

แอปพลิเคชัน Kuaglang  
คพ.232 เทคโนโลยีกลุ่มเมฆเบื้องต้น

จัดทำโดย

จิราพร	ใจสุขุข	รหัสนักศึกษา 6609520090	กลุ่มเรียน 090002
นพกรณ์	สงเคราะห์	รหัสนักศึกษา 6609540023	กลุ่มเรียน 090002
ณัฐ	เทียมวรรณ	รหัสนักศึกษา 6609611899	กลุ่มเรียน 090002
ณัฐนันท์	ดวงจินดา	รหัสนักศึกษา 6609611923	กลุ่มเรียน 090002
ธเนศพล	ประภาศิริสุลี	รหัสนักศึกษา 6609611998	กลุ่มเรียน 090002
ปฐยาวัต	พูลสวัสดิ์	รหัสนักศึกษา 6609612111	กลุ่มเรียน 090002
วชิรกรณ์	ชิน	รหัสนักศึกษา 6609612194	กลุ่มเรียน 090002
ปรัชญา	เขียวจันทร์	รหัสนักศึกษา 6609681215	กลุ่มเรียน 090002

กลุ่ม 1

ภาคการศึกษา 2/2567  
มหาวิทยาลัยธรรมศาสตร์

## สารบัญ

ที่มาและความสำคัญ .....	2
ระบบเหมาะกับสถานการณ์ที่ผู้ใช้: .....	2
กรณีการใช้งาน (USE CASES) และประโยชน์สำหรับผู้ใช้ .....	3
สถาปัตยกรรมระบบที่แสดงให้เห็นถึงบริการ AWS ที่ใช้ พร้อมคำอธิบายการทำงานร่วมกันของ องค์ประกอบต่าง ๆ ในสถาปัตยกรรม.....	4
องค์ประกอบของระบบและบทบาทหน้าที่ .....	5
การทำงานร่วมกันขององค์ประกอบในระบบ .....	6
อธิบายขั้นตอนหลักที่สำคัญในการสร้างและตั้งค่า .....	7
ผลการทดสอบการทำงานของระบบโซลูชันตามมุมมองของผู้ใช้ (END-TO-END) ทั้งหมด 3 กรณี .....	8
บทวิเคราะห์ข้อดีและข้อเสียของระบบตามกรอบ AWS WELL-ARCHITECTED FRAMEWORK .....	20
ลิงก์คลิปวิดีโอเดโมและลิงก์ GITHUB REPOSITORY.....	22
สรุปผลการดำเนินโครงการ .....	23

## ที่มาและความสำคัญ

ในปัจจุบัน ปัญหาขยะอาหาร (Food Waste) เป็นหนึ่งในประเด็นที่ส่งผลกระทบอย่างรุนแรงต่อสิ่งแวดล้อม เศรษฐกิจ และคุณภาพชีวิตของคนในสังคม โดยเฉพาะอย่างยิ่งในระดับครัวเรือนและชุมชนขนาดเล็ก ซึ่งเป็นแหล่งกำเนิดขยะอาหารจำนวนมากแต่กลับมักถูกมองข้าม แม้จะมีความพยายามในระดับประเทศเพื่อส่งเสริมการลดขยะอาหาร แต่การสื่อสารและแนวทางส่วนใหญ่ยังขาดความเชื่อมโยงกับชีวิตประจำวันของคนทั่วไปในระดับหมู่บ้านหรือครัวเรือน ส่งผลให้ประชาชนส่วนมากไม่ตระหนักถึงผลกระทบที่ตนเองมีต่อปัญหานี้ ปัญหาที่พบได้บ่อยในชีวิตประจำวัน ได้แก่: การปล่อยให้ของสดหรือวัตถุดิบหมดอายุโดยไม่รู้ตัว หรือลืมใช้ของที่ซื้อมาจนต้องทิ้ง, การซื้อของซ้ำโดยไม่จำเป็น เพราะไม่มีระบบจัดการสิ่งของในบ้านที่ดีพอ, ความต้องการที่จะแบ่งปันวัตถุดิบหรืออาหารส่วนเกินให้คนอื่น แต่ขาดช่องทางหรือไม่รู้จะให้ใคร, การไม่สามารถประเมินภาพรวม ของการบริโภคและปริมาณขยะอาหารที่ตนเองสร้างขึ้น

ด้วยเหตุนี้ โครงการ "kua-glang" จึงถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาเหล่านี้ โดยเน้นการสร้างระบบและกิจกรรมที่ส่งเสริมการมีส่วนร่วมในชุมชนผ่านรูปแบบที่เข้าใจง่าย และใช้งานได้ง่าย ไม่ว่าจะเป็นการช่วยเตือนวันหมดอายุ การแชร์วัตถุดิบส่วนเกินกับเพื่อนบ้าน หรือการให้ข้อมูลเชิงภาพรวมเพื่อกระตุ้นการเปลี่ยนแปลงพฤติกรรม โครงการนี้มุ่งหวังให้การลดขยะอาหารไม่ใช่เรื่องยุ่งยากหรือไกลตัว

### ระบบเหมาะกับสถานการณ์ที่ผู้ใช้:

- มีวัตถุดิบเหลือจากการประกอบอาหาร เช่น นมใกล้หมดอายุ ผักเหลือหลังงานเลี้ยง
- ต้องการช่วยเหลือผู้อื่นในชุมชนด้วยการแบ่งปันอาหาร
- ต้องการติดตามและจัดการอาหารของตนเองอย่างมีประสิทธิภาพ
- ต้องการร่วมเป็นส่วนหนึ่งของแนวคิด "Zero Waste" หรือ "ลดการสูญเปล่า"

## กรณีการใช้งาน (Use Cases) และประโยชน์สำหรับผู้ใช้งาน

ระบบ Kuaglang ได้รับการออกแบบเพื่ออำนวยความสะดวกในการจัดการอาหารเหลือจากครัวเรือน โดยเปิดโอกาสให้ผู้ใช้งานสามารถแบ่งปันอาหารให้แก่ผู้อื่น ตลอดจนรับอาหารจากสมาชิกในชุมชนเดียวกัน เพื่อส่งเสริมแนวคิดการลดขยะอาหาร (food waste) และเพิ่มการเข้าถึงทรัพยากรอาหารอย่างทั่วถึง ระบบสามารถตอบสนองความต้องการของผู้ใช้งานในสถานการณ์ที่แตกต่างกันผ่านกรณีการใช้งานสำคัญ ดังนี้

