

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

สถานีตรวจอากาศบนแผนที่กูเกิลแมพ

Weather Stations On Google Maps



๒๗
๐๘๗๕๕
๒๕๕๐

เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี...=4...ก.ค. 2551

b. 11943397
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานีตรวจอากาศบนแผนที่กูเกิลแมพ
Weather Stations On Google Maps



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมคอมพิวเตอร์

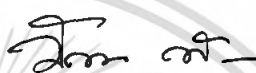
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง สถานีตรวจอากาศบนแผนที่กูเกิลแมพ

Weather Stations On Google Maps

ผู้จัดทำ

นายอัครเดช คนสมบูรณ์ รหัสนักศึกษา 43010537



อาจารย์ที่ปรึกษา

(อ. บัณฑิต พัสยา)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานีตรวจอากาศบนแผนที่ภูเกิลแมพ

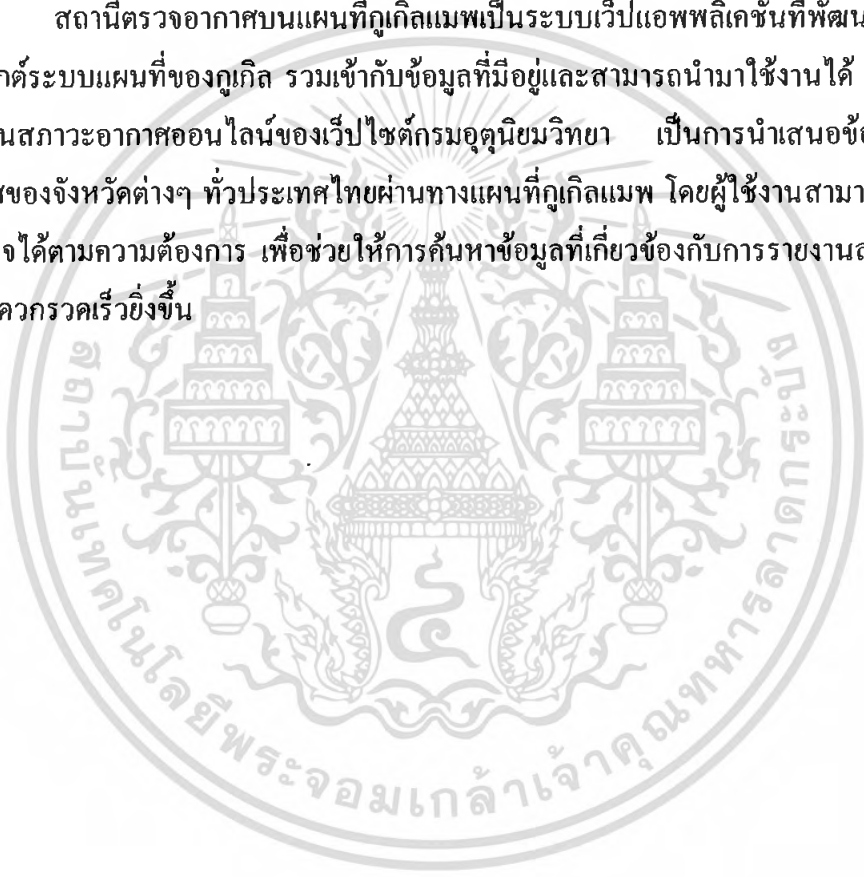
นายอัครเดช กนสมบูรณ์ 43010537

อาจารย์บัณฑิต พัสยา อาจารย์ที่ปรึกษา

ปีการศึกษา 2550

บทคัดย่อ

สถานีตรวจอากาศบนแผนที่ภูเกิลแมพเป็นระบบเว็บแอปพลิเคชันที่พัฒนาขึ้น โดยมีการประยุกต์ระบบแผนที่ของกูเกิล รวมเข้ากับข้อมูลที่มีอยู่และสามารถนำมาใช้งานได้ ในที่นี้คือระบบรายงานสภาวะอากาศออนไลน์ของเว็บไซต์กรมอุตุนิยมวิทยา เป็นการนำเสนอข้อมูลสถานีตรวจอากาศของจังหวัดต่างๆ ทั่วประเทศไทยผ่านทางแผนที่ภูเกิลแมพ โดยผู้ใช้งานสามารถเลือกดูข้อมูลที่น่าสนใจได้ตามความต้องการ เพื่อช่วยให้การค้นหาข้อมูลที่เกี่ยวข้องกับการรายงานสภาวะอากาศทำได้สะดวกรวดเร็วยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Weather Stations On Google Maps

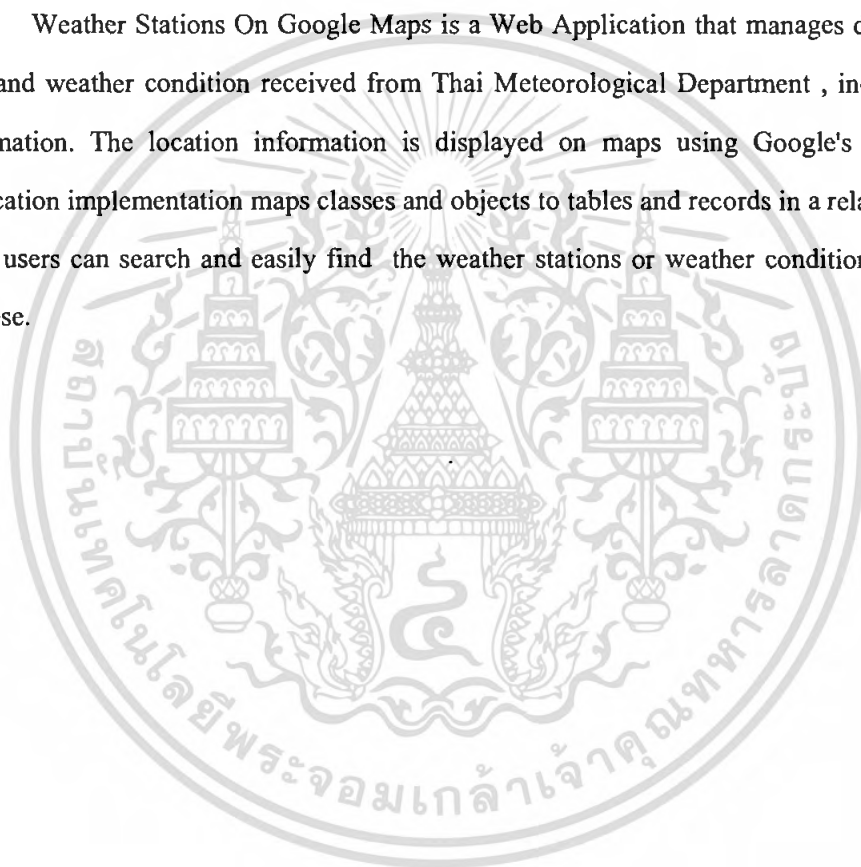
Mr.Akkradet Konsomboon 43010537

Mr.Bundit Passaya Advisor

Academic Year 2007

ABSTRACT

Weather Stations On Google Maps is a Web Application that manages data on report of Thailand weather condition received from Thai Meteorological Department , including location information. The location information is displayed on maps using Google's Map API. The application implementation maps classes and objects to tables and records in a relational database. Then users can search and easily find the weather stations or weather condition that match his purpose.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยคำแนะนำและคำปรึกษาจาก อ. บัณฑิต พัสยา ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญาานิพนธ์ ข้าพเจ้ารู้สึกซาบซึ้งในคำแนะนำ ความอนุเคราะห์จากท่านและขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

นายอัทรเดช คนสมบูรณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ	
1.1 หลักการและเหตุผล.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	3
1.4 วิธีการดำเนินการ.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 ส่วนประกอบของโครงการ.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 ภาษา PHP.....	5
2.1.1 ลักษณะของ PHP.....	5
2.1.2 ข้อดีของการเลือกใช้ PHP.....	5
2.2 ระบบฐานข้อมูล MySQL.....	6
2.2.1 ความหมายของระบบฐานข้อมูล MySQL.....	6
2.2.2 ความสามารถและการทำงานของ MySQL.....	6
2.3 ภาษา SQL.....	7
2.3.1 คำสั่งในการคิวรีเพื่อแสดงข้อมูล.....	7
2.3.2 การจำกัดผลลัพธ์ ของแถวข้อมูลที่เหมือนกัน.....	7
2.3.3 การเลือกแถวที่รู้ข้อมูลเพียงบางส่วน.....	7
2.3.4 การเลือกแถวโดยกำหนดหลายๆ เงื่อนไขร่วมกัน.....	7
2.3.5 การเรียงลำดับผลลัพธ์ของข้อมูลที่ถูกนำมาแสดง.....	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4 AJAX.....	8
2.4.1 ประวัติของ AJAX.....	8
2.4.2 หลักการทำงาน.....	9
2.4.3 ข้อดีของ AJAX.....	10
2.4.4 ปัญหาที่พบ.....	10
2.5 XML (eXtensible Markup Language)	11
2.5.1 ไวยากรณ์ทั่วไปของ XML.....	11
2.5.2 Tag, Text, Element	12
2.5.3 XML Namespaces.....	15
2.5.4 DTD และ XML Schema	17
2.5.5 การแปลงรูปแบบเอกสาร XML ด้วย XSLT.....	21
2.6 Javascript.....	21
2.6.1 ลักษณะของ Javascript.....	21
2.6.2 การทำงานของจาวาสคริปต์.....	22
2.7 Document Object Model (DOM).....	22
2.8 CronTab.....	23
2.9 cURL.....	25
2.9.1 ความหมายของ cURL.....	25
2.9.2 การติดตั้ง cURL.....	25
2.9.3 คำสั่งในการเรียกใช้งาน cURL	25

บทที่ 3 การใช้งาน Google Maps

3.1 ความรู้ที่ควรมีในการใช้งาน Google Maps API.....	26
3.1.1 ความรู้พื้นฐานเบื้องต้น.....	26
3.1.2 ความรู้เพิ่มเติมที่จำเป็นสำหรับงานพัฒนาขั้นกลาง- สูง.....	26
3.1.3 ความรู้เพิ่มเติมสำหรับงานใหญ่ งานพิเศษ.....	26
3.1.4 ความรู้อื่นที่เกี่ยวข้อง มีทั้งที่เป็นเรื่องกราฟิก แผนที่ และ อื่นๆ.....	27
3.2 ขั้นตอนหลักในการดำเนินการจัดทำระบบแผนที่ Google Maps.....	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3	การใช้งานคำสั่งที่สำคัญใน Google Maps API พร้อมทั้งตัวอย่าง.....	28
3.3.1	การกำหนดให้มีตัวควบคุมบนแผนที่.....	30
3.3.2	ตัวรับฟังอุบัติการณ์ (Event Listener).....	31
3.3.3	การเปิด Info Window.....	31
3.3.4	วัตถุซ้อนทับบนแผนที่ (Map Overlays).....	32
3.3.5	การตอบสนองต่ออุบัติการณ์ "click".....	33
3.3.6	การเปิด Info Window กำกับ Marker เพื่อแสดงข้อมูลแผนที่.....	34
3.3.7	Info Window แบบหลายแผ่นเอกสารซ้อนกัน.....	36
3.3.8	การสร้างสัญลักษณ์ (Icons).....	37
3.3.9	การใช้ Icon Class.....	38
3.3.10	การใช้ XML และ Asynchronous HTTP เพื่อนำเข้าข้อมูลแผนที่.....	40
3.3.11	ตัวควบคุมแผนที่แบบประดิษฐ์เอง (Custom Map Controls).....	41
3.3.12	วัตถุวางบนแผนที่แบบทำการเฉพาะ (Custom Overlays).....	43
3.3.13	การแสดงผลข้อมูลแผนที่จากแฟ้มข้อมูล KML/ GeoRSS.....	46
3.3.14	ภาพวางทับบนแผนที่ (Ground Overlay).....	47
บทที่ 4	การออกแบบและพัฒนา	
4.1	การออกแบบฐานข้อมูล.....	48
4.2	ภาพรวมของระบบ.....	50
4.3	โปรแกรมส่วนที่ทำหน้าที่ดึงข้อมูล.....	52
4.3.1	ส่วนดึงข้อมูล.....	52
4.3.2	ส่วนจัดเก็บข้อมูลลงฐานข้อมูล.....	54
4.4	โปรแกรมส่วนที่ทำหน้าที่จัดการฐานข้อมูล.....	57
4.5	เว็บแอปพลิเคชัน.....	57
4.5.1	ส่วนแสดงแผนที่.....	58
4.5.2	แบบฟอร์มในการรับค่าตัวรีจากผู้ใช้งาน.....	59
4.5.3	Sidebar Output.....	59
4.5.4	กระบวนการสร้างไฟล์ XML ส่งให้แผนที่.....	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 การทดลองและผลการทดลอง	
5.1 อินเทอร์เน็ตของส่วนของเว็บแอปพลิเคชัน.....	61
5.1.1 ส่วนแผนที่.....	61
5.1.2 ส่วนค้นหาข้อมูล.....	61
5.1.3 ส่วนแสดงผลข้อมูล.....	62
5.2 การทดสอบการใช้งาน.....	62
5.2.1 การกำหนดค่าที่จะค้นหาจากฐานข้อมูล.....	62
5.2.2 ผลการรันโปรแกรม.....	63
บทที่ 6 สรุปผลและแนวทางการพัฒนาต่อ	
5.1 สรุปผลการดำเนินงาน.....	66
5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข.....	66
5.3 แนวทางในการพัฒนาต่อ.....	67
บรรณานุกรม.....	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 เขตข้อมูลและคำอธิบายของตาราง Region.....	49
4.2 เขตข้อมูลและคำอธิบายของตาราง Station.....	49
4.3 เขตข้อมูลและคำอธิบายของตาราง Weather.....	49



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 เปรียบเทียบการทำงานระหว่างแอปพลิเคชันแบบเดิม กับแบบที่ใช้เอเจ็ทซ์.....	9
2.2 เปรียบเทียบการติดต่อสื่อสารระหว่างแอปพลิเคชันแบบเดิมกับแบบที่ใช้ เอเจ็ทซ์.....	10
2.3 รูปแบบของคำสั่ง crontab ทั้ง 6 ฟیلด์.....	23
3.1 แผนที่ซึ่งเคลื่อนตำแหน่งได้.....	30
3.2 การเรียกใช้ Info Window.....	32
3.3 การใช้ custom info window แสดงผลกำกับ marker.....	36
3.4 การสร้าง Tabbed info window.....	37
4.1 ER Diagram ของระบบ.....	48
4.2 ความสัมพันธ์ของตารางทั้ง 3 ในฐานะข้อมูลระบบ.....	50
4.3 ภาพรวมของระบบ Weather Station On Google Maps.....	51
4.4 ข้อมูล XML ที่ให้บริการ โดยกรมอุตุนิยมวิทยา.....	52
4.5 ไฟล์ data.txt ที่ได้จากการดึงข้อมูล.....	53
4.6 กระบวนการสร้างไฟล์ weather.xml.....	54
4.7 ไฟล์ weather.xml ที่ได้.....	54
4.8 การเรียกใช้งาน SimpleXML.....	56
4.9 ไฟล์ example.php.....	56
4.10 ส่วนประกอบของหน้าจอหลัก.....	58
4.11 ส่วนควบคุมแผนที่.....	58
4.12 การสร้างเอกสาร XML ให้แผนที่.....	60
5.1 หน้าจอหลักของระบบ.....	61
5.2 ตัวเลือกในการเรียกดูข้อมูลสภาพอากาศ.....	62
5.3 ค่าที่ใช้ในการทดลองค้นหาข้อมูลสภาพอากาศ.....	62
5.4 ผลการค้นหาข้อมูลที่แสดงบนแผนที่.....	63
5.5 การซูมเข้าเพื่อดูรายละเอียดบนแผนที่.....	64
5.6 ผลการค้นหาแสดงรายชื่อสถานีที่ค้นพบ.....	64
5.7 ข้อมูลที่อ่านได้จากการคลิกมาร์กเกอร์บนแผนที่.....	65
5.8 ผลการเลือกสถานีจากรายชื่อบน Side Bar.....	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

ปัจจุบันระบบสารสนเทศสมัยใหม่ในชีวิตประจำวันของเรามากขึ้น โดยเฉพาะอินเทอร์เน็ต นั้นเป็นช่องทางที่สำคัญในการรับส่งข่าวสารและข้อมูลสารสนเทศจากแหล่งต่างๆ ระบบสารสนเทศภูมิศาสตร์(GIS) ก็มีการพัฒนารูปแบบการทำงานจากเดิมที่เป็นระบบแบบ Standalone มาเป็นระบบ Internet GIS เพื่อจุดมุ่งหมายที่จะเผยแพร่ข้อมูลเชิงพื้นที่ ทำให้ผู้ใช้สามารถเข้าถึงข้อมูลได้สะดวกมากขึ้น

Internet GIS เป็นระบบ GIS แบบใหม่ที่กำลังได้รับความนิยมอยู่ในปัจจุบัน การทำงานจะอาศัยระบบอินเทอร์เน็ตเป็นหลัก โดยมีรูปแบบสถาปัตยกรรมแบบ 3 เทียร์ กล่าวคือผู้ใช้จะทำงานอยู่ที่เครื่องคอมพิวเตอร์บนฝั่งลูกข่าย ซึ่งจะทำการส่งคำสั่งไปยัง Application ที่อยู่บนฝั่งแม่ข่ายผ่าน http protocol แม่ข่ายจะทำการประมวลผลคำสั่งและส่งผลลัพธ์กลับมายังลูกข่าย การทำงานจะอาศัยทรัพยากรของเครื่องแม่ข่ายเป็นหลัก โดยเครื่องคอมพิวเตอร์ลูกข่ายที่ทำงานในระบบอาจจะไม่จำเป็นต้องมีสมรรถนะที่สูงมากนักก็สามารถทำงานได้ แต่เนื่องจากการทำงานจะต้องพึ่งพา ระบบอินเทอร์เน็ตในการรับส่งข้อมูลและคำสั่งระหว่างลูกข่ายกับแม่ข่าย ดังนั้นระบบอินเทอร์เน็ตจึงจำเป็นที่จะต้องมีประสิทธิภาพที่ดีจึงจะสามารถทำงานได้อย่างมีประสิทธิภาพ

หลังจากที่ Google ได้ทำการเผยแพร่ซอฟต์แวร์ Google Earth และ Google Map API ซึ่งใช้ในการบริการข้อมูลภาพถ่ายดาวเทียมรายละเอียดสูงและข้อมูลเชิงพื้นที่ผ่านระบบอินเทอร์เน็ตให้ผู้ใช้ได้นำไปใช้ ทำให้เกิดการตื่นตัวในการใช้ระบบสารสนเทศภูมิศาสตร์กันมากยิ่งขึ้น ทุกวงการต่างรู้จักกันว่า GIS และภาพถ่ายดาวเทียมรายละเอียดสูง เห็นถึงช่องทางการนำข้อมูลเชิงพื้นที่ไปใช้ประโยชน์ในงานด้านต่างๆ

จะเห็นได้ว่าผลงานระบบแผนที่บนเว็บในอดีตเป็นงานที่มีความสลับซับซ้อน ต้องลงทุนสูงในการจัดทำและบำรุงรักษา โดยเฉพาะอย่างยิ่งในด้านข้อมูลแผนที่ ภาพถ่ายทางอากาศ/ดาวเทียม และเทคโนโลยีสนับสนุน ที่ผ่านมามีงานที่จำกัดอยู่ในวงแคบเฉพาะองค์กรขนาดใหญ่เท่านั้น การให้บริการของกูเกิลในรูปแบบ Google Maps API ทำให้ผู้ใช้สามารถสร้างระบบแผนที่บนเว็บ ที่มีคุณสมบัติตามความต้องการของแต่ละบุคคลโดยไม่ต้องเสียค่าธรรมเนียม นับว่าเป็นการปฏิวัติครั้งสำคัญในวงการแผนที่บนอินเทอร์เน็ต บริการด้านแผนที่ของกูเกิลก่อให้เกิดความตื่นตัวขึ้นเป็นอย่างสูงในวงการอินเทอร์เน็ตที่มีความต้องการปรับปรุงเนื้อหาและวิธีการนำเสนอข้อมูลบนเว็บไซต์ให้น่าสนใจโดยมีแผนที่เป็นองค์ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Google Maps Application เป็นระบบ Internet GIS ที่มีซอฟต์แวร์หรือส่วนโต้ตอบ (Interface) ทำงานอยู่บนเว็บเบราว์เซอร์ ผู้ใช้ไม่จำเป็นต้องทำการติดตั้งซอฟต์แวร์ลงบนเครื่องคอมพิวเตอร์ เพียงแค่มีโปรแกรมเว็บเบราว์เซอร์มาตรฐาน เช่น Internet explorer, Mozilla, Netscape ก็สามารถทำงานได้ทันที ข้อดีของระบบคือ สามารถทำงานได้กับเครื่องคอมพิวเตอร์ทุก Platform และยังคงค่าใช้จ่ายเรื่องค่าลิขสิทธิ์ซอฟต์แวร์ ข้อเสียคือการใช้งานจำเป็นที่จะต้องเชื่อมต่อกับอินเทอร์เน็ตตลอดเวลา และมีค่าใช้จ่ายในการเชื่อมต่ออินเทอร์เน็ตตามระยะเวลาที่ทำงาน

ระบบ Google Maps API ช่วยให้นักพัฒนาสามารถพัฒนาโปรแกรมเพื่อแทรกกูเกิลแมพเข้าไปเป็นองค์ประกอบส่วนหนึ่งในเว็บเพจที่ต้องการได้โดยเขียนเป็นรหัส Html และ จาวาสคริปต์ โดยมีขีดความสามารถกว้างขวางเน้นในด้านการนำเสนอข้อมูลแผนที่ในลักษณะหมุดปัก (Push pin / Place marker) ซึ่งสามารถกำหนดให้แสดงข้อมูลประกอบแผนที่เมื่อผู้ใช้คลิกที่ตัวมาร์กเกอร์นั้นๆ หรือองค์แผนที่แบบเส้น (Polyline) พื้นที่ (Polygon) และภาพ (Ground overlay) บริการด้านแผนที่ของกูเกิลนี้เริ่มต้นตั้งแต่กลางปี ค.ศ. 2005 เป็นบริการฟรี จัดให้แก่ผู้ใช้ทั่วโลก ในการจัดบริการแผนที่นี้ ส่วนประกอบพื้นฐานสำคัญที่ดึงดูดใจให้มีผู้ใช้งานแผนที่ของกูเกิลเป็นอย่างมากคือแผนที่และภาพถ่ายดาวเทียมคุณภาพดีซึ่งครอบคลุมทั่วพื้นผิวโลกในมาตราส่วนต่างๆ ตามความเหมาะสม ทำให้การพัฒนาต่อยอดจากสิ่งที่ถูกจัดไว้ให้แล้วเป็นงานที่น่าสนใจ ไม่ต้องลงทุนจัดหาทรัพยากรที่หายากและราคาแพงเองมาใช้ในโครงการอย่างที่เคยเป็นในอดีต

เนื่องจากการจัดทำ Google Maps API เป็น โปรแกรมรหัสเปิด (Open source program) ในภาษาจาวาสคริปต์ จึงทำให้นักพัฒนาโปรแกรมสามารถเข้าไปดูรายละเอียดของรหัส โปรแกรมได้สะดวก รวมทั้งสามารถปรับเปลี่ยนแก้ไขโปรแกรมได้ ทำให้ Google Maps API มีผู้ใช้งานอย่างกว้างขวาง เหตุผลสำคัญอีกอย่างหนึ่งที่ส่งเสริมให้มีผู้ใช้งานมากคือแผนที่และภาพถ่ายดาวเทียมคุณภาพดีกว่าผู้ให้บริการแผนที่รายอื่น

และจากเหตุผลที่กล่าวข้างต้นผู้จัดทำจึงได้พัฒนาระบบเว็บแอปพลิเคชันที่มีแกนหลักในการนำเสนอข้อมูลผ่านทางกูเกิลแมพ โดยได้นำมาประยุกต์ คือรวมเข้ากับข้อมูลที่มีอยู่และสามารถนำมาใช้งานได้ ในที่นี้คือระบบรายงานสภาวะอากาศออนไลน์ของเว็บไซต์กรมอุตุนิยมวิทยา โดยเป็นการนำเสนอข้อมูลสถานีตรวจอากาศของจังหวัดต่างๆ ทั่วประเทศไทยผ่านทางแผนที่กูเกิลแมพ โดยผู้ใช้งานสามารถเลือกดูข้อมูลที่จะแสดงได้ตามความต้องการ เพื่อให้การค้นหาข้อมูลที่เกี่ยวข้องกับการรายงานสภาวะอากาศทำได้สะดวกรวดเร็วยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างระบบการนำเสนอข้อมูลสภาวะอากาศ ในรูปแบบที่ง่ายต่อการเรียกใช้ของผู้ใช้งาน โดยเป็นการผสมผสานกันระหว่าง ข้อมูลเว็บไซต์จากเว็บไซต์กรมอุตุนิยมวิทยา กับระบบบริการแผนที่ของกูเกิลแมพ

