

## บทที่ 2

### ทฤษฎีและระบบงานที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 Flutter

Flutter คือ Framework ที่ใช้สร้าง UI สำหรับ mobile application ที่สามารถทำงานข้ามแพลตฟอร์มได้ทั้ง iOS และ Android ในเวลาเดียวกัน โดยภาษาที่ใช้ใน Flutter นั้นจะเป็นภาษา dart ซึ่งถูกพัฒนาโดย Google และที่สำคัญคือเป็น open source ที่สามารถใช้งานได้แบบฟรี ๆ อีกด้วย ตัวอย่าง syntax ของภาษา dart ที่ใช้ใน Flutter ซึ่งจะมีความคล้ายกับภาษา Java เนื่องจาก dart เป็นภาษาที่รองรับ OOP และมีแนวคิดของ class และ inheritance เช่นเดียวกับภาษา Java นั่นเอง

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(new MaterialApp(
5     home: new MyApp(),
6   ));
7 }
8
9 class MyApp extends StatelessWidget {
10  @override
11  Widget build(BuildContext context) {
12    return new Scaffold(
13      appBar: new AppBar(
14        title: new Text("Example App"),
15        backgroundColor: Colors.blue,
16      ),
17      backgroundColor: Colors.blue,
18      body: new Center(
19        child: new Column(
20          mainAxisAlignment: MainAxisAlignment.center,
21          children: <Widget>[
22            new Icon(Icons.favorite, color: Colors.redAccent, size: 200.0,
23              ),
24            ],
25          ),
26        ),
27      );
28  }
29 }
```

flutter\_example hosted with ❤️ by GitHub [view raw](#)

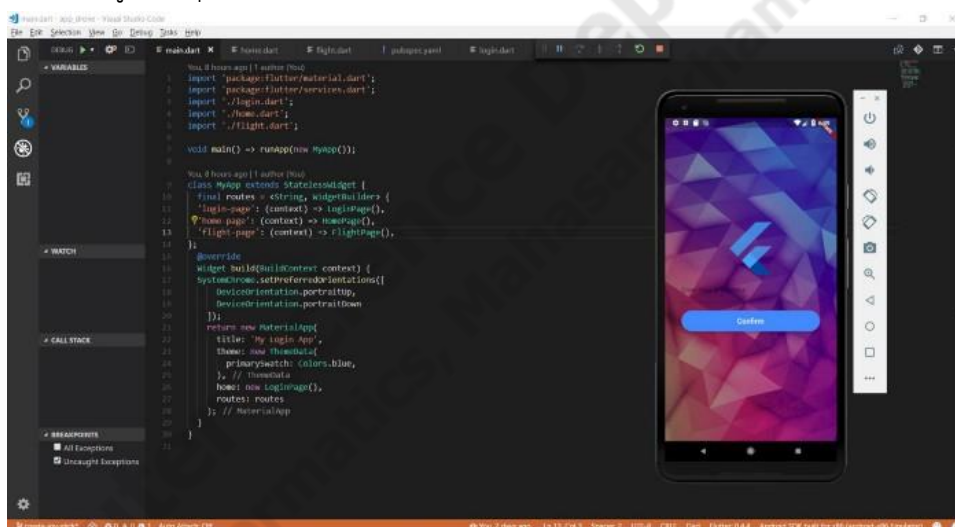
ภาพประกอบที่ 2.1 ตัวอย่างการ syntax ของภาษา dart

ซึ่งหากสังเกตจากตัวอย่างด้านบน จะเห็นว่า Flutter นั้นจะมี Widget พื้นฐานมาให้ เพื่อให้การออกแบบ UI มีความง่าย และสะดวกยิ่งขึ้น โดย Widget พื้นฐานของ Flutter หลัก ๆ จะมีอยู่ 2 ชนิดคือ StatelessWidget และ StatefulWidget โดยที่ StatelessWidget จะใช้สร้าง Widget ที่ไม่มีการจัดการสถานะการทำงานใด ๆ เช่น การแสดงข้อความ, Icon หรือรูปภาพที่ไม่มี animation เข้ามาเกี่ยวข้อง เป็น

ต้น ส่วน StatefulWidget จะใช้สร้าง Widget ที่มีการจัดการสถานะการทำงานต่าง ๆ เช่น การสร้าง Icon ที่มีการใส่ animation ให้สามารถขยับไปมาได้, ปุ่มกดต่าง ๆ บนหน้า UI เป็นต้น

### 2.1.1.1 จุดเด่นของ flutter คืออะไร

จุดเด่นหลัก ๆ ของ Flutter คือ ระบบ Hot Reload โดยเมื่อมีการทดสอบ, การสร้าง, การ add features หรือการกระทำต่าง ๆ กับ UI จะต้องมีการ reload เพื่อให้หน้า UI update ซึ่งระบบ Hot Reload จะเข้ามาช่วยในส่วนของการ reload โดยจุดเด่นของระบบนี้คือการย่นระยะเวลาที่ใช้ในการ reload ให้เหลือเพียงเสี้ยววินาทีเท่านั้น ทำให้การพัฒนา UI ของ application มีความรวดเร็วขึ้นอย่างมาก และยังมีจุดเด่นอื่น ๆ ที่ช่วยให้การพัฒนาเป็นไปได้ง่ายขึ้นไม่ว่าจะเป็น Build-In ที่ช่วยในการออกแบบ UI ให้มีความสวยงามยิ่งขึ้นอย่าง Material Design และ Cupertino (iOS-flavor), มี Framework ที่ช่วยให้การทำ animation ต่าง ๆ หรือ gesture ของ UI เป็นเรื่องง่ายยิ่งขึ้น และยังสามารถใช้งานร่วมกับ IDE ที่กำลังเป็นที่นิยมอยู่ในปัจจุบันอย่าง VS Code และ Android Studio ได้อีกด้วย



ภาพประกอบที่ 2.2 ตัวอย่างแสดงการเขียนบน VS Code

### 2.1.1.2 ข้อเสียของ Flutter

จากที่กล่าวมาว่าจุดเด่นต่าง ๆ ของ Flutter มีอะไรบ้าง คราวนี้เราจะมากล่าวถึงข้อเสียกันบ้าง โดยข้อเสียหลัก ๆ ที่พบคือ การใช้ภาษา dart ในการเขียน ซึ่งคนส่วนใหญ่อาจจะยังไม่คุ้นเคยกับ syntax ของภาษานี้สักเท่าไร ประกอบกับ community ยังเล็กอยู่เนื่องจาก Flutter ยังเปิดตัวมาได้ไม่นานนักเมื่อเทียบกับ Framework ตัวอื่น ๆ อย่าง React Native ที่มี community ค่อนข้างใหญ่ จึงทำให้ document ต่าง ๆ ยังไม่เยอะเท่าที่ควร ทำให้เวลามีปัญหาเกี่ยวกับการใช้งานอาจจะต้องมานั่งมหาวิธีแก้กันนานพอสมควร (Hizoka, 2561)

## 2.1.2 Node.js

Node.js คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript สรุปรวมๆ NodeJS ก็คือ Platform ตัวหนึ่งเขียนด้วย JavaScript สำหรับเป็น Web Server

