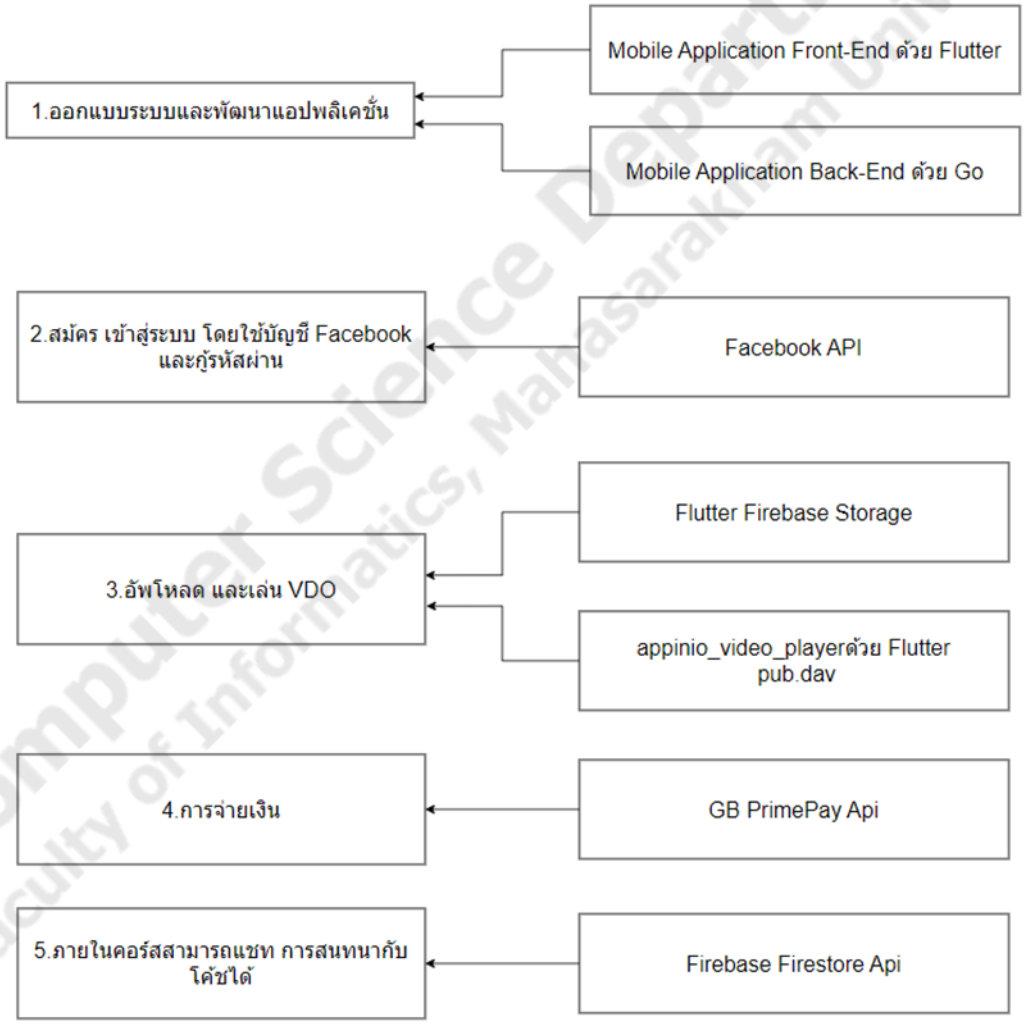
บทที่ 3

ขั้นตอนการดำเนินงาน

3.1 กรอบการดำเนินงาน

กรอบการทำงานนี้จะแสดงขั้นตอนการพัฒนาแอปพลิเคชันวางแผนการท่องเที่ยว ซึ่งมีขั้นตอน การทำงานหลักดังนี้



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบ

3.2 การออกแบบระบบ

คำอธิบาย

1. เขียนโปรแกรมออกแบบและพัฒนาแอปพลิเคชัน โค้ชออกกำลังกายสำหรับผู้ใช้งาน แอปพลิเคชันผ่านสมาร์ทโฟน โดยการสร้างแอปพลิเคชัน จะต้องมีการติดตั้งเครื่องมือดังนี้

- Visual Studio Code หรือ VS Code เป็นโปรแกรมประเภท Editor มีความสามารถในการแก้ไขและเขียนโค้ดเหมือนตัวอื่นๆ สามารถทำงานได้หลายแพลตฟอร์มทั้งวินโดวส์ แมค และลินิกซ์ รองรับได้หลายภาษา รวมถึงการติดตั้งเครื่องมือเสริมใช้งานได้ง่ายและฟรีในส่วนของภาษา Dart เอาไว้สำหรับสร้างโมเดลต่างๆไว้เก็บค่าข้อมูล

- Go เป็นภาษา Programming แบบ Open-Sourceโดยภาษา Go นั้นจะมีจุดเด่นในเรื่องของ Performance ที่สามารถทำงานได้อย่างรวดเร็วเทียบกับภาษาอื่น ๆ อีกทั้งยังมีจุดเด่นในเรื่องของ Simplicity ที่เน้นความง่ายในการเขียนและการอ่าน และยังสามารถทำ Concurrent Programming ได้ง่าย เพราะภาษา Golang ถูกออกแบบมาเพื่อทำให้ Application ที่ต้องใช้ Multi-Threading หรือ Distributed Systems เป็นเรื่องที่ง่ายขึ้น

2.การเข้าสู่ระบบโดย Facebook ในส่วนนี้ใช้ Facebook API ซึ่งเชื่อมต่อแอปพลิเคชันกับ Facebook ในสมาร์ทโฟนเพื่อติดต่อเรียกใช้ข้อมูลทำให้แอปพลิเคชันสามารถใช้งานการเข้าสู่ระบบผ่านบัญชี Facebook.

3.การอัพโหลดและเล่นวิดิโอ ในส่วนนี้จะใช้ Firebase storage เป็นบริการที่ให้คุณสามารถ Upload และ Download ไฟล์ได้อย่างมีประสิทธิภาพและปลอดภัยบน Google Cloud Storage พร้อมรองรับการขยายขนาดอัตโนมัติระดับ Petabyte ที่ให้คุณสามารถอัพโหลด รูปภาพ, ไฟล์เสียง, วิดีโอ และไฟล์อื่นๆได้ ซึ่งรองรับทั้ง Android

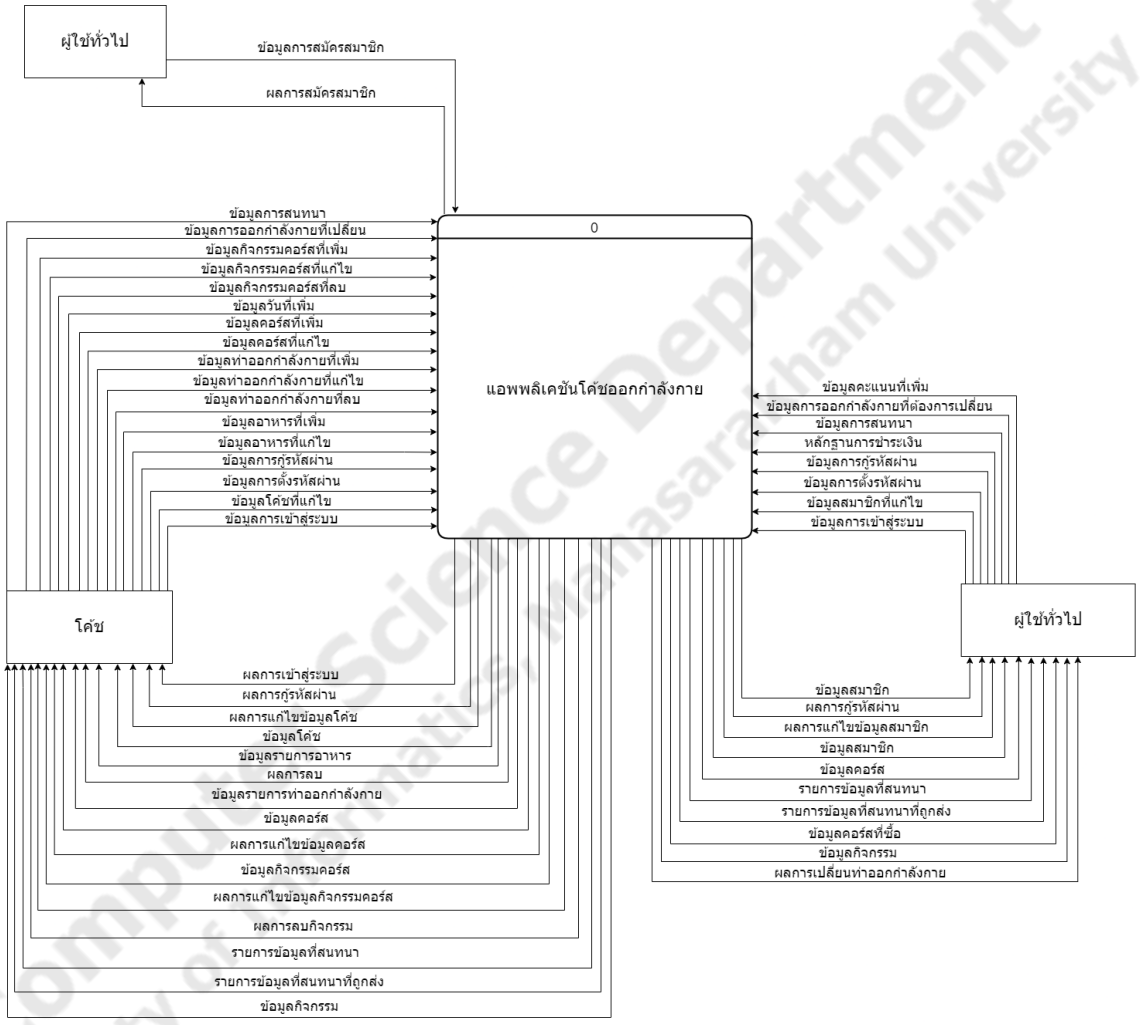
4.การจ่ายเงิน

- GB Prime Pay คือระบบรับชำระเงินออนไลน์ที่ช่วยให้ร้านค้าออนไลน์สามารถรับชำระเงินผ่านช่องทางต่างๆ ได้อย่างสะดวก รวดเร็ว และปลอดภัย

5.ภายในคอร์สสามารถแชท การสนทนากับโค้ชได้ ในส่วนนี้จะใช้ Firebase Firestore API เป็น API เป็น NoSQL Document Databaseที่เก็บข้อมูลในรูปแบบของ JSON และมีการ sync ข้อมูลแบบ

Realtimeกับทุก devices ที่เชื่อมต่อแบบอัตโนมัติในเสี้ยววินาทีรองรับการทำงานเมื่อ offline (ข้อมูลจะถูกเก็บไว้ใน local จนกระทั่งกลับมา online ก็จะทำการ sync ข้อมูลให้อัตโนมัติ)

3.3 แผนภาพบริบท (Context Diagram)

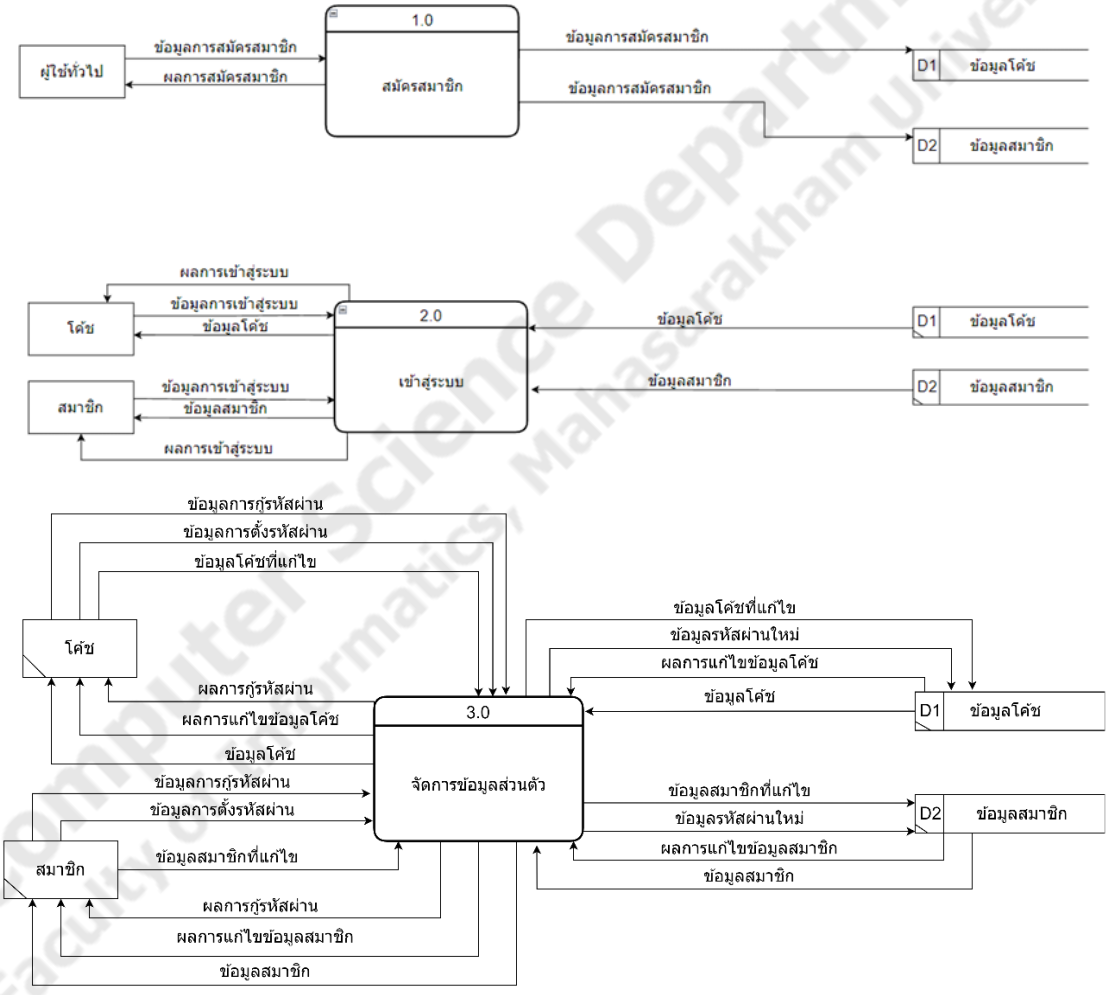


ภาพประกอบที่ 3.2 แผนภาพบริบท (Context Diagram)

