Issue

I stepped into the first issue when trying to compile the EmployeeContract.sol file. I found out

```
/home/blubu/learning/it480/employeecontract/contracts/EmployeeContract.sol:57:77:
DeclarationError: Undeclared identifier. Did you mean "Employee" or "employees"?
        function getEmployees() public isEmployee(msg.sender) onlyHR
isEmployee(_employee) view returns (address[] memory) {
                                                         ^-------^
,/home/blubu/learning/it480/employeecontract/contracts/EmployeeContract.sol:76:86:
DeclarationError: Undeclared identifier. Did you mean "Employee" or "employees"?
        function getEmployeesInfo() public isEmployee(msg.sender) onlyHR view
isEmployee(_employee) returns (Employee[] memory) {
                                                         ^-------^
```

I checked the contract and found out that it is not necessary to check is the employee since we are not getting just one individual employee but a list. And another one is that we already know who is the sender through msg.sender.

Improvement

```
modifier isAdmin() {
  require(uint256(contracts[msg.sender].position) <= 3, "You are not
administrators");
  _;
}
```

Since we know uint casting will allow positive numbers, we can simply just compare whether the position is less than or equal to the third index which is MKT position.

```
modifier onlyFN() {
   require(contracts[msg.sender].position == Position.FINC, "You are not
authorize");
   _;
}
```

Since we are dealing with some of the finance authorization, I added onlyFN which is refer to only finance position can do.

```
modifier noReentrant() {
   require(!locked, "No re-entrancy");
   locked = true;
   _;
   locked = false;
}
```

This modifier helps to prevent parallel execution of function.

```
modifier sufficientBalance() {
   require(getGrandTotal() <= balance, "You do not have sufficient balance
for all employees");
   _;
}
```

Before sending the payroll to all the employees, I made sure to check whether the balance is enough first or not yet.

```
event SentbalanceFail(
   address _receiverAddress,
   uint _balance,
   uint _timestamp
);
event DeliverPayroll(
   bool _complete,
   uint totalAmount,
   uint _timestamp
);
event Withdrawbalance(
   address _receiverAddress,
   uint _balance,
   uint _timestamp);
```

These are the addition events that logging so that we can see more changes or alert when things fail.

```
function addNewEmployee(
   address _employee,
   string memory _name,
   uint _salary,
   uint _position
)
   public
   isEmployee(msg.sender)
   onlyHR
   validPosition(uint256(_position))
{
   contracts[_employee] = Employee(
      _name,
      _salary,
      default_approvals_restart,
      Position(_position));
   employees.push(_employee);
   _approveEmployeePosition(
      _employee,
      contracts[msg.sender].position);
}
```

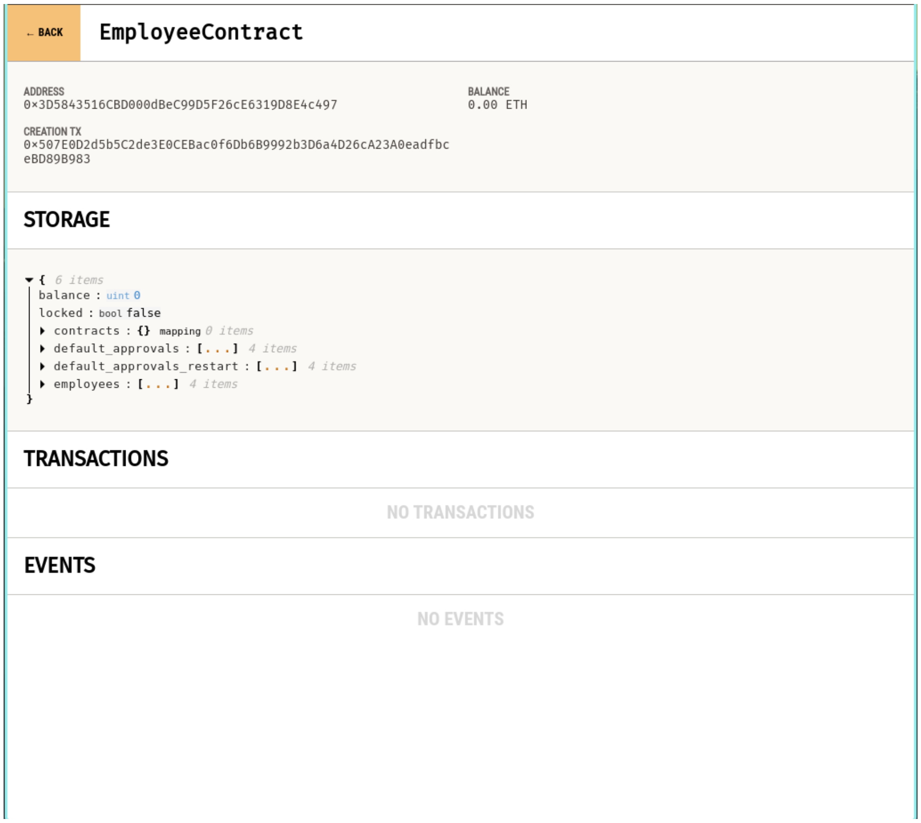
The improvement at this function was by eliminating one of the isEmployee(_employee) because _employee is not yet the employee, it doesn't make sense to addNewEmployee that is already employee.

```
function deposit() public payable {
      balance += msg.value;
      emit Depositbalance(msg.sender, msg.value, block.timestamp);
}


function getGrandTotal() public isEmployee(msg.sender) onlyFN view returns
(uint) {
      uint totalSalary = 0;
      for (uint i = 0; i < employees.length; i++) {
            totalSalary += contracts[employees[i]].salary;
      }
      return totalSalary;
}

function sendSalary(address payable _to, uint _salary) public payable {
      bool sentStatus = _to.send(_salary);
      if (!sentStatus) {
            emit SentbalanceFail(_to, _salary, block.timestamp);
      }
}

function sendPayroll() public payable isEmployee(msg.sender) onlyFN
sufficientBalance noReentrant {
      for (uint i = 0; i < employees.length; i++) {
            address payable acc = address(uint160(employees[i]));
            sendSalary(acc, contracts[employees[i]].salary);
            balance -= contracts[employees[i]].salary;
      }
      emit DeliverPayroll(true, getGrandTotal(), block.timestamp);
}
```

