

CCMP 606 - Orchestration of Cloud Resources

Assignment 1 Report

Instructor: **Yongchang He**

Student: **Hai Nam Nguyen – 000520322**

2023/2024 Winter Semester

Saskatchewan Polytechnic

Submitted:

January 29th, 2024

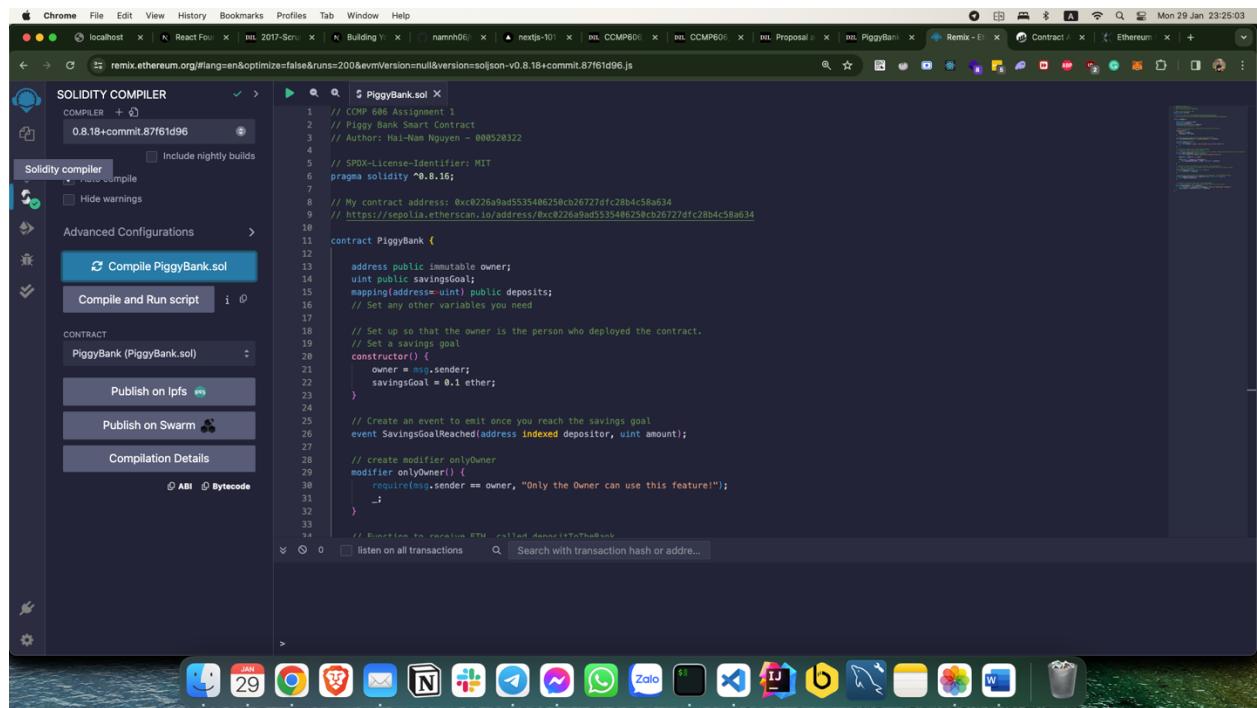
Table of Contents

I.	Compile PiggyBank.sol.....	3
II.	Sepolia ETH on MetaMask	5
III.	Smart Contract Address on the Etherscan Webpage.....	6
IV.	Contract Balance on Remix IDE	6
V.	Explanations and Show How the Functions Work	8

I. Compile PiggyBank.sol

The screenshot(s) show PiggyBank.sol has no compile errors on Remix IDE. (0.5 points)

Compile:



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.16;

// My contract address: 0xc0226a9ad5535406250cb26727dfc28b4c58a634
// https://sepolia.etherscan.io/address/0xc0226a9ad5535406250cb26727dfc28b4c58a634

contract PiggyBank {
    address public immutable owner;
    uint public savingsGoal;
    mapping(address<uint>) public deposits;
    // Set any other variables you need

    // Set up so that the owner is the person who deployed the contract.
    // Set a savings goal
    constructor() {
        owner = msg.sender;
        savingsGoal = 0.1 ether;
    }

