Blockchain - Tool Demo

Health Care Sector - Insurance Claiming System

Team Members:

19C007 - Arun M
19C057 - Nandhakumar Raj S
19C074 - Priyavarthan N

Tool Used:

Remix IDE

Tool Description:

- Remix IDE, is a no-setup tool with a GUI for developing smart contracts. Used by experts and beginners alike, Remix will get you going in double time.
- Remix plays well with other tools, and allows for a simple deployment process to the chain of our choice.
- Remix is famous for our visual debugger.
- Remix is the place everyone comes to learn Ethereum.

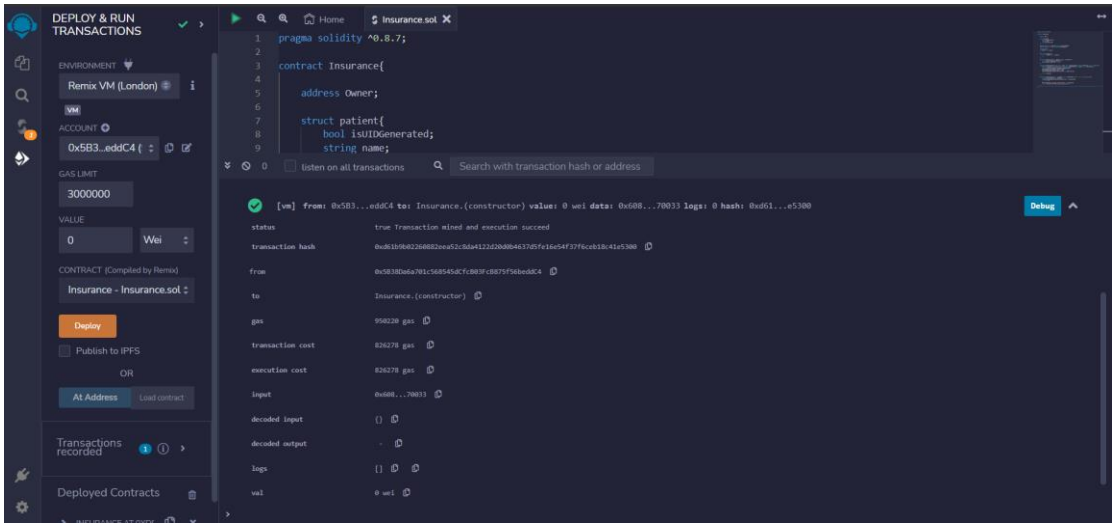
Solidity Code:

```
pragma solidity ^0.8.7;

contract Insurance{

   address Owner;

   struct patient{
      bool isUIDGenerated;
      string name;
      uint amountInsured;
   }
```

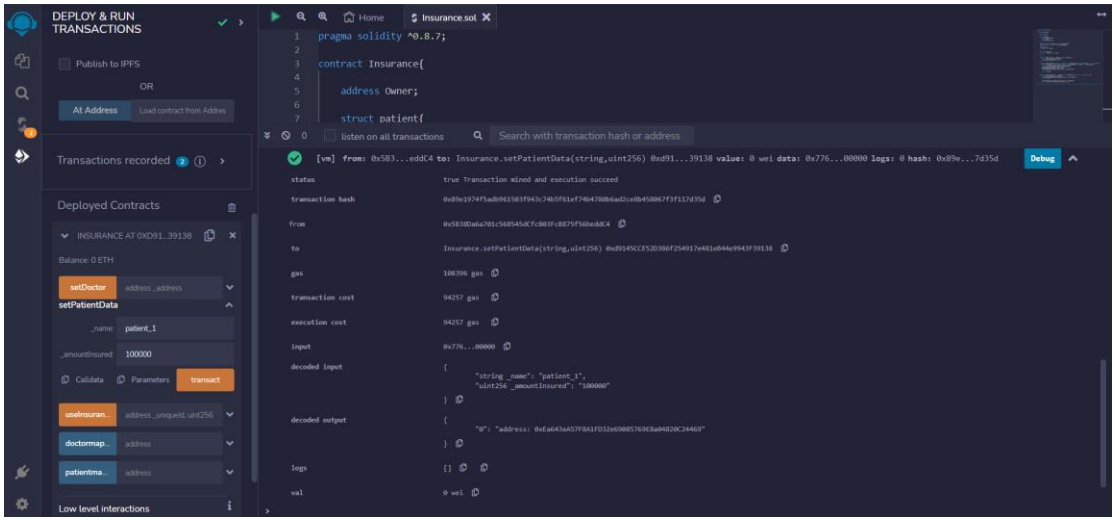
```
    mapping (address => patient) public patientmapping;
    mapping (address => bool) public doctormapping;

    constructor(){
        Owner = msg.sender;
    }

    modifier onlyOwner(){
        require (Owner == msg.sender);
        _;
    }

    function setDoctor(address _address) public onlyOwner() {
        require(!doctormapping[_address]);
        doctormapping[_address] = true;
    }

    function setPatientData(string memory _name, uint _amountInsured)
public onlyOwner() returns (address){
        address uniqueId =
address(bytes20(sha256(abi.encodePacked(msg.sender,block.timestamp))
));
        require(!patientmapping[uniqueId].isUIDGenerated);
        patientmapping[uniqueId].isUIDGenerated = true;
        patientmapping[uniqueId].name = _name;
        patientmapping[uniqueId].amountInsured = _amountInsured;

        return uniqueId;
    }

    function useInsurance(address _uniqueId, uint _amountUsed) public
returns (string memory){
        require(doctormapping[msg.sender]);
        if(patientmapping[_uniqueId].amountInsured < _amountUsed){
            revert();
        }
        patientmapping[_uniqueId].amountInsured -= _amountUsed;
        return "Insurance has been used successfully!";
    }

}
```