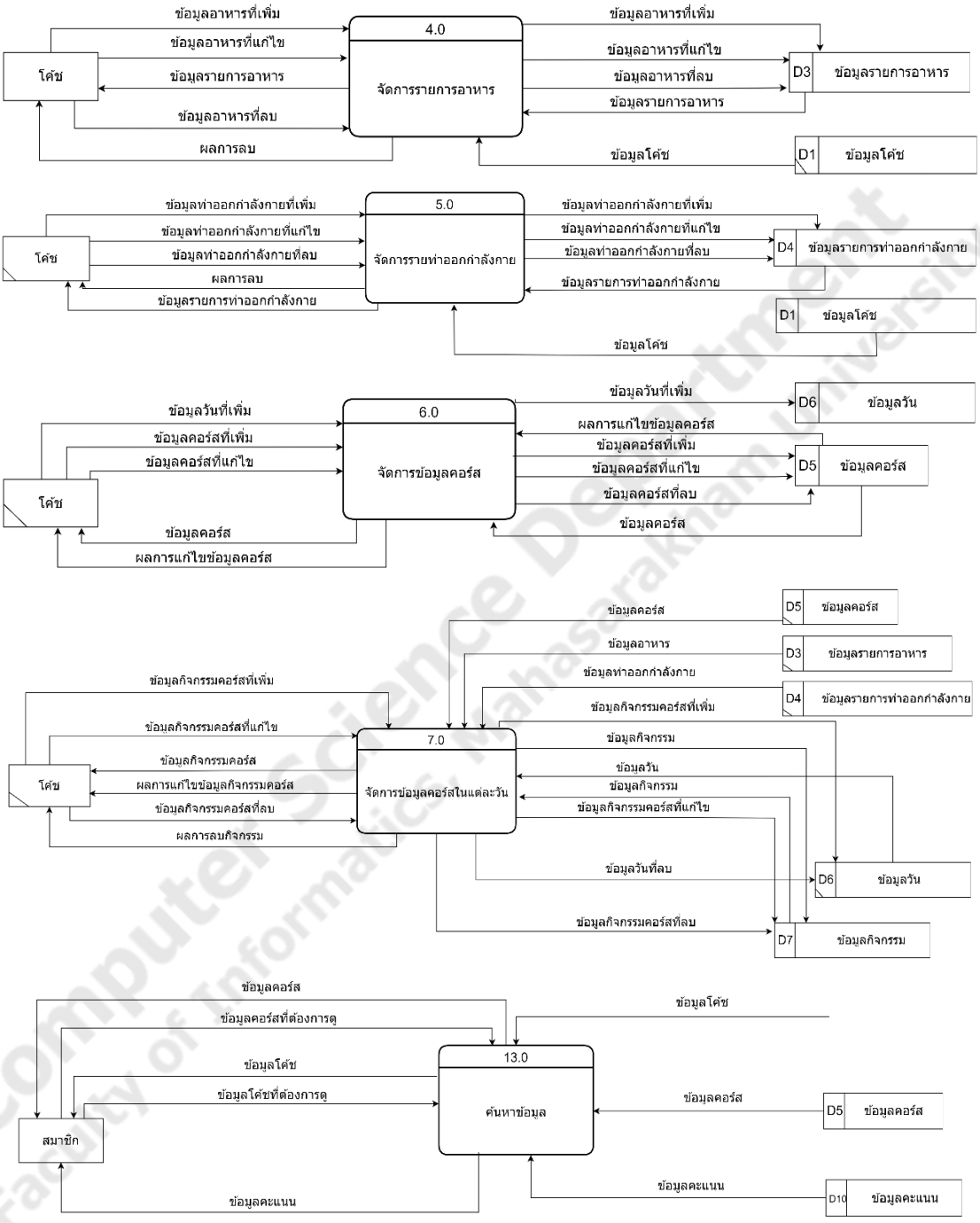
3.4 แผนภาพการไหลของข้อมูล (Data Flow Diagram)

3.4.1 Data Flow Diagram Level 1

แผนภาพกระแสข้อมูลในระดับที่แสดงขั้นตอนการทำงานหลักทั้งหมด (Process หลัก) ของระบบ แสดงทิศทางการไหลของ Data flow และแสดงรายละเอียดของแหล่งจัดเก็บข้อมูล (Data Store)



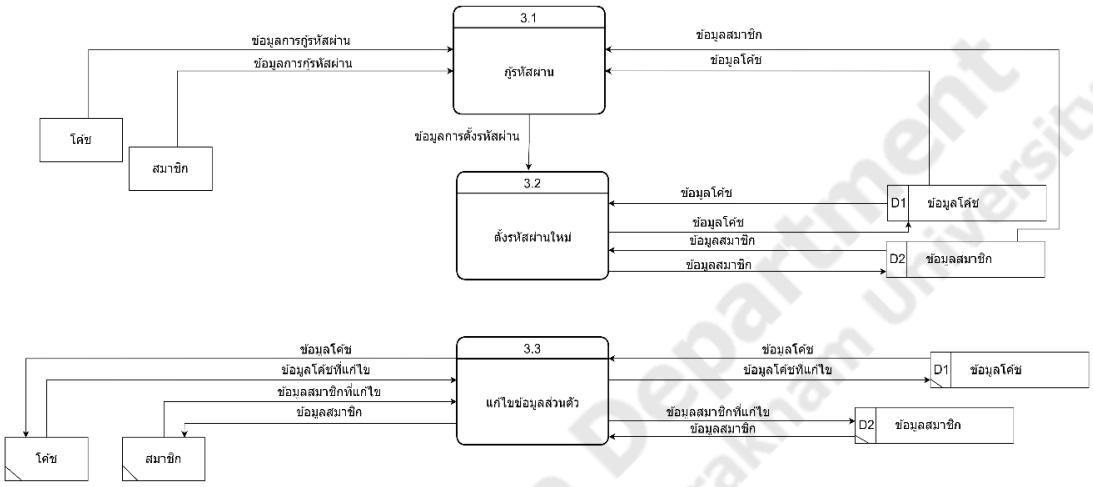
ภาพประกอบที่ 3.3 Data Flow Diagram Level 1



ภาพประกอบที่ 3.3 Data Flow Diagram Level 1 (ต่อ)

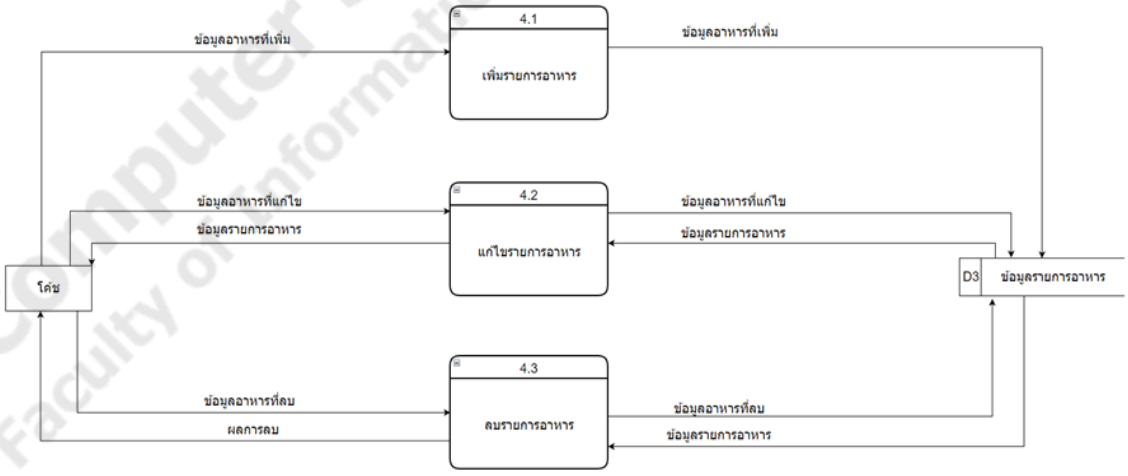
3.4.2 Data Flow Diagram Level 2

3.4.2.1 แผนภาพการไหลข้อมูลระดับ 2 กระบวนการ 3



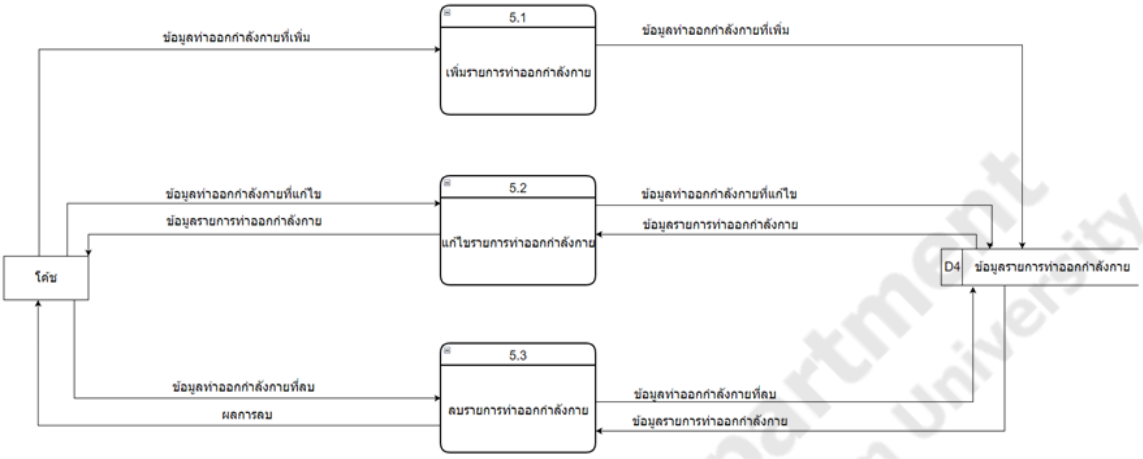
ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 Process 3

3.4.2.1 แผนภาพการไหลข้อมูลระดับ2 กระบวนการ 4



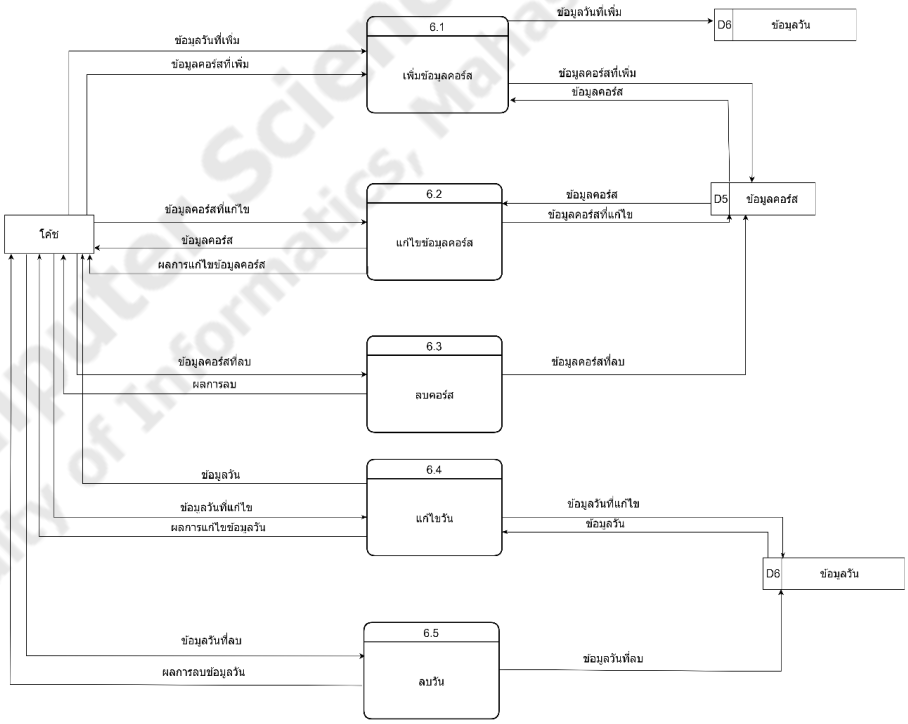
ภาพประกอบที่ 3.5 Data Flow Diagram Level 2 Process 4

3.4.2.2 แผนภาพการไหลข้อมูลระดับ2 กระบวนการ 5



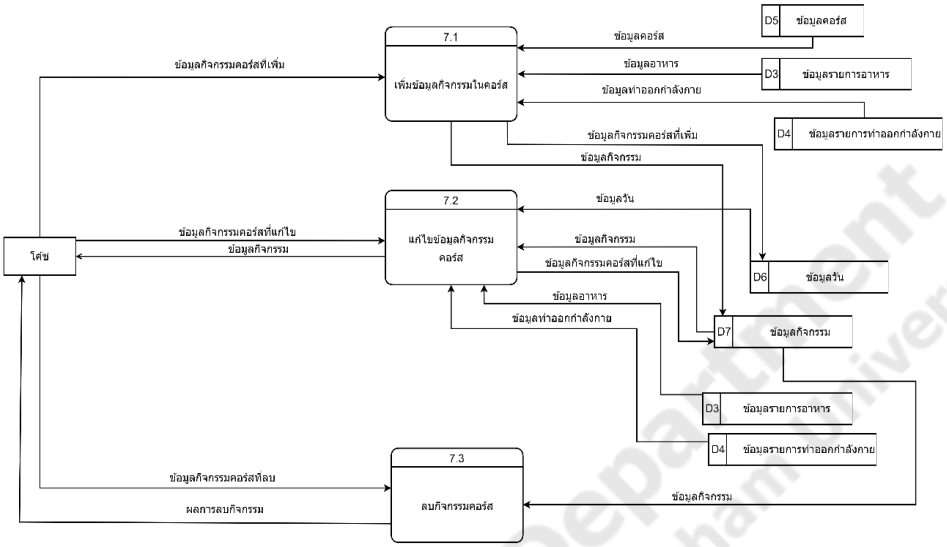
ภาพประกอบที่ 3.6 Data Flow Diagram Level 2 Process 5

3.4.2.1 แผนภาพการไหลข้อมูลระดับ2 กระบวนการ 6



ภาพประกอบที่ 3.7 Data Flow Diagram Level 2 Process 6

3.4.2.2 แผนภาพการไหลข้อมูลระดับ 2 กระบวนการ 7



ภาพประกอบที่ 3.8 Data Flow Diagram Level 2 Process 7

3.5 พจนานุกรมข้อมูล (Data Dictionary)

พจนานุกรมข้อมูลช่วยอธิบายการทำงานของแผนภาพการไหลของข้อมูล (Data Flow Dictionaries) รายละเอียดดังตารางที่ 3.1

ตารางที่ 3.1 External Entity Description

Name	Description	Input Data Flow	Output Data flow
ผู้ใช้	ผู้ใช้งานแอปพลิเคชันที่ยังไม่ได้ทำการสมัครเข้าใช้งานระบบ	- ข้อมูลสมาชิก	- ข้อมูลสมัครสมาชิก
โค้ช	ผู้ใช้งานแอปพลิเคชันที่สมัครสมาชิกแล้วและเลือกประเภทเป็นโค้ช	- ผลการเข้าสู่ระบบ - ผลการกู้รหัส - ข้อมูลโค้ช	- ข้อมูลการเข้าสู่ระบบ - ข้อมูลรหัสผ่าน - ข้อมูลการตั้งรหัสผ่าน

ตารางที่ 3.1 External Entity Description(ต่อ)

Name	Description	Input Data Flow	Output Data flow
		- ผลการแก้ไขข้อมูลโค้ช - ข้อมูลคอร์ส - ข้อมูลรายการอาหาร - ข้อมูลคลิปท่าออกกำลังกาย - ข้อมูลการออกกำลังกายที่เปลี่ยน -ข้อมูลกิจกรรมคอร์ส -ผลการแก้ไขข้อมูลกิจกรรมคอร์ส -ผลการลบกิจกรรม -ข้อมูลคอร์สที่ซื้อ -รายการข้อมูลที่สนทนา -รายการข้อมูลที่สนทนาที่ถูกส่ง	- ข้อมูลโค้ชที่แก้ไข - ข้อมูลคอร์สที่เพิ่ม - ข้อมูลคอร์สที่แก้ไข - ข้อมูลคอร์สที่ลบ - ข้อมูลวันที่เพิ่ม - ข้อมูลวันที่แก้ไข - ข้อมูลวันที่ลบ - ข้อมูลรายการอาหารที่เพิ่ม - ข้อมูลรายการอาหารที่แก้ไข - ข้อมูลรายการอาหารที่ลบ - ข้อมูลท่าออกกำลังกายที่เพิ่ม - ข้อมูลท่าออกกำลังกายที่แก้ไข - ข้อมูลท่าออกกำลังกายที่ลบ - ข้อมูลการสนทนา
สมาชิก	ผู้ใช้งานแอปพลิเคชันที่สมัครสมาชิกแล้วและเลือกประเภทเป็นสมาชิก	- ผลการเข้าสู่ระบบ - ผลการกู้รหัสผ่าน - ผลการแก้ไขข้อมูลสมาชิก - ข้อมูลสมาชิก - ข้อมูลคอร์สที่ซื้อ	- ข้อมูลการเข้าสู่ระบบ - ข้อมูลรหัสผ่าน - ข้อมูลการตั้งรหัสผ่าน - ข้อมูลสมาชิกที่แก้ไข - ข้อมูลหลักฐานการชำระเงิน

ตารางที่ 3.1 External Entity Description(ต่อ)

Name	Description	Input Data Flow	Output Data flow
		- รายการข้อมูลสมาชิก - รายการข้อมูลคอร์ส - รายการข้อมูลที่สนทนา - รายการข้อมูลที่สนทนาถูกส่ง - ข้อมูลการออกกำลังกายที่ต้องการเปลี่ยน - ข้อมูลกิจกรรม	- ข้อมูลการสนทนา - ข้อมูลการเปลี่ยนท่าออกกำลังกาย - ข้อมูลคะแนนที่เพิ่ม - ข้อมูลโค้ชที่ต้องการค้นหา

