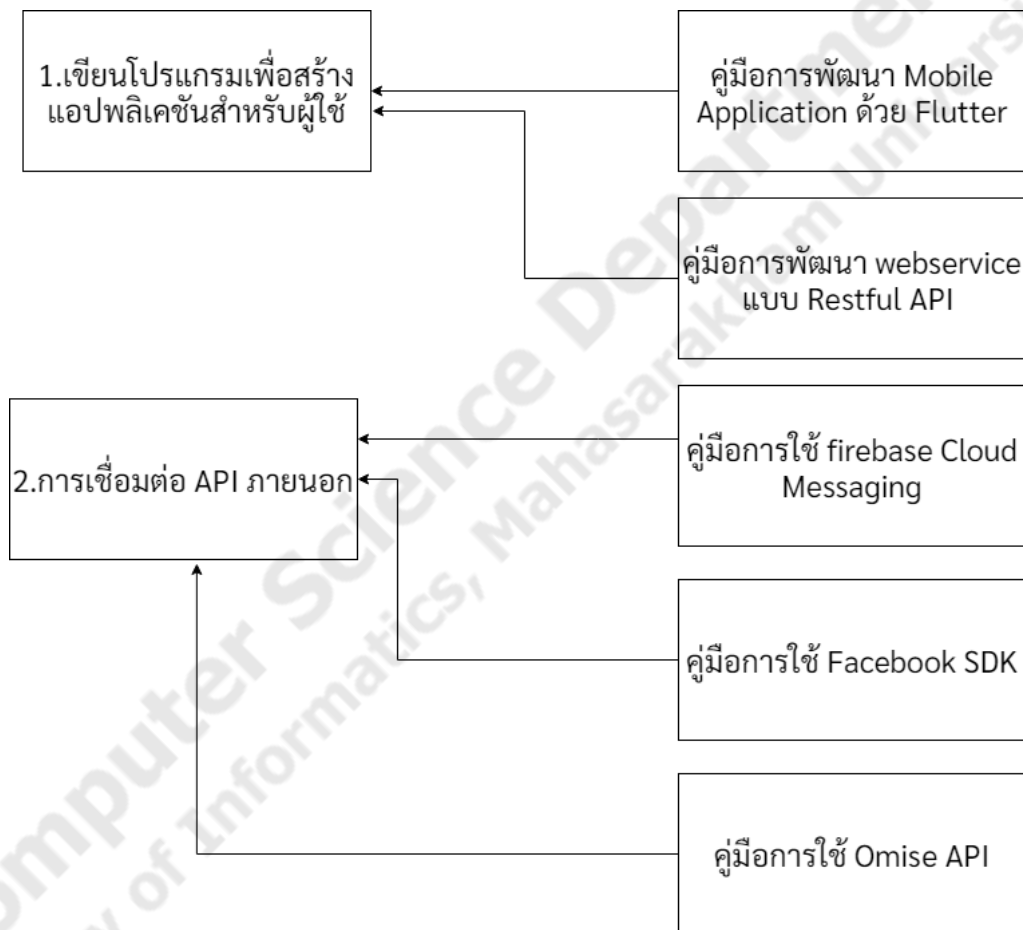


## บทที่ 3

### ขั้นตอนการดำเนินงานที่เกี่ยวข้อง

#### 3.1 กรอบดำเนินงาน



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบ

##### 3.1.1 คำอธิบาย

1.เขียนโปรแกรมเพื่อสร้าง แอปพลิเคชัน สำหรับผู้ใช้งานแอปพลิเคชันผ่านสมาร์ทโฟน  
โดยการสร้างแอปพลิเคชันจะต้องมีการติดตั้งเครื่องมือดังนี้

### 1). Visual Studio Code คือโปรแกรม Code Editor

ที่ใช้ในการแก้ไขและปรับแต่งโค้ดโดยมาจากค่ายไมโครซอฟท์ ที่มีการพัฒนาออกมาในรูปแบบของ open source จึงสามารถนำมาใช้งานได้แบบฟรีๆ และสามารถเขียนโค้ดได้หลากหลายภาษา ในการสร้างแอปพลิเคชันจะต้องติดตั้ง Flutter , Dart , Node JS สำหรับการเขียนแอปพลิเคชันใช้ dart สำหรับการรับส่งข้อมูลกับ server ใช้ Node js

เมื่อติดตั้งเสร็จแล้วทำการสร้างแอปพลิเคชันตามหน้า GUI ที่ได้ออกแบบไว้แล้วดังนี้

1. หน้าสมัครสมาชิก
2. หน้าเข้าสู่ระบบ
3. หน้าโปรไฟล์
4. หน้าเพิ่มข้อมูลอาหาร
5. หน้าหมวดหมู่อาหาร
6. หน้าแสดงฟีด
7. หน้าค้นหา
8. หน้าแสดงข้อมูลการค้นหา
9. หน้าแสดงข้อมูลอาหาร
10. หน้าการซื้อขายสูตรอาหาร
11. หน้าเติมเงิน-ถอนเงินในระบบ
12. หน้าแจ้งเตือนกิจกรรมต่าง ๆ ในระบบ

### 2) ต้องมีการเชื่อมต่อกับ API ภายนอก เพื่อให้แอปพลิเคชันทำงานได้อย่างสมบูรณ์ดังนี้

1. Firebase Cloud Messaging (FCM) คือ บริการส่งข้อความแจ้งเตือนแบบข้ามแพลตฟอร์ม (cross-platform messaging) ทั้ง Android, iOS และ Web แบบฟรีๆ โดยในการเขียนโปรแกรมด้วย flutter จะต้องลงไลบรารีชื่อ firebase\_messaging

#### 2. Facebook

SDK คือชุดอุปกรณ์ที่ช่วยให้แอปพลิเคชันแอนดรอยด์ของเราสามารถเรียกใช้บริการของ facebook ได้

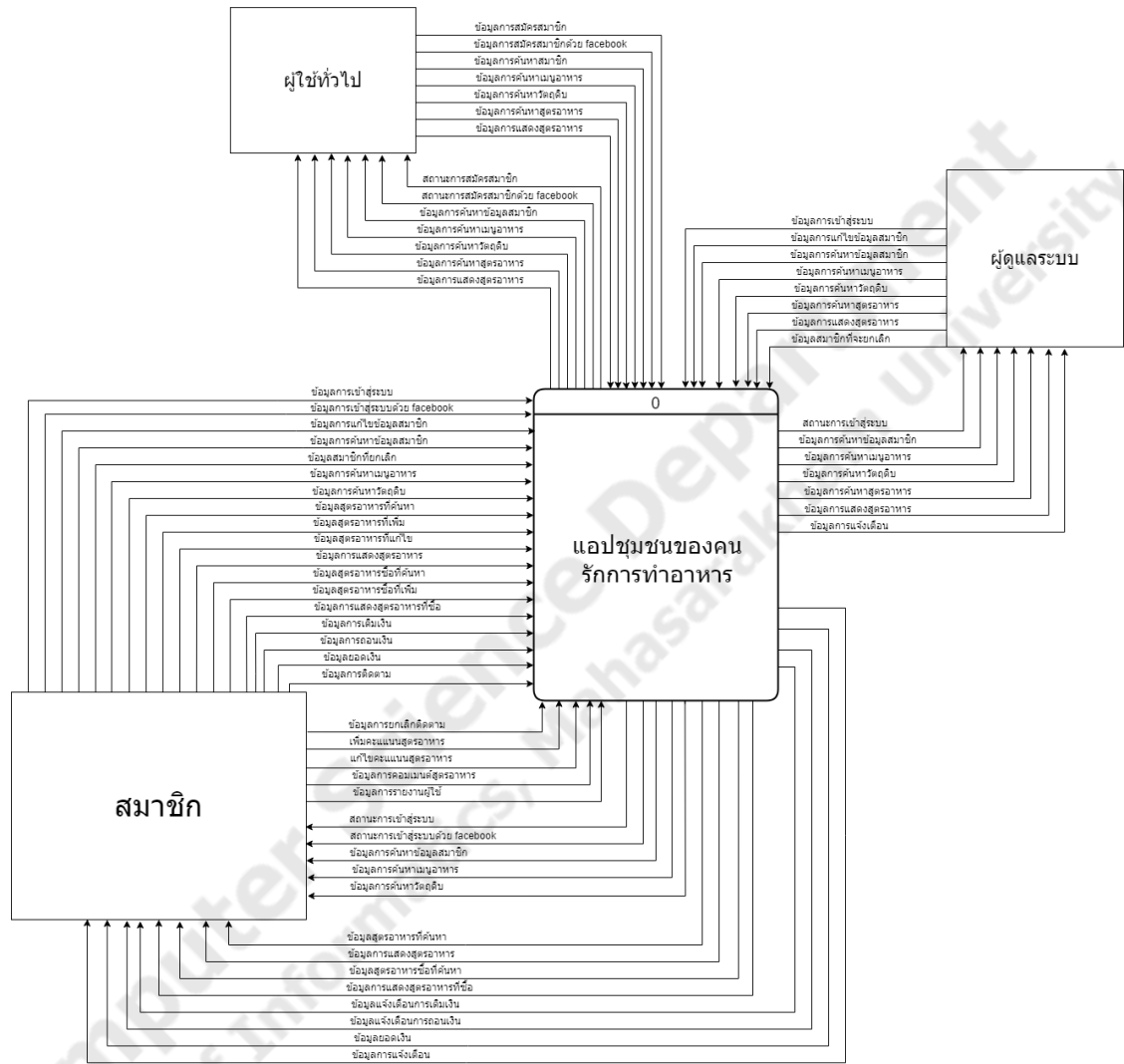
#### 3. พร้อมเพย์ Omise

เพิ่มช่องทางการรับชำระเงินผ่านพร้อมเพย์ อำนวยความสะดวกให้ผู้ซื้อสามารถทำรายการผ่านบัญชีธนาคารชั้นนำทั่วประเทศ

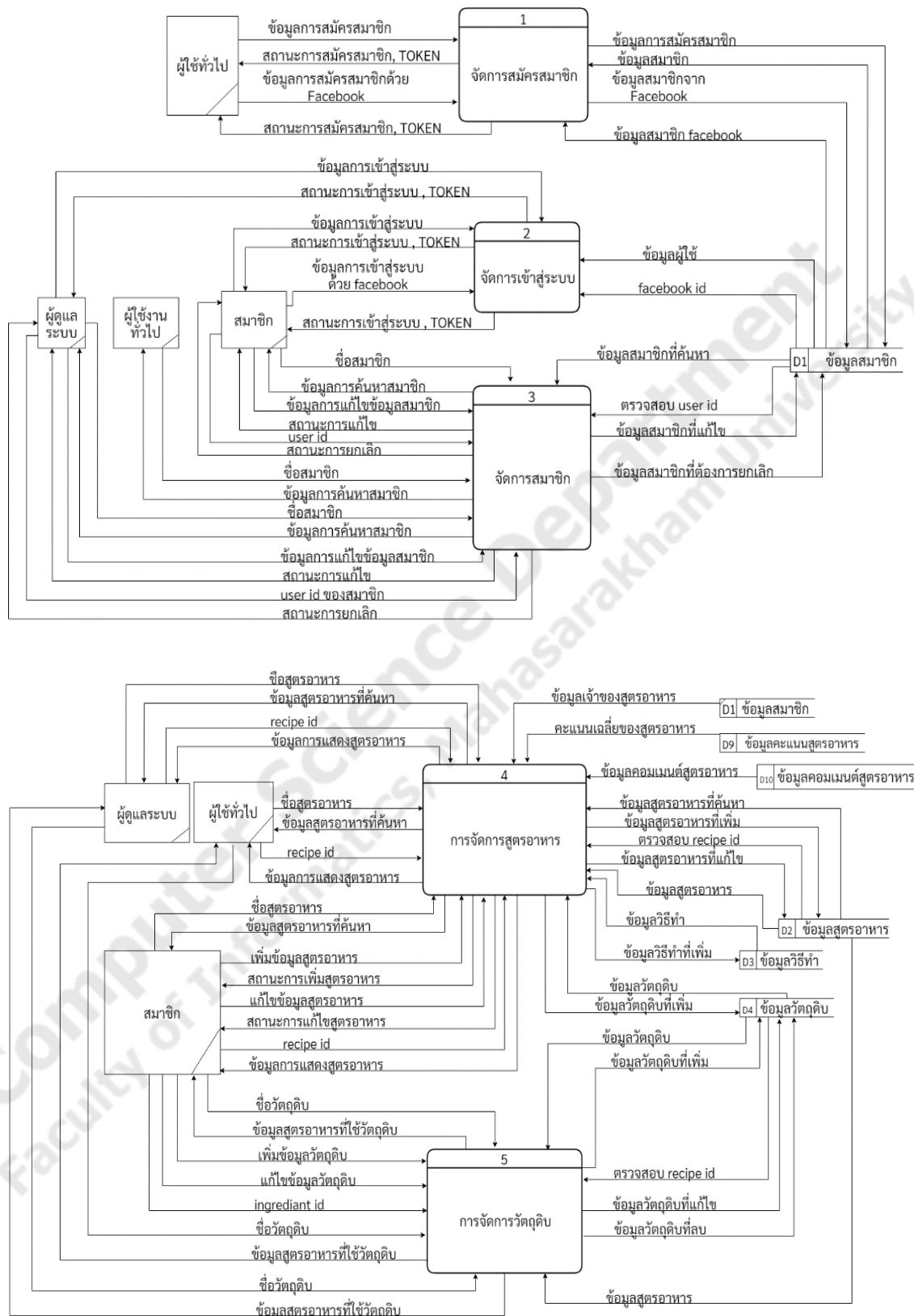
บริการพร้อมเพย์ช่วยให้ร้านค้ารับชำระเงินจากผู้ซื้อทั่วประเทศไทยได้อย่างรวดเร็วและปลอดภัย ผู้ซื้อสามารถทำรายการได้ง่ายเพียงสแกนคิวอาร์โค้ดก็สามารถชำระเงินได้ทันที

### 3.2 การออกแบบระบบ

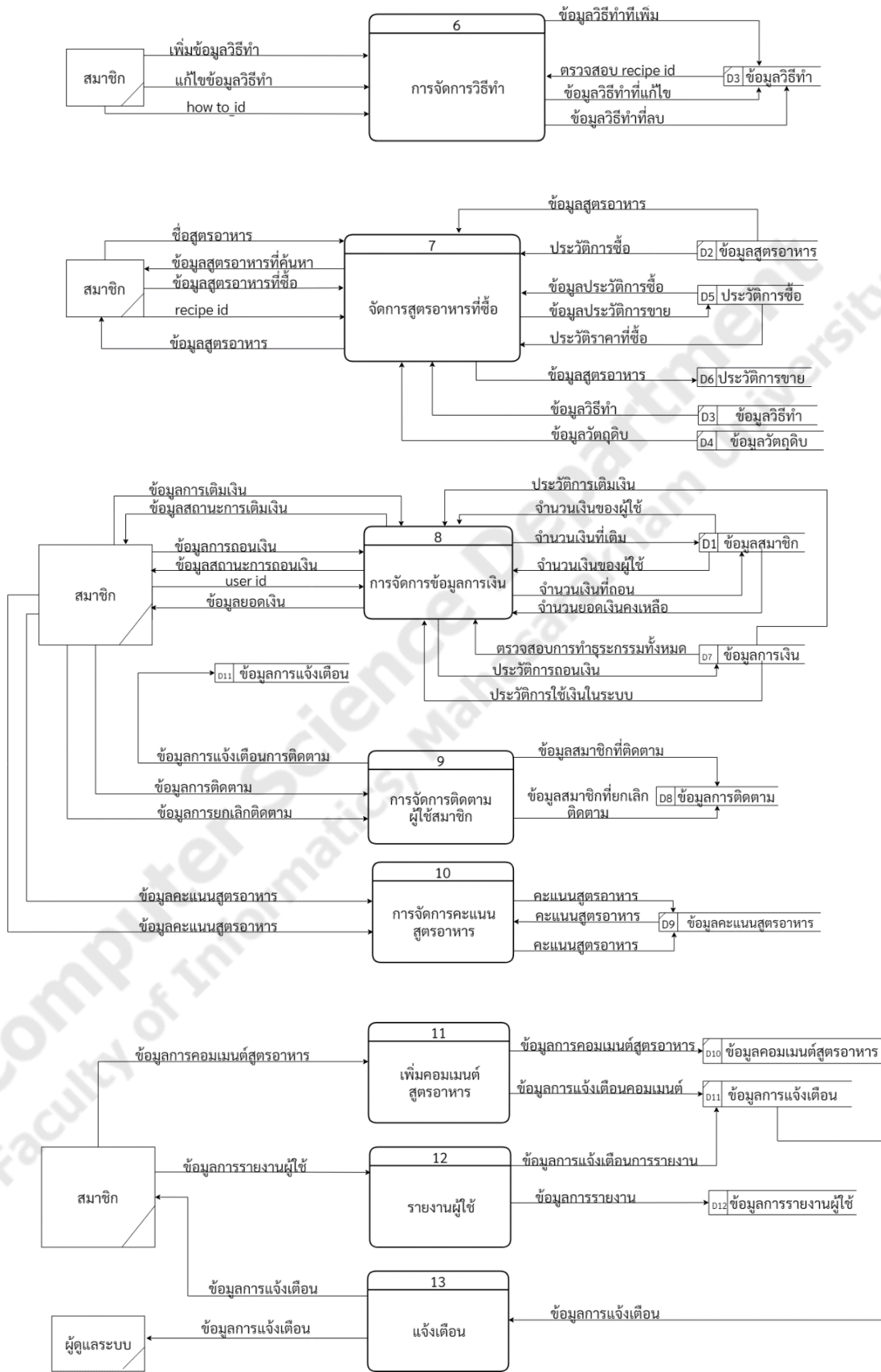
#### 3.2.1 แผนภาพการไหลของข้อมูล (Data Flow Diagram)



