

ระบบตรวจสอบการขนส่งห่วงโซ่ความเย็นสำหรับอาหาร  
ด้วยเทคโนโลยีบล็อกเชนและอินเทอร์เน็ตของสรรพสิ่ง

กฤตภาส วิเศษสิงห์

TNI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

บัณฑิตวิทยาลัย สาขาเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ปีการศึกษา 2565

IOT BASED COLD CHAIN LOGISTICS WITH BLOCKCHAIN  
FOR FOOD MONITORING APPLICATION

Krittaphas Wisessing

TNI

A Thesis Submitted in Partial Fulfilment of the Requirements  
for the Degree of Master of Science Program in Information Technology

Graduate School

Thai-Nichi Institute of Technology

Academic Year 2022

หัวข้อวิทยานิพนธ์

ระบบตรวจสอบการขนส่งห่วงโซ่ความเย็นสำหรับอาหาร  
ด้วยเทคโนโลยีบล็อกเชนและอินเทอร์เน็ตของสรรพสิ่ง

โดย

กฤตภาส วิเศษสิงห์

สาขาวิชา

เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษา

ดร.ณปภัช วิชัยดิษฐ์

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย - ญี่ปุ่น อนุมัติให้บัณฑิตวิทยาลัยนี้เป็น  
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร.พิชิต สุขเจริญพงษ์)

วันที่ ..... เดือน ..... พ.ศ. ....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ธงชัย แก้วกิริยา)

.....กรรมการ

(ดร.ภาสกร อภิรักษ์วรพินิต)

.....กรรมการ

(ดร.สรมย์พร เจริญพิทย์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ดร.ณปภัช วิชัยดิษฐ์)

กฤตภาส วิเศษสิงห์ : ระบบตรวจสอบการขนส่งห่วงโซ่ความเย็นสำหรับอาหารด้วยเทคโนโลยี  
บล็อกเชนและอินเทอร์เน็ตของสรรพสิ่ง. อาจารย์ที่ปรึกษา : ดร.ณปภัช วิชัยดิษฐ์, 69 หน้า.

การขนส่งอาหารสดหรืออาหารทะเลนั้นความสามารถในการควบคุมหรือตรวจสอบอุณหภูมิ  
ในระหว่างขนส่งได้ถือเป็นเรื่องที่สำคัญและมองข้ามไม่ได้ หากผู้บริโภคสามารถตรวจสอบและ  
รับประกันด้วยตัวเองได้ว่าสินค้ามีความปลอดภัยนอกจากจะเพิ่มความมั่นใจในสินค้าและบริการของผู้  
จัดส่งแล้วยังส่งผลสำคัญต่อการตัดสินใจในอนาคตของผู้บริโภคว่าจะเลือกใช้บริการต่อไปอีกหรือไม่  
ด้วย งานวิจัยนี้จึงมุ่งเน้นไปที่การใช้งานบล็อกเชนร่วมกับข้อมูลอุณหภูมิและพิกัดของรถขนส่งสินค้าที่ส่ง  
จากอุปกรณ์อินเทอร์เน็ตของสรรพสิ่งเพื่อพัฒนาแอปพลิเคชันขึ้นมา โดยเหตุผลที่เลือกบล็อกเชน  
เนื่องจากความโปร่งใสในการแสดงข้อมูลทั้งหมดให้แก่ผู้ใช้งานและบล็อกเชนเป็นระบบที่ไม่ต้องมี  
เครื่องศูนย์กลางในการเก็บข้อมูลทำให้ยากต่อการถูกโจมตีหรือแก้ไขข้อมูลจากภายนอก แนวคิดหลัก  
ของแอปพลิเคชันคือผู้ใช้งานทุกคนสามารถตรวจสอบบันทึกอุณหภูมิตลอดการจัดส่งและระบบจะมี  
การแจ้งเตือนไปยังพนักงานขนส่งในกรณีที่อุณหภูมิสูงหรือต่ำเกินกว่าที่กำหนดไว้

ในการนำข้อมูลจากเซนเซอร์ที่ติดตั้งบนรถขนส่งสินค้าไปบันทึกเป็นธุรกรรมที่เก็บบนเครือข่าย  
บล็อกเชน จำเป็นต้องเชื่อมการทำงานของแอปพลิเคชัน, อุปกรณ์ IoT และบล็อกเชนเข้าเป็นระบบ  
เดียวกัน โดยในงานวิจัยนี้ได้พัฒนา Web Application ด้วย Vue.js ซึ่งเป็นภาษา JavaScript และ  
พัฒนาระบบบล็อกเชนด้วยภาษาไพทอนที่รันบนเครื่องแม่ข่าย แล้วเชื่อมต่อกันผ่าน API ทำให้  
สามารถรับ-ส่งข้อมูลกันได้ และผลการวิจัยคือเครื่องแม่ข่ายทุกเครื่องที่รันเครือข่ายบล็อกเชนเดียวกัน  
จะถือชุดข้อมูลเดียวกันพร้อมทั้งตรวจสอบความถูกต้องของชุดข้อมูลนั้นด้วย และเมื่อข้อมูลแต่ละ  
บล็อกในบล็อกเชนเก็บการเข้ารหัสของข้อมูลจากบล็อกก่อนหน้าแล้วจึงมั่นใจได้ว่าข้อมูลที่ถูกเก็บนั้น  
เชื่อถือได้และสามารถย้อนรอยข้อมูลได้อย่างแน่นอน อีกทั้งระบบแจ้งเตือนยังสามารถใช้งานได้โดยจะ  
ส่งข้อความแจ้งเตือนไปยังพนักงานขนส่งผ่านทาง Line Official เมื่ออุณหภูมิหรือพิกัดที่วัดค่าได้ไม่  
ตรงตามเงื่อนไขที่กำหนดไว้

บัณฑิตวิทยาลัย  
สาขาวิชา เทคโนโลยีสารสนเทศ  
ปีการศึกษา 2565

ลายมือชื่อนักศึกษา.....  
ลายมือชื่ออาจารย์ที่ปรึกษา.....

KRITTAPHAS WISESSING : IOT BASED COLD CHAIN LOGISTICS WITH BLOCKCHAIN  
FOR FOOD MONITORING SYSTEM. ADVISOR : DR.NAPAPHAT VICHADIS, 69 PP.

In the field of raw food and seafood transportation, the importance of real-time temperature control and monitoring cannot be overstated. The assurance of product safety throughout the transport will enhance the product's trustworthiness which is one of the factors that customers consider while deciding whether to utilize the service. This research focused on the concept of using the cold chain system using Internet of Things devices and blockchain technology. The reason why blockchain has been chosen is because its data's transparency and blockchain is a decentralized system, making it difficult to attack or modify data from anywhere. The device collects temperature and geolocation and sends it to blockchain network. The idea behind the application is to allow consumers to track products through a supply chain and notify the driver when the container's temperature is out of the range.

In order to record the shipment's temperature log as a blockchain transaction, the application, the devices, and the blockchain will be connected together. In this research, the web application using Vue.js which is JavaScript framework integrated with blockchain system written in Python language running on the server. The result is that all servers running the same blockchain network will hold the same data which be validated and each block in the network can be traced to its previous block. In addition, the notification feature is also worked by sending the notification to the driver via the LINE Official Account when the measured temperature or coordinates do not meet the conditions.

Graduate School

Field of Study Information Technology

Academic Year 2022

Student's Signature.....

Advisor's Signature.....

## กิตติกรรมประกาศ

การทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอขอบคุณ ดร.ณปภัช วิชัยดิษฐ์ เป็นอย่างยิ่ง ที่ได้สละเวลาอันมีค่าในการให้ความช่วยเหลือและสนับสนุนการทำวิจัยรวมถึงให้คำปรึกษาและตรวจสอบขั้นตอนการทำวิทยานิพนธ์ด้วยความเอาใจใส่เป็นอย่างยิ่ง รวมถึงคณาจารย์และเจ้าหน้าที่ของสถาบันเทคโนโลยีไทย-ญี่ปุ่นทุกท่าน ที่ช่วยอำนวยความสะดวกในการจัดทำเอกสารรวมถึงลำดับการจัดทำวิทยานิพนธ์ฉบับนี้จนประสบความสำเร็จ

ผู้วิจัยขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ได้แก่ ผู้ช่วยศาสตราจารย์ ดร.ธงชัย แก้วกิริยา ดร.ภาสกร อภิรักษ์รพินิต และ ดร.สรมย์พร เจริญพิทย์ ที่ได้กรุณาให้ข้อเสนอแนะที่เป็นประโยชน์และช่วยตรวจสอบข้อบกพร่องต่อการทำวิทยานิพนธ์ฉบับนี้ จนผู้วิจัยสามารถทำวิทยานิพนธ์ฉบับนี้ได้เสร็จสิ้นดังที่ตั้งใจไว้

ผู้วิจัยขอขอบคุณบิดา มารดา ครอบครัวของผู้วิจัย และบุคคลรอบตัวผู้วิจัย ที่ให้กำลังใจและสนับสนุนตลอดระยะเวลาการศึกษา จนกระทั่งผู้วิจัยสามารถจัดทำวิทยานิพนธ์ฉบับนี้เสร็จสิ้น

ผู้วิจัยหวังเป็นอย่างยิ่งว่า วิทยานิพนธ์ฉบับนี้จะก่อให้เกิดประโยชน์แก่ผู้ที่สนใจ และสามารถนำไปต่อยอดหรือเป็นแนวทางในการพัฒนางานวิจัยที่เกี่ยวข้องต่อไปได้

กฤตภาส วิเศษสิงห์

TNI

THAI - JAPAN INSTITUTE OF TECHNOLOGY

## สารบัญ

|                                                  | หน้า |
|--------------------------------------------------|------|
| บทคัดย่อภาษาไทย.....                             | ง    |
| บทคัดย่อภาษาอังกฤษ.....                          | จ    |
| กิตติกรรมประกาศ.....                             | ฉ    |
| สารบัญ.....                                      | ช    |
| สารบัญตาราง.....                                 | ญ    |
| สารบัญรูป.....                                   | ฎ    |
| <br>บทที่                                        |      |
| 1    บทนำ.....                                   | 1    |
| 1.1 ความเป็นมาและความสำคัญ.....                  | 1    |
| 1.2 วัตถุประสงค์ของการวิจัย.....                 | 2    |
| 1.3 ขอบเขตในการวิจัย.....                        | 2    |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ.....               | 3    |
| 1.5 นิยามศัพท์เฉพาะ.....                         | 3    |
| 2    เอกสารและงานวิจัยที่เกี่ยวข้อง.....         | 5    |
| 2.1 ทฤษฎีที่เกี่ยวข้อง.....                      | 5    |
| 2.1.1 อินเทอร์เน็ตของสรรพสิ่ง.....               | 5    |
| 2.1.2 ระบบขนส่งห่วงโซ่ความเย็น.....              | 8    |
| 2.1.3 บล็อกเชน.....                              | 9    |
| 2.2 ทฤษฎีและเครื่องมือในการพัฒนาแอปพลิเคชัน..... | 11   |
| 2.2.1 วิทยาการเข้ารหัสลับ.....                   | 11   |
| 2.2.2 Docker.....                                | 13   |
| 2.2.3 Flask เฟรมเวิร์ก.....                      | 14   |
| 2.2.4 Vue.js.....                                | 15   |

## สารบัญ (ต่อ)

| บทที่ |                                                        | หน้า |
|-------|--------------------------------------------------------|------|
| 3     | วิธีการดำเนินงานวิจัย.....                             | 17   |
|       | 3.1 ขั้นตอนการวิจัย.....                               | 17   |
|       | 3.2 แผนงานและระยะเวลาในการดำเนินงาน.....               | 17   |
|       | 3.3 ภาพรวมของระบบ.....                                 | 19   |
|       | 3.3.1 ส่วนของผู้ใช้งานและสิทธิ์ต่าง ๆ.....             | 19   |
|       | 3.3.2 ส่วนระบบหลังบ้าน backend.....                    | 20   |
|       | 3.3.3 ส่วนของบุคคลที่สาม (Third Party).....            | 20   |
|       | 3.4 การออกแบบฐานข้อมูลของระบบ.....                     | 21   |
|       | 3.5 เครื่องมือและอุปกรณ์ที่ใช้.....                    | 25   |
|       | 3.6 การพัฒนาบล็อกเชน.....                              | 25   |
|       | 3.7 การเชื่อมโยงส่วนการทำงานของระบบ.....               | 28   |
|       | 3.7.1 รถขนส่งสินค้า.....                               | 28   |
|       | 3.7.2 เครื่องแม่ข่าย.....                              | 30   |
|       | 3.7.3 Web Application.....                             | 31   |
|       | 3.8 การวัดประสิทธิภาพของระบบ.....                      | 34   |
| 4     | ผลการวิจัย.....                                        | 35   |
|       | 4.1 ผลการทำงานของระบบจากการพัฒนาเครือข่ายบล็อกเชน..... | 35   |
|       | 4.2 ผลการพัฒนา Web Application.....                    | 38   |
|       | 4.2.1 หน้าจอหลัก.....                                  | 39   |
|       | 4.2.2 หน้าแสดงรายละเอียดการจัดส่ง.....                 | 40   |
|       | 4.2.3 หน้าสร้างการจัดส่งครั้งใหม่.....                 | 41   |
|       | 4.2.4 หน้าจัดการผู้ใช้.....                            | 42   |
|       | 4.2.5 หน้าจัดการสินค้า.....                            | 43   |
|       | 4.2.6 หน้าจัดการอุปกรณ์.....                           | 44   |
|       | 4.2.7 การแจ้งเตือนที่ได้รับจากระบบ.....                | 45   |
|       | 4.3 ผลการวัดประสิทธิภาพของระบบ.....                    | 46   |

## สารบัญ (ต่อ)

|                                        | หน้า |
|----------------------------------------|------|
| บทที่                                  |      |
| 5    บทสรุปและข้อเสนอแนะ.....          | 48   |
| 5.1 สรุปผลการวิจัย.....                | 48   |
| 5.2 ข้อเสนอแนะงานวิจัย.....            | 49   |
| บรรณานุกรม.....                        | 50   |
| ภาคผนวก.....                           | 55   |
| ภาคผนวก ก. โค้ดภาษาไพทอน (Python)..... | 56   |
| ประวัติย่อผู้วิจัย.....                | 69   |



TNII

THAI - NICHI INSTITUTE OF TECHNOLOGY

## สารบัญตาราง

| ตาราง                                                                          | หน้า |
|--------------------------------------------------------------------------------|------|
| 3.1 แผนงานและระยะเวลาดำเนินงาน.....                                            | 17   |
| 3.2 ตารางข้อมูลผู้ใช้งาน.....                                                  | 21   |
| 3.3 ตารางข้อมูลอุปกรณ์.....                                                    | 22   |
| 3.4 ตารางข้อมูลสินค้า.....                                                     | 23   |
| 3.5 ตารางข้อมูลการจัดส่ง.....                                                  | 23   |
| 3.6 ตารางรายละเอียดการจัดส่ง.....                                              | 24   |
| 3.7 ตารางข้อมูลจากอุปกรณ์.....                                                 | 25   |
| 4.1 เวลาที่ใช้ในการเก็บและส่งข้อมูลไปยังบล็อกเชนเทียบกับบล็อกเชน Ethereum..... | 47   |



TNI

## สารบัญรูป

| รูป  |                                                                                                     | หน้า |
|------|-----------------------------------------------------------------------------------------------------|------|
| 1.1  | มูลค่าผลิตภัณฑ์มวลรวมทั้งหมดเทียบกับมูลค่าด้านอาหาร.....<br>และการเกษตรของประเทศไทย.....            | 1    |
| 2.1  | ระบบขนส่งห่วงโซ่ความเย็นเริ่มตั้งแต่รับของจากแหล่งผลิตจนกระทั่ง.....<br>ถึงผู้บริโภค/ร้านค้า.....   | 8    |
| 2.2  | ตัวอย่างการนำข้อความจากอาเรียมาเข้ารหัสแบบ SHA-256.....                                             | 13   |
| 2.3  | ตัวอย่างการใช้งาน Flask เพื่อแสดงผลแบบเว็บแอปพลิเคชัน.....                                          | 14   |
| 2.4  | เปรียบเทียบการทำงานของเว็บไซต์ทั่วไป (ซ้าย) กับเว็บไซต์แบบ SPA (ขวา).....                           | 16   |
| 3.1  | ภาพรวมการทำงานทั้งหมดของระบบ.....                                                                   | 19   |
| 3.2  | ฐานข้อมูลของระบบ ประกอบด้วย ข้อมูลการขนส่ง ผู้ใช้ อุปกรณ์.....<br>สินค้า และข้อมูลจากเซนเซอร์.....  | 21   |
| 3.3  | การทำงานภายในบล็อกเชนหลังจากอุปกรณ์ส่งข้อมูลเข้ามายังเครื่องแม่ข่าย.....                            | 27   |
| 3.4  | แผนภาพแสดงการเชื่อมโยงส่วนการทำงานทั้ง 3 ส่วน.....                                                  | 28   |
| 3.5  | ตัวอย่างขั้นตอนการทำงานของบล็อกเชนเมื่อติดตั้งและรันผ่านเครื่องแม่ข่าย.....                         | 31   |
| 3.6  | ค่า User LINE Id ที่ได้หลังจากผู้ใช้งานล็อกอินเข้าสู่ระบบ.....                                      | 32   |
| 3.7  | กระบวนการตรวจสอบเงื่อนไขและส่งข้อความแจ้งเตือน.....                                                 | 33   |
| 4.1  | ตัวอย่างข้อมูลในบล็อกเชนที่เรียกออกมาผ่าน API.....                                                  | 35   |
| 4.2  | บล็อกเชนที่รันจากเครื่องแม่ข่ายคนละตัว.....                                                         | 36   |
| 4.3  | การเพิ่มโหนด (ซ้าย) และการใช้คำสั่งซิงค์เพื่อดึงข้อมูลจากโหนด.....<br>ที่มีข้อมูลมากกว่า (ขวา)..... | 37   |
| 4.4  | แบบฟอร์มสมัครสมาชิกเมื่อผู้ใช้งานเข้าสู่ระบบครั้งแรก.....                                           | 38   |
| 4.5  | หน้าจอหลักของ Web Application แยกตามประเภทผู้ใช้งาน.....                                            | 39   |
| 4.6  | หน้าจอแสดงรายละเอียดการจัดส่ง แยกตามประเภทผู้ใช้งาน.....                                            | 40   |
| 4.7  | หน้าจอเมื่อกดปุ่มสร้างการจัดส่งครั้งใหม่โดยพนักงานขนส่ง.....                                        | 41   |
| 4.8  | หน้าจัดการผู้ใช้แสดงรายชื่อผู้ใช้ทั้งหมดและการแก้ไขสิทธิ์ของผู้ใช้.....                             | 42   |
| 4.9  | หน้าจัดการสินค้าแสดงรายการสินค้าทั้งหมดและการเพิ่มสินค้าเข้าสู่ระบบ.....                            | 43   |
| 4.10 | หน้าจัดการอุปกรณ์แสดงรายการอุปกรณ์ IoT ทั้งหมดและการเพิ่ม.....<br>อุปกรณ์เข้าสู่ระบบ.....           | 44   |

### สารบัญรูป (ต่อ)

| รูป  |                                                                                           | หน้า |
|------|-------------------------------------------------------------------------------------------|------|
| 4.11 | การแจ้งเตือนที่ได้รับจากระบบเมื่อตรวจสอบเงื่อนไขอุณหภูมิหรือพิกัด.....<br>ไม่ถูกต้อง..... | 45   |
| 4.12 | เวลาที่ใช้ในการเก็บและส่งข้อมูล (ซ้าย) และหน่วยความจำที่ใช้ (ขวา).....                    | 46   |

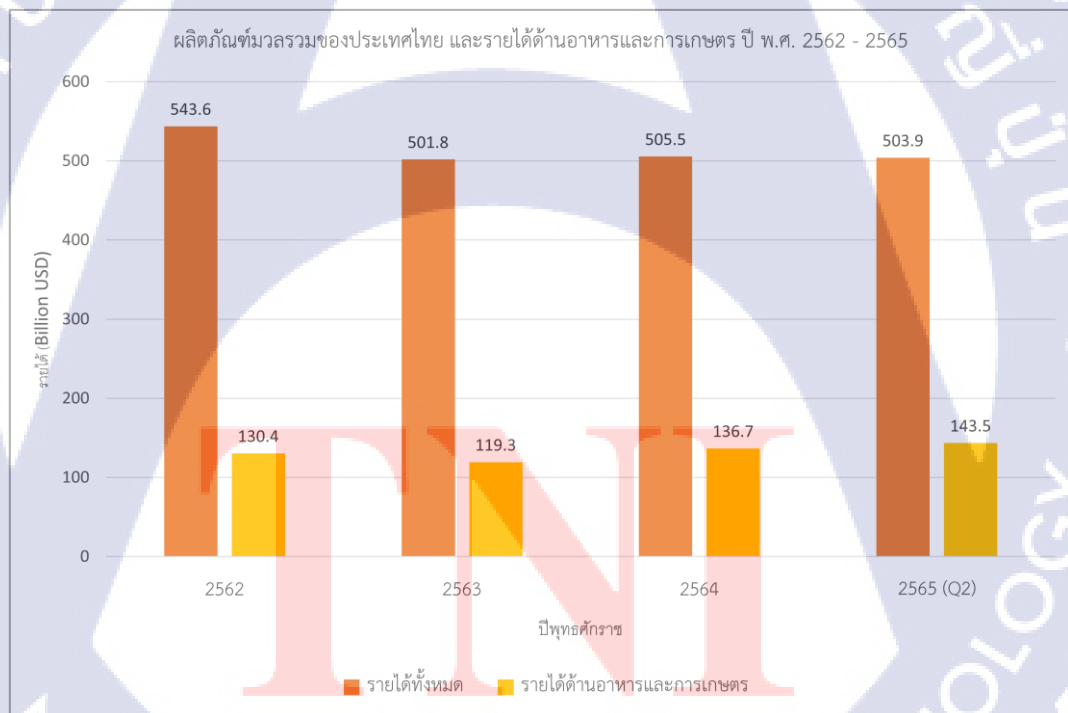


## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ประเทศไทยได้ชื่อว่าเป็นหนึ่งในแหล่งผลิตอาหารที่สำคัญในภูมิภาคเอเชียตะวันออกเฉียงใต้ และยังมีการพัฒนาการผลิตสินค้าที่เป็นอาหารออกมาในรูปแบบต่าง ๆ อย่างต่อเนื่อง แต่กลับกัน ประเทศไทยมีการลงทุนด้านการลดความสูญเสียของผลิตภัณฑ์ในระหว่างการขนส่งถือว่าอยู่ในระดับต่ำ [1] ยิ่งไปกว่านั้นในปัจจุบันยังเกิดปัญหาด้านสุขภาพซึ่งเป็นผลมาจากการเก็บรักษาอาหารระหว่างการขนส่งที่มีคุณภาพที่ไม่ดี อีกทั้งการขนส่งอาหารที่ไวต่อการเปลี่ยนแปลงของอุณหภูมิแต่ไม่มีการดูแลระหว่างการจัดส่งที่ดียังนำไปสู่ต้นเหตุของการปนเปื้อนเชื้อโรคหรือเกิดการเน่าเสียจนไม่สามารถบริโภคได้ [2], [3] ซึ่งอาจก่อให้เกิดปัญหาเพิ่มเติม คือ อาการเจ็บป่วยที่เกิดจากเชื้อโรค หรือ ความสูญเสียจากการผลิตอาหารและสินค้า



รูปที่ 1.1 มูลค่าผลิตภัณฑ์มวลรวมทั้งหมดเทียบกับมูลค่าด้านอาหารและการเกษตรของประเทศไทย

จากแหล่งอ้างอิงเรื่องผลิตภัณฑ์มวลรวม (GDP) ของประเทศไทย ในปี พ.ศ. 2563 รายได้ด้านอาหารและการเกษตรของประเทศไทยนับว่าสำคัญเป็นลำดับที่ 3 (23% จากรายได้ทั้งหมด) [4]

และจากรูปที่ 1.1 พบว่าในช่วงปี 2562 – 2563 มีมูลค่าด้านอาหารและการเกษตรลดลงจาก \$130.4 USD (4,891 ล้านบาท) เหลือ \$119.3 USD (4,475 ล้านบาท) หรือนับว่าลดลงไปร้อยละ 8.5 แต่กลับกันหากเทียบในช่วงปี 2563 – 2565 จะมีมูลค่าด้านอาหารและการเกษตรเพิ่มเป็น \$143.5 USD (5,383 ล้านบาท) หรือเพิ่มขึ้นร้อยละ 20 และยังมีแนวโน้มที่จะเพิ่มมากขึ้นอีกในปีต่อ ๆ ไป [5] นอกจากนี้ยังมีรายงานเรื่องปริมาณเศษอาหารเหลือทิ้งในประเทศไทยที่มากถึง 17 ล้านตันต่อปี [6] โดยหากแปลงเป็นเงินไทยจะมีมูลค่าถึง 1,900 ล้านบาท [7] ซึ่งบ่งชี้ถึงรายได้ที่ต้องสูญเสียไปโดยเปล่าจากการผลิตอาหารและขนส่งอาหารเหล่านั้น

จากความสำคัญของรายได้ด้านอาหารและการเกษตรรวมถึงปัญหาซึ่งสามารถก่อให้เกิดรายจ่ายสูญเสียที่ได้กล่าวมาข้างต้นจึงมีแนวคิดที่จะศึกษาและพัฒนาระบบห่วงโซ่ความเย็น (Cold Chain Logistics) ที่เป็นระบบติดตามอุณหภูมิในระหว่างการขนส่งอาหารด้วยเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง (Inter of Things: IoT) ร่วมกับบล็อกเชน โดยอุปกรณ์ IoT จะถูกติดตั้งเพื่อเก็บค่าจากสภาพแวดล้อมรอบตัวสินค้า เช่น การวัดค่าอุณหภูมิด้วยเซนเซอร์หรือการใช้โมดูลตรวจสอบพิกัดปัจจุบันของสินค้า และนำข้อมูลจากอุปกรณ์ส่งไปเก็บค่าในเครือข่ายบล็อกเชนเพื่อเพิ่มความมั่นใจในสินค้าและเพิ่มความน่าเชื่อถือแก่ผู้บริโภค ข้อมูลที่ได้จากอุปกรณ์จะแสดงผ่านทางแอปพลิเคชันที่สามารถตรวจสอบได้ตลอดเวลา และมีระบบแจ้งเตือนไปยังพนักงานขับรถในกรณีที่อุณหภูมิสูงหรือต่ำเกินกว่าที่กำหนดไว้

## 1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพัฒนาระบบจัดเก็บข้อมูลแบบบล็อกเชนร่วมกับการรับข้อมูลด้วยเทคโนโลยี IoT