### กรณีที่ 1: เพิ่มรายการอาหารเพื่อแบ่งปัน

ผู้ใช้งานสามารถเพิ่มข้อมูลเกี่ยวกับอาหารหรือวัตถุดิบที่เหลือไว้ในครัวเรือนเข้าสู่ระบบ โดยระบุชื่ออาหาร หมวดหมู่ จำนวน หน่วย วันหมดอายุ และแนบรูปภาพ ระบบจะจัดเก็บข้อมูลเหล่านี้ลงในฐานข้อมูลกลาง เพื่อให้ผู้ใช้อื่นสามารถเข้าถึงและแสดงความสนใจรับอาหารได้

ประโยชน์: ช่วยลดปริมาณของเสียจากอาหารที่ยังสามารถบริโภคได้ พร้อมทั้งส่งเสริมให้เกิดการแบ่งปันภายในชุมชนอย่างเป็นรูปธรรม

### กรณีที่ 2: การแสดงความสนใจและขอรับอาหารจากผู้อื่น

ผู้ใช้งานสามารถเข้าดูรายการอาหารที่ผู้อื่นเผยแพร่ไว้ และแสดงความสนใจเพื่อขอรับอาหารนั้น ระบบจะทำการบันทึกข้อมูลการจับคู่ และติดตามสถานะของรายการอาหารที่มีการร้องขอ รวมถึงจัดการการเปลี่ยนสถานะเมื่อมีการรับมอบสำเร็จ

ประโยชน์: เพิ่มโอกาสในการเข้าถึงอาหารโดยเฉพาะสำหรับผู้มีรายได้น้อย และส่งเสริมการใช้ทรัพยากรร่วมกันอย่างยั่งยืน

### กรณีที่ 3: การติดตามสถานะของอาหารในครัวเรือน

ผู้ใช้งานสามารถบันทึกรายการอาหารที่มีอยู่ในครัวเรือนเข้าสู่ระบบ เพื่อให้สามารถติดตามวันหมดอายุ และได้รับการแจ้งเตือนเมื่อใกล้ถึงเวลาหมดอายุกำลังใกล้เข้ามา

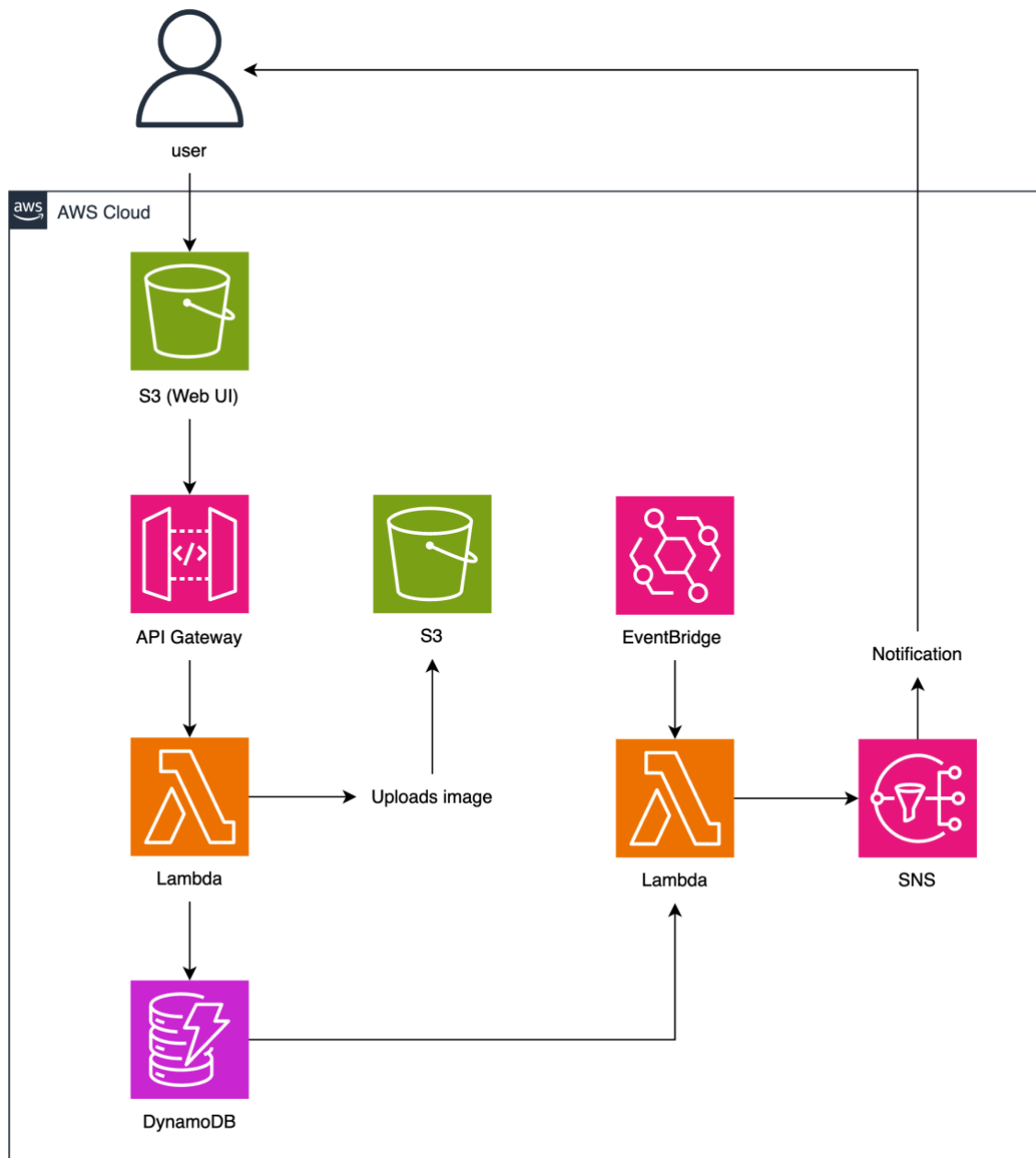
ประโยชน์: ลดโอกาสการทิ้งอาหารโดยไม่ตั้งใจ เพิ่มประสิทธิภาพในการจัดการครัว และส่งเสริมพฤติกรรมวางแผนการบริโภค

### กรณีที่ 4: การติดตามผลการใช้งานและคะแนนสะสม

ระบบจัดเก็บสถิติการใช้งานของผู้ใช้ เช่น จำนวนรายการอาหารที่แบ่งปัน จำนวนวันที่ไม่มีอาหารหมดอายุ และจำนวนการลดขยะอาหาร พร้อมทั้งให้คะแนนสะสมแก่ผู้ใช้งานตามการมีส่วนร่วม

