การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ

RESOURCE MONITORING AND VISUALIZATION INTERFACE FOR FOREIGH EXCHANGE (FX) ELECTRONIC TRADING

รายงานนี้เป็นส่วนหนึ่งของการจัดการศึกษารายวิชาสหกิจศึกษา
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 1 ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ

RESOURCE MONITORING AND VISUALIZATION INTERFACE FOR FOREIGN EXCHANGE (FX) ELECTRONIC TRADING

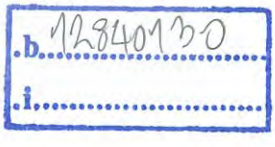
โดย

สุประวีณ์ ยิ้มเนียม

SUPRAWEE YIMNIUM

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. ธีรพงศ์ ลีลานุภาพ



รายงานนี้เป็นส่วนหนึ่งของการจัดการศึกษารายวิชาสหกิจศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ

RESOURCE MONITORING AND VISUALIZATION INTERFACE FOR FOREIGN EXCHANGE (FX) ELECTRONIC TRADING

โดย

สุประวีณ์ ยิ้มเนียม

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตรจารย์ ดร. ธีรพงศ์ ลีลานุภาพ

รายงานนี้เป็นส่วนหนึ่งของการจัดการศึกษารายวิชาสหกิจศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESOURCE MONITORING AND VISUALIZATION INTERFACE FOR FOREIGN EXCHANGE (FX) ELECTRONIC TRADING

SUPRAWEE YIMNIUM

A REPORT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR COOPERATING EDUCATION PROGRAM
THE DEGREE OF BACHELOR OF SCIENCE PROGRAM IN
INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1/ 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2015

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองรายงานโครงงานสหกิจศึกษา ประจำปีการศึกษา 2558

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ

RESOURCE MONITORING AND VISUALIZATION INTERFACE FOR FOREIGN EXCHANGE (FX) ELECTRONIC TRADING

ผู้จัดทำ

นางสาวสุประวีณ์ ยิ้มเนียม รหัสนักศึกษา 55070133

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. ธีรพงศ์ ลีลานุภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ
นักศึกษา นางสาวสุประวีณ์ ยิ้มเนียม รหัสนักศึกษา 55070133
ปริญญา วิทยาศาสตรบัณฑิต
สาขาวิชา เทคโนโลยีสารสนเทศ
ปีการศึกษา 2558
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. ธีรพงศ์ ลีลานุภาพ

บทคัดย่อ

ปัจจุบันมีความต้องการที่จะเก็บข้อมูลจำนวนมากที่รวบรวมจากการทำธุรกรรมในแต่ละวันโดยเฉพาะอย่างยิ่งการเก็บข้อมูลธุรกรรมเกี่ยวกับการแลกเปลี่ยนหลักทรัพย์เงินตราต่างประเทศ ซึ่งข้อมูลมีการเปลี่ยนแปลงเป็นจำนวนมากในแต่ละวัน ซึ่งการเข้าถึงข้อมูลจำนวนมากอาจมีผลทำให้เกิดความล่าช้า ทีม Reuters Electronic Trading (RET) ของบริษัท รอยเตอร์ ซอฟแวร์ (ประเทศไทย) จำกัด มีหน้าที่รับผิดชอบการพัฒนาและบำรุงรักษาการทำงานของซอฟแวร์ด้านการเงิน และมีการใช้งานโปรเซสต่างๆจำนวนมากอยู่บนเครื่องเซิฟเวอร์ ดังนั้นการเคลื่อนไหวของข้อมูลจะถูกเปลี่ยนแปลงตลอดเวลา การตรวจสอบการใช้ทรัพยากรของ CPU และ Memory สามารถทำได้ยาก เนื่องจากไฟล์ที่ถูกสร้างขึ้นเพื่อใช้ในการตรวจสอบประกอบด้วยข้อมูลดิบจำนวนมาก การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศจึงถูกพัฒนาเพื่อลดความยากลำบากในการตรวจสอบข้อมูล โดยระบบที่ถูกพัฒนาขึ้นสามารถรวบรวมข้อมูลและนำมาแสดงผลในรูปแบบกราฟทั้งกราฟแบบเรียลไทม์และกราฟแบบย้อนหลัง ดังนั้นผู้ใช้จึงสามารถตรวจสอบการใช้ทรัพยากรได้อย่างสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Resource Monitoring and Visualization Interface for Foreign Exchange (FX) Electronic Trading
Student	Miss Suprawee Yimnium Student ID 55070133
Degree	Bachelor of Science
Program	Information Technology
Academic Year	2015
Advisor	Assistant Professor Dr. Teerapong Leelanupab

ABSTRACT

Nowadays, there are needs to store a lot of data, collected from many transactions occurring daily. This is particularly more important for transactions in the domain of foreign exchange, which may happen over a million per day. Accessing such a large amount of data may result in the delay to respond financial needs. The RET team at Reuter Software (Thailand) Limited is responsible for developing and maintaining financial software. There are a lot of processes run on servers; so, data flow can be updated all the time. Monitoring usage of resources, i.e., CPU and memory, is extremely hard because the files generated from server have a lot of raw data. The Resource Monitoring and Visualization Interface for Foreign Exchange (FX) Electronic Trading has been developing to decrease hardness of monitoring. The system can gather data and display both real time graph and historical graph. Users can thus monitor resource usage more conveniently.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การเข้าร่วมโครงการสหกิจศึกษา ณ บริษัท รอยเตอร์ ซอฟแวร์ (ประเทศไทย) จำกัด ที่ทางคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จัดทำขึ้น ทำให้ได้รับความรู้และประสบการณ์ในการทำงานจริง ทั้งนี้โครงการได้รับการอนุเคราะห์จากคณะอาจารย์ในคณะเทคโนโลยีสารสนเทศและบุคลากรในสถานประกอบการข้างต้นที่ได้ให้ความช่วยเหลือดำเนินการติดต่อประสานงานจนโครงการสำเร็จลุล่วงไปได้ดี

ขอกราบขอบพระคุณอาจารย์ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ ดร. ธีรพงศ์ ลีลานุภาพ ที่ได้สละเวลาให้คำแนะนำให้คำปรึกษาและชี้แนะแนวทางแก้ไขปัญหาที่เป็นประโยชน์

คุณปรียวิศว์ กัณฑเจตน์ Development Manager ของทีม Reuters Electronic Trading (RET) ที่ให้การต้อนรับ การดูแล และการช่วยเหลือเป็นอย่างดี

คุณภัทร์ รัศมีไพบูลย์ Senior Software Engineer, คงสกุล ชุมคง Lead Software Engineer ที่ได้สละเวลาให้ความช่วยเหลือ ให้คำแนะนำในการวางแผนและปรับปรุงแก้ไขปัญหาต่างๆ ตลอดจนให้ความรู้และประสบการณ์ที่ดีในการทำงาน

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่ให้กำลังใจและให้การสนับสนุนในทุกๆเรื่อง ทำให้ข้าพเจ้าสามารถทำรายงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

สุประวีณ์ ยิ้มเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและที่มาของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 ขอบเขตการพัฒนาระบบ 2

1.4 แผนการดำเนินงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 ทฤษฎีและหลักการ 4

2.1 องค์ประกอบทางด้านซอฟแวร์ 4

2.1.1 ส่วนของการทำงานบนเว็บไซต์ 4

2.1.1.1 Elasticsearch 4

2.1.1.2 Logstash 6

2.1.1.3 Kibana 7

2.1.1.4 ภาษา HTML 8

2.1.1.5 ภาษา Javascript 9

2.1.1.6 ภาษา JQuery 9

2.1.1.7 ภาษา PHP 9

2.1.2 ส่วนของ LT Monitor Client 10

2.1.2.1 ภาษา JAVA 10

บทที่ 3 การวิเคราะห์และออกแบบระบบ 11

3.1 การศึกษาระบบงานเดิม 11

3.2 ปัญหาที่พบในระบบปัจจุบัน 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3 การวิเคราะห์ความต้องการระบบ (System Requirement Analysis) 12
3.3.1 ความต้องการที่เป็นหน้าที่หลักของระบบ 12
3.3.2 ความต้องการที่ไม่ใช่หน้าที่หลักขอวระบบ 12
3.4 การออกแบบระบบใหม่ 12
3.4.1 ภาพรวมของระบบ 12
3.4.2 แผนภาพยูสเคส (Use Case Diagram) 14
3.4.2.1 ผู้ที่เกี่ยวข้องในระบบ (Actor) 14
3.4.2.2 องค์ประกอบของยูสเคส 14
3.4.2.3 แผนภาพยูสเคส (Use Case Diagram) 14
3.4.2.4 รายละเอียดยูสเคส (Use Case Description) 15
3.4.3 แผนภาพกิจกรรม (Activity Diagram) 18
3.4.4 แผนภาพแสดงลำดับเหตุการณ์ (Sequence Diagram) 21

บทที่ 4 ผลการออกแบบและผลการทำงานของระบบ 22
4.1 การออกแบบระบบ 22
4.1.1 การออกแบบส่วนต่างๆบนเว็บไซต์ 22
4.1.1.1 การออกแบบส่วนประสานงานกับผู้ใช้ (GUI) 22
4.1.1.2 การออกแบบส่วนการทำงานของซอฟแวร์ 23
4.1.1.3 การออกแบบส่วนการทำงานของการจัดการข้อมูล 23
4.1.2 การออกแบบส่วนต่างๆบน JAVA application 30
4.1.2.1 การออกแบบส่วนประสานงานกับผู้ใช้ (GUI) 30
4.1.2.2 การออกแบบส่วนการทำงานของซอฟแวร์ 31
4.2 ผลการทำงานของระบบ 37
4.2.1 ผลการทำงานของเว็บไซต์ 37
4.2.2 ผลการทำงานของ JAVA application 41

บทที่ 5 สรุปผลโครงงานและข้อเสนอแนะ 45
5.1 สรุปผลโครงงาน 45
5.2 ปัญหาและอุปสรรค 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5.2.1 ปัญหาที่พบ 45
5.2.2 ข้อจำกัดของระบบ 46
5.3 ข้อเสนอแนะ 46

บรรณานุกรม 47

ภาคผนวก 48
ภาคผนวก ก. ตารางการทำงาน 49

ประวัติผู้เขียน 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 คำศัพท์ที่ใช้ใน Elasticsearch....5
3.1 รายละเอียดยูสเคส Connect with LT Monitor Server....15
3.2 รายละเอียดยูสเคส Display graph on Java Application....16
3.3 รายละเอียดยูสเคส Display graph on website....17
ก.1 การทำงาน เดือนสิงหาคม 2558....50
ก.2 การทำงาน เดือนกันยายน 2558....51
ก.3 การทำงาน เดือนตุลาคม 2558....52
ก.4 การทำงาน เดือนพฤศจิกายน 2558....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างของ Logstash, Elasticsearch และ Kibana 4
2.2 โครงสร้างของ Elasticsearch 5
2.3 Plug-in head ปลั๊กอินที่ติดตั้งเพื่อแสดงฐานข้อมูล 6
2.4 ภาพรวมการทำงานของ Logstash 6
2.5 ตัวอย่างการ Config file ในการใช้งาน Logstash 7
2.6 ส่วนแสดงผลของ Kibana 8
3.1 การทำงานแบบเก่า 11
3.2 การทำงานของระบบใหม่ 13
3.3 แผนภาพยูสเคส 14
3.4 แผนภาพแสดงลำดับเหตุการณ์ 21
4.1 การแสดงผลบนเว็บไซต์ 22
4.2 ไฟล์ข้อมูลจากเซิฟเวอร์ 23
4.3 ไฟล์การรับข้อมูลเข้า Logstash 24
4.4 ไฟล์การแปลงข้อมูลเข้า Logstash 24
4.5 ไฟล์การกำหนดปลายทางข้อมูลออกจาก Logstash 25
4.6 การรัน logstash.conf 25
4.7 การ Config Cluster name 26
4.8 การดาวโหลด Plug in head 26
4.9 การรันไฟล์ Elasticserach.bat 27
4.10 การเก็บข้อมูลบน Elasticsearch 27
4.11 การ Config การทำงานของ Kibana 28
4.12 การรัน Kibana.bat 28
4.13 การแสดงผลกราฟเส้นใน Kibana 29
4.14 Embed Visualize Kibana 29
4.15 การแสดงผลใน LT Monitor Client 30
4.16 แสดงโครงสร้างการทำงานของ Java Application 31
4.17 แผนภาพแสดงการทำงานหลักการส่งข้อมูล 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.18 การส่งข้อมูลจาก RealtimeUI 33
4.19 การส่งข้อมูลจาก LTConnectionManager ผ่าน onMessage 33
4.20 การส่งข้อมูลจาก LTConnection Manager ไปยัง LTConnection 33
4.21 การรับข้อมูลจาก LTConnectionManager 34
4.22 การส่งข้อมูลจาก LTConnection 34
4.23 การรับข้อมูลผ่าน onMessage และนำไปประมวลผลต่อ 34
4.24 แสดงโครงสร้างการทำงานของการอัพเดทข้อมูล 35
4.25 ออบเจ็คการเก็บข้อมูล 35
4.26 การรับข้อมูลผ่าน onMessage และส่งข้อมูลไป LTWriteFile 36
4.27 การสร้างออบเจ็คของ Update และเก็บลงใน HashMap updateMap 36
4.28 ตัวอย่างการตั้งค่าข้อมูลในออบเจ็คของ GraphComponent 36
4.29 การเริ่มต้นเพื่ออัพเดทข้อมูล 37
4.30 การเลือกโปรแกรมบนเว็บไซต์ 38
4.31 การเลือกรูปแบบแสดงผลกราฟบนเว็บไซต์ 38
4.32 การเลือกเวลาที่ต้องการดูกราฟบนเว็บไซต์ 39
4.33 การแสดงผลกราฟแบบ Historical Data 39
4.34 การเลือกรูปแบบของเวลาแบบ Real-time บนเว็บไซต์ 40
4.35 การแสดงผลกราฟแบบ Real-time 40
4.36 การแสดงผลกราฟแบบ Multiple graph 41
4.37 ส่วนต่างๆในโปรแกรมเมื่อมีการเริ่มใช้ 41
4.38 เมื่อมีการเชื่อมต่อ 42
4.39 การแสดงผลกราฟเส้นของ CPU 43
4.40 การแสดงผลกราฟเส้นของ Memory 43
4.41 การแสดงผลกราฟวงกลมของ CPU 44
4.42 การแสดงผลกราฟวงกลมของ Memory 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบันเทคโนโลยีกำลังก้าวเข้าสู่ Big Data เนื่องจากข้อมูลเริ่มมีจำนวนมากและหลากหลายมากขึ้น จนทำให้การเก็บข้อมูลในรูปแบบ Database ในรูปแบบธรรมดาอาจไม่สามารถรองรับการทำงานได้อย่างมีประสิทธิภาพเพียงพอ ถ้าทำได้ก็อาจไม่ครอบคลุมตามความต้องการการทำงานได้ทั้งหมด หรือการจัดการกับข้อมูลอาจต้องใช้เวลามากพอสมควร เช่น การดึงข้อมูลมาใช้ให้ตรงตามเวลา, การประมวลผลข้อมูล ซึ่งการทำงานดังกล่าวของการเก็บข้อมูลที่มีปริมาณมาก, ข้อมูลที่มีความหลากหลาย และข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลาอย่างรวดเร็วเสมือนเป็นข้อมูลแบบ Real-time เป็นหลักการที่สำคัญของเทคโนโลยี Big Data ประกอบกับทีมเซิฟเวอร์ (Server Team) ในบริษัท รอยเตอร์ ซอฟแวร์ (ประเทศไทย) จำกัด พัฒนาซอฟแวร์ในการซื้อขายหลักทรัพย์ทางการเงิน โดยงานส่วนหนึ่งที่สำคัญของทีม คือ ทำหน้าที่ดูแลและจัดการ การใช้ทรัพยากรของโปรแกรมต่าง ๆ ที่ทำงานอยู่ในแต่ละเซิฟเวอร์ ซึ่งการใช้ทรัพยากรต่าง ๆ เมื่อมีการใช้งานจะมีการอัพเดทโปรแกรมในเซิฟเวอร์ต่าง ๆ ที่มีอยู่ทั่วโลกอย่างรวดเร็วและมีจำนวนของข้อมูลมากขึ้นเรื่อย ๆ เมื่อผู้ใช้ทำการสังเกตการทำงานของเซิฟเวอร์จะต้องเข้าไปดูไฟล์ซึ่งเก็บ Log ที่ทีมได้พัฒนาขึ้น ซึ่งการสังเกตการณ์, การเปรียบเทียบการใช้ทรัพยากรของโปรแกรมในเซิฟเวอร์ และการวิเคราะห์ข้อมูลจากไฟล์ที่มีจำนวนมากนั้นอาจทำได้ไม่คล่องตัวเท่าที่ควรหรือหากเกิดปัญหา เช่น ความผิดปกติจากการใช้ทรัพยากรมากเกินไปของบางโปรแกรมในเซิฟเวอร์ อาจทำให้แก้ไขปัญหาได้ล่าช้า ซึ่งปัญหาการจัดเก็บข้อมูลดังกล่าวตรงกับหลักการทำงานของ Big Data คือ ข้อมูลมีจำนวนในปริมาณมาก (Volume) และข้อมูลมีการอัพเดทอย่างรวดเร็ว (Velocity) หรือเป็นข้อมูลประเภท Real-time ดังนั้น ระบบส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ จึงถูกพัฒนาขึ้น โดยใช้หลักการทำงานของ Big Data และนำเทคโนโลยีที่มีอยู่มาประยุกต์การทำงานเพื่อให้สอดคล้องกับระบบและความต้องการขององค์กร เพื่อให้ผู้ใช้ที่ต้องการสังเกตการณ์การใช้ทรัพยากรของเซิฟเวอร์ สามารถทำงานได้อย่างมีประสิทธิภาพ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อพัฒนาระบบสังเกตการณ์การใช้ทรัพยากร (CPU และ Memory) ของเซิฟเวอร์ (Server) ผ่านการแสดงผลบนเว็บไซต์และ JAVA application
2. เพื่อการแสดงผลที่สามารถเปรียบเทียบการใช้ทรัพยากรในแต่ละโปรเซสผ่านกราฟ
3. เพื่อการแสดงผลที่สามารถแสดงผลได้หลายกราฟในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เพื่อแสดงผลอย่างทันเวลาโดยข้อมูลจะแสดงผลแบบ Real-Time และย้อนหลัง (Historical Data)
5. เพื่อให้ผู้ใช้แก้ปัญหาจากการทำงานที่ผิดปกติของเซิฟเวอร์ได้ทันเวลา
6. เครื่องมือที่นำมาใช้ รองรับการเก็บข้อมูลแบบ Big Data จึงสามารถนำระบบไปพัฒนาต่อยอดได้

1.3 ขอบเขตการพัฒนาระบบ

ระบบที่พัฒนาเป็นเว็บไซต์ที่พัฒนาขึ้นเพื่อสังเกตการณ์การใช้ทรัพยากร (RAM and CPU) ของเซิฟเวอร์โดยแสดงผลเป็นกราฟทั้งแบบ Real time และแบบข้อมูลย้อนหลัง (Historical Data) ซึ่งการจัดการข้อมูลถูกจัดการโดยมีคุณสมบัติต่อไปนี้

- เว็บไซต์ที่พัฒนาขึ้นจะสามารถใช้งานได้ภายในเครือข่ายขององค์กรเท่านั้น
- เว็บไซต์จะสามารถใช้งานได้ก็ต่อเมื่อรันโปรแกรม 2 ตัว คือ Elasticsearch และ Kibana ก่อนเสมอ
- ไฟล์ที่นำมาเป็นข้อมูลจะมาจากโปรแกรม LT Monitor Client เท่านั้น
- การนำไฟล์เข้าไปเก็บเพื่อเป็นข้อมูลในฐานข้อมูล (Elasticsearch) จะต้องถูกนำเข้าโดย Logstash เท่านั้น
- การแสดงผลเพื่อดูข้อมูลจะมีทางเลือกคือ แบบเปรียบเทียบ (Compare) และแบบแสดงหลายกราฟ (Multiple Graph)
- การเลือกเวลาที่ใช้ในการแสดงผลจะมีแบบแสดงข้อมูลย้อนหลัง (Historical Data) และแบบ Real time
- การอัพเดทของเว็บไซต์ในรูปแบบ Real time จะสามารถทำได้ 5 วินาที และการอัพเดทบน JAVA application จะสามารถทำได้ 4 วินาที
- การอัพเดทของข้อมูลจากเซิฟเวอร์ลงไฟล์จะทำการอัพเดททุกๆ 4 วินาที

1.4 แผนการดำเนินงาน

- ศึกษาโครงสร้างและการใช้งาน Library JFreechart ในภาษา JAVA
- สร้างกราฟแบบ Realtime โดยใช้ Library JFreechart ในภาษา JAVA เป็นส่วนหนึ่งของ LT Monitor Client โดยใช้ค่า Random
- ศึกษาโครงสร้างและการใช้งาน Elasticsearch, Logstash และ Kibana (ELK)
- ศึกษาการจัดการเพื่อการทำงานร่วมกันของ ELK
- Configuration Elasticsearch , Logstash และ Kibana ให้ทำงานได้โดยใช้ไฟล์จำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ศึกษาวิธีนำข้อมูลจาก Elasticsearch, Logstash และ Kibana มาพัฒนาบนเว็บไซต์ที่จะพัฒนาขึ้น
- ศึกษาภาษา PHP , HTML , CSS , JAVASCRIPT และ JQUERY เพื่อทำเว็บเพจ
- พัฒนาเว็บไซต์
- ศึกษาหลักการเก็บข้อมูลและการทำงานของโปรแกรมในเซิฟเวอร์
- ทดสอบการทำงานโดยเชื่อมต่อกับเซิฟเวอร์จริง
- แก้ปัญหาที่เกิดจากความผิดปกติจากการทดสอบกับเซิฟเวอร์จริง
- ศึกษาการทำงานเพื่อพัฒนาโปรแกรม LT Monitor Client
- พัฒนาโปรแกรม LT Monitor Client โดยใช้ภาษา JAVA

