

โครงการเลขที่ กพ. 2565/1

เรื่อง

ระบบจัดการร้านอาหารเดลิเวอรี่

Delivery Restaurant Manager

โดย

นายธชิน วิสุทธิ์มรรคกุล 6101240160

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาบริหารคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ

พ.ศ.2565

โครงการเลขที่ กพ. 2565/1

เรื่อง

ระบบจัดการร้านอาหารเดลิเวอรี่

Delivery Restaurant Manager

โดย

นายธชิน วิสุทธิ์มรรคกุล 6101240160

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาบริหารคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ

พ.ศ.2565

หัวโครงงาน	ระบบจัดการร้านอาหารเดลิเวอรี่
โดย	นายธชิณ วิสุทธิอมรรคกุล
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์พัฒน์นรี ศรีสมพันธ์
ปีการศึกษา	2565

สาขาวิชาการคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ อนุมัติให้นับโครงการนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตร์บัณฑิต

គណន៍ការរំសៀវភៅក្នុងការអនុវត្តន៍  
ការបង្កើតរូបរាងសាស្ត្រ និងការបង្កើតរូបរាងសាស្ត្រ

## ข้อตกลงในการใช้งานซอฟต์แวร์

โครงงานชื่อ ระบบจัดการร้านอาหารเดลิเวอรี่ (Delivery Restaurant Manager) นี้เป็นผลงาน  
ที่พัฒนาขึ้นโดย นายธชิน วิสุทธิ์มรรคกุล รหัสนักศึกษา 6101240160 ภายใต้การดูแลของ อาจารย์พัฒน์นรี ศรี  
สมพันธ์ เป็นส่วนหนึ่งของการศึกษารายวิชา คพ.499 โครงงานคอมพิวเตอร์ 2 ตามหลักสูตร ปริญญาวิทยาศาสตร  
บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจ มหาวิทยาลัยพายัพ พ.ศ. 2565 โดยมีวัตถุประสงค์  
เพื่อให้นักศึกษาได้เรียนรู้และฝึกทักษะในการพัฒนาซอฟต์แวร์ ลิขสิทธิ์ของซอฟต์แวร์นี้จึงเป็นของผู้พัฒนา ซึ่ง  
ผู้พัฒนาได้อនุญาตให้ทางสาขาวิชาวิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจมหาวิทยาลัยพายัพ เผยแพร่ซอฟต์แวร์  
นี้ตาม “ต้นฉบับ” โดยไม่มีการแก้ไขดัดแปลงใดๆ ทั้งสิ้นให้แก่บุคคลทั่วไปได้ใช้เพื่อประโยชน์ส่วนบุคคล หรือ  
ประโยชน์ทางการศึกษาที่ไม่มีวัตถุประสงค์ในเชิงพาณิชย์ โดยไม่คิดค่าตอบแทนการใช้ซอฟต์แวร์ ดังนั้นสาขาวิชาฯ  
จึงไม่มีหน้าที่ในการดูแล บำรุงรักษา จัดการอบรมการใช้งาน หรือพัฒนาประสิทธิภาพซอฟต์แวร์ รวมทั้งไม่รับรอง  
ความถูกต้องหรือประสิทธิภาพการทำงานของซอฟต์แวร์ ตลอดจนไม่รับประกันความเสียหายต่างๆ อันเกิดจากการ  
ใช้ซอฟต์แวร์นี้ทั้งสิ้น

## บทคัดย่อ

โครงการระบบจัดการร้านอาหารเดลิเวอรี่ เป็นเว็บแอพพลิเคชันสำหรับการจัดการร้านอาหารที่เน้นไปที่การขายผ่านแพลตฟอร์ม (platform) ต่างๆเป็นหลัก เพื่อเก็บสถิติการขายและต้นทุนอาหารผ่านการขายในแพลตฟอร์มต่างๆ โดยการนำไปใช้ผ่านเว็บเบราว์เซอร์ (web browser) เป็นหลัก

โครงการนี้ ประกอบด้วยเนื้อหาสามส่วน ส่วนแรกเป็นการตั้งค่าข้อมูลต่างๆภายในร้านอาหาร ซึ่งประกอบด้วย 4 หัวข้อ คือ วัตถุภัยในร้าน อาหารภัยในร้าน เมนูอาหารภัยในร้าน และแพลตฟอร์มที่ร้านค้าเข้าร่วมรวมไปถึงราคาเมนูอาหารภัยในแพลตฟอร์มต่างๆ ส่วนที่สองเป็นการลงบันทึกรายจ่ายและคำสั่งซื้อ ส่วนสุดท้ายคือรายงานผลการขาย

โปรแกรมนี้พัฒนาโดยใช้ภาษา Java Script (JavaScript) เป็นหลัก โดยใช้ MERN Stack ในการพัฒนา ซึ่งประกอบด้วย ฐานข้อมูลแบบ NoSQL – MongoDB สำหรับเก็บข้อมูลอาหาร แพลตฟอร์ม รายจ่าย และคำสั่งซื้อ, ระบบ Node.js สำหรับรัน JavaScript เป็นเซิฟเวอร์, Express.js สำหรับการสร้าง Rest API เพื่อเชื่อมต่อ กับฐานข้อมูล, React ซึ่งเป็น JavaScript Library สำหรับสร้างส่วนแสดงผลของเว็บแอพพลิเคชัน

ผลที่ได้จากการนี้ ได้ช่วยให้ผู้ประกอบการร้านอาหารแบบออนไลน์จัดการรายจ่ายและคำสั่งซื้อได้สะดวกสบายมากยิ่งขึ้น ไม่ต้องขึ้นกับแอพพลิเคชันอื่นๆเพื่อเก็บส่วนต่างๆ, ลดค่าใช้จ่ายจากการจ่ายค่าスマชิกรายเดือนในแอพพลิเคชันต่างๆ และมีสถิติการขายที่ชัดเจน, เข้าถึงง่าย

## Abstract

This ‘Delivery Restaurant Manager’ project aim to develop web applications. Which is use to manage restaurant that only serve via online delivery platform(s). To collect sales statistics and foods cost.

There are three parts of the project; setting up the store’s system, orders&expenses manager and sales report. In the first part ‘setting up’ are divided into 4 sections which are materials, food, menu and platform&pricing. The second part divided into 2 sections; expenses tracking and orders tracking. And lastly sales report which show how each platforms performs.

The project developed in all JavaScript languages with MERN Stack. Which contains MongoDB a NoSQL database, Express a backend framework, React a frontend library and Node.js for JavaScript environments.

The result of this application can help the store’s personnel manage expenses and orders easier; no more depends on the separate applications. Reduce subscription fees from which applications. And easily accessible sales report.

## กิตติกรรมประกาศ

ผู้จัดทำโครงการขอขอบพระคุณอาจารย์พัฒน์นรี ศรีสมพันธ์ อาจารย์ที่ปรึกษาโครงการ ที่เคยให้คำปรึกษา และคำแนะนำ ตลอดจนให้กำลังใจในการทำโครงการนี้ด้วยดีตลอดมา รวมทั้งอาจารย์ประจำสาขาวิชา วิทยาการคอมพิวเตอร์ คณะบริหารธุรกิจมหาวิทยาลัยพายัพทุกท่าน ที่ได้ให้ความรู้ และสร้างประสบการณ์ เกี่ยวกับคอมพิวเตอร์ในด้านต่างๆ นอกจากนี้ขอขอบคุณเพื่อนๆ รุ่นพี่และรุ่นน้อง สาขาวิชาวิทยาการคอมพิวเตอร์ ได้เคยช่วยเหลือ และให้กำลังใจเสมอมา

โครงการนี้คงจะสำเร็จลงมื้อได้ หากไม่ได้กำลังใจหลักจากคุณพ่อและคุณแม่ ที่เคยดูแลและสนับสนุนอย่างเต็มที่และสม่ำเสมอ

ธชิน วิสุทธิ์มงคล

## สารบัญ

<b>บทคัดย่อ</b>	ก
<b>Abstract</b>	ข
<b>กิตติกรรมประกาศ</b>	ค
<b>สารบัญ</b>	ง
<b>สารบัญรูปภาพ</b>	ฉ
<b>สารบัญตาราง</b>	ณ
<b>บทที่ 1 บทนำ</b>	๑
1.1. ความเป็นมาและความสำคัญของโครงการ	๑
1.2. วัตถุประสงค์ของโครงการ	๑
1.3. ขอบเขตโครงการ	๒
1.4. ขอบเขตของโครงการ	๓
1.5. กรอบแนวคิดการพัฒนาโครงการ	๕
1.6. ประโยชน์ที่ได้รับ	๕
1.7. นิยามศัพท์เฉพาะ	๖
<b>บทที่ 2 แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง (Literature Review)</b>	๗
2.1 แนวคิดทฤษฎี	๗
2.2. งานวิจัยที่เกี่ยวข้อง	๑๖
<b>บทที่ 3 วิธีดำเนินโครงการ (Research Methodology)</b>	๒๑
3.1. ขั้นตอนการดำเนินงานตาม Software Process Model	๒๑
3.2. แผนภาพกราฟแสดงข้อมูล	๒๓

3.3. โครงสร้างซอฟต์แวร์ (Software Structure)	24
3.4. ข้อกำหนดซอฟต์แวร์ (Input/Output Specifications)	26
3.5. E-R Diagram	34
3.6. ตารางฐานข้อมูล (Tables)	35
3.7. ส่วนต่อประสาน (User Interface / Screen Layout)	40
<b>บทที่ 4 ผลการพัฒนาระบบ (Results)</b>	<b>53</b>
4.1. ผลการทดสอบ (Test report)	53
4.2. ผลลัพธ์ของระบบที่พัฒนาในภาพรวม (Output)	69
4.3. ผลประเมินความพึงพอใจของผู้ใช้ระบบ (Assessment)	74
4.4. คู่มือการติดตั้ง (Installation)	75
4.5. คู่มือการใช้งาน (User manual)	79
<b>บทที่ 5 บทสรุปและข้อเสนอแนะ (Conclusions and Recommendations)</b>	<b>92</b>
5.1. วัตถุประสงค์หรือประเด็นสำคัญในการพัฒนา	92
5.2. ผลการพัฒนาสามารถแก้ปัญหาได้มากน้อยเพียงใด	93
5.3. สิ่งที่คาดหวังที่ระบบจะสามารถช่วยในการทำงานได้มากน้อยเพียงใด	94
5.4. ปัญหาและข้อเสนอแนะ	94
<b>บรรณานุกรม</b>	<b>97</b>
<b>ประวัติย่อผู้พัฒนา</b>	<b>99</b>
<b>ตัวอย่างเอกสาร</b>	<b>100</b>

## สารบัญรูปภาพ

รูปประกอบ 1.1 Waterfall model	5
รูปประกอบ 1.2 Gantt Chart	5
รูปประกอบ 3.1 Dataflow Diagram	23
รูปประกอบ 3.2 Software Structure	24
รูปประกอบ 3.3 Entity Relationship Diagram	34
รูปประกอบ 3.4 Login UI	40
รูปประกอบ 3.5 UI ต่างร่างสติ๊กติการขาย	41
รูปประกอบ 3.6 UI กราฟสติ๊กติการขาย	42
รูปประกอบ 3.7 UI การจัดการคำสั่งซื้อ	43
รูปประกอบ 3.8 UI ตารางบันทึกรายจ่าย	44
รูปประกอบ 3.9 UI การสร้างรายจ่าย	45
รูปประกอบ 3.10 UI การแก้ไขรายจ่าย	45
รูปประกอบ 3.11 UI การสร้างและลบหมวดหมู่รายจ่าย	46
รูปประกอบ 3.12 UI การจัดการวัตถุคิดบ	47
รูปประกอบ 3.13 UI การแก้ไขวัตถุคิดบ	48
รูปประกอบ 3.14 UI ตารางรายการอาหาร	48
รูปประกอบ 3.15 UI การสร้างอาหาร	49
รูปประกอบ 3.16 UI การแก้ไขอาหาร	49
รูปประกอบ 3.17 UI ตารางและการลบเมนูอาหาร	50
รูปประกอบ 3.18 UI การสร้างเมนูอาหาร	50
รูปประกอบ 3.19 UI การแก้ไขเมนูอาหาร	51
รูปประกอบ 3.20 UI แพลตฟอร์มและตารางราคา	51
รูปประกอบ 3.21 UI การสร้างแพลตฟอร์ม	52
รูปประกอบ 4.1 การทำงานในภาคร่วม	69
รูปประกอบ 4.2 การเก็บสูตรอาหาร	70

รูปประกอบ 4.3 การตั้งราคาแยกตามแพลตฟอร์ม	71
รูปประกอบ 4.4 การสร้างคำสั่งชื่อ	72
รูปประกอบ 4.5 การทำคำคำสั่งชื่อ	72
รูปประกอบ 4.6 การค้นหาข้อมูลในตาราง	73
รูปประกอบ 4.7 การบันทึกคำสั่งชื่อ	73
รูปประกอบ 4.8 กราฟแสดงสถิติการขาย	74
รูปประกอบ 4.9 คำสั่งสำหรับติดตั้งโปรแกรม	76
รูปประกอบ 4.10 ตัวอย่างไฟล์ .env	77
รูปประกอบ 4.11 คำสั่งเริ่มต้นระบบ	77
รูปประกอบ 4.12 ตัวอย่างเมื่อเริ่มต้นระบบสำเร็จ	78
รูปประกอบ 4.13 ตัวอย่างหน้าเข้าสู่ระบบ	78
รูปประกอบ 4.14 ตัวอย่างการกรอกข้อมูลหน้าสมัครสมาชิก	79
รูปประกอบ 4.15 ลักษณะของหน้าเข้าสู่ระบบ	79
รูปประกอบ 4.16 ตำแหน่งเมนู Material	80
รูปประกอบ 4.17 การสร้างวัตถุดิบ	81
รูปประกอบ 4.18 ตำแหน่งเมนู Item	81
รูปประกอบ 4.19 ตำแหน่งปุ่มสร้างอาหาร	82
รูปประกอบ 4.20 Searchable dropdown	82
รูปประกอบ 4.21 ตำแหน่งเมนู Rider&Pricing	84
รูปประกอบ 4.22 ตำแหน่งปุ่มสร้างแพลตฟอร์ม	84
รูปประกอบ 4.23 การเลือกໄรเดอร์	85
รูปประกอบ 4.24 ตำแหน่งในการแก้ไขราคา	85
รูปประกอบ 4.25 ตำแหน่งเมนู Expense	86
รูปประกอบ 4.26 ตำแหน่งปุ่มแก้ไขหมวดหมู่รายจ่าย	86
รูปประกอบ 4.27 ตำแหน่งปุ่มสร้างคำสั่งชื่อ	87
รูปประกอบ 4.28 ตำแหน่งเมนู Order	88
รูปประกอบ 4.29 ส่วนสำหรับเลือกໄรเดอร์	88
รูปประกอบ 4.30 ส่วนสำหรับเลือกเมนูอาหาร	89

---

รูปประกอบ 4.31 ตำแหน่งเมนู Dashboard	89
รูปประกอบ 4.32 การเลือกไกด์เดอร์ใน Dashboard	90
รูปประกอบ 4.33 การเลือกขอบเขตเวลา	90
รูปประกอบ 4.34 กราฟสถิติการขาย	91

## สารบัญตาราง

ตาราง 3.1 Login module	26
ตาราง 3.2 Authentication module	26
ตาราง 3.3 โมดูลเพิ่มวัตถุติดบ	27
ตาราง 3.4 โมดูลแก้ไขวัตถุติดบ	27
ตาราง 3.5 โมดูลลบวัตถุติดบ	27
ตาราง 3.6 โมดูลเพิ่มอาหาร	27
ตาราง 3.7 โมดูลแก้ไขอาหาร	28
ตาราง 3.8 โมดูลลบอาหาร	28
ตาราง 3.9 โมดูลเพิ่มเมนูอาหาร	28
ตาราง 3.10 โมดูลแก้ไขเมนูอาหาร	29
ตาราง 3.11 โมดูลลบเมนูอาหาร	29
ตาราง 3.12 โมดูลเพิ่มໄรเดอร์	29
ตาราง 3.13 โมดูลลบໄรเดอร์	30
ตาราง 3.14 โมดูลแก้ไขราคาเมนูอาหาร	30
ตาราง 3.15 โมดูลเพิ่มนวดหมู่รายจ่าย	30
ตาราง 3.16 โมดูลแก้ไขหมวดหมู่รายจ่าย	30
ตาราง 3.17 โมดูลลบหมวดหมู่รายจ่าย	31
ตาราง 3.18 โมดูลเพิ่มรายจ่าย	31
ตาราง 3.19 โมดูลแก้ไขรายจ่าย	31
ตาราง 3.20 โมดูลลบรายจ่าย	32
ตาราง 3.21 โมดูลทำซ้ำรายจ่าย	32
ตาราง 3.22 โมดูลเพิ่มคำสั่งชื้อ	32
ตาราง 3.23 โมดูลลบคำสั่งชื้อ	33
ตาราง 3.24 โมดูลแสดงสถิติการขาย	33
ตาราง 3.25 ฐานข้อมูลผู้ใช้	35

ตาราง 3.26 ฐานข้อมูลวัตถุดิบ	35
ตาราง 3.27 ฐานข้อมูลอาหาร	36
ตาราง 3.28 ฐานข้อมูลเมนูอาหาร	37
ตาราง 3.29 ฐานข้อมูลแพลตฟอร์มการขาย	38
ตาราง 3.30 ฐานข้อมูลหมวดหมู่รายจ่าย	38
ตาราง 3.31 ฐานข้อมูลรายจ่าย	39
ตาราง 3.32 ฐานข้อมูลคำสั่งชิ้อ	40
ตาราง 4.1 ข้อมูลการทดสอบ	53
ตาราง 4.2 ทดสอบ ระบบตรวจสอบตัวตน	54
ตาราง 4.3 ทดสอบ การแสดงสถิติการขาย	55
ตาราง 4.4 ทดสอบ เพิ่มคำสั่งชิ้อ	56
ตาราง 4.5 ทดสอบ ลบคำสั่งชิ้อ	57
ตาราง 4.6 ทดสอบ สร้างหมวดหมู่รายจ่าย	57
ตาราง 4.7 ทดสอบ ลบหมวดหมู่รายจ่าย	57
ตาราง 4.8 ทดสอบ สร้างรายจ่าย	60
ตาราง 4.9 ทดสอบ ลบรายจ่าย	60
ตาราง 4.10 ทดสอบ ทำซ้ำรายจ่าย	61
ตาราง 4.11 ทดสอบ แก้ไขรายจ่าย	61
ตาราง 4.12 ทดสอบ เพิ่มวัตถุดิบ	62
ตาราง 4.13 ทดสอบ แก้ไขวัตถุดิบ	62
ตาราง 4.14 ทดสอบ ลบวัตถุดิบ	63
ตาราง 4.15 ทดสอบ เพิ่มอาหาร	63
ตาราง 4.16 ทดสอบ แก้ไขอาหาร	64
ตาราง 4.17 ทดสอบ ลบอาหาร	64
ตาราง 4.18 ทดสอบ เพิ่มเมนูอาหาร	65
ตาราง 4.19 ทดสอบ แก้ไขเมนูอาหาร	66
ตาราง 4.20 ทดสอบ ลบเมนูอาหาร	66
ตาราง 4.21 ทดสอบ เพิ่มแพลตฟอร์ม	67

---

ตาราง 4.22 ทดสอบ ลบแพลตฟอร์ม	67
ตาราง 4.23 ทดสอบ ตั้งราคาเมนูอาหาร	68
ตาราง 4.24 แบบประเมินความพึงพอใจของพนักงานร้าน	74
ตาราง 4.25 แบบประเมินความพึงพอใจของเจ้าของร้าน	75

# บทที่ 1

## บทนำ

### 1.1. ความเป็นมาและความสำคัญของโครงการ

จากเหตุการณ์การแพร่ระบาดของเชื้อโรคโควิด-19 ส่งผลให้ธุรกิจส่งอาหารเดลิเวอรี่เติบโตมากขึ้น สะท้อนอย่างมากในประเทศไทย แม้แต่ในช่วงสถานการณ์โควิด-19 ก็ยังคงเติบโตอย่างต่อเนื่อง ทำให้ผู้ประกอบการจำนวนมากหันมาสนใจในแพลตฟอร์มเหล่านี้ และเกิดเป็นธุรกิจร้านอาหารแบบใหม่ ที่ไม่มีหน้าร้าน ไม่มีพนักงาน รับคำสั่งซื้อผ่านระบบ online delivery อย่างเดียว

ในปัจจุบันแพลตฟอร์ม การส่งอาหารออนไลน์ในประเทศไทยมีหลากหลายเป็นอย่างมาก เช่น GrabFood, LinemanWongnai, ShopeeFood, Robinhood โดยแต่ละบริษัทก็มีระบบการให้บริการที่แตกต่างกันไป ส่งผลให้การคิดรายรับ รายจ่ายต่างๆในร้านค้าเกิดความซับซ้อนมากกว่าการขายหน้าร้านปกติทั่วไป

ผู้จัดทำโครงการจึงมีแนวคิดในการพัฒนา “ระบบจัดการร้านอาหารเดลิเวอรี่” โดยมุ่งเป้าไปที่การสร้างระบบหลังบ้านที่สามารถจัดการร้านอาหารที่ใช้ระบบส่งออนไลน์เป็นหลักได้อย่างมีประสิทธิภาพ สามารถทำบัญชีและจัดการเงินเข้าออก จากแอปพลิเคชัน ต่างๆได้ง่าย โดยมีการเพิ่ม กฎการคิดบัญชีของแต่ละแอปพลิเคชันขึ้นมา ซึ่งถือเป็น feature สำคัญของระบบ เก็บบันทึกสถิติการขายเมนูต่างๆของทุกแอปพลิเคชัน เพื่อทำแบบวิเคราะห์การขายและออกใบเรียกเข้า ใบเบี้ยนค่าใช้จ่าย ใบอนุมัติการจ่ายเงิน ฯลฯ ที่สำคัญคือ สามารถจัดการร้านอาหารเดลิเวอรี่ได้โดยตรง ไม่ต้องผ่านคนกลาง ลดต้นทุนลง ทำให้ร้านอาหารสามารถลดต้นทุนลงได้

### 1.2. วัตถุประสงค์ของโครงการ

ระบบจัดการร้านอาหารเดลิเวอรี่สามารถ

- 1.) จัดเก็บข้อมูลสูตรอาหารและวัตถุติดต่อต่างๆภายในร้าน
- 2.) กำหนดราคาอาหารและค่าธรรมเนียมของแต่ละแพลตฟอร์มแยกกันได้
- 3.) บันทึกรายจ่ายภายในร้านโดยแบ่งประเภทของรายจ่ายได้ตามที่ผู้ใช้งานต้องการ
- 4.) บันทึกคำสั่งซื้อได้ง่ายและรวดเร็ว ผ่านการคลิกหรือสัมผัสโดยไม่ต้องพิมพ์ค่าต่างๆ
- 5.) แสดงข้อมูลสถิติคำสั่งซื้อจากแพลตฟอร์มต่างๆได้ตามระยะเวลาที่กำหนด

6.) แสดงข้อมูลรายจ่ายภายในร้านเป็นตารางที่สามารถจัดเรียงหรือกรองข้อมูลได้ตามที่ต้องการ

### 1.3. ขอบเขตโครงการ

#### 1.3.1. ด้านประชากร

- 1.) เจ้าของร้าน ‘スマยดอง’
- 2.) พนักงานร้าน ‘スマยดอง’

#### 1.3.2. ด้านเทคโนโลยีที่ใช้ในการพัฒนาโครงการ

Hardware

- Desktop Computer (RAM 16GB, AMD FX-8320E, NVIDIA GeForce GTX 1050 Ti)
- Smart Phone (Xiaomi 10T pro)

2.) Software

- Microsoft Windows 10 Pro build 19043
- Visual Studio Code 1.66.2
- Notepad++ 8.3.3
- Github Desktop 3.1.2
- MongoDB Atlas
- NodeJS 16.14.2 LTS
- Express.js 4.0
- React 18.0.0
- Redux 8.0.2
- ReduxjsToolkit 1.8.3
- MaterialUI 5.10.10
- Axios 1.1.3

- JsonWebToken 8.5.1

- Mongoose 6.4.5

- dayjs 1.11.5

- Google Charts 4.0.0

### 3.2.3 Technology

- Model View Controller

- REST API

- Cloud Database

- Node.JS

- Framework

- HTTPS Request

- State management

- Immutable data

## 1.4. ขอบเขตของโครงงาน

1. ระบบตรวจสอบสิทธิในการเข้าใช้งานซอฟต์แวร์ (Authentication)

2. การตั้งค่าร้านอาหาร

2.1. ตั้งค่าสู่ตรอาหาร

2.1.1. จัดการวัตถุดิบและบรรจุภัณฑ์

2.1.1.1. จัดเก็บ ซื้อ และหน่วย

2.1.1.2. สามารถ สร้าง แก้ไข ลบ และค้นหาได้

2.1.2. จัดการอาหาร

2.1.2.1. จัดเก็บ ซื้อ และวัตถุดิบที่ประกอบอาหาร

2.1.2.2. สามารถ สร้าง แก้ไข และลบได้

### 2.1.3. จัดการเมนูอาหาร

2.1.3.1. จัดเก็บ ซื้อ อาหารในเมนู ค่าแรง และค่าน้ำมัน

2.1.3.2. สามารถ สร้าง แก้ไข และลบได้

### 2.2. ตั้งค่าแพลตฟอร์ม

2.2.1. จัดเก็บ ซื้อ ค่าธรรมเนียม ภาษี ค่าน้ำมัน ราคาของเมนูต่างๆ

2.2.2. สามารถ สร้าง และลบแพลตฟอร์มได้

2.2.3. จัดการราคาของเมนูต่างๆของแต่ละแพลตฟอร์มได้

## 3. การจัดการคำสั่งซื้อและรายจ่ายภายในร้าน

### 3.1. รายจ่าย

3.1.1. การจัดการรายจ่าย

3.1.1.1. จัดเก็บหมวดหมู่ของรายจ่าย คำอธิบาย จำนวน ราคา และวันที่รายจ่ายเกิด

3.1.1.2. สามารถ สร้าง แก้ไข ลบ และค้นหาได้

3.1.2. การจัดการหมวดหมู่รายจ่าย

3.1.2.1. แต่ละหมวดหมู่สามารถมีหมวดหมู่รองได้

3.1.2.2. สามารถ สร้าง แก้ไข และลบได้

3.1.2.3. สามารถสร้างรายการซื้อได้ง่าย

### 3.2. การจัดการคำสั่งซื้อ

3.2.1. จัดเก็บ แพลตฟอร์ม รายการอาหาร จำนวน และวันที่เกิดคำสั่งซื้อ

3.2.2. สามารถ เพิ่ม ลบ และค้นหาได้

3.2.3. การเพิ่มจะต้องสามารถเพิ่มได้โดยไม่จำเป็นต้องใช้เบื้องต้นพิมพ์

## 4. สามารถแสดงสถิติการขาย

4.1. สามารถแสดงสถิติรวม หรือแยกแต่ละแพลตฟอร์มได้

4.2. สามารถเลือกขอบเขตเวลาเป็น วัน สัปดาห์ เดือน หรือปีได้

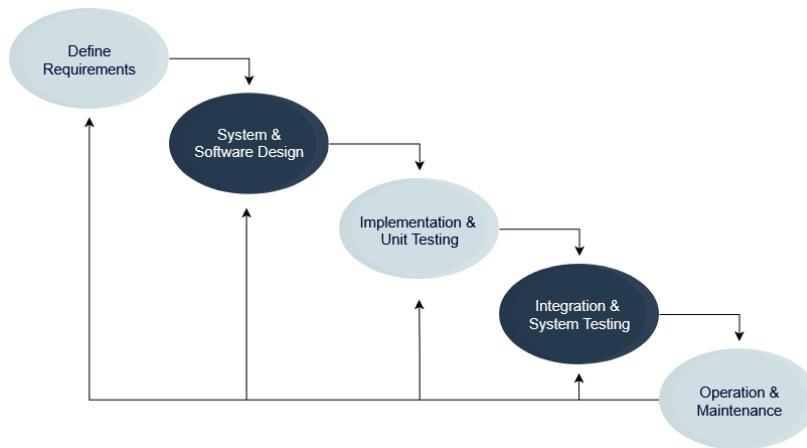
4.3. สามารถแสดงข้อมูลเป็นตารางได้

4.4. สามารถแสดงข้อมูลเป็นกราฟแท่งได้

## 1.5. กรอบแนวคิดการพัฒนาโครงการ

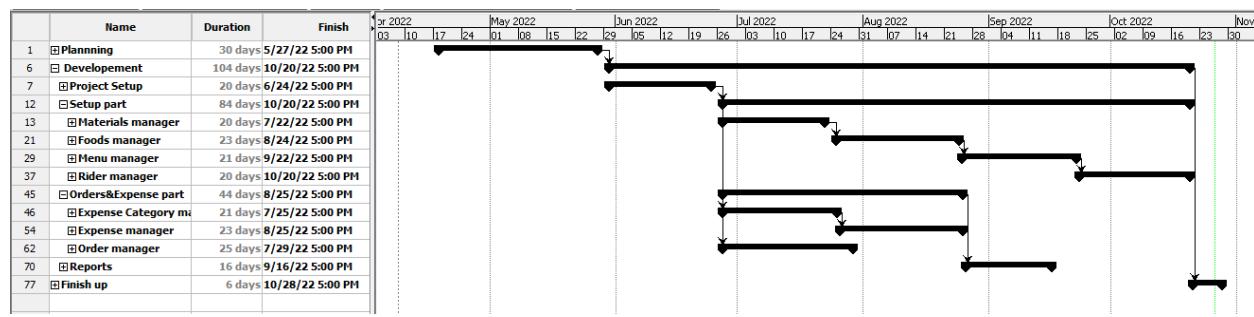
### 1.5.1. แบบจำลองกระบวนการ (Software Process Model)

