



การประยุกต์ระบบจัดการข้อมูล และเหตุการณ์จำเพาะ เพื่อระบบติดตาม
ยานพาหนะ

โดย

นาย เกียรติพัฒน์ วงล่อง

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

การประยุกต์ระบบจัดการข้อมูล และเหตุการณ์จำเพาะ เพื่อระบบติดตาม
ยานพาหนะ

โดย

นาย เกียรติพัฒน์ วงล่อง

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

Application of data management system and management of
specific events for vehicle tracking systems

BY

Mr. Kiattipat Wonglong

A FINAL-YEAR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2024

COPYRIGHT OF THAMMASAT UNIVERSITY

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย เกียรติพัฒน์ วงล่อง

เรื่อง

การประยุกต์ระบบจัดการข้อมูล และเหตุการณ์จำเพาะ เพื่อระบบติดตามยานพาหนะ

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

เมื่อ วันที่ วันที่ ชื่อเต็มของเดือน พ.ศ. 2567

อาจารย์ที่ปรึกษา

ทรงศักดิ์ ร่องวิริยะพานิช

(ผู้ช่วยศาสตราจารย์ ดร.ทรงศักดิ์ ร่องวิริยะพานิช)

กรรมการสอบโครงการพิเศษ

นุชชากร งามเสาวรส

(อาจารย์ นุชชากร งามเสาวรส)

กรรมการสอบโครงการพิเศษ

ศานตนาฏ กิจศิริานุวัตร

(ผู้ช่วยศาสตราจารย์ ดร. ศานตนาฏ กิจศิริานุวัตร)

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย เกียรติพัฒน์ วงษ์

เรื่อง

การประยุกต์ระบบจัดการข้อมูล และเหตุการณ์จำเพาะ เพื่อระบบติดตามยานพาหนะ

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

เมื่อ วันที่ วันที่ ชื่อเต็มของเดือน พ.ศ. 2567

อาจารย์ที่ปรึกษา



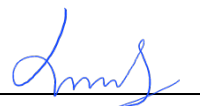
(ผู้ช่วยศาสตราจารย์ ดร.ทรงศักดิ์ ร่องวิริยะพานิช)

กรรมการสอบโครงการพิเศษ



(อาจารย์ นุชชากร งามเสาวรส)

กรรมการสอบโครงการพิเศษ



(ผู้ช่วยศาสตราจารย์ ดร. ศาตนาฏ กิจศิริานุวัตร)

หัวข้อโครงการพิเศษ

การประยุกต์ระบบจัดการข้อมูล และเหตุการณ์จำเพาะ
เพื่อระบบติดตามยานพาหนะ

ชื่อผู้เขียน

นาย เกียรติพัฒน์ วงล่อง

ชื่อปริญญา

วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

สาขาวิชา/คณะ/มหาวิทยาลัย

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์

อาจารย์ที่ปรึกษาโครงการพิเศษ

ผู้ช่วยศาสตราจารย์ ดร. ทรงศักดิ์ รองวิริยะพานิช

ปีการศึกษา

2567

บทคัดย่อ

โครงการพิเศษฉบับนี้มีวัตถุประสงค์เพื่อศึกษาและพัฒนาาระบบจัดการข้อมูลด้านโลจิสติกส์ โดยมุ่งเน้นการดูแลและบริหารจัดการข้อมูลอย่างมีประสิทธิภาพ เพื่อส่งเสริมขีดความสามารถของระบบขนส่งในภาคธุรกิจในการขนส่งทางภาค ธุรกิจ ซึ่งมีการขนส่ง สินค้าหรือทรัพยากรต่างๆ ผ่านทางบก ด้วยยานพาหนะประเภทรถยนต์ ที่ต้องมีการใช้เส้นทางในการขนส่ง และการกระทำบางอย่างอาจทำให้เกิดความเสี่ยง หรือเหตุการณ์เฉพาะ ที่ทำให้เกิดความเสียหาย ทางด้านความปลอดภัย หรือทรัพย์สินทางธุรกิจได้

ภายในช่วงระหว่างการทำโครงการนี้ได้มีการเข้าไปศึกษารวมถึงการพัฒนาระบบเพื่อใช้ในการติดตามและจัดการข้อมูล ของยานพาหนะภายใน บริษัทขนส่ง ทำให้สามารถเข้าใจถึงการทำงาน และความสำคัญในด้านต่างๆ ของข้อมูล ที่จะต้องถูกกำหนด และ ส่งผ่านยานพาหนะ เพื่อใช้ในการประมวลผลเหตุการณ์จำเพาะภายในระบบ

Thesis Title	Application of data management system and management of specific events for vehicle tracking systems
Author	Mr. Kiattipat Wonglong
Degree	Bachelor of Science
Major Field/Faculty/University	Computer Science Faculty of Science and Technology Thammasat University
Project Advisor	Assistant Professor Dr. Songsakdi Rongviriyapanis
Academic Years	2024

ABSTRACT

This special project is to study data and guidelines for creating a management system, taking care of and managing logistics data to help promote efficiency in business transportation, which involves transporting goods or resources over land using vehicles such as cars that require transportation routes and certain actions may cause risks or specific events that can damage safety or business assets.

During the implementation of this project, a system for tracking and managing vehicle data within the transportation company was studied and developed, enabling an understanding of the operation and importance of various aspects of data that must be defined and transmitted through vehicles for processing specific events within the system.

กิตติกรรมประกาศ

โครงการพิเศษระดับปริญญาตรี เรื่อง การประยุกต์ระบบจัดการข้อมูล และเหตุการณ์
จำเพาะ เพื่อระบบติดตามยานพาหนะสามารถดำเนินได้ด้วยดีเนื่องจากได้รับความอนุสงค์และการ
สนับสนุน อย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร. ทรงศักดิ์ ร่องวิริยะพานิช ที่ได้กรุณาให้คำปรึกษา
ความรู้ ข้อคิด คำแนะนำ และปรับปรุงแก้ไขข้อบกพร่องต่างๆ

ขอบคุณบริษัท TWS และทีมพัฒนา ที่ช่วยแนะนำ แนวทาง และคำแนะนำการออกแบบ
ระบบ การติดตามยานพาหนะ

นายเกียรติพัฒน์ วงล่อง

สารบัญ

	หน้า
บทคัดย่อ	1
ABSTRACT	2
กิตติกรรมประกาศ	3
สารบัญ	4
สารบัญตาราง	8
สารบัญภาพ	9
รายการสัญลักษณ์และคำย่อ	9
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	3
1.4 ประโยชน์ของโครงการ	4
1.5 ข้อจำกัดของโครงการ	5
บทที่ 2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	6
2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง	6
2.1.1 แนวคิดการติดตามและรับข้อมูล	6
2.1.2 แนวคิดการเก็บข้อมูลที่จำเป็นต่อการวิเคราะห์	7

	(5)
2.1.3 แนวคิดการออกแบบเหตุการณ์จำเพาะ	9
2.2 งานวิจัยที่เกี่ยวข้อง	11
2.2.1 การเปรียบเทียบ Redis กับฐานข้อมูลอื่น	12
บทที่ 3 วิธีการดำเนินการวิจัย	15
3.1 ภาพรวมของโครงการ	15
3.1.1 เครื่องมือที่เกี่ยวข้องกับการทำโครงการ	17
3.1.2 การออกแบบระบบแยกเหตุการณ์จำเพาะ	17
3.1.3 การออกแบบระบบฐานข้อมูล	20
3.2 การวิเคราะห์ขอบเขตและความต้องการของระบบ	31
3.3 ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย	32
3.4 ผลลัพธ์ที่คาดหวัง	32
3.5 ผลลัพธ์	32
บทที่ 4 ทฤษฎีการและแผนการดำเนินงาน	53
4.1 การจัดเตรียมฮาร์ดแวร์และซอฟต์แวร์	53
4.1.1 React.JS	53
4.1.2 NodeRed	53
4.1.3 Redis	53
4.1.4 Pandas Library	53
4.1.5 MSQl	53
4.2 แผนการดำเนินงาน	54
บทที่ 5 สรุป	55
รายการอ้างอิง	56

(6)

ภาคผนวก

58

ภาคผนวก ก. การเปรียบเทียบการทำงานระหว่าง Redis กับฐานข้อมูล

59

สารบัญตาราง

	หน้า
ตารางที่ 3.1 ระบบที่ใช้ภายในระบบการจัดการข้อมูล	20 - 21
ตารางที่ 3.2 ระบบที่ใช้ภายในระบบการจัดการข้อมูลแบบสรุป	21
ตารางที่ 3.3 Data dictionary ของ vehicle	34
ตารางที่ 3.4 Data dictionary ของ vehicle_type	35
ตารางที่ 3.5 Data dictionary ของ vehicle_driver	35
ตารางที่ 3.6 Data dictionary ของ vehicle_in_group	36
ตารางที่ 3.7 Data dictionary ของ subcontract_company	36
ตารางที่ 3.8 Data dictionary ของ mikrotik	37
ตารางที่ 3.9 Data dictionary ของ iot_board	38
ตารางที่ 3.10 Data dictionary ของ nvr	38
ตารางที่ 3.11 Data dictionary ของ provinces	39
ตารางที่ 3.12 Data dictionary ของ driver	41
ตารางที่ 3.13 Data dictionary ของ vehicle_driver	41
ตารางที่ 3.14 Data dictionary ของ station	42
ตารางที่ 3.15 Data dictionary ของ station_type	43
ตารางที่ 3.16 Data dictionary ของ aministrative_area	43
ตารางที่ 3.17 Data dictionary ของ aministrative_area	46
ตารางที่ 3.18 Data dictionary ของ danger_area_group	47
ตารางที่ 3.19 Data dictionary ของ danger_area	47
ตารางที่ 3.20 Data dictionary ของ danger_area_in_group	48
ตารางที่ 4.1 การดำเนินงานที่ผ่านมาจนถึงปัจจุบัน	54

สารบัญภาพ

	หน้า
ภาพที่ 2.1 ระบบที่ติดตั้งภายในตัวรถ	7
ภาพที่ 2.2 แสดงตัวอย่างของไอโอทีบอร์ดที่ใช้สำหรับการรับส่งและจัดเก็บข้อมูลตำแหน่ง	8
ภาพที่ 2.3 เปรียบเทียบเวลาในการ select (ดึงข้อมูล)	12
ภาพที่ 2.4 เปรียบเทียบเวลาในการ insert (เพิ่มข้อมูล)	12
ภาพที่ 2.5 ภาพการวัดประสิทธิภาพการ update ภาพระหว่างฐานข้อมูล	13
ภาพที่ 3.1 การออกแบบระบบการจัดการข้อมูลระหว่างรถขนส่งกับServer	14
ภาพที่ 3.2 การส่งข้อมูลจากกล่อง S.box สู่อุปกรณ์ที่ภาพ และ Server ประมวลผล	16
ภาพที่ 3.3 Data Flow ของระบบ EventClassifier	17
ภาพที่ 3.4 การออกแบบระบบการจัดการข้อมูล	18
ภาพที่ 3.5 ภาพ flow ของระบบที่รับ payload เพื่อใช้ในการประมวลผล	21
ภาพที่ 3.6 ภาพ flow การรับค่าจาก IoT ภายใน NodeRed	22
ภาพที่ 3.7 ภาพ flow การคัดแยกเหตุการณ์เฉพาะภายใน NodeRed	23
ภาพที่ 3.8 ภาพการหาตำแหน่งบนรูปหลายเหลี่ยมโดยวิธีการ Ray Casting Algorithm	26
ภาพที่ 3.9 ภาพ code ฟังก์ชัน isPointInPolygon ที่ถูกใช้ภายใน Node-RED	27
ภาพที่ 3.10 ภาพต่อของ code ฟังก์ชัน isPointInPolygon ที่ถูกใช้ภายใน Node-RED	28
ภาพที่ 3.11 ภาพ code ฟังก์ชันเทียบกับข้อมูล Event ที่ถูกใช้ภายใน Node-RED	30
ภาพที่ 3.12 ภาพ UI การแก้ไขเหตุการณ์เฉพาะ ภายใน React	31
ภาพที่ 3.13 ภาพการออกแบบข้อมูลภายใน SQL ของรถขนส่ง	32
ภาพที่ 3.14 ภาพการออกแบบข้อมูลภายใน SQL ของผู้ขับขี่	39
ภาพที่ 3.15 ภาพการออกแบบข้อมูลภายใน SQL ของจุดจัดส่ง	41
ภาพที่ 3.16 ภาพการออกแบบข้อมูลภายใน SQL ของพื้นที่เสี่ยง	44
ภาพที่ 3.17 ภาพ UI หน้าตารางจัดการข้อมูลรถภายใน React	48
ภาพที่ 3.18 ภาพ UI หน้าข้อมูลรถภายใน React	49
ภาพที่ 3.19 ภาพ UI หน้าแก้ไขข้อมูลรถภายใน React	50
ภาพที่ 3.20 ภาพ UI หน้าตารางจัดการข้อมูลที่ถูกลบใน React	50
ภาพที่ 3.21 ภาพ UI หน้าตารางจัดการข้อมูลประวัติภายใน React	51
ภาพที่ 3.22 ภาพ UI หน้าตารางจัดการข้อมูลเหตุการณ์เฉพาะภายใน React	51
ภาพที่ 3.23 ภาพ UI แสดงหน้าต่าง แผนที่ตำแหน่งของรถในปัจจุบัน	52

รายการสัญลักษณ์และคำย่อ

สัญลักษณ์/คำย่อ	คำเต็ม/คำจำกัดความ
IoT	Internet of Things อุปกรณ์ที่เชื่อมต่อถึงกันและเทคโนโลยีที่อำนวยความสะดวกในการสื่อสารระหว่างอุปกรณ์กับระบบคลาวด์ตลอดจนระหว่างอุปกรณ์ด้วยตัวเอง
SQL	Structured Query Language โปรแกรมสำหรับจัดเก็บและประมวลผลข้อมูลในฐานข้อมูลแบบเชิงสัมพันธ์ ฐานข้อมูลแบบเชิงสัมพันธ์เก็บข้อมูลในรูปแบบตารางที่มีแถวและคอลัมน์ที่เป็นตัวแทนของหมวดข้อมูลที่แตกต่างกันและความสัมพันธ์ต่างๆ ระหว่างค่าข้อมูล
UI	User Interface ส่วนที่เชื่อมถึงผู้ใช้งาน
Specific Event	เหตุการณ์จำเพาะที่เกิดขึ้นโดยมีการสื่อสารความหมายทางธุรกิจ
ไลบรารี	โปรแกรมในคอมพิวเตอร์เก็บรวบรวมคำสั่งเฉพาะต่างๆ
API	กลไกที่ช่วยให้ส่วนประกอบซอฟต์แวร์สองส่วนสามารถสื่อสารกันได้โดยใช้ชุดคำจำกัดความและโปรโตคอล
Ray-casting	อัลกอริทึมในเชิงเรขาคณิตที่ใช้สำหรับตรวจสอบว่าจุดหนึ่งอยู่ภายในหรือภายนอกของรูปหลายเหลี่ยม (Polygon) โดยเฉพาะในกรณีที่ทำงานกับแผนที่
Geographic Fence (GeoFence)	พื้นที่เสมือนบนแผนที่ที่ถูกกำหนดขึ้นโดยใช้พิกัดตำแหน่ง (Latitude, Longitude) เพื่อควบคุมหรือระบุขอบเขตของพื้นที่หนึ่งๆ

Latitude	ค่าที่บอกตำแหน่ง เหนือหรือใต้ จากเส้นศูนย์สูตร (Equator) โดย ค่าอยู่ระหว่าง -90° ถึง $+90^{\circ}$
Longitude	ค่าที่บอกตำแหน่ง ตะวันออกหรือตะวันตก จากเส้น Prime Meridian (เส้นเมริเดียนที่ 0° ผ่านเมืองกรีนิช ประเทศอังกฤษ) ค่าอยู่ระหว่าง -180° ถึง $+180^{\circ}$
Soft Delete	แนวทางการ “ลบข้อมูลแบบไม่ลบทิ้งจริง” จากฐานข้อมูล โดยจะ ไม่ลบแถว (row) ออกจากตาราง, แต่จะใช้ คอลัมน์พิเศษ เพื่อระบุว่าสถานะข้อมูลนั้นถูกลบแล้วหรือไม่

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ปัจจุบันรัฐบาลไทย มีเป้าหมายให้ประเทศไทยเป็นศูนย์กลางระบบโลจิสติกส์ ของภูมิภาคอาเซียน ด้วยศักยภาพการแข่งขันของประเทศไทย ประกอบกับตำแหน่งที่ตั้งทางภูมิศาสตร์ที่เป็นศูนย์กลางของอาเซียน ทำให้รองรับการขนส่งทั้งทางน้ำ ทางบก และทางอากาศ ที่มีโครงสร้างพื้นฐานรองรับอย่างเพียงพอ จึงเกิดโครงการพัฒนาระบบขนส่งต่าง ๆ ตามมาอีกมากมาย ทั้งจากการลงทุนโดยภาครัฐ และเอกชน

ในภาคเอกชน มีการแข่งขันการบริการงานด้านโลจิสติกส์ อย่างกว้างขวาง ดังจะเห็นได้จากการคุณภาพของการบริการด้านการขนส่ง ดีขึ้น ตามลำดับ สวนทางกับอัตราค่าบริการที่ถูกลง เป็นตัวชี้วัดการเติบโตของอุตสาหกรรมโลจิสติกส์ในประเทศไทย อย่างชัดเจน

เพื่อควบคุมการเติบโตของการขนส่งทางบก กรมการขนส่งทางบก มีหน้าที่รับผิดชอบ เกี่ยวกับการควบคุมดูแลกิจการขนส่งสินค้าภายในประเทศและระหว่างประเทศ ได้ออกข้อกำหนดของการตรวจติดตามรถขนส่งที่ บังคับให้รถขนส่งจะต้องมีระบบตรวจติดตาม ระบุตำแหน่งของรถขนส่ง เพื่อให้สามารถควบคุม และตรวจติดตาม รถขนส่ง ให้ปฏิบัติตามกฎเกณฑ์การใช้ทางหลวงร่วมกันอย่างถูกต้อง

ระบบตรวจติดตามรถขนส่ง โดยใช้ GPS (Global Positioning System) จึงได้รับความสนใจเป็นพิเศษ จากเอกชนผู้ให้บริการขนส่งโลจิสติกส์ โดยเฉพาะการขนส่งทางบก

การระบุตำแหน่งรถขนส่งโดย GPS ให้ความถูกต้องในระดับหนึ่ง แต่ยังไม่ครอบคลุมถึงการตรวจจับพฤติกรรมการใช้รถใช้ถนนที่ถูกต้องได้ทั้งหมด และนอกจากนี้ ยังไม่เพียงพอต่อการป้องกันการทุจริตของพนักงานขับรถ ต่อบริษัทขนส่ง และลูกค้า ได้ ปัจจุบันจึงมีการพัฒนาระบบเซ็นเซอร์ตรวจสอบสภาพของรถขนส่งโดยรอบ เช่น ระบบตรวจจับอุณหภูมิ ระบบตรวจจับการเปิดปิดประตูห้องสรรพาระ ระบบตรวจจับการเกิดอุบัติเหตุรถพลิกคว่ำ เป็นต้น

นอกจากระบบเซ็นเซอร์ทั่วไปแล้ว ระบบประมวลผลภาพ โดยการติดกล้องบันทึกภาพเรียลไทม์รอบรถ เป็นอีกหนึ่งระบบที่ได้รับความนิยมเป็นอย่างมาก เนื่องจาก ภาพที่ได้จากการบันทึกแบบเรียลไทม์ สามารถใช้เป็นประจักษ์พยานในทางกฎหมายได้