1.5 ประโยชน์ที่คาดว่าจะได้รับ

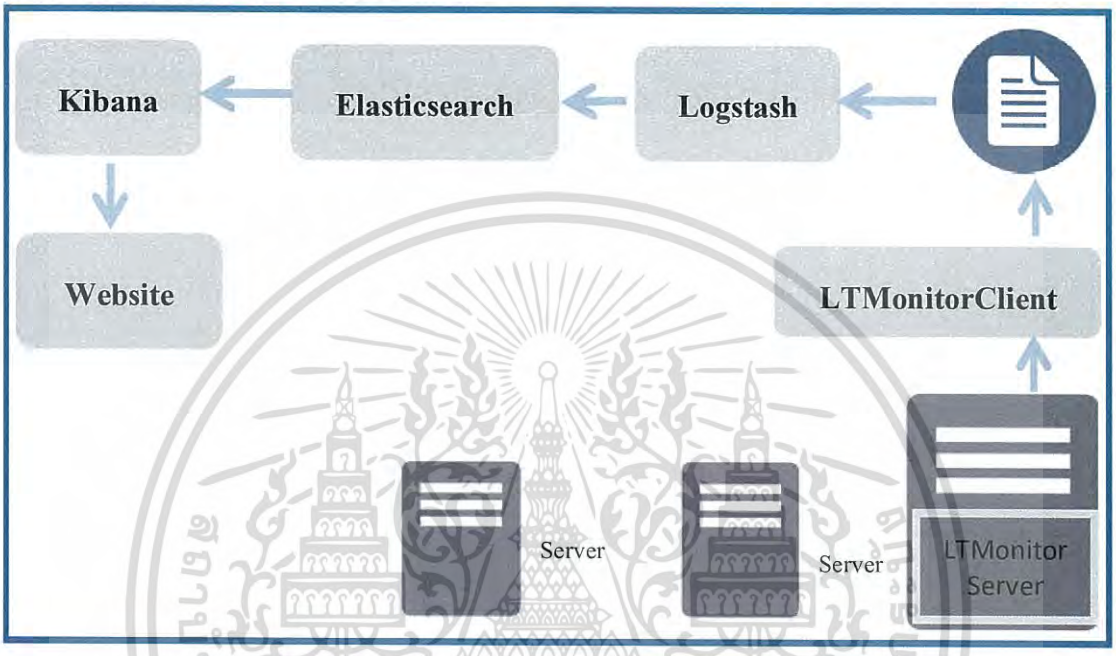
1. ได้ศึกษาความรู้ในเรื่องการจัดการข้อมูลแบบ Big Data โดยการใช้ API Elasticsearch, Logstash, Kibana (ELK)
2. การพัฒนาเว็บไซต์ด้วยภาษา PHP, HTML, JQUERY, CSS และ การพัฒนาระบบการส่งข้อมูล การแปลงข้อมูลจากเซิฟเวอร์ด้วยภาษา JAVA
3. ได้ความรู้จากการพัฒนากราฟในภาษา Java โดยใช้ Library JFreeChart
4. ผู้พัฒนาในองค์กรสามารถนำข้อมูลที่มีการอัพเดทจำนวนมากมาวิเคราะห์ เปรียบเทียบได้อย่างยืดหยุ่นและรวดเร็วเพราะสามารถจัดการข้อมูลได้แบบ Real time หรือดูข้อมูลย้อนหลังได้
5. เนื่องจากเป็นการพัฒนาบนเว็บไซต์และรองรับการเก็บข้อมูบแบบ Big Data ผู้พัฒนาในองค์กรจึงสามารถนำระบบไปพัฒนาและต่อยอดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 องค์ประกอบด้านซอฟต์แวร์

2.1.1 ส่วนของการทำงานบนเว็บไซต์



รูปที่ 2.1 โครงสร้างของ Logstash , Elasticsearch และ Kibana

2.1.1.1 Elasticsearch

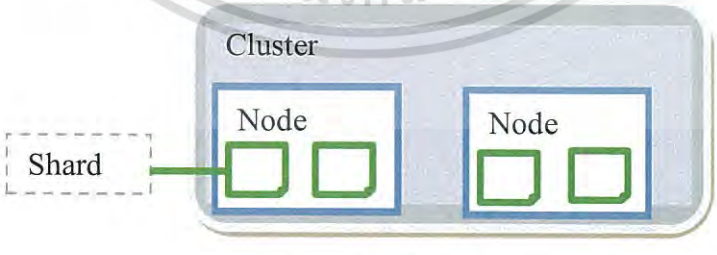
Elasticsearch เป็นเครื่องมือที่ช่วยในการเก็บข้อมูล , ค้นหาข้อมูล และวิเคราะห์ข้อมูลที่มีจำนวนมากได้อย่างรวดเร็วเป็นฐานข้อมูลประเภท NoSQL ที่สามารถเก็บข้อมูลได้ตั้งแต่ข้อมูลจำนวนน้อยไปจนถึงข้อมูลขนาดใหญ่ระดับ Big Data ได้, ประสิทธิภาพการทำงานสูงและรองรับการขยายของระบบ และสามารถเก็บข้อมูลที่มีการอัพเดท เปลี่ยนแปลงตลอดเวลาได้ จึงไม่ต้องมีการกำหนดโครงสร้างหรือจำนวนข้อมูลก่อนจึงจะสามารถเก็บข้อมูลได้เหมือนฐานข้อมูลแบบ RDBMS โดยข้อมูลที่ถูกเก็บสามารถเรียกดูได้ผ่านปลั๊กอิน head ของ Elasticsearch บนเว็บไซต์ การใช้คำศัพท์ต่างๆจะแตกต่างจากการใช้คำศัพท์ของฐานข้อมูลทั่วไป หากเปรียบเทียบกับการใช้คำใน Elasticsearch กับ RDBMS (Ralational Database Management System) จะเปรียบเทียบได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 คำศัพท์ที่ใช้ใน Elasticsearch

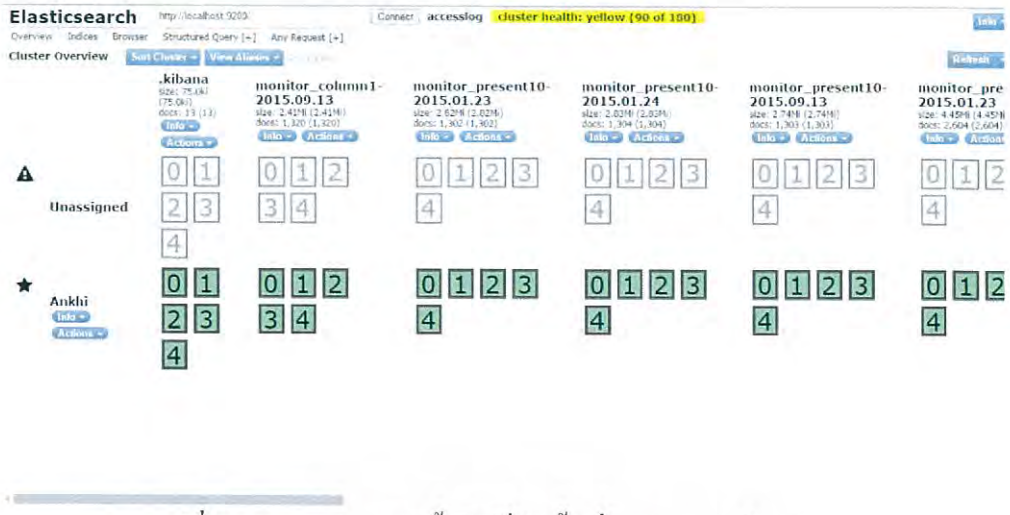
RDBMS	**Elasticsearch**
Database	Index
Table	Type
Row	Document
Column	Field
Schema	Mapping
SQL	Query DSL

การเก็บข้อมูลจะเก็บโดยส่งผ่านจาก Logstash โดยการเพิ่มฐานข้อมูลสามารถเพิ่มได้ทีละ Index ข้อมูลทั้งหมดจะถูกเก็บอยู่ใน Cluster ซึ่งภายในจะเก็บ Node (เปรียบเสมือน Server ของ Elasticsearch) ไว้ การเก็บข้อมูลหากข้อมูลใน Index มีจำนวน Document (row) มากจนเกินระดับ 1TB ไปแล้วการเข้าถึงข้อมูลที่ถูกเก็บไว้เพียง Node เดียวจำทำให้ช้ามาก เพื่อแก้ปัญหาดังกล่าว Elasticsearch จึงมีการจัดการโดยการแบ่งข้อมูลภายใน Node ออกเป็นหลายๆชิ้น เรียกว่า Shard ซึ่งเราสามารถกำหนดจำนวนของ Shard ขึ้นตอนสร้าง Index และไม่สามารถเปลี่ยนแปลงได้ภายหลัง หากไม่กำหนดจำนวนของ Shard Elasticsearch จะตั้งให้ Shard มีจำนวนเท่ากับ 5 เสมอ Elasticsearch สามารถรองรับข้อมูลแถว (Document/Rows)ได้รวมทั้งหมด 300 ล้านแถว หรือ 60 ล้านแถว (Document/Rows)ต่อ 1 Shard และหากภายใน Cluster กำหนดให้มีจำนวน Node มากกว่า 1 แล้ว Shard ของข้อมูลจะสามารถกระจายไปอยู่ใน Node ใดก็ได้ การทำงานดังกล่าวทำให้เกิดการ Scale out และประสิทธิภาพการทำงานจะเพิ่มมากขึ้น



รูปที่ 2.2 โครงสร้างของ Elasticsearch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 Plug-in head ปลั๊กอินที่ติดตั้งเพื่อแสดงฐานข้อมูล

หากต้องการตรวจสอบว่าข้อมูลถูกเก็บลงใน Elasticsearch เรียบร้อยแล้วหรือไม่ สามารถเปิดเว็บ Plug-in head เพื่อดูรายชื่อของ Index ทั้งหมดที่ถูกเก็บอยู่บน Cluster ได้

2.1.1.2 Logstash

Logstash คือ เครื่องมือสำหรับจัดการกับ log และ ข้อมูลต่างๆ โดย Logstash สามารถนำข้อมูลจำนวนมากๆมาสะสม กรอง และจัดการข้อมูลเพื่อใช้งานต่อไป ในที่นี้ Logstash จะนำข้อมูลจาก Source Data ที่ได้มาจากการแปลงข้อมูลของ LT Monitor Client ซึ่งรับข้อมูลต่างๆมาจากเซิฟเวอร์ เป็นไฟล์ข้อมูลที่รูปแบบข้อมูลถูกกำหนดขึ้นมาเอง Logstash นำข้อมูลไปประมวลผลและจัดรูปแบบของข้อมูลให้อยู่ในรูปแบบที่ต้องการและส่งออกไปเก็บไว้ที่ปลายทางที่ระบุไว้ได้ ในระบบนี้ใช้ปลายทางเป็นElasticsearch เพื่อเก็บข้อมูล ในระบบนี้ข้อมูลที่นำเข้าสู่ Logstash เป็นรูปแบบเฉพาะที่ถูกกำหนดขึ้นมาเอง หากรูปแบบข้อมูลนำเข้ามีการเปลี่ยนแปลงหรือเป็นรูปแบบอื่น Logstash จะไม่สามารถเข้าใจและจัดเก็บลงใน Elasticsearch ได้

รูปที่ 2.4 ภาพรวมการทำงานของ Logstash

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
input{
    file{
    path => ["C:/Users/U6033063/Documents/File/filewriter4.log"]
    start_position => "beginning"
    }
}

filter{
    ruby {
        code => "
        fieldArray = event['message'].split(',')
        for fields in fieldArray
            field = fields.split('=')
            if field[0] != '@timestamp' && field[0] != 'Timestamp'
                    event[field[0]] = field[1].to_f
            else
                    event[field[0]] = field[1]
            end
        end
    "
    }
    mutate
    {
        remove_field => [ "message" ]
    }
    date
    {
        match => [ "Timestamp" , "yyyy/MM/dd HH:mm:ss" ]
    }