เลือกใช้ Waterfall Model ซึ่งประกอบไปด้วย การวิเคราะห์ความต้องการ การออกแบบระบบ การพัฒนาระบบ การทดสอบระบบ การนำระบบไปใช้ และการปรับปรุงแก้ไข เนื่องจากต้องจัดทำเอกสารโครงการ ซึ่งจำเป็นต้องมีการวางแผนการพัฒนาล่วงหน้าอยู่แล้ว โมเดล Waterfall ที่ต้องการวางแผนล่วงหน้าจึงจัดได้ว่า เหมาะสมกับรูปแบบการทำงานมาก



รูปประกอบ 1.1 Waterfall model

### 1.5.2. แผนภูมิแกนต์ (Gantt Chart)



รูปประกอบ 1.2 Gantt Chart

## 1.6. ประโยชน์ที่ได้รับ

- 1.) ร้านอาหารมีระบบฐานข้อมูลสำหรับเก็บสูตรอาหาร
- 2.) ร้านอาหารสามารถบันทึกคำสั่งชื่อด้วยง่าย

- 3.) ร้านอาหารสามารถบันทึกรายจ่ายโดยแบ่งเป็นประเภทได้
- 4.) ร้านอาหารมีสถิติการขายที่สามารถดูแยกตามแพลตฟอร์มและระยะเวลาได้

## 1.7. นิยามคัพท์เฉพาะ

ผู้ใช้ หมายถึง ผู้จัดการหรือเจ้าของร้านสมายดอง

ลูกค้า หมายถึง ผู้ใช้บริการแพลตฟอร์มส่งอาหาร

ไรเดอร์ หมายถึง แพลตฟอร์มหรือบริษัทที่รับบทบาทในการรับและส่งอาหารจากร้านอาหารไปยังลูกค้า เช่น Grab, Lineman, ShopeeFood, FoodPanda

ออดิโอ หมายถึง คำสั่งซื้อที่ร้านค้าได้รับจากไรเดอร์ โดยในคำสั่งซื้อจะระบุถึงเมนูอาหาร และจำนวนที่ลูกค้าต้องการ

## บทที่ 2

# แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง (Literature Review)

### 2.1 แนวคิดทฤษฎี

#### 2.1.1. ภาษา HTML

เนื้อหา :

HTML ย่อมาจาก Hypertext Markup Language มันคือภาษาเขียนเว็บไซต์ที่ใช้เพื่อกำกับข้อมูลต่างๆ และแสดงผลคำสั่งบนหน้า Web Browser เป็นเหมือนภาษาพื้นฐานที่มีไว้ในการพัฒนาหน้าเว็บไซต์ ในปัจจุบันอยู่ภายใต้การดูแลของ องค์กร World Wide Web Consortium (W3C) และถูกพัฒนามาถึงเวอร์ชัน 5 หรือที่เรียกว่า HTML5 โดยเริ่มจาก HTML1 ที่ทำอะไรได้ไม่มากนอกจากโบว์ชาร์ออนไลน์

ข้อมูลต่างๆ บนหน้าเว็บไซต์จะถูกเชื่อมโยงกันด้วยชุดคำสั่งต่างๆ เพื่อให้แสดงผลออกมาในรูปแบบที่นักออกแบบต้องการให้เป็น ข้อมูลเหล่านั้นถูกควบคุมโดยการเขียน HTML ทั้งที่เป็นรูปภาพ ข้อความ หรือวัตถุอื่นๆ บนหน้าเว็บไซต์

โครงสร้างของ HTML จะเป็นในรูปแบบของ Tag ต่างๆ และ Web Browser จะแปลความของ Tag แต่ละ Tag ออกมาเป็นหน้าตาเว็บไซต์

แหล่งอ้างอิง :

Jirayut Intachai. (2564). HTML คืออะไร? ทำไมคนเขียนเว็บไซต์ต้องใช้งาน. จาก Terrestrial: <https://goterrestrial.com/2021/05/19/what-is-html/>

สรุปความรู้ :

HTML เป็นภาษาที่ความซับซ้อนต่ำและไม่มี Dynamic แต่เป็นรากฐานสำคัญในการเขียนเว็บแอปพลิเคชัน เพราะเป็นส่วนที่ใช้กำหนดโครงสร้างหลักของเว็บ

### 2.1.2. ภาษา CSS

เนื้อหา :

CSS คือ ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง พื้นหลัง เส้นขอบและอื่นๆ ตามที่ต้องการ CSS ย่อมาจาก Cascading Style Sheets มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกกำหนดมาตรฐานโดย W3C เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ ได้รับความนิยมอย่างแพร่หลาย

CSS มีประโยชน์อย่างหลากหลาย ซึ่งได้แก่

- ช่วยให้เนื้อหาภายในเอกสาร HTML มีความเข้าใจได้ง่ายขึ้นและในการแก้ไขเอกสารก็สามารถทำได้ง่ายกว่าเดิม เพราะการใช้ CSS จะช่วยลดการใช้ภาษา HTML ลงได้ในระดับหนึ่ง และแยกระหว่างเนื้อหา กับรูปแบบในการแสดงผลได้อย่างชัดเจน
- ทำให้สามารถดาวน์โหลดไฟล์ได้เร็ว เนื่องจาก code ในเอกสาร HTML ลดลง จึงทำให้ไฟล์มีขนาดเล็กลง
- สามารถกำหนดรูปแบบการแสดงผลจากคำสั่ง style sheet ชุดเดียวกัน ให้มีการแสดงผลในเอกสารแบบเดียวกันทั้งหน้าหรือในทุกๆ หน้าได้ ช่วยลดเวลาในการปรับปรุงและทำให้การสร้างเอกสารบนเว็บมีความรวดเร็วยิ่งขึ้น นอกจากนี้ยังสามารถควบคุมการแสดงผลให้คล้ายหรือเหมือนกันได้ในหลาย Web Browser
- ช่วยในการกำหนดการแสดงผลในรูปแบบที่มีความเหมาะสมกับสื่อต่างๆ เป็นอย่างดี
- ทำให้เว็บไซต์มีความเป็นมาตรฐานมากขึ้นและมีความทันสมัย รองรับการใช้งานในอนาคตได้ดี

แหล่งอ้างอิง :

Kipakaporn. (2561). CSS คืออะไร มีประโยชน์อย่างไร. จาก SoGoodWeb:

<https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์-อย่างไร>

สรุปความรู้ :

ภาษา CSS เป็นภาษาสำหรับตกแต่งและออกแบบ UI ของ เว็บแอปพลิเคชัน แต่ตัวภาษาเองไม่สามารถทำงานได้หากขาดโครงสร้างเว็บจาก HTML

ภาษา CSS มีเครื่องมือให้เลือกใช้เป็นจำนวนมาก ทำให้เป็นภาษาที่เรียนรู้ขั้นพื้นฐานนั้นง่าย แต่การจะเข้าใจและใช้ได้อย่างเต็มประสิทธิภาพต้องอาศัยประสบการณ์และความรู้ความเข้าใจเป็นอย่างมาก

### 2.1.3. ภาษา JavaScript

เนื้อหา :

JavaScript คือ ภาษาคอมพิวเตอร์ภาษาหนึ่งที่มีไว้เพื่อจัดการเอฟเฟกต์หรือพัฒนาของหน้าเว็บไซต์ นอกเหนือจาก HTML ที่ใช้จัดการเรื่องของเนื้อหาของเว็บและ CSS ที่ใช้จัดการเรื่องของโครงสร้างและดีไซน์ของเว็บ

JavaScript ยังสามารถใช้งานในส่วนของโปรแกรมบนคอมพิวเตอร์และเซิฟเวอร์ ยกตัวอย่างเช่น Node.js หรือในฐานข้อมูลกีฬาสามารถใช้งานได้ เช่นกัน เช่น MongoDB และ CouchDB

JavaScript สามารถช่วยเราจัดการงานต่างๆ โดยไม่จำเป็นที่จะต้องทำการโหลดหน้าเว็บใหม่ซ้ำๆ หากเราเคยเห็นการกดปุ่ม Button แล้ว ข้อความบางข้อความแสดงผลหรือซ่อนข้อความนั้นโดยไม่จำเป็นต้องโหลดหน้านั้นใหม่อีกครั้ง หรือมีการเปลี่ยนสีของตัวอักษร เปลี่ยนแปลงเนื้อหา นั่นล้วนรับ คือ หน้าเว็บไซต์นั้นๆ ถูกเรียกใช้งานด้วย JavaScript แล้ว

แหล่งอ้างอิง :

Worapong. (2563). **JavaScript คืออะไร ใช้งานอย่างไร.** จาก WebDoDee:

<https://www.webdodee.com/what-is-javascript/>

สรุปความรู้ :

ภาษา JavaScript เป็นภาษาสำหรับเขียนเว็บไซต์ที่สามารถเรียกได้ว่าเป็นภาษาสำหรับเขียนโปรแกรมจริงๆ เมื่อเทียบกับ HTML ที่ไม่สามารถเรียกได้ว่าภาษาสำหรับเขียนโปรแกรมได้เต็มปาก

แรกเริ่มเดิมที่ JavaScript ถูกพัฒนาเพื่อช่วยให้สามารถปรับปรุง/แก้ไขหน้าเว็บได้โดยที่ไม่ต้องโหลดหน้าเว็บใหม่อีกครั้ง โดยการแก้ไขส่วนต่างๆ ของ HTML โดยตรง

ปัจจุบันภาษาถูกพัฒนาจนสามารถใช้ในการเขียนโปรแกรมได้โดยตรงโดยไม่ต้องผ่าน web browser ทำให้การใช้งานมีความหลากหลายมากขึ้น ใช้เขียนโปรแกรมก็ได้ ใช้เป็น Backend ก็ได้ JavaScript จึงได้รับความนิยมมากยิ่งขึ้น

#### 2.1.4. Framework

เนื้อหา :

เฟรมเวิร์ก (Framework) หรือ โครงสร้างซอฟต์แวร์ เป็นขอบเขตของระบบงาน ที่เป็นรูปแบบที่นำกลับมาใช้ใหม่ได้สำหรับระบบซอฟต์แวร์ (หรือระบบย่อย) ซึ่งสามารถถ่ายโอนรูปของคลาสนามธรรม (Abstract class) และกับวิธีในการใช้ตัวตน (instance) ของคลาสร่วมกันจำเพาะสำหรับซอฟต์แวร์ชนิดใดชนิดหนึ่ง โครงสร้างซอฟต์แวร์ทุกโครงสร้างใช้การออกแบบเชิงวัตถุ(Object-oriented programming, OOP)

โปรแกรมของโครงสร้างซอฟต์แวร์มักจะเป็นโปรแกรมเชิงวัตถุ และ PHP Framework ในปัจจุบัน ส่วนใหญ่อยู่ภายใต้แนวคิด MVC (Model-view-controller)

แหล่งอ้างอิง :

Framework คืออะไร โครงสร้างหรือขอบเขตของระบบงาน. (2561). จาก MindPHP:

<https://www.mindphp.com/developer/25-oop-php-framework/1957-framework-คืออะไร.html>

สรุปความรู้ :

เฟรมเวิร์กคือโครงสร้างสำหรับการเขียนโปรแกรม ช่วยให้โปรแกรมมีมาตรฐานและเป็นระเบียบ และ Framework ส่วนใหญ่ยังนำเสนอเครื่องมือและโมดูลต่างๆ ที่ช่วยให้การเขียนโปรแกรมง่ายขึ้นหรือดีขึ้นอีกด้วย

#### 2.1.5. MVC Model

เนื้อหา :

MVC เป็นตัวย่อของคำว่า Model View Controller ใช้เรียกรูปแบบการพัฒนาซอฟต์แวร์ที่มีโครงสร้างซึ่งแบ่งออกมาเป็น 3 ส่วนหลัก ตามตัวย่อของชื่อ รูปแบบการพัฒนาซอฟต์แวร์แบบ MVC ถูกนำไปใช้ในขั้นตอนการพัฒนาหลากหลายภาษา เพราะ MVC เป็นเพียงหลักการออกแบบโปรแกรม ( Design Pattern ) รูปแบบหนึ่งเท่านั้น ซึ่งเป็นที่นิยมมากในการนำมาพัฒนาแอปพลิเคชัน ซอฟต์แวร์แต่ละแพลตฟอร์ม และประยุกต์ใช้ในอีกหลาย ๆ ด้าน

model คือส่วนของการเก็บรวบรวมข้อมูล ไม่ว่าข้อมูลนั้น ๆ จะถูกจัดเก็บในรูปแบบใดก็ตาม ในฐานข้อมูลแบบเป็น Object Class หรือที่นิยมเรียกว่า VO ( Value Object ) หรือเก็บเป็นไฟล์ข้อมูลเลยเมื่อข้อมูลถูกโหลดเข้ามาจากที่ต่าง ๆ และเข้ามายังส่วนของโมเดล ตัวโมเดลจะทำการจัดการระหว่างข้อมูลให้เป็นรูปแบบที่เหมาะสม เพื่อรอการร้องขอข้อมูลจากส่วนของ Controller

view คือส่วนของการแสดงผล หรือส่วนที่จะปฏิสัมพันธ์กับผู้ใช้งาน ( User Interface ) หน้าที่ของ view ในการเขียนโปรแกรมแบบ MVC คืออยู่รับคำสั่งจากส่วนของ Controller และ End User เริ่มแรกเลยตัววิวอาจจะได้รับคำสั่งจาก Controller ให้แสดงผลหน้า Home และเมื่อผู้ใช้งานหน้าเว็บกดปุ่มสั่งซื้อ View จะส่งข้อมูลไปให้ Controller เพื่อประมวลผลและแสดงบางอย่างจาก Action นั้น

controller คือส่วนของการเริ่มทำงาน และรับคำสั่ง โดยที่คำสั่นนั้นจะเกิดขึ้นในส่วนการติดต่อ กับผู้ใช้งานคือ view เมื่อผู้ใช้งานทำการ Interactive กับ UI view จะเกิดเหตุการณ์หรือข้อมูลบางอย่างขึ้น ตัววิวจะส่งข้อมูลนั้นมา�ัง controller ตัว controller จะทำการประมวลผลโดยบางคำสั่งอาจจะต้องไปติดต่อกับ model ก่อน เพื่อทำการประมวลผลข้อมูลอย่างถูกต้องเรียบร้อยแล้วก็จะส่งไปยัง view เพื่อแสดงผลตามคำสั่งที่ end user ร้องขอมา Controller จะทำหน้าที่เป็นตัวกลางระหว่าง Model และ View ให้ทำงานร่วมกันอย่างมีประสิทธิภาพและตรงกับความต้องการของ End User หากที่สุด

### แหล่งอ้างอิง :

MVC คืออะไร ทำความเข้าใจรูปแบบการเขียนโปรแกรม. (2559). จาก บริษัท โค๊ดบี จำกัด: <https://www.codebee.co.th/labs/mvc-คืออะไร-ทำความเข้าใจรูปแบบ/>

### สรุปความรู้ :

MVC คือรูปแบบการพัฒนาซอฟต์แวร์โดยการแบ่งโครงสร้างออกเป็น 3 ส่วน คือ

1. Model ส่วนของการกำหนดรูปแบบการจัดเก็บข้อมูล
2. View ส่วนของการแสดงผล
3. Controller ส่วนของการประมวลผล

#### 2.1.6. ฐานข้อมูล MongoDB

##### เนื้อหา :

MongoDB เป็น open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือไม่มี relation (ความสัมพันธ์) ของตารางแบบ SQL ทั่วไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทน การบันทึกข้อมูลทุกๆ record ใน MongoDB เราจะเรียกว่า Document ซึ่งจะเก็บค่าเป็น key และ value จะเห็นว่ามันก็คือ JSON นั่นแหละ

การเก็บข้อมูล document ใน MongoDB จะถูกเก็บไว้ใน Collections (เปรียบเทียบได้กับ Table ใน Relational Database ทั่วไป) แต่แตกต่างกันที่ collection ไม่จำเป็นที่จะต้องมี schema เมื่อ он กัน ก็สามารถบันทึกข้อมูลได้

Schemaless คือการไม่ต้องกำหนดโครงสร้างใดๆให้มันเหมือน SQL ปกติทั่วไป เช่น collection User มีเก็บแค่ name ต่อมาเราสามารถเพิ่มการเก็บ position เข้ามาได้เลย

**แหล่งอ้างอิง :**

Chai Phonbopit. (2558). **MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น.** จาก Devahoy:  
<https://www.devahoy.com/blog/2015/08/getting-started-with-mongodb>

**สรุปความรู้ :**

MongoDB คือ ระบบฐานข้อมูล opensource แบบ NoSQL ที่มีการจัดเก็บข้อมูลแบบ Document ซึ่งเก็บค่าเป็น Key และ Value คล้ายกับระบบ JSON

MongoDB เป็นฐานข้อมูลแบบ Schemaless คือ ไม่มีการกำหนดโครงสร้างข้อมูล ทำให้ฐานข้อมูลมีความยืดหยุ่นสูงและขยายได้ง่าย

### 2.1.7. Framework Express.js

**เนื้อหา :**

Express.js เอ็กเพรส ดอทเจอส เป็น Web Application Framework ชื่อดังที่ได้รับความนิยมมาก สำหรับทำงานบน platform ของ Node.js ซึ่งเป็น Server ตัวหนึ่ง โดยทั้ง Express.js และ Node.js ต่างก็ใช้ภาษา Javascript ในการพัฒนา ถ้าเป็น Web Application Framework ในสมัยก่อน คนที่พัฒนาจะต้องมีความรู้มากกว่า 1 ภาษา ภาษาที่ทำงานทางฝั่ง Server อย่าง PHP หรือ ASP และภาษาที่ทำงานทางฝั่ง Client อย่าง JavaScript เพื่อลดความยุ่งยากรวมถึงเวลาในการต้องเรียนรู้หลายภาษา ภาษาทำให้เกิด Node.js กับ Express.js เพียงแค่มีความรู้ Javascript ก็สามารถเขียนได้ทั้ง Server และ Client นอกจากนี้ถ้าใครเคยเขียน Javascript จะ

รู้ว่ามีการตอบสนองที่รวดเร็ว แน่นอนว่า Express.js ก็ยกมาเป็นข้อเด่นในเรื่องความเร็ว ในเรื่องการการเรียนรู้การเขียน Express.js จะใช้รูปแบบที่ง่ายในการเรียนรู้มากที่สุด

สำหรับการ พัฒนา Express.js ในเว็บไซต์จะพูดถึงการใช้ Routing (การกำหนดเส้นทางของระบบ) และ Middleware (การรับส่งข้อมูลของระบบ) สามารถเขียนได้ในรูปแบบ MVC ส่วนการเข้ามือต่อกับฐานข้อมูลสามารถใช้ MongoDB หรือจะใช้ MySql ก็ได้ สำหรับนามสกุลของไฟล์คือ .js ขณะนี้ได้พัฒนามาถึงเวอร์ชันที่ 4.x แล้ว

แหล่งอ้างอิง :

Ae. (2559). Express.js เอ็กเพรส คอทเจอส คืออะไร Express.js คือ เป็น Web Application Framework ซึ่งดังที่ได้รับความนิยมมาก สำหรับทำงานบน platform ของ Node.js. จาก MindPHP: <https://www.mindphp.com/คู่มือ/73-คืออะไร/3874-what-is-express-js.html>

สรุปความรู้ :

Express.js คือ Framework สำหรับ Web Application โดยทำงานบน Node.js ด้วยภาษา JavaScript. เมามะสำหรับการทำระบบหลังบ้านและ API

#### 2.1.8. Framework React

เนื้อหา :

React เป็น JavaScript Library หรือจะเรียกว่าเป็น JavaScript Framework ที่ใช้สำหรับสร้างหน้าเว็บของเราให้อย่างมีประสิทธิภาพ พร้อมด้วย action ต่างๆ ที่ทำให้เราสามารถอ่านและเขียนได้

React นั้นเป็น JavaScript Framework ที่ทำหน้าที่ Compile Code ให้กลายเป็น HTML + CSS + JavaScript เพื่อให้สามารถใช้งานบน Browser ได้ โดยที่ตัว React นั้นอาศัย NodeJS เพื่อทำการ Compile Code

จุดเด่นของ React ที่ทำให้น่าสนใจใช้งานนั้นก็คือ การที่มีระบบแคลชในตัวทำให้หน้าเว็บมีการตอบสนองที่เร็ว หมายความว่าเราสามารถนำมารันได้โดยทันที ไม่ต้องรอโหลดหน้าเว็บ ทำให้เราสามารถเปลี่ยนหน้าเว็บได้เร็วๆ แค่กดปุ่ม

ส่วนๆ เรียกว่าเป็น component แล้วนำมาประกอบกันเป็นหน้าเว็บได้ ซึ่งทำให้สามารถ นำ component ของไปใช้ซ้ำที่อื่นได้ ไม่ต้องเสียเวลาเขียนใหม่

**แหล่งอ้างอิง :**

Mos Programmer. (2562). **มาตรฐาน React กันเถอะ.** จาก Medium:

<https://mosprogramer.medium.com/มาตรฐาน-react-กันเถอะ-eb04ab8b8ec1>

**สรุปความรู้ :**

React คือ JavaScript Framework สำหรับการสร้างส่วนหน้าของ Web Application มีจุดเด่น ที่การมีระบบแแคชในตัว ทำให้หน้าเว็บตอบสนองอย่างรวดเร็วแล้วปรับแต่งได้ง่าย และยังแยกองค์ประกอบของเว็บออกเป็นส่วนๆ ซึ่งแต่ละส่วนสามารถนำไปใช้ซ้ำได้ ไม่ต้องเขียนใหม่

#### 2.1.9. Node.js

**เนื้อหา :**

Node.js คือสภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ที่ทำงานด้วย V8 engine นั้นหมายความว่าเราสามารถใช้ Node.js ในการพัฒนาแอพพลิเคชันแบบ Command line และพลิกเช่น Desktop หรือแม้แต่เว็บไซต์ได้ โดยที่ Node.js จะมี APIs ที่เรามาสามารถใช้สำหรับทำงานกับระบบปฏิบัติการ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ก เป็นต้น

Node.js ถูกพัฒนาและทำงานด้วยใช้ Chrome V8 engine สำหรับคอมไพล์ภาษา JavaScript ให้เป็นภาษาเครื่องด้วยการคอมไพล์แบบ Just-in-time (JIT) เพื่อเพิ่มประสิทธิภาพการทำงานของภาษา JavaScript จากที่แต่เดิมมันเป็นภาษาที่มีการทำงานแบบ Interpreted

Node.js เป็นโปรแกรมที่สามารถใช้ได้ทั้งบน Windows, Linux และ Mac OS X นั้นหมายความว่าคุณสามารถเขียนโปรแกรมในภาษา JavaScript และนำไปรันได้ทุกระบบปฏิบัติการที่สนับสนุนโดย Node.js นี้ เป็นแนวคิดของการเขียนครั้งเดียวแต่ทำงานได้ทุกที่ (Write once, run anywhere)

**แหล่งอ้างอิง :**

ทำความรู้จักกับ Node.js. (2564). จาก MarcusCode:

<http://marcuscode.com/tutorials/nodejs/introducing-nodejs>

### สรุปความรู้ :

Node.js คือสภาพแวดล้อมการทำงานของภาษา JavaScript นอกเว็บเบราว์เซอร์ ถูกพัฒนาและทำงานด้วยการใช้ Chrome V8 engine สำหรับคอมไพล์ภาษา JavaScript ให้เป็นภาษาเครื่องด้วยการคอมไพล์แบบ Just-in-time (JIT)

### สรุปแนวคิดทฤษฎี :

เนื่องจากผู้จัดทำต้องการให้ระบบจัดการร้านอาหารเดลิเวอรี่สามารถเข้าถึงได้ง่าย ใช้งานได้ทั้งในคอมพิวเตอร์และมือถือโดยไม่ต้องติดตั้งระบบ จึงตัดสินใจที่จะทำระบบนี้ในรูปแบบของ web application ที่มีความ responsive และสามารถขยายและพัฒนาต่อไปได้ง่ายในอนาคต

หลักจากศึกษาแนวคิดทฤษฎีการพัฒนาเว็บไซต์ จึงสรุปได้ว่า การพัฒนาเว็บไซต์จะต้องใช้ HTML เป็นพื้นฐานโครงสร้างหลัก, ใช้ CSS ในการทำให้เว็บไซต์มี UX ที่ดี มีลูกเล่นต่างๆ, ต้องมีภาษาได้อีกภาษาหนึ่งในการพัฒนาระบบหลังบ้านเพื่อประมวลผลข้อมูลต่างๆที่ผู้ใช้ส่งเข้ามา และยังต้องมีระบบฐานข้อมูลในการเก็บข้อมูลอีกด้วย

แต่การพัฒนาเว็บไซต์โดยการเขียนระบบใหม่หมดตั้งแต่ต้น ทั้งเบื้องหน้า (Frontend : HTML, CSS) และเบื้องหลัง (Backend : ระบบหลังบ้าน, ระบบฐานข้อมูล) หากผู้พัฒนาไม่มีความรู้ความสามารถในภาษาต่างๆและการจัดการระบบมากพอ ระบบจะมีความซับซ้อนและไม่เป็นระเบียบ ทำให้การขยาย การปรับปรุง หรือการส่งต่อระบบเป็นไปได้ยากหรือเชื่องช้า ผู้จัดทำจึงศึกษาทฤษฎี Framework และนำมาใช้ในการพัฒนา

หลักจากศึกษา Framework ต่างๆ ผู้จัดทำตัดสินใจใช้ MERN สเต็คในการพัฒนาระบบนี้ MERN สเต็คเป็นหนึ่งสเต็คที่ได้รับความนิยมสูงในปัจจุบัน ซึ่งประกอบไปด้วย React.js Framework สำหรับพัฒนาระบบเบื้องหน้า, Express.js Framework สำหรับพัฒนาระบบเบื้องหลัง, Node.js ระบบเบื้องหลังที่ช่วยให้ภาษา JavaScript สามารถทำงานภายนอกเว็บбраузர์ได้ และสุดท้ายคือระบบฐานข้อมูลแบบ No-SQL MongoDB