ทั้งระบบเซ็นเซอร์ และระบบประมวลผลการบันทึกภาพเรียลไทม์ ต้องการระบบการประมวลผล เพื่อระบุถึง “เหตุการณ์จำเพาะ” (Specific Event) ที่สามารถตอบโจทย์ทางธุรกิจโลจิสติกส์ ได้ เช่น การตรวจจับพฤติกรรมขับรถเร็วเกินกำหนด การเปลี่ยนช่องทางจราจรกระทัดหัน หรือ การจอดรถในที่ห้ามจอด เป็นต้น

1.2 วัตถุประสงค์

วัตถุประสงค์ของโครงการฉบับนี้คือ การพัฒนาระบบประมวลผลเหตุการณ์จำเพาะ โดยอาศัยข้อมูลจากเซ็นเซอร์และระบบบันทึกภาพแบบเรียลไทม์ เพื่อวิเคราะห์และตรวจจับพฤติกรรมการขับขี่ที่อาจเสี่ยงต่อความปลอดภัยหรือขัดต่อข้อกำหนดด้านกฎหมายการจราจรเพื่อตรวจจับเหตุการณ์จำเพาะ ที่เป็นการหลีกเลี่ยง หรือ ละเมิดกฎหมายการใช้รถใช้ถนน หรือ กฎการขับขี่เพื่อความปลอดภัยของบริษัทเอกชนที่ประกอบธุรกิจในอุตสาหกรรมโลจิสติกส์ ได้อย่างมีประสิทธิภาพ

ความต้องการของภาคธุรกิจคือการตรวจสอบ ติดตาม และวิเคราะห์ประสิทธิภาพ ของข้อมูลการขนส่ง และติดตามการเดินรถ รวมทั้งพฤติกรรมของพนักงาน เพื่อวัดประสิทธิภาพต่อเงินลงทุน โดยภาคธุรกิจสามารถนำไป ประมวลผล และประกอบการตัดสินใจ ในการวางแผนการจัดการ การจัดการเดินรถ หรือจัดการพนักงานขับรถ สภาพปัจจุบันของรถ สภาพการจราจร รวมถึงข้อมูลอุณหภูมิห้องสรวพาและปริมาณเชื้อเพลิง

ระบบประมวลผลเหตุการณ์จำเพาะ โดยการนำข้อมูลเซ็นเซอร์ที่ได้ มาแปลงและกำหนดกฎ ซึ่งเป็นไปตามที่ภาคธุรกิจกำหนด เพื่อให้ภาคธุรกิจสามารถติดตามและบันทึกเหตุการณ์จำเพาะที่เกิดขึ้น ในแต่ละช่วงเวลา เพื่อวิเคราะห์และประมวลผล ถึงประสิทธิภาพในการทำงาน ของเส้นทาง การเดินรถ ตามความต้องการทางธุรกิจที่กล่าวมาข้างต้น เมื่อระบบประมวลผลเหตุการณ์จำเพาะตรวจพบเหตุการณ์ที่ตรงกับเงื่อนไขที่กำหนด

ระบบแจ้งเตือนภาคธุรกิจ เมื่อผู้ขับขี่ หรือรถ เกิดเหตุการณ์ที่เสี่ยงจะเป็นอันตราย หรือทำให้เกิดการละเมิดกฎหมายการใช้รถใช้ถนน ภาคธุรกิจจะได้รับการแจ้งเตือน ถึงเหตุการณ์ที่กำลังเกิดขึ้นอย่างถูกต้อง

การขนส่งทางบกจะเพิ่มประสิทธิภาพการขนส่งมากขึ้น และทางภาคธุรกิจเอกชนในอุตสาหกรรมการขนส่งสามารถจัดการการขนส่งได้อย่างเป็นระเบียบและเข้าถึงง่าย โดยเป็นการยกระดับการขนส่งภายในประเทศไทย ให้มีประสิทธิภาพยิ่งขึ้น

1.3 ขอบเขตของโครงการ

ผู้ขับขี่ ขับรถอย่างปลอดภัย และถูกต้องตามกฎหมาย: การควบคุมเส้นทางการเดินทางของผู้ขับขี่ โดยการนำข้อมูลจากเซ็นเซอร์ และจีพีเอส มาวิเคราะห์ โดยจะกำหนดขอบเขต ควบคุมความเร็ว รวมถึงกำหนดพื้นที่อันตราย เพื่อไม่ให้ผู้ขับขี่ ขับรถเข้าไปในพื้นที่เสี่ยง หรือพื้นที่ที่กำลังเกิดอุบัติเหตุ และการกำหนดการปฏิบัติการขับขี่ เพื่อให้เป็นการขับขี่ที่ปลอดภัยต่อตัวผู้ขับขี่ และผู้ร่วมเดินทาง บนท้องถนน โดยตรวจสอบพฤติกรรม ว่าผู้ขับขี่ไม่ได้มีพฤติกรรมเสี่ยง เช่น การเปลี่ยนเลนกะทันหัน การขับขี่ที่เร็วเกินกำหนด เป็นต้น

ผู้ประกอบการ สามารถตรวจติดตาม ประสิทธิภาพการทำงานของรถขนส่ง และพนักงานขับรถได้: สามารถติดตามสถานะ ของรถและพนักงานขับรถในปัจจุบัน ว่ามีการใช้งาน อยู่ภายในพื้นที่ได้ หรือกำลังจะถึงสถานที่ขนส่ง ด้วยระบบติดตามผ่านแผนที่ ซึ่งเป็นการแสดงตำแหน่งรถในปัจจุบัน รวมทั้งแสดงถึงข้อมูลรถ สถานะปัจจุบันของรถ และสถานะของคนขับ ซึ่งสามารถดูได้ว่า มีการพักการแวะสถานที่พักผ่อน อย่างเพียงพอ ผ่านทางตัวแอปพลิเคชันติดตามแบบเรียลไทม์ ที่จะอัปเดตสถานะข้อมูลที่ได้จากเซ็นเซอร์ และจีพีเอสภายในรถ นำมาแสดงให้ภาคธุรกิจได้เห็นภายในเวลาใกล้เคียงกันมากที่สุด เพื่อความปลอดภัยและ ประสิทธิภาพในการติดตามการทำงานของรถขนส่ง และพนักงาน

หน่วยธุรกิจสามารถ ตรวจติดตาม การใช้น้ำมันเชื้อเพลิงที่เป็นต้นทุนแปรผันหลัก ในธุรกิจ: การใช้เซ็นเซอร์ตรวจจับของเหลวภายใน ถังน้ำมันของตัวรถ ซึ่งจะเป็นการวัดระดับน้ำมันเชื้อเพลิงในปัจจุบัน โดยนำไปเปรียบเทียบกับปริมาณน้ำมันภายในอดีตที่มีการบันทึกไว้ในตัวแอปพลิเคชัน เพื่อตรวจสอบว่า ไม่มีการทุจริตของตัวพนักงาน หรือผู้ร่วมกระทำผิด ในการขโมยน้ำมันเชื้อเพลิงซึ่งถือเป็นต้นทุนของภาคธุรกิจ ไปหาผลประโยชน์โดยมิชอบ

หน่วยธุรกิจ สามารถ ป้องกันความเสี่ยงของสรรพากรในท้องถิ่น: การใช้เซ็นเซอร์ตรวจจับความเย็น ภายในสัมภาระของรถขนส่ง สามารถตรวจสอบได้ว่าอุณหภูมินั้นมีความเสี่ยงต่อสินค้าหรือไม่

1.4 ประโยชน์ของโครงการ

โครงการนี้มีประโยชน์ทั้งในเชิงเทคโนโลยีและเชิงธุรกิจ โดยเฉพาะในด้านการจัดการโลจิสติกส์ การติดตามยานพาหนะ และการวิเคราะห์เหตุการณ์จำเพาะ โดยสามารถสรุปประโยชน์ที่ได้รับจากการพัฒนาโครงการได้ดังต่อไปนี้:

1. การวิเคราะห์ข้อมูลดิบจากเซ็นเซอร์เพื่อแปลงเป็นเหตุการณ์จำเพาะที่มีความหมาย

ระบบสามารถนำข้อมูลดิบจากอุปกรณ์เซ็นเซอร์ที่ติดตั้งในยานพาหนะ เช่น ข้อมูลความเร็วระดับน้ำมัน หรืออุณหภูมิ มาแปลงเป็นเหตุการณ์จำเพาะที่มีความหมายในเชิงธุรกิจ ด้วยการประมวลผลผ่านตรรกะและเงื่อนไขที่กำหนดไว้ เช่น การขับรถเร็วเกินกำหนดในเขตชุมชน หรืออุณหภูมิที่สูงเกินมาตรฐานในห้องเก็บสินค้า การแปลงข้อมูลให้มีความสำคัญเชิงพฤติกรรมเช่นนี้ เอื้อต่อการประเมินความเสี่ยง การวางแผน การจัดการ และการตัดสินใจเชิงกลยุทธ์ในระบบโลจิสติกส์ได้อย่างมีประสิทธิภาพ

2. การออกแบบและประยุกต์ใช้ฐานข้อมูลเพื่อรองรับข้อมูลเหตุการณ์จำเพาะ

ระบบมีการออกแบบโครงสร้างฐานข้อมูลเพื่อจัดเก็บข้อมูลที่ได้รับจากเซ็นเซอร์ โดยมีการเชื่อมโยงข้อมูลระหว่างเซ็นเซอร์, ยานพาหนะ, ผู้ขับขี่ และเหตุการณ์จำเพาะอย่างเป็นระบบ รวมถึงการนำ Redis เข้ามาช่วยในการจัดเก็บและเรียกใช้งานข้อมูลเหตุการณ์ที่เกิดขึ้นแบบเรียลไทม์ ซึ่งช่วยเพิ่มประสิทธิภาพในการประมวลผลและลดความหน่วงของระบบ

3. การพัฒนาระบบแจ้งเตือนเหตุการณ์จำเพาะแบบเรียลไทม์

ระบบสามารถแสดงผลเหตุการณ์จำเพาะที่เกิดขึ้นในปัจจุบันให้ผู้ใช้งานสามารถรับทราบได้ทันที โดยผู้ใช้งานในภาคธุรกิจสามารถเลือกดูข้อมูลเฉพาะของยานพาหนะคันใดคันหนึ่ง หรือดูภาพรวมของยานพาหนะทั้งหมด เพื่อใช้ในการวางแผน ตรวจสอบ หรือตอบสนองต่อสถานการณ์ได้อย่างทันท่วงที

4. การเชื่อมโยงระหว่างฐานข้อมูลกับระบบวิเคราะห์ภายใน (Back-end Logic)

โครงการออกแบบให้ระบบสามารถเชื่อมต่อข้อมูลจากฐานข้อมูลกับตรรกะการวิเคราะห์เหตุการณ์จำเพาะ โดยดึงข้อมูลดิบจากเซ็นเซอร์มาประมวลผลอย่างเป็นระบบ ก่อนส่งข้อมูลผลลัพธ์ไปยังส่วนติดต่อผู้ใช้ (UI) เพื่อให้ผู้ใช้งานได้รับข้อมูลที่ถูกต้อง ครบถ้วน และตรงเวลา

5. การควบคุมการเข้าถึงข้อมูลเพื่อรักษาความปลอดภัยและจริยธรรมของระบบ

ระบบถูกออกแบบให้รองรับการกำหนดสิทธิ์ในการเข้าถึงข้อมูล โดยเฉพาะข้อมูลที่มีความละเอียดอ่อน เช่น ข้อมูลการขับขี่ หรือข้อมูลที่เกี่ยวข้องกับคดีความในกรณีที่เกิด

อุบัติเหตุ ผู้ใช้ที่ไม่มีสิทธิ์จะไม่สามารถเข้าถึงข้อมูลเหล่านี้ได้ ซึ่งช่วยป้องกันการดัดแปลง ใช้งานในทางที่ผิด หรือรั่วไหลของข้อมูลนี้อาจนำไปสู่ผลกระทบทางกฎหมาย

6. การประยุกต์ใช้ความรู้ทางทฤษฎีสู่การปฏิบัติ

โครงการนี้เป็นการบูรณาการระหว่างความรู้เชิงทฤษฎีที่ได้ศึกษามา เช่น การประมวลผลข้อมูล, การออกแบบระบบฐานข้อมูล, การวิเคราะห์เหตุการณ์ และการพัฒนาแอปพลิเคชัน ร่วมกับการปฏิบัติงานจริงในบริบทของการขนส่ง ซึ่งช่วยเสริมสร้างทักษะด้านเทคโนโลยีสารสนเทศและการวางแผนระบบในโลกธุรกิจได้อย่างรอบด้าน

1.5 ข้อจำกัดของโครงการ

ข้อจำกัดในการเข้าถึงข้อมูลภาคธุรกิจ

เนื่องจากข้อมูลที่เกี่ยวข้องกับการขนส่ง การติดตามยานพาหนะ และการจัดเก็บข้อมูลจากระบบ IoT มักมีความสำคัญในเชิงกลยุทธ์ขององค์กรธุรกิจ การเปิดเผยข้อมูลบางประเภทจึงอาจถูกจำกัดสำหรับบุคคลภายนอก ส่งผลให้ทีมพัฒนาไม่สามารถเข้าถึงข้อมูลจริงบางรายการเพื่อนำมาใช้ในการออกแบบ วิเคราะห์ หรือทดสอบระบบได้อย่างเต็มที่ ซึ่งอาจกระทบต่อความแม่นยำในการสร้างสถานการณ์จำลองและเหตุการณ์จำเพาะ

ความซับซ้อนในการตีความข้อมูลดิบให้เป็นเหตุการณ์จำเพาะ

การแปลงข้อมูลดิบที่ได้รับจากเซ็นเซอร์และอุปกรณ์ต่าง ๆ ให้เป็น “เหตุการณ์จำเพาะ” จำเป็นต้องใช้หลักเกณฑ์และเงื่อนไขที่หลากหลายและมีความซับซ้อนสูง ทั้งในเชิงเวลา พื้นที่ พฤติกรรมผู้ขับขี่ และลักษณะเฉพาะของยานพาหนะ การวิเคราะห์ที่ไม่รัดกุมอาจส่งผลให้เกิดการแจ้งเตือนผิดพลาด (false positive/false negative) ซึ่งจำกัดความสามารถของระบบในการรองรับสถานการณ์ที่หลากหลาย

ข้อจำกัดด้านฮาร์ดแวร์และสัญญาณในการรับ-ส่งข้อมูล

การประมวลผลข้อมูลจากอุปกรณ์ IoT จำเป็นต้องอาศัยความสามารถของฮาร์ดแวร์ เช่น ความเร็วในการประมวลผล ความจุของหน่วยความจำ และความสามารถในการเชื่อมต่อแบบไร้สาย หากฮาร์ดแวร์มีข้อจำกัด เช่น รับสัญญาณได้ไม่ดีในบางพื้นที่ หรือมีการส่งข้อมูลล่าช้า อาจทำให้ข้อมูลที่ได้อาจไม่สมบูรณ์หรือไม่ตรงตามเวลาจริง (real-time) ซึ่งอาจกระทบต่อประสิทธิภาพของระบบติดตาม และการวิเคราะห์เหตุการณ์จำเพาะ

บทที่ 2

วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

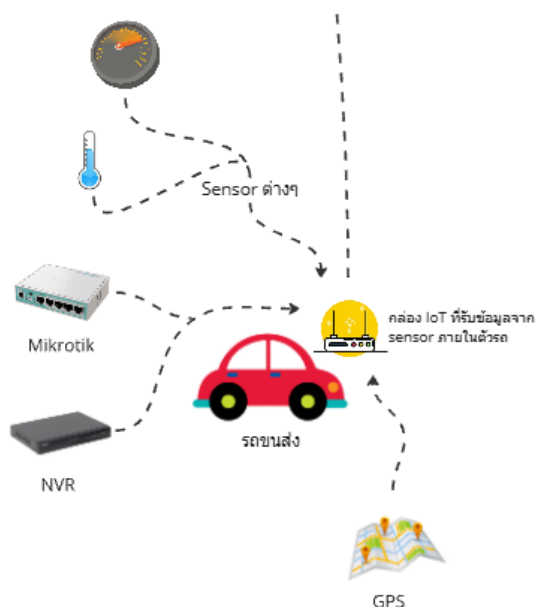
2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง

2.1.1 แนวคิดการติดตามและรับข้อมูล

2.1.1.1 แนวคิดการติดตามพาหนะ

ในการติดตามตำแหน่งของยานพาหนะอย่างแม่นยำ จำเป็นต้องมีการจัดเก็บและส่งผ่านข้อมูลสำคัญที่เกี่ยวข้องกับพิกัดตำแหน่ง ได้แก่ ค่าละติจูด (Latitude) และลองจิจูด (Longitude) ซึ่งสามารถระบุได้จากระบบระบุตำแหน่งบนพื้นโลก (Global Positioning System: GPS) โดยอุปกรณ์ GPS ที่ติดตั้งในยานพาหนะจะรับข้อมูลตำแหน่งจากดาวเทียม และทำการส่งข้อมูลเหล่านี้ไปยังระบบเซิร์ฟเวอร์ผ่านเครือข่ายอินเทอร์เน็ต เช่น เครือข่าย 4G โดยใช้งานร่วมกับอุปกรณ์ Internet of Things (IoT) ที่ถูกฝังไว้ในยานพาหนะ

ข้อมูลตำแหน่งที่ได้รับจะถูกรวบรวมแบบเรียลไทม์เพื่อให้ระบบสามารถทราบตำแหน่งของยานพาหนะในช่วงเวลานั้น ๆ ได้อย่างถูกต้อง รวมทั้งสามารถวิเคราะห์พฤติกรรม การขับขี่ หรือสภาพแวดล้อมของรถได้ เช่น ความเร็ว สถานะเครื่องยนต์ หรือระดับเชื้อเพลิง ซึ่งเป็นข้อมูลพื้นฐานในการพัฒนาระบบติดตามและบริหารจัดการยานพาหนะที่มีประสิทธิภาพ



ภาพ 2.1 ระบบที่ติดตั้งภายในตัวรถ

2.1.1.2 การบันทึกข้อมูลตำแหน่งปัจจุบัน

ระบบระบุตำแหน่งบนพื้นโลก (GPS) เป็นเทคโนโลยีที่มีหน้าที่หลักในการระบุตำแหน่งปัจจุบันของวัตถุหรือยานพาหนะในพิกัดทางภูมิศาสตร์ โดยแสดงผลในรูปของค่าละติจูดและลองจิจูด อย่างไรก็ตาม ระบบ GPS โดยตัวมันเองไม่มีความสามารถในการจัดเก็บข้อมูลย้อนหลัง หรือเก็บประวัติเส้นทางการเดินทางของยานพาหนะ

ดังนั้น การนำ GPS มาใช้ในงานติดตามยานพาหนะอย่างมีประสิทธิภาพ จึงจำเป็นต้องอาศัยระบบเสริมสำหรับจัดเก็บข้อมูลตำแหน่งในแต่ละช่วงเวลา เพื่อให้สามารถเรียกดูเส้นทางการเดินทางย้อนหลังได้ ซึ่งข้อมูลดังกล่าวจะเป็นประโยชน์ต่อการวิเคราะห์พฤติกรรมการขับขี่ วางแผนเส้นทาง และตรวจสอบเหตุการณ์ย้อนหลัง

วิธีการบันทึกข้อมูลจากระบบ GPS สามารถแบ่งออกได้เป็น 2 รูปแบบหลัก ดังนี้:

(1) การบันทึกข้อมูลแบบออฟไลน์ (Offline Logging)

เป็นการจัดเก็บข้อมูลตำแหน่งลงในหน่วยจัดเก็บภายในของอุปกรณ์ เช่น ไอโอทีบอร์ด หรือคอมพิวเตอร์ขนาดเล็ก (Single-board Computer) ที่ติดตั้งอยู่ภายในยานพาหนะ อุปกรณ์ดังกล่าวจะบันทึกข้อมูลโดยไม่เชื่อมต่อกับอินเทอร์เน็ตหรือฐานข้อมูลภายนอก ซึ่งเหมาะสำหรับสถานการณ์ที่ไม่มีสัญญาณเครือข่าย หรือใช้ในการบันทึกเพื่อดึงข้อมูลออกภายหลัง

(2) การบันทึกข้อมูลแบบออนไลน์ (Online Logging)

เป็นการส่งข้อมูลตำแหน่งแบบเรียลไทม์จากอุปกรณ์ไอโอทีหรือคอมพิวเตอร์ขนาดเล็กไปยังฐานข้อมูลหรือระบบคลาวด์ผ่านการเชื่อมต่ออินเทอร์เน็ต (เช่น ผ่าน 4G หรือ Wi-Fi) รูปแบบนี้ช่วยให้สามารถติดตามยานพาหนะในเวลาจริง และสามารถประมวลผลข้อมูลได้ทันที ทั้งนี้ ความจำเป็นในการมีหน่วยเก็บข้อมูลภายในตัวเครื่องขึ้นอยู่กับการออกแบบระบบ หากระบบมีการส่งข้อมูลอย่างต่อเนื่องไปยังเซิร์ฟเวอร์ ก็อาจไม่จำเป็นต้องมีพื้นที่จัดเก็บในอุปกรณ์



ภาพที่ 2.2 แสดงตัวอย่างของไอโอทีบอร์ดที่ใช้สำหรับการรับส่งและจัดเก็บข้อมูลตำแหน่ง ซึ่งสามารถทำหน้าที่ได้ทั้งในรูปแบบออฟไลน์และออนไลน์ตามการออกแบบของระบบ

2.1.2 แนวคิดการเก็บข้อมูลที่จำเป็นต่อการวิเคราะห์การเกิดเหตุการณ์

จำเพาะ

เพื่อให้ระบบสามารถวิเคราะห์และประเมินเหตุการณ์จำเพาะ (Specific Events) ที่อาจส่งผลกระทบต่อความปลอดภัยและประสิทธิภาพในการขนส่งได้อย่างถูกต้อง จำเป็นต้องมีการเก็บรวบรวมข้อมูลจากยานพาหนะในลักษณะต่าง ๆ ข้อมูลเหล่านี้สามารถนำมาใช้เป็นฐานในการตรวจจับและวิเคราะห์เหตุการณ์ที่ผิดปกติหรือพฤติกรรมที่ไม่เหมาะสมของผู้ขับขี่ โดยสามารถแบ่งประเภทของข้อมูลที่สำคัญได้ดังนี้

2.1.2.1 การเก็บข้อมูลความเร็วปัจจุบันของพาหนะ

การวัดความเร็ว ปัจจุบันของยานพาหนะสามารถวัดได้หลายวิธี โดยมีทั้งวิธีการใช้ จีพีเอส ในการคำนวณความเร็วในปัจจุบัน หรือเป็นการใช้เซ็นเซอร์ภายในตัวรถ เพื่อวัดความเร็วในปัจจุบัน หรือเป็นการวัดการหมุนของรอบล้อของรถ

2.1.2.2 การเก็บข้อมูลปริมาณน้ำมันปัจจุบันของพาหนะ

การนำตัววัดปริมาณของเหลวภายในถังน้ำมันมาเชื่อมต่อเข้ากับ ไอโอทีหรือคอมพิวเตอร์ขนาดเล็กภายในพาหนะ โดยทำให้สามารถคำนวณปริมาณน้ำมันภายในปัจจุบันของยานพาหนะได้และทำการเก็บข้อมูล

2.1.2.3 การเก็บข้อมูลผู้ขับขี่ปัจจุบันของพาหนะ

การเก็บข้อมูลผู้ขับขี่ภายในปัจจุบันของพาหนะ สามารถทำได้โดยใช้การติดตั้งเครื่องที่จะอ่านค่าข้อมูลบาร์โค้ดบนใบขับขี่ของผู้ขับขี่ และสามารถบันทึกข้อมูล ตัวเลขของใบขับขี่ และบันทึกข้อมูลเพื่อใช้อ้างอิงถึงตัวผู้ขับขี่ภายในปัจจุบันได้

2.1.3 แนวคิดการออกแบบเหตุการณ์จำเพาะ

เหตุการณ์จำเพาะคือเหตุการณ์ที่ยัดพาหนะกำลังเผชิญกับเหตุการณ์ที่ไม่เป็นปกติ ระหว่างการขับขี่ หรือเหตุการณ์ที่ไม่เป็นไปตามต้องการ และเสี่ยงต่อ

ความปลอดภัย ของทั้งผู้ขับขี่ ยานพาหนะ และทรัพย์สินบนท้องถนน โดยเหตุการณ์
จำเพาะจะถูกออกแบบด้วยกันดังนี้

2.1.3.1 รถมี่ความเร็วเกินกำหนด

ความหมายคือ การเคลื่อนที่ปัจจุบันของรถจะมีความเร็วเกินที่
กำหนดเช่น ยานพาหนะที่มีขนาดใหญ่จะมีความเร็ว ที่เกินกำหนดที่ซึ่ง เป็น
อันตรายต่อการขับขี่ ได้แก่ ความเร็วเกิน 80 กิโลเมตรต่อชั่วโมง และ
รถบรรทุกขนาดเล็กจะ มีความเร็วที่เกินกำหนดที่ซึ่งเป็นอันตรายต่อการขับ
ขี่ ได้แก่ความเร็ว 100 กิโลเมตรต่อชั่วโมงเป็นต้น โดยสามารถออกแบบให้
เหตุการณ์เฉพาะ วิเคราะห์การขับขี่ ความเร็วเกินกำหนดหรือให้ข้อกำหนด
พิเศษ ภายในเงื่อนไขที่กำหนด ในการเกิดเหตุการณ์เฉพาะที่เป็นเหตุการณ์
รองของเหตุการณ์เฉพาะที่กำลังเกิดขึ้นภายในปัจจุบันได้ โดยมีดังนี้

- (1) การขับขี่เกินความเร็วที่กำหนดเป็นเวลานาน โดยเป็นการ
ประยุกต์ใช้ เวลาเข้ามาเกี่ยวข้อง กับข้อมูลของความเร็วรถ
โดยเป็นการกำหนดว่าความเร็วรถจะต้องไม่เกิน กี่กิโลเมตรต่อ
ชั่วโมงภายในเวลาที่กำหนด จะถือเป็นเหตุการณ์จำเพาะ
ประเภท รถมี่ความเร็วเกินกำหนดเป็นเวลานาน
- (2) การขับขี่เกินความเร็วที่กำหนดภายในชุมชน สามารถ
ออกแบบให้ ความเร็วสัมพันธ์กับตำแหน่ง เช่น ในตำแหน่ง
ปัจจุบันรถกำลังเคลื่อนที่อยู่ในเขตที่มีชุมชนแออัดหรือเป็น
เขตที่จะต้องจำกัดความเร็ว ซึ่งเป็นความเร็วที่ต่ำกว่าที่
เหตุการณ์จำเพาะกำหนด โดยเป็นการกำหนดว่าความเร็วรถ
จะต้องไม่เกินกี่กิโลเมตรต่อชั่วโมง ภายในพื้นที่ ที่กำหนด จะ
ถือเป็นเหตุการณ์เฉพาะประเภท รถมี่ความเร็วเกินกำหนด
ภายในเขตพื้นที่ที่กำหนด
- (3) การขับขี่เกินความเร็วที่กำหนดด้วยสภาพรถที่ไม่เป็นปกติ
สามารถออกแบบให้ ระบบเซ็นเซอร์ต่างๆภายในตัวรถ มี
ความสัมพันธ์กับความเร็วรถในปัจจุบัน เช่น ปัจจุบันรถมี่
ความเร็วเกินกำหนด และมีการเปิดประตู เป็นการที่เปิดประตู
ไม่สนิท หรือ ใจโรสโคป ในตัวรถมีการเปลี่ยนแปลง ไปมาด้วย

ความถี่ที่ไม่ปกติ จะถือเป็นเหตุการณ์เฉพาะประเภทรถมี
ความเร็วเกินกำหนด โดยตัวรถมีสภาพผิดปกติ

2.1.3.2 รถมีน้ำมันลดลงผิดปกติ

ความหมายคือ การที่น้ำมันของรถมีการลดที่ผิดปกติเกินกว่าเกินที่
จะลดลงในเวลาหนึ่ง โดยออกแบบว่าหากมี ข้อมูลที่มีค่าน้ำมันลดลง
มากกว่า 10 % ภายในเวลา 20 จะถือว่ามือน้ำมันที่มีค่าลดมากกว่าปกติ

2.1.3.3 รถมีข้อมูลส่งมา มีการสูญหาย

ความหมายคือ การที่ข้อมูลที่ได้รับขาดข้อมูลที่สำคัญไป เช่น
ข้อมูลของ GPS เพื่อให้ผู้ใช้งานสามารถรู้ได้ว่า รถขนส่งคันไหนมีความ
ผิดปกติของ hardware

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 การเปรียบเทียบ Redis กับฐานข้อมูลเชิงสัมพันธ์

ในการจัดการข้อมูลปริมาณมากที่มีความเปลี่ยนแปลงอย่างต่อเนื่อง เช่น ข้อมูล
ตำแหน่งพิกัดของยานพาหนะ หรือ ข้อมูลพื้นที่ภูมิศาสตร์ ที่เกี่ยวข้องกับสถานี อำเภอ จังหวัด ฯลฯ
ระบบจำเป็นต้องมีการประมวลผลข้อมูลเหล่านี้ในลักษณะเรียลไทม์เพื่อใช้ประกอบการวิเคราะห์และ
จำแนกเหตุการณ์จำเพาะอย่างมีประสิทธิภาพ

หากใช้เพียง ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เช่น SQL Server หรือ
MySQL ในการดึงข้อมูลทุกครั้งที่มีการร้องขอ (query) จะทำให้เกิดความล่าช้าในการประมวลผล
โดยเฉพาะเมื่อมีการเข้าถึงข้อมูลจำนวนมากหรือซ้ำ ๆ จากหลายผู้ใช้งานพร้อมกัน

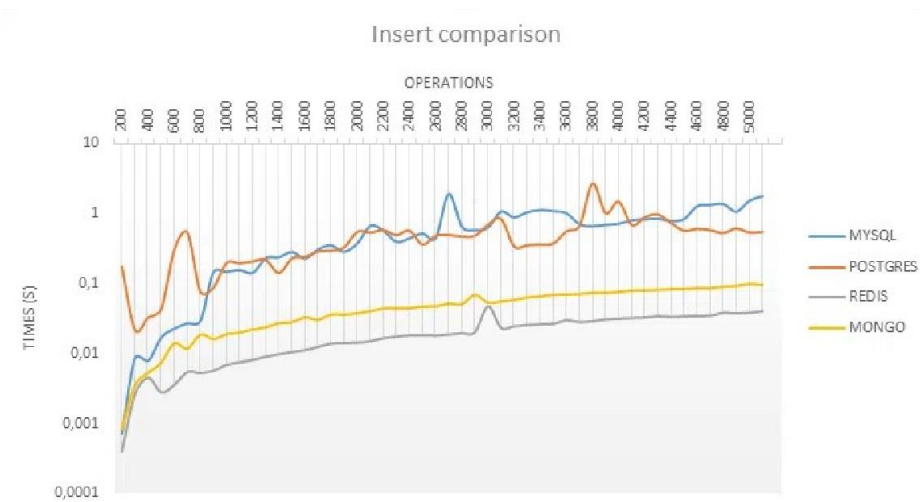
เพื่อแก้ไขปัญหานี้ ได้มีการนำ Redis ซึ่งเป็นฐานข้อมูลแบบ In-Memory เข้ามาใช้
ร่วมกับระบบ โดย Redis มีลักษณะการทำงานในรูปแบบ Key-Value Store และทำงานทั้งหมดบน
หน่วยความจำหลัก (RAM) จึงสามารถตอบสนองคำสั่งได้อย่างรวดเร็วกว่าฐานข้อมูลแบบดั้งเดิม
โดยเฉพาะในกรณีที่ต้องมีการเข้าถึงข้อมูลบ่อยครั้ง และต้องการผลลัพธ์แบบทันที (low-latency)

การทดลองเปรียบเทียบประสิทธิภาพ

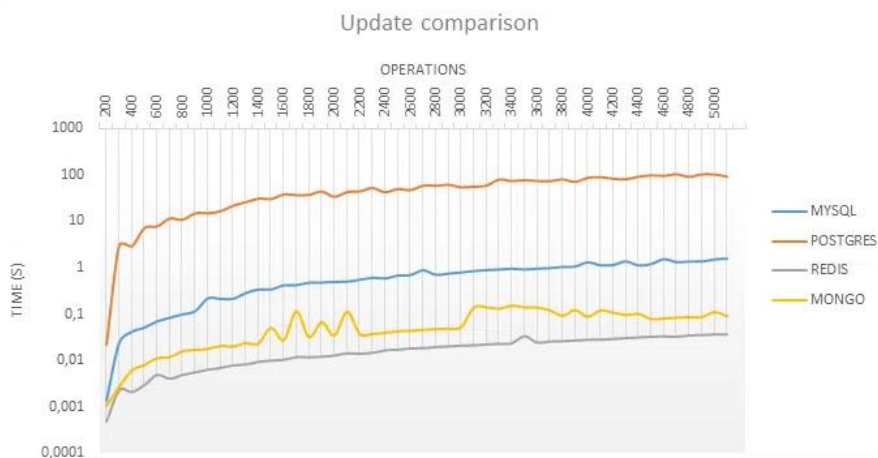
มีการทดลองเปรียบเทียบประสิทธิภาพของ Redis กับฐานข้อมูลอื่นในด้านต่าง ๆ
ได้แก่:



ภาพที่ 2.3 เปรียบเทียบเวลาในการ select (ดึงข้อมูล)



ภาพที่ 2.4 เปรียบเทียบเวลาในการ insert (เพิ่มข้อมูล)



ภาพที่ 2.5 เปรียบเทียบเวลาในการ update (อัปเดตข้อมูล)

ผลการทดลองแสดงให้เห็นว่า Redis มีประสิทธิภาพในการดำเนินการที่สูงกว่าในเกือบทุกกรณี โดยเฉพาะในการ select ข้อมูลที่ระบุจาก key ที่ชัดเจน Redis สามารถให้ผลลัพธ์ได้รวดเร็วกว่าอย่างมีนัยสำคัญ เนื่องจากไม่ต้องดำเนินการ join ตาราง หรือสแกนแถวข้อมูล (row scan) เหมือนกับระบบฐานข้อมูลเชิงสัมพันธ์

Redis จึงเหมาะสำหรับใช้ในการจัดเก็บข้อมูลที่มีการเรียกใช้งานซ้ำ ๆ หรือข้อมูลที่ต้องแสดงผลแบบเรียลไทม์ เช่น:

- ตำแหน่งปัจจุบันของยานพาหนะ
- สถานะเหตุการณ์จำเพาะที่เกิดขึ้นล่าสุด
- ค่าที่เปลี่ยนแปลงตลอดเวลา เช่น ความเร็ว หรืออุณหภูมิ

ขณะที่ SQL หรือฐานข้อมูลเชิงสัมพันธ์ยังคงเหมาะสำหรับจัดเก็บข้อมูลระยะยาว เช่น ข้อมูลผู้ซื้อ ข้อมูลประวัติรถ หรือข้อมูลสถิติที่มีโครงสร้างซับซ้อน

การใช้ **Hybrid Database Architecture** (การผสมผสาน Redis กับ SQL) จึงเป็นแนวทางที่เหมาะสมสำหรับโครงการนี้ เพื่อให้ได้ทั้งประสิทธิภาพและความถูกต้องในการจัดเก็บข้อมูลระยะสั้นและระยะยาว

บทที่ 3

วิธีการดำเนินการวิจัย

3.1 ภาพรวมของโครงการ

ระบบจัดการข้อมูลยานพาหนะที่พัฒนาขึ้นในโครงการนี้ มีวัตถุประสงค์เพื่อเชื่อมโยงและบริหารจัดการข้อมูลจากอุปกรณ์ Internet of Things (IoT) ที่ติดตั้งอยู่ในยานพาหนะ โดยระบบจะทำหน้าที่ระบุตัวตนของยานพาหนะแต่ละคันผ่านรหัสเฉพาะของอุปกรณ์ IoT จากนั้นจะผูกข้อมูลที่ได้รับเข้ากับระบบจัดการข้อมูลหลัก เพื่อใช้ในการวิเคราะห์และจำแนกเหตุการณ์จำเพาะ (Specific Events) ที่อาจเกิดขึ้นในระหว่างการเดินทางของยานพาหนะ ข้อมูลที่ถูกส่งจากอุปกรณ์ IoT จะถูกส่งต่อเข้าสู่ระบบประมวลผล ซึ่งทำหน้าที่ตรวจสอบว่าข้อมูลดิบในขณะนั้นของยานพาหนะ ตรงตามเงื่อนไขที่กำหนดไว้สำหรับการเกิดเหตุการณ์จำเพาะหรือไม่ เช่น การขับรถเร็วเกินกำหนด การหยุดรถในพื้นที่ห้ามจอด หรืออุณหภูมิของห้องบรรทุกที่สูงเกินค่ามาตรฐาน โดยระบบจะต้องสามารถแยกแยะเหตุการณ์เหล่านี้ได้แบบอัตโนมัติและแม่นยำ เพื่อให้หน่วยงานหรือผู้ประกอบการสามารถเฝ้าระวังและจัดการความเสี่ยงได้อย่างมีประสิทธิภาพ

3.1.1 ขั้นตอนการเก็บรวบรวมข้อมูลของยานพาหนะที่ติดตั้งระบบติดตาม เพื่อใช้ในการวิเคราะห์เหตุการณ์จำเพาะและการติดตามแบบเรียลไทม์



ภาพ 3.1 การออกแบบระบบการจัดการข้อมูลระหว่างรถขนส่งกับServer

จากภาพที่ 3.1 แสดงโครงสร้างการทำงานของระบบจัดการข้อมูล ซึ่งเริ่มต้นจากการติดตั้งเซ็นเซอร์ (Sensor) หลายประเภทภายในยานพาหนะเพื่อทำการตรวจวัดข้อมูลที่สำคัญได้แก่:

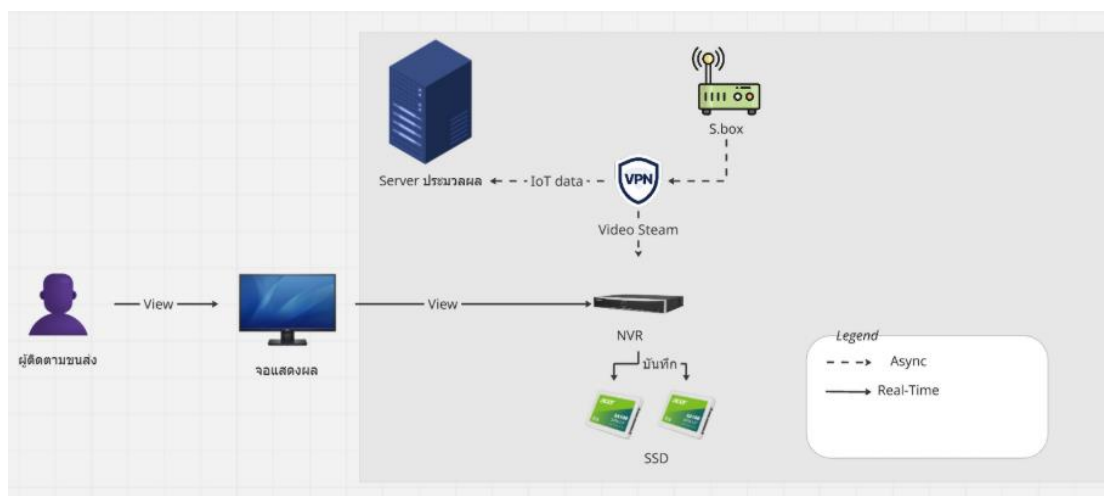
- อุณหภูมิภายในห้องเก็บสินค้า (สำหรับควบคุมคุณภาพของสินค้าประเภทอาหารหรือยา)
- ความเร็วในการเคลื่อนที่ของรถ (ใช้ตรวจสอบพฤติกรรมการขับขี่)
- ปริมาณน้ำมันเชื้อเพลิงในถัง (ใช้ตรวจสอบการสิ้นเปลืองและป้องกันการทุจริต)
- กล้องภายในตัวรถ (เพื่อตรวจสอบพฤติกรรมของผู้ขับขี่ภายในห้องโดยสาร)

ข้อมูลจากเซ็นเซอร์ทั้งหมดจะถูกรวบรวมโดยอุปกรณ์ Mikrotik ซึ่งทำหน้าที่เป็นเกตเวย์ในการรับข้อมูลจากอุปกรณ์ต่าง ๆ และส่งต่อข้อมูลไปยังเซิร์ฟเวอร์ผ่านเครือข่ายอินเทอร์เน็ต โดยใช้ซิมการ์ดที่รองรับการเชื่อมต่อผ่านเครือข่ายมือถือ (เช่น 4G)

เมื่อข้อมูลเดินทางถึงเซิร์ฟเวอร์แล้ว จะถูกส่งเข้าสู่ Mosquitto Server ซึ่งทำหน้าที่เป็น MQTT Broker ในการจัดการข้อความจากอุปกรณ์ IoT ก่อนเข้าสู่กระบวนการประมวลผลและจำแนกข้อมูลภายในระบบ

ข้อมูลที่ผ่านการจัดเก็บและแปลงแล้วจะถูกนำไปวิเคราะห์ผ่านโมดูล "ตัวจำแนกเหตุการณ์" (Event Classifier) เพื่อพิจารณาว่าข้อมูลดังกล่าวเข้าข่ายเหตุการณ์จำเพาะหรือไม่ และหากตรงตามเงื่อนไข ระบบจะทำการบันทึก จัดเก็บ หรือแจ้งเตือนให้ผู้ใช้งานได้รับทราบแบบเรียลไทม์ผ่านระบบติดตาม

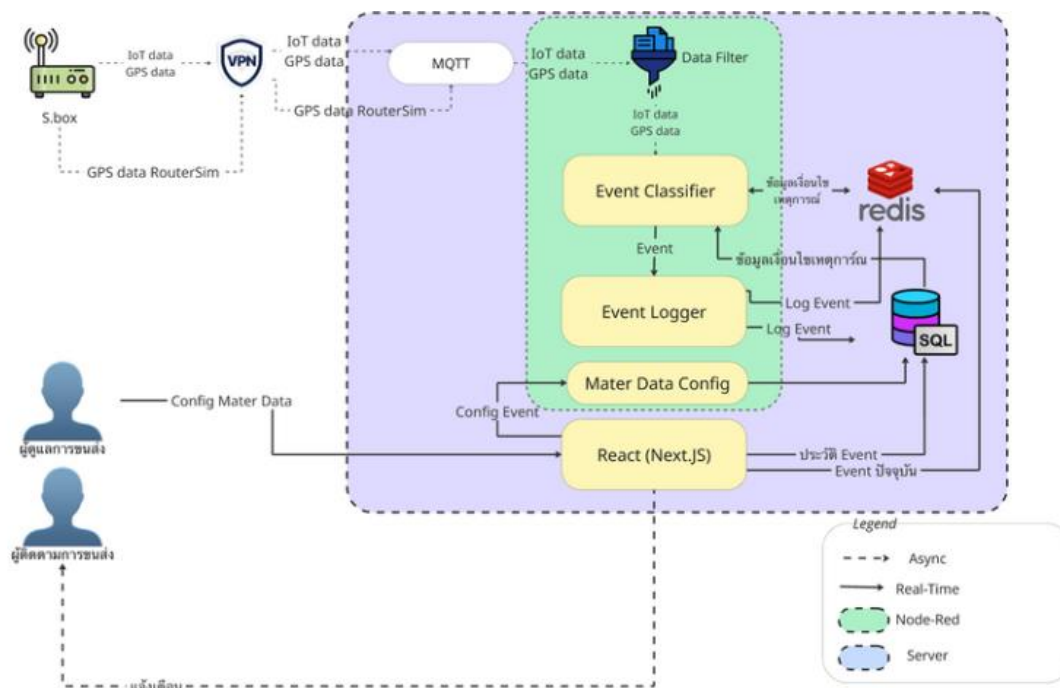
3.2 System Diagram



ภาพ 3.2 การส่งข้อมูลจากกล่อง S.box สู่ตัวบันทึกภาพ และ Server ประมวลผล

จากภาพ 3.2 แสดงถึงข้อมูล Sensor หรือเรียกว่า IoT data และภาพจากกล้องบนตัวรถที่วิ่งผ่าน VPN เพื่อความปลอดภัยของข้อมูลที่มีความสำคัญ โดยข้อมูล IoT data จะส่งต่อเข้าสู่ Server ประมวลผล เพื่อใช้ภายในระบบวิเคราะห์จำแนกเหตุการณ์จำเพาะ ส่วนภาพ Video Stream ที่เป็นภาพจากกล้องบนตัวรถ จะถูกส่งไปที่ NVR (Network Video Recorder) และบันทึกลงบน SSD (Solid State Drive) เพื่อใช้ในการดูวิดีโอย้อนหลัง โดยผู้ติดตามรถขนส่งสามารถเข้ามาดูบันทึกได้ผ่านจอแสดงผล

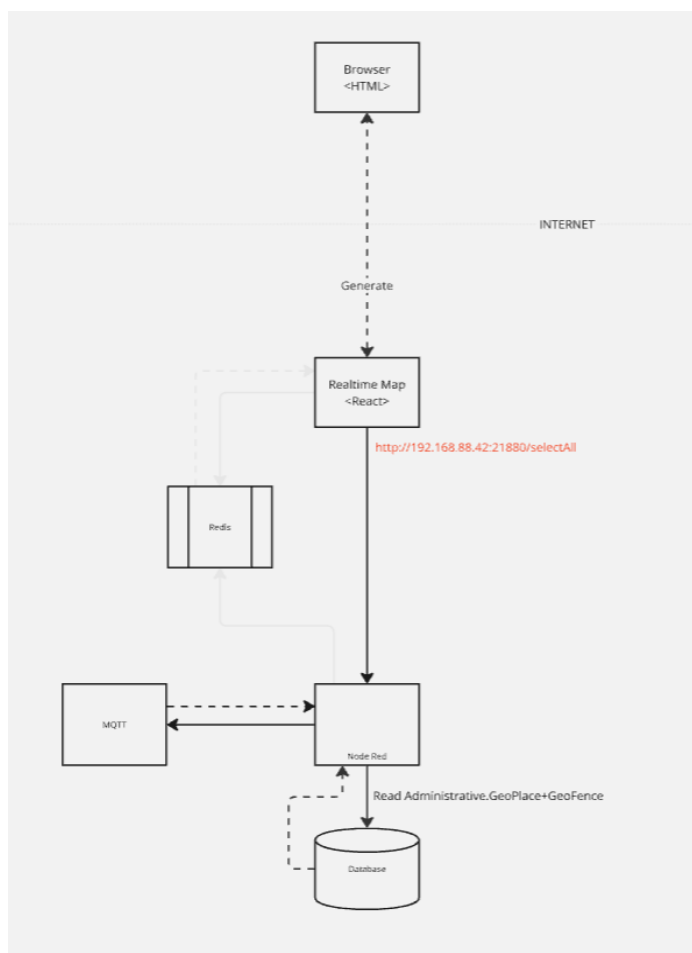
3.3 Design ของ EventClassifier



ภาพ 3.3 Data Flow ของระบบ EventClassifier

จากภาพ 3.3 แสดงถึงเส้นทางของข้อมูลที่ได้รับข้อมูลแล้วส่งต่อผ่าน MQTT Protocol (Message Queue Telemetry Transport) ทำหน้าที่เป็นตัวกระจายข้อมูลแบบ One-To-All เพื่อส่งไปยัง Topic ใดๆ บน Broker โดยทาง Node-Red จะรับข้อมูลของ Payload บน Topic ที่ถูกกำหนดไว้ เพื่อคัดแยกข้อมูลที่ถูกต้องภายใน Data Filter เพื่อนำเข้า Event Classifier เพื่อจำแนกข้อมูลเพื่อหาว่า เหตุการณ์จำเพาะ ที่เกิดขึ้นคือประเภทใด ก่อนถูกบันทึกลงใน Redis เพื่อให้ React เรียกใช้งานได้โดยไม่เสียเวลา ก็อ่าน table ใน SQL ซึ่งกำลังมีการเขียนเหตุการณ์ที่เกิดขึ้นเก็บไว้เป็นประวัติ โดยผู้ใช้งานสามารถเข้าใช้งานได้ผ่านทางหน้าเว็บซึ่งทำงานโดย React

3.4 เครื่องมือที่เกี่ยวข้องกับการทำโครงงาน



ภาพ 3.4 การออกแบบระบบการจัดการข้อมูล

จากภาพที่ 3.4 แสดงสถาปัตยกรรมระบบที่ประกอบด้วยเครื่องมือและเทคโนโลยีหลักที่ใช้ในการจัดการข้อมูลและจำแนกเหตุการณ์จำเพาะ โดยมีการแบ่งบทบาทของแต่ละส่วนออกเป็นกลุ่มตามลำดับการทำงาน ได้แก่ การรับข้อมูล (Input), การประมวลผล (Back-end), การแสดงผล (Front-end) และการจัดเก็บข้อมูล (Database/Cache)

INPUT :: MQTT	<p>MQTT (Message Queuing Telemetry Transport) เป็นโปรโตคอลสำหรับการส่งข้อความแบบ Publish/Subscribe ที่เหมาะสำหรับการสื่อสารระหว่างเครื่องต่อเครื่อง (M2M) โดยเฉพาะในระบบ Internet of Things (IoT) เช่น เซ็นเซอร์อัจฉริยะ, อุปกรณ์สวมใส่ และระบบควบคุมในยานพาหนะ จะใช้ MQTT เพื่อส่งข้อมูลไปยังเซิร์ฟเวอร์หรือคลาวด์ ซึ่งช่วยลดภาระของเครือข่ายและทำให้ระบบสามารถสื่อสารข้อมูลได้อย่างมีประสิทธิภาพแม้ในสภาพแวดล้อมที่มีแบนด์วิดท์ต่ำ</p>
INPUT :: Database - Master Data	<p>ระบบใช้ฐานข้อมูลหลัก (Master Database) ในการจัดเก็บข้อมูลพื้นฐานที่จำเป็น เช่น ข้อมูลรถ สถานีปลายทาง พื้นที่เสี่ยงภัย และเหตุการณ์จราจร ข้อมูลเหล่านี้จะถูกเชื่อมโยงกับข้อมูลปัจจุบันของรถ เพื่อใช้วิเคราะห์และจำแนกเหตุการณ์จำเพาะอย่างถูกต้อง ช่วยให้ระบบสามารถแสดงผลข้อมูลได้อย่างครบถ้วนและแม่นยำ</p>
Front End :: React.js	<p>React.js เป็นไลบรารี JavaScript ที่พัฒนาโดย Meta (เดิมคือ Facebook) และเผยแพร่ในรูปแบบ Open Source React เหมาะสำหรับการสร้างส่วนติดต่อผู้ใช้ (User Interface) ของเว็บแอปพลิเคชัน โดยมีจุดเด่นในการแบ่งโค้ดออกเป็นคอมโพเนนต์ย่อยที่สามารถนำกลับมาใช้ซ้ำได้ และมีประสิทธิภาพในการแสดงผลแบบเรียลไทม์ ช่วยให้ผู้ใช้งานสามารถเข้าถึงข้อมูลเหตุการณ์จำเพาะและสั่งงานระบบได้อย่างรวดเร็ว</p>
Back End :: Node-RED	<p>Node-RED เป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันโดยเน้นการออกแบบโฟลว์การประมวลผลข้อมูล (Data Flow Programming) ผ่านอินเทอร์เฟซแบบกราฟิก (GUI) ในโครงการนี้ Node-RED ถูกใช้เป็นระบบหลังบ้านสำหรับรับข้อมูลจาก MQTT และฐานข้อมูล ก่อนทำการประมวลผลตามตรรกะที่กำหนดเพื่อตรวจสอบว่าข้อมูลดิบที่ได้รับตรงกับเงื่อนไขของเหตุการณ์จำเพาะหรือไม่ ข้อดีของ Node-RED คือสามารถปรับแก้โฟลว์ได้ง่ายโดยไม่ต้องพัฒนาโค้ดจากศูนย์ทุกครั้ง และยังสามารถสื่อสารกับ API อื่น ๆ ได้อย่างยืดหยุ่น</p>

Redis	<p>Redis (Remote Dictionary Server) เป็นฐานข้อมูลแบบ In-Memory ซึ่งหมายถึงการจัดเก็บข้อมูลไว้ในหน่วยความจำหลัก (RAM) ทำให้สามารถเข้าถึงข้อมูลได้เร็วกว่า SQL ฐานข้อมูลแบบดิสก์ Redis เหมาะสำหรับการใช้งานที่ต้องการความเร็วสูง เช่น การจัดเก็บข้อมูลสถานะล่าสุดของรถ หรือการแคช (Cache) ข้อมูลชั่วคราว เช่น รายชื่อเหตุการณ์จำเพาะที่กำลังเกิดขึ้น ข้อมูลใน Redis จะไม่ถูกใช้แทนฐานข้อมูลหลัก แต่ใช้ร่วมกันเพื่อลดภาระของระบบและเพิ่มประสิทธิภาพในการตอบสนองของระบบในส่วนที่เกี่ยวข้องกับการแสดงผลแบบเรียลไทม์</p>
-------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

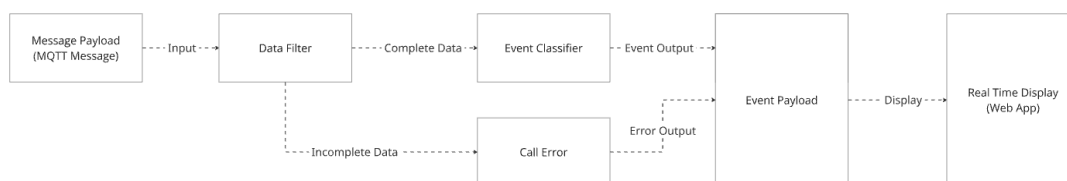
ตารางที่ 3.1 ระบบที่ใช้ภายในระบบการจัดการข้อมูล

ประเภท	เครื่องมือ/ เทคโนโลยี	บทบาทในระบบ
การรับข้อมูล (Input)	MQTT	โปรโตคอลส่งข้อมูลจาก IoT เข้าสู่ระบบด้วยความเร็วสูง
ข้อมูลหลัก (Master)	SQL Database	จัดเก็บข้อมูลยานพาหนะ สถานี พื้นที่ ฯลฯ อย่างเป็นระบบ
ส่วนติดต่อผู้ใช้	React.js	สร้าง UI สำหรับให้ผู้ใช้ดูและแก้ไขข้อมูล/เหตุการณ์ต่าง ๆ
ระบบประมวลผล	Node-RED	ทำหน้าที่วิเคราะห์ข้อมูลดิบกับเงื่อนไขเหตุการณ์จำเพาะ
แคช/สถานะล่าสุด	Redis	เก็บสถานะล่าสุดและข้อมูลเรียลไทม์เพื่อแสดงผลรวดเร็ว

ตารางที่ 3.2 ระบบที่ใช้ภายในระบบการจัดการข้อมูลแบบสรุป

3.3.1 การออกแบบระบบแยกเหตุการณ์จำเพาะ

การออกแบบเหตุการณ์จำเพาะซึ่งจะถูกใช้ภายใน Node Red และจะมีส่วนเชื่อมต่อซึ่งทำหน้าที่เป็นตัวติดต่อการใช้งานเข้ากับตัวผู้ใช้งาน ซึ่งในหน้าที่นี้จะเป็นการทำงานในส่วนของ React ซึ่งสามารถให้ผู้ใช้งานกำหนด เหตุการณ์เฉพาะ เงื่อนไขที่จะเกิด เวลาที่จะถูกกำหนดเกิดเป็นต้น โดยมีการออกแบบขั้นตอนการดำเนินงานดังนี้



ภาพที่ 3.5 ภาพ flow ของระบบที่รับ payload เพื่อใช้ในการประมวลผล

ภาพ 3.5 จะเป็นการทำงานของข้อมูลโดยชุดข้อมูลดิบที่มาจากรถขนส่ง Mosquitto หลังจากรับ Payload เข้ามาแล้วจะมีหน้าที่ส่งต่อไปให้กับตัวกรองข้อมูล โดยตัวกรองข้อมูลจะมีหน้าที่ตรวจสอบว่าข้อมูลที่ Mosquitto รับเข้ามามีข้อมูลที่ถูกต้อง และข้อมูลไม่มีการสูญหาย หากข้อมูลมีการสูญหาย และหากมีความสำคัญ เช่น ระบบติดตาม GPS Error ระบบจะส่ง payload เพื่อแจ้งกับ Real Time Display (WebApp) เพื่อแจ้งให้ผู้รู้ว่า รถขนส่งคันใดที่มีข้อมูล GPS สูญหาย โดยจะแจ้งเตือนกับผู้ว่า GPS Lost ซึ่งเป็น Event ที่มีความสำคัญเมื่อเกิดเป็นระยะเวลาจะทำการแจ้งเตือนผ่าน Notification เพื่อให้ผู้ใช้ทราบได้ง่ายยิ่งขึ้น

โดยเมื่อตัวกรองข้อมูล ยืนยันแล้วว่า ข้อมูลไม่มีการสูญหาย ระบบจะนำข้อมูลไปประมวลผลต่อในระบบ Event Classifier โดยเป็นการรันผ่าน Node Red ซึ่งเป็นส่วนเสริมของ Node JS ซึ่งเป็น Library ของ Java โดยระบบประมวลผลและจัดการ เหตุการณ์จำเพาะ จะถูกติดตั้งไว้ภายใน

(1) Node-RED

Node-RED เป็นเครื่องมือพัฒนาแอปพลิเคชันแบบ Low-Code ที่ออกแบบมาเพื่อเชื่อมโยงการประมวลผลข้อมูลแบบเชิงเหตุการณ์ (Event-Driven Programming) โดยเฉพาะในบริบทของ Internet of Things (IoT) ซึ่งสามารถสร้างโฟลว์การทำงาน (Flow) ผ่านอินเทอร์เฟซแบบกราฟิก (GUI) เพื่ออำนวยความสะดวกในการออกแบบระบบอย่างมีประสิทธิภาพ