3.6 Store Description and Data Structure

ตารางที่ 3.2 Data Store Description and Data Structure

ID	Data Store Name	Description	Data Structure
D1	ข้อมูลโค้ช	เก็บรายละเอียดข้อมูลของโค้ช	รหัสโค้ช + รูป + ชื่อ-นามสกุลโทรศัพท์ วันเกิด + เบอร์โทรศัพท์+ ชื่อเรียกในระบบ+ อีเมล+ รหัสผ่าน+ เพศ + วุฒิการศึกษา+คุณสมบัติ
D2	ข้อมูลสมาชิก	เก็บรายละเอียดข้อมูลของสมาชิก	รหัสสมาชิก + รูป + ชื่อ-นามสกุล + วันเกิด + เบอร์โทรศัพท์ + Username + Email + Password + เพศ + น้ำหนัก + ส่วนสูง
D3	ข้อมูลรายการอาหาร	เก็บรายละเอียดของข้อมูลรายการอาหาร	รหัสเมนูอาหาร + รหัสโค้ช + ชื่อเมนูอาหาร + รูปอาหาร + ส่วนประสม + แคลอรี่

ตารางที่ 3.2 Data Store Description and Data Structure (ต่อ)

ID	Data Store Name	Description	Data Structure
D4	ข้อมูลรายการท่าออกกำลังกาย	เก็บรายละเอียดของข้อมูลรายการท่าออกกำลังกาย	รหัสท่าออกกำลังกาย + รหัสโค้ช + ชื่อท่าออกกำลังกาย + จำนวนครั้ง/เซต + วิดีโอ + คำอธิบายรายละเอียดท่าออกกำลังกาย
D5	ข้อมูลคอร์ส	เก็บรายละเอียดข้อมูลของคอร์ส	รหัสคอร์ส+ รหัสโค้ช + ชื่อคอร์ส + รูปภาพ + รายละเอียดคอร์ส + ระดับความยาก + จำนวนการรับ + จำนวนวัน + ราคา + สถานะการขาย + วันที่สินสุดคอร์ส
D7	ข้อมูลกิจกรรม	เก็บรายละเอียดข้อมูลของกิจกรรม	รหัสกิจกรรม + รหัสรายการอาหาร + รหัสท่าออกกำลังกาย
D8	ข้อมูลการชำระเงิน	เก็บรายละเอียดข้อมูลการชำระเงิน	รหัสการชำระเงิน + รหัสสมาชิก + วัน/เดือน/ปี + รูป
D9	ข้อมูลการสนทนา	เก็บรายละเอียดข้อมูลการสนทนา	รหัสการสนทนา + รหัสสมาชิก + รหัสการชำระเงิน + รหัสโค้ช + ข้อความ
D10	ข้อมูลคะแนน	เก็บรายละเอียดข้อมูลของคะแนน	รหัสคะแนน + รหัสสมาชิก + รหัสคอร์ส + รายละเอียด + คะแนน + น้ำหนัก

3.7 Data Flow Description and Data Structure

ตารางที่ 3.3 Data Flow Description and Data Structure

Name	Description	Source	Description	Data Structure
ข้อมูลสมัครสมาชิก	รายละเอียดข้อมูลการสมัครสมาชิก	ผู้ใช้ทั่วไป	Process 1.0 สมัครสมาชิก	[รูปภาพ + ชื่อ-นามสกุล + วันเกิด + เบอร์โทรศัพท์ + ชื่อ

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
		Process 1.0 สมัครสมาชิก	D1 ข้อมูลโค้ช	เรียกในระบบ+ อีเมล+ รหสผ่าน+ เพศ | บัญชี Facebook]
		Process 1.0 สมัครสมาชิก	D2 ข้อมูลสมาชิก	
ข้อมูลสมาชิก	รายละเอียดข้อมูลการสมัครสมาชิก	D2 ข้อมูลสมาชิก	Process 2.0 เข้าสู่ระบบ	รหัสสมาชิก + รูป + ชื่อ-นามสกุล + วันเกิด + เบอร์โทรศัพท์ + Username + Email + Password + เพศ + น้ำหนัก + ส่วนสูง
		Process 2.0 เข้าสู่ระบบ	สมาชิก	
		D2 ข้อมูลสมาชิก	Process 3.1 กู้รหัสผ่าน	
		D2 ข้อมูลสมาชิก	Process 3.2 ตั้งรหัสผ่านใหม่	
		D2 ข้อมูลสมาชิก แก้ไขข้อมูลส่วนตัว	Process 3.2 แก้ไขข้อมูลส่วนตัว	
		D2 ข้อมูลสมาชิก	Process 11.0 เพิ่มข้อมูลคะแนน	
ข้อมูลโค้ช	รายละเอียดข้อมูลการสมัครโค้ช	D1 ข้อมูลโค้ช	Process 2.0 เข้าสู่ระบบ	รหัสสมาชิก + รูป + ชื่อ-นามสกุล + วันเกิด + เบอร์โทรศัพท์
		Process 2.0	โค้ช	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
		เข้าสู่ระบบ		+ Username + Email + Password + เพศ + Qualification + Property
		D1 ข้อมูลโค้ช	Process 3.3 แก้ไขข้อมูลส่วนตัว	
		ตั้งรหัสผ่านใหม่	D1 ข้อมูลโค้ช	
		D2 ข้อมูลสมาชิก	Process 3.2 ตั้งรหัสผ่านใหม่	
		Process 3.2 ตั้งรหัสผ่านใหม่	D2 ข้อมูลสมาชิก	
ข้อมูลรายการอาหารที่แก้ไข	ข้อมูลของรายการอาหารที่โค้ชต้องการแก้ไข	โค้ช	Process 4.2 แก้ไขรายการอาหาร	ชื่อ + รูปอาหาร + ส่วนประสม + แคลอรี่
		Process 4.2 แก้ไขรายการอาหาร	D3 ข้อมูลรายการอาหาร	
ข้อมูลโค้ชที่แก้ไข	ข้อมูลของโค้ชที่ทำการแก้ไข	โค้ช	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสสมาชิก + รูป + ชื่อ-นามสกุล + วันเกิด + เบอร์โทรศัพท์ + Username + Email + Password + เพศ + Qualification + Property
ข้อมูลรายการอาหาร	ข้อมูรายการอาหารของโค้ช	D3 ข้อมูลรายการอาหาร	Process 4.2 แก้ไขรายการอาหาร	ชื่อ + รูปอาหาร + ส่วนประสม + แคลอรี่

ตารางที่ 3.3 Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
		Process 4.2 แก้ไขรายการอาหาร	โค้ช	
		D3 ข้อมูลรายการอาหาร	Process 4.3 ลบรายการอาหาร	
		Process 4.3 ลบรายการอาหาร	โค้ช	
		D3 ข้อมูลรายการอาหาร	Process 7.1 เพิ่มข้อมูลกิจกรรมในคอร์ส	
			Process 7.2 แก้ไขข้อมูลกิจกรรมคอร์ส	
		โค้ช	Process 7.3 แก้ไขข้อมูลภายในคอร์ส	
ข้อมูลรายการอาหารที่เพิ่ม	ข้อมูลของรายการอาหารที่โค้ชต้องการเพิ่ม	โค้ช	Process 4.1 เพิ่มรายการอาหาร	ชื่อ + รูปอาหาร + ส่วนประสม + แคลอรี่
		Process 4.1 เพิ่มรายการอาหาร	D3 ข้อมูลรายการอาหาร	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Data Structure	Data Structure
ข้อมูลรายการอาหารที่แก้ไข	ข้อมูลของรายการอาหารที่โค้ชต้องการแก้ไข	โค้ช	Process 4.2 แก้ไขรายการอาหาร	ชื่อ + รูปอาหาร + ส่วนประสม + แคลอรี่
		Process 4.2 แก้ไขรายการอาหาร	D3 ข้อมูลรายการอาหาร	
ข้อมูลรายการอาหารที่ลบ	ข้อมูลของรายการอาหารที่โค้ชต้องการลบ	โค้ช	Process 4.3 ลบรายการอาหาร	[ลบข้อมูลสำเร็จ | ลบข้อมูลไม่สำเร็จ]
		Process 4.3 ลบรายการอาหาร	D3 ข้อมูลรายการอาหาร	
		โค้ช	D3 ข้อมูลรายการอาหาร	
ข้อมูลรายการท่าออกกำลังกาย	ข้อมูรายการท่าออกกำลังกายของโค้ช	D4 ข้อมูลรายการท่าออกกำลังกาย	Process 5.2 แก้ไขรายการท่าออกกำลังกาย	ชื่อ + จำนวนครั้ง/เซต + วิดิโอ + รายละเอียด
		Process 5.2 แก้ไขรายการท่าออกกำลังกาย	โค้ช	
		D4 ข้อมูลรายการท่าออกกำลังกาย	Process 5.3 ลบรายการท่าออกกำลังกาย	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
		D4 ข้อมูลรายการท่าออกกำลังกาย	Process 7.2 แก้ไขข้อมูลกิจกรรมคอร์ส	
		D4 ข้อมูลรายการท่าออกกำลังกาย	Process 10.0 ร้องขอเปลี่ยนท่า	
		Process 10.0 ร้องขอเปลี่ยนท่า	สมาชิก	
			โค้ช	
		D4 ข้อมูลรายการท่าออกกำลังกาย	Process 12.0 การจัดการสถานะการออกกำลังกาย	
		Process 12.0 การจัดการสถานะการออกกำลังกาย	สมาชิก	
ข้อมูลรายการท่าออกกำลังกายที่เพิ่ม	ข้อมูลของรายการท่าออกกำลังกายที่โค้ชต้องการเพิ่ม	โค้ช	Process 5.1 เพิ่มรายการท่าออกกำลังกาย	ชื่อ + จำนวนครั้ง/เซต + วิดีโอ + รายละเอียด
		Process 5.1 เพิ่มรายการท่าออกกำลังกาย	D4 ข้อมูลรายการท่าออกกำลังกาย	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
		Process 5.2 แก้ไขรายการท่าออกกำลังกาย	D4 ข้อมูลรายการท่าออกกำลังกาย	
ข้อมูลรายการท่าออกกำลังกายที่ลบ	ข้อมูลของรายการท่าออกกำลังกายที่โค้ชต้องการลบ	โค้ช	Process 5.3 ลบรายการอาหาร	[ลบข้อมูลสำเร็จ | ลบข้อมูลไม่สำเร็จ]
		Process 5.3 ลบรายการอาหาร	D3 ข้อมูลรายการอาหาร	
ข้อมูลคอร์ส	ข้อมูลคอร์สของโค้ช	D5 ข้อมูลคอร์ส	Process 6.1 เพิ่มข้อมูลคอร์ส	ชื่อ + รายละอียด + ระดับความยาก + รูป +จำนวนวัน + ราคา + จำนวนการรับ +สถานะ + การเปิดขาย
			Process 6.2 แก้ไขข้อมูลคอร์ส	
		Process 6.2 แก้ไขข้อมูลคอร์ส	โค้ช	
		D5 ข้อมูลคอร์ส	Process 6.6 รายงานข้อมูลคอร์ส	
		D5 ข้อมูลคอร์ส	Process 7.1 เพิ่มข้อมูลกิจกรรมในคอร์ส	
			Process 8.0 ชำระเงิน	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
			Process 12.0 การจัดการสถานะการออกกำลังกาย	
ข้อมูลคอร์สที่เพิ่ม	ข้อมูลของคอร์สที่โค้ชต้องการเพิ่ม	โค้ช	Process 6.1 เพิ่มข้อมูลคอร์ส	ชื่อ + รายละอียด + ระดับความยาก + รูป +จำนวนวัน + ราคา + + จำนวนการรับ +สถานะ + การเปิดขาย
		Process 6.1 เพิ่มข้อมูลคอร์ส	D5 ข้อมูลคอร์ส	
		Process 6.1 เพิ่มข้อมูลคอร์ส	D6 ข้อมูลวัน	
ข้อมูลคอร์สที่แก้ไข	ข้อมูลของคอร์สที่โค้ชต้องการแก้ไข	โค้ช	Process 6.2 แก้ไขข้อมูลคอร์ส	ชื่อ + รายละอียด + ระดับความยาก + รูป + ราคา + สถานะการเปิดขาย
		Process 6.2 แก้ไขข้อมูลคอร์ส	D5 ข้อมูลคอร์ส	
ข้อมูลคอร์สที่ลบ	ข้อมูลของคอร์สที่โค้ชต้องการลบ	โค้ช	Process 6.3 ลบคอร์ส	[ลบข้อมูลสำเร็จ | ลบข้อมูลไม่สำเร็จ]
		Process 6.3 ลบคอร์ส	D5 ข้อมูลคอร์ส	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
ข้อมูลวัน	ข้อมูลวันของคอร์ส	D6 ข้อมูลวัน	Process 6.4 แก้ไขวัน	ลำดับวัน
		Process 6.4 แก้ไขวัน	โค้ช	
		D6 ข้อมูลวัน	Process 7.1 เพิ่มข้อมูลกิจกรรมในคอร์ส	
			Process 7.2 แก้ไขข้อมูลกิจกรรมคอร์ส	
			Process 10.0 ร้องขอเปลี่ยนท่า	
ข้อมูลคอร์สที่ลบวัน	ข้อมูลของคอร์สที่โค้ชต้องการลบวัน	โค้ช	Process 6.5 ลบวัน	[ลบข้อมูลสำเร็จ | ลบข้อมูลไม่สำเร็จ]
		Process 6.5ลบวัน	D6 ข้อมูลวัน	
รายงานข้อมูลคอร์ส	รายงานข้อมูลคอร์สที่สมาชิกต้องการดู	สมาชิก	Process 6.6 รายงานข้อมูลคอร์ส	ชื่อคอร์ส
		Process 6.6 รายงานคอร์ส	D5 ข้อมูลคอร์ส	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
ข้อมูลกิจกรรม	เก็บข้อมูลกิจกรรมของคอร์สในแต่ละวัน	D7 ข้อมูลกิจกรรม	Process 7.2 แก้ไขข้อมูลกิจกรรมคอร์ส	{รหัสกิจกรรม + รหัสรายการอาหาร + รหัสท่าออกกำลังกาย}
		Process 7.2 แก้ไขข้อมูลกิจกรรมคอร์ส	โค้ช	
		D7 ข้อมูลกิจกรรม	Process 7.3 ลบกิจกรรมคอร์ส	
		Process 7.3 ลบกิจกรรมคอร์ส	โค้ช	
		D7 ข้อมูลกิจกรรม	Process 10.0 ร้องขอเปลี่ยนท่า	
		Process 10 ร้องขอเปลี่ยนท่า	โค้ช	
		D7 ข้อมูลกิจกรรม	Process 10.0 ร้องขอเปลี่ยนท่า	
		Process 10 ร้องขอเปลี่ยนท่า	สมาชิก	
ข้อมูลกิจกรรมที่เพิ่ม	ข้อมูลกิจกรรมที่โค้ชเพิ่มในแต่ละวันของคอร์ส	โค้ช	Process 7.1 เพิ่มข้อมูลกิจกรรมในคอร์ส	รหัสกิจกรรม + รหัสรายการอาหาร + รหัสท่า