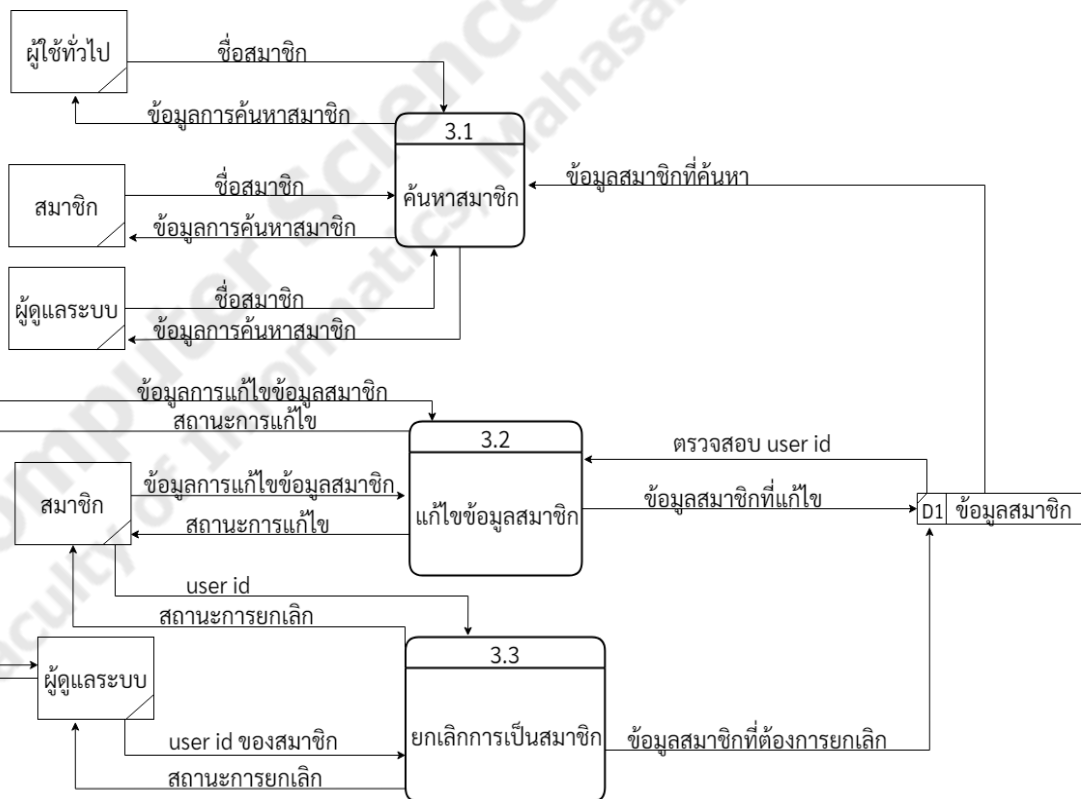
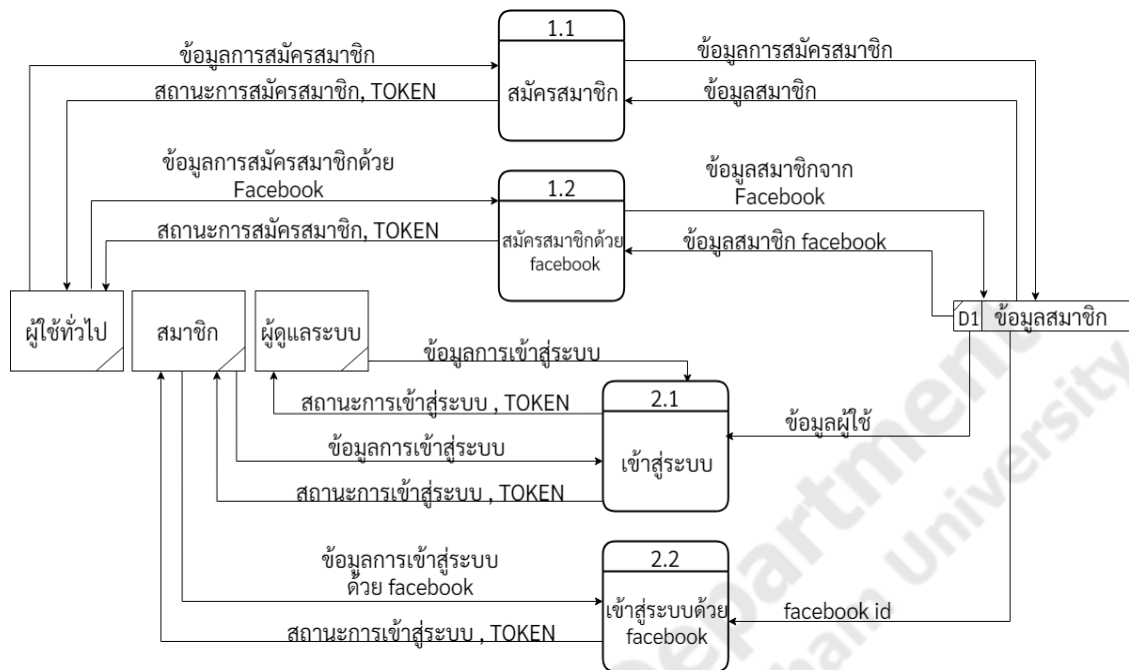
ภาพประกอบที่ 3.2 Context Diagram



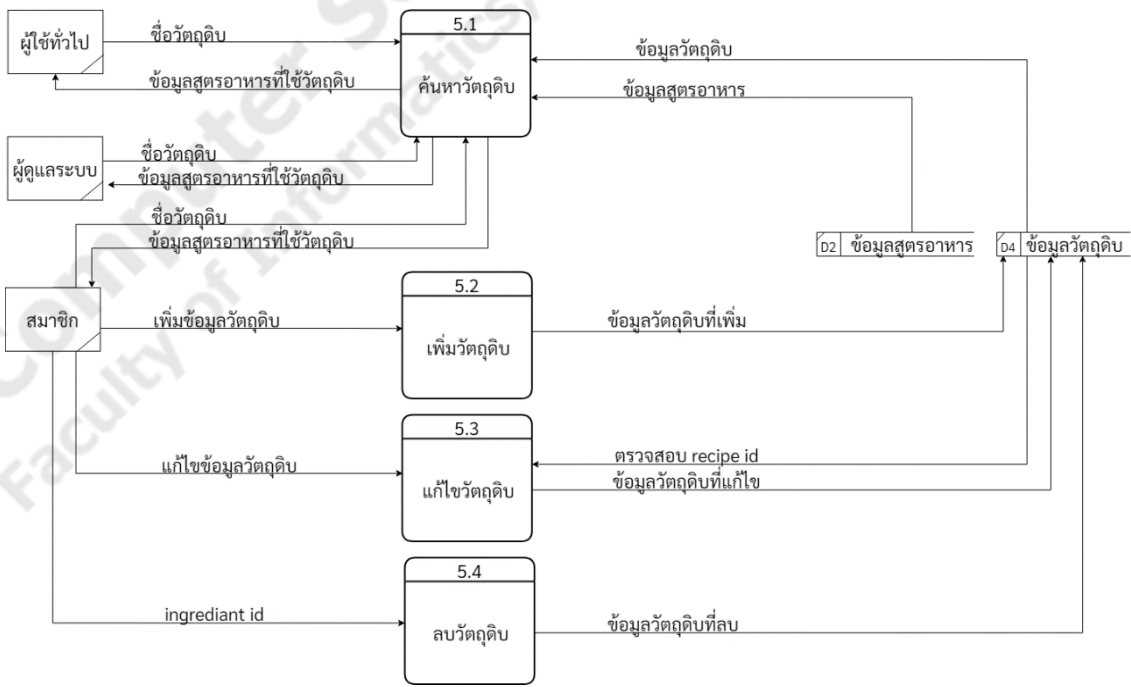
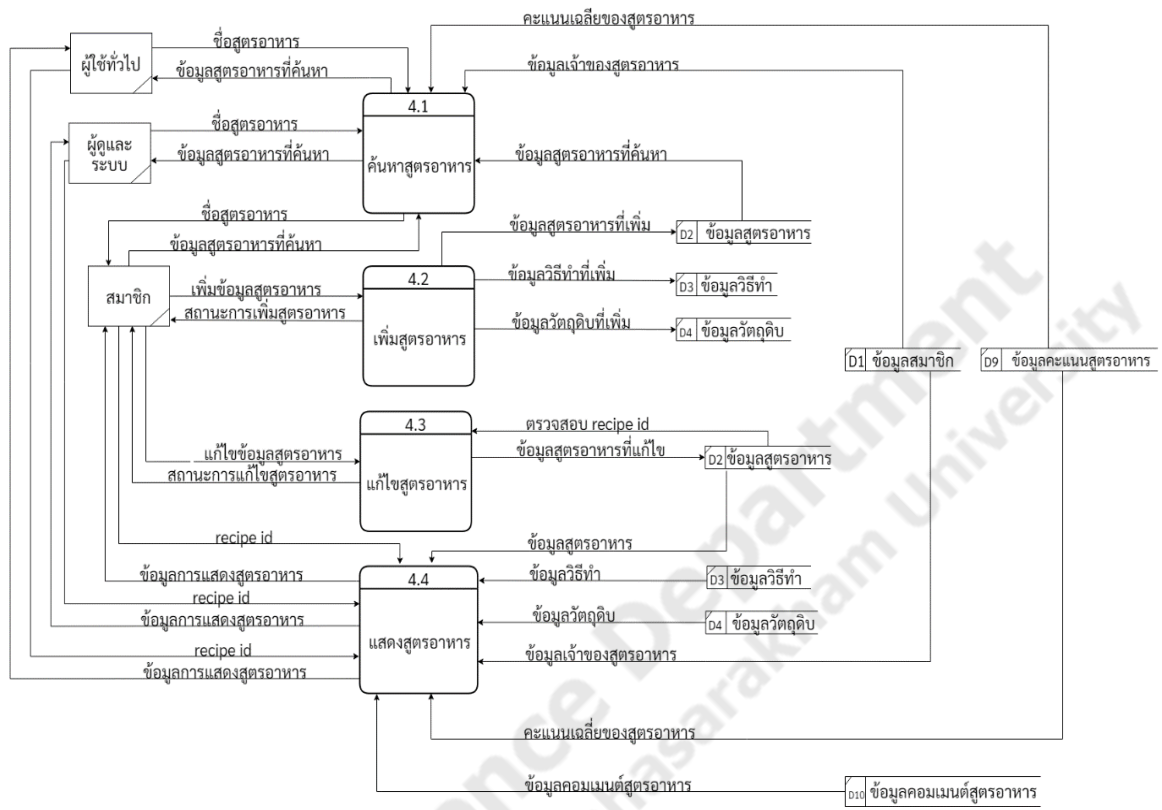
ภาพประกอบที่ 3.3 Data Flow Diagram Level 1



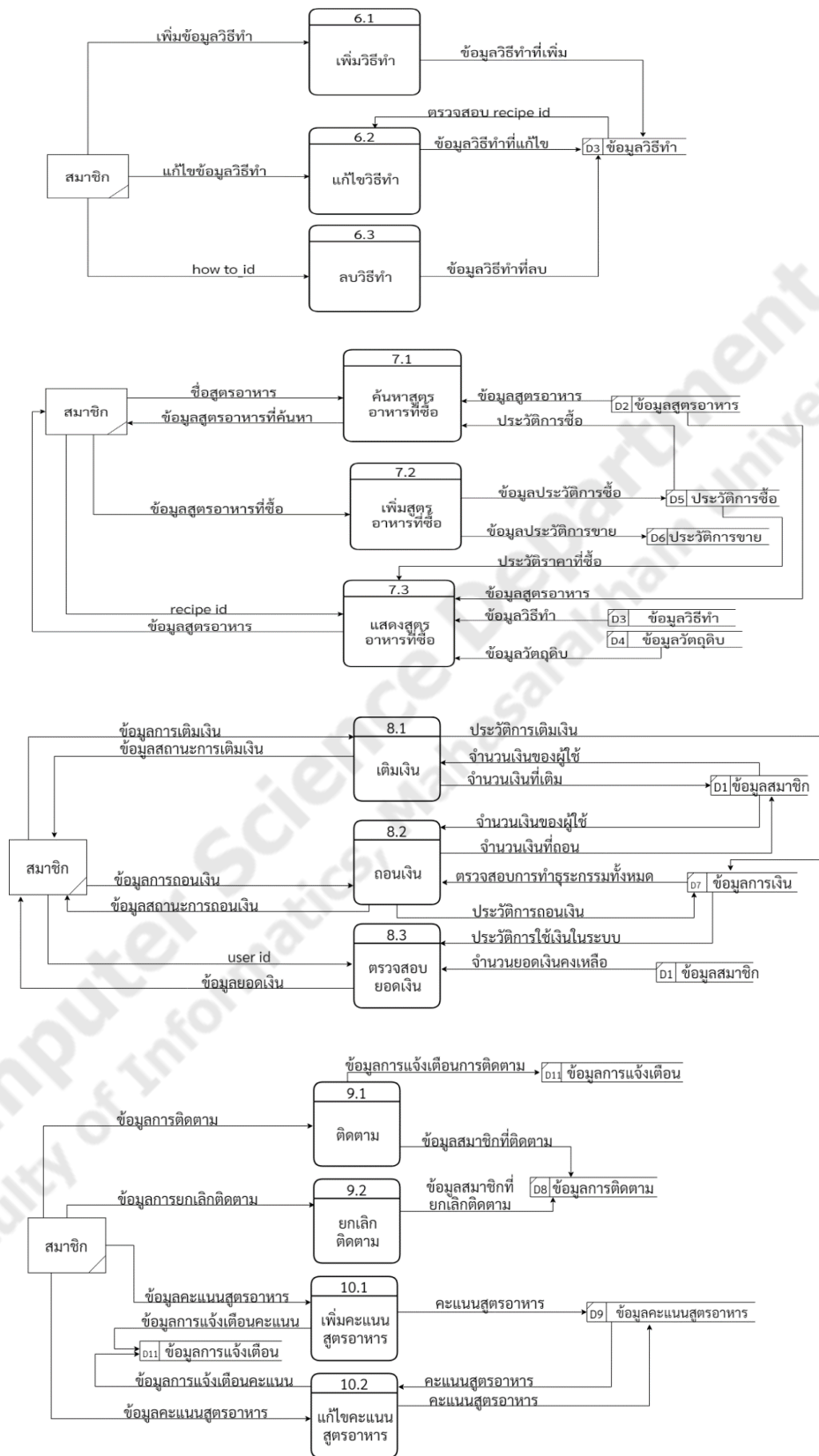
ภาพประกอบที่ 3.3 Data Flow Diagram Level 1 (ต่อ)



ภาพประกอบที่ 3.4 Data Flow Diagram Level 2



ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 (ต่อ)



ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 (ต่อ)



### 3.3 External Entity Description

พจนานุกรมข้อมูลช่วยอธิบายการทำงานของแผนภาพการไหลของข้อมูล (Data Flow Dictionaries)

ตารางที่ 3.1 External Entity Description

Name	Description	Input Data Flow	Output Data Flow
ผู้ใช้ทั่วไป	ผู้ใช้งานทั่วไปที่สามารถใช้ในบางส่วนของแอปพลิเคชัน	-ข้อมูลการสมัครสมาชิก -ข้อมูลการสมัครสมาชิกด้วย Facebook -ข้อมูลการเข้าสู่ระบบ -ข้อมูลการเข้าสู่ระบบด้วย Facebook -ข้อมูลการค้นหาข้อมูลสมาชิก -ข้อมูลการค้นหาสูตรอาหาร -ข้อมูลการแสดงสูตรอาหาร	-สถานะการสมัครสมาชิก -สถานะการสมัครสมาชิกด้วย Facebook -สถานะการเข้าสู่ระบบ -สถานะการเข้าสู่ระบบด้วย Facebook -ข้อมูลการค้นหาข้อมูลสมาชิก -ข้อมูลการค้นหาสูตรอาหาร -ข้อมูลการแสดงสูตรอาหาร
สมาชิก	ผู้ใช้งานที่เป็นสมาชิกสามารถเข้าใช้งานแอปพลิเคชันได้	-ข้อมูลการเข้าสู่ระบบ -ข้อมูลการเข้าสู่ระบบด้วย Facebook -ข้อมูลการแก้ไขข้อมูลสมาชิก -ข้อมูลการค้นหาข้อมูลสมาชิก -ข้อมูลสมาชิกที่ยกเลิก -ข้อมูลสูตรอาหารที่ค้นหา -ข้อมูลสูตรอาหารที่เพิ่ม -ข้อมูลสูตรอาหารที่แก้ไข -ข้อมูลการแสดงสูตรอาหาร -ข้อมูลสูตรอาหารซื้อที่ค้นหา -ข้อมูลสูตรอาหารซื้อที่เพิ่ม -ข้อมูลการถอนเงิน -ข้อมูลยอดเงิน -ข้อมูลการติดตาม -ข้อมูลการยกเลิกติดตาม	-สถานะการเข้าสู่ระบบ -สถานะการเข้าสู่ระบบด้วย Facebook -ข้อมูลการค้นหาข้อมูลสมาชิก -ข้อมูลสูตรอาหารที่ค้นหา -ข้อมูลการแสดงสูตรอาหาร -ข้อมูลสูตรอาหารซื้อที่ค้นหา -ข้อมูลการแสดงสูตรอาหารที่ซื้อ -ข้อมูลยอดเงิน -ข้อมูลการแจ้งเตือน

ตารางที่ 3.1 External Entity Description (ต่อ)

Name	Description	Input Data Flow	Output Data Flow
		-เพิ่มคะแนนสูตรอาหาร -แก้ไขคะแนนสูตรอาหาร -ข้อมูลการคอมเมนต์สูตรอาหาร -ข้อมูลการรายงานผู้ใช้	
ผู้ดูแลระบบ	ผู้ดูแลระบบที่มีหน้าที่จัดการข้อมูลต่าง ๆ	-ข้อมูลการเข้าสู่ระบบ -ข้อมูลการแก้ไขข้อมูลส่วนตัว -ข้อมูลการค้นหาข้อมูลสมาชิก -ข้อมูลสูตรอาหารที่ค้นหา -ข้อมูลการแสดงสูตรอาหาร -ข้อมูลสมาชิกที่จะยกเลิก	-สถานะการเข้าสู่ระบบ -ข้อมูลการค้นหาข้อมูลสมาชิก -ข้อมูลการค้นหาสูตรอาหาร -ข้อมูลการแสดงผลสูตรอาหาร -ข้อมูลการแจ้งเตือน

## 3.4 Data Flow and Data Structure Description

ตารางที่ 3.2 Data Flow and Data Structure Description

Name	Description	Source	Destination	Data Structure
ข้อมูลการสมัครสมาชิก	รายละเอียดข้อมูลผู้ใช้ที่ต้องการสมัครสมาชิก	ผู้ใช้ทั่วไป	Process 1.1 สมัครสมาชิก	รหัสสมาชิก+ชื่อ ผู้ใช้+รหัสผ่าน+ชื่อนามสกุล+(ชื่อนามแฝง)+(รูปโปรไฟล์)+สถานะ
	ตรวจสอบและเพิ่มข้อมูลผู้ใช้เข้าฐานข้อมูล	Process 1.1 สมัครสมาชิก	D1 ข้อมูลสมาชิก	ผู้ใช้งาน+สถานะการ ใช้งาน+ยอดเงินในระบบ
สถานะการสมัครสมาชิก	แจ้งสถานะการสมัครสมาชิก	Process 1 สมัครสมาชิก	ผู้ใช้ทั่วไป	สถานะ+token
ข้อมูลการสมัครสมาชิก	รายละเอียดข้อมูลผู้ใช้ที่ต้องการสมัครสมาชิก	ผู้ใช้ทั่วไป	Process 1.2 สมัครสมาชิก ด้วย Facebook	รหัสสมาชิก+ Facebook_id+ รหัสผ่าน+ชื่อนามสกุล

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
				+(ชื่อนามแฝง)+(รูปโปรไฟล์)+สถานะผู้ใช้งาน+สถานะการใช้จ่ายเงินในระบบ
	ตรวจสอบและเพิ่มข้อมูลผู้ใช้เข้าฐานข้อมูล	Process 1.2 สมัครสมาชิก ด้วย Facebook	D1 ข้อมูลสมาชิก	
สถานะการสมัครสมาชิก	แจ้งสถานะการสมัครสมาชิก	Process 1.2 สมัครสมาชิก ด้วย Facebook	ผู้ใช้ทั่วไป	สถานะ+token
ข้อมูลการเข้าสู่ระบบ	รายละเอียดข้อมูลผู้ใช้ที่ต้องการเข้าสู่ระบบ	สมาชิก	Process 2.1 เข้าสู่ระบบ	ชื่อผู้ใช้+รหัสผ่าน
ข้อมูลผู้ใช้	การดึงข้อมูลสมาชิก	D1 ข้อมูลสมาชิก	Process 2.1 เข้าสู่ระบบ	
ข้อมูลการเข้าสู่ระบบ	รายละเอียดข้อมูลผู้ใช้ที่ต้องการเข้าสู่ระบบ	ผู้ดูแลระบบ	Process 2 เข้าสู่ระบบ	
สถานะการเข้าสู่ระบบ	แจ้งสถานะการเข้าสู่ระบบ	Process 2.1 เข้าสู่ระบบ	สมาชิก	สถานะ+token
	แจ้งสถานะการเข้าสู่ระบบ		ผู้ดูแลระบบ	สถานะ+token
ข้อมูลการเข้าสู่ระบบ	รายละเอียดข้อมูลผู้ใช้ที่ต้องการเข้าสู่ระบบ	สมาชิก	Process 2.2 เข้าสู่ระบบ ด้วย Facebook	Facebook_id

