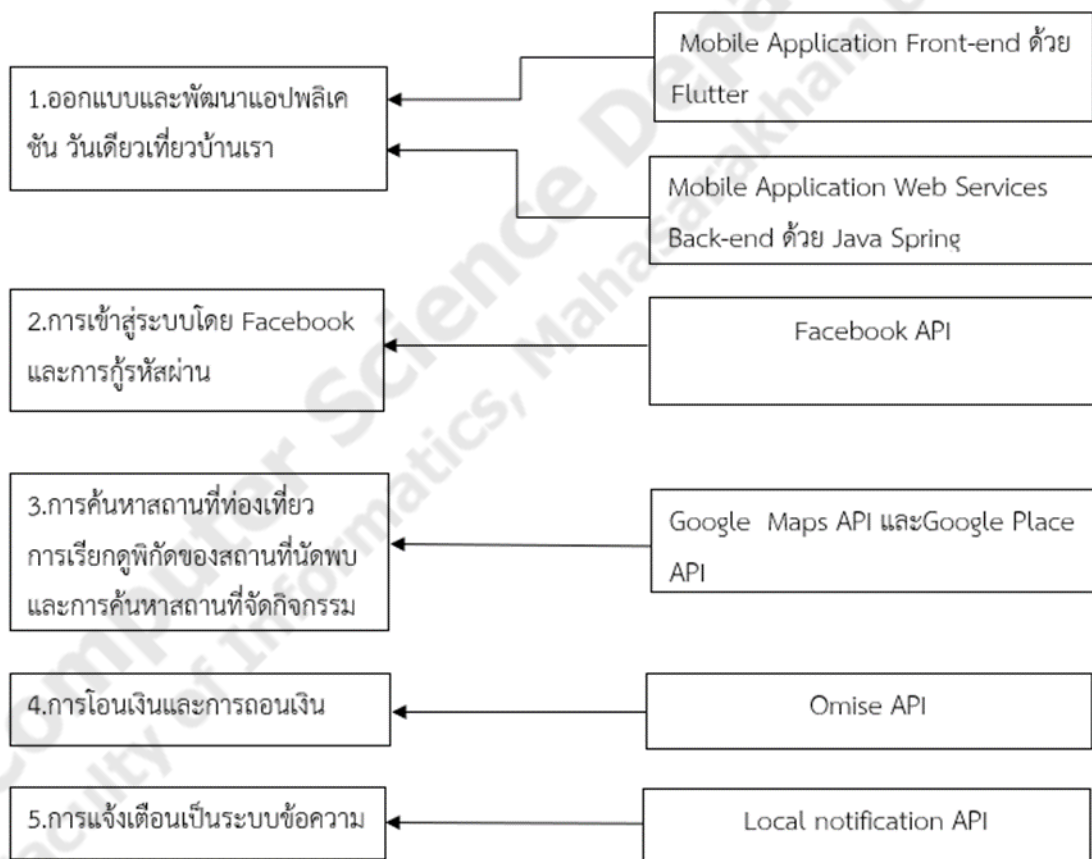


### บทที่ 3

#### ขั้นตอนการดำเนินงาน

สำหรับในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงการปริญญาโทซึ่งจะทำให้ทราบถึงการวิเคราะห์ และการออกแบบแอปพลิเคชันโดยละเอียดว่ามีแนวทางในกาดำเนินงาน หรือมีขั้นตอนการทำงานอย่างไรบ้าง โดยมีรายละเอียดดังนี้

##### 3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบ

ภาพกรอบการดำเนินงานที่ใช้อธิบายถึงขั้นตอนการดำเนินงานในการพัฒนาระบบ ดังภาพประกอบที่ 3.1

### 3.1.1 คำอธิบาย

1. เขียนโปรแกรมออกแบบและพัฒนาแอปพลิเคชัน การวิ่งออกกำลังกายคนเดียวด้วยกัน สำหรับผู้ใช้งานแอปพลิเคชันผ่านสมาร์ทโฟน โดยการสร้างแอปพลิเคชัน จะต้องมีการติดตั้งเครื่องมือ ดังนี้

1) Visual Studio Code หรือ VS Code เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ Open Source จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษาJavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมาก ในการสร้างแอปพลิเคชันนั้นจะต้องทำการติดตั้ง Flutter, Dart, Node JS สำหรับเขียนแอปพลิเคชัน ใช้ Dart สำหรับส่งข้อมูลกับ server ใช้ Node JS เมื่อติดตั้งเสร็จแล้วทำการสร้างแอปพลิเคชันตามหน้า GUI ที่ได้ออกแบบไว้แล้ว ดังนี้

1. หน้าหลักแสดงรายการทริป
2. หน้าเลือกประเภทสมัครสมาชิก
3. หน้าสมัครสมาชิก
4. หน้าล็อกอิน
5. หน้าแสดงชาร์ตคะแนนความนิยมไกด์
6. หน้าแสดงข้อมูลทริป
7. หน้าจองทริป
8. หน้าชำระเงิน
9. หน้าใบเสร็จ
10. หน้าเพิ่มข้อมูลยืนยันตัวตนของไกด์
11. หน้าสร้างทริป

2) Eclipse คือ โปรแกรม IDE ที่ใช้สำหรับช่วยนักพัฒนาโปรแกรมให้สามารถทำอะไรได้สะดวกสบายมากยิ่งขึ้นซึ่ง ณ ปัจจุบัน Eclipse นี้รองรับภาษา Java, php, C/C++, JavaScript และยังสามารถลง plugin เพิ่มเติม เช่น Google plugin for Eclipse และอื่นๆ อีกมากมายซึ่งจะช่วยให้เราสามารถทำอะไรได้ง่ายขึ้นและรวดเร็วมากขึ้น

## 2. การเข้าสู่ระบบโดย Facebook

Facebook API คือส่วนต่อประสานเชื่อมโยงกับ Facebook หมายถึงวิธีการที่ Facebook เปิดไลบรารีบริการ ช่วยให้นักพัฒนาเว็บไซต์เข้าถึงข้อมูลต่างๆของ Facebook ด้วย Facebook API เพื่อติดต่อเรียกใช้ข้อมูลจากเว็บไซต์ Facebook ได้เช่น การทำ Facebook Login เพื่อให้ลูกค้าซื้อสินค้าบนเว็บไซต์ด้วยบัญชี Facebook หรือการทำ Facebook Page Tab ร้านค้าบน Facebook, การทำ Auto Post ของเว็รด์เพลส Link ไปที่หน้า Facebook Wall หรือแฟนเพจ

3. การค้นหาสถานที่ท่องเที่ยว การเรียกดูพิกัดของสถานที่นั้นๆและการค้นหาสถานที่จัดกิจกรรม

1) Google Maps API เป็นชุด API ของ Google สำหรับพัฒนา web application และ mobile application (Android, iOS) ไว้สำหรับเรียกใช้แผนที่และชุด service ต่าง ๆ ของ Google เพื่อพัฒนา Application โดยการเขียนโปรแกรมด้วย flutter ต้องลงไลบรารีชื่อ google maps flutter เพื่อเรียกใช้ Google Map API

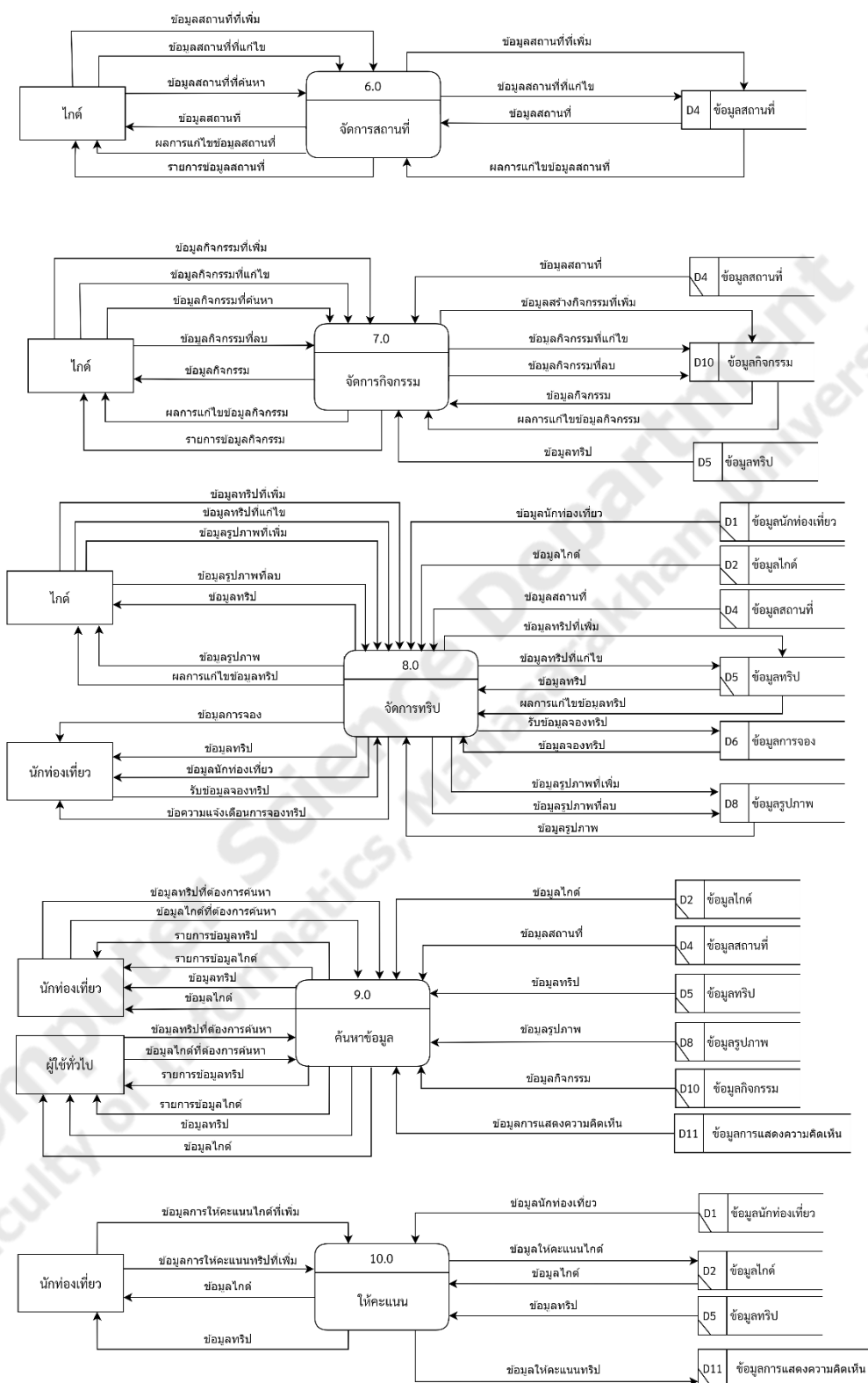
2) Google Place API เป็นชุด API ของ Google สำหรับพัฒนา แอปพลิเคชันสามารถดึง Place Picker เพื่อให้ผู้ใช้เลือกสถานที่ที่ตัวเองอยู่ในขณะนั้นได้เลย นอกจากนี้ยังมีระบบ Autocomplete ช่วยเดาชื่อสถานที่ได้แบบเดียวกับ Google Search, รายละเอียดของสถานที่นั้น เช่น หมายเลขโทรศัพท์ แบบเดียวกับที่แสดงบน Google Maps

4. การโอนเงินและการถอนเงิน Omise เป็นระบบรับชำระและจัดการเงินควบคุมได้เองทุกอย่าง ตั้งแต่การออกแบบหน้ารับชำระเงิน การรักษาความปลอดภัย ไปจนถึงการโอนเงินออกจากระบบ ซึ่งในปัจจุบันการชำระการซื้อสินค้าหรือบริการบนอินเทอร์เน็ต เราสามารถชำระเงินได้หลายรูปแบบไม่ว่าจะเป็นการโอนเงิน บิลเพย์เมนต์ อินเทอร์เน็ตแบงก์กิง บัตรเครดิตหรือบัตรเครดิต ก็สามารถทำได้บน Omise ทั้งสิ้นซึ่งสะดวกสำหรับธุรกิจ E-commerce

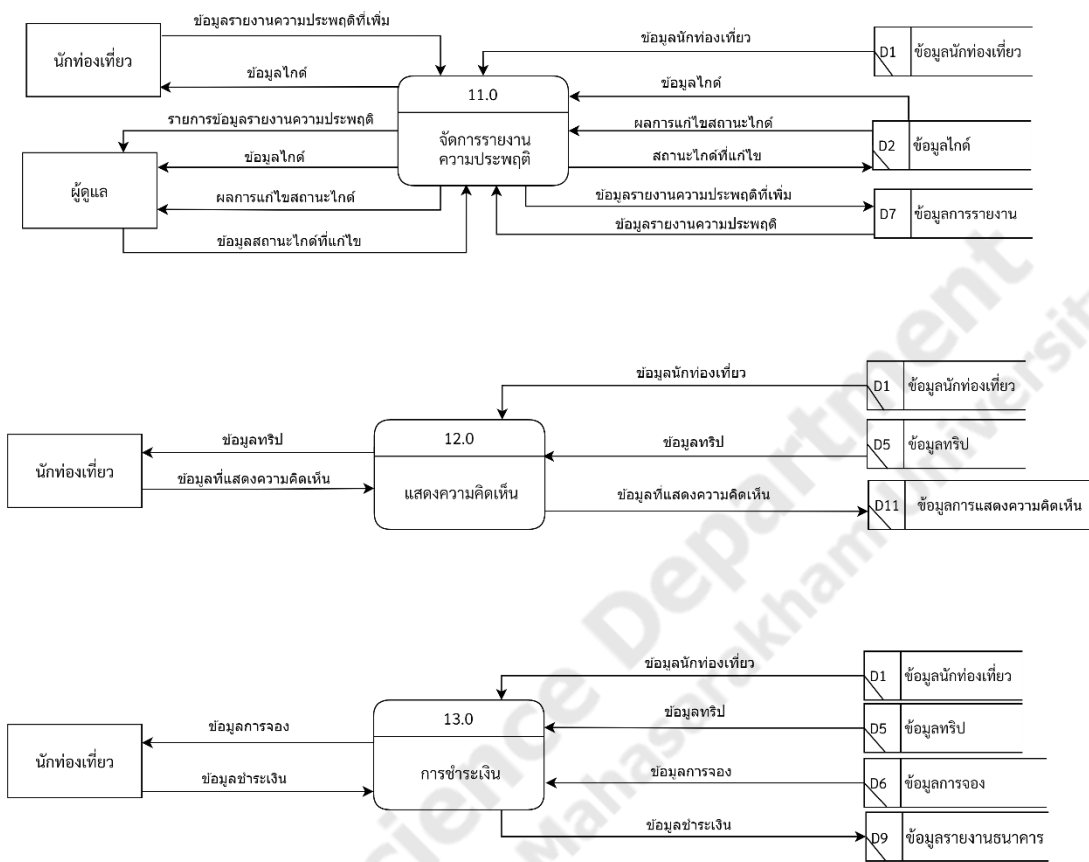
5. การแจ้งเตือนเป็นระบบข้อความ Push notification API คือ การแจ้งเตือนที่แอปพลิเคชันนำข้อมูลมาแสดงในการแจ้งเตือนของระบบปฏิบัติการนั้นๆไม่ว่าจะเป็น Mobile (iOS, Android)







ภาพประกอบที่ 3.3 DataFlow Diagram Level 1 (ต่อ)

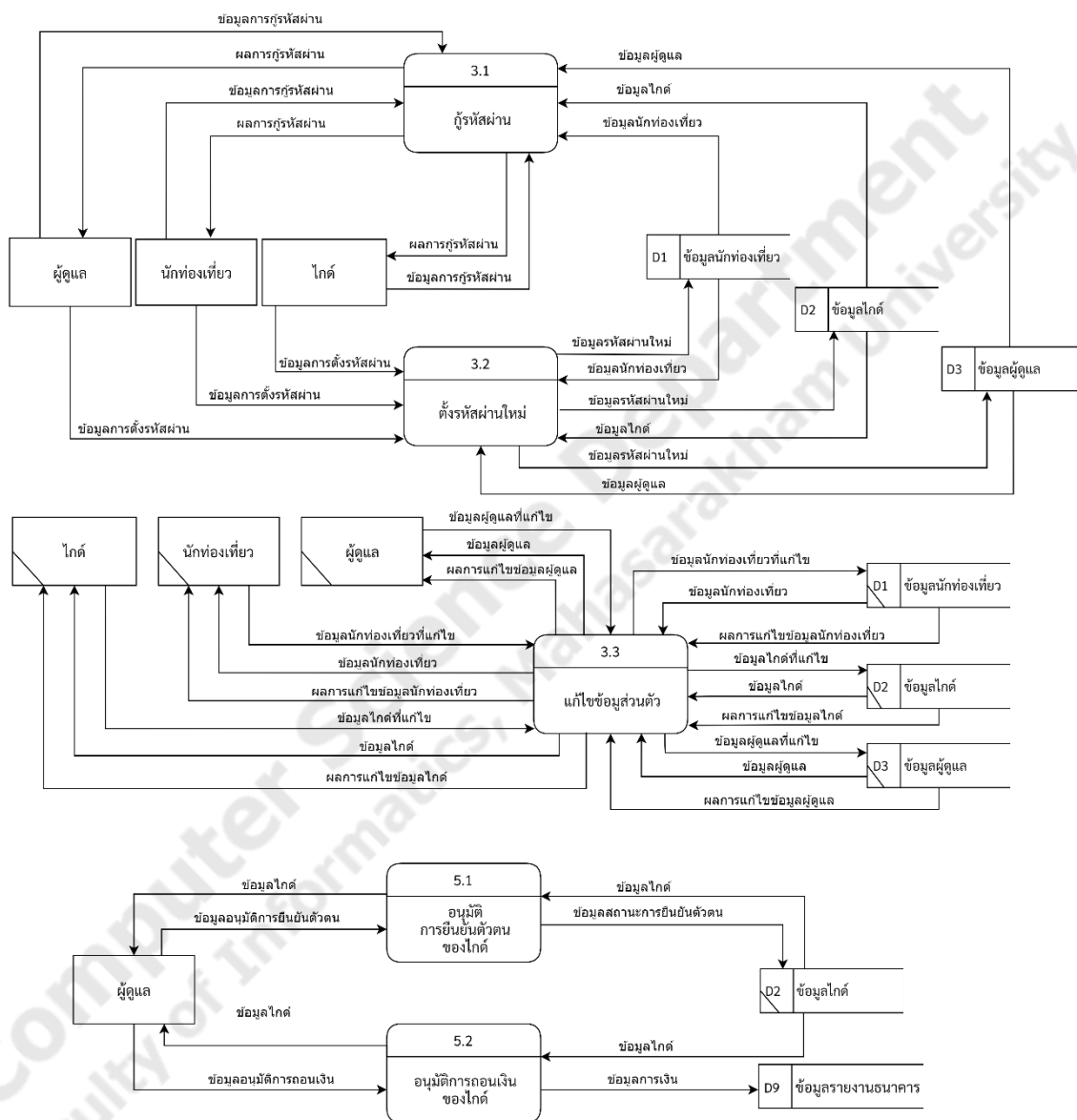


ภาพประกอบที่ 3.3 DataFlow Diagram Level 1 (ต่อ)