}

output{
    elasticsearch{
        host => localhost
        cluster => accesslog
        index => "filewriter5-%{+YYYY.MM.dd}"
    }
}
```

รูปที่ 2.5 ตัวอย่างการ Config file ในการใช้งาน Logstash

2.1.1.3 Kibana

Kibana เป็นเครื่องมือที่ใช้ในการ ค้นหา และจัดการกับข้อมูลที่ถูกเก็บไว้ใน Elasticsearch ได้ โดยการแสดงผลสามารถดึงข้อมูลมาแสดงผลได้แบบ Real-time ซึ่งสามารถเลือกรูปแบบการแสดงผลเป็นกราฟต่างๆได้ดังนี้

- Area Chart : กราฟแสดงพื้นที่ใต้กราฟ
- Data Table : การแสดงผลแบบตาราง
- Line Chart : กราฟเส้น
- Markdown Widget : การอธิบาย Dashboard
- Metric : การแสดงผลแบบตัวเลข
- Pie Chart : กราฟวงกลม
- Tile Map : แผนที่
- Vertical Bar Chart : กราฟแท่งแบบแนวตั้ง

ข้อดีของ Kibana

- รูปแบบการแสดงผลสวยงามและเข้าใจง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถนำแท็กข้อมูลมาประยุกต์และแสดงผลบนเว็บไซต์อื่นได้
- สามารถติดต่อกับ Elasticsearch ได้อย่างสะดวกเนื่องจากถูกสร้างมาเพื่อใช้งานร่วมกัน

ข้อเสียของ Kibana

- ภายในแท็กของกราฟข้อมูลที่ดึงมาแสดงผลบนเว็บต้องมีข้อมูลที่แน่นอน เช่น ขนาดต้องถูกกำหนดไว้ตั้งแต่ตอนสร้างแท็ก
- ไม่สะดวกต่อการใช้งานของผู้ใช้งานจริง เนื่องจากสามารถเข้าถึงข้อมูลบางอย่างที่ผู้ใช้ไม่ควรเข้าถึงได้

ในระบบนี้นำ Logstash มาใช้ในการพัฒนาระบบเนื่องจาก Kibana มีฟังก์ชันการสร้างกราฟที่ตรงกับเป้าหมายการใช้งานของผู้ใช้ ดังนี้

1. แสดงผลแบบ Real-time ได้
2. กราฟสามารถแสดงการเปรียบเทียบของข้อมูลได้
3. สามารถเลือกเวลาในการแสดงผลของกราฟ หรือช่วงเวลาล่าสุดได้
4. สามารถนำข้อมูลที่จำเป็นมาพัฒนาบนเว็บไซต์อื่นได้

รูปที่ 2.6 ส่วนแสดงผลของ Kibana

2.1.1.4 ภาษา HTML

ภาษา HTML เป็นภาษาที่ใช้ในการสร้างเว็บเพจ โดยโครงสร้างของภาษาที่ใช้ในการเขียนจะเป็นแท็กต่างๆ (Tag) ในการควบคุมและแสดงผลให้ออกมาบนหน้าเพจ

ข้อดีของ HTML

- สามารถทำงานบนเว็บบราวเซอร์ได้ทุกเบราเซอร์ และสามารถทำงานบนเครื่องคอมพิวเตอร์ทุกประเภทที่สนับสนุน HTML
- เป็นภาษาที่ง่ายต่อการเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสียของ HTML

- การใช้คำสั่งยังมีข้อจำกัด เนื่องจากต้องใช้คำสั่งที่มีอยู่เท่านั้น ไม่สามารถสร้างขึ้นมาเองได้

2.1.1.5 ภาษา Javascript

ภาษา Javascript เป็นภาษาที่ใช้ร่วมกับ HTML และเป็นภาษา Script ที่อยู่บนเว็บไซต์ต่างๆทำให้เว็บไซต์สามารถโต้ตอบกับผู้ใช้ได้และมีลูกเล่นทำให้เว็บเพจน่าสนใจมากขึ้น ซึ่ง Javascript เป็นสคริปฝั่งไคลเอนท์ จึงประมวลผลบนเว็บเบราเซอร์

2.1.1.6 ภาษา jQuery

ภาษา Jquery เป็นภาษาที่พัฒนามาจากภาษา Javascript ที่ใช้ในการทำงานร่วมกับ HTML เพื่อจัดการด้านไคลเอนท์(Client) หรือเบราเซอร์ (Broswer) เพื่อให้ตอบสนองต่อผู้ใช้โดยตรง

ข้อดีของ Jquery

- มีการทำงานที่เร็วกว่าภาษา javascript
- ง่ายในการเรียนรู้และใช้งาน
- มีปลั๊กอินให้เลือกใช้มาก
- รองรับการข้ามบราวเซอร์ได้ดี (Cross browser)

ข้อเสียของ Jquery

- การทำงานค่อนข้างกินทรัพยากรภายในเครื่อง
- การใช้งานยังมีปัญหาไม่รองรับบางเว็บบราวเซอร์

2.1.1.7 ภาษา PHP

PHP คือภาษาที่ใช้พัฒนาเว็บไซต์โดยเป็นลักษณะ Dynamic จึงทำให้ เว็บไซต์สามารถเปลี่ยนแปลงไปยังรูปแบบต่างๆได้ตามเงื่อนไขที่ระบุไว้ ซึ่ง PHP เป็นภาษา Script หมายความว่าสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจร่วมกับคำสั่งของ HTML ได้

ข้อดีของ PHP

- เป็น Script แบบ Server Side Script โดยการทำงานจะทำงานบนฝั่ง Server จะไม่ส่งผลต่อการทำงานของเครื่อง Client
- ความปลอดภัยอยู่ในระดับดี มีข้อผิดพลาดน้อย
- เป็นภาษาที่ได้รับความนิยม จึงง่ายต่อการศึกษาเรียนรู้

ข้อเสียของ PHP

- ขาด IDE ที่เป็นมาตรฐานกลาง ซึ่ง IDE บางตัวไม่สามารถ Debug ได้

ใบปริญญานิพนธ์ฉบับนี้นำ PHP มาใช้ในการพัฒนาระบบเนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PHP เป็นภาษาที่ง่ายต่อการเรียนรู้และพัฒนา

2.1.2 ส่วนของ LT Monitor Client

2.1.2.1 ภาษา JAVA

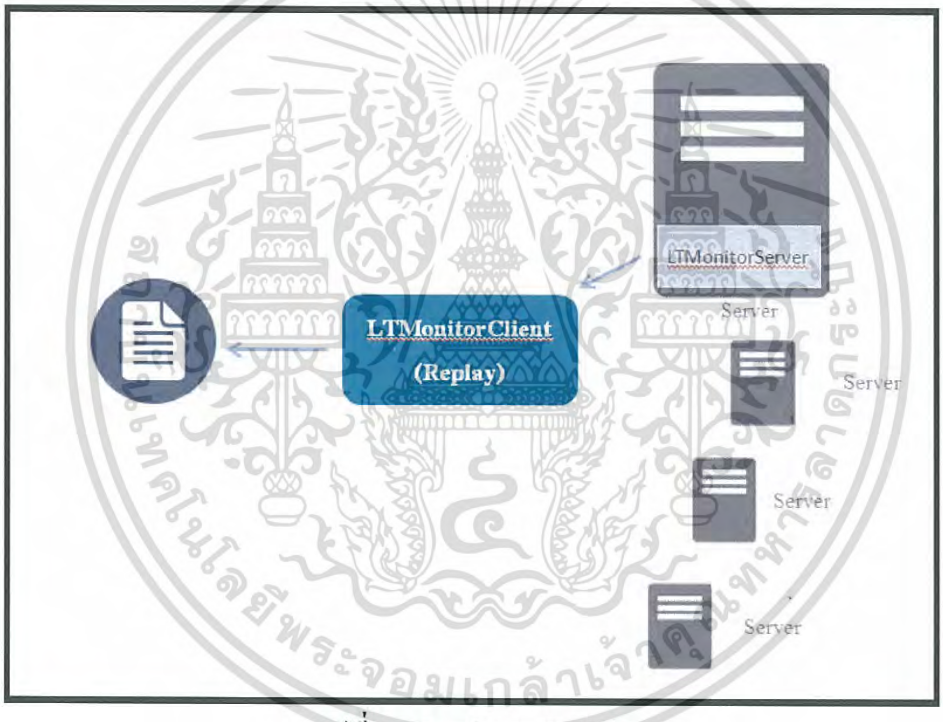
ภาษา JAVA เป็นภาษาเขียนโปรแกรมที่สามารถทำงานได้ในระบบปฏิบัติการที่หลากหลายโดยไม่ต้องแก้ไขโปรแกรมให้ทำงานได้หากต้องรันในระบบปฏิบัติการอื่น , สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object Oreinted Programming) และมี IDE , Library ต่างๆให้ใช้งานมากมาย โดยในระบบนี้ได้มีการนำ Library ที่ชื่อว่า JFreeChart มาใช้งาน เนื่องจากเป็นไลบรารี่ที่สามารถสร้างกราฟแบบ Real-time ได้ , เป็น Open Source ที่สามารถนำไปใช้ได้ฟรีและรองรับการสร้างรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 การศึกษาระบบงานเดิม

จากระบบงานเดิมเมื่อต้องการสังเกตการณ์ การใช้ทรัพยากรของแต่ละเซิฟเวอร์ จะมีโปรแกรมที่ชื่อว่า LT Monitor Server ทำหน้าที่เก็บรวบรวมข้อมูลการใช้ทรัพยากร (CPU, Memory) ของแต่ละเซิฟเวอร์นั้นๆ แล้วส่งข้อมูลมาให้โปรแกรมที่ชื่อว่า Replay (LT Monitor Client) ซึ่งทำหน้าที่นำข้อมูลที่ได้รับมาเขียนเป็นไฟล์ CSV โดยการสังเกตการณ์แบบเดิมจะดูจาก ข้อมูลในไฟล์ CSV ซึ่งมีจำนวนมากและยากต่อการสังเกต



รูปที่ 3.1 การทำงานแบบเก่า

3.2 ปัญหาที่พบในระบบปัจจุบัน

เนื่องจากการทำงานในลักษณะเดิม เมื่อผู้ใช้ต้องการสังเกตการณ์ข้อมูลการใช้ทรัพยากรในแต่ละเซิฟเวอร์ เพื่อทราบสาเหตุจากปัญหาต่างๆ เช่น มีการทำงานผิดปกติเกิดขึ้น , การใช้ทรัพยากรบางเซิฟเวอร์เกินความจำเป็น หรือเฝ้าสังเกตเพื่อให้การทำงานของเซิฟเวอร์มีประสิทธิภาพ ผู้ใช้จะต้องเข้าไปดูที่ไฟล์ CSV ซึ่งเป็นไฟล์ที่เก็บข้อมูลการใช้ CPU และ Memory ของทุกๆโปรแกรมในเซิฟเวอร์ที่มีการใช้งาน ทุกๆ 4 วินาที ดังนั้นข้อมูลจึงมีการอัพเดทเพิ่มขึ้นเรื่อยๆ ข้อมูลจึงมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนมากทำให้ยากต่อการสังเกตและเปรียบเทียบ และหากอยากดูข้อมูลในรูปแบบกราฟผู้ใช้จะสามารถดูได้จากการใช้ Excel

3.3 การวิเคราะห์ความต้องการระบบ (System Requirement Analysis)

3.3.1 ความต้องการที่เป็นหน้าที่หลักของระบบ (Functional Requirement)

- ระบบสามารถติดต่อกับเซิฟเวอร์โดยใช้ LT Monitor Client เป็นตัวทำการเชื่อมต่อได้
- ระบบสามารถแสดงรายชื่อโปรแกรมที่รันอยู่ในเซิฟเวอร์ขณะนั้นได้
- ระบบสามารถแสดงผลกราฟข้อมูลตามที่ผู้ใช้เลือกได้โดยใช้ LT Monitor Client (Java Application) และบนเว็บไซต์ได้
- ระบบสามารถแสดงผลกราฟในรูปแบบที่ผู้ใช้ต้องการได้บนเว็บไซต์ เช่น กราฟแบบเปรียบเทียบ , ดูหลายกราฟ
- ระบบสารมารถแสดงกราฟในรูปแบบเวลาที่ผู้ใช้ต้องการได้บนเว็บไซต์ เช่น ดูแบบ Realtime , ดูแบบย้อนหลัง
- เว็บไซต์ที่พัฒนาขึ้นสามารถดึงกราฟจาก Kibana ได้
- เว็บไซต์ Kibana สามารถดึงข้อมูลจากฐานข้อมูล Elasticsearch ได้
- ฐานข้อมูล Elasticsearch ต้องถูกอัพเดทจาก Logstash เมื่อมีการอัพเดทขอมูลในแต่ละครั้ง
- โปรแกรม LT Monitor Client สามารถเขียนไฟล์ในรูปแบบที่ต้องการได้ เมื่อมีข้อมูลเข้ามา

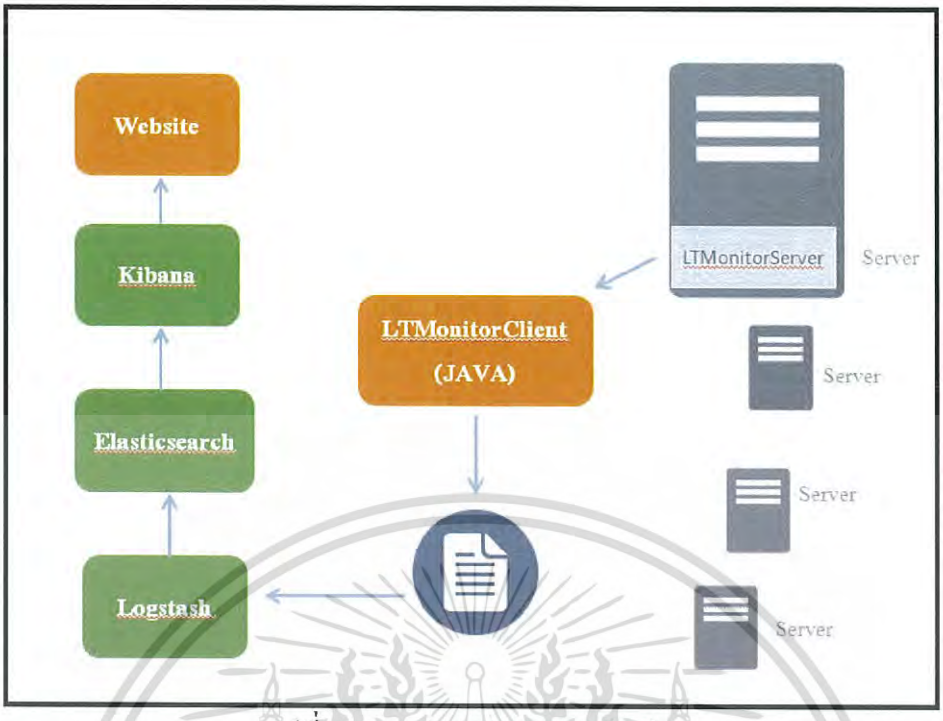
3.3.2 ความต้องการของระบบที่ไม่ใช่หน้าที่หลักของระบบ (Non-Functional Requirement)

- ระบบสามารถแสดงผลได้อย่างรวดเร็วในโปรแกรม LT Monitor Clientตามเวลาที่อัพเดทจริง
- ระบบสามารถแสดงผลได้สวยงาม
- โปรแกรม LT Monitor Client และ เว็บไซต์มีรูปแบบที่ใช้ง่าย, ง่ายต่อการใช้งาน

3.4 การออกแบบระบบใหม่

3.4.1 ภาพรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การทำงานของระบบใหม่

จากภาพแสดงภาพรวมการทำงานของระบบใหม่โดยสีส้มคือระบบที่พัฒนาเอง สีเขียวคือระบบที่เข้าไปจัดการเพื่อให้ทำงานได้ และมีการทำงานของระบบต่างๆดังนี้

- LT Monitor Server จะทำงานแบบเดิมโดยจะเป็นโปรแกรมที่อยู่ในเครื่องเซิฟเวอร์แต่ละเครื่องและส่งข้อมูลให้ LT Monitor Client เมื่อมีการเชื่อมต่อเกิดขึ้น
- LT Monitor Client เป็นแอพพลิเคชันที่พัฒนาจากภาษาจาว่า(JAVA) ซึ่งพัฒนาขึ้นมาใหม่ทั้งหมดโดยจะเชื่อมต่อกับ LT Monitor Server ,นำรับข้อมูลในรูปแบบ XMLมาเก็บเป็นไฟล์ที่มีการเปลี่ยนแปลงข้อมูลเป็นรูปแบบที่ Logstash สามารถนำไปเก็บใน Elasticsearch ได้ และสามารถแสดงผลกราฟเส้น, กราฟวงกลมของข้อมูลแบบ Real time ได้
- Logstash ทำหน้าที่นำไฟล์ข้อมูลที่ถูกสร้างจาก LT Monitor Client ไปเก็บใน Elasticsearch โดยจะต้องกำหนดรูปแบบของการอ่านข้อมูลและการเขียนข้อมูลที่จะถูกเก็บต้องอยู่ในรูปแบบที่ต้องการ
- Elasticsearch คือฐานข้อมูลประเภท NoSQL ที่สามารถเก็บข้อมูลที่เปลี่ยนแปลงหรือเพิ่มตลอดเวลาได้ (Dynamic Data) และ รองรับการเก็บข้อมูลที่มีจำนวนมากแต่ไม่ซับซ้อน
- Kibana คือเว็บแอพพลิเคชั่นที่สามารถแสดงผลข้อมูลที่อยู่ใน Elasticsearch ได้
- Website คือเว็บไซต์ที่พัฒนาขึ้นเพื่อแสดงผลกราฟทั้งแบบ Real-time และแบบย้อนหลังโดยนำเฉพาะข้อมูลกราฟมาจากเว็บไซต์ Kibana

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 แผนภาพยูสเคส (Use Case Diagram)

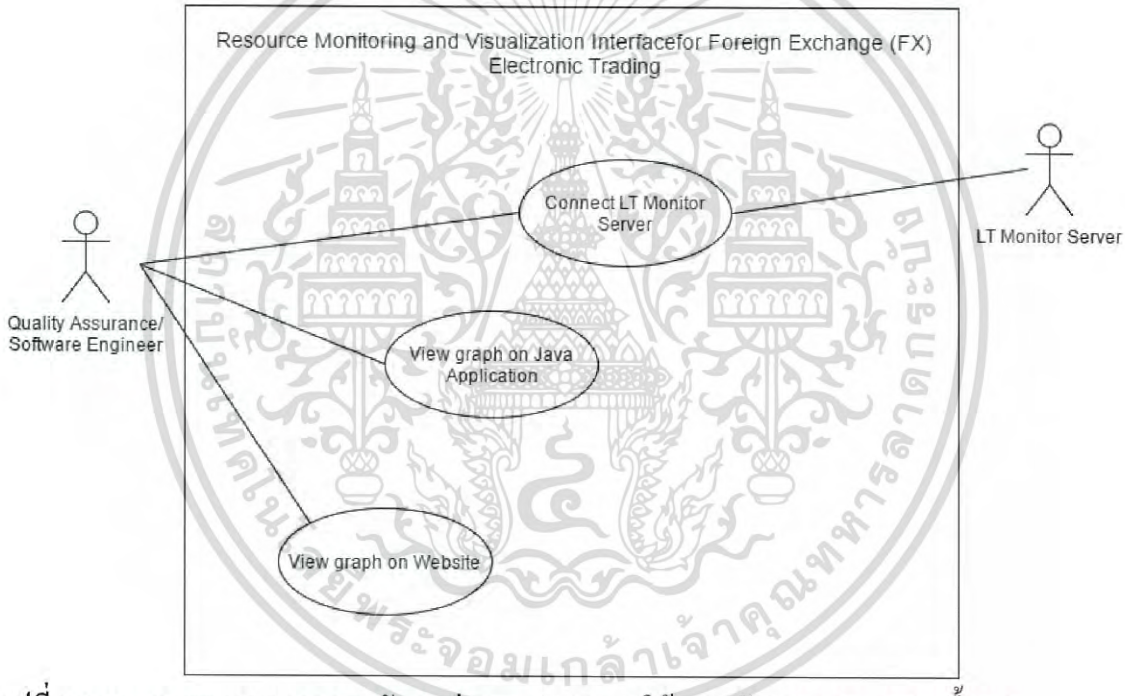
3.4.2.1 ผู้ที่เกี่ยวข้องในระบบ (Actor)

- Quality Assurance (QA)
- Software Engineer
- LT Monitor Server

3.4.2.2 องค์ประกอบของ Use Case

- Connect with LT Monitor Server
- View graph on JAVA application
- View graph on website

3.4.2.3 แผนภาพยูสเคส (Use Case Diagram)



รูปที่ 3.3 แผนภาพยูสเคสของการพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2.4 รายละเอียดยูสเคส (Use Case Description)

ตารางที่ 3.1 รายละเอียดยูสเคส Connect with LT Monitor Server

Use Case Name:	Connect with LT Monitor Server	ID: 1
Primary Actor:	Quality Assurance/Software Engineer/LT Monitor Server	
Description:	ส่วนการสร้างการเชื่อมต่อเพื่อรับส่งข้อมูลระหว่าง LT Monitor Client และ LT Monitor Server	
Pre-Condition:	ผู้ใช้เปิดโปรแกรม LT Monitor Client (JAVA application)	
Post-Condition:	ระบบได้รับข้อมูลจาก LT Monitor Server	
Basic Flow of Events	1. กรอก IP และ Port ของเซิฟเวอร์อย่างถูกต้อง 2. ผู้ใช้กดปุ่มเชื่อมต่อ Connect 3. ระบบรับการร้องขอการเชื่อมต่อและเชื่อมต่อกับ LT Monitor Server 4. ระบบอัพเดทข้อมูลในไฟล์ 5. ระบบแสดงรายชื่อโปรเซสที่มีในระบบ	
Alternative Flows	A1. เมื่อผู้ใช้กรอกข้อมูล IP หรือ Port ไม่ถูกต้อง ระบบจะไม่สามารถแสดงผลข้อมูลรายชื่อโปรเซสได้, ผู้ใช้ต้องกรอก IP, Port ใหม่ A2. เมื่อยังไม่มีการสร้างไฟล์สำหรับรองรับข้อมูลระบบจะทำการสร้างไฟล์ขึ้นมาใหม่	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 รายละเอียดยูสเคส Display graph on Java Application

Use Case Name:	Display graph on Java Application	ID: 2
Description:	ส่วนแสดงผลข้อมูลในรูปแบบกราฟบนเว็บไซต์ที่พัฒนาขึ้น	
Primary Actor:	Quality Assurance/Software Engineer	
Pre-Condition:	เมื่อสร้างการเชื่อมต่อเรียบร้อยแล้ว	
Post-Condition:	ระบบแสดงผลกราฟบนจาว่าแอพพลิเคชั่น	
Basic Flow of Events	1. ผู้ใช้เลือกโปรเซสที่ต้องการ 2. กดปุ่มแสดงผลกราฟ 3. ระบบ LT Monitor Server อัพเดทข้อมูลทุก 4 วินาที 4. LT Monitor Client เลือกโปรเซสมาจัดการเพื่อแสดงผลกราฟ 5. ระบบแสดงผลกราฟ	
Alternative Flow	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

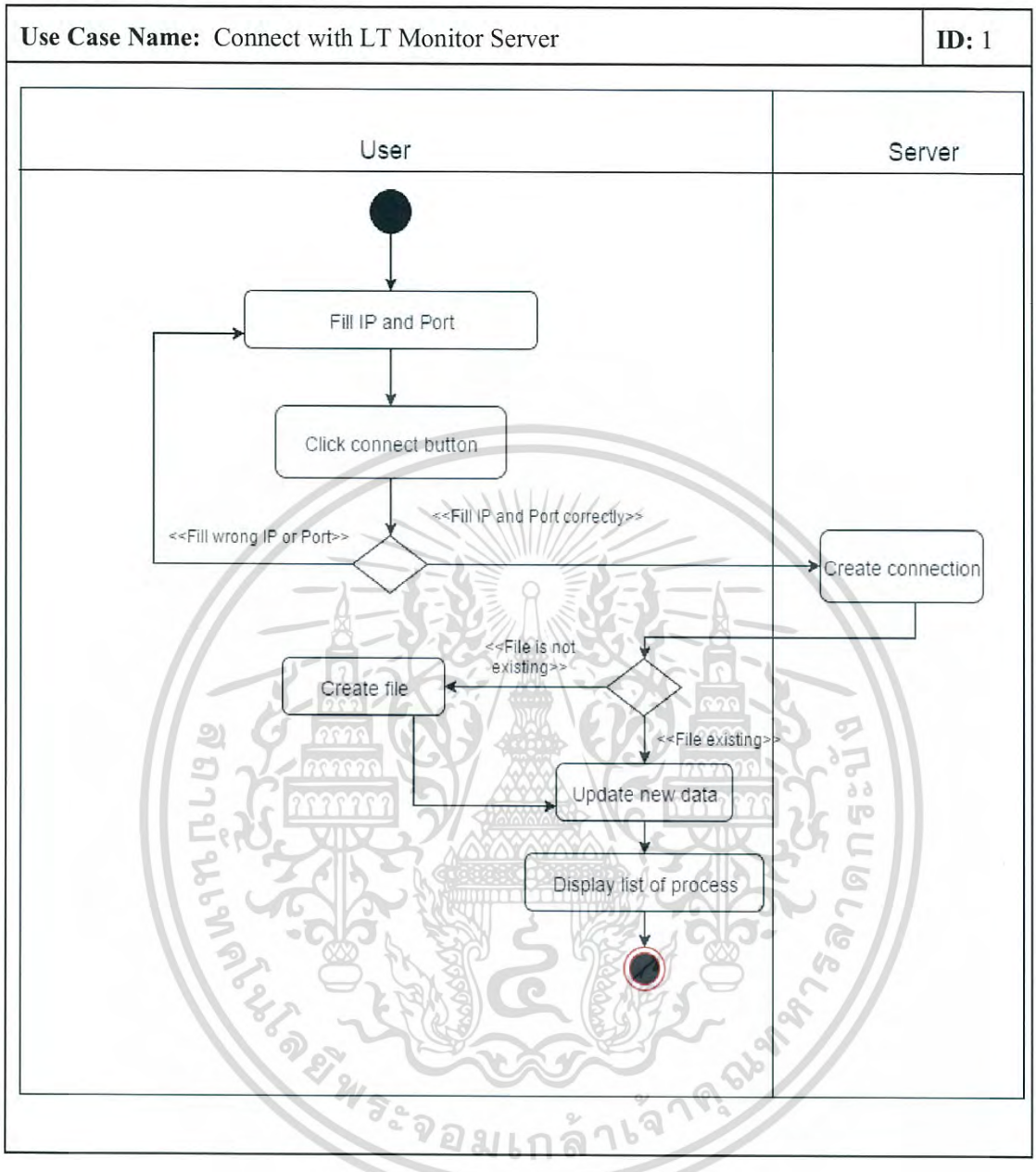
ตารางที่ **3.3** รายละเอียดยูสเคส Display graph on Website

Use Case Name:	Display graph on Website	ID: 3
Description:	แสดงผลข้อมูลในรูปแบบกราฟบนเว็บไซต์ที่พัฒนาขึ้น	
Primary Actor:	Quality Assurance/Software Engineer	
Pre-Condition:	เมื่อสร้างการเชื่อมต่อเรียบร้อยแล้ว	
Post-Condition:	ระบบแสดงผลกราฟบนเว็บไซต์	
Basic Flow of Events	1. ผู้ใช้เลือกโปรแกรมที่ต้องการดูข้อมูลภายในเซิฟเวอร์ , รูปแบบของการแสดงกราฟว่าจะให้แสดงผลแบบเปรียบเทียบ (Compare) หรือแสดงผลหลายกราฟในหน้าเดียว (Multiple Graph), รูปแบบของเวลาที่จะแสดงผลว่าจะแสดงข้อมูลย้อนหลัง (Historical Data) หรือ แบบRealtime และกรอกช่วงเวลาที่ต้องการดูข้อมูล 2. กดปุ่มแสดงผลกราฟ	
Alternative Flow	A1. เมื่อผู้ใช้ไม่เลือกโปรแกรมที่ต้องการดูข้อมูลจะมีหน้าต่างแจ้งเตือน A2. เมื่อผู้ใช้ไม่เลือกรูปแบบการแสดงผลข้อมูลจะมีหน้าต่างแจ้งเตือน A3. เมื่อผู้ใช้ไม่กรอกช่วงเวลาที่ต้องการแสดงผลข้อมูลจะมีหน้าต่างแจ้งเตือน	

146190

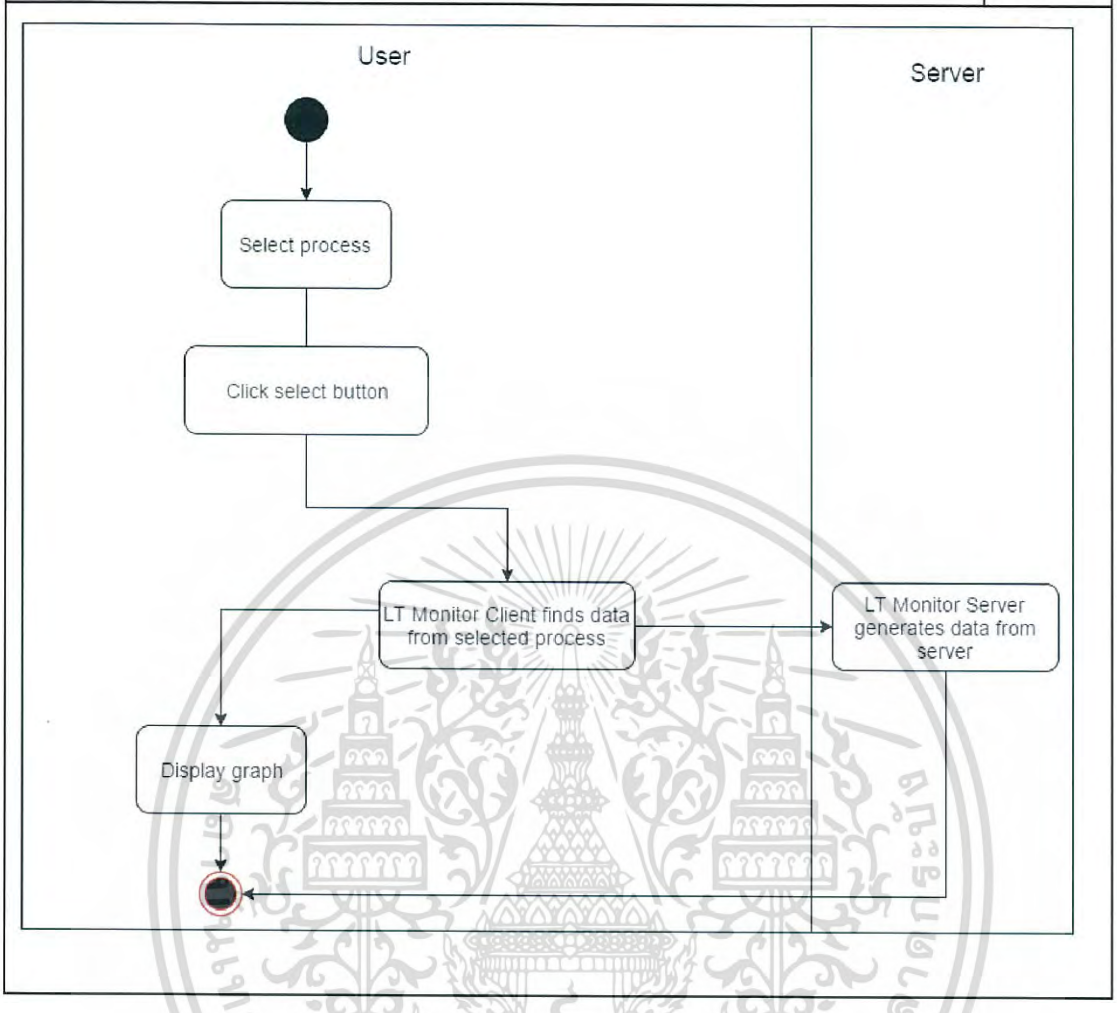
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 แผนภาพกิจกรรม (Activity Diagram)



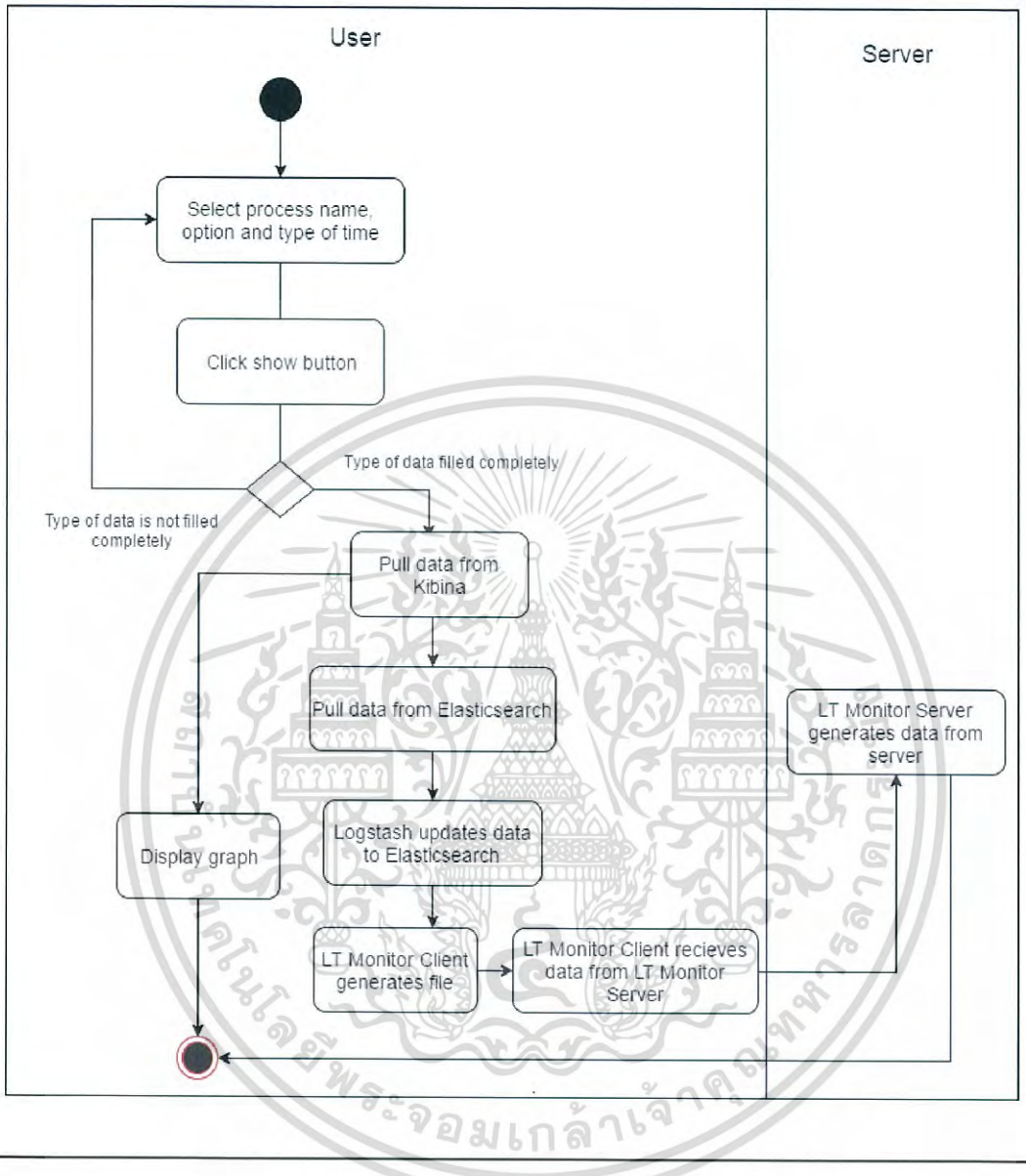
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Case Name: Display graph on Java Application	**ID:** 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

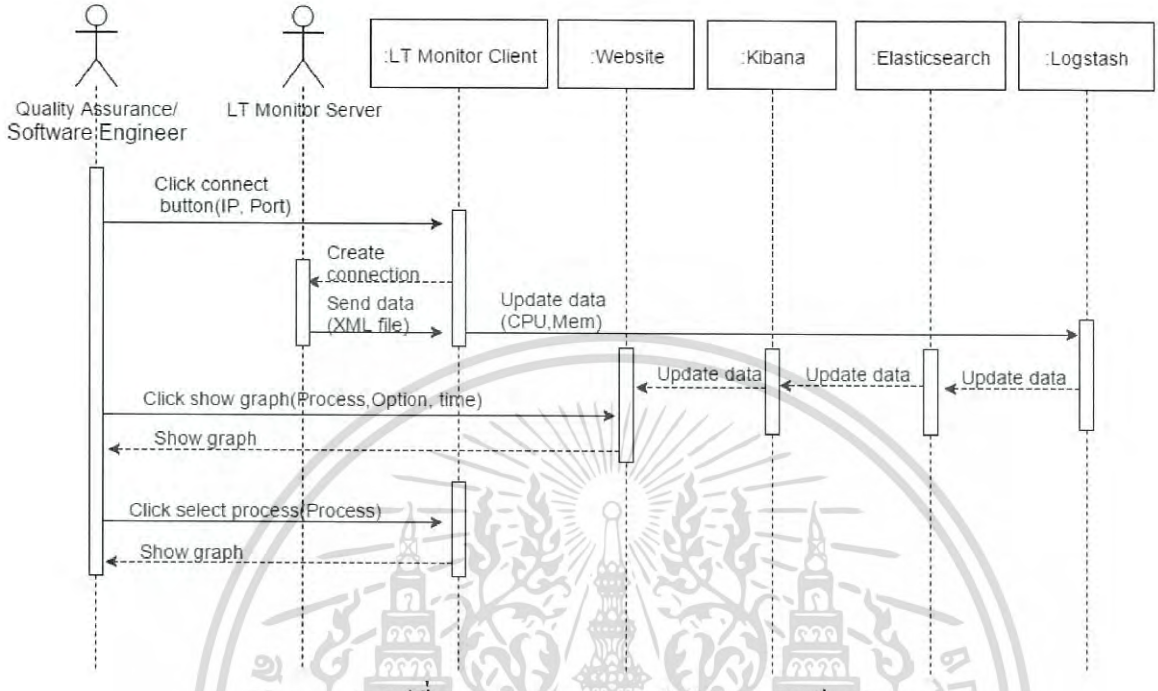
Use Case Name: Display graph on Website	**ID:** 3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 แผนภาพแสดงลำดับเหตุการณ์ (Sequence Diagram)

แผนภาพแสดงลำดับเหตุการณ์ต่างๆที่เกิดขึ้นในระบบ



รูปที่ 3.4 แผนภาพแสดงลำดับเหตุการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการออกแบบและผลการทำงานของระบบ

4.1 การออกแบบระบบ

4.1.1 การออกแบบส่วนต่างๆบนเว็บไซต์

4.1.1.1 การออกแบบส่วนประสานงานกับผู้ใช้ (GUI)

Connection Filter

- Select Program Dropdown : เลือกโปรแกรมที่ต้องการแสดงผล
- Option : เลือกรูปแบบของกราฟที่ต้องการโดย

  Compare Graph : เลือกกราฟที่แสดงผลข้อมูลหลายข้อมูลในกราฟเดียวกัน

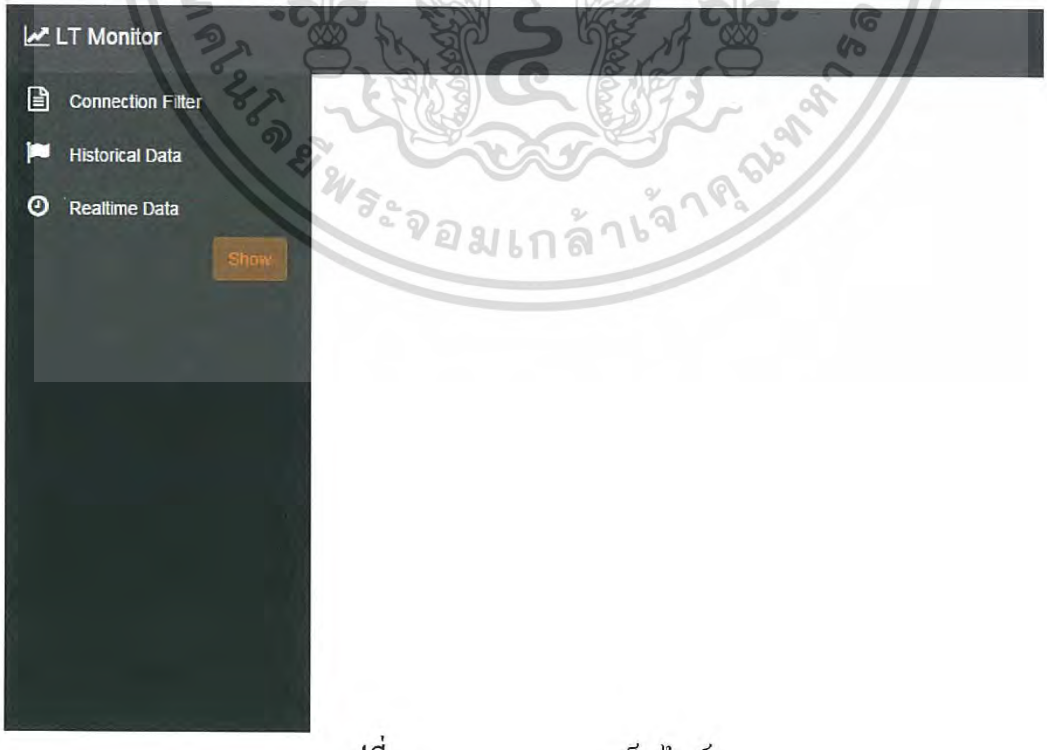
  Multiple Graph : เลือกแสดงผลหลายกราฟในหน้าเว็บเพจเดียว

Historical Data

- From : ช่องกรอกเวลาเริ่มต้นของการแสดงผลกราฟ
- To : ช่องกรอกเวลาสุดท้ายของการแสดงผลกราฟ
- Date : ช่องเลือกวันที่ต้องการ (Date Picker)

Real Time Data

- Time : เลือกเวลาล่าสุดจนถึงปัจจุบันที่ต้องการแสดงผล



รูปที่ 4.1 การแสดงผลบนเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.2 การออกแบบส่วนการทำงานของซอฟต์แวร์

การออกแบบส่วนการทำงานจะแบ่งออกเป็นส่วนเว็บไซต์ซึ่งพัฒนาขึ้นมาเองและส่วนจัดการข้อมูลซึ่งนำการทำงานของ ELK มาใช้งาน

การพัฒนาโปรแกรม (Coding)

โดยการพัฒนาบนเว็บไซต์ใช้ภาษา PHP และแบ่งการทำงานหลักๆได้ดังนี้

getProgramList.php

เป็นส่วนที่รับข้อมูลที่มีเฉพาะชื่อของโปรแกรมในเซิฟเวอร์โดยนำข้อมูลจากเว็บที่เก็บรายละเอียดของหัวคอลัมภายใน Elasticsearch มาประมวลผลเพื่อให้ได้ข้อมูลเฉพาะชื่อของหัวคอลัม

displayGraph.php

เป็นส่วนที่ทำหน้าที่รับข้อมูลที่จำเป็นจากการเลือกของผู้ใช้ผ่านหน้าเว็บไซต์เพื่อนำไปดึงกราฟมาแสดงผลตามข้อมูลนั้น โดยใช้แท็ก iframe ของ Kibana

index_graph.php

เป็นส่วนที่แสดงผลและส่วนควบคุมการทำงาน โดยจะรวบรวมส่วนการแสดงผลไว้ทั้งหมด

4.1.1.3 การออกแบบส่วนการทำงานของการจัดการข้อมูล

- การทำงานของ Logstash

1. เตรียมไฟล์ข้อมูลที่จะทำการเก็บข้อมูล ในที่นี้ใช้ไฟล์ 20151116.log ซึ่งเป็นไฟล์ที่ถูกสร้างจาก LT Monitor Client (JAVA) ซึ่งโปรแกรม LT Monitor Client แปลงให้อยู่ในรูปแบบที่ Logstash อ่านได้แล้ว