### 2.1.2.1 เหตุผลที่ได้รับความนิยมหลักๆคือ

- เขียนโค้ดเข้าใจได้ง่าย
  - มี Library ฟรีมากมาย
  - ใช้ทรัพยากรน้อย
  - เรียนรู้ได้เร็วสำหรับโปรแกรมเมอร์ทุก ๆ ระดับ
- (Association, 2559)

## 2.1.3 ฐานข้อมูล (Database)

Database หรือ ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล

### 2.1.3.1 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่าง ผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (data base management system)มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

### 2.1.3.2 โครงสร้างแฟ้มข้อมูล

ส่วนประกอบแฟ้มข้อมูล (File) ระเบียน (Record) และ เขตข้อมูล (Field) และถูกจัดการด้วยระบบเดียวกัน โปรแกรมคอมพิวเตอร์จะเข้าไปดึงข้อมูลที่ต้องการได้ อย่างรวดเร็ว ซึ่งอาจเปรียบฐานข้อมูลเสมือนเป็น electronic filing system

-บิต (bit) ย่อมาจาก Binary Digit ข้อมูลในคอมพิวเตอร์ 1 บิต จะแสดงได้ 2 สถานะคือ 0 หรือ 1 การเก็บข้อมูลต่าง ๆ ได้จะต้องนำ บิต หลายๆ บิต มาเรียงต่อกัน เช่นนำ 8 บิต มาเรียงเป็น 1 ชุด เรียกว่า 1 ไบต์ เช่น 10100001 หมายถึง ก 10100010 หมายถึง ข

-เมื่อนำ ไบต์ (byte) หลายๆ ไบต์ มาเรียงต่อกัน เรียกว่า เขตข้อมูล (field) เช่น Name ใช้เก็บชื่อ LastName ใช้เก็บนามสกุล เป็นต้น

-เมื่อนำเขตข้อมูล หลายๆ เขตข้อมูล มาเรียงต่อกัน เรียกว่า ระเบียบ (record) เช่น ระเบียบ ที่ 1 เก็บ ชื่อ นามสกุล วันเดือนปีเกิด ของ นักเรียนคนที่ 1 เป็นต้น

-การเก็บระเบียบหลายๆระเบียบ รวมกัน เรียกว่า แฟ้มข้อมูล (File) เช่น แฟ้มข้อมูล นักเรียน จะเก็บ ชื่อ นามสกุล วันเดือนปีเกิด ของนักเรียน จำนวน 500 คน เป็นต้น

-การจัดเก็บ แฟ้มข้อมูล หลายๆ แฟ้มข้อมูล ไว้ภายใต้ระบบเดียวกัน เรียกว่า ฐานข้อมูล หรือ Database เช่น เก็บ แฟ้มข้อมูล นักเรียน อาจารย์ วิชาที่เปิดสอน เป็นต้น

### 2.1.3.3 วิวัฒนาการของ database

Database ถูกพัฒนาขึ้นมาตั้งแต่ปี 1960 เริ่มต้นจาก hierarchical และ network databases จนมาถึงปี 1980 มีการนำเอา object-oriented-databases (OODBMS) มาใช้งาน ซึ่งเป็นพื้นฐานของระบบ relation database ที่ใช้กันอยู่ในปัจจุบันนี้

ในอีกมุมหนึ่ง เราสามารถจัดแบ่งประเภทของ database ตามรูปแบบของชนิดข้อมูลได้ เช่น ตัวเลข ,ตัวอักษร หรือ รูปภาพ บางครั้งก็อาจจะแบ่งตามความนิยมของ relational database เช่น distributed database, cloud database หรือ NoSQL database.

#### -Relational database

Relational database ถูกคิดค้นขึ้นโดย E.F. Codd (IBM) ในปี 1970 เริ่มต้นสร้างขึ้นมาจากกลุ่มของ table ที่มีข้อมูลภายในโดยแบ่งออกเป็นตามประเภทที่ตั้งไว้ แต่ละ table จะมีอย่างน้อย 1 ชนิดของแต่ละ column และแต่ละ row จะมีข้อมูลตามที่ชนิดที่ columns ได้กำหนดไว้

Standard Query Language (SQL) เป็นมาตรฐานที่ผู้ใช้งาน และ ระบบอื่น ๆ ไว้เชื่อมต่อกับ relational database ซึ่งง่ายต่อการเพิ่มข้อมูลเข้าไป โดยไม่กระทบต่อโปรแกรมอื่นที่ใช้งานร่วมกันอยู่

#### -Distributed database

Distributed database เป็น ฐานข้อมูลที่ถูกเก็บกระจายออกไปหลายๆที่ โดยอาศัยกระบวนการแจกจ่าย และ สำรองข้อมูล ผ่านทางระบบ network ซึ่งมีอยู่ด้วยกัน 2 แบบคือ

-homogeneous – ระบบทั้งหมดทุกที่ต้องเป็น OS และ database ชนิดเดียวกัน

-heterogeneous – ระบบทั้งหมดจะเหมือนหรือต่างกันได้ในแต่ละที่

#### -Cloud database

cloud database คือ ระบบฐานข้อมูลที่ตอบโจทย์การทำงานในยุคนี้ ก็เป็นไปได้มากที่หัวหน้าของคุณจะมีนโยบายนำระบบฐานข้อมูลแบบนี้มาใช้ในองค์กรบ้าง และถ้าคุณรู้อยู่แล้วว่าถ้าระบบนี้ถูกนำมาใช้ในบริษัท คุณจะต้องรับหน้าที่ศึกษาดูแล แล้วแบบนี้คุณจะต้องเตรียมตัวเรียนรู้พัฒนาทักษะเรื่องใดบ้างถึงจะสามารถดูแลเรื่อง cloud database ได้อย่างคล่องตัว

-NoSQL database

NoSQL database ถูกใช้ในรูปแบบ ที่เป็นการกระจายของข้อมูล จึงมีประสิทธิภาพสูงสำหรับข้อมูลขนาดใหญ่ (big data) เพราะ relational database ไม่ถูกออกมาให้รองรับข้อมูลขนาดใหญ่ จึงนิยมใช้กับการวิเคราะห์ข้อมูลขนาดใหญ่ที่ไม่ค่อยมีรูปแบบตายตัว

