

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการจัดการการเข้าถึงเอกสารเอกซเอมแอล

XML ACCESS CONTROL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

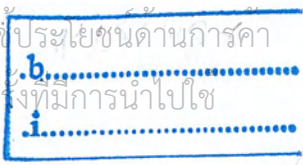
ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้ **ได้ขอยืม** สสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ **โดยทะเบียน 55097** หากมีการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
วัน,เดือน,ปี - 8 เม.ย. 2548



ระบบการจัดการการเข้าถึงเอกสารเอกซเอมแอล

XML ACCESS CONTROL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตร ปีการศึกษา 2546


ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการจัดการการเข้าถึงเอกสารเอกซเอมแอล  
XML ACCESS CONTROL

คณะผู้จัดทำ นายอนุชิน ฉัตรชินรัตน์ รหัส 43010515  
นายอนุรักษ์ อภิสมภาร รหัส 43010518



.....อาจารย์ที่ปรึกษา  
(อ. ชนา หงษ์สุวรรณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบการจัดการการเข้าถึงเอกสารเอกซเอ็มแอล

นายอนุชิน ฉัตรชินรัตน์ 43010515

นายอนุรักษ อภิสมจาร 43010518

อ.ธนา หงษ์สุวรรณ

ปีการศึกษา 2546

### บทคัดย่อ

เอกซเอ็มแอลได้กลายเป็นมาตรฐานการส่งข้อมูลบนอินเทอร์เน็ต, การส่งข้อมูลภายในเครือข่ายเดียวกัน (LAN) ทั้งยังเป็นมาตรฐานในการส่งข้อมูลของเว็บเซอร์วิส (Web Service) ที่เรียกกันว่า SOAP หรือแม้แต่ การทำ ธุรกิจบนอินเทอร์เน็ต (E-Commerce) ซึ่งข้อมูลที่น่าสนใจก็เป็นเอกสารเอกซเอ็มแอล (XML Document) ดังนั้นจึงเห็นได้ว่าเทคโนโลยีของเอกซเอ็มแอลนั้นน่าสนใจอย่างมาก

ในเมื่อเอกสารเอกซเอ็มแอลถูกนำมาใช้บนธุรกิจบนอินเทอร์เน็ต ซึ่งเป็นที่รู้กันว่าข้อมูลเป็นสิ่งสำคัญมากดังนั้น จึงมีการคิดค้นการรักษาความปลอดภัยต่างๆขึ้นมากมาย เช่น การเข้ารหัส (Encryption), การพิสูจน์ตัวตน (Authentication) เป็นต้น ดังนั้น ระบบการจัดการการเข้าถึงเอกสารเอกซเอ็มแอล (Xml Access Control) ที่คิดค้นขึ้นจะมีจุดประสงค์เพื่อความปลอดภัยเช่นเดียวกัน ซึ่งมีหน้าที่หลักสองประการด้วยกันคือ การพิสูจน์ตัวตนและการพิสูจน์สิทธิ์ (Authorization) ในการเข้าถึงเอกสารเอกซเอ็มแอล เพื่อสร้างความปลอดภัยในการนำเสนอข้อมูลที่แตกต่างกันโดยมีการกำหนดกฎการเข้าถึง (Access Rule) เป็นตัวระบุสิทธิ์ในการทำงานต่างๆ นอกจากนี้ยังประยุกต์การแปลงข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ (Relational Database) มาเป็นเอกสารเอกซเอ็มแอล ก่อนที่จะมีการตรวจสอบการเข้าถึง เพราะในความเป็นจริงการเก็บข้อมูลส่วนใหญ่จะเก็บอยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์

ดังนั้น โครงการนี้จึงนำแนวคิดระบบการจัดการการเข้าถึงเอกสารเอกซเอ็มแอล ไปประยุกต์ใช้กับการทำธุรกิจบนอินเทอร์เน็ต เพื่อให้เห็นถึงประโยชน์บางส่วนของ การนำแนวคิดนี้ไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## XML ACCESS CONTROL

Mr. Anuchin Chatchinarat

Mr. Anuruk Apisamajarn

Mr. Thana Hongsuwan Advisor

Academic Year 2003

## ABSTRACT

XML has become standard for transferring information both on internet and intranet. It's also common for transferring information on Web Service, called SOAP or even in Ecommerce. Information on display is also XML document. Then, XML technology is considered very interesting.

When XML document is used on internet's E-commerce, it is widely known that information is highly important. Various security systems have been made e.g. encryption and authentication for verification etc. Likewise, XML Document access control application is invented for the same reason. Two main applications, authentication and authorization, are required prior to access for security purpose. For reaching different levels of information, access rule is imposed to identify "right of way" in operating system. In addition, it can be modified from relational database to XML document prior to access as most of information are kept as relational database.

For this project, XML document access is modified for E-commerce application on internet to demonstrate how part of this concept can be applied for practical use.

## กิตติกรรมประกาศ

ปริญญาโทฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยคำแนะนำและความช่วยเหลือจากบุคคลหลายท่านด้วยกัน

ขอขอบพระคุณป้า, คุณพ่อ และคุณแม่ ผู้ให้กำเนิดและสนับสนุนทุกๆ อย่าง

ขอขอบพระคุณอาจารย์ธนา หงษ์สุวรรณ อาจารย์ผู้ให้ความรู้ ให้คำปรึกษาและคำแนะนำตลอดมา จนทำให้ปริญญาโทฉบับนี้สำเร็จสมบูรณ์

ขอขอบพระคุณที่เค่อ สำหรับคำแนะนำต่างๆ และความห่วงใยที่มีให้

ขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่อำนวยความสะดวกทุกสิ่งทุกอย่าง

ขอบคุณเพื่อนๆ ของพวกเรา ที่อยู่ด้วยกันมา 4 ปี และช่วยเหลือทุกสิ่งทุกอย่างตลอดมา

ขอบคุณนุ่นสำหรับกำลังใจและช่วยพิสูจน์อักษร

และขอบคุณทุกๆ คนที่คอยเป็นกำลังใจให้และความช่วยเหลือทุกสิ่งทุกอย่าง ขอขอบคุณครับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	X
สารบัญภาพ	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 Microsoft .NET Technology	3
2.1.1 .NET PLATFORM	3
2.1.1.1 ชั้นระบบปฏิบัติการ (Operating System)	3
2.1.1.2 ชั้นถัดมาแบ่งออกเป็น 3 ส่วน	3
2.1.1.3 ชั้น Visual Studio .NET เครื่องมือในการพัฒนา .NET	3
2.1.2 .NET Framework	3
2.1.2.1 Common Language Runtime (CLR)	4
2.1.2.1.1 Memory Management	6
2.1.2.1.2 Common Type System	6
2.1.2.2 .NET Framework Class Library	7
2.1.2.3 Presentation Layer	7
2.1.3 .NET Compact Framework	8
2.1.4 .NET Framework on Linux	9
2.1.4.1 .NET Framework Runtime (CLR)	9
2.1.4.2 NET Framework based class Library	9
2.1.4.3 C# Compiler	9
2.1.5 มาตรฐาน CLI และ C# ของ ECMA	9
2.2 XML	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้าที่
2.2.1 XML คืออะไร	10
2.2.1.1 ความเป็นมาของ XML (Extensive Markup Language)	10
2.2.1.2 ลักษณะโครงสร้างของ XML	13
2.2.1.3 วัตถุประสงค์หลักของ XML	16
2.2.1.4 การประยุกต์ใช้ XML	17
2.2.1.5 การใช้งานเอกสาร XML	19
2.2.1.6 XSLT และ XPath	19
2.2.1.6.1 XSLT	20
2.2.1.6.2 XSL คืออะไร	20
2.2.1.6.3 สไลด์รีด XSLT ทำงานอย่างไร	20
2.2.1.6.4 XPath	22
2.2.1.6.5 พาทบอกตำแหน่ง	22
2.2.1.7 DOM	22
2.2.1.7.1 DOM คืออะไร	23
2.2.1.7.2 DOMString	23
2.2.1.7.3 การใช้กับ DOM	24
2.2.1.8 ประโยชน์จาก XML	25
2.2.1.9 บทสรุปของ XML	26
2.3 ASP.NET เว็บแอปพลิเคชัน	26
2.3.1 ที่มาของ ASP	26
2.3.2 ที่มาของ ASP.NET	27
2.3.3 รู้จักกับ System.Web.UI Namespace	27
2.3.3.1 คลาส Control	27
2.3.3.1.1 คุณสมบัติของ Control	28
2.3.3.1.2 เมธอดในคลาส Control	28
2.3.3.2 คลาสเพจ	29
2.3.3.2.1 ตัวอย่างคุณสมบัติและเมธอดของคลาสเพจ	29
2.3.4 ไวยากรณ์สำหรับ Web Form	30
2.3.4.1 ไคเร็กทีฟ	30
2.3.4.1.1 @Page	30
2.3.4.1.2 @Control	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้าที่
2.3.4.1.3 @Import	30
2.3.4.1.4 @Register	30
2.3.4.1.5 @Assembly	31
2.3.4.1.6 @OutputCache	31
2.3.4.2 ขอบเขตการประกาศคำสั่ง	31
2.3.4.3 HTML-Control Syntax	31
2.3.4.4 Custom-Control Syntax	32
2.3.4.5 Data-Binding Expressions	32
2.3.5 การพัฒนาเว็บแอปพลิเคชันด้วย ASP.NET	33
2.3.5.1 ส่วนประกอบคอมโพเนนต์ของ Web Form	33
2.3.5.2 วงจรชีวิตของ Web Form	33
2.3.5.2.1 การกำหนดรูปแบบ	35
2.3.5.2.2 การถือครองอีเวนต์	36
2.3.5.2.3 การจบการทำงาน	36
2.3.6 การจัดการ State และ Scalability	36
2.3.6.1 ASP.NET Session-State Management	36
2.4 ADO.NET	37
2.4.1 สถาปัตยกรรม ADO.NET	37
2.4.2 ส่วนประกอบคอมโพเนนต์ของ ADO.NET	38
2.4.3 สรุปข้อเปรียบเทียบระหว่าง ADO และ ADO.NET	39
2.5 SQL 2000 Server	40
บทที่ 3 วิวัฒนาการของสถาปัตยกรรมแบบกระจาย	42
3.1 สถาปัตยกรรมแบบหนึ่งเทียร์ (One Tier Architecture/Centralize Host)	42
3.2 เครือข่ายภายในและการแบ่งปันทรัพยากรภายในเครือข่าย (PC LANs/ทรัพยากร Sharing LANs)	42
3.3 สถาปัตยกรรมแบบสองเทียร์ (ไคลเอนต์/เซิร์ฟเวอร์)	43
3.3.1 ดาต้าเบสเซิร์ฟเวอร์ (Database Server)	43
3.3.2 แอปพลิเคชันเซิร์ฟเวอร์	43
3.4 สถาปัตยกรรมแบบสามเทียร์	43
3.4.1 วินโดวส์ดีเอ็นเอ (Windows DNA)	44
3.4.1.1 ฟรีเซนต์ชันเทียร์	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้าที่
3.4.1.2 บิสิเนสโลจิกเทียร์ (Business Logic Tier)	44
3.4.1.3 คาด้าแอ็กเซสเลเยอร์ (Data Access Layer)	45
3.5 สถาปัตยกรรมแบบหลายเทียร์(N-Tier)	45
3.5.1 แอปพลิเคชันแบบหลายเทียร์ทางตรรก	45
3.5.2 แอปพลิเคชัน N-Tier โดยใช้ XML เว็บเซอร์วิส	46
3.5.3 แอปพลิเคชัน N-Tier โดยใช้รีโมต .NET	48
บทที่ 4 การวิเคราะห์และออกแบบระบบ	49
4.1 การวิเคราะห์ความต้องการของระบบ	49
4.1.1 หลักการ	49
4.1.1.1 แนวคิดการทำงานของระบบการเข้าถึงเอกสารเอกซเอมแอล	49
4.1.1.1.1 Subject	49
4.1.1.1.2 Object	49
4.1.1.1.3 Action	50
4.1.1.1.4 Permission	50
4.1.1.2 การออกแบบกฎการเข้าถึง	51
4.1.1.2.1 แนวคิดในการออกแบบคือ	51
4.1.1.2.2 ตัวอย่างการทำงาน	52
4.1.2 ขอบเขตการให้บริการ	54
4.1.2.1 บุคคลทั่วไป	54
4.1.2.2 สมาชิก	54
4.1.2.3 ผู้จัดการ	54
4.1.2.4 ผู้ดูแลระบบ	54
4.1.3 Use Case Diagram	55
4.1.3.1 การล็อกอิน (Log In)	58
4.1.3.2 การควบคุมการเข้าถึง (Access Control)	60
4.1.3.3 การแก้ไขข้อมูลส่วนตัว (Change Private Information)	61
4.1.3.4 การแก้ไขข้อมูลของผู้ดูแลระบบ Admin Change Information	61
4.1.3.4.1 การแก้ไขข้อมูลส่วนตัวข้อมูลผู้ใช้	61
4.1.3.4.2 การแก้ไขข้อมูลส่วนกฎการเข้าถึง	61
4.1.3.4.3 การแก้ไขข้อมูลส่วนฐานข้อมูล	61
4.2 การออกแบบระดับสถาปัตยกรรม (Architectural Design)	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้าที่
4.2.1 ส่วนไคลเอ็นต์ (Client)	63
4.2.2 ส่วนเว็บเซิร์ฟเวอร์ (Web Server)	63
4.2.3 ส่วนของดาต้าเบสเซิร์ฟเวอร์	64
4.2.4 รายละเอียดของระบบการทำงาน (System Specification)	64
4.3 การออกแบบระดับกลไกการทำงาน	64
4.3.1 คลาสไดอะแกรม (Class Diagram)	65
4.3.2 ดาต้าเบส (Database)	67
4.4 การออกแบบส่วนติดต่อผู้ใช้ (User Interface Design)	68
4.4.1 บุคคลทั่วไป (Public)	68
4.4.1.1 หน้าแรกของเว็บไซต์ (Index)	68
4.4.2 สมาชิก (Member)	69
4.4.2.1 การล็อกอิน (Login)	69
4.4.2.2 การเปลี่ยนแปลงข้อมูลส่วนบุคคลของสมาชิก (Change Personal Information)	70
4.4.3 ผู้จัดการ (Manager)	71
4.4.4 ผู้ดูแลระบบ (Administrator)	72
บทที่ 5 การทดลองและผลการทดลอง	76
5.1 จุดประสงค์การทดลอง	76
5.2 การเตรียมอุปกรณ์และสภาวะการทำงานเพื่อทดสอบโปรแกรม	76
5.3 ทำการทดลองโปรแกรม	76
5.3.1 ทดลองรันเรียกเว็บเพจ	76
5.3.2 ทดลอง login เข้าสู่ระบบ โดยเป็นแบบสมาชิก	77
5.3.3 ทดลอง login เข้าสู่ระบบ โดยเป็นแบบผู้จัดการ	77
5.3.4 ทดลองผู้จัดการเปลี่ยนค่า	78
5.3.5 ทดลองเข้าดูเอกสารของผู้จัดการ	79
5.3.6 ทดลองเปลี่ยนค่าในส่วนที่ไม่สามารถแก้ไขได้ในส่วนของผู้จัดการ	79
5.3.7 ทดลองผู้ดูแลระบบ login	80
5.3.8 ทดลองเข้าดูเอกสารของผู้ดูแลระบบ	80
5.3.9 ทดลองผู้ดูแลระบบเปลี่ยนข้อมูลผู้ใช้	81
5.3.10 ทดลองผู้ดูแลระบบเปลี่ยนข้อมูลกฎการเข้าถึง	81
5.3.11 ทดลองผู้ดูแลระบบแก้ไขข้อมูลสินค้า	82
5.3 ผลการทดลอง	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้าที่
บทที่ 6 บทวิจารณ์และสรุป	83
6.1 บทวิจารณ์และสรุป	83
6.2 ปัญหาและอุปสรรคที่พบ	83
6.3 แนวทางการศึกษาต่อ	84
ภาคผนวก ก	85
บรรณานุกรม	97



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ชื่อ	หน้าที่
ตารางที่ 4-1 รูปแบบการทำงาน	50
ตารางที่ 4-2 ขั้นตอนการทำงานของระบบการเข้าถึงเอกสารเอกซอแมด	51
ตารางที่ 4-3 Access Rule	52
ตารางที่ 4-4 ความหมายของแต่ละหลักของตาราง Access Rule	52
ตารางที่ 4-5 ความหมายขั้นตอนการทำงานของ Access Rule	52
ตารางที่ 4-6 Actor ของระบบ	55
ตารางที่ 4-7 Use Case ของระบบ	55
ตารางที่ 4-8 System Specification	64
ตารางที่ 4-9 แสดงการทำงานของคลาสไดอะแกรม	66
ตารางที่ 5-1 แสดงอุปกรณ์การทดลอง	76
ตารางที่ 5-2 ผลการทดลอง	82



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

ชื่อ	หน้าที่
รูปที่ 1-1 แสดงแนวคิดของระบบ	1
รูปที่ 2-1 องค์ประกอบ .NET Platform	3
รูปที่ 2-2 องค์ประกอบ .NET Framework	4
รูปที่ 2-3 รูปแบบการคอมไพล์โค้ดไปเป็น IL Code	5
รูปที่ 2-4 ส่วนประกอบคอมโพเนนต์ของ Web Form	33
รูปที่ 2-5 แสดงสถาปัตยกรรม ADO.NET	37
รูปที่ 2-6 เปรียบเทียบระหว่าง SQL Server .NET Data Provider และ OLEDB .NET Data Provider	39
รูปที่ 3-1 สถาปัตยกรรมแบบหนึ่ง-tier	42
รูปที่ 3-2 เครื่องข่ายภายในของเครื่องคอมพิวเตอร์ส่วนบุคคล	42
รูปที่ 3-3 สถาปัตยกรรมแบบ 2-tier	45
รูปที่ 3-4 สถาปัตยกรรมแบบ 2-tier ผ่านเว็บเซิร์ฟเวอร์	47
รูปที่ 3-5 สถาปัตยกรรมแบบ 3-tier ผ่าน .NET Remoting	48
รูปที่ 4-1 แนวคิดของระบบการเข้าถึงเอกสารเอกซเอมแอล	50
รูปที่ 4-2 ขั้นตอนการทำงานของ Access Rule	52
รูปที่ 4-3 เอกสารเอกซเอมแอลและโครงสร้างต้นไม้ของเอกสาร	53
รูปที่ 4-4 แสดงการทำงานกับ โครงสร้างต้นไม้	53
รูปที่ 4-5 เอกสารเอกซเอมแอลหลังจากทำการตรวจสอบสิทธิ์	54
รูปที่ 4-6 แผนภาพ Use Case ของ ผู้ดูแลระบบ	56
รูปที่ 4-7 แผนภาพ Use Case ของ สมาชิก, บุคคลทั่วไป, ผู้จัดการ	57
รูปที่ 4-8 แผนภาพ Sequence Diagram of Login ของ บุคคลทั่วไป	58
รูปที่ 4-9 แผนภาพ Sequence Diagram of Login ของ สมาชิก	59
รูปที่ 4-10 แผนภาพ Sequence Diagram of Login ของ ผู้จัดการ	59
รูปที่ 4-11 แผนภาพ Sequence Diagram of Login ของ ผู้ดูแลระบบ	60
รูปที่ 4-12 แผนภาพ Sequence Diagram of Access Control	60
รูปที่ 4-13 แผนภาพ Change Private Information	61
รูปที่ 4-14 แผนภาพ Admin Change Information	62
รูปที่ 4-15 สถาปัตยกรรมของระบบการเข้าถึงเอกสารเอกซเอมแอล	63
รูปที่ 4-16 Class Diagram	65
รูปที่ 4-17 ER Diagram ของระบบ	67
รูปที่ 4-18 Database Table	67

## สารบัญภาพ (ต่อ)

ชื่อ	หน้าที่
รูปที่ 4-20 การ Login เข้าสู่ระบบ	69
รูปที่ 4-21 แสดงหน้าแรกของสมาชิกคนที่ 1	69
รูปที่ 4-22 แสดงหน้าแรกของสมาชิกคนที่ 2	70
รูปที่ 4-23 แสดงส่วนการเปลี่ยนข้อมูลและแสดงเอกสาร XML	70
รูปที่ 4-24 แสดงการเปลี่ยนแปลงข้อมูลส่วนตัว	71
รูปที่ 4-25 แสดงหน้าเว็บเพจเอกสาร XML ของสมาชิก	71
รูปที่ 4-26 แสดงหน้าเว็บผู้จัดการ	72
รูปที่ 4-27 แสดง Drop Down Menu สำหรับแก้ไขข้อมูล	72
รูปที่ 4-28 แสดงหน้าเว็บของ Administrator	73
รูปที่ 4-29 เมนูสำหรับ Admin	73
รูปที่ 4-30 แสดงหน้าเว็บ Admin ดูเอกสาร XML	74
รูปที่ 4-31 แสดง Edit USER	74
รูปที่ 4-32 แสดง Edit RULE	75
รูปที่ 4-33 แสดง Edit DATA	75
รูปที่ 5-1 ทดสอบเรียกเก็บเพจ	76
รูปที่ 5-2 ทดลอง login เข้าระบบ โดยเป็นแบบสมาชิก	77
รูปที่ 5-3 ทดลอง login เข้าระบบโดยเป็นแบบผู้จัดการ	77
รูปที่ 5-4 ทดลองผู้จัดการเปลี่ยนค่า	78
รูปที่ 5-5 ผลการทดลองผู้จัดการเปลี่ยนค่า	78
รูปที่ 5-6 ผลการทดลองเข้าดูเอกสาร	79
รูปที่ 5-7 ผลการทดลองเปลี่ยนค่าในส่วนที่ไม่สามารถแก้ไขได้ในส่วนของผู้จัดการ	79
รูปที่ 5-8 ทดลอง login เข้าระบบ โดยเป็นแบบผู้ดูแลระบบ	80
รูปที่ 5-9 ผลการทดลองเข้าดูเอกสาร	80
รูปที่ 5-10 ผลการทดลองผู้ดูแลระบบเปลี่ยนข้อมูลผู้ใช้	81
รูปที่ 5-11 ผลการทดลองผู้ดูแลระบบเปลี่ยนข้อมูลกฎการเข้าถึง	81
รูปที่ 5-12 ผลการทดลองแก้ไขข้อมูลสินค้า	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

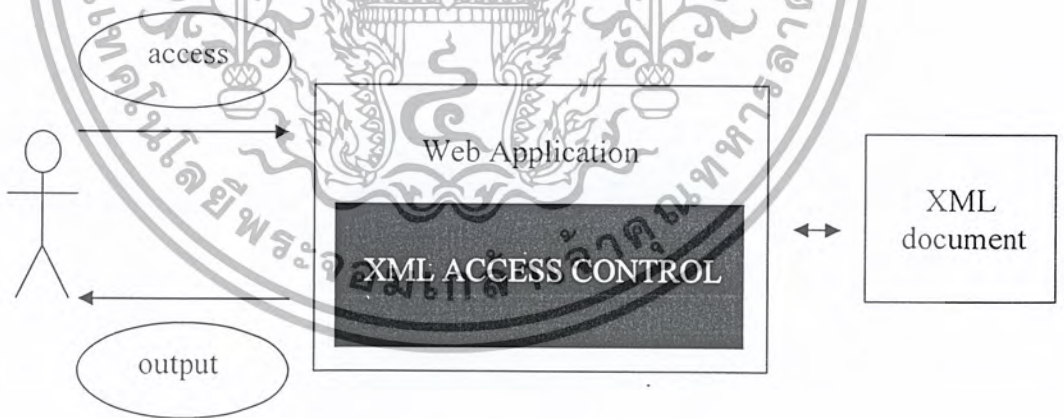
## บทนำ

### 1.1 ความเป็นมาของโครงการ

เอกซเอ็มแอล (The extensible markup language, XML) เป็นมาตรฐานหนึ่งในอนาคตของการเก็บข้อมูลและนำเสนอข้อมูลทางอินเทอร์เน็ต เนื่องจากเอกซเอ็มแอลเป็นภาษาที่ให้ความชัดเจนในรายละเอียดเกี่ยวกับข้อมูล เป็นภาษาที่เข้าใจง่าย และมีความยืดหยุ่น จึงได้ถูกนำไปใช้อย่างมากมายในโลกทุกวันนี้ โดยเฉพาะอีคอมเมิร์ซ (E-Commerce) ที่ทุกวันนี้ได้รับความนิยม

แต่ทว่าในการทำอีคอมเมิร์ซนั้นการนำเสนอข้อมูลย่อมมีการนำเสนอข้อมูลที่แตกต่างกันถึงแม้ว่าจะเป็นข้อมูลตัวเดียวกันก็ตาม อย่างเช่น การนำเสนอราคาของสินค้า บริษัทหนึ่งอาจจะนำเสนอสินค้าพิเศษสำหรับลูกค้าบางราย และนำเสนอราคามาตรฐานแก่ลูกค้าทั่วไป หากเรเก็บข้อมูลไว้ในเอกสารเอกซเอ็มแอลเดียวกัน โดยไม่มีการจัดการ ข้อมูลสำรองก็จะแสดงขึ้นมา จึงทำให้ผิดจุดประสงค์

ดังนั้นโครงการนี้จึงมองเห็นความสำคัญของการเข้าถึงข้อมูลในเอกสารเอกซเอ็มแอลจึงได้คิดค้นระบบจัดการการเข้าถึงเอกสารเอกซเอ็มแอลขึ้นมา เพื่อเพิ่มความปลอดภัยในการเข้าถึงข้อมูลเพิ่มขึ้นอีกระดับ เพื่อตอบสนองความต้องการนำไปใช้งาน



รูปที่ 1-1 แสดงแนวคิดของระบบ

โดยที่โครงการนี้ได้พัฒนาตัวระบบการจัดการการเข้าถึงเอกสารเอกซเอ็มแอล (XML ACCESS Control) โดยการนำระบบที่พัฒนาขึ้นมาไปใส่ไว้ในเว็บแอปพลิเคชัน (Web Application) เพื่อแสดงความสามารถและประโยชน์ของโครงการซึ่งการพัฒนาจะใช้เทคโนโลยีต่างๆ ซึ่งมีอยู่ในปัจจุบันมาทำการ

พัฒนาขึ้นมา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 วัตถุประสงค์

1. เพื่อเป็นการพัฒนาและทดสอบระบบการจัดการการเข้าถึงเอกสารเอกซเอม (XML Access Control) ว่าสามารถนำไปประยุกต์ใช้งานได้จริง
2. เพื่อเพิ่มความน่าเชื่อถือการเข้าถึงข้อมูลเอกซเอมแอล
3. เพื่อเป็นการเรียนรู้ในการทดลอง ออกแบบและพัฒนาซอฟต์แวร์ที่จัดการการเข้าถึงเอกสารเอกซเอมแอลตามรูปแบบมาตรฐานของเอกซเอมแอล
4. เพื่อเป็นการพัฒนาองค์ความรู้ (Knowledge Base) สำหรับนำไปใช้ในการวิจัยและพัฒนาทางการเข้าถึงข้อมูลรวมถึงการนำไปประยุกต์ใช้ในอนาคต
5. เพื่อพัฒนามาตรฐานการจัดการการเข้าถึงข้อมูลขึ้นมาเพื่อลดการนำเข้าเทคโนโลยีจากต่างประเทศ

## 1.3 ขอบเขตของโครงการ

1. ระบบการจัดการการเข้าถึงเอกสารเอกซเอมแอล (XML Access Control) พร้อมทั้งองค์ประกอบต่างๆ
2. เว็บแอปพลิเคชัน (Web Application) ของระบบการนำเสนอข้อมูลของตัวสินค้าที่มีการรวมระบบการจัดการการเข้าถึงเอกสารเอกซเอมแอลเข้าไป

## 1.4 วิธีการดำเนินงาน

1. ทำการศึกษาระบบจัดการการเข้าถึงข้อมูลที่มีอยู่ รวมทั้งหาข้อดีข้อเสียของระบบเดิมที่มีอยู่
2. ออกแบบแนวคิดการเข้าถึงข้อมูลโดยการปรับปรุงหรือเพิ่มเติมของเดิมที่มีอยู่แล้ว
3. วิเคราะห์และออกแบบระบบจากแนวคิดที่ได้ โดยมีรายละเอียดดังนี้
  - วิเคราะห์ความต้องการต่าง ๆ ของระบบ และเสนอกฎเกณฑ์และข้อกำหนดต่าง ๆ ในการออกแบบเว็บแอปพลิเคชัน โดยใช้แนวคิดที่ได้ศึกษามา
  - ออกแบบระบบโดยใช้วิธีการออกแบบแบบออบเจกต์โอเรียนเท็ด (Object Oriented) และใช้ภาษา UML ช่วยในการออกแบบ
  - ออกแบบฐานข้อมูลบนระบบแบบเรชันแนล (Relational Database) โดยใช้ ER Diagram และทำการนอร์มาไลเซชัน (Normalization)
  - ออกแบบเว็บไซต์ที่นำระบบจัดการการเข้าถึงข้อมูลเอกซเอมแอลไปใช้งาน
4. ศึกษาหลักการและทฤษฎีต่าง ๆ ที่ใช้ในการสร้างระบบ ( Tool and Programming Technique) และทำการสร้างระบบตามที่ได้ออกแบบไว้
5. ทดสอบระบบ, สรุปผลงาน และให้ข้อเสนอแนะแนวทางในการพัฒนาโครงการต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

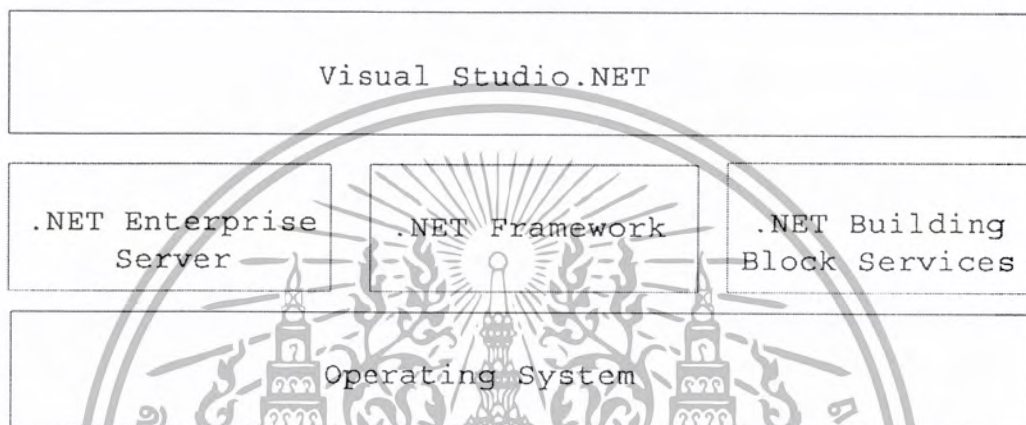
## บทที่ 2

# หลักการและทฤษฎี

### 2.1 Microsoft .NET Technology

#### 2.1.1 .NET PLATFORM

องค์ประกอบของ แพลตฟอร์ม .NET แบ่งออกได้เป็น 3 ชั้น



รูปที่ 2-1 องค์ประกอบ .NET Platform

##### 2.1.1.1 ชั้นระบบปฏิบัติการ (Operating System)

##### 2.1.1.2 ชั้นถัดมาแบ่งออกเป็น 3 ส่วน ได้แก่

- .NET Enterprise Server
  - SQL Server 2000, BizTalk Server, Exchange 2000
- .NET Framework
  - เป็น Framework ที่สำคัญในการพัฒนา .NET
- .NET Building Block Services
  - Microsoft passport, Notification and Messaging, Personalization , XML Store, Calendar, Directory and Search

##### 2.1.1.3 ชั้น Visual Studio .NET เครื่องมือในการพัฒนา .NET

#### 2.1.2 .NET Framework

.NET Framework มีองค์ประกอบ ดังนี้

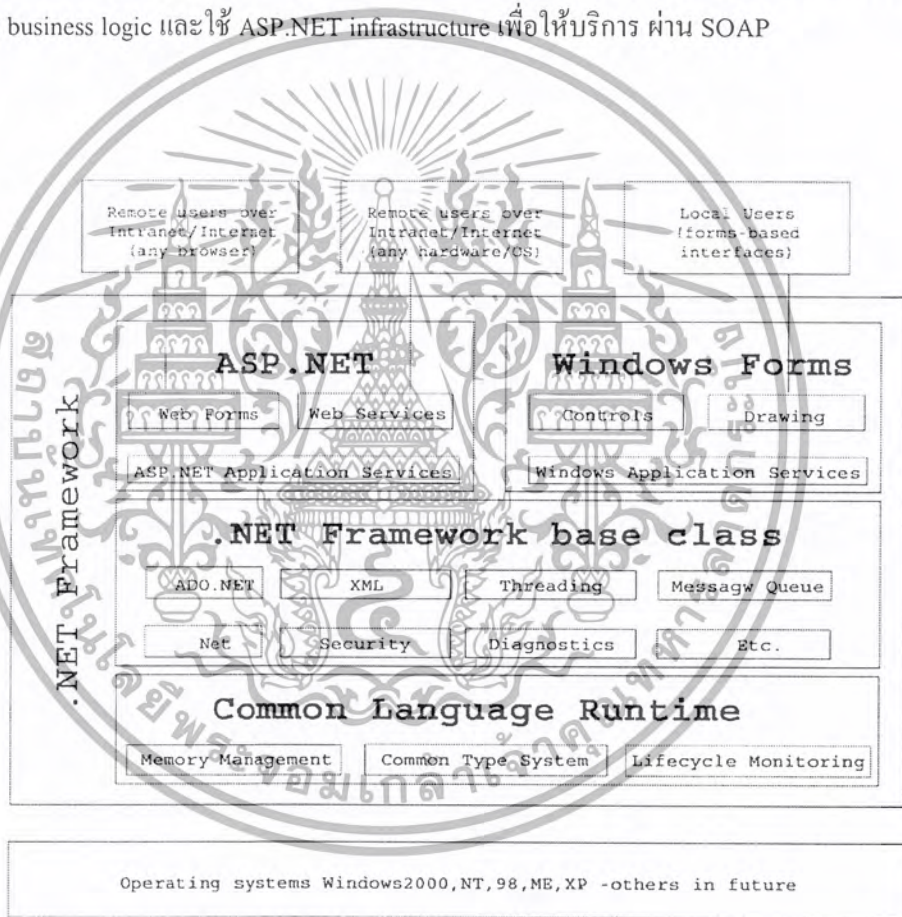
1. Common Language Runtime (CLR) มีหน้าที่ในการจัดเตรียมบริการต่างๆที่เกี่ยวข้อง

เพื่อให้โปรแกรมที่ขอใช้บริการสามารถทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. .NET Framework Class Library (Unified programming classes) คือ ไฟล์ไลบรารีที่ทำหน้าที่จัดเก็บและรวบรวมข้อมูลที่เป็นต้องเรียกใช้งาน ในการพัฒนาโปรแกรม

3. ASP.NET เป็น Web application model ซึ่งประกอบไปด้วย กลุ่มของ control และ Infrastructure ที่ใช้ในการสร้าง ASP Web applications , ASP.NET ประกอบไปด้วยส่วนของ control ซึ่งรวมถึงส่วนของ HTML ด้วย ส่วนของ control รันบนเซิร์ฟเวอร์ ส่วน HTML รันบน browser บนเซิร์ฟเวอร์นั้นส่วน control จะถูกเขียนในรูปแบบของ object-oriented programming , ASP.NET ยังให้บริการส่วน infrastructure ด้วยเช่น session state management และ process recycling ยิ่งไปกว่านั้น ASP.NET ก็ใช้คอนเซปต์นี้ในการพัฒนา จาก software เป็น service โดยผ่าน กลไกสำคัญคือ XML เราสามารถเขียน business logic และใช้ ASP.NET infrastructure เพื่อให้บริการ ผ่าน SOAP



รูปที่ 2-2 องค์ประกอบ .NET Framework

### 2.1.2.1 Common Language Runtime (CLR)

CLR จะทำหน้าที่เป็นตัวกลางในการจัดเตรียมบริการและทรัพยากรสำหรับรองรับการประมวลผล และการทำงานของโปรแกรมประยุกต์ที่ทำงานบนเทคโนโลยีของ .NET เช่น การจัดการหน่วยความจำ, ความปลอดภัยในการเข้าถึงโปรแกรม ด้วยความสามารถของ CLR ทำให้เราสามารถพัฒนาโปรแกรมโดยไม่ต้องขึ้นกับระบบปฏิบัติการ

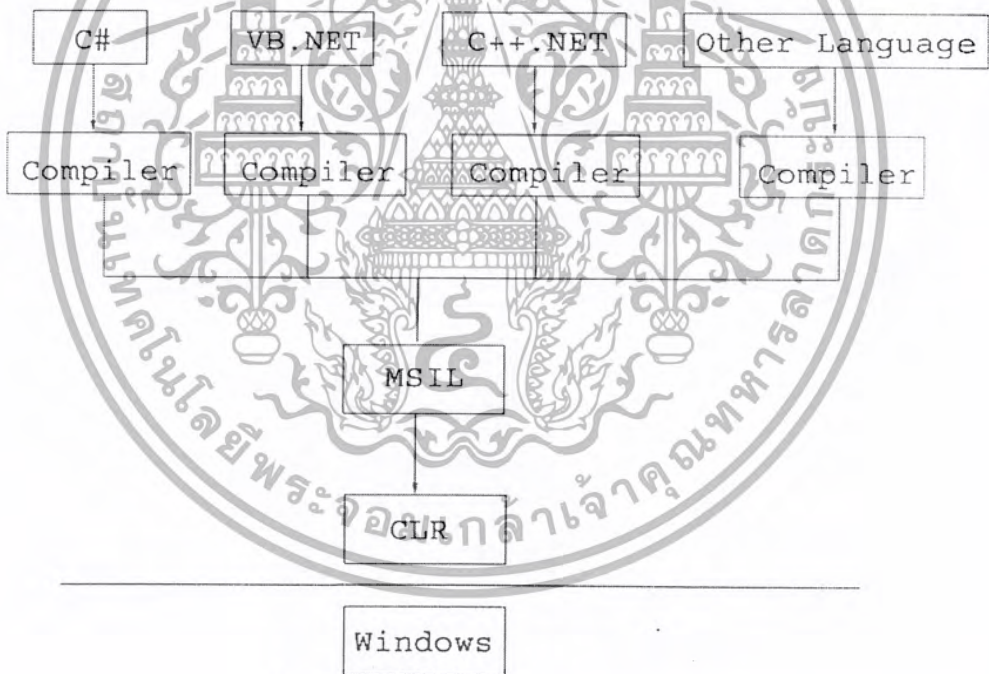
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมนั้น เริ่มจากคอมไพเลอร์ของแต่ละภาษาจะคอมไพล์โค้ดให้เป็นแบบ Microsoft's Intermediate Language (MSIL) หรือเรียกสั้นๆว่า IL Code ซึ่งจะมีลักษณะคล้ายภาษา Assembly จาก IL Code ก็แปลงเป็นโปรแกรมที่รันโดย CLR อีกทีหนึ่ง

นักพัฒนาสามารถที่จะเลือกได้ว่า เราจะ build แอปพลิเคชันไปอยู่ในรูปของ .exe หรือ MISL ซึ่งจะกลายเป็น Just-In-Time(JIT) คือเมื่อเราได้แอปพลิเคชันในรูปแบบของ MISL แล้ว เมื่อรันโปรแกรมใช้งานจริง มันจะถูก คอมไพเลอร์ JIT ทำการคอมไพล์โค้ด MSIL ในส่วนที่ต้องการใช้ไปเป็น Native Code อีกทีซึ่งนำไปให้เครื่องรันต่อ หากมีการใช้โค้ดในส่วนเดิมอีกก็ไม่ต้องมีการคอมไพล์ซ้ำ

การแปลงโค้ด MSIL ไปเป็น Native Code ประโยชน์ที่เราจะได้คือ

1. เราสามารถแปลงไปเป็น Native Code ที่เหมาะสมกับระบบปฏิบัติการนั้นรันอยู่
2. เราสามารถได้ Native Code ที่ใช้ความสามารถของ CPU ได้อย่างเต็มที่ เช่นใน CPU P4 เราสามารถใช้คำสั่งในส่วนของ SSE2 ได้



รูปที่ 2-3 รูปแบบการคอมไพล์โค้ดไปเป็น IL Code

Managed Code	เป็นโค้ดที่คอมไพล์ให้ทำงานใน .NET Framework
Native Code	เป็นโค้ดที่คอมไพล์เป็นภาษาเครื่องโดยตรง ไม่ต้องผ่านตัว CLR ในการรันโปรแกรม
Unmanaged Code	เป็นโค้ดที่ไม่ได้คอมไพล์เพื่อใหทำงานใน .NET Framework เช่น COM Object ใน version ก่อนหน้านี้ ซึ่ง .NET ก็ยังสามารถรันโค้ดพวกนี้ได้ แต่ไม่สามารถใช้ความสามารถของ .NET ได้อย่างเต็มที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2.1.1 Memory Management

สำหรับ .NET แล้วจะมี Garbage Collector (GC) ซึ่งทำหน้าที่คอยรวบรวมหาหน่วยความจำที่ถูกทิ้งไว้เป็นขยะ แต่ไม่มีใครอ้างอิงเรียกใช้แล้ว โดยการทำงานจะแตกต่างจากใน VB Runtime รุ่นเก่าที่ให้ Object รับผิดชอบในการจัดการ Garbage Collector เป็นผู้ดูแลจัดการทั้งหมด โดย GC จะเรียก method ชื่อ Object.Finalize โดยอัตโนมัติเมื่อ object เลิกใช้งาน

การจัดการหน่วยความจำที่ดีกว่าของ .NET เกิดขึ้นเมื่อ object ถูกสร้างขึ้นมา CLR จะจัดการกันพื้นที่ส่วนหนึ่งไว้สำหรับโปรแกรมที่เรียกว่า Heap โดยเริ่มต้นจากพื้นที่หน่วยความจำที่ว่าง จากนั้นก็อ้างอิงชี้ตำแหน่งไปยังส่วนบนสุดของหน่วยความจำ เมื่อหน่วยความจำถูกใช้แล้วยกเลิกไปเรื่อยๆ จะเหลือพื้นที่ว่างเป็นช่วงๆ ซึ่ง GC จะเข้ามาทำการ Compact หน่วยความจำเสียใหม่ ลองนึกถึงหลักการทำงานแบบเดียวกันกับตอนที่ทำ Defragment บนฮาร์ดดิสก์

### 2.1.2.1.2 Common Type System

ใน .NET จะมี common Type System ที่จะให้ทุกภาษามี Types ที่เหมือนกัน เป็นมาตรฐาน โดยที่ทุก Types ที่สนับสนุนโดย Common Types System นั้น จะสืบทอดคุณสมบัติมาจาก System.Object ดังนั้นคุณจะพบว่า Object ทั้งหลายส่วนมากแล้วจะสนับสนุน Method เหล่านี้คือ

Equal(Object) = Boolean

GetHashCode() = Int32

GetType() = Type

Tostring() =String

#### ข้อดีของ Common Language Runtime

- ไม่มีปัญหาเรื่องการรัน บน MS Windows Platform ต่างๆ เนื่องจาก CLR จะทำการตรวจสอบระบบปฏิบัติการให้โดยอัตโนมัติ และทำการจำลองสภาพให้เหมาะสมกับการทำงานของโปรแกรมโดยอัตโนมัติเช่นเดียวกัน

- ไม่ต้องสนใจเรื่อง Registry การเข้าถึง component ต่างๆที่เกี่ยวข้องจะอยู่ในความดูแลของ CLR ทั้งหมด เราจำเป็นต้องรู้แค่จะใช้ Namespace ตัวใดในการอ้างอิงถึงเท่านั้น

