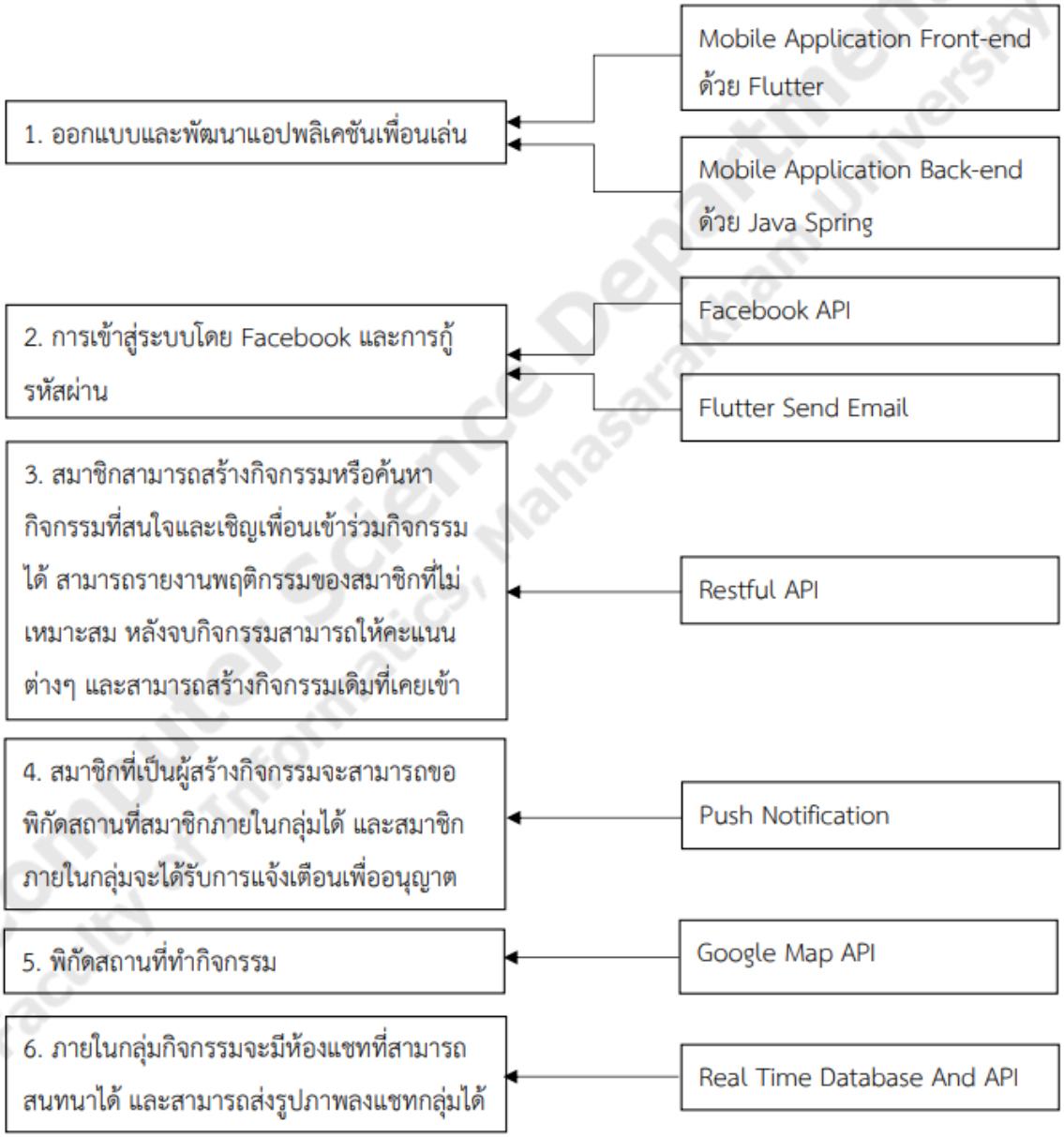
บทที่ 3

ขั้นตอนการดำเนินงาน

3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบ

คำอธิบาย

1. เขียนโปรแกรมเพื่อสร้างแอปพลิเคชันเพื่อนเล่น สำหรับให้ผู้ใช้ใช้งานแอปพลิเคชันผ่านสมาร์ทโฟน โดยการสร้างแอปพลิเคชันจะต้องมีการติดตั้งเครื่องมือดังนี้

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

- Spring framework สามารถสร้างแอปพลิเคชันได้โดยง่าย มีความยืดหยุ่นในการใช้งานใช้ติดต่อกับฐานข้อมูลฝั่ง Back end พัฒนาโดย Eclipse

2. การเข้าสู่ระบบโดย Facebook

Facebook API คือ การเข้าสู่ระบบโดย Facebook เมื่อต้องการเข้า app ไหนที่มีการขอ Permission จากผู้ใช้งานเฟซบุ๊คว่า App ต้องการขอ email ขอโพสต์ลงหน้า Wall จะต้องสร้าง App ในส่วนของ Facebook Developer เพื่อยืนยันการใช้งานบนเว็บไซต์ URL ที่สร้าง APP ID เอาไว้แล้ว

3. สมาชิกสามารถสร้างกิจกรรมหรือค้นหากิจกรรมที่สนใจและเชิญเพื่อนเข้าร่วมกิจกรรมได้ สามารถรายงานพฤติกรรมของสมาชิกที่ไม่เหมาะสม หลังจบกิจกรรมสามารถให้คะแนนต่างๆ และสามารถสร้างกิจกรรมเดิมที่เคยเข้า

RESTful – RESTful Web Service (RWS) คือ Web Service ที่ใช้สถาปัตยกรรม Rest ซึ่งตัว RWS อนุญาต ให้ระบบ Request และเข้าถึง Resource บนเว็บโดยใช้ชุดคำสั่งที่กำหนดเอาไว้ล่วงหน้า โดยที่การโต้ตอบของระบบที่ใช้ REST จะอยู่บนพื้นฐานของ Hypertext Transfer Protocol (HTTP). Request จะส่งคำขอไปยัง URI ที่กำหนด และเอา response กลับมาเป็น Payload ในแบบ HTML, XML, JSON หรือ format อื่นๆ โดย RESTful จะประกอบไปด้วย Client – ผู้ที่เข้ามาเป็น Request Resource, Server – ผู้ที่ให้บริการ Resource

4. สมาชิกที่เป็นผู้สร้างกิจกรรมจะสามารถขอพิกัดสถานที่สมาชิกภายในกลุ่มได้และสมาชิกภายในกลุ่มจะได้รับการแจ้งเตือนเพื่ออนุญาตพิกัดตนเอง

- Push Notification หรือ การแจ้งเตือน คือ การที่แอปพลิเคชันนำข้อมูลมาแสดงในแถบแจ้งเตือนของระบบปฏิบัติการนั้นๆ กำหนด ไม่ว่าจะเป็น Mobile (iOS, Android) หรือ Computer ทั่วไป ซึ่ง modern browser ในปัจจุบันก็สามารถแสดงแถบแจ้งเตือนได้แล้ว

5. พิกัดสถานที่ทำกิจกรรม

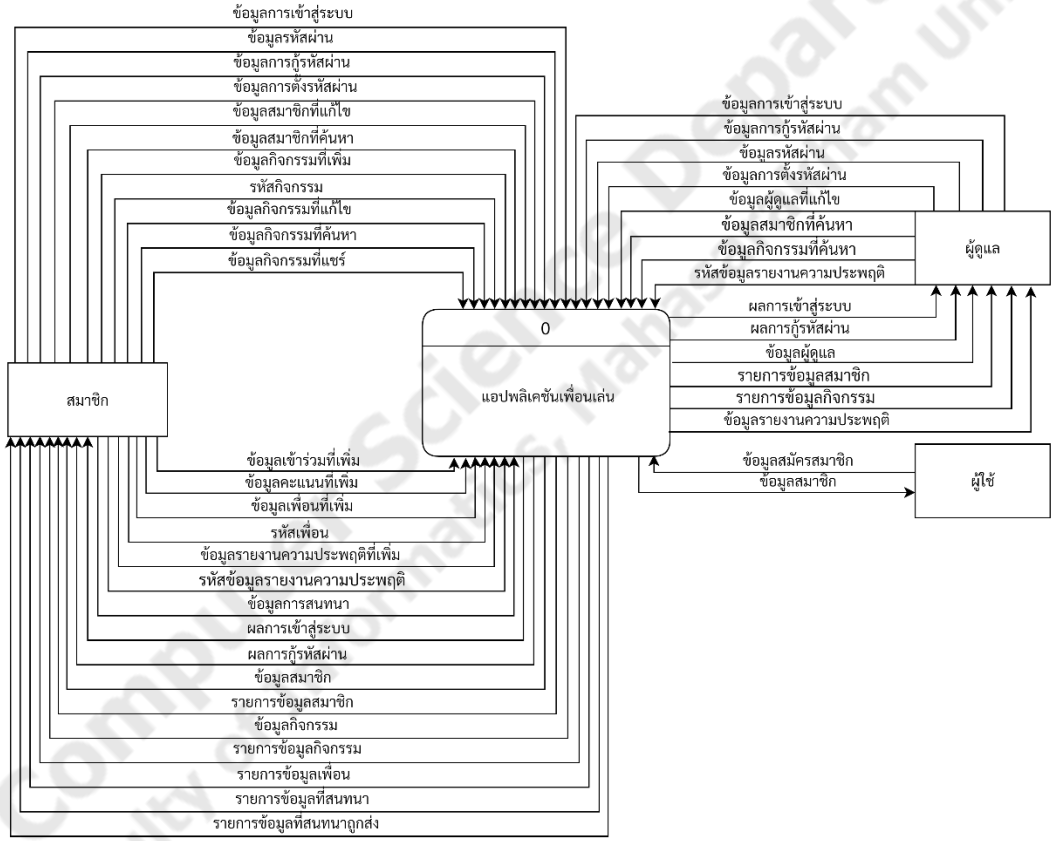
Google Map API Google Maps Platform หรือ GMP เป็น API ของ Google ที่ให้นักพัฒนาโปรแกรม หรือ Developer สามารถเรียกไปใช้งานเพื่อสร้าง Application ขององค์กรตนเอง โดย Based on Google Maps ซึ่งรองรับการทำงานทั้ง Web Application และ Mobile Application ผ่านระบบปฏิบัติการ iOS และ Android

6. ภายในกิจกรรมจะมีห้องแชทที่สามารถสนทนาได้ และสามารถส่งรูปภาพลงแชทได้อีกด้วย

- Real Time Database and API เป็น NoSQL cloud database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการ sync ข้อมูลแบบ Realtime กับทุก devices ที่เชื่อมต่อแบบอัตโนมัติในเสี้ยววินาที รองรับการทำงานเมื่อ offline (ข้อมูลจะถูกเก็บไว้ใน local จนกระทั่งกลับมา online ก็จะทำการ sync ข้อมูลให้อัตโนมัติ) รวมถึงมี Security Rules ให้เราสามารถออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการ read และ write ได้ดังใจ ทั้ง Android, iOS และ Web

3.2 การออกแบบระบบ

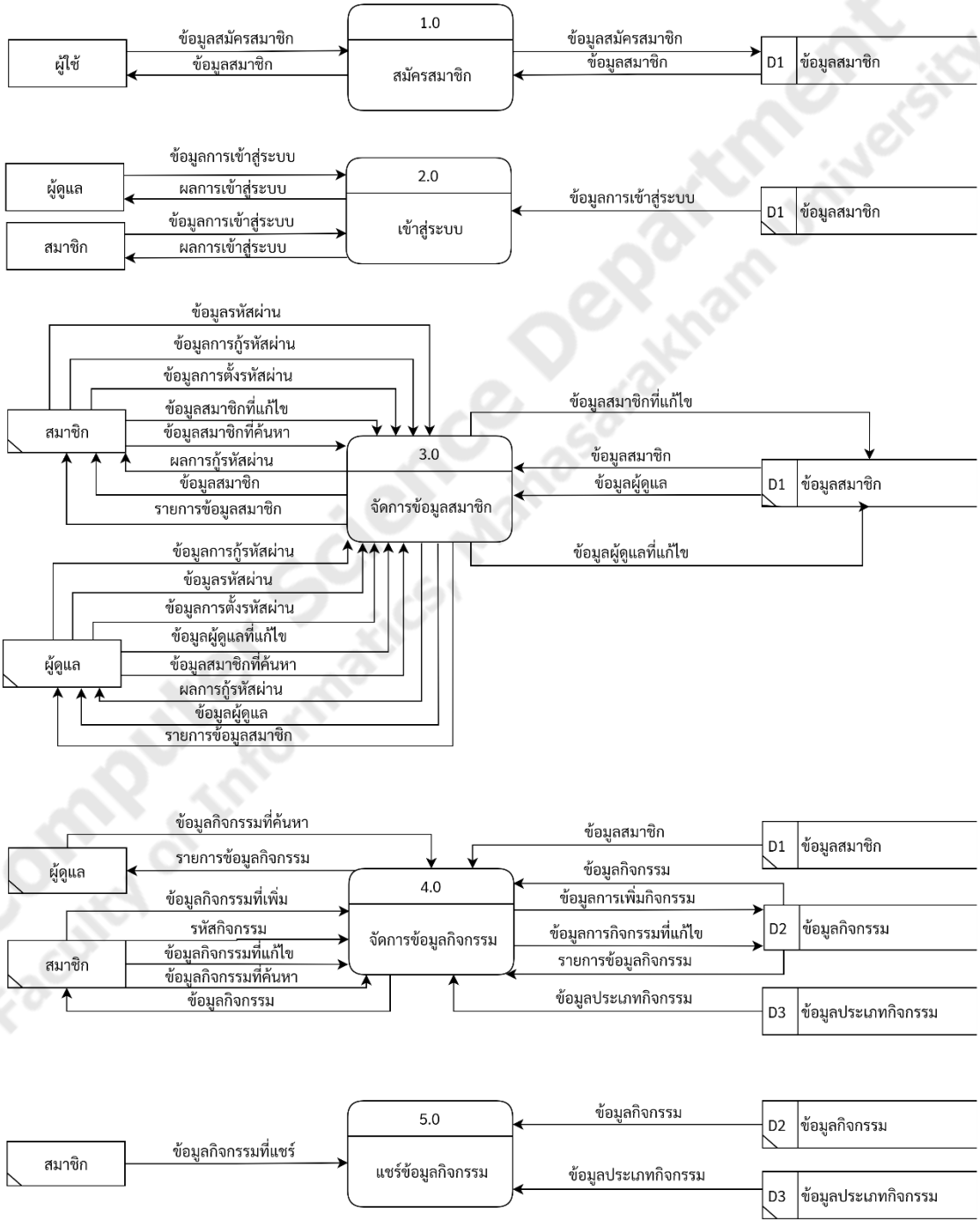
3.2.1 แผนภาพการไหลของข้อมูล (Context Diagram)



