Reflection and Traceability Report on Scanalyze AI

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

Contents

1 Changes in Response to Feedback

This section summarizes the changes we have made in regards to the feedback we received from TAs, our project supervisor, peer reviewers, and during usability testing. The changes made can be found in the table below and are linked to the associated GitHub issues.

To improve traceability, we have categorized issues into milestones such as 'TA Feedback Issues,' 'Peer Review,' and 'Final Doc Updates.' can be found in Team-16's Capstone Deliverables Project.

1.1 SRS and Hazard Analysis

Changes to SRS and Hazard Analysis along with the feedback, response, and associated issues can be found in the tables below:

Table 1: Changes for SRS Documentation

Feedback Source	**Feedback Item**	**Response**	**Issue**
TA	Content: The subtitle not being changed from the default doesn't inspire reader confidence from the start! Table of contents missing. Do you not know which organization is sponsoring this? Is there one right now? Or is there going to be? If there isn't, are they even worth including? You reference HIPAA multiple times (great!) but do not cite it. Should use symbolic constants instead of magic numbers.	The subtitle has been updated to: "Software Requirements Specification for Chest Scan: subtitle describing software", which replaces the placeholder text and indicates the content's focus clearly SRS(1). A Table of Contents is present in the document and includes all major sections (e.g., Purpose, Stakeholders, Relevant Facts, Use Cases), indicating that the issue has been resolved SRS. Section 2.1 defines the Client as: "Medical Technology Department and/or healthcare organization sponsoring the initiative," acknowledging that a formal sponsor may not be assigned yet but still including the possibility of sponsorship, which clarifies the intent behind including the section SRS (1). In Section 5.2 (Business Rules), HIPAA is referenced with a clear authority statement: "Authority for the Rule: Federal regulations and organizational policies governing health information," providing regulatory context. However, no direct URL or official legal citation is included, so partial credit may be warranted	#SRS-66

Feedback Source	Feedback Item	Response	Issue
TA	Document Organization: Doesn't describe the template or changes.	The document now includes a statement in the introduction identifying the template followed. Specifically, this SRS adheres to the Volare SRS template used in McMaster's capstone course. Structural changes and customizations are clearly documented, such as the removal of placeholder sections, the integration of symbolic constants, and the alignment of requirement identifiers (e.g., NF-SLR0) to support traceability.	#SRS-67
TA	Complete, Correct and Unambiguous: No irrelevant features" (NF-EUR0) is ambiguous. Also, you seem to imply that this should be true for 100 percent of users, which is never necessarily a good assumption. There may be some incapable of doing the tasks asked, so it's best to use a number like 80 percent or 90 percent as you have in other requirements. Check other requirements for this problem too, like NF-LR0.	Symbolic constants have now been introduced directly in the Non-Functional Requirements section (Section 12) to replace previously hardcoded values. Specific updates include: **NF-SLR0** now uses `MAX_IMAGE_GEN_TIME` instead of the magic number "60 seconds"; **NF-PAR0** uses `MIN_ACCURACY_REQUIREMENT` instead of "65%"; and **NF-PAR1** uses `CONFIDENCE_WARNING_THRESHOLD` instead of "50%".	#SRS-68

Feedback Source	Feedback Item	Response	Issue
TA	Traceable Requirements: No real traceability among requirements; no traceability table provided.	A comprehensive Requirement Traceability Table has been added as Section (BLANK), listing all non-functional requirements (NFRs) alongside the functional requirements (FRs) that satisfy them, with rationale. Additionally, inline traceability annotations (e.g., 'Traces to: FRx') have been included directly within each NFR in Section 12. All FRs and NFRs now follow a bidirectional mapping system that supports navigation and impact analysis - fulfilling the Level 4 rubric standard for traceability.	#SRS-69
TA	Verifiable Requirements: Some requirements are ambiguous even with their fit criteria, since they are also ambiguous.	NF-EUR0 has a verifiable benchmark: "At least 90 percent of users should be able to complete key tasks without additional guidance." NF-LR0 quantifies uptime: "99.9 percent uptime over a one-week period, measured by automated tools." Requirements like NF-UPR0, NF-AR1, and NF-SR2 include user study metrics or acceptance thresholds to confirm usability goals. This ensures that each requirement can be tested objectively, fulfilling the Level 4 rubric for verifiability.	#SRS-70