- รองรับการพัฒนาจากหลายภาษา ในปัจจุบันเราสามารถใช้อย่างน้อยสามภาษาทำงานร่วมกันได้ด้วยเทคโนโลยีของ COM แต่อย่างไรก็ตาม เนื่องจากโปรแกรมยังผูกติดกับภาษาอยู่ ความเร็วของโปรแกรมที่ได้ก็จะแตกต่างกัน ขึ้นอยู่กับความสามารถของ Compiler ภาษานั้นๆ Microsoft Common Language Specification จะทำให้ compiler ทุกตัวที่ใช้ .NET ทำการแปลภาษาไปยังเป้าหมายเดียวกัน คือการทำงานกับ CLR ได้อย่างไร้ปัญหา ส่งผลให้ โปรแกรมที่ได้จากและภาษามีผลลัพธ์ที่เท่าเทียมกันเมื่อทำงานอยู่ภายใต้การดูแลของ CLR

- นำ Source code มาใช้งานใหม่ได้ COM ยังถูกจำกัดการ reuse ในวงแคบเนื่องจากการผูกติดกับระบบปฏิบัติการ .NET แยกออกได้ 3 รูปแบบได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เราสามารถเขียน class ด้วยภาษาหนึ่ง สามารถเรียกใช้ผ่านอีกภาษาหนึ่งได้
2. คลาสสามารถ Inherit ข้ามภาษาได้
3. Web Service เป็นอีกรูปแบบหนึ่งโดยที่เสนอบริการให้แก่ทุกภาษาที่อ้างอิง CLR ได้
4. ชนิดของข้อมูลมีประสิทธิภาพ จากข้อผิดพลาดเกี่ยวกับตัวแปรเช่น ประกาศตัวแปรขนาด 10 ไบต์ แต่ทำงานจริงกลับมีขนาด 20 ไบต์ ถ้าไม่มีการเขียนโปรแกรมดักจับ โปรแกรมจะทำงานผิดพลาด ใน .NET สิ่งเหล่านี้จะไม่เกิดขึ้น CLR จำตรวจสอบการทำงานของ code อย่างใกล้ชิดก่อนที่จะประมวลผลขั้นสุดท้าย CLR จะไม่อนุญาตให้โปรแกรมใช้ตัวแปรที่มีการผิดพลาดเหล่านั้น

5. การ Debug ที่มีประสิทธิภาพ CLR จะทำการแยกส่วนของโปรแกรมเป็นส่วนๆทำให้การตรวจสอบข้อผิดพลาดทำได้ง่ายสะดวกและรวดเร็ว การแก้ไขสามารถแก้ไขเฉพาะส่วนที่ผิดพลาดได้โดยไม่รบกวนส่วนอื่นๆของโปรแกรม และเนื่องจาก CLR จัดการกับทุกภาษาด้วยกัน ข้อผิดพลาดต่างๆที่แสดงออกมาในแต่ละภาษาจะอยู่ในรูปแบบเดียวกัน

6. ระบบจัดสรรทรัพยากรที่ดีขึ้น การดำเนินการที่เกี่ยวข้องกับทรัพยากรต่างๆ ระหว่างโปรแกรม ไม่ว่าจะเป็นระบบการจัดการ ไฟล์ระบบเครือข่าย หรือพื้นที่ว่างของหน้าจอ เป็นต้น จะถูกดูแลจัดการอย่างใกล้ชิด โดย CLR ไม่สามารถมีโปรแกรมตัวใดตั้งหนึ่งครอบครองทรัพยากรไว้แต่เพียงผู้เดียว

7. การแจกจ่ายโปรแกรมง่ายขึ้น การติดตั้งโปรแกรมแบบเดิม โปรแกรมหนึ่งๆมีไฟล์เป็นจำนวนมาก ,ต้องมีการลงทะเบียนใน registry ,การสร้าง short cut , การจัดเตรียมระบบที่เกี่ยวข้อง ใน .NET การติดตั้งโปรแกรมเป็นไปได้โดยง่าย เนื่องจาก CLR จะทำการแยกโปรแกรมออกเป็นส่วนๆ การติดตั้งโปรแกรมจึงเป็นการ copy ไฟล์ที่เกี่ยวข้องมาไว้บน harddisk เท่านั้น ในทางกลับกัน การยกเลิกการติดตั้งก็เพียงลบไฟล์เหล่านั้นออก

8. ระบบความปลอดภัยที่ดีขึ้น ระบบรักษาความปลอดภัยใหม่นี้จะทำการตรวจสอบพฤติกรรมการทำงานของโปรแกรมอย่างใกล้ชิด อย่างในระบบเดิมนั้น การทำงานของ ภาษาสคริปต์ต่างๆหรือโปรแกรมที่มาจาก Internet อาจเป็นโปรแกรมที่สร้างช่องโหว่ให้กับระบบได้

#### 2.1.2.2 .NET Framework Class Library

ไฟล์ไลบรารีที่ทำหน้าที่จัดเก็บและรวบรวมข้อมูลจำเป็นต้องเรียกใช้งาน ในการพัฒนาโปรแกรม ซึ่งไฟล์เหล่านี้ เป็นไลบรารีร่วมกันในทุกภาษา ส่งผลให้เราสามารถ inheritance ,error handling, debugging ข้ามภาษาได้ และการที่ .NET Framework เสนอบริการสำเร็จรูปผ่าน Class Library เหล่านี้ทำให้ รูปแบบการเขียนโปรแกรมเปลี่ยนไป โดยมีความเป็นระบบ และเป็นลักษณะของ OOP เพิ่มขึ้น เช่นใน VB.NET ต้องมีการอ้างอิงถึง class library ที่เกี่ยวข้องและต้องการใช้งาน

#### 2.1.2.3 Presentation Layer

การแสดงผลสามารถทำได้ 2 รูปแบบหลัก คือในรูปแบบ Windows form และ Web form ซึ่งก่อนนี้ทั้งคำสั่ง และกระบวนการเขียนโปรแกรม ของทั้ง 2 แบบ แต่ใน .Net แล้วได้รวมทั้งสองอย่างมาทำให้เหมือนกันที่สุด คือเราเขียนโปรแกรมบน Windows อย่างไร คุณก็เขียน Web Application ด้วยวิธีเดียวกัน โดยเรียกใช้ Class ด้วยคำสั่งที่เหมือนกัน บน Windows จะอยู่ใน class ที่ชื่อ system.windows แต่ Web จะอยู่ใน class ชื่อ system.web.ui ซึ่งวิธีใช้เหมือนกันมาก ที่พิเศษคือ มี Object สำหรับ Web ให้คุณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนโปรแกรม แบบ visual basic ธรรมดา แต่เมื่อรันบน browser แล้วจะสร้าง tag html และ javascript ให้อัตโนมัติ ทำให้ browser ทั่วๆไปก็สามารถรันได้ ไม่จำเป็นต้องสนับสนุน .net โดยเฉพาะ ตัวอย่างเช่น เราสร้างเว็บเพจที่มี ปฏิทิน ด้วย(นึกถึง calendar ใน VB) ในส่วนของ รูปแบบ web จะแสดง ในรูปของ html code ส่วนปฏิทินจะอยู่ในรูปของ java script ซึ่งทางโปรแกรมจะทำการแปลงให้เอง

### 2.1.3 .NET Compact Framework

.NET Compact Framework คือ platform ของ Microsoft .NET ที่ใช้ในการพัฒนาอุปกรณ์เล็กๆ ความจริงแล้ว .NET Compact Framework ก็คือ .NET Framework ที่ตัดเอาส่วนที่ไม่สำคัญหรือไม่ได้ใช้ออกไป เพื่อให้ขนาด เล็กลงจนขนาดที่สามารถ install ลงบนอุปกรณ์เล็กๆ ที่ไม่ใช่คอมพิวเตอร์ได้ เช่น โทรศัพท์มือถือ เป็นต้น

คุณสมบัติหลักของ .NET Compact Framework

- .NET Compact Framework ถูกออกแบบให้เป็น platform ในการพัฒนา สำหรับการเขียน และเรียกใช้ XML Web Services
- เนื่องจาก .NET Compact Framework ใช้ programming model เดียวกันกับ desktop .NET Framework จึงง่ายต่อ developer ในการเขียน application ใหม่บน .NET Compact Framework
- ใช้ Visual Studio .NET ตัวเดียวกับที่ใช้ใน desktop .NET Framework ในการเขียน program device application
- ใช้ระบบ evidence-based security ที่อนุญาตให้เฉพาะ application ที่ผ่านการรับรองจากผู้ผลิต device นั้น จึงจะสามารถ access secure resource ได้ ถ้าเป็น application ที่ยังไม่ผ่านการรับรองความปลอดภัย จะ access ได้เพียง basic system resource เท่านั้น
- ออกแบบสำหรับอุปกรณ์ที่มี resource จำกัด เช่น โทรศัพท์มือถือ โดยที่ตัว framework เองใช้ memory อย่างมีประสิทธิภาพ และมีระบบดึง resource ที่ไม่ได้ใช้แล้วกลับมาจาก application อย่างมีประสิทธิภาพ
- ใช้ Just-in-time compiler ซึ่งมีประสิทธิภาพดีกว่า ระบบ device programming อื่นๆ ที่ใช้ code interpreter
- เพิ่มประสิทธิภาพและลดต้นทุนในการพัฒนา เนื่องจากใช้ programming model และ toolset ของ Visual Studio .NET อันเดียวกันกับของ desktop .NET framework ทำให้นักพัฒนาสามารถใช้ความรู้ ความชำนาญที่มีอยู่แล้วมาพัฒนา application บน อุปกรณ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1.4 .NET Framework on Linux

ไมโครซอฟท์ ทำการส่ง spec มาตรฐานของ .net framework ให้แก่ ECMA เพื่อสร้างเป็นมาตรฐานสากล ส่งผลให้ .NET Framework เป็น Language Independent และ Platform Independent ตอนนี้บริษัท XIMIAN ทำการสร้าง framework บน Linux แล้ว ประกอบไปด้วย CLR , BASE CLASS , C# Compiler

### 2.1.4.1 .NET Framework Runtime (CLR)

Ximian ตั้งชื่อ project ในการทำ .NET Framework บน Linux ว่า Mono โดยจะมีหลักอยู่ 3 โปรแกรม ดังนี้

1. mint โปรแกรมนี้คือโปรแกรม .NET Framework Runtime บน Linux โดยสามารถเอาโปรแกรมที่เขียนโดยใช้ .NET Framework บน Windows มาใช้งานได้เลย
2. pedump เป็นโปรแกรมที่ดูโครงสร้างของแฟ้ม .exe ที่เป็น Managed code
3. monodis เป็นโปรแกรมที่แสดงให้เห็นถึงภาษา MSIL ของ .exe นั้น

### 2.1.4.2 NET Framework based class Library

การ Implement Library ของ .NET นั้น มีความซับซ้อนยิ่งกว่าส่วนอื่นๆ หากสนใจสถานะว่า Library ตัวใดทำได้แล้วมากน้อยเพียงใด สามารถหาข้อมูลได้จาก <http://www.go-mono.com/class-status/index.html>

### 2.1.4.3 C# Compiler

Ximian ทำคอมไพเลอร์ภาษา C# ที่ชื่อว่า mcs (Mono CSharp compiler)

## 2.1.5 มาตรฐาน CLI และ C# ของ ECMA

เดือนสิงหาคม ปี 2000 ไมโครซอฟท์ ฮิเลตต์ เพกการ์ด และ อินเทล ได้ร่วมเป็นผู้สนับสนุนร่วมเสนอการกำหนดรายละเอียดสำหรับ CLI และ ภาษา C# ให้กับองค์กรการจัดมาตรฐานระหว่างประเทศ ECMA ผลคือ ECMA ได้ตั้งมาตรฐานกลุ่มการทำงานขึ้นมา 2 กลุ่ม สำหรับ CLI และ C# คือ TC39-TG3 และ TC39-TG2 ตามลำดับซึ่งเป็นคณะกรรมการเทคนิคซึ่งรับผิดชอบเกี่ยวกับภาษาในการเขียนโปรแกรมและการพัฒนาแอปพลิเคชัน

ระหว่างปีถัดมาผู้ให้การสนับสนุนร่วมกับสมาชิก ECMA อื่นๆ และ ผู้รับเชิญ (รวมถึง IBM Fujitsu Software, Plum Hall, Monash University และ ISE) ได้ร่วมทำการกลั่นกรองรายละเอียด เป็นมาตรฐานในธันวาคมปี 2001 สมัชชาสหประชาชาติ ECMA อนุมัติ มาตรฐานฉบับที่ 1 ของ C# และ CLI เป็น ECMA -334 และ ECMA -335 ตามลำดับ รายงานเกี่ยวกับ CLI ได้ถูกอนุมัติเป็นมาตรฐาน ECMA TR84 ด้วย

อ้างอิงจาก

<http://msdn.microsoft.com/net/ecma>

<http://www.ecma-international.org/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 XML

### 2.2.1 XML คืออะไร

เนื่องจากทุกวันนี้โลกของเรามีการนำเทคโนโลยีสารสนเทศเข้ามาใช้ในชีวิตประจำวันมากขึ้น และได้เข้ามาเป็นส่วนหนึ่งของการทำงาน ที่เห็นได้ชัดคือ เทคโนโลยีบนอินเทอร์เน็ต ไม่ว่าจะเป็นการเข้าไปเยี่ยมชมเว็บไซต์ต่าง ๆ หรือการส่งจดหมายอิเล็กทรอนิกส์ที่เรียกว่า อีเมล ซึ่งกำลังเป็นมาตรฐานของการติดต่อสื่อสารสำหรับอนาคต ทำให้ต้องมีการคิดเพื่อพัฒนาให้มีความก้าวหน้ามากขึ้น การเขียนเว็บไซต์ในปัจจุบันนี้ โดยปกติแล้วสิ่งที่จะใช้สร้างเว็บได้คือ ภาษา HTML ที่เรารู้จักกันดีเท่านั้น แต่ว่าวันนี้โลกได้มีการพัฒนาเทคโนโลยีใหม่ที่ใช้สำหรับการเขียนเว็บ นั่นคือ XML ซึ่งเป็นสิ่งที่หลายผลิตภัณฑ์ให้การสนับสนุน

Extensive Markup Language (XML) เป็นภาษาที่ให้ความชัดเจนในการให้รายละเอียดเกี่ยวกับข้อมูล และการเปลี่ยนแปลงข้อมูลโดยแอปพลิเคชันบนเว็บและใช้ฟอร์มที่ยืดหยุ่นได้ตามมาตรฐาน HTML หรือ Hyper Text Markup Language ได้เปิดโลกแห่งการแสดงข้อมูลต่างๆ มานำเสนอ ส่วน XML จะทำให้การทำงานกับข้อมูลโดยตรงที่เสริมกับการทำงานของ HTML

#### 2.2.1.1 ความเป็นมาของ XML (Extensive Markup Language)

โพรโตคอลอินเทอร์เน็ต (อินเทอร์เน็ต Protocol-IP), HyperText Markup Language และ HyperText Transport Protocol (HTTP) ได้เป็นการปฏิวัติและสร้างมิติใหม่ในการกระจายข้อมูลและสารสนเทศ การนำเสนอ ตลอดจนการค้นคืน โดยให้ผู้ใช้สามารถใช้สารสนเทศที่ต้องการได้ง่ายด้วยบราวเซอร์ และมี search engine หรือเครื่องมือในการช่วยค้นหา นอกจากนี้ยังมีการประยุกต์ไปใช้กับเครือข่ายในสำนักงานหรืออินเทอร์เน็ต และใช้สำหรับการบริการข้อมูลสำหรับลูกค้าและลูกค้าให้สามารถตอบสนองทางด้านสารสนเทศที่ต้องการ ได้อย่างมีประสิทธิภาพมากขึ้น

สำหรับ Extensive Markup Language จะให้ประโยชน์อย่างเต็มที่เมื่อทำงานร่วมกับ HTML ด้วยเหตุที่ว่า XML ได้มีความพร้อมในแง่ของรายละเอียด และการนำข้อมูลตลอดจนโครงสร้างข้อมูลมาแสดงได้ในรูปแบบ Text ผ่านทาง HTTP ที่เปิดให้ข้อมูลขึ้นใหม่และมีความสามารถในการจัดข้อมูลได้อีกด้วย ในการเขียนเว็บเพจเมื่อใช้ HTML ผู้พัฒนาสามารถกำหนดได้ว่าส่วนไหนจะเป็นตัวหนา ตัวเอียง หรือตัวอักษรเป็นแบบไหน ส่วน XML นั้นจะเป็นการเตรียมส่วนของข้อมูลที่จะนำไปใส่ในช่องที่กำหนดตามการเขียนของ HTML ยกตัวอย่างเช่น ข้อมูลด้านราคา หรือราคาที่ตั้งสำหรับการจัดรายการส่งเสริมการขาย อัตราภาษี ค่าขนส่ง เป็นต้น

XML ถือได้ว่าเป็นส่วนหนึ่งของ Standard Generalized Language Markup Language (SGML) ที่เป็นข้อกำหนดในการสร้างหรือจัดทำเอกสารในรูปแบบอิเล็กทรอนิกส์ที่กำหนดโดย W3C หรือ World Wide Web Consortium สามารถดูรายละเอียดเพิ่มเติมได้จาก <http://www.w3.org/TR/REC-xml> ที่มีโครงสร้างและรูปแบบที่เปิดให้แอปพลิเคชันต่างๆ สามารถเรียกไปใช้งานได้ เช่น บนเว็บไซต์ต่างๆ เป็นต้น และทางไมโครซอฟท์ได้มีการทำงานร่วมกับ W3C เพื่อพัฒนามาตรฐานข้อมูลบนเว็บที่ให้ HTML

สามารถแสดงข้อมูลที่ XML ได้เตรียมไว้ และทางไมโครซอฟท์เองได้มีการเปิดตัว บราวเซอร์ตั้งแต่ IE 4.0 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นต้นไป ที่สามารถเรียกดูและประมวลผลข้อมูลได้ และเป็นข้อกำหนดให้ บราวเซอร์เวอร์ชันใหม่ของ กายโมโครซอฟท์สนับสนุน XML

สิ่งที่ถือได้ว่าเป็นลักษณะเด่นของ XML นั้นจะเป็นความสะดวกในการจัดการด้านระบบการ ติดต่อกับผู้ใช้จากโครงสร้างของข้อมูล เราสามารถนำข้อมูลจากหลายแหล่งมาแสดงผลและประมวลผล ร่วมกันได้ ไม่ว่าจะเป็นข้อมูลลูกค้า รายการสั่งซื้อ ผลการวิจัย รายการรับชำระเงินข้อมูลเวชระเบียน รายการสินค้าหรือข้อมูลสารสนเทศอื่นๆ ก็สามารถแปลงให้เป็น XML ได้ และในส่วนของข้อมูลสามารถ ปรับให้เป็น HTML ได้

สำหรับประโยชน์ในการใช้งานนั้น เราจะสามารถนำมาใช้สำหรับการเข้าถึงระบบข้อมูลขนาดใหญ่ ใช้กับระบบเครือข่ายในองค์กร หรืออินเทอร์เน็ตเพื่อดูข้อมูลหรือเรียกใช้ข้อมูลที่ให้การแสดงผลทาง หน้าจอที่รวดเร็วและง่ายในการจัดการ

### XML (Extensive Markup Language) คืออะไร

Extensive Markup Language เป็นฟอร์แมตหรืออธิบายถึงรายละเอียดของโครงสร้างและแบบของ ข้อมูลเป็นภาษาหรือชุดคำสั่งเกี่ยวกับข้อมูลบนเว็บ ที่ให้การพัฒนาและมีศักยภาพในส่วนของโครงสร้าง ข้อมูลจากหลากหลายแอปพลิเคชันมานำเสนอบนเครื่องเดสก์ทอป ด้วย XML จะทำให้การจัดการข้อมูล หรือเรียกใช้ข้อมูลจากแอปพลิเคชันต่างๆ จะเข้าสู่มาตรฐานเดียวกัน

XML จะเป็นส่วนหนึ่งของ HTML ซึ่ง XML จะให้รายละเอียดเกี่ยวกับข้อมูล เช่น ชื่อเมือง อุณหภูมิ ความกดอากาศ ส่วน HTML เป็นการกำหนด tag ต่างๆ ที่จะทำให้ข้อมูลแสดงออกมาในรูปแบบ ใดหนึ่ง ซึ่งข้อมูลจะสามารถแสดงออกมาได้หลายรูปแบบ ไม่ว่าจะเป็นตารางหรือข้อความธรรมดา ขึ้นอยู่กับ การกำหนดของ HTML และในปัจจุบันนี้ ด้วย XML จะมีการให้รายละเอียดของเนื้อหาเอกสารที่เรียกว่า Document Type Definition (DTD) ที่ให้รายละเอียดเกี่ยวกับตัวเอกสารว่าจะแสดงหรือซ่อนส่วนไหนของ เอกสารบ้าง ซึ่ง DTD จะเป็นส่วนที่เพิ่มเติมสำหรับ XML ถ้าหากมีการส่งข้อมูลในรูปแบบ DTD ก็จะได้รู้กัน ว่าเป็น XML มีความหมายหลาย ๆ คำที่อธิบายลักษณะของ XML

Richard Baldwin นิยามความหมายของ XML ไว้ดังนี้

"XML ทำให้ผู้ใช้สามารถสร้างและดูแล structured documents (เอกสารที่มีโครงสร้าง) ที่บรรจุ plain text (ตัวอักษร) โดยทำให้สามารถ rendered หรือปรับเปลี่ยนการแสดงผลในรูปแบบที่หลากหลาย จุดประสงค์หลักของ XML คือการแยกส่วน ข้อมูลเพื่อประโยชน์ในการแสดงผล"

ตามข้อความด้านบน สามารถอธิบายความหมายของคำศัพท์ ได้ดังนี้

- Structured document

พิจารณาจากตัวอย่าง โครงสร้างของ หนังสือ ต่อไปนี้

o หนังสือหนึ่งเล่มประกอบด้วยเนื้อหาแต่ละบท (chapter)

o ในแต่ละบทประกอบด้วยหัวข้อย่อย (section)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- o ในแต่ละหัวข้อย่อ อาจจะถูกอธิบายหรือมีตารางข้อมูล (table) บรรจอยู่
- o ตารางข้อมูลถูกสร้างขึ้นมาจากแถว (row) และคอลัมน์ (column)

ดังนั้น จะเห็นว่าหนังสือแต่ละเล่ม มักจะมีรูปแบบ หรือโครงสร้างที่แน่นอน ซึ่งทุกคนที่หยิบหนังสือขึ้นมาอ่าน ก็จะเข้าใจโครงสร้างของหนังสือ ตามที่ได้อธิบายไว้ข้างต้น

#### - Plain text

ตัวอักษร(Character) ที่เห็นในจดหมายหรือบนกระดาษทั่วไป เป็นลักษณะตัวอักษรประเภทที่พิมพ์ได้ (alphabet and punctuation) เช่นที่เราใช้พิมพ์ในเอกสาร ในเวิร์ดหรือ โปรแกรมพิมพ์รายงานทั่วไป ซึ่งรวมถึงตัวเลขด้วย ถือว่าเป็น plain text ทั้งนี้และทั้งนั้น ผู้อ่านอาจจะสงสัยว่า แล้วตัวอักษรอะไรพิมพ์ไม่ได้หรือ ลองมองดูที่คีย์บอร์ดของท่าน แล้วดูที่เป็น Shift, Ctrl, Alt, Insert, Delete, ... เหล่านี้เป็นต้น

ASCII คือที่มาของรหัสตัวอักษร: รูปแบบที่ใช้แทนตัวอักษรที่นิยมใช้กันมากที่สุด เราเรียกว่า (ASCII) โดยจะแทนตัวอักษรด้วยรหัสหรือค่าตัวเลข เช่น ตัวอักษร "A" ถูกแทนด้วยค่า 65 ในขณะที่ตัวอักษร "B" แทนด้วยค่า 66 และ "C" คือ 67 ตามลำดับ รหัส ASCII นี้เอง ซึ่งถูกนำมาใช้งานในการแสดงข้อมูล รูปแบบเอกสาร ที่สามารถพิมพ์ให้ผู้ใช้ได้เห็นในขณะที่การบันทึกบนระบบคอมพิวเตอร์ จะใช้ในการเก็บเป็นตัวเลขแทน ASCII เป็นคำย่อที่มาจาก American Standards Committee on Information Interchange โดยเป็นองค์กรที่จะตั้งขึ้นมาดูแล รหัสคำสั่งต่างๆเหล่านี้ แต่ในความเป็นจริงนอกจาก ASCII จะเก็บบันทึกตัวอักษรแล้ว ยังมีรหัสที่แทนตัวอักษร ต่อไปเป็นความหมายที่ทำให้เข้าใจความเกี่ยวข้องระหว่างรหัส ASCII กับ Plain text

"มาตรฐานที่เกิดจาก American Standards Committee on Information Interchange คือแนวทางที่ถูกใช้ในการกำหนดลักษณะของ plain text กล่าวคือตัวอักษรที่สามารถอ่านได้ โดยไม่ต้องใช้ความสามารถ ของโปรแกรมใดมาช่วยในการอ่านตัวอักษรเหล่านั้น ข้อพิเศษของ ASCII อีกอย่างหนึ่งคือ เป็นข้อมูลที่สามารถ นำไปแสดงได้ในทุกๆแพลตฟอร์ม แต่ก็ยังมีข้อเสียที่ เป็นรูปแบบที่น่าเบื่อ (ไม่สวยงาม) ไม่สามารถแสดงในรูปแบบตัวหนา ตัวเอียง หรือขีดเส้นใต้ และไม่มีรูปแบบที่ดูเป็นกราฟฟิก แต่อย่างไรก็ตาม ถือว่าเป็นแบบในการแสดงตัวอักษรที่ง่ายที่สุด และเข้าใจได้ในทุก ๆ ระบบ"

#### ข้อมูลดิบของเอกสาร XML (Raw-XML)

เอกสาร XML ที่ถือเป็นข้อมูลดิบ คือข้อความที่เกิดจากการรวมกันของตัวอักษร เพื่อทำให้เกิดเป็นเอกสาร ก่อนที่จะนำเอกสารที่เกิดจากข้อมูลดิบ มาใช้แสดงผลตามวัตถุประสงค์ของการใช้งานอีกที ซึ่งเป็นรูปแบบที่น่าสนใจมากกว่า plain text

#### ความหมายของการปรับการแสดงผล (Rendering)

เป็นศัพท์สำหรับคอมพิวเตอร์สมัยใหม่ rendering คือรูปแบบของการนำข้อมูลดิบ หรือ plain text มาปรับเปลี่ยนใหม่ ให้อ่านน่าสนใจยิ่งขึ้น สำหรับการมองของมนุษย์ Rendering เป็นการวาดภาพ (Drawing)

ตัวอย่างของการวาดภาพในระบบคอมพิวเตอร์ เป็นการนำเอาสมการ ทางคณิตศาสตร์ มาใช้ร่วมกับตัวเลข เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆ เช่น สมการวงกลม กับค่าตัวเลขหนึ่งให้วงกลมขนาดหนึ่ง กับอีกตัวเลข ให้วงกลมในอีกหนึ่งขนาด ซึ่งตัวเลขและสมการเหล่านี้ เป็นสิ่งที่มนุษย์ หรือผู้ใช้งานไม่อาจได้เห็นเลย แต่จะเห็นเป็นรูปร่างที่เกิดขึ้นแทน

Rendering เอกสาร: เมื่อพูดถึงขบวนการ rendering ของเอกสาร มีความหมายว่าเป็นการแสดงข้อมูลในรูปแบบเอกสาร ที่มนุษย์เข้าใจ เช่นหนังสือ หนังสือพิมพ์ หรือเอกสารแบบอื่นๆ ที่มนุษย์เข้าใจและอ่านได้ Rendering ในที่นี้ หมายถึงการแสดงข้อมูลในรูปแบบ อักษรตัวหนา, อักษรตัวเอียง, อักษรขีดเส้นใต้, อักษรที่มีสี, เป็นต้น

แยกการแสดงผลด้วยเนื้อหา: ข้อมูลที่สร้างขึ้นเป็น XML ไม่มีความสามารถ หรือคุณลักษณะพิเศษอะไร ที่ทำให้การแสดงผลออกมาในรูปแบบที่น่าสนใจ เช่น ตัวหนา ตัวเอียง ตัวขีดเส้นใต้ ดังที่กล่าวผ่านมาแล้ว ให้จำแต่เพียงว่า XML ทำหน้าที่แยกข้อมูลออกเป็นเนื้อหา (Content) เท่านั้น ตามลักษณะเอกสารที่มีโครงสร้าง ส่วนหน้าที่การนำเอาเนื้อหาใน XML มาแสดงในรูปแบบที่น่าสนใจ เป็นหน้าที่การทำงานของส่วนอื่น

#### 2.2.1.2 ลักษณะโครงสร้างของ XML

XML เป็นการใช้ข้อความเพื่อแบ่งออกโครงสร้างของเอกสาร พิจารณาตัวอย่างรูปแบบโครงสร้างของหนังสือ เมื่อหนังสือประกอบด้วยจำนวนบท 2 บท ในแต่ละบทประกอบด้วยเนื้อความ (Text)

Begin Book

Begin Chapter 1

Text for Chapter 1

End Chapter 1

Begin Chapter 2

Text for Chapter 2

End Chapter 2

End Book

หนังสือที่มีอยู่ในปัจจุบัน จะมีโครงสร้างที่มีรายละเอียดที่ซับซ้อนมากกว่านี้ เช่น บทนำ, สารบัญ, เป็นต้นเช่นเดียวกัน ภายในส่วนเนื้อความ (Text) ยังประกอบด้วยโครงสร้างย่อย คือ ย่อหน้า (Paragraph) แต่ละย่อหน้ายังประกอบขึ้นจาก ประโยค คำ และตัวอักษรด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของเอกสาร XML นั้น สามารถอธิบายโดยใช้ตัวอย่างที่ 1 ได้ ดังนี้  
ตัวอย่างที่ 1

```
<?xmlversion="1.0"encoding="windows-874"?>
<mali>
<malisorn>ขึ้นต้นด้วยมะลิซ้อน</malisorn>
<malila>พอแตกใบอ่อนเป็นมะลิลา</malila>
</mali>
```

```
<?xmlstylesheet type="text/xsl" href=""?>
```

บรรทัดที่ 1 นั้นหมายความว่าเราประกาศเอกสารนี้เป็นเอกสาร XML และมีการเข้ารหัสอักขระเป็น windows-874 เพื่อให้ใช้ภาษาไทยได้ จากนั้นเราจะเห็น สิ่งที่เราคุ้นเคยที่เราเรียกว่า tag แต่จริงๆแล้วใน ภาษา XML จะแบ่ง โครงสร้างเป็น 2 ส่วนใหญ่ ๆ คือ tag & element สามารถอธิบายเพิ่มเติมได้จาก ตัวอย่างที่ 2

ตัวอย่างที่ 2

```
<root>
  <element>
    <tag></tag>
  </element>
</root>
```

ความหมายของ Tag กับ Element Tag

สำหรับใน XML แล้ว tag มีความหมายในลักษณะเดียวกับที่ใช้ใน HTML tag คือข้อความที่อยู่ระหว่าง สัญลักษณ์ < และ >

- Tag เปิด (Start tag) <book>

จากตัวอย่างที่แสดง ด้านบนถูกเรียกว่า tag เปิด ดังนั้น tag เปิดจึงมีสัญลักษณ์คือ <...>

- Tag ปิด (End Tag)

</book>

tag ที่ถูกเรียกว่า tag ปิด ต่อเมื่อใน tag มีเครื่องหมาย / อยู่หลังสัญลักษณ์ < ดังนั้นลักษณะของ tag ปิดจึง มีรูปแบบคือ </...> หากพิจารณาระหว่าง tag เปิดกับ tag ปิดแล้ว ข้อแตกต่างอีกข้อหนึ่งคือ tag เปิด เป็น tag ที่สามารถใส่ข้อมูล attribute ลงไปภายใน tag ได้ แต่ tag ปิดจะไม่ทำกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Element

ในที่นี้คือ โครงสร้างหลักของ XML ซึ่งอยู่ในรูปของ tag เช่นเดียวกัน ตามตัวอย่างข้างบน element คือ

```
<root>
  <element>

</element>
</root>
```

จะมีลักษณะซ้อนกันเป็นชั้นๆ

- Root element เป็น element แรกสุดของเอกสาร XML

element เป็นส่วนประกอบของ Content และ attribute เพื่อให้ผู้อ่านได้เข้าใจความหมายของศัพท์หลายๆ คำศัพท์ ก่อนที่จะนำไปใช้ในการสร้าง XML ซึ่งจำเป็นอย่างมาก ลองมาพิจารณา คำจำกัดความของ element กันก่อน

```
<chap number="1">Text for Chapter 1</chap>
```

สังเกตจากลักษณะ รูปทางด้านบน ทั้งหมดตั้งแต่ <chap ..... จนถึง </chap> ถูกเรียกว่า element หรือถ้ามองง่ายๆคือ element เริ่มต้นที่ tag เปิด และสิ้นสุดที่ tag ปิดใน tag คำสั่งเดียวกัน เพื่อให้แยกแยะแต่ละสิ่งที่ให้ไว้ในตัวอย่าง แสดงความหมายที่เกี่ยวข้องกับ XML ได้ดีทีเดียว

## Content

เนื้อความ หรือ Content ถือได้ว่าเป็นข้อมูลเพื่อใช้ในการแสดง ให้ผู้อ่านเอกสารได้เห็น Content คือ ตัวอักษรสีเขียว จากตัวอย่าง หรือกล่าวอีกนัยหนึ่งคือ Content อยู่หลัง tag เปิด และจบที่ก่อนถึง tag ปิดนั่นเอง

## Attribute

Attribute คือตัวอักษรสีน้ำเงิน จากตัวอย่าง และจะเห็นว่า ถูกบรรจุอยู่ภายใน tag เปิด และ attribute นี้ในบางครั้ง อาจจะมีหรือไม่มีก็ได้ รวมถึงถ้ามีอาจมีได้มากกว่าหนึ่งตัวใน tag, attribute คือข้อมูลความหมายเพิ่มเติม แน่นนอน ถ้ากล่าวถึงสิ่งของ หากเราไม่ได้ให้ความหมายเพิ่มเติมเราก็จะไม่รู้ และแยกแยะสิ่งที่เหมือนกันออกจากกันได้ เช่น ถ้าพูดถึง บท ในหนังสือ แค่นี้เป็นความหมายโดยรวม แต่ถ้าบอกว่า บทที่ 1 ในหนังสือ เลข 1 ในที่นี้คือความหมายเพิ่มเติมให้กับบท

หรืออีกซีกตัวอย่าง ถ้ากล่าวถึง แอปเปิ้ล อย่างเดียว คนทั่วไปจะนึกถึงผลไม้ที่ทานได้ แต่ถ้าบอกแอปเปิ้ล สี เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาทานาน ไม่อนุญาตให้นำไปเผยแพร่จนดานการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แดง รสเปรี้ยว นั่นคือการบอกถึง attribute 2 ตัวคือ สีแดง และ รสเปรี้ยว ทำให้ทราบว่าเป็นแอปเปิ้ลที่ไม่มีใครอยากได้มาทาน (ยกเว้นคนที่ชอบของเปรี้ยวๆ)

### 2.2.1.3 วัตถุประสงค์หลักของ XML

ความหมายของ XML ก่อนหน้านี้ ได้บอกเอาไว้ว่า" จุดประสงค์หลักของ XML คือการแยกส่วนข้อมูลเพื่อประโยชน์ในการแสดงผล "นั่นคือ เอกสาร XML ใช้สำหรับควบคุม ตัวอักษร (plain text) โดยยึดเอาข้อมูลที่มีใจความเหมือนกัน แต่สามารถนำไปแสดงผล ให้ผู้ใช้หรือผู้อ่านหลายคนได้รับรู้ข้อมูลใจความเดียวกัน เมื่อมองผ่านอุปกรณ์แสดงผลที่ต่างกัน ซึ่งลักษณะเหล่านี้ ไม่เกิดขึ้นจริงกับ เอกสารประเภทประมวลผลคำอื่นๆ word processing หลากหลายคอมพิวเตอร์และหลากหลายระบบปฏิบัติการ นี้ไม่ใช่สิ่งที่ควรจะละเลยได้เลย เห็นได้ชัดว่าหลายๆปีที่ผ่านมา คอมพิวเตอร์หลากหลายชนิดถูกผลิตออกมาเป็นจำนวนมาก เพื่อใช้งาน รวมถึงระบบปฏิบัติการ (Operation System) ก็ยังมีความแตกต่างกันออกไปด้วย นั่นคือคอมพิวเตอร์รุ่นใหม่ๆจึงมีความหลากหลาย และยุ่งยากในการทำให้เข้าใจข้อมูลเดียวกัน ซึ่งเปรียบแล้วเหมือนกับภาษามนุษย์ที่ใช้สื่อสารกัน มีมากมายหลายร้อยภาษา W3C ได้กล่าวถึงจุดมุ่งหมายหลักไว้ 10 หัวข้อ ดังนี้

1. XML มีการใช้งานโดยตรงบนเครือข่าย อินเทอร์เน็ต XML จะถูกออกแบบมาสำหรับจัดเก็บ และจัดส่งข้อมูลบนเว็บ
2. XML มีการสนับสนุนโปรแกรมที่หลากหลาย ถึงแม้ว่าวัตถุประสงค์ที่สำคัญคือ การจัดส่งข้อมูลบนเว็บผ่านทางเซิร์ฟเวอร์และ โปรแกรมเบราว์เซอร์ XML จะถูกออกแบบมาเพื่อใช้กับโปรแกรมที่มีรูปแบบต่าง ๆ ตัวอย่างเช่น การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมทางการเงิน การเผยแพร่และปรับปรุงโปรแกรมให้ทันสมัยและการเขียน Voice Script ให้สื่อสารได้ด้วยโทรศัพท์
3. XML จะต้องเข้ากันได้กับ SGML
4. XML จะต้องง่ายต่อการเขียนโปรแกรมเพื่อประมวลผลเอกสาร
5. จำนวนของทางเลือกเฉพาะของ XML ควรมีจำนวนน้อยที่สุดหรือไม่ควรมีเลย
6. เอกสาร XML จะต้องอ่านเข้าใจง่ายและมีความชัดเจน
7. XML ออกแบบมาเพื่อให้พัฒนาโปรแกรมได้อย่างรวดเร็ว
8. การออกแบบ XML ต้องมีรูปแบบที่เหมาะสมและกะทัดรัด
9. สามารถสร้างเอกสาร XML ได้ง่าย
10. Markup ของ XML ต้องไม่รบกวนมากเกินไป

ภาษาสำหรับจัดการ โครงสร้างเอกสาร

XML พยายามลดข้อจำกัดของความแตกต่าง ทางด้านระบบคอมพิวเตอร์ เพื่อสร้างเป็นภาษาที่ใช้กำหนดโครงสร้างของเอกสาร ให้สามารถเข้าใจกันได้ในทุก ๆ ระบบ XML คือภาษาที่มีลักษณะเมตา (meta language)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมตา หมายถึงรูปแบบการรวบรวมข้อมูลข่าวสาร ที่นำมาจากแหล่งข้อมูลข่าวสารอื่นๆ เช่น บทอ้างอิงในหนังสือสามารถบอกผู้อ่านได้ ถ้าถูกรวบรวมเนื้อหาไว้ด้วย XML ผู้อ่านจะสามารถรู้ว่าหัวข้อที่ตัวเองสนใจ จะไปอยู่ที่หน้าที่เท่าไรของหนังสือ ซึ่งเมื่อเปิดไปยังหน้าดังกล่าวแล้ว จะได้รับข้อมูลที่อ้างไปถึงจากหัวข้อในบทอ้างอิงนั่นเอง ดังนั้นแต่ละหัวข้อในบทอ้างอิงจึงเป็น ข้อมูล ที่ได้รวบรวมมาจากข้อมูลในเนื้อหา บทอ้างอิงจะสามารถบอกข้อมูลเกี่ยวกับ element (elements) และแอททริบิวต์ (attributes) ซึ่งจะมีเนื้อหาต่อไปได้ เราจึงใช้ XML เป็นภาษามเมตา

#### 2.2.1.4 การประยุกต์ใช้ XML

ตัวอย่างการประยุกต์ใช้งาน XML ตัวอย่างแรก ของเอกสาร XML:

```
<book>
<chap>
Text for Chapter 1
</chap>
<chap>
Text for Chapter 2
</chap>
</book>
```

จากรูปเป็น ไวยากรณ์ที่ถูกใช้ใน XML เมื่อเปรียบเทียบกับตัวอย่างนี้กับเอกสารที่เห็นในส่วนต้นของบทความสำหรับโครงสร้างของหนังสือ โดยเปรียบเทียบทีละส่วน โดยพิจารณาที่ส่วนของ element (สังเกตกลุ่ม element จากสัญลักษณ์ <...> จนถึง </...>) ดังนั้นโครงสร้างของ XML จะมี element เป็นส่วนประกอบ ใช้กำหนดโครงสร้างของเอกสาร

ตัวอย่างถัดไปด้านล่าง เป็นการปรับปรุงตัวอย่าง XML ก่อนหน้าที่มี element อยู่ โดยเพิ่มส่วนที่เรียกว่า attribute ลงใน element คือตัวเลขบอกเลขที่ของบท โดยชื่อ attribute ใช้คำว่า number และ ตามด้วยตัวเลขบอกความหมายของบท

```
<book>
<chap number="1">
Text for Chapter 1
</chap>
<chap number="2">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา หรืออ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text for Chapter 2
</chap>
</book>

```

ตัวอย่างของคำสั่ง สำหรับการกำหนด โครงสร้างสำหรับหนังสือ

```

<book>
<chap number="1">
Text for Chapter 1
</chap>
<chap number="2">
Text for Chapter 2
</chap>
</book>

```

จากตัวอย่างโครงสร้าง เมื่อพิจารณาจะเห็นได้ว่า มีรูปแบบของจำนวนบทอยู่ 2 บท ในแต่ละบทมีข้อความอยู่บางส่วน เช่น บทที่ 1 มีข้อความ Text for Chapter 1 ที่อยู่ระหว่าง element chap จะมี attribute ชื่อ number โดยบทที่ 1 ใช้ number="1" เป็นข้อมูลของ attribute

จากโครงสร้างเอกสาร XML ที่เราได้กล่าวถึงในส่วนต้น แนนอนสามารถนำไปผ่านขบวนการ render เพื่อให้ได้เอกสารใช้งานจริงๆ ได้ในหลากหลายรูปแบบ ข้อมูลเอทริบิวต์ (attribute) มีความสำคัญ ข้อมูลอยู่ใน attribute number มีความหมายที่จะสามารถทำให้กลไก render สามารถแยกแยะเอกสารตามรายละเอียด ของเนื้อหาในแต่ละบทได้ ดังนั้นหากต้องการให้มุมมองของการมองเห็นแตกต่างกัน ก็ย่อมทำได้เช่นเดียวกัน ยกตัวอย่างเช่น หากนำเอกสาร XML ผ่านขบวนการ render สำหรับหนังสือที่เป็นกระดาษ การจัดพิมพ์ย่อจะเกิดหน้าว่างหนึ่งหน้า ของแต่ละบทเสมอ ในขณะที่เอกสารตัวเดียวกัน หากนำไปผ่านขบวนการ render ของมอนิเตอร์แล้ว ก็ไม่จำเป็นต้องทำให้เกิดหน้าว่างก่อนบท

เอกสาร XML เพียงบอกแค่ ข้อมูลในส่วนที่เป็น Content ซึ่งถูกแยกแยะข้อมูลออกเป็นส่วนๆ ในลักษณะเอกสารที่มีโครงสร้าง หรือกล่าวอีกนัยหนึ่งคือ XML ใช้สำหรับแยกแยะเอกสารออกเป็นส่วนๆ เท่านั้น โดยที่ข้อมูลในเอกสาร ไม่มีส่วนที่ใช้บอกว่าควรจะแสดง Content ในลักษณะการมองเห็นแบบไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1.5 การใช้งานเอกสาร XML

