การพัฒนาระบบ Geo-IoT และ Web Map Application สำหรับการให้บริการการแพทย์ฉุกเฉิน

ฤทัยรัตน์ หะทัยทาระ

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาภูมิสารสนเทศศาสตร์
ปีการศึกษา 2566
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

การพัฒนาระบบ Geo-IoT และ Web Map Application สำหรับการให้บริการการแพทย์ฉุกเฉิน

ฤทัยรัตน์ หะทัยทาระ

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาภูมิสารสนเทศศาสตร์
ปีการศึกษา 2566
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

วิทยานิพนธ์ เรื่อง "การพัฒนาระบบ Geo-IoT และ Web Map Application สำหรับการให้บริการการแพทย์ฉุกเฉิน"

ของ ฤทัยรัตน์ หะทัยทาระ

ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาภูมิสารสนเทศศาสตร์

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการสอบวิทยานิพนธ์

(รองศาสตราจารย์ ดร.ศราวุฒิ นิลสวัสดิ์)

..... ประธานที่ปรึกษาวิทยานิพนธ์

(รองศาสตราจารย์ ดร.สิทธิชัย ชูสำโรง)

..... กรรมการที่ปรึกษาวิทยานิพนธ์

(ผู้ช่วยศาสตราจารย์ ดร.กัมปนาท ปิยะธำรงชัย)

..... กรรมการผู้ทรงคุณวุฒิภายใน

(รองศาสตราจารย์พัฒนา ราชวงศ์)

อนุมัติ

.....

(รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์)

คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง การพัฒนาระบบ Geo-IoT และ Web Map Application สำหรับการให้บริการการแพทย์ฉุกเฉิน

ผู้วิจัย ฤทัยรัตน์ หะทัยทาระ

ประธานที่ปรึกษา รองศาสตราจารย์ ดร.สิทธิชัย ชูสำโรง

กรรมการที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.กัมปนาท ปิยะธำรงชัย

ประเภทสารนิพนธ์ วิทยานิพนธ์ วท.ม. ภูมิสารสนเทศศาสตร์, มหาวิทยาลัยนเรศวร, 2566

คำสำคัญ GPS Module, pgRouting, Web Map Application, Geo-IoT, MQTT

บทคัดย่อ

ในสังคมปัจจุบัน การเกิดอุบัติเหตุหรือการเจ็บป่วยฉุกเฉินมีแนวโน้มเพิ่มมากขึ้นเรื่อยๆ จนกลายเป็นปัญหาอย่างหนึ่งของสังคม โดยสาเหตุเกิดจากการไม่ได้รับการรักษาพยาบาลอย่างถูกต้องและทันท่วงที ซึ่งปัญหาเกิดจากผู้แจ้งเหตุบอกตำแหน่งของผู้แจ้งเหตุฉุกเฉินไม่ชัดเจนหรือการเดินทางของรถโรงพยาบาลหรือรถกู้ชีพประสบปัญหากับทางจราจรที่คับคั่ง ส่งผลให้ผู้ป่วยฉุกเฉินได้รับข้อบกพร่องทางร่างกาย จนกลายเป็นเหตุให้เสียโอกาสในการรอดชีวิตได้ การศึกษาครั้งนี้จึงได้พัฒนาระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งและพัฒนาระบบค้นหาเส้นทางที่เหมาะสมที่สุดในการเข้าถึงผู้ป่วยฉุกเฉินบน Web Map Application โดยมีวัตถุประสงค์คือ 1) ออกแบบระบบแจ้งเหตุฉุกเฉินเพื่อใช้ในการติดตามตำแหน่งของผู้แจ้งเหตุได้แบบเรียลไทม์โดยใช้การ Geo-IoT และ 2) เพื่อพัฒนาระบบค้นหาเส้นทางการให้บริการการแพทย์ฉุกเฉินบน Web Map Application และ pgRouting การพัฒนาระบบค้นหาเส้นทางนี้ได้เลือกใช้ซอฟแวร์รหัสเปิด (Free and Open-Source Software for Geospatial : FOSS4G) มาใช้ในการพัฒนาระบบทั้งหมด ส่วนเงื่อนไขในการสืบค้นและวิเคราะห์โครงสร้างถนนผู้วิจัยได้ใช้ pgRouting Library เข้ามาช่วยในการวิเคราะห์หาเส้นทางที่สั้นที่สุด โดยผลการศึกษาครั้งนี้พบว่าระบบที่พัฒนาขึ้นสามารถนำไปใช้ในการสนับสนุนงานด้านการให้บริการการแพทย์ฉุกเฉินได้ ทำให้สามารถค้นหาตำแหน่งของผู้แจ้งเหตุฉุกเฉินได้รวดเร็วกว่าเดิม

Title DEVELOPMENT OF GEO-IOT AND WEB MAP APPLICATION FOR EMERGENCY MEDICAL SERVICES.

Author Rhutairat Hataitara

Advisor Associate Professor Sittichai Choosumrong, Ph.D.

Co-Advisor Assistant Professor Kampanart Piyathamrongchai, Ph.D.

Academic Paper M.S. Thesis in Geographic Information Science, Naresuan University, 2023

Keywords GPS Module, pgRouting, Web Map Application, Geo-IoT, MQTT

ABSTRACT

Present-day society Accidents and serious illnesses tend to rise steadily over time. Until it started to affect society as a whole.The cause is due to the lack of proper and timely medical treatment.The problem arises from the informer stating the location of the emergency caller is not clear or the journey of the ambulance car or emergency rescue vehicle having problems with congested traffic. As a result, emergency patients suffer physical defects.Until becoming a cause for losing the opportunity to survive. This study therefore developed an emergency notification system with positioning sensors and developed a system to find the most suitable route to reach emergency patients on Web Map Application.With the goals of 1) creating a Geo-IoT based emergency notification system to track the informant's whereabouts in real time,and 2) To develop a route search system for emergency medical services on Web Map Application and pgRouting.The development of this route search system uses the Free and Open-Source Software for Geospatial (FOSS4G) to develop the entire system. As for conditions for searching and analyzing road structures, the researcher used the pgRouting Library to help analyze the shortest path.The result of this study revealed that the developed system can be used to support the work of providing emergency medical services.It makes it possible to find the location of the emergency caller faster than before.

ประกาศคุณูปการ

วิทยานิพนธ์ระดับปริญญาโท เรื่อง “การพัฒนาระบบ Geo-IoT และ Web Map Application สำหรับการให้บริการการแพทย์ฉุกเฉิน” ฉบับนี้ สำเร็จลุล่วงได้ด้วยดีเนื่องจากบุคคลหลายท่านได้กรุณาและให้ความอนุเคราะห์ช่วยเหลือในการดำเนินการจัดทำวิทยานิพนธ์ในครั้งนี้

ขอขอบพระคุณ รองศาสตราจารย์ ดร.สิทธิชัย ชูสำโรง อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้ และผู้ช่วยศาสตราจารย์ ดร.กัมปนาท ปิยะธำรงชัย ที่ได้สละเวลาอันมีค่าเป็นที่ปรึกษาพร้อมทั้งให้คำแนะนำและให้แนวคิดตลอดจนแก้ไขข้อบกพร่องของวิทยานิพนธ์ด้วยความเอาใจใส่ตลอดระยะเวลาในการทำวิทยานิพนธ์ฉบับนี้จนทำให้วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้อย่างสมบูรณ์และทรงคุณค่า ขอขอบพระคุณคณาจารย์สาขาวิชาภูมิสารสนเทศศาสตร์ คณะเกษตรศาสตร์ทรัพยากรธรรมชาติและสิ่งแวดล้อม ที่ช่วยให้คำแนะนำในการจัดทำระบบและถ่ายทอดความรู้วิทยาการอันมีคุณค่ายิ่ง ซึ่งเป็นประโยชน์ต่อการวิจัยและด้านการดำเนินชีวิตของผู้วิจัย และขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.บุญญาภัทร ชาติพัฒนานันท์ ที่สละเวลาให้ข้อคิดเห็นชี้แนะ รวมไปถึงการให้คำแนะนำ คำปรึกษาในการดำเนินงานจนทำให้วิทยานิพนธ์ฉบับนี้ถูกต้องและสำเร็จลุล่วงได้ดี และเหนือสิ่งอื่นใด ขอขอบพระคุณบิดามารดา และครอบครัวที่เปิดโอกาสให้ได้รับการศึกษา คอยให้กำลังใจพร้อมกับการสนับสนุนในทุกๆด้านอย่างดีที่สุดเสมอมา

คุณค่าและคุณประโยชน์อันพึงจะมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบและอุทิศแด่ผู้พระคุณทุกๆท่าน ผู้วิจัยหวังเป็นอย่างยิ่งว่างานวิจัยฉบับนี้จะเป็นประโยชน์ต่อการศึกษา “การพัฒนาระบบ Geo-IoT และ Web Map Application สำหรับการให้บริการการแพทย์ฉุกเฉิน” และผู้ที่มีความสนใจไม่มากก็น้อย

ฤทัยรัตน์ หะทัยทาระ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ค

บทคัดย่อภาษาอังกฤษ ..... ง

ประกาศคุณูปการ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตการวิจัย 3

1.4 นิยามศัพท์เฉพาะ 3

1.5 กรอบแนวความคิดการวิจัย 4

บทที่ 2 เอกสารและงานวิจัยเกี่ยวข้อง 5

2.1 ระบบแจ้งเตือนเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งด้วย Geo-IoT 5

งานวิจัยที่เกี่ยวข้องด้าน GPS และ เซนเซอร์ 10

2.2 การวิเคราะห์โครงข่ายถนน (Network Analysis) 12

งานวิจัยที่เกี่ยวข้องด้าน Network Analysis 25

2.3 ซอฟต์แวร์ที่ใช้การพัฒนา Web Map Application 27

งานวิจัยที่เกี่ยวข้องด้าน Web Map Application 40

2.4 การบริการการแพทย์ฉุกเฉิน 41
บทที่ 3 การพัฒนาระบบ 47
3.1 พื้นที่ศึกษา 47
3.2 การลงพื้นที่ภาคสนามเพื่อจัดเก็บข้อมูลและสร้างภูมินาม 48
3.3 การเตรียมข้อมูลด้วยโปรแกรม QGIS 49
3.4 นำเข้าข้อมูลเข้าสู่โปรแกรม QGIS 50
3.5 การหาค่าความลาดชันของโครงข่ายถนน เพื่อใช้ในการคำนวณหาเส้นทางด้วย pgRouting 51
3.6 การจัดเตรียมฐานข้อมูลโครงข่ายถนน 60
3.7 การจัดเตรียมข้อมูลโครงข่ายถนนในโปรแกรม pgAdmin เพื่อใช้ในการคำนวณหาเส้นทางด้วย pgRouting 61
3.8 การใช้คำสั่ง SQL ค้นหาเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgRouting 66
3.9 การออกแบบการทำงานของระบบ 68
3.10 การออกแบบการทำงานของเซนเซอร์ Ublox NEO-6M GPS 69
3.11 การพัฒนาระบบแจ้งเตือนผ่าน LINE Notify 69
3.12 การออกแบบฐานข้อมูล 72
3.13 Web Map Application for Emergence Routing Service (ERRS) 73
3.14 กระบวนการการพัฒนาระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งและการค้นหาเส้นทางสำหรับการให้บริการการแพทย์ฉุกเฉินบนระบบแผนที่ออนไลน์ 74
บทที่ 4 การทดลองระบบ 76
4.1 การพัฒนาระบบ GPS Tracking 76
4.2 ผลจากการออกแบบระบบ GPS Tracking 77

4.3 ผลการแสดงผลค่าพิกัดละติจูดและลองจิจูดผ่านบน MQTT 78
4.4 ผลการพัฒนาระบบแจ้งเตือนผ่าน LINE Notify 78
4.5 ผลจากการออกแบบ Web Map Application 80
บทที่ 5 สรุป อภิปราย และข้อเสนอแนะ 82
5.1 สรุปผล 82
5.2 อภิปรายผลการวิจัย 83
5.3 ข้อเสนอแนะ 85
บรรณานุกรม 86
ภาคผนวก 91
ภาคผนวก Source Code ที่ใช้ในการพัฒนาระบบ 92
ประวัติผู้วิจัย 129

สารบัญตาราง

หน้า

ตาราง 1 คุณสมบัติของ Ublox NEO-6M GPS Chip 6

ตาราง 2 ตารางอธิบายพารามิเตอร์การร้องขอ WMS GetCapabilities 29

ตาราง 3 ตารางอธิบายพารามิเตอร์การร้องขอ WMS Get GetMap.....30

ตาราง 4 ตารางอธิบายพารามิเตอร์การร้องขอ WMS GetFeatureInfo 30

ตาราง 5 ตารางอธิบายพารามิเตอร์การร้องขอ WMS GetCapabilities 31

ตาราง 6 แสดงรายละเอียด Value ที่จัดเก็บข้อมูล GPS และผู้ใช้งาน 72

ตาราง 7 แสดงรายละเอียด Value ที่จัดเก็บข้อมูลผู้ดูแลระบบ 73

สารบัญภาพ

หน้า

ภาพ 1 กรอบแนวการพัฒนาระบบ 4
ภาพ 2 ส่วนประกอบของ Ublox NEO-6M GPS Module + Antenna 6
ภาพ 3 ส่วนประกอบของ Ublox NEO-6M GPS Module Pinout 7
ภาพ 4 บอร์ด NodeMCU ESP8266 9
ภาพ 5 วิวัฒนาการของระบบสารสนเทศภูมิศาสตร์ 14
ภาพ 6 วิวัฒนาการของเทคโนโลยีสารสนเทศ 14
ภาพ 7 รูปแบบโครงสร้างของการคำนวณด้วย pgRouting 20
ภาพ 8 การทำงานของกราฟ Network 21
ภาพ 9 การคำนวณหาเส้นทาง ครั้งที่ 1 22
ภาพ 10 ผลการคำนวณหาเส้นทาง ครั้งที่ 1 22
ภาพ 11 การคำนวณหาเส้นทาง ครั้งที่ 2 23
ภาพ 12 ผลการคำนวณหาเส้นทาง ครั้งที่ 2 23
ภาพ 13 ผลการคำนวณหาเส้นทางจาก A-Z 23
ภาพ 14 แสดงหลักการทำงานของโปรโตคอล MQTT 24
ภาพ 15 รูปแบบไฟล์ OpenStreetMap 28
ภาพ 16 โครงสร้างไฟล์ HTML 33
ภาพ 17 ขอบเขตการปกครองตำบลแม่สลองใน 47
ภาพ 18 ขอบเขตการปกครองตำบลแม่สลองนอก 48

ภาพ 19 สอบถามข้อมูลตำแหน่งสำคัญและสร้างภูมินามที่องค์การบริหารส่วนตำบลแม่สลองนอก 48
ภาพ 20 สอบถามข้อมูลตำแหน่งสำคัญและสร้างภูมินามที่องค์การบริหารส่วนตำบลแม่สลองใน 49
ภาพ 21 ชั้นข้อมูลของถนนตำบลแม่สลองนอกและตำบลแม่สลองใน 49
ภาพ 22 ภาพไฟล์ CSV ที่ได้จากการลงพื้นที่สอบถาม 50
ภาพ 23 ภาพแสดงตำแหน่งบ้านและจุดสถานที่สำคัญ 50
ภาพ 24 นำภาพ DEM เข้าโปรแกรม QGIS 51
ภาพ 25 การสร้าง Slope ขั้นตอนที่ 1 52
ภาพ 26 การสร้าง Slope ขั้นตอนที่ 2 52
ภาพ 27 การสร้าง Slope ขั้นตอนที่ 3 53
ภาพ 28 ผลของการทำ Slope ของโครงข่ายถนน 53
ภาพ 29 การทำ Buffer ขั้นตอนที่ 1 54
ภาพ 30 ผลของการทำ Buffer โครงข่ายถนน 54
ภาพ 31 การทำ Clip Raster by Mask Layer 55
ภาพ 32 ผลลัพธ์การทำ Clip Raster by Mask Layer 55
ภาพ 33 ขั้นตอนการทำ Polygonize 56
ภาพ 34 ผลลัพธ์การทำ Polygonize 57
ภาพ 35 การสร้างตาราง curve 57
ภาพ 36 การ Join Attributes by Lacation 59
ภาพ 37 ผลการของ Join Attributes by Lacation 59
ภาพ 38 การจัดเตรียมฐานข้อมูลโครงข่ายถนน ครั้งที่ 1 60

ภาพ 39 การจัดเตรียมฐานข้อมูลโครงข่ายถนน ครั้งที่ 2 60
ภาพ 40 การจัดเตรียมฐานข้อมูลโครงข่ายถนน ครั้งที่ 3 61
ภาพ 41 คำสั่งการเพิ่มคอลัมน์ เพื่อใช้ในการคำนวณหาระยะทาง 62
ภาพ 42 คำสั่งการคำนวณระยะทาง 63
ภาพ 43 คำสั่งการตั้งค่า reverse_cost ให้เท่ากับระยะทาง 63
ภาพ 44 คำสั่งแสดงข้อมูลของตาราง way_salong 64
ภาพ 45 คำสั่งคำนวณหาค่า topology สำหรับเพิ่มค่า source,target 64
ภาพ 46 ขั้นตอนการ ADD ชั้นข้อมูลจาก pgAdmin มาใส่ในโปรแกรม QGIS 65
ภาพ 47 ข้อมูลที่ได้จาก pgAdmin ในโปรแกรม QGIS 66
ภาพ 48 การใช้คำสั่ง SQL ค้นหาเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgRouting 67
ภาพ 49 ภาพผลลัพธ์การใช้คำสั่ง SQL ค้นหาเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgRouting ที่แสดงโปรแกรม QGIS 67
ภาพ 50 ภาพรวมของระบบ SENS-IoT กับ Web Map Application 68
ภาพ 51 การต่อวงจรระหว่างบอร์ด NodeMCU ESP8266 และ Ublox NEO-6M GPS 69
ภาพ 52 หน้าต่างเพจ LINE Notify สำหรับการสมัครใช้บริการ LINE Notify 69
ภาพ 53 ลงชื่อเข้าใช้งานบนหน้าเว็บโดยใส่ข้อมูลไลน์ที่เราต้องการส่งข้อมูล 70
ภาพ 54 ขั้นตอนการออก Acess Token 70
ภาพ 55 ชื่อของ Token (ชื่อของ LINE Notify) (a) เลือกห้องแชทที่ต้องการส่งข้อความแจ้งเตือน(b) จากนั้นกดปุ่มออก Token เพื่อรับ Token key 71
ภาพ 56 เลข Token ที่ได้สามารถนำไปใช้ในการเขียนโค้ดส่งข้อมูลหรือข้อความ 71
ภาพ 57 หลักการทำงานของ pgRouting บน Web Map Application 74
ภาพ 58 แผนผังการทำงานของระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่ง 75

ภาพ 59 ข้อมูลที่ส่งเข้าฐานข้อมูลในเครื่องแม่ข่าย....76
ภาพ 60 ตำแหน่ง GPS Tracking ที่ดึงข้อมูลจากฐานข้อมูลมาแสดงบนแผนที่บนเว็บไซต์.77
ภาพ 61 ระบบ GPS Tracking สำหรับแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่ง....77
ภาพ 62 ข้อมูล GPS Tracking ที่เข้า MQTT โดย (a) เป็นค่าพิกัดละติจูด และ
(b) เป็นค่าพิกัดลองจิจูด....78
ภาพ 63 การแจ้งเตือนผ่าน LINE Notify ไปยังสมาร์ทโฟนของครอบครัว (a) และเจ้าหน้าที่
ผู้ที่ดูแลระบบ (b)....79
ภาพ 64 โฟลว์ควบคุมเงื่อนไขและการทำงานบน Node-RED....79
ภาพ 65 หน้าลงทะเบียนเข้าระบบ....80
ภาพ 66 หน้าต่าง Login เข้าระบบ....80
ภาพ 67 หน้าต่างแสดงผลลัพธ์การค้นหาเส้นทางที่สั้นที่สุด....81

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบบริการการแพทย์ฉุกเฉิน (Emergency Medical System) หรือ EMS ได้ถูกพัฒนาขึ้นเพื่อช่วยลดจำนวนการสูญเสีย การทุกข์ทรมาน และการสิ้นเปลืองค่าใช้จ่าย อันเกิดจากความล่าช้าในการลำเลียงเคลื่อนย้ายผู้ป่วยไม่ถูกวิธี โดยการจัดทำระบบรับแจ้งเหตุที่ประชาชนสามารถเข้าถึงได้ง่าย มีความรู้ในการดำเนินการ และเข้าใจในระบบการแพทย์ฉุกเฉิน จึงต้องอาศัยความร่วมมือของผู้นำท้องถิ่นหรือชุมชน เพื่อให้ปฏิบัติการฉุกเฉินได้อย่างถูกวิธี ทันท่วงทีและสามารถช่วยเหลือผู้ป่วยฉุกเฉินไม่ให้เสียชีวิต เพื่อให้ผู้ป่วยได้รับประโยชน์จากระบบการแพทย์ฉุกเฉินที่มีคุณภาพและมาตรฐานในการเข้าถึงระบบบริการการแพทย์ฉุกเฉินอย่างทั่วถึงและเท่าเทียมกัน ตามพระราชบัญญัติการแพทย์ฉุกเฉิน พ.ศ. 2551 ได้กำหนดเป้าหมายที่สำคัญ คือ ลดการเสียชีวิต ความพิการ และอาการแทรกซ้อนจากการเจ็บป่วยฉุกเฉิน จึงต้องมีการพัฒนาระบบการแพทย์ฉุกเฉิน (สถาบันการแพทย์ฉุกเฉินแห่งชาติ, 2556)

หลักการสำคัญในการพัฒนาระบบการแพทย์ฉุกเฉิน ตามแผนหลักการการแพทย์ฉุกเฉินแห่งชาติ ฉบับที่ 1 เป็นการพัฒนาให้มีความก้าวหน้าอย่างต่อเนื่อง เพื่อให้สามารถเข้าถึงและให้บริการแก่ผู้ป่วยฉุกเฉินได้อย่างทันท่วงที และมีความครอบคลุมในทุกๆ ด้าน เช่น ด้านการป้องกันการเจ็บป่วยฉุกเฉินให้เกิดขึ้นน้อยที่สุด การจัดการให้ผู้ป่วยฉุกเฉินได้รับการปฏิบัติการฉุกเฉินที่ได้มาตรฐาน การพ้นภาวะฉุกเฉิน หรือได้รับการบำบัดรักษาเฉพาะอย่างทันท่วงที เป็นต้น ผู้ป่วยฉุกเฉินต้องได้รับการบริการที่ได้มาตรฐานและมีประสิทธิภาพ ซึ่งจำเป็นต้องได้รับความร่วมมือจากหน่วยงานที่เกี่ยวข้องทั้งในภาครัฐและเอกชน โดยที่องค์กรปกครองส่วนท้องถิ่นต้องเข้ามามีบทบาทในการดำเนินงาน เพื่อให้เกิดความร่วมมือในการปฏิบัติการฉุกเฉินร่วมกัน ซึ่งจะทำให้ผู้ป่วยฉุกเฉินได้รับการคุ้มครองสิทธิในการเข้าถึงระบบการแพทย์ฉุกเฉินที่ได้มาตรฐาน และมีประสิทธิภาพอย่างทั่วถึงและเท่าเทียมกัน (การแพทย์ฉุกเฉินแห่งชาติ, 2553)

เนื่องจากเทคโนโลยีสารสนเทศในยุคปัจจุบันมีการพัฒนาเพิ่มมากขึ้น ส่งผลให้ความก้าวหน้าทางเทคโนโลยีสูงขึ้นตามไปด้วยและเข้ามามีบทบาทกับวิถีชีวิตของมนุษย์มากยิ่งขึ้นเช่นกัน โดยเทคโนโลยีภูมิสารสนเทศหรือ Geo-Informatics เป็นสารสนเทศอีกประเภทหนึ่งที่เป็นที่รู้จักและใช้งานมากในปัจจุบัน ตัวอย่างเช่น แผนที่ออนไลน์ (Google Map, Google Earth, Google Street View ฯลฯ) และระบบอินเทอร์เน็ต หรือจะเป็นโปรแกรมแบบ OpenSource เช่น PostgreSQL/PostGIS, pgRouting, QGIS, Apache, PHP, HTML, OpenStreetMap และอื่นๆ ซึ่ง

ในปัจจุบันได้นำเทคโนโลยีมาประยุกต์ใช้ในชีวิตประจำวันมากยิ่งขึ้น รวมถึงนำมาประยุกต์ในด้านการค้นหาเส้นทางที่เหมาะสม เป็นต้น

ปัจจุบันในประเทศไทยมีหลายช่องทางการติดต่อการใช้บริการสถาบันการแพทย์ฉุกเฉินได้หลายช่องทาง เช่น ทางโทรศัพท์สายด่วน 1669, การใช้แอปพลิเคชัน ThaiEMS 1669, Mobile Application 1669 และ Application TTRS Video (call center ใช้สำหรับผู้บกพร่องทางการได้ยินและการพูด) เป็นต้น แต่อย่างไรก็ตามการใช้โทรศัพท์ทางสายด่วย 1669 หรือ Mobile Application 1669 เพื่อแจ้งจุดเกิดเหตุฉุกเฉินโดยผู้สูงอายุที่มีปัญหาในลักษณะดังกล่าวอาจมีปัญหาและใช้เวลานานในการสื่อสารเพื่อหาจุดเกิดเหตุ เนื่องจากผู้แจ้งเบาะแสอาจควบคุมสติไม่ได้ การบอกสถานที่เกิดเหตุไม่ถูกต้องครบถ้วน ทำให้พนักงานขับรถเลือกใช้เส้นทางที่เข้าถึงตำแหน่งของผู้ป่วยได้ล่าช้า อาจทำให้ผู้ป่วยได้รับการรักษาพยาบาลที่ล่าช้าและไม่ทันเวลา จนเป็นเหตุสูญเสียชีวิตหรืออวัยวะบางส่วนของร่างกาย ส่งผลต่อความสามารถในการทำงานตามปกติและยังอาจทำให้การบาดเจ็บหรืออาการเจ็บป่วยเล็กน้อยรุนแรงขึ้น ปัญหาเหล่านี้จึงเป็นเรื่องสำคัญที่จะต้องหาทางแก้ไขเพื่อให้กลุ่มผู้สูงอายุสามารถเข้าถึงบริการการแพทย์ฉุกเฉินได้อย่างสะดวกและรวดเร็ว นอกจากนี้ยังสามารถช่วยให้รถโรงพยาบาลหรือรถกู้ชีพฉุกเฉินเลือกใช้เส้นทางที่ใกล้ที่สุดหรือเหมาะสมที่สุดไปยังจุดเกิดเหตุฉุกเฉินได้โดยเร็วที่สุด

จากปัญหาของการเข้าถึงผู้ป่วยฉุกเฉินดังกล่าว งานวิจัยและพัฒนาฉบับนี้จึงได้จัดทำระบบค้นหาเส้นทางการให้บริการการแพทย์ฉุกเฉิน โดยการใช้อัลกอลิทึม pgRouting, Web Map Application และอุปกรณ์ Geo-IoT เพื่อพัฒนาเครื่องมือแบบ Open Source ที่มีความสามารถค้นหาจุดที่เกิดอุบัติเหตุฉุกเฉินและกำหนดเส้นทางที่สั้นที่สุดหรือเหมาะสมที่สุดไปยังจุดเกิดเหตุโดยใช้เทคโนโลยีระบบสารสนเทศภูมิศาสตร์ (GIS) ระบบกำหนดตำแหน่งบนพื้นโลก (GPS) และอินเทอร์เน็ตของสรรพสิ่ง (IoT โดยนำเทคโนโลยี IoT และเครือข่ายเซนเซอร์ไร้สายมาพัฒนาระบบการแจ้งเตือนฉุกเฉินโดยใช้เซนเซอร์ Ublox NEO-6M GPS, MQTT และ Node-RED ซึ่งระบบที่พัฒนาขึ้นจะส่งการแจ้งเตือนไปยังสมาชิกในครอบครัวหรือบุคลากรที่รับผิดชอบจัดการระบบบริการการแพทย์ฉุกเฉินผ่าน LINE Notify ได้อย่างง่าย ในขณะเดียวกันจะส่งข้อมูลไปยัง MQTT และแสดงผลบน Node-RED เพื่อส่งข้อมูลไปยังฐานข้อมูล และข้อมูลถูกส่งไปยังฐานข้อมูลแล้ว ระบบจะทำการกำหนดเส้นทางบน Web Map Application และทำการคำนวณเส้นทางที่สั้นที่สุดหรือเหมาะสมที่สุดไปยังจุดเกิดเหตุในรูปแบบ Web Map Service ต่อไป

1.2 วัตถุประสงค์

1. เพื่อพัฒนานวัตกรรมการแจ้งเหตุฉุกเฉินด้วย Geo-IoT
2. เพื่อพัฒนาระบบค้นหาเส้นทางการให้บริการการแพทย์ฉุกเฉินบนระบบแผนที่ออนไลน์

1.3 ขอบเขตการวิจัย

ในการศึกษางานวิจัยครั้งนี้ ได้ทำการศึกษาการพัฒนาระบบ Smart Emergency Notification System based on IoT (SENS-IoT) และ การพัฒนาระบบค้นหาตำแหน่งเส้นทางการให้บริการการแพทย์ฉุกเฉินบนระบบแผนที่ออนไลน์ โดยการวิจัยและพัฒนา ในส่วนการวิจัยจะศึกษาความต้องการการใช้งานของระบบแจ้งเหตุว่าต้องการให้ระบบออกมาเป็นอย่างไร ประกอบด้วยข้อมูลสำคัญอะไรบ้าง สำหรับการพัฒนา จะพัฒนาอุปกรณ์ต้นแบบสำหรับแจ้งเหตุด้วย IoT และพัฒนาระบบค้นหาเส้นทางสำหรับการให้บริการการแพทย์ฉุกเฉินบนระบบแผนที่ออนไลน์ ทั้งนี้งานวิจัยนี้ไม่ได้ลงลึกในเรื่องของการศึกษาพฤติกรรมของผู้ใช้งาน ข้อจำกัดในเรื่องของสัญญาณอินเทอร์เน็ต ค่าบริการสัญญาณ 3G ในการรับส่งข้อมูลจะอยู่นอกเหนือจากขอบเขตงานวิจัย

1.4 นิยามศัพท์เฉพาะ

1. Ublox NEO-6M GPS เป็นโมดูล NEO-6M สำหรับดึงพิกัดตำแหน่งปัจจุบัน โดยดึงค่าจากดาวเทียม ให้ค่าเป็นละติจูดและลองจิจูดสำหรับระบุตำแหน่งสื่อสารแบบ Serial Rx,Tx ใช้งานง่ายมีไลบรารีพร้อมใช้งาน ใช้ไฟเลี้ยง 3.3 - 5V

2. pgRouting เป็นเครื่องมือที่ใช้ในการคำนวณหาระยะทางและวางแผนเส้นทางในระบบข้อมูลทางพื้นที่โดยใช้ฐานข้อมูล PostgreSQL/PostGIS

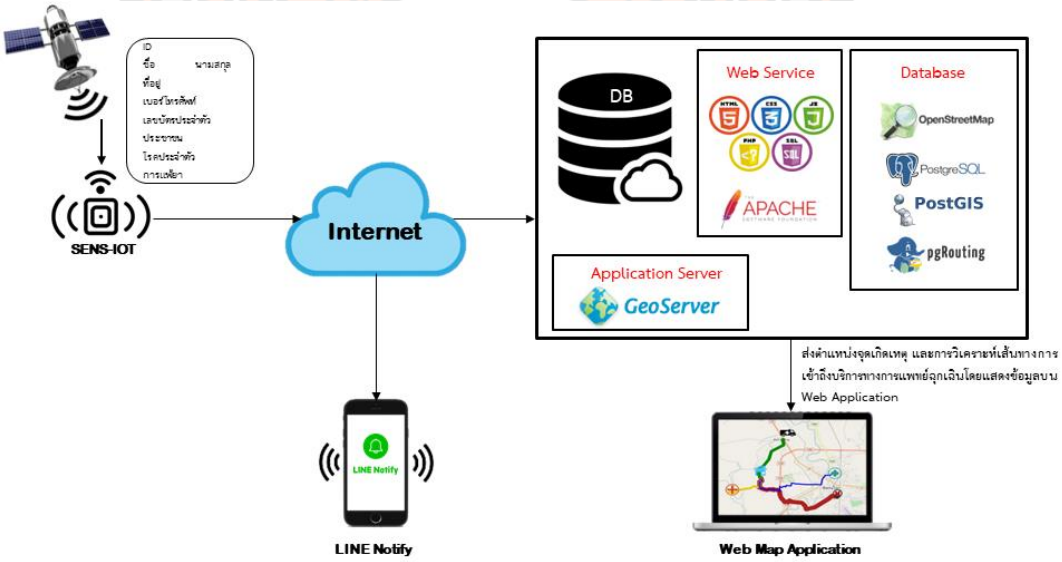
3. Web Map Application หมายถึง ข้อมูลที่แสดงลงบนแผนที่ที่มีความซับซ้อนของชั้นข้อมูล และรวมไปถึงข้อมูลที่ผ่านการวิเคราะห์ทางด้าน GIS แล้วนำเสนอในรูปแบบแผนที่ออนไลน์ ซึ่งระบบแม่ข่าย (Server) ก็จะต้องใช้โปรแกรมที่รองรับด้าน Web Map Service

4. Geo-IoT หมายถึง "Geospatial Internet of Things" หรือ "Internet of Things ที่เกี่ยวข้องกับพื้นที่" ซึ่งเป็นแนวคิดที่รวมระบบ Internet of Things (IoT) กับเทคโนโลยีที่เกี่ยวข้องกับพื้นที่หรือที่ตั้งทางภูมิศาสตร์ (Geospatial Technology) เข้าด้วยกันเพื่อสร้างความสัมพันธ์ระหว่างข้อมูลที่สร้างขึ้นโดยอุปกรณ์ IoT และข้อมูลเชิงพื้นที่หรือทางภูมิศาสตร์ เช่น ข้อมูลเกี่ยวกับตำแหน่งทางภูมิศาสตร์ (GPS coordinates) หรือข้อมูลที่เกี่ยวข้องกับพื้นที่

5. MQTT คือ โปรโตคอลในการส่งข้อมูลที่พัฒนามาเพื่อใช้ในระบบ IoT โดยปลายทางจะทำหน้าที่เป็น Client ซึ่งทำการสร้างเชื่อมต่อแบบ TCP ไปยัง Server ที่มีชื่อเรียกอีกชื่อว่า Broker ซึ่งมีหน้าที่เป็นเสมือนตัวกลางการส่งข้อมูลในการรับส่ง “Message” ระหว่าง Client ที่เป็นได้ทั้ง Publisher และ Subscriber มันถูกออกแบบให้สามารถส่งข้อมูลได้แบบ Real-Time

1.5 กรอบแนวความคิดการวิจัย

ผู้วิจัยใช้โมดูล Ublox NEO-6M GPS เป็นเครื่องตรวจจับตำแหน่งพิกัดจุดเกิดเหตุ พิกัดตำแหน่งจะส่งข้อมูลจาก Ublox NEO-6M GPS ผ่าน NodeMCU ESP8266 เข้าสู่ระบบฐานข้อมูลกลาง การทำงานของระบบ คือ เมื่อผู้สูงอายุเกิดอุบัติเหตุแล้วทำการกดปุ่มอุปกรณ์ SENS-IoT แจ้งเหตุเข้ามาในระบบและส่งการแจ้งเตือนผ่าน Line Notify เข้าสู่โทรศัพท์มือถือของเจ้าหน้าที่และสมาชิกครอบครัว จากนั้นเจ้าหน้าที่จะทำการเข้า Web Map Application ใส่เลขไอดีของผู้แจ้งเหตุฉุกเฉิน เมื่อระบบได้รายละเอียดของผู้แจ้งเหตุ อาการ ตำแหน่งที่แจ้งเหตุ ระบบค้นหาเส้นทางที่พัฒนาด้วย pgRouting ก็จะทำการคำนวณหาเส้นทางจากรถกู้ชีพและรถโรงพยาบาลที่ใกล้ที่สุดไปยังจุดแจ้งเหตุที่เร็วที่สุด ซึ่งระบบนี้สามารถกำหนดเงื่อนไขการค้นหาเส้นทางได้หลากหลายปัจจัย ขึ้นอยู่กับการคำนึงถึงความปลอดภัยของชีวิตผู้แจ้งเหตุมากที่สุดในการเดินทาง ในขณะเดียวกันระบบก็จะทำการคำนวณเส้นทางจากจุดเกิดเหตุไปยังโรงพยาบาลที่เหมาะสมที่สุด โดย GeoServer ทำหน้าที่เป็นผู้ให้บริการแผนที่ออนไลน์ในรูปแบบ Web Map Service (WMS) และแสดงข้อมูลบน Web Map Application ทำให้เจ้าหน้าที่ EMS สามารถทราบตำแหน่งจุดเกิดเหตุและเส้นทางที่ทำให้เข้าถึงตำแหน่งของผู้แจ้งเหตุได้รวดเร็วมากยิ่งขึ้น ทำให้ผู้แจ้งเหตุได้รับการรักษาพยาบาลที่ทันเวลา ซึ่งข้อมูลจะแสดงผ่านระบบแผนที่ออนไลน์พร้อมข้อมูลของตำแหน่งจุดเกิดเหตุแสดงผ่านบน Web Map Application ทำให้หน่วยปฏิบัติการ EMS สามารถออกปฏิบัติการไปยังจุดเกิดเหตุได้อย่างแม่นยำได้ แม้เจ้าหน้าที่จะไม่เคยชินกับพื้นที่ ซึ่งจะช่วยให้ผู้ป่วยฉุกเฉินที่เกิดเจ็บป่วยฉุกเฉินเข้าถึงบริการ EMS อย่างรวดเร็วและประสิทธิภาพมากยิ่งขึ้น



ภาพ 1 กรอบแนวการพัฒนาระบบ

บทที่ 2
เอกสารและงานวิจัยเกี่ยวข้อง

การพัฒนาระบบแจ้งเตือนเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งด้วย Geo-IoT ที่แสดงข้อมูลตำแหน่งที่ตั้งของจุดที่ผู้แจ้งเหตุแบบเรียลไทม์ร่วมกับระบบค้นหาเส้นทางจากตำแหน่งรถโรงพยาบาลและรถกู้ชีพ ไปยังตำแหน่งที่แจ้งเหตุ ผู้วิจัยได้ทำการศึกษาค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้องตามลำดับดังนี้

1. ระบบแจ้งเตือนเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งด้วย Geo-IoT
2. การวิเคราะห์โครงข่ายถนน (Network Analysis)
3. ซอฟต์แวร์ที่ใช้การพัฒนา Web Map Application
4. การบริการการแพทย์ฉุกเฉิน

2.1 ระบบแจ้งเตือนเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งด้วย Geo-IoT

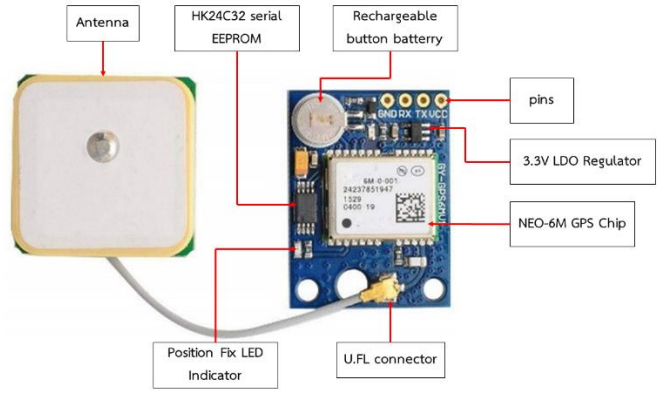
2.1.1 GPS (Global Positioning System)

GPS ย่อมาจาก Global Positioning System เป็นระบบที่ระบุตำแหน่งบนพื้นผิวโลก โดยอาศัยการคำนวณพิกัดจากดาวเทียมระบุตำแหน่ง จำนวน 24 ดวง ที่โคจรอยู่รอบโลกในระดับสูงประมาณ 20,000 กิโลเมตร ทำให้สามารถชี้บอกตำแหน่งได้ทุกแห่งบนโลกตลอดเวลา 24 ชั่วโมง ซึ่งถ้าเรามีอุปกรณ์รับข้อมูลติดตั้งอยู่จะทำให้สามารถแสดงตำแหน่งนั้นได้อย่างแม่นยำ ปัจจุบันนี้ด้วยความสามารถของ GPS ทำให้เราสามารถนำข้อมูลตำแหน่งมาใช้ประโยชน์ได้อย่างมากมาย เช่น การหาตำแหน่งที่แน่นอนบนพื้นโลกของเรา ป้องกันการหลงทาง การหาจุดอ้างอิงต่างๆ เช่น ร้านอาหาร โรงพยาบาล เป็นต้น หรือใช้ในการแนะนำเส้นทางไปยังจุดต่างๆ บนโลก ที่เรียกว่า “ระบบนำทาง” หรือ “Navigator” ซึ่งมีให้กันใช้อยู่ทั่วไปในปัจจุบัน นอกจากนี้ยังสามารถนำมาใช้ในการติดตามบุคคลหรือติดตามยานพาหนะ หรือที่เรียกว่า “Tracking GPS” เพื่อใช้ในการตรวจสอบเส้นทางการเดินทางได้อีกด้วย

2.1.2 Ublox NEO-6M GPS Modules

Ublox NEO-6M GPS เป็นหนึ่งในโมดูล GPS ที่เป็นส่วนหนึ่งของ UBLOX GPS Series สามารถติดตามดาวเทียมได้ถึง 22 ดวงใน 50 ช่องสัญญาณและบรรลุระดับความไวสูงสุดของอุตสาหกรรม เช่น การติดตาม -161 dB ในขณะที่ใช้กระแสไฟเพียง 45mA ไม่เหมือนกับโมดูล GPS อื่น ๆ มันสามารถอัปเดตตำแหน่งสูงสุด 5 ตำแหน่งต่อวินาทีด้วยความแม่นยำของตำแหน่งแนวนอน 2.5 ม. การกำหนดตำแหน่ง u-blox 6 ยังมี Time-To-First-Fix (TTFF) ที่น้อยกว่า 1 วินาที หนึ่งในคุณสมบัติที่ดีที่สุดที่ชิปมีให้คือโหมดประหยัดพลังงาน (PSM) ช่วยลดการใช้พลังงานของระบบโดยการ

เลือกสวิตช์เปิดและปิดส่วนต่างๆ ของเครื่องรับ ซึ่งช่วยลดการใช้พลังงานของโมดูลลงอย่างมากให้เหลือเพียง 11mA ทำให้เหมาะสำหรับการใช้งานที่ไวต่อพลังงาน เช่น นาฬิกาข้อมือ GPS



ภาพ 2 ส่วนประกอบของ Ublox NEO-6M GPS Module + Antenna

ตาราง 1 คุณสมบัติของ Ublox NEO-6M GPS Chip

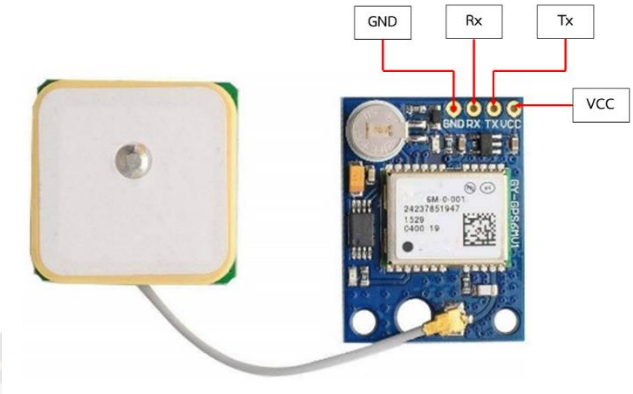
Receiver Type	50 channels, GPS L1(1575.42Mhz)
Horizontal Position Accuracy	2.5m
Navigation Update Rate	1HZ (5Hz maximum)
Capture Time	Cool start: 27sHot start: 1s
Navigation Sensitivity	-161dBm
Communication Protocol	NMEA, UBX Binary, RTCM
Serial Baud Rate	4800-230400 (default 9600)
Operating Temperature	-40°C ~ 85°C
Operating Voltage	2.7V ~ 3.6V
Operating Current	45mA
TXD/RXD Impedance	510Ω

2.1.3 Ublox NEO-6M GPS Module Pinout

โมดูล Ublox NEO-6M GPS มีทั้งหมด 4 ขาสำหรับการเชื่อมต่อกับบอร์ด โดยมีการเชื่อมต่อของแต่ขาละดังนี้

- GND เป็นขาที่จำเป็นต่อการเชื่อมต่อ ซึ่งขา GND จะทำการเชื่อมต่อกับขา GND บนบอร์ด NodeMCU หรือบนบอร์ด Arduino
- Rx เป็นขาสำหรับการสื่อสารแบบอนุกรม

- Tx เป็นขาสำหรับการสื่อสารแบบอนุกรม
- VCC เป็นแหล่งจ่ายพลังงานให้กับโมดูล คุณสามารถเชื่อมต่อโดยตรงกับขา 3.3V บนบอร์ด NodeMCU หรือเชื่อมต่อโดยตรงกับขา 5V บนบอร์ด Arduino



ภาพ 3 ส่วนประกอบของ Ublox NEO-6M GPS Module Pinout

2.1.4 NodeMCU

NodeMCU (โหนด เอ็มซียู) คือ บอร์ดคอนโทรลเลอร์ทีมีลักษณะการทำงานตามคำสั่งภาษา C คล้าย Arduino แต่มีลักษณะพิเศษกว่าตรงที่สามารถเชื่อมต่อกับ WiFi ได้ การควบคุมการทำงานสามารถใช้โปรแกรม Arduino IDE ได้เช่นเดียวกันกับบอร์ด Arduino

บอร์ดของ NodeMCU ประกอบไปด้วย ESP8266 (ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อ WiFi ได้) พร้อมอุปกรณ์อำนวยความสะดวกต่างๆ เช่น พอร์ต micro USB สำหรับจ่ายไฟ/อัปโหลดโปรแกรม, ชิพ สำหรับอัปโหลดโปรแกรมผ่านสาย USB, ชิพแปลงแรงดันไฟฟ้าและขาสำหรับเชื่อมต่ออุปกรณ์ภายนอก เป็นต้น ซึ่งในตอนนี้ทาง NodeMCU จะออกมาอีกสองรุ่น คือ รุ่น 0.9 กับ รุ่น 1.0 ทางผู้พัฒนาตั้งใจจะออก NodeMCU ให้เป็น platform ที่ออกแบบทุกอย่างเป็น Node การทำงานย่อยๆ และใช้ภาษา Lua ในการเขียนโปรแกรม แต่ด้วย platfrom ที่สะดวกในการใช้งาน ทางกลุ่มนักพัฒนาของ ESP8266 ก็เลยนำ NodeMCU (ESP8266) มาบรรจุในบอร์ดหนึ่งของ ARDUINO IDE (ESP8266) ด้วย จึงได้มีการพัฒนาต่อให้สามารถเขียนในภาษา C/C++ ซึ่งนักพัฒนาด้านไมโครคอนโทรลเลอร์คุ้นเคยกว่าภาษา Lua จึงทำให้ ได้รับความนิยมทดลองใช้กันอย่างกว้างขวาง

ข้อดีของบอร์ด Arduino ESP8266

- เป็นแบบ Open Source Project มี Source code ให้ได้เรียนรู้อยู่บน Github ตามลิงค์ https://github.com/esp8266/Arduino
- สามารถกด upload sketch ได้ เชื่อมต่อบอร์ด USB กับคอมพิวเตอร์ใช้งานง่าย ขนาดของบอร์ดต่อลง Protoboard ได้

- ชิบภายใน ESP 8266 มี CPU ขนาด 32 bit แตกต่างจาก Arduino ที่เป็น CPU 8 bit
- ถึงแม้ขา I/O จะไม่มากเท่าของ Arduino แต่เราสามารถเขียนโปรแกรมลงบนขา GPIO ได้ทุกขาพอๆกัน เป็นข้อดีที่เพิ่มมาจากความต้องการใช้ WIFI เชื่อมต่อเมื่อต้องการเล่น Arduino ทำให้ต้องซื้อ Module Wi-Fi เพิ่ม นั่นคือ NodeMCU (ESP8266) มีต้นทุนต่ากว่ามาก
- มีอุปกรณ์หลายอย่างที่ใช้งานที่แรงดัน +3.3 V เป็นส่วนใหญ่ ดังนั้นเราสามารถนำ NodeMCU (ESP8266) มาใช้เชื่อมต่อได้โดยตรง

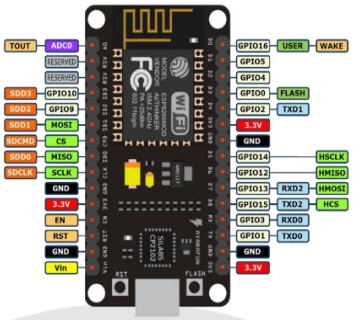
ESP8266 คือโมดูล Wi-Fi จากจีน ที่มีความพิเศษตรงที่ตัวมันสามารถโปรแกรมลงไปได้ ทำให้สามารถนำไปใช้งานแทนไมโครคอนโทรลเลอร์ได้เลย และมีพื้นที่โปรแกรมที่มากถึง 4MB ทำให้มีพื้นที่เหลือมากในการเขียนโปรแกรมลงไป

