

Reflection Report on Software Engineering

Team 17, Team MedTech Visionaries

Allison Cook

Ibrahim Issa

Mohaansh Pranjali

Nathaniel Hu

Tushar Aggarwal

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

1. Addressed missing comments within the SRS document to provide further clarification and context for each requirement.
2. The requirement for accessing x-ray images in the SRS was changed from a user upload to accessing files from the database directly, as it proved to be a safer and more convenient way to get the DICOM x-ray images, given that all the data was to be stored in the medical institution's database already.
3. The requirement for a radiology report generated by an NLP model was moved to a stretch goal, as the scope of the project was updated after Rev 0.
4. Incorporated specific dates for project planning within the SRS, providing a clear timeline for development milestones and deliverables.
5. Worked to ensure complete, correct, and unambiguous requirement definition followed by a rationale and is free of design choices
6. Adjusted the format of the Hazard Analysis document to ensure consistency and clarity in presenting potential risks and mitigations.

1.2 Design and Design Documentation

1. The backend ML model module structure was changed to incorporate an HTTP endpoint to send the diagnosis data to the frontend.
2. An update of the variables used within each module to better reflect the necessary information for the implementation

3. An update of the frontend visual design and required user input to reflect the supervisor's and professor's feedback after Rev 0 to simplify and remove unneeded redundancy in multiple input sections.
4. Updated the documents to have proper semantics for modules' access routines and increased readability with updated definitions for exceptions as per our peer feedback.
5. Ensured proper labeling and documentation of exception-handling mechanisms to facilitate troubleshooting and debugging processes.
6. Provided clear descriptions of the intended uses and functionalities of each module to improve understanding and maintainability of the system.

1.3 VnV Plan and Report

1. We updated the traceability between requirements, the VnV plan, and the report as stated in our peer review and TA comments.
2. We updated the unit tests to include the input and output states, for those that apply.
3. Updated the definitions of unit tests that were done locally and included references to the testing files.
4. Revised the VnV plan and report to use standardized test IDs consistently across all test cases, improving traceability and documentation clarity.
5. Revised the VnV report to reflect changes made to requirements based on the changes made to testing.

2 Design Iteration (LO11)

The initial design had the user upload the DICOM files from the local system, which was inefficient and defeated the purpose of having a web app for the purpose of being separate from the OS and the local machine. Later, the remote cloud database was leveraged and as a result, xray files were accessed directly from the database. Reading the files as bytes helped against storing anything as files in the intermediate stage, and so was optimal for having the whole process done without using the local file system.

Throughout the development process, our team iteratively refined the design of our project, X-RayAssist, to ensure it met the evolving needs and requirements. Initially, we started with a basic architecture that focused primarily on automated analysis of chest X-rays for disease detection. We actively sought feedback from stakeholders, including TAs, instructors, and potential users, which helped us identify areas for improvement and refinement in our design. This feedback loop enabled us to address usability issues, prioritize features,

and make necessary adjustments. We developed an initial prototype of the system to demonstrate key features and functionality. This prototype served as a proof of concept and enabled us to validate our design decisions in a practical context. Through iterative cycles of testing and refinement, we iteratively improved the design based on user feedback, usability testing results, and technical evaluations. This iterative approach allowed us to address issues and incorporate enhancements incrementally. After several iterations and refinements, we finalized the design by conducting thorough testing, validation, and documentation. The final design reflected a balance between user needs, technical considerations, and project requirements, culminating in a robust and user-friendly system architecture. Collaboration among team members played a crucial role in driving design iteration. Regular meetings, brainstorming sessions, and peer reviews enabled us to share ideas, identify design flaws, and collectively work towards optimal solutions. Overall, our design evolved iteratively in response to feedback, technological advancements, user needs, and collaborative efforts. Each iteration brought us closer to our final design, which represents a robust, user-friendly, and innovative solution for automated chest X-ray analysis.

3 Design Decisions (LO12)

1. The decision to remove the need for users to upload x-ray images was to make it efficient to access the patient's files, as the requirements stated that a database of the patient's files was already set up by the medical institution, storing all the DICOM files.
2. The decision to send encoded images to the frontend after processing by the ML model (raw xrays and heatmaps) was effective in terms of being implemented as a simple HTTP response and also promoted separation of concerns for the backend since it did not have to deal with the other patient data like personal details.

