

เว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูลการเรียกใช้ข้อมูลของลูกค้า

Web Application for Data Analysis of Customer Requested Data



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Application for Data Analysis of Customer Requested Data



A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
(COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา เว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูลการใช้ข้อมูลของลูกค้า
 Web Application for Data Analysis of Customer Requested Data

ชื่อนักศึกษา นายสหภาพ ศรีสว่างเนตร รหัสนักศึกษา 58050397

ปริญญา วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2561

อาจารย์ที่ปรึกษา ผศ.วิสันต์ ตั้งวงษ์เจริญ

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้
 สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการ
 คอมพิวเตอร์) ประจำปีการศึกษา 2561

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.อนันตพร หรรษคุณาฒย ประธานกรรมการ	อนันตพร หรรษคุณาฒย
ผศ.วิสันต์ ตั้งวงษ์เจริญ กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	เว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูลการใช้ข้อมูลของลูกค้า Web Application for Data Analysis of Customer Requested Data
ชื่อนักศึกษา	นายสหภาพ ศรีสว่างเนตร รหัสนักศึกษา 58050397
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2561
อาจารย์ที่ปรึกษา	ผศ.วิสันต์ ตั้งวงษ์เจริญ

บทคัดย่อ

บริษัท รีฟินิทฟ (ประเทศไทย) จำกัด เป็นบริษัทที่ให้บริการข้อมูลทางการเงินกับลูกค้าทั่วโลก โดยหนึ่งในระบบที่ให้บริการข้อมูลการซื้อขายหุ้นต่างๆในตลาดโลก คือ ระบบที่มีชื่อว่า Elektron Tick History (ETH) จะมีการเก็บข้อมูลการซื้อขายของตลาดหุ้นทั่วโลก เป็นข้อมูลขนาดใหญ่ตั้งแต่ปี 1996 แต่ปัจจุบันผู้ดูแลระบบไม่สามารถทราบปริมาณและรายละเอียดของการขอข้อมูลจากลูกค้าที่ทำงานอยู่ภายในระบบ ETH ณ เวลาปัจจุบันได้ ส่งผลให้ผู้ดูแลระบบไม่สามารถทราบถึงสถานการณ์ที่อาจก่อให้เกิดปัญหาต่อระบบที่กำลังเกิดขึ้นนักพัฒนาจึงได้พัฒนาเว็บแอปพลิเคชันเพื่อให้ผู้ดูแลระบบสามารถทราบถึงปัญหาดังกล่าวได้นอกจากนั้นสามารถตรวจสอบประวัติปริมาณการขอข้อมูลจากลูกค้าทั้งหมดบนระบบ ณ เวลาต่างๆมาแสดงในรูปแบบของแผนภูมิเส้นเพื่อให้ผู้ดูแลระบบสามารถนำข้อมูลไปวิเคราะห์พฤติกรรมกรขอข้อมูลของลูกค้าได้

โดยการพัฒนาอยู่ภายใต้ข้อจำกัดที่ว่ากรนำข้อมูลรายละเอียดการขอข้อมูลของลูกค้าไปใช้ต้องมีการรบกวนการทำงานของฐานข้อมูลหลักของบริษัทให้น้อยที่สุด การพัฒนาเว็บแอปพลิเคชันผู้พัฒนาต้องพัฒนาส่วนต่างๆ ซึ่งประกอบด้วย เว็บเซิร์ฟวิส โดยใช้ เฟลคเฟรมเวิร์ค ในการพัฒนาฐานข้อมูลเก็บประวัติปริมาณการขอข้อมูลจากลูกค้า ณ เวลาต่างๆและ ฐานข้อมูลเก็บรายละเอียดการขอข้อมูลของลูกค้าจากฐานข้อมูลหลักของบริษัท โดยใช้ MySQL Databaseในการบันทึก และชุดคำสั่งที่ใช้การดึงข้อมูลรายละเอียดการขอข้อมูลของลูกค้าจากฐานข้อมูลหลักของบริษัท มาเก็บไว้ในฐานข้อมูลบนเครื่องเดียวกับเว็บแอปพลิเคชันโดยใช้ภาษาไพธอนในการพัฒนาจากผลการพัฒนาทำให้ผู้ดูแลระบบสามารถลดกระบวนการตรวจสอบสถานะของระบบ ETH ณ ปัจจุบันได้และการประเมินความพึงพอใจของผู้ดูแลระบบได้ค่าเฉลี่ยที่ 4.62 อยู่ในระดับดีมาก

คำสำคัญ : ข้อมูลขนาดใหญ่ เฟลคเฟรมเวิร์ค เว็บเซิร์ฟวิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Web Application for Data Analysis of Customer Requested Data	
Student	Mr. Sahaparb Srisawangnet	Student ID 58050397
Degree	Bachelor of Science (Computer Science)	
Department	Computer Science	
Faculty	Science	
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)	
Academic Year	2018	
Advisor	Asst.Prof.WisanTangwongcharoen	

Abstract

Refinitiv (Thailand) Ltd. is a company providing financial information to customer worldwide. The one of Refinitiv Software System providing global financial information is Elektron Tick History (ETH) which had been record exchange on stock market data on the system into main database of company every times. In working, analyst cannot realize about volume of requested data and requested data detail is existing the system in currently. The result of analyst cannot aware to situations that might cause system crashes. Developer had developed web application for analyst can aware to above issues. In addition, analyst can verify the volume of all requested data on the system at various times.

The development is subject to restriction on query requested data detail from main database of the company to a minimum. In developing, Developer divide to many parts that consist of a web service using the Flask framework to develop, the database contains the volumes of historical requested data at various times, the database contain requested data detail from main database of the company that using MySQL to record and the script use to query requested data detail from main database of the company to above database in same place with web service that script is developed by python language. The result of development effect to system administrators can recognized about ETH status in currently. The assessment of the satisfaction of Administrator receiving the average score at 4.62 that is very good.

Keyword : big data , flask framework , web Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

จากการที่ข้าพเจ้าได้ทำสหกิจศึกษาหัวข้อเครื่องมือเว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูล การใช้ข้อมูลของลูกค้าบริษัททอมสันรอยเตอร์สามารถดำเนินงานสำเร็จได้ด้วยดีเนื่องจากได้รับความช่วยเหลือการสนับสนุนและความกรุณาจากองค์กรและบุคคลหลายท่านคณะผู้จัดทำขอกล่าวคำขอบคุณองค์กรและบุคคลดังต่อไปนี้

ขอขอบพระคุณอาจารย์ภาควิชาวิทยาการคอมพิวเตอร์คณะวิทยาศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้ความรู้และประสบการณ์ตลอดระยะเวลาที่ได้รับการศึกษา

ขอขอบพระคุณนายวรัทมัยชัยเกษตรสินและบุคลากรในทีม Time Series ทุกคนผู้ดูแลและให้คำปรึกษาข้าพเจ้าขณะฝึกงานอยู่ที่บริษัทรอยเตอร์ซอฟต์แวร์ (ประเทศไทย) จำกัด

ขอขอบพระคุณบริษัทรอยเตอร์ซอฟต์แวร์ (ประเทศไทย) จำกัดที่ให้โอกาสในการเข้ารับการฝึกงานและได้รับประสบการณ์ในการทำงานในสภาพแวดล้อมจริง

สหภาพ ศรีสว่างเนตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการสหกิจศึกษา.....	2
1.3 ขอบเขตของโครงการสหกิจศึกษา.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 เครื่องมือและภาษาที่ใช้ในโครงการสหกิจศึกษา.....	3
บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการพัฒนาระบบ.....	5
2.1 กระบวนการ ETL.....	5
2.2 โครงสร้างและการทำงานของระบบ Elektron Tick History (ETH).....	7
2.3การทำงานของ Web Service.....	10
2.4 เฟรมเวิร์คและไลบรารีที่ใช้ในการพัฒนาระบบ.....	13
2.4.1) การทำงานของ FLASK Framework.....	13
2.4.2) การทำงานของ amCharts.....	16
2.4.3) การทำงานของ DataTables CDN.....	18
2.4.4) การทำงานของ HADOOP.....	20
2.5 ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ.....	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.5.1) โปรแกรม Visual Studio Code.....	23
2.5.2) โปรแกรม Cisco AnyConnect Secure Mobility Client.....	23
2.5.3) โปรแกรม Trello	25
2.5.4) โปรแกรม Apache	27
บทที่ 3 ขั้นตอนในการพัฒนาระบบ.....	29
3.1 การวิเคราะห์และออกแบบระบบของเว็บแอปพลิเคชัน.....	29
3.1.1) การวิเคราะห์แหล่งข้อมูลสำหรับเชื่อมต่อ Web Service	31
3.1.2) การวิเคราะห์และออกแบบฐานข้อมูลสำหรับเชื่อมต่อ Web Service	34
3.1.3) การวิเคราะห์และออกแบบชุดคำสั่งสำหรับนำข้อมูลบันทึกการใช้งาน Client usage Database	38
3.1.4) การวิเคราะห์และออกแบบ Web Service.....	41
3.2 การพัฒนาระบบเว็บแอปพลิเคชัน	55
บทที่ 4 ผลการดำเนินงาน.....	57
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....	72
5.1 สรุปผลการดำเนินงาน.....	72
5.2 ข้อจำกัด	72
5.3 ข้อเสนอแนะ.....	73
เอกสารอ้างอิง	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3.1 Data descriptions ของ Client usage Database	35
ตารางที่ 3.2 Data descriptions ของ Job Status Database.....	37
ตารางที่ 4.1 เปรียบเทียบเวลาที่ใช้ในการดึงข้อมูลระหว่างจากฐานข้อมูลหลัก และ Client usage Database.....	69
ตารางที่ 4.2 เปรียบเทียบข้อมูลที่ถูกดูแลระบบสามารถนำมาวิเคราะห์ระหว่างก่อนและหลังมีการ พัฒนาเว็บแอปพลิเคชัน	70
ตารางที่ 4.3 ประเมินผลการพัฒนาเว็บแอปพลิเคชัน	71



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 กระบวนการ ETL	7
รูปที่ 2.2 โครงสร้างและการทำงานของระบบ Elektron Tick History (ETH).....	8
รูปที่ 2.3 Web Service Model	11
รูปที่ 2.4 การทำงานของ Web Service.....	12
รูปที่ 2.5 Logo ของ Werkzeug	13
รูปที่ 2.6 Logo ของ Jinja.....	13
รูปที่ 2.7 Flask Framework Model	14
รูปที่ 2.8 ตัวอย่างการเขียนโปรแกรมด้วย Flask Framework	15
รูปที่ 2.9 Logo ของ Flask Framework.....	15
รูปที่ 2.10 ตัวอย่างผลลัพธ์ที่ได้จากการใช้ amChart สร้างแผนภูมิข้อมูล	17
รูปที่ 2.11 ตัวอย่างการใช้งาน amCharts	17
รูปที่ 2.12 Content Delivery Network Model	18
รูปที่ 2.13 ตัวอย่างผลลัพธ์ที่ได้จากการใช้ DataTable CDN สร้างตารางข้อมูล	19
รูปที่ 2.14 ตัวอย่างการใช้งาน DataTables CDN.....	19
รูปที่ 2.15 อธิบายส่วนประกอบของ URI.....	20
รูปที่ 2.16 ระบบนิเวศของ Hadoop.....	21
รูปที่ 2.17 สถาปัตยกรรมของ Hadoop.....	22
รูปที่ 2.18 Logo ของ HADOOP	22
รูปที่ 2.19 Logo ของ Visual Studio Code	23
รูปที่ 2.20 รูปแบบการทำงานของ Cisco AnyConnect VPN.....	24
รูปที่ 2.21 Logo ของ Cisco AnyConnect Secure Mobility Client	24
รูปที่ 2.22 ตัวอย่างโปรแกรม Trello	26
รูปที่ 2.23 Logo ของ Trello.....	26
รูปที่ 2.24 การทำงานภายใน Apache	28
รูปที่ 2.25 Logo ของ Apache	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 3.1 ตัวอย่างรายละเอียดของ Requested Data ที่กำลังทำงานในระบบ ETH จากได้รับจาก YARN Resource Manager REST API.....	32
รูปที่ 3.2 ER Diagram ของ Client Usage Table	34
รูปที่ 3.3 ER Diagram ของ Client Usage Table.....	36
รูปที่ 3.4 การวิเคราะห์การทำงานของ Daily Script และการนำไปใช้	38
รูปที่ 3.5 Sequence Diagram การทำงาน Daily Script.....	39
รูปที่ 3.6 สถาปัตยกรรมของ Web Applications for Data Analysis of Customer Requested Data.....	41
รูปที่ 3.7 Use Case Diagram การทำงานเว็บแอปพลิเคชัน.....	42
รูปที่ 3.8 Sequence Diagram ของการเข้าสู่เว็บแอปพลิเคชัน	43
รูปที่ 3.9 Sequence Diagram ของการขอรายละเอียด Requested Data.....	44
รูปที่ 3.10 Sequence Diagram ของการขอข้อมูลปริมาณตลาดภายใน Requested Data.....	46
รูปที่ 3.11 Sequence Diagram ของการขอความไหลลดข้อมูลภายใน Requested Data.....	47
รูปที่ 3.12 Sequence Diagram ของการเก็บประวัติปริมาณ Requested Data	48
รูปที่ 3.13 Activity Diagram ของการเข้าสู่เว็บแอปพลิเคชัน	49
รูปที่ 3.14 Activity Diagram ของการขอรายละเอียด Requested Data.....	50
รูปที่ 3.15 Activity Diagram ของการดูรายละเอียด Requested Data ทั้งหมด.....	51
รูปที่ 3.16 Activity Diagram ของการดูรายละเอียดการใช้งานระบบของลูกค้า.....	52
รูปที่ 3.17 Activity Diagram ของการขอข้อมูลภายใน Requested Data	53
รูปที่ 3.18 แผนผังองค์ประกอบการพัฒนาเว็บแอปพลิเคชัน	55
รูปที่ 4.1 สถาปัตยกรรมของ Web Application for Data Analysis of Customer Requested Data.....	57
รูปที่ 4.2 หน้าการรอเข้าสู่เว็บแอปพลิเคชัน	58
รูปที่ 4.3 หน้าการใช้งานหลักของเว็บแอปพลิเคชัน	58
รูปที่ 4.4 หน้าการใช้งานหลักของเว็บแอปพลิเคชัน (ต่อ).....	59
รูปที่ 4.5 การเปิด/ปิดการดึงข้อมูลและการแสดงแผนภูมิเส้น.....	59

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อใช้ในการศึกษาวิจัยเท่านั้น ไม่สามารถนำ
 ไปทำซ้ำหรือเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารได้ หากมีข้อผิดพลาดประการใด
 ขออภัยเป็นอย่างสูง และขอสงวนสิทธิ์ในข้อมูลและข้อมูลอ้างอิงที่ปรากฏในเอกสารฉบับนี้
 ไม่สามารถแก้ไขได้ ทั้งนี้หากมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 4.7 ตัวอย่างการแสดงผลรายละเอียดทั้งหมดของ Requested Data บน Pop-up.....	61
รูปที่ 4.8 ตัวอย่างการค้นหาข้อความที่ตรงกับรายละเอียดของ Requested Data.....	61
รูปที่ 4.9 ตัวอย่างการดาวน์โหลดรายละเอียดทั้งหมดของ Requested Data	62
รูปที่ 4.10 ตัวอย่างรายละเอียด Requested Data ภายในไฟล์ CSV	63
รูปที่ 4.11 ตัวอย่างรายละเอียด Requested Data ภายในไฟล์ CSV (ต่อ)	63
รูปที่ 4.12 ตัวอย่างการแสดงผลปริมาณการใช้งานของผู้ใช้งานในรูปแบบแผนภูมิเส้น.....	63
รูปที่ 4.13 ตัวอย่างแผนภูมิเส้นแสดงผลปริมาณ Requested Data ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด.....	64
รูปที่ 4.14 ตัวอย่างแผนภูมิเส้นแสดงขนาดข้อมูลที่ขอ (RIC DAY) ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด.....	64
รูปที่ 4.15 ตัวอย่างแผนภูมิเส้นแสดงค่าเฉลี่ยเวลาของ Requested Data ที่ใช้ในระบบ ETH ทั้งหมดของ ผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด.....	65
รูปที่ 4.16 ตัวอย่างการแสดงผลปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาสุดท้ายที่ขอบน Requested Data	65
รูปที่ 4.17 ตัวอย่างการแสดงผลระหว่างรอ Web Service ดึงข้อมูลปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาที่ สุดท้ายที่ขอ Requested Data	66
รูปที่ 4.18 ตัวอย่างการแสดงผลปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาที่สุดท้ายที่ขอ Requested Data หลัง ได้รับข้อมูลจาก Web Service.....	67
รูปที่ 4.19 ตัวอย่างการขอข้อมูลภายใน Requested Data ภายในไฟล์ JSON.....	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

บริษัท Refinitiv เป็นบริษัทที่ให้ข้อมูลทางการเงินกับลูกค้าทั่วโลก ซึ่งความต้องการข้อมูลของลูกค้าในแต่ละวันนั้นมีปริมาณมาก ดังนั้นระบบที่ให้บริการข้อมูลกับลูกค้าจำเป็นต้องมีความสามารถรองรับจำนวนข้อมูลและการทำงานของลูกค้ายิ่งในปริมาณมากได้ ซึ่งหนึ่งในระบบที่ให้บริการข้อมูลการซื้อขายหุ้นต่างๆในตลาดโลก คือ ระบบที่มีชื่อว่า Elektron Tick History (ETH) ซึ่งจะมีการเก็บข้อมูลขนาดใหญ่ตั้งแต่ปี ค.ศ. 1996 และประมวลผลการเรียกใช้ข้อมูลของลูกค้า (Requested Data) บน HADOOP Cluster เมื่อระบบประมวลผลข้อมูลของลูกค้าสำเร็จจะนำผลลัพธ์ที่ได้เก็บไว้ที่ Amazon Web Services(AWS) โดยระบบจะจัดเก็บรายละเอียดของ Requested Data เช่น จำนวนหลักทรัพย์กับเวลาของระบบที่ใช้ไปรวมถึงยูอาร์แอล (URL) ซึ่งเป็นที่อยู่ผลลัพธ์ของข้อมูลที่ลูกค้าต้องการบน AWS นำไปเก็บไว้ที่ฐานข้อมูลหลักของบริษัท (Oracle Database) ต่อมาระบบจึงทำการส่ง URL ของ AWS ให้ลูกค้าสามารถดาวน์โหลดข้อมูลไปใช้งาน

การประมวลผลการเรียกใช้ข้อมูลของลูกค้าภายในระบบElektron Tick History (ETH)ในปัจจุบันผู้ดูแลระบบไม่สามารถทราบปริมาณและรายละเอียดของ Requested Data ที่ทำงานบนระบบ ETH ณ เวลาปัจจุบันได้ ซึ่งส่งผลให้ผู้ดูแลระบบไม่สามารถทราบถึงสถานการณ์ที่อาจก่อให้เกิดปัญหาต่อระบบที่กำลังเกิดขึ้นได้ เช่น มีปริมาณ Requested Data รอกการทำงานมากกว่าปกติ หรือมี Requested Data บางตัวที่รอกการทำงานมานานผิดปกติ เป็นต้น ดังนั้นจึงต้องมีการพัฒนาเว็บแอปพลิเคชันที่ผู้ดูแลระบบสามารถใช้ในการวิเคราะห์ปริมาณ Requested Data สามารถตรวจสอบรายละเอียดของ Requested Data ณ เวลานั้นได้และสามารถแสดงประวัติปริมาณ Requested Data ของเวลาที่ผ่านไปแล้วในรูปแบบของแผนภูมิเส้น รวมถึงค่าเฉลี่ยหรือข้อมูลสถิติต่างๆ เพื่อใช้วิเคราะห์สถานการณ์ที่กำลังเกิดขึ้นบนระบบต่อไปนี้

- 1) วิเคราะห์ปริมาณ Requested Data ที่สูงผิดปกติ
- 2) วิเคราะห์ข้อมูลการใช้งานของลูกค้าที่ส่ง Requested Data ที่มีขนาดใหญ่หลายๆ มาบ่อยครั้ง
- 3) วิเคราะห์หาความสัมพันธ์ระหว่างข้อมูลที่ขอกับเวลาที่ระบบ Elektron Tick History ใช้ในการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงการสหกิจศึกษา

- 1) เพื่อใช้ในการตรวจสอบปริมาณ Requested Data ที่อยู่บน HADOOP Cluster ณ ปัจจุบัน
- 2) เพื่อตรวจสอบรายละเอียด Requested Data ที่อยู่บน HADOOP Cluster ณ ปัจจุบัน
- 3) เพื่อเก็บประวัติปริมาณ Requested Data มาทำ Data Visualization เพื่อช่วยในการวิเคราะห์
- 4) เพื่อตรวจสอบประวัติการใช้งานของลูกค้า ที่เป็นเจ้าของ Requested Data ที่กำลังทำงานบน HADOOP Cluster ณ ปัจจุบัน
- 5) เพื่อตรวจสอบข้อมูลตลาดที่ขอมายภายใน Requested Data ที่กำลังทำงานบน HADOOP Cluster ณ ปัจจุบัน

1.3 ขอบเขตของโครงการสหกิจศึกษา

- 1) การตรวจสอบปริมาณของ Requested Data ที่กำลังทำงานอยู่บน HADOOP Cluster ของระบบ Elektron Tick History
 - 1.1. แสดงปริมาณ Requested Data แยกตามเซิร์ฟเวอร์ที่ทำงาน
 - 1.2. แสดงปริมาณ Requested Data แยกตามสถานะบน HADOOP Cluster โดยแบ่งเป็น กำลังรอการทำงานและกำลังทำงานอยู่
 - 1.3. แสดงปริมาณ Requested Data แยกตามประเภทของ Requested Data บน HADOOP Cluster โดยแบ่งเป็น Regular , Large และ HMC
 - 1.4. สามารถเปิดหรือปิด การแสดงผลปริมาณ Requested Data ในแต่ละประเภทได้
- 2) การแสดงแผนภูมิเส้นประวัติปริมาณ Requested Data เป็นระยะ 1 เดือนล่าสุด
- 3) การตรวจสอบรายละเอียดของ Requested Data ที่กำลังทำงานอยู่บน HADOOP Cluster ของระบบ Elektron Tick History
 - 3.1. แสดงรายละเอียดของ Requested Data ในรูปแบบของตารางโดยแยกตามประเภทของ Requested Data บน HADOOP Cluster โดยแบ่งเป็น Regular , Large และ HMC
 - 3.2. สามารถดาวน์โหลดรายละเอียดของ Requested Data ในรูปแบบไฟล์ CSVหรือ EXCEL
- 4) การตรวจสอบประวัติการใช้งานระบบของลูกค้าที่มี Requested Data กำลังทำงานอยู่บน HADOOP Cluster ของระบบ Elektron Tick History ในรูปแบบของแผนภูมิเส้น
 - 4.1. แสดงแผนภูมิเส้นของ ปริมาณ Requested Data ที่ลูกค้าขอเข้ามาในแต่ละวัน ในช่วง 30 วันล่าสุด
 - 4.2. แสดงแผนภูมิเส้นของ ขนาดของข้อมูลที่ลูกค้าขอเข้ามาในแต่ละวัน ในช่วง 30 วันล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปไซ้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.3. แสดงแผนภูมิเส้นของ เวลาเฉลี่ยของ Requested Data ใช้งานบนระบบ ETH ที่ลูกค้า
ขอเข้ามาในแต่ละวัน ในช่วง 30 วันล่าสุด
- 5) การตรวจสอบข้อมูลตลาดที่ลูกค้าขอมาภายใน Requested Data
 - 5.1. แสดงปริมาณตลาดที่ขอทั้งหมด
 - 5.2. แสดงช่วงเวลาเริ่มขอ และช่วงเวลาสุดท้ายที่ขอ
 - 5.3. สามารถดาวน์โหลดรายละเอียดภายใน Requested Data

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ผู้ดูแลระบบสามารถทราบถึงสถานการณ์ที่เกิดขึ้นกับระบบได้อย่างทันที
- 2) ช่วยผู้ดูแลระบบตรวจสอบหาสาเหตุของปัญหาที่เกิดขึ้นกับระบบ
- 3) ช่วยลดข้อผิดพลาดในการวิเคราะห์ข้อมูล
- 4) ช่วยลดขั้นตอนในการวิเคราะห์ข้อมูล
- 5) ช่วยให้สามารถวิเคราะห์ข้อมูลเชิงลึกได้

1.5 เครื่องมือและภาษาที่ใช้ในโครงการสหกิจศึกษา

- 1) ฮาร์ดแวร์
 - 1.1. เครื่องคอมพิวเตอร์
 - 1.1.1.หน่วยประมวลผลกลาง (CPU) Intel(R) Core(TM) i7-3720QM @2.60 GHz
 - 1.1.2.หน่วยความจำหลัก(RAM) 8 GB
 - 1.1.3.หน่วยความจำสำรอง 240 GB
 - 1.1.4.การ์ดจอ(Graphic processor) NVIDIA NVS 5200M
- 2) ซอฟต์แวร์
 - 2.1. Visual Studio Code ใช้ในการเขียนโปรแกรม
 - 2.2. MySQL ใช้เป็นฐานข้อมูลที่ใช้เก็บรายละเอียด Requested Dataใช้เป็นฐานข้อมูลที่ใช้
ในการเก็บจำนวน Requested Data เพื่อแสดงแผนภูมิเส้น
 - 2.3. MySQL Workbench ใช้เป็น Tool ในการจัดการฐานข้อมูล MySQL Server
 - 2.4. Cisco AnyConnect Secure Mobility Client ใช้ในการเชื่อมต่อ Virtual Private
Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ภาษา

- 3.1. Python 2.7 ใช้ในการพัฒนา Scripts และ Web Service
- 3.2. HTML ใช้ในการพัฒนา Template ของเว็บแอปพลิเคชัน
- 3.3. CSS ใช้ในการพัฒนา Template ของเว็บแอปพลิเคชันร่วมกับ HTML
- 3.4. JAVASCRIPT ใช้ในการพัฒนา Templateของเว็บแอปพลิเคชันและส่งRequestไปยัง Web Service



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการพัฒนาระบบ

ในการพัฒนาเว็บแอปพลิเคชันที่ใช้ในการแสดงปริมาณ Requested Data ตรวจสอบรายละเอียดของ Requested Data ณ เวลาปัจจุบัน และรวมถึงค่าสถิติของข้อมูล ผู้พัฒนาได้ศึกษาหลักการทฤษฎีต่างๆและเทคโนโลยีที่เกี่ยวข้องที่สามารถนำไปประยุกต์ใช้งาน พร้อมทั้งนำความรู้ที่ได้ไปใช้ควบคู่กับการพัฒนาเว็บแอปพลิเคชันโดยแบ่งเป็นหัวข้อต่างๆ ดังต่อไปนี้

- 2.1) กระบวนการ ETL
- 2.2) โครงสร้างและการทำงานของระบบ Elektron Tick History (ETH)
- 2.3) การทำงานของ Web Service
- 2.4) เฟรมเวิร์คและไลบรารีที่ใช้ในการพัฒนาระบบ
- 2.5) ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

2.1 กระบวนการ ETL

ETL (Extract-Transform-Load) คือกระบวนการหนึ่งในระบบ Data Warehouse โดยระบบที่ออกแบบเอาไว้จะดึงข้อมูลออกมาจากหลายๆ ที่, นำกระบวนการตรวจสอบคุณภาพของข้อมูลมาประยุกต์ใช้, มีการเชื่อมโยงและปรับข้อมูลให้เป็นไปในรูปแบบเดียวกันเพื่อให้ ข้อมูลจากหลายๆ แหล่งสามารถใช้งานร่วมกันได้ และท้ายที่สุดทำการส่งมอบ (Delivery) ข้อมูลเหล่านั้นในรูปแบบที่ง่ายต่อการใช้งาน เพื่อใช้ในการตัดสินใจขององค์กรโดยมีกระบวนการหลักๆ ที่เกี่ยวข้องกับกระบวนการต่างๆ ดังต่อไปนี้

2.1.1) Extract (กระบวนการดึงข้อมูล)

ขั้นตอนการเริ่มต้นของระบบ ETL จาก แหล่งของข้อมูล โดยทั่วไปแล้วระบบ Data warehouse จะประกอบด้วย ข้อมูลจากหลายๆ ที่ ข้อมูลที่อยู่ต่างที่กันนั้น อาจจะอยู่ในรูปแบบที่แตกต่างกันด้วยยกตัวอย่างเช่น อาจอยู่ในรูปแบบของฐานข้อมูลคนละชนิด หรือ ไม่ใช่ฐานข้อมูลแท้จริงซึ่งอาจจะเป็นระบบไฟล์ข้อมูลธรรมดา (Flat Files) หรือ ในอีกกรณีคือเป็นข้อมูลในฐานข้อมูลที่ไม่ใช่ระบบ RDBMS (Relational Database System) เช่น ข้อมูลในรูปแบบของ Information Management System (IMS) หรือ ข้อมูลอื่นใดนอกเหนือระบบเช่น Virtual Storage Access Method (VSAM) หรือ Indexed Sequential Access Method (ISAM) หรือแม้กระทั่งการดึงจากข้อมูลรูปแบบ อื่นๆ ที่ดึงกันสด ๆ จากระบบอื่นเลยเช่น Web Spider หรือ Screen scraping ข้อมูลจะถูกดึงเข้าสู่ระบบแบบ on-the-fly เป้าหมายของกระบวนการ Extract นี้คือดึงข้อมูลเข้ามาสู่รูปแบบ

เอกสารนี้มีมาตรฐานเดียวกัน เพื่อให้เหมาะสมต่อการแปลงรูปร่างของข้อมูลในขั้นตอนถัดไป เนื้อหาของเอกสารนี้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการนี้จะยังเกี่ยวกับกระบวนการอ่านข้อมูลที่ถูกต้องเข้าสู่ระบบ ส่งผลให้เกิดกระบวนการตรวจสอบข้อมูลว่า ตรงตามรูปแบบที่กำหนดไว้เบื้องต้นหรือไม่ ทั้งรูปแบบของข้อมูล และ โครงสร้างของข้อมูล ซึ่งถ้าหากเกิดข้อมูลนั้นไม่ได้มาตรฐานหรือรูปแบบที่กำหนดเอาไว้เบื้องต้น ก็จะทำให้เกิดกระบวนการปฏิเสธข้อมูลนั้น

