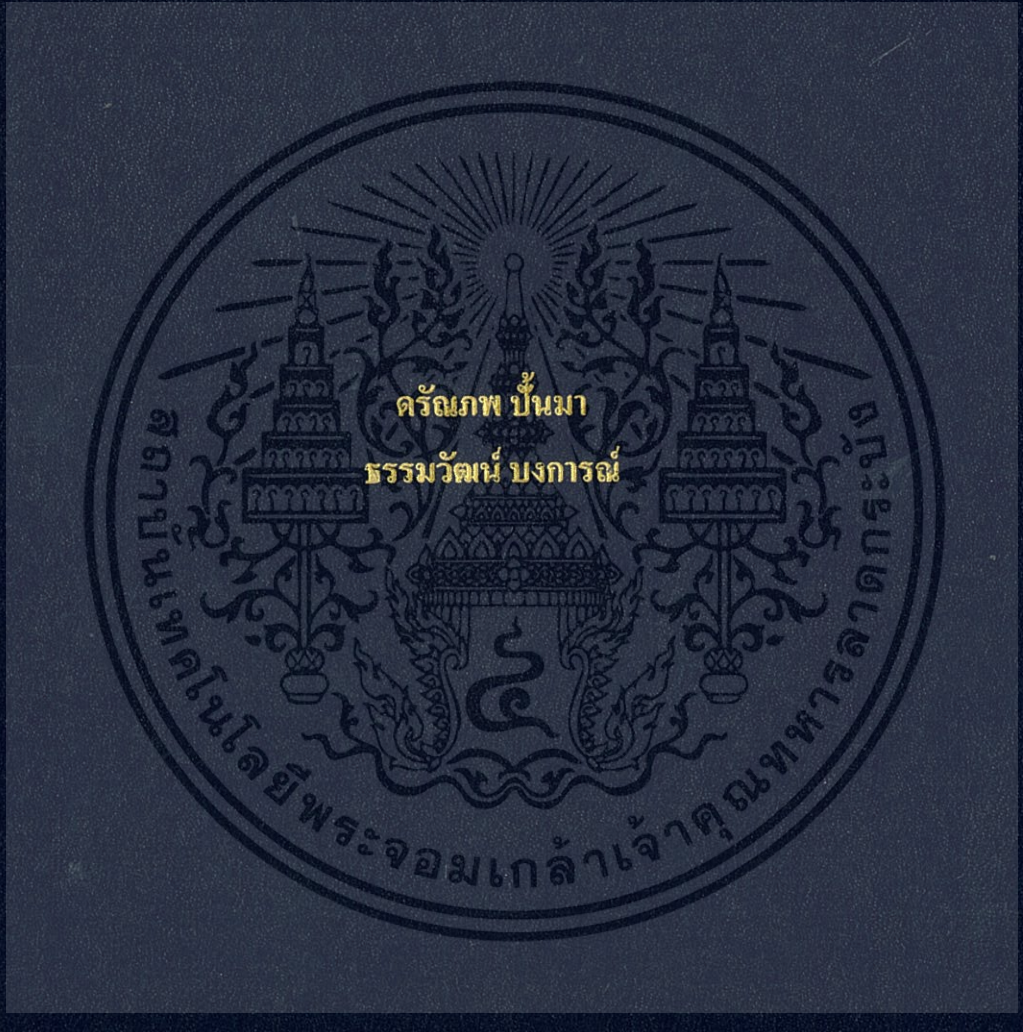
ระบบวิเคราะห์ข้อมูลจีพีเอสเพื่อการกำกับดูแลการจราจรของประเทศไทย

GPS ANALYSIS SYSTEM FOR THAILAND TRAFFIC MANAGEMENT



ดรัณภพ ปั้นมา

ธรรมวัฒน์ บงการณ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ระบบวิเคราะห์ข้อมูลจีพีเอสเพื่อการกำกับดูแลการจราจรของประเทศไทย

GPS ANALYSIS SYSTEM FOR THAILAND TRAFFIC MANAGEMENT

ดรัณภพ ปั้นมา

ธรรมวัฒน์ บงการณ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบวิเคราะห์ข้อมูลจีพีเอสเพื่อการกำกับดูแลการจราจรของประเทศไทย

GPS ANALYSIS SYSTEM FOR THAILAND TRAFFIC MANAGEMENT

ผู้จัดทำ

1. นายดรัณภพ ปั้นมา รหัสนักศึกษา 57010481

2. นายธรรมวัฒน์ บงการณ์ รหัสนักศึกษา 57010600

_____ อาจารย์ที่ปรึกษา

(ดร. ธนัญชัย ตรีภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบวิเคราะห์ข้อมูลจีพีเอสเพื่อการกำกับดูแลการจราจรของประเทศไทย

นายดรัณภพ	ปั้นมา	57010481
นายธรรมวัฒน์	บงการณ์	57010600
ดร.ธนัญชัย	ตรีภาค	อาจารย์ที่ปรึกษา

ปีการศึกษา 2560

บทคัดย่อ

ในปัจจุบัน การเดินรถของรถประจำทางสาธารณะในกรุงเทพมหานครมีการฝ่าฝืนระเบียบอยู่บ่อยครั้ง เช่น การขับออกนอกเส้นทางที่กำหนดไว้หรือการขับไม่ครบป้ายตามที่กำหนดไว้ เป็นต้น และถึงแม้ว่าจะมีเจ้าหน้าที่คอยตรวจสอบการฝ่าฝืนระเบียบเหล่านี้ การฝ่าฝืนระเบียบเหล่านี้ก็ไม่ได้ลดลงมากเท่าที่ควร เนื่องจากรถประจำทางสาธารณะในกรุงเทพนั้นมีอยู่จำนวนมาก ทำให้เจ้าหน้าที่ที่คอยตรวจสอบการฝ่าฝืนระเบียบของรถเหล่านี้ ไม่สามารถตรวจสอบได้อย่างทั่วถึง โครงการนี้จึงคิดค้นหาวิธีในการตรวจสอบการฝ่าฝืนระเบียบของรถประจำทางสาธารณะได้โดยอัตโนมัติโดยใช้ข้อมูลพิกัดจากระบบจีพีเอส เพื่อที่จะช่วยเจ้าหน้าที่ในการตรวจสอบได้อย่างทั่วถึง นอกจากนี้การแสดงผลตำแหน่งของรถด้วยเทคโนโลยีการแสดงผลที่นิยมใช้กันอยู่ในปัจจุบันยังไม่สามารถแสดงผลตำแหน่งรถจำนวนมากได้อย่างต่อเนื่อง โครงการนี้จึงมีการหาเทคโนโลยีใหม่ที่เหมาะสมในการแสดงผลข้อมูลจำนวนมาก ๆ บนเว็บบราวเซอร์ได้อย่างต่อเนื่องด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPS Analysis System For Thailand Traffic Management

Mr. Daranpob Punma 57010481

Mr. Thammawat Bongkarn 57010600

Dr. Thanunchai Threepak Advisor

Academic Year 2017

ABSTRACT

Nowadays, there are many breaking regulations bus driving in Bangkok like driving out of road or driving incomplete bus stop in the road path although department of land transport has officer who verify bus driving but the problem doesn't reduce as it should because there are many buses in Bangkok so officer can't verify thoroughly. This project will create method to verify breaking regulations bus driving automatically by using GPS data for helping officer in addition technology for rendering GPS data that using in present can't render big GPS data smoothly so this project will find new technology for rendering big GPS data smoothly too

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่โครงงานเกี่ยวกับระบบวิเคราะห์ข้อมูลจีพีเอสเพื่อการกำกับดูแลการจราจรของประเทศสามารถดำเนินผ่านมาได้ด้วยดีถึงจุดนี้ เพราะความกรุณาและความช่วยเหลืออย่างสูงจาก ดร.ธนัญชัย ตรีภาค ซึ่งเป็นอาจารย์ที่ปรึกษาของโครงงานนี้ที่ให้คำปรึกษาแนะนำและวางแผนการทำงานตลอดจนให้คำแนะนำทางทฤษฎีและแนวทางปฏิบัติ รวมทั้งสละเวลาอันมีค่าติดตามความคืบหน้าของโครงงานอย่างสม่ำเสมอ และช่วยแก้ไขข้อผิดพลาดของการทำงานด้วยความเอาใจใส่เป็นอย่างดี

ขอบพระคุณอาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ให้ความรู้ด้านทฤษฎีและการปฏิบัติในรายวิชาต่าง ๆ ที่ให้ความรู้กับการทำโครงงานนี้

สุดท้ายนี้ด้วยคุณค่าและประโยชน์อันพึงมีจากโครงงานนี้ ขอมอบแด่ผู้มีพระคุณทุกท่านที่ได้กล่าวมาข้างต้น และ ขอกราบขอบพระคุณทุกท่านมา ณ. ที่นี้

ดรัณภพ ปั้นมา

ธรรมวัฒน์ บงการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญรูป IX

บทที่ 1 บทนำ 1

1.1 ที่มาของโครงงาน 1

1.1.1 การขับรถแซงกันของรถประจำทางสายเดียวกัน 1

1.1.2 การขับรถออกนอกเส้นทาง 1

1.1.3 การขับไม่ครบป้ายตามที่กำหนดทุกป้าย 1

1.2 แนวคิดของโครงงาน 2

1.3 วัตถุประสงค์ของโครงงาน 2

1.4 เป้าหมายของโครงงาน 2

1.5 ขอบเขตของโครงงาน 2

1.6 วิธีการดำเนินงาน 3

1.7 สิ่งที่คาดว่าจะได้รับ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 React 5

2.2 Redux 6

2.3 Node.js 8

2.4 Express 9

2.5 Webpack 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.6 Google Maps React 10
2.7 Google Maps API Direction 13
2.8 MongoDB 14

บทที่ 3 การออกแบบและพัฒนา 16
3.1 การออกแบบส่วนต่าง ๆ ของระบบ 17
3.1.1 Frontend 17
3.1.2 Backend 19
3.1.3 Server 19
3.2 การออกแบบการใช้งานของ User 27
3.2.1 การใช้งานของ User ทั่วไป 27
3.2.2 การใช้งานของ เจ้าหน้าที่ 27
3.3 การออกแบบวิธีการในการตรวจสอบการกระทำความผิดของรถประจำทาง 27
3.3.1 ปัญหาการวิ่งออกนอกเส้นทาง 27
3.3.2 ปัญหาการวิ่งไม่ครบป้ายตามที่กำหนด 27
3.3.3 ปัญหาการขับแซงกันในสายเดียวกัน 28
3.4 การสร้างอัลกอริทึมตามสมมุติฐานที่ออกแบบไว้ 28
3.4.1 ภาพรวมของอัลกอริทึม 28
3.4.2 ขั้นตอนการเตรียมการก่อนการสร้างอัลกอริทึม 31
3.4.3 การตั้งค่าความสำคัญ (Priority) ให้รถประจำทางสาธารณะ 31
3.4.4 การตรวจสอบการขับออกนอกเส้นทาง 31
3.4.5 การเทียบตำแหน่งของรถเข้ากับลำดับในเส้นทางที่สร้างขึ้น 32
3.4.6 การตรวจสอบว่าข้อมูลของรถสามารถนำมาคำนวณได้หรือไม่ 33
3.4.7 การตรวจสอบการขับไม่ครบตามป้ายที่กำหนด 34
3.4.8 การตรวจสอบการขับแซงรถในสายเดียวกัน 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 4 การทดลองและผลการทดลอง 37

4.1 การทดลองเปรียบเทียบการ render GPS ของทั้ง Google Maps API และ React 37

4.1.1 การ render GPS โดยใช้ Google Maps API 37

4.1.2 การ render GPS โดยใช้ React 38

4.2 การเปรียบเทียบการใช้ทรัพยากรของเครื่องระหว่าง Google API และ React 40

4.2.1 CPU 40

4.2.2 Network (Sent Byte) 40

4.2.3 Network (Received Byte) 41

4.2.4 Memory 42

4.2.5 Rendering 43

4.3 การทดลองนำอัลกอริทึมทั้งสามอลักอริทึมมาใช้ในการตรวจสอบข้อมูลจริง 43

4.3.1 การตรวจสอบการขับรถออกนอกเส้นทาง 43

4.3.2 การตรวจสอบการขับไม่ครบตามป้ายที่กำหนด 43

4.3.3 การตรวจสอบการขับแซงกันของรถสายเดียวกัน 45

4.4 ฟีเจอร์ต่าง ๆ ในระบบ 45

4.4.1 การค้นหาเส้นทางที่ผ่านต้นทางและปลายทางตามที่ผู้ใช้ระบุ 45

4.4.2 การค้นหาสายรถประจำทางสาธารณะตามที่ผู้ใช้ระบุ 47

4.4.3 ระบบล็อกอิน 47

4.4.4 การดูภาพรวมของการกระทำผิดของรถประจำทางสาธารณะ 48

4.4.5 การดูประวัติการทำผิดของรถประจำทางสาธารณะ 48

4.4.6 การจัดการสมาชิกภายในระบบ (สำหรับเจ้าหน้าที่ที่ได้รับอนุญาตเท่านั้น) 49

4.4.7 การแก้ไขข้อมูลของผู้ใช้ 51

4.5 สรุปผลการทดลอง 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 5 บทสรุปและข้อเสนอแนะ 53
5.1 การสร้างแผนที่บนเว็บไซต์ระหว่าง Google Maps API และ Google Maps React 53
5.2 การใช้อัลกอริทึมที่สร้างขึ้นไปตรวจสอบการกระทำผิดของรถประจำทางสาธารณะ 53
5.3 ปัญหาอุปสรรคและแนวทางการแก้ไข 54
5.4 แนวทางการพัฒนาต่อ 54

ภาคผนวก 56
ก. โค้ดส่วนการทดลองเปรียบเทียบการ render GPS ของทั้ง Google API และ React 56
ก.1 Google API 56
ก.2 React 58
ข. รายละเอียดต่าง ๆ ของ API 60
ข.1 roadData 60
ข.2 user 62
ข.3 busGulity 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

3.1 รายละเอียดข้อมูลที่เก็บใน busOnRoad Database 19
3.2 รายละเอียดข้อมูลที่เก็บใน roadBusStop Database 21
3.3 รายละเอียดข้อมูลที่เก็บใน Roadmapbusses Database 22
3.4 รายละเอียดข้อมูลที่เก็บใน Busstopsequences Database 22
3.5 รายละเอียดข้อมูลที่เก็บใน BusGulity Database 23
3.6 รายละเอียดข้อมูลที่เก็บใน Path Database 24
3.7 รายละเอียดข้อมูลที่เก็บใน User Database 25
3.8 รายละเอียดข้อมูลที่เก็บใน Contract Database 26
ข.1 รายละเอียดของ API Request ต่าง ๆ ของ roadData 61
ข.2 รายละเอียดของ API Respond ต่าง ๆ ของ roadData 61
ข.3 รายละเอียดของ API Request ต่าง ๆ ของ user 63
ข.4 รายละเอียดของ API Respond ต่าง ๆ ของ user 65
ข.5 รายละเอียดของ API Request ต่าง ๆ ของ busGulity 67
ข.6 รายละเอียดของ API Respond ต่าง ๆ ของ busGulity 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป หน้า

2.1 เทคโนโลยีทั้งหมดที่ใช้ในระบบ 4

2.2 การทำงานของ state และ props ของ React 5

2.3 หลักการทำงานของ React และ Redux 6

2.4 หลักการทำงานของ redux 7

2.5 ความแตกต่างระหว่างการไม่ใช้ redux กับการใช้ redux 8

2.6 การทำงานของ Node.js 8

2.7 การทำงานของ Express 9

2.8 การทำงานของ Webpack 10

2.9 ตัวอย่างโค้ดการ import google-map-react 11

2.10 ตัวอย่างโค้ดการตั้งค่าต่าง ๆ ของ GoogleMapRect 11

2.11 การแสดงผลของ GoogleMapReact 12

2.12 ตัวอย่างการใส่ tag เพิ่มเติมใน GoogleMapReact 12

2.13 การแสดงผลหลังการเพิ่ม tag อื่น ๆ ใน GoogleMapReact 13

2.14 ข้อมูลที่ได้จากการใช้ Google Maps API Direction 14

2.15 Record ใน MongoDB โดยเปิดผ่านทาง Robo 3T 15

2.16 การเก็บข้อมูลในรูปแบบของ NoSQL และโครงสร้างข้อมูลเป็นแบบ JSON 15

3.1 องค์ประกอบโดยรวมของระบบ 17

3.2 โครงร่างหน้าเว็บไซต์ 18

3.3 โครงร่างหน้าเมนูต่าง ๆ 18

3.4 ภาพรวมการทำงานของอัลกอริทึม (ครึ่งแรก) 29

3.5 ภาพรวมการทำงานของอัลกอริทึม (ครึ่งหลัง) 30

3.6 การทำงานของอัลกอริทึมตรวจสอบการขับออกนอกเส้นทาง 32

3.7 การทำงานของการเทียบตำแหน่งรถประจำทางสาธารณะเข้ากับลำดับในเส้นทางที่สร้างขึ้น 33

3.8 การทำงานของการตรวจสอบความใช้ได้จริงของข้อมูล 34

3.9 การทำงานของอัลกอริทึมตรวจสอบการขับไม่ครบตามป้ายที่กำหนด 35

3.10 การทำงานของอัลกอริทึมตรวจสอบการแซงรถในสายเดียวกัน 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป หน้า

4.1 การ render GPS 10,000 จุดของ Google API 38
4.2 การ render GPS 10,000 จุด เมื่อมีการบวกค่า GPS ของ Google API 38
4.3 การ render GPS 10,000 จุด โดยใช้ React 39
4.4 การ render GPS 10,000 จุดของ React เมื่อมีการบวกค่า GPS 39
4.5 กราฟเปรียบเทียบ CPU Usage 40
4.6 กราฟเปรียบเทียบ Network (Sent Byte) 41
4.7 กราฟเปรียบเทียบ Network (Received Byte) 42
4.8 กราฟเปรียบเทียบ Memory Usage 42
4.9 ตัวอย่างการตรวจสอบการขับออกนอกเส้นทางของรถประจำทางสาธารณะ 43
4.10 การตรวจสอบการขับไม่ครบตามป้ายที่กำหนด 44
4.11 การกลับรถก่อนจุดที่ควรจะกลับรถ 44
4.12 การตรวจสอบการขับแซงกันของรถในสายเดียวกัน 45
4.13 แบบฟอร์มการกรอกข้อมูลเพื่อค้นหาเส้นทาง 46
4.14 ผลลัพธ์การค้นหาของการค้นหาเส้นทาง 46
4.15 เส้นทางของผลลัพธ์การค้นหาที่เลือก 46
4.16 แบบฟอร์มการค้นหาสายรถเมล์ 47
4.17 เส้นทางของสายรถเมล์ที่เลือก 47
4.18 การล็อกอินเข้าสู่ระบบ 48
4.19 ภาพรวมรายละเอียดทางสถิติของการฝ่าฝืนระเบียบ 48
4.20 รายงานความผิดทั้งหมดตามที่ผู้ใช้ต้องการ 49
4.21 ตัวอย่างรายละเอียดการฝ่าฝืนระเบียบ 49
4.22 การจัดการสมาชิกทั้งหมดของสมาชิกทั่วไป 50
4.23 การจัดการสมาชิกของผู้ดูแลหรือผู้ช่วยผู้ดูแล 50
4.24 แบบฟอร์มการเพิ่มสมาชิกใหม่ 50
4.25 แบบฟอร์มการแก้ไขข้อมูลสมาชิก 51
4.26 รายละเอียดข้อมูลของผู้ใช้ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
4.27 การแก้ไขรายละเอียดต่าง ๆ ของผู้ใช้	52
4.28 การตั้งรหัสผ่านใหม่ของผู้ใช้	52
ก.1 โค้ดการใช้ Google API	56
ก.2 โค้ดการ render GPS ของ Google API	57
ก.3 โค้ดการเพิ่มค่า GPS และการลบจุด GPS เก่า ของ Google API	57
ก.4 โค้ดการเพิ่มค่า GPS และสร้างจุด GPS ใหม่ ทุก ๆ 5 วินาที ของ Google API	57
ก.5 โค้ดการเก็บค่าการสุ่ม GPS ของ React	58
ก.6 โครงสร้างภายในฟังก์ชัน renderGPS	58
ก.7 โค้ดการนำฟังก์ชัน renderGPS มาใช้กับการเปลี่ยนแปลงของ State.dataPlot	59
ก.8 โครงสร้างภายในของฟังก์ชัน addGps	59
ก.9 การเรียกใช้ฟังก์ชัน addGps ทุก ๆ 5 วินาที	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

ในปัจจุบัน ระบบการขนส่งมวลชนด้วยรถประจำทางสาธารณะในกรุงเทพมหานคร มีปัญหาเกิดขึ้นหลายอย่างที่ทำให้ผู้ที่จะใช้บริการรถประจำทางสาธารณะต้องรอรถประจำทางสาธารณะนานกว่าที่ควรจะเป็นโดยไม่รู้ว่ารถประจำทางสาธารณะที่ตนเองกำลังรอนั้นอยู่ที่ไหน ซึ่งสาเหตุหลักที่ทำให้เกิดปัญหาส่งผลกระทบกับผู้ใช้งานคือการฝ่าฝืนระเบียบการเดินรถเช่นการวิ่งไม่ครบป้ายหรือการวิ่งออกนอกเส้นทาง โครงงานนี้จึงจะสร้างระบบเพื่อตรวจสอบการกระทำความผิด 3 ข้อที่เห็นได้บ่อยครั้งในการเดินทางของรถประจำทางสาธารณะมาตรวจสอบเพื่อลดปัญหาเหล่านี้ให้เหลือน้อยที่สุดเท่าที่จะเป็นไปได้ ซึ่งปัญหามีดังต่อไปนี้

1.1.1 การขับรถแซงกันของรถประจำทางสายเดียวกัน

ในบางครั้ง รถประจำทางสาธารณะจะเดินรถออกจากรอบการเดินรถที่กำหนด แล้วเดินรถไปแซงประจำทางสาธารณะในสายเดียวกันในรอบการเดินรถก่อนหน้า เพื่อที่จะสามารถรับผู้โดยสารให้ได้จำนวนมาก จึงเป็นสาเหตุให้ในบางครั้งรถประจำทางสายเดียวกันรับส่งผู้โดยสารอยู่ในช่วงเวลาเดียวกันหรือใกล้เคียงกัน และทำให้การเดินรถในสายนั้นขาดช่วงเป็นเวลานานเนื่องจากรอบการเดินรถขาดหายไปช่วงหนึ่ง

1.1.2 การขับรถออกนอกเส้นทาง

ในบางครั้งที่รถประจำทางสาธารณะขับออกนอกเส้นทางที่กำหนดเพราะจำเป็นต้องไปทำธุระส่วนตัวหรือเหตุจำเป็นใด ๆ ทำให้ผู้ที่จะใช้บริการต้องรอรถนานกว่าปกติ และไม่สามารถรู้ได้ว่าเมื่อใดรถประจำทางสาธารณะจะมาถึง

1.1.3 การขับไม่ครบป้ายตามที่กำหนดทุกป้าย

ในบางครั้งรถประจำทางสาธารณะรู้ว่าไม่มีคนจะลงในป้ายถัด ๆไป และขับลัดเส้นทางข้ามป้ายที่ไม่มีคนลงไป จึงทำให้ผู้ที่จะใช้บริการที่รออยู่ในป้ายที่ถูกข้ามไปนั้นต้องเสียเวลารอรถประจำทางสาธารณะรอบถัดไป

อีกทั้งยังมีปัญหาเกี่ยวกับการแสดงผลข้อมูล GPS ของรถประจำทางสาธารณะแบบ Real-time ที่ปกติจะสร้างภาระให้กับฝั่ง Server ทำให้ Server ต้องทำงานหนักมากเกินไป เป็นเหตุให้การแสดงผลเกิดความล่าช้า นอกจากนี้ที่ฝั่ง Client ยังไม่สามารถแสดงผลแผนที่ที่มี Marker ตำแหน่งเป็นจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ เนื่องจากต้องใช้เวลาประมวลผลและโหลดข้อมูลเป็นเวลานาน จึงมีแนวคิดที่จะประยุกต์ใช้เทคโนโลยีที่ทำให้สามารถประมวลผลเพื่อแสดงผลที่ Web Browser เพื่อลดการทำงานที่ฝั่ง Server ลง

1.2 แนวคิดของโครงงาน

โครงงานนี้ เป็นโครงงานที่จะแสดงถึงตำแหน่งรถประจำทางสาธารณะแบบ Real-time โดยการนำข้อมูล GPS ของรถประจำทางสาธารณะบางส่วนที่ได้มาจากองค์การขนส่งมวลชนกรุงเทพ (ขสมก.) มาพล็อตกับแผนที่ความละเอียดสูงบนหน้าเว็บไซต์ที่เขียนขึ้นเองและอัพเดตข้อมูลเองตลอดเวลา รวมถึงการนำเทคโนโลยี React มาช่วยในการแสดงผลบนฝั่ง Client เพื่อลดภาระการทำงานของฝั่ง Server และยังสามารถตรวจสอบหรือระบุปัญหาของรถประจำทางสาธารณะที่เกิดขึ้นในช่วงเวลาในแต่ละวันได้อย่างถูกต้องแล้วจัดทำเก็บไว้เป็นรายงานสถิติของปัญหาที่เกิดขึ้นในแต่ละวันไว้เพื่อที่จะนำข้อมูลในรายงานไปใช้ในการปรับปรุงและแก้ไขปัญหา