ประโยชน์: ช่วยสร้างแรงจูงใจในการใช้งานอย่างต่อเนื่องและเสริมสร้าง community ไปพร้อมกัน

สถาปัตยกรรมระบบที่แสดงให้เห็นถึงบริการ AWS ที่ใช้ พร้อมคำอธิบายการทำงานร่วมกันขององค์ประกอบต่าง ๆ ในสถาปัตยกรรม



## องค์ประกอบของระบบและบทบาทหน้าที่

### 1. Amazon S3 (Simple Storage Service)

ใช้สำหรับโฮสต์เว็บแอปพลิเคชันฝั่งผู้ใช้ (Static Website Hosting)  
และจัดเก็บไฟล์ภาพของรายการอาหารที่ผู้ใช้อัปโหลดเข้าระบบ

### 2. Amazon API Gateway

ทำหน้าที่เป็นจุดเชื่อมต่อระหว่างฝั่งผู้ใช้งาน (Client) กับฝั่งระบบ (Backend)  
โดยรับคำขอผ่าน HTTP และส่งต่อคำขอเหล่านั้นไปยัง AWS Lambda เพื่อประมวลผล

### 3. AWS Lambda

ใช้เป็นตัวกลางสำหรับประมวลผลคำขอ เช่น การเพิ่มรายการอาหาร  
การดึงข้อมูลจากฐานข้อมูล การจัดการการอัปโหลดภาพไปยัง S3  
และการกระตุ้นเหตุการณ์ในระบบผ่าน EventBridge

### 4. Amazon DynamoDB

ทำหน้าที่เป็นฐานข้อมูล NoSQL สำหรับจัดเก็บข้อมูลที่สำคัญของระบบ ได้แก่ ข้อมูลผู้ใช้  
รายการอาหาร รายการแบ่งปัน และข้อมูลสถิติการใช้งาน

### 5. Amazon EventBridge

ใช้เป็นตัวกลางในการจัดการเหตุการณ์ (Events) ที่เกิดขึ้นในระบบ เช่น  
แจ้งเตือนอาหารใกล้หมดอายุ และสามารถเรียกใช้ Lambda เพื่อจัดการได้

### 6. Amazon SNS (Simple Notification Service)

ใช้สำหรับส่งการแจ้งเตือนไปยังผู้ใช้  
หรือส่งข้อความแจ้งเตือนทางอีเมลในกรณีที่มีเหตุการณ์ที่ต้องการแจ้งให้ทราบ เช่น  
การรับอาหารสำเร็จ หรือรายการหมดอายุ

### การทำงานร่วมกันขององค์ประกอบในระบบ

- ผู้ใช้งานเข้าถึงระบบผ่านเว็บเบราว์เซอร์ โดยโหลดหน้าเว็บจาก Amazon S3
- เมื่อผู้ใช้กรอกฟอร์มหรือดำเนินการบางอย่าง เช่น การเพิ่มรายการอาหารใหม่ คำขอจะถูกส่งผ่าน API Gateway
- API Gateway ส่งคำขอไปยังฟังก์ชัน Lambda ซึ่งจะดำเนินการประมวลผล เช่น บันทึกข้อมูลไปยัง DynamoDB และจัดเก็บรูปภาพลงใน S3 (ถ้ามี)
- EventBridge trigger เพื่อให้ Lambda ทำหน้าที่ในการ แจ้งเตือนให้ผู้ใช้
- หากมีการแจ้งเตือน ระบบจะใช้ SNS เพื่อส่งข้อความไปยังผู้ใช้หรือผู้เกี่ยวข้อง

อธิบายขั้นตอนหลักที่สำคัญในการสร้างและตั้งค่า

1. การนำ Static Website Hosting ขึ้น S3
  - 1.
2. การสร้าง Lambda นำโค้ด Backend Logic ขึ้น Serveless
  1. ทำการสร้าง Lambda ขึ้นมาด้วย Node.js 22.x และเลือก Execution role เป็น LabRole
  2. นำโค้ดที่ต้องการที่อยู่ในรูป .zip กดปุ่ม Upload from อัปโหลดขึ้น Lambda
  3. จากนั้นไปที่ runtime setting กด edit
  4. ตรง Handler ใส่ src/server.handler จากนั้น Save
3. การสร้างและตั้งค่า EventBridge สำหรับการเรียกใช้งาน Lambda
  1. ไปที่แถบ Schedules ผังซ้าย
  2. Create Schedule ตั้งชื่อ แล้วเลือก Recurring Schedule เพื่อตั้งเวลาที่จะ trigger เสร็จแล้วกด next
  3. กดเลือก Lambda invoke จากนั้นเลือก Lambda function ที่เราต้องการใช้
  4. เลือก Execution role เป็น LabRole
4. การตั้งค่า S3 bucket สำหรับเก็บรูปภาพจากหน้าเว็บไซต์
  1. สร้าง S3 bucket
  2. เข้าไปที่ Permission ของ S3 ส่วนของ cors ตั้งค่าให้ AllowMethod PUT, POST, GET
5. การสร้าง Dynamodb
  1. ตั้งชื่อ table แล้วทุกอย่างคงไว้ที่ default
6. การเชื่อมต่อ API Gateway เพื่อเป็นช่องทางในการเรียก Lambda
  1. สร้าง API Gateway เลือก http จากนั้นตั้งชื่อและกด Add integration แล้วเลือก Lambda backend ของเรา
  2. จากนั้นตั้ง method ANY Resource path เป็น /{proxy+} เลือก Lambda backend
  3. create
7. การสร้าง AWS SNS
  1. สร้าง topic ตั้งชื่อ topic

การแก้ไข Lambda backend เชื่อมต่อกับ AWS SNS และ Dynamodb

ต้องเข้าไปแก้ไขให้ตรงกันในซอร์สโค้ด



ผลการทดสอบการทำงานของระบบโซลูชันตามมุมมองของผู้ใช้ (end-to-end) ทั้งหมด 3 กรณี

### 1. การลงทะเบียนผู้ใช้งาน

- Input: ชื่อผู้ใช้, อีเมล, เบอร์โทรศัพท์, ไอดีไลน์(ถ้ามี), รหัสผ่าน, ยืนยันรหัสผ่าน, รูปภาพ
- Output: ระบบจะต้องพาผู้ใช้ไปที่หน้าลงชื่อเข้าใช้