    // Create an event to emit once you reach the savings goal
    event SavingsGoalReached(address indexed depositor, uint amount);

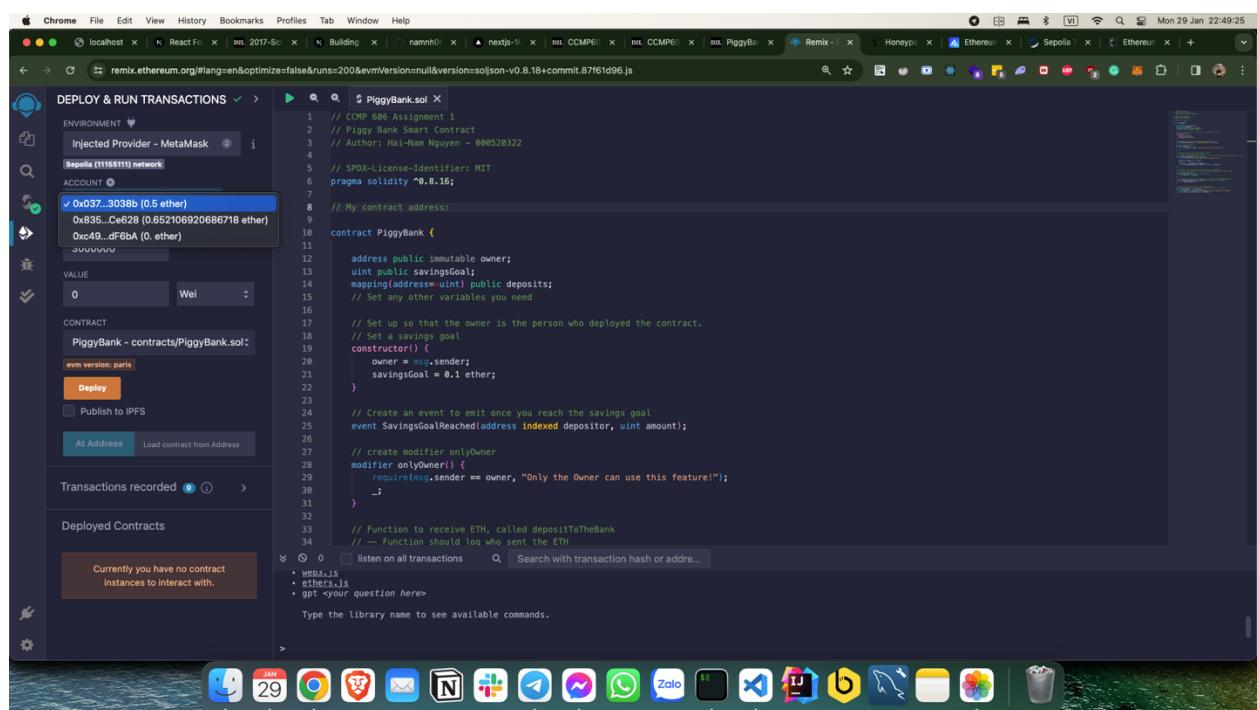
    // create modifier onlyOwner
    modifier onlyOwner() {
        require(msg.sender == owner, "Only the Owner can use this feature!");
    }
}