ในอนาคตคาดว่า รูปแบบการ rendering มีการพัฒนาไปอย่างมาก ดังจะเห็นได้จาก บริษัทยักษ์ใหญ่ ค่ายต่าง ๆ ทางด้านไอที และคอมพิวเตอร์ ได้ร่วมมือกัน และพัฒนาโครงสร้าง หรือกลไกการ rendering ไว้ในแผนการของตัวเองแล้ว หากเราสามารถนำข้อมูลของเราเอง มาเก็บไว้ในรูปแบบของ XML จะทำให้ข้อมูลของเรา มีคุณค่าในอนาคต และสอดคล้องกับเทคโนโลยีที่จะมีมาด้วย

- XML จะเกิดความสะดวกในระบบพาณิชย์อิเล็กทรอนิกส์

ตัวอย่างหนึ่งคือเทคโนโลยี XML และ Java ได้มีการกำหนดข้อตกลงต่างๆ เพื่อพัฒนาขบวนการค้า หรือ E-Commerce ในการประชุม Software Development East ที่ผ่านมา และพวกเขาเชื่อมั่นว่า ในระบบการค้าอิเล็กทรอนิกส์ ข้อมูลจะถูกเก็บอยู่ในรูปแบบของเอกสาร XML ทั้งหมด อย่างเร็วที่สุดในปลายปี 2001

- XML ทำให้ผู้ค้าและผู้ใช้มีความอิสระ

ผู้พัฒนาเทคโนโลยี มีความอิสระในการเพิ่มศักยภาพในผลิตภัณฑ์ของตัวเอง ไม่จำเป็นต้องมากังวลรูปแบบการสื่อสาร ที่จะต้องออกแบบมาเฉพาะ ทำให้ลูกค้าต้องยึดติดกับผลิตภัณฑ์รายใดรายหนึ่ง เมื่อข้อมูลอยู่ในเอกสาร XML แล้ว ผลิตภัณฑ์ต่างๆ ก็จะมีรูปแบบข้อมูลที่สามารถแลกเปลี่ยนกันได้ โดยที่ไม่ต้องไปกังวลกับเทคโนโลยี ที่ไม่สอดคล้องกันในปัจจุบันได้

- XML ลดค่าใช้จ่าย

แน่นอน ผลที่ตามมาคือ เทคโนโลยีที่แลกเปลี่ยนข้อมูลกันได้ ทำให้ค่าใช้จ่ายที่จะต้องสูญเสียไป ในปัจจุบันสำหรับการสื่อสารข้อมูล ที่มีรูปแบบที่หลากหลายถูกขจัดออกไป นั่นเป็นผลดีสำหรับบุคลากรสื่อสารด้วย XML

### 2.2.1.6 XSLT และ Xpath

XSLT เป็นภาษาที่สามารถแปลงเอกสาร XML ให้เป็นเอกสารแบบเท็กซ์ เป็น XML เองหรือเป็นฟอร์แมตอื่นๆ ได้ โดย XSLT เป็นส่วนประกอบย่อยของภาษาที่ใหญ่กว่าที่เรียกว่า XSL ซึ่ง XSLT เป็นเครื่องมือที่มีศักยภาพสำหรับการค้าอีคอมเมิร์ซและที่อื่นใดที่ XML จะถูกนำมาใช้ได้

ยกตัวอย่างเช่น ถ้าเรากำลังรันเว็บไซต์เกี่ยวกับข่าวที่นอกเหนือจากเว็บไซต์ของเราเองแล้ว เรายังต้องการให้ผู้อ่านมีหนทางที่จะได้รับข่าวของเราในรูปแบบ XML ด้วย หรืออีกนัยหนึ่งก็คือ คนทั่วไปสามารถนำข่าวที่เราจัดหาไว้ไปใช้กับแอปพลิเคชันของพวกเขาเองได้ ไม่ว่าจะด้วยวัตถุประสงค์อะไรก็ตาม อย่างเช่น บางคนอาจจะต้องการแสดงเรื่องราวดังกล่าวบนเว็บไซต์ของพวกเขาเอง แต่คนอื่นๆ ก็อาจจะต้องการใส่ข้อมูลที่ได้เข้าไปในฐานข้อมูลของพวกเขาก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี่เป็นที่มาของ XSLT กล่าวคือ แทนที่ที่คนอื่นได้รับเอกสาร XML ของเราไป เขาสามารถที่จะใช้ ภายในการแปลงรูปแบบเพื่อแปลงเอกสารเหล่านี้ไปเป็นรูปแบบอะไรก็ตามที่พวกเขาต้องการได้ ไม่ว่าจะ เป็น HTML เพื่อนำไปแสดงผลบนเว็บไซต์ เป็น XML ที่มีโครงสร้างต่างไปจากเดิมเพื่อใช้กับ แอปพลิเคชันอื่นๆ หรือแม้แต่เป็นเท็กซ์ไฟล์ธรรมดาาก็ได้

### 2.2.1.6.1 XSLT

Extensible Style sheet Language for Transformation หรือ XSLT คือภาษาที่สามารถแปลง เอกสาร XML ให้เป็นเอกสารแบบ เท็กซ์, เป็น XML เองหรือเป็นฟอร์แมตอื่นๆ ได้

XSLT เป็นส่วนประกอบย่อยของภาษาที่ใหญ่กว่าที่เรียกว่า XSL ซึ่ง XSLT เป็นเครื่องมือที่มี ศักยภาพสำหรับการค้าอีคอมเมิร์ซและที่อื่นใดที่ XML จะถูกนำมาใช้ได้ XSLT จะทำงานด้วยการค้นหา ส่วนของเอกสาร XML ที่ตรงกับชุดของเท็มเพลตที่กำหนดเอาไว้แล้ว หลังจากนั้นก็จะประยุกต์กฎในการ แปลงข้อมูล และการจัดรูปแบบเข้ากับส่วนที่ค้นพบภาษาอื่นๆ เช่น XPath ถูกนำมาใช้เพื่อช่วยในการ ค้นหาส่วนที่เหมาะสม

### 2.2.1.6.2 XSL คืออะไร

Extensible Style sheet Language มีความหมายตามชื่อของมัน นั่นคือ ภาษาที่มีลักษณะแบบ XML (XML-based) ที่ใช้ในการสร้างสไตล์ชีต กลไกของ XSL ใช้สไตล์ชีตเพื่อแปลงเอกสารเพื่อแปลง เอกสาร XML ทั้งหมดกลายเป็นเอกสารชนิดอื่นๆ และเพื่อจัดรูปแบบของเอาต์พุตเหล่านี้ คุณจะกำหนดเลย์ เอาต์ของเอกสารเอาไว้ รวมทั้งที่หาค่าข้อมูลมาใช้ภายในเอกสารอื่นชุด ก็เหมือนกับกรพูดว่า “ข้อมูลของ คุณอยู่ ณ จุดในเอกสารอินพุต ให้ทำให้เป็นแบบนี้ในเอกสารเอาต์พุตนะ” นั่นเอง ภาษาเฉพาะของ XSL จะเรียกเอกสารอินพุตว่าต้นไม้ต้นฉบับ (source tree) ส่วนเอกสารเอาต์พุตเรียกว่า ต้นไม้ผลลัพธ์ (result tree)

จริงๆ แล้วมีภาษาที่สมบูรณ์แล้วอยู่ 2 ภาษาภายใต้ร่มเงาของ XSL ได้แก่ :

- ภาษาสำหรับแปลงข้อมูลที่มีชื่อว่า XSLT
- ภาษาที่ใช้ในการอธิบายเอกสาร XML เพื่อการแสดงผล นั่นคือ XSL Formatting Object

เป็นที่แน่นอนว่าทั้งสองภาษานี้สามารถใช้เพื่อการแปลงข้อมูลของ XSLT ร่วมกันได้ และ ออบเจกต์ในการจัดรูปแบบของ XSL จะทำการปรับปรุงข้อมูลเพื่อการแสดงผลต่อไป

นอกจาก XML ที่ลงตัวแล้ว XSLT ยังให้ผลลัพธ์ที่เป็น HTML หรือแม้แต่เท็กซ์ธรรมดาได้ด้วย ในความเป็นจริงนั้น หนึ่งในการใช้งานที่เน้นมากๆ สำหรับ XSLT ก็คือ ใช้เพื่อแปลงเอกสาร XML ไป เป็นเอกสาร HTML เพื่อแสดงในบราวเซอร์

### 2.2.1.6.3 สไตล์ชีต XSLT ทำงานอย่างไร

XSLT สไตล์ชีตถูกสร้างขึ้นมาจากโครงสร้างที่เรียกว่า เท็มเพลต (Template) เท็มเพลตจะเป็น

ตัวกำหนดสิ่งที่ต้องค้นหาจากต้นไม้ต้นฉบับ และสิ่งที่ต้องใส่เข้าไปในต้นไม้ผลลัพธ์

เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XSLT เขียนขึ้นด้วย XML หมายความว่า มีอิลิเมนต์และ แอททริบิวต์พิเศษของ XSLT ที่คุณใช้ เพื่อสร้าง สไตลชีตของคุณเองได้ดังตัวอย่างต่อไปนี้ :

```
<xsl : template match="first"> My Project </xsl : template>
```

ใน XSLT เพิ่มเพลดูกำหนดขึ้นโดยใช้อิลิเมนต์ <xsl : template> ที่มี 2 ส่วนสำคัญคือ แอททริบิวต์ match และเนื้อหาของเพิ่มเพลด แอททริบิวต์ match จะกำหนดรูปแบบในต้นไม้ต้นฉบับดังในตัวอย่างข้างต้น จะใช้เพิ่มเพลดนี้กับทุกโหนดในต้นไม้ต้นฉบับที่จับคู่กับรูปแบบนั้นๆ ซึ่งในกรณีนี้ก็จะใช้เพิ่มเพลดนี้กับทุกอิลิเมนต์ที่มีชื่อว่า first

คุณได้ระบุสิ่งที่จะได้ออกมาเป็นต้นไม้ผลลัพธ์ไว้ในเนื้อหาของอิลิเมนต์ template ในที่นี้เราได้กำหนดข้อความบางอย่างขึ้นมาง่ายๆ ถ้าเพิ่มเพลดนี้ถูกรวมเข้าไปในสไตลชีตแล้วทุกๆ ครั้งที่กลไก XSLT ค้นพบอิลิเมนต์ <first> ในต้นไม้ต้นฉบับมันก็จะให้เอาต์พุตออกมาเป็นข้อความว่า “My Project”

เนื้อหาของเพิ่มเพลดสามารถเป็นอะไรที่ซับซ้อนมากกว่าการให้เอาต์พุตเป็นข้อความธรรมดา และก็เป็นอย่างนั้นเสมอ มีอิลิเมนต์ของ XSLT อยู่จำนวนหนึ่งที่คุณสามารถใส่เข้าไปในเพิ่มเพลดเพื่อให้ทำหน้าที่หลากหลายอย่างได้ ยกตัวอย่างเช่นมีอิลิเมนต์หนึ่งคือ <xsl : value-of> ที่สามารถเอาข้อมูลจากต้นไม้ต้นฉบับแล้วเพิ่มมันเข้าไปในต้นไม้ผลลัพธ์ได้ ใ้คัดต่อไปนี้จะทำงานเช่นเดียวกันกับเพิ่มเพลดก่อนหน้านี้ของเรา แต่แทนที่จะให้เอาต์พุตเป็น “My Project” เพิ่มเพลดนี้จะเพิ่มเนื้อหาของอิลิเมนต์ใดๆ ที่มีชื่อว่า first เข้าไปในต้นไม้ผลลัพธ์ :

```
<xsl : template match="first"><xsl : value-of select = "./*" /></xsl : template>
```

พูดง่ายๆ คือ ถ้ากลไกการทำงานของ XSLT ค้นพบ :

```
<first> Anuruk </first>
```

มันจะแสดงเอาต์พุต Anuruk ออกมา และถ้าค้นพบ :

```
<first> Anuchin </first>
```

ก็จะแสดงเอาต์พุต Anuchin ออกมา

การเชื่อมความสัมพันธ์ระหว่างสไตลชีตด้วยคำสั่งประมวลผล

XSL สไตลชีตสามารถเชื่อมความสัมพันธ์กับเอกสาร XML ด้วยการใช้คำสั่งประมวลผล (PI) ที่เป็นชนิดเดียวกับสไตลชีต ดังนี้

```
<?xml – stylesheet type = “text / xsl” href = “stylesheet.xml” ?>
```

อย่างไรก็ตาม เนื่องจากปัจจุบัน IES เป็นเว็บเบราว์เซอร์เดียวที่เข้าใจภาษา XML และ XSL ดังนั้นสไตลชีตของคุณจะต้องใช้ XSLT ที่เป็นไวยากรณ์เก่า เพื่อให้กลไกการทำงานของ XSLT ของ IES เข้าใจได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.2.1.6.4 XPath

สำหรับภาษาที่มีชื่อเรียกว่า XPath ซึ่งทำหน้าที่ในการชี้ตำแหน่ง (address) ของส่วนต่างๆ ของเอกสาร XML เพื่อที่จะอนุญาตให้เราได้รับชิ้นข้อมูลที่เราต้องการได้ XSL ใช้ XPath อย่างแพร่หลาย คุณสามารถใช้ XPath เพื่อที่จะบอกตำแหน่งเอกสารของคุณได้โดยการกำหนดตำแหน่งท้องถิ่น (location path) ซึ่งจะนำพา XPath ว่าคุณกำลังจะไปที่ไหนในเอกสารทีละชั้นๆ แน่นอนว่า, คุณไม่สามารถรู้ว่าคุณจะไปที่ไหนได้จนกว่าที่คุณจะรู้ว่าคุณอยู่ที่ใดเสียก่อน ดังนั้น XPath จึงจำเป็นต้องรู้จัก context node ซึ่งก็คือ ส่วนของเอกสาร XML ที่คุณกำลังจะตั้งต้นอยู่นั่นเอง (เราได้เห็นทางลัดใน XPath สำหรับ context node เริ่มต้น ซึ่งก็คือ “.” อย่างง่ายๆ)

ให้คิดว่านิพจน์ของ XPath ก็คือการบอกทิศทางตลอดเอกสาร XML นั่นเอง (เช่น “ตอนนี้คุณอยู่ตรงไหนนะ คุณต้องลงไปจากตรงนี้ แล้วเลี้ยวขวาตรงถนนถัดไป ก็จะเห็นบ้านหลังแรกอยู่ทางซ้ายมือ ฯลฯ”)

เอกสาร XML สามารถมองเป็นเหมือนต้นไม้ที่มีกิ่งและใบไม้ แต่ไม่ใช่แค่โหนดเท่านั้นที่ประกอบอยู่ในต้นไม้ดังกล่าว ยังมีแอททริบิวต์, คอมเมนต์, คำสั่งประมวลผล และสิ่งอื่นใดก็ตามที่อยู่ในเอกสารก็สามารถเป็นส่วนหนึ่งของต้นไม้ที่นั่นได้ด้วย ด้วยเหตุผลนี้มันจึงมีประโยชน์ที่จะทำงานกับความสัมพันธ์แบบแม่/ลูกของเอกสาร XML หนึ่งๆ โดยไม่ต้องไปกังวลว่ากิ่งและใบจะเป็นโหนด, แอททริบิวต์, ชิ้นส่วนของเท็กซ์หรืออะไรก็ตาม

XPath ใช้โหนดเงื่อนไข (term node) เพื่ออ้างถึงส่วนต่างๆ ของโหนด ไม่ว่าจะเป็นโหนด, แอททริบิวต์ หรืออะไรก็ตาม

ต่อไปจะพูดถึง โหนดเซต (node-set) ซึ่งเป็นคอลเลกชันของโหนด ยกตัวอย่างเช่น ถ้าคุณบอกให้ XPath ค้นหาโหนดใดๆ ที่มีแอททริบิวต์ id อยู่ มันก็อาจจะมามากกว่าหนึ่งโหนดก็ได้ XPath ก็จะคืนค่ามาเป็นคอลเลกชันของโหนดทั้งหมดในต้นไม้ต้นฉบับที่มีแอททริบิวต์ id อยู่ โหนดเซตสามารถบรรจุโหนดชนิดใดๆ ก็ได้ ไม่ใช่เพียงแค่โหนด

#### 2.2.1.6.5 พาทบอกตำแหน่ง

สิ่งแรกที่ควรสังเกตเกี่ยวกับ XPath ก็คือ คอนเซ็ปต์ของรากของเอกสาร (document root) ซึ่งไม่ใช่โหนดราก เนื่องจากมันสามารถเป็นสิ่งที่อื่นใดก็ได้ที่อยู่ตอนต้นหรือตอนท้ายของเอกสาร XML อยู่ก่อนหรืออยู่หลังโหนดราก XPath ต้องการรากของเอกสารเสมือน (virtual document root) ที่ทำหน้าที่เป็นรากของลำดับชั้นของเอกสาร โดยข้อเท็จจริงแล้ว โหนดรากถือเป็นลูกของรากของเอกสารนั่นเอง

รากของเอกสารไม่ได้ชี้เฉพาะเจาะจงไปที่โหนดใดๆ ในเอกสาร เป็นเพียงแค่ว่า (root) ตามคอนเซ็ปต์ของเอกสารเท่านั้น โดยใน XPath นั้น รากของเอกสารจะถูกระบุด้วย “/”

#### 2.2.1.7 DOM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนนี้เรามีข้อมูลของเราในรูปแบบ XML แล้ว เราสามารถที่จะทำอะไรกับข้อมูลดังกล่าวใน แอปพลิเคชันที่เราเขียนขึ้นได้ เราไม่ต้องการแค่เข้าถึงมันได้เท่านั้น แต่ต้องการแก้ไขและเพิ่มเติมมันได้อีกด้วย จึงเกิดเป็น Document Object Model (DOM) ขึ้น เพื่อให้ทางสายกลางในการทำงานกับเอกสาร XML ทั้งหมด (และเอกสารประเภทอื่นๆ) ผ่านการใช้โค้ดและยังใช้วิธีการอินเตอร์เฟซกับโค้ดดังกล่าวใน โปรแกรมที่เราเขียนขึ้นด้วย

ยกตัวอย่างเช่น DOM จะทำให้เราสร้างเอกสารและส่วนต่างๆ ของเอกสาร, ท่องไปตามเอกสาร, ย้าย, สำเนา และลบส่วนต่างๆ ของเอกสาร, เพิ่มหรือแก้ไขแอตทริบิวต์ได้

### 2.2.1.7.1 DOM คืออะไร

เอกสาร XML หนึ่งๆ ถูกกำหนดโครงสร้างขึ้นมาคล้ายกับโมเดลเชิงวัตถุเป็นอันมาก มันเป็นลำดับชั้นที่มีโหนดอื่นเป็นลูกอย่างมีประสิทธิภาพ

สังเกตว่า โมเดลเชิงวัตถุจะสามารถทำงานได้กับเอกสารนั้นๆ เท่านั้น การทำงานกับเอกสารชนิดอื่นๆ เราจะต้องสร้างโมเดลเชิงวัตถุใหม่ๆ ให้กับมัน ในขณะที่โมเดลเชิงวัตถุเฉพาะเหล่านี้จะมีประโยชน์ในหลายๆ สถานการณ์ แต่ก็ยังมีอีกหลายสถานการณ์ที่วิธีการจัดการทั่วๆ ไปก็เป็นสิ่งจำเป็น นั่นหมายความว่า เราต้องการโมเดลเชิงวัตถุที่สามารถจัดรูปแบบของเอกสาร XML ใดๆ ไม่ว่าจะถูกจัดโครงสร้างมาอย่างไร Document Object Model (DOM) จะจัดการเรื่องทั่วไปพวกนี้ให้เรา

DOM มักจะถูกใส่เข้าไปเป็นชั้นที่กั้นระหว่าง XML parser กับแอปพลิเคชันที่ต้องการข้อมูลในเอกสาร นั่นหมายความว่า parser จะอ่านข้อมูลจากเอกสาร XML แล้วป้อนข้อมูลนั้นให้กับ DOM จากนั้น DOM ก็จะถูกใช้โดยแอปพลิเคชันระดับที่สูงกว่าแอปพลิเคชันสามารถทำอะไรที่มันต้องการกับข้อมูลดังกล่าวก็ได้ รวมถึงการใส่มันเข้าไปในโมเดลเชิงวัตถุอื่นๆ ถ้ามันต้องการเช่นนั้น

ดังนั้นการที่จะเขียนแอปพลิเคชันที่จะเข้าถึงเอกสาร XML ผ่าน DOM ได้ คุณจะต้องมี XML parser และการอิมพลีเมนต์ DOM ติดตั้งเอาไว้ในเครื่องของคุณ แต่ในตอนนี้เราจะให้ความสำคัญกับการใช้ DOM ก่อนการอิมพลีเมนต์ DOM บางอย่าง

### 2.2.1.7.2 DOMString

เพื่อที่จะมั่นใจได้ว่าการอิมพลีเมนต์ DOM ทั้งหมดนั้นจะทำงานได้เหมือนๆ กัน DOM จึงได้กำหนดชนิดข้อมูล DOMString ขึ้นมา ซึ่งเป็นลำดับของหน่วย (ตัวอักษร) ขนาด 16 บิต ที่จะถูกใช้ในที่ใดๆ ที่ต้องการใช้สตริง พุคอีกอย่างหนึ่งก็คือ DOM ได้ระบุเอาไว้ว่า สตริงทั้งหมดต้องเป็น UTF-16 เท่านั้น ถึงแม้ว่าข้อกำหนดของ DOM จะใช้ชนิดข้อมูล DOMString ในทุกๆ ที่ที่มีการพูดถึงสตริง แต่จริงๆ แล้วสิ่งนี้มีไว้เพื่ออำนวยความสะดวกเท่านั้น การอิมพลีเมนต์ DOM หนึ่งๆ ไม่จำเป็นต้องสร้างออบเจ็กต์ DOMString ประเภทใดๆ ขึ้นมาใช้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเตอร์เฟซเป็นคอลเลกชันของเมธอดและพร็อพเพอร์ตี้ที่ออบเจกต์หนึ่งๆ หรือมากกว่านั้นให้การสนับสนุน

### 2.2.1.7.3 การใช้งาน DOM

Document Object Model (DOM) ไม่ได้เป็นเพียง API ที่ทำงานร่วมกับเอกสาร XML เท่านั้น แต่สามารถใช้เพื่อทำงานร่วมกับเอกสาร HTML, สไตล์ชีต CSS และเอกสารอื่นๆ ได้ด้วย โดยความเป็นจริงการนำ DOM ไปใช้สามารถกำหนดขอบเขตให้ทำงานเฉพาะกับเอกสาร XML หรือเฉพาะกับเอกสาร HTML หรือสามารถถูกสร้างให้ทำงานกับเอกสารชนิดต่างๆ ได้ด้วย เช่น DOM ที่เป็นส่วนหนึ่งของ parser XML จะมีเพียงการใช้ API ที่เฉพาะเจาะจงสำหรับ XML เท่านั้น เพราะเหตุว่า DOM ที่เป็นส่วนหนึ่งของเว็บเบราว์เซอร์จะมี API ที่เฉพาะเจาะจงสำหรับ HTML และ CSS รวมอยู่ด้วย เพื่อให้เข้าถึงข้อมูลชนิดเหล่านี้ทางโปรแกรมได้

การรวมกันของภาษาสคริปต์ DOM (ด้วย API สำหรับ HTML) กับ HTML คือสิ่งที่ทำให้ Dynamic HTML (หรือ DHTML) มีความเป็นไปได้ เนื่องจากเนื้อหาของเอกสาร HTML นั้นจะถูกนำมาแสดงผ่านโมเดลเชิงวัตถุเมื่อเว็บเพจหนึ่งๆ ถูกโหลดเข้าไปในเว็บเบราว์เซอร์ที่รับรู้ DHTML ออบเจกต์ทั้งหลายจะถูกสร้างขึ้นให้กับแต่ละอิลิเมนต์และทุกๆ อิลิเมนต์บนเพจ สิ่งนี้ทำให้คนที่เขียนเว็บเพจใส่สคริปต์ลงไปบนเพจได้เพื่อเรียกใช้เมธอดและพร็อพเพอร์ตี้ทั้งหลายของออบเจกต์เหล่านั้น

เนื่องจากมีประเภทของการนำ DOM ไปใช้ต่างๆ กัน DOM จึงจัดให้มีแกนของ DOM (DOM Core) ซึ่งเป็นเซตของอินเตอร์เฟซที่เป็นแกนหลักในการทำงานกับเอกสารขั้นพื้นฐานรวมถึงโมดูลทั้งหลายที่มีให้เลือกจำนวนหนึ่งเพื่อทำงานกับเอกสารอื่นๆ เช่น DOM HTML, DOM CSS เป็นต้น โมดูลเหล่านี้เป็นเซตของอินเตอร์เฟซเพิ่มเติมที่สามารถนำไปใช้ได้

อินเตอร์เฟซที่เป็นแกนหลักเหล่านี้สามารถแบ่งออกได้เป็นอินเตอร์เฟซมาตรฐาน (Fundamental Interface) และ อินเตอร์เฟซขยาย (Extended Interface)

- อินเตอร์เฟซมาตรฐานต้องถูกใช้งานโดยการทำงานของ DOM เท่านั้น แม้กระทั่งอินเตอร์เฟซที่จะทำงานได้กับ เอกสารแบบ non-XML เท่านั้นก็ตาม
- อินเตอร์เฟซส่วนขยายต้องถูกใช้งานโดยการทำงานของ DOM เท่านั้น ซึ่งจะทำงานกับ HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1.8 ประโยชน์จาก XML

สำหรับประโยชน์ของ XML นั้น เป็นด้านความยืดหยุ่นในการใช้งานสำหรับแอปพลิเคชันที่อิงกับ Web Base ที่ใช้งานในการค้นหาข้อมูล มีความยืดหยุ่นในการพัฒนาเว็บ สามารถผสมผสานข้อมูลจากหลายแหล่ง จากแอปพลิเคชันที่ต่างกัน สามารถแสดงข้อมูลแบบต่างๆ และสามารถ update ข้อมูลให้ทันสมัยเสมอ และคาดว่าจะเป็นมาตรฐานใหม่ของระบบเปิด ซึ่งนับเป็น format ใหม่สำหรับการส่งข้อมูลบนเว็บที่มากด้วยข้อมูลหลายแบบ แต่ส่งผ่านด้วยเทคโนโลยีที่บีบอัดข้อมูลที่ให้ความเร็วได้รับการสนับสนุนจากผลิตภัณฑ์ค่ายไมโครซอฟท์

#### สถาปัตยกรรม XML

ภาษา XML ได้รับการสนับสนุนจาก W3C ให้นักพัฒนาเว็บแอปพลิเคชันได้หันมาใช้เป็นส่วนประกอบของการพัฒนาเว็บไซต์ เพราะ XML มีประสิทธิภาพและมีความน่าเชื่อถือสูงในการแปลงข้อมูลและโครงสร้างข้อมูลให้สามารถนำไปใช้งาน สถาปัตยกรรม 3 Tiered ที่ XML สามารถสร้างขึ้นจากระบบข้อมูลที่ใช้โมเดลของ 3-tier โครงสร้างของข้อมูลต่างๆ สามารถนำมาแสดงตามข้อกำหนด หรือรูปแบบที่ต้องการตามการใช้งานได้ (คุณภาพประกอบ) เราจะเห็นโครงสร้างทางสถาปัตยกรรมของ XML ที่ยืดหยุ่นเหมาะกับการพัฒนาร่วมกับเว็บแอปพลิเคชัน

#### ส่วนประกอบของข้อมูล

XML เป็นการทำงานในระดับกลาง middle tier ที่สามารถเรียกใช้ฐานข้อมูลได้หลากหลายระบบฐานข้อมูลและ โอนข้อมูลให้อยู่ในรูปแบบของ XML และมีการให้รายละเอียดเกี่ยวกับตัวข้อมูล โครงสร้างต่างๆ ของระบบฐานข้อมูลได้ XML เป็นระบบเปิดที่นำเสนอข้อมูลในรูปแบบ text ผ่านทาง HTTP เหมือนกับ HTML แต่จะมีคุณสมบัติในการให้ข้อมูลแบบ real time อัปเดตหรือเปลี่ยนแปลงได้ตามข้อกำหนด การแสดงข้อมูลจาก XML ใน HTML จะเป็นการเพิ่มในส่วนจของรายละเอียดข้อมูล ที่มีการเรียกใช้จากแหล่งหรือฐานข้อมูลที่เชื่อมโยงกันในหลายแหล่ง เพื่อให้ HTML มีความสมบูรณ์มากขึ้น ในอนาคตการพัฒนาเว็บหรือการเขียนและสร้าง HTML ไม่จำเป็นต้องมีการเขียนชุดคำสั่งที่ยุ่งยาก ซับซ้อนมากก็สามารถทำงานร่วมกับระบบข้อมูลได้อย่างมีประสิทธิภาพ XML จะทำการกำหนดค่าสำหรับโครงสร้างข้อมูลที่จะนำไปแสดงใน HTML นอกจากนั้นยังสามารถนำไปสนับสนุนระบบการแลกเปลี่ยนข้อมูลข่าวสารทาง Electronic ได้อย่างดีอีกด้วย

#### เครื่องมือและการสนับสนุน

เครื่องมือเครื่องมือที่ใช้ในการสนับสนุนการพัฒนาาระบบที่ใช้มาตรฐาน XML ได้รับการสนับสนุนจากคู่ค้าหลายราย นอกจากนั้นได้มีการร่วมมือเพื่อกำหนดมาตรฐานในการพัฒนาแอปพลิเคชันให้รองรับการทำงาน ไม่ว่าจะเป็นทางด้านระบบข้อมูล ระบบสารสนเทศ ข่าวสารที่ต้องการความรวดเร็ว เพื่อทันต่อเวลาการนำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1.9 บทสรุปของ XML

XML มีความยืดหยุ่นพอในการนำเสนอข้อมูลแบบต่างๆ ที่สามารถให้รายละเอียดโครงสร้างข้อมูลตามระดับและความต้องการในการนำไปใช้งาน

## 2.3 ASP.NET เว็บแอปพลิเคชัน

### 2.3.1 ที่มาของ ASP

แอพลิเคชันเซิร์ฟเวอร์เพจ(ASP) ของบริษัทไมโครซอฟท์ คือเทคโนโลยีใหม่ในการสร้างเว็บเพจแบบไดนามิก หน้าของ ASP นั้น ประกอบด้วย HTML markup และสคริปต์ทางฝั่งเซิร์ฟเวอร์ ซึ่งจะประมวลผลตาม อินพุตที่ได้รับเข้ามา ณ ขณะนั้น และสร้างโค้ด HTML ขึ้นมาจากการประมวลผล สคริปต์ทางฝั่งเซิร์ฟเวอร์จะทำงานเมื่อมีการร้องขอ (request) หน้า ASP ส่งมาถึงฝั่งเซิร์ฟเวอร์ อินพุตของหน้า ASP จะมาจากบราวเซอร์ทางฝั่งไคลเอนต์ผ่านทางเมธอด HTTP POST และ GET ASP มี อ็อบเจกต์โมเดลเพื่อความสะดวกขึ้นในการทำงานของผู้พัฒนา นอกจากการใช้อ็อบเจกต์จาก ASP อ็อบเจกต์โมเดล เช่น Application , Server , Request , Respond และ Session แล้ว ผู้พัฒนายังสามารถใช้ COM คอมโพเนนต์ อื่นๆ ที่มีอยู่บนเซิร์ฟเวอร์ ได้

ในการพัฒนาแอปพลิเคชัน โดยใช้ ASP ก็ยังมีความยุ่งยากระหว่าง HTML markup และสคริปต์ทางฝั่งเซิร์ฟเวอร์ รูปแบบของหน้า ASP ทำให้เกิดความยุ่งยากในการจัดการและกรนำมาใช้ใหม่ จึงมีการพัฒนาโดยเพิ่มไฟล์และฟังก์ชันในสคริปต์ทางฝั่งเซิร์ฟเวอร์ อย่างไรก็ตาม ความพยายามนี้ทำให้เกิดข้อเสียในด้านเวลา การจัดการเครือข่ายของไฟล์ที่เกี่ยวข้อง และยังมีผลกระทบทางด้านประสิทธิภาพจากการมี include ซ้อนกัน และการจัดการออบเจกต์ ไอดี และ ขอบเขตของตัวแปร

ผู้พัฒนาที่ต้องทำงานที่เกี่ยวข้องกับการทำงานกับบราวเซอร์หลายตัวมักพบปัญหาในการทำเว็บแอปพลิเคชันให้สามารถทำงานได้กับหลายบราวเซอร์ซึ่งจะต้องคำนึงถึงความสามารถของบราวเซอร์ที่ไคลเอนต์ใช้ ซึ่งวิธีการแก้ไขปัญหามักจบลงด้วยการใช้แท็กHTMLที่เป็นแท็กพื้นฐานที่สุด และใช้สคริปต์ทางฝั่งไคลเอนต์เพื่อให้เป็นที่เข้าใจได้โดยทุกบราวเซอร์บางครั้งมีความพยายามสร้างHTML marker สำหรับแต่ละบราวเซอร์โดยเฉพาะ ซึ่งได้ผลดีแต่เสียเวลาและต้องใช้ความพยายามอย่างมาก

การเขียนสคริปต์ใน ASP ใช้ภาษาได้จำกัดเช่น VB สคริปต์ และ Java สคริปต์เป็นต้น และสคริปต์ ทางฝั่งเซิร์ฟเวอร์ จะถูกแปลทุกครั้งทีหน้าASPนั้นถูกเรียก ซึ่งไม่เป็นผลดีต่อประสิทธิภาพในการทำงาน นอกจากนั้นในเรื่องการจัดการกับสถานะ ใน ASP นักพัฒนาจะต้องทำทุกอย่างด้วยตนเอง โดยจะต้องเก็บ สถานะไว้ใน SQL เซิร์ฟเวอร์ หรือเนื้อที่เก็บข้อมูลภายนอกอื่นๆ จะเห็นได้ว่า

จะเห็นได้ว่าแม้ASP จะเป็นเทคโนโลยีที่ดีในการสร้างเว็บเพจแบบไดนามิก แต่ก็ยังมีปัญหาอยู่อีกหลายด้าน จึงมีการแก้ไขและพัฒนามาเป็น ASP.NET

### 2.3.2 ที่มาของ ASP.NET

นักพัฒนาที่ใช้ Visual Basic มักจะคุ้นเคยกับความสะดวกสบายในการเขียนโปรแกรมโดยใช้ฟอร์ม และคอนโทรลซึ่งมีรูปแบบที่สะดวก โดยลากสิ่งที่ต้องการลงมาไว้บนฟอร์มและเขียนโค้ดโดยใช้การเขียนโปรแกรมแบบ event-handling เนื่องจากวิธีนี้เป็นวิธีที่ไม่ยุ่งยาก จึงได้ถูกนำมาใช้ใน ASP.NET

ASP.NET ทำให้การพัฒนาหน้าเว็บสะดวกขึ้นด้วยการเขียนโปรแกรมโดยใช้ฟอร์ม ใน ASP.NET เราเรียกฟอร์มเหล่านี้ว่า เว็บฟอร์ม (Web Forms) ซึ่งนำมาแทนที่หน้า ASP การเขียนโปรแกรมแบบเว็บฟอร์มนั้นเป็นแบบ event base เช่นเดียวกับใน visual basic นอกจากนี้ ASP.NET ยังแยกส่วนระหว่างส่วนแอปพลิเคชันและส่วนในการแสดงผลอีกด้วย

ASP.NET เกี่ยวข้องกับรูปแบบการเขียนโปรแกรม ASP โดยมีคุณสมบัติพิเศษเพิ่มเติมขึ้นมา ดังนี้

- มีการแยกส่วนเด็ดขาดระหว่างส่วนโค้ดการประมวลผลของแอปพลิเคชันและส่วนแสดงผล (HTML markup) จึงไม่มีความสับสนในโค้ดทั้งสองส่วนอีก
- มีกลุ่มของเซิร์ฟเวอร์คอนโทรลจำนวนมากให้เลือกใช้ ซึ่งจะถูกแปลงเป็น HTML โดยอัตโนมัติ
- มีการจัดการสถานะ โดยใช้อ็อบเจกต์เซสชัน
- เป็นรูปแบบการเขียนโปรแกรมแบบ event-base ทางฝั่งเซิร์ฟเวอร์ ทำให้ง่ายขึ้น
- สามารถเขียนโค้ดส่วนแอปพลิเคชันโดยใช้ภาษาใดใน Microsoft .NET ก็ได้ เช่น ภาษา VB, C# เป็นต้น โค้ดของแอปพลิเคชันบนฝั่งเซิร์ฟเวอร์จะถูกคอมไพล์เพื่อประสิทธิภาพการทำงานที่ดีกว่า
- มี Visual Studio.NET เป็นเครื่องมืออำนวยความสะดวกในการพัฒนาเว็บฟอร์ม

### 2.3.3 รู้จักกับ System.Web.UI Namespace

ในส่วนนี้จะอธิบายถึงคลาส ที่สำคัญที่อยู่ในเนมสเปซชื่อ System.Web.UI ในเฟรมเวิร์คของ ASP.NET เนมสเปซ System.Web.UI นั้นระบุคลาสและอินเตอร์เฟซที่ใช้ในการสร้างสิ่งต่างๆ ในคลาสเว็บฟอร์ม ที่สำคัญที่สุดใน System.Web.UI คือคลาส Control ซึ่งบอกคุณสมบัติ เมธอด และ event ของแต่ละเซิร์ฟเวอร์คอนโทรลในเฟรมเวิร์คของเว็บฟอร์ม อีกคลาสที่สำคัญคือเพจ ซึ่งจะกล่าวถึงในลำดับต่อไป

#### 2.3.3.1 คลาส Control

คลาส Control เป็นรากฐานของคอนโทรลทั้งหมด ตัวอย่างเช่น textbox หรือ button ก็คือคอนโทรลโดยทั่วไปคลาส Control จะมีฟังก์ชันและคุณสมบัติของส่วนติดต่อกับผู้ใช้ทุกตัว ทุกอย่างที่เราเห็นใน ASP.NET ก็เป็นคลาสที่สืบทอดมาจากแหล่งใดแหล่งหนึ่งทั้งสิ้น

2.3.3.1.1 คุณสมบัติของ Control

คลาส Control มีคุณสมบัติที่สำคัญดังต่อไปนี้ Control, ID, Parent, EnableViewState, Visible, Context และ ViewState คุณสมบัติของคอนโทรลเหล่านี้แทนลูกของ instace ของคอนโทรลคุณสมบัติของคลาสแม่จะบอกคลาส แม่ของคอนโทรล คุณสมบัติทั้งสองนี้ ทำให้มีลำดับชั้น(hierarchy)ของคอนโทรลในหน้าเว็บคุณสมบัติ ID ทำให้คอนโทรลถูกเข้าถึงจากการเขียนโปรแกรมได้โดยใช้ IDตามด้วยจุดและชื่อคุณสมบัติหรือเมธอดอื่นๆที่เราต้องการจะเข้าถึงของคอนโทรลนั้น เช่น MyObjectId.propertyname และยังทำให้เราสามารถเขียนตัวจัดการกับเหตุการณ์ที่เกิดขึ้นกับคอนโทรลได้ด้วย

EnableViewState เป็น flag ที่บอกว่าคอนโทรลจะยังคงเก็บview ของสถานะมันไว้หรือไม่ เช่นเดียวกับ view สถานะทั้งหมดของคอนโทรลลูกของมัน คุณสมบัติของ Context ทำให้เราสามารถดึงข้อมูลของ HTTP request ได้

ViewState เป็นอินสแตนซ์ของคลาส StateBag ซึ่งใช้ในการเก็บชื่อและค่าของข้อมูลที่สามารถสร้างให้เข้าถึงได้โดยหลาย request สำหรับหน้าเพจเดียว ชื่อและค่าเหล่านี้คืออินสแตนซ์ของคลาส StateItem

2.3.3.1.2 เมธอดในคลาส Control

เมธอดของคลาส Control นั้นมีเป็นจำนวนมาก อย่างไรก็ตามเราจะกล่าวถึงเพียงเมธอดที่มีความสำคัญ

เมธอด DataBind

เป็นการโยงคอนโทรลเข้ากับแหล่งข้อมูล เมื่อเมธอด นี้ถูกเรียก แต่กของการผูกข้อมูล <##%> จะถูกนำมาพิจารณาใหม่เพื่อทำให้ข้อมูลใหม่ถูกโยงเข้ากับตำแหน่งและที่เหมาะสม

เมธอด CreateChildControls

จะถูกเรียกก่อนที่คอนโทรล compositional custom จะถูกจัดแสดงคอนโทรล compositional custom นั้นคล้ายกับคอนโทรล ActiveX ก็อิมันจะจัดการสร้างคอนโทรลอื่นๆ ในการพัฒนาคอนโทรล custom เมธอดนี้สามารถถูกoverride เพื่อให้ผู้พัฒนาคอนโทรล custom สามารถสร้าง คอนโทรลลูก ก่อนทำการแสดง คอนโทรลไม่ว่าในการทำครั้งแรกหรือครั้งต่อไปก็ตาม

เมธอด Render

คล้ายกับ CreateChildControls เริ่มต้นใช้สำหรับพัฒนาคอนโทรล custom ผู้พัฒนาคอนโทรลจะ overrideเมธอดนี้ เพื่อแสดงตัวเนื้อหาในคอนโทรลผ่านทางพารามิเตอร์ HtmlTextWriter ที่เตรียมไว้

เมธอด SaveViewState และ LoadViewState

เก็บบันทึกและเรียกซ้ำสถานะของคอนโทรล คอนโทรลของเซิร์ฟเวอร์ คงสถานะของมัน ระหว่างการrequest ผ่านทางเมธอด นี้

### 2.3.3.2 คลาสเพจ

คลาสเพจนี้สืบทอดมาจากคลาส Control ซึ่งหมายความว่ามันสืบทอดมาทั้งคุณสมบัติเมธอด และ event ที่แสดงโดยคลาส Control นอกจากการสืบทอดแล้วคลาส เพจ ระบุคุณสมบัติเมธอด และ event ที่เจาะจงมากขึ้น สำหรับหน้าเว็บในเฟรมเวิร์คของ ASP.NET

ใน ASP.NET ทั้ง Application , Request , Respond , Server และ Session เป็นคุณสมบัติของคลาส เพจ นอกเหนือจากอ็อบเจกต์เหล่านี้คลาส เพจ ยังมีคุณสมบัติอื่น ๆ อีก เช่น Cache, ErrorPage, Postback , IsValid, Trace และ Validators

#### 2.3.3.2.1 ตัวอย่างคุณสมบัติและเมธอดของคลาสเพจได้แก่

คุณสมบัติ Cache

ซึ่งไปยังอ็อบเจกต์ cache ของขอบเขตของหน้าปัจจุบัน ในที่นี้ ทรัพยากรเช่นการติดต่อฐานข้อมูล (database connection) จะถูกเก็บเพื่อนำมาใช้ใหม่โดยไม่ต้องสร้างการติดต่อใหม่ หากว่า cache item นั้นยังไม่หมดอายุ

คุณสมบัติ ErrorPage

บอกหน้าที่จะถูกเรียกให้แสดงเมื่อ error เกิดขึ้น อาจใช้ @Page แทนได้

คุณสมบัติ IsPostBack

บอกว่าการเรียกหน้าเว็บเป็นการ โหลดครั้งแรกหรือไม่ หาก IsPostBack มีค่าเป็น true แสดงว่าไม่ใช่การโหลดครั้งแรก จึงไม่ควรทำการ initialize หนักเพจซ้ำ เพื่อให้ประสิทธิภาพดีขึ้น

คุณสมบัติ Validator

รวมกลุ่มของเซิร์ฟเวอร์คอนโทรลซึ่งสามารถกำหนดระยะเวลาการทำงานของตัวเองลงในคุณสมบัติ Validator ของหน้าเว็บ เมื่อหน้าเว็บใดต้องมีการกำหนดอายุของตัวเอง ก็จะกำหนดอายุของคอนโทรลทั้งหมดในหน้านั้นแล้วกำหนดค่าที่เหมาะสมให้แก่คุณสมบัติ IsValid

