

# Programming assignment#2. Ethereum Project

2014004066 서왕규

## 1. Develop Enviroment and description of document

개발은 다음과 같은 환경에서 진행되었다.

```
npm version : 6.14.4
node version : 12.16.3
solidity version : 0.5.17
ganache version : v2.5.4
chrome version : 86.0.4240.198
```

이 문서는 2절에서 각 기능에 대한 설명을 하고, 3절은 윈도우10 기준으로 소프트웨어를 실행하는 절차와 방법, 2가지 테스트 시나리오의 수행절차를 설명한다. 4절은 프로그램의 구조와 디자인에 대하여 서술한다. 5절은 contract 코드레벨 단계의 구현에서 중점을 둔 부분을 서술한다. 6절은 3절에서 소개한 테스트 시나리오의 결과를 단계별로 자세하게 기록한다.

## 2. Description of Feature

### 2.1. Description

Contract는 현재 **최고 가격의 입찰과 그 입찰의 사용자만을 저장**한다. 그 비용보다 높은 입찰을 제시하는 사용자는 Contract로 ETH를 보내면 입찰이 완료되고, 이전의 최고 가격 입찰자에게 돈을 환불해주는 것으로, 전체 입찰 기록을 Contract가 저장하여 발생하는 비싼 Storage 연산을 줄일 수 있고, 연산하여 최고 입찰가를 찾는 opcode로 인한 거래 수수료의 비용이 올라가는 것을 막을 수 있다.

사용자 A,B,C,D, 경매 주최자 : A, 입찰자 : B,C,D가 있다.

- ① A는 사용자 등록을 한 뒤, 소유한 상품의 경매를 주최한다.
- ② B는 경매의 시작가초과~상한가이하의 입찰가로 입찰한다. 입찰을 하는 순간 B의 계좌에서 입찰가만큼의 ETH가 Contract address로 지불된다.
- ③ C는 경매의 현재최고입찰가초과~상한가이하의 입찰가로 입찰한다. 입찰을 하는 순간 C의 계좌에서 입찰가만큼 Contract address로 지불되며, 이전 입찰자인 B에게 Contract는 B가 입찰한 입찰 비용을 돌려준다.
- ④ D는 경매의 상한가로 입찰한다. 경매는 종료되며 아이템의 소유권은 A에서 D로 이동되며, D는 Contract address로 상한가를 지불한다. Contract는 이전 입찰자 C에게 돈을 환불하며, 경매 주최자 A에게 D가 지불한 금액을 지불한다.

이 때, 다음과 같은 입찰은 웹 페이지 상에 표시되지 않거나, Contract상에서 거부된다.


- ① 경매 주최자가 자신의 아이템에 입찰하는 경우
- ② 이전 최고 가격 입찰자가 자신인 경우, 다른 사람이 입찰하기 전까지 다시 입찰할 수 없다.
- ③ 상한액을 초과하는 입찰은 Contract에게 거부된다.

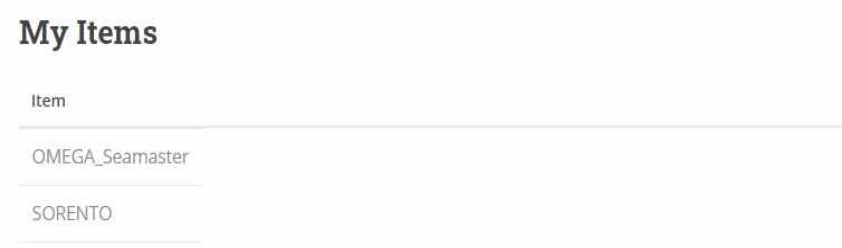
## 2.2. Features

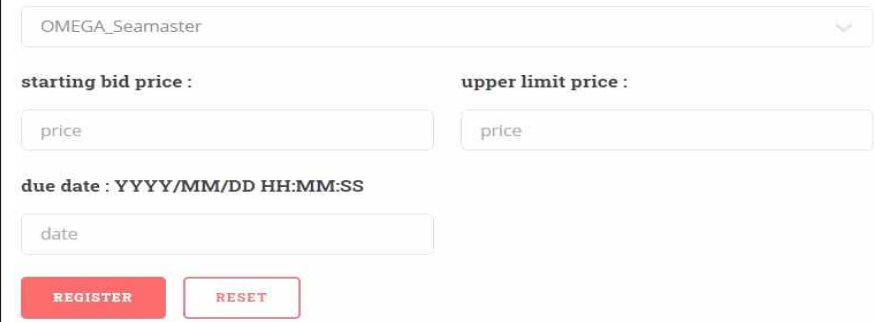
Dapp의 기능과 해당 기능에 대한 설명 및 조건, 구현 화면을 설명한다.