Feedback Source	**Feedback Item**	**Response**	**Issue**
TA	Phase In Plan: You have a phase-in plan, but do not trace that back to requirements.	The SRS now includes a direct trace from requirements to their planned implementation order. The plan: Identifies priority requirements (e.g., FR1-FR3 for core pipeline) Links those priorities to corresponding Functional/Non-Functional Requirements Specifies a timeline with target dates and phases (e.g., MVP Rev 1, Phase 2 feature rollouts) Ensures that MVP coverage (NF-REL0) includes critical features like FR1, FR3, FR4, and FR5	#SRS-71
TA	(L0 Standards): Instead of just saying ”Applicable privacy laws”, research the ones that would apply, and cite them.	The SRS now explicitly identifies and cites all relevant legal, privacy, and interoperability standards. Under Section 17 Compliance Requirements, the following are documented: HIPAA (U.S. Health Information Privacy Law) PIPEDA (Canadian Privacy Law) DICOM (Medical Imaging Format Standard) HL7 FHIR (Electronic Health Data Interoperability Standard)	#SRS-72
TA	(LO Assumptions): Should have a boundary diagram to make very clear how your project fits in with other components, and which parts you are responsible for.	Edge cases were added as specific functional requirements. Terminology was standardized across the document.	#SRS-73

Feedback Source	**Feedback Item**	**Response**	**Issue**
TA	(L0 Reflect):Good answers, but missing some questions. The rubric stated which ones should be answered.	The appendix now includes a dedicated Reflections and Learning Outcomes section addressing all required questions	#SRS-73
TA	(LOExtKnowledge): Instead of just saying "Applicable privacy laws", research the ones that would apply, and cite them.	External knowledge has been integrated through references to CheXpert, Grad-CAM, and practices from Kaggle and GitHub. These informed model design, data preprocessing, and explainability. The Self-Directed Learning Plan also incorporates literature-based strategies tailored to radiology datasets, frontend tools, and backend design. While not formally cited, these sources significantly influenced the system architecture and requirements.	#SRS-74
TA	Basis For Design: Good issues. One suggestion is to put a bit more detail in the issue title, instead of just e.g. "Functional Requirement 8"	Requirement titles have been expanded to include clear descriptions of functionality (e.g., "Upload chest X-ray images in supported formats" instead of "FR8"). All requirements include rationale, fit criteria, and defined inputs/outputs to support external implementation. The document identifies stable release priorities via NF-REL0 and includes justification where requirements are unlikely to change.	#SRS-75

Table 2: Changes for Hazard Analysis

Feedback Source	**Feedback Item**	**Response**	**Issue**
TA	Document Content: Good introduction - why is it particularly important when using ML? You are right that it is, but should explain just a bit more. Should add something about the fact that it is difficult/impossible to certify/prove algorithm correctness when using ML, or even to know how it works beyond a black box.	The introduction has been revised to emphasize the importance of hazard analysis in ML systems due to their black-box nature and the inability to formally verify algorithm correctness. The updated text clearly highlights the challenges of certifying ML models and how this uncertainty increases the necessity of risk mitigation.	#HA-76
TA	Spelling and grammar and style: Table has some disconnected borders. I recommend using Booktabs tables to fix this. Reference tables in the body of the document. The table is a bit confusing because some cells are shared then split, then shared again as you scan from left to right (e.g. H1.1 and H1.2). I see why you did it, but it can be a bit confusing sometimes.	The FMEA tables have been reformatted for improved readability. Table structure was clarified to reduce visual ambiguity (e.g., H1.1 and H1.2 are now presented with distinct rows and clearer separation of failure modes). All tables are now referenced in the document body to guide the reader.	#HA-77
TA	Boundary, Scope, Definitions and Assumptions: Which pieces of the project are in scope is unclear.	Section 3 (System Boundaries and Components) has been expanded to clearly state which components are within project scope (e.g., frontend UI, CNN model, internal APIs) and which are explicitly out of scope (e.g., hospital-managed servers, EHR systems). Section 4 also outlines critical assumptions tied to this boundary.	#HA-78

