

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาระบบพาสปอร์ตโดยใช้เทคโนโลยีอินเทอร์เน็ต  
Passport System Development using .NET Technology



1. นาย ธนวุฒิ เข้มทอง
2. นาย ภาณุรักษ์ มูลขำ

อาจารย์ที่ปรึกษา  
รศ.บรรจง ปิยทาร์ง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

b.....  
i.....

เลขหมู่.....

เลขทะเบียน.....49943

วันเดือนปี 2 เม.ย. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาระบบพาสปอร์ตโดยใช้เทคโนโลยีอินเทอร์เน็ต  
Passport System Development using .NET Technology



1. นาย ธนวุฒิ เข้มทอง
2. นาย ภาณุรักษ์ มุลขำ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

61137603x

ปริญญาโท ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาระบบพาสปอร์ตโดยใช้เทคโนโลยีอินเทอร์เน็ต

Passport System Development using .NET Technology

ผู้จัดทำ

1. นาย ธนวุฒิ เข้มทอง รหัสประจำตัว 43015358

2. นาย ภาณุรักษ์ มูลขำ รหัสประจำตัว 43015377



อาจารย์ที่ปรึกษา

(รศ. บรรจง ปิยะธำรง)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เรื่อง การพัฒนาระบบพาสปอร์ตโดยใช้เทคโนโลยีอินเทอร์เน็ต

นาย ธนวุฒิ เข้มทอง

นาย ภาณุรักษ์ มุลขำ

รศ. บรรจง ปิยธำรง อาจารย์ที่ปรึกษา

ปีการศึกษา 2545

### บทคัดย่อ

การนำเทคโนโลยีของคอมพิวเตอร์มาใช้งานในเชิงประยุกต์มิได้จำกัดวงอยู่แต่ในห้องทดลองทางวิทยาศาสตร์เท่านั้น หากแต่ได้ก้าวเข้ามาเป็นส่วนหนึ่งของชีวิตประจำวันของคนทั่วไป ปัจจัยหลักมาจากอินเทอร์เน็ตที่ได้รับความนิยมอย่างรวดเร็ว จนทำให้เกิดนวัตกรรมทางธุรกิจบนอินเทอร์เน็ตในรูปแบบต่างๆ อย่างต่อเนื่อง ไม่ว่าจะเป็นธุรกิจดอทคอม, B2B, B2C

เว็บเซอร์วิส (Web Services) คืออีกผลพวงหนึ่งที่เกิดขึ้น ซึ่งตอบสนองความต้องการการแลกเปลี่ยนข้อมูลที่กระจัดกระจายในแอปพลิเคชันและเว็บไซต์ต่างๆ กันให้ได้อย่างสมบูรณ์ ทำให้สามารถควบคุมเทคโนโลยีและข้อมูลรอบตัวได้ เกิดจากการรวบรวมเนื้อหาและข้อมูล และนำข้อมูลเหล่านั้นมาใช้งานร่วมกันได้อย่างมีประสิทธิภาพ

ระบบ พาสปอร์ต เป็นส่วนพื้นฐานของ Web Services ซึ่งทำหน้าที่เก็บข้อมูลเกี่ยวกับผู้ใช้ออนไลน์ ตั้งแต่ข้อมูลส่วนตัว ไปจนถึงสิทธิของผู้ใช้คนนั้นในการเข้าถึงข้อมูลต่างๆ ในเครือข่าย จะทำให้ผู้ใช้สามารถเข้าถึงข้อมูลได้รวดเร็วและช่วยลดขั้นตอนการซื้อขายออนไลน์ และต้องมีการรักษาความปลอดภัยทางข้อมูลที่ดี

ปัญญานิพนธ์ฉบับนี้ได้สร้างระบบ พาสปอร์ต ขึ้นมาและจำลองร้านค้าออนไลน์ โดยให้ความสำคัญกับการจัดการข้อมูลของสมาชิก ได้รับทราบเพื่อเป็นแนวทางในการพัฒนาและปรับปรุงข้อมูลโดยใช้ SQL Server 2000 และภาษาที่ใช้ในการพัฒนา คือ VB.NET และ ASP.NET ในการพัฒนาดังกล่าว

## Passport System Development using .NET Technology

Tanawut Kemthong

Phanurak Moolkham

Assoc. Prof. Bunjong Piyathumrong Advisor

### ABSTRACT

Using computer technology in term of applications is not limited only in laboratory but it's also participate in our modern life style. The main criteria of this phenomena is the rapidly growing of Internet Technology that affect to innovate many type of business online such as business.com , B2B and B2C.

Web Services are used for the needs of exchanging fragmenting datas in applications and web sites in order to control informations and technologies by gathering the informative and useful datas then apply to use all of them effectively.

Passport system is the basis of Web Services.Its function is to collect all online user's datas such as personal information through user's permissions to access any data in a network. It will allow the users to access the data quickly and reduce the procedure of on-line trading, anyway there must be an effective security system as well.

This Thesis has built up this Passport System and the online shop simulation model by emphasize the member's data management in order to be a sample prototype for anyone to develop and improve later by SQL Server 2000, VB.NET and ASP.NET as the application tools

### กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์บรรจง ปิยธำรง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ และให้คำแนะนำเสมอมา

ขอขอบพระคุณบุคลากรและคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้ประสิทธิประสาทความรู้ให้โดยตลอด ขอขอบคุณน้ำใจที่เกิดจากเพื่อนร่วมสถาบัน โดยเฉพาะอย่างยิ่งเพื่อนๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ เพื่อนห้องเน็ตเวิร์ก อีกทั้งเพื่อน พี่ น้องในชุมนุมอิเล็กทรอนิกส์ที่มีกำลังใจคอยเติมเต็มให้เสมอ

สุดท้ายขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นาย ธนะวุฒิ เข้มทอง  
นาย ภาณุรักษ์ มุลขำ

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VIII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 รู้จักกับ Microsoft.net และ Webservice	3
2.1 แนวคิดการทำงานของ .net	3
2.2 เป้าหมายของ Microsoft .net	3
2.3 มาตรฐานที่เกี่ยวข้องกับ Microsoft.net	4
2.4 พัฒนาการก่อนมาเป็น Web services	5
2.5 ความหมายของ Web Services	7
2.5.1 การทำงานของ Web Services	7
บทที่ 3 สถาปัตยกรรมของ .net	8
3.1 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม	8
3.2 สถาปัตยกรรมของ .net	10
3.2.1 เลเยอร์ Common Language Runtime	11
3.2.2 เลเยอร์ Base Class Library	13
3.2.3 เลเยอร์ Common Language Specification	14
บทที่ 4 XML	15
4.1 ข้อดีของ HTML	15
4.2 เป้าหมายหลักของ XML	16
4.3 การประยุกต์ใช้ XML	17

4.4 XML จะแทนที่ HTML หรือไม่	17
4.5 โครงสร้าง XML	18
4.6 ข้อควรระวังในการเขียน XML	19
4.7 วิเคราะห์โครงสร้างของเอกสารแบบ DOM และ SAX	20
4.7.1 การพาร์สเอกสาร XML โดยใช้ DOM (Document Object Model)	20
4.7.2 การพาร์สเอกสาร XML โดยใช้ SAX (Simple API for XML)	21
บทที่ 5 SOAP	24
5.1 แนะนำ SOAP	24
5.2 ส่วนประกอบของ SOAP	24
5.2.1 Envelope	26
5.2.2 Body	27
5.2.3 Header	27
5.3 SOAP Fault Element	28
5.4 SOAP Encoding	29
5.4.1 ตัวอย่าง Struct	30
5.4.2 ตัวอย่าง Array	31
5.5 SOAP ใน HTTP	33
5.6 SOAP สำหรับ RPC	34
5.7 SOAP Toolkit	35
5.8 โครงสร้างภายในของ SOAP Toolkit ของ Microsoft	35
5.8.1 การสร้าง SoapClient Object	36
5.8.2 การประมวลผลภายใน SoapClient Object	37
5.8.3 การสร้าง SoapServer Object	37
5.8.4 การประมวลผลภายใน SoapServer Object	39
5.9 การสร้าง SOAP Client และ SOAP Server ด้วย .NET Framework	40
5.9.1 การใช้ ASP.NET ทำ SOAP Web Service	40
5.9.2 การใช้ .NET ทำ SOAP Client	40
บทที่ 6 เทคโนโลยี ASP.net	42
6.1 ความเปลี่ยนแปลงที่เกิดขึ้นของ ASP.net	42
6.1.1 นามสกุลของไฟล์ และการแยกส่วนออกแบบกับส่วนเขียนโค้ด	42
6.1.2 การคอมไพล์โปรแกรม	43

6.2 คอนโทรลที่ใช้ออกแบบในแอปพลิเคชัน ASP.net	43
6.2.1 คอนโทรล HTML	44
6.2.2 คอนโทรล Web Forms	45
6.2.3คอนโทรล Customer	45
6.3 คอนโทรลที่ทำงานฝั่งเซิร์ฟเวอร์	46
6.4 การเขียนโปรแกรมใน ASP.net	47
6.5 สรุปตารางเปรียบเทียบระหว่าง ASP กับ ASP.net	47
บทที่ 7 ADO.net	48
7.1 รู้จัก ADO.net	48
7.2 ออบเจกต์ที่สำคัญใน ADO.net	49
7.2.1 ออบเจกต์ Connection	49
7.2.2 ออบเจกต์ Command	50
7.2.3 ออบเจกต์ Dataset	51
7.2.4 ออบเจกต์ DataAdapter	52
7.3 การสนับสนุนมาตรฐาน XML	53
7.4 สรุปตารางเปรียบเทียบ ระหว่าง ADO กับ ADO.net	53
บทที่ 8 การออกแบบ	54
8.1 ระบบร้านค้า online(Use case diagram)	54
8.2 ระบบ passport (Use case diagram)	56
8.3 Sequence diagram	58
8.4 การออกแบบฐานข้อมูล	74
บทที่ 9 Passport System	79
9.1 หลักการทำงาน	79
9.2 การเก็บรวบรวมข้อมูลส่วนตัวโดย PASSPORT	79
9.2.1 ข้อมูลชนิดใดบ้างที่ Passport เก็บรวบรวม	79
9.2.2 การกำหนดข้อมูลที่จะเก็บรวบรวม โดยไซท์ที่ทำการลงทะเบียน	79
9.3 การบริหารจัดการข้อมูลส่วนตัวของคุณ	80
9.3.1 การเข้าไปใช้ข้อมูลส่วนตัวของคุณ	80
9.3.2 การปิดบัญชี Passport ของคุณ	80
9.3.3 ความปลอดภัยและแหล่งเก็บข้อมูลส่วนตัว	80
9.3.4 การใช้งานคูปอง	80

9.4 ส่วนตัวระบบ	81
9.4.1 ความปลอดภัย	82
9.4.2 การใช้ html forms	82
9.4.3 การใช้ Cookies	82
9.4.4 การใช้ HTTP REFERER Header	83
9.4.5 การใช้ POST \$ GET method	83
9.5 Error Handling Mechanism	83
9.6 การติดตั้ง	83
9.7 การใช้งาน	84
9.7.1 หน้าแรก passport system	84
9.7.2 บริการสมาชิก	88
9.7.3 แหล่งรวม เว็บไซต์	89
9.7.4 สิทธิส่วนบุคคล	89
บทที่ 10 ระบบซื้อขายผ่านอินเทอร์เน็ต	90
10.1 ระบบซื้อขาย อุปกรณ์ คอมพิวเตอร์ บนอินเทอร์เน็ต	90
10.2 การใช้งาน เว็บ ระบบการซื้อขาย อุปกรณ์ คอมพิวเตอร์ บนอินเทอร์เน็ต	90
10.2.1 เมนูหลัก	91
10.2.2 ส่วนการค้นหาอุปกรณ์	92
10.2.3 รายการสั่งซื้อ	94
10.2.4 เว็บบอร์ด	97
บทที่ 11 วิจัยและสรุป	99
11.1 ส่วนการศึกษาระบบที่มีอยู่เดิม	99
11.1.1 ส่วนการวิเคราะห์และการออกแบบระบบ	99
11.1.2 ส่วนการพัฒนา ระบบ	99
11.1.3 ส่วนวิจัยและบทสรุปโดยรวมของระบบ	100
11.2 ปัญหาและแนวทางในการพัฒนาระบบ	100
11.2.1 ปัญหาเกี่ยวกับแหล่งอ้างอิง	100
11.2.2 การเพิ่มในส่วนของ Wallet	100
11.2.3 การรักษาความปลอดภัยของข้อมูล	100
บรรณานุกรม	101

## สารบัญรูปภาพ

	หน้า
บทที่ 2 รู้จักกับ Microsoft.net และ Webservice	3
รูปที่ 2-1 พัฒนาการก่อนมาเป็น Web services	5
รูปที่ 2-2 ยุคของการพัฒนาแอปพลิเคชันต่างๆ	6
รูปที่ 2-3 รูปแบบการทำงานของ Web Services	7
บทที่ 3 สถาปัตยกรรมของ .net	8
รูปที่ 3-1 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม แบบแอปพลิเคชันต่างๆ	8
รูปที่ 3-2 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม แบบ Component Object Model	9
รูปที่ 3-3 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม แบบ .NET	9
รูปที่ 3-4 โครงสร้างการคอมไพล์จาก Visual Studio.net	10
รูปที่ 3-5 โครงสร้างสถาปัตยกรรมของ .net	10
รูปที่ 3-6 โครงสร้าง เลเยอร์ Common Language Runtime	11
รูปที่ 3-7 การออกแบบเพื่อสนับสนุนการทำงานร่วมกันใน Common Language Runtime	12
รูปที่ 3-8 โครงสร้าง เลเยอร์ Base Class Library	13
รูปที่ 3-9 โครงสร้าง เลเยอร์ Common Language Specification	14
บทที่ 4 XML	15
รูปที่ 4-1 ตัวอย่างไฟล์ XML	18
รูปที่ 4-2 ผลลัพธ์จาก Browse ดูในโปรแกรม Internet Explorer	19
รูปที่ 4-3 ตัวอย่างไฟล์ XML	20
รูปที่ 4-4 DOM Tree	21
รูปที่ 4-5 ตัวอย่างไฟล์ XML	22
รูปที่ 4-6 SAX Event	22
บทที่ 5 SOAP	24
รูปที่ 5-1 โครงสร้างของ SOAP	24
รูปที่ 5-2 แสดงการ Client –Side Data Flow	36
รูปที่ 5-3 แสดงโครงสร้างภายในของ SOAP	37
รูปที่ 5-4 แสดง Server-Side Data Flow	38

รูปที่ 5-5 แสดง โครงสร้างภายใน SOAP Server	39
รูปที่ 5-6 การใช้ ASP.NET ทำ SOAP web service	40
รูปที่ 5-7 Runtime Callable Wrapper	40
รูปที่ 5-8 การใช้ .NET ทำ SOAP client	40
รูปที่ 5-9 COM callable wrapper ที่ทำให้ COM client สามารถเห็น .NET object	41
บทที่ 6 เทคโนโลยี ASP.net	42
รูปที่ 6-1 แสดงเว็บแอปพลิเคชันที่สร้างด้วย ASP.net จะแยกไฟล์สำหรับออกแบบ(file.aspx) กับไฟล์สำหรับเขียนโค้ดโปรแกรมจากกัน(file.aspx.cs)	42
รูปที่ 6-2 คอนโทรล HTML ที่เราสามารถเลือกสร้างได้จาก Tool Box	44
รูปที่ 6-3 คอนโทรล Web Forms ที่เราสามารถเลือกสร้างได้จาก Tool Box	45
บทที่ 7 ADO.net	48
รูปที่ 7-1 โครงสร้างของ ADO	48
รูปที่ 7- 2 Data Link Properties	49
รูปที่ 7-3 Query Builder แสดงการใช้รูปแบบคำสั่ง	50
รูปที่ 7-4 Dataset	51
บทที่ 8 การออกแบบ	54
รูปที่ 8-1 แสดง Use Case Diagram ระบบร้านค้า online	54
รูปที่ 8-2 แสดง Use case diagramระบบ passport	56
รูปที่ 8-3 แสดง Sequence diagram ของ Login	58
รูปที่ 8-4 แสดง Sequence diagram ของ Logout	59
รูปที่ 8-5 แสดง Sequence diagram ของ การค้นหาสินค้า	60
รูปที่ 8-6 แสดง Sequence diagram ของ การเปรียบเทียบสินค้า	61
รูปที่ 8-7 แสดง Sequence diagram ของ การสั่งซื้อสินค้า	62
รูปที่ 8-8 แสดง Sequence diagram ของ การสั่งซื้อสินค้า	63
รูปที่ 8-9 แสดง Sequence diagram ของ การเพิ่มรายการสินค้า	64
รูปที่ 8-10 แสดง Sequence diagram ของ การลบรายการสินค้า	65
รูปที่ 8-11 แสดง Sequence diagram ของ การแก้ไขรายการสินค้า	66
รูปที่ 8-12 แสดง Sequence diagram ของ การแก้ไขรายการสินค้า	67
รูปที่ 8-13 แสดง Sequence diagram ของ การสมัครสมาชิก	68
รูปที่ 8-14 แสดง Sequence diagram ของ การเพิ่มเดิมนสินค้า	69