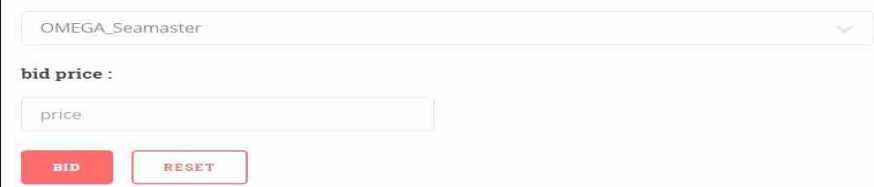
기능 번호		F001
기능 명칭		이름 등록 및 변경
상세 설명	정의	이름 등록 및 변경
	세부 내용	◎ 블록체인에 사용자 주소와 연결되는 계정 등록
입력		String(사용자 이름)
사전 조건		F002를 통해 address의 선택이 된 상태
사후 조건		Contract에 address에 대한 이름이 저장
구현 화면		<p><b>Identity</b></p> <p>Key : 0x9d2084D1b289B2c2dDFC6AdbfcCf04Cad8946C18</p> <p>Ether : 99.29639684</p> <p>Name : 왕규</p> <p>change name</p> <div>Enter user name</div> <div>CHANGE NAME</div>

기능 번호		F002
기능 명칭		사용자 변경
상세 설명	정의	사용자 변경
	세부 내용	◎ 브라우저에서 사용할 사용자의 address 선택
입력		Address
사전 조건		None
사후 조건		브라우저에서 사용할 사용자의 address가 선택 사용자 address에 대한 잔고가 표시 사용자 name이 블록체인에 등록되어 있는 경우 표시
구현 화면		<p>change user</p> <div>Enter ethereum key</div> <div>CHANGE USER</div>

기능 번호		F003
기능 명칭		아이템 등록
상세 설명	정의	아이템 등록
	세부 내용	◎ Contract에 아이템을 기록 - 선택된 사용자 계정이 소유자가 됨
입력		String(아이템 명칭 / 반점 기호 사용 금지)
사전 조건		F001, F002를 통해 주소가 선택되고, 사용자 이름이 등록되어 있어야 함.
사후 조건		Contract에 사용자 주소 소유의 아이템이 등록됨
구현 화면		

기능 번호		F004
기능 명칭		아이템 조회
상세 설명	정의	아이템 조회
	세부 내용	◎ 브라우저에 선택된 사용자 주소가 소유한 아이템의 조회
사전 조건		F002를 통해 브라우저에 사용자 주소가 선택
사후 조건		Contract에 등록된 아이템의 명칭이 출력
구현 화면		

기능 번호		F005
기능 명칭		경매 등록
상세 설명	정의	경매 등록
	세부 내용	◎ 소유한 아이템의 경매 등록 - 경매 상품 명칭은 selected를 통해 소유 아이템에서 선택 - 경매 상품의 시작액과 상한액은 ETH단위 - 마감일은 YYYY/MM/DD HH:MM:SS로 입력
입력		경매 상품 명칭, 경매 시작 비용(단위 : ETH), 경매 상한액(단위 : ETH), 마감일(포맷 YYYY/MM/SS 로 입력)
사전 조건		F001, F002를 통한 주소 선택과 사용자 등록, 경매 상품 소유
사후 조건		Contract에서 경매가 실행
구현 화면		<b>Register Auction Item</b> 

기능 번호		F006
기능 명칭		경매 입찰
상세 설명	정의	경매 입찰
	세부 내용	◎ 등록된 경매에 입찰 - 자신의 경매에는 입찰 불가 - 비용은 ETH 단위 - 현재 경매입찰가 이상, 상한액 이하로만 입찰 가능 - 상한액을 입찰할 경우 경매 종료 - 경매에 등록된 상품 중 선택하도록 UI 구성
입력		입찰 상품 이름, 입찰가
사전 조건		F001, F002를 통한 주소 선택과 사용자 등록, 경매 상품 등록
사후 조건		Contract에 자신의 경매가 입찰됨 이전 입찰은 취소되어, 입찰자에게 금액이 환수됨 입찰 비용 만큼 Contract로 지불 상한액과 동일한 금액을 입찰할 경우 경매가 종료됨
구현 화면		<b>Auction bidding</b> 