Computer Science Department  
Faculty of Informatics, Mahasarakham University

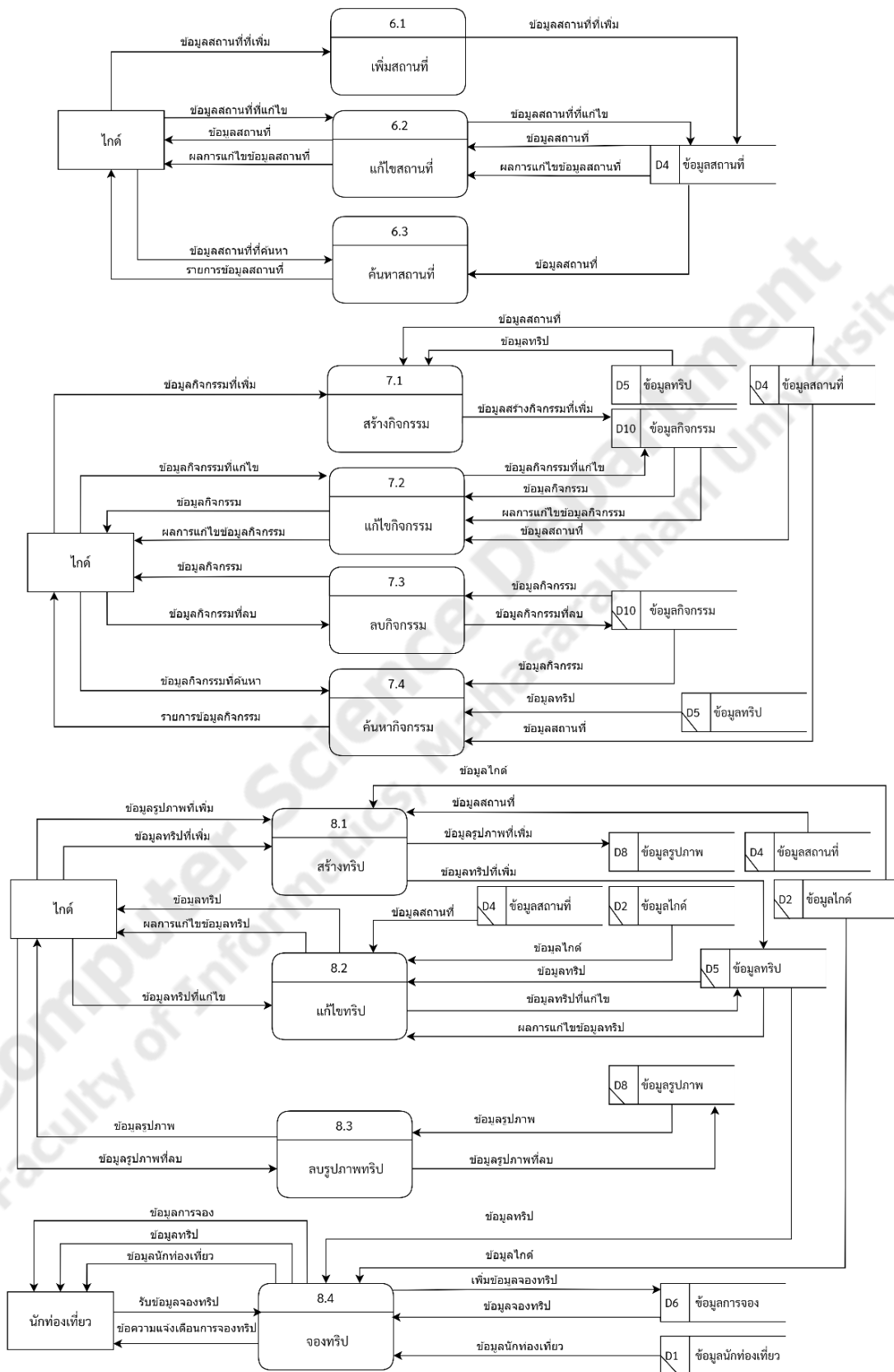
### 3.2.3 Data Flow Diagram Level 2

แผนภาพกระแสข้อมูลแสดงทิศทางการไหลของ Data Flow และแสดงรายละเอียดของแหล่งจัดเก็บข้อมูล (Data Store) ของ Data Flow Diagram Level 2 ดังภาพประกอบที่ 3.4

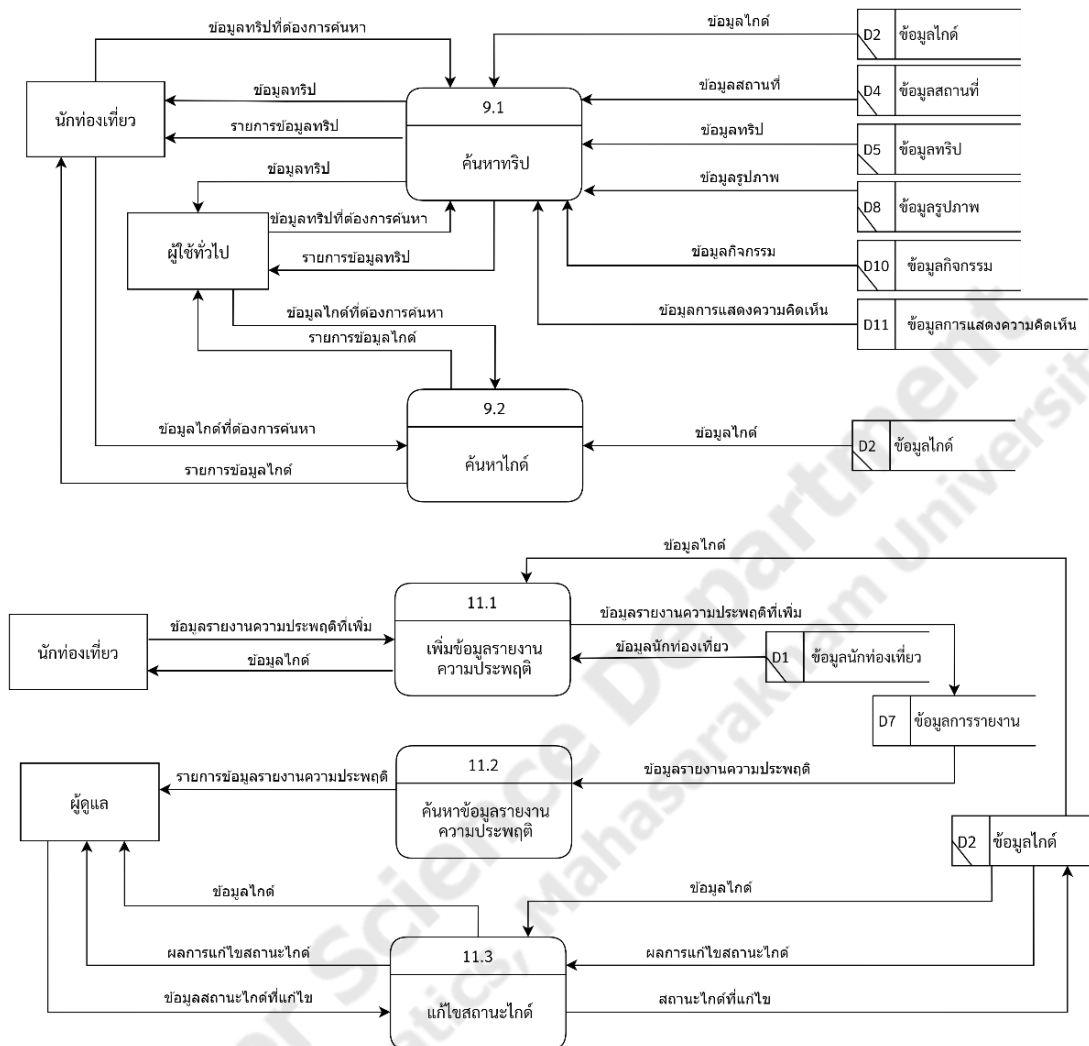


ภาพประกอบที่ 3.4 DataFlow Diagram Level 2





ภาพประกอบที่ 3.4 DataFlow Diagram Level 2 (ต่อ)



ภาพประกอบที่ 3.4 DataFlow Diagram Level 2 (ต่อ)

Computer Science Department  
 Faculty of Informatics Maharakam University

### 3.3 External Entity Description

พจนานุกรมข้อมูลช่วยอธิบายการทำงานของแผนภาพการไหลของข้อมูล (Data Flow Dictionaries)

รายละเอียดดังตารางที่ 3.1

ตารางที่ 3.1 External Entity Description

Name	Descriptions	Input Data Flow	Output Data Flow
ผู้ใช้ทั่วไป	ผู้ใช้งาน ประเภท ผู้ใช้งานทั่วไป เป็นผู้ใช้ที่ยังไม่ สมัครสมาชิก	-ข้อมูลทริป -ข้อมูลนักท่องเที่ยว -ข้อมูลไกด์ -รายการข้อมูลไกด์ -รายการข้อมูลทริป	-ข้อมูลสมัครสมาชิก -ข้อมูลทริปที่ต้องการค้นหา -ข้อมูลไกด์ที่ต้องการค้นหา
นักท่องเที่ยว	ผู้ใช้งาน ประเภท นักท่องเที่ยว เป็นผู้ใช้ที่สมัคร สมาชิก ประเภท นักท่องเที่ยว แล้ว	-ผลการเข้าสู่ระบบ -ผลการกู้รหัสผ่าน -ข้อมูลนักท่องเที่ยว -ผลการแก้ไขข้อมูล นักท่องเที่ยว -ข้อมูลการจอง -ข้อมูลทริป -ข้อความแจ้งเตือนการ จองทริป -รายการข้อมูลไกด์ -รายการข้อมูลทริป -ข้อมูลไกด์	-ข้อมูลชำระเงิน -ข้อมูลที่แสดงความคิดเห็น -ข้อมูลรายงานความประพฤติที่เพิ่ม -ข้อมูลการให้คะแนนการทริปที่เพิ่ม -ข้อมูลการให้คะแนนไกด์ที่เพิ่ม -ข้อมูลไกด์ที่ต้องการค้นหา -ข้อมูลทริปที่ต้องการค้นหา -รับข้อมูลจองทริป -ข้อมูลนักท่องเที่ยวที่แก้ไข -ข้อมูลการตั้งรหัสผ่าน -ข้อมูลการกู้รหัสผ่าน -ข้อมูลเข้าสู่ระบบนักท่องเที่ยว
ผู้ดูแล	เป็นผู้ดูแล ระบบของแอป พลิเคชันวัน เดียวเที่ยวบ้าน เรา	-ผลการเข้าสู่ระบบ -ผลการกู้รหัสผ่าน -ข้อมูลผู้ดูแล -ผลการแก้ไขข้อมูลผู้ดูแล -สถานะการเพิ่มข้อมูล ยืนยัน -ข้อมูลไกด์	-ข้อมูลเข้าสู่ระบบผู้ดูแล -ข้อมูลการกู้รหัสผ่าน -ข้อมูลการตั้งรหัสผ่าน -ข้อมูลผู้ดูแลที่แก้ไข -ข้อมูลอนุมัติการยืนยันตัวตน -ข้อมูลอนุมัติการถอนเงิน -ข้อมูลสถานะไกด์ที่แก้ไข

ตารางที่ 3.1 External Entity Description (ต่อ)

Name	Description	Input Data Flow	Output Data Flow
		-รายการข้อมูลรายงาน ความประพฤติ -ผลการแก้ไขสถานะไกด์	
ไกด์	ผู้ใช้งาน ประเภทไกด์ เป็นผู้ใช้ที่สมัคร สมาชิกประเภท ไกด์แล้ว	-ผลการเข้าสู่ระบบ -ผลการกู้รหัสผ่าน -ข้อมูลไกด์ -ผลการแก้ไขข้อมูลไกด์ -ข้อมูลสถานที่ -ผลการแก้ไขข้อมูล สถานที่ -รายการข้อมูลสถานที่ -ข้อมูลกิจกรรม -ผลการแก้ไขข้อมูล กิจกรรม -รายการข้อมูลกิจกรรม -ข้อมูลทริป -ข้อมูลรูปภาพ -ผลการแก้ไขข้อมูลทริป	-ข้อมูลเข้าสู่ระบบไกด์ -ข้อมูลการกู้รหัสผ่าน -ข้อมูลการตั้งรหัสผ่าน -ข้อมูลไกด์ที่แก้ไข -ข้อมูลยืนยันตัวตนที่เพิ่ม -ข้อมูลสถานที่ที่เพิ่ม -ข้อมูลสถานที่ที่แก้ไข -ข้อมูลสถานที่ที่ค้นหา -ข้อมูลกิจกรรมที่เพิ่ม -ข้อมูลกิจกรรมที่แก้ไข -ข้อมูลกิจกรรมที่ค้นหา -ข้อมูลกิจกรรมที่ลบ -ข้อมูลทริปที่เพิ่ม -ข้อมูลทริปที่แก้ไข -ข้อมูลรูปภาพที่เพิ่ม -ข้อมูลรูปภาพที่ลบ

### 3.4 Data Store Description and Data Structure

Data Store คือการนำข้อมูลเข้าไปจัดการในฐานข้อมูล โดยมีการแยกออกเป็นเพิ่มข้อมูล

ตารางที่ 3.2 Data Store Description and Data Structure

ID	Data Store Name	Description	Data Structure
D1	ข้อมูลนักท่องเที่ยว	เก็บรายละเอียด ข้อมูลของ นักท่องเที่ยว	อีเมลนักท่องเที่ยว + รหัสผ่าน + บัญชีFacebook + คำนำหน้าชื่อ +บ้านเลขที่ + หมู่ที่ + ตำบล + อำเภอ + จังหวัด +

ตารางที่ 3.2 Data Store Description and Structure Description (ต่อ)

ID	Data Store Name	Description	Data Structure
			รหัสไปรษณีย์ + ชื่อ + นามสกุล + ชื่อเล่น + วันเกิด + เพศ + รูปภาพโปรไฟล์ + พาสปอร์ต
D2	ข้อมูลไกด์	เก็บรายละเอียดข้อมูลของไกด์	อีเมลไกด์ + รหัสผ่าน + รูปภาพโปรไฟล์ + คำนำหน้าชื่อ + ชื่อ + นามสกุล + ชื่อเล่น + บัญชีFacebook + เบอร์โทร + วันเกิด + เพศ + บ้านเลขที่ + หมู่ที่ + ตำบล + อำเภอ + จังหวัด + รหัสไปรษณีย์ + พาสปอร์ต + หมายเลขบัตรประชาชน + รายละเอียดแนะนำตัว + รูปถ่ายหน้าคู่บัตร + บัญชีธนาคาร + สถานะการยืนยันตัวตน + สถานะไกด์ + วันที่อนุมัติ + คะแนน
D3	ข้อมูลผู้ดูแล	เก็บรายละเอียดข้อมูลของผู้ดูแล	อีเมลผู้ดูแล + รหัสผ่าน + รูปภาพโปรไฟล์ + บัญชีFacebook + คำนำหน้าชื่อ + ชื่อ + นามสกุล
D4	ข้อมูลสถานที่	เก็บรายละเอียดข้อมูลของสถานที่	รหัสสถานที่ + ละติจูด + ลองจิจูด + ชื่อสถานที่ + รายละเอียด + ภาพหน้าปก
D5	ข้อมูลทริป	เก็บรายละเอียดข้อมูลของทริป	รหัสทริป + ชื่อทริป + จำนวนนักท่องเที่ยว + จังหวัด + ละติจูด + ลองจิจูด + รายละเอียด + สถานะ + ภาพหน้าปก + ราคา
D6	ข้อมูลการจอง	เก็บรายละเอียดข้อมูลของการจอง	รหัสการจอง + ระยะเวลาจอง + วันไปเที่ยว + จำนวนนักท่องเที่ยว + สัญชาติ + ชื่อ + นามสกุล + เบอร์โทร + สถานะการจอง + รหัสทริป + รหัสนักท่องเที่ยว
D7	ข้อมูลการรายงาน	เก็บรายละเอียดข้อมูลของการรายงาน	รหัสการรายงาน + รหัสนักท่องเที่ยว + รหัสไกด์ + ข้อความ
D8	ข้อมูลรูปภาพ	เก็บรายละเอียดข้อมูลของรูปภาพ	รหัสรูปภาพ + รหัสทริป + รูปภาพ

ตารางที่ 3.2 Data Store Description and Structure Description (ต่อ)

ID	Data Store Name	Description	Data Structure
D9	ข้อมูลรายงาน ธนาคาร	เก็บรายละเอียด ข้อมูลรายงาน การเงิน	รหัสรายงานธนาคาร + อีเมลไกด์ + จำนวนเงิน + วันเวลา + รายละเอียด
D10	ข้อมูลกิจกรรม	เก็บรายละเอียด ข้อมูลวันเวลาใน การจัดกิจกรรม	รหัสกิจกรรม + รหัสสถานที่ + รหัสทริป + วันเวลา
D11	ข้อมูลการแสดงความ ความคิดเห็น	เก็บข้อการแสดง ความคิดเห็น และ คะแนน ของทริป	รหัสแสดงความคิดเห็น + รหัสทริป + ข้อความ + รูปภาพ + คะแนน

## 3.5 Data Flow Description and Data Structure

ตารางที่ 3.3 Data Flow Description and Data Structure

Name	Description	Source	Destination	Data Structure
ข้อมูลสมัคร สมาชิก	รายละเอียด ข้อมูลการ สมัครสมาชิก	ผู้ใช้ทั่วไป	Process 1.0 สมัครสมาชิก	[อีเมล + รหัสผ่าน + ชื่อ + นามสกุล   บัญชี Facebook]
		Process 1.0 สมัครสมาชิก	D1 ข้อมูล นักท่องเที่ยว	
		ผู้ใช้ทั่วไป	Process 1.0 สมัครสมาชิก	
		Process 1.0 สมัครสมาชิก	D2 ข้อมูลไกด์	
ข้อมูล นักท่องเที่ยว	เก็บข้อมูล นักท่องเที่ยว	D1 ข้อมูล นักท่องเที่ยว	Process 1.0 สมัครสมาชิก	{อีเมลนักท่องเที่ยว + รหัสผ่าน + บัญชีFacebook + คำนำหน้าชื่อ + บ้านเลขที่ + หมู่ที่ +
		Process 1.0 สมัครสมาชิก	ผู้ใช้ทั่วไป	

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		D1 ข้อมูล นักท่องเที่ยว	Process 2.0 เข้าสู่ระบบ	ตำบล + อำเภอ + จังหวัด + รหัสไปรษณีย์ + ชื่อ + นามสกุล + ชื่อ เล่น + วันเกิด + เพศ + รูปภาพโปรไฟล์ + พาสปอร์ต}
			Process 3.1 กู้รหัสผ่าน	
			Process 3.2 ตั้งรหัสผ่านใหม่	
		D1 ข้อมูล นักท่องเที่ยว	Process 3.3 แก้ไขข้อมูลส่วนตัว	
		Process 3.3 แก้ไขข้อมูลส่วนตัว	นักท่องเที่ยว	
		D1 ข้อมูล นักท่องเที่ยว	Process 8.4 จองทริป	
		Process 8.4 จองทริป	นักท่องเที่ยว	
		D1 ข้อมูล นักท่องเที่ยว	Process 10.0 ให้คะแนน	
			Process 11.1 เพิ่มข้อมูลรายงาน ความประพฤติ	
			Process 12.0 แสดงความคิดเห็น	
			Process 13.0 การชำระเงิน	
ข้อมูลไกด์	เก็บข้อมูลของ ไกด์	D2 ข้อมูลไกด์	Process 1.0 สมัครสมาชิก	{อีเมลไกด์ + รหัสผ่าน + รูปภาพโปรไฟล์ + คำนำหน้าชื่อ + ชื่อ + นามสกุล + ชื่อเล่น + เบอร์โทร
		Process 1.0 สมัครสมาชิก	ผู้ใช้ทั่วไป	