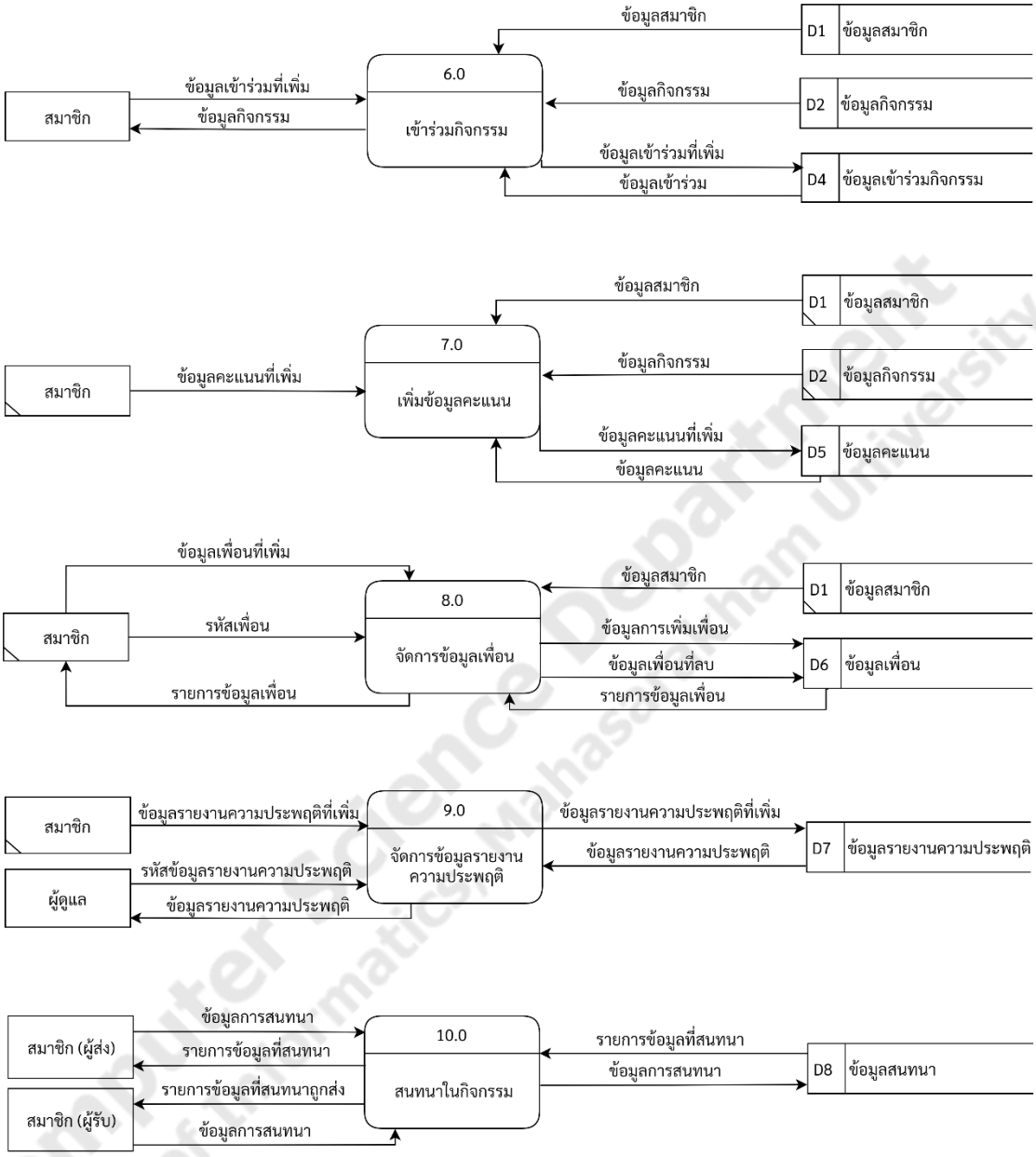
ภาพประกอบที่ 3.2 Context Diagram

3.2.2 Data Flow Diagram Level 1

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



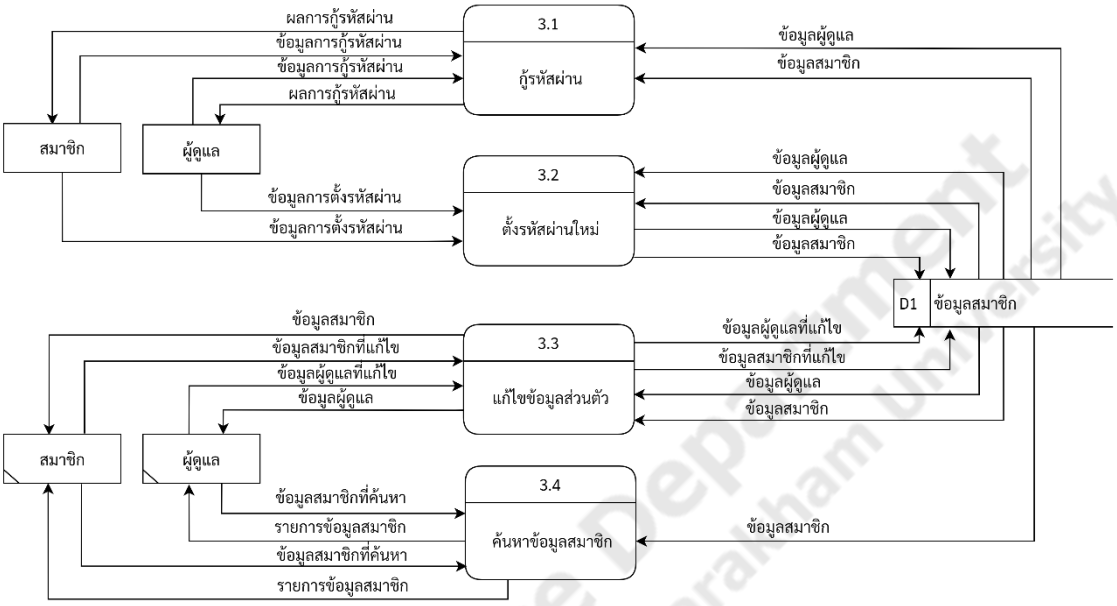
ภาพประกอบที่ 3.3 Data Flow Diagram Level 1



ภาพประกอบที่ 3.3 Data Flow Diagram Level 1 (ต่อ)

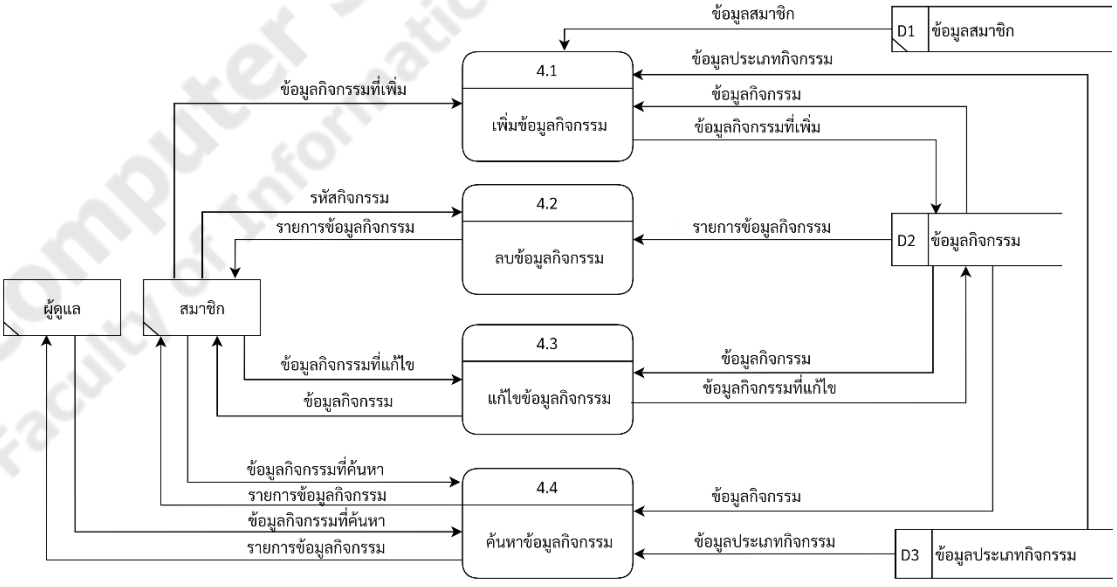
3.2.3 Data Flow Diagram Level 2

3.2.3.1 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 3



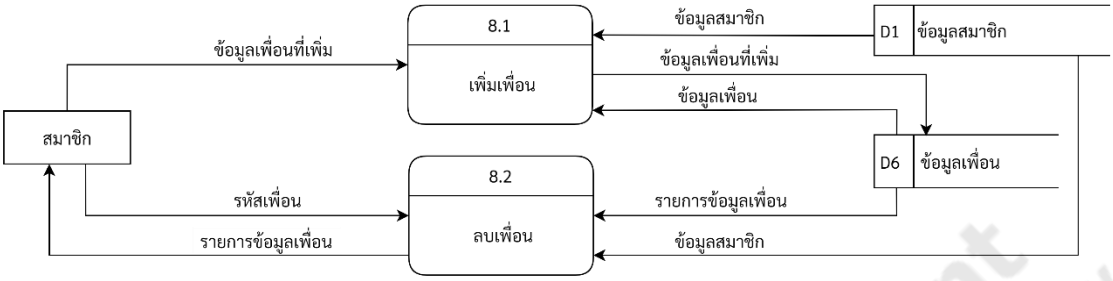
ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 Process 3

3.2.3.2 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 4



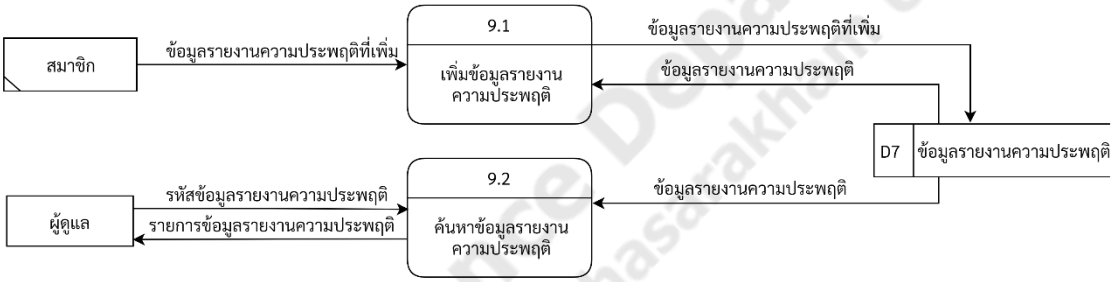
ภาพประกอบที่ 3.5 Data Flow Diagram Level 2 Process 4

3.2.3.3 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 8



ภาพประกอบที่ 3.6 Data Flow Diagram Level 2 Process 8

3.2.3.4 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 9



ภาพประกอบที่ 3.7 Data Flow Diagram Level 2 Process 9

3.3 พจนานุกรมข้อมูล (Data Dictionary)

3.3.1 External Entity Description

ตารางที่ 3.1 External Entity Description

Name	Description	Input Data Flow	Output Data flow
ผู้ใช้	ผู้ใช้งานแอปพลิเคชัน	- ข้อมูลสมาชิก	- ข้อมูลสมัครสมาชิก
ผู้ดูแล	ผู้ดูแลและจัดการระบบภายในแอปพลิเคชัน	- ผลการเข้าสู่ระบบ - ผลการกู้รหัส - ข้อมูลผู้ดูแล - ข้อมูลรายงานความประพฤติ	- ข้อมูลการเข้าสู่ระบบ - ข้อมูลรหัสผ่าน - ข้อมูลการตั้งรหัสผ่าน - ข้อมูลผู้ดูแลที่แก้ไข - รหัสข้อมูลรายงานความประพฤติ
สมาชิก	ผู้ใช้งานแอปพลิเคชัน	- ผลการเข้าสู่ระบบ - ผลการกู้รหัสผ่าน - ข้อมูลสมาชิก - รายการข้อมูลสมาชิก - ข้อมูลกิจกรรม - รายการข้อมูลกิจกรรม - รายการข้อมูลเพื่อน - รายการข้อมูลที่สนทนา - รายการข้อมูลที่สนทนาถูกส่ง	- ข้อมูลการเข้าสู่ระบบ - ข้อมูลรหัสผ่าน - ข้อมูลการกู้รหัสผ่าน - ข้อมูลการตั้งรหัสผ่าน - ข้อมูลสมาชิกที่แก้ไข - ข้อมูลสมาชิกที่ค้นหา - ข้อมูลกิจกรรมที่เพิ่ม - รหัสกิจกรรม - ข้อมูลกิจกรรมที่แก้ไข - ข้อมูลกิจกรรมที่ค้นหา - ข้อมูลกิจกรรมที่แชร์ - ข้อมูลเข้าร่วม - ข้อมูลคะแนนที่เพิ่ม - ข้อมูลเพื่อนที่เพิ่ม - รหัสเพื่อน - ข้อมูลรายงานความประพฤติที่เพิ่ม - ข้อมูลการสนทนา

3.3.2 Data Flow Description and Data Structure

ตารางที่ 3.2 Data Flow Description and Data Structure

Name	Description	Source	Destination	Data Structure
ข้อมูลสมัครสมาชิก	รายละเอียดข้อมูลการสมัครสมาชิก	ผู้ใช้	Process 1.0 สมัครสมาชิก	[รหัสสมาชิก+รูปภาพ+ชื่อ-นามสกุล+อีเมล+รหัสผ่าน+วันเดือนปีเกิด+เพศ | บัญชี Facebook]
		Process 1.0 สมัครสมาชิก	D1 ข้อมูลสมาชิก	
ข้อมูลสมาชิก	รายละเอียดข้อมูลส่วนตัวของสมาชิก	D1 ข้อมูลสมาชิก	Process 1.0 สมัครสมาชิก	{รหัสสมาชิก+รหัสเพื่อน+รหัสรายงาน+รหัสแชท+รหัสกิจกรรม+รหัสคะแนน+รหัสเข้าร่วม+สถานะ+สถานะแอดมิน+เพศ+วันเดือนปีเกิด+อีเมล+รหัสผ่าน+ชื่อ-นามสกุล+รูปภาพ+บัญชี Facebook}
		Process 1.0 สมัครสมาชิก	ผู้ใช้	
		D1 ข้อมูลสมาชิก	Process 3.1 กู้รหัสผ่าน	
		D1 ข้อมูลสมาชิก	Process 3.2 ตั้งรหัสผ่านใหม่	
		Process 3.2 ตั้งรหัสผ่านใหม่	D1 ข้อมูลสมาชิก	
		D1 ข้อมูลสมาชิก	Process 3.3 แก้ไขข้อมูลส่วนตัว	
		Process 3.3 แก้ไขข้อมูลส่วนตัว	สมาชิก	
		D1 ข้อมูลสมาชิก	Process 3.4 ค้นหาข้อมูลสมาชิก	
		D1 ข้อมูลสมาชิก	Process 4.1 เพิ่มข้อมูลกิจกรรม	

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		D1 ข้อมูลสมาชิก	Process 6.0 เข้าร่วมกิจกรรม	
		D1 ข้อมูลสมาชิก	Process 7.0 ข้อมูลคะแนน	
		D1 ข้อมูลสมาชิก	Process 8.1 เพิ่มเพื่อน	
		D1 ข้อมูลสมาชิก	Process 8.2 ลบเพื่อน	
ข้อมูลการเข้าสู่ระบบ	การตรวจสอบการเข้าสู่ระบบ	ผู้ดูแล	Process 2.0 เข้าสู่ระบบ	[อีเมล+รหัสผ่าน | บัญชี Facebook]
		D1 ข้อมูลสมาชิก	Process 2.0 เข้าสู่ระบบ	
ข้อมูลผู้ดูแล	รายละเอียดข้อมูลส่วนตัวของผู้ดูแล	D1 ข้อมูลสมาชิก	Process 3.1 กู้รหัสผ่าน	{รหัสผู้ดูแล+รหัสรายงาน+รหัสกิจกรรม+สถานะแอดมิน+เพศ+วันเดือนปีเกิด+อีเมล+รหัสผ่าน+ชื่อ-นามสกุล+รูปภาพ+บัญชี Facebook}
		D1 ข้อมูลสมาชิก	Process 3.2 ตั้งรหัสผ่านใหม่	
		Process 3.2 ตั้งรหัสผ่านใหม่	D1 ข้อมูลสมาชิก	
		D1 ข้อมูลสมาชิก	Process 3.3 แก้ไขข้อมูลส่วนตัว	
		Process 3.3 แก้ไขข้อมูลส่วนตัว	ผู้ดูแล	