1.3 วัตถุประสงค์ของโครงงาน

1) แก้ไขการไม่ทราบตำแหน่งที่ชัดเจนของรถประจำทางสาธารณะ
2) เก็บรวบรวมข้อมูลของรถประจำทางสาธารณะที่ไม่ปฏิบัติตามข้อกำหนดการเดินรถ แล้วสามารถนำข้อมูลที่ได้มาใช้เพื่อแก้ไขปัญหาดังกล่าวได้

1.4 เป้าหมายของโครงงาน

1) แสดงตำแหน่งของรถประจำทางสาธารณะแบบ Real-time process
2) สามารถระบุได้ว่ารถประจำทางสาธารณะคันใดที่ขับแซงรถในสายเดียวกัน ขับออกนอกเส้นทาง และขับไม่ครบทุกป้ายตามที่กำหนด

1.5 ขอบเขตของโครงงาน

1) แสดงตำแหน่งรถประจำทางสาธารณะได้เฉพาะรถในกรุงเทพ
2) ตรวจสอบการฝ่าฝืนระเบียบได้เฉพาะสาย 39 สาย 63 และสาย 97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 วิธีการดำเนินงาน

1) กำหนดขอบเขตของข้อมูลที่จะนำมาทำโครงงาน
2) ศึกษาเทคโนโลยีที่นำมาใช้ในการแก้ไขปัญหา ได้แก่ React Framework เพราะเป็นเทคโนโลยีเว็บไซต์ที่ช่วยให้สามารถแสดงผลข้อมูลจำนวนมากได้อย่างรวดเร็ว (Real-time process) และมี library มากมายให้เลือกใช้เพื่อเพิ่มความสะดวกในการพัฒนาเว็บไซต์
3) ออกแบบและพัฒนาเว็บไซต์ตามขอบเขตที่กำหนดไว้
4) ทดสอบระบบโดยการนำข้อมูลรถประจำทางสาธารณะบางส่วนมาทดลองใช้กับเว็บไซต์
5) ปรับปรุงและแก้ไขเว็บไซต์

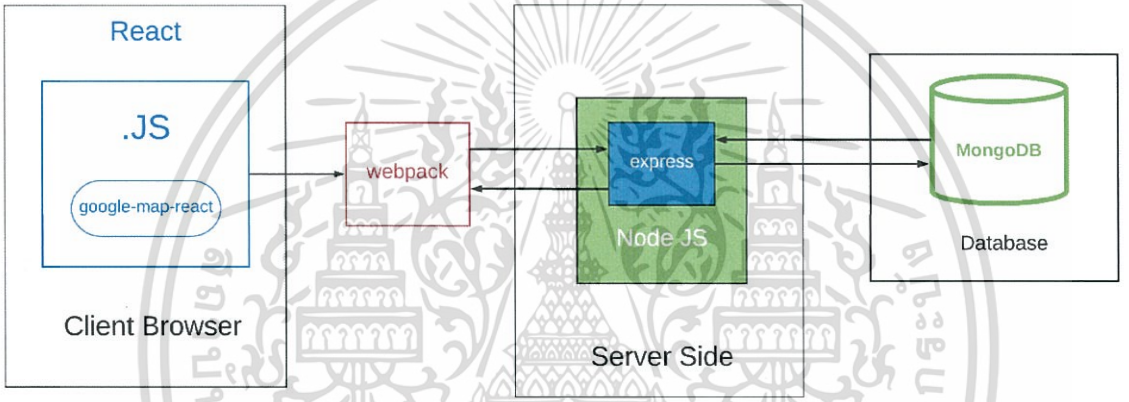
1.7 สิ่งที่คาดว่าจะได้รับ

1) องค์ความรู้ที่ได้จากการศึกษาการทำเว็บไซต์ที่ใช้การประมวลผลผ่าน Client เพื่อลดภาระในฝั่ง Server และการทำเว็บไซต์ ให้เป็น Real-time process ตลอดเวลา
2) การศึกษาเทคโนโลยีเว็บ React ซึ่งเป็นเทคโนโลยีใหม่
3) การลดปัญหาของรถประจำทางสาธารณะที่เกิดขึ้นในกรุงเทพ
4) ผู้ใช้รถประจำทาง สามารถดูตำแหน่งของรถประจำทางสาธารณะได้ง่ายและถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในการพัฒนาระบบวิเคราะห์ข้อมูล GPS เพื่อการกำกับดูแลการจราจรของประเทศไทย (GPS Analysis System For Thailand Traffic Management) มีการนำเทคโนโลยีต่าง ๆ มาใช้งานร่วมกัน เพื่อประสานการทำงานของส่วนประกอบต่าง ๆ ให้สอดคล้องกัน และได้ผลลัพธ์ตามความต้องการ ซึ่งเทคโนโลยีต่าง ๆ ดังรูปที่ 2.1 โดยมีรายละเอียดแต่ละส่วนตามหัวข้อด้านล่างนี้



รูปที่ 2.1 เทคโนโลยีทั้งหมดที่ใช้ในระบบ

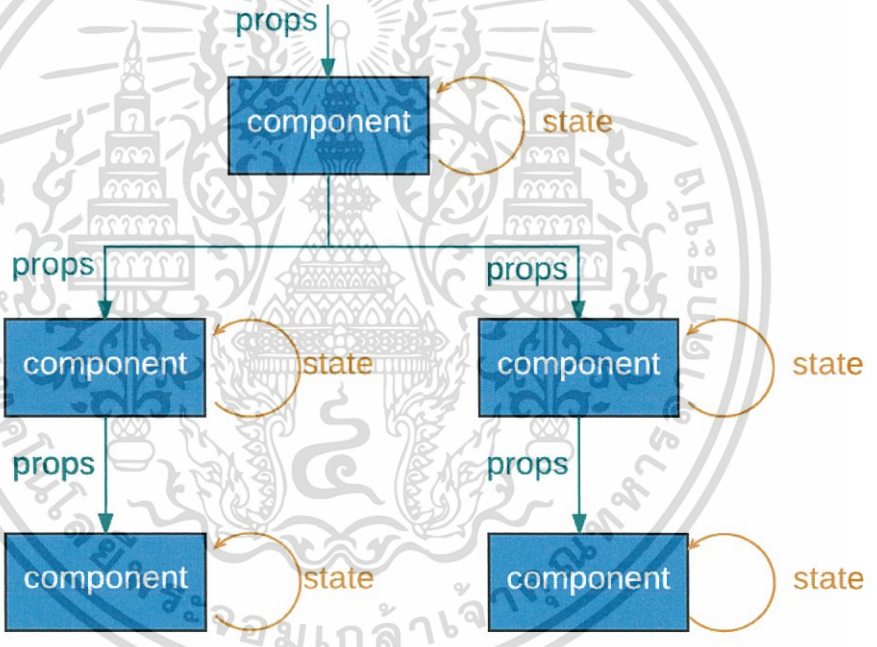
จากรูป ระบบจะแบ่งการทำงานออกเป็น 3 ส่วน ดังนี้

1) Client Browser ใช้ React เป็น Framework หลัก โดย React จะมี library google-maps-react ที่ใช้สำหรับการสร้างแผนที่บนเว็บไซต์ก่อนจะเชื่อมไปยัง Server และไฟล์บน Client Browser ทั้งหมดจะถูกรวมให้เหลือจำนวนไฟล์น้อยลงโดย Webpack
2) Server Side ใช้ Node.js เป็น Framework หลัก โดย Node.js จะมี Express ที่เป็นตัวจัดการ API จาก Client Browser และเป็นตัวกลางที่เชื่อมฝั่ง Client (ไฟล์ที่ผ่านการรวมของ Webpack) กับฝั่ง Server
3) Database ใช้ MongoDB เป็นตัวจัดการเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 React

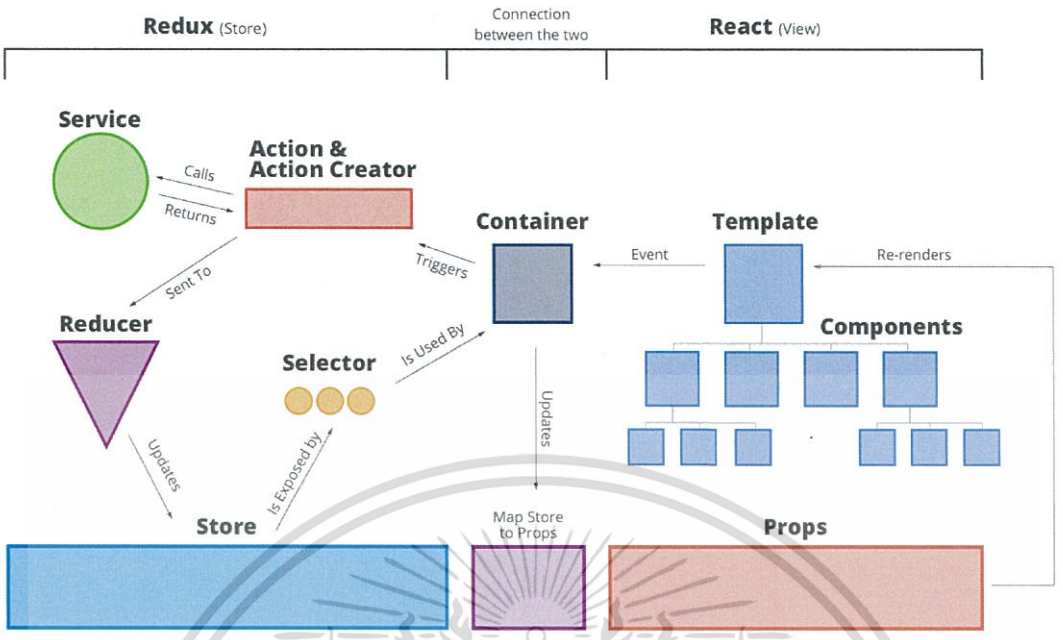
React เป็น JavaScript library ที่ทีม developer ของ Facebook พัฒนาขึ้น โดยปัจจุบัน React เป็นที่นิยมมากในการเขียนและพัฒนาเว็บไซต์ โดยใน React จะเขียนส่วนต่าง ๆ ของ Front-end ให้อยู่ในรูปขององค์ประกอบ (Component) ซึ่งแต่ละองค์ประกอบจะแยกจากกันและสามารถเรียกใช้ได้ตลอดเวลา นอกจากนี้ในแต่ละ Component ยังมี State ที่สามารถใช้ในการเก็บค่าตัวแปรต่าง ๆ และควบคุมการเปลี่ยนแปลงในแต่ละหน้า Front-end ตามที่เราต้องการ รวมถึงหาก State มีการเปลี่ยนแปลงค่า React จะ render หน้าเว็บไซต์ให้ตามการเปลี่ยนแปลงของ State โดยที่ไม่จำเป็นต้อง refresh หน้าเว็บไซต์อีกด้วย โดยในแต่ละ component สามารถส่งค่า state หากันได้โดยค่าที่ส่งไปจะถูกเรียกว่า props ตามรูปที่ 2.2 นอกจากนี้ React ยังมี Redux ที่เป็นตัวคอยจัดการและควบคุมค่า State กลาง ของ React ที่ทำให้ง่ายต่อการพัฒนามากขึ้น โดยมีการทำงานตามรูปที่ 2.3 ดังนี้



รูปที่ 2.2 การทำงานของ state และ props ของ React

(ที่มา: https://www.thitiblog.com/blog/8281)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 หลักการทำงานของ React และ Redux

(ที่มา: https://articles.coltpini.com/react-redux-architecture-overview-7b3e52004b6e)

การทำงานร่วมกันของ react และ redux โดยจากรูปจะเห็นเป็น 2 ส่วนคือ React และ Redux โดยจะมีส่วนเชื่อมกันคือ connection หน้าที่ของ React คือทำหน้าที่ในการควบคุมเรื่องการ render หน้า view (template) และจัดการรวมส่วน components ส่วน Redux จะทำหน้าที่ในการควบคุมค่า state กลาง เมื่อมี การ Action (ความต้องการเปลี่ยนแปลงค่าของ state กลางใน store โดย store คือ ตัวแปรที่ใช้เก็บค่า state กลาง) ซึ่งส่งมาจากฝั่ง react จะมีตัว reducer ที่คอยส่งค่าไป update state กลางใน store แล้วฝั่ง react ก็จะสามารถเรียกใช้ ค่า state กลาง ที่อยู่ใน store ในรูปแบบ ของ props ได้

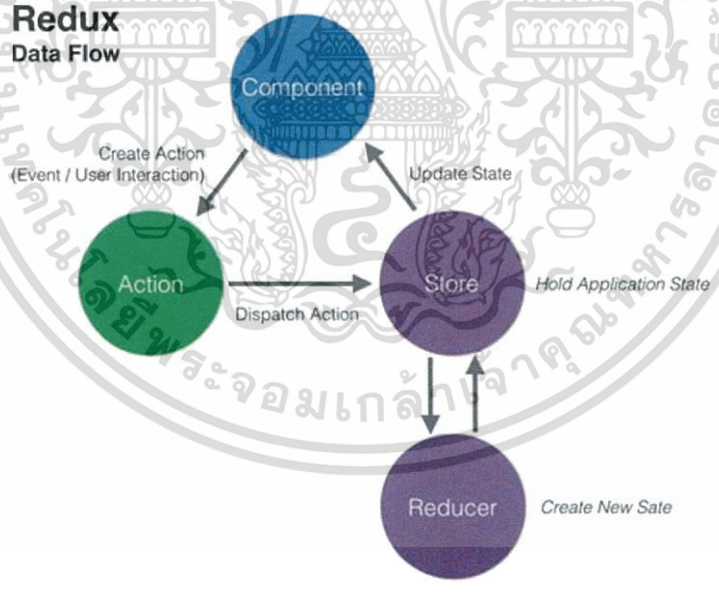
2.2 Redux

Redux เป็น library ที่ช่วยในการจัดการกับ state ของ react เนื่องจาก state ใน component ของ react จะมีการส่งค่าหากันเป็น flow เส้นตรง จึงเกิดปัญหาขึ้นว่า ถ้าหาก component ที่อยู่คนละ flow กัน ต้องการที่จะเรียกใช้หรือแก้ไขค่า state ที่เหมือนกัน ทำให้ไม่สามารถที่จะส่งค่า state นั้น ไปให้ component ที่อยู่คนละ flow การทำงานตรง ๆ ได้ ทำให้ต้องเขียนเชื่อม component ข้าม flow หรือ เขียน state นั้น ๆ ขึ้นมาใหม่ในแต่ละ component เป็นผลให้เกิดความซ้ำซ้อนของ code และยากในการ debug

หลักการทำงานของ redux ก็คือ การเก็บค่า state ตัวกลางไว้ใน store โดย component ของ react สามารถเรียกใช้หรือแก้ไขค่า state ตัวกลางใน store ได้ ดังรูปที่ 2.4 จะเห็นว่า component สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

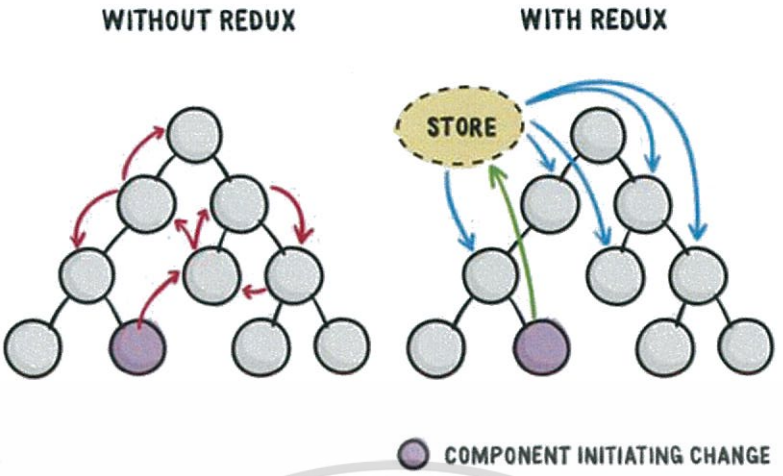
เรียกใช้ค่า state กลางจาก store ได้ และ เมื่อ component ต้องการแก้ไขค่า state กลางสามารถทำได้ ผ่าน action (คำสั่งจาก component ที่ระบุว่าต้องการเรียกใช้หรือเปลี่ยนแปลงค่าใน store) โดย store เมื่อได้รับ action จาก component แล้วจะใช้ reducer ซึ่งเป็นตัวกลางที่ใช้ในการเปลี่ยน state ใน store และเมื่อ reducer เปลี่ยนค่า state ใน store เสร็จเรียบร้อยแล้ว store จะทำการส่งค่า state กลางที่ component ต้องการไปให้ component ทำให้ถึงแม้ว่าจะเป็น component ที่อยู่คนละ flow การทำงานกัน ก็สามารถเรียกใช้หรือแก้ไข state แบบเดียวกันได้ จึงช่วยลดปัญหาความซ้ำซ้อนของ code ลงและยังง่ายต่อการพัฒนาเว็บไซต์ของผู้พัฒนาอีกด้วย ดังรูปที่ 2.5 โดยวงกลมต่าง ๆ คือ component และวงกลมสีม่วงคือ component ที่มีค่า state กลางอยู่ โดยจากภาพด้านซ้ายของรูป (without redux) จะเห็นว่าหากไม่ใช้ redux เมื่อ component สีม่วงมีการเปลี่ยนแปลง state กลางเกิดขึ้น จะต้องเขียน code คำสั่งเพื่อไปแก้ไขค่า state กลางนี้ในหน้า component อื่น ๆ (ตามลูกศรสีแดง) ซึ่งทำให้เกิดความซ้ำซ้อนของ code แต่หากมองดูในภาพด้านขวาของรูป (with redux) จะเห็นว่าหากมีการใช้ redux เข้ามาช่วย เมื่อ component สีม่วงมีการเปลี่ยนแปลงค่า state กลาง จะมีการแก้ไข ค่า state กลางใน store แล้ว component อื่น ๆ จะได้รับค่า state กลางที่มีการแก้ไขแล้วทันทีผ่าน store



รูปที่ 2.4 หลักการทำงานของ redux

(ที่มา: https://codingthesmartway.com/learn-redux-introduction-to-state-management-with-react)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

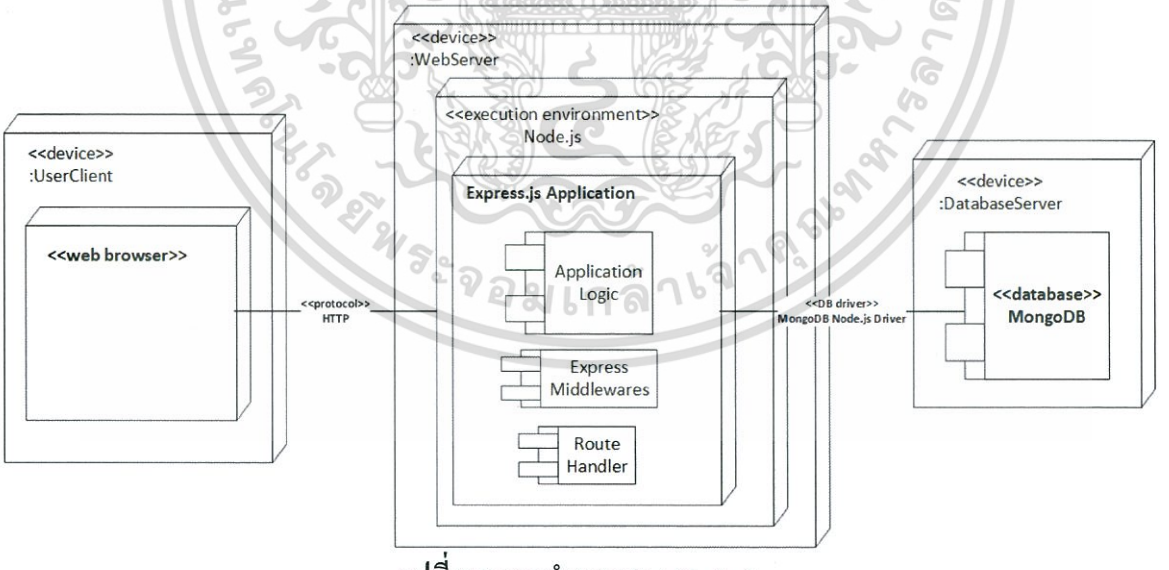


รูปที่ 2.5 ความแตกต่างระหว่างการไม่ใช้ redux กับการใช้ redux

(ที่มา: https://css-tricks.com/learning-react-redux)

2.3 Node.js

Node.js เป็น Framework ฝั่ง Server ที่เขียนโดยใช้ JavaScript โดยจะมี library หรือ package ภายในให้เลือกใช้ในการพัฒนา Server มากมายจึงเป็นที่นิยมมากในปัจจุบัน โดยหลัก ๆ แล้ว Node.js จะทำหน้าที่เป็น Webserver แล้วหน้าที่เป็นตัวกลางในการเชื่อมกันระหว่าง client side และ database



รูปที่ 2.6 การทำงานของ Node.js

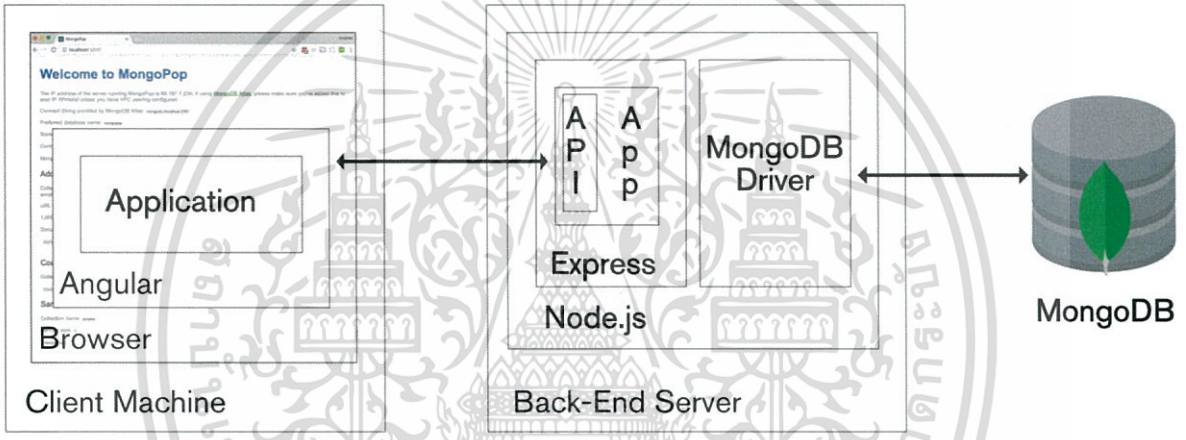
(ที่มา: rdxcgvjhbknkml)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.6 จะเห็นว่า กล่องทางฝั่งซ้าย (Client Side) จะเชื่อมกับ node.js ซึ่งเป็น webserver ผ่านโปรโตคอล HTTP แล้ว node.js จะเชื่อมกับ database ผ่าน Driver (package) ของ node.js

2.4 Express

Express เป็น Framework ที่ใช้สำหรับการเขียน Application ด้วย Node.js โดย Express จะถูกใช้เป็นตัว API Server และทำ Restful API รวมถึงช่วยในการติดต่อกับ Database อีกด้วย โดยมีการทำงานตามรูป ที่ 2.7 ดังนี้



รูปที่ 2.7 การทำงานของ Express

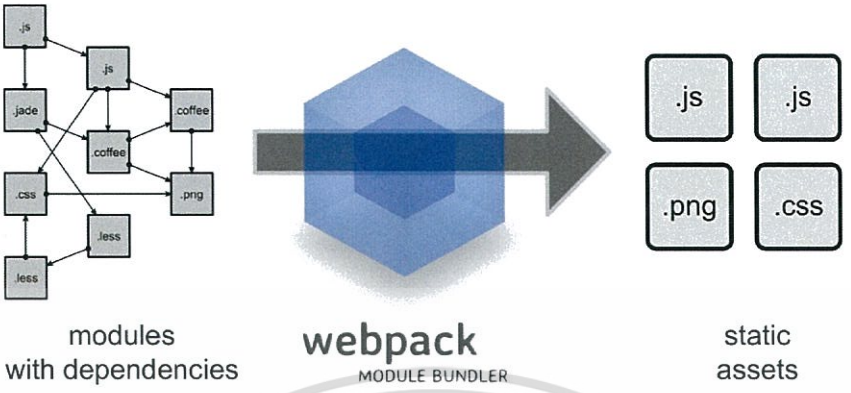
(ที่มา: http://www.clusterdb.com/tag/mern)

จากรูปที่ 2.7 จะเห็นว่า Express เป็นตัวกลางคอยจัดการ API ที่ส่งมาจาก Client และการบันทึกข้อมูลหรือเรียกใช้ข้อมูลของ Database

2.5 Webpack

Webpack เป็น module bundler สำหรับ JavaScript Applications โดยมีหน้าที่ในการรวมไฟล์ resource ที่ใช้ในการพัฒนา JavaScript Applications เช่นไฟล์ .js, .css, .scss หรือประเภทรูปภาพ (.png, .jpg) ให้อยู่ในรูปของไฟล์ที่มีจำนวนน้อยลง โดยมีการทำงานตามรูปที่ 2.8 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การทำงานของ Webpack

(ที่มา: https://medium.com/@Jasraj/dealing-with-webpack-2618f5f8e36a)

