รายงาน

เรื่อง การเก็บข้อมูลการคืนหนังสือจากการยืมหนังสือของมหาลัย ในแบบ Blockchain

จัดทำโดย

B6406325 นายปริวัตร ศรีทร

เสนอ

อาจารย์ ดร.ปริญญ์ ศรเลิศล้ำวาณิช

รายงานนี้เป็นส่วนหนึ่งของการวิชา 523355 INTRODUCTION TO BLOCKCHAIN TECHNOLOGY

สำนักวิชาวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์

ภาคเรียนที่ 2 ปีการศึกษา 2566

มหาวิทยาลัยเทคโนโลยีสุรนารี

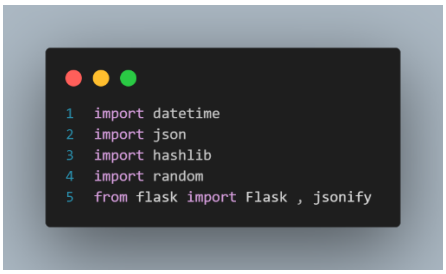
ทำไมถึงเลือกใช้การเก็บข้อมูลการคืนหนังสือจากการยืมหนังสือของมหาลัย ในแบบ Blockchain

เนื่องจากการเก็บข้อมูลแบบ Blockchain เป็นระบบที่สามารถควบคุมและปกป้องข้อมูลได้อย่างมีประสิทธิภาพ เนื่องจากข้อมูลทั้งหมดถูกเข้ารหัสและเก็บบันทึกในบล็อกที่เชื่อมต่อกัน การใช้ Blockchain สามารถช่วยป้องกันการแก้ไขข้อมูลหรือทำการเปลี่ยนแปลงโดยไม่ได้รับอนุญาต ทำให้สามารถติดตามประวัติการทำรายการ มีประโยชน์ในกรณีที่ต้องการตรวจสอบการยืมหนังสือก่อนหน้า การคืน หรือการทำรายการอื่นๆ เกี่ยวกับหนังสือ อีกทั้งช่วยลดความเสี่ยงที่เกี่ยวข้องกับการปลอมแปลงข้อมูลหรือการแก้ไขโดยไม่ได้รับอนุญาต เนื่องจากข้อมูลถูกเก็บไว้ในรูปแบบของบล็อกที่ถูกเชื่อมต่อ การทำให้ข้อมูลเปลี่ยนแปลงก็จะต้องมีการเปลี่ยนแปลงในบล็อกทั้งหมดที่ตามมา และลดความซับซ้อนในการจัดการระบบเก็บข้อมูล, เนื่องจากไม่จำเป็นต้องมีฐานข้อมูลแยกต่างหากและไม่ต้องพึ่งพาบริการกลางในการบริหารจัดการ

ในแต่ละ Block จะมีข้อมูล รหัสนักศึกษา, ชื่อผู้ยืม, ชื่อหนังสือ, วันเวลาที่ยืม สำหรับโปรเจกต์จะมีการทำงานบน Flask web server และมีการสร้าง Blockchain ซึ่งรองรับการเพิ่ม block, การ mining, การตรวจสอบความถูกต้องของ Blockchain, แก้ไขข้อมูลใน block, และการทำให้ chain ไม่ถูกต้องเพื่อทดสอบความทนทานของระบบ

อธิบายการทำงานโดยรวมของระบบ

1. ในโปรเจกต์จะใช้ภาษา python ในการเขียน Blockchain ทำให้ต้องมีการเรียกใช้ Library เพื่อช่วยในการสร้างฟังก์ชั่นของ Blockchain ดังรูป



```python
import datetime
import json
import hashlib
import random
from flask import Flask , jsonify
```

2. สร้าง Class Blockchain สำหรับฟังก์ชั่นของ Blockchain โดยเริ่มจาก ฟังก์ชั่น __init__() ทำการเริ่มต้น blockchain โดยมี self.chain และสร้าง Genesis block

```python
class Blockchain:
    def __init__(self):
        #เก็บกลุ่มของ block
        self.chain = [] #list ที่เก็บ block
        #genesis block
        self.create_block(nonce=1,previous_hash="0",id="B64xxxxx",name="suranaree",booktitle="eng5")
```

