ระบบวิเคราะห์การขึ้นลงรถเมล์ แบบเรียลไทม์
กรณีศึกษา บริษัท เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน)
IOT BIG DATA ANALYSIS OF PUBLIC BUS REAL TIME TRANSIT
CASE STUDY METRO SYSTEM CORPORATION CO LTD.

นาย วีรชัย วิเชียรรัตน์

โครงงานสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยี ไทย-ญี่ปุ่น
พ.ศ. 2561

ระบบวิเคราะห์การขึ้นลงรถเมล์ แบบเรียลไทม์

กรณีศึกษา บริษัท เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน)

IOT BIG DATA ANALYSIS OF PUBLIC BUS REAL TIME TRANSIT

CASE STUDY METRO SYSTEM CORPORATION CO LTD.

นาย วีรชัย วิเชียรรัตน์

โครงงานสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย - ญี่ปุ่น

ปีการศึกษา 2561

คณะกรรมการสอบ

..... ประธานกรรมการสอบ

(ผู้ช่วยศาสตราจารย์ ตรีรัตน เมตต์การุณ์จิต)

..... กรรมการสอบ

(อาจารย์ ดร . ปราณิสา อิศรเสนา)

..... อาจารย์ที่ปรึกษา

(อ.ลลิตา ณ หนองคาย)

..... ประธานสหกิจศึกษาสาขาวิชา

(อาจารย์ สลิลา ชีวกิดาการ)

สถาบันเทคโนโลยีไทย-ญี่ปุ่น
THAI - NICHI INSTITUTE OF TECHNOLOGY
TNI

ชื่อโครงงาน	ระบบวิเคราะห์การขึ้นลงรถเมล์ แบบเรียลไทม์ กรณีศึกษา บริษัท เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน) IoT Bigdata Analysis of Public bus real time transit Case Study Metro System Corporation Co Ltd.
ผู้เขียน	นาย วีรชัย วิเชียรรัตน์
คณะวิชา	เทคโนโลยีสารสนเทศ สาขาวิชา เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษา	อาจารย์ ลลิตา ณ หนองคาย
พนักงานที่ปรึกษา	นาย ศุภกร ฟุ้งเกียรติขจร
ชื่อบริษัท	เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน)
ประเภทธุรกิจ/สินค้า	Web Application, Software Application

บทสรุป

จากการปฏิบัติสหกิจศึกษา ได้รับมอบหมายให้ การทดสอบประสิทธิภาพการใช้งานของเว็บแอปพลิเคชันที่บริษัทกำลังทำการพัฒนาอยู่ เพื่อให้เว็บเซิร์ฟเวอร์สามารถรองรับการเข้าใช้งานของผู้ใช้งานระบบในจำนวนที่มากได้ และเพื่อทำให้ระบบมีประสิทธิภาพการใช้งานที่คงที่ ไม่พบปัญหาในการเข้าใช้งานเป็นระยะเวลาต่อเนื่องที่มีการเข้าใช้งาน การทำการทดสอบที่ได้ฝึกทำนั้น

ปัจจุบันความก้าวหน้าทางด้านเทคโนโลยีของมนุษย์มีมากขึ้น มีการพัฒนาเทคโนโลยีใหม่ ขึ้นมาใช้งานกันอย่างแผร่หลาย สร้างความสะดวกและ รวดเร็วให้แก่ผู้ใช้งาน ทั้งนี้ทางบริษัท เล็งเห็นความสำคัญของเทคโนโลยีทางด้าน Big data และ ปัญหาของรถเมล์ในปัจจุบัน จึงได้พัฒนา ระบบวิเคราะห์การขึ้นลงรถเมล์ แบบเรียลไทม์ ขึ้นเพื่อ แสดงผลวิเคราะห์ข้อมูล ปัญหาที่นั่งรถเมล์เต็ม ในปัจจุบัน

การสหกิจศึกษาครั้งนี้ ได้รับความรู้มากมายและได้รับประสบการณ์จากปัญหาที่เกิดขึ้น รวมถึงวิธีการแก้ไขปัญหาที่เกิดขึ้นนั้นด้วย ซึ่งความรู้ที่ได้จากการทำการทดสอบทั้งหมด สามารถนำไปใช้ได้ในชีวิตจริง ไม่ว่าจะเป็นการเขียนโค้ดภาษาจาวา การใช้โปรแกรม การตรวจสอบและเตรียมความพร้อม การวางแผนการทำการทดสอบ และการวิเคราะห์ข้อมูลที่ได้จากการทดสอบ

กิตติกรรมประกาศ

จากการปฏิบัติสหกิจศึกษาที่ ณ บริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่นจา กัด (มหาชน) นับตั้งแต่ วันที่ 04 พฤษภาคม พ.ศ. 2561 จนถึง 28 กันยายน พ.ศ. 2561 ทา ให้ได้รับความรู้และ ประสบการณ์จากการทางานจริงที่สำคัญ ต่อการพัฒนาตัวเองในอนาคต และ สำหรับรายงาน ปฏิบัติงานสหกิจนี้ ขอกราบขอบคุณผู้ที่ให้ความรู้และสนับสนุนการฝึกสหกิจจนสำเร็จลุล่วงไป ได้ด้วยด้วยดี

และสุดท้ายนี้ขอกราบขอบคุณพี่ๆ ทุกท่านใน บริษัทเมโทรซิสเต็มส์คอร์ปอเรชั่นจำากัด (มหาชน) และ อาจารย์ ลลิตา ณ หนองคาย ที่ให้ความช่วยเหลือและ ให้คำปรึกษาต่าง ตลอดทั้ง 4 เดือน ในการปฏิบัติสหกิจที่ผ่านมาจนสามารถสำเร็จ ลุล่วงไปได้ด้วยดี

นาย วีรชัย วิเชียรรัตน์

ผู้จัดทำ

สารบัญ

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญภาพ

ภาพที่ **หน้า**

สารบัญภาพ(ต่อ)

สารบัญตาราง

ตาราง **หน้า**

บทที่ 1
บทนำ

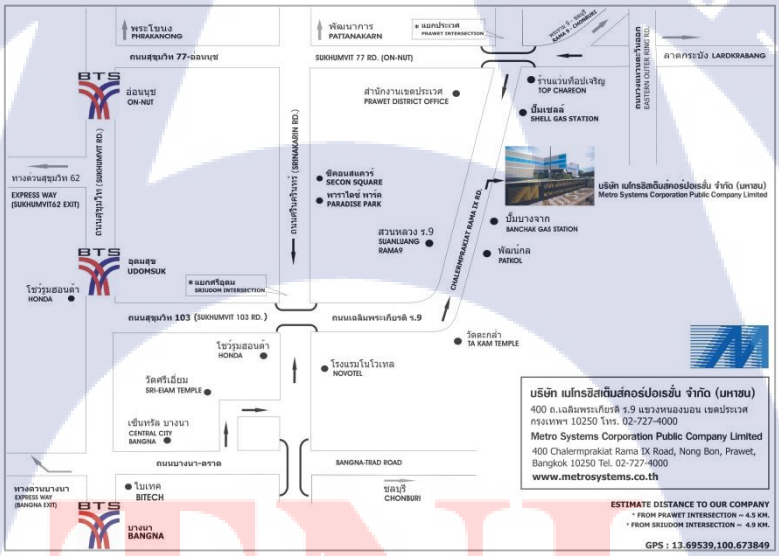
1.1 ชื่อและที่ตั้งของสถานประกอบการ

ชื่อหน่วยงาน บริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน)
(Metro Systems Corporation Public Company Limited)

ที่ตั้ง เลขที่ 400 ถนนเฉลิมพระเกียรติ ร.9 แขวงหนองบอน เขตประเวศ กรุงเทพมหานคร 10250

เว็บไซต์ http://www.metrosystems.co.th

อีเมล์ recruitment@metrosystems.co.th



ภาพที่ 1.1 สถานที่ตั้ง บริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน)

1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร

บริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน)ได้ก่อตั้งขึ้นในปี พ.ศ. 2529 ด้วยทุนจดทะเบียน 4 ล้านบาท เริ่มต้นดำเนินธุรกิจจากการเป็นตัวแทนจำหน่ายผลิตภัณฑ์ไอบีเอ็มเป็นรายแรกในภูมิภาคเอเชียตะวันออกเฉียงใต้ ต่อมาได้มีการขยายการจัดจำหน่ายโซลูชั่นและบริการจากบริษัทไอทีชั้นนำอื่น ๆ เพื่อการบริการที่ครบวงจรและครอบคลุมความต้องการของลูกค้าได้มากยิ่งขึ้น

ในปี 2538 บริษัทได้เพิ่มทุนจดทะเบียนเป็น 180 ล้านบาท เพื่อปรับเปลี่ยนรูปแบบธุรกิจโดยจดทะเบียนเป็นบริษัทมหาชนในตลาดหลักทรัพย์แห่งประเทศไทยและทำการซื้อขายหุ้นสามัญครั้งแรกเมื่อวันที่ 9 พฤษภาคม พ.ศ. 2539 ตลอดระยะเวลาการดำเนินงาน บริษัทมีการปรับเปลี่ยนมูลค่าทุนจดทะเบียนให้ตามความเหมาะสมกับขนาดของธุรกิจ โดย ณ ปัจจุบันบริษัทมีทุนจดทะเบียน 360 ล้านบาท

ภายหลังการได้รับรองระบบคุณภาพ ISO 9001: 2008 ในทุกขั้นตอนการทำงาน และระบบ ISO/IEC 20000:2005 และ ISO/IEC 27001:2005 ซึ่งเกี่ยวข้องกับการป้องกันสารสนเทศในองค์กรที่อาจส่งผลต่อการดำเนินงาน ทำให้ในปี 2554 ทางบริษัทได้เดินหน้าจัดทำระบบบริหารความต่อเนื่องทางธุรกิจ ((Business Continuity Management System : BCMS) และกำหนดเพิ่มเติมไว้ในพันธกิจหลักขององค์กร กล่าวคือ “...การมุ่งมั่นสู่ความเป็นเลิศและบริหารความต่อเนื่องในธุรกิจเทคโนโลยีสารสนเทศครบวงจร โดยบุคลากรระดับมืออาชีพ เพื่อความสำเร็จของลูกค้า ตลอดจนมีส่วนร่วมในการพัฒนาสังคมแห่งความรู้ ” ทั้งนี้ เป็นการป้องกันบรรเทาอุบัติการณ์ที่อาจก่อให้เกิดความเสียหายต่อธุรกิจ และทำให้บริษัทสามารถดำเนินธุรกิจได้อย่างต่อเนื่อง ซึ่งมุ่งเน้นการดูแลความปลอดภัย การปฏิบัติตามกฎหมาย กฎระเบียบ ข้อตกลงที่เกี่ยวข้อง ตลอดจนดูแลตรวจสอบความถูกต้องของข้อมูล ภาพลักษณ์และความน่าเชื่อถือของบริษัท

ด้วยแนวคิดที่ยึดเอาลูกค้าเป็นศูนย์กลาง บริษัทฯ จึงมุ่งแสวงหาแนวทางใหม่ๆ เพื่อตอบสนองความต้องการของลูกค้า และยังคงเดินหน้าแสวงหาผลิตภัณฑ์และบริ การต่างๆ เพื่อให้ทันกับการเปลี่ยนแปลงของอุตสาหกรรมไอที การมุ่งแสวงหาบริการที่จะทำให้ลูกค้าดำเนินธุรกิจได้อย่างมีประสิทธิภาพและต่อเนื่องทั้งในสภาวะปกติ หรือในยามเผชิญวิกฤต และยังให้ความสำคัญในการการร่วมมือกับพันธมิตรธุรกิจเพื่อนำเสนอโซลูชั่นให้ครอบคลุมความต้องการของกลุ่มลูกค้า บริษัทฯ จะยังคงเดินหน้าพัฒนาบุคลากร และผสมผสานความสามารถของซัพพลายเออร์ คู่ค้า ตลอดจนนวัตกรรมใหม่ๆ ที่เกี่ยวข้อง เพื่อตอบสนองความต้องการ สร้างความสำเร็จ และสร้างคุณค่าเพิ่มให้แก่ธุรกิจของลูกค้า

จากการเติบโตทางธุรกิจอย่างรวดเร็ว ทำให้บริษัท เมโทรซิสเต็มส์ คอร์ปอเรชั่น จำกัด (มหาชน) ได้รับมอบรางวัล ” IBM Business Partner of the Year” ซึ่งเป็นรางวัลที่ได้รับติดต่อกันเป็นเวลาถึง 25 ปี พร้อมยังได้รับรางวัล IBM The Longest Relation Business Partner, Citrix - Platinum Partner of the Year และรางวัลอื่นๆ จากคู่ค้าอีกมากมาย

ภาพที่ 1.2 Metro Proud Awards

ปัจจุบันบริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน) มีประเภทของผลิตภัณฑ์และการบริการ ซึ่งสามารถแบ่งออกเป็น 3 กลุ่มใหญ่ๆ ได้แก่

12.1 กลุ่มธุรกิจดิจิตอลโซลูชั่น

จัดจำหน่ายผลิตภัณฑ์ฮาร์ดแวร์ ประกอบด้วยเซิร์ฟเวอร์ สตอเรจ คอมพิวเตอร์ตั้งโต๊ะ โน้ตบุ๊ค เวิร์คสเตชั่น เครื่องพิมพ์ รวมถึงการให้บริการออกแบบและติดตั้งระบบคอมพิวเตอร์ ซอฟต์แวร์และโซลูชั่นต่างๆ อาทิ ระบบค้าปลีก (เครื่อง Point-Of-Sale และซอฟต์แวร์) ระบบ Video Surveillance (IP Camera และซอฟต์แวร์ระบบรักษาความปลอดภัย) รวมถึงโซลูชั่นเพื่อการ กู้คืนระบบ ระบบความปลอดภัย โมบิลิตี้

1.2.2 กลุ่มธุรกิจซอฟต์แวร์โซลูชั่น

จัดจำหน่ายซอฟต์แวร์โซลูชั่น พร้อมให้บริการติดตั้งระบบ และบริการให้คำปรึกษาด้านไอทีเพื่อการบริหารธุรกิจที่มีประสิทธิภาพ

1.2.3 กลุ่มธุรกิจซอฟต์แวร์โซลูชั่น

จัดจำหน่ายซอฟต์แวร์โซลูชั่น พร้อมให้บริการติดตั้งระบบ และบริการให้คำปรึกษาด้านไอทีเพื่อการบริหารธุรกิจที่มีประสิทธิภาพ

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร

ภาพที่ 1.3 คณะผู้บริหารบริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน)