จากรูปที่ 2.8 จะเห็นว่า เมื่อนำไฟล์ต่าง ๆ จำนวนหนึ่งมารวมผ่าน Webpack แล้ว จะทำให้ได้ไฟล์ซึ่งเกิดจากการรวมไฟล์ทั้งหมดเพียงไม่กี่ไฟล์ แล้วไฟล์ใหม่ที่เกิดจากการรวมของ Webpack จะถือเป็นตัวแทนของไฟล์ทั้งหมดก่อนการย่อย ทำให้ง่ายต่อการนำไปใช้งานเพราะเพียงแค่เรียกใช้ไฟล์ ๆ เดียวก็สามารถใช้เป็นตัวแทนไฟล์หลาย ๆ ไฟล์ได้ โดยเราจะเรียกใช้ไฟล์ที่ได้จากการรวมไฟล์ของ webpack ตอน webserver เริ่มทำงาน โดยจะให้ webserver อ่านไฟล์ที่ได้รับจากการรวมไฟล์ของ webpack เพียงไฟล์เดียวซึ่งเป็นตัวแทนของไฟล์ JavaScript Applications ทั้งหมด

2.6 Google Maps React

Google Map React เป็น library ที่มี Google Maps API จำนวนหนึ่งที่ให้ React เรียกใช้งานได้ และทำให้ React component แสดงผลบน Google Maps ได้ โดยมีขั้นตอนการติดตั้งและการทำงาน ดังนี้

1) ทำการโหลด package google-map-react

yarn add google-map-react

2) ทำการ import google-map-react มาใช้ในหน้า Frontend ตามรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import React, { Component } from 'react'
import { compose, withProps, lifecycle } from "recompose";
import { XYPlot, XAxis, YAxis, HorizontalGridLines, LineSeries, MarkSeries, FlexibleXYPlot
import * as firebase from 'firebase';

import GoogleMapReact from 'google-map-react';
import busStop from './icon.png'
import Polyline from './polyLine'
import busIcon from './front-bus.png'
import axios from 'axios';
import https from 'https';
```

รูปที่ 2.9 ตัวอย่างโค้ดการ import google-map-react

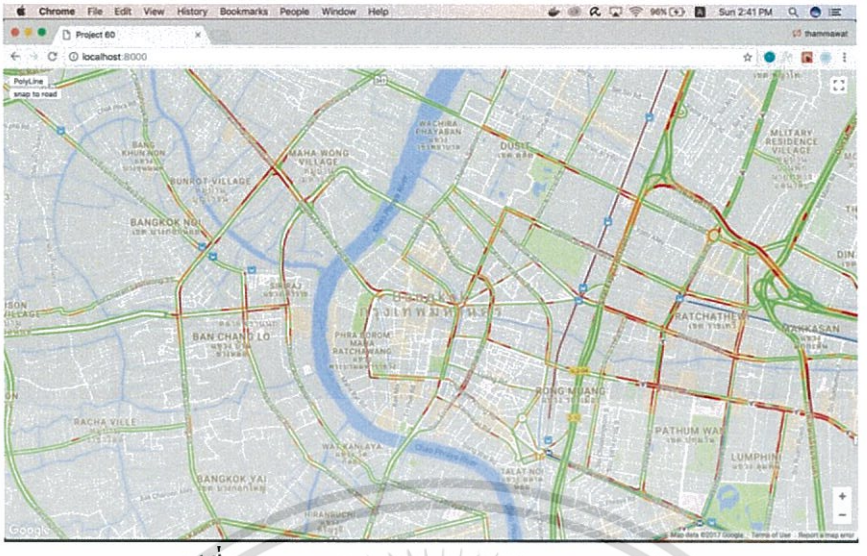
3) ทำการ ใส่ค่า tag GoogleMapReact และใส่ค่า attribute เบื้องต้นต่าง ๆ ได้แก่ การใส่ค่า center เป็นตำแหน่งพิกัดของกรุงเทพ ซึ่งก็คือ latitude 13.7563 และ longitude 100.5018 การใส่ค่า zoom เป็น 14 และการใส่ค่า layerTypes เป็น TrafficLayer เพื่อที่จะได้ทำให้ผู้ใช้เห็นความหนาแน่นของการจราจร ตามรูปที่ 2.10

```
<GoogleMapReact
  defaultCenter={this.state.center}
  defaultZoom={this.state.zoom}
  layerTypes={['TrafficLayer']}
  onGoogleApiLoaded={({ map, maps }) => { this.setState({ map: map, maps: maps, mapLc
  yesIWantToUseGoogleMapApiInternals
  style={{ zIndex: -1 }}
>
</GoogleMapReact>
```

รูปที่ 2.10 ตัวอย่างโค้ดการตั้งค่าต่าง ๆ ของ GoogleMapRect

4) ผลที่ได้คือ Google Maps ที่มีจุดศูนย์กลางคือ latitude 13.7563 , longitude 100.5018 และมี การ Zoom อยู่ที่ 14 และเห็นความหนาแน่นของการจราจรบนถนน ณ เวลานั้น ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การแสดงผลของ GoogleMapReact

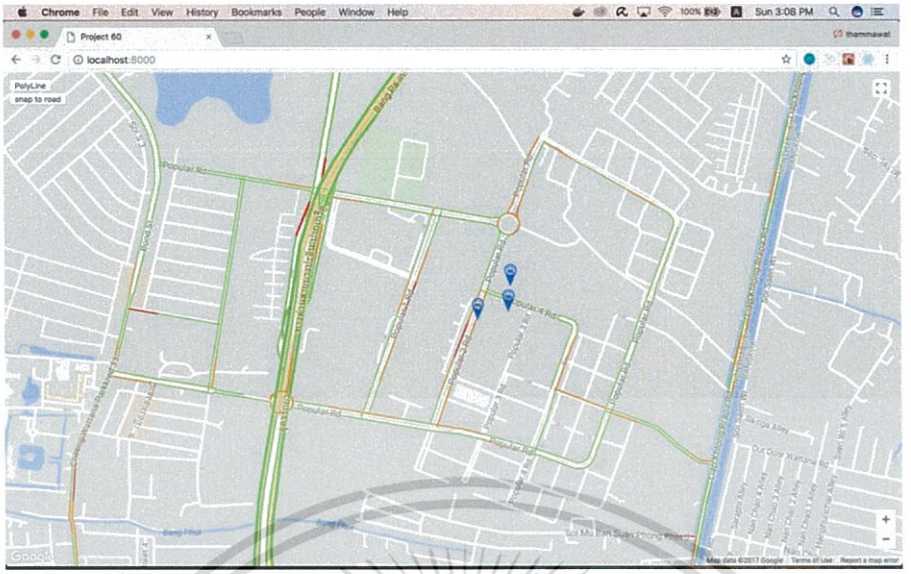
5) สามารถใส่ tag ต่าง ๆ ของ html ภายใน tag GoogleMapReact จากนั้นก็เพิ่ม attribute lat กับ lng ลงใน tag เพื่อที่จะทำให้ tag นั้น ไปแสดงผลอยู่ ณ. ตำแหน่งที่ระบุ บน Google Maps ได้ ตามรูปที่ 2.12