รูปที่ 8-15 แสดง Sequence diagram ของ การลบสินค้า	70
รูปที่ 8-16 แสดง Sequence diagram ของ การแก้ไขสินค้า	71
รูปที่ 8-17 แสดง Sequence diagram ของ การแก้ไขรหัสผ่าน	72
รูปที่ 8-18 แสดง Sequence diagram ของ การแก้ไขประวัติส่วนตัว	73
บทที่ 9 Passport System	79
รูปที่ 9-1 โครงสร้างพื้นฐาน Web sevice	79
รูปที่ 9-2 แสดง sevice ต่างๆใน รูปแบบ asmx ของ ระบบ pass sport	81
รูปที่ 9-3 แสดง sevice ตรวจสอบ e-mialใน รูปแบบ asmx	81
รูปที่ 9-4 แสดงหน้าWeb Passport	84
รูปที่ 9-5 แสดงหน้าให้กรอกที่อยู่อีเมลและรหัสผ่านของคุณ เพื่อทำการลงทะเบียนเป็นสมาชิก	85
รูปที่ 9-6 แสดงหน้ากรอกข้อมูลส่วนตัว เพื่อทำการลงทะเบียนเป็นสมาชิก	85
รูปที่ 9-7 แสดงหน้ากรอกคำถาม เพื่อใช้เมื่อลืมรหัสผ่าน	86
รูปที่ 9-8 แสดงหน้าเสร็จสิ้นการสมัคร	86
รูปที่ 9-9 หน้าการ login เพื่อเข้าสู่ระบบ passport sytem	87
รูปที่ 9-10 หน้าการทำงานของ passport	87
รูปที่ 9-11 แสดงหน้าสำหรับกรอก e-mail เมื่อ ลืมรหัสผ่าน	88
รูปที่ 9-12 เป็นหน้าที่ให้บริการสำหรับสมาชิก	88
รูปที่ 9-13 web ที่เป็นสมาชิกของ passport	89
บทที่ 10 ระบบซื้อขายผ่านอินเทอร์เน็ต	90
รูปที่ 10-1 แสดงหน้าจอหลัก ของ ระบบการซื้อ ขาย อุปกรณ์คอมพิวเตอร์	90
รูปที่ 10-2 เมื่อเราเลือก ชนิดของอุปกรณ์ที่ต้องการจากเมนูหลัก	91
รูปที่ 10-3 แสดงรายการของประเภทสินค้า	91
รูปที่ 10-4 แสดงการเปรียบเทียบ ราคาที่ต่างๆ	92
รูปที่ 10-5 แสดงส่วนการค้นหาแบบรวดเร็ว	92
รูปที่ 10-6 แสดงรายการสินค้าที่ได้จากการค้นหา	93
รูปที่ 10-7 แสดงรายการสินค้าปรากฏว่าไม่พบ	93
รูปที่ 10-8 แสดงเมนู รายการสั่งซื้อ	94
รูปที่ 10-9 แสดงผล เมื่อไม่ได้ทำการ login	94
รูปที่ 10-11 แสดงผลเมื่อทำการ login เข้าสู่ระบบ	95
รูปที่ 10-12 แสดงรายการที่สั่งเพื่อให้ระบุจำนวนที่ต้องการ	96

รูปที่ 10-13 แสดงรายการ การตั้งชื่อสินค้าทั้งหมด	96
รูปที่ 10-14 แสดงหน้าตาของ เว็บไซต์	97
รูปที่ 10-15 แสดงหน้า การตั้งกระทู้ของ เว็บไซต์	97
รูปที่ 10-16 แสดงข้อความภายในของแต่ละกระทู้	98



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 5-1 SOAP Fault Element	28
ตารางที่ 5-2 ค่า faultcode	29
ตารางที่ 5-3 SOAP Toolkit	35
ตารางที่ 6-1 สรุปตารางเปรียบเทียบระหว่าง ASP กับ ASP.net	47
ตารางที่ 7-1 สรุปตารางเปรียบเทียบ ระหว่าง ADO กับ ADO.net	53



## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญและที่มา

เทคโนโลยีอินเทอร์เน็ตนับได้ว่าเป็นเทคโนโลยีที่มีการเปลี่ยนแปลงที่รวดเร็วมาในปัจจุบัน ซึ่งจะเห็นได้ว่าคำว่า อินเทอร์เน็ตเป็นคำที่คุ้นหูกันเพียงระยะเวลาไม่ถึง 10 ปี เลย เนื่องจากการเปลี่ยนแปลงที่รวดเร็วของอินเทอร์เน็ต ทำให้ทิศทางการพัฒนาโปรแกรมและซอฟต์แวร์ได้เปลี่ยนแปลงไปจากเมื่อก่อนที่โปรแกรมหรือซอฟต์แวร์ต่างๆ ถูกพัฒนามาเพื่อใช้งานกับเครื่องคอมพิวเตอร์เท่านั้น แต่ปัจจุบันนี้ไม่เพียง บราวเซอร์บนเครื่อง พีซีเท่านั้นที่สามารถใช้เข้าถึงอินเทอร์เน็ตได้ แต่โปรแกรมและซอฟต์แวร์ทุกชนิดที่ทำงานบน พีซี เครื่อง PDA โทรศัพท์มือถือ และอุปกรณ์ต่างๆ ก็สามารถเชื่อมโยงกับอินเทอร์เน็ตได้

ด้วยเหตุผลนี้ ไมโครซอฟท์จึงเห็นว่าปัญหาหนึ่งที่กำลังเกิดขึ้นกับอินเทอร์เน็ตคือ เครื่องพีซีและอุปกรณ์ต่างๆ รวมทั้งเซิร์ฟเวอร์มากมายประกอบกันเป็นเครือข่ายหมีมานี้ ยังขาดวิธีแลกเปลี่ยนข้อมูลกันได้อย่างสมบูรณ์ ทำให้ไม่สามารถควบคุมเทคโนโลยีและข้อมูลรอบตัวได้ โดยข้อมูลที่สำคัญมักถูกเก็บไว้อย่างกระจัดกระจายในแอปพลิเคชัน และเว็บไซต์ต่างๆ ซึ่งยังขาดวิธีที่มีประสิทธิภาพในการนำข้อมูลเหล่านั้นมาใช้งานร่วมกันได้

จึงได้เกิดแนวคิดในการทำให้เว็บไซต์และโปรแกรมต่างๆ ที่ทำงานบนเครื่องพีซี และ อุปกรณ์ต่างๆ ที่อาจทำงานแยกจากกันในอดีตให้สามารถสื่อสารและแลกเปลี่ยนข้อมูลระหว่างกันได้ โดยจะสร้างซอฟต์แวร์ที่เป็นบริการ หรือ web services ไว้ในผลิตภัณฑ์ต่างๆ ของไมโครซอฟท์ ตั้งแต่ Windows ที่ใช้กับเครื่องพีซีไปจนถึง Windows ที่ใช้กับเครื่องเซิร์ฟเวอร์ และนำไปผนวกกับเว็บไซต์ต่างๆ ด้วย ซึ่งไมโครซอฟท์เล็งเห็นว่าในที่สุดผู้คนจะหันมาใช้ web service แทนแอปพลิเคชันต่างๆ แบบเดิมในเครื่องพีซี

สิ่งที่ทำแนวคิดนี้เป็นไปได้คือ XML ซึ่งเป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูล โดย Web service ต่างๆ จะใช้สื่อสารระหว่างกันได้ ไมโครซอฟท์ เชื่อว่า XML จะทำให้โลกของข้อมูลเข้าด้วยกันได้

แนวคิดนี้ จะทำให้เว็บกลายเป็นเหมือนระบบปฏิบัติการเสมือน ที่มีความสำคัญสำหรับองค์กรบริษัท เทคโนโลยี และผู้ใช้ทั่วไป เช่นเดียวกับที่ Windows มีความสำคัญกับเครื่องพีซี ซึ่งจะทำให้ Web service ทำงานได้อย่างมีประสิทธิภาพ เพราะสามารถดึงทรัพยากรของเครื่องมาใช้ได้ ในขณะที่เดียวกันเป็นการสร้างจุดแข็งของ ไมโครซอฟท์ในโลกอินเทอร์เน็ตไปในตัว เพราะใช้ฐานเป็น Windows นั่นเอง

## 1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาเทคโนโลยี ของ .NET โครงสร้างและการทำงานของ .NET ว่ามีลักษณะอย่างไร สามารถนำมาใช้ประโยชน์ได้อย่างไร
2. ศึกษาการเขียน โปรแกรมด้วย Visual Studio .net โดยใช้ Visaul Basic ,Visaul C++ และ Visaul C#
3. ศึกษาโครงสร้างการทำงานของ WebServices
4. เพื่อให้ นักศึกษาสามารถประยุกต์ความรู้ต่างๆ ที่ได้ศึกษามา ช่วยในการแก้ปัญหาและสร้างระบบงาน

## 1.3 ขอบเขตของโครงการ

โครงการนี้มุ่งเน้นไปที่การศึกษาเทคโนโลยี .NET ว่าคืออะไร มีประโยชน์อย่างไร และสามารถนำมาใช้ได้อย่างไร เพื่อให้สามารถประยุกต์การใช้งาน โดยใช้เทคโนโลยี .net ในการสร้าง Web Services ได้ โดยโครงการนี้ไม่เน้นการออกแบบ ฐานข้อมูล และรูปแบบการแสดงผล

## 1.4 วิธีการดำเนินงาน

1. ศึกษาทฤษฎีพื้นฐาน ว่า .net คืออะไร รูปแบบโครงสร้างของ .net
2. ศึกษารูปแบบขั้นตอนการติดตั้งและการใช้งาน Visual Studio .net
3. ศึกษาแนวคิดของ .net และการทำงานของ Web Services พร้อมการประยุกต์ใช้งาน
4. ศึกษาภาษาที่ใช้ในการเขียน เช่น ภาษา C# ที่เป็นภาษาใหม่ของ .net
5. ศึกษาวิธีใช้ MS SQL server 2000 เพื่อใช้ในการทำงานที่เกี่ยวกับฐานข้อมูล
6. นำความรู้มา ออกแบบ ระบบการทำงาน ของ Web Service

## บทที่ 2

### รู้จักกับ Microsoft.net และ Webservice

#### 2.1 แนวคิดการทำงานของ .net

ปัญหาของอินเทอร์เน็ตในปัจจุบันก็คือ เว็บไซต์ เครื่องพีซี เซิร์ฟเวอร์ และอุปกรณ์สื่อสารต่างๆยังขาดวิธีแลกเปลี่ยนข้อมูลกันได้อย่างสมบูรณ์ ทำให้ข้อมูลของผู้ใช้ถูกเก็บอย่างซ้ำซ้อนและกระจัดกระจาย เมื่อ .net เข้ามาก็จะทำให้เว็บไซต์และอุปกรณ์เหล่านี้สามารถสื่อสารแลกเปลี่ยนข้อมูลระหว่างกันได้ผ่านซอฟต์แวร์ที่ให้บริการเว็บ ซึ่งเรียกว่า Web service

เราสามารถสรุปได้ว่า .net ก็คือ แพลตฟอร์ม ( Platform ) ของไมโครซอฟท์สำหรับการบริการเว็บแบบ Web services (การให้บริการแอปพลิเคชัน บนเว็บที่ใช้ ภาษา XML เป็นมาตรฐาน ) เป็นซอฟต์แวร์รุ่นต่อไปที่ใช้สำหรับการติดต่อในโลกของข้อมูล และอุปกรณ์สื่อสารรูปแบบต่างๆ เข้าด้วยกัน

นั่นคือ .net เป็นนิยามของการให้บริการซอฟต์แวร์ในรูปแบบของ Services ซึ่งจะรันโดยไมโครซอฟท์อยู่บนอุปกรณ์เชื่อมต่อ หรือระบบปฏิบัติการใดๆ เลย ตามที่ไมโครซอฟท์กล่าวว่า Anytime Anywhere AnyDevice หรือ สามารถ ใช้บริการได้ ทุกที่ ทุกเวลา และทุกอุปกรณ์ เราสามารถใช้อุปกรณ์ไม่ว่าจะเป็น Mobile Phone ,PDA หรือเครื่องคอมพิวเตอร์พีซีทั่วไป ในการติดต่อสื่อสาร ทำงานกับ Service ผ่านอินเทอร์เน็ตได้เหมือนกันหมด ซึ่งสิ่งที่ทำให้สำเร็จได้นั้น ก็มาจากมาตรฐานใหม่อย่าง XML (Extensible Markup Language)และ SOAP (Simple Object Access Protocol) มาตรฐานการเชื่อมต่อข้อมูลระหว่างแพลตฟอร์มได้

#### 2.2 เป้าหมายของ Microsoft .net

เราสามารถสรุปปัจจัยต่างๆของ Microsoft.net ซึ่งเป็นเป้าหมายของเทคโนโลยีหรือแนวทางที่ถูกใช้ในอนาคตดังนี้

##### 2.2.1 Web services

ส่งผลให้การทำงานเป็นไปอย่างรวดเร็วขึ้น โดยทุกอย่างจะทำงานบนพื้นฐานของเว็บ ซึ่งจะเกี่ยวกับทั้งเรื่องของซอฟต์แวร์และทรัพยากร ในระบบเครือข่าย เช่น ข้อมูล เป็นต้น

##### 2.2.2 การทำงานร่วมกัน

หลังจากสร้าง Web services ขึ้นแล้วเว็บที่ถูกสร้างขึ้นนั้นจะถูกรวบรวมและสามารถใช้งานร่วมกันได้อย่างสะดวกสบาย

##### 2.2.3 ความเรียบง่ายและนำใช้งาน

ผู้ใช้ที่เข้ามาใช้งานไม่ว่าจากอุปกรณ์ใดก็ตาม สามารถใช้งานได้ง่ายและมีความดึงดูด นำใช้งาน

### 2.3 มาตรฐานที่เกี่ยวข้องกับ Microsoft.net

.net เป็นการกำหนด มาตรฐานของเทคโนโลยีที่สามารถใช้ซอฟต์แวร์ทำงานและแลกเปลี่ยนข้อมูลร่วมกันได้ ทำให้เกิด ผลลัพธ์ที่สามารถทำงานร่วมกันได้ระหว่างแพลตฟอร์มต่างกัน เพื่อเป็นประโยชน์ต่อผู้ใช้ มาตรฐานเทคโนโลยีหลักที่เกี่ยวข้องกับ .net คือ XML , SOAP , UDDI เราสามารถอธิบายเบื้องต้นได้ดังนี้

- **XML (Extensible Markup Language)**

XML คือรูปแบบทั่วไปสำหรับข้อมูลที่มีโครงสร้างบนเว็บเป็นชุดคำสั่งในการจัดรูปแบบข้อความของข้อมูล เพื่อการสร้าง แฟ้มที่สะดวกในการสร้างและอ่าน เข้าใจง่าย และไม่ขึ้นกับแพลตฟอร์มใดๆ ในปัจจุบันถ้าแอปพลิเคชันบนเว็บต้องการเรียกใช้ข้อมูลจากที่อื่น แอปพลิเคชันจะต้องทำ screen-scraping ซึ่งเป็นการจับภาพของหน้าจอที่มีข้อมูลซึ่งแอปพลิเคชันต้องการ และพยายามค้นหาวิธีการแปลงรูปภาพกลับไปเป็นข้อมูลจะมีแนวโน้มความผิดพลาดและไม่เกิดประสิทธิภาพ แต่ XMLจะมีรูปแบบสำหรับ XML Web services เพื่อแลกเปลี่ยนข้อมูลจริงๆ ไม่ใช่แค่เพียงรูปภาพเท่านั้น

- **SOAP (Simple Object Access Protocol)**

แต่แนวคิดของ web service ได้แก้ไขข้อจำกัดเหล่านี้ลงได้ โดยการเสนอ protocol ใหม่ที่ชื่อว่า SOAP (Simple Object Access Protocol) ซึ่งเป็น protocol ที่ใช้เรียก component ต่าง ๆ ใน platform ใด ๆ ก็ได้ เนื่องจากใช้ XML เป็นรูปแบบของข้อมูลที่ส่ง โดย protocol นี้จะทำงานอยู่บน HTTP ซึ่งเป็น protocol มาตรฐานสำหรับ web อยู่แล้ว ซึ่งนั่นหมายความว่า การเรียกใช้ component โดยอาศัย SOAP ก็คือการเรียกผ่าน web นั่นเอง

อย่างไรก็ดี SOAP ก็ยังมีข้อเสียอยู่บ้าง ได้แก่

1. เนื่องจาก SOAP message นั้นเก็บอยู่ในรูปแบบ XML ทำให้เสียเวลาในการแปลเอกสาร XML กลับมาเป็นรูปแบบที่โปรแกรมเข้าใจ
2. SOAP ทำงานอยู่กับ HTTP protocol ซึ่งมีความเร็วในการรับ และส่งข้อมูล ค่อนข้างต่ำ ด้วยเหตุนี้ทำให้ความเร็วของ SOAP อยู่ในระดับเดียวกับ HTTP ในขณะที่ protocol อื่นเช่น FTP มีความเร็วสูงกว่ามาก

- **UDDI (Universal Description, Discovery and Integration)**

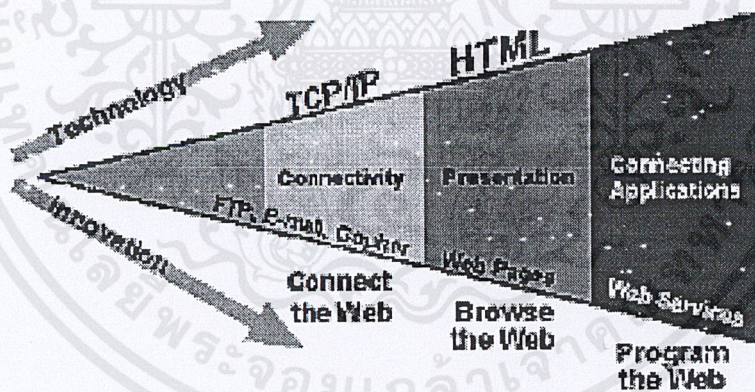
UDDI เป็นมาตรฐานที่จัดตั้งขึ้น โดยบริษัท IBM, Microsoft และบริษัทยักษ์ใหญ่ทางธุรกิจ B2B (Business-to-Business) อื่น ๆ UDDI ถูกสร้างขึ้นมาเป็นมาตรฐาน ในการค้นหาบริการ web service สำหรับคู่ค้าทางธุรกิจ ซึ่งเปรียบได้กับฐานข้อมูล ขนาดใหญ่ ซึ่งมีข้อมูลของ web service ที่เปิดให้บริการ

- **WSDL (Web Services Description Language)**

WSDL เกิดจากความร่วมมือระหว่าง IBM และ Microsoft WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะของ web service และวิธีการติดต่อกับ web service นั้น ๆ โดยใช้ไวยากรณ์ของภาษา XML ซึ่ง WSDL อยู่ในความดูแลของ W3C

#### 2.4 พัฒนาการก่อนมาเป็น Web services

อินเทอร์เน็ต เกิดจากวงการทหารของอเมริกาเพื่อที่จะเชื่อมโยงคอมพิวเตอร์พีซีต่างๆเข้าด้วยกัน การใช้งานอินเทอร์เน็ตในยุคแรกๆ เป็นการรับส่ง ไฟล์ หรือ อีเมลล์ตามปกติเท่านั้น

