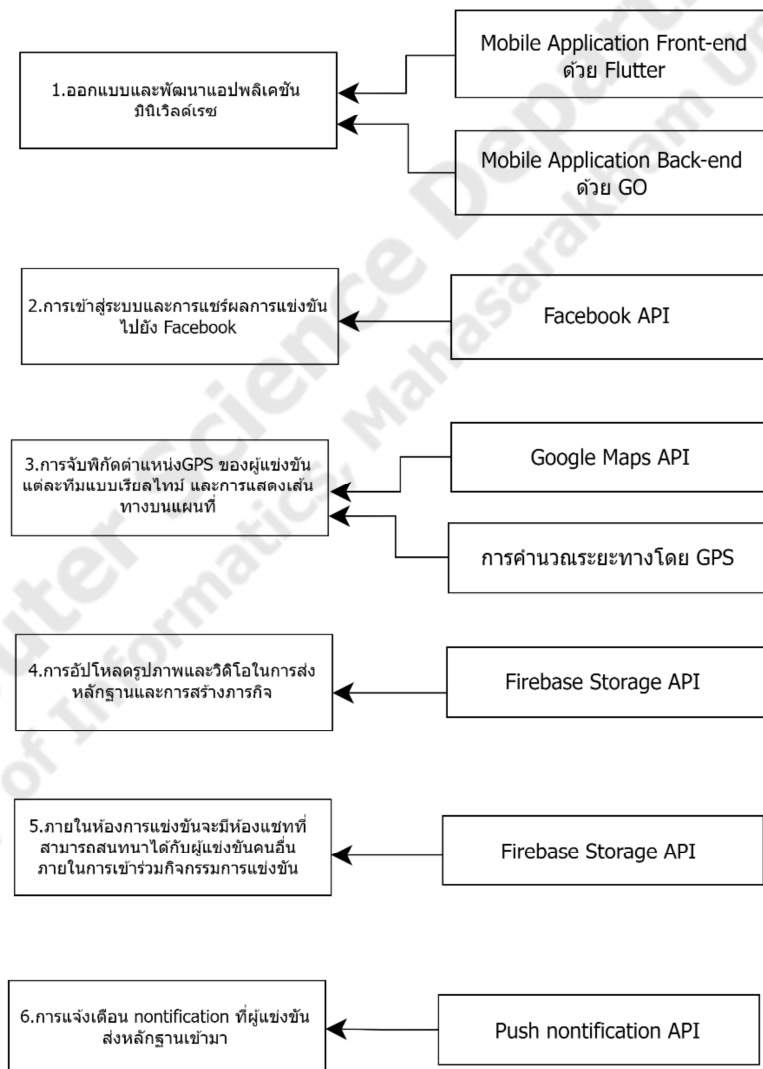


บทที่ 3

ขั้นตอนการดำเนินงาน

สำหรับบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานโครงการปริญญาโท ซึ่งจะทำให้ทราบถึงการวิเคราะห์และกระบวนการประมวลผลโดยละเอียดว่ามีแนวทางในการทำงานหรือขั้นตอนในการทำงานของระบบเป็นอย่างไร โดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้

3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบคำอธิบาย

1. เขียนโปรแกรมเพื่อสร้างแอปพลิเคชันมินิเวิร์ลด์เรซ สำหรับให้ผู้ใช้ใช้งานแอปพลิเคชันผ่านสมาร์ตโฟน โดยการสร้างแอปพลิเคชันจะต้องมีการติดตั้งเครื่องมือดังนี้

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

- Go หรือ Golang เป็นภาษาที่เหมาะสมสำหรับการทำ Web Development เพราะ Golang สามารถใช้สร้างระบบที่รองรับการทำงานใน Scale ใหญ่ที่มี Request จำนวนมากได้ ซึ่ง Golang ถูกออกแบบมาเพื่องานประเภทนี้โดยเฉพาะ การใช้งานใช้ ติดต่อกับฐานข้อมูลฝั่ง Back end ถูกพัฒนาขึ้นโดยบริษัท Google

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

3. พิกัดสถานที่ทำกิจกรรม Google Map API Google Maps Platform หรือ GMP เป็น API ของ Google ที่ให้นักพัฒนา โปรแกรม หรือ Developer สามารถเรียกไปใช้งานเพื่อสร้าง Application ขององค์กรตนเอง โดย Based on Google Maps ซึ่งรองรับการทำงานทั้ง Web Application และ Mobile Application ผ่านระบบปฏิบัติการ iOS และ Android

4. ผู้แข่งขันสามารถอัปโหลดรูปภาพและวิดีโอในการสร้างภารกิจและการส่งหลักฐานภารกิจ

- Firebase คืออะไร เป็นหนึ่งในผลิตภัณฑ์ของ Google โดย Firebase คือ Platform ที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วนของ Backend หรือ Server side ซึ่งทำให้สามารถ Build Mobile Application ได้อย่างมีประสิทธิภาพ และยังลดเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูลให้อีกด้วย

5. ภายในห้องการแข่งขันจะมีห้องแชทที่สามารถสนทนาได้กับผู้แข่งขันคนอื่นภายในการแข่งขัน

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

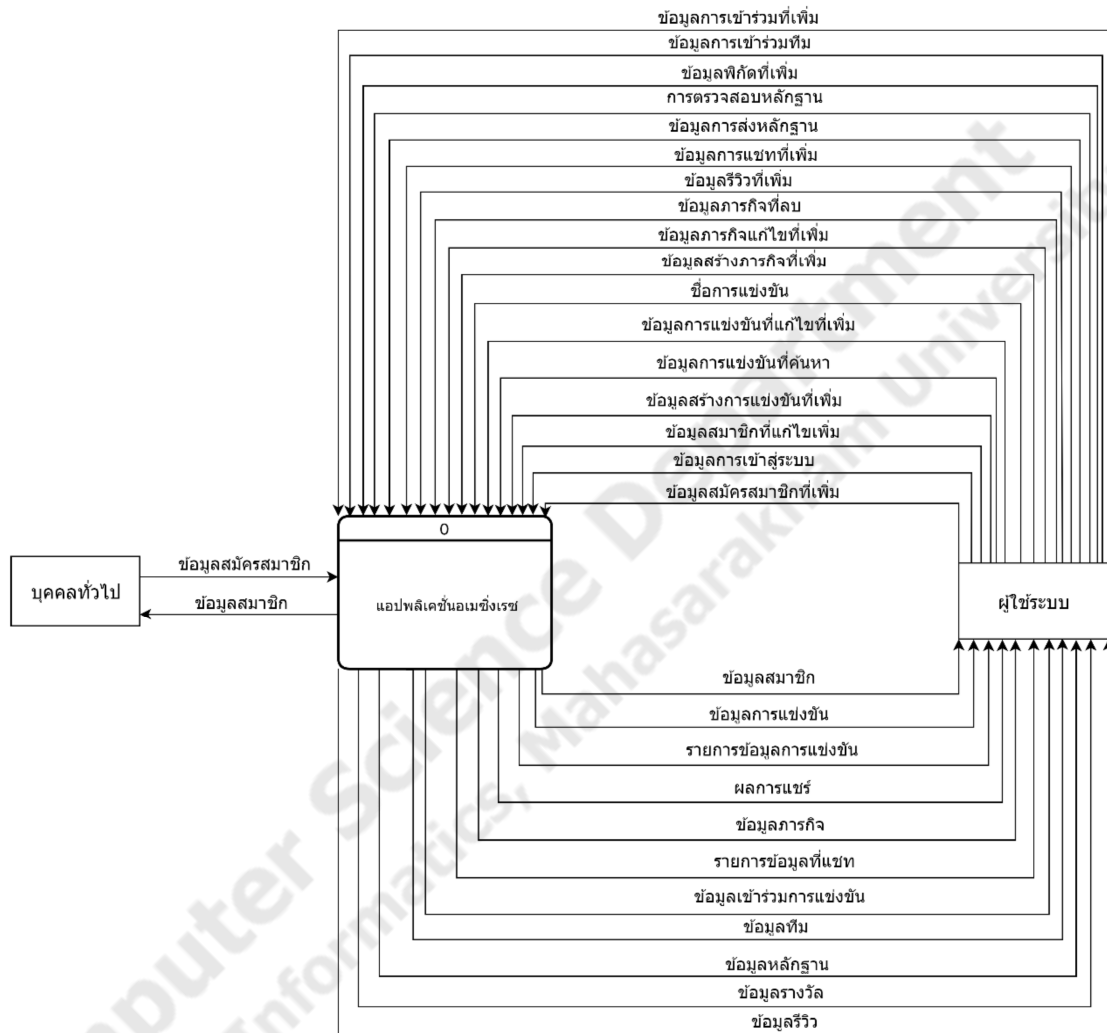
6. การแจ้งเตือน Notification ที่ผู้แข่งขันส่งหลักฐานเข้ามา

Push Notification หรือ การแจ้งเตือน คือ การที่แอปพลิเคชันนำข้อมูลมาแสดงในแถบแจ้งเตือนของระบบปฏิบัติการนั้นๆ กำหนด ไม่ว่าจะเป็น Mobile (iOS, Android) หรือ Computer ทั่วไป

ซึ่ง modern browser ในปัจจุบันก็สามารถแสดงแถบแจ้งเตือนได้แล้ว

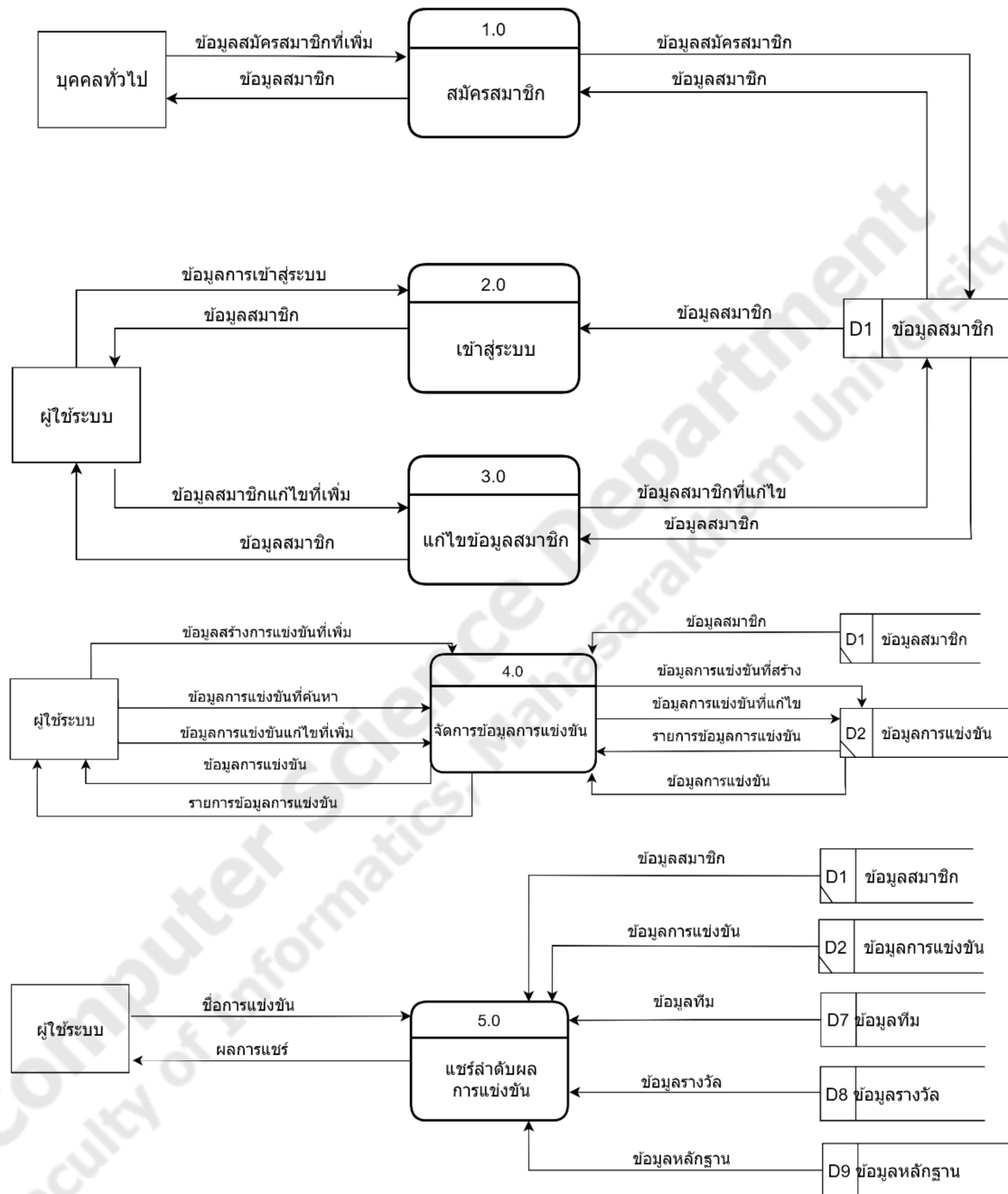
3.2 การออกแบบระบบ

3.2.1 แผนภาพบริบท (Context Diagram)

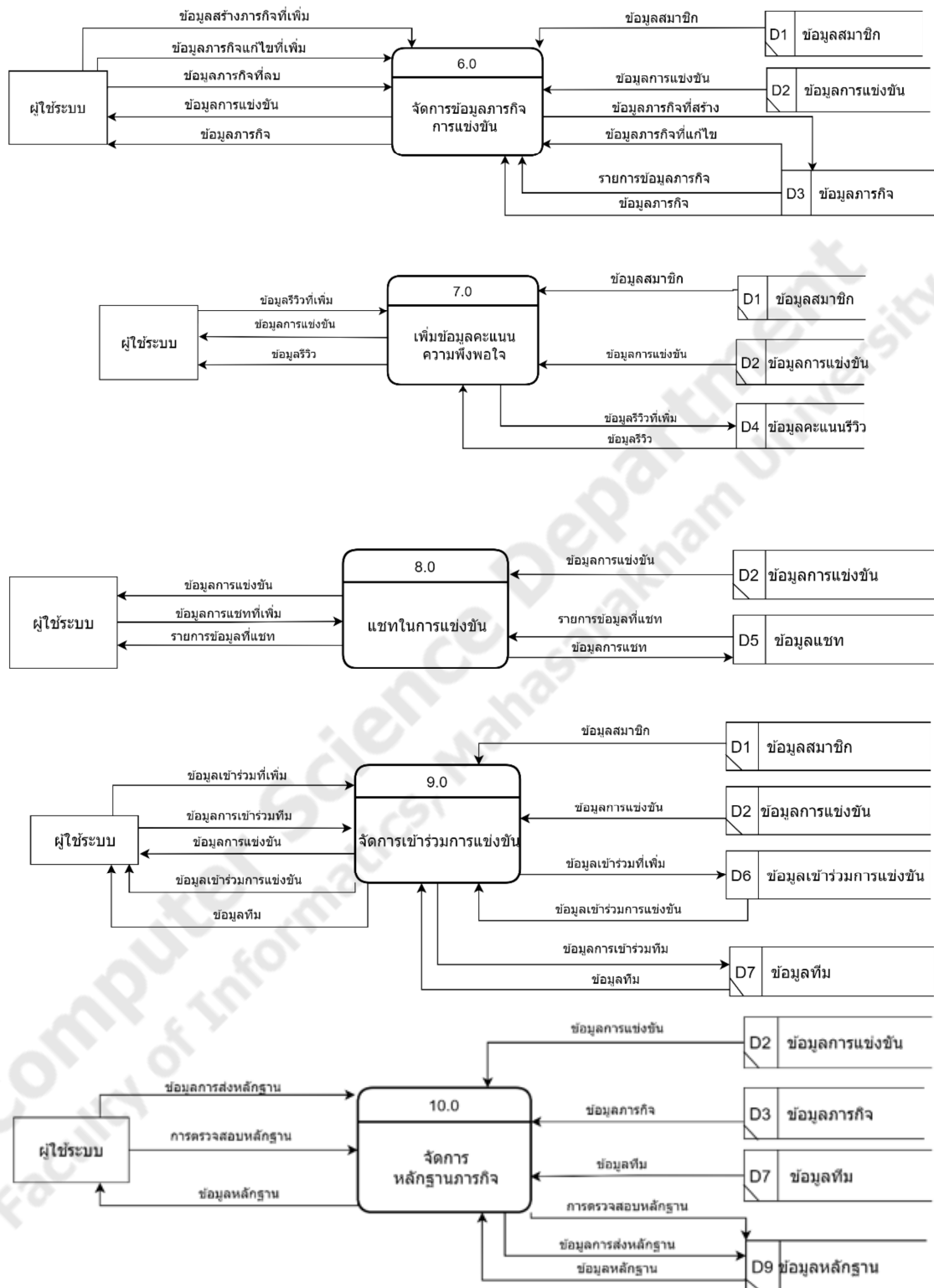


ภาพประกอบที่ 3.2 แผนภาพบริบท (Context Diagram)

