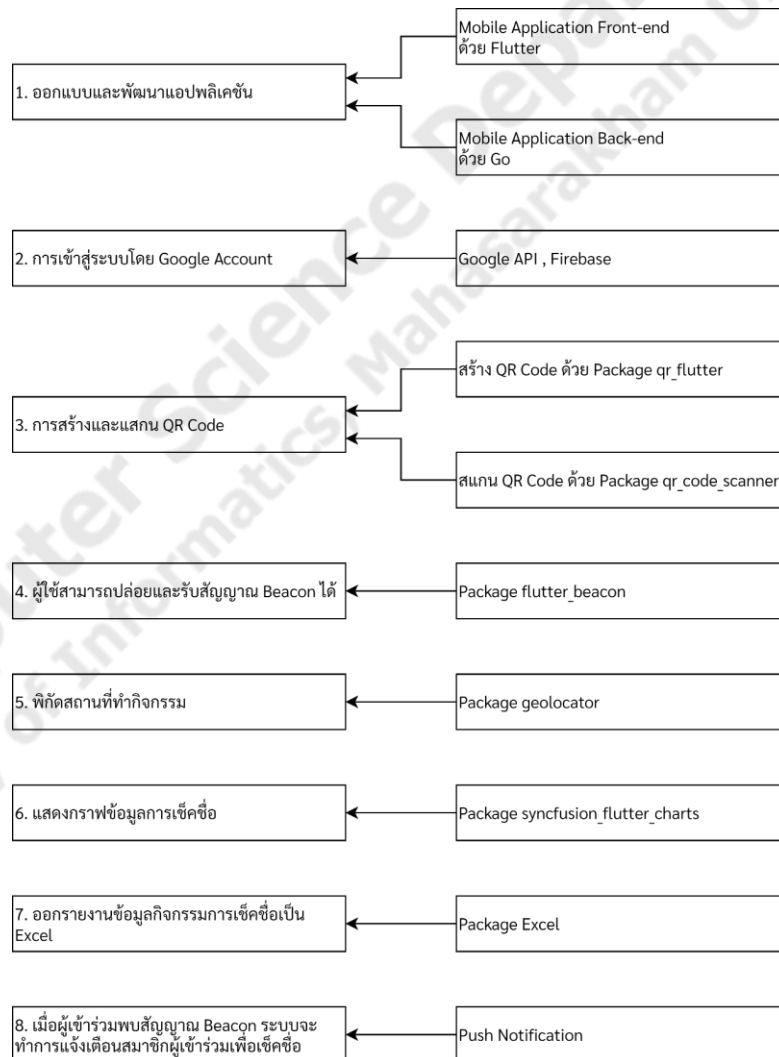


บทที่ 3

ขั้นตอนการดำเนินงาน

สำหรับในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงการปริญญาโท ซึ่งทำให้ทราบถึงการวิเคราะห์และการออกแบบระบบโดยละเอียดว่ามีแนวทางในการทำงานหรือมีขั้นตอนในการทำงานของระบบอย่างไรบ้างโดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้

3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 3.1 กรอบการพัฒนา

คำอธิบาย

1. เขียนโปรแกรมออกแบบและพัฒนาแอปพลิเคชัน จัดการการเข้าร่วมกิจกรรมโดยใช้ปีคอน สำหรับผู้ใช้งานผ่าน สมาร์ทโฟนโดยสร้างแอปพลิเคชันจะต้องมีการติดตั้งเครื่องมือดังนี้

- Visual Studio Code หรือ VS Code เป็นโปรแกรมประเภท Editor มีความสามารถในการแก้ไขและเขียนโค้ดเหมือนตัวอื่นๆ

- Flutter เป็น Framework ที่เขียนด้วยภาษา Dart ใช้ในการพัฒนาแอปพลิเคชัน ทางด้าน Front-end ถูกพัฒนาโดยบริษัท Google

- Go เป็นภาษาอยู่ในระดับ Low-level เช่นเดียวกับภาษา C ซึ่งในปัจจุบันนิยมใช้ใน การพัฒนาแอปพลิเคชันทางด้าน Back-end ถูกพัฒนาโดยบริษัท Google

2. การเข้าสู่ระบบด้วย Google Account ในส่วนนี้ใช้ Google API และ Firebase ซึ่ง เชื่อมต่อกันกับแอปพลิเคชัน และ บัญชี Google ในสมาร์ตโฟนเพื่อติดต่อเรียกใช้ข้อมูลทำให้แอปพลิเคชันสามารถใช้งานการเข้าสู่ระบบโดย Google Account ได้

3. การสร้างและสแกนคิวอาร์โค้ดซึ่งการสร้างคิวอาร์โค้ดจะมีการติดตั้ง Package qr_flutter และในส่วนของการสแกนคิวอาร์โค้ดจะติดตั้ง Package qr_code_scanner จากเว็บไซต์ pub.dev

4. การกระจายสัญญาณและรับสัญญาณ Beacon โดยการกระจายสัญญาณปีคอน และการสแกนปีคอนจะต้องติดตั้ง Package flutter_beacon และในการกระจายสัญญาณปีคอนจะต้องมีรหัส UUID ที่จะปล่อยจึงต้องติดตั้ง Package uuid จากเว็บไซต์ pub.dev

5. พิกัดสถานที่ทำกิจกรรม โดยการรับพิกัด GPS จะมีการติดตั้ง Package geolocator จาก เว็บไซต์ pub.dev เพื่อทำการรับค่าพิกัด หรือละติจูด ลองจิจูด ในตำแหน่งปัจจุบันของสมาร์ตโฟนได้

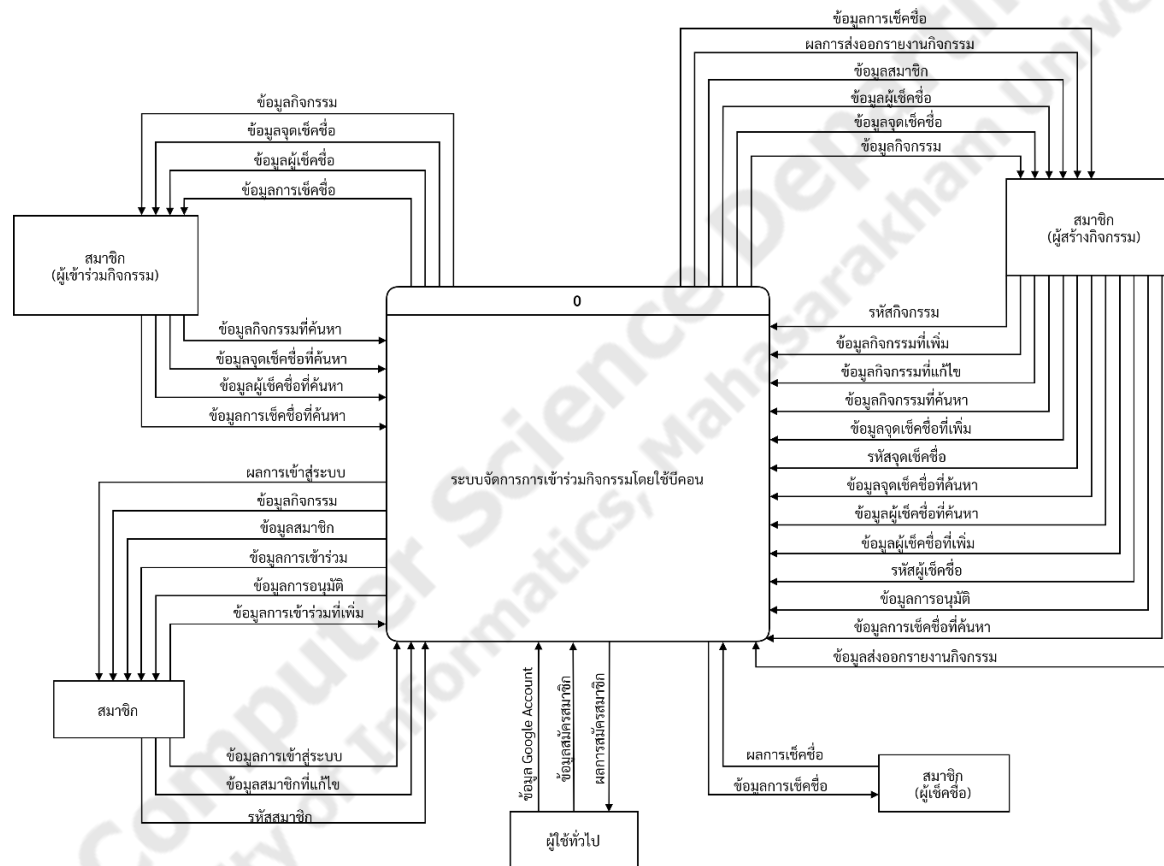
6. แสดงกราฟข้อมูลการเช็คชื่อ จะเป็นการแสดงข้อมูลจำนวนผู้ที่ทำการเช็คชื่อและจำนวนผู้ ที่ขาดการเช็คชื่อของการเข้าร่วมกิจกรรม ซึ่งจะต้องมีการติดตั้ง Package syncfusion_flutter_charts

7. ออกรายงานข้อมูลกิจกรรมการเช็คชื่อเป็นไฟล์ Excel โดยรูปแบบรายงานที่ออกไปจะเป็น ไฟล์ Excel โดยจะเป็นข้อมูลทั้งหมดของกิจกรรม และจะส่งออกไฟล์ไปทางอีเมล ซึ่งการสร้างไฟล์ Excel นั้นจะต้องมีการติดตั้ง Package syncfusion_flutter_xlsio จากเว็บไซต์ pub.dev