- Deposit function is added that is counted as be the place where the company generate the revenue, so this is where other address transfering the balance to the contract.
- Get Grand Total is another function to get all the employees's income, which we did through the iteration of for loop.
- Send payroll function is just sto iterate through the employee address and converting those address to payable one in which we can use to run the sendSalary.

Output

Testing Deposit function



EmployeeContract

ADDRESS
0×3D5843516CBD000dBeC99D5F26cE6319D8E4c497

BALANCE
0.00 ETH

CREATION TX
0×507E0D2d5b5C2de3E0CEBac0f6Db6B9992b3D6a4D26cA23A0eadfbc
eBD89B983

STORAGE

```
▾ { 6 items
  balance : uint 0
  locked : bool false
  ▸ contracts : {} mapping 0 items
  ▸ default_approvals : [...] 4 items
  ▸ default_approvals_restart : [...] 4 items
  ▸ employees : [...] 4 items
}
```

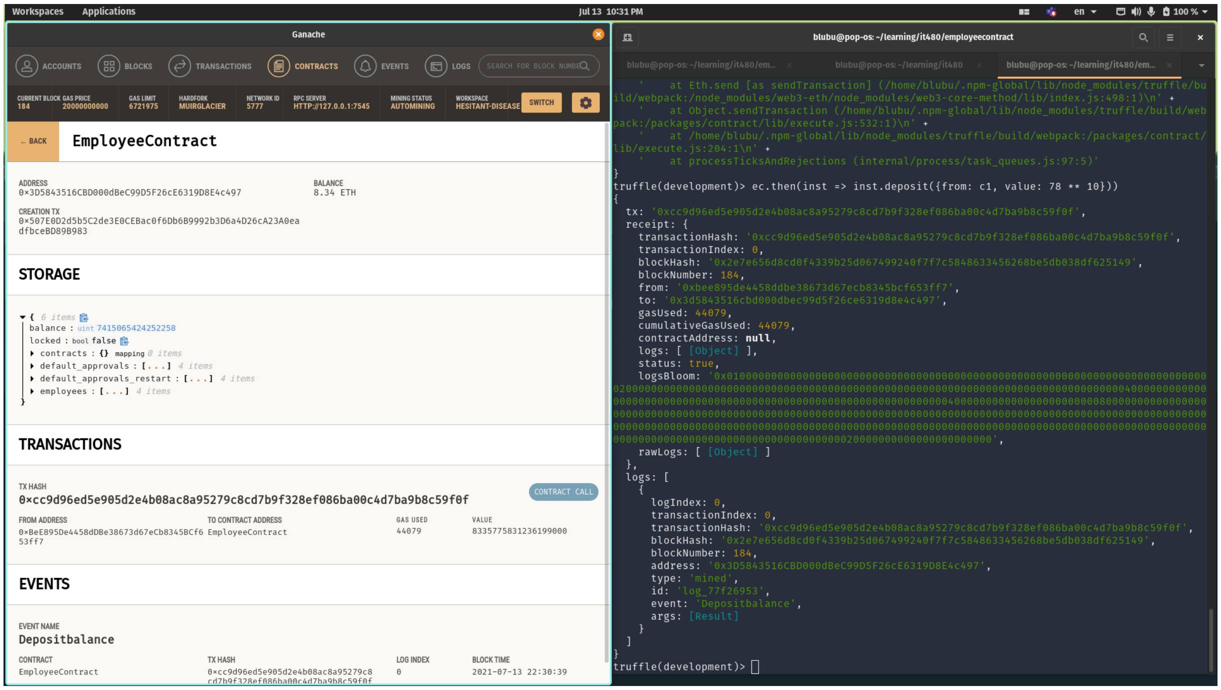
TRANSACTIONS

NO TRANSACTIONS

EVENTS

NO EVENTS

Contract state after migration. There is 0 balance right now.


Workspaces
Applications
Jul 13 10:31 PM
en
100 %
Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBE
SWITCH
← BACK
CONTRACT CALL
blubu@pop-os: ~/learning/it480/employeecontract
blubu@pop-os: ~/learning/it480/em...
blubu@pop-os: ~/learning/it480
blubu@pop-os: ~/learning/it480/em...

