

บทที่ 3

ขั้นตอนการดำเนินงาน

บทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงการปริญญาโท ซึ่งจะทำให้ทราบถึงการวิเคราะห์และการออกแบบระบบโดยละเอียด ว่ามีแนวทางในการทำงานหรือขั้นตอนในการทำงานของระบบอย่างไร โดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้

3.1 กรอบการดำเนินงาน

3.2 การไหลของข้อมูล (Data Flow Diagram)

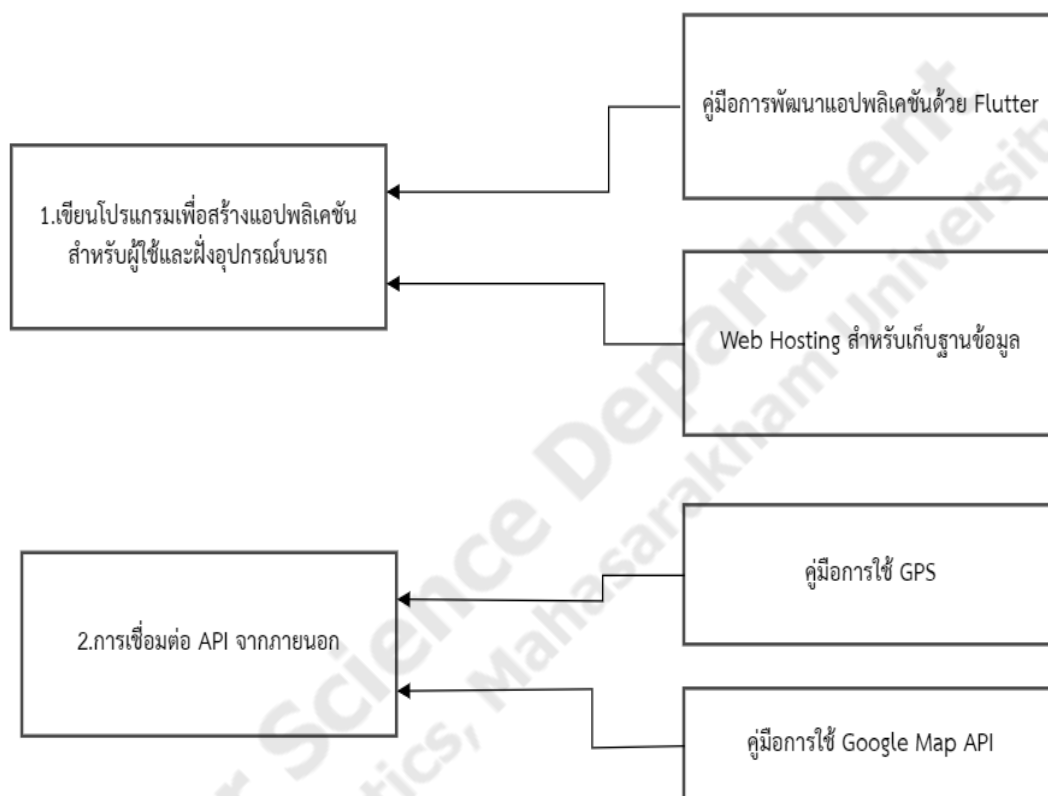
- Context Diagram
- Data Flow Diagram Level 1
- Data Flow Diagram Level 2

3.3 พจนานุกรม (Data Dictionary)

- External Entity Description
- Data Flow Description and Data Structure
- Data Store Description and Data Structure
- Process Description

3.4 ความสัมพันธ์ (ER Diagram)

3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบ

คำอธิบาย

1. สร้างแอปพลิเคชันสำหรับผู้ใช้ และ อุปกรณ์บนรถ โดยการสร้างแอปพลิเคชันต้องมีการใช้เครื่องมือดังนี้

- 1) Visual Studio Code หรือ VS Code เป็นโปรแกรม Editor ที่สามารถทำงานได้หลายแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux ซัพพอร์ตภาษาหลายภาษา มีความสามารถในการเปิดอ่าน แก้ไข และเขียนโค้ดได้เหมือน Editor ตัวอื่นๆ รวมถึงความสามารถในการติดตั้งเครื่องมือเสริม ซึ่งใช้งานง่ายและฟรี Editor นี้จะต้องติดตั้ง Flutter , Dart , php สำหรับการเขียนแอปพลิเคชัน ใช้ Flutter , Dart ในการเขียน GUI ต่างๆ ส่วนภาษา php เอาไว้สร้าง Model ต่างๆ สำหรับเก็บค่าข้อมูล เมื่อมีเครื่องมือต่างๆ ครบแล้วก็ทำการสร้างแอปพลิเคชัน

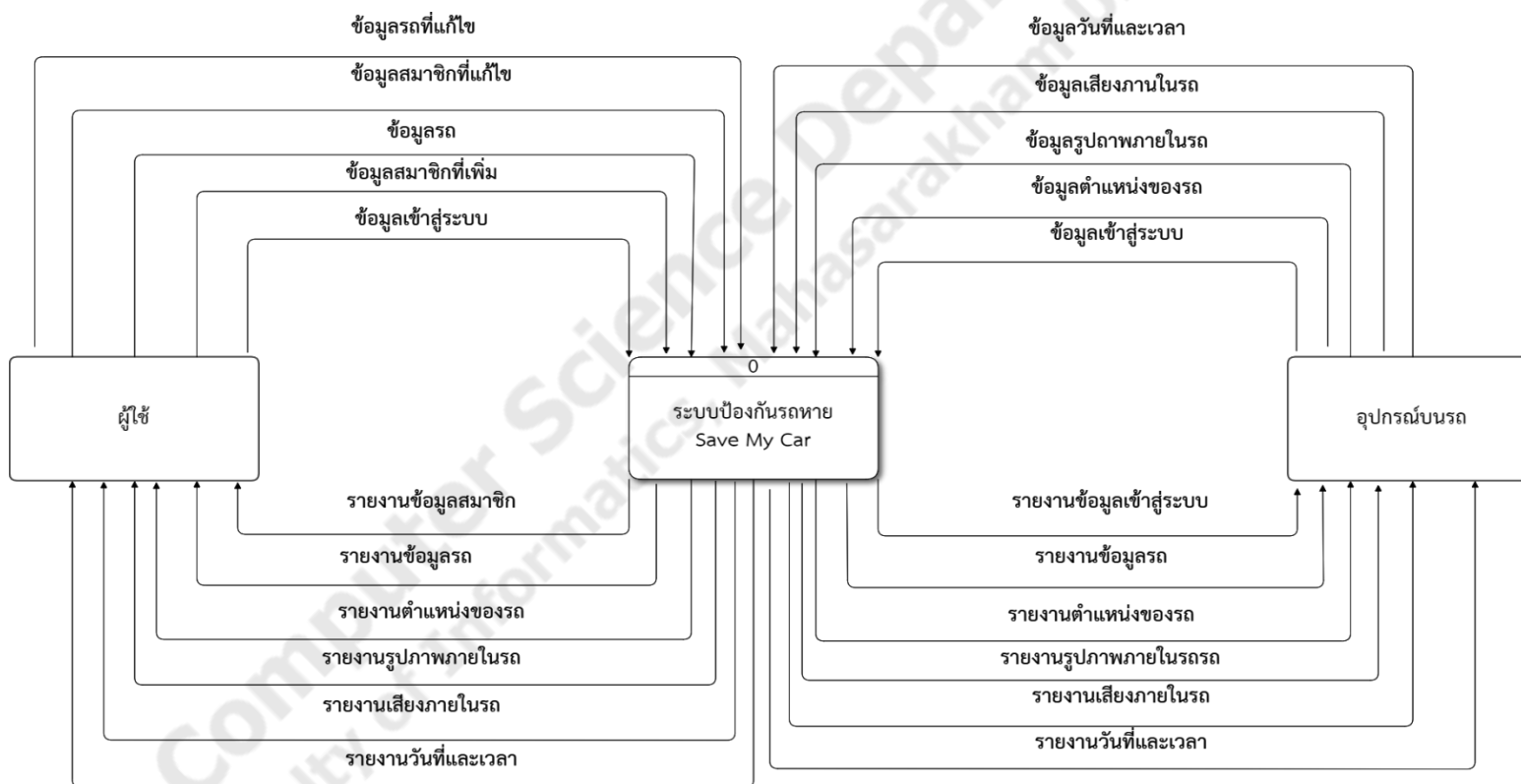
- 2) Web Hosting คือรูปแบบการให้บริการที่อนุญาตให้ผู้ให้บริการสามารถฝากเว็บไว้เพื่อออนไลน์บนอินเทอร์เน็ตได้ ซึ่งเราใช้ Web Hosting ในการเก็บตัว Model ซึ่งเขียนด้วยภาษา php เอาไว้บน Web Hosting จะได้สามารถส่งข้อมูลไปมาได้ และเอาไว้สร้าง Database เอาไว้บน Web Hosting ดังนั้นหน้าที่หลักๆของ Web Hosting คือการเป็น Server ในการเก็บและรับส่งข้อมูลนั่นเอง