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		D2 ข้อมูลไกด์	Process 2.0 เข้าสู่ระบบ	+ บัญชีFacebook + วันเกิด + เพศ +
		D2 ข้อมูลไกด์	Process 3.1 กู้รหัสผ่าน	บ้านเลขที่ + หมู่ที่ + ตำบล + อำเภอ +
		D2 ข้อมูลไกด์	Process 3.2 ตั้งรหัสผ่านใหม่	จังหวัด + รหัสไปรษณีย์ +
		D2 ข้อมูลไกด์	Process 3.3 แก้ไขข้อมูลส่วนตัว	พาสปอร์ต +บัตร ประชาชน +
		Process 3.3 แก้ไขข้อมูลส่วนตัว	ไกด์	รายละเอียดแนะนำ ตัว + รูปถ่ายคู่บัตร
		D2 ข้อมูลไกด์	Process 4.0 เพิ่มข้อมูลยืนยัน ตัวตน	+บัญชีธนาคาร + สถานะการยืนยัน ตัวตน + สถานะไกด์
		Process 4.0 เพิ่มข้อมูลยืนยัน ตัวตน	ไกด์	+ วันที่อนุมัติ + คะแนน}
		D2 ข้อมูลไกด์	Process 5.1 อนุมัติการยืนยัน ตัวตนของไกด์	
		Process 5.1 อนุมัติการยืนยัน ตัวตนของไกด์	ผู้ดูแล	
		D2 ข้อมูลไกด์	Process 5.2 อนุมัติการถอนเงิน ของไกด์	
		Process 5.2 อนุมัติการถอนเงิน ของไกด์	ผู้ดูแล	



ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		D2 ข้อมูลไกด์	Process 8.1 สร้างทริป	
			Process 8.4 จองทริป	
			Process 9.1 ค้นหาทริป	
			Process 9.2 ค้นหาไกด์	
		D2 ข้อมูลไกด์	Process 10.0 ให้คะแนน	
		Process 10.0 ให้คะแนน	นักท่องเที่ยว	
		D2 ข้อมูลไกด์	Process 11.1 เพิ่มข้อมูลรายงาน ความประพฤติ	
		Process 11.1 เพิ่มข้อมูลรายงาน ความประพฤติ	นักท่องเที่ยว	
		D2 ข้อมูลไกด์	Process 11.3 เพิ่มข้อมูลรายงาน ความประพฤติ	
		Process 11.3 เพิ่มข้อมูลรายงาน ความประพฤติ	ผู้ดูแล	
ข้อมูลเข้าสู่ระบบของไกด์	ตรวจสอบการเข้าสู่ระบบ	ไกด์	Process 2.0 เข้าสู่ระบบ	[อีเมล + รหัสผ่าน   บัญชี Facebook]

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลเข้าสู่ระบบของนักท่องเที่ยวนักท่องเที่ยว	ตรวจสอบการเข้าสู่ระบบ	นักท่องเที่ยวนักท่องเที่ยว	Process 2.0 เข้าสู่ระบบ	[อีเมล + รหัสผ่าน  บัญชี Facebook]
ข้อมูลเข้าสู่ระบบของผู้ดูแล	ตรวจสอบการเข้าสู่ระบบ	ผู้ดูแล	Process 2.0 เข้าสู่ระบบ	[อีเมล + รหัสผ่าน  บัญชี Facebook]
ผลการเข้าสู่ระบบ	แสดงผลเมื่อเข้าสู่ระบบ	Process 2.0 เข้าสู่ระบบ	ไกด์	แสดงผลเมื่อเข้าสู่ระบบ
			นักท่องเที่ยวนักท่องเที่ยว	
			ผู้ดูแล	
ข้อมูลการกู้รหัสผ่าน	ตรวจสอบข้อมูลอีเมลก่อนการกู้รหัสผ่าน	ผู้ดูแล	Process 3.1 กู้รหัสผ่าน	อีเมล
		นักท่องเที่ยวนักท่องเที่ยว	Process 3.1 กู้รหัสผ่าน	อีเมล
		ไกด์	Process 3.1 กู้รหัสผ่าน	อีเมล
ผลการกู้รหัสผ่าน	รหัสผ่านเก่าที่ได้จากการกู้	Process 3.1 กู้รหัสผ่าน	ผู้ดูแล	รหัสผ่าน
			นักท่องเที่ยวนักท่องเที่ยว	
			ไกด์	
ข้อมูลการตั้งรหัสผ่าน	รายละเอียดการตั้งรหัสผ่านใหม่	ผู้ดูแล	Process 3.2 ตั้งรหัสผ่านใหม่	รหัสผ่าน
		Process 3.2 ตั้งรหัสผ่านใหม่	D3 ข้อมูลผู้ดูแล	
		นักท่องเที่ยวนักท่องเที่ยว	Process 3.2 ตั้งรหัสผ่านใหม่	
		Process 3.2 ตั้งรหัสผ่านใหม่	D1 ข้อมูลนักท่องเที่ยวนักท่องเที่ยว	
		ไกด์	Process 3.2 ตั้งรหัสผ่านใหม่	

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 3.2 ตั้งรหัสผ่านใหม่	D2 ข้อมูลไกด์	
ข้อมูลผู้ดูแล	เก็บข้อมูล ข้อมูลผู้ดูแล	D3 ข้อมูลผู้ดูแล	Process 2.0 เข้าสู่ระบบ	อีเมลผู้ดูแล + รหัสผ่าน + รูปภาพ
			Process 3.2 ตั้งรหัสผ่านใหม่	โปรไฟล์ + บัญชีFacebook +
		D3 ข้อมูลผู้ดูแล	Process 3.3 แก้ไขข้อมูลส่วนตัว	คำแนะนำชื่อ + ชื่อ + นามสกุล
ข้อมูลผู้ดูแล ที่แก้ไข	รายละเอียด ข้อมูลผู้ดูแลที่ ต้องการแก้ไข	ผู้ดูแล	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสผ่าน + รูปภาพโปรไฟล์ +
		Process 3.3 แก้ไขข้อมูลส่วนตัว	D3 ข้อมูลผู้ดูแล	คำแนะนำชื่อ + ชื่อ + นามสกุล
ผลการแก้ไข ข้อมูลผู้ดูแล	ข้อมูลผู้ดูแลที่ แก้ไขแล้ว	D3 ข้อมูลผู้ดูแล	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสผ่าน + รูปภาพโปรไฟล์ +
		Process 3.3 แก้ไขข้อมูลส่วนตัว	ผู้ดูแล	คำแนะนำชื่อ + ชื่อ + นามสกุล
ข้อมูล นักท่องเที่ยว ที่แก้ไข	รายละเอียด ข้อมูล นักท่องเที่ยวที่ ต้องการแก้ไข	นักท่องเที่ยว	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสผ่าน + คำแนะนำชื่อ +
		Process 3.3 แก้ไขข้อมูลส่วนตัว	D1 ข้อมูล นักท่องเที่ยว	ชื่อ + นามสกุล + บ้านเลขที่ + หมู่ที่ + ตำบล + อำเภอ + จังหวัด + รหัสไปรษณีย์ + ชื่อเล่น + วันเกิด + เพศ + รูปภาพโปร+ ไฟล์ + พาสปอร์ต

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลไกด์ที่ แก้ไข	รายละเอียด ข้อมูลไกด์ที่ ต้องการแก้ไข	ไกด์	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสผ่าน + รูปภาพโปรไฟล์ +
		Process 3.3 แก้ไขข้อมูลส่วนตัว	D2 ข้อมูลไกด์	คำนำหน้าชื่อ + ชื่อ + นามสกุล + ชื่อเล่น + เบอร์โทร + วันเกิด + เพศ + บ้านเลขที่ + หมู่ที่ + ตำบล + อำเภอ + จังหวัด + รหัสไปรษณีย์ + พาสปอร์ต
ผลการแก้ไข ข้อมูล นักท่องเที่ยว	ข้อมูล นักท่องเที่ยวที่ แก้ไขแล้ว	D1 ข้อมูล นักท่องเที่ยว	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสผ่าน + คำนำหน้าชื่อ +
		Process 3.3 แก้ไขข้อมูลส่วนตัว	นักท่องเที่ยว	ชื่อ + นามสกุล + บ้านเลขที่ + หมู่ที่ + ตำบล + อำเภอ + จังหวัด + รหัสไปรษณีย์ + ชื่อเล่น + วันเกิด + รูปภาพโปรไฟล์ + พาสปอร์ต + เพศ
ข้อมูลยืนยัน ตัวตนที่เพิ่ม	รายละเอียด ข้อมูลยืนยัน ตัวตนของไกด์ ที่ต้องเพิ่ม	ไกด์	Process 4.0 เพิ่มข้อมูลยืนยัน ตัวตน	บัตรประชาชน + รายละเอียดแนะนำ ตัว + รูปถ่ายคู่บัตร
		Process 4.0 เพิ่มข้อมูลยืนยัน ตัวตน	D2 ข้อมูลไกด์	+ บัญชีธนาคาร

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
สถานะการ เพิ่มข้อมูล ยืนยัน	สถานะการ เพิ่มข้อมูล	Process 4.0 เพิ่มข้อมูลยืนยัน ตัวตน	ผู้ดูแล	ข้อมูลยืนยันตัวตนถูก เพิ่มแล้ว
ข้อมูลอนุมัติ การยืนยัน ตัวตน	รายละเอียด อนุมัติการ ยืนยันตัวตน	ผู้ดูแล	Process 5.1 อนุมัติการยืนยัน ตัวตน	สถานะไกด์ + วันที่อนุมัติ
		Process 5.1 อนุมัติการยืนยัน ตัวตน	D2 ข้อมูลไกด์	
ข้อมูลอนุมัติ การถอนเงิน	รายละเอียด อนุมัติการ ถอนเงิน	ผู้ดูแล	Process 5.2 อนุมัติการถอนเงิน ของไกด์	รหัสรายงานธนาคาร + รหัสไกด์ + จำนวนเงิน + เวลา + รายละเอียด
		Process 5.2 อนุมัติการถอนเงิน ของไกด์	D9 ข้อมูลรายงาน ธนาคาร	
ข้อมูล สถานที่ที่ เพิ่ม	ข้อมูล รายละเอียด สถานที่ที่เพิ่ม	ไกด์	Process 6.1 เพิ่มสถานที่	ละติจูด + ลองจิจูด + ชื่อสถานที่ + รายละเอียด + ภาพหน้าปก
		Process 6.1 เพิ่มสถานที่	D4 ข้อมูลสถานที่	
ข้อมูล สถานที่ที่ แก้ไข	รายละเอียด ข้อมูลสถานที่ ที่ต้องการ แก้ไข	ไกด์	Process 6.2 แก้ไขสถานที่	ละติจูด + ลองจิจูด + ชื่อสถานที่ + รายละเอียด + ภาพหน้าปก
		Process 6.2 แก้ไขสถานที่	D4 ข้อมูลสถานที่	
ข้อมูล สถานที่	รายละเอียด ข้อมูลสถานที่	D4 ข้อมูลสถานที่	Process 6.2 แก้ไขสถานที่	{รหัสสถานที่ + ละติจูด + ลองจิจูด

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 6.2 แก้ไขสถานที่	ไกด์	ชื่อสถานที่ + รายละเอียด + ภาพหน้าปก}
		D4 ข้อมูลสถานที่	Process 6.3 ค้นหาสถานที่	
			Process 7.1 สร้างกิจกรรม	
			Process 7.2 แก้ไขกิจกรรม	
			Process 7.4 ค้นหากิจกรรม	
			Process 8.1 สร้างทริป	
			Process 8.2 แก้ไขทริป	
			Process 9.1 ค้นหาทริป	
ผลการแก้ไข ข้อมูล สถานที่	ข้อมูลสถานที่ ที่แก้ไขแล้ว	D4 ข้อมูลสถานที่	Process 6.2 แก้ไขสถานที่	ละติจูด + ลองจิจูด + ชื่อสถานที่ + รายละเอียด + ภาพหน้าปก
		Process 6.2 แก้ไขสถานที่	ไกด์	
ข้อมูล สถานที่ที่ ค้นหา	รายละเอียด ข้อมูลสถานที่ ที่ต้องการ ค้นหา	ไกด์	Process 6.3 ค้นหาสถานที่	ชื่อสถานที่
รายการ ข้อมูล สถานที่	รายการของ สถานที่ที่ ค้นหา	D4 ข้อมูลสถานที่	Process 6.3 ค้นหาสถานที่	ชื่อสถานที่
		Process 6.3 ค้นหาสถานที่	ไกด์	

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูล กิจกรรมที่ เพิ่ม	ข้อมูล รายละเอียด กิจกรรมที่เพิ่ม	ไกด์	Process 7.1 สร้างกิจกรรม	วันเวลา + รหัส สถานที่ + รหัสทริป
		Process 7.1 สร้างกิจกรรม	D10 ข้อมูลกิจกรรม	
ข้อมูล กิจกรรม	เก็บข้อมูล กิจกรรม	D10 ข้อมูล กิจกรรม	Process 7.2 แก้ไขกิจกรรม	{รหัสกิจกรรม + รหัสสถานที่ + รหัสทริป + วันเวลา}
		Process 7.2 แก้ไขกิจกรรม	ไกด์	
		D10 ข้อมูล กิจกรรม	Process 7.3 ลบกิจกรรม	
		Process 7.3 ลบกิจกรรม	ไกด์	
		D10 ข้อมูล กิจกรรม	Process 7.4 ค้นหากิจกรรม	
			Process 7.4 ค้นหาทริป	
ข้อมูล กิจกรรมที่ แก้ไข	รายละเอียด ข้อมูลกิจกรรม ที่แก้ไข	ไกด์	Process 7.2 แก้ไขกิจกรรม	รหัสกิจกรรม + รหัสสถานที่ + รหัสทริป + วันเวลา
		Process 7.2 แก้ไขกิจกรรม	D10 ข้อมูลกิจกรรม	
ผลการแก้ไข ข้อมูล กิจกรรม	ข้อมูลกิจกรรม ที่แก้ไขแล้ว	D10 ข้อมูล กิจกรรม	Process 7.2 แก้ไขกิจกรรม	รหัสกิจกรรม + รหัสสถานที่ + รหัสทริป + วันเวลา
		Process 7.2 แก้ไขกิจกรรม	ไกด์	

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูล กิจกรรมที่ ลบ	ข้อมูลกิจกรรม ที่ต้องการลบ	ไกด์	Process 7.3 ลบกิจกรรม	รหัสกิจกรรม + รหัสสถานที่ +
		Process 7.3 ลบกิจกรรม	D10 ข้อมูลกิจกรรม	รหัสทริป + + วันเวลา
ข้อมูล กิจกรรมที่ ค้นหา	รายละเอียด ข้อมูลกิจกรรม ที่ต้องการ ค้นหา	ไกด์	Process 7.4 ค้นหากิจกรรม	รหัสทริป
รายการ ข้อมูล กิจกรรม	รายการของ กิจกรรมที่ได้ ค้นหา	Process 7.4 ค้นหากิจกรรม	ไกด์	รหัสกิจกรรม + รหัสสถานที่ + รหัสทริป + + วันเวลา
ข้อมูลทริปที่ เพิ่ม	ข้อมูล รายละเอียด ทริปที่เพิ่ม	ไกด์	Process 8.1 สร้างทริป	ชื่อทริป + จำนวน นักท่องเที่ยว +
		Process 8.1 สร้างทริป	D5 ข้อมูลทริป	จังหวัด + ละติจูด + ลองจิจูด + สถานะ + ราคา+รายละเอียด + ภาพหน้าปก
ข้อมูล รูปภาพ ที่เพิ่ม	ข้อมูล รายละเอียด รูปภาพที่เพิ่ม	ไกด์	Process 8.1 สร้างทริป	รหัสทริป+ รูปภาพ
		Process 8.1 สร้างทริป	D8 ข้อมูลรูปภาพ	
ข้อมูลทริปที่ แก้ไข	รายละเอียด ข้อมูลทริปที่ ต้องการแก้ไข	ไกด์	Process 8.2 แก้ไขทริป	ชื่อทริป + จำนวน นักท่องเที่ยว +
		Process 8.2 แก้ไขทริป	D5 ข้อมูลทริป	จังหวัด + ละติจูด + ลองจิจูด +ราคา + สถานะรายละเอียด +ภาพหน้าปก



ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ผลการแก้ไข ข้อมูลทริป	ข้อมูลทริปที่ แก้ไขแล้ว	D5 ข้อมูลทริป	Process 8.2 แก้ไขทริป	ชื่อทริป + จำนวน นักท่องเที่ยว + จังหวัด + ละติจูด + ลองจิจูด + สถานะ + รายละเอียด + ภาพหน้าปก + ราคา
		Process 8.2 แก้ไขทริป	ไกด์	
ข้อมูล รูปภาพที่ลบ	ข้อมูลรูปภาพ ที่ต้องการลบ	ไกด์	Process 8.3 ลบรูปภาพทริป	รหัสรูปภาพ
		Process 8.3 ลบรูปภาพทริป	D8 ข้อมูลรูปภาพ	
ข้อมูล รูปภาพ	เก็บข้อมูล รูปภาพ	D8 ข้อมูลรูปภาพ	Process 8.3 ลบรูปภาพทริป	{รหัสรูปภาพ + รหัสทริป+ รูปภาพ}
		Process 8.3 ลบรูปภาพทริป	ไกด์	
		D8 ข้อมูลรูปภาพ	Process 9.1 ค้นหาทริป	
ข้อมูลทริป	รายละเอียด ข้อมูลทริป	D5 ข้อมูลทริป	Process 7.1 สร้างกิจกรรม	{รหัสทริป + ชื่อทริป + จำนวนนักท่องเที่ยว + จังหวัด + ละติจูด + ลองจิจูด + รายละเอียด + สถานะ + ภาพหน้าปก + ราคา}
		D5 ข้อมูลทริป	Process 7.4 ค้นหากิจกรรม	
		D5 ข้อมูลทริป	Process 8.2 แก้ไขทริป	
		Process 8.2 แก้ไขทริป	ไกด์	
		D5 ข้อมูลทริป	Process 8.4 จองทริป	
		Process 8.4 จองทริป	นักท่องเที่ยว	

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		D5 ข้อมูลทริป	Process 9.1 ค้นหาทริป	
		Process 9.1 ค้นหาทริป	นักท่องเที่ยว	
		D5 ข้อมูลทริป	Process 9.1 ค้นหาทริป	
		Process 9.1 ค้นหาทริป	ผู้ใช้ทั่วไป	
		D5 ข้อมูลทริป	Process 10.0 ค้นหาทริป	
		Process 10.0 ค้นหาทริป	นักท่องเที่ยว	
		D5 ข้อมูลทริป	Process 12.0 แสดงความคิดเห็น	
		Process 12.0 แสดงความคิดเห็น	นักท่องเที่ยว	
ข้อมูลการจอง	รายละเอียด ข้อมูลการจอง	D6 ข้อมูลการจอง	Process 8.4 จองทริป	{รหัสการจอง + วัน เวลาจอง + วันไป เที่ยว + เบอร์โทร + จำนวนนักท่องเที่ยว + ชื่อ + สัญชาติ + นามสกุล + รหัสทริป + สถานะการจอง + รหัสนักท่องเที่ยว}
		Process 8.4 จองทริป	นักท่องเที่ยว	
		D6 ข้อมูลการจอง	Process 13.0 จองทริป	
		Process 13.0 จองทริป	นักท่องเที่ยว	
รับข้อมูล จองทริป	รายละเอียด ข้อมูลการจอง ทริป	นักท่องเที่ยว	Process 8.4 จองทริป	รหัสนักท่องเที่ยว + วันเวลาจอง + วันไปเที่ยว + ชื่อ + + นามสกุล + สัญชาติ