## คุณสมบัติ Trace

อ้างอิงจากอ็อบเจกต์ TraceContext ซึ่งใช้บอกค่าเตือนหรือข้อความบอกความผิดพลาดที่เกิดขึ้น สามารถกำหนดให้ทำการตรวจสอบหรือไม่ทำก็ได้โดยกำหนดใน web.config ซึ่งเป็นเท็กซ์ไฟล์ที่เก็บค่าขณะรันไทม์ของแอปพลิเคชันของASP.NET

## 2.3.4 ไวยากรณ์สำหรับ Web Form

### 2.3.4.1 ไดรเร็กทีฟ

สำหรับ ASP แบบเก่านั้นไดรเร็กทีฟจะมีรูปแบบเป็น `%@[attribute= value]%` เพราะว่าไดรเร็กทีฟใน ASP นั้นมีเพียงแบบเดียวเท่านั้น

ใน ASP.NET ได้มีการเพิ่มไดรเร็กทีฟใหม่ซึ่งได้แก่ Page, Control, Import, Register, Assemble และ OutputCache และไวยากรณ์ใหม่ที่น่าสนใจคือ `<%@ directive [attribute=value] %>` โดยถ้าการประกาศกระทำโดยใช้ไวยากรณ์แบบเก่าคือไม่มีการใส่ไดรเร็กทีฟ ก็จะหมายถึง ประกาศให้เป็นไดรเร็กทีฟแบบปกติซึ่งก็คือ Page

#### 2.3.4.1.1 @Page

ใน ASP.NET นอกจากจะประกอบด้วยไดรเร็กทีฟแบบ ASP เก่าแล้ว ยังสนับสนุน attribute ที่สำคัญเช่น ErrorPage, Inherits, Src, และ EnableViewState อีกด้วย ตัวอย่างการประกาศใช้งาน

```
<%@ Page Language="VB" ErrorPage="URL" EnableViewState="true"%>
```

#### 2.3.4.1.2 @Control

เช่นเดียวกับ Page ไดรเร็กทีฟ จะมีการประกาศใช้ในไฟล์ (.aspx) สำหรับ Control ไดรเร็กทีฟนั้นจะใช้ใน ASP.NET Control คือประกาศในไฟล์ (.ascx)

#### 2.3.4.1.3 @Import

เราสามารถใส่ import ไดรเร็กทีฟ ในการอ้างถึง เนมสเปซที่จะนำมาใช้ในหน้านั้นๆ ซึ่งโปรแกรมที่เราสร้างขึ้นจะสามารถเข้าถึงทุกๆคลาสที่มีอยู่ในเนมสเปซนั้น ตัวอย่างเช่นถ้าเราต้องการใช้งาน ADO.NET เราต้องใส่ไดรเร็กทีฟดังนี้คือ

```
<%@ Import Namespace="System.Data" %>
```

```
<%@ Import Namespace="System.Data.OleDb" %>
```

#### 2.3.4.1.4 @Register

เป็นไดรเร็กทีฟที่ใช้ ลงทะเบียนการใช้งานเซิร์ฟเวอร์คอนโทรลในหน้า Page นั้นๆ ตัวอย่างเช่น

```
<%@ Register Tagprefix="Ch07"
```

```
TagName="MyCustomControl"
```

```
Src="MyCustomControl.ascx" %>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยชื่อของคอนโทรลคือ MyCustomControl และ คำนำหน้าเมื่อการประกาศใช้คอนโทรลคือ Ch07 ที่อยู่ของคอนโทรลคือ MyCustomControl.ascx

#### 2.3.4.1.5 @Assembly

เป็นไคเร็กทีฟที่ใช้บ่งบอกว่า Page นั้นๆ เป็นของ assemble ไคตัวอย่างเช่น

```
<%@ Assembly Name = "Ch07" %>
```

หมายความว่า โปรแกรมในหน้า Page นี้สามารถเข้าถึงอะไรก็ได้ใน assembly Ch07

#### 2.3.4.1.6 @OutputCache

เราสามารถใส่ Output Cache ไคเร็กทีฟในการควบคุม ระยะเวลาของ output caching ระหว่างที่หน้า Page นั้นเปิดขึ้น

#### 2.3.4.2 ขอบเขตการประกาศคำสั่ง

เช่นเดียวกับ ASP ในการประกาศขอบเขตคำสั่งทำงาน จะหมายถึงการกำหนด ขอบเขตคำสั่งที่จะถูกประมวลผลและสั่งให้ทำงานในหน้า Page นั้นๆ ซึ่งในขอบเขตนั้นจะมี คุณลักษณะที่เรียกว่า "runat" ซึ่งเป็นการใช้กำหนดว่า ขอบเขตของคำสั่งเหล่านี้ จะประมวลผลที่ฝั่งไคลเอนต์ หรือ ฝั่งเซิร์ฟเวอร์ ซึ่งถ้าต้องการให้มีการทำงานที่ฝั่งเซิร์ฟเวอร์ จะต้องมีการกำหนดให้มี คุณลักษณะเป็น "server" และถ้าไม่มีการกำหนดจะหมายความว่าให้ทำงานในฝั่งไคลเอนต์ ตัวอย่างเช่น

```
<script runat="server" [language="codelanguage"]>
    'code
</script>
```

สำหรับการทำงานทั้งสองแบบเราสามารถกำหนดที่อยู่ของคำสั่งได้ โดยการทำเช่นนี้เป็นการแยกส่วนของคำสั่งออกจากส่วนของเอกสาร HTML การประกาศกระทำโดย

```
<script runat="server"
    [language="codelanguage"]
    [src="externalfilename"]/>
```

#### 2.3.4.3 HTML-Control Syntax

HTML คอนโทรลมีความคล้ายกับมาตรฐาน HTML ปกติอยู่มาก จะแตกต่างกันที่มีการเพิ่มการประกาศ "id" และ "runat" แอททริบิวต์ ซึ่งโดยปกติการใช้ id กับการเขียนโปรแกรมนั้นจะเจอกับการเขียนในลักษณะ Dynamic HTML จะแตกต่างกันตรงที่ HTML คอนโทรล จะเป็นลักษณะของ คอนโทรล ที่มีทำงานในฝั่งเซิร์ฟเวอร์ ตัวอย่างเช่น

```
<input id="cmd1" runat="server"
    type="button" value="Click Me" />
```

โดยสังเกตว่าการประกาศต้องเป็นแบบ server เพราะว่าเว็บฟอร์มจะมีการใช้ POST เมธอดในการควบคุม เรื่องเหตุการณ์ทำงานเมื่อทำการประมวลผลคำสั่ง ASP.NET จะทำการสร้างสคริปต์ขึ้นจากอ็อบเจกต์

HtmlControl จากตัวอย่างข้างต้น ASP.NET จะทำการสร้างคอนโทรลขึ้นจาก HtmlInputButton โดยมี id

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ cmd1 นอกจากนี้ยังสามารถควบคุม อีเวนต์ของคอนโทรลได้อีกด้วย โดยการประกาศจะกระทำผ่าน HTML Tag

ตัวอย่างเช่น ถ้าต้องการควบคุมอีเวนต์ onclick สามารถเพิ่มในส่วนของ HTML Tag ดังนี้

```
Onserverclick = "handleServerClick"
```

#### 2.3.4.4 Custom-Control Syntax

ในส่วนของ คอนโทรลที่กำหนดเอง ก็จะคล้ายกับ HTML เซิร์ฟเวอร์คอนโทรล โดย Custom คอนโทรลจะมีส่วนของ id และ runat แอททริบิวต์ แต่อย่างไรก็ตาม Custom คอนโทรลก็ไม่เหมือนกับ HTML มาตรฐาน ในการที่จะใช้งานเราสามารถประกาศตามรูปแบบดังนี้คือ

```
< tagprefix:tagname id="controlID" runat="server" eventname ="eventHandler" />
```

ในการเรียกใช้งาน จะต้องใส่ prefix เป็น asp ดังตัวอย่างต่อไปนี้

```
<asp:TextBox id=txt1 runat=server></asp:TextBox>
```

#### 2.3.4.5 Data-Binding Expressions

การผูกข้อมูลหรือ Data Source เข้ากับคอนโทรลใดๆ เราสามารถทำได้ตามไวยากรณ์ดังนี้คือ

```
<%# data-binding-expression %>
```

โดยการใช้ Data Bind นั้น Web Form จะไม่ทำการ Bind ให้อัตโนมัติ เราจะต้องทำการใส่เมธอด DataBind() เองภายในโค้ด โปรแกรม

ตัวอย่างการใช้ Data Binding ดังนี้คือ

```
<asp:Label text='<%# TestData %>' runat=server/>
```

```
<html>
```

```
<head><title>Data Binding</title></head>
```

```
<body>
```

```
<script language="C#" runat=server>
```

```
/*declare the variable we want to bind to.
```

```
Public string TestData;
```

```
Void Page_Load(Object oSender, EventArgs oEvent) {
```

```
TestData = "HelloWorld";
```

```
Page.DataBind();
```

```
}
```

```
</script>
```

```
</body>
```

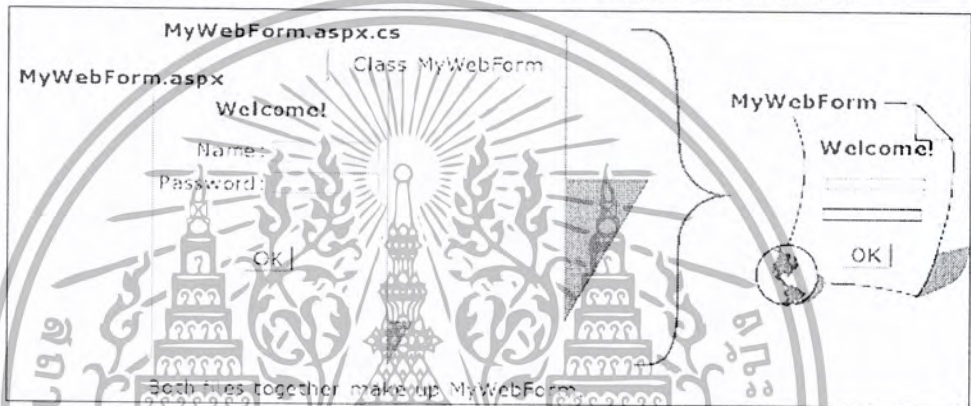
```
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.5 การพัฒนาเว็บแอปพลิเคชันด้วย ASP.NET

#### 2.3.5.1 ส่วนประกอบคอมโพเนนต์ของ Web Form

ใน Web Forms pages นั้น การเขียน program ของ user interface แบ่งเป็นสองส่วนคือ visual Component และ logic ซึ่งก็คล้ายกับการที่ Visual Basic มีการแบ่งระหว่างส่วนของ form และส่วน code เมื่อหลังที่ form นั้นทำงานด้วยตัวเอง ส่วนแสดงผลที่มองเห็นได้นั้นก็คือ Web Form page ซึ่งประกอบด้วยไฟล์ ที่มี HTML หรือ ASP.NET server control. Web Form page ทำงานเหมือนเป็น container สำหรับ static text และส่วนควบคุมที่เราต้องการแสดงผล ในส่วน logic ของ Web Form page ประกอบด้วย โค้ด ที่เรา สร้าง ขึ้นเพื่อสื่อสารกับ form logic การเขียนโปรแกรม จะอยู่แยกกับส่วน file ของ user interface ซึ่งไฟล์นี้จะ เปรียบเสมือน โค้ดที่อยู่เบื้องหลังการทำงาน สามารถเขียนด้วยภาษา Visual Basic หรือ Visual C#



รูปที่ 2-4 ส่วนประกอบคอมโพเนนต์ของ Web Form

ไฟล์ที่ประกอบด้วยคลาสต่างๆและ เบื้องหลังการทำงาน สำหรับ Web Form ทั้งหมดจะถูก คอมไพล์เป็น ไฟล์ dynamic-link library(.dll) page ที่เป็น .aspx ก็จะถูก compile เช่นกัน แต่มี กระบวนการที่ต่างกัน ในครั้งแรกที่ ผู้ใช้ เรียก file หน้า .aspx ASP.NET จะสร้างไฟล์ .NET class มาเป็น ตัวแทนของ page โดยอัตโนมัติ แล้ว คอมไพล์ เป็น ไฟล์ .dll ตัวที่สอง class ที่ถูกสร้างสำหรับ .aspx page สืบทอดคุณสมบัติมาจาก class ที่เป็น โค้ดการทำงานเบื้องหลังที่ถูก compile ไปแล้วก่อนหน้านี้ เมื่อผู้ใช้ ร้องขอ URL ของ web page ไฟล์ .dll จะ run บน server และสร้างผลเป็น HTMLแบบไดนามิคสำหรับ page ที่เรียก

#### 2.3.5.2 วงจรชีวิตของ Web Form

ใน ASP เว็บเพจจะเริ่มต้นวงจรชีวิตเมื่อไคลเอนต์เริ่มทำการร้องขอเพจที่เจาะจงมา / IIS ทำการ กระจายค่าและทำการใช้งานสคริปต์บน ASP เพจ เพื่อทำให้เป็นเนื้อความ HTML ในที่สุดการทำงาน แปลงก็จะสิ้นสุดอายุของเพจนั่นก็จะสิ้นสุด ถ้าคุณมีฟอร์มที่ส่งข้อมูลกลับไป ASP เพจก็จะเริ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการใหม่เหมือนกับ ASP กระทำคำร้องใหม่ โดยไม่ต้องรู้เกี่ยวกับขั้นตอนก่อนหน้าเลย การส่งผ่านข้อมูลกลับไปที่เพจเริ่มต้นสำหรับการประมวลผลก็เป็นการถูกอ้างแบบโพสต์แบ็ก(postback)ด้วย

ใน ASP.NET มีหลายสิ่งที่แตกต่างกันเล็กน้อย เพจยังคงเริ่มที่การร้องขอของไคลเอนท์อย่างไรก็ตามมันจะยังคงอยู่เรื่อยๆเท่าที่ไคลเอนท์ยังคงติดต่อกับเพจอยู่ เพื่อทำให้ง่ายขึ้นอาจจะกล่าวได้ว่าเพจยังคงอยู่ไปเรื่อยแต่ในความเป็นจริงมีเพียงขั้นตอนการดู(view)ของเพจระหว่างการร้องขอเท่านั้น ขั้นตอนการดูเท่านั้นที่อนุญาตให้ควบคุมบนเซิร์ฟเวอร์ที่จะปรากฏว่า ถ้ามันยังคงครองอีเวนต์ของเซิร์ฟเวอร์ เราสามารถค้นหาขั้นตอนการโพสต์แบ็กของเพจผ่านคุณสมบัติ IsPostBack ของเพจอ็อบเจกต์และการกำหนดค่าเริ่มต้นใหม่ การครองอีเวนต์ระหว่างการโพสต์แบ็กมันก็คือการทำ ASP.NET แตกต่างกว่าการพัฒนา ASP ตามแบบธรรมดาอย่างมาก

จากตัวอย่างเราทำการขยายส่วนต้น เช่นการครองโพสต์แบ็ก เมื่อทำการโหลดอีเวนต์ที่ถูกถือครองครั้งแรก เราจะทำการย้ายข้อมูลลงใน drop-down list box หลังจากนั้นเราจะระบุเวลาของอีเวนต์ที่เพิ่มขึ้นโดยไม่มีการ โหลดข้อมูลเข้ามาใหม่เท่านั้น ตัวอย่างนี้แสดงให้เห็นว่าเซิร์ฟเวอร์อีเวนต์แฮนด์เลอร์(handler)ที่ชื่อ handlerButtonClick เป็นขอบเขตของอีเวนต์ที่ชื่อ ServerClick ของปุ่ม

```
<html>
<head><title>Testing Page Event</title></head>
<body>
<script language="C#" runat="server">
    void Page_Init(Object oSender, EventArgs oEvent){
        labelInit.Text = DateTime.Now.ToString();
    }
    void Page_Load (Object oSender, EventArgs oEvent) {
        labelInit.Text = DateTime.Now.ToString();
        if (!IsPostBack){
            selectCtrl.Items.Add("Acrua");
            selectCtrl.Items.Add("BMW");
            selectCtrl.Items.Add("Cadillac");
            selectCtrl.Items.Add("Mercedes");
            selectCtrl.Items.Add("Porche");
        } else {
            labelLoad.text += "(Postback)";
        }
    }
</script>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void handleButtonClick(Object oSender, EventArgs oEvent){
    labelOutput.Text = "You've selected: "+ selectCtrl.Value;
    labelEvent.Text = DateTime.Now.ToString ( );
}
}
</script>
<form runat = server>
    Init Time: <asp:Label id=labelInit runat=server/><br/>
    Load Time: <asp:Label id=labelLoad runat=server/><br/>
    Event Time: <asp:Label id=labelEvent runat=server/><br/>
    Choice: <select id=selectCtrl runat=server/><br/>
    <asp:Label id=labelOutput runat=server/></select><br/>
    <input type=button value=update
        onClick=handleButtonClick runat=server />
</form>
</body>
</html>

```

วงจรกิจของเว็บฟอร์ม ประกอบด้วยสามขั้นตอนหลัก การกำหนดรูปแบบ การถือครองอีเวนต์ และการสิ้นสุด โดยขั้นตอนเหล่านี้จะข้ามผ่านคำร้องขอที่เหมือนกันซึ่งเข้ากับนโยบาย serving-one-page-at-a-time ที่พบใน ASP

### 2.3.5.2.1 การกำหนดรูปแบบ

ขั้นตอนการกำหนดรูปแบบนี้ โหลดอีเวนต์ของเพจจะถูกยกขึ้นมันเป็นงานของคุณในการถือครองอีเวนต์นี้เพื่อติดตั้งเพจ เพราะที่โหลดอีเวนต์จะถูกยกขึ้นเมื่อการควบคุมพร้อมแล้วงานคุณตอนนี้คือการอ่านและทำการอัปเดตคุณสมบัติการควบคุมเหมือนเป็นส่วนของการติดตั้งเพจ ในตัวอย่างโค้ดข้างต้น เราจะถือครอง โหลดอีเวนต์เพื่อย้ายข้อมูลบางอย่างลง drop-down list เราอัปเดตข้อความของการควบคุม labelLoad เพื่อแสดง เวลาโหลดอีเวนต์ที่เกิดขึ้น ในแอปพลิเคชันของคุณ คุณจะต้องนำข้อมูลที่เข้าไปได้จาก ฐานข้อมูลและทำการกำหนดค่าเริ่มต้นฟอร์มฟิลด์ด้วยค่าดีฟอลต์

คุณสมบัติ IsPostBack ของเพจแสดงว่าเพจแรกได้ถูกโหลดหรือมันเป็นการ โทสต์แบคเช่นถ้าคุณควบคุมเนื้อหารายการของข้อมูลคุณจะต้องโหลดการควบคุมเพจแรกที่ถูกโหลดโดยการตรวจสอบคุณสมบัติ IsPostBackของเพจนี้เท่านั้นเมื่อ IsPostBack มีค่าเป็นจริงคุณจะรู้ว่ารายการควบคุมอ็อบเจ็กต์ได้ถูกโหลดด้วยข้อมูลเรียบร้อยแล้ว ไม่จำเป็นที่ต้องย้ายข้อมูลในรายการใหม่ตัวอย่างโค้ดข้างต้นเราจะข้ามการย้ายข้อมูลของ drop-down และแสดงเพียง สดริงว่า "(postback)"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณอาจจะต้องกระทำการผูก(bind)ข้อมูลและกำหนดค่าใหม่ data-binding expressions ในตอนแรก และลำดับต่อมาของเพจนี้

### 2.3.5.2.2 การถือครองอีเวนต์

ในขั้นตอนกลางนี้ ฟังก์ชัน server event-handlingของเพจจะถูกเรียกเป็นผลของบางอีเวนต์เป็นการถูกเรียกให้ทำงานจากฝั่ง ไคลเอนต์ อีเวนต์เหล่านี้มาจากการควบคุมที่คุนแทนที่บนเว็บฟอร์ม

### 2.3.5.2.3 การจบการทำงาน

ที่ขั้นตอนนี้เพจจะสิ้นสุดการแปลงและถูกกำจัดทิ้งแล้วสิ่งที่คุณต้องรับผิดชอบคือการทำความสะอาดไฟล์แฮนด์เลอร์ ยกเลิกการติดต่อกับฐานข้อมูลและปล่อยอ็อบเจกต์ถึงแม้คุณจะมีงานของตัวกำจัดขยะ(garbage collection)ของ CLR ก็ตามแต่เราขังแนะนำอย่างยิ่งยวดว่าควรจำทำความสะอาดด้วยตัวเองเพราะตัวกำจัดขยะจะทำงานเป็นระยะๆเท่านั้น บนการทำงานหนักๆถ้าตัวกำจัดขยะไม่เหมาะสมจะมีทรัพยากรที่ไม่ได้คืนให้ระบบทำให้หน่วยความจำหมดไปได้และทำให้ระบบคุณค้างไป

เราสามารถทำความสะอาดตัวอย่างข้างต้นได้ด้วยการอันโหลดอีเวนต์แฮนด์เลอร์ดังที่จะแสดงต่อไปนี่เพราะว่าไม่มีแสดงในตัวอย่างเราจะแสดงเฉพาะฟังก์ชันเป็น เหน็บเฟลต

```
Void Page_Unload(Object oSender,EventArgs oEvent){
    //cleaning up code here
}
```

## 2.3.6 การจัดการ State และ Scalability

### 2.3.6.1 ASP.NET Session-State Management

ASP.NET ได้ปรับปรุงการจัดการเรื่องสเททใน ASP เดิม โดยปรับปรุงมาเป็นลักษณะโมเดลแบบ out-of-process โดยการจัดการสเททจะมองลักษณะของฟอร์มของ Web Server โดยการสั่งให้มีการสลับไปยังเพจใดๆ ในฟอร์มจะไม่เกิดการหายของเซสชัน

ในการเปลี่ยนแปลงค่าเกี่ยวกับการจัดการการบริหาร session ใน ASP.NET เราสามารถกระทำผ่านไฟล์ที่ชื่อว่า web.config โดยความเป็นจริงแล้วเราสามารถทำการกำหนดค่าได้สองทางได้แก่ทางระดับ แมทชีน และระดับ แอปพลิเคชัน โดยในระดับแมทชีนสามารถกำหนดผ่านไฟล์ที่อยู่ที่

WinNT\Microsoft. NETFramework\<version>\CONFIG\machine.config ในขณะที่ในระดับแอปพลิเคชันสามารถกำหนดผ่าน web.config ที่อยู่ใน ไดเรกทอรีเดียวกับแอปพลิเคชัน

โดยโค้ดที่อยู่ในไฟล์ web.config มีดังนี้คือ

```

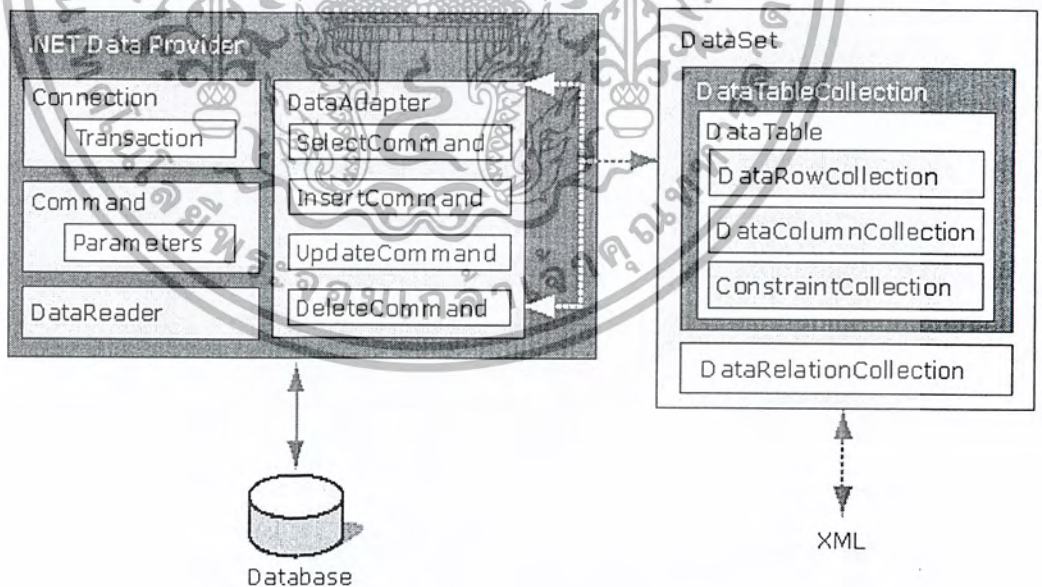
<configuration>
  <system.web>
    <sessionstate
      mode="Inproc"
      cookieless="false"
      timeout="20" />
  </system.web>
</configuration>

```

## 2.4 ADO.NET

### 2.4.1 สถาปัตยกรรม ADO.NET

แต่เดิมการประมวลผลข้อมูลนั้นเป็นแบบ Connection Based ในสถาปัตยกรรม 2-Tier ต่อมาการประมวลผลข้อมูลมีปริมาณที่มากขึ้นในสถาปัตยกรรมแบบ Multi-Tier โปรแกรมเมอร์ ต้องเปลี่ยนไปจัดการแบบ Disconnected เพื่อที่จะทำให้สามารถขยายขีดจำกัดโดยที่รองรับการใช้งานได้มากขึ้น ในอีกประเด็นหนึ่งคือ ปัจจุบันการส่งข้อมูลเป็นลักษณะผ่าน Web มากขึ้นภาษา XML มีบทบาทมากขึ้น โดยส่วนของ ADO.NET ก็จะสามารถในการรับส่งข้อมูลเป็น XML อีกด้วย ซึ่งจะอธิบายในหัวข้อต่อไป



รูปที่ 2-5 แสดงสถาปัตยกรรม ADO.NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4.2 ส่วนประกอบคอมโพเนนต์ของ ADO.NET

ADO.NET ประกอบด้วย 2 คอมโพเนนต์หลักคือ DataSet และ .NET data provider ซึ่งประกอบด้วย กลุ่มของ คอมโพเนนต์คือ Connection, Command, DataReader และ DataAdapter ดังรูปที่ 1

ADO.NET DataSet เป็นคอมโพเนนต์หลักในการเชื่อมต่อข้อมูลแบบ Disconnected, DataSet ถูกออกแบบมาให้มีความเป็นอิสระจากแหล่งข้อมูลด้วยเหตุนี้ มันจึงสามารถใช้ได้ใน Data Source หลายประเภท

DataSet ประกอบด้วยกลุ่มของ DataTable ซึ่ง Table ที่ว่านี้ก็เกิดมาจากการส่ง Query ไปดึงมาจาก ฐานข้อมูล โดย DataTable ก็จะถูกประกอบไปด้วย DataRow, DataColumn รวมถึง primary key, foreign key, ข้อจำกัดต่างๆ พวก Data

Integrity และ ความสัมพันธ์ของข้อมูล (DataRelation) ใน DataTable เอง ในส่วนของการส่งข้อมูล DataSet ได้ใช้ XML เป็นตัวกลางในการส่งข้อมูลระหว่าง Tier ทำให้สามารถนำไปใช้กับการทำ Web Service ได้

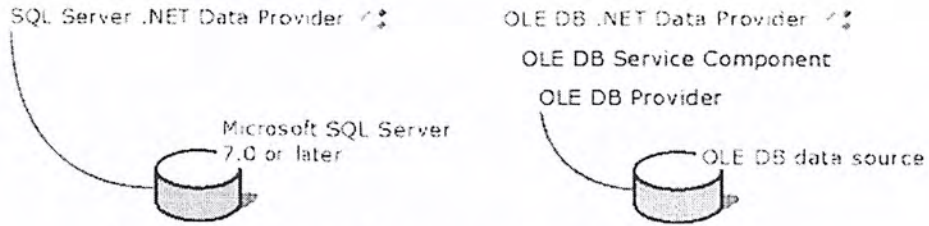
ส่วนหลักอีกส่วนหนึ่งคือ .NET Data Provider ซึ่งสร้างมาเพื่อการประมวลผลข้อมูลแบบ รวดเร็ว และแบบส่ง ไปข้างหน้าอย่างเดียว (forward-only) และเป็นการเข้าถึงข้อมูลแบบอ่านอย่างเดียว (read-only) โดยประกอบด้วยส่วนย่อยๆ ดังต่อไปนี้คือ

1. Connection จัดการเกี่ยวกับเรื่องการติดต่อกับ Data Source หรือ ฐานข้อมูลนั่นเอง
2. Command ทำเกี่ยวกับเรื่องการเข้าใช้งานฐานข้อมูลโดยใช้คำสั่งภาษา SQL รวมถึงการ ทำ stored procedure และ รับส่งข้อมูลพารามิเตอร์ต่างๆ
3. DataReader จัดการเรื่องประสิทธิภาพของการส่งข้อมูลจาก Data Source โดย DataReader นั้น ใช้ได้ในกรณีที่เราดึงข้อมูลมาทีละรายการโดยไม่ย้อนกลับเท่านั้น และเป็นการดึงข้อมูลมาอย่างเดียว เขียนกลับไปไม่ได้
4. DataAdapter จะเป็นเหมือนสะพานเชื่อมต่อระหว่าง DataSet และ Data Source

โดยการทำงานคือเมื่อมีการเปิด Connection และกำหนด Query String SQL แล้ว DataAdapter ใช้ Command Object เพื่อประมวลผลคำสั่งภาษา SQL และทำการดึงข้อมูลลงมาที่ DataSet เพื่อใช้งานต่อไป โดย .NET Framework มี .NET data provider ให้ใช้ 2 แบบคือ

1. SQL Server .NET data provider
2. OLE DB .NET data provider

เปรียบเทียบระหว่าง SQL Server .NET Data Provider และ OLEDB .NET Data Provider



รูปที่ 2-6 เปรียบเทียบระหว่าง SQL Server .NET Data Provider  
และ OLEDB .NET Data Provider

ก็คือ ถ้าใช้ SQL Server .NET Data Provider ต้องใช้กับ Microsoft SQL Server 7.0 หรือใหม่กว่าเท่านั้น โดย คลาสของ SQL Server .NET Data Provider กำหนดโดย namespace System.Data.SqlClient สำหรับเวอร์ชันที่เก่ากว่าก็ใช้ OLE DB .NET Data Provider แทน

#### การเลือกใช้งาน DataReader หรือ DataSet

ในการเลือกใช้ DataReader หรือ DataSet ใน แอปพลิเคชันนั้น ควรพิจารณาถึงชนิดของหน้าที่การทำงานของแอปพลิเคชัน โดยจะเลือกใช้ DataSet ก็ต่อเมื่อ

- มีการส่งข้อมูลระหว่างเซิร์ฟเวอร์ หรือมีดั่งมีการใช้ XML Web service
- มีการติดต่อข้อมูลแบบไดนามิก โดยผูกติดกับ Window Form หรือ การรวมข้อมูลที่เกี่ยวข้องกันมาใช้งานจากแหล่งข้อมูลหลายๆที่
- ให้มีการเก็บข้อมูลลงในแคชของเครื่อง
- มีการประมวลผลข้อมูลโดยไม่ต้องเปิดการเชื่อมต่ออยู่ตลอดเวลา ซึ่งทำให้ไคลเอนต์อื่นๆ สามารถเข้ามาใช้งานฐานข้อมูลได้มากขึ้น

โดยถ้าไม่มีความจำเป็นเหล่านี้ ก็สามารถใช้ DataReader แทนได้ โดยการทำงานจะเป็นแบบ Forward-Only and read-only คือเป็นการส่งข้อมูลให้อย่างเดียว และส่งไปข้างหน้าเท่านั้น และเป็นแบบอ่านได้อย่างเดียว คือ Update ข้อมูลกลับไปไม่ได้ โดยการใช้ DataReader จะเป็นการประหยัดหน่วยความจำลงไปได้มาก

#### 2.4.3 สรุปข้อเปรียบเทียบระหว่าง ADO และ ADO.NET

1. ในการใช้งานกับ Web ซึ่งเป็นลักษณะ ดึงข้อมูลเว็บเพจนั้นเสร็จแล้วก็ทำการตัดสายอัตโนมัติ ซึ่งส่งผลทำให้การทำงานเร็วขึ้น ใ้ว่า ADO จะไม่มี แต่ ADO.NET ทำให้มันสะดวกขึ้น โดย

อัตโนมัติแทนที่โปรแกรมเมอร์ต้องไปควบคุมทุกขั้นตอน

หมายเหตุ ( ADO.NET จะทำ snapshot ของฐานข้อมูลที่เรากำลังติดต่ออยู่เก็บไว้ในหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำของเครื่อง โดยการทำ snapshot นั้นไม่ได้ทำทั้งฐานข้อมูล แต่ทำเฉพาะส่วนที่เรา query เท่านั้น โดยเมื่อดึงข้อมูลเสร็จก็จะทำการ Disconnect ทันที )

2. ส่วนการติดต่อกับฐานข้อมูลและส่วน การจัดการ client cursor (ภาษา ADO เรียกว่า Recordset) นั้นจะเป็นอิสระต่อกันโดยสิ้นเชิง ทำให้ทั้งสองส่วนพัฒนาได้โดยอิสระ
3. ใน ADO เมื่อก่อน Recordset ไม่สามารถพัฒนาได้มากนักเพราะว่าต้องผูกติดกับ Database Server ที่ตัวเองติดต่อด้วย ดังนั้นเพื่อให้สามารถรองรับได้กับ Database ทุกประเภท ทาง Recordset จึงจำเป็นต้องถูกออกแบบมาให้ทำงานเป็นแบบกลางๆ ทำให้ความสามารถด้อยลงไป ดังนั้น ใน ADO.NET จึงแยกในส่วนการติดต่อ Database ออกมา โดยจะมีหนึ่งคลาสต่อหนึ่งต่อ หนึ่งประเภท Database เช่น ถ้าเราต่อกับ Oracle ต้องใช้คลาส หนึ่ง และถ้าต่อกับ Access ต้องใช้ อีกคลาสไม่เหมือนกับ ADO ที่ใช้ OLEDB , คลาส ADO.DB.Connection เพื่อติดต่อกับทุก Database ข้อดีข้อการแยกก็จะทำให้การเชื่อมต่อกับ DBMS แต่ละยี่ห้อเป็นไปได้อย่างอิสระและ สามารถดึงเอาประสิทธิภาพมาใช้ได้อย่างเต็มที่
4. ส่วนของ Client Cursor นั้นปรับปรุงใหม่ แทนที่จะเป็น Recordset ซึ่งรองรับเพียงแค่ตารางเดียว เหมือนเมื่อก่อน แต่ใน ADO.NET สามารถรองรับทั้ง database และแต่ละ table สามารถมี reference integrity ด้วย
5. การส่งผ่านข้อมูลระหว่าง ADO.NET และ DBMS นั้น แทนที่จะส่งโดยใช้ DCOM ซึ่งจะมี ปัญหาเกี่ยวกับ Firewall ทำให้การสื่อสารยุ่งยาก แต่ ADO.NET นั้นจะแปลงข้อมูลอยู่ในรูปของ XML ก่อนแล้วค่อยส่งออกไป ทำให้ใช้ port ของ web มาตรฐาน ทะลุกำแพง firewall ได้

## 2.5 SQL 2000 Server

SQL Server 2000 เป็นโซลูชันฐานข้อมูลและการวิเคราะห์ที่สมบูรณ์แบบ ซึ่งให้ประสิทธิภาพ, ความสามารถในการขยายระบบ และมีความเชื่อถือได้ ซึ่งเป็นสภาพแวดล้อมที่สายธุรกิจขององค์กรและ การทำธุรกิจบนอินเทอร์เน็ตต้องการ ด้วยความสามารถใหม่ๆ ในการสนับสนุน XML และ HTTP ช่วยให้การเข้าถึงข้อมูลและแลกเปลี่ยนข้อมูลเป็นไปโดยง่าย ขณะเดียวกันก็ให้ความสามารถในการวิเคราะห์ที่ ทรงประสิทธิภาพ จึงช่วยเพิ่มคุณค่าของข้อมูลให้มากขึ้น สามารถให้บริการอย่างต่อเนื่องได้ยาวนานสูงสุด มีฟังก์ชันด้านการจัดการที่ล้ำหน้า ช่วยให้การดำเนินงานประจำดำเนินไปอย่างอัตโนมัติ และมีเครื่องมือในการ เขียนโปรแกรมและเซิร์ฟวิสที่ได้รับการปรับปรุงให้ดีขึ้น ช่วยให้การพัฒนาคำเนินไปได้อย่างรวดเร็ว

นอกจากนั้นข้อดีของ SQL Server มีรายละเอียดดังต่อไปนี้

- SQL Server 2000 เป็นแกนหลักสำคัญในการจัดการและวิเคราะห์ข้อมูล สำหรับแอปพลิเคชัน และเซิร์ฟวิสจาก Microsoft .NET ซึ่งกำลังจะมาสู่ผู้ใช้ในอนาคตอันใกล้นี้
- ทำให้การรวมกันของระบบ back-end และการส่งข้อมูลข้ามไปพร้อมกันเป็นไปโดยง่าย โดยใช้ ฟังก์ชันการทำงานและการสนับสนุน XML ของ SQL Server 2000 ที่พร้อมสำหรับมาตรฐาน

อื่นๆ บนอินเทอร์เน็ต เช่น XPath, XSL และ XSLT ผู้พัฒนาเว็บสามารถเข้าถึงข้อมูลได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XML โดยไม่ต้องเขียนโปรแกรมบนฐานข้อมูลแบบสัมพันธ์อีก ในขณะที่ผู้บริหารฐานข้อมูลสามารถจัดการข้อมูลในรูปแบบ XML ได้อย่างง่ายดาย โดยใช้ Transact-SQL (T-SQL) และ stored procedure

- เชื่อมโยงไปยังฐานข้อมูล SQL Server 2000 และ OLAP cube ได้อย่างคล่องตัว โดยอาศัยเว็บและไม่ต้องเขียนโปรแกรมเพิ่มเติม มีการเชื่อมโยงฐานข้อมูลผ่านทาง HTTP ที่ได้รับการรักษาความปลอดภัย เพื่อให้แน่ใจว่าสามารถเรียกใช้ข้อมูลได้ผ่านทาง URL และมีส่วนติดต่อกับผู้ใช้ (UI) ที่เข้าใจได้ง่าย แม้ผู้ใช้จะเป็นนักพัฒนามือใหม่ก็ตาม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# วิวัฒนาการของสถาปัตยกรรมแบบกระจาย

### 3.1 สถาปัตยกรรมแบบหนึ่งเทียร์ (One Tier Architecture/Centralize Host)

สถาปัตยกรรมแบบหนึ่งเทียร์ คือระบบผู้ใช้หลายคนที่เรียบง่ายที่สุด โดยจะมีการเก็บข้อมูลและประมวลผลอยู่ที่คอมพิวเตอร์ส่วนกลางทั้งหมด (เช่น เมนเฟรมคอมพิวเตอร์หรือมินิคอมพิวเตอร์) ผู้ใช้จะติดต่อกับคอมพิวเตอร์ส่วนกลางโดยผ่านทางเทอร์มินอล ซึ่งเป็นเพียงแค่อุปกรณ์อินพุตและเอาต์พุตเพียงอย่างเดียว ไม่มีการประมวลผลในระบบเช่นนี้มีข้อเสียคือ

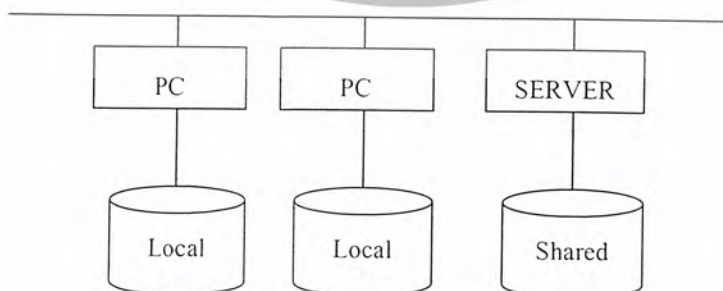
- การที่ทุกๆ กระบวนการทำงานทำในคอมพิวเตอร์ส่วนกลางทั้งหมด ทำให้ต้องแบ่งกำลังการประมวลผลไปให้ผู้ใช้แต่ละคน อาจมีผลกระทบต่อเวลาการตอบสนอง ทำให้ตอบสนองได้ช้าลง
- ถ้าเกิดความขัดข้องที่คอมพิวเตอร์ส่วนกลาง ผู้ใช้ทุกคนจะไม่สามารถใช้งานได้
- ค่าใช้จ่ายในการบำรุงรักษาทางด้านฮาร์ดแวร์ และซอฟต์แวร์สูงมาก



รูปที่ 3-1 สถาปัตยกรรมแบบหนึ่ง

### 3.2 เครือข่ายภายในและการแบ่งปันทรัพยากรภายในเครือข่าย (PC LANs/ทรัพยากร Sharing LANs)

คอมพิวเตอร์ส่วนบุคคล (Personal Computer: PC) ได้ถือกำเนิดขึ้นมาเพื่อทดแทนความไม่ยืดหยุ่นของระบบเมนเฟรม ที่ซึ่งนั่นมีค่าใช้จ่ายทางด้านซอฟต์แวร์และฮาร์ดแวร์สูงกว่าระบบเมนเฟรมมาก พีซีในยุคแรกเริ่มเป็น พีซีที่มีการใช้งานอยู่เครื่องเดียว. ในก้าวต่อมาจึงมีการเชื่อมต่อพีซีเข้าด้วยกันผ่านเครือข่ายภายใน (Local Area Network: LAN)



รูปที่ 3-2 เครือข่ายภายในของเครื่องคอมพิวเตอร์ส่วนบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนที่เป็นเซิร์ฟเวอร์นั้น อาจเป็นเซิร์ฟเวอร์ที่ให้บริการเพิ่มข้อมูล หรือเซิร์ฟเวอร์ที่ให้บริการเครื่องพิมพ์ (File/Print Server) ก็ได้ แต่แอปพลิเคชันจะยังคงทำงานอยู่บนพีซีของตัวเอง การที่ข้อมูลอยู่บนดิสก์ที่ใช้ร่วมกันที่เซิร์ฟเวอร์ และแอปพลิเคชันทำงานอยู่ที่พีซี ทำให้เกิดการถ่ายโอนข้อมูลจำนวนมาก และเป็นไปอย่างต่อเนื่อง ยกตัวอย่างเช่น การคิวรี่ข้อมูลจากข้อมูลทั้งหมด 10,000 เรคคอร์ด เซิร์ฟเวอร์จะต้องส่งข้อมูลทั้ง 10,000 เรคคอร์ดนั้นมาประมวลผลคำสั่ง SQL ที่เครื่องไคลเอนต์

### 3.3 สถาปัตยกรรมแบบสองเทียร์ (ไคลเอนต์/เซิร์ฟเวอร์)

#### 3.3.1 ดาต้าเบสเซิร์ฟเวอร์ (Database Server)

ในระบบดาต้าเบสเซิร์ฟเวอร์นั้นดาต้าและดาต้าเบสเอ็นจิน (DBMS) จะรันอยู่ที่เซิร์ฟเวอร์ ส่วนหน้าจอใช้งานจะรันอยู่บนพีซี ซึ่งเรียกว่าไคลเอนต์ ในการสืบค้นข้อมูลนั้น จะส่งคำสั่ง SQL statement ไปสืบค้นที่เซิร์ฟเวอร์และเซิร์ฟเวอร์จะส่งเฉพาะผลลัพธ์กลับไปไคลเอนต์ ซึ่งจะช่วยลดความหนาแน่นของข้อมูลในระบบเครือข่ายลงได้ ไคลเอนต์กับเซิร์ฟเวอร์จะเป็นอิสระต่อกัน แต่จะมีแอปพลิเคชันทำงานร่วมกันอยู่ แอปพลิเคชันฝั่งเซิร์ฟเวอร์คือ ดาต้าเบสเอ็นจิน ส่วนแอปพลิเคชันฝั่งไคลเอนต์ ก็คือแอปพลิเคชันต่างๆ ที่ต้องการติดต่อกับดาต้าเบส แอปพลิเคชันทั้งสองส่วนจะติดต่อสื่อสารกันผ่านเน็ตเวิร์กโพรโตคอล ซึ่งเป็นความรับผิดชอบของ DBMS โดย DBMS จะจัดเตรียมโครงสร้างพื้นฐานทั้งหมดให้

