

ระบบจัดการสิทธิการเข้าถึงด้วยแอปพลิเคชัน

แบบกระจายศูนย์บนบล็อกเชน

**Identity and Access Management Distributed Application
on Blockchain**

พงศ์ภัค พุดซ้อน

กฤษณะ วิปັນเขตร์

ฉัตรชัย นพพลั้ง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

ระบบจัดการสิทธิการเข้าถึงด้วยแอปพลิเคชันแบบกระจายศูนย์บนบล็อกเชน

ปีการศึกษา 2654

ปริญญาานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจัดการสิทธิ์การเข้าถึงด้วยแอปพลิเคชันแบบกระจายศูนย์บนบล็อกเชน

IDENTITY AND ACCESS MANAGEMENT DISTRIBUTED APPLICATION ON
BLOCKCHAIN

ผู้จัดทำ

- | | | |
|--------------------------------|--------------|----------|
| 1. นายพงศ์ภัค พุดซ้อน | รหัสนักศึกษา | 60010647 |
| 2. ว่าที่ ร.ต.กฤษณะ วิป็นเขตร์ | รหัสนักศึกษา | 61015004 |
| 3. นายฉัตรชัย นพพลั้ง | รหัสนักศึกษา | 62015019 |

อาจารย์ที่ปรึกษา
(ดร.ปริญญา เอกปริญญา)

ระบบจัดการสิทธิการเข้าถึงด้วยแอปพลิเคชัน

แบบกระจายศูนย์บนบล็อกเชน

| | | |
|------------------|------------|------------------|
| นายพงศ์ศักดิ์ | พุดซ้อน | 60010647 |
| ว่าที่ ร.ต.กฤษณะ | วิปน์เขตร์ | 61015004 |
| นายฉัตรชัย | นพพลิ่ง | 62015019 |
| ดร.ปริญญา | เอกปริญญา | อาจารย์ที่ปรึกษา |
| ปีการศึกษา 2564 | | |

บทคัดย่อ

เทคโนโลยี Blockchain เป็นเทคโนโลยีที่รู้จักในรูปแบบของ Cryptocurrencies ในปัจจุบันได้มีการนำเทคโนโลยี Blockchain มาสร้างแอปพลิเคชันใหม่ ๆ ที่ไม่เคยมีมาก่อน ในปัจจุบันการจัดการข้อมูลประจำตัวดิจิทัลนั้นพึ่งพาผู้ให้บริการมากเกินไปส่งผลให้ผู้ให้บริการจำเป็นที่จะต้องเก็บรักษาข้อมูลของผู้ใช้ด้วยตัวเอง ทำให้มีค่าใช้จ่ายที่สูง และทำให้เกิดข้อมูลซ้ำซ้อนจำนวนมากจากผู้ให้บริการรายต่าง ๆ

ระบบจัดการสิทธิการเข้าถึงด้วยแอปพลิเคชันแบบกระจายศูนย์บนบล็อกเชน เป็นบริการที่ช่วยสำหรับการจัดการข้อมูลประจำตัวดิจิทัลบน Blockchain เพื่อจัดการกับความปลอดภัยของข้อมูล ป้องกันการปลอมแปลงข้อมูล ลดการพึ่งพาผู้ให้บริการหลาย ๆ ราย กระจายอำนาจในการจัดการข้อมูลประจำตัวให้ผู้ให้บริการ ช่วยให้ผู้ใช้บริการไม่ต้องจำข้อมูลประจำตัวให้มากมาย เหมือนกับการใช้บริการอื่น ๆ

Identity and Access Management Distributed Application on Blockchain

| | | |
|-------------------------|-----------|----------|
| Mr. Phongpak | Pudsorn | 60010647 |
| Acting Sub Lt. Kritsana | Wipankhet | 61015004 |
| Mr. Chatchai | Nopplang | 62015019 |
| Dr. Parinya | Ekparinya | Advisor |
| Academic Year 2021 | | |

ABSTRACT

Blockchain technology is a technology known in the form of Cryptocurrencies
Blockchain technology is currently being used to create new applications that have never been
seen before. Nowadays, digital identity management is too dependent on service providers,
resulting in service providers having to maintain user data on their own. causing high costs and
resulting in a large amount of redundant data from different service providers.

Blockchain-based decentralized application access management system It is a service
that helps for managing digital identity on Blockchain to deal with data security. Prevent forgery
Reduce dependency on multiple service providers. Decentralize identity management for users.
Saves users from having to remember a lot of their credentials. Same as other service providers.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับคำแนะนำ คำปรึกษา ความช่วยเหลืออื่น ๆ ทั้งทางตรงและทางอ้อมจากหลายฝ่าย ปริญญานิพนธ์ฉบับนี้จะไม่สำเร็จลุล่วงไปได้ หากปราศจากคำชี้แนะของบุคคลเหล่านี้ ขอขอบคุณอาจารย์ที่ปรึกษา อาจารย์ปริญญา เอกปริญญา ที่คอยช่วยเหลือตลอดเวลาทั้งในและนอกเวลาเรียน ตั้งแต่เริ่มต้นจนปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณอาจารย์และบุคลากรต่าง ๆ ในสาขาวิชาที่ได้ให้การอบรมและสั่งสอนความรู้ต่าง ๆ มาโดยตลอด

ขอขอบคุณรุ่นพี่และเพื่อน ๆ ในสาขาวิชาที่คอยให้คำแนะนำและให้กำลังใจมาโดยตลอด
สุดท้ายนี้ขอขอบคุณ บิดา มารดา และครอบครัวที่ได้เลี้ยงดู อบรม สั่งสอน และให้การสนับสนุน พร้อมทั้งให้โอกาสในการศึกษาและให้กำลังใจเสมอมา

พงศ์ภัค พุดซ้อน
กฤษณะ วิป็นเขตร์
ฉัตรชัย นพพลั้ง

สารบัญ

| | หน้า |
|-------------------------------------|------|
| ทศด้อยภาษาไทย..... | I |
| บทศด้อยภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญตาราง..... | VI |
| สารบัญรูป..... | VII |
| | |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ความเป็นมา..... | 1 |
| 1.2 วัตถุประสงค์..... | 1 |
| 1.3 ขอบเขตของโครงการ..... | 1 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ..... | 2 |
| 1.5 แผนการดำเนินงาน..... | 3 |
| | |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้อง..... | 4 |
| 2.1 REST API..... | 4 |
| 2.2 AAA Protocol..... | 7 |
| 2.3 OAuth..... | 10 |
| 2.4 Blockchain..... | 11 |
| 2.5 Smart Contract..... | 12 |
| 2.6 Web3.js..... | 12 |
| 2.7 งานวิจัยที่เกี่ยวข้อง..... | 13 |
| | |
| บทที่ 3 วิธีการดำเนินการ..... | 14 |
| 3.1 ภาพรวม..... | 14 |
| 3.2 การออกแบบ Sequence Diagram..... | 15 |

สารบัญ (ต่อ)

| | หน้า |
|--------------------------------------|------|
| 3.3 แผนภาพ ER diagram | 24 |
| 3.4 การออกแบบ Sequence Diagram | 26 |
| 3.5 แผนภาพ ER diagram | 32 |
| บทที่ 4 ผลการดำเนินงาน | 34 |
| 4.1 ผลการทดลอง | 34 |
| บทที่ 5 สรุปผลการทดลอง | 38 |
| 5.1 สรุปผลการพัฒนาระบบ | 38 |
| 5.2 ปัญหาที่พบ | 38 |
| 5.3 แนวทางในการพัฒนาต่อ | 38 |
| บรรณานุกรม | 39 |

สารบัญตาราง

| ตาราง | หน้า |
|---|------|
| 1.1 แผนการดำเนินงาน..... | 2 |
| 2.1 การเทียบ CRUD กับ HTTP (Hypertext Transfer Protocol) and RESTful APIs..... | 6 |
| 2.2 HTTP Response Status Code..... | 6 |
| 3.1 User..... | 24 |
| 3.2 Task..... | 25 |
| 3.3 Share..... | 25 |
| 3.4 User..... | 32 |
| 3.5 clients..... | 33 |

สารบัญรูป

| รูป | หน้า |
|--|------|
| 3.1 ขั้นตอนการสมัครบัญชี..... | 15 |
| 3.2 ขั้นตอนการ Login..... | 16 |
| 3.3 ขั้นตอนการ Logout..... | 17 |
| 3.4 ขั้นตอนการสร้าง Task..... | 18 |
| 3.5 ขั้นตอนการลบ Task | 19 |
| 3.6 ขั้นตอนการแก้ไข Task..... | 20 |
| 3.7 ขั้นตอนการค้นหา Task..... | 21 |
| 3.8 ขั้นตอนการแชร์ Task..... | 22 |
| 3.9 ขั้นตอนการยกเลิกการแชร์ Task..... | 23 |
| 3.10 ER-diagram..... | 24 |
| 3.11 ขั้นตอนการสมัครบัญชี..... | 26 |
| 3.12 ขั้นตอนการ Login..... | 27 |
| 3.13 ขั้นตอนการเพิ่มผู้ขอใช้บริการ..... | 29 |
| 3.14 ขั้นตอนการเข้าสู่ระบบจากผู้ให้บริการ..... | 30 |
| 3.15 แผนภาพ ER diagram..... | 32 |
| 4.1 หน้าแรกเมื่อเข้าสู่ระบบ..... | 34 |
| 4.2 หน้าสมัครสมาชิก | 35 |
| 4.3 หน้าเข้าสู่ระบบ | 35 |
| 4.4 หน้าการสร้าง Task งาน | 36 |
| 4.5 การลบ Task | 36 |
| 4.6 หน้าหลักของ Service | 37 |

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

การพิสูจน์ตัวตนและการกำหนดสิทธิ์ถือเป็นกลไกพื้นฐานในการควบคุมการเข้าถึงระบบสารสนเทศเพื่อความปลอดภัย ด้วยความนิยมของที่เพิ่มขึ้นอย่างต่อเนื่องของบริการเครือข่ายสังคมออนไลน์บัญชีผู้ใช้ของบริการเหล่านั้นจึงถูกนำมาใช้ร่วมกับการพิสูจน์ตัวตนและการกำหนดสิทธิ์ของเว็บแอปพลิเคชันและโมบายแอปพลิเคชันอื่นๆ ถึงแม้ว่าจะเป็น การลดภาระของผู้พัฒนาและเพิ่มความสะดวกของผู้ใช้งาน แต่ก็เป็นการพึ่งพาและให้ความไว้วางใจกับผู้ให้บริการเครือข่ายสังคมออนไลน์เพียงไม่กี่ราย และเปิดโอกาสให้ผู้ให้บริการเหล่านั้นใช้ประโยชน์จากข้อมูล การกำหนดสิทธิ์ในเข้าถึงเว็บแอปพลิเคชัน และ โมบายแอปพลิเคชันของผู้ใช้โดยปริยาย

บล็อกเชน (blockchain) เป็นเทคโนโลยีที่ถูกออกแบบมาเพื่อจัดการความจำเป็นที่ต้องพึ่งพาและให้ความไว้วางใจ บุคคลที่สามรายใดรายหนึ่ง บล็อกเชนใช้การจัดเก็บข้อมูลแบบ Distributed Ledger โดยจัดเก็บข้อมูลชุดเดียวกันเอาไว้ หลายที่โดยสมาชิกหลายราย โดยบล็อกเชนรองรับแค่การเพิ่มข้อมูลเท่านั้น (Append-only) สมาชิกที่จะเพิ่มข้อมูลต้องปฏิบัติ ตามเงื่อนไขและกระบวนการที่ตกลงกันไว้ก่อนภายในเครือข่ายบล็อกเชนนั่นๆ โดยที่สมาชิกทุกรายในเครือข่ายสามารถ ตรวจสอบความถูกต้องของกระบวนการที่เกิดขึ้นได้

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อศึกษาการทำงานและโครงสร้างของเทคโนโลยี Blockchain
- 2) เพื่อศึกษาเกี่ยวกับการพัฒนาแอปพลิเคชันที่ทำงานร่วมกับ Blockchain
- 3) เพื่อพัฒนา Web Service ให้สามารถทำงานร่วมกับเครือข่าย Blockchain
- 4) เพื่อพัฒนาระบบจัดการสิทธิ์การเข้าถึงด้วยแอปพลิเคชันแบบกระจายศูนย์ ซึ่งไม่ผูกพันกับผู้ให้บริการรายใดรายหนึ่ง

1.3 ขอบเขตของโครงการ

ระบบจัดการสิทธิ์การเข้าถึงด้วยแอปพลิเคชันแบบกระจายศูนย์บนบล็อกเชน จัดทำในรูปแบบของ Web Service โดยแบ่งการทำงานดังนี้

1) ส่วนของ Web Service

จะเป็นบริการที่คอยตรวจสอบและจัดการ Request ที่เข้ามาขอข้อมูล โดยใช้ Token ในการเข้าถึงข้อมูล หลังจากที่เราตรวจสอบและได้รับสิทธิ์ โดยที่ตัว Service จะมี Web Application ในการช่วยจัดการข้อมูลให้ดูง่ายขึ้น โดยมีฟังก์ชันดังนี้

- User สามารถสมัครสมาชิกได้
- User สามารถเข้าสู่ระบบได้
- User สามารถจัดการสิทธิ์และข้อมูลต่าง ๆ ได้
- ผู้ให้บริการอื่น ๆ สามารถขอใช้งานระบบตรวจสอบสิทธิ์ได้
- สามารถเก็บข้อมูล บนเครือข่าย Ethereum Blockchain ได้

2) ส่วนของ Ethereum Blockchain

Ethereum เป็นหนึ่งในเครือข่ายที่ใช้เทคโนโลยี Blockchain ในการเก็บข้อมูลและรองรับการพัฒนาแอปพลิเคชันแบบกระจายศูนย์ ซึ่งโครงงานนำมาใช้ในการจัดเก็บข้อมูลโดยมีฟังก์ชันดังนี้