CURRENT BLOCK 184 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MUIRGLACIER | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE HESITANT-DISEASE

EmployeeContract

ADDRESS
0×3D5843516CBD000dBeC99D5F26cE6319D8E4c497

BALANCE
8.34 ETH

CREATION TX
0×507E0D2d5b5C2de3E0CEBac0f6Db6B9992b3D6a4D26cA23A0ea
dfbceBD89B983

STORAGE

```
▾ { 6 items
  balance : uint 7415065424252258
  locked : bool false
  ▸ contracts : {} mapping 0 items
  ▸ default_approvals : [...] 4 items
  ▸ default_approvals_restart : [...] 4 items
  ▸ employees : [...] 4 items
}
```

TRANSACTIONS

TX HASH
0×cc9d96ed5e905d2e4b08ac8a95279c8cd7b9f328ef086ba00c4d7ba9b8c59f0f

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0×BeE895De4458dDBe38673d67eCb8345BCf653ff7	EmployeeContract	44079	8335775831236199000

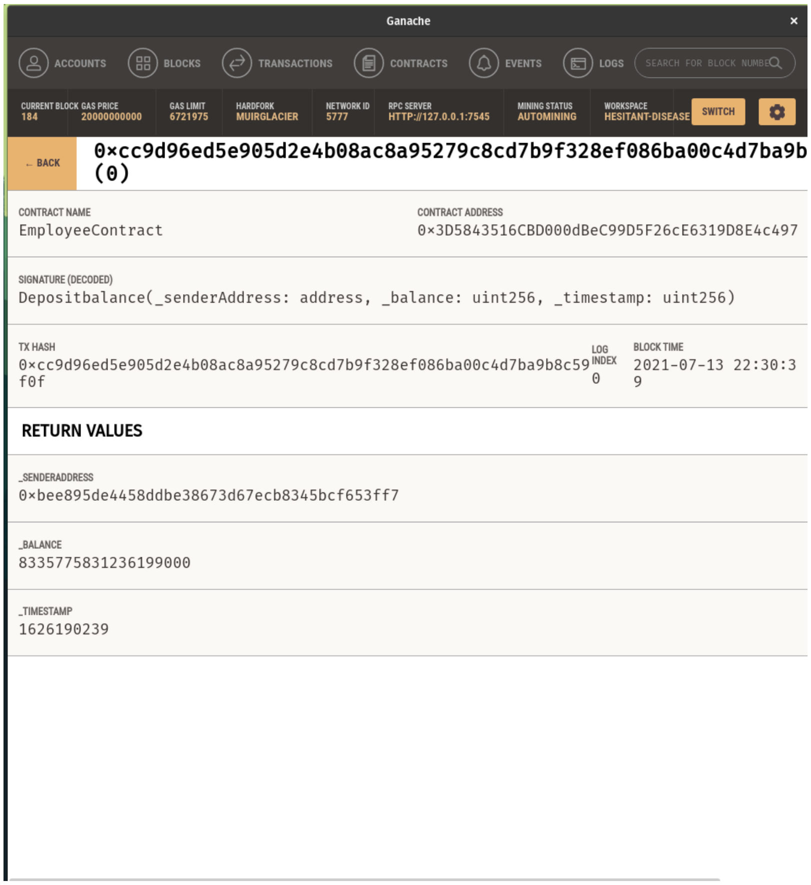
EVENTS

EVENT NAME
Depositbalance

CONTRACT	TX HASH	LOG INDEX	BLOCK TIME
EmployeeContract	0×cc9d96ed5e905d2e4b08ac8a95279c8cd7b9f328ef086ba00c4d7ba9b8c59f0f	0	2021-07-13 22:30:39

```
  '    at Eth.send [as sendTransaction] (/home/blubu/.npm-global/lib/node_modules/truffle/bu
ild/webpack:/node_modules/web3-eth/node_modules/web3-core-method/lib/index.js:498:1)\n' +
  '    at Object.sendTransaction (/home/blubu/.npm-global/lib/node_modules/truffle/build/web
pack:/packages/contract/lib/execute.js:532:1)\n' +
  '    at /home/blubu/.npm-global/lib/node_modules/truffle/build/webpack:/packages/contract/
lib/execute.js:204:1\n' +
  '    at processTicksAndRejections (internal/process/task_queues.js:97:5)'
}
truffle(development)> ec.then(inst => inst.deposit({from: c1, value: 78 ** 10}))
{
  tx: '0xcc9d96ed5e905d2e4b08ac8a95279c8cd7b9f328ef086ba00c4d7ba9b8c59f0f',
  receipt: {
    transactionHash: '0xcc9d96ed5e905d2e4b08ac8a95279c8cd7b9f328ef086ba00c4d7ba9b8c59f0f',
    transactionIndex: 0,
    blockHash: '0x2e7e656d8cd0f4339b25d067499240f7f7c5848633456268be5db038df625149',
    blockNumber: 184,
    from: '0xbee895de4458ddbe38673d67ecb8345bcf653ff7',
    to: '0x3d5843516cbd000dbec99d5f26ce6319d8e4c497',
    gasUsed: 44079,
    cumulativeGasUsed: 44079,
    contractAddress: null,
    logs: [ [Object] ],
    status: true,
    logsBloom: '0x0100000000000000000000000000000000000000000000000000000000000000000000000000
0200000000000000000000000000000000000000000000000000000000000000000000000000000040000000000000
0000000000000000000000000000000000000000000000000000040000000000000000000000000008000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000200000000000000000000000',
    rawLogs: [ [Object] ]
  },
  logs: [
    {
      logIndex: 0,
      transactionIndex: 0,
      transactionHash: '0xcc9d96ed5e905d2e4b08ac8a95279c8cd7b9f328ef086ba00c4d7ba9b8c59f0f',
      blockHash: '0x2e7e656d8cd0f4339b25d067499240f7f7c5848633456268be5db038df625149',
      blockNumber: 184,
      address: '0x3D5843516CBD000dBeC99D5F26cE6319D8E4c497',
      type: 'mined',
      id: 'log_77f26953',
      event: 'Depositbalance',
      args: [Result]
    }
  ]
}
truffle(development)> 
```