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลผู้ใช้	การดึงข้อมูลสมาชิก	D1 ข้อมูลสมาชิก	Process 2.2 เข้าสู่ระบบ ด้วย Facebook	Facebook_id
สถานะการเข้าสู่ระบบ	แจ้งสถานะการเข้าสู่ระบบ	Process 2.2 เข้าสู่ระบบ ด้วย Facebook	สมาชิก	สถานะ+token
ชื่อสมาชิก	รายละเอียดข้อมูล ผู้ใช้ที่ต้องการค้นหา ข้อมูลสมาชิก	ผู้ใช้ทั่วไป สมาชิก ผู้ดูแลระบบ	Process 3.1 ค้นหาสมาชิก	[ชื่อนามสกุล   ชื่อ นามแฝง]
ข้อมูลสมาชิกที่ค้นหา	การดึงข้อมูลสมาชิก	D1 ข้อมูลสมาชิก	Process 3.1 ค้นหาสมาชิก	
ข้อมูลการค้นหา ข้อมูลสมาชิก	รายละเอียดข้อมูล ผู้ใช้ที่ต้องการค้นหา ข้อมูลสมาชิก	Process 3.1 ค้นหาสมาชิก	ผู้ใช้ทั่วไป สมาชิก ผู้ดูแลระบบ	[ชื่อนามสกุล   ชื่อ นามแฝง] + {รหัส สมาชิก+รูปโปรไฟล์}
ข้อมูลการแก้ไข ข้อมูลสมาชิก	รายละเอียดข้อมูล ผู้ใช้ที่ต้องการแก้ไข ข้อมูลสมาชิก	สมาชิก ผู้ดูแลระบบ	Process 3.2 แก้ไขข้อมูล สมาชิก	รหัสสมาชิก+ชื่อ นามสกุล + ชื่อ นามแฝง+รูปโปรไฟล์
สถานะการแก้ไข	แก้ไขและบันทึก ข้อมูลสมาชิก	Process 3.2 แก้ไขข้อมูล สมาชิก	D1 ข้อมูล สมาชิก	สถานะการแก้ไข
User id	รายละเอียดข้อมูล ผู้ใช้ที่ต้องการยกเลิก การเป็นสมาชิก	สมาชิก	Process 3.3 ยกเลิกการ เป็นสมาชิก	รหัสสมาชิก
สถานะการ ยกเลิก	แก้ไขและบันทึก ข้อมูลสมาชิก	Process 3.3 ยกเลิกการ เป็นสมาชิก	D1 ข้อมูล สมาชิก	สถานะการยกเลิก

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
User id ของสมาชิก	รายละเอียดข้อมูลสมาชิกที่ต้องการยกเลิกการเป็นสมาชิก	ผู้ดูแลระบบ	Process 3.3 ยกเลิกการเป็นสมาชิก	รหัสสมาชิก+รหัสสมาชิกที่จะยกเลิก
สถานะการยกเลิก	แก้ไขและบันทึกข้อมูลสมาชิก	Process 3.3 ยกเลิกการเป็นสมาชิก	D1 ข้อมูลสมาชิก	สถานะการยกเลิก
ชื่อสูตรอาหาร	รายละเอียดข้อมูลผู้ใช้ที่ต้องการค้นหาข้อมูลสูตรอาหาร	ผู้ใช้ทั่วไป	Process 4.1 ค้นหาสูตรอาหาร	ชื่อสูตรอาหาร
		สมาชิก		
		ผู้ดูแลระบบ		
ข้อมูลสูตรอาหารที่ค้นหา	การดึงข้อมูลสูตรอาหาร	D2 ข้อมูลสูตรอาหาร	รหัสสูตรอาหาร+ชื่อสูตรอาหาร+รูปภาพ	
ข้อมูลเจ้าของสูตรอาหาร	การดึงข้อมูลสมาชิก	D1 ข้อมูลสมาชิก		รหัสสมาชิก+ชื่อ-นามสกุล+นามแฝง+รูปโปรไฟล์
คะแนนเฉลี่ยของสูตรอาหาร	การดึงข้อมูลคะแนน	D9 ข้อมูลคะแนนสูตรอาหาร	Process 4.1 ค้นหาสูตรอาหาร	คะแนนเฉลี่ย+จำนวนคนที่ให้คะแนน
ข้อมูลสูตรอาหารที่ค้นหา	รายละเอียดข้อมูลผู้ใช้ที่ต้องการค้นหาข้อมูลสูตรอาหาร	Process 4.1 ค้นหาสูตรอาหาร	ผู้ใช้ทั่วไป	รหัสสูตรอาหาร+ชื่อสูตรอาหาร+รูปภาพ รหัสสมาชิก+ชื่อ-นามสกุล+นามแฝง+รูปโปรไฟล์+คะแนนเฉลี่ย+จำนวนคะแนน
			สมาชิก	
			ผู้ดูแลระบบ	
เพิ่มข้อมูลสูตรอาหาร	รายละเอียดข้อมูลผู้ใช้ที่ต้องการเพิ่มข้อมูลสูตรอาหาร	สมาชิก	Process 4.2 เพิ่มสูตรอาหาร	รหัสสูตรอาหาร+ชื่อสูตรอาหาร+ราคา+สำหรับกี่คน+

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
				เวลาที่ใช้+ประเภท+ คำอธิบาย+รูปภาพ+ [รหัสวัตถุดิบ+ชื่อ วัตถุดิบ+จำนวน+ ลำดับ]+[รหัสวิธีทำ+ คำอธิบาย+ลำดับ+ path_file+ประเภท ของไฟล์]+รหัส สมาชิก
ข้อมูลสูตร อาหารที่เพิ่ม	บันทึกข้อมูลสูตร อาหาร	Process 4.2 เพิ่มสูตร อาหาร	D2 ข้อมูลสูตร อาหาร	รหัสสูตรอาหาร+ชื่อ สูตรอาหาร+ราคา+ สำหรับกี่คน+เวลาที่ ใช้+ประเภท+ คำอธิบาย+รูปภาพ+ รหัสสมาชิก
ข้อมูลวิธีทำ	บันทึกข้อมูลวิธีทำ	Process 4.2 เพิ่มสูตร อาหาร	D3 ข้อมูลวิธี ทำ	รหัสวิธีทำ+ คำอธิบาย+ลำดับ+ path_file+ประเภท ของไฟล์+รหัสสูตร อาหาร
ข้อมูลวัตถุดิบ	บันทึกข้อมูลวัตถุดิบ		D4 ข้อมูล วัตถุดิบ	รหัสวัตถุดิบ+ชื่อ วัตถุดิบ+จำนวน+ ลำดับ+รหัสสูตร อาหาร
สถานะการเพิ่ม สูตรอาหาร	รายงานสถานะการ เพิ่มสูตรอาหาร		สมาชิก	สถานะการ ดำเนินการ
แก้ไขข้อมูลสูตร อาหาร	รายละเอียดข้อมูล ผู้ใช้ที่ต้องการแก้ไข ข้อมูลสูตรอาหาร	สมาชิก	Process 4.3 แก้ไขสูตร อาหาร	รหัสสูตรอาหาร+ชื่อ สูตรอาหาร+ราคา+ สำหรับกี่คน+

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
				เวลาที่ใช้+ประเภท+ คำอธิบาย+รูปภาพ+ รหัสสมาชิก
ตรวจสอบ recipe id	ดึงข้อมูลสูตรอาหาร จากรหัสสูตรอาหาร มาใช้เปรียบเทียบกัน	Process 4.3 แก้ไขสูตร อาหาร	D2 ข้อมูลสูตร อาหาร	ข้อมูลสูตรอาหาร
ข้อมูลสูตร อาหารที่แก้ไข	แก้ไขบันทึกข้อมูล สูตรอาหาร	Process 4.3 แก้ไขสูตร อาหาร	D2 ข้อมูลสูตร อาหาร	รหัสสูตรอาหาร+ชื่อ สูตรอาหาร+ราคา+ สำหรับกี่คน+เวลาที่ ใช้+ประเภท+ คำอธิบาย+รูปภาพ+ รหัสสมาชิก
สถานะการ แก้ไขสูตร อาหาร	แจ้งสถานะการแก้ไข สูตรอาหาร		สมาชิก	สถานะ
Recipe id	แสดงข้อมูลสูตร อาหาร	Process 4.4 แสดงสูตร อาหาร	ผู้ใช้ทั่วไป สมาชิก ผู้ดูแลระบบ	รหัสสูตรอาหาร
ข้อมูลสูตร อาหาร	การดึงข้อมูลสมาชิก	D1 ข้อมูล สมาชิก	Process 4.4 แสดงสูตร อาหาร	รหัสสมาชิก
ข้อมูลวิธีทำ	ดึงข้อมูลวิธีทำ	Process 4.4 แสดงสูตร อาหาร	D4 ข้อมูลวิธี ทำ	รหัสวิธีทำ+ คำอธิบาย+ลำดับ+ path_file+ประเภท ของไฟล์+รหัสสูตร อาหาร
ข้อมูลวัตถุดิบ	ดึงข้อมูลวัตถุดิบ		D3 ข้อมูล วัตถุดิบ	รหัสวัตถุดิบ+ชื่อ วัตถุดิบ+จำนวน+ ลำดับ+รหัสสูตร

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลเจ้าของ สูตร	ดึงข้อมูลสมาชิก		D1 ข้อมูล สมาชิก	รหัสสมาชิก+ชื่อ นามสกุล+ชื่อ นามแฝง+รูปโปรไฟล์
คะแนนเฉลี่ย ของสูตรอาหาร	ดึงข้อมูลคะแนนเฉลี่ย ของสูตรอาหาร	Process 4.4 แสดงสูตร อาหาร	D9 ข้อมูล คะแนนสูตร อาหาร	คะแนนเฉลี่ย+ จำนวนคนที่ให้ คะแนน
ข้อมูลคอม เมนต์สูตร อาหาร	ดึงข้อมูลคอมเมนต์		D10 ข้อมูล คอมเมนต์สูตร อาหาร	รหัสสมาชิก+ชื่อ นามสกุล+ชื่อ นามแฝง+รูปโปร ไฟล์+ข้อความ+เวลา
ข้อมูลการแสดงผล สูตรอาหาร	รายละเอียดการแสดงผล สูตรอาหาร		ผู้ใช้ทั่วไป	รหัสสมาชิก+ชื่อ
			สมาชิก	นามสกุล+ชื่อ
			ผู้ดูแลระบบ	นามแฝง+รูปโปร ไฟล์+[รหัสวัตถุติดบ+ ชื่อวัตถุติดบ+จำนวน+ ลำดับ+รหัสสูตร อาหาร]+[รหัสวิธีทำ+ คำอธิบาย+ลำดับ+ path_file+ประเภท ของไฟล์+รหัสสูตร อาหาร]+คะแนน เฉลี่ย+จำนวนคนที่ให้ คะแนน+[รหัส สมาชิก+ชื่อ นามสกุล+ชื่อ นามแฝง+รูปโปร ไฟล์+ข้อความ+ เวลา]



ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ชื่อวัตถุดิบ	รายละเอียดชื่อ การ ค้นหาวัตถุดิบ	ผู้ใช้ทั่วไป	Process 5.1	ชื่อวัตถุดิบ
		สมาชิก	ค้นหาวัตถุดิบ	
		ผู้ดูแลระบบ		
ข้อมูลสูตร อาหารที่ใช้ วัตถุดิบ	รายละเอียดชื่อ การ ค้นหาวัตถุดิบที่ค้นหา	Process 5.1 ค้นหาวัตถุดิบ	ผู้ใช้ทั่วไป	รหัสวัตถุดิบ+ชื่อ
			สมาชิก	วัตถุดิบ+รหัสสูตร
			ผู้ดูแลระบบ	อาหาร
เพิ่มข้อมูล วัตถุดิบ	รายละเอียดการเพิ่ม ข้อมูลวัตถุดิบ	Process 5.2 เพิ่มวัตถุดิบ	สมาชิก	รหัสวัตถุดิบ+ชื่อ วัตถุดิบ+จำนวน+ ลำดับ
แก้ไขข้อมูล วัตถุดิบ	รายละเอียดการแก้ไข ข้อมูลวัตถุดิบ	Process 5.3 แก้ไขวัตถุดิบ	สมาชิก	ชื่อวัตถุดิบ+จำนวน+ ลำดับ
Ingredient id	ลบข้อมูลวัตถุดิบ	Process 5.4 ลบวัตถุดิบ	สมาชิก	รหัสวัตถุดิบ
เพิ่มข้อมูลวิธีทำ	รายละเอียดการเพิ่ม ข้อมูลวิธีทำ	Process 6.1 เพิ่มวิธีทำ	สมาชิก	รหัสวิธีทำ+ คำอธิบาย+ลำดับ+ path_file+ประเภท ของไฟล์+รหัสสูตร อาหาร
แก้ไขข้อมูลวิธี ทำ	รายละเอียดการแก้ไข ข้อมูลวิธีทำ	Process 6.2 แก้ไขวิธีทำ	สมาชิก	คำอธิบาย+ลำดับ+ path_file+ประเภท ของไฟล์+รหัสสูตร อาหาร
How to id	ลบข้อมูลวิธีทำ	Process 6.3 ลบวิธีทำ	สมาชิก	รหัสวิธีทำ
ชื่อสูตรอาหาร	รายละเอียดข้อมูล ผู้ใช้ที่ต้องการค้นหา ข้อมูลสูตรอาหารที่ชื่อ	สมาชิก	Process 7.1 ค้นหาสูตร อาหารที่ชื่อ	รหัสสูตรอาหาร+ชื่อ สูตรอาหาร+รหัส สมาชิก

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลประวัติ การซื้อ	ดึงข้อมูลประวัติการ ซื้อเพื่อนำมาตรวจ ตรวจสอบการซื้อ	Process 7.1 ค้นหาสูตร อาหารที่ซื้อ	D5 ประวัติ การซื้อ	รหัสการซื้อ+รหัส สูตรอาหาร+รหัส สมาชิก+เวลา+ราคา
ข้อมูลสูตร อาหาร	ดึงข้อมูลสูตรอาหาร		D2 ข้อมูลสูตร อาหาร	รหัสสูตรอาหาร+ชื่อ สูตรอาหาร+รูปภาพ
ข้อมูลสูตร อาหารที่ค้นหา	รายละเอียดการ ค้นหาสูตรอาหารที่ ซื้อ		สมาชิก	รหัสสูตรอาหาร+ชื่อ สูตรอาหาร+ รูปภาพ+รหัสการซื้อ
ข้อมูลสูตร อาหารที่ซื้อ	รายละเอียดการซื้อ สูตรอาหาร	สมาชิก	Process 7.2 เพิ่มสูตร อาหาร	สูตรอาหาร+รหัส สมาชิก+ราคา
ข้อมูลประวัติ การซื้อ	บันทึกข้อมูลการซื้อ สูตรอาหาร	Process 7.2 เพิ่มสูตร อาหาร	D5 ประวัติ การซื้อ	รหัสการซื้อ+รหัส สูตรอาหาร+รหัส สมาชิก+เวลา+ราคา
ข้อมูลประวัติ การขาย	บันทึกข้อมูลการขาย สูตรอาหาร		D6 ประวัติ การขาย	รหัสการขาย+รหัส สมาชิก+รหัสพาท เนอร์+รหัสสูตร อาหาร+เวลา+ราคา
Recipe id	รายละเอียดการแสดง ข้อมูลสูตรอาหารที่ซื้อ	สมาชิก	Process 7.3	รหัสสูตรอาหาร+ รหัสสมาชิก
ข้อมูลสูตร อาหาร	ดึงข้อมูลสูตรอาหารที่ ซื้อพร้อมวัตถุดิบและ วิธีทำและแสดงราคา ที่ซื้อ	Process 7.3 แสดงสูตร อาหารที่ซื้อ	D2 ข้อมูลสูตร อาหาร	รหัสสูตรอาหาร+ชื่อ สูตรอาหาร+ราคา+ สำหรับกี่คน+เวลาที่ ใช้+ประเภท+ คำอธิบาย+รูปภาพ+ รหัสสมาชิก
ข้อมูลวิธีทำ			D3 ข้อมูลวิธี ทำ	รหัสวิธีทำ+ คำอธิบาย+ลำดับ+ path_file+ประเภท

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลวัตถุดิบ			D4 ข้อมูลวัตถุดิบ	รหัสวัตถุดิบ+ชื่อวัตถุดิบ+จำนวน+ลำดับ
ประวัติราคาซื้อขาย			D5 ประวัติการซื้อขาย	ราคา
ข้อมูลสูตรอาหาร	รายละเอียดการแสดงสูตรอาหารที่ซื้อ	Process 7.3 แสดงสูตรอาหารที่ซื้อ	สมาชิก	รหัสสูตรอาหาร+ชื่อสูตรอาหาร+ราคา+สำหรับกี่คน+เวลาที่ใช้+ประเภท+คำอธิบาย+รูปภาพ+รหัสสมาชิก+[รหัสวัตถุดิบ+ชื่อวัตถุดิบ+จำนวน+ลำดับ]+[รหัสวิธีทำ+คำอธิบาย+ลำดับ+path_file+ประเภทของไฟล์]+ราคา
ข้อมูลสูตรอาหาร	ดึงข้อมูลสูตรอาหารที่ซื้อพร้อมวัตถุดิบและวิธีทำและแสดงราคา		D2 ข้อมูลสูตรอาหาร	รหัสสูตรอาหาร+ชื่อสูตรอาหาร+ราคา+สำหรับกี่คน+เวลาที่ใช้+ประเภท+คำอธิบาย+รูปภาพ+รหัสสมาชิก
ข้อมูลวิธีทำ			D3 ข้อมูลวิธีทำ	รหัสวิธีทำ+คำอธิบาย+ลำดับ+path_file+ประเภทของไฟล์
ข้อมูลวัตถุดิบ			D4 ข้อมูลวัตถุดิบ	รหัสวัตถุดิบ+ชื่อวัตถุดิบ+ลำดับ

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ประวัติราคาที่ซื้อ			D5 ประวัติการซื้อ	ราคา
ข้อมูลสูตรอาหาร	รายละเอียดการแสดงผลสูตรอาหารที่ซื้อ		สมาชิก	รหัสสูตรอาหาร+ชื่อสูตรอาหาร+ราคา+สำหรับกี่คน+เวลาที่ใช้+ประเภท+คำอธิบาย+รูปภาพ+รหัสสมาชิก+[รหัสวัตถุดิบ+ชื่อวัตถุดิบ+จำนวน+ลำดับ]+[รหัสวิธีทำ+คำอธิบาย+ลำดับ+path_file+ประเภทของไฟล์]+ราคา
ข้อมูลการเติมเงิน	รายละเอียดการเติมเงินเข้าระบบ	สมาชิก	Process 8.1 เติมเงิน	รหัสสมาชิก+จำนวนเงิน
จำนวนเงินของผู้ใช้	ดึงข้อมูลจำนวนเงินของผู้ใช้มาคำนวณ	Process 8.1 เติมเงิน	D1 ข้อมูลสมาชิก	จำนวนเงิน
จำนวนเงินที่เติม	บันทึกจำนวนเงินที่ดึงมาจากข้อมูลมาบวกเพิ่มกับจำนวนเงินที่เติม			จำนวนเงินที่บวกแล้ว
ประวัติการเติมเงิน	บันทึกธุรกรรมทางเงินลงฐานข้อมูล		D7 ข้อมูลการเงิน	รหัสการเงิน+รหัสสมาชิก+สถานะ+จำนวนเงิน+เวลา
ข้อมูลสถานะเติมเงิน	รายงานสถานะการเติมเงิน		สมาชิก	สถานะ+จำนวนเงิน
ข้อมูลการถอนเงิน	รายละเอียดการถอนเงินเข้าระบบ	สมาชิก	Process 8.2 ถอนเงิน	รหัสสมาชิก+จำนวนเงิน