ผู้จัดทำตัดสินใจใช้สเต็คนี้เนื่องจาก 1.) ระบบ Framework ทั้งเบื้องหน้าและเบื้องหลังเป็นภาษาเดียวกัน ทำให้การสื่อสารระหว่างสองฝ่ายเป็นไปได้ง่าย 2.) ระบบฐานข้อมูลแบบ No-SQL มีความยืดหยุ่นสูงและ

สามารถปรับเปลี่ยนแก้ไขได้ง่าย 3.) MERN Stack ได้รับความนิยมสูงทั้งจากผู้ว่าจ้างและผู้พัฒนาในสังคมผู้พัฒนาเว็บแอปพลิเคชัน ทำให้เทคโนโลยีต่างๆมีการบำรุงรักษาและสนับสนุนสูง 4.) เทคโนโลยีส่วนใหญ่ใน MERN Stack เป็นเทคโนโลยีที่ผู้จัดทำไม่เคยใช้งานแต่ต้องการเรียนรู้

## 2.2. งานวิจัยที่เกี่ยวข้อง

### 2.2.1. Restaurant Management System

เนื้อหา :

ภาคนิพนธ์นี้ออกแบบและพัฒนาระบบจัดการร้านอาหารให้มีประสิทธิภาพมากยิ่งขึ้น ทั้งในส่วนของการจัดเก็บข้อมูล เพื่อป้องกันการสูญหายของข้อมูลต่างๆ และช่วยให้การปฏิบัติงานมีความสะดวกรวดเร็วมากยิ่งขึ้น การพัฒนาระบบจัดการร้านอาหารโดยใช้โปรแกรม Microsoft Visual Studio 2017 ในการออกแบบ พัฒนาระบบจัดการร้านอาหาร ใช้ภาษา C# ในการเขียนโปรแกรมและใช้การจัดการฐานข้อมูลโดยใช้โปรแกรม Microsoft SQL Server 2017 ระบบจัดการร้านอาหารสามารถใช้งานได้สะดวก เช่น การจองโต๊ะ ตัดจำหน่ายวัตถุดิบ เพิ่มสูตรรับประทาน รายงานสรุปยอดแบบเลือกวันที่ ซึ่งผลที่ได้รับจากการพัฒนาระบบแสดงให้เห็นว่าระบบสามารถทำงานได้ตรงขอบเขต ลดข้อผิดพลาดในการให้บริการ ตลอดจนทำให้ลดความซับซ้อนของการสั่งซื้อ และจัดการวัตถุดิบพร้อมทั้งการตัดจำหน่ายวัตถุดิบให้มีประสิทธิภาพมากขึ้น

แหล่งอ้างอิง :

ปริญญา อภัยภักดี และณัฐพล เสาวพงษ์. ระบบจัดการร้านอาหาร. (ภาคนิพนธ์ปริญญาตรี คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยสยาม, 2562)

### 2.2.2. ระบบจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero

เนื้อหา :

โครงการนี้มีวัตถุประสงค์เพื่อ พัฒนาระบบบริหารจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero ซึ่งสามารถแยกออกได้เป็น 3 ส่วนหลักๆ คือ ส่วนของการจัดการระบบเมนูอาหาร ส่วนของการทำรายการขาย และส่วนของการอกรายงานยอดการขาย โดยส่วนของการจัดการระบบเมนูอาหาร จะมีหน้าที่ในการเก็บข้อมูลเมนูอาหารทั้งหมดของร้าน ส่วนในด้านของการทำรายการขายเป็นส่วนที่สำคัญเนื่องจากเป็นการคูณ

รายละเอียดการขายภายในร้าน และในส่วนของการออก รายงานยอดการขาย จะเป็นการสรุปยอดการขายรายวัน และยอดการขายรายเดือน

ซึ่งในการจัดทำໂຄງງານระบบจัดการการสั่งอาหารนี้ได้มีการพัฒนาและออกแบบ โดยใช้ภาษาที่ใช้ในการพัฒนาระบบ Microsoft Visual Basic 2017 และใช้ระบบฐานข้อมูลด้วยโปรแกรม SQL Server 2008

**แหล่งอ้างอิง :**

วิกรม ศรศรี และนำโชค ทองคำมูล. ระบบจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero.  
(ปริญญาตรี คณะบริหารธุรกิจ มหาวิทยาลัยราชพฤกษ์, 2560)

### 2.2.3. ระบบการจัดการร้านอาหาร

**เนื้อหา :**

ปริญญาНИพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อพัฒนาระบบการจัดการร้านอาหาร โดยระบบสามารถจัดการข้อมูลพนักงาน ข้อมูลการขาย ข้อมูลการสั่งซื้อวัตถุดิบ ข้อมูลตรวจสอบวัตถุดิบคงเหลือ และการออกรายงานสำหรับผู้บริหาร เพื่อช่วยสนับสนุนการตัดสินใจในการวางแผนบริหารกิจการให้เกิดผลกำไรสูงสุด รวมถึงการอำนวยความสะดวกต่อผู้ใช้บริการ และการปฏิบัติงานของผู้ให้บริการได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ระบบการจัดการร้านอาหาร มีลักษณะเป็นเว็บแอ�� พลิเคชัน(Web-Based Application) พัฒนาโดยภาษา PHP, HTML และใช้ MySQL client version: 5.0.51a เป็นระบบจัดการฐานข้อมูล

ระบบการจัดการร้านอาหาร คือ ซอฟต์แวร์ที่พัฒนาสำหรับจัดการร้านอาหาร เพื่อช่วยอำนวยความสะดวกในการจัดการร้านอาหาร แบ่งการใช้งานออกเป็นสองส่วน ดังนี้ พิเศษแก่ผู้ใช้ในการรับสั่งอาหาร ตรวจสอบสถานะการทำอาหาร คอมพิวเตอร์ใช้สำหรับจัดเก็บข้อมูลต่าง ๆ และลงโปรแกรมการจัดการร้านอาหาร ใช้ในการบันทึกข้อมูลต่าง ๆ ภายในร้านอาหาร บันทึกข้อมูลการจัดซื้อวัตถุดิบบันทึกข้อมูลวัตถุดิบ บันทึกข้อมูลรายการอาหาร บันทึกข้อมูลรายละเอียดการขาย จัดทำรายงาน บันทึกการเปลี่ยนแปลงข้อมูลต่าง ๆ ซึ่งทำให้มีความสามารถจัดการร้านอาหารได้อย่างมีประสิทธิภาพ มีความถูกต้องแม่นยำ สะดวกรวดเร็ว และตรวจสอบความถูกต้องของข้อมูลย้อนหลังได้

**แหล่งอ้างอิง :**

ธีรพงศ์ ชูชื่น และจิรรัช ศรีโชค. ระบบจัดการร้านอาหาร. (ปริญญาตรี คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสำเภาเจ้าพระยา, 2555)

## 2.2.4. ระบบจัดการร้านกังฟูม่าล่า เปียร์ทิมะ

เนื้อหา :

โครงการนี้มีวัตถุประสงค์เพื่อวิเคราะห์ ออกแบบ และพัฒนาระบบจัดการร้าน กังฟูม่าล่า เปียร์ทิมะ (KungFu Mhala Snow Beer) เพื่อสามารถบริหารจัดการงานที่มีให้เหมาะสมกับการทำงาน และเพื่อช่วยตรวจสอบสินค้าคงเหลือ จัดเก็บข้อมูลให้เป็นระเบียบ ค้นหาได้สะดวกรวดเร็ว ความสะดวกในการทำงานของพนักงาน ลดความผิดพลาดในการคำนวณ และจัดเก็บข้อมูล

ปัจจุบันทางร้านได้ประสบกับปัญหาการเช็คสต็อกจำนวนวัตถุดิบที่ทางร้านได้นำมาจำหน่ายในแต่ละวัน เช่น จดบันทึก จึงทำให้เกิดปัญหา ข้อมูลอาจมีผิดพลาด หรือสูญหาย ลืมจดบันทึก และทำให้การคำนวณยอดขายในแต่ละวันไม่สะดวกและไม่รวดเร็ว และยังเข้มโงยไปถึงการออกใบเสร็จให้แก่ลูกค้า เพราะทางร้านยังรับการสั่งรายการอาหารที่ลูกค้าสั่งเป็นกระดาษ ทำให้การรับอาหารในแต่ละครั้งอาจมีข้อผิดพลาดและไม่สะดวก อีกทั้งทางร้านยังไม่มีระบบให้บริการการจองโต๊ะ ก่อให้เกิดความไม่สะดวกในการติดต่อในการจองโต๊ะและยังเสียเวลาในการมานั่งรอที่หน้าร้านอีกด้วย

ดังนั้น จากปัญหาที่กล่าวข้างต้นจึงได้ทำการวิเคราะห์ หากเรานำเทคโนโลยีที่ถูกพัฒนาขึ้นมาใช้ ทั่วไปจะช่วยให้เข้ากับกิจกรรมการร้านอาหารของเรา ซึ่งจะทำให้การจัดการในธุรกิจร้านอาหารของเราดีขึ้น ซึ่งโปรแกรมนี้ ผู้ประกอบการสามารถนำเข้ามาพิจารณาที่ออกแบบรูปของสถานที่ร้านอาหารมาใช้ในโปรแกรมได้ รวมทั้งสามารถเพิ่ม, ลด, แก้ไข ตำแหน่ง สี ขนาด จำนวนคน ทราบเมนูที่ลูกค้าสั่งและเสิร์ฟได้ถูกต้องทันตามความต้องการ ทั้งยังมีระบบการจัดการรายได้ เช่น สามารถตรวจสอบยอดเงินที่ต้องชำระของลูกค้าแต่ละโต๊ะได้อย่างรวดเร็ว และยังมีตารางข้อมูลรายได้ของยอดเงินในแต่ละวันและรายชื่ออาหาร ซึ่งจะทำให้การรวมยอดขายในแต่ละวันง่ายขึ้น มีความสะดวกสบายมากขึ้นในการรวมยอดขายในแต่ละวัน

แหล่งอ้างอิง :

ปรินทร์ อ่อนโกก, นิรัชชา บุญวิเทียน และ ชัญญาณุช ศรีวรรณ. ระบบจัดการร้านกังฟูม่าล่า เปียร์ทิมะ. (ปริญญาตรี คณะวิทยาการจัดการ มหาวิทยาลัยราชภัฏอุดรธานี, 2563)

สรุปความรู้ 2.1 – 2.4 :

งานวิจัยทั้ง 4 งานเป็นงานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบจัดการร้านอาหารทั้งหมด แม้จะไม่ได้ใช้เทคโนโลยีเดียวกันกับ Delivery Restaurant Manager แต่ก็มีความคล้ายคลึงกันในหลายๆส่วน ทั้งในส่วนของรูปแบบโครงสร้างของระบบ โครงสร้างฐานข้อมูล และการส่งผ่านข้อมูล

เมื่อทำการศึกษารายละเอียดงานวิจัยทั้ง 4 แล้วพบว่า สามารถนำข้อมูล โครงสร้าง และประสบการณ์การทำงานของทั้ง 4 งานวิจัยมาปรับใช้ในระบบ Delivery Restaurant Manager ได้ เพราะมีโมดูลที่เหมือนกันหรือใกล้เคียงกันอยู่มาก เช่น การบันทึกคำสั่งซื้อ การเก็บสถิติการขาย การออกรายงาน ฯลฯ

ข้อมูลเหล่านี้จะเป็นประโยชน์อย่างมากในการเก็บความต้องการของระบบ ในส่วนที่ผู้พัฒนาหรือผู้ใช้งานเองก็คาดไม่ถึง หรือในด้านการออกแบบระบบก็สามารถนำงานวิจัยเหล่านี้มาปรับปรุงแก้ไขได้เป็นอย่างดี

### **2.2.5. เทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกำกับยาแบบอัตโนมัติสำหรับบริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย**

**เนื้อหา :**

จากสถานการณ์การใช้ยาในปัจจุบันนี้ ผู้บริโภคขาดความรู้ความเข้าใจถึงอันตรายที่เกี่ยวกับยา อีกทั้งยังไม่มีแหล่งข้อมูลที่เข้าถึงได้ง่าย ข้อมูลที่ได้รับจากเภสัชกรที่เป็นเพียงฉลากยาจากคลินิกหรือโรงพยาบาลจึงไม่เพียงพอ จากปัญหาดังกล่าวข้างต้นจึงมีแนวคิดในการสร้างระบบแสดงข้อมูลฉลากยาเอกสารกำกับยาแบบอัตโนมัติโดยการส่งข้อมูลดิจิตอลไปเก็บไว้ที่ฐานข้อมูลในเครื่องแม่ข่าย เพื่อเป็นเครื่องมือที่ช่วยในการลดความคลาดเคลื่อนทางยา (Medication Error) ที่เกิดในกระบวนการใช้ยา ตั้งแต่การสั่งใช้ยา การคัดลอกคำสั่งใช้ยา การจ่ายยา และ สุดท้ายเป็นการบริหารยา โดยการนำเทคโนโลยี Optical Character Recognition (OCR) มาทำการแปลงไฟล์ภาพเอกสารที่ได้รับการสแกนให้กลายเป็นไฟล์ข้อความตัวอักษรที่สามารถสืบค้นได้

โดยระบบที่พัฒนาขึ้นนี้ถูกแบ่งออกเป็น 4 ส่วน โดยส่วนแรกคือส่วนของการเก็บข้อมูล ซึ่งข้อมูลยาที่ใช้ในการเก็บมาจากเอกสารกำกับยา โดยใช้เทคโนโลยี OCR (Optical Character Recognition) เป็นการแปลงรูปภาพเป็นตัวอักษร ส่วนที่สองการจัดการข้อมูลยา ซึ่งเป็นส่วนที่ใช้ในการสร้าง แก้ไข ลบ และแสดงข้อมูลยา ทั้งหมดภายในระบบ ส่วนที่สามการค้นหาข้อมูลยา จะเป็นการค้นหาข้อมูลยาได้จากชื่อ สรรพคุณของยารวมถึงอาการของผู้ป่วย ส่วนที่สี่ส่วนการจัดการผู้ใช้จะควบคุมการเข้าถึงระบบในส่วนต่างๆ ของผู้ใช้แต่ละประเภท

ระบบเทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกำกับยาแบบอัตโนมัตินี้มีประสิทธิภาพในการแปลงรูปภาพเป็นตัวอักษรด้วยเทคโนโลยี OCR ที่ความแม่นยาได้ถึงร้อยละ 96.61 และสามารถช่วยลดเวลาของการเก็บข้อมูลลงฐานข้อมูลยา และช่วยให้ผู้บริโภค มีความรู้ความ

ถูกต้องเกี่ยวกับข้อมูล รู้วิธีของการเก็บรักษาของยาเพื่อให้ได้รับประโยชน์ ช่วยให้ปลอดภัยและลดปัญหาของการใช้ยาที่ผิดกับโรค

#### แหล่งอ้างอิง :

วิรุฬห์ ศรีบริรักษ์. เทคโนโลยีเอกสารสนเทศแสดงข้อมูลลักษณะเอกสารกำกับยาแบบอัตโนมัติสำหรับบริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย (คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา, 2562)

#### สรุปความรู้ :

จากเอกสารงานวิจัยข้างต้น ผู้วิจัยได้ทำการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยีที่ใกล้เคียงกันกับระบบจัดการร้านอาหารเดลิเวอรี่ คือ MongoDB, NodeJS, ทฤษฎี REST API และ Angular Framework ซึ่งเป็น Frontend framework อีกตัวหนึ่งที่มีลักษณะการใช้งานใกล้เคียงกันกับ React Framework

ทำให้งานวิจัยนี้มีประโยชน์ในส่วนของการที่ทำให้ได้เห็นถึงตัวอย่างของการนำเทคโนโลยีเหล่านี้ไปใช้งานจริง

# บทที่ 3

## วิธีดำเนินโครงการ

### (Research Methodology)

#### 3.1. ขั้นตอนการดำเนินงานตาม Software Process Model

แบบจำลองน้ำตก (Waterfall Model) เป็นแบบจำลองกระบวนการในการพัฒนาระบบรูปแบบหนึ่ง โดยในอดีตแบบจำลองน้ำตกเป็นที่นิยมใช้กันเป็นอย่างมาก ในระบบหรือซอฟต์แวร์ที่มีขนาดเล็กและไม่ซับซ้อน เนื่องจากสามารถพัฒนาทีละขั้นตอนได้ สามารถตรวจสอบทีละขั้นตอนและการหนดทรัพยากร่างๆ ในแต่ละขั้นตอนได้เป็นอย่างดี ทั้งนี้แบบจำลองน้ำตกมีหลักการคล้ายกับการไหลของน้ำตกที่ไหลจากด้านบนสู่ด้านล่าง ซึ่งในแต่ละขั้นตอนจะไม่สามารถย้อนกลับไปเพื่อแก้ไขหรือปรับปรุงขั้นตอนก่อนหน้านี้ได้ โดยแบบจำลองน้ำตกจึงเหมาะสมกับระบบหรือซอฟต์แวร์ที่มีขอบเขตและข้อกำหนดที่ชัดเจน ทั้งนี้แบบจำลองน้ำตกได้ถูกกล่าวถึงครั้งแรกจากงานวิจัยของ วินเซนต์ ดับเบิลยู รอยซ์ (Winston W. Royce's) เกี่ยวกับเรื่องการจัดการซอฟต์แวร์ที่มีขนาดใหญ่ และมีความ слับซับซ้อน โดยถูกมองว่าการพัฒนาระบบหรือซอฟต์แวร์ที่มีขนาดใหญ่และมีความซับซ้อนเกินกว่าที่จะออกแบบมาด้วยวิธีการที่ไม่เป็นระบบได้ โดยได้มีการนำเสนอรูปแบบหรือวิธีการทำงานแบบใหม่ขึ้นมาที่มีขั้นตอนการทำงานต่าง ๆ เพิ่มมากขึ้น รวมไปถึงในแต่ละช่วงการพัฒนาและออกแบบจะมีการเก็บบันทึกข้อมูลลงในเอกสารเอาไว้ เพื่อช่วยในการพัฒนาออกแบบ และรวมไปถึงการปรับปรุงระบบขึ้นใหม่ ทั้งนี้แบบจำลองดังกล่าวเป็นแบบจำลองที่เหมาะสมสำหรับการพัฒนาระบบและซอฟต์แวร์ที่เน้นด้านคุณภาพมากกว่าต้นทุนหรือระยะเวลาของงานที่มีขอบเขตและข้อกำหนดที่ชัดเจน โดยแบบจำลองน้ำตกจะมีรูปแบบหรือขั้นตอนทั้งหมด 5 ขั้นตอนหลัก ด้วยกัน

- 1.) ขั้นตอนวางแผนและวิเคราะห์ระบบ (Planning and Requirement Definition) เป็นขั้นตอนในการวางแผนการดำเนินงานพัฒนาระบบหรือซอฟต์แวร์ โดยจะทำการรวบรวมข้อมูลความต้องการต่าง ๆ จากผู้ใช้งาน และทำการวิเคราะห์ข้อมูลเพื่อหาข้อสรุปของข้อมูลและจัดทำบันทึกลงในเอกสาร (Requirement

Document) ทั้งในส่วนของความต้องการ แผนผังการทำงานระหว่างข้อมูลและระบบที่จะพัฒนา รวมไปถึงขั้นตอน และกระบวนการทำงานต่าง ๆ

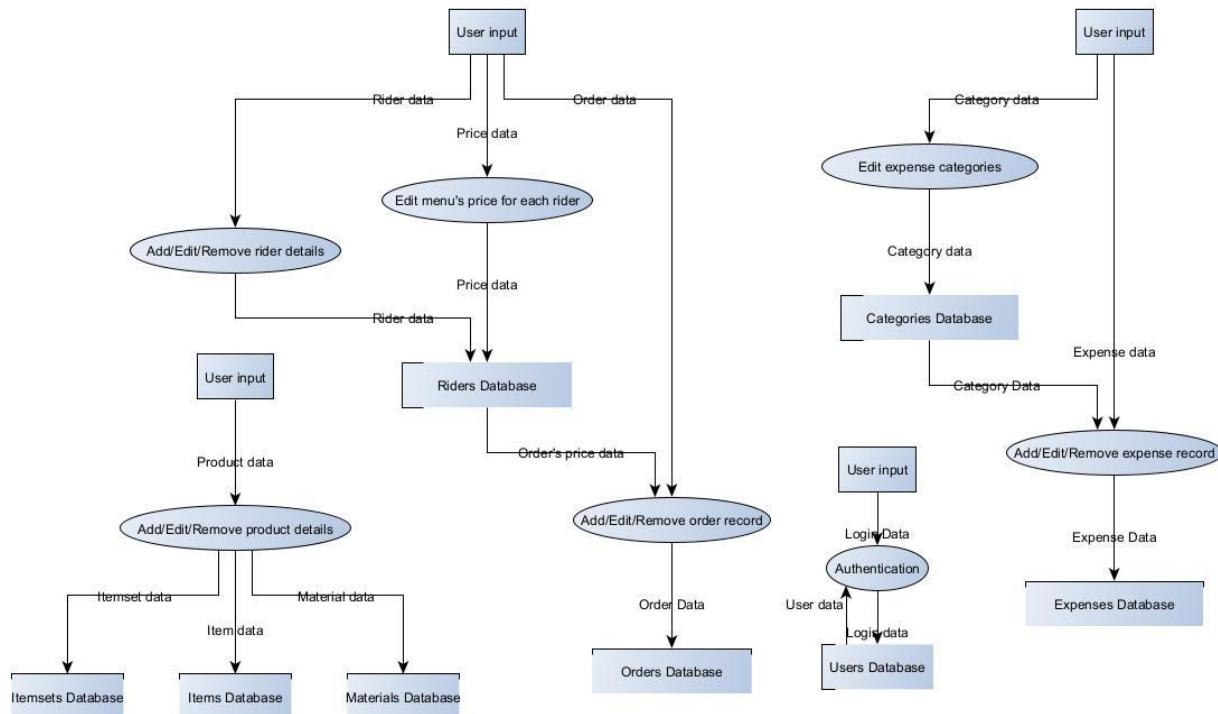
2.) ขั้นตอนการออกแบบระบบหรือซอฟต์แวร์ (System and Software Design) เป็นขั้นตอนในการออกแบบระบบและซอฟต์แวร์ให้เป็นไปตามวัตถุประสงค์หรือความต้องการของผู้ใช้งาน ไม่ว่าจะเป็นทั้งในส่วนของระบบหน้าบ้านและหลังบ้าน (User Interface) กระบวนการทำงานของระบบ (System Workflow) การจัดเก็บข้อมูลบนระบบ (Database System) รายงานเอกสารบนระบบ (Report System) เป็นต้น

3.) ขั้นตอนการพัฒนาและทดสอบระบบในระดับหน่วย (Implementation and Unit Testing) เป็นขั้นตอนที่นำข้อมูลจากขั้นตอนก่อนหน้านี้มาดำเนินการพัฒนาระบบที่สามารถใช้งานได้จริง โดยเป็นการปรับเปลี่ยนข้อมูลจากที่อยู่บนเอกสารให้กลายเป็นระบบหรือซอฟต์แวร์ที่สามารถให้ผู้ใช้งานนำไปใช้งานได้จริง โดยในขั้นตอนนี้มักจะมีการทดสอบระบบในระดับหน่วยเล็ก ๆ (Unit Test) เพื่อเป็นการตรวจสอบว่าในแต่ละส่วนของระบบหรือซอฟต์แวร์ที่พัฒนาขึ้นมาสามารถนำไปใช้ได้จริงและทำงานได้ถูกต้อง

4.) ขั้นตอนการเชื่อมต่อและทดสอบระบบ (Integration and System Testing) เป็นขั้นตอนในการทดสอบการเชื่อมต่อกับระบบอื่น ๆ ที่เกี่ยวข้องว่าในแต่ละส่วนงานที่มีการเชื่อมต่อกันสามารถทำงานร่วมกันได้อย่างถูกต้องหรือไม่ นอกจากนี้ยังรวมไปถึงการทดสอบระบบทั้งหมดก่อนที่จะนำไปใช้งานจริงว่ามีความสมบูรณ์และถูกต้องที่ตรงตามความต้องการของผู้ใช้งานหรือไม่ หากมีการทดสอบในขั้นตอนนี้แล้วพบข้อผิดพลาดหรือปัญหาเกิดขึ้นจะต้องดำเนินการพัฒนา แก้ไข และปรับปรุงระบบหรือซอฟต์แวร์นั้นใหม่ก่อนที่จะนำระบบขึ้นไปใช้งานจริง

5.) ขั้นตอนการนำระบบไปใช้งานจริงและการบำรุงรักษา (Operation and Maintenance System) เป็นขั้นตอนที่เมื่อระบบถูกนำไปใช้งานจริง ซึ่งจะต้องมีการติดตาม ตรวจสอบ เพื่อดำเนินการพัฒนาปรับปรุงและแก้ไขปัญหาหรือข้อผิดพลาดที่อาจเกิดขึ้นได้โดยหากพบปัญหาหรือข้อผิดพลาดตั้งกล่าว จะต้องทำการพัฒนา ปรับปรุงและแก้ไขเพื่อให้ระบบสามารถตอบสนองความต้องการของผู้ใช้งานได้เพิ่มเติม

### 3.2. แผนภาพกระแสข้อมูล



รูปประกอบ 3.1 Dataflow Diagram

2.1 การยืนยันตัวตน : ผู้ใช้ส่งผ่านข้อมูลในการเข้าสู่ระบบ => ระบบดึงข้อมูลผู้ใช้จากฐานข้อมูล  
=> ระบบส่งผลการล็อกอินกลับไปยังผู้ใช้

2.2 การจัดการข้อมูลสินค้า : ผู้ใช้ส่งผ่านข้อมูลสินค้าเข้าสู่ระบบ => ระบบบันทึกข้อมูลสินค้าลงในฐานข้อมูลตามข้อมูลที่ได้รับ

2.3 การจัดการໄรเดอร์ : ผู้ใช้ส่งผ่านข้อมูลໄรเดอร์เข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลໄรเดอร์

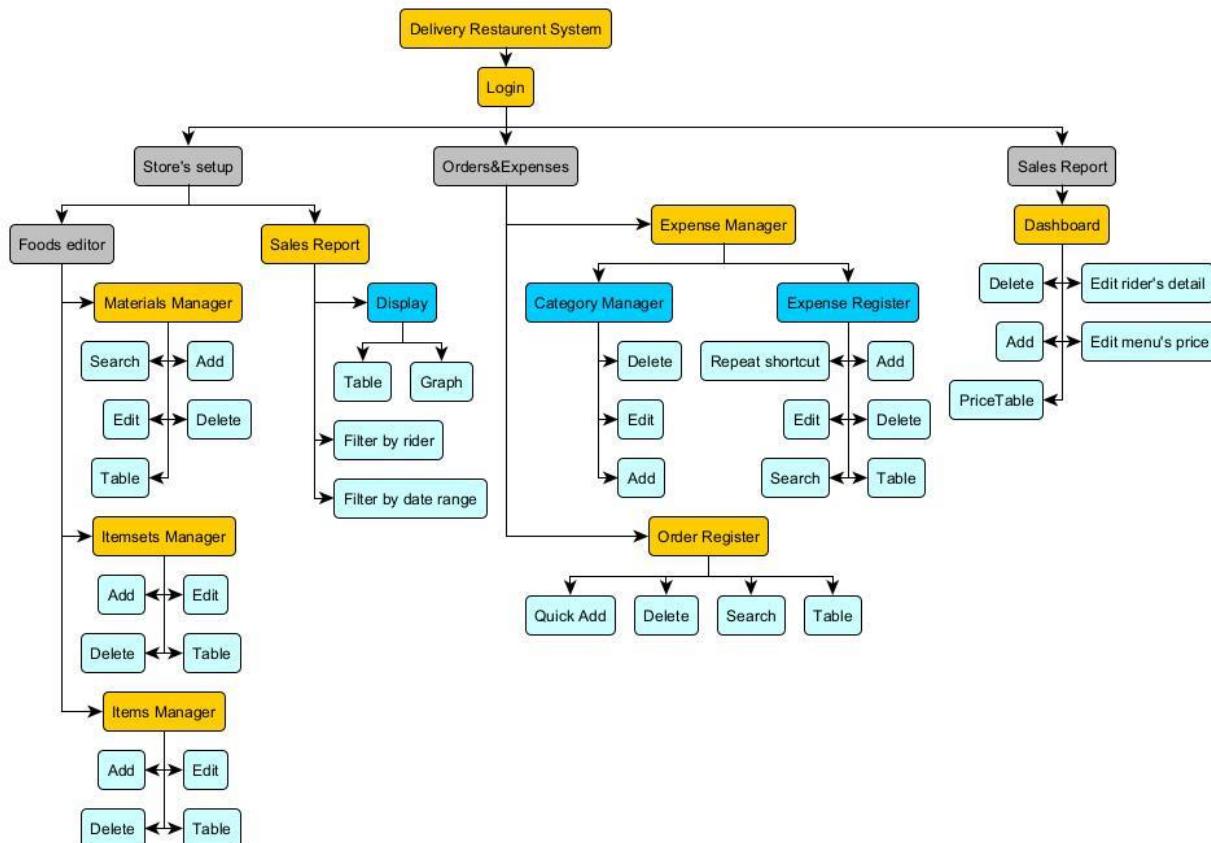
2.4 การจัดการราคาสินค้า : ผู้ใช้ส่งผ่านข้อมูลราคาเข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลໄรเดอร์