Throughout the design process of our project, various decisions were made based on careful consideration of limitations, assumptions, and constraints. We chose specific technologies based on factors such as compatibility with project requirements, team expertise, and availability of resources. For example, we opted for React.js for the frontend due to its flexibility in building interactive user interfaces, while Flask was chosen for the backend for its simplicity and scalability. To ensure maintainability and scalability, we modularized our system architecture into distinct components with well-defined responsibilities. This modular approach allowed for easier development, debugging, and future enhancements. Design decisions regarding the user interface were influenced by usability considerations and user feedback. We aimed for a clean and intuitive interface to facilitate user interaction and minimize the learning curve. Assumptions about user preferences and behavior guided the layout, navigation flow, and visual elements of the interface. Assumptions about potential security threats led to the implementation of security measures such as authentication, authorization, and

data encryption. These decisions were crucial for safeguarding sensitive patient information and ensuring compliance with privacy regulations. In summary, our design decisions were guided by a combination of project requirements, technical considerations, user feedback, and constraints. By carefully evaluating trade-offs and considering the implications of each decision, we aimed to develop a robust, user-friendly, and scalable system that meets the needs of both healthcare professionals and patients.

4 Economic Considerations (LO23)

There is a significant market for automated chest X-ray analysis solutions, driven by the increasing volume of medical imaging procedures worldwide. The market includes healthcare institutions, clinics, and diagnostic centers that require efficient and accurate interpretation of chest X-ray images in order to reduce potential customer wait times. Marketing strategies would involve targeting healthcare institutions and medical facilities through direct sales and partnerships with equipment distributors. Also, participating in industry and conference exhibitions to showcase the capabilities of the service. In terms of the associated cost, it would involve expenses related to software development, including AI model training, user interface design, backend infrastructure, quality assurance, and ongoing maintenance and support. Pricing models for this product/service could include one-time licensing fees, subscription-based plans, or pay-per-use options, depending on the number of customers.

5 Reflection on Project Management (LO24)

5.1 How Does Your Project Management Compare to Your Development Plan

When we compared our actual project management to our development plan, we found that they were fairly similar, with the biggest differences being in our team meetings.

Our planned team meetings were the set lecture and tutorial as well as a weekly meeting with our project supervisor. Unfortunately, we rarely meet during the lecture and tutorial time unless needed for an informal review; we did have 2-3 weekly meetings in the evenings for most of the fall semester, but as we shifted to implementation, we often had only 1 weekly meeting to check-in. Additionally, while we followed the set meetings with our supervisor for most of the fall semester, our weekly meetings moved to be bi-weekly/as-needed during the winter semester.

Initially, roles and responsibilities were clearly defined within the team, and each member contributed according to their assigned tasks. However, as the project progressed, there were instances where workload distribution became uneven, leading to bottlenecks and delays. In future projects, we will regularly reassess team dynamics and workload distribution to ensure a balanced workflow. For

team communication, we actively used our MS Teams as the main form of communication as planned. Additionally, all other communication rules and guidelines were followed.

The team member roles were set out as all developers, which was followed in our actual project. Additionally, Tushar completed all the assigned roles of administrator, project manager, and scrum master as needed.

In terms of workflow, most were followed, with the part of a branch 'passing all automated test cases before merging' ignored in some cases.

5.2 What Went Well?

The team had good planning in terms of repository management. Rules to protect secrets regarding API keys were well defined and issue tracking was efficient and helped the team keep up with reviews and bug fixes. Adopting agile principles enabled us to adapt to changing requirements and prioritize tasks effectively. We utilized tools like Kanban boards and task management software to track progress, identify bottlenecks, and allocate resources efficiently. Leveraging version control systems such as Git and platforms like GitHub provided a centralized repository for our codebase and documentation. This facilitated collaboration among team members, allowed for seamless integration of changes, and ensured code integrity throughout the development process. Implementing CI/CD pipelines automated the testing, building, and deployment processes, reducing manual errors and accelerating the delivery of features. This approach enabled us to maintain a stable codebase, iterate rapidly, and respond promptly to feedback. Soliciting feedback from TAs, instructors, peers, and potential users allowed us to gather valuable insights and identify areas for improvement early on. We incorporated feedback iteratively, ensuring that the final product met user expectations and quality standards.

5.3 What Went Wrong?

We started off very strong and our plans were well developed, there were very minimal issues with project management. For our technology, we ran into issues accessing the medical databases to conduct, and we only managed to gain access after our Rev 0 demo. Additionally, some of the automatic testing and code standards checks were difficult for some members to use, so there were consistent errors. We encountered various technical issues throughout the development process, including compatibility issues between different software components, bugs in the codebase, and challenges in integrating third-party libraries or APIs. These issues sometimes caused setbacks and required additional time and effort to resolve.

5.4 What Would you Do Differently Next Time?

Next time, we would aim to work even more ahead of the deadline as it feels like we've been settling for results when there is room for improvement. Addition-

ally, we were a bit lax in our updates and kept constant communication with each other during some periods of the project; we feel like with stronger communication and continuous teamwork, we would have been able to bring forth an even better project. Although these are some changes we want to make, there are many things we would repeat in future projects as we feel like they made taking on this project possible.