Contract state after using deposit function. Notice c1 refer to customer 1, in which is another account address.



CURRENT BLOCK 184 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MUIRGLACIER | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE HESITANT-DISEASE

0×cc9d96ed5e905d2e4b08ac8a95279c8cd7b9f328ef086ba00c4d7ba9b
(0)

CONTRACT NAME
EmployeeContract

CONTRACT ADDRESS
0×3D5843516CBD000dBeC99D5F26cE6319D8E4c497

SIGNATURE (DECODED)
Depositbalance(_senderAddress: address, _balance: uint256, _timestamp: uint256)

TX HASH	LOG INDEX	BLOCK TIME
0×cc9d96ed5e905d2e4b08ac8a95279c8cd7b9f328ef086ba00c4d7ba9b8c59f0f	0	2021-07-13 22:30:39

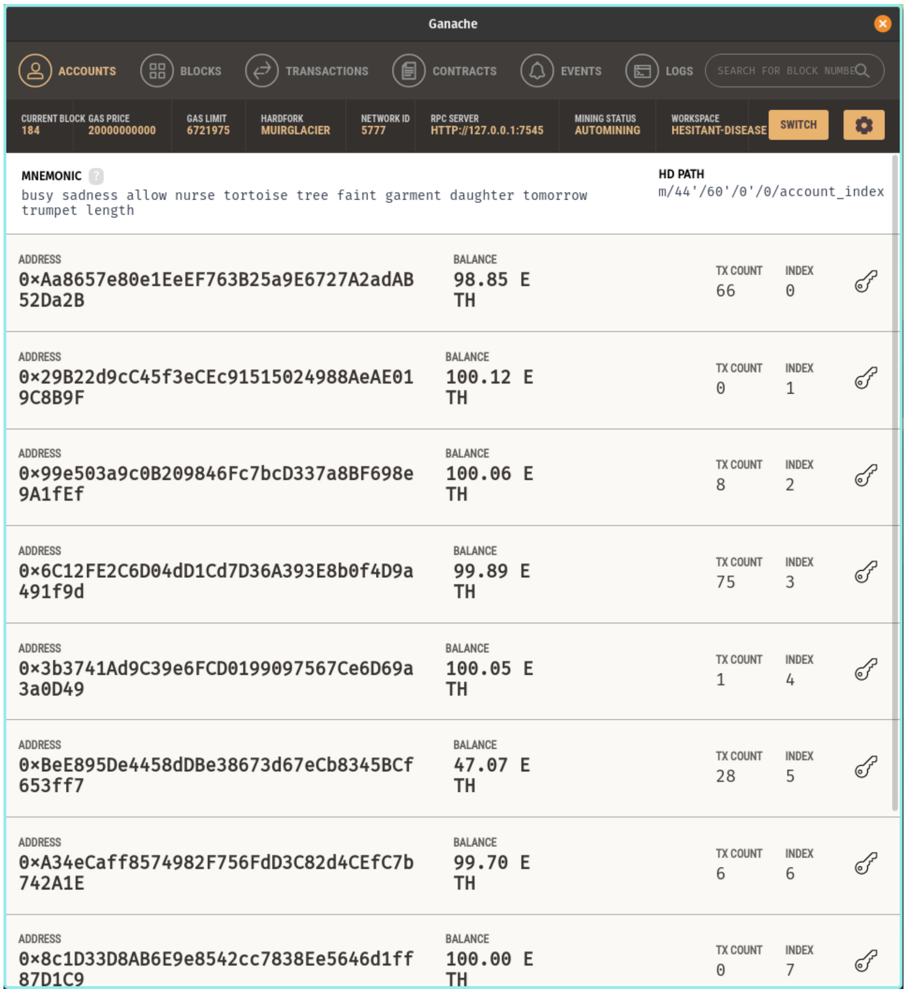
RETURN VALUES

_SENDERADDRESS
0×bee895de4458ddbe38673d67ecb8345bcf653ff7

_BALANCE
8335775831236199000

_TIMESTAMP
1626190239

This is an deposit event that is trigger when we call the deposit.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
184	20000000000	6721975	MUIRGLACIER	5777	HTTP://127.0.0.1:7545	AUTOMINING	HESITANT-DISEASE

MNEMONIC
busy sadness allow nurse tortoise tree faint garment daughter tomorrow trumpet length

