JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 26 No. 2, pp. 229-235, February 2021
https://doi.org/10.9708/jksci.2021.26.02.229

# An Evaluation Management System Using Blockchain

Su-Hyun Lee*

*Professor, Dept. of Computer Engineering, Changwon National University, Changwon, Korea

## [Abstract]

Blockchain, recognized as one of the core technologies of the 4th industrial revolution, is an Internet-based distributed data management system which does not require centralized control. Blockchain is characterized by the integrity and reliability of information, and blockchains can be used where such characteristics are required. Typical applications of blockchain include finance, transaction, and evaluation management. In this paper, we designed a blockchain-based evaluation management system that allows users to freely create and manage evaluation instances. Evaluation managers can create an evaluation instance according to their purpose and allocate evaluation items and evaluators. When the evaluators finish evaluating the evaluation items, the evaluation manager can aggregate the evaluation results for the instance. Someone, who want to perform evaluation for various purposes but do not have an evaluation system, can implement the evaluation system relatively simply by using this system. In addition, due to the characteristic of the blockchain, evaluators cannot modify the evaluation scores they have already recorded, and neither the system administrator nor the evaluation administrator can modify the evaluation scores. For this reason, the reliability of the evaluation increases.

▸ **Key words**: Blockchain, Smart Contract, Evaluation Management, Lecture Feedback, Ethereum

## [요 약]

제4차산업혁명의 핵심기술 중의 하나로 인식되고 있는 블록체인은 인터넷을 기반으로 중앙집중적인 제어가 없이도 작동되는 자율적인 분산 정보관리시스템이다. 블록체인은 정보의 무결성, 신뢰성 등을 특징으로 하며, 이러한 특징을 필요로 하는 곳에는 블록체인을 활용할 수 있다. 대표적인 응용으로는 금융, 거래, 평가관리 등을 들 수 있다. 본 논문에서는 블록체인을 기반으로 사용자들이 자유롭게 평가개체(instance)를 생성하고 관리할 수 있는 평가관리시스템을 설계하였다. 평가관리자는 자신의 목적에 맞게 평가개체를 하나 생성하고, 평가항목과 평가자들을 배정할 수 있다. 평가자들이 평가항목에 대해서 평가를 마치면 평가관리자는 해당 평가개체에 대해서 평가 결과를 집계를 할 수 있다. 여러 용도로 평가를 시행하고 싶지만 평가시스템을 갖추지 못한 경우에 본 시스템을 이용하면 이를 비교적 간단히 평가시스템을 구현할 수 있다. 또한 블록체인의 특성으로 인하여 평가자들은 자신이 이미 기록한 평가 값을 수정하지 못하며, 시스템 관리자 또는 평가 관리자 역시 평가 값을 수정하지 못한다. 이런 이유로 평가의 신뢰성은 높아진다고 볼 수 있다.

▸ **주제어**: 블록체인, 스마트 계약, 평가관리, 강의 피드백, 이더리움

• First Author: Su-Hyun Lee, Corresponding Author: Su-Hyun Lee
*Su-Hyun Lee (sleepl@changwon.ac.kr), Dept. of Computer Engineering, Changwon National University
• Received: 2019. 02. 12, Revised: 2021. 01. 25, Accepted: 2021. 01. 25.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

## I. Introduction

제4차산업혁명의 핵심기술 중의 하나로 인식되고 있는 블록체인은 인터넷을 기반으로 중앙집중적인 제어가 없이도 작동되는 자율적인 분산정보관리시스템이다. 블록체인은 정보의 무결성, 신뢰성, 기록성 등을 특징으로 하며, 이러한 특징을 필요로 하는 곳에는 블록체인을 활용할 수 있다. 대표적인 응용으로는 금융과 거래를 들 수 있다[1]. 하지만 블록체인은 정보가 블록에 등록되는 시간이 필요해서 즉시 처리되지 못하는 단점이 있으며, 블록을 참여자가 공유하기 때문에 정보에 대한 보안은 완벽하지 않다.