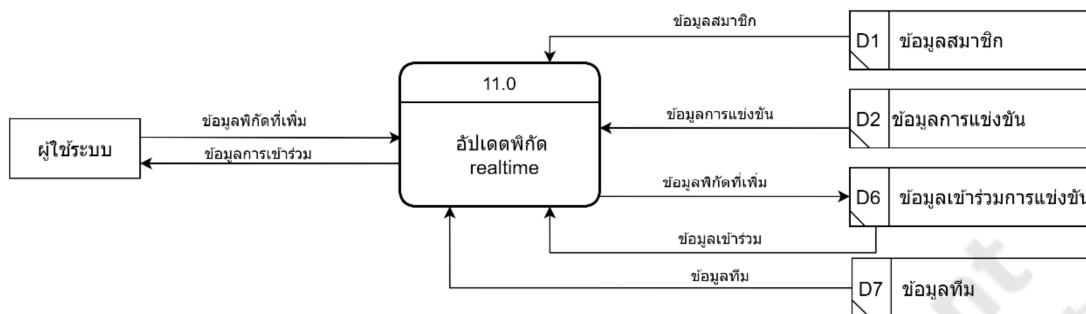
3.2.2 Data Flow Diagram Level 1



ภาพประกอบที่ 3.3 Data Flow Diagram Level 1



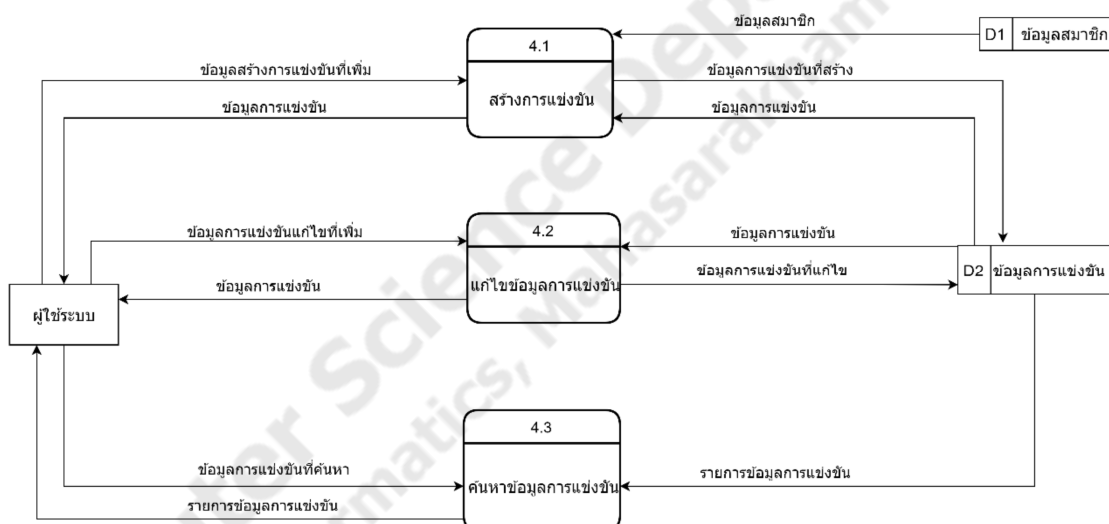
ภาพประกอบที่ 3.3 Data Flow Diagram Level 1 (ต่อ)



ภาพประกอบที่ 3.3 Data Flow Diagram Level 1 (ต่อ)

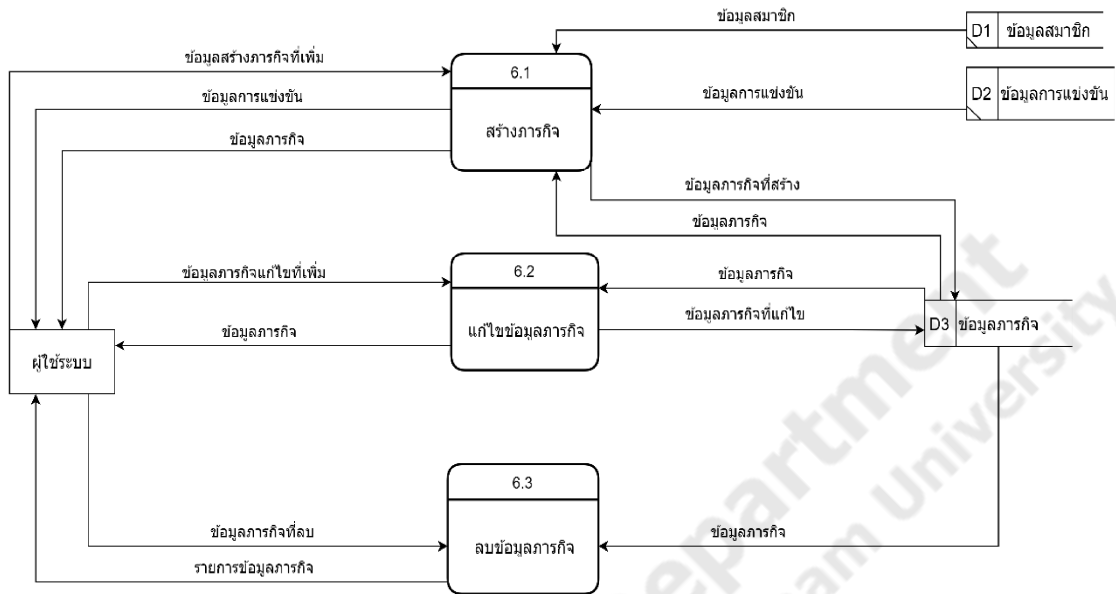
3.2.3 Data Flow Diagram Level 2

3.2.3.1 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 4



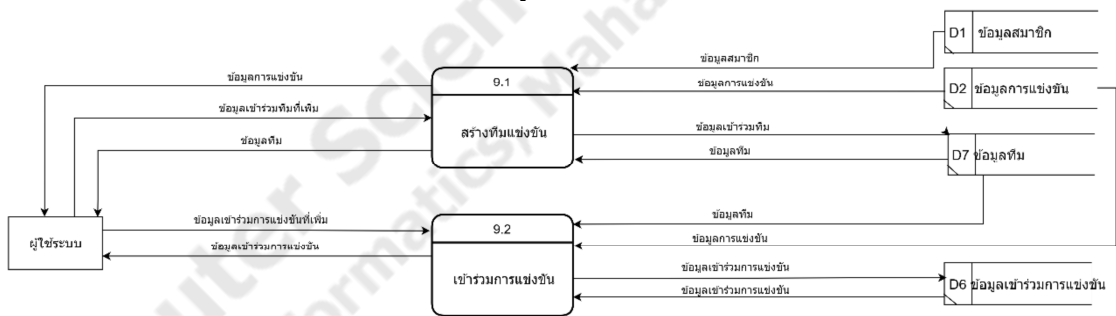
ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 Process 4

3.2.3.2 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 6



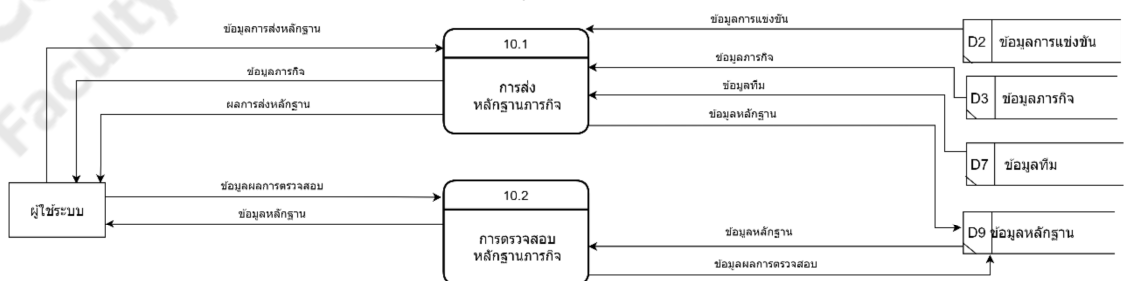
ภาพประกอบที่ 3.5 Data Flow Diagram Level 2 Process 6

3.2.3.3 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 9



ภาพประกอบที่ 3.6 Data Flow Diagram Level 2 Process 9

3.2.3.4 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 10



ภาพประกอบที่ 3.7 Data Flow Diagram Level 2 Process 10

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
บุคคลทั่วไป	ผู้ใช้งานประเภท ทัวไปที่จะต้อง ทำ การสมัครสมาชิกใน การใช้ งานแอปพลิเคชันมินิเว็ลด์เรช	- ข้อมูลสมาชิก	- ข้อมูลสมัครสมาชิก
ผู้ใช้ระบบ	ผู้ใช้งานประเภทที่ เป็นสมาชิก ในการ ใช้งานแอปพลิเคชัน มินิ เว็ลด์เรช	- ข้อมูลสมาชิก - ข้อมูลการแข่งขัน - รายการข้อมูลการแข่งขัน - ผลการแข่ง - ข้อมูลภารกิจ - รายการข้อมูลที่ แชนท์ - รายการเข้าร่วมการแข่งขัน - ข้อมูลทีม - ข้อมูลหลักฐาน - ข้อมูลรางวัล	- ข้อมูลสมัครสมาชิกที่ เพิ่ม - ข้อมูลการเข้าสู่ระบบ - ข้อมูลสมาชิกที่แก้ไขเพิ่ม - ข้อมูลสร้างการแข่งขันที่ เพิ่ม - ข้อมูลการแข่งขันที่ค้นหา - ข้อมูลการแข่งขันที่แก้ไข ที่เพิ่ม - ชื่อการแข่งขัน - ข้อมูลภารกิจที่สร้างที่ เพิ่ม - ข้อมูลภารกิจที่แก้ไข - ข้อมูลภารกิจที่ลบ - ข้อมูลรีวิวที่เพิ่ม - ข้อมูลการแชทที่เพิ่ม - ข้อมูลการแชท - ข้อมูลการส่งหลักฐาน - การตรวจสอบหลักฐาน - ข้อมูลพิกัดที่เพิ่ม - ข้อมูลการเข้าร่วมทีม - ข้อมูลการเข้าร่วมทีมที่ เพิ่ม

3.3.2 Data Flow and Data Structure Description

เป็นขั้นตอนการทำงานของระบบซึ่งทำให้เราทราบถึงการรับ-ส่งข้อมูลแสดงถึงการไหล

ของ ข้อมูลทั้ง ข้อมูลเข้า (Input) และข้อมูลส่งออก (Output) ระหว่างข้อมูลต้นทางถึงข้อมูลปลายทาง โดย อธิบายข้อมูลและขั้นตอนในการดำเนินงานดังต่อไปนี้

ตารางที่ 3.2 Data Flow and Data Structure Description

Name	Description	Source	Destination	Data Structure
ข้อมูลสมัครสมาชิกที่เพิ่ม	ข้อมูลของผู้ใช้ทั่วไป ที่สมัครเพื่อใช้งานแอปพลิเคชัน	ผู้ใช้ทั่วไป	Process 1.0 สมัครสมาชิก	[อีเมล+รหัสผ่าน+ชื่อ+นามสกุล+ชื่อในระบบ+รูปโปรไฟล์+(คำอธิบายตัวเอง) บัญชีเฟซบุ๊ก]
ข้อมูลสมัครสมาชิก	ข้อมูลของผู้ใช้ทั่วไป ที่สมัครเพื่อใช้งานแอปพลิเคชัน	Process 1.0 สมัครสมาชิก	เพิ่มข้อมูลสมาชิก	อีเมล+รหัสผ่าน+ชื่อ+นามสกุล+ชื่อในระบบ+รูปโปรไฟล์+(คำอธิบายตัวเอง) บัญชีเฟซบุ๊ก+รหัสสมาชิก
ข้อมูลสมาชิก	ข้อมูลของสมาชิก	เพิ่มข้อมูลสมาชิก	Process 1.0 สมัครสมาชิก	อีเมล+รหัสผ่าน+ชื่อ+นามสกุล+ชื่อในระบบ+รูปโปรไฟล์+(คำอธิบายตัวเอง)
		Process 1.0 สมัครสมาชิก	ผู้ใช้ทั่วไป	
		เพิ่มข้อมูลสมาชิก	Process 2.0 เข้าสู่ระบบ	
		Process 2.0 เข้าสู่ระบบ	ผู้ใช้งานระบบ	
		เพิ่มข้อมูลสมาชิก	Process 3.0 แก้ไขข้อมูลสมาชิก	
		Process 3.0 แก้ไขข้อมูลสมาชิก	เพิ่มข้อมูลสมาชิก	
ข้อมูลสมาชิกแก้ไขที่เพิ่ม	ข้อมูลที่สมาชิกทำการแก้ไข	ผู้ใช้งานระบบ	Process 3.0 แก้ไขข้อมูลสมาชิก	อีเมล+ชื่อ-นามสกุล+ชื่อในระบบ+รูปโปรไฟล์ (คำอธิบาย)

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลสมาชิกที่แก้ไข	ข้อมูลที่ทำกรแก้ไข	Process 3.0 แก้ไขข้อมูลสมาชิก	ผู้ใช้ระบบ	อีเมล+ชื่อ-นามสกุล+ชื่อในระบบ+รูปโปรไฟล์+(คำอธิบายตัวเอง)+รหัสสมาชิก
ข้อมูลสร้างการแข่งขันที่เพิ่ม	ข้อมูลที่ใช้สร้างการแข่งขัน	ผู้ใช้ระบบ	Process 4.1 สร้างการแข่งขัน	ชื่อการแข่งขัน+รายละเอียดสถานที่จัดกิจกรรม+จำนวนทีมที่เข้าร่วม+วันที่เปิดรับ+วันที่ปิดรับสมัคร+วันที่จัดกิจกรรม+เวลาเริ่มกิจกรรม+เวลาสิ้นสุดกิจกรรม
ข้อมูลการแข่งขันที่สร้าง	ข้อมูลที่ใช้สร้างการแข่งขัน	Process 4.1 สร้างการแข่งขัน	เพิ่มข้อมูลการแข่งขัน	ชื่อการแข่งขัน+รายละเอียดสถานที่จัดกิจกรรม+ จำนวนทีมที่เข้าร่วม+วันที่เปิดรับ+วันที่ปิดรับสมัคร+วันที่จัดกิจกรรม+เวลาเริ่มกิจกรรม+เวลาสิ้นสุดกิจกรรม+รหัสสมาชิก+รหัสการแข่งขัน
ข้อมูลการแข่งขันที่ค้นหา	ข้อมูลการแข่งขันที่ค้นหา	ผู้ใช้ระบบ	Process 4.1 สร้างการแข่งขัน	[หมายเลขการแข่งขัน ชื่อการแข่งขัน]
ข้อมูลการแข่งขันแก้ไขที่เพิ่ม	ข้อมูลการแข่งขันแก้ไข	ผู้ใช้ระบบ	Process 4.2 แก้ไขข้อมูลการแข่งขัน	ชื่อการแข่งขัน+รายละเอียดสถานที่จัดกิจกรรม+ จำนวนทีมที่เข้าร่วม+วันที่เปิดรับ+วันที่ปิดรับสมัคร+วันที่จัดกิจกรรม+เวลาเริ่มกิจกรรม+เวลาสิ้นสุดกิจกรรม
ข้อมูลการแข่งขันแก้ไข	ข้อมูลการแข่งขันแก้ไข	Process 4.2 แก้ไขข้อมูลการแข่งขัน	เพิ่มข้อมูลการแข่งขัน	ชื่อการแข่งขัน+รายละเอียดสถานที่จัดกิจกรรม+ จำนวนทีมที่เข้าร่วม+วันที่เปิดรับ+วันที่ปิดรับสมัคร+