```
Timestamp,GID.admin[9314] CPU(%),GID.admin[9314] MEM(MB),GID.admin[10858] CPU(%),GID.admin[10858]
2015/10/13 09:07:00,0.16,125.91016,0.09,120.68359,0.11,124.01563,0.08,110.05078,0.07,125.06250,0.
2015/10/13 09:07:04,0.12,125.91016,0.06,120.68359,0.06,124.01563,0.08,110.05078,0.16,125.06250,0.
2015/10/13 09:07:08,0.13,125.91016,0.07,120.68359,0.08,124.01563,0.08,110.05078,0.09,125.06250,0.
2015/10/13 09:07:12,0.16,125.91016,0.11,120.68359,0.06,124.01563,0.07,110.05078,0.10,125.06250,0.
2015/10/13 09:07:16,0.12,125.91016,0.07,120.68359,0.09,124.01563,0.08,110.05078,0.07,125.06250,0.
2015/10/13 09:07:20,0.14,125.91016,0.09,120.68359,0.08,124.01563,0.07,110.05078,0.07,125.06250,0.
2015/10/13 09:07:24,0.13,125.91016,0.07,120.68359,0.06,124.01563,0.07,110.05078,0.10,125.06250,0.
2015/10/13 09:07:28,0.13,125.91016,0.09,120.68359,0.08,124.01563,0.07,110.05078,0.08,125.06250,0.
2015/10/13 09:07:32,0.16,125.91016,0.10,120.68359,0.06,124.01563,0.08,110.05078,0.09,125.06250,0.
2015/10/13 09:07:37,0.12,125.91016,0.07,120.68359,0.10,124.01563,0.07,110.05078,0.07,125.06250,0.
2015/10/13 09:07:41,0.14,125.91016,0.09,120.68359,0.08,124.01563,0.07,110.05078,0.07,125.06250,0.
2015/10/13 09:07:45,0.13,125.91016,0.06,120.68359,0.06,124.01563,0.08,110.05078,0.10,125.06250,0.
2015/10/13 09:07:49,0.13,125.91016,0.12,120.68359,0.08,124.01563,0.07,110.05078,0.08,125.06250,0.
```

รูปที่ 4.2 ไฟล์ข้อมูลจากเซิฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สร้างไฟล์ที่ใช้ในการจัดการ (Config) ข้อมูลขึ้นมา โดยมีรูปแบบดังนี้

Input

ภายใน input ใส่ข้อมูลที่จะถูกนำมาเก็บไว้ใน Elasticsearch โดยมีการใส่ Path ของข้อมูลที่ถูกเก็บไว้และให้เริ่มต้นอ่านจากตำแหน่งแรก

```
input{
        file{
        path =>
["C:/project/LTMonitorClient/_name.csv"]
        start_position => "beginning"
        }
}
```

รูปที่ 4.3 ไฟล์การรับข้อมูลเข้า Logstash

Filter

ภายใน filter จะมีการกำหนดรูปแบบการอ่านข้อมูลเพื่อให้ Elasticsearch เข้าใจและกำหนดรูปแบบของเวลาให้ตรงกับรูปแบบของเวลาในไฟล์ข้อมูลที่จะเก็บ

```
filter{
        ruby {          ...
        }
        mutate{         ...
        }
        date {
                match => [ "Timestamp" , "yyyy/MM/dd
        HH:mm:ss" ]
        }
}
```

รูปที่ 4.4 ไฟล์ตัวอย่างการแปลงข้อมูลเข้า Logstash

Output

ภายใน output กำหนดว่าจะให้ผลลัพธ์ของข้อมูลไปถูกเก็บไว้ที่ไหน ในโปรเจคนี้จะให้เก็บไว้ที่ Elasticsearch จึงกำหนดให้ผลลัพธ์ปลายทางเป็น Elasticsearch และกำหนด host , cluster , index ทุกครั้ง

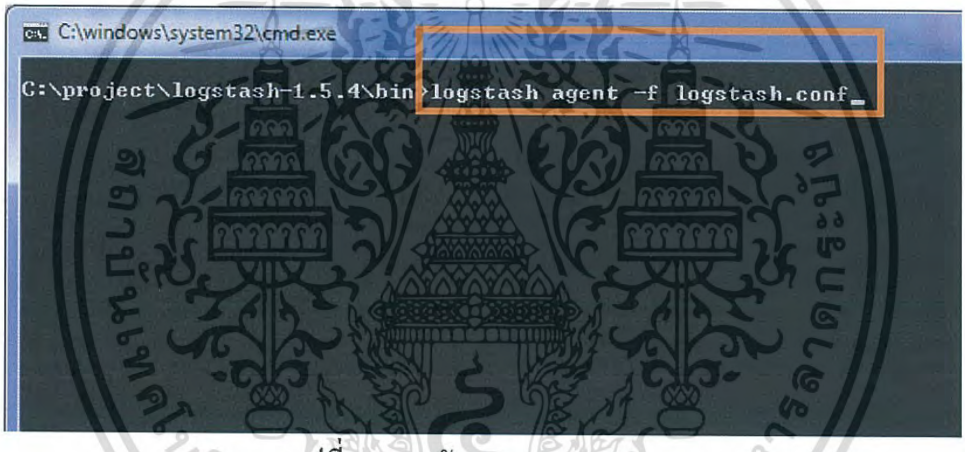
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
output{
        elasticsearch{
                host => localhost
                cluster => accesslog
                index => "test_with_perl_128-%{+YYYY.MM.dd}"
        }

}
```

รูปที่ 4.5 ไฟล์การกำหนดปลายทางข้อมูลออกจาก Logstash

3. เมื่อกำหนดรูปแบบการเก็บข้อมูลเสร็จแล้วจึงรัน คำสั่ง logstash agent –f filename.conf เพื่อทำการเก็บข้อมูลดังกล่าวใน Elasticsearch และเปิดหน้าต่างในขณะที่รันอยู่เสมอ

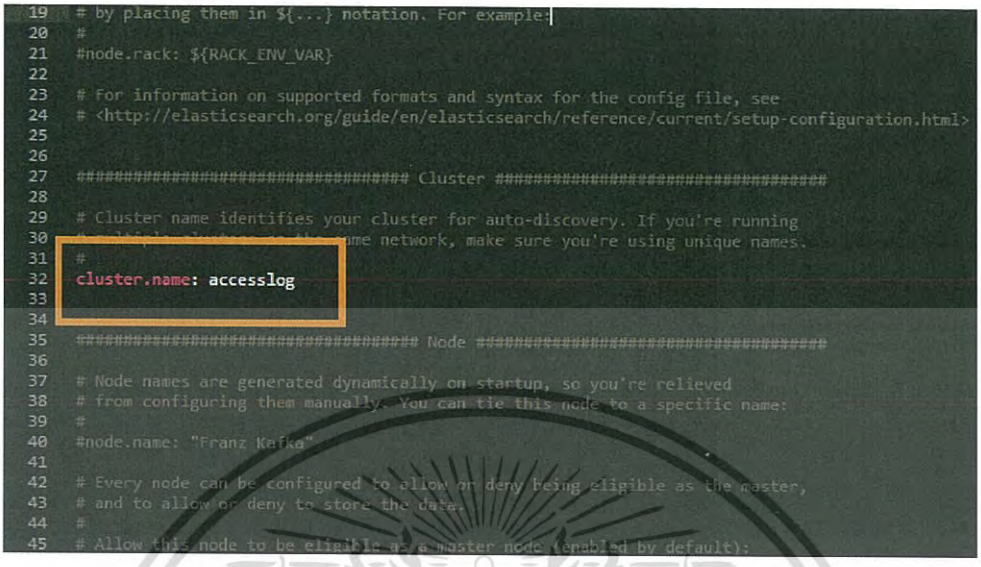


รูปที่ 4.6 การรัน logstash.conf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

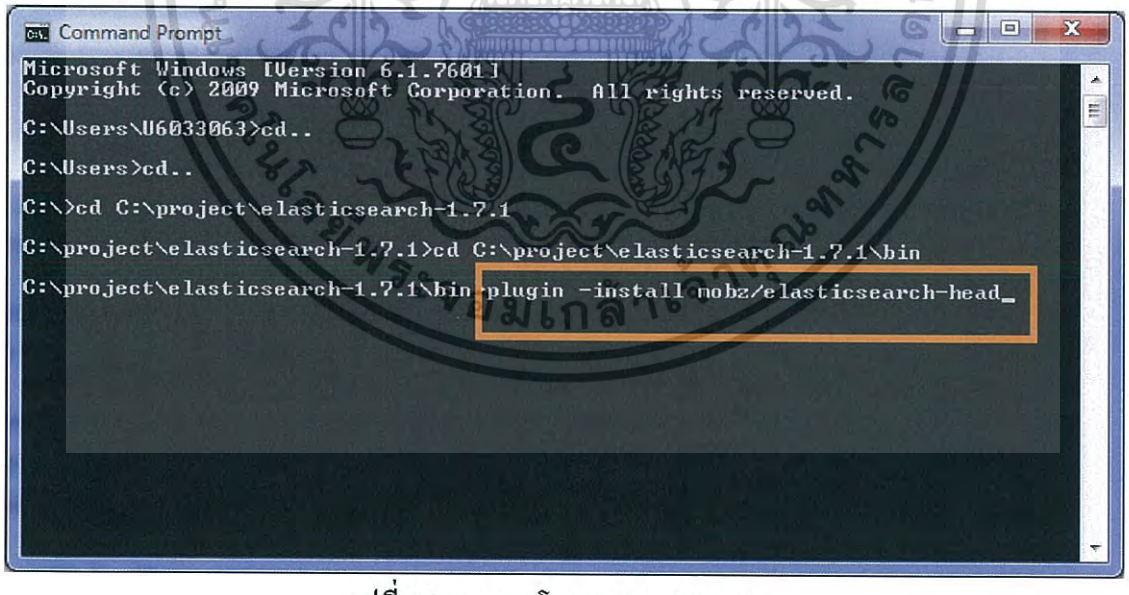
- การทำงานของ Elasticsearch

1. Config Cluster name ภายในไฟล์ elasticsearch.yml



รูปที่ 4.7 การ Config Cluster name

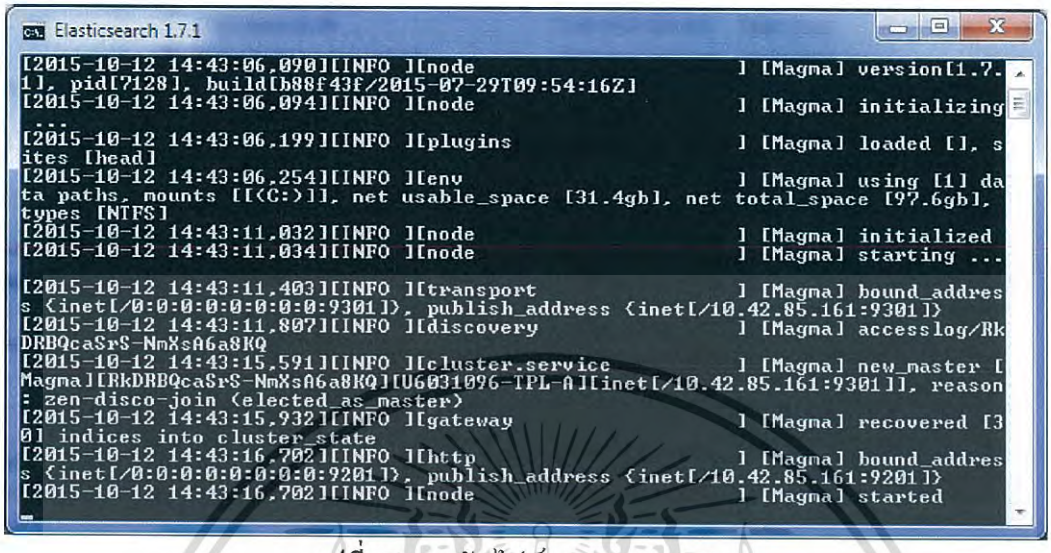
2. เพื่อที่จะสามารถดูข้อมูลที่ถูกเก็บใน Elasticsearch ได้ จึงมีการติดตั้งปลั๊กอินเพิ่ม โดยการติดตั้ง Plug in head ซึ่งเป็นปลั๊กอินเสริมของ Elasticsearch



รูปที่ 4.8 การดาวโหลด Plug in head

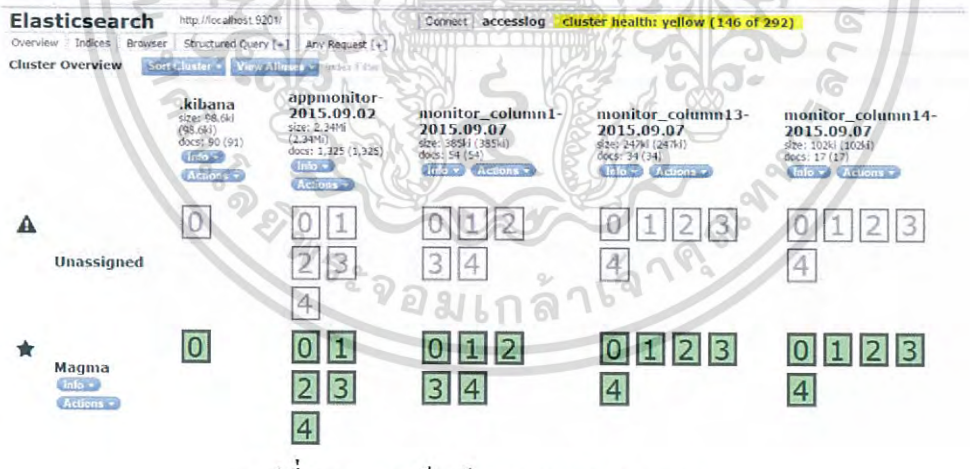
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการรัน Elasticsearch.bat เพื่อให้สามารถเปิด Plug in head ได้ และการรัน Elasticsearchต้องทำการรันทุกครั้ง, ตลอดเวลาที่ Logstash ทำการเก็บข้อมูลมาให้ elasticsearch



รูปที่ **4.9** การรันไฟล์ Elasticserach.bat

4. แสดงข้อมูลที่มีใน Elasticsearch ผ่าน http://localhost:92xx/_plugin/head/ โดยสามารถดู Index ทั้งหมดที่ถูกใส่เข้ามาใน Elasticsearch