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
จำนวนเงินของผู้ใช้	ดึงข้อมูลจำนวนเงินของผู้ใช้มาคำนวณ	Process 8.2 ถอนเงิน	D1 ข้อมูลสมาชิก	จำนวนเงิน
ตรวจสอบธุรกรรมทั้งหมด	ดึงข้อมูลการเงินมาคำนวณเพื่อตรวจสอบว่าข้อมูลการเงินนี้ไม่มีข้อผิดพลาด		D7 ข้อมูลการเงิน	ลิสต์จำนวนเงิน+สถานะ
จำนวนเงินที่ถอน	บันทึกจำนวนเงินที่ดึงมาจากข้อมูลมาลบกับจำนวนเงินที่ถอน		D1 ข้อมูลสมาชิก	จำนวนเงินที่หักไปแล้ว
ประวัติการถอนเงิน	บันทึกธุรกรรมทางเงินลงฐานข้อมูล	Process 8.2 ถอนเงิน	D7 ข้อมูลการเงิน	รหัสการเงิน+รหัสสมาชิก+สถานะ+จำนวนเงิน+เวลา
ข้อมูลสถานะการถอนเงิน	รายงานสถานะการถอนเงิน		สมาชิก	สถานะ+จำนวนเงินล่าสุด
User id	รายละเอียดแสดงยอดเงินและประวัติการใช้เงินในระบบ	สมาชิก	Process 8.3 ตรวจสอบยอดเงิน	รหัสสมาชิก
จำนวนยอดเงินคงเหลือ	ดึงจำนวนยอดเงินจากข้อมูลสมาชิก	Process 8.3 ตรวจสอบ	D1 ข้อมูลสมาชิก	จำนวนเงิน
ประวัติการใช้เงินในระบบ	ดึงข้อมูลการเติมและถอนเงินในระบบของผู้ใช้	ยอดเงิน	D7 ข้อมูลการเงิน	รหัสการเงิน+รหัสสมาชิก+สถานะ+จำนวนเงิน+เวลา
ข้อมูลยอดเงิน	แสดงรายละเอียดข้อมูลการเติม		สมาชิก	จำนวนเงิน+[รหัสการเงิน+รหัสสมาชิก]

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
	และถอนเงินในระบบ ของผู้ใช้			+สถานะ+จำนวน เงิน+เวลา]
ข้อมูลการ ติดตาม	รายละเอียดข้อมูล การติดตามสมาชิก	สมาชิก	Process 9.1 ติดตาม	รหัสสมาชิก+รหัส สมาชิกที่ติดตาม
ข้อมูลสมาชิกที่ ติดตาม	บันทึกข้อมูลการ ติดตาม	Process 9.1 ติดตาม	D8 ข้อมูลการ ติดตาม	รหัสติดตาม+รหัส สมาชิก+รหัสสมาชิก ที่ติดตาม
ข้อมูลการแจ้ง เตือนการ ติดตาม	บันทึกข้อมูลการแจ้ง เตือนการติดตาม		D11 ข้อมูล การแจ้งเตือน	รหัสการแจ้งเตือน+ รหัสสมาชิกที่ ติดตาม+สถานะ+ รายละเอียด+เวลา+ รหัสสมาชิก
ข้อมูลการ ยกเลิกติดตาม	รายละเอียดข้อมูล การยกเลิกติดตาม	สมาชิก	Process 9.2 ยกเลิกติดตาม	รหัสสมาชิก+รหัส สมาชิกที่ติดตาม
ข้อมูลสมาชิกที่ ติดตาม	ลบข้อมูลการติดตาม	Process 9.2 ยกเลิกติดตาม	D8 ข้อมูลการ ติดตาม	รหัสสมาชิกที่ติดตาม
ข้อมูลคะแนน สูตรอาหาร	รายละเอียดข้อมูล การเพิ่มคะแนน	สมาชิก	Process 10.1 เพิ่มคะแนน สูตรอาหาร	รหัสสูตรอาหาร+ รหัสสมาชิก+คะแนน
คะแนนสูตร อาหาร	บันทึกคะแนนสูตร อาหาร	Process 10.1 เพิ่ม คะแนนสูตร อาหาร	D9 ข้อมูล คะแนนสูตร อาหาร	รหัสคะแนน+ คะแนน+รหัสสูตร อาหาร+รหัส สมาชิก+คะแนน
ข้อมูลการแจ้ง เตือนคะแนน	บันทึกข้อมูลการแจ้ง เตือน		D11 ข้อมูล การแจ้งเตือน	รหัสแจ้งเตือน+รหัส สมาชิกคะแนน+ สถานะ+ รายละเอียด+เวลา+ รหัสสูตรอาหาร+ รหัสสมาชิก

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลคะแนน สูตรอาหาร	รายละเอียดข้อมูล การแก้ไขคะแนน	สมาชิก	Process 10.2 แก้ไขคะแนน สูตรอาหาร	รหัสสูตรอาหาร+ รหัสสมาชิก+คะแนน
คะแนนสูตร อาหาร	ดึงข้อมูลคะแนนสูตร อาหาร	Process 10.2 แก้ไข คะแนนสูตร อาหาร	D9 ข้อมูล คะแนนสูตร อาหาร	รหัสคะแนน+ คะแนน+รหัสสูตร อาหาร+รหัส สมาชิก+คะแนนเก่า
คะแนนสูตร อาหาร	บันทึกข้อมูลคะแนน สูตรอาหาร			รหัสคะแนน+ คะแนน+รหัสสูตร อาหาร+รหัส สมาชิก+คะแนนใหม่
ข้อมูลการแจ้ง เตือนคะแนน	บันทึกข้อมูลการแจ้ง เตือน	สมาชิก	D11 ข้อมูล การแจ้งเตือน	รหัสแจ้งเตือน+รหัส สมาชิกให้คะแนน+ สถานะ+ รายละเอียด+เวลา+ รหัสสูตรอาหาร+ รหัสสมาชิก
ข้อมูลการคอม เมนต์สูตร อาหาร	รายละเอียดการคอม เมนต์สูตรอาหาร			Process 11 เพิ่มคอมเมนต์ สูตรอาหาร
ข้อมูลการคอม เมนต์สูตร อาหาร	บันทึกรายละเอียด การคอมเมนต์สูตร อาหาร	Process 11 เพิ่มคอมเมนต์ สูตรอาหาร	D10 ข้อมูล การคอมเมนต์ สูตรอาหาร	รหัสคอมเมนต์+ รายละเอียด+รหัส สมาชิก+รหัสสูตร อาหาร
ข้อมูลการแจ้ง เตือนคอมเมนต์	บันทึกข้อมูลการแจ้ง เตือน			D11 ข้อมูล การแจ้งเตือน

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลการ รายงานผู้ใช้	รายละเอียดการ รายงานผู้ใช้	สมาชิก	Process 12 รายงานผู้ใช้	รหัสสมาชิก เป้าหมาย+รหัส สมาชิก+เวลา+ ประเภทการ รายงาน+ชื่อเรื่อง+ รายละเอียด+รูปภาพ
ข้อมูลการ รายงาน	บันทึกข้อมูลการ รายงาน	Process 12 รายงานผู้ใช้	D12 ข้อมูล การรายงาน ผู้ใช้	รหัสรายงาน+รหัส สมาชิกเป้าหมาย+ รหัสสมาชิก+เวลา+ ประเภทการ รายงาน+ชื่อเรื่อง+ รายละเอียด+รูปภาพ
ข้อมูลการแจ้ง เดือนการ รายงาน	บันทึกข้อมูลการแจ้ง เดือน		D11 ข้อมูล การแจ้งเดือน	รหัสแจ้งเดือน+รหัส สมาชิก+สถานะ+ รายละเอียด+เวลา+ รหัสสูตรอาหาร+ รหัสสมาชิก
ข้อมูลการแจ้ง เดือน	ดึงข้อมูลการแจ้ง เดือน	Process 13 แจ้งเดือน	D11 ข้อมูล การแจ้งเดือน สมาชิก ผู้ดูแลระบบ	รหัสการแจ้งเดือน+ รหัสสมาชิก+ สถานะ+ รายละเอียด+เวลา+ รหัสสูตรอาหาร+ รหัสสมาชิก



### 3.5 Data Store Description and Data Structure of Data Store

Data store คือการนำข้อมูลเข้าไปจัดการในฐานข้อมูล โดยมีการแยกออกเป็นแฟ้มข้อมูล

ตารางที่ 3.3 Data Store Description and Data Structure of data store

ID	Data store	Structure
D1	ข้อมูลสมาชิก	รหัสสมาชิก+อีเมล+รหัสผ่าน+ชื่อนามสกุล+ชื่อนามแฝง+รูปโปรไฟล์+สถานะผู้ใช้งาน+สถานะการใช้งาน
D2	ข้อมูลสูตรอาหาร	รหัสสูตรอาหาร+ชื่อสูตรอาหาร+สำหรับกี่คน+เวลาที่ใช้+ประเภท+รูปภาพ+เวลาโพสต์+ราคา
D3	ข้อมูลวิธีทำ	รหัสวิธีทำ+รหัสสูตรอาหาร+คำอธิบาย+ลำดับ+ตำแหน่งไฟล์+ประเภทไฟล์
D4	ข้อมูลวัตถุดิบ	รหัสวัตถุดิบ+รหัสสูตรอาหาร+ชื่อวัตถุดิบ+จำนวนที่ใช้+ลำดับ
D5	ประวัติการซื้อ	รหัสการซื้อ+รหัสสูตรอาหาร+รหัสสมาชิก+เวลา+ราคา
D6	ประวัติการขาย	รหัสการขาย+รหัสสมาชิก+รหัสพาหนะ+รหัสสูตรอาหาร+เวลา+ราคา
D7	ข้อมูลการเงิน	รหัสการเงิน+รหัสสมาชิก+สถานะ+จำนวนเงิน+เวลา
D8	ข้อมูลการติดตาม	รหัสการติดตาม+รหัสสมาชิก+รหัสสมาชิกที่กำลังติดตาม
D9	ข้อมูลคะแนนสูตรอาหาร	รหัสคะแนน+รหัสสมาชิก+รหัสสูตรอาหาร+คะแนน
D10	ข้อมูลคอมเมนต์สูตรอาหาร	รหัสคอมเมนต์+รหัสสูตรอาหาร+รหัสสมาชิก+รายละเอียดคอมเมนต์+เวลาคอมเมนต์

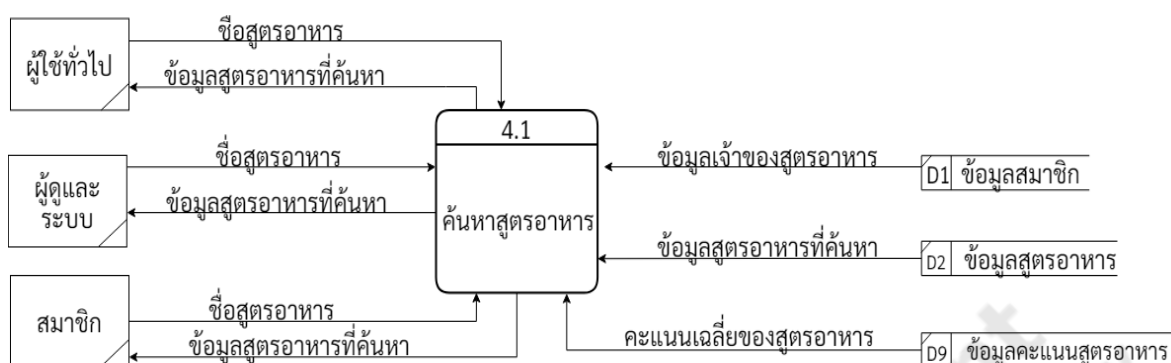
ตารางที่ 3.3 Data Store Description and Data Structure of data store (ต่อ)

ID	Data store	Structure
D11	ข้อมูลการแจ้งเตือน	รหัสการแจ้งเตือน+รหัสสมาชิก+สถานะ+คำอธิบาย+สถานะกิจกรรม+เวลา+การพบเห็น+รหัสสูตรอาหาร+รหัสสมาชิกจาก
D12	ข้อมูลการรายงานผู้ใช้	รหัสการรายงาน+รหัสสมาชิกเป้าหมาย+รหัสสมาชิก+เวลา+ประเภทการรายงาน+รหัสสูตรอาหาร+หัวข้อ+คำอธิบาย+รูปภาพ

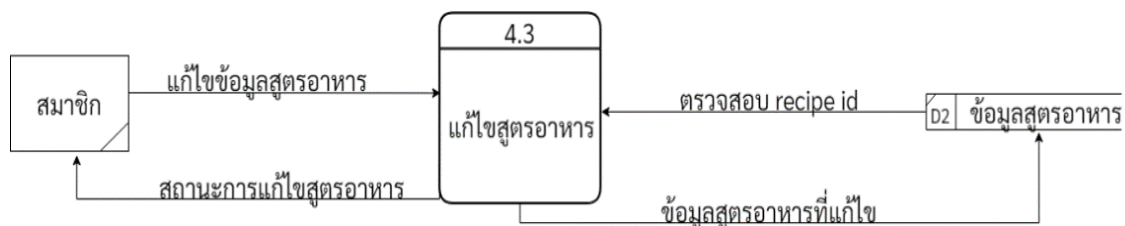
### 3.6 คำอธิบายการประมวลผล (Process Description) ระบบกลาง



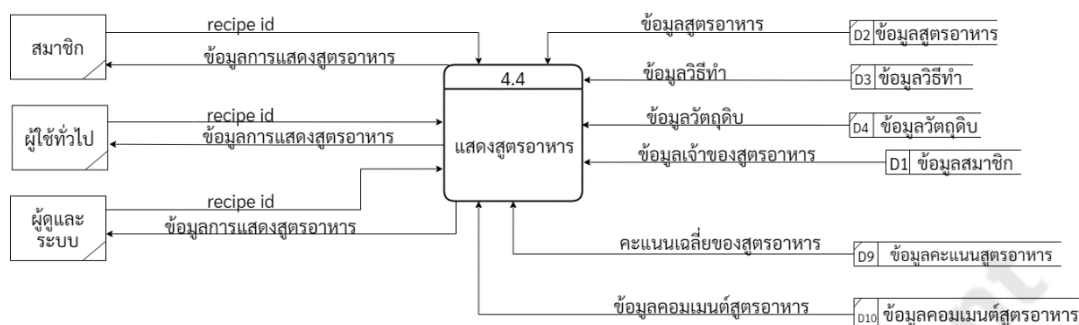
ID	3.3
NAME	ยกเลิกการเป็นสมาชิก
DESCRIPTION	การยกเลิกการเป็นสมาชิก
INPUT DATA FLOW	User id
OUTPUT DATA FLOW	-ข้อมูลสมาชิกที่ต้องการยกเลิก -สถานะการยกเลิก
PROCESS DESCRIPTION	เริ่มต้น 1.รับข้อมูล user id จากผู้ใช้ 2.เปลี่ยนแปลงสถานะการเข้าใช้งาน 3.รายงานสถานะการยกเลิก จบการทำงาน



ID	4.1
NAME	ค้นหาสูตรอาหาร
DESCRIPTION	การค้นหาสูตรอาหาร
INPUT DATA FLOW	-ข้อมูลสูตรอาหาร -ข้อมูลเจ้าของสูตรอาหาร -ข้อมูลสูตรอาหารที่ค้นหา
OUTPUT DATA FLOW	ข้อมูลสูตรอาหารที่ค้นหา
PROCESS DESCRIPTION	เริ่มต้น 1.รับข้อมูลชื่อสูตรอาหาร จาก key word 2.ตรวจสอบว่ามีข้อมูลจาก key word นี้ในระบบหรือไม่ ถ้า (มีข้อมูล) 2.1 ส่งข้อมูลของสูตรอาหาร ถ้า (ไม่มีข้อมูล) 2.2 ส่งข้อความเตือน ไม่พบข้อมูลสูตรอาหาร จบการทำงาน



ID	4.3
NAME	แก้ไขสูตรอาหาร
DESCRIPTION	การแก้ไขสูตรอาหาร
INPUT DATA FLOW	-ข้อมูลการแก้ไขสูตรอาหาร
FLOW	-ตรวจสอบ recipe id
OUTPUT DATA FLOW	-ข้อมูลสูตรอาหารที่แก้ไข
	-สถานะการแก้ไขสูตรอาหาร
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1.รับข้อมูลการแก้ไขสูตรอาหารจากผู้ใช้</li> <li>2.ตรวจสอบรหัสสูตรอาหาร และดึงข้อมูลมาเปรียบเทียบ ถ้า (มีข้อมูล)             <ol style="list-style-type: none"> <li>2.1 นำข้อมูลสูตรอาหารใหม่ไปอัปเดต</li> </ol> </li> <li>3.รายงานสถานะการแก้ไขสูตรอาหาร</li> </ol> <p>จบการทำงาน</p>



ID	4.4
NAME	แสดงสูตรอาหาร
DESCRIPTION	การแสดงสูตรอาหาร
INPUT DATA FLOW	-Recipe_ID -ข้อมูลสูตรอาหาร, ข้อมูลวิธีทำ, ข้อมูลวัตถุดิบ, ข้อมูลเจ้าของสูตรอาหาร -คะแนนเฉลี่ยของสูตรอาหาร, ข้อมูลคอมเมนต์สูตรอาหาร
OUTPUT DATA FLOW	ข้อมูลการแสดงสูตรอาหาร
PROCESS DESCRIPTION	เริ่มต้น 1.รับข้อมูลรหัสสูตรอาหาร 2.ตรวจสอบข้อมูลจาก รหัสสูตรอาหาร ถ้า (มีข้อมูลในระบบ) 2.1 ส่ง ข้อมูลการแสดงผลสูตรอาหาร ถ้า (ไม่มีข้อมูลในระบบ) 2.2 ส่ง ข้อความแสดง “ไม่มีสูตรอาหารนี้” จบการทำงาน



ID	7.1
NAME	ค้นหาสูตรอาหารที่ซื้อ
DESCRIPTION	การค้นหาสูตรอาหารที่ซื้อไว้แล้ว
INPUT DATA FLOW	-ชื่อสูตรอาหาร -ข้อมูลสูตรอาหาร -ประวัติการซื้อ
OUTPUT DATA FLOW	ข้อมูลสูตรอาหารที่ค้นหา
PROCESS DESCRIPTION	เริ่มต้น 1.รับข้อมูลชื่อสูตรอาหาร 2.ค้นหาข้อมูลประวัติการซื้อจากชื่อสูตรอาหาร ถ้า (มีข้อมูล) 2.1.ส่ง ข้อมูลสูตรอาหารที่ค้นหา จบการทำงาน



ID	7.2
NAME	เพิ่มสูตรอาหารที่ซื้อ
DESCRIPTION	การเพิ่มสูตรอาหารที่ซื้อ
INPUT DATA FLOW	ข้อมูลสูตรอาหารที่ซื้อ
OUTPUT DATA FLOW	-ข้อมูลประวัติการซื้อ -ข้อมูลประวัติการขาย
PROCESS DESCRIPTION	เริ่มต้น 1.รับข้อมูลสูตรอาหารที่ซื้อ 2.บันทึกข้อมูลประวัติการซื้อลงฐานข้อมูล 3.บันทึกข้อมูลประวัติการขายลงฐานข้อมูล จบการทำงาน