// Function to receive ETH, called deposit@theBank
// -- Function should log who sent the ETH

```

Deploy:

First, check the Injected Provider – MetaMask account



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.16;

// My contract address: 0x895..Ce626 (0.652106920886718 ether)
// 0x49..df6Ba (0. ether)
3000000

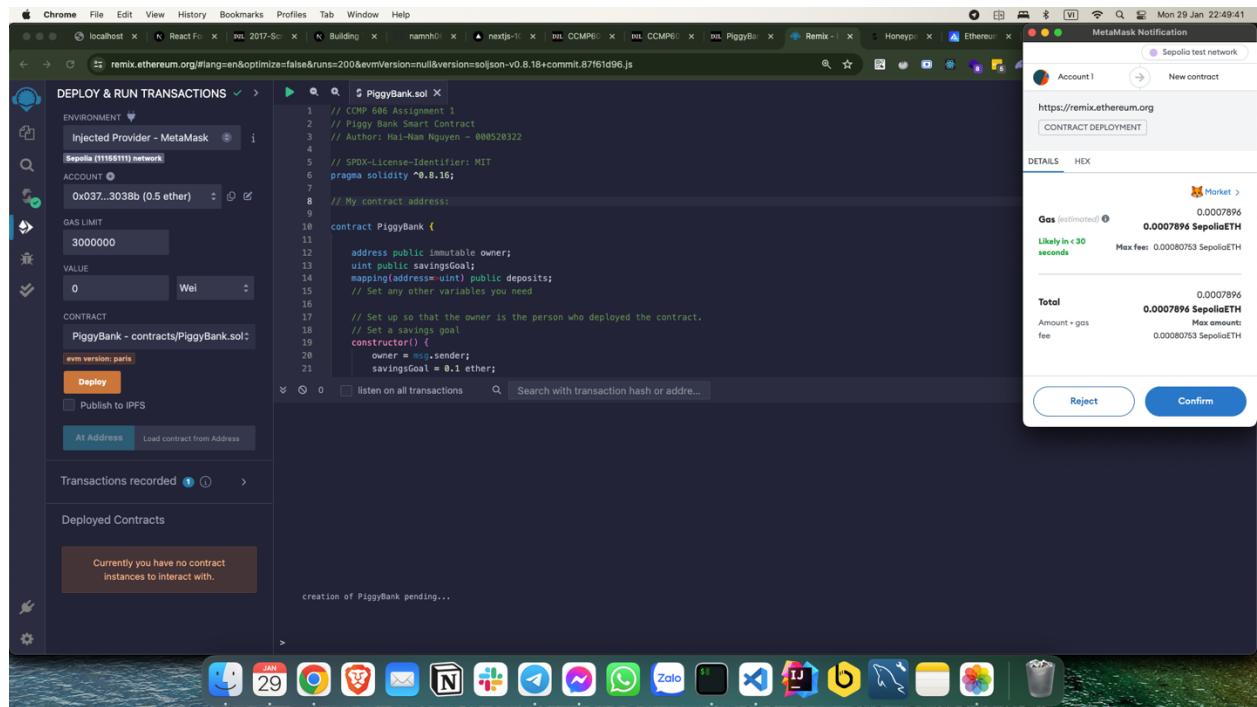
contract PiggyBank {
    address public immutable owner;
    uint public savingsGoal;
    mapping(address<uint>) public deposits;
    // Set any other variables you need

    // Set up so that the owner is the person who deployed the contract.
    // Set a savings goal
    constructor() {
        owner = msg.sender;
        savingsGoal = 0.1 ether;
    }

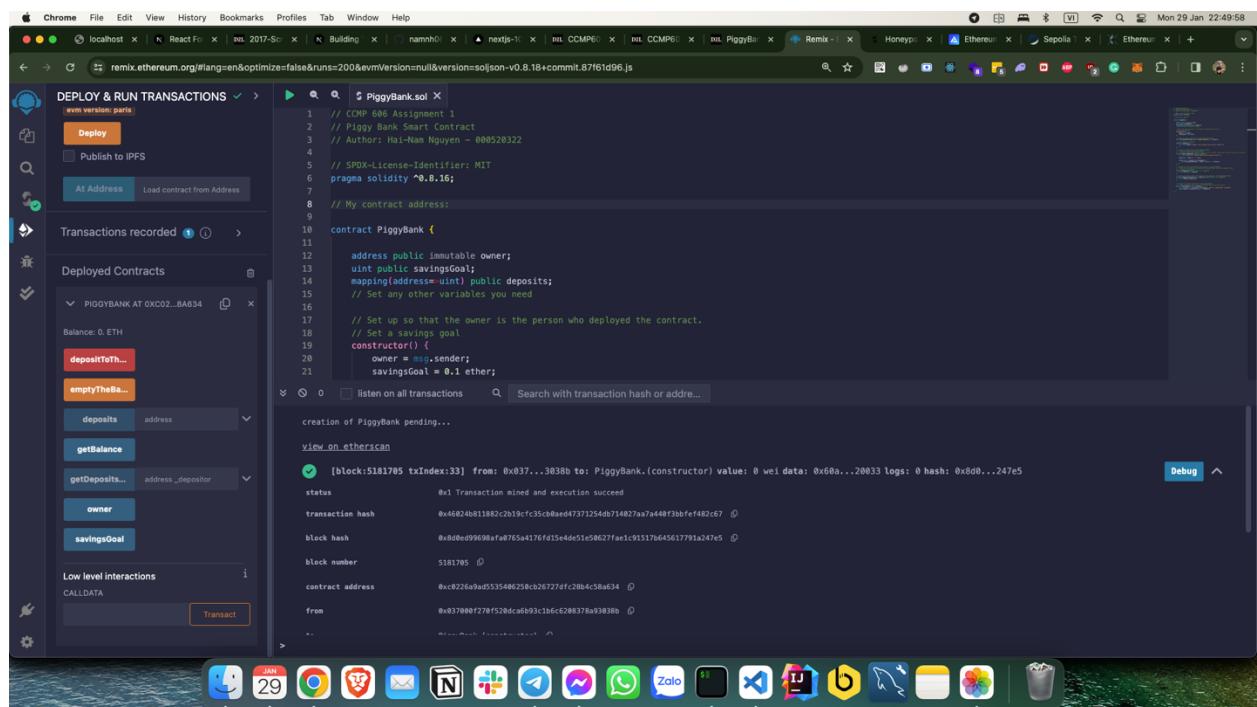
    // Create an event to emit once you reach the savings goal
    event SavingsGoalReached(address indexed depositor, uint amount);

    // create modifier onlyOwner
    modifier onlyOwner() {
        require(msg.sender == owner, "Only the Owner can use this feature!");
    }
}
```