- สามารถเก็บข้อมูลไว้บนเครือข่ายแบบกระจายศูนย์ได้
- สามารถทำงานร่วมกับ Web Service เพื่อให้ร้องขอข้อมูลหรือเรียกใช้งานได้

3) ส่วนของ Web Application ตัวอย่าง

เป็น Web Application ที่ไว้ใช้ทดสอบระบบ Web Service โดยที่ Web Application ตัวอย่างนั้น จัดทำเป็นการจัดการ To do list ที่รองรับหลาย User โดยมีฟังก์ชันดังนี้

- User สามารถสมัครสมาชิกได้
- User สามารถเข้าสู่ระบบได้
- User สามารถเข้าสู่ระบบได้โดยผ่านการ Authorized เพื่อขอสิทธิ์จาก Web Service ได้
- User สามารถ เพิ่ม ลบ แก้ไข Task ได้
- User สามารถ Share Task ให้กับ User อื่นที่ต้องการได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ผู้จัดทำได้รับความรู้ความเข้าใจ ในระบบและโครงสร้างของเทคโนโลยี Blockchain
- 2) สามารถพัฒนา Web Service ที่เก็บข้อมูลไว้บน เครือข่าย Ethereum Blockchain
- 3) ได้ระบบตัวอย่างที่ใช้ในการจัดการสิทธิ์ที่ไม่พึ่งพากับผู้ให้บริการรายใดรายหนึ่ง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 REST API

คำว่า REST นั้นย่อมาจาก Representational state transfer โดยที่ REST คือคำเรียกในเชิงของสถาปัตยกรรม ที่ใช้ประโยชน์จากเทคโนโลยี Web protocol เพื่อใช้ในการสร้าง Web Service นำเสนอครั้งแรกโดย Roy Fielding และคำว่า API มาจาก Application Program Interface โดย API จะเป็นตัวกลางที่อนุญาตให้แอปพลิเคชันแลกเปลี่ยนข้อมูลโดย Client จะใช้ API ส่ง Request และ Response [1]

REST architectural style ประกอบไปด้วย

Performance: ประสิทธิภาพในการโต้ตอบ

Scalability: มีความสามารถในการปรับขยายได้ทำให้สามารถรองรับส่วนประกอบจำนวนมาก

Simplicity: ความเรียบง่ายของอินเตอร์เฟซที่เหมือนกัน

Modifiability: ความสามารถในการปรับเปลี่ยนส่วนประกอบเพื่อตอบสนองความต้องการที่เปลี่ยนแปลงแม้ในขณะที่แอปพลิเคชันทำงานอยู่

Visibility: การมองเห็นการสื่อสารระหว่าง Component ผ่าน Service agents

Portability: ง่ายต่อการโยกย้ายในส่วนของ Program และ Data

Reliability: ความน่าเชื่อถือในการต้านทานความล้มเหลวในระบบ

2.1.1 RESTful Web Services (RWS)

Web Service ที่ใช้ REST architectural style และใช้ HTTP Methods เป็นที่รู้จักกันในชื่อ RESTful web services (RWS). RESTful Web service อนุญาตให้ระบบ Request และเข้าถึง Resource บนเว็บโดยมีชุดคำสั่งที่กำหนดเอาไว้แล้วล่วงหน้า การโต้ตอบของระบบที่ใช้ REST อยู่บนพื้นฐานของ Hypertext Transfer Protocol. Request จะส่งคำขอไปยัง URI ที่กำหนดไว้ และนำ response กลับมาเป็น Payload ในรูปแบบ HTML, XML, JSON หรือฟอร์แมต อื่น ๆ

RESTful ประกอบไปด้วย

Client ผู้ที่จะเข้ามา Request resources

Server ผู้ให้บริการ Resources

Architectural constraints of RESTful API 6 ข้อของ RESTful API ซึ่งถือเป็นสิ่งที่สำคัญในการสร้าง RESTful API ตามมาตรฐานซึ่งทำให้ง่ายต่อการพัฒนา และทำให้เป็นที่ยอมรับ

Client-server architecture: Client ไม่จำเป็นต้องรู้อะไรเกี่ยวกับ Business logic ภายใน ไม่มีหน้าที่เกี่ยวกับการจัดเก็บข้อมูล ส่วน Server มีหน้าที่เก็บ Resource และไม่จำเป็นต้องรู้อะไรเกี่ยวกับ UI Front-end หรือสถานะของผู้เรียก

Statelessness: ส่ง Request รับ Response จาก Server แล้วเลิก

Cacheability: สามารถ cache response ได้ การ Response จะต้องสามารถกำหนดได้ว่าจะ Cache หรือไม่ เพื่อป้องกันไม่ให้ User หรือ Client ได้รับข้อมูลเก่า

Layered system: ปกติ Client ไม่รู้ว่าที่ทำการเชื่อมต่อนั้น ได้เชื่อมต่อโดยตรงกับ Server ปลายทาง หรือไปยังตัวกลางอื่น ๆ ระหว่างทาง, Server ตัวกลางควรสามารถปรับปรุงความสามารถในการขยายระบบได้ โดยการใช้งานการทำ Load balance

Code on demand (optional): Server สามารถขยายได้ชั่วคราว หรือปรับแต่งการทำงานของ Client ได้ ตัวอย่างเช่น ทำ client-side scripts ใน JavaScript

Uniform interface: ถือเป็นข้อสำคัญจะที่แยกระหว่าง REST API และ Non-REST API มันแสดงให้เห็นถึงวิธีการที่จะคุยกับ Server โดยไม่คำนึงถึงประเภทของอุปกรณ์ หรือประเภทของ application Uniform interface ได้แยกออกไปอีก 4 อย่าง

1. Resource-Based: เช่น API/users
2. Manipulation of Resources Through Representations: เช่น User get user_id หรือ Request list of users แล้วทำการ Delete หรือ Modify user
3. Self-descriptive Messages: แต่ละ Message มีข้อมูลเพียงพอที่จะนำมาอธิบายวิธีการ Process message เพื่อให้ Server ทำการวิเคราะห์ได้ง่าย
4. Hypermedia as the Engine of Application State (HATEOAS): จำเป็นต้องมี Links สำหรับทุก ๆ Response เพื่อให้ Client สามารถค้นหาได้ง่าย

Article Resource ของการออกแบบ API ที่ดีควรจะเป็น CRUD (Create-Read-Update-Delete) CRUD คือ แนวคิดพื้นฐานที่ใช้ในการเขียนโปรแกรม หรือเว็บแอปพลิเคชัน เพื่อบ่งบอกลักษณะการกระทำ (เพิ่ม อ่าน แก้ไข ลบ) ข้อมูล CRUD ได้ถูกนำมาเป็น แนวคิดฟังก์ชันในหลากหลายการใช้งาน เช่น SQL, HTTP, RESTful APIs, DDS เป็นต้น [2]

ตาราง 2.1 การเทียบ CRUD กับ HTTP (Hypertext Transfer Protocol) and RESTful APIs

| CRUD | HTTP | คำอธิบาย |
|--------|--------|---|
| Create | POST | เป็นการสร้างข้อมูลใหม่ใน resource |
| Read | GET | เป็นการร้องขอข้อมูลจาก resource |
| Update | PUT | เป็นการอัปเดตข้อมูลที่มีอยู่แล้ว หรือสร้างใหม่ resource |
| Delete | DELETE | เป็นการลบข้อมูลที่มีอยู่แล้วใน resource |

ตัดส่วน Http status code ออก

2.2 Authentication, Authorization, Accounting (AAA)

1. การพิสูจน์ตัวตน (Authentication) หมายถึง กระบวนการยืนยันความถูกต้องของตัวบุคคล ที่กล่าวอ้างโดยทั่วไปกระบวนการพิสูจน์ความถูกต้องของตัวบุคคล จะอาศัยสิ่งที่เป็น Identifier ของบุคคล เช่น ชื่อ รหัสผ่าน หรือ PIN เป็นหลักฐานในการระบุตัวบุคคลของผู้ใช้ ก่อนที่จะดำเนินการตรวจสอบความถูกต้อง แสดงให้เห็นว่าการ Identification โดยส่วนใหญ่ถูกใช้ร่วมกับกระบวนการ Authentication [3]

วิธีการที่ใช้พิสูจน์ตัวบุคคลของผู้ใช้จะอาศัยสิ่งที่เกี่ยวข้องกับผู้ใช้ 3 ประการ ดังนี้

1. สิ่งที่คุณรู้ (Something you know) เช่น รหัสผ่าน (Password)
2. สิ่งที่มี (Something you have) เช่น บัตรเอทีเอ็ม บัตรประจำตัวประชาชน เป็นต้น
3. สิ่งที่เป็น (Something you are) เช่น การพิสูจน์ตัวตนด้วยลักษณะทางชีวภาพ (Biometric) เช่น ตรวจลายนิ้วมือ เส้นเลือดดำบนฝ่ามือ หรือลายม่านตา เป็นต้น

เปลี่ยนคำว่าได้แก่เป็นเช่น

2.2.1 สิ่งที่คุณรู้ (Something you know)

ที่นิยมนำมาใช้เป็นกลไกพื้นฐานของการพิสูจน์ตัวตนและสามารถนำมาใช้พิสูจน์ตัวตนได้ คือ รหัสผ่าน

รหัสผ่าน (Password) คือกลุ่มของตัวอักษรที่มีเพียงผู้ใดคนหนึ่งเท่านั้นที่ทราบรหัสผ่าน ซึ่งเสี่ยงต่อการถูกโจมตีมากที่สุด ดังนั้น โดยทั่วไปจึงกำหนดรหัสผ่านให้คาดเดายาก อย่างไรก็ตาม รหัสผ่านที่กำหนดขึ้นควรจะจดจำง่าย นั่นหมายถึง รหัสผ่านจะต้องสั้น กระชับ และอาจต้องเกี่ยวข้องกับตัวผู้ใช้ จึงจะทำให้จดจำได้ง่าย ซึ่งกลายเป็นข้อขัดแย้งกับการกำหนดรหัสผ่านให้คาดเดายากนอกจาก รหัสผ่าน แล้ว ยังมี passphrase ที่จัดว่าเป็นสิ่งที่คุณรู้ด้วย

Passphrase หมายถึง ชุดของตัวอักษร มีลักษณะคล้ายกับรหัสผ่าน แต่จะมีความยาวมากกว่ารหัสผ่านและคาดเดาได้ยากกว่า Passphrase ถูกนำมาใช้เพิ่มความปลอดภัยอีกระดับหนึ่งให้กับระบบ

สรุปคุณลักษณะของ Passphrase ได้ดังนี้

1. มีความยาวมากพอที่จะให้การคาดเดา Passphrase ปัจจุบันมีโปรแกรมสร้าง Passphrase จากวลี ต่างๆ

2. ต้องไม่สร้างจากคำคมที่โด่งดังในนวนิยาย ภาพยนตร์ หรือภาพยนตร์โฆษณา

3. ต้องไม่สามารถคาดเดาได้จากสัญชาตญาณของคน

4. ง่ายต่อการจดจำ และสะกดคำได้ถูกต้อง

5. สามารถใช้กฎเกณฑ์ต่างๆ เพื่อเพิ่มความยากต่อการคาดเดาและง่ายต่อการจดจำได้ และเมื่อผ่านกฎเกณฑ์ดังกล่าวแล้ว จะทำให้ได้ “Virtual Password”

ปัญหาที่เกิดกับการใช้รหัสผ่านเป็นสิ่งยืนยันตัวตน ทำให้ต้องมีการพิจารณาว่า รหัสผ่านที่ดีต้องมีลักษณะแบบใด ซึ่งสรุปได้ 3 มุมมอง ดังนี้

1. ความยาวของรหัสผ่าน รหัสที่มีความยาวมากกว่า ย่อมดีกว่ารหัสผ่านที่มีความยาวน้อยกว่า

2. ชุดของตัวอักษร ความแตกต่างของตัวอักษรในรหัสผ่าน 1 ชุด จะทำให้การคาดเดา รหัสผ่านยากมากขึ้น นั่นหมายความว่า ไม่ควรใช้ตัวอักษรซ้ำกันในรหัสผ่าน

3. การสุ่ม รหัสผ่านที่เกิดจากการกำหนดโดยไร้กฎเกณฑ์ หรือไร้เงื่อนไขที่ตายตัว จะช่วยให้แฮกเกอร์คาดเดาได้ยาก

2.2.2 สิ่งที่มีผู้ใช้มี (Something you have)

ที่สามารถนำมาเป็นกลไกในการพิสูจน์ตัวตนได้ เช่น บัตรเอทีเอ็ม, Smart Card และ Token สามารถแก้ปัญหาสิ่งที่ผู้ใช้รู้แต่ยากจะลืมได้เนื่องจากสิ่งที่มีผู้ใช้มี เช่น Smart Card ผู้ใช้มักจะพกติดตัวตลอดเวลา จึงสามารถนำออกใช้พิสูจน์ตัวตนได้โดยไม่ต้องจดจำ อย่างไรก็ตาม ปัญหาที่เกิดขึ้นกับสิ่งที่มีผู้ใช้มี คือ การลักขโมย

เทคโนโลยีประเภท Token และ Card ชนิดต่างๆ กลายเป็น 2 ปัจจัยหลักในกระบวนการพิสูจน์ตัวตน (Two-factor Authentication) กล่าวคือ การที่จะสามารถผ่านกระบวนการพิสูจน์ตัวตนได้จะต้องมี อุปกรณ์ ก็คือตัวบัตร และรหัสลับบางอย่าง เช่น PIN Code ดังนั้น หากขาดปัจจัยอย่างใดอย่างหนึ่งไป ก็จะไม่สามารถเข้าสู่ระบบได้