#### 2.1.3.4 การ access database

มีด้วยกัน 2 แบบคือ

- Database management system (DBMS) เป็น software ที่ควบคุมและบริหารข้อมูลภายในฐานข้อมูล
- Relational database management system (RDBMS) ถูกพัฒนาขึ้นในปี 1970 เพื่อเข้าถึงฐานข้อมูลแบบ relational และยังคงได้รับความนิยมจนถึงปัจจุบัน

#### 2.1.3.5 ประโยชน์ของฐานข้อมูล

- ลดการเก็บข้อมูลที่ซ้ำซ้อน ข้อมูลบางชุดที่อยู่ในรูปของแฟ้มข้อมูลอาจมีปรากฏอยู่หลาย ๆ แห่ง เพราะมีผู้ใช้ข้อมูลชุดนี้หลายคน เมื่อใช้ระบบฐานข้อมูลแล้วจะช่วยให้ความซ้ำซ้อนของข้อมูลลดน้อยลง
- รักษาความถูกต้องของข้อมูล เนื่องจากฐานข้อมูลมีเพียงฐานข้อมูลเดียว ในกรณีที่มีข้อมูลชุดเดียวกันปรากฏอยู่หลายแห่งในฐานข้อมูล ข้อมูลเหล่านี้จะต้องตรงกัน ถ้ามีการแก้ไขข้อมูลนี้ทุก ๆ แห่งที่ข้อมูลปรากฏอยู่จะแก้ไขให้ถูกต้องตามกันหมดโดยอัตโนมัติด้วยระบบจัดการฐานข้อมูล
- การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้สะดวก การป้องกันและรักษาความปลอดภัยกับข้อมูลระบบฐานข้อมูลจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้นซึ่งก่อให้เกิดความปลอดภัย (security) ของข้อมูลด้วย

#### 2.1.3.6 ข้อดีของฐานข้อมูล

การจัดเก็บข้อมูลเป็นฐานข้อมูลได้เปรียบกว่าการจัดเก็บข้อมูลแบบแฟ้มข้อมูล ดังนี้

- หลีกเลี่ยงความขัดแย้งของข้อมูล การจัดเก็บข้อมูลแบบแฟ้มข้อมูล โดยข้อมูลเรื่องเดียวกันอาจมีอยู่หลายแฟ้มข้อมูล ซึ่งก่อให้เกิดความขัดแย้งของข้อมูลได้ ( Inconsistency )
- สามารถใช้ข้อมูลร่วมกันได้ ฐานข้อมูลเป็นการจัดเก็บข้อมูลรวมไว้ด้วยกัน เมื่อผู้ใช้ต้องการข้อมูลจากฐานข้อมูล ซึ่งเป็นข้อมูลที่มาจากแฟ้มข้อมูลที่แตกต่างกันจะทำได้ง่าย
- สามารถลดความซ้ำซ้อนของข้อมูล การจัดเก็บข้อมูลในลักษณะแฟ้มข้อมูล อาจทำให้ข้อมูลประเภทเดียวกันถูกเก็บไว้หลาย ๆ แห่ง ทำให้เกิดความซ้ำซ้อน (Redundancy ) การนำข้อมูลมารวมเก็บไว้ในฐานข้อมูล จะช่วยลดปัญหาความซ้ำซ้อนได้
- รักษาความถูกต้อง ฐานข้อมูลบางครั้งอาจมีข้อผิดพลาดขึ้น เช่น การป้อนข้อมูลผิด ซึ่งระบบการจัดการฐานข้อมูลสามารถระบุกฎเกณฑ์เพื่อควบคุมความผิดพลาดที่อาจเกิดขึ้นได้
- สามารถกำหนดความเป็นมาตรฐานเดียวกันได้ เพราะในระบบฐานข้อมูลจะมีกลุ่มบุคคลที่คอยบริหารฐานข้อมูล กำหนดมาตรฐานต่าง ๆ ในการจัดเก็บข้อมูลในลักษณะเดียวกัน

-สามารถกำหนดระบบความปลอดภัยของข้อมูลได้ ผู้บริหารระบบฐานข้อมูลสามารถกำหนดการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนให้แตกต่างกันตามหน้าที่ ความรับผิดชอบได้ง่าย

-ความเป็นอิสระของข้อมูลและโปรแกรม โปรแกรมที่ใช้ในแต่ละแฟ้มข้อมูลจะมีความสัมพันธ์กับแฟ้มข้อมูลโดยตรงถ้าหากมีการแก้ไขเปลี่ยนแปลงโครงสร้างข้อมูลก็ทำการแก้ไขโปรแกรมนั้น ๆ

#### 2.1.3.7 ข้อเสียของฐานข้อมูล

การเก็บข้อมูลรวมเป็นฐานข้อมูลมีข้อเสีย ดังนี้คือ

-มีต้นทุนสูง ระบบฐานข้อมูลก่อให้เกิดต้นทุนสูง เช่น ซอฟต์แวร์ที่ใช้ในการจัดการระบบฐานข้อมูล บุคลากร ต้นทุนในการปฏิบัติงาน และ ฮาร์ดแวร์ เป็นต้น

-มีความซับซ้อน การเริ่มใช้ระบบฐานข้อมูล อาจก่อให้เกิดความซับซ้อนได้ เช่น การจัดเก็บข้อมูล การออกแบบฐานข้อมูล การเขียนโปรแกรม เป็นต้น

-การเสี่ยงต่อการหยุดชะงักของระบบ เนื่องจากข้อมูลถูกจัดเก็บไว้ในลักษณะเป็นศูนย์รวม (Centralized Database System ) ความล้มเหลวของการทำงานบางส่วนในระบบอาจทำให้ระบบฐานข้อมูลทั้งระบบหยุดชะงักได้ (toppiiz\_spiiz, 2561)

#### 2.1.4 MySQL

MySQL คือ open source ถูกคิดค้นโดย MySQL AB ในสวีเดน และต่อมาถูก takeover โดย Sun Microsystems ในปี 2008 และก็ควบรวมกับ Oracle ในปี 2010 มี function การทำงานแบบ relation database management system (RDBMS) โดยอาศัย Structured Query Language (SQL) เป็นภาษาในสื่อสาร โดยเจ้าตัว MySQL นี้สามารถรันได้ทั้งบน Linux, UNIX และ Windows ซึ่งด้วยความหลากหลายของมันแล้ว แต่คนก็ยังคงใช้แต่กับงาน web-based ะส่วนใหญ่ เพราะฉะนั้นจึงได้มีการออกแบบเจ้าตัว MySQL ให้เป็นส่วนหนึ่งในระบบ open source enterprise stack หรือที่เรียกว่า " LAMP" (saixiii, 2560)