대학교육에서 블록체인을 활용할 수 있는 분야는 보안성보다는 신뢰성이 중요한 분야이다. 학생들의 성적 데이터는 기록이 된 이후에는 변경이 되면 안 되기 때문에 블록체인에 저장하기에 적합하다. 반면에 학생의 인적정보는 보안성이 중요하므로 블록체인으로 저장하면 위험할 수 있다.

본 논문에서는 블록체인을 기반으로 사용자들이 자유롭게 평가개체(instance)들을 생성하고 관리할 수 있는 평가관리시스템을 설계하였다. 평가관리는 평가결과가 한번 저장되면 평가자, 평가관리자, 시스템관리자 등 누구도 수정할 수 없어야 하므로 블록체인을 활용하기에 적절한 응용분야이다.

평가관리시스템은 다음과 같이 동작한다. 평가관리자는 자신의 목적에 맞게 평가개체를 하나 생성하고, 평가항목과 평가자들을 배정할 수 있다. 평가자들이 평가항목에 대해서 평가를 마치면 평가관리자는 해당 평가개체에 대해서 평가 결과를 집계를 할 수 있다. 평가자들이 평가한 내용은 블록체인에 저장되기 때문에 평가의 신뢰성을 보장할 수 있다.

본 시스템의 응용 사례로 강의평가를 예로 보면, 평가관리자는 담당교수가 되고, 평가자는 학생이 된다. 평가관리자가 필요시에 언제든지 새로운 평가개체를 생성할 수 있다는 것이 본 논문의 중요한 기여이다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구로서 블록체인에 대해서 정리하였다. 3절에서는 평가관리시스템의 설계와 구현에 대하여 설명하였고, 4절에서는 제안한 평가관리시스템을 활용한 사례를 제시하였다. 마지막으로 5절에서는 결론을 맺는다.

## II. Preliminaries

### 1. Blockchain

블록체인은 2009년 사토시 나카모토에 의해서 만들어진 비트코인[2]에서 출발하였다. 비트코인(bitcoin)은 P2P 네트워크, 암호화, 전자서명, 해싱, POW(작업증명) 등의 기술을 이용하여 만들어진 암호화폐이다. 블록체인은 비트코인에 사용된 아이디어를 전 산업분야에 적용하고자 하는 기술이다.

블록체인은 블록들이 사슬처럼 연결되어 있다. 하나의 블록에는 거래 기록 등이 암호화되어 저장되며, 블록을 연결하기 위한 해시 값을 가지고 있다. 즉, 현재 블록은 이전 블록의 해시 값을 이용하여 만들어지게 되는데 이 때문에 블록에 저장된 정보를 변경하지 못하게 된다. 예를 들어, 1번 블록의 내용을 변경하게 되면 2번 블록의 해시 값이 변경되고 따라서 2번 블록의 내용도 변경이 일어난다. 2번 블록이 변경되었기 때문에 이어서 3번 블록도 변경이 일어난다. 이와 같은 식으로 k번 블록의 내용이 변경되면 k+1번부터 현재까지의 모든 블록을 변경해야 하고 이는 엄청난 노력이 필요하다. 설령 어렵게 블록들의 내용을 변경하였다 할지라도 블록들은 P2P 네트워크를 이용하여 전세계의 사용자가 공유하고 있기 때문에 나의 변경 사항을 전세계의 사용자들이 승인을 해 주질 않는다. 이런 이유로 블록체인에 저장된 정보는 변경이 불가능하다고 한다.

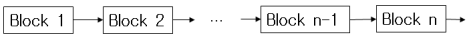


Fig. 1. Blockchain

블록이 연결할 때 사용하는 기술은 해싱(hashing)[3]이다. 해싱은 정보로부터 지문(fingerprint)을 생성하는 것으로 그림 2와 같이 주어진 정보를 256비트로 요약하여 해시값을 구한다. 해시의 중요한 특징은 다음과 같다.

(1) 입력되는 정보가 동일하면 출력되는 해시값도 동일하다.

(2) 입력 정보가 조금만 변경되어도 해시값은 크게 변하게 된다.

(3) 입력값과 해시값 사이에는 특정한 패턴이 없어서, 해시값을 이용하여 입력값을 추측하는 것은 불가능하다. 이와 같은 성질 때문에 해시 함수를 단방향 함수(one-way function)라고 한다.