Token แบ่งการทำงานออกเป็น 2 รูปแบบ คือ **ย้าย Token มาใส่ตรงนี้**

1. Synchronous Token เครื่อง Client/Token ที่ผู้ใช้พกพาไปกับเครื่อง Server จะทำงานเข้าจังหวะตามเวลา ณ ขณะที่ผู้ใช้ต้องการเข้าสู่ระบบ โดยที่เครื่อง Token และ Server จะมี Token Code จัดเก็บไว้เหมือนกัน เมื่อผู้ใช้ต้องการเข้าสู่ระบบ จะใช้ Token สร้าง Token Code ส่งไปตรวจสอบที่เครื่อง Server ซึ่งจะตรวจสอบ Token Code กับเวลาที่ส่งไป หากตรงกับเครื่อง Server มีก็จะอนุญาตให้สู่ระบบได้

2. Asynchronous Token ผู้ใช้จะต้องติดต่อเพื่อส่งคำร้องขอเข้าสู่ระบบไปยังเครื่อง Server จากนั้น Server จะส่งตัวเลขกลับมา ผู้ใช้จะป้อนตัวเลขนั้นเข้าสู่เครื่อง Token ซึ่งเครื่อง Token จะคำนวณตัวเลขชุดดังกล่าว ได้ผลลัพธ์เป็นตัวเลขอีกชุดหนึ่งออกมา เรียกว่า “Response Number” ผู้ใช้จึงสามารถนำ Response Number ไปเป็นรหัสผ่านเพื่อเข้าสู่ระบบต่อไป

2.2.3 สิ่งที่ใช้เป็น (Something you are)

ปัญหาที่เกิดขึ้นกับการพิสูจน์ตัวตนโดยใช้สิ่งที่ผู้ใช้รู้ เช่น รหัสผ่าน คือ การลืมรหัสผ่าน และปัญหาที่เกิดจากการพิสูจน์ตัวตน โดยใช้สิ่งที่ผู้ใช้มี เช่น บัตรเอทีเอ็ม คือ การลืมบัตรและทำหาย ดังนั้น จึงได้มีการพิสูจน์ตัวตนโดยใช้ สิ่งที่ใช้เป็น (Something you are) แทนนั้นคือการพิสูจน์จากลักษณะทางชีวภาพหรือทางร่างกายของผู้ใช้ เรียกว่า Biometric ยกตัวอย่างเช่น พิสูจน์ด้วยลายนิ้วมือ ลายมือ จดจำใบหน้า (Face Recognition)

การนำ Biometric มาใช้จะต้องพิจารณาถึงปัจจัย 3 ประการ คือ

1. ความน่าเชื่อถือของระบบ
2. ต้นทุนและความพร้อมใช้
3. ความเต็มใจของผู้ใช้

2. การกำหนดสิทธิ์ (Authorization) คือ การจำกัดสิทธิ์ในการกระทำใด ๆ ต่อระบบและข้อมูลในระบบของผู้ใช้ที่ผ่านการพิสูจน์ตัวตนมาแล้วแม้ว่าผู้ใช้จะผ่านการพิสูจน์ตัวตนและสามารถเข้าสู่ระบบได้แล้ว ก็ไม่สามารถกระทำการใด ๆ กับระบบและข้อมูลในระบบได้ทุกอย่างตามที่ต้องการ แต่จะทำได้เฉพาะเท่าที่ได้รับสิทธิ์ตามนโยบายความมั่นคงปลอดภัยและตามอำนาจหน้าที่ที่ตนรับผิดชอบเท่านั้น

การกำหนดสิทธิ์การเข้าใช้ระบบของผู้ใช้ 3 ลักษณะ ดังนี้

1. กำหนดสิทธิ์ผู้ใช้อย่างบุคคล เป็นการจำกัดสิทธิ์ในการใช้งานระบบของผู้ใช้แต่ละคน โดยระบบจะพิสูจน์ตัวตนของผู้ใช้แต่ละรายว่าเป็นผู้ใช้ที่ได้รับอนุญาตที่แท้จริงหรือไม่ จากนั้นก็จะอนุญาตให้ผู้ใช้ที่แท้จริงเข้าสู่ระบบได้หากได้รับการยืนยันตัวตนอย่างถูกต้อง เมื่อเข้าสู่ระบบ ผู้ใช้จะสามารถใช้ทรัพยากรเฉพาะส่วนที่อนุญาตให้ใช้ได้เท่านั้น

2. กำหนดสิทธิ์สมาชิกของกลุ่ม ในกระบวนการพิสูจน์ตัวตนของการกำหนดสิทธิ์ลักษณะนี้ ระบบจะเปรียบเทียบหลักฐานการยืนยันตัวตนของสมาชิก กับบัญชีรายชื่อของสมาชิกในกลุ่มใด ๆ ที่จัดเก็บไว้ หากถูกต้องจะอนุญาตให้เข้าใช้ระบบได้ตามสิทธิ์ที่กลุ่มนั้นได้รับ

3. กำหนดสิทธิ์การใช้งานเข้าระบบ วิธีนี้จะมีการตรวจสอบหลักฐานการยืนยันตัวตนของผู้ใช้ที่ศูนย์กลางของระบบซึ่งหลักฐานดังกล่าวจะเป็นชุดของข้อมูลที่ทุกระบบสามารถตรวจสอบได้เหมือนกัน **ตัด LDAP ออก**

3. การจัดทำประวัติการเข้าใช้ระบบ (Accountability) เป็นส่วนที่ใช้บันทึกการเข้าใช้ระบบของผู้ใช้ (System Logs) เพื่อจัดทำเป็นหลักฐานการตรวจสอบ (Audit Trail) ที่จะเป็นประโยชน์ต่อการติดตามพฤติกรรมที่น่าสงสัยได้

2.3 OAuth

OAuth ย่อมาจาก **Open Authorization** **แก้จาก authentication** เป็นมาตรฐานการยืนยันสิทธิ์ (Authentication) และการตรวจสอบสิทธิ์ (Authorization) ซึ่งเป็นมาตรฐานแบบเปิด ปัจจุบันเป็นเวอร์ชัน 2 หรือ OAuth 2.0

OAuth เป็นมาตรฐาน ที่ใช้สำหรับการกำหนดสิทธิ์ให้ application สามารถร้องขอทรัพยากรของผู้ใช้จาก application หนึ่งได้โดยที่ application นั้นไม่จำเป็นต้องทราบรหัสผ่านของผู้ใช้ แต่จะยืนยันว่าตัวเองมีสิทธิ์หรือได้รับอนุญาตให้ใช้งานโดยใช้ access token แทนรหัสผ่านของผู้ใช้ ซึ่ง access token จะมีอายุการใช้งานเพียงช่วงเวลาหนึ่งเท่านั้น [4]

OAuth มาเพื่อแก้ปัญหาเช่น Login Facebook ค้างอยู่ถ้าเราต้องการจะ Post รูปผ่าน Instagram พร้อมกับแชร์ลง Wall ของ Facebook แล้วสามารถใช้งานได้ทันที เหมือนเป็น Account เดียวกัน พวกนั้นทำงานผ่าน OAuth 2.0 อธิบายคร่าวๆ OAuth 2.0 นั้นทำอยู่สองอย่าง

1. Federation Identity ช่วยให้ user สามารถ login เข้าใช้งาน application ได้โดย account อื่นๆ เช่น สามารถ login Instagram โดยใช้ Facebook account ได้
2. Delegate Authority ขอมให้ user สามารถใช้สิทธิในการเข้าถึง Resource ของ Services อื่น ได้โดยใช้สิทธิของ user คนนั้น เช่น สามารถ Share รูปใน Instagram จาก Facebook account เป็นต้น [5]

OAuth 2.0 มีการทำงาน 4 แบบ เรียกว่า grant_type โดยที่เราจะใช้แบบใดแบบหนึ่งหรือใช้ร่วมกันหลายๆแบบก็ได้ ขึ้นอยู่กับว่าจะทำ service แบบไหน

1. Authorization Code มักจะเอาไว้ใช้กับ web server เช่นถ้าเราทำ web application แล้วต้องการให้ผู้ใช้ Login ด้วย Gmail หรือ Facebook (ในกรณีองค์กรอาจจะ Login ด้วยอีเมลองค์กร) โดยโค้ดที่เป็น back-end จะใช้ grant_type นี้

2. Implicit มักจะเอาไว้ใช้กับ client ที่ไม่มี component ของ server เช่น ใช้งานบน mobile application หรือ web application ที่ทำงานในลักษณะ stateless โดยในการติดต่อกับกับ server นั้น client จะไม่ต้องเก็บข้อมูลลับ (secret key)

3. Password Credentials สำหรับ mobile application หรือ web application ที่เป็นทางการของเราเอง ในกรณีที่เรเป็นผู้ให้บริการ OAuth service เองเนื่องจากวิธีนี้เราสามารถจัดการสิทธิ์ต่างๆ ของผู้ใช้งานได้เพียงแค่อาม username และ password โดยที่ไม่ต้อง redirect ไปมา

4. Client Credentials เหมาะสำหรับทำบางอย่างใน application โดยที่ไม่มีส่วนเกี่ยวข้องกับผู้ใช้งาน เช่น ต้องการแก้ไขชื่อของ application ที่ถูก register ไว้ หรือแก้ไขข้อมูล metadata อื่นๆ ใน application ที่ไม่เกี่ยวข้องกับผู้ใช้เลย

OpenID Connect (OIDC) เป็นส่วนหนึ่งของ OAuth 2.0 โดยที่ใช้กันอยู่ในปัจจุบัน คือ OIDC v1.0 version OIDC จะมี API ที่ใช้ง่ายกว่า ใช้กับ mobile ได้ดีกว่า และทำงานร่วมกับ OAuth 2.0 ด้วยตัว protocol เองเลย ไม่ต้องมี extension มาเสริม โดยมีวัตถุประสงค์เพื่อช่วยให้สามารถใช้ Credential จาก Web หนึ่งในการเข้าใช้บริการของอีก Web Site หนึ่งได้ โดยผู้ใช้งานมี Credential เพียง Web Site เดียว เพื่อลดการ manage password ซ้ำ Web Site โดยในตัวของ Protocol ยังมีกระบวนการ verification เพื่อทวนสอบในมุมมองของ Security ด้วย ว่าผู้ใช้งานใช้ Browser, Mobile, JavaScript client อะไร ทำการเชื่อมต่อมายัง Application ของเรา รวมถึงมี Optional features เช่น เข้ารหัส Identity data, Discovery OpenID provider และ Session Management ได้อีกด้วย

2.4 Blockchain

เทคโนโลยีบล็อกเชน (Blockchain) เป็นเทคโนโลยีที่สร้างขึ้นมาเพื่อสร้างความปลอดภัยและน่าเชื่อถือ เป็นเทคโนโลยีการประมวลผลแบบกระจายศูนย์ Distributed Ledger Technology (DLT) บล็อกเชน (Blockchain) คือการนำข้อมูลมาเก็บใส่กล่อง (Block) แล้วส่งต่อกันเปรียบเสมือนโซ่ (Chain) [6] นั่นเองโดยมีการเข้ารหัสความปลอดภัยทางคอมพิวเตอร์ ที่ทำให้ทราบว่าข้อมูลถูกเก็บเวลาไหน มีการแก้ไขเปลี่ยนแปลงหรือไม่ ใครเป็นเจ้าของและมีสิทธิในข้อมูลจริง ๆ โดยข้อมูลต่าง ๆ เหล่านั้น จะถูกทำสำเนาเก็บไว้ใน Node ที่อยู่ในเครือข่าย ซึ่งหากมีการอัปเดตข้อมูลจากเจ้าของข้อมูล ข้อมูลที่ถูกทำสำเนาเก็บไว้ใน Node จะถูกอัปเดตอัตโนมัติตามไปด้วย นั่นหมายความว่าหากมีการแก้ไขข้อมูลจากบุคคลภายนอกขึ้นมา จะต้องทำการแก้ไขในทุก Node ที่เก็บข้อมูลไว้อย่างน้อย 51% ของ Node ทั้งหมดเพื่อให้เครือข่ายยอมรับการเปลี่ยนแปลงของข้อมูลใหม่ซึ่งไม่สามารถทำได้เลย เพราะเครือข่ายบล็อกเชนนี้นั้นมีขนาดที่ใหญ่มากคอมพิวเตอร์ในปัจจุบันไม่สามารถทำได้เลย นี่จึงเป็นเหตุผลที่เทคโนโลยีบล็อกเชนนี้นั้นมีความปลอดภัย และน่าเชื่อถือปัจจุบันเทคโนโลยีบล็อกเชนนี้นั้นได้ใช้กันอย่างแพร่หลายไม่เพียงแค่ในด้านการธุรกรรมการเงินเท่านั้น ยังมีด้านอสังหาริมทรัพย์ที่

ใช้เทคโนโลยีบล็อกเชนมาเก็บข้อมูลต่าง ๆ แทนใบโอน หรือในด้านการศึกษา ได้มีการออกใบ Certificate และ Transcript บนเทคโนโลยีบล็อกเชนเรียบร้อยแล้วด้วย

2.5 Smart Contract

Smart Contract เป็นสัญญาอัจฉริยะที่ดำเนินการอัตโนมัติตามข้อกำหนดที่ระบุไว้และข้อมูลจะเก็บไว้บน Ethereum ของบล็อกเชน (Blockchain) ไม่สามารถเปลี่ยนแปลงได้ Smart Contract เป้าหมายคือลดตัวกลาง ลดค่าใช้จ่าย เพิ่มความโปร่งใส และลดข้อผิดพลาดที่เกิดขึ้นผ่าน บล็อกเชน (Blockchain) [7]

2.5.1 การทำงานของ Smart Contract [8]