### 2. ลงชื่อเข้าใช้

- Input: ชื่อผู้ใช้, รหัสผ่าน
- Output: ระบบจะต้องพาผู้ใช้ไปแสดงที่หน้าจัดการรายการอาหาร(หน้าหลัก)

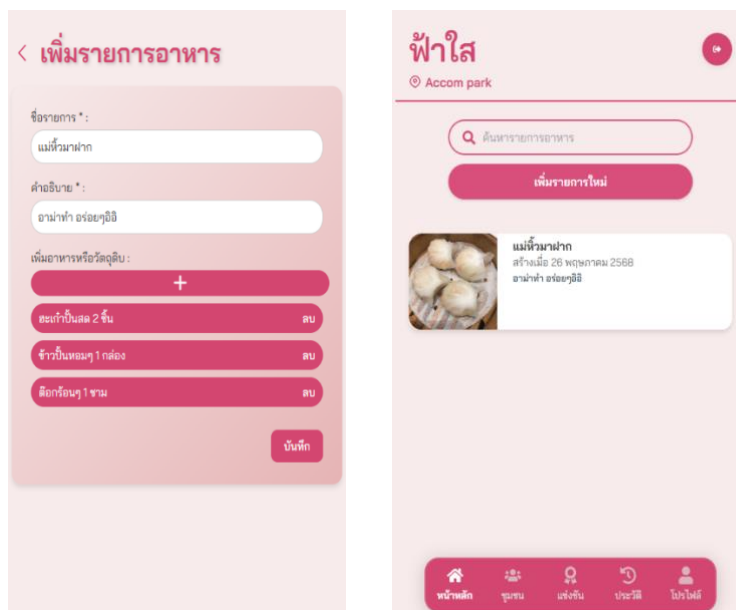
### 3. หน้าจัดการอาหาร(หน้าหลัก)

- a. Input: -
- b. Output: แสดงรายการอาหารทั้งหมดที่ผู้ใช้มี



#### 4. หน้าเพิ่มรายการอาหาร

- a. Input: ชื่อรายการอาหาร, คำอธิบาย, ข้อมูลอาหาร
- b. Output: รายการอาหารสร้างซึ่งต้องแสดงชื่ออาหาร วันที่สร้าง และคำอธิบาย



## 5. หน้าเพิ่มอาหาร

- Input: ชื่ออาหาร, วันหมดอายุ, ปริมาณ, หน่วย, หมวดหมู่, รูปภาพ
- Output: ชื่ออาหาร วันที่หมดอายุ ปริมาณ หน่วย และ รูปภาพ

The image shows two screenshots of a mobile application interface for adding food items. The left screenshot is titled 'เพิ่มอาหาร' (Add Food) and displays a form with the following fields: 'ชื่อ:' (Name) with the value 'ข้าวปั้นหอมๆ', 'วันหมดอายุ:' (Expiration Date) with the value '30/05/2025', 'ปริมาณ:' (Quantity) with the value '1', 'หน่วย:' (Unit) with the value 'กล่อง' (Box), and 'หมวดหมู่:' (Category) with a dropdown menu showing 'ของว่าง' (Snack). Below the form is a button labeled 'ถ่ายหรืออัปโหลดรูปภาพ' (Take or Upload Photo) and a photo of a food item. The right screenshot is titled 'แม่หิ้วมาฝาก' (Mom brought to share) and shows a list of food items. The first item is 'สีก้อนๆ' (Sliced pieces) with a date of '08 มิถุนายน 2568', a quantity of '1 ขาม' (1 piece), and a status of 'หมดแล้ว' (Out of stock). The second item is 'ข้าวปั้นหอมๆ' (Fragrant rice balls) with a date of '30 พฤษภาคม 2568', a quantity of '1 กล่อง' (1 box), and a status of 'หมดแล้ว' (Out of stock). Both screenshots have a bottom navigation bar with icons for 'หน้าหลัก' (Home), 'ชุมชน' (Community), 'แจ้งฉัน' (Notify me), 'ประวัติ' (History), and 'โปรไฟล์' (Profile).

6. หน้าแก้ไขและลบรายการอาหาร

- a. Input: ชื่อรายการอาหาร, คำอธิบาย
- b. Output: รายการอาหารที่ถูกอัปเดตใหม่



- c. Input: -
- d. Output: รายการอาหารหายไป

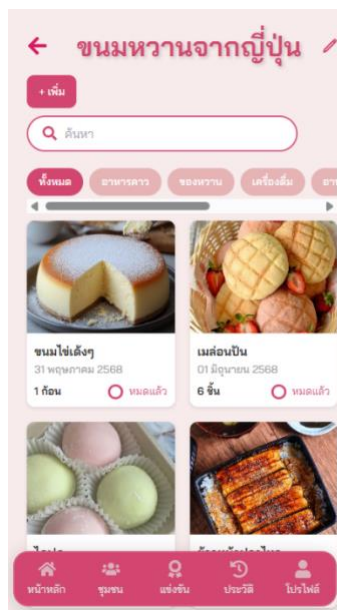


7. หน้าแสดงอาหารทั้งหมดในรายการอาหาร

7.1 แสดงอาหารทั้งหมดในรายการอาหาร

- a. Input: -

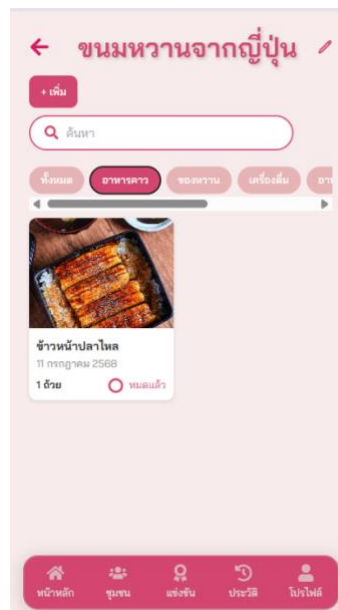
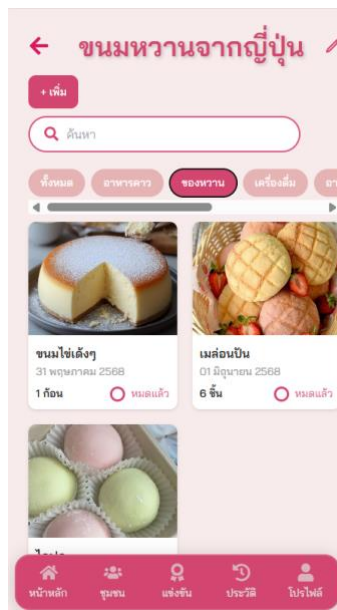
- b. Output: ชื่ออาหาร วันที่หมดอายุ ปริมาณ หน่วย และ รูปภาพของอาหารแต่ละอัน