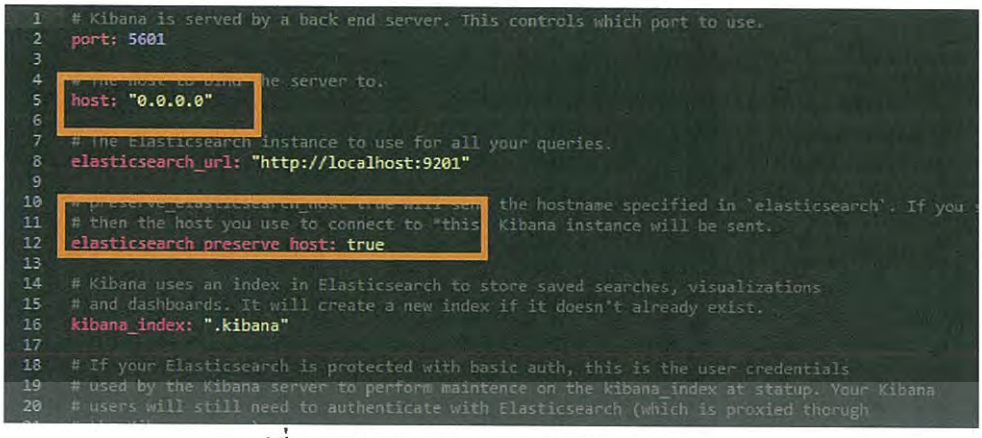
รูปที่ **4.10** การเก็บข้อมูลบน Elasticsearch

- การทำงานของ Kibana

1. ภายในไฟล์ kibana.yml มีการกำหนดดังนี้

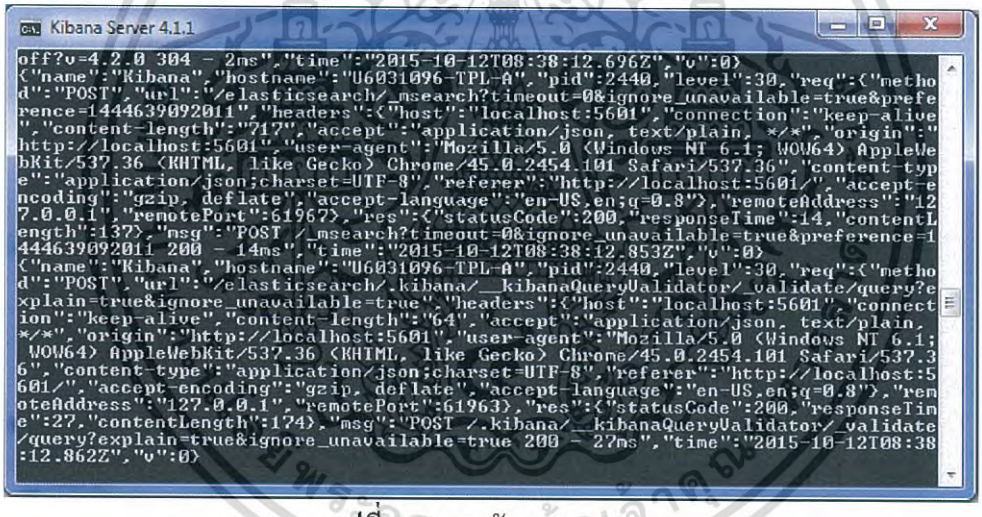
- Port : กำหนด Port ที่ต้องการให้ Kibana ใช้
- Elasticsearch_url : กำหนดเส้นทางของ Elasticsearch ให้ Kibana

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 การ Config การทำงานของ Kibana

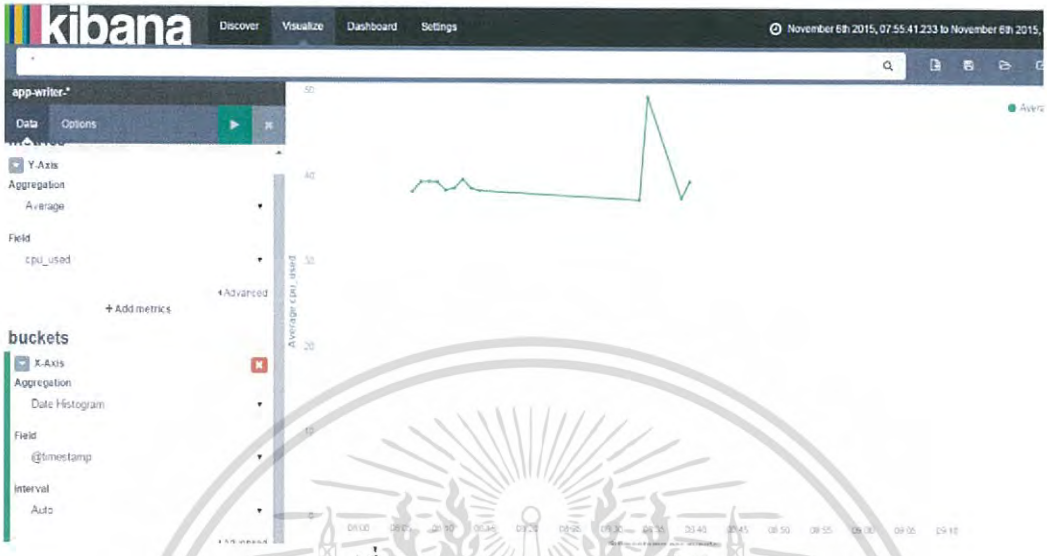
2. ทำการรัน Kibana.bat เพื่อให้สามารถเปิด Website Kibana ได้ และการรัน Kibana ต้องทำการรันทุกครั้งที่ต้องการเรียกใช้ข้อมูลจาก Kibana



รูปที่ 4.12 การรัน Kibana.bat

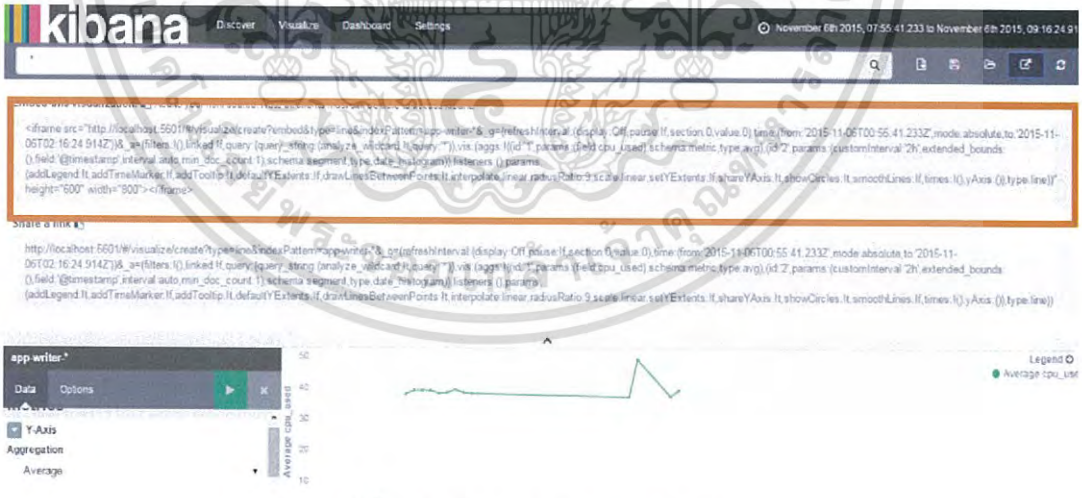
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เปิดเว็บไซต์ Kibana เพื่อดูผลการแสดงข้อมูล โดยสามารถดูข้อมูลได้จากการเพิ่มชื่อ Index ของ Elasticsearch



รูปที่ 4.13 การแสดงผลกราฟเส้นใน Kibana

4. เมื่อสามารถแสดงผลข้อมูลบนเว็บไซต์ Kibana ได้แล้ว สามารถนำ Embed Visualize (Tag iframe) ไปใช้งานต่อได้



รูปที่ 4.14 Embed Visualize Kibana

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การออกแบบส่วนต่างๆบน Java Application

4.1.2.1 การออกแบบส่วนประสานงานกับผู้ใช้ (GUI)

- ส่วนการเชื่อมต่อ (Connection) แบ่งเป็น

  IP = ช่องกรอกไอพีของเครื่องเซิฟเวอร์ที่ต้องการ

  Port = ช่องกรอกพอร์ทของเครื่องเซิฟเวอร์ที่ต้องการ

  Connect = ปุ่มเริ่มการเชื่อมต่อกับเซิฟเวอร์เครื่องดังกล่าว

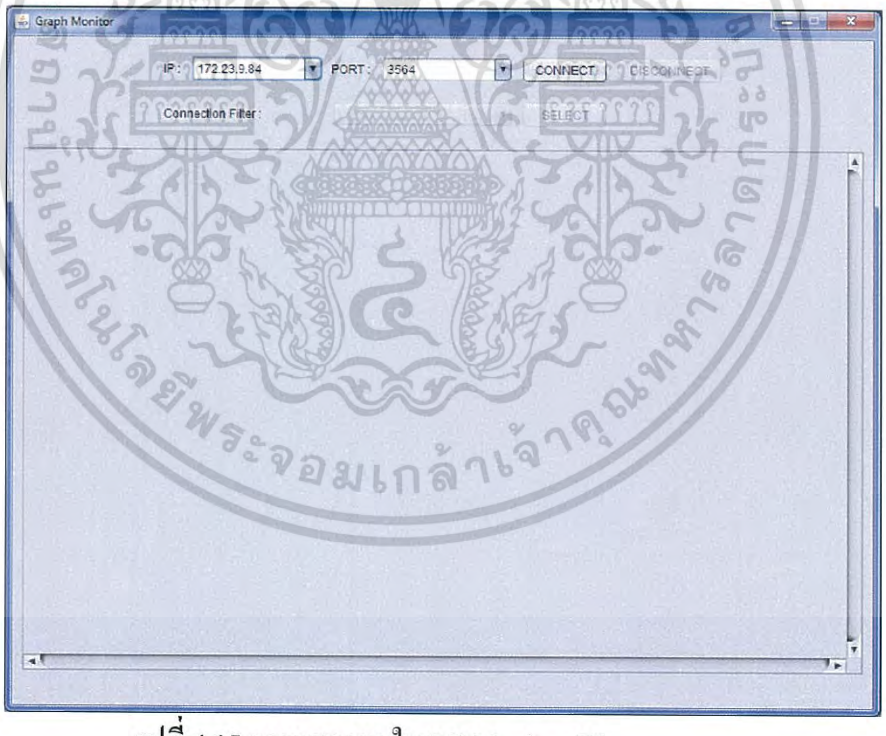
  Disconnect = ปุ่มยกเลิกการเชื่อมต่อกับเซิฟเวอร์เครื่องดังกล่าว

  ในการกรอก IP, Port หากเคยกรอกแล้ว โปรแกรมจะสามารถจำค่าได้ ทำให้ผู้ใช้ไม่ต้องกรอกซ้ำ

- ส่วนการแสดงผล

  Connection Filter ComboBox = เป็นรายชื่อโปรแกรมที่มีการทำงานอยู่บนเซิฟเวอร์เครื่องดังกล่าว

  Select = แสดงกราฟและข้อมูลแบบ Real time ตามโปรแกรมที่ได้เลือกไว้

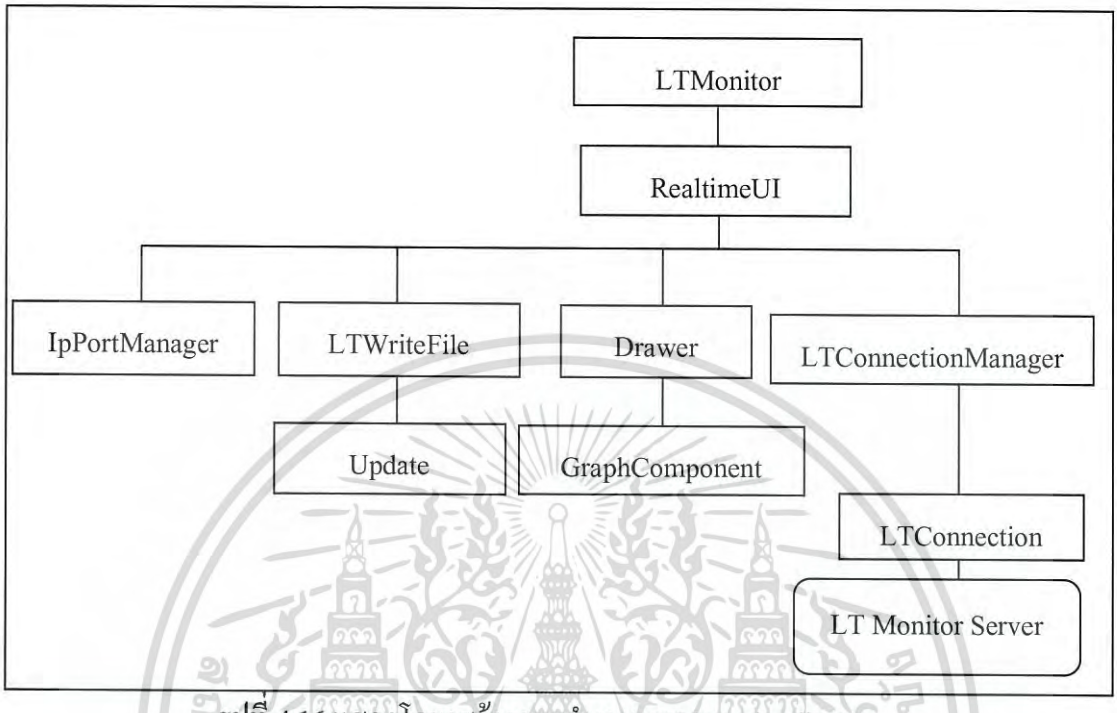


รูปที่ 4.15 การแสดงผลใน LT Monitor Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.2 การออกแบบส่วนการทำงานของซอฟต์แวร์

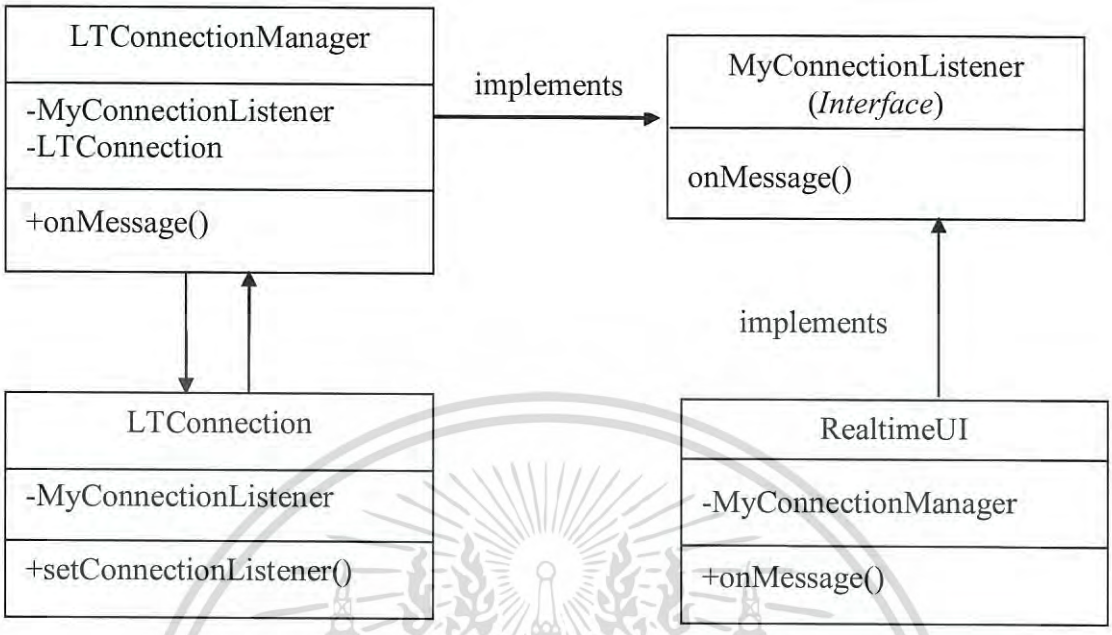
โดยการพัฒนา Java Application มีโครงสร้างการทำงานดังนี้



รูปที่ 4.16 แสดงโครงสร้างการทำงานของ Java Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **หลักการออกแบบการส่งข้อมูลจากเซิฟเวอร์**



รูปที่ 4.17 แผนภาพแสดงการทำงานหลักการส่งข้อมูล

- MyConnectionListener เป็น Interface ที่มี Method onMessage
- LTConnectionManager implements MyConnectionListener เพื่อเรียกใช้ onMessage และเก็บ LTConnection ไว้เพื่อจะรู้ว่าจะฟังข้อมูลจากใคร
- LTConnection เก็บตัวแปร LTConnectionManager ในรูปแบบ MyConnectionListener
- RealtimeUI implements MyConnectionManager เพื่อรับข้อมูลที่ถูกส่งมาผ่าน onMessage และส่ง IP, Port ไปให้ LTConnectionManager ในรูปแบบ MyConnectionListener

- **การทำงานของการส่งข้อมูล**

เมื่อเริ่มต้นการทำงาน RealtimeUI implements MyConnectionListener และจะส่งข้อมูล (IP, Port) และ RealtimeUI(this) ในรูปแบบMyConnectionListener ไปยัง LTConnectionManager ซึ่งส่วนที่รับข้อมูลนี้จึงได้ข้อมูลของ RealtimeUI เฉพาะส่วนของ MyConnectionListener เท่านั้น เนื่องจากตอนที่ LTConnectionManager รับค่าของ setConnectionListener เป็นชนิด MyConnectionListener

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RealtimeUI

```
ltconnectionManage = new LTConnectionManager(sendIP, sendPORT);
ltconnectionManage.setConnListener(this);
ltconnectionManage.connect();
```

รูปที่ 4.18 การส่งข้อมูลจาก RealtimeUI

ข้อมูลที่ถูก setConnectionListener จาก RealtimeUI จะมีแค่ชนิด MyConnectionListener

LTConnectionManager

```
public void setConnListener(MyConnectionListener myConn){
    this.myConnListener = myConn;

}
@Override
public void onMessage(String str) {
    // TODO Auto-generated method stub
    myConnListener.onMessage(str);
}
```

รูปที่ 4.19 การส่งข้อมูลจาก LTConnectionManager ผ่าน onMessage

เมื่อ LTConnectionManager ได้รับข้อมูลต่างๆที่ถูกส่งมาจาก RealtimeUI จึงนำข้อมูล IP, Port ส่งไปยัง LTConnection เพื่อสร้าง Connection ภายใน Method connect()

LTConnectionManager

```
m_oUser = new LTConnection(HOST, PORT);
m_oUser.setConnectionListener(this);
```

รูปที่ 4.20 การส่งข้อมูลจาก LTConnection Manager ไปยัง LTConnection

เมื่อ LTConnection ได้รับข้อมูลต่างๆจาก LTConnectionManager แล้ว หลังจากนำข้อมูลมาสร้าง Connection เสร็จเรียบร้อย ข้อมูล (Message) ของเซิฟเวอร์จะถูกส่งกลับมาผ่าน Method onMessage ของ Interface IDataListener ซึ่งจะส่งข้อมูลกลับมาทุก 4 วินาที ดังนั้นจะส่งข้อมูลนี้กลับไปยัง RealtimeUI เพื่อแสดงผลข้อมูลที่ถูกส่งกลับมาผ่าน Interface ที่ทำหน้าที่เป็นผู้ฟัง (Listener) ที่ชื่อว่า MyConnectionListener โดยส่งข้อมูลดังกล่าวผ่าน Method onMessage ของ MyConnectionListener

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LTConnection

```
public LTConnection(String host, int port)
{
    m_commsManager = new CommsManager();
    m_host = host;
    m_iport = port;
    // createConnectionHandle();

}

public void setConnectionListener(MyConnectionListener myConnectionListener)
{
    this.myConnectionListener = myConnectionListener;
}
```

รูปที่ 4.21 การรับข้อมูลจาก LTConnectionManager

```
@Override
public void onMessage(IConnectionHandle ch, Object data, int iType)
{
    // TODO Auto-generated method stub
    //Send Back
    myConnectionListener.onMessage((String)data);

    System.out.println(data);
}
```

รูปที่ 4.22 การส่งข้อมูลจาก LTConnection

เมื่อได้ข้อมูลกลับมาจาก LTConnection ข้อมูลจะถูกส่งต่อมาเรื่อยๆผ่าน onMessage ของ MyConnectionListener เมื่อ RealtimeUI ได้ ข้อมูล (Message) แล้วจึงส่งให้ LTWriteFile เพื่อแปลงข้อมูลและประมวลผลต่อไป

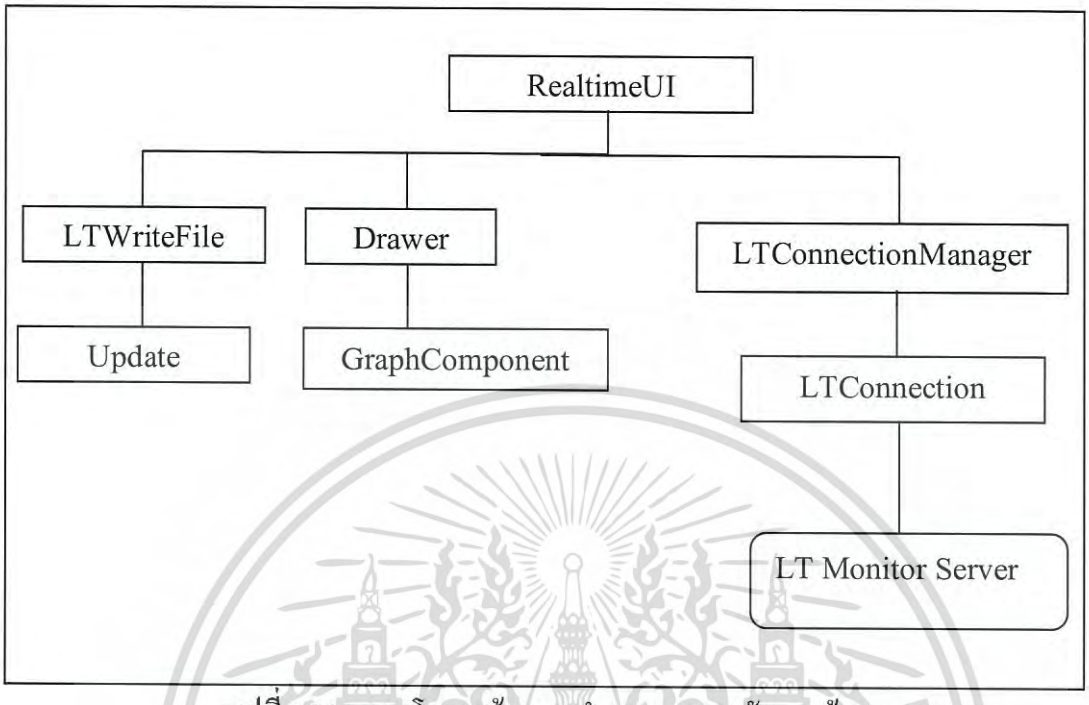
RealtimeUI

```
@Override
public void onMessage(String message) {
    // TODO Auto-generated method stub
    // ltwritefile = new LTWriteFile();
    try {
        ltwritefile.onWrite(message);
```

รูปที่ 4.23 การรับข้อมูลผ่าน onMessage และนำไปประมวลผลต่อ

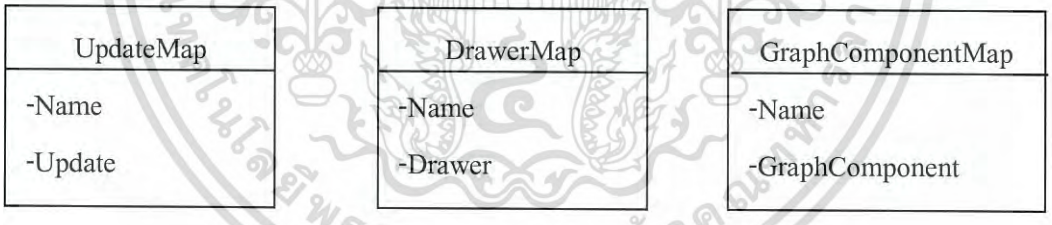
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การออกแบบการอัพเดทและเก็บข้อมูล