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
				วันที่จัดกิจกรรม+เวลาเริ่ม กิจกรรม+เวลาสิ้นสุด กิจกรรม+รหัสสมาชิก+รหัส การแข่งขัน
รายการ ข้อมูลการ แข่งขัน	รายการข้อมูลของ การแข่งขัน	เพิ่มข้อมูลการ แข่งขัน	Process 4.3 ค้นหาข้อมูล การแข่งขัน	{ชื่อการแข่งขัน+จำนวนทีม ที่เข้าร่วม+วันที่จัด กิจกรรม+กิจกรรม+เวลา สิ้นสุดกิจกรรม}
		Process 4.3 ค้นหาข้อมูล การแข่งขัน		
ข้อมูลการ แข่งขัน	ข้อมูลของการ แข่งขัน	เพิ่มข้อมูล การแข่งขัน	Process 4.1 สร้างการแข่งขัน	ชื่อการแข่งขัน+รายละเอียด สถานที่จัดกิจกรรม+จำนวน ทีมที่เข้าร่วม+วันที่เปิดรับ+ วันที่ปิดรับสมัคร+วันที่จัด กิจกรรม+เวลาเริ่ม กิจกรรม+เวลาสิ้นสุด กิจกรรม
		Process 4.1 สร้างการ แข่งขัน	ผู้ใช้ระบบ	
		เพิ่มข้อมูล การแข่งขัน	Process 4.2 แก้ไขข้อมูล การแข่งขัน	
		Process 4.2 แก้ไขข้อมูล การแข่งขัน	ผู้ใช้ระบบ	
		เพิ่มข้อมูล การแข่งขัน	Process 5.0 แชร ลำดับผลการ แข่งขัน	
		เพิ่มข้อมูล การแข่งขัน	Process 6.1 สร้างภารกิจ	
		Process 6.1 สร้างภารกิจ	ผู้ใช้ระบบ	

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		เพิ่มข้อมูล การแข่งขัน	Process 6.1 เซทในการ แข่งขัน	
		Process 6.1 เซทในการ แข่งขัน	ผู้ใช้งานระบบ	
		เพิ่มข้อมูล การแข่งขัน	Process 7.0 เพิ่มข้อมูล คะแนนความ พึงพอใจ	
		Process 7.0 เพิ่มข้อมูลคะแนน ความพึงพอใจ	ผู้ใช้งานระบบ	
		เพิ่มข้อมูลการ แข่งขัน	Process 9.0 การเข้าร่วม แข่งขัน	
		Process 9.0 การเข้าร่วมการ แข่งขัน	ผู้ใช้งานระบบ	
		เพิ่มข้อมูล การแข่งขัน	Process 10.1 การส่งหลักฐาน ภารกิจ	
		เพิ่มข้อมูล การแข่งขัน	Process 11.0 การเพิ่มพิกัด realtime	
ชื่อการแข่งขัน	ข้อมูลผลการ แข่งขันที่แชร์	ผู้ใช้งานระบบ	Process 5.0 แชร์ลำดับผลการ แข่งขัน	ชื่อการแข่งขัน
ผลการแชร์	ผลการแข่งขัน ที่แชร์	Process 5.0 แชร์ลำดับผลการ แข่งขัน	ผู้ใช้งานระบบ	ผลการแชร์
ข้อมูลทีม	ข้อมูลของทีม	เพิ่มข้อมูลทีม	Process 5.0	ชื่อทีม+รหัสสมาชิก+ชื่อ

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
			แชร์ลำดับผลการแข่งขัน	สมาชิก+รหัสการแข่งขัน
		เพิ่มข้อมูลทีม	Process9.0 การเข้าร่วมการแข่งขัน	
		Process9.0 การเข้าร่วมการแข่งขัน	ผู้ใช้ระบบ	
		เพิ่มข้อมูลทีม	Process10.1 การส่งหลักฐานภารกิจ	
		เพิ่มข้อมูลทีม	Process 11.0 การเพิ่มพิกัด Realtime	
ข้อมูลรางวัล	ข้อมูลของรางวัล	เพิ่มข้อมูลรางวัล	Process 5.0 แชร์ลำดับผลการแข่งขัน	รหัสรางวัล+ชื่อรางวัล
ข้อมูลหลักฐาน	ข้อมูลของหลักฐาน	เพิ่มข้อมูลหลักฐาน	Process 5.0 แชร์ลำดับผลการแข่งขัน	รหัสหลักฐาน+ชื่อหลักฐาน+สถานะหลักฐาน+ละติจูด+ลองจิจูด+เวลาในการส่ง+ [รูปภาพ วิดีโอ ข้อความ]+ รหัสภารกิจ+รหัสทีม+รหัสสมาชิก+รหัสการแข่งขัน
		เพิ่มข้อมูลหลักฐาน	Process 10.2 การตรวจสอบหลักฐานภารกิจ	
		Process 10.2 การตรวจสอบหลักฐานภารกิจ	ผู้ใช้ระบบ	

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลสร้าง ภารกิจ ที่เพิ่ม	ข้อมูลสร้างภารกิจ	ผู้ใช้ระบบ	Process 6.1 สร้างภารกิจ	ชื่อภารกิจ+ลำดับของภารกิจ+ รายละเอียดภารกิจ+ คำอธิบาย+[ภาพ วิดีโอ ประกอบภารกิจ]+ประเภท ภารกิจ+ระยะของการเจอคำ ใบ้
ข้อมูล ภารกิจ ที่สร้าง	ข้อมูลสร้างภารกิจ	Process 6.1 สร้างภารกิจ	แฟ้มข้อมูลภารกิจ	ชื่อภารกิจ+ลำดับของภารกิจ+ รายละเอียดภารกิจ+ คำอธิบาย+[ภาพ วิดีโอ ประกอบภารกิจ]+ประเภท ภารกิจ+ระยะของการเจอคำ ใบ้ของภารกิจ+รหัสสมาชิก+ รหัสการแข่งขัน+รหัสภารกิจ
ข้อมูล ภารกิจ	ข้อมูลของภารกิจ	แฟ้มข้อมูล ภารกิจ	Process 6.1 สร้างภารกิจ	ชื่อภารกิจ+ลำดับของภารกิจ+ รายละเอียดภารกิจ+ คำอธิบาย+[ภาพภารกิจ วิดีโอ ภารกิจ]+ประเภทภารกิจ+ ระยะคำใบ้ภารกิจ
		Process 6.1 สร้างภารกิจ	ผู้ใช้ระบบ	
		แฟ้มข้อมูล ภารกิจ	Process 6.2 แก้ไขข้อมูล ภารกิจ	
		แฟ้มข้อมูล ภารกิจ	Process 6.3 ลบข้อมูลภารกิจ	
		แฟ้มข้อมูล ภารกิจ	Process 10.1 การส่งหลักฐาน ภารกิจ	
		Process 10.1 การส่ง หลักฐาน ภารกิจ	ผู้ใช้ระบบ	

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลภารกิจที่แก้ไข	ข้อมูลภารกิจที่แก้ไข	ผู้ใช้ระบบ	Process 6.2 แก้ไขข้อมูลภารกิจ	ชื่อภารกิจ+ลำดับของภารกิจ+รายละเอียดภารกิจ+คำอธิบาย+[ภาพภารกิจ วิดีโอภารกิจ]+ประเภทภารกิจ+ระยะค่าใช้จ่าย
ข้อมูลรีวิว	ข้อมูลรีวิวหลังจบการแข่งขัน	เพิ่มข้อมูลคะแนนรีวิว	Process 7.0 เพิ่มข้อมูลคะแนนความพึงพอใจ	ข้อความรีวิว+คะแนนรีวิว+วันที่เวลา+รหัสสมาชิก+รหัสการแข่งขัน
		Process 7.0 เพิ่มข้อมูลความพึงพอใจ	ผู้ใช้ระบบ	
		Process 6.2 แก้ไขข้อมูลภารกิจ	เพิ่มข้อมูลภารกิจ	
ข้อมูลภารกิจที่ลบ	ข้อมูลภารกิจที่ลบ	ผู้ใช้ระบบ	Process 6.3 ลบข้อมูลภารกิจ	[รหัสภารกิจ ชื่อภารกิจ]
รายการข้อมูลภารกิจ	รายการข้อมูลภารกิจ	ผู้ใช้ระบบ	Process 6.3 ลบข้อมูลภารกิจ	{ชื่อภารกิจ+ลำดับของภารกิจ+รายละเอียดภารกิจ+คำอธิบาย+[ภาพภารกิจ วิดีโอภารกิจ]+ประเภทภารกิจ+ระยะค่าใช้จ่ายภารกิจ}
ข้อมูลรีวิวที่เพิ่ม	ข้อมูลรีวิวที่เพิ่ม	ผู้ใช้ระบบ	Process 7.0 เพิ่มข้อมูลคะแนนความพึงพอใจ	ข้อความรีวิว+คะแนนรีวิว
		Process 7.0 เพิ่มข้อมูลคะแนน	เพิ่มข้อมูลคะแนนรีวิว	

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลการ แชทที่เพิ่ม	ข้อมูลแชทใน หน้าล๊อบบี้ ของการแข่งขัน	ผู้ใช้ระบบ	Process 8.0 แชทในการแข่งขัน	ชื่อสมาชิก+ข้อความ+เวลา ที่ส่ง
ข้อมูลการ แชท	ข้อมูลแชทใน หน้าล๊อบบี้ ของการแข่งขัน	Process 8.0 แชทในการ แข่งขัน	เพิ่มข้อมูลแชท	ชื่อสมาชิก+ข้อความ+เวลา ที่ส่ง+รหัสสมาชิก+รหัส การแข่งขัน
รายการ ข้อมูล ที่แชท	รายการข้อมูล ที่แชท	เพิ่มข้อมูล แชท	Process 8.0 แชทในการแข่งขัน	{ชื่อสมาชิก+ข้อความ+ เวลาที่ส่ง+รหัสสมาชิก+ รหัสการแข่งขัน}
		Process 8.0 แชทในการ แข่งขัน	ผู้ใช้ระบบ(ผู้รับ)	
รายการ ข้อมูล ที่แชทถูกส่ง	รายการข้อมูล ที่แชทถูกส่ง	Process 8.0 แชทในการ แข่งขัน	ผู้ใช้ระบบ(รับ)	{ชื่อสมาชิก+ข้อความ+ เวลาที่ส่ง+รหัสข้อความ+ รหัสสมาชิก+รหัสการ แข่งขัน}
ข้อมูลเข้า ร่วมการ แข่งขัน ที่เพิ่ม	ข้อมูลเข้าร่วม ที่เพิ่มการแข่งขัน	ผู้ใช้ระบบ	Process 9.0 การเข้าร่วม การแข่งขัน	ละติจูด+ลองจิจูด+เวลาที่ เข้าร่วม+รหัสสมาชิก+รหัส ทีม
		Process 9.0 การเข้าร่วม การแข่งขัน	เพิ่มข้อมูลเข้าร่วม การแข่งขัน	
ข้อมูลการ เข้าร่วมทีม	ข้อมูลการเข้าร่วม ทีม	ผู้ใช้ระบบ	Process 9.0 การเข้าร่วม การแข่งขัน	ชื่อทีม+ชื่อสมาชิก+รหัส การแข่งขัน+รหัสสมาชิก
		Process 9.0 การเข้าร่วม การแข่งขัน	เพิ่มข้อมูลทีม	

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลเข้า ร่วมการ แข่งขันที่เพิ่ม	ข้อมูลเข้าร่วม การแข่งขัน	ผู้ใช้ระบบ	Process 9.0 การเข้าร่วม การแข่งขัน	ละติจูด+ลองจิจูด+เวลาที่ เข้าร่วม
ข้อมูลเข้า ร่วมการ แข่งขัน	ข้อมูลเข้าร่วม การแข่งขัน	Process 9.0 การเข้าร่วม การแข่งขัน	เพิ่มข้อมูลเข้าร่วม การแข่งขัน	ละติจูด+ลองจิจูด+เวลาที่ เข้าร่วม+รหัสสมาชิก+รหัส ทีม+รหัสการแข่งขัน
		เพิ่มข้อมูล เข้าร่วมการ แข่งขัน	Process 9.0 การเข้าร่วม การแข่งขัน	
		Process 9.0 การเข้าร่วม การแข่งขัน	ผู้ใช้ระบบ	
ข้อมูลการส่ง หลักฐาน	ข้อมูลของ การส่งหลักฐาน	ผู้ใช้ระบบ	Process 10.1 การส่งหลักฐาน ภารกิจ	ชื่อหลักฐาน+สถานะ หลักฐาน+ละติจูด+ ลองจิจูด+เวลาในการส่ง+ [รูปภาพ วิดีโอ ข้อความ]
ผลการส่ง หลักฐาน	ผลการส่ง หลักฐาน	Process 10.1 การส่ง หลักฐาน ภารกิจ	ผู้ใช้ระบบ	ผลการส่งหลักฐาน
ข้อมูลการ ตรวจสอบ	ข้อมูลของการ ตรวจสอบ หลักฐาน	ผู้ใช้ระบบ	Process 10.2 การการตรวจสอบ หลักฐานภารกิจ	ชื่อหลักฐาน+สถานะ หลักฐาน+ละติจูด+ ลองจิจูด+เวลาในการส่ง+ [รูปภาพ วิดีโอ ข้อความ]
		Process 10.2 การการ ตรวจสอบ	เพิ่มข้อมูลหลักฐาน	

ตารางที่ 3.2 Data Flow and Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
		หลักฐานภารกิจ		
ข้อมูลพิกัด ที่เพิ่ม	ข้อมูลพิกัด ที่เพิ่ม	ผู้ใช้ระบบ	Process 11.0 การเพิ่มพิกัด Real time	ละติจูด+ลองจิจูด+เวลาที่ เข้าร่วม
		Process 11.0 การเพิ่มพิกัด Real time	เพิ่มข้อมูลเข้าร่วม การแข่งขัน	
		Process 11.0 การเพิ่มพิกัด Real time	ผู้ใช้ระบบ	

3.3.3 Data Store Description

ตารางที่ 3.3 Data Store Description

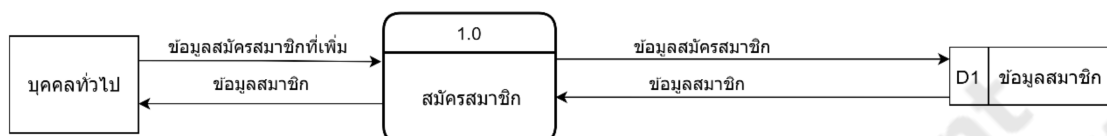
ID	Data Store	Description	Data Structure
D1	เพิ่มข้อมูลสมาชิก	เก็บรายละเอียดของผู้ใช้ระบบ	รหัสสมาชิก+อีเมล+รหัสผ่าน+ชื่อ-นามสกุล+ชื่อในระบบ+รูปโปรไฟล์+(คำอธิบายตัวเอง)
D2	เพิ่มข้อมูลการแข่งขัน	เก็บรายละเอียดของการแข่งขัน	รหัสการแข่งขัน+ชื่อการแข่งขัน+รายละเอียดสถานที่จัดกิจกรรม+ จำนวนทีมที่เข้าร่วม+วันที่เปิดรับ+วันที่ปิดรับสมัคร+วันที่จัดกิจกรรม+เวลาเริ่มกิจกรรม+เวลาสิ้นสุดกิจกรรม+รหัสสมาชิก
D3	เพิ่มข้อมูลภารกิจ	เก็บรายละเอียดของภารกิจ	รหัสภารกิจ+ชื่อภารกิจ+ลำดับของภารกิจ+รายละเอียดภารกิจ+คำอธิบาย+ภาพ วิดีโอประกอบภารกิจ+ประเภทภารกิจ+ส่งหลักฐาน+ระยะของการเจอค่าไปของภารกิจ+ละติจูด+ลองจิจูด+รหัสการแข่งขัน+รหัสสมาชิก
D4	เพิ่มข้อมูลคะแนนรีวิว	เพิ่มข้อมูลคะแนนรีวิว	รหัสรีวิว+ข้อความรีวิว+คะแนนรีวิว+วันที่เวลา+รหัสสมาชิก+รหัสการแข่งขัน

ตารางที่ 3.3 Data Store Description (ต่อ)