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 8.4 จองทริป	D6 ข้อมูลการจอง	+จำนวน นักท่องเที่ยว
ข้อความแจ้ง เตือนการ จองทริป	ข้อความแจ้ง สถานะการ จองทริป	Process 8.4 จองทริป	นักท่องเที่ยว	ข้อความแจ้งเตือน สถานะการจอง
ข้อมูลทริปที่ ต้องการ ค้นหา	รายละเอียด ข้อมูลทริปที่ ต้องการค้นหา	นักท่องเที่ยว	Process 9.1 ค้นหาทริป	[ชื่อทริป   จังหวัด]
		ผู้ใช้ทั่วไป	Process 9.1 ค้นหาทริป	
รายการ ข้อมูลทริป	รายการของ ทริปที่ค้นหา	Process 9.1 ค้นหาทริป	นักท่องเที่ยว	{ชื่อทริป + รูปภาพ หน้าปก + ราคา + คะแนน}
		Process 9.1 ค้นหาทริป	ผู้ใช้ทั่วไป	
ข้อมูลโค้ดที่ ต้องการ ค้นหา	รายละเอียด ข้อมูลโค้ดที่ ต้องการค้นหา	นักท่องเที่ยว	Process 9.2 ค้นหาโค้ด	ชื่อ
		ผู้ใช้ทั่วไป	Process 9.2 ค้นหาโค้ด	
รายการ ข้อมูลโค้ด	รายการของ ทริปที่ค้นหา	Process 9.2 ค้นหาโค้ด	นักท่องเที่ยว	{รูปภาพโปรไฟล์ + คำนำหน้าชื่อ + ชื่อ + นามสกุล + ชื่อเล่น + เบอร์โทร}
		Process 9.2 ค้นหาโค้ด	ผู้ใช้ทั่วไป	
ข้อมูลการให้ คะแนนโค้ด ที่เพิ่ม	รายละเอียด ข้อมูลการให้ คะแนนโค้ดที่ เพิ่ม	นักท่องเที่ยว	Process 10.0 ให้คะแนน	อีเมลโค้ด + คะแนน
		Process 10.0 ให้คะแนน	D2 ข้อมูลโค้ด	
ข้อมูลการให้ คะแนนทริป ที่เพิ่ม	รายละเอียด ข้อมูลการให้ คะแนนทริป	นักท่องเที่ยว	Process 10.0 ให้คะแนน	รหัสแสดงความ คิดเห็น + รหัสทริป + คะแนน

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

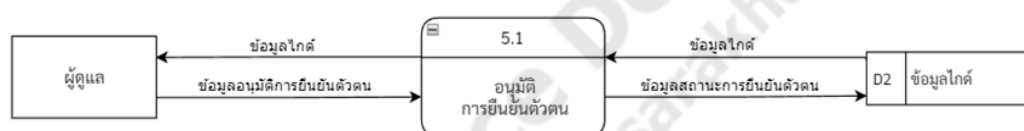
Name	Description	Source	Destination	Data Structure
		Process 10.0 ให้คะแนน	D11 ข้อมูลการแสดง ความคิดเห็น	
ข้อมูล รายงาน ความ ประพจน์ ที่เพิ่ม	รายละเอียด ข้อมูลรายงาน ความประพจน์ ที่เพิ่ม	นักท่องเที่ยวน	Process 11.1 เพิ่มข้อมูลรายงาน ความประพจน์	รหัสนักท่องเที่ยว + รหัสโค้ด + ข้อความ
		Process 11.1 เพิ่มข้อมูลรายงาน ความประพจน์	D7 ข้อมูลการรายงาน	
รายการ ข้อมูล รายงาน ความ ประพจน์	รายละเอียด ข้อมูลของการ รายงานความ ประพจน์	D7 ข้อมูลการ รายงาน	Process 11.2 ค้นหาข้อมูลรายงาน ความประพจน์	รหัสนักท่องเที่ยว + รหัสโค้ด + ข้อความ
		Process 11.2 ค้นหาข้อมูล รายงานความ ประพจน์	ผู้ดูแล	
ข้อมูล สถานะโค้ด ที่แก้ไข	ผู้ดูแล พิจารณา ข้อความ รายงานและ แก้ไขสถานะ ของโค้ด	ผู้ดูแล	Process 11.3 แก้ไขสถานะโค้ด	สถานะโค้ด
		Process 11.3 แก้ไขสถานะโค้ด	D2 ข้อมูลโค้ด	
ผลการแก้ไข สถานะโค้ด	ข้อมูลของโค้ด ที่แสดงถึงการ ถูกระงับบัญชี	D2 ข้อมูลโค้ด	Process 11.3 แก้ไขสถานะโค้ด	สถานะโค้ด
		Process 11.3 แก้ไขสถานะโค้ด	ผู้ดูแล	
ข้อมูลที่ แสดง ความคิดเห็น	รายละเอียด ข้อมูลที่แสดง ความคิดเห็น	นักท่องเที่ยวน	Process 12.0 แสดงความคิดเห็น	รหัสทวีป + ข้อความ + รูปภาพ

ตารางที่ 3.3 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 12.0 แสดงความคิดเห็น	D11 ข้อมูลการแสดง ความคิดเห็น	
ข้อมูลชำระ เงิน	รายละเอียด ข้อมูลการ ชำระเงิน	นักท่องเที่ยวน	Process 13.0 การชำระเงิน	อีเมลไคด์ + จำนวนเงิน +
		Process 13.0 การชำระเงิน	D9 ข้อมูลรายงาน ธนาคาร	วันเวลา + รายละเอียด

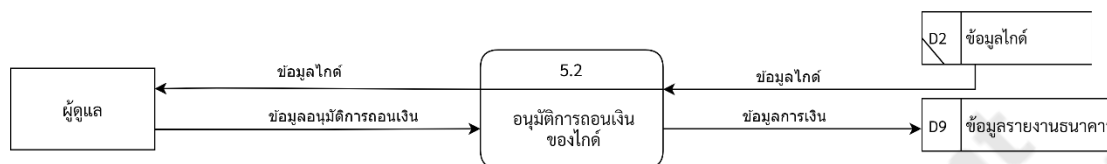
## 3.6 คำอธิบายการประมวลผล (Process Description)

## 3.6.1 Process 5.1 อนุมัติการยืนยันตัวตน



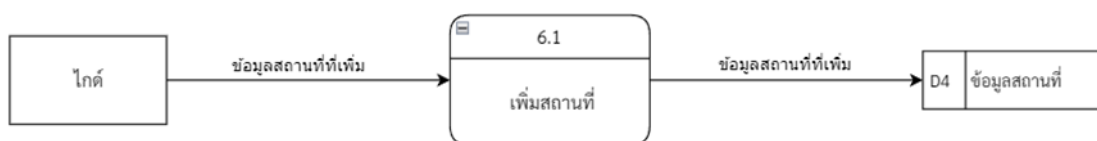
ID	5.1
Name	อนุมัติการยืนยันตัวตน
Description	ผู้ดูแลทำการตรวจสอบข้อมูลและอนุมัติการยืนยันตัวตนของไคด์
Input data flow	-ข้อมูลไคด์ -ข้อมูลอนุมัติการยืนยันตัวตน
Output data flow	-ข้อมูลไคด์ -ข้อมูลสถานะการยืนยันตัวตน
Process description	เริ่มต้น <ol style="list-style-type: none"> <li>1. ดึงข้อมูลไคด์</li> <li>2. ตรวจสอบข้อมูลไคด์ที่ดึงเข้ามา <ol style="list-style-type: none"> <li>2.1 ถ้าตรวจสอบข้อมูลมีความถูกต้อง จะส่งสถานะผ่านเข้าสู่ระบบ</li> <li>2.2 ถ้าตรวจสอบข้อมูลไม่ถูกต้อง จะส่งสถานะไม่ผ่านเข้าสู่ระบบ</li> </ol> </li> </ol> จบการทำงาน

## 3.6.2 Process 5.2 อนุมัติการถอนเงิน



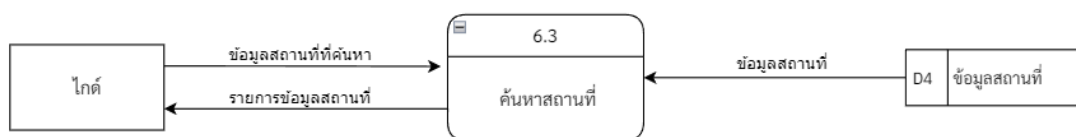
ID	5.2
Name	อนุมัติการถอนเงิน
Description	ผู้ดูแลทำการตรวจสอบข้อมูลและอนุมัติการถอนเงินของโค้ด
Input data flow	- ข้อมูลโค้ด - ข้อมูลอนุมัติการถอนเงิน
Output data flow	- ข้อมูลโค้ด - ข้อมูลการเงิน
Process description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1. ดึงข้อมูลโค้ด</li> <li>2. ผู้ดูแลตรวจสอบข้อมูลโค้ดที่รับเข้ามา             <ol style="list-style-type: none"> <li>2.1 ถ้าข้อมูลโค้ดที่ร้องขอเข้ามาถอนเงินมีความถูกต้องครบถ้วน จะส่งข้อมูลอนุมัติการถอน”ผ่านการอนุมัติ” กลับเข้าสู่ระบบ</li> <li>2.2 ถ้าข้อมูลโค้ดที่ร้องขอเข้ามาถอนเงินไม่ถูกต้องและครบถ้วน จะส่งข้อมูลอนุมัติการถอน “ไม่ผ่านการอนุมัติ” กลับเข้าสู่ระบบ</li> </ol> </li> <li>3. หลังจากอนุมัติแล้ว ข้อมูลการเงินจะถูกนำเข้าไปเก็บที่ข้อมูลรายงานธนาคาร</li> </ol> <p>จบการทำงาน</p>

## 3.6.3 Process 6.1 เพิ่มสถานที่



ID	6.1
Name	เพิ่มสถานที่
Description	เพิ่มข้อมูลสถานที่เพื่อสร้างทริป
Input data flow	- ข้อมูลสถานที่ที่เพิ่ม
Output data flow	- ข้อมูลสถานที่ที่เพิ่ม
Process description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1. เพิ่มข้อมูลสถานที่เข้าสู่ระบบ</li> <li>2. ข้อมูลสถานที่ที่เพิ่มเข้ามา จะถูกนำไปจัดเก็บลงแฟ้มข้อมูลของข้อมูลสถานที่</li> </ol> <p>จบการทำงาน</p>

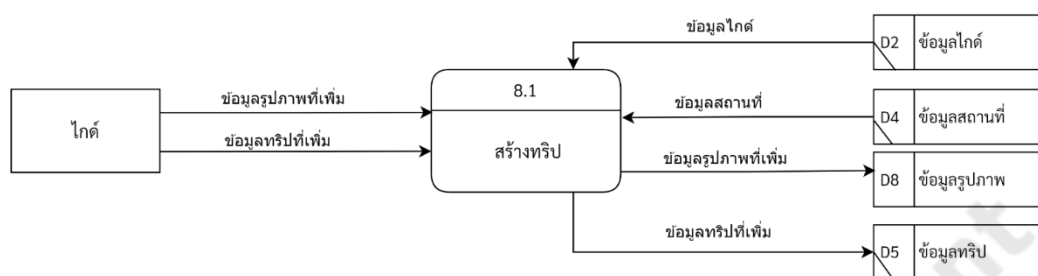
## 3.6.4 Process 6.3 ค้นหาสถานที่



ID	6.3
Name	ค้นหาสถานที่
Description	ค้นหาสถานที่
Input data flow	- ข้อมูลสถานที่ที่ค้นหา - ข้อมูลสถานที่
Output data flow	- รายการข้อมูลสถานที่
Process description	เริ่มต้น 1.รับข้อมูลที่ต้องการค้นหาสถานที่ 2.ค้นหาข้อมูลที่ต้องการค้นหาสถานที่ที่รับเข้ามาจากแฟ้มข้อมูลสถานที่ 3.ตรวจสอบ 3.1 ถ้ามีชื่อสถานที่ให้แสดงรายการข้อมูลของสถานที่ 3.2 ถ้าไม่มีชื่อของสถานที่ให้แสดงข้อความ “ไม่พบสถานที่ “ จบการทำงาน

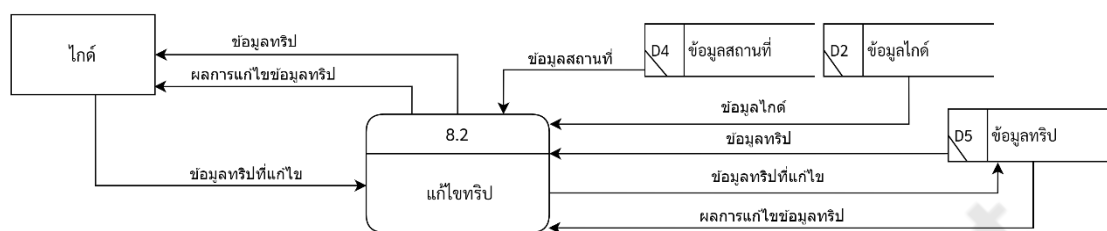


## 3.6.5 Process 8.1 สร้างทริป



ID	7.1
Name	สร้างทริป
Description	ไกด์เพิ่มข้อมูลในการสร้างทริป
Input data flow	<ul style="list-style-type: none"> <li>- ข้อมูลทริปที่เพิ่ม</li> <li>- ข้อมูลรูปภาพที่เพิ่ม</li> <li>- ข้อมูลไกด์</li> <li>- ข้อมูลสถานที่</li> </ul>
Output data flow	<ul style="list-style-type: none"> <li>- ข้อมูลรูปภาพที่เพิ่ม</li> <li>- ข้อมูลทริปที่เพิ่ม</li> </ul>
Process description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1. รับข้อมูลทริปและข้อมูลรูปภาพเข้ามา</li> <li>2. ข้อมูลทริปและข้อมูลรูปภาพ จะถูกนำไปจัดเก็บลงแฟ้มข้อมูลของข้อมูลทริปและข้อมูลรูปภาพ</li> </ol> <p>จบการทำงาน</p>

## 3.6.6 Process 8.2 แก้ไขทริป

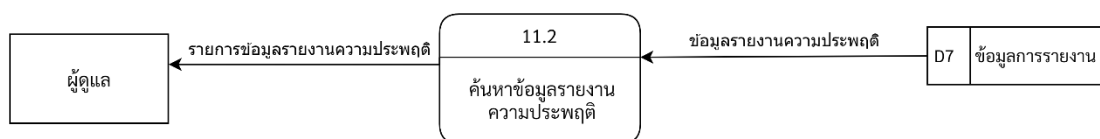


ID	8.2
Name	แก้ไขทริป
Description	โกด์ทำการแก้ไขข้อมูลของทริป
Input data flow	<ul style="list-style-type: none"> <li>- ข้อมูลทริปที่แก้ไข</li> <li>- ข้อมูลสถานที่</li> <li>- ข้อมูลโกด์</li> <li>- ข้อมูลลทริป</li> <li>- ผลการแก้ไขข้อมูลทริป</li> </ul>
Output data flow	<ul style="list-style-type: none"> <li>- ข้อมูลทริป</li> <li>- ผลการแก้ไขข้อมูลทริป</li> <li>- ข้อมูลทริปที่แก้ไข</li> </ul>
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1. ดึงข้อมูลทริป,ข้อมูลสถานที่ที่ข้อมูลโกด์ และข้อมูลทริปเข้ามา</li> <li>2. ตรวจสอบ             <ol style="list-style-type: none"> <li>2.1 ถ้าข้อมูลโกด์ตรงกันกับข้อมูลโกด์ที่ดึงมา จะสามารถแก้ไขข้อมูลทริปได้</li> <li>2.2 ถ้าข้อมูลโกด์ไม่ตรงกันกับข้อมูลโกด์ที่ดึงมา จะไม่สามารถแก้ไขข้อมูลทริปได้</li> </ol> </li> <li>3. เพิ่มข้อมูลทริปที่ต้องการแก้ไข เข้ามาเก็บที่เพิ่มข้อมูล             <ol style="list-style-type: none"> <li>3.1 ถ้าเพิ่มข้อมูลถูกต้อง และครบถ้วนให้แสดงข้อมูลทริปและข้อความ "แก้ไขข้อมูลเรียบร้อยแล้ว"</li> </ol> </li> </ol>



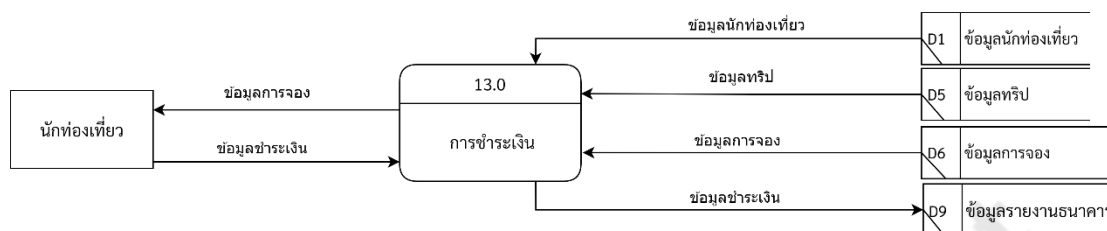


## 3.6.9 Process 11.2 ค้นหาข้อมูลรายงานความประพฤติ



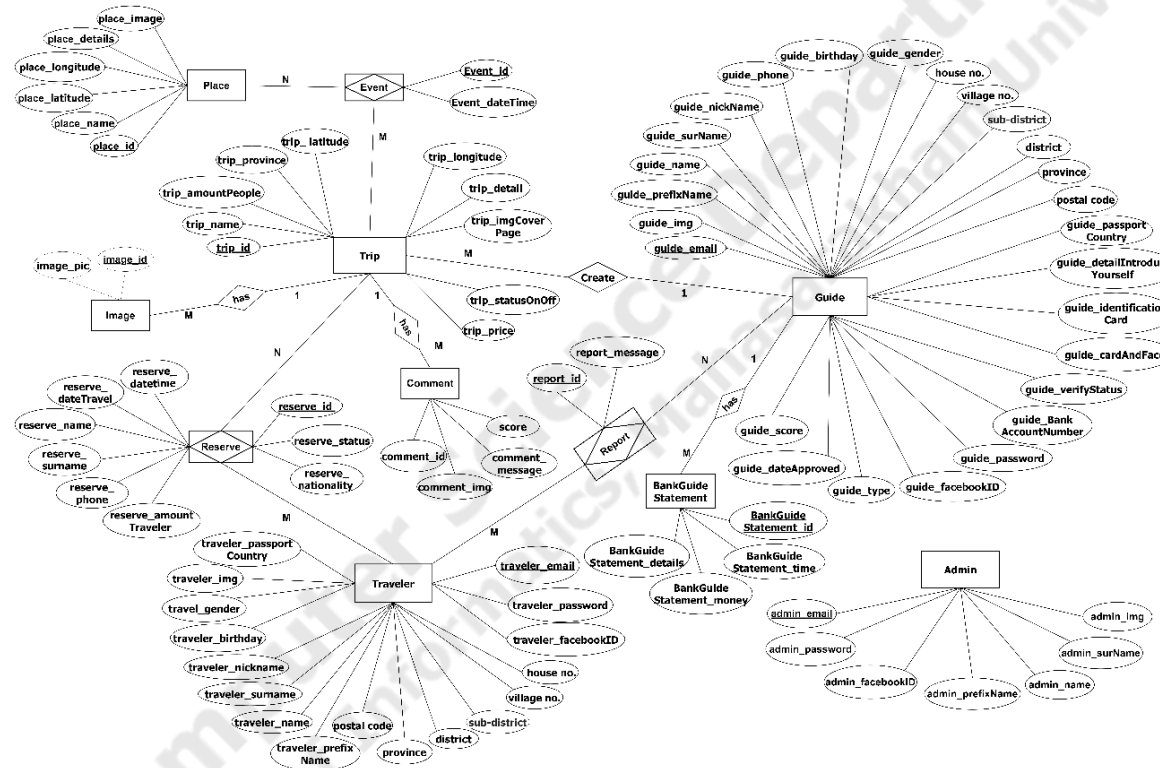
ID	11.2
Name	ค้นหาข้อมูลรายงานความประพฤติ
Description	ผู้ดูแลทำการรับข้อมูลโดยอัตโนมัติเมื่อนักท่องเที่ยวรายงานเข้ามาและตรวจสอบข้อมูลการรายงานจากนักท่องเที่ยวเพื่อพิจารณาลงโทษไกด์
Input data flow	- ข้อมูลรายงานความประพฤติ
Output data flow	- รายการข้อมูลรายงานความประพฤติ
Process description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1.ดึงข้อมูลการรายงานความประพฤติของไกด์เข้ามา</li> <li>2.ตรวจสอบข้อมูลที่ผู้นักท่องเที่ยวรายงานเข้ามา             <ol style="list-style-type: none"> <li>2.1 ถ้าตรวจสอบแล้วพบว่าไกด์มีความผิด ให้ลงโทษโดยการเปลี่ยนสถานะของไกด์เป็นถูกแบน</li> <li>2.2 ถ้าตรวจสอบแล้วพบว่าไกด์ไม่มีความผิด ผู้ดูแลระบบไม่ต้องทำอะไร</li> </ol> </li> </ol> <p>จบการทำงาน</p>