기능 번호		F007										
기능 명칭		경매 상품 조회										
상세 설명	정의	경매 상품 조회										
	세부 내용	◎ 현재 경매에 등록된 상품을 조회 - 아이템 명칭, 소유자 이름, 현재 입찰가, 상한액, 마감일 순으로 표시 - 마감일은 Date 타입으로 표시										
사전 조건		None										
사후 조건		현재 경매에 등록된 상품이 표로 출력됨										
구현 화면		<div>Registered auction items</div> <table><thead><tr><th>Item</th><th>Owner</th><th>Bid price</th><th>Upper limit price</th><th>Due date</th></tr></thead><tbody><tr><td>OMEGA_Seamaster</td><td>왕규</td><td>2ETH</td><td>15ETH</td><td>Thu Dec 03 2020 15:34:00 GMT+0900 (대한민국 표준시)</td></tr></tbody></table>	Item	Owner	Bid price	Upper limit price	Due date	OMEGA_Seamaster	왕규	2ETH	15ETH	Thu Dec 03 2020 15:34:00 GMT+0900 (대한민국 표준시)
Item	Owner	Bid price	Upper limit price	Due date								
OMEGA_Seamaster	왕규	2ETH	15ETH	Thu Dec 03 2020 15:34:00 GMT+0900 (대한민국 표준시)								

기능 번호		F008						
기능 명칭		종료 경매 내역 조회						
상세 설명	정의	종료 경매 내역 조회						
	세부 내용	◎ 종료된 경매의 표시 - 아이템 명칭, 변경된 소유자명, 최종 입찰가 순으로 기록 - 아무도 입찰하지 않고, 마감일이 지나 종료된 경매의 경우 입찰 시작가가 최종 입찰가로 기록 - 변경된 소유자는 명칭으로 기록						
사전 조건		None						
사후 조건		경매가 종료된 정보가 테이블로 출력						
구현 화면		<b>Closed auction items</b>  <table> <thead> <tr> <th>Item</th><th>Owner</th><th>Winning bid</th></tr> </thead> <tbody> <tr> <td>OMEGA_Seamaster</td><td>Hyungsuk</td><td>15ETH</td></tr> </tbody> </table>	Item	Owner	Winning bid	OMEGA_Seamaster	Hyungsuk	15ETH
Item	Owner	Winning bid						
OMEGA_Seamaster	Hyungsuk	15ETH						

### 3. Procedure of execution

#### 3.1 Execution of dapp

① 2014004066\_서왕규\_assignment.zip 의 압축을 해제한다.

② 압축이 해제된 디렉토리로 들어가 npm install

cd <디렉토리명>

npm install

```

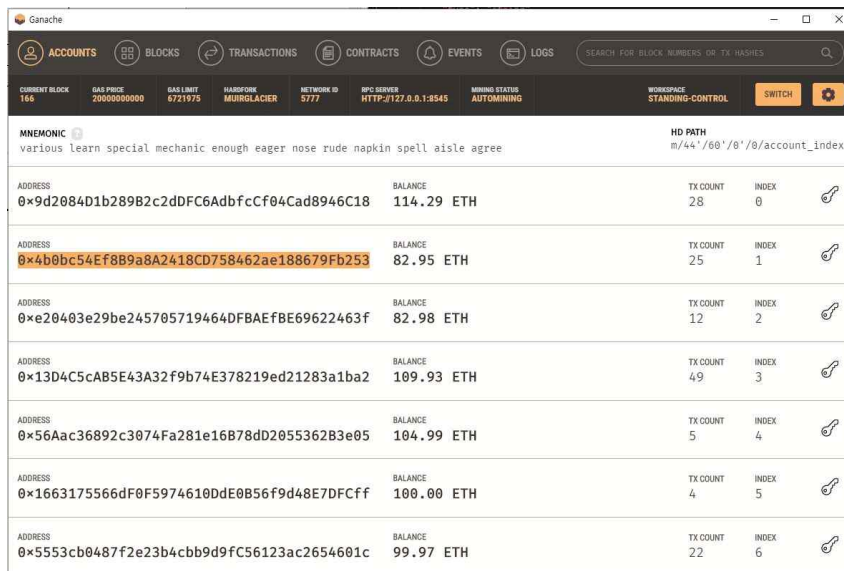
Windows PowerShell
PS C:\Users\jdhk> cd
PS C:\Users\jdhk> cd C:\Users\jdhk\OneDrive\원격\wethereum_interface
PS C:\Users\jdhk\OneDrive\원격\wethereum_interface> ls

디렉터리: C:\Users\jdhk\OneDrive\원격\wethereum_interface

Mode                LastWriteTime         Length Name
----                -
d-----         2020-12-02 오후 6:43             bin
d-----         2020-12-02 오후 6:45             public
d-----         2020-12-02 오후 6:45             routes
d-----         2020-12-02 오후 6:45             smartcontract
d-----         2020-12-02 오후 6:45             views
-a-----         2019-11-04 오후 1:28              997 app.js
-a-----         2020-11-18 오전 11:11          105793 package-lock.json
-a-----         2019-10-29 오후 5:50              315 package.json