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง นักศึกษาสหกิจ

หน้าที่งานที่ได้รับมอบหมาย Programmer

1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา

ชื่อ – นามสกุล นาย ศุภกร ฟุ้งเกียรติขจร

ตำแหน่ง Program Developer

1.6 ระยะเวลาปฏิบัติงาน

ปฏิบัติงานสหกิจศึกษาเป็นระยะเวลา 4 เดือน

นับตั้งแต่วันที่ 4 พฤษภาคม 2561 – 28 กันยายน 2561

1.7 ที่มาและความสำคัญของปัญหา

เนื่องจากปัจจุบัน รถเมล์ในประเทศไทย ขาดระยะบ่อย รอนานเกิน 2 – 3 ชม.โดยเฉพาะชั่วโมง เร่งด่วน ที่ไม่สามารถขึ้นรถได้เนื่องจาก จำนวนเต็มคันรถ ไม่สามารถขึ้นได้ ต้องรอคันถัดไป ซึ่งส่วนนี้มีคนร้องเรียน เป็นจำนวนมาก รวมถึงรถเมล์ มาไม่ตรงเวลา ทั้งนี้ ขึ้นอยู่กับหลายปัจจัยต่าง อาทิ เส้นทางยาวเกินไป หรือ เส้นทางทับซ้อนกันมากไป หรือ รถติด และ รถไม่พอ

1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงงาน

1.8.1 เพื่อศึกษาข้อมูลผู้ใช้บริการรถเมล์สาย 11 เล็ก

1.8.2 เพื่อวิเคราะห์ข้อมูลการขึ้น-ลง ของ ผู้ใช้บริการรถเมล์สาย 11 เล็ก

1.8.3 เพื่อวิเคราะห์ข้อมูลช่วงเวลาของผู้ใช้บริการรถเมล์สาย 11 เล็ก

1.8.4 เพื่อศึกษาเรียนรู้การทำงาน ประสบการณ์จริงจากสถานประกอบการ

1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงาน หรือโครงงานที่ได้รับมอบหมาย

1.9.1 ได้ข้อมูลผู้ใช้บริการรถเมล์สาย 11เล็ก

1.9.2 ได้ผลวิเคราะห์ข้อมูลการขึ้น-ลง ของ ผู้ใช้บริการรถเมล์สาย 11 เล็ก

1.9.3 ได้ผลวิเคราะห์ข้อมูลช่วงเวลาของผู้ใช้บริการรถเมล์สาย 11เล็ก

1.9.4 ได้ศึกษาเรียนรู้การทำงาน ประสบการณ์จริงจากสถาณประกอบการ

1.10 นิยามศัพท์เฉพาะ

1.10.1 รถเมล์ สาย 11 หมายถึง รถเมล์สาย 11 เล็ก ที่วิ่งจาก อู่เมกาบางนา ถึง มาบุญครอง

1.10.2 ข้อมูลผู้ใช้บริการรถเมล์ หมายถึง จำนวนคนที่ขึ้น ลง รถเมล์สาย 11 เล็ก เท่านั้น

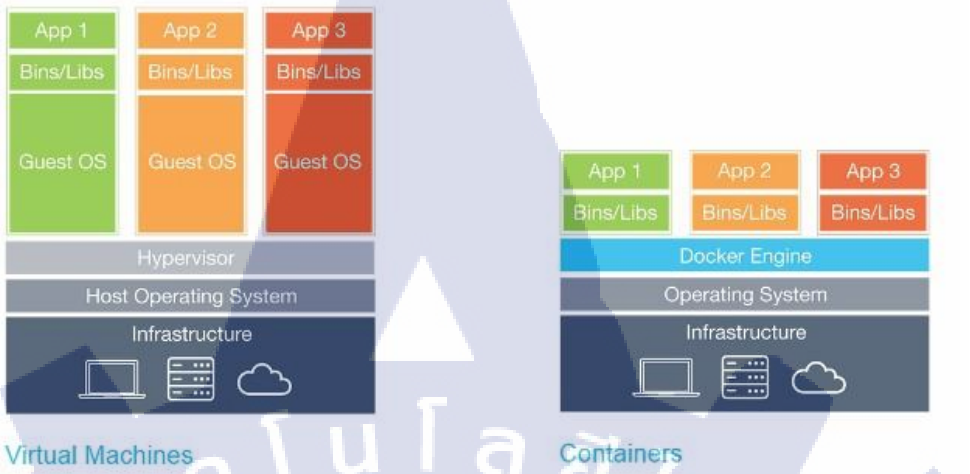
บทที่ 2
ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติ

2.1 เทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1.1 Docker

Docker[1] คือ engine ตัวหนึ่งที่มีการทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่อง server เพื่อใช้ในการ run service ที่ต้องการ มีการทำงานคล้ายคลึงกับ Virtual Machine เช่น VMWare, VirtualBox, XEN, KVM แต่ข้อแตกต่างที่ชัดเจนคือ Virtual Machine ที่รู้จักกันก่อนหน้านี้นั้น เป็นการจำลองทั้ง OS เพื่อใช้งานและหากต้องการใช้งาน service จึงต้องทำการติดตั้งเพิ่มเติมบน OS นั้น แต่สำหรับ docker แล้วจะใช้ container ในการจำลองสภาพแวดล้อมขึ้นมา เพื่อใช้งานสำหรับ 1 service ที่ต้องการใช้งานเท่านั้น โดยไม่ต้องมีส่วนของ OS เข้าไปเกี่ยวข้องเหมือน Virtual Machines อื่นๆ

ภาพที่ 2.1 ไอคอนของโปรแกรม Docker



ภาพที่ 2.2 รูปภาพ ความแตกต่างระหว่าง Virtual Machine กับ Container

องค์ประกอบของ Docker

1) **ความหมายของ Docker Image** คือต้นแบบของ Container ข้างในจะเป็น Linux ที่มีการติดตั้ง Application และ มีการ Configuration เอาไว้ ซึ่งเกิดมาจากการ build ไฟล์ Docker file ขึ้นมาเป็น image
2) **ความหมายของ Docker container** คือ Container จะถูกสร้างมาจาก Docker Image ที่เป็นต้นแบบ เกิดเป็น Container จะได้ Service หรือ Application ที่สามารถเรียกใช้งานได้ทัน
3) **ความหมายของ Docker registry** คือ สามารถสร้าง Docker Image แล้วนำไปเก็บรวบรวมไว้บน server (ลักษณะเดียวกับการเก็บ Source Code ไว้บน Github) โดย Docker registry ปัจจุบันก็มีให้เลือกใช้งานได้หลากหลายโดยมี Docker Hub เป็น Docker registry หลักในการเรียกใช้(pull) Docker Image และนอกจากนี้ยังมีผู้ให้บริการ docker registry เจ้าอื่นๆ ด้วย เช่น Gitlab, Quay.io, Google Cloud เป็นต้น

2.1.2 Apache Elasticsearch

Shay Banon ผู้สร้าง Elasticsearch [2] โดยก่อนหน้า เรียกว่า Compass สร้างขึ้นจากพื้นดินขึ้นเพื่อกระจาย "และใช้อินเตอร์เฟซทั่วไป JSON ผ่าน HTTP เหมาะสำหรับการเขียนโปรแกรมภาษาอื่นนอกจาก Java เช่นกัน Shay Banon ได้เปิดตัว Elasticsearch รุ่นแรกในเดือนกุมภาพันธ์ 2010 Elasticsearch BV ก่อตั้งขึ้นในปีพศ. 2555 เพื่อให้บริการเชิงพาณิชย์และผลิตภัณฑ์เกี่ยวกับ Elasticsearch และซอฟต์แวร์ที่เกี่ยวข้อง ในเดือนมิถุนายน 2014 บริษัท ได้ประกาศระดมทุน 70 ล้านเหรียญในรอบการระดมทุนซีรี่ส์ซีเพียง 18 เดือนหลังจากก่อตั้ง บริษัท รอบนี้นำโดย New Enterprise Associates (NEA) นักลงทุนเพิ่มเติม ได้แก่ Benchmark Capital และ Index Ventures รอบนี้จะทำให้เงินทุนทั้งหมดรวมอยู่ที่ 104 ล้านเหรียญ ในเดือนมีนาคม 2015 บริษัท Elasticsearch เปลี่ยนชื่อเป็น Elastic ในเดือนมิถุนายนปีพ. ศ. 2561 บริษัท Elastic ได้ยื่นขอเสนอขายหุ้นสามัญครั้งแรกในราคาประมาณ 1.5 ถึง 3 พันล้านดอลลาร์ เมื่อวันที่ 5 ตุลาคม พ.ศ. 2561 Elastic ได้เข้าจดทะเบียนในตลาดหลักทรัพย์นิวยอร์ก

Elasticsearch คือ ที่เก็บข้อมูลที่พัฒนาต่อยอดมาจาก Apache Lucene มีจุดเด่นในเรื่องความสามารถของการค้นหา และสรุปข้อมูลขนาดใหญ่ได้อย่างรวดเร็ว

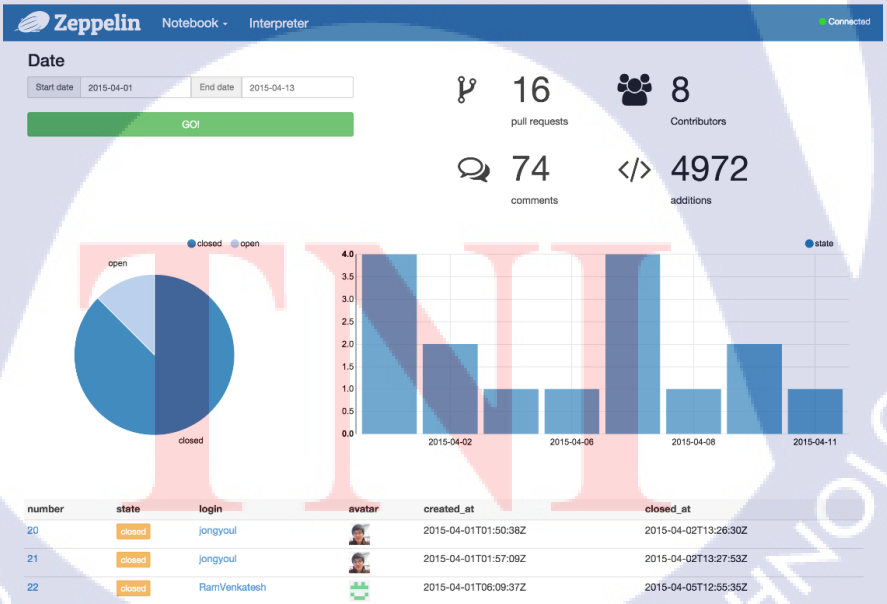
Elasticsearch ได้รับการพัฒนาในภาษาจาวาและได้รับการเผยแพร่เป็นโอเพนซอร์สภายใต้เงื่อนไขของสัญญาอนุญาต Apache ลูกค้าอย่างเป็นทางการมีอยู่ใน Java, .NET (C #), PHP, Python, Apache Groovy, Ruby และอีกหลายภาษา ตามการจัดอันดับเครื่องยนต์ดีบี Elasticsearch เป็นเครื่องมือค้นหาองค์กรที่เป็นที่นิยมใช้มากที่สุดรองลงมาคือ Apache Solr ซึ่งใช้ Lucene

Elasticsearch มีการทำ index ข้อมูลไว้ในทุกๆ field ข้อมูล ทำให้ค้นหาและสรุปข้อมูลข้อมูลขนาดใหญ่ได้อย่างรวดเร็วใกล้เคียงกับการเข้าถึงข้อมูลแบบ Near Real-time และ Elasticsearch เลือกใช้การเก็บข้อมูลในรูปแบบ JSON (JavaScript Object Notation) ซึ่งเป็นรูปแบบมาตรฐานข้อมูลที่ใช้งานได้ง่าย รับส่งข้อมูลได้หลากหลายแพลตฟอร์ม

ภาพที่ 2.3 สัญลักษณ์ของ Apache Elasticsearch

2.1.3 Zeppelin

Zeppelin[3] เป็น Web-Base Notebook สำหรับเป็นการเชื่อมต่อกับระบบ Data Analytic สามารถใช้ทำ Data-Driven แบบ Interactive และยังสามารถใช้งานพร้อมๆกันหลายคนได้ รองรับการทำงานผ่านภาษา SQL, Scala และอื่นๆ อีกมากมาย มันคือ Web-based notebook (คล้ายๆ กับ Jupyter Notebook) จะเป็นการเขียน SparkSQL, Scala, PySpark เพื่อ process big data ที่อยู่บน Spark Cluster โดย zeppelin จะมี built-in charts มาให้เราอยู่แล้ว 4 แบบคือ Pie Chart, Bar Chart, Area Chart และ Line Chart



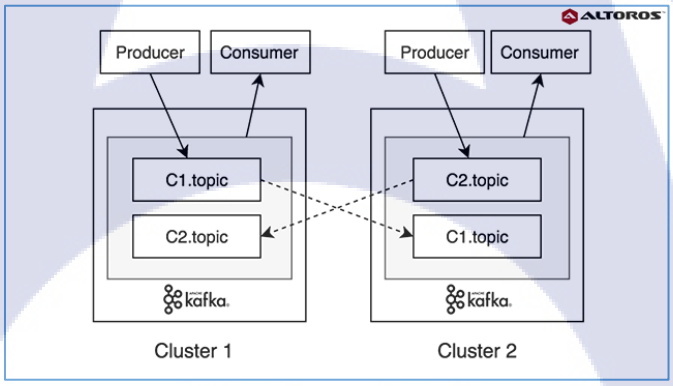
ภาพที่ 2.4 ตัวอย่าง หน้าตาของโปรแกรม Zeppelin

2.1.4 Apache Kafka

Apache Kafka [4] เป็นระบบกระจายข้อความ(message, event, log, ฯลฯ) ที่ถูกออกแบบให้เป็นศูนย์กลางของการส่งข้อความในองค์กรขนาดใหญ่ ที่ถูกพัฒนาโดย LinkedIn และปัจจุบันเป็นซอฟท์แวร์เสรี (open source)

Kafka ถูกออกแบบมาให้เร็ว, ปรับขยายได้ง่าย, มีความคงทนของข้อมูล, มี High Availability (ทำงานกันเป็น cluster ไม่ล่มง่ายๆ), และสามารถทำ Disaser Recovery ได้

โดยเริ่มแรก Kafka ถูกสร้างขึ้นโดย LinkedIn เป็น open source project ด้วยภาษา Java และ Scalaในช่วงต้นปี 2011 และถูกเผยแพร่ต่อ ผ่านทาง Apache Incubator ตั้งแต่ปี 2012 จากนั้นจึงได้แยกบริษัทออกมาจาก LinkedIn ก่อตั้งเป็น บริษัท Confluent เพื่อพัฒนา Kafka โดยเฉพาะ โดยชื่อ Kafka มาจากนักเขียนนาม Franz Kafka โดยเลือกชื่อ Kafka เพราะมันถูก optimize สำหรับการเขียน เหมือนกับงานของ Franz Kafka



ภาพที่ 2.5 แสดงกระบวนการทำงานของ Apache Kafka

2.1.5 Apache Spark

Apache Spark[5] มีพื้นฐานทางสถาปัตยกรรมที่ยืดหยุ่นในการแจกจ่ายชุดข้อมูล (RDD) ซึ่งเป็นชุดข้อมูล multiset แบบอ่านอย่างเดียวที่แจกจ่ายให้กับกลุ่มเครื่องที่เก็บรักษาไว้ในลักษณะที่ผิดพลาด ใน Spark 1.x RDD เป็นโปรแกรมประยุกต์สำหรับการเขียนโปรแกรมส่วนติดต่อหลัก (API) แต่เมื่อใช้ Spark 2.x ของ Dataset API ได้รับการสนับสนุน แม้ว่า RDD API จะไม่เลิกใช้งานก็ตาม เทคโนโลยี RDD ยังคงอยู่ภายใต้ Dataset API