ESP8266 เป็นชื่อของชิปไอซีบนบอร์ดของโมดูล ซึ่งไอซี ESP8266 ไม่มีพื้นที่โปรแกรม (flash memory) ในตัว ทำให้ต้องใช้ไอซีภายนอก (external flash memory) ในการเก็บโปรแกรม ที่ใช้การเชื่อมต่อผ่านโปรโตคอล SPI ซึ่งสาเหตุนี้เองทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่าไอซีไมโครคอนโทรลเลอร์เบอร์อื่นๆ

ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3V - 3.6V การนำไปใช้งานร่วมกับเซ็นเซอร์อื่นๆที่ใช้แรงดัน 5V ต้องใช้วงจรแบ่งแรงดันมาช่วย เพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200mA ความถี่คริสตอล 40MHz ทำให้เมื่อนำไปใช้งานอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การแสดงผลข้อมูลรวดเร็วกว่าไมโครคอนโทรลเลอร์ยอดนิยม Arduino มาก โดยขาของโมดูล ESP8266 แบ่งได้ดังนี้

- VCC เป็นขาสำหรับจ่ายไฟเข้าเพื่อให้โมดูลทำงานได้ซึ่งแรงดันที่ใช้งานได้ คือ 3.3 - 3.6V
- GND
- Reset และ CH_PD (หรือ EN) เป็นขาที่ต้องต่อเข้าไฟ + เพื่อให้โมดูลสามารถทำงานได้ทั้ง 2 ขานี้สามารถนำมาใช้รีเซ็ตโมดูลได้เหมือนกัน แตกต่างตรงที่ขา Reset สามารถลอยไว้ได้ แต่ขา CH_PD (หรือ EN) จำเป็นต้องต่อเข้าไป + เท่านั้น เมื่อขานี้ไม่ต่อเข้าไฟ + โมดูลจะไม่ทำงานทันที
- GPIO เป็นขาดิจิตอลอินพุต/เอาต์พุต ทำงานที่แรงดัน 3.3V
- GPIO15 เป็นขาที่ต้องต่อลง GND เท่านั้น เพื่อให้โมดูลทำงานได้
- GPIO0 เป็นขาสำหรับการเลือกโหมดทำงาน หากนำขานี้ลง GND จะเข้าโหมดโปรแกรม หากลอยไว้หรือนำเข้าไฟ + จะเข้าโหมดการทำงานปกติ

- ADC เป็นขาอนาล็อกอินพุต รับแรงดันได้สูงสุดที่ 1V ขนาด 10 บิต การนำไปใช้งานกับแรงดันที่สูงกว่าต้องใช้วงจรแบ่งแรงดันเข้าช่วย



ภาพ 4 บอร์ด NodeMCU ESP8266

ที่มา: https://medium.com/@pattanapong.sriph

2.1.5 Internet of Things (IoT)

Internet of Things คือ เทคโนโลยีที่เชื่อมต่อสิ่งของหรืออุปกรณ์ ต่างๆ เข้าสู่ระบบอินเทอร์เน็ต สิ่งของต่างๆ เช่น รถยนต์ ตู้เย็น โทรทัศน์ สามารถเชื่อมโยงและสื่อสารกันได้ด้วยเทคโนโลยีนี้ทำให้ผู้ใช้ทั่วไปสามารถเข้าถึงและควบคุมสิ่งของต่างๆ จากในบ้านที่ทำงานหรือที่ใดๆ ซึ่งเป็นการยกระดับความสะดวกสบายในชีวิตประจำวัน ยกตัวอย่างเช่น เจ้าของบ้านสามารถเปิดเครื่องปรับอากาศภายในบ้านก่อนกลับบ้าน เป็นต้น

เทคโนโลยีทางด้าน IoT ที่นำมาประยุกต์ใช้ในปัจจุบันนี้ แบ่งเป็น 3 ประเภท ประเภทแรก คือ เทคโนโลยีที่ทำให้สรรพสิ่งสามารถรับรู้ข้อมูล เมื่อมีการปฏิสัมพันธ์หรือเข้าใกล้ เช่น เซนเซอร์ (sensor) ซึ่งทำหน้าที่ตรวจสอบและตอบสนองต่ออินพุต (input) ต่างๆ โดยอินพุตมีหลายประเภท ได้แก่ แสง การเคลื่อนไหว ความชื้น ความดัน เป็นต้น หลังจากเซนเซอร์ได้รับอินพุตจะมีการสร้างเอาต์พุต (output) เป็นสัญญาณโดยสัญญาณที่สร้างขึ้นจะถูกแปลงไปเป็นผลลัพธ์ที่มนุษย์สามารถเข้าใจได้

ประเภทที่สอง คือ เทคโนโลยีที่ทำให้สรรพสิ่งสามารถติดต่อสื่อสารกันได้ เช่น ระบบฝังตัว (Embedded System) ที่ฝังอยู่ในเครื่องมืออิเล็กทรอนิกส์หรือเครื่องใช้ไฟฟ้า เพื่อทำให้อุปกรณ์เหล่านั้นสามารถทำตามคำสั่งได้ โดยคำสั่งต่างๆ จะถูกเขียนในชิป (Chip) หรือไมโครโพรเซสเซอร์ (Microprocessor) ที่ถูกฝังไว้ใน embedded System นอกจากนี้ เทคโนโลยีที่ทำให้อุปกรณ์สามารถติดต่อสื่อสารกันได้ ยังรวมถึงการสื่อสารแบบไร้สายที่ใช้พลังงานต่ำ เช่น Zigbee, Bluetooth, Low Power Wide Area Network (LPWAN) เป็นต้น ซึ่งเป็นเทคโนโลยีที่ทำให้เครือข่ายเชื่อมต่อในระยะทางไกลได้โดยใช้พลังงานต่ำ ซึ่งเหมาะกับการสื่อสารระหว่าง IoT

ประเภทที่สาม คือ เทคโนโลยีที่ช่วยให้สรรพสิ่งประมวลผลข้อมูล เช่น การประมวลผลแบบคลาวด์ (Cloud Computing) เป็นบริการที่ให้ใช้กำลังประมวลผล หน่วยจัดเก็บข้อมูล และโปรแกรมออนไลน์ต่างๆ จากผู้ให้บริการ อีกเทคโนโลยีหนึ่งที่เป็นที่นิยม คือ การวิเคราะห์ข้อมูลขนาดใหญ่ (Big Data Analytics) เป็นกระบวนการวิเคราะห์ข้อมูลขนาดใหญ่เพื่อค้นหารูปแบบความเชื่อมโยงของข้อมูลเหล่านั้น ผลจากการวิเคราะห์ สามารถนำไปสู่การวางแผนการดำเนินธุรกิจได้อย่างมีประสิทธิภาพ

2.1.6 LINE Notify

LINE Notify เป็นบริการของทาง LINE เป็นบริการและช่องทางที่สามารถส่งข้อความการแจ้งเตือนต่างๆ ไปยังบัญชีของผู้ใช้งานได้ ผ่านการใช้ API ซึ่งเรียกผ่าน HTTP POST โดย LINE Notify สามารถส่งแจ้งเตือนได้เฉพาะผู้ที่ขอใช้ หรือกลุ่มที่ผู้ขอใช้เป็นสมาชิกเท่านั้น ไม่สามารถส่งข้อความเข้าห้องสนทนาของเพื่อน ๆ ได้ หากต้องการให้สามารถส่งข้อความหาใครก็ได้ต้องใช้ LINE Bot API แทน

1. ในการเพิ่ม LINE Notify เป็นเพื่อนก่อนที่จะใช้งานและส่งการแจงเตือนต้องเพิ่ม LINE Notify โดยสแกน QR Code

2. การขอ Access Token ในการใช้งาน API ในทุก ๆ บริการ จะมีสิ่งที่เรียกว่า Access Token ไว้สำหรับเป็นรหัสที่ใช้ตอนจะเข้าใช้งาน API โดยรหัสนี้จะเป็นข้อความแทนจดหมายอิเล็กทรอนิกส์ และรหัสผ่านของผู้ใช้ ดังนั้นหาก Access Token ถูกเปิดเผยใช้งานได้ปกติด้วยการเข้าไปที่หน้าเว็บ https://notify-bot.line.me/my/ จากนั้นระบบจะใช้ผู้ใช้ล็อกอินด้วยบัญชีผู้ใช้งาน LINE โดยกรอกอีเมล์ และรหัสผ่านที่ได้ตั้งไว้

เมื่อผู้ใช้ส่งข้อความไปแล้ว ระบบจะส่งข้อความในรูปแบบ [ชื่อ Token]: [ข้อความ] ดังนั้นที่กรอกข้อความที่ต้องการให้แสดงเมื่อส่งข้อความ เช่น หากกรอกว่า ESP8266 เมื่อใช้ API ส่งข้อความว่า "สวัสดี" ข้อความจะขึ้นว่า "ESP8266:สวัสดี" ในช่องถัดมาจะให้เลือกว่าจะส่งข้อความเข้าไปในกลุ่มไหน หรือส่งให้ตัวผู้ใช้เท่านั้น (สกุล คำนวณชัย,2560)

งานวิจัยที่เกี่ยวข้องด้าน GPS และ เซนเซอร์

Senthilkumar, Brindha, & Bhandari. (2020) ในการพัฒนานวัตกรรมอย่างรวดเร็วทำให้ชีวิตของเรามีความต้องการน้อยลง นวัตกรรมได้ขยายความเสี่ยงของกิจกรรมเพิ่มขึ้นและอุบัติเหตุบนท้องถนนเกิดขึ้นบ่อยมากซึ่งทำให้มีผู้เสียชีวิตและทรัพย์สินเสียหายอย่างมากเนื่องจากการตอบสนองที่ไม่ดีจากผู้รับผิดชอบในการจัดการเหตุการณ์ดังกล่าว การดำเนินการรับรู้ความผิดพลาดจะเป็นทางออกที่ดีสำหรับปัญหานี้ บทความนี้ศึกษาเกี่ยวกับการใช้มาตรวัดความเร่งหรือเซนเซอร์เอียงเป็นส่วนหนึ่งของแอปพลิชันเตือนอัตโนมัติ โดยมีเป้าหมายเพื่อระบุการขับขี่ที่ไม่ปลอดภัย

สามารถใช้เป็นตัวค้นหาการชนหรือรถพลิกคว่ำของรถในระหว่างก่อนชนและหลังการชนด้วยสัญญาณจากเซนเซอร์ ซึ่งสามารถหลีกเลี่ยงหรือดูแลสถานการณ์ร้ายแรงอันเนื่องมาจากอุบัติเหตุได้โดยเร็วที่สุด เมื่อรถประสบอุบัติเหตุหรือรถเคลื่อนที่ เซนเซอร์การเอียงจะรับรู้และส่งไปยังไมโครคอนโทรลเลอร์ทันที ไมโครคอนโทรลเลอร์จะส่งข้อความเตือนผ่านโมดูล IoT รวมถึงตำแหน่งของอุบัติเหตุผ่านโมดูล GPS ไปยังตำรวจหรือกลุ่มควบคุม โดยเผยแพร่ผ่านระบบคลาวด์ ดังนั้นกลุ่มที่เปิดใช้งานวิกฤตสามารถติดตามพื้นที่ได้ทันทีผ่านโมดูล GPS ภายหลังการรับข้อมูล พื้นที่นี้ยังสามารถเห็นได้บน Google Maps ได้อีกด้วย

Majumdar, Bhargava, & Shirin. (2014) ความปลอดภัยเป็นปัญหาหลักสำหรับคนทุกหนทุกแห่งในปัจจุบัน ไม่มีใครคาดเดาอันตรายได้และตอบสนองต่ออันตรายใดๆ ได้ในเสี้ยววินาที ล่าสุดแอปมือถือได้รับการพัฒนา ซึ่งสามารถใช้เพื่อเชื่อมต่อกับผู้ติดต่อฉุกเฉินของคุณ แต่ไม่ใช่ทุกคนที่มีสมาร์ทโฟน นอกจากนี้ในการใช้แอปมือถือในเวลาอันสั้นไม่สามารถทำได้ บทความนี้ศึกษาเกี่ยวกับปุ่มกดที่จะใช้ในช่วงเวลาอุบัติเหตุฉุกเฉินที่จะรับรองความปลอดภัยในเวลาที่น้อยที่สุด โดยปุ่มกดจะต่อกับอุปกรณ์ที่ประกอบไปด้วย บอร์ด Arduino และ GSM Module ตัวบอร์ด Arduino จะส่งสัญญาณไปยังโมดูล GSM ด้วยซิมการ์ด ซึ่งจะส่งข้อความฉุกเฉินไปยังผู้ติดต่อที่กำหนดไว้ล่วงหน้าในระบบพร้อมกับพิกัด GPS ที่แน่นอนของตำแหน่งของผู้ใช้ ข้อมูลนี้สามารถใช้โดยผู้ติดต่อฉุกเฉินเพื่อระบุตำแหน่งของผู้ใช้อย่างแม่นยำและติดตามผู้ใช้ในเวลาอันสั้นที่สุด

Li. (2012) บทความนี้ได้ศึกษาเกี่ยวกับการออกแบบหน่วยที่ใช้กับ GPS ศึกษาวิธีการตั้งค่าของ GPS รุ่น LEA-4H และทักษะการแยกข้อมูล GPS Protocol วิจัยเทคโนโลยีแผนที่แบบกริดเพื่อทำเครื่องหมายซูม แปลแผนที่ และจับคู่เส้นทางรถ เพื่อตรวจสอบรถว่าอยู่ในสนามด้วยตำแหน่ง GPS เท่านั้น ซึ่งความถูกต้องและเสถียรภาพของผลการวิจัยได้รับการพิสูจน์โดยการทำงานที่เชื่อถือได้ งานวิจัยนี้เป็นข้อมูลอ้างอิงที่ดีสำหรับการจัดการระบบยานพาหนะอื่นๆอีกต่อไป

Pham, Drieberg, & Nguyen. (2013) ความสามารถในการติดตามยานพาหนะมีประโยชน์ในหลายๆ ทั้งในส่วนของแอปพลิเคชันรวมถึงการรักษาความปลอดภัยของยานพาหนะส่วนบุคคล ระบบขนส่งมวลชน การจัดการกองเรือและอื่นๆ นอกจากนี้ คาดว่าจำนวนรถบนท้องถนนทั่วโลกจะเพิ่มขึ้นอย่างรวดเร็วเช่นกัน ดังนั้น การพัฒนาระบบติดตามยานพาหนะโดยใช้ Global Positioning System (GPS) และ Global System for Mobile Communications (GSM) มีจุดมุ่งหมายเพื่อให้ผู้ใช้สามารถระบุตำแหน่งรถของตนได้อย่างง่ายดายและสะดวก ระบบจะให้ผู้ใช้มีความสามารถในการติดตามรถจากระยะไกลผ่านเครือข่ายมือถือ บทความนี้ได้ศึกษาเกี่ยวกับพัฒนาต้นแบบฮาร์ดแวร์ของระบบติดตามยานพาหนะ โดยเฉพาะระบบจะใช้ GPS เพื่อรับพิกัดของรถและส่งโดยใช้โมดูล GSM ไปยังโทรศัพท์ของผู้ใช้ผ่านเครือข่ายมือถือ ส่วนประกอบฮาร์ดแวร์หลักของระบบคือโมดูลตัวรับ GPS u-blox NEO-6Q โมดูล u-blox LEON-GIOO GSM และไมโครคอนโทรลเลอร์

Arduino Uno ระบบติดตามยานพาหนะที่พัฒนาขึ้นแสดงให้เห็นถึงความเป็นไปได้ในการติดตามยานพาหนะแบบเรียลไทม์และปรับแต่งได้ดีขึ้นต่อไปได้ในอนาคต

จากการทบทวนวรรณกรรมที่เกี่ยวข้อง ในการใช้ GPS ในการติดตามมีความถูกต้องและแม่นยำ โดยข้อมูลตำแหน่ง GPS จะถูกส่งข้อมูลแบบเรียลไทม์ ซึ่งได้นำไปใช้ในการติดตามยานพาหนะและใช้ในการแจ้งตำแหน่งช่วงเวลาอุบัติเหตุฉุกเฉิน โดยเป็นการส่งสัญญาณไปยังโมดูล GSM ด้วยซิมการ์ด ไปยังโทรศัพท์ของผู้ใช้ผ่านเครือข่ายมือถือ และการแจ้งเตือนจะอยู่ในรูปแบบข้อความ ด้วยเหตุนี้ผู้วิจัยจึงได้นำเอาตัว GPS มาใช้ในเรื่องของการติดตามตำแหน่งแจ้งเหตุฉุกเฉิน แต่ใช้เป็นตัว GPS UBLOX NEO-6M มาพัฒนาแทนตัว GSM เนื่องจากมีการพัฒนาในเรื่องของการเขียนโค้ดที่ง่ายกว่าและทำการส่งข้อมูลตำแหน่งไปยังฐานข้อมูลได้แบบเรียลไทม์ ในการส่งข้อมูลไปในฐานข้อมูลนั้น เพื่อบันทึกข้อมูลเป็นสถิติย้อนหลังได้ และในส่วนของการแจ้งเตือนจะทำการแจ้งเตือนผ่าน LINE Notify แทนการส่งข้อความ เนื่องจากการสื่อสารในยุคปัจจุบัน LINE เป็นที่นิยมใช้มากที่สุด เพราะมีการติดต่อสื่อสารได้ในหลากหลายรูปแบบ ทั้งข้อความ รูปภาพ เสียง และภาพเคลื่อนไหว เป็นต้น

2.2 การวิเคราะห์โครงข่ายถนน (Network Analysis)

2.2.1 ระบบสารสนเทศภูมิศาสตร์ (Geographic Information System : GIS)

ระบบภูมิสารสนเทศภูมิศาสตร์ หรือ GIS คือ ระบบสารสนเทศที่มีกระบวนการทำงานเกี่ยวกับข้อมูลในเชิงพื้นที่ โดยมีการใช้อุปกรณ์ทางด้านคอมพิวเตอร์ ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูลทางภูมิศาสตร์ และการออกแบบ ในการเสริมสร้างประสิทธิภาพในการจัดเก็บข้อมูล การปรับปรุงข้อมูล การคำนวณ และการวิเคราะห์ข้อมูลให้สามารถแสดงผลในรูปข้อมูลที่สามารถอธิบายสภาพต่างๆ บนพื้นผิวโลกหรืออ้างอิงทางภูมิศาสตร์ได้ เช่น การค้นหารายละเอียดของสถานที่ต่างๆ การวิเคราะห์ความเสียหายของสภาวะแวดล้อม เป็นต้น

องค์ประกอบของระบบสารสนเทศภูมิศาสตร์โดยหลักการแล้วจะประกอบด้วย 5 องค์ประกอบ

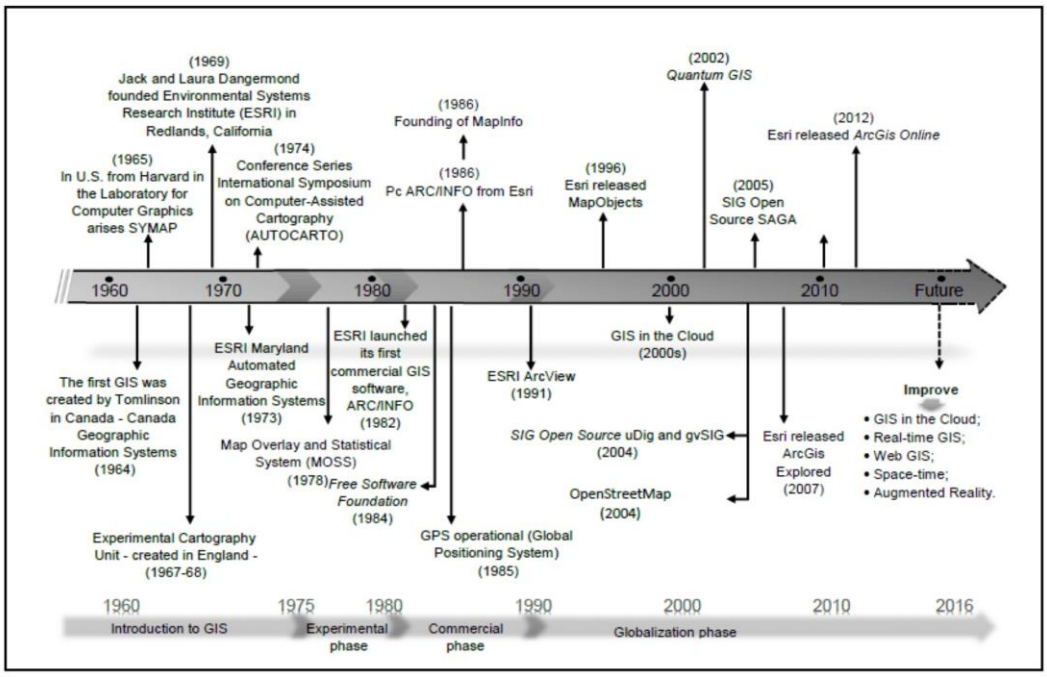
1. ฮาร์ดแวร์ (hardware) คือ เครื่องคอมพิวเตอร์ และ/หรือส่วนเชื่อมต่อ ทำหน้าที่ในการนำเข้าข้อมูล (data input) และแสดงผลลัพธ์ (data output) ที่ได้จากการวิเคราะห์ตามวัตถุประสงค์ที่กำหนดไว้

2. ซอฟต์แวร์ (software) คือ ชุดคำสั่งในเครื่องคอมพิวเตอร์ใช้ในการจัดการควบคุมการประมวลผล มี 2 ประเภท คือ (1) ซอฟต์แวร์ระบบปฏิบัติการ (operating software) มีหน้าที่ในการควบคุมการทำงานของระบบคอมพิวเตอร์ และ (2) ซอฟต์แวร์ประยุกต์ (application software) เป็นสิ่งที่พัฒนาขึ้นเพื่อการทำงานตามวัตถุประสงค์ มีความเฉพาะเจาะจงกับงานที่ต้องการ เช่น ซอฟต์แวร์ระบบสารสนเทศ และซอฟต์แวร์การวิเคราะห์ข้อมูลจากระยะไกล เป็นต้น

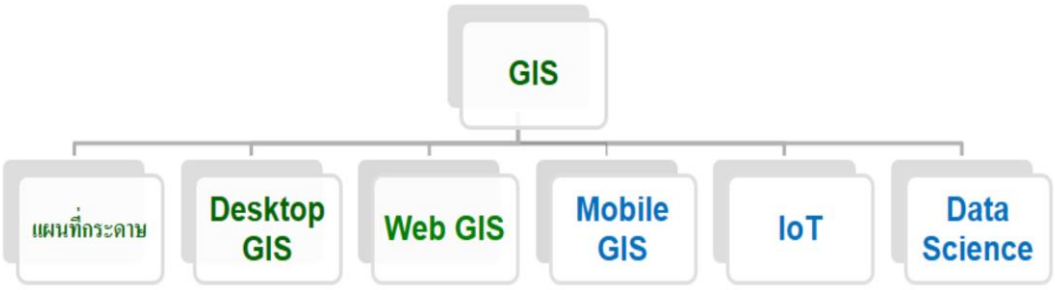
3. ข้อมูล เป็นส่วนสำคัญที่จะนำมาใช้ในระบบสารสนเทศภูมิศาสตร์ โดยการจัดเก็บจะต้องเป็นระบบตามวัตถุประสงค์ที่ต้องการ เพื่อความพร้อมและความถูกต้องในการวิเคราะห์ แหล่งข้อมูลทางภูมิศาสตร์มาจากหลากหลายแห่ง เช่น ภาพถ่ายดาวเทียม เอกสาร และภาพถ่าย เป็นต้น ซึ่งข้อมูลทางภูมิศาสตร์ ประกอบด้วย ลักษณะทางกายภาพของสิ่งที่ศึกษา ตำแหน่งหรือพิกัดทางภูมิศาสตร์ของสิ่งที่ศึกษา และเวลาที่เกิดขึ้นของสิ่งที่ศึกษา แบ่งประเภทข้อมูลออกเป็น 2 รูปแบบ คือ (1) ข้อมูลเชิงพื้นที่ (spatial data) เป็นข้อมูลที่สามารถอ้างอิงกับตำแหน่งทางภูมิศาสตร์ (geo-referenced) ซึ่งสามารถแสดงตำแหน่งของข้อมูลได้ว่าอยู่ที่ใดในระบบพิกัดทางภูมิศาสตร์ (geo-positioning) ข้อมูลเชิงพื้นที่ออกเป็น 2 แบบย่อย คือ แรสเตอร์ (raster) เป็นข้อมูลจากการกราดภาพ (scan) แผนที่ ภาพถ่ายดาวเทียม ภาพถ่ายทางอากาศ เมื่อขยายภาพจะเห็นเป็นโครงสร้างช่องสี่เหลี่ยม หรือเรียกว่า จุดภาพ หรือกริดเซลล์ (grid cell) เรียงต่อกันเป็นแนวราบและแนวดิ่ง สามารถนำมาใช้ในการจัดกลุ่มจำแนกวัตถุที่ปรากฏบนภาพ เช่น แนวถนนคอนกรีตกับถนนลูกรัง เป็นต้น และเวกเตอร์ (vector) สร้างจากการอ้างอิงแรสเตอร์ เป็นตัวแทนของสิ่งที่ปรากฏอยู่บนพื้นที่จริง แสดงผลในรูปแบบของจุด เส้น และพื้นที่ โดยจะมีมาตราส่วนเป็นตัวกำหนด เช่น 1: 1000 สามารถมองเห็นเป็นเส้นถนน มีทิศทาง ขนาดและจุดเริ่มต้น และ (2) ข้อมูลเชิงลักษณะ (attribute data หรือ non-spatial data) เป็นข้อมูลที่อธิบายคุณลักษณะต่าง ๆ (attribute) ของข้อมูลกับพื้นที่นั้น ๆ เช่น ข้อมูลประชากร การใช้ที่ดิน ข้อมูลชื่ออาคารสถานที่ ฯลฯ

4. กระบวนการ (Procedure) จัดเป็นส่วนสำคัญของระบบงานสารสนเทศภูมิศาสตร์ซึ่งในระบบสารสนเทศภูมิศาสตร์มีกระบวนการนำเข้าข้อมูล กระบวนการสร้างข้อมูล กระบวนการวิเคราะห์ข้อมูลโดยกระบวนการวิเคราะห์สามารถนำข้อมูลเชิงพื้นที่และข้อมูลเชิงลักษณะมารวมเข้าด้วยกัน ด้วยการซ้อนทับชั้นข้อมูล (data layer) เช่น การนำแผนที่ดินมาซ้อนทับกับแผนที่ธรณีวิทยาและชั้นความสูงที่แปลงค่าเป็นความลาดชัน เมื่อนำมารวมกัน ทำให้ทราบว่าดินบริเวณที่ศึกษานั้นอยู่บนชั้นหินอะไร สภาพความลาดชันและความคงทนของพื้นที่มีภูมิประเทศอย่างไร เป็นต้น

5. บุคลากร แบ่งเป็น 2 กลุ่มหลัก คือ (1) กลุ่มผู้สร้างข้อมูล เป็นผู้จัดทำ รวบรวม นำเข้า จัดเก็บ และแก้ไขข้อมูลให้ถูกต้องเป็นปัจจุบัน และ (2) กลุ่มผู้ใช้ข้อมูล ซึ่งทั้งสองกลุ่มต้องทำงานร่วมกันเพื่อให้ได้ข้อมูลแผนที่ (map data) ที่ถูกต้องตรงตามวัตถุประสงค์



ภาพ 5 วิวัฒนาการของระบบสารสนเทศภูมิศาสตร์



ภาพ 6 วิวัฒนาการของเทคโนโลยีสารสนเทศ

2.2.2 Database

Database หรือ ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลายๆ แฟ้มข้อมูล

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่างๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่างๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่างๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบ

จัดการฐานข้อมูล หรือ DBMS (Data Base Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

2.2.3 QGIS

QGIS คือโปรแกรมประเภทจัดการข้อมูล GIS (Geographic Information System) โปรแกรมหนึ่ง ซึ่งมีส่วนติดต่อผู้ใช้เป็นแบบกราฟิก (Graphic User Interface: GUI) ที่เข้าใจและใช้งานง่าย

QGIS ถูกพัฒนาขึ้นมาภายใต้สัญญาอนุญาตแบบเปิดเผยโค้ด (Open source) ซึ่งสามารถนำไปใช้งาน ได้อย่างไม่มีข้อจำกัด อีกทั้งยังสามารถนำโค้ดไปพัฒนาต่อได้อีกด้วย
การพัฒนาแบบเปิดเผยโค้ด คือ การเขียนโปรแกรมแบบเปิดเผยซอร์สโค้ด (Source code) ให้นักพัฒนาจากทั่วโลกได้ร่วมกันพัฒนาโปรแกรม ข้อดีคือการหล่อเลี้ยงโครงการพัฒนาลักษณะนี้มักมาจากเงิน ลงขันจากองค์กรใหญ่ ๆ ที่ต้องการใช้งานโปรแกรมนั้นแต่ไม่อยากซื้อของที่มีขายอยู่ในตลาดที่มีราคาแพงเกินไป ในขณะที่ต้องการใช้ความสามารถของโปรแกรมไม่มากนัก ดังนั้น QGIS จึงถูกพัฒนาขึ้นให้มีความสามารถหลากหลาย ทั้งการใช้งานทั่วไปอย่างการเรียกใช้งานข้อมูลภาพ ตาราง สืบค้นข้อมูล วิเคราะห์ข้อมูลแบบอ้างอิง ข้อมูลเชิงตำแหน่ง (Spatial query) ตลอดจนนำเสนอข้อมูลในรู้แบบแผนที่อย่างสวยงามทั้งแบบ Offline และ Online อีกด้วย

QGIS ถูกพัฒนาขึ้นโดยกลุ่มนักพัฒนาซอฟต์แวร์จากประเทศเยอรมันในปีพ.ศ. 2545 เริ่มที่เวอร์ชัน 0.001-alpha จนถึงปัจจุบันปี พ.ศ. 2564 (เดือนพฤศจิกายน) เวอร์ชันล่าสุดคือ QGIS 3.22.1 'Biatowieza' สามารถใช้ งานบนระบบปฏิบัติการ Windows , Linux หรือ Mac OS เป็นเครื่องมือที่มีประสิทธิภาพในการจัดการข้อมูล GIS และใช้ภาษา C++ เป็นหลักสามารถเชื่อมต่อและเรียกใช้ฟังก์ชันต่างๆกับ Geospatial RDBMS เช่น PostGIS/PostgreSQL และ GRASS ได้ อีกทั้งผู้ใช้สามารถพัฒนาปลั๊กอินขึ้นมาใช้เพิ่มความสามารถของโปรแกรมโดยใช้ภาษา Python ได้อีกด้วย

2.2.4 ภาษา SQL

Structured Query Language ภาษาสอบถามข้อมูลหรือภาษาจัดการข้อมูลอย่างมีโครงสร้าง มีการพัฒนาภาษาคอมพิวเตอร์และโปรแกรมฐานข้อมูลที่รองรับมากมาย เพราะจัดการข้อมูลได้ง่าย เช่น MySQL, MsSQL, PostgreSQL หรือ MS Access เป็นต้น สำหรับโปรแกรมฐานข้อมูลที่ได้รับความนิยมคือ MySQL เป็น Open Source ที่ใช้งานได้ทั้งใน Linux และ Windows ใช้ในการจัดการฐานข้อมูล จัดเก็บข้อมูล เพิ่ม ลบ แก้ไข และค้นหา รวมทั้งการจัดเรียงข้อมูล

ภาษา SQL เป็นภาษาที่ใช้ในการเขียนโปรแกรมสำหรับการจัดเก็บข้อมูล ซึ่งสามารถใช้ได้กับระบบฐานข้อมูลใดก็ได้ โดยไม่ยึดติดกับฐานข้อมูลใดเป็นหลัก นอกจากนี้ภาษา SQL ยังเป็นภาษาที่เข้าใจได้ง่ายและไม่ซับซ้อนและยังสามารถทำงานที่ซับซ้อนได้เพียงคำสั่งได้เพียงคำสั่งไม่กี่คำสั่ง ภาษา SQL จึงเป็นภาษาที่นิยมใช้กันมากสำหรับฐานข้อมูลของเว็บไซต์

ประโยชน์ของภาษา SQL

- มีการจัดการฐานข้อมูล
- สร้างตารางและข้อมูล
- สามารถเรียกใช้งานและค้นหาข้อมูล
- ไม่ยึดติดกับฐานข้อมูลใดข้อมูลหนึ่ง

2.2.5 PostgreSQL

PostgreSQL คือ ระบบจัดการฐานข้อมูลเชิงวัตถุ สัมพันธ์ เป็นโปรแกรม OpenSource ที่สามารถนำไปใช้งานได้โดยไม่มีค่าใช้จ่ายใดๆ เป็นจัดการฐานข้อมูลแบบ object-relational database management System หรือ (ORDBMS) ซึ่งมีต้นแบบระบบฐานข้อมูล POSTGRES 4.2 ของมหาวิทยาลัยแคลิฟอร์เนีย วิทยาลัยเขตเบอร์กเลย์ (UC Berkeley) ตั้งแต่ปี ค.ศ. 1977 จัดเป็น Open Source Software ที่มีประวัติยาวนานมากที่สุดตัวหนึ่งโพสต์เกรสคิวเอล (PostgreSQL) หรือนิยมเรียกว่า โพสต์เกรส (Postgres) เป็นระบบจัดการ ฐานข้อมูลในลักษณะของซอฟต์แวร์เสรีภายใต้สัญญาอนุญาตบีเอสดี ชื่อเดิมของซอฟต์แวร์คือ โพสต์เกรส ซึ่งต่อมาได้ถูกเปลี่ยนเป็นโพสต์เกรสคิวเอล โดยประกาศออกมาจากทีมหลักในปี 2550 ชื่อของ โพสต์เกรสมาจากชื่อ post-Ingres ซึ่งหมายถึงตัวซอฟต์แวร์ที่พัฒนาต่อจากซอฟต์แวร์ชื่ออินเกรส ทำหน้าที่เป็นตัวกลางสื่อสารข้อมูลส่งภาษาให้ฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซับซ้อนและความสัมพันธ์ระหว่างข้อมูลต่างๆภายในฐานข้อมูล

ส่วนประกอบของระบบ PostgreSQL องค์ประกอบพื้นฐานของระบบ PostgreSQL server ซึ่งแสดงในด้านซ้ายของ pgAdmin III มี 5 อย่าง คือ

1. Tablesspace เป็นที่ตั้งทางกายภาพของ Objects
2. Databases เป็นออบเจกต์หลักของฐานข้อมูลใน PostgreSQL ซึ่งเก็บข้อมูลทั้งหมดที่จะใช้ในระบบ เมื่อผู้ใช้เชื่อมต่อกับ Database server จะเป็นการเชื่อมต่อกับ Database objects และเข้าถึงออบเจกต์ทั้งหมด ในฐานข้อมูล ฐานข้อมูลแต่ละตัวจะประกอบด้วย objects 4 ชนิด คือ Casts, Language, Replications, และ Schemas
3. Schemas (แสดงต่อจากฐานข้อมูลแต่ละตัว) เป็น object ที่สำคัญที่สุดในฐานข้อมูล ซึ่ง schemas จะเก็บ object อื่น ๆ อีกหลายชนิด เพื่อเก็บข้อมูลในฐานข้อมูล

4. Gruop Roles ใช้เพื่อกำหนดสิทธิการเข้าถึงแบบกลุ่มของ user โดยมันจะทำหน้าที่ควบคุมการเข้าถึงในระดับ server
5. Login Roles หรือ user account คือ ผู้ใช้ฐานข้อมูลโดยผู้ดูแลฐานข้อมูลจะเป็นผู้สร้างให้กับแต่ละคน

PostgreSQL ได้รับความนิยมมากเนื่องจากชุดคุณลักษณะ ฐานข้อมูลช่วยในการพัฒนาแอปพลิเคชัน โดยการรักษาความสมบูรณ์ของข้อมูล ช่วยให้ผู้ดูแลระบบสามารถสร้างสภาพแวดล้อมที่ทนต่อความผิดพลาดได้ นอกจากนี้ยังสามารถใช้ข้ามแพลตฟอร์มที่หลากหลายและใช้ประโยชน์จากภาษาโปรแกรมทั่วไปทั้งหมด เราจะเห็นรายชื่อที่แน่นอนในภายหลัง

ฐานข้อมูลยังมีระบบล็อคขั้นสูงมาก นอกจากนี้ยังมีการควบคุมการทำงานพร้อมกันหลายเวอร์ชัน เซิร์ฟเวอร์ฐานข้อมูล PostgreSQL ยังมีฟังก์ชันการทำงานสำหรับการเขียนโปรแกรมฝั่งเซิร์ฟเวอร์สำหรับผู้ใหญ่อีกด้วย เป็นไปตามข้อกำหนด ANSI SQL และรองรับสถาปัตยกรรมเครือข่ายไคลเอ็นต์-เซิร์ฟเวอร์อย่างสมบูรณ์

PostgreSQL ยังมีความพร้อมใช้งานสูงและเซิร์ฟเวอร์สำรอง สอดคล้องกับ ANSI-SQL2008 และเชิงวัตถุ ความสามารถในการเชื่อมต่อกับคลังข้อมูลอื่นๆ เช่น NoSQL ซึ่งทำหน้าที่เป็นฮับแบบครบวงจรสำหรับระบบหลายภาษา สามารถทำได้ผ่านการสนับสนุน JSON ของฐานข้อมูล ข้อมูลของคลัสเตอร์ฐานข้อมูลเดียวจะได้รับการจัดการโดยอินสแตนซ์ PostgreSQL หนึ่งอินสแตนซ์เสมอ คลัสเตอร์ของฐานข้อมูลคือกลุ่มของเร็กคอร์ดที่เก็บไว้ในที่เดียวกันบนระบบไฟล์

นอกจากนี้ PostgreSQL ยังทำงานในหลายแพลทฟอร์มได้แก่ Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), และ Windows

ลักษณะสำคัญอีกอย่างหนึ่งคือ PostgreSQL เป็นซอฟต์แวร์แบบรหัสเปิดใช้ลิขสิทธิ์ BSD ซึ่งหมายถึงผู้ใช้สามารถนำไปใช้งานได้ฟรี นอกจากนี้ในปัจจุบัน PostgreSQL ไม่อยู่ภายใต้การควบคุมขององค์กรใดโดยเฉพาะแต่มีผู้ร่วมพัฒนาจากทั่วโลก ทำให้ PostgreSQL มีการปรับปรุงอย่างต่อเนื่อง (http://admin.wikidot.com/pgsqlintro)

ดังนั้น PostgreSQL เป็นระบบจัดการฐานข้อมูลอีกหนึ่งทางเลือกสำหรับในการทำงานการศึกษา เพื่อช่วยในการจัดการฐานข้อมูลต่าง ๆให้เป็นไปตามแผนการดำเนินการที่วางไว้ โปรแกรม PostgreSQL เป็นที่นิยมอย่างมากเพราะสามารถใช้ได้โดยไม่ต้องเสียค่าใช้จ่าย และยังมีอัปเดทให้ทันสมัยอยู่เสมอ

2.2.6 PostGIS

PostGIS คือ ส่วนขยายเพิ่มเติมที่ทำให้ฐานข้อมูล PostgreSQL สามารถรองรับข้อมูลด้านสารสนเทศภูมิ- ศาสตร์ (GIS) คือสนับสนุนข้อมูลที่สัมพันธ์เชิงพื้นที่ (Spatial) มีการเพิ่มเติมในส่วนฐานข้อมูลเชิงวัตถุสัมพันธ์ (object - reletionnal database system) ของ PostgreSQL ให้มีการรองรับระบบสารสนเทศภูมิศาสตร์ (GIS Object) เข้ามาเก็บไว้ในฐานข้อมูล (Database) PostGIS สนับสนุน GIST indexs กับ R-tree indexs และฟังก์ชัน เพื่อใช้เป็นพื้นฐานในการวิเคราะห์ GIS Object (สิทธิชัย ชูสำโรง,2559)

2.2.7 pgRouting

pgRouting Library เป็นเครื่องมือในการวิเคราะห์คำนวณหาเส้นทางจากตำแหน่งจุดเกิดเหตุฉุกเฉินไปยังตำแหน่งของรถโรงพยาบาลและรถกู้ชีพพร้อมกับตำแหน่งโรงพยาบาลหากเกิดกรณีเหตุฉุกเฉินขึ้น pgRouting Library เป็นเครื่องมือที่ใช้ในการวิเคราะห์หาเส้นทางที่สั้นที่สุด ตามระยะทางเส้นทางสัญจรปกติดังนั้น pgRouting เป็นเครื่องมือที่ทำงานร่วมกับฐานข้อมูลเชิงพื้นที่ PostgreSQL/PostGIS โดยเพิ่มฟังก์ชั่นการคำนวณหาระยะทาง (Network Analysis) และการวิเคราะห์โครงข่ายอื่นๆ(pgRouting Contributors, 2013) ซึ่ง pgRouting ได้พัฒนามาจาก pgDijkstra เขียนโดย Sylvain Pasche จาก camptocamp ต่อมาได้มีการนำไปพัฒนาต่อโดยบริษัท Orkney ประเทศญี่ปุ่น และเปลี่ยนชื่อใหม่เป็น pgRouting อย่างเป็นทางการ (Kastl and Junod, 2011) โดยวัตถุประสงค์หลักของ pgRouting คือ จัดหาฟังก์ชั่นสำหรับการใช้งานใน PostgreSQL/PostGIS เพื่อสร้างเครื่องมือในการคำนวณหาระยะทาง ซึ่งจะคล้ายๆ กับชุดคำสั่งในโปรแกรมบางโปรแกรม เช่น คำสั่งการค้นหาระยะทางที่ใกล้ที่สุดในโปรแกรม ArcGIS Desktop หรือการขอเส้นทางใน Google Maps ไม่เฉพาะในเรื่องของระยะทางบนถนนเท่านั้น แต่สามารถใช้ได้กับข้อมูลอะไรก็ได้ที่เกี่ยวกับระยะทาง การสิ้นเปลืองเวลา น้ำมัน เงิน เช่น เส้นทางเกี่ยวกับการเดินเรือและระบบเน็ตเวิร์กแม่ข่ายบนอินเตอร์เน็ต เป็นต้น นอกจาก pgRouting สามารถคำนวณหาระยะทางที่สั้นที่สุดและเร็วที่สุดแล้ว pgRouting ยังสามารถช่วยในการวางแผนการเดินทางในการจัดส่งสินค้าหลายๆ ที่ในการเดินทางครั้งเดียวกัน เช่น จะไปส่งของให้ลูกค้าทั้งหมด 4 ที่ โดยเริ่มต้นเดินทางจากโรงงานผู้ผลิต ควรจะไปส่งของให้ลูกค้ารายใดก่อนหลังตามลำดับ เพื่อช่วยในการประหยัดเวลาและน้ำมัน ซึ่งฟังก์ชันของ pgRouting มีทั้งหมดดังนี้

Topology Functions

pgr_createTopology - to create a topology based on geometry.

pgr_createVerticesTable - to reconstruct the vertices table based on the source and target information.

pgr_analyzeGraph - to analyze the edges and vertices of the edge table.

pgr_analyzeOneway - to analyze directionality of the edges.

pgr_nodeNetwork -to create nodes to a not noded edge table.

Routing Functions

All pairs - All pair of vertices.

pgr_floydWarshall - Floyd-Warshall's Algorithm

pgr_johnson- Johnson's Algorithm

pgr_astar - Shortest Path A*

pgr_bdAstar - Bi-directional A* Shortest Path

pgr_bdDijkstra - Bi-directional Dijkstra Shortest Path

dijkstra - Dijkstra family functions

pgr_dijkstra - Dijkstra's shortest path algorithm.

pgr_dijkstraCost - Use pgr_dijkstra to calculate the costs of the shortest paths.