Secondly, click Deploying:



Lastly, Confirm to see the deploy success:



Double check on Sepolia etherscan:

Transaction Details

Overview State

[This is a Sepolia Testnet transaction only]

Transaction Hash: 0x46024b811882c2b19cf35cb0aed47371254db714027ea7a440f3bbfe482c67

Status: Success

Block: 5181705 | 1 Block Confirmation

Timestamp: 29 secs ago (Jan-30-2024 04:49:48 AM +UTC)

Transaction Action: Call 0x037000f270F520DCA6b93C1B6C6208378A93038b Method by 0x037000...8A93038b

From: 0x037000f270F520DCA6b93C1B6C6208378A93038b

To: 0xc0226a9ad5535406250cb25727dfc28b4c58a634 Created

Value: 0 ETH (\$0.00)

Transaction Fee: 0.00077553800762256 ETH (\$0.00)

Gas Price: 1,591,781,04 Gwei (0.00000000159178104 ETH)

More Details: + Click to show more

II. Sepolia ETH on MetaMask

Get faucet from <https://sepoliafaucet.com/>

Transaction Details

Overview State

[This is a Sepolia Testnet transaction only]

Transaction Hash: 0x7311b83e407f36e046220cd7894471836e2bd910c3784b3199fd7a5f53f4b1ea

Status: Success

Block: 5180293 | 7 Block Confirmations

Timestamp: 1 min ago (Jan-30-2024 12:05:48 AM +UTC)

From: 0x03C352eA32DFBb757CCdf4b457E52daF6eCC21917

To: 0x037000f270F520DCA6b93C1B6C6208378A93038b

Value: 0.5 ETH (\$0.00)

Transaction Fee: 0.00013638577176 ETH (\$0.00)

Gas Price: 6,494,560,56 Gwei (0.00000000649456056 ETH)

More Details: + Click to show more

III. Smart Contract Address on the Etherscan Webpage

The address:

<https://sepolia.etherscan.io/address/0xc0226a9ad5535406250cb26727dfc28b4c58a634>

A screenshot of a Google Chrome browser window showing the Etherscan Sepolia Testnet transaction details for the address 0xc0226a9ad5535406250cb26727dfc28b4c58a634. The transaction hash is 0x46024b811882c2b19fcf35cb0aed47371254db714027aa7a440f3bbfef482c67. The status is Success, and it was included in block 5181705 with 1 block confirmation. The timestamp is 29 secs ago (Jan-30-2024 04:49:48 AM +UTC). The transaction action was a call to 0x037000f270F520DCA6b93C1B6C6208378A93038b, Method by 0x037000...8A93038b. The transaction originated from 0xc0226a9ad5535406250cb26727dfc28b4c58a634 and was created at 1.59178104 Gwei (0.00000000159178104 ETH). The transaction value was 0 ETH (\$0.00) and the fee was 0.00077553800762256 ETH (\$0.00).