Fig. 2. Hashing

사슬로 연결된 블록들은 P2P 네트워크를 이용하여 전세계의 사용자가 블록의 복사본을 공유한다. 이에 따라 모든 사용자가 같은 복사본을 가지고 있는지를 보장하는 방법이 필요하다. 이를 합의 알고리즘(consensus algorithm)[4]이라고 한다. 비트코인에서는 Proof-of-Work(POW)[5] 알고리즘을 사용한다.

## 2. Ethereum

이더리움(Ethereum)[6]은 2015년 비탈릭 부테린이 개발한 블록체인이다. 이더리움은 비트코인과 비슷하게 Ether라는 암호화폐를 발행한다. 이더리움의 가장 큰 특징은 스마트 계약(smart contract)[7, 13]을 작성할 수 있다는 것이다. 스마트 계약은 '계약'이라는 단어가 포함되어 있지만 근본적으로는 블록체인에서 실행되는 '프로그램'으로 볼 수 있으며, 다양한 범용적인 프로그램의 작성이 가능하다. 이더리움에는 다음과 같은 두 가지 유형의 계정이 있다.

● 일반 계정(사용자 계정)

개인 사용자가 사용하는 계정으로 개인키로 제어되는 계정이다. 개인이 계정과 관련된 개인키를 소유하고 있으면 Ether와 메시지를 보낼 수 있다.

● 계약 계정

스마트 계약을 위한 계정으로 자체 코드가 있고 코드에 의해 제어되는 계정이다. 따라서 계약은 하나의 계정이고 코드를 포함한다.

트랜잭션은 새 계약 계정을 만들거나 계정을 호출하여 시스템 상태를 변경할 수 있다. 일반 계정에 대한 통화는 Ether만을 계정으로 전송할 수 있지만 계약 계정에 대한 통화는 계약과 관련된 코드를 추가로 실행한다. 스마트 계약 코드는 이더리움 가상기계(Ethereum Virtual Machine; EVM)에서 실행된다.

그림 3은 이더리움 어플리케이션을 개발하는데 가장 많이 사용되고 있는 solidity 언어[8]로 작성한 간단한 스마트 계약 예제이다.

```
pragma solidity ^0.4.18;
contract SimpleStorage {
  uint dataStore;
  function set(uint x) public {
    dataStore = x;
  }
  function get() constant public returns (uint) {
    return dataStore;
  }
}
```

Fig. 3. A Sample Smart Contract

## 3. Truffle Framework

전 세계적인 네트워크에서 작동하는 이더리움은 스마트 계약을 배포하고 실행하는데 여러 절차와 제약이 있어, 스마트 계약을 개발하는 용도로는 적합하지 않다. 이에 PC 환경 내에서 자유롭게 스마트 계약을 작성하고 테스트할 수 있는 환경이 필요하다. 트러플(truffle)[9]은 이더리움 블록체인에서 작동하는 solidity 코드(스마트 계약)를 자신의 PC 환경에서 설치하여 테스트할 수 있는 도구이다. 사용자가 작성한 스마트 계약 코드는 트러플에서 컴파일되어 개인적인 블록체인으로 배포된다. 배포된 이후에는 스마트 계약이 잘 동작하는 지를 테스트할 수 있다.

트러플에서 사용하는 개인 블록체인은 가나슈(ganache)[9]이다. 가나슈는 네트워크에 연결할 필요없이 PC에서만 작동시킬 수 있어 스마트 계약을 쉽게 배포 및 테스트해 볼 수 있다. 가나슈는 테스트를 위해 기본적으로 제공되는 10개의 계정에 각각 100 Ether가 들어 있어 테스트하기에 적당하다.

# III. Evaluation Management System

## 1. Concept and Design

설문조사나 강의평가 등의 간단한 평가를 수행할 수 있는 여러 도구들이 존재하지만 이들 도구들은 평가자의 익명성이 보장되지 않고 평가결과를 수정하는 것도 가능하다. 다양한 목적의 간단한 평가가 가능하면서도 익명성과 평가결과가 보존되는 평가관리시스템을 이더리움 블록체인에서 작동되는 스마트 계약으로 구현하였다.