```
<GoogleMapReact
  defaultCenter={this.state.center}
  defaultZoom={this.state.zoom}
  layerTypes={['TrafficLayer']}
  onGoogleApiLoaded={({ map, maps }) => { this.setState({ map: map, maps: maps, ma
  yesIWantToUseGoogleMapApiInternals
  style={{ zIndex: -1 }}
>
 <img src={busStop} alt="bus stop" lat={ 13.914245} lng={100.550938  } />
<img src={busStop} alt="bus stop" lat={ 13.914} lng={100.550} />
<img src={busStop} alt="bus stop" lat={ 13.915} lng={100.551} />
</GoogleMapReact>
```

รูปที่ 2.12 ตัวอย่างการใส่ tag เพิ่มเติมใน GoogleMapReact

6) ผลที่ได้คือจะได้รูป bus stop ที่อยู่บนตำแหน่ง latitude longitude ตามที่กำหนดบน Google Maps จึงสามารถใช้ GoogleMapReact แสดงตำแหน่งป้ายรถเมย์ และตำแหน่งรถประจำทางสาธารณะได้ ดังรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การแสดงผลหลังการเพิ่ม tag อื่น ๆ ใน GoogleMapReact

2.7 Google Maps API Direction

Google Maps API Direction เป็น API maps service กลางที่ถูกพัฒนาโดย google โดยเปิดเป็น opensource ให้ผู้พัฒนาสามารถนำ service เกี่ยวกับแผนที่ต่าง ๆ บน google maps ไปพัฒนาต่อยอดบน application และเว็บไซต์ของตนเองได้

โดย Google Maps API Direction เป็น service ที่ใช้ในการหาเส้นทางแบบละเอียดระหว่างจุดเริ่มต้น (Origin) และปลายทาง (Destination) ที่เป็นจุด latitude และ longitude เรียงกันไปตามเส้นทาง โดยเรามาสามารถเพิ่ม mode การเดินทางหรือฟังก์ชันพิเศษต่าง ๆ ได้ ดังรูปที่ 2.14

ตัวอย่าง Google Maps API Direction

https://maps.googleapis.com/maps/api/directions/json?origin=Toronto&destination=Montreal&key=YOUR_GOOGLE_MAP_API_KEY

ผลลัพธ์ที่ได้คือ array ของจุด GPS ที่เป็นตำแหน่ง GPS ของถนนที่อยู่ระหว่างจุดเริ่มต้น และปลายทาง ที่เราส่งค่าไป ดังรูปที่ 2.14

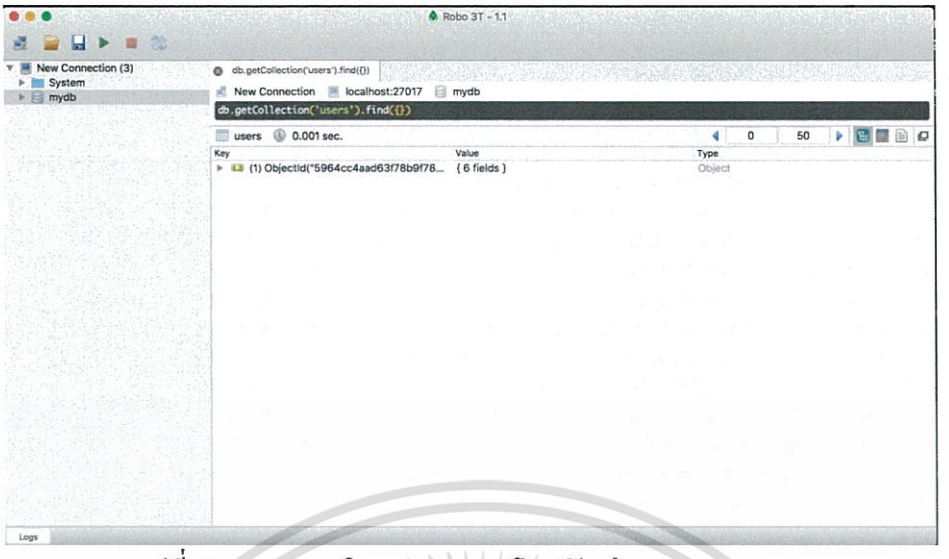
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.14 ข้อมูลที่ได้จากการใช้ Google Maps API Direction

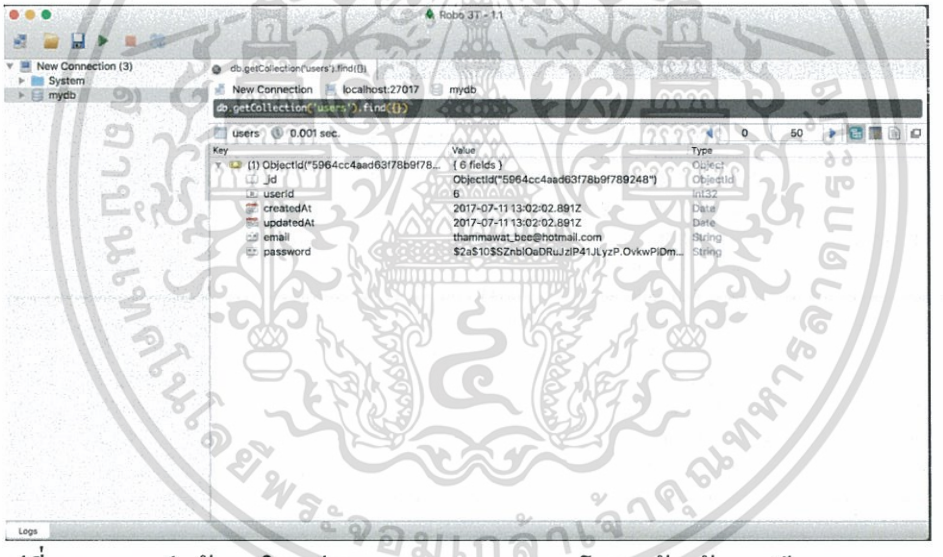
2.8 MongoDB

MongoDB เป็น open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือไม่มี relation ของตารางแบบ SQL ทั่ว ๆไป แต่จะเก็บข้อมูลแบบ JSON (JavaScript Object Natation) แทนการบันทึกข้อมูลทุก ๆ record ใน MongoDB โดย MongoDB จะมี GUI (Graphical User Interface) ให้ใช้งานเพื่อให้สะดวกต่อการบริหารจัดการข้อมูล คือ Robo 3T โดย record ใน MongoDB มีรูปแบบตามรูปที่ 2.15 และ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 Record ใน MongoDB โดยเปิดผ่านทาง Robo 3T



รูปที่ 2.16 การเก็บข้อมูลในรูปแบบของ NoSQL และโครงสร้างข้อมูลเป็นแบบ JSON

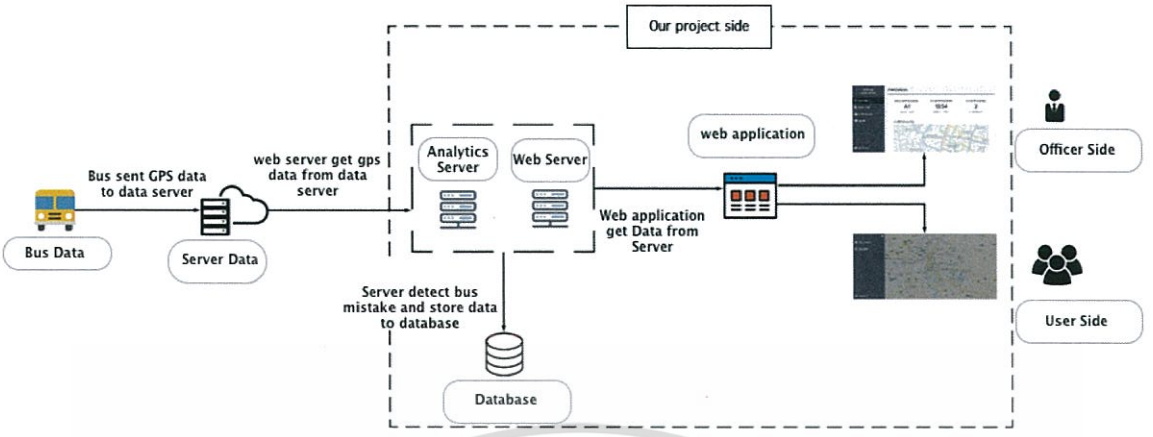
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและพัฒนา

เนื่องจากโครงการนี้เป็นโครงการที่ช่วยเหลือเจ้าหน้าที่ในเรื่องการตรวจสอบการฝ่าฝืนระเบียบของรถประจำทางสาธารณะ ทำให้การออกแบบโครงการนี้ถูกออกแบบมาให้มีส่วนที่บันทึกรายงานความผิดทั้งหมดไว้เพื่อที่เจ้าหน้าที่จะสามารถอ่านและดาวน์โหลดรายงานความผิด แล้วนำไปใช้อ้างอิงภายนอกได้ โดยรายงานความผิดเหล่านี้จะมาจากการที่อัลกอริทึมตรวจสอบการฝ่าฝืนระเบียบได้แล้วนำข้อมูลของรถประจำทางสาธารณะนั้น ๆ มาบันทึกไว้ในฐานข้อมูล โดยหน้าเว็บไซต์ฝั่งเจ้าหน้าที่จะเรียกใช้ข้อมูลจากฐานข้อมูลนี้ไปแสดงแก่เจ้าหน้าที่ แต่เนื่องจากรายงานความผิดเป็นข้อมูลที่มีจำนวนมาก ทำให้ในบางครั้ง เจ้าหน้าที่ไม่สามารถตรวจสอบรายงานความผิดได้อย่างทั่วถึง โครงการนี้จึงทำการออกแบบส่วนของการแสดงภาพรวมเพิ่มขึ้นมา เพื่อทำเป็นข้อสรุปให้เจ้าหน้าที่สามารถเข้าใจได้ไม่ยากและช่วยลดเวลาในการตรวจสอบรายงานความผิดลง แต่ถ้าหากว่ามีเจ้าหน้าที่ที่คอยทำหน้าที่ตรวจสอบข้อมูลรายงานความผิดอยู่เพียงผู้เดียว เจ้าหน้าที่คนนั้นก็จะได้รับภาระหน้าที่มากจนเกินไปและทำให้ความผิดพลาดเช่น การตรวจสอบไม่ทั่วถึงหรือการตรวจสอบล่าช้า เกิดขึ้นตามมาภายหลังได้ จึงเป็นเหตุให้โครงการนี้ออกแบบส่วนของระบบสมาชิกเพิ่มขึ้นมา เพื่อที่จะมอบหมายงานให้เจ้าหน้าที่คนอื่นสามารถเข้ามาช่วยกันตรวจสอบรายงานความผิดได้ อีกทั้งยังเป็นการลดภาระหน้าที่ที่มีต่อเจ้าหน้าที่ลงได้ด้วย แต่เนื่องจากระบบสมาชิกที่เพิ่มขึ้นมาเป็นระบบที่เพิ่ม แก้ไข และลบสมาชิกได้โดยเจ้าหน้าที่ที่ได้รับการอนุญาตแล้วเท่านั้นเพื่อเป็นการป้องกันการเพิ่ม แก้ไข หรือลบข้อมูลโดยพลการ ซึ่งถ้าหากมีการเพิ่มสมาชิกขึ้นมาใหม่แต่เจ้าหน้าที่บันทึกข้อมูลของสมาชิกใหม่ผิด สมาชิกใหม่คนนั้นก็จะไม่สามารถแก้ไขข้อมูลของตัวเองได้ ด้วยเหตุนี้โครงการนี้จึงออกแบบส่วนของการแก้ไขข้อมูลของตัวเองเพื่อแก้ปัญหาข้างต้น โดยที่กล่าวไปทั้งหมดนี้เป็นระบบส่วนการใช้งานของเจ้าหน้าที่เพียงเท่านั้น ซึ่งโครงการนี้ยังถูกออกแบบให้มีส่วนการใช้งานของผู้ใช้งานทั่วไป ได้แก่ ส่วนการค้นหาเส้นทางและการค้นหาสายรถเมล์ เพื่อที่จะช่วยผู้ใช้งานทั่วไปในการตัดสินใจเดินทางไปยังสถานที่ต่าง ๆ ได้อีกด้วย ซึ่งทั้งหมดนี้สามารถสรุปเป็นภาพรวมได้ดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 องค์ประกอบโดยรวมของระบบ

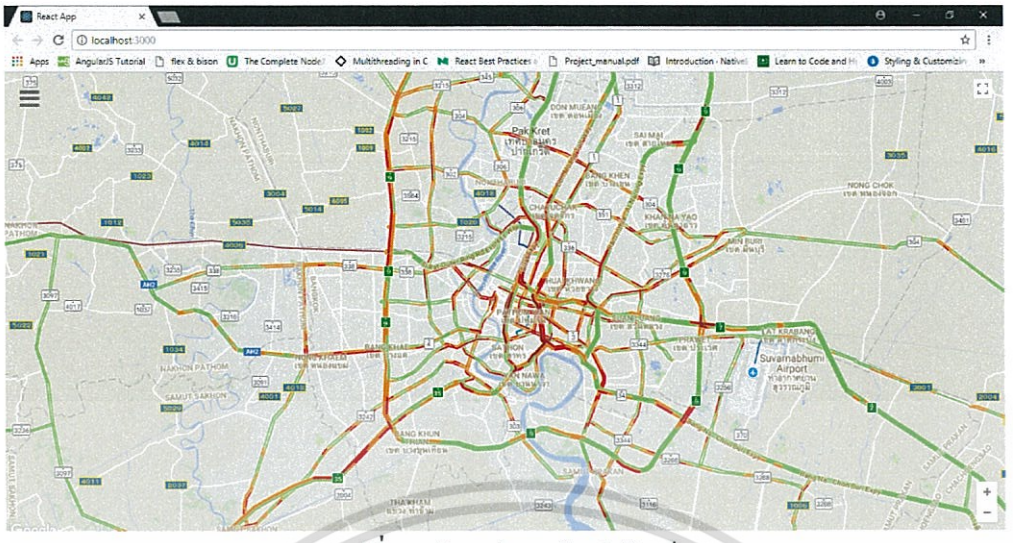
จากรูปที่ 3.1 จะเห็นว่าโครงการนี้จะรับข้อมูลของรถประจำทางสาธารณะมาจาก Data Server ภายนอก หลังจากนั้น Server ภายในโครงการนี้ ก็จะนำข้อมูลไปตรวจสอบโดยอัลกอริทึมที่สร้างขึ้นมา ถ้าหากอัลกอริทึมตรวจสอบพบการฝ่าฝืนระเบียบของรถประจำทางสาธารณะ Server ก็จะบันทึกข้อมูลของรถประจำทางสาธารณะนั้น ๆ ลงในฐานข้อมูล หลังจากนั้น ฝั่ง web application จะเรียกใช้ข้อมูลไปแสดงผลบนหน้าเว็บไซต์ โดยหน้าเว็บไซต์จะแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วนของผู้ใช้งานทั่วไปและส่วนของเจ้าหน้าที่

3.1 การออกแบบส่วนต่าง ๆ ของระบบ

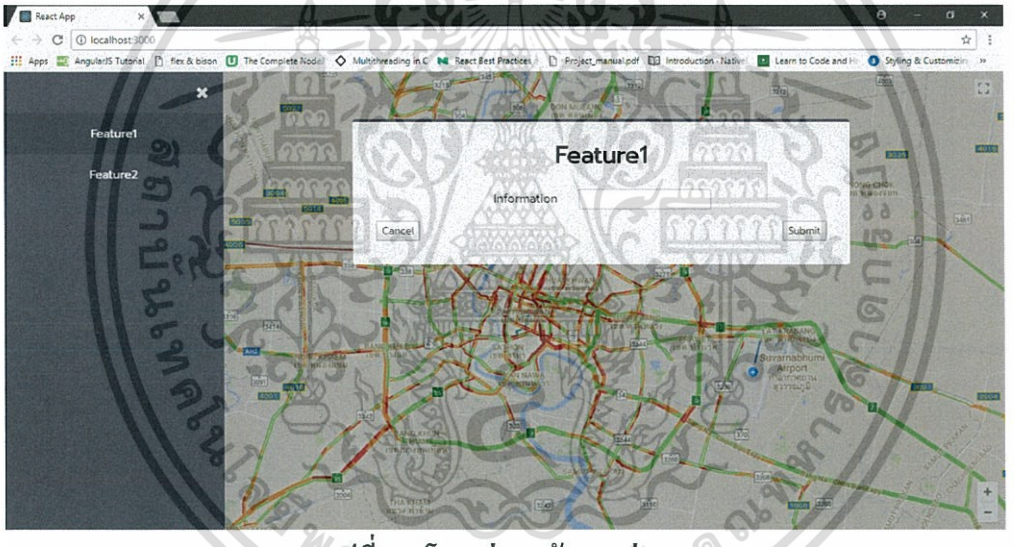
3.1.1 Frontend

ทางด้าน Frontend ได้นำ React มาใช้ในการสร้าง Component ต่าง ๆ บนหน้าเว็บรวมถึงการใส่แผนที่บนหน้าเว็บไซต์ และนำ Redux มาใช้ในการเก็บ state (ตัวแปร) ต่าง ๆ ที่จำเป็นต้องเรียกใช้งานในหลาย Component โดยการออกแบบหน้า Frontend จะมีหน้าแผนที่ไว้สำหรับดูตำแหน่งของรถประจำทางสาธารณะที่ผู้ใช้ต้องการจะโดยสาร และตำแหน่งของป้ายรถเมย์ตามสายของรถประจำทางสาธารณะนั้น ๆ รวมถึงยังมีเมนูต่าง ๆ เพื่อให้ผู้ใช้สามารถเลือกใช้งานได้อย่างมีประสิทธิภาพ ตามรูปที่ 3.2 และ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โครงร่างหน้าเว็บไซต์



รูปที่ 3.3 โครงร่างหน้าเมนูต่าง ๆ

อีกทั้งยังมีหน้าเว็บไซต์สำหรับเจ้าหน้าที่เพื่อที่จะสามารถดูรายงานความผิดของรถประจำทางสาธารณะที่กระทำความผิด โดยเจ้าหน้าที่ต้องทำการ login ก่อนจึงจะสามารถเข้ามาดูในหน้าเว็บไซต์นี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 Backend

Backend ออกแบบโดยใช้ Express ในการทำ API โดยจะมีการแบ่ง API เป็น controllers (การจัดหมวดหมู่ตามลักษณะการใช้งาน) เพื่อแยกความเป็นระเบียบในการยิง API โดยจะแบ่งออกเป็น controllers ต่าง ๆ ซึ่งสามารถดูรายละเอียดต่าง ๆ ของ API ได้ในภาคผนวก

3.1.3 Server

Server ออกแบบโดยใช้ Node.js เป็น Server ที่ทำการดึงข้อมูลจาก Server กลางที่คอยส่งข้อมูล GPS ของรถประจำทางสาธารณะทุก ๆ ประมาณ 5 วินาที แล้วทำการประมวลผลว่า รถประจำทางสาธารณะคันนั้นกระทำความผิดหรือไม่ ผ่านอัลกอริทึมต่าง ๆ หากกระทำความผิด จะถูกบันทึกข้อมูลความผิดลง Database

3.1.4 Database

Database ใช้ MongoDB ในการเก็บข้อมูลต่าง ๆ ดังนี้

1) busOnRoad เป็นข้อมูลของรถประจำทางที่นำมาคำนวณเพื่อใช้ในการตรวจสอบการกระทำความผิดของรถประจำทาง โดยมีรายละเอียดข้อมูลดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดข้อมูลที่เก็บใน busOnRoad Database

Table	Attribute	Description	Data Type	Example
busOnRoad	path	สายการวิ่งของรถ	String	39
	busID	ทะเบียนรถ	String	12-9820
	speed	ความเร็วของรถ		
	lat	ตำแหน่ง latitude	Float	13.76936
	lng	ตำแหน่ง longitude	Flat	100.54069
	cycleOnRoad	รอบที่วิ่งในสายรถ	Int	3
	currentOnRoad	ตำแหน่งลำดับใน สายการวิ่ง	Int	110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Attribute	Description	Data Type	Example
busOnRoad	currentBusStop	ตำแหน่งช่วงของ ป้าย ช่วงการวิ่งรถประจำทาง	Int	4
	passCenter	ค่าเช็กว่าผ่านจุด ศูนย์กลางการวิ่งหรือไม่	Bool	true
	canCompute	ตรวจสอบว่าค่ารถประจำทางคันนี้ สามารถนำไปคำนวณได้หรือไม่	Bool	true
	timeStamp	เวลาที่ส่งค่าข้อมูลของรถประจำทางคันนี้มา	Date	2018-04-14 13:55:28
	gulityState1	เป็นค่า state ที่ใช้ตรวจสอบว่า รถคันนี้ทำความผิด ขับออกนอกเส้นทางมาแล้วหรือไม่	Bool	true
	gulityState2	เป็นค่า stateที่ใช้ตรวจสอบว่า รถคันนี้ทำความผิดขับไม่ครบตามป้ายที่กำหนดใช่หรือไม่	Bool	false
	gulityState3	เป็นค่า state ที่ใช้ตรวจสอบว่า รถคันนี้ขับรถแซงกันในสายเดียวกันาแล้วใช่หรือไม่	Bool	true

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Attribute	Description	Data Type	Example
busOnRoad	busLog	เป็นค่า array ที่ใช้ในการเก็บข้อมูล latitude longitude ก่อนหน้าของรถประจำทาง	Array of object [{lat: Float, lng: Float}]	[{lat:13.91136 lng:100.497}, {lat: 13.91136, lng:100.497}]

2) roadBusStop เป็น table ที่ใช้บันทึกรายละเอียดข้อมูลของสายการวิ่งแต่ละสาย เช่น ชื่อ และตำแหน่ง latitude longitude ของป้ายรถประจำทางทั้งหมดที่มีในสายการวิ่งนั้น ๆ เป็นต้น โดยมีรายละเอียดข้อมูลดังตารางที่ 3.2

ตารางที่ 3.2 รายละเอียดข้อมูลที่เก็บใน roadBusStop Database

Table	Attribute	Description	Data Type	Example
roadBusStop	path	สายการวิ่งของรถ	String	39
	busStop	ป้ายรถประจำทางทั้งหมดในเส้นทางการวิ่งนั้น	Array of Object [{ sequence:int, nameTH: String, nameEG: String, Detail: String, lat: Float, lng: Float, roadIndex: Int, }]	[{ sequence:0, nameTH: หน้าตลาดไท , nameEG: thai market, Detail: ถนน พหลโยธิน , lat: 14.081765, lng: 100.6188, roadIndex: 137, }]

3) Roadmapbusses เป็น table ที่ใช้ในการเก็บตำแหน่ง latitude และ longitude ทั้งหมดของสายการวิ่งของรถประจำทางแต่ละสาย โดยมีรายละเอียดข้อมูลดังตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 รายละเอียดข้อมูลที่เก็บใน Roadmapbusses Database

Table	Attribute	Description	Data Type	Example
Roadmapbusses	path	สายการวิ่งของรถ	String	A1
	roadMap	Array ที่เก็บตำแหน่ง latitude และ longitude ของเส้นทางการวิ่งทั้งหมดของ รถประจำทาง	Array of Object [{ Index: Int, lat: Float, lng: Float }]	[{ Index: 1, lat: 13.98568, lng: 100.60381 }]