Feedback Source	Feedback Item	Response	Issue
TA	Method and its application: ”Cyberattacks” seems too general of a failure mode.	The failure mode labeled ”Cyberattacks” has been replaced with more specific risks such as ”Unauthorized Access by Outsiders” and ”Broad Internal Access.” Each is given a distinct hazard ID (e.g., H1.2, H7.2) and linked to precise causes and mitigation strategies, such as multi-factor authentication or access tiering.	#HA-79
TA	Recommended Actions: Some actions are a bit generic and should be more specific, e.g. ”Improve cybersecurity”	Generic recommended actions like ”Improve cybersecurity” have been replaced with specific, actionable controls such as ”enforce IP whitelisting,” ”rotate API keys regularly,” and ”enable end-to-end hash checks on uploads.” These changes are reflected in the Risk and Recommended Action columns of the FMEA tables.	#HA-80

1.2 Design and Design Documentation

Changes to Design and Design Documentation along with the feedback, response, and associated issues can be found in the tables below:

Table 3: Changes for MG and MIS

Feedback Source	Feedback Item	Response	Issue
TA	SoftArchitec: Timeline: Very un-detailed: no dates, etc.	A revised timeline has been created that breaks down tasks specific to our CNN-based disease classification project, including data handling, model training, frontend/backend integration, and evaluation. Each task now includes responsible team members for clarity.	#DES-95
TA	SoftArchitec: Quality Information: "Explanation of program name" was left instead of filling in that placeholder text.Great anticipated changes!UC1 is obviously out of date now, as I'm sure a lot of stuff is. No marks lost, but this is a reminder to change all of this before Rev1!.	All placeholder content has been removed, and the anticipated changes section has been fully rewritten to reflect the system's actual architecture - including CNN updates, dataset expansion, frontend improvements, and API integration. These are encapsulated within modules based on design-for-change principles.	#DES-94
TA	SoftArchitec: Presentation: Some non-Canadian spellings like behavior. Should use label and ref to properly link tables/figures (e.g. Section 6, traceability matrices, etc). Use “ and ” for proper quotation marks. Hardware/Behaviour-hiding and Hardware/Behaviour encapsulation are used interchangeably. Uses hierarchy diagram should look more like a hierarchy, difficult to tell which things use which. Should embed diagrams as PDFs for better clarity and zoom-ability..	Labeling and formatting inconsistencies have been corrected. The use hierarchy diagram has been recreated as a clean, top-down dependency chart with proper alignment and arrows. Diagrams were exported in high resolution, and references to all tables and figures have been standardized.	#DES-96

Feedback Source	Feedback Item	Response	Issue
TA	DetDesDoc: Syntax: Some tables are rendered incorrectly, with contents spilling from one column to the next.	Tables in the MIS have been reformatted to fit properly within their columns. Overflow issues were addressed, and content layout was adjusted to improve clarity and consistency, especially in the syntax and access routine sections.	#DES-97
TA	DetDesDoc: EnoughToBuild: Some details are unclear (e.g. exact pages in the UI, how they fit together - which might be a good candidate for a state machine).	The module decomposition has been clarified using a unified use hierarchy diagram and MVC categorization. Connections between modules (e.g., how the UI uses the backend, and how data flows through the model pipeline) are now explicitly described in both the MG and MIS.	#DES-98
TA	Participation in Informal Presentation with TA:Try starting earlier on your deliverable next time. Didn't come prepared with questions. Two members didn't come right away. .	Our team took the feedback seriously and restructured our final presentation preparation. Each team member was assigned clear talking points tied to their contributions, and the presentation now directly reflects the updated system architecture and documentation.	#DES-99