#### 2.1.5 Web service

##### 2.1.5.1 Web service คืออะไร

Web service (เว็บเซอร์วิส) คือการให้บริการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ในโลก internet ซึ่งถูกออกแบบมาตอนแรกเพื่อใช้ระหว่าง human-to-machine ผ่าน HTTP ต้องถูกพัฒนามาใช้กับ machine-to-machine ในรูปแบบ XML และ JSON format

ในทางเทคนิค "web service" เป็นมาตรฐานในการเชื่อมต่อ web-base application ที่วิ่งอยู่บน internet protocol โดยอาศัย XML, SOAP, WSDL และ UDDI

-XML (Extensible Markup Language) เป็น format ที่จะบอกแหล่งที่มา และ รายละเอียดข้อมูลได้

-SOAP หรือ REST เป็น protocol ที่ใช้แลกเปลี่ยนข้อมูล

-WSDL (Web Services Description Language) ใช้ในการอธิบายโครงสร้างของ service ทั้งหมด หรือ คู่มือการใช้งาน

-UDDI (Universal Description, Discovery, and Integration) เป็นส่วนกลางให้ระบบอื่นมาลงทะเบียนไว้ โดยใช้ WSDL file ทำให้ฝั่ง Client สามารถดึงเอาข้อมูลไปใช้ติดต่อกับ web service ได้ จากข้อมูลอ้างอิงหลายที่ จะพบว่ามึข้อมูลเกี่ยวกับ web service หลากหลายมาก จึงขอสรุปเป็นหัวข้อ ดังนี้

-เปิดให้ใช้งานใช้งานได้ทั้ง Internet และ Intranet

-ใช้มาตรฐาน XML format

-ไม่ได้ยึดติดกับ Operating system ชนิดใด หรือ ภาษา programming ชนิดไหน

-สามารถทำความเข้าใจคำสั่งด้วยลักษณะการทำงานแบบเดียวกับ XML

มี software จำนวนมากที่ถูกเขียนในภาษา programming ที่ต่างกัน แต่ทุกภาษาต่างก็มีความจำเป็นต้องแลกเปลี่ยนข้อมูลกับ program อื่นซึ่งตัว web service เองก็ยอมให้ทำได้ด้วยการใช้ XML tag เป็นตัวเก็บข้อมูล ซึ่งโครงสร้างข้อมูลจะถูกเก็บไว้ที่ WSDL (Web Services Description Language) ภายในจะบอกถึงรายละเอียดการทำงานของ web service ดังนี้

-ระบบหนึ่งจะทำการส่งคำสั่งไปยังอีกระบบได้อย่างไร

-parameter ไหนที่ต้องใส่ในการส่งคำสั่ง

-โครงสร้างของข้อมูล XML ต้องมีลักษณะอย่างไร

-error message ที่แสดงหมายถึงคำสั่งที่ส่งมาผิดพลาดในส่วนไหน เพื่องานต่อการแก้ไข

#### 2.1.5.2 ขั้นตอนการทำงานของ Web service

-ฝั่ง Client ทำการโหลดข้อมูลต่าง ๆ มีใช้ในการติดต่อไปยัง server เข้าสู่ SOAP message

-ส่ง SOAP message ไปยัง web service ด้วยการทำให้ HTTP POST

-ฝั่ง Web service ก็ทำการแกะข้อมูลออกจาก SOAP และเปลี่ยนเป็นคำสั่งที่ application เข้าใจ

-ในส่วน application ก็จะนำเอาคำสั่งหรือข้อมูลที่ไปทำงานจนได้ผลลัพธ์ ที่ต้องส่งกลับไปหาฝั่ง Client จัดใส่ใน HTTP format

-ฝั่ง Client ก็จะแกะข้อมูลออกจาก SOAP message เพื่อเราผลลัพธ์

#### 2.1.5.3 ข้อดีของ Webservice

-มีมาตรฐานในการใช้งาน

-มีความยืดหยุ่นในการใช้งาน สามารถปรับแต่งตามความต้องการ และใช้งานกันแพร่หลาย

- สามารถทำการ upgrade ได้อัตโนมัติ
- ลดค่าใช้จ่ายในการเชื่อมต่อหรือสื่อสาร
- ทำงานได้หลายหลายภาษา
- มีมาตรฐานความปลอดภัยเพียงพอกับองค์กรขนาดใหญ่ (saixiii, 2560)

### 2.1.6 API (Application Programming Interface)

API (Application Programming Interface) คือ ช่องทางการเชื่อมต่อระหว่างเว็บไซต์หนึ่งไปยังอีกเว็บไซต์หนึ่ง โดยอาจเป็นการเชื่อมต่อระหว่างผู้ใช้งานกับ Server หรือจาก Server หนึ่งเชื่อมต่อไปหาอีก Server หนึ่ง เพื่อให้ซอฟต์แวร์ภายนอกเข้าถึงและเปลี่ยนแปลงข้อมูลเว็บไซต์นั้น ๆ ได้ ตัวอย่างการทำงานของ API ที่เราเห็นได้ชัดและบ่อย ๆ คือ เว็บไซต์ส่วนใหญ่ในปัจจุบันจะมีตัวเลือกให้ผู้ใช้งานสามารถเข้าสู่ระบบอย่างง่าย ๆ ผ่าน Facebook Account นั้นหมายความว่าเว็บไซต์นั้น ๆ แคร่ข้อมูลร่วมกับ Facebook บนฐาน API อยู่นั่นเอง อีกหนึ่งตัวอย่างคือ Facebook : Messenger Payment ที่ช่วยให้ลูกค้าสามารถทำธุรกรรมทางการเงินบนระบบ Messenger หรือบนแอปพลิเคชันของ Facebook ได้โดยตรงผ่านการทำงานของ API ซึ่งการทำ API ถือเป็นตัวช่วยในการการสร้างโอกาสทางธุรกิจใหม่ๆ ให้แก่องค์กรต่าง ๆ รวมไปถึงการเพิ่มประสิทธิภาพในการจัดการระบบเว็บไซต์หรือแอปพลิเคชันและการนำข้อมูลไปพัฒนาต่อยอดด้วยเช่นกัน (Wongsuphasawat, 2560)