2.การใช้งาน API ภายนอกโดยใช้ดังนี้

- 1) Google Map API เป็นชุด API ของ Google ใช้พัฒนา web application และ mobile application (Android, IOS)สำหรับเรียกใช้แผนที่และชุด service ต่างๆของ Google เพื่อพัฒนา application โดยการเขียนโปรแกรมด้วย Flutter ต้องลงไลบรารีชื่อ Google_map_flutter เพื่อเรียกใช้ Google Map API
- 2) GPS หรือ Global Positioning System คือระบบกำหนดตำแหน่งบนพื้นโลก จะทำงานโดยรับสัญญาณดาวเทียม โดยสัญญาณดาวเทียมนี้ประกอบไปด้วยข้อมูลที่ระบุตำแหน่งและเวลาขณะส่งสัญญาณ ตัวเครื่องรับสัญญาณ GPS เช่น สมาร์ทโฟนโดยการเขียนโปรแกรมด้วย flutter จะต้องลงไลบรารีชื่อ location สำหรับเรียกตำแหน่งจากสมาร์ทโฟน

3.2 ผลการออกแบบและพัฒนาระบบ

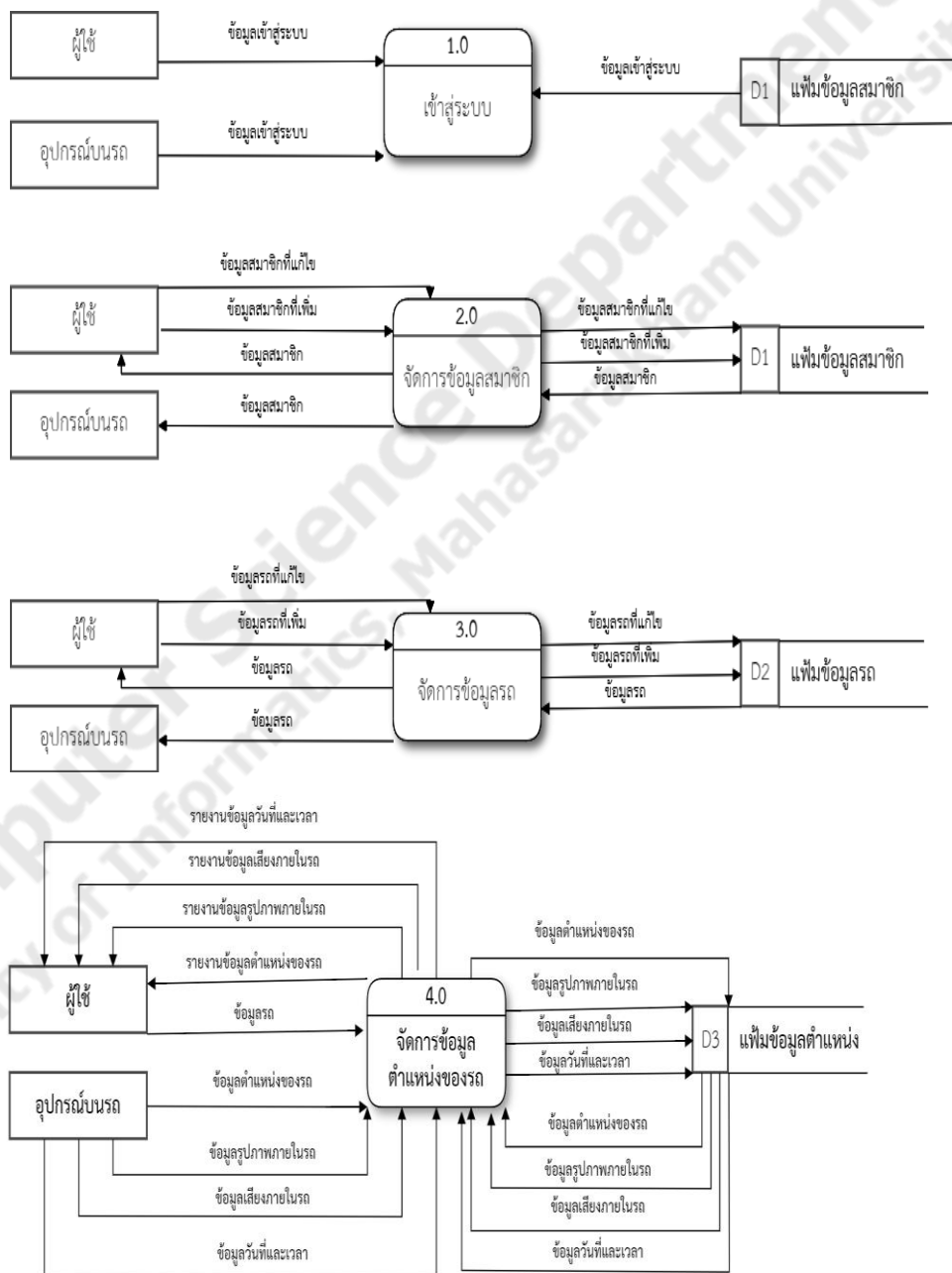
3.2.1 Context Diagram



ภาพประกอบที่ 3.2 Context Diagram ของแอปพลิเคชันป้องกันรถหาย

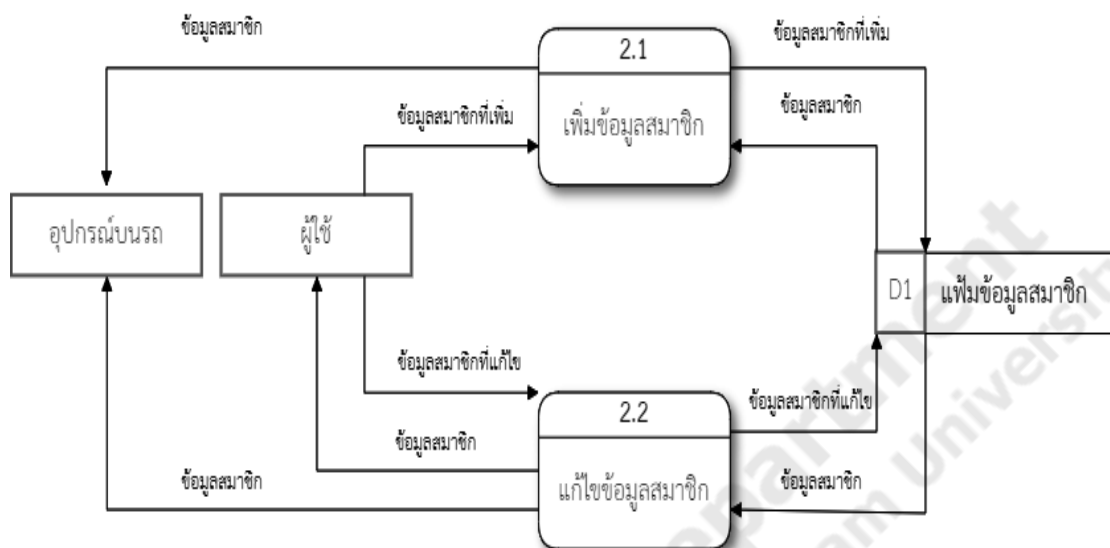
3.2.1 Data Flow Diagram Level 1

แผนภาพกระแสข้อมูลในระดับที่แสดงขั้นตอนการทำงานหลักทั้งหมด (Process หลัก) ของระบบแสดงทิศทางไหลของ Data Flow และแสดงรายละเอียดของแหล่งจัดเก็บข้อมูล (Data Store)

