Design and develop training materials and plan end user training following software deployment, include consideration of human and technical security requirements. Review common pitfalls and mitigations in common quality assurance techniques. Critically evaluate the properties of good software design including the nature and the role of associated documentation, appreciating that design increasingly covers the use of existing code and 3rd party elements that may be an alternative to development of code from scratch through make/buy decisions. Assess these properties in a security context and impact on the threat model.

I created a step-by-step end user guide to support user onboarding and system usage. (Kloda, 2025a) I also included security tips for end user and administrators to reinforce good practices. (Kloda, 2025b) As part of our internal knowledge transfer and user training process, I conducted onboarding sessions with new team members. (Kloda, 2025c; Kloda, 2025d; Kloda, 2025e) During the sessions, I explained the overall scope and goals of our project, outlined the current development stage, and showed the wireframes and user interface design. This walkthrough allowed better understanding of the project vision and enabled contributions.

As part of our design process, I created a detailed make/buy analysis to evaluate the benefits and risks of using external solutions. This analysis evaluates our decisions and helps ensure that security considerations are prioritized in our project. (Kloda, 2025f)

To enhance our approach to software quality assurance, I developed a document identifying common pitfalls and mitigation strategies. (Kloda, 2025g) This supports proactive risk management throughout the development lifecycle.

Finally, to evaluate and apply appropriate software design and development methodologies, I combined structured design and object-oriented design (OOD) principles. I also and used standard notations such as UML class diagrams and flowcharts to model data flows and system architecture, promoting clarity and modularity in our system design. (Kloda, 2025h)

# References

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/FinalSprintRecording.mp4 [Accessed 16 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/LargeSoftwareChallenge/SecurityCheatSheet.docx [Accessed 16 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/Onboarding%26PlanningSession.png [Accessed 16 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/Onboarding\_Iain.png [Accessed 16 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/OnboardingAntonio.png [Accessed 16 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/LargeSoftwareChallenge/MakeBuyConsideration\_SecurityImpact.xlsx [Accessed 16 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/movie\_recommendation\_system/blob/main/team\_documents/Quality\_assurance\_Pitfalls.pdf [Accessed 3 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Realising%20a%20minimum%20valiable%20product/SoftwareDesignDevelopmentMethodolodgy.docx [Accessed 6 May 2025].