본 평가관리시스템에서 평가관리자는 자신의 목적에 맞게 평가개체(instance)를 하나 생성하고, 평가항목과 평가자(사용자)들을 배정할 수 있다. 평가자들이 평가항목에 대해서 평가를 내리면 평가관리자는 평가개체 전체에 대해서 집계를 할 수 있다. 평가자들이 평가한 내용은 블록체인에 저장되기 때문에 평가의 신뢰성을 보장할 수 있다.

그림 4는 평가관리시스템이 동작하는 개념이다.

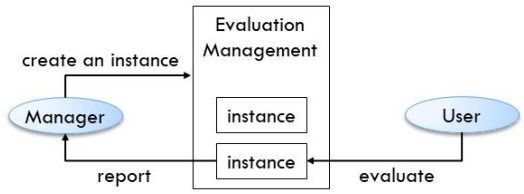


Fig. 4. The Evaluation Management System

(1) 평가개체의 생성

평가를 시행하고 싶은 사용자는 평가개체를 생성함으로써 해당 평가의 관리자가 된다. 평가개체를 생성할 시에는 평가항목의 개수를 입력하여야 한다. 평가항목은 평가의 대상이 되는 항목이며, 평가가 선거가 된다면 후보자가 평가항목이 되며, 강의평가라면 주차가 평가항목이 되고, 과제평가라면 각 과제가 평가항목이 된다.

평가개체가 생성되면 블록체인 내부에 평가항목 데이터가 입력되며 평가 ID가 반환된다. 이 평가 ID는 평가자가 평가를 시행할 때 해당 평가개체를 지정하는데 사용된다.

(2) 사용자 등록

생성된 평가개체는 인증된 사용자만이 평가를 시행할 수 있다. 이를 위하여 평가관리자는 사용자를 등록하여야 한다. 사용자를 등록할 때에는 사용자의 주소와 사용자가 사용할 수 있는 토큰의 수를 입력하여야 한다. 토큰은 사용자가 평가를 할 때 사용하며, 평가가 선거라면 총 토큰은 1이 되고, 강의평가라면 100과 같은 값을 가질 수도 있다.

사용자는 자신의 남아있는 토큰의 양을 검사할 수 있으며, 남아있는 토큰의 양을 넘지 않는 한도에서 평가항목에 배분하여 평가할 수 있다.

(3) 평가하기

생성된 평가개체에 대하여 등록된 사용자는 자신이 가진 총 토큰의 범위 내에서 해당 평가개체의 평가항목에 평가를 할 수 있다. 사용자가 평가를 할 때에는 평가점수에 해당하는 토큰의 양을 입력하여야 한다.

하나의 평가 ID에 대하여 사용자는 한 번의 평가만 가능하다. 평가의 신뢰성을 높이고자 사용자는 자신이 한 번 평가한 항목에 대해서 재평가 한다거나 평가를 수정할 수는 없다.

평가에 대한 결과로서 다음의 오류가 발생할 수 있다.

- 오류 코드 -1: 잘못된 평가 ID
- 오류 코드 -2: 허용되지 않는 사용자
- 오류 코드 -3: 이미 평가한 항목에 다시 평가를 시도
- 오류 코드 -4: 남아 있는 토큰의 양이 충분하지 않음

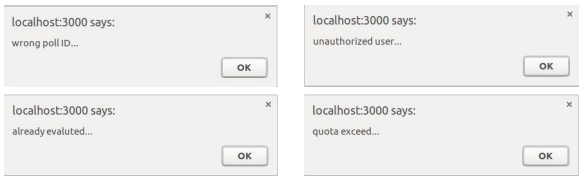


Fig. 5. Error Messages

(4) 평가 집계

평가개체의 관리자는 사용자가 평가한 결과를 집계하여 볼 수 있다. 각 평가항목에 대하여 총 토큰의 수와 평가한 사용자의 수를 집계한다.

## 2. Implementation

설계한 평가관리시스템을 이더리움 블록체인, 트러플, web3.js 등을 이용하여 구현하였다. 그림 6은 평가관리시스템의 구성요소를 보여준다.