8. การแจ้งเตือนเมื่อสมาชิกผู้เข้าร่วมพบสัญญาณ Beacon โดยใช้ Push Notification

3.2 การออกแบบระบบ

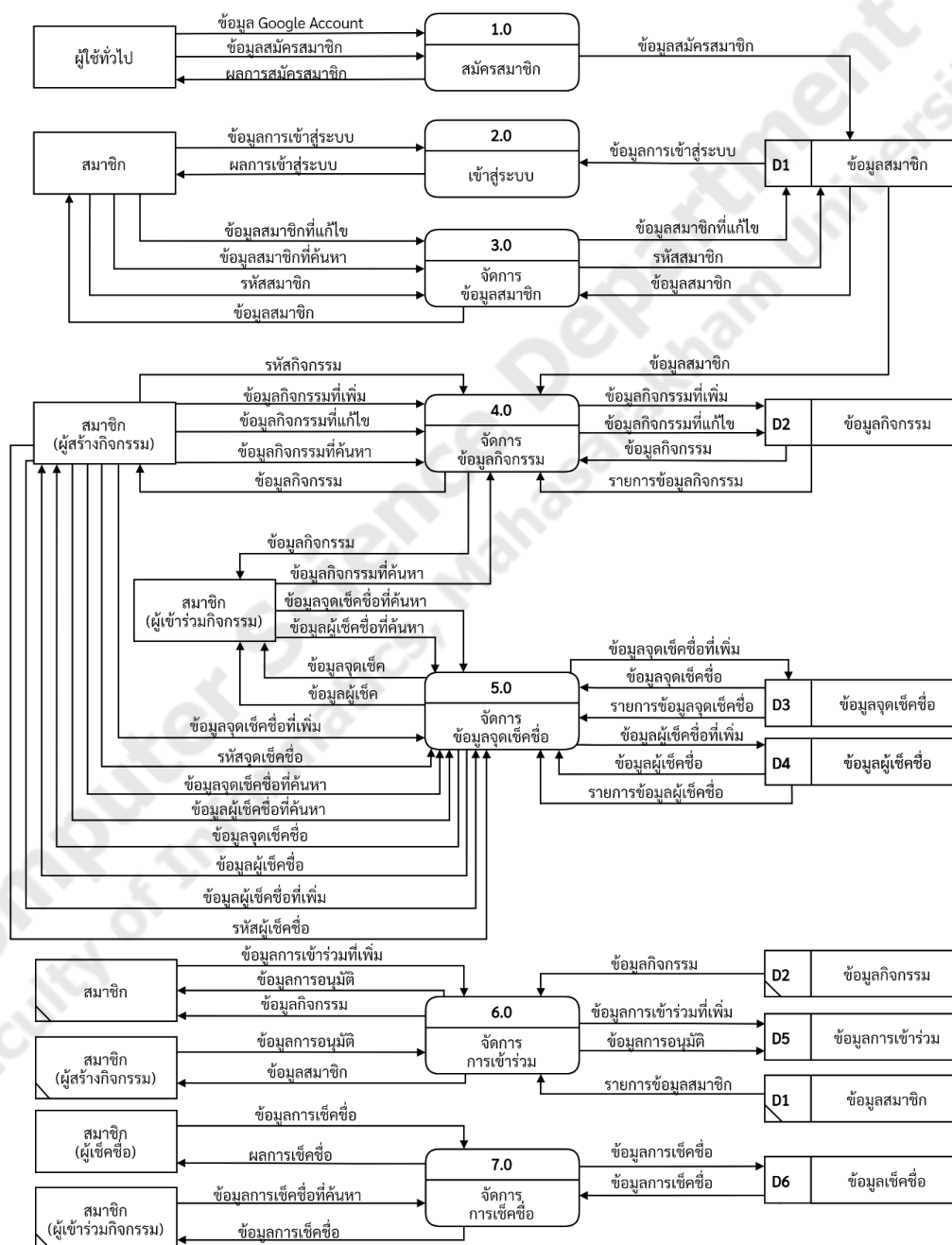
3.2.1 แผนภาพการไหลของข้อมูล (Context Diagram)



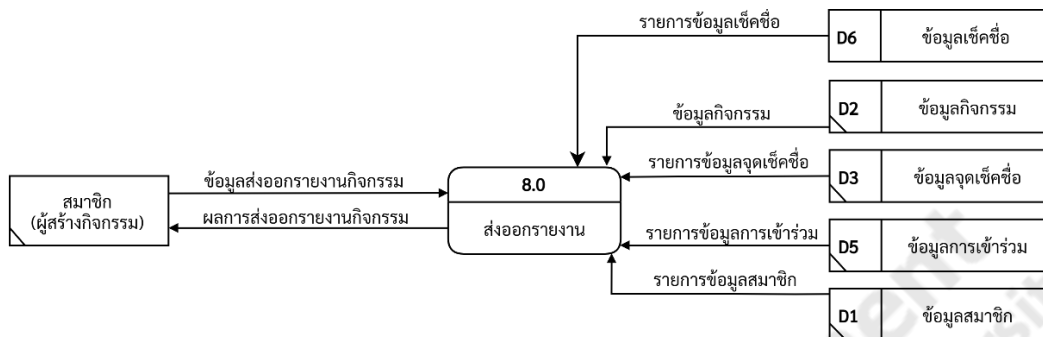
ภาพประกอบที่ 3.2 Context Diagram

3.2.2 Data Flow Diagram Level 1

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



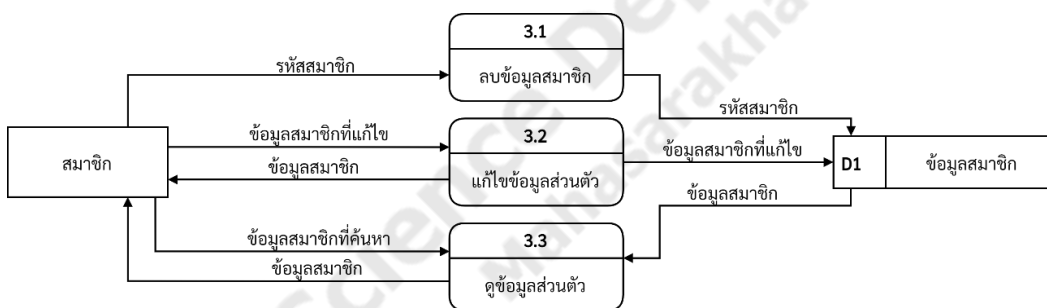
ภาพประกอบที่ 3.3 Data Flow Diagram Level 1



ภาพประกอบที่ 3.3 Data Flow Diagram Level 1 (ต่อ)

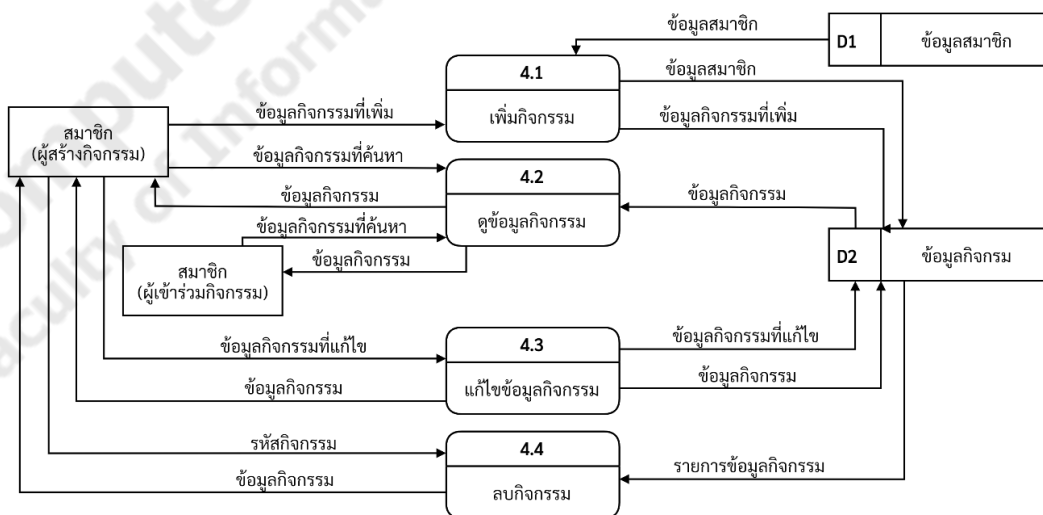
3.2.3 Data Flow Diagram Level 2

3.2.3.1 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 3



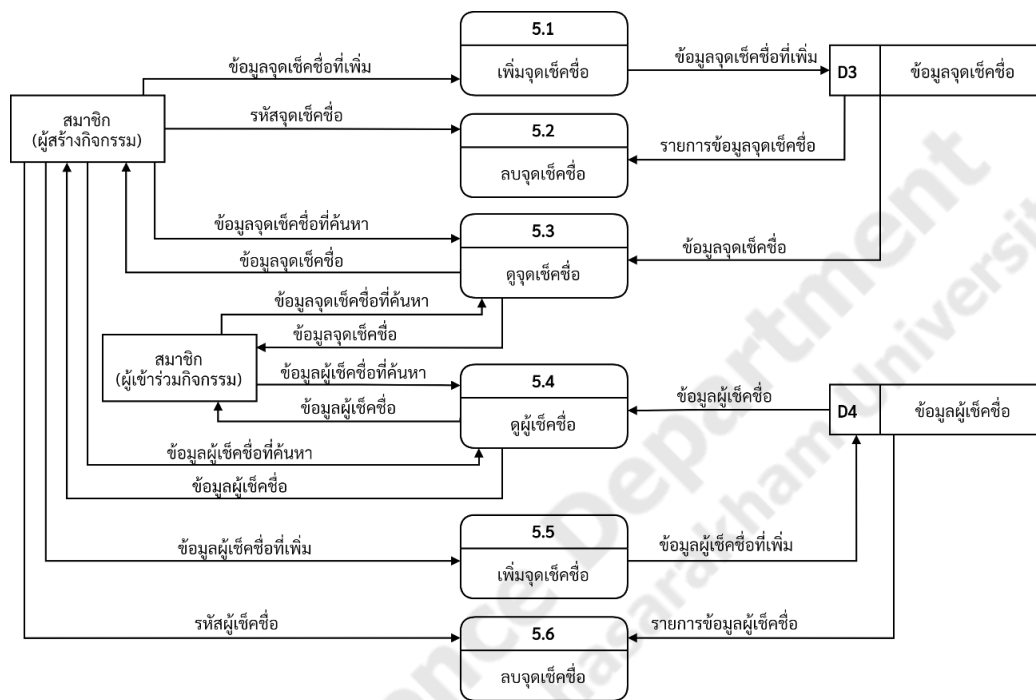
ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 Process 3

3.2.3.2 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 4



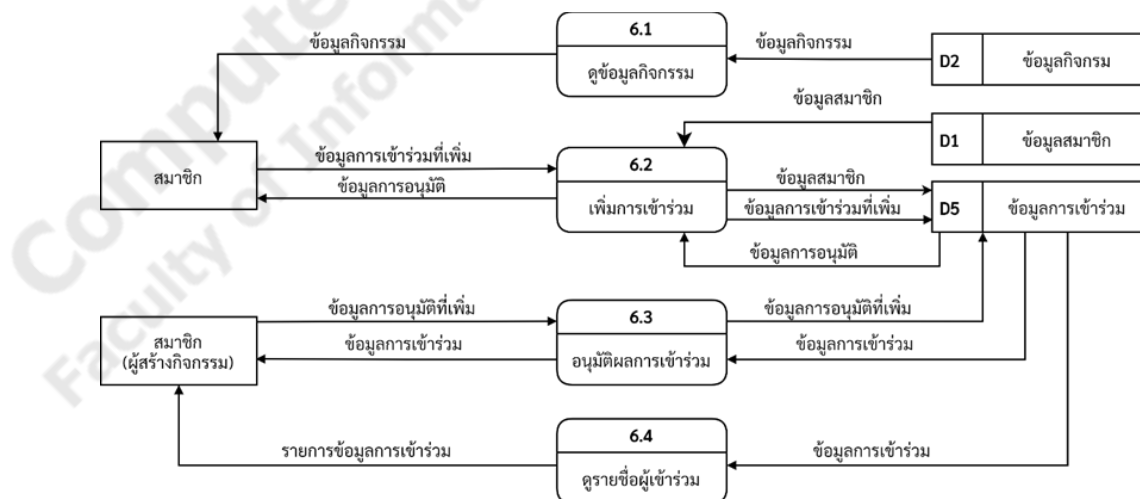
ภาพประกอบที่ 3.5 Data Flow Diagram Level 2 Process 4

