minimized data exposure	Medium	Use anonymization and access tiering	AR0, SR3	H7.2
Data Transfer Layer	Transfer interrupted or dropped	Incomplete or corrupted transmission	Timeout, unstable net, backend crash	Transfer logs, retry pattern detection	Retry queue, ACK-reply protocol	Medium	Use auto-resume upload/download mechanism with hash check	SR3	H8
Backend Server	Outage (DB/API down)	Model unavailable for triage	Container crash, memory leak, kernel fault	Liveness probe, API response time	Auto-restart on failure	Medium	Use container orchestration like Docker Compose or K8s	–	H9.1
	Overloaded model queue	Delayed results, missed deadlines	High image volume, no async queuing	Load metrics, backlog size alerts	Thread pooling, async queue	Medium	Add queue buffer and model load balancer	–	H9.2
Persistent Storage	Loss of training/report data	Re-training required, loss of trust	Disk crash, accidental deletion	Backup log schedule, system alerts	Daily snapshot + cloud sync	High	Enable automated restore from encrypted cloud archive	AR2, SR3	H10.1
	Silent data corruption	Subtle misdiagnoses or wrong history reports	Bit rot, disk error, write race	Periodic DB checksum validation	SHA256 hash on upload	Medium	Enable end-to-end integrity verification on reads	–	H10.2

6 Safety and Security Requirements

6.1 Access Requirements

AR0: Access to patient data and diagnostic records shall be restricted to authenticated and authorized medical professionals.

Rationale: This is essential to protect personal health information (PHI) and meet legal requirements such as HIPAA.

Fit Criterion: Access to the system is limited to verified users through role-based authentication and enforced by access logs and permission tiers.

AR1: The system's administrative dashboard shall only be accessible through secure login by authorized technical staff.

Rationale: Limiting system-wide configuration access reduces the likelihood of unintended disruptions or misuse.

Fit Criterion: Access attempts to the admin dashboard must be logged and blocked if credentials are not validated through 2FA.

6.2 Integrity Requirements

AR2: All sensitive medical data, including uploaded X-rays, diagnostic results, and user credentials, shall be encrypted at rest and in transit.

Rationale: Encryption ensures confidentiality and protects against interception or unauthorized modifications during communication and storage.

Fit Criterion: The system shall use AES-256 encryption for data at rest and HTTPS/TLS for all data in transit. Security audits will verify compliance.

6.3 Safety Requirements

SR0: The system shall display classification confidence scores alongside each diagnostic prediction.

Rationale: Transparency about model confidence helps radiologists assess uncertainty and make better-informed decisions.

Fit Criterion: Each diagnostic report will include labeled confidence percentages (e.g., "Pneumonia: 87%"), and a red alert will appear for scores under 50%.

SR1: Diagnostic predictions must be reviewed and approved by a qualified radiologist before being stored or exported.

Rationale: AI-generated predictions are considered assistive; final medical judgments must come from certified professionals.

Fit Criterion: Reports will be tagged as ”Pending Review” until explicitly confirmed by a logged-in radiologist. All unconfirmed results remain in a sandbox state.

SR2: Model performance shall be monitored and validated routinely using a reserved validation dataset.

Rationale: Continuous performance tracking ensures long-term reliability and early identification of model degradation.

Fit Criterion: Accuracy on the validation dataset shall remain above 55%, with drift alerts issued when performance drops more than 5% over a two-week period.

SR3: All transmission of sensitive user data shall be protected using secure encryption protocols.

Rationale: Protecting patient health data from exposure ensures legal compliance and builds user trust.

Fit Criterion: HTTPS must be enforced site-wide. TLS 1.2 or higher will be verified during deployment audits.

7 Roadmap

This section outlines the development roadmap for the Chest Scan system. It provides a timeline of key deliverables and phases, ensuring that priority features are developed, tested, and evaluated within the capstone project schedule. The plan accounts for system goals, safety/security requirements, and stakeholder needs.

Phase 1 - September-October 2024: Project Initialization and Requirements Gathering

- Finalize team and supervisor approval.
- Submit problem statement and preliminary research plan.
- Draft and submit SRS Revision 0.
- Begin hazard analysis and identify key system risks.