1.2.2 เพื่อพัฒนา Web Application ที่สามารถดูบันทึกข้อมูลอุณหภูมิในระหว่างการขนส่งสินค้าได้ และสามารถแจ้งเตือนไปยังผู้รับผิดชอบการขนส่งในขณะนั้น เมื่ออุณหภูมิรอบสินค้าสูงหรือต่ำกว่าค่าที่เหมาะสม

1.2.3 เพื่อศึกษาประสิทธิภาพในการส่งข้อมูลจากอุปกรณ์ไปยังเครือข่ายบล็อกเชน

## 1.3 ขอบเขตในการวิจัย

1.3.1 Web Application ที่พัฒนาจะเป็น Responsive Website คือการแสดงผลจะถูกปรับให้เหมาะสมขึ้นอยู่กับอุปกรณ์ที่เปิดใช้งานแอปพลิเคชัน

1.3.2 ทดสอบระบบแจ้งเตือนด้วยการเขียนโปรแกรมเพื่อส่งข้อมูลอุณหภูมิและพิกัดของสินค้า โดยตั้งค่าให้ใกล้เคียงกับข้อมูลที่ได้จากอุปกรณ์จริงมากที่สุด

1.3.3 ผู้ใช้งานที่สามารถใช้ระบบได้จะต้องมีบัญชี LINE เนื่องจากระบบต้องดึงข้อมูลผู้ใช้ผ่าน LINE

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถพัฒนาระบบจัดเก็บข้อมูลแบบบล็อกเชนร่วมกับการรับข้อมูลด้วยเทคโนโลยี IoT

1.4.2 สามารถพัฒนา Web Application ที่สามารถบันทึกข้อมูลอุณหภูมิในระหว่างการขนส่งสินค้าได้ และสามารถแจ้งเตือนไปยังผู้รับผิดชอบการขนส่งในขณะนั้น เมื่ออุณหภูมิรอบสินค้าสูงหรือต่ำกว่าค่าที่เหมาะสม

1.4.3 สามารถวัดประสิทธิภาพในการส่งข้อมูลจากอุปกรณ์ไปเก็บยังเครือข่ายบล็อกเชน

## 1.5 นิยามศัพท์เฉพาะ

1.5.1 **Web Application** หมายถึง โปรแกรมชนิดหนึ่งที่อยู่ในรูปแบบของเว็บไซต์ โดยถูกจัดเก็บไว้ในเครื่องแม่ข่าย ใช้งานผ่านทางโปรแกรมอื่น ๆ หรือ เว็บเบราว์เซอร์ โดยมีวัตถุประสงค์และความสามารถมากกว่าการอ่านเนื้อหาเพียงอย่างเดียว

1.5.2 **LINE** หมายถึง โปรแกรมแชทชนิดหนึ่ง ที่สามารถใช้งานได้ทั้งโทรศัพท์มือถือและบนคอมพิวเตอร์ รองรับข้อมูลทั้งรูปแบบข้อความ ภาพ หรือวิดีโอ เป็นต้น

1.5.3 **LINE Official Account (LINE OA)** หมายถึง บัญชี LINE ประเภทหนึ่ง ถูกออกแบบมาใช้ในเชิงธุรกิจสำหรับองค์กร หน่วยงาน หรือบุคคลที่ต้องการสื่อสารกับผู้ติดตามจำนวนมาก มีฟีเจอร์พิเศษที่ใช้งานได้ดีกว่าการทำงานผ่านบัญชี LINE ประเภทบุคคลธรรมดา และยังสามารถตั้งค่าการตอบข้อความอัตโนมัติ หรือเชื่อมต่อกับ Web Application ได้อีกด้วย

1.5.4 **Application Program Interface (API)** หมายถึง ช่องทางการเชื่อมต่อสื่อสารรูปแบบหนึ่ง ที่เป็นตัวกลางทำหน้าที่ช่วยในการเข้าถึงข้อมูลต่าง ๆ หรือเป็นการนำข้อมูลต่าง ๆ ออกจากเว็บไซต์ หรือเป็นตัวกลางในการใช้งานคำสั่งของโปรแกรมอื่น ๆ ผ่านเว็บไซต์ได้เช่นกัน

1.5.5 **Structured Query Language (SQL)** หมายถึง ภาษาที่ใช้ในการเขียนโปรแกรมชนิดหนึ่ง เป็นคำสั่งที่ใช้ในการบริหารจัดการฐานข้อมูล ที่ออกแบบมาเพื่อทำการจัดการข้อมูล ค้นหาข้อมูล ปรับปรุง เปลี่ยนแปลง เพิ่ม และ ลบข้อมูล โดยข้อมูลถูกจัดเก็บในรูปแบบตารางที่มีลักษณะเป็นคอลัมน์และแถว

1.5.6 **JavaScript Object Notation (JSON)** หมายถึง รูปแบบสำหรับแลกเปลี่ยนข้อมูล อยู่ในรูปข้อความธรรมดาที่ทั้งมนุษย์และโปรแกรมคอมพิวเตอร์สามารถอ่านเข้าใจได้ สามารถเขียนโปรแกรมเพื่อเข้าถึงข้อมูลภายใน JSON ได้ง่าย มักจะอยู่ในรูปแบบข้อมูลที่เริ่มจากเครื่องหมายปีกกาเปิด { และสิ้นสุดด้วยเครื่องหมายปีกกาปิด } โดยข้อมูลภายในจะเป็นการจับคู่กันระหว่าง ชื่อข้อมูล/ค่าของข้อมูล

1.5.7 **Backend** หมายถึง ส่วนของการทำงานเบื้องหลังที่ผู้ใช้งานไม่สามารถโต้ตอบ<sup>15</sup> โดยตรง เช่น การจัดการกับฐานข้อมูล หรือ โครงสร้างพื้นฐานของระบบ

1.5.8 **Third-Party (3<sup>rd</sup>-Party)** หมายถึง ซอฟต์แวร์ บุคคล หรือธุรกิจที่ไม่ได้เกี่ยวข้องในการพัฒนาโดยตรง หรือถูกพัฒนาโดยบุคคลอื่น

1.5.9 **โหนด (Node)** หมายถึง คอมพิวเตอร์หรืออุปกรณ์ที่เชื่อมต่อกับเครือข่ายบัสล็อกเซน



## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และได้นำเสนอตามหัวข้อต่อไปนี้

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 อินเทอร์เน็ตของสรรพสิ่ง

##### 2.1.2 การขนส่งห่วงโซ่ความเย็น

##### 2.1.3 บล็อกเชน

#### 2.2 ทฤษฎีและเครื่องมือในการพัฒนาแอปพลิเคชัน

##### 2.2.1 การเข้ารหัสข้อมูล

##### 2.2.2 Docker

##### 2.2.3 Flask เฟรมเวิร์ก

##### 2.2.4 Vue.js

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 อินเทอร์เน็ตของสรรพสิ่ง

อินเทอร์เน็ตของสรรพสิ่ง หรือ อินเทอร์เน็ตออฟติงส์นับเป็นหนึ่งในโครงสร้างทางสถาปัตยกรรมเครือข่ายที่อุปกรณ์ในเครือข่ายเดียวกันสามารถติดต่อสื่อสารกันได้ โดยที่ไม่จำเป็นต้องถูกป้อนข้อมูลด้วยมือของมนุษย์ ทำให้เราสามารถควบคุมการใช้งานของอุปกรณ์อิเล็กทรอนิกส์ที่อยู่ในเครือข่ายได้โดยตรง จากความสามารถในการเชื่อมต่ออุปกรณ์ที่หลากหลายทำให้มีการประยุกต์ใช้งานได้อย่างหลากหลายและกว้างขวาง เช่น การเชื่อมต่ออุปกรณ์เซนเซอร์ต่าง ๆ เข้าด้วยกัน ทำให้ได้รับข้อมูลที่หลากหลายได้ในเวลาเดียวกัน และสามารถนำข้อมูลเหล่านั้นมาวิเคราะห์/ประมวลผลเพื่อแสดงภาพข้อมูลหรือนำไปใช้ประกอบการตัดสินใจอื่น ๆ ได้

อินเทอร์เน็ตของสรรพสิ่งถูกนิยามขึ้นครั้งแรกในปี ค.ศ. 1999 (พ.ศ. 2542) โดย Kevin Ashton ซึ่งขณะนั้นเป็นนักวิจัยที่กำลังศึกษาเรื่อง การเชื่อมโยงอุปกรณ์ต่าง ๆ ผ่านทางการระบุเอกลักษณ์ด้วยคลื่นวิทยุ (Radio-frequency identification: RFID) โดยเขาได้กล่าวถึงอุปกรณ์อิเล็กทรอนิกส์ที่สื่อสารแบบในคลื่นเดียวกันนั้นว่า “Things” และการสื่อสารในเครือข่ายเดียวกันถูกเรียกรวม ๆ ว่าอินเทอร์เน็ต ทำให้คำว่า Internet of Things ได้ถูกใช้เป็นครั้งแรก

ประเภทของเครือข่ายที่ถูกระบุว่าเป็นอินเทอร์เน็ตของสรรพสิ่งแบบไร้สายและพบเห็นได้บ่อยครั้ง [8] แยกประเภทได้ดังนี้

#### 1. LPWAN

การสื่อสารไร้สายแบบวงกว้างที่ใช้พลังงานต่ำ (Low-Power Wide Area Network: LPWAN) ถูกออกแบบมาเพื่อใช้สื่อสารระหว่างอุปกรณ์อิเล็กทรอนิกส์ด้วยกัน มีจุดแข็งคือใช้พลังงานต่ำและสื่อสารได้ระยะไกล โดยเมื่อเทียบกับแล้วมีค่าใช้จ่ายที่น้อยกว่าการใช้งานติดต่อสื่อสารผ่านเครือข่ายโทรศัพท์มือถือ ยกตัวอย่างเช่น LoRa, NB-IoT เครือข่ายในรูปแบบ LPWAN เหมาะแก่การให้กับเครือข่ายอุปกรณ์ที่มีจำนวนมากมีการเชื่อมต่อแบบตัวต่อตัวและต้องการระยะในการติดต่อสื่อสารที่เป็นวงกว้าง

#### 2. Cellular

เทคโนโลยีการติดต่อสื่อสารที่ใช้กับเครือข่ายของโทรศัพท์มือถือเพื่อใช้เชื่อมต่ออินเทอร์เน็ตรองรับการโทรด้วยเสียงหรือแพ็คเกจด้วยวิดีโอ แต่มีข้อเสียคือใช้พลังงานสูง

ถึงแม้ว่าเครือข่ายประเภท Cellular จะไม่นิยมใช้งานกับอุปกรณ์หรือแอปพลิเคชัน IoT โดยทั่วไป แต่ก็มีอุปกรณ์ที่มีแบนด์วิธสูงที่เหมาะสมกับเครือข่ายรูปแบบนี้ เช่น การเชื่อมต่อกับยานพาหนะเพื่อแสดงภาพเส้นทางการจราจร รวมถึงการรับส่งพิกัดกับดาวเทียมด้วย

#### 3. Zigbee

เป็นโปรโตคอลมาตรฐาน IEEE 802.15.4 สำหรับการเชื่อมต่อไร้สายที่ออกแบบมาเพื่ออุปกรณ์ที่ใช้พลังงานต่ำแบนด์วิธต่ำและระยะทำการไม่ไกลมากโดยต้องมีศูนย์กลางในการเชื่อมต่อที่เรียกว่าเกตเวย์ เครือข่ายแบบ Zigbee มักถูกใช้กับการสื่อสารในงานควบคุมและระบบอัตโนมัติ เช่น การควบคุมอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ภายในบ้าน

#### 4. Bluetooth และ BLE

บลูทูธ เป็นเทคโนโลยีในการสื่อสาร ที่สามารถใช้ในการรับส่งข้อมูลระยะใกล้ ระยะ 10 – 200 เมตร ใช้ความถี่คลื่นวิทยุในช่วง 2.4 GHz โดยหลักแล้ว บลูทูธถูกออกแบบมาเพื่อใช้กับอุปกรณ์ที่มีขนาดเล็ก เนื่องจากการขนส่งข้อมูลในจำนวนที่ไม่มาก เช่น ข้อมูลภาพ เสียง นอกจากนี้ยังใช้พลังงานต่ำ และสามารถใช้งานได้ยาวนานโดยไม่ต้องเติมพลังงานบ่อย

บลูทูธพลังงานต่ำ (Bluetooth Low Energy: BLE) เป็นเทคโนโลยีบลูทูธที่ถูกพัฒนาขึ้นมา โดยมีจุดประสงค์คือมุ่งเน้นไปที่ลดการใช้พลังงานและต้นทุนที่ต่ำลงกว่าเดิมมากโดยที่ยังรักษาระยะการสื่อสารและแบนด์วิธที่ใช้ให้ใกล้เคียงกับเทคโนโลยีรูปแบบเดิม

## 5. Wi-Fi

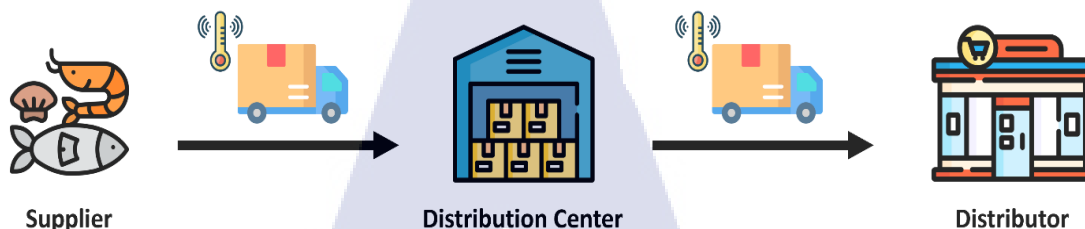
ไวไฟ (Wireless Fidelity: WIFI) เป็นโพรโตคอลมาตรฐาน IEEE 802.1.1 สำหรับการสื่อสารระหว่างอุปกรณ์ต่างๆ เป็นคลื่นวิทยุชนิดหนึ่ง เช่นเดียวกับบลูทูธและเครือข่ายเซลลูลาร์ ที่ใช้สำหรับการสื่อสารอุปกรณ์ในขนาดเล็ก เช่น บ้าน ห้างสรรพสินค้า โดยปกติ เราเตอร์ Wi-Fi จะเชื่อมต่อกับเต้ารับอินเทอร์เน็ตหรือโมเด็ม DSL/เคเบิล/ดาวเทียม ผ่านสายเคเบิลเครือข่ายสำหรับการเข้าถึงอินเทอร์เน็ต จากนั้นจะ broadcast ชื่อ Wi-Fi (SSID) ไปยังอุปกรณ์โดยรอบ เมื่ออุปกรณ์ต้องการเข้าร่วมเครือข่ายไร้สาย อุปกรณ์จะส่งสัญญาณเพื่อแจ้งให้เราเตอร์ทราบ เมื่อเราเตอร์ได้รับและยอมรับคำขอ การเชื่อมต่อจะถูกสร้างขึ้น

## 6. RFID

มีวัตถุประสงค์หลักเพื่อนำไปใช้งานแทนระบบบาร์โค้ด มี 3 องค์ประกอบหลัก คือ 1) แท็ก ภายในประกอบไปด้วย เสาอากาศ และตัวไมโครชิป เสาอากาศจะทำหน้าที่รับส่งสัญญาณคลื่นวิทยุ ระหว่าง แท็ก กับ เครื่องอ่าน 2) เครื่องอ่านป้าย ประกอบด้วย เสาอากาศ เพื่อใช้รับ - ส่งสัญญาณ สัญญาณวิทยุ ภาครับ-ภาคส่ง วงจรควบคุม การอ่าน-เขียนข้อมูล และส่วนที่ติดต่อกับคอมพิวเตอร์ 3) ระบบที่ใช้ประมวลผล ใช้สร้างข้อมูลเพื่อส่งไปยังป้ายและยังเป็นที่เก็บระบบฐานข้อมูลได้อีกด้วย จุดเด่นของ RFID อยู่ยู่ที่การอ่านข้อมูลแบบไร้สัมผัสโดยข้อมูลจะถูกเก็บไว้ในไมโครชิปที่อยู่ในแท็ก

จากความหลากหลายของอินเทอร์เน็ตของสรรพสิ่งนี้เอง ทำให้สามารถนำไปประยุกต์ใช้ได้หลายด้านเช่นกัน ยกตัวอย่างเช่น การนำ IoT ไปใช้กับอุตสาหกรรมโลจิสติกส์เพื่อการขนส่ง ทำให้การขนส่งใช้เวลาสั้นลงกว่าเดิมหลังจากประยุกต์ใช้กับอินเทอร์เน็ตของสรรพสิ่ง หรือนำไปใช้กับโรงงานผลิตทำให้สามารถผลิตสินค้าได้อัตโนมัติ เพิ่มประสิทธิภาพในการทำงานที่มากขึ้นและดึงข้อมูลการผลิตมาตรวจสอบได้พร้อมกันแบบเรียลไทม์ [9] ในด้านของอาหารและการเกษตรก็มีการประยุกต์ใช้อินเทอร์เน็ตของสรรพสิ่งด้วยเช่นกัน อย่างการรับส่งข้อมูลในการจัดส่งสินค้าผ่านเครือข่าย Wi-Fi เพื่อให้ผู้เกี่ยวข้องสามารถดูข้อมูลที่ต้องการได้ตลอดเวลาโดยไม่ต้องทำการตรวจสอบด้วยตนเอง [10]

### 2.1.2 การขนส่งห่วงโซ่ความเย็น



รูปที่ 2.1 ระบบขนส่งห่วงโซ่ความเย็นเริ่มตั้งแต่รับของจากแหล่งผลิตจนกระทั่งถึงผู้บริโภคร้านค้า

ระบบขนส่งห่วงโซ่ความเย็น หรือ Cold Chain Logistics คือกระบวนการหนึ่งในห่วงโซ่อุปทาน (Supply Chain) สำหรับสินค้าที่จำเป็นต้องมีการควบคุมตรวจสอบอุณหภูมิ ตั้งแต่ขั้นการเก็บเกี่ยวหรือผลิตไปจนถึงขั้นตอนการจัดส่งถึงสถานที่ปลายทาง มักนำไปใช้ร่วมกับสินค้าที่สามารถเน่าเสียหรือทำให้ไม่สามารถบริโภคต่อได้ เช่น วัคซีน ยา อาหารทะเล อาหารสด หรือผลิตภัณฑ์จากนม ซึ่งจำเป็นต้องถูกเก็บในอุณหภูมิที่เหมาะสมเพื่อต่ออายุการบริโภคของสินค้าเหล่านั้นให้มากที่สุด [11] ตัวอย่างเส้นทางการขนส่งห่วงโซ่ความเย็นดังแสดงในรูปที่ 1 จากภาพจะเริ่มเก็บข้อมูลการจัดส่งตั้งแต่เริ่มต้นผลิตสินค้าไปยังศูนย์กระจายสินค้าและสิ้นสุดที่มือของร้านค้าปลีก/ผู้บริโภค โดยมีการติดตั้งเซนเซอร์วัดอุณหภูมิเข้าไปในยานพาหนะขนส่งสินค้าและเซนเซอร์จะส่งข้อมูลผ่านเครือข่ายตลอดเวลาเพื่อให้คนขับสามารถตรวจสอบอุณหภูมิได้ [12] และจากการที่ข้อมูลส่งมาอย่างอัตโนมัติช่วยลดโอกาสที่จะเกิดความผิดพลาดจากทางฝั่งที่ทำการจัดส่งด้วย [13] นอกจากนี้การติดตามเส้นทางการขนส่งยังช่วยลดค่าใช้จ่ายบางส่วนได้ เช่น การผลาญเชื้อเพลิงในการจัดส่งเส้นทางระยะยาว

มีตัวอย่างการนำแนวคิดการขนส่งห่วงโซ่ความเย็นมาใช้ร่วมกับเครือข่าย NB-IoT โดยสามารถรับข้อมูลจากอุปกรณ์และวิเคราะห์แล้วประมวลผลข้อมูลได้ในเวลาเดียวกัน [14] ใต้ผลเป็นอย่างมากกับการจัดส่งสินค้าที่ไวต่ออุณหภูมิ ที่การเปลี่ยนแปลงอุณหภูมิเพียงเล็กน้อยก็อาจส่งผลกระทบต่อคุณภาพของสินค้าได้ การใช้เซนเซอร์วัดอุณหภูมิเพื่อวัดค่าอุณหภูมิร่วมกับโมดูลตรวจสอบพิกัดของรถขนส่งสินค้า โดยมีอุปกรณ์ Raspberry PI เป็นตัวส่งต่อข้อมูลและช่วยตรวจสอบเงื่อนไขของอุณหภูมิก็ช่วยให้ขนส่งสินค้าได้สะดวกมากขึ้น

### 2.1.3 บล็อกเชน

จากหัวข้อก่อนหน้านี้จะเห็นว่าการรับข้อมูลจากอุปกรณ์โดยอัตโนมัติช่วยให้สะดวกต่อการทำงานในด้านต่าง ๆ ได้มาก แต่กลับกันก็มีความเสี่ยงที่ข้อมูลที่ได้รับมานั้นอาจไม่ถูกต้องหรือเกิดความเสียหายทำให้ไม่สามารถเชื่อถือข้อมูลนั้นได้ การตรวจสอบก็ยากขึ้นเช่นกันเพราะเป็นข้อมูลที่ถูกส่งจากอุปกรณ์อย่างต่อเนื่อง

บล็อกเชน คือเครือข่ายการเก็บข้อมูลรูปแบบหนึ่งที่เป็นการจัดเก็บข้อมูลแบบกระจายศูนย์ มีความโปร่งใสต่อการตรวจสอบข้อมูล อีกทั้งยังปลอดภัยเพราะข้อมูลไม่สามารถถูกแก้ไขได้ [15] ผู้ใช้ทุกคนที่อยู่บนเครือข่ายเดียวกันหรือที่เรียกว่าโหนด จะจัดเก็บข้อมูลชุดเดียวกัน ทำให้มั่นใจได้ว่าผู้ใช้ทุกคนในระบบสามารถอัปเดตและตรวจสอบบล็อกที่มีข้อมูลที่ไม่สามารถแก้ไขได้เมื่อถูกบันทึกไว้ในบล็อกเชนแล้ว [16] ก่อนที่ข้อมูลจะถูกเพิ่มเข้าไปในบล็อก จะต้องมีการตรวจสอบข้อมูลด้วยการใช้ Consensus Algorithm ซึ่งเป็นระบบที่ทำให้ทั้งเครือข่ายมีความเห็นที่ตรงกันในข้อมูลหรือธุรกรรมนั้น ๆ เพื่อให้บล็อกเชนยังคงรักษาความสามารถในการจัดเก็บข้อมูลแบบกระจายศูนย์ [17] ซึ่ง Consensus Algorithm ก็จะมีหลากหลายชนิดแตกต่างกันขึ้นอยู่กับโครงสร้างของบล็อกเชนที่เลือกใช้ เมื่อตรวจสอบเรียบร้อยแล้วบล็อกนั้นจะถูกนำไปเรียงเรียงต่อในสายโซ่หรือเชน และนอกจากแต่ละบล็อกจะเก็บข้อมูลของตัวเองแล้วยังมีการเก็บข้อมูลที่ถูกเข้ารหัสของข้อมูลบล็อกก่อนหน้านี้ด้วยเช่นกัน (ยกเว้นบล็อกชิ้นแรก) การทำเช่นนี้ส่งผลให้ยิ่งบล็อกเชนมีความยาวมากขึ้นเท่าไรก็ยิ่งถูกดัดแปลงแก้ไขข้อมูลได้ยากขึ้นและส่งผลให้ความน่าเชื่อถือมีมากขึ้นด้วยเช่นกัน

เครือข่ายบล็อกเชนแบ่งออกได้เป็น 2 ประเภท คือ บล็อกเชนส่วนตัว (Private หรือ Permissioned Blockchain) ซึ่งการเข้าใช้งานเช่นการส่งข้อมูลหรือเข้ามาตรวจสอบข้อมูลจากบล็อกเชนประเภทนี้โดยตรง จำเป็นต้องได้รับอนุญาตจากโหนดอื่น ๆ ให้เข้าร่วมเป็นส่วนหนึ่งของเครือข่ายก่อนเท่านั้น และอีกประเภทคือบล็อกเชนสาธารณะ (Public หรือ Permissionless Blockchain) ที่ไม่ว่าใครก็สามารถเข้าร่วมเป็นโหนดของบล็อกเชนประเภทนี้ได้เลยซึ่งส่วนมากจะต้องเสียค่าธรรมเนียมในการใช้งานขึ้นอยู่กับโครงสร้างของบล็อกเชนนั่น [18]

ในช่วงหลังมานี้บล็อกเชนได้รับความนิยมมากขึ้นเรื่อย ๆ จากความสามารถในการเก็บรักษาข้อมูลที่มีความปลอดภัยสูงนี้เอง เราจึงเห็นการประยุกต์ใช้บล็อกเชนร่วมกับระบบหลากหลายประเภทรวมถึงการนำบล็อกเชนมาใช้ร่วมกับการติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตของสรรพสิ่ง การขนส่งแบบห่วงโซ่อุปทานโลจิสติกส์ [19] การเก็บข้อมูลและสกุลเงินในรูปแบบดิจิทัล [20] และเครือข่ายระบบยานยนต์อัตโนมัติ โดยเป็นที่รู้กันว่าอินเทอร์เน็ตของสรรพสิ่งกับการขนส่งแบบห่วงโซ่อุปทานมักถูกนำมาใช้ร่วมกับการเก็บข้อมูลแบบบล็อกเชน นอกจากนี้ยังสามารถนำแนวคิดเหล่านี้มาต่อยอดเพื่อพัฒนาเป็นแอปพลิเคชันที่ทำให้ผู้บริโภคหรือผู้ใช้ทั่วไปสามารถเข้ามาติดตามการจัดส่ง

ตั้งแต่จากจุดเริ่มต้นจนถึงปลายทางของสินค้าได้ในระบบติดตามสินค้าด้านการเกษตรและการขนส่งห่วงโซ่สำหรับอาหาร [21], [22]