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×Aa8657e80e1EeEF763B25a9E6727A2adAB52Da2B	98.85 ETH	66	0
0×29B22d9cC45f3eCEc91515024988AeAE019C8B9F	100.12 ETH	0	1
0×99e503a9c0B209846Fc7bcD337a8BF698e9A1fEf	100.06 ETH	8	2
0×6C12FE2C6D04dD1Cd7D36A393E8b0f4D9a491f9d	99.89 ETH	75	3
0×3b3741Ad9C39e6FCD0199097567Ce6D69a3a0D49	100.05 ETH	1	4
0×BeE895De4458dDBe38673d67eCb8345BCf653ff7	47.07 ETH	28	5
0×A34eCaff8574982F756FdD3C82d4CEfC7b742A1E	99.70 ETH	6	6
0×8c1D33D8AB6E9e8542cc7838Ee5646d1ff87D1C9	100.00 ETH	0	7

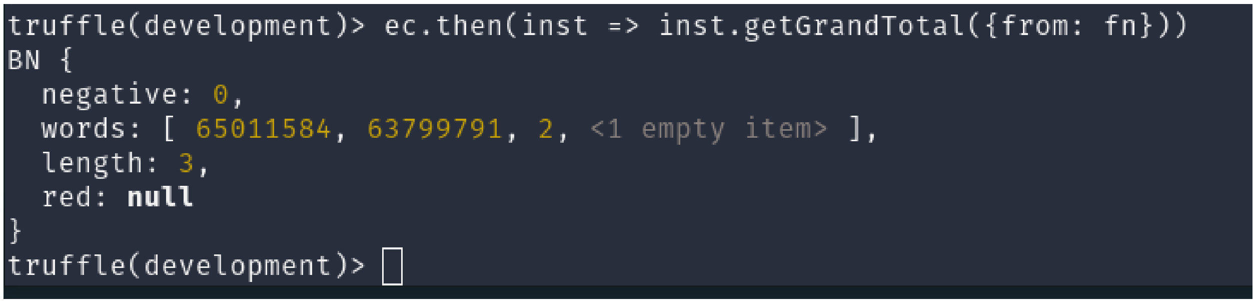
The accounts addresses are arrange by:
Address[0] refers to contract deployer
Address[1] refers to CEO
Address[2] refers to HR
Address[3] refer to FINC
Address[4] refer to MKT
Address[5] refers to customer 1
Address[6] refers to customer 2

We declare some short variables under the truffle console,
C1 refers to customer 1
C2 refers to customer 2
hr refers to HR
fn refers to FINC

Testing get total salary function

```
var EmployeeContract = artifacts.require('./EmployeeContract.sol');

module.exports = function(deployer) {
  deployer.deploy(EmployeeContract, [
      "0x29B22d9cC45f3eCEc91515024988AeAE019C8B9F",
      "0x99e503a9c0B209846Fc7bcD337a8BF698e9A1fEf",
      "0x6C12FE2C6D04dD1Cd7D36A393E8b0f4D9a491f9d",
      "0x3b3741Ad9C39e6FCD0199097567Ce6D69a3a0D49"
      ], [
      "Bro Set",
      "Nao",
      "Seren",
      "Sumi"
      ], [
      5192389813200000,
      3123123811200000,
      2981923192900000,
      1991293999900000
      ]);
}
```

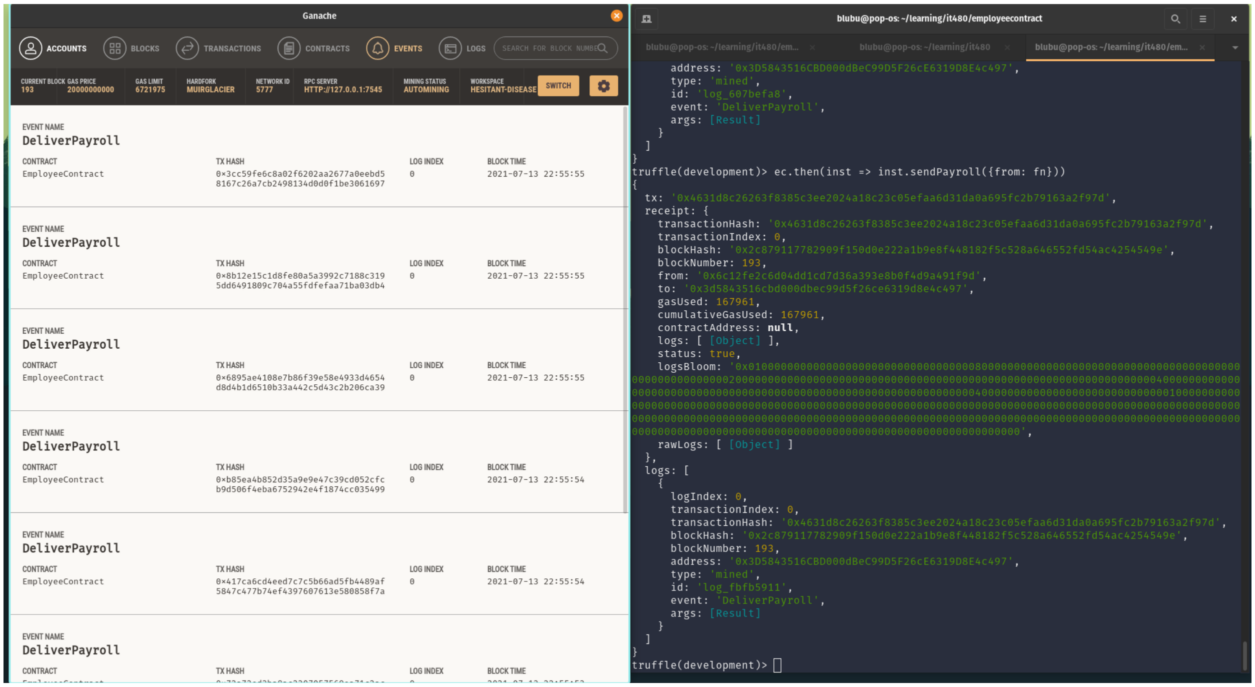
This is our migration files, the last array refers to the salary list for each position.



```
truffle(development)> ec.then(inst => inst.getGrandTotal({from: fn}))
BN {
  negative: 0,
  words: [ 65011584, 63799791, 2, <1 empty item> ],
  length: 3,
  red: null
}
truffle(development)> 
```