Spark และ RDDs ได้รับการพัฒนาในปี 2012 ในการตอบสนองต่อข้อ จำกัด ในกระบวนทัศน์คอมพิวเตอร์คลัสเตอร์ MapReduce ซึ่งกำลังโครงสร้างเชิงเส้น dataflow โดยเฉพาะอย่างยิ่งเกี่ยวกับโปรแกรมการกระจาย: โปรแกรม MapReduce อ่านข้อมูลการป้อนข้อมูลจากดิสก์ map ฟังก์ชั่นในทุกข้อมูลที่ลดผลของ แผนที่และจัดเก็บผลการลดลงบนดิสก์ ฟังก์ชั่น RDDs ของ Spark เป็นชุดทำงานสำหรับโปรแกรมแจกจ่ายที่มีรูปแบบที่ จำกัด การแจกจ่ายหน่วยความจำแบบกระจาย (จงใจ)

Spark ช่วยให้สามารถใช้อัลกอริธึมซ้ำได้ทั้งสองแบบซึ่งเข้าชมข้อมูลที่กำหนดไว้หลายครั้งในลูปและการวิเคราะห์ข้อมูลเชิงโต้ตอบ / สำรวจ ได้แก่ การสืบค้นข้อมูลในฐานข้อมูลแบบซ้ำ ๆ ความล่าช้าในการประยุกต์ใช้งานดังกล่าวอาจลดลงเมื่อเทียบกับการใช้งาน MapReduce ตามลำดับ (เช่นเดียวกับในกลุ่ม Apache Hadoop) ในขั้นตอนของการฝึกอบรมขั้นตอนอัลกอริธึมสำหรับระบบการเรียนรู้เครื่องซึ่งเป็นแรงผลักดันเริ่มต้นสำหรับการพัฒนา Apache Spark

Apache Spark ต้องมีผู้จัดการกลุ่มและระบบจัดเก็บแบบกระจาย สำหรับการจัดการกลุ่ม Spark สนับสนุนแบบสแตนด์อโลน (native Spark cluster), Hadoop YARN หรือ Apache Mesos สำหรับการจัดเก็บกระจาย Spark สามารถติดต่อกับหลากหลายรวมทั้ง Hadoop Distributed File System (HDFS) ระบบ MapR ไฟล์ (MapR-FS) คาสซานดรา OpenStack Swift, Amazon S3, Kudu หรือ สามารถใช้โซลูชันแบบกำหนดเองได้ Spark ยังรองรับโหมด local-pseudo-distribute ซึ่งมักใช้เฉพาะสำหรับวัตถุประสงค์ในการพัฒนาหรือทดสอบเท่านั้นโดยไม่จำเป็นต้องจัดเก็บแบบกระจายและสามารถใช้ระบบไฟล์ในระบบได้ ในสถานการณ์เช่นนี้ Spark จะทำงานบนเครื่องเดียวที่มีตัวประมวลผลหนึ่งตัวต่อแกน CPU

ภาพที่ 2.6 สัญลักษณ์ของ Apache Spark

2.1.6 Kibana

1) ความหมายของ โปรแกรม

เครื่องมือ Visualize สำหรับแสดงผลข้อมล ในรูปแบบต่างๆ เช่น กราฟแบบต่างๆ ตาราง แผนที่ และสามารถสร้างการแสดงผลข้อมูล หรือ Dashboard

Kibana[6] เป็นเครื่องมือที่ถูกใช้มาก สำหรับการแสดงผลข้อมูลจาก Elasticsearch ให้อยู่ในรูปแบบต่างๆ ตามที่ต้องการ

2) ประวัติ Kibana

Kibana นั้นเป็นเครื่องมือพัฒนามาเพื่อแสดงผลข้อมูลในรูปแบบกราฟ เป็นข้อมูล log ที่ส่งมาจาก Logstash และทำการจัดเก็บไว้ใน Elasticsearch

ภาพที่ 2.7 หน้าตาของ Kibana

2.1.7 ภาษา Scala

ภาษา Scala[7] อ่านว่า "สกาล่า" เปิดตัวสู่สาธารณะตั้งแต่ปี 2004 ซึ่ง ณ ปัจจุบันก็ผ่านมากว่า 12 ปีแล้ว โดยลักษณะของภาษารองรับการเขียนโปรแกรมทั้งแบบ functional และ imperative ทั้งสองแบบ ตัวภาษา Scala เป็นภาษาที่ทำงานอยู่บน JVM ทำให้สามารถเรียกใช้งานไลบรารีทั้งหมดที่สามารถทำงานบน JVM ได้ทันที

แต่ถึงอย่างนั้นนักพัฒนาก็เลือกที่จะเขียนไลบรารีขึ้นมาใหม่เพราะความต้องการทางด้านประสิทธิภาพมากกว่า หลักการเขียนของ Scala จะอิงกับหลักการของ OOP ในภาษา Java แต่จะมี syntax ที่สั้นมากทำให้บางครั้งอาจจะอ่านยากเกินไปรวมไปถึงภาษา Scala เองยินยอมให้ใช้อักขระพิเศษในการตั้งชื่อได้อีกด้วย (เหมือน JavaScript) ทำให้การ debug ในบางครั้งเป็นเรื่องน่าปวดหัวอยู่พอสมควร แต่ตรงนี้มองว่าแก้ปัญหาได้ด้วยการกำหนด coding standard ในการพัฒนาโปรแกรมแทน

อีกหนึ่งข้อที่สำคัญคือภาษา Scala มีความเป็น strong static typing ค่อนข้างมากทำให้ลดโอกาสเกิดข้อผิดพลาดในการพัฒนาโปรแกรมได้เป็นอย่างดี ในสมัยก่อนภาษาโปรแกรมเองก็มาพร้อมกับความสามารถ static typing อยู่แล้วเนื่องจากติดปัญหาด้านทรัพยากร

ในเวลาต่อมาทรัพยากรที่มากขึ้นของคอมพิวเตอร์ส่งผลให้เกิดการสร้างภาษาโปรแกรมที่มี dynamic typing (หรือเรียกว่า duck typing) ทำให้การพัฒนาโปรแกรมเป็นเรื่องง่ายเพราะตัวภาษาจะทำหน้าที่คอยทำ type casting ให้เองโดยอัตโนมัติแต่นั้นก็สร้างปัญหาให้การพัฒนาระบบที่มีขนาดใหญ่หรือนักพัฒนาที่ขาดความชำนาญทำให้เกิดข้อผิดพลาดในโปรแกรมอยู่บ่อยครั้ง

จนตอนนี้โลกของการพัฒนาซอฟต์แวร์เปลี่ยนไปสู่ยุคของการใช้งานคลาวด์และเน้น release บ่อยขึ้น ทำให้การพัฒนาโปรแกรมต้องการความรัดกุมในการทำงานมากขึ้นและใช้งานทรัพยากรน้อยลง (เพราะคลาวด์คิดค่าใช้จ่ายตามปริมาณทรัพยากรที่ถูกใช้ในการประมวลผล) ทำให้ภาษาโปรแกรมที่มีลักษณะเป็น static typing เริ่มกลับมาได้รับความนิยมอีกครั้งหนึ่ง

ภาพที่ 2.8 สัญลักษณ์ ภาษา Scala

```
object Main {
  implicit class RichC(val v: C) {
    def name = "foo"
  }

  def main(args: Array[String]) {
    val v = new C
    println(v.name)
  }
}
```

ภาพที่ 2.9 ตัวอย่าง การเขียนโปรแกรม ภาษา Scala

บทที่ 3
แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนงานการปฏิบัติงาน

ตารางที่ 3.1 ตารางการปฏิบัติงาน

หัวข้องาน	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3				เดือนที่ 4			
ศึกษา Apache Kafka																
ศึกษา Apache Spark																
ศึกษา Apache Elasticsearch และ Kibana																
พัฒนา ระบบ Apache Kafka และข้อมูลจำลอง																
พัฒนาระบบ Spark																
พัฒนาระบบ Apache Elasticsearch และ Kibana																
ทดสอบระบบ																

3.2 รายละเอียดโครงงานที่ได้รับมอบหมาย

3.2.1 ประชากรและกลุ่มตัวอย่าง

ประชากร

ประชากรที่ใช้ในการวิจัยในครั้งนี้ ได้จากการสุ่มของข้อมูล โดย ผลลัพธ์ คือ ผู้ใช้บริการรถเมล์ สาย 11 เล็ก มีผู้ใช้บริการ 400-600 คนต่อเที่ยว มีจำนวนเที่ยวรถเมล์ 2 เที่ยวต่อวัน เฉพาะ ขาไป และ ขากลับ จำนวนที่ใช้ รถเมล์ 1 คัน จะได้จำนวนผู้ใช้บริการเฉลี่ย 500 คนต่อเที่ยว แสดงว่ามีจำนวนประชากรที่ใช้ในการศึกษาเฉลี่ย 1000 คนต่อวัน โดยส่วนประกอบของข้อมูล ถูกแบ่งออกเป็น 4 คือ

1) Bus_ID คือ หมายเลข หรือ สาย ของรถเมล์

2) DateTime คือ เวลาที่บุคคลขึ้นหรือลงรถเมล์

3) Station คือ สถาณีที่รถเมลล์จอด

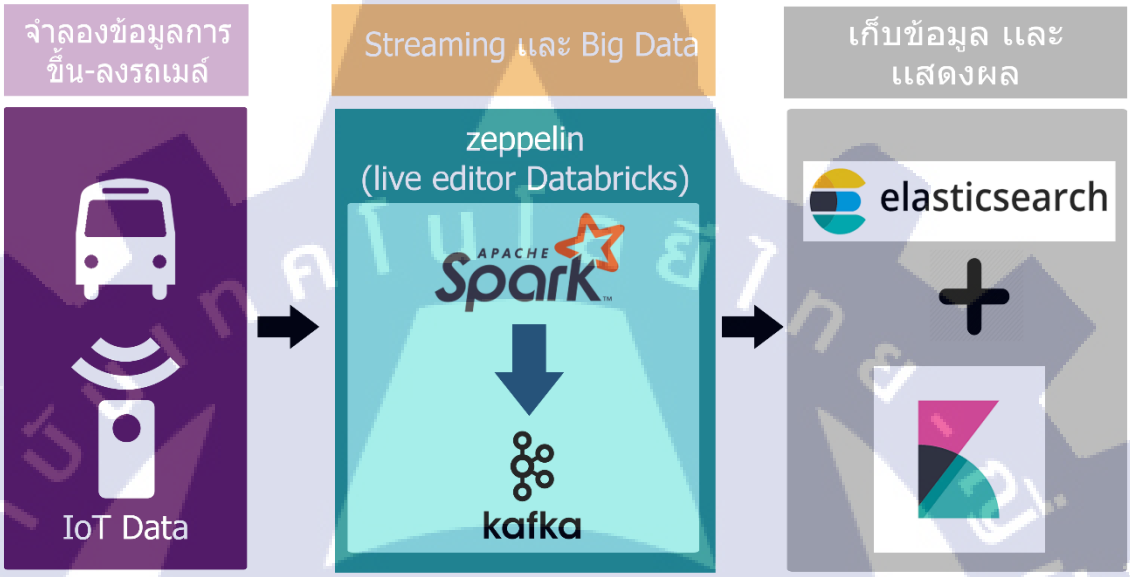
4) Staus คือ สถาณะบุคคล ขึ้น หรือ ลงรถเมล์

กลุ่มตัวอย่าง

การกำหนดขนาดของกลุ่มตัวอย่าง เพื่อให้ขนาดกลุ่มตัวอย่างที่ทำการศึกษา เป็น ตัวแทนที่สามารถให้ข้อมูลเกี่ยวกับประชากร จึงทำการการหนดขนาดตัวอย่าง โดย

- ประชากรจำนวน 500 คน
- ช่วงเวลาที่เก็บข้อมูล คือ ระยะเวลาตั้งแต่ ต้นทาง ขอรถเมล์สาย 11 ถึง ปลายทาง รถเมล์ สาย 11 รอบขาไป

3.2.2 อธิบายการทำงานของ Framework



รูปที่ 3.1 Framework การทำงานของ project

จากรูปที่ 3.1 เป็น Framework การทำงานของโครงงานโดยแบ่งขั้นตอนได้ 3 ขั้นตอนคือ

1) จำลองการเก็บข้อมูลผู้ใช้บริการ

ส่วนการพัฒนาระบบ Apache Kafka นั้น ในส่วนนี้จะเป็นการสร้างจำลองข้อมูล จาก IoT รถเมล์ โดยจะจำลองข้อมูลโดยการ Random จำนวนคนเข้าออก โดยในส่วนของโค้ดจะอธิบาย ในขั้นตอนถัดไป

2) Streaming และ Bigdata

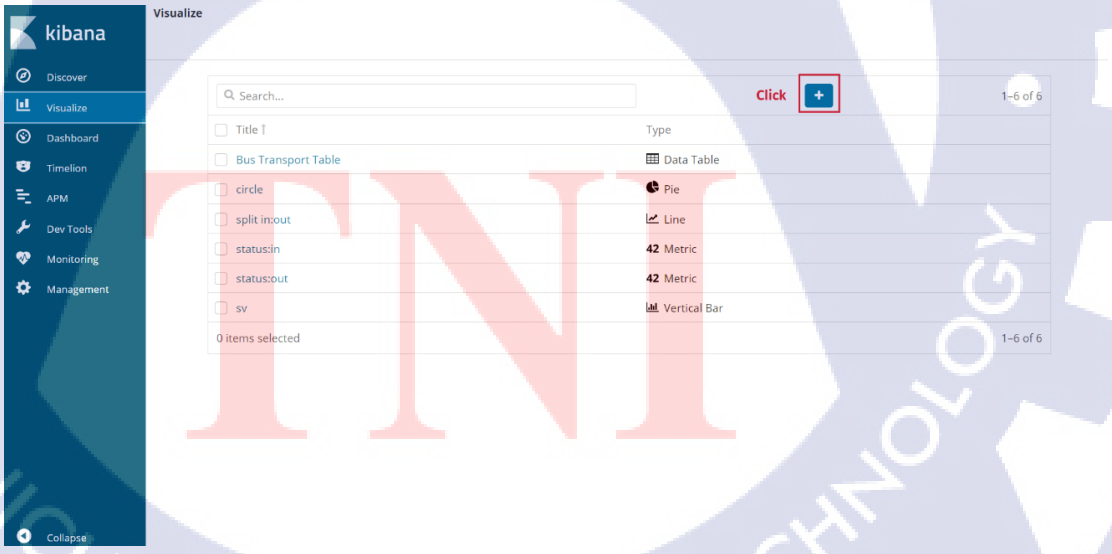
ในส่วนนี้จะเป็นขั้นตอนการเตรียมตัว ก่อนส่งข้อมูลไปสู่ Spark โดยเมื่อรับข้อมูลจากต้นทางแล้ว จะส่งข้อมูลไปยัง Spark ทันทีเปรียบเสมือนตัวกลาง

