



## โครงการวิทยาการคอมพิวเตอร์

ระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร

(Restaurant's Food Ordering and Payment System Web Application)

ผู้จัดทำ

นายจิตติณณ์ จินดานรเศรษฐ์ รหัสประจำตัวนิสิต 6110405949

นางสาวณัฐนิชา คงสุนทร รหัสประจำตัวนิสิต 6110402753

อาจารย์ที่ปรึกษาโครงการ

ผศ.ดร.สุขุมala กิตติสิน

ผศ.ดร.ชาลิต ครีสต้าพรพัฒน์

ดร.เสนาธิรักษ์ เกิดผล

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเกษตรศาสตร์ ปีการศึกษา 2564

## โครงการวิทยาการคอมพิวเตอร์

หัวข้อ ระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร

(Restaurant's Food Ordering and Payment System Web Application)

โดย นายจิตติณ พิจิตรา จินดาธนเศรษฐี รหัสประจำตัวนิสิต 6110405949

นางสาวณัฐณิชา คงสุนทร รหัสประจำตัวนิสิต 6110402753

อาจารย์ที่ปรึกษา \_\_\_\_\_

(ผศ.ดร.สุขุมมาล กิตติสิน)

\_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

อาจารย์ที่ปรึกษาร่วม \_\_\_\_\_

(ผศ.ดร.ชวัลิต ศรีสถาพรพัฒน์)

\_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

อาจารย์ที่ปรึกษาร่วม \_\_\_\_\_

(ดร.เสนาณิช ภิเศกผล)

\_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์ \_\_\_\_\_

(ผศ.ดร.พกาเกษา วงศ์ตุ้ย)

\_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

## บทคัดย่อ

หัวข้อ ระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร

(Restaurant's Food Ordering and Payment System Web Application)

โดย นายจิตติมน์ จินดานรเศรษฐ์ รหัสประจำตัวนิสิต 6110405949

นางสาวณัฐณิชา คงสุนทร รหัสประจำตัวนิสิต 6110402753

คำสำคัญ ร้านอาหาร, สั่งอาหาร, ชำระเงิน, บริการตนเอง

อาจารย์ที่ปรึกษา \_\_\_\_\_

(ผศ.ดร.สุขุมาล กิตติสิน)

ระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร คือ ระบบที่สามารถให้ลูกค้า ที่มาใช้บริการร้านอาหารสามารถทำการสแกน QR Code ของทางร้านอาหาร ผ่านโทรศัพท์มือถือหรือแท็บเล็ตของลูกค้า เพื่อดูเมนูอาหารของร้านอาหาร และสามารถสั่งรายการอาหาร พร้อมระบุตัวเลือกของอาหาร และหมายเหตุที่ต้องการได้ด้วยตนเองบนเว็บแอปพลิเคชัน โดยหลังจากการรับประทานอาหารเสร็จ สามารถตรวจสอบรายการอาหารที่สั่ง และชำระเงินได้ด้วยตนเองผ่านระบบแอปพลิเคชันธนาคารที่รองรับ โดยระบบนี้จัดทำขึ้นเพื่อเพิ่มประสิทธิภาพความพึงพอใจของลูกค้า โดยลดเวลาในการรอรับบริการจากพนักงาน ลดการสัมผัสมенูอาหารและเงินสด เพิ่มประสิทธิภาพการให้บริการของพนักงานให้รวดเร็วยิ่งขึ้นใน การให้บริการลูกค้า และร้านค้าสามารถปรับเปลี่ยนข้อมูลของเมนูได้ตามที่ต้องการ

## Abstract

Subject Restaurant's Food Ordering and Payment System Web Application

By Mr. Jittin Jindanoraseth ID. 6110405949

Miss Natnicha Khongsoontorn ID. 6110402753

Keyword Restaurant, Food Ordering, Payment, Self-Service

Project Advisor \_\_\_\_\_

(Asst. Prof. Sukumal Kitisin)

Ordering and Payment System Web Application for Restaurants is a system that enables customers to access the menu by scanning a restaurant's QR code with their smartphone or tablet, then order from available options and specify their preferences all within the web application. After the meal, customers can check their order and pay via a supported mobile banking system themselves. This system is designed to improve customer satisfaction by reducing waiting time to get the service from a server, reducing contact with menu books and cash, as well as improving speed of service. Restaurants can edit menu details at any time.

## คำนำ

เอกสารนี้เป็นเอกสารประกอบโครงการระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร จัดทำขึ้นเพื่ออธิบายที่มา การออกแบบต่าง ๆ รวมถึงขั้นตอนการดำเนินงาน การพัฒนาและการทดสอบเพื่อเป็นแนวทางในการศึกษาต่อไป

สุดท้ายนี้ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่าเอกสารฉบับนี้จะเป็นประโยชน์ต่อผู้ที่สนใจในเว็บแอปพลิเคชันเกี่ยวกับการจัดการในร้านอาหาร และสามารถนำไปพัฒนาต่อยอดได้เป็นอย่างดี หากมีข้อผิดพลาดประการใด คณะผู้จัดทำขออภัยไว้ ณ ที่นี่ด้วย

คณะผู้จัดทำ

นายจิตติณณ์ จินดานรเศรษฐ์ รหัสประจำตัวนิสิต 6110405949

นางสาวณัฐณิชา คงสุนทร รหัสประจำตัวนิสิต 6110402753

## กิตติกรรมประกาศ

โครงการเรื่องระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหารสำเร็จลุล่วงไปได้ด้วยดีเนื่องด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงการทั้งสามท่าน ผศ.ดร.สุขุมมาล กิตติสิน,  
ผศ.ดร.ชวโลท ศรีสถาพรพัฒน์ และดร.เสฎฐวิทย์ เกิดผล ที่ได้ให้คำแนะนำ ข้อปรับปรุงต่าง ๆ ตลอดจน  
ตรวจสอบการทำโครงการสมำเสมอ

คณะกรรมการขอขอบคุณท่านรวมถึงท่านที่ไม่ได้กล่าวนาม ณ ที่นี่ที่มีส่วนร่วมในการมอบความรู้เพื่อ  
พัฒนาทักษะความสามารถในอนาคตต่อไป

คณะกรรมการ

นายจิตติณณ์ จินดานรเศรษฐ์ รหัสประจำตัวนิสิต 6110405949

นางสาวณัฐณิชา คงสุนทร รหัสประจำตัวนิสิต 6110402753

## สารบัญ

### เนื้อหา

### หน้า

บทคัดย่อ.....	๗
Abstract .....	๑
คำนำ .....	๒
กิตติกรรมประกาศ .....	๓
สารบัญ .....	๘
<b>บทที่ ๑ บทนำ.....</b>	<b>๑</b>
1.1. ที่มาและความสำคัญของโครงการ .....	๑
1.2. วัตถุประสงค์.....	๒
1.3. ขอบเขตของโครงการ .....	๒
1.4. เครื่องมือที่ใช้ในการพัฒนา .....	๓
1.4.1. ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ.....	๓
1.4.2. ภาษาที่ใช้ในการพัฒนาระบบ.....	๓
1.4.3. ไลบรารีและเฟรมเวิร์คที่ใช้ในการพัฒนาระบบ .....	๓
1.5. ประโยชน์ที่คาดว่าจะได้รับ .....	๓
1.6. ระยะเวลาในการดำเนินโครงการ .....	๔
<b>บทที่ ๒ ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง .....</b>	<b>๕</b>
2.1. Spring Boot.....	๕
2.2. Spring MVC .....	๖
2.3. API.....	๖
2.4. MySQL .....	๖
2.5. Bootstrap.....	๗
2.6. jQuery.....	๗

2.7. IntelliJ IDEA .....	7
2.8. Postman.....	8
2.9. SCB Developers.....	8
2.9.1. Authentication .....	9
2.9.2. SCB EASY App Payment.....	11
2.9.3. QR Code Payment .....	13
<b>บทที่ 3      การวิเคราะห์และการออกแบบระบบ .....</b>	<b>16</b>
3.1. การวิเคราะห์และการออกแบบความต้องการของระบบ.....	16
3.1.1. ระบบการแสดงผลเมนูและการสั่งอาหาร .....	16
3.1.2. ระบบตรวจสอบรายการอาหารที่สั่ง .....	16
3.1.3. ระบบชำระเงิน.....	16
3.1.4. ระบบจัดการรายการอาหารที่สั่ง .....	17
3.1.5. ระบบจัดการเมนูอาหาร .....	17
3.1.6. หน้าแสดงใบเสร็จ .....	17
3.2. Activity Diagram.....	18
3.3. Use Case Diagram .....	21
3.4. Use Case Description .....	22
3.5. การออกแบบฐานข้อมูล .....	27
<b>บทที่ 4      การพัฒนาระบบ.....</b>	<b>32</b>
4.1. ส่วนของลูกค้า (USER) .....	32
4.1.1. หน้าแรกของเว็บไซต์ .....	32
4.1.2. หน้าเมนูสำหรับเลือกอาหารลงตะกร้า.....	34
4.1.3. การสั่งอาหารที่เลือกไว้ในตะกร้าสำหรับลูกค้ารับประทานที่ร้าน (DINE-IN).....	38
4.1.4. การสั่งอาหารที่เลือกไว้ในตะกร้าสำหรับลูกค้ารับกลับบ้าน (TAKE-OUT).....	41

4.1.5.	การชำระเงิน .....	43
4.1.6.	หน้าแสดงใบเสร็จ .....	50
4.2.	ส่วนของพนักงาน (STAFF).....	53
4.2.1.	หน้าสำหรับจัดการลูกค้า .....	53
4.2.2.	หน้าสำหรับจัดการครัว .....	59
4.3.	ส่วนของผู้ดูแล (ADMIN) .....	61
4.3.1.	สามารถจัดการทุกหน้าของพนักงานได้ .....	61
4.3.2.	หน้าสำหรับจัดการเมนูในร้าน .....	61
<b>บทที่ 5</b>	<b>การทดสอบระบบ.....</b>	<b>69</b>
5.1.	การทดสอบ API .....	70
5.1.1.	ตัวอย่างการทดสอบใช้งาน API /api/order/new สำหรับการสั่งอาหาร .....	71
5.2.	ส่วนของลูกค้า (USER) .....	74
5.2.1.	ทดสอบหน้าเลือกเมนู .....	74
5.2.2.	ทดสอบการเพิ่มลงตะกร้า .....	74
5.2.3.	ทดสอบการสั่งรายการอาหาร .....	75
5.2.4.	ทดสอบการดูบิลและสถานะสำหรับ DINE-IN.....	75
5.2.5.	ทดสอบการไปยังหน้าทางเลือกชำระเงิน .....	76
5.2.6.	การชำระเงินแบบ QR Code .....	76
5.2.7.	SCB Easy Sandbox.....	76
5.2.8.	ทดสอบการไปยังหน้าใบเสร็จ.....	76
5.3.	ส่วนของพนักงาน (STAFF).....	77
5.3.1.	ทดสอบการเปิดໂທให้บริการลูกค้า.....	77
5.3.2.	ทดสอบการดูบิลลูกค้า .....	78
5.3.3.	ทดสอบการปรับสถานะรายการอาหาร .....	80

5.3.4.	ทดสอบการปรับสถานะรายการอาหารในครัว .....	80
5.4.	ส่วนของผู้ดูแล (ADMIN) .....	81
5.4.1.	ทดสอบการเพิ่มเมนูอาหาร .....	81
5.4.2.	ทดสอบการเพิ่มกลุ่มตัวเลือก.....	82
5.4.3.	ทดสอบการเพิ่มตัวเลือก.....	83
บทที่ 6	บทสรุป .....	84
6.1.	สรุปผลการดำเนินงาน .....	84
6.2.	ปัญหาและอุปสรรค .....	85
6.3.	ประสบการณ์ที่ได้รับและสิ่งที่ได้เรียนรู้ .....	86
6.4.	แนวทางการพัฒนาต่อยอด .....	86
	บรรณานุกรม .....	88

## สารบัญรูปภาพ

รูปภาพ	หน้า
รูปภาพที่ 2.1 SCB Developers .....	9
รูปภาพที่ 2.2 Management Application .....	9
รูปภาพที่ 2.3 Client Credentials Grant Type .....	10
รูปภาพที่ 2.4 Generating a Deeplink URL .....	11
รูปภาพที่ 2.5 SCB EASY App Payment - Technical Flow .....	11
รูปภาพที่ 2.6 C Scan B Payment: Thai QR Code Tag 30 (QR 30) - Technical Flow.....	13
รูปภาพที่ 3.1 Activity Diagram แสดงการให้บริการในร้านอาหาร.....	18
รูปภาพที่ 3.2 Activity Diagram แสดงการให้บริการสั่งกลับบ้าน .....	19
รูปภาพที่ 3.3 Activity Diagram แสดงการอัปเดตเมนูอาหาร.....	20
รูปภาพที่ 3.4 Use Case Diagram .....	21
รูปภาพที่ 3.5 ER Table.....	27
รูปภาพที่ 4.1 หน้าแรกของเว็บไซต์ .....	32
รูปภาพที่ 4.2 หน้าแสดงเมนู.....	34
รูปภาพที่ 4.3 หน้าแสดงข้อมูลเมนูและเลือกเมนู .....	36
รูปภาพที่ 4.4 หน้าแสดงตະกร້າມເນຸ້ມທີ່ເລືອກ (DINE-IN).....	38
รูปภาพที่ 4.5 หน้าแสดงບິລແລະສຖານະອາຫາຣ.....	39
รูปภาพที่ 4.6 หน้าแสดงຕະກ້າມເນຸ້ມທີ່ເລືອກ (TAKE-OUT) .....	41
รูปภาพที่ 4.7 หน้าເລືອກໜ່ວຍທາງການຈໍາຮະເງິນ.....	43
รูปภาพที่ 4.8 หน้าแสดง QR Code ສໍາໜັບຈໍາຮະເງິນ ແລະການຈໍາຮະເງິນຜ່ານ QR Code .....	45
รูปภาพที่ 4.9 ການຈໍາຮະເງິນຜ່ານ SCB Easy Sandbox.....	48
รูปภาพที่ 4.10 ໃບເສົ້າ .....	50
รูปภาพที่ 4.11 หน້າຈັດການພາວຽກຢາຍໃນຮ້ານ .....	53
รูปภาพที่ 4.12 หน້າຈັດການຮ່າຍລະເວີຍດົບ .....	55
รูปภาพที่ 4.13 หน້າສໍາໜັບເປີດໂຕບົນກຳ .....	56
รูปภาพที่ 4.14 หน້າແສດງ QR Code ສໍາໜັບລູກຄ້າ DINE-IN .....	58
รูปภาพที่ 4.15 หน້າຈັດການຮ່າຍການອາຫາຣໃນຄວ້າ .....	59
รูปภาพที่ 4.16 หน້າຈັດການເມຸນ .....	61

รูปภาพที่ 4.17 หน้าจัดการหมวดหมู่ .....	62
รูปภาพที่ 4.18 หน้าจัดการกลุ่มตัวเลือก .....	63
รูปภาพที่ 4.19 หน้าจัดการตัวเลือก .....	65
รูปภาพที่ 4.20 หน้าจัดการໂຕະ .....	67
รูปภาพที่ 5.1 ตัวอย่างรายชื่อ Endpoints ในระบบ .....	70
รูปภาพที่ 5.2 การใช้งาน Postman ในการทดสอบ API.....	72
รูปภาพที่ 5.3 การตรวจสอบ Controller และ Service ควบคู่กับ Postman.....	72
รูปภาพที่ 5.4 ฐานข้อมูลมีการอัปเดตข้อมูลจากระบบ .....	73
รูปภาพที่ 5.5 หน้าเว็บมีการแสดงผลข้อมูลที่ถูกต้องหลังจากการทดสอบผ่าน API.....	73
รูปภาพที่ 5.6 นำอาหารใส่ตะกร้าได้เมื่อทำการเลือกรอบตามเงื่อนไข .....	74
รูปภาพที่ 5.7 ไม่สามารถสั่งอาหารได้เมื่อไม่มีรายการอาหารในตะกร้า .....	75
รูปภาพที่ 5.8 แท็บแสดงตะกร้าและบิลสำหรับลูกค้า DINE-IN .....	75
รูปภาพที่ 5.9 ต้องทำการเลือกหมายเลขอ้างอิงก่อนเปิดໂຕະ .....	77
รูปภาพที่ 5.10 บิล DINE-IN จะไม่สามารถชำระเงินได้หากไม่มีรายการอาหารที่สั่ง .....	78
รูปภาพที่ 5.11 บิล DINE-IN ที่ชำระเงินแล้วจะไม่สามารถ Delete รายการอาหารได.....	78
รูปภาพที่ 5.12 บิล TAKE-OUT ไม่สามารถดู QR Code และไม่สามารถเปลี่ยนสถานะรายการอาหารได้ ....	79
รูปภาพที่ 5.13 บิล TAKE-OUT ที่ชำระเงินแล้วจะไม่สามารถ Delete รายการอาหารได.....	79
รูปภาพที่ 5.14 การแสดงผลหน้าจัดการรายการอาหารในครัว.....	80
รูปภาพที่ 5.15 การเพิ่มเมนูใหม่ต้องกรอกข้อมูลที่จำเป็นให้ครบ.....	81
รูปภาพที่ 5.16 การเพิ่มกลุ่มตัวเลือกใหม่ต้องกรอกข้อมูลที่จำเป็นให้ครบ .....	82
รูปภาพที่ 5.17 การเพิ่มตัวเลือกใหม่ต้องกรอกข้อมูลที่จำเป็นให้ครบ .....	83

## สารบัญตาราง

ตาราง	หน้า
ตารางที่ 1.1 ตารางแสดงระยะเวลา 1 ภาคการศึกษา (ภาคปลาย ปีการศึกษา 2564).....	4
ตารางที่ 3.1 ตารางแสดง Use Case Description ของการสั่งอาหาร.....	22
ตารางที่ 3.2 ตารางแสดง Use Case Description ของการตรวจสอบรายการอาหาร.....	23
ตารางที่ 3.3 ตารางแสดง Use Case Description ของการชำระเงิน (ลูกค้า) .....	23
ตารางที่ 3.4 ตารางแสดง Use Case Description ของการชำระเงิน (พนักงาน, ผู้จัดการ).....	24
ตารางที่ 3.5 ตารางแสดง Use Case Description ของการเข้าสู่ระบบ.....	24
ตารางที่ 3.6 ตารางแสดง Use Case Description ของการจัดการรายการอาหาร.....	25
ตารางที่ 3.7 ตารางแสดง Use Case Description ของรายการอาหารที่ต้องทำ .....	25
ตารางที่ 3.8 ตารางแสดง Use Case Description ของการสร้าง แก้ไข หรือลบเมนู.....	26
ตารางที่ 3.9 ตารางแสดง Data Dictionary: bill .....	27
ตารางที่ 3.10 ตารางแสดง Data Dictionary: index_category .....	28
ตารางที่ 3.11 ตารางแสดง Data Dictionary: item .....	28
ตารางที่ 3.12 ตารางแสดง Data Dictionary: item_optional .....	29
ตารางที่ 3.13 ตารางแสดง Data Dictionary: optional .....	29
ตารางที่ 3.14 ตารางแสดง Data Dictionary: optional_item .....	29
ตารางที่ 3.15 ตารางแสดง Data Dictionary: order_info.....	30
ตารางที่ 3.16 ตารางแสดง Data Dictionary: payment .....	30
ตารางที่ 3.17 ตารางแสดง Data Dictionary: select_item .....	31
ตารางที่ 3.18 ตารางแสดง Data Dictionary: table_info .....	31
ตารางที่ 3.19 ตารางแสดง Data Dictionary: token .....	31

## บทที่ 1

### บทนำ

ในบทนี้จะอธิบายถึงที่มาและความสำคัญ วัตถุประสงค์ และขอบเขตของโครงการ รวมถึงเครื่องมือที่ใช้ในการพัฒนา ประโยชน์ที่คาดว่าจะได้รับ และระยะเวลาในการดำเนินโครงการ

#### 1.1. ที่มาและความสำคัญของโครงการ

ในปัจจุบันมีร้านอาหารหลายร้านมามายที่ยังคงใช้วิธีการสั่งอาหารและการชำระเงินในรูปแบบดั้งเดิมอยู่ โดยการที่ให้ลูกค้าเลือกดูเมนูและทำการสั่งอาหารผ่านพนักงาน และชำระเงินได้ที่แคชเชียร์ แต่ในบางครั้งลูกค้าอาจจะต้องรอนานกว่าได้รับประทานอาหาร ไม่ว่าจะเป็นการรอเมนูอาหารจากพนักงาน หรือการรอพนักงานมารับรายการอาหาร และเมื่อรับประทานอาหารเสร็จอาจจะต้องมาต่อคิวในการรอชำระเงิน เพราะว่ามีลูกค้าเข้ามาใช้บริการในร้านอาหารจำนวนมาก ทำให้พนักงานอาจจะไม่สามารถบริการได้อย่างทั่วถึงและรวดเร็วเท่าที่ควร

เนื่องจากสถานการณ์ COVID-19 ที่มีมาตรการป้องกันในการลดการสัมผัสเพื่อลดความเสี่ยงในการติดเชื้อ และลักษณะการให้บริการแบบบริการตนเองที่ได้รับความนิยมมากขึ้น เพื่อยกระดับการบริการของร้านอาหารให้สะดวกและรวดเร็วยิ่งขึ้น จึงจัดทำเว็บแอปพลิเคชันเพื่ออำนวยความสะดวกให้ลูกค้าสามารถมองเมนูและสามารถเลือกสั่งอาหารเมนูที่ต้องการได้ด้วยตนเอง โดยไม่จำเป็นต้องรอพนักงานมารับรายการอาหาร และหลังจากรับประทานอาหารเสร็จสามารถชำระเงินได้ด้วยตนเองโดยที่ไม่ต้องพึ่งพนักงานหรือไปชำระเงินที่แคชเชียร์

นอกจากนี้ในทางด้านของร้านอาหารยังเป็นการลดการใช้งานเมนูอาหารแบบสื่อสิ่งพิมพ์ ทำให้สามารถปรับเปลี่ยนข้อมูลหรือเพิ่มลดเมนูได้โดยที่ไม่ต้องทำสื่อสิ่งพิมพ์ขึ้นมาใหม่ และช่วยลดภาระให้กับพนักงานภายในร้านอาหารที่ต้องมารับรายการอาหารและรับชำระเงิน ทำให้การบริการโดยรวมสามารถทำได้อย่างรวดเร็วยิ่งขึ้น

## 1.2. วัตถุประสงค์

- 1) เพื่อออกแบบและพัฒนาเว็บแอปพลิเคชันที่สามารถสั่งอาหารและชำระเงินได้
- 2) เพื่อออกแบบและพัฒนาเว็บแอปพลิเคชันที่สามารถจัดการข้อมูลอาหารในร้านค้าได้
- 3) เพื่อศึกษาและออกแบบเว็บแอปพลิเคชันให้มีความสะดวกสบาย เหมาะกับผู้ใช้งาน
- 4) เพื่อศึกษาและทดสอบการใช้งานระบบการชำระเงินผ่าน SCB Developers

## 1.3. ขอบเขตของโครงการ

เว็บแอปพลิเคชันนี้จะทำให้ลูกค้าสามารถสั่งอาหารได้ด้วยตนเองโดยไม่ต้องรอพนักงานมารับรายการที่ต้องการสั่งและจ่ายเงินได้ด้วยตนเองโดยไม่ต้องรอคิวจ่ายเงิน เว็บแอปพลิเคชันนี้จะแบ่งเป็น 3 ส่วนได้แก่ ส่วนของลูกค้าร้านอาหาร ส่วนของพนักงานร้านอาหาร และส่วนของผู้จัดการร้านอาหาร โดยมีคุณสมบัติหลักดังต่อไปนี้

- 1) ลูกค้าสามารถสั่งอาหารพร้อมทั้งเลือกตัวเลือกเสริมต่าง ๆ ได้ด้วยตนเอง
- 2) ลูกค้าสามารถจ่ายเงินผ่านทางเว็บแอปพลิเคชันได้ด้วยตนเอง
- 3) พนักงานสามารถดูและจัดการรายการอาหารที่ได้รับได้
- 4) ผู้จัดการสามารถจัดการเพิ่ม ลบ และแก้ไขรายการอาหารในร้านได้

## 1.4. เครื่องมือที่ใช้ในการพัฒนา

### 1.4.1. ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

- IntelliJ IDEA (version 2021.2 Ultimate Edition)
- Postman (version 9.11.0)

### 1.4.2. ภาษาที่ใช้ในการพัฒนาระบบ

- Java (version JDK 8u321)
- JavaScript

### 1.4.3. ไลบรารีและเฟรมเวิร์คที่ใช้ในการพัฒนาระบบ

- Spring Boot (version 2.6.3)
- Bootstrap (version 5.1.3)
- jQuery (version 3.6.0)
- Bootstrap Table (version 1.19.1)