Driving Distance - Driving Distance

pgr_drivingDistance - Driving Distance

Post processing

pgr_alphaShape - Alpha shape computation

pgr_pointsAsPolygon - Polygon around set of points

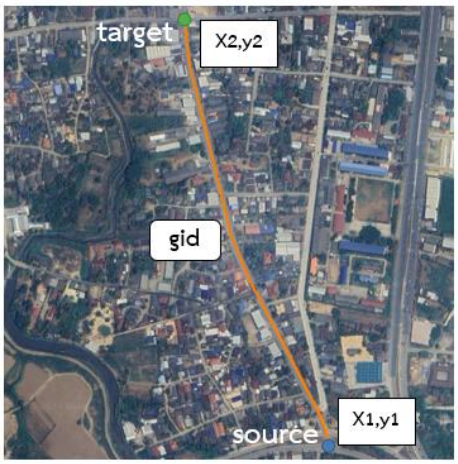
pgr_ksp - K-Shortest Path

pgr_trsp - Turn Restriction Shortest Path (TRSP)

pgr_tsp - Traveling Sales Person

โดยจะมีรูปแบบโครงสร้างของการคำนวณเส้นทางด้วย pgRouting ที่ประกอบไปด้วยคอลัมน์เหล่านี้

- ID (gid)
- Source (จุดเริ่มต้น)
- Target (จุดสิ้นสุดหรือปลายทาง)
- Length as Cost (ค่าระยะทาง ความเร็ว ประเภทถนน ฯลฯ)
- Reverse_cost (สำหรับการคำนวณในเรื่องของการใช้เส้นทางเดินรถเพียงทางเดียว) (One way)



ภาพ 7 รูปแบบโครงสร้างของการคำนวณด้วย pgRouting

ข้อดีของวิธีการหาเส้นทางจากระบบฐานข้อมูล คือ

1. สามารถทำการแก้ไขข้อมูลและรายละเอียดข้อมูลได้หลายวิธี เช่น QGIS และ uDig ผ่านทาง JDBC, ODBC หรือแก้ไขโดยตรงจากโปรแกรม P1/pgSQL อีกทั้งยังสามารถแก้ไขผ่านทางคอมพิวเตอร์หรือโทรศัพท์ก็ได้
2. ข้อมูลที่ถูกแก้ไขจะสามารถแสดงผลลัพธ์ใหม่ได้ทันที่ผ่านโปรแกรมการเส้นทาง โดยไม่ต้องทำการคำนวณใหม่อีกครั้งและตัวแปลค่าน้ำหนัก “cost” สามารถคำนวณผ่าน SQL และค่าเหล่านั้นสามารถเรียกมาคำนวณได้จากหลายๆ ตาราง หลายๆ คอลัมน์ในเวลาเดียวกัน

ปัจจุบัน pgRouting มีฟังก์ชั่นต่างๆ เพิ่มขึ้นมามากมาย ดังที่แสดงอยู่ข้างล่าง

- ฟังก์ชัน One to One เส้นทางจาก Node หนึ่งไปยังอีก Node หนึ่ง
- ฟังก์ชัน One to Many เส้นทางจาก Node หนึ่งไปยังอีกหลาย Node
- ฟังก์ชัน Many to One เส้นทางจากหลาย Node ไปยังอีก Node หนึ่ง
- ฟังก์ชัน Many to Many เส้นทางจากหลาย Node ไปยังอีกหลาย Node หนึ่ง

(Choosumrong et al., 2012)

2.2.8 อัลกอริทึมของไดก์สตรา

อัลกอริทึมของไดก์สตรา (Dijkstra's algorithm) ที่เป็นหลักการคำนวณทางคณิตศาสตร์ของ Dijkstra (E.W. Dijkstra, 1959) ซึ่งเป็นการนำเอาทฤษฎีกราฟมาใช้ในการคำนวนหาเส้นทางที่เร็วที่สุดหรือเหมาะสมที่สุดของรถพยาบาลและรถกู้ภัยฉุกเฉินไปยังจุดเกิดเหตุฉุกเฉินและขณะเดียวกันก็คำนวณหาเส้นทางจากจุดเกิดเหตุไปยังโรงพยาบาล โดยใช้เวอร์เทกซ์ (Vertex) และเส้น (Edge) แทนถนนที่เชื่อมต่อกัน กำหนดระยะทางระหว่างจุดเป็นตัวเลขลงไปในกราฟ โดยเรียกกราฟดังกล่าวว่ากราฟถ่วงน้ำหนัก (Weighted Graph) คือกราฟที่เส้นเชื่อมทุกเส้นมีค่าน้ำหนักที่มีค่าเป็นจำนวนจริงที่

ไม่ติดลบ (พีระวัฒน์ และสุเพชร จิรขจรกุล, 2557) สำหรับการคำนวณหาเส้นทางที่สั้นที่สุดจากจุดเริ่มต้น (start_vid) ไปยังจุดสิ้นสุด (end_vid) ซึ่งสามารถคำนวณเส้นทางได้ทั้งแบบ directed graph และ undirected graph (pgRouting Contributors, 2013) ดังสูตร

จากสูตร $$G = (V,E) \quad (1)$$

เมื่อ

V(G) คือ เซตของเวอร์เทกซ์ (Vertex) ในกราฟ

E(G) คือ เซตของเส้น (Edge) ในกราฟ

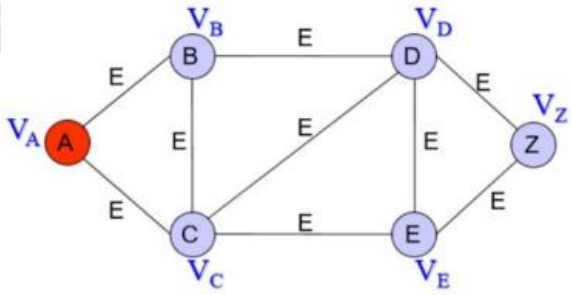
$$V(x) = \min\{V(x), W(x) + d(y)\} \quad (2)$$

- d(y) = dist(v,y) | min{distance from node start to node y}
- W(x) = weight (i,j) | Distance from i to j : edge (i,j)

โดยมีตัวอย่างการคำนวณเส้นทางจากจุด A ไปยังจุด Z ด้วยอัลกอริทึมของไดก์สตรา (Dijkstra's algorithm) ซึ่งคิดคำนวณจากสมการ **V(x) = min{V(x), W(x) + d(y)}**

เมื่อ d(y) คือ ระยะทางของ Node ที่นำมาคำนวณ โดยจะเลือก Node ที่น้อยที่สุดของค่าที่นำมาคำนวณในครั้งนั้น และ W(x) คือ ค่าระยะทางจาก Node i ไปยัง Node j

การคำนวณครั้งที่หนึ่ง โปรแกรมจะทำการคำนวณจากจุดเริ่มต้น (Node i) ไปยัง (Node j) ที่เชื่อมต่อกับ Node i โดยจะกำหนดให้จุดเริ่มต้นเท่ากับ 0 ในครั้งแรก และกำหนดให้ V(x) เท่ากับระยะทาง (E) เริ่มต้น ดังภาพ โดยจากภาพจุดที่เชื่อมต่อกับจุด A คือ จุด B และ C ดังนั้นค่าของ V(c) และ V(b) จึงเท่ากับ W(a,c) และ W(a,b) จากการคำนวณในครั้งแรก โปรแกรมจะเลือกจุด C มาเก็บไว้ ซึ่งมีระยะทางหรือค่า W น้อยที่สุด



ภาพ 8 การทำงานของกราฟ Network

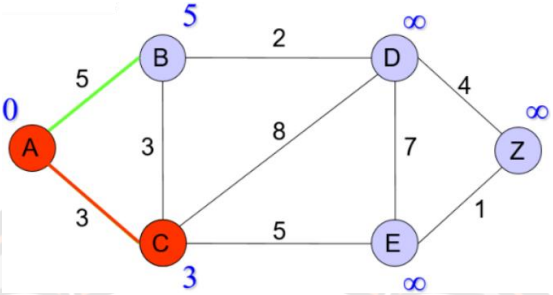
จากค่าในสมการ เราจะคำนวณเปรียบเทียบค่าระหว่าง V(b) และ V(c) ซึ่งสองค่านี้เชื่อมต่อกับ V(a) ที่เป็นจุดเริ่มต้นในการคำนวณครั้งนี้ โดยจะกำหนดให้ d(y) = 0 เนื่องจากค่าเริ่มต้น = 0 และกำหนดค่า V(x) ให้เป็น infinity จากนั้นแทนค่า W(x) ด้วยระยะทาง + ด้วยค่า d(y) ดังตัวอย่าง

จากนั้นนำผลลัพธ์ไปใส่ที่ค่า V(x) ที่จุด B และ C ซึ่งผลที่ได้มีค่าน้อยกว่าคือ 3 โปรแกรมก็จะเก็บค่า 3 ไว้ที่ Node หรือ V(c) = 3

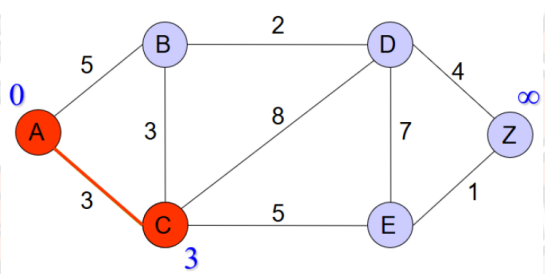
จากสูตร V(x) = min{V(x), W(x) + d(y)}

V(B) = min {∞,5+0} = 5

V(C) = min {∞,3+0} = 3



ภาพ 9 การคำนวณหาเส้นทาง ครั้งที่ 1



ภาพ 10 ผลการคำนวณหาเส้นทาง ครั้งที่ 1

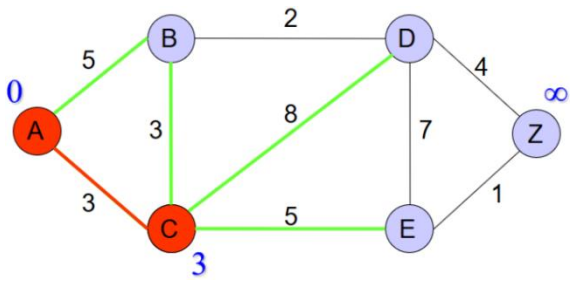
ในการคำนวณครั้งต่อไป ก็จะทำเช่นเดิมคือ คิดระยะทางจากจุดเริ่มต้นไปยังจุดที่มีการเชื่อมต่อ ตอนนี้จุดเริ่มต้นของการคำนวณจะมี 2 จุด คือ V(A) และ V(C) ซึ่งในตอนนี้ค่า d(y) ที่จุด A และ C จะมีค่าต่างกัน คือ d(y) A = 0 และ d(y) = 3 เนื่องจาก หากเริ่มต้นที่จุด C จะต้องบวกเพิ่มไปอีก 3 ซึ่งก็คือระยะจากจุดเริ่มต้นจริงๆ ไปยังจุด C จากนั้นแทนค่าในสมการดังภาพ

V(D) = min {∞,8+3} = 11

V(AB) = min {∞,5+0} = 5

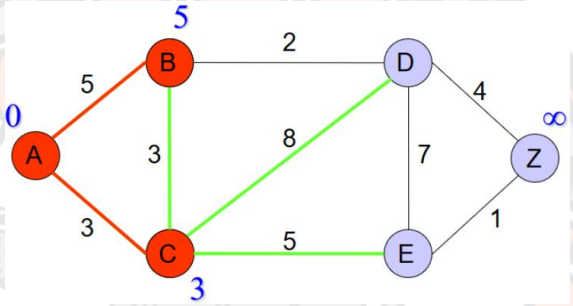
V(CB) = min {∞,3+3} = 6

V(E) = min {∞,5+3} = 8



ภาพ 11 การคำนวณหาเส้นทาง ครั้งที่ 2

ซึ่งในการคำนวณรอบนี้ จุดที่มีค่าน้อยที่สุดคือ จุด V(AB) โปรแกรมจะทำการเก็บค่า V(AB) ไปไว้ที่ Node B ดังภาพ



ภาพ 12 ผลการคำนวณหาเส้นทาง ครั้งที่ 2

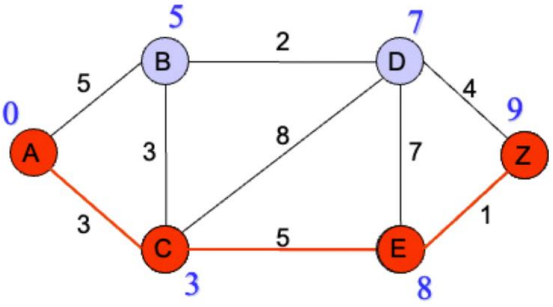
ในการคำนวณครั้งต่อไป จะเริ่มที่จุด V(B) และ V(C) จะไม่นำจุด A มาคิด เนื่องจากจุด A ได้แทนค่าเข้าไปใน V(AB) = 5 และ V(AC) = 3 เรียบร้อยแล้ว ซึ่งผลลัพธ์จากการคำนวณกราฟนี้จะได้คำตอบคือ ต้องเดินทางจากจุด A ผ่านไปยังจุด C,E และถึง Z ตามลำดับดังภาพ

V(Z) = min {∞,4+7} = 11

V(Z) = min {∞,1+8} = 9

Shortest path from A to Z = 9;

G (A - Z) = {(A,C,E,Z),(3,5,1)}

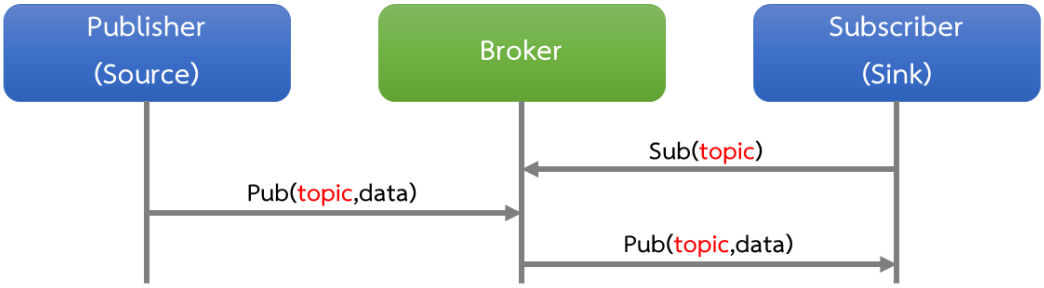


ภาพ 13 ผลการคำนวณหาเส้นทางจาก A-Z

2.2.9 โปรโตคอล MQTT

Message Queuing Telemetry Transport (ADMIN, 2559: Web Site) เป็น Protocol ที่ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (machine-to-machine) คือ อุปกรณ์กับอุปกรณ์ สนับสนุนเทคโนโลยี IoT (Internet of Things) คือ เทคโนโลยีที่อินเทอร์เน็ตเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ โทรทัศน์ ตู้เย็น เข้ากับอินเทอร์เน็ตทำให้สามารถเชื่อมโยงสื่อสารกับอุปกรณ์ต่าง ๆ ได้ โดยผ่านเครือข่ายอินเทอร์เน็ต ซึ่งจะทำให้มนุษย์สามารถควบคุมอุปกรณ์ต่าง ๆ จากที่อื่นได้ เช่น การสั่งปิด-เปิดไฟในบ้านจากที่อื่นๆ

เนื่องจากโปรโตคอลตัวนี้มีน้ำหนักเบา ออกแบบมาเพื่อใช้งานกับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก การรับส่งข้อมูลในเครือข่ายที่มีขนาดเล็ก แบนด์วิชท์ต่ำ ใช้หลักการแบบ publisher/subscriber คล้ายกับหลักการที่ใช้ใน Web Service ที่ต้องใช้ Web Server เป็นตัวกลางระหว่างคอมพิวเตอร์ของผู้ใช้ แต่ MQTT จะใช้เป็นตัวกลางที่เรียกว่า Broker เพื่อทำหน้าที่จัดการคิว รับ-ส่ง ข้อมูลระหว่างอุปกรณ์และทั้งในส่วนที่เป็น Publisher และ Subscriber ดังภาพ



ภาพ 14 แสดงหลักการทำงานของโปรโตคอล MQTT

จากภาพจะเห็นได้ว่า Topic จะเป็นตัวอ้างอิงหลัก ข้อมูลที่จะ Publisher ออกไปยัง Broker จะต้องมี topic กำกับไว้เสมอ ทางฝ่าย subscriber ก็จะอ้างถึง topic เพื่อเรียกข้อมูลที่ต้องการ เหมือนกับการสมัครเป็นสมาชิกของหนังสือพิมพ์ฉบับหนึ่ง ชื่อของหนังสือก็เปรียบเหมือน topic และผู้ผลิตก็คือ publisher เมื่อถึงเวลาที่หนังสือเสร็จ ผู้ส่ง Broker ก็จะนำหนังสือพิมพ์มาส่งให้เรา ตัวอย่าง แอปพลิเคชั่นที่ใช้งานคือ Facebook Mesenger

สรุปองค์ประกอบของ MQTT Protocol จะประกอบไปด้วย Broker , Publisher และ Subscriber แต่ละอย่างก็จะทำหน้าที่แตกต่างกันออกไปโดย Broker ทำหน้าที่เป็นตัวกลางคอยจัดการกับข้อควา โดยอ้างอิงจาก Topic Publisher จะทำหน้าที่คอยส่งข้อมูลไปยังหัวข้อนั้น ๆ Subscriber จะทำหน้าที่คอยดูการเปลี่ยนแปลงของ message ที่อ้างอิงด้วย Topic เช่น ถ้ามีหัวข้อหน้าสนใจและมีการเปลี่ยนแปลงก็จะทำการดึงข้อมูลนั้น ๆ มาใช้งาน โดย MQTT Protocol จะประกอบไปด้วย Broker , Publisher และ Subscriber ซึ่งมีหน้าที่ ดังนี้

1. Broker จะทำหน้าที่เป็นตัวกลางคอยจัดการ message โดยอ้างอิงด้วย topic ซึ่งโปรแกรมที่ใช้ทำหน้าที่เป็น MQTT Broker มีหลากหลายให้ใช้งาน เช่น โปรแกรม Mosquitto เป็น OpenSource MQTT Broker สนับสนุน MQTT Broker

2. Subscriber จะทำหน้าที่คอยดูการเปลี่ยนแปลงของ message ที่อ้างอิงด้วย topic เช่น หากหัวข้อที่สนใจมีการเปลี่ยนแปลงก็จะดึง data มาใช้งาน

3. Publisher จะทำหน้าที่ส่งข้อมูลไปยัง Topic นั้น ๆ

2.2.10 Node-RED

Node-RED เป็นเครื่องมือสำหรับนักพัฒนาโปรแกรมในการเชื่อมต่ออุปกรณ์ฮาร์ดแวร์เข้ากับ APIs (Application Programming Interface) และเป็นการเขียนโปรแกรมแบบกำหนดขั้นตอนที่มีส่วนควบคุมแบบ Block สำหรับนักพัฒนาผ่านเว็บเบราเซอร์ (Web Browser) ซึ่ง Node-RED เป็นเครื่องมือที่ช่วยให้เราทำงานแบบ Flow-Based Programming เป็นอุปกรณ์ที่ช่วยให้นักพัฒนาโปรแกรมสามารถทำงานให้เครื่องมือ IoT (Internet of Things) ทำงานได้ตามที่เราต้องการได้ง่าย โดยทำผ่านเว็บเบราวเซอร์

เนื่องจาก Node-RED เป็น Flow-Based Programming ทำให้ไม่ต้องเขียน Code ในการพัฒนาโปรแกรม แค่เพียงเลือก Node มาวางแล้วเชื่อมต่อก็สามารถควบคุม I/O ได้ โดย Node-RED จะมี Node ให้เลือกใช้งานอย่างหลากหลาย สามารถสร้างฟังก์ชัน JavaScript ได้โดยใช้ Text Editor ที่มีอยู่ใน Node-RED และยังสามารถบันทึก Function, Templates, Flows เพื่อไปใช้งานกับงานอื่นได้

งานวิจัยที่เกี่ยวข้องด้าน Network Analysis

Choosumrong, Raghavan, Delucchi, Yoshida, & Vinayaraj. (2014) ศึกษาเกี่ยวกับการค้นหาเส้นทางที่สั้นที่สุด เวลาที่น้อยที่สุด และมีความปลอดภัย โดยใช้ pgRouting ในการค้นหาระยะทาง เพื่อนำทางระหว่างจุดเริ่มต้น และจุดปลายทาง บางครั้งเส้นทางอาจมีการเปลี่ยนแปลงหรือไม่สามารถใช้งานได้ชั่วคราว และเส้นทางที่สนับสนุนแบบไดนามิก จะแสดงให้เห็นถึงระบบที่ใช้ FOSS4G และสามารถปรับแต่งสำหรับความหลากหลายตามความต้องการ การพัฒนาระบบจะใช้โปรแกรม Web Map Server, Apache, PHP, JavaScript, PostgreSQL, PostGIS, QGIS เป็นต้น นอกจากนี้ยังสามารถใช้ระบบยังรวมกับ Android SDK ที่สามารถเข้าถึงระบบผ่านเครือข่าย เพื่อเปิดใช้งานอุปกรณ์พกพา การศึกษาจะมุ้งเน้นไปที่ปัญหาอื่นๆที่เกี่ยวกับข้อมูลการสร้างแบบจำลองสำหรับนำทางยานพาหนะ pgRouting สามารถใช้เว็บเป็นการบริการประมวลผล Wi-Fi Protected Setup (WPS) โดยใช้เป็นที่รองรับระบบ ZOO

Mali, Rao, & Mantha. (2012) ศึกษาเกี่ยวกับวิธีการเข้าถึงพื้นที่เกิดอุบัติเหตุที่ใช้เวลาน้อยที่สุด เพื่อหาเส้นทางเส้นทางที่เหมาะสมในเวลาที่จำเป็น โดยทำการศึกษาการปรับปรุงเส้นทาง ซึ่งใช้ขั้นตอนวิธีการหาเส้นทางที่สั้นที่สุดของ Dijkstra's นำมาใช้ในโครงข่ายถนน เพื่อให้ได้เส้นทางที่เหมาะสมระหว่างจุดเริ่มต้น และจุดสิ้นสุด และกระบวนการลำดับชั้นเชิงวิเคราะห์ในการเปรียบเทียบและวิเคราะห์ปัจจัยโดยรวมของถนนแต่ละสายในเครือข่ายอินเตอร์เน็ต เส้นทางที่เหมาะสมนี้ไม่ได้หมายความว่าเป็นเส้นทางที่สั้นที่สุด แต่อาจหมายความว่าเป็นเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุดก็เป็นได้

Choosumrong, Raghavan, & Realini. (2010) ศึกษาเกี่ยวกับวิธีการหาเส้นทางที่สั้นที่สุด เวลาที่น้อยที่สุด และเส้นทางที่ปลอดภัย โดยใช้วิธี pgRouting ในการพัฒนาเส้นทางโดยปรับเปลี่ยนการทำงานของ pgRouting ขั้นตอนของ pgRouting จะปรับปรุงวิธีการของ Dijkstra มาใช้ และจะใช้เส้นทางแบบไดนามิกในการปรับเปลี่ยนข้อมูลตามเงื่อนไขของถนน สำหรับการออกแบบพัฒนาระบบ และโปรแกรมทางภูมิศาสตร์จะใช้ PostGIS และ pgRouting เป็นส่วนขยายสำหรับ PostgreSQL, DBMS ระบบที่พัฒนาขึ้นจะเป็นประโยชน์ในการจัดการการจราจรในภาวะฉุกเฉินและปฏิบัติการช่วยเหลือ

นิธิศักดิ์ โพธาราม วรุตม์ ศรีมณีธรรม และยุพิน สรรพคุณ (2558) ศึกษาเกี่ยวกับระบบสนับสนุนการตัดสินใจในการเดินทาง ซึ่งจะช่วยอำนวยความสะดวกให้กับผู้รับบริการ ในการคำนวณค่าสึกหรอจะมีการนำเส้นทางที่เดินทางมาคำนวณกับอุปกรณ์ของยานพาหนะที่ผู้บริการเลือกใช้ในการเดินทาง โดยใช้ทฤษฎี Dijkstra's Algorithm และทฤษฎี Branch and Bound Algorithm เพื่อเปรียบเทียบการทำงานของทั้ง 2 แบบ ว่าแบบใดมีประสิทธิภาพดีกว่ากัน เพื่อนำมาประยุกต์ใช้ในการหาเส้นทางที่สั้นที่สุดหรือหาเวลาที่น้อยที่สุด ในด้านของการทำงานของเว็บแอปพลิเคชันสนับสนุนการตัดสินใจการเลือกเส้นทาง ผู้ใช้จะต้องล็อคอินเข้าสู่ระบบเพื่อใช้งาน ผู้ใช้งานจะต้องเป็นสมาชิกของระบบนี้ด้วยจึงจะสามารถเริ่มใช้งานได้ สมาชิกจะทำการเลือกเส้นทางที่ต้องการเดินทางพร้อมกับป้อนรายละเอียดของยานพาหนะ เพื่อที่จะนำไปประมวลผลค่าสึกหรอของยานพาหนะต่อการเดินทางในแต่ละครั้ง จากการศึกษาจะพบว่า Dijkstra's Algorithm สามารถทำงานได้ตรงตามความต้องการมากกว่า และระบบสนับสนุนการตัดสินใจในการเดินทางจะช่วยอำนวยความสะดวกให้กับผู้ใช้ในการเดินทาง และการช่วยตัดสินใจก่อนการเดินทาง เพื่อประหยัดเวลา และค่าใช้จ่ายได้

จากการทบทวนวรรณกรรมที่เกี่ยวข้อง ผู้วิจัยเห็นว่าการนำ pgRouting ไปใช้ในการพัฒนาส่วนใหญ่เป็นการพัฒนาศึกษาเกี่ยวกับวิธีการเข้าถึงพื้นที่เกิดอุบัติเหตุ การจัดการการจราจรในภาวะฉุกเฉิน และการตัดสินใจในการเลือกใช้เส้นทางในการเดินทาง โดยการพัฒนาได้นำ pgRouting Library มาช่วยในเรื่องของการคำนวณระยะทางตามเส้นทางสัญจรจริง และอีกทั้งยังสามารถสร้างเงื่อนไขในการค้นหาเส้นทางได้หลากหลายวิธีขึ้นอยู่กับความเหมาะสมของแต่ละงาน ด้วยเหตุนี้ผู้วิจัย

จึงได้นำเทคนิค pgRouting Library มาช่วยในเรื่องของการคำนวณหาเส้นทางของรถโรงพยาบาล รถกู้ชีพและโรงพยาบาลที่ใกล้ที่สุดกับจุดแจ้งเหตุฉุกเฉิน โดยระบบนี้จะนำไปใช้กับผู้สูงอายุและผู้พิการ เพื่อช่วยพิจารณาในเรื่องของระยะเวลาในการเดินทางเข้าถึงตัวผู้ป่วยให้เร็วที่สุดและปลอดภัยที่สุด

2.3 ซอฟต์แวร์ที่ใช้การพัฒนา Web Map Application

2.3.1 OpenStreetMap

OpenStreetMap (OSM) เป็นโครงการความร่วมมือในสร้างแผนที่ฟรี ที่สามารถแก้ไขแผนที่โลกได้โดยมีสโลแกน The Free Wiki World Map หรือ วิกิพีเดียแผนที่โลก สร้างโดย Steve Coast ประเทศอังกฤษ ปี ค.ศ. 2004 โดยทีแรงบันดาลใจจากความสำเร็จของวิกิพีเดียต่อมาเดือนเมษายน 2006 ได้ก่อตั้งมูลนิธิ OpenStreetMap เพื่อส่งเสริมการเติบโตของ OpenStreetMap ต่อมาวันที่ 12 กันยายน ค.ศ. 2012 ได้ใช้สัญญาอนุญาต Open Database License (ODbL) ซึ่งเดิมใช้ ครีเอทีฟคอมมอนส์ CC BY-SA นักพัฒนาสามารถนำข้อมูลแผนที่และ API ไปใช้ได้ฟรี แต่ต้องให้ credit OpenStreetMap และทำตามสัญญาอนุญาต ทั้งนี้ผู้พัฒนา Freeware ด้าน GIS อย่าง QGIS ก็เขียนโปรแกรมให้สามารถนำ OpenStreetMap มาใช้ได้เช่นกัน นอกจากนี้ยังมีหลายบริษัทในวงการ IT ก็ใช้ OpenStreetMap เช่น BingMaps ของค่าย Microsoft, Apple Maps, Wikipedia, Foursquare เป็นต้น

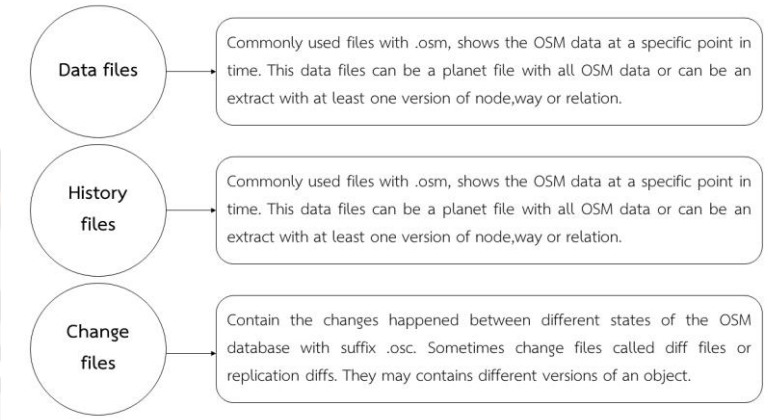
OpenStreetMap (OSM) เป็นคอลเลคชั่นขนาดใหญ่ของที่เก็บข้อมูลทางภูมิศาสตร์ที่เป็นอาสาสมัครในไฟล์ประเภทต่างๆ โดยใช้รูปแบบการเข้ารหัสที่แตกต่างกันเพื่อแปลงข้อมูลนี้เป็นบิตและไบต์ OSM เป็นความพยายามร่วมกันในการสร้างแผนที่โลกที่แก้ไขได้ฟรี ผลลัพธ์หลักของความร่วมมือนี้คือข้อมูลทางภูมิศาสตร์มากกว่าตัวแผนที่ ข้อจำกัดในการใช้หรือความพร้อมใช้งานของข้อมูลทางภูมิศาสตร์ทั่วโลกทำให้จำเป็นต้องสร้าง OSM ข้อมูลที่มีอยู่จาก OSM พร้อมที่จะแทนที่ Google Maps สำหรับแอปพลิเคชันแบบคลาสสิก (Facebook, Craigslist ฯลฯ) และข้อมูลเริ่มต้นสำหรับแอปพลิเคชันตัวรับสัญญาณ GPS แม้ว่าคุณภาพของข้อมูลจะมีความหลากหลายทั่วโลก แต่ข้อมูล OpenStreetMap สามารถนำมาเปรียบเทียบกับสิทธิบัตรได้อย่างสะดวก แหล่งข้อมูล ซึ่ง OpenStreetMap เป็นฐานข้อมูลของสารสนเทศแผนที่ ซึ่งสารสนเทศดังกล่าวสามารถนำไปใช้ได้สำหรับหลากหลายวัตถุประสงค์ และมีข้อจำกัดน้อยมาก สามารถใช้ OpenStreetMap ผ่านหน้าแผนที่หลักของ OpenStreetMap หรือใช้ผ่านบริการอื่นๆ ที่ใช้ข้อมูลของ OpenStreetMap ซึ่งดำเนินงานโดยหน่วยงานต่างๆ โดยคุณค่าของ OpenStreetMap มาจากการบริการที่แตกต่าง ซึ่งแสดงข้อมูลที่ไม่เหมือนกันโดยผลออกมาเป็นแผนที่ด้วยวิธีที่ต่างกันไป

OpenStreetMap เป็นโปรเจกต์การทำงานร่วมกันเป็นแบบเปิด เพื่อสร้างแผนที่โลกที่สามารถแก้ไขได้อย่างอิสระ OpenStreetMap ถูกจัดทำขึ้นโดยชุมชนนักทำแผนที่ ซึ่งช่วยกันดูแล

จัดการข้อมูล เกี่ยวกับถนน เส้นทาง ร้านอาหาร ทางรถไฟ สถานี และอื่นๆ จากทั่วโลก ดาวน์โหลดแผนที่ฟรีที่จัดทำขึ้นนี้ได้จากบนไซต์ OpenStreetMap: www.OpenStreetMap.org.

รูปแบบไฟล์ OpenStreetMap

รูปแบบไฟล์ OpenStreetMap มีหลายวิธีและรูปแบบไฟล์ในการจัดเก็บข้อมูลทางภูมิศาสตร์ แต่รูปแบบไฟล์ OSM ถูกจำกัดไว้ที่ OpenStreetMap OSM ได้รับการออกแบบมาโดยเฉพาะในรูปแบบมาตรฐานเพื่อให้ง่ายต่อการขนส่งผ่านอินเทอร์เน็ต รูปแบบคำสั่งที่มีโครงสร้างซึ่งเข้ารหัสใน XML ประกอบด้วยไฟล์ .osm ใน OpenStreetMap มีสามองค์ประกอบหลักในการจัดเก็บโครงสร้างข้อมูลโทโพโลยี:



ภาพ 15 รูปแบบไฟล์ OpenStreetMap

ที่มา: https://docs.fileformat.com/th/gis/osm/

2.3.2 GeoServer

เป็นซอฟต์แวร์รหัสเปิดที่ทำหน้าที่เป็นแม่ข่ายแผนที่ (Map Server) สำหรับการแลกเปลี่ยนและการแก้ไขข้อมูลภูมิสารสนเทศผ่านเครือข่าย ซอฟต์แวร์นี้ได้รับการพัฒนาด้วยภาษาจาวา (Java) และรองรับมาตรฐานเปิด (Open Standard) ที่หลากหลาย เช่น OGC 1.1.1, OGC 1.0 และ OGC WCS 1.0 เป็นต้น จุดเด่นของ GeoServer ได้แก่ ความสามารถในการแสดงผลข้อมูลได้ทั้งบน Google Maps, Google Earth, Yahoo Maps และ Microsoft Virtual Earth นอกจากนี้ยังสามารถเชื่อมต่อกับซอฟต์แวร์ ESRI ArcGIS ซึ่งเป็นแบบซอฟต์แวร์แบบมีลิขสิทธิ์ที่มีการใช้งานอย่างแพร่หลายได้อีกด้วย ความสามารถของซอฟต์แวร์นี้ยังดึงดูดผู้ใช้งานอีกหลายประการ คือ การกำหนดโครงแบบ (Configuration) ของการให้บริการ (Service) ผ่านส่วนติดต่อผู้ใช้งาน (GUI) ทำให้ผู้ใช้งานที่ไม่มีทักษะทางด้านการเขียนโปรแกรมก็สามารถสร้างการให้บริการแผนที่ได้ สถานะปัจจุบันของโครงการนี้ยังอยู่ในระยะเริ่มต้น (Incubation) ภายใต้การดูแลของ OSGeo (สารสนเทศภูมิศาสตร์,

2556) ในการเรียกใช้และปรับแต่ง Map Service โดยใช้ Geoserver สามารถทำงานผ่านทาง Web Browser ในการให้บริการของ WMS และ WFS มีรายละเอียดดังต่อไปนี้

1. การให้บริการแผนที่ออนไลน์ในรูปแบบ WMS (Web Map Service)

Web Map Service (WMS) เป็นข้อกำหนดมาตรฐานที่ใช้ในการผลิตแผนที่จากข้อมูลภูมิสารสนเทศเพื่อบริการผู้ใช้ โดยผู้ใช้สามารถเรียกข้อมูลแผนที่จากหลายๆ แหล่งมาซ้อนทับกันได้ นอกจากนี้ยังรวมไปถึงการบริการเรียกข้อมูลเชิงบรรยายที่มีความสัมพันธ์เชิงตำแหน่งกับข้อมูลภูมิสารสนเทศแม่ข่าย Web Map Service (WMS) จะทำการสร้างภาพบิตแมป (Bitmap) จากข้อมูลภูมิสารสนเทศ และทำการส่งภาพบิตแมปมายังผู้ใช้ที่อยู่ทางฝั่งลูกข่าย โดยรูปแบบของภาพบิตแมปได้แก่ PNG, GIF, JPEG จะประกอบด้วย 3 Operation ได้แก่ GetCapabilities, GetMap, GetFeatureInfo โดยมีรายละเอียดดังนี้

GetCapabilities คือการร้องขอข้อมูลรายละเอียด คุณสมบัติ และความสามารถของแม่ข่ายเพื่อขอใช้ บริการแผนที่ ในรูปแบบ XML เช่น Name (ชื่อชั้นข้อมูล), BoundingBox (ขอบเขตข้อมูล), Style (การแสดงผล), SRS (ระบบพิกัด) และอื่นๆ เป็นต้น มีโครงสร้างของเอกสารการร้องข้อดังนี้

ตาราง 2 ตารางอธิบายพารามิเตอร์การร้องขอ WMS GetCapabilities

ลำดับ	พารามิเตอร์	คำอธิบาย
1	bbox	ค่าพิกัดสองจุดประกอบเป็นรูปสี่เหลี่ยมผืนผ้า คือ พิกัดล่างซ้าย (x,y) และ พิกัดบนขวา (x,y) ตามลำดับ ทั้งนี้ต้องสัมพันธ์กับระบบพิกัดที่ระบุในตัวแปร srs ด้วย
2	styles	เลือก style สำหรับแสดงผลที่ WMS เตรียมไว้ให้(ถ้าไม่ระบุจะใช้ default ที่ผู้ใช้บริการกำหนดไว้)
3	format	format ของภาพแผนที่ ที่จะส่งกลับมาให้ผู้ขอใช้บริการ
4	request	ชื่อของ operation ดังเช่น GetMap, GetCapabilities เป็นต้น
5	layers	ชื่อของชั้นข้อมูลแบบเรียงลำดับโดยมีเครื่องหมาย “,” ขั้น
6	srs	ระบบพิกัด EPSG:4326 (ระบบพิกัดภูมิศาสตร์)

GetMap คือการร้องขอภาพแผนที่ ผู้ใช้สามารถระบุชื่อชั้นข้อมูล ขอบเขตพื้นที่ ขนาดจุดภาพ และรูปแบบแรสเตอร์ แม่ข่ายจะตอบสนองด้วยภาพแผนที่ที่ Web browser ทั่วไปอ่านได้ เช่น GIF, PNG, JPEG , TIFF และอื่นๆ เป็นต้น มีโครงสร้างของเอกสารการร้องข้อดังนี้

ตาราง 3 ตารางอธิบายพารามิเตอร์การร้องขอ WMS Get GetMap

ลำดับ	พารามิเตอร์	คำอธิบาย
1	bbox	ค่าพิกัดสองจุดประกอบเป็นรูปสี่เหลี่ยมผืนผ้า คือ พิกัดล่างซ้าย (x,y) และ พิกัดบนขวา (x,y) ตามลำดับ ทั้งนี้ต้องสัมพันธ์กับระบบพิกัดที่ระบุในตัวแปร srs ด้วย
2	styles	เลือก style สำหรับแสดงผลที่ WMS เตรียมไว้ให้(ถ้าไม่ระบุจะใช้ default ที่ผู้ใช้บริการกำหนดไว้)
3	format	format ของภาพแผนที่ ที่จะส่งกลับมาให้ผู้ขอใช้บริการ
4	request	ชื่อของ operation ดังเช่น GetMap, GetCapabilities เป็นต้น
5	layers	ชื่อของชั้นข้อมูลแบบเรียงลำดับโดยมีเครื่องหมาย “,” ขั้น
6	width	ความกว้างของภาพแผนที่ (หน่วย: pixel)
7	height	ความสูงของภาพแผนที่ (หน่วย: pixel)
8	srs	ระบบพิกัด EPSG:4326 (ระบบพิกัดภูมิศาสตร์)

GetFeatureInfo คือการร้องขอระบบให้บริการแผนที่ เพื่อทำการค้นคืนข้อมูลแผนที่ด้วยเงื่อนไขทาง ตำแหน่ง ซึ่งสามารถขอผ่าน GetFeatureInfo ได้ ผู้ใช้สามารถระบุพิกัด x,y ของจุดอ้างอิงสัมพัทธ์ทั่วไปเทียบกับภาพแผนที่ แม่ข่ายจะตอบกลับด้วยข้อมูล Attribute ในรูปแบบ HTML , GML หรือรูปแบบ ASCII อย่างง่าย โดยมีโครงสร้างของเอกสารการร้องข้อดังนี้

ตาราง 4 ตารางอธิบายพารามิเตอร์การร้องขอ WMS GetFeatureInfo

ลำดับ	พารามิเตอร์	คำอธิบาย
1	bbox	ค่าพิกัดสองจุดประกอบเป็นรูปสี่เหลี่ยมผืนผ้า คือ พิกัดล่างซ้าย (x,y) และ พิกัดบนขวา (x,y) ตามลำดับ ทั้งนี้ต้องสัมพันธ์กับระบบพิกัดที่ระบุในตัวแปร srs ด้วย
2	styles	เลือก style สำหรับแสดงผลที่ WMS เตรียมไว้ให้(ถ้าไม่ระบุจะใช้ default ที่ผู้ใช้บริการกำหนดไว้)
3	format	format ของภาพแผนที่ ที่จะส่งกลับมาให้ผู้ขอใช้บริการ
4	info_format	รูปแบบของผลลัพธ์ (default เป็น text/plain)
5	request	ชื่อของ operation ดังเช่น GetMap, GetCapabilities เป็นต้น
6	layers	ชื่อของชั้นข้อมูลแบบเรียงลำดับโดยมีเครื่องหมาย “,” ขั้น
7	query_layers	ชื่อชั้นข้อมูลที่จะสืบค้น

ลำดับ	พารามิเตอร์	คำอธิบาย
8	width	ความกว้างของภาพแผนที่ (หน่วย: pixel)
9	height	ความสูงของภาพแผนที่ (หน่วย: pixel)
10	x	ตำแหน่งของพิกัดที่จะสืบค้นในแนวแกน x (หน่วย: pixel)
11	y	ตำแหน่งของพิกัดที่จะสืบค้นในแนวแกน y (หน่วย: pixel)

2. การให้บริการแผนที่ออนไลน์ในรูปแบบ WFS (Web Feature Service)

Web Feature Service (WFS) เป็นข้อกำหนดมาตรฐานที่ใช้ในการเข้าถึงข้อมูลภูมิสารสนเทศชนิดเวกเตอร์ (จุด, เส้น, พื้นที่) จากผู้ให้บริการข้อมูล โดยผู้ใช้สามารถเข้าถึงข้อมูลด้วยการดาวน์โหลดข้อมูลภูมิสารสนเทศในรูปแบบเอกสาร XML ผ่านระบบอินเตอร์เน็ต โดยการทำงานของ Web Feature Service สามารถแบ่งออกเป็น 2 ประเภท ได้แก่ Basic WFS และ Transaction WFS (WFS-T) โดยที่ Basic WFS จะเป็นการใช้งานติดต่อผ่าน http อย่างง่าย เพื่อตรวจสอบคุณสมบัติ ความสามารถ ชั้นข้อมูล รูปแบบการตอบกลับที่ทำได้ มักจะเรียกใช้เป็นคำสั่งแรกเพื่อนำไปสร้างเป็น user interface และ WFS-T เป็นการเรียกใช้ชนิดเป็นระเบียน จะช่วยให้การสร้าง ลบ และการปรับปรุงคุณสมบัติ โดย Basic WFS ประกอบด้วย 3 Operation ได้แก่ GetCapabilities , .DescribeFeatureType , GetFeature

GetCapabilities คือ การตรวจสอบคุณสมบัติและความสามารถของแม่ข่ายเพื่อขอใช้บริการแผนที่ รูปแบบการตอบกลับที่สามารถทำงานได้ มักจะเรียกใช้เป็นคำสั่งแรกเพื่อสร้างเป็น User Interface ในกรณีที่ ผู้ใช้บริการต้องการทราบข้อมูลเกี่ยวกับคุณสมบัติของ WFS ที่เปิดให้บริการ ดังเช่น มีข้อมูลอะไรบ้าง มีโอเปอร์เรชั่น ที่ให้เรียกใช้งานอะไรบ้างนั้น ผู้ใช้บริการจะต้องเรียกข้อมูลเหล่านี้ผ่านโอเปอร์เรชั่นWFS:GetCapabilities ซึ่งสิ่งที่ WFS ส่งกลับมาให้กับผู้ขอใช้บริการนั้น คือ ไฟล์ข้อความที่มีโครงสร้างเป็น XML ส่วนใหญ่มักจะเรียกว่า "เอกสาร WFS:GetCapabilities" ตามชื่อโอเปอร์เรชั่น หรือ เรียกสั้น ๆ ว่า "เอกสาร Capabilities"

ตาราง 5 ตารางอธิบายพารามิเตอร์การร้องขอ WMS GetCapabilities

ลำดับ	พารามิเตอร์	คำอธิบาย
1	ows:ServiceIndentification	ข้อมูลเกี่ยวกับตัวบริการเอง ดังเช่น ชื่อ, เวอร์ชั่น, คำสำคัญ(keyword) เป็นต้น
2	ows:ServiceProvider	ข้อมูลเกี่ยวกับผู้ให้บริการ ดังเช่น ชื่อหน่วยงาน ,เบอร์โทรศัพท์, ที่อยู่ เป็นต้น
3	ows:OperationsMetadata	ข้อมูลเกี่ยวกับโอเปอร์เรชั่น (ฟังก์ชั่น) ที่ให้บริการ ดังเช่นGetCapabilities,GetFeature,

ลำดับ	พารามิเตอร์	คำอธิบาย
		DescribeFeatureType เป็นต้น
4	wfs:FeatureTypeList	ข้อมูลเกี่ยวกับ feature type และโอเปอร์เรชั่นของแต่ feature ที่ให้บริการ ดังเช่น ชื่อ, คำอธิบาย, ระบบพิกัด, กรอบข้อมูล(Bounding box) เป็นต้น
5	wfs:ServesGMLObjectType	แสดงข้อมูล GML Object ที่สนับสนุนและไม่ได้ร่างมาจากgml:AbstractFeatureType ที่ใช้ร่วมกับโอเปอร์เรชั่นGetGMLObject
6	ogc:Filter_Capabilities	แสดงข้อมูลเกี่ยวกับโอเปอร์เรชั่นหรือฟังก์ชั่นที่สามารถใช้ในการสืบค้นหรือกรองข้อมูลที่สนับสนุนใน WFS นี้

DescribeFeatureType คือ การบรรยายโครงสร้างของ feature ที่จะให้บริการ เช่น รายละเอียดของ ข้อมูล Attribute ที่ประกอบอยู่ด้วยกัน เมื่อทราบว่า WFS ที่จะขอใช้บริการมีข้อมูลชื่ออะไรอยู่บ้างแล้ว จากนั้นต้องการทราบว่าข้อมูลดังกล่าวมีโครงสร้างอย่างไรนั้น ผู้ใช้บริการสามารถร้องขอข้อมูลเหล่านี้ได้ผ่านโอเปอร์เรชั่น WFS:DescribeFeatureType ซึ่งเอกสารที่ตอบกลับมาจะอธิบายโครงสร้างของข้อมูล ดังเช่น ชื่อ, ชนิด เป็นต้น และเพื่อให้การพัฒนา Application สามารถเลือกหรือไม่เลือก ข้อมูลเชิงอรรถาธิบาย รวมทั้งเตรียมวิธีการแสดงผลที่เหมาะสมได้

GetFeature คือ การเรียกดูข้อมูลหรือการเข้าถึงข้อมูล feature อาจมีเงื่อนไขการกรองของการเลือกผ่าน FilterEncoding เช่น จังหวัดที่ขึ้นต้นด้วย “จังหวัด*” ผลลัพธ์ของข้อมูลจะอยู่ ในรูปแบบ GML ผู้ใช้บริการ WFS สามารถขอข้อมูลเหล่านี้ได้ด้วยโอเปอร์เรชั่น WFS:GetFeature ซึ่งบริการ WFS จะส่งข้อมูลกลับมาให้คำร้องขอข้อมูล WFS:GetFeature

2.3.3 HTML

HTML ย่อมาจาก HyperText Markup Language เป็นภาษาคอมพิวเตอร์ที่ใช้สร้างหน้าเว็บ (WebPage) ในรูปแบบของ ไฟล์ HTML (คือไฟล์ที่มีนามสกุลเป็น.htm หรือ .html) ซึ่งมีเว็บเบราว์เซอร์ (WebBrowser) เป็นโปรแกรมที่ใช้แปลงไฟล์ HTML เพื่อแสดงผลในรูปของหน้าเว็บ

ไฟล์ HTML เป็นไฟล์รหัสแอสกี (ASCII) ถูกบันทึกในรูปของไฟล์เอกสาร (Text File) ที่สามารถถูกสร้างจากโปรแกรมสร้างไฟล์ข้อความ (Text Editor) เช่น Notepad หรือ Word Processing ทั่วๆ ไป ซึ่งลักษณะของไฟล์ HTML ประกอบไปด้วยแท็ก (Tag) ต่างๆ ที่เป็นคำ สั่งของ HTML ซึ่งแท็กจะอยู่ภายในเครื่องหมาย < และ >

แท็กใน HTML แบ่งเป็น 2 ประเภท คือ คอนเทนเนอร์แท็ก (Container Tag) และแท็กเปล่า (Empty Tag) โดยที่คอนเทนเนอร์แท็ก ประกอบไปด้วยแท็กเปิดและแท็กปิด โดยที่แท็กปิดจะมีเครื่องหมาย/ นำหน้าแท็ก เช่น <H1>.....</H1> ส่วนแท็กเปล่า จะมีแท็กเปิดอย่างเดียว เช่น <HR> ซึ่งแท็กจะถูกเขียนด้วยตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กก็ได้ จะไม่มีผลต่อการแสดงผลของเว็บเบราว์เซอร์ เช่น <BR>,
,
 หรือ
 เว็บเบราว์เซอร์จะ แปลความหมายเหมือนกัน

โครงสร้างไฟล์ HTML แบ่งออกเป็น 2 ส่วน คือ ส่วนหัวเรื่อง (Head Section) และส่วนเนื้อหา (Body Section) โดยจะมีแท็ก <HTML> และ </HTML> เป็นตัวกำหนดขอบเขตไฟล์ ซึ่งส่วนหัวเรื่องมีไว้กำหนดข้อมูลเฉพาะของหน้าเว็บ เช่น ชื่อเรื่อง ของเว็บภายในแท็ก <HEAD> และ </HEAD> และสำหรับส่วนเนื้อหามีไว้กำหนดรายละเอียดต่างๆ ที่ต้องการแสดงบนหน้าเว็บ เช่น ข้อความ และรูปภาพภายในแท็ก <BODY> และ </BODY>

```
<HTML>
    <HEAD>
            <TITLE> ใส่ชื่อเรื่อง </TITLE>
    </HEAD>
    <BODY>
            ใส่เนื้อหาที่ต้องการแสดงบนหน้าเว็บ
    </BODY>
</HTML>
```

ภาพ 16 โครงสร้างไฟล์ HTML

ที่มา : https://www.bu.ac.th/knowledgecenter/executive_journal/july_sep_11/pdf/aw32.pdf

2.3.4 PHP

ในช่วงแรกภาษาที่นิยมใช้ในการทำงานบนระบบ web คือ HTML (Hypertext Markup Language) แต่ภาษา HTML เป็น Static Language ก็คือ ภาษาที่ใช้สร้างข้อมูลประเภทตัวอักษร ภาพ หรือ Object อื่นๆ ที่ไม่สามารถเปลี่ยนแปลงได้ด้วยตัวของมันเองหรือเรียกง่ายๆ ว่าข้อมูลที่คงที่นั่นเอง ก็ทำให้ไม่ยืดหยุ่นต่อมาความต้องการของมนุษย์ไม่มีวันเพียงพอ จึงได้มีการพัฒนาภาษาที่เป็น Dynamic Language ขึ้นมาก็คือ ภาษาที่มีข้อมูลถูกเปลี่ยนแปลง Auto ตามเงื่อนไขต่างๆ ที่ผู้เขียนกำหนดไว้มีการประกาศตัวแปรได้ข้อมูลที่เปลี่ยนแปลงก็เปลี่ยนแปลงตามตัวแปร ซึ่งนี่เป็นที่มาของภาษา PHP, CGI และ ASP เป็นต้น โดยเฉพาะภาษาประเภท Scripts ที่ สามารถติดต่อกับผู้ใช้ได้และหนึ่งในภาษาเหล่านั้นก็คือ PHP ซึ่งเป็นภาษาที่ได้รับความนิยมอย่างมากในปัจจุบัน ถูกสร้างขึ้น เมื่อปี 1994 ก่อน Windows 95 ปีเดียว โดยนาย Rasmus Lerdorf

PHP เป็นภาษาจำพวก scripting language คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริป เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่ง ที่ช่วยให้เราสามารถ สร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

ความสามารถของภาษา PHP

1. เป็นภาษาที่มีลักษณะเป็นแบบ Open source ผู้ใช้สามารถ Download และนำ Source code ของPHP ไปใช้ได้โดยไม่เสียค่าใช้จ่าย

2. เป็นสคริปต์แบบ Server Side Script ดังนั้นจึงทำงานบนเว็บเซิร์ฟเวอร์ ไม่ส่งผลกับการทำงานของเครื่อง Client โดยPHP จะอ่านโค้ดและทำงานที่เซิร์ฟเวอร์ จากนั้นจึงส่งผลลัพธ์ที่ได้จากการประมวลผลมาที่เครื่องของผู้ใช้ในรูปแบบของ HTML ซึ่งโค้ดของ PHP นี้ผู้ใช้จะไม่สามารถมองเห็นได้

3. PHP สามารถทำงานได้ในระบบปฏิบัติการที่ต่างชนิดกัน เช่น Unix, Windows, Mac OS อย่างมีประสิทธิภาพ เนื่องจาก PHP เป็นสคริปต์ที่ต้องทำงานบนเซิร์ฟเวอร์ ดังนั้น คอมพิวเตอร์สำหรับเรียกใช้คำสั่ง PHP จึงจำเป็นต้องติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์ไว้ด้วย เพื่อให้สามารถประมวลผล PHP ได้