2.5 การจัดการหมวดหมู่รายจ่าย : ผู้ใช้ส่งผ่านข้อมูลหมวดหมู่เข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลหมวดหมู่รายจ่าย

2.6 การบันทึกรายจ่าย ; ระบบตีกรายการหมวดหมู่รายจ่ายเพื่อนำไปแสดงผล => ผู้ใช้ส่งผ่านข้อมูลรายจ่ายเข้าสู่ระบบ => ระบบบันทึกข้อมูลลงในฐานข้อมูลรายจ่าย

2.7 การบันทึกคำสั่งชื่อ : ผู้ใช้งานผ่านข้อมูลคำสั่งชื่อเข้าสู่ระบบ => ระบบดึงข้อมูลราคาจากฐานข้อมูลໄໂເດວັນ => ระบบบันทึกข้อมูลเข้าสู่ฐานข้อมูลໂເດວັນ

### 3.3. โครงสร้างซอฟต์แวร์ (Software Structure)



### ຮູບປະກອບ 3.2 Software Structure

0. เมื่อเข้ามาในระบบ ผู้ใช้จะต้องผ่านการตรวจสอบตัวตน เมื่อผ่านการตรวจสอบตัวตนแล้ว จะสามารถเข้าถึงส่วนต่างๆของระบบได้ดังนี้
    1. ส่วนการตั้งค่าร้านค้าครั้งแรก ประกอบด้วย
      - 1.1. ส่วนการจัดอาหาร ประกอบด้วย
        - 1.1.1. หน้าจัดการวัสดุคงคลังในร้านโดยมีโมดูลต่างๆดังนี้
          - 1.1.1.1. ตารางแสดงรายการที่สามารถลบ คืนหากำจัดหรือจัดกลับได้

- 1.1.1.2. ส่วนสำหรับสร้างวัตถุดิบ
- 1.1.1.3. ส่วนสำหรับแก้ไขวัตถุดิบ
- 1.1.2. หน้าจัดการอาหาร ภายใต้ในจะมีโมดูลต่างๆดังนี้
  - 1.1.2.1. ตารางแสดงรายการที่สามารถลบอาหารได้
  - 1.1.2.2. ส่วนสำหรับสร้างอาหาร
  - 1.1.2.3. ส่วนสำหรับแก้ไขอาหาร
- 1.1.3. หน้าจัดการเมนูอาหาร ภายใต้ในจะมีโมดูลต่างๆดังนี้
  - 1.1.3.1. ตารางแสดงรายการที่สามารถลบเมนูอาหารได้
  - 1.1.3.2. ส่วนสำหรับสร้างเมนูอาหาร
  - 1.1.3.3. ส่วนสำหรับแก้ไขเมนูอาหาร
- 1.2. หน้าจัดการໄรเดอร์ ภายใต้ในจะมีโมเดลต่างๆดังนี้
  - 1.2.1. ส่วนสำหรับเลือกและแสดงรายการໄรเดอร์
  - 1.2.2. ตารางแสดงรายการราคาของเมนูอาหาร ที่สามารถค้น จัดเรียง และแก้ไขราคาได้โดยตรง
  - 1.2.3. ส่วนสำหรับการสร้างໄรเดอร์
  - 1.2.4. ส่วนสำหรับลบໄรเดอร์
2. ส่วนสำหรับจัดการบัญชีร้าน ประกอบด้วย
  - 2.1. หน้าจัดการรายจ่าย ภายใต้ในจะมีโมดูลต่างๆดังนี้
    - 2.1.1. หน้าจัดการหมวดหมู่ ซึ่งมีโมดูลสำหรับ สร้าง แก้ไข ลบ และรายการหมวดหมู่อยู่ภายใต้
    - 2.1.2. ส่วนสำหรับการสร้างรายจ่าย
    - 2.1.3. ส่วนสำหรับแก้ไขรายจ่าย
    - 2.1.4. ตารางแสดงรายจ่ายที่สามารถค้นหา และจัดเรียงรายจ่ายได้
  - 2.2. หน้าจัดการคำสั่งชื้อ ภายใต้ในจะมีโมดูลต่างๆดังนี้
    - 2.2.1. ตารางแสดงรายการคำสั่งชื้อที่สามารถค้นหา จัดเรียง และลบคำสั่งชื้อได้
    - 2.2.2. ส่วนสำหรับการสร้างคำสั่งชื้อโดยไม่จำเป็นต้องใช้แป้นพิมพ์
3. ส่วนแสดงสถิติการขาย ประกอบด้วย
  - 3.1.1. หน้าแดชบอร์ด (Dashboard) ภายใต้ในจะมีโมดูลต่างๆดังนี้

### 3.1.1.1. ส่วนแสดงผล

3.1.1.1.1. แสดงผลเป็นตาราง

3.1.1.1.2. แสดงผลเป็นกราฟ

### 3.1.1.2. ส่วนปุ่มคำสั่ง ซึ่งจะประกอบด้วย

3.1.1.2.1. ปุ่มกรองข้อมูลตามໄรเดอร์

3.1.1.2.2. ปุ่มกรองข้อมูลตามระยะเวลา

3.1.1.2.3. ปุ่มเปลี่ยนการแสดงผลระหว่างตารางและกราฟ

## 3.4. ข้อกำหนดซอฟต์แวร์ (Input/Output Specifications)

### 3.4.1. โมดูลการเข้าสู่ระบบ

Input	Process	Output
1. ชื่อผู้ใช้ (Username) 2. รหัสผ่าน (Password)	1. นำชื่อผู้ใช้ เปรียบเทียบกับข้อมูล ผู้ใช้ในฐานข้อมูล 2. นำรหัสผ่านไปเข้ารหัส จากนั้น นำไปเปรียบเทียบกับข้อมูล รหัสผ่านที่เข้ารหัสแล้วใน ฐานข้อมูล	1. ข้อมูลถูกต้อง ผู้ใช้สามารถเข้าสู่ ระบบได้ 2. ข้อมูลไม่ถูกต้อง ปฏิเสธการเข้าสู่ ระบบ และให้ผู้ใช้กรอกข้อมูลใหม่ อีกครั้ง

ตาราง 3.1 Login module

### 3.4.2. โมดูลตรวจสอบการเข้าสู่ระบบ

Input	Process	Output
1. โทเคนการเข้าสู่ระบบ (Login Token) 2. หน้าที่ต้องการเข้า	1. ตรวจสอบความถูกต้องของโทเคน 2. ตรวจสอบวันหมดอายุของโทเคน	1. ข้อมูลถูกต้อง ผู้ใช้สามารถเข้าสู่ หน้าที่ได้ 2. ข้อมูลไม่ถูกต้อง ปฏิเสธการเข้าถึง ส่งผู้ใช้ไปยังหน้าเข้าสู่ระบบ

ตาราง 3.2 Authentication module

### 3.4.3. โมดูลเพิ่มวัตถุดิบ

Input	Process	Output
1. ชื่อ <sup>*</sup> 2. ประเภท <sup>*</sup> 3. หน่วย	1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล	1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ใน ตารางข้อมูล

		2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือน ข้อผิดพลาดที่ทำให้ไม่สำเร็จ
--	--	--

ตาราง 3.3 โมดูลเพิ่มวัตถุคิบ

### 3.4.4. โมดูลแก้ไขวัตถุคิบ

Input	Process	Output
1. วัตถุคิบที่ต้องการแก้ไข 2. ชื่อ 3. ประเภท 4. หน่วย	1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล	1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.4 โมดูลแก้ไขวัตถุคิบ

### 3.4.5. โมดูลบวตถุคิบ

Input	Process	Output
1. วัตถุคิบที่ต้องการลบ	2. ค้นหาข้อมูลที่ต้องการลบ 3. ลบข้อมูลออกจากฐานข้อมูล	1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.5 โมดูลบวตถุคิบ

### 3.4.6. โมดูลเพิ่มอาหาร

Input	Process	Output
1. ชื่อ 2. รายการวัตถุคิบ ประกอบด้วย 2.1. วัตถุคิบที่ใช้ 2.2. จำนวนที่ใช้	1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล	1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.6 โมดูลเพิ่มอาหาร

### 3.4.7. โมดูลแก้ไขอาหาร

Input	Process	Output
1. อาหารที่ต้องการแก้ไข 2. ชื่อ <sup>*</sup> 3. รายการวัตถุคุณภาพ ก่อนด้วย <sup>*</sup> 3.1. วัตถุคุณภาพที่ใช้ 3.2. จำนวนที่ใช้	1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล	1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่ 2. “ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข <sup>*</sup> 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.7 โมดูลแก้ไขอาหาร

### 3.4.8. โมดูลลบอาหาร

Input	Process	Output
1. อาหารที่ต้องการลบ	1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล	1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. “ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.8 โมดูลลบอาหาร

### 3.4.9. โมดูลเพิ่มเมนูอาหาร

Input	Process	Output
1. ชื่อ <sup>*</sup> 2. รายการวัตถุคุณภาพ ก่อนด้วย <sup>*</sup> 2.1. วัตถุคุณภาพที่ใช้ 2.2. จำนวนที่ใช้ 3. ค่าแรง 4. ค่าน้ำมัน	1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล	1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.9 โมดูลเพิ่มเมนูอาหาร

### 3.4.10. โมดูลแก้ไขเมนูอาหาร

Input	Process	Output

<ol style="list-style-type: none"> <li>1. เมนูอาหารที่ต้องการแก้ไข</li> <li>2. ชื่อ</li> <li>3. รายการตั้งคุณภาพ ก่อนด้วย           <ol style="list-style-type: none"> <li>3.1. วัตถุติดที่ใช้</li> <li>3.2. จำนวนที่ใช้</li> </ol> </li> <li>4. ค่าแรง</li> <li>5. ค่าน้ำมัน</li> </ol>	<ol style="list-style-type: none"> <li>1. ตรวจสอบ Input</li> <li>2. ค้นหาข้อมูลที่ต้องการแก้ไข</li> <li>3. แก้ไขข้อมูลในฐานข้อมูล</li> </ol>	<ol style="list-style-type: none"> <li>1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางด้วยข้อมูลใหม่</li> <li>2. “ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข”</li> <li>3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ</li> </ol>
--	--	--

ตาราง 3.10 โมดูลแก้ไขเมนูอาหาร

### 3.4.11. โมดูลลบเมนูอาหาร

Input	Process	Output
<ol style="list-style-type: none"> <li>1. เมนูอาหารที่ต้องการลบ</li> </ol>	<ol style="list-style-type: none"> <li>1. ค้นหาข้อมูลที่ต้องการลบ</li> <li>2. ลบข้อมูลออกจากฐานข้อมูล</li> </ol>	<ol style="list-style-type: none"> <li>1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล</li> <li>2. “ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ</li> <li>3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ</li> </ol>

ตาราง 3.11 โมดูลลบเมนูอาหาร

### 3.4.12. โมดูลเพิ่มไรมเดอร์

Input	Process	Output
<ol style="list-style-type: none"> <li>1. ชื่อ</li> <li>2. ค่าธรรมเนียม</li> <li>3. ค่าภาษี</li> <li>4. ค่าน้ำมัน</li> </ol>	<ol style="list-style-type: none"> <li>1. ตรวจสอบ Input</li> <li>2. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล</li> </ol>	<ol style="list-style-type: none"> <li>1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล</li> <li>2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ</li> </ol>

ตาราง 3.12 โมดูลเพิ่มไรมเดอร์

### 3.4.13. โมดูลลบไรมเดอร์

Input	Process	Output
<ol style="list-style-type: none"> <li>1. ไรมเดอร์ที่ต้องการลบ</li> </ol>	<ol style="list-style-type: none"> <li>1. ค้นหาข้อมูลที่ต้องการลบ</li> <li>2. ลบข้อมูลออกจากฐานข้อมูล</li> </ol>	<ol style="list-style-type: none"> <li>1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล</li> <li>2. “ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ</li> </ol>

		3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือน ข้อผิดพลาดที่ทำให้ไม่สำเร็จ
--	--	---

ตาราง 3.13 โมดูลบีรีเอดอร์

### 3.4.14. โมดูลแก้ไขราคาเมนูอาหาร

Input	Process	Output
1. ไรเดอร์ที่ต้องการแก้ไข 2. เมนูที่ต้องการแก้ไข 3. ราคา	1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล	1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่ แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดง แจ้งเตือนว่าไม่พบข้อมูลที่ต้องการ แก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้ง เตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.14 โมดูลแก้ไขราคาเมนูอาหาร

### 3.4.15. โมดูลเพิ่มหมวดหมู่รายจ่าย

Input	Process	Output
1. ชื่อ 2. (ถ้าต้องการ) หมวดหมู่หลัก	1. ตรวจสอบ Input 2. (ถ้ามี) ค้นหาหมวดหมู่ที่อ้างถึง 3. เพิ่มข้อมูลเข้าสู่ฐานข้อมูล	1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ใน ตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือน ข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.15 โมดูลเพิ่มหมวดหมู่รายจ่าย

### 3.4.16. โมดูลแก้ไขหมวดหมู่รายจ่าย

Input	Process	Output
1. หมวดหมู่ที่ต้องการแก้ไข 2. ชื่อ 3. (ถ้าต้องการ) หมวดหมู่หลัก	1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล	1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่ แก้ไขในตารางด้วยข้อมูลใหม่ 2. ไม่พบข้อมูลที่ต้องการแก้ไข แสดง แจ้งเตือนว่าไม่พบข้อมูลที่ต้องการ แก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้ง เตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.16 โมดูลแก้ไขหมวดหมู่รายจ่าย

### 3.4.17. โมดูลบหมวดหมู่รายจ่าย

Input	Process	Output
1. หมวดหมู่ที่ต้องการลบ	1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล	1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล 2. “ไม่พบข้อมูลที่ต้องการลบ แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการลบ 3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.17 โมดูลบหมวดหมู่รายจ่าย

### 3.4.18. โมดูลเพิ่มรายจ่าย

Input	Process	Output
1. หมวดหมู่ 2. คำอธิบายรายการ 3. จำนวนที่ซื้อ 4. ราคากล่อง 5. วันเวลาที่เกิด	1. ตรวจสอบ Input 2. เพิ่มข้อมูลเข้าฐานข้อมูล	1. เพิ่มข้อมูลสำเร็จ แสดงข้อมูลใหม่ในตารางข้อมูล 2. เพิ่มข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.18 โมดูลเพิ่มรายจ่าย

### 3.4.19. โมดูลแก้ไขรายจ่าย

Input	Process	Output
1. รายจ่ายที่ต้องการแก้ไข 2. หมวดหมู่ 3. คำอธิบายรายการ 4. จำนวนที่ซื้อ 5. ราคากล่อง 6. วันเวลาที่เกิด	1. ตรวจสอบ Input 2. ค้นหาข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลในฐานข้อมูล	1. แก้ไขข้อมูลสำเร็จ แสดงข้อมูลที่แก้ไขในตารางตัวอย่างใหม่ 2. “ไม่พบข้อมูลที่ต้องการแก้ไข แสดงแจ้งเตือนว่าไม่พบข้อมูลที่ต้องการแก้ไข 3. แก้ไขข้อมูลไม่สำเร็จ แสดงแจ้งเตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ

ตาราง 3.19 โมดูลแก้ไขรายจ่าย

### 3.4.20. โมดูลลบรายจ่าย

Input	Process	Output
1. รายจ่ายที่ต้องการลบ	1. ค้นหาข้อมูลที่ต้องการลบ 2. ลบข้อมูลออกจากฐานข้อมูล	1. ลบข้อมูลสำเร็จ นำข้อมูลออกจากตารางข้อมูล

		<ol style="list-style-type: none"> <li>2. ไม่เพบข้อมูลที่ต้องการลบ และแสดงแจ้ง เตือนว่าไม่เพบข้อมูลที่ต้องการลบ</li> <li>3. ลบข้อมูลไม่สำเร็จ และแสดงแจ้งเตือน ข้อผิดพลาดที่ทำให้ไม่สำเร็จ</li> </ol>
--	--	---

ตาราง 3.20 โมดูลรายจ่าย

### 3.4.21. โมดูลทำข้าวยจ่าย

Input	Process	Output
1. รายจ่ายที่ต้องการทำข้าว 2. รายการอาหาร	<ol style="list-style-type: none"> <li>1. ค้นหาข้อมูลที่ต้องการทำข้าว</li> <li>2. เพิ่มข้อมูลในฐานข้อมูลโดยใช้ข้อมูล ของรายจ่ายที่เลือก วันเวลาที่เกิด ใช้เป็นเวลาปัจจุบัน</li> </ol>	<ol style="list-style-type: none"> <li>1. ทำข้าวข้อมูลสำเร็จ และแสดงข้อมูลใหม่ ในตารางด้วยข้อมูล</li> <li>2. ไม่เพบข้อมูลที่ต้องการทำข้าว และแสดง แจ้งเตือนว่าไม่เพบข้อมูลที่ต้องการทำ ข้าว</li> <li>3. ทำข้าวข้อมูลไม่สำเร็จ และแสดงแจ้ง เตือนข้อผิดพลาดที่ทำให้ไม่สำเร็จ</li> </ol>

ตาราง 3.21 โมดูลทำข้าวยจ่าย

### 3.4.22. โมดูลเพิ่มคำสั่งชื้อ

Input	Process	Output
<ol style="list-style-type: none"> <li>1. ไรเดอร์</li> <li>2. วันเวลาที่เกิด</li> <li>3. รายการอาหาร           <ol style="list-style-type: none"> <li>3.1. เมนูอาหาร</li> <li>3.2. จำนวนอาหาร</li> </ol> </li> <li>4. วันเวลาที่เกิด</li> </ol>	<ol style="list-style-type: none"> <li>1. ตรวจสอบ Input</li> <li>2. ค้นหาข้อมูลราคาจากฐานข้อมูลไร เดอร์</li> <li>3. เพิ่มข้อมูลเข้าสู่ฐานข้อมูลพร้อม ราคาของแต่ละเมนูและยอดรวม ทั้งหมด</li> </ol>	<ol style="list-style-type: none"> <li>1. เพิ่มข้อมูลสำเร็จ และแสดงข้อมูลใหม่ใน ตารางข้อมูล</li> <li>2. เพิ่มข้อมูลไม่สำเร็จ และแสดงแจ้งเตือน ข้อผิดพลาดที่ทำให้ไม่สำเร็จ</li> </ol>

ตาราง 3.22 โมดูลเพิ่มคำสั่งชื้อ

### 3.4.23. โมดูลลบคำสั่งชื้อ

Input	Process	Output
1. คำสั่งชื้อที่ต้องการลบ	<ol style="list-style-type: none"> <li>1. ค้นหาข้อมูลที่ต้องการลบ</li> <li>2. ลบข้อมูลออกจากฐานข้อมูล</li> </ol>	<ol style="list-style-type: none"> <li>1. ลบข้อมูลสำเร็จ นำข้อมูลออกจาก ตารางข้อมูล</li> <li>2. ไม่เพบข้อมูลที่ต้องการลบ และแสดงแจ้ง เตือนว่าไม่เพบข้อมูลที่ต้องการลบ</li> </ol>

		3. ลบข้อมูลไม่สำเร็จ แสดงแจ้งเตือน ข้อผิดพลาดที่ทำให้ไม่สำเร็จ
--	--	---

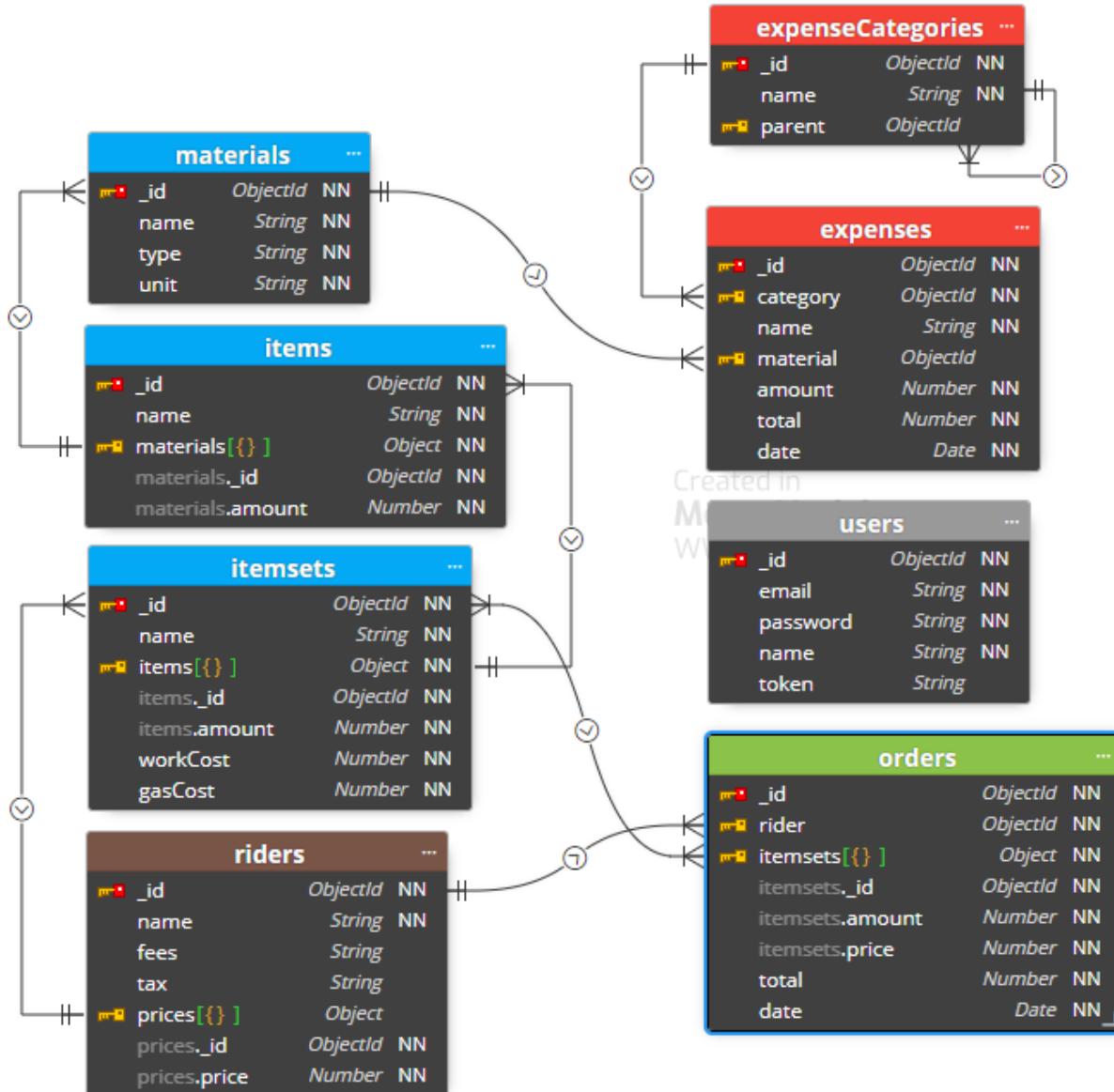
ตาราง 3.23 โมดูลคำสั่งซื้อ

### 3.4.24. โมดูลแสดงสถิติการขาย

Input	Process	Output
1. ໄറเดอร์ 2. ช่วงเวลา 3. ประเภทการแสดงผล	1. ตีงข้อมูลจากฐานข้อมูลօเดอร์ 2. คัดเอาข้อมูลที่อยู่ในช่วงเวลา 3. คัดเอาข้อมูลจากໄറเดอร์ที่เลือก 4. แปลงข้อมูลตามประเภทการ แสดงผลที่เลือก	1. แสดงสถิติการขายตามการแสดงผล ที่เลือก

ตาราง 3.24 โมดูลแสดงสถิติการขาย

### 3.5. E-R Diagram



รูปประกอบ 3.3 Entity Relationship Diagram

### 3.6. ตารางฐานข้อมูล (Tables)

#### 3.6.1. ตารางผู้ใช้ (users)

ลำดับ	ชื่อฟิลด์	ชนิดข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectId	สร้างอัตโนมัตโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	email (unique)	String	Email ของผู้ใช้	nameofemail@domain.notebook
3	password	String	Password ของผู้ใช้ที่ผ่านการเข้ารหัสแล้ว	\$2y\$10\$/uO.IASKMRL.RQSOrgWZ5.Lxu6elu1WET0.w9zWznllLwPfQgfwUEW
4	name	String	ชื่อของผู้ใช้	Nick
5	token	String	Login token session ปัจจุบันที่ generate โดย Json Web Token	eyJhbGciOiJIUzI1NilsInR5cCI6IkpxVCJ9.eyJpZCI6IjYyZDhkZTg5NWE1NmZkZjVkJNDMyMWQ0MilslmlhdCI6MTY2NjU0MTQ3NiwiZXhwIjoxNjY5MTMzNDc2fQ.fM_B_w2-dgSKcq9HvW78nN1YT-0vsRWHhQ2hAclZd_4

ตาราง 3.25 ฐานข้อมูลผู้ใช้

#### 3.6.2. ตารางวัตถุคิบ (materials)

ลำดับ	ชื่อฟิลด์	ชนิดข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectId	สร้างอัตโนมัตโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	name (unique)	String	ชื่อวัตถุคิบ	เกลือ
3	type	String	ประเภทวัตถุคิบ	material
4	unit	String	หน่วยของวัตถุคิบ	กรัม

ตาราง 3.26 ฐานข้อมูลวัตถุคิบ

### 3.6.3. ตารางอาหาร (items)

ลำดับ	ชื่อฟิลด์	ชนิด ข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectId	สร้างอัตโนมัติโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	name (unique)	String	ชื่ออาหาร	น้ำจิ่นเขี๊ยด
3	materials	Array of objects	รายการวัตถุดิบที่ใช้ภายในอาหาร	[ { _id: value, amount: value }, { _id: value, amount: value }, { _id: value, amount: value }, { _id: value, amount: value }, ]
3.1	materials._id (FK: materials)	ObjectId	ObjectId ของวัตถุดิบที่ใช้	ObjectId("62861e401904f7babfb13d")
3.2	materials.amount	Number	จำนวนวัตถุดิบที่ใช้ต่ออาหาร 1 ชุด	100

ตาราง 3.27 ฐานข้อมูลอาหาร

### 3.6.4. ตารางเมนูอาหาร (itemsets)

ลำดับ	ชื่อฟิลด์	ชนิด ข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectId	สร้างอัตโนมัติโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	name (unique)	String	ชื่อเมนูอาหาร	ชุดกุ้งดองซีอิ๊ว
3	items	Object	รายการอาหารที่ใช้ภายในเมนู	[ { _id: value, amount: value }, { _id: value, amount: value }, { _id: value, amount: value }, { _id: value, amount: value }, ]