## 1.5. ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับความรู้ในการออกแบบระบบพัฒนาระบบและเว็บแอปพลิเคชัน
- 2) ประหยัดเวลาในการรอสั่งอาหารและจ่ายเงิน
- 3) เพิ่มประสิทธิภาพในการให้บริการลูกค้าที่มาใช้บริการร้านอาหาร
- 4) ลดการใช้เมนูอาหารแบบสื่อสิ่งพิมพ์

### 1.6. ระยะเวลาในการดำเนินโครงการ

ตารางที่ 1.1 ตารางแสดงระยะเวลา 1 ภาคการศึกษา (ภาคปลาย ปีการศึกษา 2564)

ขั้นตอนการ ดำเนินงาน	ช่วงเวลาปฏิบัติงาน				
	ธ.ค	ม.ค	ก.พ	มี.ค	เม.ย
ศึกษาข้อมูล	✓				
วิเคราะห์และ ออกแบบระบบ		✓	✓		
พัฒนาระบบ		✓	✓	✓	✓
ทดสอบและแก้ไข				✓	✓
จัดทำเอกสารสรุป โครงการ				✓	✓

### สรุป

จากสิ่งที่กล่าวไปข้างต้นในบทนี้ ทำให้ทราบถึงที่มา ความสำคัญ และขอบเขตของโครงการทำให้เห็น  
ภาพรวมของโครงการเพื่อให้สามารถวางแผนในการออกแบบ พัฒนาโครงการในระยะเวลาที่กำหนดได้

## บทที่ 2

### ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

ในบทนี้จะอธิบายถึงทฤษฎีและเทคโนโลยีที่เกี่ยวข้องที่นำมาใช้พัฒนาเว็บแอปพลิเคชันในโครงงานเพื่อให้เข้าใจในสิ่งที่คณจะทำได้มากขึ้น

ในโครงงานนี้พัฒนาด้วย Spring Boot โดยใช้ Spring MVC ในการสร้าง API และจัดเก็บข้อมูลลงในฐานข้อมูล MySQL และใช้ Bootstrap และ jQuery ทำในส่วนของหน้าเว็บแอปพลิเคชัน และมีการจำลองการจ่ายเงินโดยการเชื่อมต่อระบบกับธนาคารไทยพาณิชย์

#### 2.1. Spring Boot

Spring Boot เป็นหนึ่งในโปรเจกของ Spring Framework ที่ช่วยทำให้การพัฒนาแอปพลิเคชันที่มีพื้นฐานมาจาก Spring แบบ stand-alone สามารถทำได้รวดเร็วขึ้นในระดับที่สามารถใช้งานได้จริง โดยสามารถเรียกใช้งานได้ทันที มีการทำ Auto Configuration ทำให้ไม่ต้องเสียเวลาในการ Config ด้วยตนเองทุกอย่าง [1]

คุณสมบัติของ Spring Boot ได้แก่

- สร้าง Spring applications แบบ stand-alone
- มี Embed Tomcat, Jetty or Undertow (ไม่จำเป็นต้อง deploy WAR files)
- มี POM ‘starter’ เพื่อลดความซับซ้อนในการกำหนด build configuration
- ช่วย Configure Spring แบบอัตโนมัติเมื่อได้ก็ตามที่เป็นไปได้
- มี Production-ready features เช่น metrics, health checks และ externalized configuration
- ไม่จำเป็นต้องมี XML configuration

## 2.2. Spring MVC

Framework ใช้ในการสร้างเว็บแอปพลิเคชันหรือเว็บ Service ถูกออกแบบให้การทำงานขึ้นอยู่กับ Servlet ที่ชี้อ่วงว่า DispatcherServlet ซึ่งทำหน้าที่รับส่ง Request/Response ควบคุมการทำงานทั้งหมด ในกลไกของ @Controller ยังช่วยให้สามารถสร้างเว็บไซต์และแอปพลิเคชัน RESTful ผ่าน Annotation @PathVariable และคุณสมบัติอื่น ๆ ได้อีกด้วย [2]

## 2.3. API

API ย่อมาจาก Application Programming Interface คือตัวกลางการเชื่อมต่อระหว่างซอฟต์แวร์ปลายทางกับเซิร์ฟเวอร์ที่สร้าง API คือรับคำสั่งต่าง ๆ มีหน้าที่ทำให้ซอฟต์แวร์สามารถเข้าถึงข้อมูลหรืออัปเดตข้อมูลได้ เนื่องจากมีการกำหนดขอบเขตการเข้าถึงบริการจึงทำให้ป้องกันการเข้าถึงข้อมูลทั้งหมดโดยตรงได้อีกด้วย โดยในโครงงานนี้จะนำมาใช้ในการเรียกข้อมูลที่ต้องการจากฐานข้อมูลนำไปใช้บนเว็บแอปพลิเคชัน เช่น รายการอาหารต่าง ๆ ข้อมูลรายการอาหารที่สั่ง และส่งข้อมูลจากเว็บแอปพลิเคชันไปบันทึกการเปลี่ยนแปลงบนฐานข้อมูล เช่น การแก้ไขข้อมูลรายการอาหาร การสร้างรายการอาหาร [3]

## 2.4. MySQL

โปรแกรมโอเพนซอร์สจัดการฐานข้อมูลแบบ Relational Database Management System (RDBMS) ที่ใช้ภาษา SQL รองรับระบบปฏิบัติการหลากหลายรุ่น มีความรวดเร็ว ยืดหยุ่น สามารถใช้งานร่วมกับ Web Development Platform ต่าง ๆ รองรับข้อมูลขนาดใหญ่และมีการใช้งานที่สะดวกสบายโดยข้อมูลที่เก็บจะอยู่ในรูปแบบของตาราง มีส่วนเชื่อมต่อกับภาษาที่ใช้พัฒนาโปรแกรมเพื่อให้สามารถเข้าถึงฟังก์ชันการทำงานกับฐานข้อมูล MySQL ได้ เช่น ODBC (Open Database Connector), JDBC (Java Database Connectivity) และมี API ต่าง ๆ ให้เลือกใช้มากมายในการที่เข้าถึง MySQL โดยไม่ขึ้นอยู่กับภาษาที่ใช้ในการพัฒนาโปรแกรม [4]

## 2.5. Bootstrap

Frontend Framework ที่มี HTML, CSS และ JavaScript Library ซึ่งมุ่งเน้นในด้านการออกแบบส่วนประกอบสำหรับรูปแบบ เช่น ปุ่มกด ช่องสำหรับกรอกข้อมูล แถบเมนู สำหรับพัฒนาหน้าเว็บ แบบ Responsive ที่รองรับการแสดงบนหน้าจอเมื่อถือ มีระบบ Grid ช่วยในการจัดวาง Layout ได้ทุกขนาดของหน้าจอแบบต่างๆ และมี JavaScript Plugin อีกมากมาย [5]

## 2.6. jQuery

JavaScript Library ที่ถูกออกแบบเพื่อให้สามารถแก้ปัญหาการใช้ JavaScript ที่ยุ่งยากให้ง่ายยิ่งขึ้นโดยการจัดชุดคำสั่งที่เข้าใจง่าย สามารถเขียนได้ในรูปแบบสั้น มีความยืดหยุ่นสามารถทำงานได้ในทุกบริาระเชอร์ มี Plugin จำนวนมากให้เลือกใช้ เช่น หน้าเว็บที่ต้องการความ Interactive โดย jQuery Library ประกอบด้วยฟีเจอร์ต่าง ๆ ดังนี้ HTML/DOM Manipulation, CSS Manipulation, DOM Element Selection, HTML Event Methods, Effects and Animations และ AJAX [6]

## 2.7. IntelliJ IDEA

IntelliJ IDEA คือซอฟต์แวร์ Integrated Development Environment (IDE) สำหรับภาษา JVM (Java Virtual Machine) ที่ออกแบบมาเพื่อเพิ่มประสิทธิภาพการทำงานของนักพัฒนาซอฟต์แวร์สูงสุด ถูกพัฒนาขึ้นมาโดย JetBrains มีคุณสมบัติช่วยเหลือในการเขียนโค้ด ภายในโปรแกรมมาพร้อมกับเครื่องมือต่าง ๆ และรองรับการบูรณาการร่วมกับโปรแกรมอื่น ๆ สามารถส่วนเสริมเพิ่มเติมได้ รองรับการเขียนโปรแกรมได้หลากหลายภาษา [7]

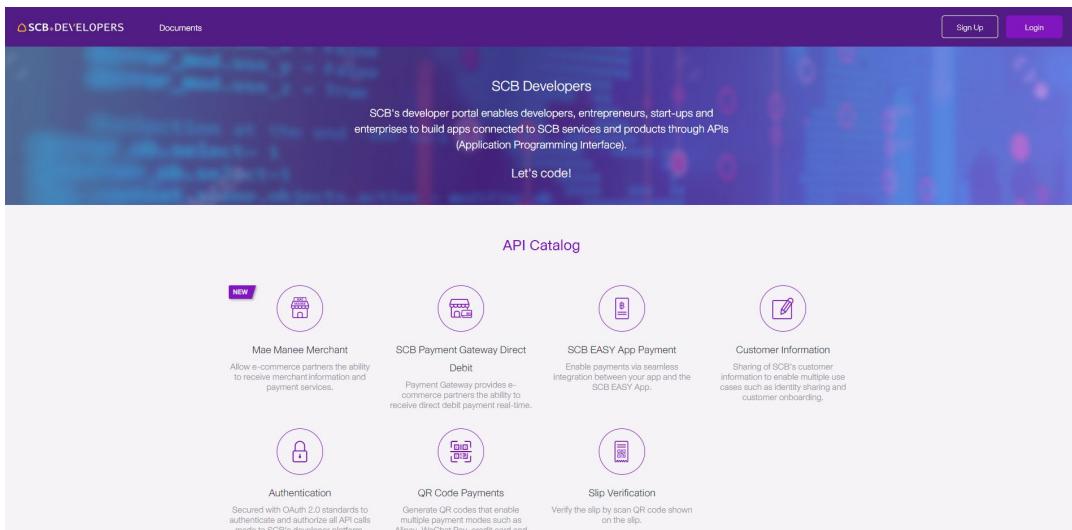
## 2.8. Postman

API Platform สำหรับพัฒนาและทดสอบ API Service ซึ่งเป็นที่นิยมอย่างกว้างขวาง เพราะการใช้งานที่ง่ายดาย มี UI สวยงาม และสามารถใช้งานได้ฟรี พัฒนาขึ้นทั่วไปคือการส่ง Request เพื่อคุณ Response ที่ได้กลับมาและยังมีฟังก์ชันอื่น ๆ ที่มีประโยชน์ต่อผู้ที่ใช้งาน Postman [8] ได้แก่

- สามารถจัดเก็บ API ที่สร้างเองหรือนำเข้ามาได้เป็น Collection เพื่อให้ง่ายต่อการค้นหาและทดสอบ
- สามารถสร้าง Environment ที่ต้องการทดสอบเก็บไว้ได้สำหรับการทดสอบครั้งต่อไปที่ต้องการทดสอบใน Environment แบบต่าง ๆ ที่แตกต่างกัน
- การเขียน Script test แต่ละ API เพื่อตรวจสอบความถูกต้องว่า Response ที่ได้รับกลับมาว่า Status code หรือเนื้อหาตรงตามที่ต้องการหรือไม่
- Collection และ Environment สามารถที่จะ Export to file เพื่อให้ผู้อื่นสามารถนำไปใช้ต่อได้

## 2.9. SCB Developers

SCB Developers หรือที่รู้จักกันในชื่อของ SCB Open API จากทางธนาคารไทยพาณิชย์ที่เปิดโอกาสให้นักพัฒนาได้มีโอกาสทดลองพัฒนาโปรแกรมขึ้นมาเพื่อทดสอบการใช้งานร่วมกับทางธนาคารไทยพาณิชย์ผ่าน API ที่ธนาคารเตรียมเอาไว้ให้ทดลองใช้งานบน Sandbox Environment โดยมีบัญชีธนาคารลูกค้าและร้านค้าจำลองให้สามารถจำลองการทำธุกรรมทางการเงินแบบเสมือนจริงได้ ตัวอย่าง API ที่มีให้บริการ เช่น SCB EASY App Payment, QR Code Payments, Slip Verification เป็นต้น [9]

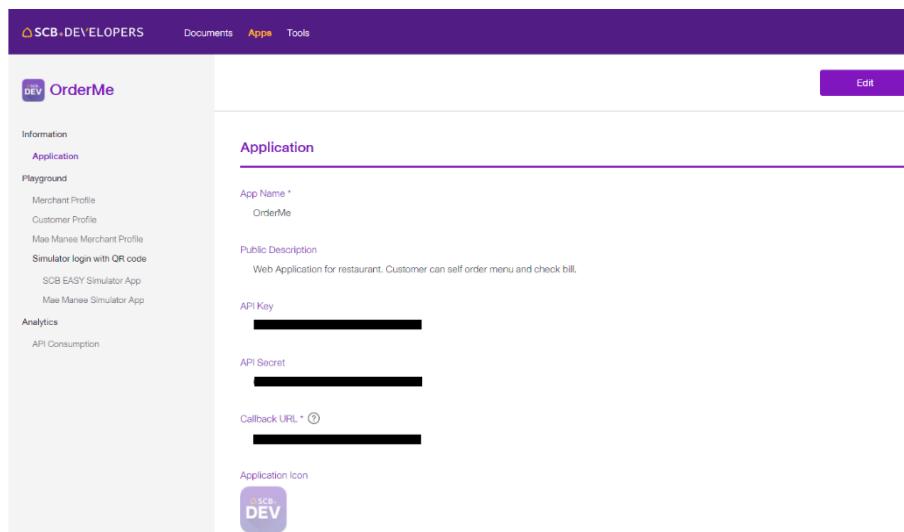


รูปภาพที่ 2.1 SCB Developers

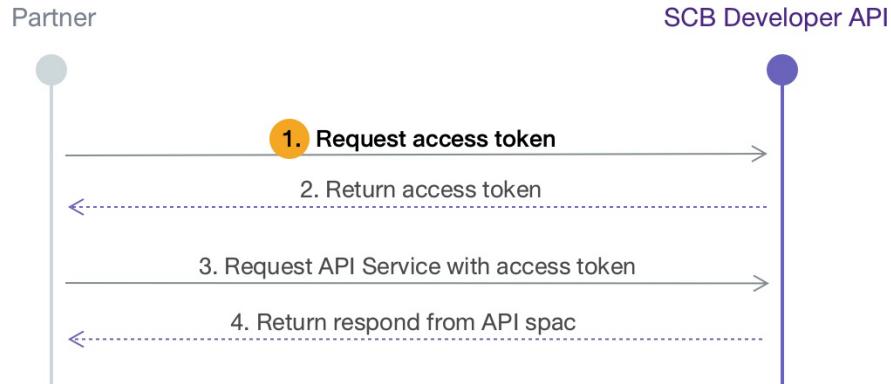
### 2.9.1.Authentication

API ทั้งหมดของ SCB's Open API Platform มีการรักษาความปลอดภัยด้วย มาตรฐาน OAuth 2.0 ดังนั้นในการจะใช้งานจึงจำเป็นจะต้องทำการ Authentication โดยการ POST Request เพื่อขอ Access Token สำหรับการใช้งาน API

API Key และ API Secret สามารถหาได้หลังจากเพิ่มแอปพลิเคชันที่ต้องการพัฒนาลง ไปในระบบของ SCB Developers



รูปภาพที่ 2.2 Management Application



รูปภาพที่ 2.3 Client Credentials Grant Type

#### Example Request

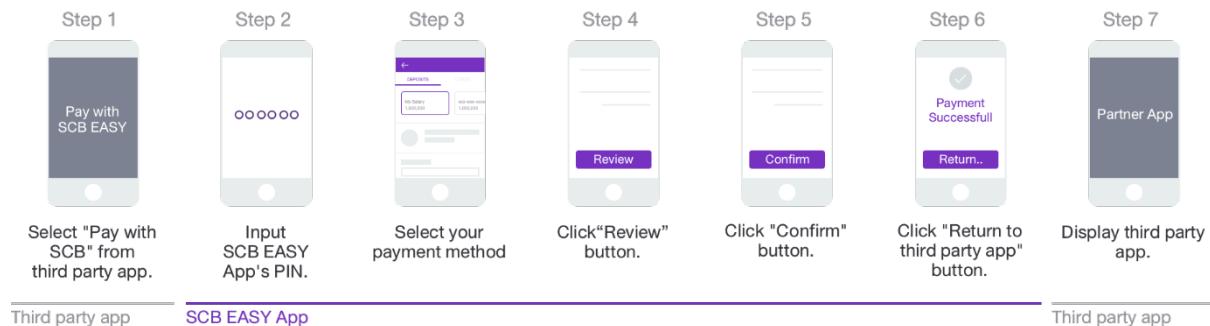
```
curl -X POST \
  https://api-sandbox.partners.scb/partners/sandbox/v1/oauth/token \
  -H 'Content-Type: application/json' \
  -H 'accept-language: EN' \
  -H 'requestUID: 85230887-e643-4fa4-84b2-4e56709c4ac4' \
  -H 'resourceOwnerId: <Your API Key>' \
  -d '{
    "applicationKey" : "<You API Key>",
    "applicationSecret" : "<Your API Secret>",
    "authCode" : "<Authorization Code>"
  }'
```

#### Example Response

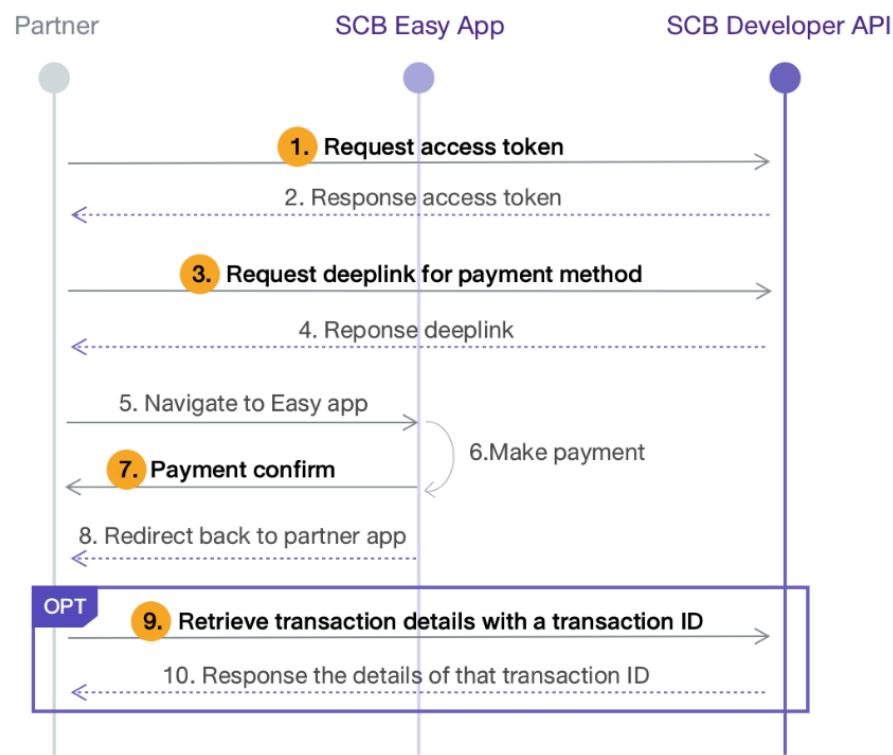
```
{
  "status": {
    "code": 1000,
    "description": "Success"
  },
  "data": {
    "accessToken": "34362373-66e8-4db0-80e5-0755b67e51f9",
    "tokenType": "Bearer",
    "expiresIn": 1800,
    "expiresAt": 1550133185,
    "refreshToken": "9e80be84-5eb7-4e8c-a885-a36ff3eb6684",
    "refreshExpiresIn": 3600,
    "refreshExpiresAt": 1550134985
  }
}
```

## 2.9.2.SCB EASY App Payment

การสร้าง Deeplink สำหรับการชำระเงินผ่าน SCB EASY สามารถทำได้ด้วยการ POST Request ตามวิธีการดังต่อไปนี้



## รูปภาพที่ 2.4 Generating a Deeplink URL



## รูปภาพที่ 2.5 SCB EASY App Payment - Technical Flow

**Example Request**

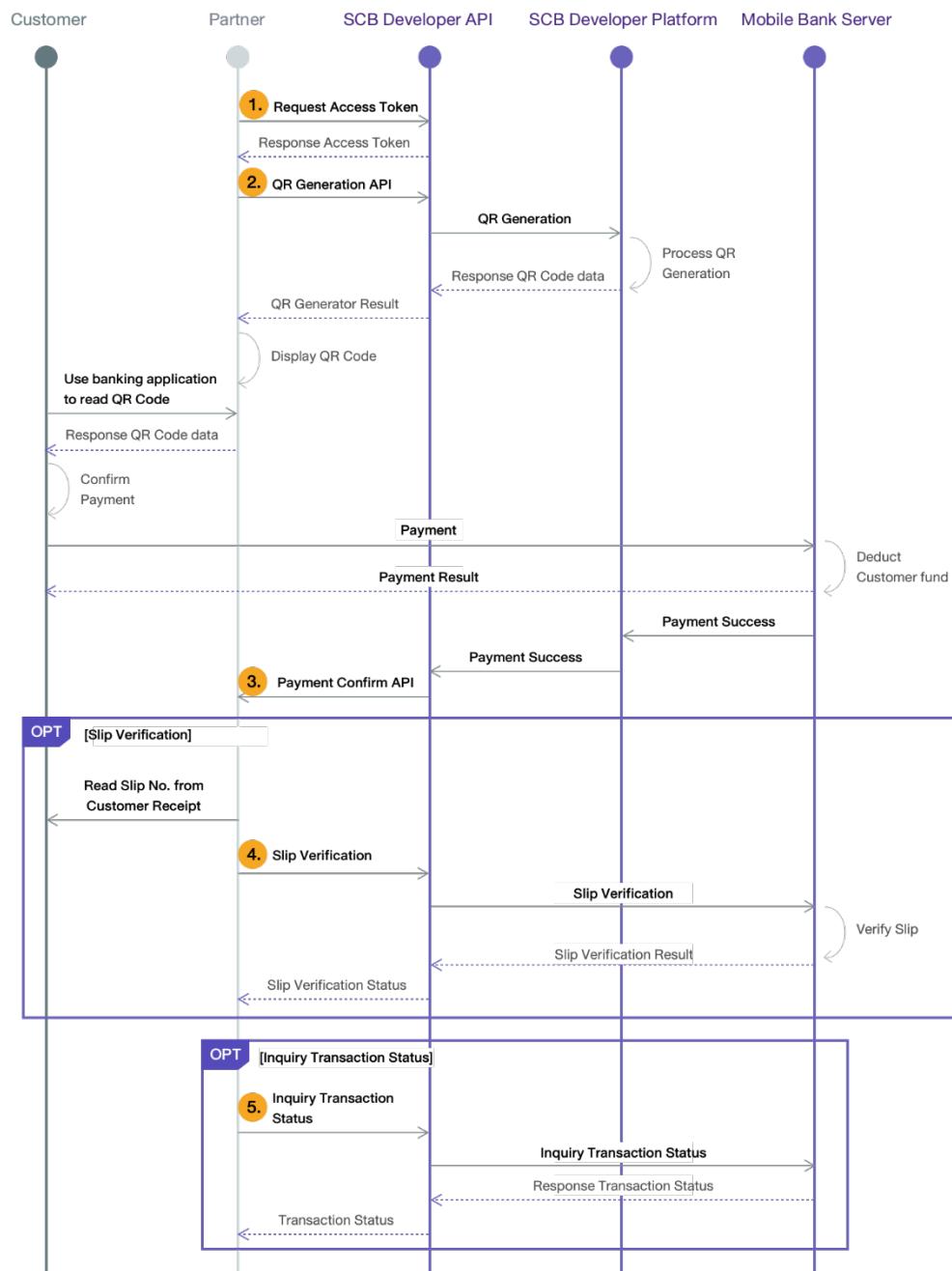
```
curl -X POST \
  https://api-sandbox.partners.scb/partners/sandbox/v3/deeplink/transactions \
  -H 'content-type: application/json' \
  -H 'authorization: Bearer <Your Access Token>' \
  -H 'accept-language: EN' \
  -H 'resourceOwnerId: <Your API Key>' \
  -H 'requestUID: d7e992f3-c9f1-4071-8a4a-6c5839c8d317' \
  -H 'channel: scbeasy' \
  -d '{
    "transactionType": "PURCHASE",
    "transactionSubType": ["BP", "CCFA", "CCIPP"],
    "sessionValidityPeriod": 60,
    "sessionValidUntil": "",
    "billPayment": {
      "paymentAmount": 10000.00,
      "accountTo": "123456789012345",
      "ref1": "ABCDEFGHIJ1234567890",
      "ref2": "ABCDEFGHIJ1234567890",
      "ref3": "ABCxxxx",
    },
    "creditCardFullAmount": {
      "merchantId": "4218170000000159",
      "terminalId": "56200003",
      "orderReference": "AA100001",
      "paymentAmount": 10000.00
    },
    "installmentPaymentPlan": {
      "merchantId": "4218170000000159",
      "terminalId": "56200003",
      "orderReference": "AA100001",
      "paymentAmount": 10000.00,
      "tenor": "12",
      "ippType": "3",
      "prodCode": "1001",
    },
    "merchantMetaData": {
      "callbackUrl": "http://my.server.com/bar",
      "merchantInfo": {
        "name": "SANDBOX MERCHANT NAME"
      }
    }
}'
```