2. เพื่อศึกษาเรื่องเว็บเซอร์วิสและเทคโนโลยีที่เกี่ยวข้อง
3. เพื่อศึกษาการออกแบบและใช้งานระบบฐานข้อมูล

1.3 ขอบเขตของโครงการ

1. ออกแบบระบบให้มีความสามารถในการติดต่อและดึงข้อมูลจากระบบบริการข้อมูลของเว็บไซต์กรมอุตุนิยมวิทยาмаเก็บในฐานข้อมูลของระบบได้
2. เว็บแอปพลิเคชันที่สร้างขึ้นมา สามารถเลือกรายละเอียดในการแสดงผลข้อมูลบนแผนที่ตามที่ใช้ต้องการ

1.4 วิธีการดำเนินการ

1. ทำการศึกษาเทคโนโลยีที่เกี่ยวข้อง
2. วิเคราะห์และออกแบบระบบ
3. ทำการพัฒนาโปรแกรม
4. วิเคราะห์ผลการพัฒนาโปรแกรม
5. ทำการแก้ไขข้อผิดพลาดต่างๆ ของโปรแกรม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เป็นประโยชน์ต่อผู้ที่ต้องการตรวจสอบข้อมูลสภาพอากาศ โดยสามารถสืบค้นข้อมูลได้สะดวกขึ้น
2. ได้รับความรู้ความเข้าใจถึงการใช้งานระบบแผนที่กูเกิลแมพ
3. ได้รับความรู้ความเข้าใจพื้นฐานการทำงานของระบบเว็บเซอร์วิส

1.6 ส่วนประกอบของโครงการ

โครงการฉบับนี้ได้แบ่งเนื้อหาออกเป็น 6 บทด้วยกันคือ

- บทที่ 1 กล่าวถึงหลักการและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของโครงการ
- บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ซึ่งประกอบด้วยทฤษฎีอะไรบ้าง
- บทที่ 3 กล่าวถึงวิธีการใช้งานแผนที่ Google Maps ซึ่งเป็นส่วนประกอบสำคัญของโครงการนี้
- บทที่ 4 กล่าวถึงขั้นตอนการออกแบบและพัฒนาโครงการ และกระบวนการสร้างส่วนต่างๆ
- บทที่ 5 กล่าวถึงการทดสอบและผลการทดลอง ของส่วนที่ได้ทำการพัฒนาขึ้น รวมทั้งผลการทดลองของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้รับจากโครงการ และข้อเสนอแนะในการพัฒนาโครงการต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ภาษา PHP

2.1.1 ลักษณะของ PHP

PHP ย่อมาจากคำว่า “Personal Home Page Tool” เป็นการเขียนคำสั่งที่เก็บและทำงานบนฝั่งเซิร์ฟเวอร์ (Server-Side Script) ซึ่งรูปแบบในการเขียนคำสั่งการทำงานนั้นจะมีลักษณะคล้ายกับภาษา Perl หรือภาษา C และสามารถที่จะใช้ร่วมกับภาษา HTML ได้อย่างมีประสิทธิภาพ ความสามารถของ PHP นั้น สามารถที่จะทำงานเกี่ยวกับ Dynamic Web ได้ทุกรูปแบบ เหมือนกับการเขียนโปรแกรมแบบ CGI (Common Gateway Interface) หรือ ASP (Active Server Pages) ไม่ว่าจะเป็นด้านการดูแลจัดการระบบฐานข้อมูล ระบบรักษาความปลอดภัยของเว็บเพจ การรับส่งลูกก็เป็นต้น คุณสมบัติที่โดดเด่นที่สุดของ PHP น่าจะเป็นการติดต่อกับโปรแกรมจัดการระบบฐานข้อมูลที่มีอยู่มากมาย แต่ความสามารถพิเศษกว่านี้ก็คือ PHP สามารถที่จะติดต่อกับบริการต่างๆ ผ่านทางโปรโตคอล (Protocol) เช่น IMAP, SNMP, NNIP, POP3, HTTP และยังสามารถติดต่อกับซ็อกเก็ต (Socket) ได้อีกด้วย

2.2.2 ข้อดีของการเลือกใช้ PHP

- ความเร็วในการพัฒนาโปรแกรม เพราะว่า PHP เป็นสคริปต์แบบ Embedded คือ สามารถแทรกร่วมกับภาษา HTML ได้อย่างอิสระ และหากเราพัฒนาโค้ดไว้ในรูปแบบของคลาสที่เขียนขึ้นเพียงครั้งเดียวแล้วเรียกใช้งานได้ตลอด ทำให้สะดวกและรวดเร็วต่อการพัฒนาโปรแกรมต่างๆ
- PHP เป็นโค้ดแบบเปิดเผย (Open Source) ความหมายเหมือนกับของฟรีนั่นเองเนื่องจาก PHP มีกลุ่มของผู้ใช้งานอยู่เป็นจำนวนมากทั่วโลก และมีเว็บไซต์อยู่เป็นจำนวนมากที่เป็นแหล่งรวบรวมซอสโค้ดโปรแกรม หรือจะเป็นบทความต่างๆ ทำให้ผู้ที่มีมือใหม่ หรือผู้ที่ต้องการศึกษาสามารถค้นหาซอสโค้ดมาเป็นแนวทางในการพัฒนาโปรแกรมได้ง่ายขึ้น
- การบริหารหน่วยความจำ (Memory Usage) มีการใช้หน่วยความจำที่ดี กล่าวคือ PHP จะไม่เรียกใช้หน่วยความจำตลอดเวลา ทำให้เซิร์ฟเวอร์ไม่จำเป็นต้องมีทรัพยากรมากนัก
- อิสระต่อระบบปฏิบัติการ เว็บแอปพลิเคชันที่ถูกสร้างขึ้นมาสามารถที่จะรันได้หลายระบบปฏิบัติการ ไม่ว่าจะเป็น Unix , Linux หรือ Windows เป็นต้นด้วยเหตุผลเหล่านี้เอง ที่ทำให้คนหลายคนหันมาเลือกใช้งาน PHP

2.2 ระบบฐานข้อมูล MySQL

2.2.1 ความหมายของระบบฐานข้อมูล MySQL

MySQL เป็น SQL (Structured Queries Language) Database server ซึ่ง SQL เป็น database language ที่นิยมมากภาษาหนึ่ง เป็น database server ขนาดเล็กซึ่งเหมาะกับแอปพลิเคชันที่มีขนาดเล็ก และ ปานกลาง อีกทั้งยังสนับสนุน standard SQL (ANSI) MySQL ถูกเขียนขึ้นมาใช้ได้หลายแพลตฟอร์มของคอมพิวเตอร์ ทั้ง Unix และ Windows ความสามารถที่สำคัญของ MySQL คือ ความเร็ว และ ความทนทาน (Robustness) MySQL ถูกสร้างเป็นกลุ่มของ routine ที่ใช้สำหรับตอบสนองการใช้งานซึ่งในปัจจุบัน MySQL ยังคงทำ การพัฒนาอย่างต่อเนื่อง MySQL เป็นฐานข้อมูลแบบโอเพนซอร์ส ที่ได้รับความนิยมในการใช้งานสูงสุด โปรแกรมหนึ่งบนเครื่องให้บริการ มีความรวดเร็วในการทำงานรองรับการทำงานจากผู้ใช้หลายๆ คนและหลายๆ งานได้ในขณะเดียวกัน MySQL ถูกพัฒนาขึ้นโดย MySQL AB โดยมีลิขสิทธิ์การใช้งาน 2 แบบ นั่นคือผู้ดูแลระบบสามารถใช้งานซอฟต์แวร์ MySQL ได้โดยไม่มีค่าใช้จ่ายใดๆ ภายใต้ลิขสิทธิ์ของ GNU (General Public License) หรืออาจเลือกใช้แบบที่มีลิขสิทธิ์ทางการค้าของ MySQL AB ซึ่งเป็นผู้ผลิตและพัฒนาซอฟต์แวร์โดยตรงก็ได้

2.2.2 ความสามารถและการทำงานของ MySQL

- MySQL ถือเป็นระบบจัดการฐานข้อมูล (DataBase Management System (DBMS))

ฐานข้อมูลมีลักษณะเป็น โครงสร้างของการเก็บรวบรวมข้อมูล การที่จะเพิ่มเติม เข้าถึงหรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอปพลิเคชันอื่นๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูลและระบบจัดการฐานข้อมูล

- MySQL เป็นระบบจัดการฐานข้อมูลแบบสัมพันธ์

ฐานข้อมูลแบบสัมพันธ์จะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็วและมีความยืดหยุ่น นอกจากนั้น แต่ละตารางที่เก็บข้อมูลสามารถเชื่อมโยงเข้าหากันทำให้สามารถรวมหรือจัดกลุ่มข้อมูลได้ตามต้องการ โดยอาศัยภาษา SQL ที่เป็นส่วนหนึ่งของโปรแกรม MySQL ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล

- MySQL แจกจ่ายให้ใช้งานแบบ Open Source

นั่นคือ ผู้ใช้งาน MySQL ทุกคนสามารถใช้งานและปรับแต่งการทำงานได้ตามต้องการ สามารถดาวน์โหลดโปรแกรม MySQL ได้จากอินเทอร์เน็ตและนำมาใช้งานโดยไม่มีค่าใช้จ่ายใดๆ นอกจากนั้น MySQL ถูกออกแบบและพัฒนาขึ้นมาเพื่อทำหน้าที่เป็นเครื่องให้บริการรองรับการจัดการกับฐานข้อมูลขนาดใหญ่ ซึ่งการพัฒนายังคงดำเนินอยู่อย่างต่อเนื่อง ส่งผลให้มีฟังก์ชันการ

ทำงานใหม่ๆ ที่อำนวยความสะดวกแก่ผู้ใช้งานเพิ่มขึ้นอยู่ตลอดเวลา รวมไปถึงการปรับปรุงด้านความต่อเนื่อง ความเร็วในการทำงาน และความปลอดภัย ทำให้ MySQL เหมาะสมต่อการนำไปใช้งานเพื่อเข้าถึงฐานข้อมูลบนเครือข่ายอินเทอร์เน็ต

2.3 ภาษา SQL

การทำงานเกี่ยวกับฐานข้อมูลต้องใช้ภาษาในการจัดการข้อมูลที่เรียกว่าภาษาเอสคิวแอล (SQL) โดยมีคำสั่งพื้นฐานที่จำเป็นในการทำงานดังนี้

2.3.1 คำสั่งในการคิวรีเพื่อแสดงข้อมูล

คำสั่งในการคิวรีเพื่อแสดงข้อมูลมีรูปแบบดังนี้

```
SELECT select_list[INTO new_table]
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_expression]
[HAVING search_conditon]
[ORDER BY order_expression [ASC|DESC]]
```

2.3.2 การจำกัดผลลัพธ์ ของแถวข้อมูลที่เหมือนกัน

เป็นไปได้ที่ในบางครั้งผลลัพธ์ของข้อมูลซึ่งได้จากการคิวรี จะมีการซ้ำกันของแถวเกิดขึ้น หากมีเป็นจำนวนมาก จะทำให้ไม่สะดวกในการนำข้อมูลไปใช้งานต่อ ดังนั้นจึงมีการใช้คีย์เวิร์ด DISTINCT ร่วมกับ SELECT เพื่อกำจัดแถวที่ซ้ำกันเหล่านี้ให้เหลือแถวเดียวเท่านั้นได้ ตัวอย่างเช่น
SELECT DISTINCT [FIRST] FROM [T-CAT]

2.3.3 การเลือกแถวที่รู้ข้อมูลเพียงบางส่วน

บ่อยครั้งที่เราทราบข้อมูลเพียงส่วน เช่น ตัวอักษรนำหน้าขึ้นต้นด้วยตัว A หรือคำว่า Computer อยู่ภายในข้อความ เป็นต้น ข้อมูลเหล่านี้สามารถนำมากำหนดเงื่อนไขได้โดยใช้คีย์เวิร์ด LIKE ประกอบร่วมกับ Wildcards ต่างๆ

2.3.4 การเลือกแถวโดยกำหนดหลายๆ เงื่อนไขร่วมกัน

ที่ผ่านมาแสดงถึงการเลือกแถวที่ตรงกับเงื่อนไขเพียงอย่างเดียวแต่ต้องการกำหนดหลายๆ เงื่อนไขพิจารณาร่วมกัน สามารถทำได้โดยการใช้โอเปอเรชันทางตรรกศาสตร์ช่วย โดยคีย์เวิร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AND ความหมายคือทุกเงื่อนไขต้องเป็นจริงทั้งหมด หรือ OR เมื่อมีเพียงเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริงเท่านั้น

2.3.5 การเรียงลำดับผลลัพธ์ของข้อมูลที่ถูกนำมาแสดง

หากต้องการให้ผลลัพธ์ที่ได้จากการคิวรีมีการเรียงลำดับกันอย่างเป็นระเบียบแล้ว เราสามารถใช้คีย์เวิร์ด ORDER BY พร้อมระบุการเรียงลำดับจากน้อยไปมาก หรือมากไปน้อยด้วยคีย์เวิร์ด ASC และ DESC ตามลำดับ

2.4 AJAX

เอแจ็กซ์ (AJAX - Asynchronous JavaScript And XML) เป็นเทคนิคในการพัฒนาเว็บแอปพลิเคชันเพื่อให้ความสามารถได้ตอบกับผู้ใช้ได้ดียิ่งขึ้น โดยการรับส่งข้อมูลในฉากหลัง ทำให้ทั้งหน้าไม่ต้องโหลดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ซึ่งช่วยทำให้เพิ่มการตอบสนอง ความรวดเร็ว และการใช้งานโดยรวม เอแจ็กซ์นั้นไม่ใช่เทคโนโลยีใหม่ แต่เป็นเทคนิคที่ได้ใช้เทคโนโลยีหลายอย่างที่มีอยู่แล้วรวมกันดังต่อไปนี้:

XHTML (หรือ HTML) และ CSS ใช้ในการแสดงผลลัพธ์และรูปแบบข้อมูล

ECMAScript เช่นจาวาสคริปต์ ในการเข้าถึง Document Object Model (DOM) เพื่อใช้ในการแสดงข้อมูลที่มีการเปลี่ยนแปลงหรือโต้ตอบกับผู้ใช้

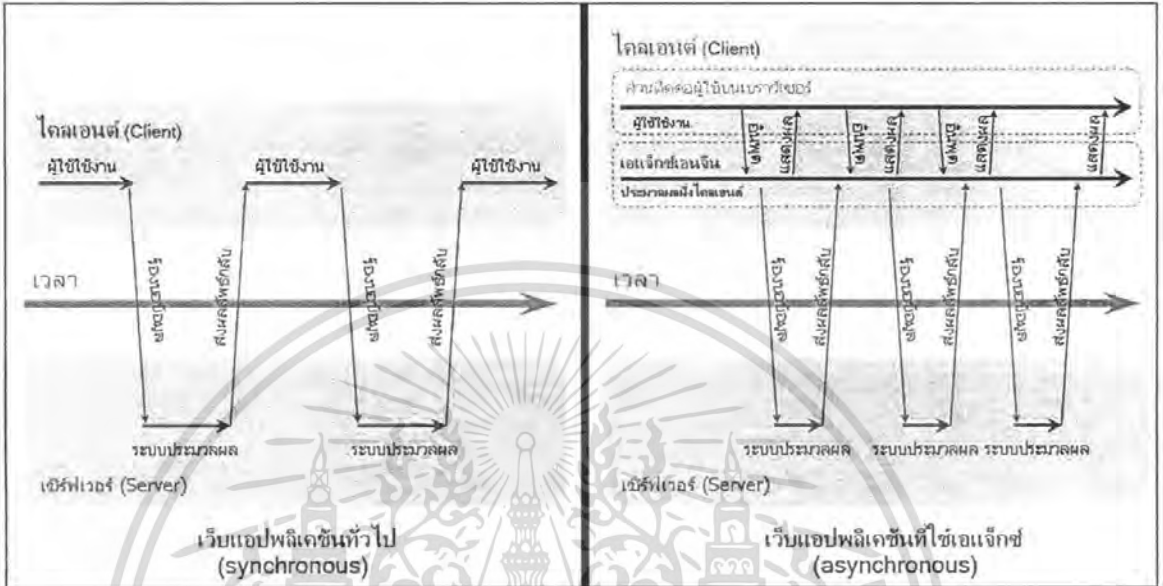
XMLHttpRequest ใช้ในการแลกเปลี่ยนข้อมูล asynchronously กับเว็บเซิร์ฟเวอร์ XML นั้นในบางครั้งใช้เป็นรูปแบบข้อมูลในการแลกเปลี่ยน ซึ่งรูปแบบอื่นก็สามารถใช้ได้เช่นกัน ไม่ว่าจะเป็น HTML, JSON, EBML, หรือ ข้อความอื่นๆ

2.4.1 ประวัติของ AJAX

Jeese Jams Garrett นั้นเป็นผู้ที่ได้บัญญัติคำเอแจ็กซ์ขึ้นเมื่อปีพ.ศ. 2548 ซึ่งได้นึกขึ้นระหว่างที่เขาอาบน้ำ เพื่อหาคำสั้นๆ สำหรับอธิบายให้ลูกค้าของเขาเกี่ยวกับเทคโนโลยีต่างๆที่ต้องการจะเสนอเอแจ็กซ์โดยตัวมันเองแล้วไม่ได้เป็นเทคโนโลยีหรือภาษาโปรแกรมชนิดใหม่ แต่เป็นการรวมกลุ่มของเทคโนโลยีที่มีใช้อยู่แล้วดังที่กล่าวข้างต้น โดยวิวัฒนาการของเอแจ็กซ์เริ่มต้นเมื่อปี ค.ศ. 2002 ไมโครซอฟท์ได้ทำการคิดค้น XMLHttpRequest ขึ้นมาเพื่อเป็นทางเลือกในการเขียนโปรแกรมบนเว็บเพจ เพื่อใช้ติดต่อกับเว็บเซิร์ฟเวอร์ในการแลกเปลี่ยนข้อมูล ซึ่งในขณะนั้นมีแค่เพียงอินเทอร์เน็ตเอกซ์พลอเรอร์ เท่านั้นที่มีความสามารถนี้ ต่อมาเว็บเบราว์เซอร์อื่นๆ เช่นจาก มอซิลลาไฟร์ฟอกซ์ ได้นำแนวคิดของ XMLHttpRequest ไปใส่ในเบราว์เซอร์ของตนด้วย จึงเริ่มทำให้มีการใช้อย่างกว้างขวางขึ้น จนปัจจุบันได้กลายเป็นมาตรฐานที่ทุกเว็บเบราว์เซอร์ต้องมีในตอนแรก นั้นไมโครซอฟท์เป็นผู้ที่ได้นำ XMLHttpRequest โดยใช้ใน Outlook Web Access ที่มาพร้อมกับ

เอกสาร Microsoft Exchange Server 2000 ต่อมาเว็บไซต์อย่างกูเกิล ได้เปิดบริการใหม่ชื่อจีเมล ซึ่งใช้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XMLHttpRequest เป็นหัวใจหลักในการดึงข้อมูลจากเว็บเซิร์ฟเวอร์ จึงทำให้แนวคิดและเทคนิคการ พัฒนาเว็บแอปพลิเคชันด้วยเอเจ็ทซ์เริ่มรู้จักเป็นที่กว้างขวางขึ้น จนปัจจุบันถือว่าเป็นหนึ่งในหัวใจ หลักของแนวคิดเรื่อง Web 2.0 ไปด้วย

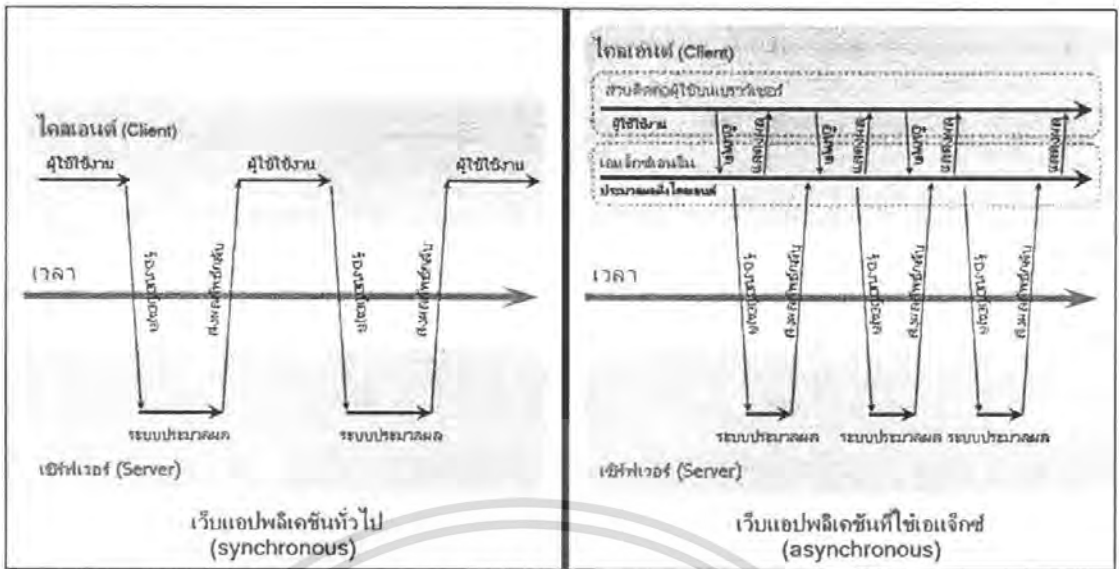


รูปที่ 2.1 เปรียบเทียบการทำงานระหว่างแอปพลิเคชันแบบเดิม กับแบบที่ใช้เอเจ็ทซ์

2.4.2 หลักการทำงาน

วิธีการทำงานของเว็บแอปพลิเคชันแบบดั้งเดิมนั้น โดยปกติแล้วเมื่อผู้ใช้ทำการร้องขอ ข้อมูลจากเซิร์ฟเวอร์ ตัวเว็บเบราว์เซอร์จะทำการส่งข้อมูลการร้องขอโดยใช้โปรโตคอล HTTP เพื่อ ติดต่อกับเว็บเซิร์ฟเวอร์ และที่เว็บเซิร์ฟเวอร์จะทำการประมวลผลจากการร้องขอที่ได้รับ และส่ง ผลลัพธ์เป็นหน้า HTML กลับไปให้ผู้ใช้ วิธีการข้างต้นเป็นวิธีการแบบการร้องขอและการตอบรับ (Request and Response) ซึ่งผู้ใช้จะต้องรอระหว่างที่เซิร์ฟเวอร์ประมวลผลอยู่ ซึ่งเป็นหลักการทำ งานแบบซิงโครนัส แต่การทำงานของเว็บแอปพลิเคชันที่ใช้เทคนิคเอเจ็ทซ์จะเป็นการทำงาน แบบอะซิงโครนัส หรือการติดต่อสื่อสารแบบไม่ต่อเนื่อง โดยเซิร์ฟเวอร์จะทำการส่งผลลัพธ์เป็น เว็บเพจให้ผู้ใช้ทันทีโดยไม่ต้องรอให้ประมวลผลเสร็จก่อน หลังจากนั้นเว็บเพจที่ผู้ใช้ได้รับจะทำ การดึงข้อมูลในส่วนต่างๆทีหลัง หรือจะดึงข้อมูลก็ต่อเมื่อผู้ใช้ต้องการเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 เปรียบเทียบการติดต่อสื่อสารระหว่างแอปพลิเคชันแบบเดิมกับแบบที่ใช้อีเจ็ทซ์

2.4.3 ข้อดีของ AJAX

- ขนาดการรับส่งข้อมูล

เทคนิคเอเจ็ทซ์นั้นสามารถสร้าง HTML ได้ในเครื่องผู้ใช้ ทำให้ขนาดข้อมูลนั้นเล็กลงในครั้งต่อไป เพราะสามารถส่งเพียงข้อมูล และคำสั่งจาวาสคริปต์ลงมาเฉพาะส่วนที่มีการเปลี่ยนแปลง แทนที่จะต้องส่งข้อมูลใหม่หมดมาทั้งหน้า. ซึ่งทั้งนี้ขึ้นอยู่กับกรอบการออกแบบของเว็บแอปพลิเคชันนั้นๆ

- ความรวดเร็วในการตอบสนอง

เนื่องจากการใช้เทคนิค เอเจ็ทซ์นั้นทำให้การเปลี่ยนแปลงต่างๆ เช่นการแก้ไข เพิ่มเติม ลบทิ้งรายการข้อมูล หรือการดึงข้อมูลที่ต้องการจะคั้นหา นั้น สามารถทำได้ในฉากหลัง ทำให้ผู้ใช้รู้สึกการตอบสนองนั้น คล้ายคลึงกับ โปรแกรมคอมพิวเตอร์ มากกว่าเว็บปกติที่ต้องรอโหลดใหม่ทั้งหน้าสำหรับการเปลี่ยนแปลงต่างๆ

2.4.4 ปัญหาที่พบ

- ปัญหาการใช้งานเกี่ยวกับปุ่ม "ย้อนกลับ"

เนื่องจากว่าเทคนิค เอเจ็ทซ์นั้นทำงานในฉากหลัง และไม่ได้เรียกหน้าใหม่ ทำให้เวลาใช้ปุ่ม "ย้อนกลับ" (Back) ในเว็บเบราว์เซอร์ อาจจะได้หน้าที่ควรจะเป็น ผู้พัฒนานั้นได้คิดค้นวิธีการแก้ไขปัญหานี้หลากหลายรูปแบบด้วยกัน หนึ่งในวิธีที่ใช้แก้ไขปัญหานี้อย่างแพร่หลายคือการใส่ IFRAME ที่มองไม่เห็นเพื่อสั่งให้เว็บเบราว์เซอร์เปลี่ยนแปลงรายการหน้าของปุ่มก่อนหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• ปัญหาในการค้นหา

เนื่องจากว่าข้อมูลสามารถเปลี่ยนแปลงโดยไม่ต้องโหลดหน้าใหม่นั้น ทำให้ยากต่อการที่จะค้นหาในสถานะปัจจุบันที่ต้องการ เนื่องจากถือว่าเป็นหน้าเดียวกัน. อย่างไรก็ตาม นักพัฒนาซอฟต์แวร์ได้คิดค้นวิธีการแก้ไขปัญหานี้ โดยการใส่ ชิ้นส่วนตัวแปร (Fragment Identifier) ใส่ในส่วนของURL ตามหลังเครื่องหมาย '#' เพื่อใช้ในการระบุสถานะของเว็บแอปพลิเคชัน สาเหตุที่ใช้วิธีนี้ได้นั้นเนื่องจากว่าจาวาสคริปต์นั้นสามารถเปลี่ยนชิ้นส่วนตัวแปรนี้ได้โดยไม่ต้องโหลดหน้าใหม่. อย่างไรก็ตามวิธีแก้ไขปัญหานี้ไม่ใช่วิธีที่สมบูรณ์แบบ

• เวลาที่ใช้ในการรับส่งข้อมูล

เนื่องจากว่าผู้ใช้อาจจะรู้สึกถึงความล่าช้าได้ ซึ่งในหลายกรณีผู้ใช้อาจจะไม่เข้าใจถึงสาเหตุ. ความล่าช้าในเทคนิค เอเจ็ซนั้นจะเห็นได้ชัดกว่าการโหลดหน้าใหม่ทั้งหน้า เนื่องจากเวลาโหลดทั้งหน้าใหม่นั้น สายตาจะได้รับข้อมูลการปรับเปลี่ยน ทำให้ผู้ใช้รับทราบว่าหน้ากำลังโหลดอยู่. ประเด็นนี้จึงเป็นปัจจัยที่ผู้พัฒนาควรที่จะคำนึงถึงระหว่างการออกแบบเว็บแอปพลิเคชัน โดยผู้พัฒนาส่วนใหญ่ นั้น ได้แก้ไขปัญหานี้โดยใช้รูปภาพแสดงว่าข้อมูลส่วนดังกล่าวกำลังโหลดอยู่ในฉากหลัง เพื่อลดความรู้สึกดังกล่าว

• ปัญหาความเข้ากันได้กับเสิร์ชเอนจิน

ผู้พัฒนายังต้องคำนึงการออกแบบ ที่จะให้เสิร์ชเอนจินต่าง ๆ นั้นสามารถอ่านและจัดทำดัชนีได้. ปัญหานี้ไม่ใช่ปัญหาที่เกิดขึ้นกับเว็บแอปพลิเคชันที่ใช้เทคนิค เอเจ็ซเท่านั้น แต่ยังเกิดขึ้นกับเว็บที่ดึงข้อมูลหน้าใหม่ทั้งหมดจากการตอบรับ เช่น กดปุ่ม เป็นต้น

• ความน่าเชื่อถือของจาวาสคริปต์

เนื่องจากว่าเอเจ็ซใช้จาวาสคริปต์ ซึ่งจาวาสคริปต์นั้นอาจทำงานแตกต่างกันขึ้นอยู่กับเว็บเบราว์เซอร์หรือรุ่นของเว็บเบราว์เซอร์นั้นๆ และหากต้องการให้เข้ากันได้หลายเว็บเบราว์เซอร์ อาจต้องการการทดสอบและตรวจสอบความเข้ากันได้บนเบราว์เซอร์ต่างๆ ซึ่งบางครั้งอาจจำเป็นที่จะต้องเขียนโค้ดแยกเช่นบางส่วนของไออี และอีกส่วนสำหรับไฟร์ฟอกซ์ เป็นต้น

2.5 XML (eXtensible Markup Language)

2.5.1 ไวยากรณ์ทั่วไปของ XML

“การสื่อสารข้อมูล” เป็นคำที่มีความหมายกว้างและแทบจะเปล่าประโยชน์ในการพยายามหาคำจำกัดความหรือหาข้อกำหนดที่สามารถครอบคลุมได้อย่างทั่วถึง จากเหตุผลข้างต้นทำให้มีข้อกำหนดที่เกี่ยวข้องกันจำนวนหนึ่ง (บางอันก็ยังคงอยู่ในส่วนของการพัฒนาระยะแรกๆ) ซึ่งทำงานควบคู่กันไปในกลุ่มของ XML ซึ่งแต่ละข้อกำหนดจะครอบคลุมในแต่ละด้านของการสื่อสารข้อมูลต่อไปนี้เป็นข้อกำหนดที่สำคัญบางเรื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- XML 1.0 เป็นข้อกำหนดพื้นฐานซึ่งเกี่ยวข้องกับการสร้างกลุ่มของเทคโนโลยี XML กล่าวถึงไวยากรณ์ซึ่งเอกสาร XML ต้องยึดถือ และกฎเกณฑ์ที่ XML Parser จำต้องยึดถือ รวมถึงทุก ๆ สิ่งที่เป็นต่อการอ่านหรือเขียนเอกสาร XML (นอกจากนี้ยังกำหนดในเรื่องของ DTDs ด้วย แม้ในบางครั้งสองเทคโนโลยีนี้จะถูกแยกจากกันก็ตาม)

- เนื่องมาจากการที่เราสามารถกำหนดโครงสร้างและชื่อของอิลิเมนต์ ขึ้นมาสำหรับใช้ในเอกสารของเราได้ จึงจำเป็นต้องมี DTD และ สกีม่า (Schemas) เพื่อช่วยในการจัดสร้างเทมเพลตสำหรับประเภทเอกสารขึ้นมา เพื่อที่จะตรวจสอบได้ว่าเอกสารต่าง ๆ เป็นไปตามที่กำหนดไว้ในเทมเพลต และนักพัฒนาคนอื่น ๆ จะสามารถสร้างเอกสารที่เข้ากันได้กับเทมเพลตนั้น ๆ

- เนมสเปซ (Namespace) ช่วยให้เราสามารถแยกแยะ XML Vocabulary แต่ละอันออกจากกันได้ ทำให้เราสามารถสร้างเอกสารที่ใช้งานได้กว้างขึ้นด้วยการรวมเอา Vocabulary หลาย ๆ อันมาไว้ในประเภทเอกสารเดียวกัน

- Xpath ใช้กำหนดรายละเอียดของภาษาที่ใช้ในการสืบค้นข้อมูลบางส่วนของเอกสาร XML ทำให้แอปพลิเคชันต่าง ๆ สามารถค้นหาข้อมูลบางส่วนของเอกสาร XML ที่ต้องการได้ แทนที่จะต้องนำเอาข้อมูลทั้งหมดของเอกสารออกมาใช้งาน ตัวอย่างเช่น Xpath สามารถใช้ค้นหา “นามสกุลทั้งหมดที่มีอยู่ในเอกสาร” ได้

- ในบางกรณีเราอาจจำเป็นต้องแสดงผลเอกสาร XML ของเรา สำหรับในกรณีที่ไม่ซับซ้อนเราสามารถ ใช้ Cascading Style Sheets Language (XSL) ที่เกิดมาจาก XSLT ซึ่งสามารถแปลงเอกสารของเราจากประเภทหนึ่งไปสู่อีกประเภทหนึ่งได้ รวมถึง Formatting Object ที่ทำงานเกี่ยวกับการแสดงผลเช่นกัน

- XLink และ Xpointer เป็นภาษาที่ใช้สำหรับเชื่อมโยงเอกสาร XML เข้าไว้ด้วยกัน ในลักษณะที่ใกล้เคียงกับไฮเปอร์ลิงก์ ในเอกสาร HTML

- Document Object Model (DOM) เป็นแอปพลิเคชันที่มักใช้ในการติดต่อกับเอกสาร XML กันอย่างแพร่หลาย รวมถึงทางเลือกอีกอย่างหนึ่งสำหรับ โปรแกรมเมอร์ที่จะใช้ติดต่อกับเอกสาร XML จาก โค้ดของ XML เอง เรียกว่า Simple API for XML (SAX)

2.5.2 Tag, Text, Element

การสร้าง Element ของ XML โดยการใช้ tag เริ่มต้น และ tag สิ้นสุด ดังตัวอย่างต่อไปนี้

```
<name> tag XML(Start tag)
<first>John</first>
<middle>Fitzgerald Johansen</middle>
<last>Doe</last>
</name> End tag
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำที่อยู่ระหว่างเครื่องหมาย < และ > เรียกว่า tag XML ข้อมูลของเราจะอยู่ใน tag หลายๆ ตัวที่ประกอบกันเป็น markup ของเอกสาร ซึ่งจะช่วยในการแยกข้อมูลเอกสารและ markup ออกจากกัน

เมื่อมี tag เริ่มต้น <name> ก็ต้องมี tag สิ้นสุดเช่นกันหรือที่เรียกว่า End tag สังเกตได้จากจะมีเครื่องหมาย / อยู่หน้า tag นั้น ในตัวอย่างนี้คือ </name> โดยที่ end tag จะต้องเหมือนกับ start tag ทุกประการเว้นแต่ เครื่องหมาย / ที่เพิ่มเข้ามาเท่านั้น

สิ่งต่างๆ ตั้งแต่ tag เริ่มต้น จนถึง tag สิ้นสุดนั้น เรียกว่า element เช่น <last>Doe</last> ทั้งหมดนี้เรียกว่า element โดยมี tag เริ่มต้นคือ <last> และ tag สิ้นสุด คือ </last> เอกสารทั้งหมดเริ่มตั้งแต่ <name> จนถึง </name> ก็ถือว่าเป็น element ด้วยเช่นเดียวกัน โดยมีความแตกต่างคือ tag บนสุดนั้น เราจะเรียกว่าเป็น root element

สิ่งที่อยู่ระหว่าง tag ใด ๆ เรียกว่า เนื้อหาของ element เช่น Doe ซึ่งมักจะเป็นข้อมูลอย่างเดียว เราจะทำการศึกษาเกี่ยวกับข้อมูลลักษณะต่างๆของเนื้อหาของ element ต่อไป

กฎของ Element

ในการที่เราจะสร้างเอกสาร XML ให้เป็น well-form XML นั้น เราจำเป็นต้องมีกฎข้อบังคับที่เป็นที่ยอมรับกันอย่างแพร่หลาย โดยจะมีกฎย่อยๆดังนี้

- ทุก tag เริ่มต้นจะต้องมี tag สิ้นสุดที่เข้าคู่กันเสมอ (เป็นความจำเป็นที่เกี่ยวกับ browser ของ XML)
- Tag ต่างๆ จะคาบเกี่ยวกันไม่ได้ (จะต้องอิงโครงสร้างแบบทรีไว้เป็นหลัก)
- เอกสารทั้งหมดจะต้องอยู่ใน root element เดียวเท่านั้น
- ชื่อของ element จะต้องเป็นไปตามกฎของชื่อ element เท่านั้น ดังกฎต่อไปนี้
- ชื่อของ element ต้องขึ้นต้นด้วยตัวอักษรหรือ “_” เท่านั้น
- หลังจากตัวเลขแล้วสามารถเป็นตัวเลขหรือเครื่องหมาย “-” หรือ “.” ก็ได้
- ชื่อของ element ต้องไม่มีช่องว่าง
- ชื่อของ element ห้ามมีเครื่องหมาย “:”
- ชื่อของ element ห้ามขึ้นต้นด้วย “xml” ไม่ว่าจะเป็นตัวพิมพ์แบบใดก็ตาม
- ต้องไม่มีช่องว่างต่อจากอักษร “<” ชื่อของ element ต้องเขียนต่อจาก <ทันที แต่ก็อนุญาตให้มีช่องว่างก่อนที่จะเป็นเครื่องหมาย “>” ได้
- ให้ความสำคัญของตัวพิมพ์เล็ก-ใหญ่ (Case-Sensitivity)

ช่องว่างใน PCDATA

ช่องว่างประเภทสิ้นสุดบรรทัด (End of Line) XML จะกำหนดให้การขึ้นบรรทัดใหม่นั้น

ใช้เป็น Line Feed เพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องว่างใน markup

ช่องว่างที่ใช้ในการขึ้นบรรทัดใหม่เป็นช่องว่างที่ไม่มีมีความหมายใดๆ เพื่อความเข้าใจได้ง่าย เรียกว่า ช่องว่างที่ไม่เกี่ยวข้อง (Extraneous White space) XML Parsers จะสามารถแยกแยะได้ว่าช่องว่างใดบ้างที่เกี่ยวข้อง

Attribute

Attribute คือ คู่ของชื่อและค่าที่สัมพันธ์กันและสัมพันธ์กับ element ด้วยสามารถเพิ่มลงไปในส่วนของ tag เริ่มต้นได้เช่น

```
<name nickname='Shiny John'>
<first>John</first>
<middle>Fitzgerald Johansen</middle>
<last>Doe</last>
</name>
```

ทุกๆ Attribute จะต้องมีค่าเริ่มต้นอยู่เสมอ แม้ว่าจะเป็นเพียงค่าสตริงว่างเท่านั้น และค่าเหล่านั้นจะต้องอยู่ในเครื่องหมายคำพูด ซึ่งอาจจะใช้เป็น เครื่องหมาย Single Quote หรือ Double Quote ก็ได้ แต่ต้องให้เหมือนกันหรือเข้าคู่กันด้วย นอกจากนี้การกำหนดชื่อของ Attribute ก็ถูกกำหนดให้มีกฎเกณฑ์ไว้เช่นเดียวกันกับการกำหนดชื่อของ element ดังที่กล่าวมาแล้ว และยังไม่อนุญาตให้มีการกำหนด attribute เกินกว่า 1 attribute ที่มีชื่อเดียวกันภายใน element เดียวกันอีกด้วย

Comment

ส่วนของคอมเมนต์ที่ไม่ได้เป็นส่วนหนึ่งของเอกสาร มีเพียงเพื่อให้ผู้อ่านสามารถเข้าใจเอกสาร XML ได้ง่ายขึ้นเท่านั้น คอมเมนต์จะเริ่มต้นด้วยตัวอักษร <!-- และจบลงด้วย -->

Element ว่าง

คือ element ที่ไม่มีข้อมูลอยู่เลยภายใน element นั้น เช่น <middle></middle> ในกรณีนี้สามารถใช้ tag พิเศษที่แทนด้วย <middle/> หรือ <middle />

การประกาศว่าเป็น XML (XML Declaration)

เพื่อกำหนดว่าเป็นเอกสารประเภท XML ดังนั้นจึงควรใส่เอาไว้

```
<?xml version='1.0' encoding='UTF-16' standalone='yes'?> หรือ <?xml version='1.0'>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะประกาศไว้ที่ส่วนหัวของเอกสาร เนื้อหาที่เป็น root element และส่วนที่จำเป็นมีเพียงแค่ ส่วน version ส่วนที่เป็น encoding หรือ standalone นั้นเป็นเพียงส่วนขยายเท่านั้น จะใส่หรือไม่ก็ได้ นอกจากนี้ ยังจำเป็นจะต้องเรียงลำดับก่อนหลังดังตัวอย่างที่อยู่ข้างต้นอีกด้วย และเวอร์ชันล่าสุดของ XML คือ 1.0

คำสั่งในการประมวลผล Processing Instruction (PI)

PI ต้องเริ่มต้นด้วย “<?” และตามด้วยชื่อของแอปพลิเคชันที่จะรับข้อมูล PItarget และข้อความที่จะส่งผ่านไปยัง application ดังตัวอย่างนี้

```
<?nameprocessor SELECT * FROM blah?>
```

ซึ่งคำสั่งนี้จะถูกส่งไปยัง Parsers แต่จะไม่ถูกส่งไปยังแอปพลิเคชัน

Namespaces

Namespaces หมายถึงอะไรก็ตามที่เราใช้จำแนก element ของเอกสาร XML และในบางครั้งเราก็ยังใช้จำแนก attribute ได้อีกด้วย เนื่องจากโปรแกรมเมอร์อาจจะใส่ข้อมูลที่แตกต่างกันลงไป ใน element เดียวกัน เช่นในกรณีที่เราต้องผสมผสาน XML ที่เป็นของเราเองและ XHTML element ที่หลากหลายจากเอกสารหลายประเภทเข้าไว้ในเอกสาร XML เพียงฉบับเดียว

การใช้คำนำหน้า (Prefix)

วิธีที่ดีที่สุดที่จะนำมาแก้ปัญหาดังกล่าวนี้คือ ทำให้ element ใดๆ element เป็นชื่อเอกสารที่ไม่ซ้ำกันเลย โดยการใส่คำนำหน้าคำหนึ่งให้แก่ทุก element ที่เป็นเอกสาร XML และใช้คำนำหน้าอีกคำหนึ่งกับ element ของ XHTML ซึ่งอาจจะเกิดความซ้ำซ้อนระหว่างคำนำหน้าเหล่านั้น จึงกำหนดให้ใช้ชื่อโดเมนเนมที่มีอยู่แล้วใน internet แล้วกำหนดให้ URI ดังกล่าวนั้นต้องถูกนำไปใช้เพื่อเป็นคำนำหน้าชื่อ

2.5.3 XML Namespaces

การที่จะใช้ XML Namespaces ในเอกสารของเรานั้น element ทั้งหมดก็ยังจำเป็นต้องถูกกำหนดค่าด้วยชื่อที่ตรงตามคุณสมบัติที่กำหนด หรือ QName ซึ่งจะประกอบไปด้วยส่วนประกอบสองส่วนนั่นคือ ส่วนชื่อเฉพาะ (local par) คือชื่อทั้งหลายที่เรากำหนดให้กับ element มาโดยตลอด และ Namespaces Prefix ซึ่งระบุว่าชื่อดังกล่าวนี้ขึ้นอยู่กับ Namespaces ใด

หลักการการทำงานของ Namespaces

ถ้าเรามีเอกสารที่มี element จาก Namespaces สองชื่อแยกออกจากกัน ซึ่งเราได้กำหนดเอาไว้ในโค้ดต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<pers:person xmlns="http://sernaferma.com/pers"
xmlns:html = "http://www.w3.org/1999/xhtml">
```

ตัวคำนำนานี้ไม่ได้มีความหมายใดๆต่อ XML Parsers เลย เพียงแต่มันถูกแทนที่ด้วย URI เท่านั้น

Default Namespaces

มีไว้เพื่อให้เราสามารถกำหนดคำนำนานี้เพียงครั้งเดียวตอนเริ่มต้น

```
<person xmlns="http://sernaferma.com/pers">
<name>
<title>Sir</title>
</name>
</person>
```

จะสังเกตได้ว่า attribute xmlns นั้นไม่ได้ระบุคำนำนานี้เพื่อใช้กับ namespaces นี้อีกต่อไปเพราะมันเป็น Default Namespaces ซึ่ง element นี้ และ element ลูกหลานใดๆทั้งหลายขึ้นอยู่กับ namespaces นี้ แม้จะได้มีการระบุอย่างชัดเจนว่าเป็น namespaces อื่นใดด้วยการใช้คำนำนานี้ แต่อิลิเมนต์ <name> และ <title> ทั้งคู่ก็ยังขึ้นอยู่กับ namespaces นี้พอดี

เราสามารถประกาศ namespaces ได้มากกว่าหนึ่ง namespaces สำหรับหนึ่งอิลิเมนต์ แต่มีเพียง namespaces เดียวเท่านั้นที่เป็นดีฟอลต์ได้ ซึ่งจะอนุญาตให้เรากำหนด XML ได้ดังนี้ :

```
<person xmlns="http://sernaferma.com/pers"
xmlns:html = "http://www.w3.org/1999/xhtml">
<name/>
<xhtml:p>This is XHTML</xhtml:p>
</person>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีนี้ทุก ๆ อติเม้นต์จะขึ้นอยู่กับ namespaces ที่เป็น <http://sernafera.com/pers> ยกเว้นอติเม้นต์ <p> ซึ่งเป็นส่วนหนึ่งของ namespaces (เราได้ประกาศ namespaces และคำนำหน้าทั้งหลาย (ถ้ามี)เอาไว้แล้วในอติเม้นต์แรก ดังนั้นทุกอติเม้นต์ในเอกสารก็สามารถใช้คำนำหน้าเหล่านี้ได้)

2.5.4 DTD และ XML Schema

การกำหนดโครงสร้างของเอกสาร XML ทำได้ 2 วิธีหลักๆคือ DTD และ XML Schema โดยไม่ว่าจะกำหนดตามวิธีใดก็แล้วแต่ เรียกว่าเอกสาร XML นั้น “valid” เหตุผลที่ต้องกำหนดโครงสร้างของเอกสาร ก็เพื่อให้เอกสาร XML นั้นมีรูปแบบที่ถูกต้อง พร้อมทั้งจะนำไปใช้งาน เพราะธรรมชาติของเอกสาร XML ก็คือการแลกเปลี่ยนข้อมูลระหว่างองค์กรหรือหน่วยงาน ดังนั้นฝ่ายที่รับเอกสาร XML จากองค์กรอื่น ต้องมั่นใจว่าเอกสารนั้นมีความถูกต้อง สามารถนำไปใช้งานได้ พุดให้ง่ายๆอีกอย่างหนึ่งก็คือ DTD หรือ XML Schema เปรียบเหมือน “สัญญา” ระหว่างหน่วยงานหรือผู้ที่เกี่ยวข้องกับเอกสาร XML นั้นๆ เมื่อใดก็ตามที่มีการสร้างเอกสาร XML ขึ้นมาเก็บข้อมูล จะต้องปฏิบัติตามข้อกำหนดใน DTD หรือ XML Schema หากไม่ทำตาม ก็ถือว่าผิดสัญญาและเอกสาร XML นั้นขอมนำไปใช้งานไม่ได้

สรุปความสำคัญของ DTD และ XML Schema ได้เป็น 2 แง่มุม คือ

1. เป็นการกำหนดรูปแบบโครงสร้างของเอกสาร XML (หรือเรียกว่า Data Model) ว่าประกอบไปด้วยแท็กอะไรบ้าง, มีแอตทริบิวต์อะไรบ้าง, ค่าของแอตทริบิวต์เป็นอะไรได้บ้าง, ลักษณะการซ้อนแท็กเป็นอย่างไรได้บ้าง เป็นต้น
2. เป็นสัญญาระหว่างองค์กร ที่จะแลกเปลี่ยนข้อมูลกันในรูปแบบของ XML เพื่อให้เอกสารที่สร้างขึ้นมามีรูปแบบตรงตามที่ตกลงกันไว้ และจะได้สามารถนำเอกสาร XML นั้นไปใช้งานได้อย่างถูกต้อง

DTD (Document Type Definition)