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
		Process 7.1 เพิ่มข้อมูลกิจกรรมในคอร์ส	D7 ข้อมูลกิจกรรม	
แก้ไขข้อมูลกิจกรรมคอร์ส	รายละเอียดข้อมูลที่โค้ชต้องการแก้ไข	โค้ช	Process 7.2 แก้ไขข้อมูลกิจกรรมคอร์ส	รหัสกิจกรรม + รหัสรายการอาหาร + รหัสท่าออกกำลังกาย
		Process 7.2 แก้ไขข้อมูลกิจกรรมคอร์ส	D7 ข้อมูลกิจกรรม	
ข้อมูลกิจกรรมคอร์สที่ลบ	ข้อมูลกิจกรรมที่โค้ชต้องการลบ	โค้ช	Process 7.3 ลบกิจกรรมคอร์ส	รหัสกิจกรรม + ชื่อกิจกรรรม
		Process 7.3 ลบกิจกรรมคอร์ส	D7 ข้อมูลกิจกรรม	
ข้อมูลการชำระเงิน	รายละเอียดข้อมูลการชำระเงิน	D8 ข้อมูลการชำระเงิน	Process 8.0 ชำระเงิน	{รหัสการชำระเงิน + รหัสสมาชิก + วัน/เดือน/ปี + รูป}
		Process 8.0 ชำระเงิน	สมาชิก	
		D8 ข้อมูลการชำระเงิน	Process 11.0 เพิ่มข้อมูลคะแนน	
		Process 11.0 เพิ่มข้อมูลคะแนน	สมาชิก	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
ชำระเงิน	หลักฐานการชำรพเงินที่เพิ่มโดยสมาชิก	สมาชิก	Process 8.0 ชำระเงิน	รหัสการชำระเงิน + รหัสสมาชิก + วัน/เดือน/ปี + รูป
		Process 8.0 ชำระเงิน	D8 ข้อมูลการชำระเงิน	
ข้อมูลการสนทนา	ข้อมูลสนทนา	โค้ช	Process 9.0 สนทนา	รหัสการสนทนา + รหัสสมาชิก + รหัสการชำระเงิน + รหัสโค้ช + ข้อความ
		Process 9.0 สนทนา	D9 ข้อมูลการสนทนา	
		สมาชิก	Process 9.0 สนทนา	
		Process 9.0 สนทนา	D9 ข้อมูลการสนทนา	
รายการข้อมูลที่สนทนา	ข้อมูลสนทนา	D9 ข้อมูลการสนทนา	Process 9.0 สนทนา	รหัสการสนทนา + รหัสสมาชิก + รหัสการชำระเงิน + รหัสโค้ช + ข้อความ
		Process 9.0 สนทนา	โค้ช	
		D9 ข้อมูลการสนทนา	Process 9.0 สนทนา	
		Process 9.0 สนทนา	สมาชิก	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

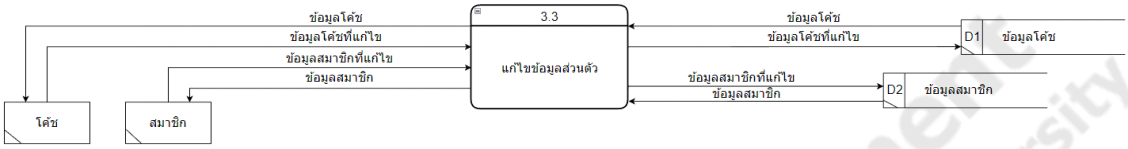
Name	Description	Source	Data Structure	Data Structure
		Process 9.0 สนทนา	โค้ช	รหัสการสนทนา + รหัสสมาชิก + รหัสการชำระเงิน + รหัสโค้ช + ข้อความ
ข้อมูลคอร์สที่ซื้อ	รายละเอียดข้อมูลการชำระเงิน	D8 ข้อมูลการชำระเงิน	Process 9.0 สนทนา	รหัสการชำระเงิน + รหัสสมาชิก + วัน/เดือน/ปี + รูป
		Process 9.0 สนทนา	โค้ช	
		D8 ข้อมูลการชำระเงิน	Process 9.0 สนทนา	
		Process 9.0 สนทนา	สมาชิก	
ข้อมูลการออกกำลังกายที่ต้องการเปลี่ยน	ข้อมูลการออกกำลังกายที่สมาชิกต้องการเปลี่ยน	สมาชิก	Process 10.0 ร้องขอเปลี่ยนท่า	ชื่อ + จำนวนครั้ง/เซต + วิดีโอ + รายละเอียดท่าออกกำลังกาย
		Process 10.0 ร้องขอเปลี่ยนท่า	D4 ข้อมูลรายการท่าออกกำลังกาย	
ข้อมูลคะแนนที่เพิ่ม	รายละเอียดข้อมูลคะแนนที่เพิ่ม	สมาชิก	Process 11.0 เพิ่มข้อมูลคะแนน	รหัสสมาชิก + รหัสคอร์ส + รายละเอียด + คะแนน + น้ำหนัก
		Process 11.0 เพิ่มข้อมูลคะแนน	D10 ข้อมูลคะแนน	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Description	Data Structure
เพิ่มข้อมูลสถานะ	รายละเอียดเพิ่มข้อมูลสถานะ	สมาชิก	Process 12.0 การจัดการสถานะการออกกำลังกาย	รหัสคอร์ส + รหัสท่าออกกำลังกาย + สถานะการออกกำลังกาย
		Process 12.0 การจัดการสถานะการออกกำลังกาย	D4 ข้อมูลรายการท่าออกกำลังกาย	
ค้นหาข้อมูล	รายละเอียดการค้นหาข้อมูล	สมาชิก	Process 13.0 การค้นหาข้อมูล	[ชื่อคอร์ส | ชื่อโค้ช]

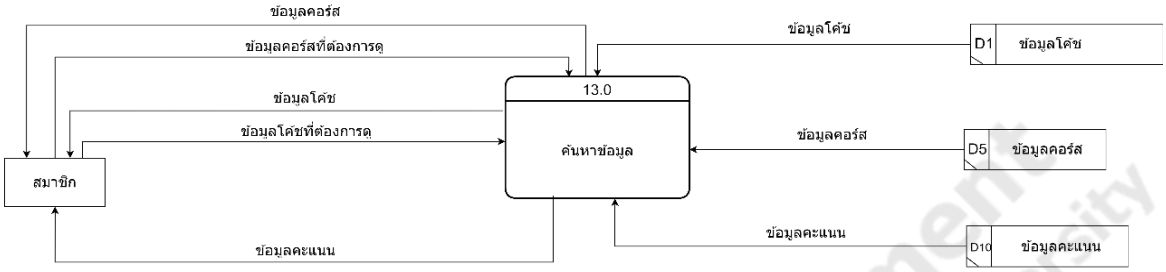
3.8 คำอธิบายการประมวลผล (Process Description)

3.8.1 Process แก้ไขข้อมูลส่วนตัว



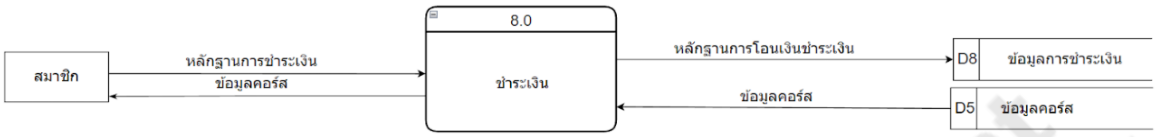
Number	3.3
Name	แก้ไขข้อมูลส่วนตัว
Description	การแก้ไขข้อมูลส่วนตัวของสมาชิก
Input data flow	- ข้อมูลสมาชิกที่แก้ไข - ข้อมูลสมาชิก - ข้อมูลโค้ช
Output data flow	- ข้อมูลโค้ช - ข้อมูลสมาชิก - ข้อมูลสมาชิกที่แก้ไข
Process description	เริ่มต้น 1. กดไอคอนแก้ไขข้อมูลส่วนตัวนั้นๆ ในหน้าแสดงข้อมูลส่วนตัว 2.ดึงข้อมูลมาเปรียบเทียบ 3.กรอกข้อมูลส่วนตัวที่ต้องการแก้ไข 4. ตรวจสอบว่าข้อมูลส่วนตัวถูกต้องหรือไม. ถ้า (ถูกต้อง) นำข้อมูลส่วนตัวใหม่ไปอัพเดตในแฟ้มข้อมูลสมาชิก ถ้า (ไม่ถูกต้อง) แจ้งเตือนการแก้ไขรายการอาหารไม่สำเร็จ จบการทำงาน

3.8.2 Process รายงานข้อมูลคอร์ส



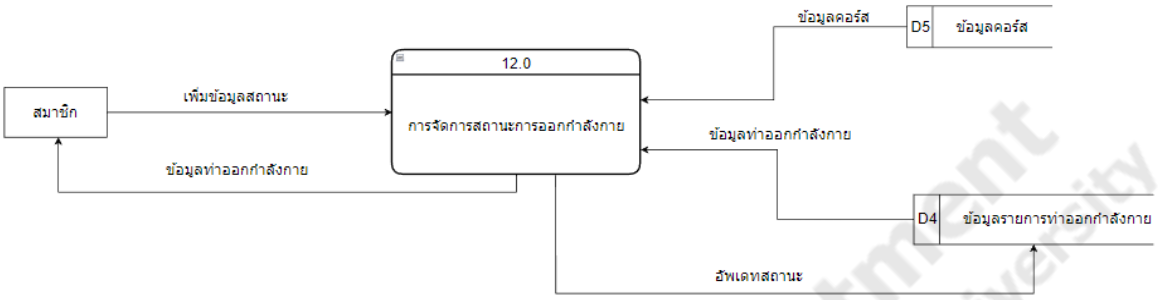
Number	13.0
Name	รายงานข้อมูลคอร์ส
Description	การแสดงข้อมูลคอร์ส
Input data flow	- ข้อมูลคอร์สที่ต้องการดู - ข้อมูลคอร์ส
Output data flow	- ข้อมูลคอร์ส
Process description	เริ่มต้น 1.เข้าหน้ารายการแสดงคอร์ส 2.เลือกคอร์สที่ต้องการดู 3.ดึงข้อมูลจากแฟ้มข้อมูลออกมาแสดงผล จบการทำงาน

3.8.3 Process การชำระเงินของสมาชิก



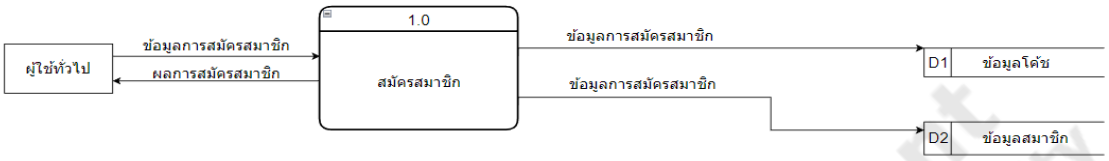
Number	8.0
Name	ชำระเงิน
Description	การชำระเงินของสมาชิก
Input data flow	- หลักฐานการชำระเงิน - ข้อมูลคอร์ส
Output data flow	- ข้อมูลคอร์ส - หลักฐานการชำระเงิน
Process description	เริ่มต้น 1.ดึงข้อมูลคอร์ส 2.เพิ่มหลักฐานการโอนเงินชำระเงิน 2.1.ถ้าเพิ่มหลักฐานการโอนเงินครบถ้วน จะสามารถทำขั้นตอนต่อไปได้ 2.2.ถ้าเพิ่มหลักฐานการโอนเงินไม่ครบถ้วน จะไม่สามารถข้ามทำขั้นตอนต่อไปได้ จบการทำงาน

3.8.4 Process การจัดการสถานะการออกกำลังกาย



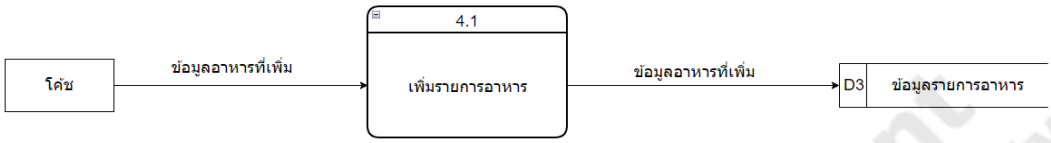
Number	12.0
Name	การจัดการสถานะการออกกำลังกาย
Description	การจัดการสถานะการออกกำลังกาย
Input data flow	- ข้อมูลคอร์ส - ข้อมูลท่าออกกำลังกาย - เพิ่มข้อมูลสถานะ
Output data flow	- ข้อมูลท่าออกกำลังกาย - อัพเดทสถานะ
Process description	เริ่มต้น 1. กดไอคอนอัพเดทสถานะท่าออกกำลังกายนั้นๆ ในหน้าแสดงข้อมูลท่าออกกำลังกาย 2.ดึงข้อมูลมาเปรียบเทียบว่าเคยอัพเดทสถานะหรือไม. 3. ตรวจสอบว่าข้อมูลสถานะถูกอัพเดทแล้วหรือไม. ถ้า (ยังไม่ถูกอัพเดท) นำข้อมูลสถานะใหม่ไปอัพเดตในแฟ้มข้อมูลรายการท่าออกกำลังกาย ถ้า (ถูกอัพเดทแล้ว) จบการทำงาน

3.8.5 Process สมัครสมาชิก



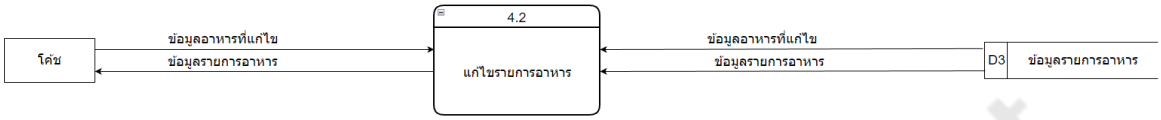
Number	1.0
Name	สมัครสมาชิก
Description	การสมัครสมาชิก
Input data flow	- ข้อมูลการสมัครสมาชิก
Output data flow	- ผลการสมัครสมาชิก - ข้อมูลการสมัครสมาชิก
Process description	เริ่มต้น 1. กรอกข้อมูลการสมัครสมาชิก 2. ตรวจสอบรายละเอียดข้อมูลการสมัครสมาชิก ถ้า (ถูกต้องและครบถ้วน) บันทึกคำขอลงแฟ้มข้อมูลผู้ใช้ ถ้า (ไม่ถูกต้องและไม่ครบถ้วน) แจ้งเตือนข้อมูลผิดพลาด ตรวจสอบข้อมูลให้ถูกต้อง 3. รับผลการสมัครสมาชิกและข้อมูลสมาชิก จบการทำงาน

3.8.6 Process เพิ่มรายการอาหาร



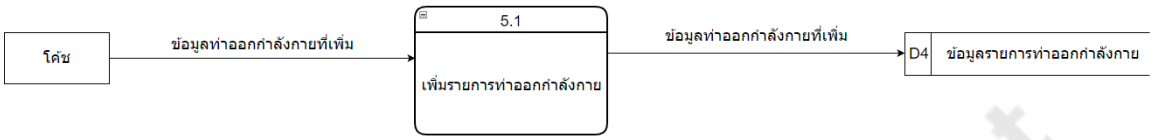
Number	4.1
Name	เพิ่มรายการอาหาร
Description	การเพิ่มรายการอาหาร
Input data flow	- ข้อมูลอาหารที่เพิ่ม
Output data flow	- ข้อมูลอาหารที่เพิ่ม
Process description	เริ่มต้น 1. กดปุ่มสร้างรายกายอาหาร 2. เพิ่มข้อมูลของรายการอาหาร และกดยืนยัน 3. ตรวจสอบว่าข้อมูลอาหารว่าครบถ้วนและถูกต้องหรือไม่ ถ้า (ครบถ้วนและถูกต้อง) บันทึกข้อมูลอาหารลงในแฟ้มข้อมูลรายการอาหาร แจ้งเตือนการสร้างรายการอาหารสำเร็จ ถ้า (ไม่ครบถ้วนและไม่ถูกต้อง) แจ้งเตือนการสร้างรายการอาหารไม่สำเร็จ จบการทำงาน

3.8.7 Process แก้ไขรายการอาหาร



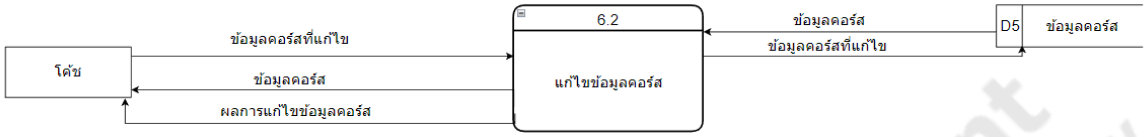
Number	4.2
Name	แก้ไขรายการอาหาร
Description	การแก้ไขรายการอาหาร
Input data flow	- ข้อมูลรายการอาหาร - ข้อมูลอาหารที่แก้ไข
Output data flow	- ข้อมูลรายการอาหาร - ข้อมูลอาหารที่แก้ไข
Process description	เริ่มต้น 1. กดไอคอนแก้ไขข้อมูลอาหารนั้นๆ ในหน้าแสดงข้อมูลอาหาร 2.ดึงข้อมูลมาเปรียบเทียบ 3.กรอกข้อมูลรายการอาหารที่ต้องการแก้ไข 3. ตรวจสอบว่าข้อมูลรายการอาหารถูกต้องหรือไม. ถ้า (ถูกต้อง) นำข้อมูลอาหารใหม่ไปอัพเดตในแฟ้มข้อมูลรายการ อาร หาร ถ้า (ไม่ถูกต้อง) แจ้งเตือนการแก้ไขรายการอาหารไม่สำเร็จ จบการทำงาน