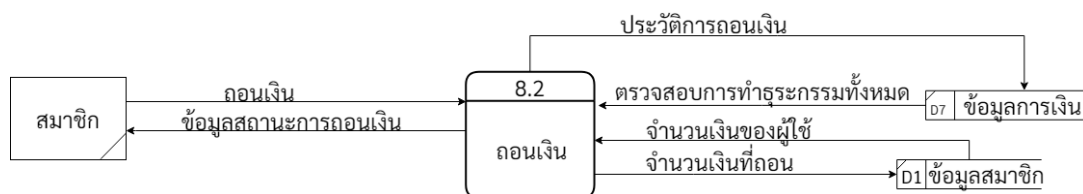


ID	8.1
NAME	เติมเงิน
DESCRIPTION	การเติมเงินเข้าระบบ
INPUT DATA FLOW	-ข้อมูลการเติมเงิน -จำนวนเงินขอผู้ใช้
OUTPUT DATA FLOW	-ประวัติการเติมเงิน -จำนวนเงินที่เติม -ข้อมูลสถานะการเติมเงิน
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1.รับข้อมูลการเติมเงิน</li> <li>2.ตรวจสอบความว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่ ถ้า (ครบถ้วน) <ol style="list-style-type: none"> <li>2.1 บันทึกข้อมูลลงฐานข้อมูล</li> <li>2.2 แสดงข้อความแจ้งเตือนการเติมเงิน</li> </ol> </li> <li>ถ้า (ไม่ครบถ้วน) <ol style="list-style-type: none"> <li>2.3 แสดงข้อความแจ้งเตือนและให้ผู้ใช้ทำการกรอกข้อมูลใหม่อีกครั้ง</li> </ol> </li> <li>3.แสดงยอดเงิน</li> </ol> <p>จบการทำงาน</p>

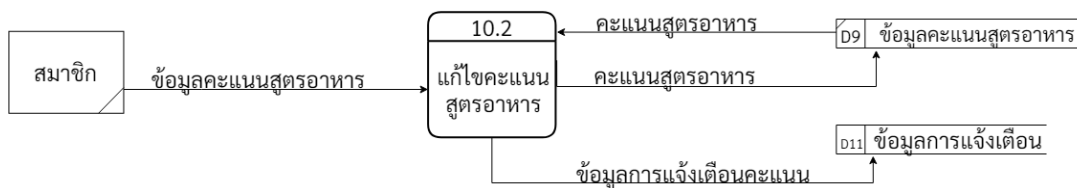




ID	8.3
NAME	ตรวจสอบยอดเงิน
DESCRIPTION	การตรวจสอบยอดเงินในระบบ
INPUT DATA FLOW	- ข้อมูลยอดเงิน - ประวัติการใช้เงินในระบบ
OUTPUT DATA FLOW	- ข้อมูลยอดเงิน
PROCESS DESCRIPTION	เริ่มต้น 1.รับข้อมูลตรวจสอบยอดเงิน 2.แสดงยอดเงินและประวัติการใช้เงิน จบการทำงาน



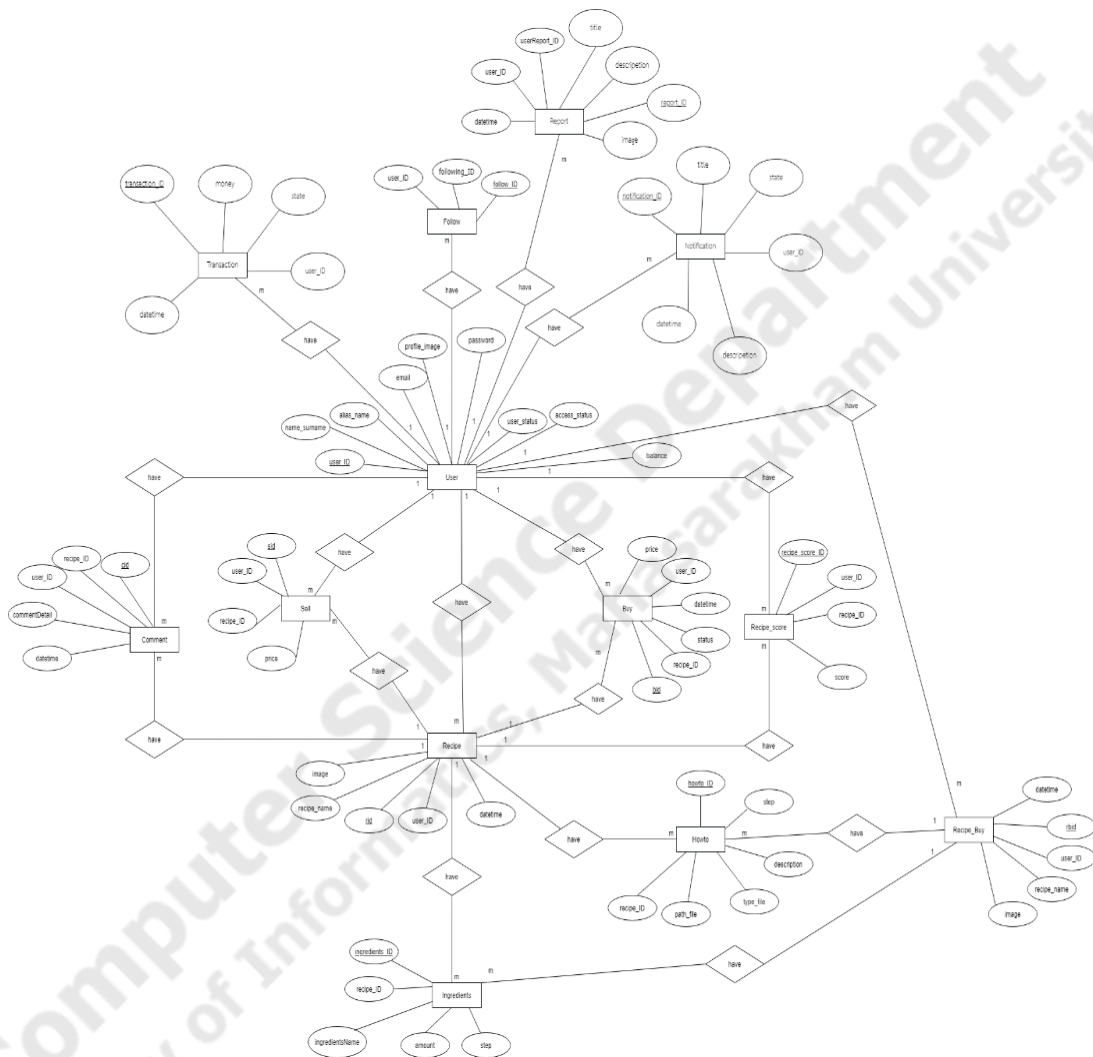
ID	8.2
NAME	ถอนเงิน
DESCRIPTION	การถอนเงินออกจากระบบ
INPUT DATA	- ข้อมูลการถอนเงิน
FLOW	- ข้อมูลแจ้งเตือนการถอนเงิน
OUTPUT DATA	- ประวัติการถอนเงิน
FLOW	- ข้อมูลแจ้งเตือนการถอนเงิน
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1.รับข้อมูลที่ต้องการถอน</li> <li>2.ตรวจสอบข้อมูลที่รับมาว่าถูกต้องครบถ้วนหรือไม่             <ul style="list-style-type: none"> <li>ถ้า (ข้อมูลถูกต้องและครบถ้วน)                 <ol style="list-style-type: none"> <li>2.1 ทำการถอนเงิน</li> <li>2.2 แสดงข้อความแจ้งเตือนการเติมเงิน</li> </ol> </li> <li>ถ้า (ข้อมูลไม่ถูกต้องและไม่ครบถ้วน)                 <ol style="list-style-type: none"> <li>2.3 แสดงข้อความเตือนถึงข้อผิดพลาด</li> </ol> </li> </ul> </li> </ol> <p>จบการทำงาน</p>



ID	10.2
NAME	แก้ไขคะแนนสูตรอาหาร
DESCRIPTION	การแก้ไขคะแนนสูตรอาหาร
INPUT DATA FLOW	- คะแนนสูตรอาหาร - ข้อมูลสูตรอาหาร - ข้อมูลสมาชิก
OUTPUT DATA FLOW	- คะแนนสูตรอาหาร - ข้อมูลการแจ้งเตือนคะแนน
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1.รับคะแนนสินค้าที่ต้องการแก้ไข</li> <li>2.ตรวจสอบความครบถ้วนในการแก้ไขคะแนนสูตรอาหาร             <ul style="list-style-type: none"> <li>ถ้า (ครบถ้วน)                 <ul style="list-style-type: none"> <li>2.1 บันทึกข้อมูลลงฐานข้อมูลคะแนนสูตรอาหาร</li> <li>2.2 บันทึกข้อมูลลงฐานข้อมูลการแจ้งเตือน</li> </ul> </li> <li>ถ้า (ไม่ครบถ้วน)                 <ul style="list-style-type: none"> <li>2.2 แสดงข้อความแจ้งเตือนและให้ผู้ใช้ทำการกรอกข้อมูลใหม่อีกครั้ง</li> </ul> </li> </ul> </li> <li>3.แสดงการแจ้งเตือนเมื่อทำการแก้ไขหรือกรอกข้อมูลครบถ้วน”</li> </ol> <p>จบการทำงาน</p>

### 3.7 แผนภาพ Entity Relationship Diagram (ER-Diagram)

แบบจำลองที่ใช้อธิบายโครงสร้างของฐานข้อมูลที่ใช้ในแอปพลิเคชันซึ่งเขียนออกมาในลักษณะของรูปภาพ การอธิบายโครงสร้างละความสัมพันธ์ของข้อมูล



ภาพประกอบที่ 3.5 แผนภาพ Entity Relationship Diagram (ER-Diagram)

### 3.8 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ตารางสมาชิก(User)

Column	Type	Description	Example	Constraint
user_ID	int(11) identity(1,1)	รหัสสมาชิก (ไม่สามารถแก้ไข ได้)	99	PK
name_surname	varchar(50)	ชื่อ-นามสกุล ผู้ใช้	ชาญชัย ดิษฐพันธุ์	-
alias_name	varchar(30)	ชื่อนามแฝงแทนตัว ผู้ใช้ (กรอกหรือไม่ กรอกก็ได้)	เซฟปก	-
email	varchar(100)	อีเมล (ใช้เป็น Username เข้าสู่ระบบ)	gucciBelt321@gm ail.com	UNIQUE
facebookID	Varchar(255)	ไอดีสำหรับ facebook	11767575228124 65	-
password	varchar(255)	รหัสผ่าน (Password)	asd- 123_lnwZa007	-
user_status	int(1)	สถานะผู้ใช้งาน (0=ผู้ดูแล, 1=ผู้ใช้ ที่เป็นสมาชิก)	0	0,1
access_status	int(1)	สถานะการใช้งาน (0=ยกเลิกสิทธิ์การ ใช้งาน, 1=มีสิทธิ์ ใช้งาน)	1	0,1

ตารางที่ 3.4 ตารางสมาชิก(User) (ต่อ)

Column	Type	Description	Example	Constraint
balance	double(10)	ยอดเงินคงเหลือ (ยอดเงินเริ่มต้น 0 บาท)	199.64	-
Profile_image	text	รูปภาพโปรไฟล์ แทนตัวผู้ใช้ (เพิ่ม หรือไม่เพิ่มก็ได้)	http://apifood.com sciproject.com/upl oadProfile\2021-0- 12- 25T1537047857.pn g	-
wallpaper	text	Wallpaper สำหรับพื้นหลัง ระบบ	assets/wallpapers/ default.jpg	-
Token_noti	text	Token สำหรับ ฟังก์ชันแจ้งเตือน	NULL	-

ตารางที่ 3.5 ตารางสูตรอาหาร

Column	Type	Description	Example	Constraint
rid	int(11) identity(1,1)	รหัสสูตรอาหาร	99	PK
user_ID	int(11)	รหัสสมาชิก	99	References for User(user_ID)
recipe_name	varchar(60)	ชื่อสูตรอาหาร	กะเพราหมูกรอบ ไฟแดง	-
suitable_for	text	เหมาะสำหรับกี่ คน	2 คน	-

ตารางที่ 3.5 ตารางสูตรอาหาร (ต่อ)

Column	Type	Description	Example	Constraint
Take_time	text	ใช้เวลาในการทำ	ภายใน 15 นาที	-
Food_category	text	ประเภทอาหาร	เมนูทอด	-
description	text	คำอธิบาย	-	-
image	text	รูปภาพ	http://apifood.com/sciproject.com/uploadProfile\25T1537047857.png	-
Datetime	datetime	วันที่และเวลา	2021-02-06 18:45:02	-
price	double	ราคาสูตรอาหาร	0	-

ตารางที่ 3.6 ตารางติดตาม(Follow)

Column	Type	Description	Example	Constraint
follow_ID	int(11) identity(1,1)	รหัสติดตาม	99	PK
user_ID	int(11)	รหัสสมาชิก	4	References for User(user_ID)
following_ID	int(11)	รหัสสมาชิกที่ติดตาม	5	References for User(user_ID)

ตารางที่ 3.7 ตารางคอมเมนต์(Comment)

Column	Type	Description	Example	Constraint
cid	int(11) identity(1,1)	รหัสคอมเมนต์	99	PK
recipe_ID	int(11)	รหัสสูตรอาหาร	99	References for Recipe(recipe_ID)
user_ID	int(11)	รหัสสมาชิก	99	References for User(user_ID)
commentDetail	text	แสดงความคิดเห็น	ทำตามแล้วอร่อยมาก ครับ	
datetime	datetime	วันที่และเวลา	2021-02-06 18:45:02	

ตารางที่ 3.8 ตารางขาย(Sell)

Column	Type	Description	Example	Constraint
sid	int(11) identity(1,1)	รหัสขาย	99	PK
user_ID	int(11)	รหัสสมาชิก	99	References for User(user_ID)
Recipe_ID	int(11)	รหัสสูตรอาหาร	99	References for Recipe(recipe_ID)
datetime	timestamp	เวลา	2021-10-01 15:48:51	
price	double	ราคา	50.00	-



ตารางที่ 3.9 ตารางซื้อ(Buy)

Column	Type	Description	Example	Constraint
bid	int(11) identity(1,1)	รหัสซื้อ	99	PK
recipe_ID	int(11)	รหัสสูตรอาหาร (แก้ไขไม่ได้)	99	References for Recipe(recipe_ID)
Datetime	datetime	วันที่และเวลา	2021-02-06 18:45:02	-
user_ID	int(11)	รหัสสมาชิก	99	References for User(user_ID)
price	double	ราคา	150.00	-

ตารางที่ 3.10 ตารางคะแนนสูตรอาหาร(Recipe score)

Column	Type	Description	Example	Constraint
recipe_score_ID	int(11) identity(1,1)	รหัสคะแนน	99	PK
user_ID	int(11)	รหัสสมาชิก	99	References for User(user_ID)
recipe_ID	int(11)	รหัสสูตรอาหาร	99	References for Recipe(recipe_ID)

ตารางที่ 3.10 ตารางคะแนนสูตรอาหาร(Recipe score) (ต่อ)

Column	Type	Description	Example	Constraint
score	int(1)	คะแนน	5	1,2,3,4,5

ตารางที่ 3.11 ตารางวิธีทำ

Column	Type	Description	Example	Constraint
Howto_ID	int(11) identity(1,1)	รหัสวิธีทำ	99	PK
recipe_ID	int(11)	รหัสสูตรอาหาร	99	References for Recipe(recipe_ID)
description	TEXT	อธิบายวิธีทำ	1.เตรียมหมู มาหมักไว้	-
step	int(11)	ลำดับ	1	-
Path_file	text	รูปภาพหรือวิดีโอ	http://apifood.com/sciproject.com/uploads\2021-02-05T161712392Z-7-9.jpg	-
Type_file	Text	ประเภทของไฟล์	Image	-

ตารางที่ 3.12 ตารางส่วนผสม

Column	Type	Description	Example	Constraint
Ingredients_ID	int(11) identity(1,1)	รหัสส่วนผสม	99	PK

ตารางที่ 3.12 ตารางส่วนผสม (ต่อ)

Column	Type	Description	Example	Constraint
recipe_ID	int(11)	รหัสสูตรอาหาร	99	References for Recipe(recipe_ID)
Ingredients Name	text	ชื่อวัตถุดิบ	ไข่ไก่	-
amount	text	จำนวนของวัตถุดิบ	2 ฟอง	-
Step	Int(11)	ลำดับ	1	-

ตารางที่ 3.13 ตารางการเงิน(Transaction)

Column	Type	Description	Example	Constraint
transaction_ID	int(11) identity(1,1)	รหัสการเงิน	99	PK
user_ID	int(11)	รหัสสมาชิก	99	References for User(user_ID)
state	varchar(20)	สถานะ	เติมเงิน	
money	double	ยอดเงิน	100	-
datetime	datetime	วันที่และเวลา	2021-02-06 18:45:02	-

ตารางที่ 3.14 ตารางการแจ้งเตือน(Notifications)

Column	Type	Description	Example	Constraint
nid	int(11) identity(1,1)	รหัสการแจ้งเตือน	99	PK
My_ID	int(11)	รหัสสมาชิก	99	References for User(user_ID)
state	varchar(20)	สถานะ	ติดตาม	
status	varchar(30)	ชื่อเรื่อง	follow	-
datetime	datetime	วันที่และเวลา	2021-02-06 18:45:02	-
description	text	อธิบาย	ได้ติดตามคุณ	-
Visited	Tinyint(1)	การถูกมองเห็น	1	-
Recipe_ID	Int(11)	รหัสสูตรอาหาร	NULL	References for Recipe(recipe_ID)
From_userid	Int(11)	รหัสสมาชิกที่ดำเนินการ	98	References for User(user_ID)

ตารางที่ 3.15 ตารางการรายงาน(Report)

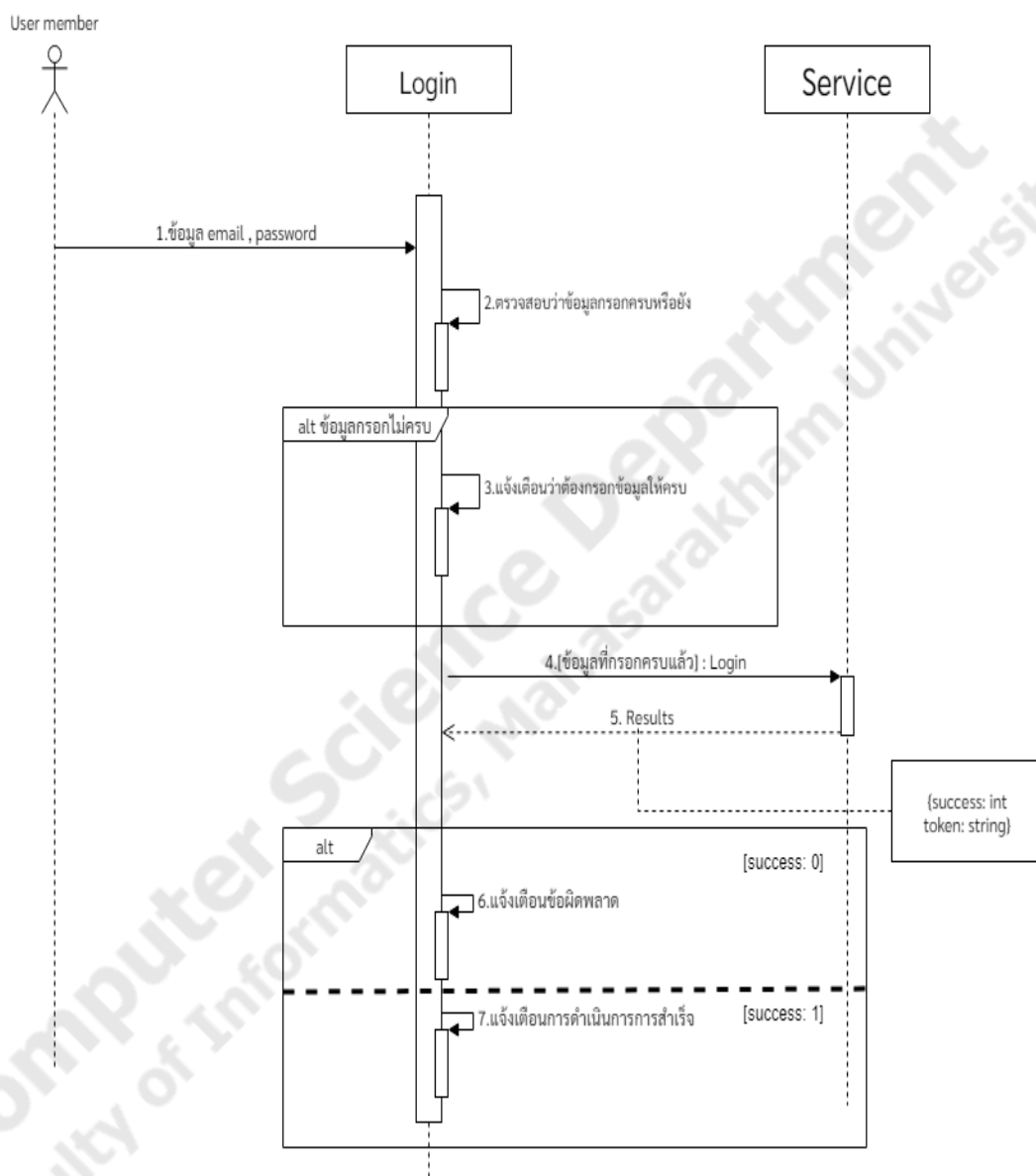
Column	Type	Description	Example	Constraint
report_ID	int(11) identity(1,1)	รหัสการรายงาน	99	PK