## 7.2 กรองประเภทของอาหารได้

Input: ประเภทของอาหาร

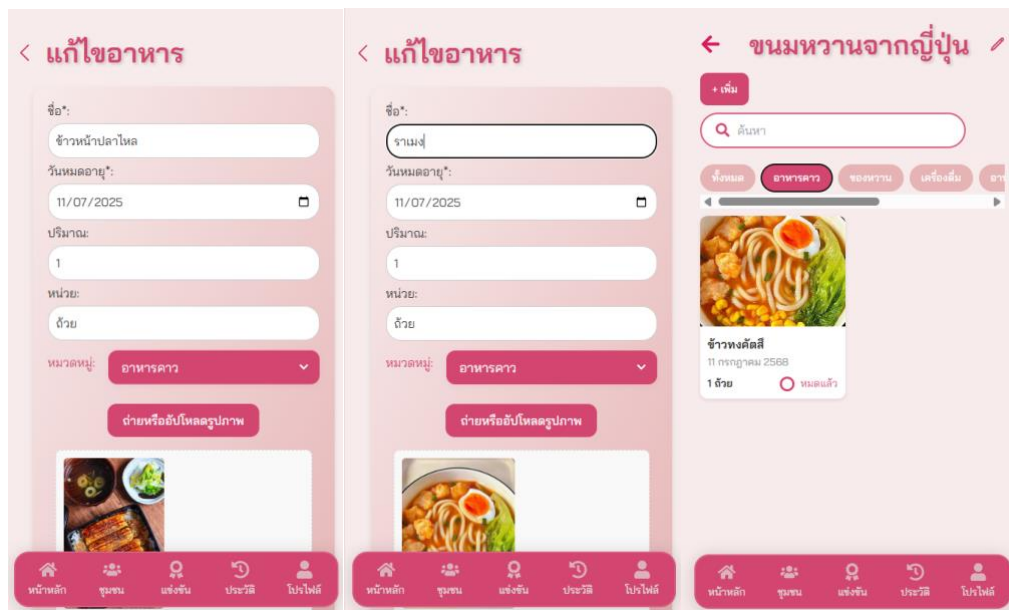
Output: แสดงอาหารตามประเภทที่ต้องการ



## 8. หน้าแก้ไขและลบอาหาร

### 8.1 แก้ไขอาหาร

- a. Input: ชื่ออาหาร, วันหมดอายุ, ปริมาณ, หน่วย, หมวดหมู่, รูปภาพที่ต้องการแก้ไข
- b. Output: แสดงข้อมูลอาหารตามที่ถูกแก้ไข

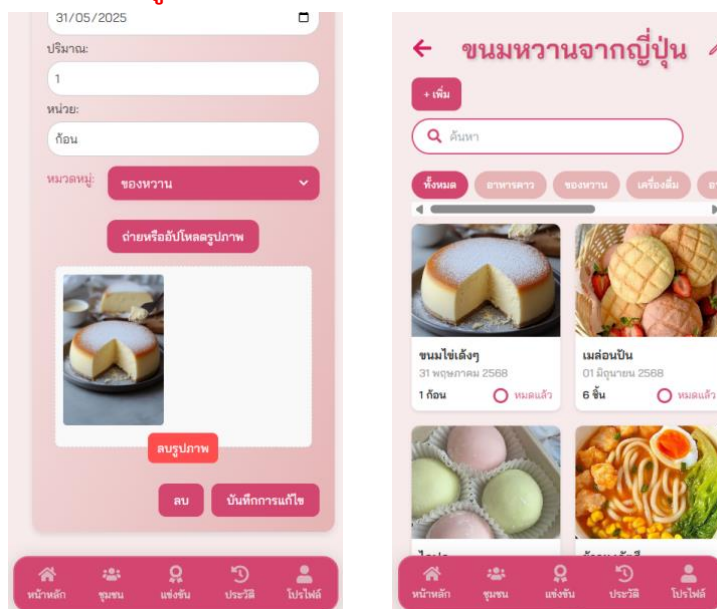


## 8.1 ลบอาหาร

c. Input: -

d. Output: อาหารหายไป

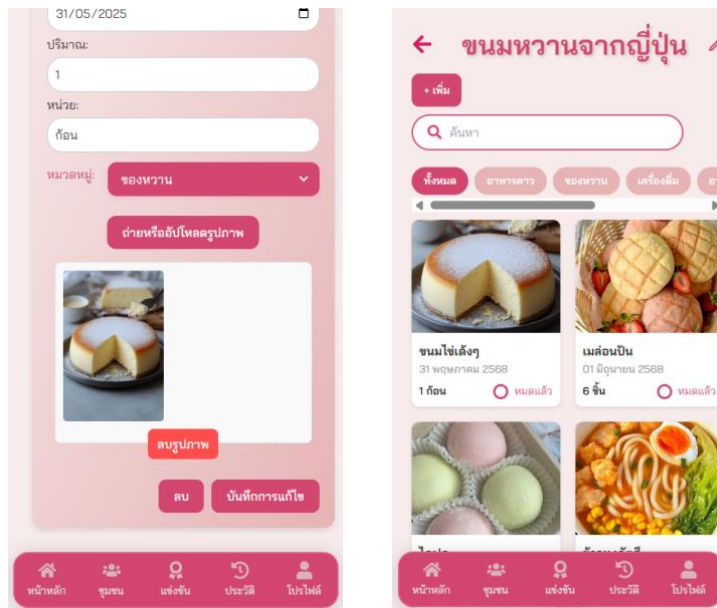
ERROR: ข้อมูลอาหารไม่หายไป ขึ้น 404 ERROR BAD REQUEST