ในด้านธุรกิจเกี่ยวกับอาหาร มีงานวิจัย [23] ที่นำระบบการเก็บข้อมูลแบบกระจายศูนย์ร่วมกับการรับส่งข้อมูลของอุปกรณ์ IoT มาเก็บบนเครือข่ายข้อมูลที่เรียกว่า BigchainDB โดยในการจัดส่งสินค้าแต่ละจุด เมื่อผู้ใช้งานที่ลงทะเบียนเป็นโหนดจะยืนยันว่าได้รับสินค้าจากผู้ใช้งานรายอื่น พวกเขาจะต้องใช้กุญแจส่วนตัว (Private Key) ซึ่งใช้ในการเข้ารหัสและยืนยันตัวตนได้ เพื่อใช้ในการลงบันทึกข้อมูลว่าสินค้าได้รับการจัดส่งผ่านจุดนั้นแล้วจริง ๆ เป็นการรักษาความปลอดภัยของข้อมูลและเพิ่มความน่าเชื่อถือ แต่ยังมีกระบวนการบางจุดที่ต้องใช้เวลามากขึ้นเทียบกับการเก็บข้อมูลแบบไม่ต้องใช้บล็อกเชน อีกทั้งหากให้ผู้ใช้งานเป็นคนจัดการกุญแจส่วนตัวด้วยตัวเองมีโอกาที่จะทำกุญแจสูญหายและส่งผลให้ไม่สามารถยืนยันตัวตนได้อีกต่อไปด้วย

จากที่กล่าวมาข้างต้น บล็อกเชนมีความสามารถในการเพิ่มความปลอดภัยและความโปร่งใสจากโครงสร้างแบบกระจายศูนย์ในการเก็บข้อมูลก็จริง แต่ในด้านระยะเวลาที่ใช้ในการรับส่งข้อมูลรวมถึงตรวจสอบความถูกต้องก็มีมากขึ้นเช่นกัน ยกตัวอย่างเช่น ข้อมูลที่ถูกส่งไปเก็บในเครือข่ายบล็อกเชนส่วนตัวระบบหนึ่งสามารถประมวลผลได้ 15 ธุรกรรมต่อวินาทีเท่านั้น [24] ซึ่งนี่ยังเป็นปัญหาหลักในการใช้งานระบบบล็อกเชนกับอุตสาหกรรมทุกด้าน จึงมีการพัฒนาบล็อกเชนหลากหลายแบบที่ลดเวลาในการรับส่งและประมวลผลด้วยการปรับโครงสร้างหรือวิธีการที่ต้องใช้ในการตรวจสอบข้อมูล เช่น บล็อกเชนที่เรียกว่า Hyperledger Sawtooth ที่เป็นที่ยอมรับในการใช้งานร่วมกับระบบขนส่งแบบห่วงโซ่และสามารถพัฒนาได้จากหลากหลายภาษา เช่น Python, JavaScript, Go, C++, Java และ RUST โดยใช้ Proof of Elapsed Time (PoET) เป็น Consensus Algorithm หลักของบล็อกเชนชนิดนี้ โดยมีหลักการว่าเมื่อจะต้องทำการตรวจสอบความถูกต้องของบล็อกเชนและข้อมูล ระบบจะสุ่มค่าเวลาให้กับทุกโหนดในเครือข่ายและโหนดที่สุ่มได้ค่าเวลาน้อยที่สุดจะเป็นผู้ชนะ (ผู้รับรองข้อมูล) ของบล็อกเชน [25] ซึ่งสามารถตรวจสอบและบันทึกข้อมูลในบล็อกเชนได้ 2000 ธุรกรรมต่อวินาที [26]

การเลือกชนิดของบล็อกเชนให้เหมาะสมกับระบบก็เป็นเรื่องหนึ่งที่สำคัญ โดยมีตัวอย่างการใช้บล็อกเชนร่วมกับระบบขนส่งห่วงโซ่ความเย็น [27] เป็นการเก็บข้อมูลการขนส่งวัคซีนในเครือข่ายบล็อกเชน โดยอุปกรณ์จะส่งอุณหภูมิและพิกัดปัจจุบันมาบันทึกไว้อย่างต่อเนื่อง ใช้ร่วมกับบล็อกเชนที่เรียกว่า Ethereum ซึ่งต้องใช้ระยะเวลาในการตรวจสอบข้อมูลเล็กน้อยแต่การดึงข้อมูลการแสดงผลกลับใช้เวลามากกว่า หากใช้ร่วมกับสินค้าที่ไวต่ออุณหภูมิเช่นวัคซีนหรืออาหารสดอาจส่งผลต่อความเร็วในการรับมือกับอุณหภูมิรอบสินค้าหากเกิดการเปลี่ยนแปลงได้ไม่ทันการไปด้วย ผู้วิจัยจึงมีแนวคิดที่จะใช้ระบบบล็อกเชนเป็นเครือข่ายเก็บข้อมูลหลักและส่งต่อข้อมูลไปเก็บยังฐานข้อมูล MySQL เพื่อให้ระบบที่จะพัฒนาสามารถดึงข้อมูลไปแสดงผลได้อย่างรวดเร็วแบบเรียลไทม์ และใน

ระหว่างการส่งต่อข้อมูลนี้จะให้ระบบประมวลผลเพื่อตรวจสอบเงื่อนไขของอุณหภูมิและพิกัดของรถขนส่งค่าความคุ้มกันไปด้วย

## 2.2 ทฤษฎีและเครื่องมือในการพัฒนาแอปพลิเคชัน

### 2.2.1 วิทยาการเข้ารหัสลับ

วิทยาการเข้ารหัสลับ หรือ Cryptography คือกระบวนการทางเทคนิคอย่างหนึ่งที่ใช้การแปลงข้อมูลเป็นรหัสลับ โดยปิดข้อมูลที่ส่ง รับ หรือจัดเก็บ ไม่ให้สามารถอ่านได้โดยง่าย โดยหลักจะมีการใช้อัลกอริทึมรูปแบบต่าง ๆ ในการแปลงข้อมูลโดยที่ฝ่ายรับข้อมูลจะมีคีย์หรือวิธีการในการถอดรหัสเพื่ออ่านข้อมูลนั้นได้ โดยข้อมูลที่ยังไม่ได้เข้ารหัสจะถูกเรียกว่า “ข้อความธรรมดา” ซึ่งผู้รับสารจะสามารถอ่านข้อมูลหรือข้อความนั้นได้โดยตรง และเรียกข้อมูลที่เข้ารหัสแล้วว่า “ข้อความไซเฟอร์” ซึ่งเป็นข้อความที่ไม่สามารถอ่านได้โดยตรง การเข้ารหัสแบ่งได้เป็น 3 ประเภท ได้แก่ อัลกอริทึมแบบคีย์สมมาตร (Symmetric key algorithms) อัลกอริทึมแบบคีย์อสมมาตร (Asymmetric key algorithms) และฟังก์ชันแฮช [28]

การเข้ารหัสแบบสมมาตร จะใช้กุญแจลับในการติดต่อกันระหว่างผู้ส่งและผู้รับข้อมูล โดยจะใช้กุญแจอันเดียวกันทั้งในการเข้ารหัสและถอดรหัสข้อความ ก่อนที่จะส่งข้อมูลที่ถูกรหัสแล้วผ่านเครือข่ายรูปแบบหนึ่ง โดยทั้งผู้ส่งและผู้รับข้อมูลจะต้องมี กุญแจ และ อัลกอริทึม ที่ตกลงร่วมกันแล้วเพื่อใช้ในการเข้ารหัสและถอดรหัส ตัวอย่างอัลกอริทึมของการเข้ารหัสแบบสมมาตร เช่น

#### 1. Advance Encryption Standard (AES)

เป็นการเข้ารหัสระดับบล็อกข้อมูล ข้อมูลจึงถูกแบ่งออกเป็นบล็อกขนาด 128 บิต ก่อนที่จะเข้ารหัสด้วยคีย์ 256 บิต โดยการเข้ารหัส AES 256 บิต ถือเป็นมาตรฐานระหว่างประเทศที่ให้ความมั่นใจด้านการรักษาความปลอดภัยของข้อมูลในระดับสูง เป็นที่ยอมรับกับหลายประเทศรวมถึงหน่วยงานอื่น ๆ มากมาย เนื่องจากการเข้ารหัส AES 256 บิต โดยพื้นฐานแล้วไม่สามารถถูกเจาะรหัสได้ ทำให้ถือเป็นมาตรฐานการเข้ารหัสที่ปลอดภัยที่สุดในปัจจุบัน สาเหตุที่ไม่สามารถถูกเจาะรหัสได้เนื่องจาก AES ครบคลุมทั้ง 128, 192 และ 256 บิต ซึ่งเป็นตัวเลขระบุจำนวนบิตของข้อมูลของกุญแจในบล็อกเข้ารหัสและถอดรหัสแต่ละชุด ทุก ๆ บิตที่ถูกเพิ่มเข้ามาหมายความว่าทำให้จำนวนคีย์ที่เกิดขึ้นเพิ่มเป็นสองเท่า ดังนั้นการเข้ารหัสด้วย AES-256 จะหมายถึง 2 ยกกำลัง 256 ซึ่งเป็นจำนวนที่มหาศาลมาก หากผู้โจมตีจะคาดเดาลำดับที่ถูกต้องของบิตทั้งหมดจะต้องใช้เวลาและกำลังในการประมวลผลที่สูงมาก

2. Data Encryption Standard (DES) ถูกพัฒนาโดย IBM และใช้กันอย่างกว้างขวางในปี 1970 เนื่องจากในตอนนั้นระบบประมวลผลต่าง ๆ ยังมีกำลังที่ไม่สูงนัก แต่ในปัจจุบันนับว่าเป็นระบบเก่าที่ไม่มีความปลอดภัยในการป้องกันข้อมูลมากพอแล้ว

ในส่วนของการเข้ารหัสแบบอสมมาตร เป็นการแก้ปัญหาโดยใช้หลักการของ กุญแจสาธารณะ (Public Key) และ กุญแจส่วนตัว (Private Key) โดยที่กุญแจสาธารณะจะถูกเปิดเผยในเครือข่าย ส่วนกุญแจส่วนตัวจะถูกเก็บไว้กับผู้ใช้งานแต่ละคนเท่านั้น ในการเข้ารหัสและถอดรหัสข้อมูลจำเป็นจะต้องใช้กุญแจเป็นคู่ โดยที่บุคคลที่เข้ารหัสข้อมูลจะได้รับ กุญแจสาธารณะ ในการเข้ารหัส ส่วนบุคคลที่สามารถถอดรหัสได้คือบุคคลที่มี กุญแจส่วนตัว ที่เข้าคู่กับกุญแจสาธารณะเท่านั้น กล่าวคือ ไม่ว่าผู้ใช้คนไหนก็สามารถเข้ารหัสได้เพราะทุกคนมีกุญแจสาธารณะ แต่คนที่สามารถถอดรหัสได้คือคนที่มีความรู้ส่วนตัวคนเดียวเท่านั้น [29] โดยกุญแจส่วนตัวจะต้องถูกเก็บไว้อย่างรัดกุม ตัวอย่างการเข้ารหัสแบบอสมมาตร เช่น อาร์เอสเอ (RSA) เป็นอัลกอริทึมการเข้ารหัสแบบกุญแจอสมมาตร โดยใช้ความรู้เรื่องของเลขคณิตมอดุลาร์ (Modular Arithmetic) มาช่วยในคำนวณภายใต้การเข้ารหัส RSA ข้อความจะถูกเข้ารหัสด้วยรหัสที่เรียกว่ากุญแจสาธารณะซึ่งสามารถแบ่งปันได้อย่างเปิดเผย เนื่องจากคุณสมบัติทางคณิตศาสตร์ที่แตกต่างกันของอัลกอริทึม RSA โดยเมื่อข้อความได้รับการเข้ารหัสด้วยกุญแจสาธารณะแล้วจะสามารถถอดรหัสได้ด้วยกุญแจอีกชนิดหนึ่งซึ่งเรียกว่ากุญแจส่วนตัวเท่านั้น ผู้ใช้ RSA แต่ละคนจะมีคู่ของกุญแจซึ่งประกอบด้วยกุญแจสาธารณะและกุญแจส่วนตัวอยู่คู่กับตัว การเข้ารหัส RSA จะใช้ร่วมกับฟังก์ชันการคำนวณทางคณิตศาสตร์ ได้แก่ ฟังก์ชันประตูกล (Trapdoor Function) และ ฟังก์ชันโทเทนต์ของออยเลอร์ (Carmichael Function)

ประเภทสุดท้ายคือฟังก์ชันแฮช หรือ Hashing Function เป็นการเข้ารหัสโดยใช้อัลกอริทึมทางคณิตศาสตร์ทางเดียวที่ใช้ในการจับคู่ข้อมูลทุกขนาดเข้ากับบิตข้อความที่มีขนาดคงที่ ฟังก์ชันแฮชมีการใช้กันอย่างแพร่หลายในการปฏิบัติด้านความปลอดภัยของข้อมูล เช่น ลายเซ็นดิจิทัล รหัสตรวจสอบข้อความ และรูปแบบการตรวจสอบอื่น ๆ โดยมีคุณสมบัติคือ 1) ข้อความเดียวกันส่งผลให้ค่าแฮชเดียวกันเสมอ 2) ข้อความสองข้อความที่แตกต่างกัน ไม่สามารถมีค่าแฮชเดียวกันได้ 3) การเปลี่ยนแปลงเล็กน้อยในข้อความควรส่งผลให้ค่าแฮชที่ได้ออกมาเปลี่ยนไปอย่างมหาศาลเพื่อให้ไม่สามารถเชื่อมโยงกับค่าข้อความดั้งเดิมได้ 4) คำนวณค่าแฮชได้อย่างรวดเร็ว ฟังก์ชันแฮชที่มักพบเจอได้บ่อย ได้แก่ MD5, SHA-1, SHA-2 [30] โดยมีความแตกต่างกัน คือ MD5 ย่อมาจาก Message-Digest Algorithm 5 คือ รูปแบบการเข้ารหัสชนิดหนึ่งให้อยู่ในอีกรูปแบบที่มีขนาดคงที่ ทำให้ไม่สามารถเรียกดูข้อมูลต้นฉบับได้ ทำได้เพียงตรวจสอบว่าข้อมูลที่ให้มาแต่ละครั้งเหมือนกันหรือไม่ โดย MD5 ผลิตข้อมูลที่มีขนาดคงที่ 128 บิต ต่อมาได้พัฒนารูปแบบการเข้ารหัสที่เรียกว่า SHA ย่อมาจาก Secure Hash Algorithm หรือ อัลกอริทึมแฮชที่ปลอดภัย โดย SHA-1 เป็นฟังก์ชันแฮชที่สามารถแปลงข้อมูลประเภทข้อความที่มีความยาวต่าง ๆ กัน ให้อยู่ในรูปแบบคงที่ 160 บิต หรือมักจะเห็นเป็นตัวเลขฐานสิบหกจำนวน 40 ตัวอักษร และได้ต่อยอดไปเป็น SHA-2 หรือที่มักเรียกว่า SHA-256

เป็นการเข้ารหัสที่สามารถแปลงข้อมูลประเภทข้อความที่มีความยาวต่าง ๆ ให้เป็นข้อมูลที่มีขนาดคงที่ โดยทั่วไปจะแสดงข้อมูลเป็นตัวเลขฐานสิบหกที่มีค่าคงที่ เช่น รหัส 64 ตัวอักษร เป็นต้น

```

“previous_block”:
“473790dc170278bf0f65ffaeeee669b4eed029c95d50b80219ee84d945341513”
“index”: 0,
“nonce”: 61093,
“timestamp”: 1660325795.1982615,
“transaction”: []

```



```
8db1ea33e6c8de473a31c842f40bf2f1478be094a0f58a2a9bc96f1d9b589762
```

รูปที่ 2.2 ตัวอย่างการนำข้อความจากอาเรย์มาเข้ารหัสแบบ SHA-256

ในงานวิจัยนี้ผู้วิจัยเลือก SHA-256 เป็นอัลกอริทึมพื้นฐานในการเข้ารหัสข้อมูลที่ใช้กับบล็อกเชน เนื่องจากข้อมูลมีขนาดที่แน่นอนง่ายต่อการตรวจสอบรวมทั้งเป็นอัลกอริทึมที่คำนวณได้อย่างรวดเร็วทำให้เหมาะกับการเก็บข้อมูลที่ต้องการความแม่นยำด้านเวลา ตัวอย่างข้อมูลที่ถูกเข้ารหัสเป็นดังรูปที่ 2.2

### 2.2.2 Docker

Docker เป็นแพลตฟอร์มซอฟต์แวร์ชนิดหนึ่งซึ่งช่วยให้ผู้ใช้สามารถสร้าง ทดสอบ และติดตั้งแอปพลิเคชันได้อย่างรวดเร็ว โดย Docker จะบรรจุซอฟต์แวร์ลงไปเป็นหน่วยที่เป็นมาตรฐานที่เรียกว่าคอนเทนเนอร์ (Container) ซึ่งจะมีทุกสิ่งที่ซอฟต์แวร์ต้องใช้ในการเรียกใช้งาน เช่น ไลบรารีเครื่องมือสำหรับระบบ รวมถึงโค้ดด้วย เมื่อใช้งาน Docker ผู้ใช้จะสามารถติดตั้งใช้งานและปรับแอปพลิเคชันให้เหมาะสมกับทุกสภาพแวดล้อม (Environment) โดย Docker สามารถรองรับการใช้งานได้กับหลากหลายระบบปฏิบัติการไม่ว่าจะเป็น Windows, Linux, MAC เป็นต้น [31] องค์ประกอบพื้นฐานของ Docker จะประกอบด้วย

1. Docker File คือ โค้ดคำสั่งสำหรับการสร้าง Docker Image แต่ละตัว

2. Docker Image คือแม่แบบที่ใช้ในการสร้าง Docker Container โดยประกอบไปด้วยแอปพลิเคชันต่าง ๆ ที่จะทำงานเมื่อมีการเรียกใช้งาน Docker Container นั้น ๆ รวมทั้งการตั้งค่าจำลองสภาพแวดล้อมที่เหมาะสมกับการทำงานของโค้ดหรือแอปพลิเคชันเหล่านั้นด้วย

3. Docker Container คือที่บรรจุรวมของแอปพลิเคชัน โค้ด ไฟล์ที่กำหนดไว้จาก Docker File สภาพแวดล้อมที่จำเป็นต่อการทำงาน รวมถึงองค์ประกอบต่าง ๆ ที่จำเป็นต่อการทำงานของแอปพลิเคชันด้วย

ความน่าสนใจที่ผู้วิจัยเลือกใช้ Docker ในการรันแอปพลิเคชันบนเครื่องแม่ข่าย คือ Docker มีขนาดเล็ก สามารถใช้งานได้ง่าย และติดตั้งได้อย่างรวดเร็ว สะดวกในการเริ่มหรือหยุดการทำงาน รวมถึงสั่งให้แอปพลิเคชันทำงานใหม่อีกครั้งได้อย่างง่ายดาย รวมถึงการย้ายไปใช้งานในเครื่องแม่ข่ายตัวอื่นก็สามารถดำเนินการได้โดยง่าย ผู้ใช้ไม่จำเป็นต้องติดตั้งระบบปฏิบัติการใหม่อีกครั้งเพื่อติดตั้ง Docker Container รวมทั้งไม่จำเป็นต้องตั้งค่าอะไรเพิ่มเติมด้วย เนื่องจากสามารถนำ Docker Image ที่ได้จาก Docker File ไปใช้งานได้เลยทันที ช่วยลดปัญหาที่สภาพแวดล้อมของเครื่องแม่ข่ายแตกต่างกันทำให้ไม่สามารถเปิดใช้งานแอปพลิเคชันได้อีกด้วย Docker ยังมีความต้องการในการใช้หน่วยความจำและหน่วยประมวลผลที่น้อยกว่าการติดตั้งอุปกรณ์จำลอง (Virtual Machine) หมายความว่า Docker ใช้งานทรัพยากรได้ประหยัดกว่า ทั้งนี้ในทรัพยากรที่มีเท่ากันผู้ใช้จะสามารถสร้าง Docker Container ได้มากกว่าการสร้างอุปกรณ์จำลองนั่นเอง

### 2.2.3 Flask เฟรมเวิร์ก



รูปที่ 2.3 ตัวอย่างการใช้งาน Flask เพื่อแสดงผลแบบเว็บแอปพลิเคชัน

Flask เฟรมเวิร์ก เป็นเฟรมเวิร์กที่ใช้สำหรับการสร้างเว็บไซต์หรือเว็บแอปพลิเคชันที่พัฒนาขึ้นมาเพื่อใช้กับภาษาไพทอนโดยเฉพาะดังตัวอย่างในรูปที่ 2.3 เนื่องจาก Flask เป็นไมโครเฟรมเวิร์กที่มีขนาดเล็กและใช้งานง่าย ไม่จำเป็นต้องใช้งานควบคู่กับไลบรารีตัวอื่น ๆ [32] นอกจาก

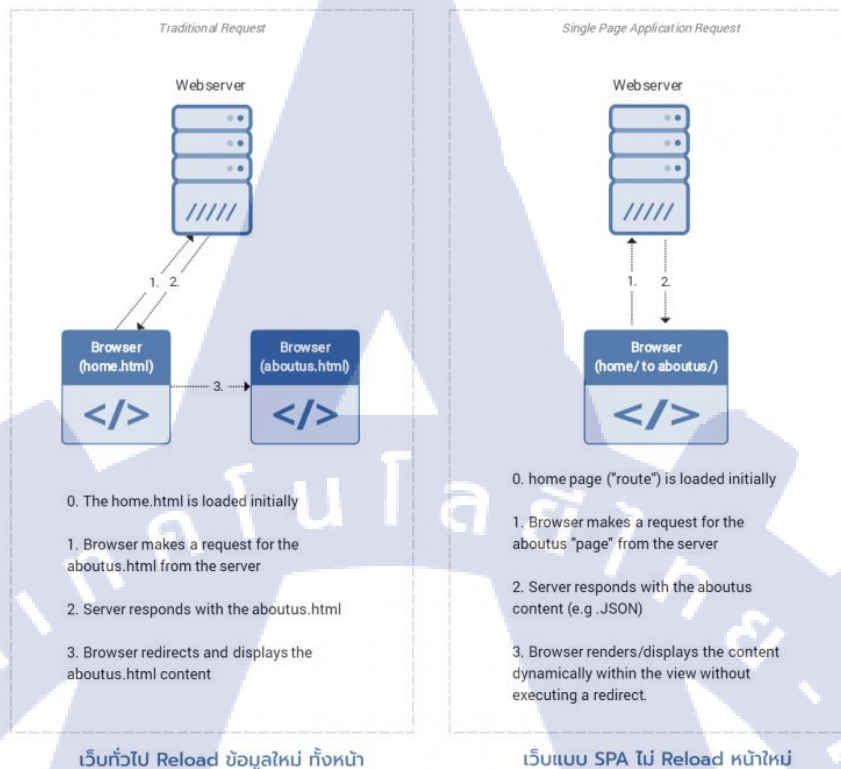
การสร้างเว็บแอปพลิเคชันแล้ว Flask ยังสามารถนำมาสร้างเป็นส่วนต่อประสานโปรแกรมประยุกต์ (API) ที่ใช้ในการติดต่อสื่อสารระหว่างผู้ใช้หรือโปรแกรมอื่น ๆ มายังเครื่องแม่ข่ายหรือเว็บแอปพลิเคชันที่ใช้งานได้อีกด้วย

Flask มีพื้นฐานมาจาก อินเทอร์เฟซเกตเวย์ของเว็บเซิร์ฟเวอร์ (WSGI) ซึ่งเป็นข้อกำหนดที่อธิบายวิธีที่เว็บเซิร์ฟเวอร์สามารถสื่อสารกับเว็บแอปพลิเคชัน และวิธีที่เว็บแอปพลิเคชันสามารถเชื่อมโยงเข้าด้วยกันเพื่อดำเนินการตามคำขอ (request) แต่ครั้งได้ที่เรียกว่า Werkzeug โดยในแอปพลิเคชันแบบ WSGI จะรับคำขอเพียงครั้งเดียวและส่งคืนการตอบกลับในแต่ละครั้ง และใช้ร่วมกับเทมเพลตเอ็นจิน ที่เป็นเทมเพลตในการแสดงผลเว็บไซต์ที่เรียกว่า jinja2 โดยหลักการทำงานคือเทมเพลตเอ็นจินจะทำการแทนค่าตัวแปรต่าง ๆ ที่ส่งจากฝั่งของเครื่องแม่ข่ายลงไปในเทมเพลต จากนั้นแปลงเทมเพลตนั้นให้อยู่ในรูปแบบของ HTML ไฟล์ แล้วส่งกลับมาแสดงยังผู้ใช้ ซึ่งทำให้การออกแบบหน้าตาของเพจทำได้ง่ายขึ้น เนื่องจากงานวิจัยนี้พัฒนาบล็อกเชนโดยใช้ภาษาไพทอนเป็นหลัก ผู้วิจัยจึงเลือก Flask ที่สามารถใช้งานได้ง่ายกับภาษาไพทอน โดยให้เป็นส่วนต่อประสานโปรแกรมประยุกต์ด้วยการรับข้อมูลจากอุปกรณ์หรือผู้ใช้งานทางเว็บแอปพลิเคชันโดยตรง ก่อนที่จะส่งข้อมูลไปประมวลผลและเก็บข้อมูลลงในบล็อกเชนต่อไป

#### 2.2.4 Vue.js

Front-end คือส่วนหน้าของเว็บไซต์หรือเว็บแอปพลิเคชันที่สามารถเห็นได้ผ่านเบราว์เซอร์ ซึ่งส่วนที่ผู้ใช้งานเห็นผ่านเว็บไซต์จะถูกเขียนขึ้นด้วย HTML, CSS หรือ JavaScript โดย Front-end Framework ก็คือชุดโค้ดที่ช่วยให้นักพัฒนาในส่วน Front-end สามารถทำงานได้อย่างสะดวกมากขึ้น โดยเป็นการรวบรวมโค้ดการทำงานของ HTML หรือ CSS มาทำในรูปแบบสำเร็จรูป ที่มีการจัดวางโครงสร้าง ปุ่ม ตัวอักษร มาให้เบื้องต้นแล้ว

Vue.js นับว่าเป็น Front-end Framework ตัวหนึ่ง ที่ช่วยให้นักพัฒนาสามารถเขียนด้วยภาษา JavaScript ได้ง่ายมากขึ้น เขียนสั้นลง ช่วยลดระยะเวลาในการพัฒนางานได้ โดยหน้าที่หลักของ Vue.js คือการพัฒนาส่วนต่อประสานกับผู้ใช้หรือก็คือส่วนที่โต้ตอบกับผู้ใช้ [33] เช่น การคลิกเพื่อแสดงผลข้อมูล หรือการผูกข้อมูลบนแบบฟอร์มหรือกล่องข้อมูลแล้วสามารถแสดงผลข้อมูลออกมาได้แบบเรียลไทม์ ซึ่งจุดเด่นของ Vue.js คือสามารถเรียนรู้ได้ไวเนื่องจากโครงสร้างการทำงานถูกออกแบบมาให้ใช้งานได้อย่างง่าย ใช้ระยะเวลาในการเรียนรู้ไม่นานก็สามารถพัฒนาเว็บแอปพลิเคชันออกมาได้ มีเครื่องมือที่ช่วยในการพัฒนามากมาย เช่น Vue-CLI, Visual Studio Code หรือ Vetur เป็นต้น อีกทั้งมีน้ำหนักเบาและประมวลผลได้อย่างรวดเร็ว เหมาะแก่การพัฒนาเว็บแอปพลิเคชันที่เน้นการแสดงผลข้อมูลที่ย่และง่ายและไม่ซับซ้อน