1.3 VnV Plan and Report

Changes to VnV Plan and Report along with the feedback, response, and associated issues can be found in the table below:

Table 4: Changes for VnV Plan

Feedback Source	**Feedback Item**	**Response**	**Issue**
TA	Content: Timeline - be sure to use symbolic constants instead of “magic numbers” like 90 percent. Note: as indicated in the comment, unit tests are not required yet as you have not done your design. But you can hopefully reuse some of these later!	All numerical thresholds used in unit and system tests were replaced with clear symbolic constants (e.g., `MIN_CLASSIFICATION_ACCURACY`, `USER_SUCCESS_THRESHOLD`). These constants are documented in Section 7.1 (Symbolic Parameters). While unit tests were not initially required, they were completed after the design was finalized, and symbolic parameters were also applied throughout.	#VnV-89
TA	Spelling and grammar and style: Be sure to use “ and ” for quotation marks. Should remove ? from usability survey questions section title, since you have them..	All quotation marks were corrected to proper LaTeX smart quotes (” ”). The section title containing a question mark (”Usability Survey Questions?”) was updated to ”Usability Survey Questions” to reflect finalized content and eliminate ambiguity.	#VnV-90
TA	System Tests for Functional Requirements are specific: Be specific. Some terms, like ”produce misleading information” are not clear.	System test derivations were rewritten to be precise and measurable. Vague terms such as ”produce misleading information” were removed. Each test now defines observable outputs with thresholded conditions (e.g., rendering within a specific duration or prediction scores exceeding a symbolic constant).	#VnV-91

Feedback Source	Feedback Item	Response	Issue
TA	Tests for Nonfunctional Requirements are specific: You mention HIPAA but provide no information about what its requirements or about how you plan to implement it.	Security-related tests were added to validate HIPAA-relevant behavior: JWT-based authentication, AES-256 encrypted report storage, role-based access restrictions, and verification of no personal data collection. Each nonfunctional test is tied to a clear outcome and test method.	#VnV-92
TA	Traceability to Requirements: Good job including traceability tables; maybe include traceability information inline as well.	Traceability matrices were expanded and refined to map all test cases (FRTC and NFRTC) to their related modules and requirements. Inline traceability was added to test derivation fields for both unit and system tests to clarify what is being verified and why.	#VnV-93

Table 5: Changes for VnV Reprt

Feedback Source	Feedback Item	Response	Issue
Peer Review	Could not validate with pilot radiologist to confirm program's accuracy and real-world use.	Acknowledged the limitation due to lack of connections and resources. Added a note in the document about plans for future clinical evaluation.	#Vnv-81
Peer Review	Add table for notable failures observed and their resolutions.	Added a table summarizing failed test cases, their root causes, and resolutions in the ”Failed Test Cases and Debugging Actions” section.	#Vnv-82

Feedback Source	Feedback Item	Response	Issue
Peer Review	Add image of usability survey and missing SR1 test case.	Included the missing details for the SR1 test case ”Consistent and Simple Style” in the ”Look and Feel Requirements” section. Screenshots of the usability survey were added so the reader knows what kind of questions were asked and how they were presented.	#Vnv-83
Peer Review	Use a table for unit testing outputs.	Compiled all unit testing outputs into a structured table with columns for test name, expected output, and actual output.	#54
Peer Review	Add front-end tests to CI pipeline.	Added front-end unit test files to the repository and described the front-end tests in the ”Frontend Unit Testing and Coverage Analysis” section.	#Vnv-84
Peer Review	Add screenshot of the CI/CD pipeline.	Unsure what to screenshot to show actual pipeline. Instead, added the CI/CD runner code to the document.	#Vnv-85
Peer Review	Update revision history with revisions.	Updated the revision history table to include all recent revisions and feedback addressed.	#Vnv-86
TA	Fix formatting	Formatting of missing newlines before ”Actual Output” and missing section text fixed. No weird upside down ! symbols found on page 14, so that is unchanged.	#Vnv-87
TA	Missing NFR tests	Added NFR test cases to the traceability matrix to show which NFRs are covered.	#Vnv-88