หลังจากนั้นข้อมูลที่ได้จากการจำลองจะถูกส่งไปให้ทางฝั่ง Apache spark โดยตัวที่ส่งทำการส่งไปคือ Apache Kafka ซึ่งส่วนนี้จะถูกส่งแบบเรียลไทม์

4) เก็บข้อมูลและ แสดงผล

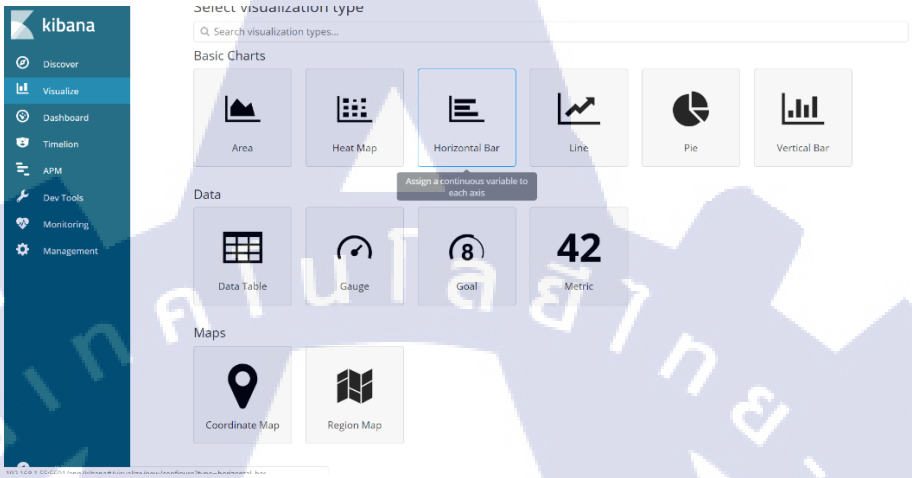
โดยในส่วนนี้ หลังจากที่ได้รับข้อมูล จาก spark แล้วข้อมูลจะถูกเก็บลง Database แบบ เรียลไทม์ และสามารถ เรียกดูข้อมูล ได้ผ่านทาง Kibana UI ของ Elasticsearch โดย จะอธิบาย ขั้นตอนในการสร้างกราฟ

1) เข้ามาที่หน้า Visualize และทำการคลิก เครื่องหมายบวกเพื่อสร้างกราฟรูปแบบใหม่ ดังรูปที่ 3.2



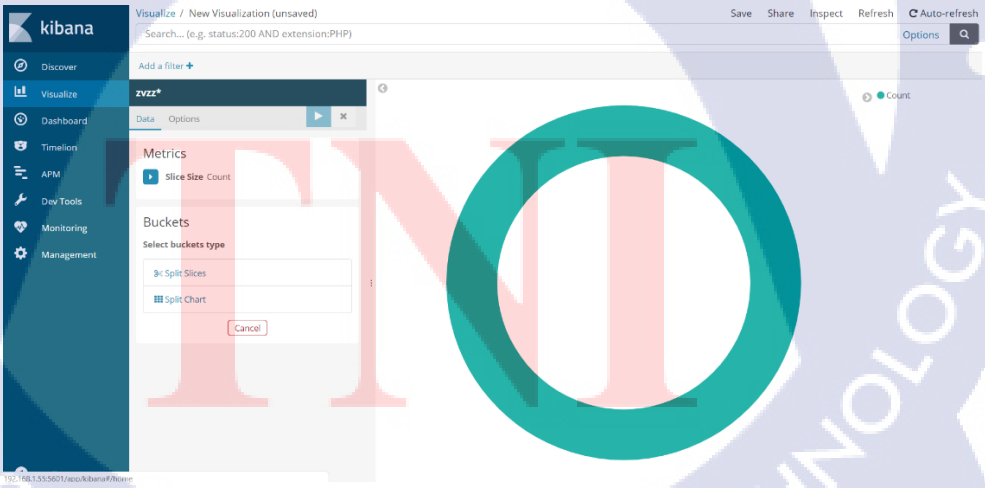
รูปที่ 3.2 อธิบายขั้นตอนสร้างกราฟ 1

2) ในส่วนนี้จะมีให้เลือกรูปแบบการสร้างกราฟมากมายโดยตัวอย่างจะทำการสร้างกราฟรูปแบบ pie ดังรูปที่ 3.3



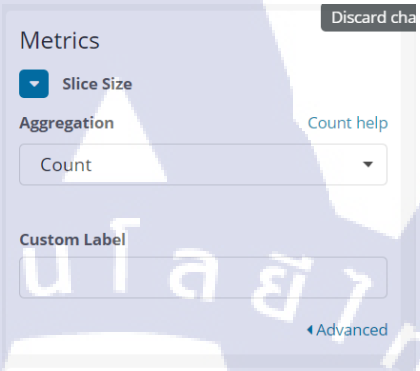
รูปที่ 3.3 อธิบายขั้นตอนสร้างกราฟ 2

3) เมื่อเข้ามาแล้วจะปรากฏ แผนภูมิกราฟที่ข้อมูลว่างเปล่า ดังรูปที่ 3.4



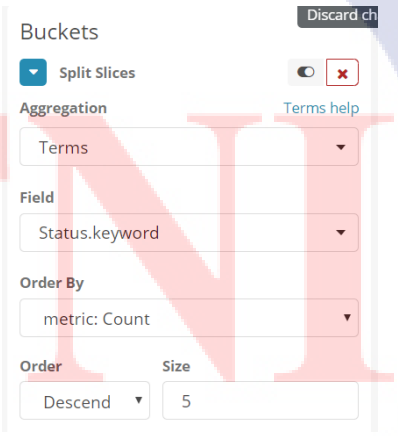
รูปที่ 3.4 อธิบายขั้นตอนสร้างกราฟ 3

4) คลิกลูกศรลง ตรง Slice Size เพื่อทำการขยายข้อมูลลง ช่อง Aggregation ให้เลือก count ดังรูปที่ 3.5



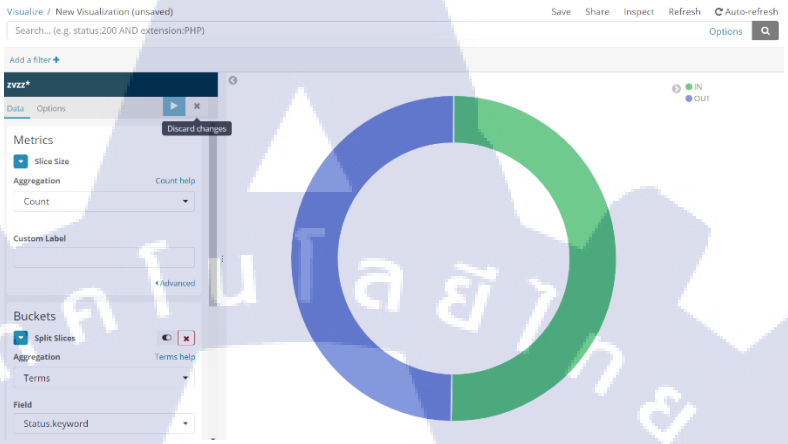
รูปที่ 3.5 อธิบายขั้นตอนสร้างกราฟ 4

5) คลิกลูกศรลง ตรง Split Slices เพื่อทำการขยายข้อมูลลง ช่อง Aggregation ให้เลือก terms Fields ให้เลือก Status ส่วน Order by ให้เลือก Metric: Count ดังรูป 3.6



รูปที่ 3.6 อธิบายขั้นตอนสร้างกราฟ 5

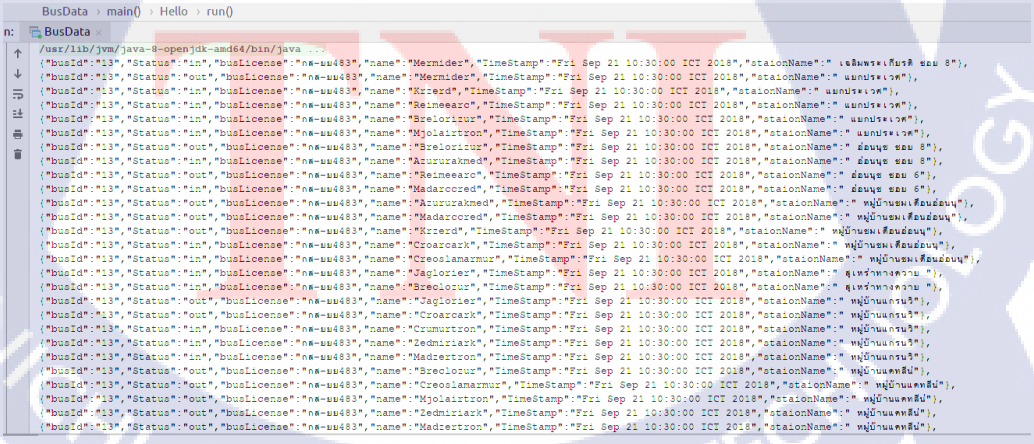
6) ซึ่งผลลัพธ์ จะได้ดังรูป 3.7 โดย แผนภูมิดังกล่าวจะเป็นแผนภูมิที่อธิบาย เปรียบเทียบ จำนวน ขาขึ้น กับ ขาลงรถเมล์



รูปที่ 3.7 อธิบายขั้นตอนสร้างกราฟ 6

3.2.3 วิธีการเก็บข้อมูล

ในส่วนของการเก็บข้อมูลนั้น ได้ทำการ จำลองข้อมูลโดยการ สุ่ม จำนวนคนเข้าออก และข้อมูลจะถูกเก็บเป็น Json ดังรูปที่ 3.8



รูปที่ 3.8 ข้อมูล IoT รถเมล์ แสดงเป็น Json

โดยในส่วน การทำงานของการจำลองข้อมูลนั้น จะอธิบายข้างล่างดังในรูปที่ 3.9 - 3.11

```
var now_date = Calendar.getInstance
now_date.set(2018, 9, 21, 5, 40, 0)
var MaxPerson: Int = 50
val dateFormatter = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss")
var countStation: Int = 1
var block: Int = 0
var outBound: Int = 0
var inBound: Int = 0
var Maxblock: Int = 0
var person = 0;
```

รูปที่ 3.9 อธิบาย โค้ด การเก็บข้อมูล 1

จากรูปที่ 3.9 ส่วนนี้จะเป็นการ ตั้งรูปแบบของเวลา และ ตั้ง ค่าเริ่มต้น ของตัวแปร ต่าง โดยจะอธิบายเป็นหมายเลข ดังรูป

1) จะเป็นการตั้งตาราง เวลา และวันที่ เริ่มต้น ของรถเมล์
2) MaxPerson คือตัวแปรที่เก็บ จำนวนบุคคลในรถเมล์ ซึ่งในรถคันนี้จะมีบุคคลได้ไม่เกิน 50 คน
3) เป็นการเซ็ตรูปแบบ เวลาข้อมูลที่ถูกส่ง โดย รูปแบบคือ MM/dd/yyyy HH:mm:ss

```
for( a <- 0 to staion.lenght) {
  //now_date = format(now_date)

  now_date.setTimeInMillis(now_date.getTimeInMillis)
  now_date.add(Calendar.MINUTE, 5)
  if (a == 0) {
    Maxblock = MaxPerson
  }
  if (a == 50) {
    for (i <- 0 until Maxblock) {}
    // System.out.println("out");
    // System.out.println("out");
    Maxblock = MaxPerson
  }

  if (now_date.getTime.getHours >= 7 && now_date.getTime.getHours <= 9) {
    try
      person = rand.nextInt(Maxblock % 10) + 20
    catch {
      case e: Exception => {}

    }
  } else if (now_date.getTime.getHours >= 17 && now_date.getTime.getHours <= 19) {
    try person = rand.nextInt(Maxblock % 10) + 20
    catch {
      case e: Exception => println(e)

    }
  } else {
    // System.out.println(Maxblock);
    person = rand.nextInt(Maxblock) % 5
  }
  inBound = person % (Maxblock)
  block += inBound
```

รูปที่ 3.10 อธิบาย โค้ด การเก็บข้อมูล 2

จากรูปที่ 3.10 ส่วนแรกจะมีการเช็คข้อมูลเวลา เมื่อถึง 7โมงเช้า จนกระทั้ง ถึง 9 โมงเช้า จะ สุ่มข้อมูลจำนวนการ ขึ้น ลง รถเมล์ โดยณ เวลานี้อัตรา การสุ่ม จะสูงขึ้น เพื่อจำลองสถานการณ์ เร่งด่วน ในช่วงเช้าที่มี คนจำนวนมากใช้รถเมล์ เป็นการกระจายข้อมูล

ส่วนที่สอง จะมีการเช็คข้อมูลเวลา เมื่อถึง 5โมงเย็น จนกระทั้ง ถึง 1 ทุ่ม ซึ่งส่วนนี้จะเหมือนส่วนแรก เพื่อจำลองสถาณการณ์เร่ง ของขากลับ

ส่วนที่สาม จะเป็นเวลาส่วนที่ไม่ใช่เวลา 7โมงเช้า ถึง 9 และ เวลา 5 โมงเย็น ถึง 1 ทุ่ม โดยส่วนนี้ อัตราการสุ่ม จะน้อย เพราะไม่ใช่เวลาเร่งรีบที่มีจำนวนคนใช้มาก

```
if (a != 0) {
  outBound = rand.nextInt(block)
  block -= outBound
  for (i <- 0 until outBound) {
      val bu  = "{\"Bus_ID\":\"" + lineNum + "\"," +
      "\"DateTime\":\"" + now_date.getTimeInMillis + "\"," +
      "\"Station\":\"" + staion(a) + "\"," +
      "\"Status\":\"" + "OUT" + "\"}"
    try {//sentJson(lineNum, dateFormatter.format(now_date.getTime), staion(a), "OUT")
        val record = new ProducerRecord(topic, s"$a", s"$bu")
      producer.send(record)
      print(bu)
    //println(bu)
    Thread.sleep(1000)}
    catch {
      case e: Exception => {}
    }
  }
}
for (i <- 0 until inBound) {
  try {//sentJson(lineNum, dateFormatter.format(now_date.getTime), staion(a), "OUT")
   val buin = "{\"Bus_ID\":\"" + lineNum + "\"," +
      "\"DateTime\":\"" + now_date.getTimeInMillis + "\"," +
      "\"Station\":\"" + staion(a) + "\"," +
      "\"Status\":\"" + "IN" + "\"}"
   val record2 = new ProducerRecord(topic, s"$a", s"$buin")
        producer.send(record2)

println(buin)
    Thread.sleep(1000)

  }
```

รูปที่ 3.11 อธิบาย โค้ด การเก็บข้อมูล 3

จากรูปที่ 3.11 โดยในส่วนนี้จะแยกเป็นสองส่วน คือ IN และ OUT ในส่วนแรก จะเป็น OUT ส่วน นี้จะเป็น ขาลง โดย เมื่อได้ข้อมูลจากการสุ่มข้อมูล แล้วนั้น ข้อมูลจะถูกเก็บในรูปแบบ String จากนั้นจะถูกแปลงข้อมูลเป็นรูปแบบข้อมูล Json และ จะถูกส่งข้อมูลไป Spark เพื่อทำการ query ข้อมูลต่อไป