3.1	items._id (FK: items)	ObjectID	ObjectId ของ วัตถุดิบที่ใช้	ObjectId("62861e401904f7babfb13d")
3.2	items.amount	Number	จำนวนอาหารที่ใช้ต่อ เมนูอาหาร 1 เมนู	1
4	workCost	Number	ค่าแรงที่ใช้ในการทำ เมนูนี้	35
5	gasCost	Number	ค่าน้ำมันที่ใช้ในการทำ เมนูนี้	7

ตาราง 3.28 ฐานข้อมูลเมนูอาหาร

### 3.6.5.

#### ตารางแพลตฟอร์มการขาย (riders)

ลำดับ	ชื่อฟิลด์	ชนิด ข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectID	สร้างอัตโนมัติโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	name (unique)	String	ชื่อแพลตฟอร์มที่ใช้	Grab
3	fees	Number	ค่าธรรมเนียมเป็น เบอร์เซ็นต์	30
4	tax	Number	ภาษีค่าธรรมเนียมเป็น เบอร์เซ็นต์	7
5	gasCost	Number	ต้นทุนค่าน้ำมันของทุก เมนูในแพลตฟอร์มนี้ ในการนี้ที่ไม่ต้องการใช้ ต้นทุนค่าน้ำมันของแต่ ละเมนู	1
6	prices	Array of objects	รายการอาหารที่ใช้ ภายในเมนู	[ { _id: value, price: value }, { _id: value, price: value }, { _id: value, price: value }, { _id: value, price: value }, ]

6.1	prices._id (FK: itemsets)	ObjectID	ObjectId ของ เมนูอาหารที่จะกำหนด ราคา	ObjectId("62861e401904f7babfb13d")
6.2	prices.price	Number	ราคาของเมนูข้างต้นใน แพลตฟอร์มนี้	259

ตาราง 3.29 ฐานข้อมูลแพลตฟอร์มการขาย

### 3.6.6. ตารางหมวดหมู่รายจ่าย (expenseCategories)

ลำดับ	ชื่อฟิลด์	ชนิด ข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectID	สร้างอัตโนมัติโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	name (unique)	String	ชื่อหมวดหมู่	Business
3	Parent (FK: expenseCategories)	ObjectID	ObjectId ของ หมวดหมู่อื่น ใน กรณีที่ต้องการ กำหนดให้หมวดหมู่ นี้เป็นหมวดหมู่รอง ของหมวดหมู่นั้นๆ	ObjectId("62861e401904f7babfb13d")

ตาราง 3.30 ฐานข้อมูลหมวดหมู่รายจ่าย

### 3.6.7. ตารางรายจ่าย (expenses)

ลำดับ	ชื่อฟิลด์	ชนิด ข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectID	สร้างอัตโนมัติโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	name	String	คำอธิบายรายการ	Business
3	category	ObjectID	ObjectId หมวดหมู่ ของรายจ่ายนี้	ObjectId("62861e401904f7babfb13d")
4	Material	ObjectID	ObjectId วัตถุดิบ ใน กรณีที่เลือกหมวดหมู่ Ingredient หรือ Packaging	ObjectId("62861e401904f7babfb13d")

5	amount	Number	จำนวนที่ซื้อ	100
6	total	Number	ค่าใช้จ่ายทั้งหมด	354
7	date	Date	วันเวลาที่เกิดรายจ่าย	ISODATE("06/16/2022")

ตาราง 3.31 ฐานข้อมูลรายจ่าย

### 3.6.8. ตารางคำสั่งซื้อ (orders)

ลำดับ	ชื่อพิล็อก	ชนิดข้อมูล	คำอธิบาย	ข้อมูลตัวอย่าง
1	_id (PK)	ObjectId	สร้างอัตโนมัติโดย MongoDB	ObjectId("62861e401904f7babfb13d")
2	rider	ObjectId	ObjectId แพลตฟอร์มที่คำสั่งซื้อนี้เกิดขึ้น	ObjectId("62861e401904f7babfb13d")
3	itemsets	Array of objects	รายการเมนูอาหารภายในคำสั่งซื้อ	[ { _id: value, amount: value, price: value }, { _id: value, amount: value, price: value } ]
3.1	itemset._id	ObjectId	ObjectId ของ เมนูอาหาร	ObjectId("62861e401904f7babfb13d")
3.2	itemset.amount	Number	จำนวนของ เมนูอาหารข้างต้น	2
3.3	itemset.price	Number	ราคากล่องจากไรเดอร์ที่เลือกของ เมนูอาหารข้างต้น	159

4	total	Number	ยอดรวมทั้งหมดของคำสั่งซื้อนี้ที่ระบบคำนวณออกมา	318
5	date	Date	วันเวลาที่เกิดคำสั่งซื้อ	ISODATE("06/16/2022")

ตาราง 3.32 ฐานข้อมูลคำสั่งซื้อ

### 3.7. ส่วนต่อประสาน (User Interface / Screen Layout)

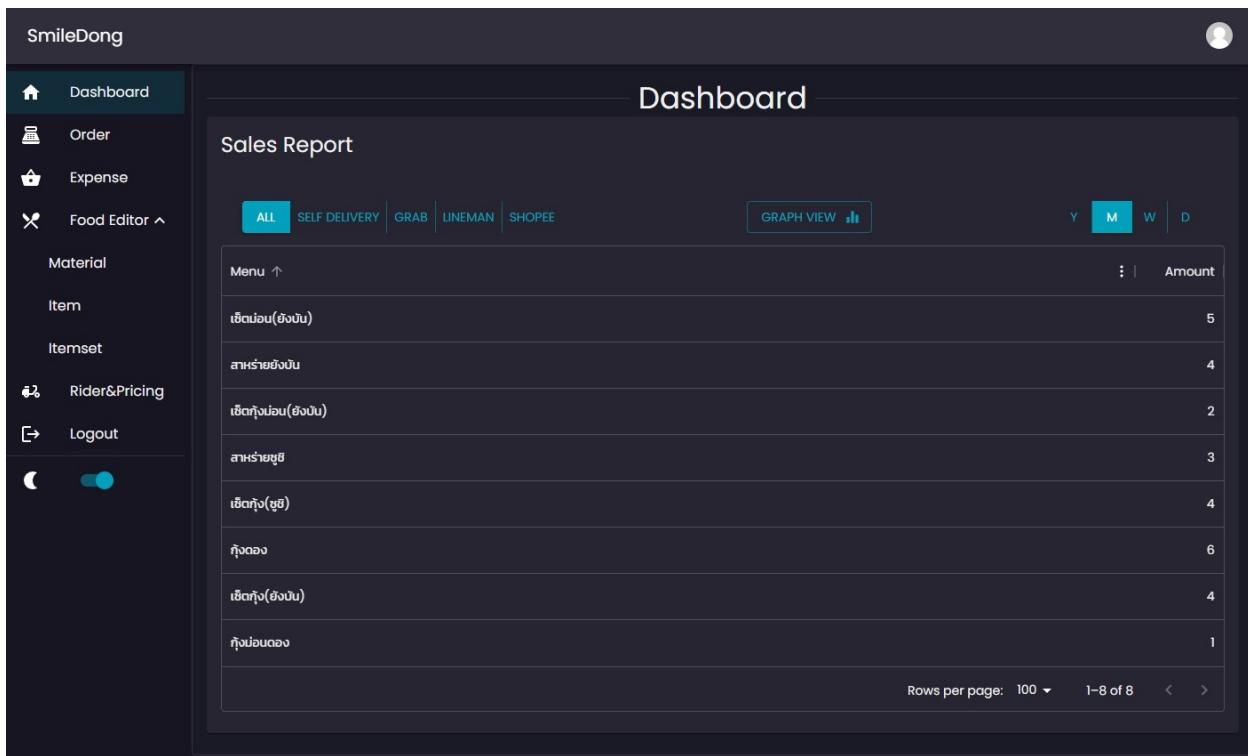
#### 3.7.1. เข้าสู่ระบบ

The image shows a dark-themed login interface for a system named "SmileDong". At the top, the brand name "SmileDong" is displayed. Below it, a large button features a right-pointing arrow icon followed by the word "Login". Underneath this button, the text "Login and start the system" is centered. There are two input fields: the first is labeled "Enter your email" and the second is labeled "Enter password". At the bottom of the form is a dark rectangular button labeled "Submit".

รูปประกอบ 3.4 Login UI

### 3.7.2. Dashboard – สิทธิการขาย

#### 3.7.2.1. การแสดงแบบตาราง



The screenshot shows the SmileDong application interface. On the left is a dark sidebar with navigation links: Dashboard (selected), Order, Expense, Food Editor, Material, Item, Itemset, Rider&Pricing, and Logout. Below the sidebar is a toggle switch. The main area is titled "Dashboard" and contains a "Sales Report" section. At the top of the report are buttons for "ALL", "SELF DELIVERY", "GRAB", "LINEMAN", and "SHOPEE". There is also a "GRAPH VIEW" button and a date selector showing "M". The report table has columns for "Menu ↑", "Amount", and a sorting icon. The data in the table is as follows:

Menu ↑	Amount
ເສັ້ນບົບ(ຫົງບັນ)	5
ສາກົ່າຍອືບບັນ	4
ເມືດຖຸນເມືນ(ຫົງບັນ)	2
ສາກົ່າຍຫຼື	3
ເມືດຖຸ(ຫຼື)	4
ກັງດອງ	6
ເມືດຖຸ(ຫົງປັນ)	4
ກັງເນັບຕອນ	1

At the bottom of the report, there are buttons for "Rows per page: 100" and "1-8 of 8".

ຮູບປະກອບ 3.5 ປີ ຕາງຮາງສົດໃກ່ການຂາຍ

### 3.7.2.2. การแสดงแบบกราฟ



รูปประกอบ 3.6 บ) กราฟสัดส่วนขาย

### 3.7.3. Order – บันทึกคำสั่งซื้อ ส่วนเพิ่มคำสั่งซื้อ, ตารางและการลบคำสั่งซื้อ

The screenshot shows the SmileDong Order Manager interface. On the left is a sidebar with navigation links: Dashboard, Order (selected), Expense, Food Editor (with Material, Item, Itemset), Rider&Pricing, and Logout. Below the sidebar are theme switcher and language selection buttons.

**Editor Section:** Contains tabs for SELF DELIVERY, GRAB, LINEMAN, and SHOPEE. Under each tab are several buttons for selecting delivery methods or items. For example, under SELF DELIVERY, there are buttons for 'กุ้งดอง', 'น้ำมันดอง', 'กุ้งเนื้อดอง', 'ເຊື້ອກຸງ(ຫຼູບ)', and 'ເຊື້ອມອນ(ຫຼູບ)'.

**Details Section:** Displays a message 'Please select rider' and a 'Totals:' section with a date/time selector and a note 'Leave empty to set to current time'. It also includes a trash can icon.

**Table Section:** A data grid showing a list of orders. The columns are Rider, Details, Items, Total, Date, and Del (Delete). The data includes:

Rider	Details	Items	Total	Date	Del
Grab	ເຊື້ອກຸງ(ຫຼູບ)x1, ກຸງນໍ້າດອນx1, ເຊື້ອກຸງ(ຫັນບັນ)x1	3	514	21/10/2022, 11:06	
Self Deliver...	ກຸງນໍ້າx1, ເຊື້ອມອນ(ຫັນບັນ)x1	2	259	21/10/2022, 11:00	
Self Deliver...	ກຸງດອນx1	1	90	21/10/2022, 05:41	
Self Deliver...	ເຊື້ອມອນ(ຫັນບັນ)x4, ສາທ່ວ່າຍອັງນັບນຳx3	2	751	20/10/2022, 00:00	
Lineman	ເຊື້ອກຸງນໍ້າ(ຫັນບັນ)x1, ສາທ່ວ່າຍອັງນັບນຳx1	2	225	20/10/2022, 00:00	
Lineman	ເຊື້ອກຸງ(ຫຼູບ)x1	1	165	20/10/2022, 00:00	
Grab	ເຊື້ອກຸງນໍ້າ(ຫັນບັນ)x1, ສາທ່ວ່າຍອັງນັບນຳx1	2	225	20/10/2022, 00:00	
Grab	ກຸງດອນx1, ສາທ່ວ່າຍອັງນັບນຳx1	2	164	20/10/2022, 00:00	
Grab	ກຸງດອນx1, ສາທ່ວ່າຍອັງນັບນຳx1	2	149	20/10/2022, 00:00	
Grab	ເຊື້ອກຸງ(ຫຼູບ)x1	1	165	20/10/2022, 00:00	
Grab	ເຊື້ອກຸງ(ຫຼູບ)x1	1	165	20/10/2022, 00:00	
Grab	ເຊື້ອກຸງ(ຫັນບັນ)x1	1	180	20/10/2022, 00:00	
Grab	ເຊື້ອກຸງ(ຫັນບັນ)x1	1	180	20/10/2022, 00:00	
Shopee	ກຸງດອນx2	1	270	20/10/2022, 00:00	

At the bottom of the table, there are buttons for 'Rows per page: 25 ▾', '1–15 of 15', and navigation arrows.

ຮັບປະກອບ 3.7 ໃນ ການຈັດການຄໍາສັ່ງຊື່

### 3.7.4. Expense - บันทึกรายจ่าย

#### 3.7.4.1. ตารางรายจ่าย การลบ และการทำซ้ำ

Expense Manager

Expense records

Category	Name	Amount	Total	Date	Actions
Ingredient	พริกแดง	500 กก	45 ₵	21/10/2022, 10:55	
Ingredient	กุ้ง	100 กก	340 ₵	21/10/2022, 10:54	
Packaging	กล่องหีบห่ำ	50 กล่อง	90 ₵	10/10/2022, 10:03	
Ingredient	พริกแดง	500 กก	45 ₵	09/10/2022, 22:57	
Ingredient	มะนาวสุก	100 ถุง	75 ₵	08/10/2022, 00:00	
Ingredient	น้ำตาลกราน	1000 กก	43 ₵	08/10/2022, 00:00	
Packaging	กล่องหีบห่ำ	50 กล่อง	90 ₵	08/10/2022, 00:00	
Equipment	เครื่องขัด	3	1500 ₵	05/10/2022, 00:00	
Equipment	เครื่องปั๊บ	1	300 ₵	05/10/2022, 00:00	
Equipment	เครื่องปั๊นไฟฟ้า	1	359 ₵	20/09/2022, 23:21	
Equipment	ถ่านชีม	50	49 ₵	02/09/2022, 23:21	
Equipment	กระดาษเชิญและของขวัญ	500	38 ₵	31/08/2022, 00:00	
Equipment	น้ำยาล้างจาน	1	14 ₵	20/08/2022, 23:25	
Ingredient	ข้าวผัดปู	1000 กก	110 ₵	01/01/1970, 07:00	

Rows per page: 100 < > 1-14 of 14

รูปประกอบ 3.8 บ) ตารางบันทึกรายจ่าย

### 3.7.4.2. ส่วนสร้างรายจ่าย

The screenshot shows the SmileDong application's expense management section. On the left is a sidebar with icons for Dashboard, Order, Expense, Food Editor (expanded to show Material, Item, Itemset), Rider&Pricing, and Logout. The main area is titled 'Expense Manager' and contains a 'Create new expense' form. The form includes fields for Category (dropdown menu), Expense description/name (text input), Amount (text input), Date&Time (date/time picker), Total (text input), and a note about the total cost for expenses. At the bottom are two buttons: 'SAVE & ADD NEW EXPENSE' and 'SAVE & GO BACK'.

รูปประกอบ 3.9 บ) การสร้างรายจ่าย

### 3.7.4.3. ส่วนแก้ไขรายจ่าย

The screenshot shows the SmileDong application's expense management section, specifically the 'Editing' view for an expense with ID 6352181b5d9cd22d258c7ee8. The sidebar and main interface are identical to the previous screenshot, showing the 'Expense Manager' screen with a form for editing an existing expense. The form includes fields for Category (dropdown menu, currently set to 'Ingredient'), Select Ingredient (dropdown menu, currently set to 'พริกไทย'), Amount (text input, currently 500), Date&Time (date/time picker, currently 10/21/2022 10:55), Total (text input, currently 45), and a note about the total cost for expenses. At the bottom is a single 'SAVE & GO BACK' button.

รูปประกอบ 3.10 บ) การแก้ไขรายจ่าย

### 3.7.5. Category - การสร้างและลบหมวดหมู่รายจ่าย

The screenshot shows the SmileDong application's interface. On the left is a dark sidebar with various icons and labels: Dashboard, Order, Expense, Food Editor (with a dropdown arrow), Material, Item, Itemset, Rider&Pricing, Logout, and a toggle switch. The main area has a header "Category Manager" and a "Create new category" section with "Name" and "Parent" fields. A blue "BACK" button is in the top right. The main content lists categories with edit and delete icons:

Name	Parent	Action
Ingredient	No parent	
Packaging	No parent	
Equipment	No parent	
ମୂଲ୍ୟାଙ୍କଣ	No parent	
ମୂଲ୍ୟାଙ୍କଣ 2	No parent	
Business	No parent	
Discount	No parent	
Monthly-fees	No parent	
Error	No parent	

### รูปประกอบ 3.11 บ การสร้างและลบหมวดหมู่รายจ่าย

### 3.7.6. Material – สร้าง ลบ และตารางข้อมูลวัตถุดิบ

Material Manager						
	Name	Type	Unit			
		Ingredient				
Material	ແກ້ໄສ 16 oz.	package	ໃບ	21/10/2022, 10:10		
Item	ພວກເມືອງ	ingredient	ກັບ	21/10/2022, 10:09		
Itemset	ກລົ່ຈຳສອງຫຸນ	package	ກລົ່ຈຳ	12/09/2022, 01:51		
Rider&Pricing	ກລົ່ຈຳຫຸນ	package	ກລົ່ຈຳ	12/09/2022, 01:51		
Logout	ຕະຫຼາຍ	package	ຖຸ	12/09/2022, 01:51		
	ຖຸຈົດ	package	ຖຸ	20/08/2022, 18:28		
	ກົມຫົວ	package	ກົມ	20/08/2022, 18:27		
	ຖຸນຫາກຮ້າຍ	package	ຖຸ	20/08/2022, 18:27		
	ຖຸພຣິກກະເຄີຍ	package	ຖຸ	20/08/2022, 18:27		
	ຖຸນປໍ່ເຈັນ	package	ຖຸ	20/08/2022, 18:27		
	ຄົກົມກົມ	ingredient	ກັບ	20/08/2022, 18:26		
	ຈຳດ້າ	ingredient	ກັບ	20/08/2022, 18:26		
	ກາຫາວ	ingredient	ກັບ	20/08/2022, 18:26		
	ເທົ່ອ	ingredient	ກັບ	20/08/2022, 18:26		
	ມະນາວູກູກ	ingredient	ຊາ	20/08/2022, 18:25		
	ບ້າດຄາມະພຮ້າວ	ingredient	ກັບ	20/08/2022, 18:25		
	ໂສຢູ່າງູກູກ	ingredient	ມີມໍລັດ	20/08/2022, 18:25		
	ໂສດາ	ingredient	ຂວາດ	20/08/2022, 18:25		
	ສາເຫຼົ່າຍບັງບັນ	ingredient	ຖຸ	20/08/2022, 18:24		
	ຄ້າໄຟ	ingredient	ໜ້ວຍ	20/08/2022, 18:17		
	ຜົກສີ	ingredient	ກັບ	20/08/2022, 17:59		
	ພົກແລງ	ingredient	ກັບ	20/08/2022, 17:55		
	ພົກເມືອງ	ingredient	ກັບ	20/08/2022, 17:54		
	ກຣະເຄີຍ	ingredient	ກັບ	20/08/2022, 17:54		
	ມະນາວູຈອງ	ingredient	ກັບ	20/08/2022, 17:51		

### รูปประกอบ 3.12 UI การจัดการวัตถุคิบ

### 3.7.6.1. ส่วนแก้ไขวัตถุดิบ

The screenshot shows the Material Manager section of the SmileDong application. On the left is a sidebar with navigation links: Dashboard, Order, Expense, Food Editor (expanded), Material (selected), Item, Itemset, Rider&Pricing, Logout, and a toggle switch. The main area is titled 'Material Manager' and contains a table of existing materials. A modal window is open in the center, allowing the creation of a new material. The modal has fields for Name (input: แก้วใส 16 oz.), Type (dropdown: Ingredient selected), and Unit (input: ลิตร). Below these are 'SAVE' and 'CLOSE' buttons. The table lists materials like แก้วใส 16 oz., ผงชาเขียว, กล่องสูชิกลุบ, กล่องห่อข้าว, ถุงชีส, ถุงเช้า, ถุงสาหร่าย, and ถุงพริกกระเทียม, each with columns for Name, Type, Unit, updatedAt, and Action (edit and delete icons).

รูปประกอบ 3.13 บ.การแก้ไขวัตถุดิบ

### 3.7.7. Item – ตาราง และการลบอาหาร

The screenshot shows the Item Manager section of the SmileDong application. The sidebar is identical to the previous screenshot. The main area is titled 'Item Manager' and contains a table of items. A 'CREATE +' button is visible at the top right. The table has columns for Name, Actions (edit and delete icons), and rows for กุ้งดอง, เม็ดดอง, ข้าว, สาหร่ายธัญ, ถังบัน, ปลาร้าสีฟ้า, and พริกกระเทียม.

รูปประกอบ 3.14 บ. ตารางรายการอาหาร

### 3.7.7.1. ส่วนสร้างอาหาร

The screenshot shows the SmileDong Item Manager interface. On the left is a sidebar with navigation links: Dashboard, Order, Expense, Food Editor (expanded), Material, Item, Itemset, Rider&Pricing, Logout, and a toggle switch. The main area is titled "Item Manager" and "Create new item". It features a large input field at the top. Below it is a table with one row, labeled "1.) ថ្វិក". The table columns are: "Material" (dropdown menu), "ចំណាំ" (text input), "តម្លៃ" (text input), and "តម្លៃសរុប" (text input). At the bottom right of the table are icons for edit and delete.

រូបថតកន្លែង 3.15 ឬ ការសរុបតម្លៃ

### 3.7.7.2. ส่วนแก้ไขอาหาร

The screenshot shows the SmileDong Item Manager interface in editing mode for item ID 62e4df69f4ebc7ddfb92968f. The sidebar is identical to the previous screenshot. The main area is titled "Editing - 62e4df69f4ebc7ddfb92968f". It displays a table with six rows, each representing an ingredient. The table columns are: "កុំពោន់" (label), "Material" (dropdown menu), "ចំណាំ" (text input), "ចំណាំសរុប" (text input), "តម្លៃ" (text input), and "តម្លៃសរុប" (text input). The last row has icons for edit and delete. At the bottom right of the table area are icons for edit and delete.

រូបថតកន្លែង 3.16 ឬ ការແកូខ្លួន

### 3.7.8. Itemset – ตาราง และการคอมเมนูอาหาร

The screenshot shows the 'Itemset Manager' section of the SmileDong application. On the left is a dark sidebar with navigation links: Dashboard, Order, Expense, Food Editor (with Material, Item, and Itemset sub-links), Rider&Pricing, Logout, and a toggle switch. The main area has a header 'Itemset Manager' and a sub-header 'Itemset table'. A 'CREATE +' button is at the top right. Below is a table with columns 'Name' and 'Actions' (edit and delete icons). The table lists seven item sets:

Name	Actions
กีดดอง	
เม่นดอง	
กุ้งเผาดอง	
ເຊົດຖຸງ(ຊີບ)	
ເຊົດນ່ອບ(ຊີບ)	
ເຊົດຖຸນ່ອບ(ຊີບ)	
ເຊົດກຸງ(ຫັນປັນ)	

ຮູບປະກອບ 3.17 UI ตารางและการคอมเมนูอาหาร

#### 3.7.8.1. ส่วนสร้างเมนูอาหาร

The screenshot shows the 'Create new itemset' form within the 'Itemset Manager' section. The sidebar is identical to the previous screenshot. The main area has a header 'Itemset Manager' and a sub-header 'Create new itemset'. It includes fields for 'Name' (input field), 'Work Cost' (0), 'Gas Cost' (0), and a 'GO BACK' button. Below these are dropdowns for '1.) ຂະ' (Item) and 'ຈຳເນວນ' (Quantity), which is set to '1'. There is also a 'ລັບ' (Delete) button with a trash icon. At the bottom are 'ບັນທຶກ' (Save) and 'ຍືນດີ' (Cancel) buttons.

ຮູບປະກອບ 3.18 ບາກາຮສ້າງມັນອາຫານ

### 3.7.8.2. ส่วนแก้ไขเมนูอาหาร

SmileDong

Itemset Manager

Editing - 62e4fa6ec7be9c26f1f7a026

← GO BACK

ลำดับ	Item	จำนวน	ต่อชุด
1.) ใช้	กุ้งดอง	1	
2.) ใช้	กุ้งดอง	1	
3.) ใช้	บีฟรันช์ฟิล์	1	
4.) ใช้	พากะกระเพยบ	1	

Work Cost: 24 Gas Cost: 4

**กุ้งดอง**

จำนวน: 1 ต่อชุด

จำนวน: 1 ต่อชุด

จำนวน: 1 ต่อชุด

จำนวน: 1 ต่อชุด

**บีฟรันช์ฟิล์**

**พากะกระเพยบ**

**กุ้งดอง**

**กุ้งดอง**

ງ្វែរក្រកណ្ឌ 3.19 បាការរក្សាទុយមនុយអាហារ

### 3.7.9. Rider&Pricing - รายชื่อไรเดอร์ การลับໄറเดอร์ และตารางแก้ไขราคา

SmileDong

Rider&Pricing Manager

SELF DELIVERY GRAB LINEMAN SHOPEE

Menu	Price
กุ้งดอง	90
បែបដោ	130
កំពង់បែបដោ	120
មិតកុង(ខ្សី)	115
មិតលែប(ខ្សី)	155
មិតកុងលែប(ខ្សី)	145
មិតកុង(ចងប៊ា)	129
មិតលែប(ចងប៊ា)	169
មិតកុងលែប(ចងប៊ា)	159

Rows per page: 100 1-13 of 13 < >

ງ្វែរក្រកណ្ឌ 3.20 បាការកូវិនុននុយតារាងរាត្រា

### 3.7.9.1. ส่วนสร้างไพรเดอร์

The screenshot shows the SmileDong software interface for managing riders and pricing. On the left, there's a sidebar with navigation links: Dashboard, Order, Expense, Food Editor (with sub-options Material, Item, Itemset), Rider&Pricing (which is selected and highlighted in blue), and Logout. The main area is titled "Rider&Pricing Manager" and features a table for managing menu items. The table has columns for Name (Rider's name), Fees (Fees in percentage), Tax (Tax in percentage), and Gas Cost (optional). The Gas Cost column includes a note: "(optional) Separate gas cost from itemset". The table lists four menu items: กุ้งดอง (Price: 90), ม่อนดอง (Price: 130), กุ้งม่อนดอง (Price: 120), and เม็ดกุ้ง(ชุด) (Price: 115).