2 Challenge Level and Extras

2.1 Challenge Level

This project was classified as an **advanced** challenge level capstone, as agreed upon with both the course instructor and teaching assistants. The advanced designation stems from the research-heavy scope of the project, the use of real-world medical data, and the goal of extending state-of-the-art methods in chest X-ray classification through both experimentation and academic synthesis.

In contrast to implementation-focused projects, this work demanded a combination of deep technical insight, data-driven experimentation, and clinical relevance. Our team not only developed a working system using a convolutional neural network (CNN) for multi-label classification but also documented our contributions in a full research paper. This paper benchmarks our model against existing approaches and introduces strategies like class balancing, margin-based loss functions, and dynamic thresholding - all essential in handling medical datasets like ChestXray14.

Additionally, the project included development of a usable MVP web interface to support radiologist-facing interactions. The mix of cutting-edge research and stakeholder-centered software design strongly aligned with the expectations for an advanced challenge level.

2.2 Extras

Several extras were incorporated into the project beyond the core requirements, each approved as part of our Problem Statement and included in our final deliverables:

Research Paper Synthesis: A major extra for this capstone was the development of a formal research paper that documents our model design, tuning strategies, experimental comparisons, and clinical implications. This paper builds on recent work in chest X-ray AI and proposes a lightweight CNN-based solution optimized for real-world clinical use, with detailed performance evaluations using recall, accuracy, and AUC-ROC. The inclusion of this paper elevates the academic impact of the project.

Usability Testing Survey (Planned): We designed a usability survey for potential end-users, including healthcare researchers or students familiar with diagnostic imaging. The survey aims to collect qualitative feedback on how intuitive and interpretable our interface and model outputs are. The testing is limited in scope due to deployment constraints but represents our intent to consider real-world human factors - a valuable extra rarely explored in research-heavy projects.

Bonus Features in MVP: Two "bonus" features were added to the MVP interface to enhance its practical value and demonstrate added functionality beyond baseline classification:

- **Model Confidence Visualization:** Instead of simply showing a binary disease label (e.g., "Effusion: Yes"), the interface displays the probability/confidence score for each of the 13 possible chest conditions. This gives users a clearer understanding of model certainty and helps in assessing borderline cases. The scores are formatted in a user-friendly, color-coded style to increase interpretability.

- **Structured Report Generation:** The interface includes a feature that translates model predictions into a short, structured summary. For instance, if the model detects Pleural Effusion and Cardiomegaly with high confidence, the report might state:
  "This chest X-ray is likely indicative of fluid accumulation and cardiac enlargement. Consider further evaluation for pleural effusion and cardiomegaly."
  This feature bridges the gap between raw model output and clinically meaningful interpretation, emulating the structured format used in real radiology workflows.

Together, these extras demonstrate an effort to go beyond technical success - adding usability, explainability, and clinical framing to a research-focused tool.

3 Design Iteration (LO11 - PrototypeIterate)

Our project's design evolved significantly throughout the capstone lifecycle. Initially, we planned to build a hybrid system using a diffusion model to generate synthetic chest X-rays and a CNN classifier to detect disease signatures. The idea was to explore generative modeling as a tool for augmenting datasets and improving classification robustness - a promising direction we identified during early literature review.

However, as we began developing our first prototype, we encountered major implementation challenges. Diffusion models are computationally expensive, requiring extensive GPU resources and long training times. Additionally, integrating them meaningfully into a medical diagnostic pipeline - especially one focused on classification and reporting - proved too ambitious for our timeline. Our first set of model tests yielded limited results, and the team began to reassess the feasibility of maintaining the diffusion pipeline while meeting our MVP goals.