### 2.1.7 เว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์ (Web Server) คือเครื่องคอมพิวเตอร์ ที่ทำการติดตั้ง Software เกี่ยวกับ Server ที่ทำหน้าที่บริการและส่งข้อมูลให้กับ Client ที่ทำการ Request ข้อมูล และ Web Server จะทำการ Response เพื่อส่งผลลัพธ์ไปยัง Client โดย Web Server ทัว ๆ ไปที่เราคุ้น ๆ กันดีคือติดตั้ง Software ที่ชื่อว่า IIS (Internet Information Services) และ Apache โดย Service เหล่านี้จะบริการข้อมูลผ่าน Port : 80 และรับส่งข้อมูลผ่าน Web Browser ส่วนมาก

#### 2.1.7.1 Xampp

Xampp คืออะไร เป็นโปรแกรม Apache web server ไว้จำลอง web server เพื่อไว้ทดสอบสคริปหรือเว็บไซต์ในเครื่องของเรา โดยที่ไม่ต้องเชื่อมต่ออินเทอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใด ๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยม , MySQL ฐานข้อมูล, Apache จะทำหน้าที่เป็นเว็บ เซิร์ฟเวอร์, Perl อีกทั้งยังมาพร้อมกับ OpenSSL , phpMyadmin (ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe โปรแกรม Xampp อยู่ภายใต้ใบอนุญาตของ GNU General Public License



แต่บางครั้งอาจจะมีการเปลี่ยนแปลงเรื่องของลิขสิทธิ์ในการใช้งาน จึงควรติดตามและตรวจสอบโปรแกรมด้วย (mindphp, 2560)

## 2.1.8 Firebase

Firebase คือ บริการที่เกี่ยวกับด้าน Back-end ต่าง ๆ เหมาะกับคนที่ทำ Front-end (โดยเฉพาะ Mobile App) แล้วอยากเชื่อมต่อกับ Back-end แต่ไม่ยอมทำ Back-end เอง เช่น เช่าเซิร์ฟเวอร์โฮสต์ ติดตั้งโปรแกรมโฮสต์ ตั้งค่าการเชื่อมต่อต่าง ๆ เอง ไหนจะเรื่องความปลอดภัยอีกละ เป็นต้น ซึ่ง Firebase ได้จัดการสิ่งเหล่านี้ให้เรียบร้อยแล้ว อีกทั้งความสามารถอื่น ๆ อีกมากมายที่ช่วยเพิ่มประสิทธิภาพและคุณภาพให้กับงานโดยที่คุณอาจไม่คาดคิด เช่น รายงานข้อผิดพลาดโฮสต์ ข้อมูลสถิติผู้ใช้งานโฮสต์ ระบบทดสอบโฮสต์ เป็นต้น

### 2.1.8.1 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) คือ บริการส่งข้อความแจ้งเตือนแบบ ข้ามแพลตฟอร์ม (cross-platform messaging) ทั้ง Android, iOS และ Web แบบฟรีๆ ชื่อเดิมก็คือ Google Cloud Messaging (GCM)

ข้อควรรู้เกี่ยวกับ Firebase Cloud Messaging (FCM)

- การส่ง Notification แบบ Topic ใน Notification Console จะต้องรอประมาณ 3-6 ชั่วโมง หลัง subscribe topic นั้นไว้ถึงจะโผล่มาให้เลือก

- ขนาดของข้อมูลที่สามารถยัดลงไปใน Notification payload ต้องไม่เกิน 2KB

- ขนาดของข้อมูลที่สามารถยัดลงไปใน Data payload ต้องไม่เกิน 4KB

- โอกาสที่ Token เกิดเปลี่ยนแปลงได้ คือ uninstall แล้ว install แอปฯ ใหม่ และ ผู้ใช้ทำการลบข้อมูลของแอปฯ (clear data app)

สรุป การส่ง Notification ด้วย Firebase Cloud Messaging (FCM) ทำให้เราสามารถส่งข้อมูลที่เร่งด่วน หรืออาจจะเป็นเป็นประโยชน์ต่อผู้ใช้ได้รวดเร็ว

## 2.1.9 Omise API

### 2.1.9.1 พร้อมเพย์ Omise

เพิ่มช่องทางการรับชำระเงินผ่านพร้อมเพย์ อำนวยความสะดวกให้ผู้ซื้อสามารถทำรายการผ่านบัญชีธนาคารชั้นนำทั่วประเทศ บริการพร้อมเพย์ช่วยให้ร้านค้ารับชำระเงินจากผู้ซื้อทั่วประเทศไทยได้อย่างรวดเร็วและปลอดภัย ผู้ซื้อสามารถทำรายการได้ง่ายเพียงสแกนคิวอาร์โค้ดก็สามารถชำระเงินได้

## การเปิดชำระเงิน

-ประเทศที่รองรับ: ประเทศไทย

-API เวอร์ชันล่าสุดที่รองรับ: 2017-11-02

ร้านค้าที่ต้องการเปิดใช้ระบบ พร้อมเพย์ กรุณาส่งอีเมลมาที่ support@omise.co ทางทีมงานโอมิเซะ จะส่งข้อกำหนดและเงื่อนไขการใช้บริการให้ร้านค้าได้พิจารณาและลงลายมือชื่อก่อนเปิดใช้งานจริง

### ขั้นตอนการรับชำระเงินผ่านพร้อมเพย์

ผู้ซื้อที่ต้องการชำระเงินผ่านพร้อมเพย์จะต้องผ่านขั้นตอนการชำระเงินและทำการยืนยันรายการแบบ ออฟไลน์ โดยผู้ซื้อจะต้องทำการสแกนคิวอาร์โค้ดโดยใช้แอปพลิเคชันอินเทอร์เน็ตแบงก์กิ้งของตนเองเพื่อยืนยันรายการ