2.1. ฟังก์ชั่น create_block() สร้างบล็อกใหม่ใน blockchain ด้วยพารามิเตอร์ที่ระบุ ซึ่งประกอบด้วย index, timestamp, nonce, data (มี student ID, ชื่อผู้ยืม, ชื่อหนังสือ, และวันที่คืน), และ hash ของบล็อกก่อนหน้า

```python
def create_block(self,nonce,previous_hash,id,name,booktitle):
        #เก็บส่วนประกอบของ block แต่ละ block
        block={
            "index":len(self.chain)+1,
            "timestamp":str(datetime.datetime.now()),
            "nonce":nonce,
            "data": {
                "รหัสนักศึกษา": id,
                "ชื่อผู้ยืม": name,
                "ชื่อหนังสือ": booktitle,
                "วันที่คืน": str(datetime.datetime.now())
            },
            "previous_hash":previous_hash
        }
        block["hash"] = self.hash(block)
        self.chain.append(block)
        return block
```

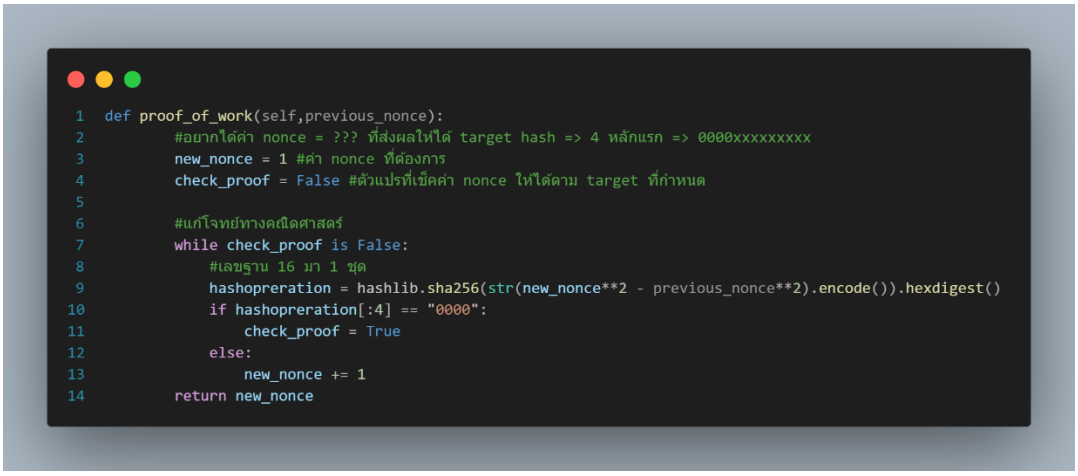
2.2 ฟังก์ชั่น get_previous_block() คืนค่าบล็อกล่าสุดใน blockchain

```python
def get_previous_block(self):
        return self.chain[-1]
```

2.3 ฟังก์ชั่น hash() ทำการแฮชบล็อกโดยใช้ขั้นตอน SHA-256

```python
def hash(self,block):
        #แปลง python object (dict) => json object
        encode_block = json.dumps(block,sort_keys=True).encode()
        #sha-256
        return hashlib.sha256(encode_block).hexdigest()
```