3.8.8 Process เพิ่มรายการท่าออกกำลังกาย



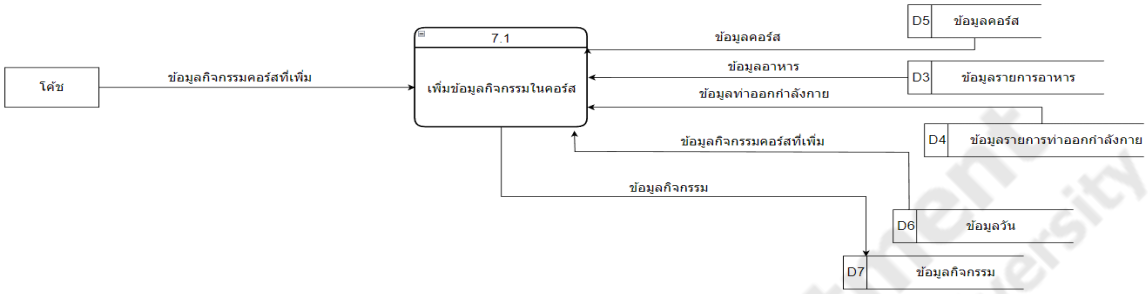
Number	5.1
Name	เพิ่มรายการท่าออกกำลังกาย
Description	การเพิ่มรายการท่าออกกำลังกาย
Input data flow	- ข้อมูลท่าออกกำลังกายที่เพิ่ม
Output data flow	- ข้อมูลท่าออกกำลังกายที่เพิ่ม
Process description	เริ่มต้น 1. กดปุ่มสร้างรายกายท่าออกกำลังกาย 2. เพิ่มข้อมูลของท่าออกกำลังกาย และกดยืนยัน 3. ตรวจสอบว่าข้อมูลท่าออกกำลังกายว่าครบถ้วนและถูกต้องหรือไม่ ถ้า (ครบถ้วนและถูกต้อง) บันทึกข้อมูลท่าออกกำลังกายลงในแฟ้มข้อมูลรายการท่าออกกำลังกาย แจ้งเตือนการสร้างท่าออกกำลังกายสำเร็จ ถ้า (ไม่ครบถ้วนและไม่ถูกต้อง) แจ้งเตือนการสร้างรายการท่าออกกำลังกายไม่สำเร็จ จบการทำงาน

3.8.9 Process แก้ไขข้อมูลคอร์ส



Number	6.2
Name	แก้ไขข้อมูลคอร์ส
Description	การแก้ไขข้อมูลคอร์ส
Input data flow	- ข้อมูลคอร์ส - ข้อมูลคอร์สที่แก้ไข
Output data flow	- ข้อมูลคอร์ส - ข้อมูลคอร์สที่แก้ไข - ผลการแก้ไขข้อมูลคอร์ส
Process description	เริ่มต้น 1. กดไอคอนแก้ไขข้อมูลคอร์สนั้นๆ ในหน้าแสดงข้อมูลคอร์ส 2.ดึงข้อมูลมาเปรียบเทียบ 3.กรอกข้อมูลคอร์สที่ต้องการแก้ไข 3. ตรวจสอบว่าข้อมูลคอร์สถูกต้องหรือไม. ถ้า (ถูกต้อง) นำข้อมูลคอร์สใหม่ไปอัพเดตในแฟ้มข้อมูลคอร์ส ถ้า (ไม่ถูกต้อง) แจ้งเตือนการแก้ไขคอร์สไม่สำเร็จ จบการทำงาน

3.8.10 Process เพิ่มข้อมูลกิจกรรมภายในคอร์ส



Number	7.1
Name	เพิ่มข้อมูลกิจกรรมภายในคอร์ส
Description	การเพิ่มข้อมูลกิจกรรมภายในคอร์ส
Input data flow	- ข้อมูลคอร์ส - ข้อมูลกิจกรรมคอร์สที่เพิ่ม - ข้อมูลอาหาร - ข้อมูลท่าออกกำลังกาย
Output data flow	- ข้อมูลกิจกรรม
Process description	เริ่มต้น 1. กดปุ่มสร้างกิจกรรมในคอร์ส 2. เพิ่มข้อมูลของกิจกรรมในคอร์ส และกดยืนยัน 3. ตรวจสอบว่าข้อมูลกิจกรรมในคอร์สว่าครบถ้วนและถูกต้องหรือไม่ ถ้า (ครบถ้วนและถูกต้อง) บันทึกข้อมูลท่าออกกำลังกายลงในแฟ้มข้อมูลกิจกรรมในคอร์ส แจ้งเตือนการสร้างกิจกรรมในคอร์สสำเร็จ ถ้า (ไม่ครบถ้วนและไม่ถูกต้อง) แจ้งเตือนการสร้างกิจกรรมในคอร์สไม่สำเร็จ จบการทำงาน

3.9 แผนภาพ Entity Relationship Diagram (ER-Diagram)

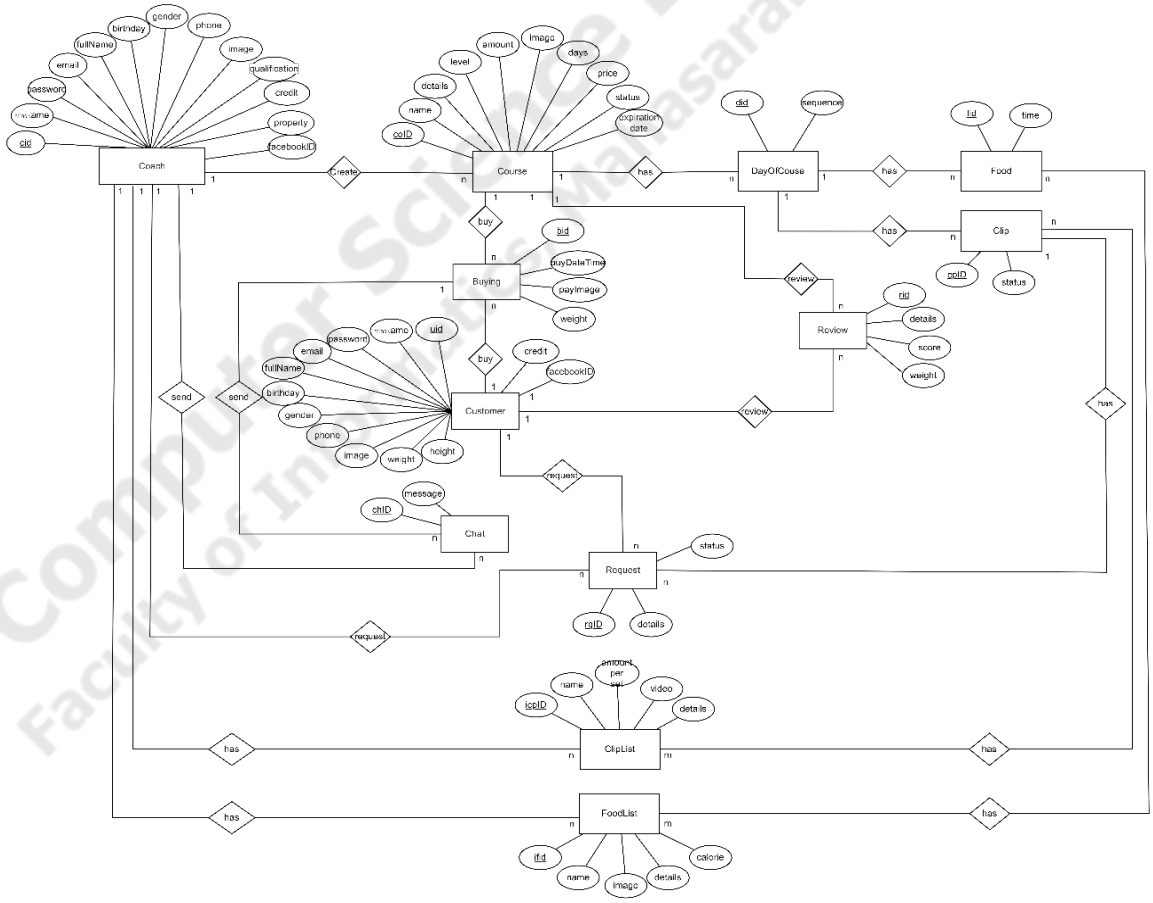
แผนภาพ Entity Relationship Diagram (ER-Diagram) เป็นแบบจำลองที่ใช้อธิบายโครงสร้างของฐานข้อมูลที่ใช้ในแอปพลิเคชัน ซึ่งเขียนออกมาในลักษณะของรูปภาพ การอธิบายโครงสร้างและความสัมพันธ์ของข้อมูล

แผนภาพ ER-Diagram ประกอบด้วยองค์ประกอบหลักสามประการ ได้แก่

1. Entity คือ วัตถุหรือสิ่งของที่เราสนใจในระบบงานนั้น ๆ เช่น พนักงาน หนังสือ หรืออาจจะเป็นสถานที่

2. Attribute คือ คุณสมบัติของวัตถุที่เราสนใจ เช่น ชื่อ นามสกุล อายุ หรือที่อยู่

3. Relationship คือ ความสัมพันธ์ระหว่างวัตถุ เช่น ความสัมพันธ์ระหว่างพนักงานกับแผนก หรือความสัมพันธ์ระหว่างหนังสือกับผู้เขียนในบทนี้ เราจะใช้แผนภาพ ER-Diagram เพื่ออธิบายโครงสร้างของฐานข้อมูลสำหรับแอปพลิเคชันฟิตเนส



ภาพประกอบที่ 3.9 แผนภาพ Entity Relationship Diagram

3.10 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ตารางสมาชิก (Customer)

Column	Type	Description	Example	Constraint
uid	Int	รหัสสมาชิก (ไม่สามารถแก้ไขได้)	1	PK
username	varchar(50)	ชื่อนามแฝงแทนตัวผู้ใช้	ปังปอนด์	Not null
password	varchar(50)	ชื่อนามแฝงแทนตัวผู้ใช้	abc1234	Not null
email	varchar(100)	อีเมล (ใช้เป็น Username เข้าสู่ระบบ)	Tpangpond@gmail.com	UNIQUE
fullName	varchar(100)	ชื่อ – นามสกุลผู้ใช้	สิรวิชญ์ เหล่าสีนาท	Not null
birthday	date	วันเกิด ผู้ใช้	2002-03-14	Not null
gender	varchar(1)	เพศ ผู้ใช้ (1=เพศชาย,2=เพศหญิง)	1	1,2 Not null
phone	varchar(10)	เบอร์โทรศัพท์ของผู้ใช้	0802280461	Not null
image	varchar(3000)	รูปโปรไฟล์แทนตัวผู้ใช้	(url)	Null
qualification	int	สถาศึกษาที่เรียนจบ	ปริญญาตรี คณะบริหารธุรกิจ มหาวิทยาลัยอัสสัมชัญ (ABAC)	Not null

ตารางที่ 3.4 ตารางสมาชิก (Customer) (ต่อ)

Column	Type	Description	Example	Constraint
weight	int	น้ำหนัก ผู้ใช้	51	Not null
height	int	ส่วนสูง ผู้ใช้	171	Not null
facebookID	varchar(100)	ไอดีสำหรับ Facebook	null	Null
price	int	จำนวนเงิน ผู้ใช้	0	Null

ตารางที่ 3.5 ตารางของโค้ช(Coach)

Column	Type	Description	Example	Constraint
cid	Int	รหัสโค้ช (ไม่สามารถแก้ไขได้)	1	PK
username	varchar(50)	ชื่อนามแฝงแทนตัวผู้ใช้	โค้ช เบเบ้	Not null
password	varchar(50)	ชื่อนามแฝงแทนตัวผู้ใช้	1234	Not null
email	varchar(100)	อีเมล (ใช้เป็น Username เข้าสู่ระบบ)	babe@gmail.com	UNIQUE
fullName	varchar(100)	ชื่อ – นามสกุล ผู้ใช้	ธันย์ชนก ฤทธินาคา	Not null
birthday	date	วันเกิด ผู้ใช้	2002-02-14	Not null

ตารางที่ 3.5 ตารางของโค้ช(Coach) (ต่อ)

Column	Type	Description	Example	Constraint
gender	varchar(1)	เพศ ผู้ใช้ (1=เพศชาย,2= เพศหญิง)	2	1,2 Not null
phone	varchar(10)	เบอร์โทรศัพท์ของผู้ใช้	0981235821	Not null
image	varchar(3000)	รูปโปรไฟล์แทนตัวผู้ใช้ (เพิ่มหรือไม่เพิ่มก็ได้)	(url)	Null
weight	int	น้ำหนัก ผู้ใช้	51	Not null
height	int	ส่วนสูง ผู้ใช้	171	Not null
qualification	int	สถาศึกษาที่โค้ชเรียนจบ	ปริญญาตรี คณะบริหารธุรกิจ มหาวิทยาลัยอัสสัมชัญ (ABAC)	Not null
property	int	รายละเอียดของโค้ช	(ชื่อเล่น: เบเบ้) เป็นเทรนเนอร์	Not null
facebookID	varchar(100)	ไอดีสำหรับ Facebook	null	Null
price	int	จำนวนเงิน ผู้ใช้	0	Null
nameCard	varchar(1000)	ชื่อบัญชีธนาคารของผู้ใช้	ธนาคารกรุงไทย	Not null

ตารางที่ 3.6 ตารางรายการอาหารของโค้ช(iFood)

Column	Type	Description	Example	Constraint
ifid	Int	รหัสของอาหาร (ไม่สามารถแก้ไขได้)	2	PK
cid	int	รหัสของโค้ช (ไม่สามารถแก้ไขได้)	1	FK reference from COACH (cid) ON DELETE CASCADE ON UPDATE CASCADE
name	varchar(100)	ชื่อของอาหาร	กระเพราหมู	Not null
image	varchar(3000)	รูปของอาหาร	(url)	Not null
details	varchar(1000)	รายละเอียดของอาหาร	หมูสับ 200 กรัม พริกกับกระเทียมสับ 2 ช้อนโต๊ะ กระชายหั่น ใบกะเพรา ซีอิ๊วขาว 1 ช้อนโต๊ะ น้ำมันหอย 1 ช้อนโต๊ะ น้ำตาลทราย 1 ช้อนชา ซีอิ๊วดำเล็กน้อย	Not null

ตารางที่ 3.7 ตารางรายการคลิปท่าออกกำลังกายของโค้ช(iClip)

Column	Type	Description	Example	Constraint
icpID	Int	รหัสของท่าออกกำลังกาย (ไม่สามารถแก้ไขได้)	1	PK
cid	int	รหัสของโค้ช (ไม่สามารถแก้ไขได้)	1	FK reference from COACH (cid) ON DELETE CASCADE ON UPDATE CASCADE
name	varchar(100)	ชื่อท่าออกกำลังกาย	ท่าเลกลันจ์	Not null
imageamount per set	varchar(50)	เซตท่าออกกำลังกาย	5 เซต	Not null
video	varchar(3000)	วิดีโอท่า	(url)	Not null
details	varchar(2000)	รายละเอียดท่า	ท่านี้ช่วยบริหารต้นขาด้านหน้า ก้น และกล้ามเนื้อแฮมสตริง ทำให้ขาและกันกระชับ กล้ามเนื้อขาเข็งแรง	Not null

ตารางที่ 3.8 ตารางคอร์ส(Course)

Column	Type	Description	Example	Constraint
coID	Int	รหัสคอร์ส(ไม่สามารถแก้ไขได้)	1	PK
cid	int	รหัสของโค้ช (ไม่สามารถแก้ไขได้)	1	FK reference from COACH (cid) ON DELETE CASCADE ON UPDATE CASCADE
bid	Int	รหัสของการสั่งซื้อ (ไม่สามารถแก้ไขได้)	null	FK reference from BUYING (bid) ON DELETE CASCADE ON UPDATE CASCADE , Null
name	varchar(50)	ชื่อคอร์ส	เผาผลาญทุกสัดส่วน	Not null
details	varchar(250)	รายละเอียดคอร์ส	โค้ชเบเบ้มีท่าออกกำลังกายง่ายๆที่บ้านมาฝาก เป็น5ท่าออกกำลังกายต่อหนึ่งวันที่จะช่วยเผาผลาญไขมันส่วนเกินทุก	Not null