## 3.6.10 Process 13.0 การชำระเงิน



ID	13.0
Name	การชำระเงิน
Description	การชำระเงิน
Input data flow	<ul style="list-style-type: none"> <li>- ข้อมูลการจอง</li> <li>- ข้อมูลนักท่องเที่ยว</li> <li>- ข้อมูลทริป</li> <li>- ข้อมูลชำระเงิน</li> </ul>
Output data flow	<ul style="list-style-type: none"> <li>- ข้อมูลการจอง</li> <li>- ข้อมูลการเงิน</li> </ul>
Process description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> <li>1.ดึงข้อมูลการจองเข้ามา</li> <li>2.เพิ่มข้อมูลชำระเงินเข้ามาเก็บที่เพิ่มข้อมูลรายงานธนาคาร <ol style="list-style-type: none"> <li>2.1 ถ้าเพิ่มข้อมูลเงินมัดจำครบถ้วน จะสามารถทำขั้นตอนต่อไปได้</li> <li>2.2 ถ้าเพิ่มข้อมูลเงินมัดจำยังไม่ครบ จะไม่สามารถข้ามทำขั้นตอนต่อไปได้</li> </ol> </li> </ol> <p>จบการทำงาน</p>

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



ภาพประกอบที่ 3.5 Entity Relationship Diagram (ER-Diagram)

Entity Relationship Diagram แสดงการออกแบบฐานข้อมูล ดังภาพประกอบที่ 3.5

### 3.8 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ข้อมูลไกด์

id	Column	Type	Description	Example Data	Constraints
1	guide_email	Varchar (100)	อีเมลของ	62011212064@msu.ac.th	PK
2	guide_facebookid	Varchar (20)	รหัสบัญชี facebook	112514381367215	Null
3	guide_password	Varchar (20)	รหัสผ่านของบัญชีเข้าสู่ระบบ	123456	Not Null
4.	guide_gender	Varchar (6)	เพศ	ชาย	Null
5	guide_prefix	Varchar (6)	คำนำหน้าชื่อ	นาย	Null
6	guide_name	Varchar (20)	ชื่อจริง	ภูมินทร์	Null
7	guide_surname	Varchar (20)	นามสกุล	โพธิ์สุวรรณ	Null
8	guide_nickname	Varchar (10)	ชื่อเล่น	โอเวน	Null
9	guide_phone	Varchar (10)	เบอร์โทร	0952438249	Null
10	guide_birthday	Date	วันเกิด	2000-09-26	Null
11	house_no	Varchar (6)	บ้านเลขที่	35	Null
12	village_no	Varchar (6)	หมู่ที่	9	Null
13	sub_district	Varchar (20)	ตำบล	หนองกุงสวรรค์	Null
14	district	Varchar (20)	อำเภอ	โกสุมพิสัย	Null
15	province	Varchar (20)	จังหวัด	มหาสารคาม	Null
16	postal_code	Varchar (6)	รหัสไปรษณีย์	44140	Null
17	guide_passport country	Varchar(20)	ประเทศของหนังสือเดินทาง	Thailand	Null



ตารางที่ 3.4 ข้อมูลไกด์ (ต่อ)

id	Column	Type	Description	Example Data	Constraints
18	guide_detail_introduceyourself	Text	รายละเอียดการแนะนำตัว	สวัสดีค่ะฉันชื่อพรประภาตอนนี้ฉันเป็นไกด์	Null
19	guide_identificationcard	Varchar (255)	รูปถ่ายบัตรประจำตัวประชาชน	<a href="https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg">https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg</a>	Null
20	guide_cardandface	Varchar (255)	รูปถ่ายบัตรประจำตัวประชาชนพร้อมกับหน้าตนเอง	<a href="https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg">https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg</a>	Null
21	guide_bankaccount	Varchar (20)	เลขบัญชีธนาคาร	0280142993	Null
22	guide_verifystatus	int	สถานะการยืนยันตัวตนของไกด์	1	Null
23	guide_type	int	สถานะการทำงานของไกด์	0	Null
24	guide_dateapproved	date	วันที่ไกด์ได้รับการยืนยันตัวตนแล้ว	2022-03-02	Null
25	guide_score	float	คะแนนไกด์	4.5	Null

ตารางที่ 3.4 ข้อมูลไกด์ (ต่อ)

id	Column	Type	Description	Example Data	Constraints
26	guide_img	Varchar(255)	รูปภาพโปรไฟล์	https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg	Null

ตารางที่ 3.5 ข้อมูลนักท่องเที่ยว

id	Column	Type	Description	Example Data	Constraints
1	traveler_email	Varchar (100)	อีเมล	62011212014@msu.ac.th	PK
2	traveler_facebookid	Varchar (20)	รหัสบัญชี facebook	112514381367216	Null
3	traveler_password	Varchar (20)	รหัสผ่านของบัญชีเข้าสู่ระบบ	123456	Not Null
4	traveler_gender	Varchar (6)	เพศ	ชาย	Null
5	traveler_prefix	Varchar (6)	คำนำหน้าชื่อ	นาย	Null
6	traveler_name	Varchar (20)	ชื่อ	ณัฐกร	Null
7	traveler_surname	Varchar (20)	นามสกุล	บุรณ์เจริญ	Null
8	traveler_nickname	Varchar (10)	ชื่อเล่น	เปรม	Null
9	traveler_birthday	date	วันเกิด	2000-06-22	Null
10	house_no	Varchar (6)	บ้านเลขที่	1	Null
11	village_no	Varchar (6)	หมู่ที่	10	Null
12	sub_district	Varchar (20)	ตำบล	ขามเรียง	Null
13	district	Varchar (20)	อำเภอ	กันทรวิชัย	Null

ตารางที่ 3.5 ข้อมูลนักท่องเที่ยว (ต่อ)

id	Column	Type	Description	Example Data	Constraints
14	province	Varchar (20)	จังหวัด	มหาสารคาม	Null
15	postal_code	Varchar (6)	รหัสไปรษณีย์	44150	Null
16	traveler_passport country	Varchar (20)	ประเทศของ หนังสือ เดินทาง	Thailand	Null
17	traveler_img	Varchar (255)	รูปภาพโปร ไฟล์	<a href="https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg">https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg</a>	Null

ตารางที่ 3.6 ข้อมูลผู้ดูแล

id	Column	Type	Description	Example Data	Constraints
1	admin_email	Varchar (100)	อีเมล	62011212064@msu.ac.th	PK
2	admin_facebook id	Varchar (20)	รหัสบัญชี facebook	112514381367217	Null
3	admin_password	Varchar (20)	รหัสผ่านของ บัญชีเข้าสู่ ระบบ	123456	Not Null
4	admin_prefix	Varchar (6)	คำนำหน้าชื่อ	นาย	Null
5	admin_name	Varchar (20)	ชื่อ	ภูมินทร์	Null
6	admin_surname	Varchar (20)	นามสกุล	โพธิ์สุวรรณ	Null

ตารางที่ 3.6 ข้อมูลผู้ดูแล (ต่อ)

id	Column	Type	Description	Example Data	Constraints
7	admin_img	Varchar (255)	รูปภาพโปรไฟล์	https://i.ytimg.com/vi/nJ5MqwU6K1Y/maxresdefault.jpg	Null

ตารางที่ 3.7 ข้อมูลรายการธนาคาร

id	Column	Type	Description	Example Data	Constraints
1	bankguidestatement_id	int	รหัสรายการธนาคาร	1	PK
2	bankguidestatement_guidemail	Varchar (100)	อีเมลของไกด์	62011212064@msu.ac.th	Not Null
3	bankguidestatement_money	int	จำนวนเงิน	1500	Not Null
4	bankguidestatement_time	datetime	เวลาที่ทำรายการ	2022-03-02 20:32:00	Not Null
5	bankguidestatement_detail	Text	รายละเอียดธุรกรรม	เงินเข้าบัญชี	Not Null

ตารางที่ 3.8 ข้อมูลทริป

id	Column	Type	Description	Example Data	Constraints
1	trip_id	int	รหัสทริป	1	PK
2	guid_email	Varchar (100)	อีเมลของไกด์	62011212064@msu.ac.th	FK references from guide(guid_email)

ตารางที่ 3.8 ข้อมูลทริป(ต่อ)

id	Column	Type	Description	Example Data	Constraints
3	trip_name	Varchar (50)	ชื่อทริป	ทริปมหกรรม เดินกินเที่ยว ทั่วจังหวัด อุบลราชธานี	Not Null
4	trip_amountpeople	int	กำหนด จำนวน นักท่องเที่ยว	4	Not Null
5	trip_latitude	double	ละติจูด	123.454	Not Null
6	trip_longitude	double	ลองจิจูด	123.562	Not Null
7	trip_province	Varchar (20)	จังหวัดที่ ต้องการ จัดทริป	มหาสารคาม	Not Null
8	trip_detail	Text	รายละเอียด เบื้องต้นของ ทริป	เป็นทริปดำนา และไถ่จับปู	Null
9	trip_imgcoverpage	Varchar (255)	ภาพหน้าปก ของทริป	<a href="https://www.chillpainai.com/src/wakeup/scoop/images/e2d7939338b633b23b87afd5d2.jpg">https://www.chillpainai.com/src/wakeup/scoop/images/e2d7939338b633b23b87afd5d2.jpg</a>	Null
10	trip_status	int	สถานะของ ทริป	1	Not Null

ตารางที่ 3.9 ข้อมูลสถานที่

id	Column	Type	Description	Example Data	Constraints
1	place_id	int	รหัสวางแผน	10	PK
3	place_name	Varchar (100)	ชื่อสถานที่ที่วางแผนที่ไว้	ร้านอาหารนายทองดี	Not Null
4	place_latitude	double	ละติจูด	4234.546	Not Null
5	place_longitude	double	ลองจิจูด	123.657	Not Null
6	place_details	Text	รายละเอียดเบื้องต้นของการวางแผน	เป็นสถานที่ที่มีการสอนทำอาหารได้	Null
7	place_image	Varchar (255)	รูปภาพของสถานที่	<a href="https://www.chillpainai.com/src/wewa-keup/scoop/images/e2d7939338b633b23b87afd5d2.jpg">https://www.chillpainai.com/src/wewa-keup/scoop/images/e2d7939338b633b23b87afd5d2.jpg</a>	Null

ตารางที่ 3.10 ข้อมูลกิจกรรม

id	Column	Type	Description	Example Data	Constraints
1	event_id	int	รหัสสถานที่	3	PK
2	trip_id	int	รหัสทริป	1	FK references from trip(trip_id)

ตารางที่ 3.10 ข้อมูลกิจกรรม (ต่อ)

id	Column	Type	Description	Example Data	Constraints
3	place_id	int	รหัสวางแผน	10	FK references from plan (place_id)
4	event_time	datetime	เวลาที่เดินทางไปสถานที่	2022-03-02 20:32:00	Not Null

ตารางที่ 3.11 ข้อมูลรูปภาพ

id	Column	Type	Description	Example Data	Constraints
1	image_id	int	รหัสรูปภาพ	3	PK
2	trip_id	int	รหัสทริป	1	FK references from trip(trip_id)
3	image_pic	Varchar (255)	รูปภาพ	<a href="https://www.chillpainai.com/src/wewa-keup/scoop/images/e2d7939338b633b23b87afd5d2.jpg">https://www.chillpainai.com/src/wewa-keup/scoop/images/e2d7939338b633b23b87afd5d2.jpg</a>	Null

ตารางที่ 3.12 ข้อมูลการจอง

id	Column	Type	Description	Example Data	Constraints
1	reserve_id	int	รหัสการจอง	3	PK
2	trip_id	int	รหัสทริป	1	FK references from trip(trip_id)
3	traveler_email	Varchar (100)	อีเมล นักท่องเที่ยว	62011212064 @msu.ac.th	FK references from traveler (traveler_ email)
4	reserve_date time	datetime	วันเวลาที่ จองทริป	2022-03-08 17:27:59	Not Null
6	reserve_date travel	date	วันที่ต้องการ ไปเที่ยว	2000-09-26	Not Null
7	reserve_amount traveler	int	จำนวน นักท่องเที่ยว	2	Not Null
8	reserve_nationality	Varchar (20)	สัญชาติ	ไทย	Null
9	reserve_name	Varchar (20)	ชื่อจริง	ภูมินทร์ประกา สิทธิ์	Null
10	reserve_surname	Varchar (20)	นามสกุล	โพธิ์สุวรรณเลิศ งาม	Null
11	reserve_phone	Varchar (10)	เบอร์โทร	0952438249	Null
12	reserve_status	int	สถานะของ การจอง	0	Not Null



ตารางที่ 3.13 ข้อมูลรายงานความประพฤติ

id	Column	Type	Description	Example Data	Constraints
1	report_id	int	รหัสการ รายงาน	1	PK
2	traveler_email	Varchar (100)	อีเมล นักท่องเที่ยว	62011212064 @msu.ac.th	FK references from traveler (traveler_ email)
3	guid_email	Varchar (100)	อีเมลของไกด์	62011212014 @msu.ac.th	FK references from guide( guide_email)
4	report_message	Text	ข้อความที่ ต้องการ รายงานความ ประพฤติไกด์	ไกด์ไม่สุภาพ ครับ	Not Null

ตารางที่ 3.14 ข้อมูลการแสดงความคิดเห็น

id	Column	Type	Description	Example Data	Constraints
1	comment_id	int	รหัสการแสดง ความคิดเห็น	1	PK
2	trip_id	int	รหัสทริป	1	FK references from trip(trip_id)
3	comment_message	Text	ข้อแสดงความคิดเห็น	ทริปนี้ดีมาก เลยครับ	Null
4	comment_score	float	คะแนน	4.5	Not Null

ตารางที่ 3.14 ข้อมูลการแสดงความคิดเห็น (ต่อ)

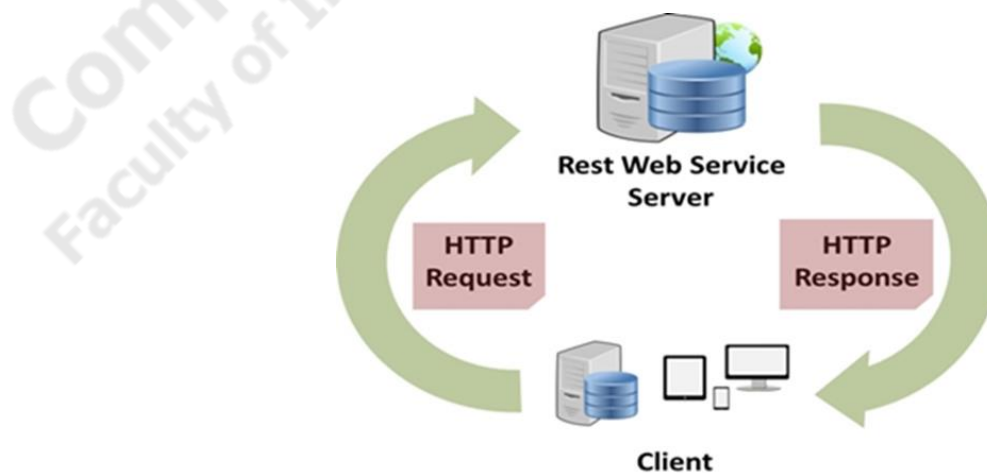
id	Column	Type	Description	Example Data	Constraints
5	comment_img	Varchar (255)	รูปภาพแสดงความคิดเห็น	https://www.chillpainai.com/src/wewa-keup/scoop/images/e2d7939338b633b23b87afd5d2.jpg	Null

### 3.9 อธิบายการทำงาน

#### 3.9.1 Web service

##### 3.9.1.1 การรับ - ส่งข้อมูลของเว็บเซอร์วิส

ในส่วนของการพัฒนาเว็บเซอร์วิสจะใช้แบบฟอร์มของ JSON ในการเรียงข้อมูลเพื่อ ส่งไปยังเซิร์ฟเวอร์ หรือแอปพลิเคชัน ให้สามารถติดต่อกันได้และรับส่งข้อมูลที่ต้องการประมวลผลได้ ด้วยการส่งผ่านข้อมูลทางอินเทอร์เน็ต ซึ่งนับว่าเป็นวิธีที่ให้ประสิทธิภาพที่ดีในการติดต่อสื่อสาร โดยใน แอปพลิเคชันได้นำเว็บเซอร์วิสมาช่วยในการพัฒนาระบบ ในส่วนของการรับ - ส่ง ข้อมูลต่างๆ เพื่อไปประมวลผลบนเซิร์ฟเวอร์และรอรับผลลัพธ์ที่ได้จากการประมวลผล ส่งมายังแอปพลิเคชัน



ภาพประกอบที่ 3.6 การรับ - ส่งข้อมูลของเว็บเซอร์วิส

### 3.9.1.2 การเข้าสู่ระบบของ Guide และ Traveler

```

3 public class UserDTO {
4     private String Email;
5     private String Facebookid;
6     private String Password ;

```

#### ภาพประกอบที่ 3.7 Data Transfer Object (DTO) ส่วนเข้าสู่ระบบ

ภาพประกอบที่ 3.11 Data Transfer Object (DTO) ส่วนเข้าสู่ระบบ Class UserDTO เป็น Class เพื่อที่จะรับข้อมูลจาก Font End โดยจะมีตัวแปรทั้งหมด 3 ตัวแปร คือ Email Facebookid และ Password ที่มีชนิดข้อมูลเป็นสตริง เพื่อรับค่า Email หรือ FacebookID และ Password มาใช้งาน

```

10
11 public interface GuideRepository extends JpaRepository<Guide, String> {
12
13     @Query("SELECT g FROM Guide g WHERE g.guideEmail = :email AND g.guidePassword = :password")
14     public List<Object[]> findGuideByGuideEmailAndGuidePassword(String email, String password);
15 }
16
17
18
19
20
21 public interface TravelerRepository extends JpaRepository<Traveler, String> {
22
23     @Query("SELECT t FROM Traveler t WHERE t.travelerEmail = :email AND t.travelerPassword = :password")
24     public List<Object[]> findTravelerByTravelerEmailAndTravelerPassword(String email, String password);
25 }
26 }
27