So we run get total function with finance account, this is the result that we got.

Testing Sending payroll


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBE
SWITCH
blubu@pop-os: ~/learning/it480/employeecontract
blubu@pop-os: ~/learning/it480/em...
blubu@pop-os: ~/learning/it480
blubu@pop-os: ~/learning/it480/em...

CURRENT BLOCK 193 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MUIRGLACIER | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE HESITANT-DISEASE

EVENT NAME
DeliverPayroll
CONTRACT EmployeeContract
TX HASH 0x3cc59fe6c8a02f6202aa2677a0eebd58167c26a7cb2498134d0d0f1be3061697
LOG INDEX 0
BLOCK TIME 2021-07-13 22:55:55

EVENT NAME
DeliverPayroll
CONTRACT EmployeeContract
TX HASH 0x8b12e15c1d8fe80a5a3992c7188c3195dd6491809c704a55fdfefaa71ba03db4
LOG INDEX 0
BLOCK TIME 2021-07-13 22:55:55

EVENT NAME
DeliverPayroll
CONTRACT EmployeeContract
TX HASH 0x6895ae4108e7b86f39e58e4933d4654d8d4b1d6510b33a442c5d43c2b206ca39
LOG INDEX 0
BLOCK TIME 2021-07-13 22:55:55

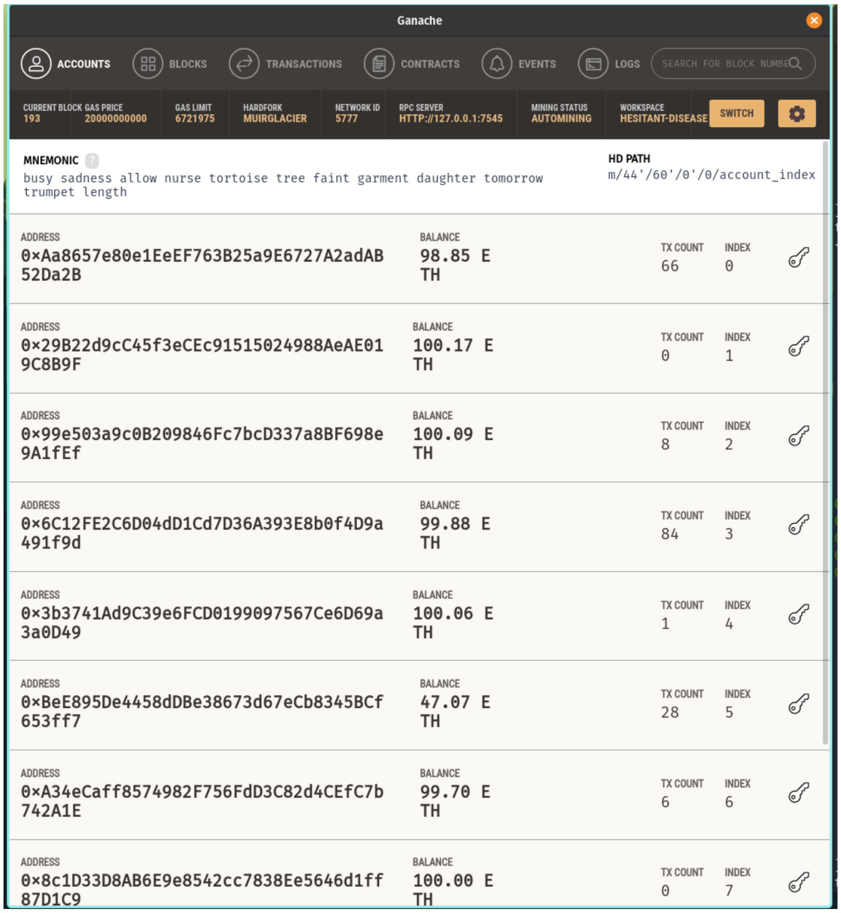
EVENT NAME
DeliverPayroll
CONTRACT EmployeeContract
TX HASH 0xb85ea4b852d35a9e9e47c39cd052cfcb9d506f4eba6752942e4f1874cc035499
LOG INDEX 0
BLOCK TIME 2021-07-13 22:55:54

EVENT NAME
DeliverPayroll
CONTRACT EmployeeContract
TX HASH 0x417ca6cd4eed7c7c5b66ad5fb4489af5847c477b74ef4397607613e580858f7a
LOG INDEX 0
BLOCK TIME 2021-07-13 22:55:54