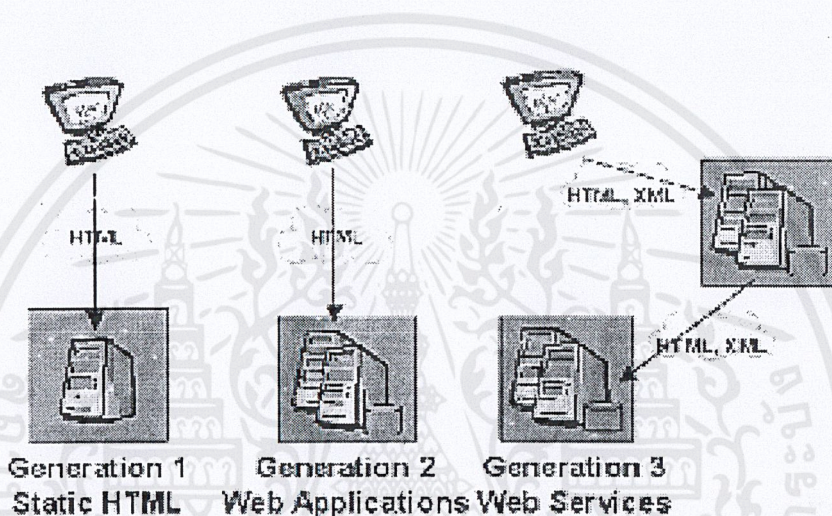


รูปที่ 2-1 พัฒนาการก่อนมาเป็น Web services

โปรโตคอล (Protocol) ซึ่งทำงานเป็นโครงสร้าง ในการติดต่อระหว่างเครื่องคอมพิวเตอร์ของเครือข่ายอินเทอร์เน็ต และยังทำงานอยู่ในปัจจุบันคือ TCP/IP จนกระทั่งถึงยุคต่อมาซึ่งเป็นการพลิกโฉมจากอินเทอร์เน็ตที่อยู่ในวงการแคบๆ หรือวงการทหาร ได้ขยายออกสู่โลกภายนอก จนกระทั่งถึงยุคในปัจจุบันนี้ (ถือเป็นยุคที่ 2 ของ อินเทอร์เน็ต ) ก็คือการคิดค้นเว็บเพจ หรือการแชร์เอกสารร่วมกัน (World Wide Web) ขึ้นมา ซึ่งเน้นเรื่องการสร้างเว็บเพจ มีภาษา html ในการสร้างเว็บแอปพลิเคชันต่างๆ บนเว็บไซต์ จากเว็บไซต์ก็เน้นในการนำเสนอ (Presentstion) หรือคุณลักษณะต่างๆที่เว็บเสนอต่อผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนนี้การพัฒนาเว็บแอปพลิเคชันก็ก้าวมาถึงจุดๆหนึ่งแล้ว เราสามารถทำให้เกิดทุกอย่างในเว็บแอปพลิเคชันได้ แต่การก้าวไปสู่ยุคที่ 3 นั้น คือตอนนี้เรามีเว็บแอปพลิเคชันอยู่ทั่วไป แทบทุกองค์กรมีเว็บไซต์เป็นของตนเอง แต่ถ้าการที่เราจะทำงานร่วมกันเหมือนในยุคที่หนึ่ง ที่ Windows ต้องการใช้งานแอปพลิเคชัน ร่วมกันก็มีเทคโนโลยี COM พัฒนาขึ้นมา เช่นเดียวกับเว็บแอปพลิเคชัน ถ้าต้องการทำงานร่วมกัน ต้องการดึงข้อมูลจากอีกเว็บไซต์หนึ่งมาแสดงผลในเว็บไซต์ของเราจะทำอย่างไร คำตอบของปัญหานี้คือ XML และ Web services นั่นเอง



ถ้าเราดูในเรื่องของเว็บแอปพลิเคชันเองก็จะแบ่งเป็น 2 ยุค เช่นเดียวกัน ในยุคแรกๆ เว็บแอปพลิเคชันจะมีการแสดงผลแบบ one- to- one ก็คือเป็นเว็บเพจแบบ static page นั่นเอง การแสดงผลก็ใช้ HTML ธรรมดา จนกระทั่งถึงยุคที่ 2 ซึ่งมีกาแสดงผลข้อมูลแบบ dynamic เว็บไซต์จะมีการดึงข้อมูลมาแสดงผลเป็นข้อมูลแบบ real time

การเรียกข้อมูลจากฐานข้อมูล (database) ที่อยู่ภายในองค์กร ไม่ใช่ปัญหาอีกต่อไปสำหรับเว็บแอปพลิเคชันในปัจจุบัน แต่ปัญหามักจะเกิดขึ้นถ้าเราต้องการจะไปดึงข้อมูลจากเว็บไซต์อื่นๆ

ถ้าเราต้องการไปดึงข้อมูล ปัจจุบัน (Real time information) จะทำอย่างไร ถ้าในปัจจุบันนี้มีอยู่ 2 วิธีคือ

1. สร้างลิงค์ไปยังเว็บไซต์นั้น โดยตรง คือ สร้าง Hyperlink เพื่อลิงค์ไปที่เว็บไซต์นั้นแล้วผู้ใช้ไปค้นหาเอง
2. ทำการตกลงซื้อข้อมูล โดยให้ทำการส่งข้อมูลมาเฉพาะ จากการตกลงกันระหว่างทั้ง 2 ฝ่าย แต่ถ้ามีการดึงที่อื่น ๆ มาอีกเราก็ต้องการรูปแบบการส่งข้อมูลแบบนี้ อีก วิธีนี้เป็นวิธีที่แก้ปัญหาได้บ้างแต่ไม่เป็นมาตรฐาน สำหรับการแก้ปัญหา นี้ก็คือ Web service กับ XML นั่นเอง

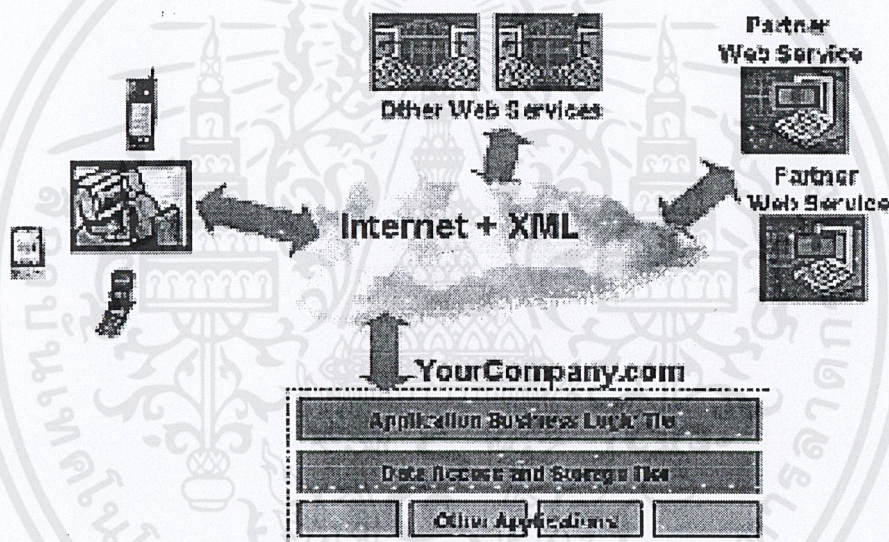
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 ความหมายของ Web Services

Web Services คือ การที่เราสร้างฟังก์ชันตัวหนึ่งฝังไว้ในตัวเว็บแอปพลิเคชันและเว็บเซิร์ฟเวอร์ของเรา นั่นเอง เพื่อให้จะให้ไคลเอนต์ (เครื่องที่ใช้บริการจากเว็บ) หรือ เว็บไซต์อื่นๆ สามารถเรียกใช้มันได้ ซึ่งหลักการของ Web service ก็คือจุดนี้นั่นเอง

### 2.5.1 การทำงานของ Web Services

การทำงานของ Web Services นั้น จะทำงานอยู่บน Internet Protocol ทั้งหมด คือว่าทำงานอยู่บน HTML, TCP/IP โดยใช้ภาษา XML เป็นภาษาที่ทำการเข้ารหัส (encode) และถอดรหัส (decode) ข้อมูลที่ส่งผ่านกันระหว่างไคลเอนต์กับเซิร์ฟเวอร์นั่นเอง การร้องขอ (request) ที่ส่งไปจากเว็บไซต์ของเราไปยังเว็บไซต์อื่นก็ จะถูกแปลงเป็น XML แล้วได้ข้อมูลเป็น Real time เวลาตอบกลับก็เป็น XML เช่นเดียวกัน



รูปที่ 2-3 รูปแบบการทำงานของ Web Services

ถ้าจะพูดในแง่ของการพัฒนา Web services ก็คือ Remote Procedure Call อย่างหนึ่ง คือการเรียกใช้ฟังก์ชันที่ไม่ได้อยู่ในเครื่องเรา แต่ว่า Remote Procedure Call อันนี้ใช้โปรโตคอลของอินเทอร์เน็ตในปัจจุบัน เพราะ ฉะนั้นจึงเหมาะกับงานที่เกี่ยวกับอินเทอร์เน็ตมากที่สุด

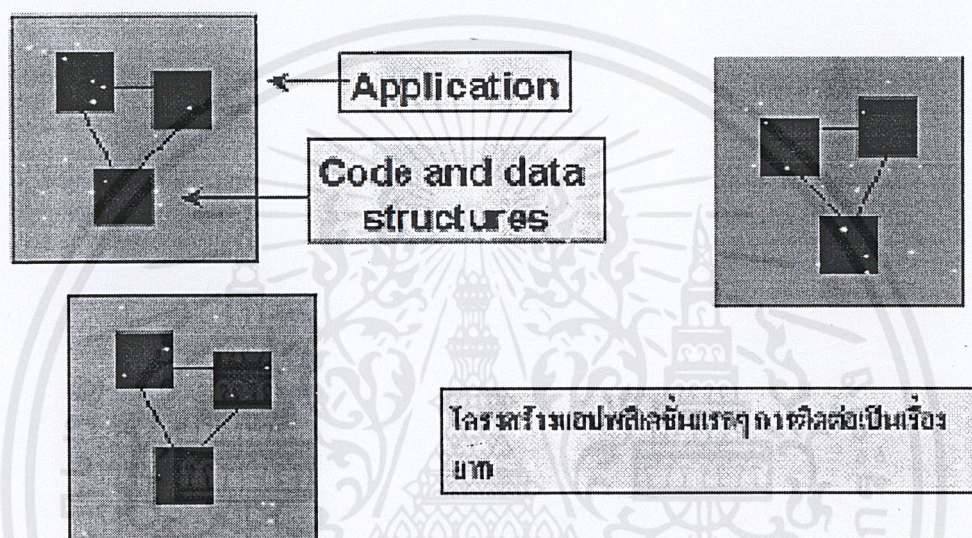
นอกจากแอปพลิเคชันจะสามารถใช้ข้อมูลร่วมกันได้แล้ว Web services ยังเป็นวิธีที่แอปพลิเคชันจะสามารถโต้ตอบกับแอปพลิเคชันอื่นได้โดยตรง ไม่ว่าจะเป็น แอปพลิเคชันที่อยู่บนเซิร์ฟเวอร์เดียวกันหรือระบบที่ห่างไกลออกไป โดยไม่คำนึงถึงระบบปฏิบัติการหรือภาษาที่ใช้ในการเขียนโปรแกรม

### บทที่ 3

#### สถาปัตยกรรมของ .net

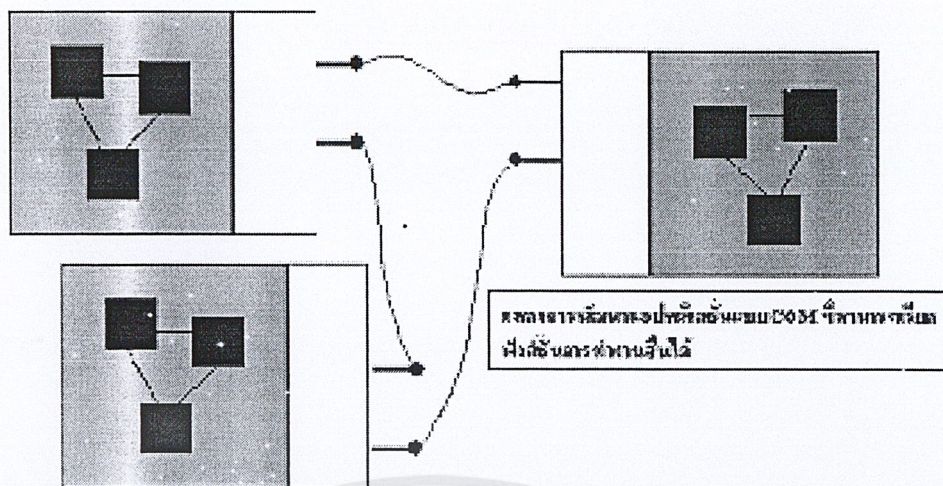
##### 3.1 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม

ก่อนที่จะมีการพัฒนาโปรแกรมเป็น Object Oriented นั้น แอปพลิเคชันแต่ละตัวเปรียบเสมือนกล่องภายในแอปพลิเคชัน ก็มีโค้ด (code) มีโครงสร้างข้อมูล (data structure) ต่างๆ ของตัวเองมีฟังก์ชันต่างๆ ของตัวแอปพลิเคชันนั้นๆ



รูปที่ 3-1 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม แบบแอปพลิเคชันต่างๆ

การที่แอปพลิเคชันต่างๆ จะมีการเรียกใช้หารทำงานของกันและกัน หรือมีการส่งผ่านข้อมูลถึงกันและกัน เป็นสิ่งที่ทำได้ยาก ซึ่งอาจต้องมีการกำหนดอะไรขึ้นมาเองระหว่าง 2 แอปพลิเคชัน นั้นๆ จนกระทั่งในยุคถัดมา ไมโครซอฟท์ได้คิดค้นเทคโนโลยี COM (Component Object Model) เป็นวิธีที่ทำให้เราเขียนโปรแกรมเป็นแบบ Object Oriented และเรียกใช้การทำงานที่มาจากต่างแอปพลิเคชันได้



รูปที่ 3-2 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม แบบ Component Object Model

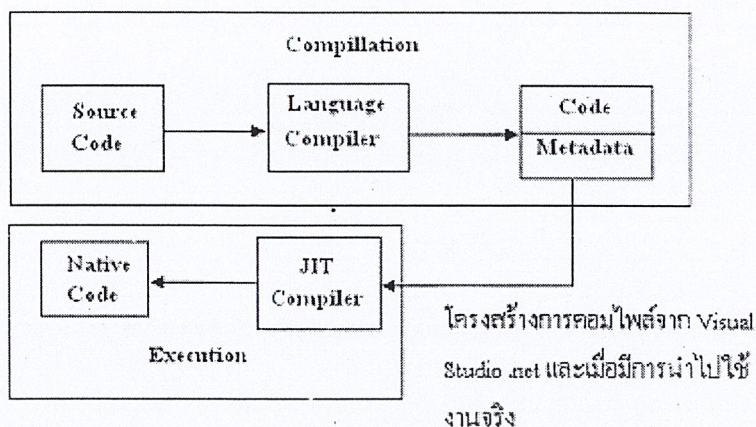
หากเราจะอธิบายให้ง่ายขึ้น ก็คือ เปรียบเทียบเหมือนเราเอาแพ็คเกจอันหนึ่งต่อแอปพลิเคชันของเราไว้และการพูดคุยกันของแอปพลิเคชันก็พูดคุยผ่านแพ็คเกจที่เราต่อเอาไว้ จะกระทั่งมาถึงตัว Visual Studio.net ที่ได้รับการออกแบบใหม่ จะเห็นว่าจากรูปเดิม กล้องแพ็คเกจหายไป ถ้าเราพัฒนาด้วยรูปแบบเทคโนโลยี .net นั้น คลาสต่างๆ สามารถติดต่อกันได้โดยตรง



รูปที่ 3-3 พัฒนาการของเทคโนโลยีการเขียนโปรแกรม แบบ .NET

การพัฒนาแอปพลิเคชันด้วย Visual Studio .net นั้น เมื่อเราคอมไพล์สิ่งที่เราจะได้ จะไม่ใช่โค้ดไบนารี (Binary code) เลขที่เดียว แต่จะได้เป็นภาษากลางอันหนึ่งเรียกว่า Microsoft Intermediate Language (MSIL) ซึ่งเป็นภาษาในระดับเลเยอร์ต่างๆ โครงสร้างของภาษา (Syntax) จะเหมือนภาษา Assembly ภายในสิ่งที่เกิดจากการคอมไพล์ก็จะเป็น MSIL ตัวนั้น ภายในตัวมันจะประกอบด้วย 2 ส่วน คือ โค้ด กับตัวแอตทริบิวต์หรือ พร็อพเพอร์ตี้ต่างๆ ที่ใช้อธิบายตัวโค้ดนั้นซึ่งเรียกว่า Meta data

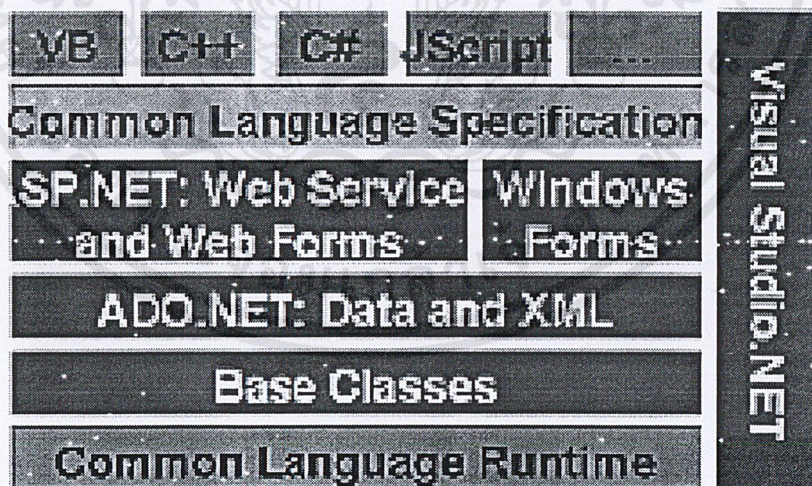
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 โครงสร้างการคอมไพล์จาก Visual Studio.net