2.4 ฟังก์ชั่น proof_of_work() อัลกอริทึม proof-of-work เพื่อค้นหา nonce ที่ตรงตามเงื่อนไขที่ระบุ (ในที่นี้คือ hash ที่เริ่มต้นด้วยศูนย์สี่ตัว)



```python
def proof_of_work(self,previous_nonce):
        #อยากได้ค่า nonce = ??? ที่ส่งผลให้ได้ target hash => 4 หลักแรก => 0000xxxxxxxxx
        new_nonce = 1 #ค่า nonce ที่ต้องการ
        check_proof = False #ตัวแปรที่เช็คค่า nonce ให้ได้ตาม target ที่กำหนด

        #แก้โจทย์ทางคณิตศาสตร์
        while check_proof is False:
            #เลขฐาน 16 มา 1 ชุด
            hashopreration = hashlib.sha256(str(new_nonce**2 - previous_nonce**2).encode()).hexdigest()
            if hashopreration[:4] == "0000":
                check_proof = True
            else:
                new_nonce += 1
        return new_nonce
```

2.5 ฟังก์ชั่น is_chain_valid() ตรวจสอบว่า blockchain ที่กำหนดมีความถูกต้องหรือไม่ โดยการตรวจสอบ hash และ nonce ของแต่ละบล็อก

```python
def is_chain_valid(self,chain):
        previous_block = chain[0]
        block_index = 1
        while block_index<len(chain):
            block = chain[block_index] # block ที่ตรวจสอบ

            if block["previous_hash"] != self.hash(previous_block):
                return False

            previous_nonce = previous_block["nonce"] #nonce block ก่อนหน้า
            nonce = block["nonce"] #nonce ของ block ที่ตรวจสอบ
            hashopreration = hashlib.sha256(str(nonce**2 - previous_nonce**2).encode()).hexdigest()

            if hashopreration[:4] != "0000":
                return False
            previous_block = block
            block_index += 1
        return True
```

2.6 ฟังก์ชั่น make_chain_invalid() แก้ไข hash ของบล็อก 3 ในลำดับเพื่อทำให้ blockchain ทั้งหมดไม่ถูกต้องสำหรับใช้กับฟังชั่น is_chain_valid()

```python
def make_chain_invalid(self):
    # Modify the hash of the last block to make it invalid
        modified_block = self.chain[3]
        modified_block["hash"] = "InvalidHash123"

        return "Chain modified successfully to make it invalid"
```

3. Flask Web Server ทำการเริ่มต้นแอปพลิเคชันเว็บ Flask, สร้างอินสแตนซ์ของคลาส Blockchain เพื่อทำงาน

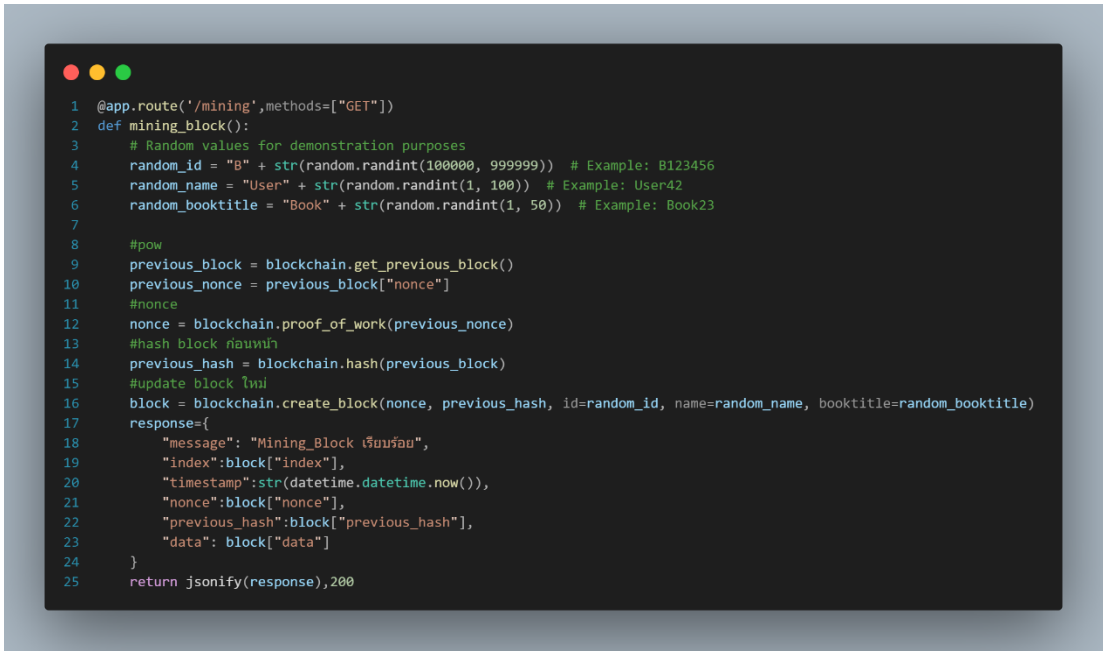
```python
#web server
app = Flask(__name__)
#ใช้งาน blockchain
blockchain = Blockchain()
```