รูปที่ 4.24 แสดงโครงสร้างการทำงานของการอัพเดทข้อมูล

โดยการเก็บข้อมูลจะเก็บเป็น Hashmap Object ที่อ้างอิงจากชื่อของโปรเซส



รูปที่ 4.25 ออบเจ็คการเก็บข้อมูล

จากหลักการส่งข้อมูลที่ได้รับจากเซิฟเวอร์ ดังนั้นข้อมูลต่างๆจะถูกส่งมาที่ onMessage ของ RealtimeUI ทุก 4 วินาที แต่ Message ที่ถูกส่งมายังเป็นรูปแบบ XML ซึ่งยังนำไปใช้งานไม่ได้ จึงนำไปแปลงให้อยู่ในรูปแบบที่ต้องการโดย LTWriteFile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RealtimeUI

```
@Override
public void onMessage(String message) {
    // TODO Auto-generated method stub
    try {
        ltwritefile.onWrite(message);
```

รูปที่ 4.26 การรับข้อมูลผ่าน onMessage และส่งข้อมูลไป LTWriteFile

LTWriteFile ทำหน้าที่นำข้อมูลที่ได้จากการเชื่อมต่อกับเซิฟเวอร์มาแปลงและเขียนเป็นไฟล์ ขณะเดียวกันก็ส่งค่าต่างๆที่เป็นข้อมูลที่จำเป็นให้กับ Update (Set Update Value) และเก็บ ชื่อของโปรเซส(โปรแกรมที่รันอยู่บนเซิฟเวอร์) และ Object ของ Update แต่ละโปรเซส ลงใน HashMap ชื่อ updateMap ซึ่งภายในคลาส Update จะทำหน้าที่ตั้งค่าและรับค่าต่างๆ ผ่าน Method get, set

LTWriteFile

```
u = new Update();
u.setStrProgramName(getDiskName);
u.setfDiskValue(getDiskUsed);
u.setfDiskPhysical(getDiskPhysical);
u.setTimeStamp(time);
RealtimeUI.updateMap.put(u.getStrProgramName(), u);
listname.add(getDiskName);
```

รูปที่ 4.27 การสร้างออบเจ็คของ Update และเก็บลงใน HashMap updateMap

เมื่อตั้งค่าข้อมูลของ Update เรียบร้อยแล้ว ในคลาส Drawer จะนำค่าจาก Update มาตั้งค่าในออบเจ็ค GraphComponent (Set GraphComponent Value) ซึ่งภายในจะมีการตั้งค่าต่างๆที่จะปรากฏใน RealtimeUI

Drawer

```
public void updateGraph(Update u)
{
    if(!strProgramName.equals("DISK") && !strProgramName.equals("SYSTEM")){
        numCpu = u.getfCpuValue();
        numMem = u.getfMemValue();
        strTimeStamp = u.getTimeStamp();

        GraphComponent graphComp = RealtimeUI.graphComponentMap.get(strProgramName);
        graphComp.setTimeStamp(strTimeStamp);
        graphComp.setTs_cpu(numCpu);
        graphComp.setTs_mem(numMem);
        graphComp.setLbStatusCpu(numCpu);
        graphComp.setLbStatusMem(numMem);
        graphComp.setLbTabStCpu(numCpu);
        graphComp.setLbTabStMem(numMem);
    }
```

รูปที่ 4.28 ตัวอย่างการตั้งค่าข้อมูลในออบเจ็คของ GraphComponent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานเริ่มต้นภายใน RealtimeUI โดยการเรียก createDrawer ซึ่งจะสร้าง Drawer, GraphComonent ใหม่และเก็บไว้ใน drawerMap, graphCpmponentMap ตามลำดับ และเมื่อมีการอัพเดทค่าจะเรียก updateGraph เพื่อทำการอัพเดท

RealtimeUI

```
private void createDrawer(String selectedFilter, int indexFromAllData,
        int iTabProgramindex) {
    drawer = new Drawer(this, selectedFilter, indexFromAllData,
            iTabProgramindex);
    // add drawer to drawerList
    drawerMap.put(selectedFilter, drawer);
    // create graphComponent}
    graphComponentMap.put(selectedFilter, new GraphComponent());
    createWrapProgram(selectedFilter);
}

private void updateGraph() {
    Iterator it = drawerMap.entrySet().iterator();
    while (it.hasNext()) {
        Map.Entry mentry = (Map.Entry) it.next();
        String programName = (String) mentry.getKey();
        Drawer d = (Drawer) mentry.getValue();

        Update u = updateMap.get(programName);
        d.updateGraph(u);
    }
```

รูปที่ 4.29 การเริ่มต้นเพื่ออัพเดทข้อมูล

4.2 ผลการทำงานของระบบ

4.2.1 ผลการทำงานของเว็บไซต์

การแสดงผลบนเว็บไซต์ ผู้ใช้สามารถเลือกรูปแบบการแสดงผลได้ทั้งแบบ Compare คือ เปรียบเทียบหลายโปรเซสในกราฟเดียว และแบบ Multiple Graph คือดูหลายกราฟ โดยการเลือกรูปแบบเวลาสามารถเลือกได้ทั้งกำหนดเวลาที่แน่นอน และเวลาย้อนหลัง 1-4 ชั่วโมงย้อนหลัง โดยมีการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เริ่มต้นผู้ใช้เลือก Program ต่างๆที่มีการทำงานอยู่บนเซิฟเวอร์

รูปที่ 4.30 การเลือกโปรแกรมบนเว็บไซต์

2. เลือกรูปแบบการแสดงผล (2) ว่าต้องการแสดงผลแบบเปรียบเทียบ (Compare Graph) หรือแสดงผลหลายกราฟ (Multiple Graphs)
3. เลือกรูปแบบของข้อมูล (3) ว่าต้องการแสดงผลข้อมูลย้อนหลัง (Historical Data) หรือข้อมูลที่อัพเดทเรียลไทม์ (Multiple Graphs)

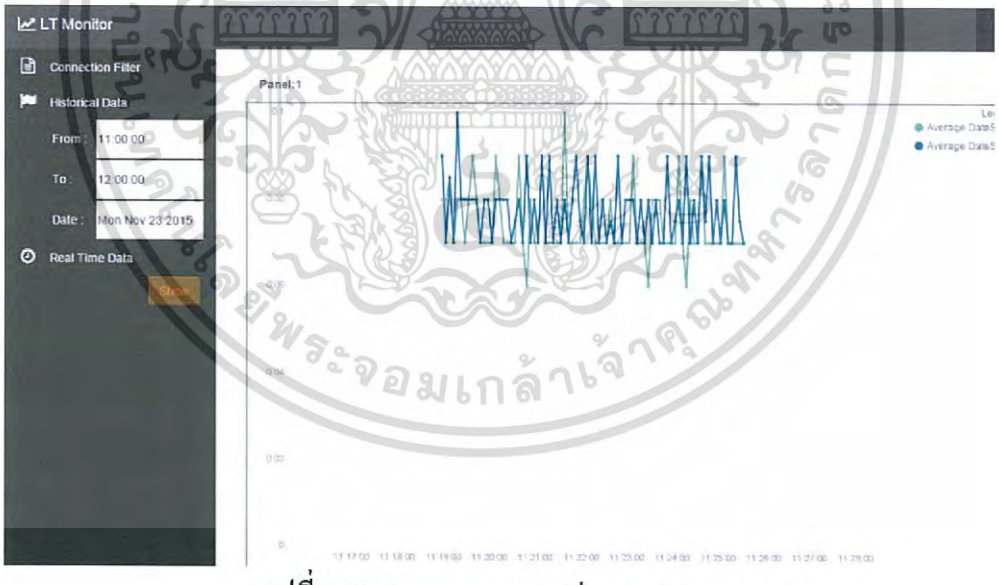
รูปที่ 4.31 การเลือกรูปแบบแสดงผลกราฟบนเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ถ้าเลือกข้อมูลย้อนหลัง กรอกเวลาเริ่มต้น-สิ้นสุด และวันที่ของข้อมูล โดยรูปแบบของเวลาคือ 00:00:00 กราฟข้อมูลจะแสดงเมื่อวัน-เวลาที่กรอกเป็นเวลาที่ LT Monitor Client ทำงานอยู่

รูปที่ **4.32** การเลือกเวลาที่ต้องการดูกราฟบนเว็บไซต์

5. แสดงผลกราฟในรูปแบบของกราฟเป็นเปรียบเทียบ (Compare) และรูปแบบของเวลา Historical Data ข้อมูลจึงแสดงหลายโปรเซสในกราฟเดียว ทำให้สามารถเปรียบเทียบการทำงานของโปรเซสต่างกันได้



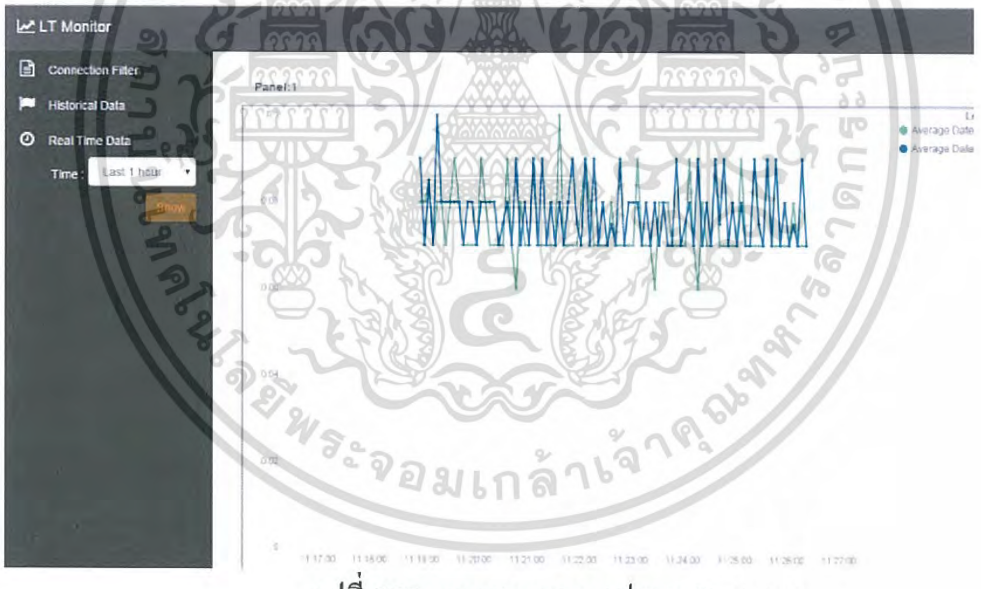
รูปที่ **4.33** การแสดงผลกราฟแบบ Historical Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ถ้าเลือกข้อมูลแบบเรียลไทม์ (Real-Time Data) สามารถเลือกได้ว่าต้องการดูล่าสุดกี่ชั่วโมง กำหนดให้เลือก 1-4 ชั่วโมง

รูปที่ 4.34 การเลือกรูปแบบของเวลาแบบ Real-time บนเว็บไซต์

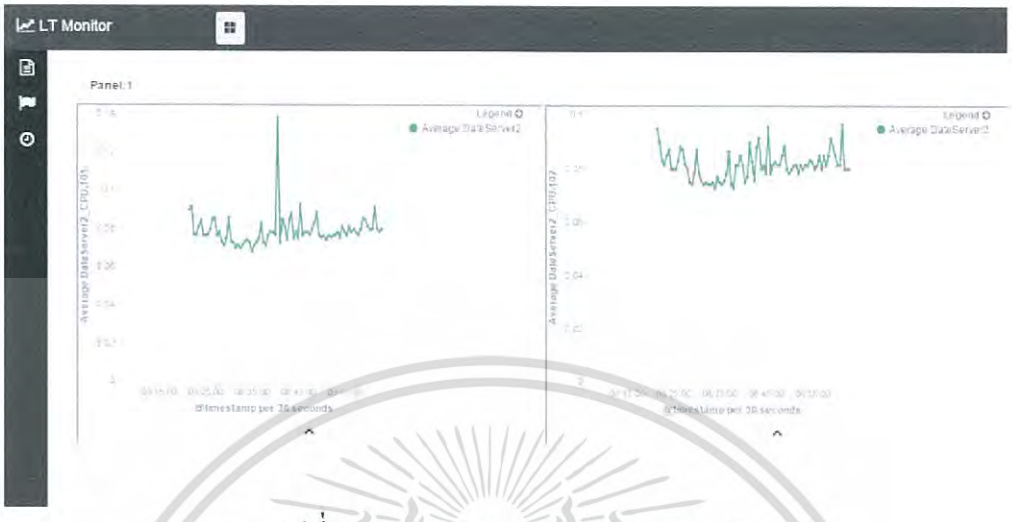
7. แสดงผลกราฟในรูปแบบของกราฟเป็นเปรียบเทียบ (Compare) และรูปแบบของเวลา Real-time Data ข้อมูลจึงแสดงหลายโปรเซสในกราฟเดียว ทำให้สามารถเปรียบเทียบการทำงานของโปรเซสต่างกันได้ และข้อมูลบนกราฟมีการอัพเดททุก 5 วินาที



รูปที่ 4.35 การแสดงผลกราฟแบบ Real-time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. แสดงผลข้อมูลกราฟในรูปแบบ Multiple Graph ซึ่งแสดงผลหลายกราฟในหนึ่งหน้าเว็บเพจ

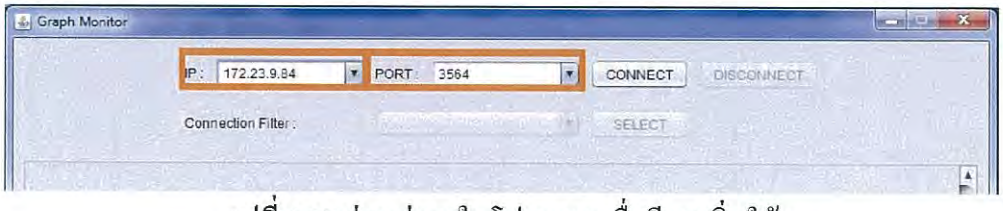


รูปที่ 4.36 การแสดงผลกราฟแบบ Multiple graph

4.2.2 ผลการทำงานของ Java Application

การทำงานบนโปรแกรมซึ่งพัฒนาโดย JAVA จะเชื่อมต่อกับเซิฟเวอร์โดยใช้ IP, Port ที่ถูกกรอกโดยผู้ใช้ จากนั้นเมื่อมีการเชื่อมต่อสำเร็จ โปรแกรมจะแสดงรายชื่อโปรเซสต่างๆที่รันอยู่บนเซิฟเวอร์เครื่องนั้นเพื่อให้ผู้ใช้เลือกโปรเซสที่จะแสดงผลเป็นกราฟ ซึ่ง System และ Disk จะแสดงผลได้ทั้งกราฟเส้นและกราฟวงกลม ส่วนโปรเซสต่างๆจะแสดงผลได้เพียงกราฟเส้น และแสดงผลอัพเดทข้อมูลทุก 4 วินาที เมื่อเลือกโปรเซสจะแสดงกราฟทั้งข้อมูล CPU, Memory ในรูปแบบของกราฟเส้น และเมื่อเลือก System, Disk โปรแกรมจะสามารถแสดงทั้งกราฟเส้นและกราฟวงกลมและขณะเดียวกันโปรแกรมจะสร้างไฟล์ Log เพื่อเป็นข้อมูล Input ให้กับการแสดงผลที่จะเกิดขึ้นบนเว็บไซต์ด้วย โดยมีการทำงานดังนี้

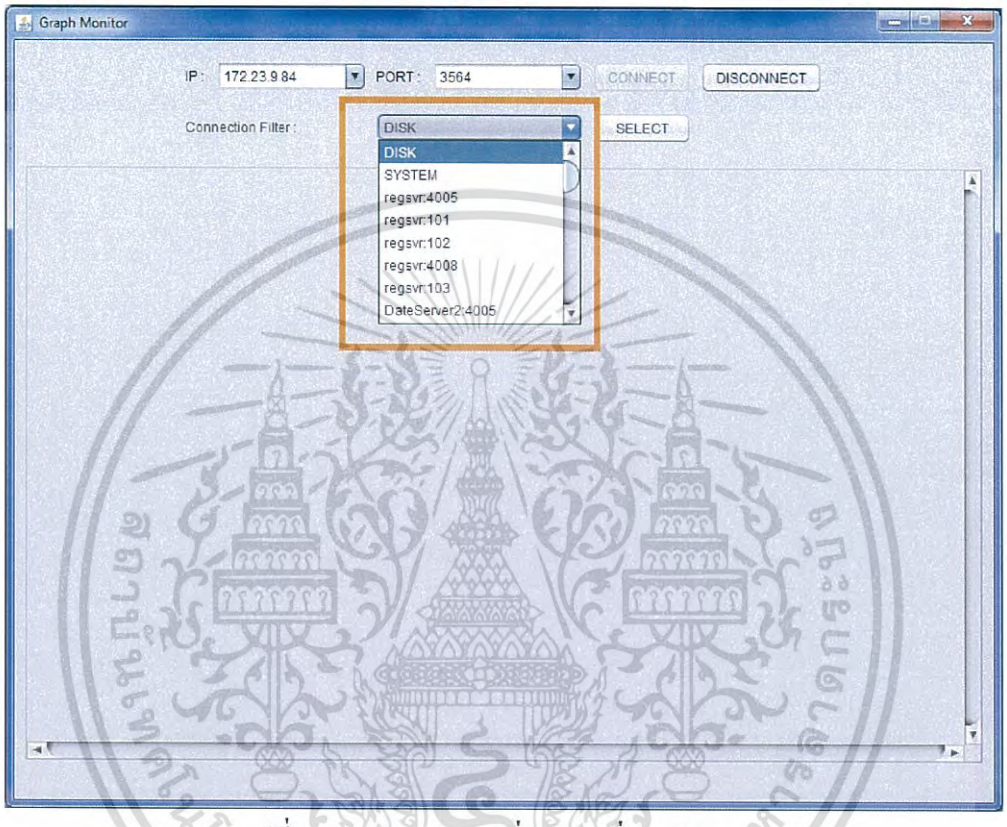
1. เริ่มต้นผู้ใช้กรอก IP และ Port ของเซิฟเวอร์ที่ต้องการดูข้อมูล โดยถ้าการเชื่มต่อยังไม่เกิดขึ้นช่อง Connection Filter และปุ่ม SELECT จะถูกตั้งค่าไม่ให้สามารถกดได้ และระบบจะจำข้อมูล IP และPort ของการเชื่อมต่อครั้งก่อน ทำให้ผู้ใช้ไม่ต้องกรอกข้อมูลเดิมซ้ำ



รูปที่ 4.37 ส่วนต่างๆในโปรแกรมเมื่อมีการเริ่มใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

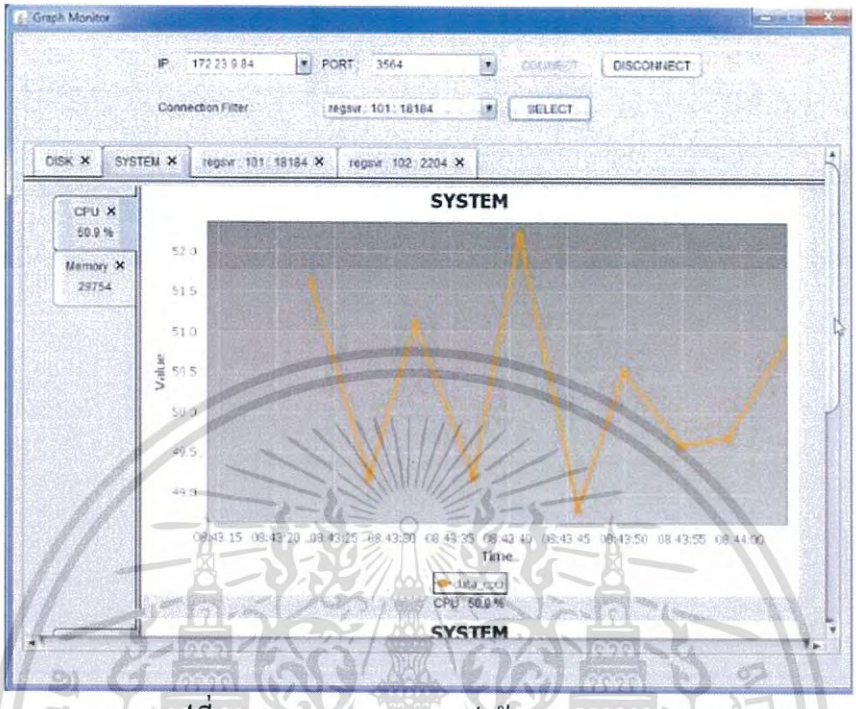
2. เมื่อกรอก IP และ Port ถูกต้อง กดปุ่ม Connect ระบบจะทำการเชื่อมต่อกับเซิฟเวอร์ดังกล่าว และแสดงโปรเซสที่ถูกใช้งานอยู่ขณะนั้นทาง Connection Filter เมื่อมีการเชื่อมต่อแล้วผู้ใช้จะไม่สามารถกดปุ่ม CONNECT ซ้ำได้ แต่สามารถยกเลิกการเชื่อมต่อ (DISCONNECT) ได้ ผู้ใช้สามารถเลือกโปรเซสที่ต้องการได้ครั้งละ 1 Process โดยการเลือกโปรเซสที่ต้องการและเลือกปุ่ม SELECT



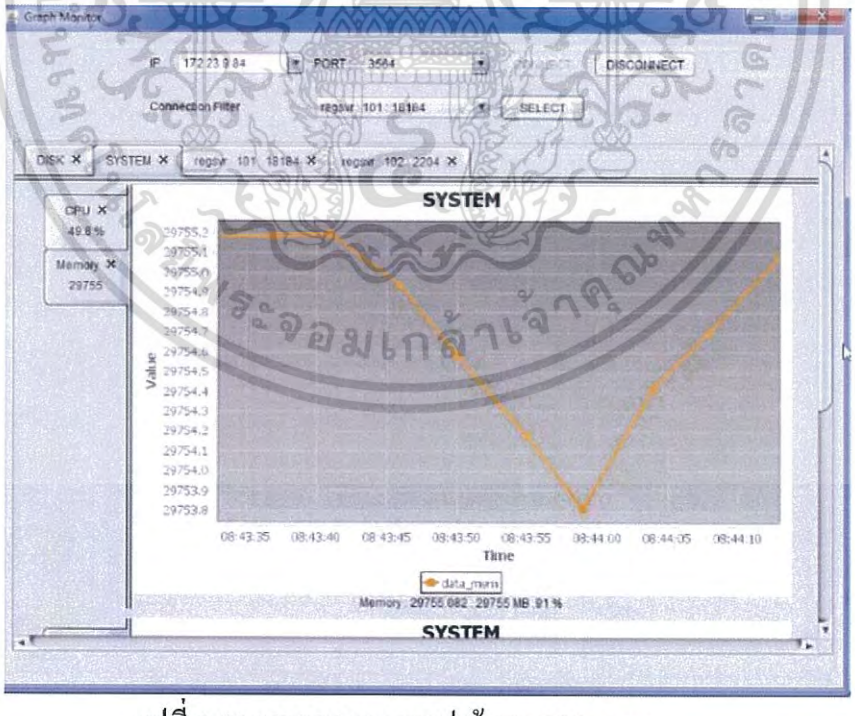
รูปที่ 4.38 การแสดงผลเมื่อมีการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ผู้ใช้เลือก Connection Filter ที่ต้องการ จากนั้นกดปุ่ม Select ระบบจะแสดงผลกราฟการใช้ CPU และ Memory ของโปรเซสที่ถูกเลือก ในที่นี้แสดงการใช้ระบบโดยรวมทั้งหมด