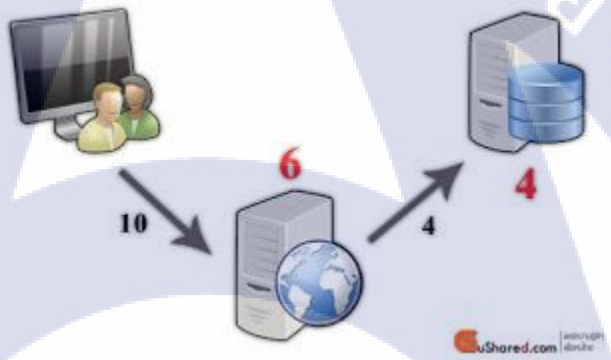
ในส่วนที่สองนั้น จะเป็น IN โดยส่วนนี้ จะเป็นข้อมูลขาขึ้น และวิธีการเก็บข้อมูล และวิธีกการส่ง จะเหมือนกับ ข้อมูลส่วนแรก

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงงาน

3.3.1 งานประจำที่ได้รับมอบหมาย

1) Stored Procedure

Stored Procedures คือการเขียน SQL ฝังไว้ที่ดาต้าเบสเพื่อ ช่วยแบ่งเบาภาระของเว็บเซิฟเวอร์ ซึ่งสามารถทำได้ทั้ง SQL Server และ Oracle ประโยชน์หลักๆที่ จะได้รับ เช่น เมื่อผู้ใช้งานเว็บไซต์ ทำอะไรขึ้นมาสักอย่าง ก็จะเกิดงานส่งไปที่เว็บเซิฟเวอร์ 10 งาน แต่ใน 10 งานนี้เป็นงานที่ดาต้าเบสสามารถทำได้ 4 งาน ก็จะให้ฝั่งดาต้าเบสมาช่วยทำ เท่ากับว่า สามารถแบ่งเบางานของฝั่งเว็บเซิฟเวอร์ได้ 4 งาน



รูปที่ 3.12 การทำงานของ Stored Procedure

การเขียน Stored Procedure จะจำเป็นมากในการพัฒนาระบบขนาดใหญ่ ที่มีการจัดเก็บข้อมูลและประมวลผลจำนวนมาก ๆ เพราะการเขียน Query ในปกติ ที่มีการติดต่อกับข้อมูลหลายครั้ง จะต้อง Select แล้วใช้โปรแกรมอ่านค่า เมื่ออ่านได้ค่าแล้วค่อยส่งไปประมวลผลที่ Database ซ้ำ ๆ ซึ่งจะเป็นการทำงานซ้ำซ้อน มีการรับส่งระหว่าง Application กับ Database เป็นสิบหรือหลายร้อยครั้ง ซึ่งผลที่ตามมาคือ Performance ของโปรแกรมจะทำงานช้ามาก ทางเลือกในการแก้ปัญหานี้ก็คือ ใช้การทำงานซ้ำซ้อนทั้งหมดนี้ที่ Database แทน โดยเพียงส่งค่า Parameters ที่จำเป็นต้องใช้ จากนั้นบน Database ก็จะนำค่า Parameters ที่ส่งไปนั้น ทำงานตามคำสั่งต่าง ๆ บน Stored Procedure ที่เขียน

ขึ้น เมื่อได้ค่าที่ต้องการค่อยส่งค่า Result กลับมายังโปรแกรม วิธีนี้จะเป็นการเพิ่มประสิทธิภาพการทำงานของโปรแกรมให้ทำงานเร็วขึ้น และลด Traffic ระหว่าง Database กับ Application ได้สูงมาก

ข้อดีการใช้ Stored Procedure

1) Syntax เขียนง่าย เข้าใจง่าย และในปัจจุบันมี Tools ที่ช่วยให้การ Debug ง่ายมาก

2) เพิ่มประสิทธิภาพการทำงานการ Query Database ได้อย่างดีเยี่ยม ลดภาระการทำงานของ Application

3) ลด Traffic ของ Network หรือระหว่าง Database กับ Application

4) การพัฒนาค่อนข้างจะเป็นระบบ แยกระหว่าง Application Logic กับ Database Logic ได้ชัดเจน เช่น ถ้าต้องการแก้ไข Logic ของ Database อาจจะเพียงแก้ไขที่ Stored Procedure เท่านั้น

ข้อเสียการใช้ Stored Procedure

1) การเขียน Stored จะผูกกับ Database นั้น ๆ เมื่อเปลี่ยน Database ไปใช้ตัวอื่น จะต้องเขียน Stored ใหม่ทั้งหมด

2) Syntax ของการเขียน Stored Procedure จะไม่เหมือนกัน

3) เมื่อนำไปใช้บน Production Server การ Debug ตรวจสอบหาข้อผิดพลาดทำได้ยากพอสมควร

ใจความสำคัญของการเขียน Stored ให้สามารถนำไปใช้งานได้จริง โดยได้ยกตัวอย่างการสร้าง Table ประกอบขึ้นมา 2 ตารางคือ CUSTOMER ดังผลลัพธ์จะได้ดังรูป 3.13 และ โครงสร้างดังรูป 3.14

CUSTOMER_ID	NAME	EMAIL	COUNTRY_CODE	BUDGET	USED
C001	Win Weerachai	win.weerachai@thaicreate.com	TH	1000000.00	600000.00
C002	John Smith	john.smith@thaicreate.com	UK	2000000.00	800000.00
C003	Jame Born	jame.born@thaicreate.com	US	3000000.00	600000.00
C004	Chalee Angel	chalee.angel@thaicreate.com	US	4000000.00	100000.00

รูปที่ 3.13 ผลลัพธ์ตาราง CUSTOMER

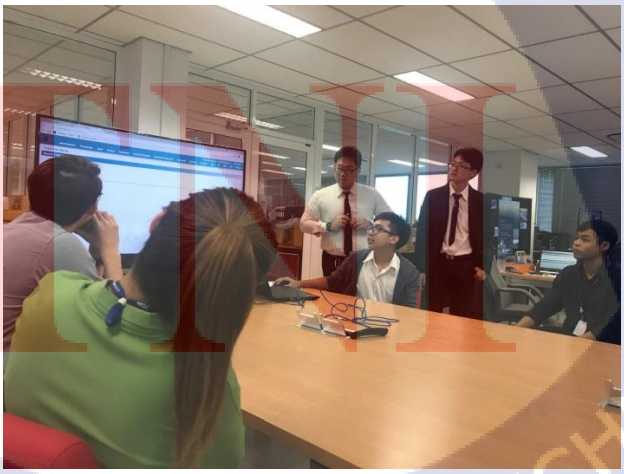
```
CREATE TABLE [CUSTOMER](
      [CUSTOMER_ID] [varchar](4) NOT NULL,
      [NAME] [varchar](50) NULL,
      [EMAIL] [varchar](50) NULL,
      [COUNTRY_CODE] [varchar](2) NULL,
      [BUDGET] [decimal](18, 2) NULL,
      [USED] [decimal](18, 2) NULL,
  CONSTRAINT [PK2_CUSTOMER] PRIMARY KEY CLUSTERED
(
      [CUSTOMER_ID] ASC
) ) ON [PRIMARY]

INSERT INTO CUSTOMER VALUES ('C001', 'Win Weerachai', 'win.weerachai@thaicreate.com', 'TH', 1000000,
 600000);
INSERT INTO CUSTOMER VALUES ('C002', 'John  Smith', 'john.smith@thaicreate.com', 'UK', 2000000, 800000);
INSERT INTO CUSTOMER VALUES ('C003', 'Jame Born', 'jame.born@thaicreate.com', 'US', 3000000, 600000);
INSERT INTO CUSTOMER VALUES ('C004', 'Chalee Angel', 'chalee.angel@thaicreate.com', 'US', 4000000,
 100000);
```

รูปที่ 3.14 โครงสร้างตาราง CUSTOMER

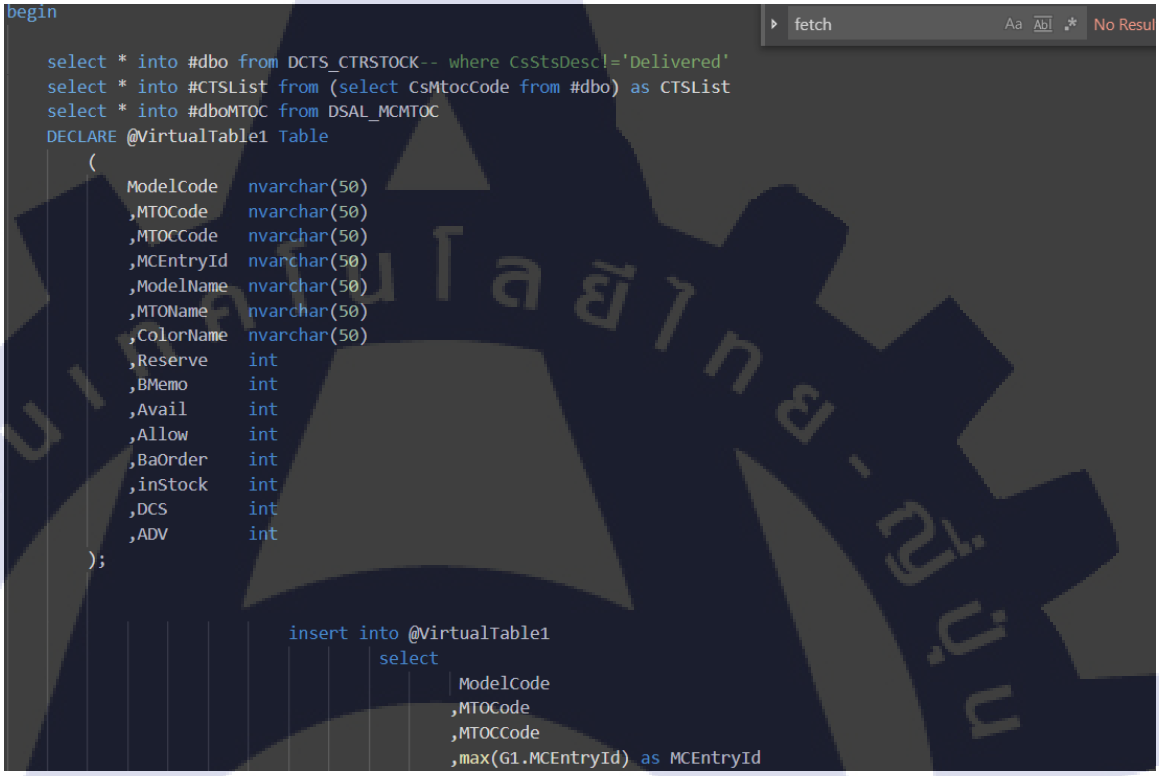
สาเหตุที่มีการนำ Stored Procedure มาใช้กับโปรเจค เพราะ โปรแกรมมีการประมวลผลในการโชว์ข้อมูล ค่อนข้างช้า โดยใช้เวลาเฉลี่ย ประมาณ 10 นาที หากมี Process ที่ใหญ่ จะใช้ระยะเวลา มาก 15 นาที เป็นอย่างต่ำ หากนำ Stored Procedure มาใช้จะช่วยลดระยะเวลาได้ มาก ถึงจากเดิม 10 นาที เหลือ ไม่ถึงนาที

เนื่องจาก Stored Procedure นำมาปรับใช้เป็นครั้งแรกจึงต้องเสนอเข้าประชุม เพื่อปรับเปลี่ยนรูปแบบจากเดิมที่ ดึงข้อมูลโดยใช้ ภาษา Lampda ใช้การประมวลผลผ่าน C# เปลี่ยนมาเป็นรูปแบบใหม่ คือ Stored Procedure โดยผู้จัดทำรายงานได้มีโอกาสเป็นผู้ทำหน้าที่ ดังรูป 3.15



รูปที่ 3.15 เสนอ Stored Procedure

ในส่วนของงานที่ผู้จัดทำรายงาน นั้นได้รับมา โดยส่วนใหญ่จะเป็น การแปลง ภาษา Lampda บน C# ให้มาอยู่ในรูปแบบ Stored Procedure ตัวอย่างดังรูป 3.16



รูปที่ 3.16 ตัวอย่าง Stored Procedure

2) Web Chat Application with CRM(Customer Relationship Management)

ผู้จัดทำรายงานได้รับมอบหมายให้ ทำระบบ Web Chat Application ให้กับบริษัทหนึ่ง ส่วนทางผู้จัดทำรายงานได้รับหน้าที่ในส่วนของ Front End โดยใช้ภาษา React Js ใน การพัฒนาแอพพลิเคชั่น

2.1 CRM (Customer Relationship Management)

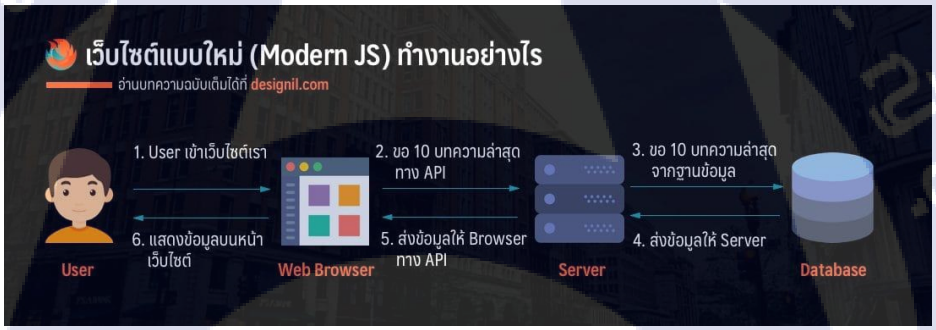
CRM คือ การบริหารลูกค้าสัมพันธ์ หรือ การบริหารความสัมพันธ์กับลูกค้า ซึ่งหมายถึงวิธีการที่จะบริหารให้ลูกค้ามีความรู้สึกผูกพันธ์กับสินค้า ,บริการ หรือองค์กร เมื่อลูกค้าเค้ามีความผูกพันธ์ในทางที่ดี แล้วก็ลูกค้านั้นไม่คิดที่จะเปลี่ยนใจไปจากสินค้าหรือบริการทำให้มีฐานลูกค้าที่

มั้นคง และนำมาซึ่งความมั่นคงของบริษัท ดังนั้น การที่จะรู้ซึ้งถึงสถานะความผูกพันธ์กับลูกค้าได้นั้น ก็ต้องอาศัยการสังเกตุพฤติกรรมของลูกค้า แล้วนำมาวิเคราะห์หาความเกี่ยวข้องระหว่าง พฤติกรรมของลูกค้ากับกลยุทธ์ทางการตลาด

2.2 React JS

ไอเดียนี้เมื่อก่อน เรียกว่า AJAX ก็คือ ใช้ JavaScript ส่ง Request ไปขอข้อมูลจาก Server แล้ว Server ตอบกลับมาเป็นข้อมูลอย่างเดียว ค่อยใช้ JavaScript เอาข้อมูลมาแสดงอีกที อย่างไรก็ตาม พอข้อมูลถูกส่งจาก Server มา Web Browser ก็เกิดปัญหา จะจัดการข้อมูลบน Web Browser อย่างไร ซึ่งเมื่อก่อนไม่มี Library คอยช่วยเหลือด้านการแสดงผล ทำให้โค้ดพันกันมั่วไปหมด