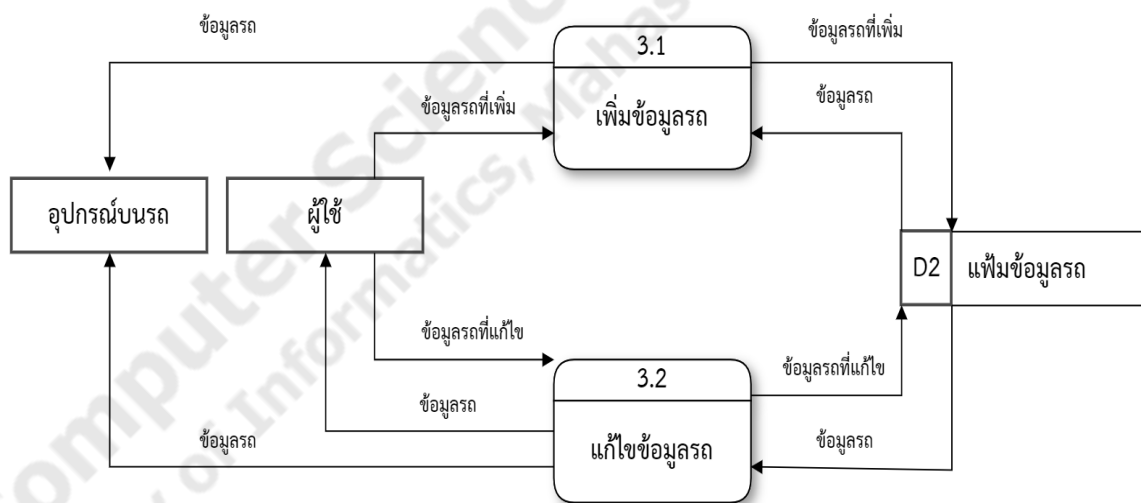


ภาพประกอบที่ 3.3 Data Flow Diagram level 1

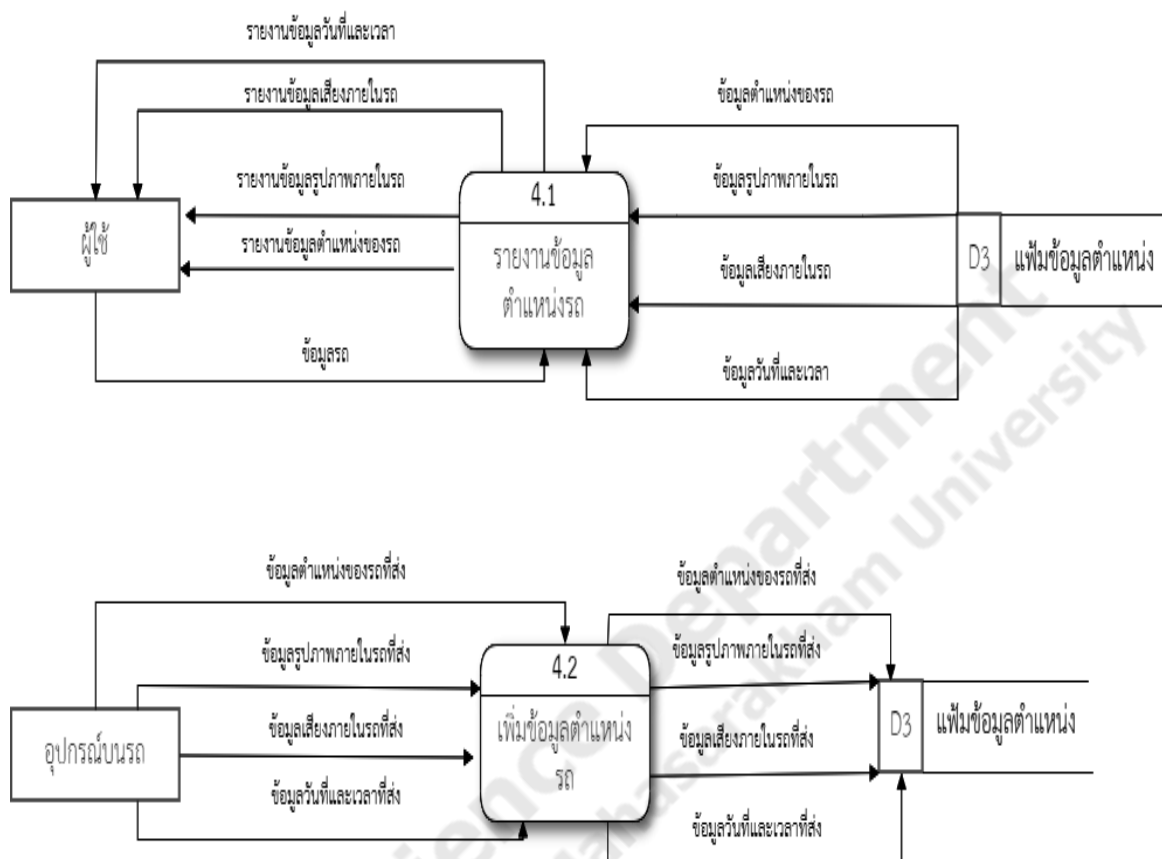
3.2.2 Data Flow Diagram Level 2



ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 Process 2



ภาพประกอบที่ 3.5 Data Flow Diagram Level 2 Process 3



ภาพประกอบที่ 3.6 Data Flow Diagram Level 2 Process 4

3.3 พจนานุกรม (Data Dictionary)

3.3.1 External Entity Description

พจนานุกรมข้อมูล เป็นส่วนที่ใช้ในการอธิบายถึงกระแสข้อมูลหรือแหล่งเก็บข้อมูลที่ปรากฏอยู่ในแผนภาพแสดงการไหลของกระแสข้อมูล

ตารางที่ 3.1 External Entity Description

Name	Description	Input Data Flow	Output Data Flow
ผู้ใช้	-จัดการข้อมูลส่วนตัวผู้ใช้ -จัดการข้อมูลรถ	-รายงานข้อมูลสมาชิก -รายงานข้อมูลรถ -รายงานตำแหน่งของรถ -รายงานรูปภาพภายในรถ -รายงานข้อมูลเสียงภายในรถ -รายงานวันที่เวลา	-ข้อมูลเข้าสู่ระบบ -ข้อมูลสมาชิกที่เพิ่ม -ข้อมูลรถ -ข้อมูลสมาชิกที่แก้ไข -ข้อมูลรถที่แก้ไข

ตารางที่ 3.1 External Entity Description(ต่อ)

Name	Description	Input Data Flow	Output Data Flow
อุปกรณ์บนรถ	-จัดการข้อมูลตำแหน่งของรถ	-รายงานข้อมูลเข้าสู่ระบบ -รายงานข้อมูลรถ -รายงานข้อมูลตำแหน่งของรถ -รายงานรูปภาพภายในรถ -รายงานเสียงภายในรถ -รายงานวันที่เวลา	-ข้อมูลเข้าสู่ระบบ -ข้อมูลตำแหน่งของรถ -ข้อมูลรูปภาพภายในรถ -ข้อมูลเสียงภายในรถ -ข้อมูลวันที่และเวลา

3.3.2 Data Store Description and Data Structure

Data Store คือ การนำข้อมูลเข้าไปจัดการในฐานข้อมูล โดยมีการแยกออกเป็นแฟ้มข้อมูล

ตารางที่ 3.2 Data Store Description and Data Structure

ID	Data Store Name	Description	Data Structure
D1	แฟ้มข้อมูลสมาชิก	เป็นแฟ้มที่เก็บข้อมูลส่วนตัวผู้ใช้	ไอดีผู้ใช้+Username + Password + คำนำหน้าชื่อ + ชื่อผู้ใช้ + นามสกุลผู้ใช้ + เบอร์โทร
D2	แฟ้มข้อมูลรถ	เป็นแฟ้มที่เก็บข้อมูลรถ	ไอดีรถ+หัวข้อ+ยี่ห้อรถ+ทะเบียนรถ+สีของรถ+รูปภาพของรถ
D3	แฟ้มข้อมูลตำแหน่งของรถ	เป็นแฟ้มที่เก็บข้อมูลตำแหน่งของรถ	ไอดีรถ+ตำแหน่งGPS+ภาพถ่าย+เสียงที่บันทึก+วันที่และเวลา

3.3.3 Data Flow Description and Data Structure

เป็นขั้นตอนการทำงานของระบบซึ่งทำให้เราทราบถึงการรับ-ส่งข้อมูลแสดงถึงการไหลของข้อมูลทั้งข้อมูลเข้า (Input) และข้อมูลส่งออก (Output) ระหว่างข้อมูลต้นทางถึงข้อมูลปลายทางโดยอธิบายข้อมูลและขั้นตอนในการดำเนินงานดังต่อไปนี้

ตารางที่ 3.3 Data Flow Description and Data Structure

Name	Description	Source	Destination	Data Structure
ข้อมูลการเข้าสู่ระบบ	การตรวจสอบ การเข้าสู่ระบบ	ผู้ใช้	Process 1.0 เข้าสู่ระบบ	Username + Password