ตารางที่ 3.8 ตารางคอร์ส(Course)(ต่อ)

Column	Type	Description	Example	Constraint
			ส่วนได้ในเวลาสั้นๆแต่ให้ประสิทธิภาพไม่แตกต่างจากการเล่นเฟิตเนส	
level	varchar(1)	ระดับความยากของคอร์ส (0=ง่าย,1=ปลานกลาง,2=ยาก)	2	0,1,2,Not null
amount	Int	จำนวนการจำกัดคนของคอร์ส	20	Not null
image	varchar(3000)	รูปของคอร์ส	(url)	Not Null
days	Int	จำนวนวันของคอร์ส	21	Not Null
price	Int	ราคาของคอร์ส	399	Not null
status	varchar(1)	การเปิดขายคอร์ส (0=ปิดการขายคอร์ส,1=เปิดขายคอร์ส)	1	0,1,NOT null
Expiration date	date	วันหมดอายุของคอร์ส	NULL	Null

ตารางที่ 3.9 ตารางวันในคอร์ส(DayOfCouse)

Column	Type	Description	Example	Constraint
did	Int	รหัสวัน (ไม่สามารถแก้ไขได้)	1	PK
coID	Int	รหัสคอร์ส(ไม่สามารถแก้ไขได้)	1	FK reference from COURSE (coID) ON DELETE CASCADE ON UPDATE CASCADE
sequence	Int	ลำดับวันในคอร์ส	2	Not null

ตารางที่ 3.10 ตารางอาหารในคอร์ส(Food)

Column	Type	Description	Example	Constraint
fid	Int	รหัสของอาหารในคอร์ส (ไม่สามารถแก้ไขได้)	4	PK
ifid	Int	รหัสของอาหาร (ไม่สามารถแก้ไขได้)	2	FK reference from iFOOD (ifid) ON DELETE CASCADE ON UPDATE

ตารางที่ 3.10 ตารางอาหารในคอร์ส(Food) (ต่อ)

Column	Type	Description	Example	Constraint
dit	Int	รหัสวันในคอร์ส (ไม่สามารถแก้ไขได้)	1	FK reference from DayOfCouse (did) ON DELETE CASCADE ON UPDATE
time	varchar(15)	เวลารับประทานอาหาร	lunch	Not null

ตารางที่ 3.11 ตารางคลิปท่าออกกำลังกายในคอร์ส(Clip)

Column	Type	Description	Example	Constraint
cpID	Int	รหัสของท่าออกกำลังกายในคอร์ส (ไม่สามารถแก้ไขได้)	1	PK
icpID	Int	รหัสของท่าออกกำลังกาย (ไม่สามารถแก้ไขได้)	1	FK reference from iClip (icpID) ON DELETE
dit	Int	รหัสวันในคอร์ส (ไม่สามารถแก้ไขได้)	1	FK reference from Day (did) ON DELETE CASCADE ON UPDATE

ตารางที่ 3.11 ตารางคลิปท่าออกกำลังกายในคอร์ส(Clip) (ต่อ)

Column	Type	Description	Example	Constraint
status	varchar(1)	Status บอกว่าผู้ใช้ออกกำลังกายหรือยัง (0=ยังไม่ออก,1=ออกกำลังกายแล้ว)	0	Not null

ตารางที่ 3.12 ตารางการซื้อคอร์ส(Buying)

Column	Type	Description	Example	Constraint
bid	Int	รหัสการซื้อคอร์ส (ไม่สามารถแก้ไขได้)	1	PK
uid	Int	รหัสสมาชิก (ไม่สามารถแก้ไขได้)	1	FK reference from CUSTOMER (uid) ON DELETE CASCADE ON UPDATE

ตารางที่ 3.13 ตารางการร้องขอเปลี่ยนท่าออกกำลังกาย(Request)

Column	Type	Description	Example	Constraint
rqID	Int	รหัสเปลี่ยนท่าออกกำลังกาย(ไม่สามารถแก้ไขได้)	1	PK
cid	int	รหัสของโค้ช (ไม่สามารถแก้ไข	1	FK reference from COACH (cid)

ตารางที่ 3.13 ตารางการร้องขอเปลี่ยนท่าออกกำลังกาย(Request)(ต่อ)

Column	Type	Description	Example	Constraint
		ได้)		ON DELETE CASCADE ON UPDATE CASCADE
uid	Int	รหัสสมาชิก (ไม่สามารถแก้ไขได้)	1	FK reference from CUSTOMER (uid) ON DELETE CASCADE ON UPDATE CASCADE
cpID	Int	รหัสของท่าออกกำลังกายในคอร์ส (ไม่สามารถแก้ไขได้)	1	FK reference from Clip (cpID) ON DELETE CASCADE ON UPDATE CASCADE
details	varchar(100)	Details บอกรายละเอียดของการเปลี่ยนท่าออกกำลังกายในแต่ละวันว่าผู้ใช้เปลี่ยนท่าเพราะอะไร	ปวดขา	Not null
status	varchar(1)	Status บอกสถานะการเปลี่ยนท่าออกกำลังกายว่าโค้ชเปลี่ยนยัง	0	0,1 Not null

ตารางที่ 3.14 ตารางพูดคุย(Chat)

Column	Type	Description	Example	Constraint
chID	Int	รหัสพูดคุย(ไม่สามารถแก้ไขได้)	1	PK
uid	Int	รหัสสมาชิก (ไม่สามารถแก้ไขได้)	1	FK reference from CUSTOMER (uid) ON DELETE CASCADE ON UPDATE CASCADE
bid	Int	รหัสการซื้อคอร์ส (ไม่สามารถแก้ไขได้)	1	FK reference from BUYING (bid) ON DELETE CASCADE ON UPDATE CASCADE
cid	int	รหัสของโค้ช (ไม่สามารถแก้ไขได้)	1	FK reference from COACH (cid) ON DELETE CASCADE ON UPDATE CASCADE
message	varchar(250)	varchar(250)	สวัดดีครับ โค้ชเบ้	Not null

3.11 การนำชุดคำสั่ง (API) มาใช้

3.11.1 การเข้าสู่ระบบโดย Facebook

Flutter facebook auth คือชุดคำสั่งของ Facebook เป็นการใช้ Facebook Login โดยไม่ต้องผ่าน Firebase ช่วยให้ผู้ใช้เข้าสู่ระบบแอปพลิเคชันด้วยการเข้าสู่ระบบด้วย Facebook เมื่อเข้าสู่แอปพลิเค

ชันด้วย Facebook ผู้ใช้สามารถให้สิทธิ์การอนุญาตแก่แอปพลิเคชันเพื่อให้รับข้อมูลหรือดำเนินการบางอย่างบน Facebookแทนผู้ใช้ได้ การเขียน Flutter ให้ Login ด้วย Facebook โดยไม่ต้องผ่าน Firebase ต้องสร้างและตั้งค่า Facebook App ก่อน

ในขั้นแรกต้องสร้าง Key hash ของเครื่องที่เขียนโปรแกรม โดยการรันคำสั่ง keytool ให้ได้ Key hashดังภาพประกอบที่3.12 Key hashคือ รหัสเฉพาะในการระบุอุปกรณ์ของเครื่องที่ต้องการพัฒนาแอปพลิเคชัน

```
C:\Users\NAHATHAI>keytool -exportcert -alias androiddebugkey -keystore "C:\Users\NAHATHAI\.android\debug.keystore"
 | "C:\OpenSSL\bin\openssl" sha1 -binary | "C:\OpenSSL\bin\openssl" base64
Enter keystore password:  android
hZiIOz+Qz5ZXOdOybDzIzCOXEIs=

Warning:
The certificate uses the SHA1withRSA signature algorithm which is considered a security risk. This algorithm will
be disabled in a future update.
```

ภาพประกอบที่ 3.10 รันคำสั่ง keytool

ในขั้นแรกต้องสร้าง Key hash ของเครื่องที่เขียนโปรแกรม โดยการรันคำสั่ง keytool ให้ได้ Key hash ดังภาพประกอบที่ 3.12 Key hash คือ รหัสเฉพาะในการระบุอุปกรณ์ของเครื่องที่ต้องการพัฒนาแอปพลิเคชัน

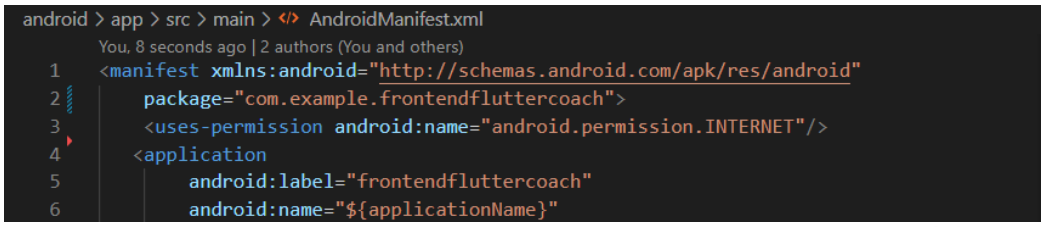
android > app > src > main > res > values > strings.xml

You, last week | 1 author (You)

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">DailyWorkoutCoaching</string>
<string name="facebook_app_id">696221145557150</string>
<string name="fb_login_protocol_scheme">fb696221145557150</string>
<string name="facebook_client_token">4fa460a20134f65808f79a3559ece9fb</string>
</resources>
```

ภาพประกอบที่ 3.11 การกำหนดค่าในไฟล์ strings.xml

บรรทัดที่3-6 เพิ่ม tag string ที่มีชื่อ facebook_app_id , fb_login_protocol_scheme และ facebook_client_token และตั้งค่าให้กับ ID ของแอพและ token client ตรงตามแอปพลิเคชันของเรา



ภาพประกอบที่ 3.12 การกำหนดค่าในไฟล์เพื่ออนุญาตการเชื่อมต่ออินเทอร์

บรรทัดที่ 3 จะทำการกำหนดค่าในไฟล์เพื่อเป็นการอนุญาตการเชื่อมต่ออินเทอร์

```
35      <meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/facebook_app_id"/>
36      <meta-data android:name="com.facebook.sdk.ClientToken" android:value="@string/facebook_client_token"/>
37      <activity android:name="com.facebook.FacebookActivity"
38      android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
39      android:label="@string/app_name" />
40      <activity
41      android:name="com.facebook.CustomTabActivity"
42      android:exported="true">
43      <intent-filter>
44      <action android:name="android.intent.action.VIEW" />
45      <category android:name="android.intent.category.DEFAULT" />
46      <category android:name="android.intent.category.BROWSABLE" />
47      <data android:scheme="@string/fb_login_protocol_scheme" />
48      </intent-filter>
49      </activity>
50  </application>
```

ภาพประกอบที่ 3.13 การกำหนดmeta-dataในไฟล์

บรรทัดที่ 35-49 แท็ก android:name กำหนดชื่อของแท็ก meta-data แท็ก android:value กำหนดค่าของแท็ก meta-data เพิ่มtag meta-data ให้กับ tag application สำหรับ ID ของแอพและ token client ของแอปพลิเคชัน

ในตัวอย่างนี้ เราจะเพิ่มแท็ก meta-data สองแท็ก:

1.com.google.android.gms.appId กำหนด ID แอปพลิเคชัน

2.com.google.firebase.client_id กำหนดโทเค็นไคลเอนต์ของแอปพลิเคชัน ต่อมาจะเป็นในส่วนของการเรียกใช้งานจาก Flutter

```
187           ElevatedButton(
188             child: Text("Login with Facebook"),
189             onPressed: () async {
190               final LoginResult result = await FacebookAuth.instance
191                   .login();
192
193               if (result.status == LoginStatus.success) {
194
195
196                 final AccessToken accessToken = result.accessToken!;
197                 final userData = await FacebookAuth.instance.getUserData();
```

ภาพประกอบที่ 3.14 สร้างปุ่มloginผ่านfacebook (Front End)

บรรทัดที่ 189 เป็นการสร้างปุ่มloginผ่าน Facebook

บรรทัดที่ 190-191 ประกาศตัวแปร result ที่มีชนิดข้อมูลเป็น LoginResult เพื่อเก็บค่าที่ได้จากการเรียกใช้งานการเข้าสู่ระบบด้วย Facebook โดยใช้ Facebook API

บรรทัดที่ 193 นำค่าที่ได้จากการเข้าสู่ระบบมาตรวจสอบ HttpStatus หากมีค่าเท่ากับ LoginStatus.success หรือการเข้าสู่ระบบสำเร็จจะทำงานบรรทัดถัดไป

บรรทัดที่ 196 ประกาศตัวแปร accessToken ที่มีชนิดข้อมูลเป็น AccessToken เพื่อเก็บค่า accessToken ที่ได้จากการเข้าสู่ระบบ accessToken มีการเก็บข้อมูลผู้ใช้จาก Facebook ไว้ในตัวแปร userData

บรรทัดที่ 197 ประกาศตัวแปร userData เพื่อเก็บข้อมูลข้องผู้ใช้ที่ได้จาก Facebook โดย Facebook API จะดึงข้อมูลของผู้ใช้ออกมาโดยใช้เมธอด getUserData()

เมื่อเราได้ ตัวแปร userData แล้วต้องการส่งค่า userData ไปยังหน้าอื่นๆ เราจำเป็นต้องสร้างตัวแปลมาเก็บค่าไว้ ใน Provider เพื่อส่งค่าตัวแปลนั้นไปยังหน้าต่างๆหรือรับค่าเข้ามา

```
3   class AppData with ChangeNotifier {
4     //Api baseurl
5     String baseurl = "http://202.28.34.197:9775";
6
7     Map<String,dynamic> userFacebook = {} ;
8   }
```

ภาพประกอบที่ 3.15 การเก็บค่าไว้ใน Provider (Front End)

บรรทัดที่ 7 สร้างตัวแปลมาเก็บค่าข้อมูลผู้ใช้มาเก็บไว้ใน userFacebook เพื่อดึงข้อมูลผู้ใช้มาใช้ในหน้าสมัครสมาชิก

ในส่วนต่อไปทำการสร้างคลาสในภาษา Go เพื่อทำการรับ Attribute ของ Entity จาก Front End ดังภาพประกอบที่ 3.18

```
 8  type LoginFBDTO struct {
 9      FacebookID string `json:"facebookID"`
10  }
```

ภาพประกอบที่ 3.16 ประกาศคลาส LogInDTO (Back End)

บรรทัดที่ 8-9 ประกาศคลาส LogInFBDTO พร้อมทั้งสร้าง Attribute ของ Entity สำหรับรับข้อมูลจาก Front End

```
 3  type CoachAndCus struct {
 4      Cid uint `gorm:"column:cid;primaryKey"`
 5      Uid uint `gorm:"column:uid;primaryKey"`
 6  }
 7
 8  func (CoachAndCus) TableName() string {
 9      return "Coach,Customer"
10  }
11
```

ภาพประกอบที่ 3.17 ประกาศคลาส CoachAndCus (Back End)

บรรทัดที่ 4-5 ประกาศคลาส CoachAndCus พร้อมทั้งสร้าง Attribute ของ Entity สำหรับเก็บฐานข้อมูล Databass

บรรทัดที่ 8-10 ประกาศคลาส TableName() ที่เก็บคลาส CoachAndCus พร้อมทั้ง return ชื่อ Table ในฐานข้อมูล

```go
func NewUserController(router *gin.Engine) {
    user := router.Group("/user")
    {

        user.POST("/loginfb", loginFBPostBody)

    }

}
```

ภาพประกอบที่ 3.18 สร้าง func เพื่อเรียก func loginFBPostBody (Back End)

บรรทัดที่ 19 สร้างตัวแปร user มาเก็บ router.Group เพื่อจัดกลุ่ม Path และ เรียกใช้ตัวแปล ping มาระบุ Path ให้กับ Controller โดยเรียกใช้เป็น POST

```go
func loginFBPostBody(ctx *gin.Context) {
    // input json
    jsonDto := dto.LoginFBDTO{}
    err := ctx.ShouldBindJSON(&jsonDto)
    fmt.Printf(jsonDto.FacebookID)
    if err != nil {
        panic(err)
    }

    coach, cus, err := userDateService.ServiceLoginFB(jsonDto.FacebookID)
    if err != nil {
        panic(err)
    }

    if coach.Cid > 0 {
        ctx.JSON(http.StatusOK, models.CoachAndCus{Cid: coach.Cid})