3.2.3.3 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 5



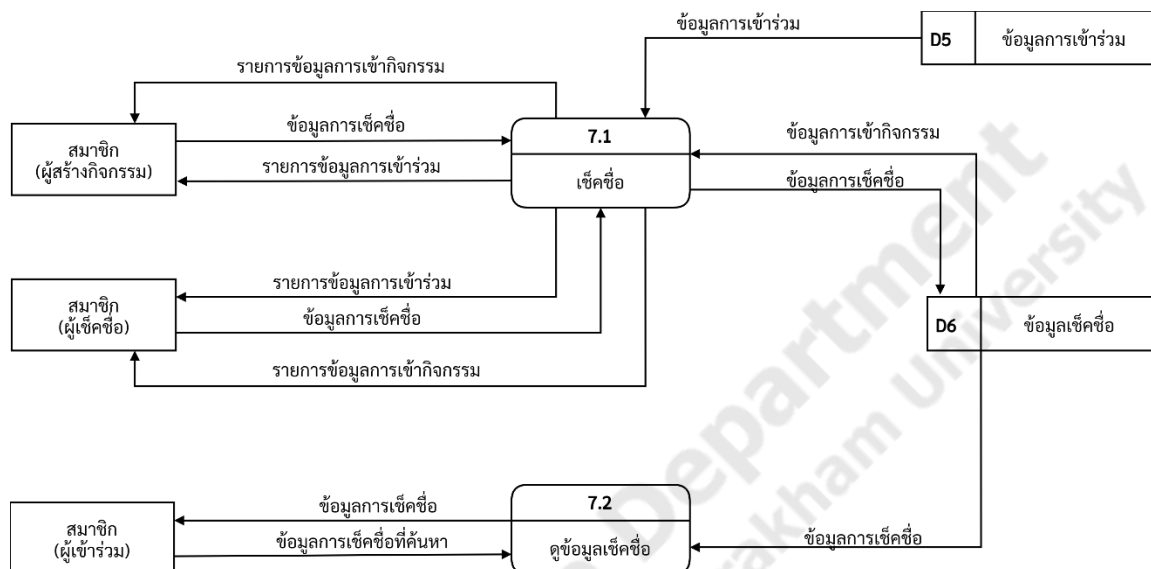
ภาพประกอบที่ 3.6 Data Flow Diagram Level 2 Process 5

3.2.3.4 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 6



ภาพประกอบที่ 3.7 Data Flow Diagram Level 2 Process 6

3.2.3.5 แผนภาพการไหลของข้อมูลระดับ 2 กระบวนการที่ 7



ภาพประกอบที่ 3.8 Data Flow Diagram Level 2 Process 7

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
ผู้ใช้ทั่วไป	บุคคล ที่ไม่ได้เป็น สมาชิกของ แอปพลิเคชัน	- ข้อมูลสมัครสมาชิก	ผู้ใช้ทั่วไป
สมาชิก	สมาชิก ทุกคนใน แอปพลิเคชัน	- ข้อมูลการเข้าสู่ระบบ - ข้อมูลสมาชิกที่แก้ไข - ข้อมูลสมาชิกที่ค้นหา - หมายเลขสมาชิก - ข้อมูลการเข้าร่วมที่เพิ่ม	- ผลการเข้าสู่ระบบ - ข้อมูลสมาชิก - ข้อมูลการอนุมัติ - ข้อมูลกิจกรรม

ตารางที่ 3.1 External Entity Description (ต่อ)

Name	Description	Input Data Flow	Output Data flow
สมาชิก (ผู้สร้าง กิจกรรม)	สมาชิกในแอป พลิเคชันที่เป็น ผู้สร้างกิจกรรม	<ul style="list-style-type: none"> - หมายเลขกิจกรรม - ข้อมูลกิจกรรมที่เพิ่ม - ข้อมูลกิจกรรมที่แก้ไข - ข้อมูลกิจกรรมที่ค้นหา - ข้อมูลจุดเช็คอินที่เพิ่ม - หมายเลขจุดเช็คอิน - ข้อมูลจุดเช็คอินที่ค้นหา - ข้อมูลผู้เช็คอินที่ค้นหา - ข้อมูลผู้เช็คอินที่เพิ่ม - หมายเลขผู้เช็คอิน - ข้อมูลการอนุมัติ - ข้อมูลส่งออกรายงานกิจกรรม 	<ul style="list-style-type: none"> - ข้อมูลกิจกรรม - ข้อมูลจุดเช็คอิน - ข้อมูลผู้เช็คอิน - ข้อมูลสมาชิก - ข้อมูลส่งออกรายงานกิจกรรม
สมาชิก (ผู้เข้าร่วม กิจกรรม)	สมาชิก ในแอปพลิเคชัน ที่เป็น ผู้เข้าร่วม กิจกรรม	<ul style="list-style-type: none"> - ข้อมูลกิจกรรมที่ค้นหา - ข้อมูลจุดเช็คอินที่ค้นหา - ข้อมูลผู้เช็คอินที่ค้นหา - ข้อมูลการเช็คอินที่ค้นหา 	<ul style="list-style-type: none"> - ข้อมูลกิจกรรม - ข้อมูลจุดเช็คอิน - ข้อมูลผู้เช็คอิน - ข้อมูลการเช็คอิน
สมาชิก (ผู้เช็คอิน)	สมาชิก ในแอปพลิเคชัน ที่เป็นผู้เข้าร่วม กิจกรรม ซึ่งถูก กำหนดสิทธิ์ ในการเช็คอิน สมาชิก ในกิจกรรม	<ul style="list-style-type: none"> - ข้อมูลการเช็คอินที่ค้นหา 	<ul style="list-style-type: none"> - ข้อมูลการเช็คอิน

3.3.2 Data Flow Description and Data Structure

เป็นขั้นตอนการทำงานของแอปพลิเคชันซึ่งทำให้เราทราบถึงการรับ-ส่งข้อมูลแสดงถึงการไหลของข้อมูลทั้งข้อมูลเข้า (Input) และข้อมูลส่งออก (Output) ระหว่างข้อมูลต้นทางถึงข้อมูลปลายทางโดยอธิบายข้อมูลและขั้นตอนในการดำเนินงานดังต่อไปนี้

ตารางที่ 3.2 Data Flow Description and Data Structure

Name	Description	Source	Destination	Data Structure
ข้อมูลสมัครสมาชิก	รายละเอียดการสมัครสมาชิกของสมาชิก	ผู้ใช้ทั่วไป	Process 1.0 สมัครสมาชิก	ชื่อจริง+นามสกุล+(รูปภาพ)+เบอร์โทรศัพท์+อีเมล+(รหัสประจำตัวนิสิตมหาวิทยาลัยมหาสารคาม)+รหัสผ่าน
		Process 1.0 สมัครสมาชิก	D1 ข้อมูลสมาชิก	
ข้อมูล Google Account	รายละเอียด Google Account ของสมาชิก	ผู้ใช้ทั่วไป	Process 1.0 สมัครสมาชิก	ชื่อจริง+นามสกุล+รูปภาพ+เบอร์โทรศัพท์+[อีเมล รหัสประจำตัวนิสิตมหาวิทยาลัยมหาสารคาม]+รหัสผ่าน
		Process 1.0 สมัครสมาชิก	D1 ข้อมูลสมาชิก	
ข้อมูลการเข้าสู่ระบบ	รายละเอียดการเข้าสู่ระบบของสมาชิก	สมาชิก	Process 2.0 เข้าสู่ระบบ	[อีเมล รหัสประจำตัวนิสิตของมหาวิทยาลัยมหาสารคาม]+รหัสผ่าน
		Process 2.0 เข้าสู่ระบบ	D1 ข้อมูลสมาชิก	
ข้อมูลสมาชิกที่แก้ไข	รายละเอียดการแก้ไขข้อมูลสมาชิก	สมาชิก	Process 3.0 จัดการข้อมูลสมาชิก	(ชื่อ)+(นามสกุล)+(รูปภาพ)+(เบอร์โทรศัพท์)+(รหัสผ่าน)

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 3.0 จัดการข้อมูล สมาชิก	D1 ข้อมูล สมาชิก	
ข้อมูลสมาชิก ที่ค้นหา	รายละเอียด การค้นหา ข้อมูลสมาชิก	สมาชิก	Process 3.0	(ชื่อจริง)+(นามสกุล)
		Process 3.0 จัดการข้อมูล สมาชิก	D1 ข้อมูล สมาชิก	
ข้อมูล กิจกรรมที่ เพิ่ม	รายละเอียด การเพิ่มข้อมูล กิจกรรม เพื่อ สร้างกิจกรรม	สมาชิก (ผู้สร้าง กิจกรรม)	Process 4.0 ข้อมูลกิจกรรม	ชื่อกิจกรรม+ รายละเอียด+(รูปภาพ)+วันที่ เริ่ม+วันที่สิ้นสุด+สถานะ
		Process 4.0 ข้อมูลกิจกรรม	D2 ข้อมูล กิจกรรม	
ข้อมูล กิจกรรมที่ แก้ไข	รายละเอียด การแก้ไข ข้อมูล กิจกรรม	สมาชิก (ผู้สร้าง กิจกรรม)	Process 4.0 ข้อมูลกิจกรรม	ชื่อกิจกรรม+ รายละเอียด+(รูปภาพ)+วันที่ เริ่ม+วันที่สิ้นสุด+สถานะ
		Process 4.0 ข้อมูลกิจกรรม	D2 ข้อมูล กิจกรรม	
ข้อมูล กิจกรรมที่ ค้นหา	รายละเอียด การค้นหา ข้อมูล กิจกรรม	สมาชิก (ผู้สร้าง กิจกรรม)สมาชิก (ผู้เข้าร่วม กิจกรรม)	Process 4.0 ข้อมูลกิจกรรม	(ชื่อกิจกรรม)
		Process 4.0 ข้อมูลกิจกรรม	D2 ข้อมูล กิจกรรม	
ข้อมูลจุดเช็ค ชื่อที่ค้นหา	รายละเอียด ข้อมูลของจุด เช็คชื่อที่ ต้องการ ค้นหา	สมาชิก (ผู้เข้าร่วม กิจกรรม)	Process 5.0 จัดการข้อมูลจุด เช็คชื่อ	(ชื่อจุดเช็คชื่อ)