**Example Response**

```
{
  "status": {
    "code": 1000,
    "description": "Deeplink successfully created"
  },
  "data": {
    "transactionId": "cae09821-a23d-4034-a35a-50c44a41ef67",
    "deeplinkUrl": "scbeasysim://purchase/254ae415-352c-48b2-9527-2e0190269976",
    "userRefId": "7efe0db1-2af6-49b1-b9e3-71a83ed80227"
  }
}
```

### 2.9.3.QR Code Payment

การสร้าง QR Code สำหรับการชำระเงินผ่าน Thai QR Code Tag 30 (QR 30) และ QR Card Scheme (QR CS) สามารถทำได้ด้วยการ POST Request ตามวิธีการดังต่อไปนี้



รูปภาพที่ 2.6 C Scan B Payment: Thai QR Code Tag 30 (QR 30) - Technical Flow

**Example Request**

```
curl -X POST \
  https://api-sandbox.partners.scb/partners/sandbox/v1/payment/qrcode/create \
  -H 'Content-Type: application/json' \
  -H 'accept-language: EN' \
  -H 'authorization: Bearer <Your Access Token>' \
  -H 'requestUIId: 390e8c45-a023-4fe2-826b-82854332e220' \
  -H 'resourceOwnerId: <Your API Key>' \
  -d '{
    "qrType": "PPCS",
    "ppType": "BILLERID",
    "ppId": "123456789012345",
    "amount": "100.00",
    "ref1": "REFERENCE1",
    "ref2": "REFERENCE2",
    "ref3": "SCB",
    "merchantId": "684349039613126",
    "terminalId": "379479514042628",
    "invoice": "INVOICE",
    "csExtExpiryTime": "60"
  }'
```

**Example Response**

```
{
  "status": {
    "code": 1000,
    "description": "Success"
  },
  "data": {
    "qrRawData": "000020101220216492984278344661552047011530376454031005802TH5922TestMerchant15548661816007BANGKOK627301232019041910394997200000005232019041910394997200000007153794795140426286304E253",
    "qrImage": "(Base64 Image)",
    "csExtExpiryTime": "2019-04-19 10:54:49",
    "responseCode": "000",
    "qrCodeType": "EM",
    "qrCodeId": "20190419111003263000000",
    "poi": "12",
    "amount": "100.00",
    "currencyCode": "764",
    "currencyName": "Baht",
    "csNote": "",
    "invoice": "INVOICE",
    "merchantId": "684349039613126",
    "merchantName": "TestMerchart",
    "csUserDefined": "",
    "terminalId": "379479514042628",
    "terminalName": "Sandbox Terminal",
    "channels": [
      {
        "seqNo": "1",
        "channelName": "VISA",
        "channelCode": "VSA"
      }
    ]
  }
}
```

## สรุป

จากทฤษฎีและเทคโนโลยีที่เกี่ยวข้องที่กล่าวมาทั้งหมด เป็นองค์ประกอบหลักสำคัญที่ใช้ในการพัฒนาระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร โดยมีทั้งส่วนที่เป็น Framework เครื่องมือ และทฤษฎี ที่ได้ถูกนำมาใช้งานร่วมกันเพื่อให้แอปพลิเคชันสามารถใช้งานได้ตามวัตถุประสงค์ที่ต้องการ

## บทที่ 3

### การวิเคราะห์และการออกแบบระบบ

ในบทนี้จะเป็นการเพิ่มความเข้าใจในหลักการทำงานของเว็บแอปพลิเคชันมากขึ้นโดยนำเสนอการออกแบบระบบในรูปแบบ Activity Diagram, Use Case Diagram และ Use Case Description ในส่วนของการออกแบบฐานข้อมูลจะนำเสนอโดยใช้ ER Table และ Data Dictionary

#### **3.1. การวิเคราะห์และการออกแบบความต้องการของระบบ**

จากการรวบรวมข้อมูลการให้บริการของร้านอาหารหลายร้านในปัจจุบัน และประสบการณ์ที่ได้รับจากการใช้บริการหลายครั้ง พบร่วมกันหลายครั้งที่ลูกค้าต้องเสียเวลาในการรอรับบริการเป็นเวลานาน ไม่ว่าจะเป็นการรอ เมนูอาหารจากพนักงาน การรอพนักงานมารับรายการอาหาร และการรอชำระเงิน ด้วยเหตุนี้จึงได้ทำการพัฒนาเว็บแอปพลิเคชันเพื่อยกระดับความสะดวกและความพึงพอใจของลูกค้า และเพิ่มประสิทธิภาพการให้บริการของร้านอาหารให้ดียิ่งขึ้น โดยมีรายละเอียดดังต่อไปนี้

##### **3.1.1. ระบบการแสดงผลเมนูและการสั่งอาหาร**

ระบบสำหรับให้ลูกค้าสามารถเลือกชมเมนูอาหาร และสั่งรายการอาหารที่ต้องการพร้อมตัวเลือกเสริมและหมายเหตุสำหรับรายการอาหารที่ต้องการสั่งได้ และมีระบบติดตามให้สามารถเลือกสั่งรายการอาหารพร้อมกันหลายรายการได้

##### **3.1.2. ระบบตรวจสอบรายการอาหารที่สั่ง**

ระบบสำหรับให้ลูกค้าสามารถตรวจสอบรายการอาหารที่สั่งทั้งหมดได้พร้อมรายละเอียดต่างๆ ไม่ว่าจะเป็น รายการที่สั่ง ตัวเลือก หมายเหตุ ราคา รวมถึงสถานะของรายการอาหารแต่ละอย่าง

##### **3.1.3. ระบบการชำระเงิน**

ระบบสำหรับให้ลูกค้าสามารถชำระเงินด้วยตนเองได้ หลังจากรับประทานอาหารเสร็จ และตรวจสอบรายการอาหารที่สั่งเรียบร้อย โดยสามารถชำระเงินได้ผ่าน SCB EASY หรือ QR Code (QR 30 & QR CS) ด้วยตนเอง หรือถ้าหากต้องการชำระด้วยเงินสดยังคงสามารถชำระเงินได้ผ่านแคชเชียร์ได้ตามปกติ

### 3.1.4.ระบบการจัดการรายการอาหารที่สั่ง

ระบบสำหรับให้พนักงานสามารถจัดการรายการอาหารที่สั่งของลูกค้าได้ เช่น การปรับ  
ลบรายการอาหารที่สั่ง ปรับเปลี่ยนสถานะรายการอาหาร

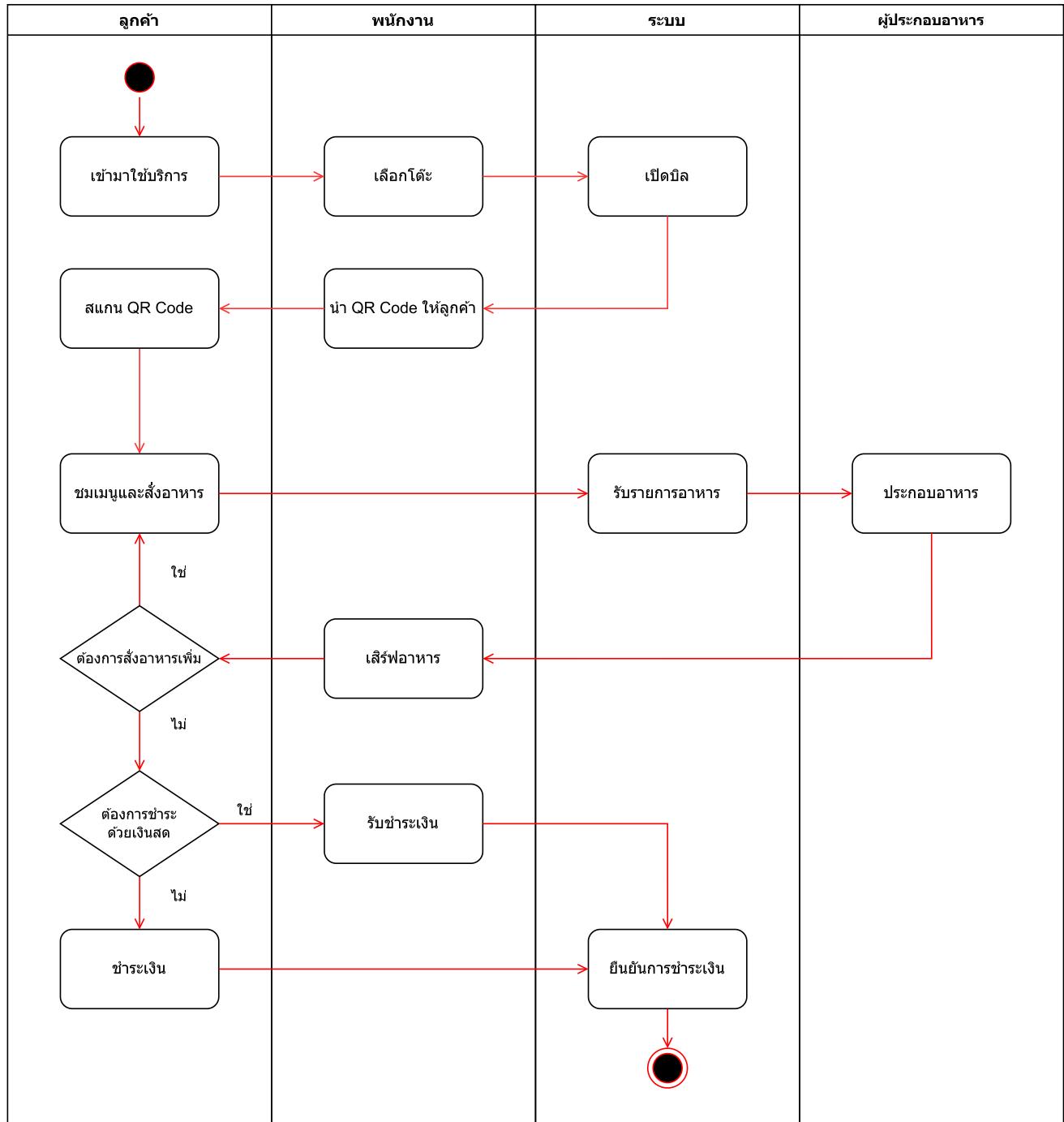
### 3.1.5.ระบบการจัดการเมนูอาหาร

ระบบสำหรับให้ผู้จัดการสามารถจัดการปรับเปลี่ยนข้อมูลของร้านอาหารต้นเองได้ เช่น  
เพิ่มลดเมนูอาหาร แก้ไขข้อมูลรายละเอียดอาหาร และจัดการตัวเลือกเสริมของอาหาร และ  
นอกจากนี้ยังสามารถปรับการแสดงผลหมวดหมู่อาหารให้เรียงตามที่ต้องการได้

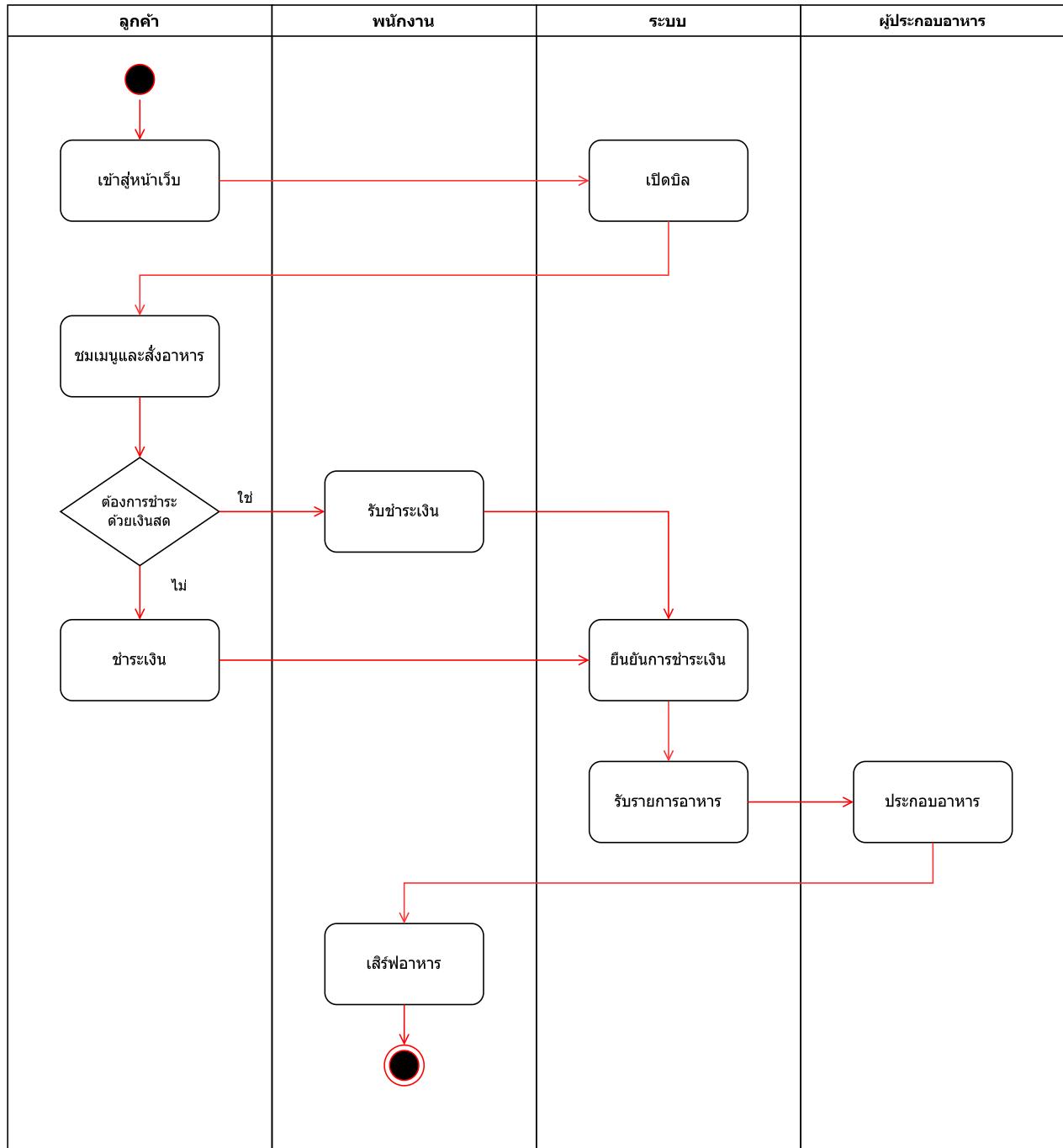
### 3.1.6.หน้าแสดงใบเสร็จ

หลังจากที่ลูกค้าทำการชำระเงินสำเร็จ ระบบจะทำการสร้างข้อมูลเกี่ยวกับใบเสร็จเก็บ  
เอาไว้ภายในฐานข้อมูล การแสดงหน้าใบเสร็จจะทำการตีงข้อมูลออกมาจากตารางที่ใช้เก็บ  
รายละเอียดการชำระเงิน เพื่อลดระยะเวลาที่ใช้ในการเข้าถึงข้อมูลรายการอาหารที่สั่งผ่าน  
ตารางความสมัพนธ์ของฐานข้อมูล และป้องกันปัญหาการแก้ไขข้อมูลรายการอาหาร ที่จะทำให้  
การแสดงผลไม่ถูกต้องเนื่องจากดึงข้อมูลมาจากฐานข้อมูลตารางอาหารในปัจจุบัน แต่  
รายละเอียดของข้อมูลอาหารอาจจะไม่ตรงกันระหว่างในอดีตกับปัจจุบัน

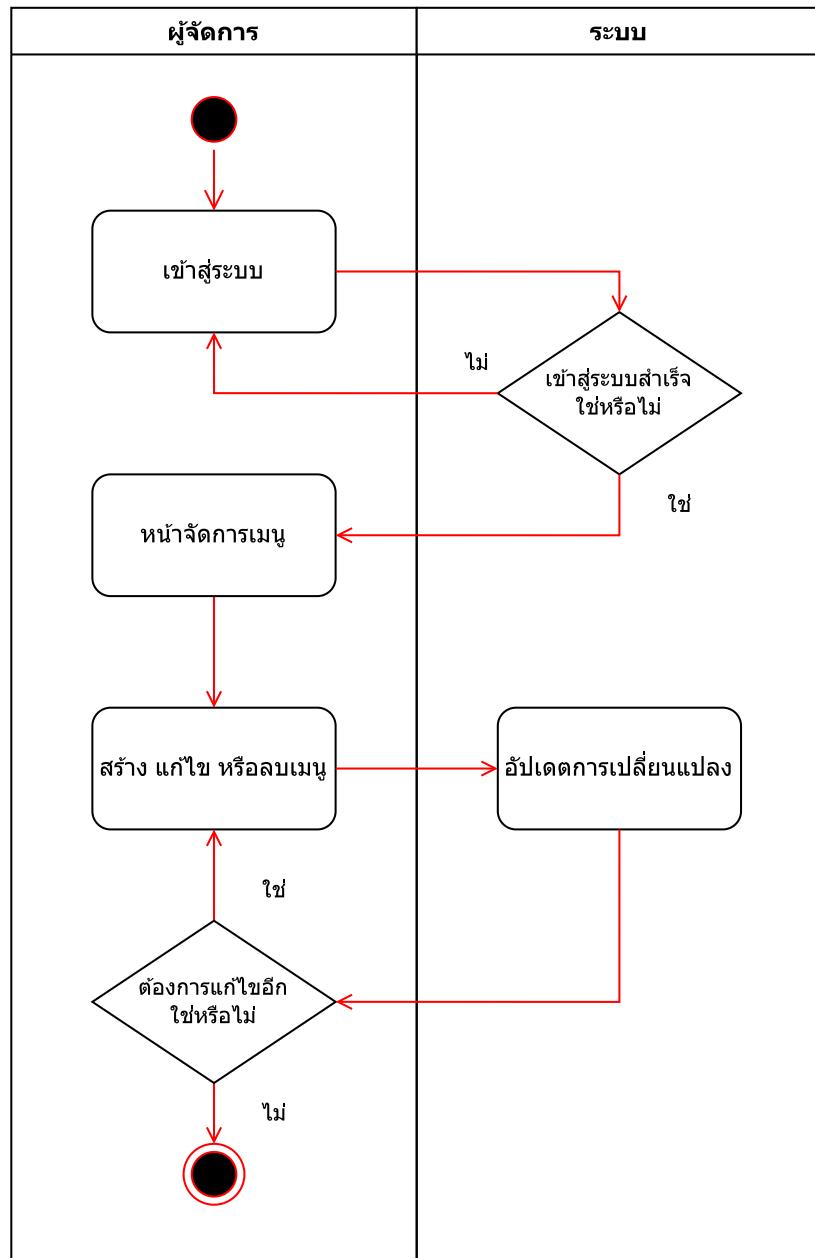
### 3.2. Activity Diagram



รูปภาพที่ 3.1 Activity Diagram แสดงการให้บริการในร้านอาหาร

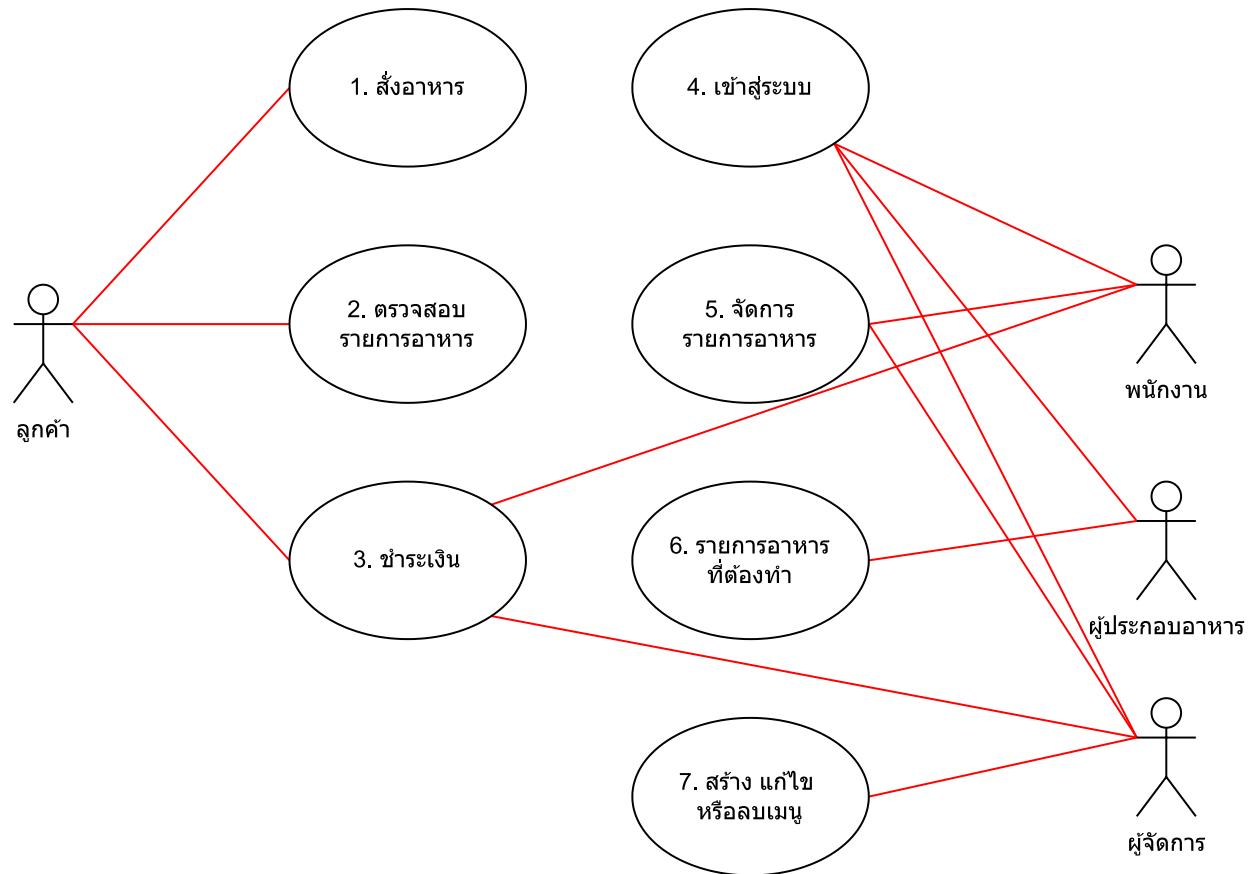


รูปภาพที่ 3.2 Activity Diagram แสดงการให้บริการสั่งกลับบ้าน



รูปภาพที่ 3.3 Activity Diagram แสดงการอัปเดตเมนูอาหาร

### 3.3. Use Case Diagram



รูปภาพที่ 3.4 Use Case Diagram

### 3.4. Use Case Description

ตารางที่ 3.1 ตารางแสดง Use Case Description ของการสั่งอาหาร

Use Case ID	1	
Use Case Name	สั่งอาหาร	
Actor	ลูกค้า	
Description	ชุมเมนู เลือกรายการอาหารที่ต้องการสั่ง พร้อมตัวเลือกและหมายเหตุ	
Precondition	สแกน QR Code จากพนักงาน	
Main Flow	ลูกค้า 2.เลือกรายการอาหารที่ต้องการ เลือก ตัวเลือกที่ต้องการ พร้อมหมายเหตุถ้า มี ระบุจำนวน และกดเพิ่มไปยังตะกร้า 5.กดไปที่ตะกร้าเพื่อตรวจสอบรายการ อาหารที่ต้องการสั่ง และกดปุ่มยืนยัน รายการอาหารเพื่อสั่งอาหาร	ระบบ 1.แสดงหน้าเมนู 3.เพิ่มรายการอาหารที่สั่งลงไปใน ตะกร้า 4.กลับไปแสดงหน้าเมนู 6.สร้างรายการอาหารที่สั่ง
Alternate Condition	-	

ตารางที่ 3.2 ตารางแสดง Use Case Description ของการตรวจสอบรายการอาหาร

<b>Use Case ID</b>	2	
<b>Use Case Name</b>	ตรวจสอบรายการอาหาร	
<b>Actor</b>	ลูกค้า	
<b>Description</b>	ตรวจสอบรายการอาหารที่สั่งไปแล้ว พร้อมสถานะของอาหาร	
<b>Precondition</b>	สแกน QR Code จากพนักงาน	
<b>Main Flow</b>	ลูกค้า	ระบบ
	1. กดไปที่รายการอาหารที่สั่ง	2. แสดงรายการอาหารที่สั่ง
<b>Alternate Condition</b>	-	

ตารางที่ 3.3 ตารางแสดง Use Case Description ของการชำระเงิน (ลูกค้า)