## 9. จัดการอาหารที่ใช้แล้ว

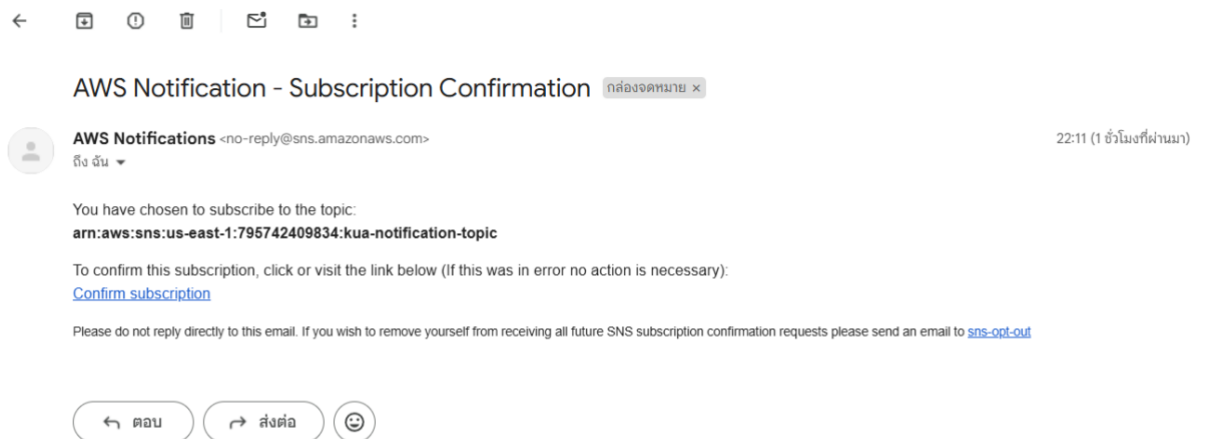
a. Input: กดว่าหมดแล้ว

b. Output: อาหารหายไป



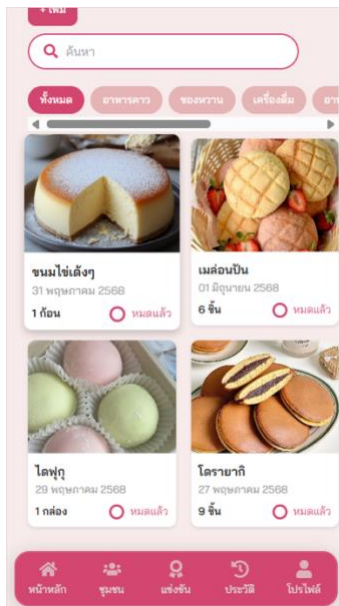
10. การแจ้งเตือน

10.1 เมื่อ user ลงทะเบียนเข้าสู่ระบบ AWS จะให้ส่ง subscription



Input: อาหารที่จะหมดอายุพ่วงนี้

Output: อาหารที่จะหมดอายุใน 1 วันและ 3 วัน ได้รับการแจ้งเตือน



Kua Notification - อาหารของคุณใกล้หมดอายุแล้ว! กล้งจดหมาย x



**AWS Notifications** <no-reply@sns.amazonaws.com>

ถึง ฉัน

23:16 (2 นาทีที่ผ่านมา)



ฟ้าใสมาก มีอาหารที่จะหมดอายุหลายรายการ

📧 รายการ: แม่หัวมาฝาก  
- สะเก๋ปิ่นสด (หมดอายุวันที่ 28/05/2568)

📧 รายการ: ขนมหวานจากญี่ปุ่น  
- โตฟูคุ (หมดอายุวันที่ 29/05/2568)  
- โศรยาภิ (หมดอายุวันที่ 27/05/2568)

--  
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:795742409834:kua-notification-topic:c7811aff-dbe2-432f-8589-915e6c679118&Endpoint=neon.poolsawat@gmail.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