EVENT NAME
DeliverPayroll
CONTRACT
TX HASH
LOG INDEX
BLOCK TIME

```
      address: '0x3D5843516CBD000dBeC99D5F26cE6319D8E4c497',
      type: 'mined',
      id: 'log_607befa8',
      event: 'DeliverPayroll',
      args: [Result]
    }
  ]
}
truffle(development)> ec.then(inst => inst.sendPayroll({from: fn}))
{
  tx: '0x4631d8c26263f8385c3ee2024a18c23c05efaa6d31da0a695fc2b79163a2f97d',
  receipt: {
    transactionHash: '0x4631d8c26263f8385c3ee2024a18c23c05efaa6d31da0a695fc2b79163a2f97d',
    transactionIndex: 0,
    blockHash: '0x2c879117782909f150d0e222a1b9e8f448182f5c528a646552fd54ac4254549e',
    blockNumber: 193,
    from: '0x6c12fe2c6d04dd1cd7d36a393e8b0f4d9a491f9d',
    to: '0x3d5843516cbd000dbec99d5f26ce6319d8e4c497',
    gasUsed: 167961,
    cumulativeGasUsed: 167961,
    contractAddress: null,
    logs: [ [Object] ],
    status: true,
    logsBloom: '0x010000000000000000000000000000000008000000000000000000000000000000000000000000
0000000000000002000000000000000000000000000000000000000000000000000000000000000004000000000000
0000000000000000000000000000000000000000000000000000040000000000000000000000000000010000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000',
    rawLogs: [ [Object] ]
  },
  logs: [
    {
      logIndex: 0,
      transactionIndex: 0,
      transactionHash: '0x4631d8c26263f8385c3ee2024a18c23c05efaa6d31da0a695fc2b79163a2f97d',
      blockHash: '0x2c879117782909f150d0e222a1b9e8f448182f5c528a646552fd54ac4254549e',
      blockNumber: 193,
      address: '0x3D5843516CBD000dBeC99D5F26cE6319D8E4c497',
      type: 'mined',
      id: 'log_fbfb5911',
      event: 'DeliverPayroll',
      args: [Result]
    }
  ]
}
truffle(development)> 
```

After running sending payroll for multiple times, we see the event is log appropriately.



CURRENT BLOCK 193 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MUIRGLACIER | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE HESITANT-DISEASE

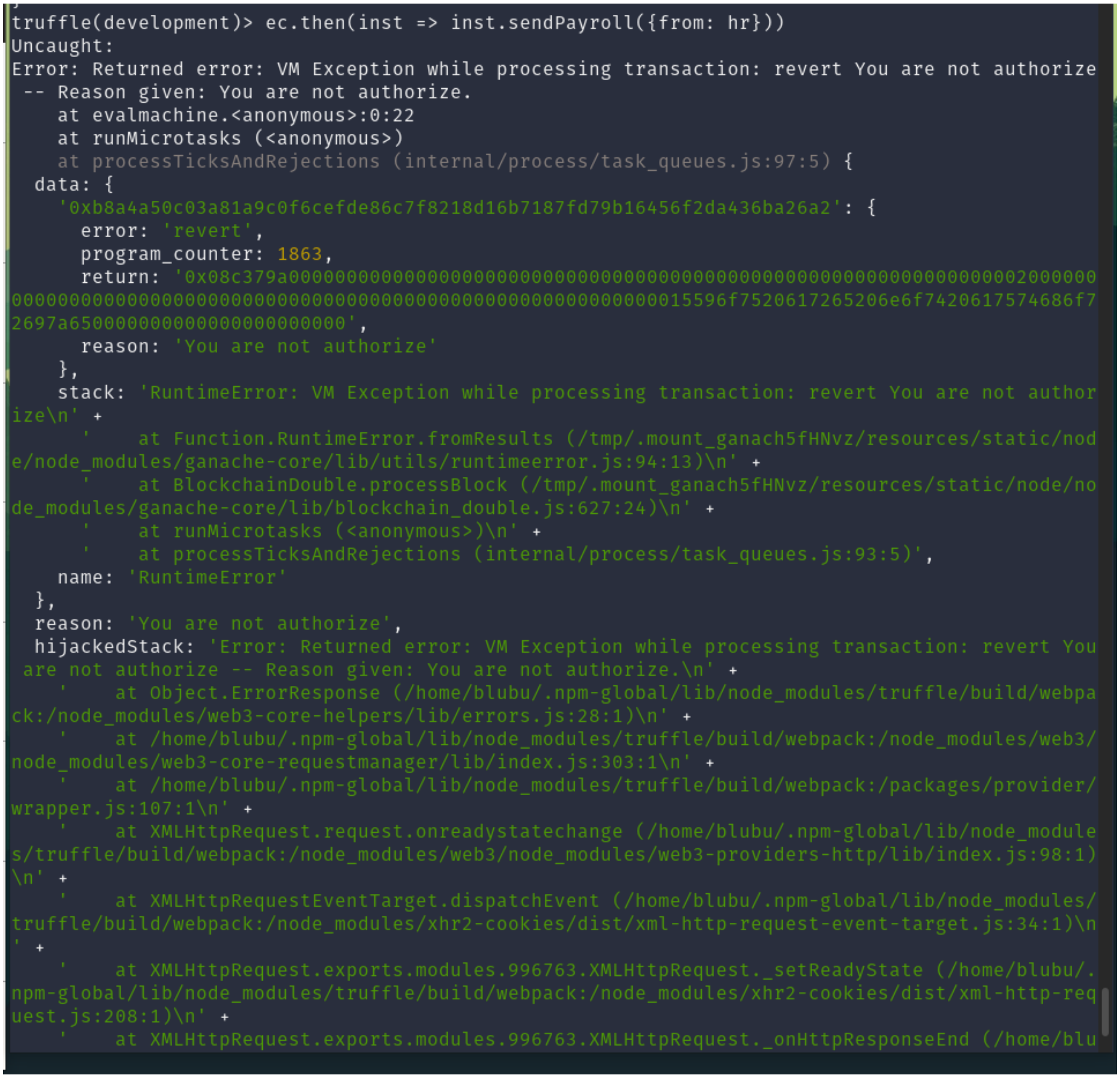
MNEMONIC
busy sadness allow nurse tortoise tree faint garment daughter tomorrow trumpet length