<b>Use Case ID</b>	3.1	
<b>Use Case Name</b>	ชำระเงิน	
<b>Actor</b>	ลูกค้า	
<b>Description</b>	ชำระเงิน สำหรับซองทาง SCB EASY และ QR Code	
<b>Precondition</b>	อยู่ในหน้าตรวจสอบรายการอาหาร	
<b>Main Flow</b>	ลูกค้า	ระบบ
	1. กดไปที่ชำระเงิน และเลือกซองทาง ชำระเงินที่ต้องการ	2. นำทางไปยัง SCB EASY หรือแสดง QR Code
	3. ทำการชำระเงินตามซองทางที่เลือก	4. ยืนยันการชำระเงิน
		5. แสดงรายละเอียดการชำระเงิน
<b>Alternate Condition</b>	ยังไม่มีรายการอาหารที่สั่ง	

ตารางที่ 3.4 ตารางแสดง Use Case Description ของการชำระเงิน (พนักงาน, ผู้จัดการ)

<b>Use Case ID</b>	3.2	
<b>Use Case Name</b>	ชำระเงิน	
<b>Actor</b>	พนักงาน, ผู้จัดการ	
<b>Description</b>	รับชำระเงินสดจากลูกค้า	
<b>Precondition</b>	อยู่ในหน้าจัดการรายการอาหาร	
<b>Main Flow</b>	พนักงาน, ผู้จัดการ	ระบบ
	1. กดเลือกบิลที่ต้องการชำระเงิน  3. รับเงินสดจากลูกค้า  4. กดชำระเงิน	2. แสดงรายละเอียดบิล  5. ยืนยันการชำระเงิน  6. แสดงรายละเอียดการชำระเงิน
<b>Alternate Condition</b>	-	

ตารางที่ 3.5 ตารางแสดง Use Case Description ของการเข้าสู่ระบบ

<b>Use Case ID</b>	4	
<b>Use Case Name</b>	เข้าสู่ระบบ	
<b>Actor</b>	พนักงาน, ผู้ประกอบอาหาร, ผู้จัดการ	
<b>Description</b>	เข้าสู่ระบบเพื่อยืนยันตัวตน และเข้าหน้าหลักของ Role	
<b>Precondition</b>	-	
<b>Main Flow</b>	พนักงาน, ผู้ประกอบอาหาร, ผู้จัดการ	ระบบ
	2. กรอก Username และ Password แล้วกด Login เพื่อเข้าสู่ระบบ	1. แสดงฟอร์มเข้าสู่ระบบ  3. ตรวจสอบความถูกต้อง  4. แสดงหน้าหลักของ Role
<b>Alternate Condition</b>	Username หรือ Password ไม่ถูกต้อง	

ตารางที่ 3.6 ตารางแสดง Use Case Description ของการจัดการรายการอาหาร

<b>Use Case ID</b>	5	
<b>Use Case Name</b>	จัดการรายการอาหาร	
<b>Actor</b>	พนักงาน, ผู้จัดการ	
<b>Description</b>	ดูบิล และแก้ไขข้อมูลที่ต้องการ	
<b>Precondition</b>	อยู่ในหน้าภาพรวมร้าน (หน้าหลัก)	
<b>Main Flow</b>	พนักงาน, ผู้จัดการ	ระบบ
	1. กดเลือกบิลที่ต้องการจัดการ 3. แก้ไขข้อมูลที่ต้องการ 4. กดบันทึกข้อมูล	2. แสดงรายละเอียดบิล 3. บันทึกข้อมูล 4. แสดงหน้าภาพรวมร้าน
<b>Alternate Condition</b>	-	

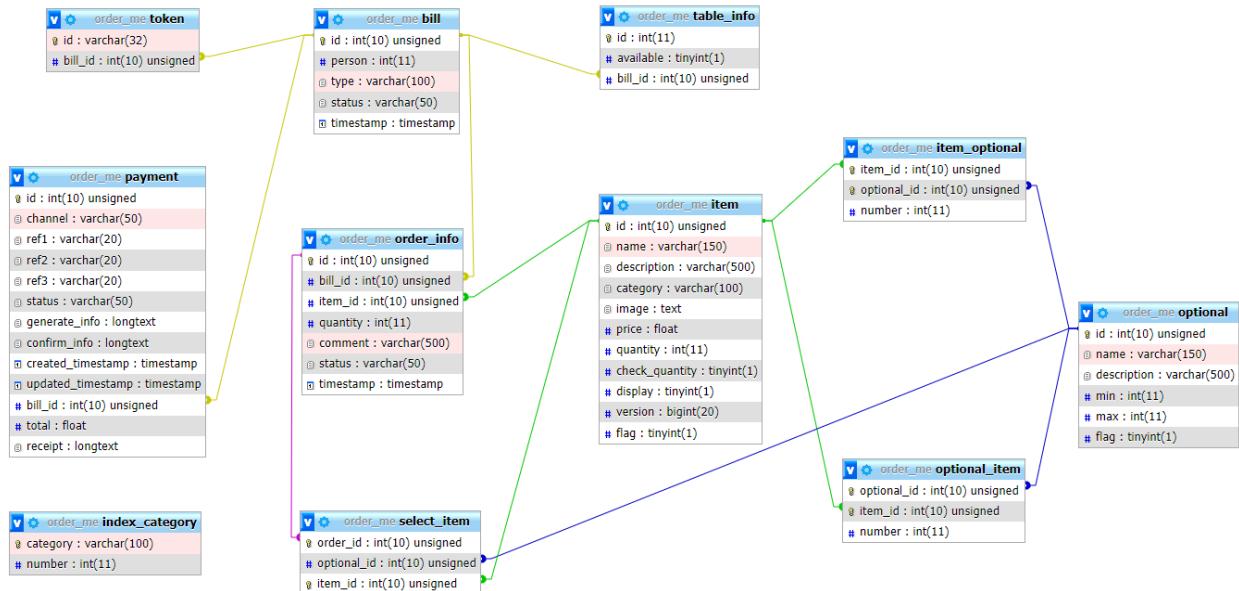
ตารางที่ 3.7 ตารางแสดง Use Case Description ของรายการอาหารที่ต้องทำ

<b>Use Case ID</b>	6	
<b>Use Case Name</b>	รายการอาหารที่ต้องทำ	
<b>Actor</b>	ผู้ประกอบอาหาร	
<b>Description</b>	ดูรายการอาหารที่ต้องทำ และปรับสถานะรายการอาหารที่สั่ง	
<b>Precondition</b>	อยู่ในหน้ารายการอาหารที่ต้องทำ (หน้าหลัก)	
<b>Main Flow</b>	ผู้ประกอบอาหาร	ระบบ
	2. ดูรายการอาหารที่ต้องทำ และ กดอัปเดตสถานะรายการอาหารที่ต้องการ	1. แสดงรายการอาหารที่ต้องทำ 3. อัปเดตข้อมูลรายการอาหารที่สั่ง 4. แสดงหน้ารายการอาหารที่ต้องทำ
<b>Alternate Condition</b>	-	

ตารางที่ 3.8 ตารางแสดง Use Case Description ของการสร้าง แก้ไข หรือลบเมนู

<b>Use Case ID</b>	7	
<b>Use Case Name</b>	สร้าง แก้ไข หรือลบเมนู	
<b>Actor</b>	ผู้จัดการ	
<b>Description</b>	สร้าง แก้ไข หรือลบเมนูในปัจจุบัน	
<b>Precondition</b>	อยู่ในหน้าสร้าง แก้ไข หรือลบเมนู (หน้าหลัก)	
<b>Main Flow</b>	<p>ผู้จัดการ</p> <p>1. กดสร้าง หรือเลือกเมนูที่ต้องการ แก้ไขหรือลบ</p> <p>3. แก้ไขข้อมูลแล้วบันทึก หรือลบเมนู</p>	<p>ระบบ</p> <p>2. แสดงฟอร์มข้อมูลของเมนูอาหาร</p> <p>4. บันทึกข้อมูล</p> <p>5. แสดงหน้าสร้าง แก้ไข หรือลบเมนู</p>
<b>Alternate Condition</b>	-	

### 3.5. การออกแบบฐานข้อมูล



รูปภาพที่ 3.5 ER Table

ตารางที่ 3.9 ตารางแสดง Data Dictionary: bill

bill: สำหรับจัดเก็บบิลในระบบ

Field Name	Description	Type	PK/FK	Null
id	id ของบิล	int unsigned	PK	
person	จำนวนคนมาใช้บริการ	int		
type	ชนิดการให้บริการ (DINE-IN, TAKE-OUT)	varchar (100)		
status	สถานะของบิล (OPEN, PAYMENT, CLOSE, VOID)	varchar (50)		
timestamp	เวลาเปิดบิล	timestamp		

ตารางที่ 3.10 ตารางแสดง Data Dictionary: index\_category

index\_category: สำหรับเก็บหมวดหมู่ของ Item เพื่อเรียงลำดับ

Field Name	Description	Type	PK/FK	Null
category	หมวดหมู่ของ Item	varchar (100)	PK	
number	เรียงลำดับ	int		

ตารางที่ 3.11 ตารางแสดง Data Dictionary: item

item: สำหรับจัดเก็บเมนูอาหาร และตัวเลือก

Field Name	Description	Type	PK/FK	Null
id	id ของอาหารหรือตัวเลือก	int unsigned	PK	
name	ชื่อ	varchar (150)		
description	รายละเอียด	varchar (500)		Yes
category	หมวดหมู่ (OPTION = ตัวเลือก)	varchar (100)		
image	สั้นทางที่เก็บไฟล์	text		Yes
price	ราคา	float		
quantity	จำนวน	int		
check_quantity	ต้องเช็คจำนวนหรือไม่	tinyint (1)		
display	แสดงรายการหรือไม่	tinyint (1)		
version	เลขเวอร์ชันของข้อมูล	bigint(20)		
flag	สถานะของข้อมูล (0 = NORMAL, -1 = DELETE)	tinyint (1)		

ตารางที่ 3.12 ตารางแสดง Data Dictionary: item\_optional

item\_optional: สำหรับจัดเก็บเมนูอาหารมีกลุ่มตัวเลือกอะไร

Field Name	Description	Type	PK/FK	Null
item_id	id ของอาหาร	int unsigned	PK/FK	
optional_id	id ของกลุ่มตัวเลือก	int unsigned	PK/FK	
number	เรียงลำดับ	int		

ตารางที่ 3.13 ตารางแสดง Data Dictionary: optional

optional: สำหรับจัดเก็บกลุ่มตัวเลือก

Field Name	Description	Type	PK/FK	Null
id	id ของกลุ่มตัวเลือก	int unsigned	PK	
name	ชื่อกลุ่มตัวเลือก	int unsigned		
description	รายละเอียด	int		
min	เลือกได้ต่ำสุด	int		
max	เลือกได้สูงสุด	int		
flag	สถานะของข้อมูล (0 = NORMAL, -1 = DELETE)	tinyint (1)		

ตารางที่ 3.14 ตารางแสดง Data Dictionary: optional\_item

optional\_item: สำหรับจัดเก็บกลุ่มตัวเลือกมีตัวเลือกอะไร

Field Name	Description	Type	PK/FK	Null
optional_id	id ของกลุ่มตัวเลือก	int unsigned	PK/FK	
item_id	id ของตัวเลือก	int unsigned	PK/FK	
number	เรียงลำดับ	int		

ตารางที่ 3.15 ตารางแสดง Data Dictionary: order\_info

order\_info: สำหรับจัดเก็บรายการอาหารที่สั่ง

Field Name	Description	Type	PK/FK	Null
id	id ของรายการอาหาร	int unsigned	PK	
bill_id	id ของบิล	int unsigned	FK	
item_id	id ของอาหาร	int unsigned	FK	
quantity	จำนวน	int		
comment	หมายเหตุ	varchar (500)		Yes
status	สถานะของรายการ (CANCEL, PENDING, ORDER, COOKING, SERVING, COMPLETE)	varchar (50)		
timestamp	เวลาที่สั่ง	timestamp		

ตารางที่ 3.16 ตารางแสดง Data Dictionary: payment

payment: สำหรับจัดเก็บรายละเอียดการชำระเงิน และใบเสร็จ

Field Name	Description	Type	PK/FK	Null
id	id ของการชำระเงิน	int unsigned	PK	
channel	ช่องทางการชำระเงิน (CASH, QR CODE, DEEPLINK)	varchar (50)		
ref1	เลขอ้างอิง 1	varchar (20)		
ref2	เลขอ้างอิง 2	varchar (20)		Yes
ref3	เลขอ้างอิง 3	varchar (20)		Yes
status	สถานะของการชำระเงิน (PAID, UNPAID)	varchar (50)		
generate_info	ข้อมูลที่สร้างขึ้นสำหรับการชำระเงิน	longtext		Yes
confirm_info	ข้อมูลสำหรับยืนยันการชำระเงิน	longtext		Yes
created_timestamp	เวลาที่สร้างการชำระเงิน	timestamp		
updated_timestamp	เวลาอัปเดตสถานะการชำระเงินล่าสุด	timestamp		
bill_id	id ของบิล	int unsigned	FK	
total	ราคา	float		
receipt	ใบเสร็จ	longtext		Yes

### ตารางที่ 3.17 ตารางแสดง Data Dictionary: select\_item

select\_item: สำหรับจัดเก็บรายการอาหารที่สั่ง ว่ามีการเลือกตัวเลือกจากกลุ่มตัวเลือก

Field Name	Description	Type	PK/FK	Null
order_id	id ของรายการอาหาร	int unsigned	PK/FK	
optional_id	id ของกลุ่มตัวเลือก	int unsigned	FK	
item_id	id ของตัวเลือก	int unsigned	PK/FK	

### ตารางที่ 3.18 ตารางแสดง Data Dictionary: table\_info

table\_info: สำหรับจัดเก็บสถานะตัวภายนอกในร้าน

Field Name	Description	Type	PK/FK	Null
id	id หมายเลขโต๊ะ	int unsigned	PK	
available	โถ่วร์ร้อมให้บริการหรือไม่	tinyint (1)		
bill_id	id ของบิล	int unsigned	FK	Yes

### ตารางที่ 3.19 ตารางแสดง Data Dictionary: token

token: สำหรับจัดเก็บเลขอ้างอิงสู่เพื่อใช้งานระบบลูกค้า

Field Name	Description	Type	PK/FK	Null
id	uid สุ่ม	varchar (32)	PK	
bill_id	id ของบิล	int unsigned	FK	

## สรุป

การนำเสนอตั้งที่กล่าวมาข้างต้นทำให้เห็นภาพรวมของระบบมากยิ่งขึ้น ง่ายต่อการทำความเข้าใจ โดยเฉพาะในส่วนที่มีการแยกสิทธิ์การใช้งานตามหน้าที่ที่แตกต่างกันและการเก็บตัวเลือกของแต่ละรายการอาหารบนฐานข้อมูล

## บทที่ 4

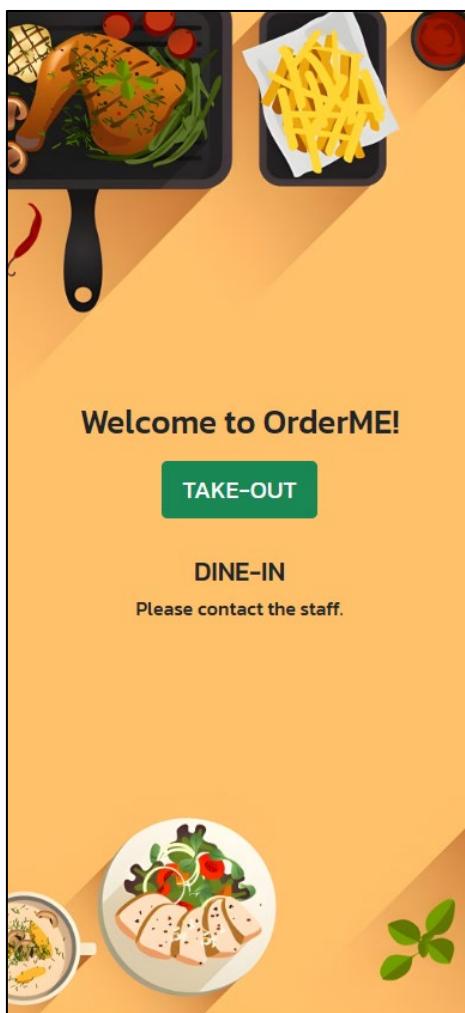
### การพัฒนาระบบ

ในบทนี้จะกล่าวถึงการพัฒนาระบบของเว็บแอปพลิเคชันในแต่ละส่วน โดยมีรายละเอียดการใช้งานหน้าเว็บเบื้องต้น และเบื้องหลังการทำงานของระบบต่าง ๆ ของเว็บแอปพลิเคชันที่พัฒนาโดยสังเขป

#### 4.1. ส่วนของลูกค้า (USER)

##### 4.1.1. หน้าแรกของเว็บไซต์

สำหรับการสั่งอาหารแบบ TAKE-OUT ให้ลูกค้าสามารถเริ่มสั่งอาหารได้โดยการกดปุ่ม TAKE-OUT แต่ถ้าหากต้องการสั่งอาหารสำหรับรับประทานภายในร้านแบบ DINE-IN ให้ทำการติดต่อพนักงานเพื่อทำการขอรับ QR Code สำหรับการสั่งอาหาร



รูปภาพที่ 4.1 หน้าแรกของเว็บไซต์

เมื่อทำการกดปุ่ม TAKE-OUT หรือสแกน QR Code สำหรับ DINE-IN ที่ได้รับจากพนักงานสำหรับรับประทานอาหารที่ร้าน ระบบจะทำการแจกคุกกี้เพื่อทำการระบุลูกค้า และนำทางลูกค้าไปยังหน้าสำหรับการสั่งอาหาร

```
<-----TAKE-OUT----->
<div class="my-1">
  <a type="button" class="btn btn-success btn-lg" href="/take-out">
    TAKE-OUT
  </a>
</div>
```

```
@GetMapping("/take-out")
public String takeOut(HttpServletRequest request, HttpServletResponse response) {
    String uid = generateNewToken();
    Cookie cookie1 = CookieUtil.createCookie("uid", uid, 30 * 60, false,
true, "/", request.getServerName());
    Cookie cookie2 = CookieUtil.createCookie("type",
ConstantUtil.TAKE_OUT, 30 * 60, false, true, "/",
request.getServerName());
    response.addCookie(cookie1);
    response.addCookie(cookie2);

    tokenService.newToken(uid);
    tokenService.autoDeleteToken(uid, 30);
    return "redirect:/main-menu";
}

@GetMapping("/dine-in/{uid}")
public String dineIn(@PathVariable String uid, HttpServletRequest request, HttpServletResponse response) {
    if(!tokenService.tokenIsPresent(uid)) return "redirect:/";
    Cookie cookie1 = CookieUtil.createCookie("uid", uid, 5 * 60 * 60,
false, true, "/", request.getServerName());
    Cookie cookie2 = CookieUtil.createCookie("type",
ConstantUtil.DINE_IN, 5 * 60 * 60, false, true, "/",
request.getServerName());
    response.addCookie(cookie1);
    response.addCookie(cookie2);
    return "redirect:/main-menu";
}
```

#### 4.1.2. หน้าเมนูสำหรับเลือกอาหารลงตะกร้า

เมนูอาหารจะจัดเรียงตามหมวดหมู่ สามารถกดหมวดหมู่อาหารที่ต้องการได้ในแถบนำทาง เพื่อนำทางไปที่อาหารหมวดหมู่นั้น

The screenshot shows the OrderME app interface with the following details:

- Header:** OrderME, Cart, Bill
- Top Navigation:** Bingchoo อ่อนก็อป, มัคจะนมสด 22 อ่อนซ์, ช็อกโกแลตวีบ
- Section 1: Bingchoo อ่อนก็อป**
  - Item 1:** บมสุดบิงชชาไทย
    - Image: A white cup filled with a light-colored drink topped with brown crumble.
    - Description: บมสุดแก้สุดละมุน ก็อปด้วยเกลี้ดน้ำแข็งบิงชูรสชาไทย เป็นขันเป็นเอกลักษณ์ เนื้อ...
    - Price: ₩ 100
  - Item 2:** รอยัลชาไทยบิงชชา
    - Image: An orange cup filled with a light-colored drink topped with brown crumble.
    - Description: รอยัลชาไทยรสชาติเป็นขันบุบบวลด กอง กองชาไทยอันเป็นเอกลักษณ์ อร่อยคุณ...
    - Price: ₩ 100
- Section 2: มัคจะนมสด 22 อ่อนซ์**
  - Item 1:** มัคจะนมสด (ชาเขียว บมสุด)
    - Image: A green cup filled with a light-colored drink.
    - Description: เมบูสุดอิตของพุก รสชาติเป็นเอกลักษณ์ กองเด่นไม่เหมือนใคร มัคจะแท้คุณภาพดี...
    - Price: ₩ 80
  - Item 2:** มัคจะใส่ชานม (ชาเขียวใส่ชานม)
    - Image: A green cup filled with a light-colored drink.
    - Description: การรวมตัวของสองเมบูขายดีตลอดกาล กั้งมัคจะและชานม รวมมกันเป็นรสชาติ...
    - Price: ₩ 85

รูปภาพที่ 4.2 หน้าแสดงเมนู

หน้าเว็บจะทำการดึงหมวดหมู่ของเมนูอาหารทั้งหมดออกมานะ หลังจากนั้นจึงเรียกเมนูอาหารของแต่ละหมวดหมู่ออกมาแสดงผลบนหน้าเว็บ

```

function getCategoryData() {
    return new Promise(function (resolve, reject) {
        $.get( '/api/item/list-category', function( data ) {
            resolve(data)
        });
    })
}

function getItemEachCategoryData(category) {
    return new Promise(function (resolve, reject) {
        $.get( '/api/item/category/' + category, function( data ) {
            resolve(data)
        });
    })
}

let allCategory = await getCategoryData()
for(let i=0; i < allCategory.length;i++) {
    let category = allCategory[i]
    createNavCategory(category)
    item = await getItemEachCategoryData(category)
    await createCard(category)
}

```

```

@GetMapping("/list-category")
public List<String> findAllFoodCategory() {
    return itemService.findAllFoodCategory();
}

@JsonView(Views.Overall.class)
@GetMapping("/category/{category}")
public List<Item> findItemByCategory(@PathVariable String category) {
    return itemService.findItemByCategory(category);
}

```

เมื่อกดแต่ละเมนูสามารถเลือกตัวเลือก ใส่ comment เพิ่มเติม และระบุจำนวนได้ จากนั้นกดปุ่ม Get In Cart เพื่อเลือกอาหารใส่ตะกร้า



รูปภาพที่ 4.3 หน้าแสดงข้อมูลเมนูและเลือกเมนู

เมื่อลูกค้ากดดูเมนูอาหาร จะทำการเรียกใช้ API เพื่อนำข้อมูลต่าง ๆ เกี่ยวกับเมนูนั้นมาแสดงในรูปแบบของ Modal ระบบจะทำการดึงข้อมูลรายละเอียดของอาหาร รวมถึงตัวเลือกต่าง ๆ ออกมาราบบกัน

```

function getItemData(id) {
    return new Promise( function (resolve, reject) {
        $.get( '/api/item/' + id ,function( data ) {
            resolve(data)
        }) ;
    })
}

async function createModal(id) {

    idItem = id
    allItem = await getItemData(id)
    startPrice = allItem.price
    if(allItem.checkQuantity){
        maxQty = allItem.quantity
    }

    $('#inputQuantity').val(1)
    $('#titleItem').text(allItem.name)
    $('#desItem').text(allItem.description)
    $('#priceItem').text('฿ ' + startPrice)
    $('#modalBody').empty()
    $('#inputComment').val(null)
    let optionData = allItem.optionalList
    createOption(optionData)

    if (allItem.image != null){
        document.getElementById('imgModal').src =
        '/images/'+allItem.image
    }

    document.getElementById('imgModal').setAttribute('onerror',"this.src='/
    images/default.png'")
    }
    else {document.getElementById('imgModal').src =
    '/images/default.png'}
}

```

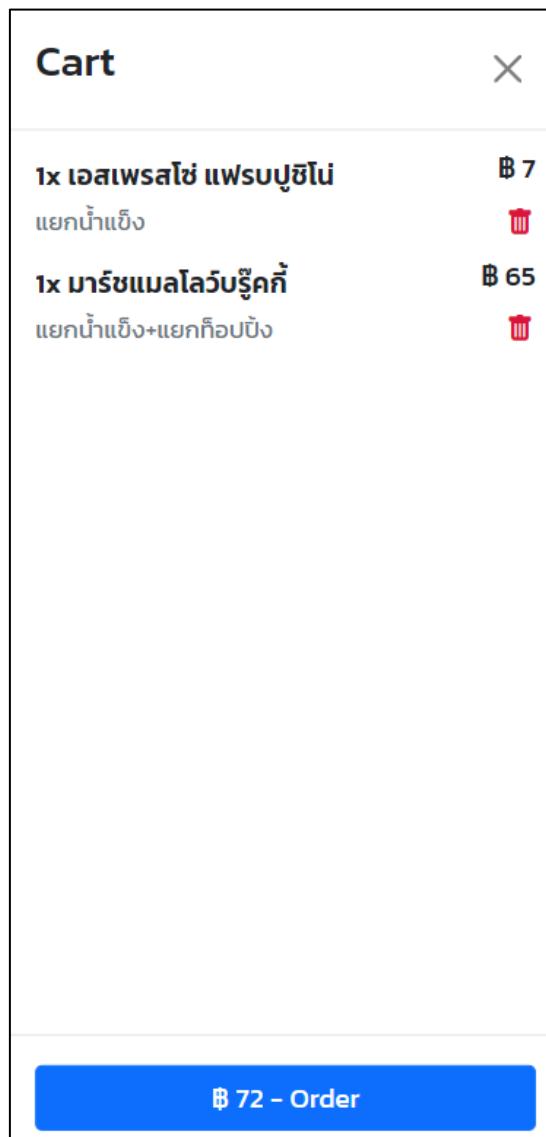
```

@ JsonView (Views.Detail.class)
@.GetMapping("/{id}")
public Item findById(@PathVariable int id) {
    return itemService.findById(id);
}