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 5.0 จัดการข้อมูลจุด เช็คชื่อ	D3 ข้อมูลจุดเช็ค ชื่อ D4 ข้อมูลผู้เช็คชื่อ	
ข้อมูลผู้เช็คชื่อ ที่ค้นหา	รายละเอียด ข้อมูลของผู้ เช็คชื่อที่ ต้องการค้นหา	สมาชิก (ผู้เข้าร่วม กิจกรรม)	Process 5.0 จัดการข้อมูลจุด เช็คชื่อ	(ชื่อจริง)+(นามสกุล)+ (เบอร์โทรศัพท์)
		Process 5.0 จัดการข้อมูลจุด เช็คชื่อ	D3 ข้อมูลจุดเช็ค ชื่อ D4 ข้อมูลผู้เช็คชื่อ	
ข้อมูลกิจกรรม ที่ค้นหา	รายละเอียด การค้นหา ข้อมูลกิจกรรม	สมาชิก	Process 6.0 จัดการการเข้าร่วม	(ชื่อกิจกรรม)
		Process 6.0 จัดการการเข้า ร่วม	D2 ข้อมูลกิจกรรม D5 ข้อมูลการเข้า ร่วม D6 ข้อมูลสมาชิก	
ข้อมูลการเข้า ร่วมที่เพิ่ม	รายละเอียด การเข้าร่วม กิจกรรม	สมาชิก	Process 6.0 จัดการการเข้าร่วม	(ชื่อ)+(นามสกุล)+ (รูปภาพ)+(เบอร์ โทรศัพท์)
		Process 6.0 จัดการการเข้า ร่วม	D2 ข้อมูลกิจกรรม D5 ข้อมูลการเข้า ร่วม D6 ข้อมูลสมาชิก	
ข้อมูลการ อนุมัติ	รายละเอียด การอนุมัติการ เข้าร่วม กิจกรรม	สมาชิก(ผู้สร้าง กิจกรรม)	Process 6.0 จัดการการเข้าร่วม	(ชื่อ)+(นามสกุล)+ (รูปภาพ)+(เบอร์ โทรศัพท์)

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 6.0 จัดการการ เข้าร่วม	D2 ข้อมูลกิจกรรม D5 ข้อมูลการเข้าร่วม D6 ข้อมูลสมาชิก	
ข้อมูลการ เช็คชื่อ	ข้อมูลการเช็ค ชื่อ	สมาชิก (ผู้เช็คชื่อ)	Process 7.0 จัดการ การเช็คชื่อ	(ชื่อ)+(นามสกุล)+(รูปภาพ) +(เบอร์โทรศัพท์)+(เวลาที่เข้า ร่วม)+(ตำแหน่งที่เช็คชื่อ)
		Process 7.0 จัดการการ เช็คชื่อ	D2 ข้อมูลกิจกรรม D3 ข้อมูลจุดเช็คชื่อ D5 ข้อมูลการเข้าร่วม D6 ข้อมูลเช็คชื่อ	
ข้อมูลการ เช็คชื่อที่ ค้นหา	รายละเอียด ข้อมูลของ การเช็คชื่อที่ ต้องการ ค้นหา	สมาชิก (ผู้เข้าร่วม กิจกรรม)	Process 7.0 จัดการการเช็คชื่อ	(ชื่อจุดเช็คชื่อ)
		Process 7.0 จัดการการ เช็คชื่อ	D2 ข้อมูลกิจกรรม D3 ข้อมูลจุดเช็คชื่อ D5 ข้อมูลการเข้าร่วม D6 ข้อมูลเช็คชื่อ	
ข้อมูล ส่งออก รายงาน กิจกรรม	ข้อมูลส่งออก รายงาน กิจกรรม	สมาชิก (ผู้สร้าง กิจกรรม)	Process 8.0 จัดการการเช็คชื่อ	(ชื่อ)+(นามสกุล)+ รายงานกิจกรรม
		Process 8.0 จัดการการ เช็คชื่อ	D2 ข้อมูลกิจกรรม D3 ข้อมูลจุดเช็คชื่อ D5 ข้อมูลการเข้าร่วม D1 ข้อมูลสมาชิก	

3.3.3 Data Store Description and Data Structure

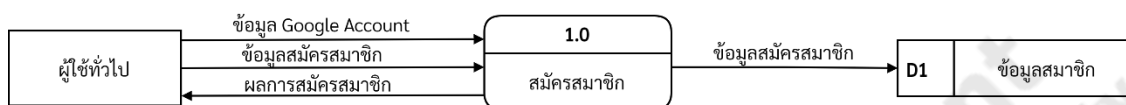
คือการนำเสนอข้อมูลเข้าไปจัดการในฐานข้อมูล โดยมีการแยกออกเป็นแฟ้มข้อมูล

ตารางที่ 3.3 Data Store Description and Data Structure

ID	Data Store	Description	Data Structure
D1	ข้อมูลสมาชิก	เก็บรายละเอียดของสมาชิกผู้ใช้งานระบบ	หมายเลขสมาชิก+อีเมล+รหัสผ่าน+เบอร์โทรศัพท์+คณะ+สาขา+มหาวิทยาลัย+หมายเลขGoogle+รูปภาพ+ชื่อจริง+นามสกุล+ชื่อเล่น+หมายเลขนักศึกษา
D2	ข้อมูลกิจกรรม	เก็บรายละเอียดข้อมูลของกิจกรรม	หมายเลขกิจกรรม+ชื่อกิจกรรม+เวลาเริ่ม+เวลาสิ้นสุด+รายละเอียดกิจกรรม+ความสัมพันธ์ส่วนตัว+รูปกิจกรรม+รหัสเข้ากิจกรรม
D3	ข้อมูลจุดเช็คอินชื่อ	เก็บรายละเอียดข้อมูลของจุดเช็คอินชื่อ	หมายเลขจุดเช็คอินชื่อ+เวลาเริ่ม+เวลาสิ้นสุด+ชื่อจุดเช็คอินชื่อ+วันที่เช็คอินชื่อ+ตำแหน่ง+UUID
D4	ข้อมูลผู้เช็คอินชื่อ	เก็บรายละเอียดข้อมูลของสมาชิกผู้เช็คอินชื่อ	หมายเลขสมาชิก
D5	ข้อมูลการเข้าร่วม	เก็บรายละเอียดข้อมูลของการเข้าร่วมกิจกรรม	หมายเลขการเข้าร่วม+เวลาที่เข้าร่วม
D6	ข้อมูลเช็คอินชื่อ	เก็บรายละเอียดข้อมูลของการเช็คอินชื่อ	หมายเลขสมาชิก

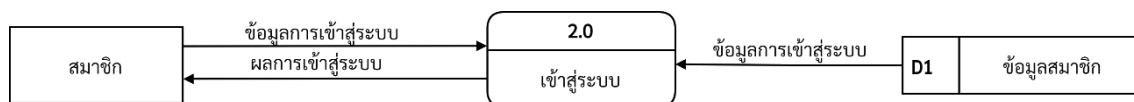
3.3.4 Process Description

3.3.4.1 Process 1.0 สมัครสมาชิก



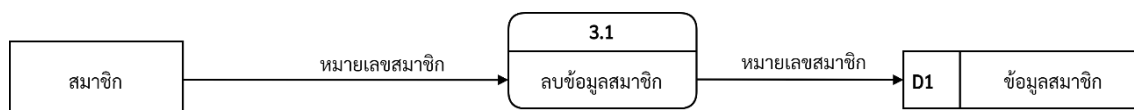
ID	1.0
Name	สมัครสมาชิก
Description	เพื่อทำการเพิ่มสมาชิกเข้าสู่ฐานข้อมูล
Input Data Flows	- ข้อมูลการสมัครสมาชิก - ข้อมูล Google Account
Output Data Flows	- ผลการสมัครสมาชิก
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลที่ต้องการสมัครสมาชิกและข้อมูล Google Account 2. ตรวจสอบว่าข้อมูลครบถ้วนและถูกต้องหรือไม่ ถ้าถูกต้องและครบถ้วน บันทึกข้อมูลลงฐานข้อมูล ถ้าไม่ถูกต้องและครบถ้วน ทำการแจ้งเตือนว่าข้อมูลผิดพลาด ไม่สามารถบันทึกข้อมูลได้ 3. แสดงข้อมูลสมัครสมาชิกของผู้ใช้ <p>จบการทำงาน</p>

3.3.4.2 Process 2.0 เข้าสู่ระบบ



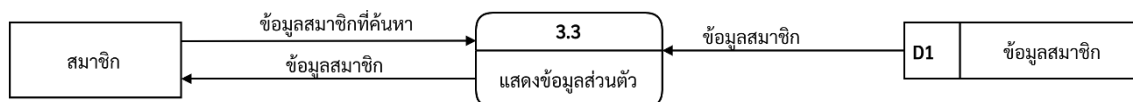
ID	2.0
Name	เข้าสู่ระบบ
Description	การเข้าสู่ระบบ
Input Data Flow	- ข้อมูลการเข้าสู่ระบบ
Output Data Flow	- ผลการเข้าสู่ระบบ
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลการเข้าสู่ระบบของผู้ใช้ 2. ตรวจสอบรายละเอียดข้อมูลผู้ใช้ <p>ถ้ามีข้อมูลผู้ใช้ในข้อมูลสมาชิก รายงานผลการเข้าสู่ระบบ</p> <p>ถ้าไม่มีข้อมูลผู้ใช้ในข้อมูลสมาชิก ทำการแจ้งเตือนว่าข้อมูลผิดพลาด ไม่สามารถเข้าสู่ระบบได้</p> <p>จบการทำงาน</p>

3.3.4.3 Process 3.1 ลบข้อมูลสมาชิก



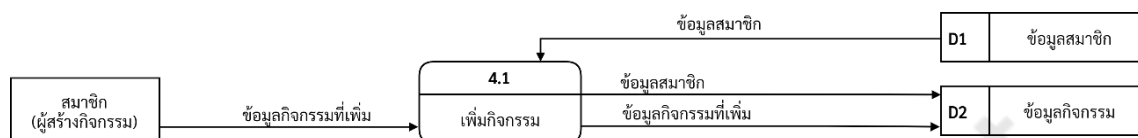
ID	3.1
Name	ลบข้อมูลสมาชิก
Description	การลบข้อมูลสมาชิก
Input Data Flow	- หมายเลขสมาชิก
Output Data Flow	-
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับหมายเลขสมาชิกของผู้ใช้ 2. ทำการลบข้อมูลสมาชิก แล้วระบบแจ้งเตือนว่า “ทำการลบข้อมูลเสร็จสิ้น” <p>จบการทำงาน</p>