    } else if cus.Uid > 0 {

        ctx.JSON(http.StatusOK, models.CoachAndCus{Uid: cus.Uid})
    } else if cus.Uid == 0 {

        ctx.JSON(http.StatusOK, models.CoachAndCus{Cid: coach.Cid, Uid: cus.Uid})

    } else if coach.Cid == 0 {

        ctx.JSON(http.StatusOK, models.CoachAndCus{Cid: coach.Cid, Uid: cus.Uid})

    }
}
```

ภาพประกอบที่ 3.19 Controller ส่วนเข้าสู่ระบบด้วย ID Facebook (Back End)

บรรทัดที่ 30 สร้างตัวแปล jsonDto เพื่อดึงตัวแปล LoginFBDTO มาเก็บ

บรรทัดที่ 31 สร้างตัวแปล err เพื่อเก็บค่า ctx.ShouldBindJSON คือการ bind request body ให้เป็นในรูปแบบ JSON กับ struct

บรรทัดที่ 33-35 นำตัวแปล err มาเช็คค่า null หากไม่มีค่า null ให้ panic

บรรทัดที่ 37 เรียกใช้เมธอด ServiceLoginFB จาก service และ ส่งค่าตัวแปล jsonDto ไปยัง service และนำ service มาเก็บค่าในตัวแปล coach,cus,err

บรรทัดที่ 38-40 นำตัวแปล err มาเช็คค่า null หากไม่มีค่า null ให้ panic

บรรทัดที่ 42-56 นำตัวแปล coach , cus , err มาเช็คค่าใน IF ถ้า Cid หรือ Uid ของผู้ใช้มีค่ามากกว่า 0 ให้แสดง models.CoachAndCus ที่ส่ง Cid และ Uid ของผู้ใช้ในรูปแบบ JSON

ในส่วนต่อไปทำการสร้างคลาส Customer เพื่อทำการรับ Attribute ของ Entity ในฐานข้อมูล

```
9  type Customer struct {
10     Uid        uint      `gorm:"column:uid;primaryKey"`
11     Username   string    `gorm:"column:username;size:50"`
12     Password   string    `gorm:"column:password;size:20"`
13     Email      string    `gorm:"column:email;size:100"`
14     FullName   string    `gorm:"column:fullName;size:100"`
15     Birthday   time.Time `gorm:"column:birthday"`
16     Gender     string    `gorm:"column:gender;size:1"`
17     Phone      string    `gorm:"column:phone;size:10"`
18     Image      string    `gorm:"column:image;size:3000"`
19     Weight     uint      `gorm:"column:weight"`
20     Height     uint      `gorm:"column:height"`
21     FacebookID string    `gorm:"column:facebookId;size:100"`
22     Price      uint      `gorm:"column:price"`
23
24     //Chats []Chat       `gorm:"foreignKey:uid"`
25     Buying []Buying `gorm:"foreignKey:uid"`
26 }
27
28 func (Customer) TableName() string {
29     return "Customer"
30 }
```

ภาพประกอบที่ 3.20 ประกาศคลาส Customer (Back End)

บรรทัดที่ 9-26 ประกาศคลาส Customer พร้อมทั้งสร้าง Attribute ของ Entity สำหรับเก็บฐานข้อมูล Databass

บรรทัดที่ 28-30 ประกาศคลาส TableName() ที่เก็บคลาส Customer พร้อมทั้ง return ชื่อ Table ในฐานข้อมูล

ในส่วนต่อไปทำการประกาศคลาสเพื่อทำการเก็บคำสั่งที่ติดต่อไปยัง Databass ดังภาพประกอบที่ 3.23

```
19  type UserData struct {
20      db *gorm.DB
21  }
22
```

ภาพประกอบที่ 3.21 ประกาศคลาส UserData (Back End)

บรรทัดที่ 19-22 ประกาศคลาส UserData เพื่อเก็บคำสั่ง db *gorm.DB เป็นการติดต่อฐานข้อมูล

```
23  // ServiceLoginFB implements UserDataService
24  func (UserData) ServiceLoginFB(FackbookID string) (*models.Coach, *models.Customer, error) {
25      repo := repository.NewUserRepository()
26      coach, cus, err := repo.LoginFB(FackbookID)
27      if err != nil {
28          panic(err)
29      }
30      return coach, cus, nil
31  }
32
```

ภาพประกอบที่ 3.22 Service ในการเข้าสู่ระบบด้วย Facebook ID (Back End)

บรรทัดที่ 25-26 ของโค้ดตัวอย่าง เราจะสร้างตัวแปล repo เพื่อเก็บ Repository ที่ต้องการเรียกใช้ และเรียกใช้ repo.LoginFB() ที่เราสร้างไว้มาและนำ FacebookID มาใส่

นี่คือสิ่งที่เกิดขึ้นในแต่ละบรรทัด:

บรรทัดที่ 25 เราสร้างตัวแปล repo ขึ้นใหม่เป็นชนิด *repository.Repository

บรรทัดที่ 26 เราเรียกใช้ repo.LoginFB() ซึ่งจะส่งค่า FacebookID ไปให้ Repository จัดการ

Repository จะดำเนินการต่อไปนี้ ค้นหาข้อมูลผู้ใช้จากฐานข้อมูลโดยใช้ FacebookID หากพบข้อมูลผู้ใชแล้ว Repository จะสร้างตัวแปล coach และ cus เพื่อเก็บข้อมูลผู้ใช้ หากไม่พบข้อมูลผู้ใช้แล้ว Repository จะสร้างตัวแปล err เพื่อเก็บข้อผิดพลาด

บรรทัดที่ 27-29 นำตัวแปล err มาเช็คค่า null หากไม่มีค่า null ให้ panic

บรรทัดที่ 30 return ค่า coach , cus , nill ที่เราได้รับจาก Repository

```
20  // LoginFB implements UserRepository
21  func (u userDB) LoginFB(fackbookID string) (*models.Coach, *models.Customer, error) {
22      coachs := models.Coach{}
23      customers := models.Customer{}
24
25      resultCoa := u.db.Where("facebookId = ?", fackbookID).Find(&coachs)
26      if resultCoa.Error != nil {
27          return nil, nil, resultCoa.Error
28      }
29
30      resultCus := u.db.Where("facebookId = ?", fackbookID).Find(&customers)
31      if resultCus.Error != nil {
32          return nil, nil, resultCus.Error
33      }
34
35      return &coachs, &customers, nil
36  }
```

ภาพประกอบที่ 3.23 Repository ในการเข้าสู่ระบบด้วย Facebook ID (Back End)

ฟังก์ชัน LoginFB() ทำงานดังนี้:

1.สร้างตัวแปล coach และ customer เพื่อเก็บค่า Coach และ Customer

2.สร้างตัวแปล result เพื่อเก็บผลลัพธ์ของคำสั่ง SQL

3.เรียกใช้คำสั่ง SQL เพื่อค้นหาข้อมูลผู้ใช้จากฐานข้อมูลโดยใช้ FacebookID

4.เก็บค่า Coach และ Customer จากฐานข้อมูลลงในตัวแปล coach และ customer ตามลำดับ

5.หากเกิดข้อผิดพลาดในขั้นตอนใด ฟังก์ชันจะส่งค่า error กลับ

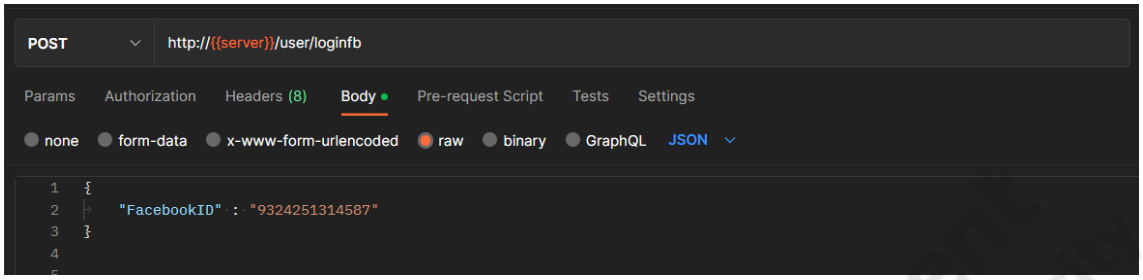
บรรทัดที่ 21 ประกาศคลาส LoginFB โดยประการตัวแปล u userDB ที่ประกาศในภาพประกอบที่ 3.24 และ รับค่า FackbookID Type เป็น string และ return ค่าเป็น models Coach,Customer และ error

บรรทัดที่ 22-23 สร้างตัวแปล coach และ customers เพื่อเก็บค่า models Coach,Customer

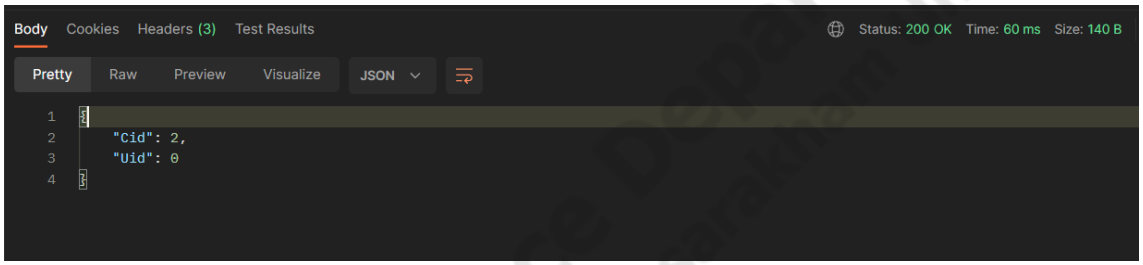
บรรทัดที่ 25-33 สร้างตัวแปล result เพื่อเก็บคำสั่ง SQL เช็ค FacebookId และใช้คำสั่ง .Find(&coach) , .Find(&customer) เพื่อหาค่าข้อมูลในฐานข้อมูล Databass เมื่อได้ค่าข้อมูลมาแล้วทำการเช็ค Error โดยการใช้คำสั่ง IF เมื่อค่า result.Error ไม่เท่ากับค่าว่าง จะ return ค่า Error ไป

บรรทัดที่ 35 return ค่า coach,cus,nill ที่เราได้รับข้อมูลในคำสั่ง.Find

ในส่วนต่อไปทำการเรียกใช้ Service โดยการใส่ FacebookID ของผู้ใช้ใน Postman



ภาพประกอบที่ 3.24 เรียกใช้ Service ล็อคอินโดยใช้ id จาก Facebook



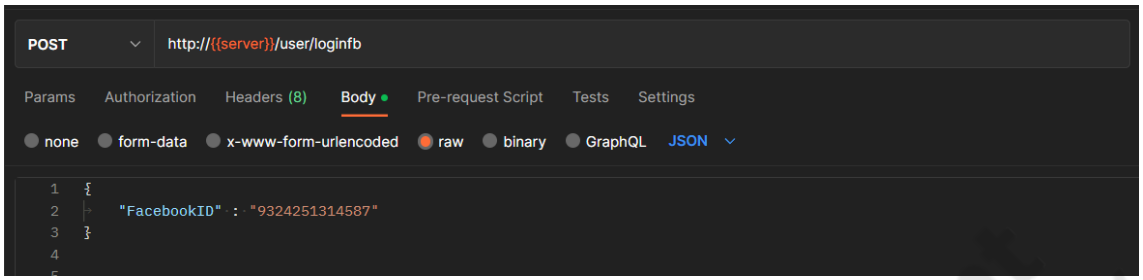
ภาพประกอบที่ 3.25 ข้อมูล user ที่ได้จากการเรียกใช้Service

ในส่วนของการเรียกใช้งานใน Flutter ทำการส่งค่าข้อมูลผู้ใช้ที่เข้าสู่ระบบด้วย Facebook ไปยัง Provider เพื่อเก็บค่า และทำการนำไปใช้หน้าต่างๆ

บรรทัดที่ 25-33 สร้างตัวแปล result เพื่อเก็บคำสั่ง SQL เช็ค FacebookId และใช้คำสั่ง .Find(&coach) , .Find(&customer) เพื่อหาค่าข้อมูลในฐานข้อมูล Databass เมื่อได้ค่าข้อมูลมาแล้วทำการเช็ค Error โดยการใช้คำสั่ง IF เมื่อค่า result.Error ไม่เท่ากับค่าว่าง จะ return ค่า Error ไป

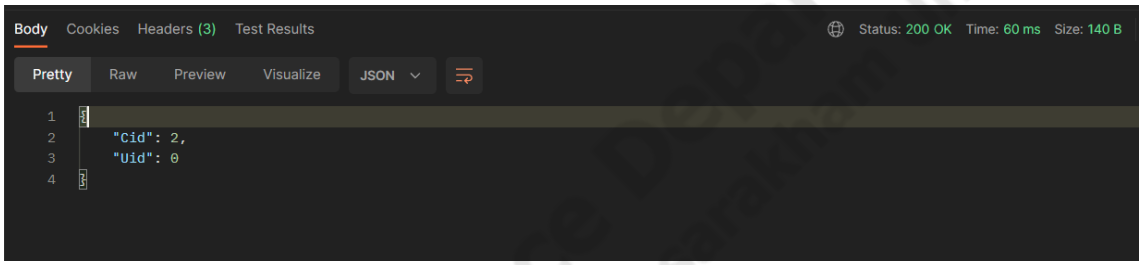
บรรทัดที่ 35 return ค่า coach,cus,nill ที่เราได้รับข้อมูลในคำสั่ง.Find

ในส่วนต่อไปทำการเรียกใช้ Service โดยการใส่ FacebookID ของผู้ใช้ใน Postman



ภาพประกอบที่ 3.26 เรียกใช้ Service ล็อคอินโดยใช้ id จาก Facebook

- FacebookID: รหัสFacebookประจำตัวของผู้ใช้



ภาพประกอบที่ 3.27 ข้อมูล user ที่ได้จากการเรียกใช้Service

- cid: รหัสประจำตัวของโค้ช
- Uid: รหัสประจำตัวของผู้ใช้

ในส่วนของการเรียกใช้งานใน Flutter ทำการส่งค่าข้อมูลผู้ใช้ที่เข้าสู่ระบบด้วย Facebook ไปยัง Provider เพื่อเก็บค่า และทำการนำไปใช้หน้าต่างๆในไฟล์ home.dart จะเรียกใช้ Provider เพื่อรับค่าข้อมูลผู้ใช้ user จากนั้นจะแสดงค่าชื่อของผู้ใช้บนหน้า Homeในไฟล์ user_provider.dart จะเก็บค่าข้อมูลผู้ใช้ไว้ในตัวแปร _user จากนั้นจะเรียกใช้ Service เพื่อรับค่าข้อมูลผู้ใช้จาก Facebook Login หากได้รับค่าข้อมูลผู้ใช้แล้ว จะเก็บค่าไว้ในตัวแปร _user และแจ้งให้ Widget ที่ subscribe รับรู้ถึงการเปลี่ยนแปลงในไฟล์ user.dart กำหนดโครงสร้างข้อมูลของข้อมูลผู้ใช้

ด้วยวิธีนี้ ข้อมูลผู้ใช้ที่เข้าสู่ระบบด้วย Facebook จะถูกเก็บไว้ใน Provider และสามารถเข้าถึงได้จากทุกหน้าใน Flutter App ได้อย่างง่ายดาย นอกจากนี้ ข้อมูลผู้ใช้ยังสามารถนำไปใช้กับฟีเจอร์อื่นๆ ใน Flutter App ได้อีกด้วย เช่น เก็บข้อมูลผู้ใช้ไว้ในฐานข้อมูล เปรียบเทียบข้อมูลผู้ใช้กับข้อมูลในระบบอื่นๆ กำหนดสิทธิการใช้งานของผู้ใช้ โดยสรุป ข้อมูลผู้ใช้ที่เข้าสู่ระบบด้วย Facebook เป็นข้อมูลสำคัญที่สามารถนำไปใช้ในการพัฒนา Flutter App ได้หลากหลายรูปแบบ