```

#### 4.1.3. การสั่งอาหารที่เลือกไว้ในตะกร้าสำหรับลูกค้ารับประทานที่ร้าน (DINE-IN)

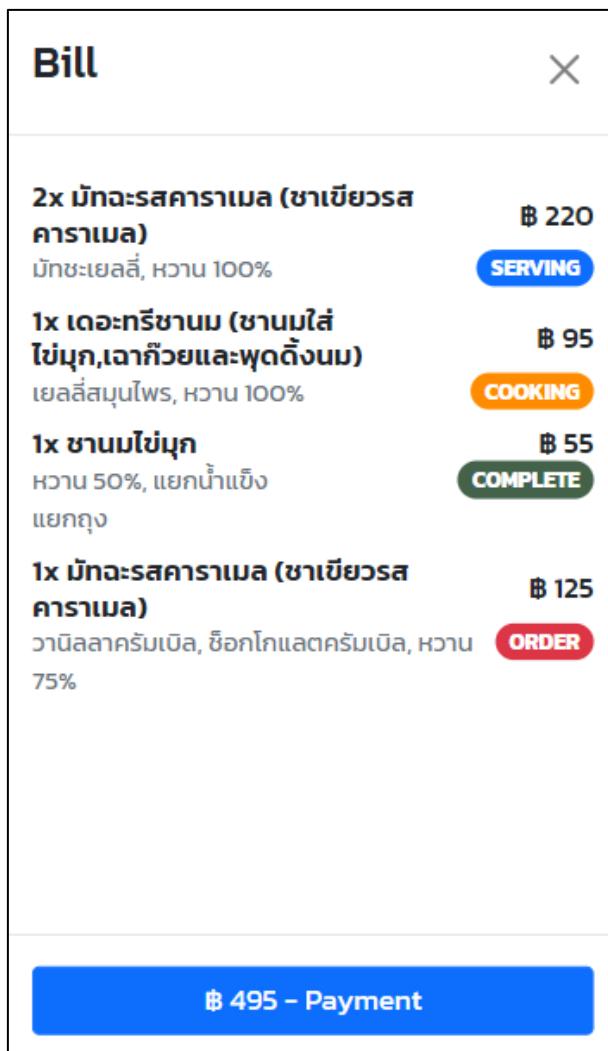
หลังจากลูกค้าได้ทำการเลือกอาหารใส่ภายในตะกร้าเรียบร้อยแล้ว ลูกค้าสามารถดูรายการอาหารที่ใส่ลงไปในตะกร้าได้โดยการกดปุ่ม Cart และสามารถทำการสั่งอาหารได้โดยการกดปุ่ม Order ภายใต้หน้าจอแสดงผลของ Cart เพื่อทำการสั่งอาหาร



รูปภาพที่ 4.4 หน้าแสดงตะกร้าเมนูที่เลือก (DINE-IN)

เมื่อสั่งอาหารแล้วเรียบร้อยแล้ว ลูกค้าสามารถดูรายละเอียดรายการอาหารที่สั่งได้ด้วยการกดปุ่ม Bill โดยจะมีรายละเอียดรายการอาหารที่สั่งทั้งหมดพร้อมกับสถานะของอาหารในปัจจุบัน และลูกค้าสามารถสั่งอาหารเพิ่มได้จนกว่าจะทำการชำระเงิน

เมื่อลูกค้าต้องการชำระเงิน สามารถทำได้โดยการกดปุ่ม Payment ภายใต้หน้าจอแสดงผลของ Bill ระบบจะนำทางไปยังหน้าสำหรับเลือกช่องทางการชำระเงิน



รูปภาพที่ 4.5 หน้าแสดงบิลและสถานะอาหาร

เมื่อลูกค้ากดดูบิล ระบบจะทำการดึงข้อมูลเกี่ยวกับบิลรวมถึงรายการอาหารที่สั่งทั้งหมด โดยจะมีรายละเอียดของอาหารที่สั่งพร้อมกับสถานะของอาหารในปัจจุบัน

```

function getBillData() {
    return new Promise(function (resolve, reject) {
        $.get('/api/bill/my-bill', function (data) {
            resolve(data)
        })
    })
}

async function createBillItem() {
    $('#BillModalBody').empty()

    let json = await getBillData()

    let orders = json.orders

    if(json.orders != null){
        if(orders.length > 0 ){
            $('#billBtn').prop('disabled', false)
        }
        else {
            $('#billBtn').prop('disabled', true)
        }

        const template = ` ${orders.map(order => `
<div>
    <div class="name-price">
        <h6 class="menu-name"> ${order.quantity}x ${order.name}</h6>
        <h6>฿ ${order.amount}</h6>
    </div>
    <div class="name-price-status">
        <div class="columnFlex">
            <small class="text-muted">${order.option}</small>
            <br>
            <small class="text-muted">${order.comment}</small>
        </div>
        <span class="badge rounded-pill
${order.status}">${order.status}</span>
    </div>
` ).join('') }
    ;
    $('#BillModalBody').append(template)
    $('#billBtnText').text('฿ '+json.subTotal)
    }
}

```

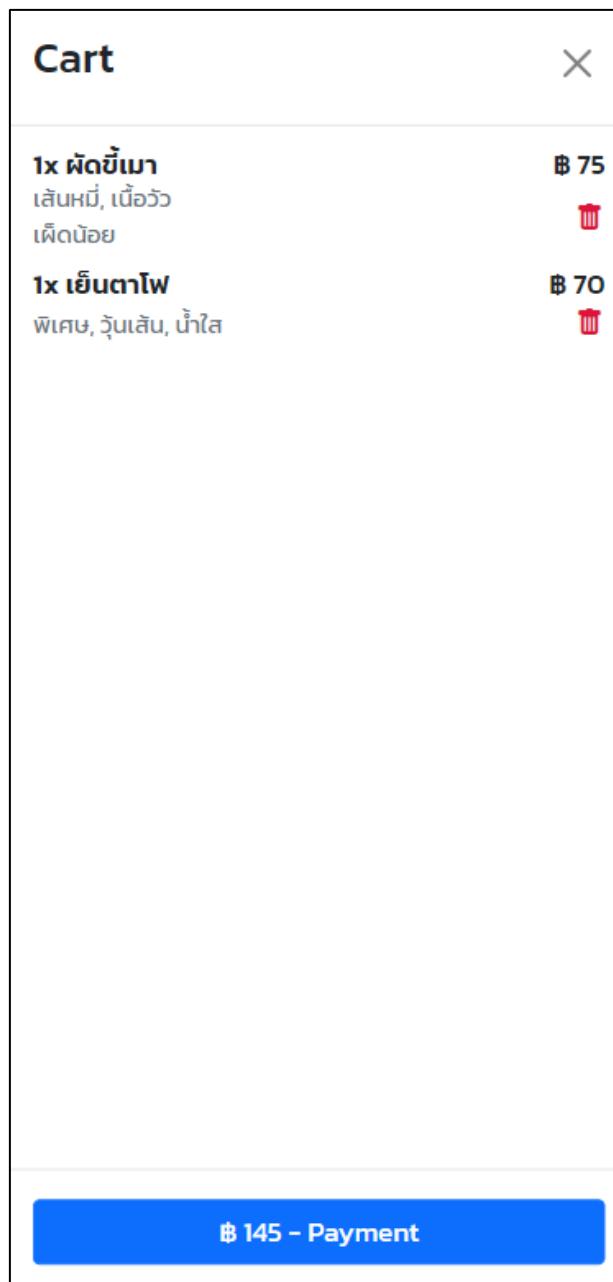
```

@GetMapping("/my-bill")
public BillDTO getBillDTO(@CookieValue(name = "uid") String uid) {
    Token token = tokenService.findById(uid);
    if(token == null) return null;
    Bill bill = token.getBill();
    if(bill == null) return null;
    return billService.getBillDTO(bill.getId());
}

```

#### 4.1.4. การสั่งอาหารที่เลือกไว้ในตะกร้าสำหรับลูกค้ารับกลับบ้าน (TAKE-OUT)

การสั่งอาหารสำหรับลูกค้า TAKE-OUT จะสามารถสั่งอาหารได้โดยการนำอาหารที่ต้องการสั่งใส่ตะกร้า และหลังจากเลือกรายการอาหารที่ต้องการเสร็จสิ้นให้ทำการกดปุ่ม Payment ภายในหน้าจอแสดงผลของ Cart เพื่อทำการชำระเงิน ระบบจะนำทางไปยังหน้าสำหรับเลือกช่องทางการชำระเงิน และการประกอบอาหารจะเริ่มต้นขึ้นเมื่อมีการชำระเงินสำเร็จแล้วเท่านั้น



รูปภาพที่ 4.6 หน้าแสดงตะกร้าเมนูที่เลือก (TAKE-OUT)

เมื่อเปิดหน้าต่างกร้าจะนำอาหารที่เลือกไว้ทั้งหมดมาแสดง ถ้าเปิดโดยที่ยังไม่เคยเลือกอาหารจะไม่สามารถซื้อระเงินได้

```

function createCartItem() {
    let total = 0

    $('#cartModalBody').empty()

    if(cartText.length > 0) {
        $('#cartBtn').prop('disabled', false)
    }
    else {
        $('#cartBtn').prop('disabled', true)
        $('#footer').hide()
    }

    for(let i=0;i<cartText.length;i++) {
        let index = i

        let template = `


<div class="name-price" data-bs-toggle="modal"
        data-bs-target="#staticBackdrop" onclick="editItem(${index})">
        <h6 class="menu-name">
            ${orderRequests[i].quantity}x ${cartText[i].name}
        </h6>
        <h6>฿ ${cartText[i].price}</h6>
    </div>

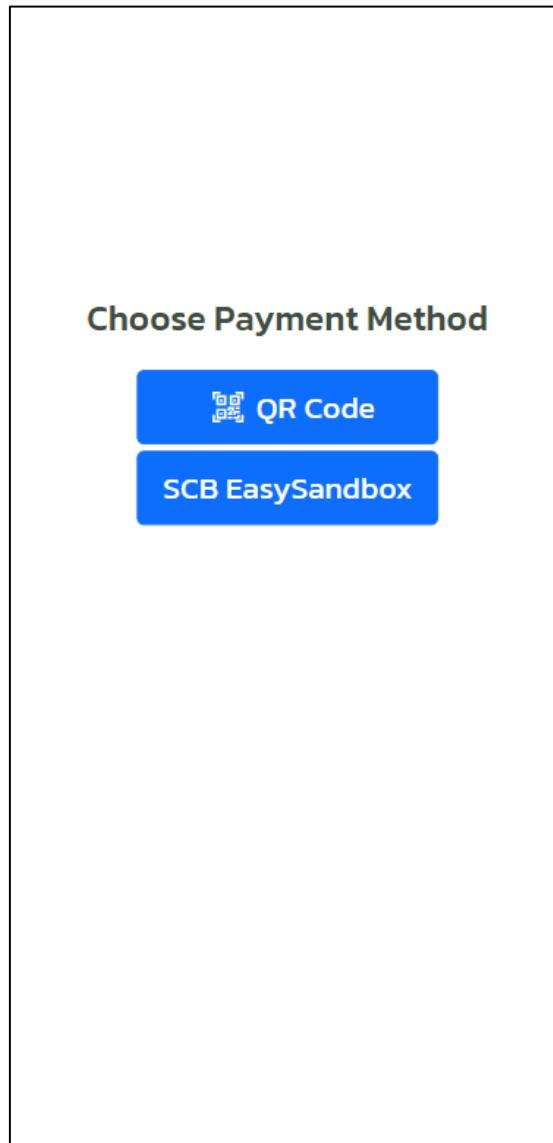
    <div class="name-price">
        <div class="columnFlex">
            <small class="text-muted">
                ${cartText[i].opText.join(', ')}</small>
            <small class="text-muted">
                ${orderRequests[i].comment}</small>
            </div>
            <button class="btn btn-delete-cart"
                type="button" onclick="deleteItem()">
                <i class="bi bi-trash-fill"></i></button>
        </div>
    </div>
    `

    $('#cartModalBody').append(template)
}
$('#cartBtnText').text('฿ '+total)
}


```

#### 4.1.5. การชำระเงิน

เมื่อลูกค้ากดปุ่ม Payment จากหน้า Bill หรือ Cart ระบบจะนำทางมายังหน้าสำหรับเลือกช่องทางการชำระเงิน โดยลูกค้าสามารถเลือกช่องทางการชำระเงินได้ทั้งหมด 3 ช่องทาง ได้แก่ QR Code, SCB Easy Sandbox และเงินสด โดยกรณีเงินสดต้องทำการติดต่อพนักงานที่เคชเชียร์



รูปภาพที่ 4.7 หน้าเลือกช่องทางการชำระเงิน

เมื่อเปิดหน้าเลือกสำหรับช่องทางการชำระเงินแต่ละปั๊มจะพาไปยัง URL ที่แตกต่างกัน สำหรับให้ลูกค้าทำการชำระเงิน

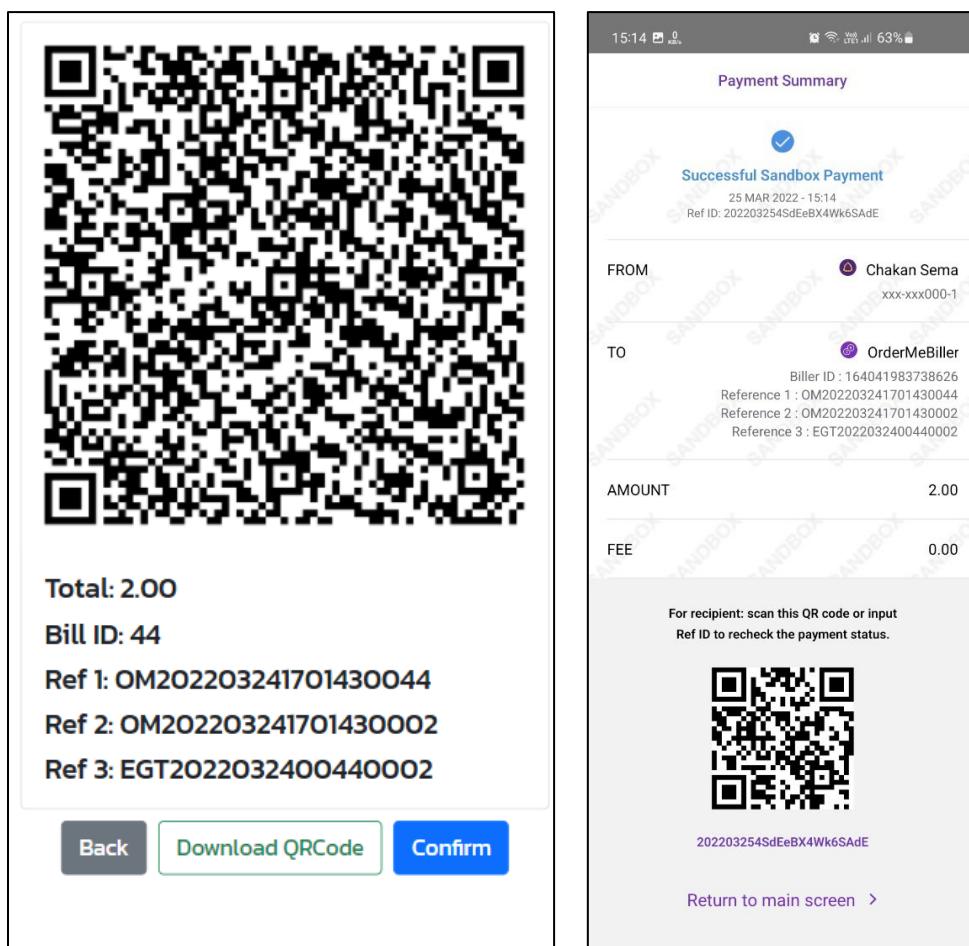
```
<div class="my-1">
    <h5>Choose Payment Method</h5>
</div>
<div class="my-1">
    <a type="button" class="btn btn-primary"
        href="/payment/qrcode">QR Code</a>
</div>
<div class="my-1">
    <a type="button" class="btn btn-primary"
        href= "/payment/deeplink">SCB EasySandbox</a>
</div>
<div class="my-1">
    <button type="button" class="btn btn-primary">Cash</button>
</div>
```

```
@GetMapping
public String payment(@CookieValue(name = "uid") String uid, Model model) {
    Token token = tokenService.findById(uid);
    if(token != null) {
        Bill bill = token.getBill();
        if(bill == null) {
            return "redirect:/main-menu";
        }
        else if(bill.getStatus().equalsIgnoreCase(ConstantUtil.OPEN)) {
            billService.setStatusPaymentBill(bill.getId());
            model.addAttribute("bill",
billService.getBillDTO(bill.getId()));
            return "payment";
        }
        else if(bill.getStatus().equalsIgnoreCase(ConstantUtil.PAYMENT)) {
            model.addAttribute("bill",
billService.getBillDTO(bill.getId()));
            return "payment";
        }
        else if(bill.getStatus().equalsIgnoreCase(ConstantUtil.CLOSE)) {
            Payment payment =
paymentRepository.findByBill_Id(bill.getId());
            if(payment.getStatus().equalsIgnoreCase(ConstantUtil.PAID))
{
                return "redirect:/receipt/"+payment.getRef1();
            }
        }
    }
    return "redirect:/";
}
```

### 1) การชำระเงินด้วย QR Code

QR Code สำหรับการชำระเงินที่แสดงผลอยู่คือชนิด Thai QR Code Tag 30 (QR 30) และ QR Card Scheme (QR CS) รวมอยู่ภายใน QR Code เดียวกัน ทำให้สามารถรองรับการชำระเงินผ่านแอปพลิเคชัน Mobile Banking ได้ด้วยการสแกน QR Code ที่ปรากฏ และเลือกช่องทางการชำระเงินผ่านบัญชีธนาคาร หรือชำระเงินผ่านบัตรเครดิต (VISA หรือ Mastercard)

การชำระเงินสามารถทำได้โดยการบันทึกรูป QR Code ลงภายในเครื่อง หรือสแกน QR Code เพื่อชำระเงินผ่านแอปพลิเคชัน Mobile Banking ที่ต้องการ เมื่อชำระเงินแล้วให้ทำการกดปุ่ม Confirm เพื่อตรวจสอบการชำระเงิน หากการชำระเงินสำเร็จระบบจะทำการนำทางเปลี่ยนไปยังหน้าใบเสร็จ



รูปภาพที่ 4.8 หน้าแสดง QR Code สำหรับชำระเงิน และการชำระเงินผ่าน QR Code

\* หมายเหตุ: เนื่องจากเป็นระบบทดสอบการชำระเงิน จึงยังไม่สามารถรับการชำระเงินได้จริง สามารถทดสอบการชำระเงินด้วย QR Code ได้ผ่าน SCB Easy Sandbox เท่านั้น \*

รายละเอียดเพิ่มเติม: <https://developer.scb/#/documents/documentation/qr-payment/thai-qr.html>

เมื่อเข้าสู่หน้าชำระเงินด้วย QR Code ระบบจะนำค่า Attribute ที่ได้รับจาก Model Attributes ของ Thymeleaf ในส่วนของ Controller Class ที่ช่วยให้สามารถเข้าถึง Spring MVC Data ได้ เพื่อทำการเตรียมข้อมูลอักษรมาแสดงผลภายใน View ของหน้าเว็บไซต์ มาแสดงผลเป็นข้อมูลต่าง ๆ โดยข้อมูล QR Code ที่ได้รับจะอยู่ในรูปแบบของ Base64 ที่สามารถนำมารอครหัสเป็นรูปภาพได้

```
<div class="card m-2" id="detailPayment">
    
    <div class="card-body">
        <h5 class="card-text" >Total:</h5>
        <p style="display: inline" th:text="${total}"></p></h5>
        <h5 class="card-text" >Bill ID:</h5>
        <p style="display: inline" th:text="${billId}"></p></h5>
        <h5 class="card-text" >Ref 1:</h5>
        <p style="display: inline" th:text="${ref1}"></p></h5>
        <h5 class="card-text" >Ref 2:</h5>
        <p style="display: inline" th:text="${ref2}"></p></h5>
        <h5 class="card-text" >Ref 3:</h5>
        <p style="display: inline" th:text="${ref3}"></p></h5>
    </div>
</div>
<div class="px-3" style="display: flex;
justify-content: center; width: 100%">
    <a class="btn btn-secondary mx-2"
        href="/payment" type="button">Back</a>
    <a class="btn btn-outline-success"
        type="button" onclick="takeshot()">Download QRCode</a>
    <a class="btn btn-primary mx-2"
        type="button" href="/payment/qrcode/success">Confirm</a>
</div>
```

ในส่วนของการดาวน์โหลดรูปภาพจะนำ Library ชื่อ html2canvas มาใช้เพื่อบันทึก HTML ในส่วนที่ต้องการให้กลายเป็นรูปภาพและใช้รหัสอ้างอิงการชำระเงิน (Ref 1) ในการตั้งชื่อไฟล์รูป

```

function takeshot() {
    let div = document.getElementById('detailPayment');
    let fileName = "[${ref1}]"
    html2canvas(div).then(
        function (canvas) {
            var a = document.createElement('a');
            a.href = canvas.toDataURL("image/jpeg")
            a.download = fileName + '.jpg';
            a.click();
        } ) }
}

```

```

@GetMapping("/qrcode")
public String generateQrCode(@CookieValue(name = "uid") String uid,
Model model) {
    Token token = tokenService.findById(uid);
    if(token != null) {
        Bill bill = token.getBill();
        if(bill == null) {
            return "redirect:/main-menu";
        }
        else if(bill.getStatus().equalsIgnoreCase(ConstantUtil.PAYMENT))
    {
        Payment payment =
paymentRepository.findByBill_Id(bill.getId());

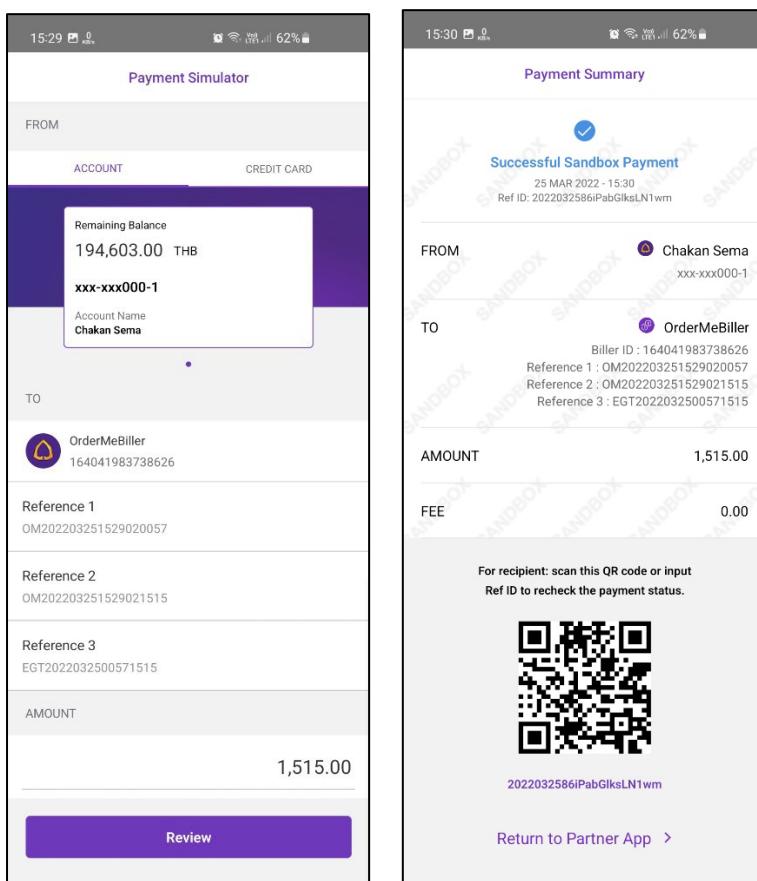
        if(payment == null ||
!payment.getChannel().equalsIgnoreCase(ConstantUtil.QR_CODE)) {
            payment = paymentService.payBill(bill.getId());
            String response =
scbSimulatorPaymentService.generateQrCode(payment);
            String qrImage =
scbSimulatorPaymentService.extractQrImageFromResponse(response);
            model.addAttribute("qrImage", qrImage);
        }
        else {
            String qrImage =
scbSimulatorPaymentService.extractQrImageFromResponse(payment.getGenerat
eInfo());
            model.addAttribute("qrImage", qrImage);
        }

        model.addAttribute("billId", payment.getBill().getId());
        model.addAttribute("ref1", payment.getRef1());
        model.addAttribute("ref2", payment.getRef2());
        model.addAttribute("ref3", payment.getRef3());
        model.addAttribute("total", df.format(payment.getTotal()));
        return "payment_qrcode";
    }
        else if(bill.getStatus().equalsIgnoreCase(ConstantUtil.CLOSE)) {
            Payment payment =
paymentRepository.findByBill_Id(bill.getId());
            if(payment.getStatus().equalsIgnoreCase(ConstantUtil.PAID))
{
                return "redirect:/receipt/"+payment.getRef1();
            }
        }
    }
    return "redirect:/";
}