เมื่อผู้ซื้อเลือกพร้อมเพย์เป็นช่องทางการชำระเงินแล้ว เว็บไซต์ของร้านค้าจะมีการแสดงคิวอาร์โค้ดเพื่อให้ผู้ซื้อสแกน ภาพด้านล่างนี้แสดงขั้นตอนการรับชำระเงิน 2 ประเภทคือ การชำระเงินผ่านเบราว์เซอร์บนคอมพิวเตอร์และการชำระเงินผ่านเบราว์เซอร์บนมือถือ เมื่อร้านค้าได้แสดงคิวอาร์โค้ดขึ้นมาแล้ว ผู้ซื้อจะต้องสแกนเพื่อชำระเงิน

### การชำระเงินผ่านเบราว์เซอร์บนคอมพิวเตอร์



### ภาพประกอบที่ 2.3 การชำระเงินผ่านเบราว์เซอร์บนคอมพิวเตอร์

ผู้ซื้อลงชื่อเข้าใช้งานแอปอินเทอร์เน็ตแบงก์กิ้งของตนเองเพื่อสแกนคิวอาร์โค้ด 2 ผู้ซื้อสแกนคิวอาร์โค้ดผ่านแอป 3 ผู้ซื้อยืนยันรายการผ่านแอป 4 เมื่อร้านค้าได้รับการแจ้งเตือนรายการสำเร็จผ่าน webhook ให้ยืนยันรายการกับผู้ซื้ออีกครั้ง 5

### การชำระเงินผ่านเบราว์เซอร์บนมือถือ



### ภาพประกอบที่ 2.4 การชำระเงินผ่านเบราว์เซอร์บนมือถือ

ผู้ซื้อทำการถ่ายภาพหน้าจอที่แสดงคิวอาร์โค้ด 1 จากนั้นลงชื่อเข้าใช้งานแอปอินเทอร์เน็ตแบงก์กิ้งของตนเอง 2 ผู้ซื้ออัปโหลดภาพถ่ายหน้าจอคิวอาร์โค้ดลงในแอป 3 ผู้ซื้อยืนยันรายการผ่านแอป 4 เมื่อร้านค้าได้รับการแจ้งเตือนรายการสำเร็จผ่าน webhook ให้ยืนยันรายการกับผู้ซื้ออีกครั้ง 5

### การติดตั้งใช้งาน

ในการสร้างรายการรับชำระเงินผ่านพร้อมเพย์ ให้ร้านค้าส่งคำสั่ง API ดังนี้

1. สร้าง payment source (type: promptpay) โดยใช้ Omise.js, omise-ios หรือ omise-android

2. สร้าง charge โดยใช้ source จากขั้นตอนแรก

3. เมื่อผู้ซื้อทำรายการสำเร็จและร้านค้าได้รับ webhook event หรือ charge.complete แล้ว เราแนะนำให้ตรวจสอบสถานะรายการด้วยตนเองอีกครั้งเพื่อความแม่นยำ

การสร้าง source เพื่อรับชำระเงินผ่านพร้อมเพย์ จะเกิดขึ้นในฝั่งของผู้ซื้อ (client-side) เช่นบนเว็บไซต์หรือโทรศัพท์มือถือของผู้ซื้อ ร้านค้าจะต้องใช้ public key

ส่วนการสร้างรายการ (charge) เพื่อรับชำระเงินผ่านพร้อมเพย์ จะเกิดขึ้นในฝั่งของร้านค้า (server-side) ร้านค้าจะต้องใช้ secret key

### การสร้าง source

เมื่อผู้ซื้อเลือกชำระเงินผ่านช่องทางนี้ ให้ร้านค้าสร้าง source ผ่าน source API โดยกำหนด amount, currency, และ type

ตัวอย่างด้านล่างนี้เป็นการสร้าง source สำหรับรายการ พร้อมเพย์ จำนวน ฿4,000 ให้แทนค่าของ omise\_public\_key และ \$OMISE\_PUBLIC\_KEY ด้วย test public key ของร้านค้าซึ่งสามารถคัดลอกได้จาก แดชบอร์ด

```
Omise.setPublicKey(omise_public_key);

Omise.createSource('promptpay', {
  "amount": 400000,
  "currency": "THB"
}, function(statusCode, response) {
  console.log(response)
});
```

ภาพประกอบที่ 2.5 Omise.js parameter type

```
curl https://api.omise.co/sources \
  -u $OMISE_PUBLIC_KEY: \
  -d "amount=400000" \
  -d "currency=THB" \
  -d "type=promptpay"
```

ภาพประกอบที่ 2.6 หากร้านค้าต้องการทดสอบให้สร้าง request โดยใช้ curl

```
{
  "object": "source",
  "id": "src_test_51j681ua86u2kmryfup",
  "livemode": false,
  "location": "/sources/src_test_51j681ua86u2kmryfup",
  "amount": 400000,
  "barcode": null,
  "created_at": "2020-10-14T03:52:01Z",
  "currency": "THB",
  "email": null,
  "flow": "offline",
  "installment_term": null,
  "name": null,
  "mobile_number": null,
  "phone_number": null,
  "scannable_code": null,
  "references": null,
  "store_id": null,
  "store_name": null,
  "terminal_id": null,
  "type": "promptpay",
  "zero_interest_installments": null,
  "charge_status": "unknown"
}
```

ภาพประกอบที่ 2.7 ตัวอย่าง output ตัวแปรของ id คือ source identifier

### การสร้างรายการรับชำระเงิน

สร้างรายการรับชำระเงินโดยระบุ parameter source, amount และ currency

1.source จะเป็นตัวกำหนด source identifier

2.amount และ currency จะต้องมีค่าตรงกับ amount และ currency ของ source

ตัวอย่างด้านล่างนี้แสดงการสร้างรายการรับชำระเงินผ่าน ขึ้นใหม่โดยใช้ curl

ในการสร้างรายการให้ร้านค้าแทน \$OMISE\_SECRET\_KEY ด้วย test secret key ของร้านค้าซึ่งสามารถคัดลอกได้จากแดชบอร์ด และแทน \$SOURCE\_ID ด้วย id ของ source