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลการเข้าสู่ระบบ	การตรวจสอบการเข้าสู่ระบบ	ผู้ดูแล	Process 2.0 เข้าสู่ระบบ	[อีเมล+รหัสผ่าน | บัญชี Facebook]
		D1 ข้อมูลสมาชิก	Process 2.0 เข้าสู่ระบบ	
		สมาชิก	Process 2.0 เข้าสู่ระบบ	
		D1 ข้อมูลสมาชิก	Process 2.0 เข้าสู่ระบบ	
ผลการเข้าสู่ระบบ	ข้อมูลการเข้าสู่ระบบ	Process 2.0 เข้าสู่ระบบ	สมาชิก	ผลการเข้าสู่ระบบ
		Process 2.0 เข้าสู่ระบบ	ผู้ดูแล	
ข้อมูลการกู้รหัสผ่าน	การตรวจสอบการกู้รหัสผ่าน	สมาชิก	Process 3.1 กู้รหัสผ่าน	อีเมล+รหัสผ่านเก่า
		ผู้ดูแล	Process 3.1 กู้รหัสผ่าน	
ผลการกู้รหัสผ่าน	รายละเอียดของการกู้รหัสผ่าน	Process 3.1 กู้รหัสผ่าน	สมาชิก	รหัสผ่านเก่า
		Process 3.1 กู้รหัสผ่าน	ผู้ดูแล	
ข้อมูลการตั้งรหัสผ่าน	รายละเอียดการตั้งรหัสผ่าน	สมาชิก	Process 3.2 ตั้งรหัสผ่านใหม่	รหัสผ่านใหม่
		ผู้ดูแล	Process 3.2 ตั้งรหัสผ่านใหม่	
ข้อมูลสมาชิกที่แก้ไข	ข้อมูลสมาชิกที่แก้ไขแล้ว	สมาชิก	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสสมาชิก+รหัสกิจกรรม+รูปภาพ+ชื่อ-นามสกุล+อีเมล+รหัสผ่าน+เพศ

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 3.3 แก้ไขข้อมูลส่วนตัว	D1 ข้อมูลสมาชิก	
ข้อมูลผู้ดูแลที่แก้ไข	ข้อมูลผู้ดูแลที่แก้ไขแล้ว	ผู้ดูแล	Process 3.3 แก้ไขข้อมูลส่วนตัว	รหัสผู้ดูแล+รูปภาพ+ชื่อ+นามสกุล+อีเมล+รหัสผ่าน+เพศ
		Process 3.3 แก้ไขข้อมูลส่วนตัว	D1 ข้อมูลสมาชิก	
ข้อมูลสมาชิกที่ค้นหา	ข้อมูลสมาชิกที่ค้นหา	สมาชิก	Process 3.4 ค้นหาข้อมูลสมาชิก	ชื่อ-นามสกุล
		ผู้ดูแล	Process 3.4 ค้นหาข้อมูลสมาชิก	
รายการข้อมูลสมาชิก	ข้อมูลสมาชิก	Process 3.4 ค้นหาข้อมูลสมาชิก	สมาชิก	{รูปภาพ+ชื่อ-นามสกุล+คะแนนความสามารถของผู้ใช้+คะแนนความพึงพอใจ+กลุ่มที่เคยเข้าร่วม}
		Process 3.4 ค้นหาข้อมูลสมาชิก	ผู้ดูแล	
ข้อมูลกิจกรรมที่เพิ่ม	การเพิ่มข้อมูลกิจกรรม	สมาชิก	Process 4.1 เพิ่มข้อมูลกิจกรรม	รูปภาพกิจกรรม+ชื่อกิจกรรม+สถานที่+คำถาม+ประเภทกิจกรรม+ค่าใช้จ่าย+วันที่และเวลา+จำนวนสมาชิก
		Process 4.1 เพิ่มข้อมูลกิจกรรม	D2 ข้อมูลกิจกรรม	

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
รหัส กิจกรรม	รหัสกิจกรรม	สมาชิก	Process 4.2 ลบข้อมูลกิจกรรม	รหัสกิจกรรม
ข้อมูล กิจกรรม	รายละเอียดของข้อมูลกิจกรรม	D2 ข้อมูลกิจกรรม	Process 4.1 เพิ่มข้อมูลกิจกรรม	{รหัสกิจกรรม+รหัสคะแนน+รหัสเข้าร่วม+รหัสสมาชิก+รหัสแชท+รหัสรายงาน+รหัสประเภทกิจกรรม+รูปภาพกิจกรรม+ชื่อกิจกรรม+คำถาม+สถานะ+ระยะเวลา+ค่าใช้จ่าย+สถานที่+วันที่และเวลา+จำนวนสมาชิก}
		D2 ข้อมูลกิจกรรม	Process 4.3 แก้ไขมูลกิจกรรม	
		Process 4.3 แก้ไขมูลกิจกรรม	สมาชิก	
		D2 ข้อมูลกิจกรรม	Process 4.4 ค้นหาข้อมูลกิจกรรม	
		D2 ข้อมูลกิจกรรม	Process 5.0 แชร์ข้อมูลกิจกรรม	
		D2 ข้อมูลกิจกรรม	Process 6.0 เข้าร่วมกิจกรรม	
		Process 6.0 เข้าร่วมกิจกรรม	สมาชิก	
		D2 ข้อมูลกิจกรรม	Process 7.0 เพิ่มข้อมูลคะแนน	
รายการข้อมูลกิจกรรม	รายละเอียดของข้อมูลกิจกรรม	D2 ข้อมูลกิจกรรม	Process 4.2 ลบข้อมูลกิจกรรม	รูปภาพกิจกรรม+ชื่อกิจกรรม+รูปภาพผู้สร้าง+ชื่อผู้สร้าง+ประเภทกิจกรรม+ค่าใช้จ่าย+สถานที่+วันที่และเวลา+จำนวนสมาชิก

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 4.2 ลบข้อมูลกิจกรรม	สมาชิก	
		Process 4.4 ค้นหาข้อมูลกิจกรรม	สมาชิก	
		Process 4.4 ค้นหาข้อมูลกิจกรรม	ผู้ดูแล	
ข้อมูลกิจกรรมที่แก้ไข	ข้อมูลกิจกรรมที่แก้ไขแล้ว	สมาชิก	Process 4.3 แก้ไขมูลกิจกรรม	รหัสกิจกรรม+รูปภาพกิจกรรม+ชื่อกิจกรรม+รูปภาพผู้สร้าง+ชื่อผู้สร้าง+ประเภทกิจกรรม+ค่าใช้จ่าย+สถานที่+วันที่และเวลา+จำนวนสมาชิก
		Process 4.3 แก้ไขมูลกิจกรรม	D2 ข้อมูลกิจกรรม	
ข้อมูลกิจกรรมที่ค้นหา	ข้อมูลกิจกรรมที่ต้องการค้นหา	สมาชิก	Process 4.4 ค้นหาข้อมูลกิจกรรม	[ประเภท|สถานที่|ชื่อกิจกรรม]
		ผู้ดูแล	Process 4.4 ค้นหาข้อมูลกิจกรรม	
ข้อมูลประเภทกิจกรรม	รายละเอียดข้อมูลประเภทกิจกรรม	D3 ข้อมูลประเภทกิจกรรม	Process 4.1 เพิ่มข้อมูลกิจกรรม	รหัสประเภทกิจกรรม+รหัสกิจกรรม+ชื่อประเภท
		D3 ข้อมูลประเภทกิจกรรม	Process 4.4 ค้นหาข้อมูลกิจกรรม	

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		D3 ข้อมูลประเภทกิจกรรม	Process 5.0 แชร์ข้อมูลกิจกรรม	
ข้อมูลกิจกรรมที่แชร์	ข้อมูลกิจกรรมที่ต้องการแชร์	สมาชิก	Process 5.0 แชร์ข้อมูลกิจกรรม	รหัสกิจกรรม+รหัสประเภทกิจกรรม+รูปภาพ+ชื่อกิจกรรม+ประเภท+วันที่และเวลา+คะแนนกิจกรรม
ข้อมูลเข้าร่วมที่เพิ่ม	ข้อมูลกิจกรรมที่เข้าร่วม	สมาชิก	Process 6.0 เข้าร่วมกิจกรรม	คำตอบ+คะแนนกิจกรรม+คำขอพิกัด+พิกัด
		Process 6.0 เข้าร่วมกิจกรรม	D4 ข้อมูลเข้าร่วมกิจกรรม	
ข้อมูลเข้าร่วม	ข้อมูลการเข้าร่วมกิจกรรม	D4 ข้อมูลเข้าร่วมกิจกรรม	Process 6.0 ข้อมูลเข้าร่วมกิจกรรม	รหัสเข้าร่วม+รหัสสมาชิก+รหัสกิจกรรม+สถานะ+คะแนนกิจกรรม+คำตอบ+คำขอพิกัด+พิกัด
ข้อมูลคะแนนที่เพิ่ม	การเพิ่มข้อมูลคะแนน	สมาชิก	Process 7.0 เพิ่มข้อมูลคะแนน	รหัสข้อมูลคะแนน+(ชื่อข้อมูลคะแนน)
		Process 7.0 เพิ่มข้อมูลคะแนน	D5 ข้อมูลคะแนน	
ข้อมูลคะแนน	รายละเอียดข้อมูลคะแนน	D5 ข้อมูลคะแนน	Process 7.0 ข้อมูลคะแนน	รหัสข้อมูลคะแนน+รหัสกิจกรรม+รหัสสมาชิก+ชื่อข้อมูลคะแนน

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลเพื่อนที่เพิ่ม	การเพิ่มเพื่อน	สมาชิก	Process 8.1 เพิ่มเพื่อน	(รหัสเพื่อน+รหัสสมาชิก+ข้อมูลเพื่อน)
		Process 8.1 เพิ่มเพื่อน	D6 ข้อมูลเพื่อน	
รหัสเพื่อน	รหัสเพื่อน	สมาชิก	Process 8.2 ลบเพื่อน	รหัสเพื่อน
ข้อมูลเพื่อน	ข้อมูลเพื่อน	D6 ข้อมูลเพื่อน	Process 8.1 เพิ่มเพื่อน	{รหัสเพื่อน+รหัสสมาชิก+สถานะ}
รายการข้อมูลเพื่อน	ข้อมูลเพื่อน	D6 ข้อมูลเพื่อน	Process 8.2 ลบเพื่อน	{สถานะการเป็นเพื่อน}
		Process 8.2 ลบเพื่อน	สมาชิก	
ข้อมูลรายงานความประพฤติที่เพิ่ม	ข้อมูลรายงานความประพฤติที่เพิ่ม	สมาชิก	Process 9.1 เพิ่มข้อมูลรายงานความประพฤติ	{รหัสรายงาน+รหัสสมาชิก+รหัสกิจกรรม+รายละเอียด+วันที่และเวลา+สถานะ+รูปภาพที่ 1+รูปภาพที่ 2}
		Process 9.1 เพิ่มข้อมูลรายงานความประพฤติ	D7 ข้อมูลรายงานความประพฤติ	
ข้อมูลรายงานความประพฤติ	ข้อมูลรายงานความประพฤติ	D7 ข้อมูลรายงานความประพฤติ	Process 9.1 เพิ่มข้อมูลรายงานความประพฤติ	{รหัสรายงาน+รหัสสมาชิก+รหัสกิจกรรม+รายละเอียด+วันที่และเวลา+สถานะ+รูปภาพที่ 1+รูปภาพที่ 2}
		D7 ข้อมูลรายงานความประพฤติ	Process 9.2 ค้นหาข้อมูลรายงานความประพฤติ	

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

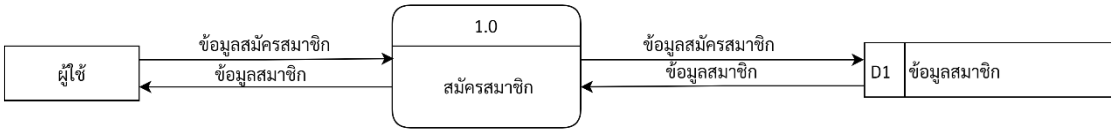
Name	Description	Source	Destination	Data Structure
รายการข้อมูลรายงานความประพฤติ	รายการข้อมูลรายงานความประพฤติ	Process 9.2 ค้นหาข้อมูลรายงานความประพฤติ	ผู้ดูแล	{รายละเอียด+วันที่และเวลา+สถานะ+รูปภาพที่ 1+รูปภาพที่ 2}
รหัสข้อมูลรายงานความประพฤติ	รหัสข้อมูลรายความประพฤติ	ผู้ดูแล	Process 9.2 ค้นหาข้อมูลรายงานความประพฤติ	รหัสข้อมูลรายงาน
ข้อมูลการสนทนา	ข้อมูลสนทนา	สมาชิก (ผู้ส่ง)	Process 10.0 สนทนาในกิจกรรม	{รหัสข้อความ+รหัสสมาชิก+รหัสกิจกรรม+ข้อความ+เวลาที่ส่ง+รูปภาพ}
		Process 10.0 สนทนาในกิจกรรม	D8 ข้อมูลสนทนา	
		สมาชิก (ผู้รับ)	Process 10.0 สนทนาในกิจกรรม	
รายการข้อมูลที่สนทนา	ข้อมูลสนทนา	D8 ข้อมูลสนทนา	Process 10.0 สนทนาในกิจกรรม	{ข้อความ+เวลาที่ส่ง+รูปภาพ}
		Process 10.0 สนทนาในกิจกรรม	สมาชิก (ผู้ส่ง)	
รายการข้อมูลที่สนทนาถูกส่ง	ข้อมูลสนทนา	Process 10.0 สนทนาในกิจกรรม	สมาชิก (ผู้รับ)	{รหัสข้อความ+รหัสสมาชิก+รหัสกิจกรรม+ข้อความ+เวลาที่ส่ง+รูปภาพ}

3.3.3 Data Store Description and Data Structure

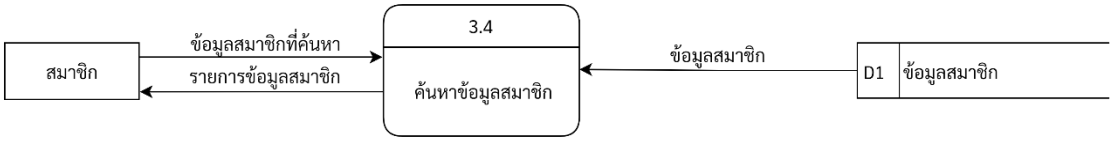
ตารางที่ 3.3 Data Store Description and Data Structure