```

#### ภาพประกอบที่ 3.8 Repositories ของการเข้าสู่ระบบของ ผู้ใช้งานประเภท Guide และ Traveler

ภาพประกอบที่ 3.12 Repositories ของการเข้าสู่ระบบของ ผู้ใช้งานประเภท Guide และ Traveler จะสร้าง Interface GuideRepository เพื่อเก็บคำสั่ง JPQL ส่วน Guide และกำหนดชนิดข้อมูลให้เป็นสตริงแล้วการสร้าง Method “findGuideByGuideEmailAndGuidePasswrod” และมี Parameter 2 ตัว คือ Email และ Password โดยมีชนิดข้อมูลเป็น “List<Object[]>” ในส่วนของการสร้าง Interface TravalerRepository ก็ทำดั่ง Interface GuideRepository แต่เป็นส่วนของ Traveler

```

28 @Service
29 public class MyService {
30     @Autowired
31     GuideRepository guideRepository;
32
33     @Autowired
34     TravelerRepository travelerRepository;
35
36     @Autowired
37     PlanRepository planRepository;
38
39     @Autowired
40     TripRepository tripRepository;
41
42     @Autowired
43     PlaceRepository placeRepository;
44
45     @Autowired
46     ReserveRepository reserveRepository;
47

```

### ภาพประกอบที่ 3.9 การผูก Repository เพื่อนำมาใช้ใน Service

ภาพประกอบที่ 3.13 การผูก Repository เพื่อนำมาใช้ใน Service จะสร้าง Service ชื่อ MyService ระบุ @Autowired เพื่อเป็นการ Connect ไปยัง database และ ทำการผูก Repository ประกอบด้วย GuideRepository TravelerRepository PlanRepository TripRepository PlaceRepository และ ReserveRepository เพื่อนำมาใช้ใน Service

```

72
73     public List<Object[]> loginTraveler(String email, String password){
74         return travelerRepository.findTravelerByEmailAndPassword(email, password);
75     }
76
77     public List<Object[]> loginGuide(String email, String password){
78         return guideRepository.findGuideByEmailAndPassword(email, password);
79     }
80

```

### ภาพประกอบที่ 3.10 Service ของการเข้าสู่ระบบของ Guide และ Traveler

ภาพประกอบที่ 3.14 Service ของการเข้าสู่ระบบของ Guide และ Traveler จะสร้าง Method “loginTraveler” โดยมีชนิดข้อมูลเป็น “List<Object[]>” เพื่อแสดงข้อมูลผู้ใช้งานที่มี Email และ Password ตรงกันกับค่าที่รับเข้ามาจาก Font End ของ Traveler ในส่วนของ Guide ก็ทำเหมือนกัน โดยจะ สร้าง Method “loginGuide” โดยมีชนิดข้อมูลเป็น “List<Object[]>”

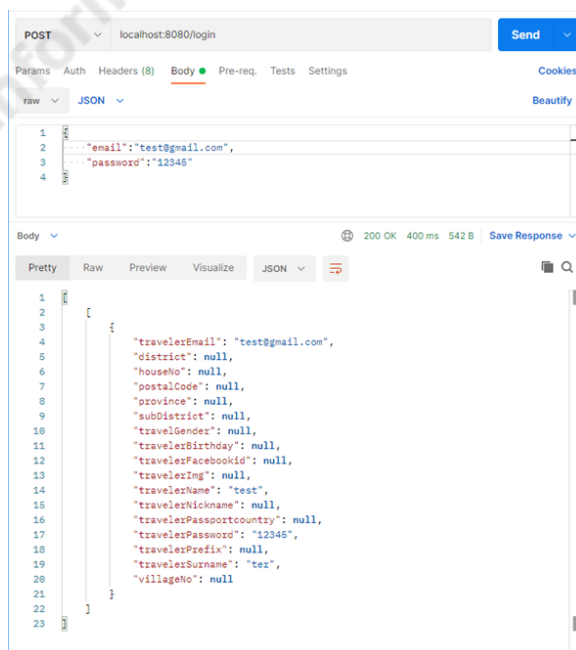
```

14
15 @RestController
16 public class LoginController {
17     @Autowired
18     private MyService service;
19
20     @PostMapping(value = "/login",
21                 consumes = MediaType.APPLICATION_JSON_VALUE,
22                 produces = MediaType.APPLICATION_JSON_VALUE)
23     public ResponseEntity<?> login(@RequestBody UserDTO user){
24         if (service.loginGuide(user.getEmail(), user.getPassword()).isEmpty() != true) {
25             return new ResponseEntity<>(service.loginGuide(user.getEmail(), user.getPassword()),
26                                     HttpStatus.OK);
27         }else if (service.loginTraveler(user.getEmail(), user.getPassword()).isEmpty() != true) {
28             return new ResponseEntity<>(service.loginTraveler(user.getEmail(), user.getPassword()),
29                                     HttpStatus.OK);
30         }else {
31             return null;
32         }
33     }
34 }

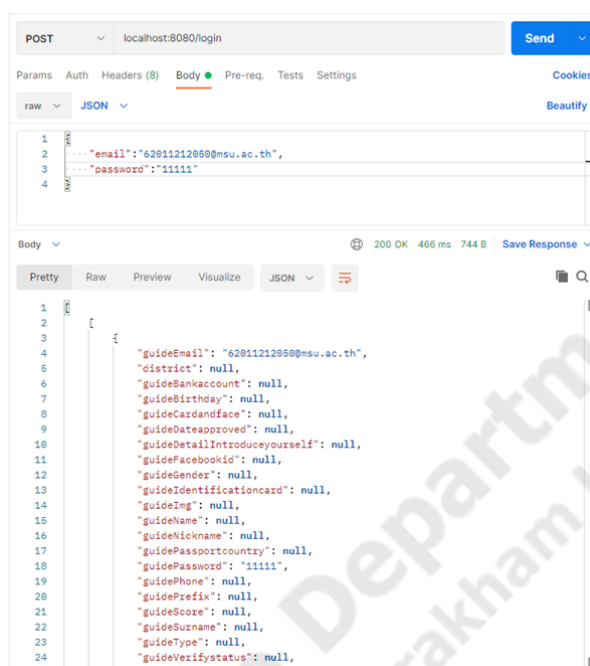
```

### ภาพประกอบที่ 3.11 ส่วนของ Controller การเข้าสู่ระบบ

ภาพประกอบที่ 3.15 ส่วนของ Controller การเข้าสู่ระบบ จะสร้าง Class “LoginController” ระบุ @RestController เพื่อให้ทราบว่า Class นี้คือ Restful Controller แล้วสร้าง ตัวแปร “MyService” ระบุ @Autowired ให้กับ “MyService” เพื่อเป็นการ Connect ไปยัง database และสร้าง Controller โดยใช้ @PostMapping เพื่อระบุ Path ให้กับ Controller โดยจะกำหนดชนิดให้ consumes และ produces เป็น JSON เพื่อระบุว่าข้อมูลมี รูปแบบ JSON จากนั้น สร้าง Method “login” และมีค่าพารามิเตอร์เป็น UserDTO ให้กับ Method เมื่อตรวจสอบว่าถ้าส่งข้อมูล Email เมื่อเป็นผู้ใช้งาน Guild จะทำงานที่ Method “loginGuide” หากเป็นผู้ใช้งาน Traveler จะทำงานที่ Method “loginTraveler”



### ภาพประกอบที่ 3.12 ตัวอย่างการเรียกใช้ Service การเข้าสู่ระบบของ Traveler



### ภาพประกอบที่ 3.13 ตัวอย่างการเรียกใช้ Service การเข้าสู่ระบบของ Guide

#### 3.9.1.3 Service ของการสมัครสมาชิกของ Guide และ Traveler

```

81 public Traveler registerTraveler(UserDTO user) {
82     Traveler traveler = new Traveler(user.getEmail(),user.getPassword(),user.getName(),user.getSurname());
83     return travelerRepository.save(traveler);
84 }
85
86 public Guide registerGuide(UserDTO user) {
87     Guide guide = new Guide(user.getEmail(),user.getPassword(),user.getName(),user.getSurname());
88     return guideRepository.save(guide);
89 }
90

```

### ภาพประกอบที่ 3.14 Service ของการสมัครสมาชิกของ Guide และ Traveler

ภาพประกอบที่ 3.18 Service ของการสมัครสมาชิกของ Guide และ Traveler จะสร้าง Method “registerTraveler” เพื่อเพิ่มข้อมูลผู้ใช้งานประเภท Traveler ลงใน database โดยจะรับข้อมูลมาจาก Font End โดยมีพารามิเตอร์ เป็น UserDTO ดังภาพประกอบที่ 3.18 ในส่วนของการเพิ่มข้อมูลผู้ใช้งานประเภท Guide จะทำงานเหมือนกันกับ ผู้ใช้งานประเภท Traveler

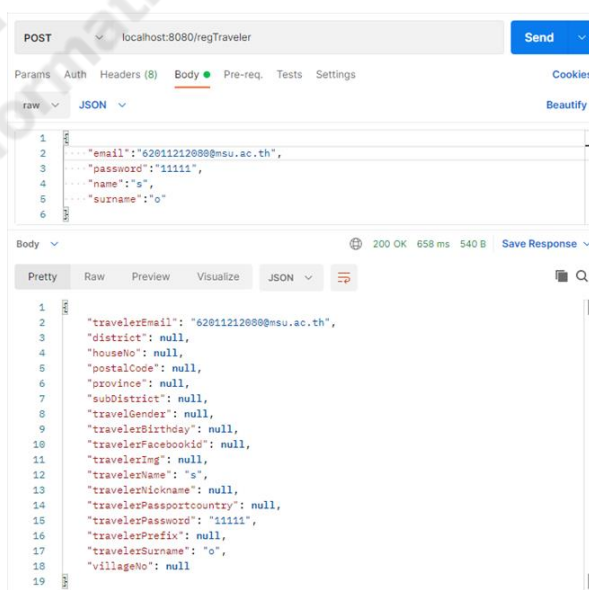
```

16 @RestController
17 public class RegisterController {
18     @Autowired
19     private MyService service;
20
21     @PostMapping(value = "/regTraveler", consumes = MediaType.APPLICATION_JSON_VALUE,
22                 produces = MediaType.APPLICATION_JSON_VALUE)
23     public ResponseEntity<> registerTraveler(@RequestBody UserDTO user) {
24         if (service.getTravelerByEmail(user.getEmail()).isEmpty() == true) {
25             return new ResponseEntity<>(service.registerTraveler(user), HttpStatus.OK);
26         }
27         return null;
28     }
29
30     @PostMapping(value = "/regGuide", consumes = MediaType.APPLICATION_JSON_VALUE,
31                 produces = MediaType.APPLICATION_JSON_VALUE)
32     public ResponseEntity<> registerGuide(@RequestBody UserDTO user) {
33         if (service.getGuideByEmail(user.getEmail()).isEmpty() == true) {
34             return new ResponseEntity<>(service.registerGuide(user), HttpStatus.OK);
35         }
36         return null;
37     }
38 }
39
40
41

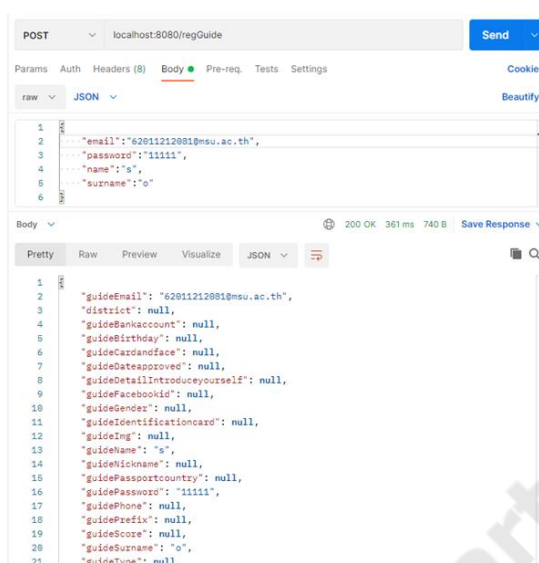
```

ภาพประกอบที่ 3.15 ส่วนของ Controller การสมัครสมาชิก

ภาพประกอบที่ 3.19 ส่วนของ Controller การสมัครสมาชิก ใน Class “RegisterController” จะระบุ @RestController เพื่อให้ทราบว่า Class นี้คือ Restful Controller แล้วสร้าง ตัวแปร “MyService” ระบุ @Autowired ให้กับ “MyService” เพื่อเป็นการ Connect ไปยัง database และสร้าง Controller โดยใช้ @PostMapping เพื่อระบุ Path ให้กับ Controller โดยจะกำหนดชนิดให้ consumes และ produces เป็น JSON เพื่อระบุว่าข้อมูลมี รูปแบบ JSON จากนั้น สร้าง Method “registerTraveler” และมีค่าพารามิเตอร์เป็น UserDTO ให้กับ Method แล้วตรวจสอบดูว่าถ้า traveler มี Email อยู่หรือไม่ ถ้าไม่มีจะเข้าเงื่อนไขในการสมัครสมาชิกในส่วนของการสร้าง Method “registerGuide” ซึ่งมีการทำงานเดียวกันกับ Method “registerTraveler”



ภาพประกอบที่ 3.16 ตัวอย่างการเรียกใช้ Service การสมัครสมาชิกของ Traveler



ภาพประกอบที่ 3.17 ตัวอย่างการเรียกใช้ Service การสมัครสมาชิกของ Guide  
3.9.1.4 การจองทริป

```

8 public class ReserveDTO {
9     private int reserveId;
10    private String reserveAmounttraveler;
11    private String reserveComment;
12    private Date reserveDatetime;
13    private Date reserveDateTravel;
14    private String reserveDeposit;
15    private String reserveImg;
16    private String reserveNationality;
17    private String reservePrice;
18    private String reserveScore;
19    private String reserveStatus;
20    private Traveler traveler;
21    private Trip trip;

```

ภาพประกอบที่ 3.18 Data Transfer Object (DTO) ส่วนการจองทริป

ภาพประกอบที่ 3.22 Data Transfer Object (DTO) ส่วนการจองทริป Class ReserveDTO เป็น Class เพื่อที่จะรับข้อมูลจาก Font End โดยจะมีตัวแปรทั้งหมด 13 ตัวแปร คือ reserveId ที่มีชนิดข้อมูลเป็นอินเทเจอร์ reserveDatetime และreserveDateTravel ที่มีชนิดข้อมูลเป็นเดท traveler ที่มีชนิดข้อมูลเป็น traveler trip ที่มีชนิดข้อมูลเป็น trip และ ตัวแปรที่เหลือจะมีชนิดข้อมูลเป็นสตริง ดังภาพประกอบที่ 3.22



```

13 public interface ReserveRepository extends JpaRepository<Reserve, String> {
14     @Query("SELECT r FROM Reserve r WHERE r.trip.tripId = :tripId")
15     public List<Reserve> findReserveByTripId(int tripId);
16
17     @Query("SELECT r FROM Reserve r WHERE r.trip.tripId = :tripId AND reserveDatetr
18     public Reserve findReserveBTripIdandDate(int tripId,Date date);
19 }
20

```

### ภาพประกอบที่ 3.19 Repositories ของการจองทริป

Repositories ของการจองทริปจะสร้าง Interface ReserverRepository เพื่อเก็บคำสั่ง JPQL ส่วน Reserve และกำหนดชนิดข้อมูลให้เป็นสตริง แล้วการสร้าง Method “findReserveByTripld” และมี tripID เป็นพารามิเตอร์ เพื่อค้นหา Reserve ที่มี tripld เท่ากับค่าที่ส่งมาจาก Font End โดยมีชนิดข้อมูลเป็น “List<Reserve >” ส่วนของ Method “findReserveByTripldandDate” เพื่อค้นหา Reserve ที่มี tripld เท่ากับค่าที่ส่งมาจาก Font End และ reserveDateTravel เท่ากับ date ที่ส่งมาจาก Font End ดังภาพประกอบที่ 3.23

```

117     public void updateReserve(ReserveDTO reserveDTO){
118         Reserve reserve = reserveRepository.findReserveBTripIdandDate(reserveDTO.g
119         reserve.setTraveler(reserveDTO.getTraveler());
120         reserve.setReserveAmounttraveler(reserveDTO.getReserveAmounttraveler());
121         reserve.setReserveNationality(reserveDTO.getReserveNationality());
122         reserve.setReserveDatetime(reserveDTO.getReserveDatetime());
123         reserve.setReserveStatus(reserveDTO.getReserveStatus());
124         reserveRepository.save(reserve);
125     }
126 }
127

```

### ภาพประกอบที่ 3.20 Service ของการจองทริป

ภาพประกอบที่ 3.24 Service ของการจองทริป จะสร้าง Method “updateReserve” เพื่ออัปเดตข้อมูลของ Reserver โดยจะรับข้อมูลมาจาก Font End โดยมีพารามิเตอร์ เป็น ReserveDTO จากนั้นสร้างตัวแปร reserve เพื่อเก็บข้อมูล reserver ที่ค้นหาได้จาก tripld และ reserveDateTravel ทำการเซ็ตค่า reserve ที่จะแก้ไข โดยรับค่าจากพารามิเตอร์ ที่รับมา แล้วใช้ Method save ในการอัปเดตข้อมูลของ Reserve

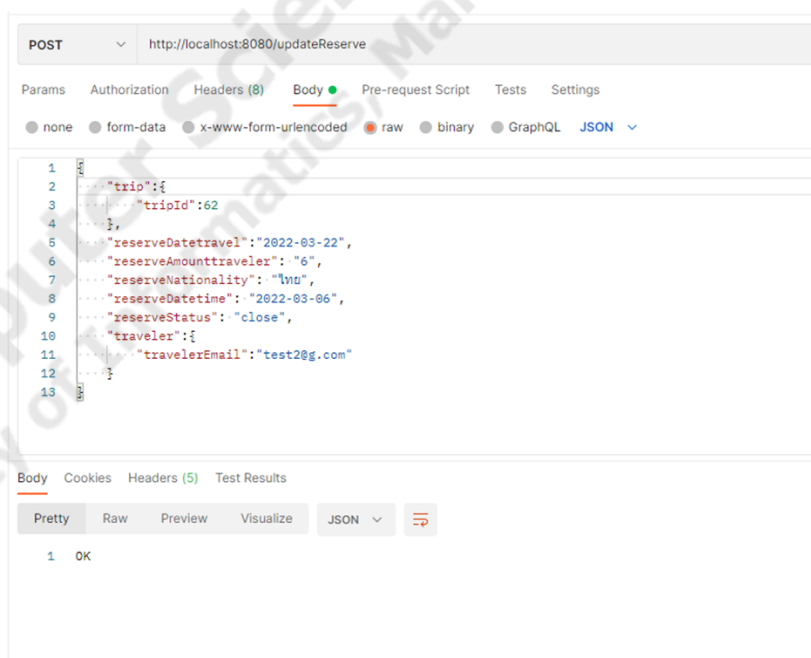
```

15 @RestController
16 public class ReserveController {
17     @Autowired
18     MyService service;
19
20
21
22
23
24
25
26
27
28
29
30 @PostMapping(value = "/updateReserve", consumes = MediaType.APPLICATION_JSON_VALUE, produces = MediaType.A
31 public ResponseEntity<?> updateReserve(@RequestBody ReserveDTO reserveDTO) {
32     service.updateReserve(reserveDTO);
33     return new ResponseEntity<>("OK",HttpStatus.OK);
34 }
35 }
--