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 1.0 เข้าสู่ระบบ	D1 เพิ่มข้อมูล สมาชิก	
ข้อมูลผู้ใช้ที่ เพิ่ม	ข้อมูลผู้ใช้ที่เพิ่ม	D1 เพิ่มข้อมูล สมาชิก	Process 2.1 เพิ่ม ข้อมูลสมาชิก	Username + Password + ชื่อและนามสกุลผู้ใช้ + เบอร์โทร + email
		Process 2.1 เพิ่มข้อมูล สมาชิก	D1 เพิ่มข้อมูล สมาชิก	
ข้อมูล สมาชิก	รายละเอียด ข้อมูลผู้ใช้	D1 เพิ่มข้อมูล สมาชิก	Process 2.1 เพิ่ม ข้อมูลสมาชิก	ชื่อและนามสกุลผู้ใช้ + เบอร์โทร +email
		Process 2.1 เพิ่มข้อมูล สมาชิก	อุปกรณ์บนรถ	
		D1 เพิ่มข้อมูล สมาชิก	Process 2.2 แก้ไขข้อมูลสมาชิก	
		Process 2.2 แก้ไขข้อมูล สมาชิก	อุปกรณ์บนรถ	
ข้อมูลการ แก้ไขผู้ใช้	การแก้ไขข้อมูล ผู้ใช้	Process 2.2 เพิ่มข้อมูล ส่วนตัว	D1 เพิ่มข้อมูล สมาชิก	ชื่อผู้ใช้ + นามสกุลผู้ใช้ + เบอร์โทร+email
ข้อมูลรถ	แสดง รายละเอียด ข้อมูลรถ	D2 เพิ่มข้อมูลรถ	Process 3.1 เพิ่ม ข้อมูลรถ	หัวข้อ+ยี่ห้อรถ + ทะเบียน รถ + สีของรถ + รูปภาพ ของรถ

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
		Process 3.1 แก้ไขข้อมูลรถ	อุปกรณ์บนรถ	
		D2 เพิ่มข้อมูลรถ	Process 3.2 แก้ไขข้อมูลรถ	
		Process 3.2 แก้ไขข้อมูลรถ	อุปกรณ์บนรถ	
		Process 4.1 รายงานข้อมูลตำแหน่งของรถ	D3 เพิ่มข้อมูลตำแหน่ง	
ข้อมูลรถที่เพิ่ม	การเพิ่มข้อมูลรถ	ผู้ใช้	Process 3.1 เพิ่มข้อมูลรถ	ไอทีรถ+หัวข้อ+ยี่ห้อรถ + ทะเบียนรถ + สีของรถ + รูปภาพของรถ
		Process 3.1 เพิ่มข้อมูลรถ	D2 เพิ่มข้อมูลรถ	
ข้อมูลรถที่แก้ไข	การแก้ไขข้อมูลรถ	D2 เพิ่มข้อมูลรถ	Process 3.2แก้ไขข้อมูลรถ	ยี่ห้อรถ + ทะเบียนรถ + สีของรถ + รูปภาพของรถ
		Process 3.2 แก้ไขข้อมูลรถ	D2 เพิ่มข้อมูลรถ	
รายงานข้อมูลตำแหน่งของรถ	ข้อมูลตำแหน่งของรถ	D3 เพิ่มข้อมูลตำแหน่ง	Process 4.1 รายงานข้อมูลตำแหน่งของรถ	ไอทีรถ+ตำแหน่งของรถ
		Process 4.1 รายงานข้อมูลตำแหน่งของรถ	ผู้ใช้	
รายงานข้อมูลวันที่และเวลา	รายละเอียดวันที่และเวลาของตำแหน่งรถ	D3 เพิ่มข้อมูลตำแหน่ง	Process 4.1 รายงานข้อมูลตำแหน่งของรถ	หัวข้อ+วันที่เวลา+ตำแหน่งรถ

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

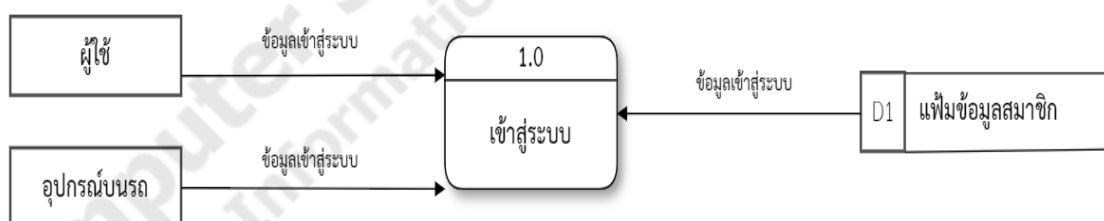
Name	Description	Source	Destination	Data Structure
		Process 4.1 รายงานข้อมูล ตำแหน่งของรถ	ผู้ใช้	
รายงานข้อมูล เสียงภายในรถ	ข้อมูลเสียงและ ตำแหน่งของรถ	D3 เพิ่มข้อมูล ตำแหน่ง	Process 4.1 รายงานข้อมูล ตำแหน่งของรถ	หัวข้อ+เสียง+วันที่ เวลา+ตำแหน่งรถ
		Process 4.1 รายงานข้อมูล ตำแหน่งของรถ	ผู้ใช้	
รายงานข้อมูล รูปภาพภายใน รถ	ข้อมูลภาพและ ตำแหน่งรถ	D3 เพิ่มข้อมูล ตำแหน่ง	Process 4.1 รายงานข้อมูล ตำแหน่งของรถ	หัวข้อ+ภาพ+วันที่ เวลา+ตำแหน่งรถ
		Process 4.1 รายงานข้อมูล ตำแหน่งของรถ	ผู้ใช้	
ข้อมูล ตำแหน่งของ รถที่ส่ง	ข้อมูลตำแหน่ง ของรถ	อุปกรณ์บนรถ	Process 4.2 เพิ่ม ข้อมูลตำแหน่งของ รถ	ตำแหน่งของรถ + วันที่และเวลา
		Process 4.2 เพิ่มข้อมูล ตำแหน่งของรถ	D3 เพิ่มข้อมูล ตำแหน่ง	
ข้อมูลรูปภาพ ภายในรถที่ส่ง	ข้อมูลภาพและ ตำแหน่งรถ	อุปกรณ์บนรถ	Process 4.2 เพิ่ม ข้อมูลตำแหน่งของ รถ	ภาพ+วันที่เวลา+ ตำแหน่งรถ
		Process 4.2 เพิ่มข้อมูล ตำแหน่งของรถ	D3 เพิ่มข้อมูล ตำแหน่ง	

ตารางที่ 3.3 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลเสียง ภายในรถที่ส่ง	ข้อมูลเสียงและ ตำแหน่งของรถ	อุปกรณ์บนรถ	Process 4.2 เพิ่ม ข้อมูลตำแหน่งของ รถ	เสียง+วันที่เวลา+ ตำแหน่งรถ
		Process 4.2 เพิ่มข้อมูล ตำแหน่งของรถ	D3 เพิ่มข้อมูล ตำแหน่ง	
ข้อมูลวันที่ และเวลาที่ส่ง	รายละเอียดวันที่ และเวลาของ ตำแหน่งรถ	อุปกรณ์บนรถ	Process 4.2 เพิ่ม ข้อมูลตำแหน่งของ รถ	วันที่เวลา+ตำแหน่ง รถ
		Process 4.2 เพิ่มข้อมูล ตำแหน่งของรถ	D3 เพิ่มข้อมูล ตำแหน่ง	

3.3.1 Process Description

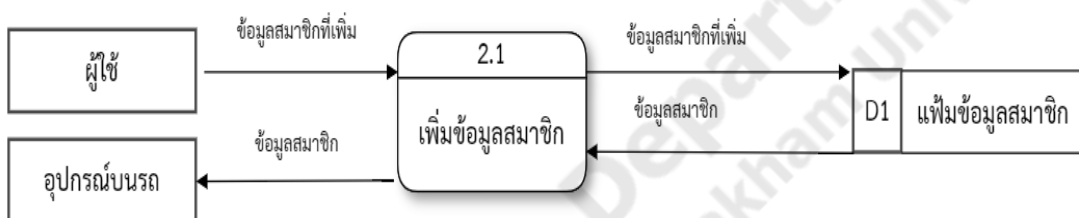
Process Description Process 1.0



ID	1.0
NAME	เข้าสู่ระบบ
DESCRIPTION	การเข้าสู่ระบบ
INPUT DATA FLOWS	-ข้อมูลเข้าสู่ระบบ
OUTPUT DATA FLOWS	-เข้าสู่ระบบสำเร็จ -เข้าสู่ระบบล้มเหลว
PROCESS DESCRIPTION	เริ่มต้น 1.กรอก Username และ Password

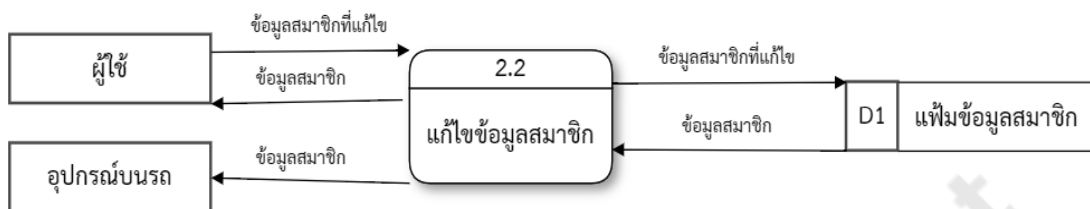
	<p>2.ตรวจสอบว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่</p> <p>ถ้า (ข้อมูลถูกต้อง)</p> <p>ให้ทำการแจ้งเตือนการเข้าสู่ระบบสำเร็จ</p> <p>ถ้า (ข้อมูลไม่ถูกต้อง)</p> <p>แสดงข้อความเตือนเข้าสู่ระบบล้มเหลว</p> <p>จบการทำงาน</p>
--	--