```

### ③ ganache 실행



- ④ remix에 접속하여, web3 provider에 contract를 deploy한다.
  - Remix에서 smartcontract/Auction.sol을 불러온다.
  - 컴파일 후, web3 provider를 통해 ganache가 운영하는 블록체인에 deploy한다.
  - public/javascript/web3.js의 contract\_address와 abi를 Remix에서 출력된 정보로 수정한다.
- ⑤ npm start를 통해 서버를 실행한다.

## 3.2 테스트 수행 절차

### [시나리오1]

경매 상품을 상한액으로 구매하여 소유권의 양도를 받는 시나리오

- ① 브라우저가 사용할 주소를 Change user에 입력하고 클릭한다.
- ② 해당 주소의 사용자 이름을 Change name에 입력하고 클릭한다.
- ③ 아이템을 등록한다.
  - 아이템 명칭에는 반점은 포함될 수 없다.
- ④ 해당 아이템을 경매에 등록한다.
  - 날짜는 YYYY/MM/DD HH:MM:SS 의 포맷으로 기록해야한다.
  - 시작가가 상한가를 넘겨서는 안된다.
  - 시작가와 상한가는 ETH 단위로 숫자만 입력한다.

- ⑤ Change user를 통해, 다른 주소의 사용자로 변환한다.
- ⑥ Change name을 통해 이름을 등록한다.
- ⑦ Registered auction items를 통해 현재 최고 입찰가 초과, 상한액 이하의 금액을 입찰한다.
  - require문을 통해 제한 조건에 맞지 않는 입찰 금액은 거부된다.
  - 입찰하는 순간 계좌의 ETH는 Contract로 이동된다. 새로운 입찰의 직전 최고 금액의 입찰자는 자신이 보낸 ETH를 돌려받는다.
- ⑧ 5,6,7 과정을 반복하며 입찰을 테스트해보고 싶은 만큼 테스트한다.
- ⑨ 상한액을 입찰하여, 경매를 종료시키고 소유권을 양도받는다.

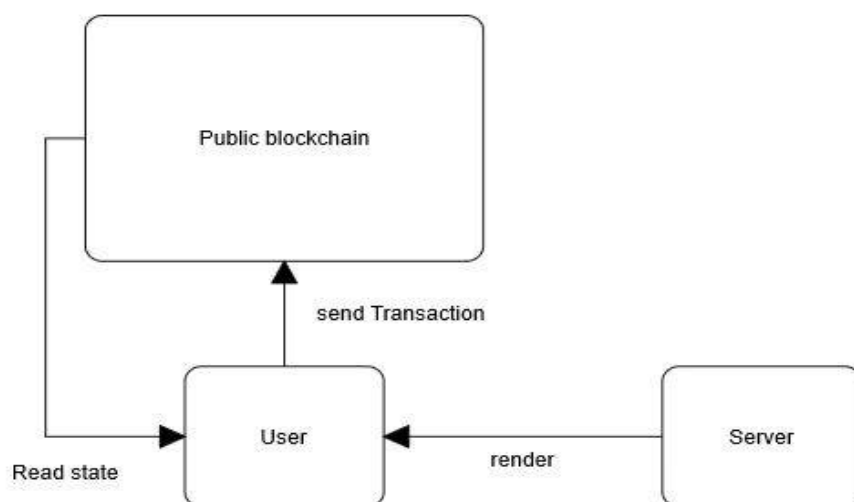
#### [시나리오2]

경매 상품의 마감일을 넘긴 상품이 최고 입찰가에 낙찰되어 양도 받는 시나리오.

