Peer-to-Peer Borrowing and Lending

FINAL EXAM

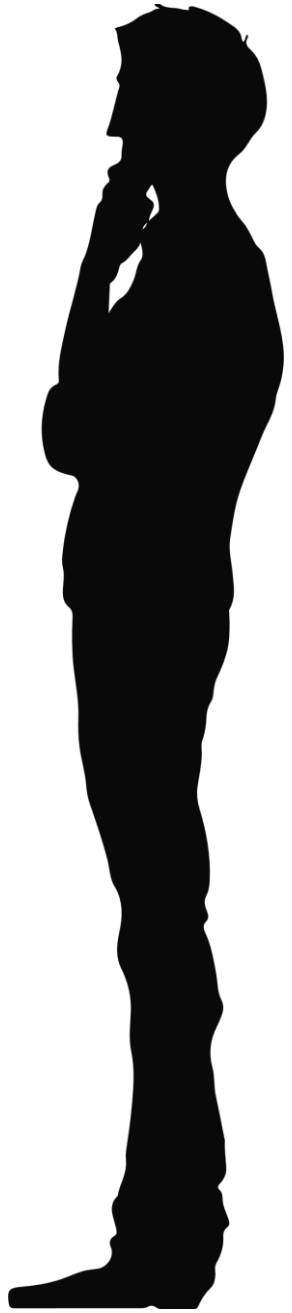
Our Team

1. Weha Kohprasert #9
Student ID: 2020731303004

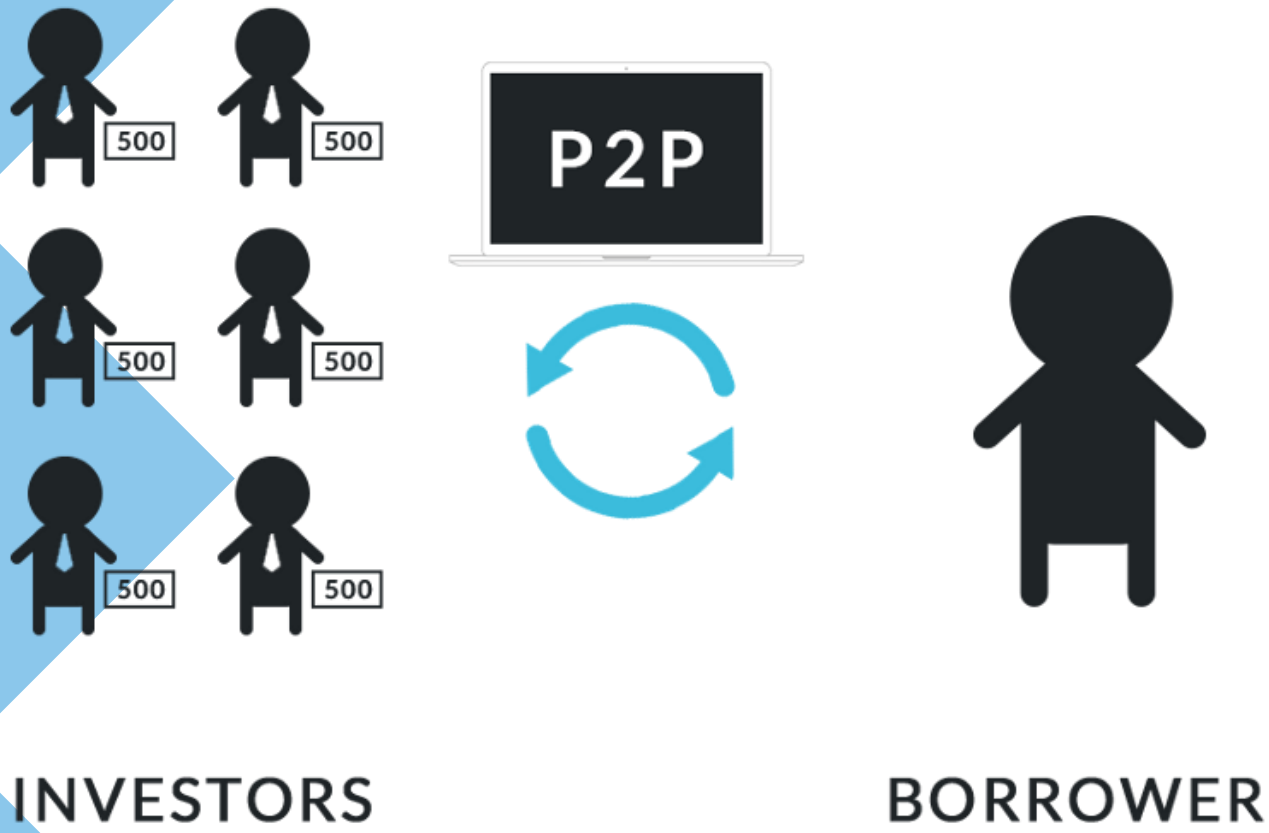
2. Pittaya Kunnawat #9
Student ID: 2020731303005