2.1.2) Transform (กระบวนการแปลงข้อมูล)

ขั้นตอนการแปลงข้อมูลนี้จะมีการใช้กฎหรือฟังก์ชัน(Function) มากมายเพื่อที่จะแปลงข้อมูลให้ได้ตามที่ต้องการก่อนที่จะ นำข้อมูลเข้าไปยังปลายทาง ข้อมูลจากต้นทางบางแหล่งข้อมูลมีความจำเป็นน้อยมากหรือแทบจะไม่ต้องการ การแปลงข้อมูลเลย แต่ในบางแหล่งอาจจะต้องการกระบวนการที่ซับซ้อน ซึ่งจะกินทรัพยากรระบบและเวลาในการประมวลผลทั้งนี้ ความซับซ้อนของข้อมูลนั้นขึ้นอยู่กับความต้องการของธุรกิจ หรือ เป้าหมายของการนำข้อมูลไปใช้งาน โดยมีกระบวนการตัวอย่างต่อไปนี้

- Selection คือ เลือกเฉพาะสดมภ์ (Column) ที่ต้องการที่จะ Load ยกตัวอย่างเช่น ถ้าต้นทางข้อมูลมี 3 Columns หรือจะเรียกว่ามี 3 attributes เช่น age และ salary จะมีการแปลงข้อมูล และ เลือกที่จะไม่มีการแปลงข้อมูลหากว่าสดมภ์ (column) salary มีค่าเป็น Null
- Translation คือ การแปลงข้อมูล ยกตัวอย่างเช่น หากข้อมูลต้นทางมีการเก็บข้อมูลเพศว่า 1 เป็นเพศชาย และ 2 เป็นเพศหญิง จะต้องมีการแปลงจากโค้ดที่กำหนดก่อนหน้านี้ให้ 1 = Male และ 2 = Female กระบวนการนี้เรียกว่า data cleaning หรือ กระบวนการทำความสะอาดข้อมูล
- Encoding free form ยกตัวอย่างเช่นการ mapping จาก “Male” ไปเป็น “1” และ “Mr” ไปเป็น “M”
- Deriving a new calculated value คือกระบวนการคำนวณเพื่อให้ได้ข้อมูลที่ต้องการ ยกตัวอย่างเช่น $\text{sale amount} = \text{qty} * \text{unit_price}$
- Filtering คือ กระบวนการกรองเฉพาะข้อมูลที่กำหนด
- Sorting คือ กระบวนการเรียงข้อมูลที่ต้องการ
- Joining คือ กระบวนการเชื่อมโยงข้อมูลระหว่างตารางข้อมูล
- Aggregation คือ กระบวนการรวบรวม และ สรุปชุดข้อมูล ยกตัวอย่างเช่น การรวมยอด (summarize) ข้อมูลจากหลาย ๆ ระเบียบจนได้มาเป็น ยอดขายรวม เป็นต้น
- Transposing or pivoting คือการสลับทิศทางตำแหน่งของการแสดงข้อมูล เช่นการย้ายระเบียบไปเป็นสดมภ์ หรือ ย้ายสดมภ์มาเป็นระเบียบ เพื่อให้ง่ายต่อการนำข้อมูลไปใช้
- Splitting column into multiple columns คือ กระบวนการหั่นข้อมูลออกเป็นสดมภ์ย่อย ยกตัวอย่างเช่นถ้าข้อมูลเป็นข้อมูลวันที่เช่น 2010/08/13 10:58:02 ก็จะมีการหั่นเป็นสดมภ์ย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ได้เป็น 2010,08,13,10,58,02

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

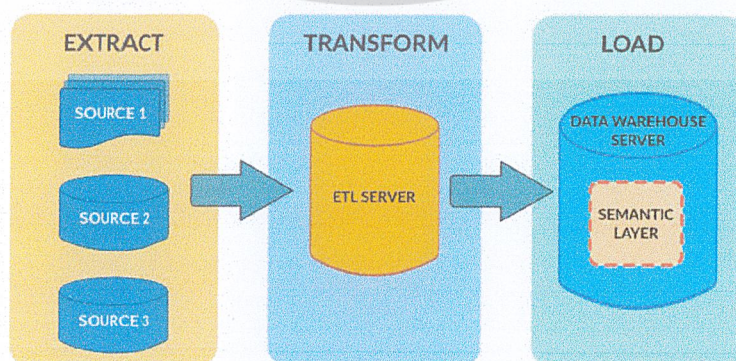
- Disaggregation คือ กระบวนการยกเลิกการรวมข้อมูล ยกตัวอย่างเช่นแทนที่จะเก็บข้อมูลเป็นระเบียบ แต่นำข้อมูลนั้นมาเชื่อมต่อกันเป็นระเบียบเดียว

2.1.3) Load (ขั้นตอนการนำข้อมูลเข้าสู่ระบบ)

ขั้นตอนการโหลดข้อมูลเข้า โดยทั่วไปจะนำข้อมูลเข้าไปในระบบ Data Warehouse ทั้งนี้ขึ้นอยู่กับความต้องการขององค์กร หรือ ธุรกิจว่าจะให้ข้อมูลไหลไปในทิศทางใด บางองค์กร หรือ บางงานจะมีการสะสมของข้อมูล ความถี่ของการนำข้อมูลเข้าสู่ระบบ อาจจะมีการล้างข้อมูลแล้วทับข้อมูลใหม่โดยทั่วไปแล้วข้อมูลของ Data Warehouse จะมีการใช้กันปีต่อปี เมื่อขึ้นปีใหม่แล้วจะมีการล้างข้อมูลของปีเก่า และ เก็บไว้ในระบบข้อมูลสำรอง เนื่องจากว่ากระบวนการนำข้อมูลเข้าจะต้องปฏิสัมพันธ์กับฐานข้อมูล (Database) ดังนั้นจะต้องมีประเด็นเรื่องของ Database Constraints, Referential Integrity, Database Trigger เข้ามาเกี่ยวข้อง ด้วยในกระบวนการนำข้อมูลเข้า ซึ่งสิ่งเหล่านี้รวม ๆ แล้วเรียกว่า กระบวนการควบคุมคุณภาพของข้อมูล (Data Quality performance of ETL process)

ความเสี่ยงของการใช้กระบวนการ ETL

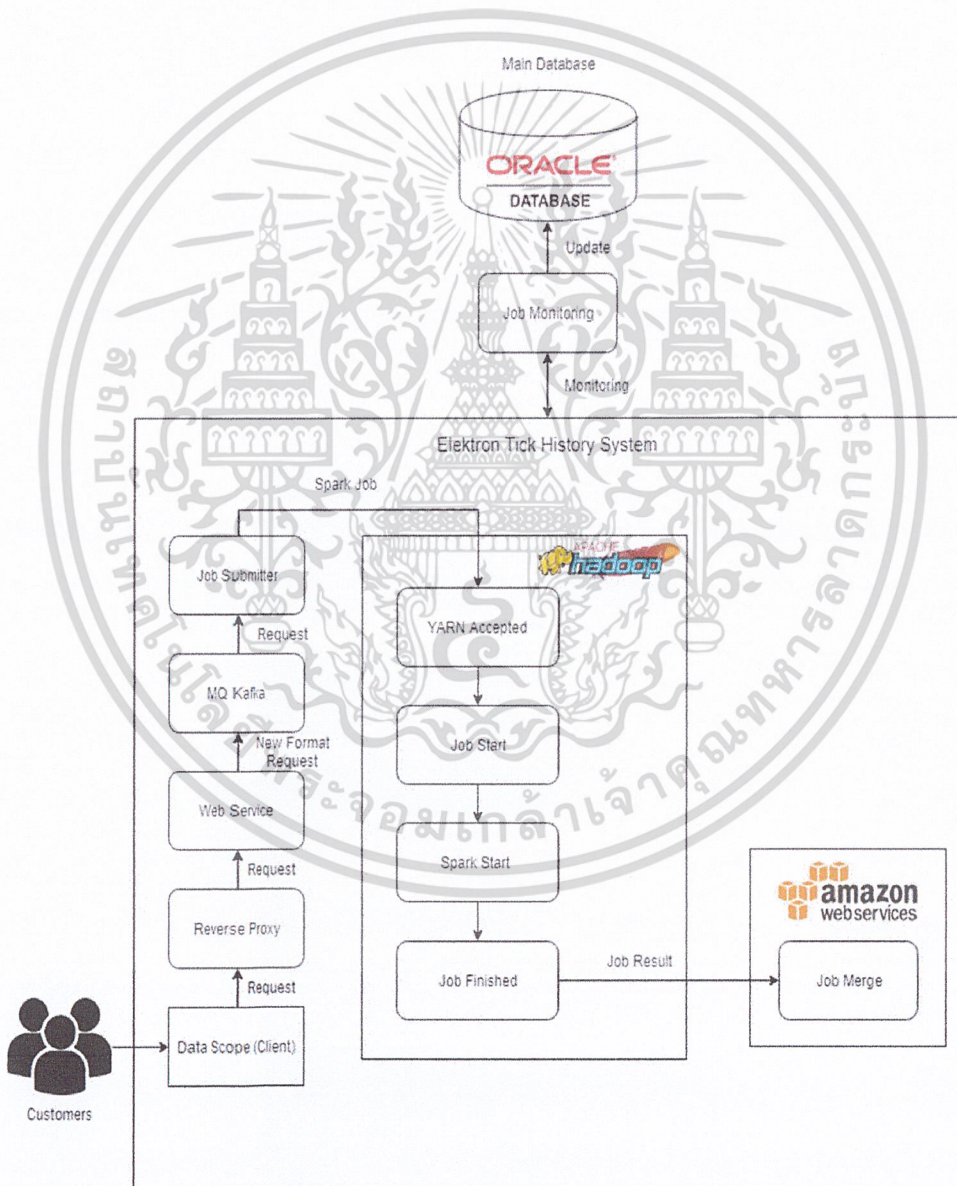
กระบวนการทำงานกับ ETL มีความสลับซับซ้อนและละเอียดอ่อนต่อข้อมูลมาก การทำงานอาจจะประสบปัญหาหนักหากใช้ระบบ ETL ที่ออกแบบมาไม่ดี รวมทั้งปริมาณข้อมูลอาจจะเกินกว่าที่ระบบจะรองรับได้ ซึ่งต้องใช้ผู้ออกแบบ (designer) มานั่งเสียเวลาตรวจสอบ ระบบ Data Warehouse เป็นระบบที่มีข้อมูลจากหลายๆแหล่งต้องปฏิสัมพันธ์ด้วย ระบบ ETL เป็นระบบเดียวที่เป็นตัวกลางในการประสานงานระบบต่างๆ เข้าด้วยกัน การออกแบบระบบ ETL เพื่อการใช้งานจะต้องคำนึงถึงความสามารถในการขยายขนาด (Scalability) ของ ETL ด้วย เพราะเมื่อปริมาณข้อมูลมีขนาดใหญ่แล้วจะมีปัญหาหรือไม่ ระบบ ETL บางระบบมีความสามารถในการรองรับข้อมูลกระจายไปในหลาย ๆ Hard disk ได้ และสามารถกระจายการทำงานของ batch ให้ไปทำงานแยกตามเครื่องอื่น ๆ เพื่อช่วยประมวลผลได้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.1 กระบวนการ ETL
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างและการทำงานของระบบ Elektron Tick History (ETH)

Elektron Tick History (ETH) เป็นระบบหนึ่งของบริษัท Refinitiv ที่ทำหน้าที่ในการให้บริการข้อมูลตลาดหุ้นทั่วโลกแก่ลูกค้า โดยตัวระบบมีบริการข้อมูลตลาดหุ้นตั้งแต่ ค.ศ. 1996 จนมาถึงปัจจุบัน โดยระบบ Elektron Tick History (ETH) ใจกลางการทำงานคือ Hadoop Framework จะทำหน้าที่ในการเก็บข้อมูลและประมวลผลข้อมูลที่มีปริมาณมหาศาล ซึ่งโครงสร้างและการทำงานของระบบ Elektron Tick History (ETH) เป็นไปดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างและการทำงานของระบบ Elektron Tick History (ETH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.2 สามารถอธิบายได้ดังนี้

ระบบ Elektron Tick History มีแหล่งที่ตั้งที่ประเทศสหรัฐอเมริกา ซึ่งการทำงานของระบบแบ่งออกเป็น 2 Server Site คือ HDC (Hazelwood, Missouri) และ PLN (Plano, Texas) โดยการทำงานเป็นไปดังต่อไปนี้

- เริ่มต้นที่ ลูกค้าส่ง Requested Data ขอข้อมูลตลาดในช่วงเวลาที่ต้องการผ่านเว็บแอปพลิเคชัน (Data Scope) ที่บริษัทเปิดให้บริการ
- เมื่อลูกค้าส่ง Requested Data ผ่านเว็บแอปพลิเคชัน (Data Scope) จะต้องผ่านจุดตรวจที่ Reverse Proxy เพื่อให้ Requested Data ถูกส่งไปยัง Web Service ของ Server ที่เปิดให้บริการอยู่ขณะนั้น
- เมื่อ Web Service ได้รับ Requested Data จะจัดรูปแบบของ Requested Data ใหม่ให้อยู่ในรูปแบบที่ตรงกับการจัดเก็บในระบบฐานข้อมูล
- เมื่อการจัดรูปแบบสำเร็จ Web Service จะส่ง Requested Data ไปที่ MQ Kafka ซึ่งเป็นแหล่งพัก Requested Data เพื่อรอ Job Submitter พร้อมนำไปทำงาน
- เมื่อ Job Submitter พร้อม จะนำ Requested Data จาก MQ Kafka มาสร้างเป็น Spark Job เพื่อจะส่งเข้า HADOOP Cluster ต่อไป
- ต่อมาเมื่อ Job เข้าสู่ HADOOP Cluster จะถูก YARN ตรวจสอบและนำไปเข้าคิวเอาไว้เพื่อรอระบบบน HADOOP Cluster พร้อมสำหรับ Job นั้น
- เมื่อระบบบน HADOOP Cluster พร้อมจะทำการนำ Job ไปทำการประมวลผล โดยการนำ Spark เข้ามาช่วยในการทำ Data Processing
- จนกระทั่งเมื่อได้ผลลัพธ์ ระบบจะทำการนำผลลัพธ์ทั้งหมดส่งไปที่ Amazon Web Service และทำการ Merge ผลลัพธ์ แล้วรอลูกค้ามาทำการดาวน์โหลดต่อไป

โดยในแต่ละ Requested Data บนระบบ ETH จะมีการเก็บและอัปเดตข้อมูลการทำงานอยู่ตลอดเวลา เช่น ชื่อบริษัท ชื่อผู้ใช้งาน สถานะของ Requested Data ณ เวลาปัจจุบัน และเวลาหลังจากเสร็จการทำงานในแต่ละส่วน เป็นต้น โดยข้อมูลดังกล่าวนี้จะถูกเก็บอยู่บนระบบฐานข้อมูลหลักของบริษัทซึ่งใช้ในการบันทึกรายละเอียดของ Requested Data ทั้งหมดมาตลอดหลายปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การทำงานของ Web Service

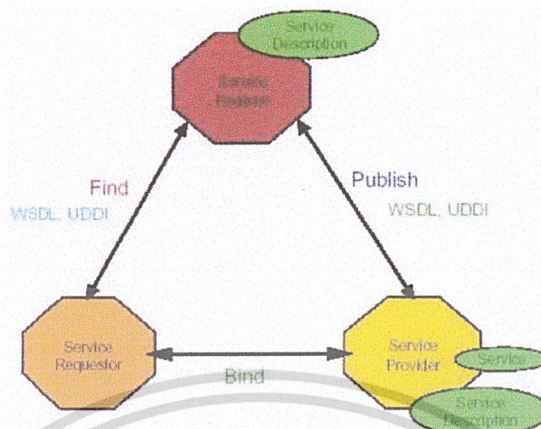
Web service (เว็บเซอร์วิส) คือ Application หรือ Program ในลักษณะให้บริการ จะถูกเรียกใช้งานจาก application อื่นในรูปแบบ RPC (Remote Procedure Call) โดยภาษาที่ถูกใช้เป็นตัวกลางในการแลกเปลี่ยนคือ XML บน protocol HTTP ซึ่งเป็น protocol สำหรับ World Wide Web ที่เป็นช่องทางที่ได้รับการยอมรับในการติดต่อสื่อสารกันระหว่าง application กับ application

Web Service ช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอปพลิเคชันที่ต่างกันเป็นไปโดยง่าย โดยแอปพลิเคชันนั้นๆ สามารถเขียนด้วย Java และรันอยู่บน Sun Solaris Application Server หรืออาจจะเขียนด้วย C++ และรันอยู่บน Windows NT หรืออาจจะเขียนด้วย Perl และรันอยู่บนเครื่อง Linux ซึ่งมาตรฐานของ Web Service ทำให้อินเทอร์เน็ตของแอปพลิเคชันเหล่านี้ ถูกอธิบายโดย WSDL และทำให้อยู่ในมาตรฐานของ UDDI หลังจากนั้น จึงสามารถติดต่อสื่อสารถึงกันโดย XML ผ่าน SOAP อินเทอร์เน็ต

Web Service สามารถถูกเรียกใช้ภายในองค์กรและภายนอกองค์กร โดยผ่านไฟร์วอลล์ ดังนั้นจึงมีองค์กรมากมาย พัฒนาระบบที่มีอยู่ของตน ให้เข้ากับ Web Service นับเป็นการลงทุนที่คุ้มค่า เนื่องด้วย Web Service จะสามารถเพิ่มศักยภาพการทำงานขององค์กร ลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กรได้

Web Service สามารถใช้ร่วมกับ Web Application โดยส่งผ่านข้อมูลทางอินเทอร์เน็ตได้ นับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้าหรือหุ้นส่วน แม้ต้องคำนึงถึงระบบรักษาความปลอดภัย และการจัดการรายการของข้อมูลอยู่ แต่ Web Service ได้ใช้มาตรฐานทั่วไปของ internet เรื่องดังกล่าวจึงนับเป็นเรื่องธรรมดาของการสื่อสารผ่านระบบอิเล็กทรอนิกส์

Web Service Model



รูปที่ 2.3 Web Service Model

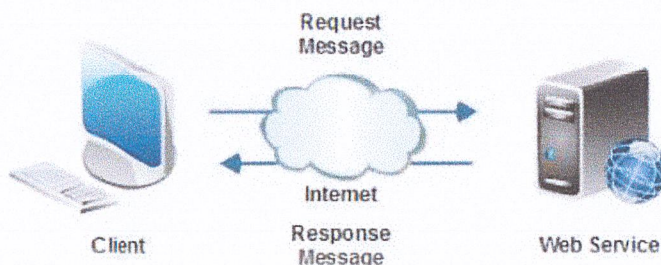
- Requestorทำหน้าที่เรียกใช้บริการจาก Provider ซึ่งสามารถค้นหาบริการที่ต้องการได้จาก UDDI registry หรือ Service Registry หรือติดต่อจาก Provider โดยตรง
- Registryทำหน้าที่เป็นตัวกลางให้ Provider มาลงทะเบียนไว้ โดยใช้ WSDL ไฟล์ บอก รายละเอียดของบริษัทและบริการที่มีให้ ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้
- Providerเป็นผู้ให้บริการ มีหน้าที่ในการเปิดบริการเพื่อรองรับการขอใช้บริการจาก Requestor ที่เรียกเข้ามาขอใช้

เทคโนโลยีที่ใช้ในการพัฒนา Web Services

- XML (Extensible Markup Language) เป็น format ที่จะบอกแหล่งที่มา และ รายละเอียดข้อมูลได้
- SOAPหรือREST เป็น protocol ที่ใช้แลกเปลี่ยนข้อมูล
- WSDL (Web Services Description Language) ใช้ในการอธิบายโครงสร้างของ service ทั้งหมด หรือ คู่มือการใช้งาน
- UDDI (Universal Description, Discovery, and Integration) เป็นส่วนกลางให้ระบบอื่น มาลงทะเบียนไว้ โดยใช้ WSDL file ทำให้ฝั่ง Client สามารถดึงเอาข้อมูลไปใช้ติดต่อกับ web service ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของ Web service



รูปที่ 2.4 การทำงานของ Web Service

จากรูปที่ 2.4 สามารถอธิบายได้ดังนี้

- ผู้ Client ทำการโหลดข้อมูลต่างๆ มีใช้ในการติดต่อไปยัง server เข้าสู่ SOAP message
- ส่ง SOAP message ไปยัง web service ด้วยการทำให้ HTTP POST
- ผู้ Web service ก็จะทำการแกะข้อมูลออกจาก SOAP และเปลี่ยนเป็นคำสั่งที่ application เข้าใจ
- ในส่วน application ก็จะนำเอาคำสั่งหรือข้อมูลที่ได้ไปทำงานจนได้ผลลัพธ์ ที่ต้องส่งกลับไปหาผู้ Client จัดใส่ใน HTTP format
- ผู้ Client ก็จะแกะข้อมูลออกจาก SOAP message เพื่อเราผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เฟรมเวิร์คและไลบรารีที่ใช้ในการพัฒนาระบบ

2.4.1) การทำงานของ FLASK Framework

Flask Framework เป็น Web Framework ที่เขียนขึ้นมาสำหรับ Python เพื่อใช้ร่วมกัน Web Server เช่น Apache และได้รับการยอมรับจาก Community Web Pages เช่น Pinterest, LinkedIn เป็นต้น โดย Flask ถูกเรียกว่า micro framework เพราะว่ามันต้องการเครื่องมือ หรือ library อะไรมาก อีกสิ่งที่ไม่จำเป็นต้องมีฐานข้อมูลด้วย แต่อย่างไรก็ตาม Flask ก็ยังรองรับการเพิ่ม extensions พิเศษได้

โดย Flask Framework มีส่วนประกอบหลัก 2 ไลบรารีหลักดังนี้

1. Werkzeug เป็นไลบรารีที่รวบรวมเครื่องมือสำหรับสร้าง WSGI application ซึ่งประกอบด้วย debugger ,ฟังก์ชันในการรับส่งข้อมูลไปกลับระหว่างฝั่งเซิร์ฟเวอร์และฝั่งผู้ใช้งาน , การควบคุมและจัดการ cache และ cookie รวมถึงระบบที่สามารถการกำหนดเส้นทางของ URL ที่มีประสิทธิภาพ



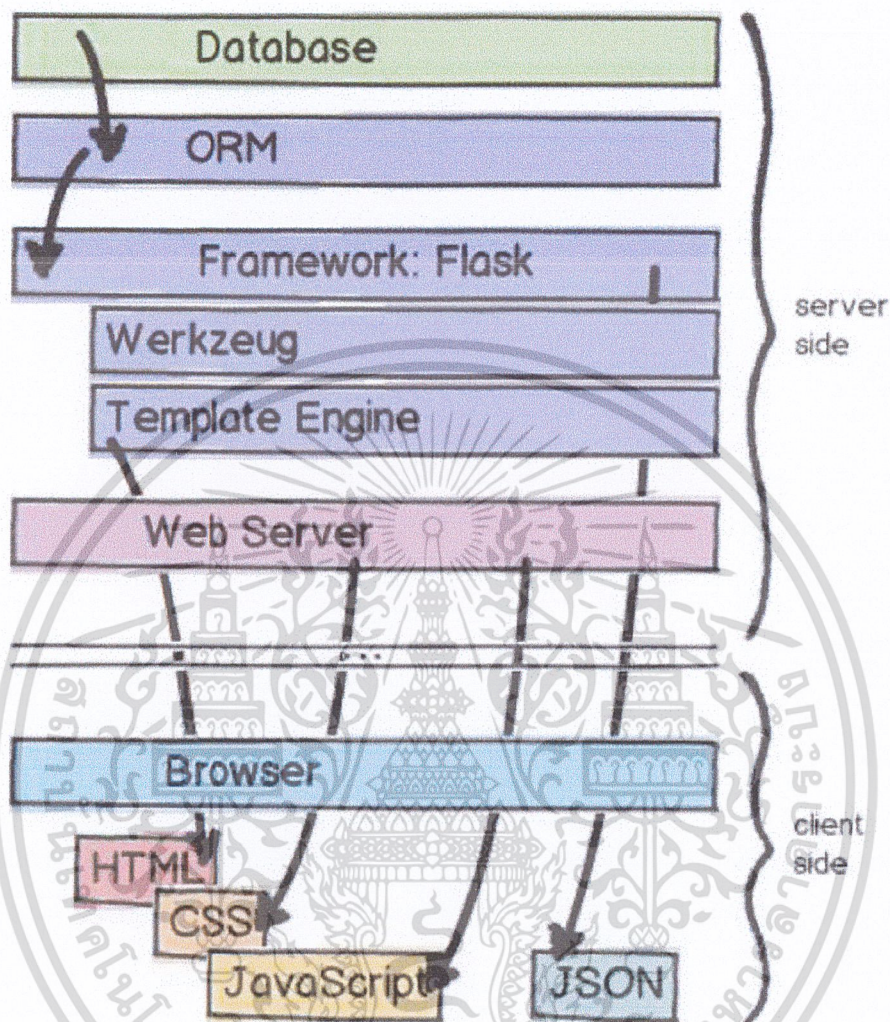
รูปที่ 2.5 Logo ของ Werkzeug

2. Jinja2 เป็นเครื่องมือใช้ในการสร้าง template แบบเต็มรูปแบบสำหรับภาษา python ซึ่งมีการสนับสนุน unicode มีการใช้กันอย่างแพร่หลายและเป็น open source ในรูปแบบของ BSD (Berkeley Software Distribution License)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.6 Logo ของ Jinja
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flask Framework Model



รูปที่ 2.7 Flask Framework Model

จากรูปที่ 2.7 สามารถอธิบายได้ดังนี้

Flask framework จะทำหน้าที่เป็น web service ในการติดต่อกับแหล่งข้อมูลต่างๆด้วยไลบรารีของภาษา python สำหรับติดต่อกับแหล่งข้อมูลแต่ละประเภท โดยจะที่จะมี template engine เรียกว่า Jinja2 ในการสร้าง HTML ในฝั่งของผู้ใช้งาน นอกจากนั้น Flask Framework สามารถรองรับการรับส่งโครงสร้างของข้อมูลในรูปแบบของ JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสามารถติดตั้งและสร้างเว็บไซต์ด้วย Flask Framework ได้ดังตัวอย่างต่อไปนี้

1. เปิด Command Line แล้วพิมพ์คำสั่ง pip install flask กด Enter
2. สร้างไฟล์ main.py ที่มีเนื้อหาภายในดังรูปที่ 2.8
3. สามารถรันการทำงานของโปรแกรมได้โดยพิมพ์คำสั่ง python main.py บน Command Line
4. สามารถดูผลลัพธ์ของโปรแกรมได้โดยเปิด web browser และใส่ URL ต่อไปนี้ localhost:3000

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World"
if __name__ == "__main__":
    app.run()
```

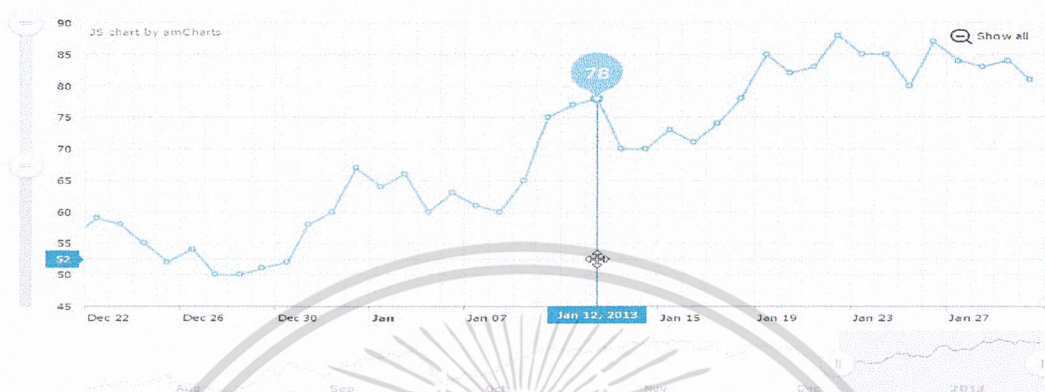
รูปที่ 2.8 ตัวอย่างการเขียนโปรแกรมด้วย Flask Framework



รูปที่ 2.9 Logo ของ Flask Framework

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างการโต้ตอบกับผู้ใช้งานด้วย JavaScript ได้ในทำนองเดียวกัน เราสามารถใช้ JavaScript ในการสร้างการโต้ตอบระหว่างกราฟิก SVG กับผู้ใช้งานได้ เช่น การใส่ click event หรือ on mouse over event ให้กับรูปทรงต่างๆ เป็นต้น



รูปที่ 2.10 ตัวอย่างผลลัพธ์ที่ได้จากการใช้ amChart สร้างแผนภูมิข้อมูล

```

<style>
#chartdiv {
width      : 100%;
height     : 500px;
font-size  : 11px;
}
</style>

<script src="https://www.amcharts.com/lib/3/amcharts.js"></script>
<script src="https://www.amcharts.com/lib/3/serial.js"></script>
<script src="https://www.amcharts.com/lib/3/plugins/export/export.min.js"></script>
<link rel="stylesheet" href="https://www.amcharts.com/lib/3/plugins/export/export.css" type="text/css" media="all" />
<script src="https://www.amcharts.com/lib/4/themes/animated.js"></script>

var chart = AmCharts.makeChart( "chartdiv", {
  "type": "serial",
  "theme": "none",
  "dataProvider": [ {
    "country": "USA",
    "visits": 2025
  }, {
    "country": "China",
    "visits": 1882
  } ],
  "valueAxes": [ {
    "gridColor": "#FFFFFF",
    "gridAlpha": 0.2,
    "dashLength": 0
  } ],
  "gridAboveGraphs": true,
  "startDuration": 1,
  "graphs": [ {
    "balloonText": "[[category]]: <b>[[value]]</b>",
    "fillAlphas": 0.8,
    "lineAlpha": 0.2,
    "type": "column",
    "valueField": "visits"
  } ],
  "chartCursor": {
    "categoryBalloonEnabled": false,
    "cursorAlpha": 0,
    "zoomable": false
  },
  "categoryField": "country",
  "categoryAxis": {
    "gridPosition": "start",
    "gridAlpha": 0,
    "tickPosition": "start",
    "tickLength": 20
  },
  "export": {
    "enabled": true
  }
} );
</script>