ID	Data Store	Description	Data Structure
D5	เพิ่มข้อมูลแชท	เก็บรายละเอียดของแชท	รหัสแชท+รหัสผู้ใช้งานระบบ+รหัสการแข่งขัน+ข้อความแชท+เวลาที่ส่ง
D6	เพิ่มข้อมูลเข้าร่วมการแข่งขัน	เก็บรายละเอียดของการเข้าร่วมการแข่งขัน	รหัสเข้าร่วมการแข่งขัน+ละติจูด+ลองจิจูด+เวลาที่เข้าร่วม+รหัสสมาชิก+รหัสทีม+รหัสการแข่งขัน
D7	เพิ่มข้อมูลทีม	เก็บรายละเอียดของทีม	รหัสทีม+ชื่อทีม+[รหัสสมาชิก]+รหัสการแข่งขัน
D8	เพิ่มข้อมูลรางวัล	เก็บรายละเอียดของรางวัล	รหัสรางวัล+ชื่อรางวัล
D9	เพิ่มข้อมูลหลักฐาน	เก็บรายละเอียดของหลักฐาน	รหัสหลักฐาน+ชื่อหลักฐาน+สถานะหลักฐาน+ละติจูด+ลองจิจูด+เวลาในการส่ง+ลิงก์วิดีโอ+ลิงก์รูปภาพ+เวลาที่ส่งภารกิจ+ข้อความหลักฐาน+รหัสภารกิจ+รหัสทีม+รหัสสมาชิก+รหัสการแข่งขัน

3.4 คำอธิบายการประมวลผล (Process Description)

3.4.1 Process 1.0 สมัครสมาชิก



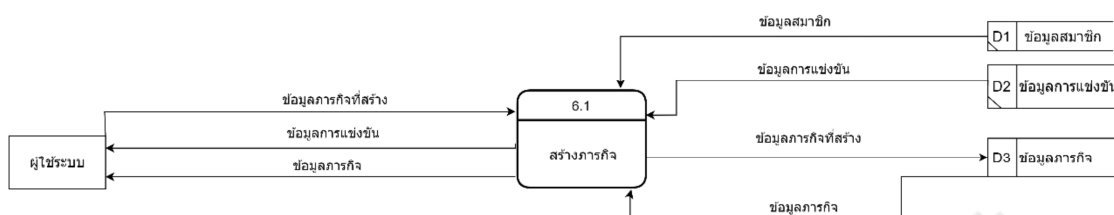
ID	1.0
Name	สมัครสมาชิก
Description	เพื่อทำการเพิ่มข้อมูลสมาชิกเข้าสู่ฐานข้อมูล
Input Data Flows	- ข้อมูลสมัครสมาชิก - ข้อมูลสมาชิก
Output Data Flows	- ข้อมูลสมัครสมาชิก - ข้อมูลสมาชิก
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลที่ต้องการสมัครสมาชิก 2. ตรวจสอบข้อมูลว่าครบถ้วนและถูกต้องหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้าข้อมูลถูกต้องไปทำข้อ (3) 2.2 ถ้าข้อมูลไม่ถูกต้องแสดงข้อความ “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. ตรวจสอบในแฟ้มข้อมูลสมาชิกว่ามีข้อมูลผู้ใช้หรือไม่ <ol style="list-style-type: none"> 3.1 ถ้ามีชื่อผู้ใช้แล้ว แสดงข้อมูลเตือน “มีบัญชีอยู่แล้ว” กลับไปทำข้อ (1) 3.2 ถ้าไม่มีชื่อผู้ใช้ในแฟ้มข้อมูล ให้บันทึกข้อมูลลงในฐานข้อมูลสมาชิก <p>จบการทำงาน</p>

3.4.2 Process 2.0 เข้าสู่ระบบ



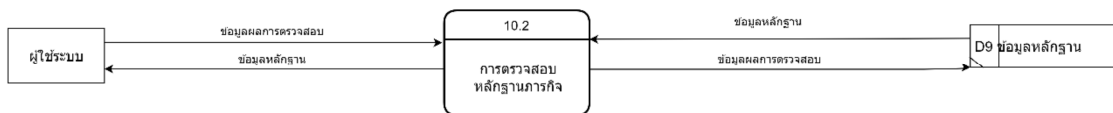
ID	2.0
Name	เข้าสู่ระบบ
Description	การเข้าสู่ระบบ
Input Data Flows	- ข้อมูลการเข้าสู่ระบบ - ข้อมูลสมาชิก
Output Data Flows	- ข้อมูลการเข้าสู่ระบบ
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลผู้ใช้ (อีเมล+รหัสผ่าน) 2. ตรวจสอบข้อมูลผู้ใช้ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลผู้ใช้ในแฟ้มข้อมูลสมาชิก ให้ทำข้อ (3) 2.2 ถ้าป้อนข้อมูลผู้ใช้ไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. ตรวจสอบข้อมูลผู้ใช้ในแฟ้มข้อมูลสมาชิก <ol style="list-style-type: none"> 3.1 ถ้ามีข้อมูลผู้ใช้ในฐานะข้อมูลให้แสดงข้อมูลการเข้าสู่ระบบ 3.2 ถ้ามีข้อมูลผู้ใช้ในฐานะข้อมูลให้แสดงข้อความ “ไม่พบบัญชี กรุณาตรวจสอบอีเมลหรือรหัสผ่านให้ถูกต้อง” <p>จบการทำงาน</p>

3.4.3 Process 6.1 สร้างภารกิจ



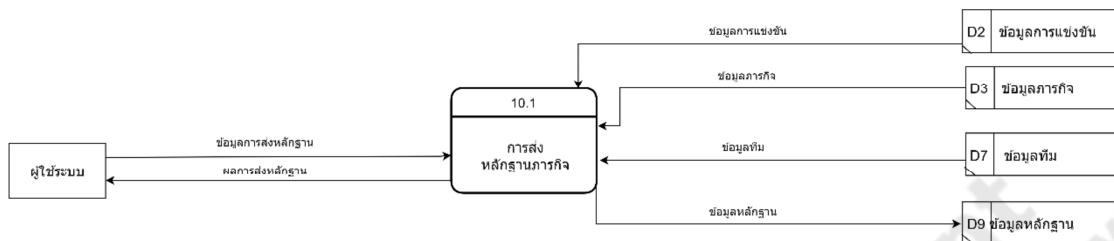
ID	6.1
Name	สร้างภารกิจ
Description	การสร้างภารกิจในการแข่งขัน
Input Data Flows	- ข้อมูลภารกิจที่สร้าง - ข้อมูลภารกิจ
Output Data Flows	- ข้อมูลภารกิจที่สร้าง
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลภารกิจที่สร้าง 2. ตรวจสอบข้อมูลครบถ้วนหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลครบถ้วน ให้ไปทำข้อ (3) 2.2 ถ้าป้อนข้อมูลไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. บันทึกข้อมูลภารกิจที่สร้างลงฐานข้อมูล <p>จบการทำงาน</p>

3.4.4 Process 10.2 การตรวจสอบหลักฐานภารกิจ



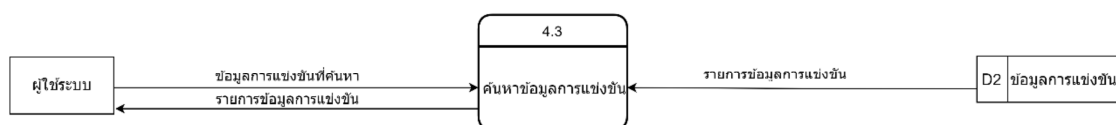
ID	10.2
Name	การตรวจสอบหลักฐานภารกิจ
Description	การสร้างภารกิจในการแข่งขัน
Input Data Flows	- ข้อมูลผลการตรวจสอบ - ข้อมูลหลักฐาน
Output Data Flows	- ข้อมูลหลักฐาน
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลผลการตรวจสอบ 2. ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลครบถ้วน ให้ไปทำข้อ (3) 2.2 ถ้าป้อนข้อมูลไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. การตรวจสอบหลักฐานภารกิจการตรวจสอบหลักฐานภารกิจ ลงในฐานข้อมูล <p>จบการทำงาน</p>

3.4.5 Process 10.1 การส่งหลักฐานภารกิจ



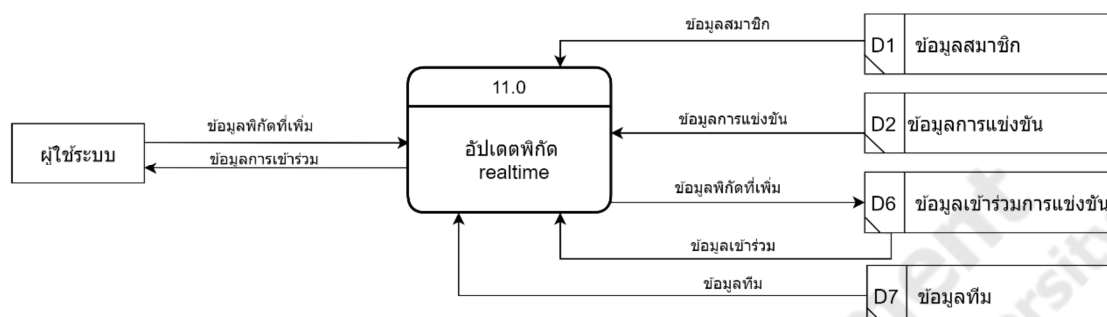
ID	10.1
Name	การส่งหลักฐานภารกิจ
Description	การส่งหลักฐานภารกิจ
Input Data Flows	- ข้อมูลการส่งหลักฐาน - ข้อมูลหลักฐาน
Output Data Flows	- ข้อมูลการส่งหลักฐาน - ผลการส่งหลักฐาน
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลการส่งหลักฐาน 2. ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลครบถ้วนให้ไปทำข้อ (3) 2.2 ถ้าป้อนข้อมูลไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. ส่งหลักฐานภารกิจ บันทึกการส่งหลักฐานภารกิจลงในฐานข้อมูล <p>จบการทำงาน</p>

3.4.6 Process 4.3 ค้นหาข้อมูลการแข่งขัน



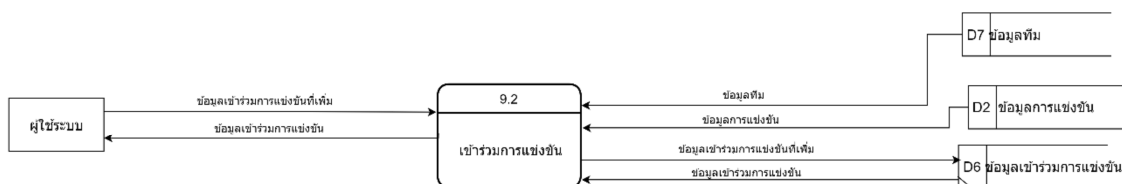
ID	4.3
Name	ค้นหาข้อมูลการแข่งขัน
Description	การค้นหาข้อมูลการแข่งขัน
Input Data Flows	- ข้อมูลการแข่งขันที่ค้นหา - รายการข้อมูลการแข่งขัน
Output Data Flows	- รายการข้อมูลการแข่งขัน - ข้อมูลการแข่งขันที่ค้นหา
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูล การ ค้นหาข้อมูลการแข่งขัน 2. ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลครบถ้วนให้ไปทำข้อ (3) 2.2 ถ้าป้อนข้อมูลไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. แสดงข้อมูลการค้นหาข้อมูลการแข่งขันให้ผู้ใช้ระบบ <p>จบการทำงาน</p>

3.4.7 Process 11.0 เพิ่มพิกัด real time



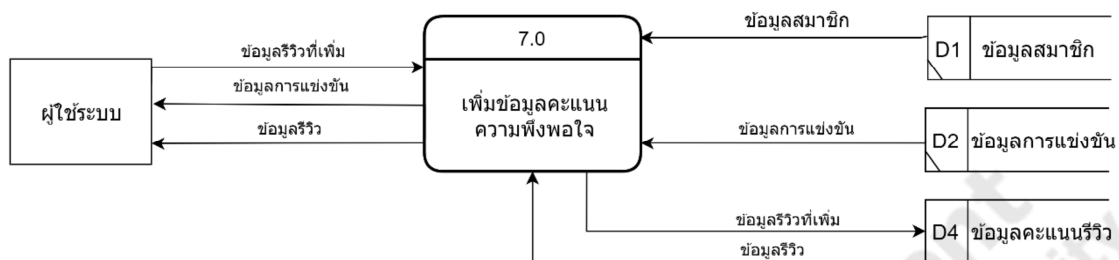
ID	11.0
Name	การเพิ่มพิกัด real time
Description	การเพิ่มพิกัด real time
Input Data Flows	- ข้อมูลพิกัดที่เพิ่ม
Output Data Flows	- ข้อมูลพิกัดที่เพิ่ม
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลเพิ่มพิกัด Real time 2. ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลครบถ้วนให้ไปทำข้อ (3) 2.2 ถ้าป้อนข้อมูลไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. เพิ่มพิกัด Realtime บันทึกการเพิ่มพิกัด Realtime ลงในฐานข้อมูล <p>จบการทำงาน</p>

3.4.8 Process 9.2 การเข้าร่วมการแข่งขัน



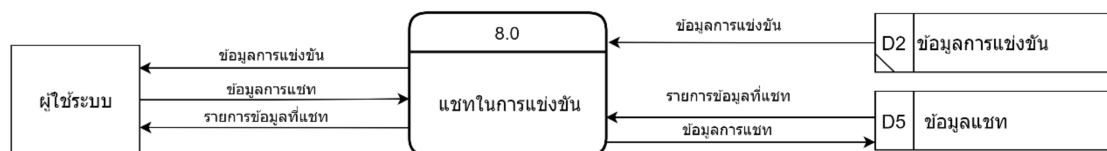
ID	9.2
Name	การเข้าร่วมการแข่งขัน
Description	การเข้าร่วมการแข่งขัน
Input Data Flows	- ข้อมูลเข้าร่วมที่เพิ่ม - ข้อมูลการเข้าร่วม
Output Data Flows	- ข้อมูลการแข่งขัน - ข้อมูลเข้าร่วม
Process Description	เริ่มต้น <ol style="list-style-type: none"> 1. รับข้อมูลการเข้าร่วมการแข่งขัน 2. ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลครบถ้วนให้ไปทำข้อ (3) 2.2 ถ้าป้อนข้อมูลไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. การเข้าร่วมการแข่งขัน บันทึกการเข้าร่วมการแข่งขันลงในฐานข้อมูล จบการทำงาน

3.4.9 Process 7.0 เพิ่มข้อมูลคะแนนความพึงพอใจ



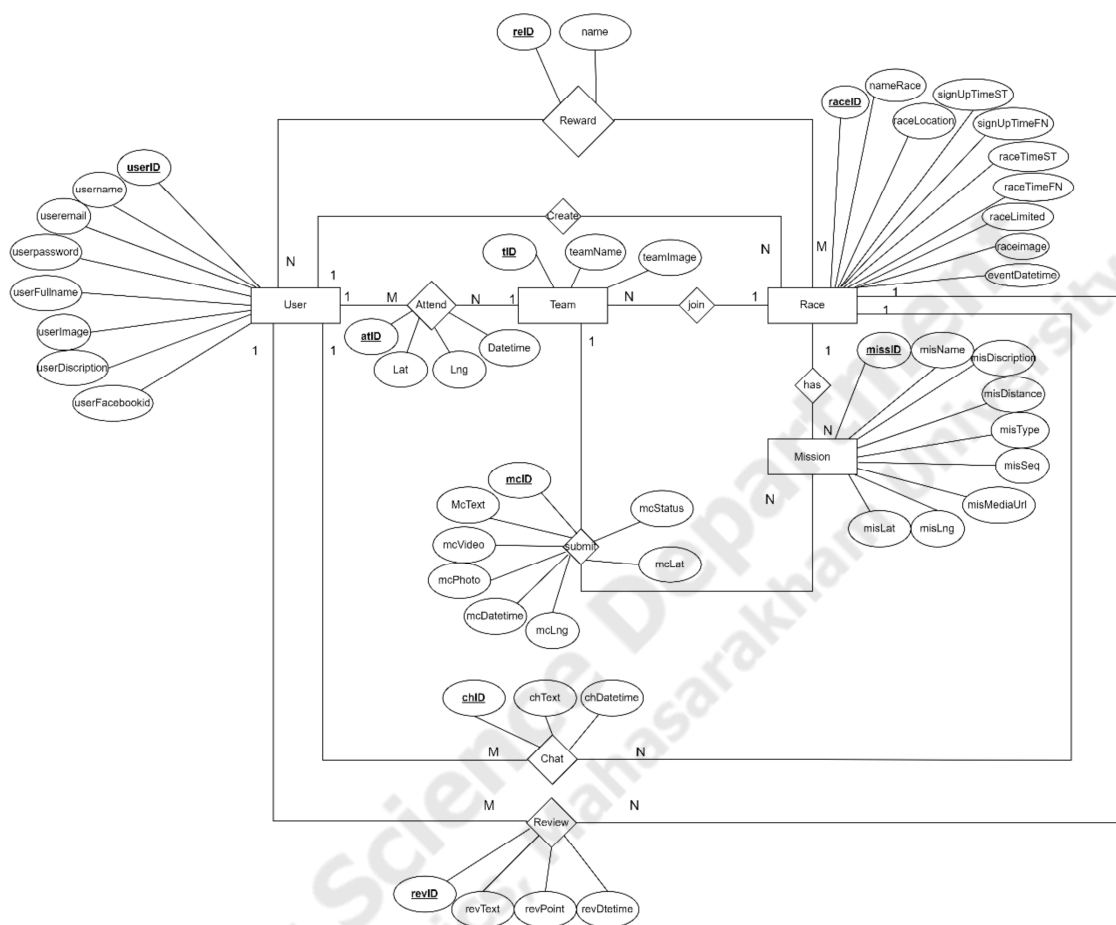
ID	7.0
Name	เพิ่มข้อมูลคะแนนความพึงพอใจ
Description	การเพิ่มข้อมูลคะแนนความพึงพอใจ
Input Data Flows	- ข้อมูลคะแนนที่เพิ่ม
Output Data Flows	- ข้อมูลการแข่งขัน - ข้อมูลคะแนนที่เพิ่ม
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลข้อมูลรีวิวที่เพิ่ม 2. ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ <ol style="list-style-type: none"> 2.1 ถ้ามีข้อมูลครบถ้วนให้ไปทำข้อ (3) 2.2 ถ้าป้อนข้อมูลไม่ถูกต้อง แสดง “กรุณาป้อนข้อมูลให้ถูกต้อง” กลับไปทำข้อ (1) 3. เพิ่มข้อมูลคะแนนความพึงพอใจ บันทึกเพิ่มข้อมูลคะแนนความพึงพอใจลงในฐานข้อมูล <p>จบการทำงาน</p>