IV. Contract Balance on Remix IDE

After deposit:

The screenshot shows the Remix IDE interface. In the top navigation bar, the URL is `remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=json-v0.8.18+commit.87f61d06.js`. The main area displays the `PiggyBank.sol` code:

```

1 // COMP 606 Assignment 1
2 // Piggy Bank Smart Contract
3 // Author: Mai-Nam Nguyen - 008520322
4
5 // SPDX-License-Identifier: MIT
6 pragma solidity ^0.8.16;

```

The sidebar on the left lists the deployed contract `PIGGYBANK AT 0x02...8A634`. The contract has a balance of 0.2 ETH. It includes several functions:

- `depositToTheBank()`
- `emptyTheBank()`
- `deposits`: Address 0x0370009270F520DCA1
- `getBalance()`
- `getDeposits()`: Address 0x0370009270F520DCA1
- `owner`: Address 0x0370009270F520DCA6B93 C1B9C6208378A90303B8
- `savingsGoal`: Address 0x0370009270F520DCA6B93 C1B9C6208378A90303B8

The transaction details show a deposit of 1000000000000000 wei to the bank. A red arrow points from the sidebar to the transaction logs, which include the event `SavingGoalReached`:

```

{
  "args": [
    {
      "name": "depositAddress",
      "value": "0x0370009270F520DCA1"
    },
    {
      "name": "event",
      "value": "SavingGoalReached"
    },
    {
      "name": "gasUsed",
      "value": "47761"
    },
    {
      "name": "gasPrice",
      "value": "0x0000000000000000000000000000000000000000000000000000000000000000"
    },
    {
      "name": "logIndex",
      "value": "0x0000000000000000000000000000000000000000000000000000000000000000"
    },
    {
      "name": "transactionIndex",
      "value": "0x0000000000000000000000000000000000000000000000000000000000000000"
    }
  ],
  "logIndex": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "transactionIndex": "0x0000000000000000000000000000000000000000000000000000000000000000"
}

```

Also on Sepolia Etherscan:

The screenshot shows the Sepolia Etherscan interface. The URL is `sepolia.etherscan.io/address/0xC0226A9aD5535406250cb26727Dfc28B4c58A634`. The page displays the following information:

- Contract**: `0xC0226A9aD5535406250cb26727Dfc28B4c58A634`
- ETH BALANCE**: 0.2 ETH
- CONTRACT CREATOR**: `0x037000...8A9303B8` at txn `0x46024b811882c2b1...`
- Multichain Info**: No addresses found

The **Transactions** tab is selected, showing the following table of recent transactions:

Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
<code>0x8667221293e1e526...</code>	<code>0xffffdf75f</code>	5181725	16 secs ago	<code>0x835EAE...d2cCe628</code>	<code>0x0226A...4c58A634</code>	0.1 ETH	0.0000774
<code>0xfe74aeeb5f7532492...</code>	<code>0xffffdf75f</code>	5181716	2 mins ago	<code>0x037000...8A9303B8</code>	<code>0xC0226A...4c58A634</code>	0.1 ETH	0.0000795
<code>0x46024b811882c2b1...</code>	<code>0x60a06040</code>	5181705	4 mins ago	<code>0x037000...8A9303B8</code>	Contract Creation	0 ETH	0.0007753

A red arrow points to the first transaction in the list.