```
206                 cidfb = int.parse(jsonEncode(userLoginCus.data.cid));
207                 if (cidfb > 0) {
208                   // ignore: use_build_context_synchronously
209                   Navigator.push(
210                     context,
211                     MaterialPageRoute(
212                       builder: (_) => HomePage(),
213                     ), // MaterialPageRoute
214                   );
215                 } else if (uidfb > 0) {
216                   log("uidfb:" + uidfb.toString());
217                   // ignore: use_build_context_synchronously
218                   Navigator.push(
219                     context,
220                     MaterialPageRoute(
221                       builder: (_) => HomePage(),
222                     ), // MaterialPageRoute
223                   );
224                 } else {
225                   // ignore: use_build_context_synchronously
226                   Navigator.push(
227                     context,
228                     MaterialPageRoute(
229                       builder: (_) => RegisterPage(),
230                     ), // MaterialPageRoute
231                   );
232                 }
```

ภาพประกอบที่ 3.28 การเช็ค ID ผู้ใช้ (Front End)

บรรทัดที่ 207-232 นำตัวแปล cidfb , uidfb มาเช็คค่าใน IF โดยใช้คำสั่ง เมื่อ cidfb , uidfb มีค่ามากกว่า 0 จะถูกส่งไปยังหน้า HomePage ถ้าหาก cidfb , uidfb มีค่าน้อยกว่า 0 จะถูกส่งไปยังหน้า สมัครสมาชิก

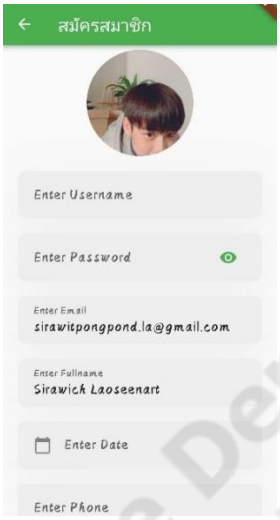
```
90   void initState() {
91     super.initState();
92
93     userFacebook = context.read<AppData>().userFacebook;
94
95     _length = userFacebook['name'].length;
96
97       _email = userFacebook['email'];
98       _fullName = userFacebook['name'];
99       _image = userFacebook['picture']['data']['url'];
100      _facebookID = userFacebook['id'];
```

ภาพประกอบที่ 3.29 การส่งค่า Facebook ไปหน้าสมัครสมาชิก (Front End)

บรรทัดที่ 90-91 เป็น Method เมื่อเข้าหน้าสมัครสมาชิกจะถูกเรียกใช้

บรรทัดที่ 93 สร้างตัวแปล userFacebook เพื่อดึงข้อมูล Provider ที่เราเก็บค่าไว้มายังหน้าสมัครสมาชิก

บรรทัดที่ 97-100 สร้างตัวแปลมาเก็บข้อมูล email,ชื่อ,รูป,facebookID ของ userFacebookที่เราดึงข้อมูลมาจาก Provider



ภาพประกอบที่ 3.30 ตัวอย่างการแสดงรูปภาพหลังดึงข้อมูลใน Provider

3.11.2 File Picker

```
16  class _UploadpicturePageState extends State<UploadpicturePage> {
17    PlatformFile? pickedFile;
18    UploadTask? uploadTask;
19
20
21    Future selectFile() async{
22      final result = await FilePicker.platform.pickFiles();
23      if(result == null) return;
24
25      setState(() {
26        pickedFile = result.files.first;
27      });
28    }
```

ภาพประกอบที่ 3.31 การเลือกรูปภาพจากไฟล์

ในส่วนของการเลือกรูปภาพในขั้นแรกก่อนที่จะใช้งาน API นี้ต้องติดตั้ง plugin ของ File Picker ก่อนซึ่ง File Picker เป็นการ import ไฟล์จากภายนอก เราจะใช้ package ที่ชื่อว่า File Picker ซึ่งเป็นตัวที่เรียกใช้งานรูปแบบวิธีการเรียกดูไฟล์ของระบบที่มีอยู่แล้วมาใช้อีกที

บรรทัดที่ 17 เป็นการประกาศตัวแปร pickedFile โดยมีการใช้APIที่มีชื่อว่า File Picker ที่เป็นตัวช่วยในการใช้คำสั่งที่สามารถเลือกรูปจากไฟล์ได้

บรรทัดที่ 21 สร้างคลาสที่ใช้ในการเลือกรูปภาพชื่อว่า selectFile

บรรทัดที่ 22 สร้างตัวแปรที่ชื่อว่าresultเป็นตังเก็บไฟล์ภาพที่เราเลือกจากไฟล์

บรรทัดที่ 23 สร้างเงื่อนไขหากresultเป็นnullจะทำการreturnกลับ

บรรทัดที่ 26 ให้ pickedFile เป็นตัวเก็บresultหรือรูปที่เราเลือกจากไฟล์

```
52          if(pickedFile != null)
53          Expanded(
54            child: Container(
55              color: Colors.green,
56
57              child: Image.file(
58                File(pickedFile!.path!),
59                width: double.infinity,
60                fit: BoxFit.cover,) // Image.file
61            ), // Container
62            ), // Expanded
63            const SizedBox(height: 12,),
64          ElevatedButton(
65            child: const Text('Select File'),
66            onPressed: selectFile ,
67            ), // ElevatedButton
68            const SizedBox(height: 12,),
```

ภาพประกอบที่ 3.32 การสร้างปุ่มการเลือกรูปจากไฟล์และการแสดงรูปภาพที่เลือก

บรรทัดที่ 52 สร้างเงื่อนไขการเลือกรูปภาพโดยเงื่อนไขคือถ้าตัวแปร pickedFile ไม่เป็นnull จะต้องทำงานที่บรรทัดถัดไป

บรรทัดที่ 53 สร้างชุดคำสั่งการแสดงรูป โดยการใช้ Widgetคือ Expanded

บรรทัดที่ 54 เป็นการกำหนดสีพื้นหลังให้รูปภาพ

บรรทัดที่ 57-58 เป็นชุดคำสั่งในการเลือกรูปภาพที่จะแสดง โดยจะเลือกไฟล์ที่จะแสดงมาจากpathของตัวแปร pickedFile

บรรทัดที่ 59-60 เป็นการกำหนดขนาดของรูปภาพ

ภาพประกอบที่ 3.33 ตัวอย่างการแสดงรูปภาพหลังเลือกไฟล์

3.11.3 Upload Files to Firebase Storage

การอัพโหลดไฟล์ขึ้นบน Firebase ซึ่งเป็นบริการพื้นที่เก็บข้อมูล เอาไว้เก็บภาพ วิดีโอ หรือไฟล์ขนาดใหญ่จากแอพของผู้ใช้ โดยการใช้ Firebase Storage ต้องสร้าง project firebase ก่อนแล้วจึงไป set up firebase ก่อนที่จะใช้งาน โดยเริ่มจากการติดตั้งและเรียกใช้ FlutterFire CLI หลังจากนั้นจะทำการเพิ่ม plugins โดยเพิ่มใน terminal ของโปรเจค ซึ่งใช้คำสั่ง flutter pub add firebase_core , flutter pub add firebase_storage , flutter pub add cloud_firestore เมื่อเราทำการเพิ่ม plugins แล้วก็จะสามารถเรียกใช้งานในการอัพโหลดไฟล์ขึ้น Firebase Storageได้ หลังจากนั้นเราจะมาเขียนคำสั่งในการอัพโหลดไฟล์ขึ้น Firebase Storage ได้ดังนี้

```dart
class _UploadpicturePageState extends State<UploadpicturePage> {
  PlatformFile? pickedFile;
  UploadTask? uploadTask;


  Future selectFile() async{
    final result = await FilePicker.platform.pickFiles();
    if(result == null) return;

    setState(() {
      pickedFile = result.files.first;
    });
  }
  Future uploadFile() async{
    final path = 'files/${pickedFile!.name}';
    final file = File(pickedFile!.path!);

    final ref = FirebaseStorage.instance.ref().child(path);
    uploadTask = ref.putFile(file);

    final snapshot = await uploadTask!.whenComplete(() => {});

    final urlDownload = await snapshot.ref.getDownloadURL();
    print('Download Link: $urlDownload');
  }
```

ภาพประกอบที่ 3.34 การ Upload Files to Firebase Storage

บรรทัดที่ 18 เป็นการประกาศตัวแปร uploadTask โดยมีการใช้ API ที่มีชื่อว่า firebase storage ที่เป็นตัวช่วยในการใช้คำสั่งที่สามารถอัพโหลดรูปจากไฟล์ได้

บรรทัดที่ 21 สร้างคลาสที่ใช้ในการอัพโหลดรูปภาพชื่อว่า uploadFile

บรรทัดที่ 30 ประกาศตัวแปร path เป็นตัวเก็บเส้นทางที่เรากำหนดว่าต้องการเก็บไฟล์ที่เราอัพโหลดไฟล์ขึ้น firebase ว่าจะให้เก็บไปยังเส้นทางไหน แล้วตามด้วยชื่อไฟล์ที่เราเลือกมาจากตัวแปร pickedFile

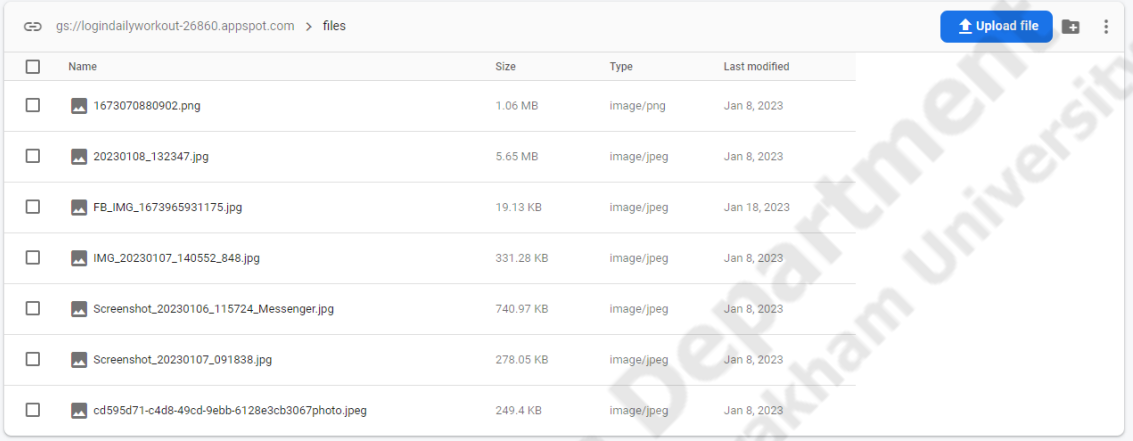
บรรทัดที่ 31 จะทำการเปลี่ยนจาก path เป็น file ไว้ในตัวแปร file

บรรทัดที่ 33 ประการตัวแปร ref ให้เป็นตัวเชื่อมตอกับ Firebase โดยใช Firebase core และ Firebase storage

บรรทัดที่ 34 ให้ตัวแปร uploadTask เก็บคำสั่งในการอัพโหลดไฟล์ขึ้นไปยัง Firebase

บรรทัดที่ 36 ประกาศตัวแปร snapshot ให้เก็บคำสั่งการอัพโหลดไฟล์ที่สำเร็จ

บรรทัดที่ 38 ให้ตัวแปร urlDownload เมื่อมีการอัพโหลดไฟล์เสร็จจะได้ลิงค์ไฟลที่อัพโหลดไปยังFirebase



ภาพประกอบที่ 3.35 ไฟล์ที่อัพโหลดเสร็จสิ้นจะถูกเก็บใน Firebase storage

3.11.4 Appinio video player

หลังจากอัพโหลดไฟล์ขึ้น Firebase Storage ได้แล้ว หากเราต้องการเรียกดูไฟล์ที่เป็นวิดีโอจำเป็นต้องเพิ่ม plugins appinio video player ซึ่ง appinio video player คือ API ที่ใช้เล่นวิดีโอ ที่มีแพ็คเกจลูกเล่น ไม่ว่าจะเป็นโหมดเล่นเต็มหน้าจอและการตั้งค่าปรับวิดีโอที่ได้ และสามารถกดหยุดและเล่นต่อได้ก่อนที่จะเรียกใช้ API นี้ได้ต้องทำการเพิ่ม plugins โดยเพิ่มใน terminal ของโปรเจค ซึ่งใช้คำสั่ง flutter pub add appinio_video_player หลังจากเพิ่ม plugins แล้วก็จะสามารถเรียกใช้ในการเขียน code การเล่นวิดีโอได้ดังต่อไปนี้

```
18  class _PlayVideoPage extends State<PlayVideoPage> {
19    late VideoPlayerController _controller;
20    late CustomVideoPlayerController _customVideoPlayerController;
21    @override
22    void initState() {
23      super.initState();
24    }
```

ภาพประกอบที่ 3.36 ประกาศตัวแปรและกำหนดค่าเริ่มต้น

บรรทัดที่ 19 ประกาศตัวแปร _controller ที่มีการใช้ API มีชื่อว่า appinio video player ที่เป็นตัวช่วยในการใช้คำสั่งที่สามารถปรับแต่งวิดีโอ

บรรทัดที่ 20 ประกาศตัวแปร _customVideoPlayerController ที่มีการใช้ API มีชื่อว่า appinio video player ที่เป็นตัวช่วยในการใช้คำสั่งที่สามารถเล่นวิดีโอ

```
49      _controller = VideoPlayerController.network('$urlDownload')
50        ..initialize().then((_) {
51          // Ensure the first frame is shown after the video is initialized, even before the play button has been pressed.
52          setState(() {});
53          _customVideoPlayerController = CustomVideoPlayerController(
54            context: context,
55            videoPlayerController: _controller,
56          );
57        });
58    }
```

ภาพประกอบที่ 3.37 การปรับแต่งวิดีโอโดยใช้แพ็คเกจ

บรรทัดที่ 49 ให้ตัวแปร _controller เก็บ url ที่ได้หลังจากการอัพโหลดไฟล์เสร็จให้อยู่ในฟังก์ชันการแสดงวิดีโอ

บรรทัดที่ 50-52 ทำการ initialize แล้วสั่งให้กำหนดค่าใหม่ให้กับ VideoPlayerController

บรรทัดที่ 53-55 เป็นการกำหนดให้วิดีโอสามารถใช้ฟังก์ชันการปรับลูกเล่นวิดีโอได้

```
70                  if (pickedFile != null)
71                    Expanded(
72                      child: SafeArea(
73                        child: CustomVideoPlayer(
74                            customVideoPlayerController:
75                                _customVideoPlayerController), // CustomVideoPlayer
76                      ), // SafeArea
77                    ), // Expanded
```

ภาพประกอบที่ 3.38 ชุดคำสั่งการเล่นวิดีโอ

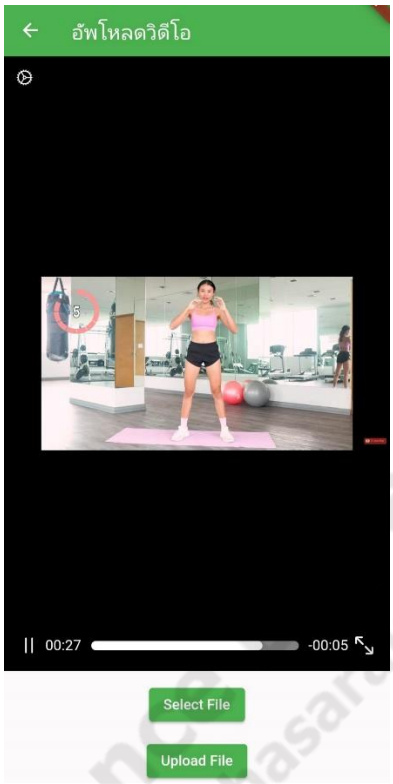
บรรทัดที่ 70 สร้างเงื่อนไขการแสดงวิดีโอโดยเงื่อนไขคือถ้าตัวแปร pickedFile ไม่เป็นnull จะต้องทำงานภายในเงื่อนไข

บรรทัดที่ 71 สร้างชุดคำสั่งการเล่นวิดีโอ โดยการใช้ Widget คือ Expanded

บรรทัดที่ 72 สร้างชุดคำสั่งการเล่นวิดีโอ โดยการใช้ Widget คือ SafeArea

บรรทัดที่ 73 สร้างชุดคำสั่งการเล่นวิดีโอ โดยการใช้ Widget คือ CustomVideoPlayer

บรรทัดที่ 74-75 เป็นการสร้างชุดคำสั่งการเล่นวิดีโอ



ภาพประกอบที่ 3.39 ตัวอย่างการเล่นวิดีโอหลังอัพโหลดไฟล์