JavaScript Framework สมัยใหม่ เช่น React ก็เลยออกมาเพื่อช่วยให้จัดการเรื่องการดึงข้อมูล การแสดงผลข้อมูลในเว็บให้ง่ายขึ้น ดังรูป 3.15



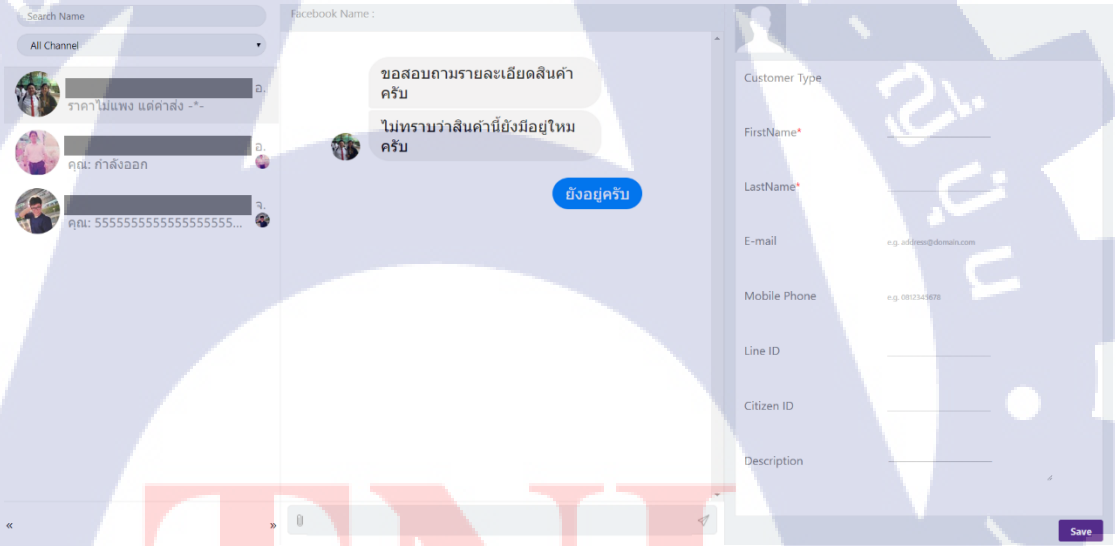
รูปที่ 3.17 Flow การทำงานเว็บไซต์สมัยใหม่

1) User เข้าเว็บไซต์
2) โหลด HTML, CSS, JS ขึ้นมาก่อน แล้ว JavaScript ค่อยขอบทความล่าสุดจาก Server
3) Server ได้รับคำขอ ก็ส่งไปขอข้อมูลจากฐานข้อมูลอีกที
4) Database ส่งข้อมูลกลับมาให้ Server
5) Server ส่งข้อมูลกลับไปให้ Browser ทาง API โดยส่วนใหญ่จะส่งเป็น JSON เพราะ JavaScript สามารถนำไปใช้งานต่อได้ง่าย
6) JavaScript นำข้อมูลที่ได้รับมาจัดการแสดงผลบนหน้าเว็บไซต์

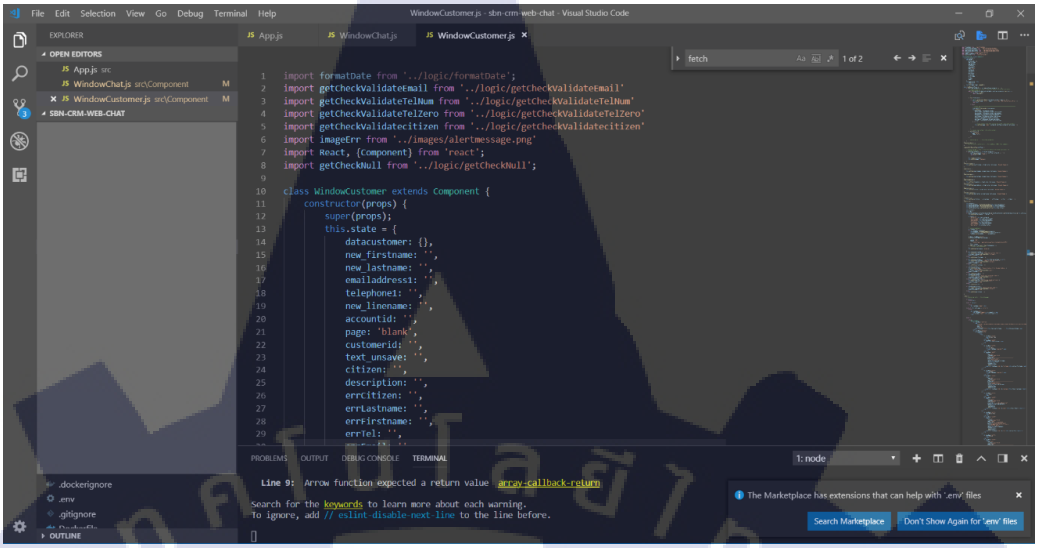
โดยโปรเจค Web Chat Application with CRM ที่ผู้จัดทำรายงานได้รับมอบหมายนั้น มีหน้าที่ในการ ดึงข้อมูล ผ่านทาง CRM ของทาง Microsoft โดยจะดึง อาทิ ชื่อ นามสกุล ข้อมูลแชท อีเมล เบอร์โทรศัพท์ อื่นๆ และ หน้าที่สำคัญหลักๆ คือ สามารถ โต้ตอบกับลูกค้าได้ โดยไม่ต้องผ่าน เฟสบุ๊ค โดยจะแบ่งเป็น 3 หน้า ดังรูป 3.15 คือ

1) หน้ารายชื่อข้อมูลที่มีการติดต่อเข้ามา
2) หน้าแชท
3) หน้าข้อมูลลูกค้า

และตัวอย่าง Source Code ดังรูป 3.16



รูปที่ 3.18 ตัวอย่างหน้าตาโปรแกรม(ตัวอย่างนี้ไม่ใช่หน้าตาจริงของโปรแกรม)



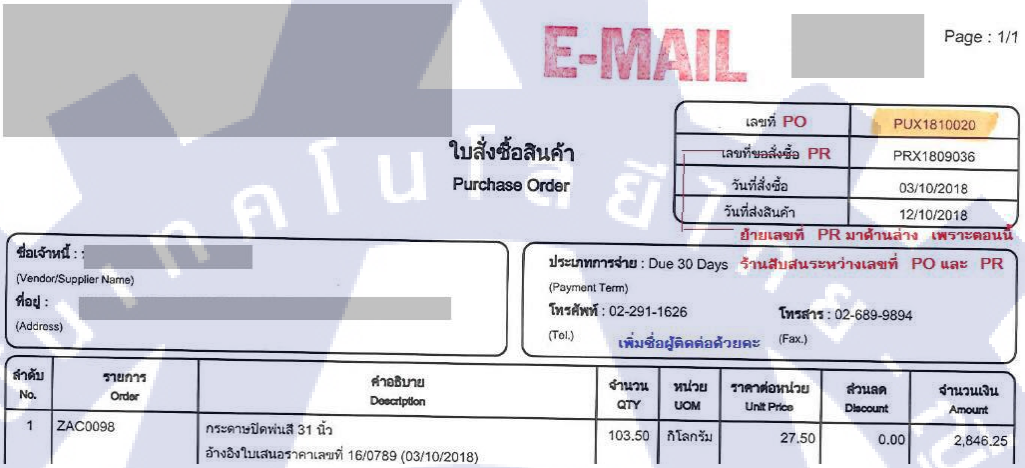
รูปที่ 3.19 ตัวอย่าง Soure Code โปรแกรม

4) แก้ Report (Crystal Report **)**

Crystal Report คือเครื่องมือที่ใช้ในการออกรายงาน ซึ่งสามารถ ออกรายงานได้หลากหลายรูปแบบ ทั้งแบบ รายงานธรรมดา แบบ Cross Tab และแบบอื่นๆ ซึ่งมีเครื่องไม้เครื่องมือที่ออกแบบมาให้ง่ายต่อการใช้งาน และการติดต่อกับฐานข้อมูลก็สามารถทำได้หลากหลาย เช่น MS SQL Server,Access,Excel,XML,ADO.Net, ตลอดจนสามารถนำข้อมูลจาก Viewer ของเครื่องมาดูก็สามารถทำได้ ซึ่งให้ความสามารถที่หลากหลาย และการ View ก็สามารถ View ได้หลากหลาย เช่น การ View ผ่านตัวโปรแกรมเอง , การ View ผ่านโปรแกรมที่เป็น โปรแกรมประยุกต์ที่ Software House ต่างๆผลิตขึ้นมา หรือแม้กระทั้ง ดูบนเว็บ ซึ่งจากความสามารถที่หลากหลายดังกล่าวจึงเป็นที่นิยมใช้งานในเชิงพาณิชย์กัน ประกอบด้วย

1) พื้นที่ต่าง ๆ ในการทำงาน (Section)
2) ส่วนที่ใช้ในการแสดงผลข้อมูล (Fields)
3) การสร้างสูตรและการใช้งานสูตร (Function)
4) การจัดกลุ่ม (Group)
5) การแสดงกราฟ (Chart)
6) การสร้างรายงานย่อย (SubReport)

ทางผู้จัดทำรายงานได้รับมอบหมายให้แก้ไข report ดังตัวอย่างรูป 3.17 ซึ่งจะมีกาแก้ข้อมูลตั้งแต่ อาทิ เช่น แก้ไขชื่อ ย้ายตำแหน่งของข้อมูล และ เพิ่มข้อมูล เป็นต้น โดยตัวอย่างจะมีการเพิ่มข้อมูลในส่วน เลขที่ PO ย้าย PR มาด้านล่าง และแก้ไขหมายเลข ด้านบน



รูปที่ 3.20 ตัวอย่าง รายงานที่แก้

3.3.2 งานโครงงานที่ได้รับมอบหมาย

1) ศึกษา Apache Kafka

Apache Kafka เป็นระบบกระจายข้อความ(message, event, log, ฯลฯ) ที่ถูกออกแบบให้เป็นศูนย์กลางของการส่งข้อความในองค์กรขนาดใหญ่ ที่ถูกพัฒนาโดย LinkedIn และปัจจุบันเป็นซอฟท์แวร์เสรี (open source)

2) ศึกษา Apache Spark

Apache Spark คือเครื่องมือ สำหรับทำ data processing ที่สร้างบน Hadoop อีกทีโดยมีการทำงานเหมือนกันกับ Map Reduce โดยมีจุดเด่นอยู่ที่ความเร็วในการประมวลผล ซึ่งเร็วกว่า MapReduce ของ Hadoop ได้ถึง 10-100 เท่า และมี Spark SQL ที่จะช่วยให้สามารถเขียน query ได้อย่างง่าย

3) ศึกษา Apache Elasticsearch และ Kibana

1) Apache Elasticsearch

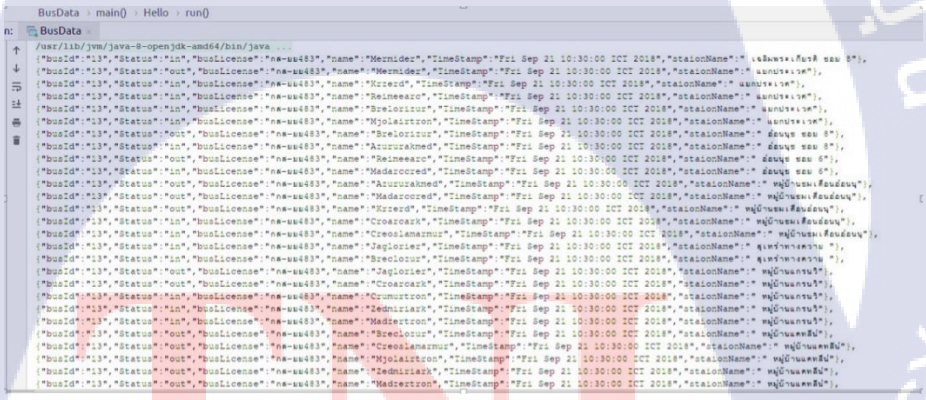
Elasticsearch คือ ที่เก็บข้อมูลที่พัฒนาต่อยอดมาจาก Apache Lucene มีจุดเด่นในเรื่องความสามารถของการค้นหา และสรุปข้อมูลขนาดใหญ่ได้อย่างรวดเร็ว

2) Kibana

Kibana นั้นเป็นเครื่องมือพัฒนามาเพื่อแสดงผลข้อมูลในรูปแบบกราฟ เป็นข้อมูล log ที่ส่งมาจาก Logstash และทำการจัดเก็บไว้ใน Elasticsearch

4) พัฒนา ระบบ Apache Kafka และข้อมูลจำลอง

ส่วนการพัฒนาระบบ Apache Kafka นั้น ในส่วนนี้จะเป็นการสร้างจำลองข้อมูล จาก IoT รถเมล์ โดยจะจำลองข้อมูลโดยการ random จำนวนคนเข้าออก โดยข้อมูลนี้จะเก็บเป็น Json ดังรูป 3.21 และข้อมูลก้อนนี้จะถูกส่งไปที่ ระบบ Spark



รูปที่ 3.21 ข้อมูล IoT รถเมล์ แสดงเป็น Json

5) พัฒนาระบบ Spark

หลังจากนั้นข้อมูลที่ได้จากการจำลองจะถูกส่งไปให้ทางฝั่ง Apache spark โดยตัวที่ส่งทำการส่งไปคือ Apache Kafka ซึ่งส่วนนี้จะถูกส่งแบบเรียลไทม์ ดังรูปที่ 3.22

```
{
  "paragraphs": [
    {
      "text": "%md\n\n# Simple Kafka Producer",
      "dateUpdated": "2018-09-28T01:56:36+0000",
      "config": {
        "enabled": true,
        "tableHide": false,
        "editorMode": "ace/mode/markdown",
        "results": {},
        "editorHide": true,
        "editorSetting": {
          "language": "markdown",
          "editOnDblClick": true
        },
        "colWidth": 12
      },
      "settings": {
        "params": {},
        "forms": {}
      },
      "results": {
        "code": "SUCCESS",
        "msg": [
          {
            "type": "HTML",
            "data": "<div class=\"markdown-body\">\n<h1>Simple
              Kafka Producer</h1>\n</div>"
          }
        ]
      },
      "apps": [],
      "jobName": "paragraph_1538099796425_1495845398",
      "id": "20171108-084210_1882269229"
```


รูปภาพ 3.22 โค้ด Kafka ที่ถูกแปลงแปลง Json แล้ว

6) พัฒนาระบบ Apache Elasticsearch และ Kibana

โดยในส่วนนี้ หลังจากที่ได้รับข้อมูล จาก spark แล้วข้อมูลจะถูกเก็บลง Database แบบเรียลไทม์ และ สามารถ เรียกดูข้อมูล ได้ผ่านทาง Kibana เป็น UI ของ Elasticsearch