จากนั้น เมื่อโค้ดที่เป็น Intermediate Language จะถูกเรียกใช้งานจริงๆ จะมีตัว Compiler (ตัวแปลภาษา) ตัวหนึ่งมาทำการคอมไพล์โค้ดตัวนั้นให้เป็นโค้ดไบนารี ซึ่ง Compiler ตัวนั้นจะเรียกว่า Just In Time Compiler (JIT Compiler) ที่เรียกว่า Just In Time เพราะว่าจะมีการคอมไพล์เมื่อมีการใช้งาน ฉะนั้นคลาสหรือโค้ดต่างๆ ที่เราพัฒนาขึ้นแล้วจะถูกคอมไพล์มาเป็น Intermediate Language ที่มีโครงสร้างของภาษาแบบเดียวกัน เพราะฉะนั้นคลาสต่างๆ ในแอปพลิเคชันจึงสามารถทำงานได้อย่างกลมกลืนกันแต่ไม่มีข้อติดขัดอะไร

### 3.2 สถาปัตยกรรมของ .net



รูปที่ 3-5 โครงสร้างสถาปัตยกรรมของ .net

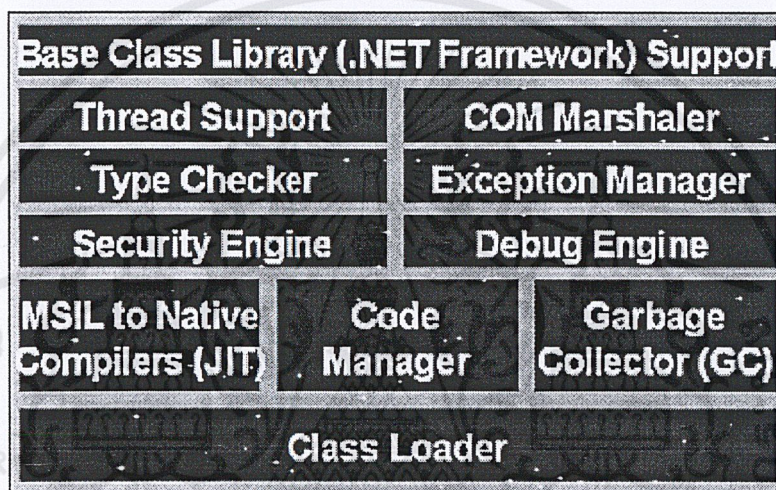
จากรูปเป็นแสดงถึงสถาปัตยกรรมของแอปพลิเคชัน .net ที่พัฒนาด้วย visual studio.net ถือว่าเป็นการเปลี่ยนแปลงครั้งสำคัญ ทั้งแพลตฟอร์มที่ใช้ในการพัฒนาแอปพลิเคชัน และสถาปัตยกรรมที่ใช้ โดยมีเลเยอร์ล่างสุดคือ .net framework SDK เปรียบเสมือน Runtime Library ที่จะรันอยู่คอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนับสนุนการทำงานของแอปพลิเคชัน จากนั้นจะเป็นเลเยอร์ของ Common Language Runtime เป็นผลลัพธ์ของการคอมไพล์ แอปพลิเคชัน .net เลเยอร์ถัดขึ้นมาเป็นเครื่องมือ (Tools) และเทคนิคต่างๆที่เราสามารถใช้พัฒนาแอปพลิเคชันได้ทั้งในเรื่องของ Web Services ,ADO.net ,ASP.net จนกระทั่งถึงเลเยอร์บนสุดคือภาษาที่ใช้ในการพัฒนาแอปพลิเคชันด้วย Visual Studio.net

### 3.2.1 เลเยอร์ Common Language Runtime

ภายในตัว Common language Runtime จะมีโมดูล (Module) ย่อยๆ ซึ่งเป็นสถาปัตยกรรมภายใน ดังรูป



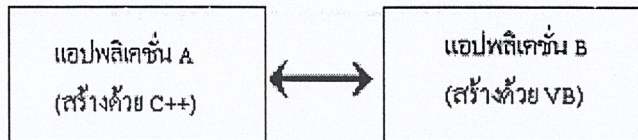
รูปที่ 3-6 โครงสร้าง เลเยอร์ Common Language Runtime

ด้านล่างสุดจะมี Class Loader ซึ่งเอาไว้โหลดโปรแกรมของเราขึ้นมาทำงาน นอกจากนั้นก็จะมี Compiler ซึ่งจะทำการคอมไพล์ภาษา Intermediate Language ให้เป็นภาษาไบনারีโดยจะมี Code manager และ Garbage Collector คอยจัดการกับหน่วยความจำที่เราจองเวลาเรียกใช้ง่าย นอกจากนี้ก็จะมีเรื่องราวระบบความปลอดภัย (Security) ในการทำงาน รวมทั้ง Debug Engine ในการดัก Runtime Error และตัว Exception Manager การตรวจเช็คชนิดของตัวแปรต่างๆ และด้านบนสุดจะเป็นการใช้งานระหว่าง Library Class ต่างๆ ซึ่งจัดเตรียมมาให้ เพราะฉะนั้นเลเยอร์ของ Common Language Runtime การคอมไพล์แอปพลิเคชันใดก็ตาม ไม่ว่าจะเป็น ASP.net หรือเขียนแอปพลิเคชันบน Windows ธรรมดา หรือจะเป็นกรเขียน Web Service ก็ตาม สิ่งที่ได้จากการคอมไพล์จะเป็น Common Language Runtime ตามแผนผังนี้

นั่นคือจากการออกแบบเพื่อสนับสนุนการทำงานร่วมกันใน Common Language Runtime จึงขัดข้อเสียดังของ COM ไปได้ เนื่องจากของ COM คือเป็นเพียงการเอาอะไรบางอย่างมาห่อนั้นๆก็จะทำให้เกิดปัญหาเรื่องการเข้ากันได้ (Compatibility) ระหว่างเวอร์ชันเดิมกับเวอร์ชันปัจจุบันแต่เราพัฒนาด้วย Visual Studio .net ข้อเสียดังของ COM ทั้งหมดก็จะถูกขจัดเสีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจริง Common Language Runtime เป็นหลักการงานที่มีวิวัฒนาการมาจาก COM อีกที่หนึ่ง เป็นการ Object Oriented ที่แกนของภาษาเลย โดย Visual Studio .net นั้นถูกออกแบบเพื่อสนับสนุน Object Oriented โดยเฉพาะ คลาสต่างๆที่อยู่ในแต่ละแอปพลิเคชัน A เราอาจจะเขียนด้วยภาษา C++ และแอปพลิเคชัน B ซึ่งเขียนด้วยภาษา VB ได้ คือการ Inherriant ข้ามภาษานั้นสามารถทำได้

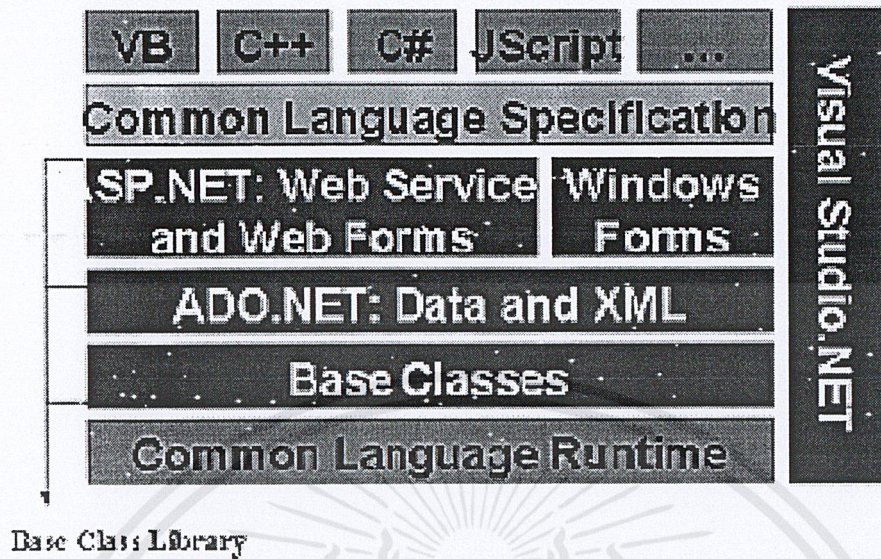


รูปที่ 3-7 การออกแบบเพื่อสนับสนุนการทำงานร่วมกันใน *Common Language Runtime*

ใน Visual Studio .net จะคอมไพล์เป็นภาษาเดียวกันคือ Intermediate Language ตามรูปก่อนหน้าก็คือคอมไพล์เป็นภาษาอันหนึ่ง เพราะฉะนั้นจึงสามารถ Inherit กันข้ามภาษาได้

นอกจากนี้ ยังสามารถทำงานด้วยกันกับ COM แบบเดิมที่เราเคยเขียนมาแล้วได้ด้วย ซึ่งใช้ Visual Studio เดิม โค้ดเหล่านี้ก็ไม่จำเป็นต้องโยนทิ้งใน Visual Studio .net เราสามารถเรียกใช้งาน COM ได้ตามปกติ และในทางกลับกัน ใน Visual Studio 6.0 ที่มีอยู่นี้สามารถเรียกใช้งานคอมโพเนนต์ที่เขียนด้วย Visual Studio .net ได้เช่นกัน คือเป็นการเข้ากันได้ทั้งสองทาง (Backward-Forward Compatibility) นี่คือข้อดีมากๆ ของ .net ทำให้เราไม่ต้องมาพัฒนาโค้ดใหม่

### 3.2.2 เลเยอร์ Base Class Library

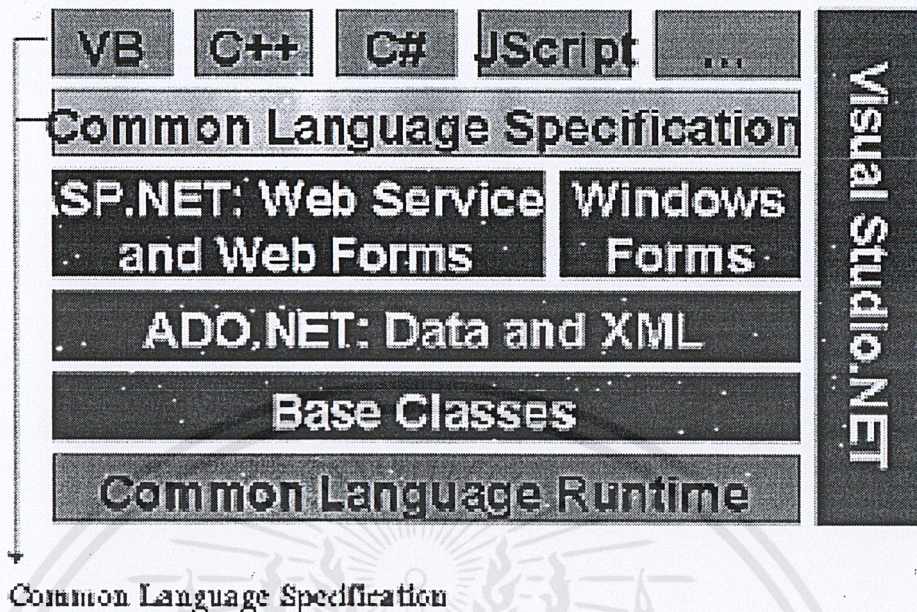


รูปที่ 3-8 โครงสร้าง เลเยอร์ Base Class Library

ตัว Base Class Library ก็คือ การที่เรารวบรวมฟังก์ชัน API (Application programming Interface) ซึ่งกระจัดกระจายอยู่ เวลาจะใช้เราต้องไปค้นหาใน Help นั่นคือ Base Class Library พยายามที่จะรวบรวม API และฟังก์ชันบางทั้งหมดเกี่ยวกับระบบเข้ามาไว้ในลักษณะของ Object Oriented ทั้งหมด โดยมีคลาสอันหนึ่งเป็นมาตรฐานเป็นคลาสที่สร้างมาในตัวระบบเรียบร้อยแล้ว ซึ่งคลาสทั้งหมดจะอยู่ภายใต้คลาสหลักอันหนึ่งที่เราเรียกว่า System

ภายในคลาสจะมีคลาสย่อยๆ มากมาย ซึ่งแต่ละอันจะสนับสนุนการทำงานที่เราต้องการได้ ไม่ว่าจะเป็นเรื่องของการทำกราฟฟิก การทำเกี่ยวกับโครงสร้างข้อมูล (Data Structure) การทำเกี่ยวกับเรื่องเครือข่าย (Network) ฟังก์ชัน API เหล่านี้จะถูกจัดกลุ่มให้เป็น Object Oriented อยู่ใน System Class การเรียกใช้งาน System Class จะสามารถเรียกได้ VB และ C++

### 3.2.3 เดเยอร์ Common Language Specification



รูปที่ 3-9 โครงสร้าง เดเยอร์ Common Language Specification

คือเครื่องมือที่ใช้ในการสร้างแอปพลิเคชัน หรือหลักการที่ใช้ในการเขียนโปรแกรมต่างๆ เช่น เรื่องของ ADO.net ,ASP.net ที่ใช้ในการพัฒนาแอปพลิเคชัน แต่สิ่งที่เหนือว่าทุกอย่างคือ ภาษาที่เราใช้งาน ภาษาต่างๆ ที่ทำงานใน .net นั้นมีข้อดีคือ ต้องสนับสนุนมาตรฐานเดียวกัน เรียกว่า Common Language Specification

ในไม่ช้าเราจะเห็นเว็บเพจเขียนด้วย COBAL ก็ได้ รวมทั้งภาษาอื่นด้วย นอกจากนี้ใน ตระกูล .net เองเราก็มี VB,VC และ C# และภาษาอื่นๆ เช่น PASCAL,Perl โดยภาษาที่ใช้งานประเภท Object ทั้งหมดสามารถทำเป็น แพลตฟอร์มของ .net ได้ เพราะว่าใน .net นั้นผลิตทุกอย่างเป็น Object Oriented ทั้งหมด

## บทที่ 4

### XML

ในปี ค.ศ. 1991 Tim Bernes-Lee ได้พัฒนา ภาษา HTML ขึ้นเพื่อใช้ในการสร้างเว็บเพจ หลังจากนั้น HTML ก็แพร่หลายและใช้กันอย่างกว้างขวาง จนทำให้มีเว็บเพจจำนวนมากมายมหาศาล HTML เป็นมาตรฐานที่ได้รับการปรับปรุงและเสริมต่อมาเป็นลำดับ แต่ HTML ก็ยังเป็นมาตรฐานที่เน้นการนำข้อมูลข่าวสารมาแสดงผลบนจอภาพ ดังนั้น โครงสร้างหลักจึงได้แก่ หัวเรื่อง ชื่อเรื่อง ภาพ ชุดตัวพิมพ์ การแสดงผลด้วยรูปภาพ เสียง และมัลติมีเดีย เพียงเท่านี้ก็ทำให้นักพัฒนาเว็บสามารถ ออกแบบเว็บเพจที่ได้ด้อย่างน่ามหัศจรรย์ และมีประสิทธิภาพยิ่ง

#### 4.1 ข้อด้อยของ HTML

ด้วยข้อจำกัดและความต้องการของผู้ใช้ยังมีอีกมาก ทำให้ HTML มีข้อด้อยบางประการที่จะต้องปรับปรุง ดังต่อไปนี้

- แท็กของ HTML มีจำกัด และไม่อนุญาตให้เราสร้างแท็กขึ้นเองได้ ทำให้การทำงานมีขีดจำกัด แม้ว่าจะมี HTML เวอร์ชันใหม่ๆ ออกมา แต่ก็ยังไม่สามารถตอบสนองกับความต้องการของผู้ใช้ได้ทั้งหมด เพราะรูปแบบแท็กของ HTML มีลักษณะตายตัวว่าแต่ละแท็กเอาไว้ทำอะไร โดยที่แท็กต่างๆ จะถูกกำหนดโดยองค์กร W3C (World Wide Web Consortium) ซึ่งเป็นองค์กรที่มีหน้าที่วางมาตรฐานการทำเอกสารต่างๆบนเว็บ และนอกจากนี้ผู้ผลิต WebBrowser ค่ายใหญ่ๆ เช่น Internet Explorer ของบริษัทไมโครซอฟต์ หรือ Netscape ของบริษัทเน็ตสเคป คอมมิวนิตี้นั้นก็พยายามดึงดูดผู้ใช้เบราว์เซอร์ให้มาใช้ผลิตภัณฑ์ของตนมากขึ้น โดยการเพิ่มเติมแท็กจากแท็กมาตรฐานที่มีอยู่ ทำให้แต่ละเบราว์เซอร์แสดงผลลัพธ์จากไฟล์ HTML เดียวกัน แตกต่างกันไป
- เอกสารที่เขียนด้วย HTML มุ่งเน้นเฉพาะการแสดงผลเป็นหลัก ส่วนการสื่อความหมายของเอกสารนั้นนับว่าน้อยมาก ทำให้การค้นหาข้อมูลที่ต้องการจากเครือข่ายอินเทอร์เน็ต เป็นไปอย่างยากลำบาก คือ ได้ผลลัพธ์ที่ไม่เกี่ยวข้องออกมาเป็นจำนวนมาก
- HTML สามารถให้มุมมองเพียงด้านเดียวแก่ผู้เข้ามาเยี่ยมชม คือเราไม่สามารถทำให้ผู้ใช้หลายๆ คนมองเห็นเพจเดียวกันแตกต่างกันได้ ดังนั้นจึงต้องใช้ ASP และ DHTML (Dynamic HTML) เพื่อให้มองเห็นได้หลายมุมมอง ซึ่งในงานหลายอย่างต้องการคุณสมบัติดังกล่าว โดยเฉพาะอย่างยิ่งการทำธุรกิจบนอินเทอร์เน็ตที่จำเป็นต้องมีการจัดแบ่งกลุ่มลูกค้า เช่น จัดกลุ่มลูกค้าให้แต่ละคน ได้มองเฉพาะในสินค้าส่วนที่เขาสนใจ
- HTML ไม่สนับสนุนหลัก “การนำกลับมาใช้ใหม่ (reuse)” การแก้ไขเอกสารแต่ละครั้งค่อนข้างลำบาก จึงไม่สะดวกกับงานที่มีการเปลี่ยนแปลงข้อมูลบ่อย
- การเปลี่ยนกลับไปมาระหว่าง HTML กับงานอื่นทำได้ยาก เช่น เมื่อเราสร้าง HTML แล้วจะนำกลับมาใช้งานใหม่ได้ยาก เช่น มีเอกสารเวิร์ดและแปลงเป็น HTML แต่หากจะแปลงกลับทำได้ยากที่จะทำได้เหมือนเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี พ.ศ. 2539 Jon Bosak ได้เสนอแนวคิดต่อ W3C (World Wide Web Consortium) และนำเสนอภาษาใหม่ โดยนำแนวทางบางอย่างมาจาก SGML Standard Generalize Markup Language ซึ่งได้รับการพัฒนามานานแล้ว แต่มีความซับซ้อน ทำให้ใช้งานยาก ข้อเสนอใหม่นี้เรียกว่า XML (Extensible Markup Language) โดยเน้นให้มีส่วนช่วยแสดงส่วนที่เป็นเนื้อหาหรือสาระของข้อมูล