ข้อเสียของไคลเอนต์เซิร์ฟเวอร์ ก็คือการที่แอปพลิเคชันนั้นรันอยู่บนไคลเอนต์ การแก้ไขแอปพลิเคชันจึงต้องนำทุกๆไคลเอนต์ ทำให้มีค่าใช้จ่ายในการบำรุงรักษาสูงมาก

#### 3.3.2 แอปพลิเคชันเซิร์ฟเวอร์

แอปพลิเคชันเซิร์ฟเวอร์จะมีความสามารถมากกว่าการแชร์ไฟล์หรือเครื่องพิมพ์ในเครือข่ายภายในหรือการจัดการดาต้าเบสในดาต้าเบสเซิร์ฟเวอร์ แอปพลิเคชันเซิร์ฟเวอร์นั้นสามารถใส่แอปพลิเคชันที่มี ฟังก์ชันการทำงานต่างๆ ได้ และเรียกใช้เมื่อต้องการได้ โดยแอปพลิเคชันที่จะเรียกใช้ฟังก์ชันที่อยู่บนเซิร์ฟเวอร์จะต้องใช้เน็ตเวิร์กโพรโตคอล เช่น TCP/IP, SPX/IPX หรือโพรโตคอลในระดับสูง เช่น DCOM, COBRA, RMI

### 3.4 สถาปัตยกรรมแบบสามเทียร์

ในระบบสถาปัตยกรรมแบบสองเทียร์นั้น ทุกๆ ไคลเอนต์ติดต่อโดยตรงกับดาต้าเบส ซึ่งการทำเช่นนี้เหมาะสมกับงานขนาดเล็ก แต่ไม่เหมาะสมกับองค์กรใหญ่ๆ ที่มีผู้ใช้จำนวนมาก โดยเฉพาะอย่างยิ่งในระบบอินเทอร์เน็ตจะมีผู้ใช้ที่ใช้งานพร้อมกันเป็นจำนวนมากในช่วงเวลาที่การใช้งานมากๆ (peak time) จะทำให้เกิดปัญหาการขาดแคลนทรัพยากรได้ ด้วยเหตุนี้ เทียร์ที่สามคือแอปพลิเคชันเทียร์หรือบิสิเนสเทียร์ จึงเกิดขึ้นมาคั่นระหว่างพรีเซนเตชันเทียร์และดาต้าเบสเทียร์เพื่อเพิ่มความยืดหยุ่น และมีความสามารถในการกระจายโหลดได้มากขึ้น ยกตัวอย่างเช่นบิสิเนสเลเยอร์สามารถจัดการพูล (Pool) การเชื่อมต่อของดาต้าเบสได้ โดยจะไม่ยกเลิกการเชื่อมต่อกับดาต้าเบสเมื่อไม่ใช้งาน แต่จะเก็บลงพูลแทน ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้รองรับไคลเอนต์ได้มากขึ้น บิสิเนสเลเยอร์นั้นจะแยกไคลเอนต์ออกจากดาต้าเบส ไม่ให้ขึ้นต่อกัน โดยตรง ดังนั้นดาต้าเบสสามารถเปลี่ยนแปลงได้ โดยไม่มีผลกระทบต่อโค้ดในพีริเซนต์ชันเลเยอร์เลย บิสิเนสเลเยอร์ก็สามารถสร้าง Business Rule ที่เป็นสามัญซึ่งจะใช้ได้ในหลายๆ ไคลเอนต์แอปพลิเคชันทำให้เกิดการนำโค้ดต้นฉบับมาใช้ใหม่ ถ้าฟังก์ชันทางธุรกิจเปลี่ยนแปลงไป ก็ปรับปรุงเฉพาะบิสิเนสเลเยอร์เท่านั้น

### 3.4.1 วินโดวส์ดีเอ็นเอ (Windows DNA)

ไมโครซอฟต์สนับสนุนการกระจายแอปพลิเคชันด้วยสถาปัตยกรรมวินโดวส์ดีเอ็นเอ (Windows Distributed Internet Application Architecture: Windows DNA)

วินโดวส์ดีเอ็นเอนำเสนอแอปพลิเคชันแบบสามเทียร์ โดยอยู่บนพื้นฐานของ Component Object Model (COM) จุดประสงค์ของวินโดวส์ดีเอ็นเอ คือ การแยกลอจิกทางธุรกิจออกจากระบบไคลเอนต์/เซิร์ฟเวอร์ไปสู่วินโดวส์เทียร์ซึ่งรันอยู่บนวินโดวส์ 2000

วินโดวส์ดีเอ็นเอไม่ใช่ผลิตภัณฑ์ของบริษัทใดๆ แต่เป็นแนวทางการพัฒนาแอปพลิเคชันแบบสามเทียร์โดยทางไมโครซอฟต์ได้จัดเตรียมแพลตฟอร์ม คอมโพเนนต์เทคโนโลยี โครงสร้างภายใน แอปพลิเคชัน บริการและเครื่องมือต่างๆ ให้ หัวใจของวินโดวส์ดีเอ็นเอ นั้น คือการรวมโมเดลการเขียนโปรแกรมที่อยู่บนพื้นฐานของ COM เข้าด้วยกัน

#### 3.4.1.1 พีริเซนต์ชันเทียร์

Rich Client เป็น Win 32 แอปพลิเคชันซึ่งสามารถเข้าถึงทุก ๆ ทรัพยากรของคอมพิวเตอร์ที่เป็นไคลเอนต์ได้ Rich Client ติดต่อกับ Middle Tier โดยผ่านทาง DCOM

Thin Client จะติดต่อกับ Web Server ผ่านทาง HTTP Protocol เทคโนโลยีของไมโครซอฟต์สำหรับ Thin Client นี้คือ Web Browser (Internet Explorer)

#### 3.4.1.2 บิสิเนสโลจิกเทียร์ (Business Logic Tier)

บิสิเนสโลจิกเลเยอร์หรือ Middle Tier นั้น เป็นส่วนที่สำคัญที่สุดของสถาปัตยกรรมแบบสามเทียร์ บริการใหม่ ๆ ของไมโครซอฟต์จะอยู่ที่เลเยอร์นี้ เช่น

- Web Service โดยผ่านทาง Internet Information Server (IIS)
- Transaction และ Component Service ผ่านทาง Microsoft Transaction Server (MTS)
- Queuing และ Asynchronous Service ผ่านทาง Microsoft Message Queue (MSMQ)
- Server Side Scripting ผ่านทาง Active Server Page (ASP) ซึ่งอยู่บน IIS

1

### 3.4.1.3 ดาต้าแอ็กเซสเลเยอร์ (Data Access Layer)

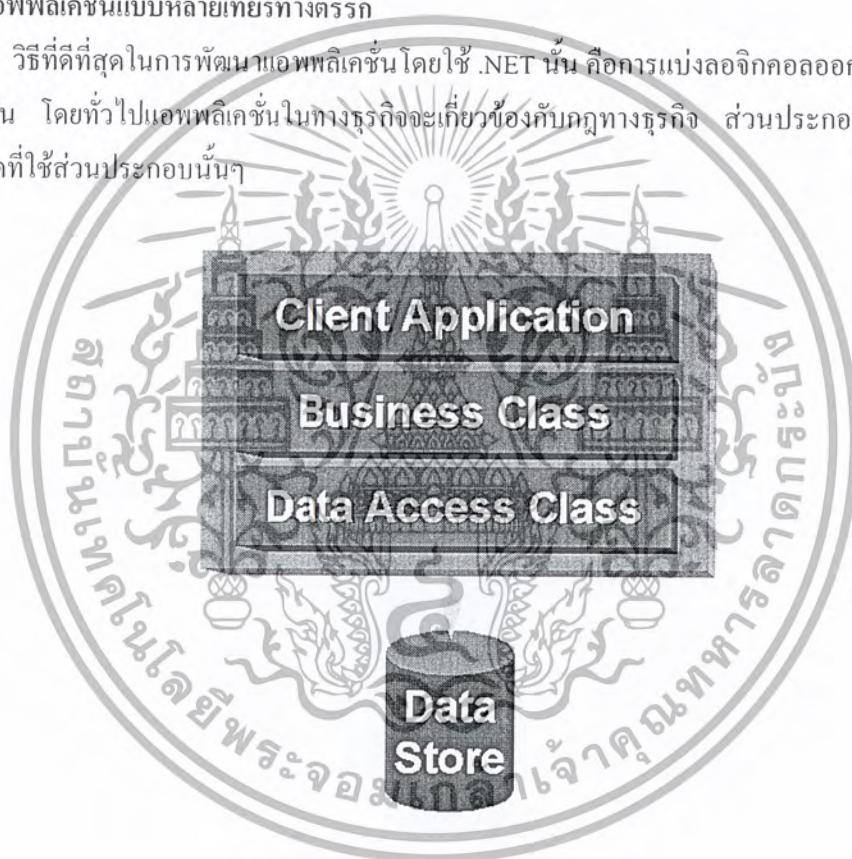
ในดาต้าแอ็กเซสเลเยอร์นี้ ไมโครซอฟต์มีผลิตภัณฑ์ที่เป็นตัวจัดการฐานข้อมูลคือ Microsoft SQL Server 2000 แต่ก็ยังกำหนด Data Access Interface ซึ่งสนับสนุนผลิตภัณฑ์อื่น ๆ ด้วย เช่น Open Database Connectivity (ODBC), OLE DB

คุณสมบัติที่สำคัญที่สุดของ Data Access Technology ของไมโครซอฟต์ ก็คือ ความสามารถในการเข้าถึงทุก ๆ แหล่งข้อมูลได้อย่างสะดวกโดยผ่าน Interface เดียวกัน ซึ่งเรียกว่า Uniform Data Access (UDA)

## 3.5 สถาปัตยกรรมแบบหลายเทียร์(N-Tier)

### 3.5.1 แอปพลิเคชันแบบหลายเทียร์ทางตรรก

วิธีที่ดีที่สุดในการพัฒนาแอปพลิเคชัน โดยใช้ .NET นั้น คือการแบ่งลอจิกคอลลอกเป็นคลาสที่ไม่ขึ้นต่อกัน โดยทั่วไปแอปพลิเคชันในทางธุรกิจจะเกี่ยวข้องกับกฎทางธุรกิจ ส่วนประกอบของชั้นข้อมูล และโค้ดที่ใช้ส่วนประกอบนั้นๆ



รูปที่ 3-3 สถาปัตยกรรมแบบ เอ็น เทียร์

สำหรับเทคนิคนี้ การใช้ลอจิกคอลลอก n-tier ในการพัฒนาแอปพลิเคชัน สามารถใช้ได้ทั้งในแอปพลิเคชันขนาดเล็ก กลาง และใหญ่ ร่วมกับเดสก์ท็อปและเว็บแอปพลิเคชัน

ทางเลือกโดยทั่วไปในการพัฒนาแอปพลิเคชัน

- สร้างยูสเซอร์อินเทอร์เฟซ (User Interface) ทั้งวินโดวส์ฟอร์ม หรือ เว็บฟอร์ม
- สร้างกฎทางธุรกิจ โดยแยกออกเป็นคลาสไลบรารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างชั้นข้อมูลโดยแยกออกเป็นคลาสไลบรารี ชั้นข้อมูลนี้จะใช้คลาสในการติดต่อไปยังตารางแต่ละตาราง เพื่อความยืดหยุ่นของคลาสค่าเจ้าท์

### ข้อดี

- การรวมกฎทางธุรกิจเป็นคอมโพเนนต์จะทำให้ง่ายต่อการสร้าง การนำไปใช้ และการนำกลับมาใช้ใหม่ ทำให้ง่ายต่อการพัฒนาและการดูแล

- สามารถใช้ภาษาระดับสูง ในการพัฒนากฎทางธุรกิจ ซึ่งตรงข้ามกับการใช้สโตรโพรซีเจอร์ (Stored procedure) และการตรวจสอบการจำกัดภาษา SQL

- รวมการเข้าถึงข้อมูลเป็นคอมโพเนนต์ ทำให้ลดการเขียนโค้ดซ้ำๆกันในแอปพลิเคชัน แต่ละฟอร์มที่ต้องการติดต่อกับฐานข้อมูลจะใช้คอมโพเนนต์ตัวเดียวกัน

- หากใช้ดาต้าเซต(DataSet) จะทำให้สะดวกต่อการค้นหาคอลัมน์โดยใช้ อินเทลลิเซนซ์ (IntelliSense) แทนที่จะต้องจำ

- รวมนิวทิน (Routines) ในการเข้าถึงข้อมูลจะช่วยให้การดูแล เนื่องจากการเปลี่ยนแปลงข้อมูลในการเข้าถึงข้อมูลจะต้องทำเพียงครั้งเดียว

- มีความยืดหยุ่นในการแบ่งสรรคอมโพเนนต์ให้กับเครื่องที่ทำงานเฉพาะอย่างได้ไม่ว่าเวลาใดก็ตาม ช่วยให้การขยายระบบให้รองรับการใช้งานได้มากขึ้นและง่ายต่อการรวบรวมโค้ด

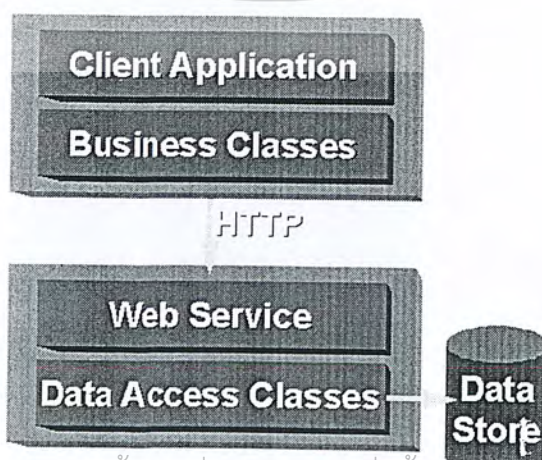
### ข้อเสีย

- การพัฒนายาวนานเนื่องจากต้องเสียเวลาในการแบ่งคอมโพเนนต์

- มีคอมโพเนนต์ที่ต้องติดตาม ทำให้การพัฒนาเมธอด (method) มีความยุ่งยากมากขึ้นเล็กน้อยที่จะทำให้นักพัฒนาใหม่จะเข้าใจ

### 3.5.2 แอปพลิเคชัน N-Tier โดยใช้ XML เว็บเซอร์วิส

สถาปัตยกรรมนี้พัฒนาการออกแบบบล็อกจิกคอล n-tier ของแอปพลิเคชันขึ้น โดยแบ่งออกเป็นหลายๆเครื่อง XML เว็บเซอร์วิสที่ติดต่อไปยังชั้นข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รูปที่ 3-4 สถาปัตยกรรมแบบ เอ็น เทียร์ ผ่านเว็บเซอร์วิส

โดยใช้ค่าเซิร์ฟเวอร์จะส่งข้อมูลกลับมาจากชั้น HTTP มายังชั้นของธุรกิจ แอปพลิเคชันทางฝั่งไคลเอนท์จะรับข้อมูลมาเพื่อใช้ในการแสดงผล

สำหรับเทคนิคนี้ การออกแบบเว็บเซอร์วิสโดยใช้ XML นั้นจะทำกับวินโดวส์ฟอร์ม แอปพลิเคชัน แต่ผู้ใช้จะต้องติดต่อไปยังเครื่องที่ต้องการและรับข้อมูลผ่านทางชั้น HTTP การเก็บกฎทางธุรกิจบนฝั่งไคลเอนท์นั้น จะทำให้เกิดปัญหาแทรฟฟิกบนเน็ตเวิร์ก แต่จะทำให้การดูแลลำบากขึ้นเมื่อมีการเปลี่ยนแปลงกฎบ่อยครั้ง

การทำงานในลักษณะนี้ จะใช้ได้กับเว็บเบสแอปพลิเคชันได้เช่นกัน โดยที่ข้อมูลจะต้องถูกเตรียมโดยเว็บเซิร์ฟเวอร์ และแสดงผลบนเว็บแอปพลิเคชัน โดยเว็บเซิร์ฟเวอร์อีกเครื่องหนึ่งทางเลือกทั่วไปในการพัฒนา

- สร้างยูสเซอร์อินเทอร์เฟซ (User Interface) ทั้งวินโดวส์ฟอร์ม หรือ เว็บฟอร์ม
- สร้างกฎทางธุรกิจโดยแยกออกเป็นคลาสไลบรารี
- สร้างชั้นข้อมูลโดยแยกออกเป็นคลาสไลบรารี ชั้นข้อมูลนี้จะใช้คลาสในการติดต่อไปยังตารางแต่ละตาราง เพื่อความยืดหยุ่นของกลาสด้าเซิร์ฟ

ข้อดี

- การรวมกฎทางธุรกิจเป็นคอมโพเนนต์จะทำให้ง่ายต่อการสร้าง การนำไปใช้ และการนำกลับมาใช้ใหม่ ทำให้ง่ายต่อการพัฒนาและการดูแล
- สามารถใช้ภาษาระดับสูง ในการพัฒนากฎทางธุรกิจ ซึ่งตรงข้ามกับการใช้สโตรโปรซีเจอร์ (Stored procedure) และการตรวจสอบการจำกัดภาษา SQL
- รวมการเข้าถึงข้อมูลเป็นคอมโพเนนต์ ทำให้ลดการเขียนโค้ดซ้ำๆกันในแอปพลิเคชัน แต่ละฟอร์มที่ต้องการติดต่อก็พื้นฐานข้อมูลจะใช้คอมโพเนนต์ตัวเดียวกัน
- หากใช้ค่าเซิร์ฟเวอร์ จะทำให้สะดวกต่อการค้นหาคลัสน์โดยใช้ อินเทลลิเซนซ์ (IntelliSense) แทนที่จะต้องจำ
- รวมรูทีน (Routines) ในการเข้าถึงข้อมูลจะช่วยในการดูแล เนื่องจากการเปลี่ยนแปลงข้อมูลในการเข้าถึงข้อมูลจะต้องทำเพียงครั้งเดียว
- มีความยืดหยุ่นในการแบ่งสรรคอมโพเนนต์ให้กับเครื่องที่ทำงานเฉพาะอย่างได้ไม่ว่าเวลาใดก็ตาม ช่วยให้การขยายระบบให้รองรับการใช้งานได้มากขึ้นและง่ายต่อการรวบรวมโค้ด
- ชั้นการเข้าถึงข้อมูลเป็นศูนย์กลางในการเข้าถึงข้อมูล
- สามารถขยายระบบให้รองรับการใช้งานได้มากขึ้น โดยการเพิ่มเว็บฟอร์มเพื่อรองรับการร้องขอเพื่อเข้าถึงฐานข้อมูลของผู้ใช้

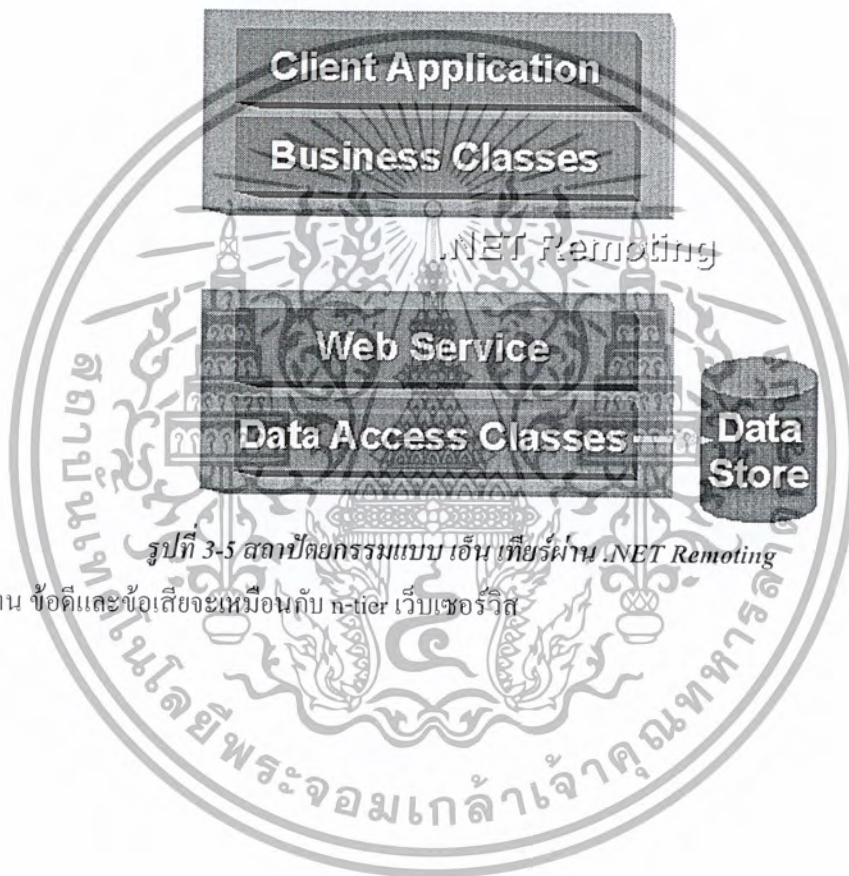
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผู้ใช้สามารถติดต่อผ่านทางอินเทอร์เน็ตและยังคงเข้าถึงข้อมูลได้จากเดสก์ท็อปแอปพลิเคชันหรือเว็บแอปพลิเคชัน

#### ข้อเสีย

- การพัฒนายาวนานเนื่องจากต้องเสียเวลาในการแบ่งคอมโพเนนต์
- มีคอมโพเนนต์ที่ต้องติดตาม ทำให้การพัฒนาเมธอด (method) มีความยุ่งยากมากขึ้นเล็กน้อยที่จะทำให้นักพัฒนาใหม่จะเข้าใจ

### 3.5.3 แอปพลิเคชัน N-Tier โดยใช้รีโมต .NET



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และการออกแบบระบบ เป็นส่วนสำคัญในการทำโครงการ การวิเคราะห์และการออกแบบที่ดี จะทำให้การสร้างระบบ (Implement) เป็นไปอย่างถูกต้องตามความต้องการและรวดเร็ว สำหรับโครงการนี้ จะใช้เทคโนโลยีของ ดอทเน็ตเทคโนโลยี (.Net Technology) ซึ่งจะมีการเขียนโปรแกรม (Programming) เป็นแบบออบเจกต์ โอเรียนเต็ล (Object Oriented) ซึ่งเป็นการมองระบบเป็นวัตถุ ซึ่งทำให้มีการมองระบบได้อย่างชัดเจนมากขึ้น ส่วนภาษาที่ใช้ในการช่วยวิเคราะห์และการออกแบบระบบ คือ ภาษา UML ซึ่งเป็นภาษาที่ช่วยสร้างไดอะแกรม (Diagram) ต่าง ๆ ในการออกแบบระบบ ซึ่งทำให้การออกแบบมีมาตรฐาน และง่ายต่อความเข้าใจของผู้พัฒนาระบบต่อไปอีกด้วย

#### 4.1 การวิเคราะห์ความต้องการของระบบ

การวิเคราะห์ระบบ เป็นการรวบรวมรายละเอียด และความต้องการต่าง ๆ ที่ได้ศึกษามา มาทำการรวบรวมและสรุปเป็นความต้องการของระบบ ซึ่งในขั้นตอนการวิเคราะห์นี้ จะใช้ยูสเคส ไดอะแกรม (Use Case Diagram) เป็นไดอะแกรม แสดงความต้องการของระบบ ซึ่งเป็นหน้าที่การทำงานต่างๆ ที่ต้องมีในระบบ ที่ได้มาจากการศึกษาและวิเคราะห์มา

##### 4.1.1 หลักการ

ระบบการจัดการการเข้าถึงเอกสารเอกซอแมล จะกระทำผ่านโปรแกรมเว็บเบราว์เซอร์ (Web Broeser) ที่ทำการเชื่อมต่อกับเว็บไซต์ (Web Site) โดยที่โปรแกรมระบบการจัดการการเข้าถึงเอกสารเอกซอแมล ซึ่งอยู่บนเครื่องเว็บเซิร์ฟเวอร์ (Web Server) จะเป็นผู้จัดการการเข้าถึงเอกสารทั้งหมด โดยที่มีผู้ใช้ที่มีระดับการเข้าถึงที่ต่างกันก็จะมีการดูเอกสารที่ต่างกัน โดยระบบนี้แบ่งออกเป็น 4 ระดับ คือ บุคคลทั่วไป (Public), สมาชิก(Member), ผู้จัดการ (Manager), ผู้ดูแลระบบ (Administrator) โดยที่ผู้ดูแลระบบจะเป็นคนกำหนดกฎการเข้าถึง (Access Rule), กำหนดกลุ่มผู้ใช้ และ จัดการฐานข้อมูล (DataBase)

##### 4.1.1.1 แนวคิดการทำงานของระบบการเข้าถึงเอกสารเอกซอแมล

การทำงาน XML ACCESS CONTROL SYSTEM จะทำหน้าที่ในการควบคุมสิทธิการเข้าถึงข้อมูลใดๆ ในหลักการจะแบ่งออกเป็น 4 ส่วนคือ (Subject, Object, Action, Permission)

##### 4.1.1.1.1 Subject

คือ จะถูกกำหนดขึ้นมาโดยอาจจะหมายถึง ตัวบุคคล กลุ่ม เช่น Tony, Room7

##### 4.1.1.1.2 Object

คือ จุดหมายของเอกสารเอกซอแมลที่ Subject ต้องการจะเข้าไปถึงโดยเราจะระบุคำนี้ด้วย Single Xpath expression ตัวอย่างเช่น ถ้าเรากำหนด read Xpath ไปที่ node ใด node หนึ่ง child ของ node นั้น เราก็สามารถ read ขึ้นมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.1.3 Action

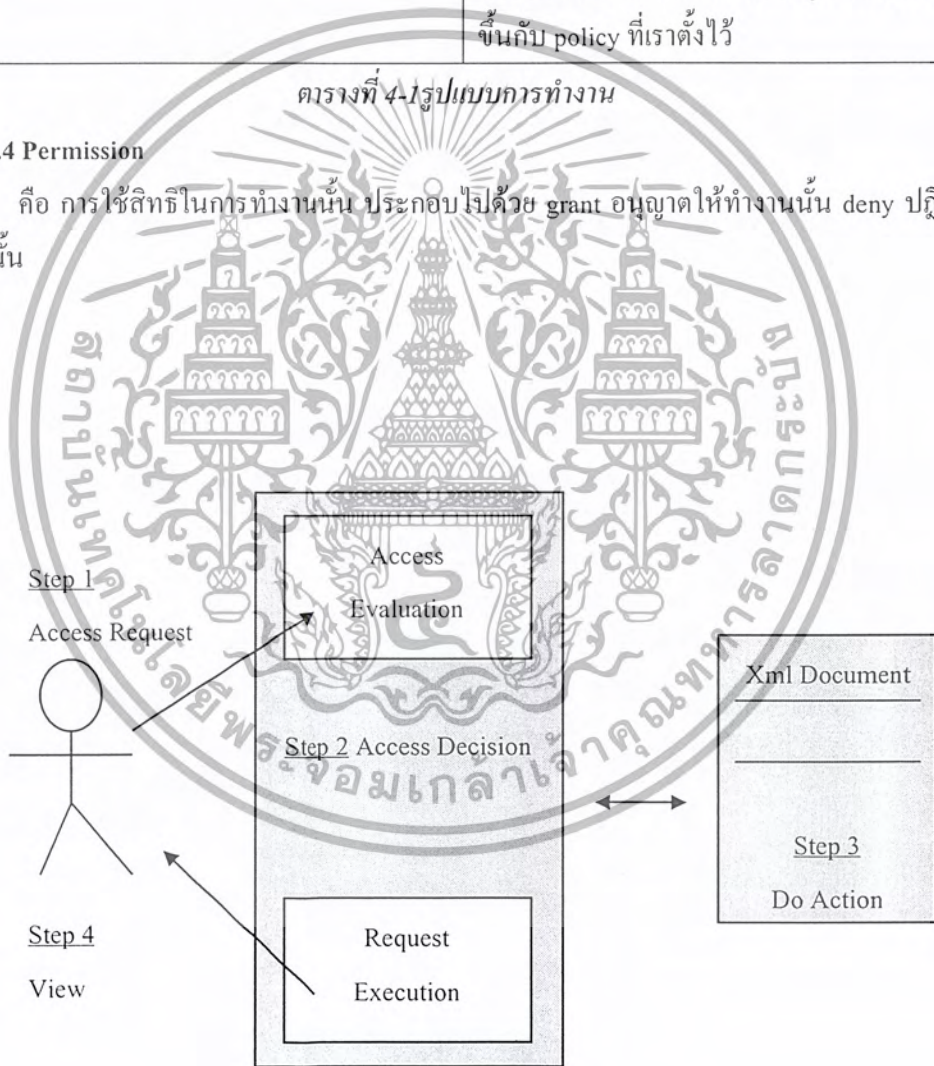
คือการที่ Subject ต้องการกระทำใดๆยัง Object โดยมี set ของ Action ประกอบไปด้วย ( read , write )

Type	Semantic
read	การอ่านค่าจากต้นไม้ข้อมูล(tree)โดยการอ่านขึ้นกับ policy ที่เราตั้งไว้
write	การเขียนแก้ไขค่าจากต้นไม้ข้อมูล โดยการแก้ไขขึ้นกับ policy ที่เราตั้งไว้

ตารางที่ 4-1 รูปแบบการทำงาน

4.1.1.1.4 Permission

คือ การใช้สิทธิในการทำงานนั้น ประกอบไปด้วย grant อนุญาตให้ทำงานนั้น deny ปฏิเสธการทำงานนั้น



รูปที่ 4-1 แนวคิดของระบบการเข้าถึงเอกสารเอกซเอมแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step1	Access Request	การที่ user กำหนด target XML Document ที่ต้องการเข้าไปถึง และ Action ที่ต้องการทำยัง target นั้น
Step2	Access Decision	หลังจากผ่านในส่วน Access Evaluation เราก็สามารถรู้ได้ว่า Action ที่เข้ามาได้หรือไม่โดยการไป lookup rule table แล้วจะรู้ค่า permission ของ Action นั้น
Step3	Do Action	ทำตาม Action ที่กำหนดจาก rule ที่ได้
Step4	View	การแสดงผลการเข้าถึงที่ Subject จะมีสิทธิข้อมูล โดยแสดงผลออกมา

#### ตารางที่ 4-2 ขั้นตอนการทำงานของระบบการเข้าถึงเอกสารเอกซเอ็มแอล

##### 4.1.1.2 การออกแบบกฎการเข้าถึง

ถือได้ว่าเป็นหัวใจสำคัญของระบบ เนื่องจากเป้าหมายของโครงการคือต้องการแยกแยะบุคคลที่จะเข้าถึงข้อมูลในระดับต่าง ๆ ตั้งแต่ระดับตารางไปจนถึงระดับฟิลด์ (field) ดังนั้นแนวคิดในการออกแบบส่วนนี้ออกแบบให้ได้ตามต้องการและถูกต้อง

โดยการแนวคิดเราจะมี ตารางข้อมูลเก็บค่าของกฎการเข้าถึงเอาไว้แล้วจะมีตารางข้อมูลผู้ใช้ เราจะนำข้อมูลมาเปรียบเทียบว่าผู้ใช้นั้นมีสิทธิเข้าถึงเอกสารแค่ไหน และมีสิทธิทำอะไรกับเอกสารบ้าง

##### 4.1.1.2.1 แนวคิดในการออกแบบคือ

- การจัดลำดับของแอคเซสรูล แอคเซสรูลที่มีผลกระทบที่มีทรานซิติฟมากกว่าควร อยู่ ลำดับต้นๆ
- หลีกเลี่ยงการขัดแย้งของแอคเซสรูล การขัดแย้งของแอคเซสรูล คือ การที่แอคเซสรูล 2 รูล ที่มีเงื่อนไขเหมือนกัน แต่ permission ต่างกัน ทำให้เกิดความขัดแย้งกัน
- หลีกเลี่ยงการซ้ำซ้อนของแอคเซสรูล การซ้ำซ้อนของแอคเซสรูลทำให้เปลืองเนื้อที่และเวลาในการ Access
- กำหนดแอคเซสรูลให้มีจำนวนน้อยที่สุดและครอบคลุมที่สุดเพื่อความรวดเร็วในการ Access
- จัดกลุ่มและลำดับของแอคเซสรูลให้เหมาะสมจัดลำดับของการ Access เฉพาะเจาะจงว่าให้อยู่ในระดับต้น เพื่อให้ความสำคัญของ Access ที่เฉพาะเจาะจงกว่า
- ตารางจะเรียงตาม Permission โดยให้ deny มีลำดับความสำคัญ (Priority) สูงที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RuleId	Subject	Object	Action	Permission

ตารางที่ 4-3 Access Rule

RuleId	คือ จำนวน Rule ที่มีในระบบ
Subject	คือ ตัวบุคคลหรือกลุ่มที่เข้ามา
Object	คือ Xpath ที่ต้องการเข้าถึง
Action	คือ การกระทำที่ต้องการทำประกอบด้วย read, write
Permission	คือ การอนุญาตให้กระทำ Action นั้นหรือไม่ ประกอบด้วย grant, deny

ตารางที่ 4-4 ความหมายของแต่ละหลักของตาราง Access Rule

#### 4.1.1.2.2 ตัวอย่างการทำงาน



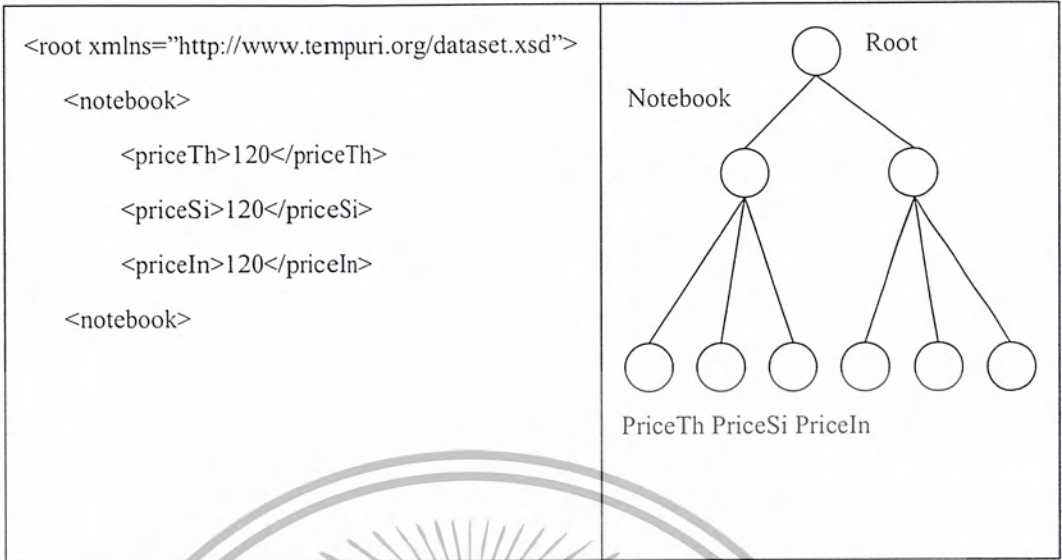
RuleID	Subject	Object	Action	Permission
1	min	/mstns:root/mstns:notebook/mstns:PriceSi	Read	deny
2	Min	/mstns:root	Read	Grant

รูปที่ 4-2 ขั้นตอนการทำงานของ Access Rule

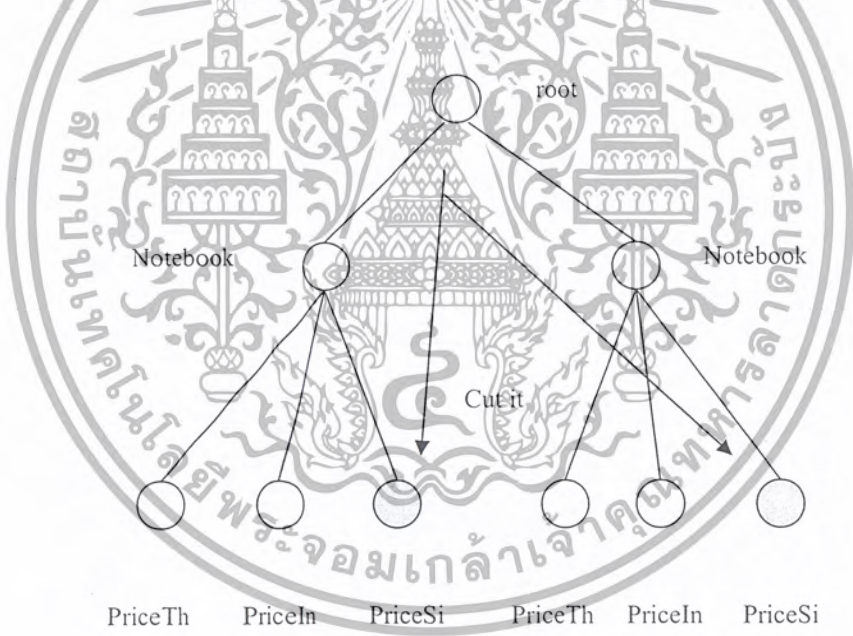
ขั้นตอนการทำงาน	การทำงาน
1 command	คำสั่งที่เข้ามาโดนมีอินพุต(Input) เป็น Subject, Object, Action
2 look up rule table	ไปค้นหาในตาราง Access Rule ว่า permission คืออะไร
3 return permission	ส่งค่า permission กลับไป

ตารางที่ 4-5 ความหมายขั้นตอนการทำงานของ Access Rule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-3 เอกสารเอกซเอมแอลและโครงสร้างต้นไม้ของเอกสาร



- Not cut node
- Cut node

รูปที่ 4-4 แสดงการทำงานกันโครงสร้างต้นไม้

หลังจากได้ค่า permission มากี่จะทำการตัดส่วนที่ไม่สามารถเข้าถึงได้ เช่นเดียวกันหากเป็นการ write ก็