4. PHP สามารถทำงานได้ในเว็บเซิร์ฟเวอร์หลายชนิด เช่น Personal Web Server (PWS), Apache, OmniHttpd และ Internet Information Service (IIS) เป็นต้น

5. ภาษา PHP สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)

6. PHP มีความสามารถในการทำงานร่วมกับระบบจัดการฐานข้อมูลที่หลากหลาย ซึ่งระบบจัดการฐานข้อมูลที่สนับสนุนการทำงานของ PHP เช่น Oracle, MySQL, FilePro, Solid, FrontBase, mSQL, Microsoft Accessและ MS SQL เป็นต้น

7. PHP อนุญาตให้ผู้ใช้สร้างเว็บไซต์ซึ่งทำงานผ่านโปรโตคอลชนิดต่างๆ ได้ เช่นLDAP, IMAP, SNMP, POP3 และ HTTP เป็นต้น

8. โค้ด PHP สามารถเขียนและอ่านในรูปแบบของ XML ได้

2.3.4 ภาษา JavaScript

JavaScript เป็นภาษาโปรแกรม (programming language) ประเภทหนึ่งที่เรียกกันว่า "สคริปต์" (script) ซึ่งมีวิธีการทำงานในลักษณะแปลความและดำเนินงานไปทีละคำสั่ง (interpret)

ภาษานี้เดิมมีชื่อว่า LiveScript ได้รับการพัฒนาขึ้นโดย Netscape ด้วยวัตถุประสงค์ เพื่อที่จะช่วยให้เว็บเพจสามารถแสดงเนื้อหาที่มีการเปลี่ยนแปลงไปได้ตามเงื่อนไขหรือสภาพแวดล้อมต่าง ๆ กันหรือสามารถโต้ตอบกับผู้ชมได้มากขึ้น ทั้งนี้เพราะภาษา HTML แต่เดิมนั้น เหมาะสำหรับใช้แสดงเอกสารที่มีเนื้อหาคงที่แน่นอนและไม่มีลูกเล่นอะไรมากมาย

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการและมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิดที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA ซึ่งเราจะพบว่าปัจจุบันจะหาเว็บเพจที่ไม่ใช้ JavaScript เลยนั้นได้ยาก

การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยเบราว์เซอร์ ดังนั้น JavaScript จึงสามารถทำงานได้เฉพาะบนเบราว์เซอร์ที่สนับสนุน ซึ่งปัจจุบันเบราว์เซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ดีสิ่งที่ต้องระวัง คือ JavaScript มีการพัฒนาเป็นเวอร์ชั่นใหม่ ๆ ออกมาด้วย (ปัจจุบันคือรุ่น 4.3) ดังนั้น ถ้านำโค้ดของเวอร์ชั่นใหม่ไปรันบนเบราว์เซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด error ได้

การทำงานของ JavaScript เกิดขึ้นบนเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไรหรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ต่างกับภาษาสคริปต์อื่น เช่น Perl , PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server-side script) ดังนั้น จึงต้องใช้บนเซิร์ฟเวอร์ที่สนับสนุนภาษาเหล่านี้เท่านั้น จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือ ไม่สามารถรับและส่งข้อมูลต่าง ๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจหรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้จึงยังคงต้องอาศัยภาษา server-side script แต่ภาษา JavaScript ก็สามารถทำงานบนเซิร์ฟเวอร์ได้แต่ต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะ

การทำงานของ JavaScript จะมีประสิทธิภาพมาก ถ้าใช้ดัดแปลงคุณสมบัติขององค์ประกอบต่าง ๆ บนเว็บเพจ (เช่น สี หรือรูปแบบของข้อความ) และสามารถรับรู้เหตุการณ์ที่ผู้ชมเว็บเพจโต้ตอบกับองค์ประกอบเหล่านั้น (เช่น การคลิก หรือเลื่อนเมาส์ไปวาง) ได้ ดังนั้นจากภาษา HTML เดิมที่มีลักษณะสถิต (static) ใน HTML เวอร์ชั่นใหม่ๆ จึงได้มีการพัฒนาให้มีคุณสมบัติบางอย่างเพิ่มขึ้นและมีลักษณะเป็นอ็อบเจกต์ (object) มากขึ้น การทำงานร่วมกันระหว่าง คุณสมบัติใหม่ของ HTML ร่วมกับ JavaScript นี้เอง ทำให้เกิดเป็นสิ่งที่เรียกว่า Dynamic HTML คือ ภาษา HTML ที่สามารถใช้สร้างเว็บเพจที่มีลักษณะพลวัต (dynamic) ได้นั่นเอง

นอกจากนี้ อีกองค์ประกอบหนึ่งที่เกี่ยวข้อง ก็คือ Cascading Style Sheet (CSS) ซึ่งเป็นภาษาที่ช่วยให้เราควบคุมรูปแบบขององค์ประกอบต่าง ๆ บนเว็บเพจ ได้อย่างมีประสิทธิภาพมากกว่า

คำสั่งหรือแท็ก (tag) ปกติของ HTML เนื่องจาก JavaScript สามารถดัดแปลงคุณสมบัติของ CSS ได้เช่นกัน ดังนั้นมันจึงช่วยให้เราควบคุมเว็บเพจ ได้อย่างมีประสิทธิภาพมากมากยิ่งขึ้นไปอีก

ข้อดีและข้อเสียของ JavaScript

การทำงานของ JavaScript เกิดขึ้นบนเว็บเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไรหรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ต่างกับภาษาสคริปต์ อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server side script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ดีจากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือ ไม่สามารถรับและส่งข้อมูลต่าง ๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชมเพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้จึงยังคงต้องอาศัยภาษา server side script อยู่ (ความจริง JavaScript ที่ทำงานบนเซิร์ฟเวอร์ก็มีซึ่งต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก)

2.3.5 CSS

ภาษา CSS คือ ชุดคำสั่งที่ใช้สำหรับการกำหนดการแสดงผลข้อมูลหน้าเว็บเพจ CSS ย่อจาก Cascading Style Sheets เป็นมาตรฐานหนึ่งของ W3C ที่กำหนดขึ้นมา เพื่อใช้ในการแต่งหน้าเว็บเพจโดยเฉพาะการใช้งาน CSS จะเข้ามาช่วยเพิ่มความสามารถให้กับ HTML และ ภาษา CSS มีความสามารถในการตบแต่งการแสดงผลข้อมูลหน้าเว็บเพจที่เหนือกว่า HTML

ความสามารถของภาษา CSS

ภาษา CSS มีความสามารถหลายอย่าง เช่น CSS สามารถทำให้ Text ที่เป็นจุด Link ไม่ให้มีการขีดเส้นใต้ได้ CSS สามารถกำหนดการ Fix ขนาดของ Font อักษรได้ คือ เมื่อผู้เยี่ยมชมปรับขนาด Font ที่ Broeser ที่ขนาดเท่าใด CSS ก็ยังคงแสดงผลขนาด Font ที่ขนาดที่ปรับไว้เสมอ ส่งผลให้เว็บเพจไม่ผิดปกติตามขนาดของ Font ที่ผู้ใช้ปรับเปลี่ยนที่ Browser , CSS ทำให้เว็บเพจโหลดเร็วขึ้น เป็นต้น เนื่องจาก CSS จะทำงานร่วมกับ HTML เป็นหลัก ดังนั้นจึงสามารถพิมพ์โค้ดของ CSS แทรกไปยังโค้ดของ HTML ได้

ประโยชน์ของ CSS

1. ช่วยให้เนื้อหาภายในเอกสาร HTML มีความเข้าใจได้ง่ายขึ้นและในการแก้ไขเอกสารก็สามารถทำได้ง่ายกว่าเดิม เพราะการใช้ CSS จะช่วยลดการใช้ภาษา HTML ลงได้ในระดับหนึ่ง และแยกระหว่างเนื้อหากับรูปแบบในการแสดงผลได้อย่างชัดเจน
2. ทำให้สามารถดาวน์โหลดไฟล์ได้เร็ว เนื่องจาก code ในเอกสาร HTML ลดลง จึงทำให้ไฟล์มีขนาดเล็กลง

3. สามารถกำหนดรูปแบบการแสดงผลจากคำสั่ง style sheet ชุดเดียวกัน ให้มีการแสดงผลในเอกสารแบบเดียวทั้งหน้าหรือในทุกๆ หน้าได้ ช่วยลดเวลาในการปรับปรุงและทำให้การสร้างเอกสารบนเว็บมีความรวดเร็วยิ่งขึ้น นอกจากนี้ยังสามารถควบคุมการแสดงผล ให้คล้ายหรือเหมือนกันได้ในหลาย Web Browser
4. ช่วยในการกำหนดการแสดงผลในรูปแบบที่มีความเหมาะกับสื่อต่างๆ ได้เป็นอย่างดี
5. ทำให้เว็บไซต์มีความเป็นมาตรฐานมากขึ้นและมีความทันสมัย สามารถรองรับการใช้งานในอนาคตได้ดี

2.3.6 Apache

Apache พัฒนามาจาก HTTPD Web Server ที่มีกลุ่มผู้พัฒนาอยู่แล้ว โดย ร็อบ แม็คคูล ที่ NCSA (National Center for Supercomputing Applications) มหาวิทยาลัยอิลลินอยส์ เออร์แบนาแชมเปญจน์ สหรัฐอเมริการ แต่หลังจากที่ แม็คคูล ออกจาก NCS แล้วเขาก็ได้หันไปสนใจกับโครงการอื่นๆ มากกว่า ทำให้ HTTPD Web Server ถูกปล่อยทิ้งไม่มีผู้ใดพัฒนาต่อเนื่องจากเป็น Web Server ที่อยู่ภายใต้ลิขสิทธิ์ คือ ทุกคนมีสิทธิ์ที่จะนำเอาซอร์สโค้ดไปพัฒนาต่อได้ทำให้มีผู้ที่พัฒนาโปรแกรมขึ้นมาเพียงเพื่ออุดช่องโหว่จากโปรแกรมเดิมเท่านั้นและยังได้รวบรวมเอาข้อมูลการพัฒนาและการแก้ไขต่างๆ แต่ข้อมูลเหล่านี้อยู่ตามที่ต่างๆ ไม่ได้รวมอยู่ในที่ที่เดียวกัน จนในที่สุด ไบอัน บีเลนดอร์ฟ ได้สร้างจดหมายกลุ่มขึ้นเพื่อนำเอาข้อมูลมาเก็บไว้ในที่ที่เดียวกันเพื่อให้สามารถเข้าถึงข้อมูลได้ง่ายขึ้นและในที่สุดกลุ่มผู้พัฒนาได้เรียกตัวเองว่า กลุ่มอะแพ ชี และได้ปล่อยซอฟต์แวร์ HTTPD Web Server ที่พัฒนาโดยการนำเอา Page หลายๆตัวที่ผู้ใช้ได้พัฒนาขึ้น มาปรับปรุงการทำงาน เพื่อให้มีประสิทธิภาพมากขึ้น

ความสามารถของ Apache

Apache เป็นซอฟต์แวร์ที่อยู่ในลักษณะของโอเพ่นซอร์ส ที่เปิดให้บุคคลทั่วไปสามารถเข้ามาพัฒนาส่วนต่างๆ ของอาปาเช่ได้ ทำให้เกิดเป็นโมดูลที่เกิดประโยชน์มาก เช่น mod_perl, mod_python หรือ mod_php ซึ่งโมดูลที่ทำให้ Apache สามารถใช้ประโยชน์และสามารถทำงานร่วมกับภาษาอื่นได้แทนที่จะเป็นเพียง Server ที่ให้บริการเพียงแค่ HTML อย่างเดียว นอกจากนี้ Apache ยังมีความสามารถอื่นๆ ด้วย เช่น การยืนยันตัวบุคคล (mod_auth, mod_access, mod_digest) หรือเพิ่มความปลอดภัยในการสื่อสารผ่าน โปรโตคอล https (mod_ssl)

นอกจากนี้ ยังมีโมดูลอื่นๆ อีกที่ได้รับความนิยมใช้กัน เช่น mod_vhost ทำให้สามารถสร้างโฮสท์เสมือน เช่น www.sample.com, wiki.sample.com, mail.sample.com หรือ www.ilovewiki.org ภายในเครื่องเดียวกันได้ หรือ mod_rewrite เป็นเครื่องมือที่ช่วยให้ url ของเว็บสามารถอ่านง่ายขึ้น

2.3.7 Web GIS / Web Map Application

Web GIS เป็นระบบสารสนเทศแบบกระจายซึ่งประกอบด้วยเซิร์ฟเวอร์และไคลเอ็นต์อย่างน้อยเซิร์ฟเวอร์เป็นเซิร์ฟเวอร์ GIS และไคลเอ็นต์เป็นเว็บเบราเซอร์แอปพลิเคชั่นเดสก์ท็อปหรือแอปพลิเคชั่นบนมือถือ ในรูปแบบที่เรียบง่ายของเว็บ GIS สามารถกำหนดเป็น GIS ใด ๆ ที่ใช้เทคโนโลยีเว็บเพื่อสื่อสารระหว่างเซิร์ฟเวอร์และไคลเอ็นต์

Web GIS นิยมใช้กันอย่างแพร่หลาย โดยเฉพาะการพัฒนาระบบข้อมูลสารสนเทศผ่านทางอินเทอร์เน็ต Web GIS นั้นเป็นระบบ GIS ที่ใช้เทคโนโลยีเว็บเพื่อการสื่อสารของระบบต่างๆ ดังนั้น Web GIS จึงประกอบด้วยเครื่องมือที่ใช้ในการจัดเก็บ เรียกค้น จัดการ วิเคราะห์ข้อมูลเชิงพื้นที่ ซึ่งอย่างน้อยที่สุด Web GIS ต้องมีหนึ่งลูกข่ายและหนึ่งเซิร์ฟเวอร์ ที่ทำงานบนระบบปฏิบัติการที่เป็น desktop app. หรือ web browser app. ที่ให้ผู้ใช้สามารถสื่อสารผ่าน server และ server จะทำหน้าที่เป็น Web server app. ด้วย

Map Application เป็นระบบ Web GIS ที่ผู้ใช้งานสามารถใช้งานผ่านเว็บเบราว์เซอร์ เช่น Internetexplorer Mozilla หรือ Netscape โดยผู้ใช้ไม่จำเป็นต้องติดตั้งซอฟต์แวร์ลงบนเครื่องคอมพิวเตอร์ ตัวอย่าง Web Map Application ได้แก่ Google Map, API, Map Server ข อ ง CATGIFT (Government Information For Thailand) เป็นต้น ซึ่งข้อดีของระบบ Web GIS แบบ Web Map Application คือ สามารถทำงานได้กับ เครื่องคอมพิวเตอร์ทุก Platform และประหยัดค่าใช้จ่ายในการจัดหาซอฟต์แวร์ แต่ข้อเสียคือในการใช้งานต้องเชื่อมต่อกับอินเทอร์เน็ตตลอดเวลา

ต่อไปนี้เป็นองค์ประกอบสำคัญบางอย่างที่จำเป็นสำหรับเว็บ GIS เซิร์ฟเวอร์มี URL เพื่อให้ลูกค้าสามารถค้นหาได้ทางเว็บลูกค้าอาศัยข้อกำหนด HTTP เพื่อส่งคำขอไปยังเซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำการดำเนินการ GIS ที่ร้องขอและส่งการตอบกลับไปยังไคลเอ็นต์ ผ่านทาง HTTP รูปแบบของการตอบกลับที่ส่งไปยังไคลเอ็นต์สามารถอยู่ในรูปแบบต่างๆ เช่น HTML, ภาพไบนารี, XML (Extensible Markup Language) หรือ JSON (JavaScript Object Notation)

ข้อดีของเว็บ

การใช้อินเทอร์เน็ตในการเข้าถึงข้อมูลผ่านทางเว็บโดยไม่คำนึงถึงว่าเซิร์ฟเวอร์และไคลเอ็นต์อาจแตกต่างกันอย่างไร GIS บนเว็บแนะนำข้อดีที่แตกต่างจาก GIS เดสก์ท็อปแบบเดิมซึ่ง ได้แก่สิ่งต่อไปนี้: การเข้าถึงทั่วโลกในฐานะผู้ใช้ ArcGIS คุณสามารถนำเสนอแอปพลิเคชันเว็บ GIS ไปทั่วโลกและสามารถเข้าถึงได้จากคอมพิวเตอร์หรือโทรศัพท์มือถือ ลักษณะทั่วโลกของเว็บ GIS สืบทอดมาจาก HTTP ซึ่งได้รับการสนับสนุนโดยทั่วไป เกือบทุกองค์กรเปิดไฟร์วอลล์ของพวกเขาที่พอร์ตเครือข่ายบางอย่างเพื่อให้คำขอ HTTP และการตอบสนองผ่านเครือข่ายท้องถิ่นของพวกเขา ซึ่งจะช่วยเพิ่มการเข้าถึงผู้ใช้จำนวนมากโดยทั่วไป GIS แบบเดสก์ท็อปแบบดั้งเดิมจะถูกใช้โดยผู้ใช้เพียงรายเดียวใน

ขณะที่ GIS เว็บสามารถใช้งานได้หลายสิบหรือหลายร้อยคนพร้อมกัน ดังนั้น GIS บนเว็บจำเป็นต้องมีประสิทธิภาพและความสามารถในการปรับขยายได้ดีกว่าเดสก์ท็อป GIS ความสามารถข้ามแพลตฟอร์มที่ดีขึ้น ลูกค้าเว็บ GIS ส่วนใหญ่เป็นเว็บเบราเซอร์ ได้แก่ Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome และอื่น ๆ เนื่องจากเว็บเบราเซอร์เหล่านี้ส่วนใหญ่ปฏิบัติตามมาตรฐาน HTML และ JavaScript GIS บนเว็บที่อาศัยไคลเอ็นต์ HTML มักจะสนับสนุนระบบปฏิบัติการต่างๆเช่น Microsoft Windows, Linux และ Apple Mac OS

ค่าใช้จ่ายต่างๆโดยเฉลี่ยของจำนวนผู้ใช้เนื้อหาทางอินเทอร์เน็ตส่วนใหญ่ไม่เสียค่าใช้จ่ายสำหรับผู้ใช้ปลายทางและนี่เป็นความจริงของ GIS บนเว็บ โดยทั่วไปคุณไม่จำเป็นต้องซื้อซอฟต์แวร์หรือจ่ายเงินเพื่อใช้ GIS บนเว็บ องค์กรที่ต้องการให้บริการ GIS แก่ผู้ใช้จำนวนมากสามารถลดต้นทุนของพวกเขาผ่านทางเว็บ GIS แทนที่จะจัดซื้อ GIS เดสก์ท็อปสำหรับผู้ใช้ทุกคนองค์กรสามารถตั้ง GIS เพียงหนึ่งเดียวและระบบเดียวนี้สามารถใช้ร่วมกันได้โดยผู้ใช้จำนวนมาก จากที่บ้านที่ทำงานใช้งานง่าย Desktop GIS เหมาะสำหรับผู้ใช้มืออาชีพที่มีเดือนฝึกอบรมและมีประสบการณ์ใน Web GIS มีไว้สำหรับผู้ชมทั่วไปรวมทั้งผู้ใช้สาธารณะที่อาจไม่รู้อะไรเกี่ยวกับ GIS พวกเขาคาดว่า GIS เว็บจะง่ายเหมือนกับการใช้เว็บไซต์ปกติ Web GIS ได้รับการออกแบบมาเพื่อความเรียบง่ายและความสะดวกสบายทำให้ใช้งานได้ง่ายกว่าเดสก์ท็อป GIS

การปรับปรุงแบบครบวงจรสำหรับเดสก์ท็อป GIS ที่จะได้รับการอัปเดตเป็นเวอร์ชันใหม่ การอัปเดตต้องได้รับการติดตั้งในคอมพิวเตอร์ทุกเครื่อง สำหรับ GIS เว็บหนึ่งการอัปเดตจะใช้ได้ผลกับลูกค้าทั้งหมด ความสะดวกในการบำรุงรักษานี้ทำให้ GIS เว็บสมบูรณ์แบบ สำหรับการส่งมอบข้อมูลแบบเรียลไทม์แอปพลิเคชันที่หลากหลายแตกต่างจาก GIS บนเดสก์ท็อปซึ่งจำกัด เฉพาะผู้เชี่ยวชาญด้านระบบสารสนเทศภูมิสารสนเทศหลายแห่งเว็บเซอร์วิส GIS สามารถใช้งานได้กับทุกคนในองค์กรหรือสาธารณชนทั่วไป ผู้ชมกลุ่มนี้มีความต้องการที่หลากหลาย แอปพลิเคชันเช่นการทำแผนที่ที่อยู่อาศัยที่มีชื่อเสียงการติดแท็กรูปถ่ายส่วนตัวการค้นหาเพื่อน ๆ และการแสดงจุดร้อน Wi-Fi เป็นตัวอย่างเว็บที่เป็นปัจจุบันของ GIS จำนวนมาก

ลักษณะเหล่านี้แสดงให้เห็นถึงข้อดีและความท้าทายที่ GIS เว็บต้องเผชิญ ตัวอย่างเช่นลักษณะการใช้งานง่ายของ GIS บนเว็บช่วยกระตุ้นการมีส่วนร่วมของประชาชน แต่ยังเป็นการเตือนให้คุณคำนึงถึงผู้ใช้อินเทอร์เน็ตที่ไม่มีภูมิหลัง GIS ตรงกันข้ามการสนับสนุนผู้ใช้จำนวนมากต้องการให้ GIS เว็บสามารถปรับขนาดได้

งานวิจัยที่เกี่ยวข้องด้าน Web Map Application

Bhatia, Singh, Litoria, & Pateriya. (2018) ความนิยมของเว็บตามสถานที่และแอปพลิเคชันมือถือเพิ่มขึ้นทุกวัน การแสดงภาพตามสถานที่นั้นมีความซับซ้อนกว่าข้อมูลแบบตารางเล็กน้อย ต้องใช้ซอฟต์แวร์เฉพาะเพื่อสร้างแอปพลิเคชัน GPS บนเว็บ เพื่อให้เข้าถึงข้อมูลเชิงพื้นที่ได้ง่ายสำหรับผู้คนจำนวนมาก บทความนี้ศึกษาเกี่ยวกับวิธีการที่เรียบง่ายและใช้งานง่ายในการสร้างแอปพลิเคชัน GIS บนเว็บ โดยใช้เครื่องมือโอเพนซอร์ซ มีการใช้ Leaflet, Geoserver, PostgreSQL เพื่อสร้างแอปพลิเคชัน ซึ่ง Leaflet ใช้เพื่อสร้าง GUI (การใช้ภาพเป็นตัวประสานกับผู้ใช้) Geoserver ทำหน้าที่เป็นซอฟต์แวร์กลางที่เชื่อม Client และ Server เข้าด้วยกัน และในส่วนของ PostgreSQL ถูกใช้เป็นที่เก็บข้อมูล Backend ซึ่งเว็บ GIS มีฟังก์ชันพื้นฐาน เช่น การเลื่อน การซูม หน้าแรก หน้าต่าง ข้อมูลการวัด ปุ่มคำอธิบาย ตำแหน่งทางภูมิศาสตร์ แถบค้นหาการเลือก เลเยอร์ แผนที่ขนาดเล็ก การระบุแหล่งที่มา อื่นๆ

กัลยรัตน์ ชื่นบาน จักกริช พฤษการ และศิวดล เสถียรพัฒนากูล (2556) ศึกษาเกี่ยวกับการพัฒนาเว็บแอปพลิเคชัน เพื่ออำนวยความสะดวกในการค้นหาเส้นทาง โดยเว็บจะแสดงบริเวณและเส้นทางที่น้ำท่วม และหาเส้นทางที่เหมาะสมแก่การเดินทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง ซอฟต์แวร์ที่ใช้ในการพัฒนาเว็บแอปพลิเคชัน ได้แก่ Appserv 2.5.10 คือ Apache 2.2.8 PHP 5.2.6 MySQL 5.0.51b phpMyAdmin-2.10.3, Windows 7 Ulitimate และ Dreamweaver CS3 โดยผู้ดูแลระบบสามารถกำหนดพื้นที่เสี่ยงภัยโดยการปักหมุนโดยใช้ Google Maps API และกำหนดพื้นที่ต้นทาง และปลายทางเพื่อหลีกเลี่ยงพื้นที่เสี่ยงภัย จากการทดสอบพบว่าระบบสามารถแสดงเส้นทางที่เลี่ยงน้ำท่วม และเป็นเส้นทางที่สั้นที่สุด ผู้ดูแลกำหนดหมุนโดยใช้ค่าละติจูด ลองจิจูด จากกูเกิลแมพและมีการกำหนดโหนดข้างเคียงของแต่ละหมุดว่ามีอะไรบ้าง เพื่อช่วยในการหาเส้นทางตามหลักอัลกอริทึม Dijkstra’s

สวรินทร์ ฤกษ์อยู่สุข วิโรจน์ ละอองมณี และภาณุ อุทัยศรี (2556) ศึกษาซอฟต์แวร์รหัสเปิดและแหล่งข้อมูลภูมิสารสนเทศแบบเปิดที่ไม่เสียค่าใช้จ่าย สำหรับการประยุกต์ใช้งานสารสนเทศภูมิสารสนเทศ เนื่องจากข้อมูลภูมิสารสนเทศ และเครื่องมือสำหรับการประมวลผลบางประเภทมีราคาที่สูง จึงได้ทำการค้นหา รวบรวม และเผยแพร่แหล่งข้อมูลสารสนเทศแบบเปิดที่ไม่เสียค่าใช้จ่ายและซอฟต์แวร์สารสนเทศที่มีประสิทธิภาพ จากการศึกษาสามารถสรุปได้ว่าซอฟต์แวร์รหัสเปิดและแหล่งข้อมูลภูมิสารสนเทศแบบเปิดที่ไม่เสียค่าใช้จ่าย ซอฟต์แวร์รหัสเปิด ได้แก่ Web Mapping, GeoServer, MapServer, OpenLayers, Quantum GIS, GRASS, Geospatial Data Abstraction Library (GDAL), OpenGIS Simple Features Reference Implementation (OGR), PostGIS, Open Source Software Image Map (OSSIM) และ แหล่งข้อมูลภูมิสารสนเทศ ได้แก่ Global Land Cover Facility (GLCF), EarthExplorer เป็นต้น จากการศึกษาจะเห็นถึงประโยชน์จากการใช้

บริการแหล่งข้อมูลภูมิสารสนเทศแบบเปิดที่ไม่มีค่าใช้จ่าย และซอฟต์แวร์รหัสเปิดเป็นพื้นฐานที่สำคัญที่จะสามารถช่วยให้การเรียน การสอนด้านภูมิสารสนเทศมีประโยชน์มากอย่างยิ่ง เนื่องจากผู้ใช้งานซอฟต์แวร์รหัสเปิดสามารถเรียนรู้ และทำความเข้าใจถึงหลักการทำงานของซอฟต์แวร์รหัสเปิดภูมิสารสนเทศได้อย่างสะดวก

จากการทบทวนวรรณกรรมที่เกี่ยวข้อง ผู้วิจัยพบว่า การพัฒนาระบบแพลตฟอร์มในรูปแบบ Web-GIS เป็นการทำงานที่มีความสะดวกรวดเร็ว เป็นการทำงานที่ไม่ต้องเสียค่าใช้จ่ายในการพัฒนาแสดงผลหรือรายงานสถานการณ์ที่เกิดขึ้นในเชิงพื้นที่ ทำให้ผู้ใช้งานสามารถเรียนรู้และทำความเข้าใจหลักการทำงานได้เป็นอย่างดี ผู้วิจัยได้นำการทบทวนวรรณกรรมในเรื่องของ Web-GIS มาใช้ในงานวิจัย โดยภาษาที่ใช้ในการพัฒนา ได้แก่ Javascript, php, html เป็นภาษาหลักในการพัฒนาขึ้นมาโดยให้สามารถดึงข้อมูลเข้าสู่ Server มาแสดงบนแผนที่ OpenLayers ได้แบบเรียลไทม์ โดย Geoserver ทำหน้าที่เป็นซอฟต์แวร์กลางที่เชื่อม Client และ Server เข้าด้วยกัน และมีโปรแกรม PosrgreSQl/PostGIS เป็นโปรแกรมในการจัดการฐานข้อมูลด้วยภาษา SQL

2.4 การบริการการแพทย์ฉุกเฉิน

2.4.1 การบริการการแพทย์ฉุกเฉิน (emergency medical service: EMS)

1. พระราชบัญญัติการแพทย์ฉุกเฉิน พ.ศ. 2551

ปัจจุบันประเทศไทยเข้าสู่ในยุคที่ 4 ของระบบการแพทย์ฉุกเฉิน ซึ่งเริ่มตั้งแต่ พ.ศ.2537 – 2556 โดยยุคที่ 4 เป็นยุคหลังประกาศใช้พระราชบัญญัติการแพทย์ฉุกเฉิน พ.ศ.2551 มีการจัดตั้งสถาบันการแพทย์ฉุกเฉินแห่งชาติ (สพฉ.) เพื่อรองรับระบบการแพทย์ฉุกเฉินที่พัฒนาขึ้นอย่างต่อเนื่อง ซึ่งภายหลังการจัดตั้ง สพฉ. ก็ได้รับการสนับสนุนให้มีการพัฒนาในส่วนอื่น ๆ เช่น การพัฒนาระบบบริการของหน่วยกู้ชีพตามโรงพยาบาล บริการห้องฉุกเฉิน และคงไว้ซึ่งบริการช่องทางด่วนสำหรับโรคต่าง ๆ นอกจากนี้ยังมีแนวคิดให้องค์กรปกครองส่วนท้องถิ่น (อปท.) เข้ามามีบทบาทหลักเฉพาะบริการกู้ชีพ มีการพัฒนา 3 องค์ประกอบ ได้แก่ กำลังคน อุปกรณ์และครุภัณฑ์ และเงินอุดหนุนบริการ (จารุวรรณ ธาดาเดช, สิริมา มงคลสัมฤทธิ์, และไพบูลย์ สุริยะวงศ์ไพศาล, 2557)

ตามพระราชบัญญัติการแพทย์ฉุกเฉิน พ.ศ.2551 ประกาศใช้ตามราชกิจจานุเบกษา เมื่อวันที่ 1 มีนาคม 2552 มีผลบังคับใช้ในวันถัดจากวันประกาศในราชกิจจานุเบกษา ให้คำนิยาม การแพทย์ฉุกเฉิน ผู้ป่วยฉุกเฉิน ปฏิบัติการแพทย์ฉุกเฉิน ดังนี้ (สถาบันการแพทย์ฉุกเฉินแห่งชาติ, 2552)

การแพทย์ฉุกเฉิน หมายถึง การปฏิบัติการฉุกเฉิน การศึกษา ค้นคว้า อบรมและวิจัยที่เกี่ยวกับการประเมิน การจัดการ บำบัดรักษา และป้องกันการเจ็บป่วยที่เกิดขึ้นอย่างฉุกเฉิน

ผู้ป่วยฉุกเฉิน หมายถึง บุคคลที่ได้รับบาดเจ็บหรือเจ็บป่วยกะทันหันซึ่งมีอันตรายต่อชีวิต หรือจำเป็นต้องได้รับการประเมิน การจัดการ และบำบัดรักษาอย่างทันท่วงที เพื่อป้องกันการเสียชีวิตหรือลดความรุนแรงของการบาดเจ็บหรือเจ็บป่วยนั้น

ปฏิบัติการแพทย์ฉุกเฉิน หมายถึง ปฏิบัติการด้านการแพทย์ฉุกเฉินนับตั้งแต่รับรู้ถึงการเจ็บป่วยฉุกเฉินจนถึงการดำเนินการให้ผู้ป่วยฉุกเฉินได้รับการรักษาให้พ้นภาวะฉุกเฉิน รวมถึงการประเมิน การจัดการ การประสานงาน การควบคุมดูแล การติดต่อสื่อสาร การลำเลียงหรือขนส่ง การวินิจฉัย และการบำบัดรักษาพยาบาลผู้ป่วยฉุกเฉินทั้งนอกสถานพยาบาลและในสถานพยาบาล

2.4.2 รูปแบบบริการการแพทย์ฉุกเฉิน

บริการการแพทย์ฉุกเฉิน หมายถึง การให้การปฏิบัติการช่วยเหลือผู้ป่วยฉุกเฉินให้พ้นจากความเจ็บป่วยที่เป็นภาวะวิกฤตโดยปฏิบัติการทั้งในภาวะปกติและภัยพิบัติ มีการปฏิบัติงานที่รวดเร็วและเป็นระบบตลอด 24 ชั่วโมง มีระบบรับแจ้งเหตุ การให้คำแนะนำเมื่อมีอาการฉุกเฉินทางสุขภาพ การประสานงานเพื่อจัดระดมทรัพยากรในพื้นที่เพื่อช่วยเหลือผู้ป่วยตั้งแต่จุดเกิดเหตุ ลำเลียงเคลื่อนย้าย และนำส่งสถานบริการสุขภาพปลายทางได้อย่างปลอดภัย (อัจฉริยะ แพงมา, 2555)

ปัจจุบันรูปแบบบริการการแพทย์ฉุกเฉินที่ใช้กันมี 2 รูปแบบ คือ (1) Anglo-American Model (AAM) เน้นการตักแล้ววิ่ง (scoop an run) ใช้เวลา ณ จุดเกิดเหตุสั้น นำส่งโรงพยาบาลอย่างรวดเร็วที่สุด มีการให้การดูแลรักษาผู้ป่วยก่อนถึงโรงพยาบาลโดยบุคลากรที่ไม่ใช่แพทย์แต่อยู่ภายใต้การกำกับดูแลของแพทย์ ใช้รถพยาบาลเป็นหลักในการเคลื่อนย้ายผู้ป่วย เป้าหมายปลายทางคือ ส่งตรงห้องฉุกเฉิน ซึ่งประเทศที่ใช้ AAM ได้แก่ สหราชอาณาจักร สหรัฐอเมริกา แคนาดา นิวซีแลนด์ ออสเตรเลีย และประเทศไทย และ (2) Franco-German Model (FGM) ซึ่งเน้นการรักษาพยาบาล ณ จุดเกิดเหตุ (stay and stabilize) มีแพทย์ให้การดูแลเป็นหลัก และมีทีมเวชกิจฉุกเฉินในการช่วย มีการใช้เทคโนโลยีขั้นสูง การลำเลียงผู้ป่วยเป้าหมายปลายทางคือ ส่งหน่วยเฉพาะทางทั้งด้วยรถพยาบาล เฮลิคอปเตอร์ และรถพยาบาลชายฝั่ง (coastal ambulance) ค่าใช้จ่ายจะต่ำกว่า AAM เนื่องจากมีผู้ป่วยจำนวนน้อยที่นำส่งโรงพยาบาล ส่วนใหญ่จะรักษา ณ จุดเกิดเหตุ ประเทศที่ใช้ AAM คือ เยอรมนี ฝรั่งเศส กรีซ มอลต้า และออสเตรีย (จารุวรรณ ธาดาเดช และคณะ, 2557)

2.4.3 เกณฑ์การคัดแยกผู้ป่วยฉุกเฉินและจัดลำดับการบริบาล

พระราชบัญญัติการแพทย์ฉุกเฉิน พ.ศ. 2551 ได้กำหนดนิยาม “ผู้ป่วยฉุกเฉิน” หมายความว่า บุคคลซึ่งได้รับบาดเจ็บหรือมีอาการป่วยกะทันหัน ซึ่งเป็นภยันตรายต่อการดำรงชีวิตหรือการทำงานของอวัยวะสำคัญ จำเป็นต้องได้รับการประเมิน การจัดการ และบำบัดรักษาอย่างทันท่วงที เพื่อป้องกันการเสียชีวิตหรือการรุนแรงขึ้นของการบาดเจ็บหรืออาการป่วยนั้น และสถาบันการแพทย์

แห่งชาติ (สพฉ.) ได้มีประกาศคณะกรรมการการแพทย์ฉุกเฉิน เรื่อง หลักเกณฑ์การประเมินเพื่อคัดแยกระดับความฉุกเฉิน และมาตรฐานการปฏิบัติการฉุกเฉิน พ.ศ. 2554 เพื่อการคุ้มครองความปลอดภัยของผู้ป่วยฉุกเฉินให้สถานพยาบาล หน่วยปฏิบัติการ และผู้ปฏิบัติการ ดำเนินการตรวจคัดแยกระดับความฉุกเฉิน และจัดให้ผู้ป่วยฉุกเฉินได้รับการปฏิบัติการฉุกเฉินตามลำดับความเร่งด่วนทางการแพทย์ฉุกเฉิน แบ่งเป็น 5 ระดับ ตามความเร่งด่วนในการปฏิบัติการ โดยกำหนดให้ใช้เกณฑ์การประเมิน เพื่อคัดแยกระดับความฉุกเฉิน ณ ห้องฉุกเฉินตามระบบ Emergency Severity Index (ESI) Version 4 แบบ 5 ระดับ คือ

ผู้ป่วยฉุกเฉินวิกฤต ได้แก่ บุคคลซึ่งได้รับการบาดเจ็บหรือมีอาการป่วยกะทันหัน ที่มีภาวะคุกคามต่อชีวิต หากไม่ได้รับการปฏิบัติการแพทย์ทันทีเพื่อแก้ไขระบบการหายใจ ระบบไหลเวียนเลือด หรือระบบประสาทแล้ว ผู้ป่วยอาจมีโอกาสเสียชีวิตได้สูง ทำให้การบาดเจ็บ อาการป่วยของผู้ป่วยฉุกเฉินนั้นรุนแรงขึ้น หรือเกิดภาวะแทรกซ้อนขึ้นได้อย่างฉับไว ให้ใช้สัญลักษณ์ “สีแดง” สำหรับผู้ป่วยฉุกเฉินวิกฤต

ผู้ป่วยฉุกเฉินเร่งด่วน ได้แก่ บุคคลที่ได้รับบาดเจ็บหรืออาการป่วย ซึ่งมีภาวะเฉียบพลันมากหรือเจ็บป่วยรุนแรง อาจจำเป็นต้องได้รับการปฏิบัติการแพทย์อย่างเร่งด่วน มิฉะนั้นอาจทำให้เกิดบาดเจ็บหรืออาการป่วยของผู้ป่วยฉุกเฉินนั้นรุนแรงขึ้น หรือเกิดภาวะแทรกซ้อนได้ ซึ่งส่งผลให้เสียชีวิตหรือพิการในระยะต่อมาได้ ให้ใช้สัญลักษณ์ “สีเหลือง” สำหรับผู้ป่วยฉุกเฉินเร่งด่วน

ผู้ป่วยฉุกเฉินไม่รุนแรง ได้แก่ บุคคลซึ่งได้รับบาดเจ็บหรือมีอาการป่วย ซึ่งมีภาวะเฉียบพลันไม่รุนแรง อาจรอรับการปฏิบัติการแพทย์ได้ในช่วงระยะเวลาหนึ่งหรือเดินทางไปรับการบริการสาธารณสุขด้วยตนเองได้ แต่จำเป็นต้องใช่ทรัพยากร และหากปล่อยไว้กินเวลาอันสมควรแล้วจะทำให้การบาดเจ็บหรืออาการป่วยขอผู้ป่วยฉุกเฉินนั้นรุนแรงขึ้นหรือภาวะแทรกซ้อนขึ้นได้ ให้ใช้สัญลักษณ์ “สีเขียว” สำหรับผู้ป่วยฉุกเฉินไม่รุนแรง

ผู้ป่วยทั่วไป ได้แก่ บุคคลที่เจ็บป่วยแต่ไม่ใช่ผู้ป่วยฉุกเฉิน ซึ่งอาจรอรับหรือเลือกสรรบริการสาธารณสุขในเวลาทำการปกติได้ โดยไม่ก่อให้เกิดอาการที่รุนแรงขึ้นหรือภาวะแทรกซ้อนตามมา ให้ใช้สัญลักษณ์ “สีขาว” สำหรับผู้ป่วยฉุกเฉินทั่วไป

ผู้รับบริการสาธารณสุขอื่น ได้แก่ บุคคลซึ่งมารับบริการสาธารณสุขหรือบริการอื่น โดยไม่จำเป็นต้องใช้ทรัพยากร ให้ใช้สัญลักษณ์ “สีดำ” สำหรับผู้บริการสาธารณสุขอื่น

เกณฑ์การคัดแยกผู้ป่วยฉุกเฉิน และจัดลำดับการบริบาล ณ ห้องฉุกเฉิน ยังถูกแบ่งออกเป็น 25 กลุ่มของอาการของผู้ป่วยอีกด้วย โดย 25 กลุ่มของอาการนี้ ได้แก่

กลุ่มอาการที่ 1 ปวดท้อง / หลัง / เชิงกรานและขาหนีบ

กลุ่มอาการที่ 2 แอนาฟิแล็กซิส / ปฏิกิริยาภูมิแพ้

กลุ่มอาการที่ 3 สัตว์กัด

กลุ่มอาการที่ 4 เลือดออก (ไร้เหตุบาดเจ็บ)

กลุ่มอาการที่ 5 หายใจยากลำบาก

กลุ่มอาการที่ 6 หัวใจหยุดเต้น

กลุ่มอาการที่ 7 เจ็บแน่นทรวงอก / หัวใจ

กลุ่มอาการที่ 8 สำลักอุดกั้นทางเดินหายใจ

กลุ่มอาการที่ 9 เบาหวาน

กลุ่มอาการที่ 10 ภยันตรายจากสภาพแวดล้อม

กลุ่มอาการที่ 11 (เว้นว่าง)

กลุ่มอาการที่ 12 ปวดศีรษะ / ลำคอ

กลุ่มอาการที่ 13 คลุ้มคลั่ง / จิตประสาท / อารมณ์

กลุ่มอาการที่ 14 ยาเกินขนาด / ได้รับพิษ

กลุ่มอาการที่ 15 มีครรภ์ / คลอด / นรีเวช

กลุ่มอาการที่ 16 ชัก

กลุ่มอาการที่ 17 ป่วย / อ่อนเพลีย (ไม่จำเพาะ) / อื่นๆ

กลุ่มอาการที่ 18 แขนขาอ่อนแรง / พูดลำบาก / ปากเบี้ยว (หลอดเลือดสมองอุดตัน/แตก)

กลุ่มอาการที่ 19 หมดสติ / ไม่ตอบสนอง / หมดสติชั่ววูบ

กลุ่มอาการที่ 20 เด็ก / ทารก (กุมารเวชกรรม)

กลุ่มอาการที่ 21 ถูกทำร้าย / บาดเจ็บ

กลุ่มอาการที่ 22 ไหม้ / ลวก – ความร้อน / กระแสไฟฟ้า / สารเคมี

กลุ่มอาการที่ 23 จมน้ำ / หน้าคว่ำจมน้ำ / บาดเจ็บเหตุดำน้ำ / บาดเจ็บทางน้ำ

กลุ่มอาการที่ 24 พลัดตกหกล้ม / อุบัติเหตุ / เจ็บปวด

กลุ่มอาการที่ 25 อุบัติเหตุยานยนต์

ขีดความสามารถของหน่วยปฏิบัติการหรือสถานพยาบาลในการปฏิบัติการฉุกเฉิน ให้ถือตามกฎหมาย กฎ คำสั่ง และข้อมูลข่าวสารเกี่ยวกับการจัดตั้งหรือการได้รับอนุญาต รวมทั้งการดำเนินการและภาระรับผิดชอบของหน่วยปฏิบัติการหรือสถานพยาบาลนั้น

2.4.4 ลักษณะการปฏิบัติการฉุกเฉินในระบบบริการการแพทย์ฉุกเฉิน

ระบบบริการการแพทย์ฉุกเฉินของประเทศไทยมีเป้าหมายของการดำเนินการเพื่อให้การดูแลรักษาผู้ป่วยฉุกเฉินได้ทันเวลา ครอบคลุม เท่าเทียมทุกพื้นที่ มีขั้นตอนและหน้าที่ ดังนี้ (สมาคมเวชศาสตร์ฉุกเฉินแห่งประเทศไทย, 2560)

1. การการพบเหตุฉุกเฉิน (detection) คือ ในขั้นตอนนี้ ผู้ที่มีความสำคัญคือ ประชาชนที่อยู่ใกล้เคียงกับผู้ป่วย เช่น ญาติ พี่น้อง ผู้ดูแล หรือตัวผู้ป่วยเอง เป็นต้น ซึ่งจะเป็นผู้ที่สังเกตพบความผิดปกติต่าง ๆ ที่เป็นภาวะเจ็บป่วยฉุกเฉิน หากพบเร็วก็สามารถแก้ไขได้เร็ว หากพบช้าก็อาจทำให้ผู้ป่วยได้รับการแก้ไขล่าช้า และเสียโอกาสไป ซึ่งอาจนำพาความสูญเสียทางร่างกาย ความพิการ และเสียชีวิตได้ ดังนั้น ในขั้นตอนนี้ประชาชนควรมีความรู้ความเข้าใจในการสังเกตภาวะฉุกเฉินของตนเอง

2. การแจ้งเหตุขอความช่วยเหลือ (reporting) หลังจากพบภาวะเจ็บป่วยฉุกเฉิน ขั้นตอนต่อไปคือ ประชาชนจะแจ้งเหตุขอความช่วยเหลือ ตามหลักการคือ การแจ้งเหตุด้วยโทรศัพท์ไปยังศูนย์รับแจ้งเหตุ ซึ่งประเทศไทยใช้หมายเลขโทรศัพท์สายตรง คือ 1669 โทรฟรี ไม่มีค่าใช้จ่าย แต่ในบางพื้นที่อาจมีหมายเลขอื่นร่วมด้วย และการใช้แอปพลิเคชัน ThaiEMS 1669 ในการแจ้งภาวะเจ็บป่วยฉุกเฉินควรทำอย่างมีสติ ประกอบด้วย อาการและสาเหตุความเจ็บป่วยฉุกเฉินที่พบ เวลา สถานที่ เส้นทางเข้าออกที่ชัดเจนเข้าใจง่าย และภัยคุกคามอื่น ๆ ที่อาจมีผลกระทบต่อการให้การช่วยเหลือผู้ป่วยด้วย

3. การรับแจ้งเหตุและสั่งการ (reporting and commanding) เป็นหน้าที่ของศูนย์สั่งการและรับแจ้งเหตุ ในประเทศไทย แต่ละจังหวัดจะมีศูนย์รับแจ้งเหตุและสั่งการ โดยมีแพทย์เป็นผู้กำกับดูแลเพื่อรองรับความถูกต้องในกระบวนการตัดสินใจที่เกี่ยวกับการแพทย์หรือเวชกรรม เจ้าหน้าที่รับแจ้งเหตุจะมีช่องทางการรับแจ้งเหตุทั้งทางวิทยุสื่อสารและทางโทรศัพท์ จากนั้นจะวิเคราะห์ข้อมูลข่าวสาร ประเมินสถานการณ์ และสั่งการและจ่ายงานให้แก่หน่วยปฏิบัติการที่เหมาะสมกับเหตุ ซึ่งมี 4 ระดับ คือ หน่วยปฏิบัติการฉุกเฉินระดับสูง (advance lift support) หน่วยปฏิบัติการฉุกเฉินระดับกลาง (intermediate life support) หน่วยปฏิบัติการฉุกเฉินระดับต้น (basic life support) และหน่วยปฏิบัติการฉุกเฉินเบื้องต้น (first responder) การปฏิบัติงานของทุกหน่วยจะเป็นไปตามมาตรฐานกำหนด ซึ่งขณะรถหน่วยปฏิบัติการออกเหตุ ศูนย์สั่งการและรับแจ้งเหตุจะทำหน้าประสานงานกับรถพยาบาล นอกจากนี้ศูนย์รับแจ้งเหตุและสั่งการยังทำหน้าที่ในการให้คำแนะนำในการปฐมพยาบาล (first aid) ของผู้ที่อยู่ ณ จุดเกิดเหตุ เนื่องจากขณะที่กำลังมีการแจ้งเหตุขอความช่วยเหลือ หรือรอรถพยาบาลมารับ ณ จุดเกิดเหตุ ซึ่งประชาชนหรือผู้พบเห็นเหตุการณ์ควรทำการปฐมพยาบาลให้กับผู้ป่วยที่มีภาวะฉุกเฉินตามความสามารถและความเหมาะสม เพื่อบรรเทาความเจ็บป่วยหรือภาวะคุกคามต่อชีวิตบางอย่างอาจได้รับการแก้ไขหรือยุติลงได้

การรับแจ้งเหตุและสั่งการมีมาตรฐานคุณภาพงาน ดังนี้ (1) การรับแจ้งเหตุ สอบถามข้อมูลได้แก่ สาเหตุ สถานที่เกิดเหตุ หมายเลขโทรศัพท์ติดต่อกลับ และอาการสำคัญ ภายใน 45 วินาที โดยเจ้าหน้าที่จะเป็นผู้สอบถามข้อมูลตามแบบบันทึกการรับแจ้งเหตุและสั่งการ (2) หน่วยปฏิบัติการออกให้การช่วยเหลือ ภายใน 90 วินาที ซึ่งระยะเวลาตั้งแต่รับแจ้งเหตุจนสามารถจัดชุดปฏิบัติการ (dispatch time) ได้ต้องไม่เกิน 2 นาที และ (3) หน่วยปฏิบัติการฉุกเฉินออกให้การช่วยเหลือผู้ป่วย