ตารางที่ 3.15 ตารางการรายงาน(Report) (ต่อ)

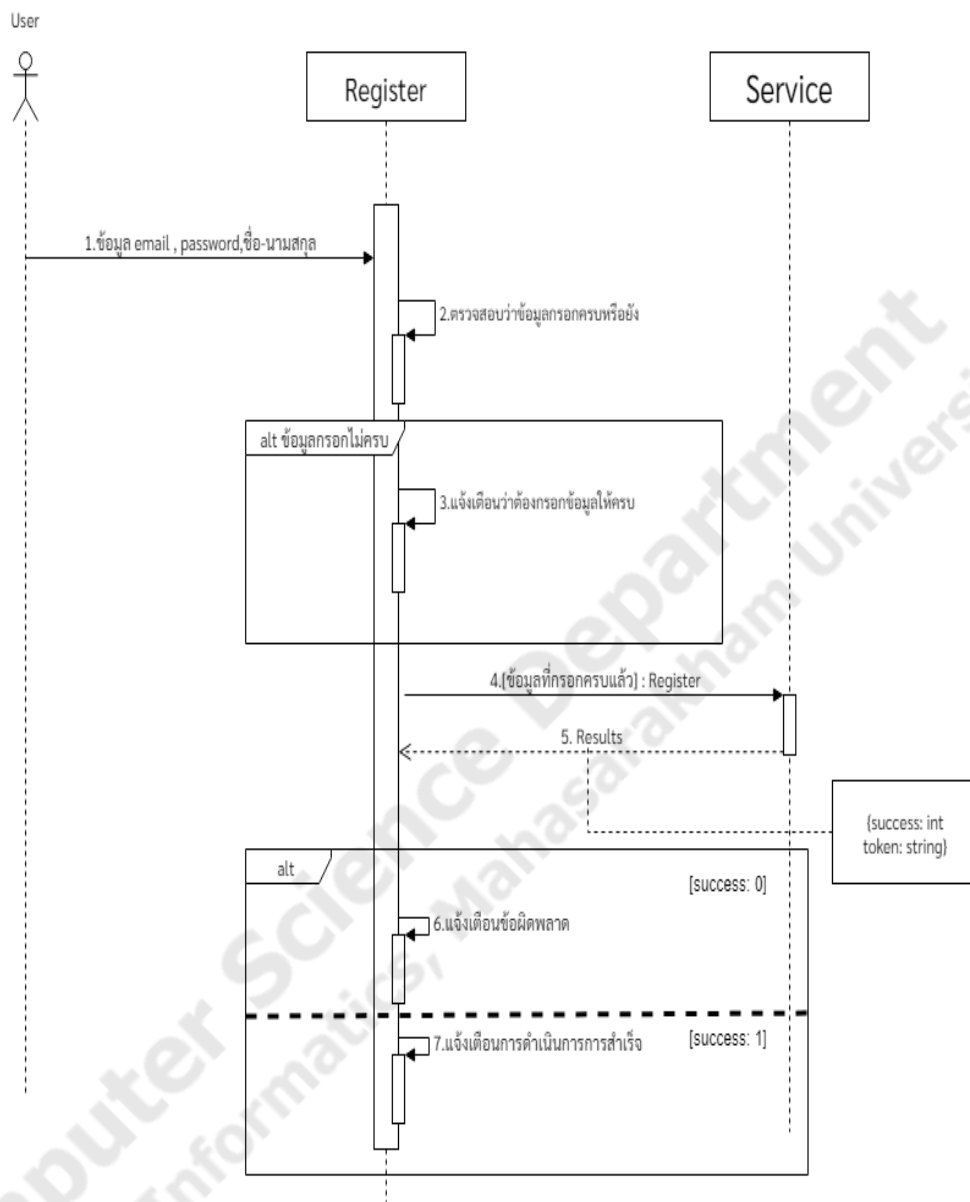
Column	Type	Description	Example	Constraint
report_ID	int(11) identity(1,1)	รหัสการรายงาน	99	PK
userTarget_ID	int(11)	รหัสสมาชิกที่ถูก รายงาน	99	References for User(user_ID )
userReport_ID	int(11)	รหัสสมาชิกที่ รายงาน	99	References forUser(user _ID)
type_report	text	ประเภทการ รายงาน	food	-
Recipe_ID	Int(11)	รหัสสูตรอาหาร	1	References for Recipe(recip e_ID)
title	varchar(30)	ชื่อเรื่อง	การรายงาน	-
datetime	datetime	วันที่และเวลา	2021-02-06 18:45:02	-
description	text	อธิบาย	โพสต์ไม่เหมาะสม	-
image	text	รูปภาพ	<a href="http://apifood.com/sciproject.com/uploads\7-9.jpg">http://apifood.com/sciproject.com/uploads\7-9.jpg</a>	-

### 3.9 การพัฒนาระบบ

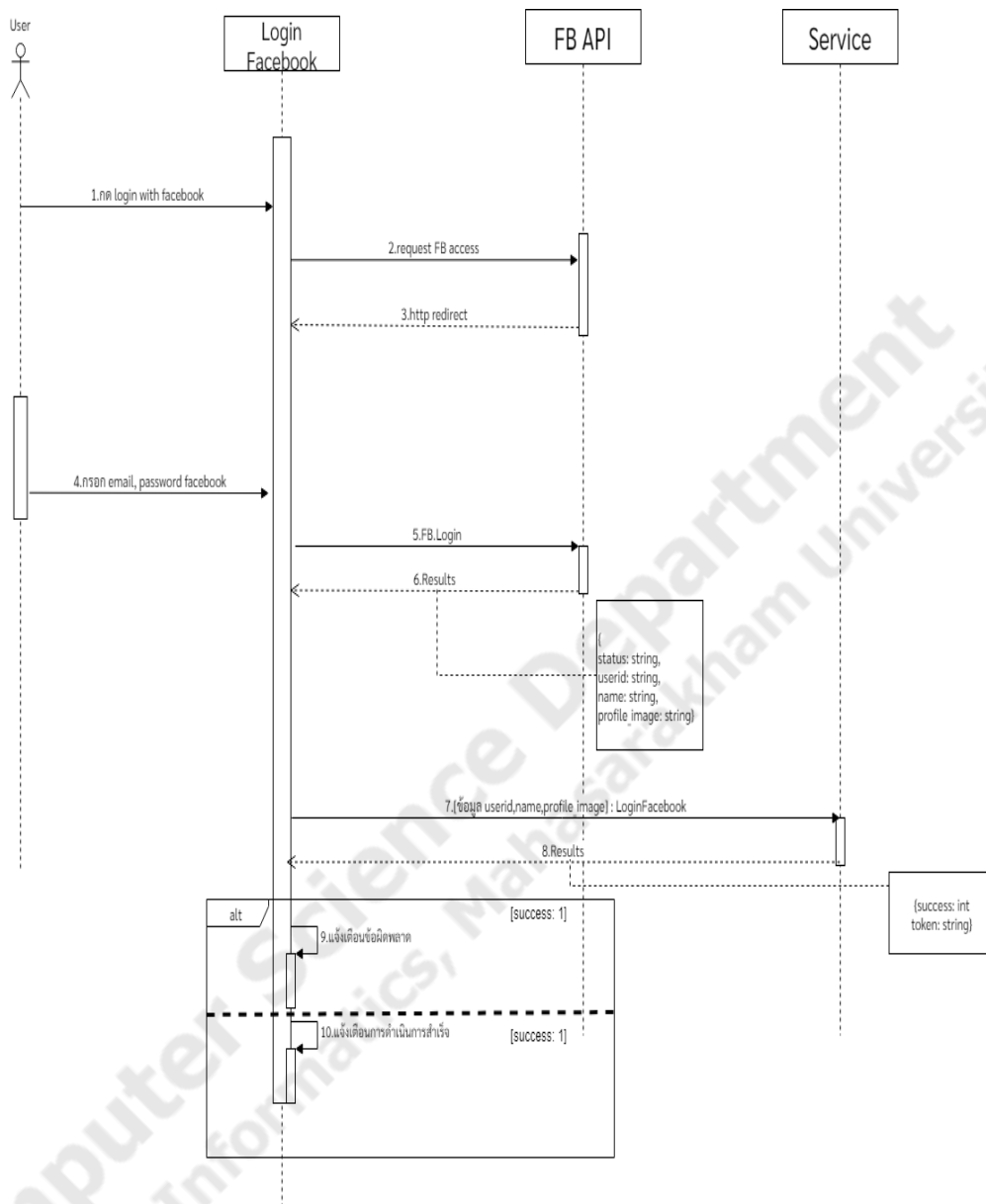
#### 3.9.1 ภาพรวมการทำงานฟังก์ชันการใช้งาน (Sequence Diagram)