After discussion with our TA and instructor, and in alignment with the research-first scope of our project, we made the decision to pivot away from image generation and focus fully on building a lightweight, interpretable, multi-label CNN classifier. This pivot allowed us to redirect effort into areas where we could still innovate - particularly on model optimization, class imbalance, confidence calibration, and report generation.

Once we shifted focus, we began iterating on various CNN backbones. We started with ResNet18, then tried DenseNet121, but both models performed poorly on recall, especially with underrepresented conditions. Based on insights from our experimental data, we selected MobileNetV2 as a more suitable alternative. Its lightweight nature aligned with deployment needs in resource-constrained environments and enabled faster training and experimentation.

From there, our design continued to evolve:

- We removed extremely imbalanced or noisy classes (like “No Finding” and “Hernia”) after they were shown to distort model learning.
- We added class-aware rebalancing and margin-based loss functions to address recall on rare diseases.
- We introduced dynamic thresholding, moving away from static 0.5 cutoffs to improve per-class decision boundaries.
- We redesigned the MVP front end to be simplified and focused, ensuring users could upload an image, see predictions, and receive a structured report - without navigating a complex UI.

The final design is the result of multiple iterations driven by technical feasibility, empirical model results, and user needs. We remained aligned with our original research goal - to improve the reliability and accessibility of automated chest X-ray interpretation - while adjusting our path to ensure the outcomes were both realistic and impactful.

4 Design Decisions (LO12)

Throughout the course of this project, our design decisions were influenced heavily by a combination of practical limitations, early assumptions, and project-level constraints. These factors shaped everything from model architecture to user interface features, and they played a critical role in the evolution of both our implementation and research strategy.

Limitations

Computational Resources: One of the earliest limitations we encountered was our limited access to high-performance GPUs. Our original design - which included training a diffusion model for synthetic X-ray generation - was quickly identified as infeasible under our available hardware constraints. Diffusion models require prolonged training and large memory allocation, which wasn’t possible on the machines we had access to. This limitation led us to abandon the generative modeling pipeline and focus instead on optimizing a lightweight CNN.

Data Quality and Balance: We used the NIH ChestX-ray14 dataset, which, while large and publicly available, has significant label imbalance and some noisy annotations due to automated NLP-based label extraction. These issues limited our ability to train a reliable classifier across all disease classes. To manage this, we made the design decision to drop problematic classes like “Hernia” and “No Finding,” and applied techniques like oversampling and margin-based loss to address the imbalance.

No Direct Radiologist Feedback: Although radiologists were identified as key stakeholders, we were unable to conduct formal usability interviews or testing due to privacy, access, and time limitations. As a result, many user-facing design choices - such as confidence score displays and report formatting - were guided by secondary research and informal peer feedback instead of expert interviews.

Assumptions

Sufficient Research Value Without Generation: After pivoting away from the diffusion model, we assumed that the research value of our project would still be significant if we focused on improving classification and explainability. This assumption was validated as we developed and evaluated our lightweight MobileNetV2 pipeline, and confirmed by feedback from our TA and course instructor.

Simplified Interface Meets Stakeholder Needs: We assumed that end-users (radiologists, researchers) would prioritize accessibility and transparency over customization. Based on this, we simplified the MVP’s design, keeping it limited to image upload, output visualization, and structured reporting. We did not implement role-based access or configuration controls, as those were deemed nonessential.

Pretrained Models Provide Sufficient Generalization: We assumed that using pretrained weights from ImageNet for MobileNetV2 would provide enough generalizable features for effective fine-tuning on chest X-rays. This held true across our experiments - pretrained models outperformed models trained from scratch consistently.

Constraints

Timeline and Scope of a Capstone Project: We had to make careful decisions about what could be achieved within the two-semester timeframe. Balancing research rigor with engineering complexity required dropping stretch goals such as free-form NLP-based report generation, multi-dataset integration, and full-stack clinical deployment. We focused instead on a realistic and well-documented MVP with an emphasis on experimentation and evaluation.