รูปที่ 2.4 เปรียบเทียบการทำงานของเว็บไซต์ทั่วไป (ซ้าย) กับเว็บไซต์แบบ SPA (ขวา) [34]

หลักการทำงานของ Vue.js คือรูปแบบ Single Page Application (SPA) ซึ่งเป็นเทคนิคการพัฒนาเว็บไซต์ที่มีลักษณะการทำงานใกล้เคียงกับแอปพลิเคชัน ซึ่งได้รับความนิยมเป็นอย่างมากในปัจจุบัน โดยความแตกต่างจากเว็บไซต์โดยทั่วไปคือการแสดงผลของข้อมูล จากรูปที่ 2.4 ในเว็บไซต์รูปแบบเก่าเมื่อผู้ใช้งานต้องการเปลี่ยน URL หรือหน้าเว็บไซต์ ตัวเว็บไซต์จะโหลดซ้ำเพื่อดึงข้อมูลใหม่มาแสดง แต่การทำงานของ SPA ได้รับการออกแบบให้ลดการโหลดซ้ำของเว็บไซต์ โดยเมื่อผู้ใช้งานต้องการเปลี่ยน URL จะไม่จำเป็นต้องโหลดซ้ำ แต่เป็นการใช้ภาษา JavaScript ในการดึงข้อมูลเพียงบางส่วนเพื่อนำมาอัปเดตเฉพาะบริเวณที่มีการเปลี่ยนแปลงของข้อมูลเท่านั้น ส่งผลให้เว็บไซต์รูปแบบ SPA มอบประสบการณ์ผู้ใช้ได้ดีกว่า สามารถตอบสนองได้อย่างรวดเร็วมากยิ่งขึ้น เหมาะกับการพัฒนาเว็บแอปพลิเคชันที่สามารถแสดงผลได้บนหลากหลายอุปกรณ์ เช่น คอมพิวเตอร์ หรือ โทรศัพท์มือถือ

### บทที่ 3

#### วิธีการดำเนินงานวิจัย

##### 3.1 ขั้นตอนการวิจัย

ในงานวิจัยครั้งนี้ ผู้วิจัยได้แบ่งขั้นตอนการดำเนินงานดังนี้

- 3.1.1 วางแผนงานและระยะเวลาในการดำเนินงาน
- 3.1.2 ออกแบบภาพรวมของระบบ
- 3.1.3 ออกแบบฐานข้อมูลของระบบ
- 3.1.4 ออกแบบระบบแจ้งเตือน
- 3.1.5 ศึกษาและพัฒนาระบบ
- 3.1.6 เชื่อมโยงการทำงานของระบบ
- 3.1.7 ทดสอบและปรับปรุงการใช้งาน
- 3.1.8 วัดประสิทธิผลของระบบ

##### 3.2 แผนงานและระยะเวลาในการดำเนินงาน

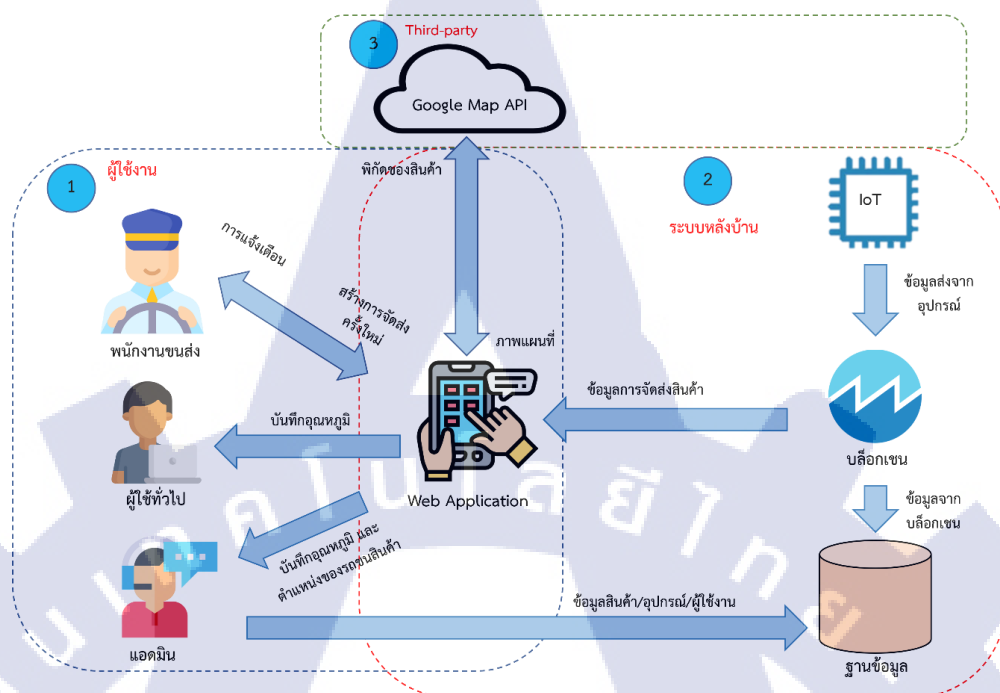
ตารางที่ 3.1 แผนงานและระยะเวลาดำเนินงาน

| แผนงาน                                                                                 | ก.พ. | มี.ค. | เม.ย. | พ.ค. | มิ.ย. | ก.ค. | ส.ค. | ก.ย. |
|----------------------------------------------------------------------------------------|------|-------|-------|------|-------|------|------|------|
| 1. ศึกษาหัวข้อและงานวิจัยที่เกี่ยวข้อง                                                 |      |       |       |      |       |      |      |      |
| 2. ออกแบบโครงสร้างระบบ ได้แก่ ความสัมพันธ์ของข้อมูลในฐานข้อมูล และ ส่วนการทำงานของระบบ |      |       |       |      |       |      |      |      |
| • ฐานข้อมูลผู้ใช้                                                                      |      |       |       |      |       |      |      |      |
| • ฐานข้อมูลสินค้าและ<br>การจัดส่ง                                                      |      |       |       |      |       |      |      |      |
| • ออกแบบข้อมูลของอุปกรณ์ให้เป็นคลาสของโฟตอน                                            |      |       |       |      |       |      |      |      |

ตารางที่ 3.1 แผนงานและระยะเวลาดำเนินงาน (ต่อ)

| แผนงาน                                                                                      | ก.พ. | มี.ค. | เม.ย. | พ.ค. | มิ.ย. | ก.ค. | ส.ค. | ก.ย. |
|---------------------------------------------------------------------------------------------|------|-------|-------|------|-------|------|------|------|
| 3. พัฒนาส่วนของบล็อกเชน                                                                     |      |       |       |      |       |      |      |      |
| 4. ติดตั้งบล็อกเชนบนเครื่องแม่ข่าย                                                          |      |       |       |      |       |      |      |      |
| 5. ติดตั้งฐานข้อมูล MySQL ตามที่ออกแบบไว้                                                   |      |       |       |      |       |      |      |      |
| 6. เขียนโปรแกรมเพื่อส่งข้อมูลไปยังบล็อกเชน                                                  |      |       |       |      |       |      |      |      |
| 7. ออกแบบและพัฒนาระบบแจ้งเตือน                                                              |      |       |       |      |       |      |      |      |
| 8. พัฒนาส่วนของ Web Application                                                             |      |       |       |      |       |      |      |      |
| <ul style="list-style-type: none"> <li>เชื่อมข้อมูลจากบล็อกเชนและฐานข้อมูล MySQL</li> </ul> |      |       |       |      |       |      |      |      |
| <ul style="list-style-type: none"> <li>เชื่อมกับ LINE</li> </ul>                            |      |       |       |      |       |      |      |      |
| <ul style="list-style-type: none"> <li>ออกแบบหน้าตา Web Application</li> </ul>              |      |       |       |      |       |      |      |      |
| 9. ทดสอบการใช้งานและปรับปรุง                                                                |      |       |       |      |       |      |      |      |
| 10. วัดประสิทธิภาพของระบบ                                                                   |      |       |       |      |       |      |      |      |
| 11. เขียนสรุปงานวิจัย                                                                       |      |       |       |      |       |      |      |      |

### 3.3 ภาพรวมของระบบ



รูปที่ 3.1 ภาพรวมการทำงานทั้งหมดของระบบ

ในการทำงานของระบบตามที่วางแผนไว้ จะสามารถแบ่งออกได้เป็น 3 ส่วน ได้แก่ ส่วนของผู้ใช้งานและสิทธิ์ต่างๆ ส่วนระบบหลังบ้าน (backend) ที่ดูแลการรับส่งข้อมูลรวมถึงการบันทึกข้อมูล และส่วนของบริษัทที่สาม (Third-Party) ตามที่แสดงไว้ในรูปที่ 3.1

#### 3.3.1 ส่วนของผู้ใช้งานและสิทธิ์ต่าง ๆ

เนื่องจากในระบบมีการแบ่งผู้ใช้งานทั้งหมดออกเป็น 3 ประเภท ซึ่งแต่ละประเภทจะมีสิทธิ์การใช้งานระบบแตกต่างกัน รวมถึงข้อมูลที่สามารถเข้าถึงได้ก็แตกต่างกันอีกด้วย โดยประเภทของผู้ใช้แบ่งออกได้ดังนี้

##### 1) ผู้ใช้ทั่วไป (User)

เมื่อใช้งานระบบเป็นครั้งแรก ทุกคนจะถูกตั้งค่าเป็นประเภทผู้ใช้ทั่วไปเสมอ ผู้ใช้ทั่วไปสามารถดูบันทึกอุณหภูมิในระหว่างการขนส่งสินค้าที่อยู่ในสถานะ ‘กำลังจัดส่ง’ หรือ ‘จัดส่งสำเร็จ’

##### 2) พนักงานขนส่ง (Driver)

จะถูกแต่งตั้งผ่านแอดมินเพื่อทำหน้าที่จัดส่งสินค้า โดยพนักงานขนส่งสามารถดูบันทึกอุณหภูมิได้เช่นเดียวกับผู้ใช้ทั่วไป และยังมีหน้าที่สำคัญในการสร้างข้อมูลการจัดส่งสินค้าใหม่ทุกครั้งเมื่อมีการเริ่มขนส่งหรือออกจากจุดเริ่มต้น ในระหว่างการขนส่งหากอุณหภูมิภายในรถขนส่งสินค้ามี

อุณหภูมิที่มากกว่า/น้อยกว่าเกณฑ์ที่กำหนดไว้ หรือ พิกัดของรถขนส่งสินค้าอยู่จุดเดิมเป็นเวลานาน จำนวนครั้งที่ตั้งค่าไว้จากระบบ จะมีการแจ้งเตือนจาก LINE Official ไปยังอุปกรณ์ของพนักงานที่รับผิดชอบเพื่อให้ทราบถึงความผิดปกติเกิดขึ้น และเมื่อขับถึงจุดหมายปลายทางแล้ว พนักงานมีหน้าที่ในการกรดยานการจัดส่งสำเร็จเพื่อปิดรอบการจัดส่งนั้น ๆ

### 3) แอดมิน (Administrator)

เป็นผู้ดูแลระบบรวมถึงดูแลข้อมูลทั้งหมด ได้แก่ ข้อมูลผู้ใช้ ข้อมูลสินค้า และข้อมูลอุปกรณ์ โดยแอดมินสามารถเพิ่ม/ลบ/แก้ไขข้อมูลเหล่านั้นได้ แอดมินยังสามารถตรวจสอบบันทึกอุณหภูมิและพิกัดปัจจุบันของรถขนส่งสินค้าในการจัดส่งทุกรอบได้เช่นกัน

### 3.3.2 ส่วนระบบหลังบ้าน backend

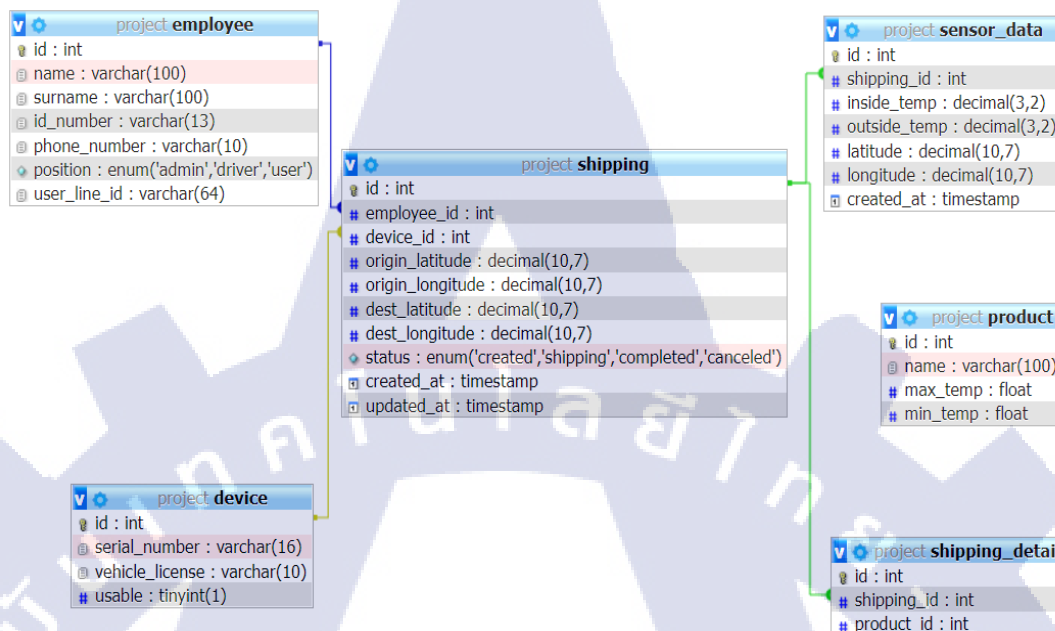
เป็นส่วนที่ใช้จัดการประมวลผลข้อมูลต่างๆ เพื่อให้ผู้ใช้งานสามารถเห็นข้อมูลผ่านทาง Web Application ได้ เริ่มจากอุปกรณ์ IoT ได้แก่ เซนเซอร์วัดอุณหภูมิที่ติดไว้ในรถขนส่งสินค้าและนอกรถขนส่งสินค้ารวมถึงโมดูลที่ใช้เก็บค่าพิกัด GPS ปัจจุบัน (ละติจูด/ลองจิจูด) ทำการส่งค่าทั้งหมดพร้อมกับหมายเลขรหัสอุปกรณ์ของ Raspberry PI ไปยัง API ที่สร้างไว้ หลังจากนั้นข้อมูลจากการจัดส่งจะถูกเก็บไว้ในบล็อกเชนเดียวกันในรูปแบบ JSON ก่อนที่จะถูกแปลงเป็นรูปแบบ SQL Schema และบันทึกข้อมูลในฐานข้อมูล MySQL เช่นเดียวกัน ในระหว่างการบันทึกข้อมูล หากระบบตรวจพบว่าข้อมูลอุณหภูมิไม่อยู่ในช่วงเกณฑ์ที่กำหนดไว้ จะส่งการแจ้งเตือนกลับไปยังพนักงานขนส่งที่รับผิดชอบ

### 3.3.3 ส่วนของบุคคลที่สาม (Third Party)

Google Maps Platform (GMP) เป็นแพลตฟอร์มแผนที่ของ Google ที่รวบรวมแผนที่ภูมิประเทศกว่า 200 ประเทศและภูมิภาค ผู้ใช้สามารถกำหนดแผนที่ กำหนดเส้นทางสำหรับจุดหมาย และใช้ประโยชน์จากแผนที่ในการทำธุรกิจต่าง ๆ ได้ โดยรองรับการทำงานทั้ง Web Application และ Mobile Application โดยแพลตฟอร์มนี้สามารถแบ่งได้ 3 กลุ่ม ได้แก่

- 1) Maps สำหรับแสดงผลข้อมูลต่างๆ รวมถึงสภาพของแผนที่จากฐานข้อมูล Google
- 2) Routes สำหรับการคำนวณเส้นทาง ระยะทางและระยะเวลาจากสถานที่หนึ่งไปยังอีกสถานที่หนึ่ง
- 3) Places สำหรับการเข้าถึงตำแหน่งของสถานที่ และข้อมูลที่จำเป็น เช่น เบอร์โทรศัพท์ ที่อยู่ เวลาทำการ

### 3.4 การออกแบบฐานข้อมูลของระบบ



รูปที่ 3.2 ฐานข้อมูลของระบบ ประกอบด้วย ข้อมูลการขนส่ง ผู้ใช้ อุปกรณ์ สินค้า และข้อมูลจาก เซนเซอร์

ฐานข้อมูล MySQL ในระบบเป็นแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นการเก็บข้อมูลในรูปแบบตาราง 2 มิติ และมีการเชื่อมโยงข้อมูลระหว่างตารางโดยใช้ส่วนที่เหมือนกันของทั้งสองตาราง (ในที่นี้คือ id) เป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลที่ออกแบบในระบบเป็นดังแสดงในรูปที่ 3.2

ตารางหลัก (Master Tables) เป็นตารางที่ข้อมูลแต่ละแถวจะไม่มีการบันทึกข้อมูลซ้ำกัน ผู้ใช้งานตำแหน่งแอดมินสามารถ เพิ่ม/แก้ไข หรือลบข้อมูลที่ไม่ถูกใช้งานได้ ได้แก่

#### 1. Employee

ตารางที่ 3.2 ตารางข้อมูลผู้ใช้งาน

| No. | ข้อมูล  | ชนิด          | รายละเอียด            | Key |
|-----|---------|---------------|-----------------------|-----|
| 1.  | id      | INT           | รหัสของตาราง employee | PK  |
| 2.  | name    | VARCHAR (100) | ชื่อของผู้ใช้งาน      |     |
| 3.  | surname | VARCHAR (100) | นามสกุลของผู้ใช้งาน   |     |

ตารางที่ 3.2 ตารางข้อมูลผู้ใช้งาน (ต่อ)

| No. | ข้อมูล       | ชนิด         | รายละเอียด                                                                                       | Key |
|-----|--------------|--------------|--------------------------------------------------------------------------------------------------|-----|
| 4.  | id_number    | VARCHAR (13) | รหัสประชาชนของผู้ใช้งาน                                                                          |     |
| 5.  | phone_number | VARCHAR (10) | เบอร์โทรศัพท์ของผู้ใช้                                                                           |     |
| 6.  | position     | ENUM         | สิทธิ์ของผู้ใช้งาน แบ่งเป็น<br>user = ผู้ใช้งานทั่วไป<br>driver = พนักงานขับรถ<br>admin = แอดมิน |     |
| 7.  | user_line_id | VARCHAR (64) | รหัส LIFF 64 หลัก ที่ได้จากบัญชี LINE ของผู้ใช้งาน                                               |     |

เก็บข้อมูลส่วนตัว (ชื่อ นามสกุล รหัสบัตรประชาชน เบอร์โทรศัพท์) และสิทธิ์การเข้าใช้งานของผู้ใช้ รวมถึง User LINE ID ด้วย เพื่อให้ระบบสามารถส่งข้อความแจ้งเตือนได้อย่างถูกต้อง

## 2. Device

ตารางที่ 3.3 ตารางข้อมูลอุปกรณ์

| No. | ข้อมูล          | ชนิด         | รายละเอียด                                                                    | Key |
|-----|-----------------|--------------|-------------------------------------------------------------------------------|-----|
| 1.  | id              | INT          | รหัสของตาราง device                                                           | PK  |
| 2.  | serial_number   | VARCHAR (16) | หมายเลขรหัสอุปกรณ์ 16 หลัก ที่อ่านได้จาก Raspberry PI                         |     |
| 3.  | vehicle_license | VARCHAR (10) | เลขทะเบียนรถที่นำอุปกรณ์ไปติดตั้ง                                             |     |
| 4.  | usable          | TINYINT (1)  | สถานะการใช้งานของอุปกรณ์ โดย<br>0 = ไม่สามารถใช้งานได้<br>1 = สามารถใช้งานได้ |     |

เก็บหมายเลขรหัสอุปกรณ์ Raspberry PI ที่เชื่อมกับหมายเลขทะเบียนรถที่ใช้จัดส่งสินค้า เพื่อให้แยกได้ว่าข้อมูลที่ส่งจากเซนเซอร์นั้นถูกส่งมาจากรถคันไหน และนำมาแสดงผลบน Web Application ได้ ค่า usable ใช้เพื่อบอกสถานะการใช้งานของอุปกรณ์ชิ้นนั้น จะใช้เพื่อตรวจสอบ

เงื่อนไขเมื่อพนักงานขนส่งจะสร้างการจัดส่งครั้งใหม่ผ่าน Web Application ซึ่งจะต้องเลือกทะเบียนรถที่ใช้ในการจัดส่งโดยจะแสดงแต่รถที่อุปกรณ์มีสถานะเป็น 1 หรือ สามารถใช้งานได้ เท่านั้น

### 3. Product

ตารางที่ 3.4 ตารางข้อมูลสินค้า

| No. | ข้อมูล   | ชนิด          | รายละเอียด                              | Key |
|-----|----------|---------------|-----------------------------------------|-----|
| 1.  | id       | INT           | รหัสของตาราง product                    | PK  |
| 2.  | name     | VARCHAR (100) | ชื่อของสินค้า                           |     |
| 3.  | max_temp | FLOAT         | อุณหภูมิสูงสุดที่เหมาะสมในการเก็บสินค้า |     |
| 4.  | min_temp | FLOAT         | อุณหภูมิต่ำสุดที่เหมาะสมในการเก็บสินค้า |     |

เก็บชื่อสินค้า อุณหภูมิสูงสุดและอุณหภูมิต่ำสุดที่เหมาะสมแก่การจัดเก็บสินค้าชนิดนั้น ในระหว่างการส่งสินค้า เมื่ออุปกรณ์วัดค่าอุณหภูมิจากสภาพแวดล้อมได้และกำลังส่งข้อมูลไปบันทึก จะต้องมีการเปรียบเทียบว่าอุณหภูมิที่วัดมามีค่ามากกว่าหรือเท่ากับเกณฑ์อุณหภูมิต่ำสุด และมีค่าน้อยกว่าหรือเท่ากับเกณฑ์อุณหภูมิสูงสุดด้วย

ตารางรายละเอียด (Transaction Tables) เป็นตารางที่รับข้อมูลปริมาณมาก ข้อมูลซ้ำกันได้และมีการเชื่อมโยงข้อมูลกับตารางอื่น ได้แก่

#### 1. Shipping

ตารางที่ 3.5 ตารางข้อมูลการจัดส่ง

| No. | ข้อมูล           | ชนิด           | รายละเอียด              | Key |
|-----|------------------|----------------|-------------------------|-----|
| 1.  | id               | INT            | รหัสของตาราง shipping   | PK  |
| 2.  | employee_id      | INT            | รหัสของตาราง employee   | FK  |
| 3.  | device_id        | INT            | รหัสของตาราง device     | FK  |
| 4.  | origin_latitude  | DECIMAL (10,7) | ละติจูดของจุดเริ่มต้น   |     |
| 5.  | origin_longitude | DECIMAL (10,7) | ลองจิจูดของจุดเริ่มต้น  |     |
| 6.  | dest_latitude    | DECIMAL (10,7) | ละติจูดของจุดหมายการส่ง |     |

ตารางที่ 3.5 ตารางข้อมูลการจัดส่ง (ต่อ)

| No. | ข้อมูล         | ชนิด           | รายละเอียด                                                                                              | Key |
|-----|----------------|----------------|---------------------------------------------------------------------------------------------------------|-----|
| 7.  | dest_longitude | DECIMAL (10,7) | ละติจูดของจุดหมายการส่ง                                                                                 |     |
| 8.  | status         | ENUM           | สถานะการจัดส่ง แบ่งเป็น<br>created = ยังไม่จัดส่ง<br>shipping = กำลังจัดส่ง<br>completed = จัดส่งสำเร็จ |     |
| 9.  | created_at     | TIMESTAMP      | เวลาที่สร้างการจัดส่ง                                                                                   |     |
| 10. | updated_at     | TIMESTAMP      | เวลาที่การจัดส่งถูกแก้ไขสถานะครั้งล่าสุด                                                                |     |

เกิดจากผู้ใช้งานตำแหน่งพนักงานขับรถกดเริ่มการจัดส่งครั้งใหม่ โดยมีการระบุพิกัดจุดเริ่มต้นและพิกัดจุดปลายทางแยกเป็นละติจูดและลองจิจูดด้วย สถานะของการจัดส่งแบ่งเป็น 3 แบบคือ ยังไม่จัดส่ง (created) กำลังจัดส่ง (shipping) จัดส่งสำเร็จ (completed) นอกจากนี้ตาราง Shipping มีการเชื่อมโยงกับตาราง employee เพื่อระบุคนขับ และเชื่อมโยงกับตาราง device เพื่อระบุหมายเลขทะเบียนรถที่ใช้ขนส่งอีก

## 2. Shipping\_detail

ตารางที่ 3.6 ตารางรายละเอียดการจัดส่ง

| No. | ข้อมูล      | ชนิด | รายละเอียด            | Key |
|-----|-------------|------|-----------------------|-----|
| 1.  | Shipping_id | INT  | รหัสของตาราง shipping | FK  |
| 2.  | Product_id  | INT  | รหัสของตาราง product  | FK  |

เชื่อมโยงกับตาราง shipping และ product เพื่อบอกว่าในการขนส่งแต่ละรอบนั้น มีสินค้าอะไรบ้าง

### 3. Sensor\_data

ตารางที่ 3.7 ตารางข้อมูลจากอุปกรณ์

| No. | ข้อมูล       | ชนิด           | รายละเอียด                      | Key |
|-----|--------------|----------------|---------------------------------|-----|
| 1.  | id           | INT            | รหัสของตาราง sensor_data        | PK  |
| 2.  | shipping_id  | INT            | รหัสของตาราง shipping           | FK  |
| 3.  | inside_temp  | DECIMAL (3,2)  | อุณหภูมิภายในรถที่วัดได้        |     |
| 4.  | outside_temp | DECIMAL (3,2)  | อุณหภูมิภายนอกที่วัดได้         |     |
| 5.  | latitude     | DECIMAL (10,7) | ละติจูดปัจจุบันของรถ            |     |
| 6.  | longitude    | DECIMAL (10,7) | ลองจิจูดปัจจุบันของรถ           |     |
| 7.  | created_at   | TIMESTAMP      | เวลาที่สร้างรายละเอียดการจัดส่ง |     |