ฉุกเฉินตามคำสั่งของศูนย์ฯ ซึ่งระยะเวลาตั้งแต่รับแจ้งเหตุถึงเวลาที่หน่วยปฏิบัติการถึงจุดเกิดเหตุหรือผู้ป่วยฉุกเฉิน (response time) ภายใน 10 นาที (ชุดปฏิบัติการระดับสูง) และภายใน 15 นาที (ชุดปฏิบัติการระดับพื้นฐาน)

4. การรักษาพยาบาล ณ จุดเกิดเหตุ (on scene care)/ ปฏิบัติการฉุกเฉินของชุดปฏิบัติการ (out-of-hospital ambulance operations) ชุดปฏิบัติการที่ออกปฏิบัติการรับผู้ป่วยฉุกเฉินตามคำสั่งของศูนย์สั่งการและแจ้งเหตุจะเป็นผู้ให้การดูแลรักษาผู้ป่วยฉุกเฉิน ณ จุดเกิดเหตุ โดยมีหลักในการดูแลคือ ใช้เวลา ณ จุดเกิดเหตุสั้น นำส่งโรงพยาบาลอย่างรวดเร็วที่สุด ซึ่งการปฏิบัติการ ณ จุดเกิดเหตุจะต้องได้รับการรับรองโดยผู้ประกอบวิชาชีพเวชกรรม ปฏิบัติการฉุกเฉินต้องอ้างอิงได้ว่าเป็นไปตามแนวทางปฏิบัติ วิธีปฏิบัติ หรือการบริหารยาจาก พ.ร.บ.เวชกรรม พ.ศ.2525 หากปฏิบัติการที่ขาดหรือเกินไปจากที่ผู้ประกอบวิชาชีพเวชกรรมรับรองไม่ว่าจะเป็นด้วยลายลักษณ์อักษรหรือวาจาให้ถือว่าเป็นความรับผิดชอบของผู้ปฏิบัติการเอง ซึ่งชุดปฏิบัติการต้องดำเนินการด้วยความรวดเร็ว สั้นกระชับและปลอดภัย การเลือกเส้นทางการเดินทางไปรับผู้ป่วยที่เหมาะสมจึงมีผลอย่างมาก

5. การลำเลียงขนย้ายและการดูและระหว่างนำส่ง (care in transit) หลักการสำคัญคือ การไม่ทำให้เกิดการบาดเจ็บซ้ำ ขณะเคลื่อนย้ายและนำส่งต้องมีการประเมินสภาพผู้ป่วยเป็นระยะ ๆ และให้การดูแลรักษาตามความเหมาะสม เช่น การให้ออกซิเจนและสารน้ำทางหลอดเลือดดำ เป็นต้น นอกจากนี้การลำเลียงขนย้ายควรให้ความสำคัญอย่างมากกับความปลอดภัยและความรวดเร็ว โดยเฉพาะอย่างยิ่งผู้ป่วยที่มีภาวะคุกคามต่อชีวิต การใช้สัญญาณฉุกเฉินและความเร็วให้เป็นไปตามระบบกำกับดูแลในพื้นที่กำหนด

6. การนำส่งสถานพยาบาล (transfer to definitive care) การนำส่งสถานพยาบาลที่ดีควรมีการแจ้งข้อมูลที่สำคัญของผู้ป่วยล่วงหน้า เพื่อให้สถานพยาบาลปลายทางได้เตรียมพร้อมรับผู้ป่วย เช่น อาการสำคัญ สาเหตุ อาการขณะนำส่ง เป็นต้น ซึ่งศูนย์รับแจ้งเหตุและสั่งการจะทำหน้าที่ในการพิจารณาสถานพยาบาลปลายทางที่เหมาะสมและประสานงานให้ หลักในการเลือกสถานพยาบาลคือความสามารถในการดูแลรักษาผู้ป่วยฉุกเฉิน ความใกล้ หรืออยู่ในเขตความรับผิดชอบ และอาจพิจารณาหลักประกันและความต้องการของผู้ป่วยหรือญาติประกอบด้วย เมื่อชุดปฏิบัติการไปถึงสถานพยาบาลปลายทาง จะส่งผู้ป่วย ณ ห้องฉุกเฉิน มีการส่งต่อข้อมูลผู้ป่วยและการรักษาเบื้องต้นที่ได้รับให้แก่พยาบาลและเจ้าหน้าที่ที่เกี่ยวข้องด้วยวาจาและลายลักษณ์อักษรเพื่อให้ผู้ป่วยได้รับการรักษาอย่างต่อเนื่องและเหมาะสมต่อไป

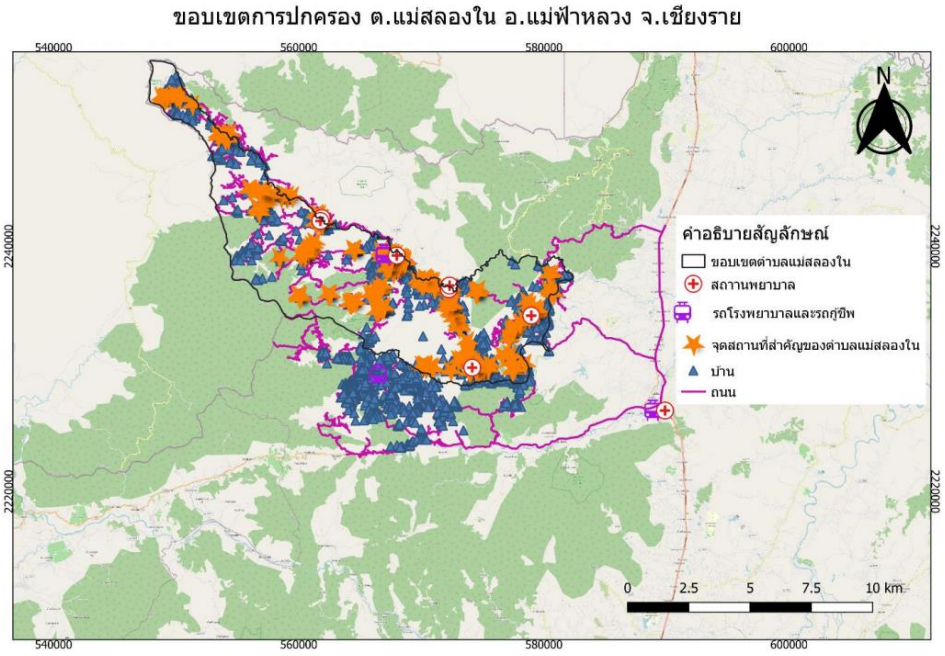
บทที่ 3
การพัฒนาระบบ

การพัฒนาระบบแจ้งเตือนเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งบนอุปกรณ์ IoT ที่แสดงข้อมูลตำแหน่งที่ตั้งของจุดผู้แจ้งเหตุแบบเรียลไทม์บนระบบแผนที่ออนไลน์ เป็นการศึกษาที่มุ่งเน้นเพื่อการพัฒนาระบบเซนเซอร์ในการประยุกต์ใช้เทคโนโลยีที่เป็นกระบวนการในการใช้เซนเซอร์ระบุตำแหน่งและเพื่อวิเคราะห์เส้นทางในการเข้าถึงสำหรับการให้บริการการแพทย์ฉุกเฉิน โดยอาศัยเทคนิควิธีการทางด้าน pgRouting เป็นเทคนิควิธีการค้นหาเส้นทาง เพื่อค้นหาเส้นทางจากตำแหน่งรถฉุกเฉิน รถกู้ชีพ ไปยังตำแหน่งที่รับแจ้ง โดยใช้วิธีการศึกษาข้อมูลจากเอกสารงานวิจัย การสืบค้นข้อมูลต่างๆ และการประยุกต์ใช้เทคโนโลยีและเทคนิควิธีการในการดำเนินงาน โดยมีขั้นตอนและวิธีการดำเนินงาน ดังนี้

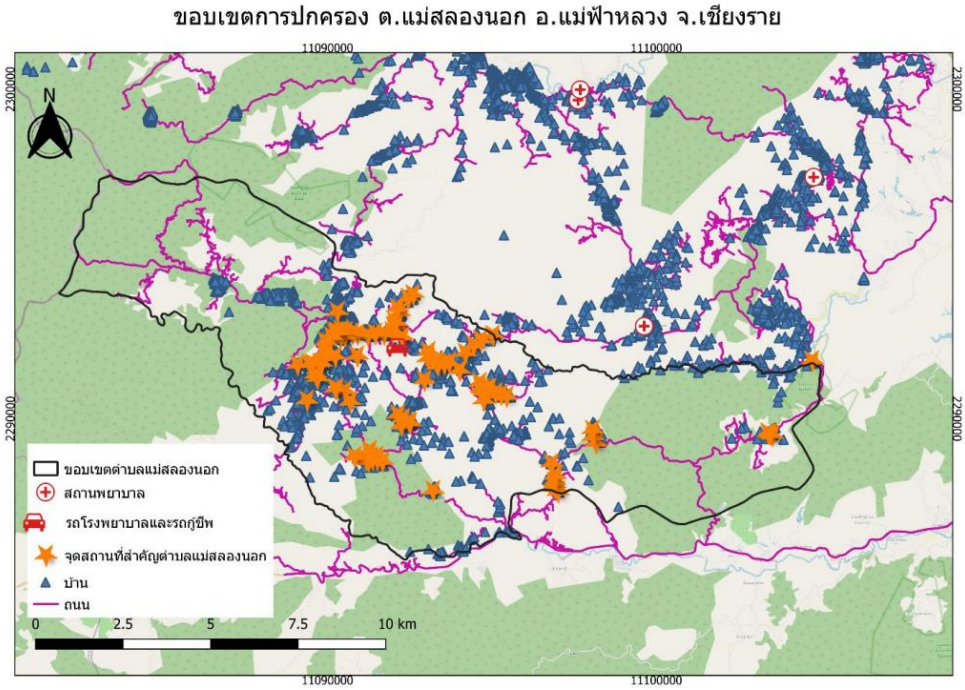
1. เตรียมข้อมูลที่ใช้ในงานวิจัย
2. ศึกษาและวิเคราะห์การหาเส้นทาง
3. พัฒนาอุปกรณ์ IoT
4. พัฒนา Web Map Application

3.1 พื้นที่ศึกษา

พื้นที่วิจัยเป็นการเก็บข้อมูลตำบลแม่สลองใน มีประชากร 25,539 คน จาก 27 หมู่บ้าน และตำบลแม่สลองนอก มีประชากร 15,681 คน จาก 13 หมู่บ้าน ตามลำดับ



ภาพ 17 ขอบเขตการปกครองตำบลแม่สลองใน



ภาพ 18 ขอบเขตการปกครองตำบลแม่สลองนอก

3.2 การลงพื้นที่ภาคสนามเพื่อจัดเก็บข้อมูลและสร้างภูมินาม

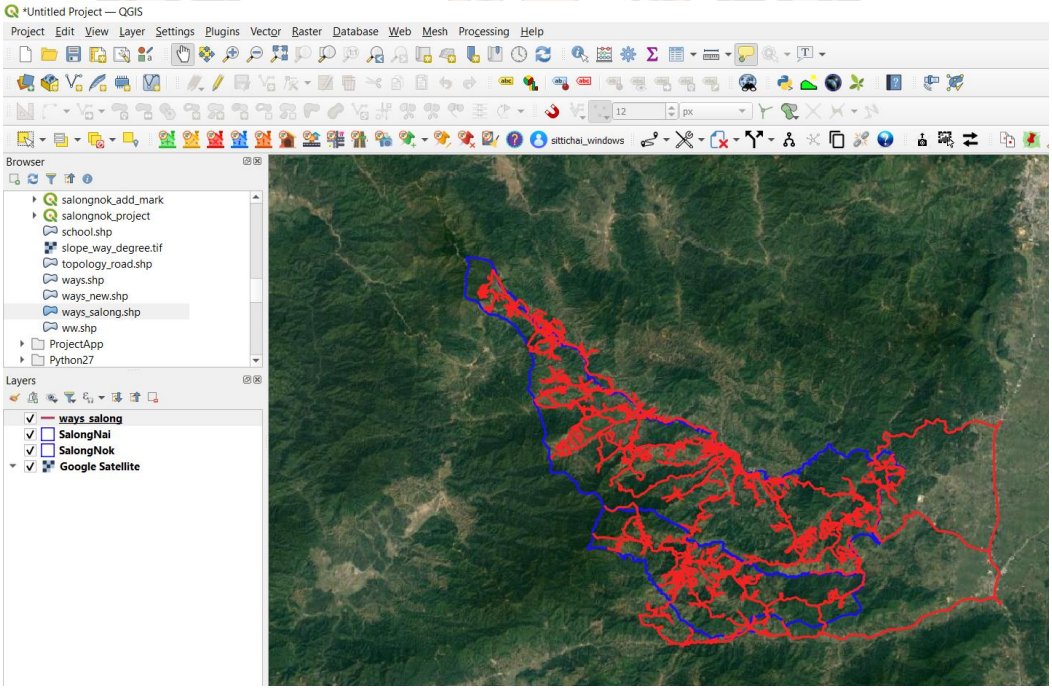
เป็นการดำเนินการเก็บข้อมูลตำแหน่งบ้านและจุดสถานที่สำคัญในองค์การบริหารส่วนตำบลแม่สลองนอกและเทศบาลตำบลแม่สลองใน

ภาพ 19 สอบถามข้อมูลตำแหน่งสำคัญและสร้างภูมินามที่องค์การบริหารส่วนตำบลแม่สลองนอก

ภาพ 20 สอบถามข้อมูลตำแหน่งสำคัญและสร้างภูมินามที่องค์การบริหารส่วนตำบลแม่สลองใน

3.3 การเตรียมข้อมูลด้วยโปรแกรม QGIS

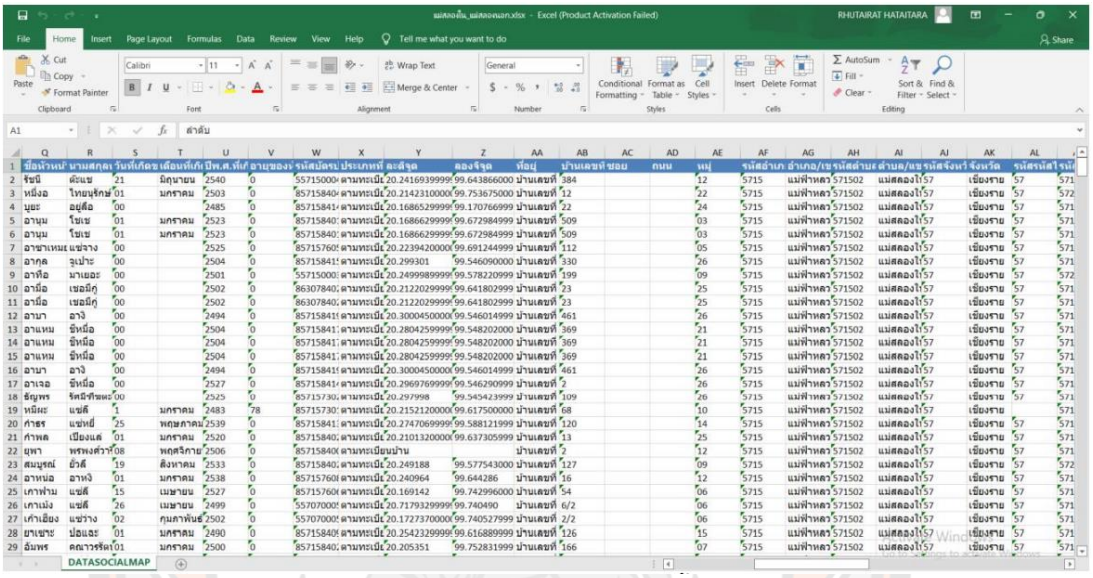
จัดเตรียมข้อมูลโครงข่ายถนนตำบลแม่สลองนอกและตำบลแม่สลองใน โดยที่ตั้งชื่อชั้นข้อมูลว่า ways_salong.shp ในโปรแกรม QGIS



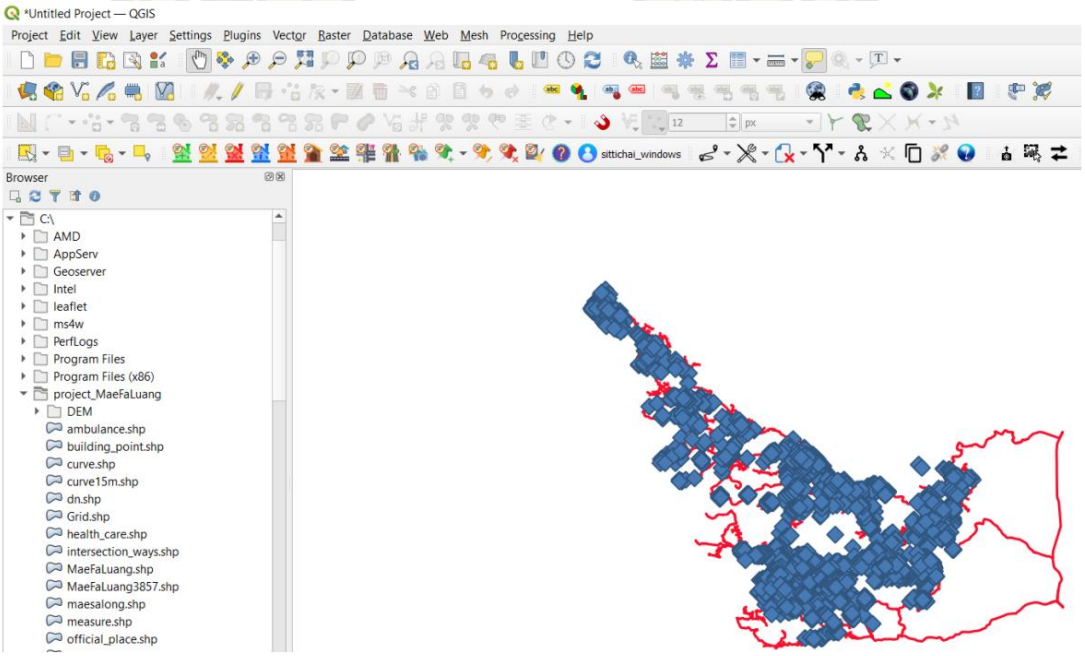
ภาพ 21 ชั้นข้อมูลของถนนตำบลแม่สลองนอกและตำบลแม่สลองใน

3.4 นำเข้าข้อมูลเข้าสู่โปรแกรม QGIS

นำเข้าข้อมูลที่ได้จากการจากการลงพื้นที่ภาคสนามและสอบถามซึ่งจะได้ข้อมูลอยู่ในรูปเป็นของไฟล์ CSV ซึ่งสามารถนำเข้าข้อมูลเข้าสู่โปรแกรม QGIS ได้ เพื่อแสดงจุดพิกัดของตำแหน่งบ้านและจุดสถานที่สำคัญที่ได้ทำการสำรวจซึ่งจะใช้คอลัมน์ละติจูดและลองจิจูด ในการกำหนดพิกัดเป็น X,Y ก็สามารถแสดงตำแหน่งบ้านและจุดสถานที่สำคัญ



ภาพ 22 ภาพไฟล์ CSV ที่ได้จากการลงพื้นที่สอบถาม

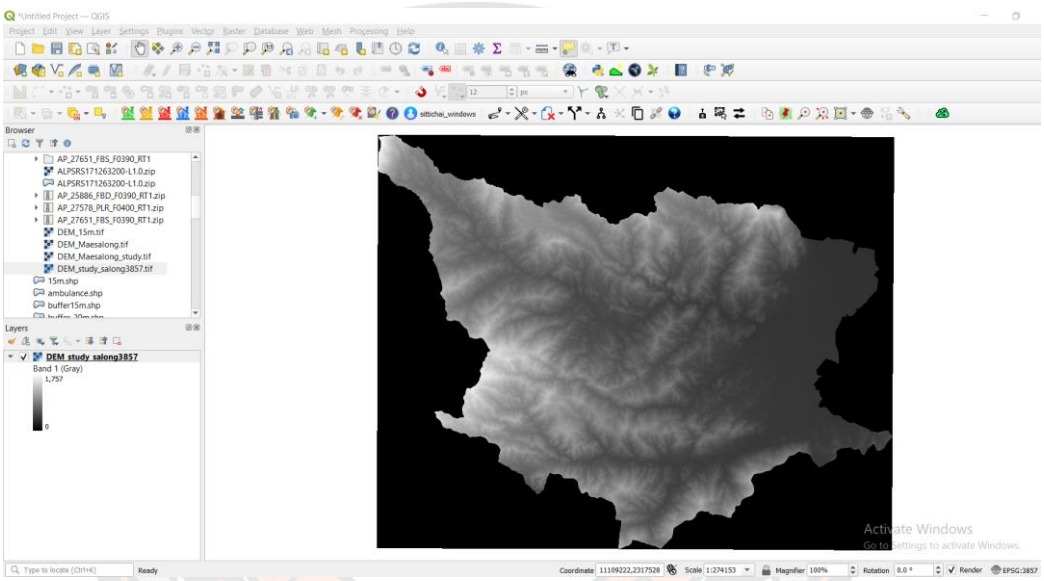


ภาพ 23 ภาพแสดงตำแหน่งบ้านและจุดสถานที่สำคัญ

3.5 การหาค่าความลาดชันของโครงข่ายถนน เพื่อใช้ในการคำนวณหาเส้นทางด้วย pgRouting

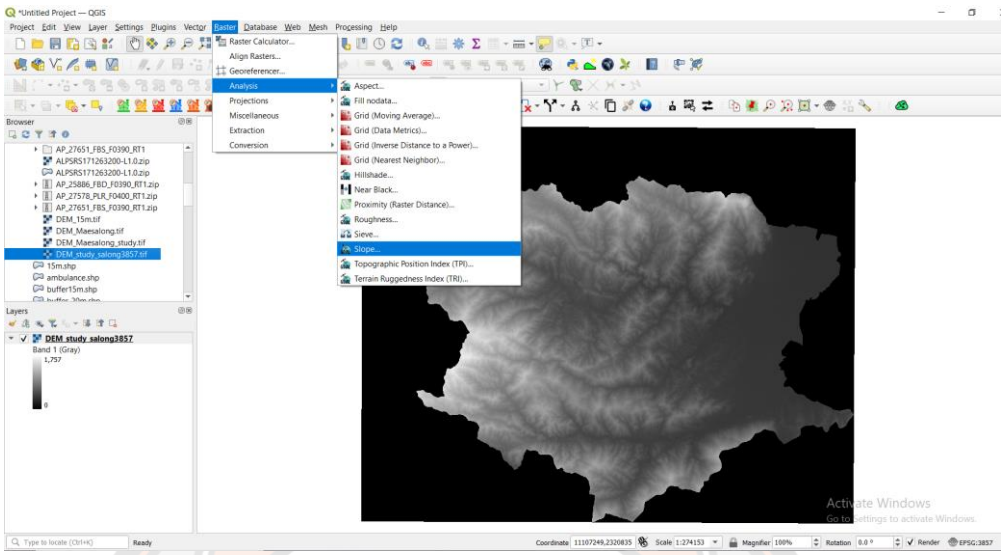
ความลาดชัน (Slope or Slant) ของโครงข่ายถนนมีบทบาทต่อผู้ขับขี่รถโรงพยาบาลและรถกู้ชีพ ผู้ขับขี่จะต้องใช้ในการตัดสินใจในการเลือกเส้นทางไปยังผู้แจ้งเหตุที่ใกล้ที่สุดและคำนึงถึงความปลอดภัยของชีวิตผู้แจ้งเหตุมากที่สุดในการเดินทาง เพื่อใช้ในการคำนวณหาเส้นทางด้วย pgRouting มีขั้นตอนดังนี้

3.5.1 ทำการดาวน์โหลด DEM พื้นที่ของการศึกษา แล้วเปิดโปรแกรม QGIS และนำเข้าข้อมูล DEM_study_salong3857

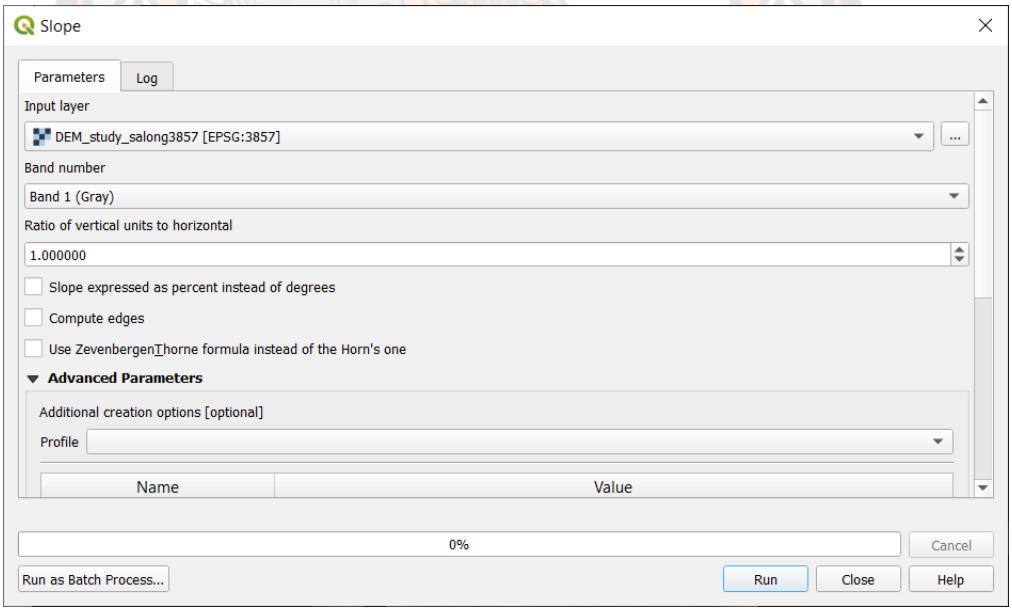


ภาพ 24 นำภาพ DEM เข้าโปรแกรม QGIS

3.5.2 สร้าง Slope คลิกที่ Raster > Analysis > Slope

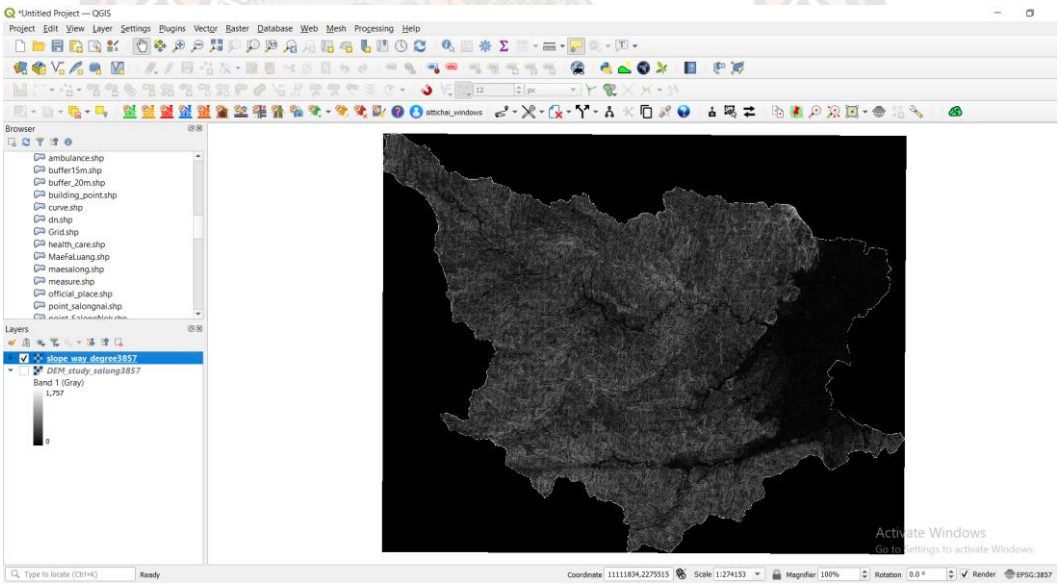


ภาพ 25 การสร้าง Slope ขั้นตอนที่ 1



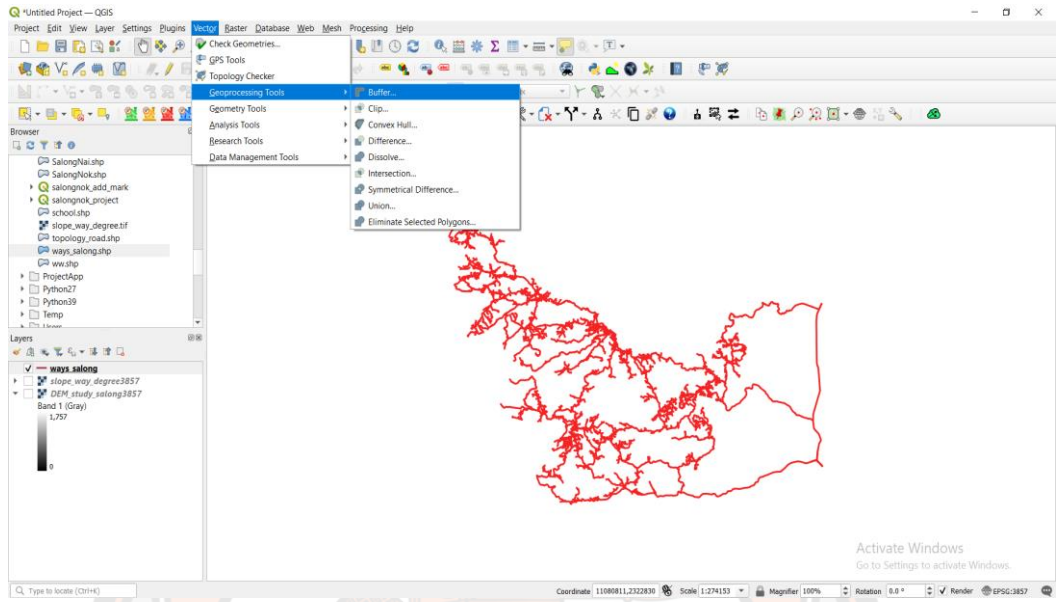
ภาพ 26 การสร้าง Slope ขั้นตอนที่ 2

ภาพ 27 การสร้าง Slope ขั้นตอนที่ 3

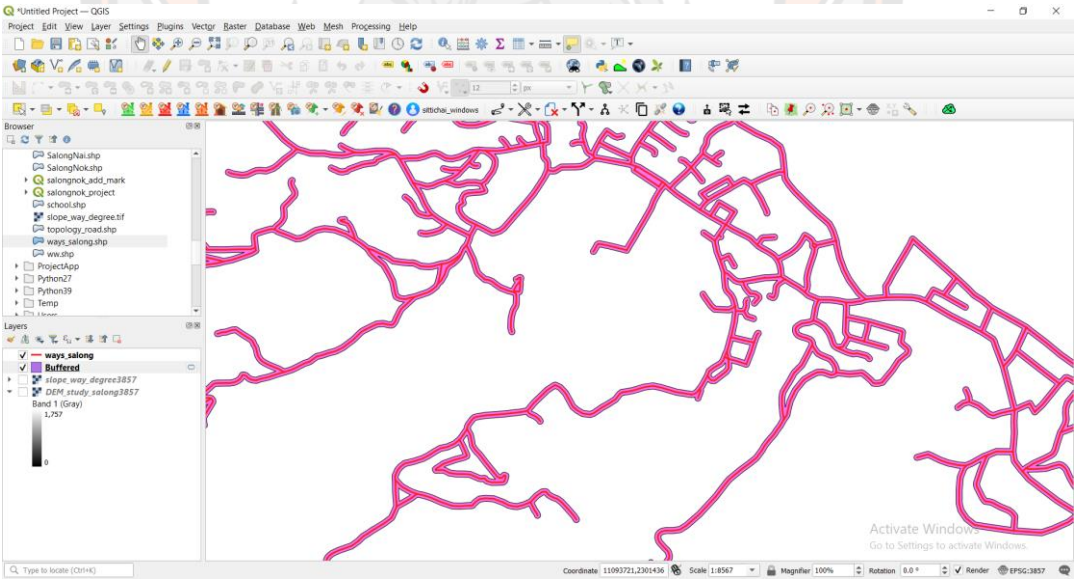


ภาพ 28 ผลของการทำ Slope ของโครงข่ายถนน

3.5.3 ทำ Buffer ด้วยการนำชั้นข้อมูล ways_salong เข้ามาในโปรแกรม QGIS จากนั้นคลิกที่ Vector > Geoprocessing Tools > Buffer

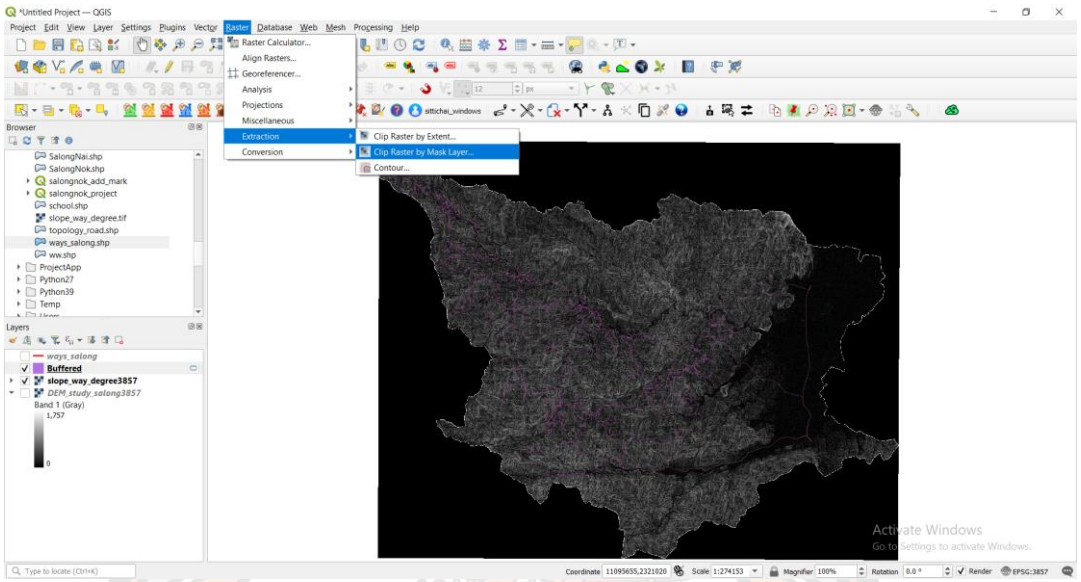


ภาพ 29 การทำ Buffer ขั้นตอนที่ 1

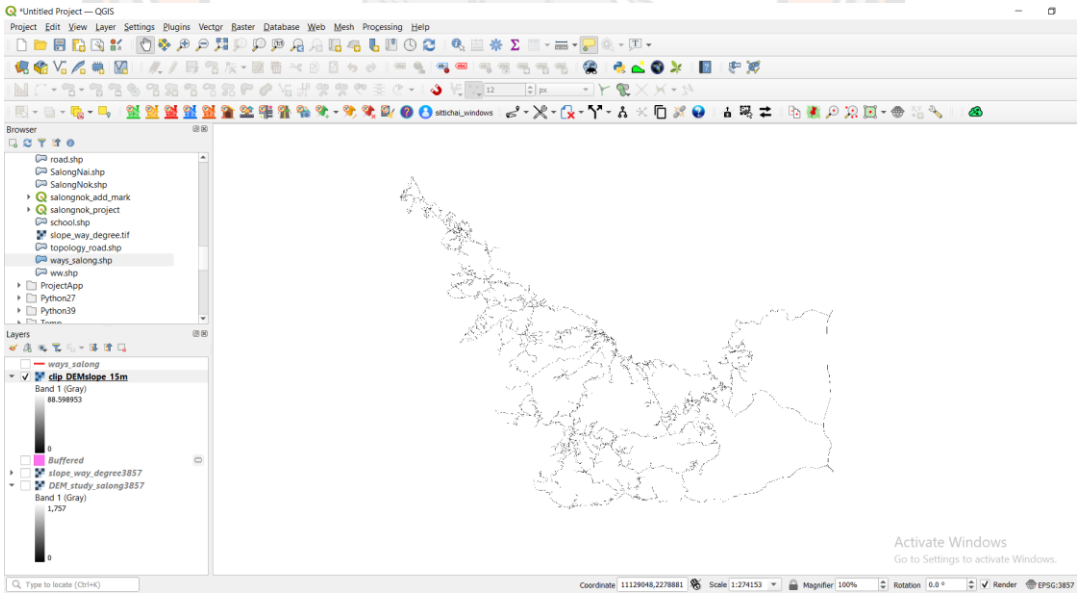


ภาพ 30 ผลของการทำ Buffer โครงข่ายถนน

3.5.4 ทำการ Clip Raster by Mask Layer ใช้ข้อมูล Buffer และ slope_way_degree3857

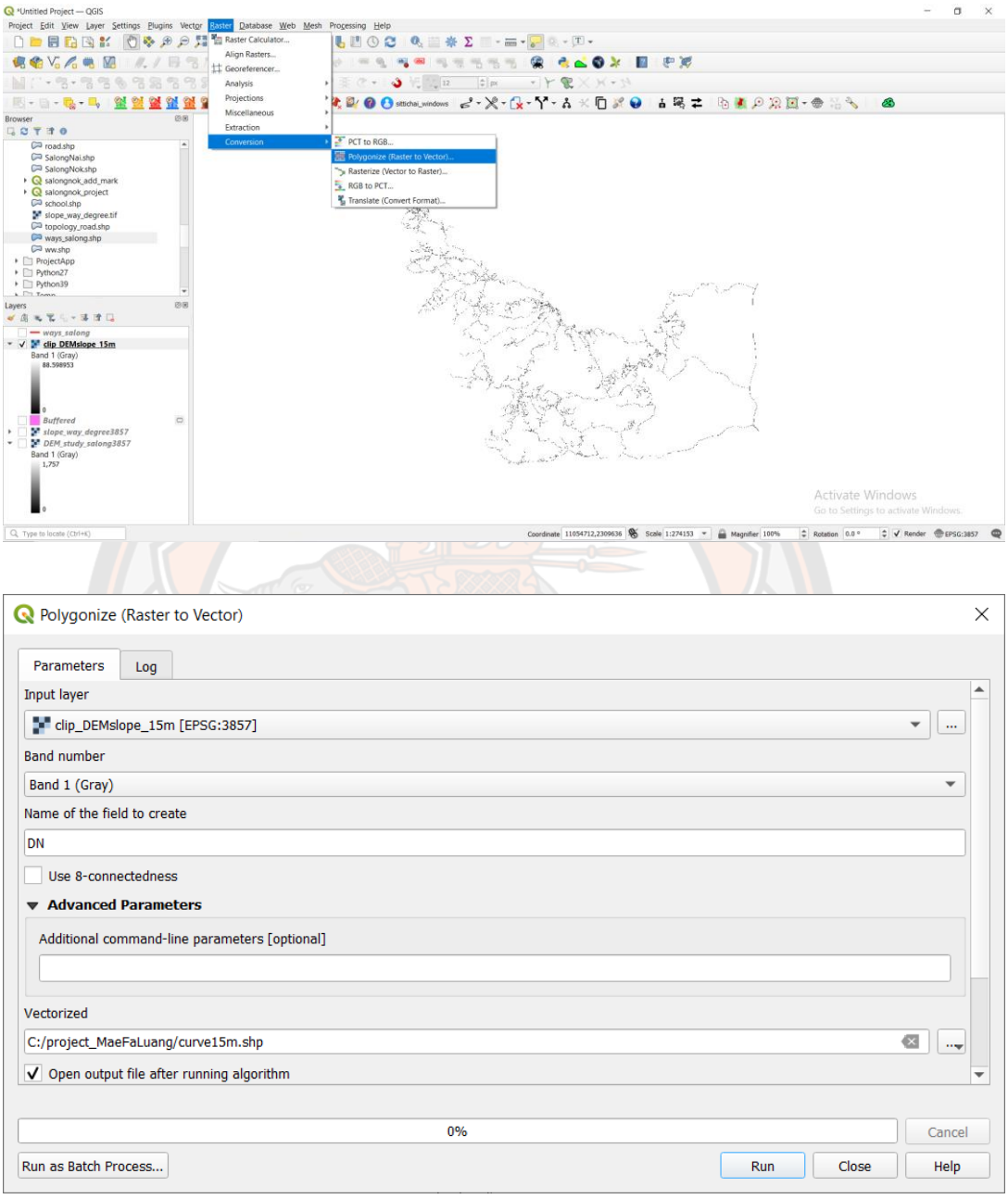


ภาพ 31 การทำ Clip Raster by Mask Layer

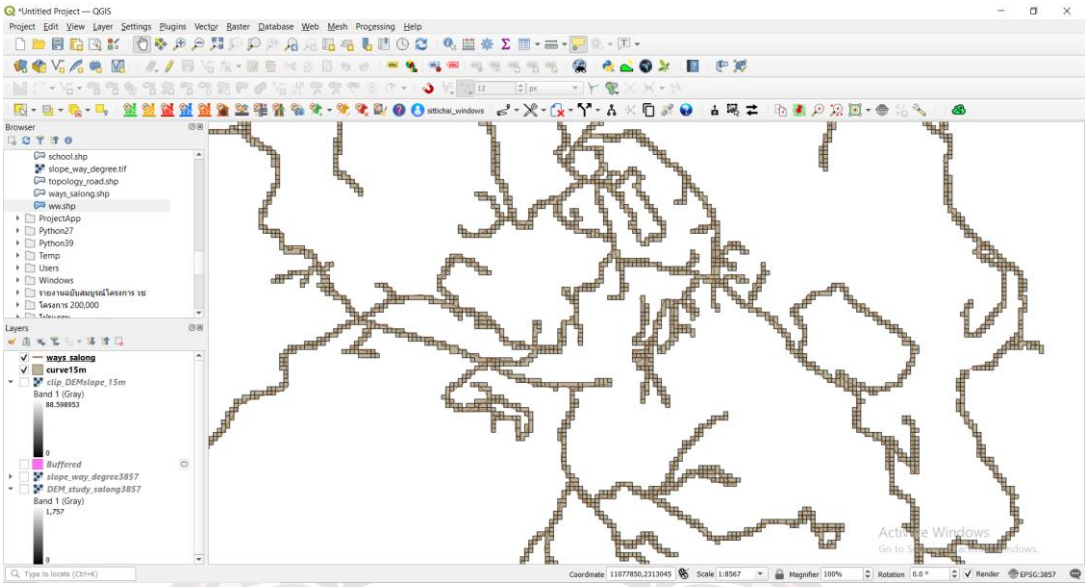


ภาพ 32 ผลลัพธ์การทำ Clip Raster by Mask Layer

3.5.5 ทำ Polygonize (Raster to Vector) ด้วยการคลิกที่ Raster > Conversion > Polygonize โดยที่ชื่อชั้นข้อมูลว่า curve15m.shp

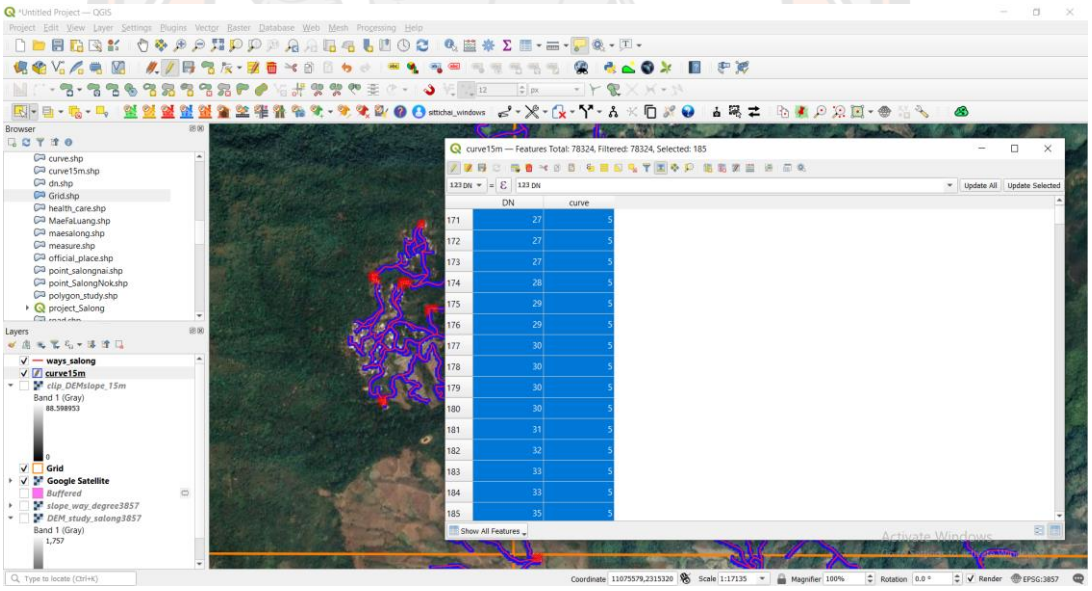


ภาพ 33 ขั้นตอนการทำ Polygonize



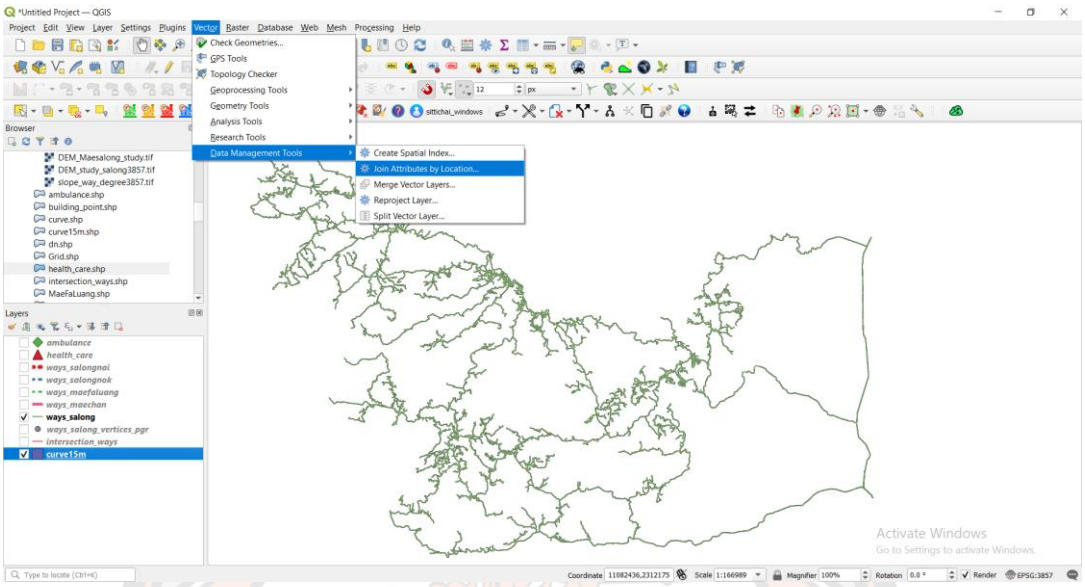
ภาพ 34 ผลลัพธ์การทำ Polygonize

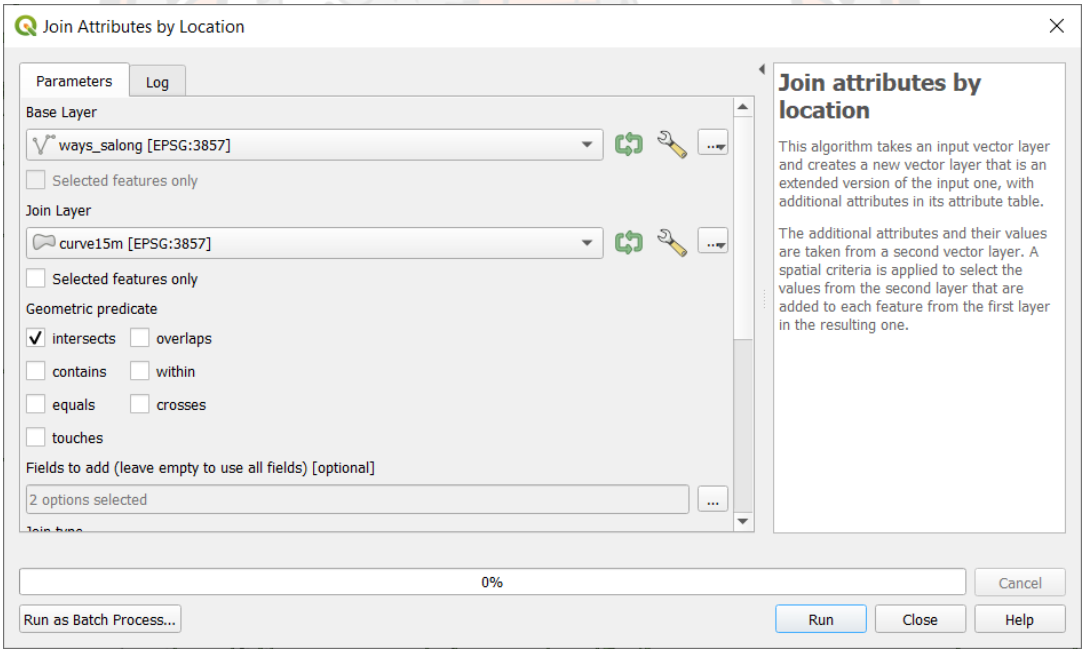
3.5.6 สร้างชื่อตารางขึ้นมาใหม่ว่า curve ในชั้นข้อมูล curve15m.shp และใส่ค่าระดับความลาดชันมากที่สุดเป็น 5