2.5.1.1 การสร้างข้อตกลง

ข้อตกลงระหว่างสองฝ่ายจะถูกแปลงเป็นรหัสคอมพิวเตอร์ จากนั้นการทำธุรกรรมต่าง ๆ ที่เกิดขึ้นจะถูกบันทึกโดยอัตโนมัติในเครือข่าย Ethereum. Smart Contract แต่ละอันจะมีหมายเลขที่อยู่เป็นของตัวเอง และเมื่อใดก็ตามที่ Smart Contract ถูกบันทึกในเครือข่าย Ethereum ใครก็ตามที่มีที่อยู่ของตัว Smart Contract นั้น ๆ จะสามารถเข้าถึง Smart Contract ได้

2.5.1.2 Triggering Events

Smart Contract จะระบุถึงจุดประสงค์ พร้อมด้วยวันหมดอายุของสัญญาเพื่อให้ตัว Smart Contract ทำงานได้ด้วยตัวมันเองโดยพิจารณาจากข้อตกลงที่ถูกแปลงเป็นรหัส ถ้าคำสั่งชุดหนึ่งถูกส่งออกมา ก็จะให้ผลในรูปแบบ ๆ หนึ่งโดย Smart Contract ก็จะทำงานไปเรื่อย ๆ จนกว่าทั้งสองฝ่ายจะยุติสัญญา

2.5.1.3 การยุติข้อตกลง

เมื่อ Smart Contract ถูกสร้างมาแล้วทั้งสองฝ่ายจำเป็นต้องบรรลุจุดประสงค์ตามที่ได้ตกลงกันไว้ในตอนแรก แต่ถ้าหากฝ่ายใดฝ่ายหนึ่งไม่ได้ทำตามที่เราไว้ในภายในระยะเวลาที่ตกลง Blockchain จะคืนเงินไปให้อีกฝ่ายหนึ่ง

Ethereum เพิ่มหัวข้อก่อนพูดถึง web3

Ethereum เป็นแพลตฟอร์มแบบ open source บนเครือข่ายเทคโนโลยี Blockchain ที่เปิดให้นักพัฒนาสามารถใช้งานได้ มีระบบ Smart contract ที่ผู้ใช้งานสามารถเขียนโปรแกรมขึ้นในรูปแบบของโค้ดคอมพิวเตอร์และกำหนดเงื่อนไขต่าง ๆ แล้วนำไปอยู่บนระบบ Blockchain ซึ่งเมื่อมีใครเข้ามาทำตามเงื่อนไขสำเร็จ ก็จะได้อผลตอบแทนไป [8]

2.6 Web3.js

Web3.js เป็นไลบรารีที่ไว้ใช้ติดต่อกับ Node ใน Ethereum ซึ่งเป็น JavaScript API ที่พัฒนาขึ้นเองโดย Ethereum เวลาใช้ต้อง Import lib ของ Web3.js เพื่อเรียกใช้ API ให้ไปเรียกใช้ Methods ใน Smart Contracts ซึ่งสามารถยิง API ให้สร้าง Transaction, get ค่าตัวแปรต่าง ๆ เรียกใช้ Methods บน Smart Contracts ที่อยู่บน Ethereum ได้ [9]

2.7 งานวิจัยที่เกี่ยวข้อง ** ยังไม่ได้แก้

2.7.1 Analysis of Identity Management Systems Using Blockchain Technology

งานวิจัยนี้จัดทำขึ้นเพื่อตรวจสอบและวิเคราะห์ระบบการจัดการข้อมูลประจำตัวโดยใช้เทคโนโลยี Blockchain เพื่อช่วยตรวจสอบและกระจายอำนาจไม่ให้อยู่ที่ตัวกลางและควบคุมผู้ใช้งานในการทำธุรกรรมบน Blockchain อีกทางหนึ่ง งานวิจัยกล่าวถึงการเน้นโมเดล IDM 3 แบบ 1.uPort แพลตฟอร์มที่ผู้ใช้ลงทะเบียนบน Ethereum 2.Soverin กลุ่มของ Blockchain ที่ทุกคนสามารถใช้ได้โดยไม่ต้องขออนุญาต 3.Shocard แพลตฟอร์มที่สร้างขึ้นเพื่อช่วยให้แน่ใจว่ามีการยืนยันตัวตนโดยมีแนวคิดการผสมผสานระหว่างเทคโนโลยี blockchain เทคโนโลยีมือถือ และไปโอเมตริกในระบบระบุตัวตนแบบรวมศูนย์

2.7.2 BIDaaS: Blockchain Based ID As a Service

งานวิจัยนี้กล่าวถึงการใช้เทคโนโลยี Blockchain สร้างบริการขึ้นสำหรับจัดการข้อมูลประจำตัวแบบดิจิทัล จัดแบ่งเป็นสามส่วนโดย 1.สร้างผู้ให้บริการชื่อ BIDaaS ขึ้นมาเพื่อจัดการข้อมูลและรับรองความถูกต้องเพื่อให้บริการแก่ Web application ที่เข้าร่วม 2.Web application ที่เข้าร่วมจะเป็นเว็บที่ลงทะเบียนกับ BIDaaS เพื่อให้บริการผู้ให้บริการ 3.ผู้ให้บริการ เป็นผู้ใช้ที่ลงทะเบียนร่วมกับ BIDaaS แต่ไม่ได้ลงทะเบียนกับ Web application ที่เข้าร่วมกับ BIDaaS ซึ่งจะเหมือนกับตัวโครงการที่ผู้จัดทำแต่แตกต่างตรงที่ผู้พัฒนาได้จัดเก็บข้อมูลโดยใช้มาตรฐาน OAuth 2.0 เพื่อรองรับการทำงานร่วมกันกับแพลตฟอร์มอื่นได้สะดวกยิ่งขึ้น

2.7.3 A Survey relates work

ใน [1] เป็นงานสำรวจแนวทางการแก้ไขปัญหาอย่างเป็นระบบและเป็นไปตามเกณฑ์ การเปรียบเทียบความสามารถของบริการที่จัดตั้งขึ้น พร้อมทั้งการปฏิบัติตามข้อกำหนด/ความรับผิดชอบ กฎระเบียบ มาตรฐาน

ใน [2] เป็นงานสำรวจตั้งแต่ปี 2557 ถึง 2561 จากบทสรุปนั้นสรุปได้ว่าข้อมูลประจำตัวนั้นง่ายต่อการแฮ็กบัญชี บริการออนไลน์ในปัจจุบันต้องพึ่งพาผู้ให้บริการออนไลน์เพื่อจัดการข้อมูลประจำตัว Blockchain สามารถเสนอวิธีแก้ปัญหามาโดยการกระจายอำนาจของความเป็นเจ้าของของ

ข้อมูลประจำตัวและเสนอให้ใช้ได้ในระดับสากล Blockchain สามารถสร้างแพลตฟอร์มที่ปลอดภัยสำหรับออนไลน์ได้ผู้ให้บริการตรวจสอบผู้ใช้ นอกจากนี้เทคโนโลยียังสามารถช่วยสร้างความไว้วางใจให้กับผู้ใช้อีกด้วยผู้ใช้ควรมีสติที่ควบคุมอย่างเต็มที่ว่าใครมีสิทธิ์ใช้ข้อมูลของพวกเขา

บทที่ 3

วิธีการดำเนินการ

3.1 ภาพรวม

ผู้จัดทำได้แบ่งออกเป็น 3 ส่วนในส่วนแรกจะเป็น Web Application ที่ใช้ทดสอบระบบที่พัฒนาขึ้น ส่วนที่สอง API Service และส่วนที่สามส่วนของ Blockchain

ส่วนแรก Web Application ที่ใช้ทดสอบระบบพัฒนาด้วยภาษา Python โดยใช้ Framework เป็น Django

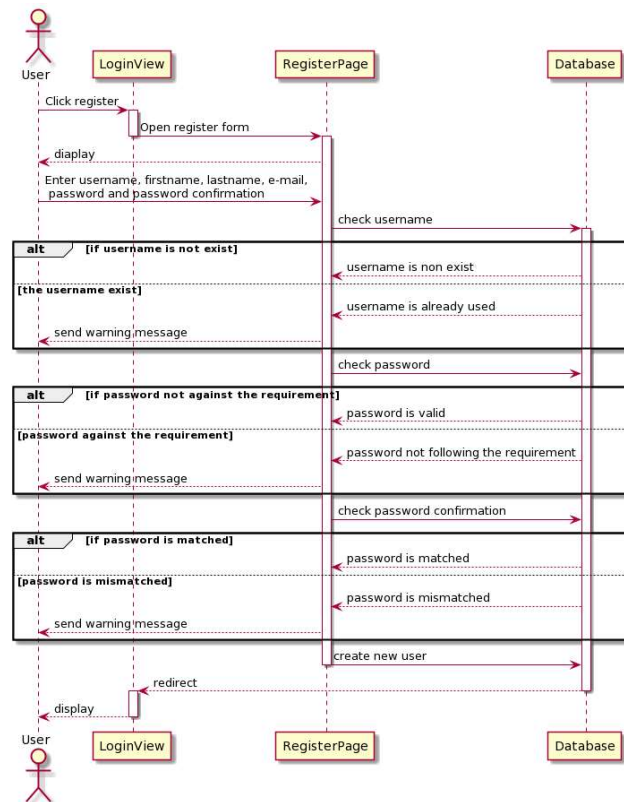
ส่วนที่สอง API Service พัฒนาโดยใช้ Node.js

ส่วนที่สาม Blockchain ออกแบบให้เป็น Database ที่ใช้เก็บข้อมูลผู้ใช้ที่สมัครและข้อมูล Service ต่าง ๆ ที่เข้ามาขอใช้บริการ

จากภาพรวมที่กล่าวมาผู้พัฒนาก็ได้ทำการออกแบบ sequence diagram ของทางฝั่ง Web Application ที่ใช้ทดสอบระบบและ sequence diagram ของ Blockchain โครงสร้างระบบที่จะกล่าวถึงดังนี้

3.2 การออกแบบ Sequence Diagram

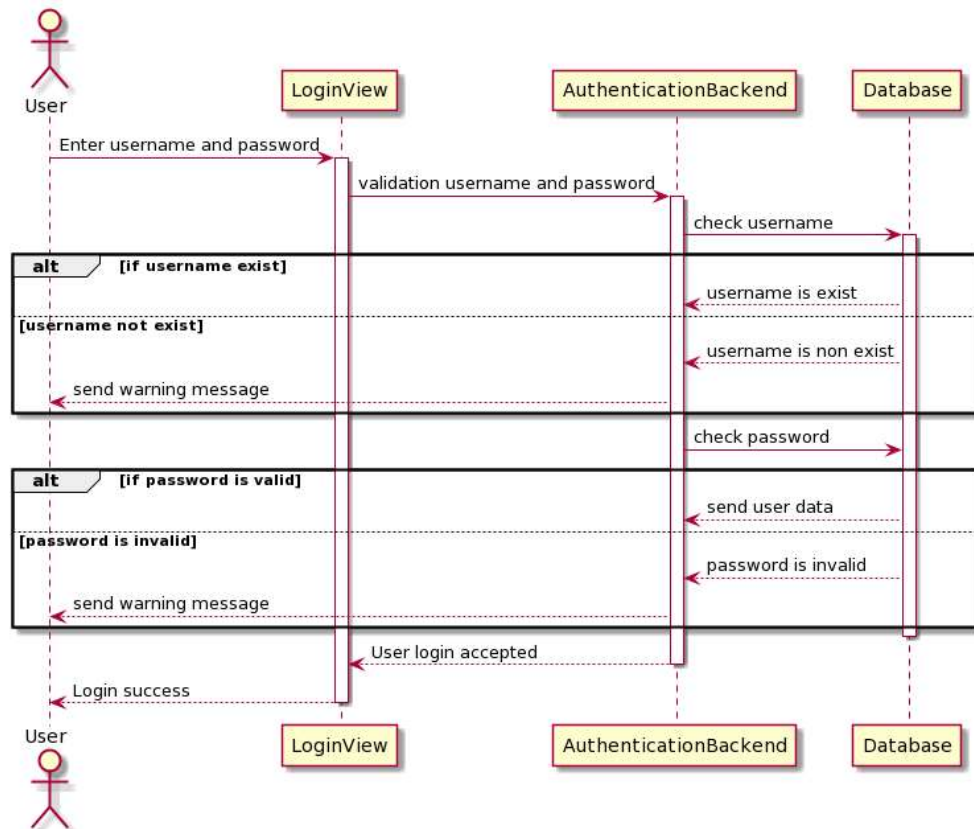
3.2.1 ขั้นตอนการสมัครบัญชี



รูปที่ 3.2.1 ขั้นตอนการสมัครบัญชี

1. การสมัครบัญชีผู้ใช้จะต้องกรอก username, password, first name, last name, email ที่ Register form เมื่อกรอกครบแล้วให้กด Register
2. จากนั้นระบบจะตรวจสอบความถูกต้องว่าตรงตาม format ที่กำหนดไว้หรือไม่ ถ้าไม่ถูกให้ผู้ใช้กรอกใหม่ ซึ่งระบบจะตรวจสอบอยู่ 3 อย่าง
 - 2.1 ตรวจสอบ username ว่ามีอยู่ในระบบหรือไม่ ถ้ามีระบบจะส่งข้อความแจ้งเตือนไปยัง User
 - 2.2 Password เป็นไปตามเงื่อนไขที่กำหนดหรือไม่ ถ้าไม่ตรงตามที่กำหนดระบบจะส่งข้อความแจ้งเตือนไปยัง User
 - 2.3 password กับ password confirmation ตรงกันหรือไม่ ถ้าไม่ตรงระบบจะส่งข้อความแจ้งเตือนไปยัง User
3. ถ้าข้อมูลทั้งหมดถูกต้องระบบจะส่งข้อมูลที่ผู้ใช้กรอกมาทั้งหมดไปเก็บไว้ใน Database
4. ทำการ redirect กลับไปยังหน้า Login