เชื่อมโยงกับตาราง shipping ทำหน้าที่เก็บข้อมูลอุณหภูมิและตำแหน่งปัจจุบันของรถขนส่งสินค้า โดยจะส่งค่าตามเวลาที่กำหนดในระบบ เช่น ทุก 10 วินาที เป็นต้น

### 3.5 เครื่องมือและอุปกรณ์ที่ใช้

3.5.1 เครื่องคอมพิวเตอร์ หรือ Virtual Machine ที่เป็นระบบปฏิบัติการ Ubuntu Server เวอร์ชัน 20.04 หน่วยความจำ 16 GB

3.5.2 Node.js เวอร์ชัน 15 ขึ้นไป

3.5.3 Python เวอร์ชัน 3.7 ขึ้นไป

### 3.6 การพัฒนาบล็อกเชน

บล็อกเชนในงานวิจัยนี้ถูกพัฒนาด้วยภาษาไพทอน โดยเขียนอยู่ในรูปของคลาสชนิดหนึ่ง ข้อมูลที่เก็บอยู่ในคลาส ได้แก่

1. Nodes เป็นเซตของโดเมนเนมหรือไอพีแอดเดรสของโหนดอื่น ๆ ที่เชื่อมต่อกับโหนดปัจจุบัน
2. Target ค่าเป้าหมายที่ใช้เปรียบเทียบกับค่า Nonce โดยเป็นข้อมูลชุดตัวเลข เช่น 0000
3. Chain อาเรย์ของบล็อกทั้งหมดโดยแต่ละบล็อกจะอยู่ในรูปของ JSON Object ที่เก็บข้อมูลที่รับมาจากอุปกรณ์เซนเซอร์อีกที ข้อมูลใน Chain ประกอบด้วย

- 1) Index คือลำดับของบล็อกขึ้นนั้นใน Chain บล็อกแรกจะมี index อยู่ที่ 1

2) Timestamp เวลาปัจจุบันที่ข้อมูลถูกบันทึกลงในบล็อก  
 3) Transaction อาเรย์ของข้อมูลที่เก็บได้จากอุปกรณ์  
 4) Hash\_of\_previous\_block ข้อมูลทั้งหมดของบล็อกชิ้นก่อนหน้าที่เข้ารหัสแบบ SHA256 เก็บในรูปแบบข้อความ

5) Nonce เป็นค่าตัวเลขพิเศษโดยมีวิธีการคำนวณดังโค้ดด้านล่าง เกิดจากการนำ Index, Hash\_of\_previous\_block, Transaction และ Nonce มาเรียงต่อกันเป็นข้อความ และนำไปเข้ารหัสแบบ SHA256 แล้วเทียบกับค่า Target ที่ตั้งไว้ ค่า nonce จะบวกตัวเองเพิ่มขึ้นเรื่อยๆ จนกว่าค่าที่คำนวณออกมาจะได้ตรงกับเงื่อนไข เช่น Target ที่ตั้งไว้มีค่าเท่ากับ 0000 ทำให้ nonce จะเข้ารหัสข้อมูลเรื่อย ๆ จนกว่าข้อมูลที่เข้ารหัสนั้นจะมีค่า 4 ตัวแรกเท่ากับ 0000 นั้นหมายความว่า ทุกครั้งที่ข้อมูลถูกส่งเข้ามาจะไม่มีทางที่จะได้ nonce ค่าเดียวกัน เพราะมีความต่างของข้อมูล ลำดับ รวมถึงเวลาที่ข้อมูลถูกส่งมาซึ่งส่งผลต่อการคำนวณด้วยเช่นกัน

**target** = กำหนดค่าเป้าหมาย เช่น 0000

```
function valid_proof(index, hashed_previous_block, transaction,
nonce) {
```

```
content = {index}{hashed_previous_block}{transaction}{nonce} ต่อ
กันเป็นข้อความ
```

```
content_hash = นำ content เข้ารหัสแบบ SHA256
```

```
if(content_hash[:len(target)]==target){
```

```
    return True
```

```
}
```

```
return False
```

```
}
```

```
function proof(index, hashed_previous_block, transaction){
```

```
    nonce = 0
```

```
    while valid_proof(index, hashed_previous_block, transaction,
nonce) ได้ค่ากลับมาเป็น False {
```

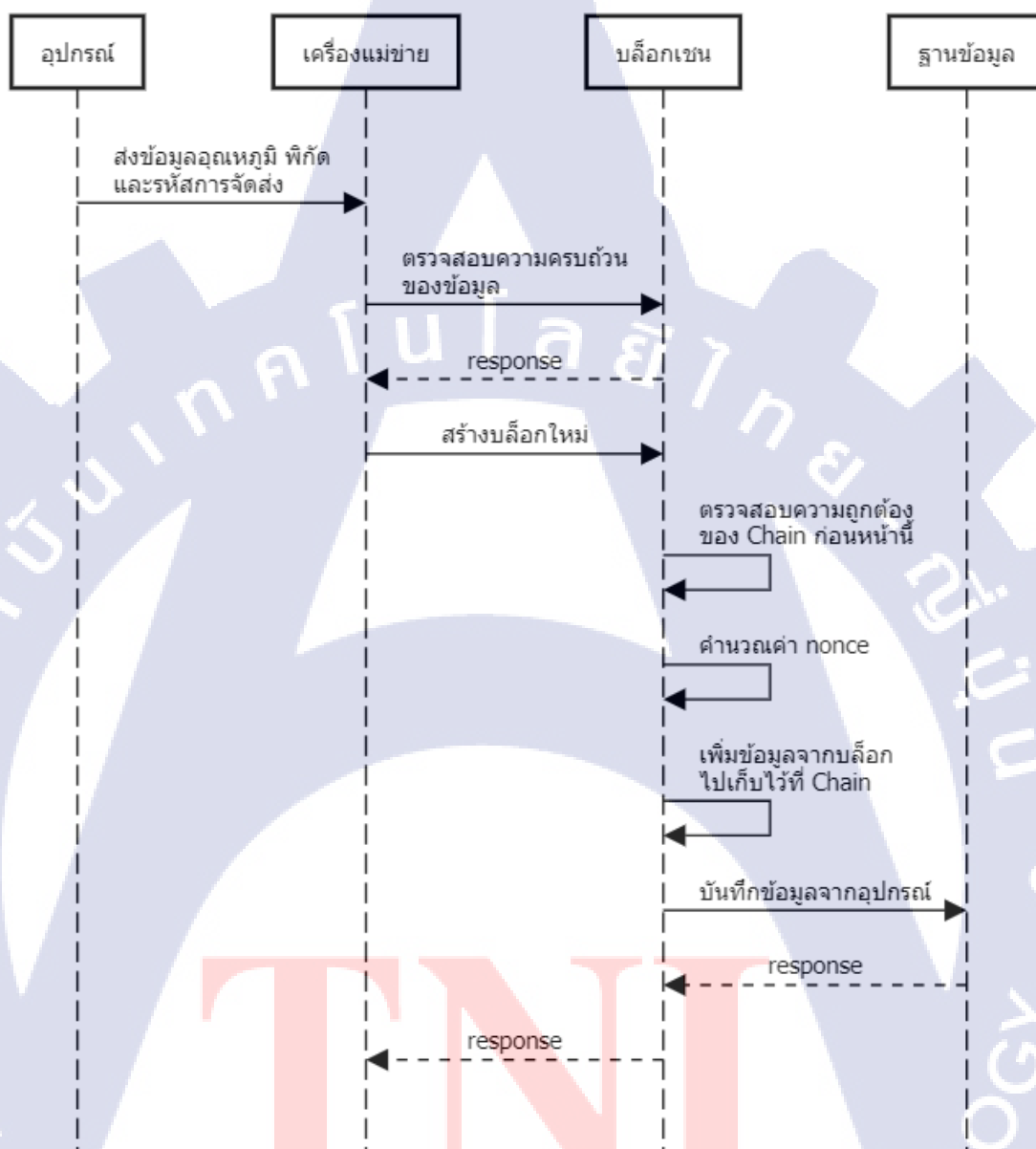
```
        nonce += 1
```

```
}
```

```
เมื่อได้ค่าจาก valid_proof กลับมาเป็น True แล้ว จะส่งค่า nonce ปัจจุบันกลับไปเก็บไว้ใน
บล็อก
```

```
}
```

4. Current\_transactions อาเรย์ของข้อมูลล่าสุดที่เก็บได้จากอุปกรณ์ก่อนที่จะตรวจสอบความถูกต้องของข้อมูลและส่งไปเก็บต่อใน Transaction ของ Chain

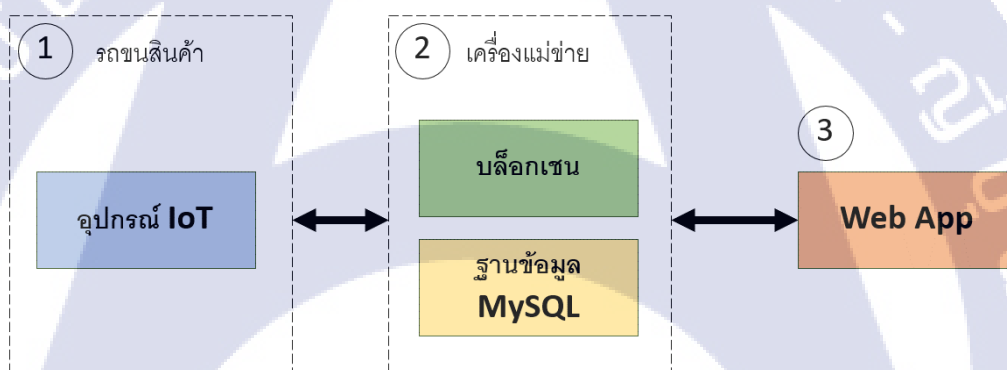


รูปที่ 3.3 การทำงานภายในบล็อกเชนหลังจากอุปกรณ์ส่งข้อมูลเข้ามายังเครื่องแม่ข่าย

เมื่อเครื่องแม่ข่ายรันบล็อกเชนขึ้นมาครั้งแรก จะสร้างบล็อกแรกที่มีข้อความ genesis\_block และเข้ารหัสแบบ SHA256 ขึ้นมา และนำบล็อกนั้นไปเก็บไว้ใน Chain หลังจากนั้นเมื่อข้อมูลถูกส่งมาจากอุปกรณ์ จะตรวจสอบว่ามีค่าที่ต้องการครบทุกค่าหรือไม่ ได้แก่ ละติจูด

ลองจินตนาการว่า อุดมภูมิภายในรถขนส่ง อุดมภูมิภายนอกรถขนส่ง และรหัสการจัดส่ง หากขาดแม้แต่ค่าใดค่าหนึ่งจะส่งรหัสตอบกลับไปยังอุปกรณ์และไม่ตรวจสอบหรือทำการเก็บข้อมูลต่อไปจนกว่าจะมีการส่งข้อมูลครั้งใหม่เข้ามา แต่ถ้าข้อมูลครบถ้วนจะนำข้อมูลไปเก็บในตัวแปร Current\_transactions ดังที่กล่าวมาข้างต้น และเริ่มการตรวจสอบความถูกต้องของข้อมูลก่อนหน้าที่เก็บอยู่ใน Chain ว่าข้อมูลถูกต้องและไม่ถูกเปลี่ยนแปลงข้อมูล โดยการนำ Hash\_of\_previous\_block ของแต่ละบล็อกมาเทียบกับข้อมูลจริงที่เข้ารหัสของบล็อกก่อนหน้าและคำนวณค่า nonce ของแต่ละบล็อกที่บันทึกไว้ว่าตรงกับที่ถูกต้องหรือไม่เพื่อเป็นการยืนยันว่าบล็อกเชนไหนนั้นยังไม่ถูกโจมตีหรือถูกเปลี่ยนแปลงข้อมูล หลังจากข้อมูลถูกบันทึกลงเครือข่ายบล็อกเชนแล้วจะถูกส่งต่อไปบันทึกที่ฐานข้อมูล MySQL โดยอัตโนมัติกระบวนการทำงานภายในบล็อกเชนดังในรูปที่ 3.3

### 3.7 การเชื่อมโยงส่วนการทำงานของระบบ



รูปที่ 3.4 แผนภาพแสดงการเชื่อมโยงส่วนการทำงานทั้ง 3 ส่วน

จากรูปที่ 3.4 เป็นแผนภาพแสดงการเชื่อมโยงส่วนการทำงานที่แยกกันในระบบ โดยเริ่มจากอุปกรณ์ IoT ที่ติดตั้งไว้กับรถขนส่งสินค้าแต่ละคัน จะส่งข้อมูลสภาพแวดล้อมที่วัดค่าได้ไปยังเครื่องแม่ข่ายที่ติดตั้งเครือข่ายบล็อกเชนและฐานข้อมูล MySQL ซึ่งเครื่องแม่ข่ายนี้จะทำหน้าที่เป็นเสมือนกลางระหว่างข้อมูลที่ได้รับจากอุปกรณ์เซนเซอร์และอุปกรณ์เคลื่อนที่ที่ผู้ใช้สามารถตรวจสอบได้ จากนั้นในส่วนของ Web Application จะดึงรายละเอียดต่างๆจากฐานข้อมูลมาแสดงได้แบบเรียลไทม์ ทั้งนี้เมื่อมีการแก้ไขข้อมูลจากผู้ใช้งานผ่านทาง Web Application ข้อมูลก็จะถูกบันทึกลงไปในฐานข้อมูล และอุปกรณ์ IoT ก็จะได้รับข้อมูลนั้นได้แบบเรียลไทม์เช่นกัน ส่วนการทำงานของระบบสามารถแจกแจงรายละเอียดได้ดังนี้

### 3.7.1 รถขนส่งสินค้า

ในรถขนส่งสินค้าแต่ละคัน จะติดตั้งอุปกรณ์ Raspberry PI 3 ที่เชื่อมต่อกับเซนเซอร์วัดอุณหภูมิจากภายนอกและภายในของรถ และเมื่อเริ่มการจัดส่งสินค้าจะมีการตั้งค่าหมายเลขรหัสอุปกรณ์ของ Raspberry PI ดังโค้ดตัวอย่างด้านล่าง

```
function get_serial() {
    serial = หมายเลขรหัสเริ่มต้น เพื่อกรณีที่ไม่สามารถดึงรหัสอุปกรณ์ได้
    f = อ่านข้อมูลของอุปกรณ์จาก /proc/cpuinfo
    for line in f:
        if line[0:6] == 'Serial':
            serial = line[10:26]
    return serial
}
```

เริ่มจากตั้งค่าเริ่มต้นให้กับตัวแปร serial และอ่านค่าจากหน่วยประมวลผลของอุปกรณ์ Raspberry PI เพื่อค้นหาค่า Serial ที่เป็นหมายเลขรหัสอุปกรณ์ 16 หลัก และส่งค่ากลับไปยังเครื่องแม่ข่ายเพื่อคำนวณและบันทึกข้อมูลต่อไป ซึ่งหมายเลขอุปกรณ์นี้จำเป็นอย่างมากในการระบุว่าค่าถูกส่งมาจากอุปกรณ์ตัวไหนได้อย่างถูกต้อง หลังจากนั้น Raspberry PI จะตรวจสอบจากฐานข้อมูลว่ามีการขนส่งที่เลือกรถขนส่งสินค้าคันปัจจุบันและอยู่ในสถานะ ‘กำลังจัดส่ง’ หรือไม่ และเก็บค่า id ของตาราง shipping เพื่อใช้ในการส่งข้อมูลทั้งอุณหภูมิจากภายนอกและภายในของรถ และพิกัดปัจจุบันจากเซนเซอร์ไปยังเครื่องแม่ข่ายตามระยะเวลาที่กำหนดไว้ ทั้งนี้ ในระหว่างการบันทึกข้อมูลจะมีการตรวจสอบว่าสินค้าทุกชนิดในรถขนส่งสินค้าอยู่ในอุณหภูมิที่เหมาะสม โดยเปรียบเทียบตามเกณฑ์อุณหภูมิสูงสุด (max\_temp) และอุณหภูมิต่ำสุด (min\_temp) ในตาราง product และพิกัดต้องไม่อยู่จุดเดิมเป็นเวลานานเกินไป หากมีเงื่อนไขใดเงื่อนไขหนึ่งที่ไม่ถูกต้อง ระบบจะส่งข้อความแจ้งเตือนไปยัง LINE ของพนักงานขนส่งที่รับผิดชอบการส่ง ณ ขณะนั้น ดังตัวอย่างด้านล่าง

```

function auto_run(){
    เชื่อมต่อกับฐานข้อมูล

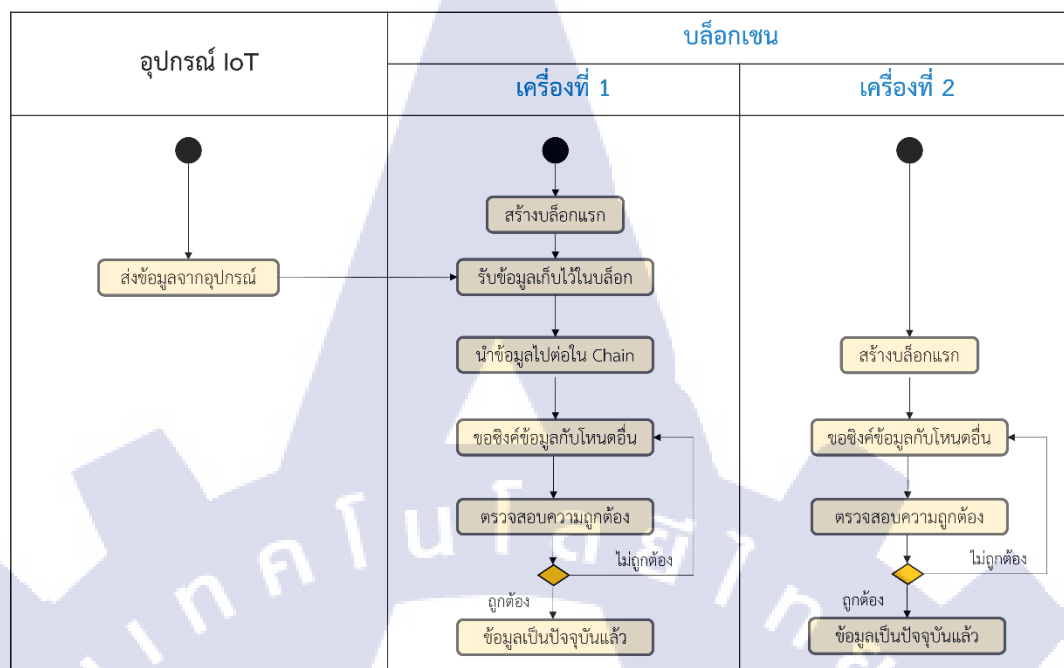
    shipping_id = หากการจัดส่งจากฐานข้อมูลที่ใช้หมายเลขอุปกรณ์เดียวกับอุปกรณ์นี้และมี
    สถานะ กำลังจัดส่ง
    if หา shipping_id ของการจัดส่งได้{
        อ่านข้อมูลจากอุปกรณ์
        location_status = เช็คเงื่อนไขของพิกัด และส่งค่ากลับมาเป็น True หรือ
        False
        temperature_status = เช็คเงื่อนไขของอุณหภูมิ และส่งค่ากลับมาเป็น True
        หรือ False
        if location_status == False {
            ส่งข้อความแจ้งเตือนเรื่องพิกัด
        }
        if temperature_status == False {
            ส่งข้อความแจ้งเตือนเรื่องอุณหภูมิ
        }
        ส่งข้อมูลจากอุปกรณ์ และรหัสการจัดส่งจาก shipping_id ไปยังบล็อกเซน
    }
    ปิดการเชื่อมต่อกับฐานข้อมูล
}

```

โค้ดดังกล่าวจะเริ่มทำงานโดยอัตโนมัติเมื่อเปิดใช้งาน Raspberry PI ที่ติดตั้งโค้ดไว้ และจะตรวจสอบเงื่อนไขรวมถึงส่งข้อมูลไปยังเครื่องแม่ข่ายทุกๆระยะเวลาที่กำหนด หากในเวลานั้นไม่มีการจัดส่งที่อยู่ในสถานะ กำลังจัดส่ง จะไม่มีการประมวลผลขึ้นต่อไปอีก แต่จะยังคงเปิดการทำงานไปเรื่อย ๆ จนกว่าจะสร้างการจัดส่งครั้งใหม่ขึ้นมา

### 3.7.2 เครื่องแม่ข่าย

เครื่องแม่ข่ายเป็นคอมพิวเตอร์ระบบปฏิบัติการ Ubuntu Server และรันโปรแกรมผ่าน Flask ซึ่งเป็นเฟรมเวิร์กสำหรับภาษาไพทอนที่เขียนขึ้นมาเพื่อใช้ร่วมกับ Webserver เช่น Apache เป็นต้น ทำให้ไพทอนสามารถแสดงผลหน้า Web Application หรือ การรับส่งค่าผ่าน API ได้ ซึ่งภายในเครื่องแม่ข่ายจะเก็บทั้งฐานข้อมูล MySQL และเปิดใช้งานบล็อกเซนด้วย



รูปที่ 3.5 ตัวอย่างขั้นตอนการทำงานของบล็อกเชนเมื่อติดตั้งและรันผ่านเครื่องแม่ข่าย

จากรูปที่ 3.5 เมื่อเครื่องแม่ข่ายเริ่มทำงาน จะเป็นการเปิดใช้งานบล็อกเชนและสร้างบล็อกแรกหรือที่เรียกว่า Genesis Block ขึ้นมาเป็นบล็อกที่ยังไม่มีข้อมูลอะไร เมื่อมีการส่งข้อมูลมาจากอุปกรณ์ IoT มายังบล็อกเชนโหนดใดโหนดหนึ่ง ข้อมูลจะถูกเก็บไว้ในบล็อกและเพิ่มเข้าไปในเรียงต่อในเชนเพื่อให้ข้อมูลมีความเชื่อมต่อกัน โดยในแต่ละบล็อกนอกจากข้อมูลปัจจุบันที่เก็บเข้ามาแล้ว จะมีการเก็บข้อมูลที่เข้ารหัสแล้วของบล็อกก่อนหน้านี้ด้วย จากนั้นโหนดจากเครื่องแม่ข่ายเครื่องอื่นที่มีข้อมูลน้อยกว่าจะใช้คำสั่งเพื่อขอชิงค์ข้อมูลด้วยการเปรียบเทียบหาโหนดที่มีจำนวนบล็อกมากที่สุด และทุกบล็อกต้องมีข้อมูลถูกต้องเมื่อได้ข้อมูลแล้ว โหนดดังกล่าวจะอัปเดตข้อมูลให้เป็นปัจจุบัน และในการตรวจสอบความถูกต้องของข้อมูลในบล็อกเชนทำได้โดยการเปรียบเทียบข้อมูลเข้ารหัสของบล็อกก่อนหน้านี้ที่บันทึกไว้ กับ ข้อมูลเข้ารหัสของบล็อกก่อนหน้านี้ที่เกิดจากการเข้ารหัสจริงที่ละบล็อก หากข้อมูลทั้งสองส่วนไม่ตรงกันจะถือว่าบล็อกเชนสายนั้นเป็นโมฆะทำให้ไม่สามารถบันทึกข้อมูลต่อไปได้ จนกว่าจะขอชิงค์กับโหนดอื่นเพื่อให้ข้อมูลในบล็อกเชนโหนดนั้นกลับมาถูกต้อง

### 3.7.3 Web Application

ทำงานด้วย Vue.js ซึ่งเป็นเฟรมเวิร์กของภาษา JavaScript ที่สามารถสร้าง Web Application เป็นแบบ SPA (Single Page Application) ได้ และมีการติดตั้ง LINE Front-end Framework (LIFF) เพื่อให้เชื่อมต่อข้อมูลผู้ใช้งานผ่านแอปพลิเคชัน LINE ได้ ดังตัวอย่างรูปที่ 3.6

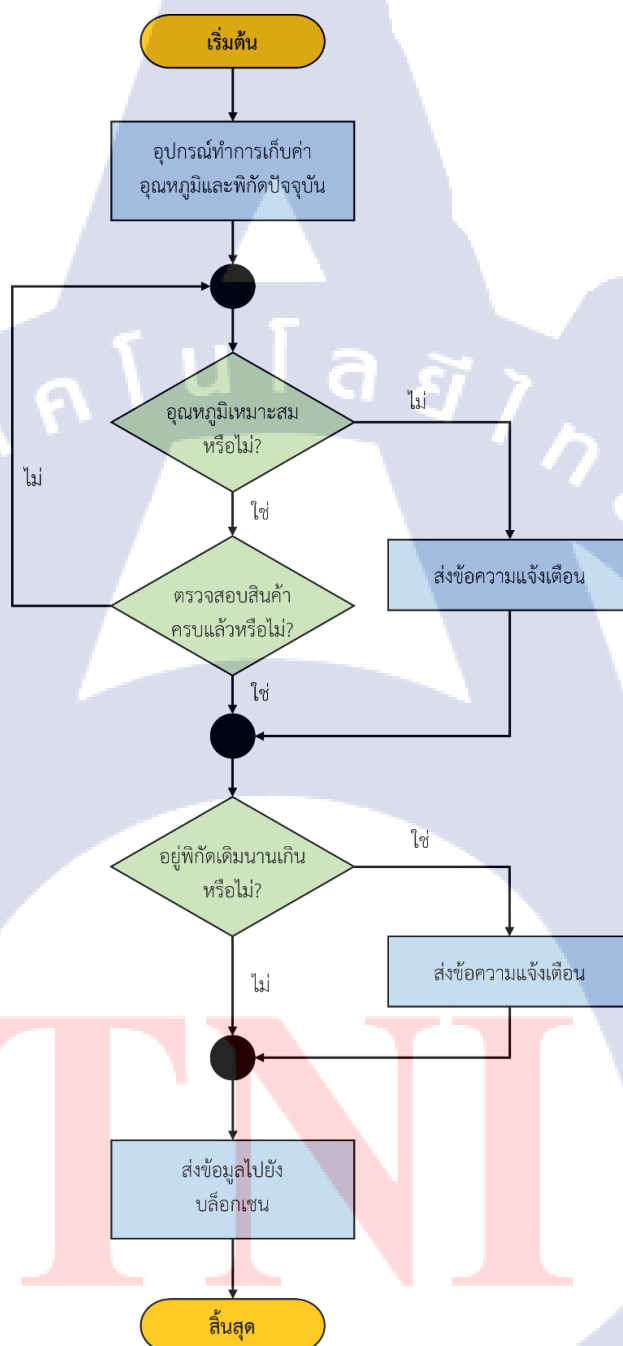


รูปที่ 3.6 ค่า User LINE Id ที่ได้หลังจากผู้ใช้งานล็อกอินเข้าสู่ระบบ LINE

User LINE Id คือรหัสประจำตัวผู้ใช้แต่ละคนสำหรับใช้เพื่อสื่อสารกับหลังบ้านผ่าน LINE API โดยจะเป็นรหัสขึ้นต้นด้วยตัว U และมีตัวอักษรหรือตัวเลขตามหลังอีก 32 ตัว โดยนอกจาก id ของผู้ใช้แต่ละคนยังจะไม่ซ้ำกันทำให้ง่ายต่อการระบุตัวตนแล้ว มันยังจำเป็นอย่างมากเพื่อใช้สำหรับการส่งข้อความหรือการแจ้งเตือนผ่านแชทบอทไปยังผู้ใช้งานซึ่งจะใช้ร่วมกับ Channel Access Token ที่เปรียบเสมือนรหัสสำหรับการเข้าใช้งาน LINE API โดยในงานวิจัยนี้ได้ใช้งาน library ชื่อ LINE Messaging API SDK [35] ของภาษาไพทอนที่ติดตั้งไว้ในอุปกรณ์ IoT เพื่อให้อุปกรณ์สามารถส่งข้อความแจ้งเตือนไปยังพนักงานขนส่งได้