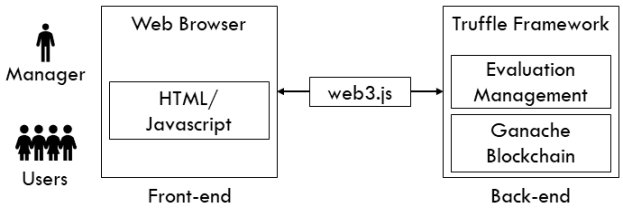


Fig. 6. Components of the System

평가관리시스템을 위한 스마트 계약을 solidity 언어로 작성하였다. 그림 7은 스마트 계약 코드의 일부이다.

```
contract EvalManager {
  struct User {
    uint16 remainToken;  //사용자에게 남아있는 토큰의 수
    bytes4 voteList;  //각 issue에 투표하였는가?
  }
  struct Issue {
    uint16 totalToken; //누적 토큰 수
    uint16 numResponse; //응답자의 수
  }
  mapping (address => address) public pollOwner;
  mapping (address => mapping (address => User)) public users;
  mapping (address => mapping (address => bool)) public userList;
  mapping (address => Issue[]) public issues;

  function vote(address poll, uint8 issueNum, uint16 token) payable public {
    if (pollOwner[poll] == address(0))  //wrong poll ID
        { errorCode = -1; return; }
    if (userList[poll][msg.sender] == false)  //unauthorized user
        { errorCode = -2; return; }
    bytes4 v = users[poll][msg.sender].voteList;
    if (v & (bytes4(0x00000001)<<issueNum) != 0)  //already evaluated
        { errorCode = -3; return; }
    if (users[poll][msg.sender].remainToken < token)  //quota exceed
        { errorCode = -4; return; }

    users[poll][msg.sender].voteList = v | (bytes4(0x00000001)<<issueNum);
    users[poll][msg.sender].remainToken -= token;
    issues[poll][issueNum].totalToken += token;
    issues[poll][issueNum].numResponse ++;
  }
```

Fig. 7. A Part of Evaluation Management Contract

web3.js[10]는 자바스크립트(JavaScript)를 기반으로 한 분산형 웹으로 이더리움 블록체인을 웹에서 사용하기 쉽도록 한다. web3.js는 내부적으로 JSON API를 호출하여 이더리움 블록체인에 접근한다. 블록체인에서 실행 중

인 스마트 계약의 함수를 실행하고자 한다면 스마트 계약의 주소, 실행할 함수, 함수의 매개변수를 전달한다. 그림 8에서는 web3.js를 사용하여 스마트 계약의 함수를 호출하는 코드를 보여준다.

```
handleVote: function(addr,user,issueNum,token){
  var evalInstance;
  App.contracts.EvalManager.deployed().then(function(instance) {
    evalInstance = instance;
    return instance.vote(addr,issueNum-1,token, {from: user, gas:3000000});
  }).then( function(result){
    evalInstance.errorCode().then(function(ret) {
      if (ret >= 0) alert("evaluation success... remaining tokens are " + ret);
      if (ret == -1) alert("wrong poll ID...");
      if (ret == -2) alert("unauthorized user...");
      if (ret == -3) alert("already evaluted...");
      if (ret == -4) alert("quota exceed...");
    });
  }).catch( function(err){
    alert("error... " + err);
    console.log(err.message);
  })
},
```

Fig. 8. An Example of Calling a Function in Smart Contract

Solidity로 작성한 스마트 계약은 트러플에서 컴파일한 후 블록체인에 배포한다. 사용하는 명령어는 다음과 같으며, 그림 9에서 스마트 계약이 컴파일되고 배포되는 절차가 실행되는 장면을 보였다.

```
$ truffle compile
$ truffle migrate --reset
```

```
ubuntu@ubuntu-vbox:~/EvalManager$ truffle compile
Compiling ./contracts/EvalManager.sol...
Compiling ./contracts/Migrations.sol...
Writing artifacts to ./build/contracts

ubuntu@ubuntu-vbox:~/EvalManager$ truffle migrate --reset
Using network 'development'.

Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0x84f792df9a8a444b89416e02e7ff382a74ad0b0efe2bbf1aa8a14ee9ce21ba1f
  Migrations: 0x8cdaf0cd259887258bc13a92c0a6da92698644c0
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying EvalManager...
  ... 0xa3a425103e930149757daba3053cb72f2c20046e19d468c468e77298fabd3766
  EvalManager: 0xf12b5dd4ead5f743c6baa640b0216200e89b60da
Saving artifacts...
```