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0xAa8657e80e1EeEF763B25a9E6727A2adAB52Da2B	98.85 ETH	66	0
0x29B22d9cC45f3eCEc91515024988AeAE019C8B9F	100.17 ETH	0	1
0x99e503a9c0B209846Fc7bcD337a8BF698e9A1fEf	100.09 ETH	8	2
0x6C12FE2C6D04dD1Cd7D36A393E8b0f4D9a491f9d	99.88 ETH	84	3
0x3b3741Ad9C39e6FCD0199097567Ce6D69a3a0D49	100.06 ETH	1	4
0xBeE895De4458dDBe38673d67eCb8345BCf653ff7	47.07 ETH	28	5
0xA34eCaff8574982F756FdD3C82d4CEfC7b742A1E	99.70 ETH	6	6
0x8c1D33D8AB6E9e8542cc7838Ee5646d1ff87D1C9	100.00 ETH	0	7

Notice the account’s balance also change (look at the second account until the fifth one)

Testing whether HR can run sendingPayroll or not



```
truffle(development)> ec.then(inst => inst.sendPayroll({from: hr}))
Uncaught:
Error: Returned error: VM Exception while processing transaction: revert You are not authorize
 -- Reason given: You are not authorize.
    at evalmachine.<anonymous>:0:22
    at runMicrotasks (<anonymous>)
    at processTicksAndRejections (internal/process/task_queues.js:97:5) {
  data: {
    '0xb8a4a50c03a81a9c0f6cefde86c7f8218d16b7187fd79b16456f2da436ba26a2': {
      error: 'revert',
      program_counter: 1863,
      return: '0x08c379a000000000000000000000000000000000000000000000000000000000000000002000000
000000000000000000000000000000000000000000000000000015596f7520617265206e6f7420617574686f7
2697a650000000000000000000000',
      reason: 'You are not authorize'
    },
    stack: 'RuntimeError: VM Exception while processing transaction: revert You are not author
ize\n' +
      '    at Function.RuntimeError.fromResults (/tmp/.mount_ganach5fHNvz/resources/static/nod
e/node_modules/ganache-core/lib/utils/runtimeerror.js:94:13)\n' +
      '    at BlockchainDouble.processBlock (/tmp/.mount_ganach5fHNvz/resources/static/node/no
de_modules/ganache-core/lib/blockchain_double.js:627:24)\n' +
      '    at runMicrotasks (<anonymous>)\n' +
      '    at processTicksAndRejections (internal/process/task_queues.js:93:5)',
    name: 'RuntimeError'
  },
  reason: 'You are not authorize',
  hijackedStack: 'Error: Returned error: VM Exception while processing transaction: revert You
 are not authorize -- Reason given: You are not authorize.\n' +
    '    at Object.ErrorResponse (/home/blubu/.npm-global/lib/node_modules/truffle/build/webpa
ck:/node_modules/web3-core-helpers/lib/errors.js:28:1)\n' +
    '    at /home/blubu/.npm-global/lib/node_modules/truffle/build/webpack:/node_modules/web3/
node_modules/web3-core-requestmanager/lib/index.js:303:1\n' +
    '    at /home/blubu/.npm-global/lib/node_modules/truffle/build/webpack:/packages/provider/
wrapper.js:107:1\n' +
    '    at XMLHttpRequest.request.onreadystatechange (/home/blubu/.npm-global/lib/node_module
s/truffle/build/webpack:/node_modules/web3/node_modules/web3-providers-http/lib/index.js:98:1)
\n' +
    '    at XMLHttpRequestEventTarget.dispatchEvent (/home/blubu/.npm-global/lib/node_modules/
truffle/build/webpack:/node_modules/xhr2-cookies/dist/xml-http-request-event-target.js:34:1)\n
' +
    '    at XMLHttpRequest.exports.modules.996763.XMLHttpRequest._setReadyState (/home/blubu/.
npm-global/lib/node_modules/truffle/build/webpack:/node_modules/xhr2-cookies/dist/xml-http-req
uest.js:208:1)\n' +
    '    at XMLHttpRequest.exports.modules.996763.XMLHttpRequest._onHttpResponseEnd (/home/blu
```

As expected that only FN can run.