Ethical and Legal Constraints: Working with medical data required us to avoid any datasets containing personally identifiable information or real patient data that required ethics approval. This constrained us to public, de-identified datasets and prevented us from exploring federated learning or hospital-specific datasets.

Group Bandwidth and Skill Distribution: Our team had diverse but finite expertise across ML, web development, and research writing. This constraint influenced how we divided tasks and what technologies we used. For example, we chose Flask and React (which team members were already familiar with) instead of learning more complex frameworks that might have slowed us down.

Together, these limitations, assumptions, and constraints helped shape a realistic, research-driven project that produced meaningful insights while staying within the bounds of a senior-level capstone.

5 Economic Considerations (LO23)

Our chest X-ray classification system was designed primarily as a research-driven and educational project. However, it has the potential to evolve into a marketable tool, especially in settings where low-cost, interpretable AI solutions are needed to assist with radiological screening.

Market Potential: The system could serve as a support tool in resource-constrained healthcare environments, such as rural clinics or international outreach settings where access to radiologists is limited. Its lightweight architecture makes it feasible to deploy on modest hardware, further increasing its suitability for these contexts.

Cost to Develop a Product Version: To convert our MVP into a production-grade system, further work would be needed in areas like security, robustness testing, regulatory approval, and UI/UX refinement. These improvements would require collaboration with medical professionals, legal advisors, and possibly clinical institutions. Based on preliminary estimates, this could require a small team and a budget in the range of $100,000-$250,000 CAD for full development, validation, and pilot testing.

Revenue Model: If developed into a commercial product, the system could be offered as:

- A subscription-based web tool licensed to small clinics and diagnostic labs.
- A lightweight software package bundled with portable X-ray machines.
- An open-source tool supported by paid services (e.g., integration, support, or training).

Pricing and Break-Even Considerations: Assuming a subscription model priced at $100/month per clinic, the break-even point could be reached after onboarding 100-200 customers, depending on development and operational costs. Larger-scale adoption would require regulatory compliance and potential integration with PACS (Picture Archiving and Communication Systems).

Non-Profit and Open Source Value: Alternatively, the tool could be released under an open-source license to support global health initiatives. In this case, impact would be measured in adoption, usage, and improvements to clinical workflows rather than revenue.

Overall, while our current system is not yet ready for commercialization, it lays the foundation for a deployable, low-cost diagnostic aid - especially valuable in under-resourced regions or educational settings.

6 Reflection on Project Management (LO24)

This section outlines our reflections on the project management approach taken throughout the capstone. It includes a comparison with our original development plan, identification of successes and challenges, and key lessons learned.

6.1 How Does Your Project Management Compare to Your Development Plan

Overall, we followed the spirit of our Development Plan, although certain details evolved based on team dynamics and project realities. We held weekly team meetings as planned, maintained active communication through Discord, and used GitHub Projects for task tracking and deliverable progress. Roles shifted slightly throughout the semester as priorities changed - for example, one member initially focused on data preprocessing later took on responsibility for documentation and presentation work.

We adhered to our selected technologies: PyTorch for model development, React for the web interface, Flask for serving the model backend, and GitHub for version control. Our workflow plan involving branching, code reviews, and issue tracking was largely maintained, particularly in the latter half of the project.

6.2 What Went Well?

- **Team Communication:** Weekly meetings and daily Discord updates helped us maintain alignment across technical and documentation tasks.
- **Issue Tracking:** Organizing feedback from TAs and peers into GitHub issues gave us a clear backlog and improved accountability.
- **Flexibility in Roles:** Team members were willing to pivot based on workload and needs, which helped address bottlenecks quickly.
- **Documentation Discipline:** By maintaining clean documentation from the beginning, we avoided last-minute rushes near major deadlines.