3.4.10 Process 8.0 แชทในการแข่งขัน



ID	8.0
Name	แชทในการแข่งขัน
Description	การแชทในการแข่งขัน
Input Data Flows	- ข้อการแชท - รายการข้อมูลที่แชท
Output Data Flows	- ข้อมูลการแข่งขัน - ข้อมูลคะแนนที่เพิ่ม
Process Description	เริ่มต้น <ol style="list-style-type: none"> 1. ส่งหรือรับข้อมูลการสนทนาในการแข่งขัน 2. เพิ่มข้อมูลลงในฐานข้อมูลการแชท 3. แสดงข้อมูลการแชทในการแข่งขัน จบการทำงาน

3.5 ความสัมพันธ์ (Entity Relationship Diagram)



ภาพประกอบที่ 3.8 ความสัมพันธ์ (Entity Relationship Diagram)

3.6 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ข้อมูลผู้ใช้งานระบบ (user)

Id	Column	Type	Description	Example Data	Constraints
1	userID	int	รหัสผู้ใช้งานระบบ	1	PK
2	userName	varchar(100)	ชื่อในระบบ	เจมส์อิชีย์	Not null
3	userMail	varchar(50)	อีเมล	jame123@gmail.com	Not null
4	userPassword	varchar(50)	รหัสผ่าน	jame1234	Not null

ตารางที่ 3.4 ข้อมูลผู้ใช้งานระบบ (user) (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
5	userFullname	varchar(100)	ชื่อ-นามสกุล	จิรศักดิ์ ทิพย์สุขศรี	Not null
6	userImage	varchar(255)	รูปภาพ	-	Null
7	userDescription	varchar(255)	คำอธิบายตัวเอง	ไว้มาเล่นด้วยกันนะ	Null
8	userFacebookid	varchar(100)	บัญชีเฟซบุ๊ก	-	Null

ตารางที่ 3.5 ข้อมูลทีม (team)

Id	Column	Type	Description	Example Data	Constraints
1	teamID	int	รหัสทีม	1	PK
2	teamName	varchar(100)	ชื่อทีม	ทีมได้อยู่	Not null
3	teamImage	varchar(255)	รูปภาพ	-	Null
4	raceID	int	รหัสการแข่งขัน	1	FK reference from race(raceID) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.6 ข้อมูลรางวัล (reward)

Id	Column	Type	Description	Example Data	Constraints
1	reID	int	รหัสรางวัล	1	PK

ตารางที่ 3.6 ข้อมูลรางวัล (reward) (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
2	reusername	varchar(100)	ชื่อเหรียญ	ทอง	Not null
3	userID	int	รหัสผู้ใช้งานระบบ	1	FK reference from user(raceID) ON DELETE CASCADE ON UPDATE CASCADE
4	raceID	int	รหัสการแข่งขัน	1	FK reference from race(raceID) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.7 ข้อมูลรีวิว (review)

Id	Column	Type	Description	Example Data	Constraints
1	revID	int	รหัสรีวิว	1	PK
2	revText	varchar(255)	ข้อความรีวิว	เป็นการแข่งขันที่ดีแต่ร้อนไปหน่อย	Null
3	revPoint	int	คะแนน(1-5)	4	Can be only 0-5
4	revDatetime	datetime	วันที่เวลา	12/12/2021 16:00:30	Not null

ตารางที่ 3.7 ข้อมูลรีวิวก (review) (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
5	userID	int	รหัสผู้ใช้งานระบบ	1	FK reference from user(raceID) ON DELETE CASCADE ON UPDATE CASCADE
6	raceID	int	รหัสการแข่งขัน	1	FK reference from race(raceID) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.8 ข้อมูลการแข่งขัน (race)

Id	Column	Type	Description	Example Data	Constraints
1	raceID	int	รหัสการแข่งขัน	1	PK
2	raceName	varchar(255)	ชื่อการแข่งขัน	มาเล่นกันเถอะมมส.	Null
3	raceLocation	varchar(255)	ชื่อสถานที่	มหาวิทยาลัยมหาสารคาม	Not null
4	signUpTimeST	date	วันที่เปิดรับสมัครการแข่งขัน	12/12/2021	Not null
5	signUpTimeFN	date	วันที่ปิดรับสมัครการแข่งขัน	13/12/2021	Not null
6	raceTimeST	time	เวลาเริ่มกิจกรรม	09:00:30	Not null
7	raceTimeFN	time	เวลาสิ้นสุดกิจกรรม	17:00:00	Not null

ตารางที่ 3.8 ข้อมูลการแข่งขัน (race) (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
8	eventDatetime	date	วันที่จัดกิจกรรม	14/12/2021	Not null
9	raceLimitteam	int	จำนวนทีม	5	Not null
10	raceimage	varchar(1000)	url รูปภาพ	MVXmYY99_40 0x400.jpg (400x400) (twimg.com)	Null
11	userID	int	รหัสผู้ใช้งาน ระบบที่สร้าง	1	FK reference from user(raceID) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.9 ข้อมูลภารกิจ (mission)

Id	Column	Type	Description	Example Data	Constraints
1	misID	int	รหัสภารกิจ	1	PK
2	misName	varchar(255)	ชื่อภารกิจ	รถไฟป่วนๆ	Not null
3	misDiscription	varchar(255)	คำอธิบายภารกิจ	จงตามรถขบวน ยาวที่วิ่งไม่ได้ที่ มมส.	Not null
4	misDistance	int	ระยะของการเจอ คำใบ้ของภารกิจ	5	Not null
5	misType	int	ประเภทของ ภารกิจ(0=เช็ค ระยะทาง ,1=สื่อ(รูปภาพ ,วิดีโอ) ,2=ข้อความ)	2	can be only 0,1 and 2

ตารางที่ 3.9 ข้อมูลภารกิจ (mission) (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
6	misSeq	int	ลำดับภารกิจ	1	Not null
7	misMediaUrl	varchar(1000)	รูปภาพหรือวิดีโอของภารกิจ	-	Null
8	misLng	Double	ลองจิจูด	41.40338	Not null
9	misLat	Double	ละติจูด	2.17403	Not null
10	raceID	int	รหัสการแข่งขัน	1	FK reference from race(raceID) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.10 ข้อมูลการส่งหลักฐาน (missionComplete)

Id	Column	Type	Description	Example Data	Constraints
1	mcID	int	รหัสหลักฐาน	1	PK
2	mcText	varchar(255)	ข้อความหลักฐาน	รถไฟ	Null
3	mcVideo	varchar(255)	ลิงก์วิดีโอ	https://www.youtube.com/watch?v=xLD3giyGs5Q	Null
4	mcPhoto	varchar(255)	ลิงก์รูปภาพ	https://www.prachachat.net/wp-content/uploads/2021/11/S_25796639.jpg	Null
5	mcDatetime	datetime	เวลาที่ส่งภารกิจ	13/12/2022	Not null

ตารางที่ 3.10 ข้อมูลการส่งหลักฐาน (missionComplete) (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
				09:00:30	
6	mcLng	double	ลองจิจูด	41.40338	Not null
7	mcLat	double	ละติจูด	2.17403	Not null
8	mcStatus	int	สถานะ(0=ภารกิจไม่สำเร็จ,1=กำลังทำภารกิจ,2=ภารกิจไม่สำเร็จ)	1	can be only 0,1 and 2
9	misID	int	รหัสภารกิจ	1	FK reference from mission(misID) ON DELETE CASCADE ON UPDATE CASCADE
10	teamID	int	รหัสทีม	1	FK reference from team(teamID) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.11 ข้อมูลแชท (Chat)

Id	Column	Type	Description	Example Data	Constraints
1	chID	int	รหัสแชท	1	PK
2	chText	varchar(255)	ข้อความแชท	สวัสดีค่ะ	Not null
3	chDatetime	datetime	เวลาที่ส่งแชท	13/12/2022 09:00:30	Not null

ตารางที่ 3.11 ข้อมูลแชท (Chat) (ต่อ)

Id	Column	Type	Description	Example Data	Constraints
4	userID	int	รหัสผู้ใช้งาน ระบบ	1	FK reference from user(raceID) ON DELETE CASCADE ON UPDATE CASCADE
5	raceID	int	รหัสการแข่งขัน	1	FK reference from race(raceID) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.12 ข้อมูลการเข้าร่วม (atID)

Id	Column	Type	Description	Example Data	Constraints
1	atID	int	รหัสการเข้าร่วม	1	PK
2	Lat	double	ลองจิจูด	41.40002	Not null
3	Lng	double	ละติจูด	2.15053	Not null
4	Datetime	datetime	เวลาการเข้าร่วม	13/12/2022 09:00:30	Not null
5	userID	int	รหัสผู้ใช้งาน ระบบ	1	FK reference from user(raceID) ON DELETE CASCADE ON UPDATE CASCADE
6	teamID	int	รหัสทีม	1	FK reference from team(teamID) ON DELETE CASCADE ON UPDATE CASCADE

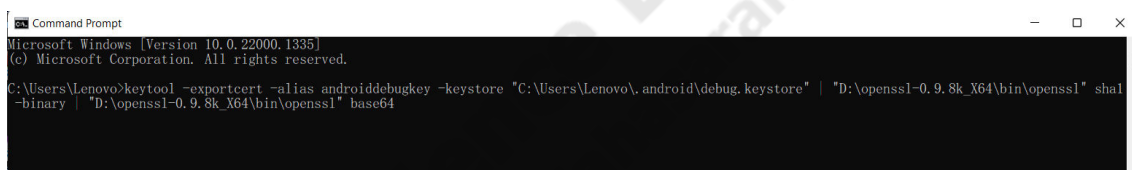
3.7 การพัฒนาระบบ

3.7.1 การเข้าสู่ระบบโดยใช้ Facebook และสมัครสมาชิกโดยใช้ Facebook

Flutter Facebook login คือ ชุดคำสั่งที่ช่วยให้ผู้ใช้เข้าสู่ระบบแอปพลิเคชันด้วย Facebook เมื่อเข้าสู่แอปพลิเคชันด้วย Facebook แล้วผู้ใช้งานจะสามารถให้สิทธิ์การอนุญาตเข้าแอปพลิเคชันเพื่อให้ รับข้อมูลหรือดำเนินการบน Facebook สามารถเข้าสู่ระบบใช้งานบน IOS, Android, Web และ Application บน Desktop เข้าสู่ระบบได้ง่ายๆเพื่อการยืนยันตัวตนและการเข้าถึงข้อมูลการแจ้งเตือน เมื่อผู้ใช้เข้าสู่ระบบผ่าน โดยจะเรียกใช้ชุดคำสั่งของ Facebook ในการเข้าสู่ระบบและใช้สิทธิ์อนุญาตใน การเข้าสู่ระบบ

ต้องสร้าง Key hash ของเครื่องที่เขียนโปรแกรม Android นั้นๆ ต้องทำทุกเครื่องที่ใช้ Build application ถ้าเป็นเครื่อง Build ก็ต้องทำใหม่

ใช้โปรแกรม openssl ในการรัน ถ้าเครื่องไม่มี openssl ก็ให้ดาวน์โหลดจาก <https://code.google.com/archive/p/openssl-for-windows/downloads>

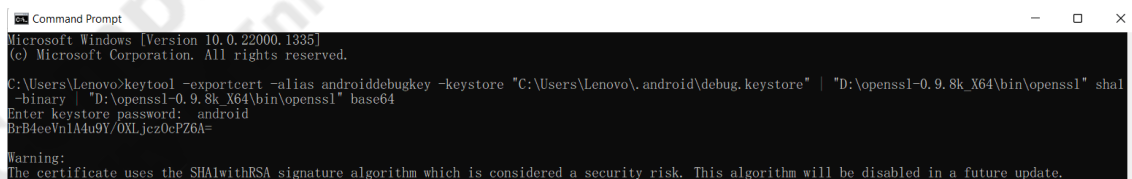


```
Microsoft Windows [Version 10.0.22000.1335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>keytool -exportcert -alias androiddebugkey -keystore "C:\Users\Lenovo\.android\debug.keystore" | "D:\openssl-0.9.8k_X64\bin\openssl" sha1 -binary | "D:\openssl-0.9.8k_X64\bin\openssl" base64
```

ภาพประกอบที่ 3.9 รันคำสั่ง keytool

รันคำสั่ง โดย C:\Users\Lenovo\.android\debug.keystore เป็น path ของ user ในเครื่อง windows และ D:\openssl-0.9.8k_X64\bin\openssl เป็น path ที่เก็บโปรแกรม openssl ที่แตก zip ไว้



```
Microsoft Windows [Version 10.0.22000.1335]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>keytool -exportcert -alias androiddebugkey -keystore "C:\Users\Lenovo\.android\debug.keystore" | "D:\openssl-0.9.8k_X64\bin\openssl" sha1 -binary | "D:\openssl-0.9.8k_X64\bin\openssl" base64
Enter keystore password: android
BrB4eeVn1A4u9V/OXLjcz0cPZ6A=

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 Key hash ที่ได้

password ให้ใส่เป็น "android" แล้วจะได้ Key hash ออกมา

```

1 <?xml version="1.0" encoding="utf-8"?> You, 2 weeks ago • FBloginVer1
2 <resources>
3 <string name="app_name">demo_login</string>
4 <string name="facebook_app_id">733574481658743</string>
5 <string name="fb_login_protocol_scheme">fb733574481658743</string>
6 <string name="facebook_client_token">acd17bb077da68857da4839ce0d089fe</string>
7 </resources>

```

ภาพประกอบที่ 3.11 สร้างไฟล์ชื่อ *strings.xml

บรรทัดที่ 3-6 เพิ่ม tag string ที่มีชื่อ facebook_app_id, fb_login_protocol_scheme และ facebook_client_token และตั้งค่าให้กับ ID ของแอปพลิเคชันและ token client ตรงตามแอปพลิเคชันของเรา

```

1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2     package="com.example.miniworldapp">
3     <uses-permission android:name="android.permission.INTERNET"/>
4     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
5     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
6     <application
7         android:label="miniworldapp"
8         android:name="${applicationName}"

```

ภาพประกอบที่ 3.12 การกำหนดค่าในไฟล์ /AndroidManifest.xml เพื่ออนุญาตการเชื่อมต่ออินเทอร์เน็ต

บรรทัดที่ 3 จะทำการกำหนดค่าไฟล์เพื่อเป็นการอนุญาตการเชื่อมต่ออินเทอร์เน็ต