Process Description Process 2.1



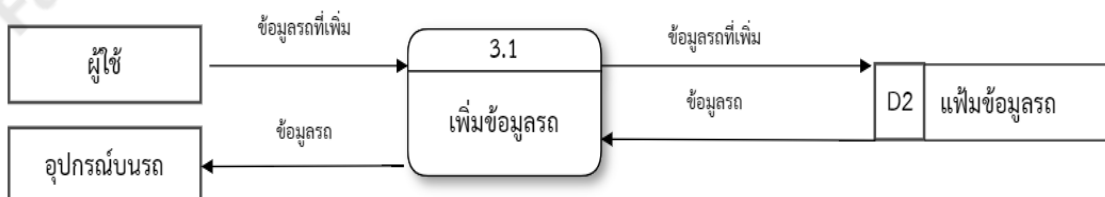
ID	2.1
NAME	การเพิ่มข้อมูลสมาชิก
DESCRIPTION	การเพิ่มข้อมูลสมาชิก
INPUT DATA FLOWS	-ข้อมูลสมาชิกที่ต้องการจะเพิ่ม -ข้อมูลสมาชิก
OUTPUT DATA FLOWS	-ข้อมูลสมาชิกที่ต้องการจะเพิ่ม -ข้อมูลสมาชิก
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <p>1.กรอกข้อมูลสมาชิกที่ต้องการจะเพิ่ม</p> <p>2.ตรวจสอบว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่</p> <p>ถ้า (ข้อมูลถูกต้องและครบถ้วน)</p> <p>ให้ทำการเพิ่มข้อมูล</p> <p>ถ้า (ข้อมูลไม่ถูกต้องและไม่ครบถ้วน)</p> <p>แสดงข้อความเตือนการเพิ่มข้อมูลผิดพลาด กรุณาเพิ่มข้อมูล</p> <p>จบการทำงาน</p>

Process Description Process 2.2



ID	2.2
NAME	แก้ไขข้อมูลสมาชิก
DESCRIPTION	การแก้ไขข้อมูลสมาชิก
INPUT DATA FLOWS	-ข้อมูลสมาชิกที่ต้องการแก้ไข -ข้อมูลสมาชิก
OUTPUT DATA FLOWS	-ข้อมูลสมาชิกที่ต้องการแก้ไข -ข้อมูลสมาชิก
PROCESS DESCRIPTION	เริ่มต้น <ol style="list-style-type: none"> 1.เลือกรายการข้อมูลสมาชิกที่ต้องการจะแก้ไข 2.กรอกข้อมูลที่จะแก้ไข 3.ยืนยันการแก้ไข ถ้า (ตกลง) ให้ทำการแก้ไข ถ้า (ยกเลิก) กลับสู่หน้าหลัก จบการทำงาน

Process Description Process 3.1



ID	3.1
NAME	การเพิ่มข้อมูลรถ
DESCRIPTION	การเพิ่มข้อมูลรถ
INPUT DATA FLOWS	-ข้อมูลรถที่ต้องการจะเพิ่ม -ข้อมูลรถ
OUTPUT DATA FLOWS	-ข้อมูลรถที่ต้องการจะเพิ่ม -ข้อมูลรถ
PROCESS DESCRIPTION	เริ่มต้น 1.กรอกข้อมูลรถที่ต้องการจะเพิ่ม 2.ตรวจสอบว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่ ถ้า (ข้อมูลถูกต้องและครบถ้วน) ให้ทำการเพิ่มข้อมูล ถ้า (ข้อมูลไม่ถูกต้องและไม่ครบถ้วน) แสดงข้อความเตือนการเพิ่มข้อมูลผิดพลาด กรุณาเพิ่มข้อมูล จบการทำงาน

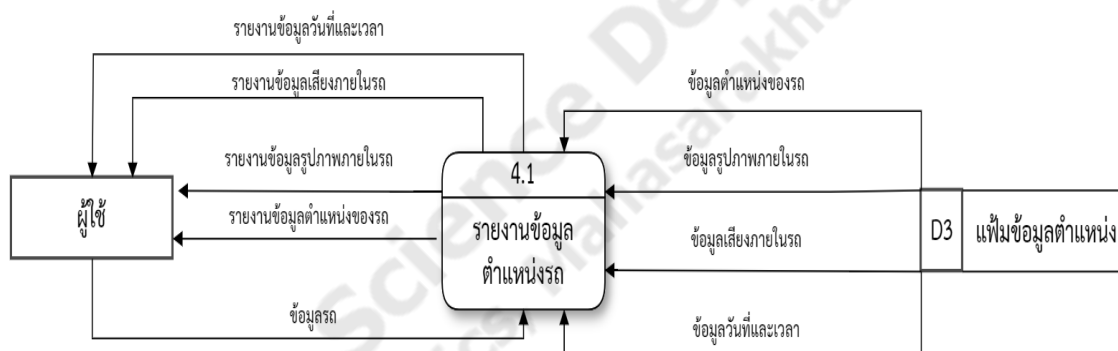
Process Description Process 3.2



ID	3.2
NAME	แก้ไขข้อมูลรถ
DESCRIPTION	การแก้ไขข้อมูลรถ
INPUT DATA FLOWS	-ข้อมูลรถที่ต้องการแก้ไข -ข้อมูลรถ
OUTPUT DATA FLOWS	-ข้อมูลรถที่ต้องการแก้ไข -ข้อมูลรถ

PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1.เลือกรายการรถที่ต้องการจะแก้ไข 2.กรอกข้อมูลที่จะแก้ไข 3.ยืนยันการแก้ไข <p>ถ้า (ตกลง)</p> <p style="padding-left: 40px;">ให้ทำการแก้ไข</p> <p>ถ้า (ยกเลิก)</p> <p style="padding-left: 40px;">กลับสู่หน้าหลัก</p> <p>จบการทำงาน</p>
---------------------	--

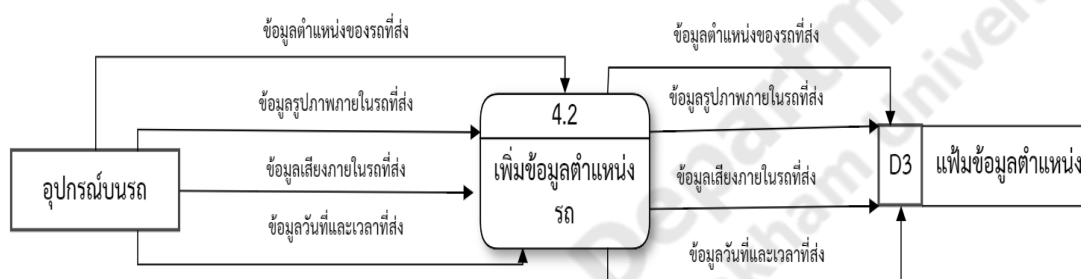
Process Description Process 4.1



ID	4.1
NAME	รายงานข้อมูลตำแหน่งรถ
DESCRIPTION	แสดงข้อมูลตำแหน่งรถ ภาพ และเสียง
INPUT DATA FLOWS	<ul style="list-style-type: none"> -ข้อมูลรถ -ข้อมูลตำแหน่งของรถ -ข้อมูลรูปภาพภายในรถ -ข้อมูลเสียงภายในรถ -ข้อมูลวันที่และเวลา
OUTPUT DATA FLOWS	<ul style="list-style-type: none"> -รายงานข้อมูลตำแหน่งของรถ -รายงานข้อมูลรูปภาพภายในรถ -รายงานข้อมูลเสียงภายในรถ