ในโครงงานนี้ ระบบ Node-RED ถูกนำมาใช้เป็นส่วนหลักในการประมวลผลและจัดการข้อมูลที่ได้รับมาจากอุปกรณ์ IoT และ Mikrotik ภายในยานพาหนะ โดยมีการออกแบบให้ แยกการทำงานออกเป็น 3 หน่วยหลัก เพื่อป้องกันปัญหาการแย่งใช้งานทรัพยากรของ CPU (CPU thread contention) และเพิ่มความเสถียรในการประมวลผลแบบเรียลไทม์ โดยเฉพาะในกรณีที่มีข้อมูล

การแยกหน่วยการทำงานภายใน Node-RED มีรายละเอียดดังนี้:

1. หน่วยการจัดการฐานข้อมูล (Database Processing Unit):

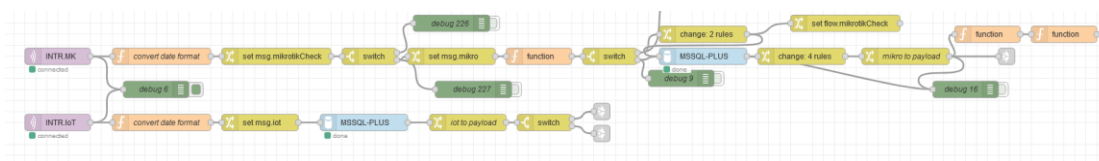
ทำหน้าที่ในการเชื่อมต่อและประมวลผลข้อมูลร่วมกับฐานข้อมูลแบบ SQL ซึ่งครอบคลุมถึงการดึงข้อมูลหลัก (Master Data) เช่น ข้อมูลรถ ข้อมูลผู้ขับขี่ ข้อมูลสถานี และข้อมูลพื้นที่อันตราย เพื่อให้ระบบสามารถอ้างอิงข้อมูลร่วมกันได้อย่างถูกต้อง

2. หน่วยวิเคราะห์เหตุการณ์จำเพาะ (Event Classifier Processing Unit):

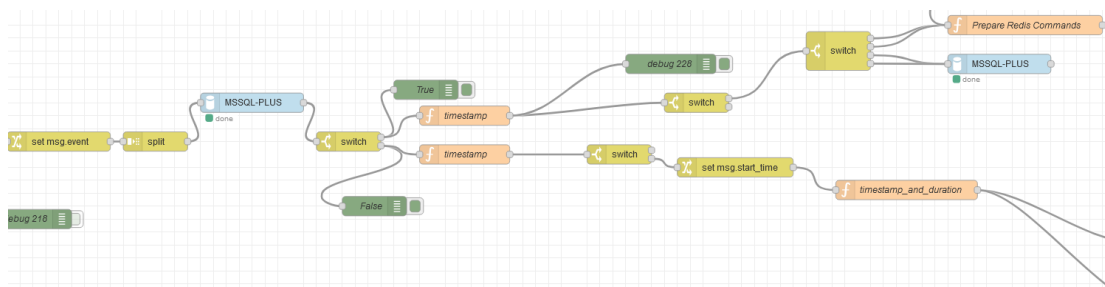
รับผิดชอบในการประมวลผลข้อมูลเรียลไทม์จากยานพาหนะ เพื่อตรวจสอบว่าข้อมูลที่ได้รับเข้าเงื่อนไขของการเกิดเหตุการณ์จำเพาะหรือไม่ โดยใช้โพล์ของ Node-RED ในการจัดลำดับการกรอง วิเคราะห์ และจัดเก็บผลลัพธ์หรือแจ้งเตือนผู้ใช้งาน

3. หน่วยการจัดทำรายงาน (Report Generation Unit):

ออกแบบไว้สำหรับการจัดทำเอกสารรายงานผลการขับขี่ หรือการใช้งานยานพาหนะตามข้อกำหนดของหน่วยงานที่เกี่ยวข้อง เช่น กรมการขนส่งทางบก (หน่วยนี้ยังไม่ได้พัฒนาในโครงงานนี้)



ภาพที่ 3.6 ภาพ flow การรับค่าจาก IoT ภายใน NodeRed



ภาพที่ 3.7 ภาพ flow การคัดแยกเหตุการณ์จำเพาะภายใน NodeRed

ภาพที่ 3.6 แสดงขั้นตอนของการรับข้อมูลจากอุปกรณ์ IoT และอุปกรณ์ Mikrotik ที่ติดตั้งภายในยานพาหนะเข้าสู่ระบบประมวลผล โดยในกระบวนการนี้ ข้อมูลจากทั้งสองแหล่งจะถูกนำมารวมกันและผ่านขั้นตอนการกรองข้อมูลก่อนเข้าสู่การประมวลผลเหตุการณ์จำเพาะในขั้นถัดไป (แสดงในภาพที่ 3.7)

ข้อมูลที่ได้รับจากอุปกรณ์ IoT โดยทั่วไปจะมีลักษณะเป็นข้อมูลจากเซ็นเซอร์ต่าง ๆ ภายในยานพาหนะ เช่น อุณหภูมิภายในห้องเก็บสินค้า หรือระดับน้ำมันเชื้อเพลิง อย่างไรก็ตาม ข้อมูลจาก IoT จะ **ไม่รวม** ตำแหน่งทางภูมิศาสตร์และความเร็วของรถ ซึ่งเป็นข้อมูลที่ได้จากอุปกรณ์ Mikrotik ซึ่งทำหน้าที่เชื่อมต่อกับ GPS และเซ็นเซอร์ความเร็ว

ด้วยเหตุนี้ จึงจำเป็นต้องมี กระบวนการกรองข้อมูล (Data Filtering) เพื่อรวมข้อมูลจากทั้งสองแหล่งให้ครบถ้วน โดยระบบจะจัดการผูกข้อมูลจาก IoT เข้ากับข้อมูลตำแหน่งและความเร็วจาก Mikrotik โดยอิงจากรหัสประจำยานพาหนะหรือหมายเลขอ้างอิง (vehicle ID) ที่สอดคล้องกันในแต่ละช่วงเวลา

เมื่อข้อมูลรวมกันเรียบร้อยแล้ว ระบบจะนำข้อมูลที่ผ่านการกรองนี้ไปตรวจสอบกับเงื่อนไขที่กำหนดไว้ล่วงหน้าในฐานข้อมูล ซึ่งเป็นชุดกฎสำหรับการตรวจจับเหตุการณ์จำเพาะ เช่น ขับรถเร็วเกินกำหนดในพื้นที่ห้ามขับเร็ว หรือมีการเปลี่ยนแปลงอุณหภูมิในห้องสินค้าเกินเกณฑ์ที่กำหนด

หากพบว่า ข้อมูลดิบที่ได้รับตรงตามเงื่อนไข ของเหตุการณ์จำเพาะ ระบบจะทำการ:

1. บันทึกข้อมูลเหตุการณ์นั้นเข้าสู่ฐานข้อมูล Redis เพื่อการเข้าถึงที่รวดเร็ว
2. ส่งข้อมูลไปยังระบบแสดงผลแบบเรียลไทม์ เพื่อแสดงสถานะปัจจุบันของยานพาหนะบนอินเทอร์เฟซผู้ใช้งาน

การใช้ Redis เป็นฐานข้อมูลแบบ In-Memory ช่วยให้สามารถเรียกดูสถานะล่าสุดของยานพาหนะได้อย่างรวดเร็วและมีประสิทธิภาพ ส่งผลให้ผู้ใช้งานสามารถติดตามเหตุการณ์ผิดปกติได้อย่างทันท่วงที

แม้ว่า Node-RED จะเป็นเครื่องมือประเภท Low-Code ซึ่งเอื้อต่อการพัฒนาแอปพลิเคชันได้อย่างรวดเร็วและง่ายดาย โดยเฉพาะในงานที่เกี่ยวข้องกับ IoT หรือการจัดการข้อมูลเชิงเหตุการณ์แบบเรียลไทม์ แต่ในกรณีของโครงการนี้ มีหลายสถานการณ์ที่การใช้โฟลว์พื้นฐานไม่เพียงพอ เนื่องจาก เงื่อนไขของเหตุการณ์จำเพาะ (Specific Events) ที่ต้องการวิเคราะห์นั้นมีความซับซ้อนสูง และต้องการตรรกะที่แม่นยำ

เพื่อรองรับความต้องการนี้ จึงจำเป็นต้องมีการเขียนฟังก์ชันเฉพาะเพิ่มเติมภายใน Node-RED โดยใช้ ภาษา JavaScript ซึ่งทำงานภายใต้แพลตฟอร์ม Node.js ซึ่ง Node-RED เองก็ถูกพัฒนาขึ้นจาก Node.js เช่นกัน

การเขียนฟังก์ชันภายใน Node-RED สามารถทำได้โดยใช้ Function Node ที่ให้ ผู้พัฒนาสามารถใส่โค้ด JavaScript ตามเงื่อนไขเฉพาะที่ต้องการ เช่น การเปรียบเทียบค่าความเร็วกับค่าที่กำหนดในพื้นที่ต่าง ๆ การคำนวณค่าเฉลี่ย การทำงานกับข้อมูลหลายแหล่งพร้อมกัน หรือ แม้แต่การตรวจสอบสถานะจากหลายเซ็นเซอร์ร่วมกันก่อนตัดสินใจว่าเป็นเหตุการณ์จำเพาะหรือไม่

ดังนั้น แม้ Node-RED จะถูกออกแบบมาเพื่อให้เข้าถึงง่ายด้วยการลากเส้นเชื่อมโยง โหนด (Node-based UI) แต่การพัฒนาในระดับลึก เช่น ในโครงการนี้ ก็สามารถปรับแต่งให้รองรับตรรกะที่ซับซ้อนได้เช่นเดียวกับระบบ Full Code ดังต่อไปนี้

1. การค้นหาตำแหน่งของรถขนส่งภายใต้การติชอบเขตภายในแผนที่

การสร้างฟังก์ชันการทำงานเพื่อค้นหาว่าตำแหน่งรถขนส่งนั้นอยู่ภายในพื้นที่ของสถานที่ขนส่ง เขตอันตราย หรือ ตำบล อำเภอ จังหวัดใด เพื่อให้ได้ข้อมูลที่สามารถเข้าใจได้ จึงได้มีการติกรอบข้อมูลที่ต้องการออกเป็น

- ตำแหน่งปัจจุบันของรถขนส่ง
- ตำแหน่งของสถานที่ที่ต้องการระบุตำแหน่งของรถ เช่น จุดส่งสินค้า เขตอันตราย หรือ ตำบล อำเภอ จังหวัด

- ขอบเขตของสถานที่ที่ต้องการระบุพื้นที่ โดยจะแบ่งเป็น รัศมี (Radius) หรือ พื้นที่รูปแบน (Polygon)

โดยแนวคิดจะเป็น เมื่อเรามีตำแหน่งของสถานที่และขอบเขตของสถานที่หลายๆ เราสามารถเอาไปหาคัดแล้ววาดขอบเขตของแต่ละพื้นที่ แล้วนำตำแหน่งรามา พิสูจน์ว่าอยู่ในตำแหน่งใด

โดยการหาพิกัดบนพื้นที่แบบรูปหลายเหลี่ยมอ้างอิงหลักการของ Ray Casting Algorithm โดยกล่าวไว้ว่า

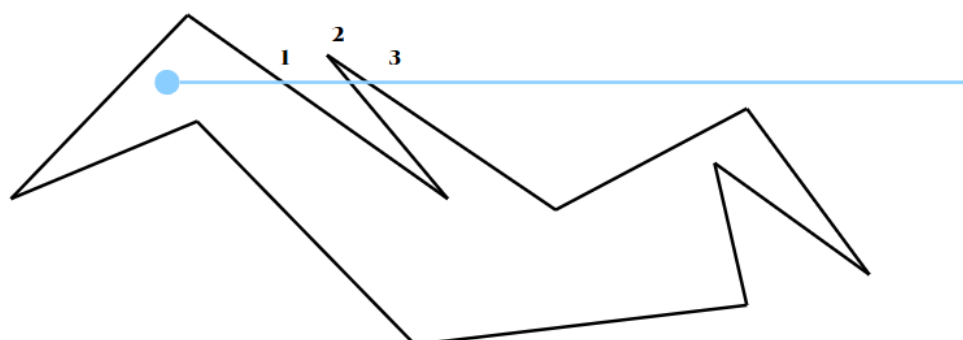
การคำนวณตำแหน่งจาก ค่าพิกัดสองตำแหน่ง เพื่อพิสูจน์ว่าตำแหน่งนั้นอยู่ ภายในรูปหลายเหลี่ยม จะอ้างอิงการคำนวณแบบ Ray Casting Algorithm ซึ่งมีวิธีดังนี้

หลักการ Ray Casting Algorithm มีการคำนวณ ดังนี้

1. กำหนดรูปหลายเหลี่ยมโดยทุกเส้นจำเป็นต้องเชื่อมกัน
2. นำจุดพิกัดหรือตำแหน่งที่อยากพิสูจน์ไปจุด ณ ตำแหน่งใดของภาพ
3. ลากเส้นผ่านรูปหลายเหลี่ยม โดยนับจุดตัดของเส้นเมื่อผ่านเส้นบนรูปหลายเหลี่ยม โดยมีการวัดผลลัพธ์ดังต่อไปนี้

วัดผลลัพธ์:

- ถ้าจำนวนจุดตัด เป็นเลขคี่ หมายถึง จุดอยู่ภายใน
- ถ้าเป็น เลขคู่ หมายถึง จุดอยู่ภายนอก



จำนวนจุดตัด เป็นเลขคี่ หมายถึง จุดอยู่ภายใน

ภาพที่ 3.8 ภาพการหาดำแหน่งบนรูปหลายเหลี่ยมโดยวิธีการ Ray Casting Algorithm

จากภาพ 3.8 เมื่อจุดพิกัดลงบนภาพแล้วลากเส้นไปทางขวาของภาพจะตัดผ่านเส้น
ของรูปหลายเหลี่ยมเป็นจำนวน 3 รอบ จึงตัดสินใจว่าจุดพิกัดนี้อยู่บนพื้นที่รูปหลายเหลี่ยมเป็นต้น

Code การทำงานของฟังก์ชันการตำแหน่ง

```

1  // ฟังก์ชันตรวจสอบตำแหน่งใน Polygon
2  const isPointInPolygon = (point, polygon) => {
3      let x = point.lng, y = point.lat;
4      let inside = false;
5      for (let i = 0, j = polygon.length - 1; i < polygon.length; j = i++) {
6          let xi = polygon[i][0], yi = polygon[i][1];
7          let xj = polygon[j][0], yj = polygon[j][1];
8
9          let intersect = ((yi > y) !== (yj > y)) && (x < (xj - xi) * (y - yi) / (yj - yi) + xi);
10         if (intersect) inside = !inside;
11     }
12     return inside;
13 };
14
15 // รับข้อมูลตำแหน่งรถ (สมมติว่า msg.payload คือข้อมูลตำแหน่งรถ)
16 const vehicleData = msg.add; // สมมติว่า msg.payload คือข้อมูลตำแหน่งรถ
17
18 // ดึงข้อมูล bangkokAreas จาก msg.bangkok ซึ่งเป็น array ของข้อมูล GeoJSON
19 const bangkokAreas = msg.bangkok.map(item => {
20     try {
21         // แปลงข้อมูลที่เป็น string ให้เป็น JSON
22         return JSON.parse(item);
23     } catch (error) {
24         console.log('Error parsing area data:', error);
25         return null; // ถ้าพาร์สไม่สำเร็จให้ส่งค่า null
26     }
27 }).filter(area => area !== null); // กรองข้อมูลที่ไม่สามารถพาร์สได้
28
29 let insideAreas = [];
30
31 // ตรวจสอบตำแหน่งในแต่ละเขต
32 bangkokAreas.forEach(area => {
33     const areaName = area.properties.ADM3_TH; // ชื่อเขต
34     const areaDetails = {
35         ADM3_TH: area.properties.ADM3_TH, // ชื่อเขตระดับ 3
36         ADM2_TH: area.properties.ADM2_TH, // ชื่อเขตระดับ 2
37         ADM1_TH: area.properties.ADM1_TH // ชื่อเขตระดับ 1
38     };
39
40     const polygons = area.geometry.coordinates;

```

ภาพที่ 3.9 ภาพ code ฟังก์ชัน isPointInPolygon ที่ถูกใช้ใน Node-RED

```

41
42     // ตรวจสอบตำแหน่งในแต่ละ Polygon (ถ้ามีหลาย Polygon)
43     for (const polygon of polygons) {
44         if (isPointInPolygon(vehicleData, polygon[0])) {
45             insideAreas.push(areaDetails); // ถ้าอยู่ในเขตนี้
46             return;
47         }
48     }
49 });
50
51 // ถ้าไม่อยู่ในเขตใดๆ ให้ใส่ข้อมูลว่า "Outside any area"
52 if (insideAreas.length === 0) {
53     insideAreas.push('อยู่นอกกรุงเทพ');
54 }
55
56 // ส่งผลลัพธ์กลับไป
57 msg.payload = {
58     vehicleId: vehicleData.id,
59     location: vehicleData,
60     insideAreas: insideAreas
61 };
62
63 return msg;
64

```

ภาพที่ 3.10 ภาพต่อของ code ฟังก์ชัน isPointInPolygon ที่ถูกใช้ภายใน Node-RED

หลักการทำงาน

- ฟังก์ชัน isPointInPolygon

ใช้อัลกอริทึม Ray-casting เพื่อตรวจสอบว่าจุดหนึ่งอยู่ภายในพิกัดของรูปหลายเหลี่ยมหรือไม่ โดยจะทำการวนลูปเช็คแต่ละเส้นของ Polygon เพื่อดูว่าจุดตัดของเส้นกับแนวแกน X ผ่านพิกัดนั้นกี่ครั้ง

- การวนลูปตรวจสอบ

ระบบจะวนลูปตรวจสอบแต่ละ GeoFence หากพบว่าตำแหน่งปัจจุบันอยู่ภายในขอบเขตใด จะทำการบันทึกชื่อของเขตพื้นที่นั้นลงใน Array ที่ชื่อ matchedZones

- รูปผลลัพธ์

```
{“inZones”: [“เขตเอกมัย”], “isInsideAny”: true}
```

หรือ

```
{“inZones”: [ ], “isInsideAny”: false}
```

inZones - แสดงรายชื่อเขตที่พบว่าพิกัดปัจจุบันอยู่ภายใน

isInsideAny - เป็นค่า Boolean ว่าพิกัดอยู่ในเขตใดหรือไม่

การนำไปใช้

ภายใน flow ข้อมูลของรถขนส่งจะถูกส่งมาพร้อมกับตำแหน่งและไอดีของรถขนส่ง โดยเมื่อผ่านฟังก์ชัน isPointInPolygon จะแนบข้อมูลลงไปใน payload โดยถ้า payload ใดมี isInsideAny เป็น True จะทำการนำข้อมูลภายใน inZones ไปแสดงในหน้า UI บนการแสดงตำแหน่งแผนที่ของรถคันแต่ละคัน