4) Busstopsequences เป็น table ที่ใช้ในการเก็บค่าช่วงการวิ่งของป้ายแต่ละป้ายของสายรถประจำทางโดยสร้างจาก การนำตำแหน่ง latitude longitude ป้ายประจำทางในสายนั้น ๆ map กับข้อมูลตำแหน่งใน roadmap (จากตาราง Roadmapbusses ใน path เหมือนกัน) แล้วนำมาสร้างเป็น array sequence โดยมีรายละเอียดข้อมูลดังตารางที่ 3.4

ตารางที่ 3.4 รายละเอียดข้อมูลที่เก็บใน Busstopsequences Database

Table	Attribute	Description	Data Type	Example
Busstopsequences	path	สายการวิ่งของรถ	String	39
	sequence	Array ที่เก็บลำดับป้ายที่อยู่ในเส้นทางการวิ่งของรถประจำทาง	Array of Int [Int,Int,int ...]	[149,209,227, 240,...]

5) BusGulity เป็น table ที่ใช้ในการเก็บการกระทำความผิดของรถประจำทาง โดยมีรายละเอียดข้อมูลดังตารางที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 รายละเอียดข้อมูลที่เก็บใน BusGulity Database

Table	Attribute	Description	Data Type	Example
BusGulity	path	สายการวิ่งของรถที่มีการกระทำความผิด	String	39
	busID	ทะเบียนของรถประจำทางที่กระทำความผิด	String	12-5581
	type	ประเภทความผิดของรถประจำทาง	String	ขับออกนอกเส้นทาง
	timeStamp	ช่วงเวลาที่กระทำความผิด	Date	2018-04-01 16:35:04
	lat	ตำแหน่ง latitude	Float	13.90766
	lng	ตำแหน่ง longitude	Float	100.50363
	state	ค่าที่ไว้ใช้สำหรับการยืนยันการทำความผิดโดยจะมีแค่ 1 กับ 0	Int	1
	busLog	เป็นค่า array ที่ใช้ในการเก็บข้อมูล latitude longitude ก่อนหน้าของรถประจำทาง	Array of object [{lat: Float, lng: Float}]	[{lat:13.91136 lng:100.497 }, {lat: 13.91136, lng:100.497963333333}]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Attribute	Description	Data Type	Example

BusGulity	overDriveOtherBus	เป็นค่า object ที่ใช้เก็บข้อมูลของรถ ที่โดนรถประจำทางที่เราตรวจสอบอยู่ขับแซง	Object
			{
			busID: String,
			lat: Float,
			lng: Float,
			}
				{
				busID:12-5587,
				lat:13.90766,
				lng: 100.50363
				}

6) Road เป็น table ที่ใช้ในการเก็บข้อมูลสายการวิ่งของรถประจำทาง โดยมีรายละเอียดข้อมูลดังตารางที่ 3.6

ตารางที่ 3.6 รายละเอียดข้อมูลที่เก็บใน Path Database

Table	Attribute	Description	Data Type	Example
Path	name	ชื่อสายการวิ่งของรถประจำทาง	String	39
	fullname	ชื่อเต็มของสายการวิ่งของรถประจำทาง	String	39 ตลาดไท-อนุสาวรีย์ชัยสมรภูมิ
	currentCycleOnRoad	จำนวนรอบการวิ่งของสาย ณ เวลาปัจจุบัน	Int	2
	busStopSequence	ข้อมูลช่วงตำแหน่งของลำดับป้ายรถประจำทาง โดย Attribute นี้จะมี relationกับตาราง busStopSequence	ObjectID	ObjectId("5a83cb89cea5942c5d275787")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Attribute	Description	Data Type	Example
Path	roadMapBus	ข้อมูลตำแหน่ง GPS ตามเส้นทางการวิ่งของรถในเส้นทางนั้น ๆ โดย Attribute นี้จะมี relationกับ ตาราง roadMapBus	ObjectID	ObjectId("5a830c32da98b4ff67464f54")
	centerPath	ลำดับใน roadMap และตำแหน่ง latitude longitude ของ จุดศูนย์กลางการวิ่งของรถ	object { Index: Int, lat:Float, lng:Float }	{ Index: 467, lat:13.7657, lng: 100.53823 }
	contract	ข้อมูลว่า สายการวิ่งนี้เชื่อมต่อกับสายไหนบ้างโดย Attribute นี้จะมี relationกับตาราง buscontact	ObjectID	ObjectId("5a9fe2c989737c169fd86d66")

7) User เป็น table ที่ใช้ในการเก็บข้อมูลของ user โดยมีรายละเอียดข้อมูลดังตารางที่ 3.7

ตารางที่ 3.7 รายละเอียดข้อมูลที่เก็บใน User Database

Table	Attribute	Description	Data Type	Example
User	firstname	ชื่อจริงของ user	String	ธรรมวัฒน์
	lastname	นามสกุลของ user	String	บงการณ์
	username	Username ของ user	String	Thammawat_57KMITL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Attribute	Description	Data Type	Example
User	password	Password ของ user	String	$2a$10$1GQc7eW 59EMNIsjx90Sza. QiQJUsmbMGvWnWKg 5z7TWq87U7aNDay
	status	สถานะของ user ในระบบ	String	admin

8) Contract เป็น table ที่ใช้ในการเก็บว่าสายรถเมย์ขับตัดกับสายไหนบ้าง โดยมีรายละเอียดข้อมูลดังตารางที่ 3.8

ตารางที่ 3.8 รายละเอียดข้อมูลที่เก็บใน Contract Database

Table	Attribute	Description	Data Type	Example
Contract	path	สายการวิ่งของรถ	String	63
	contract	Array ที่เก็บรวบรวม จุดตัด (ป้ายรถประจำทางซ้ำกัน)ของสายรถประจำทาง กับสายรถอื่น ๆ	Array of object [{ connect: { contractWith: String, contractPath: [{ nameTH: String, Sequence: Int, contractAt: Int, }] } }]	[{ connect: { contractWith: A1, contractPath: [{ nameTH: สำนักงานใหญ่ธนาคารทหารไทย, Sequence: 30, contractAt: 2, }] } }]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบการใช้งานของ User

3.2.1 การใช้งานของ User ทั่วไป

User ทั่วไปที่เข้ามาใช้ระบบ จะสามารถเลือกเมนูสำหรับการใช้งานได้ตามต้องการ โดยเมนูจะมีตัวเลือกต่าง ๆ ดังนี้

1) การหาเส้นทางที่ผ่านระหว่างต้นทางกับปลายทางที่ User ระบุ

2) การดูเส้นทางการเดินรถของรถประจำทางในแต่ละสาย

3.2.2 การใช้งานของ เจ้าหน้าที่

1) เจ้าหน้าที่จะต้องเข้ามา log in ที่หน้า Web Application ก่อนจึงจะสามารถเข้ามาหน้าเว็บที่รวบรวม รายงานความผิดของรถประจำทางสาธารณะได้ โดยจะแสดงอยู่ในรูปของ file .csv ในแต่ละวัน

2) เจ้าหน้าที่สามารถโหลดไฟล์รายงานความผิดรถประจำทางสาธารณะได้

3.3 การออกแบบวิธีการในการตรวจสอบการกระทำความผิดของรถประจำทาง

3.3.1 ปัญหาการวิ่งออกนอกเส้นทาง

สมมุติฐานในการตรวจสอบ คือ ทำการสร้างเส้นทางในแต่ละการวิ่งของรถประจำทางสาธารณะในแต่ละสายออกมา โดยใช้ Google Maps API Direction โดยจะถูกเก็บอยู่ในรูปแบบจำนวนชุด array ของ จุด GPS ที่อยู่บนเส้นทางนั้น แล้วทำการเปรียบเทียบค่า GPS ของรถประจำทางสาธารณะว่าอยู่ใกล้เคียงกับเส้นทางนั้นหรือไม่ หากอยู่ไกลเกินขอบเขตที่กำหนด จะถือว่ารถประจำทางสาธารณะคันนั้นวิ่งออกนอกเส้นทาง

3.3.2 ปัญหาการวิ่งไม่ครบป้ายตามที่กำหนด

สมมุติฐานในการตรวจสอบ คือ ใช้อัลกอริทึมวิเคราะห์จากตำแหน่งของรถประจำทางสาธารณะ ว่าเข้าใกล้ป้ายตามสายที่กำหนดอยู่ในระยะต่ำกว่า 500 เมตรหรือไม่แล้วจะทำการเพิ่มค่าให้กับตัวแปรตัวหนึ่งที่เอาไว้เก็บว่า วันนี้รถประจำทางสาธารณะคันนั้นวิ่งตรงตามป้ายไปแล้วกี่ครั้ง และเมื่อหมดเวลาวิ่งของรถประจำทางสาธารณะคันนั้น จะทำการนำค่าของตัวแปรนั้นไปเทียบกับค่าของจำนวนป้ายรถประจำทางสาธารณะทั้งหมดที่ต้องวิ่งในแต่ละวัน หากค่าของตัวแปรนั้นต่ำกว่าค่าของจำนวนป้ายที่ต้องวิ่งผ่านทั้งหมด จะถือว่ารถประจำทางสาธารณะคันนั้นวิ่งไม่ครบตามป้ายที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ปัญหาการขับแซงกันในสายเดียวกัน

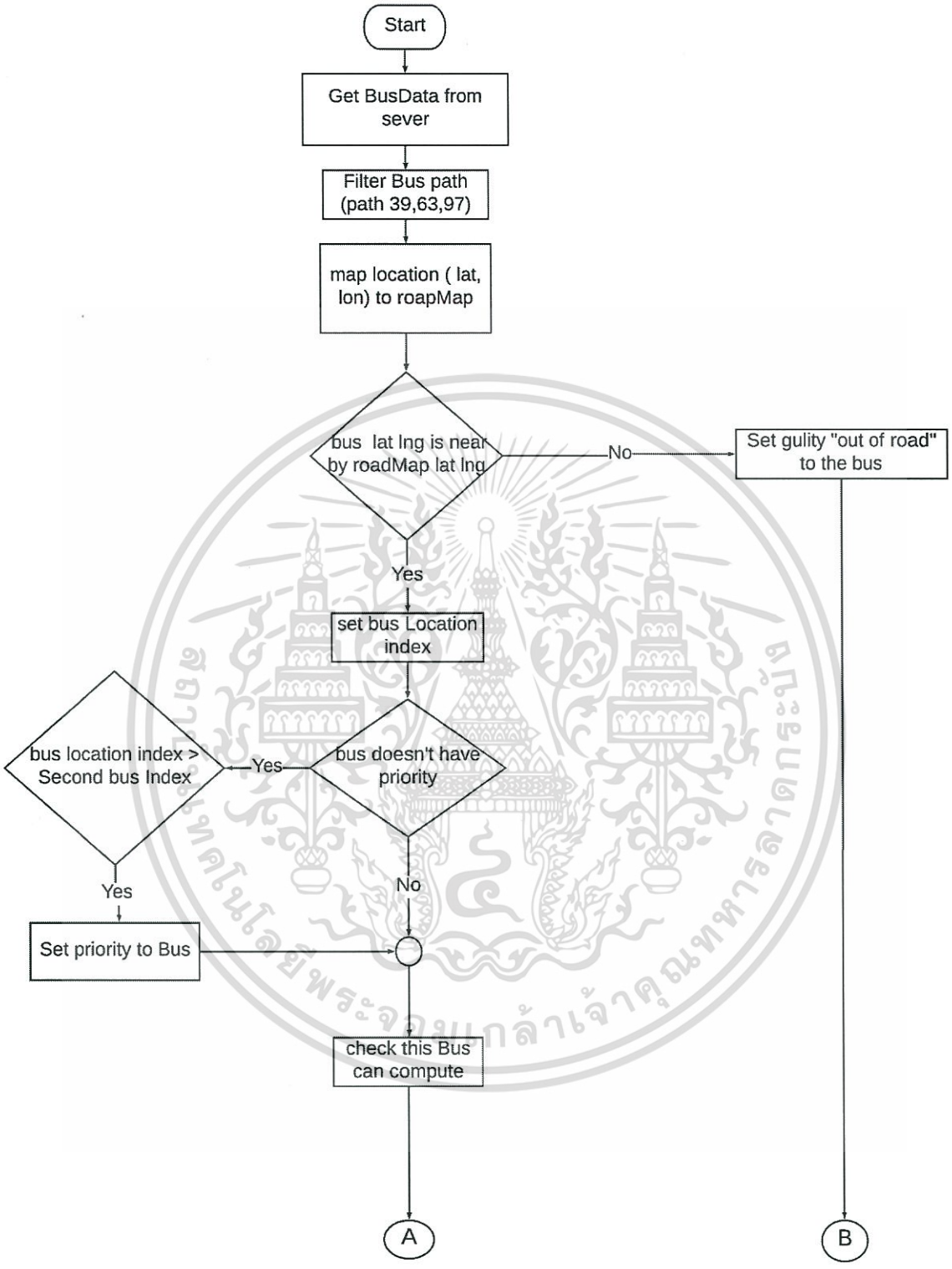
สมมุติฐานในการตรวจสอบ คือ ทำการสร้างเส้นทางในแต่ละการวิ่งของรถประจำทางสาธารณะในแต่ละสายออกมา โดยใช้ Google Maps API Direction โดยจะถูกเก็บอยู่ในรูปแบบจำนวนชุด array ของ จุด GPS ที่อยู่บนเส้นทางนั้น โดยจะมีการกำหนดค่า priority ให้กับรถประจำทางสาธารณะในสายเดียวกัน โดยค่า priority ของรถประจำทางสาธารณะนั้น จะขึ้นอยู่กับเวลาที่รถเริ่มออกตัว ซึ่งรถที่ออกตัวมาก่อน จะมี priority ที่สูงกว่ารถที่ออกตัวทีหลัง แล้วจะทำการเปรียบเทียบตำแหน่งของรถประจำทางสาธารณะ 2 คันจาก array ของเส้นทางที่ต้องวิ่ง หากพบว่า รถสาธารณะที่มี priority ต่ำกว่า อยู่ในตำแหน่ง index ของ array เส้นทางที่มากกว่าตำแหน่ง index ของ array รถสาธารณะที่มี priority สูงกว่า จะถือว่ารถประจำทางสาธารณะคันนั้น วิ่งแซงรถในสายเดียวกัน

3.4 การสร้างอัลกอริทึมตามสมมุติฐานที่ออกแบบไว้

3.4.1 ภาพรวมของอัลกอริทึม

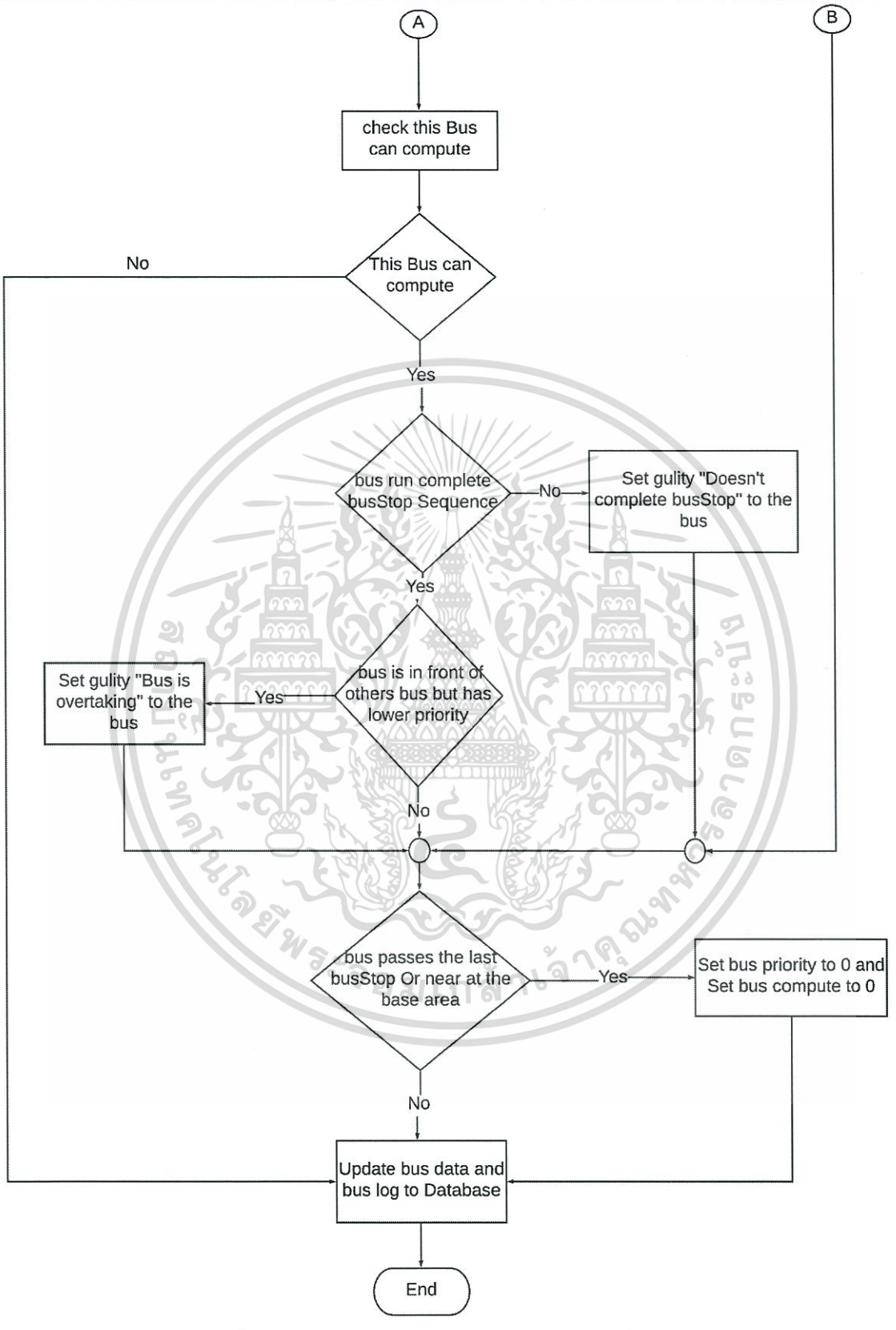
การตรวจสอบการกระทำผิดของอัลกอริทึมจะเริ่มจากการตรวจสอบการขับรถออกนอกเส้นทางก่อนจากนั้นจะทำการตั้งค่า Priority ให้กับรถประจำทางสาธารณะ การตรวจสอบว่าเราสามารถนำข้อมูลของรถประจำทางสาธารณะมาคำนวณได้หรือ ตรวจสอบการขับไม่ครบตามป้ายที่กำหนดตรวจสอบการขับรถแซงกันในสายเดียวกัน ตามลำดับ โดยก่อนจบการคำนวณอัลกอริทึมจะทำการเก็บประวัติการวิ่งของรถประจำทางสาธารณะนั้น ๆ ไว้ทุกครั้ง ดังรูปที่ 3.4 และ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ภาพรวมการทำงานของอัลกอริทึม (ครึ่งแรก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ภาพรวมการทำงานของอัลกอริทึม (ครึ่งหลัง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 ขั้นตอนการเตรียมการก่อนการสร้างอัลกอริทึม

1) กำหนดสายของรถประจำทางสาธารณะที่ต้องการจะตรวจสอบ

2) ค้นหาชื่อและตำแหน่ง GPS ของป้ายของรถประจำทางสาธารณะและ GPS จุดจอดรถ (อู่) แต่ละสาย

3) สร้างเส้นทางของรถประจำทางสาธารณะแต่ละสาย โดยเริ่มจากจุดจอดรถไปยังเส้นทางขาไปและขากลับมายังจุดจอดรถโดยดูจาก logs การวิ่งของรถประจำทางสาธารณะสายนั้น ๆ

4) ทำการ map ตำแหน่ง GPS ของแต่ละสายเข้ากับเส้นทางของรถประจำทางสาธารณะ เพื่อสร้างช่วงการวิ่งของรถโดยใช้ Google Maps API Directions แล้วทำการเก็บค่า GPS เส้นทางการวิ่งของรถทั้งหมดเข้าสู่ roadMap database

5) ทำการ map ตำแหน่ง GPS ของป้ายรถประจำทางสาธารณะเข้ากับจุด GPS ที่ใกล้ที่สุดของเส้นทางการวิ่งของรถ เพื่อที่จะได้รู้ว่าป้ายแต่ละป้ายอยู่ลำดับใดบนเส้นทางวิ่งของรถ โดยเราสามารถนำค่าลำดับของป้ายมาใช้ในการกำหนดช่วงการวิ่งของรถได้ เมื่อทำการ map สำเร็จ จะเก็บค่าอันดับ GPS ของป้ายรถประจำทางสาธารณะบนเส้นทางการวิ่งเข้าสู่ busStopSequence database

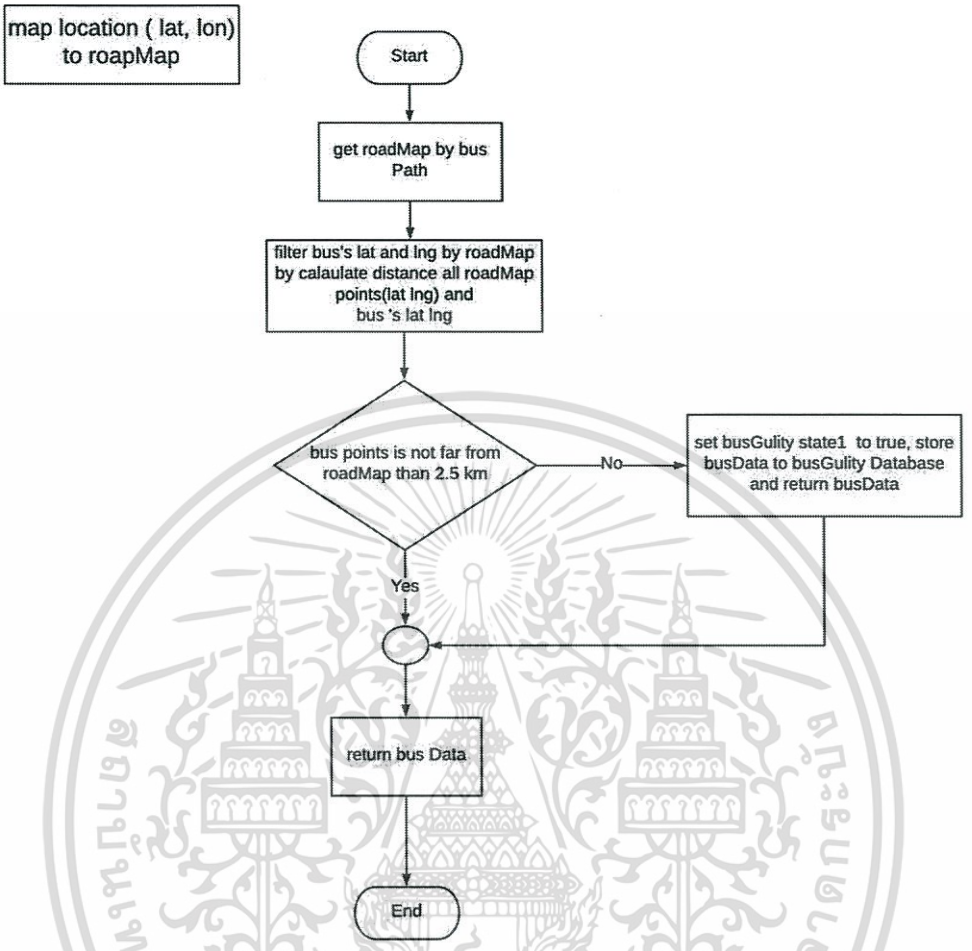
3.4.3 การตั้งค่าความสำคัญ (Priority) ให้รถประจำทางสาธารณะ

การตั้งค่า Priority ของรถประจำทางสาธารณะ จะทำการตั้งค่า Priority ต่อเมื่อรถประจำทางสาธารณะขับผ่านป้ายแรกโดย Priority ของรถที่วิ่งผ่านป้ายแรกในช่วงเวลารอบแรก ๆ จะสูงกว่ารถที่ขับผ่านป้ายแรกช่วงหลัง ๆ แล้วจะทำการรีเซ็ต Priority ก็ต่อเมื่อรถประจำทางสาธารณะขับผ่านป้ายสุดท้ายในการวิ่ง หรือระยะทางระหว่างรถสาธารณะกับจุดจอดไม่เกิน 1.5 กิโลเมตร

3.4.4 การตรวจสอบการขับออกนอกเส้นทาง

เริ่มจากการนำ latitude longitude ของรถประจำทางสาธารณะมาเปรียบเทียบกับจุด latitude longitude ของเส้นทางการวิ่งของรถสายนั้น (roadMap database) แล้วทำการกรองและเปรียบเทียบว่า หากระยะทางที่สั้นที่สุดจากตำแหน่งของรถประจำทางสาธารณะและเส้นทางการวิ่งว่า หากระยะทางที่สั้นที่สุดมากกว่า 2.5 กิโลเมตร จะถือว่ารถประจำทางสาธารณะคันนั้นกระทำความผิดคือ ขับออกนอกเส้นทาง ดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

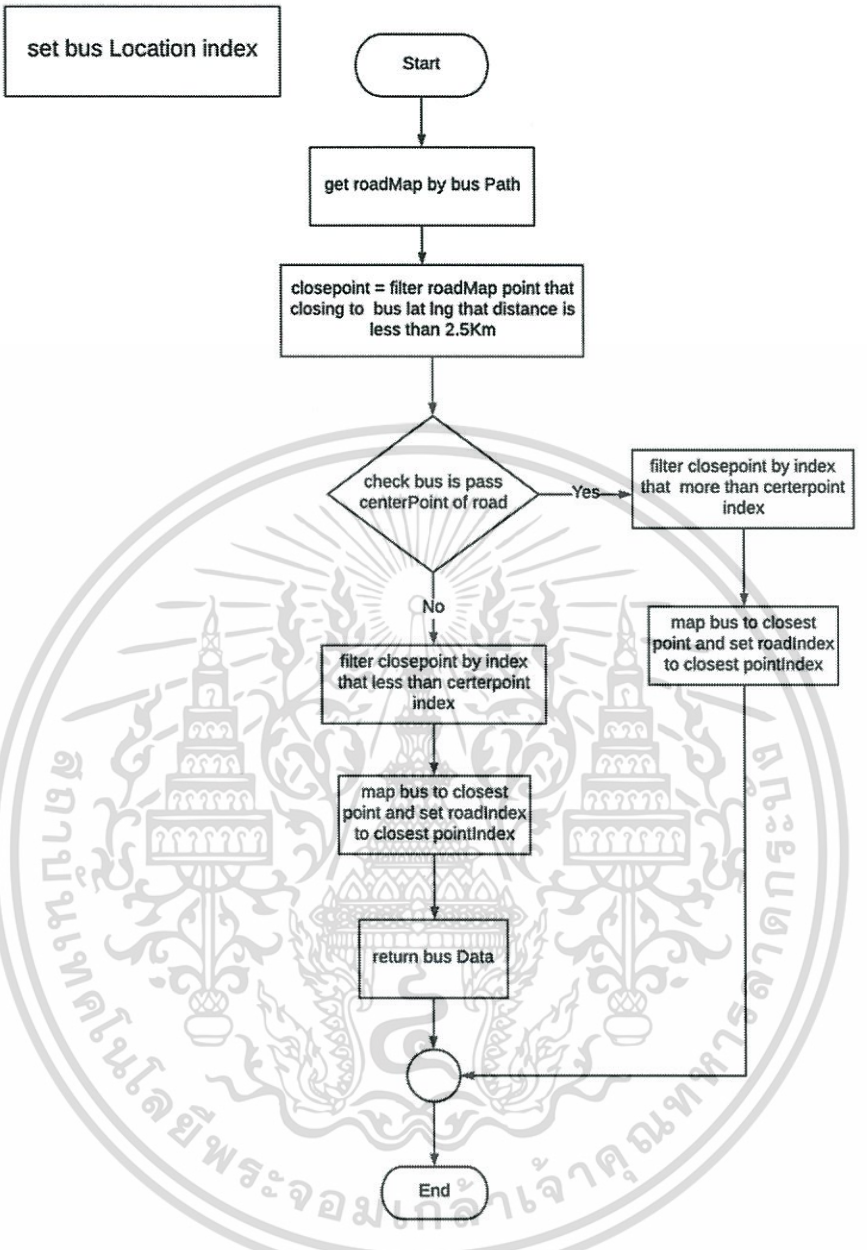


รูปที่ 3.6 การทำงานของอัลกอริทึมตรวจสอบการขับออกนอกเส้นทาง

3.4.5 การเทียบตำแหน่งของรถเข้ากับลำดับในเส้นทางที่สร้างขึ้น

