Final Project Description

Course Title: CS504070 – Service-Oriented Architecture

Instructor: Phuc H. Duong

Location: Tan Phong Campus

Course Objectives Alignment

The Final Project is designed to ensure that you understand, practice, and implement information systems by following the Software Development Life Cycle (SDLC) model. This project emphasizes the analysis, design, and implementation stages, providing you with hands-on experience in developing an information system.

Project Overview

1. Topic Selection and Initial Research

- **Objective:** You must select a project topic relevant to enterprise systems. Potential topics include e-commerce platforms, inventory management systems, customer relationship management (CRM) systems, or any other system involving complex information processing.
- **Deliverables:** A one-page project proposal that outlines your chosen topic, its relevance, and the scope of the system to be developed.

2. Analysis and Design

You will perform a comprehensive analysis and design for your chosen system. The following outputs are required:

- **Use-case Diagram:** Illustrate the main functions of the system and interactions between users and the system.
- **Data Flow Diagram (DFD):** Show the flow of information within the system, capturing processes, data stores, data flow, and external entities.
- **Entity-Relationship Diagram (ERD):** Define the data structure of the system, identifying entities, relationships, and attributes.

- **Other UML Diagrams:** Depending on the complexity of the system, you may include Class Diagrams, Sequence Diagrams, or Activity Diagrams to better capture the system's design.

3. Implementation

- **Objective:** Implement the project based on the designs you created during the analysis phase. You should focus on developing 2-3 core functions of your system, demonstrating an understanding of key software development principles.
- **Mandatory Basic Functions:**
  - User authentication (login, logout, registration)
  - Error handling
  - Basic navigation
- **Core Functions Examples:**
  - For an e-commerce website:
    - Purchasing process (e.g., product selection, checkout, payment)
    - Selling process (e.g., product listing, inventory management)
    - Package tracking and logistics (e.g., order tracking, delivery updates)
- **Deliverables:** A functional demo program or web-app that showcases the implemented features. The demo should be easy to navigate and clearly demonstrate the functionality of both basic and core functions.

Project Deliverables

1. Detailed Report (Minimum 40 Pages):

- **Format:** A4, double-spaced, 12-point font, Times New Roman.
- **Contents:**
  - Project introduction, objectives, and scope.
  - Analysis and design documentation (including all diagrams).
  - Implementation details, including explanations of chosen technologies, challenges faced, and how they were overcome.
  - Differentiation between basic functions and core functions.
  - Conclusion and potential future enhancements.

2. Functional Demo Program:

- A working application that implements the designed system.

- A video demonstration (optional but encouraged) showing the system in use, highlighting core functionalities.

Tentative Grading Rubric

The Fall, 2025 Final Project will account for 70% of your final grade and consists of two components: Project Evaluation (20%) and the Final Project itself (50%). A tentative grading rubric for both components is provided below.

Project Proposal ***(10%)***	- Project description - Clarity and relevance of the topic - Feasibility and scope
Analysis and Design ***(40%)***	- Quality and correctness of Use-case Diagram (10%) - Quality and correctness of DFD (10%) - Quality and correctness of ERD (10%) - Additional UML diagrams (10%)
Implementation ***(40%)***	- Functionality and correctness of basic functions (15%) - Functionality and correctness of core functions (20%) - API Docs (5%)
Report ***(10%)***	- Structure, clarity, and comprehensiveness. - Technical accuracy and depth of analysis. - Proper differentiation between basic and core functions.

Project Scope and Timeline

- Week 1-2: Topic selection and proposal submission.
- Week 3-5: Analysis and design phase. Diagrams and documentation.
- Week 6-9: Implementation phase. Coding and integration.
- Week 10: Final report writing and preparation of demo program.

Guidelines and Technologies

- **Technologies:** You may select the programming language and development framework most familiar to you and best suited for your project. Popular options include Java, Python, C#, .NET, Django, and React.
- **Version Control:** Use Git or another version control system to track progress and submit code.
- **Collaboration:** You can work on the project individually or in groups of up to three students. Group projects should clearly define each member's contribution.
- **Plagiarism:** All work must be original. Plagiarism will result in severe penalties, including potential course failure.