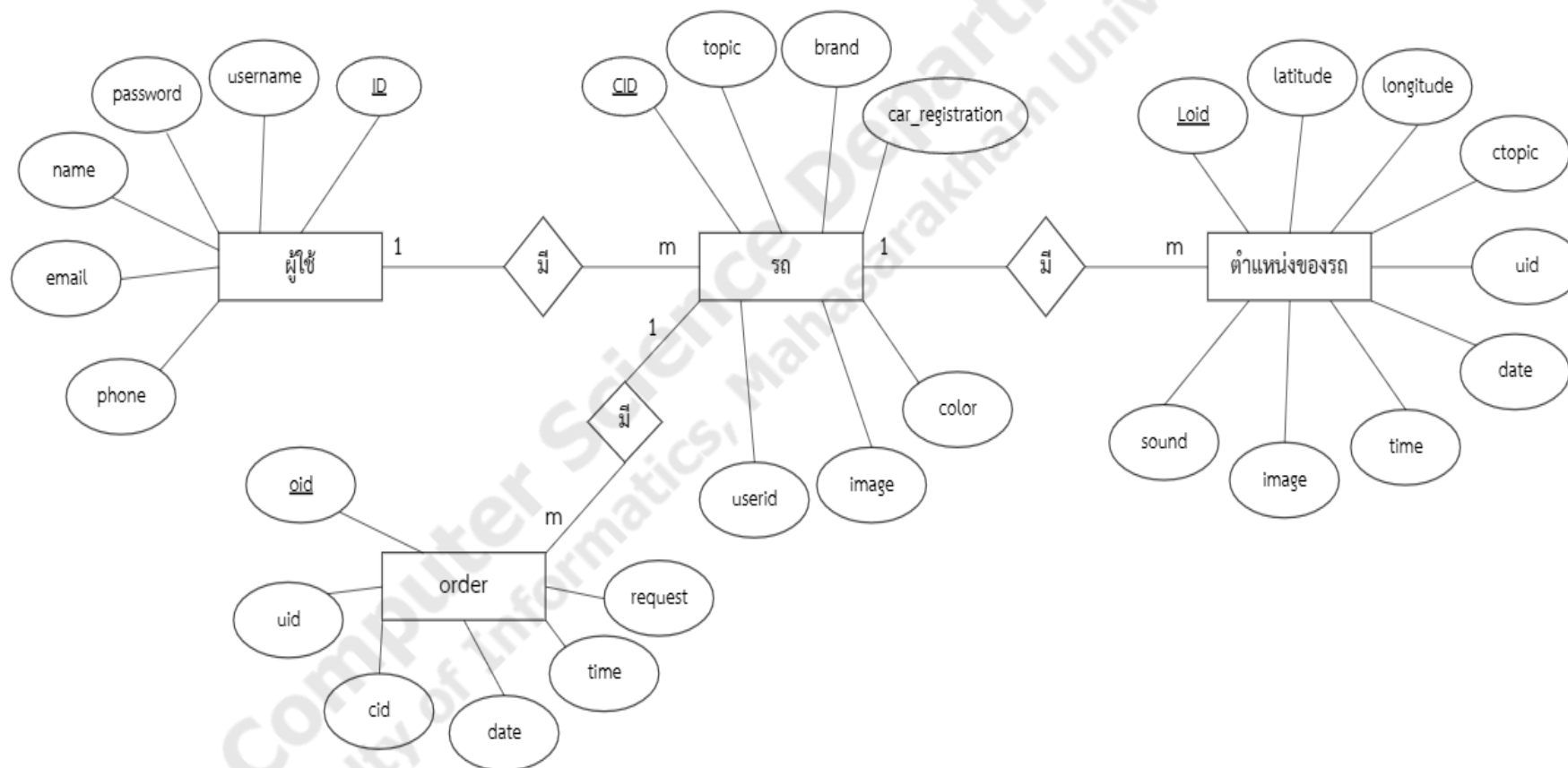
	-รายงานข้อมูลวันที่และเวลา
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1.เลือกรายการรถที่ต้องการดูตำแหน่ง ภาพ และเสียง 2.กดดูข้อมูลตำแหน่ง ภาพ และเสียง <p>จบการทำงาน</p>

Process Description Process 4.2



ID	4.2
NAME	เพิ่มข้อมูลตำแหน่งรถ
DESCRIPTION	เพิ่มข้อมูลตำแหน่งรถ ภาพ และเสียง
INPUT DATA FLOWS	<ul style="list-style-type: none"> -ข้อมูลตำแหน่งของรถที่ส่ง -ข้อมูลรูปภาพภายในรถที่ส่ง -ข้อมูลเสียงภายในรถที่ส่ง -ข้อมูลวันที่และเวลาที่ส่ง
OUTPUT DATA FLOWS	<ul style="list-style-type: none"> -ข้อมูลตำแหน่งของรถที่ส่ง -ข้อมูลรูปภาพภายในรถที่ส่ง -ข้อมูลเสียงภายในรถที่ส่ง -ข้อมูลวันที่และเวลาที่ส่ง
PROCESS DESCRIPTION	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1.เลือกรายการรถที่ต้องการส่งตำแหน่ง ภาพ และเสียง 2.กดส่งข้อมูล ตำแหน่ง ภาพ และเสียง <p>จบการทำงาน</p>

3.4 ความสัมพันธ์ (ER Diagram)



ภาพประกอบที่ 3.7 ความสัมพันธ์ (ER-Diagram)

3.4.1 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.5 ตารางผู้ใช้(User)

ID	Colum	Type	Description	Example	Constraint
1	id	int(25)	ไอดีสมาชิก	01	PK
2	username	varchar(255)	รหัสผู้ใช้	admin	Unique Key
3	password	varchar(255)	รหัสผ่าน	7afzTJdLgoKmQ6kfmY8rjA==	Not null
4	name	varchar(255)	ชื่อผู้ใช้	ธีรภัทร์	Not null
5	email	varchar(255)	อีเมล	Teerapat@gmail.com	Not null
6	phone	varchar(255)	เบอร์โทรศัพท์	099-9999999	Not null

ตารางที่ 3.6 ตารางข้อมูลรถ(Car)

ID	Colum	Type	Description	Example	Constraint
1	cid	int(255)	ไอดีรถ	01	PK
2	topic	varchar(255)	ชื่อของรถ	รถของเตี้ย	Not null
3	brand	varchar(255)	ยี่ห้อรถ	TOYOTA	Not null
4	generation	varchar(255)	รุ่นของรถ	Altis	Not null
5	car_registration	varchar(255)	ป้ายทะเบียนรถ	1150 กรุงเทพมหานคร	Not null
6	color	varchar(255)	สีรถ	แดง	Not null
7	image	text	รูปรถ	Image/logo.jpg	Not null
8	userid	int(255)	ไอดีสมาชิก	01	FK: reference from user(id) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.5 ตารางข้อมูลตำแหน่ง(Location)

ID	Colum	Type	Description	Example	Constraint
1	loid	int(255)	รหัสรถ	01	PK
2	latitude	double	ละติจูด	17.123456	Not null
3	longitude	double	ลองจิจูด	16.123456	Not null
4	cid	int(255)	ไอดีรถ	01	FK: reference from car(cid) ON DELETE CASCADE ON UPDATE CASCADE
5	uid	Int(255)	ไอดีสมาชิก	01	FK: reference from car(userid) ON DELETE CASCADE ON UPDATE CASCADE
6	date	date	วันที่	17-01-2021	Not null
7	time	time	เวลา	17.06.55	Not null
8	image	text	ไฟล์รูปภาพ	Image/logo.jpg	-
9	sound	text	ไฟล์เสียง	Sound/sound1.wav	-

ตารางที่ 3.8 ตารางข้อมูลการร้องขอ(Order)

ID	Colum	Type	Description	Example	Constraint
1	uid	int(255)	ไอดีสมาชิก	01	FK: reference from user(id) ON DELETE CASCADE ON UPDATE CASCADE

ตารางที่ 3.8 ตารางข้อมูลการร้องขอ(Order)(ต่อ)

ID	Colum	Type	Description	Example	Constraint
2	cid	int(255)	ไอดีรถ	01	FK: reference from car(cid) ON DELETE CASCADE ON UPDATE CASCADE
3	date	date	วันที่	17-01-202	Not null
4	time	time	เวลา	17.06.55	FK: reference from car(cid) ON DELETE CASCADE ON UPDATE CASCADE
5	request	Boolean	คำสั่ง	0	

3.5 ตัวอย่างโค้ดที่ใช้ในการพัฒนา

3.5.1 การนำตำแหน่งของแอปพลิเคชันบนรถไปเก็บไว้ใน Database

```

34
35 geocoding: ^1.0.5 #ติดตั้ง geocoding เวอร์ชัน 1.0.5 ใน pubspec.yaml
36 geolocator: ^6.1.1 #ติดตั้ง geolocator เวอร์ชัน 6.1.1 ใน pubspec.yaml
37

```

ภาพประกอบที่ 3.8 การติดตั้ง library เพื่อใช้ในการหาตำแหน่งปัจจุบัน

บรรทัดที่ 35 ติดตั้ง geocoding เวอร์ชัน 1.0.5 ในไฟล์ pubspec.yaml

บรรทัดที่ 36 ติดตั้ง geolocator เวอร์ชัน 6.1.1 ในไฟล์ pubspec.yaml

```

85 ..... child: Text('SEND'),
86 ..... onPressed: () {
87 .....   sendLocation();
88 .....   timer = Timer.periodic(const Duration(minutes: 1), (timer) {
89 .....     print(DateTime.now());
90 .....     sendLocation();
91 .....   }); // Timer.periodic
92 ..... }, // RaisedButton

```

ภาพประกอบที่ 3.9 ฟังก์ชันการส่งตำแหน่งของแอปพลิเคชันบนรถ

บรรทัดที่ 87 เมื่อผู้ใช้ทำการกด “ส่งตำแหน่ง” จะทำการเรียกฟังก์ชัน `sendLocation()` เพื่อเริ่มการส่งตำแหน่งของแอปพลิเคชันบนรถ