```
curl https://api.omise.co/charges \
  -u $OMISE_SECRET_KEY: \
  -d "amount=400000" \
  -d "currency=THB" \
  -d "source=$SOURCE_ID"
```

ภาพประกอบที่ 2.8 การสร้างรายการรับชำระเงิน

การสร้าง source และรายการรับชำระเงิน

ร้านค้าสามารถสร้างทั้ง source และ รายการรับชำระเงิน (charge) ผ่านการเรียกใช้ API เพียงครั้งเดียว

```
curl https://api.omise.co/charges \
  -u $OMISE_SECRET_KEY: \
  -d "amount=400000" \
  -d "currency=THB" \
  -d "source[type]=promptpay"
```

ภาพประกอบที่ 2.9 การสร้าง source และรายการรับชำระเงิน

สร้างรายการสำเร็จ

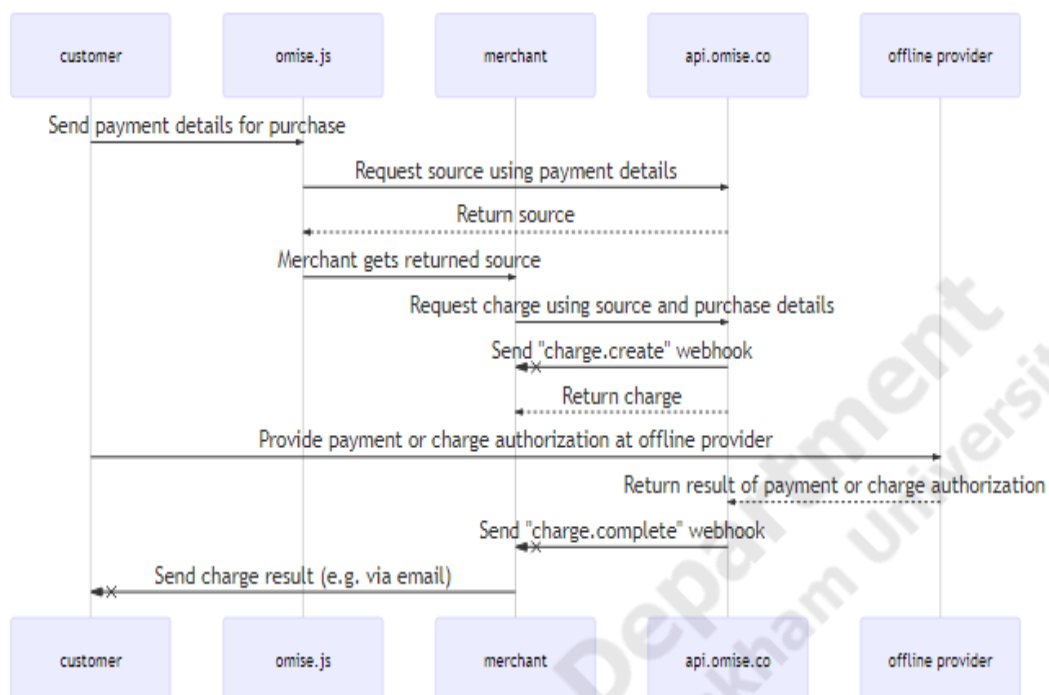
เมื่อร้านค้าสร้างรายการรับชำระเงินขึ้นแล้วจะพบว่าสถานะของรายการแสดง

เป็น pending หรือ status==pending สถานะของรายการรับชำระเงินสามารถเป็นได้

ทั้ง successful, failed และ expired

ในส่วนต่อไปเราจะอธิบายวิธีอนุมัติรายการ, การรับ event แจ้งเตือนเมื่อรายการเสร็จสิ้นผ่าน webhook และการตรวจสอบสถานะรายการ

ภาพด้านล่างนี้แสดงให้เห็นถึงขั้นตอนทั้งหมดในการรับชำระเงินแต่ละรายการ



ภาพประกอบที่ 2.10 ขั้นตอนทั้งหมดในการรับชำระเงินแต่ละรายการ

### การอนุมัติรายการ

ร้านค้าจะแสดงคิวอาร์โค้ดที่สร้างขึ้นให้กับผู้ซื้อ โดยในการอนุมัติรายการผู้ซื้อจะสแกนคิวอาร์โค้ดด้วยโทรศัพท์มือถือ  
คิวอาร์โค้ดจะอยู่ใน charge object ตามตัวอย่างด้านล่างนี้

```

charge:
  source:
    scannable_code:
      image:
        download_uri: QR code image to present to the customer
  
```

ภาพประกอบที่ 2.11 การอนุมัติรายการ

```

{
  "object": "barcode",
  "type": "qr",
  "image": {
    "object": "document",
    "livemode": false,
    "id": "docu_test_51j681klhasodhsgkli",
    "deleted": false,
    "filename": "qrcode.png",
    "location": "/charges/chrg_test_51j681j1o9tndf6h1sy/documents/docu_test_5",
    "download_uri": "https://api.omise.co/charges/chrg_test_51j681j1o9tndf6h1",
    "created_at": "2020-10-14T03:52:00Z"
  }
}

```

ภาพประกอบที่ 2.12 ตัวอย่างการ output การอนุมัติรายการ

### การรับ event แจ้งเตือนรายการสำเร็จ

ร้านค้าสามารถรับการแจ้งเตือนเมื่อมีการทำรายการเสร็จสิ้นโดยใช้งาน webhook events ในการติดตั้งให้ร้านค้ากำหนดตำแหน่งบนเซิร์ฟเวอร์เพื่อรับ webhook events และเพิ่มตำแหน่งเดียวกันนี้เป็น webhook endpoint บนแดชบอร์ด

เมื่อมีรายการเสร็จสิ้นระบบจะส่ง POST request ไปยัง endpoint นี้ พร้อมทั้งแนบสถานะการตอบกลับของรายการนั้นๆ ไปด้วย

ตัวแปรหลักหรือ key สำหรับ event object ประกอบไปด้วย charge.complete และตัวแปร data ที่มี charge object

### การตรวจสอบสถานะรายการ

เมื่อได้รับ event ของรายการที่เสร็จสิ้นแล้ว ร้านค้าสามารถตรวจสอบสถานะรายการหรือ status โดยใช้ Charge API

หากค่าของ charge.status เป็น successful หมายถึงว่ารายการรับชำระเงินสำเร็จ หากค่าของ charge.status เป็น failed รายการรับชำระเงินนั้นไม่สำเร็จ ร้านค้าสามารถตรวจสอบ failure\_code และ failure\_message ได้ใน charge object เพื่ออ่านคำอธิบายเพิ่มเติมสาเหตุข้อขัดข้องที่อาจเกิดขึ้นได้มีดังนี้

รหัสข้อขัดข้อง	รายละเอียด
failed_processing	ระบบทำรายการไม่สำเร็จ
insufficient_balance	วงเงินคงเหลือไม่เพียงพอหรือวงเงินเต็ม
payment_cancelled	ผู้ซื้อยกเลิกการชำระเงิน

ภาพประกอบที่ 2.13 การตรวจสอบสถานะรายการ

### การยกเลิกรายการและการคืนเงิน

รายการที่รับชำระผ่านพร้อมเพย์ยังไม่สามารถทำการยกเลิกรายการหรือการคืนเงินผ่านโอมิเซะได้

### ข้อจำกัด

จำนวนรับชำระขั้นต่ำ: 2000 (THB20.00)

จำนวนรับชำระสูงสุด: 15000000 (THB150,000.00)

(omise, 2562)

## 2.2 ระบบงานที่เกี่ยวข้อง

ตัวอย่างระบบที่เกี่ยวข้องกับแอปพลิเคชัน Easy Cook มี 4 ระบบดังนี้

### 2.2.1 วงใน (Wongnai)

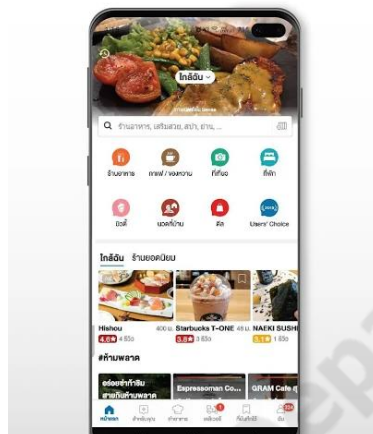