```

## 2) การชำระเงินด้วยแอปพลิเคชัน SCB Easy Sandbox

SCB Easy Sandbox สำหรับจำลองการชำระเงินผ่าน SCB Easy หลังจากทำการกดปุ่มที่หน้าเว็บระบบจะนำลูกค้ามายัง SCB Easy Sandbox เพื่อทำการชำระเงิน โดยรองรับช่องทางการชำระเงินผ่านบัญชีธนาคาร Current or savings account (CASA) หรือชำระเงินผ่านบัตรเครดิตเต็มจำนวนเงิน Credit Card with full amount payment (CCFA) และเมื่อลูกค้าชำระเงินเสร็จ SCB Easy Sandbox จะนำลูกค้ากลับมายังหน้าใบเสร็จของเรียบ



รูปภาพที่ 4.9 การชำระเงินผ่าน SCB Easy Sandbox

\* หมายเหตุ: เนื่องจากเป็นระบบทดสอบการชำระเงิน จึงยังไม่สามารถรองรับการชำระเงินได้จริง สามารถทดสอบการชำระเงินด้วย Deeplink URL ได้ผ่าน SCB Easy Sandbox เท่านั้น \*

รายละเอียดเพิ่มเติม: <https://developer.scb/#/documents/documentation/pay-with-scb/payment-via-scb.html>

```

@GetMapping("/deeplink")
public RedirectView generateDeeplink(@CookieValue(name = "uid") String
uid) {
    RedirectView redirectView = new RedirectView();
    Token token = tokenService.findById(uid);
    if(token != null) {
        Bill bill = token.getBill();
        if(bill == null) {
            redirectView.setUrl("/main-menu");
            return redirectView;
        }
        else if(bill.getStatus().equalsIgnoreCase(ConstantUtil.PAYMENT)) {
            Payment payment =
paymentRepository.findByBill_Id(bill.getId());
            if(payment == null ||
!payment.getChannel().equalsIgnoreCase(ConstantUtil.DEEP_LINK)) {
                payment = paymentService.payBill(bill.getId());
            }
            redirectView.setUrl(scbSimulatorPaymentService.generateDeeplink(payment));
        }
        else {
            redirectView.setUrl(scbSimulatorPaymentService.extractDeepLinkFromRespon
se(payment.getGenerateInfo()));
        }
        return redirectView;
    }
    else if(bill.getStatus().equalsIgnoreCase(ConstantUtil.CLOSE)) {
        Payment payment =
paymentRepository.findByBill_Id(bill.getId());
        if(payment.getStatus().equalsIgnoreCase(ConstantUtil.PAID)) {
            redirectView.setUrl("/receipt/" + payment.getRef1());
            return redirectView;
        }
    }
    redirectView.setUrl("/");
    return redirectView;
}

```

### 3) การชำระด้วยเงินสด

ติดต่อพนักงานเพื่อทำการชำระเงินด้วยเงินสดผ่านเคชเชียร์

#### 4.1.6. หน้าแสดงใบเสร็จ

เมื่อการชำระเงินสำเร็จระบบจะนำลูกค้ามายังหน้าแสดงใบเสร็จซึ่งมีรายละเอียดดังนี้

- วันและเวลาที่ทำการชำระเงิน
- ประเภทและจำนวนของลูกค้า
- หมายเลขบิล
- รายการอาหารที่สั่งทั้งหมดพร้อมราคาน้ำหนึ่ง
- รหัสอ้างอิงการชำระเงิน (Ref 1) และช่องทางการชำระเงิน

<b>OrderME</b>	
Date: 08/04/2022 06:20:04	
Type: DINE-IN	Guests: 2
Bill: 14	
<hr/>	
<b>1x เย็นตาโฟ</b> เส้นไส้หมู, ต้มยำแห้ง	฿ 60.0
<b>1x ข้าวกะเพรา</b> ไก่, ไข่ดาว, พิเศษ	฿ 80.0
<b>1x ชาบู</b> เจากี้ย, ซีอิ๊วและครัมเบิล, หวาน 75%	฿ 70.0
<b>1x มักละนэмสด (ชาเยี่ยวนมสด)</b> เยลลี่สับปะรด, หวาน 100%, แยกก๊อกปั้ง	฿ 100.0
<b>1x บะหมี่กุ้งเผา</b> ถุงเก็บความเย็นและน้ำแข็งแห้ง	฿ 110.0
<b>1x ข้าวไข่เจียว</b> หมูสับ, พิเศษ	฿ 70.0
<b>1x เดอะกรีชาบู (ชาบูไม่ใส่ไข่บุก, เจากี้ยและพุดดิ้งนม)</b> หวาน 25%, บั่น	฿ 100.0
<hr/>	
<b>Total:</b>	฿ 590.0
Ref 1:	OM202204081819260014
Paid By:	CCFA
<b>THANK YOU</b>	
<b>HAVE A NICE DAY</b>	

รูปภาพที่ 4.10 ใบเสร็จ

เมื่อเข้าสู่หน้าแสดงใบเสร็จระบบจะทำการนำข้อมูล(okma)มาแสดงในหน้าเว็บไซต์ ด้วยการใช้งานคุณสมบัติ Model Attributes ของ Thymeleaf ในส่วนของ Controller Class ที่ช่วยให้สามารถเข้าถึง Spring MVC Data ได้ เพื่อทำการเตรียมข้อมูล(okma)มาแสดงผลภายใน View ของหน้าเว็บไซต์

โดยในส่วนของหน้าเว็บไซต์จะนำข้อมูลมาใช้โดยเรียกผ่านตัวแปรในรูปแบบ \${variable}

```
<h2 class="card-title">OrderME</h2>


<h6 th:text="'Date: ' + ${date}"></h6>
    <div class="name-price">
        <h6 th:text="'Type: ' + ${type}"></h6>
        <h6 th:text="'Guests: ' + ${person}"></h6>
    </div>
    <h6 th:text="'Bill: ' + ${billId}"></h6>
</div>
<hr>


```

```
@GetMapping("/receipt/{ref1}")
public String getPageReceipt(@PathVariable String ref1, Model model) {
    ReceiptDTO receiptDTO = paymentService.getReceiptDTO(ref1);
    if(receiptDTO != null) {
        DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("dd/MM/yyyy hh:mm:ss");
        String date =
formatter.format(receiptDTO.getPayment().getUpdatedTimestamp());

        model.addAttribute("billId", receiptDTO.getBill().getBillId());
        model.addAttribute("person", receiptDTO.getBill().getPerson());
        model.addAttribute("type", receiptDTO.getBill().getType());
        model.addAttribute("orders", receiptDTO.getBill().getOrders());
        model.addAttribute("ref1", receiptDTO.getPayment().getRef1());
        model.addAttribute("channel",
receiptDTO.getPayment().getChannel());
        model.addAttribute("date", date);
        model.addAttribute("total", receiptDTO.getPayment().getTotal());
        return "receipt";
    }
    return "redirect:/";
}
```

## 4.2. ส่วนของพนักงาน (STAFF)

### 4.2.1. หน้าสำหรับจัดการลูกค้า

หากเป็นลูกค้า DINE-IN จะทำการแสดงเลขโต๊ะแต่หากเป็นลูกค้า TAKE-OUT จะทำการแสดงเลขบิลในรูปแบบการ์ด โดยแต่ละการ์ดจะมีสีแสดงสถานะที่สำคัญที่สุดของบิลนั้น ๆ ดังนี้

- **สีแดง** ทุกออเดอร์ครัวยังไม่เริ่มประกอบอาหาร
- **สีเหลือง** มีออดิอร์ที่ครัวกำลังประกอบอาหาร
- **สีน้ำเงิน** มีออดิอร์ที่รอการเติร์ฟ
- **สีเขียว** ลูกค้าได้รับทุกออเดอร์ครบถ้วน

**Dine In Table**
[Index](#)

+
1
2

10

---

**Take Out Bill**

Bill ID: 7

รูปภาพที่ 4.11 หน้าจัดการภาพรวมภายในร้าน

เมื่อเปิดหน้าจัดการลูกค้าจะทำการเรียก API เกี่ยวกับข้อมูลลูกค้าแบบ DINE-IN ทั้งหมดและ TAKE-OUT ที่ชำระเงินแล้ว หรือยังไม่ชำระเงินแต่ไม่เกิน 20 นาทีหลังจากกดสั่งอาหาร เพื่อสร้างการ์ดจากนั้นจะทำการเรียก API เกี่ยวกับสีแสดงสถานะของบิลทั้งหมดเพื่อกำหนดสีให้การ์ดแต่ละใบ

```

function getDataAllTable() {
    return new Promise(function (resolve, reject) {
        $.get( '/api/table/all', function( data ) {
            resolve(data)
        });
    })
}

function getDataAllBillTakeOut() {
    return new Promise(function (resolve, reject) {
        $.get( '/api/bill/all-id-take-out-process', function( data ) {
            resolve(data)
        });
    })
}

function getDataColourInitialize(){
    return new Promise(function (resolve, reject){
        $.get( '/api/bill/all-bill-order-status', function( data ) {
            resolve(data)
        });
    })
}

$( document ).ready(async function () {
    connect()
    allTable = await getDataAllTable()
    initializeTable(allTable);

    await initializeTakeOut()
    await initializeColour()
})

```

```

@GetMapping("/all")
public List<TableDTO> getAllTableDTOList() {
    return tableService.findAllTableDTO();
}

@GetMapping("/all-id-take-out-process")
public List<Integer> getAllBillIdTakeOutOfOrderNotCancelAndComplete() {
    return
orderService.getAllBillIdTakeOutOfOrderNotCancelAndComplete();
}

@GetMapping("/all-bill-order-status")
public List<UpdateBillOrderStatusMessage> getAllCurrentBillOrderStatus()
{
    return orderService.getAllCurrentBillOrderStatus();
}

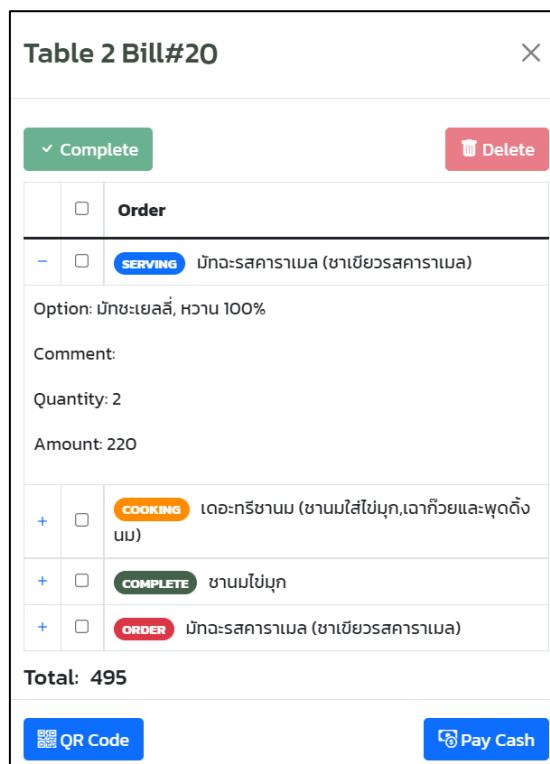
```

เมื่อกดดูรายละเอียดเกี่ยวกับแต่ละบิล ระบบจะทำการแสดงออเดอร์ทั้งหมด ซึ่งแต่ละออเดอร์จะสามารถดูรายละเอียดเพิ่มเติมได้ โดยสถานะทั้งหมดมีดังนี้

- ORDER สีแดง ลูกค้าสั่งอาหารแล้วแต่ครัวยังไม่เริ่มประกอบอาหาร
- COOKING สีเหลือง ครัวกำลังประกอบอาหาร
- SERVING สีน้ำเงิน อาหารพร้อมเสิร์ฟ
- COMPLETE สีเขียว ลูกค้าได้รับอาหารแล้ว

ในหน้าป็นักงานสามารถจัดการสิ่งต่าง ๆ ได้ดังนี้

- เลือกออเดอร์ที่ทำการเสิร์ฟสำเร็จแล้วจากนั้นกดปุ่ม Complete เพื่อเปลี่ยนสถานะให้เป็น COMPLETE หากลูกค้าชำระเงินแล้วและทุกออเดอร์มีสถานะเป็น COMPLETE ระบบจะทำการปิดโต๊ะอัตโนมัติ
- เลือกออเดอร์ที่ลูกค้าต้องการยกเลิกจากนั้นกดปุ่ม Delete เพื่อเปลี่ยนสถานะให้เป็น CANCEL
- ชำระเงินสดให้ลูกค้าโดยการกดปุ่ม Pay Cash
- เปิด QR Code ของโต๊ะนั้นอีกครั้งด้วยปุ่ม QR Code



รูปภาพที่ 4.12 หน้าจัดการรายละเอียดบิล

เมื่อเปิดหน้ารายละเอียดใบเสร็จจะทำการเรียก API เกี่ยวกับข้อมูลของบิลนั้น ๆ นำรายการที่สั่งทั้งหมดแสดงผลเป็นตาราง

```
function getDataBill(billId) {
    return new Promise(function (resolve, reject) {
        $.get( '/api/bill/' + billId, function( data ) {
            resolve(data)
        });
    })
}

async function openModalDineIn(billId, table) {
    $('.modal-footer').show()
    await createModalTable(billId)
    $('#qrBtn').prop('disabled', false)
    $('#DineInModal h3').text('Table '+table+' Bill#' + billId)
    if(allTable[index].indexOf(billId)] .paid) {
        $('.modal-footer').hide()
        $('#DineInModal h3').text
        ('Table '+table+' Bill#' + billId + ' [PAID]')
    }
}
```

```
@GetMapping("/{id}")
public BillDTO getBillDTO(@PathVariable int id) {
    return billService.getBillDTO(id);
}
```

เมื่อพนักงานต้องการเปิดໂທใหม่เพื่อให้บริการลูกค้า ให้พนักงานทำการกดปุ่ม New Table จากนั้นกรอกจำนวนลูกค้าและเลือกໂທที่ต้องการ แล้วกด Open เพื่อเปิดໂທ



รูปภาพที่ 4.13 หน้าสำหรับเปิดໂທบริการลูกค้า

หากกรอกข้อมูลครบถ้วนจะทำการ POST API เพื่อทำการเปิดบิลและผูกบิลเข้ากับหมายเลขโต๊ะ

```
function openTable() {
    let table = $('.form-select').val()
    let person= $('.form-control').val()

    $.ajax({
        url: '/api/table/open?id=' + table + '&person=' + person,
        type: 'POST',
        success: function () {
            console.log('success')
        }
    });
    $('#createDineInModal').modal('hide')
}
```

```
@PostMapping("/open")
public TableDTO openTableDineIn(@RequestParam int id, @RequestParam int
person) {
    if(tableService.tableIsAvailable(id) && person >= 1) {
        Bill bill = billService.createBill(person);
        Token token = tokenService.mappingNewTokenToBill(bill);
        Table table = tableService.mappingTableToBill(id, bill);

        TableDTO tableDTO = new TableDTO();
        tableDTO.setId(table.getId());
        tableDTO.setAvailable(table.isAvailable());
        tableDTO.setBillId(bill.getId());
        tableDTO.setToken(token.getId());
        tableDTO.setPaid(false);
        template.convertAndSend("/topic/table/update", tableDTO);

        UpdateBillOrderStatusMessage updateBillOrderStatusMessage = new
UpdateBillOrderStatusMessage(bill.getId(), ConstantUtil.COMPLETE);
        template.convertAndSend("/topic/bill/status/update",
updateBillOrderStatusMessage);
        return tableDTO;
    }
    return null;
}
```

เมื่อทำการเปิดตัวสำเร็จหน้าเว็บไซต์จะขึ้น QR Code สำหรับการสั่งอาหารโดยต้องนั่งให้ลูกค้าสแกน



รูปภาพที่ 4.14 หน้าแสดง QR Code สำหรับลูกค้า DINE-IN

เมื่อการเปิดตัวสำเร็จ ข้อมูลต้องนั่งจะถูกส่งผ่านมายัง WebSocket เพื่อนำไปสร้างการ์ดเพิ่มและแสดง QR Code สำหรับการสั่งอาหารของต้องนั่ง

```
stompClient.subscribe('/topic/table/update', async function (message) {
    console.log('success')
    await updateTable(JSON.parse(message.body))
});

function loadQr(billId) {
    $('#qrPic').empty()
    if(allTable[index].indexOf(billId)] .token != null) {
        var qrCode = new QRCode(document.getElementById("qrPic"), {
            text: window.location.origin + "/dine-in/" +
        allTable[index].indexOf(billId)] .token,
            width: 300,
            height: 300,
            colorDark : "#000000",
            colorLight : "#ffffff",
            correctLevel : QRCode.CorrectLevel.H
        });
    }
}
```

#### 4.2.2. หน้าสำหรับจัดการครัว

ออเดอร์แต่ละใบจะเรียงตามวันเวลาที่สั่งและมีสีบ่งบอกสถานะเหมือนหน้าจัดการลูกค้า

- เมื่อต้องการเริ่มทำออเดอร์ได้ให้กดปุ่ม Start เพื่อเปลี่ยนจากสถานะ ORDER เป็น COOKING ทำให้สามารถกด Checkbox อาหารในออเดอร์นั้นได้
- เมื่อทำอาหารได้เสร็จแล้วให้ทำการกด Checkbox สถานะของอาหารนั้นจะเป็น SERVING และซึ่งอาหารจะโอนเข้าด้านขวา
- หากอาหารในออเดอร์ทั้งหมดมีสถานะเป็น COMPLETE ออเดอร์นั้นจะหายไปโดยอัตโนมัติ

Table 4 #48	12:53	TAKE-OUT #55	12:54	Table 5 #50	12:55	Table 5 #50	13:00	Table 5 #50	13:03
<input type="checkbox"/> 1x บาร์บีคิวเบคอนชีสส์ บีฟชีสส์ (เบลล์ชีสส์), บีฟชีสส์		<input type="checkbox"/> 3x ชิกเกนเวอร์ชิงโค้กแอลก ครีมเบลล์ชีสส์, ครีมเบลล์ชิงโค้กแอลก <input type="checkbox"/> 1x กาแฟ แฟร์บปูร์ปีน แยกช่องบีบี, บีบี		<input checked="" type="checkbox"/> 1x กาแฟ แฟร์บปูร์ปีน แยกช่องบีบี, บีบี		<input type="checkbox"/> 1x บาร์บีคิวเบคอนชีสส์ บีฟชีสส์ (เบลล์ชีสส์), บีฟชีสส์ <input type="checkbox"/> 1x บุฟเฟ่ Milkshake ไม่หวาน 0%, เอเวอร์เบลล์ช่องบีบี (เบลล์ชีสส์), บีฟชีสส์ (เบลล์ชีสส์)		<input type="checkbox"/> 2x ชิกเกนเวอร์ชิงโค้กแอลก ครีมเบลล์ชิงโค้กแอลก	
In Process		Start		In Process		In Process		Start	

รูปภาพที่ 4.15 หน้าจัดการรายการอาหารในครัว

เมื่อเข้าสู่หน้าจัดการครัวจะทำการเรียกข้อมูลเกี่ยวกับอาหารที่มีสถานะ ORDER COOKING และ SERVING จาก API นำมาจัดกลุ่มให้อยู่ในกรุํดเดียวกันโดยดูจากเลขบิลและเวลาที่สั่ง

```
function getDataDoing() {
    return new Promise(function (resolve, reject) {
        $.get( '/api/order/doing', function( data ) {
            resolve(data)
        });
    })
}

$( document ).ready(async function () {
    connect()
    let data = await getDataDoing()
    if(data.length > 0){
        initializeOrder(data)
    }
})
```

```
@GetMapping("/doing")
public List<UpdateOrderDTO> getAllOrderNotPendingAndCancelAndComplete()
{
    return orderService.getDoingOrder();
}
```

### 4.3. ส่วนของผู้ดูแล (ADMIN)

4.3.1. สามารถจัดการทุกหน้าของพนักงานได้

4.3.2. หน้าสำหรับจัดการเมนูในร้าน

- สามารถดูเมนู กลุ่มตัวเลือกเพิ่มเติม ตัวเลือกทั้งหมดในร้าน
- สามารถสร้างเพิ่มใหม่หรือทำการลบ แก้ไขสิ่งที่มีอยู่เดิมได้ แท็บเมนูจะแสดงเมนูทั้งหมดพร้อมรายละเอียดเกี่ยวกับเมนูนั้น

Menu	Category	Option Group	Option	Table	<a href="#">Index</a>
<a href="#">Create</a>				<input type="text" value="Search"/>	
Picture	Name	Category	Price	Quantity	
Milk Tea with Boba	ชาบันโนบุก	ชาบัน 22 ออนไลน์	50	0	
Milk Tea	ชาบันรสนมเผือก	ชาบัน 22 ออนไลน์	55	0	
Milk Tea with Chocolate	ชาบันรสนมช็อกโกแลต	ชาบัน 22 ออนไลน์	55	0	
Milk Tea with Caramel	ชาบันรสคาราเมล	ชาบัน 22 ออนไลน์	60	0	

รูปภาพที่ 4.16 หน้าจัดการเมนู

เมื่อเปิดหน้าจัดการเมนูในร้านจะเรียกข้อมูลเกี่ยวกับเมนูอาหารทั้งหมดผ่าน API จากนั้นแสดงผลในรูปแบบของตารางโดยในแท็บเมนู สามารถจัดเรียงข้อมูลโดยใช้คลิ๊ก colum ซึ่ง, หมวดหมู่, ราคาและจำนวนได้

```
<table id="tableFood"
    data-url="/api/item/food"
    class="table">
    <thead>
        <tr>
            <th data-field="image" data-formatter="imageFormatter" data-halign="left" data-align="center" data-width="200">Picture</th>
            <th data-field="name" data-sortable="true" >Name</th>
            <th data-field="category" data-sortable="true" >Category</th>
            <th data-field="price" data-sortable="true" >Price</th>
            <th data-field="quantity" data-sortable="true" >Quantity</th>
        </tr>
    </thead>
    <tbody>
    </tbody>
</table>
```

```
@JsonView(Views.Overall.class)
@GetMapping("/food")
public List<Item> findAllFood() {
    return itemService.findAllFood();
}
```

แท็บหมวดหมู่จะแสดงหมวดหมู่ทั้งหมดสามารถเรียงหมวดหมู่ใหม่เพื่อเปลี่ยนลำดับการแสดงผลในหน้าเมนูของลูกค้าได้และสามารถลบหมวดหมู่ที่ไม่มีอาหารใด ๆ อ้างอิงถึงแล้วได้

Category
บังชุออนไลน์
มังกะเนะสด 22 ออนเซ็ต
ซีอิ๊วไก่และเตี๊ยบครีม 16 ออนเซ็ต
ชาบู 22 ออนเซ็ต

รูปภาพที่ 4.17 หน้าจัดการหมวดหมู่

เมื่อเปิดหน้าจัดการหมวดหมู่จะเรียกข้อมูลซึ่งและลำดับของหมวดหมู่ทั้งหมดผ่าน API จากนั้นแสดงผลในแท็บหมวดหมู่

```

function getDataAllCategory(){
    return new Promise(function (resolve,reject){
        $.get( '/api/item/list-category' , function( data ) {
            resolve(data)
        });
    })
}

async function createTableCategory(){
    let categorys = await getDataAllCategory()

    for(let i=0;i<categorys.length;i++){
        $('#tableCategory').bootstrapTable('insertRow',{
            index: i,
            row: {
                id: i,
                name: categorys[i]
            }
        })
    }
}

```

แท็บกลุ่มของตัวเลือกจะแสดงกลุ่มตัวเลือกทั้งหมดพร้อมบอกว่ากลุ่มตัวเลือกนี้ถูกใช้ในกี่เมนู และบอกจำนวนตัวเลือกที่ถูกเลือกได้ขั้นต่ำสุดและมากที่สุดเท่าไหร่

Menu	Category	Option Group	Option	Table	Index
<a href="#">Create</a>					<input type="text" value="Search"/>
ID	Name		Min	Max	Using
1	เลือก - ก็อปปี้		0	10	18
2	ระดับความหวาน		1	1	4
3	เลือกรูปแบบการเสิร์ฟ		0	1	18
4	เลือก บัน		0	1	15
5	เลือกระดับความหวาน		1	1	14
6	บรรจุภัณฑ์ - เมบูของหวาน		1	1	2
7	ก็อปปี้พิเศษเฉพาะกสุเมรุบครัม		0	5	3