2. การเปรียบเทียบข้อมูลกับ Event Condition

การสร้างฟังก์ชันเพื่อเทียบข้อมูลในปัจจุบัน และข้อมูลที่ผู้ใช้งานกำหนดกฎสำหรับเกิด event แบบง่ายๆ ไว้ โดยมีฟังก์ชันดังนี้

```

1 // ดึงข้อมูลจาก msg
2 let data = msg.payload; // ข้อมูล
3 let rules = msg.event_condition; // เงื่อนไข
4
5 let matched = true;
6
7 for (let key in rules) {
8   if (!key in data) {
9     matched = false;
10    break;
11  }
12
13  let condition = rules[key];
14  let value = data[key];
15
16  for (let operator in condition) {
17    let ruleValue = condition[operator];
18
19    switch (operator) {
20      case "<":
21        if (!(Number(value) < Number(ruleValue))) matched = false;
22        break;
23      case "<=":
24        if (!(Number(value) <= Number(ruleValue))) matched = false;
25        break;
26      case ">":
27        if (!(Number(value) > Number(ruleValue))) matched = false;
28        break;
29      case ">=":
30        if (!(Number(value) >= Number(ruleValue))) matched = false;
31        break;
32      case "=":
33        if (!(String(value) == String(ruleValue))) matched = false;
34        break;
35      case "!=":
36        if (!(String(value) != String(ruleValue))) matched = false;
37        break;
38      default:
39        matched = false;
40    }
41
42    if (!matched) break;
43  }
44
45  if (!matched) break;
46 }
47
48 msg.event_matched = matched;
49 return msg;

```

ภาพที่ 3.11 ภาพ code ฟังก์ชันเทียบกับข้อมูล Event ที่ถูกใช้ภายใน Node-RED

หลักการทำงาน

ในการวิเคราะห์และตรวจจับเหตุการณ์จำเพาะ (Specific Events) ที่เกิดขึ้นกับยานพาหนะในแต่ละช่วงเวลา ระบบจะต้องสามารถเปรียบเทียบข้อมูลปัจจุบันของรถ เช่น ความเร็ว ตำแหน่ง หรือสถานะทางกายภาพอื่น ๆ กับเงื่อนไขที่ผู้ใช้งานกำหนดไว้ล่วงหน้า เพื่อระบุว่าเหตุการณ์นั้นเกิดขึ้นหรือไม่

เพื่อให้ระบบมีความยืดหยุ่นและสามารถกำหนดกฎได้อย่างอิสระโดยผู้ใช้งาน ระบบจึงใช้การเขียนฟังก์ชันใน **Node-RED** เพื่อทำหน้าที่ตรวจสอบความสอดคล้องระหว่างข้อมูลที่ได้รับ (msg.payload) กับเงื่อนไขเหตุการณ์จำเพาะ (msg.event_condition)

(2) React

ในหน้าติดต่อระหว่างผู้ใช้ ผู้ใช้งานสามารถเข้าไปแก้ไข หรือ เพิ่มเติมตัวเหตุการณ์เฉพาะและเงื่อนไขที่จะส่งผลทำให้เกิดเหตุการณ์นั้นๆ ได้ผ่านทาง หน้าติดต่อระหว่างส่วนผู้ใช้ภายใน React โดยมีการดีไซน์ซึ่งให้ผู้ใช้สามารถ ใส่รายละเอียดต่างๆของ เหตุการณ์เฉพาะ และมีตารางให้ตั้งค่า ถึงเงื่อนไขที่จะเกิดเหตุการณ์เฉพาะได้ดังภาพ 3.6

The 'Edit Event' form includes the following fields:

- Event ID:** 15
- Event Name:** TEMP_OVER_OVERTIME
- Description:** อุณหภูมิเกินขีดจำกัด
- Severity:** HIGH
- Message Template:**
- Notification:** ☐ Mailhog ☐ Line ☐ Discord ☐ Logstash

The 'IoT Data' section contains a table for configuring event conditions:

IoT Data	Engine	Door	Emergency Button	GPS Lost	In Station	In Danger area	Speed	Temperature	Fuel	Running Time	acc-x	acc-y	acc-z	gyr-x	gyr-y	gyr-z
Engine	On															
Door	Open															
Emergency Button	On															
GPS Lost	Lost															
In Station	In															
In Danger area	In															
Speed																
Temperature																
Fuel																
Running Time																
acc-x																
acc-y																
acc-z																
gyr-x																
gyr-y																
gyr-z																

ภาพที่ 3.12 ภาพ UI การแก้ไขเหตุการณ์จำเพาะ ภายใน React

3.3.2 การออกแบบระบบฐานข้อมูล

ระบบฐานข้อมูลที่ใช้ในระบบการจัดการข้อมูล และจำแนก เหตุการณ์จำเพาะ แบ่งออกเป็น 2 ระบบ ได้แก่ ระบบที่ใช้ฐานข้อมูลแบบ sql และระบบที่ใช้ฐานข้อมูลแบบ in memory ซึ่งจะจำแนกการใช้งานฐานข้อมูล ตามปริมาณคำสั่งที่ต้องใช้ฐานข้อมูล

(1) SQL

ฐานข้อมูลประเภทนี้ ทางระบบได้ออกแบบไว้ให้รองรับข้อมูลที่มีความคงที่แล้ว หรือข้อมูลที่ไม่ต้องการเข้าถึงหรือถูกแก้ไข บ่อยครั้ง เพื่อลดปริมาณการ โหลดของข้อมูล และเพื่อให้ระบบสามารถทำงานได้อย่างรวดเร็วและมีประสิทธิภาพ เช่น เงื่อนไขของเหตุการณ์จำเพาะ ข้อมูลรถ ข้อมูลคนขับ เป็นต้น

โดยมีการออกแบบของความสัมพันธ์ของ SQL จะเป็นดังนี้

ความสัมพันธ์ของรถกับข้อมูลต่างๆ



ภาพที่ 3.13 ภาพการออกแบบข้อมูลภายใน SQL ของรถขนส่ง

Data dictionary

1. vehicles (ยานพาหนะ)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
v_id	int (PK, IDENTITY)	รหัสยานพาหนะ
v_type_id	int	อ้างอิง vehicle_type
mk_id	nvarchar(50)	รหัสอุปกรณ์ MikroTik
iot_id	nvarchar(50)	รหัสอุปกรณ์ IoT
nvr_id	nvarchar(50)	รหัสเครื่องบันทึกภาพ (NVR)
subcontract_id	int	บริษัทผู้รับเหมาที่รถสังกัด
vehicle_name	nvarchar(50)	ชื่อเรียกรถ
license_plate	nvarchar(50)	หมายเลขทะเบียนรถ
description	nvarchar(MAX)	คำอธิบายเพิ่มเติม
province_id	int	จังหวัดที่จดทะเบียน
chassis_id	nvarchar(50)	เลขตัวถังรถ
tracktor_id	nvarchar(50)	รหัสสฟวง (ถ้ามี)
status	bit	สถานะการใช้งาน
mark_delete	bit	สำหรับ soft delete
sbox_id	int	รหัสกล่องควบคุม (ถ้ามี)

ตารางที่ 3.3 Data dictionary ของ vehicles

2. vehicle_type (ประเภทยานพาหนะ)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
v_type_id	int (PK)	รหัสประเภท
type_name	nvarchar(50)	ชื่อประเภท
make	nvarchar(50)	ยี่ห้อ
model	nvarchar(50)	รุ่น
fuel_anatomy	int	ประเภทเชื้อเพลิง
mark_delete	bit	สำหรับ soft delete

ตารางที่ 3.4 Data dictionary ของ vehicle_type

3. vehicle_driver (ความสัมพันธ์รถ-คนขับ)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
v_id	int	รหัสยานพาหนะ
driver_id	int	รหัสพนักงานขับรถ
driving_time	datetime	เวลาที่เริ่มขับรถคันนี้

ตารางที่ 3.5 Data dictionary ของ vehicle_driver

เป็นตารางกลางที่ใช้เชื่อมระหว่าง vehicles กับ driver เพื่อให้สามารถจัดการกรณีที่มีรถ 1 คันอาจมีผู้ขับขี่หลายคนได้ในช่วงเวลาต่าง ๆ

4. vehicle_in_group (ความสัมพันธ์กับกลุ่มรถ)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
v_id	int	รหัสยานพาหนะ
v_group_id	int	รหัสกลุ่มรถ

ตารางที่ 3.6 Data dictionary ของ vehicle_in_group

5. subcontract_company (ข้อมูลบริษัทผู้รับเหมา)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
subcontract_id	int (PK)	รหัสบริษัท
subcontract_company	nvarchar(50)	ชื่อบริษัท
subcontract_address	nvarchar(100)	ที่อยู่
subcontract_phone	nvarchar(50)	เบอร์ติดต่อ
mark_delete	bit	soft delete

ตารางที่ 3.7 Data dictionary ของ subcontract_company

6. mikrotik (อุปกรณ์ Mikrotik ที่ติดตั้งในรถ)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
mikrotik_id	nvarchar(50) (PK)	รหัสอุปกรณ์
router_user	nvarchar(50)	ผู้ใช้
router_password	nvarchar(50)	รหัสผ่าน
ip_address	nvarchar(50)	IP Address
ip_pool, ip_route	nvarchar(50)	IP Pool, Route
sim_number	nvarchar(50)	เบอร์ SIM
vpn_type, connect_to	nvarchar(50)	VPN ประเภท/ปลายทาง
vpn_user/password	nvarchar(50)	VPN credentials
vx_lan_port, vni	int	สำหรับ tunneling
mqtt_id	int	อ้างอิง MQTT
mark_delete	bit	soft delete

ตารางที่ 3.8 Data dictionary ของ mikrotik

7. iot_board (บอร์ด IoT ที่รับข้อมูลเซ็นเซอร์)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
iot_id	nvarchar(50) (PK)	รหัสบอร์ด
wifi_ssid	nvarchar(50)	SSID
wifi_password	nvarchar(50)	รหัสผ่าน
mqtt_id	int	อ้างอิง MQTT
mark_delete	bit	soft delete

ตารางที่ 3.9 Data dictionary ของ iot_board

8. nvr (Network Video Recorder)

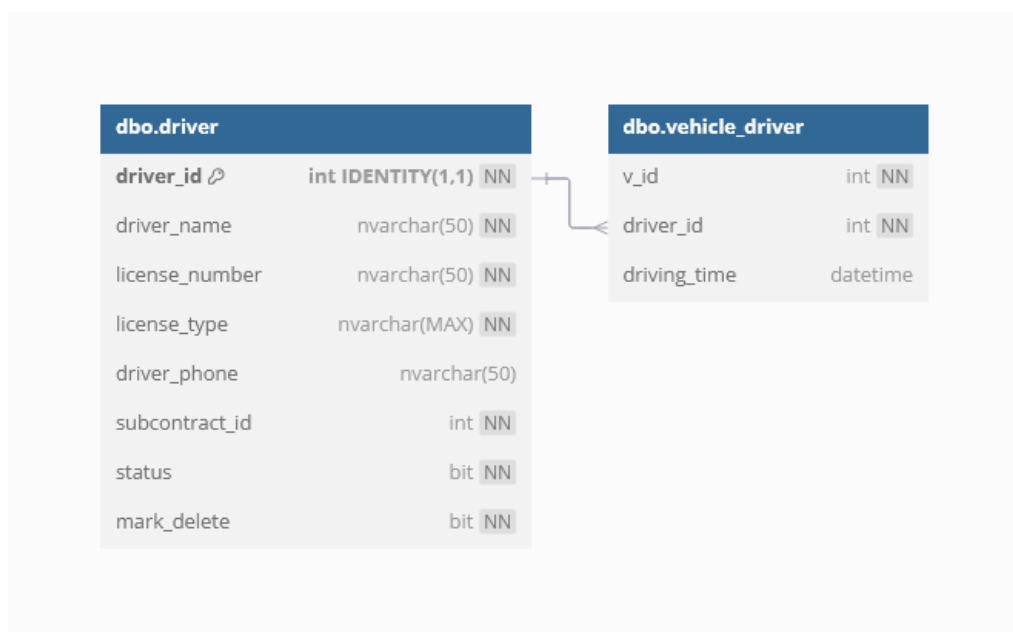
ฟิลด์	ประเภทข้อมูล	รายละเอียด
nvr_id	nvarchar(50) (PK)	รหัสอุปกรณ์
nve_ip_address	nvarchar(50)	IP ของอุปกรณ์
nvr_user/password	nvarchar(50)	ผู้ใช้/รหัสผ่าน
rtsp_url	nvarchar(MAX)	ลิงก์สำหรับ stream
mark_delete	bit	soft delete

ตารางที่ 3.10 Data dictionary ของ nvr (Network Video Recorder)

9. provinces (ข้อมูลจังหวัด)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
province_id	int (PK)	รหัสจังหวัด
adm1_en/th	nvarchar(50)	ชื่อจังหวัด (อังกฤษ/ไทย)
adm1_pcode	nvarchar(50)	รหัสจังหวัด
adm0_en/th	nvarchar(50)	ชื่อประเทศ
adm0_pcode	nvarchar(50)	รหัสประเทศ

ตารางที่ 3.11 Data dictionary ของ provinces



ภาพที่ 3.14 ภาพการออกแบบข้อมูลภายใน SQL ของผู้ขับขี่

Data dictionary

1. driver (ข้อมูลคนขับ)

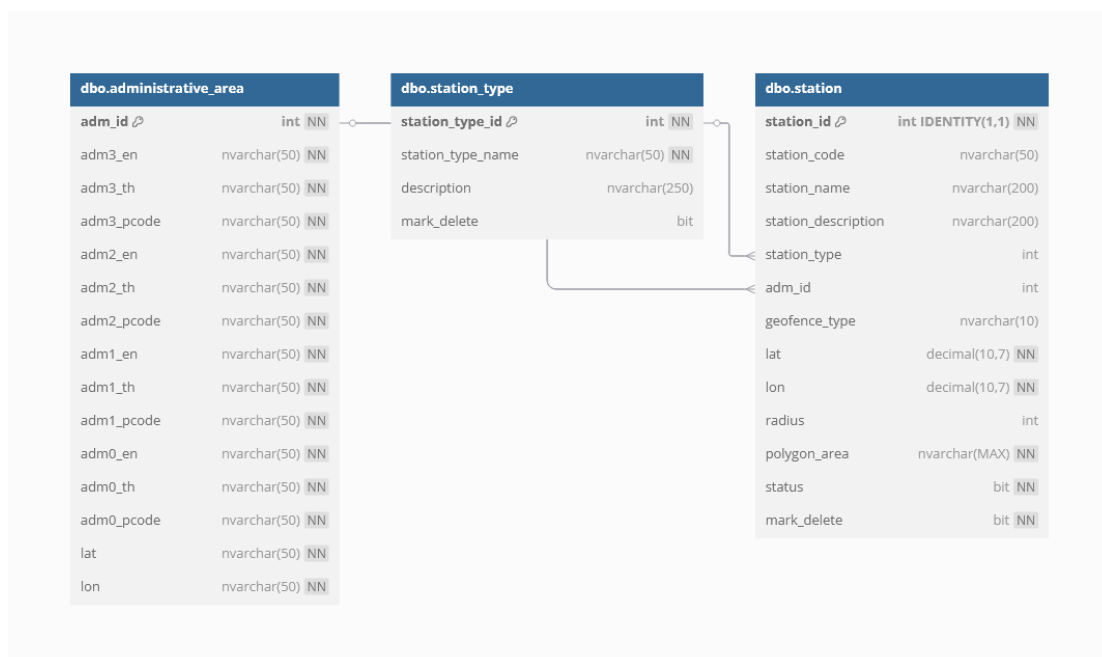
ฟิลด์	ประเภทข้อมูล	รายละเอียด
driver_id	int IDENTITY(1,1)	รหัสพนักงาน (Primary Key, เพิ่มอัตโนมัติ)
driver_name	nvarchar(50)	ชื่อพนักงานขับรถ
license_number	nvarchar(50)	เลขที่ใบขับขี่
license_type	nvarchar(MAX)	ประเภทใบขับขี่ (เช่น ท.1, ท.2 ฯลฯ)
driver_phone	nvarchar(50)	เบอร์ติดต่อ
subcontract_id	int	อ้างอิงไปยังบริษัทรับเหมา (เช่น ถ้าพนักงานไม่ได้เป็นพนักงานองค์กร)
status	bit	สถานะการใช้งาน (1 = ใช้งานอยู่, 0 = ไม่ใช้งาน)
mark_delete	bit	Soft delete (1 = ลบแล้ว, 0 = ยังใช้งาน)

ตารางที่ 3.12 Data dictionary ของ driver

2. vehicle_driver (ความสัมพันธ์รถ-คนขับ)

ฟิลด์	ประเภทข้อมูล	รายละเอียด
v_id	int	รหัสยานพาหนะ
driver_id	int	รหัสพนักงานขับรถ
driving_time	datetime	เวลาที่เริ่มขับรถคันนี้

ตารางที่ 3.13 Data dictionary ของ vehicle_driver



ภาพที่ 3.15 ภาพการออกแบบข้อมูลภายใน SQL ของจุดจัดส่ง

Data dictionary

1. station

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
station_id	int (PK, IDENTITY)	รหัสสถานี (ไล่ลำดับอัตโนมัติ)
station_code	nvarchar(50)	รหัสสถานี
station_name	nvarchar(200)	ชื่อสถานี
station_description	varchar(200)	คำอธิบายสถานี
station_type	int (FK)	อ้างอิงไปยัง station_type
adm_id	int (FK)	อ้างอิงไปยัง administrative_area
geofence_type	nvarchar(10)	ประเภทของ geofence เช่น circle หรือ polygon
lat, lon	decimal(10,7)	พิกัดตำแหน่งจริงของสถานี
radius	int	รัศมี (ถ้า geofence เป็นแบบวงกลม)
polygon_area	nvarchar(MAX)	พื้นที่ polygon (ถ้า geofence เป็นแบบพื้นที่)
status	bit	สถานะการใช้งานของสถานี
mark_delete	bit	เครื่องหมายสำหรับลบ (soft delete)

ตารางที่ 3.14 Data dictionary ของ station

2. station_type

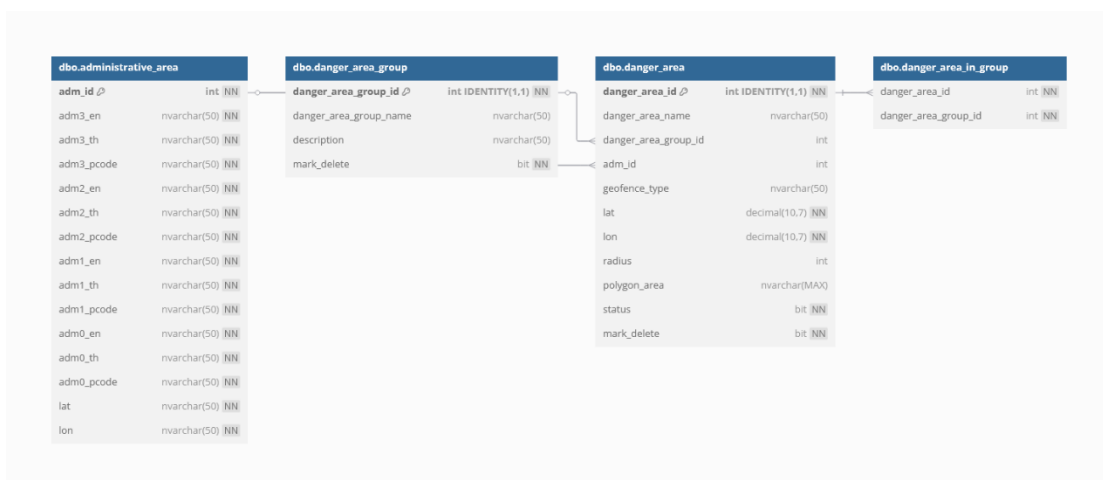
ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
station_type_id	int (PK)	รหัสประเภทสถานี
station_type_name	nvarchar(50)	ชื่อประเภทสถานี
description	nvarchar(250)	รายละเอียดเพิ่มเติมของประเภท
mark_delete	bit	เครื่องหมายสำหรับลบ (soft delete)

