Project

IEMS 5725 Blockchain and Applications
(Fall, 2024-25)
Full Mark: 100

Deadline: 23:59 HKT
14 December 2024

As an IT engineer, you are tasked with building a smart-contract-based retail transaction system that allows users to create profiles, purchase products, and manage transactions on the Ethereum blockchain.
In your code, you can assume all users are honest.
The mark will be composed of 75% for programming (with 5% for general quality, *e.g.*, code readability) and 25% for the report. The bonus will be added in a way according to the general class performance.

1 Deliverables (25 Points)

1. The smart contract code (with comments): **[studentID-codes].txt**,
2. A documentation: **[studentID-documentation].pdf** describing how to run and use your smart contract.

   If you think the comments you left on your codes are enough, you do not need to provide documentation. If you do, please **make it concise and no more than 2 pages.**
3. A report: **[studentID-report].pdf** discussing the following (**be concise and no more than 3 pages**):

   (a) advantages of using blockchain in retail transactions (10 points);

   (b) (at least) three problems of the current project and give some potential solutions (15 points).
   - There is no limitation on the problem types. You can think about malicious users and explore things from any perspective, such as code bugs, system/management, security, privacy, *etc*. (For code bugs, it is recommended to use analysis tools mentioned in Ch.8B.)
   - If you believe your code and design are “perfect”, explain clearly the (three) special mechanisms you introduced.

2 Basic Requirements (75 Points)

The smart contract is supposed to achieve the following features (10 points for each).

1. Buyer Registration: A user can register by generating the profile with their name, email, and shipping address. Only one profile can be registered per address. Once registered, the buyer can view and update/edit their own profile information.
2. Seller Registration: Only **one** address can register as the seller by depositing a certain amount of cryptocurrency (Ethereum coins), and the seller address is not allowed to be a buyer. Once registered, the seller can add products for sale with a name, price, URL, and inventory (A product ID will be generated). Only the seller address is allowed to add and update/edit product information.
3. Shopping Moment: Anyone can input the product id and view the corresponding product information, including the name, price, URL, and inventory. Everyone has a wish list and can add the product id to his/her own list.
4. Transaction Initiation: A buyer can initiate a new transaction by specifying the product ID and quantity. The total cost of the transaction is calculated based on the selected product price. The buyer should hold enough money and transfer it to the smart contract to proceed with the transaction.
5. Transaction Information: A user can only view their own transactions, while the seller can view all. (Note that you need to consider what attributes are required in order to achieve the requirements, as they are not explicitly specified here.)
6. Return Request: A buyer can request a return before completing the transaction. The seller can get the transaction information to see the transaction status and approve a return upon request. Once a return is approved, the money contained in the transaction should be transferred back to the buyer.
7. Transaction Completion: The buyer can mark a transaction as completed. Once a transaction is completed, the total cost of the transaction should be transferred to the seller’s account. Also, no other modifications or actions can be done except viewing the transaction information.

3 Bonus Feature (extra 10 Points)

A full bonus will be given if you complete the following two requirements.

1. Trading with ERC20 Token: Instead of buying items with Ethers directly, your smart contract now requires buyers and sellers to trade with ERC20 tokens instantiated in your contract. Your smart contract should create a new ERC20 token and allow users to exchange between your token and ethers with a fixed rate.
2. Seller Penalty: Design a penalty mechanism for the seller. If certain condition(s) are(is) triggered, the seller will be punished by deducting a certain deposit. (Briefly explain how your mechanism works by commenting.)

4 General Guidelines

1. Develop the smart contract using Solidity.
2. Leave necessary and concise comments on the codes **(important!)**
3. Use Remix to compile, deploy, and test the smart contract. You may use other development environments, e.g., Hardhat.

5 Optional: Other Individual Topics (Subject to Approval)

Other topics must be approved by the teaching team and share **the same deadline**.

If you have your own ideas about the project (e.g., those outlined in Ch.8B), please email us with a formal proposal by **12 April** (this Friday) and expect a reply within 2 working days.

6 (Electronic) Submission

Deadline: 23:59 HKT, 14 December 2024.
Penalty will be applied to late submission. 0 mark for submission after 20 December.

Please compress the three files into **[studentID].zip** and submit them to Blackboard before the deadline.

If you have any questions, please raise them on Blackboard, Piazza, or email us.