หลังจากที่เราทำการตรวจสอบการขับออกนอกเส้นทาง แล้วหากรถประจำทางสาธารณะที่เราตรวจสอบอยู่ไม่ออกนอกเส้นทาง เราจะทำการกรองจุด GPS ของเส้นทางการวิ่ง ที่ใกล้เคียงกับตำแหน่งของรถประจำทางสาธารณะ แล้วทำการตรวจสอบว่ารถประจำทางคันนี้ ผ่านศูนย์กลางของเส้นทางหรือไม่ หากพบว่ายังไม่ผ่านจุดศูนย์กลางการวิ่งจะทำการกรอง GPS ที่มีลำดับตำแหน่งน้อยกว่า ลำดับตำแหน่ง GPS ของจุดศูนย์กลาง แล้วทำการ map เข้ากับลำดับจุดที่ใกล้เคียงที่สุด แต่หากพบว่า ผ่านจุดศูนย์กลางการวิ่งจะทำการกรอง GPS ที่มีลำดับตำแหน่งมากกว่า ลำดับตำแหน่ง GPS ของจุดศูนย์กลาง แล้วทำการ map เข้ากับลำดับจุดที่ใกล้เคียงที่สุด ดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



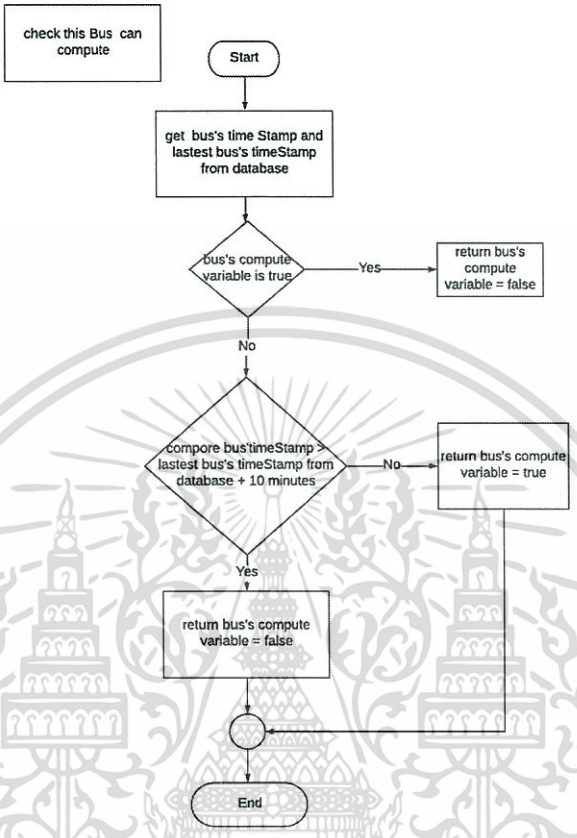
รูปที่ 3.7 การทำงานของการเทียบตำแหน่งรถประจำทางสาธารณะเข้ากับลำดับในเส้นทางที่สร้างขึ้น

3.4.6 การตรวจสอบว่าข้อมูลของรถสามารถนำมาคำนวณได้หรือไม่

เนื่องจากในบางครั้งข้อมูลรถประจำทางสาธารณะที่ได้รับมามีความล้าช้านานกว่าปกติ ทำให้เกิดให้เกิดความคลาดเคลื่อนในการคำนวณ จึงจำเป็นต้อง ทำการเปรียบเทียบเวลาที่ได้รับข้อมูลของรถประจำทางสาธารณะปัจจุบัน กับ เวลาที่เก็บข้อมูลของรถประจำทางสาธารณะไว้ล่าสุด (จาก database) หากพบว่าห่างกันเกิน 10 นาที จะไม่ทำการคำนวณอัลกอริทึมตรวจสอบที่เหลือ ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้แก่การตรวจสอบการขับไม่ครบตามป้ายที่กำหนด และการตรวจสอบการขับแซงกันในสายเดียวกัน ดังรูปที่ 3.8

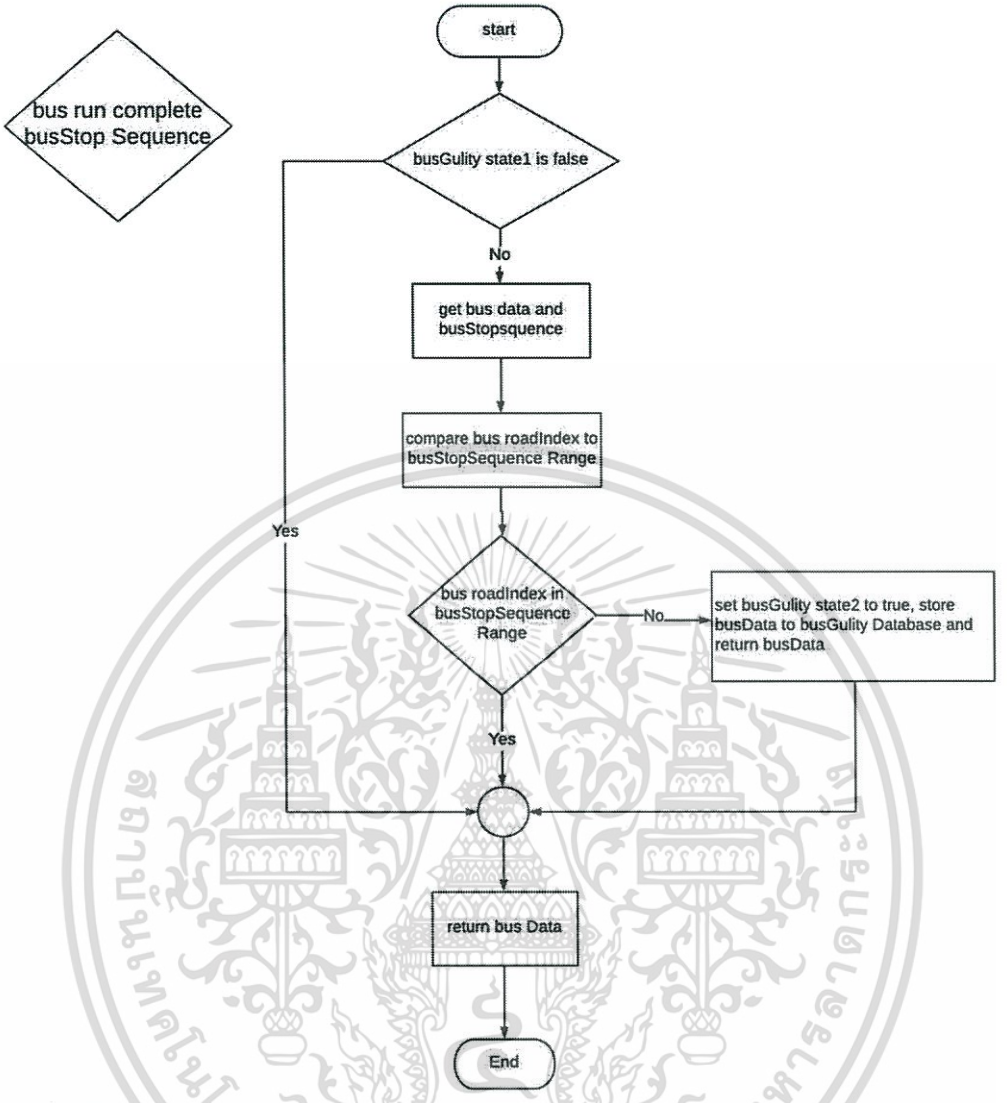


รูปที่ 3.8 การทำงานของการตรวจสอบความใช้ได้จริงของข้อมูล

3.4.7 การตรวจสอบการขับไม่ครบตามป้ายที่กำหนด

เราจะนำลำดับตำแหน่งบนเส้นทางของรถประจำทางสาธารณะประจำทางมาเปรียบเทียบกับช่วงเวลาการวิ่งของรถ (busStopSequence database) โดยเราจะทำการ ตรวจสอบว่า ลำดับของรถประจำทางต้องอยู่ในช่วงลำดับของการวิ่งของรถที่ควรจะอยู่ โดยหากไม่อยู่ในช่วงลำดับของการวิ่งของรถ ที่กำหนดไว้ถือว่าขับไม่ครบตามป้ายที่กำหนด ดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

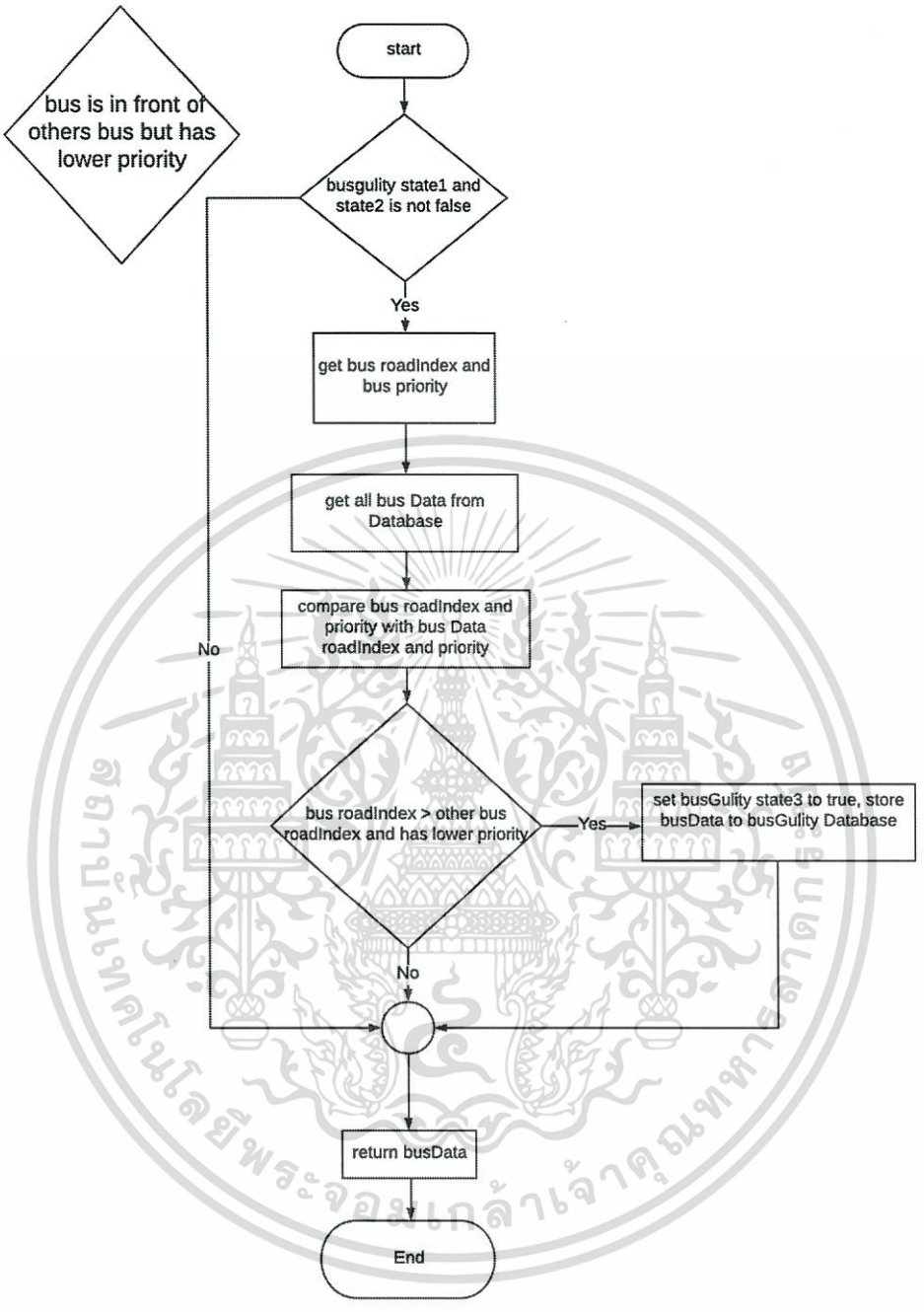


รูปที่ 3.9 การทำงานของอัลกอริทึมตรวจสอบการขับไม่ครบตามป้ายที่กำหนด

3.4.8 การตรวจสอบการขับแซงรถในสายเดียวกัน

เริ่มทำการตรวจสอบโดยการตรวจสอบลำดับของตำแหน่งของรถประจำทางสาธารณะคันนั้น ๆ กับลำดับของตำแหน่งของรถประจำทางสาธารณะคันอื่นในสายเดียวกัน โดยหากพบว่าลำดับของตำแหน่งของรถประจำทางสาธารณะที่ตรวจสอบอยู่ มีลำดับที่สูงกว่าหรือก็คืออยู่นำหน้ารถประจำทางสาธารณะคันที่มี priority สูงกว่าจะถือว่า รถคันนั้นขับแซงรถในสายเดียวกัน ดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การทำงานของอัลกอริทึมตรวจสอบการแซงรถในสายเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

เพื่อให้เห็นผลลัพธ์ที่ชัดเจนของประสิทธิภาพการทำงานของเทคโนโลยี React ที่จะใช้แสดงผลตำแหน่งของรถบนแผนที่แทนวิธีการใช้ Google Maps API รูปแบบเดิม จึงทำการทดลองเพื่อเปรียบเทียบการใช้ทรัพยากรของระบบได้แก่ CPU, Memory, Network ของ client ดังนี้

4.1 การทดลองเปรียบเทียบการ render GPS ของทั้ง Google Maps API และ React

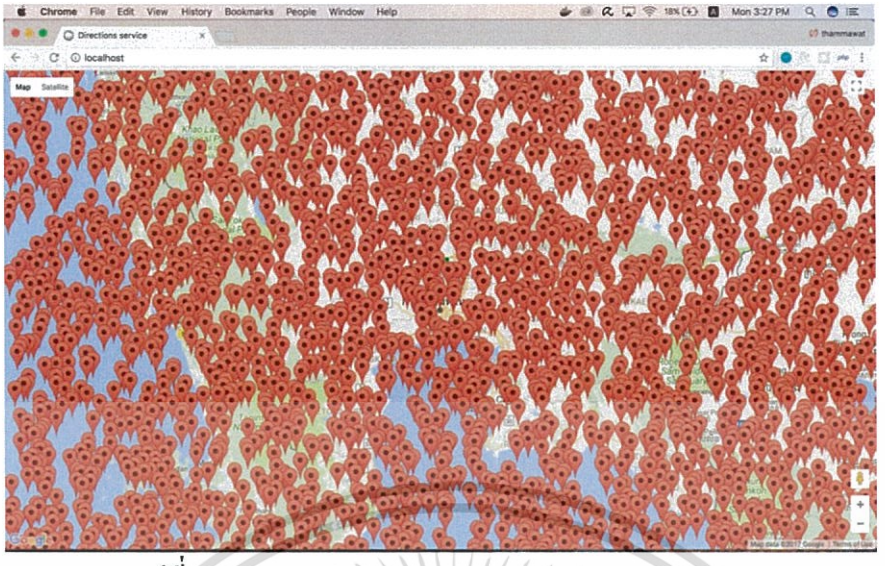
การ render GPS 10,000 จุดบนแผนที่โดยให้ทุกจุดเคลื่อนที่และ render ใหม่(ลบจุดเก่าและสร้างจุดใหม่)ทุก ๆ 5 วินาที เป็นเวลา 5 นาที ซึ่งครั้งแรกใช้ Google Maps API ธรรมดาในการทดลอง ส่วนครั้งที่สองใช้เทคโนโลยี React ในการทดลอง แล้วทำการเปรียบเทียบการใช้ทรัพยากรของเครื่องโดยสิ่งที่ใช้เป็นตัววัดผลในการทดลองนี้คือ CPU (%), Network Sent Bytes, Network Received Bytes และ Memory

สมมุติฐานของการทดลองนี้คือ การใช้ React ในการ render GPS 10,000 จุด จะใช้ทรัพยากรของเครื่องน้อยกว่า การใช้ Google Maps API ธรรมดาในการ render GPS 10,000 จุด

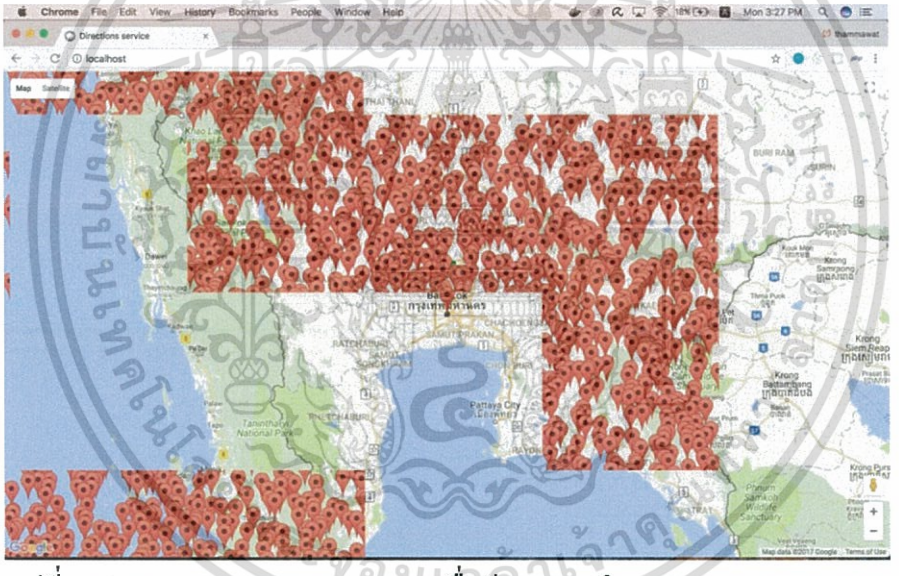
4.1.1 การ render GPS โดยใช้ Google Maps API

ในตอนแรกจะทำการ สุ่มค่า GPS 10,000 จุดขึ้นมาโดยใช้ฟังก์ชันสุ่มเลขของ JavaScript ค่าของ GPS latitude จะมีค่าระหว่าง 5.02 ถึง 19.920 และ ค่า longitude จะมีค่าระหว่าง 97.02 ถึง 104.920 แล้วจะใช้วิธีการส่งตำแหน่ง GPS ไปยัง Google Maps เพื่อในการ render marker ลงบน map โดยตรง แล้วกำหนดให้ทุก ๆ 5 วินาที มีการบวกค่า GPS ทุกจุด ทั้งค่า latitude และ longitude ครั้งละ 0.01 โดยได้ผลการ render ออกมาดังรูปที่ 4.1 และ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 การ render GPS 10,000 จุดของ Google API

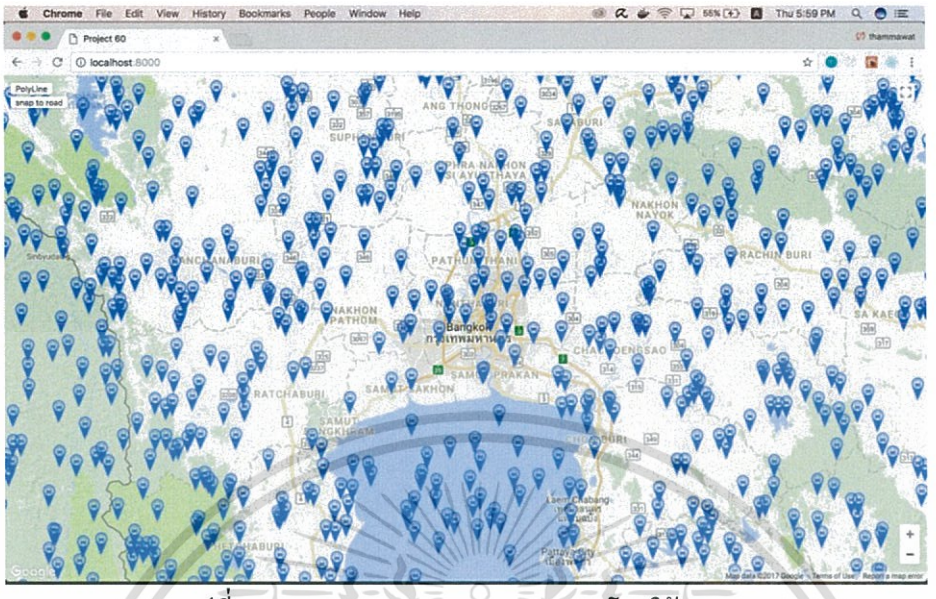


รูปที่ 4.2 การ render GPS 10,000 จุด เมื่อมีการบวกค่า GPS ของ Google API

4.1.2 การ render GPS โดยใช้ React

ในตอนแรกจะทำการสุ่มค่า GPS 10,000 จุดขึ้นมาโดยในการสุ่มค่าของ GPS latitude จะมีค่าระหว่าง 5.02 ถึง 19.920 และ ค่า longitude จะมีค่าระหว่าง 97.02 ถึง 104.920 แล้วทำการ render จุด GPS ลงบน map โดยตรงผ่าน library google-maps-react แล้วกำหนดให้ทุก ๆ 5 วินาที มีการบวกค่า GPS ทุกจุด ทั้งค่า latitude และ longitude ครั้งละ 0.01 โดยได้ผลการ render ออกมาดังรูปที่ 4.3 และ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 การ render GPS 10,000 จุด โดยใช้ React

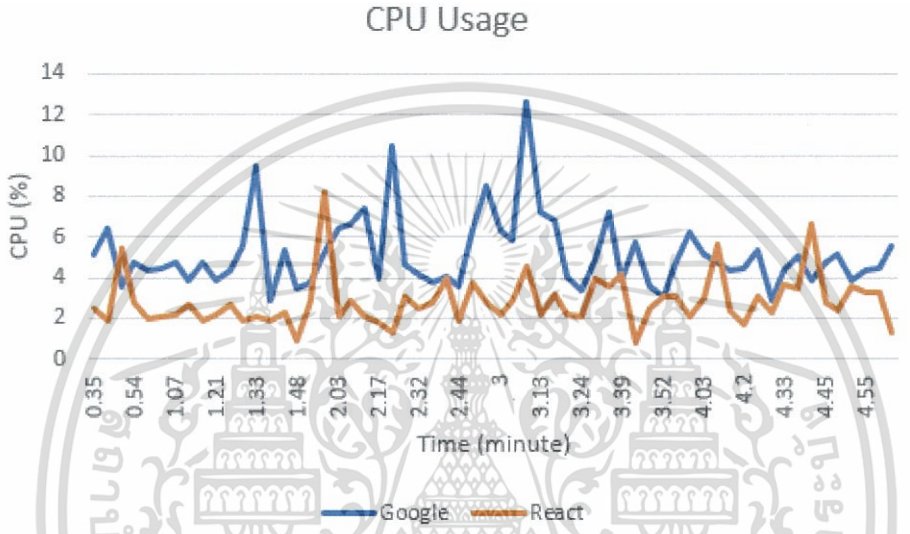
รูปที่ 4.4 การ render GPS 10,000 จุดของ React เมื่อมีการบวกค่า GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การเปรียบเทียบการใช้ทรัพยากรของเครื่องระหว่าง Google API และ React

4.2.1 CPU

เมื่อนำค่า CPU ระหว่าง Google API และ React มาเปรียบเทียบกัน พบว่า React ใช้ CPU น้อยกว่า Google API ดังที่ปรากฏในรูปที่ 4.5 โดย Google API มีค่า CPU สูงสุดและต่ำสุดอยู่ที่ 12.5 และ 1.2 ส่วนค่า CPU สูงสุดและต่ำสุดของ React อยู่ที่ 8.2 และ 0.8

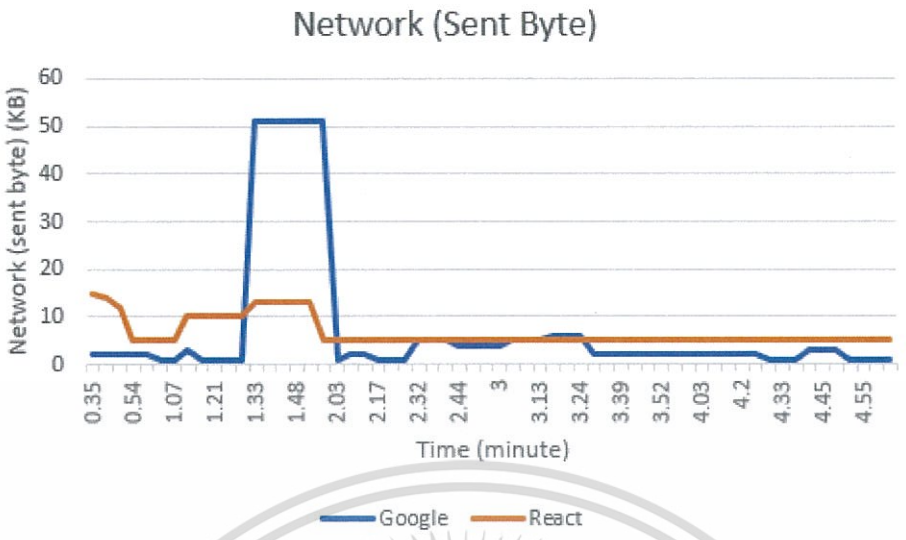


รูปที่ 4.5 กราฟเปรียบเทียบ CPU Usage

4.2.2 Network (Sent Byte)

เมื่อนำค่า Network (Sent Byte) ระหว่าง Google API และ React มาเปรียบเทียบกัน พบว่าไม่มีความแตกต่างกันมากนักระหว่าง Google API และReact ถึงแม้ว่าจะมีบางช่วงที่มีค่า Network (Sent Byte) ของ Google API สูงขึ้นไปมากก็ตาม ดังที่ปรากฏในรูปที่ 4.6 โดย Google API มีค่า Network (Sent Byte) สูงสุดและต่ำสุดอยู่ที่ 60 KB และ 0.758 KB ส่วนค่า Network (Sent Byte) สูงสุดและต่ำสุดของ React อยู่ที่ 14 KB และ 5 KB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

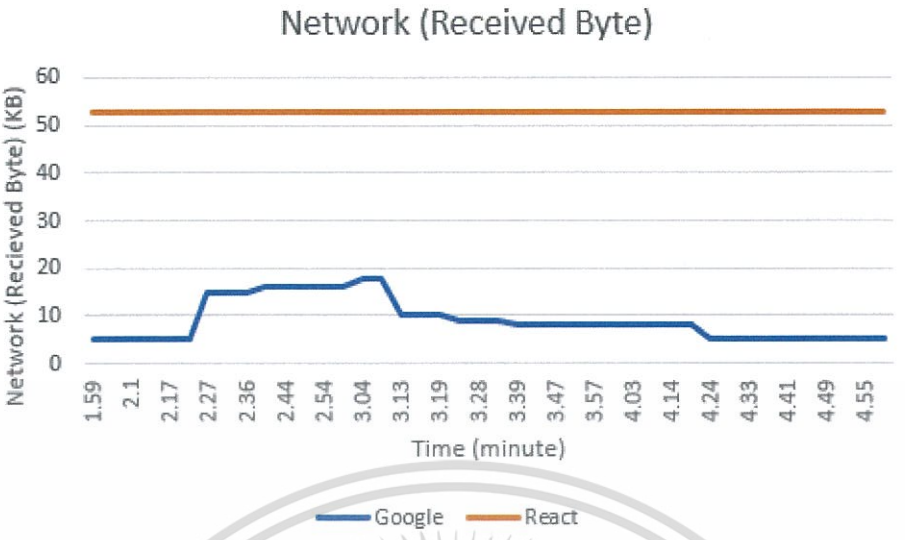


รูปที่ 4.6 กราฟเปรียบเทียบ Network (Sent Byte)

4.2.3 Network (Received Byte)

เมื่อนำค่า Network (Received Byte) ระหว่าง Google API และ React มาเปรียบเทียบกัน พบว่าไม่มีความแตกต่างกันมากนักระหว่าง Google API และReact ถึงแม้ว่าจะมีบางช่วงที่มีค่า Network (Received Byte) ของ React สูงขึ้นไปมากก็ตาม ดังที่ปรากฏในรูปที่ 4.7 โดย Google API มีค่า Network (Received Byte) สูงสุดและต่ำสุดอยู่ที่ 2100 KB และ 5 KB ส่วนค่า Network (Received Byte) สูงสุดและต่ำสุดของ React อยู่ที่ 4200 KB และ 16 KB

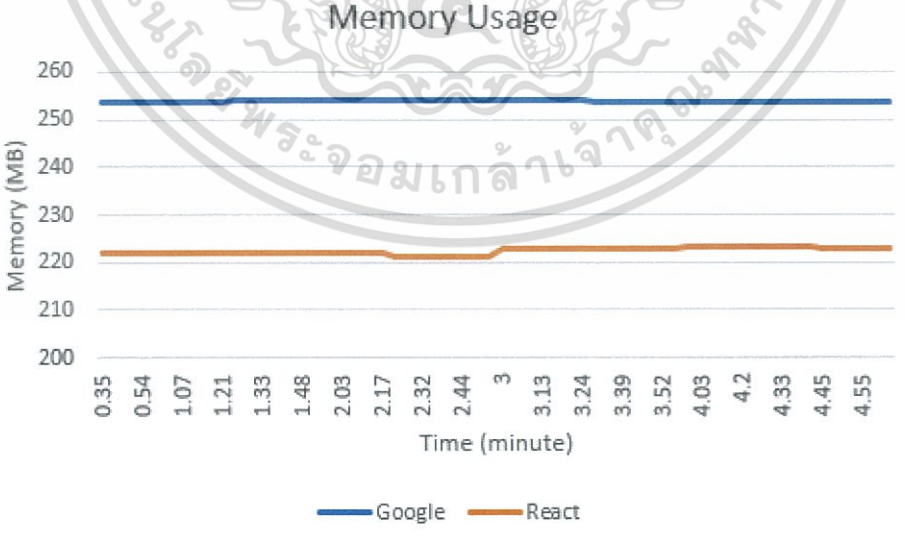
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 กราฟเปรียบเทียบ Network (Received Byte)

4.2.4 Memory

เมื่อนำค่า Memory ระหว่าง Google API และ React มาเปรียบเทียบกัน พบว่า React ใช้ Memory น้อยกว่า Google API ดังที่ปรากฏในรูปที่ 4.8 โดย Google API มีค่า Memory สูงสุดและต่ำสุดอยู่ที่ 254.1 MB และ 253.9 MB ส่วนค่า Memory สูงสุดและต่ำสุดของ React อยู่ที่ 223.1 MB และ 222.0 MB



รูปที่ 4.8 กราฟเปรียบเทียบ Memory Usage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 Rendering

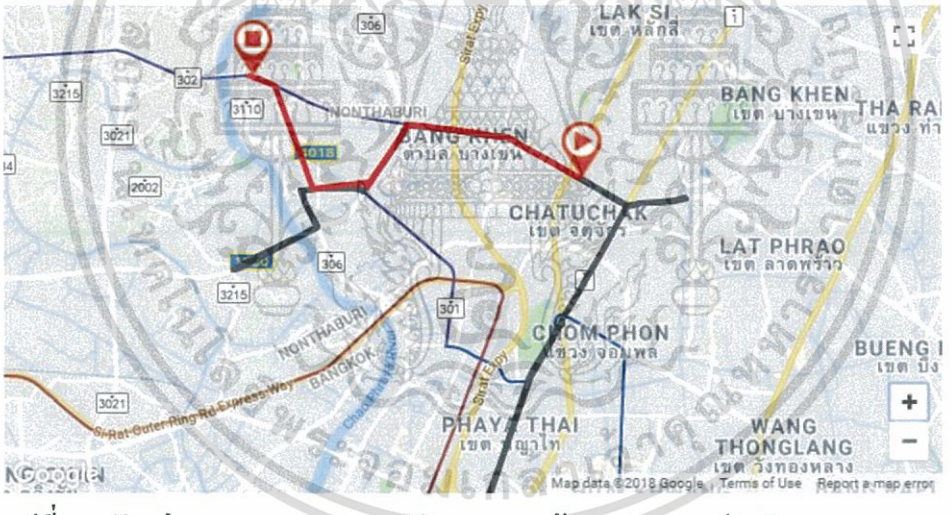
จากการสังเกตและการทดลอง พบว่าขณะ render จุด การ render ของ React มีความไหลลื่น (Smooth) และต่อเนื่อง (Continuous) มากกว่าการ render ของ Google API

4.3 การทดลองนำอัลกอริทึมทั้งสามอัลกอริทึมมาใช้ในการตรวจสอบข้อมูลจริง

เมื่อทำการสร้างอัลกอริทึมเรียบร้อยแล้ว จึงทำการนำอัลกอริทึมมาทดลองใช้ตรวจสอบการกระทำผิดกับข้อมูลรถประจำทางสาธารณะแบบ Realtime ได้ผลการทดลองออกมาดังนี้

4.3.1 การตรวจสอบการขับรถออกนอกเส้นทาง

การตรวจสอบการขับรถออกนอกเส้นทางสามารถตรวจสอบได้จริง ถ้าหากรถประจำทางสาธารณะขับออกจากเส้นทางที่รถคันนั้นควรจะอยู่เกิน 2.5 กิโลเมตร ดังรูปที่ 4.9 โดยเส้นทางสีแดงเป็นเส้นทางการวิ่งของรถประจำทางสาธารณะ ส่วนเส้นทางสีดำเป็นเส้นทางที่รถประจำทางสาธารณะควรจะวิ่งอยู่ และเครื่องหมายรูปสามเหลี่ยมหมายถึงจุดเริ่มต้นการบันทึกการวิ่งของรถ ส่วนเครื่องหมายรูปสี่เหลี่ยมหมายถึงจุดสิ้นสุดการบันทึกการวิ่งของรถ



รูปที่ 4.9 ตัวอย่างการตรวจสอบการขับออกนอกเส้นทางของรถประจำทางสาธารณะ

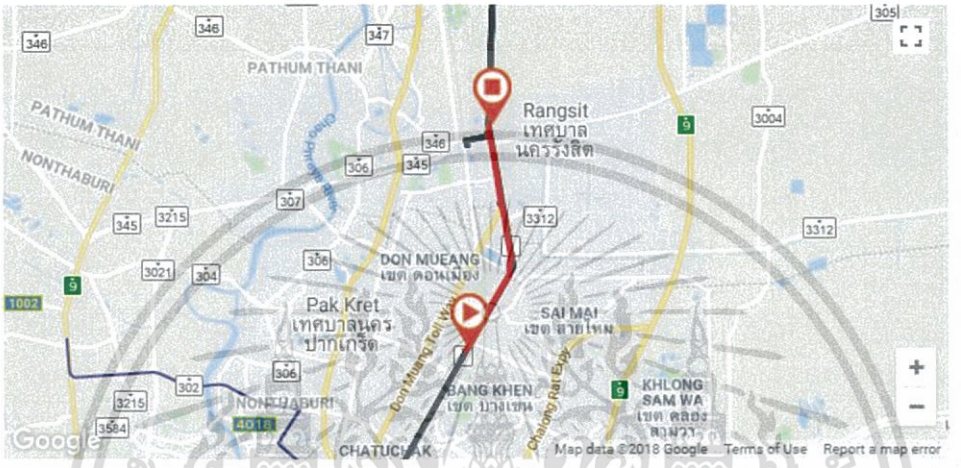
แต่ในกรณีที่ขับออกนอกเส้นทางแต่ไม่เกิน 2.5 กิโลเมตร อัลกอริทึมนี้ก็จะไม่สามารถตรวจสอบได้ว่ารถคันนั้นขับออกนอกเส้นทาง

4.3.2 การตรวจสอบการขับไม่ครบตามป้ายที่กำหนด

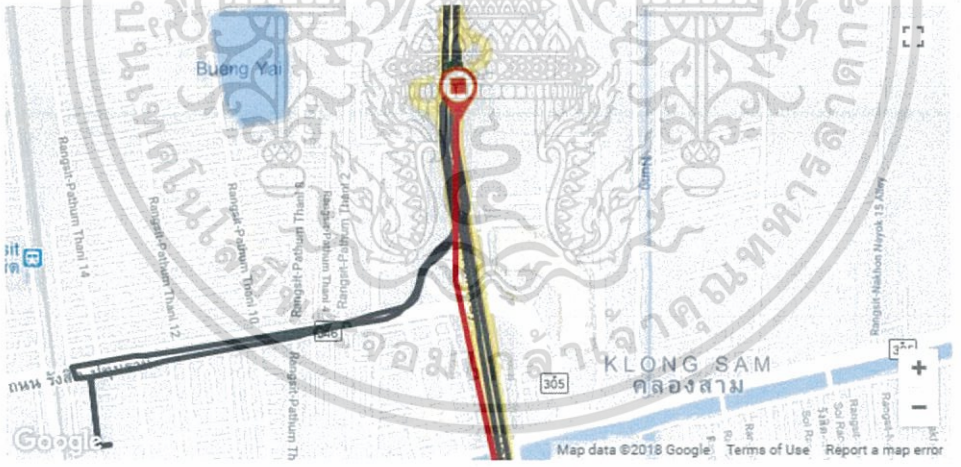
การตรวจสอบการขับไม่ครบตามป้ายที่กำหนดสามารถตรวจสอบได้จริง ถ้าหากรถประจำทางสาธารณะขับไม่ครบตามป้ายที่กำหนดหรือก็คือมีการกลับรถก่อนจุดที่ควรจะกลับรถ แล้ววิ่งต่อไปในฐานะรถที่วิ่งขากลับ ดังรูปที่ 4.10 และ 4.11 โดยเส้นทางสีแดงเป็นเส้นทางการวิ่งของรถประจำทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาธารณะ ส่วนเส้นทางสีดำเป็นเส้นทางที่รถประจำทางสาธารณะควรจะวิ่งอยู่ และเครื่องหมายรูปสามเหลี่ยมหมายถึงจุดเริ่มต้นการบันทึกการวิ่งของรถ ส่วนเครื่องหมายรูปสี่เหลี่ยมหมายถึงจุดสิ้นสุดการบันทึกการวิ่งของรถ ซึ่งในรูปที่ 4.11 จะเห็นได้ว่ารถคันนี้วิ่งอยู่ในเส้นขาไปอยู่ (เส้นทางด้านซ้าย) และต่อมา ตำแหน่งของรถคันนั้นก็เปลี่ยนไปอยู่ในเส้นขากลับ (เส้นทางด้านขวา) ทั้ง ๆ ที่รถคันนี้ควรจะวิ่งต่อขึ้นไปด้านบนอีกตามเส้นทางสีดำในรูป 4.10 จึงสามารถสรุปได้ว่ารถคันนี้วิ่งไม่ครบตามป้ายที่กำหนด



รูปที่ 4.10 การตรวจสอบการขับไม่ครบตามป้ายที่กำหนด



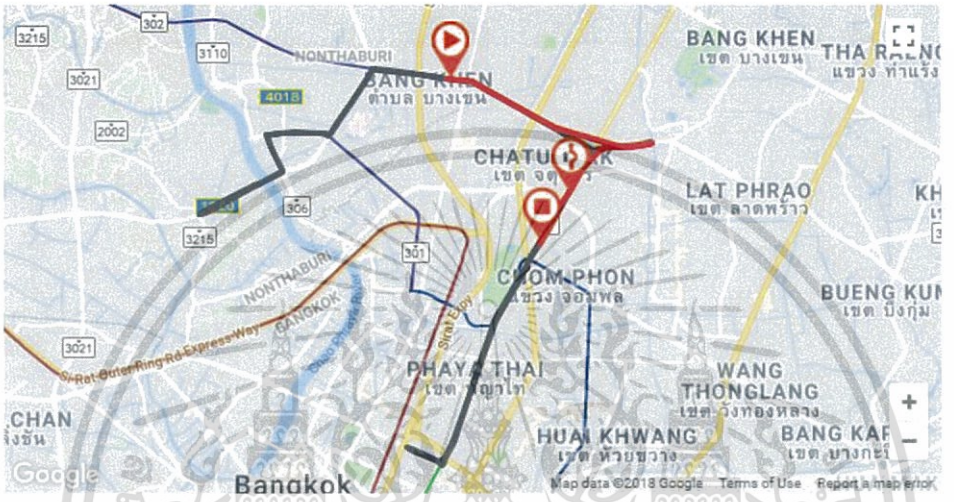
รูปที่ 4.11 การกลับรถก่อนจุดที่ควรจะกลับรถ

แต่ในกรณีที่มีการเปลี่ยนเส้นทางการขับเนื่องจากเหตุสุดวิสัยต่าง ๆ เช่น มีการก่อสร้างหรือซ่อมแซ่มถนน ทำให้จำเป็นต้องใช้ทางเบี่ยง เป็นต้น ก็จะทำให้อัลกอริทึมนี้ไม่สามารถตรวจสอบได้ว่ารถคันนั้นวิ่งครบตามป้ายที่กำหนดหรือไม่ เนื่องจากอัลกอริทึมนี้จำเป็นต้องอ้างอิงกับเส้นทางการวิ่งที่สร้างขึ้นมาอย่างละเอียดถึงจะสามารถตรวจสอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การตรวจสอบการขับแซงกันของรถสายเดียวกัน

การตรวจสอบการขับแซงกันของรถสายเดียวกันสามารถตรวจสอบได้จริง ถ้าหากรถประจำทางสาธารณะขับแซงรถสายเดียวกันที่ขับออกมาก่อน ดังรูปที่ 4.12 โดยเส้นทางสีแดงเป็นเส้นทางการวิ่งของรถประจำทางสาธารณะ ส่วนเส้นทางสีดำเป็นเส้นทางที่รถประจำทางสาธารณะควรจะวิ่งอยู่ และเครื่องหมายรูปสามเหลี่ยมหมายถึงจุดเริ่มต้นการบันทึกการวิ่งของรถ เครื่องหมายรูปสี่เหลี่ยมหมายถึงจุดสิ้นสุดการบันทึกการวิ่งของรถ ส่วนเครื่องหมายลูกศรหมายถึงการแซงกันของรถ