<div id="chartdiv"></div>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่บนสื่อออนไลน์
รูปที่ 2.11 ตัวอย่างการใช้งาน amCharts

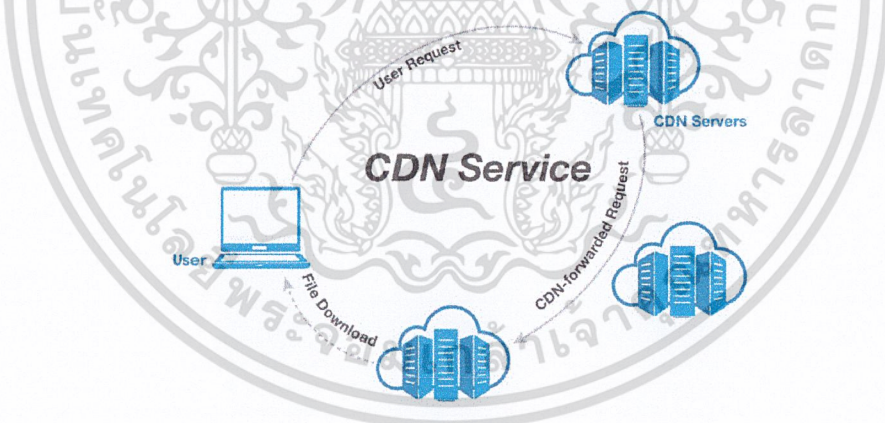
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตีตลับส่งเนื้อหา และต้องขออนุญาตทุกครั้งที่มีการนำไปใช้

2.4.3) การทำงานของ DataTables CDN

DataTables CDN (Content Delivery Network) เป็น plug-in สำหรับ jQuery ซึ่งเป็น Library ของ JavaScript ที่มีความยืดหยุ่นสูง ที่ทำให้ตาราง table บน HTML ให้มีความสามารถมากขึ้น เช่น การเรียงลำดับข้อมูลบนตารางแบบ real-time การค้นหาข้อมูลบนตารางแบบ real-time เป็นต้น

โดย CDN (Content Delivery Network) ในที่นี้คือระบบเครือข่ายขนาดใหญ่ประกอบด้วย ศูนย์ข้อมูล (Data Center) จำนวนมากที่เชื่อมต่อกันด้วยอินเทอร์เน็ต หน้าที่ของ CDN คือการส่งข้อมูลให้ถึงมือผู้ชมอย่างรวดเร็ว มีประสิทธิภาพและพร้อมให้ผู้ชมเข้าถึงข้อมูลได้ตลอดเวลา โดยวิธีการทำงาน จะกระจาย traffic ไปยัง server ต่างๆจำนวนมากบนเครือข่ายเพื่อเป็นการช่วยลดภาระการรองรับ traffic จำนวนมหาศาลบน server เครื่องใดเครื่องหนึ่งลง เช่น เมื่อมีผู้ชมจากที่ใดๆ ก็ตามเรียก URL ของเว็บไซต์ที่มีโฮสต์หลักที่ใช้ CDN เทคโนโลยี CDN จะรีไทร์คำขอไปยัง server ในเครือข่ายที่อยู่ใกล้พื้นที่ที่ผู้ชมเรียกดูมากที่สุด จึงช่วยให้เว็บไซต์ ดำเนินได้อย่างคล่องตัวไม่หยุดชะงักในเรื่องของการเชื่อมต่อผ่านเครือข่ายอินเทอร์เน็ตและลดปัญหาในการเชื่อมต่อไปยังระบบลงได้มาก

Content Delivery Network Model



รูปที่ 2.12 Content Delivery Network Model

คุณสมบัติของ Content Delivery Network (CDN)

- เหมาะสำหรับ Content ที่มีความจำเป็นต้องใช้ Bandwidth สูง เช่น ไฟล์วิดีโอ หรือไฟล์รูปภาพ ผู้ใช้บริการสามารถจัดการข้อมูลได้ทุกที่ ทุกเวลา ผ่านระบบของเรา
- สามารถรองรับผู้ใช้งานจำนวนมากจากทั่วทุกมุมโลกได้ในเวลาพร้อมๆ กัน บนระบบ CDN ที่ช่วยให้ Content ของท่านอยู่ใกล้ผู้ชมมากที่สุด ด้วยเครื่องแม่ข่ายที่ถูกกระจายติดตั้งยังศูนย์ข้อมูลต่างๆ ทั้งในประเทศ และต่างประเทศ ซึ่งจะช่วยให้อัปโหลดข้อมูลถึงผู้เรียกใช้ได้อย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีการแบ่งภาระงาน โดยการกระจาย traffic ที่มีอยู่จำนวนมาก ไปยังเครื่องแม่ข่ายที่มีอยู่จำนวนมาก เพื่อลดภาระหน้าที่ต่อเครื่อง อันจะเป็นการคงไว้ซึ่งประสิทธิภาพในการส่งข้อมูลถึงผู้ใช้ให้ได้ตามมาตรฐาน

#	First Name	Last Name	Email	Gender	Country	Salary (\$)	Actions
4	Emily	Hayes	ehayes3@elp.com	Female	Brazil	3355.26	Edit Delete
5	Sara	Anderson	sanderson4@netvibes.com	Female	Sierra Leone	4930.68	Edit Delete
7	Margaret	Baker	mbaker5@moonfruit.com	Male	Sierra Leone	3527.62	Edit Delete
8	Loi	Sanders	lsanders7@wix.com	Female	Indonesia	4694.34	Edit Delete
9	Harry	Alvarez	h Alvarez@reverbnation.com	Male	China	62.22	Edit Delete
11	Randy	Ryan	rryan@zenithy.com	Male	Mexico	8057.42	Edit Delete
12	Antonio	Carter	acarterb@photobucket.com	Male	Nigeria	1347.99	Edit Delete
13	Fred	Dean	fdean@c.wikispaces.com	Male	Honduras	3424.44	Edit Delete
14	Tina	Berry	tberryd@vixied.com	Female	Sierra Leone	7709.93	Edit Delete
15	Andrew	Harvey	aharveye@gmail.com	Male	Ghana	9007.19	Edit Delete

รูปที่ 2.13 ตัวอย่างผลลัพธ์ที่ได้จากการใช้ DataTable CDN สร้างตารางข้อมูล

```

<table id="example" class="display" style="width:100%">
  <thead>
    <tr>
      <th>Name</th>
      <th>Position</th>
      <th>Office</th>
      <th>Age</th>
      <th>Start date</th>
      <th>Salary</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Tiger Nixon</td>
      <td>System Architect</td>
      <td>Edinburgh</td>
      <td>61</td>
      <td>2011/04/25</td>
      <td>$320,800</td>
    </tr>
    <tr>
      <td>Garrett Winters</td>
      <td>Accountant</td>
      <td>Tokyo</td>
      <td>63</td>
      <td>2011/07/25</td>
      <td>$170,750</td>
    </tr>
  </tbody>
</table>
<script>
  <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
  <script src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
  <script src="https://cdn.datatables.net/1.10.19/css/jquery.dataTables.min.css"></script>
  $(document).ready(function() {
    $('#example').DataTable();
  });
</script>

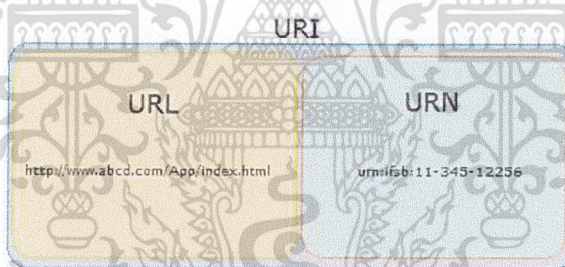
```

รูปที่ 2.14 ตัวอย่างการใช้งาน DataTables CDN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4) การทำงานของ HADOOP

HADOOP เป็น Java programming framework ที่รองรับการทำงานที่ต้องประมวลผลและเก็บข้อมูลขนาดใหญ่ เป็นส่วนหนึ่งของ apache project โดย HADOOP ถูกออกแบบมาให้เป็น application ที่สามารถทำงานได้บนระบบแบบ node หรือมี hardware จำนวนหลายๆเครื่องพร้อมกัน เพื่อรองรับข้อมูลขนาดใหญ่ ซึ่งใช้การแตก file system ออกมากระจายตาม node ให้สามารถทำงานได้รวดเร็วพร้อมทั้งส่งข้อมูลหากันระหว่าง node ทั้งหมด รวมถึงมีความสามารถในการจัดการ node มีเสียหายได้โดยไม่ทำให้เกิดข้อมูลสูญหาย ปัจจุบัน HADOOP ถูกนำมาใช้ในงาน big data จำพวก การคำนวณข้อมูลทางวิทยาศาสตร์เฉพาะทาง, เชิงธุรกิจ รวมถึงวางแผนการขาย และประมวลผลข้อมูลจำนวนมาก หรือ internet of things (IOT) ซึ่ง HADOOP ยังสามารถขยายระบบเพิ่มเพื่อรองรับการทำงานที่สูงขึ้น โดยอาศัย software package เช่น Apache Spark ที่เป็น engine สำหรับ big data ที่ใช้ประมวลผลข้อมูลแบบ streaming และรองรับ SQL นอกจากนั้นยังมี HADOOP YARN web service REST APIs ที่สามารถสิทธิ์การเข้าถึง Cluster Node และ Application และประวัติการใช้ข้อมูลผ่าน A Uniform Resource Identifier (URI) ซึ่งประกอบด้วย URL (Uniform Resource Locator) และ URN (Uniform Resource Name)



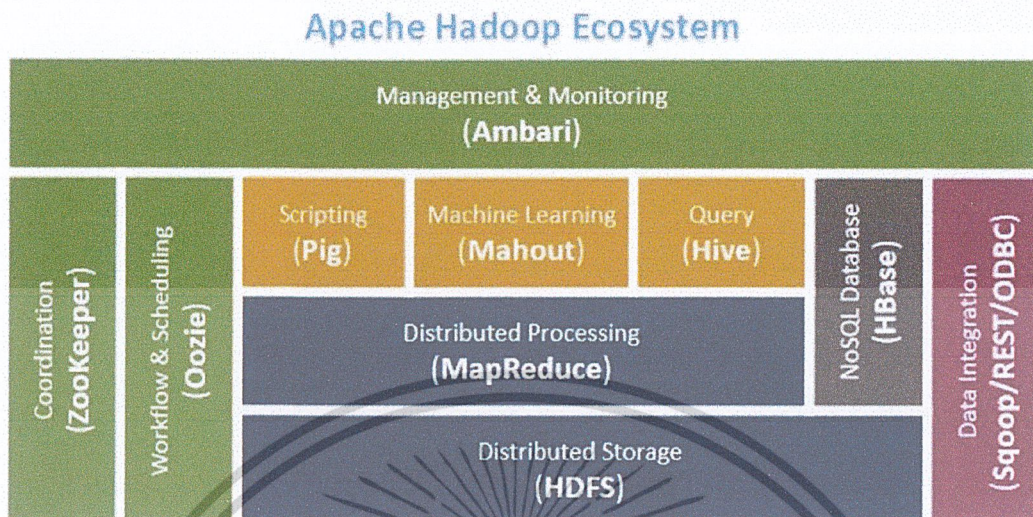
รูปที่ 2.15 อธิบายส่วนประกอบของ URI

องค์ประกอบหลักของ Hadoop

- Hadoop Distributed File System (HDFS) โมดูลนี้จะเอาไว้ใช้จัดเก็บข้อมูลที่จะนำมาวิเคราะห์ให้อยู่ในรูปแบบที่สามารถเข้าถึงได้อย่างรวดเร็ว รวมไปถึงการสำรองข้อมูลดังกล่าวให้โดยอัตโนมัติ
- Hadoop Common เป็น library ที่จำเป็นในการใช้งานเปรียบได้กับ kernel
- Hadoop Yet Another Resource Negotiator (YARN): ทำหน้าที่จัดการทรัพยากรที่มีอยู่ รวมถึงรอบการทำงานของ application
- MapReduce ส่วนโมดูลนี้จะเอาไว้ใช้เกี่ยวกับการประมวลผลข้อมูลปริมาณมหาศาลที่ได้เก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 เอาไว้
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบนิเวศของ Hadoop



รูปที่ 2.16 ระบบนิเวศของ Hadoop

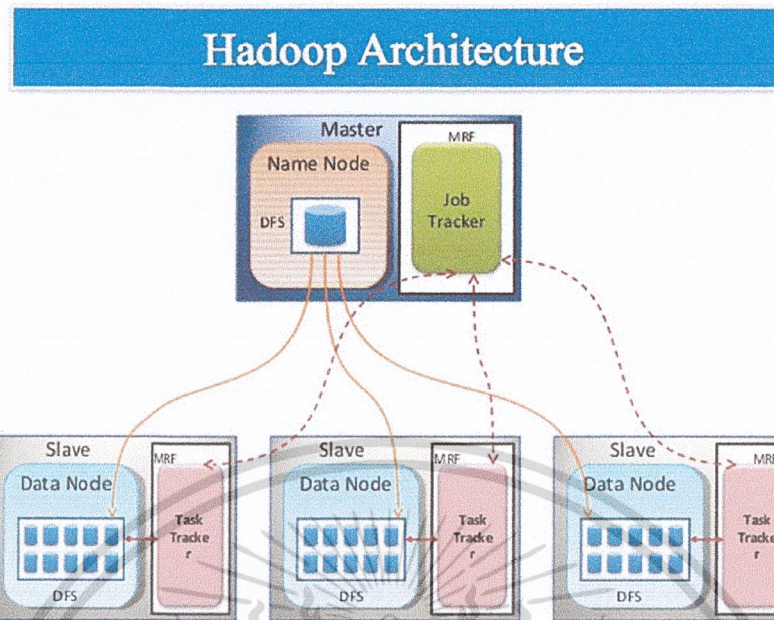
กระบวนการประมวลผลของ MapReduce

กระบวนการทำงานแบ่งออกเป็น 2 ขั้นตอนดังต่อไปนี้

ขั้นตอนแรก Map ซึ่งทำหน้าที่จับคู่ข้อมูล Input เป็น Key กับ Value ได้ออกมาเป็น Intermediate key/value pair (มันคือ Key กับ Value ที่ได้ออกมาหลังจากฟังก์ชัน Map นั้นแหละไม่มีอะไรมาก) จากนั้นเฟรมเวิร์คของ MapReduce จะรวม Value ที่มี Key เหมือนกันเอาไว้ด้วยกัน ตอนที่รวมกันนี้เราเรียกว่า Shuffle

ขั้นตอนที่สอง Reduce ซึ่งเป็นขั้นตอนที่รับ Intermediate key/value pair จากขั้นตอน Map มาแล้วดึงเอาข้อมูลที่ Key ตรงกันมาดำเนินการร่วมกันเพื่อให้ได้ผลลัพธ์ที่ขนาดเล็กที่สุดที่เป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่า สถาบันยกรั่มของ Hadoop ให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 สถาปัตยกรรมของ Hadoop

หน้าที่ของโหนดภายใน Hadoop

- Data Node เป็น Node ที่ทำหน้าที่เก็บ Block ของไฟล์เอาไว้ และรับผิดชอบในการประมวลผล Block นั้นๆ
- Name Node เป็น Node ที่ทำหน้าที่รวบรวมผลของการประมวลผล Block ต่างๆ จาก Data Node ทั้งหมด Name Node จะรู้ทุกอย่างเกี่ยวกับไฟล์ต้นฉบับ ไม่ว่าจะเป็นชื่อไฟล์, ขนาด รวมถึงที่อยู่ของแต่ละ Block ที่ถูกกระจายออกไปตาม Data Node ต่างๆ



รูปที่ 2.18 Logo ของ HADOOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

2.5.1) โปรแกรม Visual Studio Code

Visual Studio Code คือ โปรแกรมที่ใช้ในการแก้ไขและปรับแต่งโค้ด ที่มีการตัดฟังก์ชันใน ส่วนของ GUI designer ออกไปสามารถทำงานข้าม platform ได้ทั้งบน Windows และ Linux รวมทั้งซัพพอร์ตภาษาหลายร้อยภาษา นอกจากนั้น Visual Studio Code ยังมี extension ที่ช่วยในการ อำนวยความสะดวกในการเขียนโปรแกรมและสามารถเชื่อมต่อกับเครื่องมืออื่นๆเพื่อทำงานร่วมกันได้ เช่น Git Docker เป็นต้น



รูปที่ 2.19 Logo ของ Visual Studio Code

2.5.2) โปรแกรม Cisco AnyConnect Secure Mobility Client

Cisco AnyConnect Secure Mobility Client คือ แอปพลิเคชันที่ช่วยให้พนักงานสามารถ เชื่อมต่อกลับมายังระบบเครือข่ายของบริษัทจากที่ไหนก็ได้ ผ่านระบบ VPN ที่มีการเข้ารหัสแบบ SSL รวมทั้งมีฟีเจอร์ในการตรวจสอบเซอร์วิส ความคุ้มครองการใช้งานและความปลอดภัยบนเว็บช่วยลดความ ยุ่งยากในการเข้าถึงปลายทางที่ปลอดภัยและให้การรักษาความปลอดภัยที่จำเป็น เพื่อช่วยให้องค์กร ของคุณปลอดภัยและมีการป้องกันอย่างมีประสิทธิภาพ

ความสามารถและประโยชน์ของ Cisco AnyConnect Secure Mobility Client

Access from anywhere คือ ให้ผู้ใช้สามารถเข้าถึงเครือข่ายองค์กรได้อย่างปลอดภัยจากอุปกรณ์ ไตๆได้ทุกที่ทุกเวลา

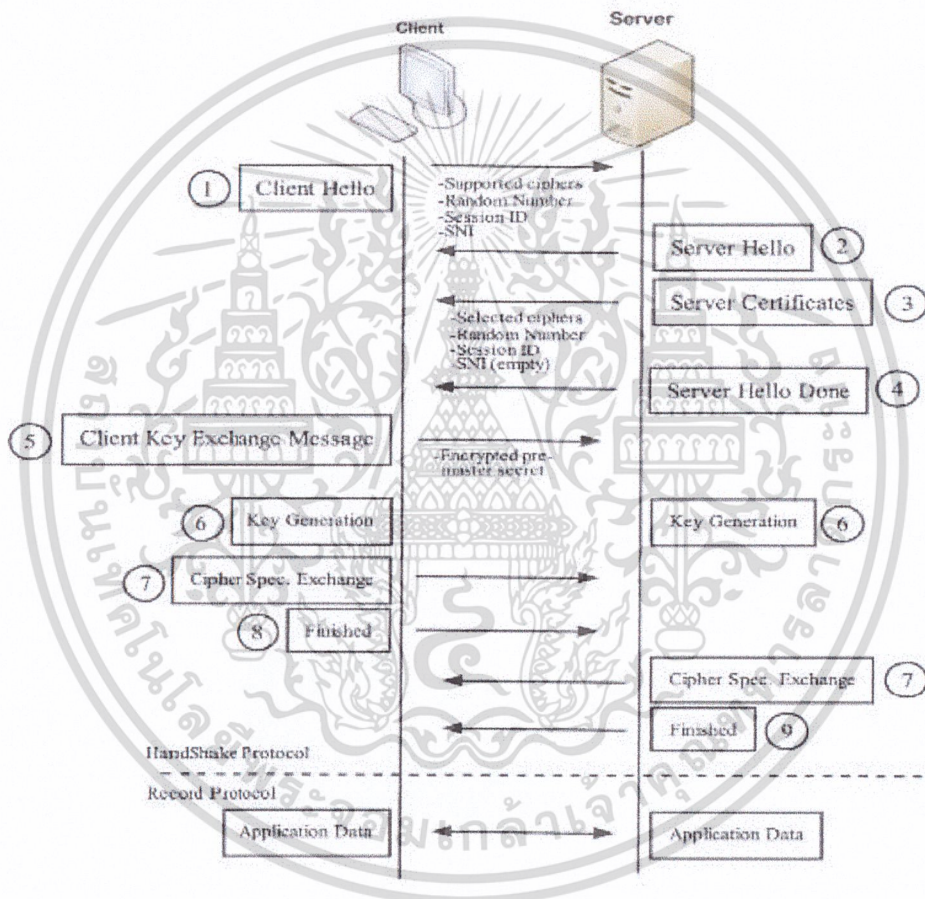
Greater visibility คือ ทำความเข้าใจเกี่ยวกับพฤติกรรมของผู้ใช้และปลายทางมากขึ้น ด้วย โมดูล Network Visibility Module (NVM) และปรับปรุงการดำเนินงานของเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Comprehensive protection คือ ป้องกันภัยคุกคามต่างๆ ตัวอย่างเช่น Cisco Identity Services Engine (ISE) สามารถป้องกันไม่ให้อุปกรณ์ที่ไม่ได้รับอนุญาตสามารถเข้าถึงเครือข่ายได้ และCisco Umbrella Roaming ทำการขยายการป้องกันเมื่อผู้ใช้ปิด VPN

Simplified management and usability คือ สามารถบริหารจัดการได้ง่ายทั้งภายในและภายนอกองค์กร ลดความซับซ้อนของการจัดการ สามารถใช้งานระบบได้อย่างมีประสิทธิภาพ

รูปแบบการทำงานของ Cisco AnyConnect VPN



รูปที่ 2.20 รูปแบบการทำงานของ Cisco AnyConnect VPN



รูปที่ 2.21 Logo ของ Cisco AnyConnect Secure Mobility Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3) โปรแกรม Trello

Trello (เทลโล) คือ collaboration software ไว้สำหรับบริหารจัดการ การทำงานร่วมกัน ระหว่าง ทีมพัฒนาและลูกค้า หรือทุก ๆ แผนกที่มีส่วนในการพัฒนาและตรวจสอบคุณภาพของงาน และ process การทำงาน ที่สำคัญ trello เป็นโปรแกรมที่เราสามารถใช้งานได้แบบฟรี ๆ ไม่มีค่าใช้จ่าย ให้เราสามารถใช้งานได้ทั้งเวอร์ชัน mobile หรือจะใช้บนเว็บไซต์หลักก็ได้ เป็นเครื่องมือที่มีประสิทธิภาพและใช้งานง่ายมากสำหรับการควบคุมดูงาน เพราะเครื่องมือที่ใช้บริหารจัดการโครงการ หรือ Project Management ประเภทนี้ มีให้ใช้งานกันหลากหลายค่ายขึ้นอยู่กับผู้ใช้จะชอบค่ายไหน แต่จากเครื่องมือที่ฟรีและทีมงานใช้แล้วรู้สึกว่ามีค่อนข้างดีคือ Trello การทำงานที่ง่าย ไม่ซับซ้อน เหมาะสำหรับใช้ส่วนตัวและองค์กร ทีมงานหรือแม้แต่จะใช้กับลูกค้าเพื่อบริหารงานร่วมกันก็ได้

จุดเด่นของ Trello

- เป็นซอฟต์แวร์ฟรี มีการพัฒนาอย่างต่อเนื่อง
- เป็นซอฟต์แวร์ที่ช่วยในการบริหารจัดการงานแต่ละส่วนได้เป็นอย่างดี สามารถลดกระบวนการทำงานบางอย่างลงไปได้
- มีความเป็น CRM สร้างความเชื่อมั่นให้กับลูกค้าได้ เพราะลูกค้าเองสามารถตรวจสอบขั้นตอนการทำงานได้เอง
- ไม่ซับซ้อน UX และ UI สามารถเรียนรู้ ทำความเข้าใจได้ง่าย ใช้งานง่าย ที่สำคัญเร็วมาก
- มีให้เราสามารถใช้งานได้ทั้ง Mobile Apps และบน Website

การสร้าง Trello Board, List และ Card

Board ในโปรแกรม Trello เปรียบได้กับการสร้าง 1 โปรเจกต์ และภายใน Board ก็จะมี List หรือส่วนการทำงานหลัก ๆ ของโปรเจกต์ว่า มีองค์ประกอบอะไรบ้าง ส่วน Card คือหัวข้อย่อยของงานที่อยู่ภายใน List

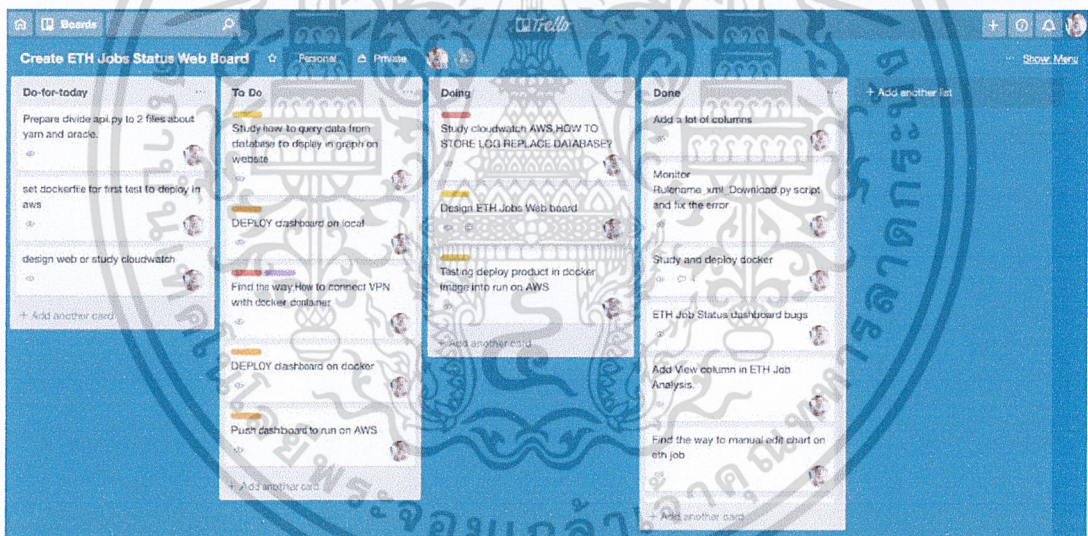
1. เข้าสู่ระบบที่ trello.com คลิกที่ปุ่ม “Create new board” ตั้งชื่อตามโปรเจกต์ที่กำลังพัฒนา เลือกว่า board จะเป็นแบบ private, public หรือ team กดปุ่ม Create
2. คลิก “Add a list” เพื่อแยกส่วนการทำงานหลักของโปรเจกต์
3. ภายใน List คลิก “Add a card” เพื่อสร้างหัวข้อย่อยของการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คลิกปุ่มเครื่องหมายบวกมุมบนขวามือ เป็นปุ่มสำหรับ invite เพื่อ assign คนที่จะต้องดูแลส่วนต่าง ๆ ของโปรเจค รวมไปถึงลูกค้าเจ้าของโปรเจคที่จะสามารถเข้ามาดู Board และ Process การทำงานได้พิมพ์ที่อยู่อีเมลล์และชื่อของคนที่เราต้องการจะ invite

5. หลังจากที่เร invite ทั้งฝั่งนักพัฒนาและลูกค้าเรียบร้อยแล้ว ให้ทำการ assign งานของแต่ละคนโดยการลากที่รูปคนมาวางไว้ใน card แต่ละส่วน หรืออาจจะกดที่ปุ่ม “change member” เพื่อกำหนดงานให้สมาชิกแต่ละคนคลิกปุ่มแก้ไข card เลือก “change member”

ความสามารถของ trello มีอีกมากมาย เช่น สามารถแนบไฟล์ได้ การใส่สถานะการทำงาน แต่ละขั้นตอน ไปจนถึงการปิดบอร์ดเมื่อโปรเจคเสร็จ ตรวจสอบและส่งงานและโปรเจคออนไลน์ trello เป็นโปรแกรมที่ทุกคนสามารถใช้งานได้ฟรี UX ค่อนข้างใช้งานง่าย ในขั้นตอนของการสร้างและใช้ส่วนต่าง ๆ ค่อนข้างเร็ว



รูปที่ 2.22 ตัวอย่างโปรแกรม Trello

Trello

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.23 Logo ของ Trello

2.5.4) โปรแกรม Apache

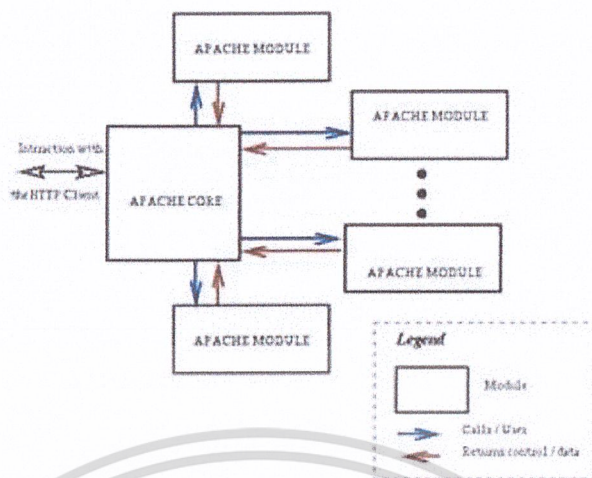
Apache คือซอฟต์แวร์ที่ทำหน้าที่เป็น Web Server โดยให้บริการโปรโตคอล HTTP ที่พอร์ต 80 และเป็นซอฟต์แวร์แบบ Open Source ซึ่งจะคอยจัดการร้องขอข้อมูล การส่งข้อมูลกลับไป จัดการกระบวนการและจัดลำดับ เก็บ log ที่มีการเข้าถึงรวมกระทั่งข้อผิดพลาด รวมทั้งสามารถนำเอา module มาใช้ร่วมกับ Web Server

ลักษณะทางกายภาพของอาปาเซ่

อาปาเซ่ถูกสร้างขึ้นจากการนำเอาโปรแกรมขนาดเล็กที่ทำหน้าที่แตกต่างกันหลายโมดูลมาทำงานร่วมกันเป็นเว็บเซิร์ฟเวอร์ จึงทำให้มีส่วนประกอบเป็นโมดูล (ที่พัฒนาด้วยภาษาซี) ส่วนที่ทำหน้าที่บริหารจัดการทั้งหมด (Core.c) โมดูลส่วนที่ทำหน้าที่บริหารหน่วยความจำ (Memory Management) และส่วนสุดท้ายทำหน้าที่บริหารโปรเซสงานย่อย (Child Process) ที่รองรับการบริการที่เรียกเข้ามาจำนวนมากพร้อม ๆ กัน (Multi-Processing Models) ซึ่งอาปาเซ่มีโมเดลการทำงานด้านนี้ 3 โมเดลด้วยกัน คือ Workers สำหรับรองรับงานจำนวนมาก โดยไม่ต้องการหน่วยความจำมาก Pre fork สำหรับงานที่ต้องการประสิทธิภาพและความเร็วแต่จะต้องใช้ทรัพยากรระบบมาก และ Per Child ถูกสร้างขึ้นมาเพื่อรองรับงานได้แตกต่างกันโดยแยกตามยูสเซอร์ที่ร้องขอบริการเข้ามา