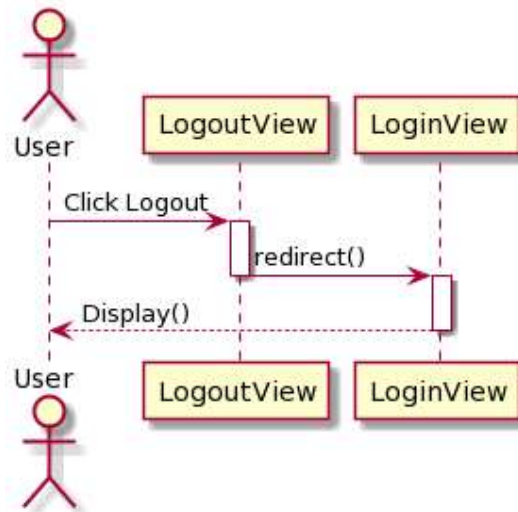
3.2.2 ขั้นตอนการ Login



รูปที่ 3.2.2 ขั้นตอนการ Login

1. ในการ Login เราต้องทำการกรอก username และ password ที่หน้า Login แล้วกด Login
2. จากนั้นระบบจะนำข้อมูลที่กรอกไปตรวจสอบว่าตรงกับข้อมูลที่มีอยู่ใน Database หรือไม่ หากไม่มีข้อมูลหรือไม่ถูกต้องจะไม่สามารถ Login ได้ และระบบจะทำการแจ้งเตือน User ว่ามีข้อผิดพลาดเกิดขึ้น
3. หาก Login สำเร็จจะทำการ redirect ไปยังหน้า Homepage

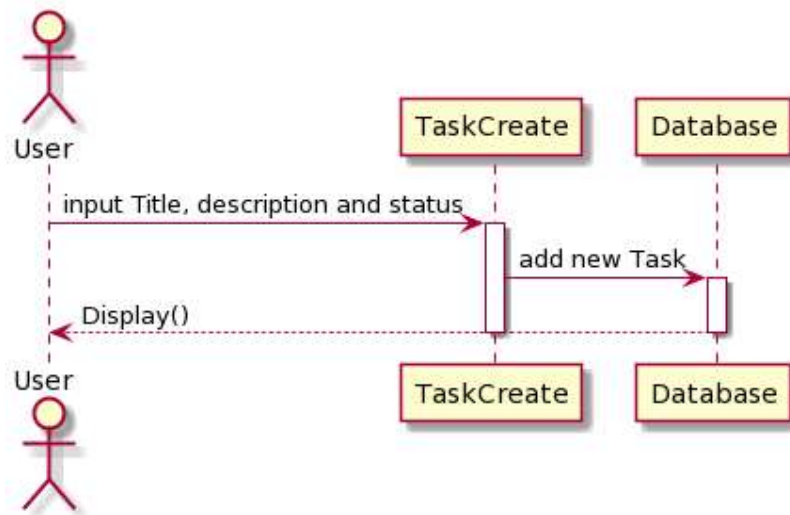
3.2.3 ขั้นตอนการ Logout



รูปที่ 3.2.3 ขั้นตอนการ Logout

1. เมื่อผู้ใช้คลิก Logout ระบบจะนำผู้ใช้ออกจากระบบกลับไปยังหน้า Login

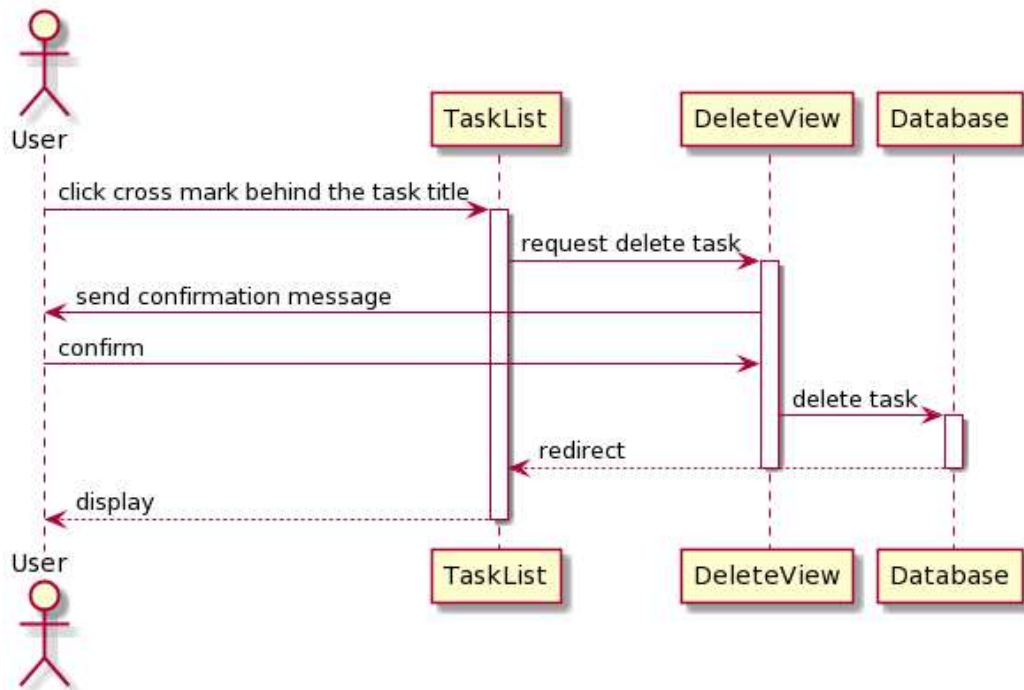
3.2.4 ขั้นตอนการสร้าง Task



รูปที่ 4 ขั้นตอนการสร้าง Task

1. ในการสร้าง Task ให้ผู้ใช้คลิก Create ซึ่งจะเป็นการเปิดฟอร์มสำหรับการสร้าง task ขึ้นมา
2. ให้ผู้ใช้กรอกข้อมูลตามแบบฟอร์มซึ่งประกอบไปด้วย ชื่อ task ,คำอธิบายของ task และสถานะของ task เมื่อกรอกเสร็จให้กด submit
3. จากนั้นระบบจะนำข้อมูลไปเก็บใน Database
4. ทำการ redirect กลับไปยังหน้า Homepage

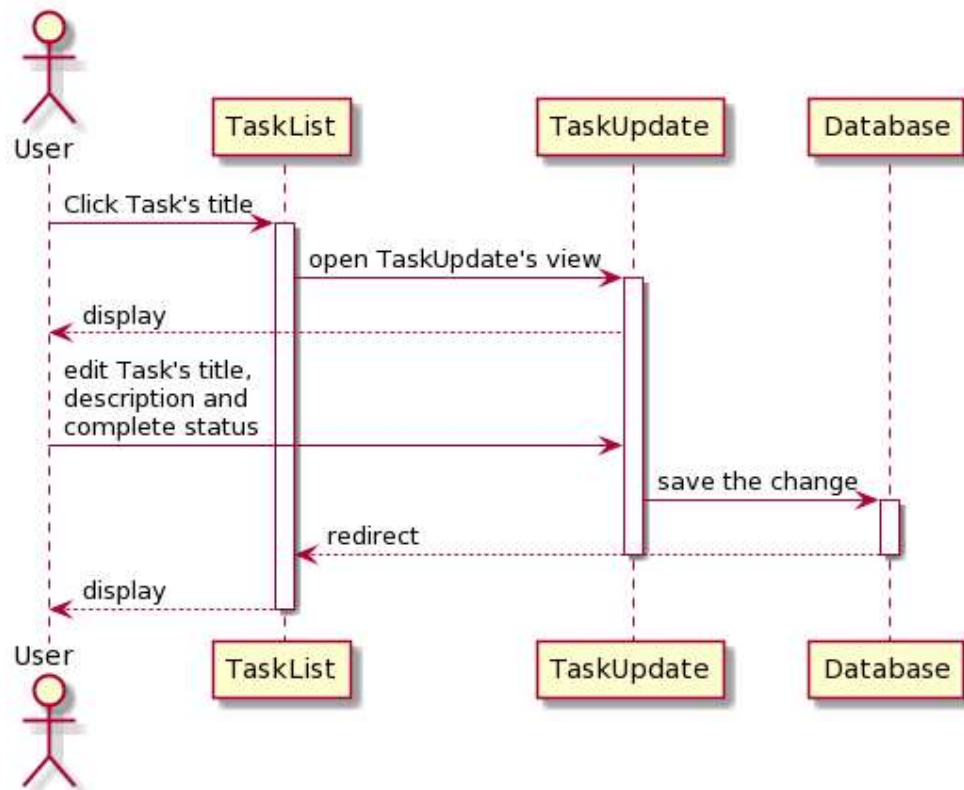
3.2.5 ขั้นตอนการลบ Task



รูปที่ 5 ขั้นตอนการลบ Task

1. คลิกกากบาทด้านหลังของ task ที่เราต้องการจะลบ
2. ระบบจะส่ง message กลับมาหา user เพื่อถามว่าต้องการจะลบจริงๆหรือไม่
3. กดยืนยัน
4. จากนั้นระบบจะทำการลบ task ดังกล่าวออกจาก Database
5. ทำการ redirect กลับมายัง Homepage

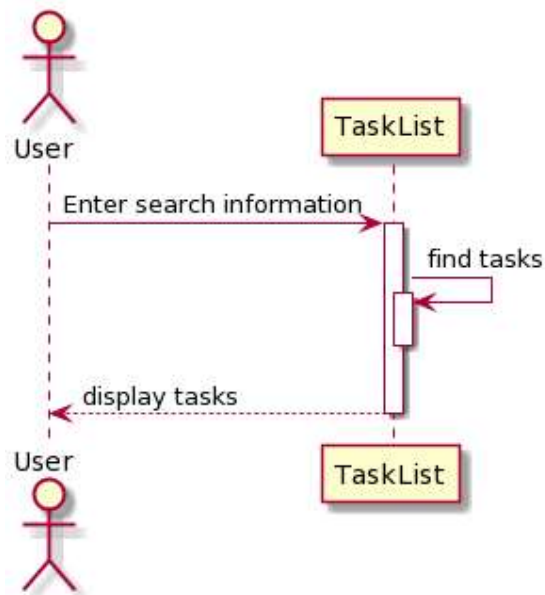
3.2.6 ขั้นตอนการแก้ไข Task



รูปที่ 6 ขั้นตอนการแก้ไข Task

1. คลิกชื่อ task ที่เราต้องการแก้ไขข้อมูล
2. ระบบจะเปิดหน้าต่างสำหรับการแก้ไขของ task ขึ้นมาให้เราทำการแก้ไขซึ่งสิ่งที่สามารถแก้ไขได้มีดังนี้ 1.ชื่อ task 2.คำอธิบายของ task 3.สถานะของ task
3. กดยืนยัน
4. ระบบจะนำข้อมูลไปแก้ไขใน Database แล้วทำการ redirect กลับมาหน้า Homepage

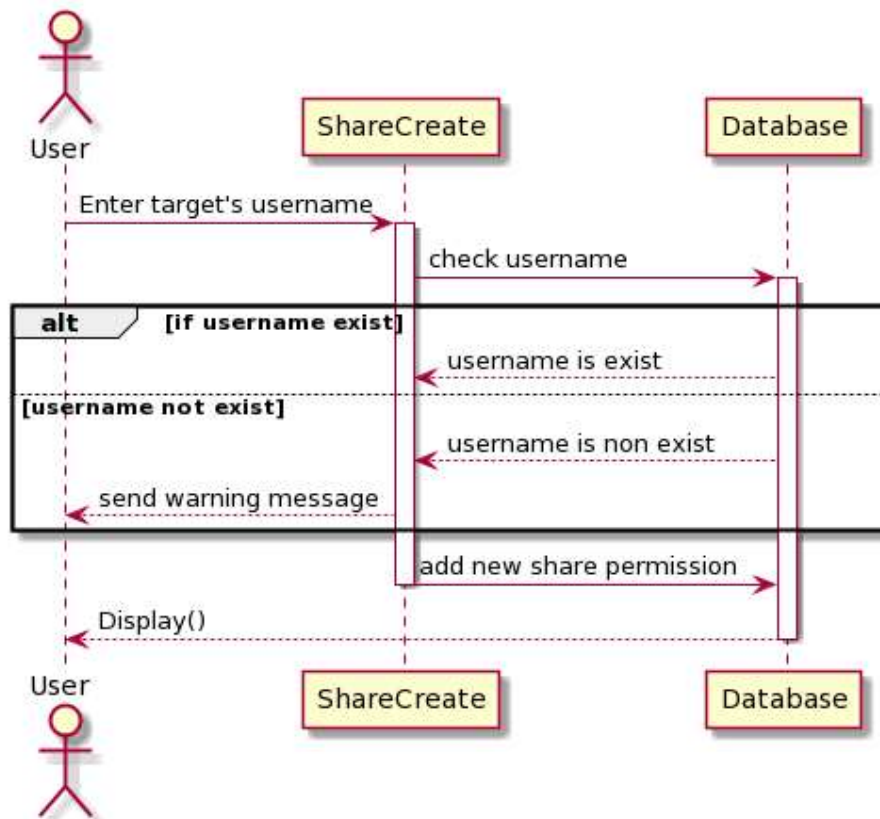
3.2.7 ขั้นตอนการค้นหา Task



รูปที่ 7 ขั้นตอนการค้นหา Task

1. กรอกชื่อ task ที่เราต้องการหาลงในช่องสำหรับการค้นหา
2. จากนั้นระบบจะทำการค้นหาชื่อ task ที่ user ต้องการจากใน List ที่ user คนนั้นมีอยู่
3. แสดงผล ซึ่งในกรณีที่ไม่มีพบข้อมูล task ที่ตรงกับข้อความที่เราค้นหาจะไม่มี task ปรากฏให้เห็น

