In [ ]:

```
//SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.6.0;

contract bank {
    struct client{
        uint id;
        address ads;
    }

    uint count;
    client[] clients;

    constructor() public {
        count = 0;
    }

    receive() external payable {

    }

    modifier onlyClient() {
        bool isClient = false;

        for(uint i=0; i<clients.length; i++) {
            if(clients[i].ads == msg.sender) {
                isClient = true;
                break;
            }
        }

        require(isClient);
        _;
    }

    function join_as_client() public payable {
        clients.push(client(count++, msg.sender));
    }

    function deposit() public payable onlyClient {
        address(this).transfer(msg.value);
    }

    function withdraw(uint v) public payable onlyClient {
        msg.sender.transfer(v * 1 ether);
    }

    function get_ether_value() public view returns(uint) {
        return address(this).balance;
    }

    function get_client_balance() public view returns(uint) {
        return address(msg.sender).balance;
    }
}
```

In [ ]:

```
//SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.6.0;

contract student_management {
    struct student {
        uint id;
        string name;
        string department;
    }

    student[] students;

    function add_data(uint id, string memory name, string memory department) public {
        student memory stud = student(id, name, department);
        students.push(stud);
    }

    function get_data(uint id) public view returns(string memory name, string memory depart
        for(uint i=0; i<students.length; i++) {
            student memory stud = students[i];
            if(id == stud.id) {
                return(stud.name, stud.department);
            }
        }
        return("Not Found", "Not Found");
    }
}
```