V. Explanations and Show How the Functions Work

a. Function depositToTheBank

```
34  // Function to receive ETH, called depositToTheBank
35  // -- Function should log who sent the ETH
36  // -- Function should check balance to know if you've reached savings goal and emit the event if you have.
37  function depositToTheBank() external payable {
38      require(msg.value > 0, "Deposit amount must be greater than 0!");
39
40      deposits[msg.sender] += msg.value;
41
42  if(address(this).balance >= savingsGoal) {
43      emit SavingsGoalReached(msg.sender, address(this).balance);
44  }
45 }
```

Overview, this function check how much the ether will be sent to the contract, it must be greater than 0 otherwise the error log will display. After require check, the mapping deposits will store the address of who send ether and cumulative the value he/she sent. The last one is the If condition to check whether the balance of this contract is equal or larger than savingsGoal or not. If yes, the event SavingsGoalReached will be emitted.

b. Function getBalance

```
47  // Function to return the balance of the contract, called getBalance
48  // -- Note: you will need to use address(this).balance which returns the balance in Wei.
49  // -- 1 Eth = 1 * 10**18 Wei
50  function getBalance() public view returns (uint) {
51      return address(this).balance;
52  }
```

This function can be called by anyone and return the balance of this contract in wei.

c. Function getDepositsValue

```
54  // Function to look up how much any depositor has deposited, called getDepositsValue
55  function getDepositsValue(address _depositor) public view returns (uint) {
56      return deposits[_depositor];
57  }
```

This function receives an address and return the wei that this address sent to the contract if exists.

d. Function emptyTheBank

```

60      // Function to withdraw (send) ETH, called emptyTheBank
61      // -- Only the owner of the contract can withdraw the ETH from the contract
62  ↘  function emptyTheBank() public onlyOwner {
63      require(address(this).balance >= savingsGoal, "Have not reached goal to empty");
64      payable(owner).transfer(address(this).balance);
65  }
66 }
```

This function stricted by onlyOwner can call.

```

28      // create modifier onlyOwner
29  ↘  modifier onlyOwner() {
30      require(msg.sender == owner, "Only the Owner can use this feature!");
31      _;
32 }
```

It checks whether the balance of this contract is greater or equal the savingsGoal. If the balance matched condition, the ether in this contract will be transferred to the owner address.

After this function invoked, the contract address looks like this:

The screenshot shows a web browser displaying the Etherscan interface for a smart contract at address 0xC0226A9aD5535406250cb26727DFc28B4c58A634. The page includes a search bar, navigation links, and tabs for Overview, More Info, and Multichain Info. The Transactions tab is active, showing four recent transactions. The first three are transfers from the contract to another address (0x037000...8A93038b), while the fourth is a 'Contract Creation' event. The browser's status bar indicates it's running on a Mac OS X system.

Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
0xe4506c12ee3c9bc3...	0x0e505040	5181738	15 secs ago	0x037000...8A93038b	0xC0226A...4c58A634	0 ETH	0.00004819
0x8667221293e1e526...	0xffffd75f	5181725	2 mins ago	0x835EAE...d2cCe628	0xC0226A...4c58A634	0.1 ETH	0.0000774
0xfe74aeeb5f7532492...	0xffffd75f	5181716	4 mins ago	0x037000...8A93038b	0xC0226A...4c58A634	0.1 ETH	0.00007955
0x46024b811882c2b1...	0x60a06040	5181705	6 mins ago	0x037000...8A93038b	Contract Creation	0 ETH	0.00077553

Functions worked:

The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar titled "DEPLOY & RUN TRANSACTIONS" with tabs for "Transactions recorded" and "Deployed Contracts". Under "Deployed Contracts", a list of deployed contracts is shown, with "PIGGYBANK AT 0x02...634" expanded. It displays the balance (0.2 ETH), several functions (depositToTheBar, emptyTheBar, deposits, getBalance, getDeposits, owner, savingsGoal), and their corresponding function signatures. A red arrow points from the left sidebar towards the right-hand code editor area where the Solidity code for the PiggyBank contract is displayed.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.18+commit.87f61d96js;

// COMPILED: 0x835eaeda583bf356f17838609ce941dd2cce628
// Piggy Bank Smart Contract
// Author: Hai-Nam Nguyen - 0x0528322
// SPDX-License-Identifier: MIT
// pragma solidity ^0.8.18;
```

Below the code editor, the status bar shows the date as "JAN 29" and the time as "Mon 29 Jan 22:54:02".