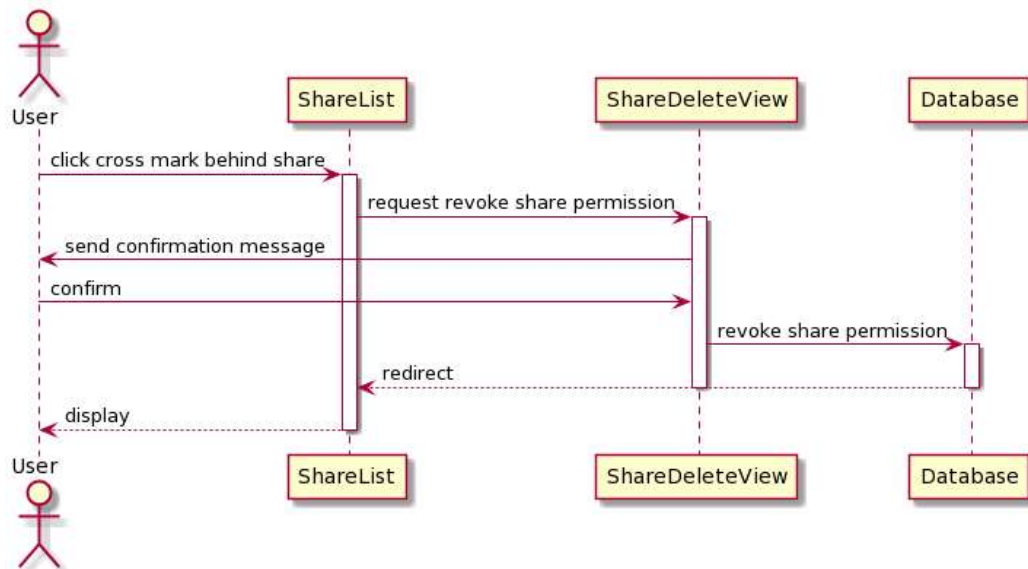
3.2.8 ขั้นตอนการแชร์ Task



รูปที่ 8 ขั้นตอนการแชร์ Task

1. ให้คลิก share to ตรง task ที่เราต้องการแชร์ให้ผู้อื่น
2. ระบบจะขึ้นฟอร์มให้กรอกชื่อ username ของ user ที่เราต้องการแชร์ให้ ซึ่งจะแชร์ได้ก็ต่อเมื่อ username ที่กรอกไปมีข้อมูลใน database แล้วหาก username ที่กรอกไปไม่มีข้อมูลอยู่ใน database จะไม่สามารถแชร์ได้ และระบบจะส่งข้อความแจ้งเตือนกลับไปยัง user

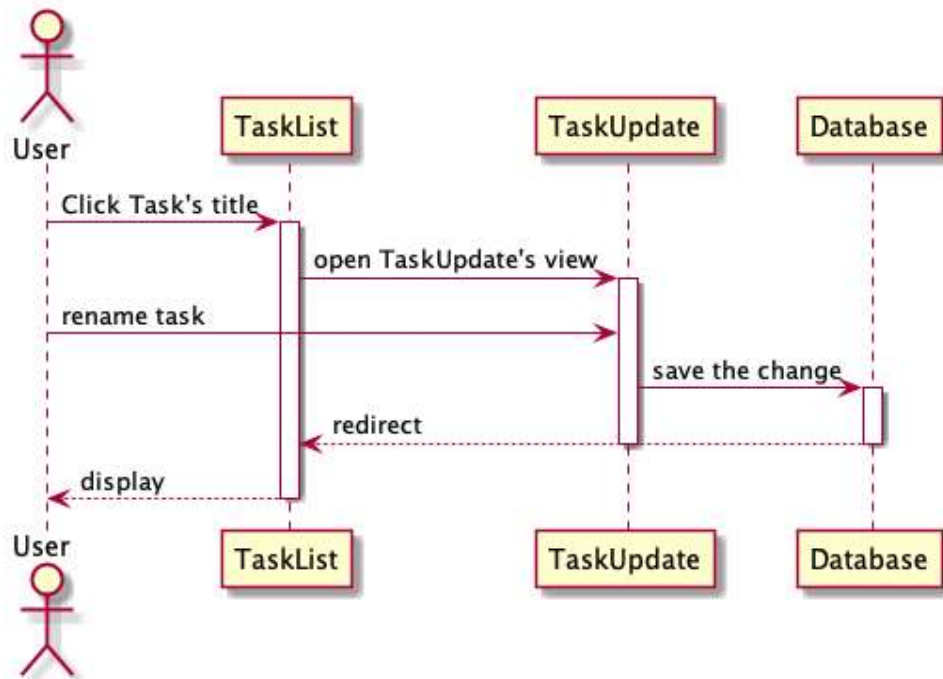
3.2.9 ขั้นตอนการยกเลิกการแชร์ Task



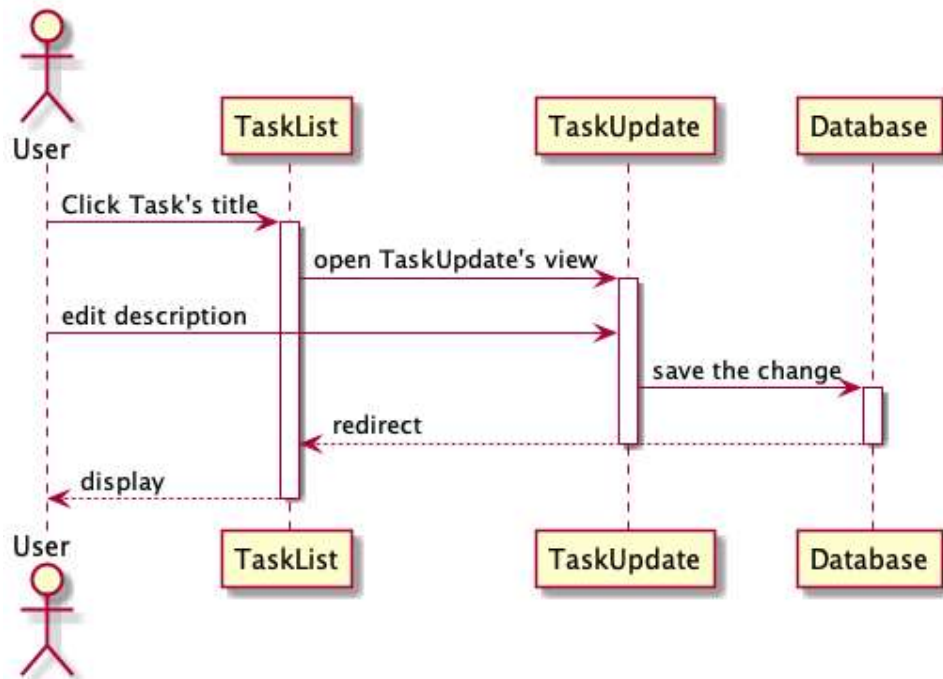
รูปที่ 9 ขั้นตอนการยกเลิกการแชร์ Task

1. ให้คลิกกากบาทด้านหลังการแชร์ที่เราต้องการยกเลิก
2. ระบบจะส่ง message กลับมาหา user เพื่อถามว่าต้องการจะยกเลิกจริงๆหรือไม่
3. กดยืนยัน
4. จากนั้นระบบจะทำการยกเลิกการแชร์ task ดังกล่าวกับ user คนนั้น

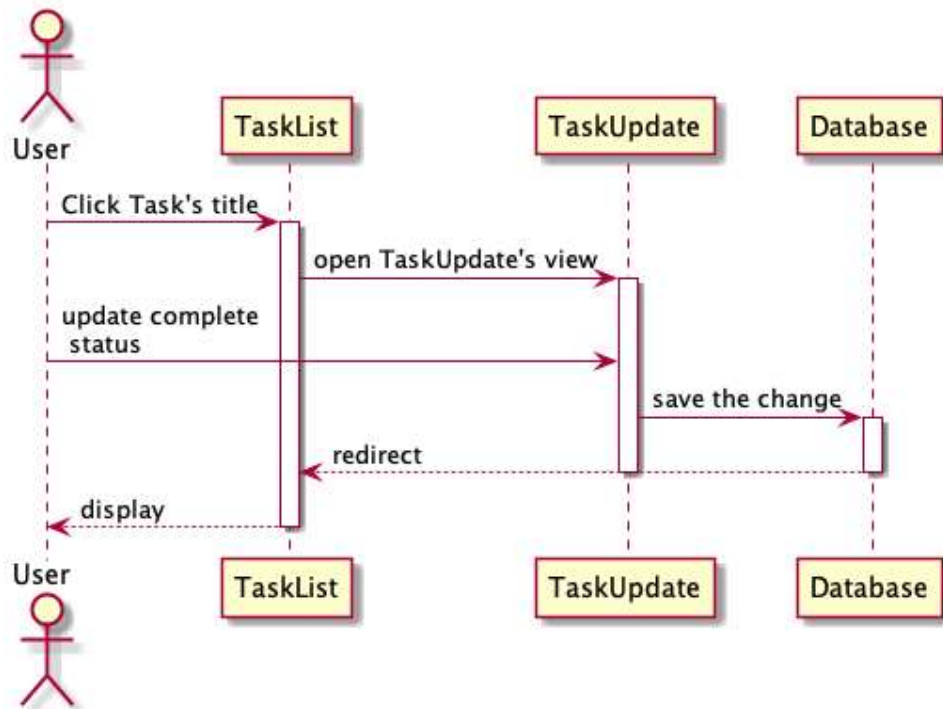
3.2.9 ขั้นตอนการ



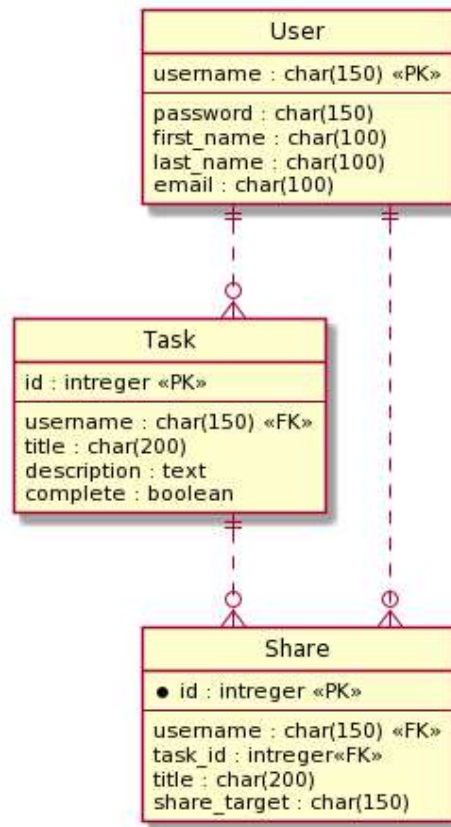
3.2.9 ขั้นตอนการ



3.2.9 ขั้นตอนการ



3.3 แผนภาพ ER diagram



รูปที่ 10 ER-diagram

ตาราง 3.3.1 User

เป็นตารางเก็บข้อมูล user

| name | type | key | description |
|------------|-----------|-----|-------------|
| username | Char(150) | PK | id ของ user |
| password | Char(150) | - | รหัสผ่าน |
| email | Char(100) | - | อีเมล |
| first_name | Char(100) | - | ชื่อ |
| last_name | Char(100) | - | นามสกุล |

ตาราง 3.3.2 Task

เป็นตารางเก็บเก็บข้อมูลของ Task

| name | type | key | description |
|-------------|-----------|-----|---------------------|
| id | integer | PK | id ของ task |
| username | Char(150) | FK | Id ของผู้สร้าง task |
| title | Char(200) | - | ชื่อ task |
| description | text | - | คำอธิบายของ task |
| complete | boolean | - | สถานะของ task |