บรรทัดที่ 88 จะอัปเดตตำแหน่งของรถด้วยฟังก์ชัน `sendLocation()` ทุกๆ 1 นาที

```

127 Future sendLocation() async {
128   var url = Uri.parse("http://savemycarproject.cocopatch.com/location.php");
129   var data = {
130     "latitude": latitude.toString(),
131     "longitude": longitude.toString(),
132     "ctopic": args,
133     "uid": userid,
134     "date": now.toString(),
135     "time": DateTime.now().toString(),
136   };
137   var response = await http.post(url, body: data);

```

ภาพประกอบที่ 3.10 ฟังก์ชันการส่งตำแหน่งของรถไปยัง Database

บรรทัดที่ 128 เป็นคำสั่งอ่านค่าจากลิงค์ไปเก็บไว้ในตัวแปร “url”

บรรทัดที่ 130 ดึงค่าละติจูดจากแอปพลิเคชันบนรถมาเก็บใน “latitude”

บรรทัดที่ 131 ดึงค่าลองจิจูดจากแอปพลิเคชันบนรถมาเก็บใน “longitude”

บรรทัดที่ 132 ดึงค่าหัวข้อรถจากแอปพลิเคชันบนรถมาเก็บใน “ctopic”

บรรทัดที่ 133 ดึงค่าไอดีผู้ใช้จากแอปพลิเคชันบนรถมาเก็บใน “uid”

บรรทัดที่ 134 ดึงค่าวันที่ปัจจุบันจากแอปพลิเคชันบนรถมาเก็บใน “date”

บรรทัดที่ 135 ดึงค่าเวลาปัจจุบันจากแอปพลิเคชันบนรถมาเก็บใน “time”

บรรทัดที่ 137 เป็นคำสั่งส่งค่าจาก model ไปที่ database

	loid	latitude	longitude	ctopic	uid	date	time
<input type="checkbox"/> Edit Copy Delete	7	17.970471666666665	104.05085	dd	1	2021-07-08	14:27:56
<input type="checkbox"/> Edit Copy Delete	8	17.970455	104.0507931	dd	2	2021-07-08	14:31:41
<input type="checkbox"/> Edit Copy Delete	9	17.970455	104.0507931	dd	2	2021-07-08	14:32:00
<input type="checkbox"/> Edit Copy Delete	10	17.970333333333333	104.05059000000001	dd	2	2021-07-08	14:32:16
<input type="checkbox"/> Edit Copy Delete	11	17.9704547	104.0507904	dd	2	2021-07-08	14:32:41
<input type="checkbox"/> Edit Copy Delete	12	17.970106666666666	104.05072333333334	dd	1	2021-07-08	15:22:06

ภาพประกอบที่ 3.11 ค่าตำแหน่งปัจจุบันของแอปพลิเคชันบนรถใน Database

3.5.2 การนำรูปภาพภายในรถไปเก็บไว้ใน Database

```

32
33 image_picker: ^0.6.7 #ทำการติดตั้ง image_picker เวอร์ชัน ^0.6.7 ลงใน pubspec.yaml
34

```

ภาพประกอบที่ 3.12 การติดตั้ง library เพื่อใช้ในถ่ายรูป

บรรทัดที่ 33 ติดตั้ง image_picker เวอร์ชัน 0.6.7 ลงใน pubspec.yaml

```

152 ..... RaisedButton(
153 .....   shape: RoundedRectangleBorder(
154 .....     borderRadius: BorderRadius.circular(15)), // Rounded
155 .....   color: Colors.deepPurple[100],
156 .....   child: Text("UPLOAD"),
157 .....   onPressed: () {
158 .....     if (_image == null) {
159 .....       Fluttertoast.showToast(
160 .....         msg: "กรุณาลoadรูปก่อนอัปโหลด",
161 .....         toastLength: Toast.LENGTH_SHORT);
162 .....     } else {
163 .....       _upload(_image);
164 .....       Fluttertoast.showToast(
165 .....         msg: "อัปโหลดสำเร็จ",
166 .....         toastLength: Toast.LENGTH_SHORT);

```

ภาพประกอบที่ 3.13 ฟังก์ชันการถ่ายรูปจากกล้อง

บรรทัดที่ 163 เมื่อทำการกดถ่ายรูปจะไปเรียกใช้งาน เมธอด _upload

```

224 //อัปโหลดไฟล์ภาพ และข้อมูลส่งไปDatabase
225 Future _upload(File imageFile) async {
226   var stream =
227     new http.ByteStream(DelegatingStream.typed(imageFile.openRead()));
228   var length = await imageFile.length();
229   var uri = Uri.parse("http://savemycarproject.cocopatch.com/location.php");
230   var request = new http.MultipartRequest("POST", uri);
231   var multipartFile = new http.MultipartFile("image", stream, length,
232     filename: basename(imageFile.path));
233   request.fields["latitude"] = latitude.toString();
234   request.fields["longitude"] = longitude.toString();
235   request.fields["ctopic"] = args;
236   request.fields["uid"] = userid;
237   request.fields["date"] = now.toString();
238   request.fields["time"] = DateTime.now().toString();
239   request.files.add(multipartFile);
240
241   var response = await request.send();
242
243   if (response.statusCode == 200) {
244     print("Image Uploaded");
245   } else {
246     print("Upload Failed");
247   }
248   response.stream.transform(utf8.decoder).listen((value) {
249     print(value);
250   });
251 }

```

ภาพประกอบที่ 3.14 การนำรูปภาพภายในรถไปเก็บไว้ใน Database

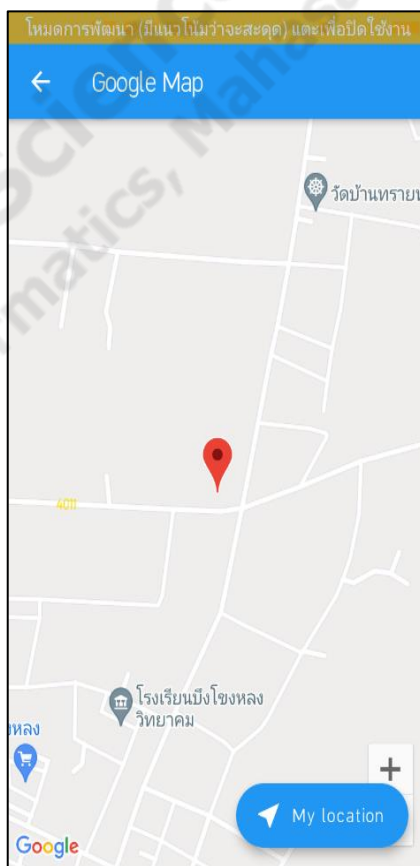
- บรรทัดที่ 229 เป็นคำสั่งอ่านค่าจากลิงค์ไปเก็บไว้ในตัวแปร “uri”
- บรรทัดที่ 230 เป็นคำสั่งส่งค่าจาก model ไปที่ database
- บรรทัดที่ 233 ดึงค่าละติจูดจากแอปพลิเคชันบนรถมาเก็บใน “latitude”
- บรรทัดที่ 234 ดึงค่าลองจิจูดจากแอปพลิเคชันบนรถมาเก็บใน “longitude”
- บรรทัดที่ 235 ดึงค่าหัวข้อรถจากแอปพลิเคชันบนรถมาเก็บใน “ctopic”
- บรรทัดที่ 236 ดึงค่าไอดีผู้ใช้จากแอปพลิเคชันบนรถมาเก็บใน “uid”
- บรรทัดที่ 237 ดึงค่าวันที่ปัจจุบันจากแอปพลิเคชันบนรถมาเก็บใน “date”
- บรรทัดที่ 238 ดึงค่าเวลาปัจจุบันจากแอปพลิเคชันบนรถมาเก็บใน “time”

3.5.3 การใช้งาน Google Map api

33 google_maps_flutter: ^0.5.21 #<----- ติดตั้ง dependencies google_maps_flutter เวอร์ชัน ^0.5.21 ก่อนเพื่อที่จะได้เรียกใช้งาน Library ได้