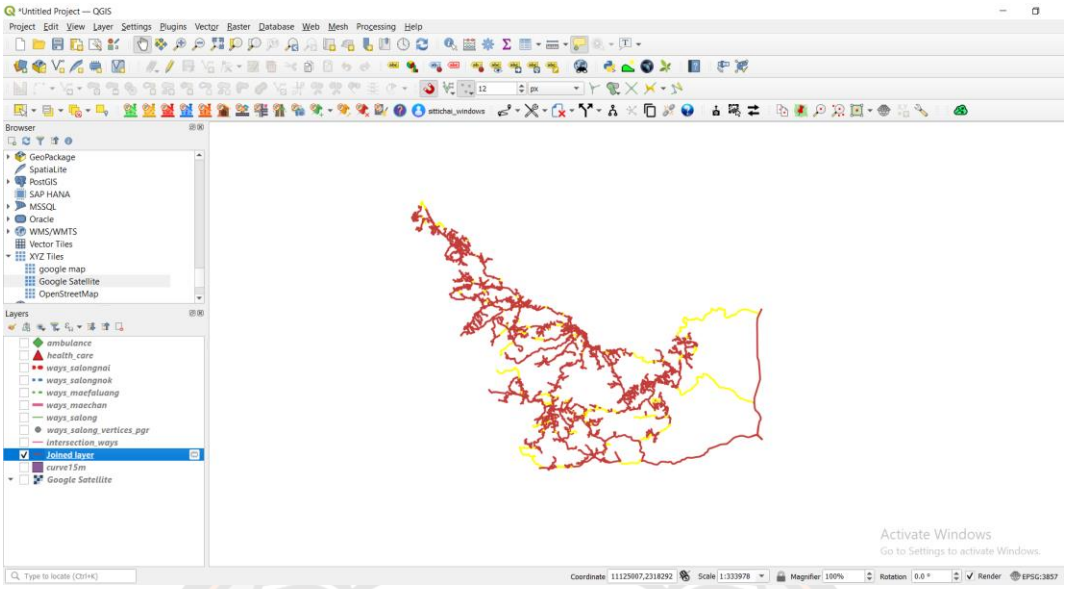


ภาพ 35 การสร้างตาราง curve

3.5.7 ดำเนินการ Join Attributes by Lacation ข้อมูล ways_salong กับชั้นข้อมูล curve15m







ภาพ 36 การ Join Attributes by Lacation

3.5.8 ผลของการ Join Attributes by Lacation คอลัมน์ที่ชื่อ curve คือ การกำหนดค่าของ Slope ซึ่งมีการกำหนดค่า Slope ดังนี้ ค่า Slope 5% มีความลาดชันสูงสุด ค่า Slope 4% มีความลาดชัน สูง ค่า Slope 3% มีความลาดชันสูงปานกลาง Slope 2% มีความลาดชันต่ำ และ Slope 1% มีความลาดชันต่ำสุด จากนั้นผู้วิจัยนำคอลัมน์ curve ไปใส่ในสมการ s=vt + curve เพื่อนำไปใช้ในการคำนวณเส้นทางด้วย pgRouting ในขั้นตอนต่อไป

Joined layer — Features Total: 4079, Filtered: 4079, Selected: 1

	gid	id	source	target	reverse_cost	length	DN	curve
1	19	NULL	102	104	979.633618828...	979.633618828...	22	5
2	57	NULL	124	125	18.7085983902...	18.7085983902...	12	5
3	58	NULL	125	126	28.2109971787...	28.2109971787...	8	5
4	59	NULL	125	127	51.481073052552	51.4810730525...	11	5
5	108	NULL	171	176	817.570316773...	817.570316773...	25	5
6	109	NULL	176	177	1073.61407170...	1073.61407170...	28	5
7	146	NULL	211	63	98.247914593296	98.247914593296	20	5
8	150	NULL	97	212	987.2331938073	987.2331938073	15	5
9	269	NULL	327	328	188.000014831...	188.000014831...	7	5
10	270	NULL	328	329	23.3845791494...	23.3845791494...	6	5
11	271	NULL	329	330	156.978349789...	156.978349789...	9	5
12	279	NULL	337	329	267.711959927...	267.711959927...	6	5
13	292	NULL	347	348	326.525048861...	326.525048861...	12	5
14	295	NULL	349	353	4032.15086559...	4032.15086559...	21	5
15	319	NULL	375	376	177.712925749...	177.712925749...	22	5
16	409	NULL	450	451	117.090162687...	117.090162687...	20	5

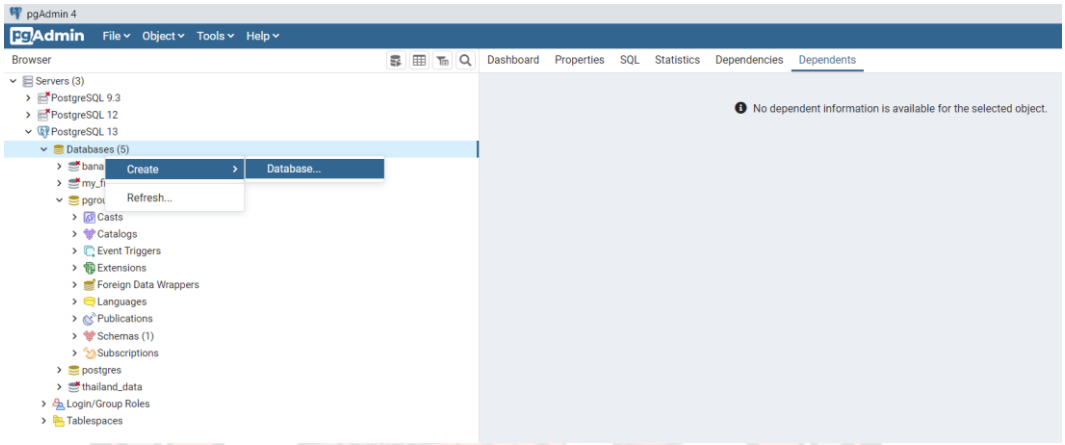
Show All Features

ภาพ 37 ผลการของ Join Attributes by Lacation

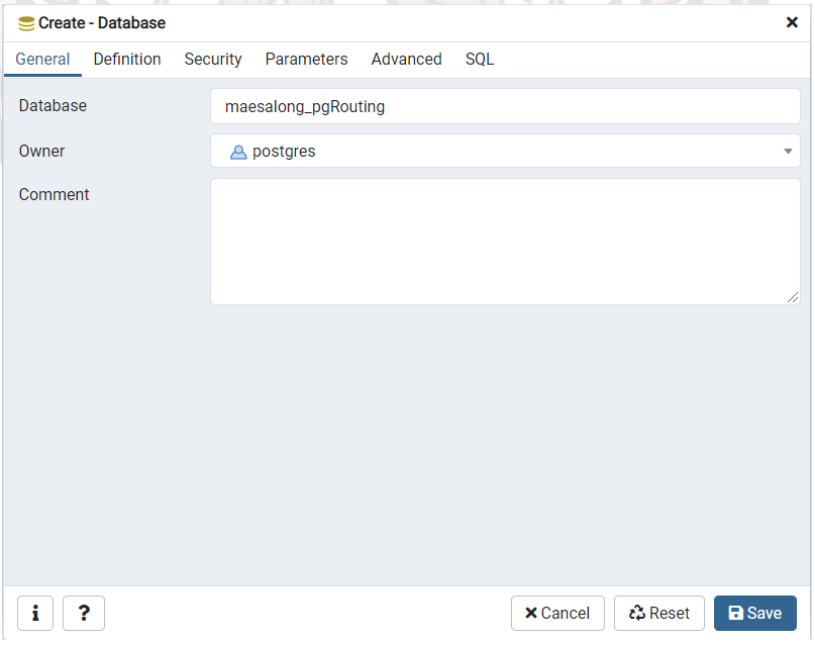
3.6 การจัดเตรียมฐานข้อมูลโครงข่ายถนน

ในการจัดเตรียมชั้นข้อมูล SQL ไปยังฐานข้อมูล เริ่มต้นด้วยการสร้างฐานข้อมูล จากนั้นจึงเพิ่มฟังก์ชัน PostGIS และเพิ่มฟังก์ชันการกำหนดเส้นทาง โดยเพิ่ม pgRouting ลงในฐานข้อมูล มีขั้นตอนดังนี้

3.6.1 เปิดโปรแกรม pgAdmin เพื่อสร้างฐานข้อมูล

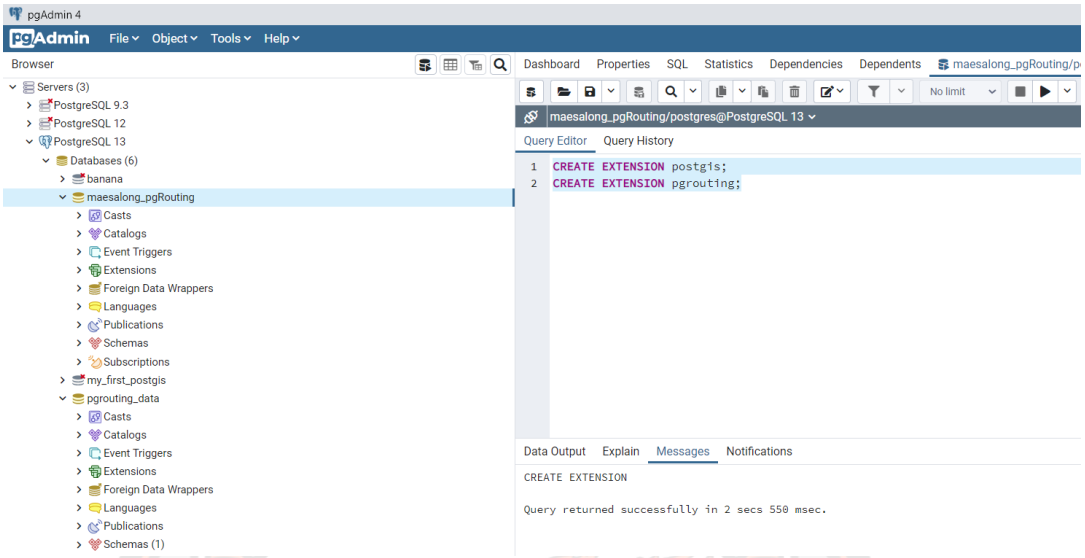


ภาพ 38 การจัดเตรียมฐานข้อมูลโครงข่ายถนน ครั้งที่ 1



ภาพ 39 การจัดเตรียมฐานข้อมูลโครงข่ายถนน ครั้งที่ 2

3.6.2 ใช้คำสั่ง CREATE EXTENSION postgis; และคำสั่ง CREATE EXTENSION pgrouting;



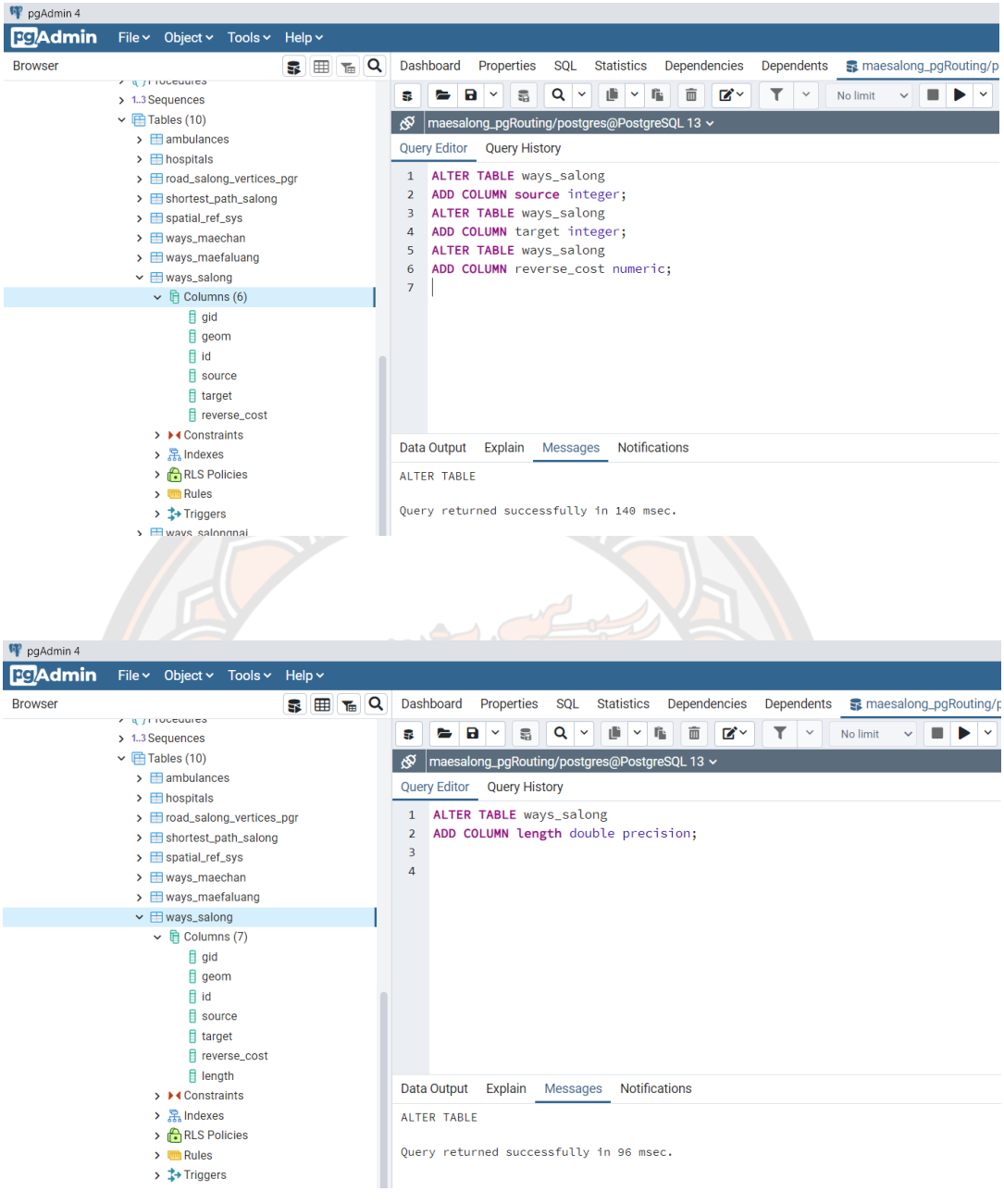
ภาพ 40 การจัดเตรียมฐานข้อมูลโครงข่ายถนน ครั้งที่ 3

3.7 การจัดเตรียมข้อมูลโครงข่ายถนนในโปรแกรม pgAdmin เพื่อใช้ในการคำนวณหาเส้นทางด้วย pgRouting

ก่อนที่จะใช้ชั้นข้อมูลโครงข่ายถนนสำหรับบริการการกำหนดเส้นทาง จำเป็นต้องเพิ่มคอลัมน์ของข้อมูลถนนและสร้าง Topology ของโครงข่ายถนนที่สามารถแยกและจับโหนดทั้งหมดของข้อมูลเพื่อแยกส่วนของโครงข่ายถนน คำสั่งต่อไปนี้ใช้เพื่อเพิ่มคอลัมของข้อมูลถนนและสร้าง Topology

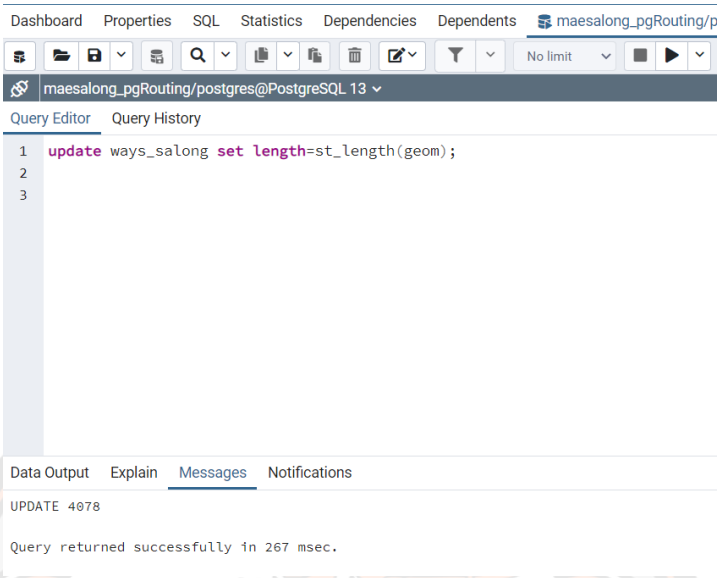
3.7.1 การเพิ่มคอลัมน์ source,target และ reverse_cost เข้าไปในเทเบิล ways_salong โดยใช้คำสั่ง

```
ALTER TABLE ways_salong
ADD COLUMN source integer;
ALTER TABLE ways_salong
ADD COLUMN target integer;
ALTER TABLE ways_salong
ADD COLUMN reverse_cost numeric;
ALTER TABLE ways_salong
ADD COLUMN length double precision;
```



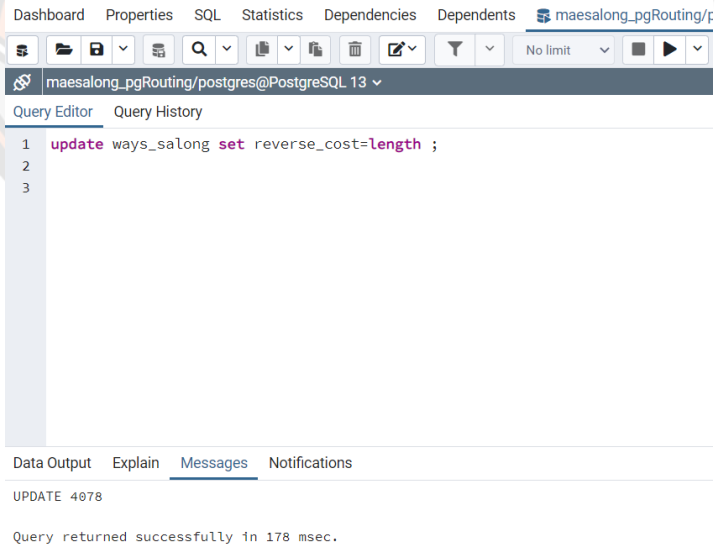
ภาพ 41 คำสั่งการเพิ่มคอลัมน์ เพื่อใช้ในการคำนวณหาระยะทาง

3.7.2 การคำนวณระยะทางสำหรับคอลัม length โดยใช้คำสั่ง update ways_salong set length=st_length(geom);



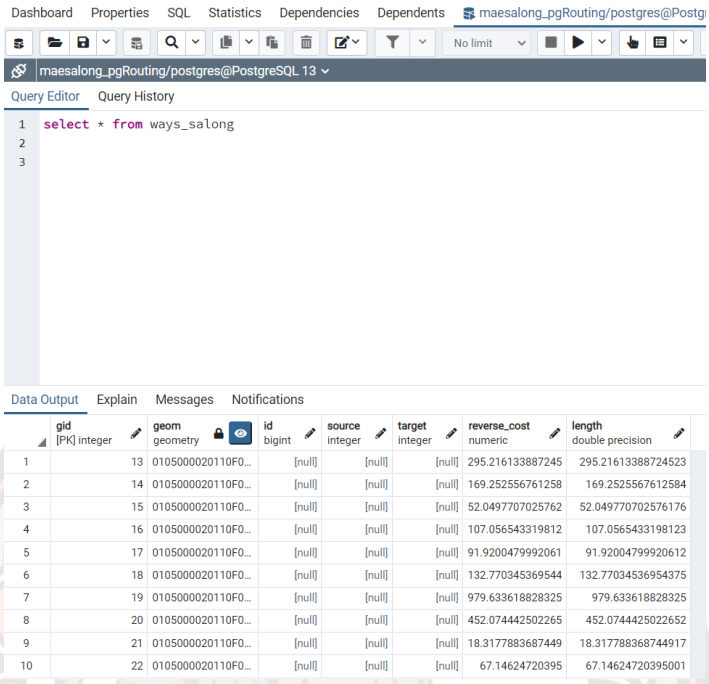
ภาพ 42 คำสั่งการคำนวณระยะทาง

3.7.3 การตั้งค่า reverse_cost ให้เท่ากับระยะทาง เข้าไปในชั้นข้อมูล ways_salong โดยใช้คำสั่ง update ways_salong set reverse_cost=length;



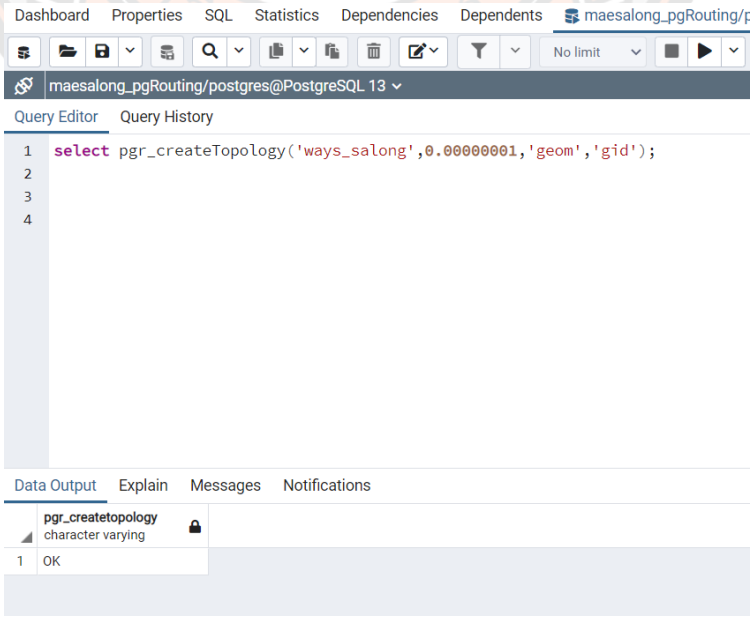
ภาพ 43 คำสั่งการตั้งค่า reverse_cost ให้เท่ากับระยะทาง

3.7.4 การแสดงรายละเอียดหัวตารางและค่าดังรูป โดยใช้คำสั่ง select * from ways_salong



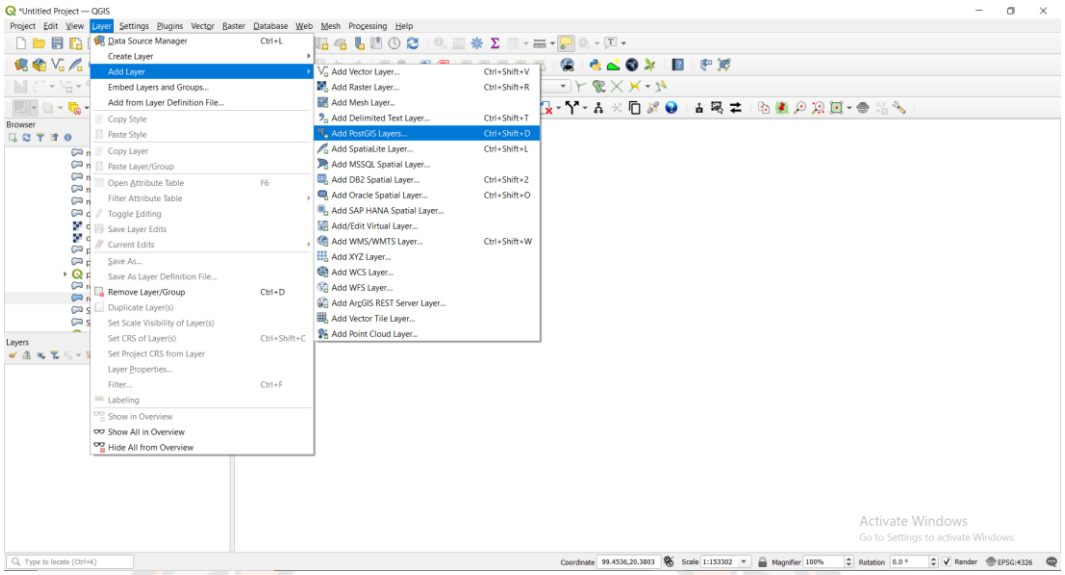
ภาพ 44 คำสั่งแสดงข้อมูลของตาราง way_salong

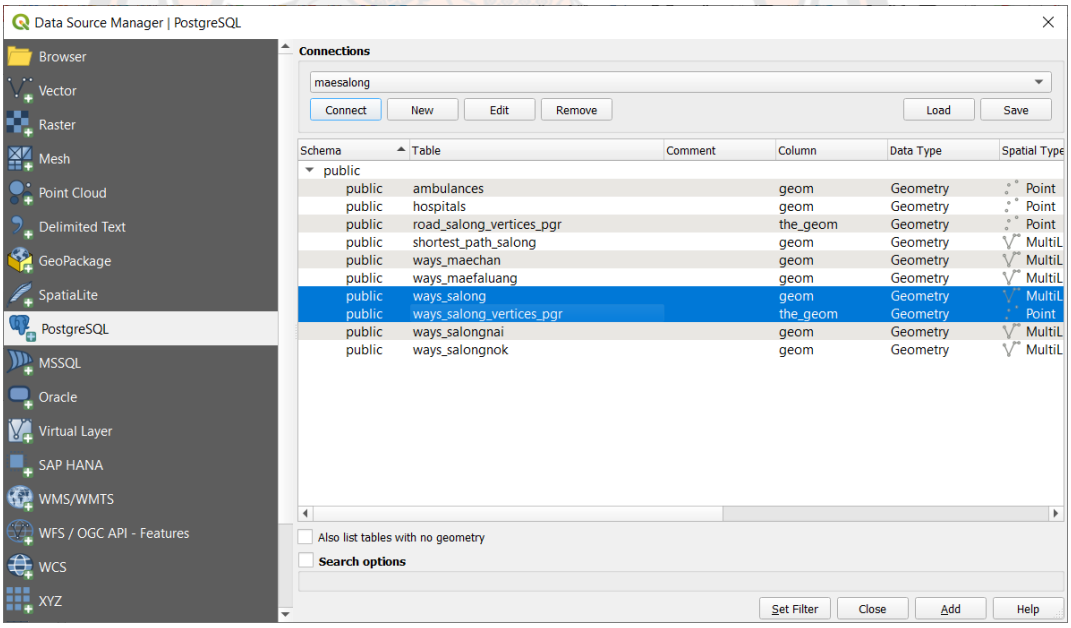
3.7.5 การคำนวณหาค่า topology สำหรับเพิ่มค่า source,target โดยใช้คำสั่ง select pgr_createTopology(‘ways_salong',0.0000001,'geom','gid');



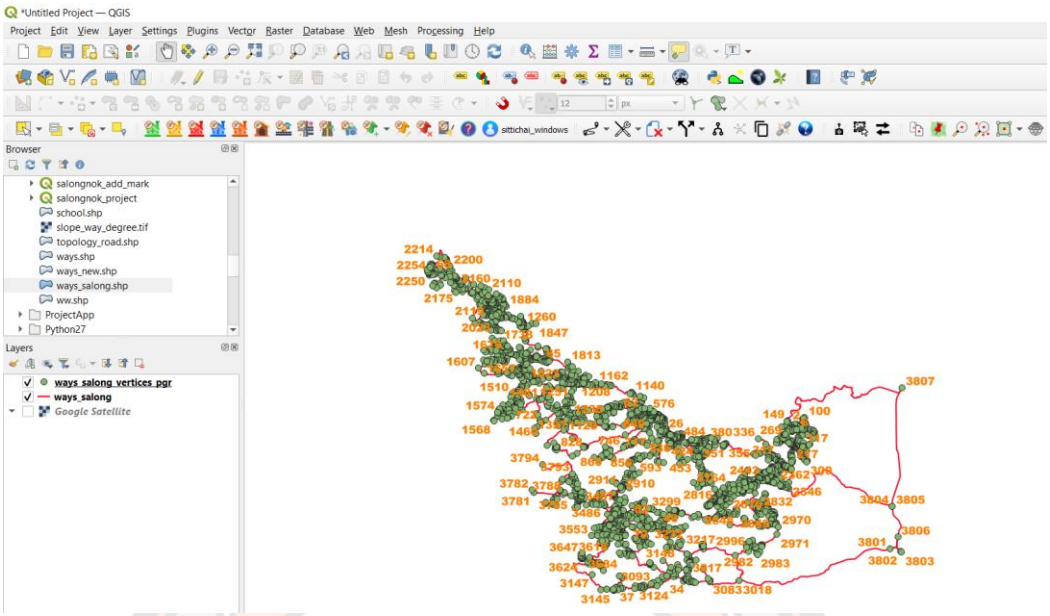
ภาพ 45 คำสั่งคำนวณหาค่า topology สำหรับเพิ่มค่า source,target

3.7.6 กลับไปที่ QGIS แล้วทำการ ADD ชั้นข้อมูลจาก pgAdmin มาใส่ในโปรแกรม QGIS





ภาพ 46 ขั้นตอนการ ADD ชั้นข้อมูลจาก pgAdmin มาใส่ในโปรแกรม QGIS



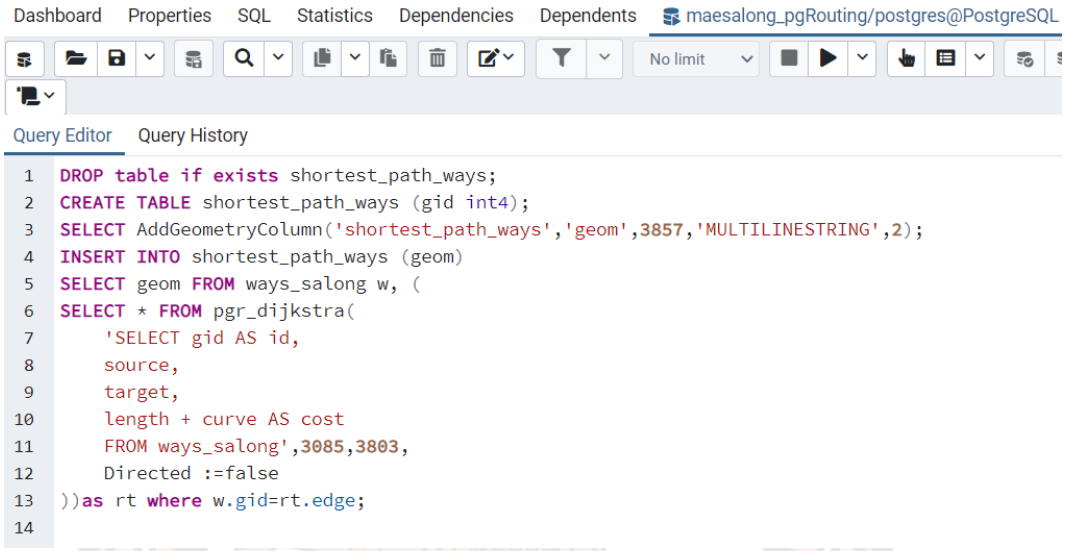
ภาพ 47 ข้อมูลที่ได้จาก pgAdmin ในโปรแกรม QGIS

3.8 การใช้คำสั่ง SQL ค้นหาเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgRouting

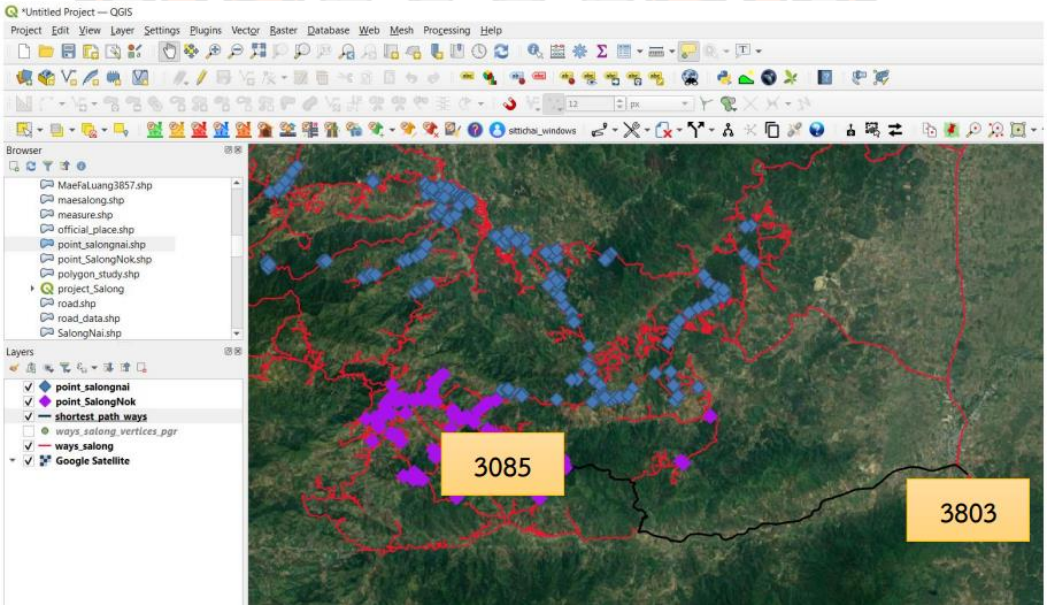
ชุดคำสั่ง Dijkstra Algorithm เป็นชุดคำสั่งตัวแรกของ pgRouting ซึ่งใช้เพื่อค้นหาระยะทางที่สั้นที่สุดจากจุด A ไปยังจุด B สำหรับชุดคำสั่งนี้เราต้องการจุดเริ่มต้น (source) และจุดสิ้นสุด (target) ในการหาระยะทางที่สั้นที่สุด ซึ่งจะคำนวณจากคอลัมน์ length as cost โดยใช้คำสั่ง SQL ค้นหาเส้นทางที่สั้นที่สุดด้วยฟังก์ชันpgRouting โดยการใช้คำสั่ง ดังนี้

```
DROP table if exists shortest_path_ways;
CREATE TABLE shortest_path_ways (gid int4);
SELECT AddGeometryColumn('shortest_path_ways','geom',3857,'MULTILINESTRING',2);
INSERT INTO shortest_path_ways (geom)
SELECT geom FROM ways_salong w, (
SELECT * FROM pgr_dijkstra(
        'SELECT gid AS id,
        source,
        target,
        length + curve AS cost
        FROM ways_salong',3085,3803,
        Directed :=false
))as rt where w.gid=rt.edge;
```

คำสั่ง SQL ถูกสร้างขึ้นในฐานข้อมูลด้วยฟังก์ชัน pgRouting ดังแสดงในภาพที่ 48 จากนั้นเพิ่มชั้นข้อมูลที่ชื่อ shortest_path_ways จากฐานข้อมูลเข้ามาในโปรแกรม QGIS ซึ่งจะไปปรากฏอยู่ในฐานข้อมูล maesalong ที่เราสร้างขึ้น ผลลัพธ์จะปรากฏระยะทางที่สั้นที่สุดจากจุดที่ 3085 ไปยังจุดที่ 3803 (ระยะทางที่สั้นที่สุด คือ เส้นสีดำ) ดังแสดงในภาพที่ 49



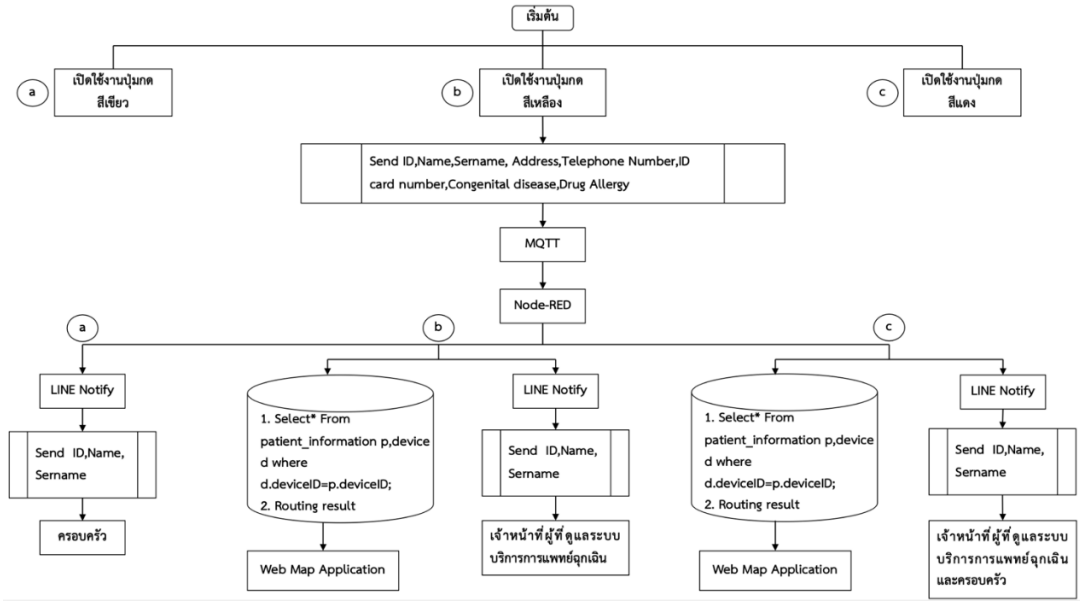
ภาพ 48 การใช้คำสั่ง SQL ค้นหาเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgRouting



ภาพ 49 ภาพผลลัพธ์การใช้คำสั่ง SQL ค้นหาเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgRouting ที่แสดงโปรแกรม QGIS

3.9 การออกแบบการทำงานของระบบ

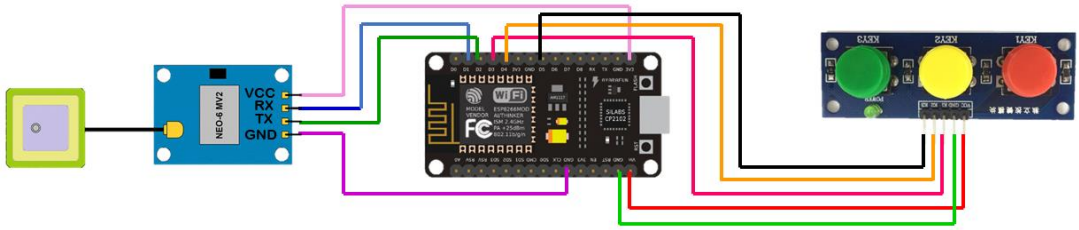
การพัฒนาระบบแจ้งเหตุฉุกเฉินอัจฉริยะบน IoT (SENS-IoT) พร้อมเซนเซอร์ระบุตำแหน่งเพื่อเข้าถึงบริการการแพทย์ฉุกเฉินด้วยโมดูล GPS และ Web GIS การพัฒนาและการทำงานของระบบมีรูปแบบดังนี้ (a) เมื่อผู้ใช้ประสบอุบัติเหตุและเปิดใช้งานปุ่มกดสีเขียว อุปกรณ์ SENS-IoT จะแจ้งเตือนไปยังระบบเมื่อเกิดเหตุและส่งการแจ้งเตือนผ่าน Line Notify ไปยังโทรศัพท์มือถือของครอบครัวทันที (b) เมื่อผู้ใช้เกิดอุบัติเหตุฉุกเฉินแล้วทำการเปิดใช้งานปุ่มกดสีเหลืองอุปกรณ์ SENS-IoT แจ้งเหตุเข้ามาในระบบและข้อความจะแจ้งเตือนผ่าน Line Notify เข้าสู่โทรศัพท์มือถือของเจ้าหน้าที่ผู้ที่ดูแลระบบบริการการแพทย์ฉุกเฉินทันที เมื่อระบบได้รับละติจูด ลองจิจูด และรายละเอียดผู้ใช้แล้ว ระบบจะค้นหาเส้นทางที่พัฒนาด้วย pgRouting และคำนวณเส้นทางจากรถพยาบาลหรือรถกู้ภัยฉุกเฉินที่ใกล้ที่สุด ไปที่จุดแจ้งเตือนโดยเร็วที่สุด ข้อมูลจะแสดงบน Web Map Application และ (c) เมื่อผู้ใช้ประสบอุบัติเหตุและเปิดใช้งานปุ่มสีแดง อุปกรณ์ SENS-IoT จะแจ้งเหตุให้ระบบทราบและจะส่งข้อความทาง Line Notify ไปยังโทรศัพท์มือถือของบุคลากรที่รับผิดชอบจัดการระบบบริการการแพทย์ฉุกเฉินและสมาชิกในครอบครัวทันที และเมื่อระบบได้รับละติจูด ลองจิจูด และรายละเอียดผู้ใช้ ระบบจะค้นหาเส้นทางที่พัฒนาด้วย pgRouting และคำนวณเส้นทางจากรถพยาบาลหรือรถกู้ภัยฉุกเฉินที่ใกล้ที่สุด ไปที่จุดแจ้งเหตุฉุกเฉินโดยเร็วที่สุด ข้อมูลจะแสดงบน Web Map Application ดังผังงานในภาพ 50 ด้านล่าง



ภาพ 50 ภาพรวมของระบบ SENS-IoT กับ Web Map Application

3.10 การออกแบบการทำงานของเซนเซอร์ Ublox NEO-6M GPS

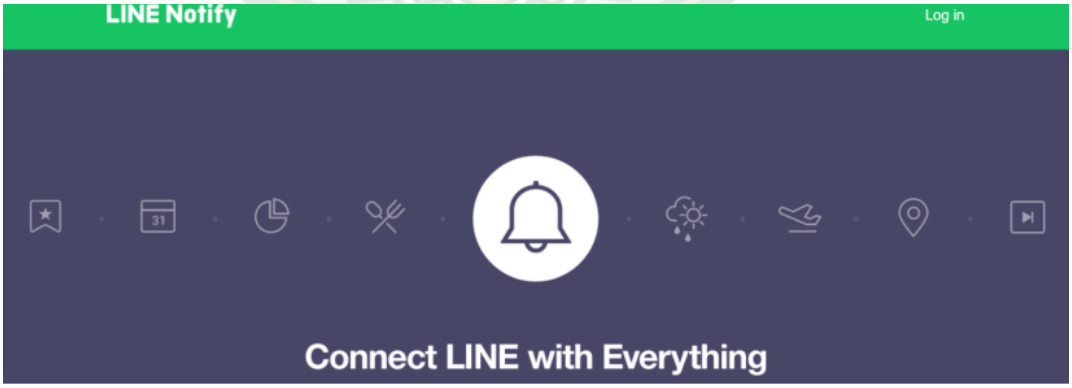
การพัฒนาระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งบน IoT ที่แสดงข้อมูลตำแหน่งที่ตั้งของผู้แจ้งเหตุแบบเรียลไทม์ ได้ออกแบบการทำงานของเซนเซอร์ที่เป็นในส่วนของจีพีเอส โดยมีรายละเอียดของการออกแบบการทำงานและวิเคราะห์ระบบ ดังนี้ การออกแบบการทำงานของเซนเซอร์ Ublox NEO-6M GPSขั้นตอนแรกในการจัดการกับการทำงานของเซนเซอร์ คือ โปรแกรม Arduino IDE ซึ่งเป็นโปรแกรมสำหรับเขียนภาษา C เพื่อกำหนดคำสั่งเงื่อนไขการทำงานให้กับเซนเซอร์ด้วยการอัปโหลดคำสั่งที่เขียนไปยังตัวเซนเซอร์ (NodeMCU ESP8266 + Ublox NEO-6M GPS) ที่เชื่อมต่อกับคอมพิวเตอร์ โดยทำการเชื่อมต่อระหว่างบอร์ด NodeMCU ESP8266 และ Ublox NEO-6M GPS ดังภาพ 51



ภาพ 51 การต่อวงจรระหว่างบอร์ด NodeMCU ESP8266 และ Ublox NEO-6M GPS

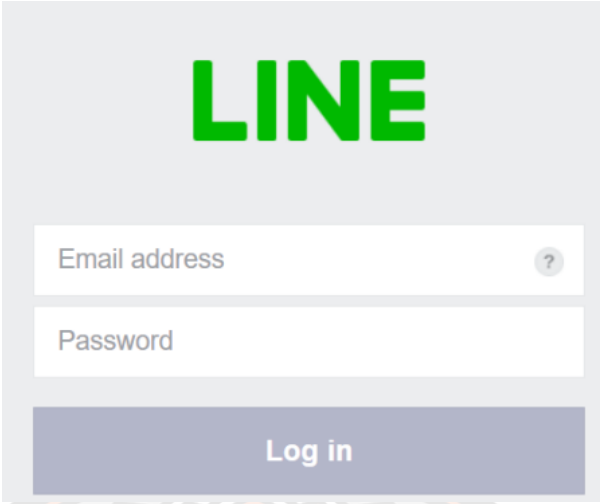
3.11 การพัฒนาระบบแจ้งเตือนผ่าน LINE Notify

ก่อนที่จะทำการเชื่อมโค้ดของเซนเซอร์ Ublox NEO-6M GPS กับ LINE Notify ต้องทำการสมัครใช้งานกับ LINE Notify เพื่อรับ Token ที่จะนำมาใช้งาน ซึ่ง 1 Token สามารถส่งข้อมูลไปยังผู้ใช้งานหรือกลุ่มต่างๆที่ผู้ใช้งานอยู่ได้ ซึ่ง LINE Notify สามารถดึงข้อมูลจากฐานข้อมูลได้โดยมีการตอบสนองต่อกับเซนเซอร์ร่วมกับผู้ใช้งาน

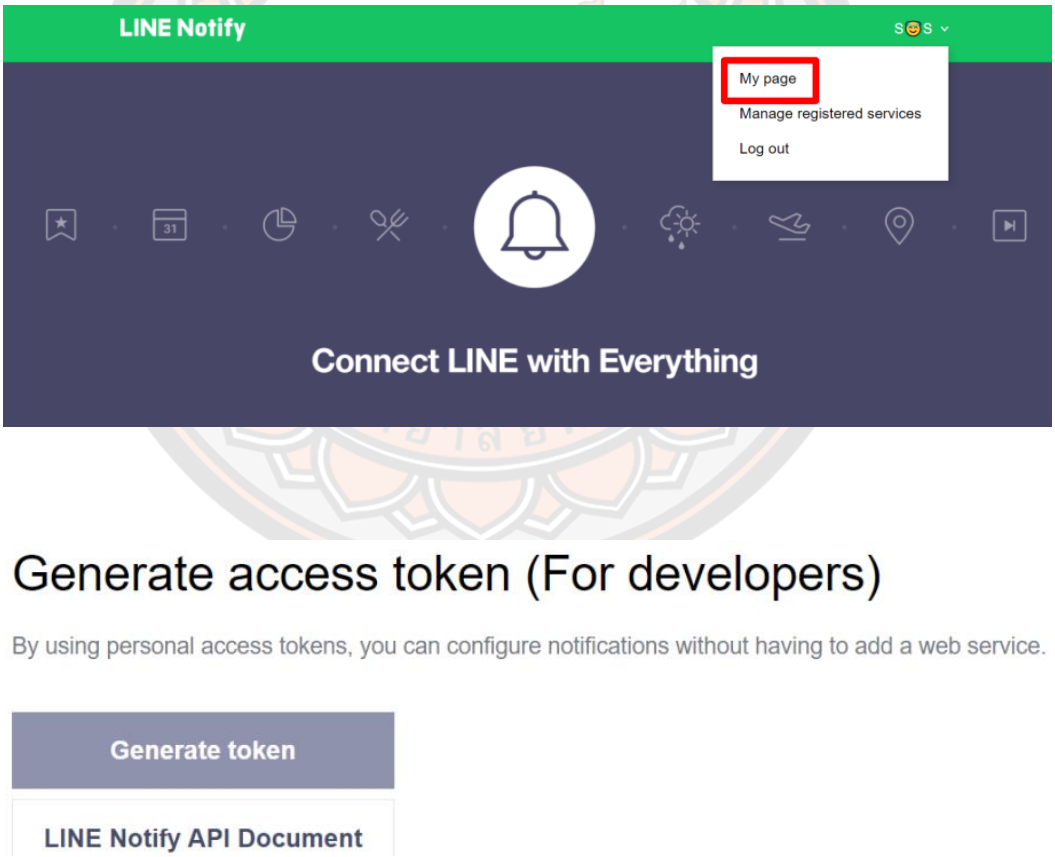


ภาพ 52 หน้าต่างเพจ LINE Notify สำหรับการสมัครใช้บริการ LINE Notify

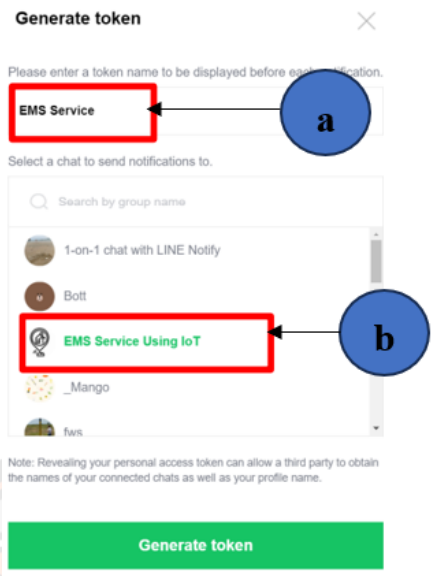
(สามารถเข้าไปที่: https://notify-bot.line.me/en/)