ID	Data Store Name	Description	Data Structure
D1	ข้อมูลสมาชิก	เป็นที่เก็บข้อมูลสมาชิก	รหัสสมาชิก+สถานะผู้ดูแล+ชื่อผู้ใช้งาน+รหัสผ่าน+อีเมล+วันเดือนปีเกิด+เพศ+รูปภาพ+สถานะ+บัญชี Facebook
D2	ข้อมูลกิจกรรม	เป็นที่เก็บข้อมูลกลุ่มกิจกรรม	รหัสกิจกรรม+ชื่อกิจกรรม+รูปภาพ+สถานที่+คำถาม+คำขอพิกัด+สถานะ+วันที่และเวลา+ระยะเวลา+ค่าใช้จ่าย+จำนวนคน
D3	ข้อมูลประเภทกิจกรรม	เป็นที่เก็บข้อมูลประเภทกิจกรรม	รหัสประเภทกิจกรรม+ชื่อประเภทกิจกรรม
D4	ข้อมูลเข้าร่วมกิจกรรม	เป็นที่เก็บข้อมูลการเข้าร่วมกิจกรรม	รหัสเข้าร่วมกิจกรรม+สถานะ+คะแนนกิจกรรม+คำตอบ+คำขอพิกัด+จีพีเอส
D5	ข้อมูลคะแนน	เป็นที่เก็บข้อมูลคะแนนความชอบ	รหัสคะแนน+ชื่อคะแนน
D6	ข้อมูลเพื่อน	เป็นที่เก็บข้อมูลเพื่อน	รหัสเพื่อน+สถานะ
D7	ข้อมูลรายงานความประพฤติ	เป็นที่เก็บข้อมูลการรายงาน	รหัสรายงาน+รูปภาพที่1+รูปภาพที่2+สถานะ+วันที่และเวลา+รายละเอียด
D8	ข้อมูลสนทนา	เป็นที่เก็บข้อมูลการสนทนา	รหัสข้อความ+รหัสกิจกรรม+รหัสผู้ส่ง+ชื่อผู้ส่ง+ข้อความที่สนทนา+วันที่และเวลา

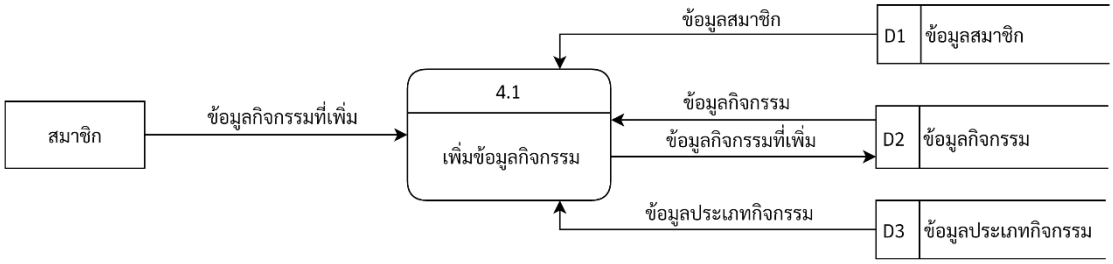
3.3.4 Process Description



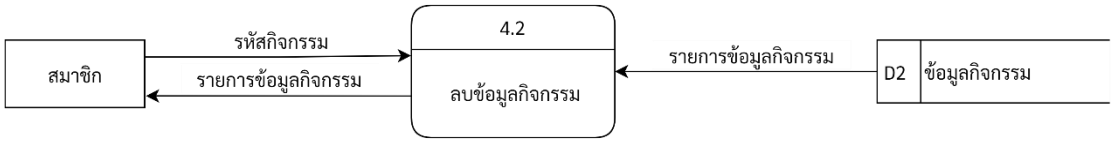
Number	1.0
Name	สมัครสมาชิก
Description	เพื่อทำการเพิ่มข้อมูลสมาชิกเข้าสู่ฐานข้อมูล
Input data flow	- ข้อมูลสมัครสมาชิก - ข้อมูลสมาชิก
Output data flow	- ข้อมูลสมัครสมาชิก - ข้อมูลสมาชิก
Process description	เริ่มต้น 1. รับข้อมูลที่ผู้ใช้สมัครสมาชิก 2. ตรวจสอบข้อมูลว่าครบถ้วนและถูกต้องหรือไม่ 2.1 ถ้าข้อมูลครบถ้วนและถูกต้องทุกรายการให้แสดงข้อความ “สมัครสมาชิกสำเร็จ” 2.2 ถ้าข้อมูลครบถ้วนแต่ไม่ถูกต้องให้แสดงข้อความ “กรุณาป้อนข้อมูลให้ถูกต้อง” 2.3 ถ้าข้อมูลไม่ครบถ้วนให้แสดงข้อความ “กรุณาป้อนข้อมูลให้ครบถ้วน” 3. ตรวจสอบในแฟ้มข้อมูลสมาชิกว่ามีข้อมูลผู้ใช้แล้วหรือไม่ 3.1 ถ้ามีชื่อผู้ใช้แล้วให้แสดงข้อความ “พบข้อมูลซ้ำในระบบ” 3.2 ถ้าไม่มีชื่อผู้ใช้ในแฟ้มข้อมูลให้บันทึกข้อมูลสมาชิกลงในแฟ้มข้อมูล จบการทำงาน



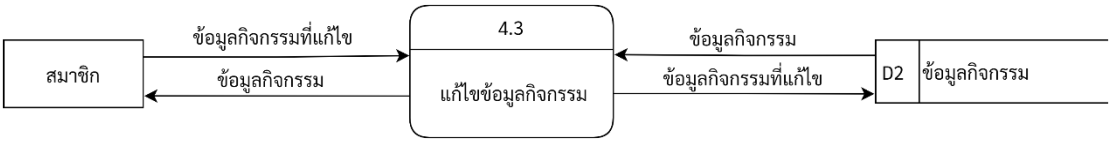
Number	3.4
Name	ค้นหาข้อมูลสมาชิก
Description	เพื่อทำการค้นหาข้อมูลสมาชิก
Input data flow	- ข้อมูลสมาชิกที่ค้นหา - ข้อมูลสมาชิก
Output data flow	- รายการข้อมูลสมาชิก
Process description	เริ่มต้น 1. รับข้อมูลการค้นหาข้อมูลสมาชิก 2. ค้นหาข้อมูลสมาชิกที่รับเข้ามาจากแฟ้มข้อมูลสมาชิก 3. ตรวจสอบในแฟ้มข้อมูลสมาชิก 3.1 ถ้ามีชื่อสมาชิกแล้วให้แสดงข้อมูลสมาชิก 3.2 ถ้าไม่มีชื่อสมาชิกให้แสดงข้อความ “ไม่พบผู้ใช้งานนี้” จบการทำงาน



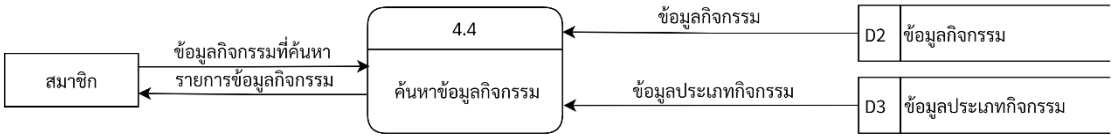
Number	4.1
Name	เพิ่มข้อมูลกิจกรรม
Description	เพื่อทำการเพิ่มข้อมูลกิจกรรมลงฐานข้อมูล
Input data flow	- ข้อมูลกิจกรรมที่เพิ่ม - ข้อมูลสมาชิก - ข้อมูลกิจกรรม - ข้อมูลประเภทกิจกรรม
Output data flow	- ข้อมูลกิจกรรมที่เพิ่ม
Process description	เริ่มต้น 1. รับข้อมูลกิจกรรมที่ต้องการเพิ่ม 2. ตรวจสอบข้อมูลว่าครบถ้วนและถูกต้องหรือไม่ 2.1 ถ้าข้อมูลครบถ้วนและถูกต้องทุกรายการให้แสดงข้อความ “สำเร็จ” 2.2 ถ้าข้อมูลครบถ้วนแต่ไม่ถูกต้องให้แสดงข้อความ “กรุณาป้อนข้อมูลให้ถูกต้อง” 2.3 ถ้าข้อมูลไม่ครบถ้วนให้แสดงข้อความ “กรุณาป้อนข้อมูลให้ครบถ้วน” 3. ตรวจสอบในแฟ้มข้อมูลกิจกรรมว่ามีข้อมูลกิจกรรมแล้วหรือไม่ 3.1 ถ้ามีชื่อกิจกรรมแล้วให้แสดงข้อความ “พบข้อมูลซ้ำในระบบ” 3.2 ถ้าไม่มีชื่อกิจกรรมในแฟ้มข้อมูลให้บันทึกข้อมูลกิจกรรมลงในแฟ้มข้อมูล จบการทำงาน



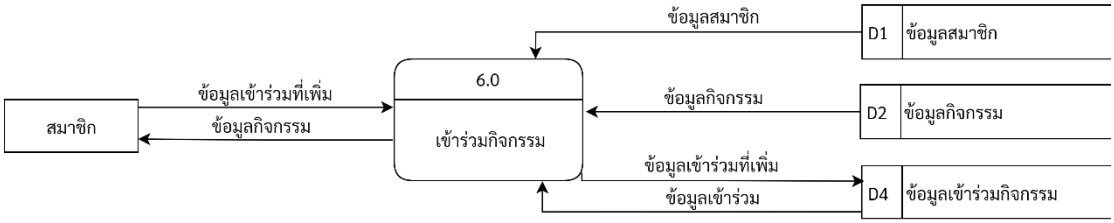
Number	4.2
Name	ลบข้อมูลกิจกรรม
Description	เพื่อทำการลบข้อมูลกิจกรรมออกจากฐานข้อมูล
Input data flow	- รหัสกิจกรรม - รายการข้อมูลกิจกรรม
Output data flow	- รายการข้อมูลกิจกรรม
Process description	เริ่มต้น 1. รับข้อมูลกิจกรรมที่ต้องการลบ 2. ตรวจสอบข้อมูลจากแฟ้มข้อมูลกิจกรรม 2.1 ถ้ามีข้อมูลกิจกรรมให้ทำการลบข้อมูลกิจกรรมนั้นและแสดงข้อความ “ลบสำเร็จ” 2.2 ถ้าไม่มีข้อมูลกิจกรรมให้แสดงข้อความ “ไม่พบกิจกรรมนี้” จบการทำงาน



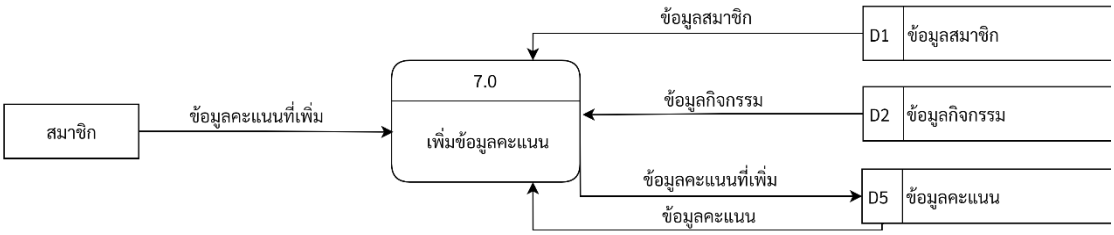
Number	4.3
Name	แก้ไขข้อมูลกิจกรรม
Description	เพื่อทำการแก้ไขข้อมูลกิจกรรม
Input data flow	- ข้อมูลกิจกรรมที่แก้ไข - ข้อมูลกิจกรรม
Output data flow	- ข้อมูลกิจกรรมที่แก้ไข - ข้อมูลกิจกรรม
Process description	เริ่มต้น 1. รับข้อมูลกิจกรรมที่ต้องการแก้ไข 2. ตรวจสอบข้อมูลว่าที่กรอกเข้ามาถูกต้องหรือไม่ 2.1 ถ้าข้อมูลครบถ้วนและถูกต้องทุกรายการให้ทำการบันทึกข้อมูลลงในแฟ้มข้อมูลกิจกรรมและแสดงข้อความ “แก้ไขสำเร็จ” 2.2 ถ้าข้อมูลไม่ถูกต้องครบถ้วนให้แสดงข้อความ “กรุณาป้อนข้อมูลให้ถูกต้อง” จบการทำงาน



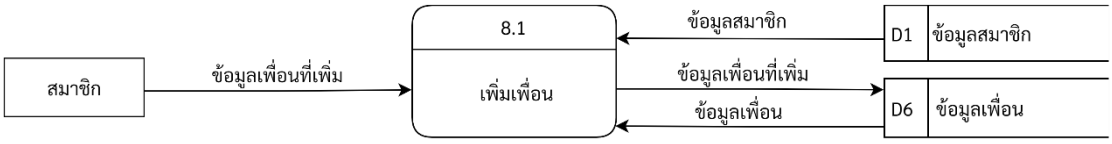
Number	4.4
Name	ค้นหาข้อมูลกิจกรรม
Description	เพื่อทำการค้นหาข้อมูลกิจกรรม
Input data flow	- ข้อมูลกิจกรรมที่ค้นหา - ข้อมูลกิจกรรม - ข้อมูลประเภทกิจกรรม
Output data flow	- รายการข้อมูลกิจกรรม
Process description	เริ่มต้น 1. รับข้อมูลการค้นหาข้อมูลกิจกรรม 2. ค้นหาข้อมูลกิจกรรมที่รับเข้ามาหรือค้นหาจากประเภทของกิจกรรมจากแฟ้มข้อมูลกิจกรรมและแฟ้มข้อมูลประเภทกิจกรรม 3. ตรวจสอบ 3.1 ถ้ามีชื่อกิจกรรมแล้วให้แสดงหน้ากิจกรรม 3.2 ถ้าไม่มีชื่อกิจกรรมให้แสดงข้อความ “ไม่พบกิจกรรมนี้” จบการทำงาน



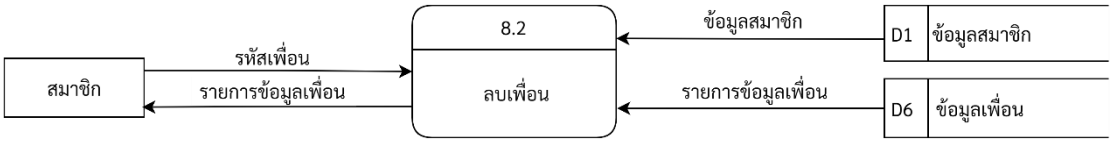
Number	6.0
Name	เข้าร่วมกิจกรรม
Description	เพื่อทำการเข้าร่วมกิจกรรมที่สนใจ
Input data flow	- ข้อมูลเข้าร่วมที่เพิ่ม - ข้อมูลสมาชิก - ข้อมูลกิจกรรม - ข้อมูลเข้าร่วม
Output data flow	- ข้อมูลกิจกรรม - ข้อมูลเข้าร่วมที่เพิ่ม
Process description	เริ่มต้น 1. รับข้อมูลการเข้าร่วมกิจกรรม 2. ตรวจสอบข้อมูลเข้าร่วมกิจกรรมจากแฟ้มข้อมูลเข้าร่วมกิจกรรม 2.1 ถ้ามีการตอบคำถามจึงจะสามารถเข้าร่วมกิจกรรมได้ 2.2 ถ้าไม่มีการตอบคำถามจะไม่สามารถเข้าร่วมกิจกรรมได้ จบการทำงาน



Number	7.0
Name	เพิ่มข้อมูลคะแนน
Description	เพื่อทำการเพิ่มคะแนนกิจกรรมหรือคะแนนความพึงพอใจหลังจบกิจกรรม
Input data flow	- ข้อมูลคะแนนที่เพิ่ม - ข้อมูลสมาชิก - ข้อมูลกิจกรรม - ข้อมูลคะแนน
Output data flow	- ข้อมูลคะแนนที่เพิ่ม
Process description	เริ่มต้น 1. รับข้อมูลเพิ่มคะแนนเข้ามา 2. ตรวจสอบคะแนนที่รับเข้ามาว่าคือคะแนนทักษะ คะแนนความพึงพอใจ หรือคะแนนกิจกรรม 2.1 ถ้ากรอกข้อมูลคะแนนถูกต้องให้แสดงข้อความ “เพิ่มคะแนนสำเร็จ” จบการทำงาน