3.3.4.4 Process 3.3 แสดงข้อมูลส่วนตัว



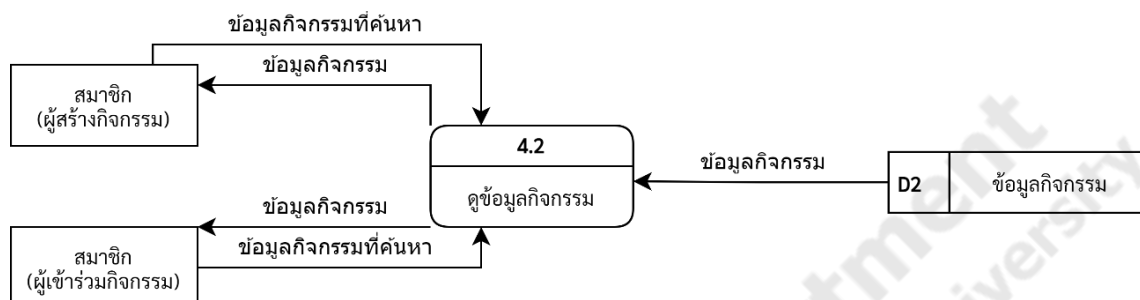
ID	3.3
Name	แสดงข้อมูลส่วนตัว
Description	การแสดงผลของข้อมูลของสมาชิกในระบบ
Input Data Flow	- ข้อมูลสมาชิกที่ค้นหา
Output Data Flow	- ข้อมูลสมาชิก
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลสมาชิกที่ต้องการค้นหาในข้อมูลสมาชิก 2. ตรวจสอบข้อมูลสมาชิกที่ต้องการค้นหา <p>ถ้ามีข้อมูล แสดงข้อมูลสมาชิกที่ต้องการค้นหา</p> <p>ถ้าไม่มีข้อมูล ทำการแจ้งเตือนว่า “ไม่พบข้อมูลในระบบ”</p> <p>จบการทำงาน</p>

3.3.4.5 Process 4.1 เพิ่มกิจกรรม



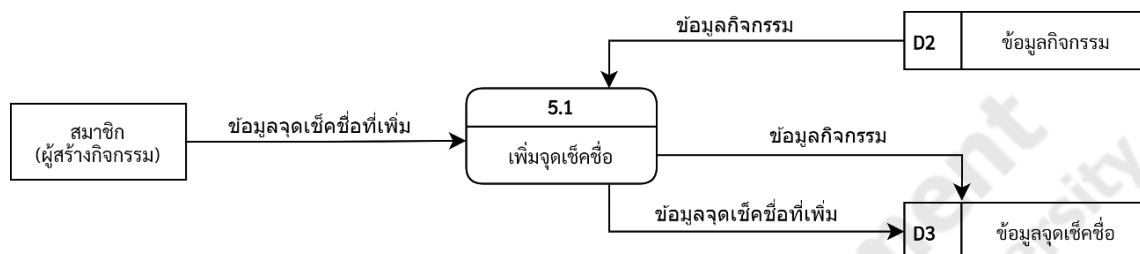
ID	4.1
Name	เพิ่มกิจกรรมในข้อมูลกิจกรรม
Description	การเพิ่มกิจกรรมในข้อมูลกิจกรรม
Input Data Flow	- ข้อมูลกิจกรรมที่เพิ่ม
Output Data Flow	- ข้อมูลสมาชิก
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลกิจกรรมที่ต้องการเพิ่ม 2. ตรวจสอบข้อมูลว่าครบถ้วนหรือไม่ <p>ถ้าข้อมูลครบถ้วน เพิ่มข้อมูลกิจกรรมเข้าไปในข้อมูลกิจกรรม</p> <p>ถ้าข้อมูลไม่ครบถ้วน ทำการแจ้งเตือนว่า “กรุณาป้อนข้อมูลให้ครบ”</p> <p>จบการทำงาน</p>

3.3.4.6 Process 4.2 ดูข้อมูลกิจกรรม



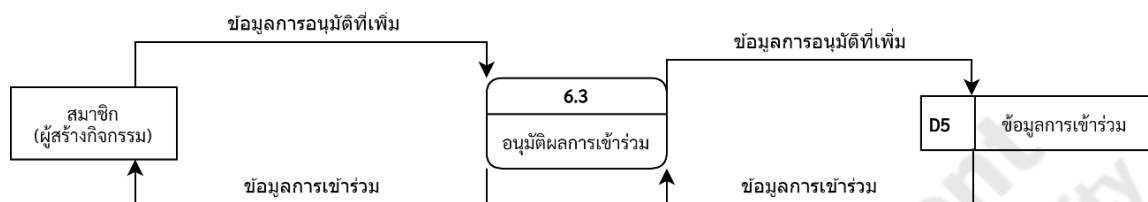
ID	4.2
Name	ดูข้อมูลกิจกรรม
Description	การดูรายละเอียดข้อมูลกิจกรรมในระบบ
Input Data Flow	- ข้อมูลกิจกรรมที่ค้นหา
Output Data Flow	- ข้อมูลกิจกรรม
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลกิจกรรมที่ต้องการค้นหา 2. ตรวจสอบข้อมูลว่ามีข้อมูลของกิจกรรมหรือไม่ <p>ถ้าพบข้อมูล ระบบจะแสดงข้อมูลของกิจกรรมนั้น</p> <p>ถ้าไม่พบข้อมูล ทำการแจ้งเตือนว่า “ไม่มีข้อมูลกิจกรรมในระบบ”</p> <p>จบการทำงาน</p>

3.3.4.7 Process 5.1 เพิ่มจุดเช็คชื่อ



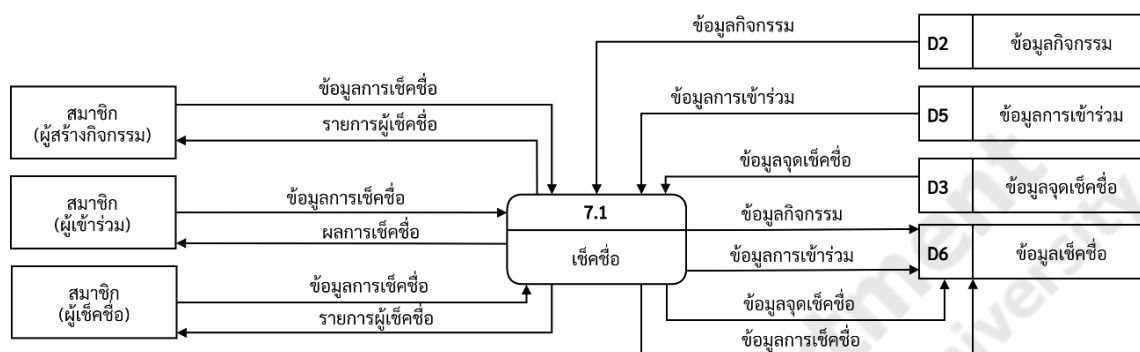
ID	5.1
Name	เพิ่มจุดเช็คชื่อ
Description	การเพิ่มรายละเอียดจุดเช็คชื่อ
Input Data Flow	- ข้อมูลจุดเช็คชื่อที่เพิ่ม
Output Data Flow	- ข้อมูลกิจกรรม
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลจุดเช็คชื่อที่ต้องการเพิ่ม 2. ตรวจสอบข้อมูลว่าข้อมูลจุดเช็คชื่อถูกต้องหรือไม่ ถ้าถูกต้อง ระบบทำการเพิ่มจุดเช็คชื่อในข้อมูลจุดเช็คชื่อ ถ้าไม่ถูกต้อง ทำการแจ้งเตือนว่า “ข้อมูลไม่ถูกต้อง โปรดลองใหม่อีกครั้ง” <p>จบการทำงาน</p>

3.3.4.8 Process 6.3 อนุมัติผลการเข้าร่วม



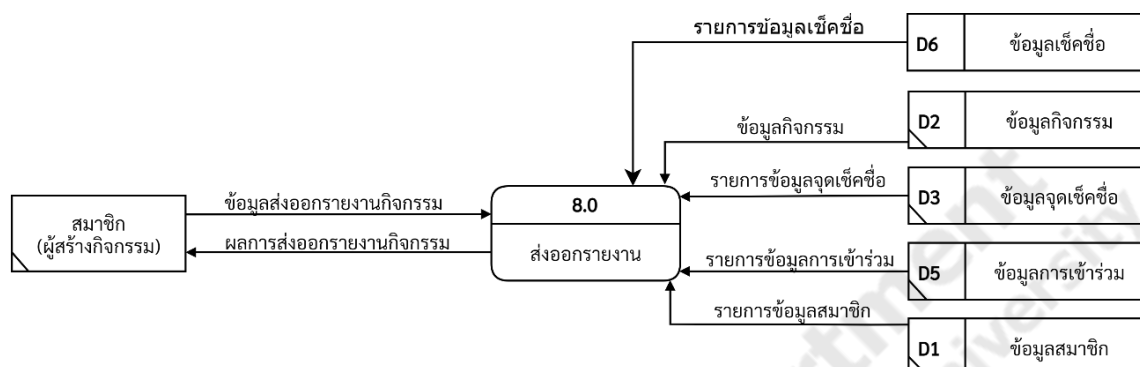
ID	6.3
Name	อนุมัติผลการเข้าร่วม
Description	การอนุมัติผลการเข้าร่วมกิจกรรม
Input Data Flow	- ข้อมูลการอนุมัติที่เพิ่ม
Output Data Flow	- ข้อมูลการเข้าร่วม
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลการอนุมัติที่ต้องการเพิ่ม 2. ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ <p>ถ้าถูกต้อง</p> <p>ระบบทำอนุมัติให้สมาชิกเข้าร่วมกิจกรรม</p> <p>ถ้าไม่ถูกต้อง</p> <p>ทำการแจ้งเตือนว่า “ไม่อนุมัติให้เข้าร่วมกิจกรรม”</p> <p>จบการทำงาน</p>

