Programming Project: Smart Contract in Retail

— IEMS 5725 Blockchain and Applications

Deadline: 23 April 2023

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 named as a **[studentID-codes].txt** file.
2. A documentation named as a **[studentID-documentation].pdf** file, which describes 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** on the following discussions (**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 gives some potential solutions (15 points).
   There is no limitation on the problem types, which means that you can think about malicious users and explore things from any perspective, such as system/management, security, privacy, *etc*.
   (If you believe your code is "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 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, and inventory. Only the seller address is allowed to add product information.
3. Product Information: Anyone can input the product id and view the corresponding product information, including the name, price, and inventory.
4. Transaction Initiation: A user 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 his/her own transactions, while the seller can view all. (Note that you need to think about what attributes are needed 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 15 Points)

A full bonus will be given if you complete **any three** of the following.

1. Profile Update: Provide a way that enables a buyer to update/edit their own profile information.
2. Product Review: Provide a way that enables the buyer to rate the purchased product by a numeric scale of 0-5 (5/0 means very satisfied/not satisfied at all).
3. Multi-Product Purchase: Provide a way that enables a buyer to buy a list of products in one transaction.
4. 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.

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.

5 (Electronic) Submission

Deadline: 23 April 2023.

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

If you have any questions, feel free to raise them on Blackboard, Piazza, or email us.