CSCI 6313 Introduction to Blockchains

Assignment 1- part B

This is the second and a major part of your assignment. Your task is to create a smart contract as per below specifications. The smart contract is for a test Ethereum blockchain, such as Ropsten or Rinkeby. We recommend Rinkeby as you can acquire ETH easier. You also need to create a Dapp (Distribute Application that uses blockchain smart contracts), wherein the Dapp is a relatively simple skeleton of a part of a larger application.

Consider the following simple scenario involving three actors/participants. Actors involved include vendor, buyer, and a notary. Buyer and seller negotiate an agreement, e.g., for a sale of a large and expensive piece of equipment. They agreed on price and terms of delivery that were captured in a purchase agreement, which is submitted to the notary. Each of the buyer and seller now needs to approve the document. To simplify, the smart contract needs to support the following:

- The smart contract if first created and given to the notary. The notary then invokes your smart contract method to store the purchase agreement document, which is passed as a string. The method calculates the hash code of the agreement and stores it and the agreement on the blockchain. The notary then notifies the buyer and seller that a contract is ready for approval.
- Each of the buyer and a seller then:
    - Retrieves the hash code and the agreement from the smart contract.
    - It calculates the hash code of the retrieved document and compares it to the hash code retrieved from the smart contract. If the hash codes do not match, it (buyer or seller) indicates cancellation of the agreement by calling a smart contract method. If the hash codes match, then the actor reviews the contract and calls a smart contract method to either approve or cancel the agreement.
      (Note that, as the agreement is stored on the blockchain, its hash code is not required now. However, it will be required in the following assignments.)

Smart contract methods

- *submitAgreement* … invoked by the notary to submit the agreement (agreement is represented by a string)
- *retrieveAgreement* … invoked by either the seller or buyer to retrieve the agreement and its hash code
- *approveAgreement* … invoked by either the seller or buyer to approve the agreement

Additional Requirements

- Naming Conventions for your scripts/code
    - Identifiers that you create in your software (contract methods or application programs) must be such that they end with the last three digits of your Dalhousie student ID (Banner ID).

Submission requirements

Your submission should have the usual packaging (front page, TOC (Table of Contents), etc.). This assignment is simple and hence the submission may also be relatively simple (commentary not needed). Submit the code for your smart contract methods and the code for your Dapp. Instructions on the manner of your submission will follow (e.g., code on a git vs as .sol files).

For an appeal of your grade, see instructions in the course syllabus.