Fig. 9. Compile and Deploy the Smart Contract

## IV. Lecture Feedback System

현재 각 대학에서 사용하고 있는 강의평가시스템은 학기말에 한 번 피드백을 받을 수 있다. 이런 경우에 강의 피드백은 피드백을 받은 해당 학기의 강의 개선을 위해서는 사용될 수가 없고 다음 학기의 강의에 반영된다. 이런 단점 때문에 많은 대학에서는 학기 중간에 중간강의평가를 실시하고는 있으나 피드백의 주기가 너무 길다는 단점이 있다. 이에 본 논문에서 제안한 평가관리시스템을 이용하여 매주 강의피드백이 이루어지는 평가개체를 생성하였다.

그림 10은 강의피드백이 동작하는 개념이다.

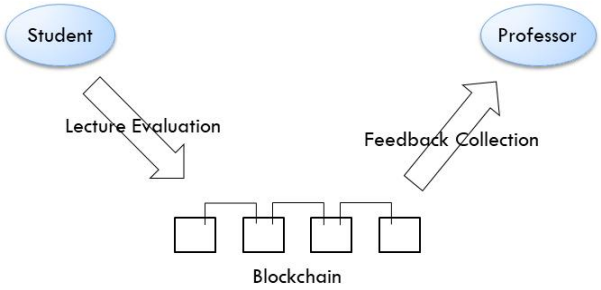


Fig. 10. The Lecture Feedback

담당교수가 강의평가를 시행하려면 평가개체를 생성하여야 한다. 그림 11은 평가관리시스템에서 하나의 평가개체를 생성하는 그림이다. 평가 ID는 자동으로 생성되며 각 평가에서 이 ID로서 다른 평가와 구별된다.

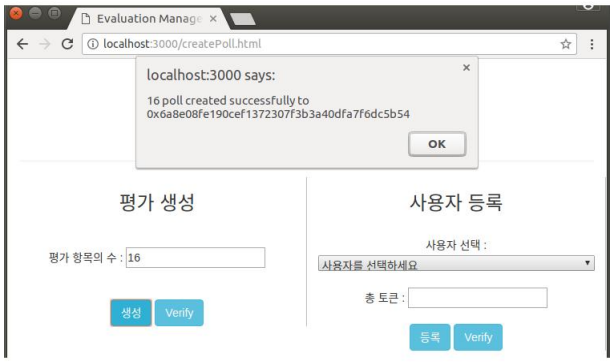


Fig. 11. Creation of an Evaluation Instance

학생은 학기초에 개인별로 n개의 토큰을 부여받는다. 이 토큰은 매주 강의평가 시에 사용할 수 있는데 한 학기 동안 학생은 자신이 부여한 토큰의 합이 n이 넘지 않도록 하여야 한다. 그림 12는 해당 평가 ID에 사용자를 등록하는 그림이다. 사용자를 등록할 때는 사용자 주소와 토큰의 수를 입력한다.