```

ภาพประกอบที่ 3.21 ส่วนของ Controller ของการจองทริป

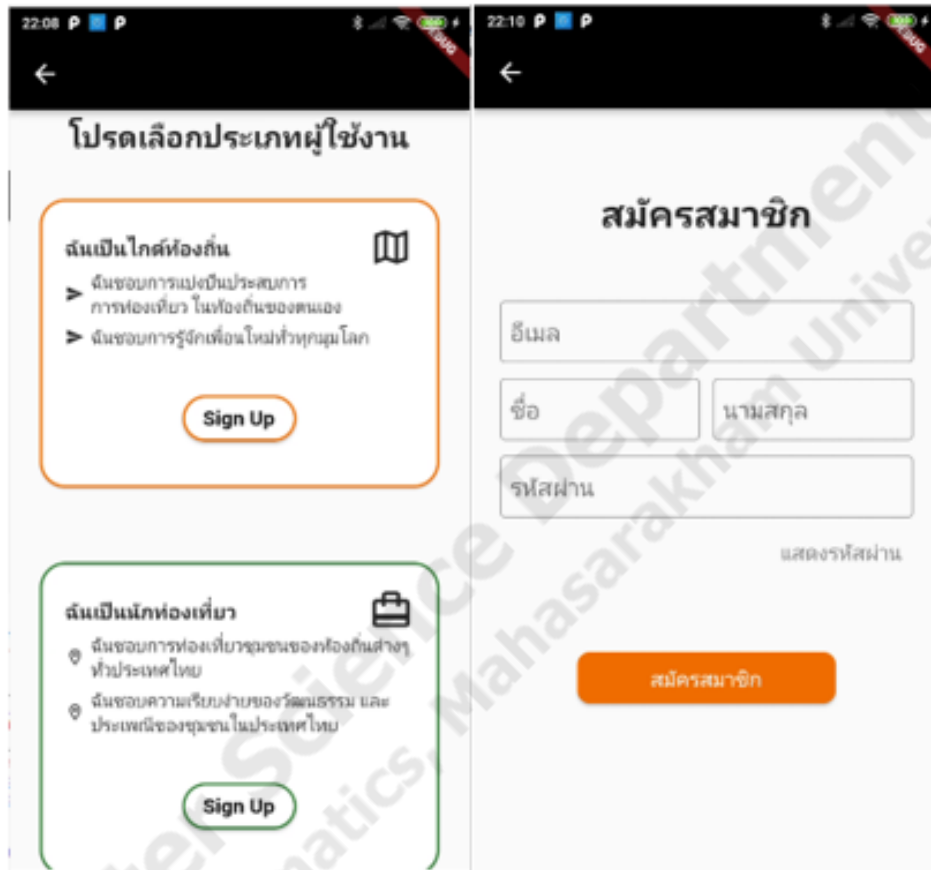
ภาพประกอบที่ 3.25 ส่วนของ Controller ของการจองทริป ใน Class “RegisterController” จะระบุ @RestController เพื่อให้ทราบว่า Class นี้คือ Restful Controller แล้วสร้าง ตัวแปร “MyService” ระบุ @Autowired ให้กับ “MyService” เพื่อเป็นการ Connect ไปยัง database และสร้าง Controller โดยใช้ @PostMapping เพื่อระบุ Path ให้กับ Controller โดยจะกำหนดชนิดให้ consumes และ produces เป็น JSON เพื่อระบุว่าข้อมูลมี รูปแบบ JSON จากนั้น สร้าง Method “updateReserve” และมีค่าพารามิเตอร์เป็น ReserveDTO ให้กับ Method จะใช้ Method “updateReserve” ให้การอัปเดตข้อมูลใน Database



ภาพประกอบที่ 3.22 ตัวอย่างการเรียกใช้ Service การจองทริป

### 3.9.2 หน้าจอแอปพลิเคชันและการเรียกใช้งาน API

#### 3.9.2.1 หน้าการสมัครสมาชิก Guide และ Traveler



ภาพประกอบที่ 3.23 หน้าการสมัครสมาชิก Guide และ Traveler

ภาพประกอบที่ 3.27 หน้าการสมัครสมาชิก Guide และ Traveler เป็นหน้าที่ใช้สำหรับการสมัครสมาชิกของผู้ใช้งาน ไกด์ และ นักท่องเที่ยว โดยจะให้เลือกประเภทผู้ใช้งานก่อนที่จะกรอกข้อมูลในการสมัครสมาชิกจะให้กรอก อีเมล ชื่อ นามสกุล และ รหัสผ่าน เมื่อกด “สมัครสมาชิก” ระบบจะทำการเพิ่มข้อมูลผู้ใช้งานที่ป้อนเข้ามาลงใน database ดัง ภาพประกอบที่ 3.28 การสมัครสมาชิก

```

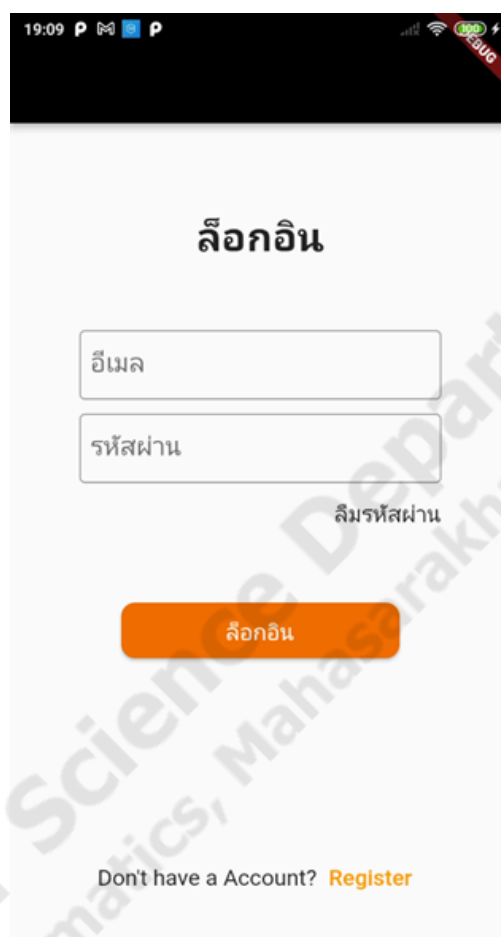
143     onPressed: () async {
144       var user = User(
145         email: emailController.text,
146         password: passWordController.text,
147         name: nameController.text,
148         surname: surNameController.text); // User
149       String json = userToJson(user);
150       var response = await http.post(
151         Uri.parse('http://$path:8080/${typeModes}'),
152         body: json,
153         headers: {
154           HttpHeaders.contentTypeHeader: 'application/json'
155         });
156       if (response.body.isNotEmpty) {
157         Navigator.push(
158           context,
159           MaterialPageRoute(builder: (context) => Login()),
160         ).then((value) => null);
161       }
162     },

```

ภาพประกอบที่ 3.24 การสมัครสมาชิก

ภาพประกอบที่ 3.28 การสมัครสมาชิก เป็นการทำงานในการสมัครสมาชิก เมื่อผู้ใช้งานกรอกข้อมูลแล้วกดปุ่ม “สมัครสมาชิก” ระบบจะรับข้อมูลจากผู้ใช้งานกรอกเข้ามา แล้วระบบจะเรียกใช้ WebService Register จะมีพารามิเตอร์ “typeModes” จะตรวจสอบว่าเป็นผู้ใช้งานประเภทใด จากนั้นระบบจะใช้ Method Post ในการ Request ข้อมูล จากนั้นนำ Response ของการ Request ข้อมูลที่ได้มาตรวจสอบ และถ้า Response มีข้อมูล จะไปที่หน้าล็อกอิน

### 3.9.2.2 หน้าการเข้าสู่ระบบด้วย Email และ Password



19:09 P ๙ P

ล็อกอิน

อีเมล

รหัสผ่าน

ลืมรหัสผ่าน

ล็อกอิน

Don't have a Account? Register

ภาพประกอบที่ 3.25 หน้าการเข้าสู่ระบบด้วย Email และ Password

ภาพประกอบที่ 3.29 หน้าการเข้าสู่ระบบด้วย Email และ Password เป็นหน้าที่ใช้สำหรับการเข้าสู่ระบบของผู้ใช้งาน ไรต์ และ นักท่องเที่ยว โดยจะให้กรอก อีเมล รหัสผ่าน เมื่อกด “ล็อกอิน” ระบบจะทำการตรวจสอบข้อมูลผู้ใช้งานที่ป้อนเข้ามา ดัง ภาพประกอบที่ 3.30 ตรวจสอบการเข้าสู่ระบบ

```

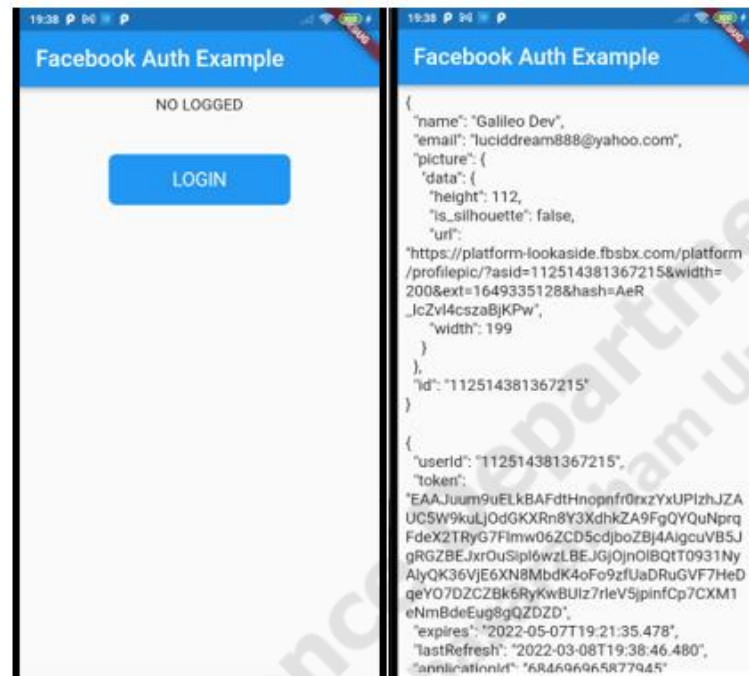
90     onPressed: () async {
91       var user = User(
92         email: emailController.text,
93         password: passWordController.text); // User
94       String json = userToJson(user);
95       var response = await http.post(
96         Uri.parse('http://$path:8080/login'),
97         body: json,
98         headers: {
99           HttpHeaders.contentTypeHeader: 'application/json'
100        });
101       if (response.body.isNotEmpty) {
102         var result = convert.jsonDecode(response.body);
103         bool typeMode = result.toString().contains("guide");
104         if (typeMode == true) {
105           Navigator.push(
106             context,
107             MaterialPageRoute(
108               builder: (context) => CreateTrip(
109                 email: emailController.text,
110               )), // CreateTrip // MaterialPageRoute
111             ).then((value) => null);
112         } else {
113           Navigator.push(
114             context,
115             MaterialPageRoute(
116               builder: (context) => ReserveTips(
117                 email: emailController.text,
118               )), // ReserveTips // MaterialPageRoute
119             ).then((value) => null);
120         }
121       }
122     },

```

### ภาพประกอบที่ 3.26 ตรวจสอบการเข้าสู่ระบบ

ภาพประกอบที่ 3.30 ตรวจสอบการเข้าสู่ระบบ เป็นการทำงานในการเข้าสู่ระบบ เมื่อผู้ใช้งานกรอกข้อมูล แล้วกดปุ่ม “ล็อกอิน” ระบบจะรับข้อมูลที่ผู้ใช้งานกรอกเข้ามา แล้วระบบจะเรียกใช้ WebService Login โดยการใช Method Post ในการ Request ข้อมูล จากนั้นนำ Response ของการ Request ข้อมูลที่ได้มาตรวจสอบ และถ้า Response มีข้อมูล จะไปที่หน้าหลัก และจะแสดงว่าข้อมูลในการเข้าสู่ระบบในครั้งนั้น เป็นของผู้ใช้งานบัญชีใด

## 3.9.2.3 หน้าการล็อกอินด้วย Facebook API



ภาพประกอบที่ 3.27 หน้าการล็อกอินด้วย Facebook API

```

55 | Future<void> _login() async {
56 |   final LoginResult result = await FacebookAuth.instance.login();
57 |   if (result.status == LoginStatus.success) {
58 |     _accessToken = result.accessToken;
59 |     _printCredentials();
60 |     final userData = await FacebookAuth.instance.getUserData();
61 |     _userData = userData;
62 |   } else {
63 |     print(result.status);
64 |     print(result.message);
65 |   }
66 |
67 |   setState(() {
68 |     _checking = false;
69 |   });
70 | }

```

ภาพประกอบที่ 3.28 การเชื่อมต่อ Facebook Login

ภาพประกอบที่ 3.32 การเชื่อมต่อ Facebook Login เป็นการทำงานในการเข้าสู่ระบบด้วย Facebook API เมื่อผู้ใช้งานกดปุ่ม “LOGIN” ระบบจะเรียกใช้ Facebook API โดยจะไปดึงข้อมูลจาก แอป Facebook ที่มีอยู่ในมือถือ ด้วย Method “FacebookAuth.instance.login()” หากแอป Facebook ไม่เคยเชื่อมต่อกับระบบ Facebook API จะทำการ ให้ทำการเชื่อมต่อ จากนั้นนำ จะดึง ข้อมูลด้วย Method “FacebookAuth.instance.getUserData()” จากนั้นแสดงข้อมูลผู้ใช่ออกมา

#### 3.9.2.4 หน้าการจองทริป

#### ภาพประกอบที่ 3.29 หน้าการจองทริป

ภาพประกอบที่ 3.33 หน้าการจองทริป เป็นหน้าที่ใช้สำหรับการจองทริป ของผู้ใช้งาน นักท่องเที่ยว โดยจะให้กรอก คำนำหน้า ชื่อ นามสกุล อีเมล เบอร์มือถือ และ สัญชาติ เมื่อกด “ยืนยันการจอง” ระบบจะทำการอัปเดตข้อมูลของการจองที่ป้อนเข้ามา ดัง ภาพประกอบที่ 3.34 การจองทริป



```

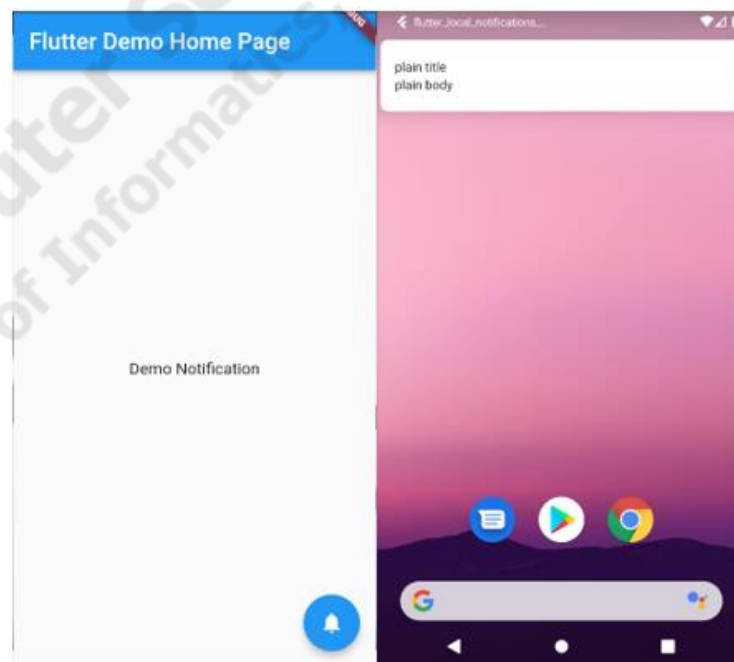
239 onPressed: () async {
240   ReserveUpdate reserve = ReserveUpdate(
241     trip: Trip(tripId: widget.tripId),
242     reserveDatetravel: widget.date,
243     reserveAmounttraveler: widget.amountOfPeople,
244     reserveNationality: nationality.text,
245     reserveDatetime: DateTime.now(),
246     reserveStatus: "close",
247     traveler: Traveler(travelerEmail: widget.email!)); //
248   String json = reserveUpdateToJson(reserve);
249   var response = await http.post(
250     Uri.parse('http://$path:8080/updateReserve'),
251     body: json,
252     headers: {
253       HttpHeaders.contentTypeHeader: 'application/json'
254     });
255   if (response.statusCode == 200) {
256     showDialog(

```

### ภาพประกอบที่ 3.30 การจองทริป

ภาพประกอบที่ 3.34 การจองทริป เป็นการทำงานในการจองทริป เมื่อผู้ใช้งานกดปุ่ม “ยืนยันการจอง” ระบบจะรับข้อมูลที่ผู้ใช้งานกรอกเข้ามา ระบบจะเรียกใช้ Webservice UpdateReserve โดยการใช Method Post ในการ Request ข้อมูล จากนั้นนำ Response ของการ Request ข้อมูลที่ได้มาตรวจสอบ และถ้า Response มีข้อมูล จะทำการแสดง Dialog ออกมา

#### 3.9.2.5 หน้าการแจ้งเตือน Local Notification



ภาพประกอบที่ 3.31 หน้าการแจ้งเตือน Local Notification

```

6  class NotificationApi {
7      static final flutterLocalNotificationsPlugin =
8          FlutterLocalNotificationsPlugin();
9      static final onNotifications = BehaviorSubject<String>();
10     static Future _notificationDetails() async {
11         return const NotificationDetails(
12             android: AndroidNotificationDetails(
13                 'nextflow_noti_001',
14                 'แจ้งเดือนทั่วไป',
15                 importance: Importance.max,
16             ), // AndroidNotificationDetails
17             iOS: IOSNotificationDetails(),
18         ); // NotificationDetails
19     }

```

### ภาพประกอบที่ 3.32 การสร้างตัวแปรของ Notification

ภาพประกอบที่ 3.36 การสร้างตัวแปรของ Notification เป็นการตั้งค่าของ LocalNotification โดยจะตั้งค่า Android และ ios ของ LocalNotification

```

21     static Future init({bool initScheduled = false}) async {
22         var android = AndroidInitializationSettings("@mipmap/ic_launcher");
23         var iOS = IOSInitializationSettings();
24         var settings = InitializationSettings(android: android, iOS: iOS);
25
26         final details =
27             await flutterLocalNotificationsPlugin.getNotificationAppLaunchDetails();
28         if (details != null && details.didNotificationLaunchApp) {
29             onNotifications.add(details.payload!);
30         }
31
32         await flutterLocalNotificationsPlugin.initialize(
33             settings,
34             onSelectNotification: (payload) async {
35                 onNotifications.add(payload!);
36             },
37         );
38     }
20

```

### ภาพประกอบที่ 3.33 การตั้งค่าของ LocalNotification

ภาพประกอบที่ 3.37 การตั้งค่าของ LocalNotification เป็นการตั้งค่าของ LocalNotification โดยจะตั้งค่าในส่วนของ background ของ LocalNotification ซึ่งจะต้องใส่ icon และ payload เมื่อ กดแท็บการแจ้งเตือน

```

40  static Future<void> showNotification({
41    int id = 0,
42    String? title,
43    String? body,
44    String? payload,
45    required DateTime scheduledDate,
46  }) async {
47    flutterLocalNotificationsPlugin.schedule(
48      id,
49      title,
50      body,
51      scheduledDate,
52      await _notificationDetails(),
53      payload: payload,
54    );
55  }
56 }
57