#### 4.2 เป้าหมายหลักของ XML

เป้าหมายที่สำคัญของ XML เน้นที่จะนำไปใช้งานในอินเทอร์เน็ต จึงมีเป้าหมายหลักดังนี้

- เพื่อเป็นมาตรฐานในการทำเอกสารบนเครือข่ายอินเทอร์เน็ตที่ผู้ใช้ทุกคนต่างมีความเข้าใจ และมีข้อตกลงที่ตรงกัน นั่นคือเอกสาร XML จะต้องเป็นรูปแบบที่สามารถใช้ได้ทั่วไปบนอินเทอร์เน็ต
- เพื่อรองรับการแลกเปลี่ยนข้อมูลระหว่างแอปพลิเคชัน โดยไม่ต้องขึ้นกับแพลตฟอร์มหรือผู้ผลิตรายใดรายหนึ่ง
- มาตรฐานของ XML จะต้องเข้ากันได้กับมาตรฐานของ SGML ซึ่ง XML นั้นนับเป็นส่วนหนึ่งของ SGML ทำให้องค์กรกำลังใช้ SGML อยู่แล้วก็สามารถนำ XML มาใช้ร่วมด้วยกันได้
- เอกสาร XML จะต้องมีรูปแบบที่ผู้พัฒนาสามารถเขียน โปรแกรมเพื่อนำเอาเอกสารดังกล่าวไปประมวลผลได้โดยง่าย
- การออกแบบเอกสาร XML ควรทำได้ด้วยความรวดเร็ว
- เป็นการแยกส่วนของข้อมูลออกจากการแสดงผลอย่างชัดเจน คือ XML จะไม่บอกเราว่าการแสดงผลที่ได้จะเป็นเช่นไร แต่จะบอกว่าตัวเนื้อข้อมูลจริงๆ นั่นคืออะไร ทำให้การเปลี่ยนแปลงข้อมูลจะไม่ส่งผลกระทบต่อ การแสดงผลเลย ในการกลับกันการแก้ไขส่วนการแสดงผลก็จะมีผลกระทบต่อเนื้อข้อมูลจริงๆ เช่นกัน นอกจากนี้การนำข้อมูลไปแสดงผลนั้น เราก็จะสามารถเลือกได้ว่าจะใช้อะไรเป็นตัวดึงข้อมูลมาแสดงผล เช่น ใช้ CSS, XSL, VBScript, JavaScript เป็นต้น
- เอกสารที่เขียนด้วย XML จะเป็นเอกสารที่สื่อความหมายได้ดีเพราะภาษา XML จะมีลักษณะของการให้ความหมายแก่ข้อมูล โดยผู้สร้างเอกสารจะเป็นผู้กำหนดนิยามและความหมายของแท็กต่างๆ เองตามข้อตกลงของ W3C ซึ่งจะทำให้การอ่านเอกสารเป็นไปได้ง่ายและเป็นมาตรฐาน เอกสารจะมีความชัดเจน ไม่กำกวม
- XML ออกแบบอย่างพิถีพิถันเน้นความจำเป็น กระทัดรัด เข้าใจง่าย และได้ประโยชน์กว้างขวาง
- XML สนับสนุนการประยุกต์เข้ากับงานต่าง ๆ และสนับสนุน โปรแกรมประยุกต์ต่าง ๆ
- XML เน้นเรื่องการประมวลผลเอกสาร จึงเหมาะกับงานทางด้านการวิเคราะห์เอกสาร การผลิตเอกสาร การแลกเปลี่ยน และการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเขียน XML ทำได้ตั้งแต่การใช้ Text editor ทั่ว ๆ ไป และไม่ต้องการเครื่องมือที่ซับซ้อน อย่างไรก็ตามก็ย่อมต้องมีผู้เขียน XML editor ให้ใช้งานได้ง่ายขึ้น
- XML เป็นมาตรฐานที่กำหนดแล้วใช้งานได้ทันที โดยที่ Browser และอุปกรณ์ต่าง ๆ พร้อม ใช้งานร่วมกัน

#### 4.3 การประยุกต์ใช้ XML

ในปัจจุบันมีการจัดการเอกสารจำนวนมากการบนเครือข่าย เช่น การสร้างห้องสมุดดิจิทัล การ แลกเปลี่ยนข้อมูลข่าวสารระหว่างกัน การประยุกต์ใช้ XML จึงทำได้อย่างกว้างขวาง ตัวอย่างเช่น

- XML สามารถใช้ได้หลากหลายภาษา และผสมกันได้หลากหลายภาษา เนื่องจาก XML สนับสนุน UNICODE
- การพัฒนา XML Processor ทำให้สามารถดึงเอกสาร XML มาใช้งานได้ง่าย และใช้ร่วมกับ โปรแกรมประยุกต์อื่นได้ง่าย เช่น โปรแกรม DB2, Oracle, SAP เป็นต้น
- XML ช่วยทำให้เกิดการรับส่งข้อมูลแบบ EDI โดยทำให้แนวทางการเชื่อมโยงและสร้าง ความเป็นเอกสารหรือมาตรฐานระหว่างองค์กร
- XML มีสภาพช่วยในการขนส่งข้อมูลไปยังปลายทางเพื่อให้แปลความหมายและใช้งานได้ อย่างเต็มประสิทธิภาพ
- มีการสร้างการประยุกต์ และนำเสนอผลลัพธ์ไปใช้งานจาก XML ได้มาก
- การประยุกต์การดำเนินกิจกรรมบนเครือข่ายมีมาก เช่น e-Business, EDI, e-Commerce การ จัดการ Supply chain, Demand chain management การดำเนินการแบบ intranet และ web base application

#### 4.4 XML จะแทนที่ HTML หรือไม่

แม้ว่าแต่เดิม XML จะถูกสร้างขึ้นมาเพื่อแก้ไขข้อบกพร่องของภาษา HTML แต่ XML ก็ไม่ใช่สิ่ง ที่จะมาแทน HTML อย่างที่หลายๆ คนเข้าใจผิด เพราะ XML เป็นเพียงส่วนเนื้อหาของเอกสารเท่านั้น ส่วนในเรื่องการแสดงผลนั้น ลำพัง XML อย่างเดียวไม่สามารถบอกได้ว่าจะแสดงผลเป็นเช่นไร (ซึ่งก็ คือ XML ไม่สามารถแสดงผลได้ด้วยตัวของมันเอง) ดังนั้น HTML ก็จะต้องคงอยู่ต่อไปอย่างแน่นอน อย่าง น้อยก็อีกระยะหนึ่ง ในขณะที่เดียวกัน HTML ก็จะมีพัฒนาการและการปรับเปลี่ยน โดยดึงเอาข้อดีของทั้ง XML และ HTML มารวมกันกลายเป็น XHTML ขึ้นมาใช้งานแทน

#### 4.5 โครงสร้าง XML

XML ถูกวางให้มีบทบาทสำคัญในฐานะที่เป็นตัวกลางของการสื่อสารระหว่างโปรแกรม เพราะเราสามารถสื่อสารระหว่างระบบงานที่แตกต่างกันโดยใช้ความสามารถของ XML SOAP และ HTTP ซึ่งจะไม่มีปัญหาเกี่ยวกับ firewall เพราะ HTTP ใช้ port 80 ในการติดต่อสื่อสาร นี่ก็คือตัวอย่างของไฟล์ XML

```

Prolog
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="show_book.xsl"?>
<!DOCTYPE catalog SYSTEM "catalog.dtd">
<!--catalog last updated 2000-11-01-->
<Catalog xmlns="http://www.example.com/catalog/">
  <book id="bk101">
    <author>#71;Abercrombie, Kim</author>
    <title>XML Developer's #x47;uide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description><![CDATA[An in-depth look at
Document Elements
creating applications with XML, using <, >, ]]> and
&amp;. </description>
  </book>
  <book id="bk109">
    <author>Kress, Peter</title>
    <title>Paradox Lost</title>
    <genre>Science Fiction</genre>
    <price>6.95</price>
    <publish_date>2000-11-02</publish_date>
    <description>After an inadvertant trip through
a Heisenberg Uncertainty Device, James Salway discovers the
problems of being quantum.</description>
  </book>
</catalog>

```

รูปที่ 4-1 ตัวอย่างไฟล์ XML

จากภาพข้างบนจะเห็นว่า XML จะประกอบด้วย 2 ส่วนคือ

1. ส่วนที่เป็น Prolog โดยบรรทัดแรกจะเหมือนการประกาศทั่วไปว่า XML ตัวนี้เป็น version 1 ซึ่งเป็น version ปัจจุบัน และถ้าเราต้องการให้สามารถแสดงภาษาไทยได้ ก็จะต้องเปลี่ยน encoding เป็น encoding="windows-874" ถัดบรรทัดลงมาจะเป็นการ link file XSL เข้ามาในไฟล์ XML เพื่อจัดรูปแบบการแสดงผล (เหมือนกับที่เรา link ไฟล์ CSS เข้ามาช่วยจัดรูป HTML ของเรา) บรรทัดต่อมา เป็นการกำหนดรูปแบบของไฟล์ XML ว่าถูกต้องตามที่กำหนดใน DTD หรือไม่ ที่ต้องมีการทำเช่นนี้ ก็เพราะว่าเราสามารถเพิ่มเติมส่วนต่างๆ เข้าใน XML ของเราได้ตามชอบใจ ซึ่งถ้าเราใช้ของเราคนเดียวก็จะมีปัญหาอะไร แต่ถ้าเราต้องมีการแลกเปลี่ยนข้อมูลกัน ซึ่งก็จะใช้รูปแบบเดียวกัน โดย DTD จะเป็นตัวกำหนดรูปแบบความถูกต้องของไฟล์ XML (Well Formed)

2. ส่วนที่เป็น Document Elements คือส่วนที่เป็นข้อมูล โดยจะแบ่งข้อมูลเป็นส่วนๆ เป็น element โดย element บนสุดจะเรียก Root tag หรือ โหนดแม่ แล้วจึงค่อยแตกเป็น โหนดลูก จากตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

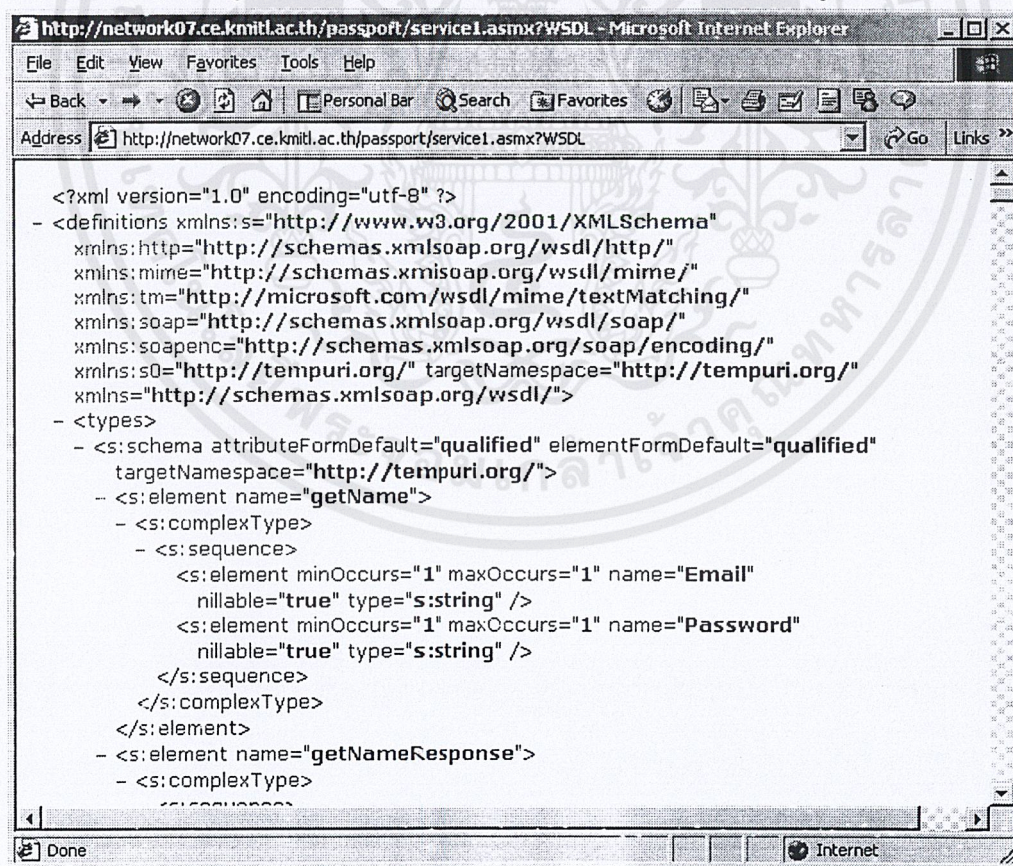
ข้างบนนี้ เราจะมี catalog เป็น Root node และมี โหนดลูกเป็น book แต่ละเล่ม และใน โหนดลูกก็สามารถ จะมี element ย่อย ๆ ได้อีกตามความต้องการ และในแต่ละ element ก็อาจจะมี attribute ได้เหมือนกันโดย จะอยู่ในรูป " " จากตัวอย่างข้างบน เราใช้ attribute id เป็นตัวแยก book แต่ละเล่ม

#### 4.6 ข้อควรระวังในการเขียน XML

ในการสร ารเอกสาร XML มีข้อควรระวัง ดังต่อไปนี้

- เมื่อเปิดแท็กแล้วจะต้องปิดแท็กด้วยเสมอ เช่น <p> ย่อหน้าที่ 1 </p>
- ในแท็กของ XML นั้น อักษรตัวใหญ่และเล็กนั้นมีความแตกต่างกัน (Case Sensitive) เช่น <Ticket>จะไม่เหมือนกับ <ticket>
- การจัดการเรื่องตำแหน่งแท็กของ XML ต้องเป็นไปตามลำดับ คือจำเป็นต้องเปิดและ ปิดเป็น วงๆ ไป เช่น <b><i> ข้อความนี้จะเป็นตัวหนาและเอียง </i></b>
- จำเป็นต้องมี Root Tag โดยมีแท็กลูกอื่นๆ อยู่ในวงของแท็กแม่นี้ และถ้าแท็กลูกมีแท็ก หลาน แท็กหลานก็จะต้องอยู่เป็นวงๆ ภายในกรอบของแท็กลูก จะไม่มีการฉีกวงได้ เหมือนใน HTML
- attribute ของ XML ต้องอยู่ ภายในกรอบของเครื่องหมายพินหนุ (") เสมอ

เมื่อเราพิมพ์ Browse ดูไฟล์นี้ใน โปรแกรม Internet Explorer จะได้ผลลัพธ์ ดังรูป



รูปที่ 4-2 ผลลัพธ์จาก Browse ดูในโปรแกรม Internet Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.7 วิเคราะห์โครงสร้างของเอกสารแบบ DOM และ SAX

ในการพัฒนาแอปพลิเคชัน โดยมีการนำ XML ไปใช้งานนั้น สิ่งหนึ่งที่ผู้พัฒนาจำเป็นต้องทำคือ การพาร์ส (parse) เอกสาร XML เพราะเนื่องจากว่าโปรแกรมจะมองเห็นเอกสารเป็นเพียงอักขระธรรมดาๆ เท่านั้น เราจึงต้องมีการเขียนโปรแกรมเพื่อให้แอปพลิเคชันสามารถเข้าใจได้ว่าเอกสาร XML ของเรามีการจัดโครงสร้างเป็นอย่างไร โดยสิ่งที่เราใช้ในการพาร์สเอกสารเราจะเรียกว่าพาร์เซอร์ (Parser) จากตัวอย่างที่ผ่านมาแล้ว เราสามารถใช้โปรแกรม Internet Explorer มาใช้กับเอกสาร XML ของเราได้ เนื่องจากโปรแกรมนี้อาจได้มีการฝังโปรแกรมพาร์เซอร์เอาไว้ในตัวอยู่แล้ว เรียกว่า Microsoft XML Parser เราจึงไม่ต้องทำการกำหนดการอ้างอิงพาร์เซอร์เหมือนเวลาที่เรเขียนโปรแกรมทั่วไป ในปัจจุบันมีมาตรฐานหรือวิธีการหลักๆ อยู่ 2 อย่างในการพาร์สเอกสาร XML ได้แก่ DOM (Document Object Model) และ SAX (Simple API for XML) ซึ่งทั้ง 2 วิธีนี้ต่างก็มีจุดเด่นและจุดด้อยอยู่ในตัวเอง การใช้งานจึงขึ้นกับความต้องการในการใช้งานของแอปพลิเคชันที่ใช้เอกสาร XML นั้นว่ามีความเหมาะสมกับวิธีการใดมากที่สุด

##### 4.7.1 การพาร์สเอกสาร XML โดยใช้ DOM (Document Object Model)

XML DOM เป็น API (Application Programming Interface) สำหรับเอกสาร XML ซึ่งจะกำหนดว่าเอกสาร XML จะมีการเข้าถึง การโยกย้ายและการจัดการต่างๆ ได้อย่างไรบ้าง ด้วยวิธีการแบบ DOM ผู้พัฒนาจะสามารถเข้าไปในโครงสร้างระดับต่างๆ ของเอกสาร XML ได้ รวมทั้งยังสามารถเพิ่ม ลบ และแก้ไขอีลิเมนต์ต่างๆ ของเอกสารได้อีกด้วย