Name	Fees	Tax	Gas Cost
Rider's name	Fees in percentage	Tax in percentage	(optional) Separate gas cost from itemset
Menu			Price
กุ้งดอง			90
ม่อนดอง			130
กุ้งม่อนดอง			120
เม็ดกุ้ง(ชุด)			115

รูปประกอบ 3.21 บ) การสร้างแพลตฟอร์ม

## บทที่ 4

### ผลการพัฒนาระบบ (Results)

โครงการพัฒนาโปรแกรมจัดการร้านอาหารเดลิเวอรี่ถือว่าสำเร็จลุล่วงได้ดี โดยตัวโปรแกรมสามารถใช้ได้อย่างถูกต้องและตรงกับเป้าหมายที่ผู้ใช้ต้องการ เช่น การจัดเก็บสูตรอาหารได้อย่างแม่นยำ การจัดเก็บราคาอาหารแยกตามแพลตฟอร์ม การบันทึกรายจ่ายได้อย่างละเอียด และมีระบบการค้นหาหลายชั้น การบันทึกคำสั่งซื้อได้สะดวกรวดเร็ว ไม่จำเป็นต้องป้อนค่าจากแป้นพิมพ์ การคำนวนและแสดงผลสถิติที่เจ้าของร้านต้องการ โดยรวมแล้วผู้ใช้นั้นพึงพอใจอย่างมาก ทั้งยังมีความต้องการที่จะนำโปรแกรมนี้ไปใช้งานและต่อยอดพัฒนาร้านค้าต่อไป

#### 4.1. ผลการทดสอบ (Test report)

วิธีการทดสอบที่เลือกใช้คือการทดสอบแบบ Black-Box Testing โดยให้ผู้ที่ทดสอบเป็นที่ไม่มีความรู้ความเข้าใจในเบื้องหลังการทำงานของระบบภายใน (เจ้าของร้าน) เพื่อทดสอบทุกส่วนที่เป็นจุดประสงค์ของการพัฒนาโปรแกรม จากผู้ที่จะนำไปใช้จริง โดยได้ผลการทดสอบดังนี้

<b>PROJECT :</b>	Delivery Restaurant Manager		
<b>TESTER :</b>	Thanyaporn Anekawieng	<b>TEST DATE :</b>	25/10/2022
<b>PREPARED BY :</b>	Thachin Wisutimarkul	<b>DOCUMENTED BY :</b>	Thachin Wisutimarkul

ตาราง 4.1 ข้อมูลการทดสอบ

<b>MODULE :</b> Login	<b>TEST OBJECTIVE :</b> Authentication test
<b>TEST DATA :</b> email = admin@smiledong.com, password = 00000000	

Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Empty input		Toast message “Invalid credentaisl”	Pass
2	Empty email	password = 00000000	Toast message “Invalid credentaisl”	Pass
3	Empty password	email = admin@smiledong.com	Toast message “Invalid credentaisl”	Pass
4	Wrong email	email = ad@smiledong.com, password = 00000000	Toast message “Invalid credentaisl”	Pass
5	Wrong password	email = admin@smiledong.com, password = 0000	Toast message “Invalid credentaisl”	Pass
6	Correct email and password	email = admin@smiledong.com, password = 00000000	Redirected to dashboard	Pass

ตาราง 4.2 ทดสอบ ระบบตรวจสอบตัวตน

MODULE : Dashboard		TEST OBJECTIVE : Sales report display test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Default display	Display = default Rider = default Time = default	Show tables with sales of every rider in span of 1 month	Pass
2	Select rider	Display = default Rider = GRAB Time = default	Show tables with sales of rider grab in span of 1 month	Pass

3	Select time range	Display = default Rider = default Time = Y	Show tables with sales of every rider in span of 1 year	Pass
4	Toggle graph view	Display = graph Rider = default Time = default	Show bar chart with sales of every rider in span of 1 month	Pass
5	Graph select rider	Display = graph Rider = GRAB Time = default	Show bar chart with sales of rider grab in span of 1 month	Pass
6	Graph select time range	Display = graph Rider = default Time = Y	Show graph with sales of every rider in span of 1 year	Pass

ตาราง 4.3 ทดสอบ การแสดงผลต่อการขาย

MODULE : Order		TEST OBJECTIVE : Add new order test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Empty input		Nothing happened	Pass
2	Select item without select rider		Unable to click item therefore nothing happened	Pass
3	Only send time data	Time = 25/10/2022	Nothing happened	Pass
4	Only select rider	Rider = Grab	Nothing happened	Fail : Saved with empty items
5	Select rider then select item	Item = pickled shrimp Rider = Grab	Saved with Grab as rider and 1 pickled shrimp as items	Pass

6	Select rider then select item 3 times	Item = pickled shrimp Rider = Grab	Saved with Grab as rider and 3 pickled shrimp as items	Pass
7	Select rider then select several items	Item = [pickled shrimp, rice, seaweed] Rider = Lineman	Saved with Lineman as rider and 1 pickled shrimp, 1 rice, 1 seaweed as items	Pass
8	1. Select rider 2. Select several items 3. Select first item 2 more times	Item = [pickled shrimp, rice, seaweed] Rider = Lineman	Saved with Lineman as rider and 2 pickled shrimp, 1 rice, 1 seaweed as items	Pass
9	1. Select rider 2. Select several items 3. Reduce first item by 1	Item = [pickled shrimp, rice, seaweed] Rider = Lineman	Saved with Lineman as rider and 1 rice, 1 seaweed as items	Pass
10	1. Select rider 2. Select several items 3. Select first item 4 more times 4. Reduce first item by 2	Item = [pickled shrimp, rice, seaweed] Rider = Lineman	Saved with Lineman as rider and 3 pickled shrimp, 1 rice, 1 seaweed as items	Pass

ตาราง 4.4 ทดสอบเพิ่มคำสั่งชื้อ

MODULE : Order		TEST OBJECTIVE : Delete order test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click delete icon on order		Order removed from table	Pass

ตาราง 4.5 ทดสอบ ลบคำสั่งซื้อ

MODULE : Expense Category		TEST OBJECTIVE : Add new category test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Empty input		Nothing happened	Pass
2	Empty name	Parent = Business	Nothing happened	Pass
3	Empty parent	Name = testing	Add new main category name 'testing'	Pass
4	Duplicate name	Name = Business	Nothing happened	Pass
5	Select Business as parent	Name = testing Parent = Business	Add new sub category name 'testing' under 'Business' category	Pass

ตาราง 4.6 ทดสอบ สร้างหมวดหมู่รายจ่าย

MODULE : Expense Category		TEST OBJECTIVE : Delete category test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Delete sub-category		Category removed from the list	Pass
2	Delete main category		All sub-category under main category and the main category itself is removed from the list	Pass

ตาราง 4.7 ทดสอบ ลบหมวดหมู่รายจ่าย

MODULE : Expense		TEST OBJECTIVE : Add new expense test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]

1	Empty input Save & Go Back		Toast message with error that occurred	Pass
2	Only send category input Save & Go Back	Category ingredient =	Redirect to expense table  Toast message with error that occurred	Pass
3	Only send description input Save & Go Back	Description = testing	Redirect to expense table  Toast message with error that occurred	Pass
4	Only send amount input Save & Go Back	Amount = 5	Redirect to expense table  Toast message with error that occurred	Pass
5	Only send total input Save & Go Back	Total = 200	Redirect to expense table  Toast message with error that occurred	Pass
6	Only send date&time input Save & Go Back	Date&Time =10/04/2022 00:00	Redirect to expense table  Toast message with error that occurred	Pass
7	Select category ingredient Save & Go Back	Category ingredient =	Redirect to expense table  Description input form turn into searchable dropdown input with ingredient list	Pass

8	Select category packaging  Save & Go Back	Category = packaging	Redirect to expense table  Description input form turn into searchable drop-down input with packaging list	Pass
9	Select category with sub-category	Category = Business	New searchable drop-down input appear beside category input call Sub-category input with Business's sub category list	Pass
10	Select ingredient category  Select ingredient Empty amount, total and date  Save & Go Back	Category ingredient =  Ingredient = Shrimp	Redirect to expense table  Toast message with error that occurred	Pass
11	Select ingredient category  Select ingredient Enter amount Empty total and date  Save & Go Back	Category ingredient =  Ingredient = Shrimp Amount = 100	Redirect to expense table  Toast message with error that occurred	Pass
12	Select ingredient category  Select ingredient Enter amount	Category ingredient =  Ingredient = Shrimp Amount = 100	Redirect to expense table  Saved as Ingredient category, Descript as	Pass

	Enter total Empty date Save & Go Back	Total = 345	Shrimp, Amount 100, Total 345 and Date&Time is current time.	
13	Fill every input Save & Go Back	Category = ingredient Ingredient = Shrimp Amount = 100 Total = 345 Date&Time = 05/10/2022, 00:00	Redirect to expense table Saved as Ingredient category, Descript as Shrimp, Amount 100, Total 345 and Date&Time is 05/10/2022, 00:00	Pass
14	Fill every input Save & Add new expense	Category = ingredient Ingredient = Shrimp Amount = 100 Total = 345 Date&Time = 05/10/2022, 00:00	Saved as details Clear every input except Category and Date&Time	Pass

ตาราง 4.8 ทดสอบ สร้างรายจ่าย

MODULE : Expense		TEST OBJECTIVE : Delete expense test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click delete icon on expense		Expense removed from table	Pass

ตาราง 4.9 ทดสอบ ลบรายจ่าย

MODULE : Expense		TEST OBJECTIVE : Repeat expense test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]

1	Click duplicate icon on expense		New expense add to the table with same details as select order except Date; which save as current time Toast message “Create successfully”	Pass
---	---------------------------------	--	--	------

ตาราง 4.10 ทดสอบ ทำซ้ำรายจ่าย

MODULE : Expense		TEST OBJECTIVE : Edit expense test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click edit icon on expense	Category = Equipment Description = Glove Amount = 50 Total = 49 Date&Time = 10/27/2022 16:36	Open edit form with current expense data	Pass
2	Modify input Save & Go back	Description = Scissors Amount = 2 Total = 69	Redirect to expenses table Glove expense at 10/27/2022 16:36 change to Scissors with amount of 2 and total of 69	Pass

ตาราง 4.11 ทดสอบ แก้ไขรายจ่าย

MODULE : Materials		TEST OBJECTIVE : Add new material test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]

1	Empty input Click ADD		Toast message with error that occurred	Pass
2	Empty unit Click ADD	Name = test	Toast message with error that occurred	Pass
3	Empty name Click ADD	unit = testUnit	Toast message with error that occurred	Pass
4	Fill all input Click ADD	Name = test Unit = testing Type = materials	Add new material name test, type materials, unit testing into materials table	Pass
5	Duplicate name Click ADD	Name = shrimp Unit = testing Type = materials	Toast message with error that occurred	Pass

ตาราง 4.12 ทดสอบเพิ่มวัตถุคิบ

MODULE : Materials		TEST OBJECTIVE : Edit material test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click edit icon on material	Name = test Type = Packaging Unit = testing	Open edit form with current material data	Pass
2	Modify input Save	Name = test2 Type = Ingredient Unit = testing2	test material change to test2 with type Ingredient and unit testint2	Pass
3	Modify to duplicate name and Save	Name = Shrimp	Toast message with error that occurred	Pass

ตาราง 4.13 ทดสอบแก้ไขวัตถุคิบ

MODULE : Materials		TEST OBJECTIVE : Delete material	

Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click delete icon on material		Material removed from table	Pass

ตาราง 4.14 ทดสอบ ลบวัตถุคืน

MODULE : Items		TEST OBJECTIVE : Add new item		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Save with empty input		Nothing happen	Pass
2	Save with only name	Name = test	Redirect to tables New Item name test appear in the table	Pass
3	Enter name Add new material Empty material input Save	Name = test	Nothing happen	Pass
4	Enter name Add material with default amount Save	Name = test Material = Shrimp	Save with name test and Shrimp amount 1	Pass
5	Add several materials and save	Name = test Material = [Shrimp, Salmon] Amount = [1,10]	Save with name test 1 Shrimp, 10 Salmon,	Pass

ตาราง 4.15 ทดสอบ เพิ่มอาหาร

MODULE : Items		TEST OBJECTIVE : Edit item test	

Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click edit icon on material	Name = test Material = [Shrimp, Salmon] Amount = [1,10]	Open edit form with current item data	Pass
2	Modify input Save	Name = DoubleTrouble Material = [Shrimp, Salmon] Amount = [5,45]	test item name change to DoubleTrouble Shrimp amount change to 5 Salmon amount change to 45	Pass

ตาราง 4.16 ทดสอบแก้ไขข้อมูล

MODULE : Items		TEST OBJECTIVE : Delete item		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click delete icon on item		item removed from table	Pass

ตาราง 4.17 ทดสอบลบรายการ

MODULE : Itemsets		TEST OBJECTIVE : Add new itemsets		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Save with empty input		Nothing happen	Pass
2	Save with only name	Name = test	Redirect to tables New Item name test appear in the table	Pass
3	Enter name	Name = test	Nothing happen	Pass

	Add new item Empty item input Save			
4	Enter name Add item with default amount Save	Name = test Material = Pickled Shrimp Save	Save with name test Pickled Shrimp amount 1 workcost 0 and gascost 0	Pass
5	Add several items and save	Name = test Material = [PickledShrimp, PickledSalmon] Amount = [1,10]	Save with name test 1 PickledShrimp, 10 PickledSalmon, workcost 0 and gascost 0	Pass
6	Add several items, fill every inputs and save	Name = test Material = [PickledShrimp, PickledSalmon] Amount = [1,10] workCost = 20 gasCost = 6	Save with name test 1 PickledShrimp, 10 PickledSalmon, workCost 20 and gasCost 6	

ตาราง 4.18 ทดสอบเพิ่มเมนูอาหาร

MODULE : Itemsets		TEST OBJECTIVE : Edit itemset test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click edit icon on material	Name = test Material = [PickledShrimp, PickledSalmon] Amount = [1,10] workCost = 20 gasCost = 6	Open edit form with current item data	Pass

2	Modify input Save	Name = MeatMeat Amount = [2,2]	test item name change to MeatMeat PickledShrimp amount change to 2 PickledSalmon amount change to 2	Pass
---	----------------------	-----------------------------------	--	------

ตาราง 4.19 ทดสอบแก้ไขเมนูอาหาร

MODULE : Itemsets		TEST OBJECTIVE : Delete itemset test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click delete icon on itemset		itemset removed from table	Pass

ตาราง 4.20 ทดสอบลบเมนูอาหาร

MODULE : Riders&Pricing		TEST OBJECTIVE : Add new rider test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click + icon		Rider create form open	Pass
2	Click - icon		Rider create form close without saving	Pass
3	Save with empty input		Toast message with error that occurred	Pass
4	Save with only name input	Name = test	Toast message with error that occurred	Pass
5	Save with only fees input	Fees = 30	Toast message with error that occurred	Pass
6	Save with only tax input	Tax = 7	Toast message with error that occurred	Pass

7	Save with only gas cost input	GasCost = 1	Toast message with error that occurred	Pass
8	Save without gas cost input	Name = test Fees = 30 Tax = 7	Save test to rider database with fees 30, tax 7 and gasCost 0  test appear on rider list.  When access test; prices table show every menu with 0 in price	Pass
9	Save with every input filled	Name = test Fees = 30 Tax = 7 GasCost = 1	Save test to rider database with fees 30, tax 7 and gasCost 1  test appear on rider list.  When access test; prices table show every menu with 0 in price	Pass

ตาราง 4.21 ทดสอบ เพิ่มแพลตฟอร์ม

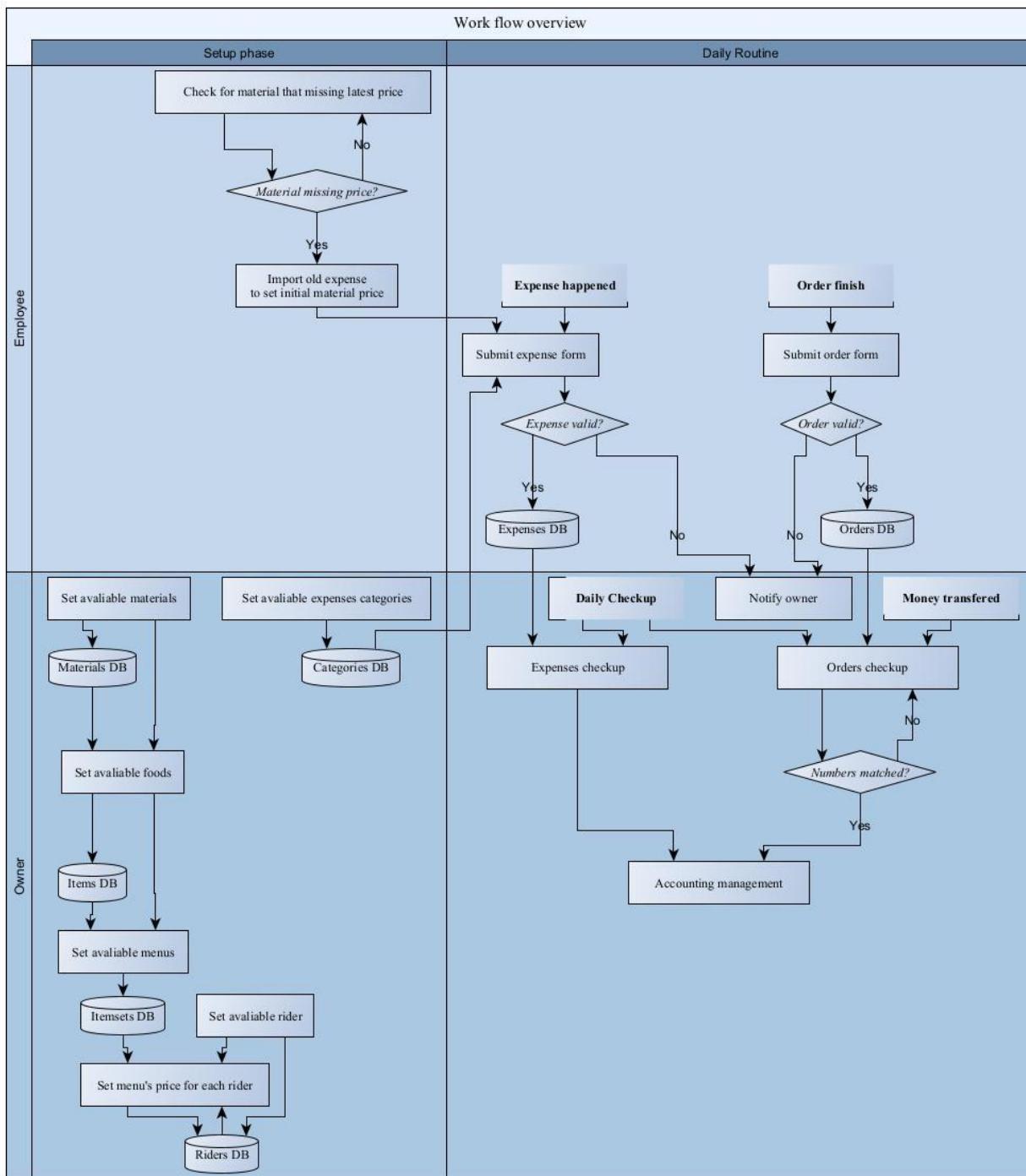
MODULE : Riders&Pricing		TEST OBJECTIVE : Delete rider test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Click + icon		Rider create form open	Pass

ตาราง 4.22 ทดสอบ ลบแพลตฟอร์ม

MODULE : Riders&Pricing		TEST OBJECTIVE : Set menu price test		
Case no.	Detail	Data	Expected Results	Status [Fail/Pass]
1	Double click on price number		Enter price editing mode	Pass
2	Enter price in string	Price = abcd	Nothing happened	Pass
3	Enter price in number	Price = 100	Price save as 100	Pass

ตาราง 4.23 ทดสอบ ตั้งราคาเมนูอาหาร

## 4.2. ผลลัพธ์ของระบบที่พัฒนาในภาพรวม (Output)



§ ประกอบ 4.1 การทำงานในภาพรวม

โดยรวมแล้วระบบสามารถรวมกระบวนการทำงานทั้งหมดไว้ในแอพพลิเคชันเดียวได้สำเร็จ  
ไม่มีความจำเป็นต้องแยกการทำงานไปไว้ในหลายระบบหลายอุปกรณ์เหมือนระบบเก่าที่ทางร้านเคยใช้งาน

ผู้ใช้สามารถเก็บข้อมูลสูตรอาหารและวัตถุดิบต่างๆที่ใช้ภายในร้านได้อย่างสมบูรณ์

**Item Manager**

Editing - 63027d506e9fca2920674207 ← GO BACK

บันดลง					
1.) ใช้	Material น้ำตาลกราย	จำนวน	5	กรับ	ต่อชุด
2.) ใช้	Material น้ำตาลมะพร้าว	จำนวน	5	กรับ	ต่อชุด
3.) ใช้	Material ไข่มุกากุย	จำนวน	1	มีลลิลิต	ต่อชุด
4.) ใช้	Material น้ำเปล่า	จำนวน	58.3333	มีลลิลิต	ต่อชุด
5.) ใช้	Material ค่าไฟ	จำนวน	16	มีลลิลิต	ต่อชุด
6.) ใช้	Material ซีอิ๊วขาว	จำนวน	1.5	กรับ	ต่อชุด
<span style="font-size: 2em; color: #ccc;">✚</span> <span style="color: green;">💾</span> <span style="color: red;">✖</span>					

รูปประกอบ 4.2 การเก็บสูตรอาหาร

แต่ละแพลตฟอร์มจะมีราคาของทุกเมนูที่แตกต่างกัน สามารถปรับแต่งได้ตามที่ผู้ใช้ต้องการ  
แบบที่แอพพลิเคชันจัดการคำสั่งซื้อทั่วไปไม่สามารถทำได้ง่ายๆ

Menu	Price
กุ้งดอง	129
มือบดอง	199
กุ้งมือบดอง	169
เข็มกุ้ง(ชุด)	165
เข็มมือบด(ชุด)	230
เข็มกุ้งมือบด(ชุด)	200
เข็มกุ้ง(ยังบัน)	180
เข็มมือบด(ยังบัน)	240
เข็มกุ้งมือบด(ยังบัน)	205
⋮	⋮
Rows per page:	100
	1-13 of 13
	< >

รูปประกอบ 4.3 การตั้งราคาแยกตามแพลตฟอร์ม

ส่วนการลงรายจ่ายภายในร้านมีระบบช่วยเหลือให้เลือกวัตถุดีป้าได้ง่ายขึ้น สามารถแบ่งหมวดหมู่ได้อย่างชัดเจน สามารถสร้างหมวดหมู่อยู่เพื่อความแม่นยำในการบันทึกรายจ่าย ระบบยังมีคำสั่งในการทำข้าวคำสั่งซื้อเพื่อความรวดเร็วในการเพิ่มรายจ่ายที่มีการเพิ่มบ่ออยๆ อีกด้วย

**Expense Manager**

**Create new expense**

Category	Business	Sub-Category	Monthly-fees	CATEGORY EDIT
Expense description/name	Foodpanda	Amount	1	
Date&Time	Total	200	B	Total cost for this expenses (฿)
<input type="button" value="SAVE &amp; ADD NEW EXPENSE"/>		<input type="button" value="← SAVE &amp; GO BACK"/>		

ข้อปฏิรักษ์ฉบับ 4.4 การสร้างคำสั่งซื้อ

**Expense Manager**

**Expense records**

**CREATE +**

**CATEGORY EDIT**

Category	Name	Amount	Total	Date ↓	Actions
Ingredient	เมนูอุ๊ก	100 ถุง	75 ฿	27/10/2022, 21:49	
Equipment	น้ำแข็ง	1	69 ฿	27/10/2022, 16:36	
Ingredient	ผัก	100 กก	345 ฿	27/10/2022, 16:28	
Ingredient	พริกแดง	500 กรัม	45 ฿	21/10/2022, 10:55	
Ingredient	ผัก	100 กก	340 ฿	21/10/2022, 10:54	
Packaging	กล่องเบเยิลชุบ	50 กล่อง	90 ฿	10/10/2022, 10:03	
Ingredient	พริกแดง	500 กรัม	45 ฿	09/10/2022, 22:57	
Ingredient	เมนูอุ๊ก	100 ถุง	75 ฿	08/10/2022, 00:00	
Ingredent	น้ำตาลคราฟ	1000 กรัม	43 ฿	08/10/2022, 00:00	

ข้อปฏิรักษ์ฉบับ 4.5 การทำรายการซื้อ

ในส่วนของการแสดงรายจ่าย ระบบสามารถคัดกรองข้อมูลได้อย่างละเอียด ค้นหาตามชื่อ ค้นหาตามประเภท ค้นหาตามจำนวน ฯลฯ รวมถึงยังสามารถจัดเรียงข้อมูลได้ตามต้องการอีกด้วย

Expense records					<a href="#">CREATE +</a>
Category	Name	Amount	Total	Date	Actions
Ingredient	กุ้ง	100 บาท	345 บาท	27/10/2022, 16:28	
Ingredient	กุ้ง	100 บาท	366 บาท	27/10/2022, 00:00	
Ingredient	กุ้ง	100 บาท	354 บาท	25/10/2022, 00:00	
Ingredient	กุ้ง	100 บาท	372 บาท	23/10/2022, 00:00	
Ingredient	กุ้ง	100 บาท	340 บาท	21/10/2022, 10:54	
Ingredient	กุ้ง	100 บาท	331 บาท	21/10/2022, 00:00	

รูปประกอบ 4.6 การค้นหาข้อมูลในตาราง

ในการบันทึกคำสั่งซื้อ ผู้ใช้งานสามารถบันทึกคำสั่งซื้อได้โดยโดยตรงเพียงเลือกแพลตฟอร์ม เลือกเมนูอาหารและกดบันทึก ทำให้กระบวนการในการบันทึกคำสั่งซื้อเป็นไปอย่างรวดเร็วและง่ายดายกว่าระบบเก่า ของทางร้านมาก

**Order Manager**

**Editor**

[SELF DELIVERY](#) [GRAB](#) [LINEMAN](#) [SHOPEE](#) [TEST](#)

[กุ้งดอง](#) [มีนบดง](#) [กุ้งมีนบดง](#) [เขิตกุ้ง\(ธูป\)](#) [เขิตบอน\(ธูป\)](#)

[เขิตกุ้งบอน\(ธูป\)](#) [เขิตกุ้ง\(ยังปัน\)](#) [เขิตบอน\(ยังปัน\)](#) [เขิตกุ้งบอน\(ยังปัน\)](#)

[สาหร่ายยูชี](#) [สาหร่ายยังปัน](#) [ข้าวเปลือก](#) [บุเดลลิงชาบะ](#)

**Details**

Order from Grab

+ 2x เขิตบอน(ยังปัน)	480฿
- 1x กุ้งดอง	129฿
+ 1x สาหร่ายยูชี	20฿
<b>Totals: 629</b>	

Date&Time

รูปประกอบ 4.7 การบันทึกคำสั่งซื้อ

ระบบมีการแสดงผลการขายที่รวมผลการขายของทุกแพลตฟอร์มไว้ด้วยกัน และยังสามารถกำหนดช่วงระยะเวลาที่ต้องการดูสถิติได้อีกด้วย



ຮູບປະກອບ 4.8 ການແຄສດສັນທິການຂາຍ

### 4.3. ລົບປະເມີນຄວາມພຶ້ງພອໃຈຂອງຜູ້ໃຊ້ຮບ (Assessment)

ພນັກງານຮ້ານ

ຂໍ້ຕຳດາມ	ຮະດັບຄວາມພຶ້ງພອໃຈ				
	ນ້ອຍທີ່ສຸດ	ນ້ອຍ	ປານກລາງ	ມາກ	ມາກທີ່ສຸດ
1. ຄວາມຈ່າຍໃນການໃຊ້ຈໍານານ				/	
2. ກຽບວຸນການທຳການຂອງຮະບບ			/		
3. ຄວາມຄຸກຕ້ອງຂອງຮະບບ					/
4. ຕຽດຕາມວັດຖຸປະສົງຄົກທີ່ ຕ້ອງການ					/
5. ກາຣອອກແບບເຂົ້າໃຈຈ່າຍ ໝາຍ ໜັບໜັນ	ໄມ່			/	
6. ຄວາມສະດວກໃນການໃຊ້ຈໍານານ				/	
7. ຄວາມເໜາມສົມໃນການໃຊ້ຈໍານານ				/	
8. ຄວາມພຶ້ງພອໃຈໃນການໃຊ້ຈໍານານ				/	
9. ປະໂຍື່ນໃນການໃຊ້ຈໍານານ				/	