จากรูปที่ 3.7 ก่อนที่จะส่งข้อมูลไปยังเครือข่ายบลูทูธ อุปกรณ์จะมีการตรวจสอบเงื่อนไขของอุณหภูมิและพิกัดว่าเหมาะสมหรือไม่ โดยจะเป็นการเปรียบเทียบค่าอุณหภูมิภายในรถที่วัดได้จากอุปกรณ์กับอุณหภูมิต่ำสุดและอุณหภูมิสูงสุดที่เหมาะสมของสินค้าทุกชิ้นที่บันทึกในฐานข้อมูลเมื่อผู้ใช้สร้างข้อมูลการจัดส่งแต่ละครั้ง หากอุณหภูมิไม่เหมาะสมกับสินค้าชิ้นใดชิ้นหนึ่งในขณะจัดส่งจะทำการเพิ่มจำนวนครั้งที่เงื่อนไขผิดพลาดไว้กับตัวแปรที่ตั้งค่าไว้และเมื่อตัวแปรนั้นมีจำนวนครั้งมากกว่าหรือเท่ากับค่าที่กำหนด ยกตัวอย่างเช่น อุณหภูมิไม่ตรงเงื่อนไขเป็นเวลา 3 ครั้ง อุปกรณ์จะส่งข้อความแจ้งเตือนไปยังพนักงานขนส่ง เช่นเดียวกันกับการตรวจสอบเงื่อนไขพิกัด

ปัจจุบันของรถขนส่งสินค้าที่หากรถอยู่พักติดเป็นเวลามากกว่าหรือเท่ากับจำนวนครั้งที่กำหนด อุปกรณ์ก็จะทำการส่งข้อความแจ้งเตือนไปยังพนักงานขนส่ง



รูปที่ 3.7 กระบวนการตรวจสอบเงื่อนไขและส่งข้อความแจ้งเตือน

### 3.8 การวัดประสิทธิภาพของระบบ

ในการวัดประสิทธิภาพของระบบรับ-ส่งข้อมูลระหว่างอุปกรณ์และเครือข่ายบล็อกเชนมีเกณฑ์ในการวัดคือ

3.8.1 เวลาที่ใช้ตั้งแต่เก็บข้อมูลจากอุปกรณ์จนกระทั่งข้อมูลถูกบันทึกในเครือข่ายบล็อกเชนและฐานข้อมูล

3.8.2 หน่วยความจำที่ใช้ระหว่างการเก็บและส่งข้อมูล  
โดยในการวัดประสิทธิภาพมีขั้นตอนดังตัวอย่างด้านล่าง

```
function ส่งข้อมูลไปยังบล็อกเชน(int shipping_id):
    start_time = เวลาที่เริ่มรันคำสั่ง
    url = api ของบล็อกเชนที่ใช้รับข้อมูล
    data = ข้อมูลอุณหภูมิและพิกัดที่เก็บได้จากอุปกรณ์
    ระบุรหัสการจัดส่งที่จะใช้ส่งใน data ด้วย shipping_id

    try:
        response = ส่ง data ไปยัง url และรอฝั่งเครื่องแม่ข่ายส่งผลสำเร็จกลับมา
        memory_usage = อ่านค่าหน่วยความจำที่ถูกใช้งานในการส่งข้อมูล
        using_time = เวลาปัจจุบัน - start_time
        บันทึกค่า memory_usage ในไฟล์ text
        บันทึกค่า using_time ในไฟล์ text
        print("ส่งข้อมูลสำเร็จ")
    except:
        print("ไม่สามารถส่งข้อมูลได้")
```

## บทที่ 4

### ผลการวิจัย

จากกรอบแนวคิดและการออกแบบวิธีวิจัยในบทที่ 3 ผู้วิจัยได้ดำเนินการพัฒนาระบบตรวจสอบการขนส่งห่วงโซ่ความเย็นสำหรับอาหารด้วยเทคโนโลยีบล็อกเชนและอินเทอร์เน็ตของสรรพสิ่ง และนำผลการวิจัยมาแสดง ดังต่อไปนี้

- 4.1 ผลการทำงานของระบบจากการพัฒนาเครือข่ายบล็อกเชน
- 4.2 ผลการพัฒนา Web Application
- 4.3 ผลการวัดประสิทธิภาพของระบบ

#### 4.1 ผลการทำงานของระบบจากการพัฒนาเครือข่ายบล็อกเชน

ในการพัฒนาระบบเพื่อใช้ในการตรวจสอบการขนส่งห่วงโซ่ความเย็นสำหรับอาหารด้วยเทคโนโลยีบล็อกเชน ผู้วิจัยเสนอวิธีการพัฒนาระบบเครือข่ายบล็อกเชนโดยใช้ภาษาไพธอนในการเขียนโปรแกรมและทำการรันโค้ดไพทอนบนเครื่องแม่ข่ายโดยใช้ library Flask Python ในการกระจาย API เป็นหลัก ทำให้อุปกรณ์จากเครือข่ายภายนอกสามารถส่งข้อมูลเข้ามาบันทึกในเครือข่ายบล็อกเชนได้ ซึ่งสามารถใช้งานได้ดังแสดงตัวอย่างในรูปที่ 4.1

```
"chain": [
  {
    "hash_of_previous_block": "181cfa3e85f3c2a7aa9fb74f992d0d061d3e4a6d7461792413aab3f97bd3da95",
    "index": 0,
    "nonce": 61093,
    "timestamp": 1660325795.1982615,
    "transaction": []
  },
  {
    "hash_of_previous_block": "357c4825eb075e9f9f3b00f36c19ce4da77cd719dd91f7e53186fdb3dc78d2e3",
    "index": 1,
    "nonce": 30267,
    "timestamp": 1661366268.6751916,
    "transaction": [
      {
        "inside_temp": -6,
        "latitude": 13.7373131,
        "longitude": 100.6246604,
        "outside_temp": -9,
        "shipping_id": 2
      }
    ]
  }
],
"length": 1
```

①

---

②

รูปที่ 4.1 ตัวอย่างข้อมูลในบล็อกเชนที่เรียกออกมาผ่าน API

ส่วนที่ 1 คือข้อมูลเริ่มต้นเมื่อรับบล็อกเชนบนเครื่องแม่ข่ายที่เรียกว่า Genesis Block ซึ่งจะพบว่าในส่วน transaction จะยังไม่มีข้อมูลอะไร และมี index ของบล็อกเริ่มต้นที่ 0

ส่วนที่ 2 เมื่ออุปกรณ์ส่งข้อมูลที่วัดค่าได้มายังเครือข่ายบล็อกเชนและตัวบล็อกได้รับการยืนยันความถูกต้องแล้ว ข้อมูลจะถูกเพิ่มเข้ามา โดยส่วนที่ได้จากอุปกรณ์จะอยู่ในฟิลด์ที่ชื่อว่า transaction ประกอบด้วย อุณหภูมิภายในรถขนส่งสินค้า อุณหภูมิภายนอกขนส่งสินค้า ละติจูด ลองจิจูด และหมายเลข id ของการจัดส่ง และมีฟิลด์ hash\_of\_previous\_block ที่เก็บข้อมูลที่ถูกรับแล้วของบล็อกก่อนหน้ารวมถึงฟิลด์ timestamp ที่เก็บเวลาในการบันทึกปรากฏอยู่ในทุกบล็อกอีกด้วย

```

GET blockchain-api.depa.or.th/blockchain
Status: 200 OK Size: 715 Bytes Time: 288 ms
Response:
{
  "transaction": {
    "inside_temp": -6,
    "latitude": 13.7373131,
    "longitude": 100.6246604,
    "outside_temp": -9,
    "shipping_id": 2
  },
  "hash_of_previous_block": "4a8c1bd34809b39c6afbdffcd882f0922c861d496bcb1ad4e9a0a379fec7e1dd9",
  "index": 2,
  "nonce": 8171,
  "timestamp": 1661366278.570287,
  "transaction": [
    {
      "inside_temp": -9,
      "latitude": 13.7373131,
      "longitude": 100.6246604,
      "outside_temp": -6,
      "shipping_id": 2
    }
  ],
  "length": 3
}

GET http://209.15.96.110:6000/blockchain
Status: 200 OK Size: 189 Bytes Time: 22 ms
Response:
{
  "chain": [
    {
      "hash_of_previous_block": "181cfa3e85f3c2a7aa9fb74f992d0d061d3e4a6d7461792413aab3f97bd3da95",
      "index": 0,
      "nonce": 61093,
      "timestamp": 1661369985.0304964,
      "transaction": [
        ]
      },
    {
      "length": 1
    }
  ]
}

```

รูปที่ 4.2 บล็อกเชนที่รันจากเครื่องแม่ข่ายคนละตัว

อีกคุณสมบัติของบล็อกเชนคือการที่ไม่จำเป็นต้องมีคอมพิวเตอร์ศูนย์กลางในการเก็บข้อมูลเพียงเครื่องเดียว แต่เครื่องแม่ข่ายอื่น ๆ ที่ร่วมใช้เครือข่ายบล็อกเชนเดียวกันก็จะได้อีกข้อมูลชุดเดียวกันไปด้วย โดยในงานวิจัยนี้ผู้วิจัยออกแบบระบบให้เมื่อมีโหนดใหม่เพิ่มเข้ามาในเครือข่าย โหนดดังกล่าวจะสามารถใช้คำสั่งซิงค์ เพื่อขอดึงข้อมูลเชนที่มีอยู่ในปัจจุบันไปเก็บไว้ที่โหนดของตนได้ และ

เมื่อมีข้อมูลจากอุปกรณ์ถูกส่งไปที่โหนดใดจะถูกตั้งค่าให้ใช้คำสั่งซิงค์ก่อนโดยอัตโนมัติ ทั้งนี้เพื่อให้เป็นข้อมูลที่ใหม่ที่สุดก่อนจะทำการบันทึกข้อมูลและตรวจสอบข้อมูลให้มีความถูกต้องอยู่เสมออีกด้วย

ในรูปที่ 4.2 คือตัวอย่างการรับบล็อกเชนจากเครื่องแม่ข่ายคนละตัว และยังไม่ได้ทำการเชื่อมต่อทั้งสองเข้าด้วยกันทำให้ข้อมูลในปัจจุบันจะเป็นข้อมูลที่แตกต่างกันสังเกตได้จากความยาวของบล็อกเชนที่ไม่เท่ากัน หากผู้ใช้ต้องการให้ทั้งสองโหนดมีข้อมูลเดียวกันสามารถทำได้โดยใช้คำสั่งเพิ่มโหนด (add\_nodes) แล้วป้อนไอพีแอดเดรสหรือโดเมนของโหนดที่ต้องการให้ซิงค์ข้อมูลด้วยลงไป จากนั้นใช้คำสั่งซิงค์ดังแสดงในรูปที่ 4.3 ซึ่งโดยปกติแล้วเมื่อรับบล็อกเชนในเครื่องแม่ข่ายตัวบล็อกเชนจะใช้คำสั่งซิงค์โดยอัตโนมัติตามระยะเวลาที่ตั้งค่าไว้ แต่การเพิ่มโหนดจะต้องทำด้วยตัวเองเพื่อให้แน่ใจว่าต้องการเชื่อมต่อข้อมูลจากโหนดดังกล่าวจริงและเป็นการป้องกันการโจมตีจากผู้ไม่ประสงค์ดี

The screenshot displays two REST client panels. The left panel shows a POST request to `http://209.15.96.110:6000/nodes/add_nodes` with a JSON body containing a list of nodes. The right panel shows a GET request to `http://209.15.96.110:6000/nodes/sync` with a JSON response. The response is highlighted with a red box.

**POST Request:**

```
POST http://209.15.96.110:6000/nodes/add_nodes
{
  "nodes": [
    "https://blockchain-api.depa.or.th"
  ]
}
```

**GET Request:**

```
GET http://209.15.96.110:6000/nodes/sync
```

**Response:**

```
{
  "inside_temp": -6,
  "latitude": 13.7373131,
  "longitude": 100.6246684,
  "outside_temp": -9,
  "shipping_id": 2,
  "hash_of_previous_block": "4a8c1bd34809b39c6afbdffcd882f0922c861d496bc1ad4e9a0a379fec7e1dd9",
  "index": 2,
  "nonce": 8171,
  "timestamp": 1661366278.570287,
  "transaction": [
    {
      "inside_temp": -9,
      "latitude": 13.7373131,
      "longitude": 100.6246684,
      "outside_temp": -6,
      "shipping_id": 2
    }
  ],
  "message": "Blockchain has been updated with new data"
}
```

รูปที่ 4.3 การเพิ่มโหนด (ซ้าย) และการใช้คำสั่งซิงค์เพื่อดึงข้อมูลจากโหนดที่มีข้อมูลมากกว่า (ขวา)

## 4.2 ผลการพัฒนา Web Application

ผู้ใช้งานสามารถเข้าใช้งาน Web Application ได้ผ่านทาง LINE Official Account ที่สร้างไว้ โดยเป็นการเข้าผ่าน LIFF และต้องเข้าสู่ระบบด้วยบัญชี LINE ของผู้ใช้ หลังจากที่ใช้กดเข้าสู่ระบบแล้วจะดึงค่า User LINE ID เพื่อตรวจสอบว่ามีอยู่ในฐานข้อมูลหรือไม่ หากมีแล้วจะดึงสิทธิ์ของผู้ใช้มาได้ แต่หากไม่มี ผู้ใช้จะต้องสมัครสมาชิกใหม่โดยมีค่าที่ต้องป้อนได้แก่ ชื่อ นามสกุล รหัสประชาชน และเบอร์โทรศัพท์ จากนั้นกดยืนยันเพื่อสมัครสมาชิกในระบบ ดังรูปที่ 4.4 ซึ่งสมาชิกที่สมัครเข้ามาใหม่จะได้รับสิทธิ์ ผู้ใช้งานทั่วไปเท่านั้น จนกว่าสมาชิกที่มีสิทธิ์แอดมินจะปรับให้เป็นสิทธิ์อื่น

**สมัครสมาชิก**

ชื่อ

นามสกุล

รหัสประชาชน

เบอร์โทรศัพท์

**ยืนยัน**

รูปที่ 4.4 แบบฟอร์มสมัครสมาชิกเมื่อผู้ใช้งานเข้าสู่ระบบครั้งแรก

#### 4.2.1 หน้าจอหลัก



รูปที่ 4.5 หน้าจอหลักของ Web Application แยกตามประเภทผู้ใช้งาน

ตัวอย่างรูปที่ 4.5 แสดงหน้าจอหลักหลังจากที่ผู้ใช้กดเข้าสู่ระบบมาแล้ว หน้าจอหลักจะแสดงข้อมูลการจัดส่งแต่ละครั้ง โดยมีสีแตกต่างกันขึ้นอยู่กับสถานะของการจัดส่ง คือ

**สีน้ำเงิน** หมายถึง การจัดส่งที่ถูกสร้างขึ้นมา แต่ยังไม่เริ่มการขนส่งและยังไม่มีบันทึกข้อมูลอุณหภูมิในระบบ

**สีเหลือง** หมายถึง การจัดส่งที่กำลังอยู่ในการขนส่ง เป็นสถานะที่อุปกรณ์ IoT สามารถส่งข้อมูลมายังเครือข่ายบล็อกเชนและฐานข้อมูลได้

**สีเขียว** หมายถึง การจัดส่งที่จัดส่งสำเร็จหรือสินค้าถึงที่หมายปลายทางแล้ว และผู้ใช้สามารถบันทึกข้อมูลอุณหภูมิของการจัดส่งนั้นแบบย้อนหลังได้

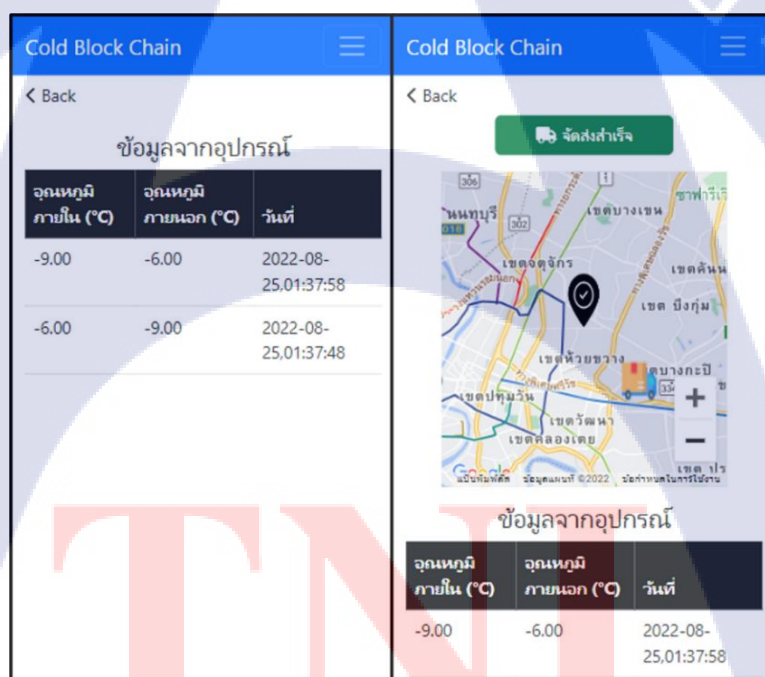
นอกจากนี้ ส่วนของหน้าจอหลักรวมถึงแถบเมนูด้านบนของผู้ใช้ยังมีความแตกต่างกันแยกตามสิทธิ์การใช้งานของผู้ใช้ ดังนี้

(ซ้าย) : หน้าจอหลักสำหรับผู้ใช้ที่มีสิทธิ์ ผู้ใช้งานทั่วไป จะแสดงข้อมูลการจัดส่งแต่ละครั้ง โดยมีรายละเอียดคือ จุดเริ่มต้น (ละติจูด ลองจิจูด) ปลายทาง (ละติจูด ลองจิจูด) ชื่อพนักงานขนส่ง และหมายเลขทะเบียนรถ และมีปุ่ม Detail เพื่อเข้าไปดูรายละเอียดของการขนส่งนั้น

(กลาง) : หน้าจอหลักสำหรับผู้ใช้ที่มีสิทธิ์ พนักงานขนส่ง จะแสดงข้อมูลการจัดส่งเช่นเดียวกับผู้ใช้ที่มีสิทธิ์ ผู้ใช้ทั่วไป แต่มีจุดแตกต่างคือจะมีปุ่ม เริ่มการขนส่งครั้งใหม่ (+ New Shipment) ที่เชื่อมไปยังหน้าจอสร้างการขนส่ง

(ขวา) : หน้าจอหลักสำหรับผู้ใช้ที่มีสิทธิ์ แอดมิน จะแสดงข้อมูลการจัดส่งเช่นเดียวกับผู้ใช้ที่มีสิทธิ์ ผู้ใช้ทั่วไป แต่มีจุดแตกต่างคือแถบเมนูด้านบนจะเพิ่มปุ่มที่เชื่อมไปยังส่วนการจัดการข้อมูลต่าง ๆ ได้แก่ การจัดการข้อมูลผู้ใช้ การจัดการข้อมูลสินค้า และการจัดการข้อมูลอุปกรณ์

#### 4.2.2 หน้าแสดงรายละเอียดการจัดส่ง



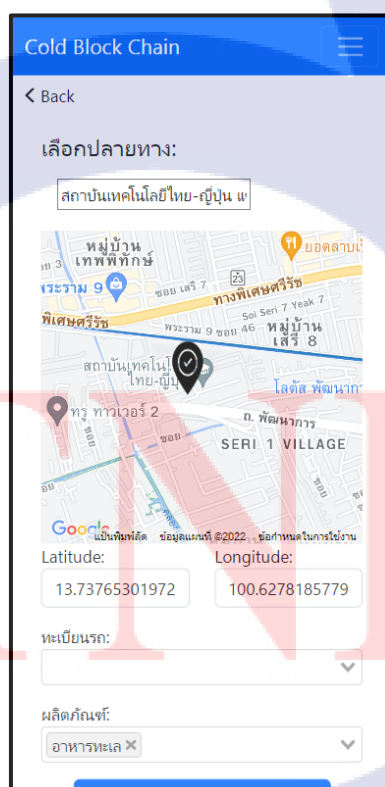
รูปที่ 4.6 หน้าจอแสดงรายละเอียดการจัดส่ง แยกตามประเภทผู้ใช้งาน

หน้าแสดงรายละเอียดการจัดส่งดังแสดงในรูปที่ 4.6 จะแบ่งเป็น 2 แบบ คือ สำหรับผู้ใช้งานทั่วไป (ซ้าย) สำหรับพนักงานขนส่งและแอดมิน (ขวา) โดยมีข้อแตกต่างกันคือ ผู้ใช้งานทั่วไป จะเห็นข้อมูลอุณหภูมิภายในและอุณหภูมิภายนอกของรถสินค้าพร้อมทั้งเวลาที่ข้อมูลถูกส่งมาจากอุปกรณ์ ส่วนพนักงานขนส่งและแอดมินจะเห็นภาพแสดงแผนที่ ที่ได้จากการใช้งาน Google Maps

Platform เพิ่มเข้ามาด้วย แผนที่แสดงค่าละติจูดและลองจิจูดของสถานที่ปลายทางที่พนักงานขนส่งตั้งไว้แล้วแสดงออกมาเป็นสัญลักษณ์เครื่องหมายลูกศรดำเพื่อบอกพิกัดที่พนักงานขนส่งต้องไป ค่าละติจูดและลองจิจูดล่าสุดที่ส่งมาจากอุปกรณ์จะแสดงออกมาในรูปรถขนส่งสินค้าเพื่อบอกว่า ณ เวลาปัจจุบัน รถขนส่งสินค้าอยู่ที่ตำแหน่งใดแล้ว นอกจากนี้ผู้ใช้ที่เป็นพนักงานขนส่งที่รับผิดชอบดูแลการจัดส่งครั้งนั้นจะเห็นปุ่มสำหรับเปลี่ยนสถานะการจัดส่งเพิ่มขึ้น โดยการจัดส่งที่มีสถานะยังไม่จัดส่งจะเป็นปุ่มเพื่อเปลี่ยนเป็นสถานะกำลังจัดส่ง และการจัดส่งที่มีสถานะกำลังจัดส่งจะเป็นปุ่มเพื่อเปลี่ยนเป็นสถานะจัดส่งสำเร็จ

การแสดงผลในหน้ารายละเอียดการจัดส่งยังมีความแตกต่างกันเล็กน้อยตามสถานะการจัดส่ง คือ สถานะยังไม่จัดส่งจะไม่มีตารางบันทึกข้อมูลอุณหภูมิ ส่วนสถานะกำลังจัดส่งจะไม่แสดงพิกัดล่าสุดที่เป็นสัญลักษณ์รูปรถขนส่งสินค้าบนแผนที่แต่จะแสดงเพียงพิกัดของสถานที่ปลายทางเพียงอย่างเดียว

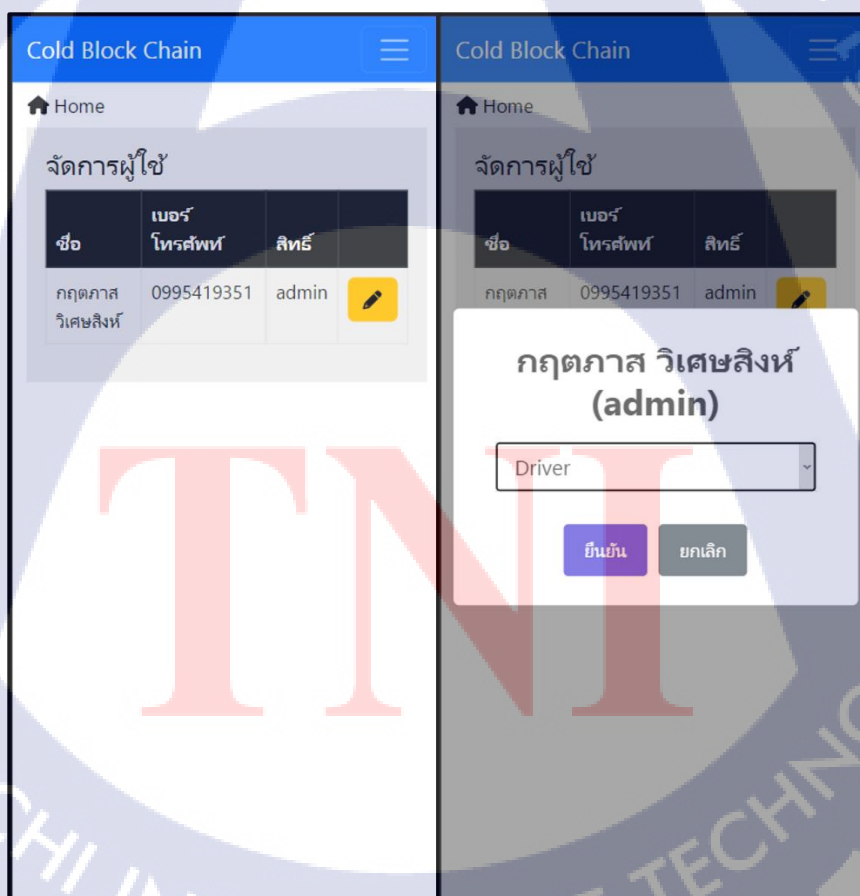
#### 4.2.3 หน้าสร้างการจัดส่งครั้งใหม่



รูปที่ 4.7 หน้าจอเมื่อกดปุ่มสร้างการจัดส่งครั้งใหม่โดยพนักงานขนส่ง

หน้าจอสร้างการจัดส่งครั้งใหม่ดังแสดงในรูปที่ 4.7 ต้องเป็นผู้ใช้งานสิทธิ์พนักงานขนส่งเท่านั้นจึงจะสามารถใช้งานได้ โดยผู้ใช้งานสามารถเลือกสถานที่ปลายทางในการจัดส่งสินค้าด้วยการเลื่อนและคลิกจุดที่ต้องการบนแผนที่หรือสามารถค้นหาด้วยชื่อสถานที่นั้นในกล่องค้นหาด้านบนก็ได้เช่นกัน เมื่อพิมพ์ชื่อสถานที่และกดเลือกอันที่ต้องการแล้วจะเป็นการเปลี่ยนจุดกึ่งกลางของแผนที่เป็นพิกัดของสถานที่ที่เลือกไปและให้ผู้ใช้งานคลิกแผนที่เพื่อเลือกสถานที่ปลายทางได้ ทุกครั้งที่ผู้ใช้คลิกบนแผนที่จะเป็นการเปลี่ยนค่าละติจูดและลองจิจูดตามที่แสดงในรูปเพื่อใช้บันทึกบนบล็อกเชนและฐานข้อมูล ผู้ใช้งานจะต้องเลือกทะเบียนรถคันที่จะใช้จัดส่งซึ่งจะเป็นทะเบียนรถที่อยู่ในสถานะพร้อมใช้งานจากฐานข้อมูลเท่านั้น หลังจากนั้นเป็นการเลือกสินค้าที่ต้องจัดส่งโดยสามารถเลือกสินค้าได้หลายชนิด และเมื่อกดปุ่มยืนยันการสร้างการจัดส่งครั้งใหม่แล้วหน้าจอ Web Application ของผู้ใช้งานจะเปลี่ยนหน้าเป็นหน้าจอรายละเอียดการจัดส่งเพื่อให้ผู้ใช้กดเริ่มต้นการจัดส่งได้โดยอัตโนมัติ

