INFO 5100 Spring 2025 Final Project
Prof Bugrara, K
Due Date: Apr 20 2025 at 11:59PM
Team: 3 members, if 4 members then you must demonstrate 25% increase in functionality and features.

Exploring the implementation of a communication ecosystem for multi-party applications

The objective of the final project is for you to explore and the design and implementation of software systems for complex problems. You will practice our building block techniques to program complex software. You will be play multiple roles in this project to build good understanding of what it means to experience the development and management of information systems. Such roles include: a functional architect (analyst) understanding the problem and designing the use-cases, and user processes, a designer where you develop the object model and user-interaction protocols, and a programmer where you do the java development, an innovator and communicator of new ideas where you find creative things that no one else thought about before. And you present your ideas clearly and completely.

The project focus will be on the construction and operations of a digital platform in some application domain. The software will serve as a communication system for specialists and operational personnel. The entities will be enterprises, organizations, departments, administrators, supplying companies, and most importantly individual persons who are users and subject of the work. Such connectivity opens the possibility for many useful uses that are critical to the safety and well-being of people, effectiveness in delivering quality service, and efficiencies to ensure that the services are affordable. There are a number of uses of such an eco-system.

Your job is to select a multi-party problem that will require the cooperative effort of multiple enterprises and associated organizational units cross national boundaries. The objective is to design and implement a system where **the whole is greater than the sum of the parts.** In other words, the collaborative effort

of the partnership leads to greater value and benefit than the individual entities operating on their own.

Deliverables

1) A running swing application that addresses the challenges outlined above at an eco-system scale.
2) A presentation that outlines details specific to the design and implementation of your solution. This must include: the definition of the problem you solved, stakeholder and their contributions, use-cases, design, and implementation techniques followed. You must explain why each enterprise is essential to delivering the total value as outlined in your problem statement.
3) A robust role-based authentication module with strong username and password capabilities.
4) A reporting module with summarized view of the data in your system. This could include performance data that is important at the system or network level.
5) A configuration module with test cases that will populate your model and show the correctness of your solution. You might want to integrate with a Faker module from faker.com for random data generation.

You are required to defend your solution through a 30 minutes in-person presentation. This examination will include detailed review of your code and validity of your engineering techniques.

This is a team project with up to 3-4 participants. However, each participant must be able to answer any questions in relation to the design and implementation of the project.

Implementation Criteria

- The project must be able to demonstrate the learning of ecosystem models
- For teams of 3:
    - 1 network
    - 4 enterprises
    - 6 organizations
    - 8 unique roles (Excluding system/enterprise/organization/network admins)
    - 4+ work requests, with:
        - At least 1 cross-organization request
        - At least 1 cross-enterprise request
- For teams of 4:
    - 1 network
    - 5 enterprises
    - 8 organizations
    - 10 unique roles (Excluding system/enterprise/organization/network admins)
    - 5+ work requests, with same cross-entity requirements as above
- Work Area Management for each role
- UI/UX design and Alignment
- CRUD operations – largely used wherever necessary mocking a real-world application
- Form Validations – email, name, age, etc.
- Unique ID
- Robust Class Design Model
- Relevant data types
- Status management of different requests
- Pre-populated data for analytics and better demo of project.
    - Use of Faker module is required https://github.com/DiUS/java-faker
- A robust role-based authentication module with strong username and password capabilities.
- A reporting module with summarized view of the data in your system.

    - This could include performance data that is important at the system or network level
- System Admin CRUD
- Enterprise Admin CRUD
- Error handling and NULL checks. No errors on the output.
- Advanced features, such as mail, AI, maps, SMS integration, APIs, etc., are plus

Use of database is optional; you can use any databases – object storage/structured.

Presentation & Documentations

- A 5-minute presentation slide including:
    - Problem Statement
    - UML Class Diagram indicating attributes, method, relationships.
    - Brief advanced feature introduction
- 20 minutes demonstration of Swing application
    - Must demonstrate individual contributions to the project
- High-level Component Diagram
- UML Class diagram including attributes, methods, and relationships

Grading Breakdown

Component	Weight
Implementation & Functionality	70%
Presentation & Documentation	20%
Q&A (Individual Understanding)	10%

Presentation Structure (30 mins)

- **5 minutes** – Introduction & Presentation Slides
- **20 minutes** – Live Swing Application Demo
- **10 minutes** – Q&A (all team members must be able to explain any part of the project)

Documents need to be submitted on canvas (do not zip):

- Project Proposal in .doc, .docx, or .pdf
- Presentation slides in .ppt, .pptx, or .pdf
- High-level component diagram in .pdf
- UML class diagram in .pdf

Documents need to be pushed to GitHub:

- Project Proposal in .doc, .docx, or .pdf
- Presentation slides in .ppt, .pptx, or .pdf
- High-level component diagram in .pdf
- UML class diagram in .pdf
- A complete final project folder