```

### ภาพประกอบที่ 3.34 ฟังก์ชันการแจ้งเตือน Notification

ภาพประกอบที่ 3.38 ฟังก์ชันการแจ้งเตือน Notification เป็นฟังก์ชันการแจ้งเตือน โดยจะมีพารามิเตอร์ id title body payload เพื่อจะแจ้งเตือนในเวลาถัดไป

```

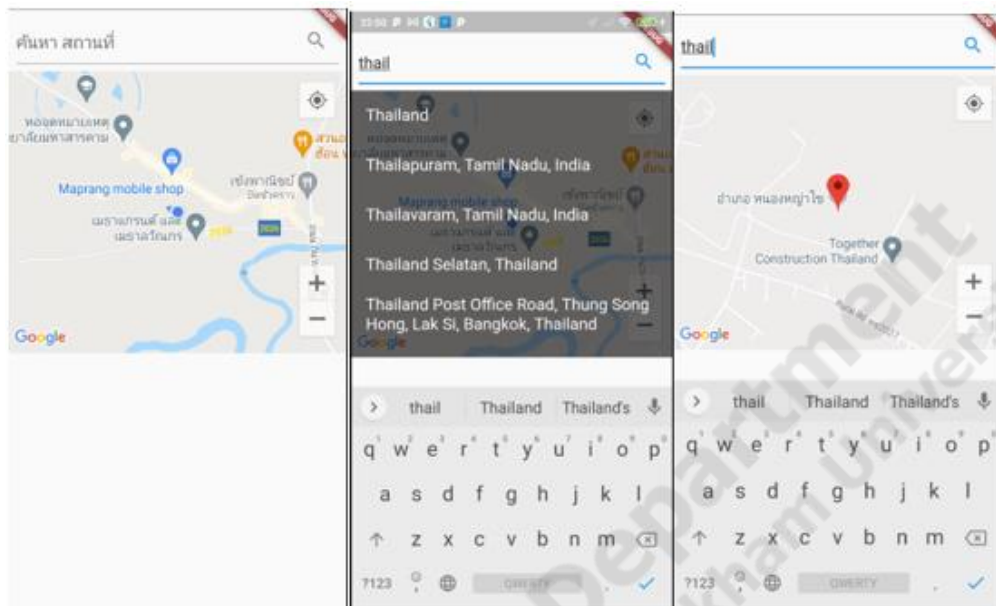
126  floatingActionButton: FloatingActionButton(
127    onPressed: () => NotificationApi.showNotification(
128      id: 0,
129      title: 'dwd',
130      body: 'daw',
131      scheduledDate: DateTime.now().add(Duration(seconds: 12)),
132      payload: 'dsad'),
133    tooltip: 'Increment',
134    child: const Icon(Icons.notifications),
135  ), // This trailing comma makes auto-formatting nicer for build methods. /
136  ); // Scaffold
137  }
138  }

```

### ภาพประกอบที่ 3.35 การเรียกใช้ Notification

ภาพประกอบที่ 3.39 การเรียกใช้ Notification เป็นการทำงานการแจ้งเตือนเมื่อผู้ใช้งานกดปุ่ม ระบบจะทำการแจ้งเตือน ตามพารามิเตอร์ที่ได้

## 3.9.2.6 หน้าการค้นหาสถานที่



ภาพประกอบที่ 3.36 หน้าการค้นหาสถานที่

```

114 Future<void> _goToPlace(Geometry place) async {
115   final GoogleMapController controller = await _mapController.future;
116   controller.animateCamera(CameraUpdate.newCameraPosition(
117     CameraPosition(
118       target:
119         LatLng(place.geometry.location.lat, place.geometry.location.lng),
120       zoom: 15), // CameraPosition
121   ));
122
123   allMarkers.add(Marker(
124     markerId: MarkerId('myLocation'),
125     draggable: false,
126     onTap: () {
127       print('Marker Tapped');
128     },
129     position:
130       LatLng(place.geometry.location.lat, place.geometry.location.lng)));
131 }
132 }
133

```

ภาพประกอบที่ 3.37 ฟังก์ชันการไปหาสถานที่ และ marker จุด

ภาพประกอบที่ 3.41 ฟังก์ชันการไปหาสถานที่ และ marker จุด เป็นการทำงานการเปลี่ยนมุมมองไปยังสถานที่ ที่ค้นหา โดยรับ พารามิเตอร์ เป็นชนิดข้อมูล Geometry เมื่อระบบเปลี่ยนมุมมองเสร็จ ระบบจะทำการ ตั้งค่าจุด ด้วย Method “AllMarker.Add()” เพื่อเพิ่ม จุด ไว้ใน Map โดยพารามิเตอร์ที่ได้ จะทำดัง ภาพประกอบที่ 3.42 การเรียกใช้ Geolocator เพื่อค้นหา location

```

3  class GeolocatorService {
4  Future<Position> getLocation() async {
5
6  return await Geolocator.getCurrentPosition(
7  |   desiredAccuracy: LocationAccuracy.high);
8  }
9  }

```

ภาพประกอบที่ 3.38 การเรียกใช้ Geolocator เพื่อค้นหา location

ภาพประกอบที่ 3.42 การเรียกใช้ Geolocator เพื่อค้นหา location เป็น Method เพื่อค้นหา location โดยจะดึง location ของผู้ใช้งานจากอุปกรณ์

```

11 class ApplicationBloc with ChangeNotifier {
12   final geolocatorService = GeolocatorService();
13   final placesService = PlacesService();
14
15   Position? currentLocation;
16   List<PlaceSearch> searchResults = [];
17   StreamController<Geometry> selectedLocation = StreamController<Geometry>();
18
19   ApplicationBloc() {
20     setCurrentLocation();
21   }
22
23   setCurrentLocation() async {
24     currentLocation = await geolocatorService.getCurrentLocation();
25     notifyListeners();
26   }
27
28   searchPlaces(String searchTerm) async {
29     searchResults = await placesService.getAutocomplete(searchTerm);
30     notifyListeners();
31   }
32
33   setSelectedLocation(String placeId) async {
34     selectedLocation.add(await placesService.getPlace(placeId));
35     searchResults = [];
36     notifyListeners();
37   }
38
39   @override
40   void dispose() {
41     selectedLocation.close();
42     super.dispose();
43   }
44 }
45

```

ภาพประกอบที่ 3.39 การอัปเดตข้อมูล Location

ภาพประกอบที่ 3.43 การอัปเดตข้อมูล Location เป็น Class ApplicationBloc with ChangeNotifier ที่ใช้ร่วมกับ Provider โดยเรียกใช้ GeolocatorService และ placesService จะเรียกใช้ Method setCurrentLocation() เพื่อที่จะตั้งค่าของข้อมูล currentLocation ใน Method “setCurrentLocation()” จะดึงข้อมูล location ที่ค้นหา ที่มีชนิดข้อมูลเป็น Position Method “searchPlaces()” จะดึงข้อมูลของสถานที่ ที่ค้นหา มาเก็บไว้ในตัวแปร ที่มีชนิดข้อมูลเป็น “List<PlaceSearch>” Method “selectedLocation” จะเก็บค่า location ที่ค้นหา ที่มีชนิดข้อมูลเป็น StreamController แบบ Geometry แล้วจะเซตค่า searchResults เท่ากับว่าง ในแต่ละ Method จะมีพารามิเตอร์ ที่ได้จากการเรียกใช้ Google Place API ดังภาพประกอบที่ 3.44 การค้นหาสถานที่ด้วย Google Place API

```

9   Future<List<PlaceSearch>> getAutocomplete(String input) async {
10      var url =
11          'https://maps.googleapis.com/maps/api/place/autocomplete/json?input=$input&types=geocode&key=$apiKey';
12      var response = await http.get(Uri.parse(url));
13      var json = convert.jsonDecode(response.body);
14      var jsonResult = json['predictions'] as List;
15      return jsonResult.map((place) => PlaceSearch.fromJson(place)).toList();
16  }
17
18  Future<Geometry> getPlace(String input) async {
19      var url =
20          'https://maps.googleapis.com/maps/api/place/details/json?place_id=$input&key=$apiKey';
21      var response = await http.get(Uri.parse(url));
22      var json = convert.jsonDecode(response.body);
23      var jsonResult = json['result'] as Map<String, dynamic>;
24      return Geometry.fromJson(jsonResult);
25  }
26  }
27

```

ภาพประกอบที่ 3.40 การค้นหาสถานที่ด้วย Google Place API

ภาพประกอบที่ 3.44 การค้นหาสถานที่ด้วย Google Place API เป็นการทำงานในการดึงข้อมูลมาจาก Google Place API จะมี 2 Method คือ “getAutocomplete” ที่มีพารามิเตอร์เป็นสตริงใช้ในการค้นหาสถานที่ โดยจะเรียกใช้ Google Place API จะมีพารามิเตอร์ดังนี้ คือ สถานที่ และ apiKey ที่ได้จาก Google API ระบบจะ filter คำว่า “predictions” แล้ว แปลงเป็นชนิดข้อมูลเป็น List แล้วส่งค่าที่แปลงออกมาเป็น List Method “getPlace” มีพารามิเตอร์เป็นสตริงใช้ในการค้นหารายละเอียดสถานที่ โดยจะทำเหมือนกันกับ Method “getAutocomplete” ต่างกันตรงที่พารามิเตอร์ ซึ่งจะรับ PlaceID แทน เพื่อจะค้นหารายละเอียดสถานที่ จาก Google Place API แล้วทำการ filter คำว่า “result” แล้ว แปลงเป็นชนิดข้อมูลเป็น “Map<String, dynamic>” และ ส่งค่าที่แปลงออกมาเป็น JSON

### 3.9.2.7 การชำระเงินผ่าน OmiseAPI

```

56 OmiseFlutter omise = OmiseFlutter('pkey_test_5r6ejyowihj88k82b4');
69 exampleCreateSource() async {
70 // See Omise API documentation for details
71 // https://www.omise.co/sources-api
72 final source =
73   await omise.source.create(10000, "thb", "internet_banking_bay");
74   String basicAuth = 'Basic ' +
75     base64Encode(utf8.encode('skey_test_5olayop35lp6i4dydu:'));
76
77   Map<String, String> headerMap = {};
78   headerMap['authorization'] = basicAuth;
79   headerMap['Cache-Control'] = 'no-cache';
80   headerMap['Content-Type'] = 'application/x-www-form-urlencoded';
81
82   Map<String, String> data = {};
83   data['amount'] = '1000';
84   data['currency'] = 'thb';
85   data['card'] = source.id.toString();
86
87   var response = await http.post(Uri.parse('https://api.omise.co/charges'),
88     headers: headerMap, body: data);
89
90   print(response.body);
91 }

```

#### ภาพประกอบที่ 3.41 การชำระเงินผ่าน OmiseAPI

ภาพประกอบที่ 3.45 การชำระเงินผ่าน OmiseAPI เป็นการทำงานในการชำระเงินผ่าน OmiseAPI โดยจะต้องนำ Public key มาใช้เป็นพารามิเตอร์ ซึ่งจะได้จากการสมัครสมาชิกของ Omise ในการชำระเงินนั้น จะต้องใช้ webservice “Charges” ของ Omise โดยการใช้ Method Post ในการ Request ข้อมูลก่อนที่จะใช้งานได้จะต้องทำการสร้าง Token ในการติดต่อกับ OmiseAPI ขึ้นมาก่อน โดยจะใช้ Method “Source” ในการสร้าง Token จึงจะสามารถ เรียก webservice “Charges” ได้

### 3.9.2.8 การตรวจสอบหมายเลขบัตรประจำตัวประชาชน

```

1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019

```

```

final splitted = idCardString.split('');
int sum = 0;
for (var i = splitted.length, j = 0; i > 1; i--, j++) {
  var x = int.parse(splitted[j]) * i;
  sum = sum + x;
}
int digit = 11 - (sum % 11);
String sumDigit = digit.toString().substring(digit.toString().length - 1);
if (sumDigit != splitted.last) {
  errorProcess("เลขบัตรประชาชนไม่ถูกต้อง ", ColorsBuild.Red);
  return;
}

```

#### ภาพประกอบที่ 3.42 การตรวจสอบหมายเลขบัตรประจำตัวประชาชน

ภาพประกอบที่ 3.45 การตรวจสอบหมายเลขบัตรประจำตัวประชาชนในการตรวจสอบเลขที่บัตรประชาชนนั้นทำได้โดยการใช้ Check Digit ซึ่งใช้ตัวเลขหลักสุดท้ายในการตรวจสอบ Algorithm ของวิธีการใช้ Check Digit มีดังนี้

ตัวเลขบนบัตรประชาชนจะมีทั้งหมด 13 หลัก นำเลขใน 12 หลักแรก มาคูณกับเลขประจำตำแหน่ง (เลขประจำหลักได้แก่ 13 บวก 1 ลบด้วยตำแหน่งที่) จะได้ตัวเลขประจำตำแหน่ง จากนั้นเอาผลคูณของทั้ง 12 หลักมารวมกัน แล้วหารเอาเศษด้วย 11 เอาเศษที่ได้จากการหารในข้อ 2 มาลบด้วย 11 เท่านั้นก็ได้เลขที่เป็น Check Digit แล้ว (ถ้าผลจากข้อ 2 ได้ 10 ให้เอาเลขหลักหน่วยเป็น Check Digit ก็คือ 0 )

### 3.10 แบบประเมินระบบ

การใช้งานระบบสามารถวิเคราะห์ความพึงพอใจของผู้ใช้งานระบบตามเกณฑ์ที่กำหนด โดยแบ่งแบบประเมินออกเป็น 5 ส่วน

ส่วนที่ 1 ข้อมูลทั่วไป

ส่วนที่ 2 แบบประเมินด้านความต้องการของผู้ใช้ระบบ

ส่วนที่ 3 แบบประเมินด้านการทำงานได้ตามฟังก์ชันงานของระบบ

ส่วนที่ 4 แบบประเมินด้านความพึงพอใจต่อการใช้งานระบบ

ส่วนที่ 5 สิ่งที่คุณประเมินต้องการแนะนำเพิ่มเติม

คำชี้แจง ให้ทำเครื่องหมาย ✓ ลงในช่องแบบประเมินที่ตรงกับระดับความคิดเห็นของท่าน มากที่สุด

ส่วนที่ 1 ข้อมูลทั่วไป

- เพศ  ชาย  หญิง
- อายุ  น้อยกว่า 18 ปี  18-25 ปี  
 25-30 ปี  30-40 ปี  
 40 ปีขึ้นไป
- อาชีพ  นักเรียน/นักศึกษา  ครู/อาจารย์  
 บุคคลทั่วไป  อื่นๆ .....

ส่วนที่ 2 แบบประเมินด้านความต้องการของผู้ใช้ระบบ รายละเอียดดังตาราง

ตารางที่ 3.15 แบบประเมินด้านความต้องการของระบบ

ลำดับ	รายการประเมิน	ระดับความพึงพอใจ				
		5	4	3	2	1
1	ระบบสามารถสมัครสมาชิกได้					
2	ระบบสามารถเข้าสู่ระบบได้					
3	ระบบสามารถแก้ไขข้อมูลส่วนตัวได้					



ตารางที่ 3.15 แบบประเมินด้านความต้องการของระบบ (ต่อ)

ลำดับ	รายการประเมิน	ระดับความพึงพอใจ				
		5	4	3	2	1
4	ระบบสามารถค้นหาสถานที่ท่องเที่ยวได้					
5	ระบบสามารถสร้างทริปได้					
6	ระบบสามารถจองทริปได้					
7	ระบบสามารถชำระเงินได้					
8	ระบบสามารถออกใบเสร็จได้					

ส่วนที่ 3 แบบประเมินการทำงานได้ตามฟังก์ชันงานของระบบ รายละเอียดดังตาราง

ตารางที่ 3.16 แบบประเมินด้านการทำงานได้ตามฟังก์ชันงานของระบบ

ลำดับ	รายการประเมิน	ระดับความพึงพอใจ				
		5	4	3	2	1
1	ความถูกต้องในการสมัครสมาชิก					
2	ความถูกต้องในการเข้าสู่ระบบ					
3	ความถูกต้องในการแก้ไขข้อมูลส่วนตัว					
4	ความถูกต้องในการค้นหาสถานที่ท่องเที่ยว					
5	ความถูกต้องในการสร้างทริป					
6	ความถูกต้องในการจองทริป					
7	ความถูกต้องในการชำระเงิน					
8	ความถูกต้องในการออกใบเสร็จ					

ส่วนที่ 4 แบบประเมินด้านความพึงพอใจต่อการใช้งานระบบ รายละเอียดดังตาราง

ตารางที่ 3.17 แบบประเมินด้านความพึงพอใจต่อการใช้งานระบบ

ลำดับ	รายการประเมิน	ระดับความพึงพอใจ				
		5	4	3	2	1
1	ระบบใช้งานง่าย					
2	ขนาดตัวอักษร สีตัวอักษร รูปแบบตัวอักษรอ่านได้ง่าย					
3	ความเหมาะสมในการใช้สีของแอปและตัวอักษร					

ตารางที่ 3.17 แบบประเมินด้านความพึงพอใจต่อการใช้งานระบบ(ต่อ)

ลำดับ	รายการประเมิน	ระดับความพึงพอใจ				
		5	4	3	2	1
4	ความเหมาะสมในการใช้ชนิดตัวอักษรบนหน้าจอ					
5	ความเหมาะสมในการเลือกใช้สัญลักษณ์หรือรูปภาพ					
6	ความเหมาะสมในการเลือกใช้ข้อความสื่อความหมาย					

ส่วนที่ 5 สิ่งที่คุณประเมินต้องการแนะนำเพิ่มเติม

.....

.....

.....

เกณฑ์การประเมิน

- 5 ตรงกับ ระดับความพึงพอใจ มากที่สุด
- 4 ตรงกับ ระดับความพึงพอใจ มาก
- 3 ตรงกับ ระดับความพึงพอใจ ปานกลาง
- 2 ตรงกับ ระดับความพึงพอใจ น้อย
- 1 ตรงกับ ระดับความพึงพอใจ น้อยที่สุด / ควรปรับปรุง

วิธีการคำนวณแบบประเมิน ค่าเฉลี่ยเลขคณิต

การหารผลรวมของข้อมูลทั้งหมดด้วยจำนวนข้อมูลทั้งหมดค่าเฉลี่ยเลขคณิตข้อมูลที่แจกแจงความถี่ โดยหาจากสูตร

$$\bar{X} = (W_1X_1 + W_2X_2 + W_3X_3 + \dots + W_nX_n) / N$$

เมื่อ  $\bar{X}$  = ค่าเฉลี่ย

W = จำนวนระดับความพึงพอใจ

X = ระดับความพึงพอใจ (ตั้งแต่ระดับ 1 ถึง 5)

N = จำนวนผู้ประเมิน

ตัวอย่างการคำนวณ

เมื่อ N = 87

$$\begin{aligned}\bar{X} &= ((15*5) + (19*4) + (17*3) + (19*2) + (15*1))/87 \\ &= 2.93\end{aligned}$$

ซึ่งเกณฑ์ในการวัดมีดังนี้

ค่าเฉลี่ย ตั้งแต่ 3.00 ดีมาก

ค่าเฉลี่ย 2.50 – 2.99 ดี

ค่าเฉลี่ย 2.00 – 2.49 พอใช้

ค่าเฉลี่ย 1.50 – 1.99 ควรปรับปรุง

ค่าเฉลี่ย น้อยกว่า 1.5 ควรปรับปรุงอย่างยิ่ง

ดังนั้นจากตัวอย่าง จะเห็นได้ว่าระบบมีค่าเฉลี่ยอยู่ในระดับดี ค่าเฉลี่ยรวมในแต่ละด้านคิดได้จากผลรวมค่าเฉลี่ยทุกข้อหารด้วยจำนวนทั้งหมด และค่าเฉลี่ยรวมของระบบคิดได้จากผลรวมค่าเฉลี่ยทุกด้านหารด้วยจำนวนด้านทั้งหมด