ตารางที่ 3.15 Data dictionary ของ station_type

3. administrative_area

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
adm_id	int (PK)	รหัสพื้นที่การปกครอง (Primary Key)
adm3_en, adm3_th	nvarchar(50)	ชื่อตำบล (ภาษาอังกฤษ/ไทย)
adm3_pcode	nvarchar(50)	รหัสตำบล
adm2_en, adm2_th	nvarchar(50)	ชื่ออำเภอ
adm2_pcode	nvarchar(50)	รหัสอำเภอ
adm1_en, adm1_th	nvarchar(50)	ชื่อจังหวัด
adm1_pcode	nvarchar(50)	รหัสจังหวัด
adm0_en, adm0_th	nvarchar(50)	ชื่อประเทศ
adm0_pcode	nvarchar(50)	รหัสประเทศ
lat, lon	nvarchar(50)	พิกัดละติจูดและลองจิจูดของพื้นที่

ตารางที่ 3.16 Data dictionary ของ administrative_area



ภาพที่ 3.16 ภาพการออกแบบข้อมูลภายใน SQL ของพื้นที่เสี่ยง

Data dictionary

1. administrative_area

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
adm_id	int	PK, รหัสพื้นที่การปกครอง
adm3_en/th	nvarchar(50)	ชื่อเขต/อำเภอ
adm3_pcode	nvarchar(50)	รหัสอำเภอ
adm2_en/th	nvarchar(50)	ชื่อจังหวัด
adm2_pcode	nvarchar(50)	รหัสจังหวัด
adm1_en/th	nvarchar(50)	ชื่อภูมิภาค
adm1_pcode	nvarchar(50)	รหัสภูมิภาค
adm0_en/th	nvarchar(50)	ชื่อประเทศ
adm0_pcode	nvarchar(50)	รหัสประเทศ
lat	nvarchar(50)	ละติจูดกลางพื้นที่
lon	nvarchar(50)	ลองจิจูดกลางพื้นที่

ตารางที่ 3.17 Data dictionary ของ administrative_area

2. danger_area_group

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
danger_area_group_id	int IDENTITY	PK, รหัสกลุ่มพื้นที่
danger_area_group_name	nvarchar(50)	ชื่อกุ่ม
description	nvarchar(50)	คำอธิบาย
mark_delete	bit	ใช้ลบแบบ Soft Delete

ตารางที่ 3.18 Data dictionary ของ danger_area_group

3. danger_area

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
danger_area_id	int IDENTITY	PK, รหัสพื้นที่อันตราย
danger_area_name	nvarchar(50)	ชื่อพื้นที่
danger_area_group_id	int	FK → กลุ่มพื้นที่อันตราย
adm_id	int	FK → เขตพื้นที่การปกครอง
geofence_type	nvarchar(50)	ประเภทขอบเขต เช่น circle, polygon
lat, lon	decimal(10,7)	พิกัดศูนย์กลาง
radius	int	รัศมี (สำหรับ geofence แบบวงกลม)
polygon_area	nvarchar(MAX)	พิกัดพื้นที่แบบหลายจุด (Polygon)
status	bit	สถานะเปิด/ปิด
mark_delete	bit	ใช้ลบแบบ Soft Delete

ตารางที่ 3.19 Data dictionary ของ danger_area

4. danger_area_in_group

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
danger_area_id	int	FK → danger_area
danger_area_group_id	int	FK → danger_area_group

ตารางที่ 3.20 Data dictionary ของ danger_area_in_group

(2) Redis

ฐานข้อมูล ชนิดนี้ทางระบบได้ออกแบบให้รองรับข้อมูลที่มีการอ่านเขียน หรือแก้ไข ในปริมาณที่มาก ในเวลาอันสั้น เช่น ข้อมูลตำแหน่งปัจจุบันของรถ ประวัติการเดินทางของรถ สถานที่ที่รถจะต้องเคลื่อนที่ผ่าน หรือสถานะเหตุการณ์ปัจจุบันของรถ เพื่อให้สามารถใช้งาน และดึงข้อมูล ออกมาคำนวณได้อย่างรวดเร็ว และมีประสิทธิภาพ

3.4 การวิเคราะห์ขอบเขตและความต้องการของระบบ

ระบบสามารถทำการเพิ่มแก้ไขหรือดัดแปลง ข้อมูลของยานพาหนะที่จะถูกเชื่อมต่อ เข้ากับระบบ การติดตามของยานพาหนะ ซึ่งจะถูกเชื่อมโยงกับระบบของการจำแยกเหตุการณ์เฉพาะ โดยมีระบบการทำงานอยู่ในส่วนของ React ซึ่งเป็นส่วนติดต่อของผู้ใช้ ซึ่งผู้ใช้จะต้องล็อกอิน เข้ามา เพื่อกำหนดรู้ที่สามารถเข้าถึงการแก้ไขดัดแปลง ข้อมูลต่างๆได้ รวมถึงการดัดแปลง ข้อมูลที่จะถูกใช้เป็นกฎเกณฑ์ในการจำแยก เหตุการณ์จำเพาะ

3.5 ประเด็นที่น่าสนใจและสิ่งที่ท้าทาย

3.5.1 การใช้งานฐานข้อมูลแบบ In Memory

การสร้างระบบที่ไม่เพียงแต่มีฐานข้อมูลที่สามารถใช้งานหรือแก้ไข การทำงานของข้อมูล ที่ดำเนินการภายใต้ระบบ ได้อย่างมีประสิทธิภาพ แต่ต้องคำนึงถึง ความรวดเร็วในการใช้งานของข้อมูลในปริมาณที่มากในเวลาจำกัด เพื่อให้ในภาคธุรกิจนั้นสามารถดึง ประโยชน์ของระบบแอปพลิเคชันขึ้นมาใช้งาน ได้อย่าง รวดเร็ว และมีประสิทธิภาพ ดังนั้นการใช้งานฐานข้อมูลแบบ In Memory จึงเป็นการวัดว่า ระบบแอปพลิเคชันนั้นสามารถใช้งานข้อมูลประเภทนี้ ในการดำเนินการกับข้อมูลที่มีขนาดใหญ่ ภายในปัจจุบันได้หรือไม่

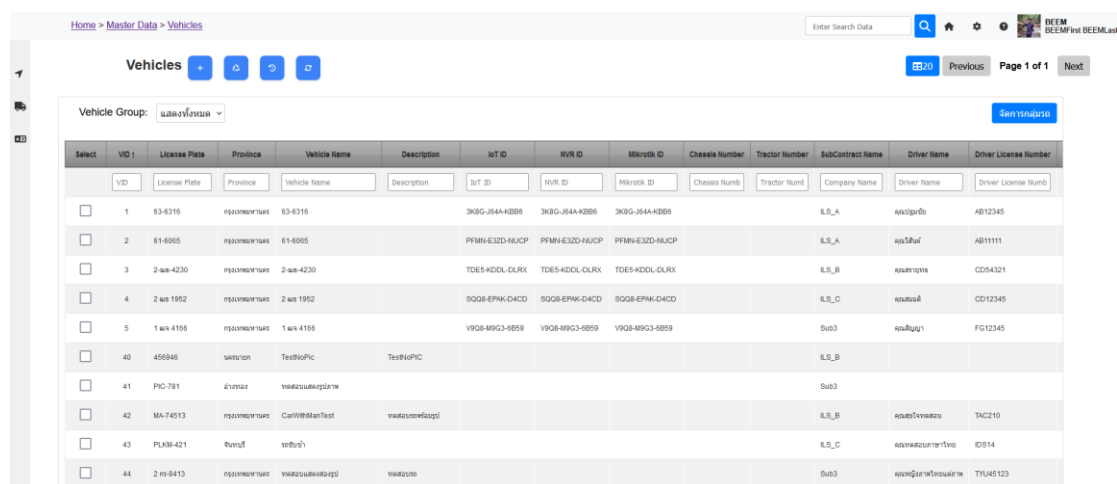
3.6 ผลลัพธ์ที่คาดหวัง

ระบบจัดการข้อมูลและการจำแนกเหตุการณ์จำเพาะสามารถนำ การเรียนรู้ทางคอมพิวเตอร์ มาประยุกต์ ยุคใช้และสามารถแสดงข้อมูลต่างๆที่มีความหมายเก็บผู้ใช้งาน ได้อย่างครบถ้วน โดยผู้ใช้งานสามารถ เลือกรฟังก์ชันการใช้งาน ได้ว่าต้องการรับข้อมูลประเภทที่เป็น เงื่อนไขการทำเนคจำเพาะ เพื่อวัดประสิทธิผลของการทำงานของระบบการจำแนกเหตุการณ์เฉพาะ โดยเหตุการณ์จะต้องผูกกับ ข้อมูลซึ่งอยู่ในระบบการจัดการข้อมูล และการติดตามด้วยยานพาหนะได้อย่างครบถ้วนสมบูรณ์

3.7 ผลลัพธ์

ผลลัพธ์เบื้องต้นของการ เตรียมตัวทำโครงการ เรื่องนี้ ทางเราได้มีการสร้างระบบ จัดการข้อมูล รวมทั้งระบบจำแนกเหตุการณ์จำเพาะเบื้องต้น มีดังนี้

ระบบจัดการข้อมูลสามารถเชื่อมกับข้อมูล ภายในฐานข้อมูล และสามารถ แก้ไข ข้อมูลที่อยู่ในระบบ หรืออัปเดตข้อมูลต่างๆ ได้ โดยมีส่วนติดต่อผู้ใช้งาน โดยผู้ใช้งานสามารถเข้าถึงได้ และทำการแก้ไขข้อมูล หรือการจัดการข้อมูล ดูรายการข้อมูล ทำการรู้ข้อมูลหรือกู้ข้อมูลได้ผ่านทาง ระบบจัดการข้อมูล ซึ่งถูกดำเนินงาน ผ่านทาง React โดยตัวระบบการจัดการข้อมูล

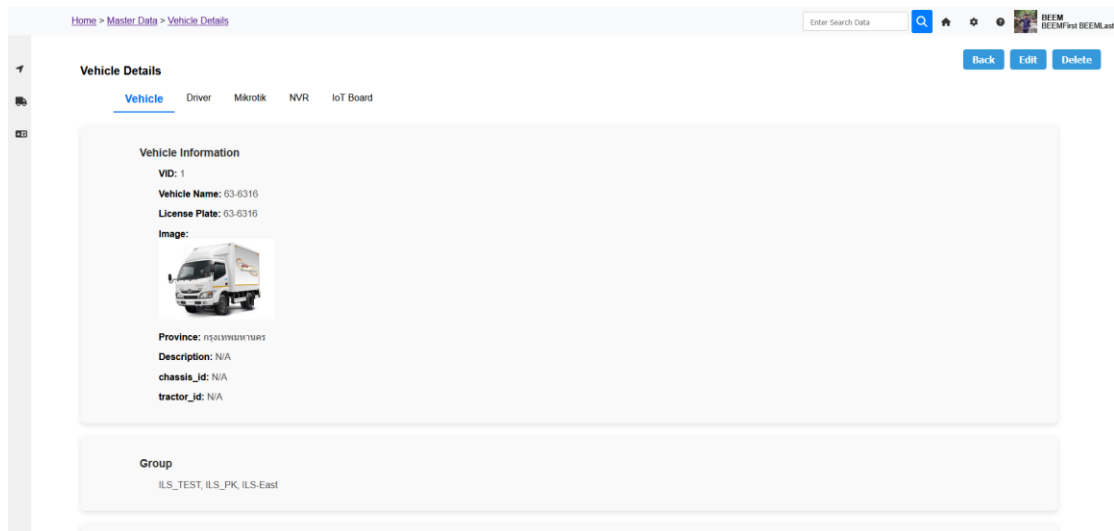


Select	VD ID	License Plate	Province	Vehicle Name	Description	IoT ID	NVR ID	Micro ID	Chassis Number	Tractor Number	SubContract Name	Driver Name	Driver License Number
<input type="checkbox"/>	1	63-6316	กรุงเทพมหานคร	63-6316		3K8G-J64A-K8B6	3K8G-J64A-K8B6	3K8G-J64A-K8B6			LS_A	สมชาย	AB12345
<input type="checkbox"/>	2	61-6065	กรุงเทพมหานคร	61-6065		PFMN-E3ZD-HJCP	PFMN-E3ZD-HJCP	PFMN-E3ZD-HJCP			LS_A	สมชาย	AB11111
<input type="checkbox"/>	3	2-64-4230	กรุงเทพมหานคร	2-64-4230		TDES-KDOL-OLRX	TDES-KDOL-OLRX	TDES-KDOL-OLRX			LS_B	สมชาย	CD54321
<input type="checkbox"/>	4	2-64-1952	กรุงเทพมหานคร	2-64-1952		SOQ8-EPK-K4CD	SOQ8-EPK-K4CD	SOQ8-EPK-K4CD			LS_C	สมชาย	CD12345
<input type="checkbox"/>	5	1-64-4156	กรุงเทพมหานคร	1-64-4156		VQ08-M9C3-8B59	VQ08-M9C3-8B59	VQ08-M9C3-8B59			Sub3	สมชาย	FG12345
<input type="checkbox"/>	40	455948	นนทบุรี	TestPic	TestPic						LS_B		
<input type="checkbox"/>	41	PIC-781	ฉะเชิงเทรา	ทดสอบรถบรรทุก							Sub3		
<input type="checkbox"/>	42	MA-74513	กรุงเทพมหานคร	CarWithMainTest	ทดสอบรถบรรทุก						LS_B	สมชาย	TK210
<input type="checkbox"/>	43	PLK8-421	เชียงใหม่	รถบรรทุก							LS_C	สมชาย	ED14
<input type="checkbox"/>	44	2-64-8413	กรุงเทพมหานคร	ทดสอบรถบรรทุก	ทดสอบรถ						Sub3	สมชาย	THU45123

ภาพที่ 3.17 ภาพ UI หน้าตารางจัดการข้อมูลรถภายใน React

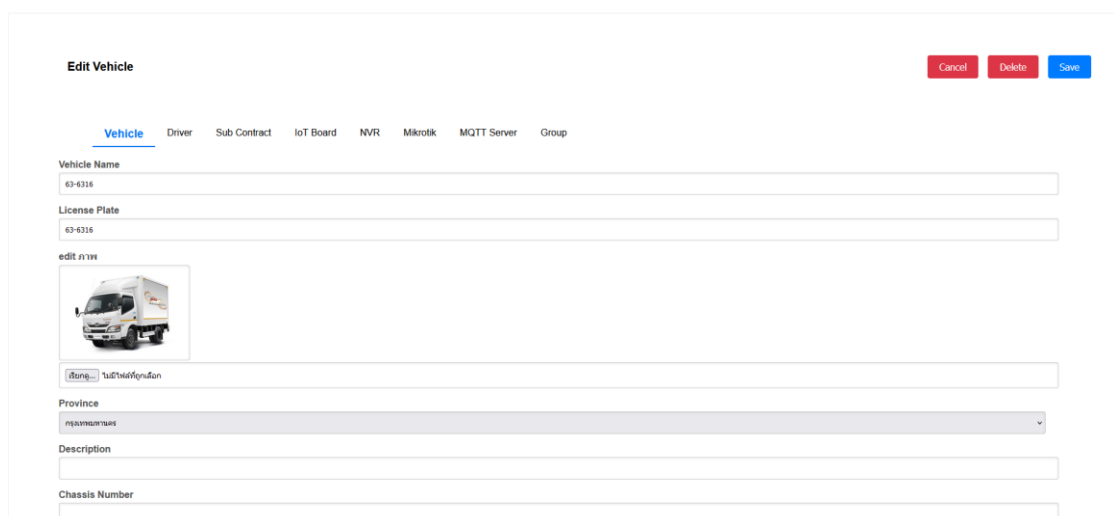
ภาพ 3.17 แสดงถึงหน้าต่างของตารางการจัดการข้อมูลของรถ ภายในระบบ ที่ สามารถเชื่อมต่อเข้ากับระบบการติดตามแบบเรียลไทม์ โดยการออกแบบอ้างอิงถึงประสิทธิภาพต่อ ผู้ใช้ ว่าผู้ใช้สามารถเข้าใช้งานได้โดยเข้าใจได้ง่ายที่สุดและมีประสิทธิภาพที่สุด โดยการวาง แลกการ ควบคุมไว้ทางด้านซ้ายบน แลกการควบคุมตารางไว้ทางด้านขวาบน ส่วนด้านที่เหลือจะเป็นส่วนของ

ตารางข้อมูล โดยผู้ใช้สามารถค้นหา จัดเรียง ข้อมูลภายในตารางได้ตามผู้ต้องการ โดยมีเมนูทางด้านซ้าย แถบการเข้าถึงทางด้านบน



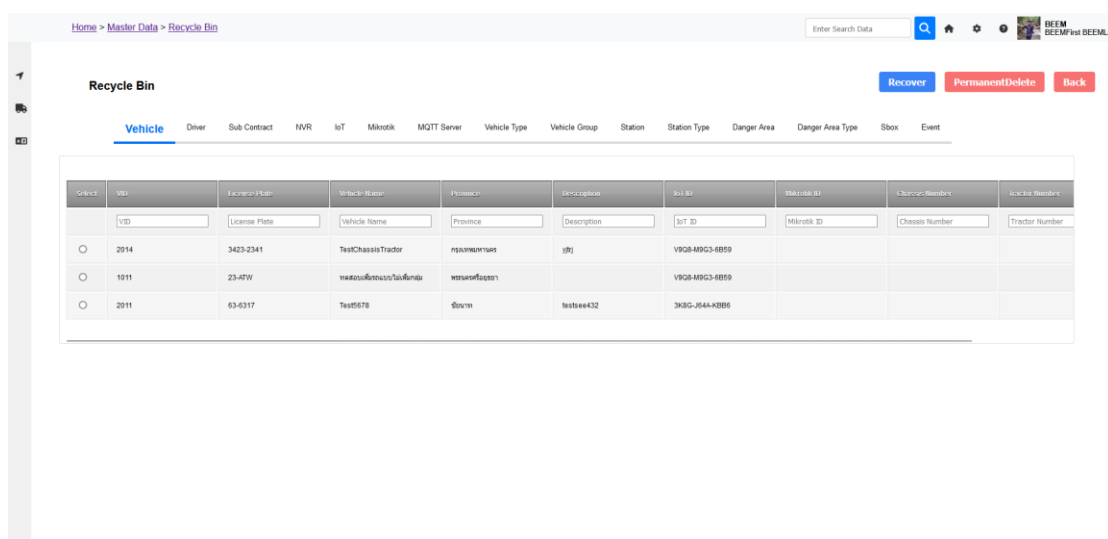
ภาพที่ 3.18 ภาพ UI หน้าข้อมูลรถภายใน React

ภาพ 3.18 แสดงถึง หน้าต่าง การตรวจสอบรายละเอียด ข้อมูลของยานพาหนะ ซึ่งจะสามารถเชื่อมโยงเข้ากับข้อมูลของ ผู้ขับขี่ ตัวระบบไอโอที และระบบกล้องภายในตัวรถได้ โดยมีการจัดเรียง UI โดยผู้ใช้สามารถ ตรวจสอบข้อมูลประเภทต่างๆ ได้ผ่านทางแถบแสดงข้อมูลที่ให้ใช้งาน 3 รายการเพื่อเข้าถึง ข้อมูลที่เกี่ยวข้องกับยานพาหนะคันนี้ได้