7) ทดสอบระบบ

จากการทดสอบระบบ พบว่า ระบบที่แสดงข้อมูลเป็นกราฟนั้นยังไม่เรียลไทม์ เท่าที่ควรอาจจะมีความล่าช้าบ้าง ทั้งนี้ ขึ้นอยู่กับปัจจัย ต่างๆ อาทิเช่น Internet เป็นต้น

บทที่ 4
ผลการดำเนินงาน การวิเคราะห์และสรุปผลต่างๆ

4.1 ขั้นตอนและผลการดำเนินงาน

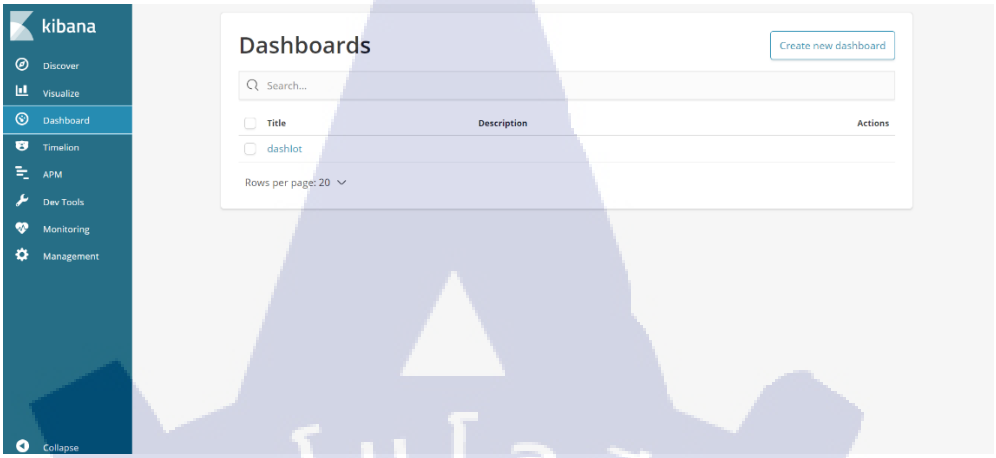
4.1.1 แสดงกราฟข้อมูล

1) เข้าไปที่ลิงค์ localhost:5601 พอเข้าไปแล้วจะพบกับหน้าแรกของ Kibana ดังรูปที่ 4.2

รูปที่ 4.1 หน้าแรกของ Kibana

2)คลิกเมนู Dashboard

3)เมื่อคลิกแล้วจะพบกับหน้า แรก Dashboard รวม จากนั้นเลือก Dashboard ที่ต้องการ ในทีนี้จะเลือก dashIot ดังรูป 4.2



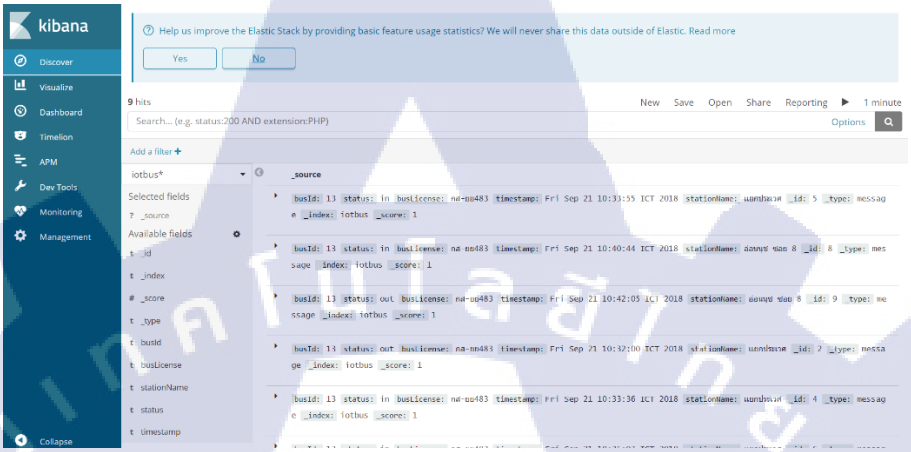
รูปที่ 4.2 หน้าแรกของ Dashboard

3) เมื่อคลิกมาแล้วจะพบกับกราฟข้อมูล ดังรูป 4.2 โดยกราฟนี้จะแสดงเป็นเรียลไทม์

รูปที่ 4.3 หน้าแรกของ Dashboard 2

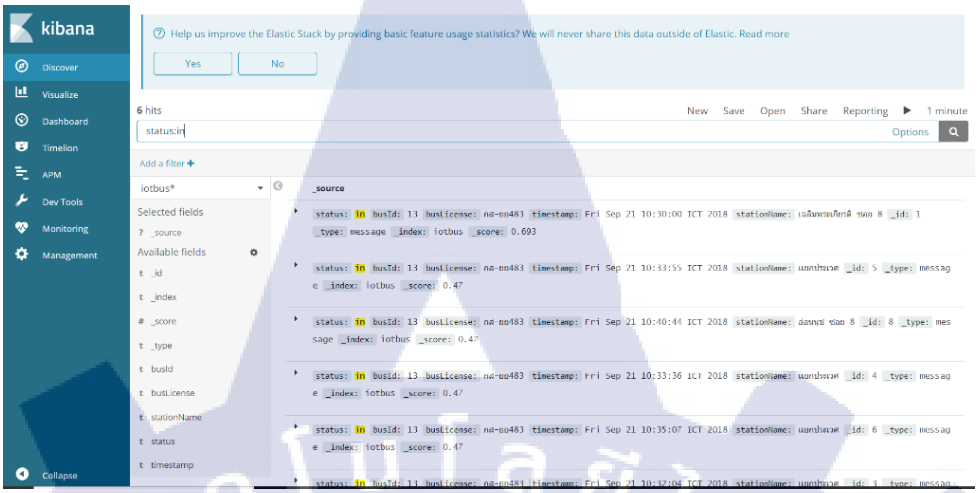
4.1.2 การแสดงผลข้อมูลผ่าน Discover

1) คลิกเมนู Discover เพื่อค้นหาข้อมูล



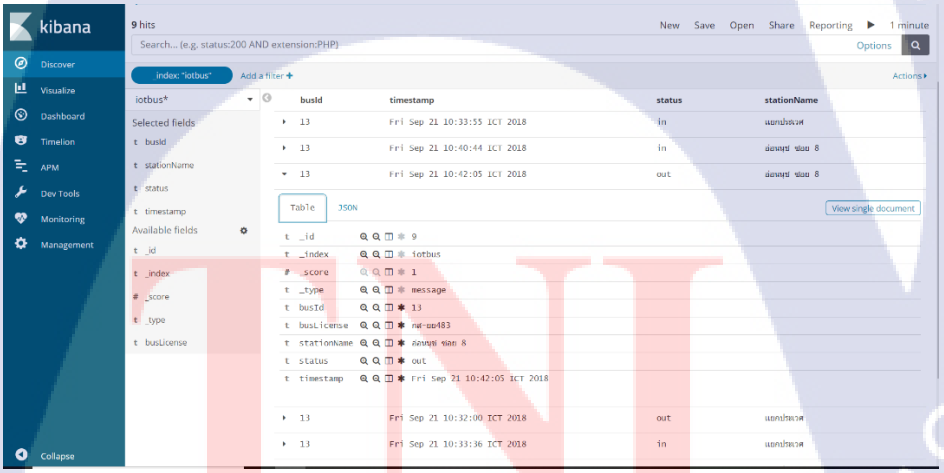
รูปที่ 4.4 หน้าเมนู Discover

2) สามารถกำหนดฟิลด์ในการแสดงและค้นหาข้อมูล เช่น ระบุเงื่อนไข Status:in โปรแกรมจะไฮไลต์ข้อมูลที่ค้นหา ดังรูปที่ 4.4 เพื่อค้นหา หรือ เรียก ข้อมูล บุคคล ขาขึ้น หรือ ในกรณีที่จะหาข้อมูลเฉพาะ สถานี เช่น Station.keyword : " สถานีรถไฟมักกะสัน " ส่วนนี้จะโชว์ข้อมูลเฉพาะที่ ป้าย สถานีรถไฟมักกะสัน เท่านั้น ดังรูป 4.5



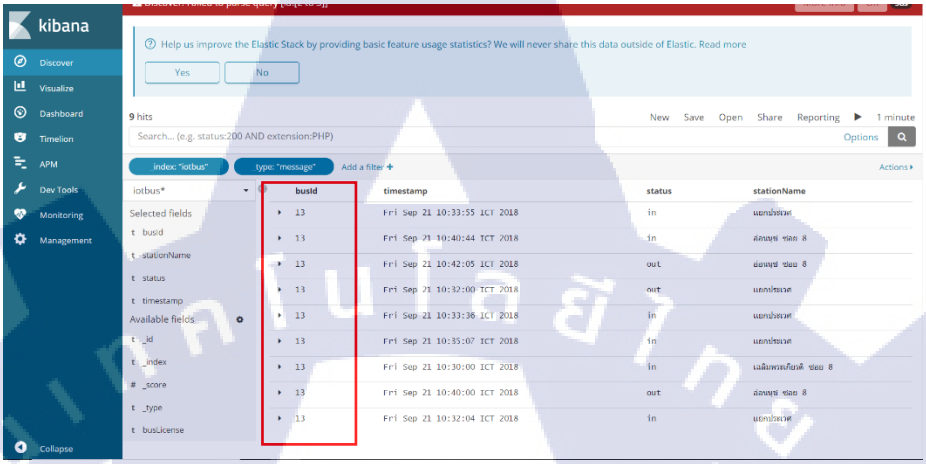
รูปที่ 4.5 ค้นหาข้อมูล

3) สามารถแสดงข้อมูลในตารางโดยคลิก ไอคอนลูกศร จะขยายข้อมูล เพื่อดู type หรือขนาด ได้ ดังรูป 4.6



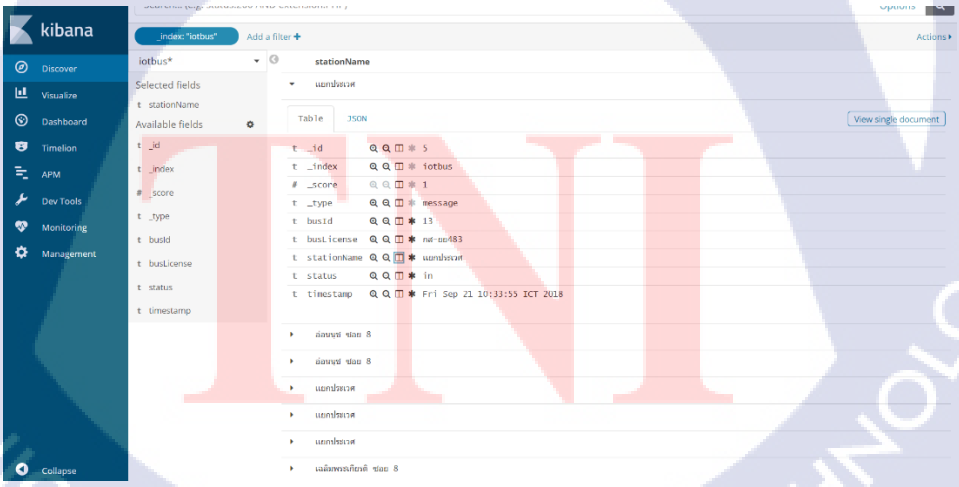
รูปที่ 4.6 ขยายข้อมูลตาราง

4) สามารถ Filter for value เครื่องมือสำหรับกรองค่าข้อมูลที่เลือก เช่น busId : “13” ดังรูป 4.7



รูปที่ 4.7 รูป Filter for value

5) สามารถ Toggle column ใน table เครื่องมือสำหรับกรองเฉพาะข้อมูลของคอลัมน์ที่เลือก เช่น กรองข้อมูลเฉพาะ ป้ายรถเมล์ ก็จะปรากฏ ข้อมูลฉพาะป้ายรถเมล์เท่านั้น ดังรูป 4.8



รูปที่ 4.8 รูป Toggle column in table

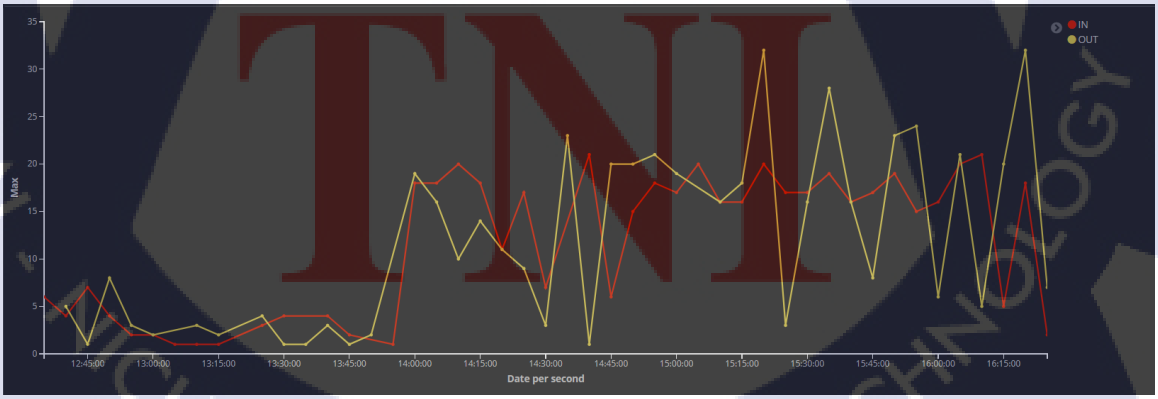
6) สามารถบันทึกเงื่อนไขการค้นหาข้อมูล โดยเลือกที่ Save ดังรูป 4.9

รูปที่ 4.9 การเซฟ ข้อมูล

4.2 ผลการวิเคราะห์ข้อมูล

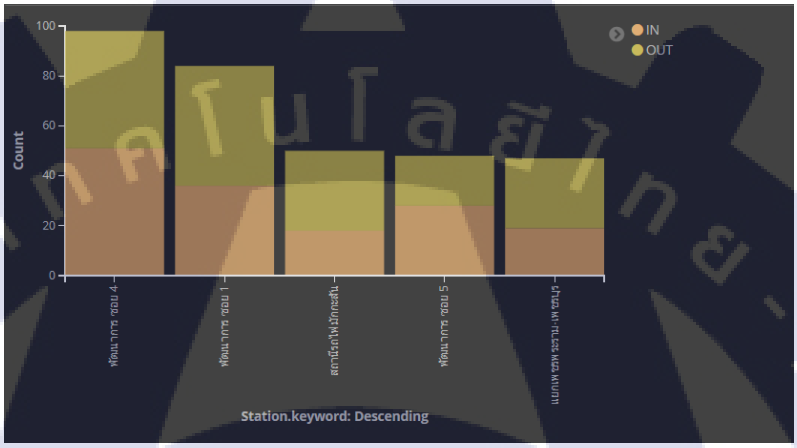
โดยกราฟหลักจะมี 3 กราฟ คือ กราฟเปรียบเทียบจำนวนคนขึ้น-ลง ในทุก 15 นาที , แผนภูมิคอลัมน์แบบเรียงซ้อน เปรียบเทียบการขึ้น-ลง ของ แต่ละป้าย และ แผนภูมิวงกลม 5 อันดับ จำนวนคนขึ้น-ลง