ภาพประกอบที่ 3.15 การติดตั้ง library เพื่อเรียกใช้ Google map api

บรรทัดที่ 33 ติดตั้ง google_map_flutter เวอร์ชัน 0.5.21 ลงในไฟล์ pubspec.yaml



ภาพประกอบที่ 3.16 หน้าแสดงตำแหน่งที่เรียกมาจาก database


```

34
35 @Override
36 Widget build(BuildContext context) {
37   // TODO: implement build
38   return Scaffold(
39     appBar: AppBar(title: Text("Google Map")),
40     body: Container(child: GoogleMap(markers: { // เรียกใช้ API Google Map
41       Marker( // สร้าง Marker ลงบนตำแหน่งที่เราเรียก
42         markerId: MarkerId("1"),
43         position: LatLng(double.parse(latitude.toString()),double.parse(longitude.toString())),
44         // ปัก Marker ลงบนตำแหน่ง ละติจูด ลองจิจูด ที่เราเรียกออกมาจากค่าเดิมสละโดยค่าที่เราได้ออแรกมีค่าเป็น String จึงต้องแปลงค่าให้เป็น double ก่อน
45       ), // Marker
46     },
47     ),
48     mapType: MapType.normal, // เช็กรูปแบบของ Map ให้เป็นแบบปกติ
49     initialCameraPosition: CameraPosition(target:LatLng(double.parse(latitude.toString()),double.parse(longitude.toString())),
50     // ตำแหน่งของกล้องจะเริ่มที่ ละติจูด ลองจิจูด ที่เราเรียกมาจากค่าเดิม จะเป็นตำแหน่งเดียวกันกับที่ marker อยู่
51     zoom: 15.5,
52     )), // CameraPosition // GoogleMap // Container
53     floatingActionButton: FloatingActionButton.extended(
54       onPressed: (){
55       },
56       label: Text("My location"),
57       icon: Icon(Icons.near_me),
58     ), // FloatingActionButton.extended
59   ); // Scaffold
60 }
61

```

ภาพประกอบที่ 3.17 การเรียกใช้ข้อมูลละติจูดลองจิจูดที่ส่งมา

บรรทัดที่ 40 ติดตั้ง เรียกใช้ API google map

บรรทัดที่ 43 คำสั่งปัก Marker ลงบนตำแหน่ง ละติจูด ลองจิจูด ที่เราทำการเรียกออกมาจาก Database ตอนแรกมีค่าเป็น String จึงต้องแปลงค่าให้เป็น double ก่อน

บรรทัดที่ 48 เช็กรูปแบบของ Map ให้เป็นปกติ

บรรทัดที่ 49 ตำแหน่งของกล้องจะเริ่มที่ ละติจูด ลองจิจูด ที่เราเรียกมาจาก Database จะเป็นตำแหน่งเดียวกันกับที่ Marker อยู่

3.5.4 การสมัครสมาชิก

```

21 class registerState extends State<register>{
22
23   TextEditingController username,password,name,email,phone; //ประกาศตัวแปรเพื่อใช้ในการควบคุม text
24
25
26 @override
27 void initState() {
28   super.initState();
29   username = TextEditingController();
30   password = TextEditingController();
31   name = TextEditingController();
32   email = TextEditingController();
33   phone = TextEditingController();
34 }
35

```

ภาพประกอบที่ 3.18 การประกาศตัวแปรเพื่อใช้ในการควบคุม Text หน้าสมัครสมาชิก

บรรทัดที่ 23 การประกาศตัวแปรเพื่อใช้ในการควบคุม Text หน้าสมัครสมาชิก

```

85
86     textFiled("กรอก Username",username),
87     textFiled("กรอก Password",password),
88     textFiled("กรอก ชื่อ-นามสกุล",name),
89     textFiled("กรอก Email",email),
90     textFiled("กรอก เบอร์โทรศัพท์",phone),
91     // textFiled ใส่ตัวแปรที่ประกาศไว้ เพื่อที่จะได้เก็บค่าที่เขียนลงใน textFiled ลงในตัวแปร
92
93
94     Row(
95         mainAxisAlignment: MainAxisAlignment.center,
96         children: <Widget>[
97             RaisedButton(
98                 child: Text('ยืนยัน'),
99                 onPressed: () {
100                     RegisterUser(username.text, password.text,name.text,email.text,phone.text);
101                     //เมื่อกดปุ่ม ยืนยัน ให้ส่งค่าไปที่เมธอด RegisterUser พร้อมกับเรียกใช้งานเมธอด RegisterUser
102                     Navigator.pop(context);
103                 },

```

ภาพประกอบที่ 3.19 การใช้งานText ที่ประกาศตัวแปรไว้ หน้าสมัครสมาชิก

บรรทัดที่ 86-90 การใช้งาน TextFiled ใส่ตัวแปรที่ประกาศไว้ เพื่อที่จะได้เก็บค่าที่เขียนลงใน TextFiled ลงในตัวแปร

บรรทัดที่ 100 เมื่อกดปุ่มยืนยัน ให้ส่งค่าไปที่เมธอด RegisterUser พร้อมกับเรียกใช้งาน เมธอด RegisterUser

```

120
121 void RegisterUser(String username,String password,String name,String email,String phone){
122     //var url = "https://projectsavemycar.000webhostapp.com/register.php";
123     var url = "http://savemycarproject.cocopatch.com/user.php";//เก็บลิงค์ url ที่เป็นโค้ด php ที่ใช้ติดต่อกับ sql ไว้ในตัวแปร
124     var data = {"username":username,"password":password,"name":name,"email":email,"phone":phone};//ข้อมูลที่จะส่งเข้าไปใน sql
125     var res = http.post(url,body: data); //ทำการส่งข้อมูลไปที่ server
126 }
127
128

```

ภาพประกอบที่ 3.20 การใช้งานเมธอด RegisterUser

บรรทัดที่ 123 เก็บลิงค์ url ที่เป็นโค้ด PHP ที่ใช้ติดต่อกับ Sql ไว้ในตัวแปร

บรรทัดที่ 124 ข้อมูลที่จะส่งเข้าไปใน Sql

บรรทัดที่ 125 ทำการส่งข้อมูลไปที่ server

ภาพประกอบที่ 3.21 หน้าแสดงการสมัครสมาชิก

3.5.5 การแก้ไขข้อมูลผู้ใช้

```

34   TextEditingController username,password,name,email,phone; //ประกาศตัวแปรเพื่อใช้ในการควบคุม text
35   String keepPhone;
36   @override
37   void initState() {
38       // TODO: implement initState
39       super.initState(
40       );
41       username = TextEditingController();
42       password = TextEditingController();
43       name = TextEditingController();
44       email = TextEditingController();
45       phone = TextEditingController();
46
47   }

```

ภาพประกอบที่ 3.22 การประกาศตัวแปรเพื่อใช้ในการควบคุม Text หน้าแก้ไขข้อมูลผู้ใช้

บรรทัดที่ 34 การประกาศตัวแปรเพื่อใช้ในการควบคุม Text หน้าแก้ไขข้อมูลผู้ใช้

```

63 Text(args),
64 textFiled("กรอก ชื่อ-นามสกุล",name),
65 textFiled("กรอก Email",email),
66 textFiled("กรอก เบอร์โทรศัพท์",phone),
67 // textFiled ใส่ตัวแปรที่ประกาศไว้ เพื่อที่จะได้เก็บค่าที่เขียนลงใน textFiled ลงในตัวแปร
68
69 Row(
70   mainAxisAlignment: MainAxisAlignment.center,
71   children: <Widget>[
72     RaisedButton(
73       child: Text('ยืนยัน'),
74       onPressed: () {
75         UpdateUser(args,name.text,email.text,phone.text);
76         //เรียกใช้งานเมธอด UpdateUser และส่งค่าเข้าไปในเมธอด
77         print("กดแล้ว");
78       },

```

ภาพประกอบที่ 3.23 การใช้งานText ที่ประกาศตัวแปรไว้ หน้าแก้ไขข้อมูลผู้ใช้

บรรทัดที่ 64-66 การใช้งาน TextFiled ใส่ตัวแปรที่ประกาศไว้ เพื่อที่จะได้เก็บค่าที่เขียนลงใน TextFiled ลงในตัวแปร

บรรทัดที่ 74 เมื่อกดปุ่มยืนยัน เรียกใช้งานเมธอด UpdateUser พร้อมกับให้ส่งค่าไปที่เมธอด

```

95 void UpdateUser(String args,String name,String email,String phone){
96
97   var url = "http://savemycarproject.cocopatch.com/updateProfile.php"; //เก็บลิ้ง url ที่เป็นโค้ด php ที่ใช้ติดต่อกับ sql ไว้ในตัวแปร
98   var data = {"id":args,"name":name,"email":email,"phone":phone}; //ข้อมูลที่จะส่งเข้าไปใน sql เพื่อแก้ไขโดยข้อมูลมาจาก textFiled ที่รับเข้ามา
99
100  var res = http.post(url,body: data); //ทำการส่งข้อมูลไปที่ server
101
102 }

```

ภาพประกอบที่ 3.24 การใช้งานเมธอด UpdateUser

บรรทัดที่ 97 เก็บลิ้ง url ที่เป็นโค้ด PHP ที่ใช้ติดต่อกับ Sql ไว้ในตัวแปร

บรรทัดที่ 98 ข้อมูลที่จะส่งเข้าไปใน Sql เพื่อแก้ไขโดยข้อมูลมาจาก textFiled ที่รับเข้ามา

บรรทัดที่ 125 ทำการส่งข้อมูลไปที่ server