รูปที่ 4.12 การตรวจสอบการขับแซงกันของรถในสายเดียวกัน

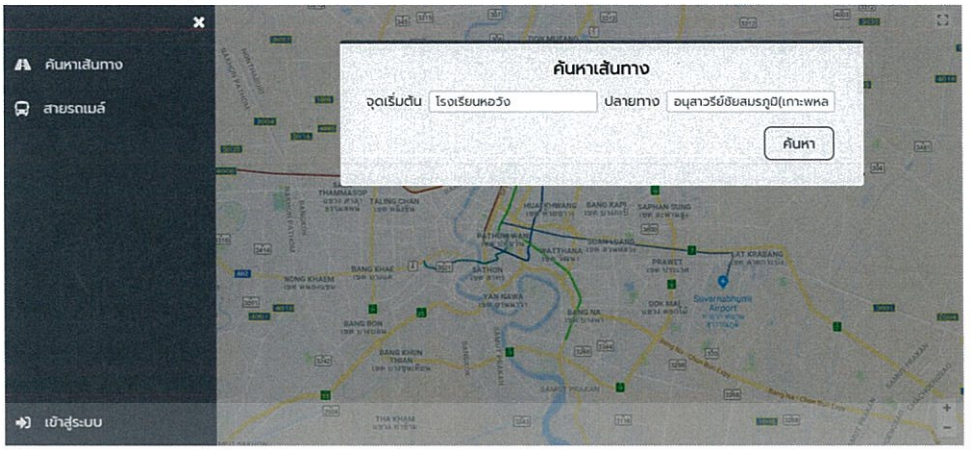
และเช่นเดียวกับอัลกอริทึมของการตรวจสอบการขับไม่ครบตามป้ายที่กำหนด ถ้าหากมีการเปลี่ยนเส้นทางการวิ่ง อัลกอริทึมนี้ก็จะไม่สามารถตรวจสอบได้ว่ามีการขับแซงกันหรือไม่ เนื่องจากจำเป็นต้องอ้างอิงกับเส้นทางการวิ่งที่สร้างขึ้นมาแบบละเอียดเช่นเดียวกัน

4.4 ฟีเจอร์ต่าง ๆ ในระบบ

4.4.1 การค้นหาเส้นทางที่ผ่านต้นทางและปลายทางตามที่ผู้ใช้ระบุ

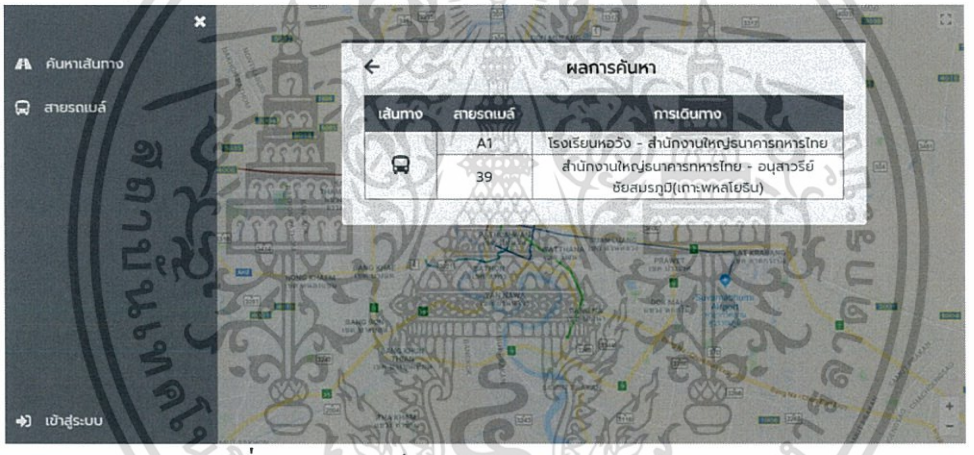
เริ่มแรก ผู้ใช้ต้องกดปุ่มเมนูและเลือกไปที่ตัวเลือก “ค้นหาเส้นทาง” ซึ่งจะมีฟอร์มมาให้ผู้ใช้กรอกจุดเริ่มต้นและปลายทางที่ผู้ใช้ต้องการจะไป เมื่อผู้ใช้กรอกจุดเริ่มต้นและปลายทางเรียบร้อยแล้ว ให้กดไปที่ปุ่มค้นหาเพื่อค้นหาเส้นทางตามจุดเริ่มต้นและปลายทางตามที่ผู้ใช้ต้องการ ดังรูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

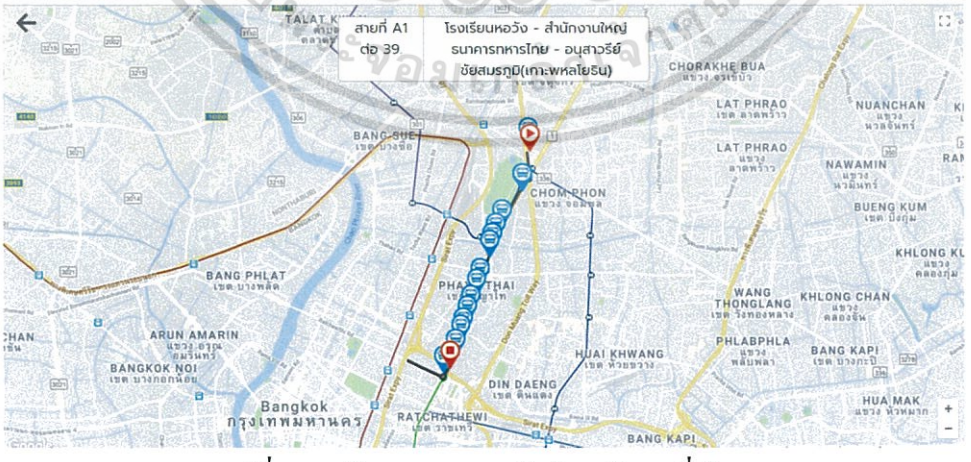


รูปที่ 4.13 แบบฟอร์มการกรอกข้อมูลเพื่อค้นหาเส้นทาง

หลังจากนั้น จะมีผลลัพธ์ของการค้นหาแสดงขึ้นมาให้เลือกเส้นทางที่ผู้ใช้ต้องการจะไป แล้วกดสัญลักษณ์รูปรถบัสของเส้นทางนั้น ๆ เพื่อดูเส้นทางนั้น ๆ บนแผนที่ ดังรูปที่ 4.14 และ 4.15



รูปที่ 4.14 ผลลัพธ์การค้นหาของการค้นหาเส้นทาง

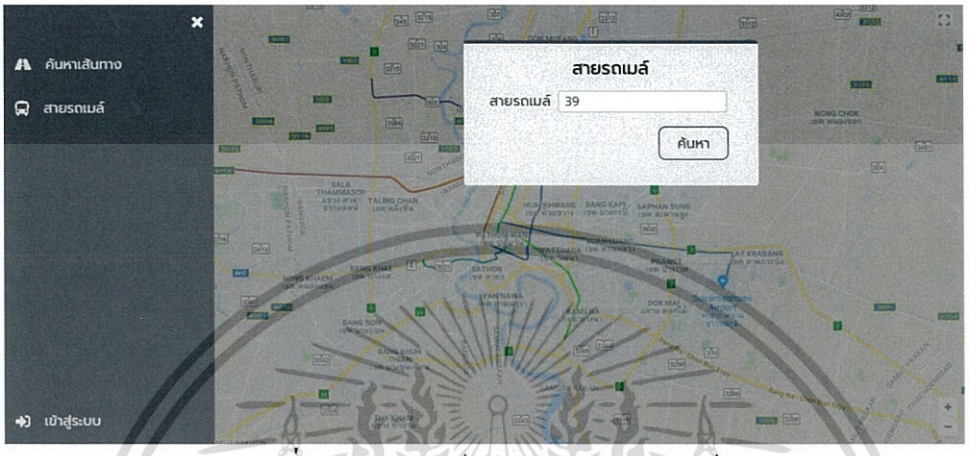


รูปที่ 4.15 เส้นทางของผลลัพธ์การค้นหาที่เลือก

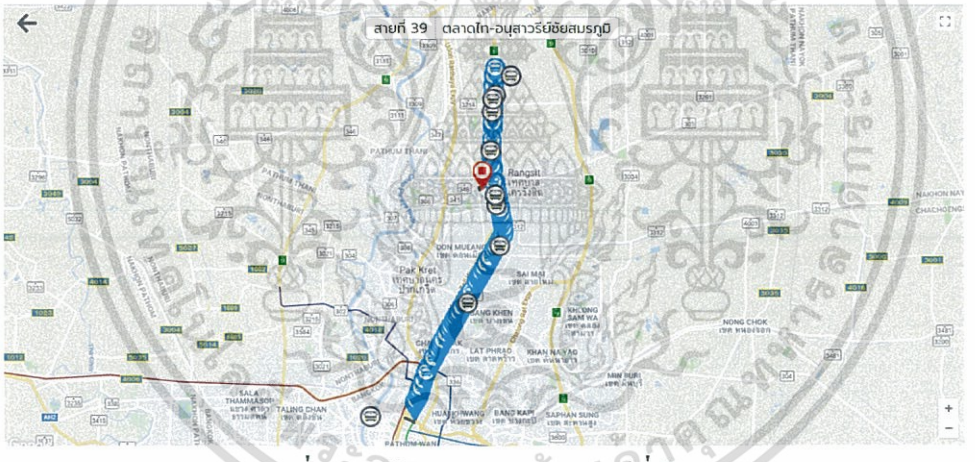
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การค้นหาสายรถประจำทางสาธารณะตามที่ผู้ใช้ระบุ

เริ่มแรก ผู้ใช้ต้องกดปุ่มเมนูและเลือกไปที่ตัวเลือก “สายรถเมล์” ซึ่งจะมีฟอร์มมาให้ผู้ใช้กรอกสายรถเมล์ที่ผู้ใช้ต้องการค้นหา เมื่อผู้ใช้กรอกสายรถเมล์แล้ว ให้กดไปที่ปุ่มค้นหาเพื่อค้นหาเส้นทางตามสายรถเมล์ที่ผู้ใช้ต้องการ ดังรูปที่ 4.16 และ 4.17 ตามลำดับ



รูปที่ 4.16 แบบฟอร์มการค้นหาสายรถเมล์

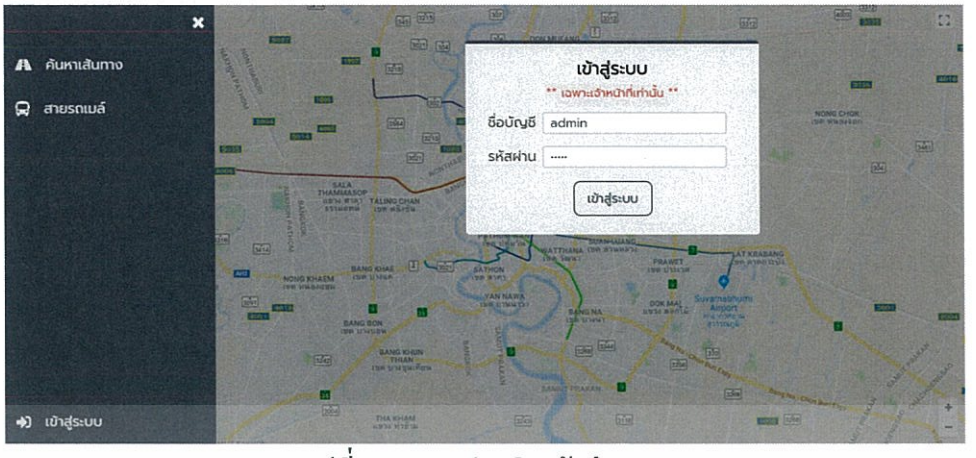


รูปที่ 4.17 เส้นทางของสายรถเมล์ที่เลือก

4.4.3 ระบบล็อกอิน

เริ่มแรก ผู้ใช้ต้องกดปุ่มเมนูและเลือกไปที่ตัวเลือก “เข้าสู่ระบบ” ซึ่งจะมีฟอร์มมาให้ผู้ใช้กรอกชื่อบัญชีและรหัสผ่าน โดยการที่จะมีชื่อบัญชีและรหัสผ่าน จะได้มาจากการที่ผู้ดูแลระบบหรือผู้ช่วยผู้ดูแลระบบเพิ่มให้เท่านั้น เมื่อผู้ใช้กรอกชื่อบัญชีและรหัสผ่านแล้ว ให้กดปุ่มเข้าสู่ระบบเพื่อเข้าไปใช้งานในส่วนของเจ้าหน้าที่ ดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 การล็อกอินเข้าสู่ระบบ

4.4.4 การดูภาพรวมของการกระทำผิดของรถประจำทางสาธารณะ

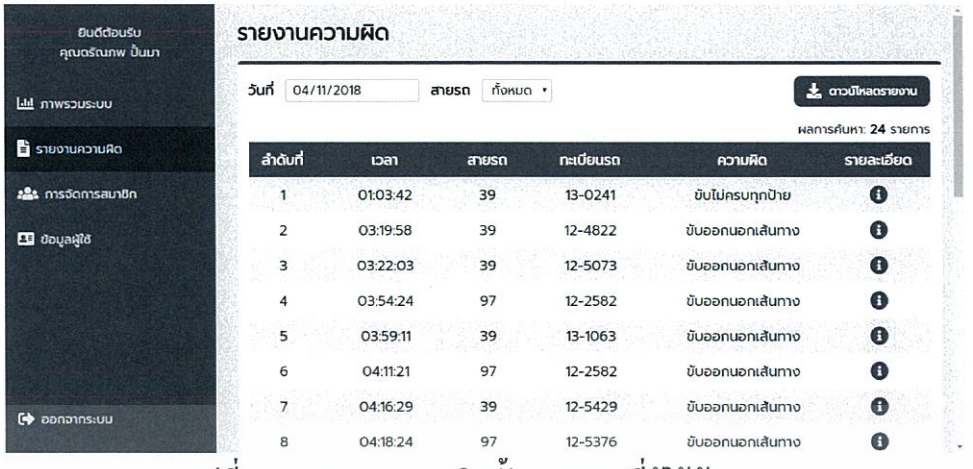
ผู้ใช้สามารถตรวจดูข้อมูลทางสถิติต่าง ๆ ของรถประจำทางสาธารณะ ได้แก่ สายรถเมล์ที่ทำผิดบ่อยที่สุด, ความผิดที่ทำบ่อยที่สุด, ทะเบียนรถที่ทำผิดบ่อยที่สุด และประวัติการทำผิด แล้วนำข้อมูลต่าง ๆ เหล่านี้ไปใช้ประโยชน์ในอนาคตได้ ดังรูปที่ 4.19

รูปที่ 4.19 ภาพรวมรายละเอียดทางสถิติของการฝ่าฝืนระเบียบ

4.4.5 การดูประวัติการทำผิดของรถประจำทางสาธารณะ

ผู้ใช้สามารถดูรายงานความผิดของรถประจำทางสาธารณะตามวันและสายรถที่ผู้ใช้เลือกได้ อีกทั้งยังสามารถดาวน์โหลดรายงาน เพื่อที่จะนำไปใช้อ้างอิงได้อีกด้วย ดังรูปที่ 4.20 และ 4.21 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



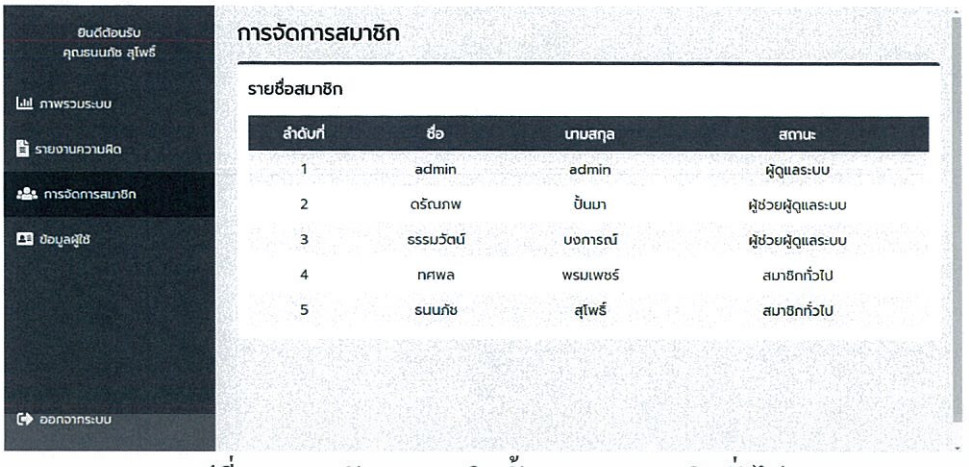
รูปที่ 4.20 รายงานความผิดทั้งหมดตามที่ผู้ใช้ต้องการ

รูปที่ 4.21 ตัวอย่างรายละเอียดการฝ่าฝืนระเบียบ

4.4.6 การจัดการสมาชิกภายในระบบ (สำหรับเจ้าหน้าที่ที่ได้รับอนุญาตเท่านั้น)

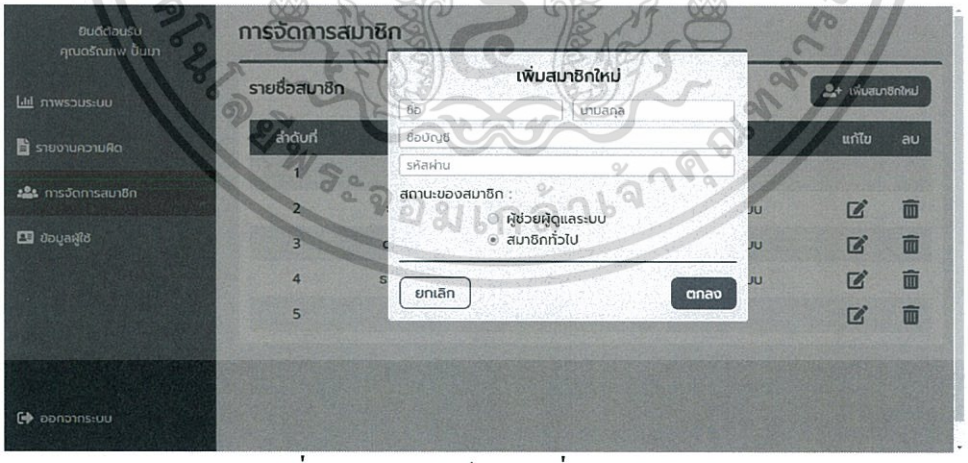
ผู้ใช้สามารถมองเห็นสมาชิกทั้งหมดภายในระบบได้ และสามารถรู้ได้ว่าสมาชิกคนใดมีสถานะเป็นผู้ดูแลระบบ ผู้ช่วยผู้ดูแลระบบ หรือสมาชิก ถ้าหากผู้ใช้ไม่ได้รับอนุญาตจากระบบ หรือก็คือมีสถานะเป็นสมาชิก ก็จะทำให้ผู้ใช้ไม่สามารถ เพิ่ม แก้ไข หรือลบข้อมูลสมาชิกที่อยู่ในระบบได้ ดังรูปที่ 4.22 แต่ถ้าหากผู้ใช้ได้รับอนุญาตจากระบบ หรือก็คือมีสถานะเป็นผู้ดูแลระบบ หรือผู้ช่วยผู้ดูแลระบบ ก็จะสามารถเพิ่ม แก้ไข หรือลบสมาชิกที่อยู่ในระบบได้ ดังรูปที่ 4.23 , 4.24 และ 4.25 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



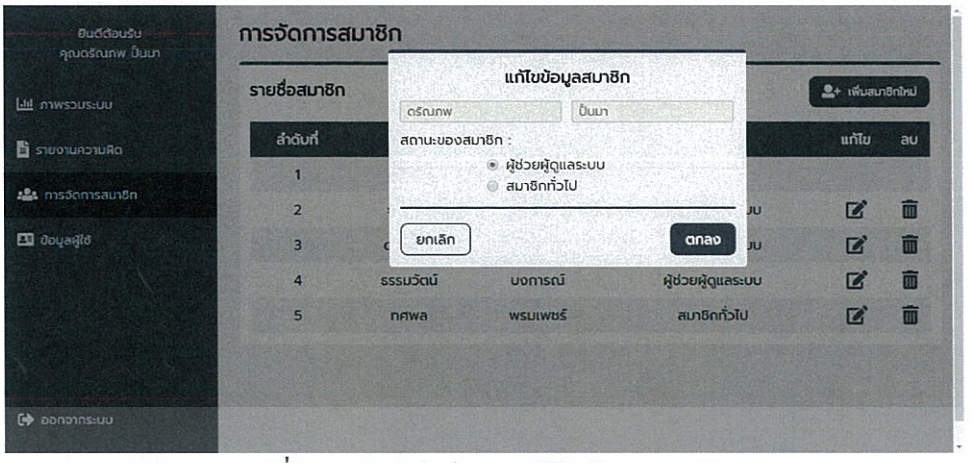
รูปที่ 4.22 การจัดการสมาชิกทั้งหมดของสมาชิกทั่วไป

รูปที่ 4.23 การจัดการสมาชิกของผู้ดูแลหรือผู้ช่วยผู้ดูแล



รูปที่ 4.24 แบบฟอร์มการเพิ่มสมาชิกใหม่

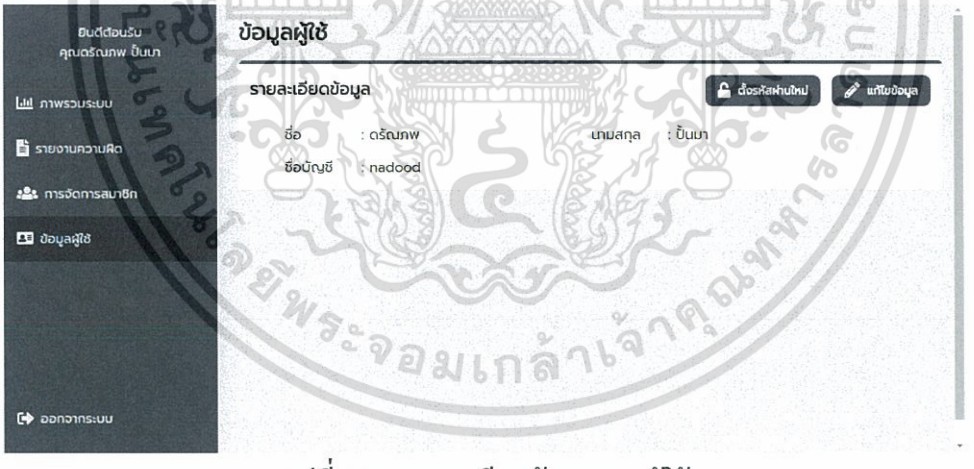
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 แบบฟอร์มการแก้ไขข้อมูลสมาชิก

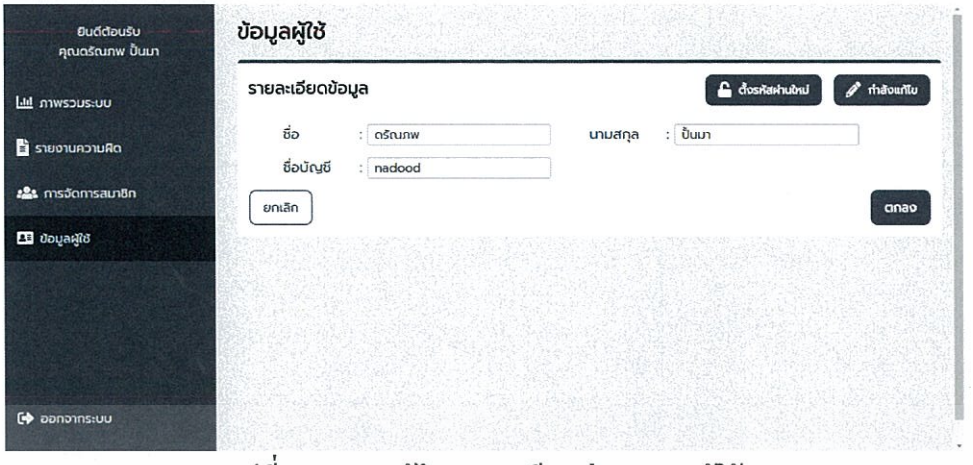
4.4.7 การแก้ไขข้อมูลของผู้ใช้

ผู้ใช้สามารถตรวจสอบและแก้ไขข้อมูลของตัวเองได้ ไม่ว่าจะเป็นชื่อ นามสกุล ชื่อบัญชี หรือรหัสผ่าน เนื่องจากการเพิ่มสมาชิกจะถูกเพิ่มโดยผู้ดูแลระบบหรือผู้ช่วยผู้ดูแลระบบ ซึ่งอาจจะเกิดการใส่ข้อมูลที่ผิดพลาดขึ้น ผู้ใช้ก็สามารถแก้ไขข้อมูลของตัวเองให้ถูกต้องได้ ดังรูปที่ 4.26 และ 4.27 แต่ถ้าหากผู้ใช้ต้องการที่จะเปลี่ยนรหัสผ่าน ผู้ใช้ต้องใส่รหัสผ่านเก่า รหัสผ่านใหม่ และยืนยันรหัสผ่านใหม่ก่อนถึงจะสามารถเปลี่ยนรหัสผ่านได้ ดังรูปที่ 4.28

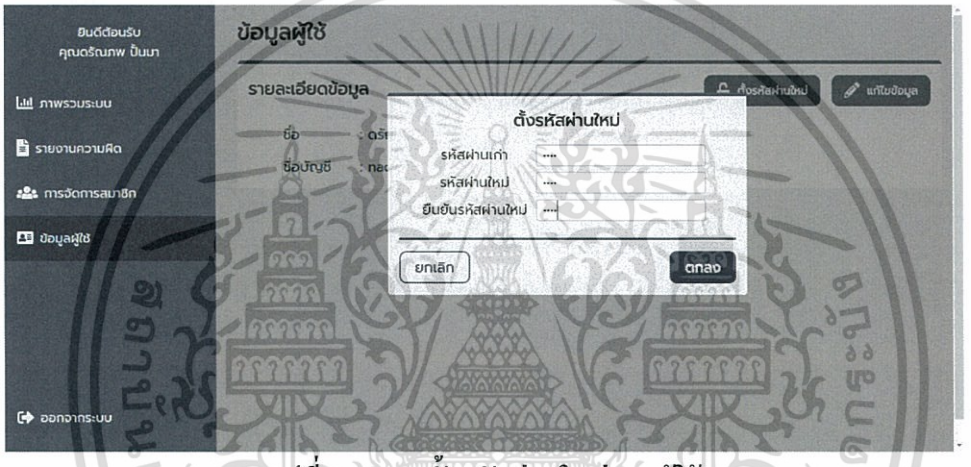


รูปที่ 4.26 รายละเอียดข้อมูลของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 การแก้ไขรายละเอียดต่าง ๆ ของผู้ใช้



รูปที่ 4.28 การตั้งรหัสผ่านใหม่ของผู้ใช้

4.5 สรุปผลการทดลอง

พบว่า React มีการใช้ทรัพยากร CPU และ memory น้อยกว่า Google API แต่จะใช้ทรัพยากร Network ทั้ง Network (Sent Bytes) และ Network (Received Byte) เท่า ๆ กัน ส่วนทางด้านอัลกอริทึมที่สร้างขึ้นก็สามารถคำนวณได้อย่างถูกต้อง แต่สามารถคำนวณได้เฉพาะรถที่วิ่งอยู่ในเส้นทางที่สร้างไว้เท่านั้น อีกทั้งถ้าหากข้อมูลที่ถูกส่งมามีความล่าช้ามากเกินไป ก็จะทำให้ไม่สามารถคำนวณข้อมูลช่วงนั้นได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

จากการทดลองและการศึกษาที่ผ่านมา สามารถนำมาสรุปเป็นประเด็นหลัก ๆ และนำข้อสรุปที่ได้มาวางแผนแนวทางการพัฒนาต่อได้ดังต่อไปนี้

5.1 การสร้างแผนที่บนเว็บไซต์ระหว่าง Google Maps API และ Google Maps React

ในการสร้างแผนที่บนเว็บไซต์ การสร้างด้วย Google API ต้องเขียนเป็น tag JavaScript (<script>) แทรกเข้าไปใน code HTML และทำการกำหนดค่าต่าง ๆ แล้วต้องทำการส่ง API ไปขอการใช้งานฟังก์ชันต่าง ๆ บน Google Maps API แต่การใช้ package google-maps-react นั้น แค่ประกาศ tag <GoogleMapReact> ก็สามารถสร้างแผนที่และเรียกใช้ฟังก์ชันต่าง ๆ บน Google Maps ได้ โดยใช้การเขียน code ที่น้อยกว่า

ในการ render จุด GPS 10,000 จุด การเขียนใช้ Google Maps API ต้องทำการลบจุดที่เคย render ไปรอบก่อนหน้าทุกครั้งเพื่อไม่ให้เห็นเป็นรูปจุดซ้อนจุด จึงดูไม่ smooth เท่าที่ควร แต่การ render ด้วย google-maps-react นั้นเนื่องจากการทำงานของ React เป็นการเปลี่ยนหน้า Frontend ของเว็บไซต์ตามที่เรากำหนด เราจึงเก็บจุด GPS 10,000 จุด เป็น state หนึ่งของ React และเมื่อค่าของจุด GPS มีการเปลี่ยนแปลง React จะ render หน้า Frontend ใหม่โดยที่ไม่ต้องลบจุดที่เคย render ไปในรอบก่อนหน้า รวมถึงยังทำให้หน้าเว็บไซต์ดู smooth มากกว่า การใช้ Google Maps API

5.2 การใช้อัลกอริทึมที่สร้างขึ้นไปตรวจสอบการกระทำผิดของรถประจำทางสาธารณะ

ในการยืนยันว่าอัลกอริทึมสามารถใช้ได้จริง จะต้องนำประวัติการขับที่ได้จากผลการตรวจสอบของอัลกอริทึม มาพล็อตเทียบกับแผนที่เพื่อสังเกตหาข้อผิดพลาด และเมื่อนำมาเทียบกันแล้ว พบว่าผลการตรวจสอบถูกต้องตามที่ควรจะเป็น จึงเป็นสิ่งพิสูจน์ได้ว่าอัลกอริทึมที่สร้างขึ้นสามารถนำไปใช้ได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ปัญหาอุปสรรคและแนวทางการแก้ไข

ปัญหาในเรื่องของข้อมูลรถประจำทางสาธารณะที่ส่งมาจาก data server โดยปัญหาที่พบคือ เราไม่สามารถนำข้อมูลมาคำนวณหรือแสดงผลแบบ real time ได้เนื่องจากข้อมูลจาก data server จะถูกส่งมาใหม่ทุก ๆ 1 นาที และอีกปัญหาหนึ่งที่พบคือ บางครั้งข้อมูลรถประจำทางที่เราต้องนำมาใช้ในการคำนวณหาการกระทำความผิดของรถประจำทางก็ไม่ถูกส่งมาเป็นเวลานานมาก และไม่สามารถบอกได้ว่าข้อมูลจะถูกส่งมาใหม่เมื่อใด จึงทำให้ log ของรถประจำทางที่เก็บไว้ใน database มีค่าก้าวกระโดด และทำให้การคำนวณคลาดเคลื่อน

เนื่องจากการคำนวณการกระทำความผิดของรถประจำทางสาธารณะต้องรัน server ที่ใช้ internet ตลอดเวลาแต่เนื่องด้วย internet มหาลัยบังคับให้ผู้ใช้งาน internet ต้องทำการ authentication ทุกครั้งเมื่อต้องการใช้ internet จึงทำให้บางครั้งเมื่อ หมดช่วง authentication ต้องไปทำการ authentication ใหม่จึงทำให้บางครั้งการคำนวณการกระทำความผิดของรถประจำทางสาธารณะขาดช่วง

5.4 แนวทางการพัฒนาต่อ

1) เพิ่มฟีเจอร์การคำนวณระยะเวลาของรถประจำทางสาธารณะกับตำแหน่งของป้ายรถประจำทางสาธารณะ ที่ผู้ใช้งานกำลังรอรถอยู่ ณ. ตอนนั้น โดยการคำนวณระยะทางระหว่างตำแหน่งของรถประจำทางสาธารณะกับตำแหน่งของผู้ใช้งาน แล้วหารด้วยความเร็วของรถประจำทางสาธารณะคันนั้น ๆ
2) เพิ่มจำนวนสายรถให้ครอบคลุมทั่วกรุงเทพฯ โดยการสร้างเส้นทางการเดินรถอย่างละเอียดของรถประจำทางสาธารณะทุกสาย
3) เพิ่มการแสดงข้อมูลของรถประจำทางสาธารณะบนแผนที่ โดยการนำข้อมูลของรถประจำทางสาธารณะของแต่ละคันมาแสดงพร้อมกันกับตำแหน่งของรถประจำทางสาธารณะคันนั้น ๆ
4) เพิ่มส่วนสรุปสถิติการกระทำผิดรายวัน โดยการนำข้อมูลของแต่ละวันมาคำนวณและแสดงออกมาเป็นผลสรุป
5) เพิ่มการแจ้งเตือนบนแผนที่เมื่อมีการกระทำผิดเกิดขึ้น โดยการเช็คความผิดกับข้อมูล Realtime ตลอดเวลา และเมื่อพบความผิด ก็แจ้งเตือนไปบนแผนที่ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Ivan Starkov. 2016. google-map-react.[Online].

Available: https://github.com/istarkov/google-map-react

Joy Warugu. 2017. Setup a React Environment Using webpack and Babel.[Online]

Available: https://scotch.io/tutorials/setup-a-react-environment-using-webpack-and-babel

Google Developers. 2017. Direction.[Online]

Available: https://developers.google.com/maps/get-started/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ก. โค้ดส่วนการทดลองเปรียบเทียบการ render GPS ของทั้ง Google API และ React

ก.1 Google API

1) ในส่วนของโค้ด Google API นั้น จะให้ทำการ render GPS 10,000 จุด โดยใช้การสุ่มจุด GPS ขึ้นมา 10,000 จุด โดยให้ค่า latitude อยู่ในช่วง 5.02 ถึง 19.92 และ longitude อยู่ในช่วง 97.02 ถึง 104.92 โดยใช้ฟังก์ชันสุ่มของ JavaScript และทำการเก็บค่าไว้ในตัวแปร data ตามรูปที่ ก.1