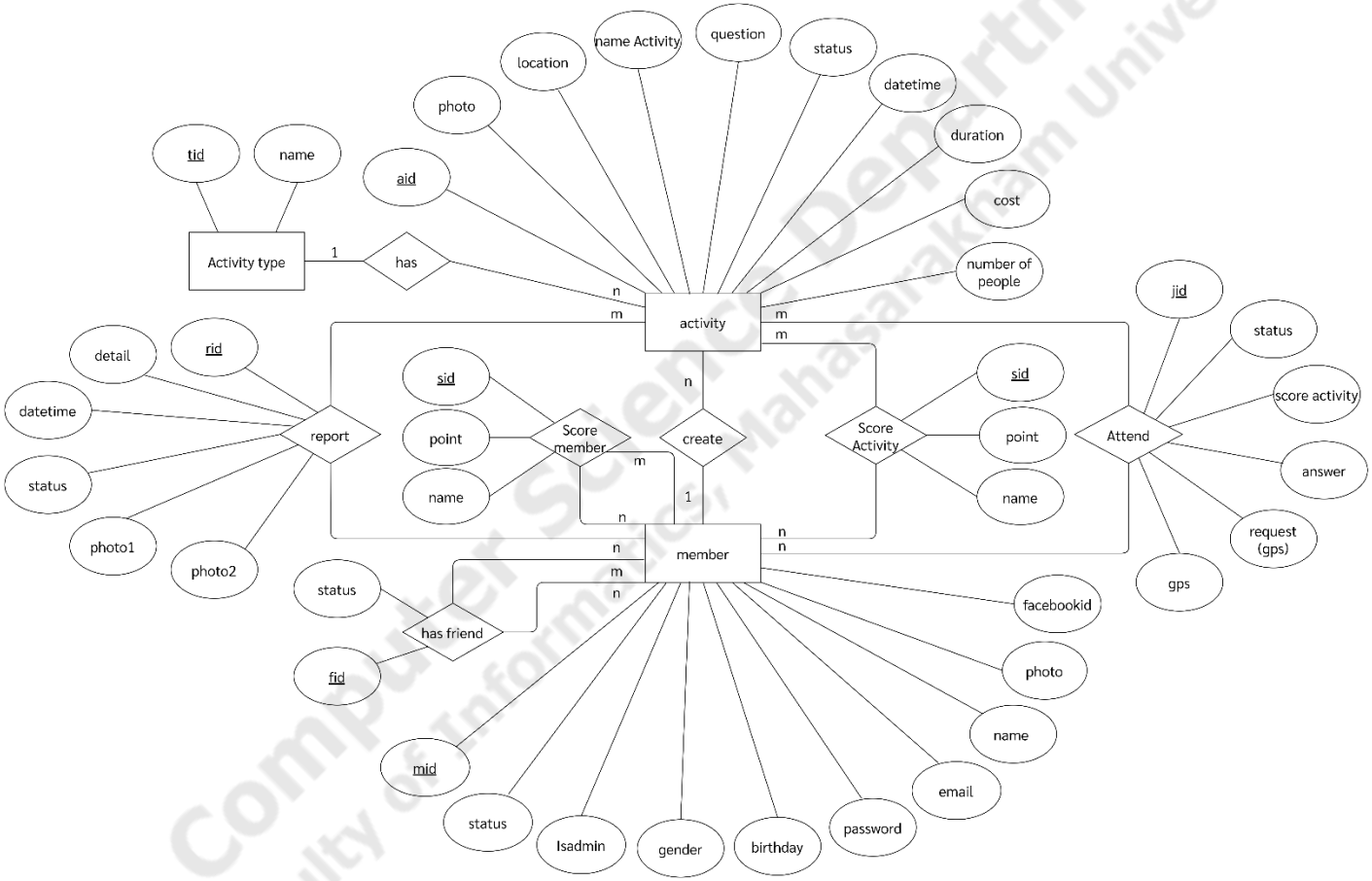


Number	8.1
Name	เพิ่มเพื่อน
Description	เพื่อทำการเพิ่มเพื่อนในกิจกรรม
Input data flow	- ข้อมูลเพื่อนที่เพิ่ม - ข้อมูลสมาชิก - ข้อมูลเพื่อน
Output data flow	- ข้อมูลเพื่อนที่เพิ่ม
Process description	เริ่มต้น 1. รับข้อมูลเพื่อนเข้ามา 2. ตรวจสอบข้อมูลเพื่อนที่รับเข้ามาว่าถูกต้องหรือไม่ 2.1 ถ้าข้อมูลเพื่อนถูกต้องให้แสดงข้อความ “เพิ่มเพื่อนสำเร็จ” 2.2 ถ้าข้อมูลเพื่อนไม่ถูกต้องให้แสดงข้อความ “ไม่พบข้อมูลเพื่อน” จบการทำงาน



Number	8.2
Name	ลบเพื่อน
Description	เพื่อทำการลบเพื่อน
Input data flow	- รหัสเพื่อน - ข้อมูลสมาชิก - รายการข้อมูลเพื่อน
Output data flow	- รายการข้อมูลเพื่อน
Process description	เริ่มต้น 1. รับข้อมูลเพื่อนเข้ามา 2. ตรวจสอบข้อมูลเพื่อนที่รับเข้ามาว่าถูกต้องหรือไม่ 2.1 ถ้าข้อมูลเพื่อนถูกต้องให้แสดงข้อความ “ลบเพื่อนสำเร็จ” 2.2 ถ้าข้อมูลเพื่อนไม่ถูกต้องให้แสดงข้อความ “ไม่พบข้อมูลเพื่อน” จบการทำงาน

3.4 ความสัมพันธ์ (Entity Relationship Diagram)



ภาพประกอบที่ 3.8 แผนภาพ Entity Relationship Diagram

3.5 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ข้อมูลสมาชิก

Id	Column	Type	Description	Example Data	Constraints
1	m_id	int	รหัสสมาชิก	1	PK
2	m_status	int	สถานะสมาชิก (0=โดนระงับ, 1=ใช้งานได้)	0	Can be only 0 and 1
3	m_Isadmin	int	สถานะสมาชิก (0=เป็นสมาชิก ,1=เป็นแอดมิน)	1	Can be only 0 and 1
4	m_gender	varchar (5)	เพศ	หญิง	Not null
5	m_birthday	date	วันเดือนปีเกิด	24/08/2564	Not null
6	m_password	varchar (255)	รหัสผ่าน	admin123	Not null
7	m_email	varchar (50)	อีเมล	admin123 @gmail.com	Not null
8	m_name	varchar (50)	ชื่อ	นิสากรณ์ พลพิมพ์	Not null
9	m_photo	varchar (50)	รูปภาพ	-	Null
10	m_facebookid	varchar (50)	บัญชีเฟซบุ๊ก	-	Not null
11	playerID	int	รหัสสำหรับรับการแจ้งเตือน	e725278a-6e30-11ec-a8a9-32f918caeeed	Not null

ตารางที่ 3.5 ข้อมูลกิจกรรม

Id	Column	Type	Description	Example Data	Constraints
1	a_id	int	รหัสกิจกรรม	1	PK
2	a_photo	varchar (255)	รูปภาพ	-	Null
3	a_location	varchar (255)	สถานที่	ร้านแชมป์	Not null
4	a_name	varchar (100)	ชื่อกิจกรรม	กินหมูทะ	Not null

ตารางที่ 3.5 ข้อมูลกิจกรรม (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
5	a_question	varchar (255)	คำถาม	กินจุไหม	Not null
6	a_status	int	สถานะกิจกรรม (0=โดนระงับ, 1=ใช้งานได้)	1	Can be only 0 and 1
7	a_datetime	date	วันที่เวลา	12/22/2021 16:21:30	Not null
8	a_duration	int	ระยะเวลา	2 ชั่วโมง	Not null
9	a_cost	int	ค่าใช้จ่าย	159	Not null
10	a_number	int	จำนวนคน	4	Not null
11	m_id	int	รหัสสมาชิก	1	FK reference from member(m_id) ON DELETE CASCADE ON UPDATE CASCADE
12	t_id	int	รหัสประเภทกิจกรรม	1	FK reference from activitytype (t_id) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.6 ข้อมูลประเภทกิจกรรม

Id	Column	Type	Description	Example Data	Constraints
1	t_id	int	รหัสประเภท	1	PK
2	t_name	varchar (255)	ชื่อของประเภท	กีฬา	Not null

ตารางที่ 3.7 ข้อมูลเข้าร่วมกิจกรรม

Id	Column	Type	Description	Example Data	Constraints
1	j_id	int	รหัสเข้าร่วม	1	PK
2	j_status	int	สถานะการเข้าร่วม (0=ยังไม่เข้าร่วม, 1=เข้าร่วมแล้ว)	0	Can be only 0 and 1
3	j_score	int	คะแนนกิจกรรม	1	Not null
4	j_answer	varchar (255)	คำตอบ	คุณต้องการเข้าร่วมหรือไหม	Not null
5	j_request	int	คำขอพิกัด	ต้องการแชร์พิกัดหรือไหม	Not null
6	j_gps	int	พิกัด	-	Not null
7	m_id	int	รหัสสมาชิก	1	FK reference from member (m_id) ON DELETE CASCADE ON UPDATE CASCADE
8	a_id	int	รหัสกิจกรรม	1	FK reference from activity (a_id) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.8 ข้อมูลเพื่อน

Id	Column	Type	Description	Example Data	Constraints
1	f_id	int	รหัสเพื่อน	1	PK
2	f_status	int	สถานะการเป็นเพื่อน (0=ไม่เป็นเพื่อน, 1=เป็นพื่อน)	0	Can be only 0 and 1
3	m_id1	int	รหัสสมาชิกคนที่ 1	1	FK reference from member (m_id) ON DELETE CASCADE ON UPDATE CASCADE
4	m_id2	int	รหัสสมาชิกคนที่ 2	1	FK reference from member (m_id) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.9 ข้อมูลการรายงาน

Id	Column	Type	Description	Example Data	Constraints
1	r_id	int	รหัสแชท	1	PK
2	r_detail	varchar (255)	รายละเอียดข้อความการรายงาน	บัญชีของ นิสากรณ์ พลพิมพ์ มีพฤติกรรมที่น่าสงสัย	Not null

ตารางที่ 3.9 ข้อมูลการรายงาน (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
3	r_datetime	datetime	เวลาที่ส่งรายงาน	12/22/2021 16:21:30	Not null
4	r_status	int	สถานะการเข้าร่วม (0=รายงานไม่สำเร็จ, 1=รายงานสำเร็จ)	0	Can be only 0 and 1
5	r_photo1	varchar (255)	รูปภาพ	-	Null
6	r_photo2	varchar (255)	รูปภาพ	-	Null
7	m_id	int	รหัสสมาชิก	1	FK reference from member (m_id) ON DELETE CASCADE ON UPDATE CASCADE
8	a_id	int	รหัสกิจกรรม	1	FK reference from activity (a_id) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.10 ข้อมูลคะแนน

Id	Column	Type	Description	Example Data	Constraints
1	s_id	int	รหัสเพื่อน	1	PK
2	s_name	varchar (255)	ชื่อคะแนน	ทักษะ	Not null

ตารางที่ 3.10 ข้อมูลคะแนน (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
3	s_point	varchar (50)	คะแนน	5	-
4	r_status	int	สถานะการเข้าร่วม (0=รายงานไม่สำเร็จ, 1=รายงานสำเร็จ)	0	Can be only 0 and 1
5	r_photo1	varchar (255)	รูปภาพ	-	Null

ตารางที่ 3.11 Document ใน Collection ข้อมูล Author

Attribute	Type	Description	Example Data
id	int	รหัสของผู้ส่งข้อความ	1
first	String	ชื่อผู้ส่งข้อความ	นิสากรณ์

ตารางที่ 3.12 Document ใน Collection ข้อมูลข้อความที่เป็น image

Attribute	Type	Description	Example Data
author	object	ข้อมูลผู้ส่งข้อความ	{id}, {name}
createdAt	int	วันที่ส่งข้อความ	1640685887960
id	int	รหัสข้อความ	2e75e0cf-8fbf-48f-9c9f-cf1c3f2a588b
name	String	ชื่อของรูปภาพ	346cf44b.jpg
type	String	ประเภทของข้อความ	image
url	String	ที่อยู่ของรูปภาพ	http://firebasestorage.googleapis.com
height	int	ความสูงของรูปภาพ	1295
width	int	ความกว้างของรูปภาพ	865
size	int	ขนาดของรูปภาพ	74430

ตารางที่ 3.13 Document ใน Collection ข้อมูลข้อความที่เป็น text

Attribute	Type	Description	Example Data
author	object	ข้อมูลผู้ส่งข้อความ	{id}, {name}
createdAt	int	วันที่ส่งข้อความ	1640760390041
id	int	รหัสข้อความ	626626d0-08ab-4608-866c-90c0300dabff
type	String	ประเภทของข้อความ	text
text	String	เนื้อหาของข้อความ	ใกล้ถึงเวลากิจกรรมแล้ว

3.6 การพัฒนาระบบ

3.6.1 การเข้าสู่ระบบโดยใช้ Facebook Email และ Password

Flutter Facebook login คือ ชุดคำสั่งที่ช่วยให้ผู้ใช้เข้าสู่ระบบแอปพลิเคชันด้วย Facebook เมื่อเข้าสู่แอปพลิเคชันด้วย Facebook แล้วผู้ใช้จะสามารถให้สิทธิ์การอนุญาตเข้าแอปพลิเคชันเพื่อให้รับข้อมูลหรือดำเนินการบน Facebook สามารถเข้าสู่ระบบใช้งานบน IOS, Android, Web และ Application บน Desktop เข้าสู่ระบบได้ง่ายๆเพื่อการยืนยันตัวตนและการเข้าถึงข้อมูลการแจ้งเตือนเมื่อผู้ใช้เข้าสู่ระบบผ่าน โดยจะเรียกใช้ชุดคำสั่งของ Facebook ในการเข้าสู่ระบบและใช้สิทธิ์อนุญาตในการเข้าสู่ระบบ

```
3  public class LogInDTO {
4      private String email;
5      private String password;
6      private String mFacebookid;
```

ภาพประกอบที่ 3.9 ประกาศคลาส LogInDTO พร้อมทั้งสร้าง Attribute ของ Entity

บรรทัดที่ 3-6 ประกาศคลาส LogInDTO พร้อมทั้งสร้าง Attribute ของ Entity สำหรับรับข้อมูลจาก Front End

```java
package net.csmsu.playMateAppBack.controller;

import java.util.List;

@RestController
public class UserController {
    @Autowired
    Loginservice loginservice;

    @GetMapping("/member")
    public List<Member> getAllUser() {
        return loginservice.getAllUsers();
    }
```

ภาพประกอบที่ 3.10 สร้างเมธอด getAllUser

บรรทัดที่ 18 ระบุ @RestController เพื่อให้รู้ว่าคลาสนี้คือ Restful Controller

บรรทัดที่ 20 ระบุ @AutoWired เข้ามา

บรรทัดที่ 23 ระบุ @GetMapping เพื่อระบุ Path ให้กับ Controller

บรรทัดที่ 24 สร้างเมธอดของ Path ที่สร้างขึ้นนั้น getAllUser ซึ่งจะเป็นการไปเรียก getAllUsers ของ loginservice

```java
    //Login facebook
    @PostMapping(value = "/login",
            consumes = MediaType.APPLICATION_JSON_VALUE,
            produces = MediaType.APPLICATION_JSON_VALUE)
    public List<Member> getUserByFacebookId(@RequestBody LogInDTO logInDTO){
        System.out.println(logInDTO.getmFacebookid());
        return loginservice.getUserByFacebookId(logInDTO);
    }
```