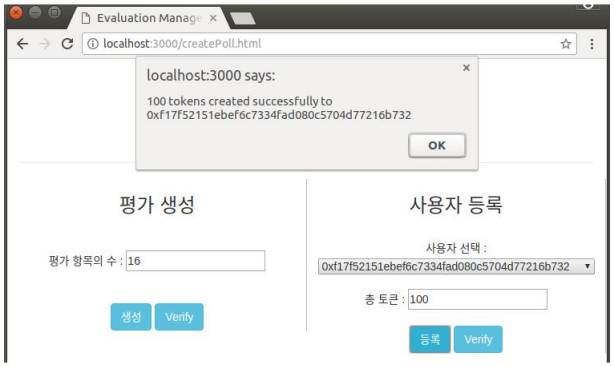


Fig. 12. Registration of a User

토큰은 강의에 대한 이해도나 강의의 전반적인 강의평을 나타내는 용도로 사용한다. 학생은 매주의 강의에 대한 자신의 강의평가점수를 토큰의 수로 표현하여 블록체인에 제출한다. 그림 13은 학생이 평가를 하는 화면이다. 화면에서 본인의 남아있는 토큰의 수를 확인할 수 있고, 평가점수를 제출할 수 있다. 한 번 평가가 이루어지면 평가점수를 수정하거나 삭제할 수 없다.

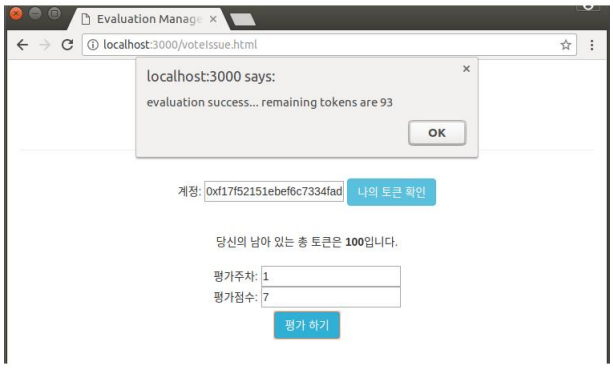


Fig. 13. Screen for Student

담당교수는 매주 제출된 피드백 토큰의 수를 합산하여 토큰의 값으로 강의에 대한 피드백을 산출한다. 학생들의 피드백 값은 학생의 개인적인 기준에 의하여 부여하였으므로, 이를 합산한 값에 수치적인 의미를 부여할 수는 없다. 다만, 담당교수는 매주 합산된 값의 추이를 파악할 수 있고, 이를 토대로 학생들의 피드백을 받을 수 있다. 그림 14는 담당교수가 평가를 집계한 화면이다.

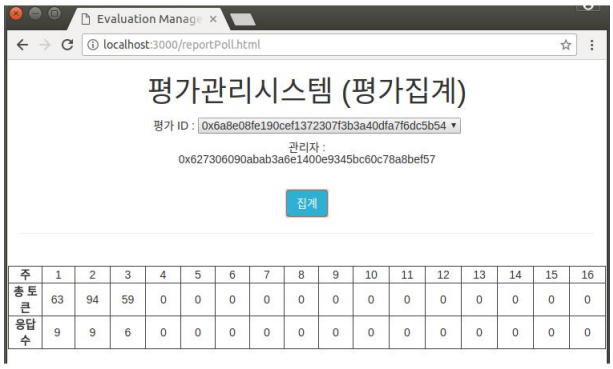


Fig. 14. Screen for Professor

## V. Conclusions

본 논문에서는 다양한 목적의 평가에서 사용할 수 있는 평가관리시스템을 이더리움 블록체인을 이용하여 스마트 계약으로 구축하였다.

개인적으로 간단한 평가를 수행할 수 있는 여러 도구들이 존재하지만 이들 도구들은 평가자의 익명성이 보장되지 않고 평가결과를 수정하는 것도 가능하다. 다양한 목적의 간단한 평가가 가능하면서도 익명성과 평가결과가 보존되는 평가관리시스템을 설계하고 구현하였다. 평가가 필요시에 별도의 시스템을 구축할 필요가 없이 언제든지 새로운 평가개체를 생성할 수 있다는 것이 본 논문의 중요한 기여이다.

본 평가관리시스템의 사례로서 강의피드백시스템을 제시하였다. 현재 각 대학에서는 독자적인 강의평가시스템이 있으며, 이 시스템을 변경하려면 대학의 정책적 결정이 우선시되어야 하고 또한 시스템을 변경하는 비용이 발생하게 된다. 매주 강의평가를 하는 시스템이 도입되지 않은 대학에서 매주 강의 피드백을 받으려면 새로운 시스템을 구축하여야 하지만, 블록체인을 이용하면 이를 비교적 간단히 구현할 수 있다. 또한 교수 또는 과목 단위로 자유롭게 구성하는 것이 가능한 장점이 있다.