## 11. หน้าโปรไฟล์

### 11.1 แสดงโปรไฟล์

a. Input: -



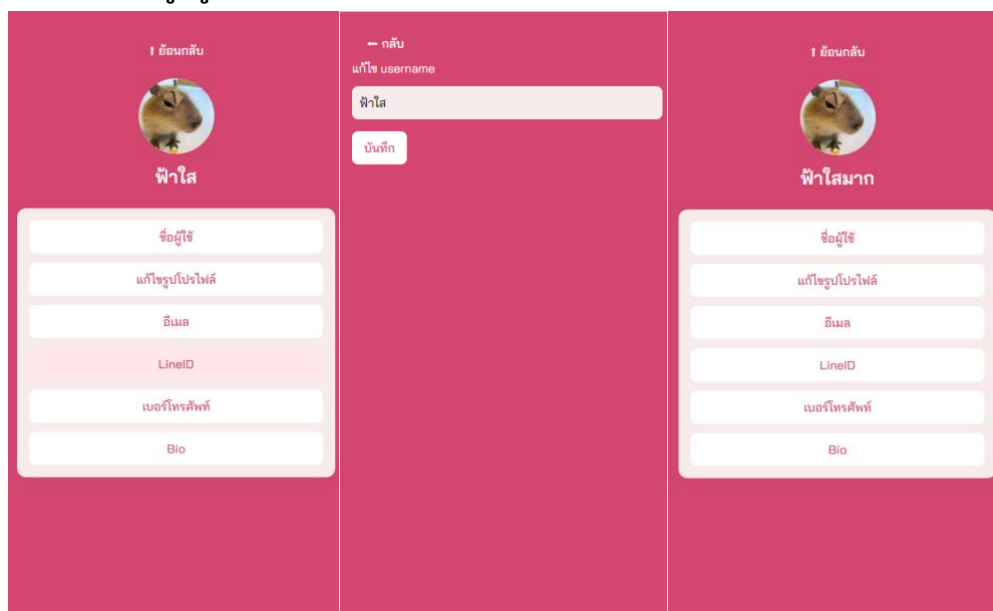
b. Output: ข้อมูลโปรไฟล์ของผู้ใช้



## 11.2 แก้ไขโปรไฟล์

c. Input: ข้อมูลที่ต้องการแก้ไข

d. Output: ข้อมูลถูกอัปเดต



## 12. หน้าชุมชน

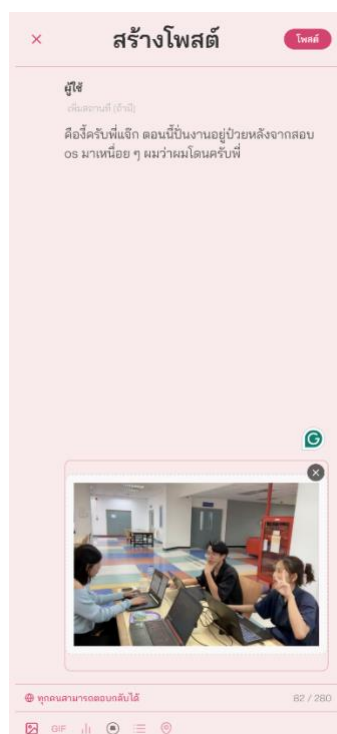
### 12.1. แสดงหน้าชุมชน หมวดสำหรับคุณ

- Input –
- Output หน้าชุมชนหมวดสำหรับคุณ



### 12.2. กดไอคอนเพิ่ม

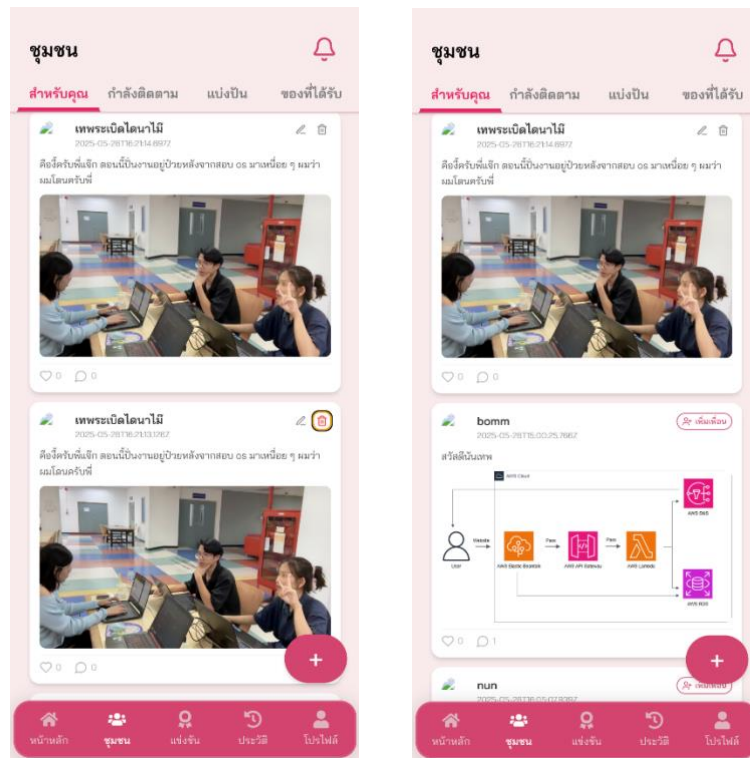
- Input: caption , รูปภาพ
- b. หน้าสำหรับการสร้างโพสต์



### 12.3. กดไอคอนลบ

a. Input -

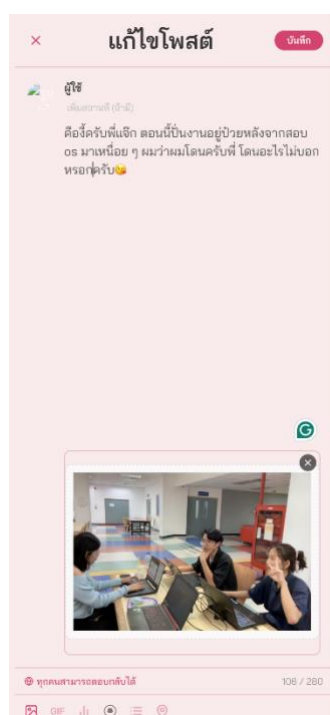
b. Output - ลบโพสต์



### 12.4. แก้ไขโพสต์

a. Input - ข้อความ หรือ รูปภาพที่ต้องการโพสต์

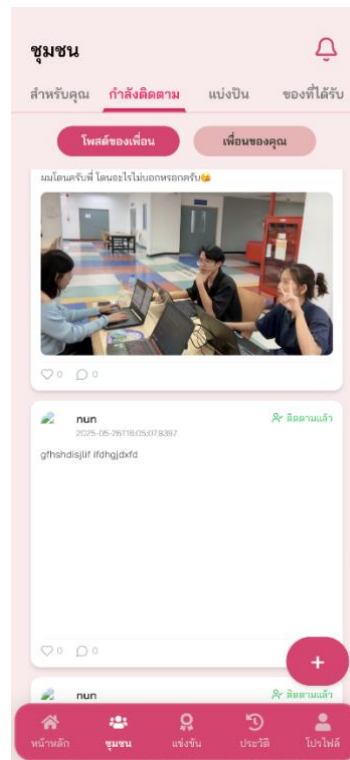
b. Output - ข้อมูลโพสต์ถูกอัปเดต



## 12.5. แสดงหน้าชุมชน หมวดกำลังติดตาม (หน้าโพสต์ของเพื่อน)

a. Input -

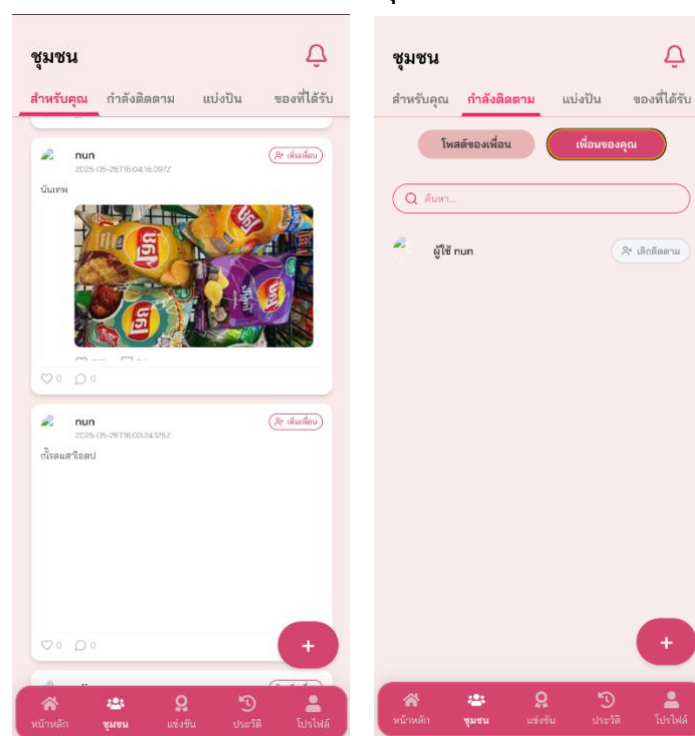
b. Output - โพสต์ต่างๆของเพื่อนคุณ



## 12.6. แสดงหน้าชุมชน หมวดกำลังติดตาม (เพื่อนของคุณ)

a. Input -

b. Output - แสดงรายชื่อเพื่อนของคุณและสามารถเลิกติดตามได้



บทวิเคราะห์ข้อดีและข้อเสียของระบบตามกรอบ AWS Well-Architected Framework

ระบบ "Kuaglang" พัฒนาด้วยบริการของ AWS ในรูปแบบ serverless เพื่อรองรับการขยายตัว  
ประหยัดต้นทุน และจัดการระบบอย่างมีประสิทธิภาพ การวิเคราะห์นี้อ้างอิงตาม AWS Well-Architected  
Framework