นอกจากนั้นอาปาเซ่มีโมดูล http_core.c ทำหน้าที่รองรับการประมวลผลด้วยโปรโตคอล HTTP (Hyper Text Transfer Protocol) ซึ่งจะจัดการกับส่วนเอนเตอร์ตามมาตรฐาน NCSA และโมดูล mod_so.c จะทำหน้าที่ติดต่อประสานการทำงานโมดูลภายในเข้ากับกับ Shared Modules ซึ่งโมดูลภายนอกเรียกว่า Dynamic Shared Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 การทำงานภายใน Apache

โดยหน้าที่หลักของ web server ทั่วไปมีดังนี้

- คอยจัดการ Request ก็คือ การร้องขอข้อมูล
- คอยจัดการ Response ก็คือ การส่งข้อมูลกลับไป
- คอยจัดการ Process และจัดลำดับ ของ request และ response
- คอยเก็บ logs ที่มีการ access เข้ามารวมกระทั่ง error ต่างๆ ที่ web server พบ เช่น ไม่เจอไฟล์ชื่อนี้
- สามารถ เอา module มาใช้ร่วมกับ web server ได้ ยกตัวอย่างเช่น Apache นั้น ไม่สามารถ run ไฟล์ .php ได้ ต้อง เรียกใช้ module php อีกที หรือ การ rewrite url ใช้เดียวกันต้องใช้ mod_rewrite ในการสร้าง



APACHE

รูปที่ 2.25 Logo ของ Apache

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนในการพัฒนาระบบ

ในการพัฒนาเว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูลการเรียกใช้ข้อมูลของลูกค้าบนระบบ Elektron Tick History (ETH) หรือ Web Applications for Data Analysis of Customer Requested Data มีขั้นตอนการพัฒนาระบบดังต่อไปนี้

3.1 การวิเคราะห์และออกแบบระบบของเว็บแอปพลิเคชัน

จากศึกษาการทำงานของระบบ Elektron Tick History (ETH) ซึ่งทำให้ทราบว่าในปัจจุบัน ผู้ดูแลระบบต้องเผชิญกับปัญหาในการทำงานร่วมกับระบบ Elektron Tick History (ETH) ดังต่อไปนี้

1. การตรวจสอบปริมาณ Requested Data ที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบันได้ ใช้เวลานานและค้างช้ายุ่งยาก
2. การตรวจสอบรายละเอียด Requested Data ที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบันได้ ใช้เวลานานและค้างช้ายุ่งยาก
3. ข้อมูล Requested Data ที่ได้จากแหล่งข้อมูลไม่ได้อยู่ในรูปแบบที่ง่ายต่อการทำไปใช้และทำความเข้าใจ รวมถึงข้อมูลถูกเก็บไว้หลายแหล่ง

ดังนั้นจึงได้มีการพัฒนาเว็บแอปพลิเคชันที่ผู้ดูแลระบบสามารถใช้ในการตรวจสอบและวิเคราะห์ปริมาณของ Requested Data และรายละเอียดของ Requested Data ณ เวลานั้นได้ ซึ่งออกแบบให้สามารถทำงานตาม Functional Requirement ต่อไปนี้

Functional Requirement

- สามารถตรวจสอบปริมาณ Requested Data ที่กำลังรอการทำงานและกำลังงานอยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน
- มีการแยกแสดงปริมาณ Requested Data ตาม Web Server Site , State และ Queue อย่างชัดเจน
- มีการบันทึกปริมาณ Requested Data ลงฐานข้อมูลทุก 5 นาที
- สามารถตรวจสอบปริมาณ Requested Data ณ เวลาที่ผ่านไปแล้ว ในรูปแบบของแผนภูมิเส้น
- สามารถ เปิดหรือปิด การแสดงผลปริมาณ Requested Data และแผนภูมิเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปริมาณ Requested Data และแผนภูมิเส้น บนหน้าเว็บจะมีการอัปเดตค่า แสดงผลทุก 5 นาที
- สามารถแสดงรายละเอียดของแต่ละ Requested Data ในรูปแบบของตาราง
- สามารถดาวน์โหลดตารางรายละเอียดของ Requested Data ในรูปแบบไฟล์ CSV หรือ Excel
- มีช่องค้นหา “ข้อความ” ที่ตรงกับในตารางรายละเอียดของ Requested Data
- สามารถตรวจสอบปริมาณ Requested Data ทั้งหมดในแต่ละวันของผู้ใช้งานที่เป็นเจ้าของ Requested Data ซึ่งทำงานอยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบันมาแสดงในรูปแบบของแผนภูมิเส้น
- สามารถตรวจสอบขนาดของข้อมูลที่ขอ (RIC DAY) ทั้งหมดที่ขอในแต่ละวัน ของผู้ใช้งานที่เป็นเจ้าของ Requested Data ซึ่งทำงานอยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน มาแสดงในรูปแบบของแผนภูมิเส้น
- สามารถตรวจสอบค่าเฉลี่ยเวลาที่ระบบ ETHที่ใช้ในการประมวลผล Requested Data(TOTAL_TIME)ทั้งหมดในแต่ละวัน ของผู้ใช้งานที่เป็นเจ้าของ Requested Data ที่ทำงานบน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน มาแสดงในรูปแบบของแผนภูมิเส้น
- สามารถตรวจสอบปริมาณข้อมูลตลาดที่ขอทั้งหมด , ช่วงเวลาที่เริ่มขอ และช่วงเวลาสุดท้ายที่ขอ ของ Requested Data ซึ่งทำงานอยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน
- สามารถดาวน์โหลดรายละเอียดการขอข้อมูลภายใน Requested Data(Request JSON) ที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน ในรูปแบบไฟล์ JSON

โดยสามารถแบ่งขั้นตอนการวิเคราะห์และออกแบบระบบได้ดังต่อไปนี้

3.1.1) การวิเคราะห์แหล่งข้อมูลสำหรับเชื่อมต่อ Web Service

3.1.2) การวิเคราะห์และออกแบบฐานข้อมูลสำหรับเชื่อมต่อ Web Service

3.1.3) การวิเคราะห์และออกแบบชุดคำสั่ง สำหรับนำข้อมูลบันทึกลง Client usage Database

3.1.4) การวิเคราะห์และออกแบบ Web Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1) การวิเคราะห์แหล่งข้อมูลสำหรับเชื่อมต่อ Web Service

โดยแหล่งข้อมูลประกอบด้วย 2 แหล่ง ดังต่อไปนี้

3.1.1.1) YARN Resource Manager REST API

จะให้ข้อมูลทรัพยากรของระบบที่ Requested Data แต่ละตัวใช้ขณะอยู่บนระบบ ETH ในรูปแบบของ XML หรือ JSON Format ซึ่งข้อมูลที่น่ามาใช้ร่วมกับเว็บแอปพลิเคชันมีดังต่อไปนี้