4. การเชื่อมต่อกับเว็บไซต์อย่างง่ายเพื่อเรียกใช้งานฟังก์ชั่น Blockchain

4.1 /get_chain แสดง blockchain ทั้งหมดและความยาวของมัน

```python
@app.route('/get_chain',methods=["GET"])
def get_chain():
    response={
        "chain":blockchain.chain,
        "length":len(blockchain.chain)
    }
    return jsonify(response),200
```

4.2 /mining ขุดบล็อกใหม่โดยสร้างข้อมูลสุ่ม (student ID, ชื่อผู้ยืม, ชื่อหนังสือ) และค้นหา nonce ที่ถูกต้องโดยใช้อัลกอริทึม proof-of-work



```python
@app.route('/mining',methods=["GET"])
def mining_block():
    # Random values for demonstration purposes
    random_id = "B" + str(random.randint(100000, 999999))  # Example: B123456
    random_name = "User" + str(random.randint(1, 100))  # Example: User42
    random_booktitle = "Book" + str(random.randint(1, 50))  # Example: Book23

    #pow
    previous_block = blockchain.get_previous_block()
    previous_nonce = previous_block["nonce"]
    #nonce
    nonce = blockchain.proof_of_work(previous_nonce)
    #hash block ก่อนหน้า
    previous_hash = blockchain.hash(previous_block)
    #update block ใหม่
    block = blockchain.create_block(nonce, previous_hash, id=random_id, name=random_name, booktitle=random_booktitle)
    response={
        "message": "Mining_Block เรียบร้อย",
        "index":block["index"],
        "timestamp":str(datetime.datetime.now()),
        "nonce":block["nonce"],
        "previous_hash":block["previous_hash"],
        "data": block["data"]
    }
    return jsonify(response),200
```

4.3 /check ตรวจสอบว่า blockchain ปัจจุบันถูกต้องหรือไม่

```python
@app.route('/check',methods=["GET"])
def is_valid():
    is_valid = blockchain.is_chain_valid(blockchain.chain)
    if is_valid:
        response={"message":"Blockchain Valid"}
    else :
        response={"message":"Blockchain Is Not Valid"}
    return jsonify(response),200
```

4.4 /edit แก้ไขข้อมูลในบล็อก (ยกตัวอย่างบล็อกที่ index 3)

```python
@app.route('/edit', methods=["GET"])
def edit_data():
    # ตรวจสอบว่ามี Block อยู่หรือไม่
    if len(blockchain.chain) < 4:
        response = {"message": "Blockchain does not have enough blocks"}
        return jsonify(response), 400

    block_to_edit = blockchain.chain[3]

    # แก้ไขข้อมูลใน Block ที่ index = 3
    block_to_edit["data"]["รหัสนักศึกษา"] = "B20000"
    block_to_edit["data"]["ชื่อผู้ยืม"] = "นาย"
    block_to_edit["data"]["ชื่อหนังสือ"] = "ช่างๆๆๆ"

    block_to_edit["hash"] = blockchain.hash(block_to_edit)

    response = {
        "message": "Data Updated Successfully",
        "index": block_to_edit["index"],
        "timestamp": block_to_edit["timestamp"],
        "data": block_to_edit["data"],
        "nonce": block_to_edit["nonce"],
        "previous_hash": block_to_edit["previous_hash"],
        "hash": block_to_edit["hash"]
    }
    return jsonify(response), 200
```