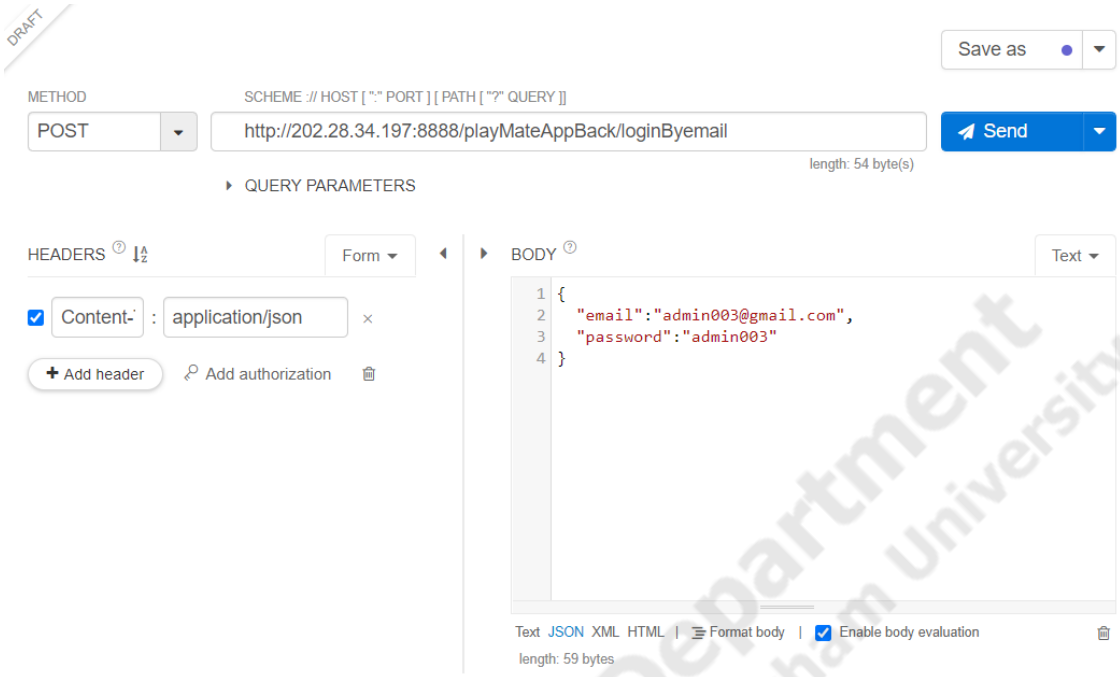
ภาพประกอบที่ 3.11 Controller ส่วนเข้าสู่ระบบด้วย Facebook

บรรทัดที่ 29 สร้าง Controller โดยใช้ @PostMapping เพื่อรับการร้องขอแบบ Post

บรรทัดที่ 30 ระบุ consumes เป็น JSON เพื่อระบุว่าข้อมูลที่เข้ามาจะเป็นรูปแบบ JSON

บรรทัดที่ 32 เรียกใช้เมธอด getUserByFacebookId จาก service

บรรทัดที่ 34 ส่งค่า DTO ให้กับเมธอด



ภาพประกอบที่ 3.12 เรียกใช้ Service ล็อคอินโดยใช้ id จาก Facebook

```
24    public List<Member> getUserByFacebookId(LogInDTO logInDTO) {
25        List<Member> users = userRepository.findBymFacebookid(logInDTO.getmFacebookid());
26        for(Member user : users) {
27            user.setMPassword(null);
28        }
29        return users;
30    }
```

ภาพประกอบที่ 3.13 Service ในการเข้าสู่ระบบด้วย Facebook

บรรทัดที่ 24 ประกาศเมธอดในการส่งข้อมูล ถูกเรียกจาก getUserByFacebookId

บรรทัดที่ 25 ประกาศตัวแปร users ที่มีค่าจากเมธอด findBymFacebookid โดยจะส่งค่า facebookId

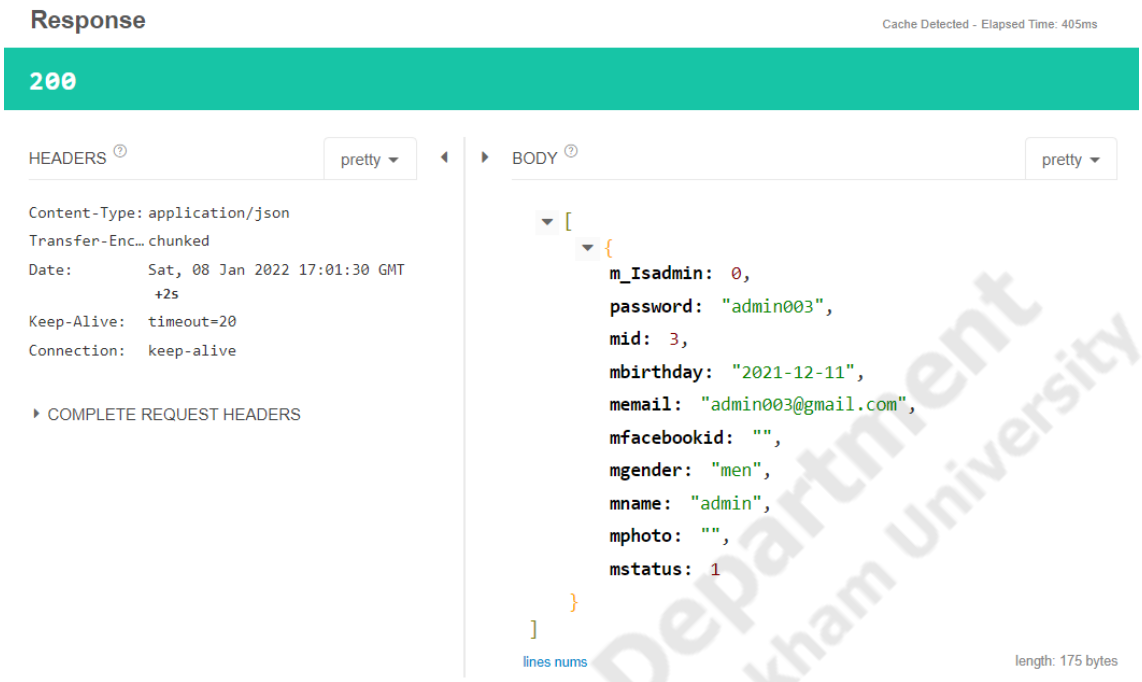
บรรทัดที่ 26-27 นำข้อมูลที่ถูกส่งกลับมา มากำหนดค่าให้รหัสผ่านมีค่าเป็น null

บรรทัดที่ 29 ส่งค่าข้อมูลกลับไปที่ getUserByFacebookId

```
10    public List<Member> findBymFacebookid(String fkId);
```

ภาพประกอบที่ 3.14 Repository ในการเข้าสู่ระบบด้วย Facebook

บรรทัดที่ 10 ประกาศเมธอดในการส่งข้อมูล Member ถูกเรียกจาก getUserByFacebookId ที่ส่งค่า fkId เป็นค่าของ facebookId ของผู้ใช้



ภาพประกอบที่ 3.15 ข้อมูล user ที่ได้จากการเรียกใช้ Service

```
210  Future<void> LoginByFacebook (String fid) async{
211    var json = {"mfacebookid": fid};
212    String value = await rootBundle.loadString('assets/config/config.yaml');
213    dynamic conf = loadYaml(value);
214    String url = conf['server']['host'];
215    String port = conf['server']['port'];
216
217    var response = await http.post(Uri.parse("http://$url:$port/playMateAppBack/login"),
218    headers: <String, String>{
219      'Content-Type':'application/json; charset=UTF-8',
220    },
221      body: jsonEncode(json));
222      //log(response.statusCode.toString());
223
224      if(response.statusCode == 200){
225        ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text("Login Success")));
226        Navigator.push(context, MaterialPageRoute(builder: (context) => HomePage()));
227      } else {
228        ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text("Login Error")));
229      }
230  }
```

ภาพประกอบที่ 3.16 ฟังก์ชันการเข้าสู่ระบบผ่าน Facebook

บรรทัดที่ 211 ประกาศตัวแปร json มาเก็บค่า Facebook ID

บรรทัดที่ 212 ประกาศตัวแปร value มาเก็บค่าของข้อมูลใน ‘assets/config/config.yaml’ ซึ่งโหลดมาเก็บเป็น String

บรรทัดที่ 213 ประกาศตัวแปร conf เพื่อดึงค่าจากตัวแปร value และโหลดมาเก็บเป็นไฟล์ yaml

บรรทัดที่ 214 ประกาศตัวแปร url ดึงค่าจาก server host

บรรทัดที่ 215 ประกาศตัวแปร port ดึงค่าจาก server port

บรรทัดที่ 217 ทำการเรียก service โดยใช้การ Post ในการส่งข้อมูลที่เป็น Json โดยที่ข้อมูลในการส่งจะเป็น 'Content-Type'

บรรทัดที่ 224 ถ้า response มีค่า statusCode เท่ากับ 200 นั่นแปลว่าการเข้าสู่ระบบผ่าน Facebook ID สำเร็จจะโชว์ข้อความที่ SnackBar ว่า "Login Success" และจะเข้าสู่หน้า Homepage

บรรทัดที่ 228 หากไม่สามารถเรียก service ได้จะโชว์ข้อความที่ SnackBar ว่า "Login Error"

```
37      //Login emailpassword
38      @PostMapping(value = "/loginByemail",
39              consumes = MediaType.APPLICATION_JSON_VALUE,
40              produces = MediaType.APPLICATION_JSON_VALUE)
41      public ResponseEntity<?> getUserBymEmailAndmPassword(@RequestBody LogInDTO logInDTO){
42          System.out.println(logInDTO.getEmail()+" "+logInDTO.getPassword());
43          List<Member> users = loginservice.getUserBymEmailAndmPassword(logInDTO);
44          if(users.size() == 0) {
45              return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
46          }
47          return new ResponseEntity<>(users, HttpStatus.OK);
48      }
49  }
```

ภาพประกอบที่ 3.17 Controller ส่วนเข้าสู่ระบบด้วย Email และ Password

บรรทัดที่ 38 สร้าง Controller โดยใช้ @PostMapping เพื่อรับการร้องขอแบบ Post

บรรทัดที่ 39 ระบุ Consumes เป็น JSON เพื่อระบุว่าข้อมูลที่เข้ามาจะเป็นรูปแบบ JSON

บรรทัดที่ 41 เรียกใช้เมธอด getUserBymEmailAndmPassword จาก service

บรรทัดที่ 43 ประกาศ users มาเก็บค่า DTO ให้กับเมธอด getUserBymEmailAndmPassword

บรรทัดที่ 44-45 ถ้า users เท่ากับ 0 จะส่ง Unauthorized ซึ่งไม่สามารถส่งข้อมูลของผู้ใช้ออกมาได้

บรรทัดที่ 47 ถ้า users ไม่เท่ากับ 0 จะส่งค่า DTO ให้กับเมธอด

```
32      public List<Member> getUserBymEmailAndmPassword(LogInDTO logInDTO) {
33          return userRepository.findByEmailAndPassword(logInDTO.getEmail(), logInDTO.getPassword());
34      }
```

ภาพประกอบที่ 3.18 Service ในการเข้าสู่ระบบด้วย Email และ Password

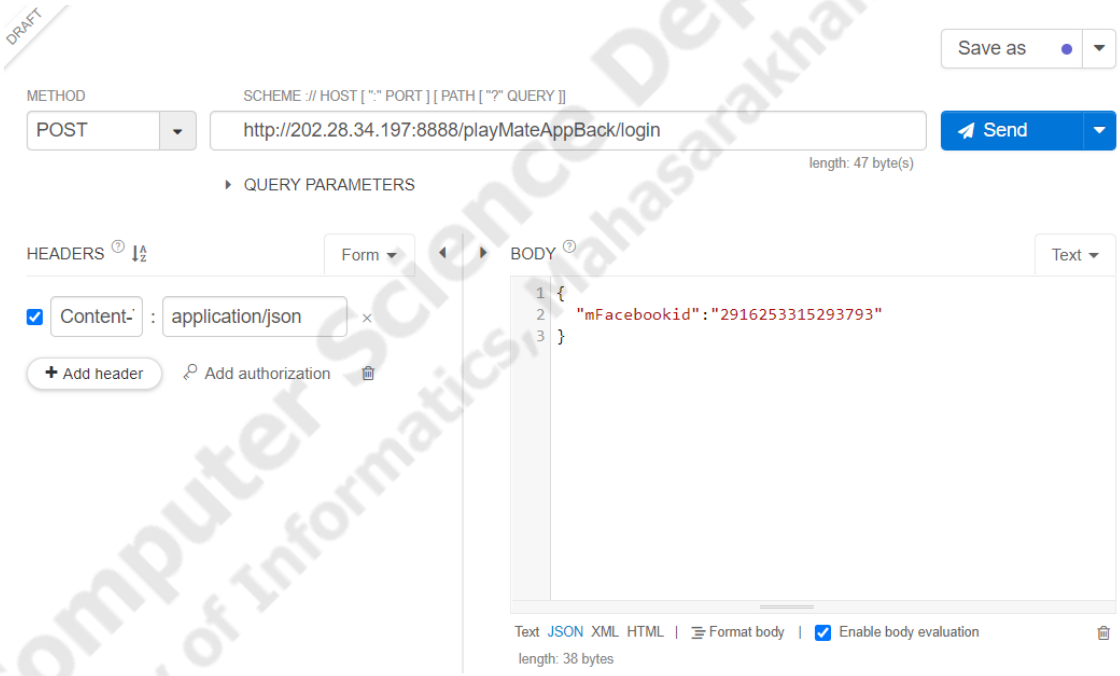
บรรทัดที่ 32 ประกาศเมธอดในการส่งข้อมูล ถูกเรียกจาก getUserBymEmailAndmPassword

บรรทัดที่ 33 ส่งค่าที่ได้จากเมธอด findByEmailAndPassword โดยจะส่งค่า email และ password

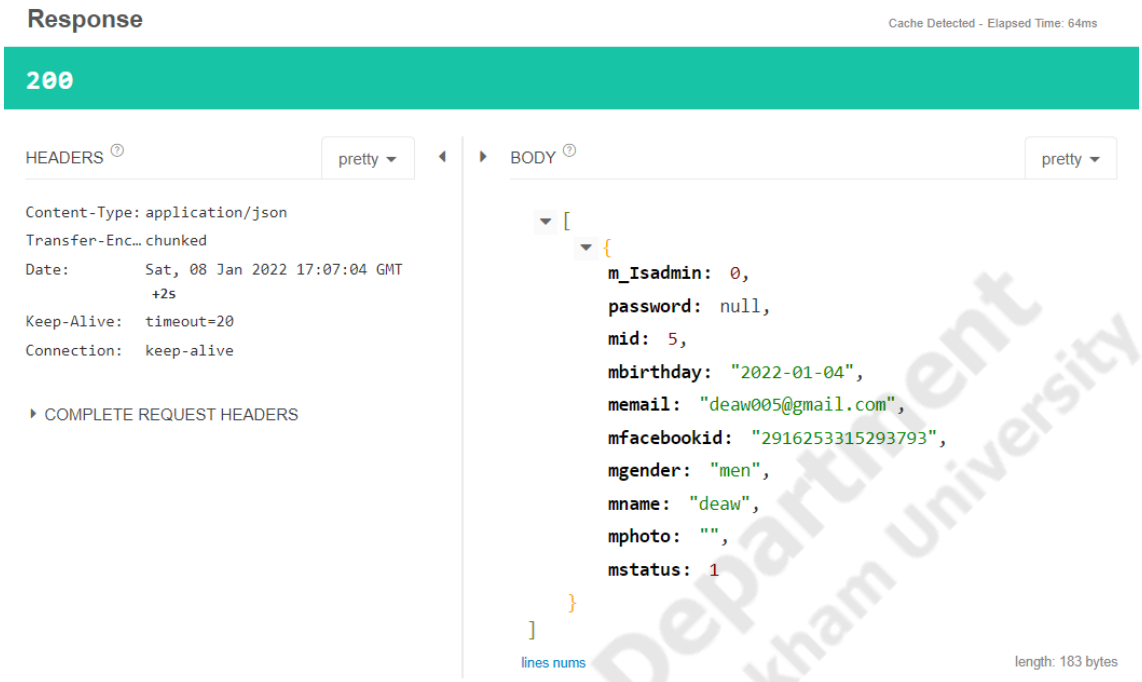
```
11    public List<Member> findByEmailAndPassword(String email, String password);
```