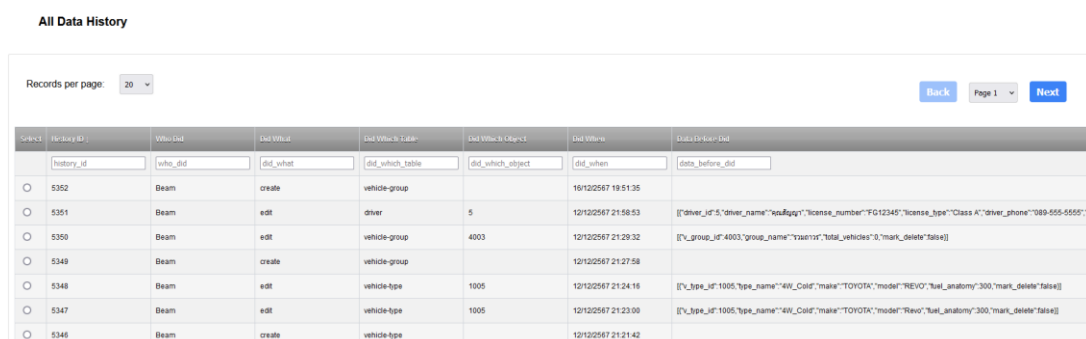
ภาพที่ 3.19 ภาพ UI หน้าแก้ไขข้อมูลรถภายใน React

ภาพ 3.19 เป็นภาพแสดง การแก้ไขข้อมูลผ่านทาง UI โดยผู้ใช้ 3 รอดแก้ไขข้อมูลรถ ข้อมูลคนขับที่ผูกกับรถ และอื่นๆ รวมทั้งกลุ่มรถ ได้ผ่านทางแถบควบคุมทางด้านบนซ้าย และ 3 รอดลบ ยืนยันการแก้ไข หรือยกเลิก ได้ผ่านทาง แถบควบคุมด้านบนขวา



ภาพที่ 3.20 ภาพ UI หน้าตารางจัดการข้อมูลที่ถูกลบใน React

ภาพ 3.20 เป็นภาพ การควบคุมการทำงานของ การกู้คืนหรือการลบ ของระบบการจัดการข้อมูล โดยผู้ใช้สามารถ เลือก ข้อมูลที่ถูกลบ ออกจาก ระบบการจัดการข้อมูลเพื่อ กู้คืน หรือการลบแบบถาวรได้ผ่านทางหน้าต่างตาราง ที่มีแถบควบคุม ที่สามารถเลือก ตารางที่ต้องการกู้คืน หรือการลบแบบถาวร



ภาพที่ 3.21 ภาพ UI หน้าตารางจัดการข้อมูลประวัติภายใน React

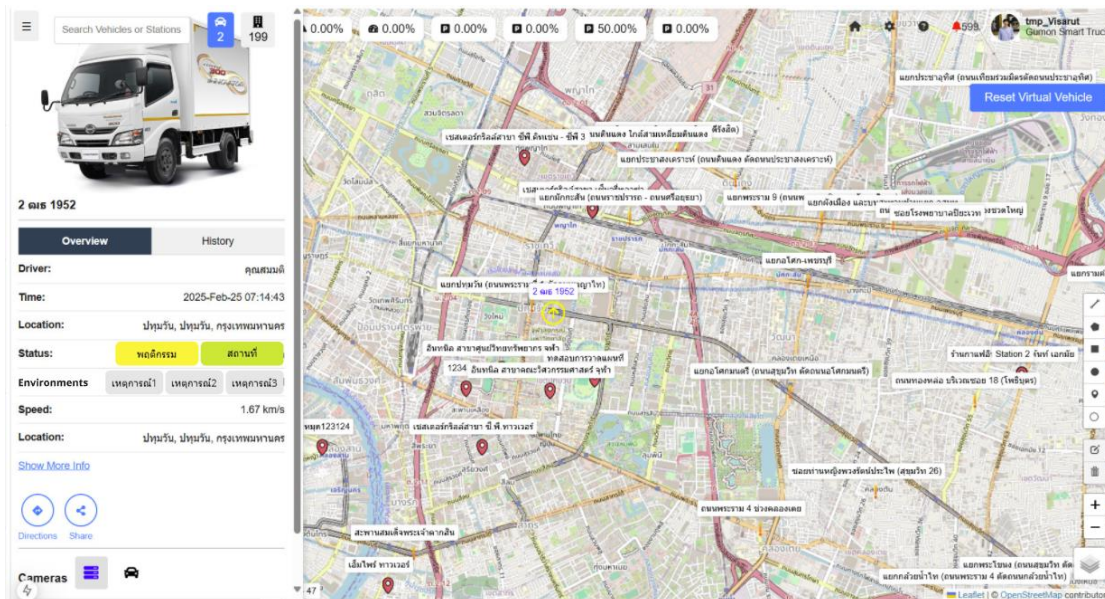
ภาพ 3.21 แสดงถึง ระบบการเก็บบันทึกประวัติการเปลี่ยนแปลง หรือ การดำเนินงาน ภายในระบบการจัดการข้อมูล เพื่อให้ผู้ใช้สามารถ เรียกคืนหรือย้อนกลับการดำเนินงาน ภายในระบบ ของการจัดการข้อมูลได้ และ โดยผู้ใช้สามารถ ค้นหาหรือจัดเรียง ตารางได้ตามผู้กำหนด

Event Detail

Back Edit

ภาพที่ 3.22 ภาพ UI หน้าตารางจัดการข้อมูลเหตุการณ์จำเพาะภายใน React

ภาพ 3.22 แสดงถึงหน้าต่างจัดการข้อมูลเหตุการณ์จำเพาะ ซึ่งสามารถ กำหนดเงื่อนไข ของเหตุการณ์เฉพาะที่ จำแนกข้อมูลดิบที่ยานพาหนะได้ส่งมา ผ่านทางระบบจำแนก เหตุการณ์จำเพาะ ซึ่งถูกทำงานอยู่ภายในหลังบ้าน โดยผู้ใช้สามารถแก้ไขและกำหนดเงื่อนไข ที่ที่จะทำให้เกิดเหตุการณ์เฉพาะได้ผ่านทางหน้าแก้ไขเหตุการณ์เฉพาะนี้



ภาพ 3.23 ภาพ UI แสดงหน้าต่าง แผนที่ตำแหน่งของรถในปัจจุบัน

ภาพ 3.23 แสดงถึงแผนที่ที่มีข้อมูลตำแหน่งพิกัดรถภายในปัจจุบัน พร้อมข้อมูลรถที่ขนส่งที่ผู้ใช้งานสามารถเลือกรถที่กำลังส่งสัญญาณได้โดยจะแสดงรายละเอียด

- ทะเบียนรถ
- ชื่อผู้ขับ
- เวลาข้อมูลปัจจุบัน
- ตำแหน่งปัจจุบันของรถ
- พฤติกรรมของรถ
- ความเร็ว

และมีข้อความแจ้งเตือนอยู่บนตำแหน่งขวบน ในกรณีที่เกิดเหตุการณ์ ที่สำคัญต่อการติดตามรถขนส่ง

บทที่ 4

ทรัพยากรและแผนการดำเนินงาน

4.1 การจัดเตรียมฮาร์ดแวร์และซอฟต์แวร์

ฮาร์ดแวร์ที่ จำเป็นต้องใช้ในการสร้างระบบการจัดการข้อมูล และการจำแนกเหตุการณ์ จำเพาะ จำเป็นจะต้องถูกติดตั้ง และถูกทำงานดังนี้

4.1.1 React.JS

ติดตั้ง React.JS ลงในเครื่องคอมพิวเตอร์ที่พัฒนาระบบการจัดการข้อมูล เพื่อที่จะสามารถดึงข้อมูลจากไลบรารีที่ระบบต้องการที่ใช้งาน เพื่อติดตั้งในการรันระบบขึ้นบน เซิร์ฟเวอร์หรือการทดสอบภายในเครื่องคอมพิวเตอร์ของผู้พัฒนา

4.1.2 NodeRed

NodeRed เป็นส่วนเสริมของ NodeJS ช่วยในการทำ แอปพลิเคชัน API หลังบ้าน โดยง่ายต่อการจัดการและการใช้งาน เนื่องจากมีการทำงานที่ถูกเรียกว่า flow ซึ่งแสดงผลในรูปแบบ UI และสามารถ ทำงานและแก้ไขได้บนเครื่องเซิร์ฟเวอร์ โดยไม่ต้องมีการสร้างใหม่หลายครั้ง

4.1.3 Redis

ฐานข้อมูลแบบ In Memory ติดตั้งบนเซิร์ฟเวอร์เพื่อใช้ในการ วางฐานข้อมูลและตรวจสอบข้อมูลภายในฐานข้อมูล ที่เชื่อมต่ออยู่กับระบบของการจัดการข้อมูลและระบบจำแนกเหตุการณ์เฉพาะ

4.1.4 Pandas Library

ติดตั้งไลบรารีที่สามารถจัดการข้อมูลต่าง ๆ เช่น การโหลดข้อมูลไฟล์ CSV เข้ามาแล้วแสดงข้อมูลออกมาในรูปแบบคล้ายกับ Table โดยมีการแบ่งข้อมูลเป็น Row กับ Column เรียกว่า Data Frame ใช้ในการทำ Data Analysis ไว้ใช้สำหรับตรวจสอบข้อมูล และจัดข้อมูลต่าง ๆ โดยเครื่องมือที่เหมาะสมกับการทำ Data Preparation ก่อนที่จะนำไปใช้เทรน Machine Learning

4.1.5 MSQl

โปรแกรมที่ช่วยสำหรับการ จัดการข้อมูลแบบ SQL ถูกพัฒนาโดยไมโครซอฟท์ ซึ่งช่วยให้เราจัดการเข้าถึงการสร้างฐานข้อมูลแบบตารางได้โดยง่าย และไม่ต้องใช้คำสั่งในการจัดการข้อมูลมากนัก

4.2 แผนการดำเนินงาน

ตารางที่ 4.1 แผนการดำเนินงาน

ขั้นตอน	มค.67				กพ.67				มีค.67				เมย.67				พค.67			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.การออกแบบ database																				
2. การจัดการข้อมูลรถ																				
3. การทดสอบข้อมูลกับระบบ																				
4. สร้างหน้าต่าง UI ด้วย React																				
5. สร้างหน้าต่างสำหรับทดสอบและสร้างข้อมูลทดสอบ																				
6. การสรุปผลลัพธ์ที่ได้																				

ตารางที่ 4.1 การดำเนินงานที่ผ่านมาจนถึงปัจจุบัน

บทที่ 5

สรุป

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนา ระบบจัดการข้อมูลและการจำแนกเหตุการณ์จำเพาะ สำหรับการติดตามและวิเคราะห์สถานะของยานพาหนะ โดยอาศัยข้อมูลจากเซ็นเซอร์และอุปกรณ์ IoT ที่ติดตั้งภายในรถ เช่น Mikrotik และอุปกรณ์ตรวจวัดต่าง ๆ ซึ่งสามารถบันทึกค่าความเร็ว ระดับน้ำมัน อุณหภูมิ รวมถึงข้อมูลผู้ขับขี่ แล้วนำเข้าสู่ระบบประมวลผลผ่าน Node-RED และจัดแสดงผลผ่าน React .s

ระบบมีความสามารถในการ:

- ผสานข้อมูลจากแหล่งต่าง ๆ เข้าด้วยกัน เช่น ข้อมูลจาก GPS และ IoT
- กรองข้อมูล และตรวจสอบว่าเข้าข่ายเหตุการณ์จำเพาะหรือไม่
- วิเคราะห์ข้อมูลแบบเรียลไทม์ และแจ้งเตือนเมื่อเกิดเหตุการณ์ผิดปกติ
- จัดเก็บข้อมูลด้วยฐานข้อมูล SQL และ Redis เพื่อการประมวลผลอย่างมีประสิทธิภาพ
- แสดงผลข้อมูลบนระบบผู้ใช้ (UI) ที่ออกแบบด้วย React ให้สามารถเพิ่ม/แก้ไขเงื่อนไขของเหตุการณ์จำเพาะได้อย่างสะดวก

โครงการนี้ยังคำนึงถึง ความปลอดภัยของข้อมูล โดยมีระบบจำกัดสิทธิ์การเข้าถึง ป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต และมีหลักจริยธรรมในการออกแบบที่เหมาะสมต่อการนำไปใช้งาน ในภาคธุรกิจ โดยเฉพาะอย่างยิ่งในบริบทที่อาจเกี่ยวข้องกับกฎหมายหรือเหตุการณ์อุบัติเหตุ แม้ระบบจะมีข้อจำกัด เช่น การเข้าถึงข้อมูลเชิงลึกของภาคธุรกิจ และความซับซ้อนในการวิเคราะห์ ข้อมูลดิบให้กลายเป็นเหตุการณ์จำเพาะ แต่ก็ยังเป็นโครงการที่สามารถนำไปพัฒนาต่อยอดสู่ระบบบริหารจัดการขนส่งในระดับองค์กรได้จริง ช่วยสนับสนุนการวางแผนด้านความปลอดภัย และการบริหารต้นทุนอย่างมีประสิทธิภาพ

รายการอ้างอิง

“GPS Tracking และรูปแบบของ GPS Tracking ที่ใช้กับยานพาหนะ” [ออนไลน์]. เข้าถึงได้จาก: <https://www.cartrack.co.th/blog/gps-tracking-aelaruupaebbkh-ng-gps-tracking-thiiaichkabyaanphaahna>, 2560. [สืบค้นเมื่อ 21 ตุลาคม 2567].

Jordan, M. (2023). React: Building interactive user interfaces. React. <https://react.dev>

Node-RED Team. (2023). Node-RED: Low-code programming for event-driven applications. Node-RED. <https://nodered.org>

Microsoft Corporation. (2023). SQL Server Management Studio (SSMS). Microsoft. <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>

Mindphp. (n.d.). รู้จักกับ Visual Studio Code (วิหวล สตุติโอ โค้ด) โปรแกรมฟรี จากค่ายไมโครซอฟท์. Mindphp. <https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1/microsoft/4829-visual-studio-code.html>

Sakul Montha. (2021). Redis คืออะไร. Medium. <https://iamgique.medium.com/what-redis-is-4381ff32880d>

Amazon Web Services. (n.d.). Docker คืออะไร. AWS. <https://aws.amazon.com/th/docker/>

Chacon, S., & Straub, B. (2014). Pro Git 2nd Edition. Git-SCM.

Maciej Majerczyk. (2019). Database comparison: SQL vs NoSQL (MySQL vs PostgreSQL vs Redis vs MongoDB). Profil Software. <https://profil-software.com/blog/development/database-comparison-sql-vs-nosql-mysql-vs-postgresql-vs-redis-vs-mongodb/>

Amazon Web Services. (n.d.). API คืออะไร? Amazon Web Services. <https://aws.amazon.com/th/what-is/api/>

Thamakorn Torcheewee. (n.d.). AI และ Machine Learning (ML) แตกต่าง
กันอย่างไร? Cloud Ace. [https://cloud-ace.co.th/blogs/o0v9a6-ai-machine-learning-ml-
ai-ml-goog](https://cloud-ace.co.th/blogs/o0v9a6-ai-machine-learning-ml-ai-ml-goog)

Ajmera, G. (2022, December 14). *Exploring algorithms to determine
points inside or outside a polygon*. Medium.
[https://medium.com/@girishajmera/exploring-algorithms-to-determine-points-inside-
or-outside-a-polygon-038952946f87](https://medium.com/@girishajmera/exploring-algorithms-to-determine-points-inside-or-outside-a-polygon-038952946f87)

ภาคผนวก

ภาคผนวก ก. การเปรียบเทียบการทำงานระหว่าง Redis กับฐานข้อมูล

ภาคผนวก ก. การเปรียบเทียบการทำงานระหว่าง Redis

ฐานข้อมูล SQL และ NoSQL ข้อแตกต่างที่สำคัญระหว่างสองสิ่งนี้คือฐานข้อมูล SQL หรือที่เรียกว่าฐานข้อมูลเชิงสัมพันธ์ (RDBMS) มีโครงสร้างเชิงสัมพันธ์และ NoSQL ไม่ได้ใช้ความสัมพันธ์ ฐานข้อมูล SQL สามารถปรับขนาดในแนวดิ่งได้ ซึ่งหมายความว่าเครื่องที่ดีที่สุดเครื่องหนึ่งจะทำงานให้กับคุณ ในทางกลับกัน ฐานข้อมูล NoSQL สามารถปรับขนาดได้ในแนวนอน ซึ่งหมายความว่าเครื่องขนาดเล็กหลายเครื่องจะทำงาน

หลังจากรวบรวมฐานข้อมูลเพื่อการเปรียบเทียบแล้ว ฉันจำเป็นต้องใช้ภาษาที่จะช่วยขับเคลื่อนพวกเขาได้ดี ฉันไม่จำเป็นต้องพิจารณาเรื่องนี้มากนักเพราะภาษาหลักของฉันคือ Python มีตัวขับเคลื่อนที่ดีมากมายสำหรับฐานข้อมูลเหล่านี้

ใครเวอร์ใช้สำหรับฐานข้อมูลที่ระบุคือ:

MySQL -> MySQL Connector

Postgres -> psycopg2

Redis -> redis-py

MongoDB -> PyMongo

ในการจัดการฐานข้อมูลทั้งหมดในแอปพลิเคชันเดียว ฉันได้สร้าง Docker-compose ซึ่งจัดการฐานข้อมูลในรูปแบบบริการ การใช้ Docker เพื่อจัดการฐานข้อมูลทำให้ฉันสามารถละเว้นการติดตั้งทุกฐานข้อมูลในเครื่องได้

บทสรุป

การเปรียบเทียบนี้แสดงเวลาสำหรับการดำเนินการประเภทใดประเภทหนึ่งและกรณีการใช้งานฐานข้อมูลครั้งเดียว ในกรณีส่วนใหญ่ Redis มีประสิทธิภาพที่ดีที่สุดเนื่องจากเป็นเพียงพื้นที่จัดเก็บคีย์/ค่า ดังนั้นเมื่อฉันดำเนินการเพื่อค้นหาบันทึกที่มีค่าที่ระบุ มันก็สมเหตุสมผลที่ควรจะชนะ

NoSQL อาศัยการลดศีลธรรมและสร้างการปรับให้เหมาะสมสำหรับกรณีที่ผิดปกติ ตัวอย่างเช่น หากเราโพสต์บล็อก ทุกอย่างที่เกี่ยวข้องกับ ข้อความ ความคิดเห็น การถูกใจ ฯลฯ จะถูกจัดเก็บไว้ในเอกสารเดียว ดังนั้นจึงไม่จำเป็นต้องดำเนินการรวมใดๆ

ในการเปรียบเทียบนี้ เราจะเห็นว่าการดำเนินการ CRUD เดิมนั้นเร็วกว่ามากในฐานข้อมูล NoSQL แต่เรายังต้องจำไว้ว่า SQL สามารถดำเนินการได้มากกว่านั้นอีกมากมาย นอกจากนี้ความเร็วของฐานข้อมูลยังขึ้นอยู่กับแอปพลิเคชันที่คุณกำลังสร้าง

สำหรับบทความอื่นๆ เกี่ยวกับการเขียนโปรแกรม คุณสามารถดูบล็อกการพัฒนาซอฟต์แวร์ของเราได้ ซึ่งคุณสามารถเรียนรู้วิธีใช้ Django Rest Framework อย่างมีประสิทธิภาพมากขึ้น วิธีพัฒนาทักษะการเขียนโค้ด และอื่นๆ