- 1) Id คือ Unique key ที่ถูกสร้างขึ้นเพื่ออ้างอิง Requested Data แต่ละตัว
- 2) User คือ ชื่อผู้ใช้งานที่ขอ Requested Data
- 3) allocatedMB คือ ปริมาณหน่วยความจำหลักที่จองให้ Requested Data แต่ละตัว
- 4) allocatedVcore คือ กำลังไฟที่จองให้ Requested Data แต่ละตัว
- 5) allocatedSeconds คือ เวลาที่ Requested Data ใช้หน่วยความจำหลักตั้งแต่เริ่มจนถึงปัจจุบัน
- 6) allocatedVcore คือ เวลาที่ Requested Data ใช้กำลังไฟตั้งแต่เริ่มจนถึงปัจจุบัน
- 7) Queue คือ ขนาดของ Requested Data โดยถูกแบ่งได้ดังนี้
 - Regular คือ Requested Data ขนาดปกติ
 - Large คือ Requested Data ขนาดใหญ่
 - HMC คือ Requested Data ที่ผู้พัฒนาส่งมา
- 8) State คือ สถานะการทำงานอยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน โดยถูกแบ่งได้ดังนี้
 - Pending คือ Requested Data รอการประมวลผล
 - Running คือ Requested Data อยู่ระหว่างการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{"apps":
  { "app": [
    {
      "id": "application_1476912658570_0002",
      "user": "user2",
      "name": "word count",
      "queue": "default",
      "state": "FINISHED",
      "finalStatus": "SUCCEEDED",
      "progress": 100,
      "trackingUI": "History",
      "trackingUrl": "http://host.domain.com:8088/cluster/app/application_1476912658570_0002",
      "diagnostics": "...",
      "clusterId": 1476912658570,
      "applicationType": "MAPREDUCE",
      "applicationTags": "",
      "startedTime": 1476913457320,
      "finishedTime": 1476913761898,
      "elapsedTime": 304578,
      "amContainerLogs": "http://host.domain.com:8042/node/containerlogs/container_1476912658570_0002_02_000001/user2",
      "amHostHttpAddress": "host.domain.com:8042",
      "allocatedMB": 0,
      "allocatedVCores": 0,
      "runningContainers": 0,
      "memorySeconds": 206464,
      "vcoreSeconds": 201,
      "queueUsagePercentage": 0,
      "clusterUsagePercentage": 0,
      "preemptedResourceMB": 0,
      "preemptedResourceVCores": 0,
      "numNonAMContainerPreempted": 0,
      "numAMContainerPreempted": 0,
      "logAggregationStatus": "DISABLED"
    }
  ]
}

```

รูปที่ 3.1 ตัวอย่างรายละเอียดของ Requested Data ที่กำลังทำงานในระบบ ETH จากได้รับจาก YARN Resource Manager REST API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2) ฐานข้อมูลหลัก

เป็นฐานข้อมูลหลักที่ใช้ในการเก็บข้อมูลจาก Requested Data จำพวก ชื่อผู้ใช้งาน ชื่อบริษัท และเวลาที่ขอทั้งหมด รวมถึงเวลาที่สำเร็จกระบวนการจากส่วนต่างๆภายในระบบ ETH ตลอดเวลาซึ่งจำเป็นต้องใช้ VPN ในการเชื่อมต่อเพื่อดึงข้อมูลจากส่วนนี้ และทาง Web Service ของเว็บแอปพลิเคชันจะต่อเข้ากับฐานข้อมูลหลักก็ต่อเมื่อต้องการข้อมูลที่เป็น Real-Time เท่านั้น โดยข้อมูลจาก ฐานข้อมูลหลัก ที่นำมาใช้กับเว็บแอปพลิเคชันมีต่อไปนี้

- JOB_SUBMITTED_TIME คือเวลาที่ Requested Data เข้าถึง WebServer แต่ละ Site
- WS_QUEUE_TIME คือ เวลาที่ Requested Data ทำงานเสร็จจาก Web Server แต่ละ Site
- JS_DEQUEUE_TIME คือเวลาที่ Requested Data เข้าถึง Job Submitter
- YARN_ACCEPTED_TIME คือ เวลาที่ Requested Data เข้าถึง HADOOP Cluster และเข้ารอในคิว
- JOB_START_TIME คือ เวลาที่ Requested Data เริ่มถูกประมวลผล
- SPARK_START_TIME คือ เวลาที่ Spark ของ HADOOPCluster เริ่มประมวลผล
- JOB_FINISH_TIME คือ เวลาที่ Requested Data ประมวลผลเสร็จ
- MERGE_TIME คือ เวลาที่ Requested Data ถูก Merge สำเร็จ
- RIC_DAY คือ ขนาดข้อมูลที่ Requested Data ขอ
- REQUEST_JSON คือ รายละเอียดการขอข้อมูลภายใน Requested Data

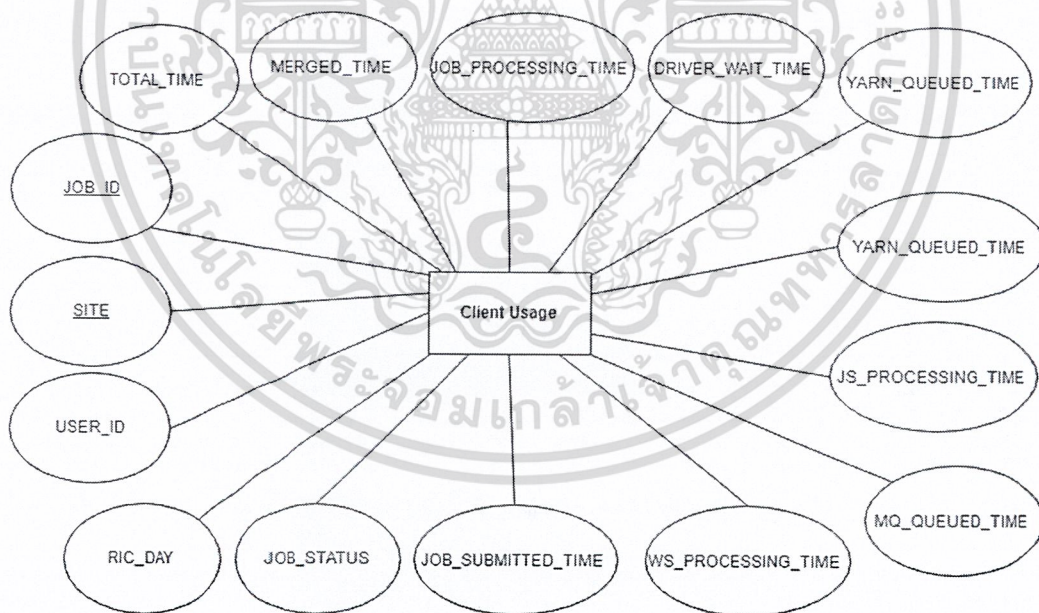
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2) การวิเคราะห์และออกแบบฐานข้อมูลสำหรับเชื่อมต่อ Web Service

3.1.2.1) Client usage Database

เป็นฐานข้อมูลMysql Database ที่ถูกสร้างขึ้นใหม่เพื่อทำงานคู่กับ Web Service ของเว็บแอปพลิเคชันและใช้ในการประกอบการวิเคราะห์รูปแบบอื่น โดยสาเหตุที่จำเป็นต้องสร้างฐาน Client usage Database เนื่องจาก ฐานข้อมูลหลักของบริษัทต้องบันทึกรายละเอียดของ Requested Dataที่เข้ามาทำงานบนระบบ ETH ทั้งหมดอยู่ตลอดเวลาเพื่อลดภาระในการทำงานฐานข้อมูลหลักให้มากที่สุด โดยฐานข้อมูลใหม่นี้จะนำเฉพาะข้อมูลที่จำเป็นจากฐานข้อมูลหลักบางส่วนมาบันทึกและนำข้อมูลบางส่วนมาผ่านการประมวลผลก่อนจะทำการบันทึกเพื่อให้สามารถนำไปใช้งานต่อและวิเคราะห์ง่ายขึ้น โดยข้อมูลจาก Client usage Database ที่นำมาใช้กับระบบของเว็บแอปพลิเคชันมีโครงสร้างของ Table และรายละเอียดมีดังต่อไปนี้

ER Diagram ของ Client Usage Table



รูปที่ 3.2 ER Diagram ของ Client Usage Table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 Data descriptions ของ Client usage Database

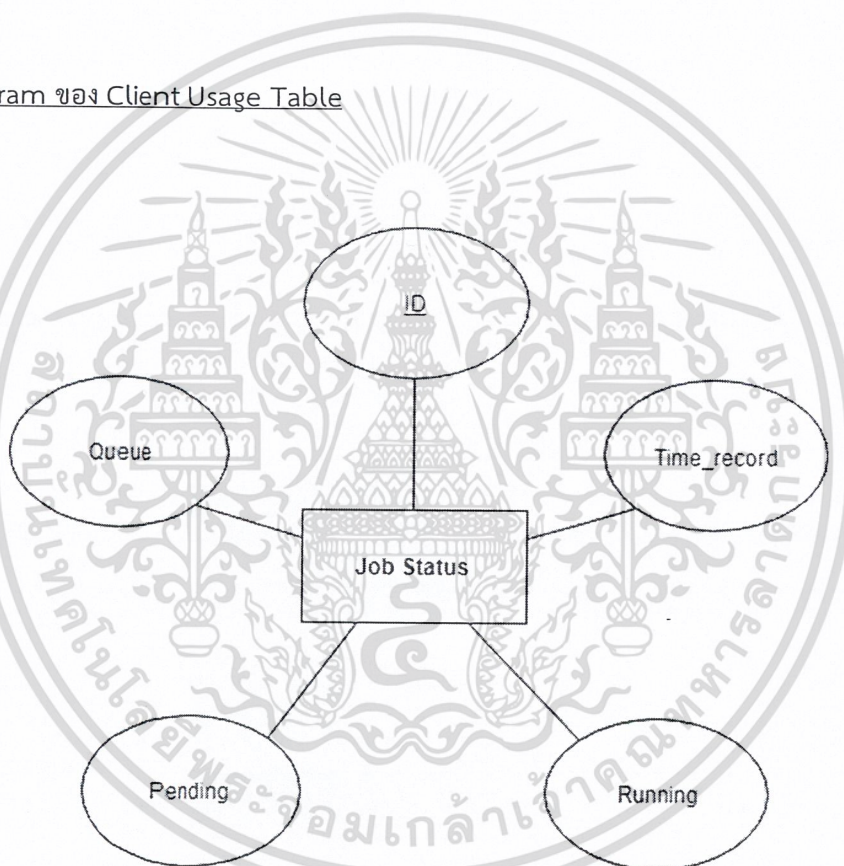
ชื่อฟิลด์	ประเภทข้อมูล	ประเภทคีย์	ความหมาย	ข้อกำหนด
JOB_ID	VARCHAR	PK	ไอดีของ Requested Data ที่ถูกสร้างขึ้นเพื่อระบุแต่ละ Requested Data	NOT NULL
SITE	VARCHAR	PK	ชื่อของ Server Site ที่รับ Requested Data ไปประมวลผล	NOT NULL
USER_ID	VARCHAR		ไอดีผู้ใช้งานที่ขอ Requested Data	
RIC_DAY	INT		ขนาดข้อมูลที่ Requested Data ขอ	
JOB_STATUS	VARCHAR		สถานะของ Requested Data ณ ปัจจุบัน	
JOB_SUBMITTED_TIME	DATETIME		เวลาที่ Requested Data เข้าถึง Web Server แต่ละ Site	
WS_PROCESSING_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในส่วนของ Web Service ของแต่ละ Site	
MQ_QUEUED_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในส่วนของ MQ Kafka	
JS_PROCESSING_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในส่วนของ Job Submitter	
YARN_QUEUED_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในส่วนของ YARN QUEUE	
DRIVER_WAIT_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในส่วนการรอ Spark ทำงาน	
JOB_PROCESSING_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในส่วนของ การประมวลผล	
MERGED_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในส่วนของ Merge ผลลัพธ์บน Amazon Web Service	
TOTAL_TIME	DOUBLE		ช่วงเวลาที่ Requested Data ใช้ในการประมวลผลทั้งหมดบนระบบ ETH	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1) Job Status Database

เป็นฐานข้อมูลMysql Database ที่ถูกสร้างขึ้นใหม่เพื่อทำงานคู่กับ Web Service ของเว็บแอปพลิเคชันโดยเฉพาะ เพื่อบันทึกปริมาณข้อมูลของ Requested Dataที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน ทุก 5 นาทีจาก YARN Resource Manager REST API แล้วนำข้อมูลเหล่านี้ไปแสดงในรูปแบบของแผนภูมิเส้นบนเว็บแอปพลิเคชัน โดยมีโครงสร้างของ Table และรายละเอียดดังต่อไปนี้

ER Diagram ของ Client Usage Table



รูปที่ 3.3 ER Diagram ของ Client Usage Table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 Data descriptions ของ Job Status Database

ชื่อฟิลด์	ประเภทข้อมูล	ประเภทคีย์	ความหมาย	ข้อกำหนด
ID	VARCHAR	PK	ไอดีซึ่งจะถูกสร้างขึ้นอัตโนมัติ เมื่อบันทึกข้อมูลใน Mysql เพื่อให้ฐานข้อมูลสามารถระบุถึงข้อมูลแต่ละตัว	NOT NULL AUTO_INCREMENT
Queue	VARCHAR		ค่าที่บ่งบอกถึงประเภทของแต่ละ Requested Data ซึ่งประกอบด้วย Regular, Large และ HMC	
Pending	INT		ค่าที่บ่งบอกถึงปริมาณ Requested Data ที่กำลังรอการประมวลผลที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน	
Running	INT		ค่าที่บ่งบอกถึงปริมาณ Requested Data ที่กำลังการประมวลผลบนที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน	
Time_record	DATETIME		เวลาที่มีการบันทึกข้อมูล	

สาเหตุที่เลือกใช้ Mysql Database ในการบันทึกข้อมูลมีดังต่อไปนี้

- เป็นระบบฐานข้อมูลแบบ OpenSource
- รองรับข้อมูลขนาดใหญ่
- นอกจากเว็บแอปพลิเคชันที่จะทำงานคู่กับระบบฐานข้อมูลนี้แล้ว ในบางเวลา ผู้ดูแลระบบจำเป็นต้องใช้ข้อมูลที่อยู่ในระบบฐานข้อมูลไปใช้การวิเคราะห์ในรูปแบบอื่นๆ การนำ MySQL Database ที่ใช้ภาษา SQL ในการดึงข้อมูล เช่นเดียวกับฐานข้อมูลหลัก ทำให้ง่ายต่อการใช้งานสำหรับผู้ดูแลระบบ

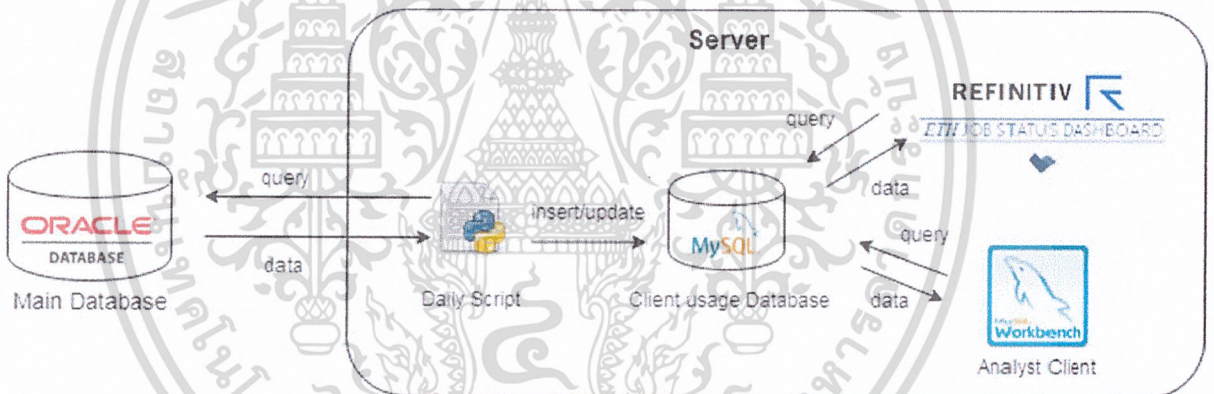
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3) การวิเคราะห์และออกแบบชุดคำสั่งสำหรับนำข้อมูลบันทึกลง Client usage Database

หลังจากการวิเคราะห์และออกแบบ Client usage Database สำหรับนำข้อมูลจากฐานข้อมูลหลักมาผ่านการประมวลผลและบันทึก ซึ่งชุดคำสั่งที่หน้าทีดังกล่าวคือชุดคำสั่ง Daily Script

Daily Script เป็น ชุดคำสั่ง ที่ใช้สำหรับการดึงข้อมูลรายละเอียดของ Requested Data จากฐานข้อมูลหลักบางส่วนมาผ่านการประมวลผล แล้วนำไปบันทึกลง Client usage Database เพื่อใช้งานคู่กับเว็บแอปพลิเคชันและสามารถให้ผู้ใช้และระบบสามารถนำข้อมูลไปวิเคราะห์ในรูปแบบอื่นต่อไปได้ ดังรูปที่ 3.4

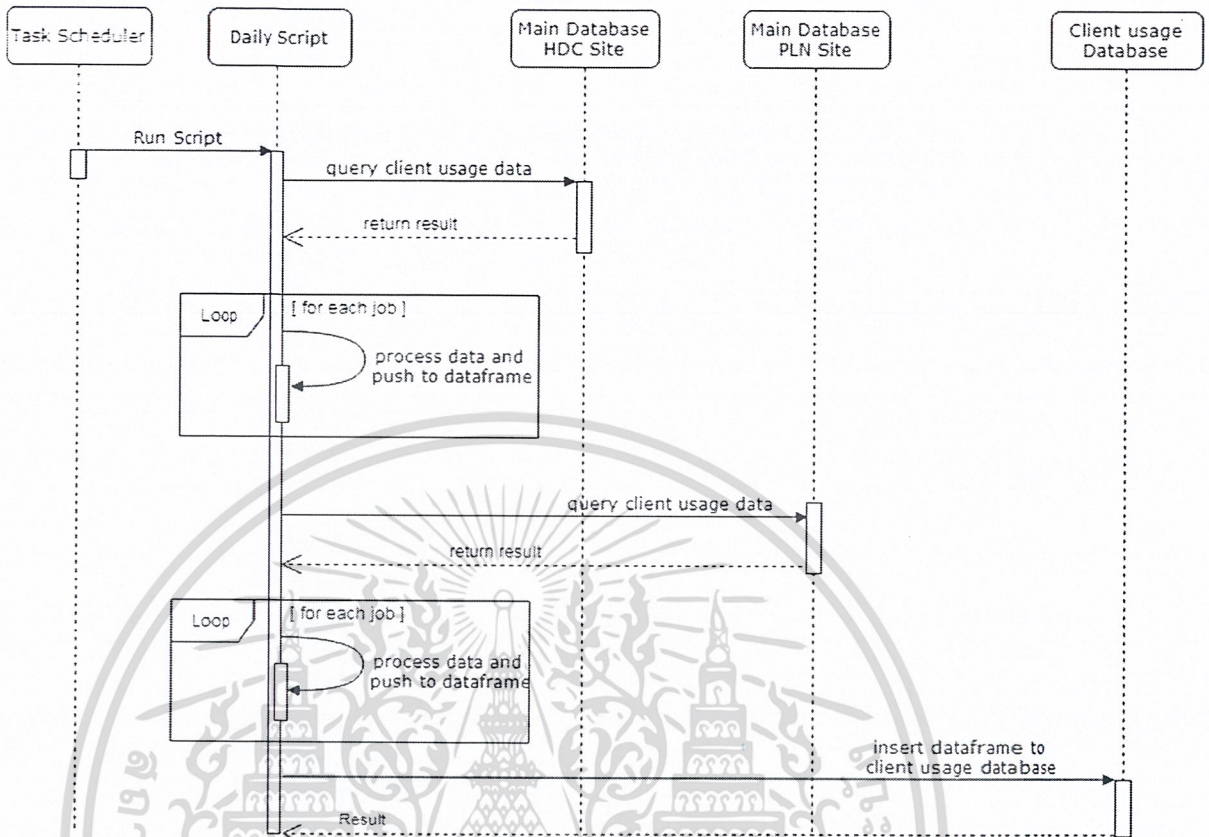
โดยชุดคำสั่งนี้จะถูกตั้งให้ทำงานทุกวันผ่าน Task Scheduler



รูปที่ 3.4 การวิเคราะห์การทำงานของ Daily Script และการนำไปใช้

ขั้นตอนการทำงานของชุดคำสั่ง Daily Script สามารถอธิบายด้วยแผนภาพความสัมพันธ์ระหว่างวัตถุ (Sequence Diagram) ซึ่งเป็นการออกแบบขั้นตอนการทำงานของ Daily Script เพื่อแสดงขั้นตอน เวลา และโปรแกรมที่เกี่ยวข้องกับการทำงาน ดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Sequence Diagram การทำงาน Daily Script

จากรูปที่ 3.5สามารถอธิบายได้ดังนี้

- เมื่อถึงเวลาที่ตั้งไว้ Task Scheduler จะสั่งให้ DailyScript ทำงานเพื่อดึงข้อมูล Requested Data ที่ทำงานเมื่อวานทั้งหมด
- Daily Script จะทำการดึงข้อมูลรายละเอียดจากฐานข้อมูลหลักบนฝั่ง HDC ServerSite ดังต่อไปนี้

○ JOB_ID	○ JS_DEQUEUE_TIME
○ USER_ID	○ YARN_ACCEPTED_TIME
○ RIC_DAY	○ JOB_START_TIME
○ JOB_STATUS	○ SPARK_START_TIME
○ JOB_SUBMITTED_TIME	○ JOB_FINISH_TIME
○ WS_QUEUE_TIME	○ MERGED_TIME

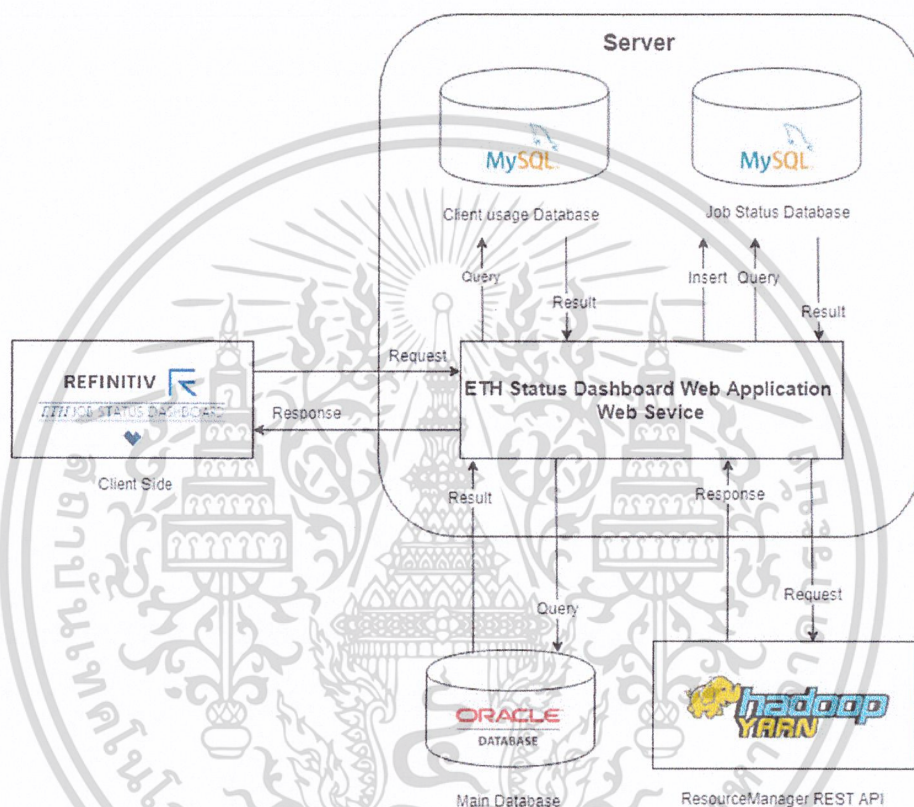
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำงานประมวลผลข้อมูลให้อยู่ในรูปแบบดังต่อไปนี้
 - JOB_ID
 - USER_ID
 - RIC_DAY
 - JOB_STATUS
 - JOB_SUBMITTED_TIME
 - WS_PROCESSING_TIME
 - สามารถคำนวณได้จาก “WS_QUEUE_TIME - JOB_SUBMITTED_TIME”
 - MQ_QUEUED_TIME
 - สามารถคำนวณได้จาก “JS_DEQUEUE_TIME - WS_QUEUE_TIME”
 - JS_PROCESSING_TIME
 - สามารถคำนวณได้จาก “YARN_ACCEPTED_TIME - JS_DEQUEUE_TIME”
 - YARN_QUEUED_TIME
 - สามารถคำนวณได้จาก “SPARK_START_TIME - YARN_ACCEPTED_TIME”
 - DRIVER_WAIT_TIME
 - สามารถคำนวณได้จาก “JOB_START_TIME - SPARK_START_TIME”
 - JOB_PROCESSING_TIME
 - สามารถคำนวณได้จาก “JOB_FINISH_TIME - JOB_START_TIME”
 - MERGED_TIME
 - สามารถคำนวณได้จาก “MERGED_TIME - JOB_FINISH_TIME”
 - TOTAL_TIME
 - สามารถคำนวณได้จาก “MERGED_TIME - JOB_SUBMITTED_TIME”
- นำข้อมูลของแต่ละ Requested Data ลง Dataframe
- ทำการดึงข้อมูลรายละเอียดจากฐานข้อมูลหลักบนฝั่ง PLN ServerSiteและประมวลผลให้อยู่ในรูปแบบเช่นเดียวกับ HDC Server Site
- นำข้อมูลทั้งหมดบน Dataframe บันทึกลงบน Client usage Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4) การวิเคราะห์และออกแบบ Web Service

Web Serviceของเว็บแอปพลิเคชันพัฒนาด้วย Flask Framework ของภาษา Python โดยจะทำหน้าที่ในการต่อกับแหล่งข้อมูลและฐานข้อมูลซึ่งประกอบด้วย YARN Resource Manager REST API, ฐานข้อมูลหลัก , Client usage Database และ Job Status Database เพื่อดึงข้อมูลและประมวลผลแล้วส่งต่อไปยังฝั่งผู้ใช้งานเพื่อแสดงผลดังรูปที่ 3.6

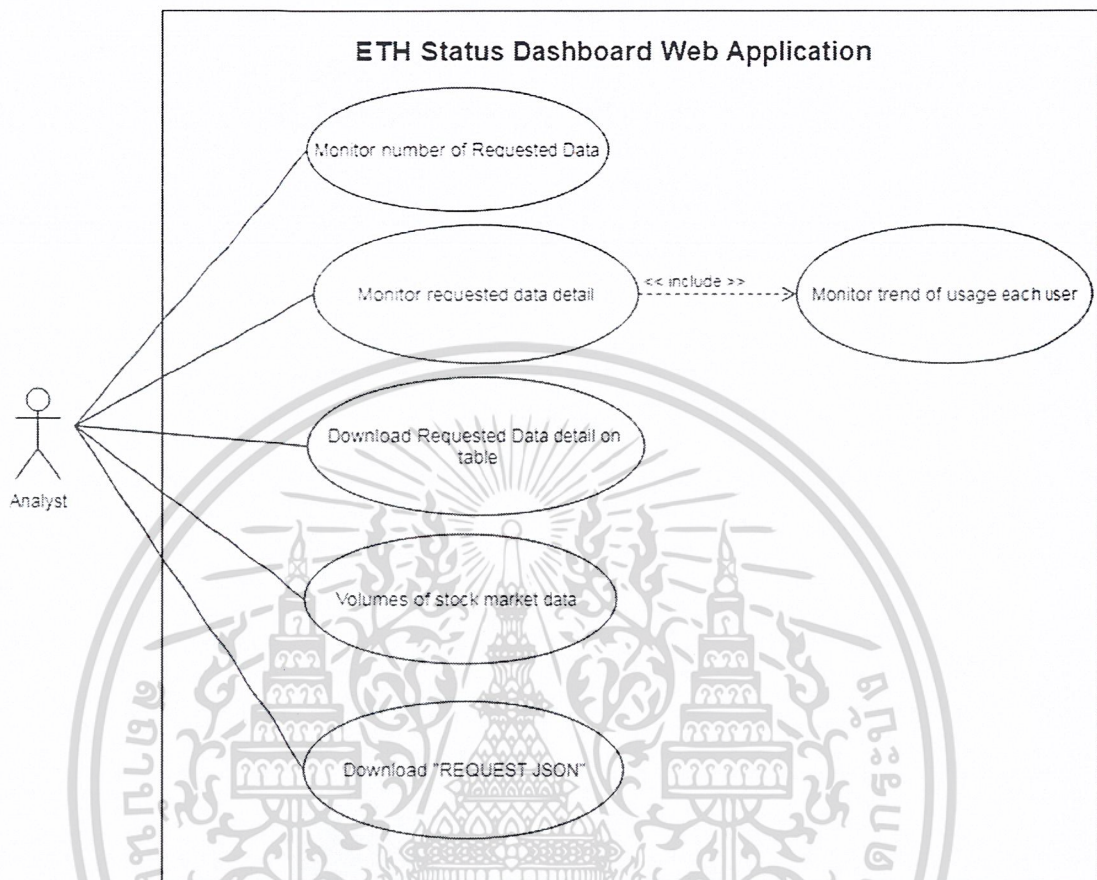


รูปที่ 3.6 สถาปัตยกรรมของ Web Applications for Data Analysis of Customer Requested Data

ขั้นตอนการทำงานของเว็บแอปพลิเคชันสามารถอธิบายด้วยแผนภาพ (Diagram) เพื่อให้เห็นภาพรวมการทำงาน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.1)แผนภาพการทำงานของผู้ใช้ระบบ (Use Case Diagram)



รูปที่ 3.7 Use Case Diagram การทำงานเว็บแอปพลิเคชัน

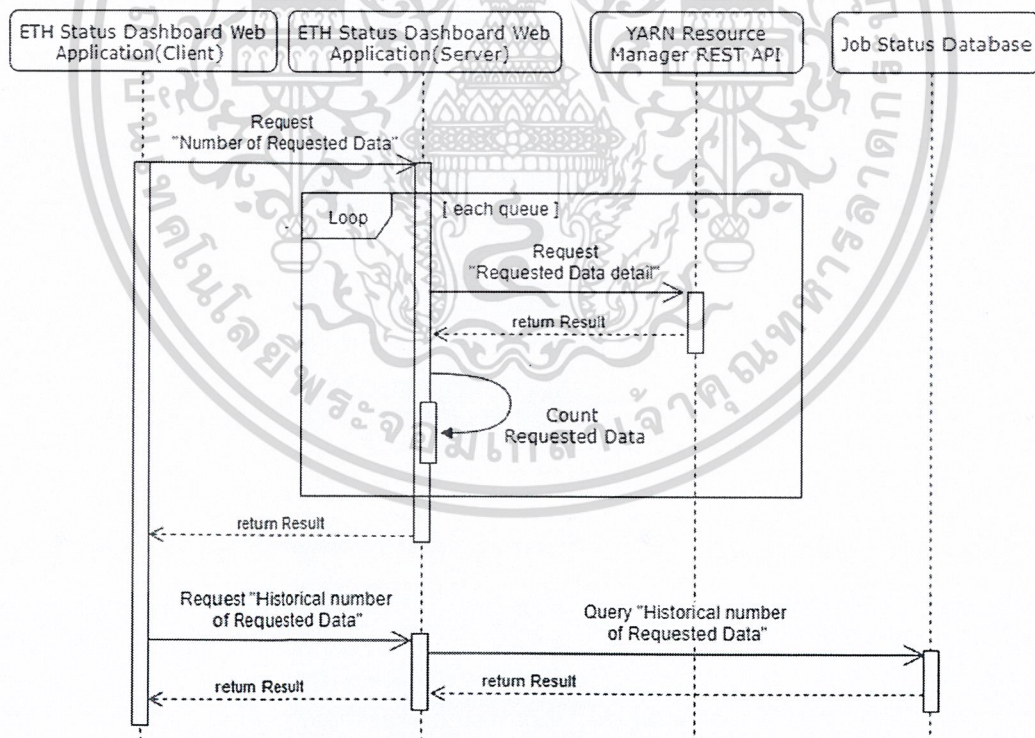
จากรูปที่ 3.7 สามารถอธิบายได้ดังนี้

- Monitor number of Requested Data คือ ผู้ดูแลระบบสามารถเข้าสู่หน้าเว็บแอปพลิเคชันผ่าน IP ของ Server เพื่อตรวจสอบปริมาณของ Requested Data