Showing 1 to 7 of 7 rows

รูปภาพที่ 4.18 หน้าจัดการกลุ่มตัวเลือก

เมื่อเปิดหน้าจัดการเมนูในร้านจะเรียกข้อมูลเกี่ยวกับกลุ่มตัวเลือกทั้งหมดและจำนวนการถูกใช้งานแต่ละกลุ่มตัวเลือกผ่าน API จากนั้นแสดงผลในแท็บกลุ่มตัวเลือก

```

function getDataOptionGroup(){
    return new Promise(function (resolve,reject){
        $.get( '/api/optional/all', function( data ) {
            resolve(data)
        });
    })
}

function getDataOptionGroupUsing(){
    return new Promise(function (resolve,reject){
        $.get( '/api/optional/using', function( data ) {
            resolve(data)
        });
    })
}

optionGroup = await getDataOptionGroup()
let optionGroupUsing = await getDataOptionGroupUsing()

for(i=0;i<optionGroup.length;i++){
    $('#tableOptionGroup').bootstrapTable('insertRow',{
        index: i,
        row: {
            id: optionGroup[i].id,
            name: optionGroup[i].name,
            using :optionGroupUsing[optionGroup[i].id],
            min:optionGroup[i].min,
            max:optionGroup[i].max
        }
    })
}
}

```

```

@JsonView(Views.Overall.class)
@GetMapping("/all")
public List<Optional> findAll() {
    return optionalService.findAll();
}

@GetMapping("/using")
public Map<Integer, Integer> optionalUsingCount() {
    return optionalService.optionalUsingCount();
}

```

แท็บตัวเลือกจะแสดงตัวเลือกทั้งหมดพร้อมบอกว่าตัวเลือกนี้ถูกใช้ในกี่กลุ่มตัวเลือก

ID	Name	Price	Quantity	Using
1	เจ้ากี้ย	15	0	1
2	ไข่บุกบรรจุภัณฑ์	10	0	1
3	เยลลี่สูตรไข่	15	0	1
4	มัคชาเยลลี่	15	0	1
5	คั่วแดงอะซูกิ	15	0	1
6	พุดดิ้งนมสด	15	0	1
7	บรรจุภัณฑ์เยลลี่	15	0	1
8	เยลลี่วานิลลา	15	0	1
9	วาปิลลาครีมเบลล์	15	0	1
10	ช็อกโกแลตครีมเบลล์	15	0	1

< 1 2 3 >

Showing 1 to 10 of 27 rows 10 ▲ rows per page

รูปภาพที่ 4.19 หน้าจัดการตัวเลือก

เมื่อเปิดหน้าจัดการเมนูในร้านจะเรียกข้อมูลเกี่ยวกับตัวเลือกทั้งหมดและจำนวนการถูกใช้งานแต่ละตัวเลือกผ่าน API จากนั้นแสดงผลในแท็บตัวเลือก

```

function getDataOption() {
    return new Promise(function (resolve, reject) {
        $.get( '/api/item/category/OPTION', function( data ) {
            resolve(data)
        });
    })
}

function getDataOptionUsing() {
    return new Promise(function (resolve, reject) {
        $.get( '/api/optional/item-using', function( data ) {
            resolve(data)
        });
    })
}

option = await getDataOption()
let optionUsing = await getDataOptionUsing()

for(i=0;i<option.length;i++){
    $('#tableOption').bootstrapTable('insertRow', {
        index: i,
        row: {
            id : i,
            idItem: option[i].id,
            name: option[i].name,
            using :optionUsing[option[i].id]
        }
    })
}

```

```

@GetMapping("/item-using")
public Map<Integer, Integer> itemOptionUsingCount() {
    return optionalService.itemOptionUsingCount();
}

```

แท็บเต็มจะแสดงเลขตัวทั้งหมดที่ร้านมี สามารถสร้างเลขตัวที่ไม่ซ้ำกันได้และสามารถลบเลขตัวใด

Table showing IDs 1 to 10:

ID
1
2
3
4
5
6
7
8
9
10

Showing 1 to 10 of 10 rows

รูปภาพที่ 4.20 หน้าจัดการตัว

เมื่อเปิดหน้าจัดการเมนูในร้านจะเรียกข้อมูลเกี่ยวกับตัวทั้งหมดผ่าน API จากนั้นแสดงผลรูปแบบตารางในแท็บเต็ม สามารถจัดเรียงข้อมูลด้วยเลขตัวใด

```
<table id="tableNumber"
      data-toggle="table"
      data-pagination="true"
      data-pagination-h-align="left"
      data-pagination-detail-h-align="right"
      data-search="true"
      data-toolbar="#toolbarTable"
      data-url="/api/table/all"
>
  <thead >
    <tr>
      <th data-field="id" data-sortable="true">ID</th>
    </tr>
  </thead>
  <tbody>
  </tbody>
</table>
```

## สรุป

จากโค้ดบางส่วนที่นำมาอ้างอิงในบทนี้แสดงให้เห็นว่าข้อมูลส่วนใหญ่ที่นำมาแสดงผลนั้นมาจากการเรียกใช้ API เป็นส่วนใหญ่ หากหน้าได้ต้องการอัปเดตตลอดเวลาจะทำการเรียกใช้ WebSocket แทนที่เพื่อให้ทราบถึงข้อมูลที่มีการเปลี่ยนแปลง

## บทที่ 5

### การทดสอบระบบ

ในบทนี้จะกล่าวถึงการทดสอบการใช้งานระบบ เพื่อทดสอบว่าการทำงานของระบบสามารถทำงานได้อย่างถูกต้อง และมีการแก้ไขจัดการปัญหาของระบบที่เกิดขึ้นได้ หาข้อผิดพลาดของระบบในรูปแบบต่าง ๆ ที่เกิดขึ้นของเว็บแอปพลิเคชัน เพื่อนำไปพัฒนาปรับปรุงแก้ไขให้สามารถรองรับป้องกันแก้ไขปัญหาได้ดียิ่งขึ้น

เว็บแอปพลิเคชันมีการรักษาความปลอดภัยด้วยการใช้งาน Basic Authentication ในการจำกัดสิทธิ์การเข้าถึงหน้าเว็บและการใช้งาน API ดังนั้นการเข้าถึงหน้าเว็บแอปพลิเคชันในส่วนของพนักงานหรือผู้ดูแลสามารถเข้าถึงได้จาก URL Path /login เพื่อทำการล็อกอินเข้าสู่ระบบสำหรับพนักงานหรือผู้ดูแล โดยบัญชีผู้ใช้ในระบบปัจจุบันมีดังนี้

- Username: admin | Password: password | Roles: ADMIN, STAFF
- Username: staff | Password: password | Roles: STAFF

```
src/main/java/th/ku/orderme/security/SecurityConfig.java

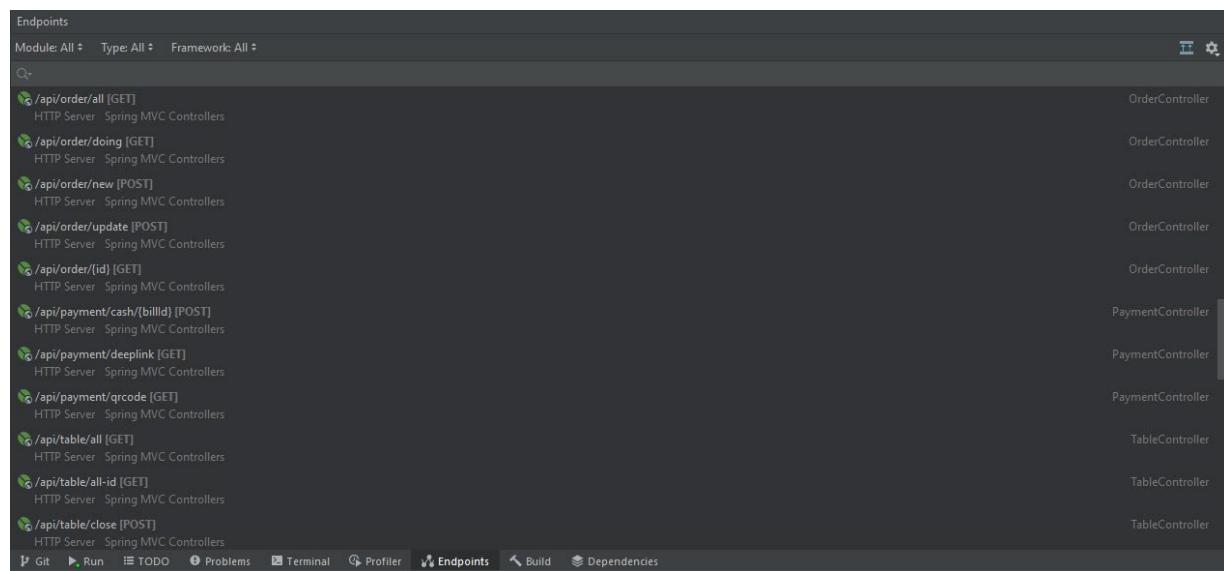
public void configureGlobal(AuthenticationManagerBuilder auth) throws
Exception {

auth.inMemoryAuthentication().withUser("admin").password(passwordEncoder
().encode("password")).roles("ADMIN", "STAFF");

auth.inMemoryAuthentication().withUser("staff").password(passwordEncoder
().encode("password")).roles("STAFF");
}
```

## 5.1. การทดสอบ API

API ในระบบทั้งหมดในช่วงระหว่างการพัฒนาจนกระทั่งพัฒนาเสร็จสิ้น จะมีการทดสอบความถูกต้องของ API ผ่านโปรแกรม Postman ในการใช้งาน HTTP Method GET, POST และ DELETE ข้อมูลฝ่าย REST API เพื่อให้มั่นใจได้ว่าการทำงานของระบบส่วนของ Controller, Service และ Repository ทั้งหมดสามารถทำงานร่วมกันได้อย่างถูกต้อง เมื่อได้ผลลัพธ์การทำงานที่ถูกต้องแล้วจึงนำไปเชื่อมต่อเข้ากับระบบหน้าบ้านเพื่อให้สามารถใช้งานในการตั้งและส่งข้อมูลต่าง ๆ ระหว่างหน้าเว็บกับเซิร์ฟเวอร์ได้อย่างสะดวกและง่ายดาย



รูปภาพที่ 5.1 ตัวอย่างรายชื่อ Endpoints ในระบบ

### 5.1.1. ตัวอย่างการทดสอบใช้งาน API /api/order/new สำหรับการสั่งอาหาร

ใช้ HTTP Method POST เพื่อส่งข้อมูลไปยัง API Endpoint คือ /api/order/new ในรูปแบบของ JSON Request Body โดยเบื้องหลังการทำงานของ API จะมีส่วนของ Controller, Service และ Repository ที่จะทำการตรวจสอบเงื่อนไขต่าง ๆ ที่ระบุเอาไว้ ถ้าหากเงื่อนไขถูกต้องทั้งหมดจึงสามารถสั่งอาหารได้ โดยเงื่อนไขมีดังนี้

- ตารางฐานข้อมูล token: ใน HTTP Request Header จำเป็นจะต้องมี Cookie ชื่อ uid ที่ถูกใช้เป็น id ของตาราง token สำหรับการระบุ bill\_id เพื่อใช้สำหรับสั่งอาหาร
- ตารางฐานข้อมูล bill: แล้วที่มี id ตรงกับ bill\_id จำเป็นจะต้องมี status เป็น OPEN ถึงจะสามารถสั่งอาหารได้
- ตารางฐานข้อมูล item: ที่ใช้เก็บข้อมูลเกี่ยวกับอาหารหรือตัวเลือก ถ้าหากข้อมูลภายในແ昆นั้นส่วนของคอลัมน์ check\_quantity มีค่าเป็น 1 หมายความว่าจำเป็นต้องมีการตรวจสอบ quantity ถ้าหากว่าไม่เพียงพอจะไม่สามารถสั่งอาหารได้ หรือถ้าหากว่า flag มีค่าเป็น -1 ก็จะไม่สามารถสั่งอาหารได้เช่นกัน เนื่องจากมีความหมายว่า item ถูกลบแล้ว
- ตารางฐานข้อมูล optional: ที่ใช้เก็บข้อมูลเกี่ยวกับกลุ่มตัวเลือก ถ้าหากข้อมูลภายในແ昆นั้นส่วนของคอลัมน์ flag มีค่าเป็น -1 จะไม่สามารถสั่งอาหารได้เนื่องจากมีความหมายว่า optional ถูกลบแล้ว
- ตารางฐานข้อมูล item\_optional: จำเป็นจะต้องมีแล้วที่ Composite Foreign Key ตรงกับที่ถูกระบุมาในการสั่งอาหาร
- ตารางฐานข้อมูล optional\_item: จำเป็นจะต้องมีแล้วที่ Composite Foreign Key ตรงกับที่ถูกระบุมาในการสั่งอาหาร

เมื่อเงื่อนไขทั้งหมดถูกต้องการสั่งอาหารจึงจะถึงว่าสำเร็จ ระบบจะมีการบันทึกข้อมูลการสั่งอาหารลงภายในฐานข้อมูลดังนี้

- ตารางฐานข้อมูล order: เก็บข้อมูลอาหารที่สั่งในส่วนของ quantity และ comment
- ตารางฐานข้อมูล select\_item: เก็บข้อมูลอาหารที่สั่งในส่วนของการเลือกตัวเลือกจากกลุ่มตัวเลือกของออเดอร์นั้น

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request URL:** POST Order /new (Cart) to localhost:8080/api/order/new.
- Body (JSON):**

```

1 {
2   "billId": 66,
3   "orderRequests": [
4     {
5       "itemId": 1,
6       "quantity": 1,
7       "comment": "ຂໍ້ມູນແນວທຸນ",
8       "selectItems": [
9         {
10           "optionalId": 1,
11           ...
12         }
13       ]
14     }
15   ]
16 }

```
- Response Body:**

```

1 Bill ID: 66
2 Timestamp: 2022-04-05T18:47:44.517
3 -----
4 1x ຂໍາມ 75.0
5   ວ່າງ 100%, ເຄົກວິ, ໄນກຳປັ້ງ
6   ດີຈຳກຳມືກຳ
7 2x ພັກລົມເສດ (ນາເຊົາວົມເສດ) 220.0
8   ວ່າງ 75%, ພັກລົມເສດ (ເຈລື້ອກ້ອນ), ດັວຍລວມຫຼັກ
9   ດີຈຳກຳມືກຳ
10 -----
11 Total: 295.0

```
- Bottom Tools:** Cookies, Capture requests, Bootcamp, Runner, Trash.

ຮູບພາບທີ 5.2 ການໃຊ້ຈຳນວດ Postman ໃນການທົດສອບ API

The screenshot shows the IntelliJ IDEA interface with two panes:

- Left Pane:** OrderController.java code review. It highlights several lines of code with annotations like `@PostMapping`, `@RequestBody`, and `OrderDetail`.
- Right Pane:** Postman interface showing the same API request and response as in the previous screenshot.

ຮູບພາບທີ 5.3 ການທົດສອບ Controller ແລະ Service ຄວບຄັ້ງ Postman

The image shows five separate phpMyAdmin database interfaces side-by-side, each displaying a different table from a database named 'order'. The tables are:

- order\_items**: Shows items available for delivery. One row is selected for deletion.
- order\_main**: Shows basic order details. One row is selected for deletion.
- order\_opt**: Shows optional items. One row is selected for deletion.
- order\_opt\_detail**: Shows detailed optional item information. One row is selected for deletion.
- order\_out**: Shows take-out orders. One row is selected for deletion.

รูปภาพที่ 5.4 ฐานข้อมูลมีการอัปเดตข้อมูลจากระบบ

The image displays two main sections of a web application for managing bills:

- Dine In Table**: Shows a table with rows numbered 1, 4, and 9. A modal window titled "Table 9 Bill#66" is open, showing an order for "DINE-IN" with a quantity of 1. Buttons for "Complete", "Order", and "Delete" are visible.
- Take Out Bill**: Shows two bill entries:
  - Bill ID: 54**: Total amount 295, with a note: "Option: หวาน 100%, เวลาจิ้ง, ไม่บุก内地".
  - Bill ID: 55**: Total amount 220, with a note: "Option: หวาน 75%, บีบี-เจลล์ (เจลล์เจลล์), ร่องดังจางๆ".
- Order Details**: Four columns showing order status and timestamp:
  - Table 5 #50**: Status 13:03, Comment: "x ชีสเค้กช็อกโกแลต สูตรเบเกอรี่"
  - TAKE-OUT #57**: Status 15:28, Comment: "Ix คุกกี้Milkshake หวาน 25%, กินง่ายมาก"
  - TAKE-OUT #65**: Status 22:44, Comment: "Ix ชาเย็นผลไม้ชีสเค้ก หวาน 100%, เวลาจิ้ง, ไม่บุก内地"
  - Table 9 #66**: Status 18:47, Comment: "2x บีบี-เจลล์ (เจลล์เจลล์)"

รูปภาพที่ 5.5 หน้าเว็บมีการแสดงผลข้อมูลที่ถูกต้องหลังจากการทดสอบผ่าน API

## 5.2. ส่วนของลูกค้า (USER)

### 5.2.1. ทดสอบหน้าเลือกเมนู

ลูกค้าจะไม่สามารถเข้าหน้าเลือกเมนูได้ และถูกพาไปหน้าแรกของเว็บไซต์แทนหากไม่ได้เข้าเว็บจาก URL สำหรับลูกค้าแบบ TAKE-OUT หรือสแกนจาก QR Code ที่ได้รับจากพนักงานสำหรับลูกค้าแบบ DINE-IN

### 5.2.2. ทดสอบการเพิ่มลงตะกร้า

หากเลือกตัวเลือกไม่ถึงขั้นต่ำที่กำหนดไว้ในแต่ละกลุ่มตัวเลือกทั้งหมด จะไม่สามารถเพิ่มอาหารที่ต้องการลงตะกร้าได้



**ข้าวกระเทียมพริกไทย**  
สามารถเลือกเนื้อสตว์ได้

---

**ข้าวกระเทียมพริกไทย** 1 Required

เลือก 1

- หมูอึ้ง
- ไก่
- ตับ +฿ 5
- กุ้ง +฿ 20
- ปลาหมึก +฿ 20

**65 – Get in Cart**



**ข้าวกระเทียมพริกไทย**  
สามารถเลือกเนื้อสตว์ได้

---

**ข้าวกระเทียมพริกไทย** 1 Required

เลือก 1

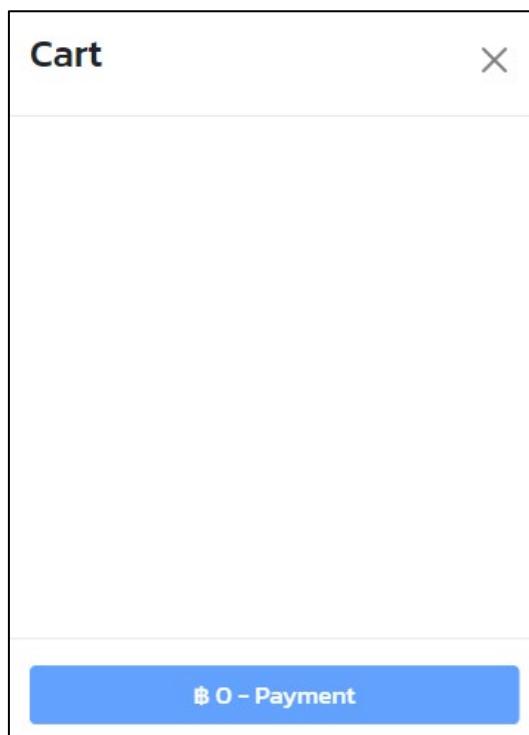
- หมูอึ้ง
- ไก่
- ตับ +฿ 5
- กุ้ง +฿ 20
- ปลาหมึก +฿ 20

**65 – Get in Cart**

รูปภาพที่ 5.6 นำอาหารใส่ตะกร้าได้เมื่อทำการเลือกรอบตามเงื่อนไข

### 5.2.3. ทดสอบการสั่งรายการอาหาร

หากไม่มีรายการอาหารที่เลือกในตะกร้าจะไม่สามารถกดสั่งรายการอาหารได้ หากการสั่งรายการอาหารสำเร็จจะกลับมาอย่างหน้าเมนูอาหารดังเดิม



รูปภาพที่ 5.7 ไม่สามารถสั่งอาหารได้เมื่อไม่มีรายการอาหารในตะกร้า

### 5.2.4. ทดสอบการดูบิลและสถานะสำหรับ DINE-IN

ลูกค้า DINE-IN เท่านั้นที่สามารถเห็นและกดปุ่ม Bill เพื่อดูบิลได้ บิลสามารถดูสถานะของอาหารได้



รูปภาพที่ 5.8 แท็บแสดงตะกร้าและบิลสำหรับลูกค้า DINE-IN

### 5.2.5. ทดสอบการไปยังหน้าทางเลือกชำระเงิน

เมื่อลูกค้าเข้าสู่หน้าชำระเงินแล้วจะไม่สามารถกลับไปสั่งอาหารเพิ่มได้อีก หากกดกลับไปจะยังอยู่หน้าชำระเงินดังเดิม และหากชำระเงินสำเร็จจะถูกนำทางไปยังหน้าใบเสร็จ

### 5.2.6. การชำระเงินแบบ QR Code

เมื่อกดปุ่ม Confirm แล้ว ถ้าหากการชำระเงินสำเร็จจะไปยังหน้าใบเสร็จแต่หากยังไม่สำเร็จจะยังอยู่หน้า QR Code ดังเดิม หรือถ้าหากกดปุ่ม Back จะกลับมายังหน้าเลือกช่องทางการชำระเงิน

### 5.2.7. SCB Easy Sandbox

เมื่อทำการชำระเงินสำเร็จผ่าน SCB Easy Sandbox ระบบจะพากลับมายังหน้าเว็บในส่วนของหน้าใบเสร็จ หากยังไม่ได้ทำการชำระเงินสามารถเลือกช่องทางการชำระเงินในหน้าเลือกช่องทางการชำระเงินได้

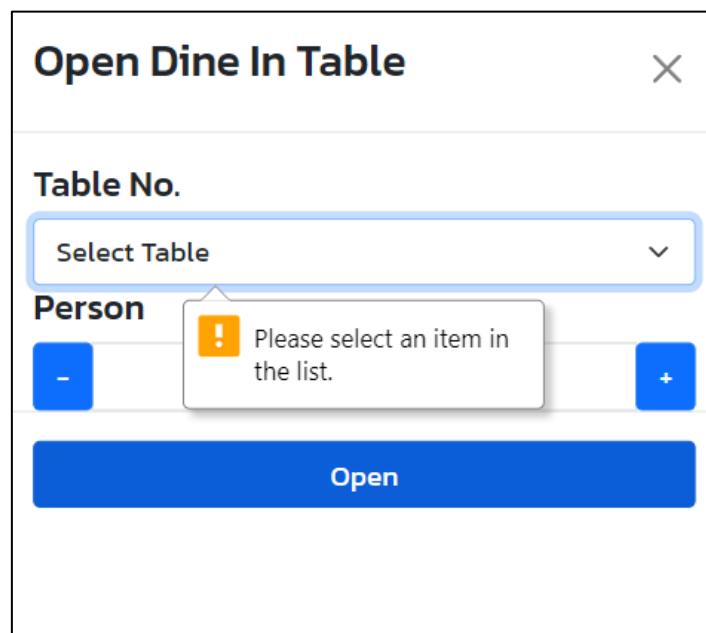
### 5.2.8. ทดสอบการไปยังหน้าใบเสร็จ

สามารถไปยังหน้าใบเสร็จได้เมื่อชำระเงินสำเร็จ และใบเสร็จจะแสดงผลเฉพาะเลข Ref1 ที่ถูกต้องและมีการสร้างใบเสร็จในระบบแล้วเท่านั้น

### 5.3. ส่วนของพนักงาน (STAFF)

#### 5.3.1. ทดสอบการเปิดโต๊ะให้บริการลูกค้า

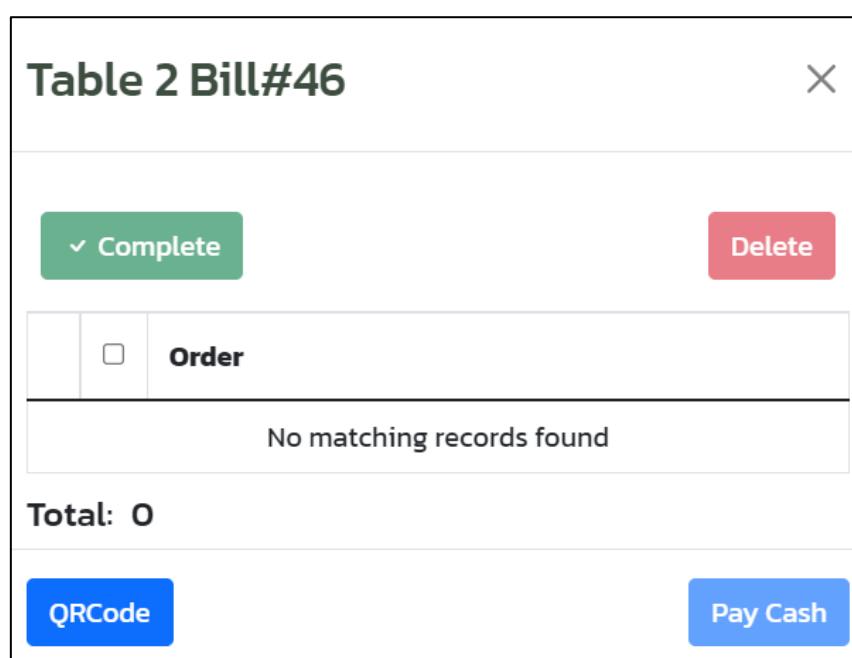
จำนวนคนขั้นต่ำคือ 1 จะมีการแจ้งเตือนหากไม่มีการเลือกโต๊ะ