블록체인 강의피드백시스템은 블록체인의 특성으로 인하여 학생은 자신이 이미 기록한 피드백 값을 수정하지 못하며, 학교의 시스템 관리자 또는 담당교수 역시 피드백 값을 수정하지 못하므로 강의평가의 신뢰성 향상을 기대할 수 있다.

구현된 평가관리시스템에서 평가자는 평가항목에 대하여 자신이 가지고 있는 토큰 수의 범위 안이라면 얼마든지 평가 값을 부여할 수 있다. 평가를 더 정교하게 하려면 평가항목에 부여할 수 있는 최댓값을 지정할 수 있도록 하는 방안도 가능하다. 예를 들어, 한 평가자가 평가항목에 부여할 수 있는 최대 토큰이 5라고 한다면 5점 척도로 평가를 시행하는 것이 가능하다.

현재는 평가개체 및 평가항목의 이름과 설명을 블록체인에 포함할 수가 없고 웹페이지에서만 작성하게 된다. 이런 경우에 평가관리자는 해당 평가를 위한 웹페이지를 만들어야 하는 부담이 있다. 추후에 평가개체 및 평가항목의 이름과 설명을 블록체인에 입력할 수 있도록 개선할 필요가 있다.

또한 시스템은 관리자가 시스템에 사용자를 등록하기 때문에 완전한 익명성이 보장된다고 볼 수 없다. 완전한 익명성을 보장하기 위해서는 관리자는 인증코드만 발급하고 사용자 등록은 인증코드로 허가받은 사용자가 직접 등록하는 방안을 고려할 만하다.

## ACKNOWLEDGEMENT

This research is financially supported by Changwon National University in 2019~2020.

## REFERENCES

[1] Dae-Hong Min, et al., "Blockchain, where is it applied?," ETRI Insight Report 2018-001, May 2018. (in Korean)
[2] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," October 2008. (https://bitcoin.org/bitcoin.pdf)
[3] NIST, "Secure hash standard (SHS)," FIPS PUB 180-4, National Institute of Standards and Technology, August 2015. (https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf)
[4] Jong-Cheol Im, Hyun-Kyung Yoo, Ji-Young Kwak and Sun-Mi Kim, "Blockchain and Consensus Algorithm," Electronics and Telecommunications Trends, Vol. 33 No. 1, February 2018. (in Korean)
[5] Adam Back, "Hashcash - a denial of service counter-measure," August 2002. (http://www.hashcash.org/papers/hashcash.pdf)
[6] Ethereum Project, https://www.ethereum.org/
[7] Wesley Egbertsen, et al., "Replacing Paper Contracts With Ethereum Smart Contracts," June 2016. (https://allquantor.at/blockchainbib/pdf/egbertsen2016replacing.pdf)
[8] Chris Dannen, "Introducing Ethereum and Solidity," Apress, 2017.
[9] Truffle Suite, https://www.trufflesuite.com/
[10] Fabian Vogelsteller, et al., "web3.js Documentation, Release 1.0.0," June 2019. (https://web3js.readthedocs.io/_/downloads/en/v1.2.6/pdf/)
[11] Gwyduk Yeom, "Blockchain-Based Mobile Cryptocurrency Wallet," Journal of The Korea Society of Computer and Information, Vol. 24 No. 8, August 2019.
[12] Min-Ho Kwon and Myung-Joon Lee, "A robust execution scheme for Ethereum blockchain application services," Journal of The Korea Society of Computer and Information, Vol. 25 No. 3, pp. 73-80, March 2020
[13] P. McCorry, et al., "A smart contract for boardroom voting with maximum voter privacy," International Conference on Financial Cryptography and Data Security, April 2017.

## Author

Su-Hyun Lee received the B.S. in Computer Science from Kwangwoon University, Korea in 1987. He received the M.S. and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology(KAIST), Korea, in 1989, 1994, respectively. Dr. Lee is a Professor in the Department of Computer Engineering, Changwon National University since 1996. He is interested in computer algorithm, programming languages, compiler, and blockchain.