- Monitor Requested Data Detail คือ ผู้ดูแลระบบสามารถดูรายละเอียด Requested Data จากฐานข้อมูลหลักและรวมถึงปริมาณ Requested Data ทั้งหมดแต่ละวัน ,ขนาดข้อมูล Requested Data (RIC DAY)ทั้งหมดที่ขอแต่ละวัน และค่าเฉลี่ย เวลาของ Requested Data ที่ทำงานในระบบETH (TOTAL_TIME) ทั้งหมดแต่ละวัน ในระยะเวลา 30 วันล่าสุดของผู้ใช้งานที่เป็นเจ้าของ Requested Data จากClient usage database โดยการกดปุ่มที่ชื่อ Queue ใน Server Site และ State ที่ต้องการทราบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Download Requested Data Detail on table คือ ผู้ดูแลระบบสามารถดาวน์โหลดตารางข้อมูลรายละเอียดของ Requested Data ที่แสดงอยู่บนหน้าเว็บแอปพลิเคชัน ในรูปแบบของไฟล์ CSV หรือ Excel โดยทำงานเมื่อกดปุ่มตามชื่อประเภทไฟล์ที่ต้องการบริเวณส่วนบนของตารางข้อมูล
- Volumes of Stock market data คือ ผู้ดูแลระบบสามารถให้แสดงปริมาณข้อมูลตลาดที่ขอทั้งหมด , ช่วงเวลาที่เริ่มขอ และช่วงเวลาสุดท้ายที่ขอ ของ Requested Data แต่ละตัวจาก ฐานข้อมูลหลัก โดยทำงานเมื่อกดปุ่ม “Request Data” เพื่อขอข้อมูล
- Download “Request JSON” คือ ผู้ดูแลระบบสามารถดาวน์โหลด RequestJSON ซึ่งเก็บรายละเอียดการขอข้อมูลภายใน Requested Data แต่ละตัวจาก ฐานข้อมูลหลัก โดยทำงานเมื่อกดปุ่ม “Download”

3.1.4.2) แผนภาพแสดงทำงานแบบลำดับปฏิสัมพันธ์ (Sequence Diagram)



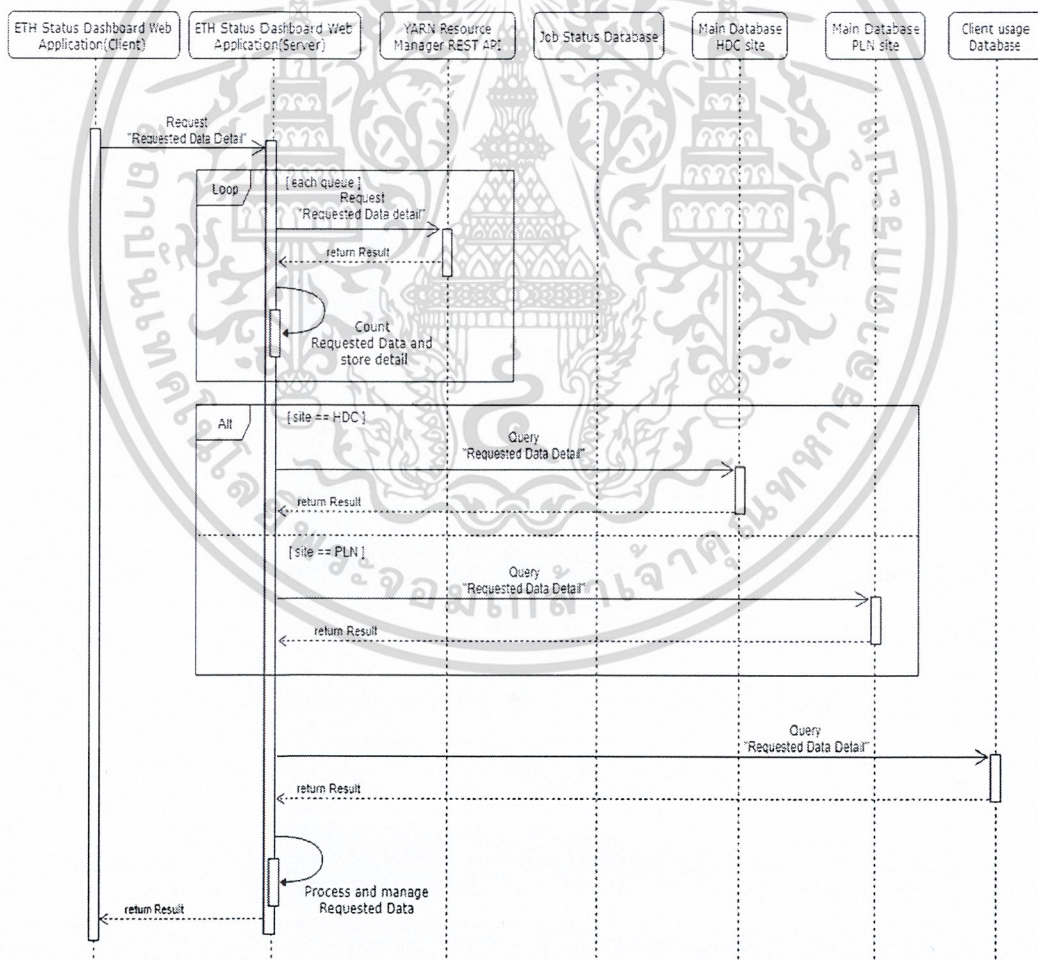
รูปที่ 3.8 Sequence Diagram ของการเข้าสู่เว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8 สามารถอธิบายได้ดังนี้

เมื่อมีการเข้าสู่เว็บแอปพลิเคชัน ผู้ใช้งาน จะมีการขอข้อมูลปริมาณ Requested Data ที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน ไปที่ฝั่ง Web Service และฝั่ง Web Service จะทำการเชื่อมต่อกับ YARN Resource Manager REST API เพื่อรับข้อมูล Requested Data ที่อยู่บนระบบ ETH เมื่อฝั่ง Web Service ได้รับข้อมูลจะทำการนับปริมาณ Requested Data แต่ละ State และ Queue แล้วส่งให้ ฝั่งผู้ใช้งาน แสดงผล

เมื่อมีการเข้าสู่ เว็บแอปพลิเคชัน ผู้ใช้งาน จะมีการขอประวัติข้อมูลปริมาณ Requested Data ณ เวลาต่างๆ ทั้งหมดที่เคยบันทึกไว้ ไปที่ฝั่ง Web Service และฝั่ง Web Service จะทำการดึงข้อมูลที่บันทึกไว้ทั้งหมดกับ Job Status Database เมื่อฝั่ง Web Service ได้รับข้อมูลจะจัดรูปแบบข้อมูลแล้วส่งต่อให้ฝั่ง ผู้ใช้งาน แสดงผลในรูปแบบของแผนภูมิ



รูปที่ 3.9 Sequence Diagram ของการขอรายละเอียด Requested Data

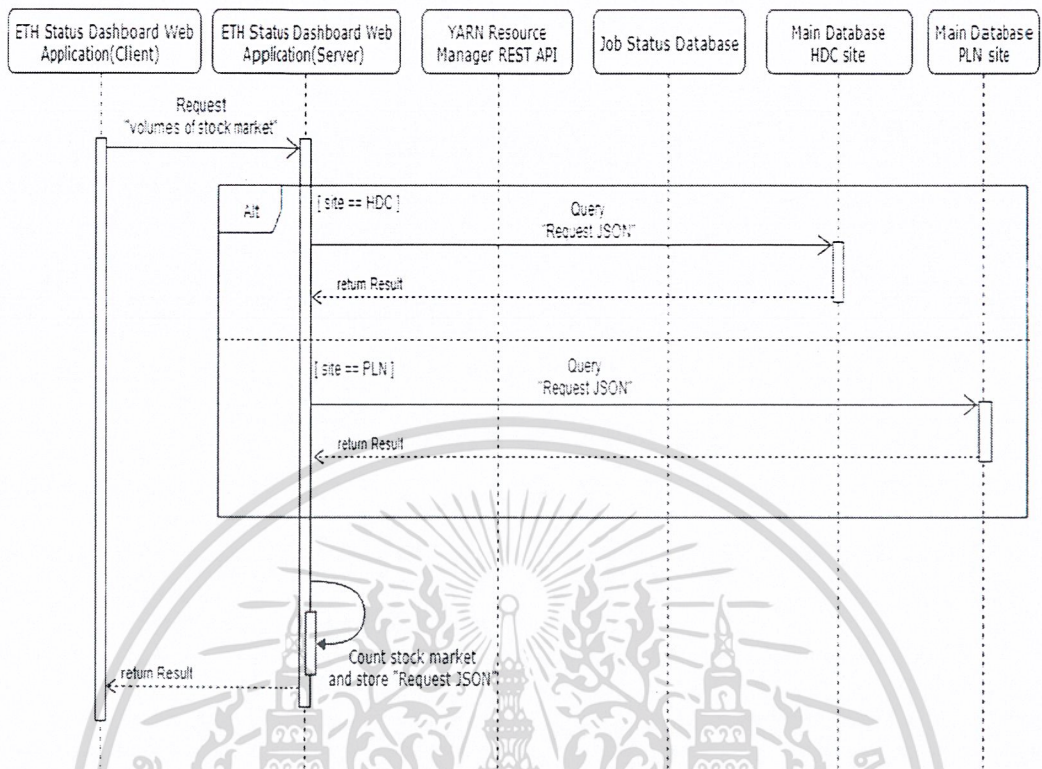
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.9 สามารถอธิบายได้ดังนี้

เมื่อ ผู้ใช้งาน มีการขอรายละเอียด Requested Data ในแต่ละ State และ Queue ไปยัง ฟังก์ชัน Web Service และฟังก์ชัน Web Service จะมีการขอข้อมูลปริมาณ Requested Data ที่กำลังทำงานอยู่ที่อยู่บน HADOOP Cluster ของระบบ ETH ณ ปัจจุบัน กับ YARN Resource Manager REST API อีกครั้งแต่จะมีการเก็บรายละเอียด Requested Data เอาไว้ ต่อจากนั้นจะมีการตรวจสอบ Server Site ที่ Requested Data ทำงานอยู่ แล้วฟังก์ชัน Web Service จะทำการดึงข้อมูลพื้นฐานข้อมูลหลัก ที่ Server Site ตรงตามฟังก์ชันผู้ใช้งาน ขอมมา เมื่อได้รับข้อมูล Requested Data ฟังก์ชัน Web Service จะทำการนำข้อมูลจากทั้ง YARN Resource Manager REST API และ ข้อมูลหลัก มาหาความสัมพันธ์กันและประมวลผลให้อยู่ในรูปแบบที่เหมาะสมแล้วส่งต่อไปให้ผู้ใช้งาน แสดงผล

เมื่อ ผู้ใช้งาน มีการขอรายละเอียด Requested Data ในแต่ละ State และ Queue ไปยัง ฟังก์ชัน Web Service และฟังก์ชัน Web Service จะมีการขอข้อมูลปริมาณ Requested Data ทั้งหมด , ขนาดข้อมูลที่ขอ (RIC DAY) ทั้งหมด และค่าเฉลี่ย เวลาของ Requested Data ที่ทำงานในระบบ ระบบ ETH(TOTAL_TIME) ทั้งหมด แต่ละวัน ในระยะเวลา 30 วันล่าสุดของผู้ใช้งานที่เป็นเจ้าของ Requested Data ที่ถูกขอรายละเอียดไปที่ฟังก์ชัน Web Service และฟังก์ชัน Web Service จะทำการดึงข้อมูลที่ Client usage Database แทนการดึงข้อมูลจากฐานข้อมูลหลัก โดยตรงเนื่องจากเป็นการขอข้อมูลจากช่วงของข้อมูลที่ค่อนข้างกว้าง จนทำให้กระบวนการทำงานฐานข้อมูลหลักมากเกินไปเมื่อได้รับข้อมูลฟังก์ชัน Web Service จะจัดรูปแบบข้อมูลแล้วส่งต่อไปให้ผู้ใช้งาน แสดงผลในรูปแบบของแผนภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

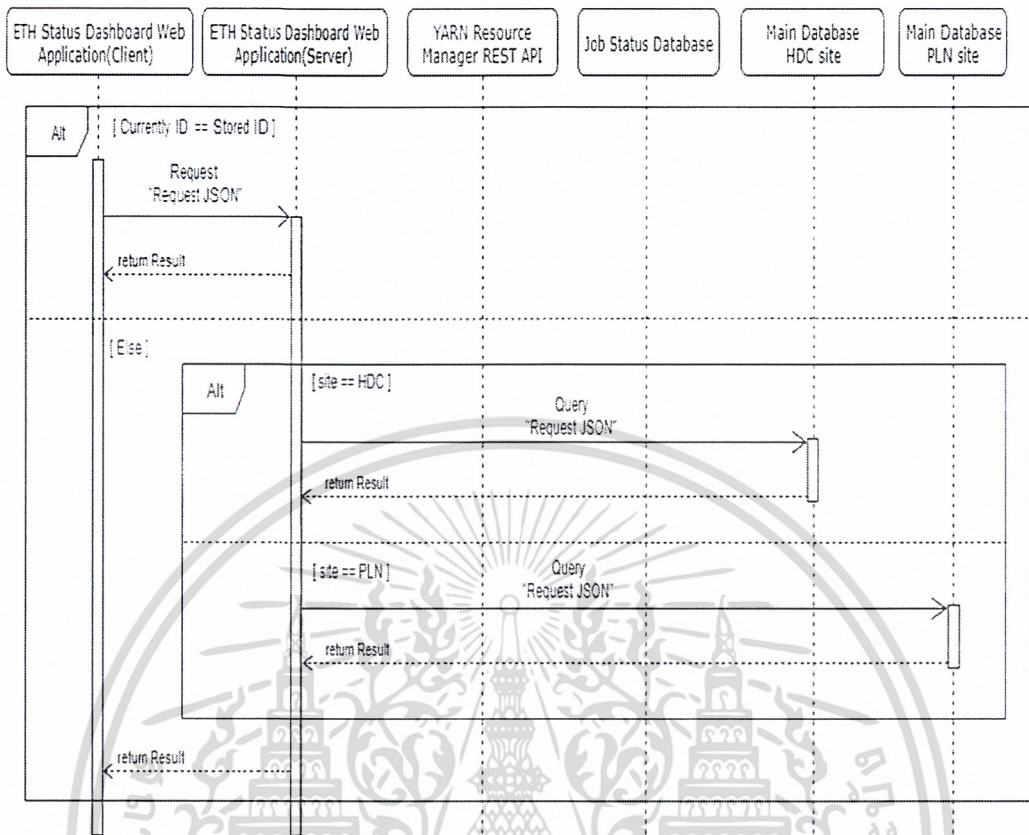


รูปที่ 3.10 Sequence Diagram ของการขอข้อมูลปริมาณตลาดภายใน Requested Data

จากรูปที่ 3.10 สามารถอธิบายได้ดังนี้

เมื่อฝั่งผู้ใช้งาน มีการขอปริมาณข้อมูลตลาด ช่วงเวลาที่เริ่มขอ และช่วงเวลาสุดท้าย จะมีการตรวจสอบ Server Site ที่ Requested Data ทำงาน แล้วส่งค่าข้อมูลไปยังฝั่ง Web Service และฝั่ง Web Service จะทำการดึงข้อมูล RequestJSON ซึ่งเก็บรายละเอียดการขอข้อมูลภายใน Requested Data ที่ฐานข้อมูลหลักซึ่ง ServerSite ตรงตามที่ ฝั่งผู้ใช้งาน ขอมมา เมื่อฝั่ง Web Service ได้รับข้อมูล จะประมวลผลเพื่อนับปริมาณข้อมูลตลาด จัดรูปแบบข้อมูลและจัดเก็บแล้วส่งต่อให้ฝั่งผู้ใช้งานแสดงผล

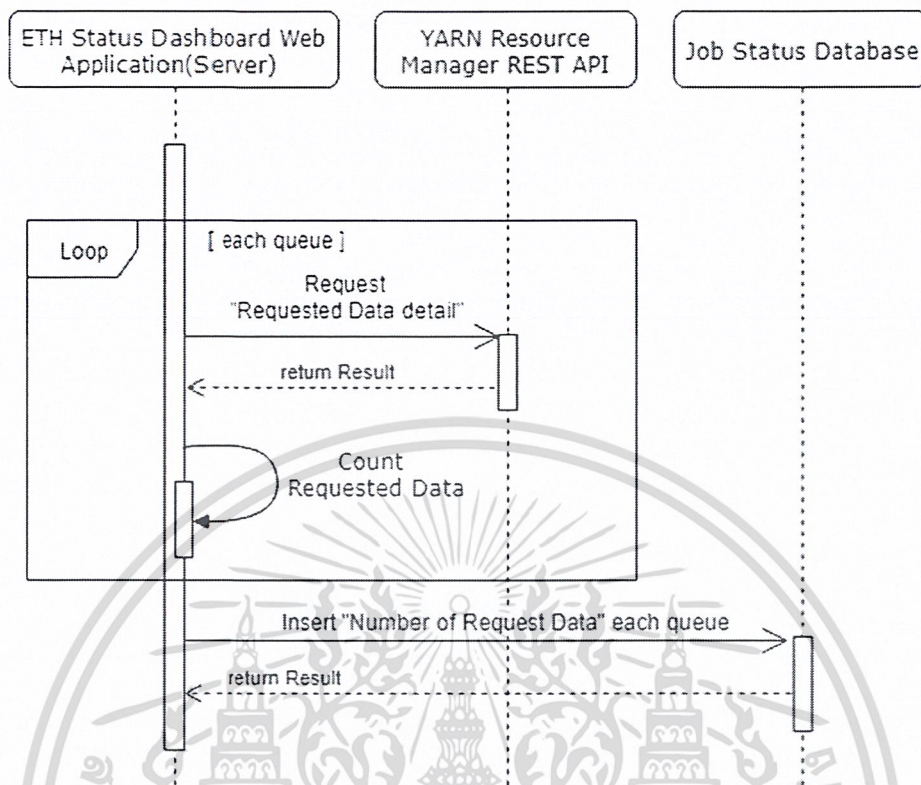
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 Sequence Diagram ของการขอตัวนำโหลดข้อมูลภายใน Requested Data

จากรูปที่ 3.11 สามารถอธิบายได้ดังนี้

เมื่อฝั่งผู้ใช้งาน มีการขอตัวนำโหลด RequestJSON ไปยังฝั่ง Web Service และฝั่ง Web Service จะทำการตรวจสอบว่าข้อมูลที่เก็บไว้ตรงกับที่ผู้ใช้งานต้องการหรือไม่ ถ้าตรงกับที่ผู้ใช้งาน ต้องการจะนำข้อมูลที่เก็บไว้ ส่งกลับให้ ฝั่งผู้ใช้งาน ตัวนำโหลด ถ้าไม่ตรงจะทำการติดต่อกับ ฐานข้อมูลหลักเพื่อขอข้อมูลมาประมวลผลและส่งต่อไปยังฝั่งผู้ใช้งานตัวนำโหลด



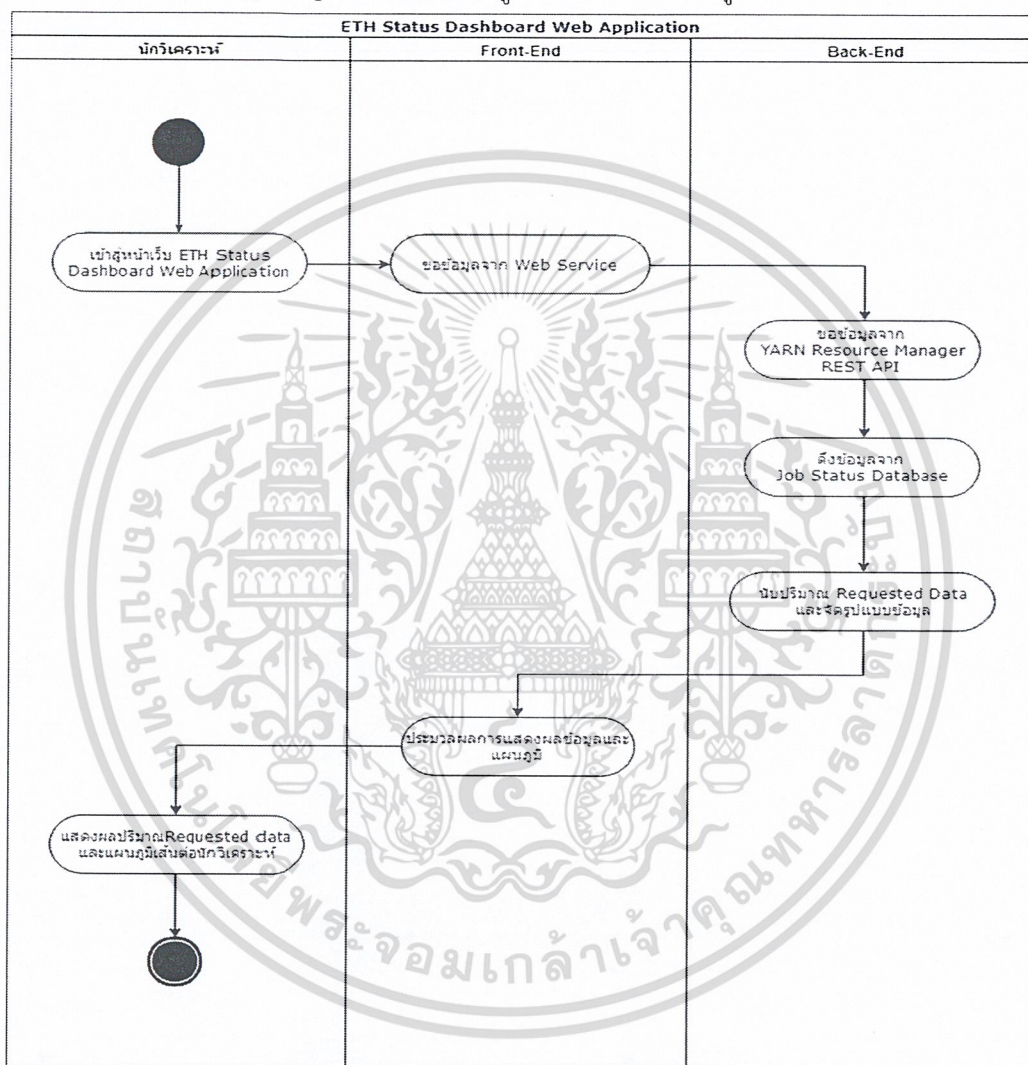
รูปที่ 3.12 Sequence Diagram ของการเก็บประวัติปริมาณ Requested Data จากรูปที่ 3.12 สามารถอธิบายได้ดังนี้

ในทุกๆ 5 นาทีฝั่ง Web Service จะทำการเชื่อมต่อกับ YARN Resource Manager REST API เพื่อรับข้อมูล Requested Data ที่กำลังทำงานบน HADOOP Cluster ของระบบ ETH ณ ปัจจุบันเมื่อฝั่ง Web Service ได้รับข้อมูลจะทำการนับปริมาณ Requested Data แต่ละ State และ Queue แล้วนำข้อมูลไปบันทึกบน Job Status Database

3.1.4.3) แผนภาพกิจกรรม (Activity Diagram)

จะแสดงกิจกรรมที่เกิดขึ้นในลักษณะกระแสการไหลของการทำงานทั้งหมดบนหน้าเว็บแอปพลิเคชัน ดังนี้

- Activity Diagram ของการเข้าสู่เว็บแอปพลิเคชัน ดังรูปที่ 3.13



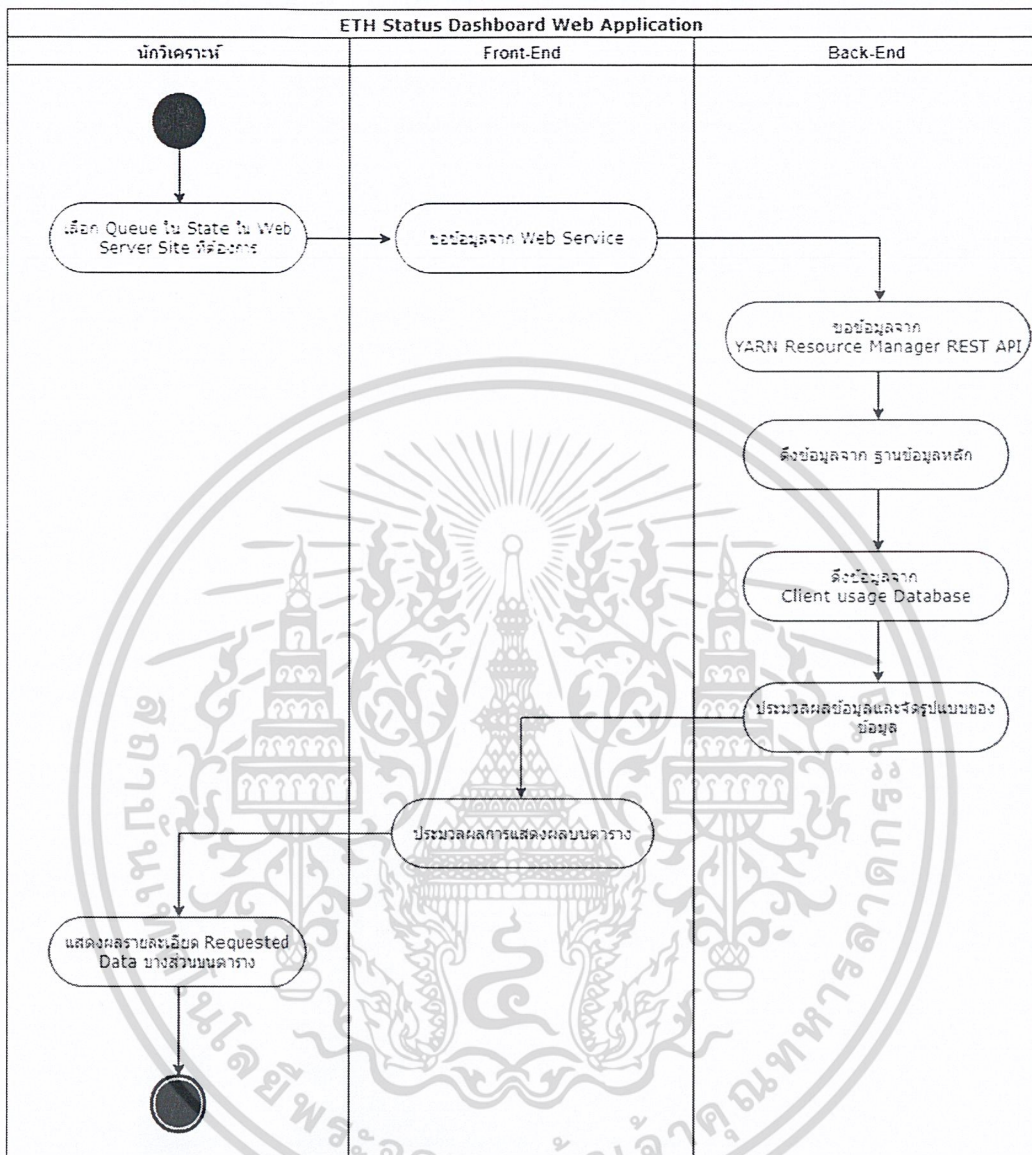
รูปที่ 3.13 Activity Diagram ของการเข้าสู่เว็บแอปพลิเคชัน

จากรูปที่ 3.13 สามารถอธิบายได้ดังนี้

เมื่อผู้ใช้งานเข้าสู่เว็บแอปพลิเคชันในฝั่งผู้ใช้งานจะทำการส่งคำขอข้อมูลไปยังฝั่ง Web Service หลังจากนั้นฝั่ง Web Service จะทำการดึงข้อมูลจาก YARN Resource Manager REST API และ Job Status Database มาประมวลผลให้ได้รูปแบบข้อมูลที่ต้องการ แล้วส่งต่อไปยังฝั่งผู้ใช้งานจัดการแสดงผลข้อมูลในรูปแบบของปริมาณและแผนภูมิแก่ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยผู้จัดทำเอกสารนี้ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Activity Diagram ของการขอรายละเอียด Requested Data ดังรูปที่ 3.14



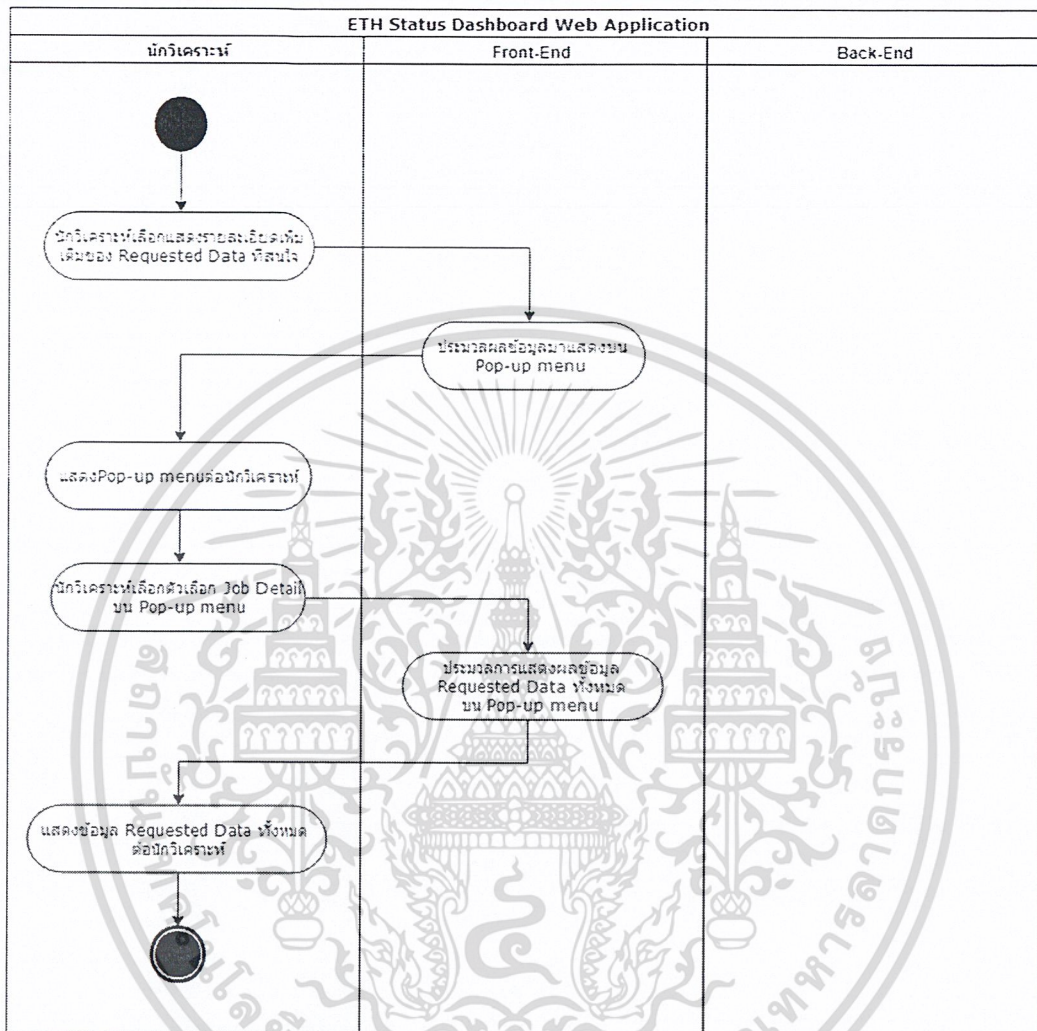
รูปที่ 3.14 Activity Diagram ของการขอรายละเอียด Requested Data

จากรูปที่ 3.14 สามารถอธิบายได้ดังนี้

เมื่อผู้ใช้งานกดขอรายละเอียด Requested Data ในฝั่งผู้ใช้งานจะทำการส่งคำขอข้อมูลไปยังฝั่ง Web Service หลังจากนั้นฝั่ง Web Service จะทำการดึงข้อมูลจาก YARN Resource Manager REST API , Client Usage Database และ ฐานข้อมูลหลัก มาประมวลผลให้ได้รูปแบบข้อมูลที่ต้องการ แล้วส่งต่อไปยังฝั่งผู้ใช้งานจัดการแสดงผลข้อมูลรายละเอียดหลักให้อยู่ในรูปแบบตารางข้อมูลและรายละเอียดเพิ่มเติมในส่วนของ pop-up แก่ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Activity Diagram ของการดูรายละเอียด Requested Data ทั้งหมด ดังรูปที่ 3.15



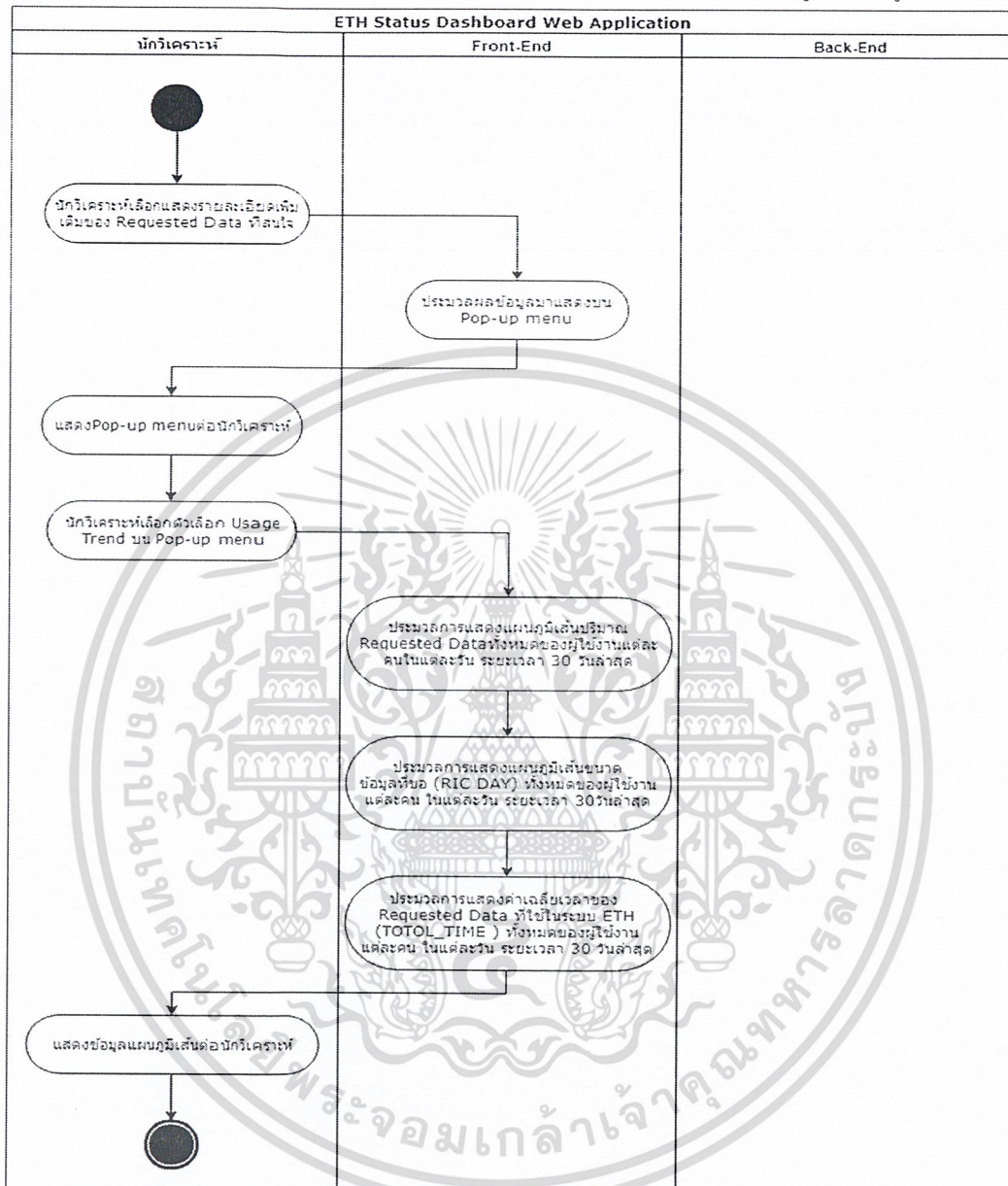
รูปที่ 3.15 Activity Diagram ของการดูรายละเอียด Requested Data ทั้งหมด

จากรูปที่ 3.15 สามารถอธิบายได้ดังนี้

เมื่อผู้ใช้งานต้องการดูรายละเอียดเพิ่มเติมของแต่ละ Requested Data แล้วกดปุ่มขอรายละเอียดเพิ่มเติม เว็บแอปพลิเคชันฝั่งผู้ใช้งานจะทำการประมวลผลนำรายละเอียดทั้งหมดของ Requested Data ไปแสดงบน pop-up ในรูปแบบของตารางข้อมูลส่วนของแถบ “Job Detail” แก่ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Activity Diagram ของการดูรายละเอียดการใช้งานระบบของลูกค้า ดังรูปที่ 3.16

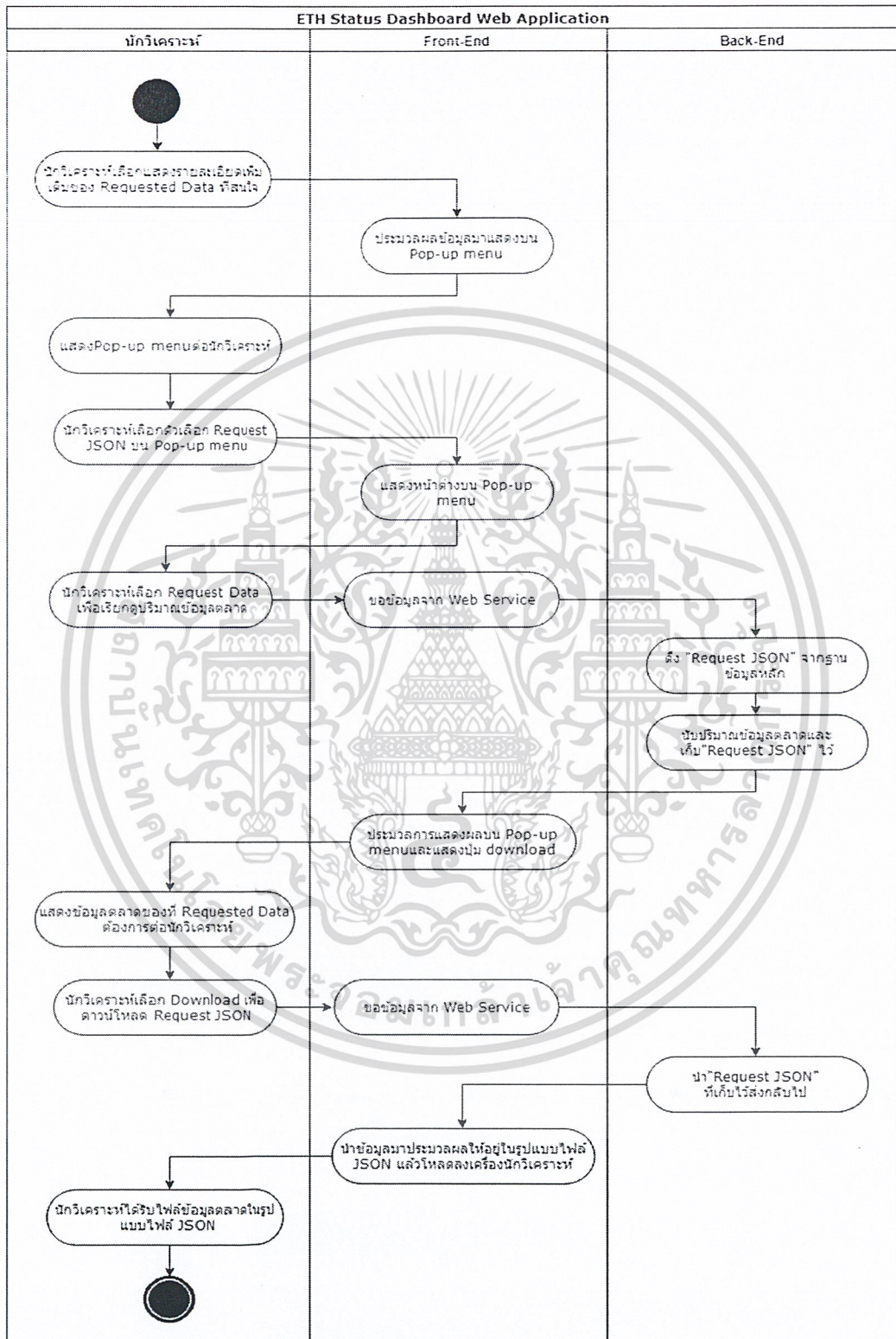


รูปที่ 3.16 Activity Diagram ของการดูรายละเอียดการใช้งานระบบของลูกค้า

จากรูปที่ 3.16 สามารถอธิบายได้ดังนี้

เมื่อผู้ใช้งานต้องการดูประวัติการใช้งานของลูกค้าที่ขอ Requested Data แต่ละตัว ในช่วงระยะเวลา 1 เดือนล่าสุด แล้วกดปุ่มขอรายละเอียดเพิ่มเติม เว็บแอปพลิเคชันฝั่งผู้ใช้งานจะทำการประมวลผลนำประวัติการใช้งานของลูกค้าที่เป็นเจ้าของ Requested Data ในระยะเวลา 1 เดือนล่าสุดไปแสดงบน pop-up ในรูปแบบของแผนภูมิเส้นส่วนของแถบ “Usage Trend” แก่ผู้ใช้งาน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Activity Diagram ของการขอข้อมูลภายใน Requested Data ดังรูปที่ 3.17



รูปที่ 3.17 Activity Diagram ของการขอข้อมูลภายใน Requested Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.17 สามารถอธิบายได้ดังนี้

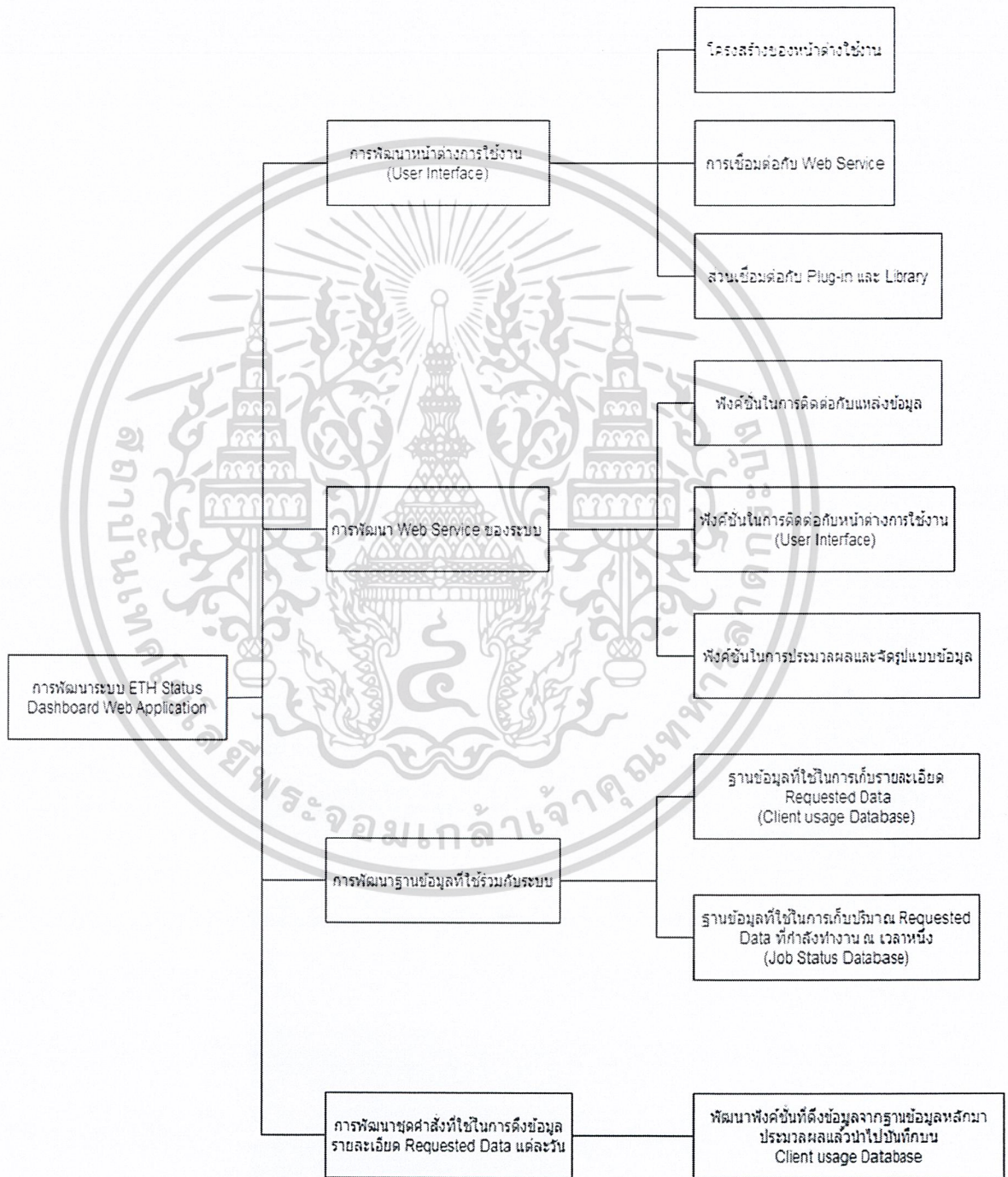
เมื่อผู้ใช้งานผู้ใช้งานต้องการดูข้อมูลปริมาณตลาดและรายละเอียดตลาดที่ขอมายภายในแต่ละ Requested Data แล้วกดปุ่มขอรายละเอียดเพิ่มเติม เว็บแอปพลิเคชันฝั่งผู้ใช้งานจะทำการประมวลผลแสดงบน pop-up ในส่วนของแถบ “Usage Trend” แก่ผู้ใช้งานผู้ใช้งานต้องทำการกดปุ่ม “Request Data” เพื่อให้เว็บแอปพลิเคชันฝั่งผู้ใช้งานทำการส่งคำขอข้อมูลไปยังฝั่ง Web Service หลังจากนั้นฝั่ง Web Service จะทำการดึงข้อมูลจาก ฐานข้อมูลหลัก มาทำการประมวลผลให้ได้ข้อมูลปริมาณตลาดภายใน Requested Data แล้วเพื่อส่งกลับไปให้ฝั่งผู้ใช้งานแสดงผลแก่ผู้ใช้งานขณะเดียวกันปุ่มดาวน์โหลดจะปรากฏขึ้นมา เมื่อผู้ใช้งานกดขอดาวน์โหลดรายละเอียดตลาด เว็บแอปพลิเคชันฝั่งผู้ใช้งานทำการส่งคำขอข้อมูลไปยังฝั่ง Web Service หลังจากนั้นฝั่ง Web Service จะทำการทำรายละเอียดตลาดที่เก็บไว้ส่งไปให้ฝั่งผู้ใช้งาน แล้วฝั่งผู้ใช้งานจะทำการประมวลผลให้อยู่ในรูปของไฟล์ แล้วทำการดาวน์โหลดลงเครื่องผู้ใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การพัฒนาระบบเว็บแอปพลิเคชัน

จากขั้นตอนการศึกษาการทำงานของระบบ Elektron Tick History (ETH) ขั้นตอนการวิเคราะห์และออกแบบระบบของเว็บแอปพลิเคชันทำให้ผู้พัฒนาทราบถึงขั้นตอนต่างๆในการพัฒนาระบบ ซึ่งสามารถแบ่งองค์ประกอบได้ตามแผนผัง ดังรูปที่ 3.18



รูปที่ 3.18 แผนผังองค์ประกอบการพัฒนาเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิใช่ผู้ผู้ให้เห็นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.18 สามารถอธิบายได้ดังนี้

ในส่วนของการพัฒนาเว็บแอปพลิเคชันสามารถแบ่งออกได้เป็น 4 ส่วนใหญ่

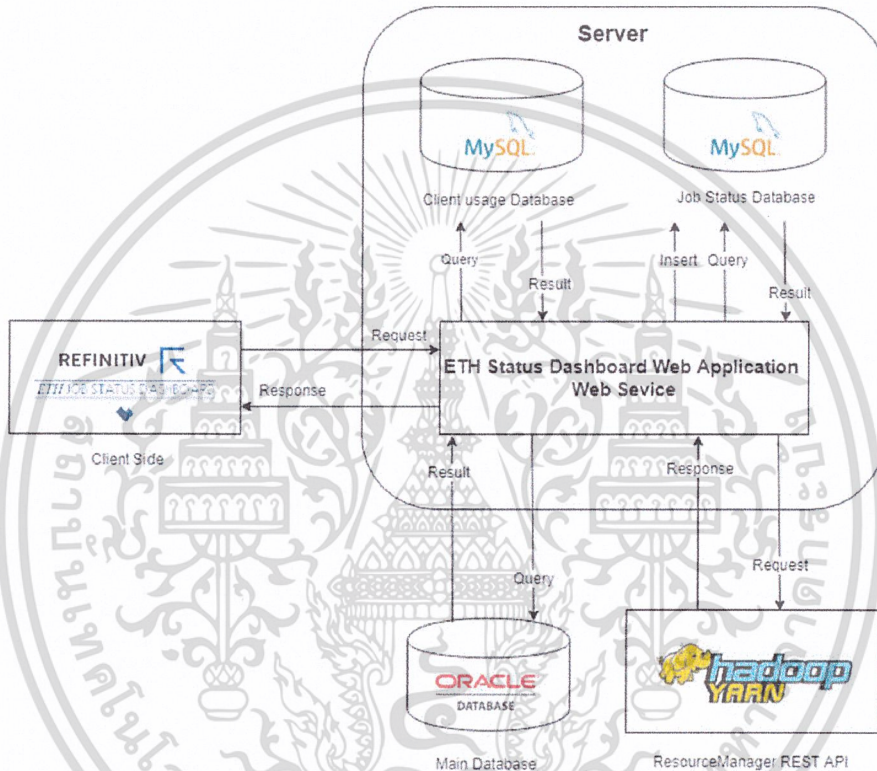
- 1) การพัฒนาหน้าตาการใช้งาน
 - 1.1. โครงสร้างของหน้าตาการใช้งาน
 - พัฒนาขึ้นด้วยการใช้ภาษา HTML ,CSS ,JAVASCRIPT และ BOOTSTRAP FRAMRWORK
 - 1.2. การเชื่อมต่อกับ Web Service
 - พัฒนาขึ้นด้วยการใช้ภาษา JAVASCRIPT และ jQuery FRAMRWORK
 - 1.3. ส่วนเชื่อมต่อกับ Plug-in และ Library
 - เชื่อมต่อกับ Plug-in และ Library ภายนอกด้วยภาษา HTML
- 2) การพัฒนา Web Service ของระบบ
 - 2.1. ฟังก์ชันในการติดต่อกับแหล่งข้อมูล
 - เชื่อมต่อกับฐานข้อมูลด้วย Library ของภาษา Python ที่มีชื่อว่า requests cx_oracle และ mysql-connector
 - 2.2. ฟังก์ชันในการติดต่อกับหน้าตาการใช้งาน
 - เชื่อมต่อกับหน้าตาผู้ใช้งานข้อมูลด้วย Framework ของภาษา Python ที่มีชื่อว่า Flask Framework
 - 2.3. ฟังก์ชันในการประมวลผลและจัดรูปแบบข้อมูล
 - ประมวลผลและจัดรูปแบบข้อมูลด้วย Library ของภาษา Python ที่มีชื่อว่า ujon datetime และ jsonpath
- 3) การพัฒนาฐานข้อมูลที่ใช้ร่วมกับระบบ
 - 3.1. ฐานข้อมูล Client Usage Database
 - ประเภทของฐานข้อมูลเป็นประเภท RDMS
 - 3.2. ฐานข้อมูล Job Status Database
 - ประเภทของฐานข้อมูลเป็นประเภท RDMS
- 4) การพัฒนาชุดคำสั่งในการนำข้อมูลมาเก็บที่ Client Usage Database
 - พัฒนาด้วยภาษา Python โดยใช้ Library ที่มีชื่อว่า pandas ช่วยในการจัดการข้อมูลและใช้ Library ที่มีชื่อว่า multiprocessing ช่วยในการเพิ่มความเร็วในการทำงานแบบ Process-based parallelism

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

ผลการดำเนินงานของการพัฒนาเว็บแอปพลิเคชันเพื่อใช้ในการวิเคราะห์และตรวจสอบปริมาณและรายละเอียดของ Requested Data ที่ทำงานอยู่ระบบ ETHสามารถอธิบายผลการดำเนินงานได้ดังต่อไปนี้