- ① 브라우저가 사용할 주소를 Change user에 입력하고 클릭한다.
- ② 해당 주소의 사용자 이름을 Change name에 입력하고 클릭한다.
- ③ 아이템을 등록한다.
- ④ 해당 아이템을 경매에 등록한다.
- ⑤ Change user를 통해, 다른 주소의 사용자로 변환한다.
- ⑥ Change name을 통해 이름을 등록한다.
- ⑦ Registered auction items를 통해 현재 최고 입찰가 초과, 상한액 미만의 금액을 입찰한다.
  - require문을 통해 제한 조건에 맞지 않는 입찰 금액은 거부된다.
  - 입찰하는 순간 계좌의 ETH는 Contract로 이동된다. 새로운 입찰의 직전 최고 금액의 입찰자는 자신이 보낸 ETH를 돌려받는다.
- ⑧ 5분뒤 브라우저에 새로그침을 하여, 입찰이 종료되고, 최고 금액 입찰자에게 소유권이 양도되고, 경매 개설자에게 수입이 입금되는지 확인한다.

## 4. Design

### 4.1. Design of architecture



과제의 소프트웨어 아키텍처는 다음과 같다.

사용자는 브라우저를 통해 서버로 접속한다. 서버는 사용자의 요청을 받고, 브라우저에 페이지를 렌더링한다. 유저는 받은 코드를 통해, 블록체인에 직접 Transaction을 보내 Contract의 상태를 변경하거나, Contract address의 state를 읽어 자신의 브라우저에 반영하게 된다.

이러한 소프트웨어의 구조는 다음과 같이 개선될 수 있다. 퍼블릭 블록체인에서 함수 실행에 필요한 opcode를 줄이고 블록체인 스토리지에 기록하는 양을 없애기 위해, 4.1.2절에서 제시하는 여러 변수들은 서버가 데이터베이스를 갖는 것으로 없애거나 변경할 수 있다. 블록체인에서 발생하는 이벤트를 읽어, 서버가 사용자에게 해당 데이터가 있는 위치나 배열의 크기를 제안할 수 있기 때문에, 원하는 데이터의 위치를 찾기 위해 블록체인 내에서 iteration하거나 storage에 기록해야하는 부담을 줄일 수 있다.

## 4.2. Design of contract storage

컨트랙트는 다음과 같은 storage 변수를 갖는다.

구조체명	타입	변수명	설명
AuctionData	uint	start	현재 입찰 최고가. 기본값은 설정된 입찰 최소 금액
	uint	upper	상한가
	uint	endTime	마감일
	bool	valid	경매의 유효 여부(경매가 등록되지 않거나, 종료된 경우 false)
	address	buyer	현재 입찰 최고가를 입찰한 사용자 주소
	string	name	경매 상품 이름
	bool	lock	재진입 공격을 막기 위한 락
Item	string	name	아이템 명칭
	bool	valid	사용자가 아이템을 소유하고 있는지 여부
User	string	name	유저 이름
	mapping (uint => Item)	items	사용자의 아이템
	mapping (uint => AuctionData)	auctions	사용자가 등록한 경매
	uint	itemcount	사용자의 아이템 개수
	uint	auctioncount	사용자의 경매 등록 상품 개수

itemcount와 auctioncount는 등록할 때마다 증가할뿐 감소되지 않는다. storage의 iteration가 능한 구조체의 원소 삭제 연산이 gas fee를 증가시키기 때문에 valid 변수를 추가하는 것으로 삭제 연산을 수행하지 않고, 유효한 지 여부를 기록한다.



타입	변수	설명
mapping (address => User)	users	주소와 유저를 mapping
mapping (string => address)	items	아이템명을 통해 소유주를 mapping
mapping (string => bool)	isAuctioning	아이템명을 통해 경매여부를 mapping
address[]	userslist	유저 주소 리스트
uint	auctionedItemCount	현재까지 경매에 등록된 숫자
uint	closedCount	현재까지 완료된 경매 숫자

### 4.3. Design of server

Express 프레임 워크로 구현된 서버는 사용자의 브라우저에 index.ejs와 web3.js를 랜더링한다. 사용자의 브라우저는 코드를 받아 수행하는 것으로 퍼블릭 블록체인과 통신한다.