Output Screenshots:

Deploying the Contract after Compilation:



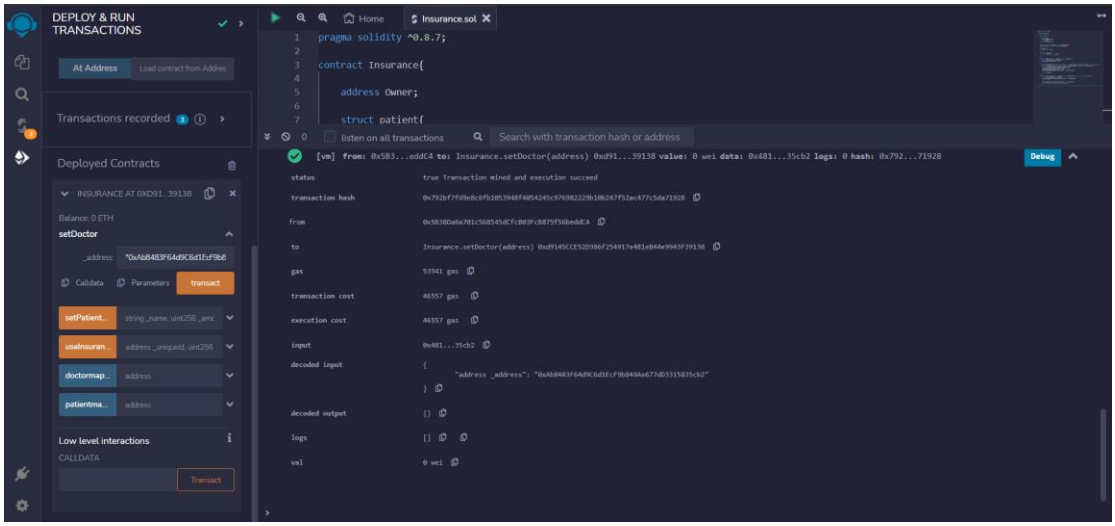
Using the setPatientData function:



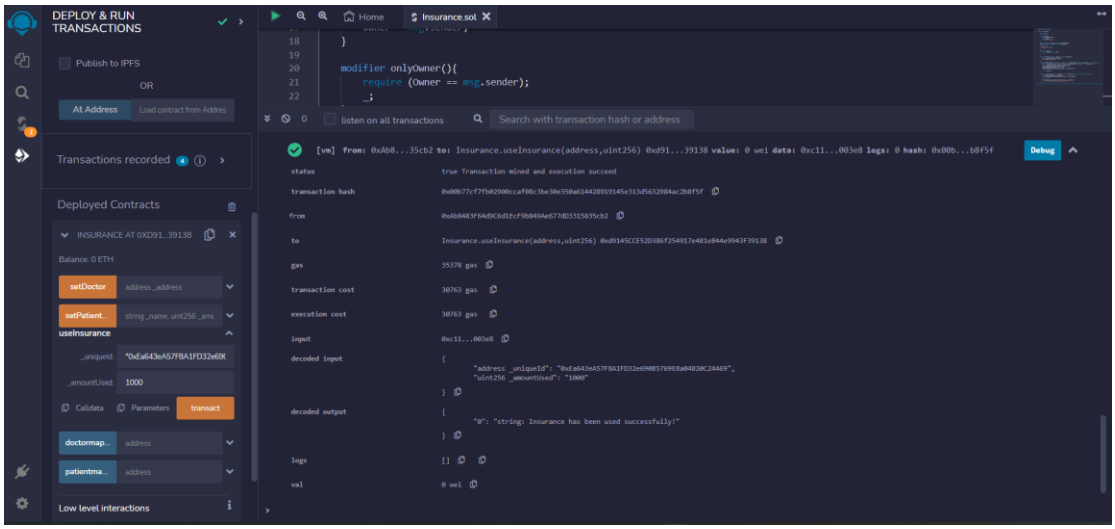
Generated Patient id:

`0xEa643eA57F8A1FD32e69085769E8a04820C24469`

Setting 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 as doctor using setDoctor function:



Performing Insurance claim using useInsurance function:



Updated Insurance balance in the patient mapping: