

การนำเสนอข้อมูลเชิงวิเคราะห์

Data Visualization



นางสาวโสณิยา พัทธา รหัส 51362190

เลขที่การรับสมัคร
..... 2.5.ค.อ. 2556
เลขทะเบียน
..... 16290256
เลขเรียกหนังสือ
..... ผร.
มหาวิทยาลัยนเรศวร
..... ๓๓๖๓

2554

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554



## ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ                      การนำเสนอข้อมูลเชิงวิเคราะห์  
ผู้ดำเนินโครงการ                    นางสาวโสณิยา พัวทา รหัส 51362190  
อาจารย์ที่ปรึกษา                      ดร.วรลักษณ์ คงเด่นฟ้า  
สาขาวิชา                                วิศวกรรมคอมพิวเตอร์  
ภาควิชา                                    วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
ปีการศึกษา                                2554

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครพนม อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

.....ประธานกรรมการ  
(อาจารย์รัฐภูมิ วรรณสาสน์)

.....กรรมการ  
(อาจารย์ภาณุพงศ์ สอนคม)

.....กรรมการ  
(อาจารย์เศรษฐา ตั้งคำวานิช)

.....กรรมการ  
(ดร.วรลักษณ์ คงเด่นฟ้า)

ชื่อหัวข้อโครงการ      การนำเสนอข้อมูลเชิงวิเคราะห์  
ผู้ดำเนินโครงการ      นางสาว โสนิยา พัวทา รหัส 51362190  
ที่ปรึกษาโครงการ      ดร.วรศักดิ์      คงเด่นฟ้า  
สาขาวิชา      วิศวกรรมคอมพิวเตอร์  
ภาควิชา      วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
ปีการศึกษา      2554

---

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ต้องการนำเสนอข้อมูลเชิงวิเคราะห์ กรณีศึกษาจำนวนประชากร จังหวัดแม่ฮ่องสอนและงบประมาณการเงินขององค์การบริหารส่วนตำบลแม่नाเตง อำเภอป่าเย็บ จังหวัดแม่ฮ่องสอน ตั้งแต่ปีงบประมาณ 2552-2554 โดยรูปแบบการนำเสนอที่จะทำการดึงรูปแบบการนำเสนอจาก Google Visualization API มาใช้กับข้อมูลดังกล่าวซึ่งถูกเก็บไว้ในฐานข้อมูลมานำเสนอ

จากผลการทดลองสามารถนำรูปแบบการนำเสนอจาก Google Visualization API มาใช้กับข้อมูลที่ถูกเก็บไว้ในฐานข้อมูลมานำเสนอได้

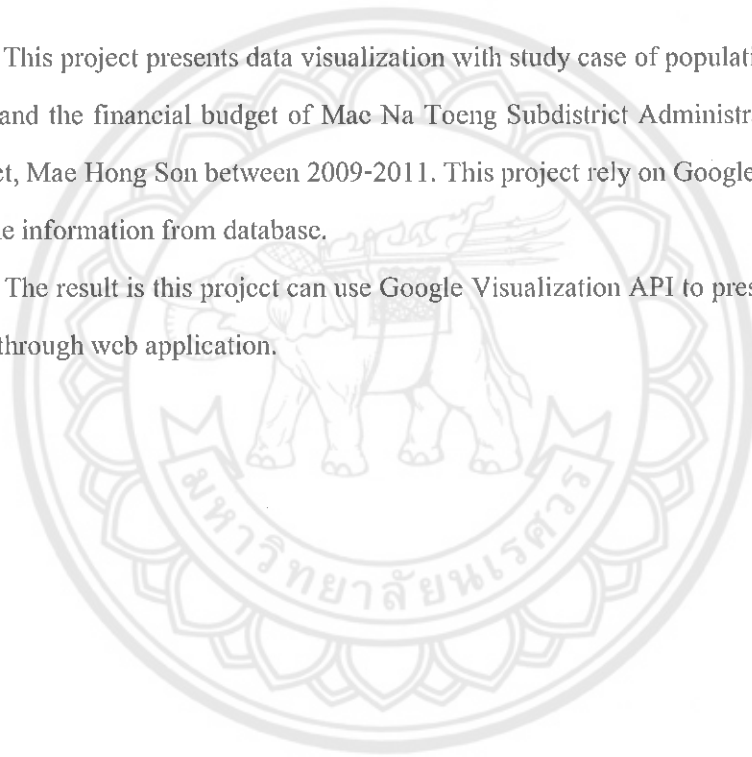
**Project title** Data Visualization  
**Name** Miss Soniya Phuata ID. 51362190  
**Project advisor** Dr. Woralak Kongdanfha  
**Major** Computer Engineering  
**Department** Electrical and Computer Engineering  
**Academic year** 2011

---

### **Abstract**

This project presents data visualization with study case of population of Mae Hong Son province and the financial budget of Mae Na Toeng Subdistrict Administration Organization in Pai district, Mae Hong Son between 2009-2011. This project rely on Google Visualization API to present the information from database.

The result is this project can use Google Visualization API to present information from database through web application.





## กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากความช่วยเหลือจากหลายๆฝ่ายด้วยกัน ผู้จัดทำโครงการจึงขอถือโอกาสนี้ ขอกราบขอบพระคุณ ดร.วรลักษณ์ คงเด่นฟ้า ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการและให้ความกรุณาให้คำปรึกษา คำแนะนำ และให้ความรู้ในการทำโครงการ ผู้ดำเนินโครงการขอขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านตลอดไป

ขอขอบพระคุณคณะกรรมการทุกท่าน คือ อาจารย์รัฐภูมิ วรรณสาสน์ อาจารย์กาญจนาพงศ์ สอนคม และอาจารย์เศรษฐา ตั้งคำวานิช ซึ่งให้คำแนะนำ ชี้แนะแนวทาง และข้อคิดเห็นต่างๆที่เป็นประโยชน์ในโครงการนี้ ทำให้โครงการออกมาสมบูรณ์แบบยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆตลอดระยะเวลา 4 ปี ซึ่งเป็นความรู้ที่สามารถนำไปใช้ในการทำโครงการนี้และยังสามารถนำไปใช้ในการประกอบอาชีพในอนาคต

สุดท้ายนี้เหนือสิ่งอื่นใด ผู้จัดทำโครงการขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ผู้มอบความรัก ความเมตตากรุณา และเป็นกำลังใจเสมอมา เป็นที่ปรึกษาในทุกๆเรื่อง ไม่ว่าจะเป็นด้านความรู้ ด้านการทำงาน ด้านการใช้ชีวิต อีกทั้งยังสนับสนุนทางการเงิน ความสำเร็จในครั้งนี้จะไม่เกิดขึ้นเลยถ้าหากขาดความรัก ความห่วงใย และกำลังใจเหล่านี้ ผู้จัดทำโครงการจึงขอกราบขอบพระคุณคุณพ่อ คุณแม่ เป็นอย่างสูง และขอขอบคุณทุกๆ คนในครอบครัวของผู้จัดทำที่ไม่ได้กล่าวมา ณ ที่นี้ด้วย

นางสาวโสณิยา พัวทา

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง .....	ช
สารบัญรูป .....	ฅ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตของโครงการ .....	2
1.4 แผนการดำเนินงาน.....	3
1.5 ผลที่คาดว่าจะได้รับ.....	4
1.6 งบประมาณ .....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ฐานข้อมูล.....	5
2.1.1 รูปแบบของระบบฐานข้อมูล .....	5
2.1.2 อีอาร์โมเดล.....	6
2.2 การติดต่อกับฐานข้อมูลด้วยวิธี LINQ to SQL.....	6
2.2.1 ชุดเครื่องมือ LINQ .....	7
2.2.2 การทำ Mapping ของ LINQ To SQL .....	8
2.3 Pivot.....	9
2.4 เว็บแอปพลิเคชัน .....	11
2.5 เจสัน.....	12
2.5.1 ไวยากรณ์เจสัน .....	12
2.5.2 ชนิดของข้อมูลภายในเจสัน .....	13

## สารบัญ(ต่อ)

	หน้า
2.5.3 เปรียบเทียบเจสันกับเอ็กซ์เอ็มแอล .....	13
2.5.4 ประโยชน์ของเจสัน .....	14
2.6 เอแจ็กซ์.....	14
2.6.1 ประวัติความเป็นมา.....	14
2.6.2 ที่มาของปัญหา.....	15
2.6.3 โครงสร้างของเอแจ็กซ์ .....	16
2.6.4 การทำงานของเอแจ็กซ์.....	17
2.6.5 ข้อดีของเอแจ็กซ์.....	19
2.7 จาวาสคริปต์ .....	19
2.8 เจคิววี.....	20
2.9 Application Programming Interface .....	21
2.10 Google Visualization API.....	21
2.10.1 รูปแบบของบริการ Google Charts .....	22
<b>บทที่ 3</b> วิธีการดำเนินโครงการ .....	43
3.1 หลักการทำงาน.....	43
3.2 การออกแบบและจัดทำฐานข้อมูล.....	45
3.3 การดึงข้อมูลจากฐานข้อมูล.....	48
3.3.1 ข้อมูลประชากรจังหวัดแม่ฮ่องสอน.....	48
3.3.1.1 การนำเสนอจำนวนประชากรของตำบลในแต่ละอำเภอ .....	48
3.3.1.2 การกำหนดขอบเขตในการนำเสนอข้อมูล.....	49
3.3.1.3 ขั้นตอนการแบ่งประเภทของข้อมูล.....	51
3.3.2 การสรุปข้อมูลการเงินขององค์การบริหารส่วนตำบลแม่नादेङ.....	52
3.3.2.1 การสรุปข้อมูลเกี่ยวกับรายรับ.....	52
3.3.2.2 การสรุปข้อมูลเกี่ยวกับรายจ่าย .....	53
3.4 การนำเสนอข้อมูล.....	54

## สารบัญ(ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง.....	56
4.1 Home page.....	56
4.2 Population page.....	57
4.3 Finance page.....	70
4.3.1 รายได้และรายจ่ายขององค์การบริหารส่วนตำบลแม่नाเติง.....	70
4.3.2 สรุปรายการรายได้.....	74
4.3.3 สรุปรายจ่าย.....	76
4.3.4 พื้นที่บริการ.....	79
4.4 Contact me page.....	81
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	82
5.1 สรุปผลการดำเนินงาน.....	82
5.2 ปัญหาที่พบ.....	83
5.3 แนวทางแก้ไขและข้อเสนอแนะ.....	83
เอกสารอ้างอิง.....	85
ภาคผนวก.....	86
ประวัติผู้จัดทำโครงการ.....	102

## สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินการ .....	1
2.1 ตัวอย่างข้อมูลดิบที่ทำการจัดเก็บ .....	10
2.2 การ Pivot ข้อมูล.....	11
2.3 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Area Chart.....	24
2.4 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Bar Chart.....	25
2.5 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Bubble Chart .....	26
2.6 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Candlestick Chart.....	27
2.7 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Column Chart.....	28
2.8 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Combo Chart.....	29
2.9 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Gauge.....	30
2.10 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Regions .....	31
2.11 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Markers.....	32
2.12 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Map Chart .....	33
2.13 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Line Chart .....	34
2.14 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Org Chart.....	35
2.15 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Pie Chart.....	36
2.16 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Scatter Chart.....	37
2.17 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Combo Chart.....	38
2.18 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Table.....	39
2.19 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Tree Map.....	40
2.20 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Motion Chart.....	41
2.21 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Annotated Time Line .....	42
5.1 เปรียบเทียบรูปแบบการนำเสนอของ Google Visualization API กับ Many Eyes .....	82

## สารบัญรูป

รูปที่	หน้า
2.1 การแปลงโครงสร้างจากฐานข้อมูลให้มาอยู่ในรูปของอ็อบเจกต์.....	6
2.2 สถาปัตยกรรม LINQ.....	7
2.3 การทำ Mapping ของ LINQ To SQL.....	8
2.4 ตัวอย่างไวยากรณ์เจสัน.....	12
2.5 ตัวอย่างข้อมูลประกอบไปด้วยคีย์และค่าข้อมูล.....	13
2.6 เปรียบเทียบการทำงานแบบเดิมกับเอแจ็กซ์.....	16
2.7 สถาปัตยกรรมของเอแจ็กซ์.....	17
2.8 การทำงานแบบ Asynchronous และการอัปเดตหน้าเว็บแบบบางส่วน.....	18
2.9 เว็บไซต์แนะนำการใช้ Google Chart.....	22
2.10 แสดงตัวอย่างรูปแบบการนำเสนอของ Google Chart.....	23
2.11 แสดงรูปแบบการนำเสนอแบบ Area Chart.....	24
2.12 แสดงรูปแบบการนำเสนอแบบ Bar Chart.....	25
2.13 แสดงรูปแบบการนำเสนอแบบ Bubble Chart.....	26
2.14 แสดงรูปแบบการนำเสนอแบบ Candlestick Chart.....	27
2.15 แสดงรูปแบบการนำเสนอแบบ Column Chart.....	28
2.16 แสดงรูปแบบการนำเสนอแบบ Combo Chart.....	29
2.17 แสดงรูปแบบการนำเสนอแบบ Gauge.....	30
2.18 แสดงรูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Regions.....	31
2.19 แสดงรูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Markers.....	32
2.20 แสดงรูปแบบการนำเสนอแบบ Map Chart.....	33
2.21 แสดงรูปแบบการนำเสนอแบบ Line Chart.....	34
2.22 แสดงรูปแบบการนำเสนอแบบ Org Chart.....	35
2.23 แสดงรูปแบบการนำเสนอแบบ Pie Chart.....	36
2.24 แสดงรูปแบบการนำเสนอแบบ Scatter Chart.....	37
2.25 แสดงรูปแบบการนำเสนอแบบ Stepped Area Chart.....	38
2.26 แสดงรูปแบบการนำเสนอแบบ Table.....	39
2.27 แสดงรูปแบบการนำเสนอแบบ Tree Map.....	40
2.28 แสดงรูปแบบการนำเสนอแบบ Motion Chart.....	41

## สารบัญรูป(ต่อ)

รูปที่	หน้า
2.29 แสดงรูปแบบการนำเสนอแบบ Annotated Time Line .....	42
3.1 หลักการทำงานของระบบการนำเสนอข้อมูลเชิงวิเคราะห์.....	43
3.2 แสดงความสามารถของระบบในรูปแบบ Use case diagram .....	44
3.3 แสดงการออกแบบฐานข้อมูลแบบ Conceptual diagram .....	46
3.4 ขั้นตอนทำงานการนำเสนอข้อมูลของประชากรในตำบลของแต่ละอำเภอ.....	48
3.5 ขั้นตอนการทำงานของการสรุปข้อมูลเพื่อกำหนดขอบเขต .....	50
3.6 ขั้นตอนการทำงานของการแบ่งประเภทของข้อมูล.....	51
3.7 หลักการทำงานการสรุปเกี่ยวกับข้อมูลรายรับรายจ่าย.....	53
3.8 แสดงรูปแบบการจัดเก็บข้อมูลของ Google Visualization API .....	54
3.9 แสดงระบบการติดต่อการใช้รูปแบบการนำเสนอของ Google Visualization API .....	55
4.1 แสดงหน้าเว็บแอปพลิเคชันหน้า Home .....	56
4.2 แสดงหน้าเว็บแอปพลิเคชันของการนำเสนอข้อมูลเกี่ยวกับประชากร.....	57
4.3 แสดงรายการฟังก์ชันการแสดงผลข้อมูลของประชากรของตำบลในแต่ละอำเภอ.....	57
4.4 แสดงการฟังก์ชันการกำหนดขอบเขตของจำนวนประชากรในแต่ละตำบล .....	58
4.5 แสดงข้อมูลประชากร โดยไม่ถูกกำหนดขอบเขตของจำนวนประชากร.....	59
4.6 แสดงข้อมูลประชากรที่ถูกกำหนดขอบเขตล่างของจำนวนประชากร .....	60
4.7 แสดงข้อมูลประชากรที่ถูกกำหนดขอบเขตบนของจำนวนประชากร .....	61
4.8 แสดงข้อมูลประชากรที่ถูกกำหนดขอบเขตล่างและขอบเขตบน.....	62
4.9 แสดงฟังก์ชันการเลือกประเภทของข้อมูล .....	63
4.11 แสดงความหนาแน่นของประชากรของแต่ละอำเภอในจังหวัดแม่ฮ่องสอน .....	64
4.12 แสดงแผนที่เมื่อทำการเลือกรายชื่ออำเภอ.....	65
4.13 แสดงการเลือกตำแหน่งบนแผนที่.....	66
4.15 แสดงเครื่องมือแก้ไขรูปแบบการนำเสนอหน้าเริ่มต้น.....	67
4.16 แสดงเครื่องมือแก้ไขแผนภูมิหน้าแผนภูมิ .....	67
4.17 แสดงเครื่องมือแก้ไขแผนภูมิหน้ากำหนดค่า.....	68
4.18 แสดงแผนภูมิเมื่อทำการปรับเปลี่ยนรูปแบบ.....	69
4.19 แสดงการนำเสนอข้อมูลรายได้และรายจ่ายของ อบต.แม่มาเต็ง .....	70
4.20 แสดงรูปแบบกราฟแบบ Area Chart.....	71

## สารบัญรูป(ต่อ)

รูปที่	หน้า
4.21 แสดงรายจ่ายการใช้งบประมาณตามแผนงาน.....	71
4.22 แสดงประเภทงานการใช้งบประมาณของแผนงานการศึกษา.....	72
4.23 แสดงรายการการใช้งบประมาณประเภทงานบริหารการศึกษาทั่วไป.....	72
4.24 แสดงประเภทรายได้.....	73
4.25 แสดงหมวดหมู่รายได้ของรายได้ประเภทรายได้ที่มีใช้ภายในอาคาร.....	73
4.26 แสดงรายการรายได้ของรายได้หมวดหมู่ ค่าธรรมเนียม ค่าปรับและใบอนุญาต.....	74
4.27 แสดงการนำเสนอข้อมูลรายการรายได้ด้วย Motion Chart.....	75
4.28 แสดงการนำเสนอข้อมูลรายการรายได้แบบตาราง.....	75
4.29 แสดงการนำเสนอข้อมูลรายการรายจ่ายแบบตาราง.....	76
4.30 แสดงการนำเสนอข้อมูลรายจ่ายค่าตอบแทนให้สอยและวัสดุแบบตาราง.....	77
4.31 แสดงการนำเสนอข้อมูลรายจ่ายค่าตอบแทนให้สอยและวัสดุด้วย Pie Chart.....	77
4.32 แสดงการนำเสนอข้อมูลรายการรายจ่ายด้วย Motion Chart.....	78
4.33 แสดงการนำเสนอข้อมูลรายการรายจ่ายด้วย Column Chart.....	78
4.34 แสดงการนำเสนอข้อมูลพื้นที่บริการของ อบต.แม่नादेिङ.....	79
4.35 แสดงการนำเสนอข้อมูลพื้นที่ของตำบลแม่नादेिङ.....	80
4.36 แสดงการนำเสนอข้อมูลพื้นที่ของตำบลแม่नादेिङด้วย Column Chart.....	80
4.37 แสดงการนำเสนอข้อมูลในหน้า Contact me.....	81



# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของโครงการ

ปัจจุบันองค์กรและสถาบันต่างๆ มีการจัดเก็บข้อมูลที่เกี่ยวข้องกับองค์กรและสถาบันของตนเองไว้ ด้วยเทคโนโลยีด้านการเก็บข้อมูลสมัยใหม่ที่ถูกพัฒนาอย่างก้าวหน้า ทำให้สามารถทำการจัดเก็บข้อมูลได้สะดวกและสามารถจัดเก็บข้อมูลได้ในปริมาณมาก ดังนั้นการนำเสนอข้อมูลจากข้อมูลที่มีจำนวนมากหรือมีลักษณะของข้อมูลที่มีความซับซ้อนนั้น จะส่งผลให้การนำเสนอข้อมูลมีประสิทธิภาพมีดีเท่าที่ควร การนำเสนอที่ดีจึงควรผ่านการกรองข้อมูลและการสรุปข้อมูลที่ต้องการให้ตรงตามจุดประสงค์ของการนำเสนอ

การสรุปข้อมูลจากข้อมูลจริงที่ถูกจัดเก็บไว้ในฐานข้อมูล จะช่วยให้ผู้ใช้หรือบุคคลกลุ่มเป้าหมายในการนำเสนอ สามารถได้รับข้อมูลในขอบเขตที่ผู้ใช้หรือบุคคลกลุ่มเป้าหมายต้องการ เนื้อหามีความกระชับ เข้าใจได้ง่าย และสามารถมองภาพรวมได้ การสรุปข้อมูลก่อนการนำเสนอ จึงมีความสำคัญและส่งผลกระทบต่อประสิทธิภาพของการนำเสนอข้อมูลเป็นอย่างมาก เพราะสิ่งสำคัญที่สุดของการนำเสนอข้อมูลคือ ข้อมูลที่ต้องถูกสรุปและคัดกรองให้ตรงตามจุดประสงค์และความต้องการของผู้ใช้หรือบุคคลกลุ่มเป้าหมาย

การนำเสนอข้อมูล นอกเหนือจากข้อมูลสำหรับนำเสนอแล้ว รูปแบบการนำเสนอก็เป็นอีกปัจจัยหนึ่งที่สร้างความน่าสนใจและช่วยให้ผู้ใช้หรือบุคคลกลุ่มเป้าหมายสามารถเข้าใจและวิเคราะห์ข้อมูลจากการนำเสนอได้ดียิ่งขึ้น ซึ่งถ้าเราพิจารณาถึงตัวบุคคลและให้ความสำคัญกับบุคคลที่ต้องการนำเสนอข้อมูลแล้ว จะเห็นว่าแต่ละบุคคลมีความชอบและความถนัดในการมองภาพและวิเคราะห์ข้อมูลในมุมมองที่ไม่เหมือนกัน ดังนั้นเพื่อให้การนำเสนอข้อมูลมีประสิทธิภาพยิ่งขึ้น เราจึงควรคำนึงถึงรูปแบบในการนำเสนอให้เหมาะสมกับจุดประสงค์ รูปแบบข้อมูล และกลุ่มบุคคลเป้าหมายที่ต้องการนำเสนอ ผู้จัดทำจึงเห็นว่า การนำเสนอในรูปแบบที่ผู้ใช้สามารถเลือกรูปแบบการนำเสนอได้ตามต้องการ เป็นอีกแนวทางหนึ่งที่จะช่วยเพิ่มประสิทธิภาพในการนำเสนอข้อมูลให้บรรลุตามจุดประสงค์ที่วางไว้ มีความน่าสนใจและเกิดช่วยให้กลุ่มบุคคลเป้าหมายสามารถวิเคราะห์ข้อมูลจากการนำเสนอข้อมูลและเกิดประโยชน์สูงสุดในการนำเสนอได้

การนำเสนอข้อมูลในปัจจุบัน มีเทคโนโลยีที่ช่วยในการออกแบบและจัดทำกรนำเสนอมากมาย และหนึ่งในเทคโนโลยีที่น่าสนใจ คือ Google Visualization API ซึ่งเป็นรูปแบบการนำเสนอข้อมูลที่ Google ผลิตขึ้นและอนุญาตให้สามารถเรียกรูปแบบการนำเสนอมาใช้กับข้อมูลของเราโดยไม่คิดค่าบริการและมีรูปแบบการนำเสนอที่หลากหลาย น่าสนใจ เหมาะกับยุคสมัยปัจจุบัน ที่เทคโนโลยีและแอปพลิเคชันต่างๆ มีการพัฒนาและก้าวหน้าอย่างรวดเร็ว

## 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อนำเสนอข้อมูลจากฐานข้อมูลในรูปแบบที่หลากหลาย
- 1.2.2 เพื่อนำเสนอข้อมูลโดยใช้รูปแบบการนำเสนอข้อมูลจาก Google Visualization API

## 1.3 ขอบเขตของโครงการ

โครงการนี้เป็นการศึกษาเกี่ยวกับการดัดรูปแบบการนำเสนอข้อมูลจาก Google Visualization API มานำเสนอข้อมูลที่ได้จากข้อมูลที่ถูกสรุปแสดงบนเว็บแอปพลิเคชัน โดยระบบมีความสามารถดังนี้

- 1.3.1 ระบบสามารถดึงข้อมูลจากฐานข้อมูลขึ้นมานำเสนอได้
- 1.3.3 ระบบสามารถดัดรูปแบบการนำเสนอจาก Google Visualization API มาแสดงกับข้อมูลที่ต้องการนำเสนอได้
- 1.3.4 ระบบสามารถให้ผู้ใช้เลือกรูปแบบการนำเสนอข้อมูล ตามรูปแบบที่ออกแบบไว้ได้



## 1.4 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	ปีการศึกษา 2554											เม.ย	พ.ค.
	มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.			
1. ศึกษาข้อมูลและเลือกหัวข้อโครงการ													
2. ออกแบบ และกำหนดขอบเขตของโครงการ													
3. ศึกษาข้อมูลที่เกี่ยวข้อง													
4. ออกแบบ จัดทำฐานข้อมูล													
5. พัฒนาโปรแกรมในส่วนการนำเสนอ													
6. พัฒนาโปรแกรมในส่วนของการดึงข้อมูลจากฐานข้อมูล													
7. พัฒนาโปรแกรมการนำเสนอข้อมูลทั้งระบบ													
8. ทดสอบและแก้ไขโปรแกรม													
8. สรุปผลการดำเนินงาน และจัดทำรูปเล่ม													

## 1.5 ผลที่คาดว่าจะได้รับ

1.5.1 ระบบสามารถนำเสนอข้อมูลจากฐานข้อมูลได้

1.5.2 ได้ระบบการนำเสนอข้อมูลจากการดึงรูปแบบการนำเสนอข้อมูลจาก Google Visualization API มาแสดงข้อมูลจากการสรุป

1.5.3 สามารถเลือกรูปแบบการนำเสนอข้อมูลได้ ตามรูปแบบที่ออกแบบไว้

## 1.6 งบประมาณ

ค่าถ่ายเอกสารและจัดทำรูปเล่มฉบับสมบูรณ์	เป็นเงิน	700	บาท
ค่าหนังสือกับวัสดุสำนักงาน	เป็นเงิน	300	บาท
	รวมเป็นเงินทั้งสิ้น	1000	บาท

หมายเหตุ ขออนุญาตด้วยเจตีย์ทุกรายการ



## บทที่ 2

### หลักการและทฤษฎีที่เกี่ยวข้อง

การจัดทำโครงการนำเสนอข้อมูลเชิงวิเคราะห์นี้ เป็นโครงการที่เกี่ยวข้องกับการสรุปข้อมูลและการนำเสนอข้อมูลกรณีศึกษาข้อมูลประชากรจังหวัดแม่ฮ่องสอนและงบประมาณรายรับรายจ่ายขององค์การบริหารส่วนตำบลแม่มาเต็ง อำเภอป่าเย็บ จังหวัดแม่ฮ่องสอน บนเว็บไซต์แอปพลิเคชัน ดังนั้นจึงจำเป็นต้องศึกษาข้อมูลและทฤษฎีการปฏิบัติที่เกี่ยวข้องกับเรื่องที่ได้กล่าวมาข้างต้น เพื่อให้การจัดทำโครงการและการทำงานมีความถูกต้อง เหมาะสม และมีประสิทธิภาพสูงสุด ซึ่งทฤษฎีที่เกี่ยวข้องนั้นสามารถอธิบายได้ดังนี้

#### 2.1 ฐานข้อมูล (Database)

ฐานข้อมูล หมายถึง กลุ่มข้อมูลที่มีความสัมพันธ์กันและถูกนำมารวมกันโดยมีโครงสร้างเดียวกัน ถูกควบคุม ดูแล และจัดการ โดยซอฟต์แวร์ระบบจัดการข้อมูล (DBMS) เพื่อตอบสนองความต้องการสารสนเทศขององค์กรเพื่อการใช้งานร่วมกันของผู้ใช้

##### 2.1.1 รูปแบบของระบบฐานข้อมูล

รูปแบบของระบบฐานข้อมูล มีอยู่ด้วยกัน 3 ประเภท คือ

- ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นการเก็บข้อมูลในรูปแบบที่เป็นตาราง (Table) หรือเรียกว่า ความสัมพันธ์ (Relation) มีลักษณะเป็น 2 มิติ คือเป็นแถว (row) และเป็นคอลัมน์ (column) การเชื่อมโยงข้อมูลระหว่างตาราง จะเชื่อมโยง โดยใช้แอททริบิวต์หรือคอลัมน์ที่เหมือนกันทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลเชิงสัมพันธ์นี้จะเป็นรูปแบบของฐานข้อมูลที่นิยมใช้ในปัจจุบัน

- ฐานข้อมูลแบบเครือข่าย (Network Database) ฐานข้อมูลแบบเครือข่ายจะเป็นการรวมระเบียน (Record) ต่าง ๆ และความสัมพันธ์ระหว่างระเบียน แต่จะต่างกับฐานข้อมูลเชิงสัมพันธ์ คือ ในฐานข้อมูลเชิงสัมพันธ์จะแฝงความสัมพันธ์เอาไว้ โดยระเบียนที่มีความสัมพันธ์กันจะต้องมีค่าของข้อมูลในแอททริบิวต์ใดแอททริบิวต์หนึ่งเหมือนกัน แต่ฐานข้อมูลแบบเครือข่าย จะแสดงความสัมพันธ์อย่างชัดเจน

- ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database) ฐานข้อมูลแบบลำดับชั้น เป็นโครงสร้างที่จัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบพ่อ-ลูก (Parent-Child Relationship Type : PCR Type) หรือเป็นโครงสร้างรูปแบบต้นไม้ (Tree) ข้อมูลที่จัดเก็บในที่นี้ คือ ระเบียน ซึ่งประกอบด้วยค่าของเขตข้อมูล (Field) ของเอนทิตีหนึ่ง ๆ ฐานข้อมูลแบบลำดับชั้นนี้คล้ายคลึงกับ

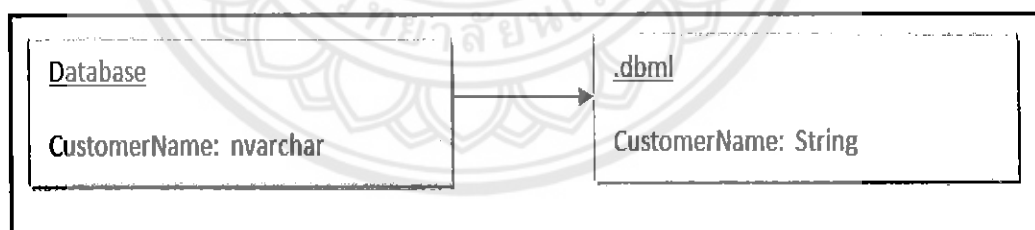
ฐานข้อมูลแบบเครือข่าย แต่ต่างกันว่าฐานข้อมูลแบบลำดับชั้น มีกฎเพิ่มขึ้นมาอีกหนึ่งประการ คือในแต่ละกรอบจะมีลูกศรวิ่งเข้าหาได้ไม่เกิน 1 หัวลูกศร

### 2.1.2 อีอาร์โมเดล (Entity - relationship Model : ER Model)

อีอาร์โมเดล(Entity - relationship Model : ER Model) เป็นแบบจำลองที่ใช้อธิบายความสัมพันธ์ระหว่างข้อมูล ซึ่งจะประกอบไปด้วยเอนทิตี(Entity) คือ แนวคิดรวบยอดที่ถูกกำหนดขึ้นเพื่อเป็นตัวแทนของข้อมูล โดยข้อมูลที่อยู่กลุ่มเดียวกัน จะมีคุณสมบัติพื้นฐานเหมือนกัน แอททริบิวต์(Attribute) คือ คุณสมบัติหรือลักษณะที่ใช้อธิบายรายละเอียดของเอนทิตีหรือความสัมพันธ์ (Relationship) ซึ่งความสัมพันธ์ที่กล่าวถึงนี้ เป็นความสัมพันธ์ระหว่างเอนทิตีตั้งแต่ 1 เอนทิตีขึ้นไป โดยความสัมพันธ์ที่แท้จริงเกิดจากสมาชิกของเอนทิตีที่สัมพันธ์กัน

## 2.2 การติดต่อกับฐานข้อมูลด้วย LINQ to SQL

LINQ To SQL(.NET Language-Integrated Query for Relational Data) เปลี่ยนชื่อมาจาก DLINQ และเป็น O/RM (Object-relational mapping) ที่จะ Implement ลงใน Visual Studio Orcas (VS2008) O/RM เป็นเทคนิคในการเขียน โปรแกรมเพื่อทำการแปลงโครงสร้างจากฐานข้อมูลให้มาอยู่ในรูปของอ็อบเจกต์ และในการทำ Mapping ออกมาจะต้องทำให้รูปแบบทางฝั่งฐานข้อมูลและทางฝั่งของอ็อบเจกต์สามารถเข้ากันได้



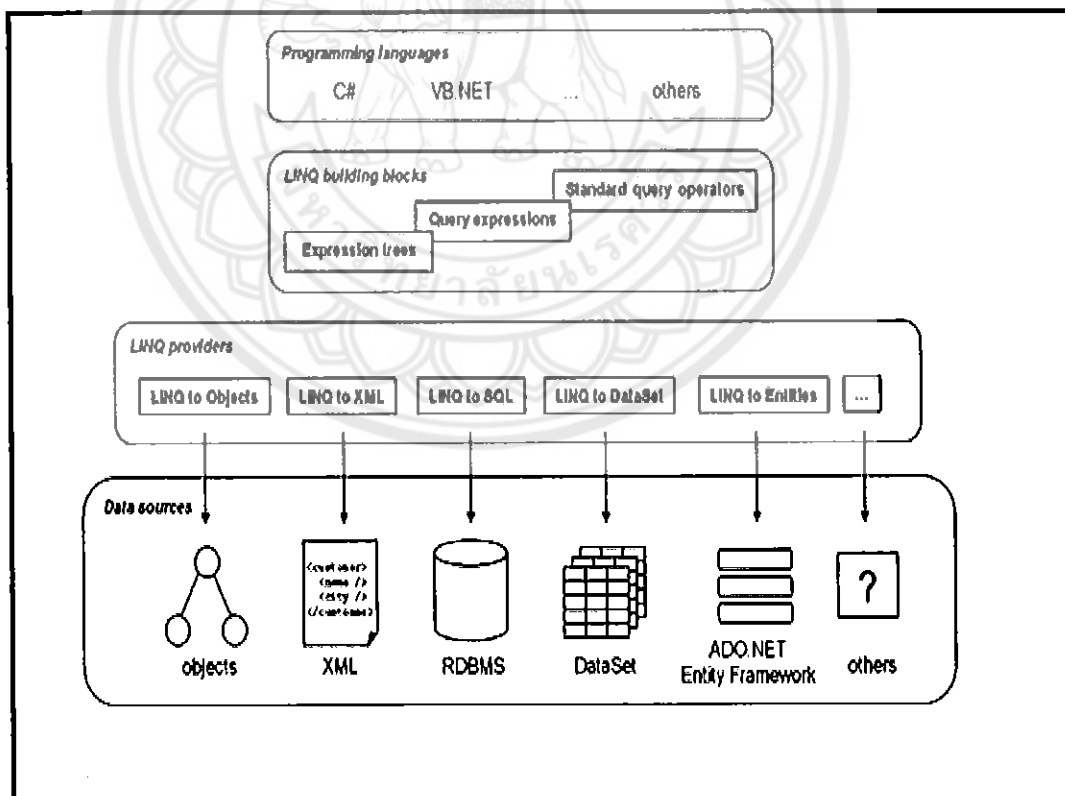
รูปที่ 2.1 การแปลงโครงสร้างจากฐานข้อมูลให้มาอยู่ในรูปของอ็อบเจกต์

โดยสร้าง virtual object database ขึ้นมาซึ่งจะต้องใช้เครื่องมือที่เป็น ORM Tools ในการสร้างสิ่งเหล่านี้ออกมา นอกจากนั้นตัว ORM Tools ก็จะช่วยสร้าง Class ออกมาแทน Object ต่างๆ ในฐานข้อมูล เช่น Table, View และสร้าง Function, Method ต่างๆ เพื่อใช้ในการ Select, Insert, Delete, Update ข้อมูลต่างๆ เพื่อจัดการกับข้อมูลในฐานข้อมูล ส่วนใน LINQ To SQL สิ่งที่เป็น virtual object database ก็คือ Linq to SQL File (.dbml)

### 2.2.1 ชุดเครื่องมือ LINQ

LINQ เป็นชุดเครื่องมือเพื่อใช้ทำงานกับข้อมูล ซึ่ง LINQ เป็นเครื่องมือที่มีเครื่องมือมาให้หลายอย่างดังนี้

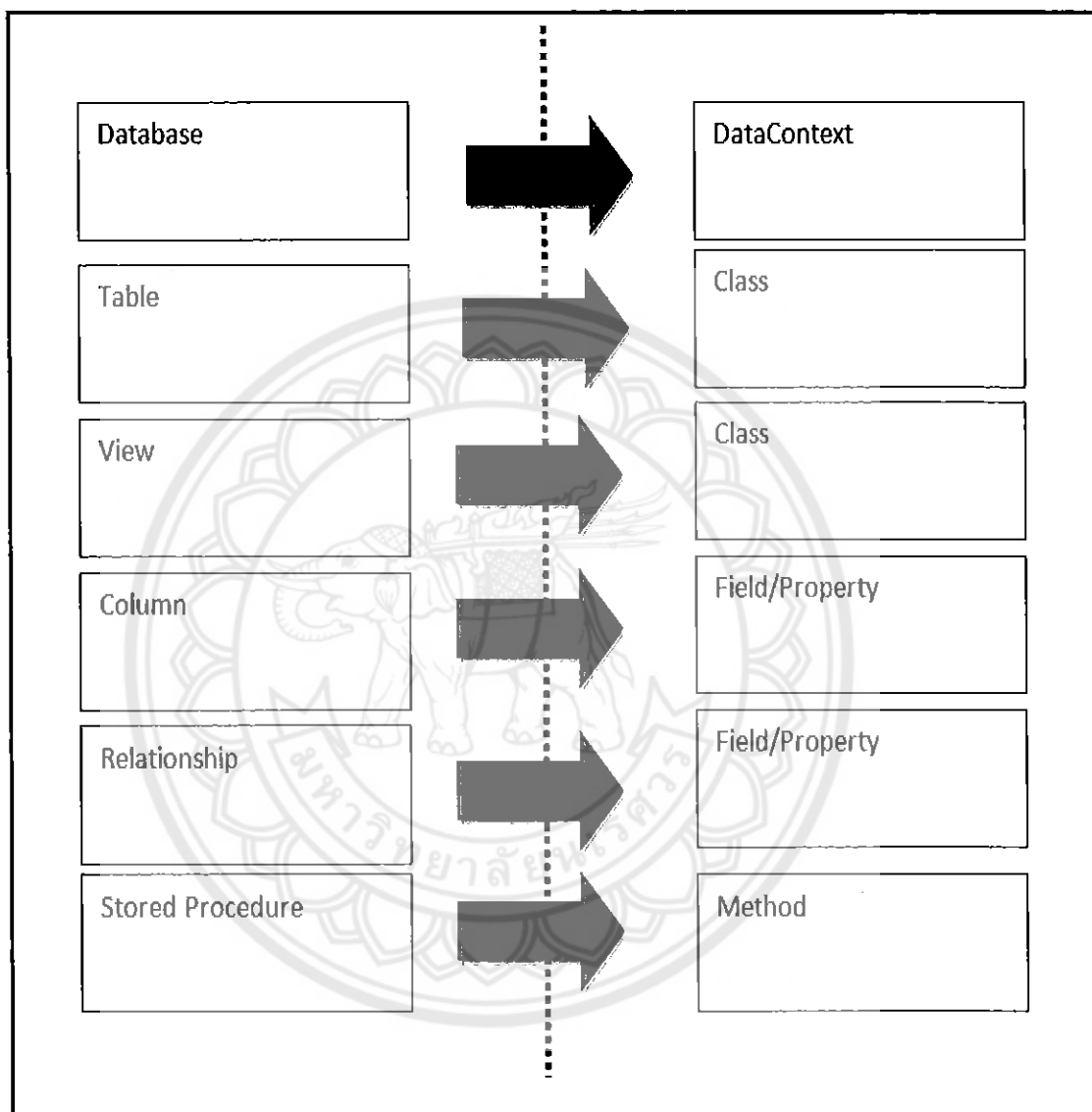
- LINQ to Object เป็นเครื่องมือที่ช่วยในการทำการควิรี่กับข้อมูลที่อยู่ภายในค่าคอลลเลกชันภายในหน่วยความจำ
- LINQ to DataSet เป็นเครื่องมือที่ช่วยในการทำการควิรี่กับข้อมูลที่อยู่ภายในค่าเซต
- LINQ to SQL: เป็นเครื่องมือที่ช่วยในการทำการควิรี่กับข้อมูลที่อยู่ภายในฐานข้อมูล
- LINQ to Entity: เป็นเครื่องมือที่ช่วยในการทำการควิรี่กับข้อมูลที่อยู่ในอ็อบเจกต์แบบ ADO.NET เอนทิตีเฟรมเวิร์ค
- LINQ to XML: เป็นเครื่องมือที่ช่วยในการทำการควิรี่กับข้อมูลที่อยู่ในแฟ้มข้อมูล XML



รูปที่ 2.2 สถาปัตยกรรม LINQ

### 2.2.2 การทำ Mapping ของ LINQ To SQL

การทำ Mapping ของ LINQ To SQL มีลักษณะดังรูปด้านล่าง โดยที่ฝั่งซ้ายจะแทนโครงสร้างของฐานข้อมูลและทางฝั่งขวาเป็นโครงสร้างของอ็อบเจกต์



รูปที่ 2.3 การทำ Mapping ของ LINQ To SQL

การทำ Mapping จากฐานข้อมูล จะมีส่วนที่เพิ่มเติมพิเศษขึ้นมาคือ DataContext ซึ่งตัว DataContext นี้เปรียบเหมือนท่อส่งหลักที่ใช้ในการดึงข้อมูลจากฐานข้อมูลและทำการส่งข้อมูลที่มีการเปลี่ยนแปลงกลับลงไปยังฐานข้อมูล เราสามารถที่จะใช้ตัว DataContext ได้เหมือนกับ ADO.NET Connection ทั่วไป (เช่น SqlConnection โดยที่มันสามารถระบุ Connection String ได้ หรือจะใช้จาก app.config ก็ได้)



### หน้าที่หลักของ DataContext คือ

1. การสร้างการเชื่อมต่อไปยังฐานข้อมูล Database
2. ทำการแปลง Query Syntax ให้เป็นคำสั่ง SQL เพื่อ Execute ไปยังตาราง
3. ทำการแปลงอ็อบเจกต์ต่างๆ ที่ถูกคิวรีขึ้นมา ให้กลับลงไปยังฐานข้อมูลในรูปแบบที่

Database Modeling(.dbml) ได้ทำการ Mapping เข้ามาเช่นการแปลงอ็อบเจกต์เหล่านั้นกลับ ไปเป็น Table หรือ Stored Procedure ต่างๆ เป็นต้น

### 2.3 Pivot

ในการทำรายงานสำหรับการนำเสนอข้อมูล ข้อมูลที่ดีเป็นสิ่งสำคัญมากที่จะช่วยให้ผู้ใช้สามารถตัดสินใจกำหนดทิศทางหรือ แก้ปัญหาต่างๆจากการวิเคราะห์ข้อมูลได้อย่างมั่นใจ ขอยกตัวอย่างกรณีของการผลิต ก่อนที่เราจะรวบรวมข้อมูล เราก็ต้องทราบก่อนว่า เราต้องการปรับปรุงพัฒนาในด้านใดบ้าง ลองนึกถึงภาพใหญ่ก่อนก็จะเห็นว่า ในส่วนของการผลิตนั้นสามารถพัฒนาปรับปรุงได้โดยการใช้วัตถุดิบให้คุ้มค่า, ใช้เครื่องจักรให้มีประสิทธิภาพ, ลดของเสีย, ลดเวลาการทำงาน ฯลฯ ซึ่งก็จะต้องมีการเก็บข้อมูลโดยการทำแบบฟอร์มให้เจ้าหน้าที่หน้างานบันทึก เช่นกรณีของเสีย ในแบบฟอร์มใบสั่งงานก็จะต้องมีช่องให้ลงจำนวนของเสีย และสาเหตุของเสียสำหรับแต่ละรายการไว้ อย่างไรก็ตามข้อมูลดิบตรงนี้แทบจะไม่มีประโยชน์อะไรเลย ถ้าเราไม่สามารถนำมาสรุปรวบรวมให้เห็นภาพ เพราะถ้าเราดูจากบันทึกใบสั่งที่ละใบ จะละลานตาไปหมด ขั้นตอนรวบรวมและสรุปข้อมูลนี้เรานำวิธีที่เรียกว่า pivot table มาใช้

การ Pivot Table นี้ เป็นการสรุปข้อมูล โดยการนำข้อมูลดิบมาทำการจัดกลุ่มให้ข้อมูลที่มีค่าเดียวกันอยู่ด้วยกันหรือรวมกันและข้อมูลที่มีค่าแตกต่างกันก็ถูกแยกเป็นอีกกลุ่มหนึ่ง ซึ่งประเด็นหลักของการ Pivot Table นี้ คือการเลือกคอลัมน์ใดๆ คอลัมน์หนึ่งขึ้นมานำเสนอกับอีกคอลัมน์ซึ่งมีความสัมพันธ์กันได้ โดยคอลัมน์เหล่านั้นอาจจะได้มาจากจัดกลุ่มข้อมูลทำการแบ่งไว้หรือเป็นคอลัมน์เดิมที่มีอยู่แล้วก็ได้

ตารางที่ 2.1 ตัวอย่างข้อมูลดิบที่ทำการจัดเก็บข้อมูล

ใบสั่งเลขที่	รายการที่	วัสดุ	ความหนา	จำนวนที่สั่ง	จำนวนของเสีย	สาเหตุ
P125	1	SUS304	2	15	0	
P125	2	SUS304	2	10	0	
P126	1	SS400	4	8	1	D01
P126	2	SS400	8	20	0	
P126	3	SUS316	2	4	0	
P127	1	SUS304	5	100	5	D05
P128	1	SUS304	4	25	0	
P128	2	SUS304	8	80	4	D02
P128	3	SS400	4	20	0	
P128	4	SS400	8	15	0	
P129	1	SUS304	3	35	6	D01
P129	2	SUS304	2	20	1	
P130	1	SUS316	5	18	0	
P212	2	SUS316	2	10	3	D02
P212	3	SS400	2	30	2	D02
P212	4	SS400	3	50	0	
P278	1	SUS304	3	80	4	D01
P278	2	SUS304	5	10	1	D05
P278	3	SUS304	5	40	0	

จากตารางที่ 2.1 จะเห็นได้ว่าข้อมูลที่ถูกจัดเก็บเป็นข้อมูลดิบที่มีบางข้อมูลในคอลัมน์ซ้ำกันบ้าง ข้อมูลถูกจัดเก็บอย่างกระจัดกระจาย เมื่อทำการ Pivot ข้อมูลเหล่านี้ ด้วยโปรแกรม MS-Excel โดยเรากำหนดลักษณะของรายงานที่ต้องการสรุป เช่น ผลรวมของเสียสำหรับแต่ละประเภทวัสดุ, ผลรวมของเสียสำหรับแต่ละความหนา, จำนวนของเสียของแต่ละสาเหตุ ฯลฯ ถ้าเรามีข้อมูลดิบมากกว่านี้ เช่นวันที่ผลิตของแต่ละใบสั่ง ก็สามารถสรุปเป็นของเสีย แต่ละสัปดาห์ หรือ เดือน หรือ ถ้าเรามีการบันทึกหมายเลขเครื่องจักร หรือ ชื่อพนักงานที่ปฏิบัติงาน ก็จะได้ข้อมูลที่ละเอียดขึ้น ซึ่งก็อาจจะทำให้เราเห็นได้ว่า ของเสียแบบนี้มักจะมาจากรุ่นนี้ หรือพนักงานคนนี้

ตารางที่ 2.2 การ Pivot ข้อมูล

ความหนา	ข้อมูล	วัสดุ			ผลรวมทั้งหมด
		SS400	SUS304	SUS316	
2	ผลรวม ของ จำนวนที่ตั้ง	30	45	14	89
	ผลรวม ของ จำนวนของเสีย	2	1	3	6
3	ผลรวม ของ จำนวนที่ตั้ง	50	115		165
	ผลรวม ของ จำนวนของเสีย	0	10		10
4	ผลรวม ของ จำนวนที่ตั้ง	28	25		53
	ผลรวม ของ จำนวนของเสีย	1	0		1
5	ผลรวม ของ จำนวนที่ตั้ง		150	18	168
	ผลรวม ของ จำนวนของเสีย		6	0	6
8	ผลรวม ของ จำนวนที่ตั้ง	35	80		115
	ผลรวม ของ จำนวนของเสีย	0	4		4
ผลรวม ผลรวม ของ จำนวนที่ตั้ง		143	415	32	590
ผลรวม ผลรวม ของ จำนวนของเสีย		3	21	3	27

จากหลักการทำงานของ MS-Excel ด้วยการ Pivot tableข้างต้น จะทำให้ได้มาซึ่งรายงานการสรุปออกมาเป็นตัวเลขชัดเจนนั้น ทำให้เราสามารถวิเคราะห์ข้อมูล จัดลำดับความสำคัญของสาเหตุ และกำหนดทิศทางในการแก้ไขปัญหาได้ง่ายขึ้น

## 2.4 เว็บแอปพลิเคชัน (Web application)

เว็บแอปพลิเคชันคือ แอปพลิเคชันที่เข้าถึงด้วยเว็บเบราว์เซอร์ผ่านเครือข่ายคอมพิวเตอร์อย่างอินเทอร์เน็ต หรืออินทราเน็ต เว็บแอปพลิเคชันเป็นที่นิยมมากเนื่องจากมีความสามารถในการอัปเดตและดูแลโดยไม่ต้องแจกจ่ายและติดตั้งซอฟต์แวร์บนเครื่องของผู้ใช้

เว็บแอปพลิเคชันมีจุดเด่นที่สำคัญ คือ การทำงานที่หลากหลายและไม่จำกัดพื้นที่ ซึ่งเปิดโอกาสให้ผู้ใช้สามารถเข้าถึงข้อมูลที่จำเป็นได้ทุกที่

สถาปัตยกรรมแบบเครื่องลูกข่าย (เครื่องลูกข่าย) และโปรแกรมบริการ (Server)

เครื่องลูกข่าย (เครื่องลูกข่าย) คือ แอปพลิเคชัน โปรแกรมที่ต้องการข้อมูลหรือบริการจากโปรแกรมอื่น โดยโปรแกรมที่ให้บริการจะรันอยู่บนโปรแกรมบริการ

โปรแกรมบริการ (Server) คือส่วนที่ทำหน้าที่ในการให้บริการและข้อมูลแก่เครื่องลูกข่าย โดยโปรแกรมบริการ จะแปลความหมายของคำขอบริการ แล้วนำข้อมูลนั้นส่งกลับไปยังลูกข่าย

## 2.5 เจสัน (JSON)

สัญกรณ์เชิงวัตถุจาวาสคริปต์ (JSON : JavaScript Object Notation) หรือเจสันเป็นไวยากรณ์ที่ใช้สำหรับจัดเก็บและแลกเปลี่ยนข้อความเหมือนกับเอ็กซ์เอ็มแอล(XML) มีขนาดเล็ก เร็ว และจำแนกได้ง่ายกว่าเอ็กซ์เอ็มแอล เป็นไวยากรณ์การรับส่งข้อความที่ไม่ขึ้นอยู่กับแพลตฟอร์ม(Independent Platform) หมายความว่า ทุกๆภาษาสามารถใช้ไวยากรณ์นี้ในการแลกเปลี่ยนข้อความระหว่างกันได้ ตัวอย่างของไวยากรณ์เจสันแสดงดังรูปต่อไปนี้

```

JSON Example
{
  "employees": [
    { "firstName": "John", "lastName": "Doe" },
    { "firstName": "Anna", "lastName": "Smith" },
    { "firstName": "Peter", "lastName": "Jones" }
  ]
}

```

รูปที่ 2.4 ตัวอย่างไวยากรณ์เจสัน

จากรูปแสดงถึงอ็อบเจกต์ "employees" ที่เป็นอาร์เรย์ซึ่งภายในอ็อบเจกต์ "employees" ประกอบไปด้วยระเบียน 3 ระเบียนหรืออ็อบเจกต์ 3 อ็อบเจกต์ ใน 1 อ็อบเจกต์ จะประกอบไปด้วยคู่ของคีย์กับค่าข้อมูล ("Key" : "Value") การเข้าถึงข้อมูลในแต่ละระเบียนนั้นเราจะอ้างผ่านคีย์ของระเบียนนั้นๆ เช่น

ต้องการเข้าถึงข้อมูลชื่อ(firstName)ของอ็อบเจกต์ "employees" ในระเบียนที่ 3 สามารถทำได้ดังนี้

employees[2].firstName จะได้ข้อมูลเป็น "Peter"

ต้องการเข้าถึงข้อมูลนามสกุล(lastName)ของอ็อบเจกต์ "employees" ในระเบียนที่ 1 สามารถทำได้ดังนี้

employees[0].lastName จะได้ข้อมูลเป็น "Doe"

### 2.5.1 ไวยากรณ์เจสัน (JSON Syntax)

- ข้อมูลประกอบไปด้วยคู่ของคีย์และค่าข้อมูล "Key" : "Value"
- ข้อมูลแต่ละข้อมูลจะแยกจากกันด้วยเครื่องหมายจุลภาค ","

- ข้อมูลหลายๆข้อมูลรวมกันเป็น 1 อีปเจกต์หรือ 1 ระเบียบ
- กำหนดให้ใน 1 อีปเจกต์หรือ 1 ระเบียบจะเปิดและปิดด้วยเครื่องหมายปีกกา "{" และ "}"
- อีปเจกต์แต่ละอีปเจกต์จะแยกจากกันด้วยเครื่องหมายจุดภาค ","
- กรณีมีหลายๆ อีปเจกต์รวมกันเป็นอาร์เรย์ 1 ก็จะต้องเปิดและปิดด้วยเครื่องหมายปีกกา "[" และ "]"

```
"firstName" : "John"
```

รูปที่ 2.5 ตัวอย่างข้อมูลประกอบไปด้วยคู่ของคีย์และค่าข้อมูล

### 2.5.2 ชนิดของข้อมูลภายในเจสัน

- ชนิดของข้อมูลในเจสันสามารถเป็นได้ดังนี้
- ข้อมูลตัวเลข ได้แก่ ตัวเลขจำนวนเต็ม(integer) และตัวเลขทศนิยม(floating point) ไม่ต้องมีเครื่องหมาย "" หุ้ม เช่น "data1" : 2 , "data2" : 8.41 เป็นต้น
  - ข้อมูลตัวอักษร(string) กำหนดให้ต้องมีเครื่องหมาย "" หุ้ม เช่น "word" : "na5cent" เป็นต้น
  - ข้อมูลทางตรรกะ(logical) ประกอบไปด้วย จริง(true) และเท็จ(false) ไม่ต้องมีเครื่องหมาย "" หุ้ม
  - ข้อมูลที่เป็นอาร์เรย์ อยู่ภายในวงเล็บ [ ]
  - ข้อมูลที่เป็นอีปเจกต์อยู่ภายในวงเล็บ { }
  - ข้อมูลที่ไม่ทราบชนิด(null)

### 2.5.3 เปรียบเทียบเจสันกับเอ็กซ์เอ็มแอล

- เจสันและเอ็กซ์เอ็มแอล เป็นข้อความธรรมดา(plain text)
- เจสันและเอ็กซ์เอ็มแอล มีความเป็นลำดับชั้น(hierarchical)
- เจสันไม่ต้องมีแท็ก (Tag)กำกับ แต่เอ็กซ์เอ็มแอลต้องมีแท็กกำกับ
- เจสันมีความไวยากรณ์ที่สั้นกะทัดรัด
- เจสันสามารถอ่านและเขียนได้เร็วกว่า
- เจสันไม่มีคำสงวน

## 2.5.4 ประโยชน์ของเจสัน

- เนื่องจากการใช้ไวยากรณ์เจสันมีขนาดเล็ก ส่งผลให้การรับส่งข้อมูลเป็นได้ได้อย่างรวดเร็ว รวมถึงการประมวลผลที่เร็วขึ้น เพราะเป็นแค่ข้อความธรรมดา
- เนื่องจากเจสันมีการทำงานแบบ cross platform คือใช้กับภาษาอะไรก็ได้ เราจึงสามารถเขียน โปรแกรมให้ภาษาหนึ่งให้ติดต่อกับอีกภาษาหนึ่งได้ โดยการใช้เจสันนั้น เป็นตัวกลาง เช่น ใช้ php ติดต่อกับ android(java), php ติดต่อกับ javascript และอื่นๆ เป็นต้น

## 2.6 เอแจ็กซ์ (AJAX)

ปัจจุบันนี้ ลักษณะการทำงานแบบเครื่องลูกข่าย (เครื่องลูกข่าย) และโปรแกรมบริการ (Server) เริ่มถูกนำมาใช้งานอย่างแพร่หลายในลักษณะการติดต่อสื่อสารผ่านทางเว็บเบราว์เซอร์ ซึ่งการทำงานแบบนี้ จะมีการทำงานโดยเครื่องลูกข่ายจะร้องขอและต้องการข้อมูลบางอย่างจากโปรแกรมบริการ ดังนั้นการโหลดและการรีเฟรชหน้าจอ เป็นสิ่งที่หลีกเลี่ยงไม่ได้ จึงเป็นผลให้การทำงานของฝั่งเครื่องลูกข่ายหรือผู้ใช้ต้องหยุดการ โหลดและการรีเฟรชหน้าจอ ซึ่งถือว่าการทำงานที่ไม่มีประสิทธิภาพ

เอแจ็กซ์ไม่ใช่ชื่อของการเขียนโปรแกรมหรือเป็นชื่อของภาษาที่ใช้ในการเขียนโปรแกรม แต่เป็นชุดของเทคโนโลยีต่างๆ Ajax ย่อมาจาก Asynchronous JavaScript And XML ซึ่งหมายถึงการทำงานร่วมกันของจาวาสคริปต์และXML แบบ Asynchronous มีหลักการทำงาน 2 ประเด็น คือ การอัปเดตหน้าจอแบบบางส่วน และการติดต่อสื่อสารกับโปรแกรมบริการ โดยใช้หลักการ Asynchronous ทำให้ผู้ใช้ไม่ต้องหยุดการทำงาน เพื่อรอการประมวลผลจากโปรแกรมบริการ รวมถึงการโหลดและการรีเฟรชหน้าจอของเบราว์เซอร์ทางฝั่งเครื่องลูกข่ายมีการใช้เอแจ็กซ์ โดยการเพิ่มเลเยอร์ระหว่าง ผู้ใช้เบราว์เซอร์กับโปรแกรมบริการ ทำให้ผู้ใช้สามารถทำงานได้โดยไม่ต้องรอให้เครื่องลูกข่ายติดต่อไปยังโปรแกรมบริการ รวมถึงการ โหลดและการรีเฟรชหน้าจอทั้งหมดด้วย ดังนั้นผู้ใช้สามารถใช้งานแอปพลิเคชันได้อย่างมีประสิทธิภาพมากขึ้น

### 2.6.1 ประวัติความเป็นมา

ในช่วงแรกๆ ของการพัฒนา คือปี ค.ศ. 1997 นั้น Component แรกที่เกิดขึ้นทางฝั่งเครื่องลูกข่าย ถูกเขียนขึ้น โดยทีมพัฒนา Outlook Web Access ซึ่งต่อมาถูกนำมาใช้เป็นส่วนหนึ่งของ Internet Explorer 5.0 นั่นก็คือจุดเริ่มต้นที่เริ่มรู้จักการทำงานแบบเอแจ็กซ์และในปี ค.ศ. 2005

Google ได้ใช้การติดต่อสื่อสารแบบ Asynchronous เพื่อเป็นรากฐานที่ทำให้รู้จักกับเอแจ็กซ์กัน อย่างแพร่หลาย การทำงานแบบ เครื่องลูกข่าย - Server ถูกนำมาใช้งานเป็นจำนวนมาก เช่น การติดต่อกับฐานข้อมูลที่โปรแกรมบริการ หรือการให้บริการทางอินเทอร์เน็ต ซึ่ง Google เป็นผู้ลงทุน ลงแรงอย่างหนัก ในพัฒนาและการทดสอบเอแจ็กซ์จึงสังเกตได้ว่า ผลผลิตใหญ่ของ Google ในช่วงต้นปีค.ศ. 2005 จึงเป็นการนำ เอแจ็กซ์มาประยุกต์ใช้งาน เช่น Gmail, Google Map, Google Suggest และ Google Group เป็นต้น

## 2.6.2 ที่มาของปัญหา

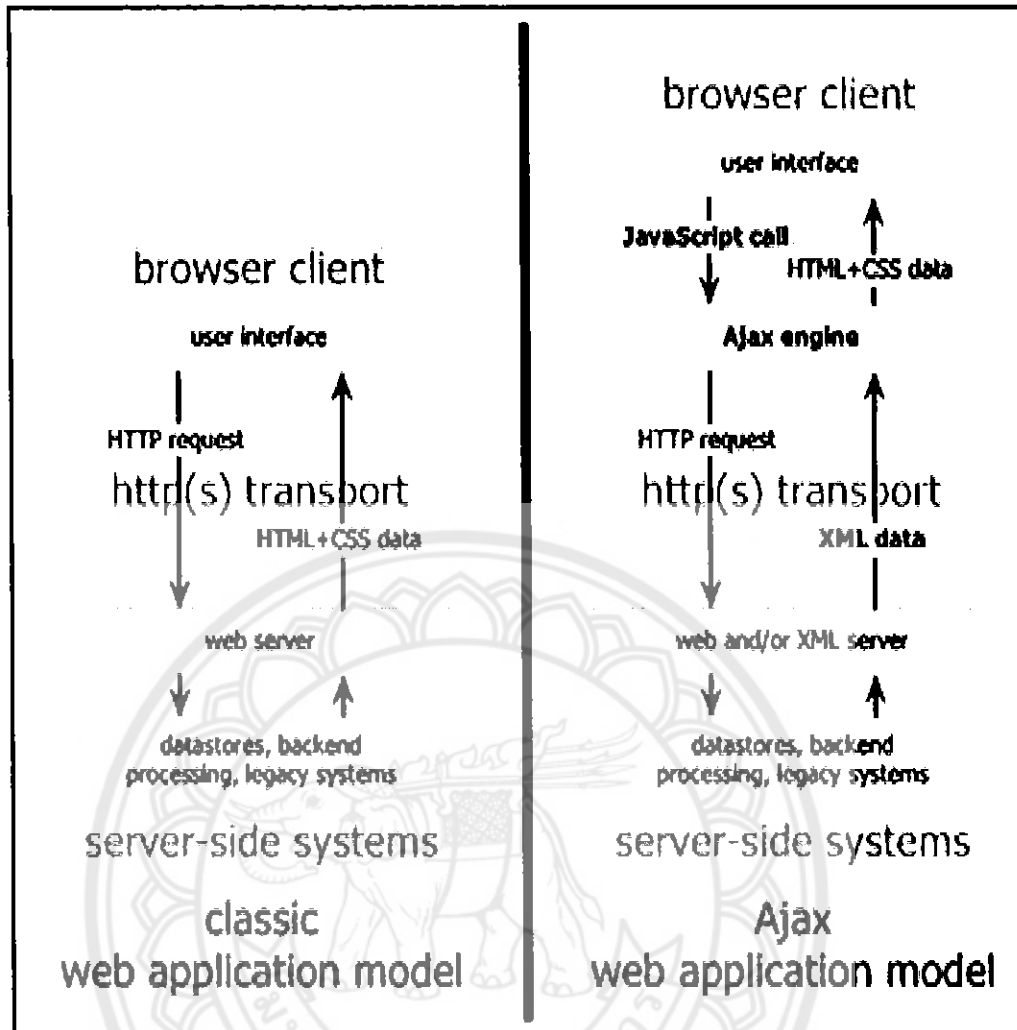
เนื่องจากแอปพลิเคชันที่ใช้งานในปัจจุบันนี้ มีหลักการที่ทำงานแล้วเกิดการสูญเสียเวลาและทรัพยากรของผู้ใช้ในการรอคอยการทำงานต่างๆ ทำให้ผู้ใช้ต้องหยุดคอย ดังนั้นการทำงานของผู้ใช้จึงเป็นไปอย่างไม่ต่อเนื่อง ซึ่งหลักการดังกล่าวคือ

- "Click, wait, and refresh" user interaction paradigm

การที่เบราว์เซอร์ตอบสนองต่อการทำงานของผู้ใช้ โดยจะทิ้งหน้าเว็บที่แสดงอยู่ในขณะนั้น แล้วไปทำการส่ง HTTP request กลับไปยังโปรแกรมบริการแทน ซึ่งทำให้ผู้ใช้ไม่สามารถทำอะไรได้เลยในขณะนั้น นอกจากการรอคอยเมื่อโปรแกรมบริการทำการประมวลผลเสร็จ ก็จะส่งหน้า HTML กลับมายังเบราว์เซอร์ ต่อจากนั้นเบราว์เซอร์ก็จะรีเฟรชและแสดงหน้า HTML หน้าใหม่ และนี่เองที่ทำให้ผู้ใช้สามารถใช้งานต่อไปได้ จะเห็นว่า ผู้ใช้มีช่วงเวลาของการหยุดรอคอยเป็นเวลานานสำหรับการประมวลผลของโปรแกรมบริการและการรีเฟรชหน้า HTML ใหม่ทั้งหน้า ซึ่งเป็นสิ่งที่ไม่มีประสิทธิภาพในเชิง Dynamic ของการทำงานบนเว็บแอปพลิเคชัน

- Synchronous "request/response" communication mode

การที่เบราว์เซอร์เริ่มทำการร้องขอข้อมูล และโปรแกรมบริการก็ตอบสนองเฉพาะการร้องขอที่เบราว์เซอร์ร้องขอมาโปรแกรมบริการจะไม่สามารถส่งข้อมูลได้ถ้าเบราว์เซอร์ไม่ได้ร้องขอข้อมูลในขณะนั้น ซึ่งถือว่าการติดต่อสื่อสารเป็นแบบทิศทางเดียว วงจรการ request/response แบบ synchronous คือ การทำงานแบบประสานจังหวะระหว่างเบราว์เซอร์กับโปรแกรมบริการทำให้เกิดความล่าช้าในการทำงานทำให้ผู้ใช้ทำอะไรไม่ได้อีก นอกจากการคอยการตอบสนองกลับมาจาก เมื่อโปรแกรมบริการประมวลผลเสร็จ

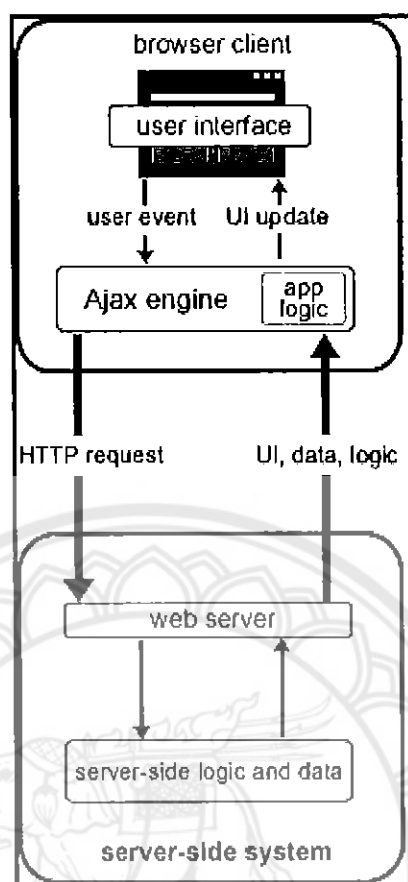


รูปที่ 2.6 เปรียบเทียบการทำงานแบบเดิมกับเอแจ็กซ์

### 2.6.3 โครงสร้างของเอแจ็กซ์

มุมมองของโครงสร้างทางซอฟต์แวร์ของเอแจ็กซ์ต่างจากเว็บแอปพลิเคชันในทุกวันนี้ เนื่องจากมีการเพิ่ม engine ทางฝั่งเครื่องลูกข่าย





รูปที่ 2.7 สถาปัตยกรรมของเอแจ็กซ์

จากรูป Ajax engine นี้ อยู่ระหว่าง User Interface กับ โปรแกรมให้บริการ ซึ่งจะมองว่าเป็นการทำงานที่เครื่องลูกข่าย การทำงานต่างๆของผู้ใช้โปรแกรมจะไปเรียก Ajax engine ตัวนี้ขึ้นมาแทนที่การร้องขอหน้าเว็บจากโปรแกรมบริการ โดยตรง และจะใช้โครงสร้างข้อมูลแบบXML ในการขนย้ายข้อมูลระหว่างโปรแกรมบริการกับ Ajax engine เมื่อเบราว์เซอร์ทำการร้องขอข้อมูลจากโปรแกรมบริการ นอกจากนี้ Ajax engine ไม่ต้องทำการติดตั้ง ไม่ใช่ปลั๊กอิน (plug-in) และไม่สามารถดาวน์โหลดได้เพราะเอแจ็กซ์เป็นแนวคิดในการแก้ปัญหาคาการหยุดชะงักการทำงานของผู้ใช้

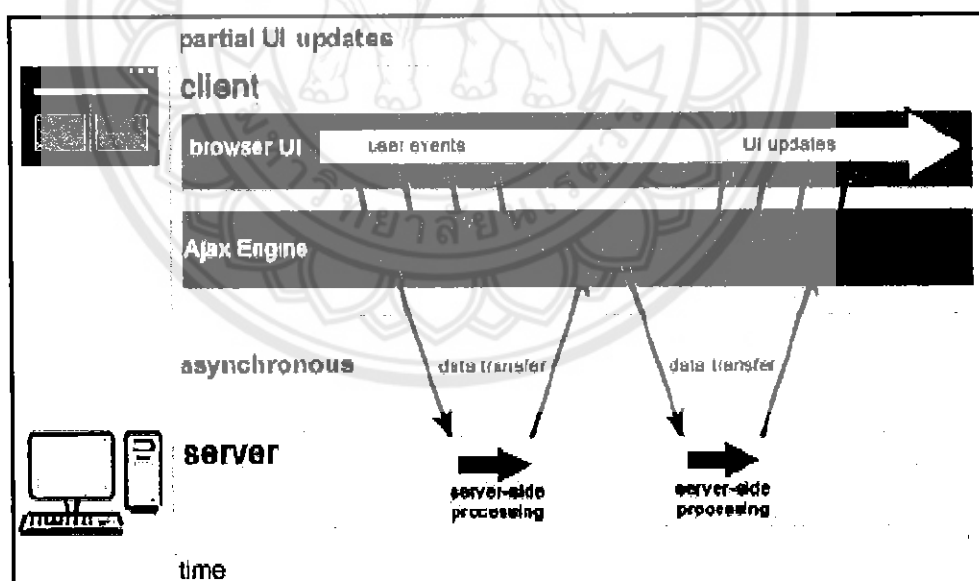
#### 2.6.4 การทำงานของเอแจ็กซ์

เอแจ็กซ์จะช่วยลดการติดต่อระหว่างเครื่องลูกข่ายกับโปรแกรมบริการ โดยในการโหลดหน้าเว็บนั้น เบราวเซอร์จะโหลดข้อมูลจาก AJAX engine แทนการร้องขอข้อมูลจากโปรแกรมบริการโดยตรง ดังนั้นเอแจ็กซ์จะทำหน้าที่ทั้งการ render ส่วนติดต่อกับผู้ใช้และติดต่อไปยังโปรแกรมแล้ว AJAX engine อนุญาตให้การกระทำต่างๆ ในเว็บแอปพลิเคชันเป็นแบบ

Asynchronous คือความเป็นอิสระในการติดต่อไปยังโปรแกรมบริการนั่นเอง ดังนั้นผู้ใช้จะไม่พบกับบราวเซอร์หน้าขาวๆ อีกต่อไป และไม่ต้องรอกการโหลดข้อมูลต่างๆ จากโปรแกรมบริการ

การอัปเดตหน้าจอบางส่วน แทนที่การ "click, wait, and refresh" ระหว่างที่เกิดการทำงานแบบการติดต่อสื่อสารของผู้ใช้ user interface ที่ต้องนำมาแสดงซ้ำในหน้าเว็บที่ร้องขอไปยังโปรแกรมบริการจะถูกจัดเป็นข้อมูลใหม่เมื่อถูกอัปเดตแล้ว การหยุดชะงักของ user interface จึงไม่เกิดขึ้น เพราะหน้าเว็บนั้นยังคงถูกแสดงอยู่และสามารถใช้งานได้ โดยปราศจากการหยุดชะงักการทำงานของผู้ใช้ การอัปเดตหน้าเว็บบางส่วนสามารถทำให้หน้าเว็บทำงานต่อไปได้ ถึงจะไม่ใช่ว่าทั้งหมด แต่อย่างน้อยก็ทำให้การทำงานไม่จำเป็นต้องหยุดชะงักเลย

การติดต่อแบบ Asynchronous เข้ามาแทนที่การ "synchronous request/response model" สำหรับแอพลิเคชันการ request/response จะทำแบบ asynchronous ซึ่งคือการติดต่อสื่อสารกับโปรแกรมบริการ แบบอิสระ โดยทำการลดการติดต่อระหว่างบราวเซอร์กับโปรแกรมบริการ ผลที่ได้ก็คือผู้ใช้สามารถใช้งานเว็บแอปพลิเคชันได้ในขณะที่เครื่องลูกข่าย ทำการร้องขอข้อมูลจากโปรแกรมบริการ อยู่เบื้องหลัง เมื่อข้อมูลเดินทางมาถึงบราวเซอร์ก็จะอัปเดตหน้า user interface ที่ต้องการข้อมูลใหม่ ส่วนหน้า user interface ที่ไม่ต้องการอัปเดตก็จะแสดงส่วนนั้นต่อไป



รูปที่ 2.8 การทำงานแบบ Asynchronous และการอัปเดตหน้าเว็บแบบบางส่วน

### 2.6.5 ข้อดีของเอแจ็ทซ์

- ตอบสนองต่อผู้ใช้ได้อย่างรวดเร็วเนื่องจากการอัปเดต แบบบางส่วน
- ผู้ใช้ไม่ต้องหยุดรอคอยการประมวลผลของโปรแกรมบริการ เนื่องจากการติดต่อแบบ Asynchronous
- รองรับกับบราวเซอร์หลักๆที่สามารถใช้จาวาสคริปต์ได้
- ทำให้การประมวลผลที่โปรแกรมบริการมีความรวดเร็วขึ้นเนื่องจากการ

ประมวลผลที่โปรแกรมบริการลดลง

- ไม่ต้องทำการติดตั้ง หรือใช้ปลั๊กอิน (Plugs-in)
- ไม่ยึดติดกับแพลตฟอร์ม หรือภาษาที่ใช้ในการเขียนโปรแกรม
- เป็นเทคโนโลยีใหม่ที่ไม่ได้เป็นของนักพัฒนาเว็บแอปพลิเคชันคนใด นั่นคือทุกคนมีสิทธิ์เข้ามาพัฒนาแอปพลิเคชันตัวนี้

## 2.7 จาวาสคริปต์ (Javascript)

จาวาสคริปต์เป็นภาษาโปรแกรม (Programming language) ประเภทหนึ่ง ที่เรียกกันว่า สคริปต์ (script) ซึ่งมีวิธีการทำงานในลักษณะแปลความและดำเนินงานไปที่ละคำสั่ง (interpret) ภาษานี้เดิมมีชื่อว่า LiveScript ได้รับการพัฒนาขึ้นโดย Netscape ด้วยวัตถุประสงค์ เพื่อที่จะช่วยให้เว็บเพจสามารถแสดงเนื้อหาที่มีการเปลี่ยนแปลงไปได้ ตามเงื่อนไขหรือสภาพแวดล้อมต่างๆกัน หรือสามารถโต้ตอบกับผู้ใช้ได้มากขึ้น ทั้งนี้เพราะภาษา HTML แต่เดิมนั้น เหมาะสำหรับใช้แสดงเอกสารที่มีเนื้อหาคงที่แน่นอน และไม่มีลูกเล่นอะไรมากมายนัก

เนื่องจากจาวาสคริปต์ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิดที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA ซึ่งเราจะพบว่าปัจจุบัน จะหาเว็บเพจที่ไม่ใช้จาวาสคริปต์เลยนั้น ได้ยากเต็มที

การทำงานของจาวาสคริปต์จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ ดังนั้นจาวาสคริปต์จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุนจาวาสคริปต์แล้ว อย่างไรก็ตามสิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นรุ่นใหม่ๆออกมาด้วย ดังนั้น ถ้านำโค้ดของรุ่นใหม่ๆไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิดข้อผิดพลาดได้

การทำงานของจาวาสคริปต์เกิดขึ้นบนบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้โปรแกรมบริการอะไร หรือที่ไหน ก็ยังคงสามารถใช้จาวาสคริปต์ในเว็บเพจได้ ต่างกับ

ภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องโปรแกรมบริการ (เรียกว่า server-side script) ดังนั้นจึงต้องใช้บนโปรแกรมบริการที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ตามถ้าจากลักษณะดังกล่าวก็ทำให้จาวาสคริปต์มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับโปรแกรมบริการโดยตรง เช่น การอ่านไฟล์จากโปรแกรมบริการเพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ใช้ เพื่อนำไปเก็บบนโปรแกรมบริการ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่ (ความจริงจาวาสคริปต์ ที่ทำงานบนโปรแกรมบริการก็ยังมีซึ่งต้องอาศัยโปรแกรมบริการที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก)

การทำงานของจาวาสคริปต์จะมีประสิทธิภาพมาก ถ้ามันสามารถคัดแปลงคุณสมบัติขององค์ประกอบต่างๆ บนเว็บเพจ (เช่น สี หรือรูปแบบของข้อความ) และสามารถรับรู้เหตุการณ์ ที่ผู้ใช้เว็บเพจโต้ตอบกับองค์ประกอบเหล่านั้น เช่น การคลิก หรือเลื่อนเมาส์ไปวาง ได้ ดังนั้นจากภาษา HTML เดิม ที่มีลักษณะสถิต (static) ใน HTML รุ่นใหม่ๆ จึงได้มีการพัฒนาให้มีคุณสมบัติบางอย่างเพิ่มขึ้น และมีลักษณะเป็นอ็อบเจกต์ มากขึ้น การทำงานร่วมกันระหว่างคุณสมบัติใหม่ของ HTML ร่วมกับจาวาสคริปต์นี้เอง ทำให้เกิดเป็นสิ่งที่เรียกว่า Dynamic HTML คือภาษา HTML ที่สามารถใช้สร้างเว็บเพจที่มีลักษณะพลวัต (dynamic) ได้นั่นเอง นอกจากนี้อีกองค์ประกอบหนึ่งที่เกี่ยวข้อง ก็คือ Cascading Style Sheet (CSS) ซึ่งเป็นภาษาที่ช่วยให้เราควบคุมรูปแบบ ขององค์ประกอบต่างๆ บนเว็บเพจ ได้อย่างมีประสิทธิภาพมากกว่าคำสั่ง หรือแท็ก (tag) ปกติของ HTML เนื่องจากจาวาสคริปต์สามารถคัดแปลงคุณสมบัติของ CSS ได้เช่นกัน ดังนั้นมันจึงช่วยให้เราควบคุมเว็บเพจ ได้อย่างมีประสิทธิภาพมากยิ่งขึ้นไปอีก

## 2.8 เจควีรี่ (jQuery)

เจควีรี่เป็นจาวาสคริปต์ไลบรารี (JavaScript Library) ที่มีการรวบรวมฟังก์ชันของจาวาสคริปต์ต่าง ๆ ให้อยู่ในรูปแบบแพทเทิร์นเฟรมเวิร์ค (Patterns Framework) ที่สะดวกและง่ายต่อการใช้งาน มีความยืดหยุ่นรองรับต่อการใช้งานข้ามเบราว์เซอร์ (Cross Browser) คือไม่ว่าจะใช้งานบนเว็บเบราว์เซอร์ในไลบรารีของเจควีรี่ จะมีการเลือกใช้ฟังก์ชันที่สามารถทำงานและเหมาะสมต่อการแสดงผลในเว็บเบราว์เซอร์ที่กำลังรันอยู่ ซึ่งช่วยลดปัญหาการทำงานที่ผิดพลาดในฝั่งของไคลเอนต์ (เครื่องลูกข่าย) ได้ ปัจจุบันมีเว็บเบราว์เซอร์เกิดขึ้นมากมาย เช่น Chrome, Firefox หรือ Safari และบางคำสั่งของจาวาสคริปต์ไม่สามารถทำงานในเว็บเบราว์เซอร์บางตัวได้ ด้วยเหตุผลนี้เองการใช้เจควีรี่ จึงเป็นทางเลือกที่สามารถช่วยแก้ปัญหาเหล่านี้เป็นได้อย่างดี ทั้งยังสะดวกต่อการใช้งาน เพราะเป็นภาษาที่เข้าใจง่ายและเขียนได้ในรูปแบบสั้นๆ รองรับการทำงานทั้งในเอชทีเอ็มแอล (HTML) รูปแบบเดิม หรือ CSS, element, DOM element, effect การจัดการ Event ต่าง ๆ หรือแม้กระทั่งการพัฒนาอ็อบเจกต์ด้วยเจควีรี่ ก็สามารถทำได้อย่างง่ายดาย โดยภาษาเหล่านี้ยังคงทำงาน

อยู่ภายใต้คำสั่งของภาษาจาวาสคริปต์ แต่การเรียกใช้งานเฟรมเวิร์คหรือฟังก์ชันต่าง ๆ จะถูกกำหนดรูปแบบโดยรูปแบบที่ได้ถูกออกแบบไว้ไลบรารีของเจคิววี

## 2.9 Application Programming Interface (API)

Application Programming Interface คือ ช่องทางการเชื่อมต่อจากระบบหนึ่งไปสู่ระบบอื่นๆ เพื่อเรียกใช้บริการหรือข้อมูลจากผู้ให้บริการ โดยการเรียกใช้นั้นก็แตกต่างกันไปขึ้นอยู่กับผู้ให้บริการ หรืออาจกล่าวได้ว่าเป็นไลบรารี ที่องค์กร บริษัท หรือนักพัฒนาได้สร้างขึ้นมาทำงานอย่างใดอย่างหนึ่ง เพื่อให้ นักพัฒนาไม่ต้องเขียนระบบหรือฟังก์ชันส่วนนั้นเองมากมาย เป็นการลดความยุ่งยาก ซับซ้อน และน่าเบื่อของการพัฒนา ซึ่งพบว่าสามารถช่วยให้เราได้พัฒนาแอปพลิเคชันได้ง่ายและรวดเร็ว เป็นระบบมากขึ้นด้วย ซึ่งโดยสรุปแล้วเอพีไอ คือ โมดูลหรือไลบรารีตัวหนึ่งที่สามารถนำมาพัฒนาร่วมกับแอปพลิเคชันได้ โดยไม่ต้องเข้าไปเกี่ยวข้องกับโค้ดของเอพีไอ

เอพีไอ แบ่งเป็น 2 แบบ คือ

- เอพีไอที่ขึ้นกับภาษา (language-dependent API) คือ เอพีไอ ที่สามารถการเรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาใดภาษาหนึ่ง
- เอพีไอไม่ขึ้นกับภาษา (language-independent API) คือ เอพีไอ ที่สามารถเรียกใช้จากโปรแกรมหลายๆภาษา

## 2.10 Google Visualization API

Google Visualization API เป็นรูปแบบการนำเสนอที่ Google ได้พัฒนาขึ้น เพื่อให้ผู้พัฒนาเว็บแอปพลิเคชัน สามารถนำมาประยุกต์ใช้กับระบบงานภายใน หรือภายนอกองค์กรบนแพลตฟอร์มของเว็บไซต์ หรือ โปรแกรมฝังบนเว็บไซต์ที่อยู่ภายใต้โครงการ Google Labs ซึ่งบริการดังกล่าวจะเป็นบริการที่จะช่วยให้หน้าจอรายงานทุกรูปแบบเป็นไปอย่างมีประสิทธิภาพ

Google Charts เป็นเครื่องมือที่ในการสร้างแผนภูมิรูปภาพ หรือที่เราเรียกว่ากราฟ (Graphs) หรือชาร์ต (Charts) ที่เราเอาไว้นำเสนอรายงานต่างๆ โดยบริการของ Google Charts นี้สามารถเรียกใช้ในรูปแบบของ Visualization API หรือส่วนต่อประสานโปรแกรมของ Google ที่จะแปลงข้อมูลจากฐานข้อมูลสถิติต่างๆ จากเว็บไซต์ของคุณ ให้แสดงผลออกมาเป็นรูปแบบแผนภูมิที่เรียบง่าย ไปจนถึงรูปแบบที่มีลำดับชั้นของข้อมูลที่ซับซ้อน หรือมีขนาดใหญ่ โดยอาศัยหลักการเชื่อมต่อสื่อสารรับส่งข้อมูลบนสถาปัตยกรรมอินเทอร์เน็ตแบบเครื่องลูกข่ายโปรแกรมบริการ

**Display live data on your site**  
Google chart tools are powerful, simple to use, and free. Try out our rich gallery of interactive charts and data tools.

**Pie Chart - view source**

Category	Percentage
Work	45.8%
Eat	29.2%
Comute	6.3%
Watch TV	1.2%
Sleep	17.5%

**Rich Gallery**  
Choose from a variety of charts. From simple scatter plots to hierarchical treemaps, find the best fit for your data.

**Customizable**  
Make the charts your own. Configure an extensive set of options to perfectly match the look and feel of your website.

**HTML5 / SVG**  
Cross-browser compatibility (adopting VML for older IE versions) and cross-platform portability to iOS and new Android releases. No plugins are needed.

**Free**  
Use the same chart tools Google uses, completely free and with three years' backward compatibility guaranteed.

**Controls and Dashboards New!**  
Easily connect charts and controls into an interactive dashboard.

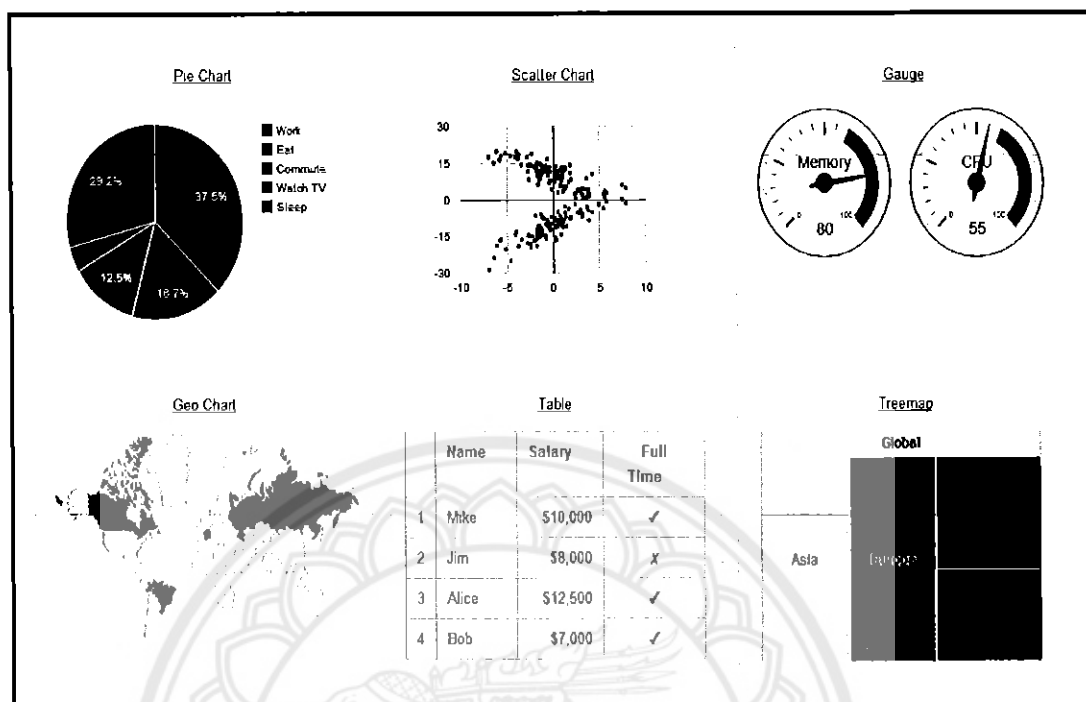
**Dynamic Data**  
Connect to your data in real time using a variety of data connection tools and protocols.

รูปที่ 2.9 เว็บไซต์แนะนำการใช้ Google Chart

ภายใต้ชุดพัฒนาของส่วนต่อประสาน โปรแกรมอย่าง Visualization API ของ Google ทำให้สามารถนำชุดข้อมูลสถิติที่อยู่ในรูปของตาราง (Spread Sheets) มาทดสอบผ่านหน้าจอทดสอบก่อนนำไปใช้งานร่วมกับบริการ Google Docs หรือประยุกต์ให้ซับซ้อนขึ้นมาหน่อยโดยการพัฒนาร่วมกับการเขียนโปรแกรมเพื่อดึงข้อมูลจากฐานข้อมูลขึ้นมาเป็นค่าตัวแปร เพื่อให้แสดงผลเป็นกราฟ หรือแผนภูมิที่สวยงามผ่านโปรแกรมท่องเว็บไซต์อย่างเว็บเบราว์เซอร์ได้ทุกประเภทที่เป็นเว็บเบราว์เซอร์มาตรฐาน

### 2.10.1 รูปแบบของบริการ Google Charts

รูปแบบแผนภูมิรูปภาพ หรือกราฟ ที่สร้างขึ้นจาก API ของ Google Charts นั้นพัฒนาจากภาษา JavaScript ทำให้สะดวกในการนำไปใช้กับเอกสารประเภท HTML หรือเอกสารสำหรับแสดงหน้าเว็บไซต์ทุกประเภท อีกทั้ง Google Charts ยังมีรูปแบบของกราฟ หลากหลายรูปแบบให้เลือกนำไปใช้ในการประกอบหน้าจอรายงานสถิติให้เหมาะสมกับชุดข้อมูล และสถานการณ์ที่ต้องนำข้อมูลไปใช้ เช่น Pie chart, Line Charts, Bars Charts และรูปแบบต่างๆ อีกมากมาย ทำให้ง่ายต่อกลุ่มผู้พัฒนาเว็บไซต์ที่สามารถนำ Scripts ในการแสดงผลของกราฟที่ Google Charts สร้างขึ้นมา นำไปใช้ร่วมกับรูปแบบดีไซน์ภายในเว็บไซต์ของตน

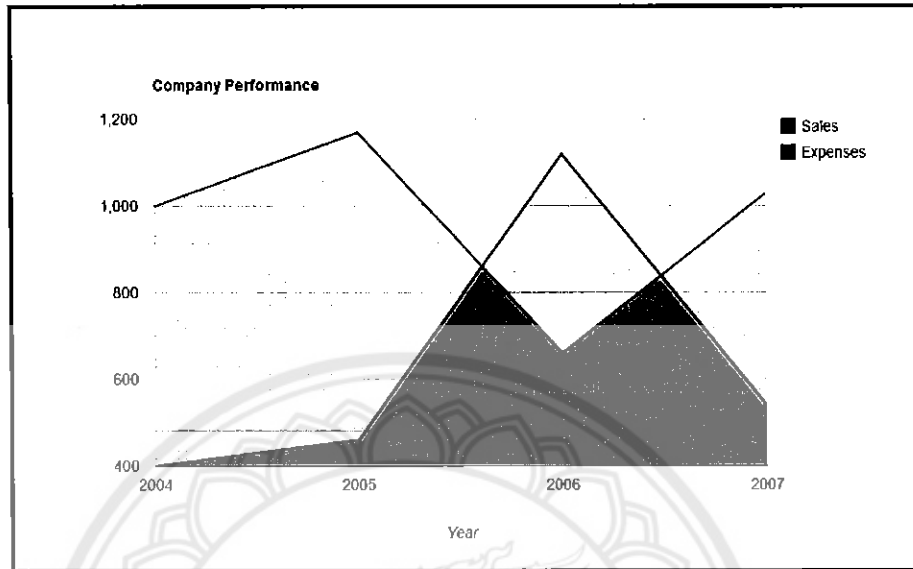


รูปที่ 2.10 แสดงตัวอย่างรูปแบบการนำเสนอของ Google Chart

รูปแบบกราฟ หรือแผนภูมิของ Google Charts สามารถเลือกรับรูปแบบให้มีกรโต้ตอบ หรือแอนิเมชันเล็กน้อย พร้อมกับระบบ Dashboard ที่ง่ายต่อการบริหารจัดการกราฟ และเก็บชุดข้อมูลกราฟที่ได้สร้างไว้บนบริการ Google Charts เพื่อความสะดวกในการกลับมาสำสกริปต์ของ Google Charts ไปใช้บนหน้าเว็บไซต์ของคุณได้ตลอดเวลา นอกจากรูปแบบการโต้ตอบ และ Dashboard ในการควบคุมแล้ว มาตรฐานของภาษาโปรแกรมบนสกริปต์ของ Google Charts ที่สร้างขึ้นนั้นยังรองรับเทคโนโลยีเปิดในอนาคตอย่าง HTML5 และยังทำงานข้ามแพลตฟอร์มไปแสดงผลบนหน้าจอสมาร์ตโฟนระบบปฏิบัติการ Andriod, ระบบปฏิบัติการ iOS ของ Apple อย่าง iPhone และ iPad โดยไม่ต้องติดตั้งส่วนเสริม หรือ Plug-in เพิ่มลงในสมาร์ตโฟนเลย

## รูปแบบกราฟของ Google Chart มี ดังนี้

### รูปแบบการนำเสนอแบบ Area Chart



รูปที่ 2.11 แสดงรูปแบบการนำเสนอแบบ Area Chart

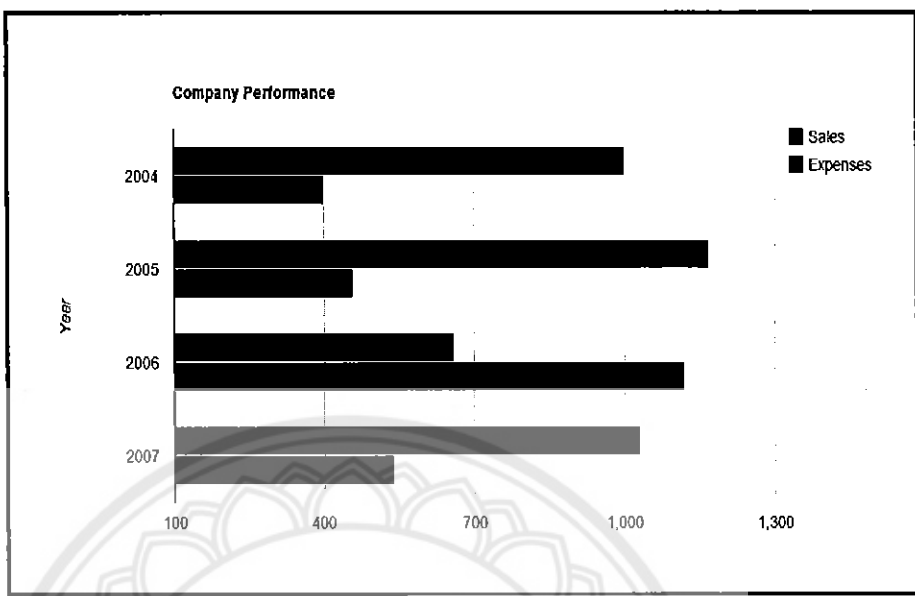
รูปแบบการนำเสนอแบบ Area Chart เป็นรูปแบบการนำเสนอแบบพื้นที่ โดยสามารถแสดงได้ในแบบ Simple คือ แสดงข้อมูลเดียว หรือ Multiple คือแสดงข้อมูลเปรียบเทียบกันหลายๆ ชนิด ซึ่งในรูป 2.11 เป็นการแสดงการนำเสนอแบบ Multiple โดยมีข้อกำหนดพื้นฐานคือ คอลัมน์แรกข้อมูลสามารถเป็นชนิดตัวอักษร, ชนิดตัวเลข, ชนิดวันที่ หรือชนิดเวลา คอลัมน์แรกนี้จะถูกนำไปใช้เป็นแกน X และคอลัมน์ต่อไปจะถูกนำไปใช้เป็นข้อมูลแกน Y โดยกำหนดให้เป็นข้อมูลชนิดตัวเลข

ตารางที่ 2.3 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Area Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	---	คอลัมน์ที่ n
ชนิดของข้อมูล	<ul style="list-style-type: none"> <li>• String</li> <li>• number</li> <li>• date</li> <li>• datetime</li> <li>• timcofday</li> </ul>	Number	---	Number



รูปแบบการนำเสนอแบบ Bar Chart



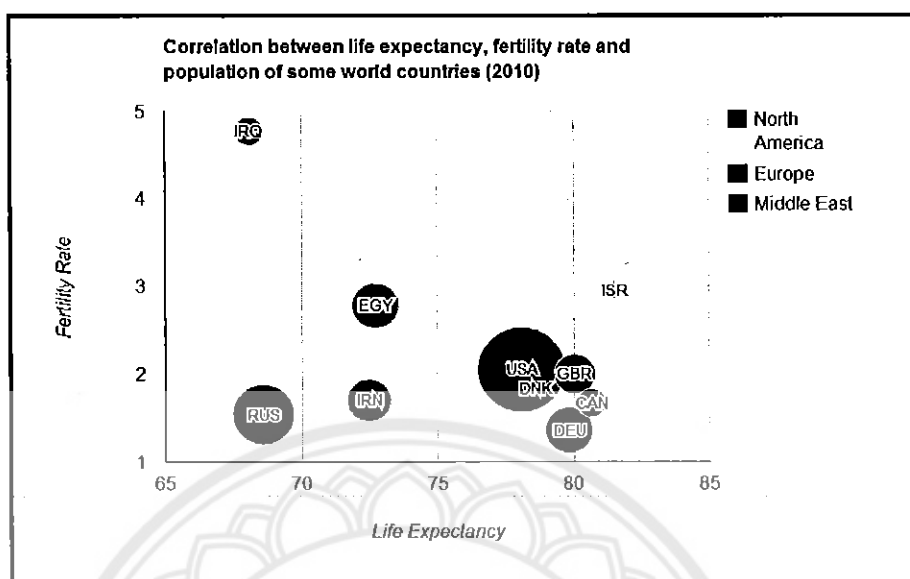
รูปที่ 2.12 แสดงรูปแบบการนำเสนอแบบ Bar Chart

จากรูปที่ 2.12 เป็นรูปแบบการนำเสนอแบบกราฟแท่ง ซึ่งรูปแบบการนำเสนอที่สามารถใช้กับข้อมูลที่ต้องการแสดงแบบ Simple หรือ Multiple ก็ได้ มีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกกำหนดให้เป็นข้อมูลชนิดตัวอักษร จะถูกนำไปใช้เป็นแกน Y และข้อมูลคอลัมน์ต่อไปกำหนดให้เป็นข้อมูลชนิดตัวเลข และถูกนำไปใช้เป็นแกน X

ตารางที่ 2.4 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Bar Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	---	คอลัมน์ที่ n
ชนิดของข้อมูล	String	Number	---	Number

## รูปแบบการนำเสนอแบบ Bubble Chart



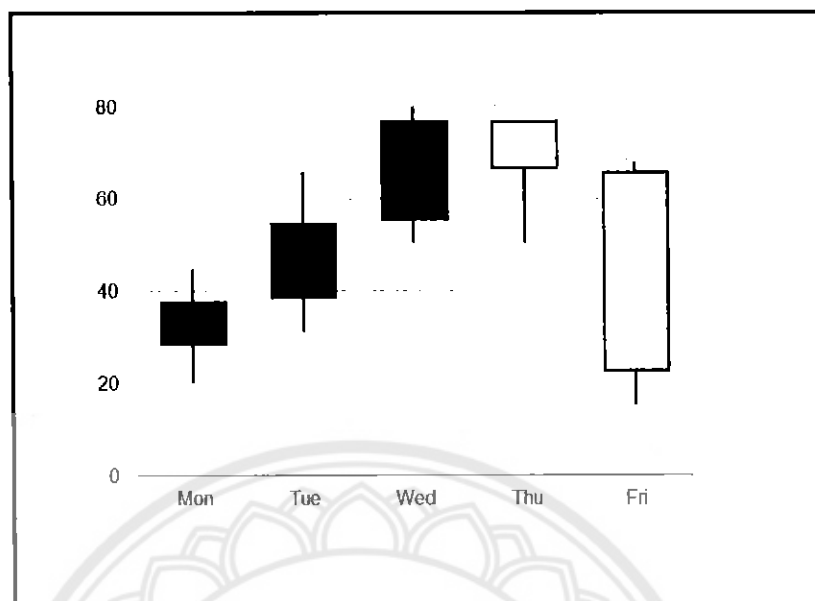
รูปที่ 2.13 แสดงรูปแบบการนำเสนอแบบ Bubble Chart

จากรูปที่ 2.13 เป็นรูปแบบการนำเสนอแบบ Bubble Chart รูปแบบการนำเสนอนี้มีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกกำหนดให้เป็นข้อมูลชนิดตัวอักษร จะถูกนำไปใช้เป็นคำอธิบายของการแสดงข้อมูล และข้อมูลคอลัมน์ที่สองและสามจะเป็นตัวเลขระบุตำแหน่งของ Bubble ซึ่งแทนด้วยแกน X และแกน Y

ตารางที่ 2.5 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Bubble Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	คอลัมน์ที่ 2	คอลัมน์ที่ 3 (optional)	คอลัมน์ที่ 4 (optional)
ชนิดของข้อมูล	String	Number	Number	string or Number	Number

## รูปแบบการนำเสนอแบบ Candlestick Chart



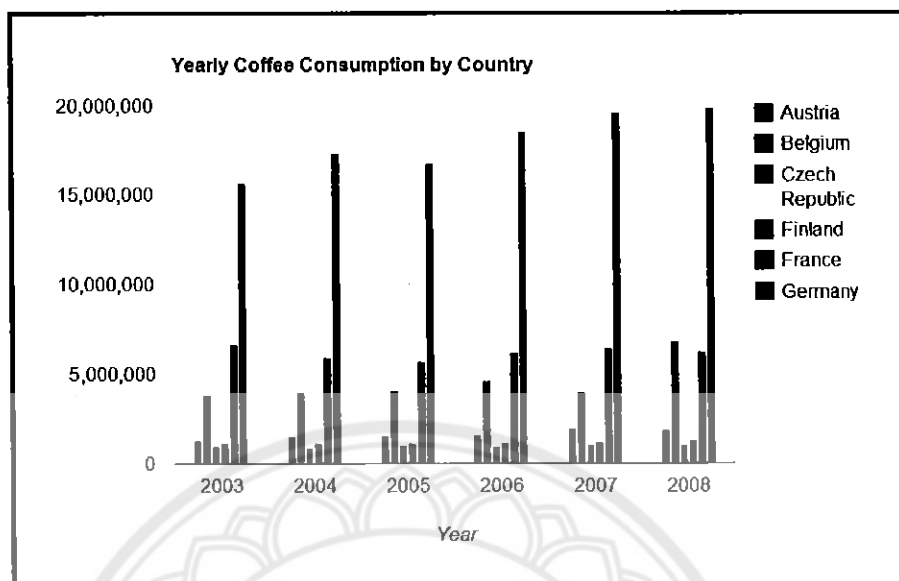
รูปที่ 2.14 แสดงรูปแบบการนำเสนอแบบ Candlestick Chart

จากรูปที่ 2.14 เป็นรูปแบบการนำเสนอแบบ Candlestick Chart รูปแบบการนำเสนอนี้มีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกกำหนดให้เป็นข้อมูลชนิดตัวอักษร จะถูกนำไปใช้เป็นคำอธิบายของการแสดงข้อมูลในแกน X และข้อมูลคอลัมน์ที่สองเป็นข้อมูลชนิดตัวเลขแสดงค่าต่ำสุด คอลัมน์ที่สามเป็นข้อมูลชนิดตัวเลขระบุค่าต่ำสุดซึ่งเป็นค่าเริ่มต้นของเครื่องหมาย คอลัมน์ที่สี่เป็นข้อมูลชนิดตัวเลขแสดงค่าสูงสุดของเครื่องหมาย และคอลัมน์ที่ห้าเป็นค่าสูงสุดที่สามารถไปได้

ตารางที่ 2.6 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Candlestick Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	คอลัมน์ที่ 2	คอลัมน์ที่ 3	คอลัมน์ที่ 4
ชนิดของข้อมูล	String	Number	Number	Number	Number

### รูปแบบการนำเสนอแบบ Column Chart



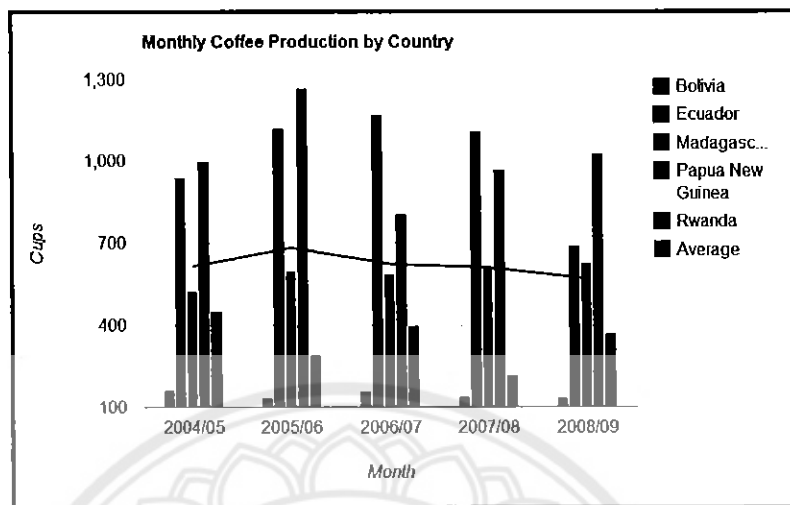
รูปที่ 2.15 แสดงรูปแบบการนำเสนอแบบ Column Chart

จากรูปที่ 2.15 เป็นรูปแบบการนำเสนอแบบ Column Chart รูปแบบการนำเสนอนี้มีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกกำหนดให้เป็นข้อมูลชนิดตัวอักษร จะถูกนำไปใช้เป็นคำอธิบายของการแสดงข้อมูลในแกน X และข้อมูลคอลัมน์ต่อไปจะเป็นข้อมูลชนิดตัวเลข เพื่อใช้ในการระบุค่าในแกน Y

ตารางที่ 2.7 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Column Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	---	คอลัมน์ที่ n
ชนิดของข้อมูล	String	Number	---	Number

## รูปแบบการนำเสนอแบบ Combo Chart



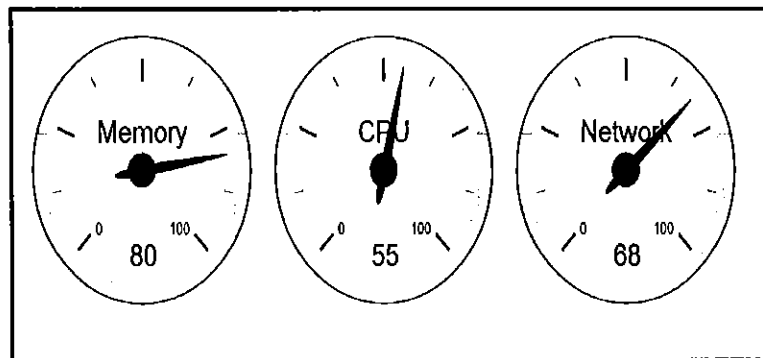
รูปที่ 2.16 แสดงรูปแบบการนำเสนอแบบ Combo Chart

จากรูปที่ 2.16 เป็นรูปแบบการนำเสนอแบบ Combo Chart รูปแบบการนำเสนอนี้มีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกกำหนดให้เป็นข้อมูลชนิดตัวอักษร จะถูกนำไปใช้เป็นตัวอธิบายของการแสดงข้อมูลในแกน X และข้อมูลคอลัมน์ต่อไปจะเป็นข้อมูลชนิดตัวเลข เพื่อใช้ในการระบุค่าในแกน Y

ตารางที่ 2.8 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Combo Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	---	คอลัมน์ที่ n
ชนิดของข้อมูล	String	Number	---	Number

## รูปแบบการนำเสนอแบบ Gauge



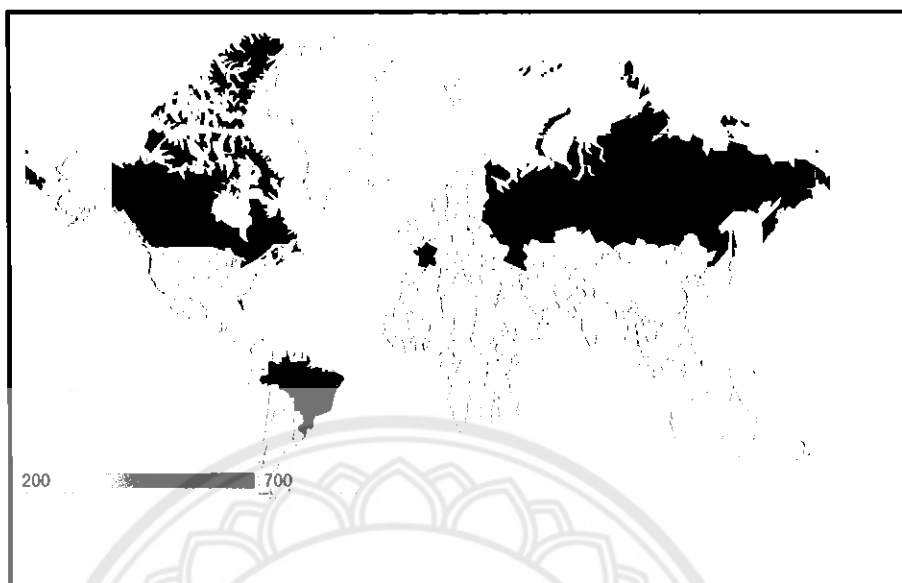
รูปที่ 2.17 แสดงรูปแบบการนำเสนอแบบ Gauge

จากรูปที่ 2.17 เป็นรูปแบบการนำเสนอแบบ Gauge หรือมาตรวัด รูปแบบการนำเสนอนี้มีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกกำหนดให้เป็นข้อมูลชนิดตัวอักษร จะถูกนำไปใช้เป็นคำอธิบายของการแสดงข้อมูลมาตรวัดและข้อมูลคอลัมน์ที่สองเป็นข้อมูลตัวเลขจะถูกนำไปแสดงเป็นค่าในมาตรวัด

ตารางที่ 2.9 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Gauge

	คอลัมน์ที่ 0	คอลัมน์ที่ 1
ชนิดของข้อมูล	String	Number

### รูปแบบการนำเสนอแบบ Geo Chart

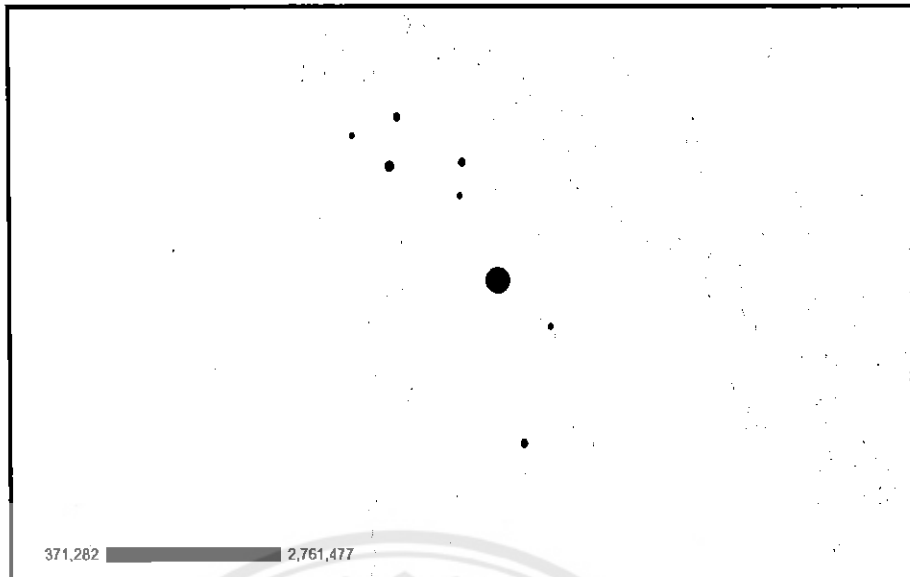


รูปที่ 2.18 แสดงรูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Regions

จากรูปที่ 2.18 เป็นรูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Regions คือแผนที่ที่แสดงเป็นภูมิภาค สามารถนำไปใช้กับการเปรียบเทียบค่าต่างๆ ในระดับประเทศ และระดับจังหวัด โดยมีข้อกำหนดพื้นฐานคือ คอลัมน์แรกจะต้องเป็นข้อมูลชนิดตัวอักษร โดยระบุชื่อประเทศ หรือจังหวัด และคอลัมน์ที่สองจะระบุข้อมูลชนิดตัวเลข

ตารางที่ 2.10 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Regions

	คอลัมน์ที่ 0	คอลัมน์ที่ 1
ชนิดของข้อมูล	country name	Number



รูปที่ 2.19 แสดงรูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Markers

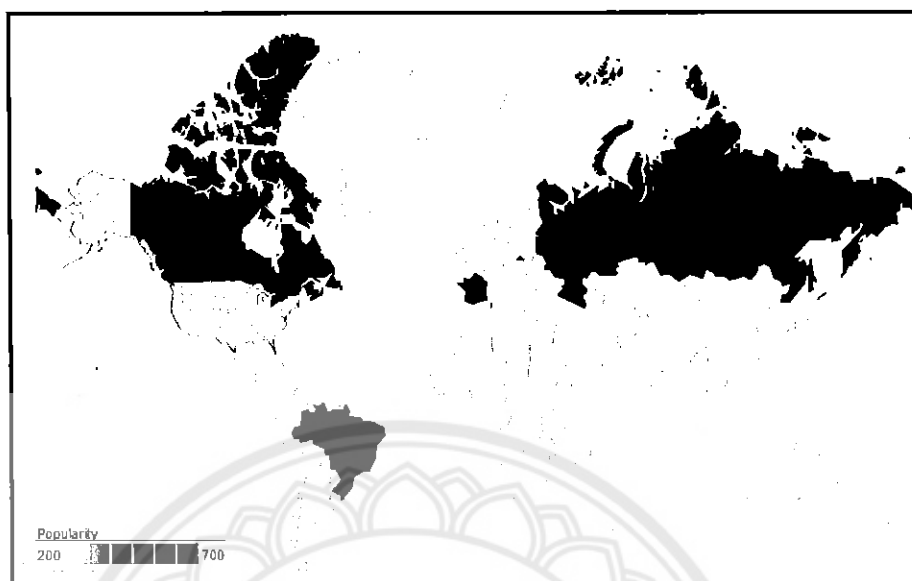
รูปที่ 2.19 เป็นรูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Markers คือเป็นการทำเครื่องหมายในพิกัดหรือตำแหน่งที่ต้องการบนแผนที่โลก ซึ่งกำหนดพื้นฐานในการใช้แผนภูมิชนิดนี้มี 2 แบบ คือ แบบแรกกำหนดให้คอลัมน์แรกให้ระบุข้อมูลเป็นชนิดตัวอักษรโดยระบุที่อยู่ของสถานที่นั้นๆ ที่ต้องการ และคอลัมน์ที่สองกำหนดให้เป็นข้อมูลชนิดตัวเลข แบบที่สองกำหนดให้คอลัมน์แรกระบุละติจูดและลองจิจูดของพื้นที่นั้นๆ และคอลัมน์ที่สองกำหนดให้ข้อมูลเป็นชนิดตัวเลข

ตารางที่ 2.11 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Geo Chart ในลักษณะ Markers

	คอลัมน์ที่ 0	คอลัมน์ที่ 1
ชนิดของข้อมูลแบบที่ 1	String address	Number
ชนิดของข้อมูลแบบที่ 2	First column is the latitude, and the second column is the longitude.	Number



## รูปแบบการนำเสนอแบบ Geo Map



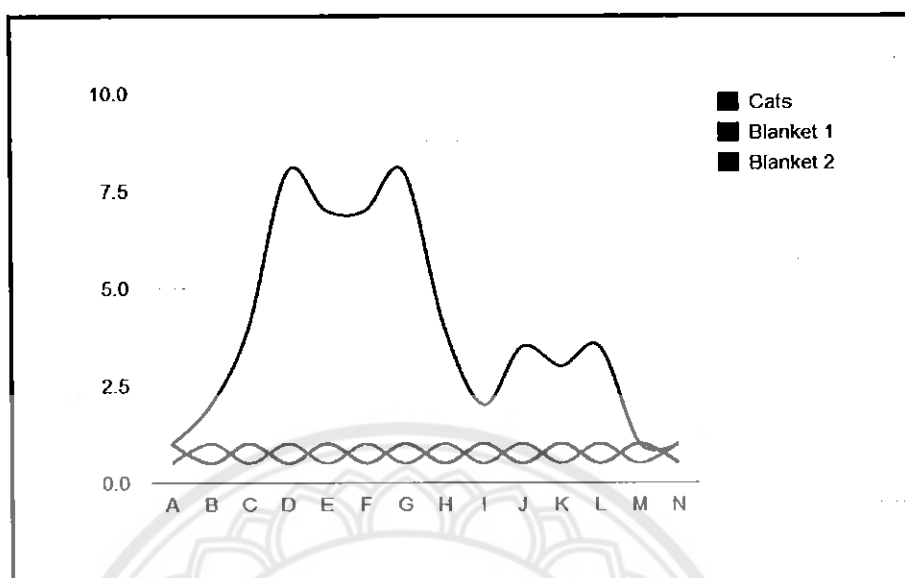
รูปที่ 2.20 แสดงรูปแบบการนำเสนอแบบ Map Chart

จากรูปที่ 2.20 เป็นรูปแบบการนำเสนอแบบ Map Chart ในลักษณะ Regions คือแผนภูมิที่แสดงเป็นภูมิภาค สามารถนำไปใช้กับการเปรียบเทียบค่าต่างๆ ในระดับประเทศ และระดับจังหวัด ซึ่งแผนภูมินี้ถูกพัฒนามาจาก Geo Chart ทำให้การแสดงผลมีความน่าสนใจมากขึ้น โดยมีข้อกำหนดพื้นฐานคือ คอลัมน์แรกจะต้องเป็นข้อมูลชนิดตัวอักษร โดยระบุชื่อประเทศ หรือจังหวัด และคอลัมน์ที่สองจะระบุข้อมูลชนิดตัวเลข

ตารางที่ 2.12 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Map Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1
ชนิดของข้อมูล	country name	Number

## รูปแบบการนำเสนอแบบ Line Chart



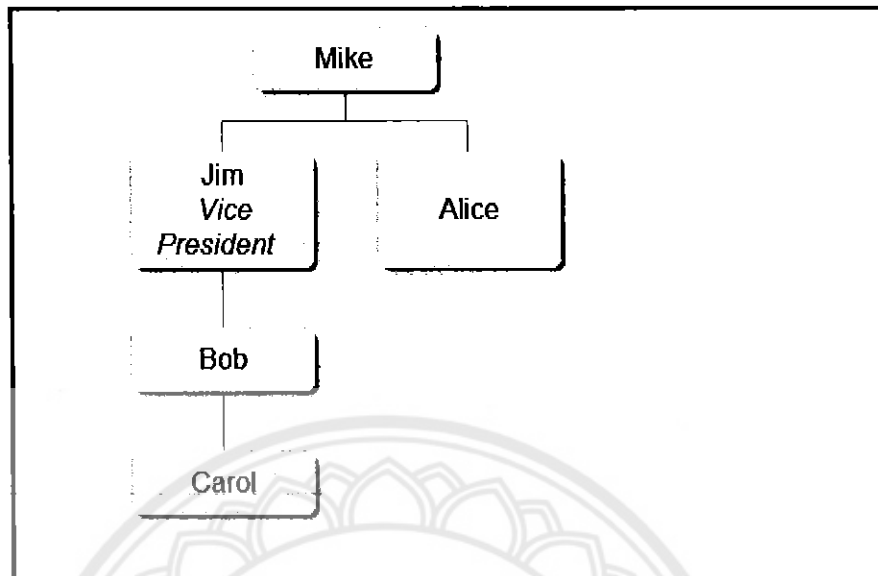
รูปที่ 2.21 แสดงรูปแบบการนำเสนอแบบ Line Chart

จากรูปที่ 2.21 เป็นรูปแบบการนำเสนอแบบ Line Chart โดยมีข้อกำหนดพื้นฐานคือ คอลัมน์แรกสามารถเป็นข้อมูลชนิดตัวอักษร ข้อมูลชนิดตัวเลข ข้อมูลวันที่ และอื่นๆ เพื่อใช้เป็นข้อมูลในแกน X และคอลัมน์อื่นๆ เป็นข้อมูลชนิดตัวเลขเพื่อระบุตำแหน่งในแกน Y

ตารางที่ 2.13 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Line Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	---	คอลัมน์ที่ n
ชนิดของข้อมูล	<ul style="list-style-type: none"> <li>● String</li> <li>● number</li> <li>● date</li> <li>● datetime</li> <li>● timeofday</li> </ul>	Number	---	Number

## รูปแบบการนำเสนอแบบ Org Chart



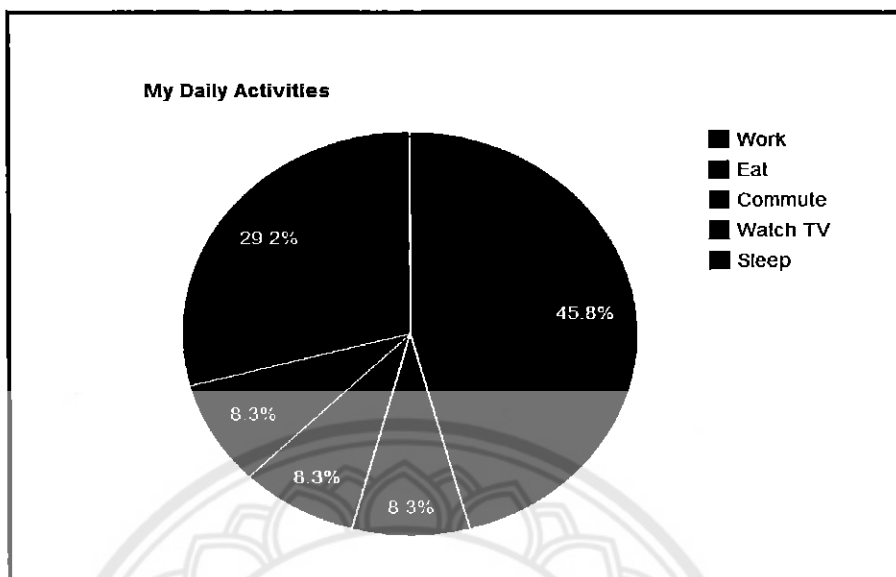
รูปที่ 2.22 แสดงรูปแบบการนำเสนอแบบ Org Chart

จากรูปที่ 2.22 เป็นรูปแบบการนำเสนอแบบ Org Chart โดยมีข้อกำหนดพื้นฐานคือ คอลัมน์แรกเป็นข้อมูลชนิดตัวอักษร คอลัมน์ที่สองเป็นข้อมูลชนิดตัวอักษรเช่นกัน และคอลัมน์ที่สามเป็นข้อมูลชนิดตัวอักษรเพื่อระบุเป็นความสัมพันธ์

ตารางที่ 2.14 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Org Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	คอลัมน์ที่ 2
ชนิดของข้อมูล	String	String	Tooltip

รูปแบบการนำเสนอแบบ Pie Chart



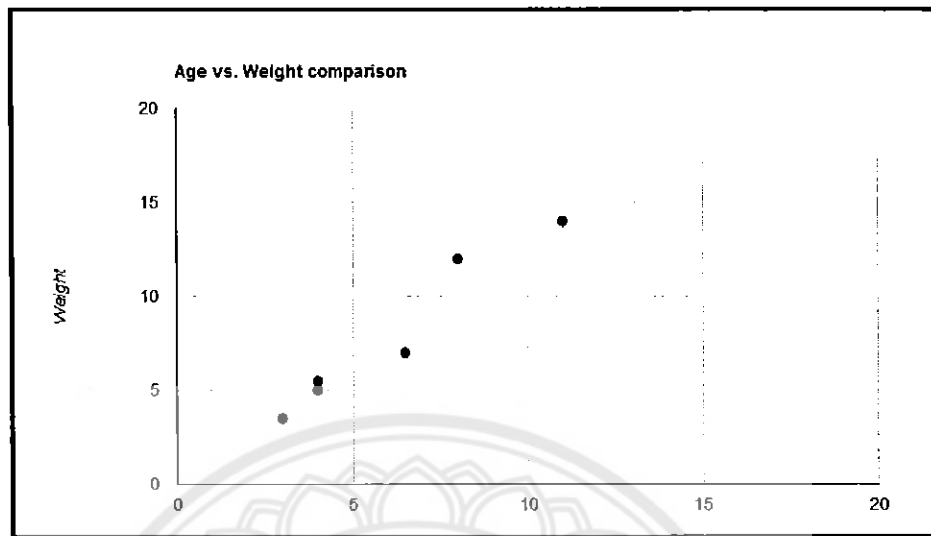
รูปที่ 2.23 แสดงรูปแบบการนำเสนอแบบ Pie Chart

จากรูปที่ 2.23 เป็นรูปแบบการนำเสนอแบบ Pie Chart หรือกราฟวงกลม โดยมีข้อกำหนดพื้นฐานคือ คอลัมน์แรกกำหนดให้เป็นชนิดตัวอักษรเพื่อใช้แสดงชื่อของข้อมูลแต่ละส่วน และคอลัมน์ที่สองเป็นข้อมูลชนิดตัวเลขเพื่อใช้ในการแบ่งสัดส่วนของข้อมูล

ตารางที่ 2.15 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Pie Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1
ชนิดของข้อมูล	String	Value

### รูปแบบการนำเสนอแบบ Scatter Chart



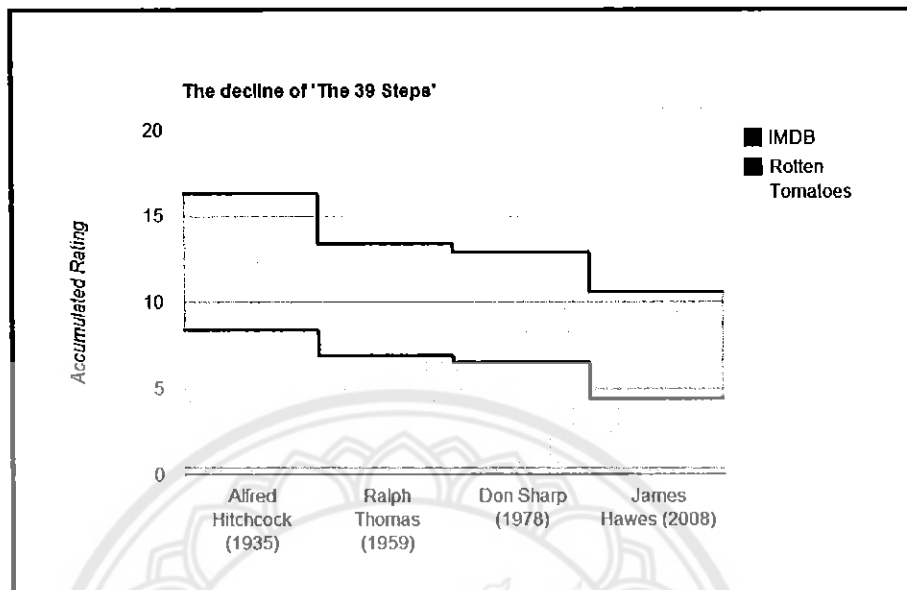
รูปที่ 2.24 แสดงรูปแบบการนำเสนอแบบ Scatter Chart

จากรูปที่ 2.24 เป็นรูปแบบการนำเสนอแบบ Scatter Chart รูปแบบการนำเสนอนี้มีข้อกำหนดพื้นฐาน กำหนดให้ทุกๆ คอลัมน์เป็นชนิดตัวเลขทั้งหมด กราฟชนิดนี้จะช่วยนำเสนอแนวโน้มของข้อมูล

ตารางที่ 2.16 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Scatter Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	---	คอลัมน์ที่ n
ชนิดของข้อมูล	Number	Number	---	Number

### รูปแบบการนำเสนอแบบ Stepped Area Chart



รูปที่ 2.25 แสดงรูปแบบการนำเสนอแบบ Stepped Area Chart

จากรูปที่ 2.25 เป็นรูปแบบการนำเสนอแบบ Stepped Area Chart รูปแบบการนำเสนอนี้มีข้อกำหนดพื้นฐานคือ คอลัมน์แรกกำหนดให้เป็นข้อมูลชนิดตัวอักษร เพื่อใช้แสดงชื่อในแกน X คอลัมน์ที่เหลือกำหนดให้เป็นข้อมูลชนิดตัวอักษร เพื่อกำหนดข้อมูลในแกน Y

ตารางที่ 2.17 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Combo Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	---	คอลัมน์ที่ n
ชนิดของข้อมูล	String	Number	---	Number

## รูปแบบการนำเสนอแบบ Table

	Name	Salary	Full Time Employee
1	Mike	\$10,000	✓
2	Jim	\$8,000	X
3	Alice	\$12,500	✓
4	Bob	\$7,000	✓

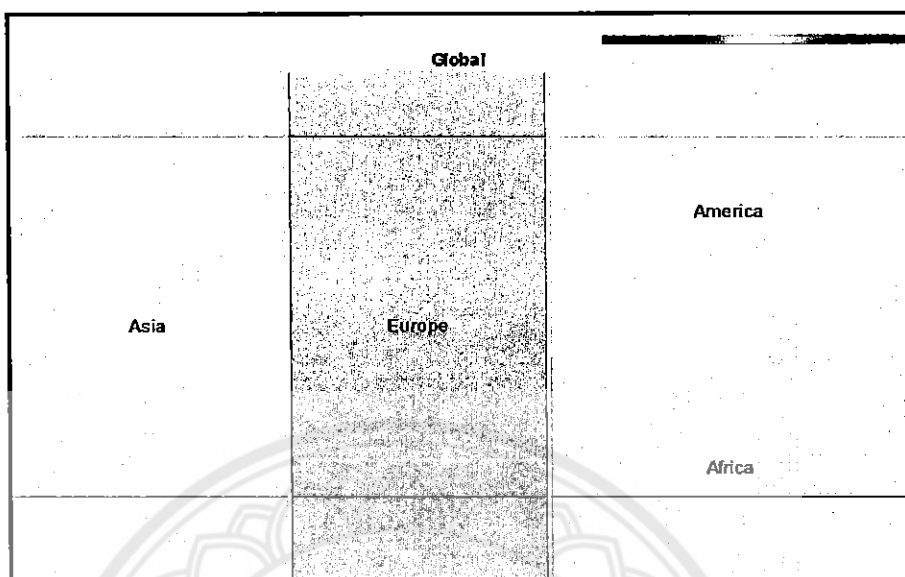
### รูปที่ 2.26 แสดงรูปแบบการนำเสนอแบบ Table

จากรูปที่ 2.26 เป็นรูปแบบการนำเสนอแบบ Table หรือตาราง ซึ่งรูปแบบการนำเสนอชนิดนี้ จะมีความยืดหยุ่นในการรับข้อมูลเป็นอย่างมาก เพราะสามารถรองรับชนิดข้อมูลได้ทุกแบบ

### ตารางที่ 2.18 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Table

	คอลัมน์ที่ 0	---	คอลัมน์ที่ n
ชนิดของข้อมูล	<ul style="list-style-type: none"> <li>• String</li> <li>• number</li> <li>• date</li> <li>• datetime</li> <li>• timeofday</li> <li>• etc</li> </ul>	---	<ul style="list-style-type: none"> <li>• String</li> <li>• number</li> <li>• date</li> <li>• datetime</li> <li>• timeofday</li> <li>• ect</li> </ul>

### รูปแบบการนำเสนอแบบ Tree Map



รูปที่ 2.27 แสดงรูปแบบการนำเสนอแบบ Tree Map

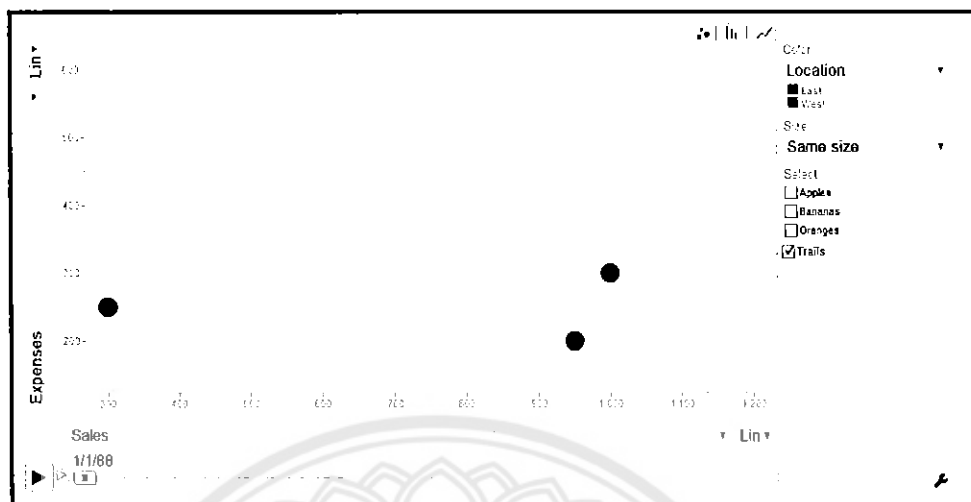
จากรูปที่ 2.27 เป็นรูปแบบการนำเสนอแบบ Tree Map โดยมีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกเป็นชนิดตัวอักษร คอลัมน์ที่สองเป็นชนิดตัวอักษร คอลัมน์ที่สามเป็นชนิดตัวเลขและ คอลัมน์ที่สี่เป็นชนิดตัวเลข ซึ่งจะระบุหรือไม่ระบุก็ได้

ตารางที่ 2.19 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Tree Map

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	คอลัมน์ที่ 2	คอลัมน์ที่ 3 (option)
ชนิดของข้อมูล	String	String	Number	Number



## รูปแบบการนำเสนอแบบ Motion Chart



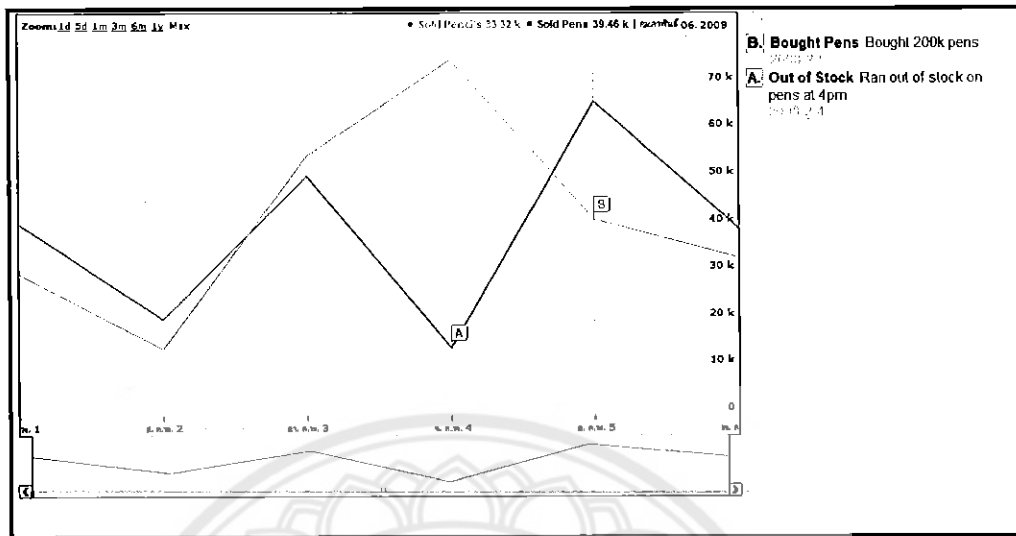
รูปที่ 2.28 แสดงรูปแบบการนำเสนอแบบ Motion Chart

จากรูปที่ 2.28 เป็นรูปแบบการนำเสนอแบบ Motion Chart รูปแบบการนำเสนอชนิดนี้ประกอบไปด้วยรูปแบบการนำเสนอ 3 แบบด้วยกันคือ Scatter Chart Column Chart และ Line Chart ซึ่งการนำเสนอแบบ Scatter Chart และ Column Chart นั้นจะสามารถแสดงแอนิเมชันการเคลื่อนที่ของข้อมูลตามช่วงเวลาได้

ตารางที่ 2.20 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Motion Chart

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	คอลัมน์ที่ 2	คอลัมน์ที่ 3	คอลัมน์ที่ 4
ชนิดของข้อมูล	String	Date	Number	Number	String

## รูปแบบการนำเสนอแบบ Annotated Time Line



รูปที่ 2.29 แสดงรูปแบบการนำเสนอแบบ Annotated Time Line

จากรูปที่ 2.29 เป็นรูปแบบการนำเสนอแบบ Annotated Time Line เป็นการนำเสนอแบบกราฟเส้น แต่มีแอนิเมชันในการเลือกดูเป็นช่วงเวลา โดยมีข้อกำหนดพื้นฐาน คือ คอลัมน์แรกเป็นข้อมูลประเภทวันที่หรือช่วงเวลา คอลัมน์คอลัมน์ที่สองและสามเป็นข้อมูลชนิดตัวเลข และคอลัมน์ที่สี่เป็นข้อมูลประเภทตัวอักษร

ตารางที่ 2.21 ชนิดข้อมูลในการใช้รูปแบบการนำเสนอแบบ Annotated Time Line

	คอลัมน์ที่ 0	คอลัมน์ที่ 1	คอลัมน์ที่ 2	คอลัมน์ที่ 3
ชนิดของข้อมูล	Date	Number	Number	String

### บทที่ 3

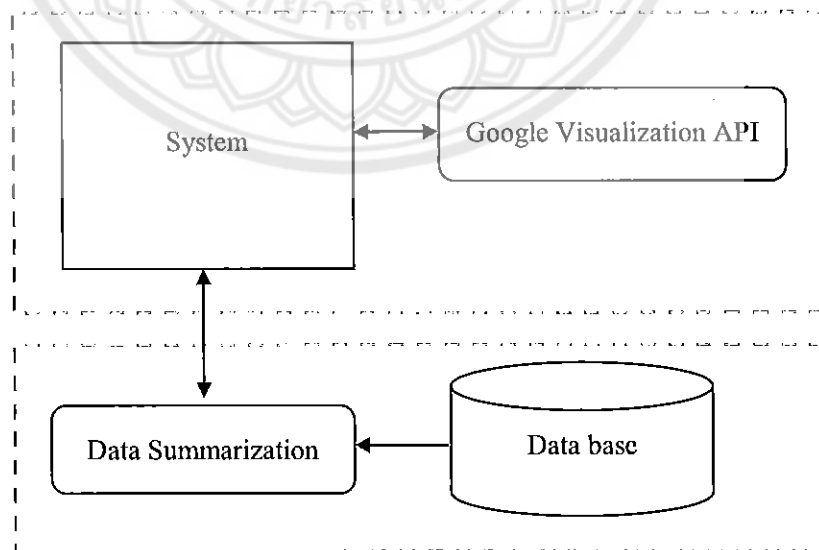
#### วิธีการดำเนินโครงการ

การจัดทำโครงการการนำเสนอข้อมูลเชิงวิเคราะห์นี้ แบ่งระบบการทำงานออกเป็น 2 ส่วน คือ ส่วนแสดงผลหรือส่วนนำเสนอและส่วนของดึงข้อมูลจากฐานข้อมูล ซึ่งส่วนของการแสดงผลนั้นผู้จัดทำต้องการใช้รูปแบบการนำเสนอจาก Google Visualization API และการดึงข้อมูลนั้นผู้จัดทำได้นำข้อมูลจากฐานข้อมูลที่จัดทำขึ้นเพื่อใช้ในการนำเสนอข้อมูลกรณีศึกษาประชากรของจังหวัดแม่ฮ่องสอนและงบประมาณการเงินขององค์การบริหารส่วนตำบลแม่มาเต็ง อำเภอป่าซาง จังหวัดแม่ฮ่องสอน ซึ่งมีขั้นตอนการทำงานดังนี้

1. ออกแบบฐานข้อมูล และจัดทำฐานข้อมูล
2. การดึงข้อมูลจากข้อมูลฐานข้อมูล
3. การแสดงผลข้อมูลด้วยรูปแบบการนำเสนอจาก Google Visualization API

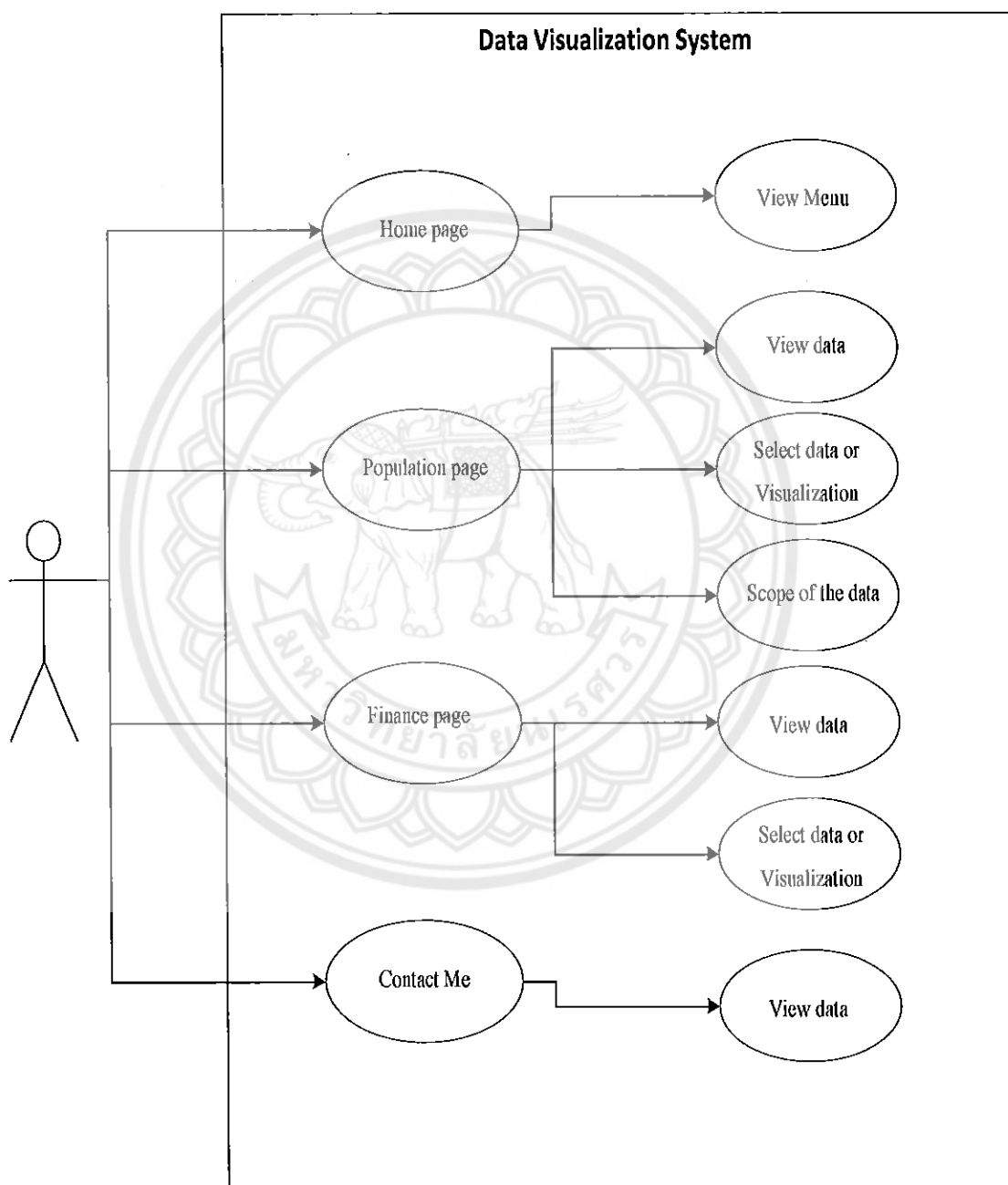
#### 3.1 หลักการทำงาน

โปรแกรมการนำเสนอข้อมูลเชิงวิเคราะห์นี้ เป็นการนำเอาข้อมูลที่ถูกจัดเก็บไว้ในฐานข้อมูล มาสรุปและนำเสนอเฉพาะข้อมูลที่ต้องการ โดยการแสดงผลนั้น จะทำการแสดงผลบนเว็บแอปพลิเคชันในรูปแบบการนำเสนอที่หลากหลาย โดยรูปแบบการนำเสนอที่ผู้จัดทำได้นำรูปแบบการนำเสนอของ Google Visualization API มาใช้ในการนำเสนอข้อมูล



รูปที่ 3.1 หลักการทำงานของระบบการนำเสนอข้อมูลเชิงวิเคราะห์

จากรูปที่ 3.1 แสดงหลักการทำงานของระบบการนำเสนอข้อมูลเชิงวิเคราะห์ ซึ่งระบบจะทำหน้าที่ดึงข้อมูลที่สรุปจากฐานข้อมูลไปแสดงบนเว็บแอปพลิเคชัน และรูปแบบการนำเสนอที่แสดงจะดึงรูปแบบการนำเสนอจาก Google Visualization API มาแสดงกับข้อมูลที่มีอยู่ ซึ่งผู้ใช้สามารถเลือกรูปแบบในการนำเสนอได้ในรูปแบบที่ผู้จัดทำได้ทำการกำหนดไว้ได้



รูปที่ 3.2 แสดงความสามารถของระบบในรูปแบบ Use case diagram

รูปที่ 3.2 เป็นการออกแบบการทำงานของโปรแกรมในรูปแบบของ Use case diagram โดยโปรแกรมจะแบ่งเป็น 4 ส่วน คือ

**Home page** เมื่อทำการคอมไพล์โปรแกรมขึ้นมา จะแสดงหน้า Home เป็นหน้าแรก โดยหน้า Home นี้จะประกอบด้วยเมนู ที่ทำหน้าที่ลิงค์ไปยัง หน้าหลักอื่นๆ

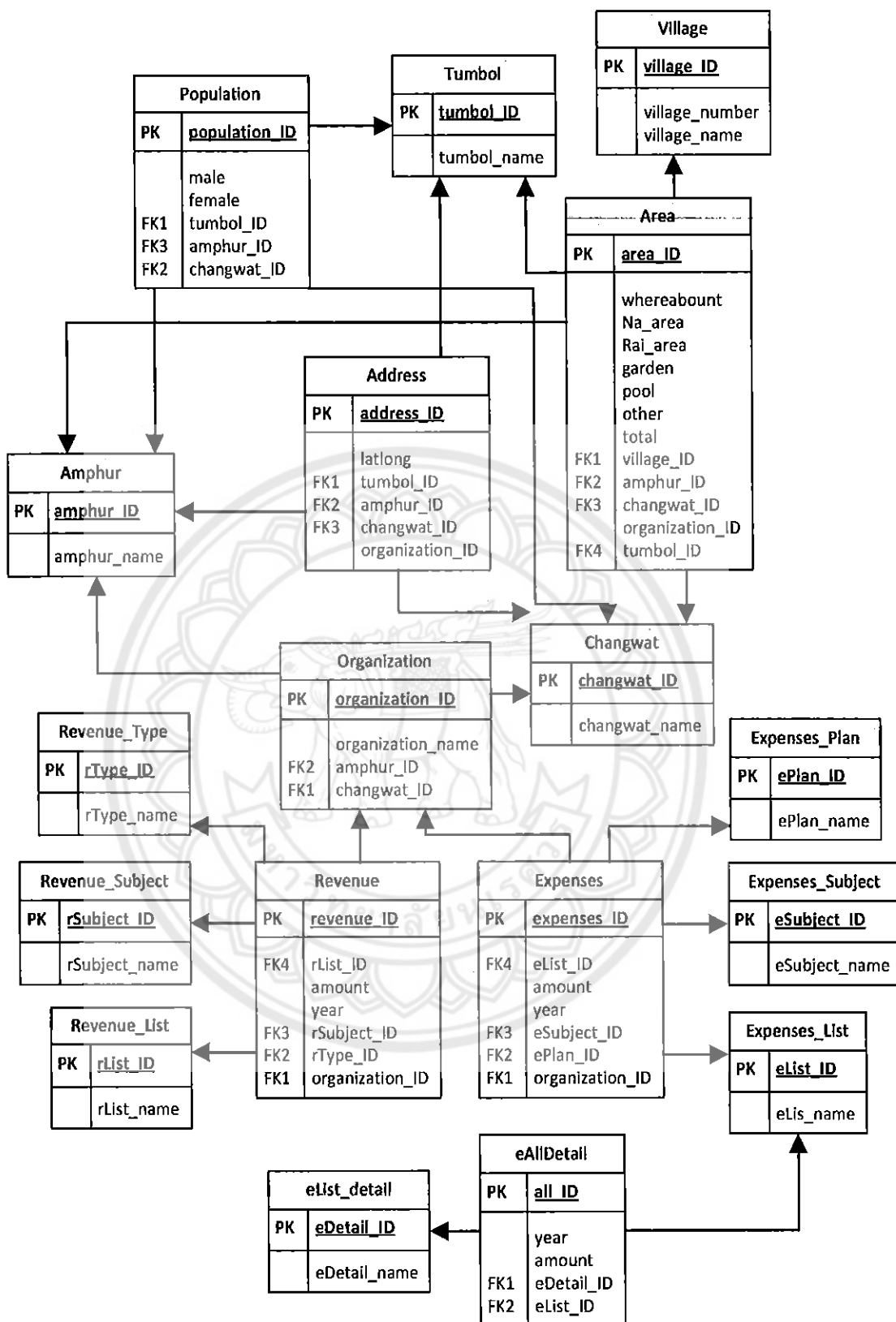
**Population page** ทำหน้าที่แสดงข้อมูลประชากรจังหวัดแม่ฮ่องสอน สามารถข้อมูลประชากรของอำเภอที่ต้องการได้ สามารถกำหนดข้อมูลเริ่มต้นและข้อมูลสูงสุดตามที่ต้องการได้ และสามารถเลือกปรับเปลี่ยนรูปแบบการนำเสนอได้

**Finance page** ทำหน้าที่แสดงข้อมูลเกี่ยวกับงบประมาณการเงิน ตั้งแต่ปี 2552-2554 ขององค์การบริหารส่วนตำบลแม่มาตัง อำเภอปาย จังหวัดแม่ฮ่องสอน โดยออกแบบให้สามารถเลือกข้อมูลที่ต้องการได้ และสามารถเลือกรูปแบบการนำเสนอได้

**Contact me page** ทำหน้าที่แสดงข้อมูลเกี่ยวกับผู้จัดทำ

### 3.2 การออกแบบฐานข้อมูลและจัดทำฐานข้อมูล

การออกแบบและจัดทำฐานข้อมูล ผู้จัดทำใช้โปรแกรม SQL Server Management Studio ในการจัดทำฐานข้อมูลซึ่งข้อมูลที่นำมาศึกษาและจัดทำระบบการนำเสนอข้อมูลเชิงวิเคราะห์นั้น เป็นข้อมูลเกี่ยวกับประชากรของจังหวัดแม่ฮ่องสอนในแต่ละตำบล และข้อมูลเกี่ยวกับงบประมาณการเงินขององค์การบริหารส่วนตำบลแม่มาตัง อำเภอปาย จังหวัดแม่ฮ่องสอน ตั้งแต่ปีงบประมาณ 2552 -2554 โดยมีการออกแบบการจัดเก็บข้อมูลดังนี้



รูปที่ 3.3 แสดงการออกแบบฐานข้อมูลแบบ Conceptual diagram

จากรูปที่ 3. 3 เป็นการแสดงการออกแบบฐานข้อมูลแบบ Conceptual diagram ซึ่งการออกแบบฐานข้อมูลนี้จะประกอบด้วยตารางทั้งหมด 18 ตาราง โดยแต่ละตารางมีหน้าที่ดังนี้ คือ

1. ตาราง Village ออกแบบไว้เพื่อใช้จัดเก็บข้อมูลชื่อหมู่บ้านและเลขที่ของหมู่บ้าน
2. ตาราง Tambol ออกแบบไว้เพื่อใช้จัดเก็บรายชื่อของตำบล
3. ตาราง Amphur ออกแบบไว้เพื่อใช้จัดเก็บรายชื่อของอำเภอ
4. ตาราง Changwat ออกแบบไว้เพื่อใช้จัดเก็บรายชื่อของจังหวัดต่าง
5. ตาราง Population ออกแบบไว้เพื่อใช้จัดเก็บจำนวนประชากรในแต่ละตำบล โดยจะระบุอำเภอและจังหวัด เพื่อให้ทราบว่าตำบลนั้นๆ อยู่ในอำเภอและจังหวัดใด
6. ตาราง Address ใช้เก็บตำแหน่งในรูปแบบละติจูดลองจิจูดและของตำบลต่างๆ
7. ตาราง Area ออกแบบไว้เพื่อใช้จัดเก็บการใช้พื้นที่ของแต่ละตำบล
8. ตาราง Organization ออกแบบเพื่อใช้ในการจัดเก็บรายชื่อองค์การบริหารส่วนตำบล
9. ตาราง Revenue\_Type ใช้จัดเก็บรายชื่อประเภทของรายได้
10. ตาราง Revenue\_Subject ใช้จัดเก็บรายชื่อหมวดหมู่ของรายได้
11. ตาราง Revenue\_List ใช้จัดเก็บรายชื่อรายการของรายได้
12. ตาราง Revenue ใช้จัดเก็บจำนวนเงินรายได้ในแต่ละรายการ
13. ตาราง Expenses\_Plan ใช้จัดเก็บรายชื่อแผนงานของรายจ่าย
14. ตาราง Expenses\_Subject ใช้จัดเก็บรายชื่อประเภทงานของรายจ่าย
15. ตาราง Expenses\_List ใช้จัดเก็บรายชื่อรายการของรายจ่าย
16. ตาราง Expenses ใช้จัดเก็บจำนวนเงินรายจ่ายในแต่ละรายการ
17. ตาราง eList\_detail ใช้จัดเก็บรายชื่อรายละเอียดของแต่ละรายการรายจ่าย
18. ตาราง eAllDetail ใช้จัดเก็บจำนวนเงินของรายละเอียดรายการรายจ่าย

### 3.3 การดึงข้อมูลจากฐานข้อมูล

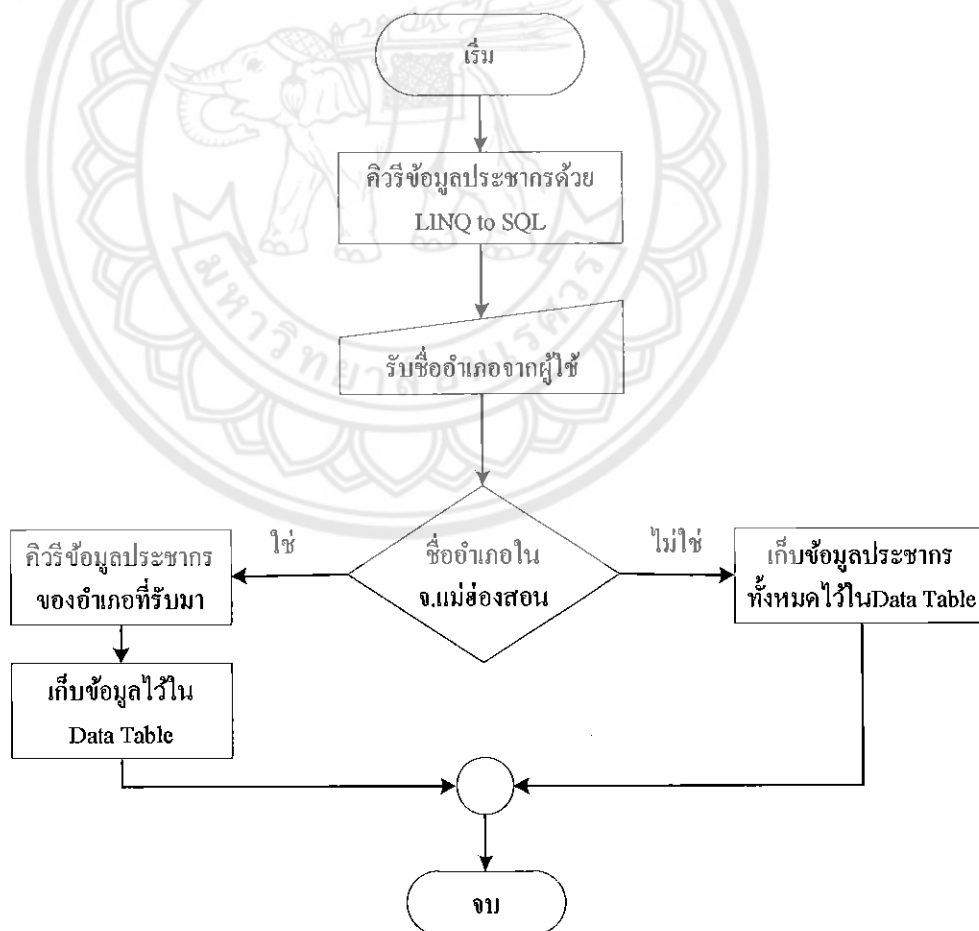
การดึงข้อมูลจากฐานข้อมูล เพื่อนำมาใช้เป็นข้อมูลในการนำเสนอข้อมูลด้วยรูปแบบการนำเสนอจาก Google Visualization API นั้น ผู้จัดทำได้ออกแบบวิธีการเตรียมข้อมูลสำหรับนำเสนอ 2 แบบด้วยกัน คือ การสรุปข้อมูลด้วยการคิวรีข้อมูลจาก LINQ to SQL และการสรุปข้อมูลด้วยการนำข้อมูลที่คิวรีมาเข้าสู่กระบวนการ Pivot

#### 3.3.1 ข้อมูลประชากรจังหวัดแม่ฮ่องสอน

ข้อมูลของประชากรจังหวัดแม่ฮ่องสอนที่ถูกจัดเก็บไว้ในฐานข้อมูลประกอบ จำนวนประชากรชาย จำนวนประชากรหญิงและจำนวนประชากรทั้งหมดของแต่ละตำบล ดังนั้นจึงแบ่งข้อมูลเป็น 3 ส่วนคือ

##### 3.3.1.1 การนำเสนอจำนวนประชากรของตำบลในแต่ละอำเภอ

การสรุปข้อมูลเพื่อนำเสนอจำนวนประชากรของตำบลในแต่ละอำเภอนั้นมีขั้นตอนการทำงานดังนี้ คือ



รูปที่ 3.4 ขั้นตอนการทำงานการนำเสนอข้อมูลของประชากรในตำบลของแต่ละอำเภอ



จากรูปที่ 3.4 มีการควิรีข้อมูลจากฐานข้อมูลด้วยวิธีการ LINQ TO SQL โดยในควิรีข้อมูลประชากรนี้ จะทำการเชื่อมตาราง 3 ตาราง ซึ่งประกอบด้วย ตาราง Population ตาราง Tumbol และ ตาราง Amphur เมื่อรับข้อมูลชื่ออำเภอจากผู้ใช้ ในที่นี้เราใช้เครื่องมือใน ASP.NET เข้ามาช่วย คือ Dropdownlist โดยให้ผู้ใช้เลือกผ่าน Dropdownlist ถ้ามีการเลือกอำเภอเกิดขึ้นจากผู้ใช้ ข้อมูลอำเภอจะถูกส่งไปควิรีจากฐานข้อมูล และแสดงจำนวนประชากรในตาราง Population ของอำเภอที่เลือกไว้ มาเก็บไว้ใน Data Table หากไม่มีการเลือกอำเภอเกิดขึ้นจะทำการควิรีข้อมูลประชากรทั้งหมดจากตาราง Population ออกมาเก็บไว้ใน Data Table

การออกแบบฟังก์ชันการเรียกดูข้อมูลประชากรของตำบลในแต่ละอำเภอนั้น เพื่อต้องการให้ผู้ใช้สามารถเลือกแสดงข้อมูลเฉพาะในอำเภอที่ต้องการได้

### 3.3.1.2 การกำหนดขอบเขตในการนำเสนอข้อมูล

การกำหนดขอบเขตของข้อมูลเป็นการกำหนดของข้อมูลจำนวนประชากรของขอบเขตล่างถึงขอบเขตบน หรือกล่าวอีกนัยหนึ่งว่า เป็นค่าเริ่มต้นและค่าสูงสุด การกำหนดขอบเขตของข้อมูลประชากรนี้ มีขั้นตอนการทำงานดังรูปที่ 3.5 ขั้นตอนแรกของการทำงาน คือ การเชื่อมตาราง Population ตาราง Tumbol และตาราง Amphur และทำการควิรีข้อมูลประชากรขึ้นมา ในส่วนของการรับตัวเลขกำหนดขอบเขตข้อมูลนั้น เราใช้เครื่องมือที่ชื่อว่า Textbox ใน ASP.NET มาใช้ โดยจะรับค่าเริ่มต้นในการกำหนดขอบเขตจาก Textbox ช่องแรก และรับค่าสูงสุดจาก Textbox ช่องที่สอง

การรับค่าตัวแปรใน Textbox จะเกิดขึ้นจะเกิดขึ้น 4 กรณีด้วยกัน คือ

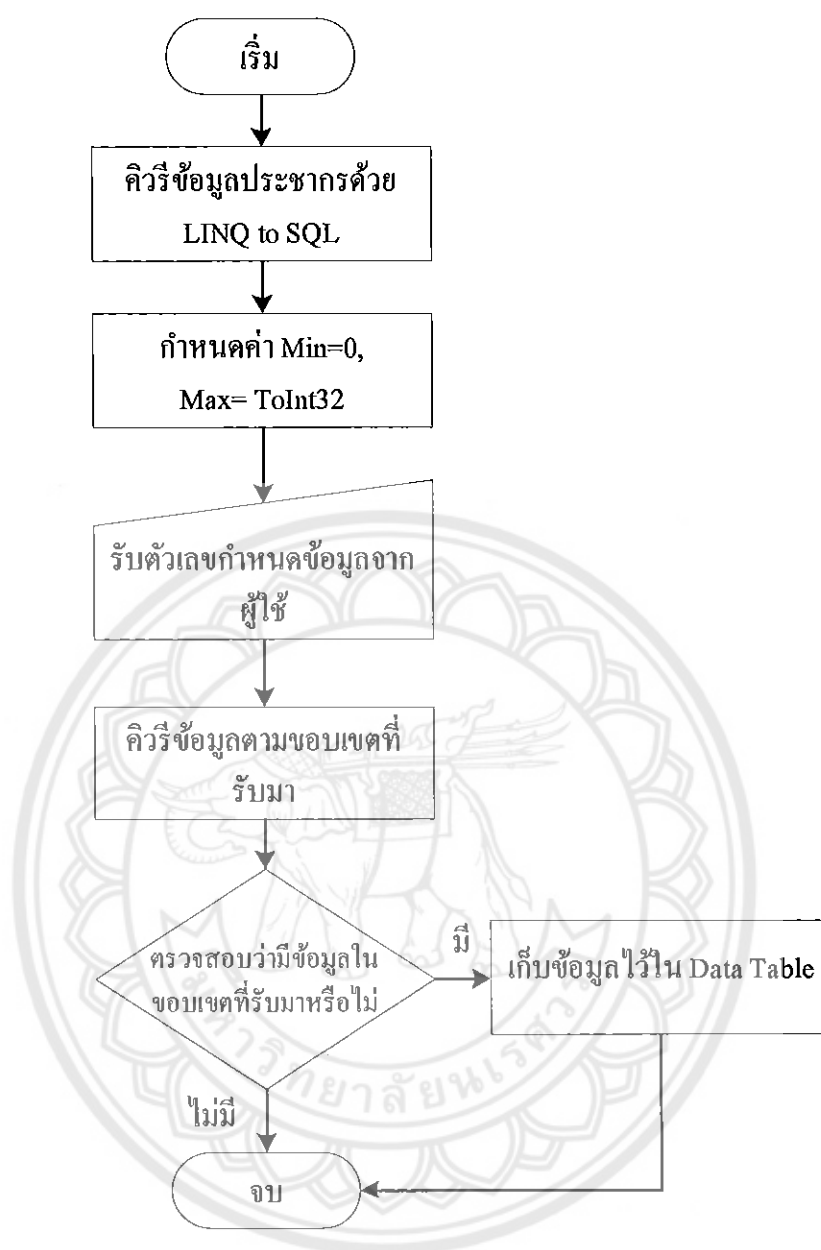
กรณีที่ 1 ไม่มีการกำหนดค่าเริ่มต้นและค่าสูงสุด

กรณีที่ 2 มีการกำหนดค่าเริ่มต้นแต่ไม่มีการกำหนดค่าต่ำสุด

กรณีที่ 3 มีการกำหนดค่าสูงสุดแต่ไม่มีการกำหนดค่าต่ำสุด

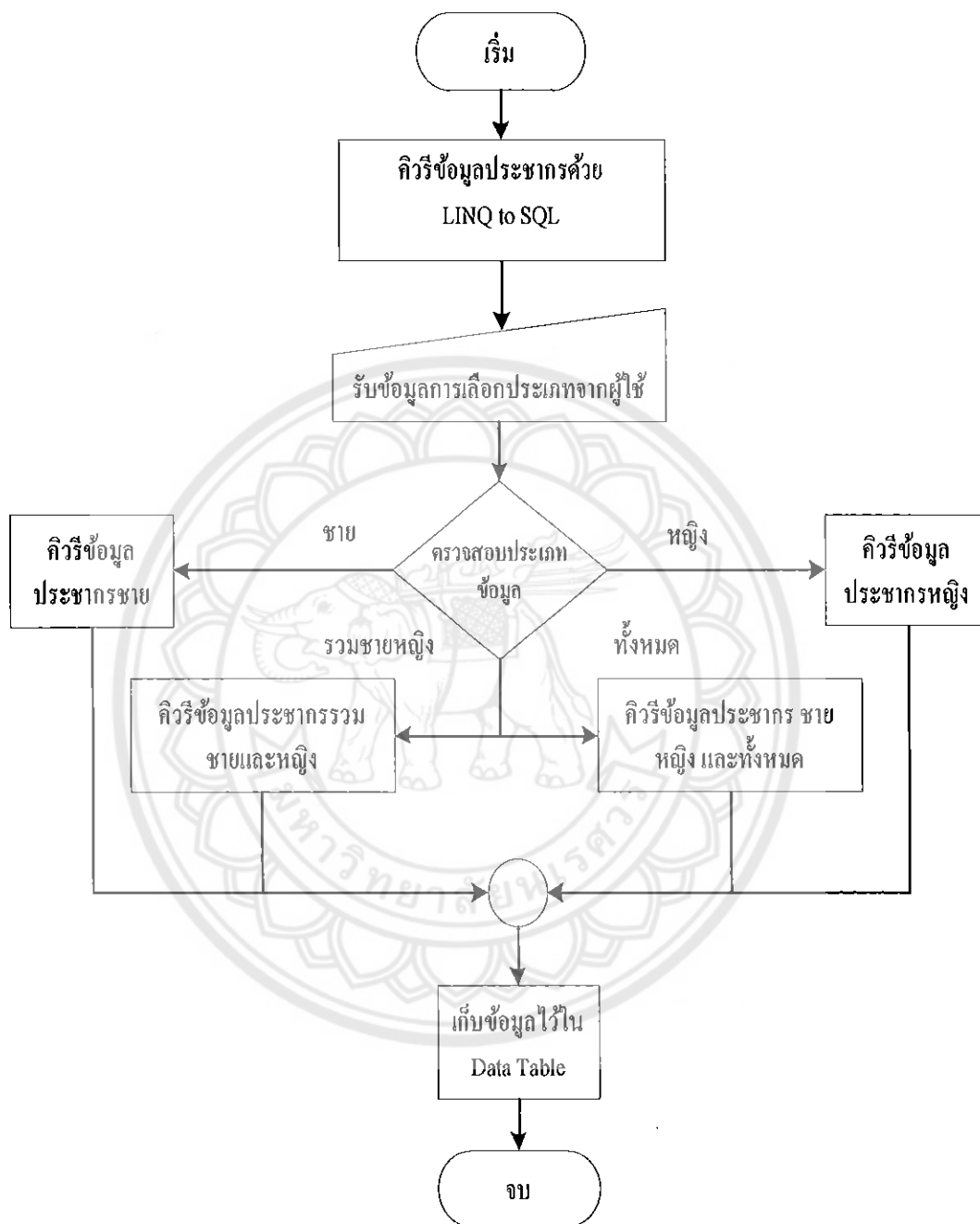
กรณีที่ 4 มีการกำหนดค่าเริ่มต้นและค่าสูงสุด

จากทั้ง 4 กรณีของการสรุปข้อมูลเพื่อนำมาใช้ในการกำหนดขอบเขตข้อมูล จะเห็นได้ว่า กรณีที่ 4 เป็นกรณีเดียวที่ผู้ใช้กรอกข้อมูลทั้งค่าเริ่มต้นและค่าสูงสุด แต่ในกรณีที่ 1, 2 และ 3 นั้น ผู้ใช้ไม่ได้กรอกข้อมูลลงไปให้ครบทุกข้อมูล หรือกรอกข้อมูลเพียงบางข้อมูลเท่านั้น ดังนั้นจึงมีการกำหนดค่าเริ่มต้นให้กับการค้นหาข้อมูลของโปรแกรม คือค่าเริ่มต้น ใช้ตัวแปร min และค่าสูงสุด ใช้ตัวแปร max โดยกำหนดให้ min มีค่าเท่ากับ 0 และ max มีเท่ากับ int.MaxValue (ToInt32) หรือมีค่าเท่ากับ  $2^{32}$  ให้กับกรณีที่ 1 2 และ 3



รูปที่ 3.5 ขั้นตอนการทำงานของ การสรุปข้อมูลเพื่อกำหนดขอบเขต

### 3.3.1.3 ขั้นตอนการแบ่งประเภทของข้อมูล



รูปที่ 3.6 ขั้นตอนการทำงานของการแบ่งประเภทของข้อมูล

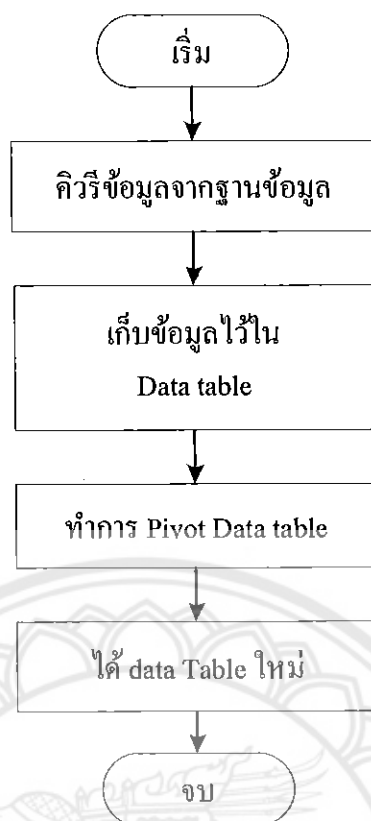
จากรูปที่ 3.6 ขั้นตอนการทำงานของ การสรุปข้อมูลเพื่อแบ่งประเภทข้อมูลนั้น จะมีการคิวรีข้อมูลเกี่ยวกับประชากร มาเก็บไว้ในตัวแปร เมื่อผู้ใช้ทำการเลือกประเภทข้อมูล ระบบจะทำการคิวรีข้อมูลให้ตามเงื่อนไขที่รับเข้ามา โดยการคิวรีนี้ยังอยู่ภายใต้เงื่อนไขการกำหนดขอบเขตของข้อมูลใน 4 กรณีที่กล่าวไปแล้วในหัวข้อของการสรุปข้อมูลเพื่อกำหนดขอบเขต ดังนั้นข้อมูลที่ได้จากการสรุปข้อมูลเพื่อแบ่งประเภทข้อมูลนี้จึงเป็นข้อมูลที่ผ่านการสรุปข้อมูลทั้งแบบกำหนดขอบเขตของข้อมูลและการแบ่งประเภทของข้อมูลด้วย

### 3.3.2 การสรุปข้อมูลการเงินขององค์การบริหารส่วนตำบลแม่नाเต็ง

การสรุปข้อมูลการเงินขององค์การบริหารส่วนตำบลแม่नाเต็ง อำเภอป่า จังหัดแม่ฮ่องสอนนั้น เป็นการนำเสนอข้อมูลเกี่ยวกับงบประมาณการเงินของปีงบประมาณ 2552, งบประมาณปี 2553 และงบประมาณปี 2554 ทั้งรายจ่ายและรายรับมานำเสนอข้อมูล ผู้จัดทำได้ออกแบบข้อมูลสำหรับการนำเสนอไว้ดังนี้

#### 3.3.2.1 การสรุปข้อมูลเกี่ยวกับรายรับ

จากลักษณะข้อมูลรายรับขององค์การบริหารส่วนตำบลแม่नाเต็ง จะมีการเก็บข้อมูลรายรับแบ่งเป็น 2 ประเภท คือ รายรับจากภาษีและรายรับที่ไม่ใช่ภาษี และประเภทของรายรับแต่ละรายการก็แบ่งเป็นหมวดหมู่รายรับ และหมวดหมู่รายรับก็มีรายการของรายรับ ซึ่งหลักการทำงานของ การสรุปข้อมูลรายรับขององค์การบริหารส่วนตำบลแม่नाเต็ง จะทำการคิวรีข้อมูลที่ต้องการจากฐานข้อมูลมาเก็บไว้ใน data table เมื่อได้ data table แล้ว นำ data table นั้นมาทำการ Pivot เพื่อให้ได้รูปแบบข้อมูลที่ต้องการและเหมาะกับการนำเสนอมากขึ้น และเก็บข้อมูลที่ได้ออกมาไว้ใน data table ใหม่ ดังรูปที่ 3.7



รูปที่ 3.7 หลักการทำงานการสรุปเกี่ยวกับข้อมูลรายรับรายจ่าย

ดังนั้นจึงทำการออกแบบการสรุปข้อมูลเกี่ยวกับรายรับได้ดังนี้

- รายรับปีงบประมาณ 2552 2553 และ 2554
- รายรับจำแนกตามประเภท
- รายรับจำแนกตามหมวดหมู่
- รายรับจำแนกตามรายการ
- สรุปรายการของรายรับ

### 3.3.2.2 การสรุปข้อมูลเกี่ยวกับรายจ่าย

จากลักษณะข้อมูลรายจ่ายขององค์การบริหารส่วนตำบลแม่नाเต็ง จะมีการเก็บข้อมูลรายจ่ายการใช้งบประมาณเป็น 11 แผนงาน และแผนงานของรายจ่ายแต่ละรายการก็สามารถแบ่งเป็นประเภทงานของรายจ่ายได้ และประเภทงานของรายจ่ายก็มีรายการของรายจ่าย ดังนั้นจึงทำการสรุปข้อมูลเกี่ยวกับรายรับได้ดังนี้

- รายจ่ายปีงบประมาณ 2552 2553 และ 2554
- รายจ่ายจำแนกตามแผนงาน
- รายจ่ายจำแนกตามประเภทงาน

- รายจ่ายจำแนกตามรายการ
- สรุปรายการรายจ่าย
- แสดงรายละเอียดรายการรายจ่าย

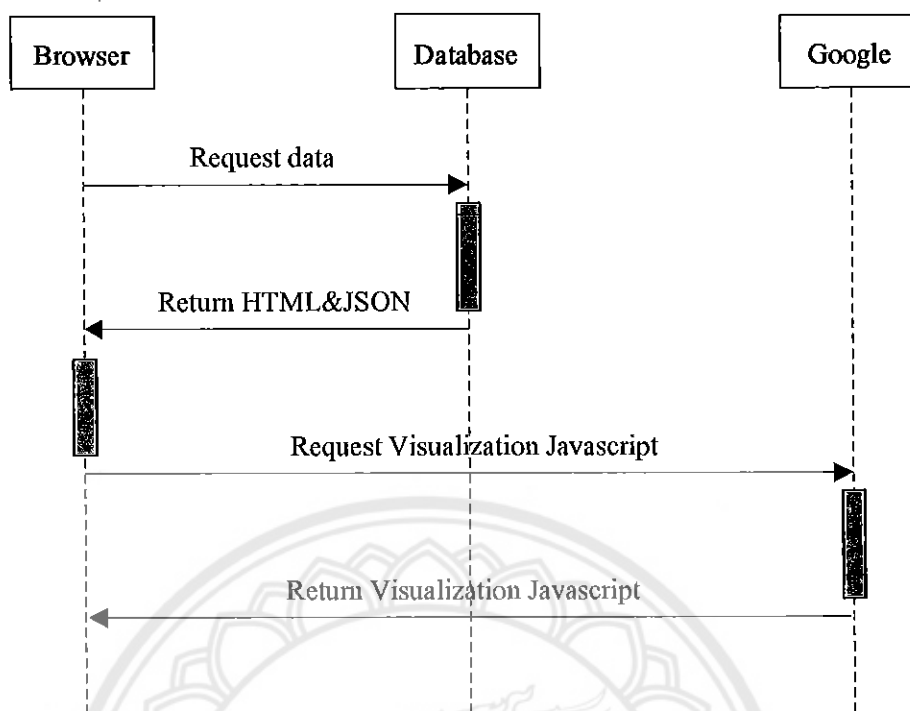
### 3.4 การนำเสนอข้อมูล

การนำเสนอข้อมูลด้วยรูปแบบการนำเสนอของ Google Visualization API นั้น มีหลักการทำงานคือ สามารถรับข้อมูลจากแหล่งเก็บข้อมูลใดๆ ก็ได้ ไม่ว่าจะเป็นฐานข้อมูล โปรแกรมแผ่นตารางทำการ (spread sheet) หรือแหล่งจัดเก็บข้อมูลอื่นๆ โดยมีข้อกำหนดให้นำข้อมูลที่ต้องการไปจัดเก็บในรูปแบบของ data table เพื่อนำไปใช้เป็นข้อมูลในการสร้างรูปแบบการนำเสนอให้กับข้อมูล



รูปที่ 3.8 แสดงรูปแบบการจัดเก็บข้อมูลของ Google Visualization API

จากรูปที่ 3.8 จะเห็นได้ว่า การนำเสนอข้อมูลด้วยรูปแบบการนำเสนอของ Google Visualization API นั้น มีความยืดหยุ่นในการรับข้อมูลจากแหล่งข้อมูลต่างๆ โดยมีเงื่อนไขการแสดงผลข้อมูล คือ ข้อมูลที่จะนำไปทำการนำเสนอจะต้องอยู่ในรูปแบบของ data table เพื่อให้สามารถนำข้อมูลที่จัดเก็บใน data table นั้นสามารถนำไปสร้างการนำเสนอเป็นกราฟรูปแบบต่างๆ ได้



รูปที่ 3.9 แสดงระบบการติดต่อการใช้รูปแบบการนำเสนอของ Google Visualization API

จากรูปที่ 3.9 แสดงการติดต่อของระบบ โดยขั้นตอนแรกผู้ใช้จะทำการเปิดเบราว์เซอร์ หลังจากนั้นเบราว์เซอร์จะทำการร้องขอข้อมูลจากฐานข้อมูลเพื่อนำข้อมูลมาแสดง และเบราว์เซอร์จะได้รับข้อมูลในรูปแบบของเจสันและ HTML กลับมา ต่อมาเบราว์เซอร์ก็จะร้องขอรูปแบบการนำเสนอข้อมูลกับ Google และ Google จะทำการส่งรูปแบบการนำเสนอในรูปแบบของ Visualization Javascript กลับมายังเบราว์เซอร์ และเบราว์เซอร์ก็จะสามารถแสดงข้อมูลและกราฟต่างๆ ที่นำรูปแบบการนำเสนอจาก Google Visualization API นำเสนอบนเบราว์เซอร์แก่ผู้ใช้

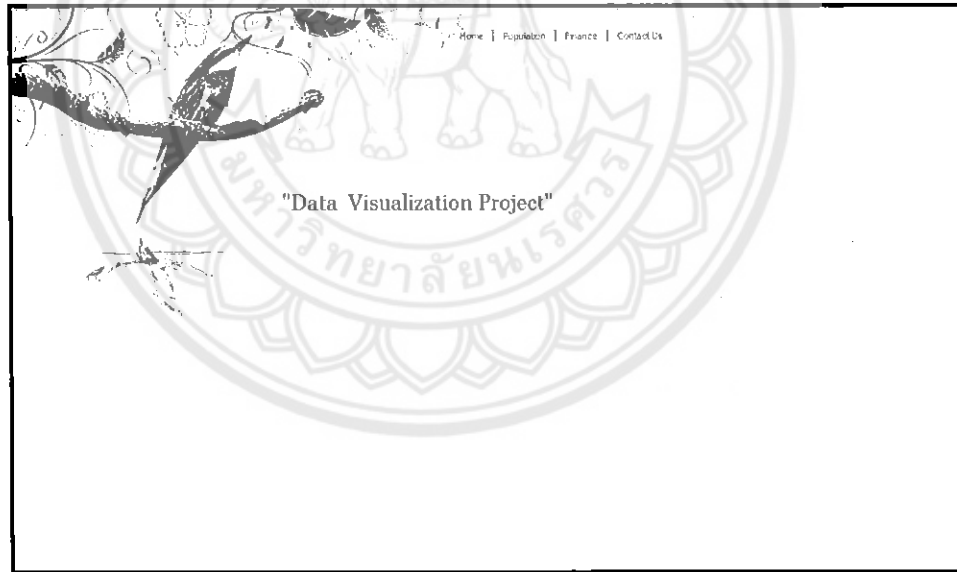
## บทที่ 4

### ผลการทดลอง

จากการดำเนินโครงการ เราได้โปรแกรมการนำเสนอข้อมูลเชิงวิเคราะห์กรณีศึกษาข้อมูลประชากรจังหวัดแม่ฮ่องสอนและงบประมาณการเงินขององค์การบริหารส่วนตำบลแม่มาตัง อำเภอปาย จังหวัดแม่ฮ่องสอน ตั้งแต่ปีงบประมาณ 2552 ถึง 2554 จากที่กล่าวมาแล้วในบทที่ 3 โปรแกรมจะถูกนำเสนอแบ่งเป็น 4 ส่วน คือ Home page Population page และ Contact me page ซึ่งมีผลการทดลองมีดังนี้

#### 4.1 Home page

เมื่อทำการคอมไพล์โปรแกรมจะแสดงหน้า Home บนเบราว์เซอร์ โดยในหน้า Home นั้น จะแสดงเมนูเพื่อให้ลิงก์ไปยังหน้าหลักอื่นๆ ซึ่งประกอบไปด้วย 4 เมนู คือ Home Population Finance และ Contact me



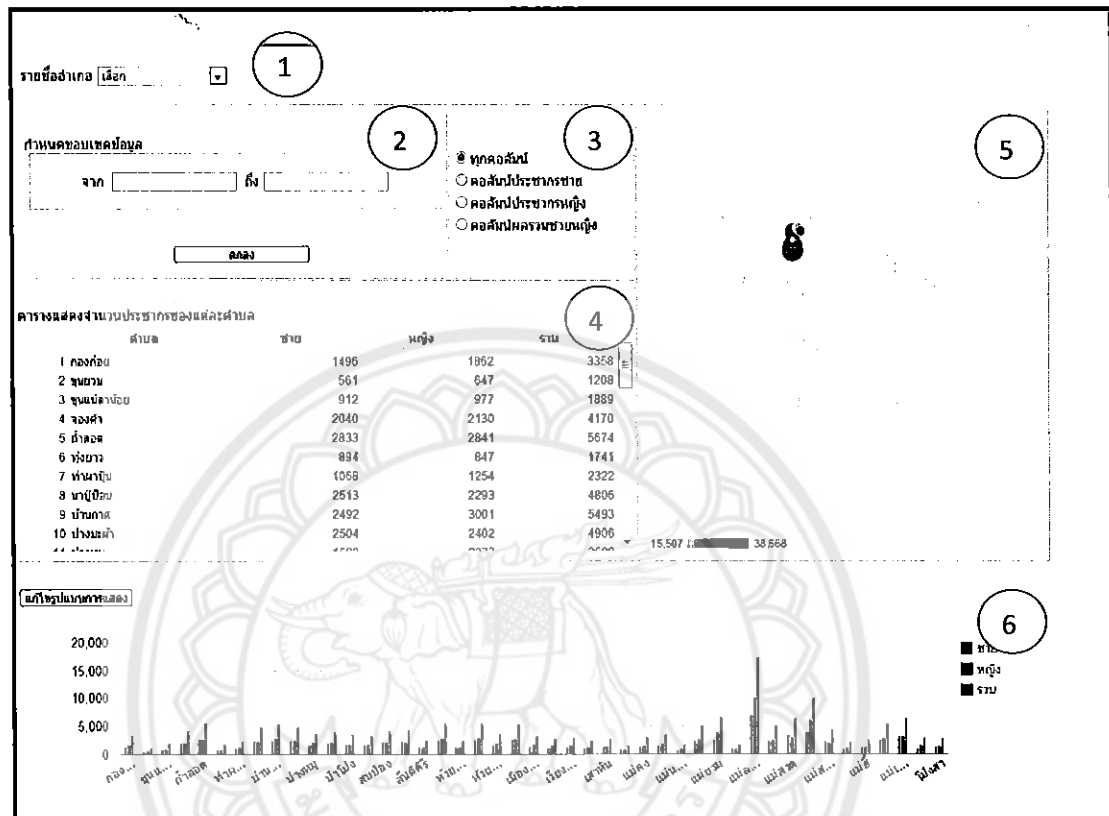
รูปที่ 4.1 แสดงหน้าเว็บแอปพลิเคชันหน้า Home

จากรูปที่ 4.1 จะแสดงเมนูอยู่ด้านบนสุด โดยเมนู Home เป็นการเชื่อมโยงมาหน้าแรกดังรูปที่ 4.1 เมนู Population ทำหน้าที่เชื่อมโยงไปยังหน้า Population ซึ่งทำการแสดงข้อมูลเกี่ยวกับประชากร เมนู Finance ทำหน้าที่เชื่อมโยงไปยังหน้า Finance ทำการแสดงผลเกี่ยวกับงบประมาณการเงินและหน้า Contact me ทำหน้าที่เชื่อมโยงไปยังหน้า Contact me ซึ่งจะแสดงผลเกี่ยวกับข้อมูลผู้จัดทำโครงการ



## 4.2 Population page

หน้า Population จะนำเสนอข้อมูลเกี่ยวกับประชากรของจังหวัดแม่ฮ่องสอน โดยได้ออกแบบให้โปรแกรมสามารถแสดงข้อมูลประชากรดังรูปที่ 4.2



รูปที่ 4.2 แสดงหน้าเว็บแอปพลิเคชันของการนำเสนอข้อมูลเกี่ยวกับประชากร

จากรูปที่ 4.2 จะเห็นได้ว่าหน้าเว็บแอปพลิเคชันจะประกอบไปด้วย 6 ส่วนการนำเสนอ ซึ่งแต่ละส่วนผู้จัดทำได้แทนด้วยหมายเลขลงไปเพื่อความสะดวกในการอธิบาย

หมายเลข 1 เป็นฟังก์ชันที่ใช้เลือกการแสดงผลของประชากรของตำบลในแต่ละอำเภอ โดยจะมีรายการให้เลือกดังนี้

The dropdown menu shows the following options:

- เลือก
- เลือก
- อำเภอเมืองแม่ฮ่องสอน
- อำเภอขุนยวน
- อำเภอป่า
- อำเภอแม่สะเรียง
- อำเภอแม่ลาน้อย
- อำเภอสบเมย
- อำเภอปางมะผ้า

รูปที่ 4.3 แสดงรายการฟังก์ชันการแสดงผลของประชากรของตำบลในแต่ละอำเภอ

- ฟังก์ชันเลือก คือ การเลือกแสดงข้อมูลประชากรของแต่ละตำบลทั้งหมดในจังหวัดแม่ฮ่องสอน
- ฟังก์ชันอำเภอเมืองแม่ฮ่องสอน คือ การเลือกแสดงข้อมูลจำนวนประชากรของแต่ละตำบลในอำเภอเมืองแม่ฮ่องสอน
- ฟังก์ชันอำเภอขุนยวม คือ การเลือกแสดงข้อมูลจำนวนประชากรของแต่ละตำบลในอำเภอขุนยวม
- ฟังก์ชันอำเภอปาย คือ การเลือกแสดงข้อมูลจำนวนประชากรของแต่ละตำบลในอำเภอปาย
- ฟังก์ชัน อำเภอแม่สะเรียง คือ การเลือกแสดงข้อมูลจำนวนประชากรของแต่ละตำบลในอำเภอแม่สะเรียง
- ฟังก์ชันอำเภอแม่ลาน้อย คือ การเลือกแสดงข้อมูลจำนวนประชากรของแต่ละตำบลในอำเภอแม่ลาน้อย
- ฟังก์ชันอำเภอสบเมย คือ การเลือกแสดงข้อมูลจำนวนประชากรของแต่ละตำบลในอำเภอสบเมย
- ฟังก์ชันอำเภอปางมะผ้า คือ การเลือกแสดงข้อมูลจำนวนประชากรของแต่ละตำบลในอำเภอปางมะผ้า

หมายเลข 2 เป็นฟังก์ชันสำหรับกำหนดขอบเขตของจำนวนประชากรของแต่ละตำบลที่ถูกเลือกมาจากฟังก์ชันการแสดงผลของประชากรของตำบลในแต่ละอำเภอ

**กำหนดขอบเขตข้อมูล**

จาก  ถึง

รูปที่ 4.4 แสดงการฟังก์ชันการกำหนดขอบเขตของจำนวนประชากรในแต่ละตำบล

การใช้งานของฟังก์ชันการกำหนดขอบเขตจำนวนประชากรของแต่ละตำบลนี้เป็นการกำหนดค่าต่ำสุดและสูงสุดให้กับระบบเพื่อนำเสนอข้อมูลที่อยู่ในช่วงที่ผู้ใช้ได้กำหนดไว้ โดยสามารถกำหนดได้ 4 กรณีด้วยกัน คือ

1. กรณีไม่มีการกำหนดขอบเขต หรือกล่าวอีกนัยหนึ่งว่า ไม่มีการกรอกข้อมูลตัวเลขลงในฟังก์ชัน ระบบจะทำการแสดงข้อมูลประชากรทั้งหมด โดยไม่มีการกำหนดขอบเขตจำนวนประชากรที่แสดง

**กำหนดขอบเขตข้อมูล**

จาก  ถึง

**ทุกคนสัมพันธ์**

**ความสัมพันธ์ชาย**

**ความสัมพันธ์หญิง**

**ความสัมพันธ์รวมชายหญิง**

---

**ตารางแสดงจำนวนประชากรของแต่ละตำบล**

	ตำบล	ชาย	หญิง	รวม
1	กองก้อย	1496	1862	3358
2	ขุนยวม	561	647	1208
3	ขุนแม่ลาเมือง	912	977	1889
4	จองคำ	2040	2130	4170
5	ตำลอบ	2833	2841	5674
6	ทุ่งยาว	894	847	1741
7	ท่าผาใหม่	1068	1254	2322
8	นาปู่ป้อม	2513	2293	4806
9	บ้านกาต	2492	3001	5493
10	ปางมะพร้าว	2504	2402	4906
11	.....	1200	2070	3270

รูปที่ 4.5 แสดงข้อมูลประชากร โดยไม่ถูกกำหนดขอบเขตของจำนวนประชากร

2. กรณีกำหนดเฉพาะขอบเขตล่าง คือมีการกรอกตัวเลขเฉพาะช่องแรก ดังรูปที่ 4.6 เป็นการกำหนดค่าต่ำสุด ให้ระบบแสดงข้อมูลที่มีจำนวนประชากรเท่ากับและมากกว่าจำนวนตัวเลขที่กรอกเข้าไป

**กำหนดขอบเขตข้อมูล**

จาก  ถึง

ทุกคอลัมน์

คอลัมน์ประชากรชาย

คอลัมน์ประชากรหญิง

คอลัมน์ผลรวมชายหญิง

**ตารางแสดงจำนวนประชากรของแต่ละตำบล**

	ตำบล	ชาย	หญิง	รวม	
1	กองก้อย	1496	1862	3358	↑ ↓
2	จองคำ	2040	2130	4170	
3	ท่าลวด	2833	2841	5674	
4	ท่าผามิม	1068	1254	2322	
5	นาปู่ป้อม	2513	2293	4806	
6	บ้านกาต	2492	3001	5493	
7	ปางมะผ้า	2504	2402	4906	
8	ปางหมู	1620	2073	3698	
9	ป่าแม่	2024	2094	4118	
10	ป่าโปง	1711	1826	3537	

รูปที่ 4.6 แสดงข้อมูลประชากรที่ถูกกำหนดขอบเขตล่างของจำนวนประชากร

รูปที่ 4.6 เมื่อทำการใส่ตัวเลขให้กับ Textbox ช่องแรก แล้วกดตกลง ระบบจะทำการคิวรีข้อมูลประชากรของตำบลที่มีประชากรตั้งแต่ 1000 คนมาทำการนำเสนอ ซึ่งจากรูปจะเห็นได้ว่าทุกคอลัมน์จะทำการแสดงข้อมูลประชากรตั้ง 1000 คนทั้งหมด เนื่องจากไม่ได้ระบุให้กำหนดค่าเริ่มต้นและค่าสิ้นสุดให้กับคอลัมน์ใดคอลัมน์หนึ่ง

3. กรณีกำหนดเฉพาะขอบเขตบน คือมีการกรอกตัวเลขเฉพาะช่องสุดท้าย ดังรูปที่ 4.7 เป็นการกำหนดค่าสูงสุด ให้ระบบแสดงข้อมูลที่มีจำนวนประชากรน้อยกว่าและเท่ากับจำนวนตัวเลขที่กรอกไป

**กำหนดขอบเขตข้อมูล**

จาก  ถึง

ทุกคนสัมพันธ์

คอลัมน์ประชากรชาย

คอลัมน์ประชากรหญิง

คอลัมน์ผลรวมชายหญิง

---

**ตารางแสดงจำนวนประชากรของแต่ละตำบล**

	ตำบล	ชาย	หญิง	รวม
1	ขุนวม	561	647	1208
2	ขุนแม่ลาน้อย	912	977	1889
3	ทุ่งยาว	894	847	1741
4	แม่กิ๊	809	744	1553
5	แม่น้ำจาง	811	937	1748
6	แม่ขวมน้อย	898	914	1812

รูปที่ 4.7 แสดงข้อมูลประชากรที่ถูกกำหนดขอบเขตบนของจำนวนประชากร

รูปที่ 4.7 เมื่อทำการใส่ตัวเลขให้กับ Textbox ช่องที่สอง แล้วกดตกลง ระบบจะทำการคิวรีข้อมูลประชากรของตำบลที่มีประชากรตั้งแต่ 0 ถึง 2000 คนมาทำการนำเสนอ ซึ่งจากรูปจะเห็นว่าทุกคอลัมน์จะทำการแสดงข้อมูลประชากรไม่เกิน 2000 คนทั้งหมด เนื่องจากไม่ได้ระบุให้กำหนดค่าเริ่มต้นและค่าสิ้นสุดให้กับคอลัมน์ใดคอลัมน์หนึ่ง

4. กรณีกำหนดขอบเขตบนและขอบเขตล่างบน คือมีการกรอกตัวเลขทั้ง 2 ช่อง ดังรูปที่ 4.8 เป็นการกำหนดว่า ให้ระบบแสดงข้อมูลที่มีจำนวนประชากรระหว่างตัวเลขช่องแรกถึงตัวเลขช่องที่สองที่กรอกไป

**กำหนดขอบเขตข้อมูล**

จาก  ถึง

ทุกคนสัมพันธ์

คอลัมน์ประชากรชาย

คอลัมน์ประชากรหญิง

คอลัมน์ผลรวมชายหญิง

**ตารางแสดงจำนวนประชากรของแต่ละตำบล**

	ตำบล	ชาย	หญิง	รวม
1	ท่าผามิ้ม	1068	1254	2322
2	สันตศีร์	1228	1220	2448
3	ห้วยบุ่ง	1259	1238	2497
4	เมืองแปง	1161	1560	2731
5	เวียงเหนือ	1339	1551	2890
6	เวียงใต้	1202	1302	2504
7	เสาหิน	1391	1417	2808
8	แม่ฮี้	1318	1333	2651
9	แม่โถ	1159	1704	2863
10	โป่งสา	1491	1487	2981

รูปที่ 4.8 แสดงข้อมูลประชากรที่ถูกกำหนดขอบเขตล่างและขอบเขตบน

รูปที่ 4.8 เมื่อทำการใส่ตัวเลขให้กับ Textbox ทั้งสองช่อง แล้วกดตกลง ระบบจะทำการคิวรีข้อมูลประชากรของตำบลที่มีประชากรตั้งแต่ 1000 ถึง 3000 คนมาทำการนำเสนอ ซึ่งจากรูปจะเห็นได้ว่าทุกคอลัมน์จะทำการแสดงข้อมูลประชากรตั้งแต่ 1000 คน ถึง 3000 คนทั้งหมด เนื่องจากว่าไม่ได้ระบุให้กำหนดค่าเริ่มต้นและค่าสิ้นสุดให้กับคอลัมน์ใดคอลัมน์หนึ่ง

หมายเลข 3 เป็นฟังก์ชันการกำหนดประเภทข้อมูลประชากรของแต่ละอำเภอที่ต้องการแสดง ซึ่งมีให้เลือกใช้ด้วยกัน 4 แบบ คือ

ทุกคอลัมน์

คอลัมน์ประชากรชาย

คอลัมน์ประชากรหญิง

คอลัมน์ผลรวมชายหญิง

รูปที่ 4.9 แสดงฟังก์ชันการเลือกประเภทของข้อมูล

- ฟังก์ชันทุกคอลัมน์ คือ การแสดงข้อมูลของประชากร คอลัมน์ประชากรชาย คอลัมน์ประชากรหญิงและคอลัมน์ประชากรทั้งหมด
- ฟังก์ชันคอลัมน์ประชากรชาย คือ การแสดงข้อมูลเฉพาะคอลัมน์ประชากรชาย
- ฟังก์ชันคอลัมน์ประชากรหญิง คือ การแสดงข้อมูลเฉพาะคอลัมน์ประชากรหญิง
- ฟังก์ชันคอลัมน์ประชากรรวมชายหญิง คือ การแสดงข้อมูลประชากรทั้งหมด

หมายเลข 4 เป็นการแสดงข้อมูลในรูปแบบตาราง ที่ทำการดึงรูปแบบการนำเสนอจาก Google Visualization API มาใช้ โดยในตารางนี้จะทำการแสดงข้อมูลประชากรจากการคิวรีข้อมูลของฟังก์ชันใน หมายเลข 1 หมายเลข 2 และหมายเลข 3 มาแสดงผล ซึ่งตารางนี้มีความสามารถในการจัดเรียงข้อมูลในแต่ละคอลัมน์ ดังนี้

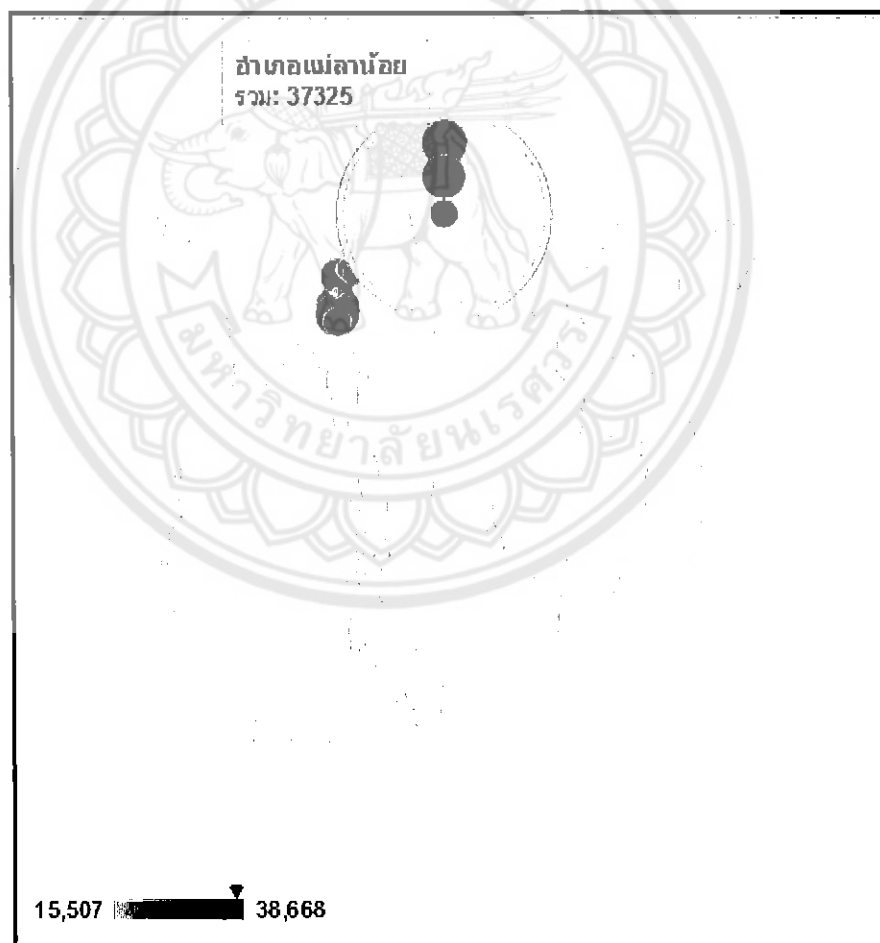
	ตำบล	ชาย	หญิง	รวม
1	กองก้อย	1496	1862	3358
2	ขุนยวม	561	647	1208
3	ขุนแม่ลาน้อย	912	977	1889
4	จองคำ	2040	2130	4170
5	ตำลอบ	2833	2841	5674
6	ทุ่งยาว	894	847	1741
7	ท่าผาปืม	1068	1254	2322
8	นาปู่ป้อม	2513	2293	4806
9	บ้านกาศ	2492	3001	5493
10	ปางมะผ้า	2504	2402	4906

รูปที่ 4.10 แสดงข้อมูลประชากรในรูปแบบของตารางโดยนำรูปแบบการนำเสนอจาก Google Visualization API มาใช้

จากรูปที่ 4.10 เป็นการนำเสนอข้อมูลในรูปแบบตารางซึ่งเป็นรูปแบบการนำเสนออย่างหนึ่งของ Google Visualization API โดยตารางนี้จะมีคุณสมบัติที่ช่วยในการจัดเรียงข้อมูลในแต่ละคอลัมน์ ดังนี้

- คอลัมน์ตำบล สามารถจัดเรียงข้อมูลตามลำดับตัวอักษรได้
- คอลัมน์ชาย สามารถจัดเรียงข้อมูลจากน้อยไปมาก หรือมากไปน้อยได้
- คอลัมน์หญิง สามารถจัดเรียงข้อมูลจากน้อยไปมาก หรือมากไปน้อยได้
- คอลัมน์รวม สามารถจัดเรียงข้อมูลจากน้อยไปมาก หรือมากไปน้อยได้

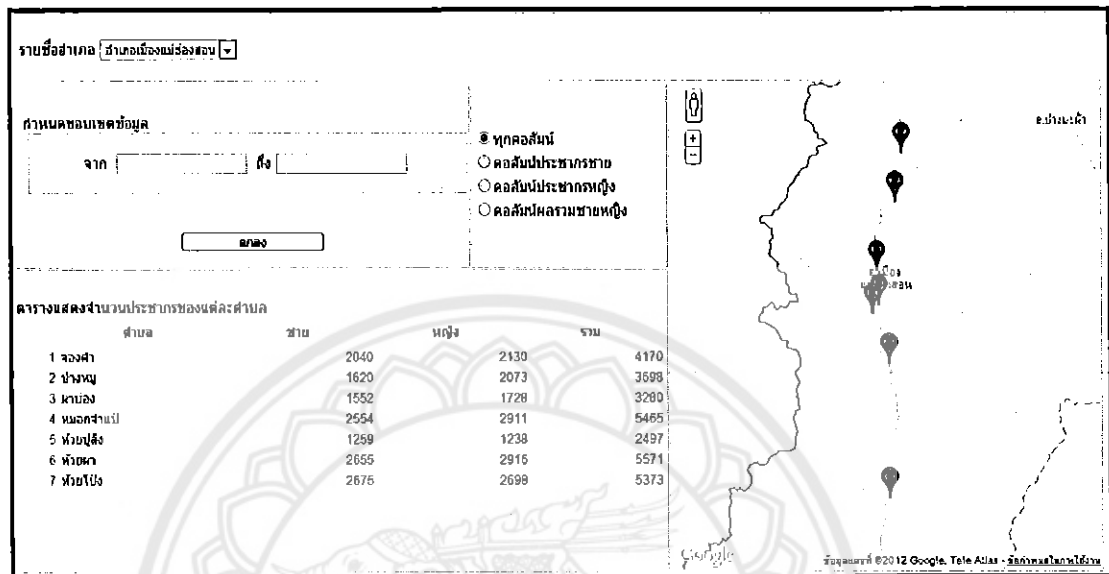
หมายเลข 5 เป็นการแสดงตำแหน่งต่างๆ ของแต่ละตำบลในจังหวัดแม่ฮ่องสอน



รูปที่ 4.11 แสดงความหนาแน่นของประชากรของแต่ละอำเภอในจังหวัดแม่ฮ่องสอน

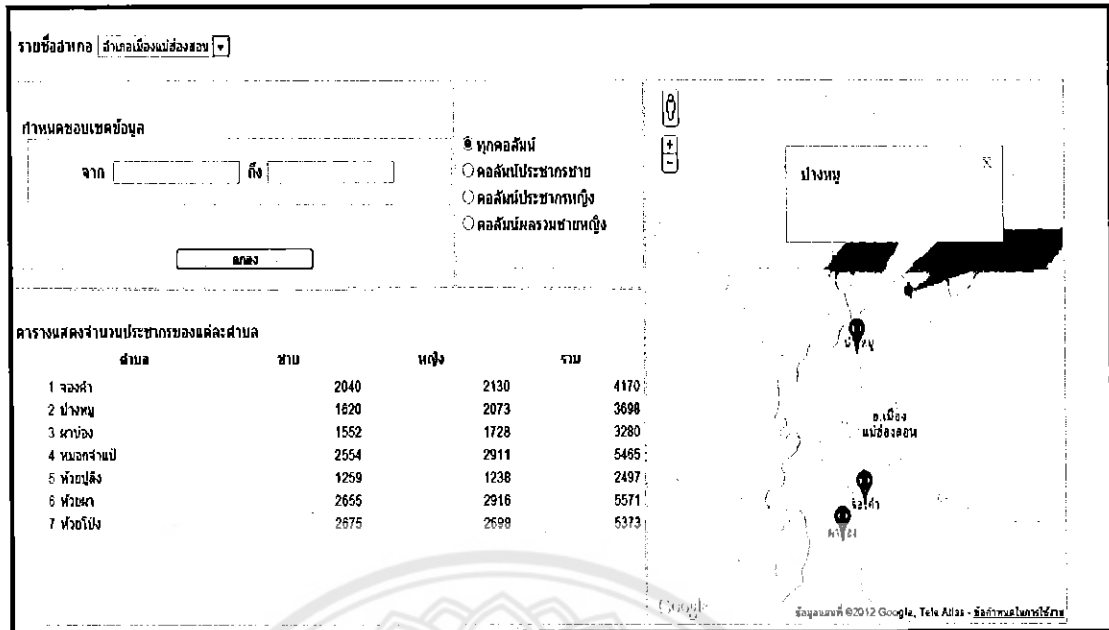


เมื่อทำการเลือกอำเภอที่ต้องการจากฟังก์ชันหมายเลข 1 แล้ว แผนที่ จะเปลี่ยนไปตามอำเภอที่เลือกที่ โดยจะแสดงตำบลของอำเภอนั้นๆ และมีจุดระบุตำแหน่งสีแดงบนแผนที่ เมื่อทำการคลิกที่รายชื่อตำบลในตาราง แผนที่ จะเปลี่ยนจุดระบุตำแหน่งจากสีแดงเป็นสีน้ำเงิน เพื่อให้ทราบว่าตำบลที่เลือกอยู่ตำแหน่งใดของแผนที่ ดังรูปที่ 4.12



รูปที่ 4.12 แสดงแผนที่เมื่อทำการเลือกรายชื่ออำเภอ

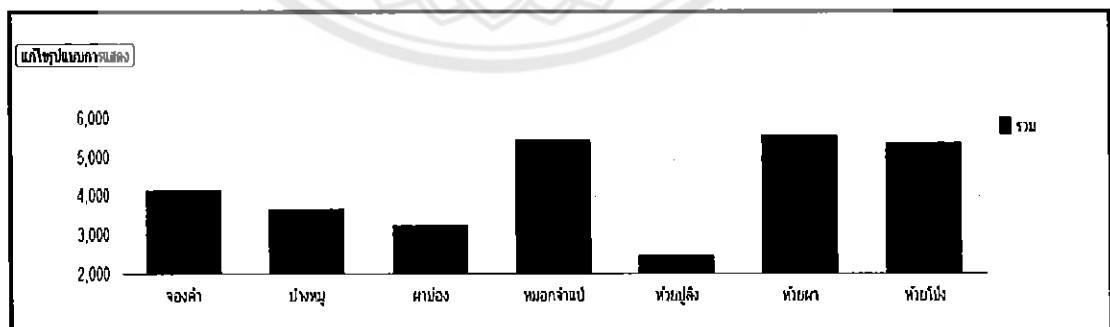
จากรูปที่ 4.12 เมื่อทำการเลือกรายชื่ออำเภอ จากรูปนี้เลือกอำเภอเมืองแม่ฮ่องสอน แผนที่ จะเปลี่ยนไปแสดงตำแหน่งของตำบลในอำเภอเมืองแม่ฮ่องสอน เมื่อมีการคลิกที่จุดระบุตำแหน่งบนแผนที่ก็จะปรากฏข้อความที่แสดงชื่อของตำบลที่เลือกบนแผนที่ และจะเกิดแถบสีน้ำเงินบนตารางที่ตำบลเดียวกันกับที่ทำการเลือกบนแผนที่



รูปที่ 4.13 แสดงการเลือกตำแหน่งบนแผนที่

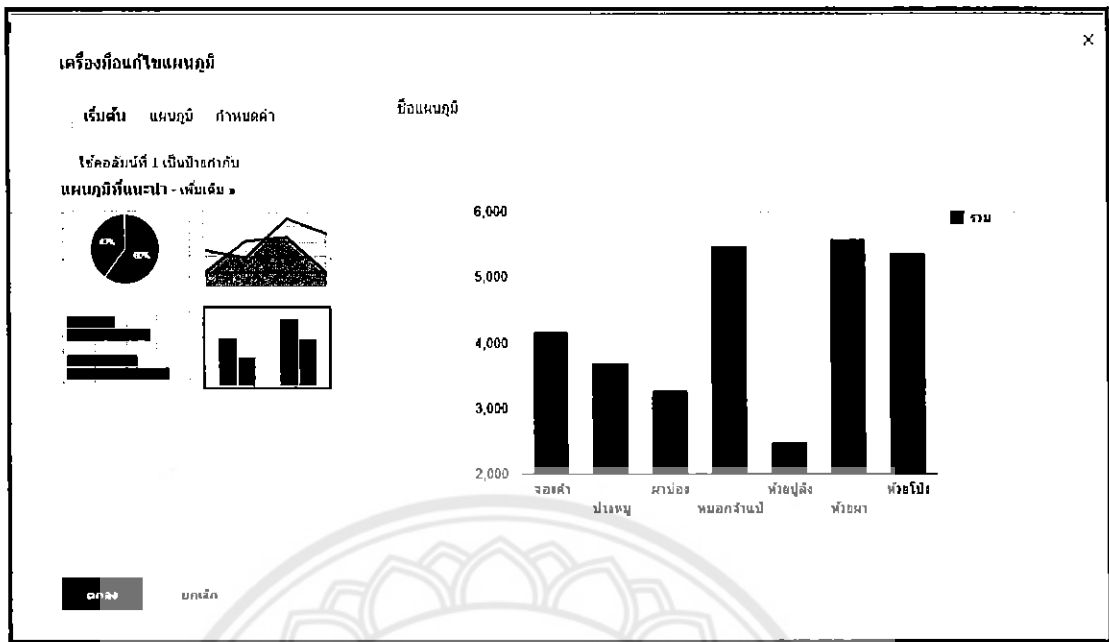
รูปที่ 4.13 เมื่อทำการคลิกที่ตำบลปางหมูในแผนที่ จุดระบุตำแหน่งสีแดงก็จะเปลี่ยนเป็นสีน้ำเงินและจะมีข้อความแสดงขึ้นมาแจ้งเป็นชื่อตำบล และจะเกิดแถบสีน้ำเงินบนตารางแถวตำบลปางหมู

หมายเลข 6 เป็นการนำเสนอข้อมูล ที่ทำการดึงรูปแบบการนำเสนอจาก Google Visualization API มาใช้ โดยในการนำเสนอนี้จะทำการแสดงข้อมูลประชากรจากการสรุปข้อมูลของฟังก์ชันใน หมายเลข 1 หมายเลข 2 และหมายเลข 3 มาแสดงผล ซึ่งการนำเสนอนี้สามารถปรับเปลี่ยนรูปแบบของการนำเสนอได้ ตามลักษณะของข้อมูลที่ได้รับเข้าไป



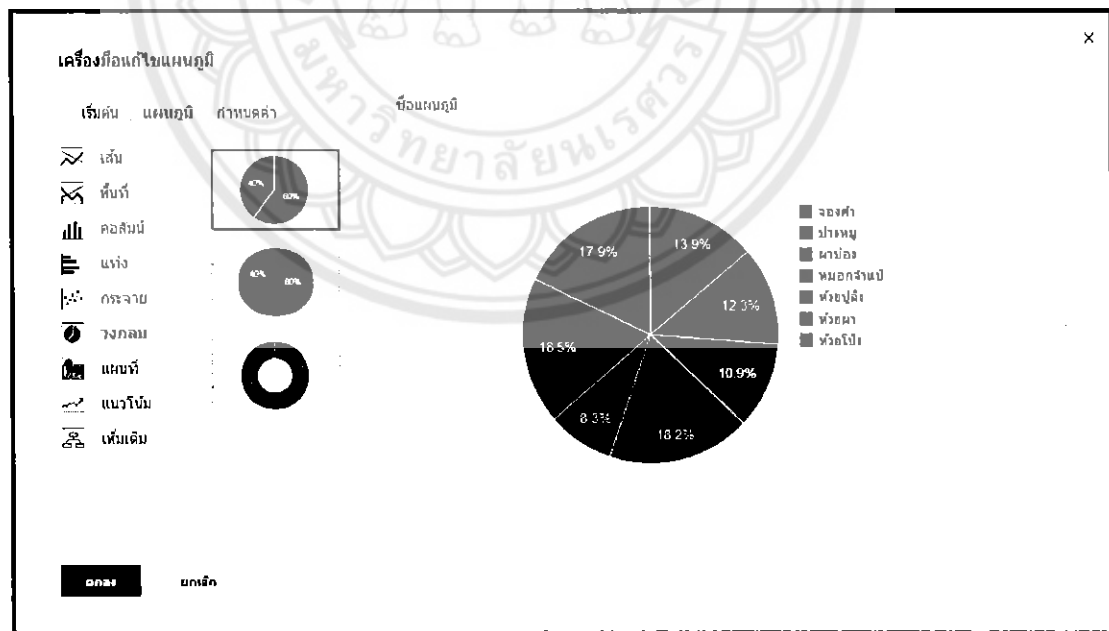
รูปที่ 4.14 แสดงการนำเสนอข้อมูลประชากรของอำเภอเมืองแม่ฮ่องสอน จังหวัดแม่ฮ่องสอน

เมื่อคลิกที่ปุ่ม “แก้ไขรูปแบบการแสดงผล” จะสามารถทำการแก้ไขรูปแบบการนำเสนอได้ ซึ่งหน้าแรกที่ปรากฏจะเป็นหน้าแนะนำการนำเสนอเบื้องต้น หน้าแรกนี้ที่ปรากฏนี้จะเปลี่ยนแปลงรูปแบบการแนะนำการนำเสนอ โดยขึ้นอยู่กับรูปแบบข้อมูลที่ได้รับเข้าไป



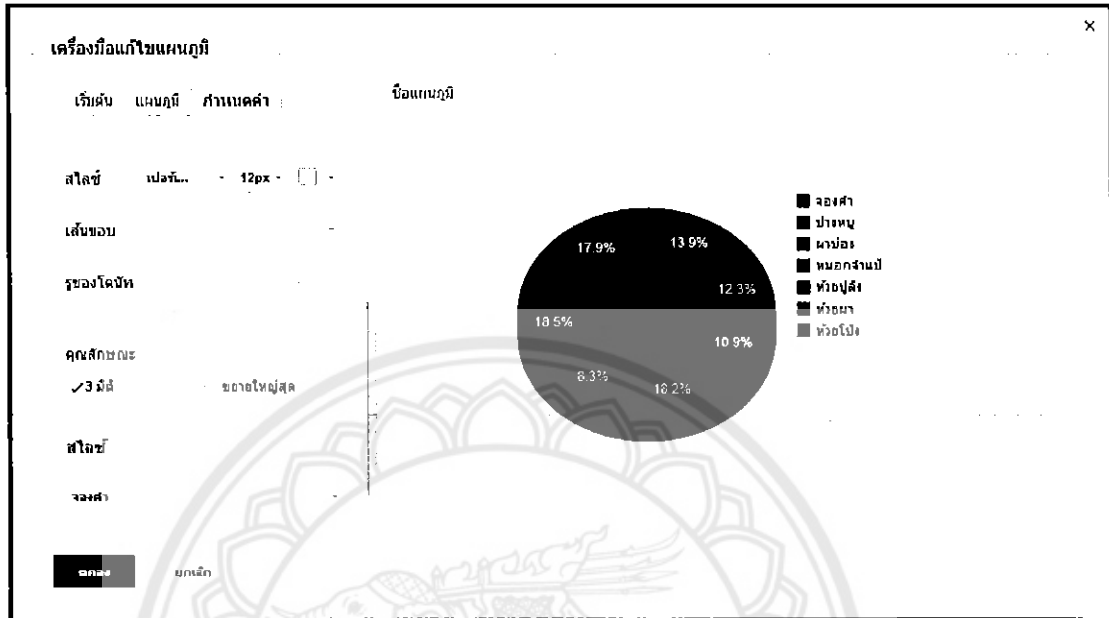
รูปที่ 4.15 แสดงเครื่องมือแก้ไขรูปแบบการนำเสนอหน้าเริ่มต้น

รูปที่ 4.15 แสดงเครื่องมือการแก้ไขรูปแบบการนำเสนอหน้าเริ่มต้น โดยหน้านี้จะแนะนำรูปแบบการนำเสนอที่เหมาะสมกับชนิดของข้อมูลที่ได้รับเข้ามา



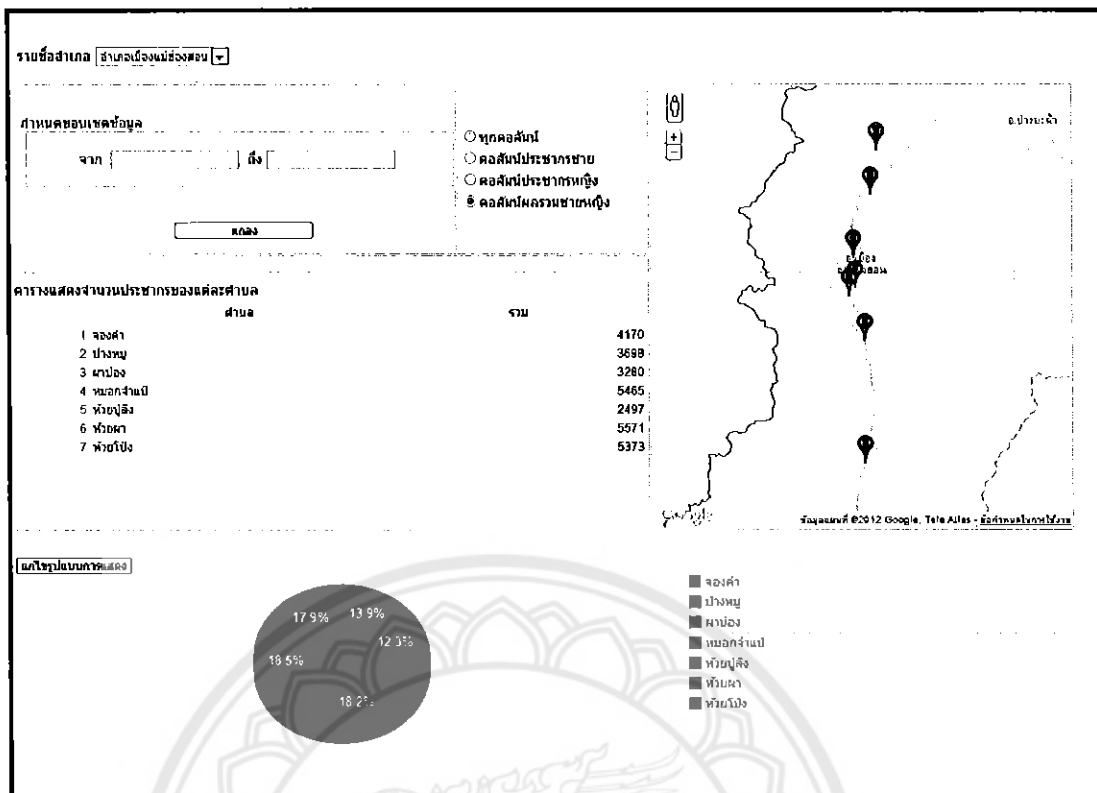
รูปที่ 4.16 แสดงเครื่องมือแก้ไขแผนภูมิหน้าแผนภูมิ

นอกจากรูปแบบการนำเสนอที่แนะนำจากหน้าเริ่มต้นแล้ว เมื่อคลิกที่ฟังก์ชันเมนู สามารถเลือกรูปแบบการนำเสนออื่นๆ ได้ เช่น การนำเสนอแบบเส้น การนำเสนอแบบพื้นที่ การนำเสนอแบบคอลัมน์ การนำเสนอแบบวงกลม ดังรูปที่ 4.16



รูปที่ 4.17 แสดงเครื่องมือแก้ไขแผนภูมิหน้ากำหนดค่า

รูปที่ 4.17 เป็นหน้ากำหนดค่าของเครื่องมือแก้ไขแผนภูมิ จะเกี่ยวกับกับการกำหนดลักษณะเบื้องต้นของการนำเสนอ เช่น สีที่ใช้ในการนำเสนอข้อมูล สีตัวอักษร แบบตัวอักษร และพื้นหลัง เป็นต้น เมื่อทำการกำหนดรูปแบบที่ต้องการ แล้วคลิกตกลง จะได้รับรูปแบบการนำเสนอที่กำหนดไว้



รูปที่ 4.18 แสดงแผนภูมิเมื่อทำการปรับเปลี่ยนรูปแบบ

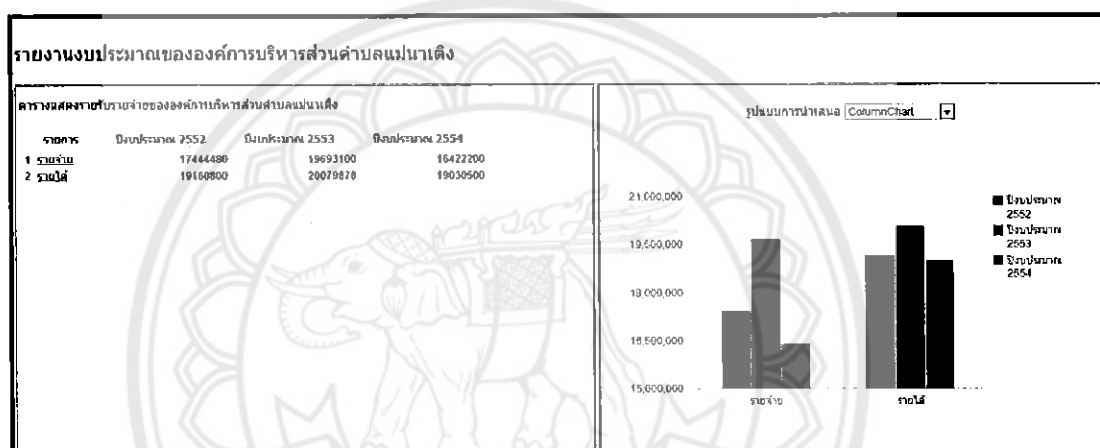
รูปที่ 4.18 เป็นการแสดงการนำเสนอในรูปแบบของกราฟวงกลม 3 มิติ ซึ่งได้จากการปรับเปลี่ยนและกำหนดค่าในปุ่มแก้ไขรูปแบบการแสดงผล โดยกราฟวงกลมนี้จะได้รับข้อมูลจากการเลือกรายชื่ออำเภอเมืองแม่ฮ่องสอน และการเลือกแสดงข้อมูลรวมชายหญิง โดยไม่ได้มีการกำหนดขอบเขตจำนวนประชากร

### 4.3 Finance page

หน้า Finance จะนำเสนอข้อมูลเกี่ยวกับงบประมาณการเงินขององค์การบริหารส่วนตำบลแม่नादेิง อำเภอป่าเยื้อง จังหวัดแม่ฮ่องสอน ตั้งแต่ปีงบประมาณ 2552 – 2554 และพื้นที่บริการ โดยจะแบ่งการนำเสนอออกเป็น 4 ส่วนคือ

#### 4.3.1 รายได้และรายจ่ายขององค์การบริหารส่วนตำบลแม่नादेิง

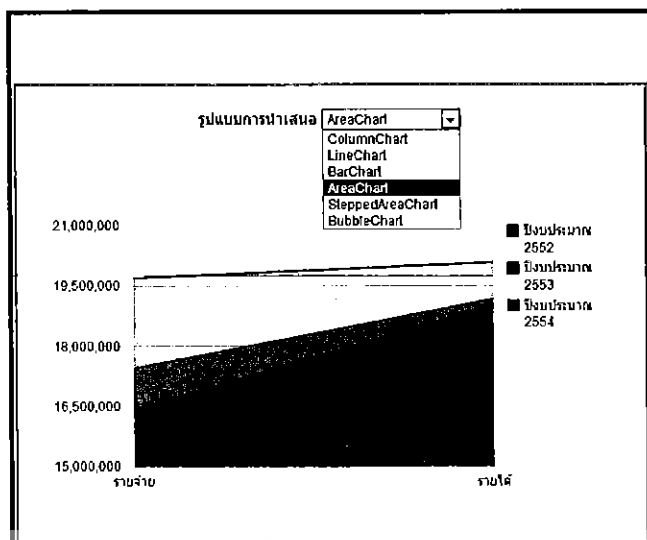
การนำเสนอข้อมูลรายได้และรายจ่ายขององค์การบริหารส่วนตำบลแม่नादेิง ตั้งแต่ปีงบประมาณ 2552-2554 นั้น จะนำเสนอในรูปแบบของตาราง โดยใช้รูปแบบการตารางของ Google Visualization API และแสดงข้อมูลในรูปแบบกราฟคอลัมน์เป็นกราฟเริ่มต้น



รูปที่ 4.19 แสดงการนำเสนอข้อมูลรายได้และรายจ่ายของ อบต.แม่नादेิง

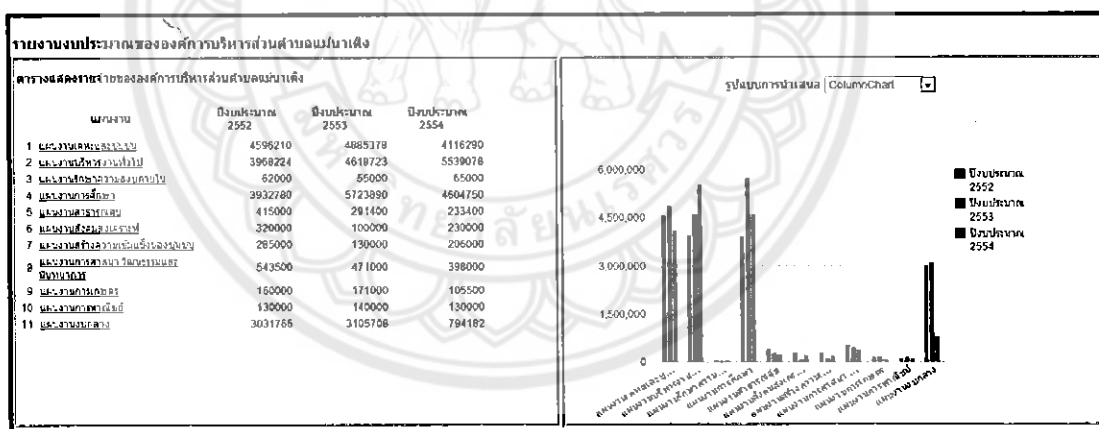
จากรูปที่ 4.19 จะเห็นว่าการนำเสนอรายได้และรายจ่ายถูกนำเสนอด้วยตารางและกราฟที่นำรูปแบบการนำเสนอมาจาก Google Visualization API และกราฟที่นำเสนอสามารถเลือกรูปแบบกราฟรูปแบบอื่นได้ตามรูปที่ 4.20

สำหรับตารางที่ใช้นำเสนอข้อมูลเมื่อทำการคลิกรายการในคอลัมน์รายการ ในที่นี้คือรายจ่ายและรายรับ จะเชื่อมโยงไปยังรายละเอียดในแต่ละรายการ



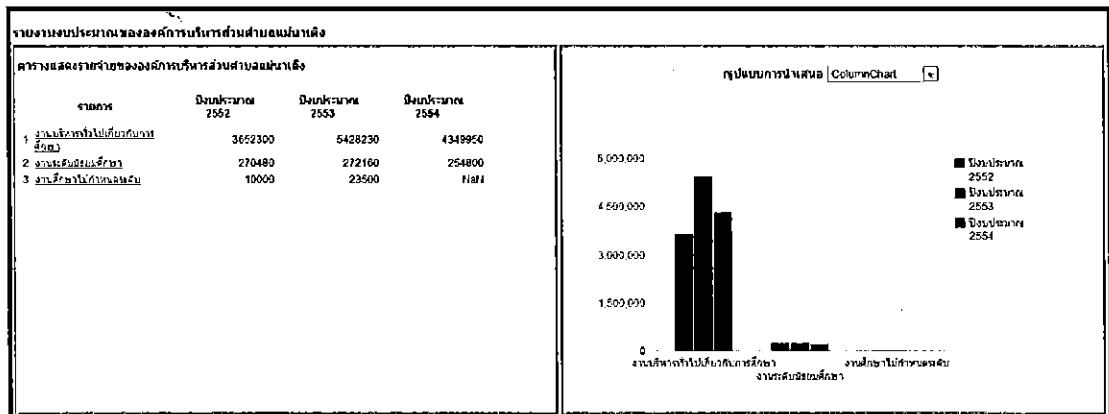
รูปที่ 4.20 แสดงรูปแบบกราฟแบบ Area Chart

รูปที่ 4.20 แสดงรูปแบบกราฟแบบ Area Chart ของข้อมูลรายได้และรายจ่ายในปีงบประมาณ 2552, 2553 และ 2554 และสามารถเปลี่ยนรูปแบบกราฟเป็น Line Chart, Bar Chart, Stepped Area Chart, Bubble Chart หรือ Column Chart ได้



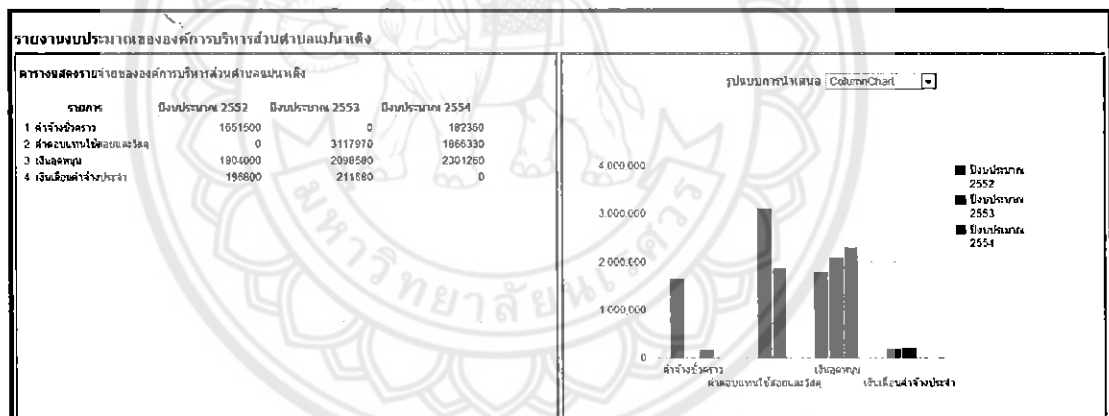
รูปที่ 4.21 แสดงรายจ่ายการใช้งบประมาณตามแผนงาน

รูปที่ 4.21 เมื่อทำการคลิกการรายการรายจ่าย (จากรูปที่ 4.20) ระบบจะทำการคิวรีข้อมูลแผนงานของรายจ่ายแต่ละแผนงานตั้งแต่ปีงบประมาณ 2552-2544 ว่ามีการใช้จ่ายในแต่ละแผนงานเป็นจำนวนเงินเท่าไร และแสดงกราฟด้วยกราฟแบบคอลัมน์เป็นกราฟเริ่มต้น และสามารถเปลี่ยนรูปแบบกราฟได้ เช่นเดียวกับรูปที่ 4.20



รูปที่ 4.22 แสดงประเภทงานการใช้งบประมาณของแผนงานการศึกษา

รูปที่ 4.22 แสดงประเภทงานการใช้งบประมาณของแผนงาน เมื่อทำการคลิกรายการแผนงาน (รูปที่ 4.21) ในที่นี้ขอยกตัวอย่างแผนงานการศึกษาจะแสดงประเภทงานที่ใช้งบประมาณไปใช้จ่ายทั้งหมด ในปี 2552-2554 และแสดงรูปแบบข้อมูลด้วยกราฟคอลัมน์เป็นกราฟเริ่มต้น และสามารถเปลี่ยนรูปแบบกราฟได้ เช่นเดียวกับรูปที่ 4.20



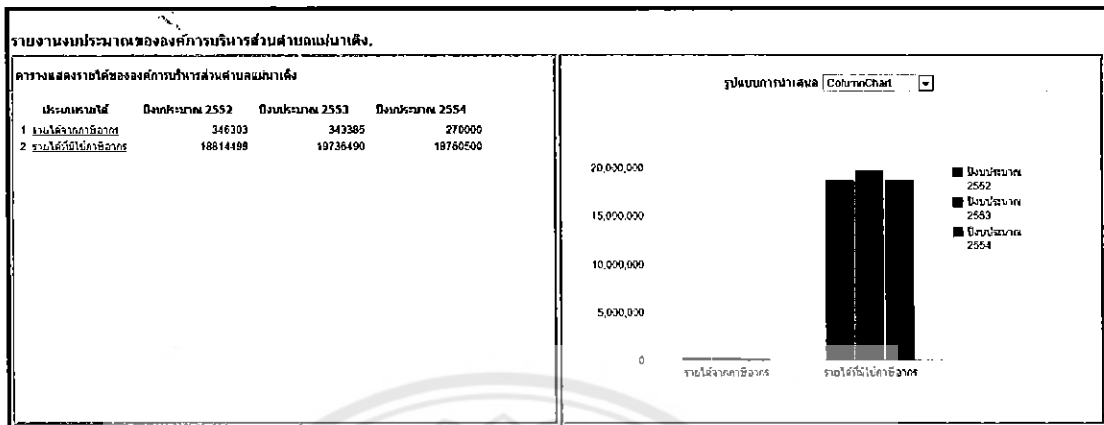
รูปที่ 4.23 แสดงรายการการใช้งบประมาณประเภทงานบริหารการศึกษาทั่วไป

รูปที่ 4.23 แสดงรายการการใช้งบประมาณตามประเภทงาน เมื่อทำการคลิกรายการประเภทงาน (รูปที่ 4.22) ในที่นี้ขอยกตัวอย่างประเภทงานบริหารการศึกษาทั่วไป จะแสดงรายการที่ใช้งบประมาณไปใช้จ่ายทั้งหมด ในปี 2552-2554 และแสดงรูปแบบข้อมูลด้วยกราฟคอลัมน์เป็นกราฟเริ่มต้น และสามารถเปลี่ยนรูปแบบกราฟได้ เช่นเดียวกับรูปที่ 4.20

จากการทำงานของระบบในส่วนของการจ่าย จะเห็นได้ว่าเริ่มต้นระบบจะนำเสนอรายการทั้งหมดของทั้ง 3 ปีงบประมาณ เมื่อทำการคลิกไปรายการ ระบบก็จะนำเสนอรายการตามแผนงาน โดยแผนงานของรายการก็จะถูกแบ่งเป็นประเภทงานที่ใช้จ่าย และประเภทงานก็ยังมีรายการของ

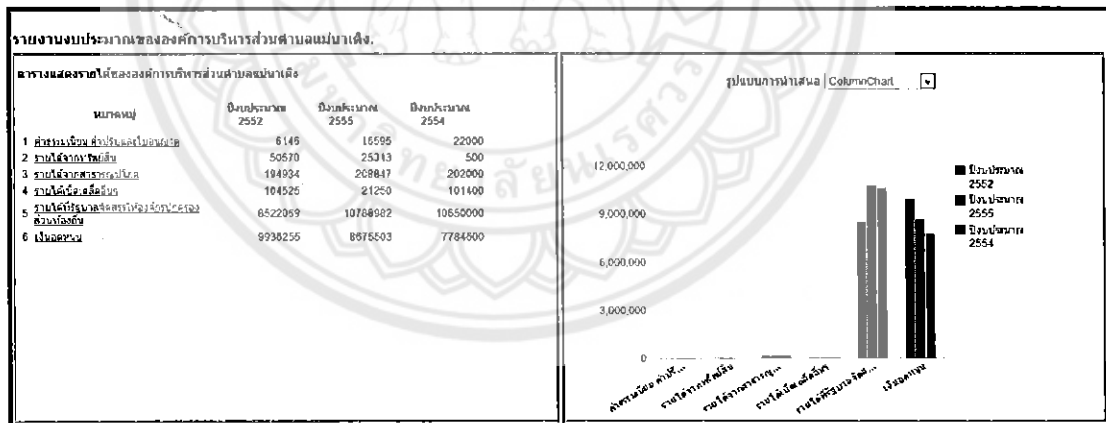


รายจ่ายที่ถูกแบ่งเป็นรายการการใช้จ่าย 7 รายการตามหลักการบันทึกรายจ่ายขององค์การบริหารส่วนตำบลแม่มาเต็ง



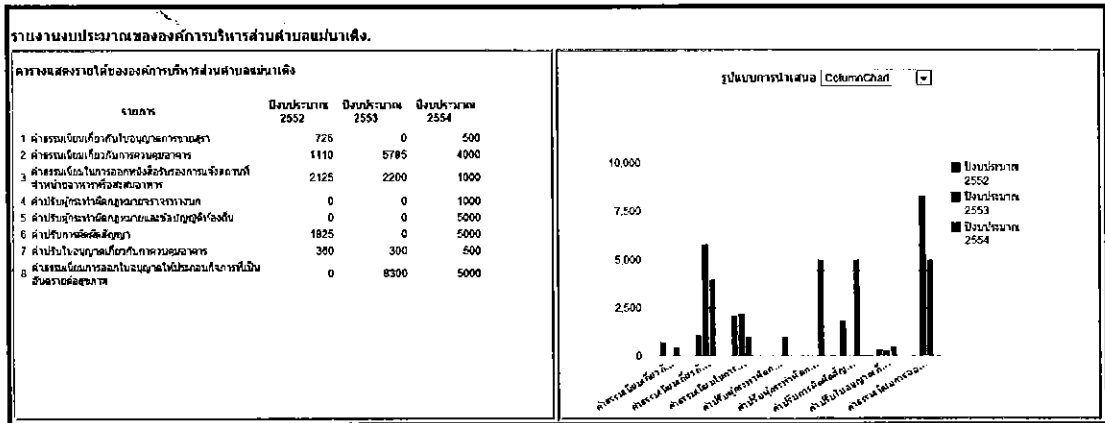
รูปที่ 4.24 แสดงประเภทรายได้

รูปที่ 4.24 แสดงประเภทของรายได้ โดยประเภทของรายได้จะแบ่งเป็น 2 ประเภท คือ ประเภทรายได้จากภาษีอากรและรายได้ที่มีใช้ภาษีอากร ซึ่งการแยกประเภทรายได้นี้เป็นไปตามหลักการการทำบัญชีรายได้ขององค์การบริหารส่วนตำบลแม่มาเต็ง



รูปที่ 4.25 แสดงหมวดหมู่รายจ่ายของรายได้ประเภทรายได้ที่มีใช้ภาษีอากร

รูปที่ 4.25 แสดงหมวดหมู่ของรายจ่าย เมื่อทำการคลิกที่รายการประเภทของรายได้ ระบบจะทำการคิวรีข้อมูลหมวดหมู่ของรายการประเภทรายได้นั้นๆ ออกมาแสดง เพื่อให้ผู้ใช้สามารถดูรายละเอียดของประเภทรายได้นั้นๆได้ ว่าประกอบด้วยรายจ่ายอะไรบ้าง และมีจำนวนเงินในหมวดหมู่ต่างๆ จำนวนเท่าใด



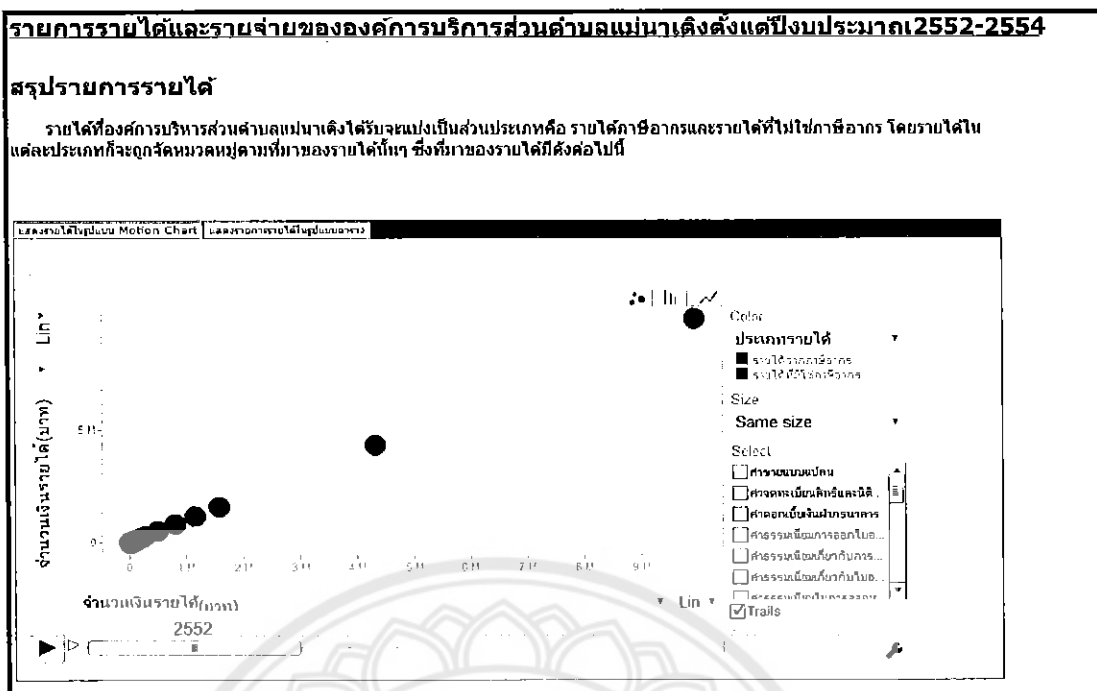
รูปที่ 4.26 แสดงรายการรายได้ของรายได้หมวดหมู่ ค่าธรรมเนียม ค่าปรับและใบอนุญาต

รูปที่ 4.26 แสดงรายการรายได้ของหมวดหมู่ต่างๆ เมื่อทำการคลิกที่รายการหมวดหมู่ใดหมวดหมู่หนึ่งระบบจะทำการคิวรีรายการของรายได้มานำเสนอให้กับผู้ใช้

จากการทำงานของระบบในส่วนของรายได้ จะเห็นได้ว่าเริ่มต้นระบบจะนำเสนอรายได้ทั้งหมดของทั้ง 3 ปีงบประมาณ เมื่อทำการคลิกไปรายได้ ระบบก็จะนำเสนอประเภทของรายได้โดยประเภทของรายได้ก็จะถูกแบ่งเป็นหมวดหมู่ตามลักษณะของรายได้ และหมวดหมู่รายได้ก็ยังมีรายการของรายได้ ซึ่งเป็นที่มาของรายได้ต่างๆ

### 4.3.2 สรุปรายการรายได้

จากการนำเสนอข้อมูลรายการรายได้แยกตามหมวดหมู่ตามหัวข้อ 4.3.1 ที่ผ่านมา ทำให้ทราบรายการรายได้นั้นเฉพาะหมวดหมู่ที่ผู้ใช้ทำการคลิกเข้าไปที่หมวดหมู่ที่ต้องการ หากผู้ใช้ต้องการทราบรายการรายได้ของทุกหมวดหมู่ ผู้จัดทำได้นำมาเสนอในส่วนของการสรุปรายการรายได้ โดยนำเสนอในรูปแบบของ Motion Chart และตาราง ซึ่ง Motion Chart นี้มีความสามารถในการแสดงภาพเคลื่อนไหวทางการเงินจากปี 2552-2554 และยังสามารถเปลี่ยนรูปแบบการนำเสนอเป็น Column Chart และ Line Chart ได้อีกด้วย นอกจากนี้ สามารถเลือกแสดงข้อมูลใดข้อมูลหนึ่งด้วยการเลือกที่ Select ข้อมูลที่ถูกเลือกจะแสดงด้วยสีปกติ แต่ข้อมูลอื่นๆที่ไม่ได้ถูกเลือกก็จะแสดงด้วยสีที่จางลง



รูปที่ 4.27 แสดงการนำเสนอข้อมูลรายการรายได้ด้วย Motion Chart

รูปที่ 4.27 แสดงการนำเสนอข้อมูลรายการรายได้ด้วย Motion Chart เป็นการนำเอาข้อมูลรายการรายได้นำเสนอ โดยระบุสีที่แตกต่างกัน เพื่อแยกรายได้ตามประเภทของรายได้ โดยสีนำเงินคือ รายได้จากภาษีอากร และสีเขียวคือ รายได้ที่ไม่ใช่ภาษีอากร

**รายการรายได้และรายจ่ายขององค์การบริหารส่วนตำบลแม่มาเด็งตั้งแต่ปีงบประมาณ 2552-2554**

**สรุปรายการรายได้**

รายได้ที่องค์การบริหารส่วนตำบลแม่มาเด็งได้รับจะแบ่งเป็นส่วนประเภทคือ รายได้จากภาษีอากรและรายได้ที่ไม่ใช่ภาษีอากร โดยรายได้ในแต่ละประเภทก็จะถูกจัดหมวดหมู่ตามที่มาของรายได้ใดๆ ซึ่งที่มาของรายได้มีดังต่อไปนี้

รายการรายได้	ปีงบประมาณ 2552	ปีงบประมาณ 2553	ปีงบประมาณ 2554
1 ค่ามิโรงเรือนแะว่ที่ดิน	266886	265945	200000
2 ค่ามิโรงห้องที่	10878	11861	10000
3 ค่ามิป่า	68540	65579	60000
4 ค่าธรรมเนียมเกี่ยวกับใบอนุญาตราชการ	726	0	500
5 ค่าธรรมเนียมเกี่ยวกับค่าธรรมเนียมอาคาร	1110	5795	4000
6 ค่าธรรมเนียมในการออกใบอนุญาตของรถจักรยานยนต์ที่จำหน่ายอาหารหรือขนมอาหาร	2125	2200	1000
7 ค่ารับผู้กระทำผิดกฎหมายจราจรทางบก	0	0	1000
8 ค่ารับผู้กระทำผิดกฎหมายและข้อบัญญัติท้องถิ่น	0	0	5000
9 ค่ารับการศึกษาศึกษา	1825	0	5000
10 ค่าปรับใบอนุญาตเกี่ยวกับควบคุมอาคาร	360	300	500
11 ค่าธรรมเนียมในการออกใบอนุญาตใบประกอบกิจการที่เป็นอันตรายต่อสุขภาพ	0	8300	5000
12 ค่าดอกเบี้ยเงินฝากธนาคาร	50570	25313	500
13 รายได้จากสาธารณูปโภคและการพาณิชย์	193434	208097	200000

รูปที่ 4.28 แสดงการนำเสนอข้อมูลรายการรายได้แบบตาราง

รูปที่ 4.28 แสดงการนำเสนอข้อมูลรายการรายได้แบบตาราง การแสดงรายการรายได้แบบตารางนี้ จะช่วยให้ผู้ใช้สามารถเลือกการนำเสนอแบบอื่นนอกเหนือจากกราฟได้

### 4.3.3 สรุปรายการรายจ่าย

การสรุปรายการรายจ่ายนี้ เป็นการสรุปรายการรายจ่ายทุกแผนงานเมื่อแยกตามรายการรายจ่ายแล้ว แต่ละรายการมีค่าใช้จ่ายในแต่ละปีงบประมาณจำนวนเท่าใด ซึ่งการนำเสนอรายจ่ายเช่นนี้จะทำให้ผู้ใช้สามารถวางแผนการใช้จ่ายงบประมาณในแต่ละรายการได้ดีขึ้น เพราะเห็นภาพรวมการใช้จ่ายงบประมาณแต่ละรายการ

สรุปรายการรายจ่าย			
การใช้จ่ายงบประมาณขององค์การบริหารส่วนตำบลแม่มาดี จังหวัดภูเก็ต เป็นงวดปีงบประมาณ โดยในแต่ละแผนงานจะประกอบด้วยประเภทงานต่างๆ และในประเภทงานนั้นๆ ก็จะถูกบันทึกโดยแบ่งเป็นรายการการใช้จ่ายงบประมาณ ซึ่งรายการรายจ่ายต่างๆ มีดังนี้			
รายการรายจ่าย	ปีงบประมาณ 2552	ปีงบประมาณ 2553	ปีงบประมาณ 2554
1 ค่าอาหารคนปกติ	100000	1011000	265000
2 เงินเดือนค่าจ้างประจำ	2044520	2739890	1788660
3 ค่าจ้างชั่วคราว	1999380	519000	832720
4 ค่าตอบแทนให้สัตยาบันและวัสดุ	2898744	9369602	6144708
5 เงินอุดหนุน	2927000	2549240	2914260
6 ค่าวัสดุที่ได้นับและสิ่ง ยืมสร้าง	993000	398859	3682670
7 ราคาค่าอื่นๆ	5881836	3105708	794182

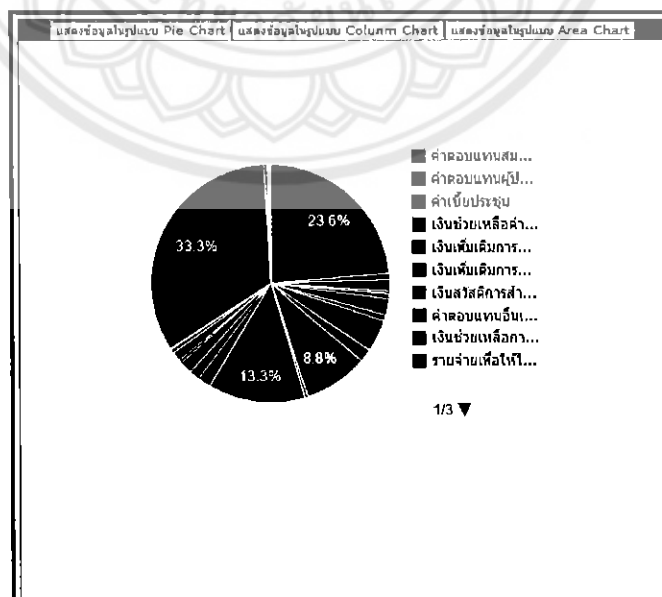
รูปที่ 4.29 แสดงการนำเสนอข้อมูลรายการรายจ่ายแบบตาราง

รูปที่ 4.29 เป็นการสรุปรายจ่ายตามรายการของรายจ่ายในรูปแบบของตาราง โดยนำเสนอจำนวนเงินรายจ่ายตั้งแต่ปีงบประมาณ 2552 ถึง 2554 ซึ่งข้อมูลรายการรายจ่ายแต่ละแถว สามารถเชื่อมโยงไปยังรายละเอียดของรายการรายจ่ายแต่ละรายการได้ ดังรูปที่ 4.30

แสดงรายจ่ายค่าตอบแทนใช้สอยและวัสดุ	
รายการ	งบประมาณ 2554
1 ค่าตอบแทนสมาชิกสภา อปท.	1357800
2 ค่าตอบแทนผู้มีอิทธิพลราชการขึ้นเป็นประโยชน์ต่อ อปท.	49200
3 ค่าตอบแทนการออกบริการงานแพทย์ฉุกเฉิน	1400
4 ค่าเบี้ยประชุม	88000
5 ค่าตอบแทนการปฏิบัติงานนอกเวลาราชการ	5000
6 เงินช่วยเหลือการศึกษาบุตร	5000
7 เงินช่วยเหลือค่ารักษาพยาบาล	20000
8 เงินเพิ่มเติมการครองชีพชั่วคราว พนักงานส่วนตำบล	41940
9 เงินเพิ่มเติมการครองชีพชั่วคราวของพนักงานจ้าง	125100
10 เงินสวัสดิการสำหรับผู้ปฏิบัติงานประจำสำนักงานพื้นที่พิเศษ	48000
11 ค่าตอบแทนอื่นเป็นกรณีพิเศษ	250000
12 เงินช่วยเหลือการดำรงชีวิตประจำวันของเด็กนักเรียนด้อยคุณภาพ	100000
13 รายจ่ายเพื่อให้อำนาจบริหาร	503048
14 รายจ่ายเกี่ยวกับการรับรองและพิธีการ	22000
15 รายจ่ายเนื่องจากการปฏิบัติงานราชการที่เข้าลักษณะรวมรายจ่ายอื่นๆ	761000
16 ค่าบำรุงรักษาและซ่อมแซม	100000
17 ประกันทรัพย์สินสำนักงาน	80000
18 วัสดุไฟฟ้าและวิทยุ	2000
19 วัสดุงานบ้านงานครัว	5000
20 วัสดุเชื้อเพลิงและหล่อลื่นสำหรับรถยนต์และรถจักรยานยนต์	100000
21 วัสดุเชื้อเพลิงและหล่อลื่นสำหรับรถแทรกเตอร์	3000
22 วัสดุโฆษณาและเผยแพร่	5000
23 วัสดุคอมพิวเตอร์	20000
24 วัสดุวิทยาศาสตร์หรือการแพทย์	75000
25 วัสดุการเกษตร	30000
26 วัสดุสิ้นเปลือง	10000
27 วัสดุอาหารเสริม(นม)	1911380
28 วัสดุการศึกษา	23400
29 วัสดุงานบ้านงานครัวในศูนย์พัฒนาเด็กเล็ก	1000

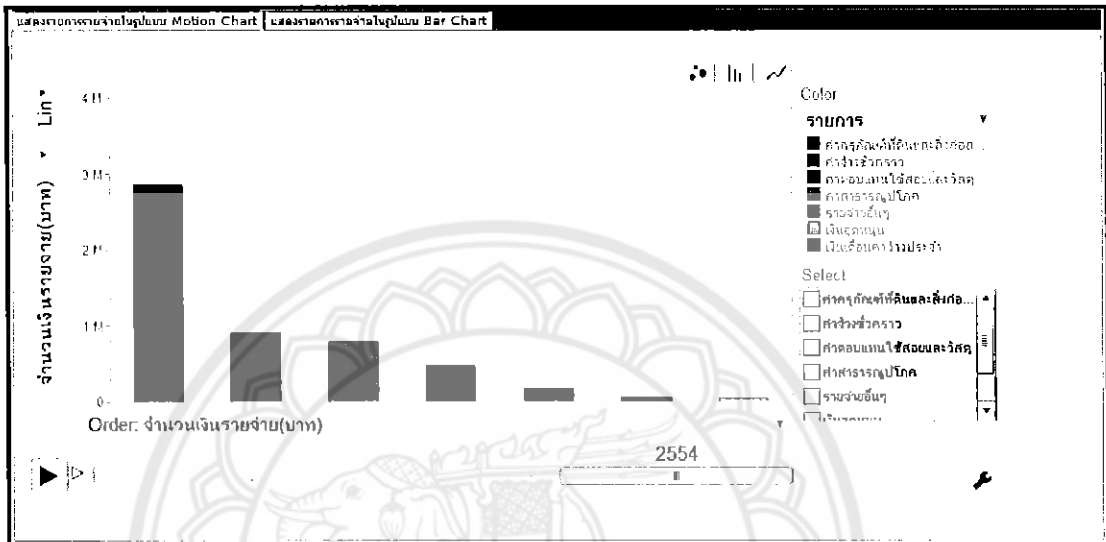
รูปที่ 4.30 แสดงการนำเสนอข้อมูลรายจ่ายค่าตอบแทนใช้สอยและวัสดุแบบตาราง

รูปที่ 4.30 แสดงรายละเอียดของรายจ่ายค่าตอบแทนใช้สอยและวัสดุ ซึ่งละเอียดค่าใช้จ่ายนี้จะถูกนำเสนอในรูปแบบของตาราง การแสดงข้อมูลรายละเอียดนี้จะช่วยให้ผู้ใช้สามารถตรวจสอบความถูกต้องในการใช้จ่ายงบประมาณรายจ่ายได้



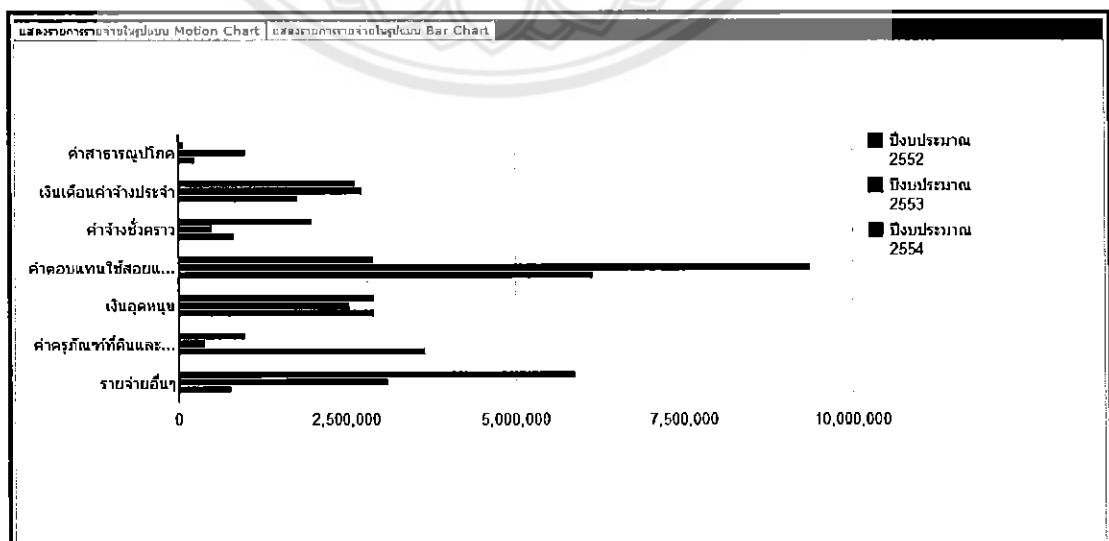
รูปที่ 4.31 แสดงการนำเสนอข้อมูลรายจ่ายค่าตอบแทนใช้สอยและวัสดุด้วย Pie Chart

รูปที่ 4.31 แสดงการนำเสนอข้อมูลรายละเอียดของรายจ่ายด้วยกราฟ ซึ่งจะนำเสนอด้วย Pie Chart เป็นกราฟเริ่มต้น และสามารถเลือกการนำเสนอแบบ Column Chart หรือ Area Chart ได้ การนำเสนอรายการรายจ่าย นอกจากจะนำเสนอข้อมูลเป็นตารางดังที่กล่าวมาข้างต้นแล้ว (รูปที่ 4.29) ยังมีการนำเสนอในรูปแบบของ Motion Chart ทำให้ผู้ใช้สามารถเลือกรูปแบบการนำเสนอได้มากขึ้น และช่วยให้การนำเสนอมีความน่าสนใจยิ่งขึ้น



รูปที่ 4.32 แสดงการนำเสนอข้อมูลรายการรายจ่ายด้วย Motion Chart

รูปที่ 4.32 แสดงการนำเสนอข้อมูลรายการรายจ่ายด้วย Motion Chart แบบ Column Chart โดยรายการรายจ่ายแต่ละรายการจะแสดงข้อมูลแทนด้วยสีที่แตกต่างกัน และสามารถเลือกแสดงข้อมูลในปริมาณประมาณต่างๆ ที่กำหนดไว้ได้

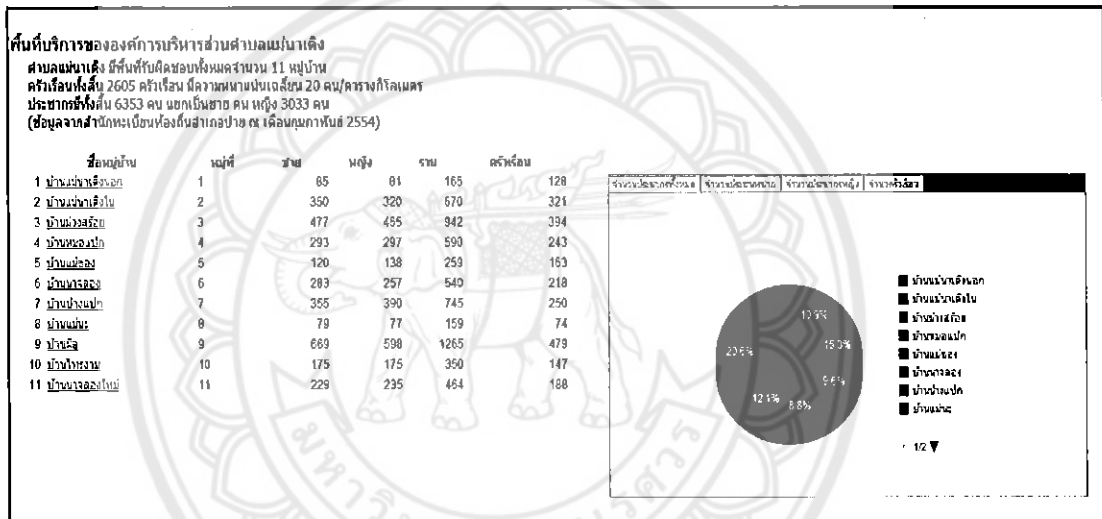


รูปที่ 4.33 แสดงการนำเสนอข้อมูลรายการรายจ่ายด้วย Column Chart

รูปที่ 4.33 แสดงการนำเสนอข้อมูลรายการรายจ่ายด้วย Column Chart การนำเสนอด้วย Column Chart นี้จะช่วยนำเสนอข้อมูลสำหรับเปรียบเทียบการใช้จ่ายแต่ละรายการของปีงบประมาณทั้ง 3 ปีได้ ทำให้เห็นได้ว่าการใช้จ่ายงบประมาณปีใดมีความผิดปกติจากปีงบประมาณอื่นๆ

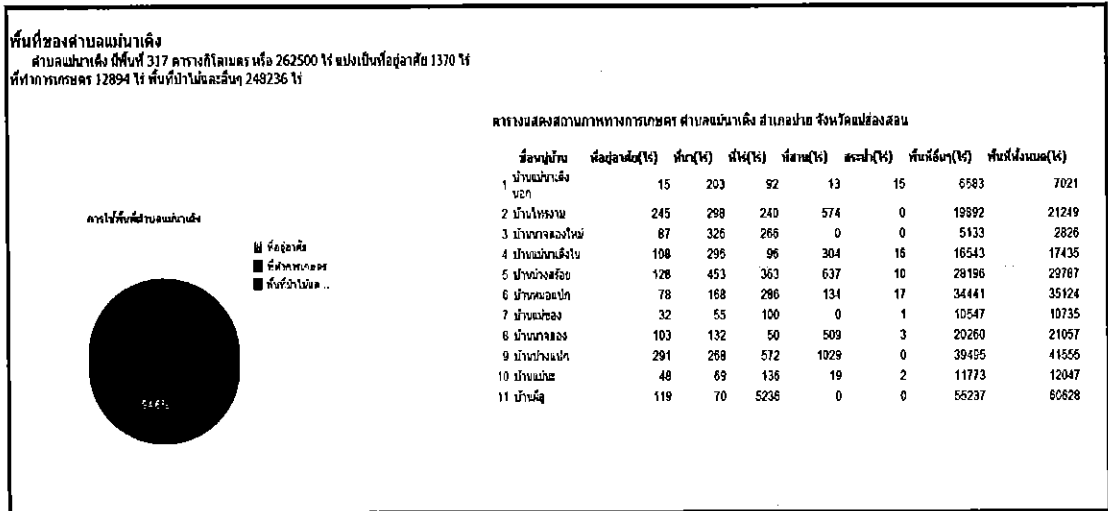
#### 4.3.4 พื้นที่บริการ

การนำเสนอพื้นที่บริการขององค์การบริหารส่วนตำบลนั้น เป็นการนำเสนอเพิ่มเติมนอกเหนือจากงบประมาณการเงิน เพื่อแสดงให้เห็นว่าองค์การบริหารส่วนตำบลแม่มาตังนั้นมีหมู่บ้านที่ต้องให้บริการก็หมู่บ้าน มีประชากรและหลังคาเรือนจำนวนเท่าใด เพื่อเป็นส่วนประกอบในการตัดสินใจวางแผนค่าใช้จ่าย



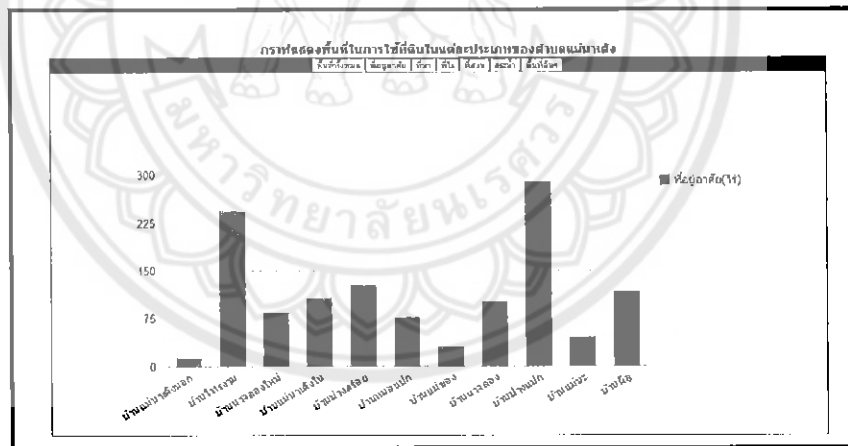
รูปที่ 4.34 แสดงการนำเสนอข้อมูลพื้นที่บริการของ อบต.แม่มาตัง

รูปที่ 4.34 แสดงการนำเสนอข้อมูลหมู่บ้าน จำนวนประชากร และจำนวนครัวเรือนในพื้นที่บริการขององค์การบริหารส่วนตำบลแม่มาตัง โดยแสดงในรูปแบบของตารางและแสดงในรูปแบบของ Pie Chart ของข้อมูลจำนวนประชากรทั้งหมด ข้อมูลประชากรชายในแต่ละหมู่บ้าน ข้อมูลประชากรหญิงในแต่ละหมู่บ้าน และข้อมูลจำนวนหลังคาเรือนในแต่ละหมู่บ้าน



รูปที่ 4.35 แสดงการนำเสนอข้อมูลพื้นที่ของตำบลแม่नाเติง

รูปที่ 4.35 แสดงการนำเสนอข้อมูลพื้นที่ของตำบลแม่नाเติง เมื่อทำการคลิกรายชื่อหมู่บ้านใด หมู่บ้านหนึ่งในตารางการนำเสนอพื้นที่บริการขององค์การบริหารส่วนตำบลแม่नाเติง (รูปที่ 4.34) ระบบจะทำการนำเสนอข้อมูลเกี่ยวกับที่ดินที่ในแต่ละหมู่บ้าน ใช้



รูปที่ 4.36 แสดงการนำเสนอข้อมูลพื้นที่ของตำบลแม่नाเติงด้วย Column Chart

รูปที่ 4.36 เป็นการนำเสนอข้อมูลด้วย Column Chart เพื่อเปรียบเทียบการใช้ที่ดินแต่ละหมู่บ้าน โดยหน้าแรกจะแสดงที่ดินทั้งหมดเปรียบเทียบกัน หน้าที่สองแสดงที่อยู่อาศัยของแต่ละหมู่บ้านเปรียบเทียบกัน หน้าสามแสดงที่นาของแต่ละหมู่บ้านเปรียบเทียบกัน หน้าสี่แสดงที่ไร่ของแต่ละหมู่บ้านเปรียบเทียบกัน หน้าห้าแสดงพื้นที่สวนของแต่ละหมู่บ้านเปรียบเทียบกัน หน้าหกแสดงพื้นที่สระน้ำของแต่ละหมู่บ้านเปรียบเทียบกัน และหน้าที่เจ็ดแสดงพื้นที่อื่นๆ ของแต่ละหมู่บ้านเปรียบเทียบกัน



#### 4.4 Contact me page

หน้า Contact me จะเป็นการแสดงข้อมูลเบื้องต้นเกี่ยวกับผู้จัดทำโครงการ และสามารถเชื่อมโยงไปยังหน้าหลักอื่นๆ คือ หน้า Home หน้า Population และหน้า Finance



รูปที่ 4.37 แสดงการนำเสนอข้อมูลในหน้า Contact me

รูปที่ 4.37 แสดงการนำเสนอข้อมูลในหน้า Contact me ซึ่งประกอบไปด้วย ชื่อโครงการ ชื่อผู้จัดทำโครงการ ที่ปรึกษาโครงการ สาขาที่เรียน คณะ และปีการศึกษา

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 สรุปผลการดำเนินงาน

จากการดำเนินโครงการการนำเสนอข้อมูลเชิงวิเคราะห์ ระบบสามารถทำการเชื่อมต่อกับฐานข้อมูลด้วยวิธีการ LINQ to SQL และคิวรีข้อมูลที่ต้องการเพื่อนำมาทำการนำเสนอ และนำหลักการ Pivot เข้ามาช่วยในการนำเสนอข้อมูลบางประเภท แล้วทำการนำเสนอข้อมูลด้วยรูปแบบการนำเสนอข้อมูลจาก Google Visualization API ซึ่งผลการทดลองจากบทที่ 4 ที่ผ่านมามีความต้องการตามเป้าหมาย และตรงตามจุดประสงค์ที่ตั้งไว้ การนำเสนอข้อมูลด้วยการเรียกใช้ API นี้มีหลายที่ที่เปิดบริการให้สามารถเรียกใช้ได้ โดยรูปแบบการนำเสนอที่น่าสนใจอีกแบบ คือ Many Eyes เป็นรูปแบบการนำเสนอข้อมูลที่เปิดให้บริการ API เช่นเดียวกับ Google Visualization API

ตารางที่ 5.1 เปรียบเทียบรูปแบบการนำเสนอของ Google Visualization API กับ Many Eyes

คุณสมบัติ	Google Visualization API	Many Eyes
ผู้ให้บริการ	Google	IBM
รูปแบบการนำเสนอ	รูปแบบการนำเสนอเรียบง่าย ปรับแต่งรูปแบบได้ และมีการ โต้ตอบกับผู้ใช้	รูปแบบการนำเสนอมีความ หลากหลาย มีลูกเล่นการ นำเสนอที่ดีกว่า และมีการ โต้ตอบกับผู้ใช้
ค่าใช้จ่าย	ฟรี	ฟรีเฉพาะรูปแบบที่อนุญาต
อายุซอฟต์แวร์	ไม่มีกำหนด	5 ปี
ความสามารถในการแสดงผลบนเว็บ	Image & Live Visualization	Image & Live Visualization
ปลั๊กอินที่จำเป็น	ไม่จำเป็นต้องติดตั้งปลั๊กอิน รองรับเทคโนโลยี HTML5 สามารถทำงานข้าม แพลตฟอร์มได้	Java หรือ Flash ขึ้นอยู่กับ รูปแบบการนำเสนอ

## 5.2 ปัญหาที่พบ

จากการทำโครงการครั้งนี้ มีอุปสรรคและปัญหาของการทำงาน คือ

1. การดำเนินการของโครงการ จากการศึกษาข้อมูลการใช้รูปแบบการนำเสนอข้อมูลด้วย Google Visualization API ในอินเทอร์เน็ตพบว่า การนำเสนอข้อมูลด้วยรูปแบบการนำเสนอจาก Google Visualization API ข้อมูลที่ถูกนำเสนอด้วยภาษาไทยไม่สามารถทำการนำเสนอข้อมูลได้ จึงทำการบันทึกข้อมูลในฐานะข้อมูลเป็นภาษาอังกฤษ แต่เมื่อปรึกษากับเพื่อนที่มีประสบการณ์ในการใช้เจสันในการรับส่งข้อมูล แล้วพบว่า การดึงข้อมูลด้วยเจสันนั้นต้องทำการแปลงข้อมูลก่อนเพื่อให้เจสันสามารถรับส่งข้อมูลเป็นภาษาไทยได้ ทำให้ต้องทำการบันทึกข้อมูลลงในฐานข้อมูลใหม่เป็นภาษาไทย และออกแบบหน้าเว็บแอปพลิเคชันใหม่ให้เหมาะสมกับข้อมูลที่จะนำเสนอเป็นภาษาไทย

2. การนำรูปแบบการนำเสนอจาก Google Visualization API ประเภท Geo Chat นั้นซึ่งเหมาะกับการนำเสนอข้อมูลเพื่อแสดงความหนาแน่นของประชากร ไม่สามารถนำ Geo Chart ที่แสดงข้อมูลในระดับจังหวัดมาใช้ในการนำเสนอได้ เพราะข้อมูลที่นำเสนอเป็นข้อมูลในระดับตำบลและอำเภอ

3. การดำเนินโครงการมีการปรับเปลี่ยนขอบเขตและรูปแบบการทำงานของระบบ อยู่บ่อยครั้ง เพื่อความเหมาะสมของโครงการ แต่ก็ส่งผลให้เกิดความล่าช้าในการดำเนินโครงการ

## 5.3 แนวทางแก้ไขและข้อเสนอแนะ

1. การทดสอบและตรวจสอบความถูกต้องของระบบ ควรตรวจสอบอย่างความละเอียดและรอบคอบ เพื่อป้องกันไม่ให้เกิดข้อผิดพลาด หรือความคลาดเคลื่อนของข้อมูลที่อาจเกิดขึ้นได้ในการนำเสนอ

2. การศึกษาหาข้อมูลจากอินเทอร์เน็ตอย่างเดียว อาจทำให้ได้รับข้อมูลบางอย่างที่คลาดเคลื่อนหรือไม่ถูกต้อง ดังนั้นเมื่อมีการศึกษาข้อมูลทางอินเทอร์เน็ต ควรจะศึกษาจากเว็บไซต์ที่มีความน่าเชื่อถือ หรือศึกษาจากหลายๆ เว็บไซต์แล้วนำข้อมูลมาพิจารณาไตร่ตรองและอิงกับทฤษฎีที่ศึกษามา

3. การนำรูปแบบการนำเสนอจาก Google Visualization API ประเภท Geo Chat มาใช้ระดับอำเภอหรือตำบล จำเป็นต้องใช้ Geo Chart อีกแบบที่เป็นแบบ markers ซึ่งจะแสดงเป็นลักษณะของวงกลมแถบสีบนแผนที่แทนเส้นแบ่งขอบเขตพื้นที่โดยตรง โดยจัดเก็บข้อมูลเป็นละติจูดและลองจิจูด เพื่อให้สามารถแสดงจุดวงกลมในตำแหน่งที่ทำการได้ ข้อดีคือ สามารถแสดงความหนาแน่นของประชากรในระดับอำเภอหรือตำบลได้ ข้อเสียคือ การแสดงจุดวงกลมความหนาแน่นอาจอยู่ติดกันจนทำให้ดูข้อมูลลำบาก

4. ระบบควรทำการติดตั้งโปรแกรมบริการ (Server) เพื่อให้เครื่องลูกข่าย (Client) สามารถเข้ามาใช้บริการได้และเกิดการใช้งานจริง

5. การดำเนินโครงการหรือการทำงานต่างๆ ย่อมมีการปรับเปลี่ยนหรือเปลี่ยนแปลงตามความเหมาะสม ดังนั้นควรจะมีการประเมินงานเป็นระยะๆ เพื่อปรับแผนการทำงานให้สอดคล้องกับสถานการณ์ที่เกิดขึ้น ซึ่งจะช่วยให้การทำงานและประสบความสำเร็จภายในระยะเวลาที่กำหนดไว้



## เอกสารอ้างอิง

- [1] “Google Chart Tools”. [online]. Available :  
<https://developers.google.com/chart/interactive/docs/index?hl=th-TH>. 2554.
- [2] “Ajax Tutorial”. [online]. Available : <http://www.thaicreate.com/tutorial/ajax.html>.  
 2554
- [3] “Social Visualization Software Review”. [online]. Available :  
[http://infosthetics.com/archives/2010/04/social\\_visualization\\_software\\_review\\_many\\_eyes.html](http://infosthetics.com/archives/2010/04/social_visualization_software_review_many_eyes.html). 2554
- [4] “Introducing JSON”. [online]. Available :  
<http://james.newtonking.com/projects/json/help/>. 2554
- [5] “Google code”. [online]. Available :  
[http://code.google.com/apis/ajax/playground/?type=visualization#dependent\\_controls](http://code.google.com/apis/ajax/playground/?type=visualization#dependent_controls)
- [6] ลากลอย วานิชอังกูร.(2552). **เรียนรู้ด้วยตนเอง Database/Query/Stored Procedure**.  
 กรุงเทพมหานคร : บริษัทซีเอ็คยูเคชั่น จำกัด(มหาชน).
- [7] ชีรเดช รักไทย.(2553). **เก่ง C# ให้ครบสูตรฉบับ OOP**. กรุงเทพมหานคร : สำนักพิมพ์  
 วิตตี้กรุ๊ป

## ภาคผนวก

### Google Chart Gallery

#### Google Visualization API

การใช้รูปแบบการนำเสนอของ Google visualization API นั้น ต้องมีการติดต่อกับ Google ด้วยจาวาสคริปต์เพื่อร้องขอรูปแบบการนำเสนอจาก Google ซึ่งรูปแบบแต่ละรูปแบบของการนำเสนอจะมีการใช้จาวาสคริปต์ในการร้องขอรูปแบบการนำเสนอข้อมูล ดังต่อไปนี้

##### 1. JavaScript Code: Area Chart

```
<html>
<head>
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
var data = google.visualization.arrayToDataTable([
  ['Year', 'Sales', 'Expenses'],
  ['2004', 1000, 400],
  ['2005', 1170, 460],
  ['2006', 660, 1120],
  ['2007', 1030, 540]
]);

var options = {
  title: 'Company Performance',
  hAxis: {title: 'Year', titleTextStyle: {color: 'red'}}
};

var chart = new google.visualization.AreaChart(document.getElementById('chart_div'));
```

```

        chart.draw(data, options);
    }
</script>
</head>
<body>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

## 2. JavaScript Code: Bar Chart

```

<html>
<head>
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>
    <script type="text/javascript">
        google.load("visualization", "1", {packages:["corechart"]});
        google.setOnLoadCallback(drawChart);
        function drawChart() {
            var data = google.visualization.arrayToDataTable([
                ['Year', 'Sales', 'Expenses'],
                ['2004', 1000, 400],
                ['2005', 1170, 460],
                ['2006', 660, 1120],
                ['2007', 1030, 540]
            ]);

            var options = {
                title: 'Company Performance',
                vAxis: {title: 'Year', titleTextStyle: {color: 'red'}}
            };

            var chart = new google.visualization.BarChart(document.getElementById('chart_div'));
            chart.draw(data, options);
        }
    </script>
</head>
<body>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

```

</script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

### 3. JavaScript Code: Bubble Chart

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['ID', 'Life Expectancy', 'Fertility Rate', 'Region', 'Population'],
        ['CAN', 80.66, 1.67, 'North America', 33739900],
        ['DEU', 79.84, 1.36, 'Europe', 81902307],
        ['DNK', 78.6, 1.84, 'Europe', 5523095],
        ['EGY', 72.73, 2.78, 'Middle East', 79716203],
        ['GBR', 80.05, 2, 'Europe', 61801570],
        ['IRN', 72.49, 1.7, 'Middle East', 73137148],
        ['IRQ', 68.09, 4.77, 'Middle East', 31090763],
        ['ISR', 81.55, 2.96, 'Middle East', 7485600],
        ['RUS', 68.6, 1.54, 'Europe', 141850000],
        ['USA', 78.09, 2.05, 'North America', 307007000]
      ]);

      var options = {
        title: 'Correlation between life expectancy, fertility rate and population of some world

```



```

countries (2010)',
    hAxis: {title: 'Life Expectancy'},
    vAxis: {title: 'Fertility Rate'},
    bubble: {textStyle: {fontSize: 11}}
  };

  var chart = new google.visualization.BubbleChart(document.getElementById('chart_div'));
  chart.draw(data, options);
}
</script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

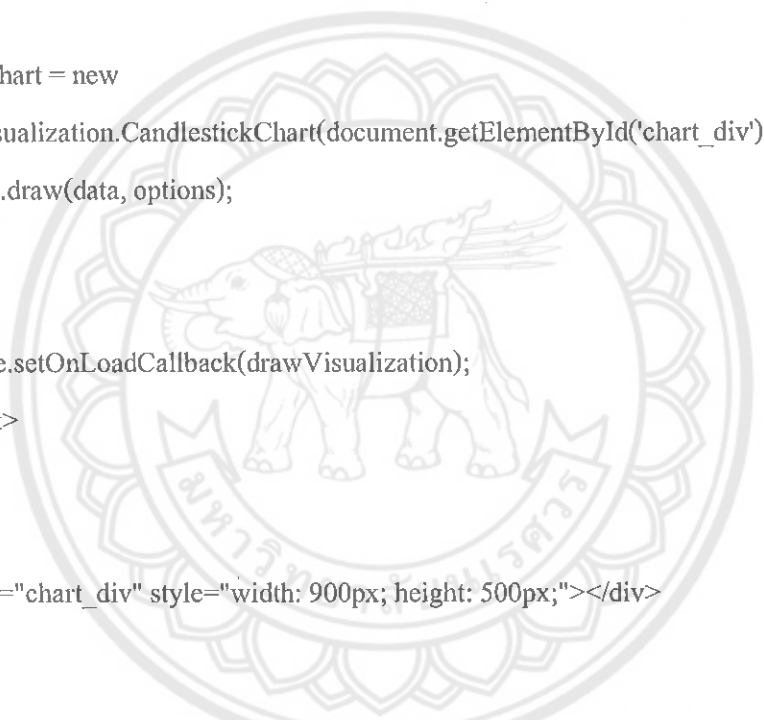
#### 4. JavaScript Code: Candlestick Chart

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  <title>
    Google Visualization API Sample
  </title>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load('visualization', '1', {packages: ['corechart']});
  </script>
  <script type="text/javascript">
    function drawVisualization() {
      var data = google.visualization.arrayToDataTable([
        ['Mon', 20, 28, 38, 45],
        ['Tue', 31, 38, 55, 66],

```

```
    ['Wed', 50, 55, 77, 80],  
    ['Thu', 77, 77, 66, 50],  
    ['Fri', 68, 66, 22, 15]  
    // Treat first row as data as well.  
  ], true);  
  
  var options = {  
    legend:'none'  
  };  
  
  var chart = new  
google.visualization.CandlestickChart(document.getElementById('chart_div'));  
  chart.draw(data, options);  
}  
  
google.setOnLoadCallback(drawVisualization);  
</script>  
</head>  
<body>  
  <div id="chart_div" style="width: 900px; height: 500px;"></div>  
</body>  
</html>
```



## 5. JavaScript Code: Column Chart

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Year', 'Sales', 'Expenses'],
        ['2004', 1000, 400],
        ['2005', 1170, 460],
        ['2006', 660, 1120],
        ['2007', 1030, 540]
      ]);

      var options = {
        title: 'Company Performance',
        hAxis: {title: 'Year', titleTextStyle: {color: 'red'}}
      };

      var chart = new google.visualization.ColumnChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

## 6. JavaScript Code: Combo Chart

```

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>
      Google Visualization API Sample
    </title>
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>
    <script type="text/javascript">
      google.load('visualization', '1', {packages: ['corechart']});
    </script>
    <script type="text/javascript">
      function drawVisualization() {
        // Some raw data (not necessarily accurate)
        var data = google.visualization.arrayToDataTable([
          ['Month', 'Bolivia', 'Ecuador', 'Madagascar', 'Papua New Guinea', 'Rwanda', 'Average'],
          ['2004/05', 165, 938, 522, 998, 450, 614.6],
          ['2005/06', 135, 1120, 599, 1268, 288, 682],
          ['2006/07', 157, 1167, 587, 807, 397, 623],
          ['2007/08', 139, 1110, 615, 968, 215, 609.4],
          ['2008/09', 136, 691, 629, 1026, 366, 569.6]
        ]);

        var options = {
          title : 'Monthly Coffee Production by Country',
          vAxis: {title: "Cups"},
          hAxis: {title: "Month"},
          seriesType: "bars",
          series: {5: {type: "line"}}
        };

```

```

var chart = new google.visualization.ComboChart(document.getElementById('chart_div'));
chart.draw(data, options);
}
google.setOnLoadCallback(drawVisualization);
</script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

## 7. JavaScript Code: Gauge

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load('visualization', '1', {packages:['gauge']});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Label', 'Value'],
        ['Memory', 80],
        ['CPU', 55],
        ['Network', 68]
      ]);
      var options = {
        width: 400, height: 120,
        redFrom: 90, redTo: 100,
        yellowFrom: 75, yellowTo: 90,
        minorTicks: 5
      };

```

```

var chart = new google.visualization.Gauge(document.getElementById('chart_div'));
chart.draw(data, options);
}
</script>
</head>
<body>
  <div id='chart_div'></div>
</body>
</html>

```

### 8. JavaScript Code: Geo Chart

```

<html>
<head>
  <script type='text/javascript' src='https://www.google.com/jsapi'></script>
  <script type='text/javascript'>
    google.load('visualization', '1', {'packages': ['geochart']});
    google.setOnLoadCallback(drawRegionsMap);

    function drawRegionsMap() {
      var data = google.visualization.arrayToDataTable([
        ['Country', 'Popularity'],
        ['Germany', 200],
        ['United States', 300],
        ['Brazil', 400],
        ['Canada', 500],
        ['France', 600],
        ['RU', 700]
      ]);

      var options = {};

      var chart = new google.visualization.GeoChart(document.getElementById('chart_div'));

```

```

    chart.draw(data, options);
  };
</script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

### 9. JavaScript Code: Line Chart

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Year', 'Sales', 'Expenses'],
        ['2004', 1000, 400],
        ['2005', 1170, 460],
        ['2007', 1030, 540] ]);
      var options = { title: 'Company Performance' };
      var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

## 10. JavaScript Code: Pie Chart

```
<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Task', 'Hours per Day'],
        ['Work', 11],
        ['Eat', 2],
        ['Commute', 2],
        ['Watch TV', 2]
      ]);
      var options = {
        title: 'My Daily Activities'
      };
      var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>
```



## 11. JavaScript Code: Scatter Chart

```
<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Age', 'Weight'],
        [ 8, 12],
        [ 4, 5],
        [ 3, 3.5],
        [ 6.5, 7]
      ]);

      var options = {
        title: 'Age vs. Weight comparison',
        hAxis: {title: 'Age', minValue: 0, maxValue: 15},
        vAxis: {title: 'Weight', minValue: 0, maxValue: 15},
        legend: 'none'
      };

      var chart = new google.visualization.ScatterChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>
```

## 12. JavaScript Code: Stepped Area Chart

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Director (Year)', 'Rotten Tomatoes', 'IMDB'],
        ['Alfred Hitchcock (1935)', 8.4, 7.9],
        ['Ralph Thomas (1959)', 6.9, 6.5],
        ['James Hawes (2008)', 4.4, 6.2]
      ]);

      var options = {
        title: 'The decline of \'The 39 Steps\'',
        vAxis: {title: 'Accumulated Rating'},
        isStacked: true
      };

      var chart = new
google.visualization.SteppedAreaChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>

```

### 13. JavaScript Code: Table

```
<html>
<head>
  <script type='text/javascript' src='https://www.google.com/jsapi'></script>
  <script type='text/javascript'>
    google.load('visualization', '1', {packages:['table']});
    google.setOnLoadCallback(drawTable);
    function drawTable() {
      var data = new google.visualization.DataTable();
      data.addColumn('string', 'Name');
      data.addColumn('number', 'Salary');
      data.addColumn('boolean', 'Full Time Employee');
      data.addRows([
        ['Mike', {v: 10000, f: '$10,000'}, true],
        ['Jim', {v: 8000, f: '$8,000'}, false],
        ['Alice', {v: 12500, f: '$12,500'}, true],
        ['Bob', {v: 7000, f: '$7,000'}, true]
      ]);

      var table = new google.visualization.Table(document.getElementById('table_div'));
      table.draw(data, {showRowNumber: true});
    }
  </script>
</head>

<body>
  <div id='table_div'></div>
</body>
</html>
```

#### 14. JavaScript Code: TreeMap

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["treemap"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      // Create and populate the data table.
      var data = google.visualization.arrayToDataTable([
        ['Location', 'Parent', 'Market trade volume (size)', 'Market increase/decrease (color)'],
        ['Global', null, 0, 0],
        ['America', 'Global', 0, 0],
        ['Europe', 'Global', 0, 0],
        ['Asia', 'Global', 0, 0],
        ['Australia', 'Global', 0, 0],
        ['Africa', 'Global', 0, 0],
        ['Brazil', 'America', 11, 10],
        ['USA', 'America', 52, 31],
        ['Mexico', 'America', 24, 12],
        ['Canada', 'America', 16, -23],
        ['France', 'Europe', 42, -11],
        ['Zair', 'Africa', 8, 10]
      ]);

      // Create and draw the visualization.
      var tree = new google.visualization.TreeMap(document.getElementById('chart_div'));
      tree.draw(data, {
        minColor: '#f00',
        midColor: '#ddd',
        maxColor: '#0d0',
        headerHeight: 15,

```

```
        fontColor: 'black',
        showScale: true});
    }
</script>
</head>

<body>
  <div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>
```



## ประวัติผู้จัดทำโครงการ



ชื่อ นางสาว โสณิยา พัวทา  
 ภูมิลำเนา 34 หมู่ 5 ต.บ้านดิว อ.ห่มล็ก จ.เพชรบูรณ์  
 ประวัติการศึกษา  
 – จบระดับมัธยมศึกษาจากโรงเรียนตั่ววิทยาคม  
 – ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4  
 สาขาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยนเรศวร

E-mail: krachao4444@hotmail.com