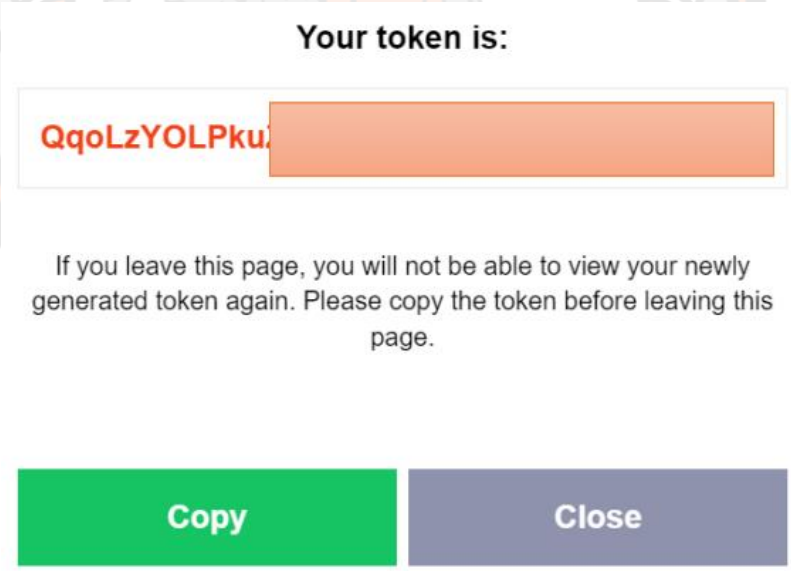
ภาพ 53 ลงชื่อเข้าใช้งานบนหน้าเว็บโดยใส่ข้อมูลไลน์ที่เราต้องการส่งข้อมูล



ภาพ 54 ขั้นตอนการออก Acess Token



ภาพ 55 ชื่อของ Token (ชื่อของ LINE Notify) (a) เลือกห้องแชทที่ต้องการส่งข้อความแจ้งเตือน(b) จากนั้นกดปุ่มออก Token เพื่อรับ Token key



ภาพ 56 เลข Token ที่ได้สามารถนำไปใช้ในการเขียนโค้ดส่งข้อมูลหรือข้อความ

จากนั้นนำรหัส Token ที่ได้ไปใส่ใน Node-RED เพื่อใส่เงื่อนไขให้เซนเซอร์สามารถส่งข้อความไปทางไลน์ได้เมื่อผู้แจ้งเหตุทำการกดปุ่มอุปกรณ์ SENS-IoT แจ้งเหตุเข้ามาในระบบ

3.12 การออกแบบฐานข้อมูล

ฐานข้อมูลที่เก็บข้อมูลในรูปแบบของ JSON มีการเชื่อมต่อข้อมูลแบบเรียลไทม์ โดยตัวอุปกรณ์เซนเซอร์จะต้องทำการเชื่อมต่อกับอินเทอร์เน็ตเพื่อทำการรับส่งข้อมูลจากอุปกรณ์เข้าสู่ฐานข้อมูล ซึ่งในหัวข้อนี้จะกล่าวถึงการออกแบบโครงสร้างและรูปแบบข้อมูลที่ได้จัดเก็บข้อมูลบนฐานข้อมูล โดยมีรายละเอียดดังนี้

3.12.1 ส่วนของการเก็บข้อมูล GPS และผู้ใช้งาน

ในส่วนของ GPS และข้อมูลของผู้ใช้งานจะมีการเก็บข้อมูลแบบเรียลไทม์ โดยเก็บข้อมูลไว้ภายในฐานข้อมูล sensor ซึ่งจะอยู่ภายใต้เทเบิลที่ชื่อว่า emergency ที่ประกอบไปด้วยข้อมูลดังตาราง 6 ดังนี้

ตาราง 6 แสดงรายละเอียด Value ที่จัดเก็บข้อมูล GPS และผู้ใช้งาน

ชื่อตัวแปร	ความหมาย	รูปแบบการจัดเก็บ
id	จำนวนข้อมูลที่เข้าสู่ฐานข้อมูล	serial
lat	ตำแหน่งของข้อมูลจุดละติจูด	numeric
long	ตำแหน่งของข้อมูลจุดลองจิจูด	numeric
time	เวลา ที่ข้อมูลเข้า	time without time zone
date	วัน/เดือน/ปี/เวลา ที่ข้อมูลเข้า	date
id_number	รหัสอุปกรณ์	numeric
name	ชื่อผู้ใช้งาน	text
sername	นามสกุลผู้ใช้งาน	text
house_number	บ้านเลขที่	numeric
moo	หมู่	numeric
tambon	ตำบล	text
amphoe	อำเภอ	text
province	จังหวัด	text
post_code	รหัสไปรษณีย์	numeric
number_phone	เบอร์โทรศัพท์	numeric
id_card	เลขบัตรประจำตัวประชาชน	numeric
health	โรคประจำตัว	text
allergy	การแพ้ยา	text
geom	ค่าตำแหน่งจุด	Geometry (point,4326)

3.12.2 ส่วนของการเก็บข้อมูลผู้ดูแลระบบ

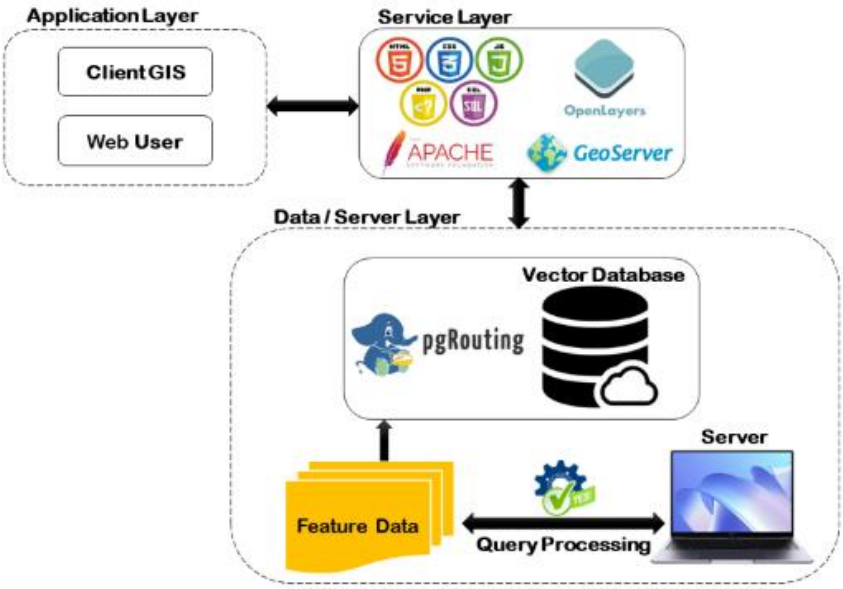
ในส่วนของฐานข้อมูลผู้ดูแลระบบ มีการเก็บข้อมูลขของผู้ดูแลระบบไว้สำหรับใช้ในการ Login เข้าใช้งานหน้าเว็บไซต์ โดยข้อมูลของผู้ดูแลระบบจะเก็บไว้ภายในฐานข้อมูล sensor ซึ่งจะอยู่ภายใต้เทเบิลที่ชื่อว่า adminemergency ที่ประกอบไปด้วยข้อมูลดังตาราง 7 ดังนี้

ตาราง 7 แสดงรายละเอียด Value ที่จัดเก็บข้อมูลผู้ดูแลระบบ

ชื่อตัวแปร	ความหมาย	รูปแบบการจัดเก็บ
name	ชื่อ-นามสกุล	text
tell	เบอร์โทรศัพท์	numeric
usera	ชื่อสำหรับ Login เข้าใช้งานเว็บไซต์	text
pass	รหัสผ่านสำหรับ Login เข้าใช้งาน	text

3.13 Web Map Application for Emergence Routing Service (ERRS)

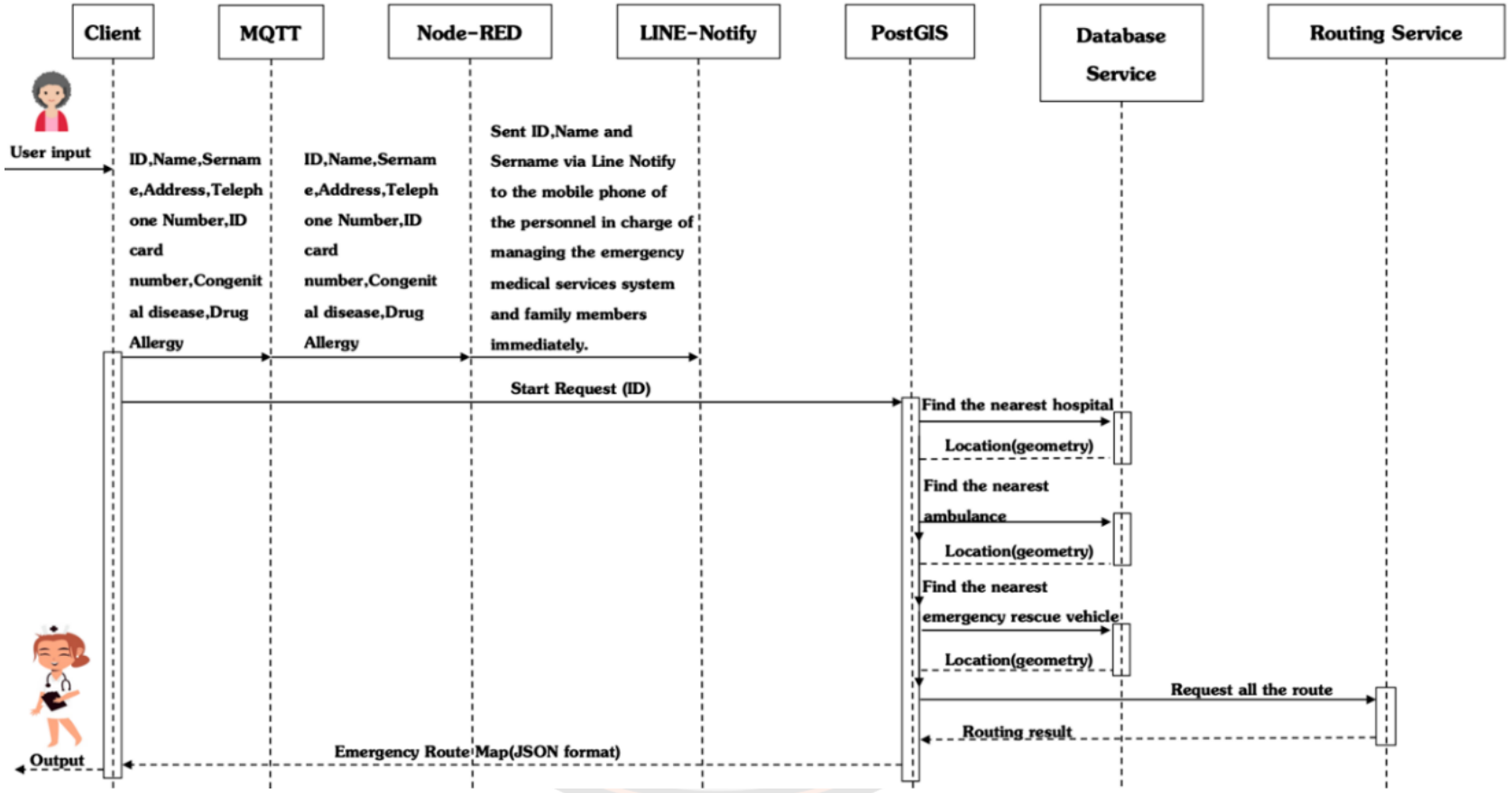
Web Map Application for Emergence Routing Service (ERRS) ได้รับการติดตั้งโดย FOSS4G ซึ่ง PostgreSQL/PostGIS ทำหน้าที่ในการจัดเก็บฐานข้อมูลถนนและข้อมูลตำแหน่งที่ได้รับจากอุปกรณ์ SENS-IoT ซึ่งการพัฒนาหน้าเว็บอินเตอร์เฟสได้ใช้ Open Layers ดังนั้นในการสร้างหน้าเว็บอินเตอร์เฟสจึงต้องใช้ภาษาในการพัฒนาได้แก่ JavaScript, HTML, CSS และ PHP ดังภาพ 57 แสดงข้อมูลตำแหน่งโดยทำการส่งข้อมูลเข้าสู่ระบบฐานข้อมูลและเมื่อระบบได้รายละเอียดของผู้ใช้งานเรียบร้อยแล้ว ระบบก็จะทำการค้นหาเส้นทางที่พัฒนาด้วยโปรแกรม pgRouting ซึ่งทำหน้าที่ในการตรวจสอบฐานข้อมูลถนนเพื่อคำนวณเส้นทาง โดยข้อมูลเครือข่ายถนนจะถูกนำมาใช้สำหรับการค้นหาเส้นทางที่ใกล้ที่สุดโดยใช้อัลกอริทึม pgRouting จากนั้นเราจะนำเข้าข้อมูลเครือข่ายถนนที่อยู่ในรูปแบบ shapefile ไปยังฐานข้อมูลในการกำหนดเส้นทางโดยใช้ PostgreSQL/PostGIS ผ่าน QGIS หลังจากนั้นนำเข้าข้อมูลลงในฐานข้อมูล ขั้นตอนต่อไปเป็นการกำหนดเส้นทางถนนโดยเราต้องสร้างคอลัมน์ gid, source, target และ cost ก่อน ซึ่งจะใช้ในการจัดเก็บพิกัดของต้นทางและเป้าหมาย เพื่อใช้ค้นหารถพยาบาลและรถกู้ภัยฉุกเฉินและค้นหาเส้นทางที่เร็วที่สุดหรือเหมาะสมที่สุดไปยังจุดเกิดเหตุฉุกเฉิน ซึ่งได้ใช้คำสั่ง SQL โดยจะเขียนอยู่ในรูปแบบไฟล์ PHP และผลลัพธ์จะถูกส่งในรูปแบบ GeoJSON เพื่อแสดงบนหน้าเว็บแอปพลิเคชัน



ภาพ 57 หลักการทำงานของ pgRouting บน Web Map Application

3.14 กระบวนการการพัฒนาระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งและการค้นหาเส้นทางสำหรับการให้บริการการแพทย์ฉุกเฉินบนระบบแผนที่ออนไลน์

การศึกษานี้ได้พัฒนาระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งและข้อมูลฉุกเฉินด้านสุขภาพสำหรับผู้สูงอายุและผู้พิการ ในภาพ 58 ในขั้นตอนที่ 1 เมื่อผู้สูงอายุหรือผู้พิการกดปุ่มบนอุปกรณ์ SENS-IoT เพื่อขอความช่วยเหลือ ขั้นตอนการทำงานของระบบจะเริ่มต้นขึ้น ตำแหน่งและรหัสอุปกรณ์จะถูกส่งไปยังเซิร์ฟเวอร์ฐานข้อมูล จากนั้นระบบจะค้นหารายละเอียดของผู้ป่วยโดยค้นหาจากรหัสอุปกรณ์ในฐานข้อมูลของผู้ป่วย ในขั้นตอนที่ 2 หลังจากได้รับรายละเอียดของผู้ป่วย รวมทั้งอาการและตำแหน่งแล้ว ตำแหน่งนั้นจะถูกกำหนดเป็นจุดสิ้นสุด ในขั้นตอนที่ 3-5 ระบบจะค้นหารถพยาบาลที่ใกล้ที่สุด โรงพยาบาลที่ใกล้ที่สุด และรถกู้ภัยฉุกเฉินที่ใกล้ที่สุดจากตำแหน่งของอุปกรณ์และกำหนดให้เป็นจุดเริ่มต้น (x1, y1) จากนั้นระบบจะกำหนดเส้นทางที่สั้นที่สุดหรือเหมาะสมที่สุดไปยังจุดเกิดเหตุฉุกเฉินในขั้นตอนที่ 2 จากนั้นระบบจะส่งคืนผลลัพธ์การกำหนดเส้นทางและแสดงบนหน้า Web Map Application



ภาพ 58 แผนผังการทำงานของระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่ง

บทที่ 4
การทดลองระบบ

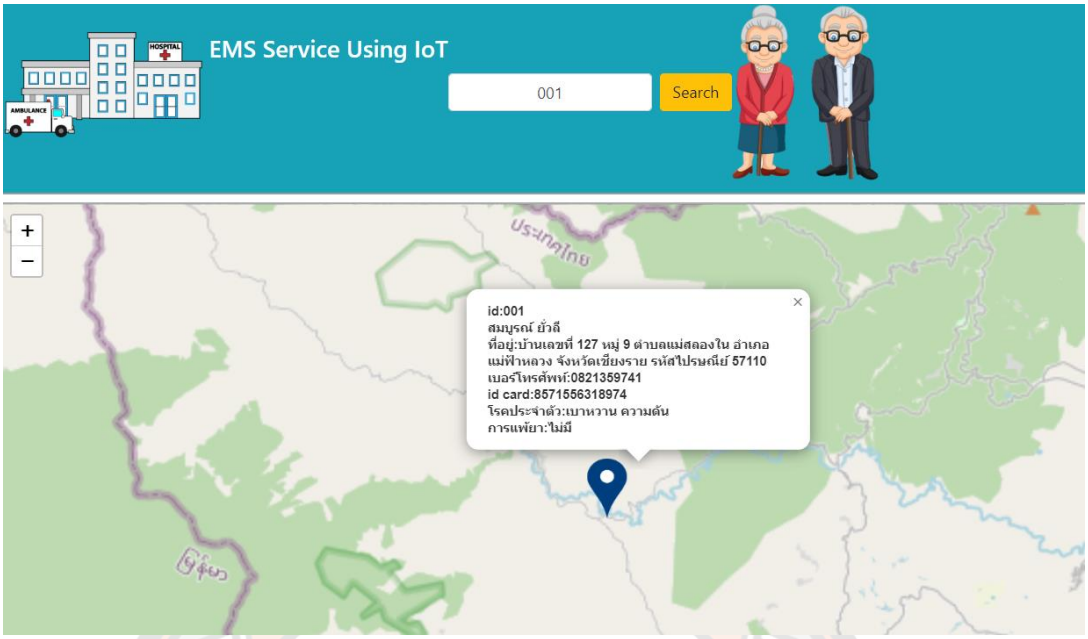
การพัฒนาระบบเซนเซอร์เพื่อแสดงข้อมูลตำแหน่งที่ตั้งของจุดผู้แจ้งเหตุแบบเรียลไทม์บนระบบแผนที่ออนไลน์และวิเคราะห์เส้นทางในการเข้าถึงสำหรับการให้บริการการแพทย์ฉุกเฉิน เป็นการประยุกต์ใช้เทคโนโลยีในการพัฒนาร่วมกัน เพื่อให้ได้อุปกรณ์ Smart Emergency Notification System based on IoT (SENS-IoT) ที่ใช้ในการระบุตำแหน่งในการเข้าถึงสำหรับการให้บริการการแพทย์ฉุกเฉิน และกระบวกการในการพัฒนาระบบวิเคราะห์เส้นทางในการเข้าถึงสำหรับการให้บริการการแพทย์ฉุกเฉินที่มีระบบการทำงานที่สัมพันธ์กัน โดยในบทนี้จึงเป็นการทดสอบการทำงานและพัฒนาระบบ Ublox NEO-6M GPS โดยทดสอบระบบการส่งตำแหน่งเมื่อสัญญาณ Ublox NEO-6M GPS ทำงานโดยการที่กดปุ่มอุปกรณ์ SENS-IoT ระบบจะแจ้งไปทางไลน์ของเจ้าหน้าที่และรับค่าตำแหน่งปัจจุบันแสดงบนแผนที่ออนไลน์ พร้อมทั้งทดสอบระบบการค้นหาเส้นทางที่เหมาะสมที่สุดในการเข้าถึงผู้แจ้งเหตุบนระบบแผนที่ออนไลน์ ระบบแจ้งเตือนเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่งบนอุปกรณ์ IoT ที่แสดงข้อมูลตำแหน่งที่ตั้งของจุดที่ผู้แจ้งเหตุแบบเรียลไทม์ร่วมกับระบบค้นหาเส้นทางจากตำแหน่งรถฉุกเฉิน รถกู้ชีพ ไปยังตำแหน่งที่รับแจ้งนี้มีความผิดพลาดมากน้องเพียงใด ดังผลการวิจัยต่อไปนี้

4.1 การพัฒนาระบบ GPS Tracking

การทดลองออกแบบและพัฒนาอุปกรณ์ระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่ง ที่แสดงข้อมูลตำแหน่งที่ตั้งของผู้แจ้งเหตุแบบเรียลไทม์ โดยตำแหน่งที่ตั้งของผู้แจ้งเหตุจะถูกส่งไปยังระบบฐานข้อมูลแบบ Real-Time โดยทำผ่านโปรแกรม Node-RED (ภาพ 64) และดึงข้อมูลตำแหน่งมาแสดงบนแผนที่เพื่อทดสอบการส่งตำแหน่งมายังฐานข้อมูลและแสดงบนแผนที่ ได้ผลการทดลองดังภาพ 59

id	lat	long	time	date	id_number	name	sername	house_number	moo	tambon	amphoe	province	post_code	number_phone	id_card	health	allergy
6471	16.746306	100.195794	17:35:31.173886	2023-10-03	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6460	16.746306	100.195794	14:10:06.403009	2023-09-28	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6472	16.746306	100.195794	17:35:51.234274	2023-10-03	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6461	16.746306	100.195794	14:15:00.757941	2023-09-28	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6473	16.746306	100.195794	17:36:25.485209	2023-10-03	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6462	16.746306	100.195794	14:15:34.804912	2023-09-28	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6474	16.746306	100.195794	17:36:37.141613	2023-10-03	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6463	16.746306	100.195794	14:15:56.996775	2023-09-28	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี
6475	16.746306	100.195794	17:36:48.303366	2023-10-03	1	สมบูรณ์	ยั่วลี	127	9	แม่สลองใน	แม่ฟ้าหลวง	เชียงราย	57110	821359741	8571556318974	เบาหวาน ความดัน	ไม่มี

ภาพ 59 ข้อมูลที่ส่งเข้าฐานข้อมูลในเครื่องแม่ข่าย



ภาพ 60 ตำแหน่ง GPS Tracking ที่ดึงข้อมูลจากฐานข้อมูลมาแสดงบนแผนที่บนเว็บไซต์

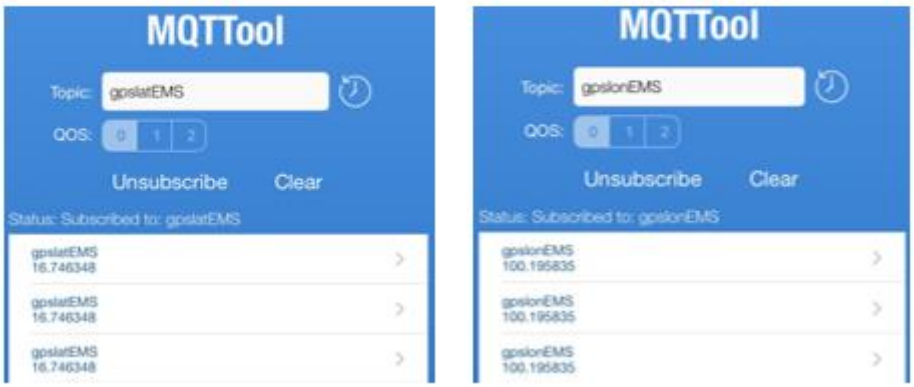
4.2 ผลจากการออกแบบระบบ GPS Tracking

ภาพ 61 ระบบ GPS Tracking สำหรับแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่ง

4.3 ผลการแสดงผลค่าพิกัดละติจูดและลองจิจูดผ่านบน MQTT

การพัฒนาระบบรับส่งข้อมูลผ่าน MQTT Protocol สามารถทำได้โดยการสร้าง Topic ให้แต่ละอุปกรณ์ จากนั้น MQTT Broker จะทำหน้าที่เป็นตัวกลางในการส่งข้อมูลไปยังผู้ใช้ที่ request เข้ามาตาม Broker และ Topic นั้นๆ ในส่วนของ IoT จะใช้ MQTT Message ในการรับส่งข้อมูลซึ่งจะประกอบไปด้วย 2 ส่วนหลัก คือ Topic และ Message ซึ่ง Topic ทำหน้าที่เป็นตัวอ้างอิงหลักของข้อมูลที่จะส่ง Message ออกไปยัง IoT และในส่วนของ Message ทำหน้าที่เป็นหน่วยข้อมูลของระบบที่ใช้ส่งข้อมูลให้ผู้ใช้ทราบถึงสถานะการทำงาน ดังภาพ 67 ด้านล่าง

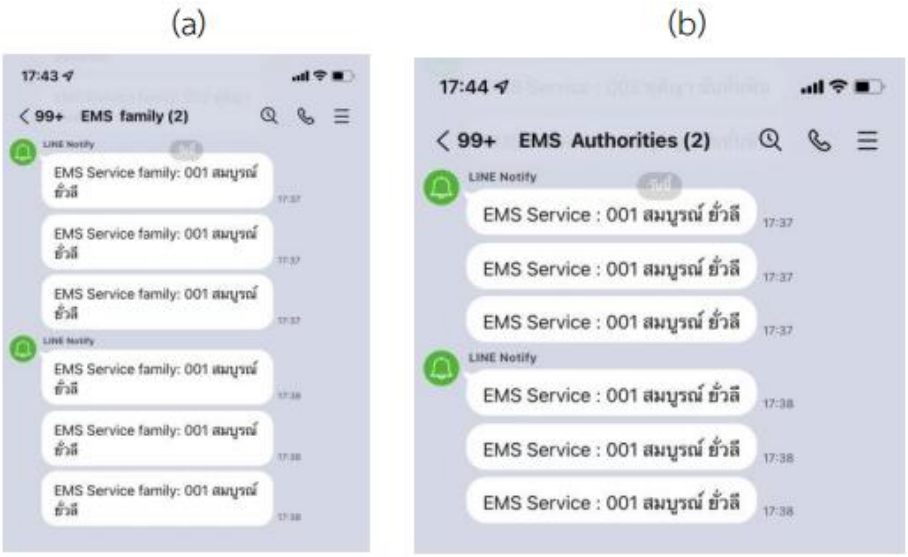
(a) (b)



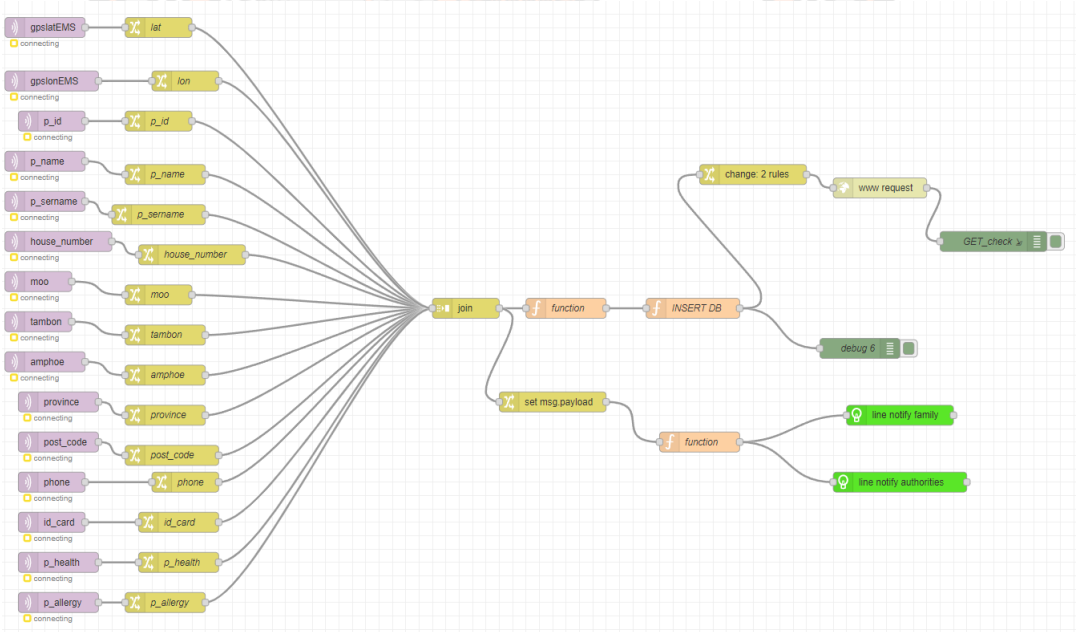
ภาพ 62 ข้อมูล GPS Tracking ที่เข้า MQTT โดย (a) เป็นค่าพิกัดละติจูด และ (b) เป็นค่าพิกัดลองจิจูด

4.4 ผลการพัฒนาระบบแจ้งเตือนผ่าน LINE Notify

ในส่วนของการพัฒนาระบบแจ้งเตือนผ่าน LINE Notify ทำโดยผ่านโปรแกรม Node-RED (ภาพ 64) เมื่อผู้ใช้ทำการกดปุ่มแจ้งเหตุฉุกเฉินเข้ามาระบบจะทำการแจ้งเตือนไปยังสมาร์ทโฟน ของครอบครัว (a) และเจ้าหน้าที่ผู้ที่ดูแลระบบ (b)



ภาพ 63 การแจ้งเตือนผ่าน LINE Notify ไปยังสมาร์ทโฟนของครอบครัว (a) และเจ้าหน้าที่ผู้ที่ดูแลระบบ (b)

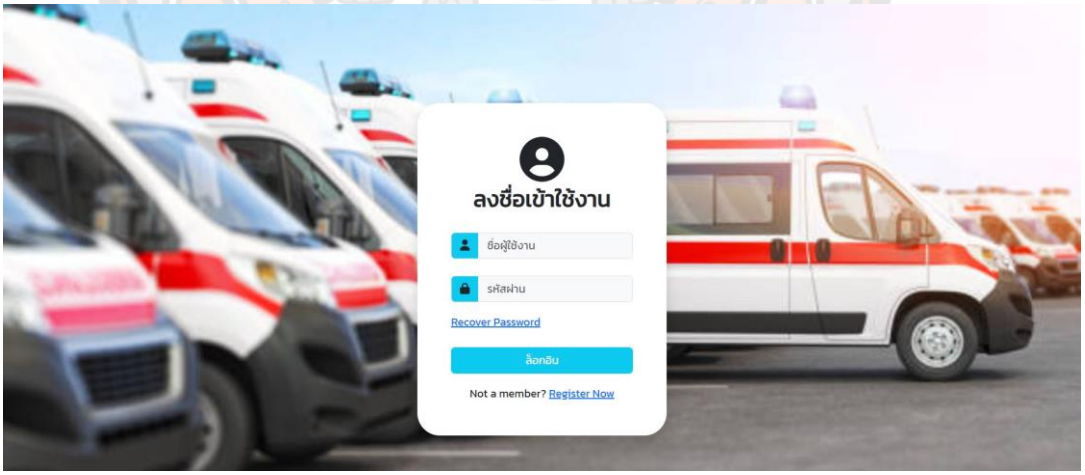


ภาพ 64 โฟลว์ควบคุมเงื่อนไขและการทำงานบน Node-RED

4.5 ผลจากการออกแบบ Web Map Application

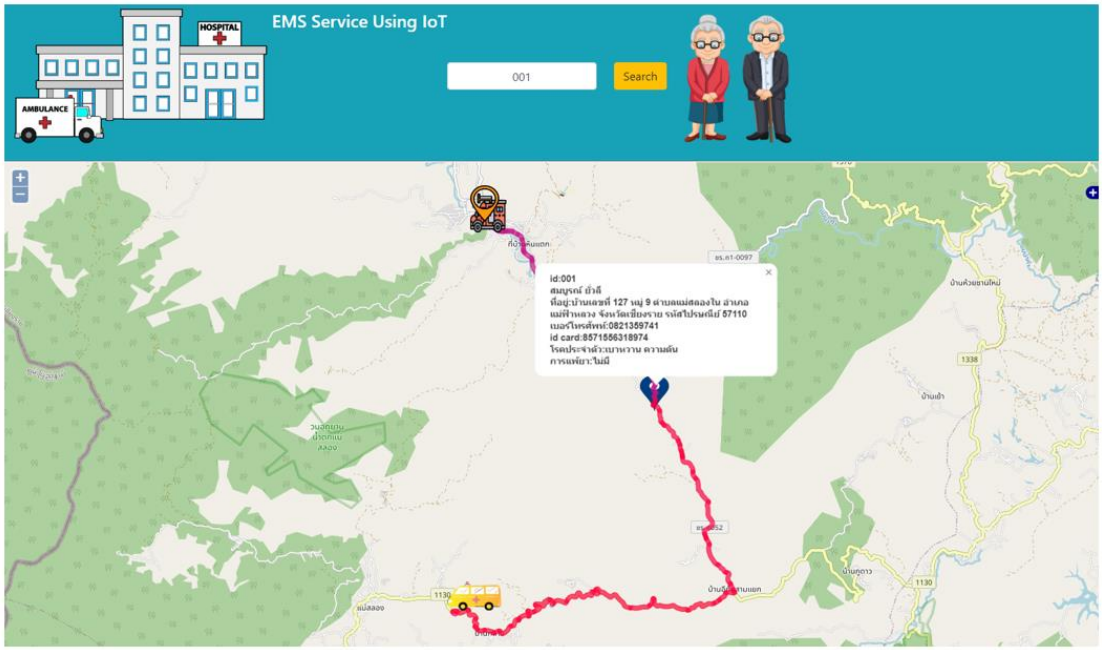
ในส่วนของเว็บไซต์เป็นส่วนที่ออกแบบมาสำหรับให้ผู้ดูแลระบบ สามารถควบคุมและดูแลการใช้งานของระบบได้ โดยหน้าแรกจะเป็นหน้าลงทะเบียน สำหรับเข้าใช้งานระบบ เมื่อลงทะเบียนเรียบร้อยแล้ว (ภาพ 65) ต่อไปก็เข้าสู่หน้า Login ทำการ Login เข้าใช้งานระบบ (ภาพ 66)

ภาพ 65 หน้าลงทะเบียนเข้าระบบ



ภาพ 66 หน้าต่าง Login เข้าระบบ

รูปภาพ 67 เป็นการแสดงผลของหน้า Web Map Application ที่ใช้แผนที่ OpenStreeMap ในการแสดงตำแหน่งของผู้ใช้งานระบบที่ต้องการเรียกรถกู้ชีพและรถโรงพยาบาล จากการกดปุ่มอุปกรณ์ SENS-IoT ภายในหน้า Web Map Application จะแสดงเส้นทางจากรถกู้ชีพและรถโรงพยาบาลที่ใกล้ที่สุดไปยังจุดแจ้งเหตุ ในขณะเดียวกันระบบก็จะทำการคำนวณเส้นทางจากจุดเกิดเหตุไปยังโรงพยาบาลที่เหมาะสมที่สุด โดยใช้อัลกอริทึม pgRouting ในการคำนวณค้นหาเส้นทาง



ภาพ 67 หน้าต่างแสดงผลลัพธ์การค้นหาเส้นทางที่สั้นที่สุด

บทที่ 5
สรุป อภิปราย และข้อเสนอแนะ

จากที่ผู้วิจัยได้ทำการพัฒนาระบบ GPS Tracking ในการส่งตำแหน่งปัจจุบันไปยัง Server และพัฒนาระบบแจ้งเตือนผ่านการกดปุ่มอุปกรณ์ SENS-IoT ขึ้นมาได้ดำเนินงานตามโครงสร้างที่วางเอาไว้โดยผู้วิจัยได้ทำการพัฒนาระบบแจ้งเหตุฉุกเฉินติดตามตำแหน่งแจ้งเหตุฉุกเฉินให้กับผู้ใช้งานได้แบบเรียลไทม์ พร้อมทั้งคำนวณหาเส้นทางของรถโรงพยาบาล รถกู้ชีพและโรงพยาบาลที่ใกล้ที่สุดไปยังจุดแจ้งเหตุฉุกเฉินได้อย่างรวดเร็ว นอกจากนี้ยังสามารถนำระบบนี้ไปประยุกต์ใช้ในสถานการณ์อื่นๆ ได้อีกต่อไปในอนาคต ในส่วนของบทที่ 5 จะกล่าวถึงสรุป อภิปราย และข้อเสนอแนะ การพัฒนาระบบ Geo-IoTและ Web Map Application สำหรับการให้บริการการแพทย์ฉุกเฉิน

5.1 สรุปผล

การพัฒนาอุปกรณ์การแจ้งเหตุฉุกเฉินมีส่วนประกอบ 5 ส่วน คือ 1) Ublox NEO-6M GPS 2) NodeMCU ESP8266 3) ปุ่มกดขอความช่วยเหลือ 4) แหล่งจ่ายไฟ 5) Pocket Wi-Fi ซึ่งการพัฒนาระบบ มีวัตถุประสงค์เพื่อพัฒนานวัตกรรมการแจ้งเหตุฉุกเฉินด้วย Geo-IoT เพื่อใช้แจ้งเหตุฉุกเฉินในการขอความช่วยเหลือ โดยการขอความช่วยเหลือจะมีการทำงานทั้งหมด 3 รูปแบบ คือ 1) ผู้ใช้งานกดปุ่มสีเขียวบนอุปกรณ์ SENS-IoT ระบบจะแจ้งเตือนผ่าน Line Notify ไปยังโทรศัพท์มือถือของสมาชิกครอบครัว 2) ผู้ใช้งานกดปุ่มสีเหลืองบนอุปกรณ์ SENS-IoT ระบบจะแจ้งเตือนผ่าน Line Notify ไปยังโทรศัพท์มือถือของเจ้าหน้าที่ และ 3) ผู้ใช้งานกดปุ่มสีแดงบนอุปกรณ์ SENS-IoT ระบบจะแจ้งเตือนผ่าน Line Notify ไปยังโทรศัพท์มือถือของเจ้าหน้าที่และสมาชิกครอบครัว จากการทดลองระบบพบว่า ฟังก์ชั่นการทำงานทุกส่วนสามารถทำงานได้ถูกต้องตามที่ได้ออกแบบไว้ ทั้งในการทดลองระบบก่อนนำไปใช้งานจริง และทดลองในสถานที่ใช้งานจริง ทำให้การแจ้งเหตุและบอกพิกัดจุดเกิดเหตุส่วนใหญ่ที่ใช้การโทรศัพท์ผ่านเบอร์ 1669 ลดการซักถามระหว่างเจ้าหน้าที่ dispatch center กับผู้แจ้งเหตุฉุกเฉิน ช่วยให้ระยะเวลาปฏิบัติการฉุกเฉิน (emergency operation time) ในการเข้าถึงผู้แจ้งเหตุรวดเร็วมากยิ่งขึ้น ระยะเวลาปฏิบัติการฉุกเฉินมีความสำคัญต่อการปฏิบัติการฉุกเฉิน และมีผลต่ออัตราการรอดชีวิตและลดความพิการของผู้ป่วยฉุกเฉินได้ ในขณะเดียวกันผู้วิจัยได้พัฒนาระบบค้นหาเส้นทางการให้บริการการแพทย์ฉุกเฉินบน Web Map Application ให้กับเจ้าหน้าที่ dispatch center สามารถค้นหาตำแหน่งผู้แจ้งเหตุจากเลขไอดีอุปกรณ์ SENS-IoT จากหน้า Web Map Application ได้อย่างรวดเร็ว

ดังนี้บนหน้า Web Map Application จะแสดงเส้นทางคำนวณระยะทางจากรถพยาบาลที่อยู่ใกล้จุดเกิดเหตุและเหมาะสมที่สุดไปจุดเกิดเหตุ และตำแหน่งโรงพยาบาลที่ใกล้ที่สุด การพัฒนาระบบค้นหาเส้นทางที่เหมาะสมที่สุดด้วย Dijkstra’s algorithm เป็นการทำงานร่วมกับ

PostgreSQL/PostGIS เข้ามาช่วยในการวิเคราะห์ระบบโครงข่ายถนนสำหรับการบริการเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgr_Dijkstra ซึ่งผู้วิจัยได้เพิ่มเงื่อนไข Slope เข้าไปในฟังก์ชัน pgr_Dijkstra เพื่อหลีกเลี่ยงเส้นทางที่มี Slope สูงที่อาจทำให้รถเกิดอุบัติเหตุได้ง่ายหรืออาจทำให้การเดินทางบนเนินเขาหรือพื้นที่ภูเขาใช้เวลานานและช่วยในการวางแผนเส้นทางที่มีความเหมาะสมและความปลอดภัยสำหรับการขับรถหรือการเดินทางของรถพยาบาล ส่งผลให้การออกปฏิบัติการรับผู้ป่วยฉุกเฉินมีความถูกต้องแม่นยำมากยิ่งขึ้น นอกจากนี้เจ้าหน้าที่สามารถทราบข้อมูลรายละเอียดแต่ละตำแหน่งได้ เช่น ชื่อ-นามสกุล ที่อยู่ เบอร์โทรศัพท์เจ้าของอุปกรณ์หรือคนที่เกี่ยวข้อง เลขบัตรประจำตัวประชาชน โรคประจำตัว และการแพ้ยา เพื่อแสดงให้เจ้าหน้าที่เห็นว่า ผู้ป่วยมีระดับความรุนแรงของโรคที่ต้องได้รับการเข้าช่วยเหลืออย่างเร่งด่วน

5.2 อภิปรายผลการวิจัย

ระบบแจ้งเหตุฉุกเฉินด้วยเซนเซอร์ระบุตำแหน่ง มีความสอดคล้องกับงานของ (Majumdar et al., 2010) บทความนี้ศึกษาเกี่ยวกับปุ่มกดที่จะใช้ในช่วงเวลาอุบัติเหตุฉุกเฉิน มีการพัฒนาระบบการแจ้งเหตุฉุกเฉินด้วยบอร์ด Arduino และ GSM module โดยพิกัด GPS จะถูกส่งในรูปแบบของข้อความและแสดงตำแหน่งของผู้แจ้งเหตุฉุกเฉินบน Google Map แต่แตกต่างกันที่ผู้วิจัยพัฒนาปุ่มกดที่ใช้ในการแจ้งเหตุฉุกเฉินด้วยบอร์ด NodeMCU ESP8266 และ Ublox NEO-6M GPS Module ซึ่งมีการพัฒนาการเขียนโค้ดที่ง่ายกว่า บอร์ด NodeMCU ESP8266 มีสายอากาศในตัว จึงทำให้สามารถเชื่อมต่อ Wi-Fi ได้โดยไม่ต้องต่ออุปกรณ์หรือสายอากาศเพิ่มเติม โดยข้อมูลจะถูกส่งในรูปแบบการแจ้งเตือนผ่าน Line Notify ซึ่ง Line Notify ในปัจจุบันเป็นที่นิยมใช้กันอย่างแพร่หลายและแสดงตำแหน่งของผู้แจ้งเหตุฉุกเฉินบน Web Map Application

ในการพัฒนาระบบด้าน Web Map Application ของ (กัลยรัตน์ ชื่นบาน,2556) ใช้ซอฟแวร์พัฒนาเว็บแอปพลิเคชัน ได้แก่ Appserv Apache, PHP, MySQL, phpMyAdmin, Windows 7 Ulitimate และ Dreamweaver CS3 โดยผู้ดูแลระบบสามารถกำหนดพื้นที่เสี่ยงภัยโดยการปักหมุนโดยใช้ Google Maps API และกำหนดพื้นที่ต้นทาง และปลายทางเพื่อหลีกเลี่ยงพื้นที่เสี่ยงภัย ต่างจากงานวิจัยนี้ตรงที่ผู้วิจัย ได้มีการพัฒนา Web Map Application ด้วย PosrgreSQL/PostGIS เนื่องจากมีการทำงานที่ง่ายกว่า MySQL , phpMyAdmin และการแสดงข้อมูลผ่าน Google Maps ก็มีข้อจำกัดในการเรียกข้อมูลออกมาแสดงและการใช้บริการการแสดงตำแหน่งของผู้แจ้งเหตุฉุกเฉินบน Google Map ไม่สามารถแก้ไขข้อมูล อัปเดทข้อมูล หรือสร้างเงื่อนไขและเพิ่มปัจจัยของเส้นทางในการเดินรถได้ เหมือนกับ pgRouting Library

การพัฒนาระบบค้นหาเส้นทางที่เหมาะสมที่สุดในกรณีที่เกิดเหตุฉุกเฉินโดยใช้ pgRouting และ ซอฟต์แวร์รหัสเปิด (Free and Open Source Software for Geospatial: FOSS4G) มีความ

สอดคล้องกับงานของ (Choosumrong et al., 2019) ได้มีการพัฒนาระบบค้นหาเส้นทางที่เหมาะสมที่สุดด้วย pgRouting algorithm ซึ่งเป็นการทำงานร่วมกับ PosrgreSQL/PostGIS เข้ามาช่วยในการวิเคราะห์ระบบโครงข่ายถนนสำหรับการบริการเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgDijkstra โดยมีการสาธิตสองสถานการณ์สำหรับการใช้งานบริการหน้าเว็บ ERDP ในสถานการณ์แรกเป็นเหตุฉุกเฉินทางการแพทย์ ERDP จะคำนวณเส้นทางระหว่างตำแหน่งของผู้ป่วย รถฉุกเฉินไปโรงพยาบาลในบริเวณที่เกิดอุบัติเหตุ โดยคำนึงถึงปัจจัยแบบไดนามิก เช่น สภาพของโครงข่ายถนน สภาพของผู้ป่วย และความพร้อมของสิ่งอำนวยความสะดวกทางการแพทย์และความเชี่ยวชาญในโรงพยาบาลเป้าหมาย ในสถานการณ์ที่สองเป็นสถานการณ์ภัยพิบัติ ซึ่งมีความแตกต่างจากงานวิจัยนี้ตรงที่ผู้วิจัยได้นำข้อมูล Slope ของพื้นที่สูงมาใช้ในการคำนวณเส้นทาง เพื่อช่วยในการวางแผนเส้นทางที่มีความเหมาะสมและความปลอดภัยสำหรับการขับรถหรือการเดินทางของรถพยาบาล และสามารถใช้ในการประมาณเวลาที่จะใช้ในการเดินทางได้ โดยเฉพาะในสถานที่ที่มีเนินเขาหรือพื้นที่ภูเขา เนื่องจากถนนที่มี Slope สูงอาจทำให้รถเกิดอุบัติเหตุได้ง่ายหรืออาจทำให้การเดินทางใช้เวลามากขึ้น ดังนั้นการใช้ข้อมูล Slope ในการเดินทางเป็นเครื่องมือที่สำคัญในการวางแผนและจัดการการเดินทางให้มีประสิทธิภาพและปลอดภัยมากยิ่งขึ้น

การคำนวณเส้นทางแบบสัญจรจริง (Driving Distance) มีความสอดคล้องกับงาน (Humhong et al., 2016) บทความนี้ศึกษาเกี่ยวกับการพัฒนาระบบสนับสนุนการตัดสินใจเชิงพื้นที่สำหรับการพิจารณาจัดสรรผู้เข้าพักอาศัยในหอพักบุคลากรของมหาวิทยาลัยนเรศวร โดยใช้ pgRouting algorithm ซึ่งเป็นเครื่องมือที่ทำงานร่วมกับ PostgreSQL/PostGIS เข้ามาช่วยในการวิเคราะห์ระบบโครงข่ายถนนสำหรับการบริการเส้นทางที่สั้นที่สุดด้วยฟังก์ชัน pgDijkstra ส่วนการวิเคราะห์พื้นที่ให้บริการได้เปรียบเทียบการวิเคราะห์ 2 วิธีการคือ 1) วิเคราะห์แบบรัศมี 25 กิโลเมตร โดยใช้ฟังก์ชัน ST_Buffer และ 2) วิเคราะห์แบบพื้นที่ให้บริการซึ่งคำนวณจากระยะทางสัญจร 25 กิโลเมตร โดยใช้ฟังก์ชัน pgr_drivingDistance และ pgr_alphaShape ผลการศึกษาครั้งนี้พบว่าหากใช้การพิจารณาแบบ Buffer ผู้ขอพักอาศัยไม่สามารถเข้าพักในหอพักบุคลากรของมหาวิทยาลัยนเรศวรได้เนื่องจากตำแหน่งบ้านตั้งอยู่ในเขตการให้บริการของมหาวิทยาลัยนเรศวร แต่ถ้าหากใช้ระบบการตัดสินแบบ Driving Distance ผู้ขอพักอาศัยจะได้สิทธิ์ในการเข้าพักในหอพักบุคลากร เนื่องจากว่าหากพิจารณาจากระยะทางที่ใช้สัญจรจริงแล้ว ตำแหน่งบ้านของผู้ขอพักอยู่นอกพื้นที่รัศมีการให้บริการ 25 กิโลเมตรตามระยะทางสัญจรจริง แต่ในงานของผู้วิจัยไม่ได้ใช้การคำนวณ Buffer เนื่องจากไม่ได้ให้ความแม่นยำเท่ากับ Driving Distance เพราะการคำนวณแบบ Buffer ไม่ได้พิจารณาสภาพการจราจรและข้อมูลถนนจริง ซึ่งอาจส่งผลให้การนำทางไม่แม่นยำเท่าที่ควร และในสถานการณ์ฉุกเฉินที่จราจรหรือสภาพถนนมีการเปลี่ยนแปลงอย่างรวดเร็ว การใช้ Buffer อาจไม่สามารถสร้างเส้นทางที่ปรับปรุงได้อย่างทันท่วงที

5.3 ข้อเสนอแนะ

1. สัญญาณ GPS ถ้าใช้ในบ้านอาจจะต้องรอสัญญาณ ในอนาคตสามารถเพิ่มการแจ้งเตือนพิกัดตำแหน่งผ่าน LINE Bot ได้ ที่เหมือนกับการ Share Location

2. ระยะเวลาของการใช้งานอุปกรณ์ผู้ใช้จำเป็นต้องเปลี่ยนแหล่งจ่ายไฟ เพื่อให้เซนเซอร์ยังสามารถใช้งานได้ต่อไป

3. นอกจาก Slope แล้ว การคำนวณค้นหาเส้นทางที่ดีที่สุดสามารถเพิ่มเงื่อนไขเพิ่มเติมได้ดังนี้ เส้นทางที่ไฟจราจรติดต่อกัน ข้อจำกัดในการเลี้ยว สัญลักษณ์การเดินรถและประเภทถนน

บรรณานุกรม

บรรณานุกรม

บรรณานุกรม

กัญญา วังศรี.(2556).การบริการการแพทย์ฉุกเฉินในประเทศไทย. สืบค้นเมื่อวันที่ 3 มกราคม 2566,จาก http://202.28.95.4/library/main/eproceeding/Lec_69_73.pdf

กัลยรัตน์ ชื่นบาน. (2556). เว็บแอปพลิเคชันสำหรับแสดงพื้นที่ที่ประสบอุทกภัยและบอกเส้นทางเพื่อหลีกเลี่ยงพื้นที่เสี่ยงภัย.สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน.