รูปภาพที่ 5.9 ต้องทำการเลือกหมายเลขโต๊ะก่อนเปิดโต๊ะ

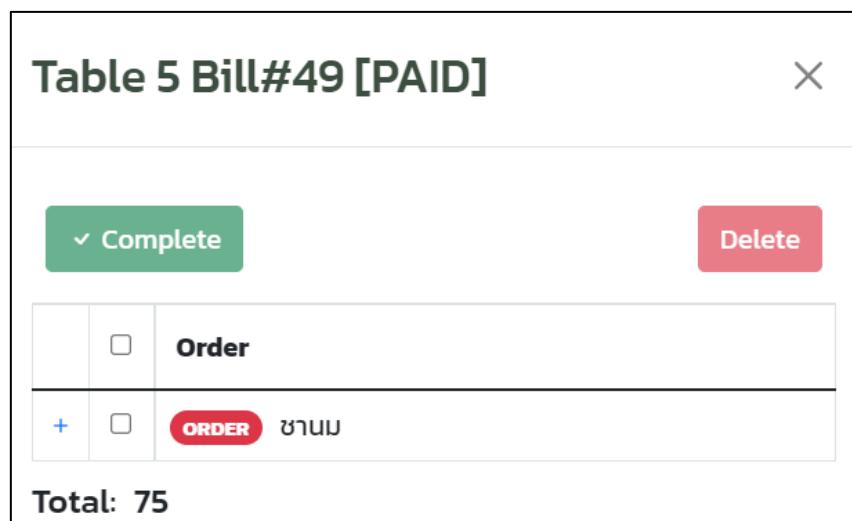
### 5.3.2. ทดสอบการดูบิลลูกค้า

ลูกค้าแต่ละประเภทจะมีหน้าบิลที่แตกต่างกันสามารถเปลี่ยนสถานะและจ่ายเงินได้



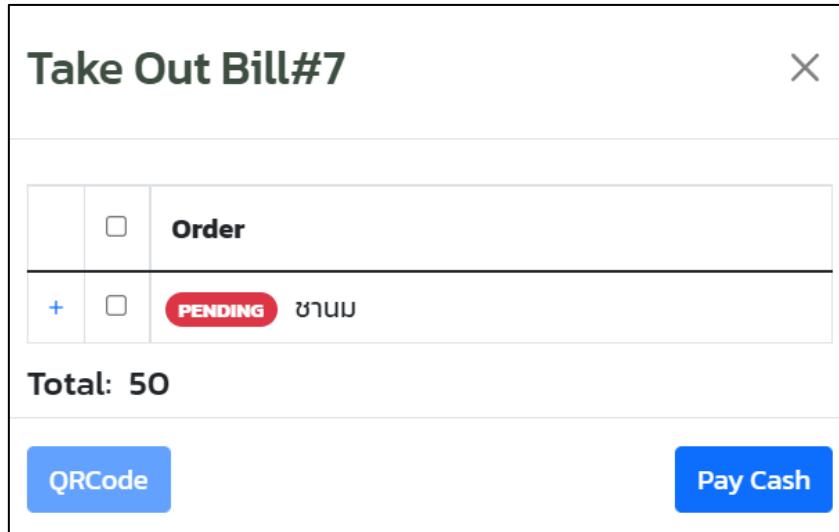
รูปภาพที่ 5.10 บิล DINE-IN จะไม่สามารถชำระเงินได้หากไม่มีรายการอาหารที่สั่ง

- ลูกค้า DINE-IN ที่ชำระเงินแล้วจะไม่สามารถกดปุ่ม Delete และจ่ายเงินได้



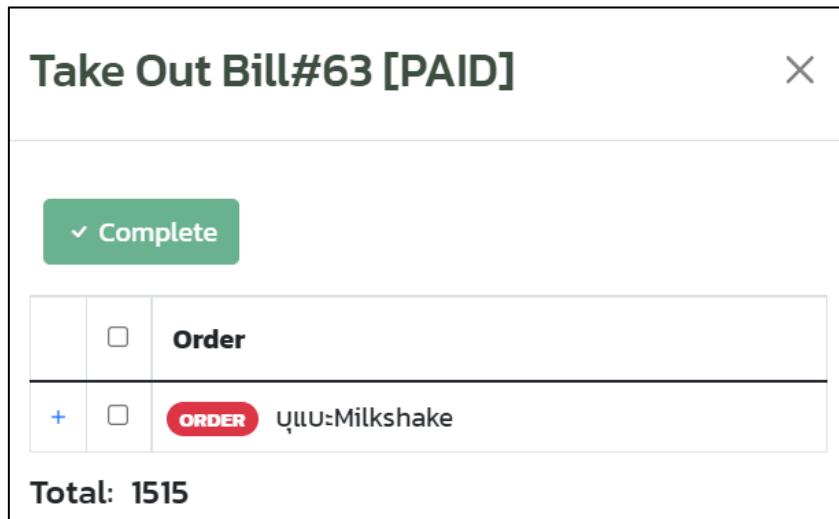
รูปภาพที่ 5.11 บิล DINE-IN ที่ชำระเงินแล้วจะไม่สามารถ Delete รายการอาหารได้

- ลูกค้า TAKE-OUT ที่ยังไม่ชำระเงินจะไม่สามารถกดปุ่มดู QR Code และเปลี่ยนสถานะได้



รูปภาพที่ 5.12 ปิล TAKE-OUT ไม่สามารถดู QR Code และไม่สามารถเปลี่ยนสถานะรายการอาหารได้

- ลูกค้า TAKE-OUT ที่ชำระเงินแล้วจะไม่สามารถกดปุ่มดู QR Code, กดปุ่มจ่ายเงินและกดปุ่ม Delete ได้



รูปภาพที่ 5.13 ปิล TAKE-OUT ที่ชำระเงินแล้วจะไม่สามารถ Delete รายการอาหารได้

### 5.3.3. ทดสอบการปรับสถานะรายการอาหาร

- หากกดปุ่ม Complete รายการอาหารที่เลือกไว้ทั้งหมดจะถูกปรับสถานะเป็น COMPLETE แต่หากรายการอาหารมีสถานะเป็น COMPLETE อยู่แล้วจะไม่มีการเปลี่ยนแปลง
- เมื่อกดปุ่ม Delete รายการอาหารที่เลือกไว้ทั้งหมดจะถูกปรับสถานะเป็น CANCEL แต่หากรายการอาหารมีสถานะเป็น COMPLETE จะไม่สามารถเปลี่ยนให้เป็น CANCEL ได้

### 5.3.4. ทดสอบการปรับสถานะรายการอาหารในครัว

- หากไม่กดปุ่ม Start เพื่อเปลี่ยนสถานะอาหารจาก ORDER เป็น COOKING จะไม่สามารถกด Check อาหารที่ต้องการเปลี่ยนสถานะเป็น SERVING ได้
- เมื่อกดปุ่ม Start แล้วจะไม่สามารถกดชำได้อีก
- เมื่อกด Check เปลี่ยนสถานะอาหารเป็น SERVING แล้วจะไม่สามารถกด check อีกครั้ง เปลี่ยนสถานะได้อีก

TAKE-OUT #63	21:39	Table 5 #49	21:40
<input type="checkbox"/> 1x บุฟเฟ่:Milkshake ไม่หวาน 0%, บรรจุภัณฑ์เจลลี่		<input type="checkbox"/> 1x ชาเย็น หวาน 100%, ใบบุกนำไปฝัง, ครั้มเบลวานิลลา	<input checked="" type="checkbox"/> ชาเย็น หวาน 100%, ใบบุกนำไปฝัง, ครั้มเบลวานิลลา
<b>Start</b>		<b>In Process</b>	<b>In Process</b>

รูปภาพที่ 5.14 การแสดงผลหน้าจัดการรายการอาหารในครัว

## 5.4. ส่วนของผู้ดูแล (ADMIN)

### 5.4.1. ทดสอบการเพิ่มเมนูอาหาร

สามารถเพิ่มเมนูอาหารได้ด้วยปุ่ม Create ในแท็บเมนูของหน้าจัดการเมนูในร้านจากนั้น กรอกข้อมูลอาหารที่ต้องการเพิ่ม หากกรอกไม่ครบหรือไม่ถูกต้องจะมีการแจ้งเตือนแต่ถ้ากรอกครบถ้วนและถูกต้องจะกลับมาอยังแท็บเมนู หากไม่ใส่ราคาระบบจะใส่ 0 ให้โดยอัตโนมัติ

**Name**

**Description**

**Category**  

Select...

! Please fill out this field.

**Picture**  

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Choose File
No file chosen
Clear

**Option Group**

ID	Group Name
	No matching records found

รูปภาพที่ 5.15 การเพิ่มเมนูใหม่ต้องกรอกข้อมูลที่จำเป็นให้ครบ

#### 5.4.2. ทดสอบการเพิ่มกลุ่มตัวเลือก

สามารถเพิ่มกลุ่มตัวเลือกได้ด้วยปุ่ม Create ในแท็บกลุ่มตัวเลือกของหน้าจัดการเมนู ในร้านจากนั้นกรอกข้อมูลกลุ่มตัวเลือกที่ต้องการเพิ่ม จำนวนตัวเลือกที่สามารถเลือกได้มาก ที่สุดจะมากกว่าหรือเท่ากับจำนวนตัวเลือกที่สามารถเลือกได้ขึ้นต่ำเสมอ หากไม่กรอกชื่อกลุ่ม ตัวเลือกหรือไม่เลือกตัวเลือกอย่างน้อย 1 อย่างจะมีการแจ้งเตือนแต่ถ้ากรอกครบถ้วนจะกลับมายังแท็บกลุ่มตัวเลือก

**Group Name**

**Description**
!
Please fill out this field.

Select Option
Create Option

ID	Option	Price	Remove
No matching records found			

MIN
-
0
+
MAX
-
0
+

Back
Create

รูปภาพที่ 5.16 การเพิ่มกลุ่มตัวเลือกใหม่ต้องกรอกข้อมูลที่จำเป็นให้ครบ

#### 5.4.3. ทดสอบการเพิ่มตัวเลือก

สามารถเพิ่มตัวเลือกได้ด้วยปุ่ม Create ในแท็บตัวเลือกของหน้าจัดการเมนูในร้าน จากนั้นกรอกข้อมูลตัวเลือกที่ต้องการเพิ่ม หากไม่กรอกชื่อตัวเลือกหรือไม่จะมีการแจ้งเตือนแต่ ถ้ากรอกครบถ้วนจะกลับมาอยู่แท็บตัวเลือก หากไม่ใส่ราคาระบบจะใส่ 0 ให้โดยอัตโนมัติ

The screenshot shows a user interface for adding a new item option. The form has the following fields:

- Name:** An input field containing a single character.
- Quantity:** A radio button group where the "No" option is selected. A validation message "Please fill out this field." is shown in a box above the field.
- Price:** An empty input field.

At the bottom right is a blue "Create" button, and at the bottom left is a "Back" button.

รูปภาพที่ 5.17 การเพิ่มตัวเลือกใหม่ต้องกรอกข้อมูลที่จำเป็นให้ครบ

#### สรุป

จากการทดสอบการใช้งานเว็บแอปพลิเคชันที่พัฒนาขึ้นมา ช่วยให้สามารถตรวจสอบหาข้อผิดพลาด ของโปรแกรมที่พัฒนาขึ้นมาได้ และได้ทำการปรับปรุงแก้ไขข้อผิดพลาดที่พบในระหว่างการทดสอบจนกระทั่ง ได้ผลลัพธ์ที่ถูกต้องตามที่ต้องการ

## บทที่ 6

### บทสรุป

#### 6.1. สรุปผลการดำเนินงาน

สรุปผลการดำเนินงานตลอดช่วงเวลาของการพัฒนาโครงการระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร ในช่วงแรกจะเป็นการออกแบบคุณสมบัติของเว็บแอปพลิเคชันให้ตรงตามความต้องการของผู้ใช้งานแต่ละบทบาท โดยศึกษาอ้างอิงจากประสบการณ์ตรงที่พบเจอในสถานะลูกค้าที่ได้ปรับปรุงระบบอาหารตามร้านอาหาร และระบบการบริหารจัดการของร้านอาหารที่เคยพูดเจอ ให้มีความสะดวกในการใช้งานมากขึ้น เพื่อแก้ไขปัญหาต่าง ๆ ที่เคยพูดเจอในการให้บริการ

หลังจากออกแบบคุณสมบัติเรียบร้อยแล้ว ส่วนต่อมาคือการออกแบบระบบฐานข้อมูลและการทำงานภายใต้โปรแกรมให้สามารถตอบสนองความต้องการของผู้ใช้งานได้ โดยได้มีการออกแบบให้มีการเก็บข้อมูลให้ครบถ้วนตามคุณสมบัติที่ต้องมี และออกแบบให้สามารถรองรับการแก้ไขเพิ่มเติมได้ภายใต้อนาคต

หลังจากนั้นจึงเริ่มการพัฒนาโปรแกรม โดยมีการแบ่งหน้าที่ในการพัฒนาออกเป็นส่วนของ การพัฒนาระบบหน้าบ้าน และส่วนของการพัฒนาระบบหลังบ้าน และนำทั้งสองส่วนมาเชื่อมต่อ เพื่อให้สามารถทำงานร่วมกันได้ และในระหว่างที่มีการพัฒนาได้ทำการศึกษาหาข้อมูลและเทคนิคต่าง ๆ เพื่อนำมาประยุกต์ใช้งานกับโปรแกรมมากมายเพื่อให้สามารถพัฒนาโปรแกรมที่มีคุณสมบัติตามที่ต้องการได้

หลังจากนั้นจึงเข้าสู่ช่วงของการทดสอบโปรแกรม และทำการแก้ไขข้อผิดพลาดต่าง ๆ ของโปรแกรมจนกว่าได้โปรแกรมที่สมบูรณ์มีคุณสมบัติครบตามที่ต้องการ และใช้งานได้ปกติโดยไม่มีปัญหาเกิดขึ้น

## 6.2. ปัญหาและอุปสรรค

ปัญหาและอุปสรรคที่พบเจอในการพัฒนาระบบมีหลายอย่าง ไม่ว่าจะเป็นความไม่รู้ หรือไม่มีประสบการณ์ในการเขียนชุดคำสั่งบางอย่าง ทำให้การที่จะเขียนชุดคำสั่งบางอย่างในช่วงแรกเป็นไปได้ยาก ต้องทำการศึกษาหาข้อมูลก่อนการเขียน เพื่อให้สามารถเขียนออกมากแล้วได้ผลลัพธ์ตามที่ต้องการ และในบางครั้งอาจจะมีการเขียนชุดคำสั่งที่มีการออกแบบมาไม่ดี ทำให้ในการใช้งานจริงอาจพบปัญหาได้ในบางครั้ง จนสุดท้ายต้องมาทำการแก้ไขเพิ่มเติมการเขียนชุดคำสั่งโดยการออกแบบให้มีการทำงานที่ดียิ่งขึ้น

ในการออกแบบชุดคำสั่งระบบหลังบ้านบางส่วนมีการออกแบบให้สามารถรองรับการใช้ได้งานผ่าน REST API และ WebSocket แต่เนื่องจากฐานข้อมูลบางตารางมีการอ้างอิงความสัมพันธ์กับตารางอื่น ทำให้เกิดปัญหาขึ้นไม่สามารถใช้งานได้เมื่อใช้งานผ่าน WebSocket ในบางกรณี เช่น กรณีที่ต้องการดึง Object ออกมาจากภายในอีก Object และไม่สามารถดึงข้อมูลออกมาได้เนื่องจากข้อมูลเป็น Null เพราะปัญหา LazyInitializationException จาก Spring Data JPA ที่ใช้งาน จึงต้องทำการแก้ไขโดยการ Query ข้อมูลจากฐานข้อมูลตัวยุตนเองในบางครั้ง

เนื่องจากระบบมีการเชื่อมต่อในส่วนของการชำระเงินเข้ากับทางธนาคารไทยพาณิชย์ทำให้ในบางครั้งมีเหตุการณ์ที่ไม่สามารถทำการชำระเงินผ่าน SCB Easy Sandbox หรือไม่สามารถตรวจสอบการชำระเงินได้ เนื่องจากเป็นปัญหาจากทางฝ่ายของธนาคาร

ในการใช้งานระบบได้มีการให้ระบบทำงานอยู่บน Localhost Server ซึ่งสามารถเชื่อมต่อเข้ามาจากอินเทอร์เน็ตสาธารณะได้ด้วยการใช้งาน DDNS (Dynamic Domain Name System) ของผู้ให้บริการอินเทอร์เน็ตที่ใช้บริการอยู่ เพื่อให้ทำการ Forward Port จาก Public Domain มาสู่ Router ที่ใช้งานภายในบ้าน และให้ Router ทำการ Forward Port มาสู่ Localhost Server ซึ่งในบางครั้งอาจจะเกิดปัญหาขึ้นที่ DDNS ที่ใช้งานของผู้ให้บริการ หรือเกิดปัญหาขึ้นใน Router ที่ใช้งานอยู่ทำให้มีบางครั้งไม่สามารถเข้าถึงจากภายนอกได้จึงต้องทำการหาสาเหตุและแก้ไขว่าเกิดจากอะไร เพื่อให้ระบบกลับมาใช้งานได้อย่างปกติ

### 6.3. ประสบการณ์ที่ได้รับและสิ่งที่ได้เรียนรู้

จากการพัฒนาโปรแกรมระบบสั่งอาหารและชำระเงินผ่านเว็บแอปพลิเคชันสำหรับร้านอาหาร ทำให้ได้รับความรู้ต่าง ๆ มากมายที่นำมาประยุกต์ใช้งานในโปรแกรม และได้เรียนรู้ทดลองสิ่งใหม่ ๆ ที่ไม่เคยทำมาก่อน เป็นการทำให้ได้ลองฝึกทักษะการออกแบบและเขียนโปรแกรมจากที่เรียนรู้มามาก ภายในห้องเรียนและภายนอกห้องเรียน นอกจากนี้ยังได้ฝึกทักษะในการแก้ไขปัญหาที่เกิดขึ้นต่าง ๆ ในระหว่างการพัฒนาโปรแกรม และเสริมสร้างแนวคิดในการแก้ไขปรับปรุงปัญหาที่พบเจอในชีวิตประจำวันอีกด้วย

### 6.4. แนวทางการพัฒนาต่อยอด

แนวทางในการพัฒนาต่อยอดที่ต้องการทำคือ การปรับปรุงคุณภาพหน้าตาของโปรแกรมให้มีความสวยงาม เรียบง่าย และใช้งานได้สะดวก เข้าใจง่ายมากขึ้นกว่าเดิม เพื่อยกระดับความพึงพอใจของผู้ใช้งานที่ได้รับจากโปรแกรม

การปรับปรุงประสิทธิภาพของโปรแกรมให้ดียิ่งขึ้นในการใช้งาน ไม่ว่าจะเป็นชุดคำสั่งในส่วนของหน้าบ้านที่มีการใช้ทรัพยากรลดลง โดยใช้งานชุดคำสั่งที่มีประสิทธิภาพดียิ่งขึ้น หรือชุดคำสั่งส่วนของหลังบ้านที่มีการออกแบบให้มีการใช้ทรัพยากรลดลง เช่นเดียวกัน เช่น การปรับการออกแบบฐานข้อมูลใหม่ให้ลดความซ้ำซ้อนลงกว่าเดิม มีระบบการทำงาน Caching ข้อมูลเพื่อเพิ่มความรวดเร็วในการตอบสนองตึงข้อมูลจากระบบ และลดการเข้าถึงฐานข้อมูลโดยไม่จำเป็น

และสุดท้ายคือการปรับปรุงเพิ่มเติมคุณสมบัติใหม่มากมาย เช่น

- การอกรายงานข้อมูลเพื่อนำไปต่อยอดใช้ประโยชน์สำหรับภายในร้านอาหารไม่ว่าจะเป็นรายงานประจำวัน เดือน ปี เกี่ยวกับข้อมูลของอาหาร หรือข้อมูลเกี่ยวกับลูกค้าที่เข้ามาใช้บริการ เพื่อให้สามารถช่วยในการตัดสินใจเพื่อจัดเตรียมสต็อกวัตถุติด หรือพนักงานที่ต้องทำงานได้
- การอกรายงานเกี่ยวกับบัญชีทางการเงินเพื่อให้สามารถยื่นเอกสารส่งกรมหรือธนาคารต่าง ๆ ได้
- การส่งออกข้อมูลที่สามารถนำไปวิเคราะห์ต่อด้วยเทคนิคการวิเคราะห์ข้อมูลของ Data Science ได้ เพื่อนำไปหาข้อมูลที่เป็นประโยชน์สำหรับร้านอาหาร
- สามารถเรียกพนักงานได้ผ่านหน้าเว็บของลูกค้า
- สามารถค้นหารายการอาหารในหน้าเมนูของลูกค้าได้
- สามารถเรียงลำดับอาหารภายในหมวดหมู่อาหารได้

- รองรับการเปลี่ยนตัวแปรสกุลเงินที่ปรากฏในระบบ
- รองรับระบบสมาชิกเพื่อสะสมแต้มสำหรับการใช้บริการร้านอาหาร
- รองรับการใช้งานส่วนลด ผ่านคูปองส่วนลด หรือส่วนลดจากพนักงานผ่านแคชเชียร์
- รองรับการคำนวณ Service Charge และ VAT
- รองรับการชำระเงินผ่านบัตรเครดิตและเดบิตผ่านแคชเชียร์ด้วยเครื่อง EDC
- รองรับการชำระเงินผ่านบัตรเครดิตและเดบิตด้วยตนเองผ่าน Payment Gateway ของธนาคาร
- รองรับการชำระเงินผ่าน Mobile Banking ด้วย Deeplink หลากหลายธนาคารมากขึ้น
- รองรับการชำระเงินผ่านบริการ Wallet ต่าง ๆ เช่น TrueMoney Wallet, Dolfin Wallet, Rabbit LINE Pay, ShopeePay รวมถึง PayPal

## บรรณานุกรม

- [1] pool13433. (2564). **Spring Boot ทำความรู้จัก เริ่มติดตั้งและเริ่มใช้งานเบื้องต้น EP1.** [ออนไลน์] เข้าถึงได้จาก PoolsawatBlogs: <https://www.poolsawat.com/spring-boot-ทำความรู้จัก-เริ่มติดตั้งและเริ่มใช้งานเบื้องต้น EP1/>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [2] Johnson, Rod, et al. (ม.ป.ป.). **Web MVC framework.** [ออนไลน์] เข้าถึงได้จาก Spring: <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [3] Talil. (2564). **API หรือ Application Programming Interface คืออะไร ?.** [ออนไลน์] เข้าถึงได้จาก Thaiware: <https://tips.thaiware.com/1728.html>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [4] วิจิตรา พรมจารย์. (2564). เรียนรู้คำสั่ง Sql ไปด้วยกัน step by step. [ออนไลน์] เข้าถึงได้จาก NSTRU BLOG: [https://race.nSTRU.ac.th/home\\_ex/blog/topic/show/2940](https://race.nSTRU.ac.th/home_ex/blog/topic/show/2940). (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [5] มนพ กองอุ่น. (2559). **Bootstrap คืออะไร?.** [ออนไลน์] เข้าถึงได้จาก Programmer Thailand: <https://www.programmerthailand.com/tutorial/post/view/96/bootstrap-คืออะไร>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [6] WEBDOODEE. (2563). **jQuery คืออะไร ใช้งานอย่างไร.** [ออนไลน์] เข้าถึงได้จาก WEBDOODEE: <https://www.webdodee.com/what-is-jquery/>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [7] JetBrains. (2564). **IntelliJ IDEA overview.** [ออนไลน์] เข้าถึงได้จาก JetBrains: <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [8] Waris Limtoprasert. (2564). **เคล็ด(ไม่)ลับการใช้ Postman - Big Data Thailand.** [ออนไลน์] เข้าถึงได้จาก Big Data Thailand: <https://bigdata.go.th/big-data-101/เคล็ดไม่ลับการใช้postman/>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).
- [9] SCB Developers. (ม.ป.ป.). **SCB Developers.** [ออนไลน์] เข้าถึงได้จาก SCB Developers: <https://developer.scb/>. (วันที่ค้นข้อมูล 5 กุมภาพันธ์ 2565).