Phase 2 - November 2024: Proof of Concept and Data Pipeline

- Implement initial pipeline for X-ray image upload and preprocessing.
- Demonstrate integration of publicly available datasets (CheXpert, MIMIC-CXR).
- Present Proof of Concept with frontend upload + backend model response.
- Begin writing V&V Plan Revision 0.

Phase 3 - December 2024 - January 2025: Core Feature Implementation

- Train and validate CNN model for disease classification (FR1-FR4, SR0).
- Integrate Grad-CAM for visual explainability (FR5).
- Enable confidence score and report generation (FR4, SR0).
- Begin unit and integration testing across frontend/backend.

Phase 4 - February 2025: Testing and Security Hardening

- Implement access controls, encryption (AR0-AR2, SR3).
- Conduct usability tests and address report display (FR6).
- Demonstrate working system in Revision 0 milestone.

Phase 5 - March 2025: Finalization and Review

- Finalize validation dataset and assess model performance (SR2).
- Conduct V&V testing and performance audits.
- Prepare final demonstration and documentation (SRS Rev. 1, Design Rev. 1).

Future Work (Post-Capstone)

- Expand disease classification labels (e.g., nodules, tuberculosis).
- Integrate role-based dashboards for institutions.
- Deploy scalable API endpoints for clinical trials or research partners.
- Conduct regular security audits and model revalidation (SR1, SR2).

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. **What went well while writing this deliverable?**
   One aspect that went well was defining the safety and security requirements for the project. Since we had previously created a comprehensive SRS document, we were able to re-use and adapt many of the non-functional requirements, which helped streamline the hazard analysis. Our assumptions were straightforward to align as we had already discussed and agreed on them based on prior team meetings and system planning. Furthermore, dividing the system into frontend, backend, and model components gave structure to our risk identification, which made the FMEA process more manageable. The collaborative team environment also played a big role; brainstorming was smooth and delegation of sections was efficient.

2. **What pain points did you experience during this deliverable, and how did you resolve them?**
   One of the main challenges was defining the system boundaries, especially in a project like ours that involves a CNN model and multiple stakeholders (radiologists, backend engineers, users). It was initially unclear which parts to include within scope, particularly when it came to integration with hospital IT infrastructure or EHRs. This complexity was amplified by the research-based nature of our project, where boundaries may evolve over time. To address this, we adopted a flexible scope definition that could be revised later. We focused on core operational components (upload, classification, report generation) for this version of the hazard analysis while noting the potential for future updates.

3. **Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones, how did they come about?**
   Previously identified risks:

- Fail to Authenticate User (Web Application)
- Fail to Load X-ray Images (Frontend Image Viewer)
- Incorrect Disease Report Display (UI Output Mapping)
- False Positive / False Negative Diagnoses (CNN Model)
- Model Overfitting (Training Process)
- Cyberattacks / Unauthorized Access (Security Layer)
- Network Failures and Server Downtime (Infrastructure)
- Data Corruption or Loss (Persistent Storage)

New risks identified during this deliverable:

- *Image Upload Failure to Backend:* This arose while defining system boundaries and exploring data flow - we realized that image preprocessing or encoding errors could occur prior to classification.
- *Incorrect Display of Reports:* While analyzing how the frontend renders classification data, we considered that parsing or ID mismatch could cause diagnostic results to be shown for the wrong patient.
- *Unauthorized Access to Admin Portal:* This came up while documenting access requirements - since model parameters and user data are sensitive, exposing admin-level functionality without 2FA was flagged as a major security risk.

4. **Other than the risk of physical harm, list at least 2 other types of risk in software products. Why are they important to consider?**
   1. Security Risks: These include unauthorized access, data leaks, and vulnerabilities in authentication systems. In healthcare applications, such risks have severe legal and ethical implications under laws like HIPAA. They can undermine user trust and cause long-term damage to system credibility.

   2. Reliability Risks: If the system frequently crashes, returns incorrect predictions, or experiences downtime, clinical workflows are disrupted. This not only causes inefficiencies but may delay diagnoses and compromise patient care. High reliability is essential in any safety-critical software.