작성한 web3.js의 코드는 과제에서 기존에 제시한 모든 함수를 구현해서 사용한다.

solidity로 작성된 .sol파일을 compile 및 deploy하고, abi와 deploy된 contract 주소를 web3.js에 수정하는 것으로, 블록체인과 통신하는 코드를 작성할 수 있다.

contract.methods(methodName).call() : View function(수수료 없음)

contract.methods(methodName).send() : contract 상태 변경

이 두 함수를 통해, 블록체인과 브라우저는 통신할 수 있다.

#### ① preventDefault

```
$(document).ready(function() {
  startDapp();
  $('form').submit(function(event) {
    var formId = this.id;
    form = this;
    event.preventDefault();
    setTimeout( function () {
      form.submit();
    }, 1000);
  });
});
```

form submit으로 인해, 블록체인과 통신하기 이전에 새로운 페이지로 브라우저가 갱신되어 블록체인에 거래가 발송되지 않을 수 있다. form의 submit을 일정 시간 미루는 것으로 이를 방지한다.

#### ② auctionEnd

```
var getRegisteredAuctionItems = async function() {
  var address = $('#address').text();
  var array = await contract.methods.getAllRegisteredItems().call({from:address});
  for (var i in array['0']){
    if(array['0'][i] !== ""){
      var date = new Date(array['4'][i] * 1000);
      console.log(Date.now()/1000 , Number(array['4'][i]));
      if((Date.now()/1000) > Number(array['4'][i])){
        console.log("aa");
        await contract.methods.auctionEnd(array['0'][i]).send({from:address, gas:6721975});
      }
    } else{
      var text = "<tr><td>" + array['0'][i]
      + "</td><td>" + array['1'][i]
      + "</td><td>" + (Number(array['2'][i])/Math.pow(10,18)).toString() + "ETH"
      + "</td><td>" + (Number(array['3'][i])/Math.pow(10,18)).toString() + "ETH"
      + "</td><td>" + date.toString()
      + "</td></tr>";
      $('#registeredCars').append(text);
    }
  }
}
```

페이지가 갱신될 때마다, startDapp함수를 통해 getRegisteredAuctionItems이 실행된다. 이때, 기한이 지난 경매를 종료시켜주는 auctionEnd를 블록체인상에 요청한다.

실제 서비스의 구현을 가정한다면, auctionEnd함수는 수수료를 요구하므로, 해당 서비스의 운영 기관이 일괄적으로 수행하거나, 기능을 추가하여, 경매의 개최자가 수행하는 것이 더 바람직할 것이다.

## 5. Implementation of contract

Auction.sol이 한 경매단위의 contract가 아닌, 경매들을 모은 Factory의 형식을 지니기 때문에 interface가 되는 contract인 IAuction.sol을 상속받지 않고 구현했다. IAuction.sol파일을 계약내의 함수의 코드의 특징을 서술한다.

### ① Modifier onlyUser

```
modifier onlyUser() {  
    require(bytes(users[msg.sender].name).length != 0, "You must register first");  
    _;  
}
```

함수가 onlyUser로 지정되는 경우, 사용자 등록을 한 뒤에만 사용할 수 있도록 한다.

view function과 registerName을 제외한 모든 함수는 onlyUser로 지정된다.

### ② Lock

각 경매는 lock이라는 boolean 변수를 갖는다. 이는

(1) 구현한 Auction contract에 각 사용자의 balance를 기록하지 않고, 최종 입찰가와 입찰자만을 기록하기 때문

(2) 모든 블록체인의 함수의 수행은 Atomic하게 수행되지만, CA(contract account)가 자신의 함수 이외의 다른 함수의 수행에서, 반드시 순서대로 실행된다는 보장이 없기 때문

이 두가지 이유로 재진입 공격이 가능하기 때문에 추가한 변수이다.