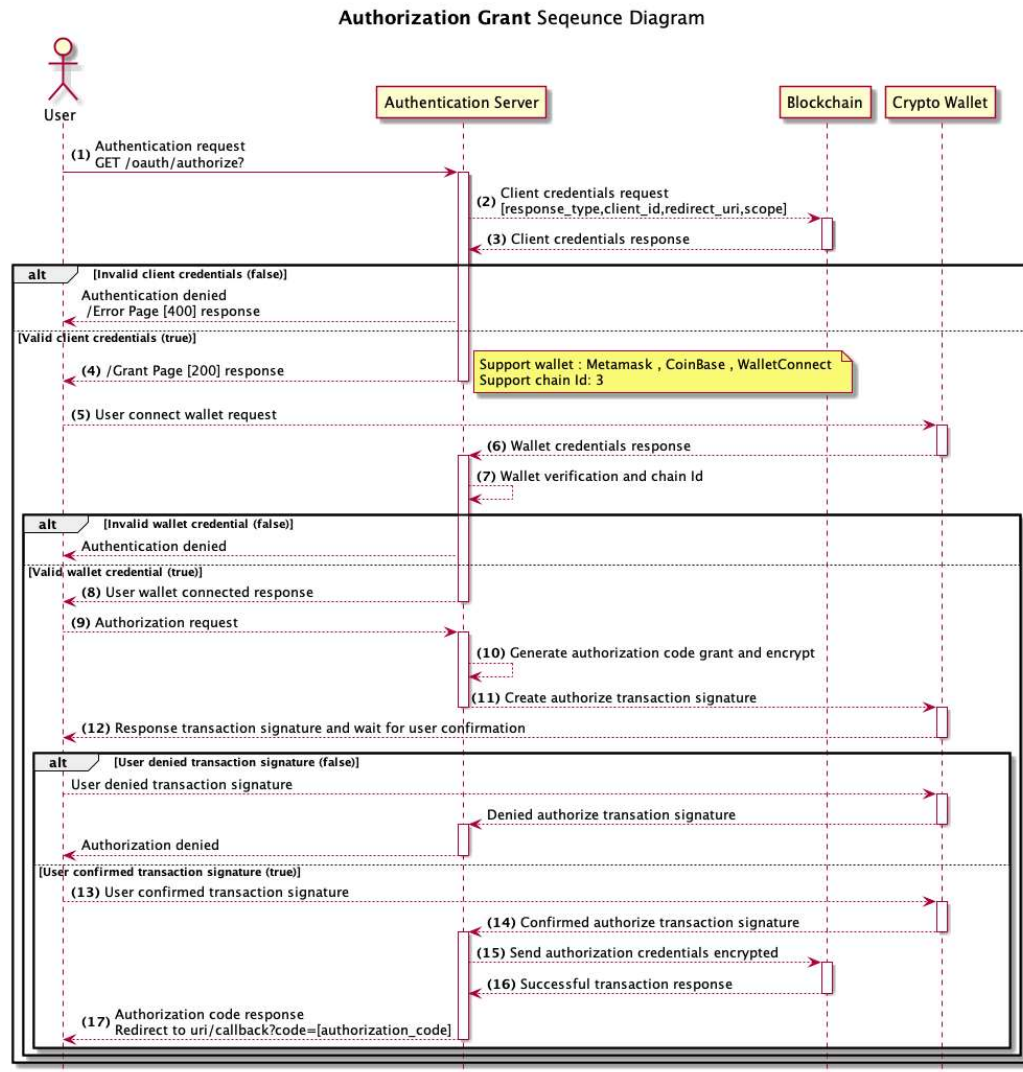
ตาราง 3.3.3 Share

เป็นตารางเก็บข้อมูลการแชร์ Task ของ user

| name | type | key | description |
|--------------|-----------|-----|---|
| id | integer | PK | id ของการแชร์ |
| task_id | integer | FK | Id ของ task |
| username | Char(150) | FK | Id ของผู้แชร์ task |
| title | Char(200) | - | ชื่อ task |
| Share_target | Char(150) | - | Username ของ user ที่เราต้องการ แชร์ task ด้วย |

3.4 การออกแบบ Sequence Diagram

3.4.1 ขั้นตอนการ



1. ผู้ใช้งานขอรับรองความถูกต้องที่ web server
2. ระบบทำการขอข้อมูลของผู้ใช้จาก blockchain
3. Blockchain ให้ข้อมูลผู้ใช้งานออกมาที่ server
4. ถ้าหากไม่เจอข้อมูล หน้าเว็บจะแสดง page error 400
5. ทำการเชื่อมต่อ wallet
6. wallet ส่งข้อมูลกลับมา
7. wallet เช็คว่า chain id และ network ถูกหรือไม่ ถ้าไม่ถูก หน้าเว็บจะขึ้น Authorization denied
8. แสดงหน้าการเชื่อมต่อ wallet สำเร็จ

9. ผู้ใช้ขออนุญาตใช้สิทธิ์
10. สร้างโค้ดอนุญาตใช้สิทธิ์แล้วทำการเข้ารหัสเก็บไว้
11. สร้าง transection ไปที่ wallet
12. สร้างหน้ายืนยันการสร้าง transection และรอผู้ใช้กดยืนยัน ถ้าผู้ใช้ปฏิเสธ ระบบก็จะ denied ผู้ใช้งาน
13. ผู้ใช้กดยืนยัน transection
14. wallet ยืนยันการขออนุญาตมาที่ server
15. ส่ง Credential เข้ารหัสไปเก็บไว้ที่ blockchain
16. transection สำเร็จ
17. โค้ดการ authorization แสดงให้ผู้ใช้งานเห็น

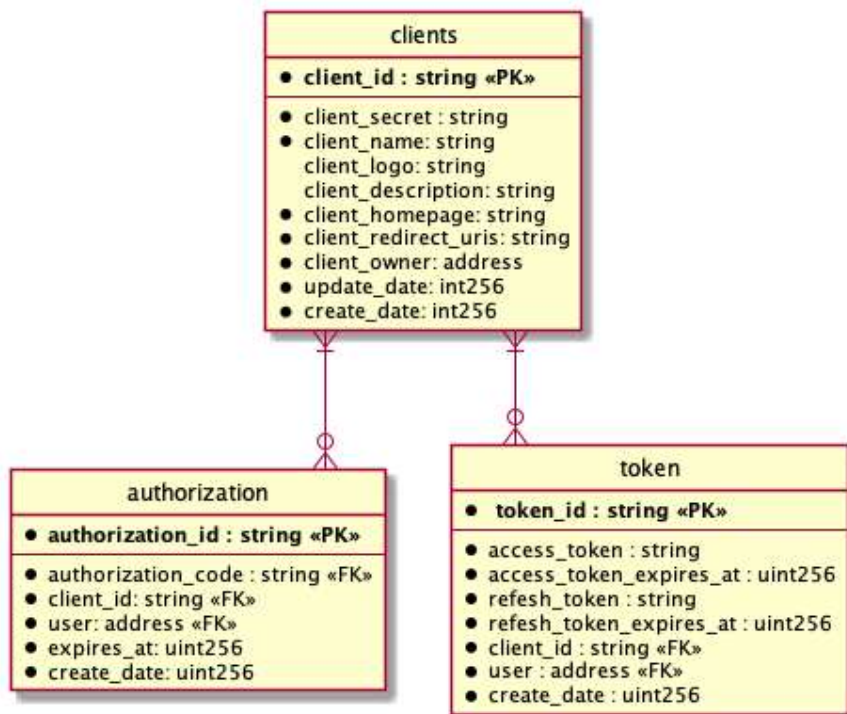
3.4.1 ขั้นตอนการ

3.4.1 ขั้นตอนการ

3.4.1 ขั้นตอนการ

3.4.1 ขั้นตอนการ

3.5 แผนภาพ ER diagram



รูปที่ 15 แผนภาพ ER diagram

ตาราง 3.5.1 Clients

เป็นตารางเก็บข้อมูล clients

| name | type | key | description |
|----------------------|---------|-----|---------------------|
| client_id | string | PK | id ของ client |
| client_secret | string | - | รหัสลับของ client |
| client_name | string | - | ชื่อ client |
| client_logo | string | - | โลโก้ |
| client_description | string | - | คำอธิบาย |
| client_homepage | string | - | หน้าแรกของ client |
| client_redirect_uris | string | - | Call back ส่ง token |
| client_owner | address | - | ชื่อเจ้าของ client |
| update_date | int256 | - | วันที่อัปเดตข้อมูล |
| create_date | int256 | - | วันที่สร้าง |

ตาราง 3.5.2 Authorization

เป็นตารางเก็บข้อมูลของ authorization

| name | type | key | description |
|--------------------|---------|-----|---------------------------|
| authorization_id | string | PK | id ของ authorization |
| authorization_code | string | FK | Code ของ authorization |
| client_id | string | FK | id ของ client |
| user | address | FK | ผู้ใช้งาน |
| expires_at | uint256 | - | เวลาหมดอายุ authorization |
| create_date | uint256 | - | วันที่สร้าง |

ตาราง 3.5.3 token

เป็นตารางเก็บข้อมูลของ token

| name | type | key | description |
|--------------------------|---------|-----|---------------------------------|
| token_id | string | PK | Id ของ token |
| access_token | string | - | Access token |
| access_token_expires_at | uint256 | - | เวลาหมดอายุของ Access token |
| refresh_token | string | - | refresh token |
| refresh_token_expires_at | uint256 | - | เวลาหมดอายุของ refresh token |
| client_id | string | FK | id ของ client |
| user | address | FK | ผู้ใช้งาน |
| create_date | uint256 | - | วันที่สร้าง |

บทที่ 4

ผลการดำเนินงาน

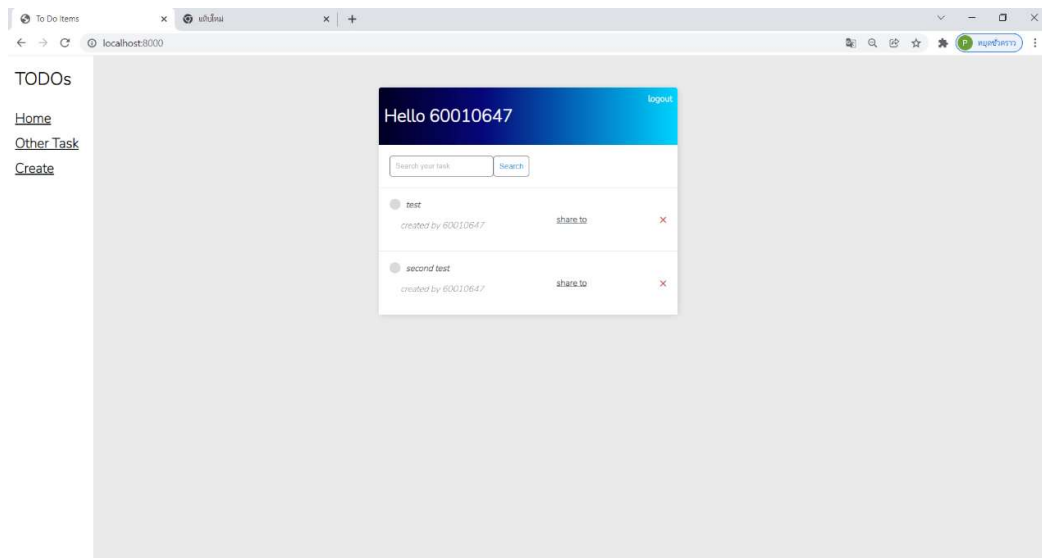
4.1 ผลการทดลอง

แยกส่วนการทำงานหลัก ๆ ออกเป็น Web application Demo และ Web service ทดสอบ แล้วทำการสังเกตผลลัพธ์ของแต่ละหน้าจอ ในส่วนแรกจะเป็นของ Web application Demo

4.1.1 หน้าหลัก

เมื่อเข้ามาใน Web Application ในหน้านี้จะเป็นหน้าแรกหลังจาก login เข้ามาในระบบ และเป็นส่วนที่จะเชื่อมโยงไปยังส่วนต่าง ๆ ของเว็บ ได้แก่

1. หน้า Home
2. ปุ่ม Create
3. ปุ่ม Other Task
4. ปุ่ม Logout



รูปที่ 4.1 หน้าแรกเมื่อเข้าสู่ระบบ

4.1.2 หน้าสมัครสมาชิก

ในหน้านี้ผู้ใช้บริการต้องกรอกข้อมูลเพื่อสมัครใช้งาน web application เมื่อสมัครเสร็จแล้วจะพาไปยังหน้า Login

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/register/'. The main content is a 'Register' form with a blue header. The form contains the following fields and elements:

- Username:** A text input field with a placeholder 'I' and a note: 'Required: 150 characters or fewer: Letters, digits and @/+/./ only'.
- First name:** A text input field.
- Last name:** A text input field.
- Email:** A text input field.
- Password:** A text input field with a list of requirements:
 - Your password can't be too similar to your other personal information.
 - Your password must contain at least 8 characters.
 - Your password can't be a commonly used password.
 - Your password can't be entirely numeric.
- Password confirmation:** A text input field with a placeholder 'enter' and a note: 'This same password as before, for verification'.
- Register:** A button to submit the form.
- Already have an account? Login:** A link to the login page.

รูปที่ 4.2 หน้าสมัครสมาชิก

4.1.3 หน้าเข้าสู่ระบบ

ในหน้านี้ผู้ใช้ที่สมัครสมาชิกแล้วสามารถเข้าสู่ระบบได้เลย หรือหากยังไม่ได้สมัครสมาชิก แต่มีบัญชีผู้ให้บริการของ Google หรือ Facebook ก็สามารถกดเพื่อเข้าใช้งานในระบบได้ทันที

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/login/'. The main content is a 'Login' form with a blue header. The form contains the following fields and elements:

- Username:** A text input field with a placeholder 'I'.
- Password:** A text input field.
- Login:** A button to submit the form.
- Social Login:** Two buttons for 'Facebook' and 'Google'.
- Don't have an Account? Register:** A link to the registration page.