DOM เป็นมาตรฐานของ W3C โดยมีเป้าหมายคือ ต้องสามารถสร้างอินเตอร์เฟซที่เป็นมาตรฐาน และจะต้องใช้ได้กับแอปพลิเคชันได้หลากหลายการทำงานของ DOM จะเริ่มต้นเมื่อโปรแกรมหรือแอปพลิเคชันเรียกพาร์เซอร์เพื่อทำการโหลดเอกสาร XML เข้าสู่หน่วยความจำของคอมพิวเตอร์ เมื่อเอกสารถูกโหลดเสร็จเรียบร้อยแล้ว DOM ก็จะทำให้โปรแกรมสามารถดึงข้อมูลมาใช้งานหรือดำเนินการต่างๆ กับข้อมูลได้ DOM จะมีมุมมองไปยังเอกสาร XML เป็นโครงสร้างแบบต้นไม้ (Tree) โดยระดับบนสุดของทรีเราจะเรียกว่า documentElement ซึ่งอีลิเมนต์ดังกล่าวจะแตกสาขาออกไปเป็นอีลิเมนต์ย่อยๆ ตั้งแต่ 1 อีลิเมนต์ขึ้นไป ซึ่งเราจะเรียกว่า โหนดลูก (childNodes) จากเอกสาร XML ที่มีรายละเอียดดังด้านล่างนี้

```

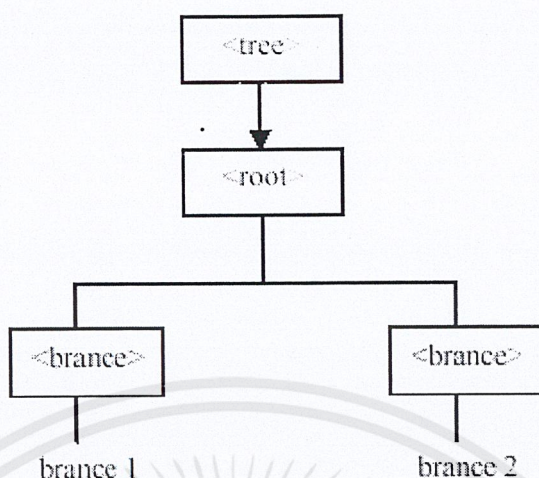
<tree>
  <root>
    <brance> brance 1 </brance>
    <brance> brance 2 </brance>
  </root>
</tree>

```

รูปที่ 4-3 ตัวอย่างไฟล์ XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DOM จะมองเอกสารข้างต้นเป็น คังนี้คือ



รูปที่ 4-4 DOM Tree

#### จุดเด่นของ DOM

- โปรแกรมจะพาร์สเอกสารเพียงครั้งเดียว เมื่อ DOM ได้สร้างทรีขึ้นมาแล้ว ทรีนั้นก็อยู่ในหน่วยความจำตลอดจนกว่าเราจะสั่งยกเลิก และในขณะที่ทรีอยู่ในหน่วยความจำ เราก็สามารถเข้าถึงข้อมูลที่อยู่ภายในทรีได้อย่างรวดเร็ว
- ข้อมูลที่สร้างเป็นทรีจะทำให้ข้อมูลเกิดความสัมพันธ์ ทำให้การเรียกใช้ทำได้ง่าย

#### จุดด้อยของ DOM

- ในกรณีที่เอกสารมีขนาดใหญ่และซับซ้อน การสร้างทรีจะทำให้ช้า และกินหน่วยความจำมาก
- ผู้พัฒนาโปรแกรมจำเป็นต้องเข้าใจถึงโครงสร้างโดยรวมของเอกสารเป็นอย่างดี จึงจะสามารถเขียนโปรแกรมโดยใช้ DOM ได้ กล่าวคือในส่วนของการทำงาน โค้ดด้วยวิธีการของ DOM นั้น จะค่อนข้างยาวและยุ่งยากอยู่พอสมควร

#### 4.7.2 การพาร์สเอกสาร XML โดยใช้ SAX (Simple API for XML)

SAX เป็น API ที่มีลักษณะการทำงานก็คือเมื่อมีเหตุการณ์ (เช่นการร้องขอข้อมูล) ใดๆ เกิดขึ้น หรือเรียกโมเดลการทำงานแบบนี้ว่า event-based model กล่าวคือ SAX จะไม่มีการสร้างภาพรวมของเอกสารไว้ก่อนเลยว่า เอกสารนั้นมีโครงสร้างเป็นอย่างไร เมื่อแอปพลิเคชันมีการร้องขอมาจึงจะรายงานหรือทำงานตามเหตุการณ์ของการทำพาร์สซึ่ง (parsing) ไปยังแอปพลิเคชัน

หลักการการทำงานของ SAX จะมองเพียงจุดเริ่มต้นและจุดสิ้นสุดของเอกสารและของอีลีเมนต์เท่านั้น และจะสนใจเฉพาะสิ่งที่ต้องการเท่านั้น ทำให้ SAX เข้าถึงข้อมูลได้ง่ายกว่า DOM รวมถึงการเขียนโปรแกรมก็ทำได้ง่ายกว่า เพราะมองข้อมูลในลักษณะเส้นตรงและสามารถพาร์สเอกสารที่มีขนาดใหญ่มากกว่าจำนวนหน่วยความจำที่เรียกได้ ดังนั้น โปรแกรมระบบฐานข้อมูลหลายๆ หลายบริษัทจึงนิยมใช้วิธีนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเอกสาร XML ที่มีรายละเอียดดังด้านล่างนี้ SAX จะมองเอกสารข้างต้นไปที่ละบรรทัด ในลักษณะเหตุการณ์เชิงเส้น ดังนี้

```

<tree>
  <root>
    <brance> brance 1 </brance>
    <brance> brance 2 </brance>
  </root>
</tree>

```

รูปที่ 4-5 ตัวอย่างไฟล์ XML

SAX จะมองเอกสารข้างต้นไปที่ละบรรทัด ในลักษณะเหตุการณ์เชิงเส้น ดังนี้

start document  
 start element: doc  
 start element: element1  
 characters: Value 1  
 end element: element1  
 start element: element2  
 characters: Value 2  
 end element: element2  
 end element: doc  
 end document

พาร์สเอกสาร  
 ตั้งแต่บรรทัดแรก  
 จนถึงบรรทัดสุดท้าย  
 แบบเส้นตรง

รูปที่ 4-6 SAX Event

ตัวอย่างการพาร์สแบบ SAX ข้างต้นเป็นการพาร์สทั้งเอกสาร ในทางปฏิบัติแล้วเรามักไม่ได้มีการพาร์สเอกสารทั้งหมด เมื่อเราได้ข้อมูลที่เราต้องการแล้วเราก็จะหยุด วิธีการพาร์สแบบ SAX จึงทำงานได้อย่างรวดเร็วและเขียนโปรแกรมได้ง่ายมาก

ตัวอย่างการพาร์สแบบ SAX ข้างต้นเป็นการพาร์สทั้งเอกสาร ในทางปฏิบัติแล้วเรามักไม่ได้มีการพาร์สเอกสารทั้งหมด เมื่อเราได้ข้อมูลที่เราต้องการแล้วเราก็จะหยุด วิธีการพาร์สแบบ SAX จึงทำงานได้อย่างรวดเร็วและเขียนโปรแกรมได้ง่ายมาก

#### จุดเด่นของ SAX

- เข้าถึงข้อมูลได้ง่ายและรวดเร็ว
- เขียนโปรแกรมได้ง่าย เนื่องจากการเข้าถึงข้อมูลเป็นลักษณะเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้หน่วยความจำน้อย สามารถเข้าถึงข้อมูลหรือเอกสารที่มีขนาดใหญ่กว่าหน่วยความจำที่มีของเครื่องได้

#### จุดด้อยของ SAX

- ไม่สามารถบอกความสัมพันธ์ของข้อมูลได้
- เป็นวิธีการที่ไม่ค่อยมีประสิทธิภาพหากเป็นการเข้าถึงข้อมูลเดิมบ่อยๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

SOAP

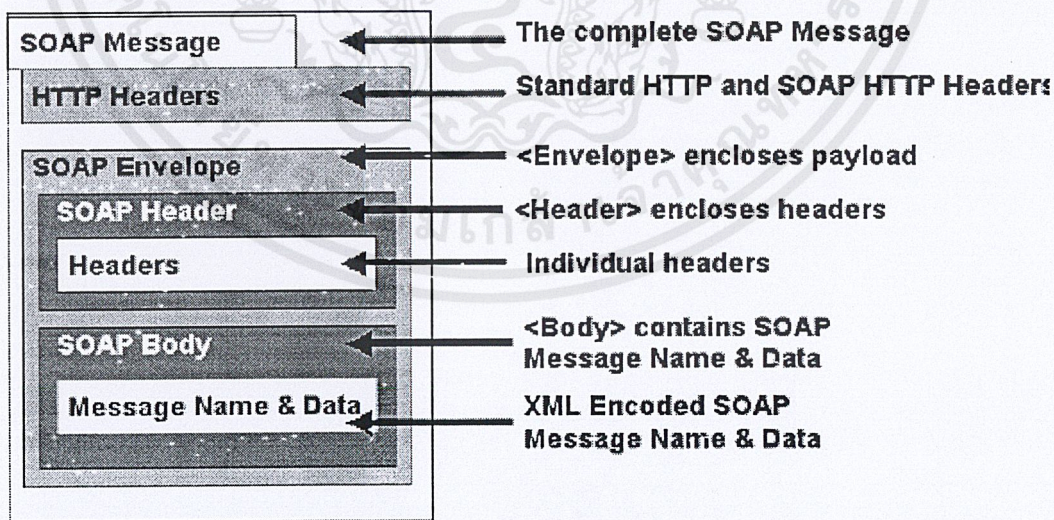
5.1 แนะนำ SOAP

SOAP (Simple Object Access Protocol) เป็น โพรโตคอลที่ใช้ XML เป็นพื้นฐาน เพื่อให้ซอฟต์แวร์คอมพิวเตอร์ และแอปพลิเคชันสามารถติดต่อกันผ่าน HTTP ซึ่งเป็นมาตรฐานอินเทอร์เน็ต โพรโตคอลได้เนื่องจากการสื่อสารระหว่างแอปพลิเคชัน เป็นสิ่งจำเป็นต่อการพัฒนาแอปพลิเคชันบนอินเทอร์เน็ตเป็นอย่างมาก แต่แอปพลิเคชันแบบกระจายกันทำงาน (distributed application) ในปัจจุบันใช้ RemoteProcedure Call (RPC) สื่อสารกันระหว่างออบเจกต์ เช่น DCOM และ CORBA ซึ่งไม่ได้ใช้พอร์ตเดียวกับHTTP ที่เป็นพอร์ตมาตรฐานสำหรับให้บริการเว็บ ดังนั้น RPC จึงนำมาปรับใช้กับอินเทอร์เน็ตได้ยาก และมีปัญหาทางด้านความปลอดภัย ไฟร์วอลล์และพร็อกซีเซิร์ฟเวอร์จะไม่ยอมให้ส่งข้อมูลชนิดนี้ได้ตามปกติ วิธีที่คิดก็คือใช้ HTTP เพราะเป็นที่ยอมรับโดยอินเทอร์เน็ตบราวเซอร์ และเซิร์ฟเวอร์ทุกชนิด ซึ่ง SOAP ถูกสร้างมาเพื่อใช้ในกรณีนี้

ข้อดีของ SOAP ก็คือ SOAP ไม่ขึ้นกับคอมพิวเตอร์เทคโนโลยี และภาษาการเขียนโปรแกรมใดๆ สามารถเขียนได้ง่ายและขยายเพิ่มเติมได้

5.2 ส่วนประกอบของ SOAP

SOAP เมสเสจใช้ไวยากรณ์ของ XML ในการสร้าง ประกอบด้วย 3 อีลีเมนต์มาตรฐาน คือ SOAP Envelope, SOAP Header และ SOAP Body



รูปที่ 5-1 โครงสร้างของ SOAP

SOAP เมสเสจ จะต้องเป็นไปตามกฎนี้

- Envelope เป็นอีลีเมนต์ที่อยู่บนสุด ต้องมีอีลีเมนต์นี้เสมอ
- Header อาจจะมีหรือไม่มีก็ได้ แต่ถ้ามีต้องเป็น child element แรกของ Envelope
- Body ต้องเป็น child element แรกของ envelope หรือ header

ตัวอย่างของ SOAP เมสเสจ ตัวอย่างนี้คือ GetLastTradePrice SOAP Request ที่ส่งไปยัง StockQuote service ประกอบด้วยสตริงพารามิเตอร์ symbol และค่านำ float กลับมากับ SOAP response SOAP envelope element อยู่ที่จุดบนสุดของเอกสาร XML XML namespace ใช้เพื่อป้องกันการสับสนของ SOAP identifier

SOAP request จะเป็นดังนี้

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response จะเป็นดังนี้

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope
```

```

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
<SOAP-ENV:Body>
  <m:GetLastTradePriceResponse xmlns:m="Some-URI">
    <Price>34.5</Price>
  </m:GetLastTradePriceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

นอกจากนี้ใน SOAP เมสเสจ จะมีการใช้ XML namespace ทุกๆ อีลีเมนต์ในเอกสารจะขึ้นต้นด้วย namespace namespace จะถูกกำหนดโดยใช้ xmlns attribute ดังนี้

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

```

แต่หากไม่ต้องการให้ขึ้นต้นด้วย namespace ก็สามารถทำได้ ดังนี้

```

<Envelope
  xmlns="http://schemas.xmlsoap.org/soap/envelope/">

```

### 5.2.1 Envelope

envelope element เป็นอีลีเมนต์บนสุดของเอกสาร XML ที่ใช้แสดงเมสเสจ อาจประกอบด้วยแอตทริบิวต์ คือ envelope namespace และ encodingStyle

1. Envelope Namespace – SOAP เมสเสจจะแสดงเวอร์ชัน โดยใช้ namespace ของ envelope element namespace จะถูกนำไปใช้ ขึ้นต้น envelope, header และ body element

2. EncodingStyle Attribute – ใช้แสดงวิธี serialization ของ SOAP เมสเสจ แอตทริบิวต์นี้

สามารถ

มีได้ในทุกอีลีเมนต์และจะมีผลกับ content ของอีลีเมนต์นั้นและ child element ทั้งหมดของมันที่ไม่ได้ประกาศแอตทริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.2.2 Body

เป็นส่วนของข้อมูลที่ใช้ในการแลกเปลี่ยน โดยทั่ว ๆ ไปข้อมูลจะเป็นการ marshall RPC call และ error report child element ของ body element จะถูกเรียกว่า Body entry body entry จะต้องเป็นไปตามกฎนี้

- body entry จะต้องแสดงด้วยชื่อเต็มประกอบด้วย namespace URI และ ชื่อของมัน (local name)
- SOAP encodingStyle attribute อาจจะมีได้ เพื่อแสดงถึงวิธี encode ของ body entry นั้น

### 5.2.3 Header

เป็นส่วนเพิ่มเติมของเมสเซจสามารถประกอบด้วยอินฟอร์เมชันที่ระบุไปยังแอปพลิเคชัน ส่วนเพิ่มเติมนี้อาจนำไปใช้เพื่อเป็น authentication, transaction management เป็นต้น ทุก ๆ child element ของ header element จะถูกเรียกว่า Header entry header entry จะต้องเป็นไปตามกฎดังนี้

- header entry จะต้องแสดงด้วยชื่อเต็มประกอบด้วย namespace URI และ ชื่อของมัน (local name)
- อาจมี SOAP encodingStyle attribute ที่ใช้สำหรับ header entry
- SOAP mustUnderstand attribute และ SOAP actor attribute จะมีหรือไม่มีก็ได้ เพื่อใช้บอกว่าจะจัดการกับ entry นั้นอย่างไรและโดยใคร

```
<soap:Header>
<m:local xmlns:m="http://www.w3schools.com/local/"
soap:actor="http://www.w3schools.com/appml" />
<m:language>en</m:language>
<m:currency>USD</m:currency>
</m:local>
</soap:Header>
```

1) SOAP Actor Attribute – SOAP เมสเซจสามารถถูกส่งไปยังปลายทางผ่านตัวกลางของ SOAP (SOAP intermediary) ตัวกลางของ SOAP คือแอปพลิเคชันที่มีความสามารถทั้งรับและส่งต่อ SOAP เมสเซจ ในการส่งต่อตัวกลางจะต้องไม่ส่งต่อ header element ไปให้กับแอปพลิเคชัน โดยตัวกลางนี้จะถูกกำหนดเป็น URI ถ้าไม่มี attribute นี้หมายถึงผู้รับ SOAP เมสเซจเป็นปลายทางสุดท้าย

2) SOAP MustUnderstand Attribute – ใช้แสดงว่า header entry นี้จำเป็นหรือเป็นเพียงออปชัน สำหรับผู้รับที่จะนำไปประมวลผล ค่าของมันคือ “1” หรือ “0” ถ้าไม่มีการระบุจะมีค่าเท่ากับ “0” โดย “0” หมายถึงเป็นออปชัน และ “1” หมายถึงจำเป็น โดยจะต้องประมวลผลให้ถูกต้องตาม semantic หรือต้อง fail เช่น เมสเซจเป็นส่วนหนึ่งของ transaction ถ้าปลายทางไม่รองรับ transaction จะต้องไม่ประมวลผลและส่งผลผิดพลาดกลับไปในกรณี ที่ mustUnderstand attribute เป็น “1” แต่ถ้าเป็น “0” จะยังคงประมวลผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 SOAP Fault Element

ข้อความแสดงความผิดพลาดจากแอปพลิเคชันของ SOAP จะเก็บอยู่ใน fault element ซึ่งถ้ามีจะตั้ง  
องปรากฏใน body element เพียงครั้งเดียวใน SOAP เมสเซจ ตัวอย่างอาจเป็นดังนี้

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<soap:Body>
<soap:Fault>
<faultcode>soap:MustUnderstand</faultcode>
<faultstring>Mandatory Header error.</faultstring>
<faultactor>http://www.wrox.com/heroes/endpoint.asp</faultactor>
<detail>
<w:source xmlns:w="http://www.wrox.com/">
<module>endpoint.asp</module>
<line>203</line>
</w:source>
</detail>
</soap:Fault>
</soap:Body>
</soap:Envelope>
```

SOAP fault element มี element ย่อย ๆ ดังตารางนี้

Sub Element	Description
<faultcode>	โค้ดที่ระบุถึงการ error
<faultstring>	ข้อความการ error
<faultactor>	ใครเป็นสาเหตุของการ error
<detail>	ระบุอินฟอร์เมชันของการ error

ตารางที่ 5-1 SOAP Fault Element

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ faultcode สามารถมีค่าได้ ดังตารางนี้

Error	Description
VersionMismatch	namespace ภายใน SOAP Envelope element ไม่ถูกต้อง
MustUnderstand	child element ของ Header element กับ mustUnderstand attribute ที่มีค่า "1" ผู้รับไม่รองรับ
Client	เมสเสจมีรูปแบบไม่ถูกต้อง หรือมีอินโฟร์เมชันที่ไม่ถูกต้อง
Server	เกิดปัญหาเกี่ยวกับเซิร์ฟเวอร์ ไม่สามารถประมวลผลได้

ตารางที่ 5-2 ค่า faultcode

#### 5.4 SOAP Encoding

SOAP encoding มีวิธีการ map จาก type ของโปรแกรมมิ่ง ไปเป็น XML 2 วิธี คือจากภายนอก โดยใช้ WSDL (Web Services Description Language) ที่บอกถึง type ของข้อมูลที่รับหรือส่ง หรือใช้ xsi:typeattribute ในกรณีที่ใช้ภาษาที่ไม่รองรับ WSDL โดยทั้ง 2 วิธีจะใช้ XML schema ในการระบุ type ตัวอย่าง ของการใช้ xsi จะเป็นดังนี้

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <soap:Body>
    <m:MixedMessage xmlns:m="http://www.wrox.com/mix/">
      <param1 xsi:type="xsd:string">OU812</param1>
      <param2 xsi:type="xsd:integer">2001</param2>
      <param3 xsi:type="xsd:double">3.14159</param3>
    </m:MixedMessage>
  </soap:Body>
</soap:Envelope>
```

SOAP รองรับ type ของข้อมูลได้ทั้ง simple type และ compound type เช่น struct และ array

### 5.4.1 ตัวอย่าง Struct

struct book ที่เขียนด้วยภาษา c++ จะเป็นดังนี้

```
struct Book
{
    string author;
    string name;
    int page;
};
```

Soap เมสเซจที่ไม่ใช้ WSDL แต่ใช้ xsi จะเป็นดังนี้

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:searchBook xmlns:ns1="http://soapinterop.org/" SOAPENV:
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <bookResult xmlns:ns2="http://soapinterop.org/xsd" xsi:type="ns2:Book">
        <author xsi:type="xsd:string">Ed Roman</author>
        <name xsi:type="xsd:string">Mastering EJB</intro>
        <page xsi:type="xsd:int">500</page>
      </bookResult>
    </ns1:searchBook>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

แต่ถ้าใช้ WSDL ในส่วนของการประกาศ schema จะต้องประกอบด้วย schema ดังนี้

```
<types>
<schema targetNamespace='http://soapinterop.org/xsd'
xmlns='http://www.w3.org/2001/XMLSchema'
xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wSDL='http://schemas.xmlsoap.org/wSDL/'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

elementFormDefault='qualified'>
<complexType name="Book">
<sequence>
<element name='author' type='string'/>
<element name='name' type='string'/>
<element name='page' type='int'/>
</sequence>
</complexType>
</schema>
</types>

```

#### 5.4.2 ตัวอย่าง Array

ตัวอย่างนี้เป็น array ของ struct book ในตัวอย่างก่อน Soap เมสเซจที่ไม่ใช่ WSDL แต่ใช้ xsi จะเป็นดังนี้

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<ns1:BookArrayResult xmlns:ns1="http://soapinterop.org/" SOAPENV:
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<BookArray xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns2:Array" xmlns:ns3="http://soapinterop.org/xsd"
ns2:arrayType="ns3:Book[3]">
<item xsi:type="ns3:Book">
<author xsi:type="xsd:string">Ed Roman</author>
<name xsi:type="xsd:string">Mastering EJB</name>
<page xsi:type="xsd:int">500</page>
</item>
<item xsi:type="ns3:Book">
<author xsi:type="xsd:string">Roger Session</author>
<name xsi:type="xsd:string">The Battle of the middleware</name>
<page xsi:type="xsd:int">350</page>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</item>
<item xsi:type="ns3:Book">
<author xsi:type="xsd:string">Chris Dix</author>
<name xsi:type="xsd:string">Professional XML Web Service</name>
<page xsi:type="xsd:int">550</page>
</item>
</BookArray>
</ns1:BookArrayResult>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

แต่ถ้าใช้ WSDL ในส่วนของการประกาศ schema จะต้องประกอบด้วย schema ดังนี้

```

<types>
<schema targetNamespace='http://soapinterop.org/xsd'
xmlns='http://www.w3.org/2001/XMLSchema'
xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
elementFormDefault='qualified'>
<complexType name="Book">
<sequence>
<element name='author' type='string'/>
<element name='name' type='string'/>
<element name='page' type='int'/>
</sequence>
</complexType>
<complexType name='ArrayOfBook'>
<complexContent>
<restriction base='SOAP-ENC:Array'>
<attribute ref='SOAP-ENC:arrayType' wsdl:arrayType='typens:Book[]'/>
</restriction>
</complexContent>
</complexType>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</schema>
</types>
```

## 5.5 SOAP ใน HTTP

ในการส่ง SOAP ผ่านทาง HTTP จะต้องใช้ content-type เป็น text/xml แต่ใน SOAP request จะต้องมี header SOAPAction ภายใน HTTP header SOAPAction จะเป็นตัวบอกให้เซิร์ฟเวอร์รู้ว่า HTTP Post นั้นเป็น SOAP เมสเสจ และค่าของ header คือ URI ที่แสดงถึงจุดหมายของ SOAP เมสเสจ ส่วน SOAP response จะต้องมี status code ตามมาตรฐานของ HTTP โดย 200-299 แสดงว่าสำเร็จ แต่ถ้า response เมสเสจ เป็นการfault แล้ว status code จะต้องเป็น 500 ซึ่งแสดงถึง internal server error ตัวอย่างของ response ที่มี status code เป็น 500 อาจเป็นดังนี้

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/xml
Content-Length: ###
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<soap:Body>
<soap:Fault>
<faultcode>soap:VersionMismatch</faultcode>
<faultstring>The SOAP namespace is incorrect.</faultstring>
<faultactor>http://www.wrox.com/endpoint.asp</faultactor>
<detail>
<w:errorinfo xmlns:w="http://www.wrox.com/">
<desc>The SOAP namespace was blank.</desc>
</w:errorinfo>
</detail>
</soap:Fault>
</soap:Body>
</soap:Envelope>
```

## 5.6 SOAP สำหรับ RPC

จุดประสงค์ของการออกแบบ SOAP คือทำ RPC โดยใช้ XML การเรียก RPC นั้นจะ map เข้ากับ HTTP request และ RPC response จะ map กับ HTTP response

ในการทำ mehod call จำเป็นที่จะต้องมียูริลโฟร์เมชันดังนี้

1. URI ของออบเจ็กต์ปลายทาง (target object)
2. ชื่อของเมธอด
3. คำอธิบายของเมธอด (method signature) ส่วนนี้เป็นออปชัน
4. พารามิเตอร์สำหรับเมธอด
5. ข้อมูลส่วนหัว (header data) เป็นออปชัน

พิจารณาเมธอดดังนี้

```
double GetStockQuote ( [in] string sSymbol);
```

ถ้า namespace ของเมธอดคือ “http://www.worxstox.com/” แล้วการเรียกเมธอด โดย request stock quote ใช้ symbol OU812 จะเป็นดังนี้

```
<q:GetStockQuote xmlns:q="http://www.wroxstox.com/">
<q:sSymbol xsi:type="xsd:string">OU812</q:sSymbol>
</q:GetStockQuote>
```

ชื่อเมธอดและชื่อของอีลีเมนต์จะต้อง match กันเช่นเดียวกับพารามิเตอร์ สำหรับ response จะต้องชื่ออีลีเมนต์จะต้องเป็นชื่อเมธอดตามด้วย response ดังนี้

```
<q:GetStockQuoteResponse xmlns:q="http://www.wroxstox.com/">
<q:ret xsi:type="xsd:double">100.0</q:ret>
</q:GetStockQuoteResponse>
```

ชื่ออีลีเมนต์ของค่าที่คืนมาสามารถเปลี่ยนได้ โดยจะกำหนดได้จากโปรแกรมที่เขียนหรือจาก WSDL

## 5.7 SOAP Toolkit

SOAP toolkit คือเครื่องมือที่จะทำหน้าที่ในการประมวลผล SOAP request และส่ง response กลับไปให้ไคลเอนต์ โดย SOAP toolkit แต่ละตัวใช้ได้บนแพลตฟอร์มที่ต่างกัน รองรับเทคโนโลยี web service ต่างกัน บางตัวอาจจะรองรับ WSDL หรือ UDDI ในขณะที่บางตัวรองรับอย่างใดอย่างหนึ่งหรือไม่รองรับทั้งสองอย่าง และถึงแม้ว่า SOAP จะเป็นอิสระต่อแพลตฟอร์ม แต่ใน SOAP toolkit บางตัวยังไม่สามารถทำงานข้ามผลิตภัณฑ์ (interoperability) ได้ เนื่องจากรองรับเทคโนโลยีของ web service ไม่เหมือนกัน เช่น xsi ในบางผลิตภัณฑ์จะบังคับให้ SOAP เมสเสจจะต้องระบุ type ด้วย xsi หากไม่มีก็จะ fail ซึ่งในปัจจุบันแต่ละผลิตภัณฑ์ก็พยายามพัฒนาให้สามารถทำงานด้วยกันได้

ชื่อ	แพลตฟอร์ม	แหล่งที่มา
4s4c	COM	<a href="http://www.4s4c.com">http://www.4s4c.com</a>
A SOAP for RPC NT Service	COM	<a href="http://www.whitemesa.com">http://www.whitemesa.com</a>
Apache Axis	Java	<a href="http://xml.apache.org/axis">http://xml.apache.org/axis</a>
Apache SOAP	Java	<a href="http://xml.apache.org/soap">http://xml.apache.org/soap</a>
CapeConnect	Java	<a href="http://www.capeclear.com">http://www.capeclear.com</a>
Glue	Java	<a href="http://www.themindelectric.com">http://www.themindelectric.com</a>
IBM Web Services Toolkit	Java	<a href="http://www.alphaworks.ibm.com">http://www.alphaworks.ibm.com</a>
IdooXoap for Java and C++	Java , C++	<a href="http://www.zvon.org">http://www.zvon.org</a>
Microsoft SOAP Toolkit	COM	<a href="http://www.microsoft.com">http://www.microsoft.com</a>
PocketSOAP	COM	<a href="http://www.pocketsoap.com">http://www.pocketsoap.com</a>
SOAP::Lite	Perl	<a href="http://www.soaplite.com">http://www.soaplite.com</a>
Visual Studio .NET	COM	<a href="http://www.microsoft.com">http://www.microsoft.com</a>

ตารางที่ 5-3 SOAP Toolkit

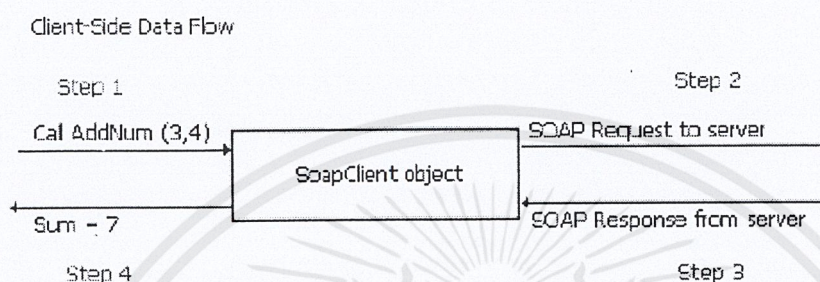
## 5.8 โครงสร้างภายในของ SOAP Toolkit ของ Microsoft

The Web Services Description Language (WSDL) เป็นรูปแบบหนึ่งของ XML เพื่อใช้อธิบายว่าเซิร์ฟเวอร์นั้นมีเซอร์วิส (Services) และ operation ภายในเซอร์วิสอะไรให้ใช้บ้าง ซึ่ง operation ภายในเซอร์วิสนั้นจะอธิบายรูปแบบที่ไคลเอนต์ (client) ต้องใช้ในการเรียก operation The Web Services Meta Language (WSML) file เป็นส่วนที่เพิ่มขึ้นมาจากไฟล์ WSDL เพื่อใช้ในการเชื่อมโยงแต่ละ operation ในเซอร์วิสกับ COM object โดย WSML file จะพิจารณาว่า COM object ตัวใดที่จะต้องถูกโหลด (load) ขึ้นมาเมื่อมีการเรียกใช้ operation ในเซอร์วิสทั้งไฟล์ WSDL และ WSML จะต้องอยู่ที่เซิร์ฟเวอร์ โดยไคลเอนต์ที่ต้องการที่จะส่ง SOAP request ไปที่เซิร์ฟเวอร์ จะต้องได้รับ WSDL จากเซิร์ฟเวอร์ก่อน เพื่อหารูปร่างแบบของ SOAP request ในการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.8.1 การสร้าง SoapClient Object

ทางฝั่งไคลเอนต์ เมื่อแอปพลิเคชันส่ง message ไปที่ Soap Client object เพื่อเรียกใช้ operation หนึ่งๆนั้น(ตามรูปนี้คือการบวกเลข 2 ตัว) SoapClient object จะทำการสร้าง SOAP request ส่งไปที่เซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ทำการประมวลผลเสร็จเรียบร้อยแล้ว ก็จะส่งผลลัพธ์ในรูปแบบของ SOAP response กลับมาให้ไคลเอนต์ หลังจากนั้น SoapClient object จะทำการแปลและส่ง message ที่เก็บผลลัพธ์ของ operation นั้นกลับไปที่แอปพลิเคชัน ตามไดอะแกรมด้านล่าง



รูปที่ 5-2 แสดงการ Client-Side Data Flow

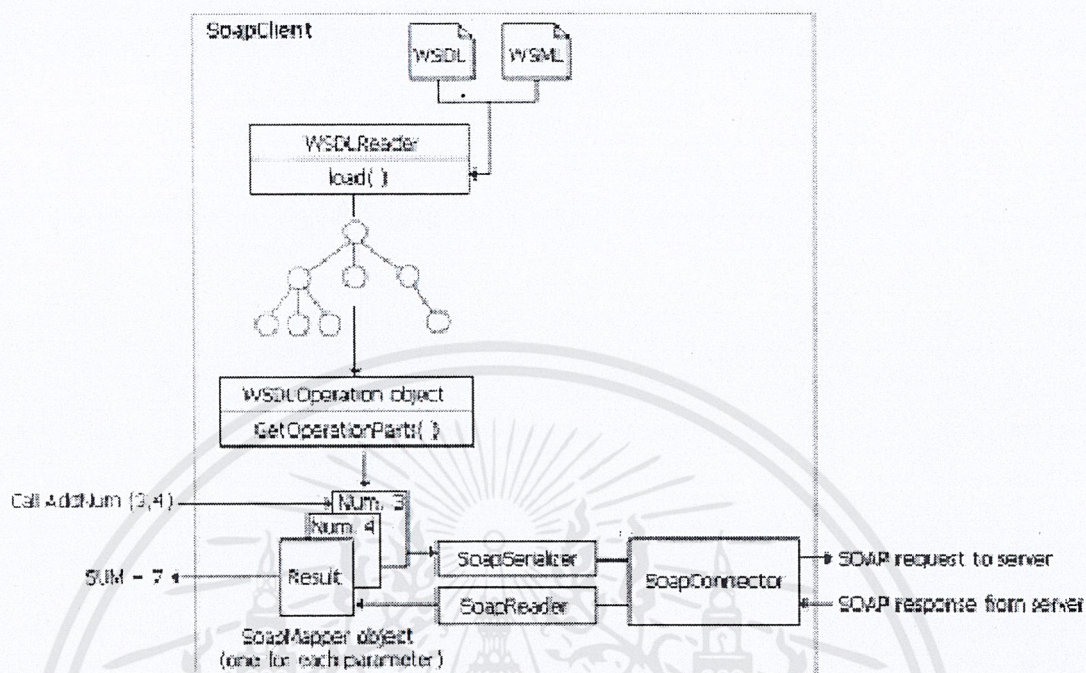
แอปพลิเคชันต้อง instantiate Soap object ก่อนที่จะทำการเรียกใช้ operation ใดๆ (SoapClient object นี้ถูกสร้างเก็บไว้ในไฟล์ MSSOAP1.dll) โดยใช้เมธอด SoapClient.mssoapinit (ชื่อไฟล์ WSDL, ชื่อเซิร์ฟเวอร์, ข้อมูลพอร์ต) โดยข้อมูลพอร์ตจะใช้ในการแบ่งแยกช่องทางที่ใช้ในการส่ง SOAP message ในระหว่าง initialize SoapClient object จะตรวจสอบเมธอดทั้งหมดที่อยู่ในไฟล์ WSDL ด้วยการทำงานแบบไดนามิกนี้ทำให้เราสามารถที่จะเรียกใช้เมธอดใดก็ได้ที่ได้อธิบายไว้ในไฟล์ WSDL ดังแสดงตาม Microsoft Visual Basic Scripting Edition (VBScript) code ด้านล่างนี้

```

SET soapclient = CreateObject("MSSOAP.SoapClient")
Call soapclient.mssoapinit("http://localhost/MyWSDL.wsdl", "TestService", "TestPort")
' Now the client can call an operation listed in the portType element
' specified when calling mssoapinit().
wscript.echo soapclient.AddNumbers(2, 3)
  