ภาษาSGML จะต้องมียเอกสารที่เรียกว่า DTD เพื่อใช้ในการนิยามโครงสร้างของเอกสารหรือพูดอีกอย่างหนึ่งคือ จะอาศัย DTD เป็นต้นแบบเพื่อสร้างเอกสาร SGML ที่จะใช้งานออกมา (จะต้องมีเอกสาร DTD กำกับเอกสาร SGML ทุกครั้ง) ถ้าจะอธิบายเพิ่มเพื่อความชัดเจนขึ้นอีกคือ DTD เปรียบเสมือนพิมพ์เขียวที่ใช้สร้างบ้าน โดยที่พิมพ์เขียว 1 ใบจะนำไปสร้างบ้านก็หลังก็ได้ รายละเอียดในพิมพ์เขียวก็คือองค์ประกอบเกี่ยวกับ โครงสร้างของบ้านนั่นเอง

ส่วนภาษาXML ซึ่งเป็นภาษาที่ได้รวมเอาข้อดีของภาษา SGML และภาษา HTML เข้าไว้ด้วยกันนั้น ไม่จำเป็นต้องมี (แต่ควรจะมี) เอกสาร DTD ก็ได้

ลักษณะของเอกสาร DTD

ต่อไปนี่คือตัวอย่าง DTD ง่ายๆของเอกสาร XML ที่ใช้เก็บข้อมูลแค็ตตาล็อกหนังสือ

- 1: <!ELEMENT BookCatalogue (Book)*>
- 2: <!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>
- 3: <!ELEMENT Title (#PCDATA)>
- 4: <!ELEMENT Author (#PCDATA)>
- 5: <!ELEMENT Date (#PCDATA)>
- 6: <!ELEMENT ISBN (#PCDATA)>
- 7: <!ELEMENT Publisher (#PCDATA)>

คราวนี้ลองมาดูตัวอย่างของเอกสาร XML ที่เป็นไปตามข้อกำหนดใน DTD นี้บ้าง

- 1: <?xml version="1.0"?>
- 2: <BookCatalogue>
- 3: <Book>
- 4: <Title>เปิดมิติ Mobile Internet ด้วย WAP</Title>
- 5: <Author>ตราวุธ อ้อยศรีสกุล</Author>
- 6: <Date>1 กุมภาพันธ์ 2550</Date>
- 7: <ISBN>974-87931-0-9</ISBN>
- 8: <Publisher>วิศิต์ กรู๊ป</Publisher>
- 9: </Book>
- 10: </BookCatalogue>

ประกาศ DTD ในเอกสาร XML

DTD จะประกาศไว้ในส่วนของ Document Type Declaration โดยเราจะแทรก DTD ไว้ในตัวเอกสาร XML เลยก็ได้ คือเป็น Internal DTD หรือจะแยกเป็นไฟล์ต่างหากก็ได้เช่นกัน (มีนามสกุล .dtd) ก็คือเป็น External DTD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XML Schema

XML Schema เป็นอีกวิธีหนึ่งในการกำหนดโครงสร้างของเอกสาร XML และเป็นวิธีที่ใหม่กว่า DTD เพราะ DTD มีใช้มาตั้งแต่สมัยภาษา SGML แล้ว แต่ XML Schema เพิ่งเป็นมาตรฐานของ W3C เมื่อประมาณกลางปีพ.ศ. 2544

ข้อดีของ XML Schema ที่เหนือกว่า DTD

มูลเหตุเริ่มจากกลุ่มคนที่ใช้ DTD กับเอกสาร XML เกิดความรู้สึกว่า DTD ใช้ไวยากรณ์ที่แตกต่างจาก XML เพราะ XML มีลักษณะเป็นแท็ก แต่ DTD ไม่เป็นเช่นนั้น ทำให้เกิดอุปสรรคในการทำงาน ถ้า DTD ซึ่งเป็นเสมือนพิมพ์เขียวที่กำหนดโครงสร้างเอกสาร XML มีไวยากรณ์แบบ XML ด้วย ย่อมจะดีกว่า นอกจากนั้น DTD ยังรู้จักชนิดของข้อมูลเพียงไม่กี่แบบและยังระบุเจาะจงชนิดข้อมูลไม่ได้ เช่นเราไม่สามารถกำหนดได้ว่าข้อมูลในอติเมนต์ <quantity> ต้องเป็นตัวเลขที่มีค่าตั้งแต่ 0-999 เท่านั้น เป็นต้น แต่ XML Schema ไม่มีข้อจำกัดนี้เลย เพราะรองรับชนิดข้อมูลไม่ต่ำกว่า 41 ชนิด และที่เขี่ยมกว่านั้นคือ เป็นชนิดข้อมูลที่รองรับในโปรแกรมฐานข้อมูลที่มีอยู่ในปัจจุบันด้วย นั่นหมายความว่าเราสามารถนำข้อมูลที่อยู่ในรูปเอกสาร XML ไปเก็บไว้ในฐานข้อมูลได้ง่ายๆ โดยพิจารณาจากชนิดข้อมูลใน XML Schema ไม่เพียงแค่นั้น XML Schema ยังยอมให้เรา

- กำหนดรูปแบบข้อมูลเองได้ เช่น กำหนดว่าข้อมูลหมายเลขโทรศัพท์ต้องมีลักษณะรูปแบบ (pattern) เป็น “d-dddd-dddd” คือตัวเลข 1 ตัวตามด้วยขีด (-) และตัวเลข 4 ตัว จากนั้นจึงตามด้วยขีด (-) อีกครั้งและตัวเลขอีก 4 ตัว เป็นต้น
- สร้างชนิดของข้อมูลใหม่โดยอิงจากชนิดของข้อมูลที่มีอยู่เดิม
- กำหนดจำนวนและลำดับของอติเมนต์ลูกได้ (อติเมนต์ลูกคืออติเมนต์ที่อยู่ภายในอติเมนต์ปัจจุบัน)

และที่สำคัญที่สุดคือ XML Schema รองรับการใช้เนมสเปซ (namespace) ซึ่งถือเป็นคอนเซ็ปต์ใหม่และมีประโยชน์อย่างมากในการพัฒนา XML

รูปลักษณะหน้าตาของ XML Schema

ก่อนอื่นเพื่อความคุ้นเคย เราจะมาดูว่า XML Schema มีลักษณะเป็นอย่างไร ซึ่งจะอธิบายโดยเปรียบเทียบ DTD กับ XML Schema ของเอกสารเดียวกัน จะให้เห็นความแตกต่างอย่างชัดเจน สมมติว่า DTD ของแคตตาล็อกหนังสือ มีลักษณะดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1: <!ELEMENT BookCatalogue (Book)*>
- 2: <!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>
- 3: <!ELEMENT Title (#PCDATA)>
- 4: <!ELEMENT Author (#PCDATA)>
- 5: <!ELEMENT Date (#PCDATA)>
- 6: <!ELEMENT ISBN (#PCDATA)>
- 7: <!ELEMENT Publisher (#PCDATA)>

เมื่อเปลี่ยนมาใช้ XML Schema จะมีลักษณะดังนี้

- 1: <?xml version="1.0"?>
- 2: <xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema" targetNamespace="http://www.publishing.org" xmlns="http://www.publishing.org" elementFormDefault="qualified">
- 3: <xsd:element name="BookCatalogue">
- 4: <xsd:complexType>
- 5: <xsd:sequence>
- 6: <xsd:element ref="Book" minOccurs="0" maxOccurs="unbounded"/>
- 7: </xsd:sequence>
- 8: </xsd:complexType>
- 9: </xsd:element>
- 10: <xsd:element name="Book">
- 11: <xsd:complexType>
- 12: <xsd:sequence>
- 13: <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
- 14: <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
- 15: <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
- 16: <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

17: <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
18: </xsd:sequence>
19: </xsd:complexType>
20: </xsd:element>
21: <xsd:element name="Title" type="xsd:string"/>
22: <xsd:element name="Author" type="xsd:string"/>
23: <xsd:element name="Date" type="xsd:string"/>
24: <xsd:element name="ISBN" type="xsd:string"/>
25: <xsd:element name="Publisher" type="xsd:string"/>
26: </xsd:schema>

```

2.5.5 การแปลงรูปแบบเอกสาร XML ด้วย XSLT (XSL Transformation Language)

เป็นข้อเสนอแนะจาก W3C ในการกำหนดชุด Element ของ XML ที่สามารถจะถูกใช้ในการสร้างสไลด์ที่แปลงเอกสาร XML ให้อยู่ในรูปแบบใด ๆ ซึ่งในปัจจุบันนี้ใช้ในการแปลง XML รูปหนึ่ง ไปเป็นอีกรูปแบบหนึ่ง โดยการทำงานจะทำโดยนำเอกสาร XML ที่เหมาะสมกับ XML นั้นไปทำการแปลงรูปแบบเพื่อสร้างเอกสารเอาต์พุต ซึ่งขึ้นอยู่กับคำสั่งและเนื้อหาภายในสไลด์จึงสามารถสร้างรูปแบบเอาต์พุตใด ๆ ก็ได้ตามที่ต้องการในการทำงาน

2.6 Javascript

2.6.1 ลักษณะของ Javascript

จาวาสคริปต์เป็นภาษายุคใหม่สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง เราสามารถเขียนโปรแกรมจาวาสคริปต์เพิ่มเข้าไปในเว็บเพจเพื่อใช้ประโยชน์สำหรับงานด้านต่างๆ ทั้งการคำนวณ การแสดงผล การรับ-ส่งข้อมูล และที่สำคัญคือ สามารถโต้ตอบกับผู้ใช้ได้อย่างทันทีทันใด นอกจากนี้ยังมีความสามารถด้านอื่นๆ อีกหลายประการที่ช่วยสร้างความน่าสนใจให้กับเว็บเพจของเราได้อย่างมาก ภาษาจาวาสคริปต์ถูกพัฒนาโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับบริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง Live Script ใหม่เมื่อปีพ.ศ. 2538 แล้วตั้งชื่อใหม่ว่าจาวาสคริปต์ โครงสร้างภาษาของจาวาสคริปต์มีความคล้ายคลึงกับจาวา มาก โดยจาวาสคริปต์เป็นคอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้เมนต์ (Complement) ของจาวาสามารถติดต่อกับส่วนต่าง ๆ ของจาวาแอปพลิเคชันโดยสคริปต์ที่เขียนขึ้นมาได้ คำสั่งของจาวาสคริปต์สามารถนำมาใช้แสดง, กำหนดคุณสมบัติ, สอบถามสถานะ หรือ ควบคุมการทำงานของแอปพลิเคชันและปลั๊กอิน นอกจากนี้ จาวาสคริปต์ยังสนับสนุนรูปแบบนิพจน์และการควบคุมพื้นฐานของภาษาจาวาอีกด้วย จาวาสคริปต์ได้ถูกออกแบบมาเพื่อใช้เป็นส่วนเพิ่มขยายในภาษา HTML โดยเฉพาะ ช่วยให้สามารถควบคุมเว็บเพจได้อย่างง่ายดาย เหมาะกับการทำงานอย่างรวดเร็ว และเน้นที่ความถูกต้องเป็นสิ่งสำคัญ ภาษาจาวาประกอบไปด้วย เอ็กซ์คลูซีฟ (Exclusive) ของคลาสและเมธอด ต้องมีการกำหนดคลาสและเมธอด และเน้นเรื่องความถูกต้อง โปรแกรมที่เขียนในภาษาจาวาจะมีความสมบูรณ์กว่าการเขียนด้วยจาวาสคริปต์ จาวาสคริปต์เป็นภาษาแบบอินเทอร์พรีเตอร์ (Interpreter) ฉะนั้นเพียงเขียนคำสั่ง ในภาษาจาวาสคริปต์ เก็บไว้เป็นเท็กซ์ไฟล์ร่วมกับเว็บเพจ HTML ก็ทำงานได้แล้ว ไฟล์คำสั่งในภาษาจาวาสคริปต์ อาจมีส่วนขยายเป็น .htm หรือ .html เหมือนกับไฟล์เว็บเพจทั่วไป หรือมีส่วนขยายเป็น .js ก็ได้ แต่ภาษาจาวาเป็นภาษาแบบคอมไพเลอร์ (Compiler) คำสั่งในภาษาจาวาจึงต้องเขียนเก็บไว้เป็นเท็กซ์ไฟล์มีส่วนขยายเป็นคอตจาวา หลังจากนั้นต้องนำไฟล์ดังกล่าวไปผ่านการคอมไพล์ให้เป็นไบต์โค้ด (ไฟล์ที่มีส่วนขยายเป็นคอตคลาส) เสียก่อน จากนั้นจึงนำไฟล์มาสร้างเป็นอ็อบเจกต์และแอปพลิเคชันเพื่อใช้งานต่อไป

2.6.2 การทำงานของจาวาสคริปต์

จาวาสคริปต์เป็นภาษาสคริปต์เชิงวัตถุ (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนเอกสารด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ทำงานร่วมกับ ภาษา HTML และภาษาจาวาได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server) โดยมีลักษณะการทำงานดังนี้

1. Navigator JavaScript เป็น Client-Side JavaScript

ซึ่งหมายถึง จาวาสคริปต์ที่ถูกแปลทางฝั่งไคลเอนต์ (หมายถึงฝั่งเครื่อง คอมพิวเตอร์ของผู้ใช้ ไม่ว่าจะเป็นเครื่องพีซี เครื่องแมคอินทอช หรือ อื่น ๆ) จึงมีความเหมาะสมต่อการใช้งานของผู้ใช้ทั่วไปเป็นส่วนใหญ่

2. LiveWire JavaScript เป็น Server-Side JavaScript

ซึ่งหมายถึงจาวาสคริปต์ที่ถูกแปลทางฝั่งเซิร์ฟเวอร์ (หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ให้บริการเว็บ โดยอาจจะเป็นเครื่องของซัน ซิลิคอนกราฟิกส์ หรือ อื่นๆ) สามารถใช้ได้เฉพาะกับ Live Wire ของเน็ตสเคป โดยตรง

2.7 Document Object Model (DOM)

เป็นวิธีการอธิบายว่าข้อมูลต่างๆ ในเอกสาร HTML หรือ XML จัดเรียงตัวแบบเชิงวัตถุ (Object Oriented) อย่างไร DOM เป็นวิธีในการสร้าง API ให้สามารถควบคุมเนื้อหา โครงสร้าง เอกสารนี้ (Object Oriented) อย่างไร สำหรับการใช้งานเอกสาร HTML หรือ XML โดยผู้ใช้งานสามารถเข้าถึงเนื้อหาเอกสารได้โดยไม่ต้องรู้โครงสร้างเอกสารล่วงหน้า อย่างไรก็ตาม การเข้าถึงเนื้อหาเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และรูปแบบของเอกสารได้ เดิมทีนั้นเว็บเบราว์เซอร์แต่ละค่ายมีการพัฒนา DOM โดยใช้จาวาสคริปต์ที่แตกต่างกันออกไป ทาง W3C จึงได้ร่างมาตรฐานกลางที่เรียกว่า W3C Document Object Model (W3C DOM) โดยไม่ขึ้นกับภาษาโปรแกรมมิ่งภาษาใดภาษาหนึ่ง การจัดข้อมูลในรูปแบบ DOM นั้นใช้แผนภูมิต้นไม้ในการบอกความสัมพันธ์ระหว่างวัตถุแต่ละชิ้น อย่างไรก็ตาม DOM มีจุดอ่อนในเรื่องประสิทธิภาพเมื่อต้องประมวลผลเอกสารขนาดใหญ่ ซึ่ง DOM จำเป็นต้องอ่านเอกสารให้ครบทั้งหมดก่อน จึงจะเริ่มประมวลผล ข้อเสียนี้จึงเกิดการออกแบบ SAX ที่ใช้วิธีอ่านเอกสารทีละส่วนแทน

2.8 Crontab

คำสั่ง Crontab เป็นคำสั่งในการทำ Schedule ในการสั่งโปรแกรม หรือสคริปต์ต่าง ๆ ทำงานตามเวลาที่กำหนด บนระบบ Unix/ Linux ซึ่งอำนวยความสะดวกได้มากทีเดียว งานบางอย่างที่จำเป็นต้องทำซ้ำ ๆ ในเวลาเดียวกันเช่นสั่งให้โปรแกรมทำงาน ทุกวัน ทุกสัปดาห์ ทุกเดือน หรือทุกเที่ยงคืนของวันอาทิตย์ คำสั่งและตัวเลือกของ crontab มีดังนี้

#crontab filename การนำเอาคำสั่ง crontab เข้ามาจาก ไฟล์อื่น
 #crontab -e แก้ไข crontab ปัจจุบัน (ส่วนมากผมจะเพิ่มคำสั่งลงไปที่นี่เลย)
 #crontab -l ดูคำสั่ง crontab ทั้งหมดที่มีอยู่
 #crontab -r ลบคำสั่ง crontab ที่มีทั้งหมด
 #crontab -u user เป็นคำสั่งของผู้ดูแลระบบเท่านั้น เพื่อใช้ดู แก้ไข ลบ crontab ของ user แต่
 ละคน เมื่อเรียกคำสั่งตามข้างบนแล้ว crontab จะเข้าสู่ระบบการกำหนด หรือแก้ไข ซึ่งการกำหนดหรือแก้ไขนี้จะเหมือนกับการใช้งาน vi รูปแบบของคำสั่ง crontab มีทั้งหมด 6 필ด์ดังนี้

1	2	3	4	5	6
minute	hour	day	month	weekday	command

รูปที่ 2.3 รูปแบบของคำสั่ง crontab ทั้ง 6 필ด์

ฟิลด์ 1 - 5 เป็นการกำหนดเวลา และฟิลด์ ที่ 6 เป็นการกำหนดคำสั่ง

ความหมายของแต่ละฟิลด์ มีดังต่อไปนี้

minute มีค่า 0 - 59 เวลาเป็นนาที จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงนาทีที่กำหนด

hour มีค่า 0 - 23 เวลาเป็นชั่วโมง จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงชั่วโมงที่กำหนด

day มีค่า 1 - 31 เวลาเป็นวัน จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงวันที่กำหนด

month มีค่า 1 - 12 เวลาเป็นเดือน จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงเดือนที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนสิทธิ์ในเนื้อหา เมื่อผู้ดูแลระบบหรือเจ้าหน้าที่ดำเนินการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

weekday มีค่า 0 - 6 วันจะงแต่ละสัปดาห์ มีค่าดังนี้ (อาทิตย์ = 0, จันทร์ = 1, อังคาร = 2, พุธ = 3, พฤหัส = 4, ศุกร์ = 5 ,เสาร์ = 6)

command คำสั่ง เราสามารถกำหนดคำสั่งได้มากมาย รวมทั้งสคริปต์ต่าง ๆ ตามที่เราต้องการ

ตัวอย่างการเรียกใช้งาน Crontab

ตัวอย่างที่ 1

```
45 04 * * * /usr/sbin/chkrootkit && /usr/bin/updatedb
```

หมายถึงเป็นการกำหนดให้ crontab มีการรัน updatedb ซึ่งจะเป็นการอัปเดต slocate database ทุก ๆ เช้า เวลา 4:45am วิธีการทำก็คือให้พิมพ์คำว่า crontab -e และพิมพ์ข้อความหนึ่งบรรทัดดังกล่าวได้ บรรทัดต่อไปนี้ :

```
# m h dom mon dow command
```

แล้วให้บันทึกการแก้ไขและออกจากอีดีเตอร์

ถ้าเรากำหนด crontab ไม่ถูกต้องตามรูปแบบจะมีข้อความฟ้องดังตัวอย่างดังนี้ :

```
"/tmp/crontab.P7vQuf/crontab":2: bad day-of-month
errors in crontab file, can't install.
```

```
Do you want to retry the same edit?
```

เราสามารถ ใช้ double-ampersand (&&) ในส่วนของ command section เพื่อรันคำสั่งหลาย ๆ คำสั่ง ที่ติดต่อกัน (consecutively) ได้

ตัวอย่างที่ 2

```
45 04 * * * /usr/sbin/chkrootkit && /usr/bin/updatedb
```

หมายถึงเป็นการรัน chkrootkit และ updatedb ที่เวลา 4:45am ของทุกวัน

ตัวอย่างที่ 3

```
10 3 * * * /usr/bin/foo
```

หมายถึงเป็น /usr/bin/foo ที่เวลา 2:10pm ของทุกวันจันทร์

2.9 cURL

2.9.1 ความหมายของ cURL

Function กลุ่ม cURL (Libcurl) พัฒนาโดย Daniel Stenberg ทำหน้าที่เป็นไลบรารีหุ้มห่อ (Wrapper Library) สำหรับติดต่อกับเซิร์ฟเวอร์ผ่านโปรโตคอลต่างๆ เช่น http, https, ftp, gopher, telnet, dict, file, และ ldap protocols. นอกจากนี้ยังรองรับ HTTPS certificates, HTTP POST, HTTP PUT, FTP Uploading , HTTP form based upload, proxies, cookies, และ User + Password Authentication โดยการใช้งาน cURL Functions สามารถใช้งานร่วมกับภาษาต่างๆ มากกว่า 30 ภาษา หนึ่งในนั้นคือภาษา PHP ซึ่ง cURL Functions สามารถทำงานร่วมกับภาษา PHP ตั้งแต่เวอร์ชัน 4.0.2 เป็นต้นไป

2.9.2 การติดตั้ง cURL

การที่จะเรียกใช้งาน Libcurl ใน PHP นั้นจำเป็นที่จะต้องติดตั้ง libcurl package เสียก่อน สำหรับระบบที่ทำงานบน Windows จำเป็นที่จะต้องทำการติดตั้งไฟล์ที่ชื่อ libeay32.dll และ sslay32.dll เพิ่มลงไปไนไดเรกทอรี ..windows/system เสียก่อน ต่อจากนั้นให้ทำการเปิดใช้งาน cURL Functions โดยการแก้ไขไฟล์ php.ini ด้วยการลบเครื่องหมาย ; หน้าบรรทัดที่มีคำว่า “; extension=php_curl.dll” ในไฟล์ php.ini ออกแล้วทำการรีสตาร์ทเว็บเซิร์ฟเวอร์ (ในที่นี้คือทำการรีสตาร์ท Apache Server) จึงจะสามารถใช้งาน cURL Functions ร่วมกับภาษา PHP ได้

2.9.3 คำสั่งในการเรียกใช้งาน cURL

ขั้นตอนการใช้งาน cURL มีดังนี้

1. ตั้งค่าเริ่มต้นให้ cURL session ด้วยคำสั่ง curl_init()
2. เซตค่าตัวเลือกต่างๆ สำหรับการรับส่งข้อมูลด้วยคำสั่ง curl_setopt()
3. รัน cURL ด้วยคำสั่ง curl_exec()
4. หลังจากทำงานเสร็จ ทำการปิดการใช้งาน cURL session ด้วยคำสั่ง curl_close()

เราจะได้โค้ดสำหรับการดึงข้อมูลด้วย cURL ดังนี้

```
$url = curl_init($init.$station_id);
curl_setopt($url, CURLOPT_FILE, $file);
curl_setopt($url, CURLOPT_HEADER, 0);
curl_exec($url);
curl_close($url);
```

บทที่ 3

การใช้งาน Google Maps

ระบบแผนที่ Google Maps มีองค์ประกอบอย่างน้อย 2 ส่วน ส่วนแรกคือ Application Program Interface (API) และทรัพยากรสนับสนุนจากเซิร์ฟเวอร์ของผู้ให้บริการ และส่วนที่ 2 คือ โปรแกรมและทรัพยากรของผู้พัฒนาแอปพลิเคชันนั้น (ติดตั้งอยู่ที่เว็บไซต์ของผู้พัฒนา) ตัวอย่างกรณีของระบบเรา ก็จะหมายถึงหน้าเว็บเพจที่สร้างขึ้นให้มีส่วนประกอบที่เป็นแผนที่ที่สามารถทำงานแบบโต้ตอบกับผู้ใช้ได้ โดยมีแผนที่ฐานเป็นส่วนที่ถูกล็อกไว้ให้แล้ว และมีข้อมูลแผนที่ของเราซ้อนทับในลักษณะหมุดปัก (ที่สามารถแสดงข้อความซึ่งแฝงอยู่เมื่อผู้ใช้คลิกที่หมุดเหล่านั้น) หรือลักษณะอื่นที่สลับซับซ้อนกว่านั้น

3.1 ความรู้ที่ควรมีในการใช้งาน Google Maps API

ในการจัดทำระบบแผนที่ Google Maps ระดับต้นจำเป็นต้องใช้ความรู้พื้นฐานในการสร้างเว็บเพจ ด้วยภาษา HTML / XHTML การเขียน โปรแกรมจาวาสคริปต์ ความรู้เรื่องการแผนที่ และภูมิศาสตร์ และ ความรู้ในด้านเทคโนโลยี XML ในระดับปฏิบัติการที่สูงขึ้นมีความต้องการที่สลับซับซ้อนมากขึ้นตามลำดับ

3.1.1 ความรู้พื้นฐานเบื้องต้น

1. HTML / XHTML
2. Javascript
3. ความรู้วิชาภูมิศาสตร์ และการแผนที่

3.1.2 ความรู้เพิ่มเติมที่จำเป็นสำหรับงานพัฒนาขั้นกลาง-สูง

1. XML เทคโนโลยี
2. AJAX (Javascript + XML)
3. JSON (JavaScript Object Notation)

3.1.3 ความรู้เพิ่มเติมสำหรับงานใหญ่ งานพิเศษ

1. Server side scripting (PHP, ASP, JSP เป็นต้น)
2. Web database software (MySQL , Database Sever อื่นๆ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 ความรู้อื่นที่เกี่ยวข้อง มีทั้งที่เป็นเรื่องกราฟิก แผนที่ และ อื่นๆ

1. Web map service (WMS) concept
2. SVG (Scalable Vector Graphic)
3. GML (Geographic Markup Language)
4. GIS, GPS, Cartography
5. ระบบแผนที่อื่นๆ

3.2 ขั้นตอนหลักในการดำเนินการจัดทำระบบแผนที่ Google Maps

1. ยื่นคำขอ API Key สำหรับ URL ของเว็บไซต์ที่จะใช้บริการ Google Maps API เว็บไซต์นั้นเป็นที่ที่เราจะนำเว็บเพจที่มี Google Maps เป็นองค์ประกอบไปวางไว้
2. เขียน โปรแกรมสร้างเว็บเพจที่ใช้ Google Maps API ช่วยสร้างแผนที่
3. อัปเดตงานที่ทำเสร็จแล้วขึ้นสู่เว็บไซต์ และทดลองใช้งาน

เค้าโครงของเพิ่มข้อมูล HTML สำหรับการสร้างระบบแผนที่ Google Maps อย่างง่าย

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<title>My Google Maps Mash-up</title>
<style type="text/css">
v\:* {
behavior:url(#default#VML);
}
</style>
<script src="http://maps.google.com/maps?file=api&v=2&key=abcdefg..."
type="text/javascript"></script>
</head>
<body onload='onloadFunction()' onunload='unloadGmap0'>
```

```

<!-- body content here -->
<!-- some javascript here -->
</body>
</html>

```

ข้อความ "v=2" ซึ่งแทรกอยู่ใน "http://maps.google.com/maps?file=api&v=2 ..." เป็นพารามิเตอร์ระบุความต้องการขอใช้ Google Maps API เวอร์ชัน 2 เพื่อทำการประมวลผล ในส่วน

```

<!-- body content here --> ควรมี <div id='map' /> เพื่อทำหน้าที่เป็นขอบเขตแสดงผลแผนที่และ
<!-- some javascript here --> เตรียมไว้เป็นที่ของจาวาสคริปต์ซึ่งจะนำมาแทรกภายหลัง

```

3.3 การใช้งานคำสั่งที่สำคัญใน Google Maps API พร้อมทั้งตัวอย่าง

ตัวอย่างทั้งหมดที่ยกมาแสดงในที่นี้จะมีให้เฉพาะส่วนประกอบที่เกี่ยวกับการใช้ Google Maps API เท่านั้นเพื่อความกระชับ ทั้งนี้ยกเว้นตัวอย่างแรกที่ทำหน้าที่แนะนำให้กับผู้ใช้ นอกจากนี้ยังได้ละเว้นส่วนประกอบของโปรแกรมที่ทำหน้าที่สำคัญที่ควรมีในการทำงานจริง เช่น ตรวจสอบความสามารถของเบราว์เซอร์ที่ผู้ใช้ติดตั้งอยู่ว่าทำงานเข้ากันได้หรือไม่ การตรวจสอบป้องกันการทำงานผิดพลาด การทำให้มีคุณลักษณะเป็นมิตรกับผู้ใช้ ฯลฯ

ตัวอย่าง 1

ตัวอย่างแรกนี้แสดงการสร้างแผนที่อย่างง่ายให้ศูนย์กลางขอบเขตแผนที่อยู่ที่เมือง Palo Alto แห่งรัฐแคลิฟอร์เนียซึ่งมีพิกัดภูมิศาสตร์ ละติจูด / ลองจิจูดเป็น 37.4419, -122.1419 องศา ตามลำดับ (สำหรับกรุงเทพฯ ให้ใช้ 13.80608, 100.57983)

```

<!-- This web page displays a 500x300 map centered on Palo Alto, California: -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Google Maps JavaScript API Example</title>
<script src="http://maps.google.com/maps?file=api&v=2&key=abcdefg"
type="text/javascript"></script>

```

```

<script type="text/javascript">
//

function load() {
if (GBrowserIsCompatible()) {
var map = new GMap2(document.getElementById("map"));
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
}
}
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body onload="load()" onunload="GUnload()"&gt;
&lt;div id="map" style="width: 500px; height: 300px"&gt;&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>
</div>
<div data-bbox="131 522 216 543" data-label="Section-Header">
<h2>ตัวอย่าง 2</h2>
</div>
<div data-bbox="131 563 557 587" data-label="Text">
<p>ตัวอย่างแผนที่ซึ่งเคลื่อนตำแหน่งได้พร้อมมีแอนิเมชัน</p>
</div>
<div data-bbox="131 605 870 632" data-label="Text">
<p>แผนที่ในตัวอย่างนี้เมื่อเปิดแสดงผลแล้ว จะนิ่งอยู่ 2 วินาที แล้วจะเลื่อนภาพไปยังตำแหน่งใหม่</p>
</div>
<div data-bbox="131 659 844 731" data-label="Text">
<pre>
var map = new GMap2(document.getElementById("map"));
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
window.setTimeout(function() {map.panTo(new GLatLng(37.4569, -122.1569));},1000);
</pre>
</div>
<div data-bbox="65 898 928 950" data-label="Page-Footer">
<p>เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า<br/>ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้</p>
</div>
```



รูปที่ 3.1 แผนที่ซึ่งเคลื่อนตำแหน่งได้

3.3.1 การกำหนดให้มีตัวควบคุมบนแผนที่

เราสามารถสร้างองค์ประกอบที่ทำหน้าที่โต้ตอบกับผู้ใช้เพื่อควบคุมการใช้งานแผนที่ได้ โดยใช้เมธอด `addControl`

ตัวอย่าง 3

ตัวอย่างต่อไปนี้ แสดงการสร้างตัวควบคุมแผนที่ 2 ตัว ซึ่งได้แก่

1. `GSmallMapControl` (ใช้เลื่อนภาพ/ ขยายภาพ) และ
2. `GMapTypeControl` (ใช้เลือกแสดงภาพบนฉากหลังเป็น Map / Satellite หรือ Hybrid)

ให้เป็นองค์ประกอบส่วนหนึ่งบนแผนที่

```
var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ตัวรับฟังอุบัติการณ์ (Event Listener)

เนื่องจากอุบัติการณ์ทั้งหลายต้องขึ้นทะเบียน (Register) ไว้ในระบบก่อนจึงจะมีผลในการใช้งาน กล่าวคือ สามารถตอบสนองต่อการใช้งานแผนที่แบบโต้ตอบได้ การจัดระบบเพื่อรับฟังอุบัติการณ์ดังกล่าวทำขึ้นได้โดยใช้ GEvent.addListener method เป็นตัวกระทำ

ตัวอย่าง 4

ตัวอย่างต่อไปนี้เป็นสาธิตการใช้อุบัติการณ์ "moveend" เป็นตัวจุด (หรือลั่นไก) ให้เกิดการแสดงค่าพิกัดละติจูด-ลองจิจูด ของจุดกลางแผนที่ค่าใหม่ล่าสุด โดยแสดงผลทุกครั้งที่มีการเลื่อนแผนที่จากที่หนึ่งไปยังที่ใหม่เสร็จสิ้นลง

```
var map = new GMap2(document.getElementById("map"));
GEvent.addListener(map, "moveend", function() {
    var center = map.getCenter();
    document.getElementById("message").innerHTML = center.toString();
});
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
```

3.3.3 การเปิด Info Window

"Info Window" เป็นบัญชีแสดงผลแบบผุดขึ้น (Pop-up) ที่เปิดได้ทั้งข้อความและ/หรือรูปภาพ มีไว้เพื่อให้นำเสนอข้อมูลเชิงปริภูมิ (Spatial data) ณ ตำแหน่งที่ต้องการบนแผนที่ สร้างขึ้นได้ด้วย openInfoWindow method โดยป้อนค่าพิกัดตำแหน่งและ html element เป็นองค์ประกอบ ทั้งนี้ html element ก็คือส่วนของเนื้อหาที่เขียนเป็นภาษา html ตามที่ต้องการนำเสนอใน Info Window หรือใช้จาวาสคริปต์สร้างเป็น html element ก็ได้

ตัวอย่าง 5

ตัวอย่างนี้จะเปิด "Info Window" สำหรับตำแหน่งจุดกลางแผนที่โดยมีข้อความ "Hello, world" แสดงอยู่ภายในบัญชี

```

var map = new GMap2(document.getElementById("map"));
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
map.openInfoWindow(map.getCenter(),document.createTextNode("Hello, world"));

```



รูปที่ 3.2 การเรียกใช้ Info Window

3.3.4 วัตถุซ้อนทับบนแผนที่ (Map Overlays)

Overlays หมายถึงวัตถุที่สร้างขึ้นวางทับอยู่บนแผนที่ถนนและ/ หรือภาพถ่ายดาวเทียมที่เป็นฉากหลัง (เช่น marker, polyline, polygon, และ ground overlay) ตัวอย่างนี้แสดงการสร้าง marker จำนวน 10 ตัวเพื่อวางไว้ที่ตำแหน่งต่างๆ แบบสุ่ม และเขียน polyline ซึ่งภายในเส้นประกอบขึ้นด้วยจุด 5 จุดแบบสุ่ม ในรหัสของตัวอย่างข้างล่าง คำสั่ง GMarker(point) ทำหน้าที่เขียน marker ณ ตำแหน่งที่กำหนดด้วยค่าละติจูด- ลองจิจูดภายในตัวแปร "point" โดยใช้สัญลักษณ์ปริยาย (defaulticon)ซึ่งมีลักษณะเป็นหมุดปักอยู่บนแผนที่

ตัวอย่าง 6

```

var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

map.setCenter(new GLatLng(37.4419, -122.1419), 13);
/* สร้าง marker วางไว้ในที่ต่างๆ แบบสุ่มบนแผนที่จำนวน 10 ตัว */
var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
    var point = new GLatLng(southWest.lat() + latSpan * Math.random(),
        southWest.lng() + lngSpan * Math.random());
    map.addOverlay(new GMarker(point));
}
/* สร้าง polyline เส้นหนึ่งให้มีจุดจากต้นถึงปลายเส้น 5 จุดแบบสุ่ม โดยทำการจัดลำดับจุดเหล่านั้น
ตามค่าลองจิจูดเพื่อหลีกเลี่ยงไม่ให้เส้นพาดผ่านตัวเอง */
var points = [];
for (var i = 0; i < 5; i++) {
    points.push(new GLatLng(southWest.lat() + latSpan * Math.random(),
        southWest.lng() + lngSpan * Math.random()));
}
points.sort(function(p1, p2) {
    return p1.lng() - p2.lng();
});
map.addOverlay(new GPolyline(points));

```

3.3.5 การตอบสนองต่ออุบัติการณ์ "click"

การคลิกทำให้เกิดการกระทำอย่างหนึ่ง เมื่อผู้ใช้คลิกตำแหน่งบนแผนที่ จะต้องทำการ register ตัวรับฟัง (listener) เพื่อทำหน้าที่รับฟังอุบัติการณ์ "click" ที่เกิดขึ้นบน GMap2 instance เมื่อมีอุบัติการณ์ชนิดนั้นเกิดขึ้น ตัวรับฟังจะรัน ไก่ ก่อให้เกิด event handler จะได้รับอาร์กิวเมนต์ 2 ตัว คือ

1. marker ตัวที่ถูกคลิก (ถ้ามี)
2. ค่าพิกัดตำแหน่ง GLatLng ของตำแหน่งที่ถูกคลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

ถ้าไม่มี marker ใดถูกคลิกอาร์กิวเมนต์ตัวแรกจะมีค่าเป็น null
หมายเหตุ: ขณะนี้ ในกลุ่ม built-in overlay มีเพียง Marker เท่านั้น ที่สามารถตอบสนองอุบัติการณ์ "click" ได้ ส่วน overlay ชนิดอื่น เช่น GPolyline ยังไม่มีคุณสมบัติดังกล่าว

ตัวอย่าง 7

ในตัวอย่างต่อไปนี้ เมื่อผู้ใช้แผนที่คลิกที่ตำแหน่งบนแผนที่ที่ไม่ถูกตัว marker ที่มีอยู่แล้ว จะเกิด marker ตัวใหม่เพิ่มขึ้น ณ ตำแหน่งนั้น แต่ถ้าคลิกโดนตัว marker ใดที่มีอยู่ marker นั้นจะถูกลบทิ้งไป

```
var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
GEvent.addListener(map, "click", function(marker, point) {
  if (marker) {
    map.removeOverlay(marker);
  }
  else {
    map.addOverlay(new GMarker(point));
  }
});
```

3.3.6 การเปิด Info Window กำกับ Marker เพื่อแสดงข้อมูลแผนที่

ตัวอย่างต่อไปนี้จะแสดงการใช้ custom info window แสดงผลกำกับ marker โดยให้ตอบสนองต่ออุบัติการณ์คลิก (click event) ทั้งนี้ให้มีผลต่อการคลิก marker ทุกตัว ในการนี้เราใช้ความสามารถที่เป็นข้อดีของ JavaScript function closure ให้เกิดการเปลี่ยนแปลงเนื้อหาที่นำเสนอภายใน info window ที่เป็นเรื่องของ marker แต่ละตัวเป็นการเฉพาะ

ตัวอย่าง 8

การใช้ custom info window แสดงผลกำกับ marker โดยให้ตอบสนองต่ออุบัติการณ์คลิก เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
/* สร้าง marker ตัวหนึ่งให้ตรงตำแหน่งที่ต้องการ ใช้เลขหมายกำกับบน marker ตามที่ถูกระบุ
*/
function createMarker(point, number) {
var marker = new GMarker(point);
GEvent.addListener(marker, "click", function() {
marker.openInfoWindowHtml("Marker #<b>" + number + "</b>");
});
return marker;
}
/* สร้าง markers บนแผนที่จำนวน 10 ตัว ให้วางอยู่ในตำแหน่งต่างๆแบบสุ่ม */
var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
var point = new GLatLng(southWest.lat() + latSpan * Math.random(),
southWest.lng() + lngSpan * Math.random());
map.addOverlay(createMarker(point, i + 1));
}

```



รูปที่ 3.3 การใช้ custom info window แสดงผลกำกับ marker

3.3.7 Info Window แบบหลายแผ่นเอกสารซ้อนกัน

การจัดให้ Info Window มีรูปแบบที่สลับซับซ้อนแบบ Multiple named tabs สามารถทำได้โดยใช้ เมธอดสองตัวซึ่งได้แก่ `openInfoWindowTabs()` และ `GInfoWindowTab`

ตัวอย่าง 9

ตัวอย่างนี้แสดงการสร้าง Tabbed info window อย่างง่าย โดยกำหนดให้ปรากฏขึ้นด้านบนของ marker ตัวหนึ่ง

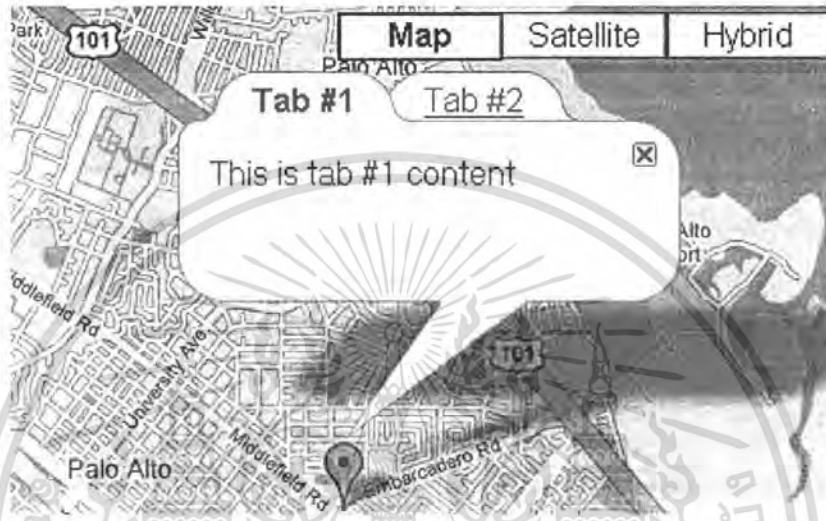
```
var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
/* จัดเนื้อหาที่นำเสนอภายใน info window ของเรา */
var infoTabs = [
  new GInfoWindowTab("Tab #1", "This is tab #1 content"),
  new GInfoWindowTab("Tab #2", "This is tab #2 content")
];
/* วาง marker ที่ตำแหน่งกลางแผนที่และเปิด info window โดยอัตโนมัติ */
var marker = new GMarker(map.getCenter());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GEvent.addListener(marker, "click", function() {
marker.openInfoWindowTabsHtml(infoTabs);
});
map.addOverlay(marker);
marker.openInfoWindowTabsHtml(infoTabs);

```



รูปที่ 3.4 การสร้าง Tabbed info window

3.3.8 การสร้างสัญลักษณ์ (Icons)

สัญลักษณ์หรือ ไอคอนในที่นี้หมายถึงรูปภาพที่สื่อความหมายอย่างใดอย่างหนึ่ง ทำหน้าที่เป็นองค์ประกอบในการสร้าง marker หรือหมุดที่วางหรือปักบนแผนที่

ตัวอย่าง 10

ตัวอย่างนี้แสดงการสร้างสัญลักษณ์หรือ ไอคอนชนิดใหม่เพิ่มเติมส่วนที่มีอยู่แล้วโดยใช้ Google Ride Finder "mini" marker เป็นแบบอย่าง ในกระบวนการนี้มีการกำหนดสิ่งต่างๆ ที่ใช้ประกอบกันเป็น marker ที่ประสงค์ ซึ่งได้แก่ รูปภาพส่วนหน้า (Foreground image) รูปภาพส่วนเงา (Shadow image) และ พิกัดตำแหน่งภูมิศาสตร์ซึ่งระบุตำแหน่งที่เราต้องการปัก marker บนแผนที่ และ marker นั้นเป็นตัวที่จะมี info window ผุดขึ้นเมื่อมีการแสดงข้อมูลที่กำกับอยู่ ซึ่งจะเกิดขึ้นเมื่อผู้ใช้คลิกที่ตัว marker

```

var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
/* สร้าง marker ขนาดจั่วโดยใช้ base icon ที่เตรียมไว้ */
var icon = new GIcon();
icon.image = "http://labs.google.com/ridefinder/images/mm_20_red.png";
icon.shadow = "http://labs.google.com/ridefinder/images/mm_20_shadow.png";
icon.iconSize = new GSize(12, 20);
icon.shadowSize = new GSize(22, 20);
icon.iconAnchor = new GPoint(6, 20);
icon.infoWindowAnchor = new GPoint(5, 1);
/* สร้าง marker จำนวน 10 ตัววางไว้ในที่ต่างๆ แบบสุ่มบนแผนที่ */
var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
var point = new GLatLng(southWest.lat() + latSpan * Math.random(),
    southWest.lng() + lngSpan * Math.random());
map.addOverlay(new GMarker(point, icon));
}

```

3.3.9 การใช้ Icon Class

ในทางปฏิบัติเราอาจใช้ icon ต่างๆ เป็นสัญลักษณ์แผนที่โดยสร้างให้มีส่วนที่แตกต่างกันตรง foreground เพียงอย่างเดียว แต่เหมือนกันที่ shape และ shadow ซึ่งวิธีดำเนินการที่ง่ายสุดเพื่อให้ได้คุณสมบัติตามแบบที่กล่าวคือ การใช้ Copy constructor ของ GIcon class ซึ่งจะช่วยคัดลอกคุณสมบัติทั้งหมดไปยัง icon ตัวใหม่ซึ่งจะเป็นตัวที่เราทำการปรับตั้งค่าให้ได้คุณสมบัติที่ประสงค์

ตัวอย่าง 11

```

var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
/* สร้าง base icon เตรียมไว้รองรับการทำ marker โดยกำหนดค่า shadow มีติต่างๆ และ ฯลฯ ของ
icon */
var baseIcon = new GIcon();
baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.png";
baseIcon.iconSize = new GSize(20, 34);
baseIcon.shadowSize = new GSize(37, 34);
baseIcon.iconAnchor = new GPoint(9, 34);
baseIcon.infoWindowAnchor = new GPoint(9, 2);
baseIcon.infoShadowAnchor = new GPoint(18, 25);
/* ฟังก์ชันใช้สร้าง marker แบบที่แต่ละตัวมีส่วนประกอบเป็นตัวอักษรตรงตามค่าดัชนีของมันเอง
*/
function createMarker(point, index) {
var letter = String.fromCharCode("A".charCodeAt(0) + index);
var icon = new GIcon(baseIcon);
icon.image = "http://www.google.com/mapfiles/marker" + letter + ".png";
var marker = new GMarker(point, icon);
GEvent.addListener(marker, "click", function() {
marker.openInfoWindowHtml("Marker <b>" + letter + "</b>");
});
return marker;
}
/* สร้าง marker จำนวน 10 ตัววางไว้แบบสุ่มบนแผนที่ */
var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();

```

```

var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
var point = new GLatLng(southWest.lat() + latSpan * Math.random(),
    southWest.lng() + lngSpan * Math.random());
map.addOverlay(createMarker(point, i));
}

```

3.3.10 การใช้ XML และ Asynchronous HTTP เพื่อนำเข้าข้อมูลแผนที่

ตัวอย่างนี้แสดงการดาวน์โหลดไฟล์ข้อมูลแบบสถิตย์ (static file) ชื่อ "data.xml" ซึ่งบรรจุรายการค่าพิกัด ละติจูด / ลองจิจูด ในรูปแบบ XML โดยใช้เมธอด GDownloadUrl หลังจากที่ได้ดาวน์โหลดเสร็จ ก็ทำการแปลงข้อมูล (parse) ด้วยเมธอด GXml.parse และนำค่าที่ได้ไปประมวลผลต่อตามลำดับ โดยให้ผลลัพธ์เป็น marker วางบนแผนที่ในตำแหน่งตามค่าที่กำหนดในเอกสาร XML ที่ใช้เป็นข้อมูลป้อนเข้า

ตัวอย่าง 12

```

var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);

// ทำการ download เพิ่มข้อมูลชื่อ data.xml และ load เข้าสู่แผนที่
// รูปแบบและเนื้อหาของข้อมูลเป็นทำนองนี้:
// <markers>
// <marker lat="37.441" lng="-122.141"/>
// <marker lat="37.322" lng="-121.213"/>
// </markers>
GDownloadUrl("data.xml", function(data, responseCode) {
var xml = GXml.parse(data);
var markers = xml.documentElement.getElementsByTagName("marker");
for (var i = 0; i < markers.length; i++) {
var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),

```

```
parseFloat(markers[i].getAttribute("lng"))));
map.addOverlay(new GMarker(point));
}
});
```

3.3.11 ตัวควบคุมแผนที่แบบประดิษฐ์เอง (Custom Map Controls)

ใน Google Maps เวอร์ชัน 2 ผู้ใช้สามารถสร้างตัวควบคุม (เช่น ตัวควบคุมการย่อ/ ขยาย หรือเลื่อนภาพ) ขึ้นใหม่ได้โดยทำ subclass จากตัวคลาส GControl ที่มีอยู่แล้ว ตัวอย่างที่จะแสดง เป็นการสร้างตัวควบคุมแบบมีข้อความกำกับที่ทำหน้าที่ย่อขยายภาพ ซึ่งจะแตกต่างจากตัวย่อขยายภาพ มาตรฐานแบบกราฟิกล้วนที่มีอยู่แล้ว

ในตัวอย่างนี้มีการใช้คลาสที่เกี่ยวข้องชื่อ GTextualZoomControl เพราะมีเมธอดที่จำเป็นในการทำงาน ดังนี้

1. initialize() method ใช้สร้าง DOM element ของตัวควบคุมแผนที่ที่ต้องการสร้างขึ้น
2. getDefaultPosition method เพื่อส่งคืนค่า GControlPosition ที่จะใช้เมื่อตัวควบคุมแผนที่ของเราได้สร้างขึ้นสำเร็จ
3. setButtonStyle_ method สำหรับตั้งค่า CSS ให้ได้รูปแบบของตัวควบคุมที่เหมาะสม

ตัวควบคุมแผนที่ทุกตัวต้องจัดวางไว้ใน map container ซึ่งเป็น div element ตัวหนึ่งเพื่อให้สามารถเข้าถึงได้ด้วย GMap2.getContainer() method

ตัวอย่าง 13

```
/* ตัว TextualZoomControl เป็นตัวควบคุมที่เป็นปุ่มมีข้อความ "Zoom In" และ "Zoom Out" มีความแตกต่างจากตัวควบคุมแบบสัญลักษณ์ (icon) ที่ใช้อยู่บน Google Maps ตามปกติ
*/
function TextualZoomControl() {
}
TextualZoomControl.prototype = new GControl();
```

/* สร้าง div element เพื่อรองรับปุ่มควบคุมแผนที่แต่ละตัวจนครบทุกตัว แล้วจัดวาง div element

ทั้งหมดไว้ด้วยกันใน div container ตัวหนึ่ง ซึ่งจะถูกจัดวางไว้ใน map container ต่อไป

*/

```
TextualZoomControl.prototype.initialize = function(map) {
var container = document.createElement("div");
var zoomInDiv = document.createElement("div");
this.setButtonStyle_(zoomInDiv);
container.appendChild(zoomInDiv);
zoomInDiv.appendChild(document.createTextNode("Zoom In"));
GEvent.addDomListener(zoomInDiv, "click", function() {
map.zoomIn();
});
var zoomOutDiv = document.createElement("div");
this.setButtonStyle_(zoomOutDiv);
container.appendChild(zoomOutDiv);
zoomOutDiv.appendChild(document.createTextNode("Zoom Out"));
GEvent.addDomListener(zoomOutDiv, "click", function() {
map.zoomOut();
});
map.getContainer().appendChild(container);
return container;
}
```

/* จัดตำแหน่งของตัวควบคุมที่สร้างขึ้น ไว้ที่ด้านบนซ้ายของแผนที่ และตั้งค่า padding ไว้ 7 หน่วย
จุดภาพ ค่าอื่นๆ ปล่อยให้ไปตามค่าปริยาย */

```
TextualZoomControl.prototype.getDefaultPosition = function() {
return new GControlPosition(G_ANCHOR_TOP_LEFT, new GSize(7, 7));
}
```

/* กำหนดรูปลักษณะของตัวควบคุมแผนที่ซึ่งจะสร้างให้เป็นปุ่มกด โดยตั้งค่า CSS ตามความต้องการ
การ */

```
TextualZoomControl.prototype.setButtonStyle_ = function(button) {
button.style.textDecoration = "underline";
```

```

button.style.color = "#0000cc";
button.style.backgroundColor = "white";
    button.style.font = "small Arial";
button.style.border = "1px solid black";
button.style.padding = "2px";
button.style.marginBottom = "3px";
button.style.textAlign = "center";
button.style.width = "6em";
button.style.cursor = "pointer";
}
var map = new GMap2(document.getElementById("map"));
map.addControl(new TextualZoomControl());
map.setCenter(new GLatLng(37.441944, -122.141944), 13);

```

3.3.12 วัตถุวางบนแผนที่แบบทำเป็นการเฉพาะ (Custom Overlays)

วัตถุวางบนแผนที่แบบมาตรฐานมีอยู่ 4 อย่าง (GMarker, GPolyline, GPolygon และ Ground Overlay) อาจไม่เพียงพอหรือสะดวกในการออกแบบงานแผนที่ Google Maps API เวอร์ชัน 2 นี้จึงเปิดโอกาสให้ผู้ใช้สร้างวัตถุ overlay ที่ทำขึ้นเป็นการเฉพาะเพิ่มเติม เพื่อให้การออกแบบและประกอบแผนที่ทำได้ดี และหลากหลายขึ้นกว่าเดิม

ในหัวข้อนี้จะแสดงการสร้าง Map overlay ลักษณะรูปสี่เหลี่ยมผืนผ้า (rectangle) โดยผลที่ได้จะเป็น subclass ของคลาส GOverlay ทำนองเดียวกับ marker และ polyline ซึ่งเป็น map overlay เพียง 2 ชนิดที่มีอยู่แล้ว ในการเขียนโปรแกรมจะต้องทำให้ rectangle มีเมธอด 4 ตัวที่จำเป็นขั้นต้น ซึ่งได้แก่ initialize(), remove(), copy(), และ redraw()

initialize()	ทำหน้าที่สร้าง DOM elements ให้ส่งผลเป็นรูปสี่เหลี่ยมผืนผ้าวางบนแผนที่ตามต้องการ
remove()	ทำหน้าที่กำจัดตัวเองออกไปจากแผนที่
copy()	ทำหน้าที่สำเนาตัวเองเพื่อนำไปใช้เป็นอินสแตนซ์ตัวใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

redraw() ทำหน้าที่สร้างตัวเองขึ้นใหม่บนแผนที่ให้ถูกต้องตรงตำแหน่งตามมาตราส่วนและการฉายแผนที่ในสถานะปัจจุบัน

DOM element ทุกตัวที่ใช้ประกอบเป็น overlay จะถูกจัดไว้บน map pane ซึ่งมีค่า z-order ตามลำดับที่ในการแสดงผล เช่น polylines ซึ่งถูกกำหนดให้อยู่ระดับเดียวกับ map layer จะถูกวางไว้ที่ G_MAP_MAP_PANE รูปเงาของ Markers ถูกวางที่ G_MAP_MARKER_SHADOW_PANE ตัว Marker ถูกจัดไว้ที่ G_MAP_MARKER_PANE

การจัดวาง overlay elements ให้อยู่บน pane ที่เหมาะสมช่วยให้เกิดการแสดงผลที่สมควร กล่าวคือ polylines อยู่ต่ำกว่า marker shadows และ info window แต่อยู่เหนือกว่า overlays อื่นบนแผนที่ ในตัวอย่างต่อไปนี้ overlay ของเราจะถูกกำหนดให้อยู่ระดับเดียวกับแผนที่ ดังนั้นจึงต้องใช้ G_MAP_MAP_PANE ซึ่งมีค่า z-order น้อยสุด ทั้งนี้เป็นไปในแนวเดียวกันกับ GPolyline ซึ่งเป็น overlay ปกติ

ตัวอย่าง 14

รูปสี่เหลี่ยมผืนผ้า (Rectangle) ที่เราจะสร้างขึ้นเป็น overlay อย่างง่าย จะมีขอบเขต ละติจูด / ลองจิจูด บนแผนที่ตามที่ผู้ใช้กำหนดตามใจชอบ ทั้งนี้รวมถึงลักษณะเส้นขอบเขตซึ่งกำหนดได้ด้วย ความหนา (weight) สี (color) และ ระดับความทึบแสง (opacity)

```
function Rectangle(bounds, opt_weight, opt_color) {
  this.bounds_ = bounds;
  this.weight_ = opt_weight || 2;
  this.color_ = opt_color || "#888888";
}

Rectangle.prototype = new GOverlay();
/* สร้าง div element เพื่อใช้เป็นตัวแทนรูปสี่เหลี่ยมผืนผ้า */
Rectangle.prototype.initialize = function(map) {
  var div = document.createElement("div");
  div.style.border = this.weight_ + "px solid " + this.color_;
  div.style.position = "absolute";
  /* รูปสี่เหลี่ยมผืนผ้าของเราจะวางนอนที่ระดับเดียวกับแผนที่ ดังนั้น จึงต้องจัดให้อยู่ที่
  MAP_PANE pane ซึ่งมีค่า z-index เท่ากับของ map แต่จะอยู่ต่ำกว่า marker shadows */
```

```

map.getPane(G_MAP_MAP_PANE).appendChild(div);
this.map_ = map;
this.div_ = div;
}
/* ลบล้าง div ตัวหลักออกไปจาก map pane */
Rectangle.prototype.remove = function() {
this.div_._parentNode.removeChild(this.div_);
}
/* ทำการลอกข้อมูลเพื่อใช้สร้างรูปสี่เหลี่ยมผืนผ้าขึ้นใหม่ */
Rectangle.prototype.copy = function() {
return new Rectangle(this.bounds_, this.weight_, this.color_,
this.backgroundColor_, this.opacity_);
}
/* สร้างรูปสี่เหลี่ยมผืนผ้าขึ้นใหม่ (Redraw) ให้สอดคล้องกับการฉายแผนที่และระดับการย่อขยาย
ขณะนั้น */
Rectangle.prototype.redraw = function(force) {
if (!force) return;
/* คำนวณค่าพิกัดของจุดมุมขอบเขตสองจุดตรงข้ามกันของ div เพื่อนำไปกำหนดขนาดและตำแหน่ง
ของรูปสี่เหลี่ยมผืนผ้าที่จะสร้างขึ้น */
var c1 = this.map_.fromLatLngToDivPixel(this.bounds_.getSouthWest());
var c2 = this.map_.fromLatLngToDivPixel(this.bounds_.getNorthEast());
/* จัดขนาดและตำแหน่งของ div ให้เหมาะสม โดยเอาค่าพิกัดขอบเขตของมันเป็นหลัก */
this.div_.style.width = Math.abs(c2.x - c1.x) + "px";
this.div_.style.height = Math.abs(c2.y - c1.y) + "px";
this.div_.style.left = (Math.min(c2.x, c1.x) - this.weight_) + "px";
this.div_.style.top = (Math.min(c2.y, c1.y) - this.weight_) + "px";
}
var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.4419, -122.1419), 13);
/* สร้างรูปสี่เหลี่ยมผืนผ้าให้อยู่บริเวณกลางแผนที่ โดยให้มีขนาดประมาณ 1/4 ของแผนที่ */

```

```

var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngDelta = (northEast.lng() - southWest.lng()) / 4;
var latDelta = (northEast.lat() - southWest.lat()) / 4;
var rectBounds = new GLatLngBounds(
new GLatLng(southWest.lat() + latDelta, southWest.lng() + lngDelta),
new GLatLng(northEast.lat() - latDelta, northEast.lng() - lngDelta));
map.addOverlay(new Rectangle(rectBounds));

```

3.3.13 การแสดงข้อมูลแผนที่จากแฟ้มข้อมูล KML/ GeoRSS

Google Maps API สนับสนุนการใช้ข้อมูลนำเข้าที่อยู่ในฟอร์แมต KML และ GeoRSS เพื่อแสดงผลเป็นกราฟิกบนแผนที่ ข้อมูลภายใน KML หรือ GeoRSS มีทั้งที่เป็นรหัสสำหรับสร้างองค์แผนที่ประเภทจุด เส้น และพื้นที่ ดังนั้นเมื่อนำมาแสดงผลบนแผนที่ที่เกิดขึ้น จะได้ว่าวัตถุที่เป็น marker, polyline, และ polygon ในทางเทคนิค วัตถุที่ถูกสร้างขึ้นบนแผนที่นี้จะเป็น GGeoXml object โดยมีเมธอด addOverlay() เป็นกระบวนการทำหน้าที่สร้าง และเราสามารถลบวัตถุประเภทนี้ทิ้งได้โดยใช้คำสั่งremoveOverlay()

ตัวอย่าง 15 การใช้ข้อมูลรูปแบบ KML และ GeoRSS

```

// ตัวอย่างการใช้ข้อมูลรูปแบบ KML และ GeoRSS
// GGeoXml ใช้ข้อความระบุ URL ของแฟ้มข้อมูล KML หรือ GeoRSS เพื่อสร้าง GGeoXml object
บนแผนที่
// ในการประมวลผล Maps API สามารถแยกแยะได้เองว่าเป็นไฟล์ KML หรือ GeoRSS
var map;
var geoXml = new GGeoXml("http://mapgadgets.googlepages.com/cta.kml");

function onLoad() {
if (GBrowserIsCompatible()) {
map = new GMap2(document.getElementById("map"));
map.addControl(new GLargeMapControl());

```

```

map.setCenter(new GLatLng(41.875696,-87.624207), 11);
map.addControl(new GLargeMapControl());
map.addOverlay(geoXml);
}
}

```

3.3.14 ภาพวางทับบนแผนที่ (GroundOverlay)

GGroundOverlay (มีตั้งแต่ GMap version 2.79) คือวัตถุบนแผนที่ Google Maps สำหรับรูปภาพสี่เหลี่ยมผืนผ้า (เช่น แผนที่ภาพถ่ายทางอากาศ) ที่นำมาวางทับบนแผนที่ การเขียนรหัสกำหนดรูปภาพมาวางบนแผนที่จะต้องระบุ URL และขอบเขตของภาพโดยใช้ Constructor ดังนี้ GGroundOverlay (imageUrl, bounds) ในที่นี้ bounds เป็นรหัสกำหนดขอบเขตของรูปภาพ ซึ่งต้องใช้ Constructor ดังนี้

```
GLatLngBounds(SW_latlong, NE_latlong)
```

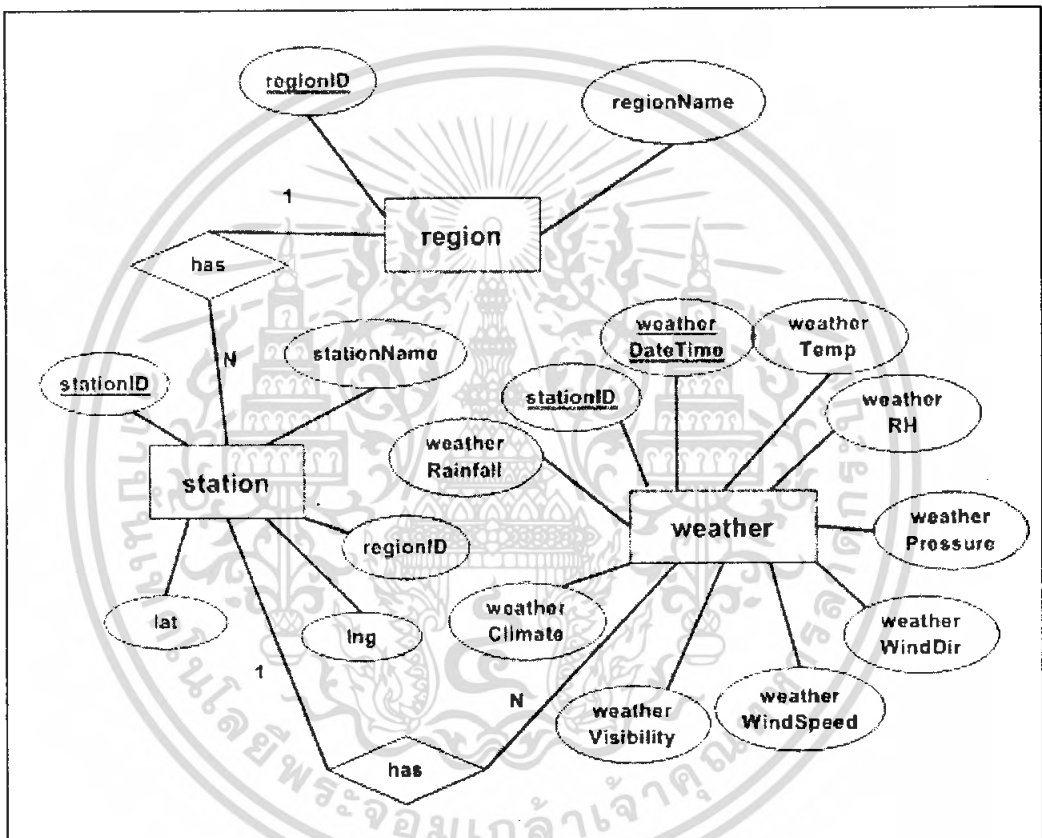
นอกจากนั้น ยังสามารถใช้ ไฟล์ KML กำหนดรายละเอียดของ Ground Overlay ได้อีกด้วย

บทที่ 4

การออกแบบและพัฒนา

4.1 การออกแบบฐานข้อมูล

ระบบฐานข้อมูลที่ใช้เก็บรายละเอียดข้อมูลในโครงการนี้คือ MySQL โดยหลังจากที่ได้สำรวจข้อมูลที่เกี่ยวข้องทั้งหมด สามารถนำมาเขียนเป็น ER Diagram ได้ดังนี้



รูปที่ 4.1 ER Diagram ของระบบ

หลังจากนั้นจึงได้ทำการนอร์มัลไลเซชันได้เป็นตารางทั้งหมด 3 ตารางดังนี้

1. ตาราง Region

ตาราง Region จะเก็บค่าภาคของแต่ละสถานีรวมทั้งรหัสของแต่ละภาคเอาไว้ โดยใช้ในการอ้างอิงเมื่อผู้ใช้งานต้องการค้นหาสถานีตรวจอากาศจำแนกตามภาค

เขตข้อมูล (Field)	คำอธิบาย
regionID	ใช้เพื่อเก็บรหัสของภาค
regionName	ใช้เพื่อเก็บชื่อภาค

ตารางที่ 4.1 เขตข้อมูลและคำอธิบายของตาราง Region

2. ตาราง Station

ตาราง Station จะเก็บค่าตำแหน่งที่ตั้งสถานีเป็นค่าละติจูด/ ลองจิจูด โดยอ้างอิงจากข้อมูลที่ไปติดต่อขอรับมาจากกรมอุตุนิยมวิทยา

เขตข้อมูล (Field)	คำอธิบาย
stationId	ใช้เพื่อเก็บรหัสของสถานี
stationName	ใช้เพื่อเก็บชื่อสถานี
lat	เก็บค่าตำแหน่งที่ตั้งของสถานีเป็นค่า Latitude
lng	เก็บค่าตำแหน่งที่ตั้งของสถานีเป็นค่า Longitude
regionID	ใช้เพื่อเก็บรหัสของภาค

ตารางที่ 4.2 เขตข้อมูลและคำอธิบายของตาราง Station

3. ตาราง Weather

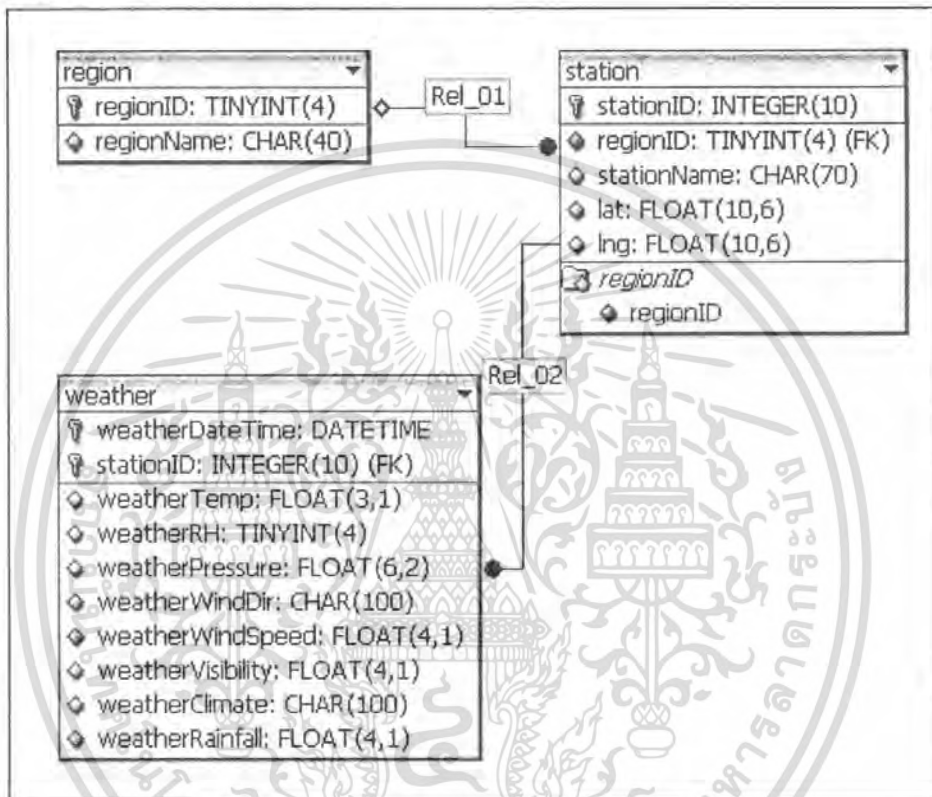
เขตข้อมูล (Field)	คำอธิบาย
weatherDateTime	ใช้เพื่อเก็บวัน-เวลา ที่ทำการเก็บข้อมูล
StationID	ใช้เพื่อเก็บรหัสของสถานี
weatherTemp	ใช้เพื่อเก็บค่าอุณหภูมิ
weatherRH	ใช้เพื่อเก็บค่าความชื้นสัมพัทธ์
weatherPressure	ใช้เพื่อเก็บค่าความกดอากาศ
weatherWindDir	ใช้เพื่อเก็บค่าทิศทางลม
weatherSpeed	ใช้เพื่อเก็บค่าความเร็วลม
weatherVisibility	ใช้เพื่อเก็บค่าทัศนวิสัย
weatherClimate	ใช้เพื่อเก็บค่าลักษณะอากาศ
weatherRainfall	ใช้เพื่อเก็บค่าปริมาณน้ำฝนสะสม

ตารางที่ 4.3 เขตข้อมูลและคำอธิบายของตาราง Weather

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง Weather เป็นตารางหลักของระบบที่มีข้อมูลเปลี่ยนแปลงอยู่ตลอดเวลาโดยตารางนี้จะเก็บรายละเอียดข้อมูลที่ดึงมาจากเว็บไซต์ของกรมอุตุนิยมวิทยา ประกอบด้วยค่าที่เกี่ยวข้องกับสภาวะอากาศต่างๆ และวันที่ทำการเก็บข้อมูล

สามารถเขียนความสัมพันธ์ของตารางทั้ง 3 ได้ดังนี้



รูปที่ 4.2 ความสัมพันธ์ของตารางทั้ง 3 ในฐานข้อมูลระบบ

4.2 ภาพรวมของระบบ

ระบบ Weather Station On Google Maps ประกอบด้วยโปรแกรม 3 ส่วนคือ

1. โปรแกรมส่วนที่ทำหน้าที่ดึงข้อมูล

เป็นโปรแกรมส่วนที่ทำหน้าที่ดึงข้อมูลสภาพอากาศของแต่ละสถานี จากเว็บไซต์ของกรมอุตุนิยมวิทยา มาเก็บในฐานข้อมูลของระบบ ตามช่วงเวลาที่เรากำหนด ในที่นี้คือทุก 3 ชั่วโมงตามการปรับปรุงข้อมูลจากเว็บไซต์ (<http://www.tmd.go.th/xml/index.html>)

2. โปรแกรมส่วนที่ทำหน้าที่จัดการฐานข้อมูล

โปรแกรมส่วนจัดการฐานข้อมูลของระบบ ในส่วนนี้จะทำหน้าที่จัดการว่าจะทำการเก็บข้อมูลไว้เป็นระยะเวลาเท่าใด เช่น กำหนดว่าต้องการเก็บข้อมูลย้อนหลัง 3 เดือนล่าสุด ถ้า