ຕາງໆ 4.24 ແນບປະເມີນຄວາມພຶ້ງພອໃຈຂອງພນັກງານຮ້ານ

ເຈົ້າຂອງຮ້ານ

ข้อคำถาม	ระดับความพึงพอใจ				
	น้อยที่สุด	น้อย	ปานกลาง	มาก	มากที่สุด
1. ความง่ายในการใช้งาน					/
2. กระบวนการทำงานของระบบ				/	
3. ความถูกต้องของระบบ					/
4. ตรงตามวัตถุประสงค์ที่ต้องการ					/
5. การออกแบบเข้าใจง่าย ไม่ซับซ้อน				/	
6. ความสะดวกในการใช้งาน				/	
7. ความเหมาะสมในการใช้งาน				/	
8. ความพึงพอใจในการใช้งาน				/	
9. ประโยชน์ในการใช้งาน					/

ตาราง 4.25 แบบประเมินความพึงพอใจของเจ้าของร้าน

#### 4.4. คู่มือการติดตั้ง (Installation)

4.4.1. คุณสมบัติของเครื่องคอมพิวเตอร์ที่สามารถติดตั้งโปรแกรมนี้ได้ (Hardware และ Software requirement)

##### Hardware

1. Server
  - a. CPU 3GHz or faster with 4 or more cores
  - b. At least 4GB Memory
  - c. Hard disk free space at least 1.2GB
2. Client
  - a. Pentium 4 processor or higher
  - b. At least 1GB Memory
  - c. Hard disk free space at least 100MB

##### Software

1. Server

- a. Linux
  - b. SmartOS
  - c. macOS 10.7 or higher
  - d. Windows Server 2008 or higher
  - e. Node.js 16.0 or higher
  - f. NPM 8.0 or higher
  - g. Git 2.37 or higher
2. Client
- a. Windows 7, Windows 8, Windows 8.1, Windows 10 or later
  - b. macOS High Sierra 10.13 or later
  - c. 64-bit Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, or Fedora Linux 32+
  - d. Android 7.0 Nougat or later

#### 4.4.2. รายละเอียดการติดตั้ง

0. จัดเตรียมฐานข้อมูล MongoDB ผู้ใช้สามารถเลือกที่จะใช้ local server ผ่าน community server หรือใช้บริการ MongoDB Atlas ก็ได้ โดยสามารถศึกษาเพิ่มเติมได้ที่ <https://www.mongodb.com/docs/atlas/getting-started/>

1. ตรวจสอบให้แน่ใจว่า server สามารถใช้ Node.js และ git ได้
2. เปิด command prompt หรือ terminal ในตำแหน่งที่ต้องการติดตั้ง
3. คัดลอกคำสั่งต่อไปนี้ลงใน command prompt หรือ terminal ที่ใช้แล้วรันคำสั่ง

```
git clone https://github.com/nickethachin/DeliveryRestaurantManager.git --branch CS499 && cd DeliveryRestaurantManager && npm install --force && npm install -prefix frontend -force && npm run build -prefix frontend
```

```
C:\>git clone https://github.com/nickethachin/DeliveryRestaurantManager.git --branch CS499 && cd DeliveryRestaurantManager && npm install --force && npm install -prefix frontend -force && npm run build -prefix frontend
```

รูปประกอบ 4.9 คำสั่งสำหรับติดตั้งโปรแกรม

4. ร่อนการติดตั้งโปรแกรมเสรีจสมบูรณ์
5. สร้างไฟล์ใหม่ภายในโฟลเดอร์ของโปรแกรม ตั้งชื่อว่า “.env”
6. แก้ไขไฟล์ .env ด้วยเนื้อหาดังนี้

```
NODE_ENV = production
```

```
PORT = พอร์ตที่ต้องการให้โปรแกรมรัน
```

```
MONGO_URI = URI สำหรับเชื่อมต่อ กับ MongoDB
```

```
JWT_SECRET = รหัสสำหรับสร้างและตรวจสอบ login Token
```

#### 6.1 ตัวอย่างไฟล์ .env

```
.env
1 NODE_ENV = development
2
3 PORT = 5000
4
5 MONGO_URI = mongodb+srv://
   restaurant:00000000@mycluster.a0bcdef.mongodb.
   net/databaseName?retryWrites=true&w=majority
6
7 JWT_SECRET = DRMSecret
8
```

รูปประกอบ 4.10 ตัวอย่างไฟล์ .env

7. จากนั้นสามารถรันคำสั่ง npm start ได้ เมื่อต้องการรันเซิฟเวอร์

```
D:\Projects\DeliveryRestaurantManager>npm start
```

รูปประกอบ 4.11 คำสั่งเริ่มต้นระบบ

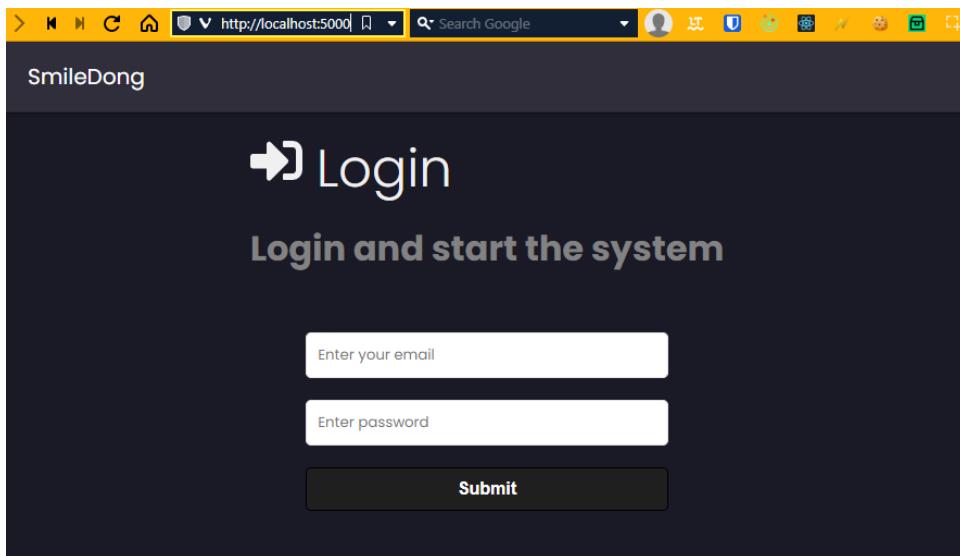
8. ร่อนโปรแกรมทำงานจนเสร็จสมบูรณ์

```
> deliveryrestaurantmanager@1.0.0 start
> node backend/server.js

Server started on port 5000
MongoDB Connect: ac-iuks7vb-shard-00-02.t6okwi.y.mongodb.net
```

รูปประกอบ 4.12 ตัวอย่างเมื่อเริ่มต้นระบบสำเร็จ

9. เมื่อเชิฟเวอร์ทำงานแล้ว เราสามารถเข้าถึงโปรแกรมได้โดยการเข้าผ่าน Web browser และใช้ url ของเครื่อง และ port ที่กำหนด ในกรณีที่เครื่องที่รันเชิฟเวอร์กับเครื่องที่จะเข้าใช้โปรแกรมเป็นเครื่องเดียวกัน และ port ที่กำหนดคือ 5000 ผู้ใช้สามารถเข้าถึงโปรแกรมได้ที่ <http://localhost:5000>



รูปประกอบ 4.13 ตัวอย่างหน้าเข้าสู่ระบบ

10. เมื่อเข้ามาแล้วระบบจะต้องการให้เราล็อกอิน ให้เราไปที่ <http://localhost:5000/register> เพื่อสมัครสมาชิกก่อน โดยซ่องแรกให้กรอกชื่อที่ต้องการ ช่องที่สองกรอกอีเมล์ ช่องสามและสี่กรอกพาสเวิร์ดที่ต้องการ จากนั้นกดปุ่ม Submit ด้านล่าง

SmileDong

## Register

Please create an account





**Submit**

รูปประกอบ 4.14 ตัวอย่างการกรอกข้อมูลหน้าสมัครสมาชิก

11. เมื่อสมัครสมาชิกสำเร็จและเข้าสู่หน้า Dashboard ได้ ก็ถือได้ว่าติดตั้งเสร็จสมบูรณ์แล้ว

#### 4.5. คู่มือการใช้งาน (User manual)

- เมื่อเข้าใช้งานครั้งแรก ผู้ใช้จะต้องเข้าสู่ระบบก่อน ซึ่งแรกใส่ email และซองที่สองใส่ password ที่ได้สมัครไว้ในตอนติดตั้งโปรแกรม

➔ Login

Login and start the system

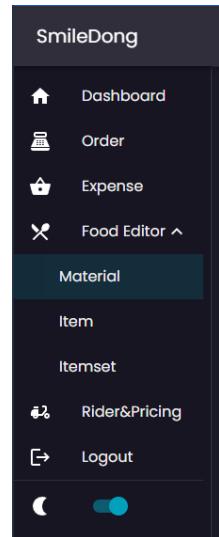


**Submit**

รูปประกอบ 4.15 ลักษณะของหน้าเข้าสู่ระบบ

2. จากนั้น ผู้ใช้จะต้องตั้งค่าเมนูอาหาร ไรเดอร์ และราคาอาหาร โดยกระบวนการนี้ จะทำเพียงครั้งเดียวเท่านั้น เว้นเสียแต่ว่า จะมีการปรับราคาหรือสูตรอาหาร

### 2.1 คลิกไปที่เมนู Material ทางฝั่งซ้ายของโปรแกรม



รูปประกอบ 4.16 ตำแหน่งเมนู Material

2.2 จากนั้น ทำการเพิ่มวัตถุดิบภายในร้านทีละอย่าง สามารถทำได้โดยการกรอกลงไปภายในช่องข้อมูลด้านบน จากซ้ายไปขวา ช่องที่ต้องกรอกจะเป็น ชื่อ ประเภท และหน่วย แล้วกดปุ่ม ADD

ชื่อ คือชื่อของวัตถุดิบที่จะเพิ่ม

ประเภท คือ ประเภทว่าเป็น Ingredient (วัตถุดิบ) หรือ Packaging (บรรจุภัณฑ์)

หน่วย คือ หน่วยเรียกของวัตถุคิบน้ำ เช่น ชิ้น ตัว กรัม ขวด มิลลิกรัม

Name	Type	Unit	updatedAt	Action
แพ็คไข่ 16 oz.	package	ไข่	21/10/2022, 10:10	<span>edit</span> <span>delete</span>
ผงชาเขียว	ingredient	กรัม	21/10/2022, 10:09	<span>edit</span> <span>delete</span>
กล่องสองหัว	package	กล่อง	12/09/2022, 01:51	<span>edit</span> <span>delete</span>
กล่องหนึ่งหัว	package	กล่อง	12/09/2022, 01:51	<span>edit</span> <span>delete</span>
ตะเกียง	package	ตู่	12/09/2022, 01:51	<span>edit</span> <span>delete</span>
ถุงซีล	package	ถุง	20/08/2022, 18:28	<span>edit</span> <span>delete</span>
ถุงข้าว	package	ถุง	20/08/2022, 18:27	<span>edit</span> <span>delete</span>
ถุงสาหร่าย	package	ถุง	20/08/2022, 18:27	<span>edit</span> <span>delete</span>
ถุงพลาสติกใส่เก็บ	package	ถุง	20/08/2022, 18:27	<span>edit</span> <span>delete</span>

รูปประกอบ 4.17 การสร้างวัตถุคิบ

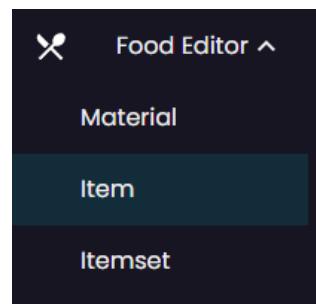
2.2.1 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตารางด้านล่าง

2.2.2 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.2.3 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

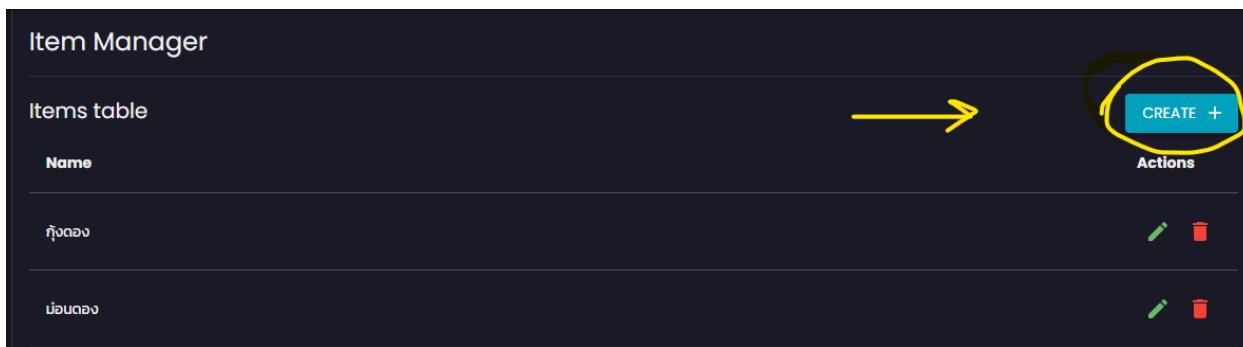
2.3 ทำชาไปเรื่อยๆจนกว่าจะลงทะเบียนวัตถุคิบและบรรจุภัณฑ์ที่ใช้ภายในร้านจนหมด

2.4 ไปที่เมนู Item ทางด้านซ้ายของโปรแกรม



รูปประกอบ 4.18 ดำเนินการ Item

## 2.5 กดปุ่ม Create ทางด้านขวาเพื่อเริ่มสร้างอาหาร



รูปประกอบ 4.19 ตำแหน่งปุ่มสร้างอาหาร

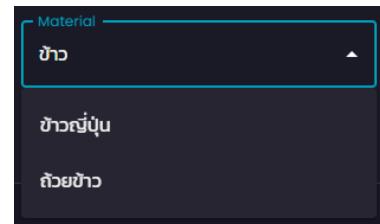
## 2.6 สร้างอาหารโดยกรอกข้อมูลดังนี้

### 2.6.1 ซ่องว่างซ่องแรก กรอกชื่ออาหารที่ต้องการ

### 2.6.2 กรอกวัตถุดิบที่ใช้

2.6.2.1 กดปุ่ม เพื่อเพิ่มวัตถุดิบ

2.6.2.2 ในช่อง Material สามารถกดปุ่มสามเหลี่ยมเล็กๆ หรือพิมพ์หาช่องที่ใช้



รูปประกอบ 4.20 Searchable dropdown

2.6.2.2.1 หากไม่มีวัตถุดิบที่ต้องการ กลับไปทำข้อ 2.2

2.6.2.3 ในช่องจำนวน ให้กรอกจำนวนของที่ใช้ต่ออาหาร 1 ชุด

2.6.2.4 สามารถกดปุ่ม เพื่อเปิดเครื่องคิดเลขได้

2.6.3 กดปุ่ม ทางด้านขวา เพื่อบันทึกอาหาร

2.6.4 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตาราง

2.6.5 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

2.6.6 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.6.7 ทำขั้นตอนมีอาหารเท่าที่ต้องการ

2.7 สร้างเมนูอาหารโดยกรอกข้อมูลดังนี้

2.7.1 ช่องว่างซ่องแรก กรอกชื่อเมนูอาหารที่ต้องการ

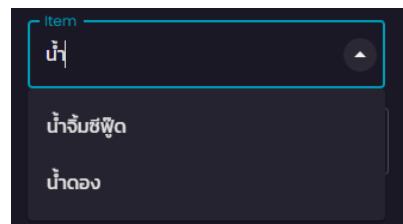
2.7.2 ช่อง Work Cost กรอกต้นทุนค่าแรงที่ใช้ในการทำเมนูนี้

2.7.3 ช่อง Gas Cost กรอกต้นทุนค่าน้ำมันรถที่ใช้ในการทำเมนูนี้

2.7.4 กรอกอาหารที่ใช้ประกอบเมนูนี้

 เพื่อเพิ่มวัตถุดิบ

2.7.4.2 ในช่อง Item สามารถกดปุ่มสามเหลี่ยมเล็กๆ หรือพิมพ์ชื่อที่ใช้



2.7.4.2.1 หากไม่มีอาหารที่ต้องการ กลับไปทำข้อ 2.6

2.7.4.3 ในช่องจำนวน ให้กรอกจำนวนของที่ใช้ต่อเมนู 1 เมนู

 เพื่อเปิดเครื่องคิดเลขได้

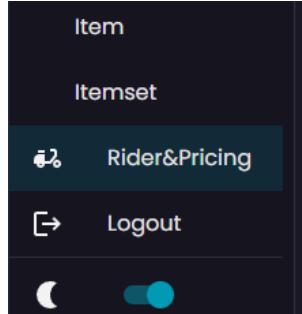
2.7.5 กดปุ่ม  ทางด้านขวา เพื่อบันทึกอาหาร

2.7.6 เมื่อเพิ่มสำเร็จข้อมูลจะถูกเพิ่มไปยังตาราง

2.7.7 ผู้ใช้สามารถกดปุ่มดินสอสีเขียวทางด้านขวาเพื่อแก้ไขข้อมูลได้

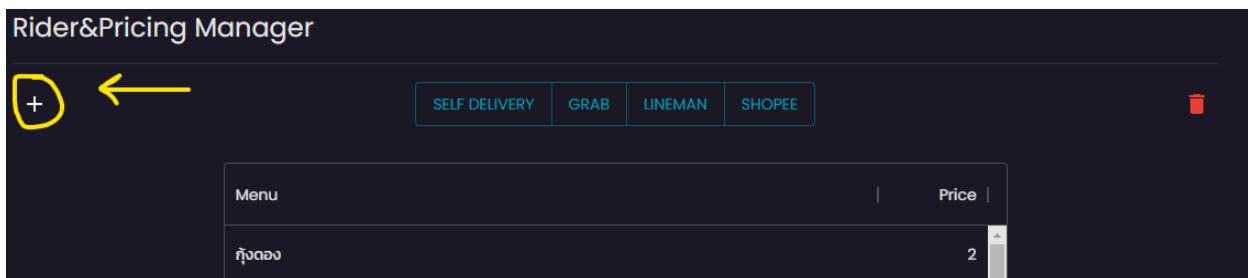
2.7.8 ผู้ใช้สามารถกดปุ่มถังขยะสีแดงทางด้านขวาเพื่อลบข้อมูลได้

2.8 ไปที่เมนู Rider&Pricing เพื่อตั้งค่าไรเดอร์



รูปประกอบ 4.21 ตำแหน่งเมนู Rider&Pricing

2.8.1 กดปุ่ม ทางซ้ายเพื่อเปิดเมนูสร้างไรเดอร์



รูปประกอบ 4.22 ตำแหน่งปุ่มสร้างแพลตฟอร์ม

2.8.2 กรอกข้อมูลดังนี้ เพื่อสร้างไรเดอร์

Name: ชื่อไรเดอร์

Fees: ค่าธรรมเนียม (%)

Tax: ภาษีค่าธรรมเนียม (%)

Gas Cost: ต้นทุนค่าน้ำมันสำหรับทุกเมนูในไรเดอร์นี้

2.8.3 กดปุ่ม ทางขวาของหน้าจอเพื่อบันทึกไรเดอร์ใหม่

2.8.4 ทำขั้นตอนได้ไรเดอร์ครบถ้วน

## 2.9 ตั้งค่าราคา

### 2.9.1 เลือกไกด์เดอร์ที่ต้องการแก้ไขราคา

Rider&Pricing Manager

Menu	Price
กุ้งดอง	90
เม่นดอง	130
กุ้งเม่นดอง	120

วันที่ประมวลผล 4.23 การเลือกไกด์เดอร์

### 2.9.2 ดับเบิลคลิกที่ราคาของเมนูที่ต้องการแก้ไข

Menu	Price
กุ้งดอง	90
เม่นดอง	130
กุ้งเม่นดอง	120
เม็ดกุ้ง(ชูชิ)	115
เม็ดเม่น(ชูชิ)	155

วันที่ประมวลผล 4.24 ทำใหม่นั่นในการแก้ไขราคา

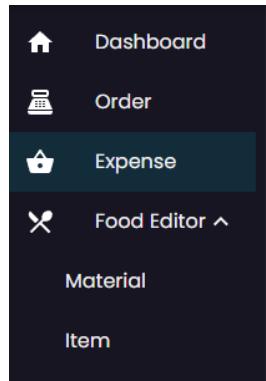
### 2.9.3 แก้ไขราคาเป็นราคาน้ำที่ต้องการแล้วกด Enter หรือคลิกที่พื้นที่ว่างข้าง

นอก

### 2.9.4 ทำซ้ำจนแก้ไขราคาที่ต้องการจนครบ

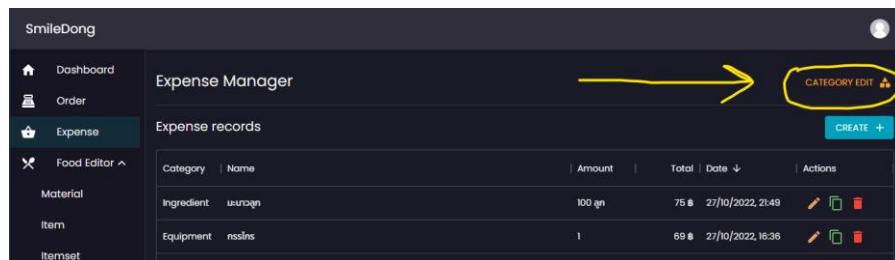
## 2.10 ตั้งค่าหมวดหมู่รายจ่าย

### 2.10.1 ไปที่เมนู Expense



รูปประกอบ 4.25 ตำแหน่งเม뉴 Expense

### 2.10.2 กดที่ Category Edit ทางด้านขวาของหน้าจอ



รูปประกอบ 4.26 ตำแหน่งปุ่มแก้ไขหมวดหมู่รายจ่าย

### 2.10.3 หากต้องการเพิ่มหมวดหมู่หลัก ให้ใส่ชื่อหมวดหมู่ที่ช่อง Name และกด



### 2.10.4 หากต้องการเพิ่มหมวดหมู่รอง

#### 2.10.4.1 กรอกชื่อที่ต้องการ

#### 2.10.4.2 เลือกหมวดหมู่หลักจากช่อง Parent

#### 2.10.4.3 กดบันทึก

### 2.10.5 ผู้ใช้สามารถแก้ไขหมวดหมู่ได้โดยการกดที่ชื่อหมวดหมู่ด้านล่าง



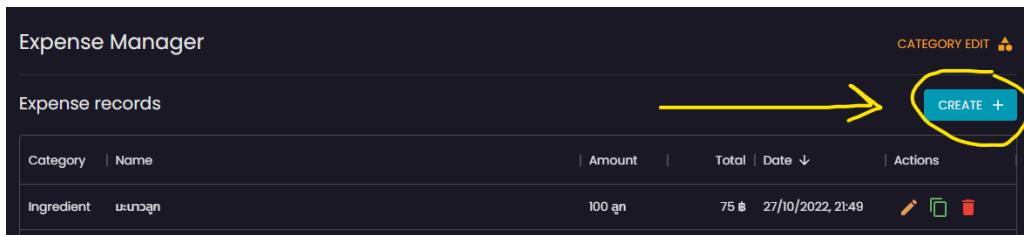
### 2.10.6 ผู้ใช้สามารถลบหมวดหมู่ได้โดยการกด ทางด้านขวา

### 2.11 เสริจสินขั้นตอนการตั้งค่าร้านค้าครั้งแรก

### 3. การจัดการรายจ่าย

#### 3.1 ไปที่เมนู Expense

3.2 กดปุ่ม Create ทางด้านขวาของหน้าจอเพื่อเพิ่มรายจ่าย



รูปประกอบ 4.27 ตำแหน่งปุ่มสร้างคำสั่งซื้อ

#### 3.2.1 เลือกหมวดหมู่ที่ซ่อง Category

3.2.1.1 กรณีที่หมวดหมู่ที่เลือกมีหมวดหมู่ร่อง สามารถเลือกหมวดหมู่

รองได้

#### 3.2.2 เลือก material ในกรณีที่เลือกหมวดหมู่เป็น Ingredient หรือ Packaging

3.2.3 กรอกชื่อหรือคำอธิบายเมื่อเลือกหมวดหมู่อื่น

3.2.4 กรอกจำนวนที่ซื้อ

3.2.5 กรอกยอดรวมทั้งหมด

3.2.6 สามารถเลือกวันเวลาที่ค่าใช้จ่ายเกิดขึ้นได้ หากปล่อยว่างจะใช้วันเวลาปัจจุบัน

3.2.7.1 กด Save & Add new expense เพื่อบันทึกและเตรียมสร้างรายการลดไป

3.2.7.2 กด Save & Go back เพื่อบันทึกและกลับไปยังหน้าตารางรายการ

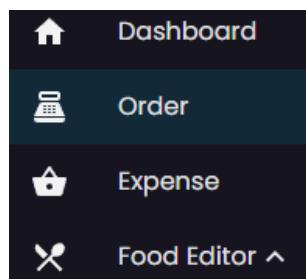
3.3 ผู้ใช้สามารถกดปุ่ม เพื่อแก้ไขรายการที่เลือกได้

3.4 ผู้ใช้สามารถกด  เพื่อสร้างรายการใหม่จากข้อมูลของรายการที่เลือก

3.5 ผู้ใช้สามารถกด  เพื่อลบรายการที่เลือกได้

#### 4. การจัดการคำสั่งซื้อ

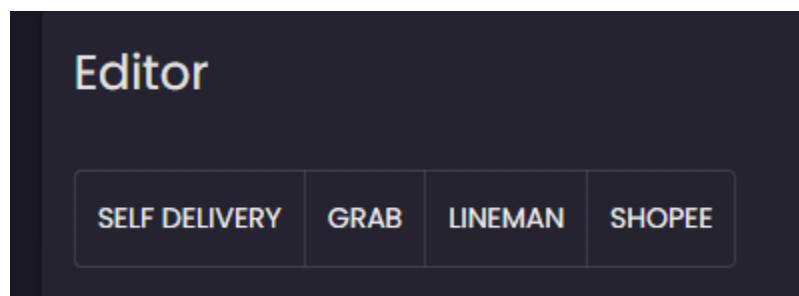
##### 4.1 กดที่ Order เมนูทางด้านซ้ายของโปรแกรม



รูปประกอบ 4.28 ตำแหน่งเมนู Order

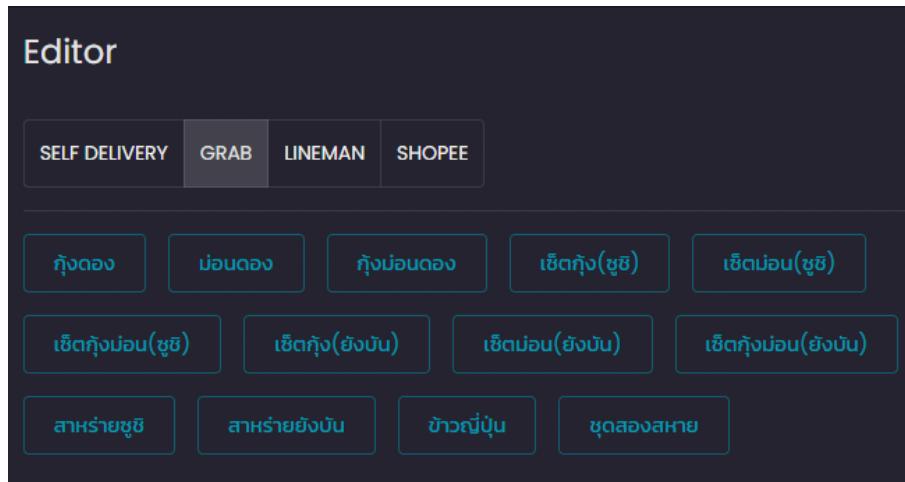
##### 4.2 สร้างคำสั่งซื้อใหม่

###### 4.2.1 เลือกไกด์เดอร์ที่ต้องการ



รูปประกอบ 4.29 ล้วนสำหรับเลือกไกด์เดอร์

###### 4.2.2 คลิกซ้ายที่เมนูที่ต้องการเพื่อเพิ่มเมนูเข้าสู่คำสั่งซื้อ



รูปประกอบ 4.30 ส่วนสำหรับเลือกเมนูอาหาร

4.2.3 ผู้ใช้สามารถกดที่เมนูซ้ำได้เพื่อเพิ่มจำนวนเมนูเดิมลงในคำสั่งชื้อ



4.2.4 ผู้ใช้สามารถใช้ปุ่ม เพื่อเพิ่มหรือลดจำนวนเมนูอาหารได้

4.2.5 ผู้ใช้สามารถเลือกวันเวลาที่คำสั่งซื้อเกิดหรือปล่อยว่างเพื่อใช้เวลาปัจจุบันได้