```

1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="com.example.miniworldapp">
3   <uses-permission android:name="android.permission.INTERNET" />
4   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
5   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
6   <application
7     android:label="miniworldapp"
8     android:name="${applicationName}"
9     android:icon="@mipmap/ic_launcher">
10    <meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/facebook_app_id"/>
11    <meta-data android:name="com.facebook.sdk.ClientToken" android:value="@string/facebook_client_token"/>
12    <activity android:name="com.facebook.FacebookActivity"
13      android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
14      android:label="@string/app_name" />
15    <activity
16      android:name="com.facebook.CustomTabActivity"
17      android:exported="true">
18      <intent-filter>
19        <action android:name="android.intent.action.VIEW" />
20        <category android:name="android.intent.category.DEFAULT" />
21        <category android:name="android.intent.category.BROWSABLE" />
22        <data android:scheme="@string/fb_login_protocol_scheme" />
23      </intent-filter>
24    </activity>
25    <intent-filter>
26      <action android:name="android.intent.action.MAIN" />
27      <category android:name="android.intent.category.LAUNCHER" />
28      <action android:name="android.intent.action.VIEW" />
29      <category android:name="android.intent.category.DEFAULT" />
30      <category android:name="android.intent.category.BROWSABLE" />
31      <data android:scheme="@string/fb_login_protocol_scheme" />
32    </intent-filter>
33  </application>
34  <meta-data
35    android:name="flutterEmbedding"
36    android:value="2" />
37 </manifest>
38
39

```

ภาพประกอบที่ 3.13 การกำหนด meta-data ในไฟล์ /AndroidManifest.xml

บรรทัดที่ 35-49 เพิ่ม tag meta-data ให้กับ tag application สำหรับ ID ของแอปพลิเคชัน และ token client ของแอปพลิเคชันในส่วนของการเรียกใช้งานจาก Flutter

```

47 ElevatedButton(
48   onPressed: () async {
49     final LoginResult result = await FacebookAuth.instance.login();
50
51     if (result.status == LoginStatus.success) {
52
53       final AccessToken accessToken = result.accessToken!;
54       log(accessToken.token);
55       final userData = await FacebookAuth.instance.getUserData();
56       log(userData.toString());

```

ภาพประกอบที่ 3.14 สร้างปุ่ม login ผ่าน facebook (Front End)

บรรทัดที่ 37 เป็นการสร้างปุ่ม login ผ่าน Facebook

- บรรทัดที่ 38** ประกาศตัวแปร result ที่มีชนิดข้อมูลเป็น LoginResult เพื่อเก็บค่าที่ได้จากการเรียกใช้งานการเข้าสู่ระบบด้วย Facebook โดยใช้ Facebook API
- บรรทัดที่ 40** นำค่าที่ได้จากการเข้าสู่ระบบมาตรวจสอบ HttpStatus หากมีค่าเท่ากับ LoginStatus.success หรือการเข้าสู่ระบบสำเร็จจะทำงานบรรทัดถัดไป
- บรรทัดที่ 42** ประกาศตัวแปร accessToken ที่มีชนิดข้อมูลเป็น AccessToken เพื่อเก็บค่า AccessToken ที่ได้จากการเข้าสู่ระบบ accessToken มีการเก็บข้อมูลผู้ใช้งาน Facebook ไว้ในตัวแปร userData
- บรรทัดที่ 44** ประกาศตัวแปร userData เพื่อ เก็บข้อมูลของผู้ใช้ที่ได้จาก Facebook โดย Facebook API จะดึงข้อมูลของผู้ใช้ ออกมาโดยใช้ เมธอด getUserData() เมื่อเราได้ตัวแปร userData แล้วต้องการส่งค่า userData ไปยังหน้าอื่นๆ เราจำเป็นต้องสร้างตัวแปรมาเก็บค่าไว้ใน Provider เพื่อส่งค่าตัวแปรนั้นไปยังหน้าต่างๆหรือรับค่าเข้ามา

```

3 class AppData with ChangeNotifier {
4   //Api baseurl
5   String baseurl = "http://202.28.34.197:9131";
6
7   Map<String,dynamic>userFacebook ={};
8 }

```

ภาพประกอบที่ 3.15 การเก็บค่าไว้ใน provider (Front End)

- บรรทัดที่ 7** สร้างตัวแปรมาเก็บค่าข้อมูลผู้เข้ามาเก็บไว้ใน userFacebook เพื่อดึงข้อมูลผู้ใช้ในหน้าสมัคร สมาชิก

ในส่วนต่อไปทำการสร้างคลาสในภาษา Go เพื่อทำการรับ Attribute ของ Entity จาก Front End

```

8 type LoginFaceDTO struct {
9   UserFacebookID string `json:"facebookid"`
10 }

```

ภาพประกอบที่ 3.16 ประกาศคลาส LoginFaceDTO Attribute ของ Entity (BackEnd)

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

```

15 func UserController(router *gin.Engine) {
16   user := router.Group("/user")
17
18   user.POST("/loginfacebook", loginface)
19 }
20

```

ภาพประกอบที่ 3.17 สร้าง Func เพื่อเรียก func loginface (BackEnd)

บรรทัดที่ 15-19 สร้างตัวแปร user มาเก็บ router.Group เพื่อจัดกลุ่ม Path และเรียกใช้ตัวแปร มารระบุ Path ให้กับ Controller โดยเรียกเป็น POST

```

22 func loginface(c *gin.Context) {
23     loginfaceDTO := dto.LoginFaceDTO{}
24     err := c.ShouldBindJSON(&loginfaceDTO)
25     fmt.Print(loginfaceDTO)
26     if err != nil {
27         c.JSON(http.StatusBadRequest, err)
28     }
29     user := UserServ.LoginFace(loginfaceDTO)
30
31     if user.UserID == 0 {
32         c.JSON(http.StatusOK, user)
33     } else {
34         c.JSON(http.StatusOK, user)
35     }
36 }

```

ภาพประกอบที่ 3.18 Controller ส่วนเข้าสู่ระบบด้วย ID Facebook (Back End)

บรรทัดที่ 23 สร้างตัวแปร LoginfaceDTO เพื่อดึงตัวแปร LoginFaceDTO มาเก็บ

บรรทัดที่ 24 สร้างตัวแปร err เพื่อเก็บค่า c.ShouldBindJSON คือการ bind request body ให้เป็นรูปแบบ JSON กับ Struct

บรรทัดที่ 26-28 นำตัวแปร err มาเช็ค null หากไม่มีค่า null จะ แสดง StatusBadRequest

บรรทัดที่ 31-35 นำตัวแปร UserID มาเช็ค หากเท่ากับ 0 จะแสดง StatusOK

ในส่วนต่อไปทำการสร้างคลาส User เพื่อทำการรับ Attribute ของ Entity ในฐานข้อมูล

```

3  type User struct {
4      UserID      int32   `gorm:"column:userID;primaryKey"`
5      UserName     string  `gorm:"column:userName; size=100"`
6      UserMail     string  `gorm:"column:userMail; size=50"`
7      UserPassword string  `gorm:"column:userPassword; size=50"`
8      UserFullname string  `gorm:"column:userFullname; size=50"`
9      UserImage    string  `gorm:"column:userImage; size=255"`
10     UserDiscription string `gorm:"column:userDiscription; size=255"`
11     UserFacebookID string `gorm:"column:userFacebookID; size=100"`
12 }
13
14
15 func (User) TableName() string {
16     return "user"
17 }
18

```

ภาพประกอบที่ 3.19 ประกาศคลาส User พร้อมทั้งสร้าง Attribute ของ Entity (Back End)

บรรทัดที่ 3-13 ประกาศคลาส User พร้อมทั้งสร้าง Attribute ของ Entity สำหรับเก็บฐานข้อมูล Database

บรรทัดที่ 15-17 ประกาศคลาส TableName() ที่เก็บคลาส User พร้อม return ชื่อ Table ในฐานข้อมูล

ในส่วนต่อไปทำการประกาศคลาสเพื่อทำการเก็บคำสั่งที่ติดต่อไป Database

```

10 type userRepo struct {
11     db *gorm.DB
12 }

```

ภาพประกอบที่ 3.20 ประกาศคลาส UserRepo (Back End)

บรรทัดที่ 10-12 ประกาศคลาส userRepo เพื่อเก็บคำสั่ง db*gorm.DB เป็นการติดต่อกับฐานข้อมูล

```

14 // LoginFace implements UserService
15 func (UserServ) LoginFace(logface dto.LoginFaceDTO) model.User {
16     user := userRepo.LoginFace(logface)
17
18     if user.UserID == 0 {
19         return model.User{}
20     } else {
21         fmt.Printf("%v", &user.UserFacebookID)
22         return user
23     }
24
25 }

```

ภาพประกอบที่ 3.21 Service ในการเข้าสู่ระบบ ด้วย Facebook ID (Back End)

บรรทัดที่ 16 สร้างตัวแปร user เพื่อเก็บ userRepo ที่ต้องการเรียกใช้และเรียกใช้ userRepo.LoginFace ที่สร้างไว้และนำ logface มาใส่

บรรทัดที่ 18-23 นำตัวแปร UserID มาเช็ค หากเท่ากับ 0 จะ return model.User หากเข้าอีกเงื่อนไข จะ print user.UserFacebookID และ return user

```

14 // LoginFace implements UserRepository
15 func (userRepo) LoginFace(logface dto.LoginFaceDTO) model.User {
16     var user model.User
17     result := db.Where("UserFacebookID=?", logface.UserFacebookID).First(&user)
18     if result.Error != nil {
19         return model.User{}
20     }
21     return user
22 }

```

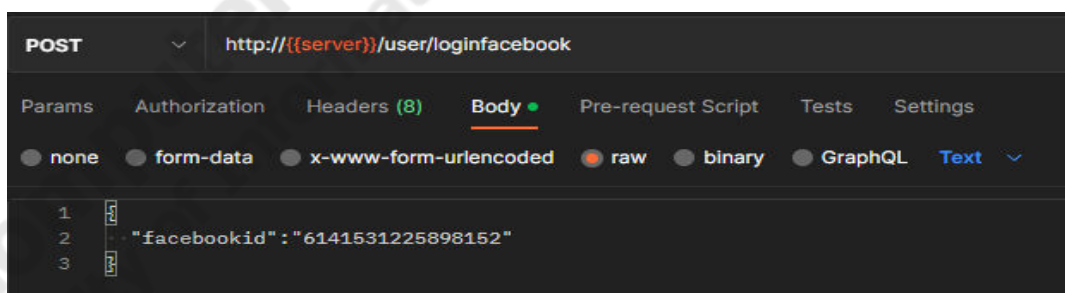
ภาพประกอบที่ 3.22 Repository ในการเข้าสู่ระบบด้วย FacebookID (Back End)

บรรทัดที่ 15 ประกาศคลาส LoginFace โดยประกาศตัวแปร userRepo และรับค่า logface Type เป็น dto และ return ค่าเป็น model.User

บรรทัดที่ 16 ประกาศตัวแปร user เพื่อเก็บค่า model.User

บรรทัดที่ 17-21 สร้างตัวแปร result เพื่อเก็บคำสั่ง SQL เช็ค UserFacebookID และใช้คำสั่ง First (&user) เพื่อหาค่าข้อมูลในฐานข้อมูล Database เมื่อได้ค่าข้อมูลแล้วมาทำการเช็ค Error โดยการใช้คำสั่ง IF เมื่อ result.Error ไม่เท่ากับค่าว่าง จะ return model.User หากไม่อยู่ในเงื่อนไขจะ return user ไป

ในส่วนต่อไปทำการเรียกใช้ Service โดยการใส่ FacebookID ของผู้ใช้ใน PostMan



ภาพประกอบที่ 3.23 เรียกใช้ Service ล็อคอินโดยใช้ id จาก Facebook



ภาพประกอบที่ 3.24 ข้อมูล User ที่ได้จากการเรียกใช้ Service

ในส่วนของการเรียกใช้งานใน Flutter ทำการส่งค่าข้อมูลผู้ใช้ที่เข้าสู่ระบบ ด้วย Facebook ไปยัง Provider เพื่อเก็บค่าและทำการนำไปใช้หน้าต่างๆ

```

58 context.read<AppData>().userFacebook = userData;
59 idFB = userData['id'];
60 LoginFbdto loginFb = LoginFbdto(facebookid: idFB);
61 var login = await loginfbService.fblogin(loginFb);
62
63 log(jsonEncode(login.data.userId));
64 fbid = int.parse(jsonEncode(login.data.userId));

```

ภาพประกอบที่ 3.25 การส่งตัวแปร userData ของผู้ใช้ไปเก็บไว้ใน Provider (Front End)

- บรรทัดที่ 58** ดึงเอาตัวแปร userData ที่เก็บข้อมูลผู้ใช้ส่งค่าไปยัง Provider ที่รับค่าไว้เพื่อส่งข้อมูลผู้ใช้ไปยังหน้าหลักหรือหน้าสมัครสมาชิก
- บรรทัดที่ 59** สร้างตัวแปร idFB มาเก็บ ID ผู้ใช้โดยการดึง userData
- บรรทัดที่ 60** สร้างตัวแปร loginFB เพื่อเรียกใช้ model database ที่ชื่อ LoginDTO
- บรรทัดที่ 61** สร้างตัวแปร login ที่เรียก service เมื่อได้ข้อมูล idfacebookก็นำไปเก็บไว้ที่ login
- บรรทัดที่ 63** log เพื่อดูค่า userId
- บรรทัดที่ 64** แปลงค่า fbid จาก String เป็น int เพื่อนำไปเช็คค่าใน if

```

71 if(fbid > 0){
72     Navigator.push(context,
73         MaterialPageRoute(builder: (context) => const NewHome()));
74 }else{
75     Navigator.push(context,
76         MaterialPageRoute(builder: (context) => const RegisterPage()));
77 }

```

ภาพประกอบที่ 3.26 การเช็ค ID ผู้ใช้ (Front End)

- บรรทัดที่ 56-66** นำตัวแปร fbid มาเช็คค่าใน IF โดยใช้คำสั่ง เมื่อ fbid มีค่ามากกว่า 0 จะถูกส่งไปหน้า Home ถ้าหาก fbid มีค่าน้อยกว่า 0 จะถูกส่งไปยังหน้าสมัครสมาชิก

```

43 @override
44 void initState() {
45   super.initState();
46
47   registerService =
48     RegisterService(Dio(), baseUrl: context.read<AppData>().baseUrl);
49
50   userFacebook = context.read<AppData>().userFacebook;
51
52   _length = userFacebook['name'].length;
53   if (_length > 0) {
54     email.text = userFacebook['email'];
55     userFullName.text = userFacebook['name'];
56     image = userFacebook['picture']['data']['url'];
57
58     idFB = userFacebook['id'];

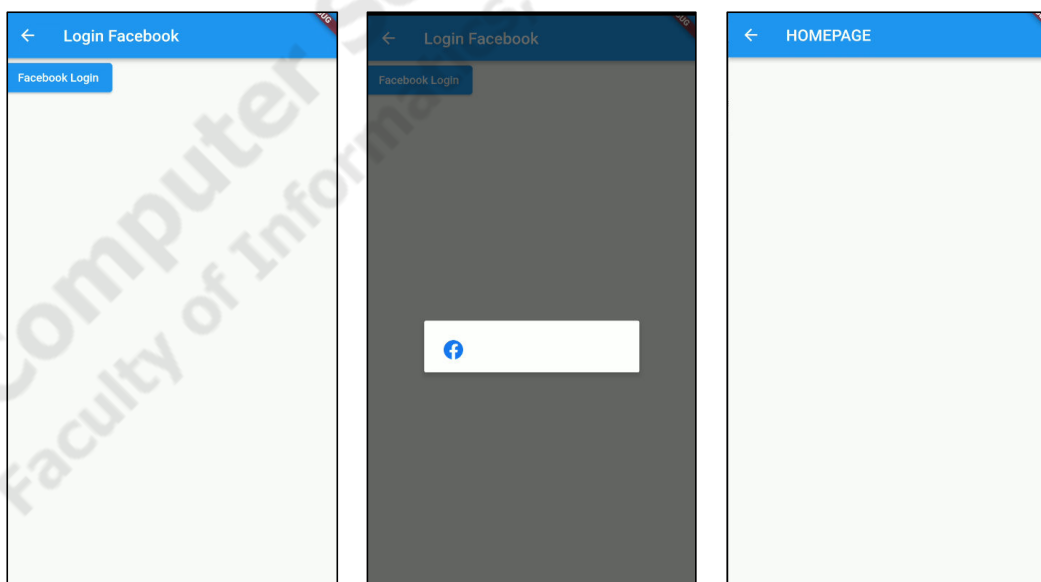
```

ภาพประกอบที่ 3.27 การส่งค่า Facebook ผู้ใช้ไปหน้าสมัครสมาชิก (Front End)

บรรทัดที่ 42-43 เป็น Method เมื่อเข้าหน้าสมัครสมาชิกจะถูกเรียกใช้

บรรทัดที่ 48 สร้างตัวแปร userFacebook เพื่อดึงข้อมูล Provider ที่เราเก็บค่าไว้มายังหน้าสมัครสมาชิก

บรรทัดที่ 50-56 สร้างตัวแปรมาเก็บข้อมูล email, ชื่อ, รูป, facebookID ของ userFacebook ที่เราดึงข้อมูลมาจาก Provider



ภาพประกอบที่ 3.28 ตัวอย่างการลือคอินด้วย Facebook

3.7.2 Google Maps API และ GPS

Google Maps API และ GPS Google Maps API เป็นชุด API ของ Google สำหรับพัฒนา Mobile application ไว้สำหรับเรียกใช้แผนที่และซุก Service ต่าง ๆ ของ Google เพื่อพัฒนา Application โดยการเขียน โปรแกรมด้วย Flutter ต้องลงไลบรารีชื่อ Google_maps_flutter เพื่อเรียกใช้ Google Maps API

GPS คือ ระบบกำหนดตำแหน่งบนพื้นโลก จะทำงานโดยรับสัญญาณ ตัวเครื่องรับ สัญญาณ GPS เช่น สมาร์ทโฟน โดยการคำนวณระยะทางโดย GPS ด้วยการเขียนโปรแกรมด้วย Flutter จะต้องเพิ่ม dependencies โดยใช้งาน 3 ตัว คือ Google Maps , Location สำหรับรับค่าตำแหน่ง และ url launcher สำหรับเปิด google maps จากแอปพลิเคชัน เริ่มใช้งาน google maps โดยการ import package google maps