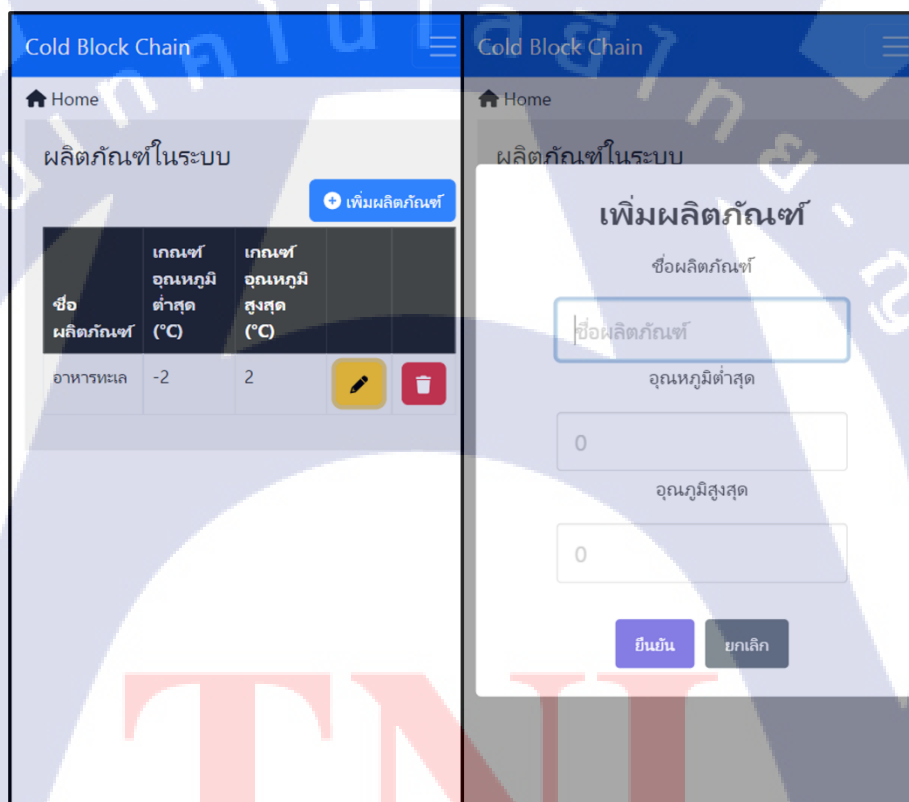
#### 4.2.4 หน้าจัดการผู้ใช้



รูปที่ 4.8 หน้าจัดการผู้ใช้แสดงรายชื่อผู้ใช้ทั้งหมดและการแก้ไขสิทธิ์ของผู้ใช้

**รูปที่ 4.8** แสดงถึงหน้าการจัดการผู้ใช้ โดยผู้ใช้งานที่มีสิทธิ์แอดมินเท่านั้นที่สามารถเข้าสู่หน้านี้ได้ หน้าจอจะแสดงรายชื่อผู้ใช้งานทั้งหมดที่ลงทะเบียนในระบบ เบอร์โทรศัพท์ของสมาชิกในกรณีที่เกิดปัญหาระหว่างการขนส่งเพื่อให้แอดมินสามารถติดต่อกับสมาชิกได้ และแสดงสิทธิ์การใช้งานของสมาชิกแต่ละคนด้วย เมื่อแอดมินกดปุ่มสี่เหลี่ยมที่มีสัญลักษณ์รูปดินสอจะแสดงหน้าต่างบอกชื่อและสิทธิ์ปัจจุบันของสมาชิกคนนั้นและสามารถปรับเปลี่ยนสิทธิ์ของสมาชิกผ่าน Dropdown List ด้านล่าง

#### 4.2.5 หน้าจัดการสินค้า

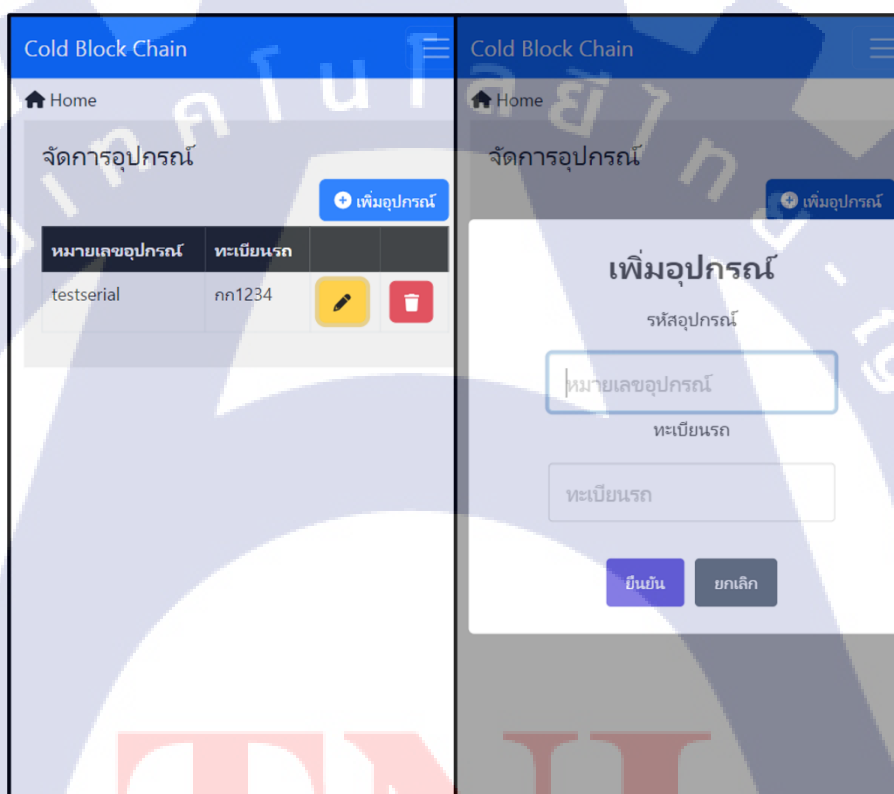


**รูปที่ 4.9** หน้าจัดการสินค้าแสดงรายการสินค้าทั้งหมดและการเพิ่มสินค้าเข้าสู่ระบบ

**รูปที่ 4.9** แสดงถึงหน้าการจัดการสินค้า โดยผู้ใช้งานที่มีสิทธิ์แอดมินเท่านั้นที่สามารถเข้าสู่หน้านี้ได้ หน้าจอจะแสดงรายการชนิดสินค้าทั้งหมดในระบบรวมถึงเกณฑ์อุณหภูมิต่ำสุดและเกณฑ์อุณหภูมิสูงสุด (หน่วยองศาเซลเซียส) ที่เหมาะสมในการจัดเก็บและขนส่งสินค้า แอดมินสามารถเพิ่มสินค้าชนิดใหม่เข้าไปในระบบได้ด้วยการกดปุ่มเพิ่มผลิตภัณฑ์โดยจะแสดงหน้าต่างให้กรอกชื่อผลิตภัณฑ์รวมถึงค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุดในแบบฟอร์ม แอดมินยังสามารถแก้ไขข้อมูลของ

สินค้าและลบสินค้าออกจากระบบได้เช่นกันแต่การลบสินค้าจะต้องเป็นสินค้าที่ไม่ได้กำลังจัดส่งและไม่เคยถูกจัดส่งไปแล้วเท่านั้นเพื่อป้องกันความขัดแย้งของการเชื่อมโยงข้อมูลในกรณีที่ไม่แน่ใจว่าจัดส่งเก่าเคยจัดส่งสินค้าอะไรบ้าง เกณฑ์อุณหภูมิของสินค้าทุกชนิดที่บันทึกในระบบจะถูกนำไปตรวจสอบเงื่อนไขผ่านทางโค้ดส่วนที่ประมวลผลอยู่ในอุปกรณ์เพื่อให้มั่นใจว่าสินค้ายังอยู่ในอุณหภูมิที่เหมาะสมเพื่อลดอัตราความสูญเสียจากสินค้าเน่าเสียหรือไม่สามารถบริโภคได้

#### 4.2.6 หน้าจัดการอุปกรณ์

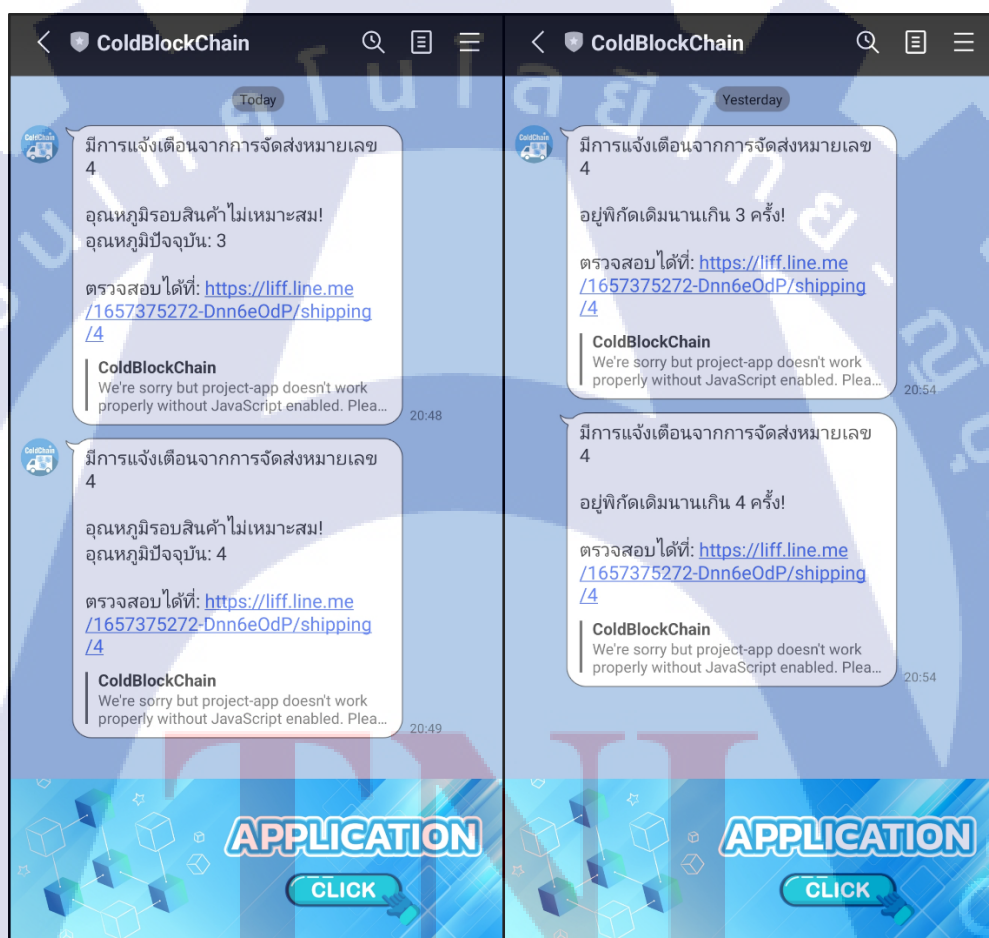


รูปที่ 4.10 หน้าจัดการอุปกรณ์แสดงรายการอุปกรณ์ IoT ทั้งหมดและการเพิ่มอุปกรณ์เข้าสู่ระบบ

**รูปที่ 4.10** แสดงถึงหน้าการจัดการอุปกรณ์ โดยผู้ใช้งานที่มีสิทธิ์แอดมินเท่านั้นที่สามารถเข้าสู่นี้นี้ได้ หน้าจอจะแสดงรายการอุปกรณ์ IoT ทั้งหมดในระบบรวมถึงหมายเลขทะเบียนรถที่ติดตั้งอุปกรณ์ชิ้นนั้นด้วย แอดมินสามารถเพิ่มอุปกรณ์ชิ้นใหม่พร้อมหมายเลขทะเบียนรถเข้าไปในระบบได้ด้วยการกดปุ่มเพิ่มอุปกรณ์โดยจะแสดงหน้าต่างให้กรอกหมายเลขรหัสอุปกรณ์ 16 หลักของ Raspberry PI และหมายเลขทะเบียนรถในแบบฟอร์ม แอดมินยังสามารถแก้ไขข้อมูลของอุปกรณ์และลบอุปกรณ์ออกจากระบบได้เช่นกันแต่การลบอุปกรณ์จะต้องเป็นอุปกรณ์ที่ไม่ได้กำลังจัดส่งและ

ไม่เคยถูกจัดส่งเช่นเดียวกับการลบสินค้า หลังจากที่เราแอดมินเพิ่มอุปกรณ์เข้าสู่ระบบแล้ว เมื่อเปิดการทำงานของอุปกรณ์ที่ติดตั้งบนรถขนส่งสินค้าจะมีการดึงค่าหมายเลขรหัสเพื่อค้นหาการจัดส่งที่กำลังใช้งานอุปกรณ์ชิ้นนั้นและมีสถานะการจัดส่งเป็นกำลังจัดส่ง หากค้นเจอจะส่ง id การจัดส่งครั้งนั้นกลับไปยังอุปกรณ์ และทุก ๆ ระยะเวลาที่ตั้งค่าไว้อุปกรณ์จะอ่านค่าอุณหภูมิและวัดพิกัดปัจจุบันส่งกลับไปที่เราแม่ข่าย

#### 4.2.7 การแจ้งเตือนที่ได้รับจากระบบ

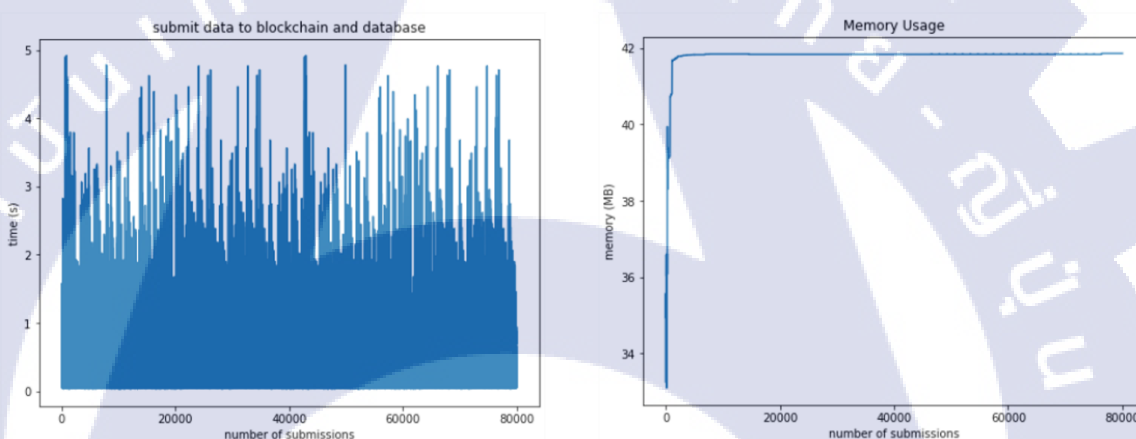


รูปที่ 4.11 การแจ้งเตือนที่ได้รับจากระบบเมื่อตรวจสอบเงื่อนไขอุณหภูมิหรือพิกัดไม่ถูกต้อง

จากรูปที่ 4.11 เป็นตัวอย่างการแจ้งเตือนที่จะได้รับจากระบบ ในช่วงเวลาที่อุปกรณ์จะส่งข้อมูลมาบันทึกยังบล็อกเชนและฐานข้อมูล หากตรวจสอบพบว่าอุณหภูมิในรถขนส่งสินค้าเปรียบเทียบกับแล้วมากกว่าเกณฑ์อุณหภูมิสูงสุดของสินค้าหรือน้อยกว่าเกณฑ์อุณหภูมิต่ำสุดของสินค้า เกินจำนวน

ครั้งที่กำหนด หรือ พิกัดทั้งละติจูดและลองจิจูดของรถขนส่งสินค้าอยู่จุดเดิมเป็นระยะเวลานานกว่าที่กำหนด โดยในการทดลองตัวอย่างคือ อุณหภูมิเกินเป็นจำนวน 3 ครั้งหรืออยู่พิกัดเดิมมากกว่า 30 วินาทีจึงจะส่งแจ้งเตือนมายังพนักงานขนส่งที่รับผิดชอบการจัดส่งครั้งนั้นผ่านทาง LINE Official Account เป็นข้อความแจ้งเตือนโดยบอกรายละเอียดว่าการแจ้งเตือนนี้เกิดจากเงื่อนไขใดพร้อมทั้งแนบข้อมูลมาด้วย เช่น หากเงื่อนไขคืออุณหภูมิไม่เหมาะสมจะบอกอุณหภูมิที่ตรวจพบและหากเงื่อนไขคืออยู่พิกัดเดิมเป็นเวลานานเกินไปจะบอกจำนวนครั้งที่ตรวจสอบแล้วพบว่าผิดเงื่อนไขอีกด้วย รวมถึงส่วนท้ายของข้อความจะแนบ URL ของการจัดส่งที่เกิดปัญหามาด้วย เมื่อผู้ใช้กด URL ตัว Web Application จะเปิดหน้าจอแสดงรายละเอียดของการจัดส่งนั้นขึ้นมาทันที

#### 4.3 ผลการวัดประสิทธิภาพของระบบ



รูปที่ 4.12 เวลาที่ใช้ในการเก็บและส่งข้อมูล (ซ้าย) และหน่วยความจำที่ใช้ (ขวา)

จากการวัดประสิทธิภาพของระบบซึ่งมีเกณฑ์ในการวัดคือ 1) เวลาที่ใช้ตั้งแต่เก็บข้อมูลจากอุปกรณ์จนกระทั่งข้อมูลถูกบันทึกในเครือข่ายบล็อกเชนและฐานข้อมูล 2) หน่วยความจำที่ใช้ระหว่างการเก็บและส่งข้อมูลโดยได้เก็บค่าจากการส่งข้อมูลทั้งหมด 80,000 ครั้ง ผลที่ได้ดังแสดงในรูปที่ 4.12 คือใช้หน่วยความจำเฉลี่ย 41.79 เมกะไบต์ เมื่อส่งข้อมูลครั้งแรกจะใช้หน่วยความจำ 33.25 เมกะไบต์ โดยมีอัตราการเพิ่มการใช้งานหน่วยความจำอยู่ที่ 0.0082 เมกะไบต์/200 ครั้ง แต่ใช้มากที่สุดไม่เกิน 42 เมกะไบต์

ในส่วนของการวัดประสิทธิภาพเรื่องเวลาที่ใช้ในการเก็บและส่งข้อมูล ได้ผลดังตารางที่ 4.1

ตารางที่ 4.1 เวลาที่ใช้ในการเก็บและส่งข้อมูลไปยังบล็อกเชนเทียบกับบล็อกเชน Ethereum [27]

| รายการ                           | เวลาเฉลี่ย (วินาที) | เวลามากที่สุด (วินาที) |
|----------------------------------|---------------------|------------------------|
| 1. บล็อกเชนที่พัฒนาด้วยภาษาไพทอน | 0.34                | 4.92                   |
| 2. บล็อกเชน Ethereum             | 11                  | 71                     |

ผลลัพธ์จากการประเมินประสิทธิภาพของระบบที่รับข้อมูลจากอุปกรณ์แล้วส่งข้อมูลไปยังเครือข่ายบล็อกเชนที่พัฒนาด้วยภาษาไพทอน โดยที่รันบนเครื่องแม่ข่ายของตัวเองจะใช้เวลาน้อยกว่าบล็อกเชนของ Ethereum



## บทที่ 5

### บทสรุปและข้อเสนอแนะ

การวิจัยเรื่อง ระบบตรวจสอบการขนส่งห่วงโซ่ความเย็นสำหรับอาหารด้วยเทคโนโลยีบล็อกเชนและอินเทอร์เน็ตของสรรพสิ่ง มีวัตถุประสงค์คือ 1) เพื่อพัฒนาระบบจัดเก็บข้อมูลแบบบล็อกเชน ร่วมกับการรับข้อมูลด้วยเทคโนโลยี IoT 2) เพื่อพัฒนา Web Application ที่สามารถดูบันทึกข้อมูล อุณหภูมิระหว่างการขนส่งสินค้าได้ และสามารถแจ้งเตือนไปยังผู้รับผิดชอบการขนส่งในขณะนั้น เมื่ออุณหภูมิรอบสินค้าสูงหรือต่ำกว่าค่าที่เหมาะสม 3) เพื่อศึกษาประสิทธิภาพในการส่งข้อมูลจากอุปกรณ์ไปยังเครือข่ายบล็อกเชน ได้นำเสนอข้อสรุปการวิจัยตามลำดับดังนี้

#### 5.1 สรุปผลการวิจัย

#### 5.2 ข้อเสนอแนะงานวิจัย

#### 5.1 สรุปผลการวิจัย

จากการพัฒนาระบบบล็อกเชนด้วยภาษาไอทอนรวมทั้งโปรแกรมส่วนอื่นบนเครื่องแม่ข่ายที่รันผ่าน Flask Python และใช้ฐานข้อมูลเป็น MySQL ร่วมกับการพัฒนา Web Application ด้วยเฟรมเวิร์ก Vue.js โดยใช้ภาษา JavaScript เป็นหลัก ร่วมกับการออกแบบส่วนที่ใช้ในการเชื่อมต่อกับผู้ใช้งานด้วย HTML แอปพลิเคชันสามารถใช้งานได้ และข้อมูลจากอุปกรณ์ถูกบันทึกลงในบล็อกเชน รวมถึงฐานข้อมูลด้วย และบล็อกเชนที่รันในเครื่องแม่ข่ายต่างกันยังสามารถเชื่อมข้อมูลเดียวกันทำให้สามารถมั่นใจในตัวข้อมูลได้ เพราะยังมีจำนวนเครื่องที่เข้าร่วมเป็นโหนดในเครือข่ายบล็อกเชนมากเท่าใด หมายความว่าข้อมูลจะยิ่งสูญหายหรือถูกโจมตีเพื่อแก้ไขข้อมูลได้ยากขึ้นเท่านั้น และยังเพิ่มความไว้วางใจในตัวข้อมูลให้กับผู้บริโภคด้วย ระบบแจ้งเตือนเมื่ออุณหภูมิของรถขนส่งสินค้ามากกว่าเกณฑ์อุณหภูมิสูงสุดหรือต่ำกว่าเกณฑ์อุณหภูมิสูงสุดของสินค้าจะเป็นการส่งข้อความไปยัง LINE Official ของพนักงานขนส่งที่รับผิดชอบโดยแจ้งรายละเอียดการแจ้งเตือนด้วยและพนักงานสามารถเข้าไปตรวจสอบได้โดยตรงผ่าน URL ที่แนบมากับข้อความแจ้งเตือน ทั้งนี้เพื่อลดโอกาสที่สินค้าจะเสียหายพลอยทิ้งไว้โดยที่พนักงานขนส่งไม่ทราบอุณหภูมิที่เปลี่ยนไปและส่งผลกระทบต่อความมั่นใจในสินค้าของผู้บริโภคด้วย และจากการวัดประสิทธิภาพของการจัดเก็บข้อมูลด้วยบล็อกเชนที่รันบนเครื่องแม่ข่ายของตัวเองใช้เวลาเฉลี่ย 0.34 วินาที โดยที่ใช้เวลามากที่สุดคือ 4.92 วินาที ซึ่งเมื่อเปรียบเทียบกับ การรับบล็อกเชนผ่านเครือข่าย Ethereum ถือว่าลดเวลาในการตรวจสอบและเก็บข้อมูลค่อนข้างมาก อีกทั้งหน่วยความจำที่ต้องการในการรันอุปกรณ์และส่งข้อมูลเฉลี่ยที่ 33.25 เมกะไบต์ หมายความว่า

อุปกรณ์ประมวลผลขนาดเล็ก เช่น Raspberry PI ที่มีหน่วยความจำ 512 เมกะไบต์ถึง 1 กิกะไบต์ (1,024 เมกะไบต์) ก็สามารถใช้ในการส่งข้อมูลได้โดยไม่มีปัญหา

## 5.2 ข้อเสนอแนะงานวิจัย

5.2.1 สามารถพัฒนาต่อยอดระบบนี้ในอนาคตได้ เช่น การนำข้อมูลจากการเดินทางจัดส่งเพื่อนำไปหาเส้นทางการจัดส่งที่ดีที่สุดในแต่ละครั้ง ซึ่งจะสามารถลดการบริโภคเชื้อเพลิงที่เป็นค่าใช้จ่ายสำคัญ

5.2.2 หากนำไปใช้กับระบบการขนส่งจริง จะสามารถวัดอัตราการลดลงของอาหารเน่าเสียจากอุณหภูมิไม่ปกติและขยะที่ลดลงจากการขนส่งล่าช้าได้