เอกสารนี้เป็นเอกสารต้นฉบับสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<root xmlns="http://www.tempuri.org/dataset.xsd">
  <notebook>
    <priceTh>120</priceTh>
    <priceIn>120</priceIn>
  </notebook>

```

รูปที่ 4-5 เอกสารเอกซเมมแอลคั้งจากทำการตรวจสอบสิทธิ

#### 4.1.2 ขอบเขตการให้บริการ

ระบบการจัดการการเข้าถึงเอกสารเอกซเมมแอล จะนำเสนอผ่านเว็บไซต์ขายคอมพิวเตอร์ โดยจะให้บริการบุคคลทั่วไปที่ต้องการดูข้อมูลทั่วไปของราคาคอมพิวเตอร์หรือตามที่คุณผู้ดูแลระบบตั้งค่าไว้ ซึ่งใครก็สามารถเข้าชมข้อมูลของบุคคลทั่วไปได้ แต่สำหรับสมาชิกและผู้จัดการจะต้องมีการพิสูจน์ตัวตน (Authentication) ก่อนเข้าถึงข้อมูลของบุคคลนั้น ซึ่งก็จะถูกตั้งค่าโดยผู้ดูแลระบบเช่นเดียวกัน ส่วนการบริการจะแบ่งออกเป็น 2 รูปแบบใหญ่คือ การเข้าถึงข้อมูล (Access Data) การแก้ไขข้อมูล (Update Data) โดยแบ่งออกเป็น 4 ระดับคือ

##### 4.1.2.1 บุคคลทั่วไป

จะสามารถแค่การเข้าถึงข้อมูล

##### 4.1.2.2 สมาชิก

ก็มีความสามารถแค่เข้าถึงข้อมูลเช่นเดียวกันแต่จะมีสิทธิที่เข้าถึงได้มากกว่า

##### 4.1.2.3 ผู้จัดการ

ก็สามารถเข้าถึงข้อมูลและยังสามารถแก้ไขข้อมูลตามสิทธิที่ได้กำหนดไว้

##### 4.1.2.4 ผู้ดูแลระบบ

มีความสามารถทั้งการเข้าถึงข้อมูลและการแก้ไขข้อมูล โดยมีสิทธิสูงสุดสามารถกำหนดข้อมูล (Data) กำหนดกฎการเข้าถึง (Rule)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.3 Use Case Diagram

แผนภาพยูสเคส แสดงฟังก์ชันการทำงานที่สำคัญ ๆ ของระบบ ซึ่งมีรายละเอียด ดังนี้

- Actor เป็นผู้ใช้ที่มีส่วนเกี่ยวข้องกับการเข้าใช้ระบบ ประกอบไปด้วย

ชื่อ	รายละเอียด
public	ผู้ใช้ทั่วไปที่ไม่ได้ลงทะเบียนเป็นสมาชิกของเว็บไซต์ขายคอมพิวเตอร์
member	เป็นสมาชิกของเว็บไซต์ขายคอมพิวเตอร์
manager	เป็นผู้จัดการของเว็บไซต์ขายคอมพิวเตอร์ สามารถแก้ไขข้อมูลได้
admin	มีความสามารถทั้งการเข้าถึงข้อมูลและการแก้ไขข้อมูล โดยมีสิทธิสูงสุด สามารถกำหนดข้อมูลกำหนดกฎการเข้าถึง

ตารางที่ 4-6 Actor ของระบบ

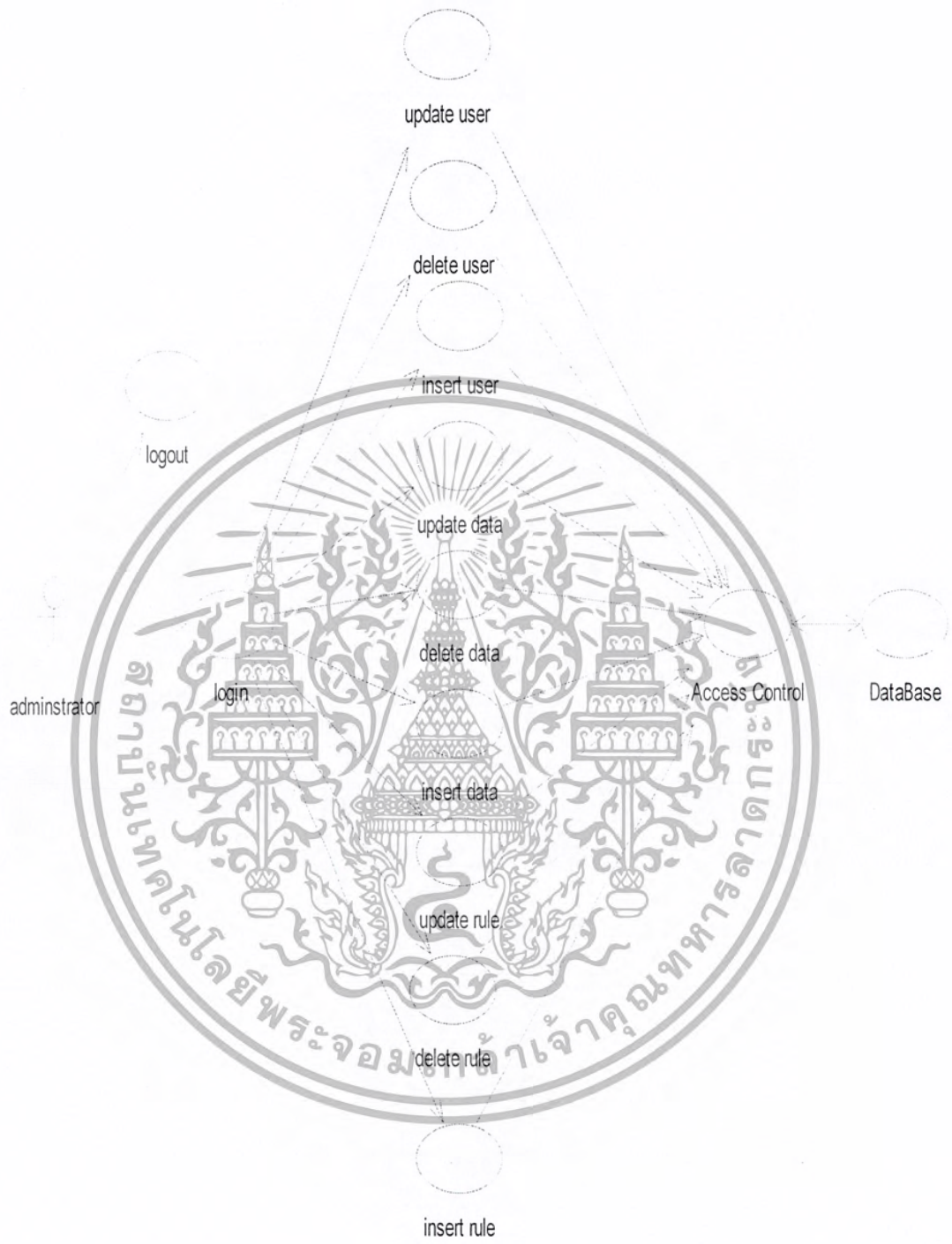
- Use Case แสดงถึง ฟังก์ชันการทำงานหลักที่มีในระบบ ซึ่งประกอบไปด้วย

ชื่อ	รายละเอียด
login	การ login ของสมาชิก,ผู้จัดการ,ผู้ดูแลระบบ เข้าสู่ระบบ
logout	การ logout ของสมาชิก,ผู้จัดการ,ผู้ดูแลระบบ ออกจากระบบ
Access Control	ส่วนการพิสูจน์ตัวตนและพิสูจน์สิทธิในการเข้าระบบและเข้าถึงข้อมูล
View Data	ส่วนการแสดงผล
*update data	ผู้ดูแลระบบหรือผู้จัดการ ทำการแก้ไขข้อมูล
*delete data	ผู้ดูแลระบบ ทำการลบข้อมูล
*insert data	ผู้ดูแลระบบ ทำการเพิ่มข้อมูล
*update rule	ผู้ดูแลระบบ นำการแก้ไขกฎการเข้าถึง
*delete rule	ผู้ดูแลระบบ ทำการแก้ไขกฎการเข้าถึง
*insert rule	ผู้ดูแลระบบ ทำการเพิ่มกฎการเข้าถึง
*update user	สมาชิกหรือผู้จัดการ ทำการแก้ไขข้อมูลส่วนตัว ผู้ดูแลระบบทำการแก้ไขข้อมูลของผู้ใช้ระบบ
*delete user	ผู้ดูแลระบบทำการลบข้อมูลของผู้ใช้ระบบ
*insert user	ส่วนตัว ผู้ดูแลระบบทำการลบของผู้ใช้ระบบ
Database	ส่วนที่ทำการติดต่อกับดาตาเบส เซิร์ฟเวอร์ (DataBase Server)

ตารางที่ 4-7 Use Case ของระบบ

**หมายเหตุ \*** สิทธิเหล่านี้อาจมีการเปลี่ยนแปลงได้ หากมีการแก้ไขการเข้าถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 แผนภาพ Use Case ของ ผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

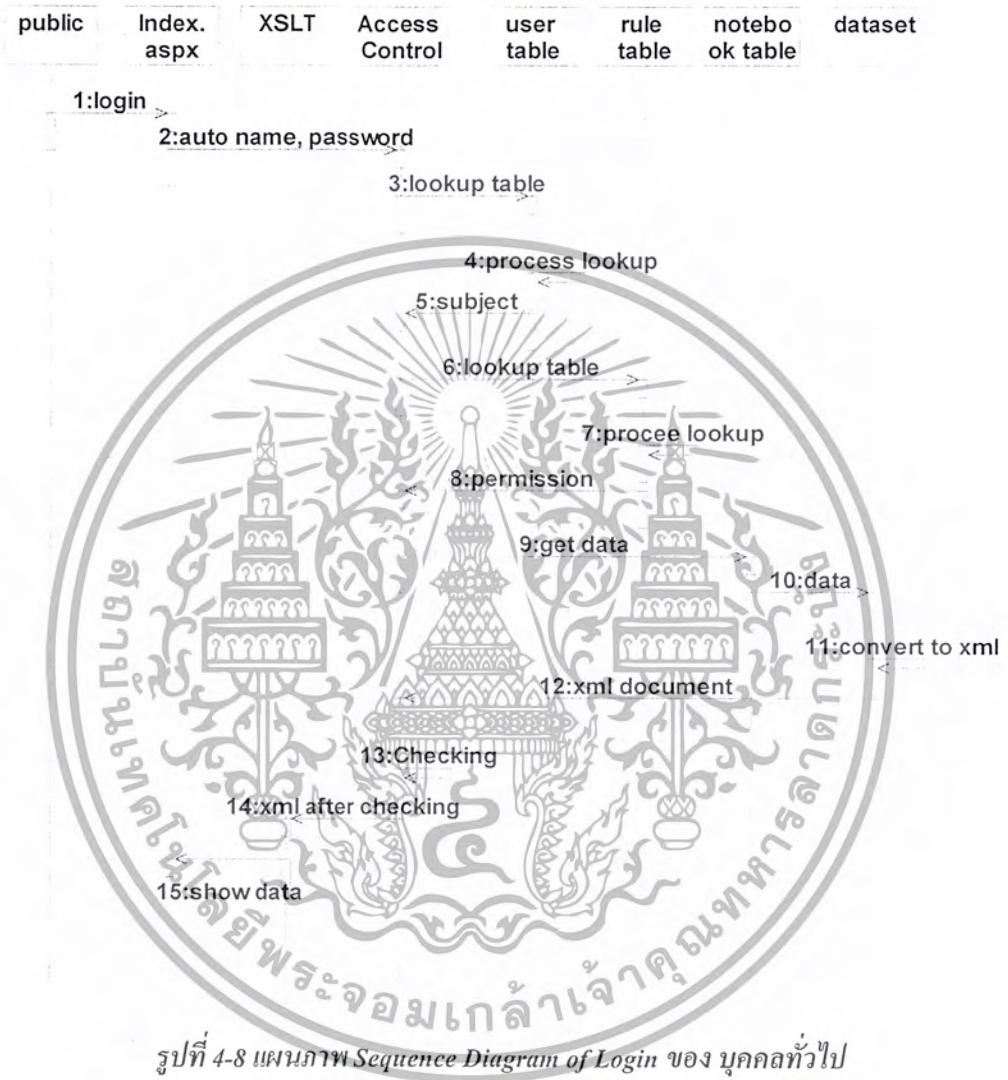


รูปที่ 4-7 แผนภาพ Use Case ของ สมาชิก, บุคคลทั่วไป, ผู้จัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.3.1 การล็อกอิน (Log In)

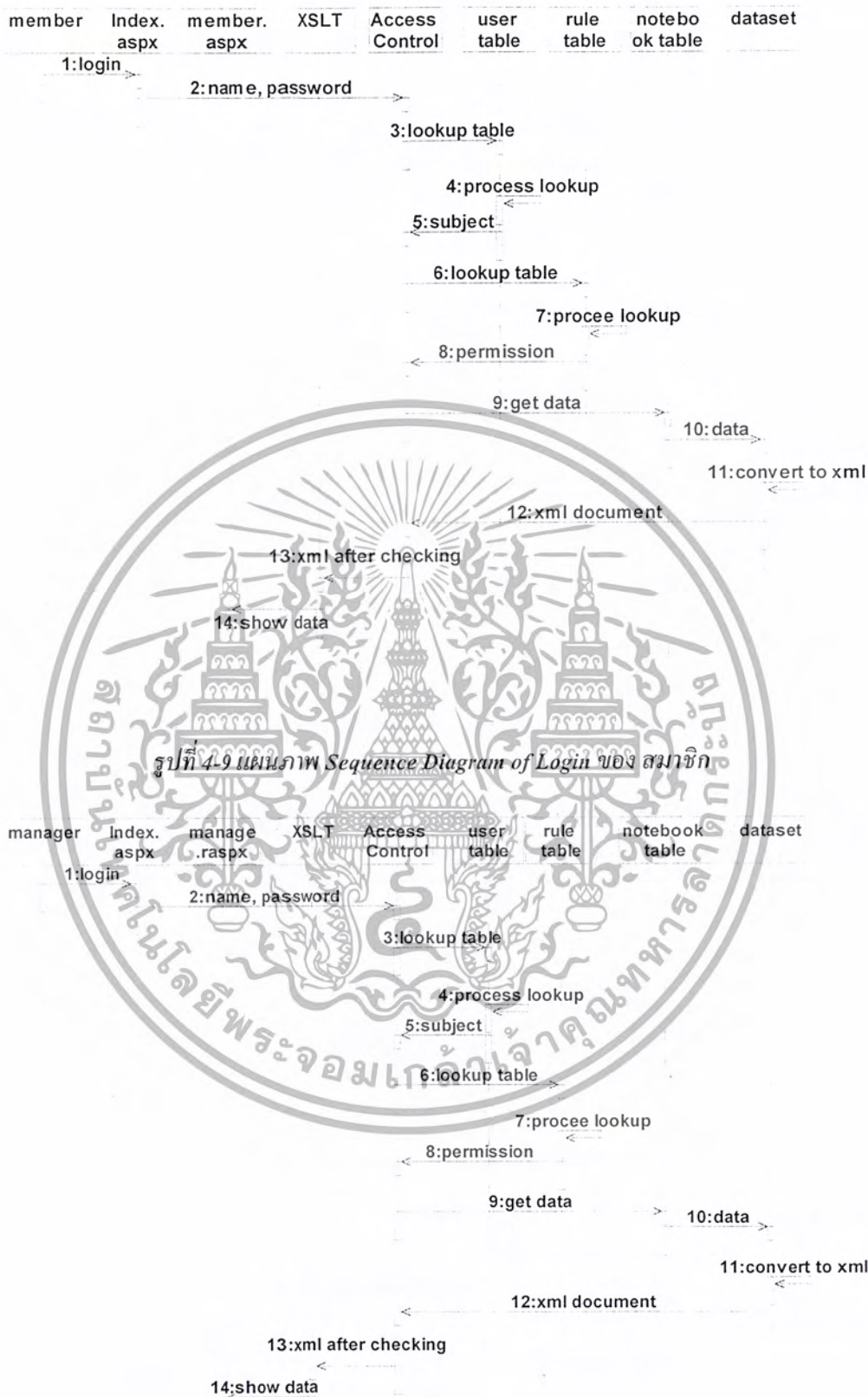
เป็นการเข้าสู่ระบบโดยค่าเริ่มต้นของทุกคนจะเป็นแบบบุคคลทั่วไปโดยมีการล็อกอินแบบอัตโนมัติเข้าที่หน้าเว็บเพจแรกที่ชื่อว่าอินเด็กซ์ (Index.aspx) โดยไม่ต้องมี Username และ Password



รูปที่ 4-8 แผนภาพ Sequence Diagram of Login ของ บุคคลทั่วไป

ในส่วนของผู้ใช้และผู้จัดการจะต้องมีการส่ง Username และ Password ไปที่เว็บเซิร์ฟเวอร์ เพื่อทำการพิสูจน์ตัวตนและพิสูจน์สิทธิการเข้าถึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

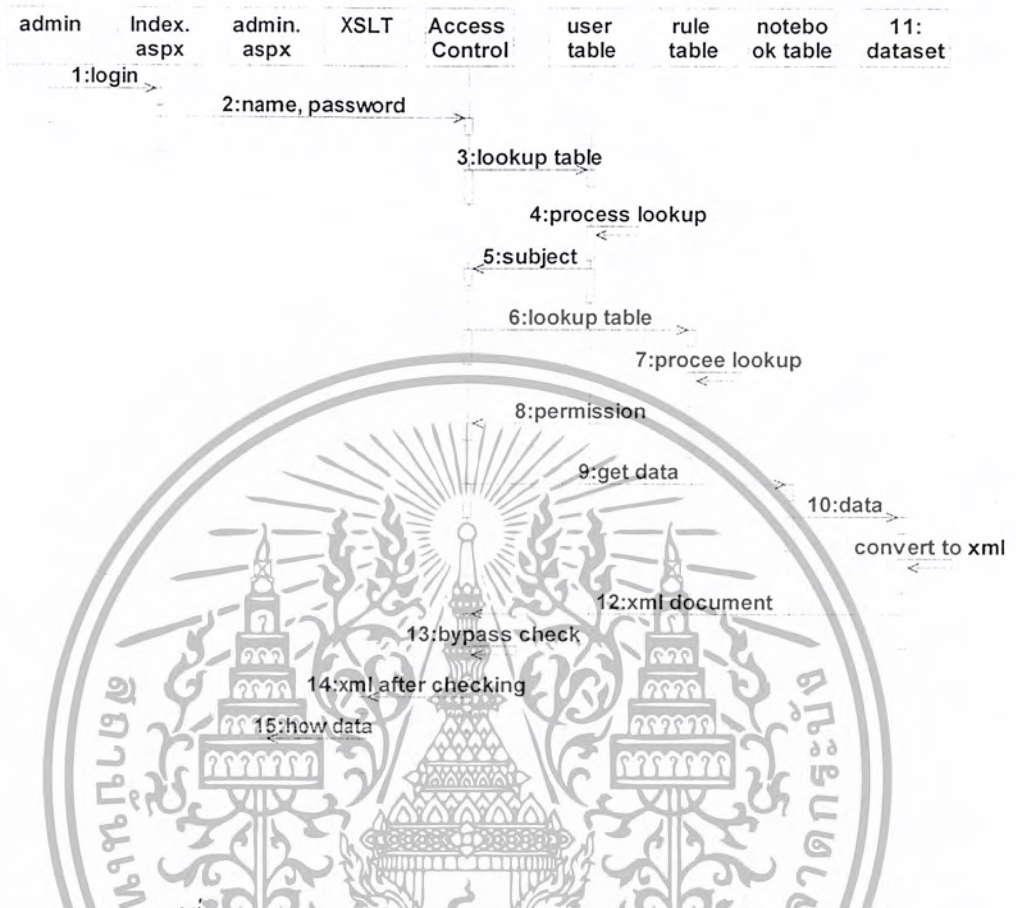


รูปที่ 4-9 แผนภาพ Sequence Diagram of Login ของ สมาชิก

รูปที่ 4-10 แผนภาพ Sequence Diagram of Login ของ ผู้จัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4-10 แผนภาพ Sequence Diagram of Login ของผู้จัดการไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

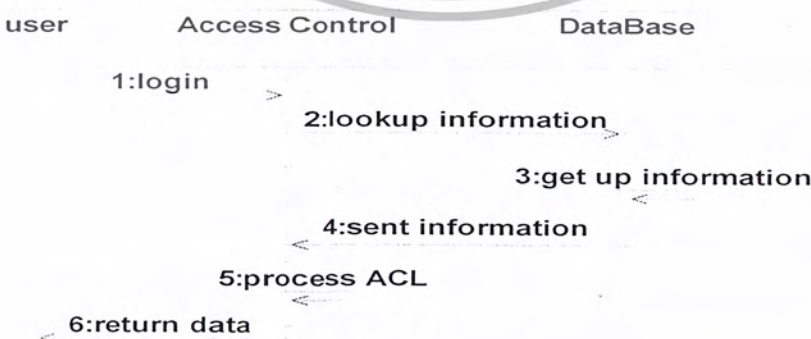
ในการล็อกอินของผู้ดูแลระบบจะต่างจากผู้ใช้คนอื่นคือจะมีการพิสูจน์ตัวตนเท่านั้นจะไม่มี การพิสูจน์สิทธิ โดยถือว่ามีสิทธิการเข้าถึงทุกอย่าง



รูปที่ 4-11 แผนภาพ Sequence Diagram of Login ของ ผู้ดูแลระบบ

#### 4.1.3.2 การควบคุมการเข้าถึง (Access Control)

Access Control จะทำการตรวจสอบข้อมูล (Information) ต่างๆที่ส่งเข้ามาโดยตรวจสอบกับฐานข้อมูล ทั้งการพิสูจน์ตัวตนและพิสูจน์สิทธิ

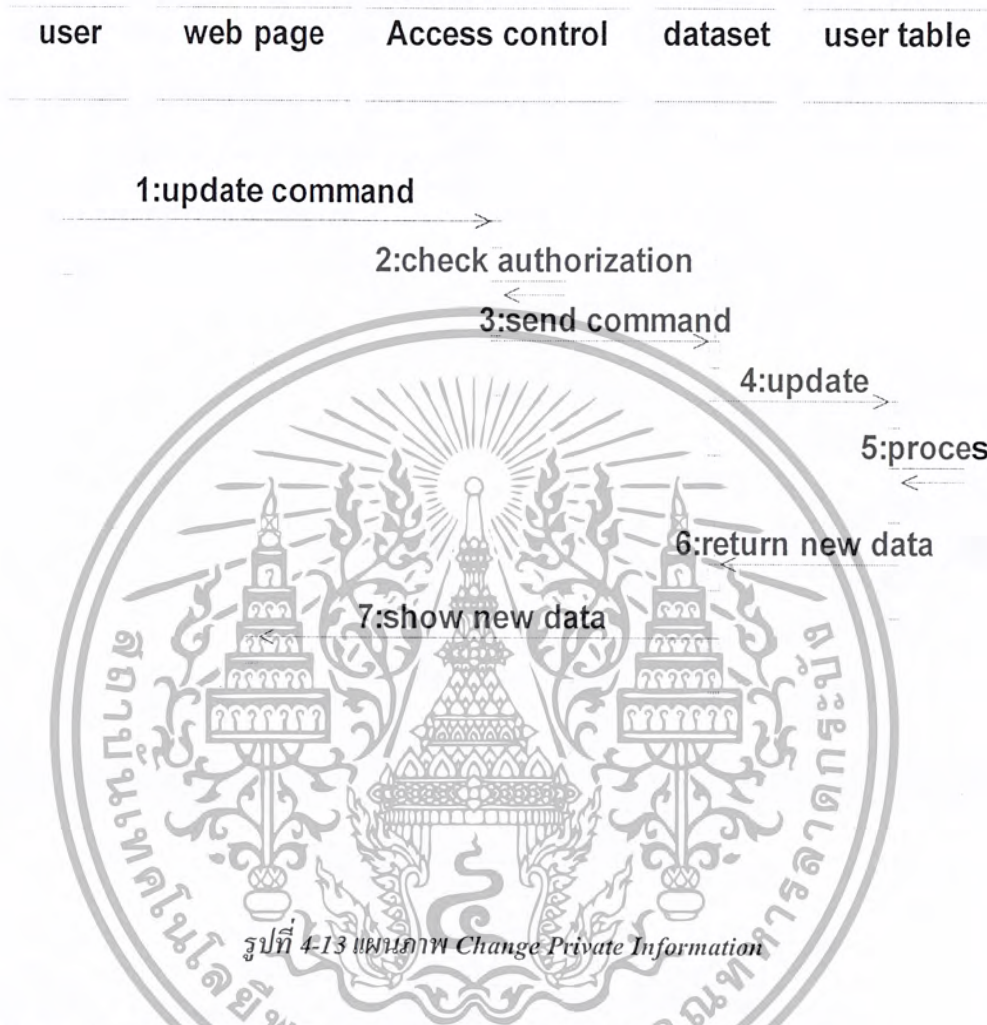


รูปที่ 4-12 แผนภาพ Sequence Diagram of Access Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในองค์กรหรือหน่วยงานนั้นๆ ข้อมูลทั้งหมดในเอกสารนี้จะไม่ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.3.3 การแก้ไขข้อมูลส่วนตัว (Change Private Information)

ผู้ใช้งานระบบมีสิทธิที่จะแก้ไขข้อมูลส่วนตัว 2 ส่วนด้วยกันคือ Username และ Password



#### 4.1.3.4 การแก้ไขข้อมูลของผู้ดูแลระบบ Admin Change Information

ผู้ดูแลระบบสามารถเข้าแก้ไขข้อมูลเพิ่มข้อมูลหรือลบข้อมูลได้ทั้ง 3 ส่วนด้วยกัน ประกอบด้วย

##### 4.1.3.4.1 การแก้ไขข้อมูลส่วนตัวข้อมูลผู้ใช้

คือข้อมูลของสมาชิกหรือผู้จัดการที่มีในระบบ

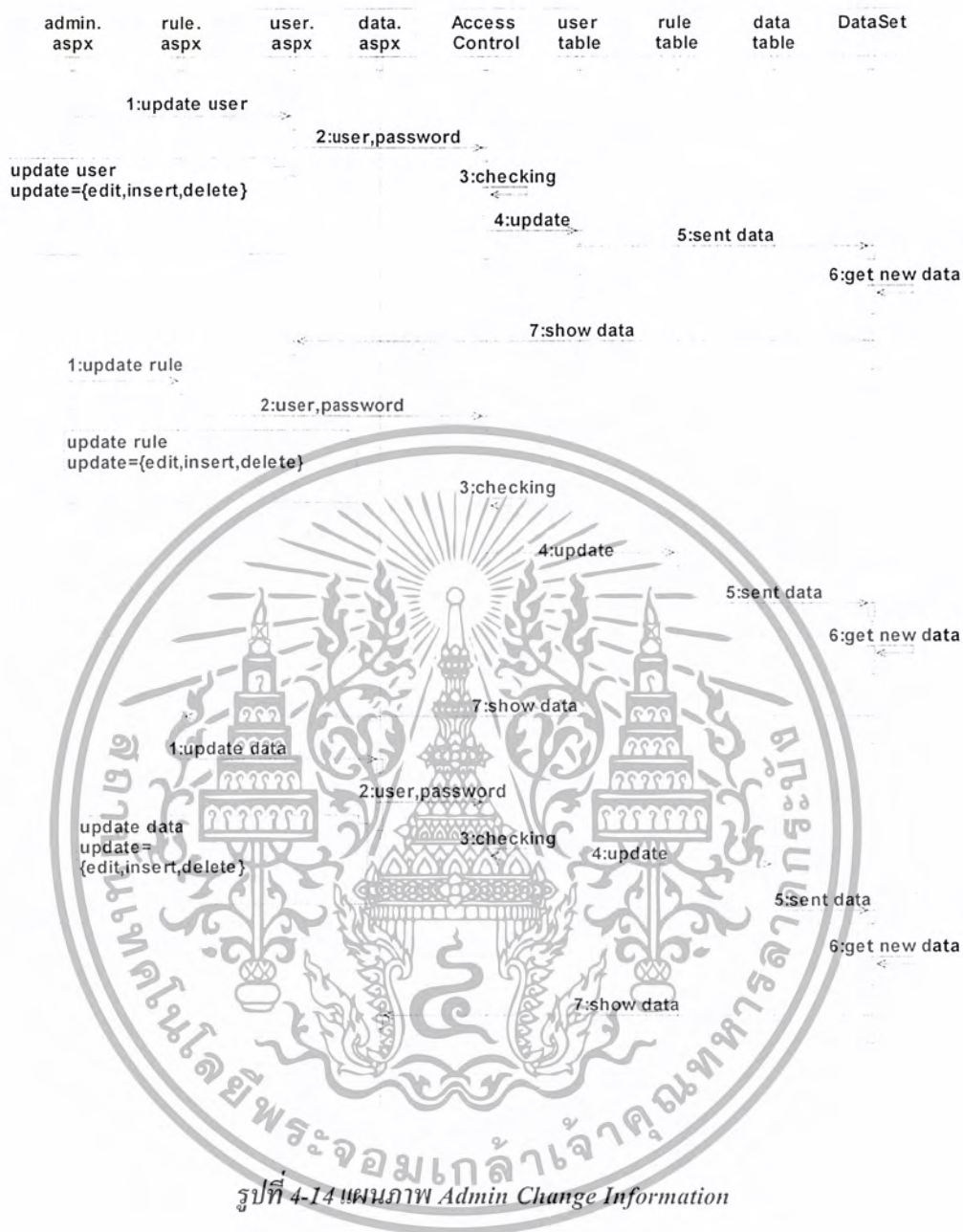
##### 4.1.3.4.2 การแก้ไขข้อมูลส่วนกฎการเข้าถึง

คือข้อมูลของกฎการเข้าถึงของสมาชิกที่มีในส่วนของข้อมูลผู้ใช้

##### 4.1.3.4.3 การแก้ไขข้อมูลส่วนฐานข้อมูล

คือข้อมูลของสินค้าที่มีอยู่ในร้าน

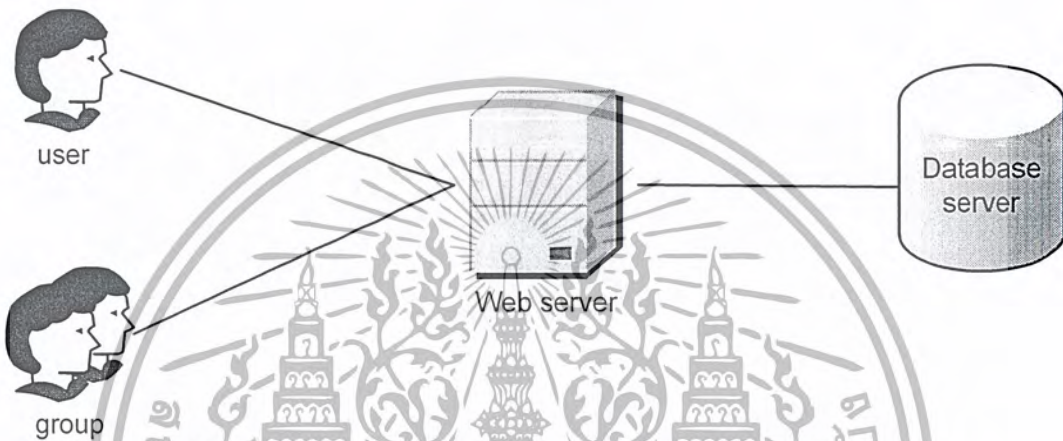
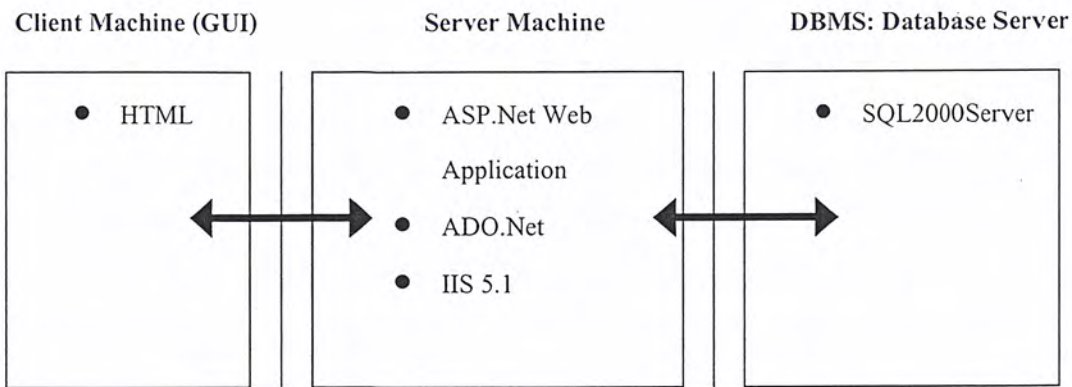
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



#### 4.2 การออกแบบระดับสถาปัตยกรรม (Architectural Design)

การออกแบบระดับสถาปัตยกรรม เป็นการกำหนดกลุ่มการทำงานของระบบ ซึ่งเกี่ยวข้องกับโครงสร้างฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) โดยทั่วไป การออกแบบสถาปัตยกรรมของฮาร์ดแวร์ จะเป็นเครื่องบ่งชี้โครงสร้างของสถาปัตยกรรมซอฟต์แวร์ ซึ่งการที่สถาปัตยกรรมจะทำงานได้อย่างดีนั้น ต้องขึ้นอยู่กับทั้งฮาร์ดแวร์ และซอฟต์แวร์ ดังนั้น การออกแบบจึงต้องคำนึงถึงการทำงานร่วมกันด้วย เพื่อให้ระบบมีการทำงานที่ถูกต้องตามต้องการ, มีประสิทธิภาพอย่างน้อยที่สุดในระดับที่ยอมรับได้, ระบบทำงานได้อย่างปลอดภัย, เชื่อถือได้, มีค่าใช้จ่ายที่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-15 สถาปัตยกรรมของระบบการเข้าถึงเอกสารเอกซเอมเอลด

สถาปัตยกรรมของระบบของระบบการเข้าถึงเอกสารเอกซเอมเอลดผ่านเว็บเพจ จะใช้เทคโนโลยีสถาปัตยกรรมแบบ ทรีเทียร์ (3-tier) แบ่งได้ 3 ส่วนคือ ส่วนไคลเอ็นต์ (Client), ส่วนเว็บเซิร์ฟเวอร์ และส่วนดาต้าเบสเซิร์ฟเวอร์

4.2.1 ส่วนไคลเอ็นต์ (Client)

ส่วนของไคลเอ็นต์ ซึ่งเป็นส่วนของผู้ใช้บริการเว็บไซต์ ซึ่งเป็นอินเทอร์เฟซ (Interface) ที่ใช้ติดต่อกับผู้ใช้ ผู้ใช้จะใช้เว็บเบราว์เซอร์ (Web Browser) ในการแสดงหน้าเว็บไซต์ โดยการอ่านแท็ก (Tag) ของภาษา HTML แล้วแสดงเป็นหน้าเว็บไซต์ออกมา ตัวอย่างของเว็บเบราว์เซอร์ ได้แก่ Internet Explorer, Netscape เป็นต้น

4.2.2 ส่วนเว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์ เป็นเครื่องคอมพิวเตอร์ที่คอยให้บริการการขายเครื่องคอมพิวเตอร์โดยกระทำผ่านโปรโตคอลมาตรฐาน HTTP (Hypertext Transfer Protocol)

ในส่วนของเว็บเซิร์ฟเวอร์ ระบบจะใช้ไอไอเอส เวอร์ชัน 4.1 (IIS version 4.1) เป็นเว็บ เซิร์ฟเวอร์ซึ่งสามารถทำการรัน เอเอสพีคอตเน็ต (ASP.Net) ซึ่งเป็นภาษาที่ใช้เขียนติดต่อรหว่างเว็บเซิร์ฟเวอร์กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเบราว์เซอร์ของไคลเอ็นต์ได้ และมี (ADO.Net : Active Data Object) เป็นตัวที่ใช้ติดต่อระหว่างเว็บเซิร์ฟเวอร์กับดาต้าเบสเซิร์ฟเวอร์

#### 4.2.3 ส่วนของดาต้าเบสเซิร์ฟเวอร์

ดาต้าเบสเซิร์ฟเวอร์ เป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่เก็บข้อมูล (Data) ต่าง ๆ และข้อมูลที่เก็บไว้เพื่อเป็นข้อมูลของระบบ ดาต้าเบสเซิร์ฟเวอร์เป็นส่วนที่ให้บริการข้อมูลแก่เครื่องเว็บเซิร์ฟเวอร์ สำหรับในระบบจะใช้ฐานข้อมูลของเอสคิวแอลสองพันเซิร์ฟเวอร์ (SQL2000Server) เป็นดาต้าเบสเซิร์ฟเวอร์

#### 4.2.4 รายละเอียดของระบบการทำงาน (System Specification)

เป็นการอธิบายรายละเอียดสเปกเครื่องคอมพิวเตอร์ของระบบที่ใช้ในการทำงาน

Web Server	
Hardware	- CPU Intel Pentium 4 2.4 GHz - Harddisk 30 GB - Ram 412 MB
Operating System	Windows XP Sp1
Web Server Program	IIS 4.1
Database Client Program	SQL 2000 Server
Database Server	
Hardware	- CPU Intel Pentium 4 2.4 GHz
Operating System	Windows XP Sp1
Database Management System	SQL Server Enterprise Manager

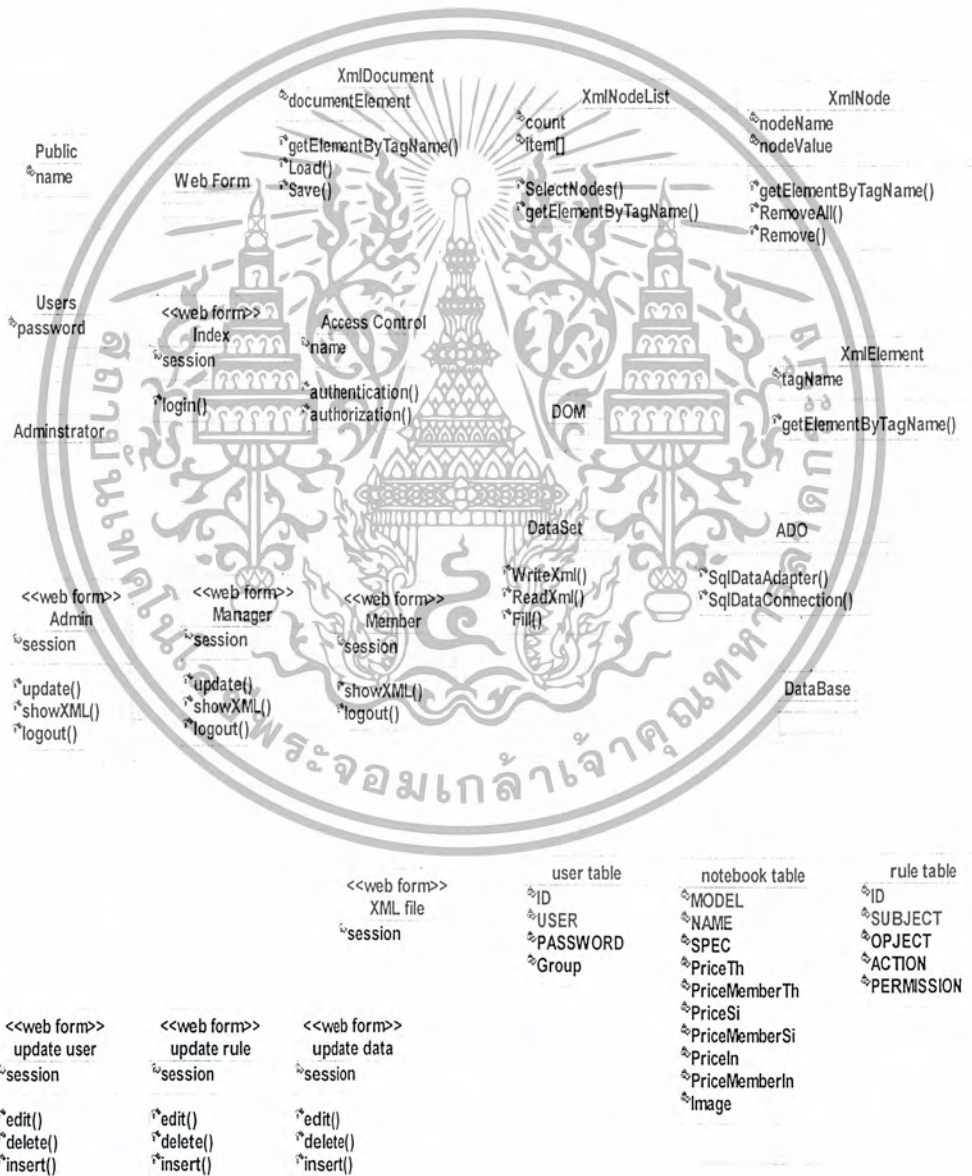
ตารางที่ 4-8 System Specification

#### 4.3 การออกแบบระดับกลไกการทำงาน

การออกแบบระบบในระดับกลไกการทำงาน เป็นการออกแบบที่จะนำผู้พัฒนาเข้าสู่การสร้าง (Implement) ระบบจริงได้อย่างมาก ซึ่งในการออกแบบขั้นตอนนี้ จะบอกวิธีการทำงานต่าง ๆ ของโปรแกรมในระบบ ส่วนต่างๆของโปรแกรมที่เกี่ยวข้อง โดยไดอะแกรม UML ที่ใช้ในขั้นตอนนี้ คือ คลาสไดอะแกรม (Class Diagram) ซึ่งแสดงถึงส่วนต่างๆของระบบ

### 4.3.1 คลาส ไดอะแกรม (Class Diagram)

ในการออกแบบระดับกลไกการทำงานใช้การออกแบบแบบ UML (Unified Modeling Language) ซึ่งเป็นโมเดลที่เป็นมาตรฐานอยู่ในขณะนี้ตัว UML นั้นเป็นกระบวนการความคิด (Methodology) และโมเดลที่สนับสนุนการวิเคราะห์และออกแบบระบบเชิงวัตถุที่สามารถเข้าใจได้ง่าย และสามารถประกอบในขั้นตอนการลงมือเขียนโปรแกรมได้ง่าย ใน UML คลาสจะเขียนเป็นรูปสี่เหลี่ยม โดยมีชื่อของคลาสดูด้านใน โดยอาจจะแบ่งเป็น 3 ส่วนโดยส่วนบนสุดเป็นชื่อคลาส ส่วนตรงกลางใช้แสดงรายละเอียดของแอททริบิวต์ ซึ่งเป็นการบอกลักษณะของคลาสนั้น ๆ และส่วนล่างใช้แสดงรายการของโอเปอเรชัน (Operation) ซึ่งเป็นการบอกความสามารถของคลาสนั้น



รูปที่ 4-16 Class Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

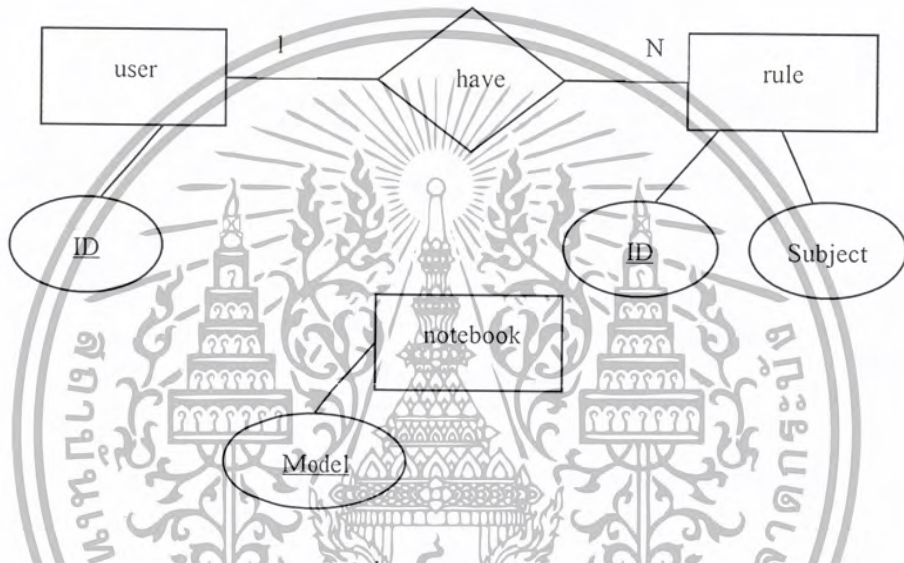
Class	การทำงาน
Public	แทนบุคคลทั่วไปที่เข้ามาชมเว็บไซต์
Users	แทนสมาชิกของระบบ
Administrator	แทนผู้ดูแลระบบ
Web Form	ทำหน้าที่สร้างเว็บฟอร์มขึ้นมาแสดง
Index	เป็นคลาสสร้างหน้าแรกของเว็บไซต์
Admin	เป็นคลาสสร้างหน้าของเว็บไซต์หลังจากผู้ดูแลระบบ login เข้าไป
Manager	เป็นคลาสสร้างหน้าของเว็บไซต์หลังจากผู้จัดการ login เข้าไป
Member	เป็นคลาสสร้างหน้าของเว็บไซต์หลังจากสมาชิก login เข้าไป
XML file	เป็นคลาสสร้างหน้าของเว็บไซต์ของเอกสารเอกซเอมแอล
Update user	เป็นคลาสสร้างหน้าของเว็บไซต์ที่ต้องการจัดการข้อมูลของผู้ใช้
Update rule	เป็นคลาสสร้างหน้าของเว็บไซต์ที่ต้องการจัดการการเข้าถึงของผู้ใช้
Update data	เป็นคลาสสร้างหน้าของเว็บไซต์ที่ต้องการจัดการฐานข้อมูล
Access Control	เป็นคลาสที่ทำหน้าที่ตรวจสอบบุคคลและสิทธิในการเข้าถึง
ADO	เป็นคลาสที่ติดต่อกับฐานข้อมูล
DataSet	เป็นคลาสภายในของ ADO ทำหน้าที่เก็บข้อมูล
DataBase	คลาสของฐานข้อมูล
user table	ตารางข้อมูลของผู้ใช้
notebook table	ตารางข้อมูลของสินค้า
rule table	ตารางข้อมูลของกฎการเข้าถึง
DOM	เป็นคลาสทำหน้าที่จัดการเอกสารเอกซเอมแอล
XmlElement	เป็นคลาสภายในของ DOM ทำหน้าที่จัดการเกี่ยวกับ Element
XmlNode	เป็นคลาสภายในของ DOM ทำหน้าที่จัดการเกี่ยวกับ Node
XmlNodeList	เป็นคลาสภายในของ DOM ทำหน้าที่จัดการเกี่ยวกับ Node List
XmlDocument	เป็นคลาสภายในของ DOM ทำหน้าที่จัดการเกี่ยวกับ Document

#### ตารางที่ 4-9 แสดงการทำงานของคลาสไออะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ดาต้าเบส (Database)

ในส่วนของการออกแบบฐานข้อมูล ซึ่งเป็นส่วนที่ใช้ในการเก็บข้อมูลต่าง ๆ ที่เกิดขึ้นในระบบ ต้องมีการออกแบบให้มีความสะดวกในการเรียกใช้ และเกิดความซ้ำซ้อนของข้อมูลน้อยที่สุด สำหรับส่วนการออกแบบฐานข้อมูลของระบบนี้ ได้ใช้ ER Diagram ซึ่งเป็นไดอะแกรม ที่ใช้ในการออกแบบฐานข้อมูลแบบรีชันแนล และทำการแมป (Map) ออกเป็นตารางของฐานข้อมูล โดยทำ Normalization เพื่อให้เกิดความซ้ำซ้อนของข้อมูลน้อยที่สุด



รูปที่ 4-17 ER Diagram ของระบบ

1.user

ID	USER	PASSWORD	GROUP
----	------	----------	-------

2.rule

ID	SUBJECT	OBJECT	ACTION	PERMISSION
----	---------	--------	--------	------------

3.notebook

Model	Name	Spec	PriceTh	PriceMemberTh
PriceSi	PriceMemberSi	PriceIn	PriceMemberIn	Image

รูปที่ 4-18 Database Table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 การออกแบบส่วนติดต่อผู้ใช้ (User Interface Design)

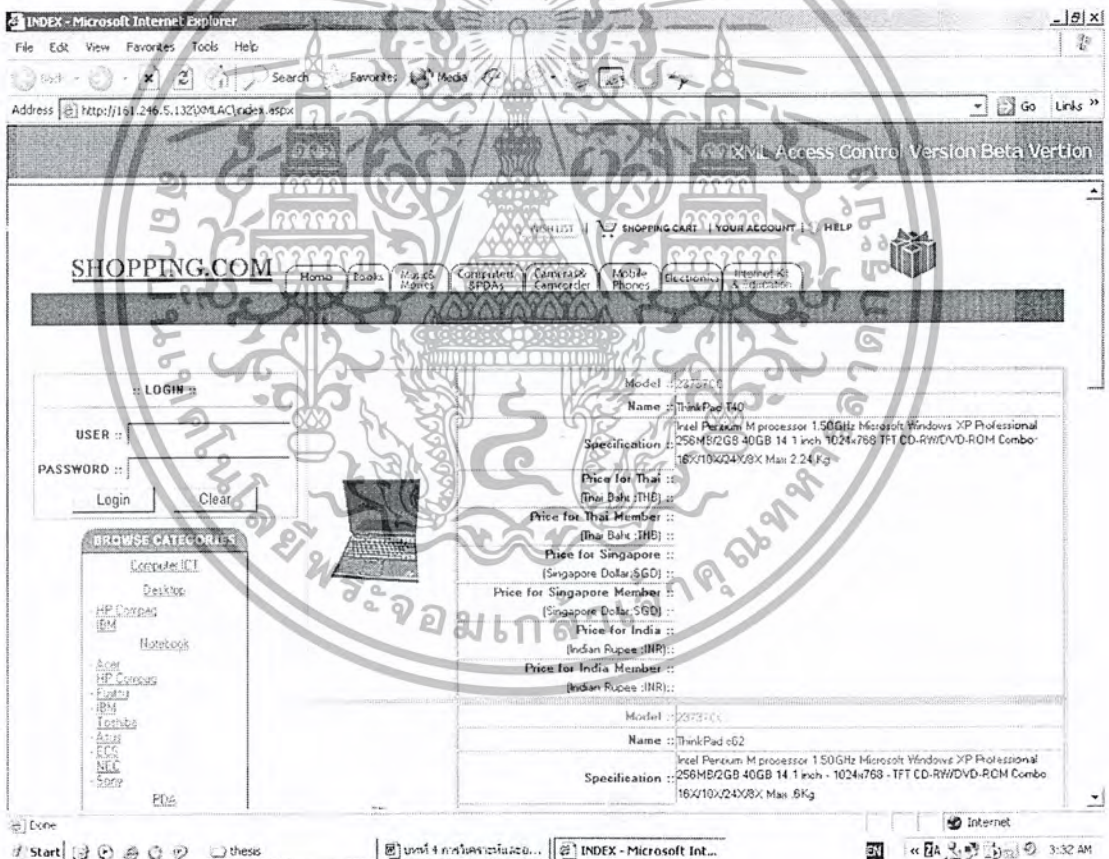
เป็นการออกแบบส่วนของระบบที่ทำการติดต่อกับผู้ใช้ ซึ่งก็คือ หน้าจอเว็บไซต์ ซึ่งการออกแบบส่วนติดต่อกับผู้ใช้นั้นมีความสำคัญมาก นอกจากจะออกแบบให้มีความน่าสนใจแล้ว ยังต้องออกแบบให้มีความสะดวกกับผู้ใช้ (User Friendly) มากที่สุดอีกด้วย

สำหรับการออกแบบส่วนติดต่อกับผู้ใช้ จะใช้การออกแบบส่วนที่ติดต่อกับแอกเตอร์ (Actors) ต่างๆ ที่มีในระบบ ซึ่งประกอบด้วย

##### 4.4.1 บุคคลทั่วไป (Public)

บุคคลทั่วไปในที่นี้หมายถึงผู้ใช้งานทั่วไป ที่เข้ามาเยี่ยมชมเว็บไซต์ และไม่ได้เป็นสมาชิกของระบบ บุคคลที่ไม่ได้เป็นสมาชิก จะถูกจำกัดสิทธิในการใช้เว็บไซต์โดยสามารถเข้าชมได้เพียงหน้าแรกของเว็บไซต์ (Index) เท่านั้น และไม่สามารถกระทำกิจกรรมอื่นๆ ได้

##### 4.4.1.1 หน้าแรกของเว็บไซต์ (Index)



รูปที่ 4-19 หน้าแรกของเว็บไซต์

เป็นหน้าแรกเมื่อผู้ใช้เปิด URL : <http://161.246.4.132/XMLAC/index.aspx> เพื่อเข้าสู่ระบบ จะแบ่งเป็น 2 ส่วนด้วยกัน ส่วนทางซ้ายมือเป็นส่วนของ Login ส่วนทางขวามือจะเป็นส่วนแสดงผล ในการเข้าหน้าแรก ไม่จำเป็นต้องใส่ User และ Password, ข้อมูลก็ยังคงแสดงออกมา เนื่องจากมีการพิสูจน์ตัวตนแบบ

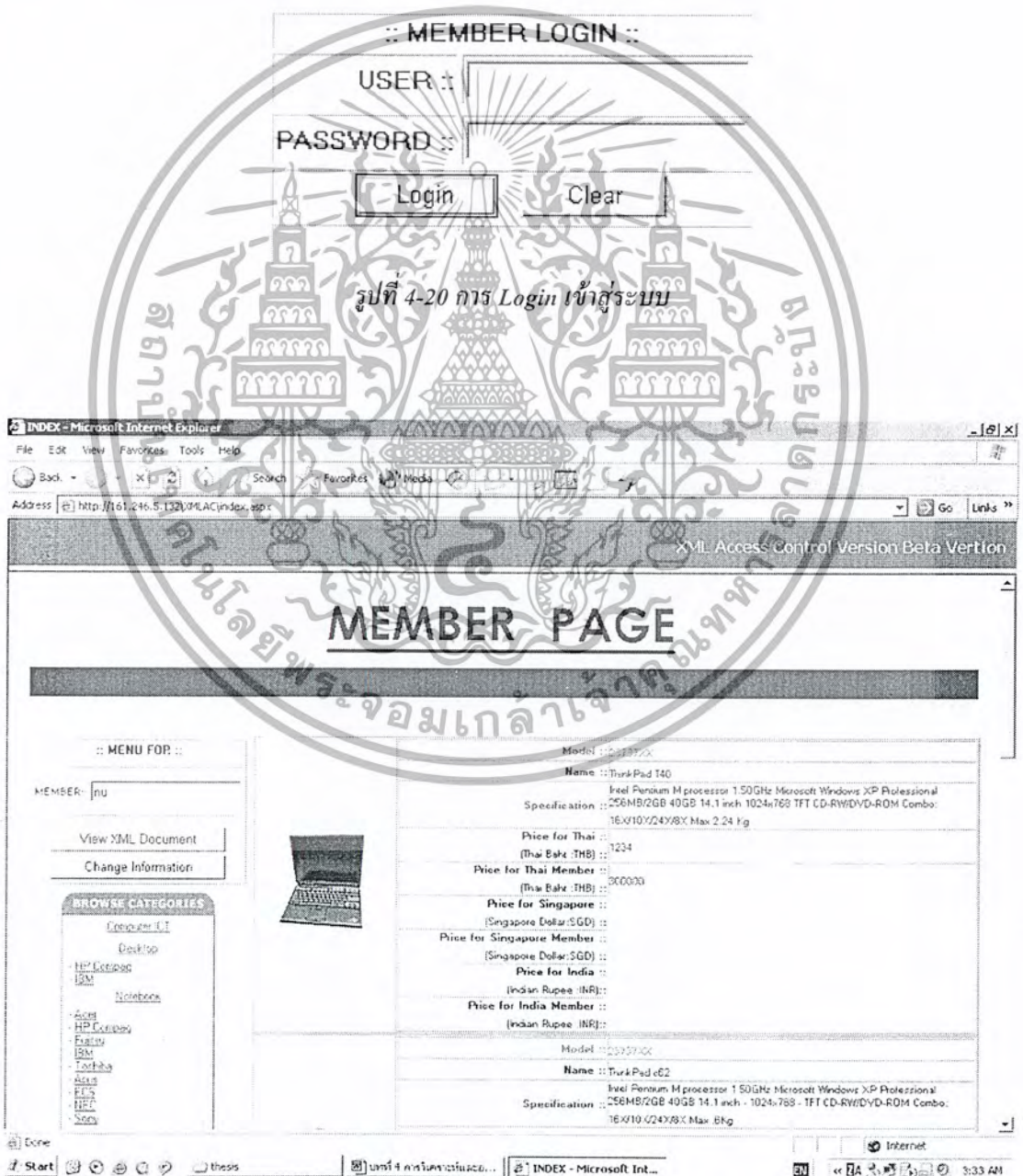
อัตโนมัติ โดยจะถือว่าบุคคลที่เข้ามาเยี่ยมชมเป็นบุคคลทั่วไปก็จะแสดงข้อมูลของบุคคลทั่วไปออกมา ด้านการคำนวณราคาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.2 สมาชิก (Member)

สมาชิกของเว็บไซต์เป็นบุคคลทั่วไปที่ได้มีการลงทะเบียนเป็นสมาชิกของทางเว็บไซต์แล้ว โดยการลงทะเบียนเป็นสมาชิกในที่นี้จะกำหนดให้ผูู้ดูแลระบบเป็นคนกำหนด โดยสมาชิกจะแบ่งออกเป็นสองแบบคือ บุคคลหรือกลุ่ม(Group)

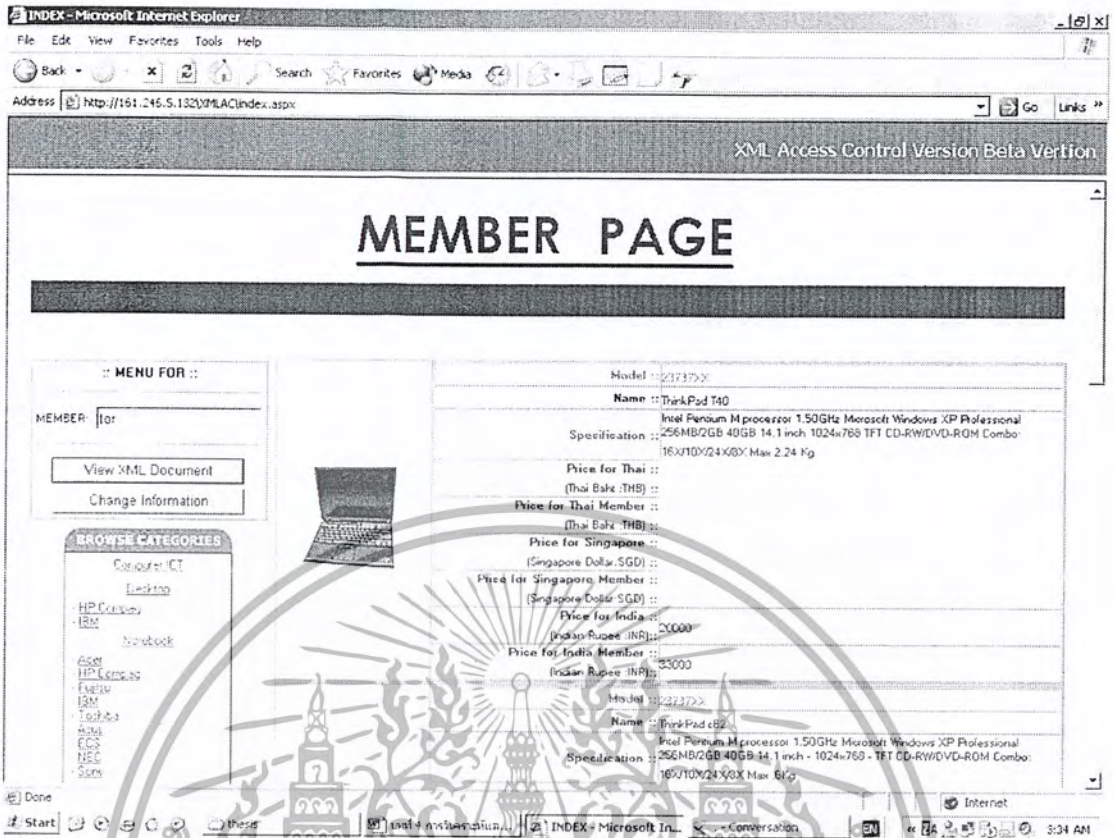
##### 4.4.2.1 การล็อกอิน (Login)

เป็นการเข้าสู่ระบบของสมาชิกโดยการใส่ Username และ Password ในหน้าแรกของเว็บไซต์ สมาชิกต้องทำการ Login เข้าสู่ระบบก่อน จึงจะสามารถใช้บริการในสถานะของสมาชิกได้ การ Login จะอยู่ทางซ้ายมือของหน้าแรก



รูปที่ 4-21 แสดงหน้าแรกของสมาชิกคนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

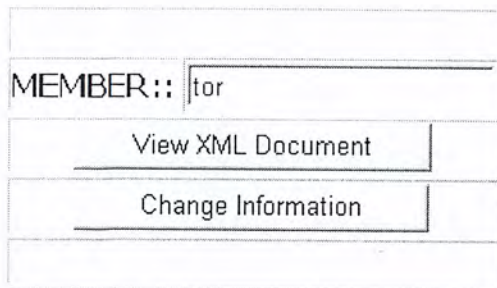


รูปที่ 4-22 แสดงหน้าแรกของสมาชิกคนที่ 2

จากรูปที่ 4-21 และ 4-22 แสดงให้เห็นถึงสมาชิกที่ต่างกัน ก็จะเห็นข้อมูลที่ต่างกัน ซึ่งจะเป็นไปตามกฎการเข้าถึง (Access Rule) ที่กำหนดไว้

4.4.2.2 การเปลี่ยนแปลงข้อมูลส่วนบุคคลของสมาชิก (Change Personal Information)

สมาชิกสามารถเปลี่ยนข้อมูลตัวต่างๆ ที่ระบุไว้ในการลงทะเบียนได้ ซึ่งประกอบด้วย ชื่อ และรหัสผ่าน และยังสามารรถ ดูตัวเอกสาร XML ได้



รูปที่ 4-23 แสดงส่วนการเปลี่ยนข้อมูลและแสดงเอกสาร XML

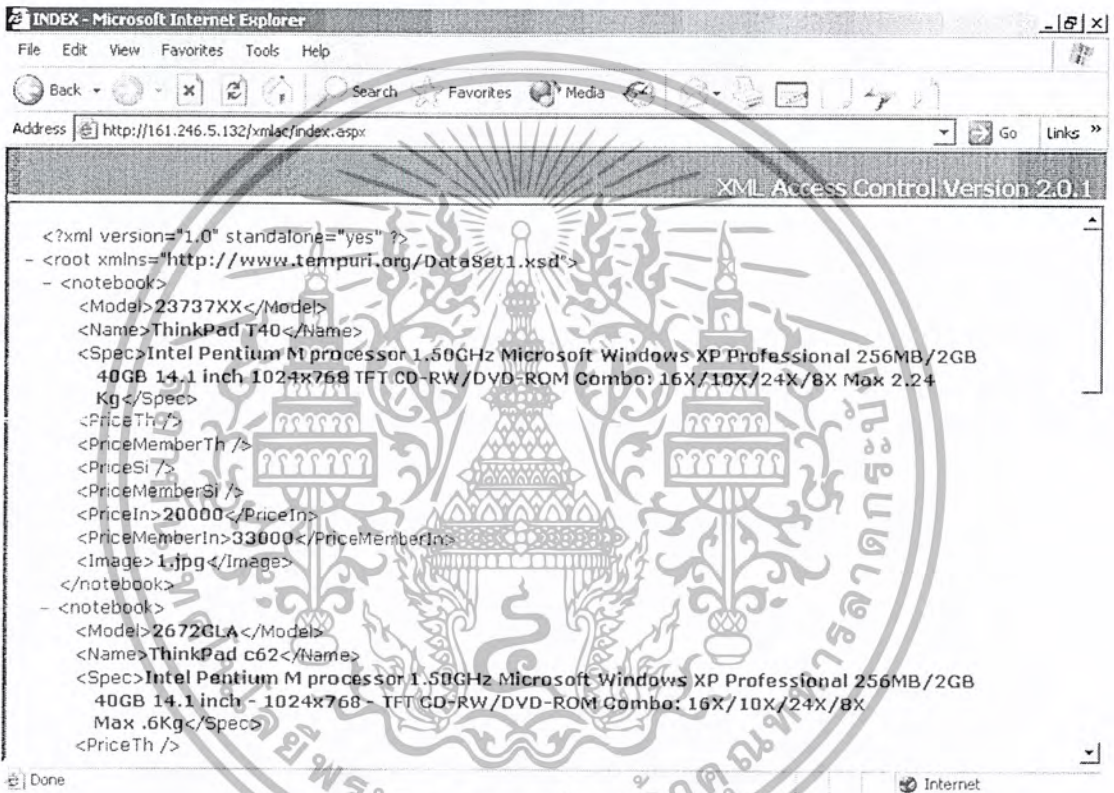
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Edit USER PASSWORD**

Edit tor tor

Edit	USER	PASSWORD
Update Cancel	tor	tor

รูปที่ 4-24 แสดงการเปลี่ยนแปลงข้อมูลส่วนตัว



รูปที่ 4-25 แสดงหน้าเว็บเพจเอกสาร XML ของสมาชิก

#### 4.4.3 ผู้จัดการ (Manager)

เป็นสมาชิกแบบหนึ่งที่มีสิทธิพิเศษนอกเหนือจากการดูข้อมูลยังสามารถแก้ไขข้อมูลโดยที่ผู้จัดการแต่ละคน สามารถแก้ไขข้อมูลได้แตกต่างกัน ตามที่กฎการเข้าถึงกำหนดเอาไว้ โดยการที่จะเข้าถึงหน้าเว็บเพจผู้จัดการ ต้องมีการพิสูจน์ตัวตนก่อน โดยการ Login เช่นเดียวกับสมาชิก ดังเช่นข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-26 แสดงหน้าเว็บผู้จัดการ

รูปที่ 4-27 แสดง Drop Down Menu สำหรับแก้ไขข้อมูล

#### 4.4.4 ผู้ดูแลระบบ (Administrator)

ผู้ดูแลระบบมีหน้าที่ดูแลระบบให้ดำเนินไปได้อย่างราบรื่น โดยมีส่วนที่ต้องรับผิดชอบ 2 ส่วน ได้แก่ จัดการเกี่ยวกับ User , ข้อมูล โดย User จะเกี่ยวกับการจัดการ เพิ่ม ลบ User โดยกำหนดและแก้ไข Password ให้กับ User ในส่วนของข้อมูล มีหน้าที่ เพิ่ม ลบ และแก้ไขข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDEX - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://161.246.5.132/XMLAC/index.aspx

XML Access Control Version Beta Verion

## ADMINISTRATOR PAGE

Edit with XPATH >> XPATH = [Model] OLD VALUE = NEW VALUE = Update Value

:: MENU FOR ::  
 ADMIN [ching]  
 View XML Document  
 Edit User  
 Edit Rule  
 Edit Data

Model	Name	Spec	PriceTh	PriceMemberTh	PriceSi	PriceMemberSi	PriceIn	PriceMemberIn	Image
23737CC	ThinkPad T40	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch 1024x768 TFT CD- RW/DVD-ROM Combo 15V/10K/24K/24K Max. 2.24 kg	123	12345	10000	3500	20000	33000	1.jpg
2672GLA	ThinkPad e62	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch 1024x768 TFT CD- RW/DVD-ROM Combo 15V/10K/24K/24K Max. 2.24 kg	91111	30000	1	3500	2000	30000	2.jpg
2672PBA	ThinkPad e65	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch 1024x768 TFT CD- RW/DVD-ROM Combo 15V/10K/24K/24K Max. 2.24 kg with Bluetooth	30000	30000	10000	51321	20000	32123	3.jpg
2682K4A	ThinkPad X40	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch 1024x768 TFT CD- RW/DVD-ROM Combo 15V/10K/24K/24K Max. 2.24 kg	45000	321546	65402	32132	654132	54654	4.jpg

Done Start

รูปที่ 4-28 แสดงหน้าเว็บของ Administrator

Administrator: [ching]

View XML Document

Edit User

Edit Rule

Edit Data

รูปที่ 4-29 เมนูสำหรับ Admin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDEX - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address: http://161.246.5.132/XMLAC/index.aspx

XML Access Control Version Beta Ver1.0n