รูปที่ 4.3 หน้าเข้าสู่ระบบ

4.1.4 หน้าการสร้าง Task งาน ยังไม่แคปรูปใหม่

ในหน้านี้เมื่อผู้ใช้เข้ามาจะสามารถสร้าง Task ได้โดยใส่ชื่อ Task รายละเอียดต่าง ๆ และสามารถเลือก ผู้ใช้คนอื่นที่จะแชร์ Task งานร่วมได้ด้วย

รูปที่ 4.4 หน้าการสร้าง Task งาน

4.1.5 การลบ Task

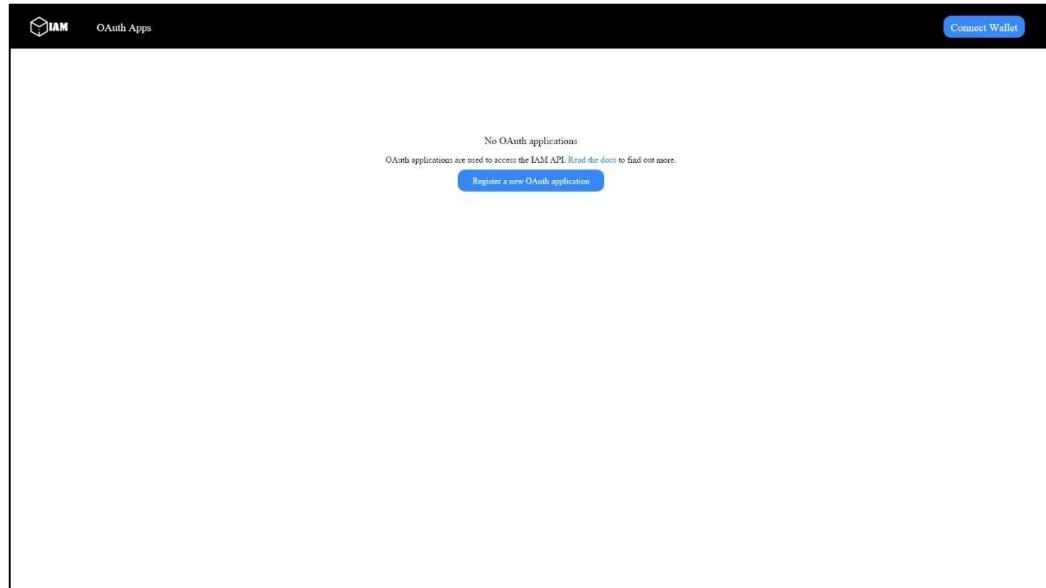
เมื่อผู้ใช้กดกากบาทมุมบนขวาของ Task ในหน้าแรก ระบบก็จะเปลี่ยนหน้าไปถามผู้ใช่ว่าจะลบหรือไม่

รูปที่ 4.5 การลบ Task

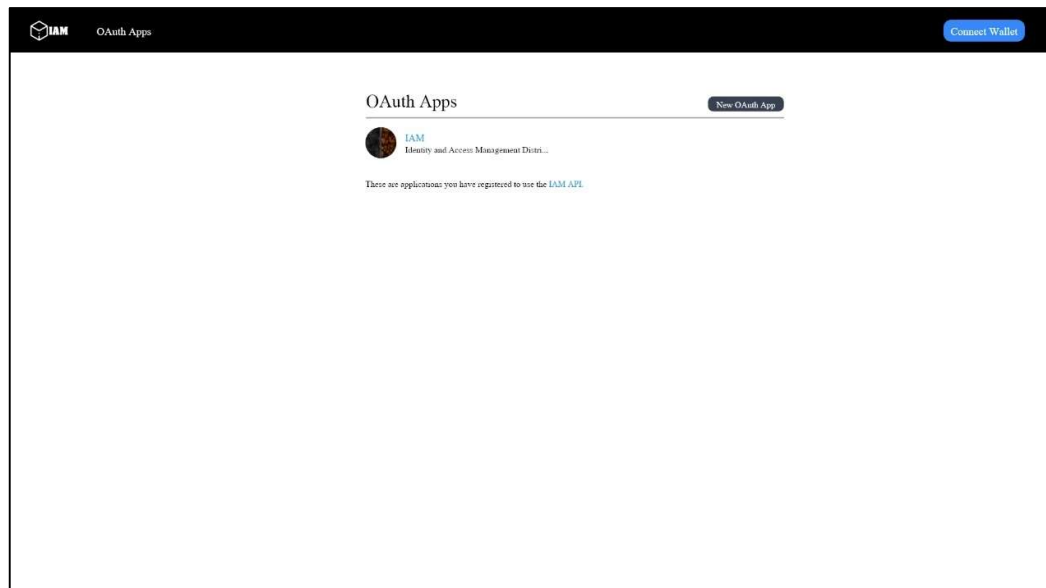
4.2 Web service

4.2.1 หน้า Home page ของ Web service

ในหน้า Home page จะมีปุ่มให้ทำการเชื่อมต่อกับ MetaMask เมื่อทำการเชื่อมต่อแล้วหากยังไม่เคยมีการสร้าง OAuth Application จะแสดงคำว่า No OAuth applications ดังรูป สามารถกดปุ่ม Register a new application เพื่อไปสร้าง OAuth Application ได้

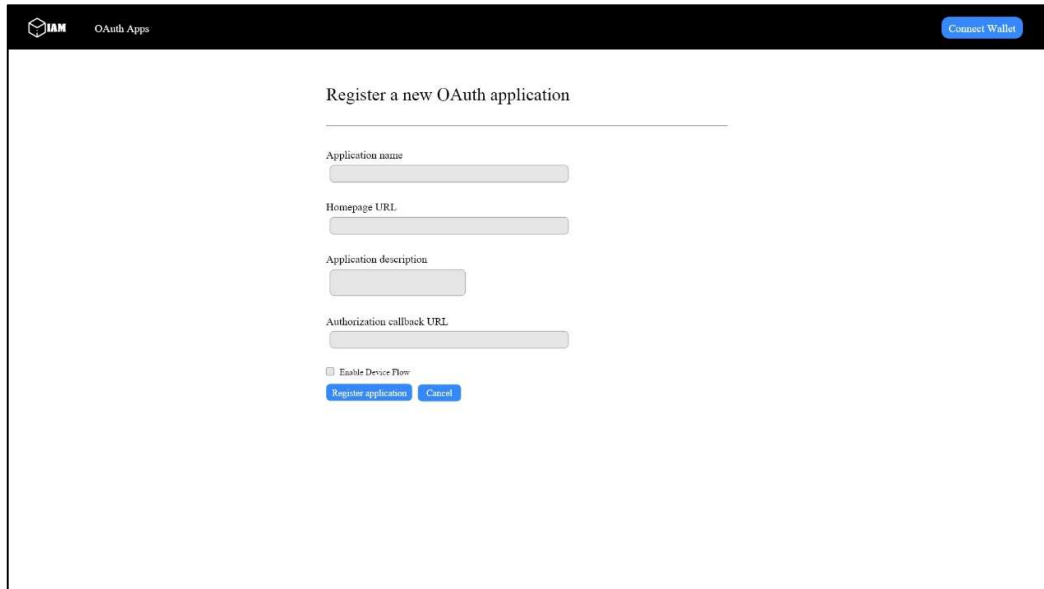


หรือถ้าหากเคยสร้างแล้ว จะแสดงข้อมูลที่เคยสร้างไว้ขึ้นมาแล้วมีปุ่ม New OAuth App เพื่อไว้สร้างเพิ่มด้วย



4.2.2 หน้า Register a new application

ในหน้านี้ จะให้กรอกข้อมูลของ Web application ของเราจะมี ชื่อของ Web application URL หน้าแรกของเว็บเรา คำอธิบายเว็บว่าตัวเว็บของเราเกี่ยวกับอะไรและ URL ในการให้สิทธิ์ Authorization กลับไป ทำการเรียกฟังก์ชันเพื่อเก็บข้อมูลไว้บน blockchain




The screenshot shows the 'Register a new OAuth application' page in the AWS IAM console. The page has a dark header with the IAM logo and 'OAuth Apps' text. A 'Connect Wallet' button is in the top right. The main content area is white and contains the following fields:

- Application name:** A text input field.
- Homepage URL:** A text input field.
- Application description:** A text input field.
- Authorization callback URL:** A text input field.
- Enable Device Flow:** A checkbox.
- Buttons:** 'Register application' (blue) and 'Cancel' (grey).


4.2.3 หน้า Info OAuth App

ในหน้านี้จะแสดงข้อมูลต่าง ๆ ของ Web application ที่ได้ลงทะเบียนไป จะมี ชื่อผู้ที่เป็นเจ้าของ application จำนวนผู้ใช้งานที่ขอใช้สิทธิ์ Credential ต่าง ๆ มีให้อัปโหลดรูปภาพ และแสดงข้อมูลที่เรากรอกไว้ในที่สมัครตอนแรกด้วย สามารถแก้ไขและอัปเดตได้

 OAuth Apps

Connect Wallet

IAM

 Clutchbox owns this application.

Transfer ownership

0 user

Revoke all user tokens

Client ID


cx123456cx7x09x123x456x

Client Secrets

Generate a new client secret

You need a client secret to authenticate as the application to the API.

Application logo

 Drag & drop

Upload new logo

You can also drag and drop a picture from your computer.

Application name

Homepage URL

Application description

Authorization callback URL

☐ Enable Device Flow

Update application

Cancel

บทที่ 5

สรุปผลการทดลอง

บทนี้จะเป็นการกล่าวถึงเนื้อหาที่เป็นผลสรุปและภาพรวมของระบบจัดการสิทธิ์การเข้าถึงด้วยแอปพลิเคชันแบบกระจายศูนย์บนบล็อกเชนสิ่งที่จะกล่าวถึงในส่วนแรกจะเป็นสรุปผลสิ่งที่ได้ทำไปแล้ว, สรุปผลการพัฒนาระบบ, ปัญหาที่พบ และ แนวทางในการพัฒนาต่อ

5.1 สรุปผลการพัฒนาระบบ ยังไม่ได้มีการทดสอบระบบ

ในส่วนของ Web application demo ที่ใช้ทดสอบระบบ ได้ดำเนินการสร้างหน้า Home page, Register page, Login page, Task create page, Delete task page, Update task page, Share page แต่ยังมีบางจุดที่ยังไม่เป็นไปตามที่คาดหวังและจะทำการพัฒนาต่อไป

ในส่วนของ Web service ได้ทำการสร้างหน้า register และหน้า login ขึ้นมาแล้วสามารถจัดเก็บข้อมูลบน Ethereum test net ได้และแสดงข้อมูลทั้งหมดเมื่อเข้าสู่ระบบได้

5.2 ปัญหาที่พบ

1. การทำ Service มีปัญหาในเรื่องของ Concept ของ OAuth ที่ต้องนำมาปรับใช้กับตัวโครงงานให้ได้ตามมาตรฐาน และระบบการเก็บข้อมูลที่มีการจัดเก็บที่ต่างไปจากการเก็บข้อมูลโดยทั่วไป
2. การ Login ด้วย Google เมื่อเข้าสู่ระบบครั้งแรกผ่านไปสามารถเข้าใช้งานได้ปกติ แต่เมื่อเข้าสู่ระบบครั้งที่สอง ระบบยืนยันสิทธิ์ของ Google ไม่ยุติการเชื่อมต่อให้แม้ Access Token จะหมดอายุแล้วก็ตาม

5.3 แนวทางในการพัฒนาต่อ

ในวิชาโครงงาน 2 ทางผู้จัดทำจะปรับแก้ในส่วนของ web application demo ที่ใช้ทดสอบระบบ โดยปรับปรุงเว็บเพจและเพิ่มประสิทธิภาพให้ได้ตามที่คาดหวัง และจะทำการเปลี่ยนการเข้าสู่ระบบให้ไปใช้ของที่ผู้จัดทำพัฒนาขึ้นเองและฝั่ง service ปรับ Font-end และทำการเชื่อมต่อ Web Demo และ Service ที่ออกแบบ

บรรณานุกรม

Samia El Haddouti and M. Dafir Ech-Cherif El Kettani. 2019. **Analysis of Identity Management Systems Using Blockchain Technology**. [online].

Available: <https://ieeexplore.ieee.org/document/8742375>

Jong-Hyouk Lee. 2017. **BIDaaS: Blockchain Based ID As a Service**. [online].

Available: <https://ieeexplore.ieee.org/document/8187625>

[1] Shu Yun Lim, Pascal Tankam Fotsing, Abdullah Almasri, Omar Musa, Miss Laiha Mat Kiah, Tan Fong Ang and Reza Ismail. 2018. **Blockchain Technology the Identity Management and Authentication Service Disruptor: A Survey**. [online].

Available: https://www.researchgate.net/publication/328919940_Blockchain_Technology_the_Identity_Management_and_Authentication_Service_Disruptor_A_Survey

[2] Michael Kuperberg. 2019. **Blockchain-Based Identity Management: A Survey From the Enterprise and Ecosystem Perspective**. [online].

Available: https://www.researchgate.net/publication/335077636_Blockchain-Based_Identity_Management_A_Survey_From_the_Enterprise_and_Ecosystem_Perspective

[1] Sakul Montha. 2019. REST กับ RESTful API ต่างกันนะรู้ยัง. [online].

Available: <https://iamgique.medium.com/restful-api-กับ-rest-api-ต่างกันนะรู้ยัง-2c70c42990e3>

[2] nich. 2019. Restful api คืออะไร. [online].

Available: <https://www.4x-treme.com/restful-api-คืออะไร/>

[3] Peerapon Rattanapaiboon. 2017. บทที่ 07 การควบคุมการเข้าถึง. [online].

Available: <https://sites.google.com/site/peempeerapon/kar-brihar-khwam-mankhng-sarsnthes/bth-thi-7>

[4] <http://sdm.ubu.ac.th/blog/suttichai-160>

[5] <https://blog.tamacorp.co/รู้จัก-oauth-2-0-กันก่อน-part-i/>

[6] <https://www.finnomena.com/coinman/blockchain/>

- [7] <https://www.mdsiglobal.com/smart-contract/>
- [8] <https://support.bitkub.com/hc/th/articles/360004414672-Smart-Contract-คืออะไร-และทำงานอย่างไร->
- [9] <https://zipmex.com/th/coin-info/eth-th/>
- [10] <https://poolsawat-com.medium.com/เขียน-โค้ด-javascript-web3-js-c1dc1fb20493>

ภาคผนวก ก

ขั้นตอนการ deploy

ทุกกลุ่มจัดทำภาคผนวก ก ในรายงาน อธิบายขั้นตอนการ deploy ที่ทำจริง พร้อมรูปประกอบ

กลุ่มที่ไม่มี App ก็ให้ทำภาคผนวก ก โดยให้ระบุว่า ไม่มี Application ในโครงงาน

สำหรับ Web App ตอน demo ให้ส่ง link ของ web app ให้กรรมการด้วย

สำหรับ Mobile App ตอน demo ให้สาธิตจาก smartphone ด้วย

ภาคผนวก ข

กลุ่ม 3 คน แบบมีเงื่อนไข

ในรายงานขึ้นสอบให้เขียน ภาคผนวก ข อธิบายว่าได้ทำตามเงื่อนไขของแต่ละกลุ่มแล้ว พร้อมรูปประกอบ