3. Chawakorn Amatawanichkul #9
Student ID: 2020731303007

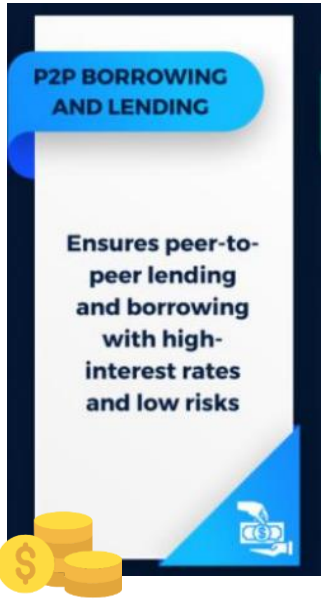
4. Cherdphong Vanijyananda #10
Student ID: 2030731303003



Peer-to-Peer Lending หรือ P2P Lending



คือ การทำธุรกรรมสินเชื่อโดยตรงระหว่างผู้กู้และผู้ให้กู้ ผ่านผู้ให้บริการสินเชื่อระหว่างบุคคลกับบุคคลผ่านระบบหรือเครือข่ายอิเล็กทรอนิกส์ (P2P Platform) โดยไม่ผ่านสถาบันการเงิน การให้บริการสินเชื่อP2P สามารถลดต้นทุนการดำเนินธุรกิจเช่นการสร้างเครือข่ายสาขาทั่วประเทศ ค่าใช้จ่ายบุคลากรจำนวนมาก และค่าใช้จ่ายการบริหารงานลงได้ รูปแบบการให้บริการสินเชื่อ P2P ผ่านระบบหรือเครือข่ายอิเล็กทรอนิกส์ เน้นการดำเนินธุรกิจด้วยต้นทุนที่ต่ำ ทำให้ผู้กู้ได้รับอัตราดอกเบี้ยต่ำในขณะที่ผู้ให้กู้ได้รับผลตอบแทนที่คุ้มค่า เป็นทางเลือกในการลงทุนเทียบกับการลงทุนในเงินฝาก ตราสารหนี้ และกองทุน



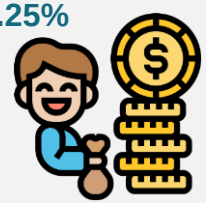
Interest Return

Bank (Centralize)
0.25% - 2.00%

P2P
(BLOCKCHAIN)
2.00% - 5.00%

P2P LENDING CONCEPT

Traditional Lending



Interest return
0.25%

Saving

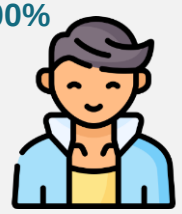
Total Cost
5.00%

- Operation Cost
- Money Reserve
- Risk Cost

Verify

Money Transfer

Interest pay
7.00%

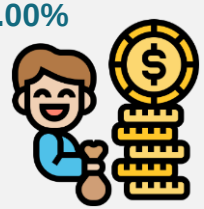


Borrower

Peer to Peer Lending (P2P)

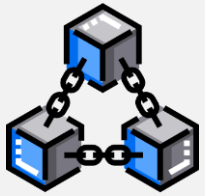
Money Transfer

Interest return
4.00%



Saving

Total Cost
1.00%



Verify via Smart Contact

Interest pay 5.00%

Borrower

Solidity Code

Select version 5.17

```
contract peertopeer{
    mapping (address => uint256)  balance;
    mapping (address => uint256)  balance2;
    mapping (address => uint256)  datecheck;
    mapping (address => uint256)  rateallow;
    mapping (address => uint256)  rateallow2;
    mapping (address => uint256)  amountallow;
    mapping (address => uint256)  amountallow2;
    mapping (address => address)  addressallow;
    mapping (address => address)  addressallow2;
    mapping (address => address)  transationhappen;
    mapping (address => uint256)  timestamp1;
    address[] account;
    address public owner = msg.sender;
    // interest rate;
    uint256 public rate = 5;
```

สร้างตัวแปรสำหรับการกู้ยืมของระบบ

```
 constructor() public{
      owner =  msg.sender;
    }

    function depositforlending() public payable
returns(uint256){

    if(0 == balance[msg.sender]){
        account.push(msg.sender);
          }
        balance[msg.sender] += msg.value;
        return balance[msg.sender];
    }
```