3.3.4.9 Process 7.1 เช็คชื่อ



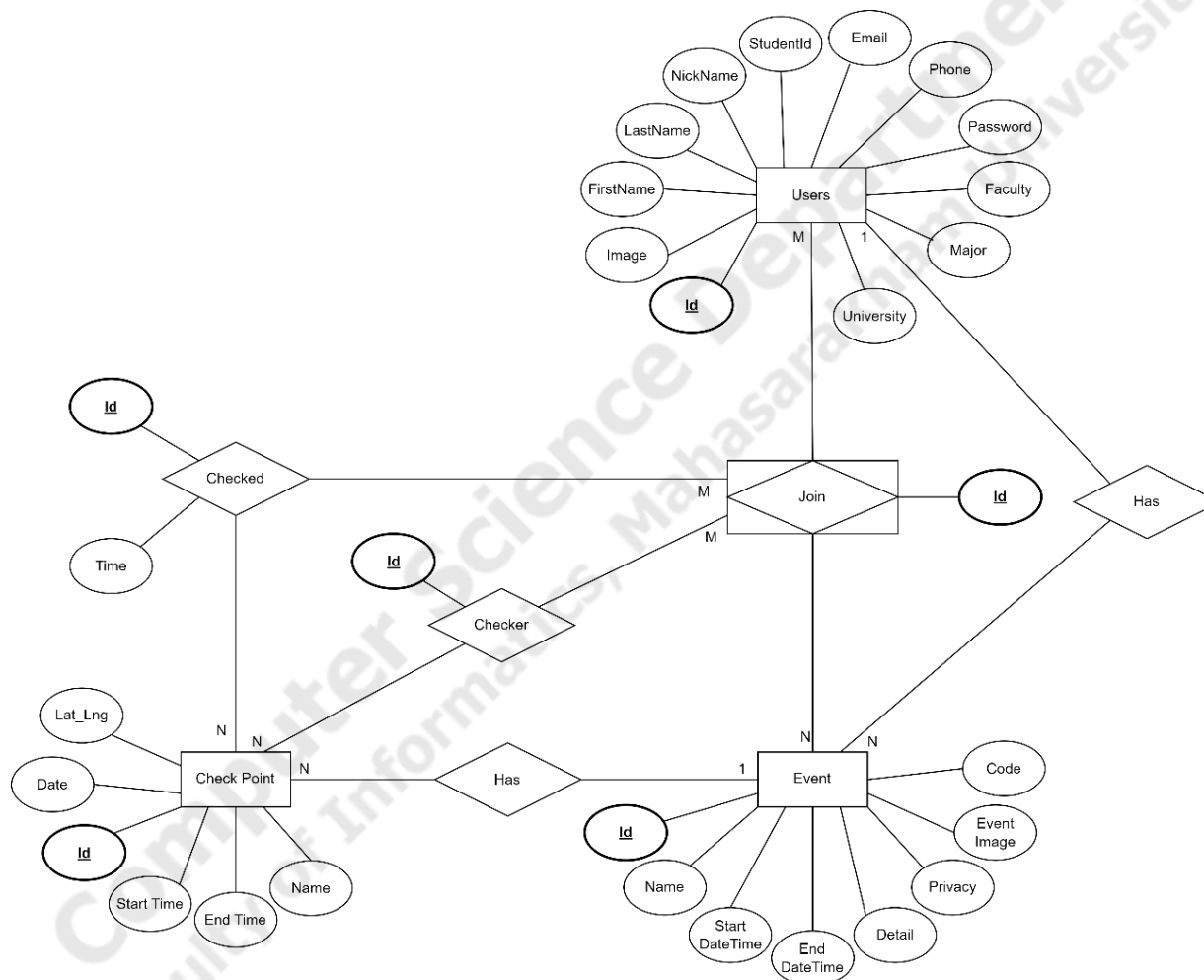
ID	7.1
Name	เช็คชื่อ
Description	การเช็คชื่อในกิจกรรม
Input Data Flow	- ข้อมูลการเช็คชื่อ
Output Data Flow	- รายการข้อมูลการเข้าร่วม - ข้อมูลกิจกรรม - ข้อมูลจุดเช็คชื่อ
Process Description	เริ่มต้น <ol style="list-style-type: none"> 1. รับข้อมูลการเช็คชื่อ 2. ตรวจสอบว่าข้อมูลครบถ้วนหรือไม่ ถ้าข้อมูลครบถ้วน ระบบทำการเช็คชื่อเข้าในระบบ ถ้าไม่ครบถ้วน ทำการแจ้งเตือนว่า “ทำการเช็คชื่อใหม่อีกครั้ง” จบการทำงาน

3.3.4.10 Process 8.0 ส่งออกรายงาน



ID	8.0
Name	ส่งออกรายงาน
Description	การส่งออกรายงาน
Input Data Flow	- ข้อมูลส่งออกรายงานกิจกรรม
Output Data Flow	- รายการข้อมูลเช็คชื่อ - ข้อมูลกิจกรรม - รายการข้อมูลจุดเช็คชื่อ - รายการข้อมูลการเข้าร่วม - รายการข้อมูลสมาชิก
Process Description	เริ่มต้น <ol style="list-style-type: none"> 1. รับข้อมูลส่งออกรายงาน 2. ตรวจสอบว่าข้อมูลครบถ้วนหรือไม่ ถ้าข้อมูลครบถ้วน ระบบทำการส่งออกรายงาน ถ้าไม่ครบถ้วน ทำการแจ้งเตือนว่า “ไม่สามารถส่งออกรายงานได้” จบการทำงาน

3.4 ความสัมพันธ์ (Entity Relationship Diagram)



ภาพประกอบที่ 3.9 Entity Relationship Diagram

3.5 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ข้อมูลสมาชิก (Users)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
Id	รหัสผู้ใช้	int		PK		7
FirstName	ชื่อจริง	varchar	50			Borikarn
LastName	นามสกุลจริง	varchar	50			Kunnapab
NickName	ชื่อเล่น	varchar	50			Bin
Email	อีเมล	varchar	70			Borikarn.bin@msu.ac.th
Password	รหัสผ่าน	varchar	255			B098765431
Phone	หมายเลขโทรศัพท์	varchar	15			0996351142
Image	รูปประจำตัว	varchar	500			-
University	มหาวิทยาลัย	varchar	128			มหาวิทยาลัยมหาสารคาม
StudentId	รหัสนิสิต	varchar	15			63011212041
Faculty	คณะ	varchar	50			วิทยาการสารสนเทศ
Major	สาขา	varchar	50			วิทยาการคอมพิวเตอร์

ตารางที่ 3.5 ข้อมูลกิจกรรม (Event)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
Id	รหัสกิจกรรม	int		PK		12
Name	ชื่อกิจกรรม	varchar	255			อบรม Figma

ตารางที่ 3.5 ข้อมูลกิจกรรม (Event) (ต่อ)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
StartDate Time	วันเวลาเริ่มต้น กิจกรรม	datetime				2022-09-01 09:30:00
EndDate Time	วันเวลาสิ้นสุด กิจกรรม	datetime				2022-10-01 16:30:00
Description	รายละเอียด กิจกรรม	varchar	1000			การใช้Figma ออกแบบ
Privacy	สถานะความ เป็นส่วนตัว	varchar	50			ส่วนตัว
Image	รูปกิจกรรม	varchar	500			-
Code	รหัสกิจกรรม	varchar	15			FG63X1
UserId	รหัสผู้ใช้	int		FK	user	7

ตารางที่ 3.6 ข้อมูลการเข้าร่วม (Join)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
Id	รหัสผู้เข้าร่วม	int		PK		1
UserId	รหัสผู้ใช้	int		FK	users	7
EventId	รหัสกิจกรรม	int		FK	event	12

ตารางที่ 3.7 ข้อมูลจุดเช็คอิน (CheckPoint)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
Id	รหัสจุดเช็คอิน	int		PK		22
Name	ชื่อจุดเช็คอิน	varchar	30			เช็คอินครั้งที่1
StartTime	เวลาเริ่มต้นใน การเช็คอิน	time				12:00:00

ตารางที่ 3.7 ข้อมูลจุดเช็คชื่อ (CheckPoint) (ต่อ)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
EndTime	เวลาสิ้นสุดในการเช็คชื่อ	time				13:45:00
Date	วันที่เช็คชื่อ	date				2022-09-01
Lat_Lng	ค่าพิกัด	varchar	255			16.24641659 8636216, 103.2518969 0721659
EventId	รหัสกิจกรรม	int		FK	event	12

ตารางที่ 3.8 ข้อมูลผู้เช็คชื่อ (Checker)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
Id	รหัสผู้ที่จะเช็ค	int		PK		67
JoinId	รหัสผู้เข้าร่วม	int		FK	join	1
Check PointId	รหัสจุดเช็คชื่อ	int		FK	checkPoint	22

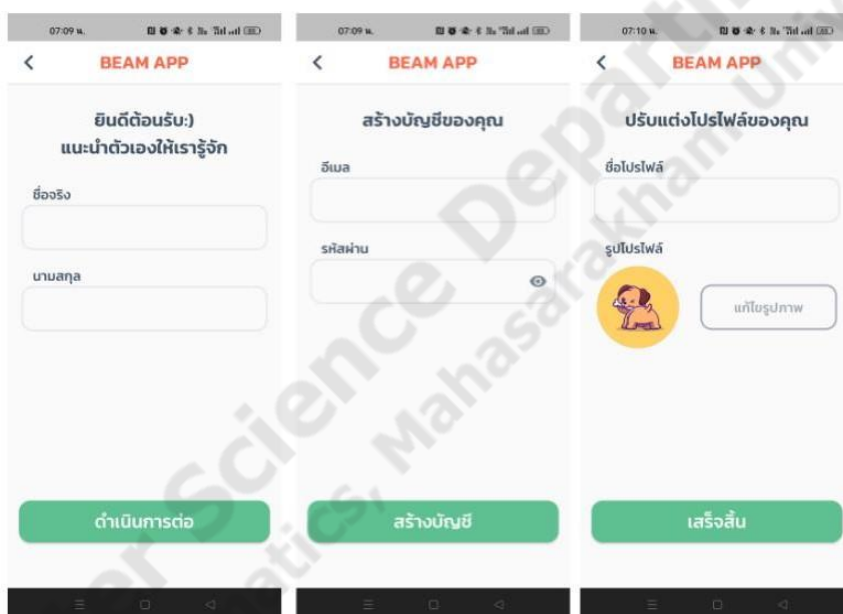
ตารางที่ 3.9 ข้อมูลเช็คชื่อ (Checked)

Attribute name	Description	Type	Length	Constraint	Reference Table	Sample Data
Id	รหัสเช็คชื่อ	int		PK		12
Time	วันเวลาเช็คชื่อ	time stamp				2022-09-01 12:14:00
Check PointId	รหัสจุดเช็คชื่อ	int		FK	checkPoint	22
JoinId	รหัสผู้เข้าร่วม	int		FK	join	1

3.6 การพัฒนาระบบ

3.6.1 การสมัครสมาชิก

การสมัครสมาชิกจะแบ่งเป็น 3 หน้า หน้าแรกเป็นการให้ผู้ใช้กรอก ชื่อ-นามสกุล หน้าที่สองจะให้ผู้ใช้กรอก อีเมลและรหัสผ่าน หน้าสามให้ผู้ใช้ทำการกรอกชื่อเล่นและปรับแต่งรูปภาพโปรไฟล์ จากนั้นนำข้อมูลที่ได้จาก 3 หน้า บันทึกลงฐานข้อมูลเพื่อนำไปใช้ในการเข้าสู่ระบบในรอบต่อ ๆ ไป



