SYSC 4907 – Fall 2026/Winter 2027

Supervisor	Jason Jaskolka
Co-supervisor	
Course section	SYSC 4907 H
Project ID	H1
Project title	VESSEL: Versatile Environment for Secure Systems Engineering Lifecycles
Project description	Are you ready to redefine how security is integrated into software engineering? This project challenges you to design and prototype the core of a cutting-edge platform that operationalizes the NIST Systems Security Engineering Framework. The NIST Systems Security Engineering Framework provides a disciplined, structured, and standards-based approach to engineering trustworthy secure systems, emphasizing a holistic integration of security throughout all stages of the system life cycle. The framework is built around three interrelated contexts (problem, solution, and trustworthiness) which guide the identification of stakeholder security objectives, the development of effective security solutions, and the demonstration of system trustworthiness through evidence and assurance. This approach ensures that security is not an afterthought but a foundational element of every system, guiding engineers to address threats, validate solutions, and prove trustworthiness through evidence-based assurance. This *multi-year project* aims to create a new, open, and extensible platform (known as *VESSEL: Versatile Environment for Secure Systems Engineering Lifecycles*) that operationalizes the NIST Systems Security Engineering Framework, meaning to turn its principles and processes into practical, usable software tools that actively support secure system design and engineering in real-world projects. This involves: • *Problem Context*: Helping users clearly define and understand security problems and objectives, such as identifying what needs to be protected, what threats exist, and what the critical assets are. The platform will support analyses that clarify stakeholder needs and protection requirements, setting the foundation for all subsequent design decisions. • *Solution Context*: Enabling users to architect, design, and implement security solutions that directly address the identified problems and requirements. The platform will provide tools for modeling system architecture, allocating security requirements, and designing security controls—while also capturing evidence (like test results or design reviews) to demonstrate that the solution meets its security objectives. • *Trustworthiness Context*: Supporting the creation and maintenance of structured assurance cases—clear, evidence-based arguments that the system is trustworthy and meets its security claims. The platform will help users gather, organize, and

present evidence to show that the system is secure, resilient, and meets stakeholder expectations, even as the system evolves.

By embedding these three contexts into the platform's workflows, the project aims to make security a continuous, iterative, and visible part of system engineering. The platform is envisioned to guide users through the necessary security analyses at each stage (such as risk analysis, requirements evaluation, vulnerability assessment, and assurance case development), helping engineering teams and stakeholders make informed, confident decisions about system security.

Project Objectives:

Phase 1 of this multi-year project aims to:

1. Establish a clear alignment between software development lifecycle stages (requirements, design, implementation, testing) and the NIST Systems Security Engineering Framework three core contexts: problem (security objectives), solution (secure design), and trustworthiness (evidence-based assurance).
2. Design and prototype the core architecture of the VESSEL platform, focussing on creating a modular, extensible architecture with clearly defined components that allow users to define custom threat and system representations, ensuring adaptability to diverse project needs.
3. Conduct experiments to validate critical design decisions to ensure the platform is both powerful and practical.
4. Build a foundation for future teams by delivering clear documentation, a well-structured codebase, and a roadmap for ongoing development and future enhancements, including AI/ML module integration and advanced user assistance features.

Project Expectations:

This project is open to students enrolled in the Software Engineering program.

1. **Application of Software Engineering Principles:** You are required to apply established software engineering methodologies from the initial requirements gathering and analysis, through system architecture and detailed design, to implementation, testing, verification, validation, and maintenance. The project should reflect a systematic and disciplined approach to software development.
2. **Consideration of Constraints:** Your solution must demonstrate thoughtful consideration of relevant constraints, which may include health and safety, environmental impact, societal needs, adherence to standards, and other interdisciplinary factors pertinent to your project context.
3. **Comprehensive Documentation:** You are expected to produce and maintain clear, comprehensive, and well-organized documentation for your software system. This includes, but is not limited to, requirements specifications, design documents, test plans, user manuals, and maintenance guides. The project codebase must be maintained in a structured, version-controlled repository with meaningful commit messages, clear directory organization, and thorough inline code comments. All documentation and code should be prepared with future

	development in mind, enabling subsequent teams to readily understand, maintain, and extend the platform without reliance on external sources or prior team members. Regular updates to both documentation and the repository are expected throughout the project to ensure accuracy and continuity for future contributors. 4. **Demonstration of Artifacts:** Throughout the project, you will be required to present various software artifacts during team meetings. These may include prototypes, design sketches, test harnesses, infrastructure components, and code repositories. Regular demonstrations will be used to assess progress and provide feedback. Successful completion of this capstone project will require not only technical proficiency but also the ability to integrate engineering principles with real-world constraints, communicate effectively through documentation, and engage in collaborative development practices. These expectations align with professional standards in the software engineering discipline. **References:** References related to the NIST Systems Security Engineering Framework will be provided. **Prerequisites:** Enrolment in Software Engineering; proficiency in software design and development; familiarity with the software development lifecycle (SDLC); basic knowledge of system security concepts; experience with version control systems (e.g., Git); teamwork and communication skills; analytical and problem-solving skills.
Program(s)	Software
Maximum number of students	5
Meeting time with supervisor (optional)	Students undertaking this project must be available to meet to discuss project progress, address challenges, and collaborate effectively with team members and the project supervisor. Regular meetings are expected to be primarily in person during regular working hours (Monday-Friday 9:00AM-4:00PM) held every 1-2 weeks for 30-60 minutes.
Do you want the student to contact you before the office assign this project to them ? (Yes/No)	Yes