```
<script>
  function initMap() {
    var myLatLng = { lat: 13.7563, lng: 100.5018 };
    var data = []
    var gmarkers = []
    set = () => {
      var i = 0;
      for (i = 0; i < 10000; i++) {
        var xlat = (Math.random() * (19.920 - 5.0200) + 5.0200).toFixed(2)
        var ylng = (Math.random() * (104.920 - 97.0200) + 97.0200).toFixed(2)
        data.push({ lat: parseFloat(xlat), lng: parseFloat(ylng) })
      }
    }
    set()
```

รูปที่ ก.1 โค้ดการใช้ Google API

2) ทำการ render GPS 10,000 จุดลงบน map ผ่านฟังก์ชัน renderGPS ดังรูปที่ ก.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
renderGPS = () => {
  const showList = data.map((element, index) => {
    plotmarker = new google.maps.Marker({
      position: { lat: element.lat, lng: element.lng },
      map: map,
    })
    gmarkers.push(plotmarker);
  })
}
```

รูปที่ ก.2 โค้ดการ render GPS ของ Google API

3) ฟังก์ชัน addGPS จะทำการบวกค่า latitude และ longitude 0.01 ให้กับจุด GPS 10,000 จุดแต่ก่อนจะทำการวนลูปเพิ่มค่านั้น จะทำการเรียกใช้ฟังก์ชัน removeMarker ซึ่งทำหน้าที่ในการลบค่า marker ที่อยู่บน map ก่อนหน้านั้นให้หายไป ดังรูปที่ ก.3

```
addGPS = () => {
  removeMarker()
  let newGPS = data
  var i = 0
  for (i = 0; i < 10000; i++) {
    x[i].lat = parseFloat(x[i].lat) + parseFloat(0.01)
    x[i].lng = parseFloat(x[i].lng) + parseFloat(0.01)
  }
  data = newGPS
}

removeMarker = () => {
  for (i = 0; i < gmarkers.length; i++) {
    gmarkers[i].setMap(null);
  }
}
```

รูปที่ ก.3 โค้ดการเพิ่มค่า GPS และการลบจุด GPS เก่า ของ Google API

4) ส่วนสุดท้ายเป็นส่วนที่ ตั้งให้ ฟังก์ชัน addGPS และ renderGPS ทำงานเองอัตโนมัติทุก ๆ 5 วินาที ดังรูปที่ ก.4

```
this.interval = setInterval(this.addGPS, 5000);
this.interval = setInterval(this.renderGPS, 5000);
```

รูปที่ ก.4 โค้ดการเพิ่มค่า GPS และสร้างจุด GPS ใหม่ ทุก ๆ 5 วินาที ของ Google API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 React

1) ฟังก์ชัน setGPS เป็นขั้นตอนแรกที่จะทำการสุ่มจุด GPS มา 10,000 จุด โดย _generateRandomLat() และ _generateRandomLng() นั้นเป็นฟังก์ชันสุ่มค่า latitude กับ longitude ตามลำดับ แล้วทำการเก็บค่าที่สุ่มได้ทั้งหมดไว้ใน State.dataPlot ดังรูปที่ ก.5

```
setGPS = () => {
  var i = 0;
  var count = 0;
  for (i = 0; i < 10000; i++) {
    var xlat = _generateRandomLat().toFixed(3)
    var ylng = _generateRandomLng().toFixed(3)
    data.push({ lat: xlat, lng: ylng })
  }
  this.setState({ dataPlot: data })
}
```

รูปที่ ก.5 โค้ดการเก็บค่าการสุ่ม GPS ของ React

2) ฟังก์ชัน generateRandomLat จะทำการสุ่มค่าทศนิยมตั้งแต่ 5.02 ถึง 19.920 และฟังก์ชัน generateRandomLng จะทำการสุ่มค่าทศนิยมตั้งแต่ 97.02 ถึง 104.920 ส่วน ฟังก์ชัน add จะทำการบวกค่าที่ส่งเข้าไป 0.01

3) ฟังก์ชัน renderGPS จะทำการ render จุด GPS ที่ถูกเก็บไว้ใน State.dataPlot มีโครงสร้างภายในฟังก์ชัน ดังรูปที่ ก.6

```
renderGPS = list => {
  const showList = list.map((element, index) => {
    return (
      <img src={busStop} key={index} alt="bus stop" lat={element.lat} lng={element.lng} />
    )
  })
  return showList
}
```

รูปที่ ก.6 โครงสร้างภายในฟังก์ชัน renderGPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การเรียกใช้ฟังก์ชัน renderGPS โดยจะส่งค่า State.dataPlot เข้าไป โดยถ้าค่า State.dataPlot มีการเปลี่ยนแปลง จะทำการเรียกใช้ตัวเองใหม่ทุกครั้งอัตโนมัติ ดังรูป ก.7

```
<GoogleMapReact
  defaultCenter={this.state.center}
  defaultZoom={this.state.zoom}
  layerTypes={['TrafficLayer']}
  onGoogleApiLoaded={({ map, maps }) => { this.setState({ map: map, maps: maps, mapLoaded: true }
  yesIWantToUseGoogleMapApiInternals
  style={{ zIndex: -1 }}
>
  {this.renderGPS(this.state.dataPlot)}
</GoogleMapReact>
```

รูปที่ ก.7 โค้ดการนำฟังก์ชัน renderGPS มาใช้กับการเปลี่ยนแปลงของ State.dataPlot

5) ฟังก์ชัน addGps คือฟังก์ชันที่ใช้ในการบวกค่าของ GPS 10,000 จุด โดยจะบวกค่า latitude และ longitude 0.01 แล้วทำการเก็บค่าที่บวกเสร็จสิ้นทั้งหมดใส่ State.dataPlot โดยมีโครงสร้างของฟังก์ชันดังรูปที่ ก.8

```
addGps = () => {
  let newGPS = this.state.dataPlot
  var i = 0
  for (i = 0; i < 10000; i++) {
    x[i].lat = _add(newGPS[i].lat).toFixed(3)
    x[i].lng = _add(newGPS[i].lng).toFixed(3)
  }
  this.setState({ dataPlot: x })
}
```

รูปที่ ก.8 โครงสร้างภายในของฟังก์ชัน addGps

6) ฟังก์ชัน addGps จะถูกเรียกใช้ทุก ๆ 5 วินาที ดังรูปที่ ก.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
componentDidMount = () => {
  this.interval = setInterval(this.addGps, 5000);
}
```

รูปที่ ก.9 การเรียกใช้ฟังก์ชัน addGps ทุก ๆ 5 วินาที

ข. รายละเอียดต่าง ๆ ของ API

ข.1 roadData

ส่วนที่จัดการในเรื่องของการค้นหาสายรถ การค้นหาป้ายรถประจำทาง การค้นหาเส้นทางการเดินรถ ซึ่งมี API ต่าง ๆ ดังนี้

1) API GET roadData/

เป็น API ที่ใช้ดึงข้อมูล เส้นทางการวิ่งของรถประจำทาง

2) API GET roadData/busStop

เป็น API ที่ใช้ดึงข้อมูล ป้ายรถประจำทาง

3) API POST roadData/findRoadPath data{startBusStop,endBusStop}

เป็น API ที่ใช้เมื่อ user ต้องการทำการเส้นทางการเดินทางของรถประจำทาง โดยส่งค่าป้ายต้นทางและป้ายปลายทางที่ user ต้องการมาทำการประมวลผลเพื่อหาเส้นทางที่การวิ่งของรถที่สั้นที่สุด และตรวจสอบว่า user ส่งค่าป้ายต้นทางและป้ายปลายทาง ถูกหลักการเดินทางหรือไม่ (โดยป้ายปลายทางจะต้องอยู่ในตำแหน่งที่อยู่ถัดจากป้ายต้นทาง ไม่ได้อยู่ในตำแหน่งก่อนหน้าป้ายต้นทาง) หากพบว่า user เลือกใส่ป้ายต้นทางและป้ายปลายทาง ถูกต้องจะทำการส่ง array ที่เก็บลำดับเส้นทางของสายรถประจำทางที่ user ต้องการเดินทางกลับไป แต่หากพบว่า user ใส่ค่าป้ายต้นทางและป้ายปลายทางกลับมาไม่ถูกต้อง จะทำการส่ง ค่ากลับไปว่าไม่สามารถค้นหาเส้นทางได้

4) API POST roadData/roadPathway data(roadPath[ ],startBusStop,endBusStop)

เป็น API ที่ใช้เมื่อ user ทำการเลือกเส้นทางการวิ่งหลังจากเลือกเส้นทางการเดินทางแล้ว โดยจะต้องมีค่าที่ส่งมาคือ roadPath[] เป็น array ของเส้นทางการวิ่งของรถประจำทางที่ user เลือกป้ายต้นทางและป้ายปลายทาง โดยข้อมูลที่ส่งกลับไปคือค่า GPS ของเส้นทางการเดินทางที่ user เลือก

และมีรายละเอียดต่างของ request และ response ของแต่ละ API ดังตารางที่ ข.1 และ ข.2 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.1 รายละเอียดของ API Request ต่าง ๆ ของ roadData

API path	Type	Parameter	Example
roadData/	GET	None	-
roadData/busStop	GET	None	-

roadData/findRoadPath	POST	data: {
		startBusStop: String
		endBusStop: String
		}
			data: {
			startBusStop: “โรงเรียนหอวัง”,
			endBusStop: “อนุสาวรีย์ชัย สมรภูมิ”
			}

roadData/roadPathway	POST	data: {
		startBusStop: String,
		endBusStop: String,
		roadPath: Array,
		}
			data: {
			startBusStop: “โรงเรียนหอวัง”,
			endBusStop: “อนุสาวรีย์ชัย สมรภูมิ”,
			roadPath: [‘A1’,’39’’]
			}

ตารางที่ ข.2 รายละเอียดของ API Respond ต่าง ๆ ของ roadData

API path	Response data	Example

roadData/	roadData: {
	Name: String,
	Fullname: String,
	centerPath: object,
	busStopSequence,
	}
		roadData: {
		name: 39,
		Fullname: 39 ตลาดไท-อนุสาวรีย์ชัย สมรภูมิม
		…...
		}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API path	Response data	Example

roadData/busStop	busStop: [{
	busRoad: String,
	busStop: Array of Object
	}]
		busStop: [{
		busRoad: 39,
		busStop: [
		{
		nameTH: หน้าตลาดไท,
		lat: 14.0817653550191,
		lng: 100.61889152044,
		},
		…
		]}]

roadData/findRoadPath	{
	Result: String,
	roadPath: Array of Array
	roadType: String,
	}
		{
		Result: “success”,
		roadPath: [[‘A1’,’39’], …],
		roadType: “multiple”,
		}

roadData/roadPathway	{
	roadWay: Array of Object,
	}
		{
		roadWay: [
		{ lat: 13.81796, lng:100.55907},
		{ lat: 13.81626, lng: 100.55929},
		…..]
		}

ข.2 user

ส่วนที่จัดการในเรื่องของ การเพิ่ม user เข้าระบบ การแก้ไขข้อมูล user การลบข้อมูล user ออกจากระบบ การ login และ logout ของ user ซึ่งมี API ต่าง ๆ ดังนี้

1) API POST user/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น API ที่ใช้ในการดึงข้อมูล user จาก database โดยทางฝั่ง client ต้องส่ง token มาด้วย เพื่อเป็นการยืนยันว่าเป็น user ที่มีสิทธิเข้าสู่หน้าของเจ้าหน้าที่ได้

2) API POST user/addUser

เป็น API ที่ใช้เพิ่ม user เข้าสู่ระบบโดยฝั่ง client ต้องส่ง token และ ต้องส่ง status มายืนยันว่า status ของ user เป็น admin หรือ assistant ด้วย จึงจะสามารถเพิ่ม user เข้าสู่ระบบได้

3) API POST user/removeUser

เป็น API ที่ใช้ลบ user ออกจากระบบ โดยฝั่ง client ต้องส่ง token และ ต้องส่ง status มายืนยันว่า status ของ user เป็น admin หรือ assistant ด้วย จึงจะสามารถลบ user ออกจากระบบได้

4) API POST user/editStatusUser

เป็น API ที่ใช้ในการเปลี่ยน status ของ user โดย client ต้องส่ง token และต้องส่ง status มายืนยันว่า status ของ user เป็น admin หรือ assistant ด้วย จึงจะสามารถ status ของ user ได้

5) API POST user/editPasswordUser

เป็น API ที่ใช้ในการเปลี่ยนรหัสของ user โดย client ต้องส่ง token และ รหัสเก่ามาเพื่อยืนยันก่อนที่จะทำการเปลี่ยนรหัสใหม่เข้า database

6) API POST user/editProfileUser

เป็น API ที่ใช้ในการเปลี่ยนข้อมูล profile ของ user โดย client ต้องส่ง token มาเพื่อยืนยันตัวตนของ user

7) API POST user/login

เป็น API ที่ใช้ในการ ตรวจสอบ username password ของ user ก่อนจะเข้าสู่ระบบ โดยเมื่อตรวจสอบแล้วพบว่า username และ password ถูกต้องจะทำการส่ง token กลับไปยังฝั่ง web browser

และมีรายละเอียดต่าง ๆ ของ request และ response ของแต่ละ API ดังตารางที่ ข.3 และ ข.4 ตามลำดับ

ตารางที่ ข.3 รายละเอียดของ API Request ต่าง ๆ ของ user

API path	Type	Parameter	Example

user/	GET	data: {
		Username: String,
		Token: String,
		}
			data: {
			username: “Thammawat”,
			token: “$2a$10$GqY.nQN….”,
			}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API path	Type	Parameter	Example

user/addUser	POST	data: {
		adminUsername: String,
		Token: String,
		adminStatus: String,
		Username: String,
		Password: String,
		Firstname: String,
		Lastname: String,
		Status: String,
		}
			data: {
			adminUsername: “admin”,
			Token: “$2a$10$GqY.nQN….”,
			adminStatus: “admin”,
			Username: “Thammawat”,
			Password: “1212312121”,
			Firstname: “ธรรมวัฒน์”,
			Lastname: “บงการณ์”,
			Status: “member”,
			}

user/removeUser	POST	data: {
		adminUsername: String,
		Token: String,
		adminStatus: String,
		username: String,
		}
			data: {
			adminUsername: “admin”,
			Token: “$2a$10$GqY.nQN….”,
			adminStatus: “admin”,
			username: “thammawat”,
			}

user/editStatusUser	POST	data: {
		adminUsername: String,
		Token: String,
		username: String,
		userStatus: String,
		}
			data: {
			adminUsername: “admin”,
			Token: “$2a$10$GqY.nQN….”,
			username: “Thammawat”,
			userStatus: “admin”,
			}

user/editPasswordUser	POST	data: {
		username: String,
		Token: String,
		password: String,
		newPassword: String
		}
			data: {
			username: “admin”,
			Token: “$2a$10$GqY.nQN….”,
			password: “345345434534”,
			newPassword: “1221213213”
			}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API path	Type	Parameter	Example

user/editProfileUser	POST	data: {
		username: String,
		token: String,
		newUsername: String,
		newFirstname: String,
		newLastName: String
		}
			data: {
			username: “admin”,
			Token: “$2a$10$GqY.nQN….”,
			newUsername: “Newadmin”,
			newFirstname:
			“NewFirstName”,
			newLastName:
			“NewLastName”
			}

user/login	POST	data: {
		username: String,
		password: String,
		}
			data: {
			username: “admin”,
			Password: “1212312121”
			}

ตารางที่ ข.4 รายละเอียดของ API Respond ต่าง ๆ ของ user

API path	Response data	Example

user/	{
	user: Array of Object,
	}
		{
		user: [{
		firstname: “ธรรมวัฒน์”, lastname: “บงการณ์”,
		statue:”member”,
		},
		...
		]}

user/addUser	{
	result: String,
	}
		{
		result: “User has Created”
		}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API path	Response data	Example

user/removeUser	{
	result: String,
	}
		{
		result: “User remove success”
		}

user/editStatusUser	{
	result: String,
	}
		{
		result: “User edit success”
		}

user/editPasswordUser	{
	result: String,
	}
		{
		result: “User change password success”,
		}

user/editProfileUser	{
	result: String,
	}
		{
		result: “User edit profile success”,
		}

user/login	{
	result: String,
	token: String,
	}
		{
		result: “User edit profile success”,
		token: “$2a$10$GqY.nQN….”,
		}

ข.3 busGulity

ส่วนที่จัดการในเรื่องการรับข้อมูลการกระทำความผิดของรถประจำทางสาธารณะจาก database มาแสดงบนหน้าเว็บไซต์ ซึ่งมีรายละเอียดต่าง ๆ ของ request และ response ของ API นี้ดังตารางที่ ข.5 และ ข.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข.5 รายละเอียดของ API Request ต่าง ๆ ของ busGulity

API path	Type	Parameter	Example

/busGulity	POST	data: {
		username: String,
		token: String,
		}
			data: {
			username: “admin”,
			Token: “$2a$10$GqY.nQN….”,
			}

ตารางที่ ข.6 รายละเอียดของ API Respond ต่าง ๆ ของ busGulity

API path	Response data	Example

/busGulity	{
	busGulity: Array of object,
	}
		{
		busGulity: [{
		path: “97”,
		busID: “12-5581”,
		yype: “ขับออกนอกเส้นทาง”,
		timeStamp: “2018-04-01 16:34:55”,
		lat: 13.80469,
		lng: 100.280046666667,
		buslock: [
		{
		lat :13.80469,
		lng: 100.280046666667
		},
		{
		lat:13.80469,
		lng: 100.280046666667
		},
		…
		]}, ...]}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้