ได้

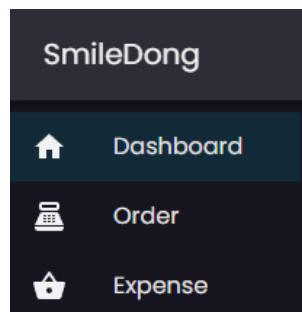
4.2.6 กดปุ่ม เพื่อบันทึกคำสั่งซื้อลงในระบบ



4.3 ผู้ใช้สามารถกดปุ่ม เพื่อลบคำสั่งซื้อที่ไม่ต้องการได้

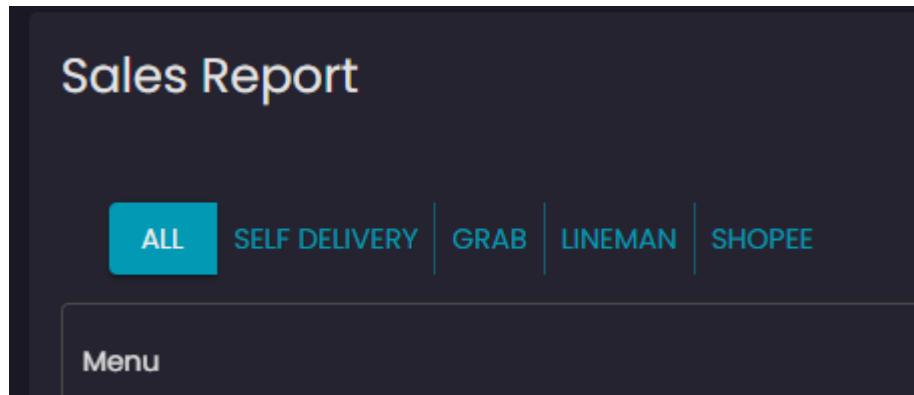
## 5. สถิติการขาย

### 5.1 ไปที่เมนู Dashboard



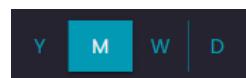
รูปประกอบ 4.31 ตำแหน่งเมนู Dashboard

5.2 ผู้ใช้สามารถเลือกไกด์เดอร์ที่ต้องการได้



รูปประกอบ 4.32 การเลือกไกด์เดอร์ใน Dashboard

5.3 ผู้ใช้สามารถเลือกขอบเขตระยะเวลาได้



รูปประกอบ 4.33 การเลือกขอบเขตเวลา

D = คำสั่งซื้อที่เกิดขึ้นใน 24 ชั่วโมงที่ผ่านมา

W = คำสั่งซื้อที่เกิดขึ้นใน 7 วันที่ผ่านมา

M = คำสั่งซื้อที่เกิดขึ้นใน 1 เดือนที่ผ่านมา

Y = คำสั่งซื้อที่เกิดขึ้นใน 1 ปีที่ผ่านมา

5.4 ผู้ใช้สามารถกดปุ่ม เพื่อเปิดหรือปิดโหมดแสดงสถิติผ่านกราฟเท่านั้น



รวมประกอบ 4.34 กราฟสถิติการขาย

## บทที่ 5

### บทสรุปและข้อเสนอแนะ (Conclusions and Recommendations)

#### 5.1. วัตถุประสงค์หรือประเด็นสำคัญในการพัฒนา

โครงการระบบจัดการร้านอาหารเดลิเวอรี่ ถูกพัฒนาขึ้นเพื่อช่วยเหลือร้านอาหารรูปแบบใหม่ ที่ได้รับความนิยมมากขึ้นในช่วงปี 2021 ที่ผ่านมา ร้านอาหารประเภทใหม่ที่กล่าวถึงนี้ คือร้านอาหารที่ไม่มีหน้าร้านไม่มีโต๊ะรองรับลูกค้า เพียงเปิดเมนูตามแพลตฟอร์มส่งอาหารออนไลน์ต่างๆ ร้านอาหารประเภทนี้จะพบรอบเจอบัญหาที่ว่า แพลตฟอร์มมีมากมายหลายบริษัท แต่ละบริษัทก็ต่างคิดค่าบริการ ภาษี ค่าสมาชิก แตกต่างกันไป ทำให้ต้นทุนอาหารที่ผลิต ราคาอาหารที่ขาย และค่าใช้จ่ายอื่นๆ มีความซับซ้อนมากกว่าร้านอาหารทั่วไปที่ไม่มีบริการส่งอาหารออนไลน์ หรือใช้แพลตฟอร์มส่งอาหารแค่แพลตฟอร์มเดียว

ในตัวอย่างร้านอาหาร ‘สมายดอง’ ที่ผู้จัดทำได้เลือกใช้ในการวิจัยข้อมูล ทางร้านไม่มีระบบ POS เนื่องจากระบบ POS ทั่วไปไม่รองรับการที่เมนูหนึ่งเมนูมีราคา 4 – 5 ราคา ตามแต่แพลตฟอร์มส่งอาหารที่เลือกใช้ทำให้ทางร้านต้องใช้ระบบการบันทึกคำสั่งซื้อที่ยุ่งยาก วุ่นวาย ไม่สะดวกในการใช้งาน

นอกจากนี้ ทางร้านเองยังประสบปัญหาในการบันทึกข้อมูลคำสั่งซื้อในระยะยาว และจัดทำเป็นสถิติการขาย เพื่อปรับปรุงร้านและจัดทำโปรโมชั่นต่างๆ โดยทั่วไปแล้ว แต่ละแพลตฟอร์มก็มีการเก็บสถิติของตัวเอง แต่ธุรกิจการเก็บเหล่านี้ก็ไม่เป็นมาตรฐานเดียวกัน การค้นหาข้อมูลก็ยากลำบาก ทำให้การรวบรวมข้อมูลการขายมาไว้ ณ ที่เดียวกันไม่ใช่เรื่องง่าย

อีกปัญหานึงที่ทางร้านพบเจอก็คือ ระบบที่ใช้บันทึกข้อมูลการขายอยู่ส่วนหนึ่ง ระบบที่ใช้บันทึกรายจ่ายภายในร้านก็อยู่อีกส่วนหนึ่ง เมื่อทางร้านต้องการจะวิเคราะห์ข้อมูลเพื่อนำไปใช้ในการจัดการบัญชีหรือ

อื่นๆ ก็ต้องค่อยๆ เก็บข้อมูลทีละระบบ ก่อนที่จะนำมาร่วมกันแล้ววิเคราะห์ออกมาได้ และระบบเหล่านี้ก็ยังมีค่าใช้จ่ายหรือค่าสมาชิกรายเดือนแยกกันไปต่างหาก

ส่วนของระบบบันทึกรายจ่ายแต่เดิมของร้านนั้น ทางร้านเองค่อนข้างพอดีกับระบบเดิม แต่หลังจากการวิเคราะห์การทำงานแล้วนั้น ผู้จัดทำเล็งเห็นว่า ระบบเดิมยังไม่มีการจัดเก็บข้อมูลที่เป็นมาตรฐานมากพอ ทำให้การค้นหาข้อมูลรายจ่ายมีความยากลำบากอยู่บ้าง และบางรายจ่ายที่เกิดขึ้นซ้ำๆ ก็ยังคงต้องกรอกข้อมูลใหม่ตั้งแต่ต้นทุกวันเพื่อบันทึกรายจ่ายนั้น

## 5.2. ผลการพัฒนาสามารถแก้ปัญหาได้มากน้อยเพียงใด

โปรแกรมจัดการร้านอาหารเดลิเวอรี่ที่พัฒนาขึ้นมานั้น ได้รวมระบบทุกอย่างที่ทางร้านต้องการไว้ ทำให้สามารถทำงานได้อย่างครบวงจรภายในที่เดียว ไม่ว่าจะเป็นการจัดเก็บข้อมูลสูตรอาหาร การบันทึกรายจ่าย การบันทึกคำสั่งชิ้อ และการทำสถิติการขาย

ระบบบันทึกรายจ่าย ถูกสร้างขึ้นเพื่อคงไว้ซึ่งทุกอย่างที่ผู้ใช้ชื่นชอบในระบบเก่า และเพิ่มเติมคำสั่งใหม่ๆ ที่จะช่วยให้การทำงานง่ายขึ้น เช่น ปุ่ม Repeat ที่ปรากฏที่รายจ่ายเก่าๆ ที่เมื่อกดไปแล้ว ระบบจะเพิ่มรายการใหม่ในเวลาปัจจุบัน ด้วยข้อมูลของรายจ่ายที่เลือก หรือ การเลือกวัตถุดิบจากฐานข้อมูลโดยตรงแทนการพิมพ์ชื่อสินค้าที่ชื่อ ทำให้การเก็บข้อมูลมีมาตรฐาน ค้นหาได้ง่าย แต่ยังคงระบบพิมพ์ชื่อรายจ่ายเอาไว้ เมื่อเลือกหมวดหมู่อื่นๆ ที่ไม่ใช่วัตถุดิบ ทำให้ระบบยังมีความยืดหยุ่นอยู่เช่นเดิม ทำให้ปัญหาในการค้นหารายจ่าย และความยุ่งยากในการทำรายการจ่ายหมดไป

ระบบบันทึกคำสั่งชิ้อ ถูกออกแบบแบบสำหรับการใช้งานที่สะดวก รวดเร็ว และเข้าถึงได้ง่าย ในส่วนระบบบันทึกคำสั่งชิ้อนั้น ได้ออกแบบมาให้ responsive เป็นพิเศษ ทำให้สามารถใช้งานได้ง่ายทั้งใน Desktop PC หรือ Mobile Device นอกจากนี้ ในส่วนของการบันทึกข้อมูลนั้น ผู้ใช้สามารถเพิ่มข้อมูลได้เพียงการกดเลือกเท่านั้น ไม่จำเป็นต้องพิมพ์ค่าข้อมูลใดๆ ส่วนราคา ยอดรวม และรายละเอียดอื่นๆ นั้น ระบบจะคำนวณให้โดยอัตโนมัติ จากการที่ผู้ใช้ได้กำหนดราคาต่างๆ ไว้ล่วงหน้า ตามแพลตฟอร์มส่งอาหารที่เลือกใช้ ทำให้ปัญหาในการบันทึกคำสั่งชิ้อ และความยุ่งยากในการเข้าถึงระบบบันทึกคำสั่งชิ้อหมดไป

ระบบยังสามารถแสดงสถิติการขายได้ในทุกที่ทุกเวลา ทั้งในรูปแบบตาราง และรูปแบบกราฟ โดยผู้ใช้สามารถเลือกได้ว่าต้องการจะดูสถิติการขายจากแพลตฟอร์มใดบ้าง และยังเลือกช่วงเวลาที่ต้องการได้อีกด้วย ทำให้จากเดิมที่การรวบรวมข้อมูลการขายต่างๆ ที่เป็นไปเมื่อต่อ หรือเป็นไปได้ยาก สามารถทำได้ง่ายๆ เพียงกดเลือกสิ่งที่ต้องการ

### 5.3. สิ่งที่คาดหวังที่ระบบจะสามารถช่วยในการทำงานได้มากน้อยเพียงใด

ผู้จัดทำคาดหวังว่าระบบปัจจุบันสามารถช่วยให้ทางร้านสามารถจัดการร้านได้อย่างสะดวก รวดเร็ว และถูกต้องมากยิ่งขึ้น เนื่องด้วยเหตุผลดังนี้

1. ระบบบันทึกสูตรอาหารที่ละเอียดแม่นยำ จะช่วยให้ทางร้านสามารถทำอาหารในปริมาณที่ต้องการได้ง่ายมากขึ้นกว่าระบบถ่ายทอดมาก
2. ระบบบันทึกรายจ่ายที่มีมาตรฐานและสามารถทำซ้ำรายจ่ายได้อย่างง่ายดาย จะช่วยให้เวลาที่ใช้ในการลงบัญชีรายจ่ายน้อยลง และการค้นหาข้อมูลสินค้าย้อนหลังทำได้ง่ายและแม่นยำขึ้น
3. ระบบบันทึกคำสั่งซื้อที่ใช้ได้ง่าย และเข้าถึงได้จากโทรศัพท์มือถือ จะช่วยให้เวลาที่ใช้ในการบันทึกคำสั่งซื้อน้อยลงมากๆ และถูกต้องมากขึ้น เมื่อเทียบกับระบบเก่าที่ต้องกดหาข้อมูลจากในตารางและบันทึกเองด้วยมือ นอกจากนี้ทางร้านจะมีประวัติการขายให้ดูย้อนหลังได้อีกด้วย
4. ระบบแสดงสถิติการขาย จะช่วยให้ทางร้านสามารถตรวจสอบยอดขายได้ตามต้องการ ทั้งการแปลงสถิติตามแพลตฟอร์ม หรือกระทั่งการเลือกช่วงระยะเวลาที่ต้องการ จากเดิมที่การตรวจสอบเพื่อทำสถิติในระยะเวลาเดือนทำได้ยาก และในระยะหกเดือนขึ้นไปไม่สามารถตรวจสอบได้เลย

### 5.4. ปัญหาและข้อเสนอแนะ

ปัญหาที่พบเจอสามารถแบ่งออกได้เป็น 2 ส่วนคือ ปัญหาทางกระบวนการทำงานของทางร้าน และปัญหาทางเทคนิคในการพัฒนาโปรแกรม

#### ปัญหาทางกระบวนการทำงาน

1. Flow การทำงานของระบบเดิมมีความซับซ้อนและไม่ตรงกับหลักในการพัฒนาโปรแกรม โดยทั่วไป และทางร้านเองก็ยืนยันว่าไม่ต้องการจะเป็นระบบการทำงานตรงนี้ เนื่องจากความเคยชินและความเชี่ยวชาญของทั้งพนักงานและเจ้าของร้านเอง ปัญหานี้ได้รับการแก้ไขโดยการออกแบบโครงสร้างโปรแกรมอย่างละเอียด เพื่อให้ตรงกับความต้องการของผู้ใช้ และความเข้าใจของผู้จัดทำ

2. ทางร้านไม่มีการตัดตันทุนค่าแรงเป็นรายเดือน แต่ใช้วิธีกำหนดให้แต่ละเมนูอาหาร มีต้นทุนค่าแรงที่แตกต่างกันออกไป ปัญหานี้ได้รับการแก้ไขโดย การสร้าง field ใหม่ในฐานข้อมูลเมนูอาหาร เพื่อเก็บต้นทุนค่าแรงในแต่ละเมนู
3. ทางร้านไม่มีการตัดตันทุนค่าน้ำมันตามการใช้งานจริง แต่ใช้วิธีกำหนดให้แต่ละเมนูอาหาร มีต้นทุนค่าน้ำมันที่แตกต่างกันออกไป ปัญหานี้ได้รับการแก้ไขโดย การสร้าง field ใหม่ในฐานข้อมูลเมนูอาหาร เพื่อเก็บต้นทุนค่าน้ำมันในแต่ละเมนู
4. ทางร้านต้องการให้บางแพลตฟอร์ม ไม่ใช้ต้นทุนค่าน้ำมันจากเมนูอาหาร แต่ทุกเมนูจะใช้ต้นทุนค่าน้ำมันเท่ากัน เพียงแต่ แต่ละแพลตฟอร์มก็จะมีต้นทุนค่าน้ำมันตรงนี้ที่แตกต่างกันออกไป ปัญหานี้ได้รับการแก้ไขโดย การที่กำหนด field ใหม่ในฐานข้อมูลแพลตฟอร์มส่งอาหาร แล้วให้ระบบคำนวณต้นทุนจากตรงนี้ก่อน หากแพลตฟอร์มไม่มีค่านี้ จึงจะใช้ค่าต้นทุนน้ำมันจากเมนูอาหารนั้นๆ

### ปัญหาทางเทคนิค

1. ผู้จัดทำต้องการจะใช้ MERN Stack ในการพัฒนาโปรแกรม เพื่อเรียนรู้ JavaScript ในเชิงลึก และเรียนรู้เทคโนโลยีใหม่ๆที่ไม่เคยได้ลองใช้ ทั้ง ฐานข้อมูลแบบ NoSQL, การทำงานกับ JavaScript Library, การใช้ Framework ในการเขียน REST API และอื่นๆอีกมากมาย ทำให้คุณภาพของโค้ดภาษาในโปรแกรมนั้นมีความไม่เท่ากัน โดยที่ผลงานหรือ components ในช่วงแรกจะมีคุณภาพต่ำ และผลงานต่อๆไปก็มีคุณภาพสูงขึ้นเรื่อยๆ ตามการเรียนรู้ของผู้จัดทำ ปัญหานี้ได้รับการแก้ไขโดย การกลับมาเริ่มต้น (refactor) หรือกระทำการรีวอร์ค (rework) โค้ดเก่าๆใหม่ทั้งหมด ทำให้คุณภาพโค้ดมีความใกล้เคียงกัน
2. หมวดหมู่ค่าใช้จ่ายมีการเก็บข้อมูลแบบ Self one to many กล่าวคือ ในข้อมูลหนึ่งอาจมี ถึงข้อมูลอื่นๆใน collection เดียวกัน เป็นปัญหาในส่วนของการทำ Sub-category เมื่อผู้ใช้ต้องการลบหมวดหมู่ที่เป็น Sub-category จะมีความยุ่งยากมากกว่าปกติ หรือเมื่อผู้ใช้ต้องการเปลี่ยน Parent ของ Sub-category การยกย้าย Embedded documents นั้น ไม่ใช่เรื่องง่าย ปัญหานี้ได้รับการแก้ไขโดย การปรับเปลี่ยนแนวคิดและ Algorithm ใหม่ ทั้งหมด แทนที่จะใช้ Embedded documents เพื่อเก็บข้อมูล Sub-category ก็ให้ทุก category อยู่ใน level ที่เท่ากันทั้งหมด และเฉพาะ category ที่ไม่มี parent เท่านั้น ระบบจะจัดว่าเป็น Main category

3. Form ในการเก็บส่วนประกอบของอาหารหรือเมนูอาหารนั้นมีความ dynamic ที่สูงมาก ผู้ใช้สามารถเพิ่มและลดจำนวน field ได้ตามชอบ ทำให้การทำ form component ทำได้ยาก และการส่งต่อข้อมูลก็มีความซับซ้อน เนื่องจากที่ผ่านมา แต่ละ input ใน form จะใช้ state แยกกัน ปัญหานี้ได้รับการแก้ไขโดย การกำหนด state ให้เป็น Array และส่งต่อข้อมูลด้วยค่า index ของ input ที่ loop ขึ้นมา
4. การกำหนดราคาของแต่ละเมนูมีความคล่องตัวต่ำ ทำให้ใช้เวลาในการกำหนดราคาของทุกเมนูในทุกๆแพลตฟอร์ม ปัญหานี้ได้รับการแก้ไขโดย การเปลี่ยนจากการกรอกข้อมูลในรูปแบบเดิมมาเป็นการใช้ DataGridView ของ MaterialUI และใช้ feature ที่ชื่อว่า cell editing
5. ในการที่ React จะทำการ re-render component แต่ละครั้งนั้น React จะเปรียบเทียบ state แต่ละ state ว่ามีการเปลี่ยนอะไรหรือไม่ หากไม่เปลี่ยนก็จะไม่มีการ re-render เกิดขึ้น ซึ่งใน JavaScript การเปรียบเทียบค่าระหว่าง Object กับ Object หรือ Array กับ Array จะเปรียบเทียบกันด้วยค่า reference กล่าวคือ จะไม่มีการเปรียบเทียบค่าภายในของ Object เรียกว่า Shallow compare ปัญหาที่เกิดขึ้นคือ เมื่อมีการเปลี่ยนแปลงข้อมูลภายใน state แบบ Object หรือ Array ไม่ว่าค่าภายในจะแตกต่างจากเดิมเท่าไร React ก็จะไม่ทำการ re-render components ปัญหานี้ได้รับการแก้ไขโดย ทุกๆการแก้ไข state ที่เป็น Object หรือ Array จะต้องเป็นการสร้างใหม่ทุกครั้งโดยการใช้ spread operator ช่วยแทนที่จะกำหนดค่าเข้าไปในตัวแปรโดยตรง
6. หลังจากผู้จัดทำได้เรียนรู้และทำความเข้าใจ React มากขึ้น ผู้จัดทำจึงได้ข้อสรุปว่า การพัฒนาโปรแกรมในลักษณะนี้ ไม่มีความจำเป็นต้องต้องใช้ Redux state management ปัญหานี้ไม่ได้รับการแก้ไข เนื่องจากหากต้องการนำ Redux ออก จะต้องพัฒนาโปรแกรมใหม่ทั้งหมดตั้งแต่ต้น และทางผู้จัดทำก็ไม่เห็นลึกลงผลกระทบในภาพรวม จึงไม่มีความจำเป็นต้องเริ่มพัฒนาใหม่ทั้งหมดเพื่อแก้ปัญหานี้ แต่หากต้องการจะแก้ไข สิ่งที่ต้องทำคือ เอาระบบ Redux ออกทั้งหมด แล้วทดแทนด้วยการให้ component แม่ fetch ข้อมูลจาก API และใช้ Context hook provide ข้อมูลลงไปที่ component ลูก

## บรรณานุกรม

Jirayut Intachai. (2564). **HTML คืออะไร? ทำไมคนเขียนเว็บไซต์ต้องใช้งาน.** จาก Terrestrial: <https://goterrestrial.com/2021/05/19/what-is-html/>

Kipakaporn. (2561). **CSS คืออะไร มีประโยชน์อย่างไร.** จาก SoGoodWeb: <https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์อย่างไร>

Worapong. (2563). **JavaScript คืออะไร ใช้งานอย่างไร.** จาก WebDoDee: <https://www.webdodee.com/what-is-javascript/>

Framework คืออะไร โครงสร้างหรือขอบเขตของระบบงาน. (2561). จาก MindPHP: <https://www.mindphp.com/developer/25-oop-php-framework/1957-framework-คืออะไร.html>

MVC คืออะไร ทำความสะอาดเข้าใจรูปแบบการเขียนโปรแกรม. (2559). จาก บริษัท โค๊ดบี จำกัด: <https://www.codebee.co.th/labs/mvc-คืออะไร-ทำความสะอาดเข้าใจรูป/>

Wim Hoogenraad. (2562). **ฐานข้อมูล NoSQL คืออะไร.** จาก ITpedia: <https://th.itpedia.nl/2019/06/03/wat-zijn-nosql-databases/>

Chai Phonbopit. (2558). **MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น.** จาก Devahoy: <https://www.devahoy.com/blog/2015/08/getting-started-with-mongodb>

Express.js เอ็กเพรส ดอทเจอส คืออะไร Express.js คือ เป็น Web Application Framework ซึ่งดังที่ได้รับความนิยมมาก สำหรับทำงานบน platform ของ Node.js. จาก MindPHP: <https://www.mindphp.com/คู่มือ/73-คืออะไร/3874-what-is-express-js.html>

Mos Programmer. (2562). **มาเร็จก React กันเถอะ.** จาก Medium: <https://mosprogrammer.medium.com/มาเร็จก-react-กันเถอะ-eb04ab8b8ec1>

**ทำความสะอาดเข้าใจรูปแบบการเขียนโปรแกรม** (2564). จาก MarcusCode: <http://marcuscode.com/tutorials/nodejs/introducing-nodejs>

ปริญญา อภัยภักดี และณัฐพล เสาวพงษ์. ระบบจัดการร้านอาหาร. (ภาคนิพนธ์ปริญญาตรี คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยสยาม, 2562)

วิกรม ศรศรี และนำโชค ทองلامมูล. ระบบจัดการการสั่งอาหาร กรณีศึกษา ร้านอาหาร Zero. (ปริญญาตรี คณะบริหารธุรกิจ มหาวิทยาลัยราชพฤกษ์, 2560)

ธีรพงศ์ ชูชื่น และจิรวัช ศรีโชค. ระบบจัดการร้านอาหาร. (ปริญญาตรี คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสำเภาเจ้าพระยา, 2555)

ปรินทร อ่อนโภค, นิรัชชา บุญวิเทียน และ ชัญญาณุช ศรีวรรณ. ระบบจัดการร้านกังฟูหม่าล่า เปียร์ทิมະ. (ปริญญาตรี คณะวิทยาการจัดการ มหาวิทยาลัยราชภัฏอุดรธานี, 2563)

วิรุฬห์ ศรีบริรักษ์. เทคโนโลยีเกสช์สารสนเทศแสดงข้อมูลถูกต้องตามเอกสารกำกับยาแบบ อัตโนมัติสำหรับบริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย (คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา, 2562)

## ประวัติย่อผู้พัฒนา

ชื่อ – สกุล : ชิน วิสุทธิ์มรรคกุล

รหัสนักศึกษา : 6101240160

หมายเลขโทรศัพท์ที่ติดต่อได้สะดวก : 083-622-2054

Email address : nickethachin@gmil.com

ส่วนโปรแกรมที่ได้พัฒนาในโครงการนี้

- Database models
- Controllers
- REST API, Misc. API
- Backend routing
- Authentication & Middleware
- Dynamic navigating menu
- Register module
- Login module
- Dark mode theme
- Light & Dark mode switch
- Materials manager module
- Items manager module
- Itemsets manager module
- Expense category manager module
- Expenses manager module
- Orders manager module
- Dashboard module
- Frontend routing
- Redux state management

## ตัวอย่างเอกสาร

M ID	M Name	Today
Shrimp	กุ้งดอง (129.-)	
setSN	กุ้งดองชูชี (165.-)	1
setSYB	กุ้งดองย่างมัน (180.-)	
Salmon	ม่อนดอง (199.-)	
setSalN	ม่อนดองชูชี (230.-)	
setSalYB	ม่อนดองย่างมัน (240.-)	
SLS	กุ้งเผา (169.-)	
setSLS-N	กุ้งเผาชูชี (200.-)	1
setSLS-YB	กุ้งเผาย่างมัน (205.-)	2
PWater	น้ำดื่ม ชาด (49.-)	
SeaSause	น้ำจิ่น ชาด (49.-)	
swSS	สาหร่ายชูชี (20.-)	1
riceJP	ข้าวผัดปีปุย (25.-)	
swYB	สาหร่ายย่างมัน (35.-)	
SpiceGarlic	พริกกระเทียม (15.-)	
ThaiTea	ชาไทย (45.-)	
<b>Total</b>		
Grab		
M ID	M Name	Amount
Shrimp	กุ้งดอง (129.-)	2
setSN	กุ้งดองชูชี (165.-)	1
setSYB	กุ้งดองย่างมัน (180.-)	
Salmon	ม่อนดอง (199.-)	
setSalN	ม่อนดองชูชี (230.-)	
setSalYB	ม่อนดองย่างมัน (240.-)	
SLS	กุ้งเผา (169.-)	3
setSLS-N	กุ้งเผาชูชี (200.-)	
setSLS-YB	กุ้งเผาย่างมัน (205.-)	
swSS	สาหร่ายชูชี (20.-)	1
riceJP	ข้าวผัดปีปุย (25.-)	
swYB	สาหร่ายย่างมัน (35.-)	1
Wasabi	ราชาเมี (5.-)	2
<b>Total</b>		

ตัวอย่างระบบบันทึกคำสั่งซื้อแบบเก่าของทางร้าน

**X Add transaction** SAVE

---

฿ 0

---

 Materials

---

 Write note

---

 Today

---

 SmileDong

---

**ADD MORE DETAILS** ▾

ตัวอย่างระบบบันทึกรายจ่ายแบบเก่า