6.3 What Went Wrong?

- **Early Timeline Underestimation:** Our original scope included advanced tasks like synthetic image generation and complex report generation, which we later dropped due to feasibility concerns.
- **Testing Delays:** Model testing and performance evaluation were postponed until later than ideal, which limited our ability to iterate on early results.
- **Lack of Stakeholder Interviews:** Despite planning to involve radiologists, we did not secure direct feedback from clinical experts due to access constraints.

6.4 What Would You Do Differently Next Time?

If we were to begin this project again, we would:

- Schedule formal checkpoints for model evaluation earlier in the term to catch issues sooner.
- Refine our scope earlier based on feasibility - starting with a simpler system and layering complexity as time allows.
- Attempt to secure expert interviews during the first month to ensure user-facing design decisions are grounded in stakeholder needs.
- Consider modular delegation from the start (e.g., separate backend, frontend, and modeling branches), which would allow for clearer parallel development.

Overall, our project management evolved to meet the demands of a complex research and engineering task, and we adapted well to challenges as they arose.

7 Reflection on Capstone

The capstone experience was a deep dive into both research-driven development and real-world engineering tradeoffs. It allowed us to apply what we've learned across our undergraduate studies while also highlighting where academic preparation ends and practical, self-guided learning begins. We experienced firsthand the challenge of navigating uncertain outcomes, rapidly evolving technical decisions, and the pressure to balance innovation with feasibility.

This project demanded more than just technical output - it pushed us to think critically, justify our decisions in writing, and communicate effectively as a team. Ultimately, it simulated a professional research and development environment, giving us a clearer understanding of what it takes to build something meaningful from scratch.

7.1 Which Courses Were Relevant

Several of our upper-year and core courses directly contributed to the success of this project:

- **Machine Learning 4A:** This course provided the theoretical foundation for training, validating, and evaluating our CNN model. It also introduced key topics like overfitting, loss functions, and performance metrics that were central to our experiments.
- **Software Design 4AA4:** Our module-based system architecture and emphasis on encapsulation and interface design were heavily inspired by design principles learned in this course. Parnas-style decomposition was especially useful during the Module Guide phase.

- **Human-Computer Interaction 4HC3:** We leaned on lessons from this course to create a minimal, goal-driven MVP interface. Understanding usability heuristics helped us prioritize explainability and interpretability in our user-facing features.

- **Databases 3DB3:** While our backend data storage was simple, our familiarity with data schemas and interactions between APIs and databases came directly from this course.

- **Capstone I (SFWRENG 4G06 - Term 1):** The first half of capstone was essential for learning how to scope, research, write formal documents (like the SRS and Design Doc), and organize weekly deliverables.

7.2 Knowledge/Skills Outside of Courses

Despite our strong academic foundation, the project demanded new knowledge and skillsets not directly covered in our coursework:

- **Advanced CNN Techniques:** We had to explore methods like margin-based loss functions, dynamic thresholding, and class-aware rebalancing - topics that weren't deeply explored in our ML course but proved critical for multi-label classification on imbalanced medical datasets.

- **Medical Imaging Context:** Understanding radiological conditions, clinical terminology, and dataset annotation methods (e.g., NLP-based label extraction) required extra reading and domain-specific learning.

- **Scientific Writing:** Writing a full research paper was a new experience for many of us. We had to learn how to structure findings, cite sources correctly, synthesize related work, and communicate results effectively in an academic tone.

- **Cloud and GPU Performance Constraints:** Navigating the limitations of free-tier GPUs and optimizing training performance for our experiments led us to learn more about hardware bottlenecks, runtime logging, and batch processing.

- **Team Documentation Habits:** While documentation was emphasized in classes, this project taught us the importance of maintaining consistent internal and external documentation - especially for shared APIs and preprocessing pipelines.

This project challenged us to bridge the gap between theory and application, and in doing so, it prepared us for both industry roles and academic research paths. It was one of the most demanding and rewarding experiences of our undergraduate career.