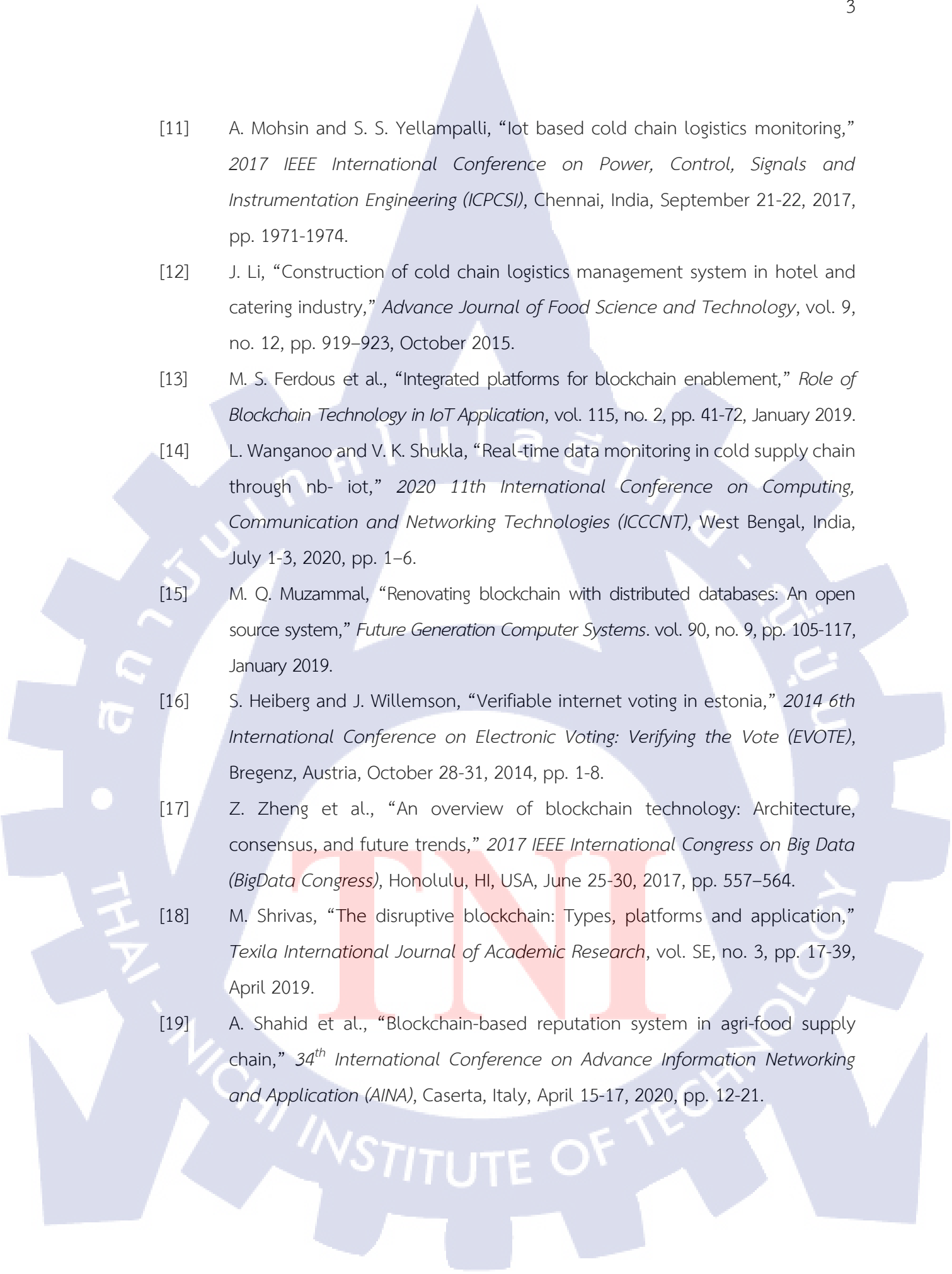
TNI

บรรณานุกรม

TNI

## บรรณานุกรม

- [1] Logistics Development Strategy Division, *Thailand's Logistics Report 2019*, Bangkok: Office of the National Economic and Social Development Council, 2020.
- [2] H. Li and P. Pan, "Food waste in developed countries and cold chain logistics," *E3S Web of Conferences*, vol. 251, no. 11, pp. 3001, January 2021.
- [3] K. Liljestrand, "Logistics solutions for reducing food waste," *International Journal of Physical Distribution & Logistics Management*, vol. 47, no. 4, pp. 318 – 339, May 2017.
- [4] Statista, "Food & Beverages - Thailand: Statista Market Forecast," [Online]. Available: <https://www.statista.com/outlook/dmo/ecommerce/food-personal-care/food-beverages/thailand>. [Accessed: October 05, 2021].
- [5] Macroeconomic Strategy and Planning Division, "NESDC Economic Report Thai Economic Performance in Q4 and 2021 and Outlook for 2022," [Online]. Available: [https://www.nesdc.go.th/nesdb\\_en/article\\_attach/article\\_file\\_20220228094737.pdf](https://www.nesdc.go.th/nesdb_en/article_attach/article_file_20220228094737.pdf). [Accessed: September 24, 2022].
- [6] K. Wichiantanon, "Combatting Food Waste," [Online]. Available: <https://www.bangkokpost.com/life/social-and-lifestyle/2074863/combating-food-waste>. [Accessed: October 05, 2021].
- [7] GUNDIG, "Wasting Food Is Wasting Life. How Much Food Do You Waste?," [Online]. Available: <https://www.respectfood.com/#waste-calculator>. [Accessed: September 17, 2022].
- [8] BEHR TECH, "6 Leading Types of IoT Wireless Tech and Their Best Use Cases," [Online]. Available: <https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/>. [Accessed: August 25, 2022].
- [9] R. T. Hasanat et al., "An iot based real-time data-centric monitoring system for vaccine cold chain," *2020 IEEE East-West Design Test Symposium (EWDTS)*, Batumi, Georgia, September 4-7, 2020, pp. 1-5.
- [10] B. R. Haverkort and A. Zimmermann, "Smart industry: How ict will change the game!," *IEEE Internet Computing*, vol. 21, no. 1, pp. 8-10, February 2017.

- 
- [11] A. Mohsin and S. S. Yellampalli, "Iot based cold chain logistics monitoring," *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, India, September 21-22, 2017, pp. 1971-1974.
- [12] J. Li, "Construction of cold chain logistics management system in hotel and catering industry," *Advance Journal of Food Science and Technology*, vol. 9, no. 12, pp. 919-923, October 2015.
- [13] M. S. Ferdous et al., "Integrated platforms for blockchain enablement," *Role of Blockchain Technology in IoT Application*, vol. 115, no. 2, pp. 41-72, January 2019.
- [14] L. Wanganoo and V. K. Shukla, "Real-time data monitoring in cold supply chain through nb- iot," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, West Bengal, India, July 1-3, 2020, pp. 1-6.
- [15] M. Q. Muzammal, "Renovating blockchain with distributed databases: An open source system," *Future Generation Computer Systems*, vol. 90, no. 9, pp. 105-117, January 2019.
- [16] S. Heiberg and J. Willemson, "Verifiable internet voting in estonia," *2014 6th International Conference on Electronic Voting: Verifying the Vote (EVOTE)*, Bregenz, Austria, October 28-31, 2014, pp. 1-8.
- [17] Z. Zheng et al., "An overview of blockchain technology: Architecture, consensus, and future trends," *2017 IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, June 25-30, 2017, pp. 557-564.
- [18] M. Shrivastava, "The disruptive blockchain: Types, platforms and application," *Texila International Journal of Academic Research*, vol. SE, no. 3, pp. 17-39, April 2019.
- [19] A. Shahid et al., "Blockchain-based reputation system in agri-food supply chain," *34<sup>th</sup> International Conference on Advance Information Networking and Application (AINA)*, Caserta, Italy, April 15-17, 2020, pp. 12-21.

- [20] M. Memon et al., "Blockchain beyond bitcoin: Blockchain technology challenges and real-world applications," *2018 International Conference on Computing, Electronics Communications Engineering (iCCECE)*, Southend, UK, August 16-17, 2018, pp. 29–34.
- [21] S. Madumidha et al., "Transparency and traceability: In food supply chain system using blockchain technology with internet of things," *2019 3<sup>rd</sup> International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, April 23-25, 2019, pp. 983–987.
- [22] T. Surasak et al., "Thai agriculture products traceability system using blockchain and internet of things," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 76, pp. 578-583, October 2019.
- [23] F. Tian, "A supply chain traceability system for food safety based on haccp, blockchain & internet of things," *2017 International Conference on Service Systems and Service Management*, Dalian, China, June 16-18, 2017, pp. 1-6.
- [24] M. Schaffer et al., "Performance and scalability of private ethereum blockchains," *Business Process Management: Blockchain and Central and Eastern Europe Forum*, Vienna, Austria, August 1-6, 2019, pp. 103-118.
- [25] L. Chen et al., "On security analysis of proof-of-elapsd-time (poet)," *International Symposium on Stabilization, Satety, and Security of Distributed Systems*, Boston, MA, USA, November 5-8, 2017, pp. 282–297.
- [26] B. Ampel et al., "Performance modeling of Hyperledger sawtooth blockchain," *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Shenzhen, China, July 1-3, 2019, pp. 59-61.
- [27] R. D. Mendonca et al., "Blockcoldchain: Vaccine Cold Chain Blockchain," [Online]. Available: <https://arxiv.org/abs/2104.14357>. [Accessed: October 05, 2021]
- [28] S. R. Salzedo, *Cryptography: Springer Undergraduate Mathematics Series*, Berlin: Springer, 2018.
- [29] Information Security, "การเข้ารหัสข้อมูล," [Online]. Available: <http://it02-security.blogspot.com/2015/03/2.html>. [Accessed: 25 กันยายน 2565].

- [30] ทีมสนับสนุน SSL, “Cryptographic Hash Function คืออะไร,” [Online]. Available: <https://www.ssl.com/th/คำถามที่พบบ่อย/ฟังก์ชันแฮชการเข้ารหัสลับคืออะไร/>. [Accessed: 25 กันยายน 2565].
- [31] B. Patchara, “Docker คืออะไร? เครื่องมือยอดนิยมที่จะช่วย Dev ทำงานเร็วขึ้น 2 เท่า!,” [Online]. Available: <https://blog.skooldio.com/what-is-docker/#Docker>. [Accessed: 26 กันยายน 2565].
- [32] Python Basic, “What is Flask Python,” Python Tutorial [Online]. Available: <https://pythonbasics.org/what-is-flask-python/>. [Accessed: September 26, 2022].
- [33] INSAWASD, “Vue.js คืออะไร มีดีอย่างไร ทำไมถึงต้องใช้,” [Online]. Available: <https://insawasd.com/what-is-vuejs/>. [Accessed: 26 กันยายน 2565].
- [34] 8columns, “เว็บแบบ SPA คืออะไร,” [Online]. Available: <https://8columns.com/page/4/>. [Accessed: 26 กันยายน 2565].
- [35] Line, “LINE Messaging API SDK for Python,” [Online]. Available: <https://github.com/line/line-bot-sdk-python> [Accessed: June 15, 2022].

ภาคผนวก

TNI

ภาคผนวก ก.

โค้ดภาษาไพทอน (Python)

TNI

### โค้ดสร้างฐานข้อมูลเริ่มต้น

```
# library ที่จำเป็นในการสร้างฐานข้อมูล
import os
import mysql.connector
from dotenv import load_dotenv

# ทำให้โปรแกรมสามารถอ่านข้อมูลจากไฟล์ .env ได้
load_dotenv()

# กำหนดข้อมูลที่จำเป็นในการสร้างตารางเก็บไว้ในตัวแปรที่เป็น Array JSON
# ชื่อของตารางจะอยู่ที่ฟิลด์ name และคอลัมน์ต่าง ๆ จะอยู่ที่ฟิลด์ columns
tables = [
    {
        "name": "employee",
        "columns": {
            "id": "INT AUTO_INCREMENT PRIMARY KEY",
            "name": "VARCHAR(100)",
            "surname": "VARCHAR(100)",
            "id_number": "VARCHAR(13)",
            "phone_number": "VARCHAR(10)",
            "position": "ENUM('admin', 'driver', 'user')",
            "user_line_id": "VARCHAR(64)"
        }
    },
    {
        "name": "device",
        "columns": {
            "id": "INT AUTO_INCREMENT PRIMARY KEY",
            "serial_number": "VARCHAR(16)",
            "vehicle_license": "VARCHAR(10)",
            "usable": "TINYINT(1) DEFAULT 0"
        }
    }
]
```

```

},
{
  "name": "product",
  "columns": {
    "id": "INT AUTO_INCREMENT PRIMARY KEY",
    "name": "VARCHAR(100)",
    "max_temp": "FLOAT",
    "min_temp": "FLOAT"
  }
},
{
  "name": "shipping",
  "columns": {
    "id": "INT AUTO_INCREMENT PRIMARY KEY",
    "employee_id": "INT",
    "device_id": "INT",
    "origin_latitude": "DECIMAL(10,7)",
    "origin_longitude": "DECIMAL(10,7)",
    "dest_latitude": "DECIMAL(10,7)",
    "dest_longitude": "DECIMAL(10,7)",
    "status": "ENUM('created','shipping','completed','canceled')",
    "created_at": "TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP",
    "updated_at": "TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP",
    "FOREIGN KEY(employee_id)": "REFERENCES employee(id)",
    "FOREIGN KEY(device_id)": "REFERENCES device(id)",
  }
},
{
  "name": "sensor_data",
  "columns": {
    "id": "INT AUTO_INCREMENT PRIMARY KEY",

```

```

        "shipping_id": "INT",
        "inside_temp": "DECIMAL(3,2)",
        "outside_temp": "DECIMAL(3,2)",
        "latitude": "DECIMAL(10,7)",
        "longitude": "DECIMAL(10,7)",
        "created_at": "TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP",
        "FOREIGN KEY(shipping_id)": "REFERENCES shipping(id)",
    }
},
{
    "name": "shipping_detail",
    "columns": {
        "id": "INT AUTO_INCREMENT PRIMARY KEY",
        "shipping_id": "INT",
        "product_id": "INT",
        "collected_date": "TIMESTAMP",
        "FOREIGN KEY(shipping_id)": "REFERENCES shipping(id)",
        "FOREIGN KEY(product_id)": "REFERENCES product(id)",
    }
},
]

```

*# เชื่อมต่อกับฐานข้อมูลและส่งตัวแปรที่เป็นตัวเชื่อมฐานข้อมูลกลับไป*

```

def declare_database(db_name=None):
    mydb = mysql.connector.connect(
        host=host,
        user=username,
        password=password,
        database=db_name,
    )
    return mydb

```

*# สร้างฐานข้อมูลชื่อเดียวกับ db\_name โดยมีการตรวจสอบว่ายังไม่เคยมีฐานข้อมูลนี้มาก่อน*

```
def create_database(db, db_name):
    cursor = db.cursor()
    sql = f"CREATE DATABASE IF NOT EXISTS {db_name}"
    try:
        cursor.execute(sql)
    except:
        pass
    cursor.close()
    db.close()
```

*# วนลูปสร้างตารางตามที่กำหนดไว้ในตัวแปร tables*

*# โดยมีการตรวจสอบว่าตารางนั้น ๆ ยังไม่เคยมีในฐานข้อมูลนี้มาก่อน*

```
def create_tables(db):
    for table in tables:
        cursor = db.cursor()
        sql = f"CREATE TABLE IF NOT EXISTS {table['name']} ("
        sql += ", ".join(f"{key} {value}" for key, value in table["columns"].items())
        sql += ")"
        try:
            cursor.execute(sql)
        except:
            pass
        cursor.close()
```

*# ดึงตัวแปร host, username, password จากไฟล์ .env ที่สร้างแยกไว้*

*# หากไม่เจอไฟล์ .env หรือไม่พบตัวแปรที่สร้างไว้ จะใช้ค่าเริ่มต้นที่อยู่ใน else แทน*

```
host = os.getenv("db_host") if os.getenv("db_host") else "localhost"
username = os.getenv("db_username") if os.getenv("db_username") else "root"
password = os.getenv("db_password") if os.getenv("db_password") else ""
```

# ฟังก์ชันหลักสำหรับเรียงลำดับการเรียกใช้ฟังก์ชันที่ประกาศไว้ก่อนหน้านี้

```
if __name__ == "__main__":
    db_name = os.getenv("db_name") if os.getenv("db_name") else "project"
    db = declare_database()
    create_database(db, db_name)
    new_db = declare_database(db_name)
    create_tables(new_db)
```

โค้ดส่งข้อมูลจากอุปกรณ์ไปยังบล็อกเชน

# library ที่จำเป็นในการอ่านค่าจากอุปกรณ์และฐานข้อมูล

```
import dht
import os
import mysql.connector
import requests
import threading
from machine import Pin
from gps import *
from dotenv import load_dotenv
# library เพื่อเชื่อมต่อกับ line bot
from linebot import LineBotApi
from linebot.models import TextSendMessage
# ทำให้โปรแกรมสามารถอ่านข้อมูลจากไฟล์ .env ได้
load_dotenv()
```

# ดึงตัวแปร host, username, password และ db\_name

# รวมถึงจำนวนที่ใช้ตรวจสอบเงื่อนไขอุณหภูมิและเงื่อนไขพิกัด จากไฟล์ .env ที่สร้างแยกไว้

# หากไม่เจอไฟล์ .env หรือไม่พบตัวแปรที่สร้างไว้ จะใช้ค่าเริ่มต้นที่อยู่ใน else แทน

```
host = os.getenv("db_host") if os.getenv("db_host") else "localhost"
username = os.getenv("db_username") if os.getenv("db_username") else "root"
```

```

password = os.getenv("db_password") if os.getenv("db_password") else ""
db_name = os.getenv("db_name") if os.getenv("db_name") else "project"
max_temperature_count = int(os.getenv("temp_count")) if os.getenv("temp_count")
else 3
max_location_count = int(os.getenv("location_count")) if
os.getenv("location_count") else 3
inside_pin = int(os.getenv("inside_pin")) if os.getenv("inside_pin") else 14
outside_pin = int(os.getenv("outside_pin")) if os.getenv("outside_pin") else 15

# กำหนดให้เซนเซอร์อ่านค่าจากอุปกรณ์โดยใช้ PIN จากตัวแปร inside_pin และ outside_pin
# และกำหนดตัวแปรที่จะใช้อ่านค่าพิกัด
inside_temp_sensor = dht.DHT22(Pin(inside_pin))
outside_temp_sensor = dht.DHT22(Pin(outside_pin))
gpsd = gps(mode=WATCH_ENABLE|WATCH_NEWSTYLE)

# สร้างคลาสของ Device โดยข้างในคลาสจะมีฟังก์ชันและตัวแปรอยู่
class Device(object):

    # ฟังก์ชันสำหรับอ่านค่าหมายเลขรหัสอุปกรณ์และส่งกลับไปผ่านตัวแปร serial
    # ถ้าไม่สามารถอ่านค่าได้ จะส่งค่ากลับไปเป็นค่าเริ่มต้นที่กำหนดไว้
    def get_serial(self):
        serial = "testserial"
        try:
            f = open('/proc/cpuinfo', 'r')
            for line in f:
                if line[0:6] == 'Serial':
                    serial = line[10:26]
            f.close()
        except:
            pass
        return serial

```

```

# ตัวแปรเริ่มต้นของคลาส Device เมื่อโปรแกรมถูกรันขึ้นครั้งแรก
def __init__(self):
    self.serial = self.get_serial()
    self.inside_temp = None
    self.outside_temp = None
    self.latitude = None
    self.longitude = None
    self.location = None
    self.temp_fault_count = 0
    self.location_fault_count = 0

# ฟังก์ชันเพื่อค้นหาจากฐานข้อมูลว่ามีการจัดส่งที่ใช้หมายเลขรหัสอุปกรณ์นี้
# และอยู่ในสถานะ 'กำลังจัดส่ง' หรือไม่ ถ้าเจอจะส่งค่ารหัสการจัดส่งกลับมา
# รวมถึงค่า User LINE ID ของพนักงานขนส่งที่รับผิดชอบการจัดส่งนั้นด้วย
def check_active_shipping(self, mydb):
    cursor = mydb.cursor(dictionary=True)
    sql = f"SELECT * FROM device WHERE serial_number = '{self.serial}'"
    cursor.execute(sql)
    device_result = cursor.fetchone()
    cursor.close()
    if device_result is not None:
        cursor_2 = mydb.cursor(dictionary=True)
        sql_2 = """
            SELECT s.id, s.status, e.user_line_id
            FROM shipping AS s
            JOIN employee AS e ON s.employee_id = e.id
            WHERE device_id = %s AND status = 'shipping' ORDER BY created_at
            DESC;
        """
        val = (device_result['id'],)
        cursor_2.execute(sql_2, val)

```

```

shipping_result = cursor_2.fetchone()
cursor_2.close()

if shipping_result is not None:
    return shipping_result['id'], shipping_result['user_line_id']
return False, False

# ฟังก์ชันเพื่ออ่านข้อมูลอุณหภูมิและพิกัดจากอุปกรณ์
def read_data(self):
    self.inside_temp = inside_temp_sensor.temperature()
    self.outside_temp = outside_temp_sensor.temperature()
    location = gpsd.next()
    self.latitude = getattr(location, 'lat', 0.0)
    self.longitude = getattr(location, 'lon', 0.0)

# ส่งข้อมูลไปยังบล็อกเชน
def send_data(self, shipping_id):
    # อ่านค่า URL ของปลายทางที่ต้องส่งจากไฟล์ .env
    base_url = os.getenv('base_url')
    url = f"{base_url}/transactions/new"
    data = self.__dict__.copy()
    data['shipping_id'] = shipping_id
    # จัดข้อมูลในตัวแปร data ให้เหลือแต่ฟิลด์ที่ต้องการส่ง
    select_keys = ['inside_temp', 'outside_temp', 'latitude', 'longitude',
'shipping_id']
    for key in list(data.keys()):
        if key not in select_keys:
            del data[key]
    try:
        requests.post(url=url, json=data)
    except:
        pass

```

```

# ฟังก์ชันตรวจสอบอุณหภูมิของสินค้า
def check_temperature(self, shipping_id):
    db = connect_db()
    cursor = db.cursor(dictionary=True)
    # ดึงข้อมูลสินค้าทุกชิ้นที่อยู่ในการจัดส่งครั้งนี้
    sql = """
        SELECT s.id, p.name, p.max_temp, p.min_temp FROM shipping_detail AS s
        JOIN product AS p ON s.product_id = p.id
        WHERE s.id = %s;
    """
    val = (shipping_id,)
    cursor.execute(sql, val)
    result = cursor.fetchall()
    cursor.close()
    check = True
    # ตรวจสอบว่าสินค้าทุกชิ้นอยู่ในอุณหภูมิที่เหมาะสมหรือไม่
    # ถ้ามีสินค้าชิ้นใดชิ้นหนึ่งที่ผิดเงื่อนไข จะเพิ่มจำนวนที่ใช้ค้เงื่อนไขผิดพลาด
    (temp_fault_count)
    for row in result:
        if (self.inside_temp > row['max_temp'] or self.inside_temp <
row['min_temp']):
            self.temp_fault_count += 1
            check = False
            break
    if check:
        self.temp_fault_count = 0
    # หากจำนวนที่ใช้ค้เงื่อนไขผิดพลาดมากกว่าหรือเท่ากับที่กำหนดไว้ จะส่งค่ากลับไปเป็น
    False
    if self.temp_fault_count >= max_temperature_count:
        return False, self.inside_temp

```

```

return True, 0

# ฟังก์ชันพิกัดของรถขนส่งสินค้า
def check_location(self):
    # ตรวจสอบว่าพิกัดของรถขนส่งสินค้าอยู่พิกัดเดียวกับการตรวจสอบครั้งก่อนหน้าหรือไม่
    # ถ้าอยู่จุดเดียวกัน จะเพิ่มจำนวนที่ใช้เครื่องยนต์ผิดพลาด (location_fault_count)
    if self.location == (self.latitude, self.longitude):
        self.location_fault_count += 1
    else:
        self.location = (self.latitude, self.longitude)
        self.location_fault_count = 0
    # หากจำนวนที่ใช้เครื่องยนต์ผิดพลาดมากกว่าหรือเท่ากับที่กำหนดไว้ จะส่งค่ากลับไปเป็น
    False
    if self.location_fault_count >= max_location_count:
        return False, self.location_fault_count
    return True, 0

# ระยะเวลาที่จะให้โปรแกรมอ่านค่าจากอุปกรณ์และส่งข้อมูลใหม่
period = 10

# สร้างตัวแปรที่เป็นคลาส Device เมื่อเริ่มรันโปรแกรม
device = Device()

# ฟังก์ชันสำหรับส่งแจ้งเตือนไปยัง LINE Official
def line_push_message(user_id, shipping_id, message, data):
    text_message = f"มีการแจ้งเตือนจากการจัดส่งหมายเลข {shipping_id}\n\n"
    if message == "location":
        text_message += f"อยู่พิกัดเดิมนานเกิน {data} ครั้ง!\n\n"
    elif message == "temperature":
        text_message += f"อุณหภูมิรอบสินค้าไม่เหมาะสม! อุณหภูมิปัจจุบัน: {data}\n\n"
    text_message += f"ตรวจสอบได้ที่: {os.getenv('liff_url')+'/shipping/'+str(shipping_id)}"
    line_bot_api = LineBotApi(os.getenv('channel_access_token'))

```

```

line_bot_api.push_message(user_id,
    TextSendMessage(text=text_message))

# เชื่อมต่อกับฐานข้อมูลและส่งตัวแปรที่เป็นตัวเชื่อมฐานข้อมูลกลับไป
def connect_db():
    mydb = mysql.connector.connect(
        host=host,
        user=username,
        password=password,
        database=db_name,
    )
    return mydb

# ฟังก์ชันสำหรับเรียงลำดับการเรียกใช้ฟังก์ชันที่ประกาศไว้ก่อนหน้า
# และจะทำงานใหม่โดยอัตโนมัติทุกๆระยะเวลาที่กำหนดไว้ (period)
def auto_run():
    threading.Timer(period, auto_run).start()
    mydb = connect_db()
    shipping_id, employee_line_id = device.check_active_shipping(mydb=mydb)
    if shipping_id:
        device.read_data()
        location_status, location_data = device.check_location()
        temperature_status, temperature_data =
device.check_temperature(shipping_id)
        # ถ้าเงื่อนไขพิกัดส่งค่ากลับมาเป็น False จะส่งแจ้งเตือนเรื่องพิกัด
        if not location_status:
            line_push_message(employee_line_id, shipping_id, message="location",
data=location_data)
        # ถ้าเงื่อนไขอุณหภูมิส่งค่ากลับมาเป็น False จะส่งแจ้งเตือนเรื่องอุณหภูมิ
        if not temperature_status:

```

```
line_push_message(employee_line_id, shipping_id,  
message="temperature", data=temperature_data)  
# ส่งข้อมูลไปยังบล็อกเชน โดยแวนบรหัสการจัดส่งไปด้วย  
device.send_data(shipping_id=shipping_id)  
mydb.close()  
  
if __name__ == "__main__":  
    auto_run()
```



THAI-NICHI INSTITUTE OF TECHNOLOGY

TNI