4.5 /modified แก้ไขบล็อกสุดท้ายในลำดับเพื่อทำให้ blockchain ทั้งหมดไม่ถูกต้อง

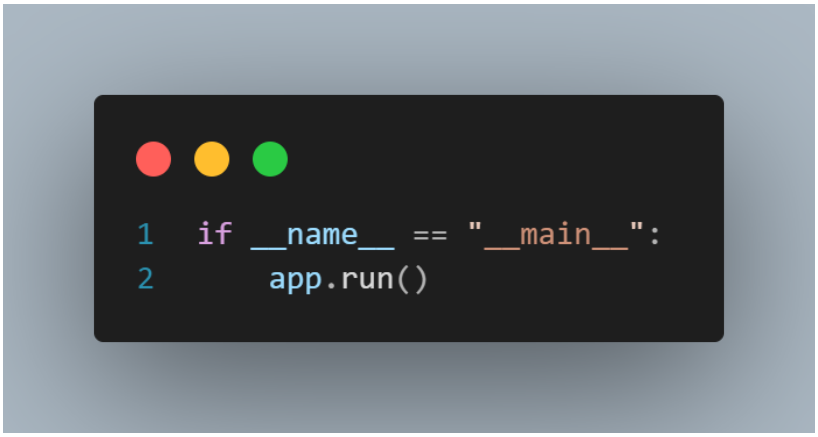
```python
@app.route('/modified', methods=["GET"])
def make_invalid():
    result = blockchain.make_chain_invalid()

    if result.startswith("Chain modified"):
        response = {"message": result}
    else:
        response = {"error": result}

    return jsonify(response), 200 if result.startswith("Chain modified") else 400

```

4.6 app.run() เริ่มต้นเซิร์ฟเวอร์เว็บ Flask



```python
if __name__ == "__main__":
    app.run()
```