โดยข้อมูลนั้นจะมาจากการ สร้างจำลองข้อมูล จาก IoT รถเมล์ โดยจะจำลองข้อมูลโดยการสุ่ม จำนวนคนเข้าออก ดังรูป 4.10



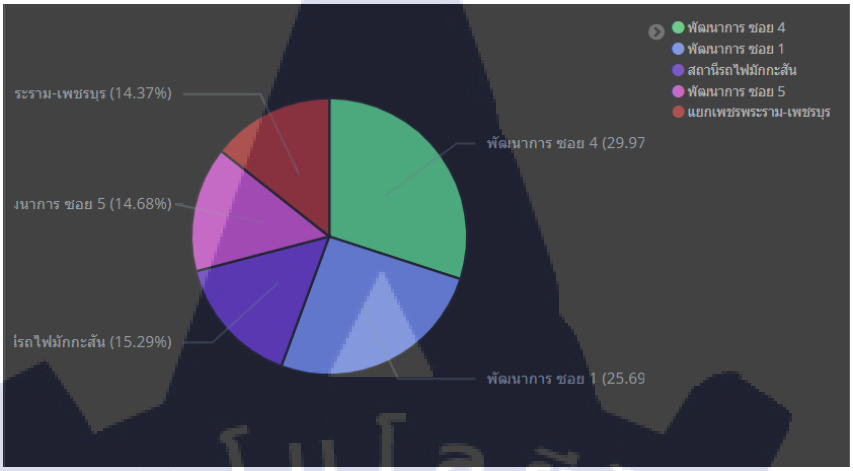
รูปที่ 4.10 กราฟเปรียบเทียบจำนวนคนขึ้น-ลง ในทุก 15 นาที

จากภาพที่ 4.10 เป็นกราฟที่แสดงถึงในแต่ละ15 นาที มีความถี่ ในการขึ้นลงของผู้โดยสารมากน้อยแค่ไน โดยอ้างอิงข้อมูลมากจากข้อมูลที่ต้นทางทางส่งมา และกราฟสี แดงจะแสดง จะแสดงความถี่ จำนวนคนขึ้นรถเมล์ ส่วนกราฟสีเหลืองนั้น จะแสดงความถี่ จำนวนคนลงรถเมล์ ลง โดยมีการเปรียบเทียบกันระว่างทั้งสอง



รูปที่ 4.11 แผนภูมิคอลัมน์แบบเรียงซ้อน เปรียบเทียบการขึ้น-ลง ของ แต่ละป้าย

จากรูปที่ 4.11 เป็นกราฟที่แสดงถึงการขึ้นลงของผู้โดยสาร โดย กราฟ สีเขียวจะจะมีข้อมูลของผู้โดยสารขาขึ้นส่วน กราฟสีน้าเงินจะเป็นข้อมูลของผู้โดยสารขาลง โดยมีการเปรียบเทียบกันระว่างทั้งสอง



รูปที่ 4.12 แผนภูมิวงกลม 5 อันดับ จำนวนคนขึ้น-ลง

จากรูป ที่ 4.12 เป็นแผนภูมิที่แสดงถึง จำนวนความถี่ ขึ้น-ลง รถเมล์ ตามสถาณที่ ที่มีคนขึ้นลงมากที่สุด 5 อันดับ โดยผลลัพธ์จะถูกเสนอเป็นเปอเซ็น

4.3 วิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายการปฏิบัติงานหรือจัดทำโครงงาน

ตารางที่ 4.1 เปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมาย

วัตถุประสงค์	จุดมุ่งหมายการปฏิบัติงาน
1.เพื่อศึกษาข้อมูลผู้ใช้บริการรถเมล์สาย 11	ข้อมูลของผู้ใช้งานรถเมล์สาย 11 ซึ่งเป็นข้อมูลสมมุติ จากประสบการณ์ของผู้ใช้งานรถเมล์
2.เพื่อวิเคราะห์ข้อมูลการขึ้น-ลง ของ ผู้ใช้บริการรถเมล์สาย 11 เล็ก	ผลวิเคราะห์การขึ้นลง ของ ผู้ใช้บริการรถเมล์สาย 11 เล็ก ในรูปแบบของแผนภูมิกราฟ คือ 1) แผนภูมิคอลัมน์แบบเรียงซ้อน เปรียบเทียบการขึ้น-ลง ของ แต่ละป้าย 2) แผนภูมิวงกลม 5 อันดับ จำนวนคนขึ้น-ลง

ตารางที่ 4.1 เปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมาย (ต่อ)

วัตถุประสงค์	จุดมุ่งหมายการปฏิบัติงาน
3.เพื่อวิเคราะห์ข้อมูลช่วงเวลาของผู้ใช้บริการรถเมล์สาย 11 เล็ก	ผลวิเคราะห์ข้อมูลช่วงเวล ของผู้ใช้บริการรถเมล์สาย 11 ในการเปรียบเทียบ การขึ้น ลง ทุก 15นาที แบบเรียลไทม์ เพื่อ ให้พิจารณาปล่อยรถ

จุดมุ่งหมายของโปรแกรม คือ การค้นคว้าข้อมูลเพิ่มเติมนั้นช่วยให้การปฏิบัติงานให้มีประสิทธิภาพมากขึ้น เช่น การค้นหา ข้อมูลเกี่ยวกับการเปลี่ยนตารางรถเมล์, ปัญหาต่างๆ ของรถเมล์ตามกระทู้ต่างๆ เนื่องจากข้อมูลเหล่านี้เป็นหาจริง และ ข้อมูลจริงที่ถูกต้อง จากนั้น มาอางอิง เพื่อ สุ่มข้อมูลที่มีความใกล้เคียงมากที่สุด และ ตรงกับวัตถุประสงค์ที่ต้องการ งานที่ได้ปฏิบัตินั้นช่วยให้ได้ประสบการณ์ที่ดี ได้รับทักษะทางด้านต่างๆ แนวคิดหลกัการวางแผนงานให้สามารถปฏิบัติงานได้อย่างมีระบบแบบ แผน ไดรับททักษะการพูด หรืออธิบายให้ผู้ฝังเข้าใจเจตนาของการปฏิบัติงานนั้นๆ และการปฏิติงาน ร่วมกับผู้อีกให้งานออกมามีประสิทธิภาพ

ในส่วนของงานที่ได้รับมอบหมายให้พัฒนาระบบนั้นสำเร็จไปได้ด้วยดี สามารถนำผลงานในส่วนนั้นรวมกับระบบหลักเพื่อนำเสนอให้บุคคลภายนอกได้ ในส่วนของกระบวนการพิมพ์รายงาน ที่เกี่ยวข้องกับระบบนั้น ได้เปลี่ยนแปลงจากการสร้างรายงานภายในระบบโดยตรงเป็นการใช้ โปรแกรมอื่นที่มีความสามารถถ้าด้านการสร้าง และออกรายงานเข้ามาร่วมพัฒนาระบบแทน เพื่อ เสริมข้อจำกัดของโปรแกรมหลักให้ได้ผลลัพธ์ของรายงานที่มีประสิทธิภาพมากขึ้น

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินโครงงาน

จากการที่ได้มาสหกิจศึกษาที่ บริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จ ากัด (มหาชน) ซึ่งในขณะที่ผมได้เข้ามาสหกิจศึกษาอยู่นั้น ทางบริษัทได้มีการทำโครงงานใหญ่ที่ต้องจัดส่งมอบให้ลูกค้าอยู่ และมีความต้องการทำให้รายงานแสดงผลเร็วขึ้น ผมจึงได้มีส่วนร่วมในการเข้าทำโครงงานนี้ด้วย ซึ่งงานที่ผมได้รับมอบหมายนั้นมีตั้งแต่ ทำ Store Procedures แก้บัค ทำ Web Chat Application จากนั้นทางบริษัท มีความต้องการที่จะทำ IoT Big data stream เพื่อนำข้อมูล มาทำเป็นกราฟให้อ่านง่าย โดยกราฟ ที่ได้ จะต้อง เรียลไทม์

5.2 แนวทางการแก้ไขปัญหา

1 . ควรเพิ่มประสิทธิภาพในการทำ Coding ให้มากขึ้น เพื่อไม่ให้เกิด Bug หรือ Error เวลาทำการทดสอบ

2. ควรศึกษาข้อมูลให้มาก ก่อนลงมือทำ

5.3 ข้อเสนอแนะจากการดำเนินงาน

ควรมีการเตรียมตัว วางแผน ให้ละเอียดรอบคอบมากกว่านี้ก่อนทำการทดสอบ เพื่อที่จะได้ไม่ให้เกิดปัญหาในเรื่องของความไม่แน่นอนในการทำการทดสอบหรือการเปลี่ยนแผนการทดสอบ 4-5 บรรทัด

เอกสารอ้างอิง

[1] hostpacific , 2017, ทำความรู้จัก Docker และการใช้งานบน CentOS 7 [Online], 13 September 2018, Available : https://www.hostpacific.com/using-docker-on-centos7/ .

[2] Somkiat Puisungnoen, 2016, Elasticsearch กับการจัดการข้อมูล [Online], 13 September 2018, Available : http://www.somkiat.cc/elasticsearch-your-data-flow/ .

[3] Somkiat Puisungnoen, 2014, ว่าด้วยเรื่องของ Fast Data [Online], 14 September 2018, Available : http://www.somkiat.cc/fast-data/.

[4] Somkiat Puisungnoen, 2018, CC :: SOMKIAT บันทึกการเรียน Kafka 101 :: Part 1 เรื่อง Messaging system [Online], 13 September 2018, Available : http://www.somkiat.cc/note-kafka-101-part-1-messaging-system .

[5] Bhuridech Sudsee , 2017, เกริ่นก่อน Apache Spark คืออะไร? [Online], 15 September 2018, Available : https://medium.com/@aorjoa/เกริ่นก่อน-apache-spark-คืออะไร-e2cebd87af2d.

[6] Somkiat Puisungnoen, 2017, Kibana คืออะไร ติดตั้งอย่างไร ใช้งานอย่างไร พร้อมตัวอย่าง [Online], 18 September 2018, Available : https://medium.com/open-source-technology/ kibana-คืออะไร-ติดตั้งอย่างไร-ใช้งานอย่างไร-พร้อมตัวอย่าง-cbf604b62ab9.

[7] Trust me I'm Petdo 2015, รู้จักกับภาษา Scala พี่น้องของ Java [Online], 19 September 2018, Available : https://www.nomkhonwaan.com/2016/02/13/scala-1-getting-started-with-scala.

ประวัติผู้จัดทำโครงงานสหกิจ

ชื่อ-สกุล	นาย วีรชัย วิเชียรรัตน์
วันเดือนปีเกิด	12 ธันวาคม 2539
ประวัติการศึกษา	
ระดับประถมศึกษา	ประถมศึกษา โรงเรียนกฤตศิลป์วิทยา
ระดับมัธยมศึกษา	มัธยมศึกษาตอนต้น โรงเรียนเตรียมอุดมศึกษาพัฒนาการ มัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษาพัฒนาการ
ระดับอุดมศึกษา	คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ พ.ศ. 2557 สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ทุนการศึกษา	-
ประวัติการฝึกอบรม	- Basic Front End Programmer สวทน.
ผลงานที่ได้รับการตีพิมพ์	- ไม่มี -

ภาคผนวก ก.

อธิบายการทำงานของโค้ด Apache Spark

อธิบายการทำงานของ Spark

```
import org.apache.kafka.clients.consumer.ConsumerRecord
import org.apache.kafka.common.serialization.StringDeserializer
import org.apache.spark.streaming.kafka010._
import org.apache.spark.streaming.kafka010.LocationStrategies.PreferConsistent
import org.apache.spark.streaming.kafka010.ConsumerStrategies.Subscribe

// Kafka parameters
val kafkaParams = Map(
    "bootstrap.servers" -> s"$myIP:9092",
    "value.deserializer" ->classOf[StringDeserializer],
    "key.deserializer" -> classOf[StringDeserializer],
    "auto.offset.reset" -> "latest",
    "enable.auto.commit" -> (false: java.lang.Boolean)
    )

// Consume the "train" topic

// Consume the "test" topic
val testInputStream = KafkaUtils.createDirectStream[String, String](
  ssc,
  PreferConsistent,
  Subscribe[String, String](Array("test"), kafkaParams + ("group.id" -> "group2"))
)
```

รูปที่ 1.1 อธิบายโค้ด ขั้นตอนที่ 1

จากรูปที่ 1.1 ส่วนบนจะมีการ import ไลบรารี่ apache spark เพื่อทำการใช้ function ของ Spark ได้ โดยขั้นตอนถัดไป ต้องทำการ map ไป IP ที่ต้องการ ตามด้วย หมายเลข Port 9092 ซึ่ง Port นี้ถูกสำรองไว้ให้ Kafka เชื่อมต่อกับ Spark โดยเฉพาะ

testInputStream คือ ตัวแปรที่เก็บ ค่าการเชื่อมต่อ ระหว่าง Spark และ Kafka โดยมี Header คือ “ Test ” และ Group id คือ group2

```
import org.apache.spark.mllib.linalg.Vectors
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.sql.types.{StringType, StructType, TimestampType,DateType,LongType}

val schema = new StructType()
      .add("Bus_ID",StringType)
      .add("DateTime",TimestampType)
      .add("Station",StringType)
      .add("Status",StringType)

  testInputStream.foreachRDD { (rdd, time) =>
      val data = rdd.map(record => record.value)
      val json = spark.read.schema(schema).json(data)
     // val result = json.groupBy($"action").agg(count("*").alias("count"))
      json.show

      import org.elasticsearch.spark.sql._
      json.saveToEs("zvzz/docs",  Map("es.nodes" -> "elk", "es.port" -> "9200"))
}
```

1 2 3

รูปที่ 1.2 อธิบายโค้ด ขั้นตอนที่ 2

จากรูปที่ 1.2 หมายเลข 1 จะเป็น การกำหนดต้นแบบ schema ของข้อมูล โดย กำหนดให้ Bus_ID คือ StringType, DateTime คือ TimestampType ,Station คือ StringType และ Status คือ StringType

หมายเลข 2 จะเป็นการอ่านข้อมูล จาก testInputStream ที่เก็บตัวแปรการเชื่อมต่อไว้แล้ว และทำการลูปข้อมูล โดย ข้างในลูปนั้น จะมีตัวแปร ที่ชื่อว่า data และ json ซึ่งตัวแปร data จะเก็บ ข้อมูลที่ถูกส่งมา โดยข้อมูลนี้จะถูกเข้ากระบวนการ map ทั้งหมด ตัวแปร json จะเก็บข้อมูล data โดยแปลง data ให้อยู่ในรูป json โดยอ้างอิง schema ตามตัวแปร schema

หมายเลข 3 จะเข้าสู่กระบวนการ ส่งข้อมูลไป Elasticsearch โดย header จะเป็น zvzz และจะถูกส่งไปยัง port 9200

ภาคผนวก ข.

รายงานประจำสัปดาห์