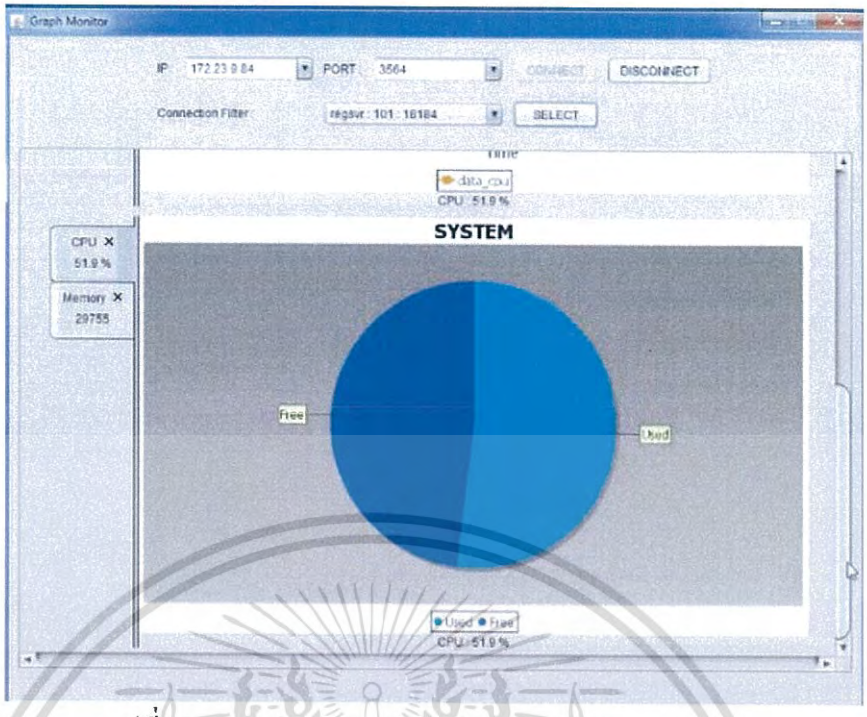


รูปที่ 4.39 การแสดงผลกราฟเส้นของ CPU

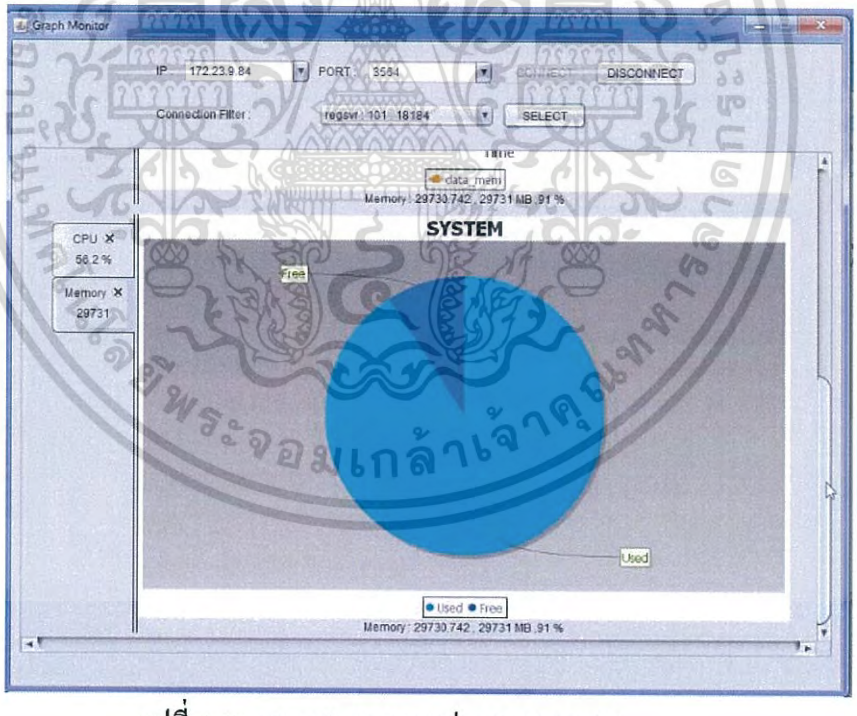


รูปที่ 4.40 การแสดงผลกราฟเส้นของ Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 การแสดงผลกราฟวงกลมของ CPU



รูปที่ 4.42 การแสดงผลกราฟวงกลมของ Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลโครงงาน และข้อเสนอแนะ

ในบทนี้จะกล่าวถึงบทสรุปของการฝึกงานสหกิจกับ บริษัท รอยเตอร์ ซอฟแวร์ (ประเทศไทย) จำกัด ในโครงงานการพัฒนาการพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ โดยเป็นการพัฒนาแอพพลิเคชั่นจาก JAVA application ที่สามารถแสดงผลเป็นกราฟแบบเรียลทามและแบบดูข้อมูลย้อนหลังได้และการพัฒนาเว็บแอพพลิเคชั่นที่สามารถเชื่อต่อกับเซิฟเวอร์และแสดงกราฟข้อมูลแบบเรียลทามได้เพื่อสังเกตการณ์การ, ตรวจสอบใช้ทรัพยากรของเซิฟเวอร์

5.1 สรุปผลโครงงาน

การพัฒนาการพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศนี้มีวัตถุประสงค์เพื่อสังเกตการณ์, ตรวจสอบการใช้ทรัพยากร (CPU, Memory) ของแต่ละเซิฟเวอร์ ซึ่งในเซิฟเวอร์แต่ละเครื่องจะมีโปรเซสถูกใช้งานอยู่จำนวนมาก หากมีการทำงานผิดปกติเกิดขึ้นผู้ใช้สามารถตรวจสอบได้ในขณะนั้นว่าเกิดความผิดปกติขึ้นที่โปรเซสใด และเซิฟเวอร์จะมีการทำงานตลอดเวลา (24 ชั่วโมง) ดังนั้นผู้ใช้สามารถย้อนดูการใช้ทรัพยากรของโปรเซสในอดีตขณะที่ผู้ใช้ไม่ได้สังเกตการณ์การณ์หรือเมื่อผู้ใช้ไม่อยู่ได้ และผู้ใช้สามารถสังเกตการณ์ข้อมูลของกราฟจากการแสดงผลแบบเปรียบเทียบได้ผ่านเว็บไซต์ โดยการแสดงผลจะแสดงผลผ่านจาว่าแอพพลิเคชั่นที่เป็นการแสดงผลแบบเรียลทามและเว็บแอพพลิเคชั่นที่เป็นการแสดงผลกราฟทั้งแบบเรียลทาม (Real time data) และย้อนหลัง (Historical data)

5.2 ปัญหาและอุปสรรค

ในการพัฒนาระบบตั้งแต่เริ่มต้นพัฒนาจนถึงการทดสอบระบบ สามารถสรุปปัญหาที่พบและข้อจำกัดของระบบได้ดังนี้

5.2.1 ปัญหาที่พบ

- ข้อมูลในเรื่อง Elasticsearch , Logstash , Kibana มีให้ศึกษาไม่มาก
- การเขียนโปรแกรมให้ Logstash สามารถอ่านข้อมูลลง Elasticsearch ไม่มีรูปแบบการเขียนที่แน่นอน การอ่านข้อมูล Sensitive ต่อ Error เช่น ชื่อไฟล์ข้อมูลห้ามเป็นตัวใหญ่, ชื่อ Index ห้ามเป็นตัวใหญ่
- การเขียนไฟล์ข้อมูลจากเซิฟเวอร์ต้องเขียนในรูปแบบที่ Logstash สามารถนำข้อมูลมาใช้ได้ บางรูปแบบ Logstash ก็ไม่สามารถถูกเขียนรูปแบบการ Config ได้ ซึ่งรูปแบบที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเขียนให้ Logstash อ่านลง Elasticsearch ได้ในระบบนี้คือรูปแบบที่ไม่มีมาตรฐานและกำหนดขึ้นเอง (.log)

- ปัญหาเรื่อง Timezone ข้อมูลที่เป็นข้อมูลดิบภายในไฟล์จากเซิฟเวอร์จะอยู่ใน Timezone GMT จึงไม่ตรงกับเวลาเครื่องที่เป็น GMT + 7 ดังนั้นเมื่อนำกราฟจาก Kibana มาใช้ต้องใช้ Tag การแสดงข้อมูลของ Kibana ซึ่งเวลาที่อยู่ภายใน Tag ใช้คำว่า now เป็นตัวอ้างอิง ดังนั้นจึงต้องนำ now (เวลาเครื่อง) มาคำนวณให้เป็นเวลาที่ต้องการซึ่งอ้างอิงกับ GMT + 7 (เปลี่ยน GMT+7 ให้เป็น GMT)

5.2.2 ข้อจำกัดของระบบ

- ข้อมูลที่นำมาแสดงผลบนเว็บมีความล่าช้า (Delay) กว่าข้อมูลจริง ดังนั้นหากต้องการสังเกตการณ์แบบ Real-time สามารถทำได้ใน JAVA application ซึ่งตรงเวลามากกว่าและอัพเดททุก 4 วินาที
- ต้องมีการเปิดโปรแกรม LT Monitor Client เสมอเมื่อต้องการข้อมูล หากเวลาที่ต้องการดูข้อมูลไม่ได้มีการทำงานของ LT Monitor Client จะไม่สามารถแสดงกราฟแบบย้อนหลังบนเว็บไซต์ได้

5.3 ข้อเสนอแนะ

ข้อมูลที่นำมาแสดงผลบนเว็บไซต์ในรูปแบบ Real-time ไม่ควรมีความล่าช้า และควรอัพเดทข้อมูลให้มีความเร็วเท่ากับ JAVA application ซึ่งอัพเดทข้อมูลทุก 4 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] "**Basic Concepts**". [Online]. Available:
https://www.elastic.co/guide/en/elasticsearch/reference/current/_basic_concepts.html

[2] Somkiat Puisungnoen. **"Elasticsearch กับการจัดการข้อมูล"**. [Online]. Available:
http://www.somkiat.cc/elasticsearch-your-data-flow/

[3] Somkiat Puisungnoen. "Logstash". [Online]. Available:
http://www.somkiat.cc/manage-log-message-with-logstash/

[4]"**Kibana 4 Video Tutorials, Part 2**" [Online]. Available: https://www.elastic.co/blog/kibana-4-video-tutorials-part-2

[5] "**Kibana 4 Video Tutorials, Part 4**" [Online]. Available: https://www.elastic.co/blog/kibana--4video-tutorials-part-4

[6]"**Skip header Line**" [Online]. Available: https://discuss.elastic.co/t/skip-header-line-in-csv-input-v-1-5-0/1386

[7]"**Elasticsearch-head**" [Online]. Available: https://github.com/mobz/elasticsearch-head

[8]"**JFreeChart**" [Online]. Available:
http://www.tutorialspoint.com/jfreechart/jfreechart_timeseries_chart.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ตารางการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 การทำงาน เดือนสิงหาคม 2558

วัน/เดือน/ปี	รายการ
4/8/2015	เลือกโปรแกรมที่ใช้ทำโปรเจค(Select program for project)
5/8/2015	ออกแบบหน้าแสดงผลการใช้งานสำหรับผู้ใช้(Design User Interface)
6/8/2015	ออกแบบหน้าแสดงผลการใช้งานสำหรับผู้ใช้(Design User Interface)
7/8/2015	สร้างการฟแบบ Real time โดยใช้ Library JFreeChart และ Random Data (Create real-time graph by JFreeChart Library and random data)
10/8/2015	1. สร้างการฟแบบ Real time โดยใช้ Library JFreeChart และ Random Data (Create real-time graph by JFreeChart Library and random data) 2. สร้างหน้าแสดงผลการใช้งานของผู้ใช้ (Create user interface by SWING and AWT)
11/8/2015	สร้างหน้าแสดงผลการใช้งานของผู้ใช้ (Create user interface by SWING and AWT)
12/8/2015	สร้างหน้าแสดงผลการใช้งานของผู้ใช้ (Create user interface by SWING and AWT)
13/8/2015	สร้างหน้าแสดงผลการใช้งานของผู้ใช้ (Create user interface by SWING and AWT)
14/8/2015	สร้างหน้าแสดงผลการใช้งานของผู้ใช้ (Create user interface by SWING and AWT)
17/8/2015	สร้างหน้าแสดงผลการใช้งานของผู้ใช้ (Create user interface by SWING and AWT)
18/8/2015	จัดการส่วนการโต้ตอบกับผู้ใช้ (Handle event of each button)
19/8/2015	จัดการส่วนการโต้ตอบกับผู้ใช้ (Handle event of each button)
20/8/2015	จัดการส่วนการโต้ตอบกับผู้ใช้ (Handle event of each button)
21/8/2015	1. จัดการส่วนการโต้ตอบกับผู้ใช้ (Handle event of each button) 2. แสดงผลกราฟจากการเลือกของผู้ใช้ (Show graph from selected program)
24/8/2015	แสดงผลกราฟจากการเลือกของผู้ใช้ (Show graph from selected program)
25/8/2015	1. แสดงผลกราฟจากการเลือกของผู้ใช้ (Show graph from selected program) 2. พรีเซนต์ส่วนแสดงผลและการทำงาน (Present user interface and process)
26/8/2015	ศึกษาการใช้งาน Elasticsearch,Logstash,Kibana
27/8/2015	ศึกษาการใช้งาน Elasticsearch,Logstash,Kibana

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 (ต่อ) การทำงาน เดือนสิงหาคม 2558

28/8/2015	ศึกษาการใช้งาน Elasticsearch,Logstash,Kibana
31/8/2015	ศึกษาการใช้งาน Elasticsearch,Logstash,Kibana

ตารางที่ ก.2 การทำงาน เดือนกันยายน 2558

วัน/เดือน/ปี	รายการ
1/9/2015	ประยุกต์การใช้งาน Elasticsearch,Logstash,Kibana
2/9/2015	ประยุกต์การใช้งาน Elasticsearch,Logstash,Kibana
3/9/2015	1. ประยุกต์การใช้งาน Elasticsearch,Logstash,Kibana 2. เก็บข้อมูลลงElasticsearch ผ่านการใช้Logstash
4/9/2015	1. ประยุกต์การใช้งาน Elasticsearch,Logstash,Kibana 2. เก็บข้อมูลลงElasticsearch ผ่านการใช้Logstash
7/9/2015	เก็บข้อมูลลงElasticsearch ผ่านการใช้Logstash
8/9/2015	เก็บข้อมูลลงElasticsearch ผ่านการใช้Logstash
8/9/2015	หาวิธีการรับข้อมูลจาก Kibana
9/9/2015	หาวิธีการรับข้อมูลจาก Kibana
10/9/2015	สร้างหน้าแสดงการใช้งานของWebsite
11/9/2015	สร้างหน้าแสดงการใช้งานของWebsite
14/9/2015	หาวิธีการดึงข้อมูล(ชื่อคอลัม)จากElasticsearchเข้ามาบนWebsite เพื่อเป็นข้อมูลในการเลือกโปรแกรม
15/9/2015	หาวิธีการดึงข้อมูล(ชื่อคอลัม)จากElasticsearchเข้ามาบนWebsite เพื่อเป็นข้อมูล(Dropdown)ในการเลือกโปรแกรม
16/9/2015	หาวิธีการเลือกเฉพาะข้อมูลมาแสดงผลบนDropdownแบบเลือกได้หลายค่า โดยใช้ข้อมูลจำลอง
17/9/2015	หาวิธีการเลือกเฉพาะข้อมูลมาแสดงผลบนDropdownแบบเลือกได้หลายค่า โดยใช้ข้อมูลจำลอง
18/9/2015	หาวิธีการเลือกเฉพาะข้อมูลมาแสดงผลบนDropdownแบบเลือกได้หลายค่า โดยใช้ข้อมูลจำลอง
21/9/2015	แสดงผลข้อมูลที่เลือกมาจากการเลือกโปรแกรมที่ต้องการ
22/9/2015	แก้ปัญหาการRefreshค่าในdropdown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.2 (ต่อ) การทำงาน เดือนกันยายน 2558

23/9/2015	1. สร้างกราฟจากการเลือกข้อมูลแบบเปรียบเทียบ(หลายโปรแกรมในกราฟเดียว) 2. สร้างการจากการเลือกข้อมูลแบบหลายกราฟในหน้าเว็บเพจ
24/9/2015	1. ควบคุมการแสดงผลตามที่ผู้ใช้เลือกในแบบHistorical data และ Real time 2.สร้างส่วนแสดงผลควบคุมการกดปุ่มและการเลือก
25/9/2015	1. ควบคุมการแสดงผลตามที่ผู้ใช้เลือกในแบบHistorical data และ Real time 2.สร้างส่วนแสดงผลควบคุมการกดปุ่มและการเลือก
28/9/2015	เชื่อมเวลาของการแสดงผลข้อมูลในKibanaและเวลาของข้อมูลในไฟล์ให้อยู่ในรูปแบบเดียวกัน(GMT)
29/9/2015	แก้ปัญหาเวลาของข้อมูลไม่ตรงกัน(Timezone)
30/9/2015	แก้ปัญหาเวลาของข้อมูลไม่ตรงกัน(Timezone)

ตารางที่ ก.3 การทำงาน เดือนตุลาคม 2558

วัน/เดือน/ปี	รายการ
1/10/2015	1.แสดงกราฟแบบหลายกราฟในเว็บเพจ(Multiple graph) 2.จัดcode
2/10/2015	1.ทำเอกสารคณะ 2.จัดcode
5/10/2015	ทดสอบการติดต่อกับเซิฟเวอร์
6/10/2015	แก้ไขปัญหาTimezone
7/10/2015	หาวิธีรับชื่อหัวคอลัมแบบdynamic
8/10/2015	หาวิธีรับชื่อหัวคอลัมแบบdynamic
9/10/2015	ทำเอกสารคณะ
12/10/2015	ทำLTmonitorClient
13/10/2015	หาวิธีทำให้เพิ่มข้อมูลในelasticsearchแล้ว ได้ชื่อหัวคอลัมเอง(โดยเขียนวิธีget data ใน logstash.conf)
14/10/2015	ทำให้แอพLT monitor ติดต่อกับserverได้ โดยเอาโค้ด DebugReplyableConnection มาใช้
13/10/2015	หาวิธีทำให้เพิ่มข้อมูลในelasticsearchแล้ว
13/10/2015	ได้ชื่อหัวคอลัมเอง(โดยเขียนวิธีget data ใน logstash.conf)
14/10/2015	ทำให้แอพLT monitor ติดต่อกับserverได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.3 (ต่อ) การทำงาน เดือนตุลาคม 2558

14/10/2015	โดยเอาโค้ด DebugReplyableConnection มาใช้
15/10/2015	1. Send / Get message between my app and server 2. Remember input 3. Refractor class
16/10/2015	1. Write message in file 2. Create file per day
19/10/2015	เขียนข้อมูลลงไฟล์ในformatที่ต้องการได้โดยให้ข้อมูลแสดงตามลิสที่กำหนดไว้/ถ้าไม่มีข้อมูลให้เป็น0
20/10/2015	formatถ้าข้อมูลน้อยจะเป็นnumberได้แต่พอข้อมูลเยอะจะเป็นStringหมด
21/10/2015	1. ทำ Automatic data ได้ 2. ทำให้ข้อมูลเป็น float ได้
22/10/2015	สร้างกราฟแบบ Real time บนJava
23/10/2015	สร้างกราฟแบบ Real time บนJava
26/10/2015	เตรียมตัวPresentอาจารย์นิเทศน์
27/10/2015	สร้างกราฟแบบ Real time บนJava โดยใช้ข้อมูลจริง
28/10/2015	สร้างกราฟแบบ Real time บนJava โดยใช้ข้อมูลจริง
29/10/2015	สร้างกราฟแบบ Real time บนJava โดยใช้ข้อมูลจริง
30/10/2015	สร้างกราฟแบบ Real time บนJava โดยใช้ข้อมูลจริง

ตารางที่ ก.4 การทำงาน เดือนพฤศจิกายน 2558

วัน/เดือน/ปี	รายการ
2/11/2015	เปลี่ยนstructureให้เป็นobject oriented
3/11/2015	เปลี่ยนstructureให้เป็นobject oriented
4/11/2015	เปลี่ยนstructureให้เป็นobject oriented
5/11/2015	เปลี่ยนstructureให้เป็นobject oriented
6/11/2015	เพิ่มกราฟresourceรวมและdiskของserver
9/11/2015	เพิ่มpie chart
10/11/2015	เพิ่มpie chart
11/11/2015	เพิ่มpie chart
12/11/2015	เพิ่มการจำIP,Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.4 (ต่อ) การทำงาน เดือนพฤศจิกายน 2558

13/11/2015	แก้ไข Error ใน Java application
16/11/2015	1.แก้รายละเอียดข้อมูลใน Java application 2.ทำเล่มรายงาน
17/11/2015	เพิ่มรายละเอียดข้อมูลใน Java application
19/11/2015	แก้ไข Error UI ใน Java application
23/11/2015	ส่งงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ – นามสกุล	นางสาวสุประวีณ์ ยิ้มเนียม
วัน เดือน ปีเกิด	27 พฤษภาคม 2536
ที่อยู่	1172/1 หมู่ 9 ซอย พัฒนาสุข ถนนเทพารักษ์ ต.เทพารักษ์ อ.เมือง จ.สมุทรปราการ 10270
โทรศัพท์	086-4045962
อีเมล	suprawee.yimnium@gmail.com

ประวัติการศึกษา

2558	วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ

Resource Monitoring and Visualization Interface for Foreign Exchange (FX) Electronic Trading

สุประวีณ์ ยิ้มเนียม[1] และ ผศ. ดร. ธีรพงศ์ ลีลานุภาพ[2]

[1]*คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง*

Emails: suprawee.yimnium@gmail.com, teerapong@it.kmitl.ac.th

บทคัดย่อ

ปัจจุบันมีความต้องการที่จะเก็บข้อมูลจำนวนมากที่รวบรวมจากการทำธุรกรรมในแต่ละวันโดยเฉพาะอย่างยิ่งการเก็บข้อมูลธุรกรรมเกี่ยวกับการแลกเปลี่ยนหลักทรัพย์เงินตราต่างประเทศซึ่งข้อมูลมีการเปลี่ยนแปลงเป็นจำนวนมาก ทีม Reuters Electronic Trading (RET) ของบริษัท รอยเตอร์ ซอฟแวร์ (ประเทศไทย) จำกัด มีหน้าที่รับผิดชอบการพัฒนาและบำรุงรักษาการทำงานของซอฟแวร์ด้านการเงิน และมีการใช้งานโปรเซสต่างๆจำนวนมากอยู่บนเครื่องเซิฟเวอร์ ดังนั้นการเคลื่อนไหวของข้อมูลจะถูกเปลี่ยนแปลงตลอดเวลา การตรวจสอบการใช้ทรัพยากรของ CPU และ Memory สามารถทำได้ยาก การพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศจึงถูกพัฒนาขึ้นเพื่อลดความยากลำบากในการตรวจสอบข้อมูล โดยระบบที่ถูกพัฒนาขึ้นสามารถรวบรวมข้อมูลและนำมาแสดงผลในรูปแบบกราฟ ดังนั้นผู้ใช้จึงสามารถตรวจสอบการใช้ทรัพยากรได้อย่างสะดวก