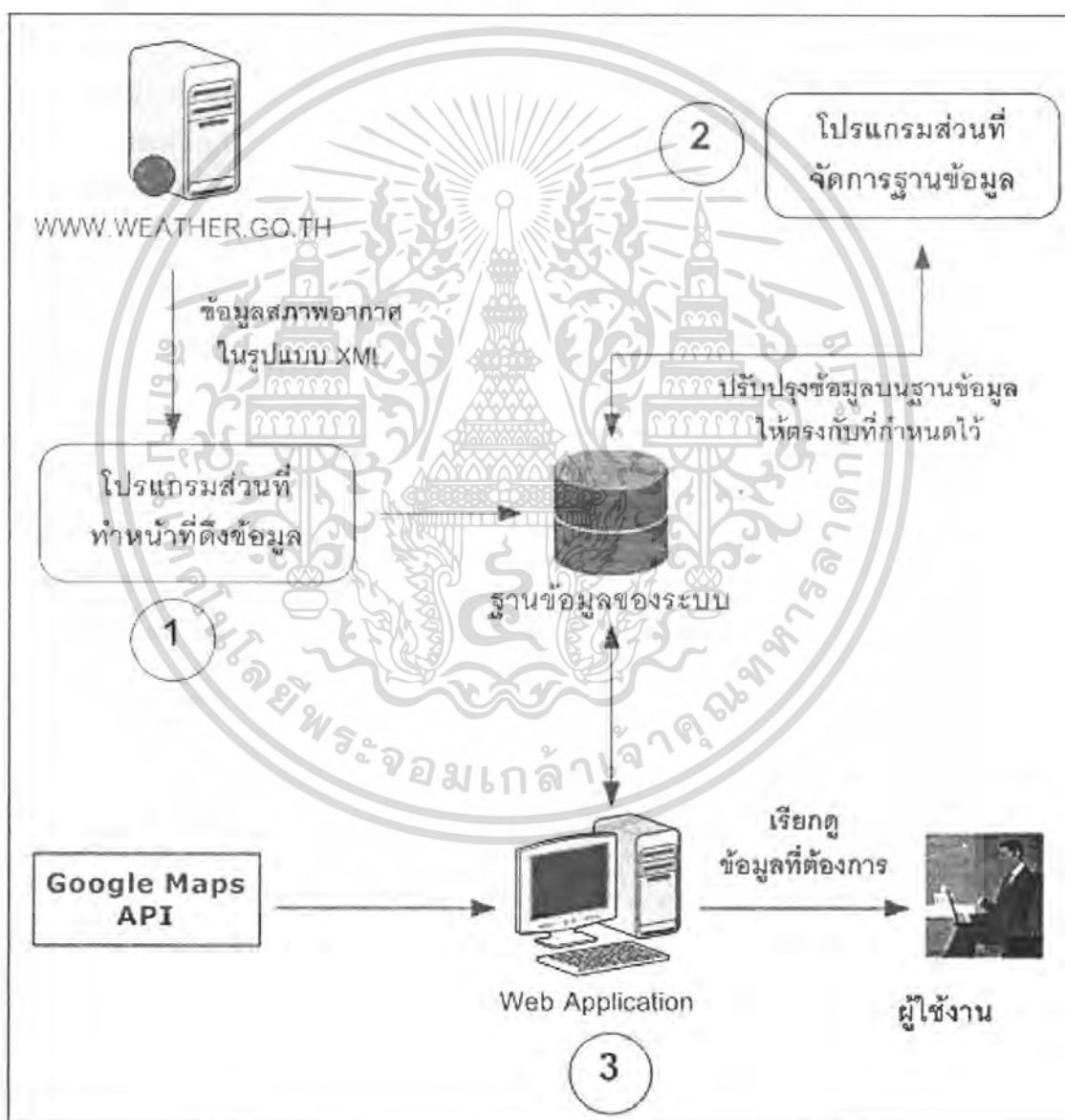
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำไปใช้ประโยชน์อื่นใดได้
 นอกจากนี้จากนั้นก็ทำการลบข้อมูลออกจากฐานข้อมูล เนื่องจากระบบมีการดึงข้อมูลสภาพ
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อากาศของสถานีต่างๆ ทั่วประเทศไทยจำนวน 109 สถานี ทุก 3 ชั่วโมง ดังนั้น ฐานข้อมูลจะมีขนาดใหญ่มากหากเก็บข้อมูลทั้งหมดเอาไว้

3. เว็บแอปพลิเคชัน

โปรแกรมส่วนของเว็บแอปพลิเคชันเป็นส่วน Front End ที่ติดต่อกับผู้ใช้งาน โดยในส่วนนี้จะแสดงข้อมูลสภาพอากาศที่ผู้ใช้งานสนใจผ่านทางแผนที่กูเกิลแมพ

สามารถเขียนเขียนความสัมพันธ์ของระบบ จากโปรแกรมทั้ง 3 ส่วน ได้ดังนี้



รูปที่ 4.3 ภาพรวมของระบบ Weather Station On Google Maps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 โปรแกรมส่วนที่ทำหน้าที่ดึงข้อมูล

ทำหน้าที่ในการดึงข้อมูลรายงานสภาวะอากาศ จากเว็บไซต์ของกรมอุตุนิยมวิทยามาเก็บลงในฐานข้อมูลของระบบ โดยข้อมูลที่ดึงมานั้นจะเป็นข้อมูลที่อยู่ในรูปแบบเอกสาร XML โดยข้อมูลจะมีการอัปเดตทุก 3 ชั่วโมง คือเวลา 01.00 น. , 04.00 น. , 07.00 น. ,10.00 น. , 13.00 น. , 16.00 น. ,19.00 น. , 22.00 น. ของทุกวัน โปรแกรมในส่วนนี้เขียนด้วยภาษา PHP มีขั้นตอนการทำงานดังนี้

4.3.1 ส่วนดึงข้อมูล

จะใช้ PHP ร่วมกับ cURL Functions ในการทำงาน โดยข้อมูลสภาพอากาศของแต่ละสถานีที่เราจะดึงมานั้นจะมีรูปแบบเป็น XML ดังตัวอย่าง

```

Source of: http://www.tmd.go.th/xml/weather_report.php?StationNumber=...
File Edit View Help
<?xml version="1.0" encoding="TIS-620"?><rss version="2.0"
  xmlns:content="http://purl.org/rss/1.0/modules/content/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<channel>
  <title>รายงานสภาวะอากาศ - ชัยภูมิ : อ.เมือง จ.ชัยภูมิ</title>
  <link>http://www.tmd.go.th</link>
  <description>รายงานสภาวะอากาศ โดย กรมอุตุนิยมวิทยา
  กระทรวงเทคโนโลยีสารสนเทศ และการสื่อสาร</description>
  <language>th-TH</language>
  <copyright>Copyright 2006,Thai Meteorological Deaprtment</copyright>
  <ttl>20</ttl>
  <item>
    <title>รายงานสภาวะอากาศ - ชัยภูมิ : อ.เมือง จ.ชัยภูมิ วันที่ : 21/09/2007 เวลา
    16:00 นาฬิกา </title>
    <pubDate>2007-09-21T16:00:00+07:00</pubDate>
    <guid>48403</guid>
    <link>http://www.tmd.go.th/province.php?StationNumber=48403</link>
    <description><![CDATA[
      <b>อุณหภูมิต่ำ : </b>31.1 องศาเซลเซียส <br />
      <b>ความชื้นสัมพัทธ์ : </b>47 % <br />
      <b>ความกดอากาศ : </b>1007.08 มิลลิบาร์ <br />
      <b>ทิศทางลม : </b>ทิศเหนือ <b>ความเร็ว </b>7.4
      กม./ชม.<br />
      <b>ทัศนวิสัย : </b>12.0 กิโลเมตร <br />
      <b>ลักษณะอากาศ : </b>มีเมฆบางส่วน <br />
      <b>ฝนสะสมวันนี้ : </b>0.0 มิลลิเมตร
    ]]></description>
  </item>
</channel>
</rss>
Line 27, Col 7
  
```

รูปที่ 4.4 ข้อมูล XML ที่ให้บริการโดยกรมอุตุนิยมวิทยา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ตามกฎหมายของงานเพื่อการศึกษา และผู้สงวนลิขสิทธิ์ขอสงวนสิทธิ์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่รันคำสั่งในการดึงไฟล์ XML ด้วย cURL เราจะได้ข้อมูลของทุกสถานีไว้ในเท็กซ์ไฟล์ ในที่นี้ตั้งชื่อว่า data.txt ซึ่งเป็นเท็กซ์ไฟล์ที่รวมข้อมูลของทุกสถานีเอาไว้ทั้งหมด 109 สถานี (จำนวนสถานีตรวจอากาศทั้งหมด)

```

1 <?xml version="1.0" encoding="TIS-620"?><rss version="2.0"
2   xmlns:content="http://purl.org/rss/1.0/modules/content/"
3   xmlns:dc="http://purl.org/dc/elements/1.1/"
4 <channel>
5 <title>รายงานสภาวะอากาศ - แม่ฮ่องสอน : อ.เมือง จ.แม่ฮ่องสอน</title>
6 <link>http://www.tmd.go.th</link>
7 <description>รายงานสภาวะอากาศ โดย กรมอุตุนิยมวิทยา กระทรวงเทคโนโลยีสารสนเทศ
8   และการสื่อสาร</description>
9 <language>th-TH</language>
10 <copyright>Copyright 2006,Thai Meteorological Deaprtment</copyright>
11 <ttl>20</ttl>
12 <item>
13 <title>รายงานสภาวะอากาศ - แม่ฮ่องสอน : อ.เมือง จ.แม่ฮ่องสอน วันที่ : 23/09/2007
14   เวลา 07:00 นาฬิกา </title>
15 <pubDate>2007-09-23T07:00:00+07:00</pubDate>
16 <guid>48300</guid>
17 <link>http://www.tmd.go.th/province.php?StationNumber=48300</link>
18 <description><![CDATA[
19 <b>อุณหภูมิ : </b>22.9 องศาเซลเซียส <br />
20 <b>ความชื้นสัมพัทธ์ : </b>89 % <br />
21 <b>ความกดอากาศ : </b>1009.83 มิลลิบาร์ <br />
22 <b>ทิศทางลม : </b> ทิศตะวันตกก่อนไปทางใต้ <b>ความเร็ว </b>5.6 กม./ชม.<br />
23 <b>ทัศนวิสัย : </b>3.0 กิโลเมตร <br />
24 <b>ลักษณะอากาศ : </b> มีหมอกน้ำค้าง <br />
25 <b>ฝนสะสมวันนี้ : </b>0.0 มิลลิเมตร
26 ]]></description>
27 </item>
28 </channel>
29 </rss>
30 <?xml version="1.0" encoding="TIS-620"?><rss version="2.0"
31   xmlns:content="http://purl.org/rss/1.0/modules/content/"
32   xmlns:dc="http://purl.org/dc/elements/1.1/"
33 <channel>
34 <title>รายงานสภาวะอากาศ - ก้านพงเพชร : อ.เมือง จ.ก้านพงเพชร</title>
35 <link>http://www.tmd.go.th</link>
36 <description>รายงานสภาวะอากาศ โดย กรมอุตุนิยมวิทยา กระทรวงเทคโนโลยีสารสนเทศ
37   และการสื่อสาร</description>
38 <language>th-TH</language>
39 <copyright>Copyright 2006,Thai Meteorological Deaprtment</copyright>
40 <ttl>20</ttl>
41 <item>
42 <title>รายงานสภาวะอากาศ - ก้านพงเพชร : อ.เมือง จ.ก้านพงเพชร วันที่ : 23/09/2007
43   เวลา 07:00 นาฬิกา </title>
44 <pubDate>2007-09-23T07:00:00+07:00</pubDate>
45 <guid>48380</guid>
46 <link>http://www.tmd.go.th/province.php?StationNumber=48380</link>
47 <description><![CDATA[
48 <b>อุณหภูมิ : </b>25.3 องศาเซลเซียส <br />
49 <b>ความชื้นสัมพัทธ์ : </b>92 % <br />
50 <b>ความกดอากาศ : </b>1008.66 มิลลิบาร์ <br />
51 <b>ทิศทางลม : </b> ทิศตะวันตกก่อนไปทางเหนือ <b>ความเร็ว </b>3.7 กม./ชม.<br />
52 <b>ทัศนวิสัย : </b>4.0 กิโลเมตร <br />
53 <b>ลักษณะอากาศ : </b> มีหมอกน้ำค้าง <br />
54 <b>ฝนสะสมวันนี้ : </b>0.0 มิลลิเมตร
55 ]]></description>

```

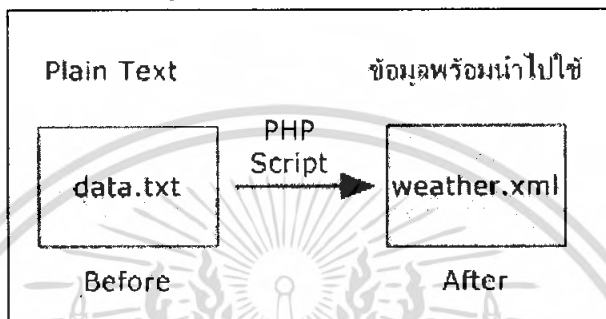
รูปที่ 4.5 ไฟล์ data.txt ที่ได้จากการดึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ส่วนจัดเก็บข้อมูลลงฐานข้อมูล ส่วนนี้มีหน้าที่สำคัญ 2 อย่างคือ

1. จัดการไฟล์ data.txt

ไฟล์ data.txt ที่ได้เป็นเท็กซ์ไฟล์ที่มีข้อมูลของทุกสถานีเก็บไว้ในไฟล์เดียวกัน แต่ยังไม่ใช้รูปแบบ XML ที่จะสามารถเรียกข้อมูลส่วนที่ต้องการออกมาใช้งานได้ ดังนั้นเราจำเป็นต้องนำข้อมูล data.txt มาแก้ไข คือเพิ่มเติมหรือตัดข้อมูลบางส่วนออกเพื่อให้ได้ไฟล์ XML ที่มีลักษณะ Well Formed ซึ่งวิธีการจัดการในที่นี้จะใช้สคริปต์ PHP เพื่ออ่านไฟล์ data.txt เข้ามาแล้วใช้ฟังก์ชันในการจัดการสตริงแก้ไขข้อมูลเหล่านั้นซึ่งมีขั้นตอนการทำงานดังภาพ



รูปที่ 4.6 กระบวนการสร้างไฟล์ weather.xml

ไฟล์ weather.xml ที่ได้มีลักษณะดังนี้

```

2 <?xml version="1.0" encoding="UTF-8"?>
3
4 <stations>
5 <station>
6 <guid>48300</guid>
7 <pubDate>2007-09-25T01:00:00+07:00</pubDate>
8 <temp>24.7 </temp>
9 <RH> 92 </RH>
10 <pressure> 1007.11 </pressure>
11 <windDir> ลมสงบ </windDir>
12 <windSpeed> </windSpeed>
13 <visibility> 3.0 </visibility>
14 <climate> มีหมอกน้ำค้าง </climate>
15 <rainFall> 0.0 </rainFall>
16 </station>
17
18 <station>
19 <guid>48380</guid>
20 <pubDate>2007-09-25T01:00:00+07:00</pubDate>
21 <temp>26.2 </temp>
22 <RH> 93 </RH>
23 <pressure> 1005.14 </pressure>
24 <windDir> ลมสงบ </windDir>
25 <windSpeed> </windSpeed>
26 <visibility> 6.0 </visibility>
27 <climate> มีหมอกน้ำค้าง </climate>
28 <rainFall> 0.0 </rainFall>
29 </station>

```

รูปที่ 4.7 ไฟล์ weather.xml ที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนอ่านข้อมูลลงไปเก็บในฐานข้อมูล

เมื่อได้ไฟล์ weather.xml แล้วขั้นต่อมาคือการดึงข้อมูลของแต่ละสถานีมาเก็บลงในฐานข้อมูล โดยกระบวนการนำข้อมูล XML มาใช้งานในแอปพลิเคชันนั้นจะมี XML Parser เป็นตัวกลางในการดึงข้อมูลจากเอกสาร XML และแอปพลิเคชันซึ่งเป็น API ชนิดหนึ่ง โดย API ที่นิยมใช้กันมากคือ DOM SAX และ SimpleXML ซึ่งต่างก็มีวิธีการดึงข้อมูลที่แตกต่างกันคือ

DOM จะมองเอกสาร XML ในลักษณะของโครงสร้างต้นไม้ (Tree) DOM ย่อมาจาก Document Object Model โดยมีหลักการในการอ่านเอกสาร XML มาวางเป็น Tree ในหน่วยความจำของเครื่องที่กำลังทำงาน ประกอบด้วยอิลิเมนต์ หรือเอททริบิวต์ต่างๆ การเข้าถึงข้อมูลจึงเป็นการเดินไปตามกิ่งก้านต่างๆ ทั้งเป็นแบบต่อเนื่องไปเรื่อยๆ หรือจะอ้างอิงกิ่งก้านเฉพาะเจาะจงลงไป หรือเข้าถึงแบบสุ่มก็ได้ ข้อจำกัดของ DOM อยู่ตรงปริมาณหน่วยความจำของเครื่องว่าจะสามารถรองรับข้อมูลได้ใหญ่แค่ไหน เพราะวิธี DOM จะทำการอ่านข้อมูลทั้งหมดมาเก็บไว้ในหน่วยความจำเพียงครั้งเดียว แต่ข้อดีคือสามารถจัดการ โปรแกรมได้ง่ายกว่ามาก

SAX จัดการเอกสาร XML ด้วยแนวทาง Event-Driven ย่อมาจาก Simple API for XML โดย SAX จะไม่โหลดข้อมูลทั้งหมดเข้ามาในหน่วยความจำ แต่จะอ่านเอกสารจากดิสก์ตั้งแต่เริ่มต้นไล่ไปเรื่อยๆ แล้วจะมีการสร้างอีเวนต์ออกมาเช่น เปิดอิลิเมนต์ ปิดอิลิเมนต์ หรือเจอเอททริบิวต์ เป็นต้น คนเขียนโปรแกรมก็ต้องทำหน้าที่ดักจับอีเวนต์เหล่านี้มาจัดการกับข้อมูลต่อ เช่นเจออีเวนต์ เมื่อถึงอิลิเมนต์ชื่อ X ก็ต้องทำหน้าที่ในการดึง X มาใช้งานเป็นต้น โดย SAX จะเหมาะสำหรับการทำงานข้อมูลเล็กน้อยบางอย่าง ซึ่งหากใช้ DOM ก็ต้องโหลดข้อมูลปริมาณมากทั้งหมดภายในครั้งเดียวซึ่งจะเป็นการสิ้นเปลืองหน่วยความจำเครื่องอย่างมาก

SimpleXML เป็นฟังก์ชันเสริมที่ถูกเพิ่มเข้ามาในโปรแกรมภาษา PHP ตั้งแต่เวอร์ชัน 5 เป็นต้นมา เพื่อรองรับการทำงานร่วมกับ XML ให้สมบูรณ์ขึ้น โดย SimpleXML จะใช้วิธีการอ่านข้อมูลเข้ามาแล้วเปลี่ยนเป็นออบเจกต์ และปรับให้อิลิเมนต์ภายในเอกสารนั้นเป็นคุณสมบัติของออบเจกต์ ซึ่งสามารถเข้าถึงได้โดยใช้การอ้างอิงแบบออบเจกต์อย่างง่าย รวมทั้งยังสามารถใช้งาน SimpleXML ร่วมกับ DOM ด้วยวิธีการคอนเวิร์ท โหนดกลับไปกลับมาระหว่าง โหนดของ DOM อิลิเมนต์กับ โหนดของ SimpleXML อิลิเมนต์ สิ่งที่ได้คือความง่ายในการสั่งงานไปยังอิลิเมนต์ในหลายระดับของ XML Hierarchy เพื่อเข้าถึงข้อมูล เหมาะสำหรับนำไปใช้กับการทำงานร่วมกับไฟล์ XML ที่มีความซับซ้อนไม่มากนัก หรือมีรูปแบบที่แน่นอนตายตัวอย่างเช่น ระบบของเราเป็นต้น ในที่นี้จะใช้ SimpleXML ร่วมกับ PHP ในการดึงข้อมูลแต่ละแท็กจากเอกสาร XML ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างวิธีเรียกใช้งาน SimpleXML

```

2 <?php
3 include 'example.php';
4
5 $xml = new SimpleXMLElement ($xmlstr);
6
7 echo $xml->movie[0]->plot;
8 ?>

```

รูปที่ 4.8 การเรียกใช้งาน SimpleXML

การเรียกใช้ SimpleXML เมื่อต้องการดึงค่าที่อยู่ในแท็ก plot จากไฟล์ example.php

```

1
2 <?php
3 $xmlstr = <<<XML
4 <xml version='1 0' standalone = 'yes'?>
5 <movies>
6 <movie>
7 <title> PHP: Behind the Parser</title>
8 <characters>
9 <character>
10 <name>Ms. Coder</name>
11 <actor>Olivia Actora</actor>
12 </character>
13 <character>
14 <name>Mr. Coder</name>
15 <actor>El Act</actor>
16 </character>
17 </characters>
18 <plot>
19 So, this language. It's like, a programming language. Or is it a scripting language?
20 </plot>
21 <great-lines>
22 <line>PHP solves all my web problems</line>
23 </great-lines>
24 <rating type="thumb">7</rating>
25 <rating type="stars">5</rating>
26 </movie>
27 </movies>
28 XML:
29 ?>
30

```

รูปที่ 4.9 ไฟล์ example.php

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนตั้งเวลาการทำงาน

การทำงานของโปรแกรมส่วนที่ดึงข้อมูลนั้นจะต้องมีการทำงานทุกๆ 3 ชั่วโมงคือ ทุกครั้งที่ข้อมูลจากเว็บไซต์ของกรมอุตุนิยมวิทยา มีการอัปเดตก็จะมีการเรียกให้ระบบดึงข้อมูลทำงานด้วย ในที่นี่จะใช้วิธีการตั้งเวลาในการรันสคริปต์ โดยใช้โปรแกรม crontab ซึ่งเป็นโปรแกรมในระบบ Unix เข้ามาช่วย โดยเราสามารถตั้งเวลาให้ crontab ทำงานตลอดเวลา 03.00 น. , 06.00 น. , 09.00 น. , 12.00 น. , 15.00 น. , 18.00 น. , 21.00 น. , 24.00 น. ได้ด้วย คำสั่งดังนี้

```
0 3,6,9,12,15,18,21,24 * * * /home/username/grabdata.php >/dev/null 2>&1
```

สตริง “>/dev/null 2>&1” ด้านหลังคำสั่ง crontab มีความหมายเพื่อไม่ต้องการให้ระบบส่งเมลแจ้งเตือนการทำงานของ crontab กลับมาที่เรา ซึ่งการส่งเมลแจ้งเตือนการทำงานนั้นจะทำงานโดยอัตโนมัติเมื่อ crontab ทำงาน

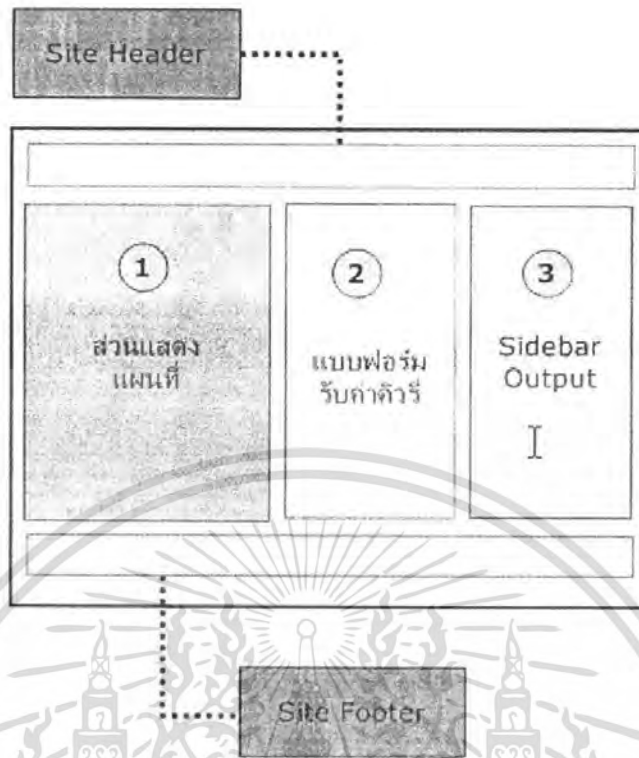
4.4 โปรแกรมส่วนที่ทำหน้าที่จัดการฐานข้อมูล

เขียนด้วยภาษา PHP เหมือนโปรแกรมส่วนที่ทำหน้าที่ดึงข้อมูลและใช้ CronTab ตั้งเวลาทำงานเหมือนกัน แต่จะตั้งค่าให้รันสคริปต์ทุก 24 ชั่วโมง โดยโปรแกรมส่วนนี้จะทำหน้าที่ลบข้อมูลในตาราง weather ในฐานข้อมูลระบบ ส่วนที่ไม่ต้องการออกไป โดยเราจะกำหนดรายละเอียดการเก็บค่าในฐานข้อมูลดังนี้

- ข้อมูลวันปัจจุบัน เก็บข้อมูลทุก 3 ชั่วโมงเอาไว้
- ข้อมูลย้อนหลัง เก็บข้อมูลเอาไว้วันละ 1 ครั้ง (1 record ต่อ 1 สถานี)
- เก็บข้อมูลย้อนหลังเป็นเวลา 3 เดือน