การจัดการระบบเครือข่ายบนระบบปฏิบัติการลีนุกซ์.(2561).Apache.สืบค้นเมื่อวันที่ 3 มกราคม 2566,จาก https://kridsana.cmtc.ac.th/load/Linux/unix08-WEB%20Server.pdf

ธีระ ศิริสมุด, ปญาดา ชื่นสำโรง, พรทิพย์ วชิรดิลก, และ สุรเดช ดวงทิพย์สิริกุล. (2565). การวิจัยเพื่อพัฒนาคุณภาพของระบบบริการการแพทย์ฉุกเฉินสู่ภาวะปกติใหม่ (EMS new normal) :แนวทางเพิ่มประสิทธิภาพด้านเวลาในการเข้าถึงระบบบริการการแพทย์ฉุกเฉินของผู้ป่วยฉุกเฉินวิกฤตในประเทศไทย. สืบค้นเมื่อวันที่ 29 กันยายน 2566,จาก https://www.niems.go.th/1/UploadAttachFile/2022/EBook/416748_20221108154200.pdf

เนติมา อุดร.(2562). **Web GIS**.สืบค้นเมื่อวันที่ 3 มกราคม 2566, จาก https://www.agi.nu.ac.th/nred/Document/isPDF/2562/geo_2562_04_FullPaper.pdf

นิธิศักดิ์ โพธาราม, วรุฒม์ ศรีมณีธรรมม, และ ยุพิน สรรพคุณ “เว็บแอพพลิเคชั่นระบบสนับสนุนการตัดสินใจการเลือกเส้นทาง (“Web Application Decision Support Route Selection System)” วารสารสถาบันวิจัย มข. ปีที่ 4 ฉบับที่ 1 (มกราคม – เมษายน 2559)

บริษัท โปรซอฟท์ เว็บ จำกัด.CSS คืออะไร มีประโยชน์ อย่างไร.สืบค้นเมื่อวันที่ 3 มกราคม 2566, จาก https://blog.sogoodweb.com/Article/Detail/79237/CSS

พระราชบัญญัติการแพทย์ฉุกเฉิน.(2551).การบริการการแพทย์ฉุกเฉิน (emergency **medical service: EMS)**.สืบค้นเมื่อวันที่ 3 มกราคม 2566,จาก http://nih.dmsc.moph.go.th/law/pdf/003.pdf

พีระวัฒน์ แกล้ววิการณ์, และ สุเพชร จิรขจรกุล. (2014). การประยุกต์ระบบสารสนเทศภูมิศาสตร์เพื่อการวิเคราะห์พื้นที่การให้บริการศูนย์การแพทย์ฉุกเฉิน จังหวัดเลย. Thai Journal of Science and Technology, 3(3), 137-147.

วิชิต เหล็กคำ.(2559). MQTT.สืบค้นเมื่อวันที่ 3 มกราคม 2566, จาก http://etheses.psru.ac.th/libirpsru/sites/default/files/site/default/research/15A.pdf

ศูนย์วิจัยภูมิสารสนเทศเพื่อประเทศไทย.ระบบสารสนเทศภูมิศาสตร์. สืบค้นเมื่อวันที่ 4 มกราคม 2566, จาก http://www.gisthai.org/about-gis/gis.html/

สถาบันการแพทย์ฉุกเฉินแห่งชาติ.(2566).เกณฑ์การคัดแยกผู้ป่วยฉุกเฉินและจัดลำดับการบริบาล ณ ห้องฉุกเฉินตามหลักเกณฑ์ที่ กพฉ.กำหนด.สืบค้นเมื่อวันที่ 3 มกราคม 2566,จาก https://www.niems.go.th/1/upload/migrate/file/255605231830151553_6BQPypE0cjSQopFX.pdf

สิทธิชัย ชูสำโรง. (2565). THE ONLINE GIS. พิษณุโลก:สำนักพิมพ์รัตนสุวรรณการพิมพ์ 3.

สมเกียรติ ช่อเหมือน.Node-RED คืออะไร.สืบค้นเมื่อวันที่ 3 มกราคม 2566,จาก http://courseware.npru.ac.th/admin/files/20190923100134_2dbc75d256b744e05178ad20a449326d.pptx

สวรินทร์ ฤกษ์อยู่สุข และคณะ. (2556). ซอฟต์แวร์รหัสเปิดและแหล่งข้อมูลภูมิสารสนเทศแบบปิดที่ไม่เสียค่าใช้จ่ายสำหรับการประยุกต์ใช้งานภูมิสารสนเทศ. สำนักวิชาวิทยาศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.

อภิชัย วิจักษณ์ประเสริฐ, และ คณิต ไข่มุกด์. (2553). ระบบสารสนเทศภูมิศาสตร์เพื่อการจัดการขยะมูลฝอยและการบำบัดน้ำเสียขององค์กรปกครองส่วนท้องถิ่น. สาขาวิชาเทคโนโลยีสารสนเทศสำนักวิชาเทคโนโลยีสังคม มหาวิทยาลัยเทคโนโลยีสุรนารี.

Bhatia, T. S., Singh, H., Litoria, P. K., & Pateriya, B. (2018). Web GIS Development using Open Source Leaflet and Geoserver Toolkit [J]. International Journal of Computer Science and Technology, 9(3), 29-33.

Choosumrong, S., Raghavan, V., Delucchi, L., Yoshida, D., & Vinayaraj, P. (2014). Implementation of dynamic routing as a web service for emergency routing decision planning.

Choosumrong, S., Humhong, C., Raghavan, V., & Fenoy, G. (2019). Development of optimal routing service for emergency scenarios using pgRouting and FOSS4G. Spatial Information Research, 27, 465-474.

Choosumrong, S., Raghavan, V., & Realini, E. (2010). Implementation of dynamic cost based routing for navigation under real road conditions using FOSS4G and OpenStreetMap. Proceedings of GIS-IDEAS 2010, Hanoi, Vietnam, 9-11 December 2010, 53-58.

Humhong, C., Choosumron, S., & Homhuan, S. (2016). Development of Spatial Decision Support System for Considering Staff Dormitory of Naresuan University using FOSS4G/Development of Spatial Decision Support System for Considering Staff Dormitory of Naresuan University using FOSS4G For consideration of the allocation of guests living in personnel dormitories. of Naresuan University With open code software. Naresuan University Journal: Science and Technology (NUJST), 24(1), 115-127.

Global Positioning System.สืบค้นเมื่อ 4 มกราคม 2566, จาก https://www.prosoftgps.com/Article/Detail/70661

Java JavaScript. (2566). JavaScript คืออะไร. สืบค้นเมื่อวันที่ 3 มกราคม 2566, จาก https://www.mindphp.com/

Li, X. F. (2012). The Research based on u-Blox GPS for Material Yard Vehicle Monitor. In Advanced Engineering Forum (Vol. 6, pp. 536-541). Trans Tech Publications Ltd.

Majumdar, N., Bhargava, P., & Shirin, R. K. (2014). Emergency panic button using microcontrollers.International Journal of Computer Applications, 975, 8887.

NodeMCU, สืบค้นเมื่อวันที่ 3 มกราคม 2566.จาก https://www.geocities.ws/sunya/microcontroller/arduino/NodeMCU_LAB1.pdf

OpenStreetMap.สืบค้นเมื่อวันที่ 4 มกราคม 2566, จาก https://docs.fileformat.com/th/gis/osm/

Pham, H. D., Drieberg, M., & Nguyen, C. C. (2013, December). Development of vehicle tracking system using GPS and GSM modem. In 2013 IEEE conference on open systems (ICOS)(pp. 89-94). IEEE.

PHP. (2565). PHP คืออะไร. สืบค้นเมื่อวันที่ 3 มกราคม 2566, จาก https://www.mindphp.com/

Sayan Sunya.Node MCU.สืบค้นเมื่อวันที่ 4 มกราคม 2566, จาก https://www.geocities.ws/sunya/microcontroller/arduino/NodeMCU_LAB1.pdf

Senthilkumar, S., Brindha, K., & Bhandari, S. (2020). Vehicle accident management and control system using MQTT. International Journal of Advances in Applied Sciences (IJAAS), 9(1), 1-11.

u-blox 6 GPS Modules, สืบค้นเมื่อวันที่ 3 มกราคม 2566.จาก https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/

Varsha Mali, Madhuri Rao and S. S. Mantha. (2012). Enhanced Routing in Disaster Management based on GIS. Thadomal Shahani Engineering College, Bandra, Mumbai50.

ภาคผนวก

ภาคผนวก

ภาคผนวก Source Code ที่ใช้ในการพัฒนาระบบ

Source Code ที่ใช้ในการพัฒนาระบบการทำงานของระบบเซ็นเซอร์รูปแบบ (Wi-Fi-Internet) ในการบันทึกค่าข้อมูลไปยัง MQTT (gps_tracking.ino)

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
const char* ssid = "xxxxxxxxxx";
const char* password = "xxxxxxxxx";
const char* username = "xxxxxxxxx"; // my AskSensors username
const char* pubTopiclat = "gpslatEMS"; // publish/username/apiKeyIn
const char* pubTopiclon = "gpslonEMS"; // publish/username/apiKeyIn
const char* pubTopicid = "p_id"; // publish/username/apiKeyIn
const char* pubTopicname = "p_name"; // publish/username/apiKeyIn
const char* pubTopicsername = "p_sername"; // publish/username/apiKeyIn
const char* pubTopichouse= "house_number"; // publish/username/apiKeyIn
const char* pubTopicmoo= "moo"; // publish/username/apiKeyIn
const char* pubTopictambon= "tambon"; // publish/username/apiKeyIn
const char* pubTopicamphoe = "amphoe"; // publish/username/apiKeyIn
const char* pubTopicprovince= "province"; // publish/username/apiKeyIn
const char* pubTopicpostcode = "post_code"; // publish/username/apiKeyIn
const char* pubTopicphone= "phone"; // publish/username/apiKeyIn
const char* pubTopiccard= "id_card"; // publish/username/apiKeyIn
const char* pubTopichealth = "p_health"; // publish/username/apiKeyIn
const char* pubTopicallergy = "p_allergy"; // publish/username/apiKeyIn
const unsigned int writeInterval = 25000; // write interval (in ms)
static const int RXPin = D1, TXPin = D2;
static const uint32_t GPSBaud = 9600;
```

```
//AskSensors MQTT config
const char* mqtt_server = "xxxxxxxxx";
unsigned int mqtt_port = 1883;
// objects
WiFiClient espClient;
PubSubClient client(espClient);
TinyGPSPlus gps; // The TinyGPS++ object
SoftwareSerial ss(RXPin, TXPin); // The serial connection to the GPS device

// setup
void setup() {
Serial.begin(115200);
Serial.println("*****************************************************");
Serial.println("********** Program Start : ESP32 publishes NEO-6M GPS position to
AskSensors over MQTT");
Serial.print("********** connecting to WIFI : ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("->WiFi connected");
Serial.println("->IP address: ");
Serial.println(WiFi.localIP());
client.setServer(mqtt_server, mqtt_port);
client.setCallback(callback);

// GPS baud rate
ss.begin(GPSBaud);
```

```
}
// loop
void loop() {
if (!client.connected())
reconnect();
client.loop();
// This sketch displays information every time a new sentence is correctly encoded.
while (ss.available() > 0)
if (gps.encode(ss.read()))
displayInfo();
if (millis() > 5000 && gps.charsProcessed() < 10)
{
Serial.println(F("No GPS detected: check wiring."));
while(true);
}
if (gps.location.isValid()) {
double latitude = (gps.location.lat());
double longitude = (gps.location.lng());
Serial.println("********** Publish MQTT data to ASKSENSORS");
char mqtt_payload_lat[50] = "";
snprintf (mqtt_payload_lat, 50, "%lf", latitude);
Serial.print("Lat: ");
Serial.println(mqtt_payload_lat);
client.publish(pubTopiclat, mqtt_payload_lat);
char mqtt_payload_lon[50] = "";
snprintf (mqtt_payload_lon, 50, "%lf", longitude);
Serial.print("Lon: ");
Serial.println(mqtt_payload_lon);
client.publish(pubTopiclon, mqtt_payload_lon);
client.publish(pubTopicid, "001");
client.publish(pubTopicname, "rhutairat");
```

```
client.publish(pubTopicsername, "hataitara");
client.publish(pubTopichouse, "99");
client.publish(pubTopicmoo, "9");
client.publish(pubTopictambon, "Tha Pho");
client.publish(pubTopicamphoe, "Mueng");
client.publish(pubTopicprovince, "Phitsanulok");
client.publish(pubTopicpostcode, "65000");
client.publish(pubTopicphone, "0829083651");
client.publish(pubTopiccard, "1234567894561");
client.publish(pubTopichealth, "diabetes");
client.publish(pubTopicallergy, "no");
Serial.println("********** End ");
delay(1000);
} else {
Serial.println(F("INVALID"));
}
}
// GPS displayInfo
void displayInfo() {
}
//MQTT callback
void callback(char* topic, byte* payload, unsigned int length) {
Serial.print("Message arrived [");
Serial.print(topic);
Serial.print("] ");
for (int i = 0; i < length; i++) {
Serial.print((char)payload[i]);
}
Serial.println();
}
//MQTT reconnect
```

```
void reconnect() {
// Loop until we're reconnected
while (!client.connected()) {
Serial.print("********** Attempting MQTT connection...");
// Attempt to connect
if (client.connect("ESP32Client", username, "")) {
Serial.println("-> MQTT client connected");
} else {
Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println("-> try again in 5 seconds");
// Wait 5 seconds before retrying
delay(5000);
}
}
}
```

Source Code การเรียกแผนที่มาแสดงหน้าเว็บ (index_emergency.html)

```
<html>
<head>
<title>pgRouting On the Web</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<meta name ="viewport" content="width=device-width,initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
scr="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></scr
ipt>
<script
```

```
scr="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.js"></script
>
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
integrity="sha512-
xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sM
ZMZ19scR4PsZChSR7A=="
crossorigin=""/>
<!-- Make sure you put this AFTER Leaflet's CSS -->
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
integrity="sha512-
XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVU
earIOBhiXZ5V3ynxwA=="
crossorigin=""></script>
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-
ajax/2.1.0/leaflet.ajax.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-
ajax/2.1.0/leaflet.ajax.js"></script>
<script src="https://code.jquery.com/jquery-1.11.3.min.js"></script>
<script type="text/javascript" src="http://media.scraperwiki.com/js/jquery-
1.5.2.min.js"></script>
<script type="text/javascript"
src="http://www.openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript"
src="http://www.openstreetmap.org/openlayers/OpenStreetMap.js"></script>
<script src="http://maps.google.com/maps/api/js?v=3.6&amp;sensor=false"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
<script type="text/javascript">

var  start, stop,
result,ways_nai,ways_chan,ways_faluang,hospitalspoint,emspoint,hospitls_chan,ems,vi
ll,ambulane ;
```

```
var map, click, status;
OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control, {
defaultHandlerOptions: {
'single': true,
'double': false,
'pixelTolerance': 0,
'stopSingle': false,
'stopDouble': false
},
initialize: function(options) {
this.handlerOptions = OpenLayers.Util.extend(
{}, this.defaultHandlerOptions
);
OpenLayers.Control.prototype.initialize.apply(
this, arguments
);
this.handler = new OpenLayers.Handler.Click(
this, {
'click': this.trigger
}, this.handlerOptions
);
},
trigger: function(e) {
var xy = map.getLonLatFromPixel(e.xy);
var retOut = ""+xy.lon+" "+xy.lat+"";
if (status == 'start') {
$("#start").val(retOut);
start.removeFeatures(start.features );


var poi = new OpenLayers.Geometry.Point(xy.lon, xy.lat);
```

```
var feature = new OpenLayers.Feature.Vector(poi);
start.addFeatures([feature]);
getroutsalongnok();
getroutsalongnai();
getroutmaechan();
getroutmaefaluang();
getambulance();
gethospitals();
gethospitalsems();
getems();
genvill();
genambulane ();
} else if (status == 'end') {
$("#end").val(retOut);
stop.removeFeatures(stop.features );
var poi = new OpenLayers.Geometry.Point(xy.lon, xy.lat);
var feature = new OpenLayers.Feature.Vector(poi);
stop.addFeatures([feature]);
getroutsalongnok();
getroutsalongnai();
getroutmaechan();
getroutmaefaluang();
getambulance();
gethospitals();
gethospitalsems();
getems();
genvill();
genambulane ();
}}});

function setcoordinates(val){
```

```
click.activate();
status = val;
}
//กำหนดค่ากึ่งกลางแผนที่และการซูมเริ่มต้น
var lat=20.273036152384464;
var lon=99.63793251080035;
var zoom=12;
var emsIcon = L.icon ({
iconUrl: 'http://localhost/online_gis/icon.png',
iconSize: [50,95],
iconAnchor:[22,94],
popupAnchor:[12,-90]
})
//Initialise the 'map' object
function init(){
map = new OpenLayers.Map({
div: "map",
projection: "EPSG:3857",
displayProjection: "EPSG:4326",
maxResolution: 'auto',
//units:"m",
});
map.addControl(new OpenLayers.Control.Attribution());
map.addControl(new OpenLayers.Control.MousePosition());
//map.addControl(new OpenLayers.Control.PanZoomBar());
map.addControl(new OpenLayers.Control.LayerSwitcher());
// Add Base Map
var mapnik = new OpenLayers.Layer.OSM();

//Add map layers
map.addLayers([mapnik]);
```

```
//กำหนดค่ากึ่งกลางแผนที่และการซูมเริ่มต้น
map.setCenter(new OpenLayers.LonLat(lon, lat).transform(
new OpenLayers.Projection("EPSG:4326"),
map.getProjectionObject()), zoom);
click = new OpenLayers.Control.Click();
map.addControl(click);
var stop_style = OpenLayers.Util.applyDefaults({
externalGraphic: "https://cdn0.iconfinder.com/data/icons/small-n-flat/24/678111-
map-marker-512.png",
graphicWidth: 40,
graphicHeight: 50,
graphicYOffset: -26,
graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);

var start_style = OpenLayers.Util.applyDefaults({
externalGraphic: "http://localhost/online/lo.png",
graphicWidth: 40,
graphicHeight: 50,
graphicYOffset: -26,
graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);
var result_style = OpenLayers.Util.applyDefaults({
strokeWidth: 7,
strokeColor: "#FF0040",
hoverFillOpacity: 0.7,
strokeOpacity: 0.7,
fillOpacity: 0.6
}, OpenLayers.Feature.Vector.style['default']);

var ways_salongnai = OpenLayers.Util.applyDefaults({
```

```
strokeWidth: 7,
strokeColor: "#0000FF",
hoverFillOpacity: 0.7,
strokeOpacity: 0.7,
fillOpacity: 0.6
}, OpenLayers.Feature.Vector.style['default']);

var ways_maechan = OpenLayers.Util.applyDefaults({
strokeWidth: 7,
strokeColor: "#FFD700",
hoverFillOpacity: 0.7,
strokeOpacity: 0.7,
fillOpacity: 0.6
}, OpenLayers.Feature.Vector.style['default']);

var ways_maefaluang = OpenLayers.Util.applyDefaults({
strokeWidth: 7,
strokeColor: "#C71585",
hoverFillOpacity: 0.7,
strokeOpacity: 0.7,
fillOpacity: 0.6
}, OpenLayers.Feature.Vector.style['default']);

var hospitalspoint_style = OpenLayers.Util.applyDefaults({
externalGraphic: "http://localhost/online/amb1.png",
graphicWidth: 50,
graphicHeight: 45,
graphicYOffset: -25,
graphicXOffset: -10,
graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);
```

```
var emspoint_style = OpenLayers.Util.applyDefaults({
externalGraphic: "http://localhost/online/hospi.png",
graphicWidth: 40,
graphicHeight: 50,
graphicYOffset: -45,
graphicXOffset: -10,
graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);

var  hospitls_style = OpenLayers.Util.applyDefaults({
strokeWidth: 7,
strokeColor: "#FF0000",
hoverFillOpacity: 0.7,
strokeOpacity: 0.7,
fillOpacity: 0.6
}, OpenLayers.Feature.Vector.style['default']);

var ems_style = OpenLayers.Util.applyDefaults({
externalGraphic: "http://localhost/online/cara.png",
graphicWidth: 75,
graphicHeight: 40,
graphicYOffset: -45,
graphicXOffset: -10,
graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);

var vill_style = OpenLayers.Util.applyDefaults({
externalGraphic: "http://localhost/online/hospi.png",
graphicWidth: 40,
graphicHeight: 50,
```

```
graphicYOffset: -45,
graphicXOffset: -10,
graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);

var ambu_style = OpenLayers.Util.applyDefaults({
externalGraphic: "http://localhost/online/cara.png",
graphicWidth: 75,
graphicHeight: 40,
graphicYOffset: -45,
graphicXOffset: -10,
graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);

start     = new OpenLayers.Layer.Vector("จุดเกิดเหตุ", {style: start_style});
way_nok   = new OpenLayers.Layer.Vector("รถกู้ชีพแม่สลองนอก", {style: result_style});
ways_nai    = new OpenLayers.Layer.Vector("รถกู้ชีพแม่สลองใน", {style: ways_salongnai});
ways_chan    = new OpenLayers.Layer.Vector("รถโรงพยาบาลแม่จัน", {style:
ways_maechan});
ways_faluang    = new OpenLayers.Layer.Vector("รถโรงพยาบาลแม่ฟ้าหลวง", {style:
ways_maefaluang});
hospitalspoint      = new OpenLayers.Layer.Vector("โรงพยาบาลที่อยู่ใกล้ที่สุด", {style:
hospitalspoint_style});
ems      = new OpenLayers.Layer.Vector("รถกู้ชีพที่อยู่ใกล้ที่สุด", {style: ems_style});

map.addLayers([start,way_nok,ways_nai,ways_chan,ways_faluang,hospitalspoint,ems]);

"http://localhost/cgi-
bin/mapserv.exe?map=C:/ms4w/Apache/htdocs/pgRouting/mapfiles/mymapfile.map",
{transparent: 'true', layers: 'plk_hospitals', format: 'png',numZoomLevels:'20'}
);
```

```
}
function getroutsalongnok() {
var url = 'http://localhost/pgRouting/salongnok.php?';
var lon = $("#start").val();
var lat = $("#end").val();
var mtd = $("#method").val();
//// alert (url);
$.ajax({
url: url+'startpoint='+lon+'&method='+mtd+'&region=ways&srid=3857',
success: function(data){
var GeoJSON = new OpenLayers.Format.GeoJSON();
var features = GeoJSON.read(data);
result.removeFeatures(result.features);
result.addFeatures(features);
}
});
//alert (result);
}
function getroutsalongnai() {
var url = 'http://localhost/pgRouting/salongnai.php?';
var lon = $("#start").val();
var lat = $("#end").val();
var mtd = $("#method").val();
//// alert (url);
$.ajax({
url: url+'startpoint='+lon+'&method='+mtd+'&region=ways&srid=3857',
success: function(data){
var GeoJSON = new OpenLayers.Format.GeoJSON();
var features = GeoJSON.read(data);
ways_nai.removeFeatures(ways_nai.features);
ways_nai.addFeatures(features);
```

```
}
});
//alert (result);
}
function getroutmaechan() {
var url = 'http://localhost/pgRouting/maechan.php?';
var lon = $("#start").val();
var lat = $("#end").val();
var mtd = $("#method").val();
//// alert (url);
$.ajax({
url: url+'startpoint='+lon+'&method='+mtd+'&region=ways&srid=3857',
success: function(data){
var GeoJSON = new OpenLayers.Format.GeoJSON();
var features = GeoJSON.read(data);
ways_chan.removeFeatures(ways_chan.features);
ways_chan.addFeatures(features);
}
});
//alert (result);
}
function getroutmaefaluang() {
var url = 'http://localhost/pgRouting/maefaluang.php?';
var lon = $("#start").val();
var lat = $("#end").val();
var mtd = $("#method").val();
//// alert (url);
$.ajax({
url: url+'startpoint='+lon+'&method='+mtd+'&region=ways&srid=3857',
success: function(data){
var GeoJSON = new OpenLayers.Format.GeoJSON();
```

```
var features = GeoJSON.read(data);
ways_faluang .removeFeatures(ways_faluang .features);
ways_faluang .addFeatures(features);
}
});
//alert (result);
}
function getambulance() {
var url = 'http://localhost/pgRouting/ambulance.php?';
var lon = $("#start").val();
var lat = $("#end").val();
var mtd = $("#method").val();
//// alert (url);
$.ajax({
url: url+'startpoint='+lon+'&method='+mtd+'&region=ways&srid=3857',
success: function(data){
var GeoJSON = new OpenLayers.Format.GeoJSON();
var features = GeoJSON.read(data);
hospitalspoint .removeFeatures(hospitalspoint.features);
hospitalspoint .addFeatures(features);
}
});
//alert (result);
}
function gethospitals() {
var url = 'http://localhost/pgRouting/hospitals.php?';
var lon = $("#start").val();
var lat = $("#end").val();
var mtd = $("#method").val();
//// alert (url);
$.ajax({
```

```
url: url+'startpoint='+lon+'&method='+mtd+'&region=ways&srid=3857',
success: function(data){
var GeoJSON = new OpenLayers.Format.GeoJSON();
var features = GeoJSON.read(data);
emspoint  .removeFeatures(emspoint .features);
emspoint  .addFeatures(features);
}
});
//alert (result);
}

function getems() {
var url = 'http://localhost/pgRouting/point_amb.php?';
var lon = $("#start").val();
var lat = $("#end").val();
var mtd = $("#method").val();
//// alert (url);
$.ajax({
url: url+'startpoint='+lon+'&method='+mtd+'&region=ways&srid=3857',
success: function(data){
var GeoJSON = new OpenLayers.Format.GeoJSON();
var features = GeoJSON.read(data);
ems .removeFeatures(ems .features);
ems .addFeatures(features);
}
});
//alert (result);
}
function toggleControl(element) {
for (key in controls) {
if (element.value == key && element.checked) {
```

```
controls[key].activate();
}   else {
controls[key].deactivate();
}
}
}
</script>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body onload="init()">

<nav class="navbar navbar-expand-sm bg-info navbar-info">
<h5 class="text-white h4">
<a>  <img src ="hos.png" width=350px class="text-white h4 align-text-top">
EMS Service Using IoT
</a>
</h5 >
<form action="emergency.php" method="post" name="form" target="iframe_target">
<form class="form-inline">
<input class="form-control mr-sm-4 text-center" type="ID" placeholder="ID" aria-
label="ID">
<button  class="btn btn-warning mr-sm-4"  type="submit"
onclick="getroutsalongnok(); getroutsalongnai(); getroutmaechan();
getroutmaefaluang(); getambulance(); gethospitals(); getems(); ">Search</button>
</form>
<a class="navbar-brand" href="#"><img src ="en1.png" width=150px class="text-white
h4 align-text-top"></a>
</nav>
<table width="80%" height="90%" align="left" border="1">
<tr> <td>
<div id="map" style="width: 100%; height: 100%;"></div>
```

```
</td></tr>
</table>
</div>
</form>
<fieldset>
<iframe id="iframe_target" name="iframe_target" src=" index_emergency.html"
style="width:0;height:0;border:0px solid #fff;"></iframe>
</fieldset>
</body>
</html>
```

คำสั่งในการลงทะเบียนเข้าสู่ระบบ (register.html)

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
crossorigin="anonymous">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css" integrity="sha512-
iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46Zmdt
Ac9eNBvH0H/ZpiBw==" crossorigin="anonymous" referrerpolicy="no-referrer" />
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Kanit&display=swap');
      *{font-family: 'Kanit', sans-serif;}
```

```
      body {
  background-image:
url('https://media.istockphoto.com/id/1333266664/th/%E0%B8%A3%E0%B8%B9%E0
%B8%9B%E0%B8%96%E0%B9%88%E0%B8%B2%E0%B8%A2/%E0%B8%A3%E0%B8%
96%E0%B8%9E%E0%B8%A2%E0%B8%B2%E0%B8%9A%E0%B8%B2%E0%B8%A5%E0
%B9%83%E0%B8%99%E0%B9%81%E0%B8%96%E0%B8%A7%E0%B8%9A%E0%B8%9
9%E0%B8%96%E0%B8%99%E0%B8%99%E0%B8%82%E0%B8%AD%E0%B8%87%E0%
B8%95%E0%B8%B6%E0%B8%81%E0%B8%A3%E0%B8%B0%E0%B8%9F%E0%B9%89
%E0%B8%B2%E0%B9%83%E0%B8%99%E0%B9%80%E0%B8%A1%E0%B8%B7%E0%B
8%AD%E0%B8%87.jpg?s=612x612&w=0&k=20&c=4VZa94quvGCvL3OwD3pR_a8yiff0K8
H6xVUhcWobJwk=');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
    </style>
</head>
<div class="d-flex justify-content-center align-items-center vh-100">
    <div class="bg-white p-4 rounded-4 shadow">
     <div class="text-center"><i class="fa-solid fa-circle-user fa-4x "></i></div>
    <div class="text-center fs-2 fw-bold">สมัครสมาชิก</div>
    <div class="input-group mt-4">
<input class="form-control bg-light" type="name" placeholder="ชื่อ">
    </div>
    <div class="input-group mt-4">
<input class="form-control bg-light" type="sername" placeholder="นามสกุล">
    </div>
    <div class="input-group mt-4">
        <input class="form-control bg-light" type="text" placeholder="ชื่อผู้ใช้งาน">
            </div>
            <div class="input-group mt-4">
```

```
                <input class="form-control bg-light" type="password"
placeholder="รหัสผ่าน">
                    </div>
    <div class="btn btn-info text-white w-100 mt-4">ลงทะเบียน</div>
    <div class="text-center mt-3"> <div class="not-member">
        Already have an account? <a href="index_login.html">Login here</a>
    </div></div>
    </div>
</div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
crossorigin="anonymous"></script>
  </body>
</html>
```

คำสั่งในการ Login เข้าสู่ระบบ (login.html)

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap demo</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
crossorigin="anonymous">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css" integrity="sha512-
```

iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46Zmdt
Ac9eNBvH0H/ZpiBw==" crossorigin="anonymous" referrerpolicy="no-referrer" />

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Kanit&display=swap');
*{font-family: 'Kanit', sans-serif;}
body {
background-image:
url('https://media.istockphoto.com/id/1310709707/th/%E0%B8%A3%E0%B8%B9%E0
%B8%9B%E0%B8%96%E0%B9%88%E0%B8%B2%E0%B8%A2/%E0%B8%A3%E0%B8%
96%E0%B8%9E%E0%B8%A2%E0%B8%B2%E0%B8%9A%E0%B8%B2%E0%B8%A5%E0
%B9%83%E0%B8%99%E0%B9%81%E0%B8%96%E0%B8%A7%E0%B8%9A%E0%B8%9
9%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%88%E0%B8%AD%E0%B8%94%E0
%B8%A3%E0%B8%96.jpg?s=612x612&w=0&k=20&c=ZQ2S5y7c_mrybI0ZCDM6-
DQtL9qBOb62v9C07zJckEk=');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
</style>
</head>
<div class="d-flex justify-content-center align-items-center vh-100">
<div class="bg-white p-5 rounded-5 shadow">
<div class="text-center"><i class="fa-solid fa-circle-user fa-4x "></i></div>
<div class="text-center fs-2 fw-bold">ลงชื่อเข้าใช้งาน</div>
<div class="input-group mt-4">
<div class="input-group-text bg-info"><i class="fa-solid fa-user"></i> </div>
<input class="form-control bg-light" type="password" placeholder="ชื่อผู้ใช้งาน">
</div>
<div class="input-group mt-4">
<div class="input-group-text bg-info"><i class="fa-solid fa-lock"></i> </div>
<input class="form-control bg-light" type="text" placeholder="รหัสผ่าน">
```

```
</div>
<div class="text-right mt-3">
<a href="#">Recover Password</a></div>
<div class="btn btn-info text-white w-100 mt-4">ล็อกอิน<a href="
index_emergency.html.html"></a></div>
<div class="text-center mt-3"> <div class="not-member">
Not a member? <a href="singup.html">Register Now</a>
</div></div>
</div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
crossorigin="anonymous"></script>
</body>
</html>
```

คำสั่งในการค้นหาโรงพยาบาลที่มีแพทย์เฉพาะทางที่เหมาะสมกับอาการของผู้แจ้งเหตุฉุกเฉิน (hospital.php)

```
<?php
$host = "host=localhost"; //host ที่ใช้ในการติดต่อกับ Server
$port = "port=5432"; //หมายเลข port ที่ใช้ (บางเครื่องอาจจะใช้ 5433หรือเลขอื่น)
$dbname = "dbname=sensors";
$credentials = "user=postgres password=xxxxxxxxxx";
//โครงสร้างชุดคำสั่งทำสำหรับเชื่อมต่อกับฐานข้อมูล PosgresSQL
$db=pg_connect("$host $port $dbname $credentials");
$sql="SELECT *,ST_AsGeoJSON(ST_Transform(geom,3857)) as geojson from point_hos
;" ;
// คำสั่งประมวลผลระหว่าง PHP กับ SQL
```

```
$query=pg_query($db,$sql);
//บรรทัดที่ 17-35 เป็นคำสั่งการส่งค่า array ในรูปแบบ Geojson
$geojson=array(
'type' => 'FeatureCollection',
'features' => array()
);
while ($edge = pg_fetch_assoc($query)){
$feature = array(
'type'=> 'Feature',
'geometry' => json_decode($edge['geojson'],true),
'crs'=> array(
'type'=>'EPSG',
'properties'=> array('code'=>'3857')
),
'properties'=>array(
'gid'=>$edge['gid'],
'geom' => $edge['geom'],
'name' => $edge['name']
)
);
array_push($geojson['features'],$feature);
}
pg_close($db);
echo json_encode($geojson);
?>
```

คำสั่งในการค้นหาเส้นทางในการเดินทางจากตำแหน่งผู้แจ้งเหตุฉุกเฉินไปโรงพยาบาลแม่ฟ้าหลวง (maefahluang.php)

```
<?php
// Database connection settings
define("PG_DB"  , "sensors");
```

```
define("PG_HOST", "localhost");
define("PG_USER", "postgres");
define("PG_PORT", "5432");
define("PG_PASS", "xxxxxxxx");
define("TABLE",   "ways");

// คำสั่งรับค่าพิกัดจากจุด start
$start = explode(' ',$_REQUEST['startpoint']);
$startPoint = array($start[0], $start[1]);
$startEdge = findNearestEdge($startPoint);
//////////////////$endEdge    = findNearestEdge($endPoint);
// FUNCTION findNearestEdge
function findNearestEdge($lonlat) {
// Connect to database
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
$sql = "SELECT gid, source, target, geom,
ST_Distance(geom, ST_geometryFromText(
'POINT(".$lonlat[0]." ".$lonlat[1].")', 3857)) AS dist
FROM ".TABLE."
ORDER BY dist LIMIT 1";
$query = pg_query($con,$sql);
$edge['gid']       = pg_fetch_result($query, 0, 0);
$edge['source']    = pg_fetch_result($query, 0, 1);
$edge['target']    = pg_fetch_result($query, 0, 2);
$edge['geom']      = pg_fetch_result($query, 0, 3);
// echo "$sql"; //การใส่ echo เป็นการเช็คโค้ดแต่ละบรรทัดที่เราต้องการเช็ค การเช็คทุกครั้งต้อง
ปิด echo json_encode($geojson);
//  echo  $edge['source'] ;  echo "</br>";
//  echo  $edge['target'] ;  echo "</br>";
// echo  $edge['geom'] ;
```

```
// Close database connection
pg_close($con);
return $edge;
}
$sql = "
SELECT st_asgeojson(geom) as geojson from ".TABLE." w,(
SELECT *    FROM pgr_dijkstra(
'SELECT
gid AS id,
source, target,
length+ curve AS cost
FROM ".TABLE."',
915,
".$startEdge['target'].",
directed := false)) as rt where w.gid=rt.edge;
";
//echo "$sql";
// Connect to database
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
// Perform database query
$query = pg_query($con,$sql);
//echo $sql;
// Return route as GeoJSON
$geojson = array(
'type'       => 'FeatureCollection',
'features'   => array()
);
// Add edges to GeoJSON array
while($edge=pg_fetch_assoc($query)) {
$feature = array(
```

```
'type' => 'Feature',
'geometry' => json_decode($edge['geojson'], true),
'crs' => array(
'type' => 'EPSG',
'properties' => array('code' => '3857')
),
'properties' => array(
'gid' => $edge['id'],
'length' => $edge['length']
)
);
// Add feature array to feature collection array
array_push($geojson['features'], $feature);
}
// Close database connection
pg_close($con);
// Return routing result
header('Content-type: application/json',true);
echo json_encode($geojson);
?>
```

คำสั่งในการค้นหาเส้นทางในการเดินทางจากตำแหน่งผู้แจ้งเหตุฉุกเฉินไปโรงพยาบาลแม่จัน (maechan.php)

```
<?php
// Database connection settings
define("PG_DB"  , "sensors");
define("PG_HOST", "localhost");
define("PG_USER", "postgres");
define("PG_PORT", "5432");
define("PG_PASS", "xxxxxxxx");
define("TABLE",   "ways");
```

```
// คำสั่งรับค่าพิกัดจากจุด start
$start = explode(' ',$_REQUEST['startpoint']);
$startPoint = array($start[0], $start[1]);
$startEdge = findNearestEdge($startPoint);
function findNearestEdge($lonlat) {
// Connect to database
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
$sql = "SELECT gid, source, target, geom,
ST_Distance(geom, ST_geometryFromText(
'POINT(".$lonlat[0]." ".$lonlat[1].")', 3857)) AS dist
FROM ".TABLE."
ORDER BY dist LIMIT 1";
$query = pg_query($con,$sql);
$edge['gid']      = pg_fetch_result($query, 0, 0);
$edge['source']   = pg_fetch_result($query, 0, 1);
$edge['target']   = pg_fetch_result($query, 0, 2);
$edge['geom']     = pg_fetch_result($query, 0, 3);
// echo "$sql"; //การใส่ echo เป็นการเช็คโค้ดแต่ละบรรทัดที่เราต้องการเช็ค การเช็คทุกครั้งต้อง
ปิด echo json_encode($geojson);
//  echo  $edge['source'] ;  echo "</br>";
//  echo  $edge['target'] ;  echo "</br>";
// echo  $edge['geom'] ;
// Close database connection
pg_close($con);
return $edge;
}
$sql = "
SELECT st_asgeojson(geom) as geojson from ".TABLE." w,(
SELECT *   FROM pgr_dijkstra(
'SELECT
```

```
gid AS id,
source, target,
length+ curve AS cost
FROM ".TABLE."',
3803,
".$startEdge['target'].",
directed := false)) as rt where w.gid=rt.edge;
";
//echo "$sql";
// Connect to database
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
// Perform database query
$query = pg_query($con,$sql);
//echo $sql;
// Return route as GeoJSON
$geojson = array(
'type'      => 'FeatureCollection',
'features'  => array()
);
// Add edges to GeoJSON array
while($edge=pg_fetch_assoc($query)) {
$feature = array(
'type' => 'Feature',
'geometry' => json_decode($edge['geojson'], true),
'crs' => array(
'type' => 'EPSG',
'properties' => array('code' => '3857')
),
'properties' => array(
'gid' => $edge['id'],
```

```
'length' => $edge['length']
)
);
// Add feature array to feature collection array
array_push($geojson['features'], $feature);
}

// Close database connection
pg_close($con);
// Return routing result
header('Content-type: application/json',true);
echo json_encode($geojson);
?>
```

คำสั่งในการค้นหาเส้นทางในการเดินทางจากตำแหน่งผู้แจ้งเหตุฉุกเฉินไปในโรงพยาบาลส่งเสริมสุขภาพตำบลแม่สลองใน (maesalongnai.php)

```
<?php
// Database connection settings
define("PG_DB"  , "sensors");
define("PG_HOST", "localhost");
define("PG_USER", "postgres");
define("PG_PORT", "5432");
define("PG_PASS", "xxxxxxxxxx");
define("TABLE",   "ways");
// คำสั่งรับค่าพิกัดจากจุด start
$start = explode(' ',$_REQUEST['startpoint']);
$startPoint = array($start[0], $start[1]);
$startEdge = findNearestEdge($startPoint);
function findNearestEdge($lonlat) {
// Connect to database
```

```
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
$sql = "SELECT gid, source, target, geom,
ST_Distance(geom, ST_geometryFromText(
'POINT(".$lonlat[0]." ".$lonlat[1].")', 3857)) AS dist
FROM ".TABLE."
ORDER BY dist LIMIT 1";
$query = pg_query($con,$sql);
$edge['gid']       = pg_fetch_result($query, 0, 0);
$edge['source']    = pg_fetch_result($query, 0, 1);
$edge['target']    = pg_fetch_result($query, 0, 2);
$edge['geom']      = pg_fetch_result($query, 0, 3);
// echo "$sql"; //การใส่ echo เป็นการเช็คโค้ดแต่ละบรรทัดที่เราต้องการเช็ค การเช็คทุกครั้งต้อง
ปิด echo json_encode($geojson);
//  echo  $edge['source'] ;  echo "</br>";
//  echo  $edge['target'] ;  echo "</br>";
// echo  $edge['geom'] ;
// Close database connection
pg_close($con);
return $edge;
}
$sql = "
SELECT st_asgeojson(geom) as geojson from ".TABLE." w,(
SELECT *    FROM pgr_dijkstra(
'SELECT
gid AS id,
source, target,
length+ curve AS cost
FROM ".TABLE."',
914,
".$startEdge['target'].",
```

```
directed := false)) as rt where w.gid=rt.edge;
";
//echo "$sql";
// Connect to database
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
// Perform database query
$query = pg_query($con,$sql);
//echo $sql;
// Return route as GeoJSON
$geojson = array(
'type'      => 'FeatureCollection',
'features'  => array()
);
// Add edges to GeoJSON array
while($edge=pg_fetch_assoc($query)) {
$feature = array(
'type' => 'Feature',
'geometry' => json_decode($edge['geojson'], true),
'crs' => array(
'type' => 'EPSG',
'properties' => array('code' => '3857')
),
'properties' => array(
'gid' => $edge['id'],
'length' => $edge['length']
)
);
// Add feature array to feature collection array
array_push($geojson['features'], $feature);
}
```

```
// Close database connection
pg_close($con);
// Return routing result
header('Content-type: application/json',true);
echo json_encode($geojson);
?>
```

คำสั่งในการค้นหาเส้นทางในการเดินทางจากตำแหน่งผู้แจ้งเหตุฉุกเฉินไปในโรงพยาบาลส่งเสริมสุขภาพตำบลแม่สลองนอก (maesalongnok.php)

```
<?php
// Database connection settings
define("PG_DB"  , "sensors");
define("PG_HOST", "localhost");
define("PG_USER", "postgres");
define("PG_PORT", "5432");
define("PG_PASS", "xxxxxxxxx");
define("TABLE",   "ways");
// คำสั่งรับค่าพิกัดจากจุด start
$start = explode(' ',$_REQUEST['startpoint']);
$startPoint = array($start[0], $start[1]);
$startEdge = findNearestEdge($startPoint);
function findNearestEdge($lonlat) {
// Connect to database
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
$sql = "SELECT gid, source, target, geom,
ST_Distance(geom, ST_geometryFromText(
'POINT(".$lonlat[0]." ".$lonlat[1].")', 3857)) AS dist
FROM ".TABLE."
ORDER BY dist LIMIT 1";
$query = pg_query($con,$sql);
```

```
$edge['gid']       = pg_fetch_result($query, 0, 0);
$edge['source']    = pg_fetch_result($query, 0, 1);
$edge['target']    = pg_fetch_result($query, 0, 2);
$edge['geom']      = pg_fetch_result($query, 0, 3);
// echo "$sql"; //การใส่ echo เป็นการเช็คโค้ดแต่ละบรรทัดที่เราต้องการเช็ค การเช็คทุกครั้งต้อง
ปิด echo json_encode($geojson);
//  echo  $edge['source'] ;  echo "</br>";
//  echo  $edge['target'] ;  echo "</br>";
// echo  $edge['geom'] ;
// Close database connection
pg_close($con);
return $edge;
}
$sql = "
SELECT st_asgeojson(geom) as geojson from ".TABLE." w,(
SELECT *   FROM pgr_dijkstra(
'SELECT
gid AS id,
source, target,
length+ curve AS cost
FROM ".TABLE."',
3290,
".$startEdge['target'].",
directed := false)) as rt where w.gid=rt.edge;
";
//echo "$sql";
// Connect to database
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS."
user=".PG_USER);
// Perform database query
```

```
$query = pg_query($con,$sql);
//echo $sql;
// Return route as GeoJSON
$geojson = array(
'type'      => 'FeatureCollection',
'features'  => array()
);
while($edge=pg_fetch_assoc($query)) {
$feature = array(
'type' => 'Feature',
'geometry' => json_decode($edge['geojson'], true),
'crs' => array(
'type' => 'EPSG',
'properties' => array('code' => '3857')
),
'properties' => array(
'gid' => $edge['id'],
'length' => $edge['length']
));
// Add feature array to feature collection array
array_push($geojson['features'], $feature);
}
// Close database connection
pg_close($con);
// Return routing result
header('Content-type: application/json',true);
echo json_encode($geojson);
?>
```

คำสั่งในการค้นหาข้อมูลผู้แจ้งเหตุฉุกเฉิน (data_user.php)

```
<?php
$host = "host=localhost"; //host ที่ใช้ในการติดต่อกับ Server
```

```
$port = "port=5432"; //หมายเลข port ที่ใช้ (บางเครื่องอาจจะใช้ 5433หรือเลขอื่น)
$dbname = "dbname=sensors";
$credentials = "user=postgres password=xxxxxxxxxx";
//โครงสร้างชุดคำสั่งทำสำหรับเชื่อมต่อกับฐานข้อมูล PosgresSQL
$db=pg_connect("$host $port $dbname $credentials");
//คำสั่ง sql สำหรับเรียกข้อมูลขอบเขตจังหวัดจากตารางชื่อ thailand_data มาแสดง โดยทำการ
กำหนดมา 30 จังหวัด
$sql="SELECT *,ST_AsGeoJSON(ST_Transform(geom,3857)) as geojson from emergency
;" ;
$query=pg_query($db,$sql);
$geojson=array(
'type' => 'FeatureCollection',
'features' => array()
);
while ($edge = pg_fetch_assoc($query)){
$feature = array(
'type'=> 'Feature',
'geometry' => json_decode($edge['geojson'],true),
'crs'=> array(
'type'=>'EPSG',
'properties'=> array('code'=>'3857')
),
'properties'=>array(
'gid'=>$edge['gid'],
'geom' => $edge['geom'],
'id_number' => $edge['id_number'],
'name' => $edge['name'],
'sername' => $edge['sername'],
'house_number' => $edge['house_number'],
'moo' => $edge['moo'],
'tambon' => $edge['tambon'],
```

```
'amphoe' => $edge['amphoe'],
'province' => $edge['province'],
'post_code' => $edge['post_code'],
'number_phone' => $edge['number_phone'],
'id_card' => $edge['id_card'],
'health' => $edge['health'],
'allergy' => $edge['allergy']
));
array_push($geojson['features'],$feature);
}
pg_close($db);
echo json_encode($geojson);
?>
```