재진입 공격이란, 악의적인 Contract를 구현하여, 경매 Contract의 auctionBidding를 수행하고, auctionbid는 악의적인 Contract의 fallback함수에게 ETH를 전송한다. 악의적인 Contract의 fallback함수는 다시 경매 Contract의 auctionBidding를 수행하여, Contract의 ETH를 가져간다. 이를 막기 위한 일반적인 방법은 각 사용자의 balance를 transfer 전에 업데이트하는 방법이나, 과제의 구현에서 각 사용자의 balance를 유지하지 않고, 최종 입찰가와 입찰자만을 기록하여 사용자의 balance를 알 수 있기 때문에, lock을 사용하여 재진입을 막는다.

```
function auctionBidding(string memory name) payable public onlyUser {
    uint count = 0;
    address owner = items[name];
    for(uint i = 0; i < users[owner].auctioncount; i++){
        if(keccak256(abi.encodePacked((name))) == keccak256(abi.encodePacked((users[owner].auctions[count].buyer)))){
            count = i;
            break;
        }
    }
    require(users[owner].auctions[count].lock == false, "Wait... it is locked");
    users[owner].auctions[count].lock = true;
    AuctionData memory target = users[owner].auctions[count];
    require(target.valid == true, "It is already ended");
    require(target.buyer != msg.sender, "Can not bid before other bid to item");
    require(target.start < msg.value, "It is less than now");
    require(target.upper >= msg.value, "Do not bid over than upper");
    if(target.buyer != owner){
        users[owner].auctions[count].buyer.transfer(target.start);
    }
    users[owner].auctions[count].buyer = msg.sender;
    users[owner].auctions[count].start = msg.value;
    if(target.upper == msg.value){
        auctionEnd(name);
    }
    users[owner].auctions[count].lock = false;
}
```

## 6. Result

3.2절의 두 가지 시나리오를 통합한 시나리오를 테스트한다. 시나리오는 다음과 같다.

“0x18b5981FD6eEFa9F02F96D386F7d8A26419B2727” : 왕규

“0x4CD40DC0a91840a8060649A857f7781E0c87BD8e” : 효일

“0xE0004A954396B3535A332DB3e3766B9f0e5f545d” : 성빈

“0x666F385A8204b664818375dC81735Bd6968bB5df” : 태상

왕규는 item KIA\_SORRENT, APARTMENT 두 아이템을 경매에 올리고, 효일, 성빈, 태상은 입찰한다. KIA\_SORRENT는 효일이 상한가에 낙찰 받고, APARTMENT는 태상이 마감일이 지나 최고 입찰자로 선정되어 낙찰 받는 시나리오를 테스트한다.

- ① 브라우저가 사용할 주소를 Change user에 입력하고 클릭한다.
- ② 해당 주소의 사용자 이름을 Change name에 입력하고 클릭한다.

## Identity

**Key : 0x18b5981FD6eEFa9F02F96D386F7d8A26419B2727**

**Ether : 99.99829402**

**Name : 왕규**

- ③ 아이템을 등록한다.
  - 아이템 명칭에는 반점은 포함될 수 없다.
  - KIA\_SORRENT, APARTMENT 등록

## My Items

Item
KIA_SORRENT
APARTMENT

- ④ 해당 아이템을 경매에 등록한다.
- 날짜는 YYYY/MM/DD HH:MM:SS 의 포맷으로 기록해야한다.
  - 시작가가 상한가를 넘겨서는 안된다.
  - 시작가와 상한가는 ETH 단위로 숫자만 입력한다.

## Register Auction Item

KIA_SORRENT	
starting bid price :	upper limit price :
5	20
due date : YYYY/MM/DD HH:MM:SS	
2020/12/04 12:00:00	
REGISTER	RESET

## Registered auction items

Item	Owner	Bid price	Upper limit price	Due date
KIA_SORRENT	왕규	5ETH	20ETH	Fri Dec 04 2020 12:00:00 GMT+0900 (대한민국 표준시)
APARTMENT	왕규	1ETH	10ETH	Thu Dec 03 2020 10:15:00 GMT+0900 (대한민국 표준시)

- ⑤ Change user를 통해, 다른 주소의 사용자로 변환한다.
- ⑥ Change name을 통해 이름을 등록한다.
- ⑦ Registered auction items를 통해 현재 최고 입찰가 초과, 상한액 이하의 금액을 입찰한다.
- require문을 통해 제한 조건에 맞지 않는 입찰 금액은 거부된다.
  - 입찰하는 순간 계좌의 ETH는 Contract로 이동된다. 새로운 입찰의 직전 최고 금액의 입찰자는 자신이 보낸 ETH를 돌려받는다.
- ⑧ 5,6,7 과정을 반복하며 입찰을 테스트해보고 싶은 만큼 테스트한다.
- ⑨ 상한액을 입찰하여, 경매를 종료시키고 소유권을 양도받는다.