ภาพประกอบที่ 3.10 ตัวอย่างหน้าจอการสมัครสมาชิกในแต่ละหน้า

```

103 | child: DefaultButton(
104 |   text: "ดำเนินการต่อ",
105 |   backgroundColor: cGreen,
106 |   onPressed: () {
107 |     String firstName = _textFirstName.text.trim();
108 |     String lastName = _textLastName.text.trim();
109 |     context.read<UserProvider>().user.firstName = firstName;
110 |     context.read<UserProvider>().user.lastName = lastName;
111 |     Get.to(() => const RegisterPage1());
112 |   },
113 | ), // DefaultButton

```

ภาพประกอบที่ 3.11 การเก็บข้อมูลชื่อ-นามสกุลจากหน้าแรก

- บรรทัดที่ 107 รับค่าของชื่อจาก TextField เก็บไว้ในตัวแปร firstName
- บรรทัดที่ 108 รับค่าของนามสกุลจาก TextField เก็บไว้ในตัวแปร lastName
- บรรทัดที่ 109 นำค่าที่เก็บไว้ในตัวแปร firstName บันทึกลงไปใน UserProvider
- บรรทัดที่ 110 นำค่าที่เก็บไว้ในตัวแปร lastName บันทึกลงไปใน UserProvider
- บรรทัดที่ 111 เป็นคำสั่งที่ใช้เปลี่ยนไปที่หน้าถัดไป

```

110 child: DefaultButton(
111   text: "สร้างบัญชี",
112   backgroundColor: cGreen,
113   onPressed: () async {
114     String email = _textEmail.text.trim();
115     String password = _textPassword.text.trim();
116     context.read<UserProvider>().user.email = email;
117     context.read<UserProvider>().user.password = password;
118     Get.to(() => const RegisterPage2());
119   },
120 ), // DefaultButton

```

ภาพประกอบที่ 3.12 การเก็บข้อมูลอีเมล-รหัสผ่านจากหน้าที่สอง

- บรรทัดที่ 114 รับค่าอีเมลจาก TextField เก็บไว้ในตัวแปร email
- บรรทัดที่ 115 รับค่ารหัสผ่านจาก TextField เก็บไว้ในตัวแปร password
- บรรทัดที่ 116 นำค่าที่เก็บไว้ในตัวแปร email บันทึกลงไปใน UserProvider
- บรรทัดที่ 117 นำค่าที่เก็บไว้ในตัวแปร password บันทึกลงไปใน UserProvider
- บรรทัดที่ 118 เป็นคำสั่งที่ใช้เปลี่ยนไปที่หน้าถัดไป

```

177 context.read<UserProvider>().user.nickName = nickName;
178 context.read<UserProvider>().user.image = imgUrl;
179 var register = await _userService.register(context.read<UserProvider>().user);
180 if (register.response.statusCode == 201) {
181   if (register.data.rowsAffected > 0) {
182     toast(context, "Registration successfully.");
183     Get.to(() => const HomeScreen());
184   } else if (register.data.rowsAffected == 0) {
185     toast(context, "already exist.");
186   } else {
187     toast(context, "Registration failed.");
188   }
189 }

```

ภาพประกอบที่ 3.13 การเก็บชื่อเล่น-รูปภาพ และการบันทึกข้อมูลลงฐานข้อมูล

- บรรทัดที่ 177 นำค่าที่เก็บไว้ในตัวแปร nickName บันทึกลงไปใน UserProvider
- บรรทัดที่ 178 นำค่าที่เก็บไว้ในตัวแปร imageUrl บันทึกลงไปใน UserProvider
- บรรทัดที่ 179 นำ Object user ที่อยู่ใน UserProvider บันทึกลงในฐานข้อมูล
- บรรทัดที่ 182-183 แสดงข้อความในกรณีที่ สมัครสมาชิกสำเร็จและสั่งเปลี่ยนไปที่หน้าเข้าสู่ระบบ
- บรรทัดที่ 185 แสดงข้อความในกรณีที่ อีเมลที่ใช้สมัครมีอยู่แล้ว
- บรรทัดที่ 187 แสดงข้อความในกรณีที่ มีข้อผิดพลาด หรือกรอกข้อมูลไม่ครบ

```

38 // Register implements UserService
39 func (userService) Register(userModel model.User) int64 {
40
41     user := userRepo.GetUserByEmail(userModel.Email)
42
43     if user.ID == 0 {
44         hash := HashPassword(userModel.Password)
45         userModel.Password = hash
46         rowsAffected := userRepo.Register(userModel)
47
48         if rowsAffected > 0 {
49             return 1
50         } else if rowsAffected == 0 {
51             return 0
52         } else {
53             return -1
54         }
55     } else {
56         return 0
57     }
58 }

```

ภาพประกอบที่ 3.14 ฟังก์ชันในการสมัครสมาชิก

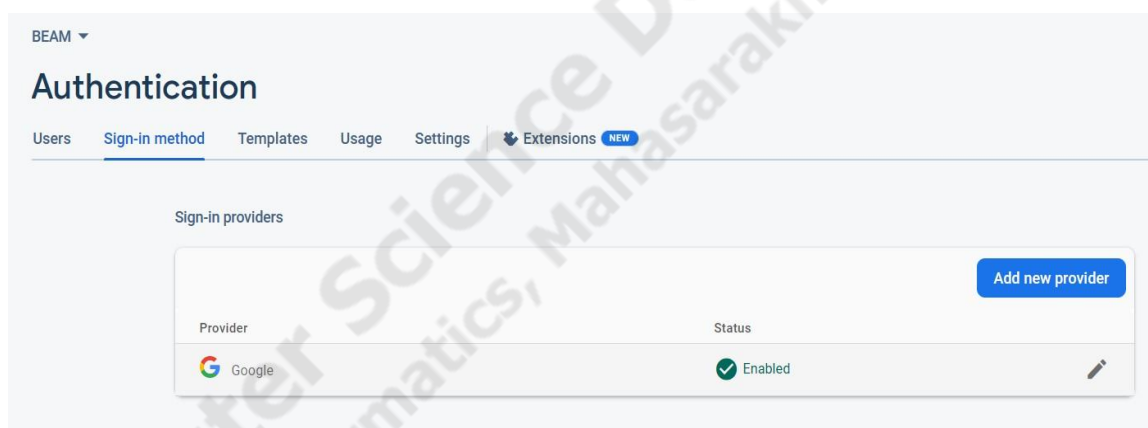
จากภาพประกอบที่ 3.14 เป็นฟังก์ชันของการสมัครสมาชิกโดยรับ ค่า Object userModel เข้ามาทำงานในฟังก์ชันเพื่อสมัครสมาชิก

- บรรทัดที่ 41 ดึงข้อมูลของผู้ใช้เก็บไว้ในตัวแปร user โดยการส่งค่าอีเมลใน Object userModel ที่รับจากพารามิเตอร์ ไปตรวจสอบ
- บรรทัดที่ 43 เป็นคำสั่งที่ใช้ตรวจสอบว่ามีผู้ใช้อีเมลผู้แล้วหรือไม่ หากไม่มี (user.Id == 0) ให้ทำงานในบรรทัดที่ 44-57

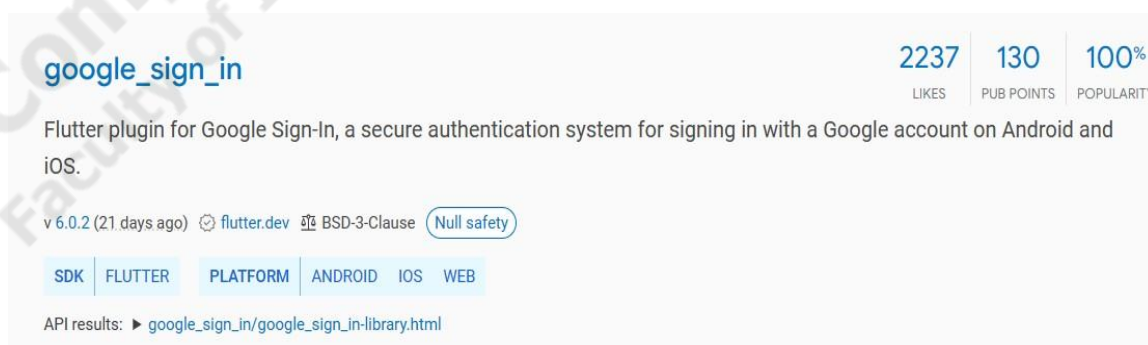
- บรรทัดที่ 44** นำค่ารหัสผ่านใน Object userModel ที่รับมาจากพารามิเตอร์ มาเข้ารหัส และเก็บไว้ในตัวแปร hash
- บรรทัดที่ 45** แทนที่ค่ารหัสผ่านที่เข้ารหัสแล้ว(hash) ใน Object userModel
- บรรทัดที่ 46** ทำการบันทึกข้อมูล Object userModel ลงในฐานข้อมูล

3.6.2 การเข้าสู่ระบบโดย Google Account

การเข้าสู่ระบบด้วย Google Account จะต้องมีการติดตั้ง Package google_sign_in จากเว็บไซต์ pub.dev และจะต้องมีการเปิดการใช้งาน Google Authentication Firebase ของ แอปพลิเคชันของเราด้วย ดังภาพประกอบที่ 3.15 และภาพประกอบที่ 3.16



ภาพประกอบที่ 3.15 ตัวอย่างการเปิดการใช้งาน Google Authentication ใน Firebase



ภาพประกอบที่ 3.16 Package ที่ใช้ในการเข้าสู่ระบบด้วย Google Account


```

100     GoogleAccountButton(
101         onPressed: () async {
102             await context.read<GoogleSignInProvider>().login();
103         },
104     ), // GoogleAccountButton

```

ภาพประกอบที่ 3.17 ปุ่มกดที่ใช้ในการเข้าสู่ระบบ Google Account

บรรทัดที่ 102 ทำการเรียกใช้ Method login() ที่อยู่ใน class GoogleSignInProvider

```

6     class GoogleSignInProvider with ChangeNotifier {
7         GoogleSignIn googleSignIn = GoogleSignIn();
8         GoogleSignInAccount? googleSignInAccount;
9         late UserService userService;
10        String? id;
11        String? displayName;
12        String? email;
13        String? photoUrl;
14
15        Future<void> login() async {
16            googleSignInAccount = await googleSignIn.signIn();
17            if (googleSignInAccount != null) {
18                id = googleSignInAccount!.id;
19                displayName = googleSignInAccount!.displayName;
20                email = googleSignInAccount!.email;
21                photoUrl = googleSignInAccount!.photoUrl;
22            }
23        }
24
25        Future<void> logout() async {
26            if (await googleSignIn.isSignedIn()) {
27                await googleSignIn.signOut();
28            }
29        }
30    }

```

ภาพประกอบที่ 3.18 ข้อมูลต่างๆที่ใช้งานเกี่ยวกับการเข้าสู่ระบบด้วย Google Account

บรรทัดที่ 10-13 ประกาศตัวแปรที่เก็บข้อมูลของ Google Account มี id, displayName, email, photoUrl เป็นต้น