```
import 'package:google_maps_flutter/google_maps_flutter.dart';
```

ภาพประกอบที่ 3.29 import package google maps

```
28 final Completer<GoogleMapController> _controller = Completer();
29 LatLng centerMap = const LatLng(16.245916, 103.252182);
30 List<LatLngDto> latlngDtos = [];
31 | late Future<void> loadDataMethod;
32 late AttendService attendService;
33 Set<Marker> markers = {};
```

ภาพประกอบที่ 3.30 ประกาศตัวแปรที่ใช้ใน class

- บรรทัดที่ 26 สร้าง Class showmap เพื่อโชว์แผนที่และระบุตำแหน่ง
- บรรทัดที่ 28 สร้าง GoogleMapController โดยใช้ Completer class ซึ่งก็คือ class ที่เอาไว้สร้าง future ที่จะแสดงแผนที่ google map ขึ้นมา
- บรรทัดที่ 29 สร้างตัวแปร centerMap เพื่อกำหนดค่าละติจูด ลองจิจูดของหน้าเริ่มต้นของแผนที่
- บรรทัดที่ 30 ประกาศตัวแปร LatLngDTO เป็น List เพื่อเรียกใช้งาน model ที่เก็บ database model ไว้
- บรรทัดที่ 31 กำหนดตัวแปร loadDataMethod เป็น Future<void>
- บรรทัดที่ 32 กำหนดตัวแปร AttendService เพื่อเรียกใช้ service
- บรรทัดที่ 33 ประกาศ Set เพื่อเก็บข้อมูลละติจูด ลองจิจูดของ marker

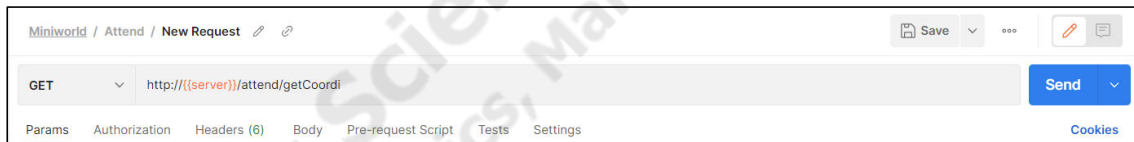
```

4 import 'package:miniworldapp/model/attend.dart';
5
6 part 'attend.g.dart';
7 @RestApi()
  You, 2 weeks ago | 1 author (You)
8 abstract class AttendService {
9   factory AttendService(Dio dio, {String baseUrl}) = _AttendService;
10
11   @GET("/attend/getCoordi")
12   Future<HttpResponse<List<LatlngDto>>> attend();
13 }

```

ภาพประกอบที่ 3.31 สร้าง RestClient โดยใช้ Retrofit

- บรรทัดที่ 4 เรียกใช้งาน API ที่มี path เป็น /attend สร้างไฟล์ชื่อ attend.dart ใน model
- บรรทัดที่ 6 เป็นการ generate คำสั่งใหม่เป็นไฟล์ชื่อ attend.g.dart
- บรรทัดที่ 7 ระบุ @RestApi เพื่อให้รู้ว่าคลาสนี้คือ Restful API
- บรรทัดที่ 11 สร้าง Method ไปเรียก API เรียกไปยัง path ที่มีชื่อ /attend/getCoordi
- บรรทัดที่ 12 สร้าง Method ของ Path ชื่อ LatlngDTO เพื่อเป็นการเรียก Method LatlngDTO ในไฟล์ attend ในโฟลเดอร์ model



ภาพประกอบที่ 3.32 ตัวอย่างการเรียกใช้ Service การเรียกดูข้อมูลในฐานข้อมูล getCoordi



ภาพประกอบที่ 3.33 ตัวอย่างข้อมูลที่ได้มาจากการเรียกใช้ Service


```

44 Future _goToMarker() async {
45   var latlngs = await attendService.attend();
46
47   for (var latlng in latlngs.data) {
48     var marker = Marker(
49       markerId: MarkerId(latlng.atId.toString()),
50       position: LatLng(latlng.lat!.toDouble(),latlng.lng!.toDouble()),);
51     markers.add(marker);
52   }
53   setState(() {
54
55   });

```

ภาพประกอบที่ 3.34 Method สำหรับนำ marker มาใช้กับ Google Maps

- บรรทัดที่ 44 สร้าง Method สำหรับนำ marker มาใช้กับ Google Maps
 บรรทัดที่ 45 สร้างตัวแปร latlngs เพื่อเรียกใช้ service ของ attend
 บรรทัดที่ 47 for สำหรับสร้าง marker ในแผนที่
 บรรทัดที่ 48 ประกาศตัวแปร marker และ กำหนด Set
 บรรทัดที่ 49 MarkerId สำหรับเก็บ id ของพิกัดละติจูดและลองจิจูด
 บรรทัดที่ 50 LatLng สำหรับเก็บข้อมูลพิกัดละติจูดและลองจิจูด
 บรรทัดที่ 51 เพิ่มข้อมูลเข้าตัวแปร marker เพื่อนำไปแสดงในแผนที่

```

65 @override
66 void initState() {
67   super.initState();
68   attendService =
69   | | | AttendService(Dio(), baseUrl: context.read<AppData>().baseUrl);
70 }

```

ภาพประกอบที่ 3.35 สร้าง initState เพื่อสร้าง object ของ service

- บรรทัดที่ 66 สร้าง Method initState โดยต้องส่ง baseUrl (จาก provider) เข้าไปด้วย
 บรรทัดที่ 67-69 สร้าง initState เพื่อสร้าง object ของ service และ async method ที่จะใช้

```

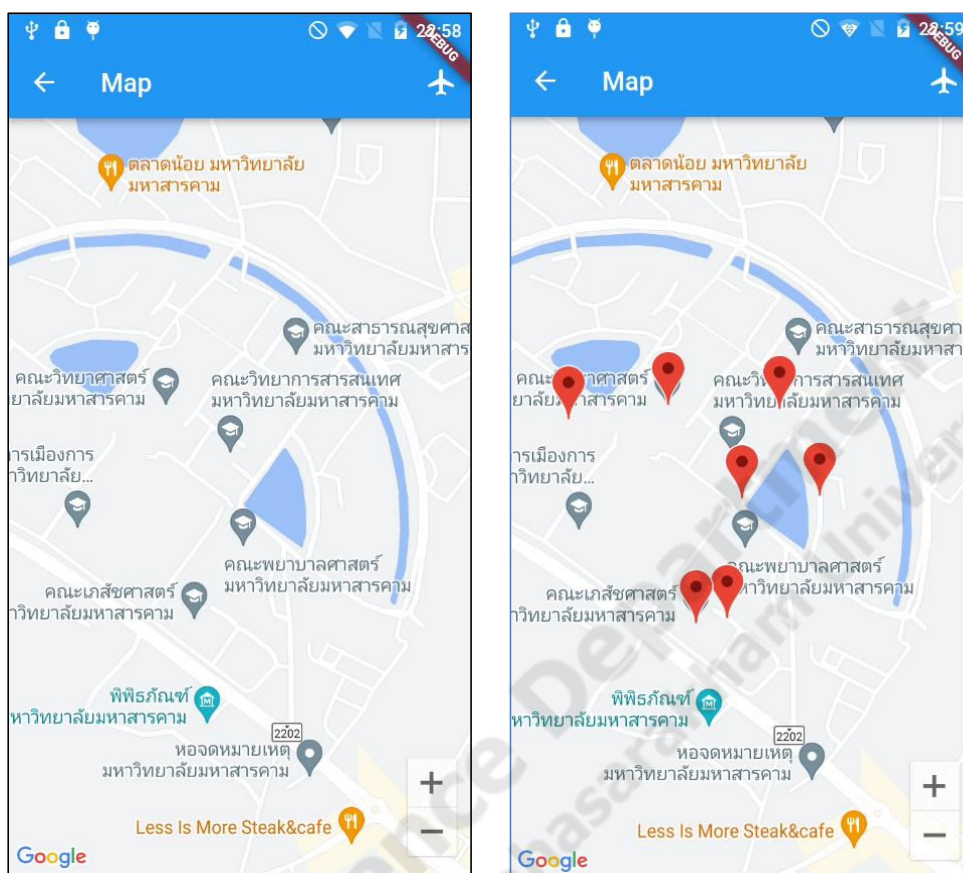
76  ▾  IconButton(
77      icon: const Icon(Icons.airplanemode_active),
78      onPressed: _goToMarker), // IconButton
79  ], // <Widget>[] // AppBar
80  ▾  body: GoogleMap(
81      markers: markers,
82      mapType: MapType.normal,
83  ▾  initialCameraPosition: CameraPosition(
84      target: centerMap,
85      zoom: 16,
86  ), // CameraPosition
87  ▾  onMapCreated: (GoogleMapController controller) {
88      _controller.complete(controller);
89  },
90  )); // GoogleMap // Scaffold
91  }
92  }

```

You, 2 weeks ago • updateLatLng ...

ภาพประกอบที่ 3.36 สร้าง Widget เพื่อแสดงผลแผนที่

- บรรทัดที่ 76-78 สร้างปุ่มเรียกใช้ Method `_goToMarker` เพื่อ mark จุดบนแผนที่
- บรรทัดที่ 80 เพิ่ม Widget ที่ชื่อว่า `GoogleMap` เพื่อแสดงแผนที่
- บรรทัดที่ 81 เรียกใช้ตัวแปร `markers` ที่เก็บข้อมูลละติจูดและลองจิจูด และแสดง `marker` ที่เป็นรูปหมุดหรือไอคอนต่างๆ
- บรรทัดที่ 82 แสดง `Type` ของแผนที่ ที่จะแสดงบนหน้าจอ
- บรรทัดที่ 83-85 `initialCameraPosition` คือการกำหนดพิกัดจุดเริ่มต้นสถานที่ที่จะให้ `googleMap` แสดงขึ้นมา
- บรรทัดที่ 87-88 การเรียกใช้ `controller`



ภาพประกอบที่ 3.37 ตัวอย่างการแสดงผลพิกัดและตำแหน่งบนแผนที่

3.7.3 Upload ไฟล์รูปภาพและวิดีโอ

```

22 class _uploadPageState extends State<UploadPage> {
23   PlatformFile? pickedFile;
24   UploadTask? uploadTask;
25   VideoPlayerController? videoPlayerController;
26   CustomVideoPlayerController? _customVideoPlayerController;
27   bool isImage = true;

```

ภาพประกอบที่ 3.38 ฟังก์ชันการเลือกไฟล์จากเครื่อง

- บรรทัดที่ 23 สร้าง Object เพื่อใช้สำหรับการเลือกไฟล์
- บรรทัดที่ 24 สร้าง Object เพื่อใช้ Upload ไฟล์ไปยัง Firebase
- บรรทัดที่ 25 สร้าง Object เพื่อใช้ฟังก์ชันการควบคุมวิดีโอ
- บรรทัดที่ 26 สร้าง Object เพื่อใช้ฟังก์ชันการแสดงผลวิดีโอ
- บรรทัดที่ 27 สร้างตัวแปรสำหรับเช็ค true false

```

54 Future selectFile() async {
55     final result = await FilePicker.platform.pickFiles();
56     if (result == null) return;
57     pickedFile = result.files.single;
58     log(pickedFile!.extension.toString());
59     if (pickedFile!.extension == 'jpg' || pickedFile!.extension == 'PNG') {
60         setState(() {
61             isImage = true;
62         });
63     } else {
64         isImage = false;
65     }
66 }

```

ภาพประกอบที่ 3.39 การเลือกไฟล์ โดยใช้ File picker

- บรรทัดที่ 54 สร้าง Future Method selectFile เพื่อรอเรียกใช้ในการเลือกไฟล์
 บรรทัดที่ 55-66 เป็นการฟังกั้นการเลือกไฟล์ในเครื่องและเช็คไฟล์ว่ามีนามสกุลไฟล์ตรงกับที่ต้องการหรือไม่

```

29 Future uploadFile() async {
30     final path = 'files/${pickedFile!.name}';
31     final file = File(pickedFile!.path!);
32     final ref = FirebaseStorage.instance.ref().child(path);
33
34     setState(() {
35         uploadTask = ref.putFile(file);
36     });
37     final snapshot = await uploadTask!.whenComplete(() {});
38
39     final urlDownload = await snapshot.ref.getDownloadURL();
40     log('Download Link:$urlDownload');
41     videoPlayerController = VideoPlayerController.network(urlDownload)
42     ..initialize().then((_) {
43         log(videoPlayerController.toString());
44         _customVideoPlayerController = CustomVideoPlayerController(
45             context: context,
46             videoPlayerController: videoPlayerController!,
47         );
48         Image.file(File(pickedFile!.path!));
49         setState(() {});
50     });
51 }

```

ภาพประกอบที่ 3.40 การอัปโหลดรูปภาพและวิดีโอผ่าน Firebase core และ Firebase storage

- บรรทัดที่ 29 สร้าง Future Method uploadFile เพื่อรอเรียกใช้ในการอัปโหลดไฟล์
 บรรทัดที่ 30 สร้างตัวแปร สำหรับ path อยู่ไฟล์ที่เราเลือก
 บรรทัดที่ 31 แปลง path เป็น file

- บรรทัดที่ 32 เชื่อมต่อกับ Firebase โดยใช้ Firebase core และ Firebase storage
 บรรทัดที่ 35 อัปโหลดไฟล์ขึ้นไปยัง Firebase
 บรรทัดที่ 37-39 เมื่ออัปโหลดไฟล์เสร็จจะได้ ลิงค์ไฟล์ที่อัปโหลดไปยัง Firebase
 บรรทัดที่ 41-48 ลิงค์ที่ได้ ใส่ไปในฟังก์ชันแสดง รูปภาพ และวิดีโอให้แสดงผลออกมา