## KIA\_SORRENTO

### 1. 효일의 10 ETH 입찰

유저 등록(입찰 전 잔고)

## Identity

**Key : 0x4CD40DC0a91840a8060649A857f7781E0c87BD8e**

**Ether : 99.99859402**

**Name : 효일**

10ETH 입찰

## Auction bidding

KIA\_SORRENT

bid price :

10

BID

RESET

입찰된 내역

### Registered auction items

Item	Owner	Bid price	Upper limit price	Due date
KIA_SORRENT	왕규	10ETH	20ETH	Fri Dec 04 2020 12:00:00 GMT+0900 (대한민국 표준시)
APARTMENT	왕규	1ETH	10ETH	Thu Dec 03 2020 10:15:00 GMT+0900 (대한민국 표준시)

입찰 후 효일 잔고

## Identity

**Key : 0x4CD40DC0a91840a8060649A857f7781E0c87BD8e**

**Ether : 89.99753244**

**Name : 효일**

## 2. 성빈의 15 ETH 입찰

입찰된 내역

### Registered auction items

Item	Owner	Bid price	Upper limit price	Due date
KIA_SORRENT	왕규	15ETH	20ETH	Fri Dec 04 2020 12:00:00 GMT+0900 (대한민국 표준시)
APARTMENT	왕규	1ETH	10ETH	Thu Dec 03 2020 10:15:00 GMT+0900 (대한민국 표준시)

이전 입찰자 비용 환불

## Identity

Key : 0x4CD40DC0a91840a8060649A857f7781E0c87BD8e

Ether : 99.99753244

Name : 효일

입찰 후 성빈 잔고

## Identity

Key : 0xE0004A954396B3535A332DB3e3766B9f0e5f545d

Ether : 84.99736086

Name : 성빈

## 3. 효일의 상한가(20 ETH)입찰

입찰 후 효일 잔고

## Identity

Key : 0x4CD40DC0a91840a8060649A857f7781E0c87BD8e

Ether : 79.9942633

Name : 효일

입찰 결과(소유권 이전)

### Closed auction items

Item	Owner	Winning bid
KIA_SORRENT	효일	20ETH

### My Items

Item
KIA_SORRENT

이전 입찰자 환불

## Identity

**Key : 0xE0004A954396B3535A332DB3e3766B9f0e5f545d**

**Ether : 99.99736086**

**Name : 성빈**

### APARTMENT

#### 1. 태상의 8ETH 입찰

입찰 전 태상 잔고

## Identity

**Key : 0x666F385A8204b664818375dC81735Bd6968bB5df**

**Ether : 99.99859402**

**Name : 태상**

입찰 후 태상 잔고

## Identity

**Key : 0x666F385A8204b664818375dC81735Bd6968bB5df**

**Ether : 91.99509402**

**Name : 태상**



입찰 화면

## Auction bidding

APARTMENT

bid price :

8

BID

RESET

입찰 내역

## Registered auction items

Item Owner Bid price Upper limit price Due date

APARTMENT 왕규 8ETH 10ETH Thu Dec 03 2020 10:15:00 GMT+0900 (대한민국 표준시)

## 2. 경매 마감시간 종료

마감시간 종료 후 결과

### My Items

Item

APARTMENT

### Auction bidding

bid price :

price

BID

RESET

### Closed auction items

Item Owner Winning bid

KIA\_SORRENT 효일 20ETH

APARTMENT 태상 8ETH

오전 10:15  
2020-12-03



◎ 최종 결과

ADDRESS	BALANCE
0x18b5981FD6eEFa9F02F96D386F7d8A26419B2727	127.99 ETH
ADDRESS	BALANCE
0x4CD40DC0a91840a8060649A857f7781E0c87BD8e	79.99 ETH
ADDRESS	BALANCE
0xE0004A954396B3535A332DB3e3766B9f0e5f545d	100.00 ETH
ADDRESS	BALANCE
0x666F385A8204b664818375dC81735Bd6968bB5df	92.00 ETH

왕규는 KIA\_SORRENTO를 효일에게 소유권을 넘기고, 20 ETH를 받고, APARTMENT의 소유권을 대상에게 넘기고 8ETH를 받음. : 128ETH

효일은 20ETH를 지불 : 80ETH

대상은 8ETH를 지불 : 92 ETH

성빈은 KIA\_SORRENTO에 입찰하였으나, 낙찰받지 못함.