รูปที่ 4.1 สถาปัตยกรรมของ Web Application for Data Analysis of Customer Requested Data

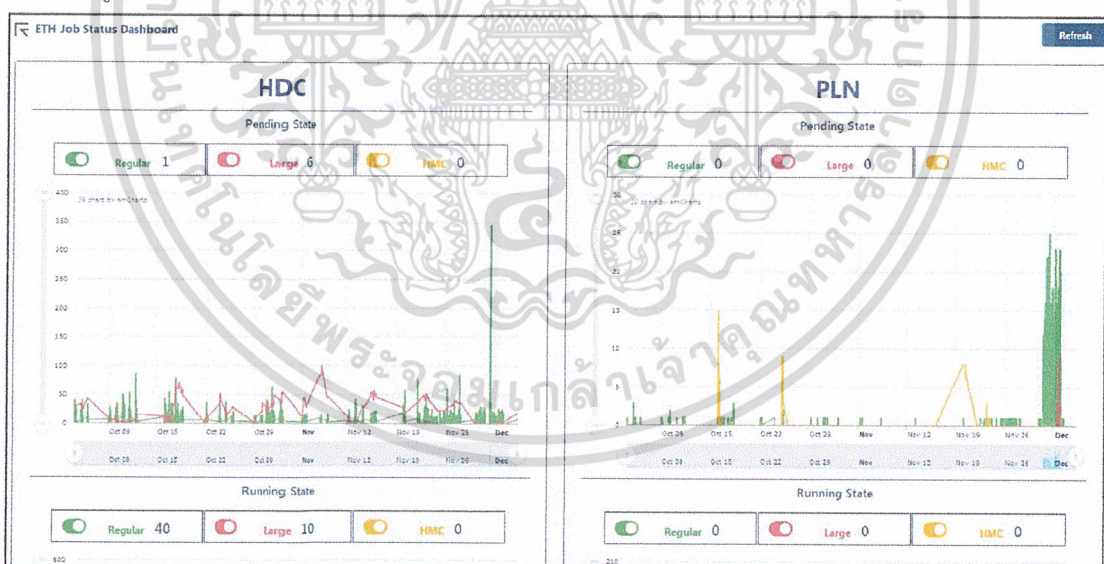
จากรูปที่ 4.1 ทางผู้พัฒนาได้มีการพัฒนา Web Service ของเว็บแอปพลิเคชันที่ทำการเชื่อมต่อกับฐานข้อมูลหลัก, YARN Resource Manager API , Client usage database และ Job Status database เพื่อนำข้อมูลจากทั้ง 4 แหล่งนี้มาประมวลผลแล้วส่งต่อให้เว็บแอปพลิเคชันฝั่งผู้ใช้งานแสดงข้อมูลเพื่อให้ผู้ดูแลระบบนำไปตรวจสอบและวิเคราะห์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



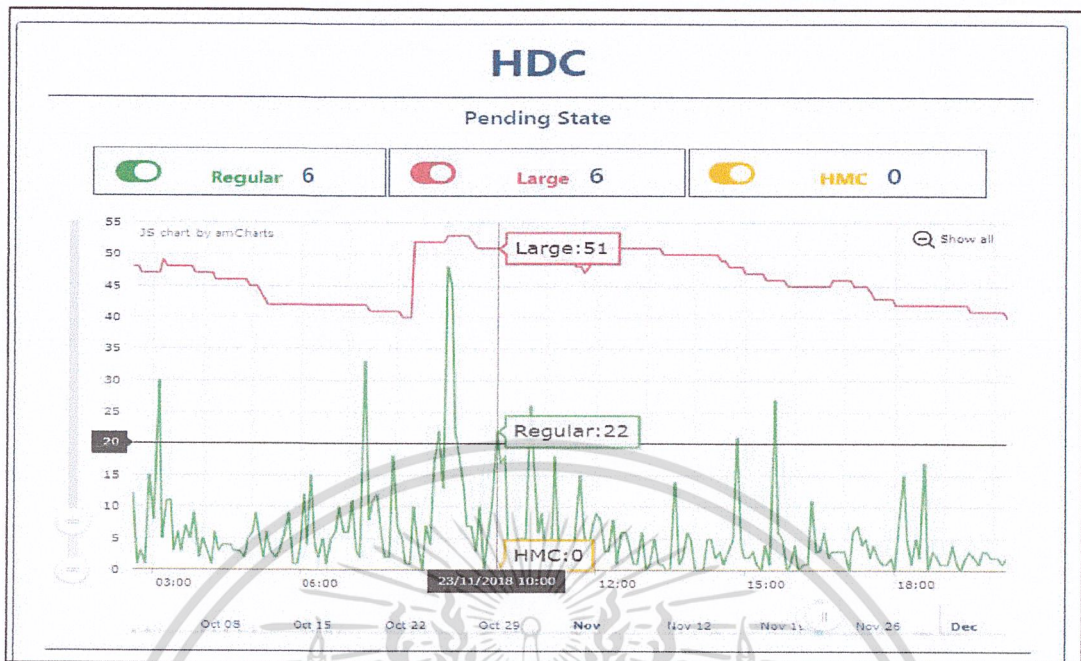
รูปที่ 4.2 หน้าการรอเข้าสู่เว็บแอปพลิเคชัน

จากรูปที่ 4.2 เป็นหน้าการรอเข้าสู่เว็บแอปพลิเคชัน ในส่วนของผู้ใช้งาน เนื่องจากต้องรอ Web Service ทำการนับปริมาณ Requested Data จาก YARN Resource Manager API และรอการขนส่งข้อมูลประวัติปริมาณ Requested Data จาก Job Status database เพื่อนำมาแสดงบนส่วนของผู้ใช้งานเสร็จสิ้นก่อน



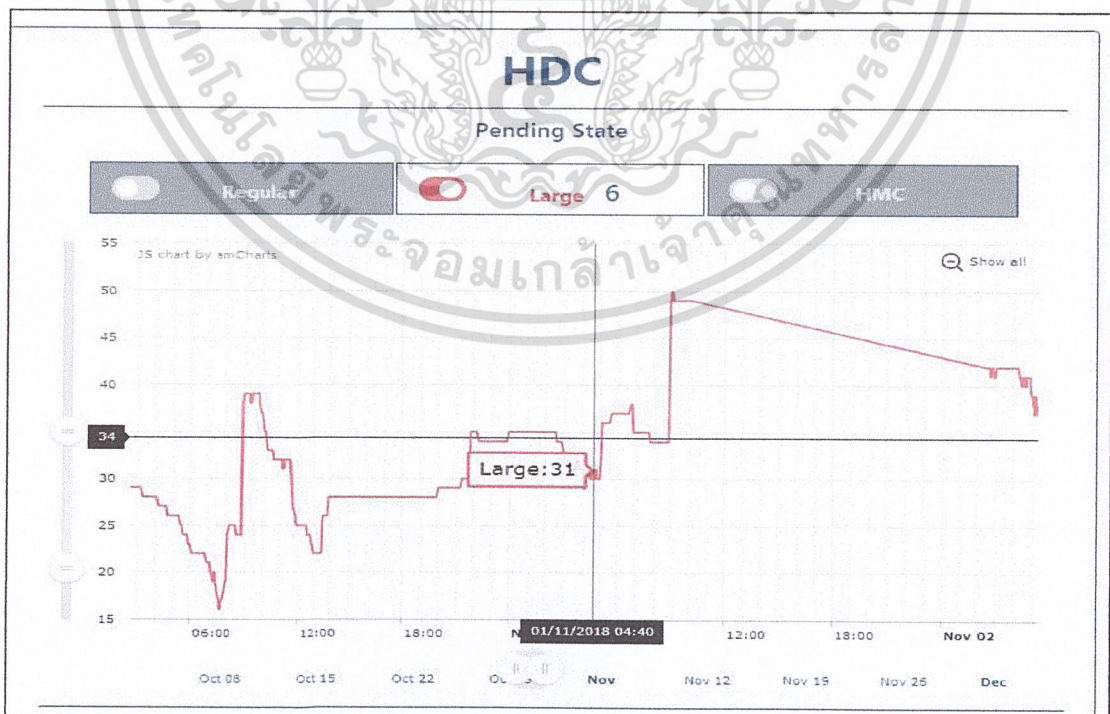
รูปที่ 4.3 หน้าการใช้งานหลักของเว็บแอปพลิเคชัน

จากรูปที่ 4.3 เป็นหน้าการใช้งานหลักของเว็บแอปพลิเคชัน ซึ่งมีการระบุปริมาณ Requested Data ที่อยู่บนระบบ ETH ขณะนั้น โดยจะมีการแสดงปริมาณ Requested Data แยกตาม Web Server Site , State และ Queue ที่ Requested Data กำลังรอหรือทำงานอยู่ และเอกสารแสดงประวัติปริมาณ Requested Data ในรูปแบบแผนภูมิเส้นนั้น ไม่นิยามให้เข้าไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 หน้าการใช้งานหลักของเว็บแอปพลิเคชัน(ต่อ)

จากรูปที่ 4.4 แผนภูมิเส้นซึ่งแสดงประวัติปริมาณ Requested Data ผู้ใช้งานสามารถย่อหรือขยาย เพื่อตรวจสอบปริมาณในช่วงเวลาที่ต้องการซึ่งสามารถทำได้โดยนำลูกศรเมาส์ไปชี้ในช่วงเวลาที่ต้องการ แผนภูมิเส้นจะแสดงตัวเลขที่บ่งบอกถึงปริมาณ Requested Data ในช่วงเวลาดังกล่าว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.5 การเปิด/ปิดการดึงข้อมูลและการแสดงแผนภูมิเส้น
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.5 เป็นการ เปิด/ปิด การดึงข้อมูลบนฝั่ง Web Service และการแสดงแผนภูมิเส้นของแต่ละ Web Server Site , State และ Queue บนส่วนของผู้ใช้งาน เพื่อให้สามารถปรับใช้ให้เข้ากับวัตถุประสงค์ของการวิเคราะห์ตามสถานการณ์

The screenshot shows the 'HDC Pending State' interface. At the top, there are three toggle buttons: 'Regular' (off), 'Large' (on, with a count of 6), and 'HMC' (off). Below these are 'Copy', 'CSV', and 'Excel' buttons, and a search field. The main part of the interface is a table with the following columns: Job Submitted Time, Pending Time, Company Name, Username, Ric Day, Allocated MB, Allocated Vcore, and More. The table contains six rows of data, all with a pending time of 0 days. The background features a large watermark of the Thai national emblem.

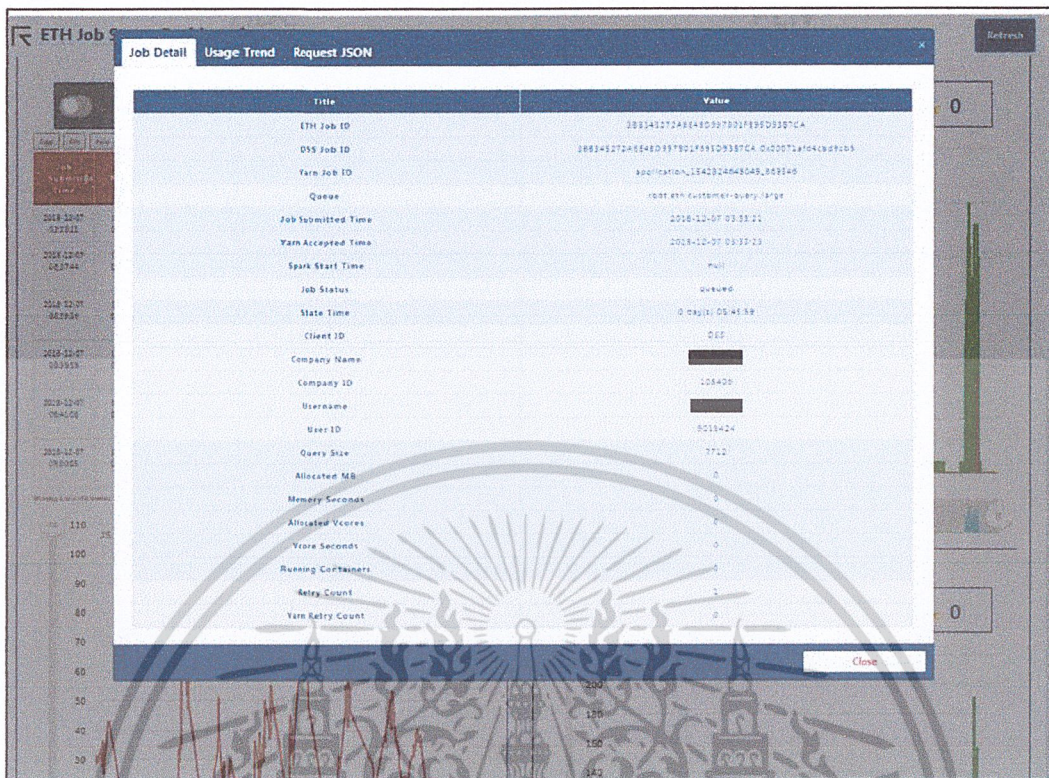
Job Submitted Time	Pending Time	Company Name	Username	Ric Day	Allocated MB	Allocated Vcore	More
2018-12-07 03:33:21	0 day(s) 05:33:38	[Redacted]	[Redacted]	7712	0	0	[More]
2018-12-07 06:37:44	0 day(s) 02:29:12	[Redacted]	[Redacted]	782228	0	0	[More]
2018-12-07 06:39:16	0 day(s) 02:27:41	[Redacted]	[Redacted]	782228	0	0	[More]
2018-12-07 06:39:59	0 day(s) 02:26:56	[Redacted]	[Redacted]	779369	0	0	[More]
2018-12-07 06:41:06	0 day(s) 02:25:52	[Redacted]	[Redacted]	780309	0	0	[More]
2018-12-07 09:00:05	0 day(s) 00:06:53	[Redacted]	[Redacted]	4876	0	0	[More]

Showing 1 to 6 of 6 entries

รูปที่ 4.6 ตัวอย่างการแสดงผลรายละเอียดหลัก Requested Data บนตาราง

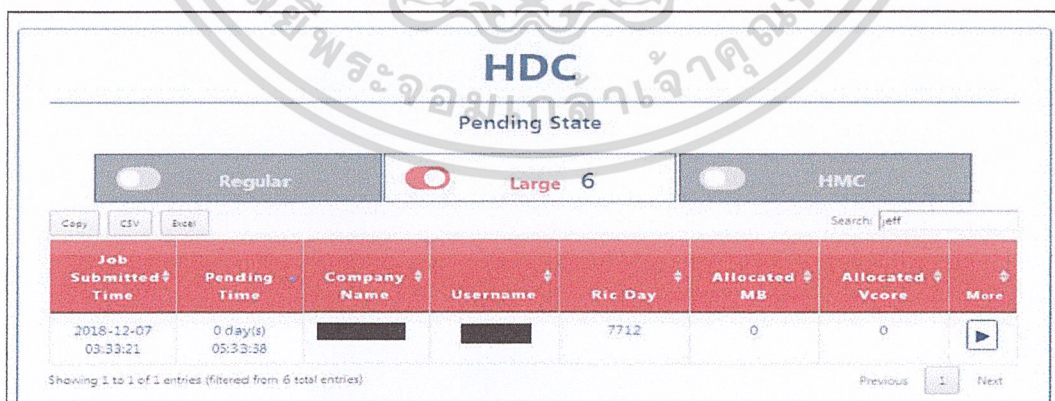
จากรูปที่ 4.6 เป็นการแสดงผลรายละเอียด Requested Data จากการกดที่ปุ่มตามชื่อ Queue ในแต่ละ State แต่ละ Web Server Site โดย Web Service จะทำการดึงข้อมูลรายละเอียดมาจากรฐานข้อมูลหลัก ซึ่งจะแสดงผลรายละเอียดหลักในรูปแบบตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ตัวอย่างการแสดงผลรายละเอียดทั้งหมดของ Requested Data บน Pop-up

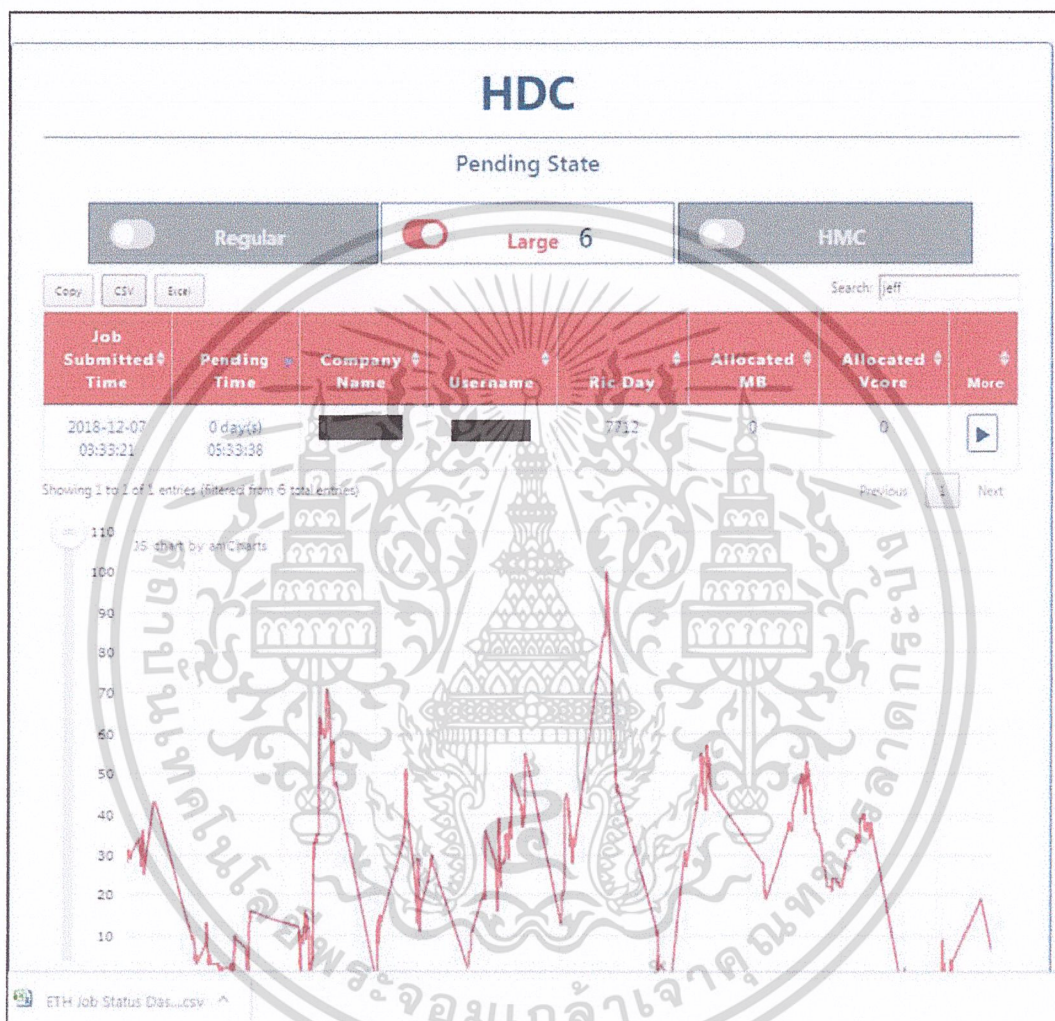
จากรูปที่ 4.7 เป็นการแสดงผลรายละเอียดทั้งหมดของ Requested Data ที่ได้มาจาก YARN Resource Manager API และ ฐานข้อมูลหลัก ซึ่งสามารถเปิดดูรายละเอียดทั้งหมดได้จากการกดปุ่มที่คอลัมน์ ชื่อ “More” บนตารางรายละเอียด Requested Data ที่สนใจแล้วเลือกแถบชื่อ “Job Detail” ของ Pop-up



รูปที่ 4.8 ตัวอย่างการค้นหาข้อความที่ตรงกับรายละเอียดของ Requested Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 เป็นการค้นหาข้อความที่ตรงกับข้อมูลภายในส่วนใดส่วนหนึ่งจากรายละเอียด Requested Data ทั้งหมด โดยสามารถค้นหาได้ผ่านช่อง Search ด้านบนของตารางรายละเอียด Requested Data



รูปที่ 4.9 ตัวอย่างการดาวน์โหลดรายละเอียดทั้งหมดของ Requested Data

จากรูปที่ 4.9 เป็นการดาวน์โหลดรายละเอียด Requested Data ทั้งหมดในรูปแบบของไฟล์ CSV หรือ Excel ซึ่งสามารถดาวน์โหลดได้โดยการกดปุ่มตามชื่อประเภทไฟล์ที่ต้องการ ในส่วนบนของตารางรายละเอียด Requested Data ซึ่งผลลัพธ์ของไฟล์เป็นไปดังรูปที่ 4.10 และ รูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

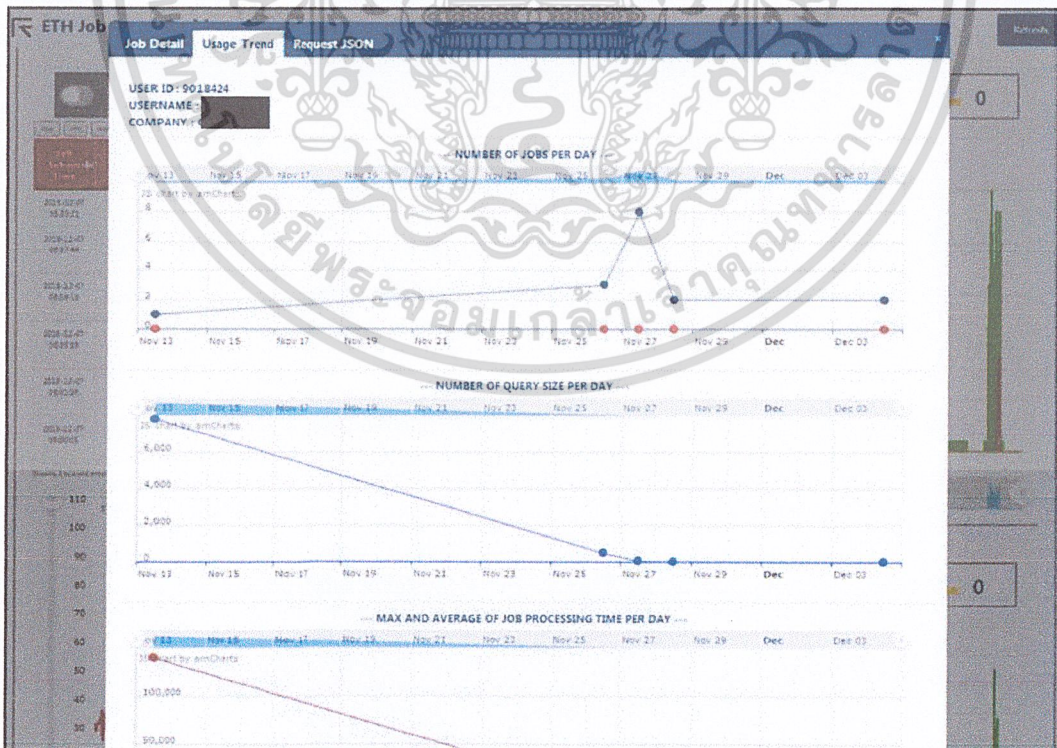
	A	B	C	D	E	F	G	H
1	Job ID	Name	Queue	Job Submitted Time	Yarn Accepted Time	Spark Sta	Job Status	Pending Time
2	3BB345272A	3BB345272A8E48C	root.eth.c	12/7/2018 3:33	12/7/2018 3:33	null	queued	0 day(s) 05:33:38
3	2BDCD0AF4C	2BDCD0AF4C0746	root.eth.c	12/7/2018 6:37	12/7/2018 6:37	null	queued	0 day(s) 02:29:12
4	F122BA5536C	F122BA5536C4452	root.eth.c	12/7/2018 6:39	12/7/2018 6:39	null	queued	0 day(s) 02:27:41
5	EEB3C61690F	EEB3C61690FB452	root.eth.c	12/7/2018 6:39	12/7/2018 6:40	null	queued	0 day(s) 02:26:56
6	4C2264DC005	4C2264DC00914D	root.eth.c	12/7/2018 6:41	12/7/2018 6:41	null	queued	0 day(s) 02:25:52
7	8E636103888	8E636103888748D	root.eth.c	12/7/2018 9:00	12/7/2018 9:00	null	queued	0 day(s) 00:06:53
8								

รูปที่ 4.10 ตัวอย่างรายละเอียด Requested Data ภายในไฟล์ CSV

I	J	K	L	M	N	O	P	Q	R	S	T	U	V				
Client ID	Company Name	Company	Username	User ID	Ric Day	Allocated	Memory S	Allocated	Vcore	Sec	Running	C	Retry	Cou	Yarn	Retry	Count
DSS		108406		9018424	7712	0	0	0	0	0	0	1	0				
DSS		107135		9014867	782228	0	0	0	0	0	0	1	0				
DSS		107135		9014867	782228	0	0	0	0	0	0	1	0				
DSS		107135		9014867	773353	0	0	0	0	0	0	1	0				
DSS		107135		9014867	780306	0	0	0	0	0	0	1	0				
DSS		65508		9015042	4875	0	0	0	0	0	0	1	0				

รูปที่ 4.11 ตัวอย่างรายละเอียด Requested Data ภายในไฟล์ CSV (ต่อ)

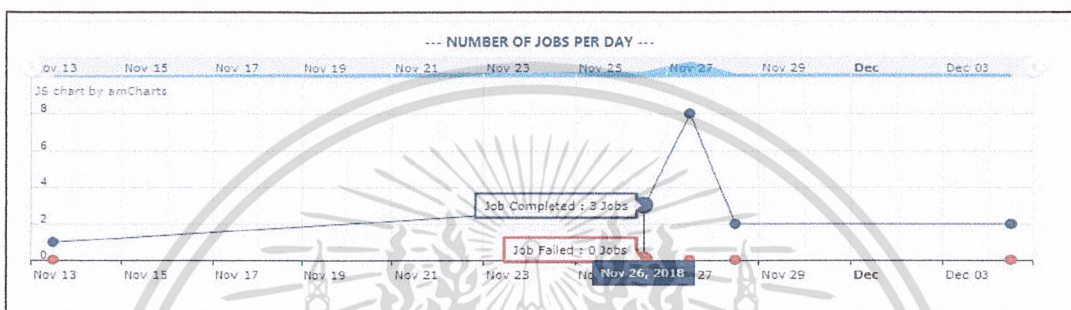
จากรูปที่ 4.10 และรูปที่ 4.11 เป็นตัวอย่างไฟล์ CSV ซึ่งภายในได้เก็บรายละเอียด Requested Data ทั้งหมดที่แสดงบน pop-up ในแถบชื่อ “Job Detail” ของ Requested Data ทุกตัวในแต่ละ Queue เพื่อใช้ในการบันทึกปัญหาที่พบและนำไปประกอบการวิเคราะห์ แก้ไข



รูปที่ 4.12 ตัวอย่างการแสดงผลการใช้งานของผู้ใช้งานในรูปแบบแผนภูมิเส้น

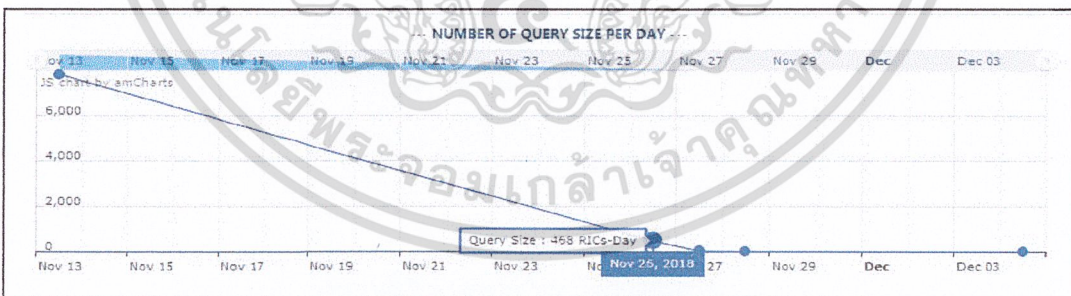
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้เฉพาะเพื่อความใช้งานเท่านั้น ไม่ควรเผยแพร่ให้วงใช้ประโยชน์ด้านการค้า
 ไม่ควรกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.12 เป็นการแสดงปริมาณการใช้งานของผู้ใช้งานในรูปแบบแผนภูมิเส้นซึ่งประกอบด้วยปริมาณ Requested Data ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด ดังรูปที่ 4.13 ขนาดข้อมูลที่ขอ (RIC DAY) ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด ดังรูปที่ 4.14 ค่าเฉลี่ยเวลาของ Requested Data ที่ใช้ในระบบ ETH (TOTAL_TIME) ทั้งหมดของผู้ใช้งานแต่ละคนในแต่ละวันระยะเวลา 30 วันล่าสุด ดังรูปที่ 4.15 ซึ่งสามารถเปิดดูในส่วนนี้ได้โดยที่แถบชื่อ “Usage Trend” ของ Pop-up



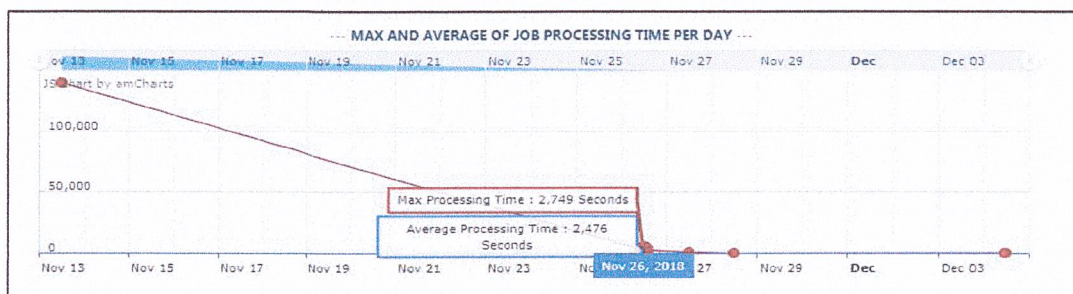
รูปที่ 4.13 ตัวอย่างแผนภูมิเส้นแสดงปริมาณ Requested Data ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด

จากรูปที่ 4.13 เป็นส่วนของการแสดงปริมาณ Requested Data ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด จะประกอบด้วย 2 เส้น คือเส้นสีน้ำเงิน ซึ่งจะบ่งบอกถึงปริมาณ Requested Data ที่ทำงานสำเร็จและเส้นสีแดง ซึ่งจะบ่งบอกถึง ปริมาณ Requested Data ที่ทำงานไม่สำเร็จ



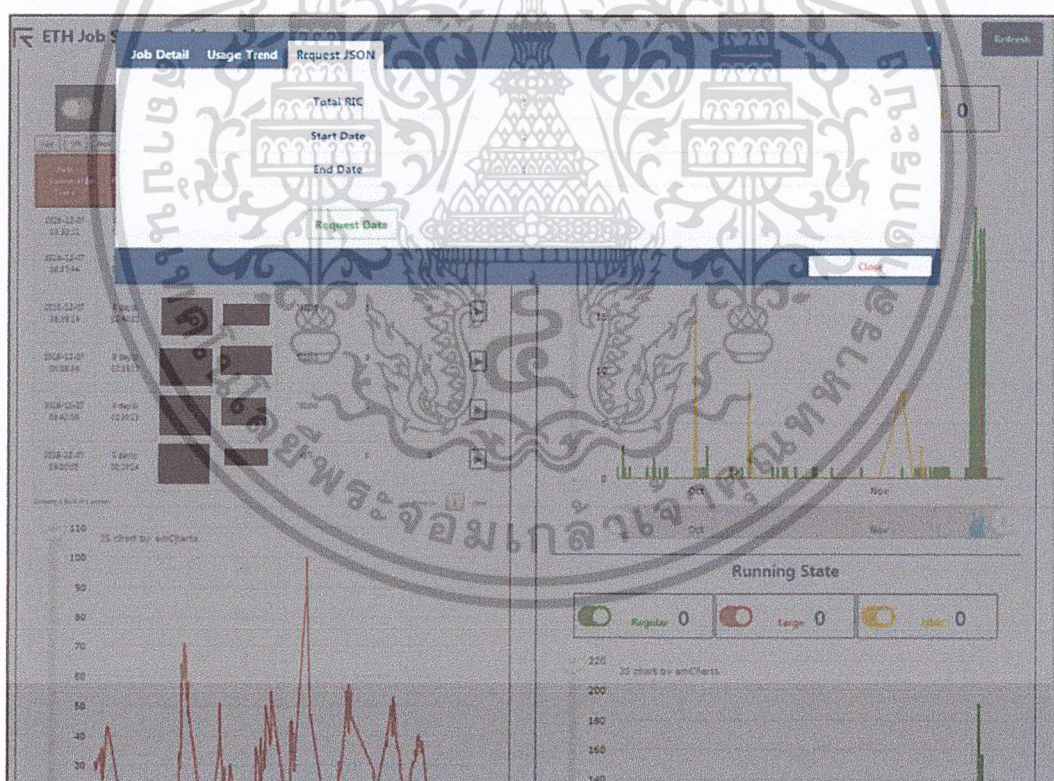
รูปที่ 4.14 ตัวอย่างแผนภูมิเส้นแสดงขนาดข้อมูลที่ขอ (RIC DAY) ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด

จากรูปที่ 4.14 เป็นส่วนของการแสดงขนาดข้อมูลที่ขอ (RIC DAY) ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด โดยในแต่ละค่าขนาดข้อมูลที่ขอ (RIC DAY) ในแต่ละวันที่แสดงบนแผนภูมิเส้น ได้จากผลรวมค่าขนาดข้อมูลที่ขอ (RIC DAY) ของทุก Requested Data ของเอกสารนี้ทุกคำ ทั้งวัน ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 ตัวอย่างแผนภูมิเส้นแสดงค่าเฉลี่ยเวลาของ Requested Data ที่ใช้ในระบบ ETH ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด

จากรูปที่ 4.15 เป็นส่วนของการแสดงค่าเฉลี่ยเวลาของ Requested Data ที่ใช้ในระบบ ETH ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน ระยะเวลา 30 วันล่าสุด จะประกอบด้วย 2 เส้น คือ เส้นสีน้ำเงิน ซึ่งจะบ่งบอกถึง ค่าเฉลี่ยเวลาของ Requested Data ที่ใช้ในระบบ ETH ทั้งหมดและเส้นสีแดง ซึ่งจะบ่งบอกถึง ค่าเวลาสูงสุดของ Requested Data ที่ใช้ในระบบ ETH จากทั้งหมด



รูปที่ 4.16 ตัวอย่างการแสดงผลปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาสุดท้ายที่ขอบน Requested Data

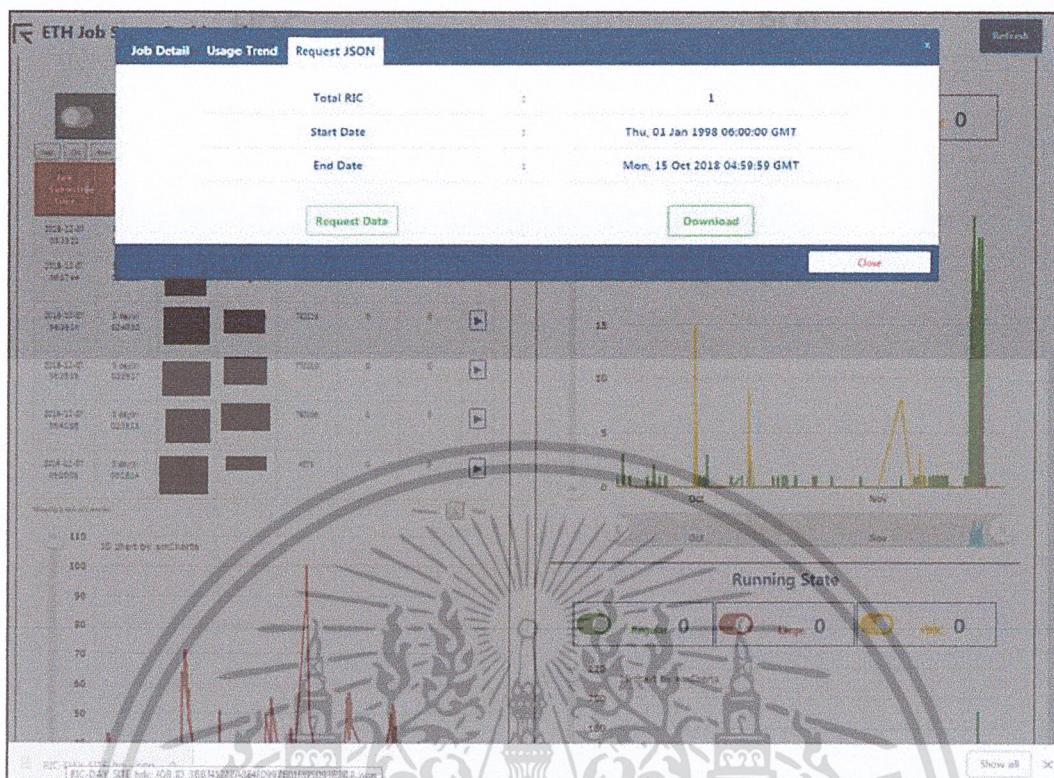
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.16 เป็นการแสดงปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาสุดท้ายที่ขอบน Requested Data ซึ่งสามารถเปิดดูในส่วนนี้ได้โดยที่แถบชื่อ “Request JSON” ของ Pop-up ซึ่งในส่วนนี้จะไม่มีการดึงข้อมูลจนกว่าผู้ใช้งานจะกดปุ่ม “Request Data” เพื่อให้ Web Service ทำการดึงข้อมูล Request JSON ซึ่งเก็บรายละเอียดการขอข้อมูลภายใน Requested Data มาจากฐานข้อมูลหลัก และเมื่อฝั่งผู้ใช้งานได้รับรายละเอียด ปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาสุดท้ายที่ขอบน Requested Data ปุ่ม Download จะแสดงขึ้นมาให้สามารถดาวน์โหลดรายละเอียดข้อมูลที่ขอภายใน Requested Data ในรูปแบบไฟล์ JSON ดังรูปที่ 4.18 โดยรายละเอียดตลาดที่ขอภายในไฟล์ JSON เป็นไปดังรูปที่ 4.19



รูปที่ 4.17 ตัวอย่างการแสดงผลระหว่างรอ Web Service ดึงข้อมูลปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาที่สุดท้ายที่ขอ Requested Data

จากรูปที่ 4.17 ด้วยเหตุที่รายละเอียดการขอข้อมูลภายใน Requested Data มีขนาดที่ไม่แน่นอน ซึ่งอาจมีขนาดใหญ่จนการดึงข้อมูลไปรบกวนการทำงานของ ฐานข้อมูลหลักและใช้เวลาในการดึงข้อมูลมาก จึงออกแบบให้สามารถดึงข้อมูลได้ก็ต่อเมื่อมีการกดปุ่ม “Request Data” เมื่อกดแล้วจะแสดงคำว่า “Loading” ขณะรอ Web Service ส่งข้อมูลมาให้



รูปที่ 4.18 ตัวอย่างการแสดงผลปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาที่สุดท้ายที่ขอ Requested Data หลังได้รับข้อมูลจาก Web Service

จากรูปที่ 4.18 เป็นการแสดงผลปริมาณข้อมูลตลาด เวลาที่เริ่มและเวลาที่สุดท้ายที่ขอ Requested Data หลังได้รับข้อมูลจาก Web Service แล้วมีการแสดงปุ่ม Download ขึ้นมาให้สามารถดาวน์โหลดรายละเอียดข้อมูลที่ขอภายใน Requested Data ในรูปแบบไฟล์ JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1  {} RIC-DAY_SITE_hdc_JOB_ID_3BB345272A8E48D997B01F895D93B7CA.json x
2      { "clientID":"DSS",
3        "displaySourceRIC":false,
4        "displayUTC":false,
5        "fileName":"Emini Intraday Volume",
6        "items":[{"
7            "domain":"6",
8            "end":"2018-10-15T04:59:59.000Z",
9            "ric":"ESc1","start":"1998-01-01T06:00:00.000Z"
10           }],
11       "jobName":"0x00071afd4cbd9cb5",
12       "maxMergeSize":"0GiB",
13       "sortBy":"ric",
14       "userID":"9018424",
15       "view":{"
16           "interval":900,
17           "messageTypes":[
18               {
19                 "fields":["Volume"],
20                 "name":"Intraday"
21               },
22               {
23                 "name":"summarised"
24               }
25           ]
26       }