5. ตัวอย่าง Blockchain 10 chain


```
{
    "chain": [
        {
            "data": {
                "ชื่อผู้ยืม": "suranaree",
                "ชื่อหนังสือ": "eng5",
                "รหัสนักศึกษา": "B64xxxxx",
                "วันที่คืน": "2024-01-26 13:30:03.260779"
            },
            "hash": "12a4cffdaa57a697e901cef80263dedaf413bf1cca12a6c19f2689b6c2ac61c7",
            "index": 1,
            "nonce": 1,
            "previous_hash": "0",
            "timestamp": "2024-01-26 13:30:03.260779"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User4",
                "ชื่อหนังสือ": "Book8",
                "รหัสนักศึกษา": "B885533",
                "วันที่คืน": "2024-01-26 13:30:37.835307"
            },
            "hash": "1d94fa391efa524ca6583773c22ab7487eb4b9dc58ca4154cd2539e7b279a5c3",
            "index": 2,
            "nonce": 533,
            "previous_hash": "a4f26d51aca1c455f3a80179baf2d28468c08a1b6470068904226900lf42ded3",
            "timestamp": "2024-01-26 13:30:37.835307"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User36",
                "ชื่อหนังสือ": "Book27",
                "รหัสนักศึกษา": "B764003",
                "วันที่คืน": "2024-01-26 13:30:38.680076"
            },
            "hash": "d9940761297a3abd7e827a89524364420ec0d5940a8a66073e34276c88168b19",
            "index": 3,
            "nonce": 45293,
            "previous_hash": "a638c4a505e8f7de31f51995b36620bc9d5bc2bbfa306b21a22336c71f0bad05",
            "timestamp": "2024-01-26 13:30:38.680076"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User35",
                "ชื่อหนังสือ": "Book42",
                "รหัสนักศึกษา": "B119336",
                "วันที่คืน": "2024-01-26 13:30:40.067848"
            },
            "hash": "6afde5e6917f39d4f045426af52de323873180f9cebe4f43a47ace57cd52115d",
            "index": 4,
            "nonce": 21391,
            "previous_hash": "2755b19265a52245fce08bbbcf69ee0589fb621a0d3c4f96c52870e03c33b2a8",
            "timestamp": "2024-01-26 13:30:40.067848"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User79",
                "ชื่อหนังสือ": "Book6",
                "รหัสนักศึกษา": "B636357",
                "วันที่คืน": "2024-01-26 13:30:40.631362"
            },
            "hash": "026fc8a40b1999248dee2d49aa45da9f2562cccbb044cbb524e10504070 5dc53",
            "index": 5,
            "nonce": 8018,
            "previous_hash": "9156d837d7e264a80fc0e47793acf3d9054daa2bc4a9690b6787d0af87e60084",
            "timestamp": "2024-01-26 13:30:40.631362"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User57",
                "ชื่อหนังสือ": "Book29",
                "รหัสนักศึกษา": "B376627",
                "วันที่คืน": "2024-01-26 13:30:41.266942"
            },
            "hash": "93de1fb8933d5a81d7d8e9e0fcc1310f6912d0fecb11acf9150b84c9eb2fbd72",
            "index": 6,
            "nonce": 48191,
            "previous_hash": "d3381c5284ff82508357c4cf4b2a5237da9e5d4f91de5092391779d74a66f106",
            "timestamp": "2024-01-26 13:30:41.266942"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User19",
                "ชื่อหนังสือ": "Book4",
                "รหัสนักศึกษา": "B513357",
                "วันที่คืน": "2024-01-26 13:30:41.805433"
            },
            "hash": "e5e67de2212d92e5293cf19655291fbdbaad9492c90d5cec0603988af7a3fe8e",
            "index": 7,
            "nonce": 19865,
            "previous_hash": "66a544d75809e32e252d3cd6c1579dfb10183669ad40ed5959bd4e0049d4835f",
            "timestamp": "2024-01-26 13:30:41.805433"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User52",
                "ชื่อหนังสือ": "Book40",
                "รหัสนักศึกษา": "B796857",
                "วันที่คืน": "2024-01-26 13:30:42.474043"
            },
            "hash": "4f139f0b3c47e335965b2d3256cee3e36ea8fdaf14dfb70f702ce921e75b1862",
            "index": 8,
            "nonce": 95063,
            "previous_hash": "19d3e880c3fa2c489c7255ae6f66fc1173f3c47adfbcd9b111ab9ecb88e52099",
            "timestamp": "2024-01-26 13:30:42.474043"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User7",
                "ชื่อหนังสือ": "Book34",
                "รหัสนักศึกษา": "B267479",
                "วันที่คืน": "2024-01-26 13:30:42.962488"
            },
            "hash": "dfed19a439773de29e0f4ea5f300ff73280bc705c904c64dd857be8a12057e85",
            "index": 9,
            "nonce": 15457,
            "previous_hash": "e38819a0e5e72c865172a7f34b26fe2a939994997338b4b2d297c8889aa4260b",
            "timestamp": "2024-01-26 13:30:42.962488"
        },
        {
            "data": {
                "ชื่อผู้ยืม": "User92",
                "ชื่อหนังสือ": "Book9",
                "รหัสนักศึกษา": "B372454",
                "วันที่คืน": "2024-01-26 13:30:43.490969"
            },
            "hash": "c5cedca5bf182327adbf67e1b43567b352e7743adc9638fce07aa0dd6bccd78f",
            "index": 10,
            "nonce": 15479,
            "previous_hash": "309f1c4587ed17efa4878d8541eed84bea80c535703ac1ecc4b9cc39635a5c73",
            "timestamp": "2024-01-26 13:30:43.490969"
        }
    ],
    "length": 10
}
```

5.1 เมื่อเรียกใช้ /check ในกรณีที่ Blockchian มีความถูกต้อง

5.2 เมื่อเรียกใช้ /check ในกรณีที่ Blockchian ถูกแก้ไขข้อมูลบางส่วน (เรียกใช้ /modified)