4.5 เว็บแอปพลิเคชัน

ส่วนนี้จะเป็นส่วน Front End ที่ติดต่อกับผู้ใช้งาน มีลักษณะเป็นหน้าเว็บเพจที่เขียนโดยโค้ด HTML , PHP , JavaScript และ AJAX มีการออกแบบ User Interface ดังนี้

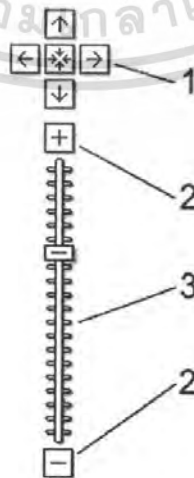


รูปที่ 4.10 ส่วนประกอบของหน้าจอหลัก

รายละเอียดของแต่ละส่วนประกอบด้วย

4.5.1 ส่วนแสดงแผนที่

บนแผนที่ Google Maps มีส่วนต่างๆ ดังนี้



รูปที่ 4.11 ส่วนควบคุมแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ลูกศร ใช้สำหรับเลื่อนแผนที่ไปตามทิศทางที่ต้องการ
2. ชูม คลิก + เพื่อซูมเข้า และคลิก - เพื่อซูมออก
3. ชูมสไลเดอร์ ใช้การซูมแผนที่เข้าออก โดยการเลื่อนจากแถบนี้

4.5.2 แบบฟอร์มในการรับค่าคิวรีจากผู้ใช้งาน

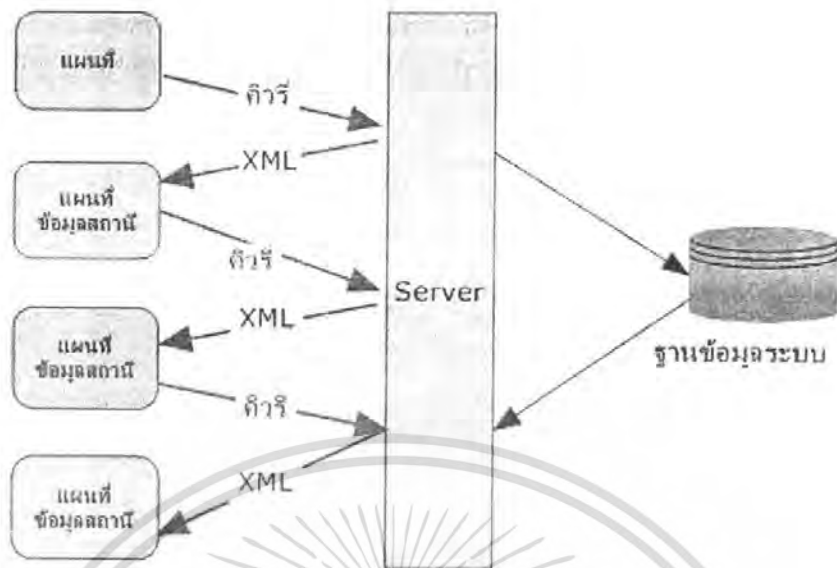
ส่วนนี้จะเป็นการให้ผู้ใช้งานเลือกรายละเอียดที่ต้องการแสดงผลบนแผนที่ โดยจะรับข้อมูลมาจากฟอร์ม HTML หลังจากนั้นจะส่งค่าไปยังสคริปต์ PHP สคริปต์ก็จะทำการเรียกค่าในฐานข้อมูลที่ตรงกับคิวรีขึ้นมา โดยสร้างเป็นเอกสาร XML ส่งกลับมา สาเหตุที่ต้องสร้างเป็นรูปแบบ XML นั้นเนื่องจากแผนที่ Google Maps จะรับค่าข้อมูลที่อยู่ในรูปแบบ XML เท่านั้น โดยการทำงานในส่วนนี้จะมีลักษณะการทำงานแบบ AJAX คือการรับและส่งข้อมูลระหว่างแอปพลิเคชันกับฐานข้อมูลนั้นจะมีการทำงานแบบอะซิงโครนัส กล่าวคือ เป็นการติดต่อสื่อสารแบบไม่ต่อเนื่อง โดยเซิร์ฟเวอร์จะทำการส่งผลลัพธ์ให้ผู้ใช้งานที่โดยไม่ต้องรอให้ประมวลผลเสร็จก่อน หลังจากนั้นเมื่อผู้ใช้ได้รับข้อมูลจะทำการดึงข้อมูลในส่วนต่างๆ ที่หลัง หรือจะดึงข้อมูลก็ต่อเมื่อผู้ใช้ต้องการเท่านั้น ซึ่งการทำงานในลักษณะนี้มีข้อดีคือทำให้การโหลดข้อมูลที่จะแสดงบนแผนที่ทำได้รวดเร็วขึ้น โดยไม่ต้องทำการรีเฟรชหน้าจอใหม่ทุกครั้งที่มีการติดต่อกับฐานข้อมูล นี่เป็นข้อดีอย่างหนึ่งของ Google Maps ที่เหนือกว่าระบบแผนที่แบบเดิม

4.5.3 Sidebar Output

ส่วนนี้จะแสดงผลที่ได้จากการค้นหาข้อมูล โดยแสดงผลเป็นลิสต์รายชื่อสถานที่ที่ตรงกับการค้นหา ผู้ใช้งานสามารถคลิกที่รายชื่อสถานที่ที่ต้องการจากรายชื่อเหล่านี้ เพื่อดูรายละเอียดสภาพอากาศที่สนใจได้โดยตรง

4.5.4 กระบวนการสร้างไฟล์ XML ส่งให้แผนที่

แผนที่ Google Maps นั้นทำงานได้โดยอาศัยกลุ่มของจาวาสคริปต์คลาสเป็นหลัก การที่จะนำข้อมูลของเรามาแสดงบนแผนที่ได้ ข้อมูลที่ส่งให้แผนที่ต้องเป็นไฟล์ XML ซึ่งเราจะทำการสร้างไฟล์ XML ที่เก็บข้อมูลสภาพอากาศที่ตรงกับคำค้นของผู้ใช้งาน จากข้อมูลที่มีอยู่ในฐานข้อมูล MySQL



รูปที่ 4.12 การสร้างเอกสาร XML ให้แผนที่

โดยการสร้างไฟล์ XML ส่งให้แผนที่นั้นมึกระบวนการสร้างดังนี้

1. ผู้ใช้งานเลือกค่าตัวเลือกที่ต้องการ
2. ส่งค่าตัวแปรเหล่านั้นไปยังคิวรีสตริง เพื่อส่งข้อมูลไปยังสคริปต์ PHP ที่ทำหน้าที่ประมวลผลที่เซิร์ฟเวอร์
3. สคริปต์นำค่าในฐานข้อมูลที่ตรงกับคิวรีสตริงมาสร้างเอกสาร XML
4. ส่งเอกสาร XML ให้แผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงรายละเอียดของระบบ และผลที่ได้จากการทดสอบการใช้งานดังนี้

5.1 อินเทอร์เฟซของส่วนของเว็บแอปพลิเคชัน

หน้าจอหลักประกอบด้วย 3 ส่วนคือ แผนที่ ส่วนค้นหาข้อมูล และ ส่วนแสดงผลการค้นหาคำ

ค้นหาคำ



รูปที่ 5.1 หน้าจอหลักของระบบ

5.1.1 ส่วนแผนที่

ส่วนนี้อยู่ด้านซ้ายมือสุดของหน้าจอ ประกอบด้วยตารางขนาดกว้าง 480 พิกเซล สูง 500 พิกเซล คำเริ่มต้นจะแสดงแผนที่ประเทศไทย โดยยังไม่มีรายละเอียดสถานีใดๆ ขึ้นมา

5.1.2 ส่วนค้นหาข้อมูล

ในส่วนนี้ผู้ใช้งานสามารถเลือกค่าตัวเลขต่างๆ ได้ดังนี้

วันที่ เดือน วันที่: 11 เดือน: กันยายน 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

อุณหภูมิ

อุณหภูมิ: - องศาเซลเซียส

ส่งข้อมูล

รูปที่ 5.2 ตัวเลือกในการเรียกดูข้อมูลสภาพอากาศ

ซึ่งในรายละเอียดของแต่ละสถานที่เก็บในฐานข้อมูลนั้นจะมีข้อมูลค่าต่างๆ มากกว่านี้ แต่เราจะเลือกเฉพาะค่าที่สำคัญและผู้ใช้ต้องการทราบที่สุดมาไว้ในส่วนของการค้นหาข้อมูล ส่วนค่าอื่นๆ นั้นสามารถดูได้จากรายละเอียด เมื่อกดคลิกที่มาร์กเกอร์บนแผนที่หรือคลิกที่ส่วนแสดงผลข้อมูลด้านขวามือ

5.1.3 ส่วนแสดงผลข้อมูล ส่วนนี้จะเป็นผลการค้นหาที่ตรงกับคำค้นในส่วนที่ 2 โดยจะแสดงผลลัพธ์ในลักษณะของรายชื่อสถานีเรียงลงมาตามลำดับ

5.2 การทดสอบการใช้งาน

การทดสอบการใช้งานจะเป็นการทดลองป้อนค่าเพื่อทดสอบการแสดงผลที่ได้ ดังนี้

5.2.1 กำหนดค่าที่จะค้นหาจากฐานข้อมูล

ทดลองป้อนค่าดังนี้

รายละเอียดที่ต้องการ

วันที่:

เดือน: 2550

ภาค:

อุณหภูมิ:

ต่ำสุด องศาเซลเซียส

สูงสุด องศาเซลเซียส

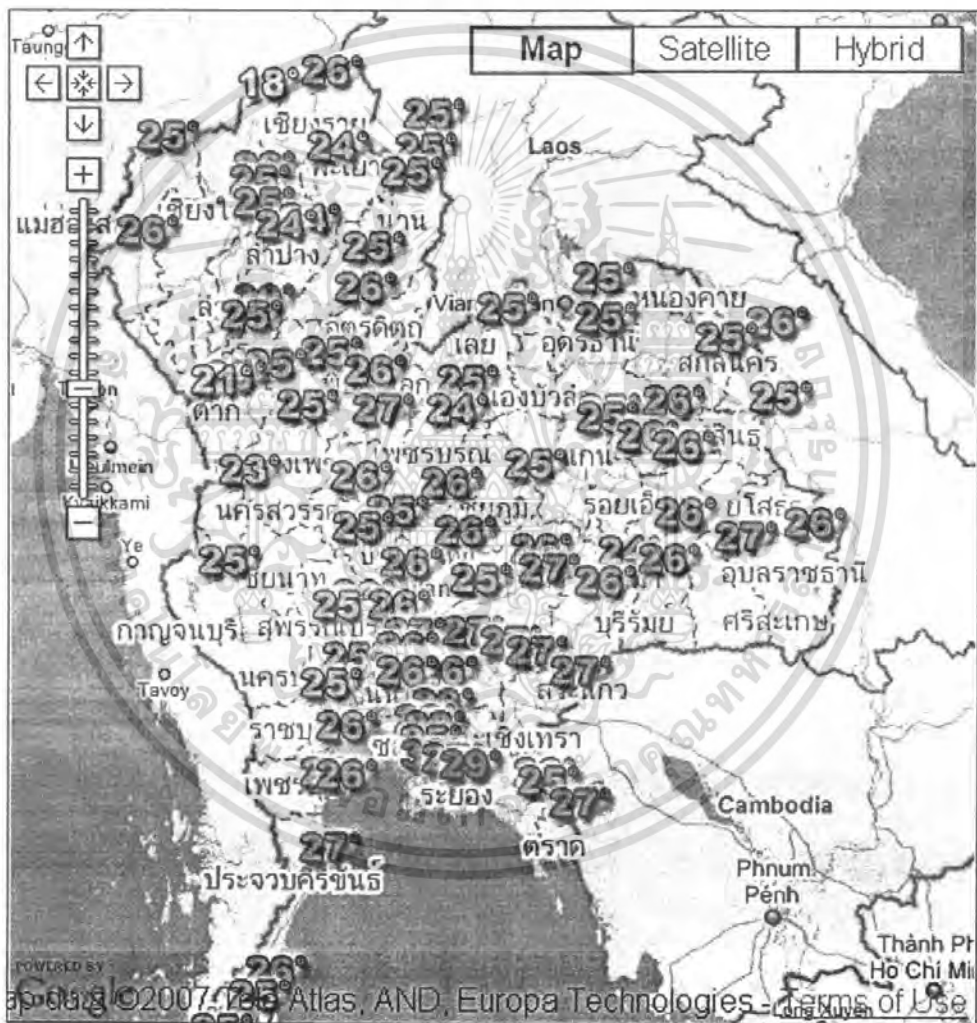
รูปที่ 5.3 ค่าที่ใช้ในการทดลองค้นหาข้อมูลสภาพอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้สำหรับการแจ้งข้อบกพร่องเท่านั้น ไม่สามารถเผยแพร่ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของค่าต่างๆ ที่ป้อนจากตัวอย่างคือ ผู้ใช้งานต้องการให้แผนที่แสดงผลข้อมูลของสถานีตรวจอากาศย้อนหลังไปเมื่อวันที่ 15 เดือนกันยายน พ.ศ. 2550 โดยให้แสดงผลสถานีทั้งหมดทั่วประเทศไทย ที่มีอุณหภูมิตั้งแต่ 16-40 องศาเซลเซียส เราจะทดสอบระบบโดยพิจารณาผลการทำงานดังนี้

5.2.2 ผลการรันโปรแกรม

ผลการทำงานที่แสดงบนแผนที่ เมื่อผู้ใช้สั่ง Update Map แผนที่ จะแสดงมาร์กเกอร์ของสถานีต่างๆ ตามตำแหน่งที่ตั้ง อ้างอิงจากค่าในฐานข้อมูล ในที่นี้เราจะเห็นสถานีทั้งหมดบนแผนที่



รูปที่ 5.4 ผลการค้นหาข้อมูลที่แสดงบนแผนที่

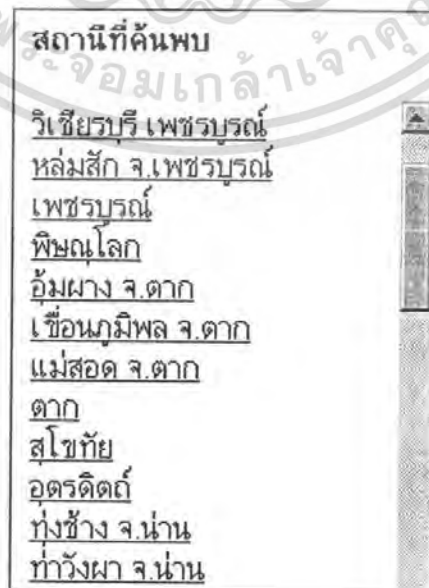
โดยผู้ใช้งานสามารถซูมเข้า-ออก เพื่อเลือกดูรายละเอียดบนแผนที่ให้ชัดเจนขึ้น ได้ด้วยการคลิกที่แถบซูมสไลด์อร์บนฝั่งซ้ายมือของแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 การซูมเข้าเพื่อดูรายละเอียดบนแผนที่

ผลการทำงานที่แสดงบนส่วนแสดงผลข้อมูล ในส่วนนี้จะสร้างลิสต์รายชื่อสถานที่จากคำค้นของผู้ใช้งานที่ตรงกับข้อมูลที่แสดงบนแผนที่



รูปที่ 5.6 ผลการค้นหาค้นหาแสดงรายชื่อสถานที่ที่ค้นพบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ใช้เห็นจำเป็นต้องดำเนินการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

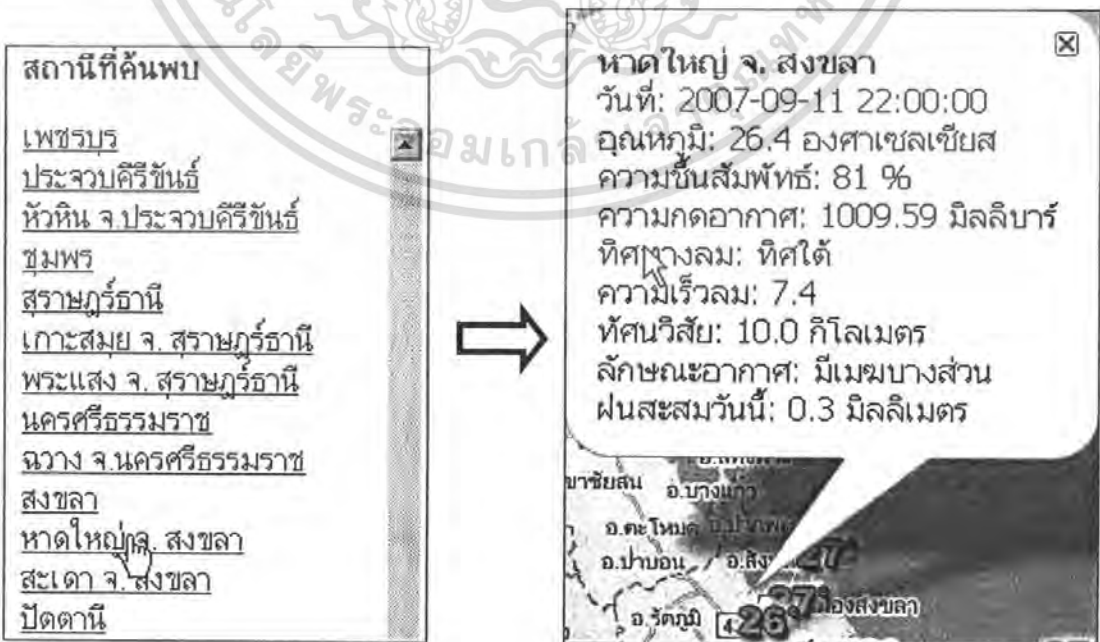
นั่นคือผลลัพธ์ทั้ง 2 ส่วนมีการเชื่อมโยงกันอยู่ ผู้ใช้งานสามารถเลือกคลิกที่อันใดอันหนึ่ง เพื่อดูรายละเอียด จะได้ผลลัพธ์เหมือนกัน

เมื่อคลิกที่มาร์กเกอร์จะมีหน้าต่างข้อมูลแสดงขึ้นมาดังนี้



รูปที่ 5.7 ข้อมูลที่อ่านได้ จากการคลิกมาร์กเกอร์บนแผนที่

ส่วนต่อมาจะแสดงผลที่ได้จากการค้นหาข้อมูล โดยแสดงผลเป็นลิสรายชื่อสถานที่ที่ตรงกับ การค้นหา ผู้ใช้งานสามารถคลิกที่รายชื่อสถานที่ที่ต้องการจากรายชื่อเหล่านี้ เพื่อดูรายละเอียดสภาพ อากาศที่สนใจได้โดยตรง



รูปที่ 5.8 ผลการเลือกสถานที่จากรายชื่อบน Side Bar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลและแนวทางการพัฒนาต่อ

5.1 สรุปผลการดำเนินงาน

หลังจากทดสอบการทำงาน พบว่าระบบทำงานได้ตรงตามที่ตั้งเป้าไว้ โดยสามารถแสดงผลสถานีตรวจอากาศที่ต้องการได้อย่างถูกต้องตามค่าที่ค้นหา รวมทั้งแต่ละสถานี ก็สามารถแสดงรายละเอียดสภาพอากาศได้ครบถ้วน ทำให้ผู้ใช้งานสามารถเรียกดูสภาพอากาศของสถานีที่ต้องการได้ง่ายขึ้น และสามารถอ่านรายงานสภาพอากาศได้อย่างสะดวกรวดเร็ว และเห็นภาพมากกว่าระบบแบบเดิม

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

1. เนื่องจากระบบแผนที่ Google Maps มีการใช้งานจาวาสคริปต์เป็นหลัก ดังนั้นหากผู้ใช้งานเรียกใช้งานเว็บแอปพลิเคชันของระบบเราจากโปรแกรมบราวเซอร์ที่เป็นรุ่นเก่า หรือไม่รองรับจาวาสคริปต์จะทำให้ไม่สามารถแสดงผลการทำงานได้อย่างถูกต้อง ซึ่งอาจจัดการปัญหานี้ด้วยการเขียนรหัสโปรแกรมสำหรับตรวจสอบและแจ้งเตือนผู้ใช้งานให้ทราบสาเหตุที่การแสดงผลไม่ถูกต้อง โดยหากต้องการเขียนรหัสตรวจสอบขีดความสามารถของบราวเซอร์ว่ามีความเข้ากันได้กับ Google Maps API หรือไม่ สามารถใช้ Global method ที่ชื่อ GBrowserIsCompatible() ซึ่งเป็นส่วนหนึ่งใน Google Maps API ให้ทำหน้าที่ตรวจสอบก็ได้ ขณะที่จัดทำโครงการนี้ เว็บบราวเซอร์รุ่นปัจจุบันของ MS Internet Explorer, Mozilla Firefox, Netscape Navigator และ Opera ต่างก็สามารถใช้งาน Google Maps ได้ดี

2. การที่เราดึงข้อมูลจากระบบภายนอก ในที่นี้คือเว็บไซต์กรมอุตุนิยมวิทยาทำให้ระบบต้องพึ่งพาข้อมูลส่วนนี้อยู่ตลอด หากเครื่องเซิร์ฟเวอร์ของกรมอุตุนิยมวิทยาล่ม ทำงานขัดข้อง หรือมีการอัปเดตข้อมูลล่าช้า ก็จะส่งผลกระทบต่อระบบของเราโดยตรงอย่างหลีกเลี่ยงไม่ได้

3. การดึงข้อมูลจากเว็บไซต์กรมอุตุนิยมวิทยานั้น จะต้องดึงข้อมูลที่มีขนาดใหญ่พร้อมๆ กันคือข้อมูลทั้งหมด 109 สถานี ดังนั้นเครื่องเซิร์ฟเวอร์ระบบของเราที่จะติดต่อไปดึงข้อมูลมานั้นจำเป็นต้องเชื่อมต่อด้วยอินเทอร์เน็ตความเร็วสูง การดึงข้อมูลจึงจะเสร็จสมบูรณ์

5.3 แนวทางในการพัฒนาต่อ

อินเทอร์เน็ตที่ติดต่อกับผู้ใช้งานนั้น สามารถพัฒนาเพิ่มในส่วนของการค้นหาข้อมูลที่อาจเพิ่มรายละเอียดอย่างอื่นเข้าไปได้อีก ยกตัวอย่างเช่น ให้ผู้ใช้งานสามารถค้นหาสถานที่ที่ต้องการจากค่าความชื้นสัมพัทธ์ ความกดอากาศ ทิศทางลม หรือทัศนวิสัยในการมองเห็น เป็นต้น รวมทั้งรายละเอียดของแต่ละสถานที่ที่แสดงหลังจากผู้ใช้คลิกบนมาร์กเกอร์อาจเพิ่มระบบการแสดงผลสภาพอากาศแบบแผนภูมิเส้นกราฟ ซึ่งจำเป็นต้องมีการเรียกใช้งานฟังก์ชันของ PHP ในการสร้างกราฟเพิ่มเติม โดยจะทำให้ผู้ใช้งานสามารถอ่านรายงานสภาพอากาศได้ง่ายมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] วิชา ศิริธรรมจักร. 2006. WEB PROGRAMMING ด้วย AJAX และ PHP.
- [2] สวัสดิ์ชัย เกรียงไกรเพชร. การใช้ Google Maps API เพื่อสร้าง Google Map Mashup อย่างง่าย. “<http://lecturer.eng.chula.ac.th/fsvskk/gglmap2/get-start-gmap-apiv2.html>”. 2006
- [3] Andre Lewis , Michael Purvis , Jeffrey Sambells and Cameron Turner. 2006. Beginning Google Maps Applications with PHP and Ajax: From Novice to Professional. USA : Apress.
- [4] Bill Evjen, Kent Sharkey, Thiru Thangarathinam, Michael Kay, Alessandro Vernet, Sam Ferguson. 2004. Professional XML. USA : Wrox.
- [5] Lee Babin. 2006. Beginning Ajax with PHP: From Novice to Professional. USA : Apress.
- [6] Martin C. Brown. 2006. Hacking GoogleMaps and GoogleEarth. USA : ExtremeTech.
- [7] Rael Dornfest , Paul Bausch ,Tara Calishain. 2006. Google Hacks: Tips & Tools for Finding and Using the World's Information. USA : O'Reilly Media, Inc.
- [8] Rich Gibson and Schuyler Erle. 2006. Google Maps Hacks. USA : O'Reilly Media, Inc.
- [9] Sas Jacobs. 2006. Beginning XML with DOM and Ajax: From Novice to Professional. USA : Apress.