กำหนดเงื่อนไขของการกู้ยืม

```
 function paymoney(uint value, address payable adds) payable
public {
        balance[msg.sender] -= value;
        balance[adds] += value;
        adds.transfer(value);
    }
        function checkbalance(address adds)  public view
returns(uint256){
        return balance[adds];
    }
```

กำหนดเงื่อนไขของการทำ Transaction

```
//////spare money
        function sparemoney(address addrx, uint amountx) public
returns(bool){
        require(balance[msg.sender] >= amountx ," money enough");
        require(amountx >= 1000000000," minimun is 1000000000");
        addressallow[msg.sender] = addrx;
        addressallow2[addrx] = msg.sender;
        amountallow[msg.sender] = amountx;
        amountallow2[addrx] = amountx;
        rateallow[msg.sender] = 5;
        rateallow2[addrx] = 5;
        datecheck[msg.sender] = now;
        return true;
    }
```

กำหนดเงื่อนไขจำนวนเงินให้เพียงพอต่อการกู้ยืม

```
 function checkamountcanborrow(address addsr) public view returns
(uint amountx,uint rata){
        amountx = 0;
        rata = 0;
        amountx = amountallow2[addsr];
        rata = rateallow2[addsr];
    }

}
```

ขั้นตอนการตรวจสอบจำนวนที่สามารถกู้ยืมได้

```
 function borrowmoney(uint256 withdrawamount,address addsr1,
address payable addsr2) public returns(uint256){
        //require(addressallow[addsr1] == addsr2, "not correct");
        require(withdrawamount == amountallow[addsr1], "amount is no
enough");
        transationhappen[addsr2] = addsr1;
        timestamp1[addsr2] = now;
        balance2[addsr1] = withdrawamount;
        balance2[addsr2] = withdrawamount;
        balance[addsr1] -= withdrawamount;
        balance[addsr2] += withdrawamount;
        addsr2.transfer(withdrawamount);
    }

}
```

ขั้นตอนการกู้ยืม

```
 function checkamountpayandgetbackmoney()public view returns
(uint256 a){
        a = balance2[msg.sender]*(1000000+50000);
        a = a/1000000;
    }
```

ขั้นตอนการตรวจสอบการคืนเงิน

```
 function paymoneyback(uint256 value, address payable adds) payable public{
      //require(balance[msg.sender] >= checkamountpayandgetbackmoney(),
"not enough");
      //require(value ==  (balance2[msg.sender]*(1000000+50000))/1000000,
"not enough");
      uint256 b = 0;
      b = balance2[msg.sender]*(1000000+100000);
      b = b/1000000;
      //*(now-timestamp1[msg.sender])
      balance[msg.sender] -= value;
      balance2[msg.sender] -= value;
      balance[adds] = balance[adds] + value - b ;
      balance2[adds] -= value;
      interestpooling += b;
      value = value - b;
      adds.transfer(value);
   }

      function checkamountpoolinginterest()public view returns(uint256 c){
        c = interestpooling;

   }
   //0xaC7C9881aeFf16D9557A85d5C640C54D0851B687
0x0C923dA8AC3a1eC8C43feE12fF9DdE847EfF11D6
0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
}
```

ขั้นตอนการคืนเงิน

8 STEP TO BORROWING AND LENDING

1. Add Liquidity (Lending)

Lender add money for lending, this process to guarantee that you have enough money for lending. and set personal interest you want

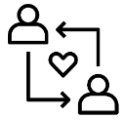
2. Borrowing Request

Request amount of money you want to borrowing. And maximum interest you affordable to absorb

3. Matching Peer

[Matching similarity criteria]

Lender

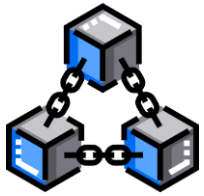


Borrower

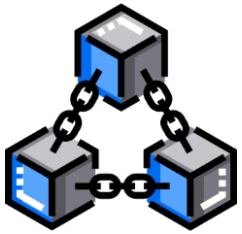
Both agree with borrowing condition (such as borrow amount , interest pay , tenor)

4. Create Smart Contract

Lender create smart contracts through front-end applications with agreed criteria.



5. Send Smart Contract to Borrower



Lender send smart contracts to Borrower to request money as agreed.

6. Ask for Money

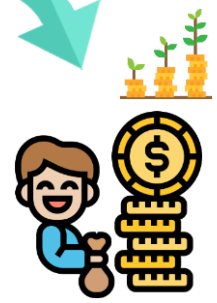
Borrower withdraw money under agreed amount in smart contract

7. Check Daily Payment

Borrower can check daily payment (included interest in daily basis)

8. Payback money with interest

Borrower payback money to lender by address in smart contract (cannot pay to another address)



Thank you