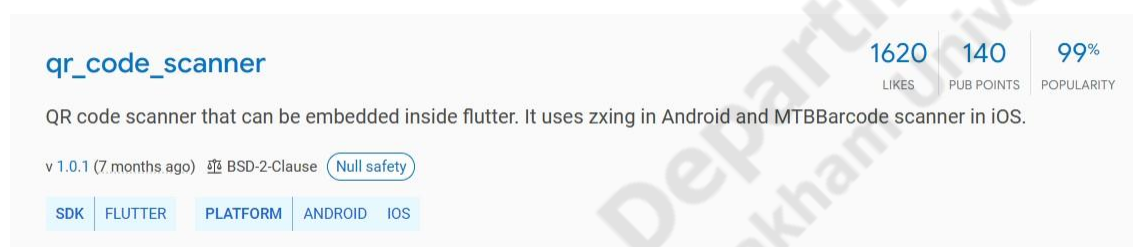
บรรทัดที่ 16 เรียกใช้คำสั่งเพิ่มเข้าสู่ระบบ

บรรทัดที่ 17-22 ตรวจสอบว่าหากมีการเข้าสู่ระบบสำเร็จ ทำการเก็บค่าข้อมูลจาก Google Account ไว้ในตัวแปร id, displayName, email, photoUrl ที่ประกาศไว้

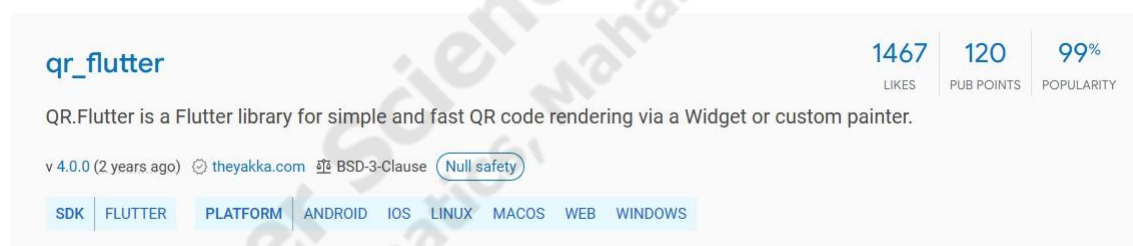
บรรทัดที่ 25-29 Method `logout()` จะทำการเช็คค่าว่าหากมีการเข้าสู่ระบบอยู่จะทำการ เรียกใช้คำสั่งในการออกจากระบบ

3.6.3 การสร้างและสแกนคิวอาร์โค้ด

การสร้างคิวอาร์โค้ดจะมีการติดตั้ง Package `qr_flutter` ในส่วนของการสแกนคิวอาร์ โค้ดจะติดตั้ง Package `qr_code_scanner` จากเว็บไซต์ `pub.dev` ดังภาพประกอบที่ 3.19 ภาพประกอบที่ 3.20



ภาพประกอบที่ 3.19 Package ที่ใช้ในการสแกนคิวอาร์โค้ด



ภาพประกอบที่ 3.20 Package ที่ใช้ในการสร้างคิวอาร์โค้ด

```

76  ✓ String exampleData() {
77  ✓   var data = {
78     "uuid": const Uuid().v4(),
79     "major": math.Random().nextInt(10),
80     "minor": math.Random().nextInt(10),
81   };
82   return data.toString();
83 }

```

ภาพประกอบที่ 3.21 ข้อมูลตัวอย่างของ คิวอาร์โค้ด ซึ่งเป็นการสร้าง UUID

```

50  QrImage(
51      data: exampleData(),
52      size: size.width * 0.7,
53      backgroundColor: Colors.white,
54  ), // QrImage
55  const Gap(50),
56  DefaultButton(
57      text: "สร้าง QR Code",
58      backgroundColor: cBlue,
59      onPressed: () {
60          setState(() {});
61      },
62  ), // DefaultButton
63  DefaultButton(
64      text: "สแกน QR Code",
65      backgroundColor: cGreen,
66      onPressed: () {
67          Get.to(() => const DemoQrScanner());
68      },
69  ), // DefaultButton

```

ภาพประกอบที่ 3.22 การสร้างคิวอาร์โค้ด

- บรรทัดที่ 50-54 class QrImage เป็นการแสดงรูปภาพของคิวอาร์โค้ดซึ่งกำหนด Attribute data เป็นข้อมูลของคิวอาร์โค้ด มีข้อมูลตัวอย่างจาก method exampleData()
- บรรทัดที่ 60 ทุกครั้งที่กดปุ่มสร้างคิวอาร์โค้ด จะทำการเรียก method setState() ซึ่งจะมีการสร้างคิวอาร์โค้ดใหม่ทุกครั้งเมื่อมีการ setState()
- บรรทัดที่ 67 คำสั่งที่ใช้เปลี่ยนหน้าแอปพลิเคชันไปที่หน้าสแกนคิวอาร์โค้ด

```

31  QRView buildQrView(BuildContext context) {
32      return QRView(
33          key: qrKey,
34          onQRViewCreated: _onQRViewCreated,
35          overlay: QrScannerOverlayShape(
36              cutOutSize: MediaQuery.of(context).size.width * 0.9,
37              borderWidth: 10,
38              borderLength: 20,
39              borderRadius: 15,
40              borderColor: cOrange,
41          ), // QrScannerOverlayShape
42      ); // QRView
43  }

```

ภาพประกอบที่ 3.23 การสร้างหน้าต่าง Overlay ในการสแกนคิวอาร์โค้ด

```

45   QRViewController? controller;
46   Barcode? result;
47
48   void _onQRViewCreated(QRViewController controller) {
49     setState(() {
50       this.controller = controller;
51     });
52     controller.scannedDataStream.listen((scanData) {
53       setState(() {
54         result = scanData;
55       });
56     });
57     controller.pauseCamera();
58     controller.resumeCamera();
59   }

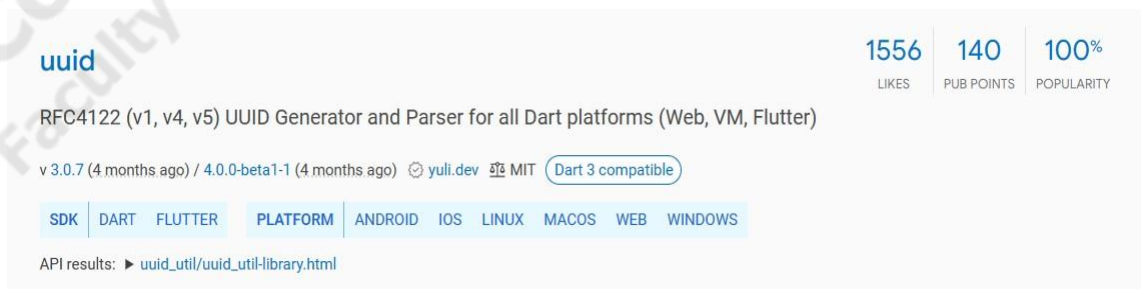
```

ภาพประกอบที่ 3.24 การสแกนคิวอาร์โค้ด

- บรรทัดที่ 34 เรียกใช้ method_onQRViewCreated เพื่อให้กล้องพร้อมที่จะสแกน
- บรรทัดที่ 52-56 สั่งเริ่มสแกนหาบาร์โค้ดเมื่อกำลังตรวจพบบาร์โค้ดจะทำการเก็บข้อมูลที่ สแกนได้ไว้ในตัวแปร result

3.6.4 การกระจายสัญญาณบีคอนและการสแกนบีคอน

การกระจายสัญญาณบีคอนและการสแกนบีคอนจะต้องติดตั้ง Package flutter_beacon และในการกระจายสัญญาณบีคอนจะต้องมีรหัส UUID ที่จะปล่อยจึงต้องติดตั้ง Package uuid



uuid 1556 140 100%
LIKES PUB POINTS POPULARITY

RFC4122 (v1, v4, v5) UUID Generator and Parser for all Dart platforms (Web, VM, Flutter)

v 3.0.7 (4 months ago) / 4.0.0-beta1-1 (4 months ago) [yuli.dev](#) [MIT](#) [Dart 3 compatible](#)

[SDK](#) [DART](#) [FLUTTER](#) [PLATFORM](#) [ANDROID](#) [IOS](#) [LINUX](#) [MACOS](#) [WEB](#) [WINDOWS](#)

API results: [▶ uuid_util/uuid_util-library.html](#)

ภาพประกอบที่ 3.25 Package ที่ใช้ในการสร้างรหัส UUID

```

106 |         onPressed: () async {
107 |             if (await flutterBeacon.isBroadcasting()) {
108 |                 await flutterBeacon.stopBroadcast();
109 |             } else {
110 |                 Uuid genUuid = const Uuid();
111 |                 String uuid = genUuid.v4();
112 |                 await flutterBeacon.startBroadcast(
113 |                     BeaconBroadcast(
114 |                         proximityUUID: uuid,
115 |                         major: math.Random().nextInt(10),
116 |                         minor: math.Random().nextInt(10),
117 |                     ), // BeaconBroadcast
118 |                 );
119 |             }
120 |             setState(() {});
121 |         },
122 |     ), // FlatButton

```

ภาพประกอบที่ 3.26 การกระจายสัญญาณบีคอน

- บรรทัดที่ 107-108 เมื่อมีการกดปุ่มกระจายสัญญาณตรวจสอบการกระจายสัญญาณบีคอน หากมีการกระจายสัญญาณอยู่ให้หยุดการกระจายสัญญาณ
- บรรทัดที่ 110-111 สร้างรหัส UUID และเก็บในตัวแปร uuid
- บรรทัดที่ 112-118 ทำการเริ่มกระจายสัญญาณบีคอนโดยการส่งค่า uuid ที่สร้างมาไว้ออกไป

```

101 |     final regions = <Region>[];
102 |
103 |     if (Platform.isIOS) {
104 |         regions.add(
105 |             Region(
106 |                 identifier: 'Apple Airlocate',
107 |                 proximityUUID: 'E2C56DB5-DFFB-48D2-B060-D0F5A71096E0',
108 |             ),
109 |         );
110 |     } else {
111 |         regions.add(
112 |             Region(identifier: 'com.beacon'),
113 |         );
114 |     }
115 |
116 |     _streamRanging = flutterBeacon.ranging(regions).listen(
117 |         (RangingResult result) {
118 |             log(result.beacons.length.toString());
119 |         },
120 |     );

```

ภาพประกอบที่ 3.27 การสแกนหาสัญญาณบีคอน

- บรรทัดที่ 103-104 กำหนด Region identifier ของแพลตฟอร์ม iOS และ Android
- บรรทัดที่ 116-120 ทำการเริ่มสแกนหาสัญญาณบีคอน