ภาพประกอบที่ 3.19 Repository ในการเข้าสู่ระบบด้วย Email และ Password

บรรทัดที่ 11 ประกาศเมธอดในการส่งข้อมูล Member ถูกเรียกจาก getUserBymEmailAndmPassword ที่ส่งค่า email และ password เป็นค่าของ email และ password ของผู้ใช้



ภาพประกอบที่ 3.20 เรียกใช้ Service ล็อคอินโดยใช้ Email และ Password



ภาพประกอบที่ 3.21 ข้อมูล user ที่ได้จากการเรียกใช้ Service

```
232   Future<void> LoginByEmailAndPassword(String email, String password) async{
233     var json = {"email": email,"password": password};
234     String value = await rootBundle.loadString('assets/config/config.yaml');
235     dynamic conf = loadYaml(value);
236     String url = conf['server']['host'];
237     String port = conf['server']['port'];
238
239     var response = await http.post(Uri.parse("http://$url:$port/playMateAppBack/loginByemail"),
240     headers: <String, String>{
241       'Content-Type': 'application/json; charset=UTF-8',
242     },
243       body: jsonEncode(json));
244
245       log(response.statusCode.toString());
246     if(response.statusCode == 200){
247       ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text("LoginEmail Success")));
248       Navigator.push(context, MaterialPageRoute(builder: (context) => HomePage()));
249     } else {
250       ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text("LoginEmail Error")));
251     }
252   }
253 }
```

ภาพประกอบที่ 3.22 ฟังก์ชันการเข้าสู่ระบบผ่าน Email และ Password

บรรทัดที่ 233 ประกาศตัวแปร json มาเก็บค่า Email และ Password

บรรทัดที่ 234 ประกาศตัวแปร value มาเก็บค่าของข้อมูลใน ‘assets/config/config.yaml’ ซึ่งโหลดมาเก็บเป็นไฟล์ String

บรรทัดที่ 235 ประกาศตัวแปร conf เพื่อดึงค่าจากตัวแปร value และโหลดมาเก็บเป็นไฟล์ yaml

บรรทัดที่ 236 ประกาศตัวแปร url ดึงค่าจาก server host

บรรทัดที่ 237 ประกาศตัวแปร port ดึงค่าจาก server port

บรรทัดที่ 239 ทำการเรียก service โดยใช้การ Post ในการส่งข้อมูลที่เป็น Json โดยที่ข้อมูลในการส่งจะเป็น ‘Content-Type’

บรรทัดที่ 246 ถ้า response มีค่า statusCode เท่ากับ 200 นั่นแปลว่าการเข้าสู่ระบบผ่าน Email และ Password สำเร็จจะโชว์ข้อความที่ SnackBar ว่า "Login Success" และจะเข้าสู่หน้า Homepage

บรรทัดที่ 250 หากไม่สามารถเชื่อมต่อได้จะโชว์ข้อความที่ SnackBar ว่า "Login Error"

3.6.2 Flutter Chat UI

```
119  void _handleImageSelection() async {
120    final result = await ImagePicker().pickImage(
121      imageQuality: 70,
122      maxWidth: 1440,
123      source: ImageSource.gallery,
124    );
125
126    if (result != null) {
127      final bytes = await result.readAsBytes();
128      final image = await decodeImageFromList(bytes);
129
130      final message = types.ImageMessage(
131        author: _user,
132        createdAt: DateTime.now().millisecondsSinceEpoch,
133        height: image.height.toDouble(),
134        id: const Uuid().v4(),
135        name: result.name,
136        size: bytes.length,
137        uri: result.path,
138        width: image.width.toDouble(),
139      ); // types.ImageMessage
140      _addMessage(message);
141    }
142  }
```

ภาพประกอบที่ 3.23 ฟังก์ชันการเลือกโฟลเดอร์หรือแหล่งข้อมูลของไฟล์ที่เป็นรูปภาพ

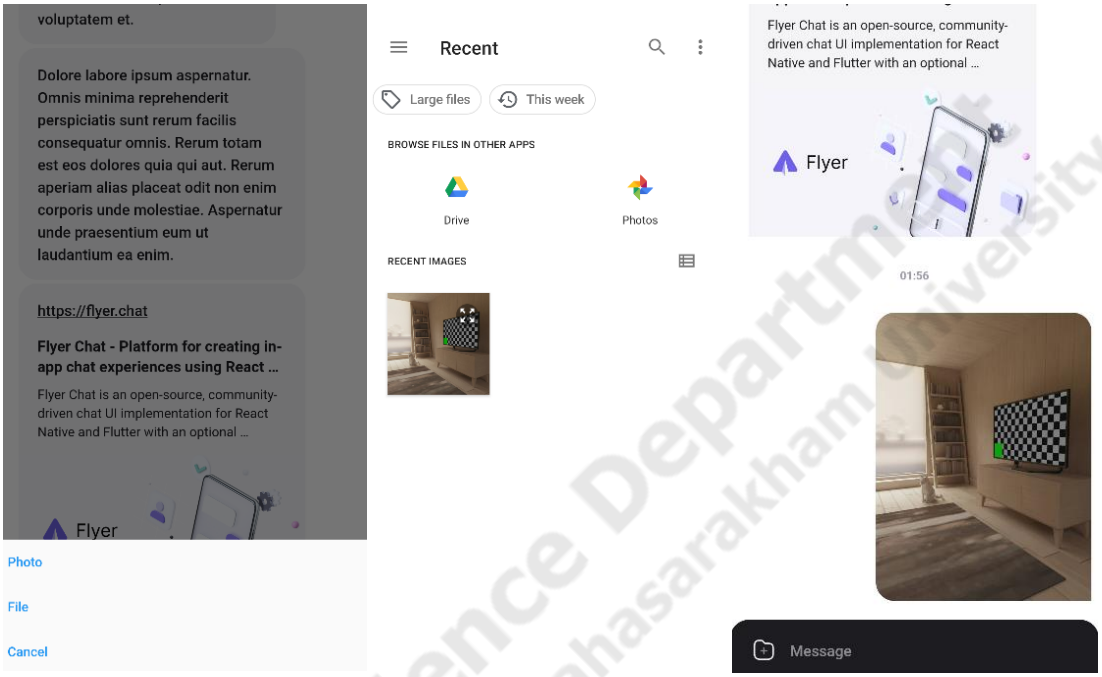
เมื่อกดปุ่ม “Photo” จะทำการเรียกฟังก์ชัน _handleImageSelection() และแสดงโฟลเดอร์หรือที่เก็บข้อมูลไฟล์รูปภาพ หากเลือกไฟล์รูปภาพแล้ว รูปภาพจะถูกนำไปโชว์ที่หน้าจอการสนทนา

บรรทัดที่ 117 สร้างตัวแปรมาใช้สำหรับการเลือกไฟล์รูปภาพและกำหนดคุณสมบัติของไฟล์รูปภาพ

บรรทัดที่ 123 ถ้า result ไม่เท่ากับ Null จะทำการสร้างตัวแปร bytes มาอ่านไฟล์รูปภาพที่เลือกมา และนำไป decode เก็บข้อมูลในตัวแปร image

บรรทัดที่ 127-136 เก็บคุณสมบัติของ image ไว้ใน message โดยที่จะเก็บข้อมูลไอดีคนส่ง สร้างเมื่อเวลา หรือ ข้อมูลไอดีของข้อความที่ไม่ซ้ำกัน

บรรทัดที่ 138 จะทำงานส่งข้อมูลไปที่ฟังก์ชัน _addMessage เพื่อบันทึกข้อมูล



ภาพประกอบที่ 3.24 ตัวอย่างการเลือกโฟลเดอร์หรือแหล่งข้อมูลของไฟล์ที่เป็นรูปภาพ

```
148   void _handlePreviewDataFetched(
149     types.TextMessage message,
150     types.PreviewData previewData,
151   ) {
152     final index = _messages.indexWhere((element) => element.id == message.id);
153     final updatedMessage = _messages[index].copyWith(previewData: previewData);
154
155     WidgetsBinding.instance?.addPostFrameCallback((_) {
156       setState(() {
157         _messages[index] = updatedMessage;
158       });
159     });
160   }
```

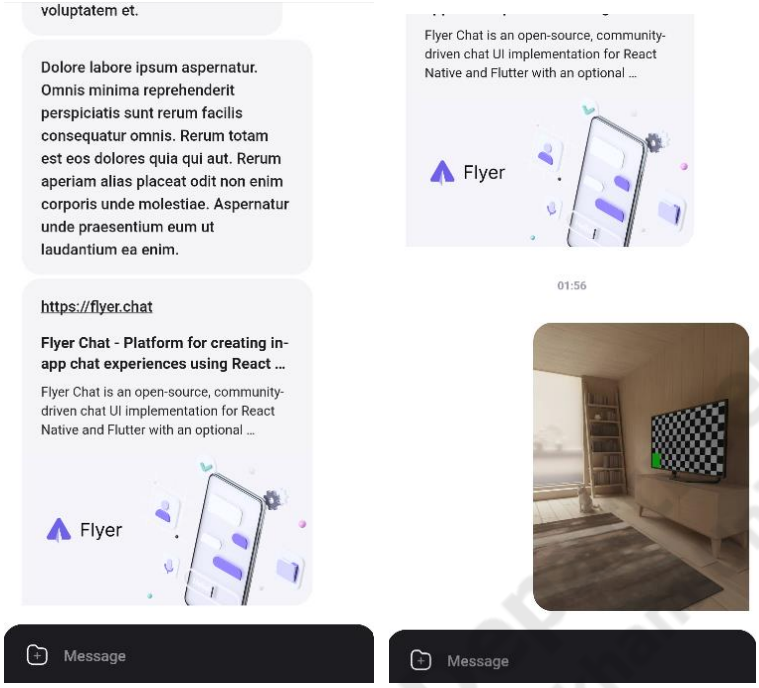
ภาพประกอบที่ 3.25 ฟังก์ชันการอัปเดตข้อมูล

เป็นฟังก์ชันงานที่ถูกใช้เมื่อมีการเพิ่มข้อมูลทั้งไฟล์รูปภาพและการเพิ่มข้อความ

บรรทัดที่ 152 ประกาศตัวแปร index เก็บข้อมูลที่ถูกเพิ่มเข้ามา

บรรทัดที่ 153 ประกาศตัวแปร updateMessage เพื่อทำการอัปเดตข้อมูลที่ถูกเพิ่มเข้ามา

บรรทัดที่ 155-158 ทำการโชว์ข้อมูลที่ถูกเพิ่มเข้ามาใหม่



ภาพประกอบที่ 3.26 ตัวอย่างการอัปเดตข้อมูล

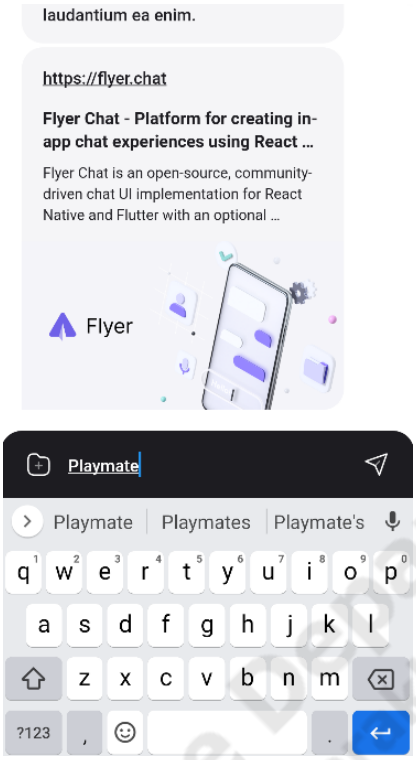
```
165  void _handleSendPressed(types.PartialText message) {
166    final textMessage = types.TextMessage(
167      author: _user,
168      createdAt: DateTime.now().millisecondsSinceEpoch,
169      id: const Uuid().v4(),
170      text: message.text,
171    ); // types.TextMessage
172    _addMessage(textMessage);
173  }
```

ภาพประกอบที่ 3.27 ฟังก์ชันการกดส่งข้อความ

ฟังก์ชัน _handleSendPressed จะทำงานเมื่อกดปุ่มไอค่อนส่งข้อความและเพิ่มข้อมูลไปที่ฟังก์ชัน _addMessage

บรรทัดที่ 163 สร้างตัวแปร textMessage เก็บคุณสมบัติของ TextMessage เช่นข้อมูลไอดีคนส่ง เวลาที่สร้างข้อมูล ไอดีข้อมูลของข้อความที่ถูกส่ง

บรรทัดที่ 172 จะทำงานส่งข้อมูลไปที่ฟังก์ชัน _addMessage เพื่อบันทึกข้อมูล



ภาพประกอบที่ 3.28 ตัวอย่างการกดส่งข้อความ

```
40  @override
41  void initState() {
42    super.initState();
43    _loadMessages();
44  }
```

ภาพประกอบที่ 3.29 ฟังก์ชันการเรียกใช้งาน

```
173 void _loadMessages() async {
174   final response = await rootBundle.loadString('assets/messages.json');
175   final messages = (jsonDecode(response) as List)
176       .map((e) => types.Message.fromJson(e as Map<String, dynamic>))
177       .toList();
178
179   setState(() {
180     _messages = messages;
181   });
182 }
```

ภาพประกอบที่ 3.30 ฟังก์ชันการดึงข้อมูล

ถูกเรียกใช้ในฟังก์ชัน initState() จะทำหน้าที่ดึงข้อมูลจากไฟล์ ‘assets/messages.jon’ มาโชว์

บรรทัดที่ 174 ทำการโหลดไฟล์จาก ‘assete/massages.json’ และเก็บไว้ที่ response

บรรทัดที่ 175-181 แปลง response เป็น Object ชนิด List จากนั้นจะทำการวนลูปแต่ละ Object ให้เป็น toList และ setState ข้อมูลใน messages

```
184   @override
185   Widget build(BuildContext context) {
186     return Scaffold(
187       body: SafeArea(
188         bottom: false,
189         child: Chat(
190           messages: _messages,
191           onAttachmentPressed: _handleAtachmentPressed,
192           onMessageTap: _handleMessageTap,
193           onPreviewDataFetched: _handlePreviewDataFetched,
194           onSendPressed: _handleSendPressed,
195           user: _user,
196         ), // Chat
197       ), // SafeArea
198     ); // Scaffold
199   }
```

