SEC 205: Distributed Ledger and Blockchain Assessment Instruction

General Information

Competency Code: SEC-205
Competency Title: Distributed Ledger and Blockchain
Competency Semester: Fall 2024
Instructor Name: Charnon Pattiyanon, Ph.D. (charnon@cmkl.ac.th)

Assessment Overview

The SEC-205 Distributed Ledger and Blockchain competency aims to introduce students to the decentralized technologies, including blockchain, ethereum and smart contract development, and web 3.0 applications. These advanced technologies show significant benefits to the world in many real-world applications, like Decentralized Finance (DeFi), Decentralized Identities, and Non-Fungible Tokens (NFTs). In this competency, you will learn to develop smart contracts with Solidity programming language on an Ethereum virtual machine.

In this assessment, students will have the opportunity to apply their knowledge and demonstrate their skills in the Solidity development by establishing a Web3.0 development project from real-world use cases. Students will be asked to develop smart contracts to perform operations of the application on blockchain.

Assessing Skills

- **[SEC-205:00010]** Successful students will be able to learn and understand how blockchain technology works in real world use cases.
- **[SEC-205:00020]** Successful students will be able to design and develop a web 3.0 applications that use the blockchain technology as an underlying mechanism
- **[SEC-205:00030]** Successful students will be able to analyze and criticize blockchain privacy and security comprehensively.

Pre-cautions

- Please express your answers to each required outcome based on your own ideas and perspective. Plagiarism is unacceptable. If I find that either content or ideas are remarkably similar between two or more students without any sound reasons, scores of all students will be deducted as a consequence of illegal actions.
- Students are expected to demonstrate a deep understanding of the subject matter through critical analysis and original insights. Overreliance on AI-generated content without providing substantial original thought will negatively impact the assessment score.
- Justifications are critical explanations of why your answers or outcomes are expressed the way they are. They should be written in a "why" style. For example, "I believe that this privacy principle is possessed by the target system because . . . ." There will not be a one-size-fits-all solution or criticism for writing a justification. Your analysis skills will be evaluated through the clarity of your justifications.
- I encourage students to ask me any questions to address their curiosity about the assessment via email or the discussion page on Canvas. However, please do not submit your assessment report and ask for my feedback on it. I will consider it as a report submission.
- This assessment will provide some optional outcomes. They are not required to be included in the report. They will not affect the score of this assessment. However, they will be considered in cases where you receive a borderline score between two mastery levels or a low score on this assessment. The optional outcomes will not exceed 10% of the overall score, depending on the instructor's discretion.

Assessment Instruction

The total score for this assessment is **300 points**, with each assessing skill contributing 100 points. Please carefully follow the instruction below:

- **Task 1:** *(100 Points)* Select *one* web 3.0 application from the following list: (1) E-Commerce Application, (2) Asset/Inventory Management, and (3) Financial/Banking Applications, but does not limited to. You can propose to develop any other applications that you are interested in. Then, specify system requirements and design an architecture, data structures, use case scenarios, and smart contract structure of the selected web 3.0 application. Finally, write a design specification report following the template.
- **Task 2:** *(80 Points)* Develop a Solidity project to implement all required smart contracts for handling transactions on blockchain. These smart contracts must cover all use case scenarios specified in the design specification. Write test cases to show the expected and actual outcomes. (Optional) Develop a front-end application that could connect with the blockchain. Then, deploy on a local test network using Hardhat.
- **Task 3:** *(20 Points)* Record a video file in mp4, mkv, or mov format (no more than 5 minutes), describing your Solidity source code and showing the results of each smart contract based on the use case scenarios. Then, save the video file and upload it on a cloud storage.
- **Task 4:** *(100 Points)* Analyze the security and privacy protection of the implemented smart contracts. Write a report to make suggestions on how to secure the implemented application and preserve privacy of user personal information.

Please write a report using the following **template** and save it as **a single PDF file**.

SEC-205 Distributed Ledger and Blockchain
Assessment Report

Name: [Your full name] ([Your nickname])
Email: [Your CMKL email address]

Application Name: [Name of the selected application]

System Description:
[Describe your selected application]

At least answer the following questions:
(1) What are the purposes and objectives of this applications?
(2) Who will be the user of this application?
(3) Does this application fit with the blockchain technology? Why?
(4) Why do you select this application?

System Architecture:
[Put your system architecture figure here]

Contextual or deployment diagrams should be used to describe the system architecture. At a minimum, the architecture should illustrate how many components the application has and how these components interact with each other.

System Requirements:
Functional Requirements:

- [Functional Requirements 1]
- ...

Non-functional Requirements:

- [Non-functional Requirements 1]
- [Non-functional Requirements 2]
- ...

Other Requirements:

- ...

Data Structure:
[Put your data structure design here]

A class diagram or Entity-Relation (ER) diagram can be used to describe the data structure in the application. *Hint:* Please be aware that data in the application's database and on the blockchain are different. The design of the data structure should consider the different purposes of storages and the sensitivity of the data.

Use Case Scenarios:
[Put your use case scenarios here]

The following table can be used as a template to define each use case scenario.

Use Case ID	Ex. UC-001
Use Case Name	[Describe the name of the use case scenario e.g., create a bank account]
Use Case Description	[Describe the use case scenario in detail]
Stakeholders	– Stakeholder 1 – Stakeholder 2 – ...
Steps/Procedures	1. ... 1a. ... 1b. ... 2. ... 3. ...

Smart Contract List:

- [Smart Contract 1 Name] : [Describe Smart Contract 1 in detail, explaining the purposes of the smart contract, as well as its inputs and outputs]
- [Smart Contract 2 Name] : [Describe Smart Contract 2 in detail, explaining the purposes of the smart contract, as well as its inputs and outputs.]
- ...

Link to the Video File: [Link to the cloud storage of the video]

Security Analysis:
[Put your analysis report on the security of the application]

At least answer the following questions:
(1) Do you think this application is secure in terms of data protection?
(2) Are personal data objects published on the blockchain?
(3) Are there any security protections implemented in the application?
- If so, what are they and how do they protect? Do they protect the application at every layer?
- If not, what security risks could threaten the application? How would you suggest improving the application to protect data securely?

Privacy Analysis:
[Put your analysis report on the privacy preservation of personal data within the application]

At least answer the following questions:
(1) Are there any personal data or information used in the application?
(2) How can the application ensure anonymity, conditional privacy, and selective disclosure? If the application cannot ensure these, how would you suggest improving it?
(3) There are many techniques to preserve privacy. Which techniques are already included in the application, and which are not? Why? Use the following table to answer this question.

Techniques	**Included?**	**Justifications**
Zero-Knowledge Proof	No	This technique is not included because . . .
. . .	. . .	. . .