Wongnai (wongnai.com) จะช่วยค้นหาร้านอาหาร ร้านเสริมสวย สปา โรงแรม ที่พัก และสถานที่ท่องเที่ยว ใกล้ตัว ในกรุงเทพฯ, เชียงใหม่, ชลบุรี, พัทยา, อโยธยา, โคราช, ขอนแก่น, หัวหิน, ภูเก็ต, หาดใหญ่ และจังหวัดอื่น ๆ ทั่วประเทศไทย ที่แนะนำโดยทีมงาน Wongnai และรีวิวจากสมาชิกมากกว่า 3 ล้านคนที่ใช้บริการจริง จากฐานข้อมูลกว่า 250,000 ร้านทั่วประเทศไทย และมากกว่า 10 ล้านรีวิวและรูปภาพใน Wongnai

ค้นหาร้าน ค้นหาร้านใกล้ตัวคุณ พร้อมที่อยู่ร้าน, เบอร์โทรศัพท์, เมนู และเวลาเปิด-ปิด-ร้านอาหาร เช่น บุฟเฟ่ต์, ปิ้งย่าง, ชาบู, ซูชิ, สเต็ก, ส้มตำ, อาหารไทย, ก๋วยเตี๋ยว, อาหารทะเล ฯลฯ



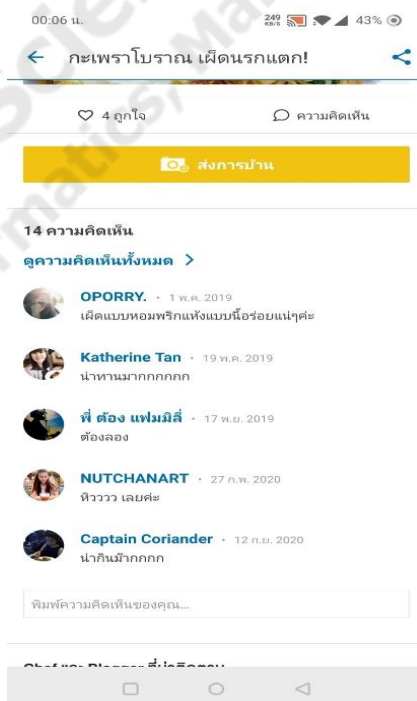
## ค้นหา

ร้านอาหารใกล้ตัว,  
ร้านเสริมสวย, ร้านเดลิเวอรี่,  
สูตรอาหาร, ที่พัก และที่เที่ยง



ภาพประกอบที่ 2.14 ตัวอย่างหน้าแสดงการค้นหา (Wongnai)

รีวิว อ่านรีวิวร้านจากสมาชิกวงในเพื่อช่วยให้คุณตัดสินใจเลือกร้านได้ง่ายขึ้น พร้อมโพสรูปหรือรีวิว เพื่อแบ่งปันข้อมูลกับเพื่อนสมาชิกได้ทุกเมื่อ



ภาพประกอบที่ 2.15 หน้าแสดงตัวอย่างการรีวิว (Wongnai)

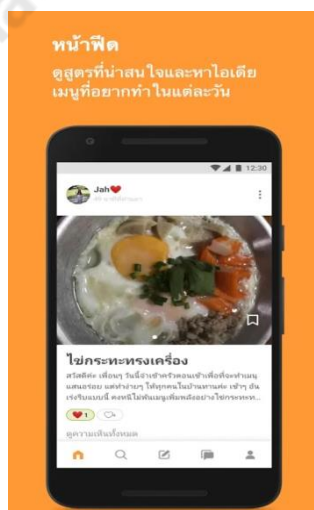
คู่มืออาหาร คู่มือการทำอาหารหลากหลายเมนูที่ทำตามง่ายๆ ได้ที่บ้าน



ภาพประกอบที่ 2.16 หน้าแสดงตัวอย่างสูตรอาหารต่าง ๆ (Wongnai)

## 2.2.2 Cookpad

Cookpad คอมมิวนิตีของคนรักการทำอาหารที่ใหญ่ที่สุดในโลก มีผู้ค้นเข้ามาแชร์สูตรอาหารหลายพันสูตรต่อวัน รวมถึง ไอเดีย และประสบการณ์ที่ได้จากการเรียนทำอาหาร หรือฝึกทำอาหาร ตบโจทย์สำหรับทุกคนไม่ว่าจะเป็นคนที่ชอบการสร้างสรรค์เมนู หรือชอบทำตามสูตรต่าง ๆ เพราะแต่ละสูตรใน Cookpad ผ่านการสร้างสรรค์ ฝึกฝน และทดลองโดยผู้คนที่ทำอาหารทานเองที่บ้าน และแบ่งปันสู่คนที่ชอบทำอาหารทานเองเหมือนกัน



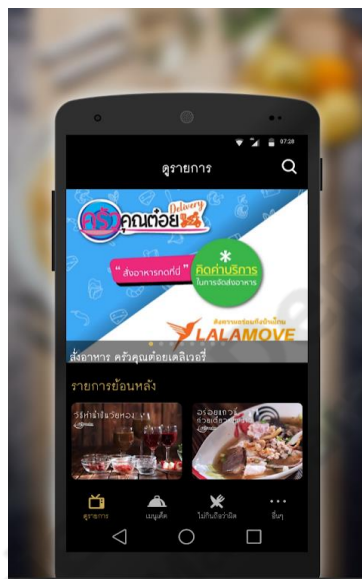
ภาพประกอบที่ 2.17 หน้าแสดงตัวอย่างหน้าพืด (Cookpad)

ค้นพบคอลเลกชันที่รวบรวมสูตรอาหารที่หลากหลาย และค้นหาวิธีทำอาหารได้ตามต้องการ สามารถค้นหาจาก วัตถุดิบ ชื่อเมนูยอดนิยม และอื่น ๆ



### 2.2.3 ครีว้คุณต๋อย

ภายในแอปพลิเคชันครีว้คุณต๋อย จะรวมเอาเมนูเด็ด ร้านดัง ร้านอร่อยทุก ๆ ร้านที่เคยออกอากาศในรายการครีว้คุณต๋อย แอปพลิเคชันประกอบด้วยสูตรอาหารและเคล็ดลับการปรุงอาหารเมนูต่าง ๆ จากเจ้าของสูตรต้นตำรับตัวจริงและยังสามารถใช้แอปพลิเคชันค้นหาอาหารร้านอาหารอร่อยแฉวนี้ รวมทั้งใช้เป็นข้อมูลในการหาร้านอาหารในต่างจังหวัด โดยจะมีพิกัดสถานที่ตั้งร้านอาหารขึ้นชื่อว่าถ้าไปจังหวัดนั้น



ภาพประกอบที่ 2.20 หน้าแสดงตัวอย่างหน้าพีด (ครีว้คุณต๋อย)



ภาพประกอบที่ 2.21 หน้าแสดงรายการอาหาร (ครีว้คุณต๋อย)



ภาพประกอบที่ 2.22 หน้าแสดงตัวอย่างแสดงสูตรและวิธีทำอาหาร (ครัวคุณต๋อย)

#### 2.2.4 เมนูอาหาร สูตรอาหารง่าย ๆ

แอปสอนทำอาหารแบบด้วยขั้นตอนง่ายๆ ภาพประกอบชัดเจน เมนูหลากหลาย



ภาพประกอบที่ 2.23 หน้าแสดงตัวอย่างหน้าฟีด (เมนูอาหาร สูตรอาหาร)



ภาพประกอบที่ 2.24 หน้าแสดงตัวอย่างขั้นตอนวิธีการทำอาหารต่างๆ (เมนูสูตรอาหาร)

### 2.3 ตารางเปรียบเทียบ

ตารางที่ 2.1 ตารางเปรียบเทียบระบบที่เกี่ยวข้อง

ฟังก์ชันการทำงาน	Wongnai	Cookpad	ครัว คุณ ต๋อย	เมนูอาหาร	ระบบ ที่ พัฒนา
สมัครเป็นสมาชิก	✓	✓			✓
ล็อกอินและล็อกเอาต์ ระบบ	✓	✓			✓
ยกเลิกการเป็นสมาชิก	✓	✓			✓
แก้ไขข้อมูลของตนเอง	✓	✓			✓
สามารถค้นหาชื่อสมาชิก	✓	✓			✓
สามารถค้นหาชื่อ เมนูอาหารและวัตถุดิบ	✓	✓	✓		✓
หน้าฟีดแสดงเมนูอาหาร	✓	✓	✓	✓	✓
สามารถเพิ่มสูตรอาหาร	✓	✓	✓	✓	✓
สามารถแก้ไขและลบสูตร อาหาร	✓	✓	✓	✓	✓

ตารางที่ 2.1 ตารางเปรียบเทียบระบบที่เกี่ยวข้อง (ต่อ)

ฟังก์ชันการทำงาน	Wongnai	Cookpad	ครัว คุณ ต๋อย	เมนูอาหาร	ระบบ ที่ พัฒนา
สามารถซื้อขายสูตรอาหาร ได้	✓	✓			✓
ระบบการติดตามผู้ใช้ที่เป็น สมาชิกด้วยกัน	✓	✓			✓
สามารถรีวิวและคอมเมนต์ สูตรอาหารได้	✓	✓			✓
ฝั่งผู้ดูแลระบบสามารถ จัดการสมาชิกได้	✓	✓			✓
สามารถเติมเงินเข้าระบบ เพื่อซื้อสูตรอาหารได้					✓
สามารถถอนเงินในระบบ ออกมาเป็นเงินจริงได้					✓
สามารถสั่งอาหารเดลิเวอรี่ ได้	✓				
สามารถหาเส้นทาง ร้านอาหารผ่านgoogle maps ได้	✓				