```

รูปที่ 4.19 ตัวอย่างการขอข้อมูลภายใน Requested Data ภายในไฟล์ JSON

จากรูปที่ 4.19 เป็นการแสดงรายละเอียดการขอข้อมูลภายใน Requested Data ในไฟล์ JSON ซึ่งจะบ่งบอกถึงชื่อตลาดที่ขอทั้งหมด ช่วงเวลาที่ขอทั้งหมดและรูปแบบของข้อมูลที่ต้องการ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการดำเนินงานของการพัฒนาเว็บแอปพลิเคชันเพื่อใช้ในการวิเคราะห์และตรวจสอบปริมาณและรายละเอียดของ Requested Data ที่ทำงานอยู่ระบบ ETH โดยการวิเคราะห์และออกแบบ Client usage database มาใช้งานร่วมกับเว็บแอปพลิเคชัน เพื่อลดภาระจากการทำงานของฐานข้อมูลหลักและลดเวลาที่ใช้ในการดึงข้อมูล โดยผลการทดสอบการดึงข้อมูลระหว่างฐานข้อมูลหลักและ Client usage database เป็นดังตารางที่ 4.1

ตารางที่ 4.1 เปรียบเทียบเวลาที่ใช้ในการดึงข้อมูลระหว่างจากฐานข้อมูลหลักและ Client usage Database

รายการเปรียบเทียบ	ฐานข้อมูลหลัก	Client Usage Database
ปริมาณ Requested Data ทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน 30 วันล่าสุด	≈30 วินาที	≈10 วินาที
ขนาดข้อมูลทั้งหมดของผู้ใช้งานแต่ละคน ในแต่ละวัน 30 วันล่าสุด	≈30 วินาที	≈10 วินาที
ค่าเฉลี่ยเวลาของ Requested Data ทั้งหมดที่ใช้ในระบบวัน ของผู้ใช้งานแต่ละคน ในแต่ละวัน 30 วันล่าสุด	≈30 วินาที	≈10 วินาที
เวลาที่ใช้รวม	≈90วินาที	≈30 วินาที

ตารางที่ 4.1 สามารถสังเกตได้ว่าการดึงข้อมูลข้างต้นจาก Client usage database ใช้เวลาในการดึงข้อมูลน้อยกว่า ฐานข้อมูลหลักค่อนข้างมาก เนื่องจากข้อมูลอยู่บนเครื่อง Server เดียวกับเว็บแอปพลิเคชัน และข้อมูลเวลาทั้งหมดที่ Requested data ทำงานบนระบบ ETH (TOTAL_TIME) ในส่วนของฐานข้อมูลหลักไม่มีการบันทึกไว้ จึงต้องใช้เวลาในการคำนวณ แต่ในส่วนของ Client usage database มีการคำนวณและบันทึกผ่านทางชุดคำสั่ง Daily Script แล้ว จึงทำให้ฐานข้อมูลไม่เสียเวลาในการคำนวณข้อมูลส่วนนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากมีการพัฒนาเว็บแอปพลิเคชันมาช่วยในการวิเคราะห์ระบบ ETH ทำให้ผู้ดูแลระบบได้สามารถตรวจสอบและวิเคราะห์ข้อมูลรูปแบบใหม่ ซึ่งสามารถเปรียบเทียบข้อมูลกับผู้ดูแลระบบสามารถนำมาวิเคราะห์ระหว่างก่อนและหลังมีการพัฒนาเว็บแอปพลิเคชันเป็นดังตารางที่ 4.2

ตารางที่ 4.2 เปรียบเทียบข้อมูลจากผู้ดูแลระบบสามารถนำมาวิเคราะห์ระหว่างก่อนและหลังมีการพัฒนาเว็บแอปพลิเคชัน

รายการเปรียบเทียบ	ก่อนมีการพัฒนาเว็บแอปพลิเคชัน	หลังมีการพัฒนาเว็บแอปพลิเคชัน
ตรวจสอบปริมาณ Requested Data ณ ปัจจุบัน	ยุ่งยากและใช้เวลานาน	ตรวจสอบได้ทันที
ตรวจสอบประวัติปริมาณ Requested Data	ยุ่งยากและใช้เวลานาน	ตรวจสอบได้ทันที
แสดงรูปแบบของประวัติปริมาณ Requested Data เป็นแผนภูมิเส้น	ต้องใช้ซอฟต์แวร์เฉพาะ	ตรวจสอบได้ทันที
ตรวจสอบรายละเอียด Requested Data	ยุ่งยากและใช้เวลานาน	ตรวจสอบได้ทันที
ตรวจสอบรายละเอียดข้อมูลตลาดของ Requested Data	ยุ่งยากและใช้เวลานาน	ตรวจสอบได้ทันที

ตารางที่ 4.2 สามารถกล่าวได้ว่าก่อนมีการพัฒนาเว็บแอปพลิเคชันการตรวจสอบทั้งปริมาณและรายละเอียด Requested Data ที่กำลังอยู่บนระบบ ETH ณ ปัจจุบันค่อนข้างใช้เวลาและมีหลายขั้นตอนในที่จะตรวจสอบ ทำให้การแก้ปัญหาที่อาจเกิดขึ้นล่าช้าไปด้วย แต่ปัจจุบันหลังมีการพัฒนาเว็บแอปพลิเคชันขึ้นมามีการตรวจสอบข้อมูลในส่วนนี้สามารถเปิดตรวจสอบไว้ได้ตลอดเวลา ทำให้ผู้ดูแลระบบสามารถแก้ปัญหาได้ทันทีที่พบเจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้นได้มีการให้ผู้ดูแลระบบจำนวน 1 คนและนักวิเคราะห์จำนวน 2 คน ทำการประเมินผลความพึงพอใจต่อการใช้งานเว็บแอปพลิเคชันดังกล่าวที่ 4.3 โดยในช่องคะแนน 5 = พึงพอใจมากที่สุด, 4 = พึงพอใจมาก, 3 = ปานกลาง, 2 = น้อย, 1 = น้อยที่สุดหรือไม่พึงพอใจ

ตารางที่ 4.3 ประเมินผลการพัฒนาเว็บแอปพลิเคชัน

ลำดับ	รายการการประเมิน	คะแนนเฉลี่ย
1	รูปแบบการใช้งานเข้าใจง่าย	5
2	ความสะดวก ใช้อำนวยต่อการใช้งาน	4.67
3	ความสวยงามของ User interface	5
4	การจัดการกับข้อผิดพลาดที่เกิดขึ้น	4
5	ความรวดเร็วในการทำงานและแสดงผล	3.67
6	ลักษณะของเว็บแอปพลิเคชันตรงตามความต้องการของผู้ดูแลระบบ	4.3
7	ผลลัพธ์บนเว็บแอปพลิเคชันอยู่ในรูปแบบที่อ่านและเข้าใจได้ง่าย	5
8	ผลลัพธ์ในไฟล์ซึ่งดาวน์โหลดจากเว็บแอปพลิเคชันตรงตามความต้องการ	5
9	ผลลัพธ์ทั้งหมดมีความถูกต้องแม่นยำ	5
	เฉลี่ย	4.62

จากผลการประเมินการพัฒนาเว็บแอปพลิเคชันจากผู้ใช้งานระบบจำนวน 3 คนในตารางที่ 4.3 สามารถสรุปได้ว่าคุณภาพของเว็บแอปพลิเคชันอยู่ในเกณฑ์ที่ค่อนข้างดีด้วย รูปแบบการใช้งานที่เข้าใจง่าย มีประโยชน์ต่อการทำงาน ผลลัพธ์ที่แสดงบนเว็บแอปพลิเคชันตรงตามความต้องการและเข้าใจได้ง่าย แต่ด้วยข้อมูลที่มีปริมาณมาก จึงทำให้เว็บแอปพลิเคชันใช้เวลาในการประมวลผลนานในบางส่วน รวมถึงการจัดการข้อผิดพลาด เช่น ถ้าผู้ใช้ต้องการยกเลิกค่าขอข้อมูลที่ส่งไป ซึ่งไม่สามารถทำได้เนื่องจากไม่มีการพัฒนาความสามารถในส่วนดังกล่าวไว้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการพัฒนาเว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูลการเรียกใช้งานข้อมูลของลูกค้าบนระบบ ETH หรือ Web Applications for Data Analysis of Customer Requested Data สามารถสรุปผลการดำเนินงานได้ต่อไปนี้

ระบบเว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูลการเรียกใช้งานข้อมูลลูกค้าบนระบบ ETH ที่พัฒนาขึ้นสามารถทำให้ผู้ดูแลระบบ ตรวจสอบปริมาณ Requested Data ณ ปัจจุบันได้อย่างทันที ส่งผลให้ผู้ดูแลระบบสามารถทราบได้ถึงข้อผิดพลาดหรือสิ่งผิดปกติที่กำลังเกิดขึ้นและสามารถตรวจสอบรายละเอียดของ Requested Data ที่เป็นปัญหาทั้ง ชื่อผู้ใช้งาน ชื่อบริษัท รูปแบบการใช้งานของผู้ใช้ตลอดระยะเวลา 30 วันล่าสุดรวมถึงปริมาณและรายละเอียดการขอข้อมูลภายใน Requested Data (Request JSON) ได้ จากความสามารถทั้งหมดที่กล่าวมาของเว็บแอปพลิเคชันเพื่อการวิเคราะห์ข้อมูลการเรียกใช้งานข้อมูลของลูกค้าบนระบบ ETH ส่งผลให้ผู้ดูแลระบบสามารถลดกระบวนการในการตรวจสอบสถานะปริมาณและรายละเอียดของ Requested Data ที่อยู่บนระบบ ETH ณ ปัจจุบัน ทำให้สามารถตรวจสอบและแก้ปัญหาที่เกิดขึ้นได้รวดเร็วยิ่งขึ้น และการที่ระบบสามารถตรวจสอบประวัติการใช้งานของลูกค้าในรูปแบบของแผนภูมิเส้นได้ ทำให้ช่วยนักวิเคราะห์ลดขั้นตอนในการทำ Data Visualization เพื่อใช้ในการตรวจสอบพฤติกรรมขอข้อมูลของลูกค้า นอกจากนี้การสร้างฐานข้อมูล Client Usage Database มาช่วยในการเก็บรายละเอียด Requested Data บางส่วนจากฐานข้อมูลหลักเพื่อช่วยแบ่งเบาภาระฐานข้อมูลหลักและช่วยเพิ่มความเร็วในการทำงานของเว็บแอปพลิเคชัน แล้วยังทำให้ผู้ดูแลระบบมีรายละเอียด Requested Data ที่จำเป็นเก็บไว้ในเครื่องเซิร์ฟเวอร์ของตัวเอง ส่งผลให้เมื่อต้องการข้อมูลในการนำไปวิเคราะห์ ทำให้ไม่จำเป็นต้องไปรบกวนฐานข้อมูลหลักของบริษัทในการขอข้อมูลรายละเอียดส่วนนี้ ซึ่งจากการประเมินการใช้งานจากผู้ดูแลระบบได้ค่าเฉลี่ย 4.62 อยู่ในระดับดีมาก

5.2 ข้อจำกัด

เครื่องเซิร์ฟเวอร์ที่เว็บแอปพลิเคชันทำงานต้องเป็นระบบปฏิบัติการ Windows เท่านั้นและไม่สามารถดูรายละเอียด Requested Data ได้ถ้าเครื่องเซิร์ฟเวอร์ไม่มีการเชื่อมต่อ VPN ของบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

เนื่องจากในปัจจุบันผู้ดูแลระบบจะสามารถทราบถึงสิ่งผิดปกติหรือข้อผิดพลาดที่เกิดขึ้นกับระบบ ETH ก็ต่อเมื่อเข้าเว็บแอปพลิเคชันเท่านั้น จึงควรมีการพัฒนาให้เว็บแอปพลิเคชันสามารถตรวจจับปริมาณมากเกินไปของ Requested Data หรือมี Requested Data ทำงานอยู่บนระบบมากกว่าที่กำหนด ให้มีการส่งอีเมลแจ้งเตือนให้ผู้ดูแลระบบทราบทันที



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

จักรกฤษณ์ แสงแก้ว. (2550). การเขียนโปรแกรมภาษาไพธอนด้วยตนเอง. สืบค้นเมื่อ 16 กันยายน 2561, จากสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

บัญชา ปะสีละเตสัง. (2551). ออกแบบและพัฒนาเว็บด้วย DHTML(HTML/CSS/JavaScript). สืบค้นเมื่อ 16 กันยายน 2561, จากสำนักหอสมุดกลางสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

บัญชา ปะสีละเตสัง. (2556). การสร้างเว็บไซต์ด้วย HTML5 ร่วมกับ CSS3 และjQuery .สืบค้นเมื่อ 16 กันยายน 2561, จากสำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ประชา พฤกษ์ประเสริฐ. (2553). สร้างเว็บและเพิ่มลูกเล่นด้วย JavaScript DHTML. สืบค้นเมื่อ 16 กันยายน 2561, จากสำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ยุพินไทย รัตนานนท์. (2540). การประมวลผลเพิ่มข้อมูล. สืบค้นเมื่อ 16 กันยายน 2561, จากสาขา ศาสตร์คอมพิวเตอร์มหาวิทยาลัยธรรมศาสตร์.

ฤกษ์ชัย พุประทีปศิริ. (2554). โครงสร้างข้อมูลและอัลกอริทึม. สืบค้นเมื่อ 16 กันยายน 2561, จาก มหาวิทยาลัยเทคโนโลยีมหานคร.

วรรณวิภา ติตถะสิริ. (2548). คู่มือ SQL ด้วยตัวเอง. สืบค้นเมื่อ 16 กันยายน 2561, จากคณะ เทคโนโลยีสารสนเทศมหาวิทยาลัยรังสิต.

สมพงษ์ อริสริวงค์. (2554). เขียนโปรแกรมบนฐานข้อมูล MySQL. สืบค้นเมื่อ 16 กันยายน 2561, จากสำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

โอภาส เอี่ยมสิริวงค์. (2548). การวิเคราะห์และออกแบบระบบ. สืบค้นเมื่อ 16 กันยายน 2561, จาก สำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

amcharts. (2549). About amCharts. สืบค้นเมื่อ 27 กันยายน 2561, จาก <https://www.amcharts.com/about/>.

ismnida's Blog. (2552). ETL คืออะไร. สืบค้นเมื่อ 27 กันยายน 2561, จาก

<https://ismnida.wordpress.com/2009/09/26/etl/> ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/.

json.(ไม่ปรากฏวันที่พิมพ์).Introducing JSON. สืบค้นเมื่อ 27 กันยายน 2561, จาก <http://www.json.org/>.

Krissanawat Kaewsanmuang. (2560). jQuery คืออะไร?. สืบค้นเมื่อ 27 กันยายน 2561, จาก <https://medium.com/@krissanawat/jquery%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-1cfa58e4a311>.

mindphp.com. (2560). รู้จักกับ Visual Studio Code (วิชาล สตูดิโอ โค้ด) โปรแกรมฟรีจากค่าย ไมโครซอฟท์. สืบค้นเมื่อ 1 ตุลาคม 2560, จาก <https://mindphp.com/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1/microsoft/4829-visual-studio-code.html>.

msit.mut.ac.th. (2561). Cisco AnyConnect VPN. สืบค้นเมื่อ 1 ตุลาคม 2560, จาก <http://www.msit.mut.ac.th/index.php/blog/cisco-anyconnect-vpn>.

odebee.co.th. (2560). วิธีใช้ trello โปรแกรมบริหารจัดการงานของดีและฟรี. สืบค้นเมื่อ 1 ตุลาคม 2560, จาก <https://www.codebee.co.th/labs/%E0%B8%A7%E0%B8%B4%E0%B8%98%E0%B8%B5%E0%B9%83%E0%B8%8A%E0%B9%89-trello/>.

Somkiat Puisungnoen. (2559). สรุปโครงสร้างข้อมูลของ NoSQL Database. สืบค้นเมื่อ 27 กันยายน 2561, จาก <http://www.somkiat.cc/types-of-nosql-database/>.

Suphakit Annopornchai.(2560).Flask คืออะไร. สืบค้นเมื่อ 27 กันยายน 2561, จาก <https://saixiii.com/python-flask-web-application/>.

Suphakit Annopornchai.(2560). Apache คืออะไร?. สืบค้นเมื่อ 1 ตุลาคม 2560, จาก <https://saixiii.com/apache-webserver/>.

thaicreate.com. (2554). MySQL Workbench Tools. สืบค้นเมื่อ 1 ตุลาคม 2560, จาก <https://www.thaicreate.com/software/mysql-workbench.html>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มโครงการพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา

วันที่ 25 เดือน พฤษภาคม พ.ศ. 2662

ข้าพเจ้า นาย/นาง/นางสาว สันทาน ศรีสว่างเนตร รหัสประจำตัว 58050397

นาย/นาง/นางสาว รหัสประจำตัว

นาย/นาง/นางสาว รหัสประจำตัว

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา วิทยาการคอมพิวเตอร์ ภาควิชา วิทยาการคอมพิวเตอร์

ขอรับรองว่าโครงการพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา เรื่อง

ชื่อภาษาไทย เว็บไซต์แอปพลิเคชันเพื่อการบริหารจัดการข้อมูลรายจ่ายของลูกค้า

ชื่อภาษาอังกฤษ Web Application for Data Analysis of Customer Requested Data

ปีการศึกษา 2561

เป็นผลงานวิจัยที่ได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มโครงการพิเศษ/ปัญหาพิเศษ/สหกิจศึกษาฉบับสมบูรณ์แล้ว

โปรแกรมอักษรวินิจฉัย 1.9.6 % หรือโปรแกรม Turnitin %

ลงชื่อ สันทาน ศรีสว่างเนตร ลงชื่อ ลงชื่อ

(สันทาน ศรีสว่างเนตร) () ()

นักศึกษา

นักศึกษา

นักศึกษา

ข้าพเจ้า ศ. / รศ. / ผศ. / ดร. / อ. วิวัฒน์ ตั้งวงษ์เจริญ อาจารย์ที่ปรึกษาโครงการพิเศษ/ปัญหาพิเศษ/สหกิจ

ศึกษา ได้ตรวจสอบโครงการพิเศษ/ปัญหาพิเศษ/สหกิจศึกษาของนักศึกษาข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ ลงชื่อ ลงชื่อ

อาจารย์ที่ปรึกษา

อาจารย์ที่ปรึกษาร่วม

อาจารย์ที่ปรึกษาร่วม