ภาพประกอบที่ 3.6 Sequence Diagram การ Login เข้าสู่ระบบ

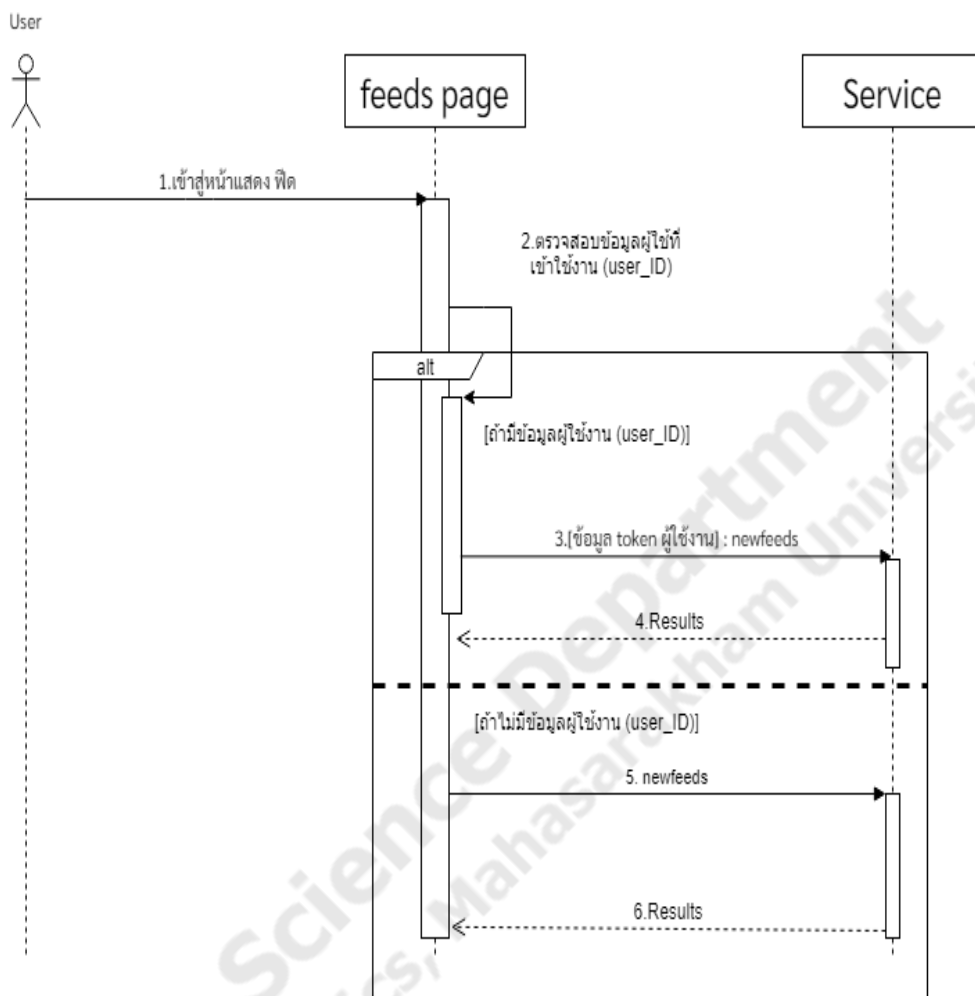


ภาพประกอบที่ 3.7 Sequence Diagram การ Register

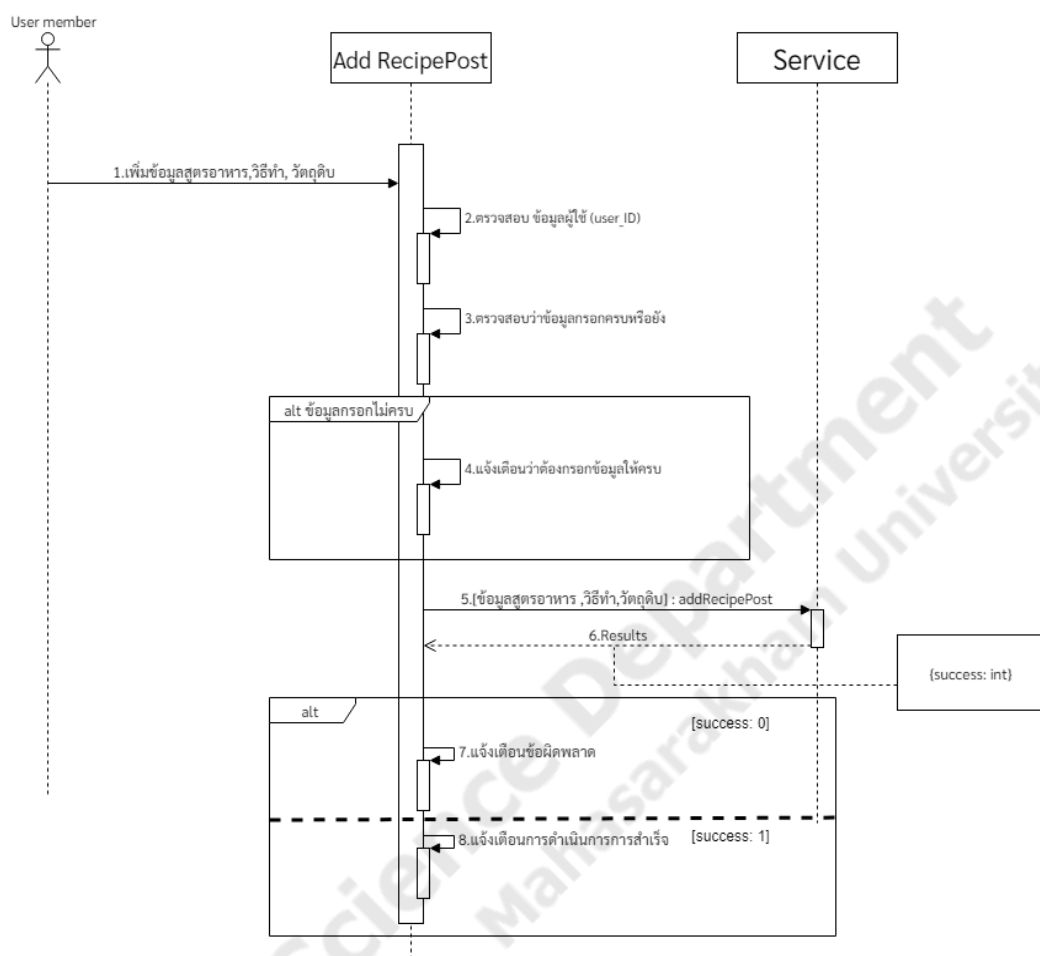


ภาพประกอบที่ 3.8 Sequence Diagram การ Login ด้วย Facebook

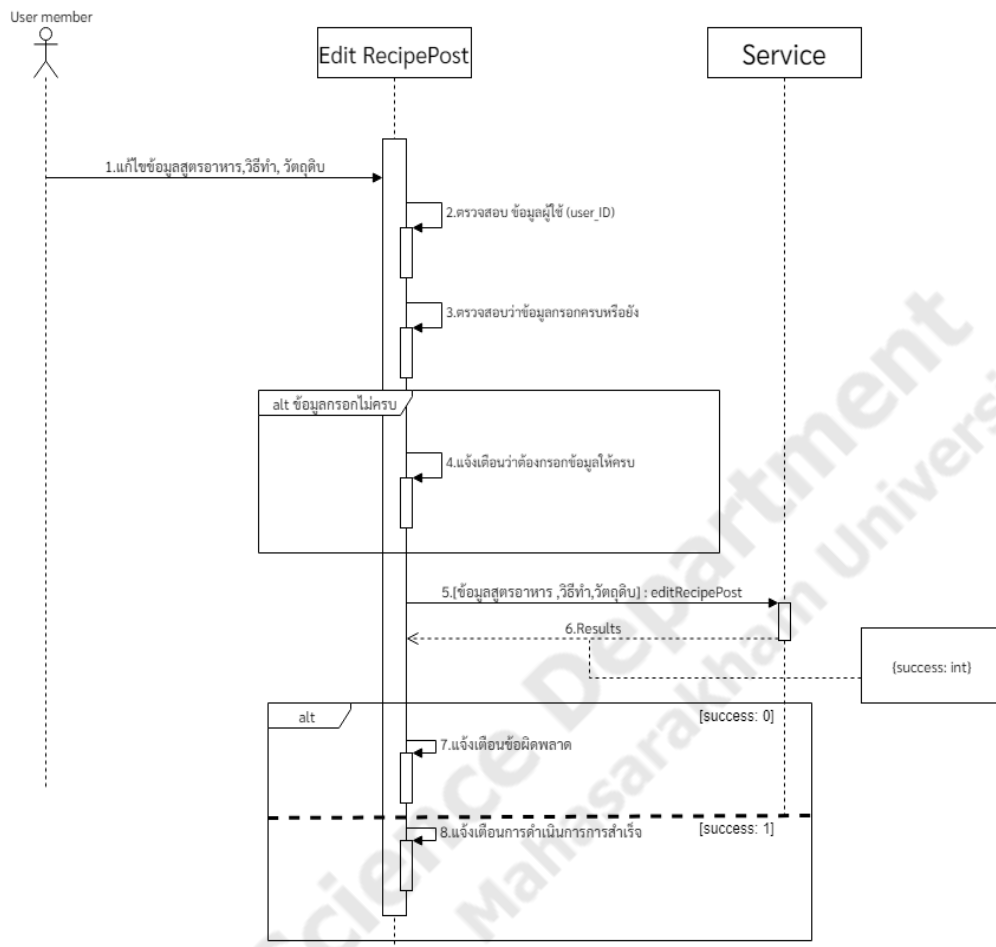




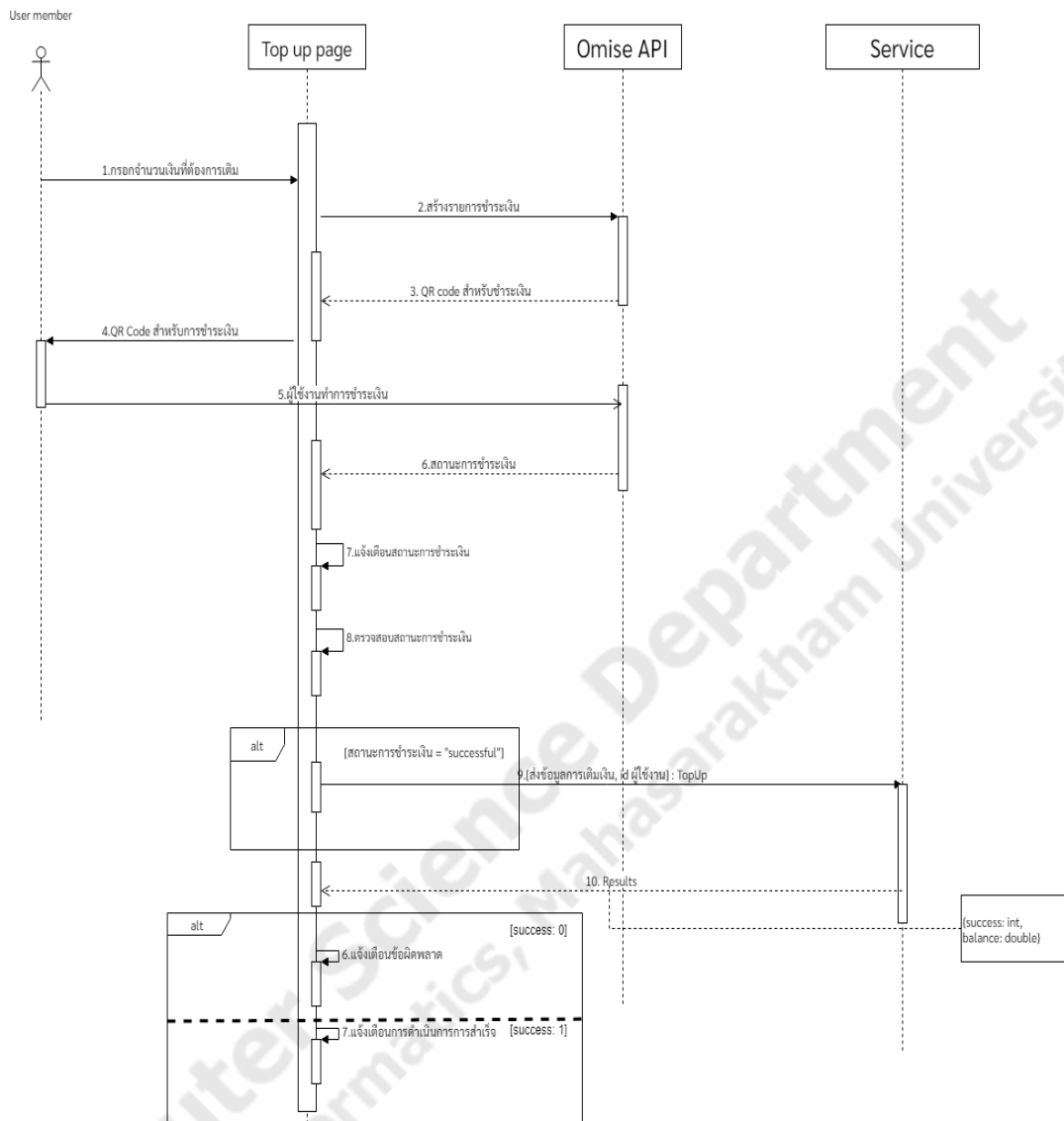
ภาพประกอบที่ 3.9 Sequence Diagram หน้าฟีด



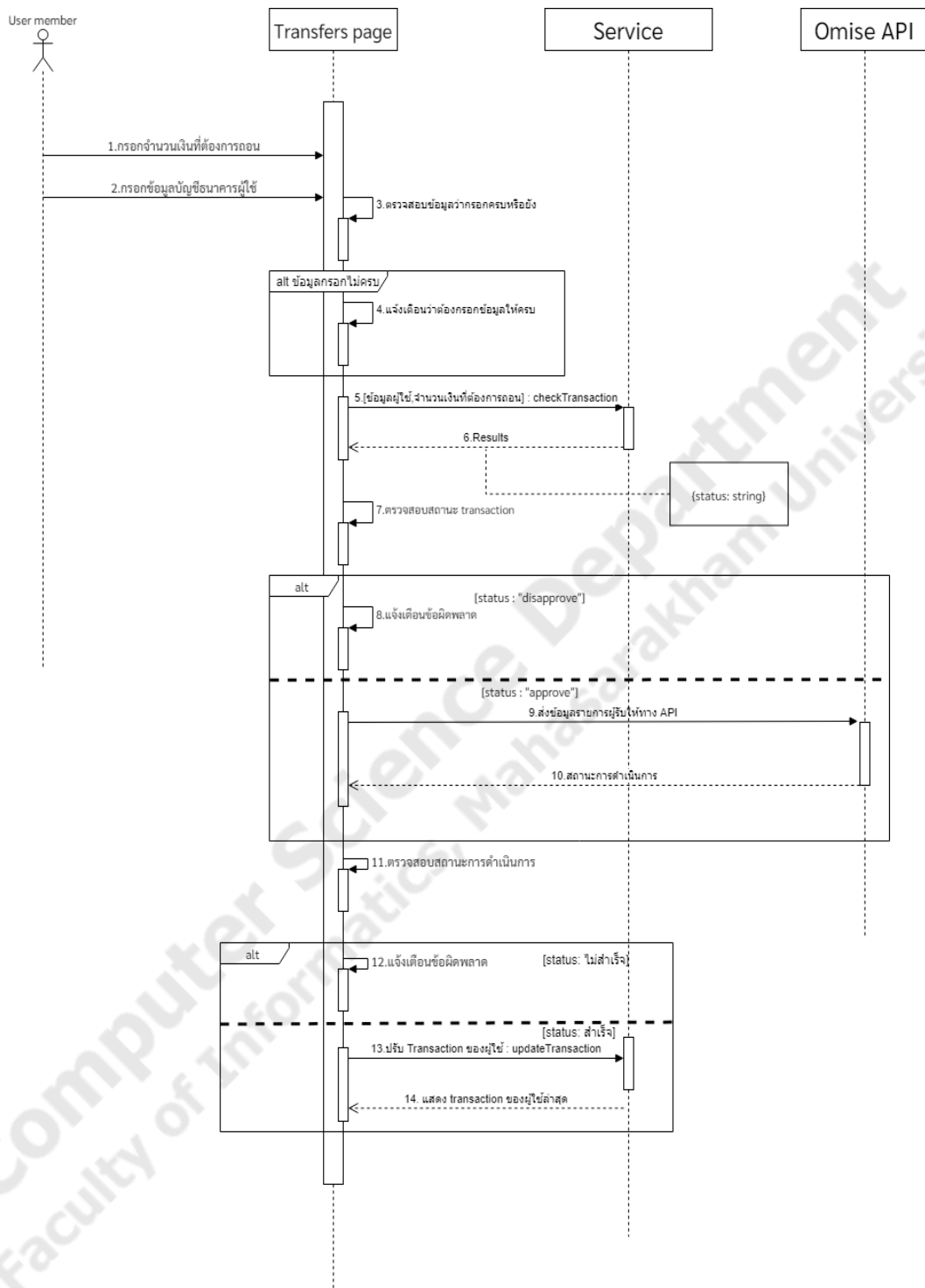
ภาพประกอบที่ 3.10 Sequence Diagram การเพิ่มโพสต์สูตรอาหาร



ภาพประกอบที่ 3.11 Sequence Diagram การแก้ไขโพสต์สูตรอาหาร



ภาพประกอบที่ 3.12 Sequence Diagram การเติมเงินเข้าระบบ



ภาพประกอบที่ 3.13 Sequence Diagram การถอนเงินออกจากระบบ

### 3.9.2 การเริ่มต้นโปรแกรม

```

Run | Debug
9  void main() {
10 |   runApp(MyApp());
11 | }
12
13 class MyApp extends StatelessWidget {
14 |   // This widget is the root of your application.
15 |   @override
16 |   Widget build(BuildContext context) {
17 |     return MaterialApp(
18 |       home: LoginPage(),
19 |       routes: {
20 |         '/register-page': (context) => RegisterPage(),
21 |         '/register2-page': (context) => RegisterPage2(),
22 |         '/register3-page': (context) => RegisterPage3(),
23 |         '/login-page': (context) => LoginPage()
24 |       },
25 |     ); // MaterialApp
26 |   }
27 | }

```

ภาพประกอบที่ 3.14 คำสั่งเริ่มต้นของโปรแกรม



ภาพประกอบที่ 3.15 หน้า Login

จากภาพประกอบที่ 3.14 คำสั่งเริ่มต้นของโปรแกรม บรรทัดที่ 10 เป็นคำสั่งเริ่มต้นของโปรแกรม เพื่อรันแอปพลิเคชัน ในคำสั่งเรียกใช้ MyApp() ซึ่งใน MyApp จะทำการ return MaterialApp ซึ่งจะเรียกใช้ class LoginPage และจะได้ผลลัพธ์เป็น ภาพประกอบที่ 3.15 หน้า Login

### 3.9.3 การเข้าสู่ระบบ

```

24
25 Future<LoginModel> logins(String email, String password) async {
26   final String apiUrl = "http://apifood.comsciproject.com/pjUsers/signin";
27
28   final response = await http
29     .post(Uri.parse(apiUrl), body: {"email": email, "password": password});
30
31   if (response.statusCode == 200) {
32     final String responseString = response.body;
33
34     return loginModelFromJson(responseString);
35   } else {
36     return null;
37   }
38 }

```

ภาพประกอบที่ 3.16 ฟังก์ชันการส่งข้อมูล Login

```

187 router.post("/signin", (req, res) => {
188   let body = req.body
189
190   pool.query("SELECT `password` FROM `pj_user` WHERE `email` = ?", [body.email], (err, results, fields) => {
191     if (err) {
192       return res.json({
193         success: 0,
194         message: err
195       });
196     }
197     if (results[0] == null) {
198       return res.json({
199         success: 0,
200         message: "อีเมล หรือ รหัสผ่าน ไม่ถูกต้อง"
201       });
202     }
203     let check = passwordHash.verify(body.password, results[0].password)
204     if (check == false) {
205       return res.json({
206         success: 0,
207         message: "รหัสผ่านไม่ถูกต้อง"
208       });
209     }
210     pool.query("SELECT * FROM `pj_user` WHERE `email` = ? and `password` = ?", [body.email, results[0].password], (err, results, field) => {
211
212       jwt.sign({ user: results[0].user_ID }, key, (err, token) => {
213         return res.json({
214           success: 1,
215           token
216         });
217       });
218     });
219   });
220 }
221

```

ภาพประกอบที่ 3.17 Service การ Login

```

169  ✓  onPressed: () async {
170      final String email = _ctrlEmail.text;
171      final String password = _ctrlPassword.text;
172
173      final LoginModel login = await logins(email, password);
174
175      _login = login;
176
177  ✓  if (_login.success == 1) {
178
179      //Token
180      Token_jwt().storeToken(_login.token);
181  ✓  Navigator.pushReplacement(
182      context,
183  ✓  new MaterialPageRoute(
184      /*check()*/
185      builder: (context) =>
186  |      new SlidePage()), // MaterialPageRoute
187  ✓  ).then((value) {
188
189      setState(() {});
190
191  |  });
192  }

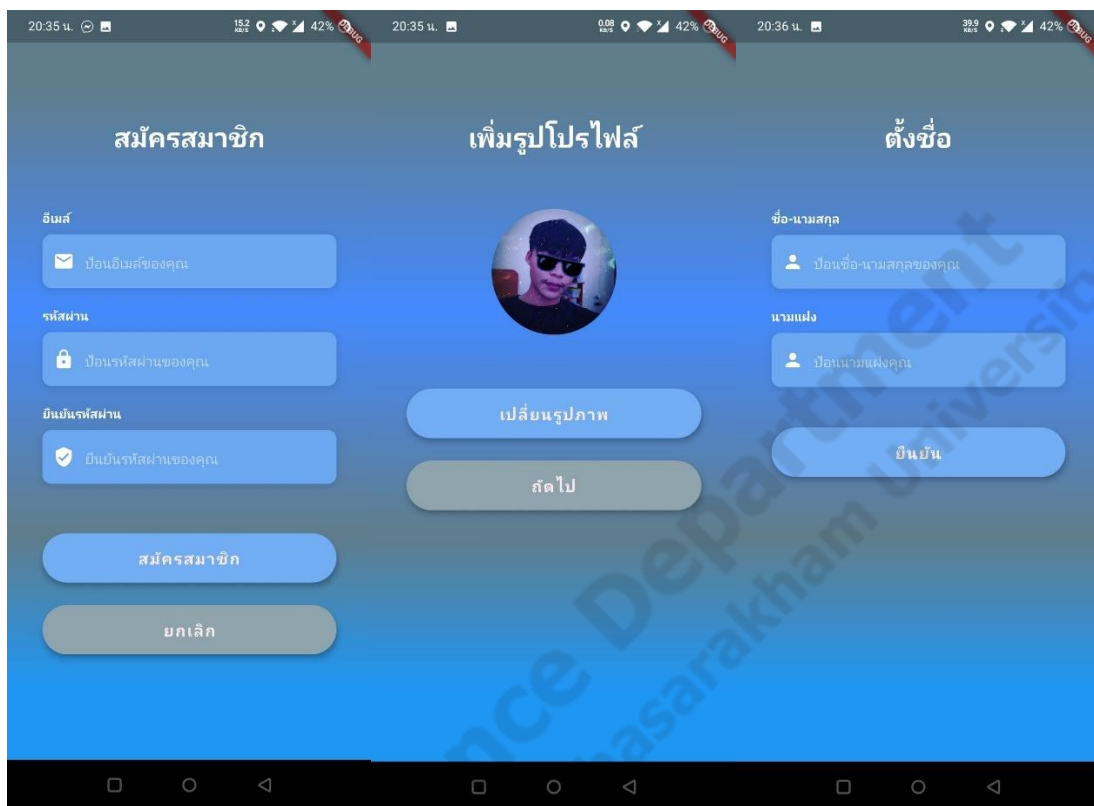
```

ภาพประกอบที่ 3.18 ตรวจสอบเงื่อนไขหลังส่งข้อมูล email password

จากภาพประกอบที่ 3.16 ฟังก์ชันการส่งข้อมูล Login จะส่งข้อมูล email และ password ไปยัง Service จากนั้นฝั่ง Backend ก็ทำการ select ข้อมูล password โดย email ที่ส่งมา ผลลัพธ์จะเป็น password ที่ถูก hash ไว้จากนั้นจะนำ password ที่ได้มานั้นมา verify แล้วถ้า verify แล้วผลลัพธ์ที่ได้เป็น true ก็ทำการ select เอา user\_ID จากนั้นสร้าง token ด้วย jsonwebtoken แล้ว นำ user\_ID มาใส่ไว้ใน payload แล้ว return ค่าออกไปเป็น success: 1, token ตามภาพประกอบที่ 3.17 Service การ Login และจากภาพประกอบที่ 3.18 ตรวจสอบเงื่อนไขหลังส่งข้อมูล email password จะตรวจสอบข้อมูลที่ Service ส่งกลับมา ถ้า success เป็น 1 ให้ดำเนินการในหน้าต่อไป



## 3.9.4 การสมัครสมาชิก



ภาพประกอบที่ 3.19 หน้า สมัครสมาชิก

```

18 Future<RegisterModel> registers(String email, String password) async {
19 |
20   final String apiUrl = "http://apifood.comsciproject.com/pjUsers/signupNewStep1";
21
22   final response = await http
23     .post(Uri.parse(apiUrl), body: {"email": email, "password": password});
24
25   if (response.statusCode == 200) {
26     final String responseString = response.body;
27
28     return registerModelFromJson(responseString);
29   } else {
30     return null;
31   }
32 }

```

ภาพประกอบที่ 3.20 ฟังก์ชันการส่งข้อมูลหน้า สมัครสมาชิก email password

```

42
43 Future<Register2Model> registers2(String tokens, File profile_image) async {
44     // final String apiUrl = "http://apifood.comsciproject.com/pjUsers/signin";
45     final String apiUrl =
46         "http://apifood.comsciproject.com/pjUsers/signupNewStep2";
47
48     final mimeTypeData =
49         lookupMimeType(profile_image.path, headerBytes: [0xFF, 0xD8]).split('/');
50
51     final imageUploadRequest = http.MultipartRequest('POST', Uri.parse(apiUrl));
52
53     final file = await http.MultipartFile.fromPath(
54         'profile_image', profile_image.path,
55         contentType: new MediaType(mimeTypeData[0], mimeTypeData[1]));
56     imageUploadRequest.files.add(file);
57     imageUploadRequest.fields['token'] = tokens;
58
59
60     var streamedResponse = await imageUploadRequest.send();
61     var response = await http.Response.fromStream(streamedResponse);
62
63     if (response.statusCode == 200) {
64         final String responseString = response.body;
65
66         return register2ModelFromJson(responseString);
67     } else {
68         return null;
69     }
70 }
71

```

ภาพประกอบที่ 3.21 ฟังก์ชันการส่งข้อมูลหน้า สมัครสมาชิก เพิ่มรูปโปรไฟล์

```

24 Future<Register2Model> registers2(
25     String name_surname, String alias_name) async {
26     final String apiUrl =
27         "http://apifood.comsciproject.com/pjUsers/signupNewStep3";
28
29     print("token" + token);
30
31     final response = await http.post(Uri.parse(apiUrl),
32         body: {"name_surname": name_surname, "alias_name": alias_name},
33         headers: {"Authorization": "Bearer $token"});
34
35     if (response.statusCode == 200) {
36         final String responseString = response.body;
37
38         return register2ModelFromJson(responseString);
39     } else {
40         return null;
41     }

```

ภาพประกอบที่ 3.22 ฟังก์ชันการส่งข้อมูลหน้า สมัครสมาชิก เพิ่มชื่อ-นามสกุล และนามแฝง

```

83
84 router.post("/signupNewStep1", (req, res) => {
85   let body = req.body
86   //console.log()
87   body.password = passwordHash.generate(body.password)
88   //let img = "http://apifood.comsciproject.com/uploadProfile/img_avatar.png"
89   let path = "http://apifood.comsciproject.com/uploadProfile/img_avatar.png"
90   let alias = "ท่านสมาชิก"
91
92   pool.query("INSERT INTO `pj_user` (`email`,`facebookID`, `password`, `name_surname`, `alias_name`, `user_status`, `access_status`, `balance`, `profile_i
93   if (err) {
94     if (err.errno == 1062) {
95       return res.json({
96         success: 0,
97         message: "อีเมล "+body.email+ " ถูกใช้งานแล้ว"
98       })
99     }
100    return res.json({
101      success: 0,
102      message: err
103    })
104  }
105  pool.query("select `user_ID` from `pj_user` where email = ? and password = ?", [body.email, body.password], (err, results1, field) => {
106    if (err) {
107      return res.json({
108        message: err,
109        success: 0
110      })
111    }
112  })
113
114  jwt.sign({ user: results1[0].user_ID }, key, (err, token) => {
115    return res.json({
116      success: 1,
117      token
118    })
119  })
120 })
121 })
122