```

96 | ElevatedButton(
97 |     onPressed: selectFile,
98 |     child: const Text('Select File'),
99 | ), // ElevatedButton
100 | const SizedBox(height: 2),
101 | buildProgress(),
102 | ElevatedButton(
103 |     onPressed: uploadFile,
104 |     child: const Text('Upload File'),
105 | ), // ElevatedButton

```

ภาพประกอบที่ 3.41 การสร้างปุ่มเพื่อเรียกใช้ method

- บรรทัดที่ 96-105 สร้างปุ่มขึ้นมาสองปุ่ม “Select File” เพื่อเรียกใช้ Method ที่สร้างขึ้นสำหรับเลือกไฟล์และปุ่ม “uploadFile” สำหรับอัปโหลดไฟล์และบันทึกลง Firebase

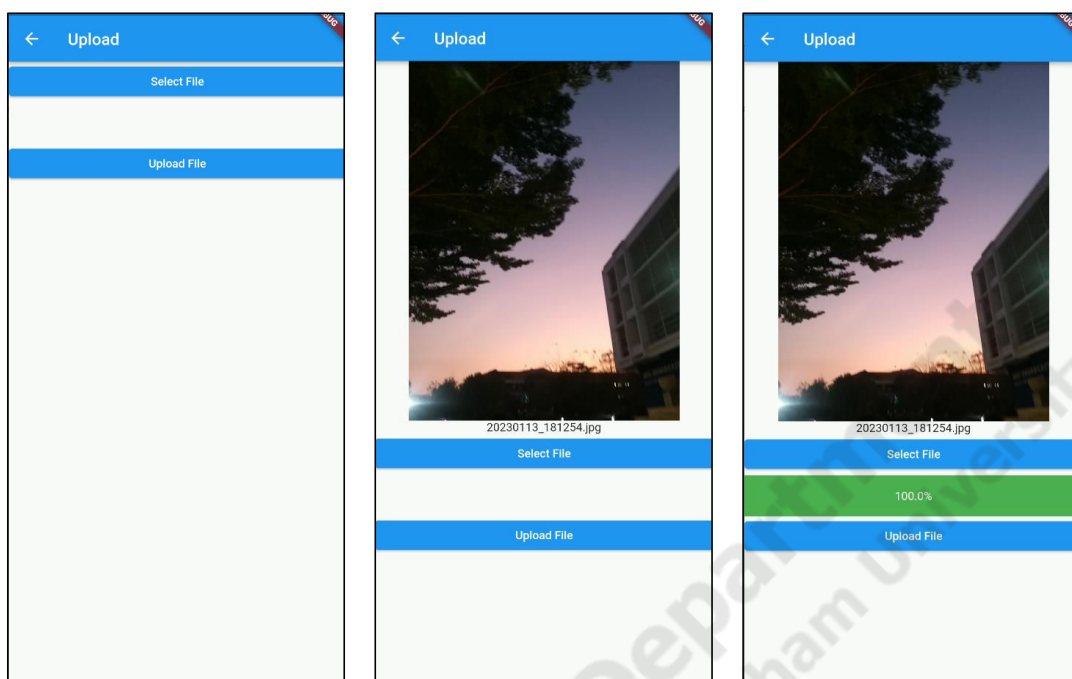
```

80 | Column(
81 |     mainAxisAlignment: MainAxisAlignment.center,
82 |     children: [
83 |         isImage == true
84 |         ? Image.file(
85 |             File(pickedFile!.path!),
86 |             width: size.width * 0.8,
87 |         ) // Image.file
88 |         : (_customVideoPlayerController != null)
89 |         ? CustomVideoPlayer(
90 |             customVideoPlayerController:
91 |                 _customVideoPlayerController!) // CustomVideoPlayer
92 |             : Container(),
93 |         Text(pickedFile!.name),
94 |     ],
95 | ), // Column

```

ภาพประกอบที่ 3.42 การแสดงผลของการเลือกและอัปโหลดของรูปภาพและวิดีโอ

- บรรทัดที่ 80-95 การแสดงผลและเช็คเงื่อนไขในการเลือกไฟล์ว่าเป็นรูปภาพหรือวิดีโอ เมื่อเลือกเสร็จจะแสดงผล



ภาพประกอบที่ 3.43 ตัวอย่างการอัปโหลดรูปภาพ

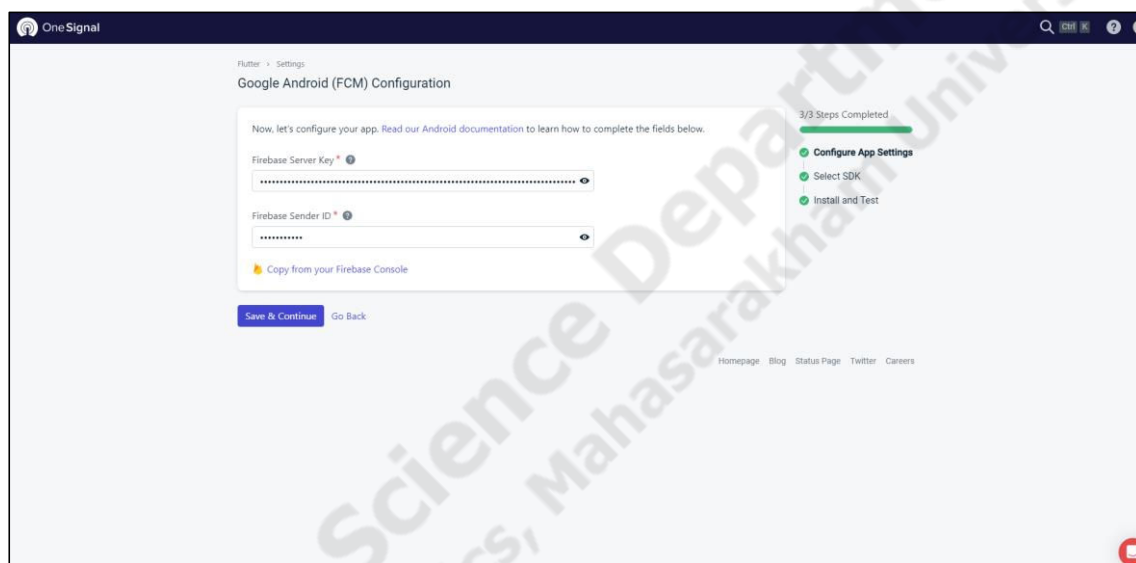


ภาพประกอบที่ 3.44 ตัวอย่างการอัปโหลดวิดีโอ

3.7.4 Push Notification

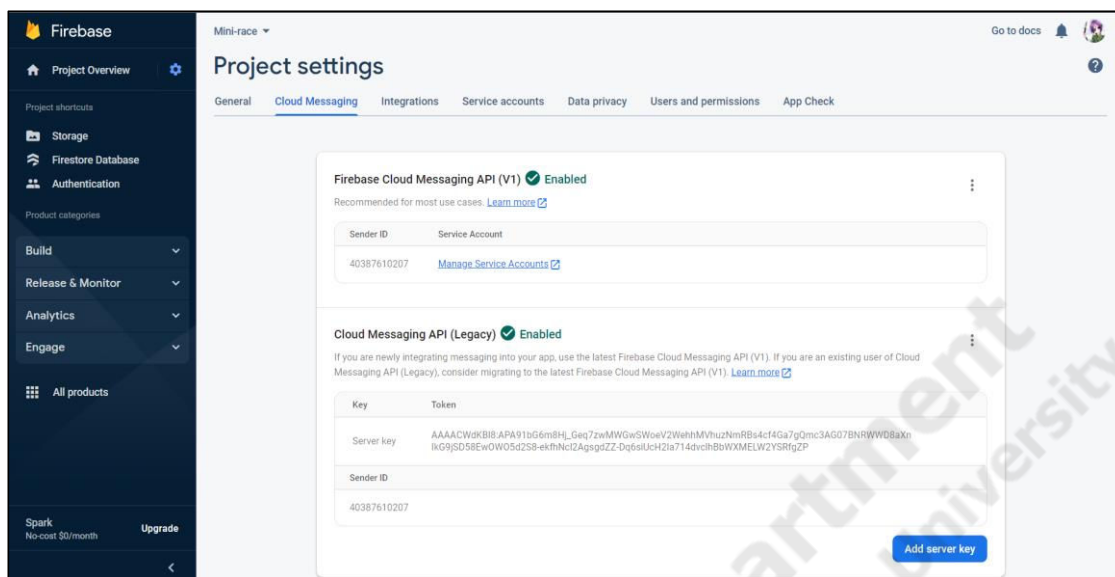
Notification เป็นหนึ่งในช่องทางของแอนดรอยด์ที่เปิดให้แอปพลิเคชันสามารถส่งข้อความได้ ผู้ใช้เห็นได้โดยไม่ต้องเปิดแอปขึ้นมา และผู้ใช้สามารถสั่งงานบางงานผ่าน Notification ตัวนั้นๆ

ได้อีกด้วย คือคำจำกัดความสั้นๆของ Notification ซึ่งเป็นหนึ่งในความสามารถของระบบแอนดรอยด์ที่เปิดให้นักพัฒนาระบบสามารถใช้งาน Notification ในเหตุการณ์ต่างๆที่เหมาะสมของแอปพลิเคชันตนเองได้ การทำงานจะทำงานอยู่บน Android Framework ซึ่งเป็นหัวใจหลักในการทำงานของระบบแอนดรอยด์ที่มีไว้ให้แอปต่างๆเรียกใช้งานซึ่งใน Android Framework ก็จะมีการแบ่งการทำงานแยกแตกต่างกันออกไปตามหน้าที่ และส่วนที่ทำหน้าที่เกี่ยวกับระบบ Notification จะมีชื่อว่า Notification Manager และ Status Bar ของแอนดรอยด์ที่ทำหน้าที่แสดง Notification จากแอปพลิเคชันต่าง ก็จะรับข้อมูลมาจาก Notification Manager ดังนั้นเมื่อใดที่แอปพลิเคชันในเครื่องส่ง Notification ไปให้ Notification Manager ก็จะถูกส่งไปเพื่อแสดงใน Notification Drawer ที่อยู่ใน Status Bar



ภาพประกอบที่ 3.45 การลือคอินผ่าน Onesignal เพื่อสร้างแอปหรือเว็บไซต์

OneSignal คือเครื่องมือที่ช่วยในการทำ Push Notification ได้อย่างง่าย โดยการทำงานของ OneSignal จะทำการสร้าง Key มาให้และมี API ให้เราสามารถส่งการจาก Service ได้ ซึ่งถ้าเราต้องการจะส่ง Notificaiton ไปหาทุกๆ User ใน Platform เราแค่ส่งไปบอก OneSignal แล้วก็ทำการกระจาย Notificaiton ให้เรา เมื่อทำการลือคอิน OneSignal แล้วจึงทำการสร้างแอปหรือเว็บไซต์ขึ้นมา จากนั้นจะมีการกำหนดค่าแอปโดยเราใช้ Firebase.google



ภาพประกอบที่ 3.46 การลือคอิน Firebase เพื่อรับ Server Key และ Sender ID

เมื่อเข้ามาที่ Firebase แล้วให้ทำการลือคอินและสร้างโปรเจ็คให้เรียบร้อย และเมื่อต้องการแก้ไขข้อมูลให้เลือกที่ปุ่มฟันเฟือง Setting จากนั้นให้เลือกไปที่ Cloud Messaging และทำการ Copy Server Key และ Sender ID นำไปใส่ที่ OneSignal ช่อง Firebase Server Key และ Firebase Sender ID



ภาพประกอบที่ 3.47 รับรหัสแอป OneSignal SDK

เมื่อทำการใส่ Firebase Server Key และ Firebase Sender ID เรียบร้อยแล้วเราจะได้รับรหัสแอปนอกจากนั้นสร้างและเรียกใช้แอป OneSignal SDK เมื่อตั้งค่าอย่างถูกต้องแล้ว จะจัดการสมัครรับการแจ้งเตือนอุปกรณ์โดยอัตโนมัติ

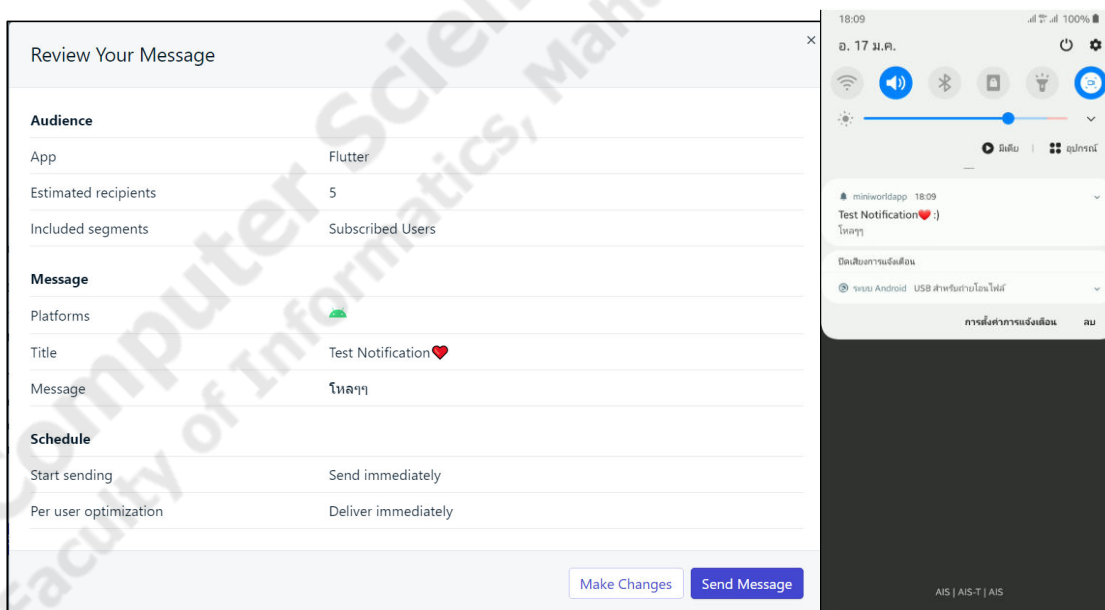

```

83 Widget build(BuildContext context) {
84   OneSignal.shared.setAppId("9670ea63-3a61-488a-afcf-8e1be833f631");
85
86   return Scaffold(
87     appBar: AppBar(
88       title: const Text('Nontification'),
89     ), // AppBar
90     body: Column(
91       children: [
92         ElevatedButton(
93           child: const Text("Send noti"),
94           onPressed: () => _handleSendNotification(), // ElevatedButton
95         ],
96       ), // Column
97     ); // Scaffold
98   }
99 }

```

ภาพประกอบที่ 3.48 เรียกใช้ OneSignal และกำหนดค่า OneSignal SDK

บรรทัดที่ 84 เรียกใช้ OneSignal และกำหนดค่า OneSignal SDK ที่ได้เป็นการเชื่อมแอปเราไปยัง OneSignal และ OneSignal จะเชื่อมไปยัง Firebase และ Firebase จะเชื่อมไปยังปลายทาง อื่นๆ



ภาพประกอบที่ 3.49 ส่งข้อความจาก OneSignal ไปที่ Service และส่งไปปลายทาง

จากนั้นทดลองส่งข้อความจาก OneSignal ไปที่ Service จากนั้นข้อมูลจะถูกส่งไปยังปลายทางอื่นๆที่ต้องการและทดสอบรันอีมีเลเตอร์เพื่อดูว่ามีข้อความการแจ้งเตือนถูกส่งมาหรือไม่

```

21 void _handleSendNotification() async {
22   var deviceState = await OneSignal.shared.getDeviceState();
23
24   if (deviceState == null || deviceState.userId == null) return;
25
26   var playerId = deviceState.userId;
27
28   var imageUrlString =
29     "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRNoy-7N8x4HgYJQuQC3i7Sw8nj9EawzrvhRw&usqp=CAU"
30
31   var notification1 = OSCreateNotification(
32     //playerID
33
34     playerIds: [
35       'b8742e68-2547-4cca-90a0-d1561a5654cc',
36     ],
37     content: "โหลๆ",
38     heading: "Test Notification ❤️ :)",
39     iosAttachments: {"id1": imageUrlString},
40     bigPicture: imageUrlString,
41     buttons: [
42       OSActionButton(text: "ตกลง", id: "id1"),
43       OSActionButton(text: "ยกเลิก", id: "id2")
44     ]); // OSCreateNotification
45   var response1 = await OneSignal.shared.postNotification(notification1);
46 }

```

ภาพประกอบที่ 3.50 การสร้าง Object ของ Notification

- บรรทัดที่ 22** ดึงค่าจาก OneSignal สถานะของเครื่องที่รันอยู่ปัจจุบัน
- บรรทัดที่ 24** ดึงค่า id จาก OneSignal ไม่ได้หรือมีค่าเป็น 0 จะแสดงค่า Null
- บรรทัดที่ 26** ถ้าดึงค่า id จาก OneSignal ได้จะนำมาเก็บใน player id เป็น id ของผู้รับและ deviceStatus คือ player id ของเครื่องนี้จะไม่สามารถส่งหาคนอื่นได้
- บรรทัดที่ 31-44** สร้าง Object ของ Notification และนำ id ของเครื่องปลายทางที่เราจะ ส่งใส่ที่ playerIds และข้อความที่จะส่งประกอบด้วย content และ heading คือหัวเรื่องและเนื้อเรื่อง
- บรรทัดที่ 45** เรียกใช้ OneSignal ให้ส่ง Notification แล้ว Notification จะส่ง Object ขึ้น Server และ Server จะทำการตรวจเช็คข้อความและเลข id จากนั้นจะส่งไปที่เครื่องปลายทาง เมื่อส่งเสร็จสามารถนำ response1 ไปเช็คได้ว่าส่งไปที่เครื่องปลายทางที่คน