```
<?xml version="1.0" standalone="yes" ?>
<root xmlns="http://www.tempuri.org/DataSet1.xsd">
  <notebook>
    <Model>23737CC</Model>
    <Name>ThinkPad T40</Name>
    <Spec>Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch 1024x768 TFT
    CD-RW/DVD-ROM Combo: 16X/10X/24X/8X Max 2.24 Kg</Spec>
    <PriceTh>123</PriceTh>
    <PriceMemberTh>12345</PriceMemberTh>
    <PriceSi>10000</PriceSi>
    <PriceMemberSi>3500</PriceMemberSi>
    <PriceIn>20000</PriceIn>
    <PriceMemberIn>33000</PriceMemberIn>
    <Image>1.jpg</Image>
  </notebook>
  <notebook>
    <Model>2672GLA</Model>
    <Name>ThinkPad c62</Name>
    <Spec>Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch - 1024x768 -
    TFT CD-RW/DVD-ROM Combo: 16X/10X/24X/8X Max .6Kg</Spec>
    <PriceTh>91111</PriceTh>
    <PriceMemberTh>30000</PriceMemberTh>
    <PriceSi>1</PriceSi>
    <PriceMemberSi>3500</PriceMemberSi>
    <PriceIn>2000</PriceIn>
    <PriceMemberIn>30000</PriceMemberIn>
    <Image>2.jpg</Image>
  </notebook>
  <notebook>
    <Model>2672PBA</Model>
    <Name>ThinkPad m55</Name>
    <Spec>Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch - 1024x768 -
```

Done Start

Internet 3:36 AM

รูปที่ 4-30 แสดงหน้าเว็บ Admin ดูเอกสาร XML

Edit	ID	USER	PASSWORD	GROUP	Delete Item
<a href="#">Insert New USER</a>	0				<a href="#">Delete</a>
<a href="#">Edit</a>	0	public	public	public	<a href="#">Delete</a>
<a href="#">Edit</a>	1	ching	ching	admin	<a href="#">Delete</a>
<a href="#">Edit</a>	2	ed	ed	member	<a href="#">Delete</a>
<a href="#">Edit</a>	3	tor	tor	member	<a href="#">Delete</a>
<a href="#">Edit</a>	4	nu	nu	member	<a href="#">Delete</a>
<a href="#">Edit</a>	5	min	min	manager	<a href="#">Delete</a>
<a href="#">Edit</a>	6	wan	wan	manager	<a href="#">Delete</a>
<a href="#">Edit</a>	7	am	am	manager	<a href="#">Delete</a>

รูปที่ 4-31 แสดง Edit USER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Edit	ID	Subject	Object	Action	Permission	Delete Item
<a href="#">Insert New RULE</a>	0					<a href="#">Delete</a>
<a href="#">Edit</a>	0	public	/mstns:root/mstns:notebook/mstns:PriceTh	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	1	public	/mstns:root/mstns:notebook/mstns:PriceSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	2	public	/mstns:root/mstns:notebook/mstns:PriceIn	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	3	public	/mstns:root/mstns:notebook/mstns:PriceMemberTh	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	4	public	/mstns:root/mstns:notebook/mstns:PriceMemberSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	5	public	/mstns:root/mstns:notebook/mstns:PriceMemberIn	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	6	min	/mstns:root/mstns:notebook/mstns:PriceSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	7	min	/mstns:root/mstns:notebook/mstns:PriceIn	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	8	min	/mstns:root/mstns:notebook/mstns:PriceMemberSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	9	min	/mstns:root/mstns:notebook/mstns:PriceMemberIn	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	10	wan	/mstns:root/mstns:notebook/mstns:PriceTh	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	11	wan	/mstns:root/mstns:notebook/mstns:PriceSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	12	wan	/mstns:root/mstns:notebook/mstns:PriceMemberTh	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	13	wan	/mstns:root/mstns:notebook/mstns:PriceMemberSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	14	am	/mstns:root/mstns:notebook/mstns:PriceTh	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	15	am	/mstns:root/mstns:notebook/mstns:PriceIn	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	16	am	/mstns:root/mstns:notebook/mstns:PriceMemberTh	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	17	am	/mstns:root/mstns:notebook/mstns:PriceMemberIn	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	18	nu	/mstns:root/mstns:notebook/mstns:PriceSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	19	nu	/mstns:root/mstns:notebook/mstns:PriceIn	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	20	nu	/mstns:root/mstns:notebook/mstns:PriceMemberSi	read	deny	<a href="#">Delete</a>
<a href="#">Edit</a>	21	nu	/mstns:root/mstns:notebook/mstns:PriceMemberIn	read	deny	<a href="#">Delete</a>

รูปที่ 4-32 แสดง Edit RULE

Edit	Model	Name	Spec	PriceTh	PriceMemberTh	PriceSI	PriceMemberSI	PriceIn	PriceMemberIn	Delete Item
<a href="#">Insert New RULE</a>										<a href="#">Delete</a>
<a href="#">Edit</a>	23737CC	ThinkPad T40	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch-1024x768 TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max 2.24 Kg	333	12345	10000	3000	30000	33000	<a href="#">Delete</a>
<a href="#">Edit</a>	2672GLA	ThinkPad e62	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch-1024x768 TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max 6Kg	91111	30000		3000	2000	30000	<a href="#">Delete</a>
<a href="#">Edit</a>	2672PEA	ThinkPad m55	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch-1024x768 TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max 6Kg with Bluetooth	80000	30000	10000	51321	20000	32123	<a href="#">Delete</a>
<a href="#">Edit</a>	2682K4A	ThinkPad X40	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch-1024x768 TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max 6Kg	45000	321546	65402	32132	654132	54654	<a href="#">Delete</a>
<a href="#">Edit</a>	2723BAA	ThinkPad x40	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch-1024x768 TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max 6Kg with Bluetooth	92000	465464	231321	6546	321321	65464	<a href="#">Delete</a>

รูปที่ 4-33 แสดง Edit DATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

## การทดลองและผลการทดลอง

### 5.1 จุดประสงค์การทดลอง

- 1) เพื่อทดสอบโปรแกรมที่สร้างขึ้นตามแนวคิดการเพิ่มความปลอดภัย ในการเข้าถึงเอกสารเอก-  
ชอมแอล ทดสอบการทำงานว่าถูกต้องหรือไม่และเหมาะสมกับการพัฒนาเป็นระบบที่นำไปใช้จริง  
หรือไม่
- 2) เพื่อเป็นแนวทางในการอ้างอิงในการสรุปและวิเคราะห์โดยใช้เหตุผลและผลการทดลองเป็น  
พื้นฐาน

### 5.2 การเตรียมอุปกรณ์และสถานะการทำงานเพื่อทดสอบโปรแกรม

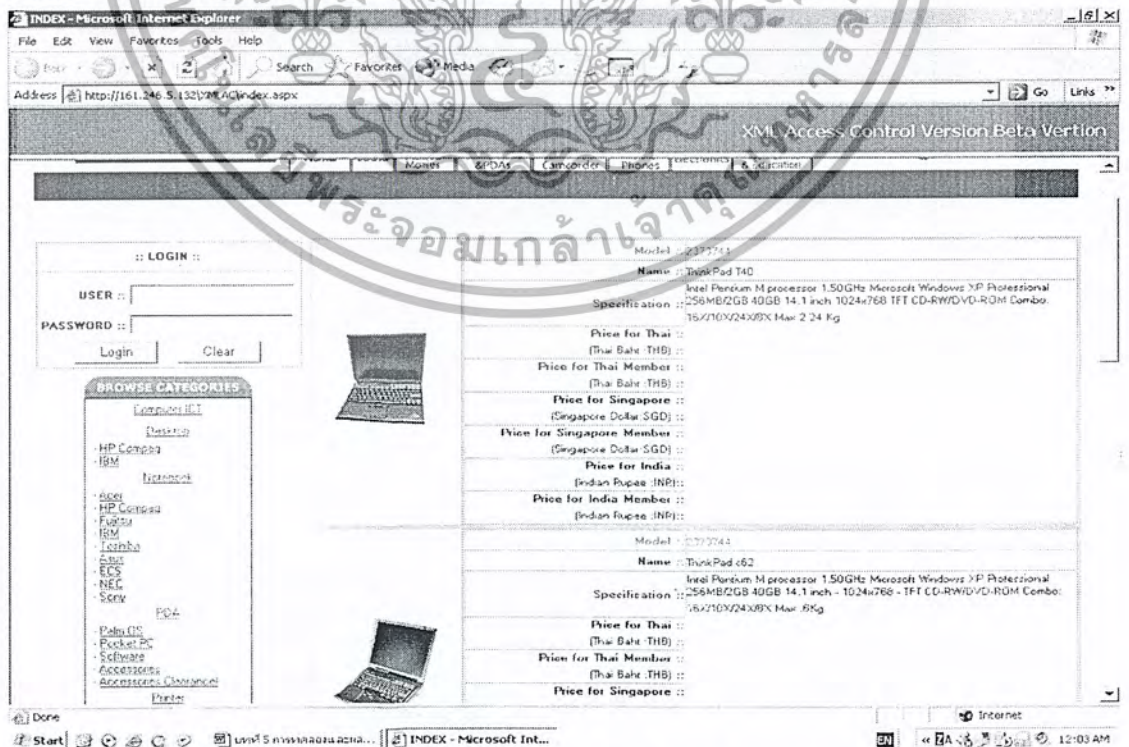
เตรียมอุปกรณ์ที่จะใช้รันตัวระบบ ประกอบด้วย

เครื่องเว็บเซิร์ฟเวอร์	ทำหน้าที่ลงโปรแกรมที่สร้างขึ้นมา
เครื่องคาล์เบสเซอร์ฟเวอร์	ทำหน้าที่เก็บคาล์เบส
เครื่องไคลเอ็นต์	ทำหน้าที่ติดต่อเข้าไปใช้งาน มีได้ตั้งแต่ 2 เครื่องขึ้นไป

ตารางที่ 5-1 แสดงอุปกรณ์การทดลอง

### 5.3 ทำการทดลองโปรแกรม

#### 5.3.1 ทดลองรันเรียกเว็บเพจ



รูปที่ 5-1 ทดสอบเรียกเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.2 ทดลอง login เข้าสู่ระบบโดยเป็นแบบสมาชิก

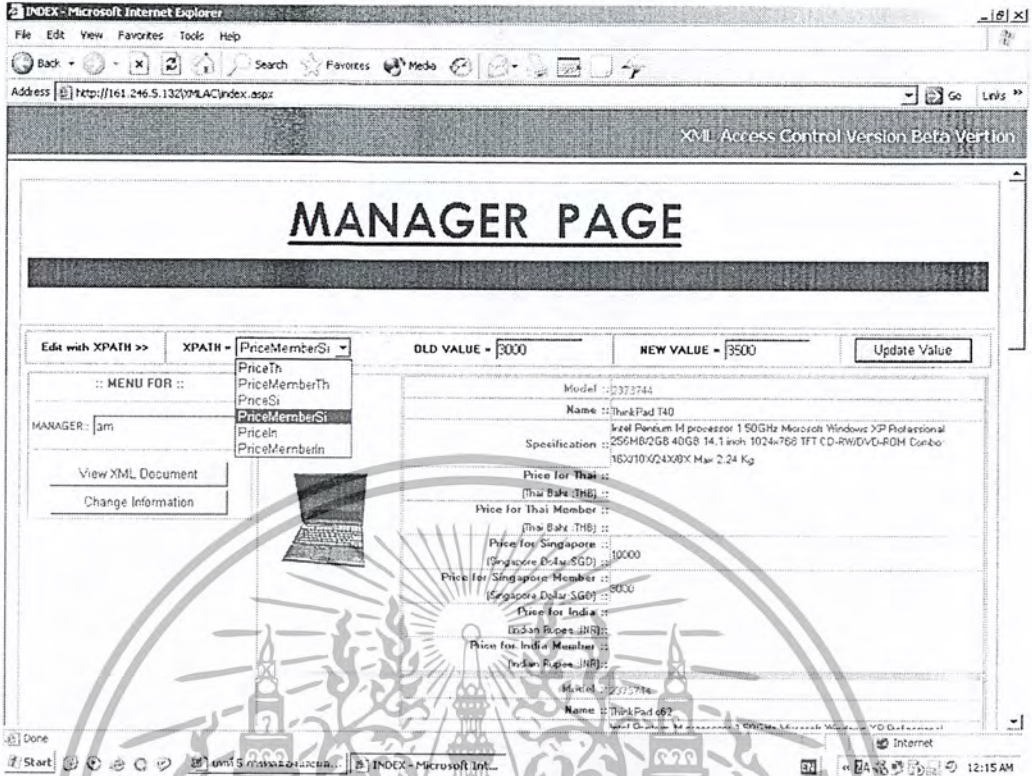
รูปที่ 5-2 ทดลอง login เข้าสู่ระบบโดยเป็นแบบสมาชิก

### 5.3.3 ทดลอง login เข้าสู่ระบบโดยเป็นแบบผู้จัดการ

รูปที่ 5-3 ทดลอง login เข้าสู่ระบบโดยเป็นแบบผู้จัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.4 ทดลองผู้จัดการเปลี่ยนค่า



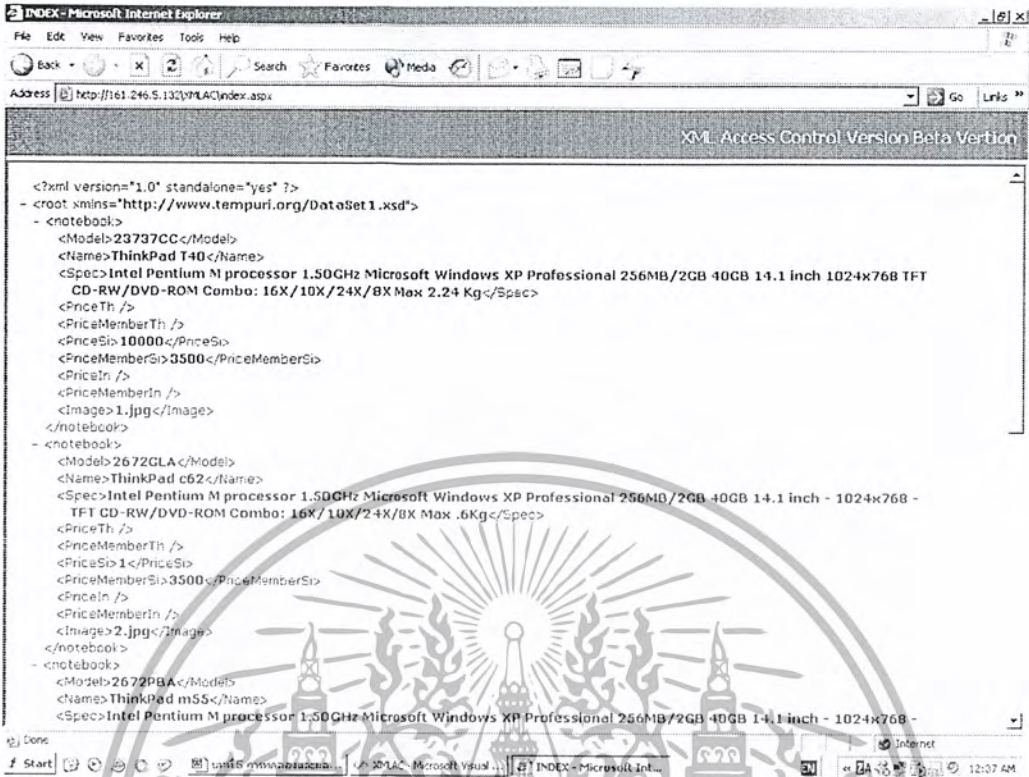
รูปที่ 5-4 ทดลองผู้จัดการเปลี่ยนค่า



รูปที่ 5-5 ผลการทดลองผู้จัดการเปลี่ยนค่า

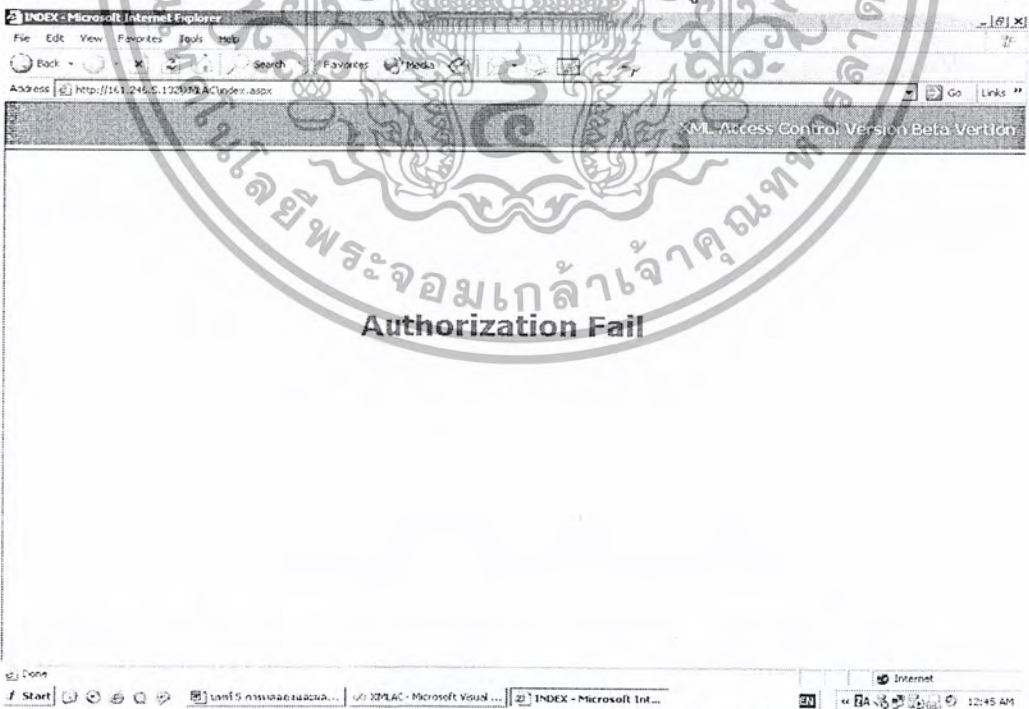
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.5 ทดลองเข้าดูเอกสารของผู้จัดการ



รูปที่ 5-6 ผลการทดลองเข้าดูเอกสาร

### 5.3.6 ทดลองเปลี่ยนค่าในส่วนที่ไม่สามารถแก้ไขได้ในส่วนของผู้จัดการ



รูปที่ 5-7 ผลการทดลองเปลี่ยนค่าในส่วนที่ไม่สามารถแก้ไขได้ในส่วนของผู้จัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.7 ทดลองผู้ดูแลระบบ login

XML Access Control Version Beta Ver1.0

## ADMINISTRATOR PAGE

Edit with XPATH >> XPATH = Model OLD VALUE = NEW VALUE = Update Value

ADMIN: ching

View XML Document

Model	Name	Spec	PriceTh	PriceMemberTh	PriceSi	PriceMemberSi	PriceIn	PriceMemberIn	Image
23737CC	ThinkPad T40	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch 1024x768 TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max 2.24 Kg	123	12345	10000	3500	20000	30000	1.jpg
2672GLA	ThinkPad c62	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch - 1024x768 - TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max .6Kg	91111	30000	1	3500	2000	30000	2.jpg
2672PBA	ThinkPad m55	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch - 1024x768 - TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max .6kg with Bluetooth	80000	30000	10000	51321	20000	32123	3.jpg
2682K4A	ThinkPad T60	Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch - 1024x768 - TFT CD-RW/DVD-ROM Combo 16X/10X/24X/8X Max .6kg with Bluetooth	45000	32154	65000	32132	65432	54654	4.jpg

รูปที่ 5-8 ทดลอง login เข้าสู่ระบบโดยเป็นแบบผู้ดูแลระบบ

### 5.3.8 ทดลองเข้าดูเอกสารของผู้ดูแลระบบ

XML Access Control Version Beta Ver1.0

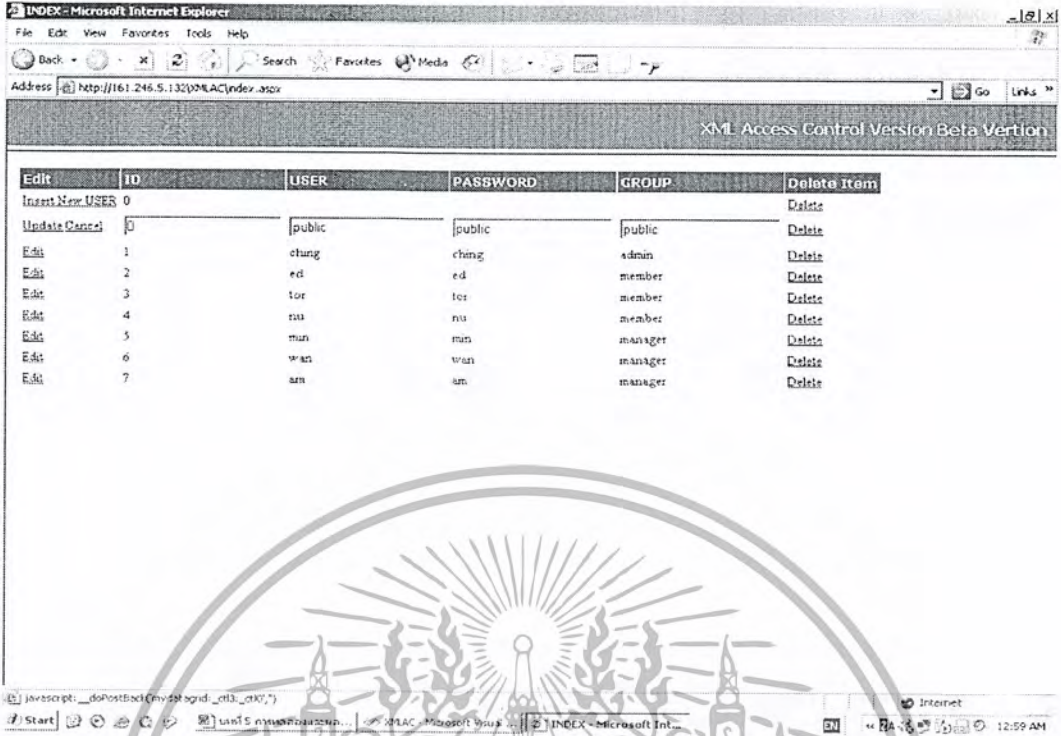
```