คำสำคัญ – Reuters Electronic Trading; CPU; Memory

1.บทนำ

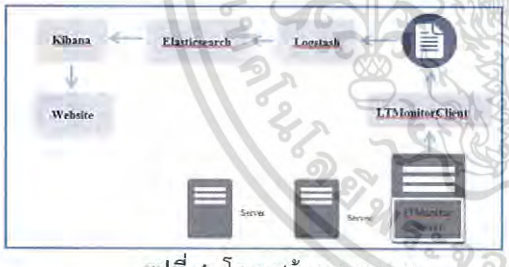
ปัจจุบันเทคโนโลยีกำลังก้าวเข้าสู่ Big Data เนื่องจากข้อมูลเริ่มมีจำนวนมากและหลากหลายมากขึ้น จนทำให้การเก็บข้อมูลในรูปแบบ Database ในรูปแบบธรรมดาอาจไม่สามารถรองรับการทำงานได้อย่างมีประสิทธิภาพเพียงพอ ถ้าทำได้ก็อาจไม่ครอบคลุมตามความต้องการการทำงานได้ทั้งหมด หรือการจัดการกับข้อมูลอาจต้องใช้เวลามากพอสมควร เช่น การดึงข้อมูลมาใช้ให้ตรงตามเวลา, การประมวลผลข้อมูล ซึ่งการทำงานดังกล่าวของการเก็บข้อมูลที่มีปริมาณมาก, ข้อมูลที่มีความหลากหลาย และข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลาอย่างรวดเร็วเสมือนเป็นข้อมูลแบบ Real-time เป็นหลักการที่สำคัญของเทคโนโลยี Big Data ประกอบกับทีมเซิฟเวอร์ (Server Team) ในบริษัท รอยเตอร์ ซอฟแวร์ (ประเทศไทย) จำกัด พัฒนาซอฟแวร์ในการซื้อขายหลักทรัพย์ทางการเงิน โดยงานส่วนหนึ่งที่สำคัญของทีม คือ ทำหน้าที่ดูแลและจัดการ การใช้ทรัพยากรของโปรแกรมต่าง ๆ ที่ทำงานอยู่ในแต่ละเซิฟเวอร์ ซึ่งการใช้ทรัพยากรต่าง ๆ เมื่อมีการใช้งานจะมีการอัพเดทโปรแกรมในเซิฟเวอร์ต่าง ๆ ที่มีอยู่ทั่วโลกอย่างรวดเร็วและมีจำนวนของข้อมูลมากขึ้นเรื่อย ๆ เมื่อผู้ใช้ทำการสังเกตการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเซิฟเวอร์จะต้องเข้าไปดูไฟล์ซึ่งเก็บ Log ที่ทีมได้พัฒนาขึ้น ซึ่งการสังเกตการณ์, การเปรียบเทียบการใช้ทรัพยากรของโปรแกรมในเซิฟเวอร์ และการวิเคราะห์ข้อมูลจากไฟล์ที่มีจำนวนมากนั้นอาจทำได้ไม่คล่องตัวเท่าที่ควรหรือหากเกิดปัญหา เช่น ความผิดปกติจากการใช้ทรัพยากรมากเกินไปของบางโปรแกรมในเซิฟเวอร์ อาจทำให้แก้ไขปัญหาได้ล่าช้า ซึ่งปัญหาการจัดเก็บข้อมูลดังกล่าวตรงกับหลักการทำงานของ Big Data คือ ข้อมูลมีจำนวนในปริมาณมาก (Volume) และข้อมูลมีการอัพเดทอย่างรวดเร็ว (Velocity) หรือเป็นข้อมูลประเภท Real-time ดังนั้น ระบบส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศ จึงถูกพัฒนาขึ้น โดยใช้หลักการทำงานของ Big Data และนำเทคโนโลยีที่มีอยู่มาประยุกต์การทำงานเพื่อให้สอดคล้องกับระบบและความต้องการขององค์กร เพื่อให้ผู้ใช้ที่ต้องการสังเกตการณ์การใช้ทรัพยากรของเซิฟเวอร์ สามารถทำงานได้อย่างมีประสิทธิภาพ

2.ทฤษฎีและหลักการ

หลักการและภาพรวมของโปรแกรมสามารถแสดงเป็นภาพรวมได้ดังนี้



รูปที่ 1. โครงสร้างของระบบ

1.1 ส่วนของการทำงานบนเว็บไซต์

การทำงานบนเว็บไซต์ประกอบด้วยการประยุกต์ใช้ส่วนต่างๆดังนี้

Elasticsearch

Elasticsearch เป็นเครื่องมือที่ช่วยในการเก็บข้อมูล , ค้นหาข้อมูล และวิเคราะห์ข้อมูลที่มีจำนวนมากได้อย่างรวดเร็วเป็นฐานข้อมูลประเภท NoSQL ที่สามารถเก็บข้อมูลได้ตั้งแต่ข้อมูลจำนวนน้อยไปจนถึงข้อมูลขนาดใหญ่ระดับ Big Data ได้, ประสิทธิภาพการทำงานสูงและรองรับการขยายของระบบ และสามารถเก็บข้อมูลที่มีการอัพเดทเปลี่ยนแปลงตลอดเวลาได้ จึงไม่ต้องมีการกำหนดโครงสร้างหรือจำนวนข้อมูลก่อนจึงจะสามารถเก็บข้อมูลได้เหมือนฐานข้อมูลแบบ RDBMSโดยข้อมูลที่ถูกเก็บสามารถเรียกดูได้ผ่านปลั๊กอิน head ของ Elasticsearch บนเว็บไซต์

Logstash

Logstash คือ เครื่องมือสำหรับจัดการกับ log และ ข้อมูลต่างๆ โดย Logstash สามารถนำข้อมูลจำนวนมากๆมาสะสม กรอง และจัดการข้อมูลเพื่อใช้งานต่อไป ในที่นี้ Logstash จะนำข้อมูลจาก Source Data ที่ได้มาจากการแปลงข้อมูลของ LT Monitor Client ซึ่งรับข้อมูลต่างๆมาจากเซิฟเวอร์ เป็นไฟล์ข้อมูลที่รูปแบบข้อมูลถูกกำหนดขึ้นมาเอง Logstash นำข้อมูลไปประมวลผลและจัดรูปแบบของข้อมูลให้อยู่ในรูปแบบที่ต้องการ และ ส่งออกไปเก็บไว้ที่ปลายทางที่ระบุไว้ได้ ในระบบนี้ใช้ปลายทางเป็น Elasticsearch เพื่อเก็บข้อมูล ในระบบนี้ข้อมูลที่นำเข้าสู่ Logstash เป็นรูปแบบเฉพาะที่ถูกกำหนดขึ้นมาเอง หากรูปแบบข้อมูลนำเข้ามีการเปลี่ยนแปลงหรือเป็นรูปแบบอื่น Logstash จะไม่สามารถเข้าใจและจัดเก็บลงใน Elasticsearch ได้

Kibana

Kibana เป็นเครื่องมือที่ใช้ในการ ค้นหา และจัดการกับข้อมูลที่ถูกเก็บไว้ใน Elasticsearch ได้ โดยการแสดงผลสามารถดึงข้อมูลมาแสดงผลได้แบบ Real-time

ภาษา HTML

ภาษา HTML เป็นภาษาที่ใช้ในการสร้างเว็บเพจ โดยโครงสร้างของภาษาที่ใช้ในการเขียนจะเป็นแท็กต่างๆ (Tag) ในการควบคุมและแสดงผลให้ออกมาบนหน้าเพจ

ภาษา Javascript

ภาษา Javascript เป็นภาษาที่ใช้ร่วมกับ HTML และเป็นภาษา Script ที่อยู่บนเว็บไซต์ต่างๆทำให้เว็บไซต์สามารถโต้ตอบกับผู้ใช้ได้และมีลูกเล่นทำให้เว็บเพจน่าสนใจมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง Javascript เป็นสคริปฝั่งไคลเอนท์ จึงประมวลผลบนเว็บเบราเซอร์

ภาษา jQuery

ภาษา jQuery เป็นภาษาที่พัฒนามาจากภาษา Javascript ที่ใช้ในการทำงานร่วมกับ HTML เพื่อจัดการด้านไคลเอนท์ (Client) หรือเบราเซอร์ (Broswer) เพื่อให้ตอบสนองต่อผู้ใช้โดยตรง

ภาษา PHP

PHP คือภาษาที่ใช้พัฒนาเว็บไซต์โดยเป็นลักษณะ Dynamic จึงทำให้ เว็บไซต์สามารถเปลี่ยนแปลงไปยังรูปแบบต่างๆได้ตามเงื่อนไขที่ระบุไว้ ซึ่ง PHP เป็นภาษา Script หมายความว่าสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจร่วมกับคำสั่งของ HTML ได้

1.2 ส่วนของ LT Monitor Client

ภาษา JAVA

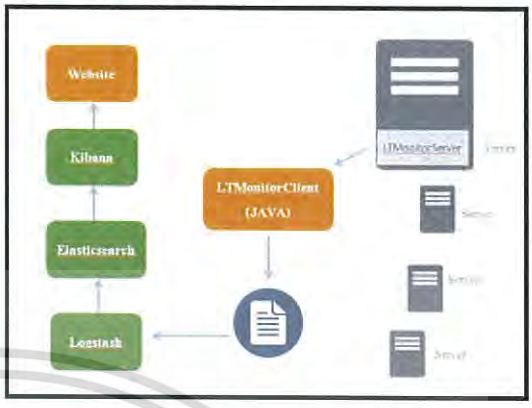
ภาษา JAVA เป็นภาษาเขียนโปรแกรมที่สามารถทำงานได้ในระบบปฏิบัติการที่หลากหลายโดยไม่ต้องแก้ไขโปรแกรมให้ทำงานได้หากต้องรันในระบบปฏิบัติการอื่น , สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object Oreinted Programming) และมี IDE , Library ต่างๆให้ใช้งานมากมาย โดยในระบบนี้ได้มีการนำ Library ที่ชื่อว่า JFreeChart มาใช้งาน เนื่องจากเป็นไลบรารี่ที่สามารถสร้างกราฟแบบ Real-time ได้และ เป็น Open Source ที่สามารถนำไปใช้ได้ฟรีและรองรับการสร้างรูปภาพ

3.การวิเคราะห์และออกแบบระบบ

3.1 ปัญหาที่พบในระบบปัจจุบัน

เนื่องจากการทำงานในลักษณะเดิม เมื่อผู้ใช้ต้องการสังเกตการณ์ข้อมูลการใช้ทรัพยากรในแต่ละเซิฟเวอร์ เพื่อทราบสาเหตุจากปัญหาต่างๆ เช่น มีการทำงานผิดปกติเกิดขึ้น , การใช้ทรัพยากรบางเซิฟเวอร์เกินความจำเป็นหรือเฝ้าสังเกตเพื่อให้การทำงานของเซิฟเวอร์มีประสิทธิภาพ ผู้ใช้จะต้องเข้าไปดูที่ไฟล์ CSV ซึ่งเป็นไฟล์ที่เก็บข้อมูลการใช้ CPU และ Memory ของทุกๆโปรแกรมในเซิฟเวอร์ที่มีการใช้งาน ทุกๆ 4 วินาที ดังนั้นข้อมูลจึงมีการอัพเดทเพิ่มขึ้นเรื่อยๆ ข้อมูลจึงมีจำนวนมากทำให้ยากต่อการสังเกตและเปรียบเทียบ และหากอยากดูข้อมูลในรูปแบบกราฟผู้ใช้จะสามารถดูได้จากการใช้ Excel

3.2 การออกแบบระบบใหม่



รูปที่ 2. การทำงานของระบบใหม่

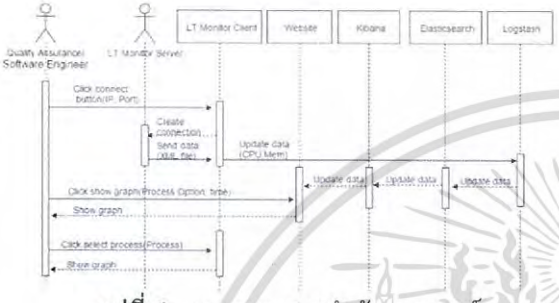
จากภาพแสดงภาพรวมการทำงานของระบบใหม่โดยสีส้มคือระบบที่พัฒนาเอง สีเขียวคือระบบที่เข้าไปจัดการเพื่อให้ทำงานได้ และมีการทำงานของระบบต่างๆดังนี้

- LT Monitor Server จะทำงานแบบเดิมโดยจะเป็นโปรแกรมที่อยู่ในเครื่องเซิฟเวอร์แต่ละเครื่องและส่งข้อมูลให้ LT Monitor Client เมื่อมีการเชื่อมต่อเกิดขึ้น
- LT Monitor Client เป็นแอพพลิเคชันที่พัฒนาจากภาษาจาว่า(JAVA) ซึ่งพัฒนาขึ้นมาใหม่ทั้งหมดโดยจะเชื่อมต่อกับ LT Monitor Server ,นำรับข้อมูลในรูปแบบ XMLมาเก็บเป็นไฟล์ที่มีการเปลี่ยนแปลงข้อมูลเป็นรูปแบบที่ Logstash สามารถนำไปเก็บใน Elasticsearch ได้ และสามารถแสดงผลกราฟเส้น, กราฟวงกลมของข้อมูลแบบ Real time ได้
- Logstash ทำหน้าที่นำไฟล์ข้อมูลที่ถูกสร้างจาก LT Monitor Client ไปเก็บใน Elasticsearch โดยจะต้องกำหนดรูปแบบของการอ่านข้อมูลและการเขียนข้อมูลที่จะถูกเก็บต้องอยู่ในรูปแบบที่ต้องการ
- Elasticsearch คือฐานข้อมูลประเภท NoSQL ที่สามารถเก็บข้อมูลที่เปลี่ยนแปลงหรือเพิ่มตลอดเวลาได้ (Dynamic Data) และ รองรับการเก็บข้อมูลที่มีจำนวนมากแต่ไม่ซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Kibana คือเว็บแอพพลิเคชั่นที่สามารถแสดงผลข้อมูลที่อยู่ใน Elasticsearch ได้
- Website คือเว็บไซต์ที่พัฒนาขึ้นเพื่อแสดงผลกราฟทั้งแบบ Real-time และแบบย้อนหลัง โดยนำเฉพาะข้อมูลกราฟมาจากเว็บไซต์ Kibana

3.2 แผนภาพแสดงลำดับเหตุการณ์ (Sequence Diagram)

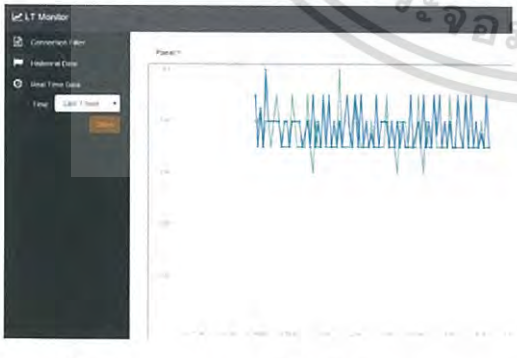


รูปที่ 3. แผนภาพแสดงลำดับเหตุการณ์

4.ผลการออกแบบ และ ผลการทำงานของระบบ

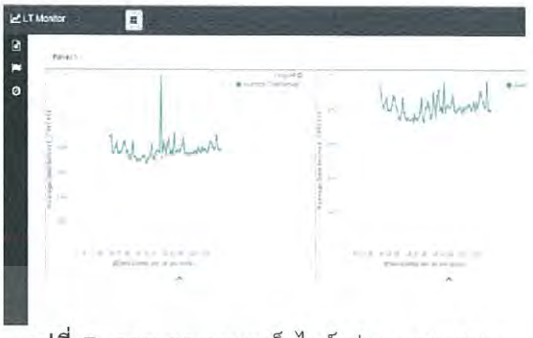
4.1 ผลการทำงานของเว็บไซต์

การแสดงผลบนเว็บไซต์ ผู้ใช้สามารถเลือกรูปแบบการแสดงผลได้ทั้งแบบ Compare คือเปรียบเทียบหลายโปรเซสในกราฟเดียว และแบบ Multiple Graph คือดูหลายกราฟโดยการเลือกรูปแบบเวลาสามารถเลือกได้ทั้งกำหนดเวลาที่แน่นอน และเวลาย้อนหลัง 1-4 ชั่วโมงย้อนหลัง



รูปที่ 4. การแสดงผลบนเว็บไซต์รูปแบบ Compare

จากภาพแสดงกราฟแบบเปรียบเทียบซึ่งจะมีข้อมูลกราฟของ 2 โปรเซสที่ผู้ใช้เลือก เพื่อให้เห็นการเปรียบเทียบที่ชัดเจน

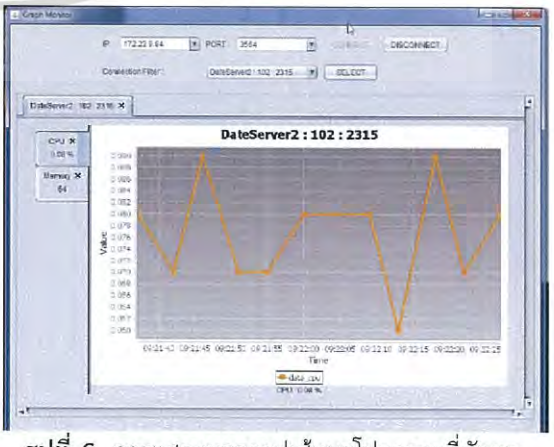


รูปที่ 5. การแสดงผลบนเว็บไซต์รูปแบบ Multiple Graph

จากภาพแสดงข้อมูลกราฟของ 2 โปรเซสแบบแยกกันทำให้มาสารถเห็นภาพรวมของข้อมูลมากขึ้น

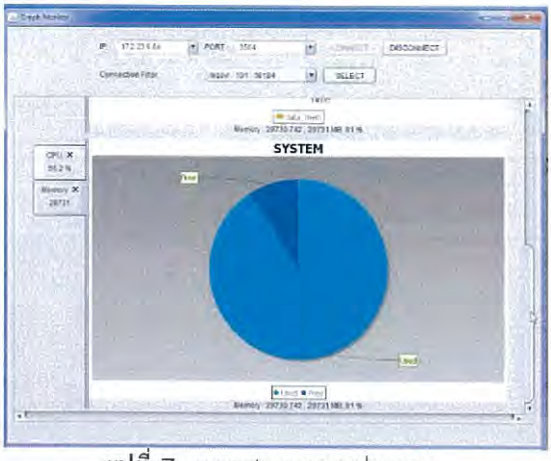
4.2 ผลการทำงานของ Java Application

การทำงานบนโปรแกรมซึ่งพัฒนาโดย JAVA จะเชื่อมต่อกับเซิฟเวอร์โดยใช้ IP, Port ที่ถูกกรอกโดยผู้ใช้ จากนั้นเมื่อมีการเชื่อมต่อสำเร็จ โปรแกรมจะแสดงรายชื่อโปรเซสต่างๆ ที่รันอยู่บนเซิฟเวอร์เครื่องนั้นเพื่อให้ผู้ใช้เลือกโปรเซสที่จะแสดงผลเป็นกราฟ ซึ่ง System และ Disk จะแสดงผลได้ทั้งกราฟเส้นและกราฟวงกลม ส่วนโปรเซสต่างๆจะแสดงผลได้เพียงกราฟเส้น และแสดงผลอัพเดทข้อมูลทุก 4 วินาที เมื่อเลือกโปรเซสจะแสดงกราฟทั้งข้อมูล CPU, Memory ในรูปแบบของกราฟเส้น และเมื่อเลือก System, Disk โปรแกรมจะสามารถแสดงทั้งกราฟเส้นและกราฟวงกลม และขณะเดียวกันโปรแกรมจะสร้างไฟล์ Log เพื่อเป็นข้อมูล Input ให้กับการแสดงผลที่จะเกิดขึ้นบนเว็บไซต์ด้วย



รูปที่ 6. การแสดงผลกราฟเส้นบนโปรแกรมที่พัฒนาโดยJava

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7. การแสดงผลกราฟวงกลม

5 สรุปผลโครงงาน

การพัฒนาการพัฒนาส่วนแสดงผลการใช้งานทรัพยากรของระบบซื้อขายเงินตราต่างประเทศนี้มีวัตถุประสงค์เพื่อสังเกตการณ์, ตรวจสอบการใช้ทรัพยากร (CPU, Memory) ของแต่ละเซิฟเวอร์ ซึ่งในเซิฟเวอร์แต่ละเครื่องจะมีโปรเซสถูกใช้งานอยู่จำนวนมาก หากมีการทำงานผิดปกติเกิดขึ้น ผู้ใช้สามารถตรวจสอบได้ในขณะนั้นว่าเกิดความผิดปกติขึ้นที่โปรเซสใด และเซิฟเวอร์จะมีการทำงานตลอดเวลา (24 ชั่วโมง) ดังนั้นผู้ใช้สามารถย้อนดูการใช้ทรัพยากรของโปรเซสในอดีตขณะที่ผู้ใช้ไม่ได้สังเกตการณ์การณ์หรือเมื่อผู้ใช้ไม่อยู่ได้ และผู้ใช้สามารถสังเกตการณ์ข้อมูลของกราฟจากการแสดงผลแบบเปรียบเทียบได้ผ่านเว็บไซต์ โดยการแสดงผลจะแสดงผลผ่านจาว่าแอพพลิเคชั่นที่เป็นการแสดงผลแบบเรียลทามและเว็บแอพพลิเคชั่นที่เป็นการแสดงผลกราฟทั้งแบบเรียลทาม (Real time data) และย้อนหลัง (Historical data)

เอกสารอ้างอิง

[1] "**Basic Concepts**".2015. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/_basic_concepts.html

[2] Somkiat Puisungnoen. **"Elasticsearch กับการจัดการข้อมูล**"2014. [Online]. Available: http://www.somkiat.cc/elasticsearch-your-data-flow/.

[3] Somkiat Puisungnoen. **"Logstash"**.2014. [Online]. Available:: http://www.somkiat.cc/manage-log-message-with-logstash/.

[4]" **Kibana 4 Video Tutorials, Part 2**".2015. [Online]. Available: https://www.elastic.co/blog/kibana-4-video-tutorials-part-2.

[5] **"Kibana 4 Video Tutorials, Part 4**". 2015. [Online]. Available: https://www.elastic.co/blog/kibana-4-video-tutorials-part-4.

[4]"**Skip header Line**".2015. [Online]. Available: https://discuss.elastic.co/t/skip-header-line-in-csv-input-v-1-5-0/1386.

[5]"**Elasticsearch-head**".2015. [Online]. Available: https://github.com/mobz/elasticsearch-head.

[6]"**JFreeChart**".2015. [Online]. Available: http://www.tutorialspoint.com/jfreechart/jfreechart_timeseries_chart.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้