```

ภาพประกอบที่ 3.23 Service การสมัครสมาชิกสแต็ปแรก

```

124 router.post("/signupNewStep2", upload.single("profile_image"), (req, res) => {
125   let body = req.body
126   let path = pathHttp + req.file.path
127   jwt.verify(body.token, key, (err, authData) => {
128     if (err) { res.json({ success: 0, message: err }) }
129     let uid = authData.user
130
131     pool.query("update `pj_user` set `profile_image`=? where user_ID = ?", [path, uid], (err, results, field) => {
132       if (err) {
133         return res.json({
134           message: err,
135           success: 0
136         })
137       }
138       if (results.affectedRows == 1) {
139         return res.json({
140           success: 1
141         })
142       } else {
143         return res.json({
144           success: 0
145         })
146       }
147     })
148   })
149 })

```

ภาพประกอบที่ 3.24 Service การสมัครสมาชิกสแต็ปสอง

```

151 router.post("/signupNewStep3", auth.verifyToken, (req, res) =>{
152   jwt.verify(req.token, key, (err, authData) =>{
153     if(err){res.json({success: 0, message: err});}
154     let uid = authData.user
155     let body = req.body
156     pool.query("select * from pj_user where user_ID = ?", [uid], (err, results, field) =>{
157
158       let oldData = results[0]
159       if (results == "") {
160         return res.json({
161           success: 0,
162           message: "nodata"
163         })
164       }
165
166       let jsonOldData = JSON.parse(JSON.stringify(oldData))
167       let newData = mergeJSON.merge(jsonOldData, body)
168
169       pool.query("update `pj_user` set `name_surname`=?, `alias_name` = ? where user_ID = ?", [newData.name_surname, newData.alias_name, uid], (err, results)
170         =>{
171         if(err){
172           res.json({
173             message: err,
174             success : 0
175           })
176         }
177         else{
178           return res.json({
179             success : 1
180           })
181         }
182       })
183     })
184   })
185 })
186

```

ภาพประกอบที่ 3.25 Service การสมัครสมาชิกสตีปสาม

จากภาพประกอบที่ 3.20 ฟังก์ชันการส่งข้อมูลหน้า สมัครสมาชิก email password จะทำการส่งข้อมูลไปยัง Service การสมัครสมาชิก ตามภาพประกอบที่ 3.23 Service การสมัครสมาชิกสตีปแรก ฟังก์ชัน Backend นี้จะทำการ hash password ของผู้ใช้ แล้ว insert ข้อมูลไปใน Database แล้วก็จะ select user\_ID ออกมาใส่ไว้ใน payload แล้วส่ง token มาเก็บไว้ในตัวแอป จากนั้น ภาพประกอบที่ 3.21 ฟังก์ชันการส่งข้อมูลหน้า สมัครสมาชิก เพิ่มรูปโปรไฟล์ จะทำการส่งรูปภาพเป็นไฟล์ ผ่าน fromdata และภาพประกอบที่ 3.24 Service การสมัครสมาชิกสตีปสอง จะรอรับไฟล์ที่ส่งมาจากฝั่ง app และ update path ของรูปภาพลงใน database และภาพประกอบที่ 3.22 ฟังก์ชันการส่งข้อมูลหน้า สมัครสมาชิก เพิ่มชื่อ-นามสกุล และนามแฝง ก็จะส่งข้อมูลไปยัง Service สตีปสามตามภาพประกอบที่ 3.25 Service การสมัครสมาชิกสตีปสาม โดยจะ update ชื่อ-นามสกุล และนามแฝง ของผู้ใช้งานที่ส่งมา



ภาพประกอบที่ 3.26 หน้า เพิ่มสูตรอาหาร

```

39 // CreatePostModel
40 Future<CreatePostModel> createPosts(
41     String tokens, File image, String recipe_name, double price) async {
42
43     final String apiUrl = "http://apifood.comsciproject.com/pjPost/createPost";
44
45     mimeTypeData =
46         | lookupMimeType(image.path, headerBytes: [0xFF, 0xD8]).split('/');
47
48     final imageUploadRequest = http.MultipartRequest('POST', Uri.parse(apiUrl));
49
50     final file = await http.MultipartFile.fromPath('image', image.path,
51         contentType: new MediaType(mimeTypeData[0], mimeTypeData[1]));
52
53     imageUploadRequest.files.add(file);
54     imageUploadRequest.fields['token'] = tokens;
55     imageUploadRequest.fields['recipe_name'] = recipe_name;
56     imageUploadRequest.fields['price'] = price.toString();
57     var streamedResponse = await imageUploadRequest.send();
58     var response = await http.Response.fromStream(streamedResponse);
59
60     if (response.statusCode == 200) {
61         final String responseString = response.body;
62
63         return createPostModelFromJson(responseString);
64     } else {
65         return null;
66     }
67

```

ภาพประกอบที่ 3.27 ฟังก์ชันการส่งข้อมูล การสร้าง Post

```

83 Future<AddIngredientsArrayModel> addIngredients(String recipe_ID,
84 List<String> ingredientName, List<String> amount, List<String> step) async {
85     final String apiUrl =
86         "http://apifood.comsciproject.com/pjPost/addIngredientsArray";
87
88
89     // List<st>
90     var data = {
91         "recipe_ID": recipe_ID,
92         "ingredientName": ingredientName,
93         "amount": amount,
94         "step": step
95     };
96
97     final response = await http.post(Uri.parse(apiUrl),
98         body: jsonEncode(data),
99         headers: {
100             "Authorization": "Bearer $token",
101             "Content-Type": "application/json"
102         });
103
104
105     if (response.statusCode == 200) {
106         final String responseString = response.body;
107
108         return addIngredientsArrayModelFromJson(responseString);
109     } else {
110         return null;
111     }
112
113

```

ภาพประกอบที่ 3.28 ฟังก์ชันการส่งข้อมูล การเพิ่มข้อมูลวัตถุดิบ

```

114 Future<AddHowtoArrayModels> addHowtos(
115     String recipe_ID,
116     List<String> description,
117     List<String> step,
118     List<String> path_file,
119     List<String> type_file) async {
120     final String apiUrl = "http://apifood.comsciproject.com/pjPost/addHowtoArray";
121
122     var data = {
123         "recipe_ID": recipe_ID,
124         "description": description,
125         "step": step,
126         "path_file": path_file,
127         "type_file": type_file
128     };
129
130     print(jsonEncode(data));
131     final response = await http.post(Uri.parse(apiUrl),
132         body: jsonEncode(data),
133         headers: {
134             "Authorization": "Bearer $token",
135             "Content-Type": "application/json"
136         });
137
138     if (response.statusCode == 200) {
139         final String responseString = response.body;
140
141         return addHowtoArrayModelsFromJson(responseString);
142     } else {
143         return null;
144     }
145
146

```

ภาพประกอบที่ 3.29 ฟังก์ชันการส่งข้อมูล การเพิ่มวิธีทำ

```

140
147 < Future<UploadHowtoFileModels> addloadHowtoFiles(File image) async {
148   // final String apiUrl = "http://apifood.comsciproject.com/pjUsers/signin";
149   final String apiUrl =
150     | "http://apifood.comsciproject.com/pjPost/uploadHowtoFile";
151
152   mimeTypeData =
153     | lookupMimeType(image.path, headerBytes: [0xFF, 0xD8]).split('/');
154
155   final imageUploadRequest = http.MultipartRequest('POST', Uri.parse(apiUrl));
156
157   < final images = await http.MultipartFile.fromPath('file', image.path,
158     | contentType: new MediaType(mimeTypeData[0], mimeTypeData[1]));
159
160   imageUploadRequest.files.add(images);
161
162   var streamedResponse = await imageUploadRequest.send();
163   var response = await http.Response.fromStream(streamedResponse);
164
165   < if (response.statusCode == 200) {
166     final String responseString = response.body;
167     return uploadHowtoFileModelsFromJson(responseString);
168   } else {
169     return null;
170   }
171 }
172

```

ภาพประกอบที่ 3.30 ฟังก์ชันการส่งข้อมูล การเพิ่มรูปภาพหรือวิดีโอของวิธีทำ

```

118 router.post("/createPost",uploadRecipe.single("image"),(req,res) =>{
119   //console.log(req,file)
120   let path = pathHttp + req.file.path
121   let body = req.body
122
123   jwt.verify(body.token,key,(err,authData) =>{
124     if(err){
125       return res.json({
126         message: err
127       })
128     }
129
130     let uid = authData.user
131     //console.log(parseFloat(body.price))
132     pool.query("INSERT INTO `pj_recipe` (`user_ID`, `recipe_name`, `image`, `date`,`price`) VALUES (?, ?, ?, now(),?)",[uid,body.recipe_name,path,body.price]
133     if(error){
134       res.json({
135         message: error
136       })
137     }
138
139
140     if(results.affectedRows == 1){
141       pool.query("SELECT `rid` FROM `pj_recipe` WHERE `user_ID` = ? ORDER BY `rid` DESC LIMIT 1",[uid],[error,result,field] =>{
142         return res.json({
143           success: 1,
144           recipe_ID: result[0].rid
145         })
146       }
147     }else{
148       return res.json({
149         success: 0,
150         message: "something failed!"
151       })
152     }
153   })
154 })
155 })
156

```

ภาพประกอบที่ 3.31 Service ในการสร้างPost

```

187 router.post("/addIngredientsArray",auth.verifyToken,(req,res) =>{
188
189     jwt.verify(req.token,key,(err,authData) =>{
190         if(err){
191             return res.json({
192                 message: err
193             })
194         }
195
196         let body = req.body
197         var countLoop = 0
198         //console.log(body.amount)
199         for(var i=0;i<body.step.length;i++){
200             pool.query("INSERT INTO 'pj_ingredients' ('rid', 'ingredientName', 'amount', 'step') VALUES (?, ?, ?, ?)",[body.recipe_ID,body.ingredientName[i],body.
201             if(error){
202                 res.json({
203                     success: 0,
204                     message: err
205                 })
206             }
207             if(result.affectedRows == 1){
208
209                 countLoop +=1
210                 if(countLoop == body.step.length){
211                     res.json({
212                         success: 1
213                     })
214                 }
215             }else{
216                 return res.json({
217                     success: 0
218                 })
219             }
220         })
221     })
222 })
223 })
224 })

```

ภาพประกอบที่ 3.32 Service การเพิ่มวัตถุดิบที่ใช้

```

229 router.post("/addHowtoArray",auth.verifyToken,(req,res) =>{
230
231     jwt.verify(req.token,key,(err,authData) =>{
232         if(err){
233             res.json({
234                 message: err
235             })
236         }
237
238         let body = req.body
239         var countLoop = 0
240         for(var i=0;i<body.step.length;i++){
241             //console.log(body.type_file[i])
242             pool.query("INSERT INTO 'pj_howto' ('rid', 'description', 'step', 'path_file', 'type_file') VALUES (?, ?, ?, ?, ?)",[body.recipe_ID,body.description[i],
243             if(error){
244                 res.json({
245                     success: 0,
246                     message: err
247                 })
248             }
249             if(result.affectedRows == 1){
250                 countLoop +=1
251                 if(countLoop == body.step.length){
252                     res.json({
253                         success: 1
254                     })
255                 }
256             }else{
257                 return res.json({
258                     success: 0
259                 })
260             }
261         })
262     })
263 })
264 })
265 })

```

ภาพประกอบที่ 3.33 Service การเพิ่มวิธีทำ

```

73 router.post("/uploadHowtoFile",uploadHowto.single("file"),(req,res) =>{
74     console.log(req.file)
75     let type = req.file.mimetype
76     let typeArr = type.split('/')
77     type = typeArr[0]
78     let path = pathHttp + req.file.path
79     res.json({
80         path: path,
81         type: type
82     })
83 })
84

```

ภาพประกอบที่ 3.34 Service การเพิ่มไฟล์ รูปภาพ วิดีโอ



จากภาพประกอบที่ 3.26 หน้า เพิ่มสูตรอาหาร เป็นหน้าในการกรอกข้อมูลจะมีการกรอก ชื่อสูตรอาหาร , รูปปกอาหาร ส่วนผสม วิธีทำและราคาสูตรอาหาร ส่วนในฟังก์ชันการส่งข้อมูลการสร้างโพสต์ ตามภาพประกอบที่ 3.27 ฟังก์ชันการส่งข้อมูล การสร้าง Post และฝั่ง Service ก็จะรับข้อมูลจากแอปที่ส่งมา ตามภาพประกอบที่ 3.31 Service ในการสร้างPost ที่จะบันทึกข้อมูลลงฐานข้อมูลแล้ว return ค่า Recipe\_ID ที่ได้จากการ insert แล้วจาก Recipe\_ID ที่ส่งมานั้นจะได้นำไปใช้เป็น foreign key ของการเพิ่มข้อมูลวัตถุดิบ ตามภาพประกอบที่ 3.28 ฟังก์ชันการส่งข้อมูล การเพิ่มข้อมูลวัตถุดิบ และวิธีทำ ตามภาพประกอบที่ 3.29 ฟังก์ชันการส่งข้อมูล การเพิ่มวิธีทำ โดยการเพิ่มวิธีทำนั้นจะสามารถเพิ่มรูปหรือวิดีโอแนบไปพร้อมกับรายละเอียดวิธีทำนั้นได้ โดยจะเรียกใช้ Service ในการอัปโหลด file ก่อน ตามภาพประกอบที่ 3.34 Service การเพิ่มไฟล์ รูปภาพ วิดีโอ โดยเมื่อไฟล์ถูกอัปโหลดเข้าไปเสร็จทาง Service จะ return ค่าที่เป็น path ที่เก็บของไฟล์และประเภทของไฟล์นั้นมา เพื่อใช้ในการเก็บข้อมูลนี้ลงในฐานข้อมูล ตามภาพประกอบที่ 3.33 Service การเพิ่มวิธีทำ



ภาพประกอบที่ 3.35 หน้าแสดงข้อมูลโปรไฟล์

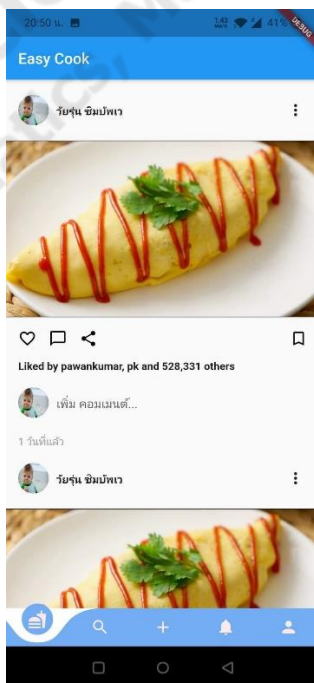
```

267 router.get("/mypost/:id",auth.verifyToken,(req,res) =>{
268   jwt.verify(req.token,key,(err,authData) =>{
269     if(err){
270       return res.json({
271         message: err
272       })
273     }
274   })
275
276   let id = req.params.id
277   pool.query("SELECT pj_recipe.rid, pj_user.user_id,pj_user.name_surname, pj_user.alias_name,pj_user.profile_image,pj_recipe.recipe_name, pj_recipe.image, pj_recipe.steps FROM pj_recipe JOIN pj_user ON pj_recipe.user_id = pj_user.user_id WHERE pj_recipe.rid = ? ORDER BY step ASC", [id], (error, result, field) => {
278     //console.log(result[0])
279     var data = {}
280     var dataHowto = []
281     var loop = 0
282     var countLoop = result.length
283     result.forEach(element => {
284       console.log(element.rid)
285       pool.query("SELECT * FROM 'pj_ingredients' WHERE rid= ? ORDER BY step ASC", [element.rid], (error, result1, field) => {
286         data.push(result1)
287         loop+=1
288       })
289       if(countLoop == loop){
290         var loop1 = 0
291         var countLoop1 = result.length
292         result.forEach(element => {
293           pool.query("SELECT * FROM 'pj_Howto' WHERE rid= ? ORDER BY step ASC", [element.rid], (error, result2, field) => {
294             dataHowto.push(result2)
295             loop1+=1
296             if(countLoop1 == loop1){
297               res.json({post: result, indient: data, Howto: dataHowto})
298             }
299           })
300         })
301       }
302     })
303   })
304 }
305 )
306 }
307 }

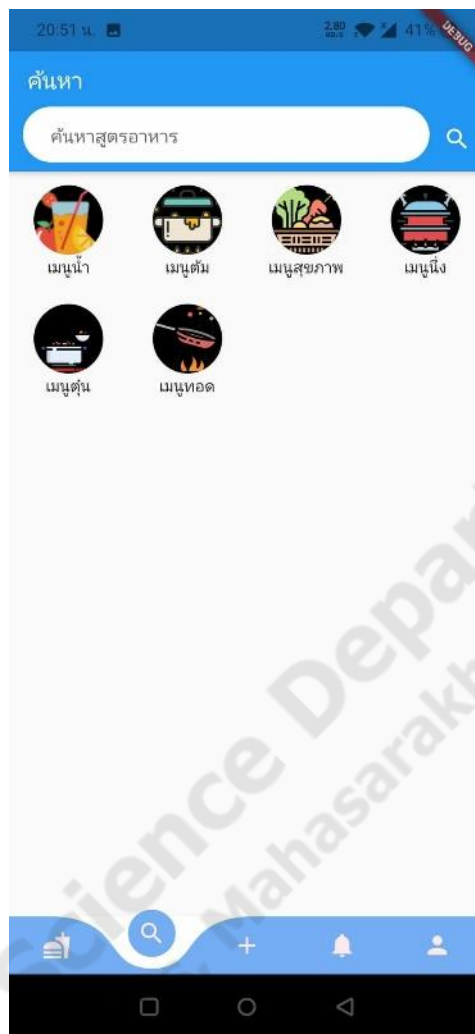
```

ภาพประกอบที่ 3.36 Service การแสดงรายละเอียดของ post

จากภาพประกอบที่ 3.35 หน้าแสดงข้อมูลโปรไฟล์ จะนำข้อมูลที่ได้จากการเป็นสมาชิก มาแสดงในแอปพลิเคชัน โดยจะแสดง ชื่อ รูปโปรไฟล์ จำนวนโพสต์ จำนวนคนติดตาม จำนวนคนที่กำลังติดตาม และ โพสต์สูตรอาหารของผู้ใช้ โดยโพสต์สูตรอาหารจะแสดงเป็น List ออกมาโดยเรียกข้อมูลผ่าน Service ตามภาพประกอบที่ 3.36 Service การแสดงรายละเอียดของ post ให้มาแสดงในแอปพลิเคชัน



ภาพประกอบที่ 3.37 หน้าการแสดงผลข้อมูลโพสต์สูตรอาหาร feeds



ภาพประกอบที่ 3.38 หน้าแสดงฟังก์ชันการค้นหา

จากภาพประกอบที่ 3.37 หน้าการแสดงผลข้อมูลโพสต์สูตรอาหาร feeds จะแสดงข้อมูลที่เกี่ยวข้องกับโพสต์สูตรอาหาร โดยจะประกอบไปด้วย ชื่อ รูปภาพโปรไฟล์ รูปปกสูตรอาหาร ชื่อสูตรอาหาร ปุ่มการให้คะแนนสูตรอาหาร คอมเมนต์ และส่วนการเลือกโพสต์มาแสดงนั้นจะนำโพสต์ของผู้ใช้งานที่เราติดตามหรือ โพสต์ที่ได้รับความนิยม ส่วนฟังก์ชันในการค้นหานั้น สามารถค้นหาได้จากชื่อสูตรอาหาร หรือ วัตถุดิบที่ใช้ในสูตรอาหาร