<?xml version="1.0" standalone="yes" ?>
<root xmlns="http://www.temple.org/DataSet1.xsd">
  <notebook>
    <Model>23737CC</Model>
    <Name>ThinkPad T40</Name>
    <Spec>Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch 1024x768 TFT CD-RW/DVD-ROM Combo: 16X/10X/24X/8X Max 2.24 Kg</Spec>
    <PriceTh>123</PriceTh>
    <PriceMemberTh>12345</PriceMemberTh>
    <PriceSi>10000</PriceSi>
    <PriceMemberSi>3500</PriceMemberSi>
    <PriceIn>20000</PriceIn>
    <PriceMemberIn>30000</PriceMemberIn>
    <Image>1.jpg</Image>
  </notebook>
  <notebook>
    <Model>2672GLA</Model>
    <Name>ThinkPad c62</Name>
    <Spec>Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch - 1024x768 - TFT CD-RW/DVD-ROM Combo: 16X/10X/24X/8X Max .6Kg</Spec>
    <PriceTh>91111</PriceTh>
    <PriceMemberTh>30000</PriceMemberTh>
    <PriceSi>1</PriceSi>
    <PriceMemberSi>3500</PriceMemberSi>
    <PriceIn>2000</PriceIn>
    <PriceMemberIn>30000</PriceMemberIn>
    <Image>2.jpg</Image>
  </notebook>
  <notebook>
    <Model>2672PBA</Model>
    <Name>ThinkPad m55</Name>
    <Spec>Intel Pentium M processor 1.50GHz Microsoft Windows XP Professional 256MB/2GB 40GB 14.1 inch - 1024x768 -
  
```

รูปที่ 5-9 ผลการทดลองเข้าดูเอกสาร

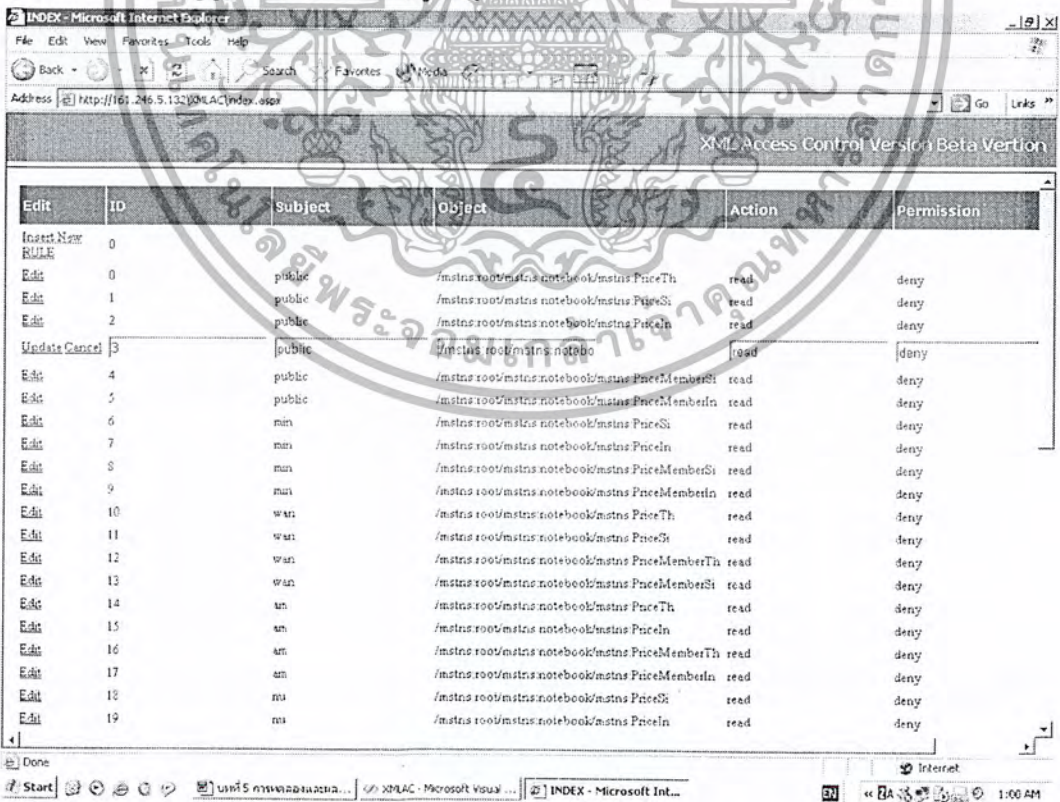
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.9 ลองผู้ดูแลระบบเปลี่ยนข้อมูลผู้ใช้



รูปที่ 5-10 ผลการทดลองผู้ดูแลระบบเปลี่ยนข้อมูลผู้ใช้

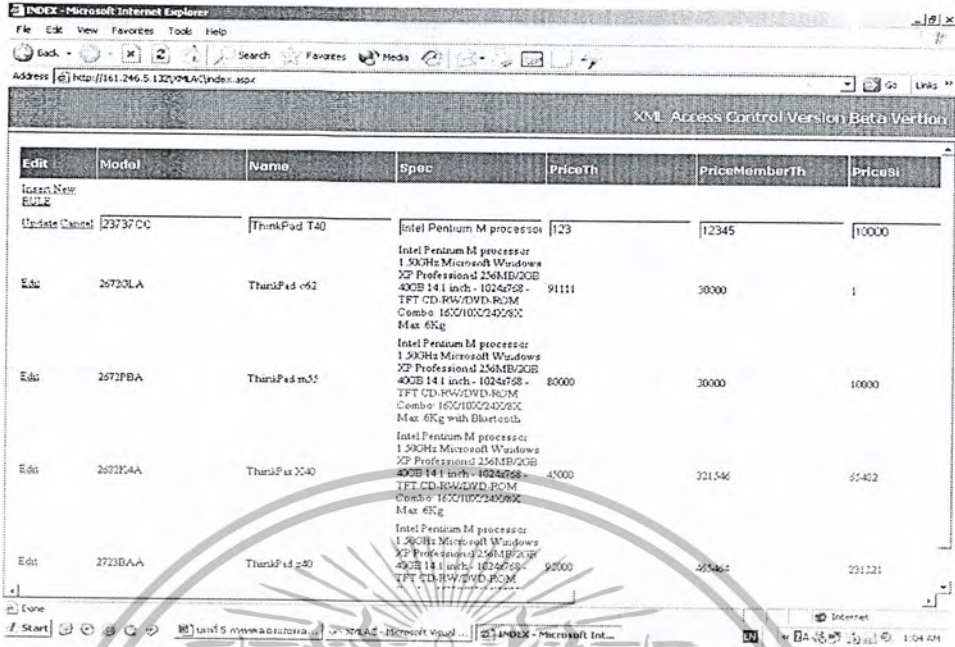
### 5.3.10 ทดลองผู้ดูแลระบบเปลี่ยนข้อมูลกฎการเข้าถึง



รูปที่ 5-11 ผลการทดลองผู้ดูแลระบบเปลี่ยนข้อมูลกฎการเข้าถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.11 ทดลองผู้ดูแลระบบแก้ไขข้อมูลสินค้า



รูปที่ 5-12 ผลการทดลองแก้ไขข้อมูลสินค้า

5.3 ผลการทดลอง

การทดลองที่		ผลการทดลอง	
		ทำงานถูกต้อง	ทำงานไม่ถูกต้อง
5.3.1	ทดลองรันเรียกเว็บเพจ	×	
5.3.2	ทดลอง login เข้าสู่ระบบโดยเป็นแบบสมาชิก	×	
5.3.3	ทดลอง login เข้าสู่ระบบโดยเป็นแบบผู้จัดการ	×	
5.3.4	ทดลอง login เข้าสู่ระบบโดยเป็นแบบผู้จัดการ	×	
5.3.5	ทดลองผู้จัดการเปลี่ยนค่า	×	
5.3.6	ทดลองเจ้าดูแลเอกสารของผู้จัดการ	×	
5.3.7	ทดลองผู้ดูแลระบบ login ทดลองเจ้าดูแลเอกสารของผู้ดูแลระบบ	×	
5.3.10	ทดลองผู้ดูแลระบบเปลี่ยนข้อมูลผู้ใช้	×	
5.3.11	ทดลองผู้ดูแลระบบเปลี่ยนข้อมูลลูกค้าเข้าถึง	×	
5.3.12	ทดลองผู้ดูแลระบบแก้ไขข้อมูลสินค้า	×	

ตารางที่ 5-2 ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

# บทวิจารณ์และสรุป

### 6.1 บทวิจารณ์และสรุป

โครงการนี้เป็นการศึกษา ออกแบบ และพัฒนาระบบการเข้าถึงเอกสารเอกซเอมแอล และนำระบบที่ออกแบบและพัฒนาไปประยุกต์ใช้กับโปรแกรมประยุกต์บนเว็บไซต์ (Web Application) ซึ่งแนวคิดหลักของการศึกษารายละเอียดเน้นไปที่เทคโนโลยีของเอกซเอมแอล (XML Technology) และเทคโนโลยีอื่น ๆ ที่เกี่ยวข้อง เช่น DOM ซึ่งเป็น API ที่ใช้ติดต่อกับเอกสารเอกซเอมแอล และนอกจากเทคโนโลยีที่ต้องศึกษาแล้ว ตัวโครงการจะเน้นถึงหลักการออกแบบกฎในการเข้าถึงตัวเอกสารเอกซเอมแอล

หลักจากผ่านในส่วนแรกในการศึกษาและออกแบบกฎ ในส่วนที่สองจะทำการนำระบบที่ออกแบบในการเข้าถึงเอกสารมาประยุกต์ใช้กับโปรแกรมประยุกต์บนเว็บไซต์ โดยระบบที่นำเสนอเป็นการจัดการของร้านค้าแห่งหนึ่งที่ต้องการแยกระดับของราคาของที่มีในร้านให้สมาชิกหรือผู้เข้าชมเห็นในระดับที่แตกต่างกัน โดยสามารถตั้งค่าระดับการเข้าถึงของแต่ละคนได้ ซึ่งจะก่อให้เกิดประโยชน์ต่อการรักษาความลับ หรือการนำเสนอข้อมูลที่ไม่จำเป็นของฝ่ายหนึ่งให้อีกฝ่ายหนึ่งดูหรือเข้าถึงได้

หลังจากการทดสอบตัวโปรแกรมประยุกต์หลังจากเสร็จสิ้น ปรากฏว่าสามารถทำงานได้ถูกต้องตามเป้าหมายที่วางไว้ คือ สามารถควบคุมการเข้าถึงเอกสารเอกซเอมแอลได้ สามารถตั้งค่าการเข้าถึงได้ แต่คนที่จะเป็นผู้ดูแลระบบจะต้องมีความสามารถและเข้าใจในเทคโนโลยีของเอกซเอมแอลอย่างดี และที่สำคัญคือเทคโนโลยีในเรื่องของเอกซเอมแอล (XPath) ซึ่งเป็นการกำหนดเส้นทางการเข้าถึงโหนด (Node) ในเอกสารเอกซเอมแอล เพราะหากมีการตั้งค่าผิดก็จะทำให้ระบบผิดพลาดที่ ซึ่งเป็นจุดที่ละเอียดอ่อนมาก

การป้องกันความปลอดภัยของข้อมูลนั้นมีมากมายหลายวิธีสำหรับโครงการนี้ก็เป็นส่วนหนึ่งของการเพิ่มความปลอดภัยในการใช้ข้อมูลเอกสารเอกซเอมแอลบนอินเทอร์เน็ต ในส่วนของการนำเสนอข้อมูลเท่านั้น ในความเป็นจริงในการนำไปใช้งานก็ต้องมีการเพิ่มความปลอดภัยในจุดอื่น ๆ อีกมากมาย เพื่อเพิ่มความปลอดภัยในจุดอื่นที่โครงการนี้ไม่ได้วิเคราะห์ เพื่อเพิ่มประสิทธิภาพในการทำงานด้านความปลอดภัยมากยิ่งขึ้น

### 6.2 ปัญหาและอุปสรรคที่พบ

- 1) ความไม่เป็นอิสระของระบบการควบคุมการเข้าถึงเอกสารเอกซเอมแอล เนื่องจากต้องพัฒนาควบคู่ไปกับการออกแบบโปรแกรมประยุกต์ที่จะนำไปใช้ร่วมกัน หากมีการแก้ไขข้อมูลบางส่วน ก็ต้องมีการแก้โปรแกรมในบางส่วนด้วยเช่นกัน
- 2) เนื่องจากเทคโนโลยีของเอกซเอมแอลเป็นเทคโนโลยีใหม่และมีเทคโนโลยีที่เกี่ยวข้องอีกมาก ทำให้เสียเวลานานการค้นคว้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 แนวทางการศึกษาต่อ

- 1) ควรแยกระบบการเข้าถึงเอกสารเอกซเอมแอลออกจากโปรแกรมประยุกต์เพื่อให้เกิดเป็นอิสระต่อกันโดยไม่ต้องค่อยแก้โปรแกรมเวลาเปลี่ยนเนื้อหาข้อเอกสารเอกซเอมแอล

- 2) เนื่องจากโครงการนี้สร้างบริการหลักสองอย่างคือ

- การอ่านข้อมูลจากเอกสารเอกซเอมแอล (Read)
- การแก้ไขข้อมูลจากเอกสารเอกซเอมแอล (Write)

เนื่องจากบริการที่กล่าวมาน่าจะเพิ่มบริการอีกสองอย่างคือ

- การเพิ่มแท็กข้อมูล (Append Tag)
- การลบแท็กข้อมูล (Delete Tag)

เพื่อเพิ่มประสิทธิภาพในการทำงานในอนาคต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

XML Document Object Model

ภาคผนวกนี้จะแสดงอินเตอร์เฟซทั้งหมดที่อยู่ในแกนระดับ 2 ของ DOM ซึ่งมีทั้งอินเตอร์เฟซมาตรฐาน (Fundamental) และ อินเตอร์เฟซส่วนขยาย (Extended) โดยรวมถึงพรีอปเพอร์ตีและเมธอดของมันทั้งหมด ตัวอย่าง ของวิธีการใช้อินเตอร์เฟซเหล่านี้จะมีให้เห็นในบทที่ 6

ข้อมูลเพิ่มเติมของอินเตอร์เฟซเหล่านี้สามารถดูได้ที่ <http://www.w3.org/TR/1999/CR-DOM-Level-2-19991210/core.html>

อินเตอร์เฟซมาตรฐาน

อินเตอร์เฟซมาตรฐานของ DOM เป็นอินเตอร์เฟซที่ในการใช้งาน DOM จะต้องมีไว้ให้แม้ว่ามันไม่ได้ถูกออกแบบให้ใช้งานกับเอกสาร XML เลยก็ตาม

อินเตอร์เฟซ DOMException

ออบเจกต์ที่กำลังใช้งานอินเตอร์เฟซ DOMException จะถูกส่งออกมาถ้ามีข้อผิดพลาดเกิดขึ้นใน DOM

พรีอปเพอร์ตี	คำอธิบาย
code	เลขจำนวนเต็มซึ่งจะแสดงถึงโค้ดข้อยกเว้น (Exception Code) ที่อินเตอร์เฟซ DOMException นั้นรายงานออกมา

พรีอปเพอร์ตี โค้ด จะให้ค่าดังต่อไปนี้

โค้ดข้อยกเว้น	ค่าจำนวนเต็ม	คำอธิบาย
INDEX_SIZE_ERR	1	อินเด็กซ์หรือขนาดเป็นค่าลบ หรือมากกว่าค่าที่อนุญาต
DOMSTRING_SIZE_ERR	2	ขอบเขตของข้อความที่ไม่ลงตัวใน DOMString
HIERARCHY_REQUEST_ERR	3	โหนดได้ถูกแทรกไว้ในที่ที่ไม่สัมพันธ์กัน
WRONG_DOCUMENT_ERR	4	โหนดถูกใช้ในเอกสารที่แตกต่างไปจากที่ได้สร้างมันขึ้นมาและไม่สนับสนุนการใช้งานเอกสารนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดข้อบกพร่อง	จำนวนข้อผิดพลาด	คำอธิบาย
INVALID_CHARACTER_ERR	5	อักขระได้ถูกส่งผ่านซึ่งไม่ถูกกฎ XML
NO_DATA_ALLOWED_ERR	6	ข้อมูลได้ถูกระบุในโหนดที่ไม่ได้สนับสนุนข้อมูลนั้น
NO_MODIFICATION_ALLOWED_ERR	7	พยายามที่จะแก้ไขออบเจกต์ที่ไม่อนุญาตให้แก้ไขได้
NOT_FOUND_ERR	8	พยายามที่จะอ้างถึงโหนดที่ไม่ได้มีอยู่
NOT_SUPPORTED_ERR	9	การใช้งานที่ไม่สนับสนุนชนิดของออบเจกต์ที่มีการร้องขอ
INUSE_ATTRIBUTE_ERR	10	พยายามที่จะทำการเพิ่มออบเจกต์ที่ซ้ำกัน
INVALID_STATE_ERR	11	พยายามที่จะใช้ออบเจกต์ที่ไม่ใช้ออบเจกต์หรือไม่มีอยู่ หรือใช้ไม่ได้
SYNTAX_ERR	12	สตริงที่ผิดพลาดถูกส่งผ่าน
INVALID_MODIFICATION_ERR	13	พยายามที่จะทำการแก้ไขชนิดของออบเจกต์ที่สำคัญ
NAMESPACE_ERR	14	พยายามที่จะสร้างหรือเปลี่ยนแปลงออบเจกต์ในวิธีการที่เข้ากันไม่ได้กับเนมสเปซ
INVALID_ACCESS_ERR	15	พารามิเตอร์ถูกส่งผ่านหรือปฏิบัติการที่พยายามนำโดยที่ไม่สนับสนุนโดยออบเจกต์ที่สำคัญ

### อินเตอร์เฟซ Node

อินเตอร์เฟซ โหนด เป็นอินเตอร์เฟซพื้นฐานซึ่งออบเจกต์ DOM ส่วนใหญ่ถูกสร้างขึ้นมา และประกอบไปด้วยเมธอดและแอททริบิวต์ซึ่งสามารถนำไปใช้ได้กับชนิดของโหนดทั้งหมด อินเตอร์เฟซยังรวมเมธอดและแอททริบิวต์ช่วยเหลือซึ่งสามารถนำไปใช้ได้กับชนิดของโหนดบางชนิดเท่านั้น

พรีอ็อปเพอร์ตี้	คำอธิบาย
nodeName	ชื่อของโหนด (คือค่าที่แตกต่างกันโดยขึ้นอยู่กับ nodeType ซึ่งจะแสดงในตารางต่อไป)
nodeValue	ค่าของโหนด (คือค่าที่แตกต่างกันโดยขึ้นอยู่กับ nodeType ซึ่งจะแสดงในตารางต่อไป)
nodeType	ชนิดของโหนด จะเป็นหนึ่งในค่าจากตารางต่อไป
parentNode	โหนดซึ่งเป็นโหนดแม่
childNodes	NodeList จะประกอบไปด้วยโหนดซึ่งเป็นลูกทั้งหมด ถ้าไม่มีลูกอยู่เลย NodeList ที่ว่างเปล่าจะถูกคืนกลับ โดยมีค่าที่ไม่เป็น NULL
firstChild	เป็นโหนดลูกอันดับแรก ถ้าไม่มีลูก คืนค่า NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พรีอปเพอร์ดี	คำอธิบาย
lastChild	เป็นโหนดลูกอันดับสุดท้าย ถ้าไม่มีลูก คืนค่า NULL
previousSibling	โหนดที่อยู่ก่อนโหนดนี้ ถ้าไม่มีโหนดที่อยู่ก่อน คืนค่า NULL
nextSibling	โหนดที่ตามหลังโหนดนี้ ถ้าไม่มีโหนดที่อยู่ก่อน คืนค่า NULL
attributes	NamedNodeMap ประกอบไปด้วยแอททริบิวต์สืของโหนดนี้ ถ้าโหนดไม่เป็นอิลิเมนต์ คืนค่า NULL
ownerDocument	เอกสารที่ซึ่ง โหนดนี้อยู่
namespaceURI	เนมสเปซ URI ของโหนดนี้จะคืนค่า NULL ถ้าเนมสเปซไม่ได้ถูกระบุ
prefix	คำนำหน้าเนมสเปซของโหนดนี้ ถ้าไม่มีเนมสเปซที่กำหนดไว้จะคืนค่า NULL
localName	จะคืนค่าส่วนท้องถิ่นของ QName ของโหนดนี้

ค่าของพรีอปเพอร์ดีของ nodeName และ nodeValue ขึ้นอยู่กับค่าของพรีอปเพอร์ดีของ nodeType ซึ่งสามารถที่จะคืนค่าหนึ่งในค่าคงที่ดังต่อไปนี้

พรีอปเพอร์ดีค่าคงที่ของ nodeType	พรีอปเพอร์ดี nodeName	พรีอปเพอร์ดี nodeValue
ELEMENT_NODE	ชื่อแท็ก	NULL
ATTRIBUTE_NODE	ชื่อแอททริบิวต์	ค่าของแอททริบิวต์
TEXT_NODE	#text	เนื้อหาของแท็กซ์โหมด
CDATA_SECTION_NODE	#cdata-section	เนื้อหาของเซกชัน CDATA
ENTITY_REFERENCE_NODE	ชื่อของเอนติตี้ที่ถูกอ้างอิง	NULL
ENTITY_NODE	ชื่อเอนติตี้	NULL
PROCESSING_INSTRUCTION_NODE	เป้าหมาย	เนื้อหาทั้งหมดที่ไม่รวมเป้าหมาย
COMMENT_NODE	#comment	เนื้อหาของคอมเมนต์
DOCUMENT_NODE	#document	NULL
DOCUMENT_TYPE_NODE	ชื่อชนิดของเอกสาร	NULL
DOCUMENT_FRAGMENT_NODE	#document-fragment	NULL
NOTATION_NODE	ชื่อหมายเหตุ	NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด	คำอธิบาย
insertBefore(newChild, refChild)	แทรกโหนด newChild ก่อน refChild ที่มีอยู่ ถ้า refChild เป็น NULL แทรกโหนดที่ท้ายสุดของลิสต์ คืนค่าโหนดที่ได้ทำการแทรก
replaceChild(newChild, oldChild)	แทนที่ oldChild ด้วย newChild คืนค่า oldChild
removeChild(oldChild)	ลบ oldChild ออกจากลิสต์ คืนค่า oldChild
appendChild(newChild)	เพิ่ม newChild ที่ท้ายสุดของลิสต์ คืนค่า oldChild
hasChildNodes()	คืนค่าบูลีน true ถ้าโหนดนั้นมีลูกหรือถ้าไม่ใช่ก็จะเป็น false
cloneNode(deep)	คืนค่าสำเนาของโหนดนี้ ถ้าบูลีนของพารามิเตอร์ deep เป็น true มันก็จะทำการ โคลนแบบเวียนเกิดในต้นไม้ย่อยที่อยู่ใต้โหนดนี้ หรือ มิฉะนั้น ก็จะ โคลน โหนดตัวมันเองเท่านั้น
normalize()	ถ้ามี โหนดลูก Text ที่ต่อกันมากมาย (จากการเรียกก่อนหน้าไปยัง Text.splitText()) เมธอดนี้จะรวมมันอีกครั้งหนึ่ง จะไม่มีการคืนค่า
Supports(feature, version)	ระบุว่าการใช้งานของ DOM นั้นสนับสนุน feature ที่ส่งผ่านมาให้หรือไม่ คืนค่าบูลีน true ถ้าสนับสนุนมิฉะนั้นก็จะเป็น false

### อินเตอร์เฟซ Document

อินเตอร์เฟซ Document จะแทนที่เอกสาร XML ทั้งฉบับ-ออบเจกต์นี้ยังถูกใช้ในการสร้างโหนดอื่นๆ ขณะรันไทม์อีกด้วย อินเตอร์เฟซ Document ขยายอินเตอร์เฟซ Node

พร็อพเพอร์ตี้	คำอธิบาย
doctype	คืนค่าออบเจกต์ DocumentType ซึ่งชนิดของเอกสารที่สัมพันธ์กับเอกสารนี้ ถ้าเอกสาร ไม่มีการระบุชนิดของเอกสารคืนค่า NULL
implementation	ออบเจกต์ DOMImplementation ที่ใช้สำหรับเอกสารนี้
documentDocument	อิลิเมนต์รากสำหรับเอกสารนี้

เมธอด	คำอธิบาย
createElement(tagName)	สร้างอิลิเมนต์ด้วยชื่อระบุ
createDocumentFragment()	สร้างออบเจกต์ DocumentFragment ที่ว่างเปล่า
createTextNode(data)	สร้างโหนด Text ที่ประกอบด้วยข้อความใน data
createComment(data)	สร้างโหนด Comment ที่ประกอบด้วยข้อความใน data
createCDATASection(data)	สร้างโหนด CDATASection ที่ประกอบด้วยข้อความใน data
createProcessingInstruction(target, data)	สร้างโหนด ProcessingInstruction พร้อมด้วย target และ data ที่ระบุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พรีอ็อปเพอเรเตอร์	คำอธิบาย
createAttribute(name)	สร้างแอททริบิวต์ด้วย name ที่ระบุ
createEntityReference(name)	สร้างการอ้างอิงของเอนทิตี ด้วย name ที่ระบุ
getElementsByTagName (tagName)	คืนค่า NodeList ของอิลิเมนต์ทั้งหมดในเอกสารด้วย tagName ถูกคืนค่าในลำดับเอกสาร
importNode (importedNode, deep)	นำเข้าโหนด importedNode จากเอกสารอื่นเข้ามาในเอกสารนี้ โหนดต้นฉบับจะไม่ถูกลบออกจากเอกสารเก่าทันทีเพียงแต่ถูกโคลนเท่านั้น (บุคลิของพารามิเตอร์ deep นั้น ต้นไม้ย่อยที่อยู่ในระดับใต้โหนดนี้จะถูกโคลนด้วย แบบ shallow จะมีเพียง โหนดตัวมันเองเท่านั้นที่ถูกโคลน) คืนค่าเป็น โหนดใหม่
createElementNS (namespaceURI, qualifiedName)	สร้างอิลิเมนต์ด้วยเนมสเปซและ QName ที่ระบุ
createAttributeNS (namespaceURI, qualifiedName)	สร้างแอททริบิวต์ด้วยเนมสเปซและ QName ที่ระบุ
getElementsByTagNameNS (namespaceURI, localName)	คืนค่า NodeList ของอิลิเมนต์ทั้งหมดในเอกสารที่มีชื่อท้องถิ่นที่ระบุ และอยู่ในเนมสเปซที่ระบุ โดย namespaceURI
getElementById(elementID)	คืนค่าอิลิเมนต์ด้วย ID ที่ระบุใน elementID ถ้ามันไม่ใช่อิลิเมนต์ มันจะคืนค่า NULL

หมายเหตุ : เมธอด createXXX() ทั้งหมดคืนค่าเป็นโหนดที่ถูกสร้างขึ้นมา

### อินเตอร์เฟซ DOMImplementation

อินเตอร์เฟซ DOMImplementation จะเตรียมเมธอดซึ่งไม่ได้ระบุไปยังเอกสารใดๆ แต่ระบุไปยังเอกสารที่ใช้งาน DOM นี้ คุณสามารถนำออบเจกต์ DOMImplementation จากพรีอ็อปเพอเรเตอร์ implementation ของอินเตอร์เฟซ Document มาใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด	คำอธิบาย
hasFeature(feature, version)	คืนค่าบูลีน โดยจะชี้ให้เห็นการใช้งานของ DOM ว่าสนับสนุน feature ที่ร้องขอหรือไม่ , version เป็นหมายเลขเวอร์ชันของฟีเจอร์ที่ทดสอบ
createDocumentType(qualifiedName, publicID, systemID, internalSubset)	สร้างออบเจกต์ DocumentType ด้วยแอททริบิวต์ที่ระบุ
createDocument(namespaceURI, qualifiedName, doctype)	สร้างออบเจกต์ Document ด้วยอิลิเมนต์ของเอกสารที่ระบุโดย qualifiedName หรือปเพอร์ดี doctype ต้องอ้างอิงไปยังออบเจกต์ที่มีชนิดเป็น DocumentType

**อินเตอร์เฟซ DocumentFragment**

อินเตอร์เฟซ DocumentFragment เป็นที่ใช้ในการเก็บข้อมูลชั่วคราวสำหรับกลุ่มของโหนด ปกติแล้วจะใช้ในการแทรกโหนดเหล่านี้กลับไปนอกเอกสารจากจุดที่เอามา

อินเตอร์เฟซ DocumentFragment ขยายอินเตอร์เฟซ Node โดยที่ไม่มีพรีอเพอร์ดีหรือเมธอดเพิ่มเติม

**อินเตอร์เฟซ NodeList**

อินเตอร์เฟซ NodeList ประกอบไปด้วยกลุ่มของโหนดที่เรียงลำดับ โดยเข้าถึงผ่านทางดัชนีจำนวนเต็ม (integral index)

พรีอเพอร์ดี	คำอธิบาย
length	จำนวนของโหนดที่ประกอบอยู่ในลิสต์นี้ ช่วงระยะของโหนดลูกที่ใช้ได้จะมีดัชนีเป็น 0 ไปจนถึง length -1 โดยรวม
item(index)	คืนค่า Node ภายในลิสต์ โดยใช้ index มีค่าเท่ากับหรือมากกว่าค่าของ length คืนค่าเป็น NULL

**อิลิเมนต์ Element**

จัดเตรียมพรีอเพอร์ดีและเมธอดสำหรับการทำงานกับอิลิเมนต์

อินเตอร์เฟซ Element ขยายอินเตอร์เฟซ Node

พรีอเพอร์ดี	คำอธิบาย
tagname	ชื่อของอิลิเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด	คำอธิบาย
getAttribute(name)	คืนค่าของแอททริบิวต์ด้วย name ที่ระบุ หรือสตริงที่ว่างเปล่าถ้าแอททริบิวต์นั้น ไม่มีค่าหรือค่าคือฟอลด์ระบุ
setAttribute(name, value)	ตั้งค่าของแอททริบิวต์ที่ระบุไปเป็น value ใหม่ ถ้าไม่มีแอททริบิวต์ดังกล่าวอยู่ ก็จะสร้างแอททริบิวต์ใหม่ตาม name
removeAttribute(name)	ลบแอททริบิวต์ที่ระบุ ถ้าแอททริบิวต์มีค่าคือฟอลด์ มันก็จะถูกแทนที่โดยทันทีด้วยแอททริบิวต์ที่เหมือนกัน ที่ประกอบด้วยด้วยค่าคือฟอลด์นี้
getAttributeNode(name)	คืนค่าโหนด Attr ที่ประกอบด้วยแอททริบิวต์ name คืนค่า NULL ถ้าไม่มีแอททริบิวต์ดังกล่าว
setAttributeNode(newAttr)	เพิ่มแอททริบิวต์โหนดใหม่ ถ้ามีแอททริบิวต์ที่มีชื่อเดียวกันอยู่ มันก็จะถูกแทนที่ถ้า Attr นั้นถูกแทนที่มันจะถูกคืนค่าหรือมีจะนั้นคืนค่า NULL
removeAttributeNode(oldAttr)	ลบ โหนด Attr ที่ระบุแล้วคืนค่ามัน ถ้าแอททริบิวต์มีค่าคือฟอลด์มันจะถูกแทนที่ โดยทันทีด้วยแอททริบิวต์ที่เหมือนกันที่ประกอบด้วยค่าคือฟอลด์นี้
getElementsByTagName(name)	คืนค่า NodeList ของการสืบทอดทั้งหมดของ โหนด name ที่ให้มา
getAttributeNS(namespaceURI, localName)	คืนค่าของแอททริบิวต์ที่ระบุหรือสตริงว่างเปล่าถ้าแอททริบิวต์นั้น ไม่มีค่าที่ระบุหรือค่าคือฟอลด์
setAttributeNS(namespaceURI, QualifiedName, value)	ตั้งค่าของแอททริบิวต์ที่ระบุไปเป็น value ใหม่ ถ้า ไม่มีแอททริบิวต์ดังกล่าวอยู่ แอททริบิวต์ใหม่พร้อมด้วยนามสเปซ URI และ QName จะถูกสร้างขึ้น
removeAttributeNS(namespaceURI, localName)	ลบแอททริบิวต์ที่ระบุถ้าแอททริบิวต์มีค่าคือฟอลด์มันก็จะถูกแทนที่โดยทันทีด้วยแอททริบิวต์ที่เหมือนกัน ที่ประกอบด้วยค่าคือฟอลด์นี้
getAttributeNodeNS(namespaceURI, localName)	คืนค่าโหนด Attr ที่ประกอบด้วยแอททริบิวต์ที่ระบุ คืนค่า NULL ถ้าไม่มีแอททริบิวต์ดังกล่าวอยู่
setAttributeNodeNS(newAttr)	เพิ่ม โหนด Attr ใหม่ไปยังลิสต์ ถ้าแอททริบิวต์พร้อมด้วยนามสเปซ URI ที่เหมือนกันและชื่อท้องถิ่นนั้นมีอยู่ มันก็จะถูกแทนที่ ถ้าออบเจกต์ Attr ถูกแทนที่มันก็จะถูกคืนค่าหรือมีจะนั้นคืนค่า NULL
getAttributeByTagNameNS(namespaceURI, localName)	คืนค่า NodeList ของอิลิเมนต์ทั้งหมดที่ตรงกับข้อกำหนดเหล่านี้

## อินเทอร์เฟซ NameNodeMap

เม็ปของโหนดที่มีชื่อหรือ named node map นั้นจะใช้แสดงคอลเลกชันของโหนดที่ไม่เรียงลำดับ ถูกค้นคืนด้วยชื่อ

พรีอ็อปเพอเรเตอร์	คำอธิบาย
length	จำนวนของโหนดในเม็ป

เมธอด	คำอธิบาย
getNamedItem(name)	คืนค่า Node ซึ่ง nodeName เหมือนกับ name ที่กำหนดให้ หรือเป็น NULL ถ้าโหนดดังกล่าวไม่มีอยู่
setNamedItem(arg)	พารามิเตอร์ arg เป็นออบเจกต์ Node ออบเจกต์หนึ่งที่ถูกเพิ่มเข้ามาในลิสต์พรีอ็อปเพอเรเตอร์ nodeName ถูกใช้สำหรับชื่อของโหนดในเม็ปนี้ ถ้าโหนดที่มีชื่อเดียวกันมีอยู่แล้ว มันจะถูกแทนที่ ถ้า Node ที่กำหนดด้วย name และคืนค่ามัน
removeNamedItem(name)	ลบ Node ที่กำหนดด้วย name และคืนค่ามัน
item(index)	คืนค่า Node ณ index ที่กำหนด ถ้า index เท่ากันหรือมากกว่า length จะคืนค่า NULL
getNameItemNS(namespaceURI, localName)	คืนค่า Node โดยการจับคู่เนมสเปซ URI กับชื่อโลคอล หรือ NULL ถ้าโหนดที่ว่าไม่มีอยู่
setNameItemNS(arg)	พารามิเตอร์ arg เป็นออบเจกต์ Node ออบเจกต์หนึ่งที่ถูกเพิ่มเข้ามาในลิสต์ ถ้าโหนดที่มีเนมสเปซ URI และชื่อโลคอลเดียวกันมีอยู่แล้ว มันจะถูกแทนที่ ถ้า Node นั้นถูกแทนที่ มันจะถูกคืนค่า หรือมิฉะนั้นคืนค่า NULL
removeNamesItemNS(namespaceURI, localName)	ลบโหนดที่กำหนด แล้วคืนค่ามัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### อินเตอร์เฟซ Attr

จัดเตรียมพรีอ็อปเพอร์ตีสำหรับการติดต่อกับแอททริบิวต์

อินเตอร์เฟซ Attr ขยายอินเตอร์เฟซ Node

พรีอ็อปเพอร์ตี	คำอธิบาย
name	ชื่อของแอททริบิวต์
specified	บูลีนเป็นการชี้ว่าแอททริบิวต์นั้นถูกกำหนดขึ้น (true) หรือเป็นแอททริบิวต์คีย์โวลด์ (false)
value	ค่าของแอททริบิวต์
ownerElement	ออบเจกต์ Element จะแสดงถึงอีลิเมนต์ซึ่งแอททริบิวต์นี้อยู่

### อินเตอร์เฟซ CharacterData

จัดเตรียมพรีอ็อปเพอร์ตีและเมธอดสำหรับการทำงานกับข้อมูลตัวอักษร

อินเตอร์เฟซ CharacterData ขยายอินเตอร์เฟซ Node

พรีอ็อปเพอร์ตี	คำอธิบาย
data	ข้อความในโหนด CharacterData
length	จำนวนอักขระในโหนด

เมธอด	คำอธิบาย
substringData(offset, count)	คืนค่าส่วนของสตริง โดยเริ่มต้นที่ offset ซึ่งจะคืนค่าจำนวนอักขระที่ระบุใน count หรือจนกระทั่งหมดสตริงซึ่งขึ้นอยู่กับว่าสิ่งใดมีค่าน้อยกว่ากัน
appendData(arg)	เพิ่มอักขระใน arg ในส่วนท้ายของสตริง
insertData(offset, arg)	แทรกสตริงใน arg ในส่วกลางของสตริง โดยเริ่มต้นที่ตำแหน่งที่ระบุโดย offset
deleteData(offset, count)	ลบส่วนของสตริง โดยเริ่มต้นที่ offset จะลบจำนวนอักขระที่ระบุใน count หรือจนกระทั่งหมดสตริง ซึ่งขึ้นอยู่กับว่าสิ่งใดมีค่าน้อยกว่ากัน
replaceData(offset, count, arg)	แทนที่ส่วนของสตริง โดยเริ่มต้นที่ offset ซึ่งจะแทนที่จำนวนอักขระที่ระบุใน count หรือจนกระทั่งหมดสตริง ซึ่งขึ้นอยู่กับว่าสิ่งใดมีค่าน้อยกว่ากัน โดยพารามิเตอร์ arg ก็คือสตริงใหม่ที่จะถูกแทรกเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### อินเทอร์เฟซ Text

จัดเตรียมเมธอดเพิ่มเติมสำหรับการทำงานกับ โหนดข้อความ (เท็กซ์ โหนด)

อินเทอร์เฟซ Text ขยายอินเทอร์เฟซ CharacterData

เมธอด	คำอธิบาย
splitText	แยก โหนด Text เดียวออกเป็น โหนด Text สองส่วนด้วยกัน ข้อความทั้งหมดก่อนถึงตำแหน่ง offset จะ ไปเป็น โหนด Text แรก ข้อความทั้งหมดโดยเริ่มต้นที่ตำแหน่ง offset จนถึง ตำแหน่งสุดท้ายแล้วจะไปเป็น โหนด Text ที่สอง

### อินเทอร์เฟซ Comment

จะทำการเอนแคปซูเลต (ถ่ายถอดคุณสมบัติ) คอมเมนต์ของ XML

อินเทอร์เฟซ Comment ขยายอินเทอร์เฟซ CharacterData โดยไม่มีพรีอ็อปเพอเรเตอร์หรือเมธอดเพิ่มเติม

### อินเทอร์เฟซส่วนขยาย

อินเทอร์เฟซส่วนขยายของ DOM Extend จำเป็นต้องถูกเตรียมโดยการทำงานของ DOM ซึ่งจะทำงานกับเอกสารของ XML

### อินเทอร์เฟซ CDATASection

จะทำการเอนแคปซูเลต (ถ่ายถอดคุณสมบัติ) ส่วน CDATA ของ XML

อินเทอร์เฟซ CDATASection จะขยายอินเทอร์เฟซ Text โดยที่ไม่มีพรีอ็อปเพอเรเตอร์หรือเมธอดใดๆเพิ่ม

### อินเทอร์เฟซ ProcessingInstruction

จัดเตรียมพรีอ็อปเพอเรเตอร์สำหรับการทำงานกับ ProcessingInstruction (PI) ใน XML

อินเทอร์เฟซ ProcessingInstruction ขยายอินเทอร์เฟซ Node

พรีอ็อปเพอเรเตอร์	คำอธิบาย
target	PI เป้าหมาย หรือ ในอีกความหมายหนึ่งก็คือชื่อแอปพลิเคชันซึ่ง PI จะต้องถูกส่งผ่าน เนื้อหารายละเอียดของ PI
data	เนื้อหาของ PI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### อินเทอร์เฟซ DocumentType

จัดเตรียมพร้อมเพอร์ตีสำหรับทำงานกับชนิดเอกสารของ XML สามารถที่จะถูกค้นคืนจากพร้อมเพอร์ตี doctype ของอินเทอร์เฟซ Document (ถ้าเอกสารไม่มีชนิดของเอกสาร doctype จะคืนค่า NULL)

DocumentType ขยายอินเทอร์เฟซ Node

พร้อมเพอร์ตี	คำอธิบาย
name	ชื่อของ DTD
entities	NamedNodeMap ที่บรรจุเอนติตี้ทั้งหมดที่ประกาศไว้ใน DTD (ภายในและภายนอก) เอนติตี้ของพารามิเตอร์จะไม่ถูกบรรจุไว้ และเอนติตี้ที่ซ้ำซ้อนจะถูกยกเลิก โดยจะขึ้นกับกฎที่ใช้โดย XML parser
notations	NamedNodeMap ที่บรรจุบันทึกย่อ (notation) ที่บรรจุอยู่ใน DTD ส่วนที่ซ้ำซ้อนจะถูกยกเลิก
publicID	ตัวบ่งชี้ (id) สาธารณะของสับเซตภายนอก
systemID	ตัวบ่งชี้ (id) ระบบของสับเซตภายนอก
internalSubset	สับเซตภายในในรูปแบบของสตริง

### อินเทอร์เฟซ Notation

จัดเตรียมคุณสมบัติสำหรับทำงานกับบันทึกย่อ (notation) ของ XML โดยบันทึกย่อนี้จะมีลักษณะอ่านได้เพียงอย่างเดียวใน DOM

อินเทอร์เฟซ Notation ขยายอินเทอร์เฟซ Node

พร้อมเพอร์ตี	คำอธิบาย
publicID	ตัวบ่งชี้ (id) สาธารณะของ Notation นี้ ถ้าไอเดนติไฟเลอร์สาธารณะไม่ได้ถูกระบุจะคืนค่า NULL
systemID	ตัวบ่งชี้ (id) ระบบของ Notation ถ้าตัวบ่งชี้ (id) สาธารณะไม่ได้ถูกระบุจะคืนค่า NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### อินเทอร์เฟซ Entity

จัดเตรียมคุณสมบัติสำหรับการทำงานกับเอนทิตีที่ parse แล้วและยังไม่ได้ parse โหนด Entity มีลักษณะอ่านได้อย่างเดียว

อินเทอร์เฟซ Entity ขยายอินเทอร์เฟซ Node

พรีอเพอร์ตี้	คำอธิบาย
publicID	ตัวบ่งชี้ (id) สาธารณะที่สัมพันธ์กับเอนทิตี หรือ NULL ถ้าไม่มีการระบุไว้
systemID	ตัวบ่งชี้ (id) ระบบที่สัมพันธ์กับเอนทิตี หรือ NULL ถ้าไม่มีการระบุไว้
notationName	สำหรับเอนทิตีที่ยังไม่ได้ parse จะเป็นชื่อของ notation สำหรับเอนทิตี, จะเป็น NULL สำหรับ เอนทิตีที่ parse แล้ว

### อินเทอร์เฟซ EntityReference

จะเอนแคปซูเลต (ถ้ายกออกคุณสมบัติ) เอนทิตีอ้างอิงของ XML

อินเทอร์เฟซ EntityReference ขยายอินเทอร์เฟซ Node โดยไม่มีพรีอเพอร์ตี้หรือเมธอดใดๆ

เพิ่มเติม



## บรรณานุกรม

- [1] ฮันเตอร์, เดวิท (2002) : “คัมภีร์การใช้เอกซเอนแอล ฉบับสมบูรณ์”, ซีเอ็ดยูเคชั่น, กรุงเทพฯ
- [2] แมคเคลเลอร์, สจ๊วต (2003) : “การโจมตีและเทคนิคการป้องกันเว็บไวด์”, ซีเอ็ดยูเคชั่น, กรุงเทพฯ
- [3] สุวัฒนา สุขสมจินต์ (2003) : “คัมภีร์การใช้ Visual C# ฉบับสมบูรณ์”, ซีเอ็ดยูเคชั่น, กรุงเทพฯ
- [4] ศุภชัย สมพานิช 2546 : “คู่มือการเขียนโปรแกรม Visual C#.Net ฉบับโปรแกรมเมอร์”, อินโฟเพรส, กรุงเทพฯ
- [5] มณีโชติ สมานไทย 2546 : “การเขียนโค้ด ASP.Net ฉบับสมบูรณ์”, อินโฟเพรส, กรุงเทพฯ
- [6] ยี่งยศ สันติประเสริฐ 2546 : “เรียนรู้ ASP.NET ด้วยภาษา C#”, ซีเอ็ดยูเคชั่น, กรุงเทพฯ
- [7] เรืองไร รังสิพล 2544 : “เปิดโลก Firewall”, โปรวิชั่น, กรุงเทพฯ
- [8] มหาวิทยาลัยสุโขทัยธรรมาธิราช 2544 : “การจัดการระบบฐานข้อมูล”, สำนักพิมพ์มหาวิทยาลัยสุโขทัยธรรมาธิราช
- [9] สมพร จิวรสกุล 2545 : “คู่มือการติดตั้ง และใช้งาน Microsoft SQL Server 2000 ฉบับสมบูรณ์”, อินโฟเพรส, กรุงเทพฯ
- [10] นาย ธนาพล อีสวกุล, นาย ธนพงศ์ จารุสริระกุล 2543 : “การประกวดราคอบนอินเทอร์เน็ต”, ปริญญานิพนธ์ สจล., กรุงเทพฯ
- [11] Russ Basiura, Mike Batong, Brandon Bohling, Mike Clake, Andreas Eide, Robert Eisenberg, Kavin Hofman, Brian Loesgen, Chris Miller, Matthew Reynolds, Bill Semof, Srinivasa Sivakumar (2002): “Professional ASP.NET Web Services”, Wrox, UK
- [12] Ivar Jakocobson, Grady Booch, James Rumbaugh (1998) : “The Unified Software Development Process”, Addison Wesley, UK
- [13] msdn 2002 : “Developing Microsoft ASP.NET Web Application Using Visual Studio.NET Delivery Guide”, Microsoft Corporation, USA
- [14] <http://www.trl.ibm.com/projects/xml/xacl/xmlac-proposal.html>
- [15] <http://www.thescarms.com/XML/DOMTutorial.asp>
- [16] <http://www.w3schools.com/dom/default.asp>
- [17] <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnxml/html/xml092099.asp>
- [18] <http://www.w3.org>