ภาพประกอบที่ 3.31 การเรียกใช้งานฟังก์ชันทั้งหมด

บรรทัดที่ 190 เป็นการเรียกใช้ฟังก์ชัน _messages

บรรทัดที่ 191 เป็นการเรียกใช้ฟังก์ชัน _handleAtachmentPressed

บรรทัดที่ 192 เป็นการเรียกใช้ฟังก์ชัน _handleMessageTap

บรรทัดที่ 193 เป็นการเรียกใช้ฟังก์ชัน _handlePreviewDataFetched

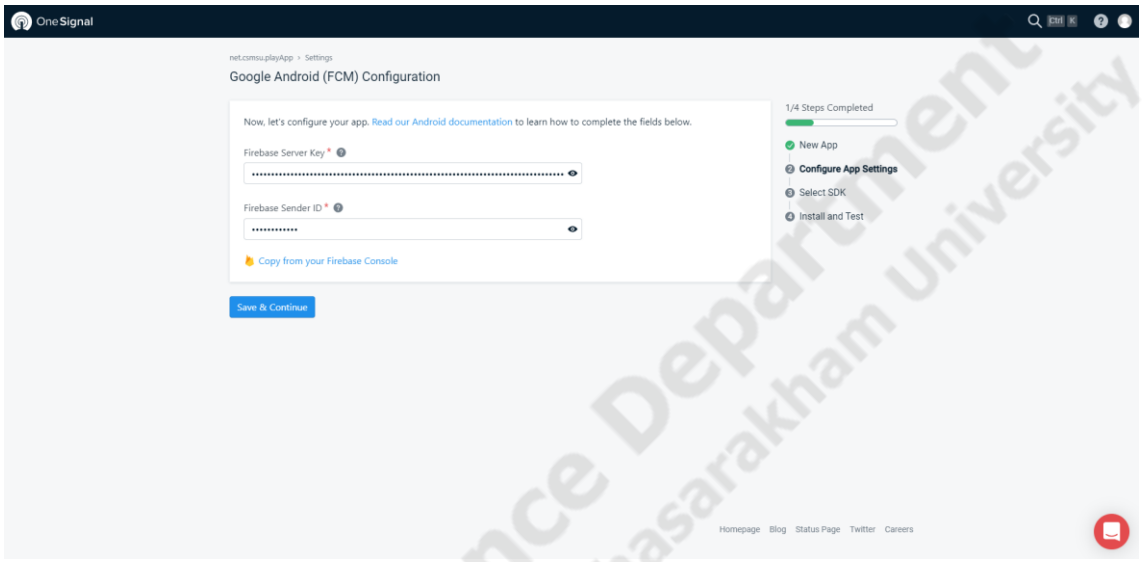
บรรทัดที่ 194 เป็นการเรียกใช้ฟังก์ชัน _handleSendPressed

บรรทัดที่ 195 เป็นการเรียกใช้ฟังก์ชัน _user

3.6.3 Push Notification

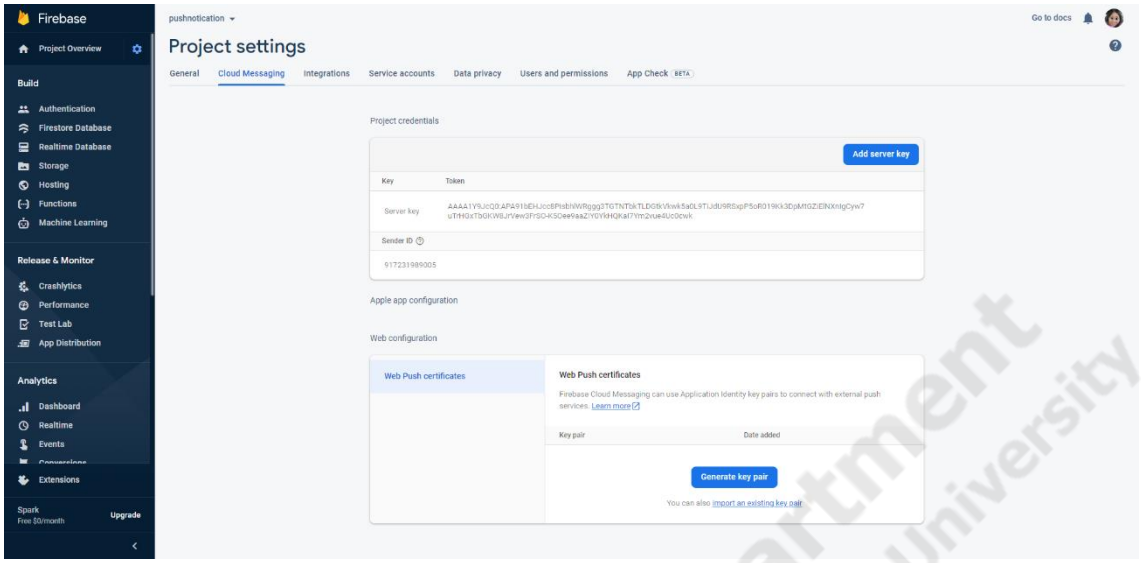
Notification เป็นหนึ่งในช่องทางของแอนดรอยด์ที่เปิดให้แอปสามารถส่งข้อความให้ ผู้ใช้เห็นได้ โดยผู้ใช้ไม่จำเป็นต้องเปิดแอปขึ้นมา และผู้ใช้ก็สามารถสั่งงานบางอย่างผ่าน Notification ตัว นั้นๆ กลับมาได้อีกด้วย นี่คือคำจำกัดความสั้นๆ ของ Notification ซึ่งเป็นหนึ่งในความสามารถของแอนดรอยด์ที่เปิดให้นักพัฒนาสามารถใช้งาน Notification ในสถานการณ์ต่างๆ ที่เหมาะสมกับแอปของตนเองได้ การทำงานจะทำงานอยู่บน Android Framework ซึ่งเป็นหัวใจหลักในการทำงานของ ระบบแอนดรอยด์ที่มีไว้ให้แอปต่างๆเรียกใช้งานซึ่งใน Android Framework ก็จะมีการแบ่งการทำงาน

แยกกันออกไปตามหน้าที่ และส่วนที่ทำหน้าที่เกี่ยวกับระบบ Notification ก็จะมีชื่อเรียกว่า Notification Manager และ Status Bar ของแอนดรอยด์ที่ทำหน้าที่แสดง Notification จากแอปต่างๆก็จะ รับข้อมูลมาจาก Notification Manager ดังนั้นเมื่อใดก็ตามที่แอปในเครื่องส่ง Notification ไปให้ Notification Manager ก็จะถูกส่งไปเพื่อแสดงใน Notification Drawer ที่อยู่ใน Status Bar



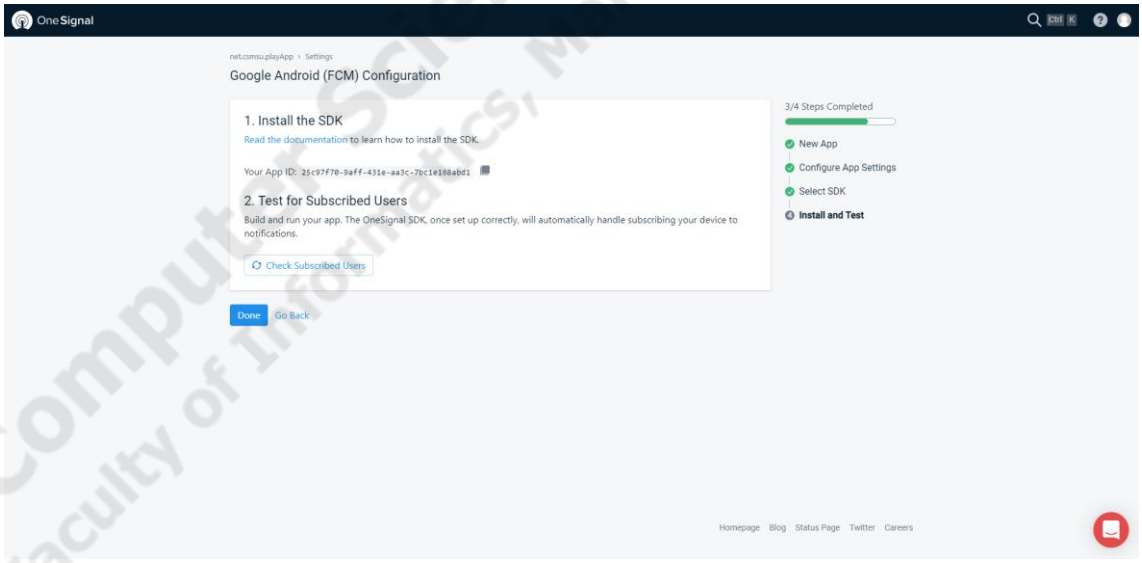
ภาพประกอบที่ 3.32 การล็อคอินผ่าน OneSignal เพื่อสร้างแอปหรือเว็บไซต์

OneSignal คือเครื่องมือที่ช่วยในการทำ Push Notification ได้อย่างง่าย โดยการทำงานของ OneSignal จะทำการสร้าง Key มาให้และมี API ให้เราสามารถสั่งการจาก Service ได้ ซึ่งถ้าเราต้องการจะส่ง Notification ไปหาทุกๆ User ในทุก Platform เราแค่ส่งไปบอก OneSignal แล้ว OneSignal จะทำการกระจาย Notification ให้เราเอง เมื่อเราทำการล็อคอิน OneSignal แล้วจึงทำการสร้างแอปหรือเว็บไซต์ขึ้นมา จากนั้นจะมีการกำหนดค่าแอปโดยเราใช้ Firebase.google



ภาพประกอบที่ 3.33 การล็อคอิน Firebase เพื่อรับ Server Key และ Sender ID

เมื่อเข้ามาที่ Firebase แล้วให้ทำการล็อคอินและสร้างโปรเจ็คให้เรียบร้อย และเมื่อต้องการแก้ไขข้อมูลให้เลือกมาที่ปุ่มเฟือง Settings จากนั้นให้เลือกไปที่ Cloud Messaging และทำการ Copy Server Key และ Sender ID นำไปใส่ที่ OneSignal ช่อง Firebase Server Key และ Firebase Sender ID



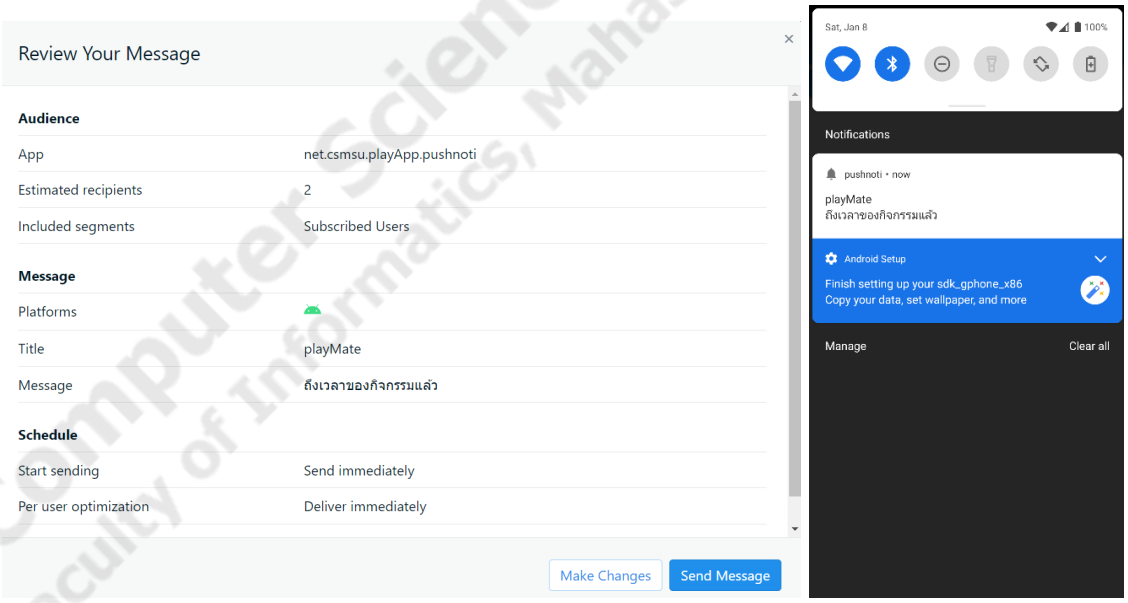
ภาพประกอบที่ 3.34 รับรหัสแอป OneSignal SDK

เมื่อทำการใส่ Firebase Server Key และ Firebase Sender ID เรียบร้อยแล้วเราจะได้รหัสแอปจากนั้นสร้างและเรียกใช้แอป OneSignal SDK เมื่อตั้งค่าอย่างถูกต้องแล้ว จะจัดการการสมัครรับการแจ้งเตือนอุปกรณ์โดยอัตโนมัติ

```
13  Widget build(BuildContext context) {
14    OneSignal.shared.setAppId("3347f005-8a62-452e-a43a-d9a74a931573");
15    return Scaffold(
16      appBar: AppBar(
17        title: Text("push noti"),
18      ), // AppBar
19      body: Container(
20        child: Column(
21          children: [
22            ElevatedButton(
23              child: Text("send noti"),
24              onPressed: () => _handleSendNotification()),  // ElevatedButton
25            ],
26        ), // Column
27      ), // Container
28    ); // Scaffold
29  }
```

ภาพประกอบที่ 3.35 เรียกใช้ OneSignal และกำหนดค่า OneSignal SDK

บรรทัดที่ 14 เรียกใช้ OneSignal และกำหนดค่า OneSignal SDK ที่ได้เป็นการเชื่อมแอปเราไปยัง OneSignal และ OneSignal จะเชื่อมไปยัง Firebase และ Firebase จะเชื่อมไปยังปลายทางอื่นๆ



ภาพประกอบที่ 3.36 ส่งข้อความจาก OneSignal ไปที่ Service และส่งไปยังปลายทาง

จากนั้นทดลองส่งข้อความจาก OneSignal ไปที่ Service จากนั้นข้อมูลจะถูกส่งไปยังปลายทางอื่นๆที่ต้องการและทดสอบรันอีมูเลเตอร์เพื่อดูว่ามีข้อความการแจ้งเตือนถูกส่งมาหรือไม่

```
31    void _handleSendNotification() async {
32      var deviceState = await OneSignal.shared.getDeviceState();
33
34      if (deviceState == null || deviceState.userId == null)
35          return;
36
37      var playerId = deviceState.userId!;
38
39      var imgUrlString =
40          "http://cdn1-www.dogtime.com/assets/uploads/gallery/30-impossibly-cute-puppies/impossibly-cute-puppy-2.jpg";
41
42      var notification = OSCreateNotification(
43          playerIds: ['a8b04952-6e31-11ec-b2ba-265afbb2c3c5'],//playerId
44          content: "mix noti",
45          heading: "Test Notification",
46          iosAttachments: {"id1": imgUrlString},
47          bigPicture: imgUrlString,
48          buttons: [
49            OSActionButton(text: "test1", id: "id1"),
50            OSActionButton(text: "test2", id: "id2")
51          ]); // OSCreateNotification
52
53      var response = await OneSignal.shared.postNotification(notification);
54    }
55  }
```

ภาพประกอบที่ 3.37 การสร้าง Object ของ Notification

บรรทัดที่ 32 ดึงค่าจาก OneSignal สถานะของเครื่องที่รันอยู่ปัจจุบัน

บรรทัดที่ 34-35 ถ้าดึงค่า id จาก OneSignal ไม่ได้หรือมีค่าเป็น 0 จะแสดงค่า null

บรรทัดที่ 37 ถ้าดึงค่า id จาก OneSignal ได้จะนำมาเก็บใน player id เป็น id ของผู้รับ และ deviceStatus คือ player id ของเครื่องนี้จะไม่สามารถส่งหาคนอื่นได้

บรรทัดที่ 42-51 เป็นการสร้าง Object ของ Notification และนำ id เครื่องปลายทางที่เราจะส่งใส่ที่ playerIds และข้อความที่จะส่งประกอบไปด้วย content และ heading คือ หัวข้อความและเนื้อข้อความ

บรรทัดที่ 53 เรียกใช้ OneSignal ให้ส่ง Notification แล้ว Notification จะส่ง Object ขึ้นไป Server แล้ว Server จะทำการตรวจเช็คข้อความและเลข id จากนั้นจะส่งไปที่เครื่องปลายทาง เมื่อส่งเสร็จสามารถนำ response ไปเช็คได้ว่าส่งไปที่เครื่องปลายทางกี่คน