## 1. การปฏิบัติงาน

ข้อดี :

- ใช้ AWS Lambda ที่ไม่ต้องดูแลเซิร์ฟเวอร์และปรับขนาดอัตโนมัติตามความต้องการ
- ใช้สถาปัตยกรรมแบบไมโครเซอร์วิส ช่วยให้ดูแลและจัดการโค้ดแต่ละส่วนได้แยกกัน

ข้อเสีย :

- ขาดระบบ CI/CD ทำให้ต้อง deploy โค้ดด้วยตนเองทุกครั้ง

## 2. ความปลอดภัย (Security)

ข้อดี :

- ทุกบริการเชื่อมต่อผ่าน IAM Role ทำให้ควบคุมสิทธิ์ได้อย่างแม่นยำ

ข้อเสีย :

- S3 และ DynamoDB ยังไม่ได้กำหนดการเข้ารหัสและนโยบายความปลอดภัยอย่างครบถ้วน

## 3. ความน่าเชื่อถือ (Reliability)

ข้อดี :

- บริการหลักอย่าง Lambda และ DynamoDB มีความพร้อมใช้งานสูงโดยพื้นฐาน

## 4. ประสิทธิภาพการทำงาน (Performance Efficiency)

ข้อดี :

- Lambda ปรับขนาดอัตโนมัติและไม่มีค่าใช้จ่ายเมื่อไม่มีการใช้งาน
- DynamoDB เหมาะกับงานที่มีการใช้งานสูงและไม่ซับซ้อน

## 5. ค่าใช้จ่าย(Cost Optimization)

ข้อดี :

- ใช้ Lambda และ DynamoDB แบบจ่ายตามการใช้งาน ประหยัดในช่วงที่มีผู้ใช้น้อย
- ไม่มีค่าใช้จ่ายจากเซิร์ฟเวอร์ที่ต้องเปิดตลอดเวลา

ข้อเสีย :

- การไม่กำหนดเวลาหมดอายุหรือวงจรชีวิตใน S3 อาจทำให้เสียค่าใช้จ่ายเกินจำเป็น

## 6. ความยั่งยืน (Sustainability)

ข้อดี :

- ใช้สถาปัตยกรรมแบบ serverless ช่วยประหยัดพลังงานและทรัพยากร

ข้อเสีย :

- ขาดระบบจัดการทรัพยากรที่หมดอายุโดยอัตโนมัติ เช่น การลบไฟล์ที่ไม่ใช้ใน S3

ลิงก์คลิปวิดีโอเดโมและลิงก์ github repository

Link : คลิปวิดีโอเดโมการทำงานของระบบ ( [Demo Drive](#) )

Link : Git Repository ( [Github repository](#) )

## สรุปผลการดำเนินโครงการ

โครงการระบบ Kua Glang ได้รับการพัฒนาและออกแบบโดยใช้สถาปัตยกรรมแบบไร้เซิร์ฟเวอร์บนแพลตฟอร์ม AWS ซึ่งสามารถตอบโจทย์ปัญหาการจัดการอาหารเหลือในครัวเรือนและชุมชนได้อย่างมีประสิทธิภาพ ระบบช่วยให้ผู้ใช้งานสามารถเพิ่มข้อมูลอาหารที่ต้องการแบ่งปัน ติดตามวันหมดอายุ และรับการแจ้งเตือนอย่างเหมาะสม รวมถึงรองรับการแจ้งเตือนผ่านช่องทางอีเมลหรือ ทำให้ผู้ใช้มีเครื่องมือที่ช่วยลดปริมาณอาหารเหลือทิ้งได้จริง

ในกระบวนการพัฒนาระบบ ได้มีการทดสอบการทำงานของแต่ละองค์ประกอบและการทำงานร่วมกัน พบว่าระบบสามารถรองรับคำขอและจัดการข้อมูลได้อย่างถูกต้อง นอกจากนี้ การเลือกใช้บริการ AWS ทำให้ระบบสามารถขยายตัวได้ตามจำนวนผู้ใช้ โดยไม่ต้องลงทุนโครงสร้างพื้นฐานเพิ่มเติม

## แนวทางพัฒนาต่อในอนาคต

1. เพิ่มระบบยืนยันตัวตนและสิทธิ์ผู้ใช้งาน (Authentication & Authorization) เพื่อเพิ่มความปลอดภัยในการเข้าถึงข้อมูลและป้องกันการใช้งานที่ไม่เหมาะสม ควรนำระบบจัดการผู้ใช้ เช่น Amazon Cognito มาใช้งาน
2. พัฒนาอินเทอร์เฟซให้ใช้งานง่ายและเพิ่มลูกเล่นของอินเทอร์เฟซให้ดูมีความเป็นแอปคอมมูนิตีมากขึ้น รองรับอุปกรณ์หลากหลายปรับปรุงหน้าจอผู้ใช้ให้รองรับการใช้งานบนโทรศัพท์มือถือและแท็บเล็ต รวมทั้งเพิ่มฟีเจอร์แจ้งเตือนแบบ push notification
3. เสริมระบบแจ้งเตือนและติดตามสถานะแบบ real-time ใช้บริการ AWS เช่น Amazon SNS และ AWS AppSync เพื่อให้การแจ้งเตือนรวดเร็วและแม่นยำมากขึ้น
4. เพิ่มระบบวิเคราะห์ข้อมูลและรายงานเชิงลึกพัฒนาระบบวิเคราะห์แนวโน้มการใช้และการลดของเสีย เพื่อช่วยให้ผู้ดูแลระบบสามารถวางแผนพัฒนาและสนับสนุนผู้ใช้ได้ดียิ่งขึ้น
5. เพิ่มความเสถียรของระบบมากขึ้นการจัดการ error และ test case ที่มากขึ้น



6. เพิ่มหน้าประวัติของผู้ใช้โดยจะมีหน้าแสดงรายการอาหารทั้ง หน้าประวัติการแบ่งปัน และหน้าประวัติที่ได้รับของผู้ใช้ เพื่อเก็บประวัติการใช้งานของผู้ใช้



7. หน้าแบ่งปันกับของที่ได้รับ พัฒนาต่อยอดไปในทางด้านการอำนวยความสะดวกให้ผู้ใช้ ถ้ากลุ่มรับอาหารของผู้ใช้นั้นจะเข้ามาที่หน้าจัดการอาหารของเราอัตโนมัติ