```

## 5.8.2 การประมวลผลภายใน SoapClient Object

การทำงานภายใน SoapClient object แสดงได้ดังรูปด้านล่าง



รูปที่ 5-3 แสดงโครงสร้างภายในของ SOAP

WSDL Reader object จะทำการโหลดไฟล์ WSDL และไฟล์ WSML เข้าไปไว้ใน DOM แล้วทำการวิเคราะห์ โดย WSDLReader object จะสร้าง WSDLOperation object เพื่อทำการเรียกใช้งาน operation (ตามรูปนี้คือ AddNum(3, 4)) หลังจากนั้น WSDLOperation จะเรียกเมธอด GetOperationParts เพื่อดึงรูปแบบ message ของอินพุตและเอาที่พู่ทของ operation ออกมา แล้ว SoapClient จะสร้าง Soap Mapper objects ให้กับแต่ละส่วนที่ได้จากเมธอด GetOperationParts และโหลดค่าที่ได้มาจากการเรียกใช้ operation เข้าไว้ใน objects เหล่านี้ พอถึงขั้นตอนนี้ SoapSerializer objects จะสร้าง SOAP request message จาก SoapMapper objects ที่เหมาะสมและส่งไปที่เซิร์ฟเวอร์เมื่อเซิร์ฟเวอร์ประมวลผลเสร็จแล้วก็จะส่ง SOAP response กลับไปที่ไคลเอนต์ SoapClient object จะทำการโหลดผลลัพธ์ของ operation ไปเก็บใน SoapMapper object ที่เหมาะสมแล้วส่งผลลัพธ์กลับไปให้แอปพลิเคชันของผู้ใช้

## 5.8.3 การสร้าง SoapServer Object

ส่วนทางฝั่งเซิร์ฟเวอร์ เราสามารถกำหนดรูปแบบการรับ SOAP request (SOAP listener) ได้ 2 รูปแบบ คือ Internet Server API (ISAPI) server และ Active Server Pages (ASP) server ในไฟล์ WSDL จะมี URL ที่ระบุว่าจะใช้ไฟล์ไคลเอนต์เซิร์ฟเวอร์ในการรับ SOAP request เพื่อใช้ในการพิจารณาว่าจะรับ SOAP request อย่างไรบนเซิร์ฟเวอร์ ตามโค้ดด้านล่างนี้ ตัวพิมพ์หนาจะเป็นส่วนที่ใช้ในการเรียกไปที่ ISAPI listener

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

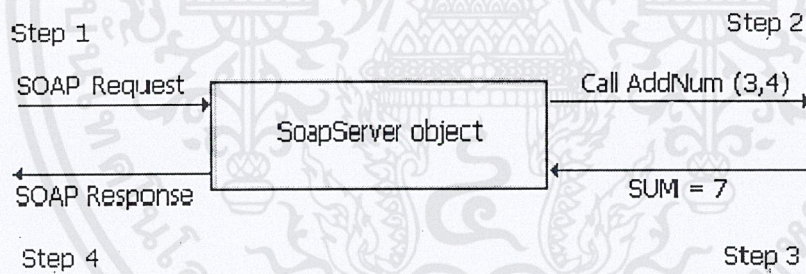
```

<definitions>
...
<service name='DocSample1' >
<port name='DocSample1PortType' binding='tns:DocSample1Binding' >
<soap:address
location='http://localhost/DocSample1Test/DocSample1.wsdl' />
</port>
</service>
...
</definitions>

```

### การทำงานทางฝั่งเซิร์ฟเวอร์

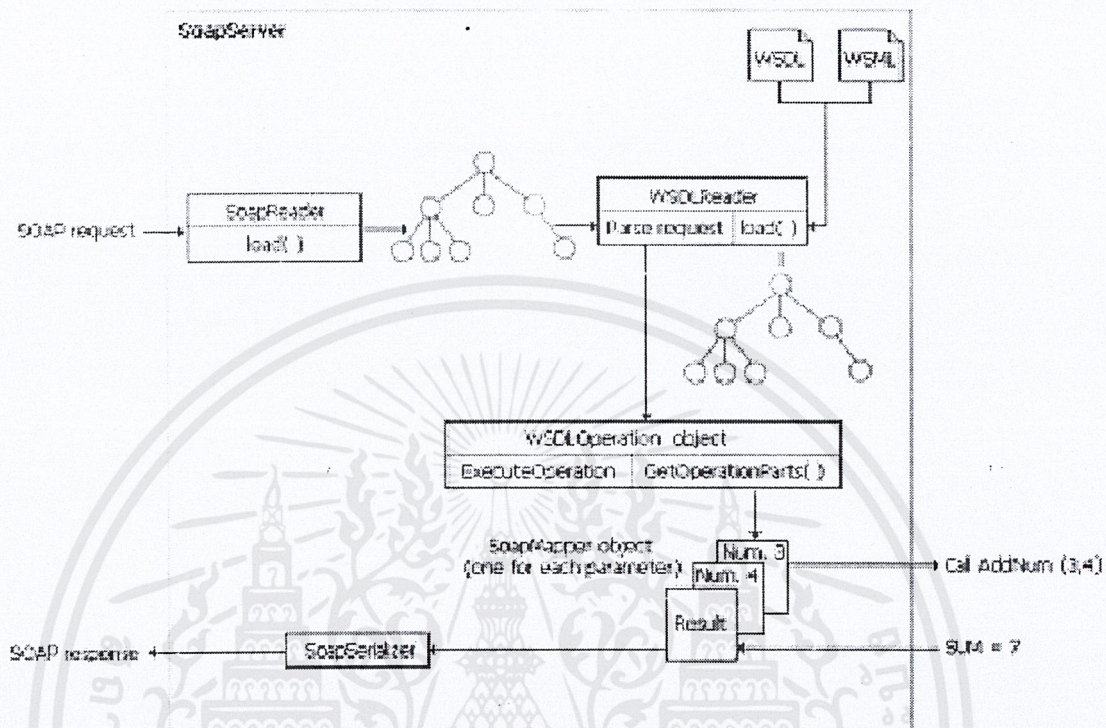
ไม่ว่า SOAP request จะเรียกไปยัง ISAPI หรือ ASP listener ก็ตาม ข้อมูลที่รับส่งก็ยังคงเหมือนกัน เมื่อ SoapServer object ได้รับ SOAP request จากไคลเอนต์ ก็จะทำการแปลและเรียกไปยังเมธอด COM ที่ตรงกับ operation ที่ต้องการ (ตามไดอะแกรมนี้คือเมธอด AddNum) หลังจากนั้นเมธอดที่ถูกเรียกใช้นั้นก็ จะส่งผลลัพธ์กลับมาให้ SoapServer object ซึ่งมันก็จะแปลงผลลัพธ์ให้อยู่ในรูปของ SOAP response แล้วจึงส่งกลับไปให้ไคลเอนต์



รูปที่ 5-4 แสดง Server-Side Data Flow

### 5.8.4 การประมวลผลภายใน SoapServer Object

การทำงานภายใน SoapServer object แสดงได้ดังรูปด้านล่าง



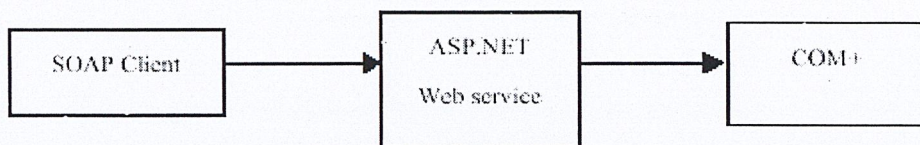
รูปที่ 5-5 แสดงโครงสร้างภายใน SOAP Server

เมื่อได้รับ SOAP request จากไคลเอนต์แล้ว SoapReader object ก็จะโหลด request message ไปไว้ใน DOM structure หนึ่ง ในขณะที่ WSDLReader จะโหลดไฟล์ WSDL และไฟล์ WSML ไปไว้ในอีก DOM structure หนึ่ง ตัว WSDLReader จะแปล request และสร้าง WSDLOperation object ขึ้นเพื่อใช้ใน requested operation หลังจากนั้น WSDLOperation object จะเรียกใช้เมธอด GetOperationParts เพื่อดึงรูปแบบ message ของอินพุตและเอาที่พู่ทของ operation ออกมา แล้ว SoapServer จะสร้าง Soap Mapper objects ให้กับแต่ละส่วนที่ได้จากเมธอด GetOperationParts และโหลดค่าที่ได้มาจากการเรียกใช้ operation เข้าไว้ใน objects เหล่านี้ พอถึงขั้นตอนนี้ SoapServer object จะทำการเรียกไปที่ COM method ที่ตรงกับที่ร้องขอเข้ามาเมื่อ COM method ประมวลผลเสร็จแล้วก็จะส่งผลลัพธ์กลับไป SoapServer object ซึ่ง SoapServer object ก็จะทำการโหลดผลลัพธ์ของ operation ไปเก็บใน SoapMapper object ที่เหมาะสม แล้วจึงใช้ SoapSerializer object ส่ง SOAP response message กลับไปให้ไคลเอนต์แอปพลิเคชันของผู้ใช้

## 5.9 การสร้าง SOAP Client และ SOAP Server ด้วย .NET Framework

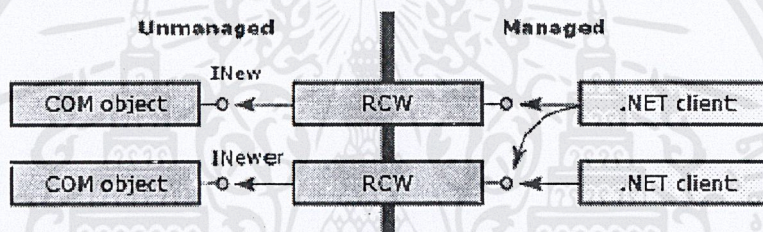
### 5.9.1 การใช้ ASP.NET ทำ SOAP Web Service

ใช้ความสามารถในการทำ COM+ Interop โดยสร้างคลาสของ ASP.NET มาเรียกใช้งาน businesslogic ที่สร้างด้วย VB6



รูปที่ 5-6 การใช้ ASP.NET ทำ SOAP web service

อาศัยความสามารถในการทำ marshall / unmarshall object ให้อยู่ในรูปแบบ SOAP ของ ASP.NET โดย ASP.NET ต้องทำ Interop โดยอาศัย Runtime Callable Wrapper ช่วยจำลอง COM object ให้ .NET มองเห็นเป็น .NET object ธรรมดา



รูปที่ 5-7 Runtime Callable Wrapper

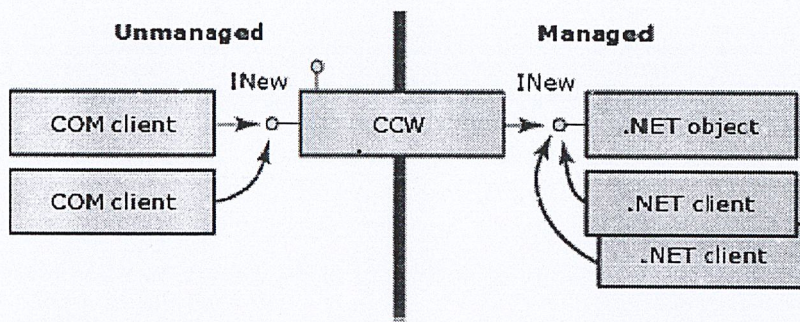
### 5.9.2 การใช้ .NET ทำ SOAP Client

ในทางกลับกันที่ COM ต้องการใช้บริการจาก SOAP ก็กระทำผ่าน .NET web service client



รูปที่ 5-8 การใช้ .NET ทำ SOAP client

แต่การที่จะทำให้ COM ติดต่อกับ .NET class ได้นั้นต้องผ่านกระบวนการ COM Interop โดยการสร้าง COM callable wrapper ซึ่งจะทำให้ COM สามารถเห็น .NET class เป็น COM object ธรรมดา



รูปที่ 5-9 COM callable wrapper ที่ทำให้ COM client สามารถเห็น .NET object



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### เทคโนโลยี ASP.net

#### 6.1 ความเปลี่ยนแปลงที่เกิดขึ้นของ ASP.net

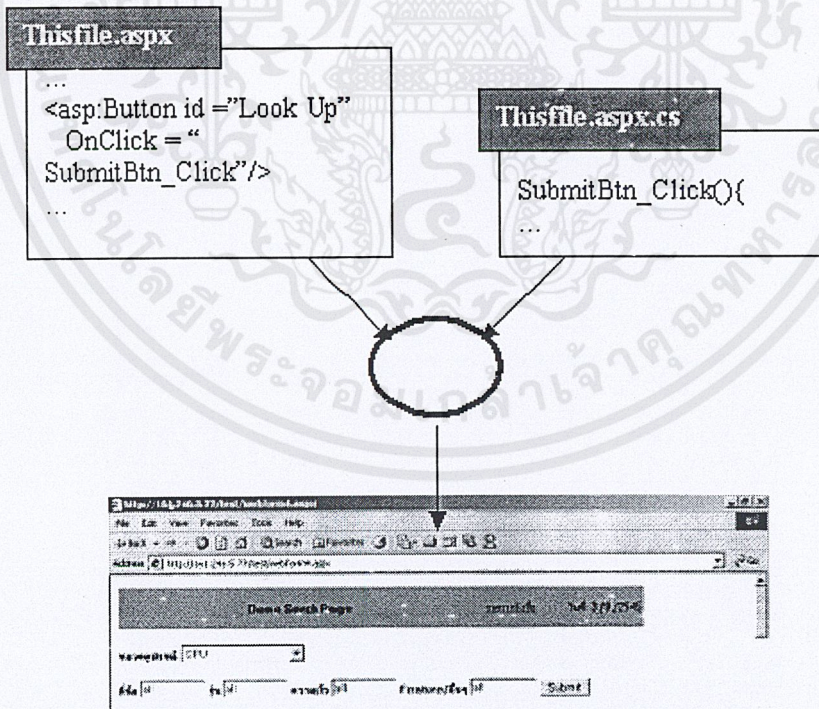
เราสามารถเลือกสร้างแอปพลิเคชัน ASP.net ได้ทันทีจาก Visual Studio.net ซึ่งการสร้างแอปพลิเคชันประเภทนี้ นอกจากจะสนับสนุนการทำงานบนอุปกรณ์ต่างๆ เช่น บราวเซอร์บนเครื่อง พีซี , PDA , Moblie phone แล้ว ตัว ASP.net มีสิ่งที่มีการเปลี่ยนแปลงมากที่สุด

##### 6.1.1 นามสกุลของไฟล์ และการแยกส่วนออกแบบกับส่วนเขียนโค้ด

เริ่มจากนามสกุลของไฟล์ที่สร้าง จะเปลี่ยนจาก .asp เป็น .aspx และ 1 หน้าของเว็บเพจของไฟล์ .aspx ประกอบด้วย 2 ส่วน คือ

1. ไฟล์ที่มีนามสกุล .aspx
2. ไฟล์ที่มีนามสกุลในแบบ Visual c# หรือ Visual Basic

เป็นการแยกระหว่างการออกแบบหน้าเว็บเพจกับการเขียน โปรแกรม ปัจจุบันการเขียนเว็บเพจจะรวมโค้ดทั้งหมดไว้ ไม่ว่าจะเขียนด้วย VBScript หรือ JavaScript และ โครงสร้างของภาษา HTML จะรวมกันที่ไฟล์ ASP อย่างเดียวกัน นี้ทำให้ประสบปัญหาหากแต่ไฟล์ .aspx จะแยกออกมา 2 ส่วนตามที่กล่าวมาแล้ว ทำให้การพัฒนาโปรแกรมเป็นเรื่องง่ายขึ้น



รูปที่ 6-1 แสดงเว็บแอปพลิเคชันที่สร้างด้วย ASP.net จะแยกไฟล์สำหรับออกแบบ(file.aspx) กับไฟล์สำหรับเขียนโค้ดโปรแกรมจากกัน(file.aspx.cs)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.1.2 การคอมไพล์โปรแกรม

ไฟล์ที่สร้างขึ้นจาก ASP.net เมื่อถึงตอนทำงาน คือ เมื่อทำการคอมไพล์แอปพลิเคชัน ทั้ง 2 ไฟล์ จะถูกนำมารวมกัน และคอมไพล์ออกมาเป็น Common Language Runtime จากนั้นเมื่อมีการเรียกใช้งาน ก็จะถูกคอมไพล์เป็น ไบนารีโค้ดและนำไปรันเพื่อแสดงผลได้

ส่วนเว็บเพจที่เป็น ASP แบบเดิม จะมี ASP เป็นตัวแปลภาษา (interpret) คือทุกครั้งที่เว็บเพจจะถูกนำไปแสดงผล ก็จะต้องคอมไพล์ พอผู้ใช้คนที่ 2 เข้ามาเรียกเว็บเพจ ก็จะถูกคอมไพล์อีก ซึ่งต่างจาก ASP.net

โครงสร้างของไฟล์ .aspx ในบรรทัดแรก จะมีส่วนที่เป็นตัวบอกว่าอีกไฟล์หนึ่งเป็นไฟล์เขียนโค้ดโปรแกรม นั้น เป็นภาษาอะไร ไฟล์ชื่ออะไร ซึ่งเป็นการลิงค์ระหว่างไฟล์ Visual Basic หรือ Visual C# ที่ทำงานอยู่เบื้องหลังแอปพลิเคชันนี้

```
<%@Page language="vb"
Codebehind = "events_II.aspx.vb"%>
```

นอกจากนี้ยังมีแท็กคำสั่งพิเศษคือ runat="server" เพื่อบอกให้ทราบว่าการทำงานของโปรแกรม จะอยู่บนเซิร์ฟเวอร์

```
<Form runat="server"> ... </Form>
```

### 6.2 คอนโทรลที่ใช้ออกแบบในแอปพลิเคชัน ASP.net

การเขียนโปรแกรมใน ASP.net จะใช้ภาษาใดก็ได้ใน .net ทั้งภาษาหลักใน Visual Studio.net อย่าง Visual C#, Visual Basic .net หรือภาษาอื่นๆ ที่สนับสนุน .net เช่น ภาษา Pascal ก็สามารถเขียนได้ โดยแอปพลิเคชัน ที่สร้างขึ้นนี้สามารถรันได้บนทุกอุปกรณ์ที่สนับสนุน .net

แอปพลิเคชันที่สร้างด้วย ASP.net นั้น จะเป็นการแยกระหว่างส่วนที่ใช้เขียนโค้ด โปรแกรม กับ ส่วนที่ใช้ออกแบบออกจากกัน โดยฟอร์มที่อยู่ใน ASP.net ก็จะคล้ายๆ กับ Windows Form ซึ่งคอนโทรล 3 ประเภทที่ใช้ในการสร้าง ASP.net มีดังนี้

### 6.2.1 คอนโทรล HTML

คอนโทรล HTML นั้นจะมีโครงสร้างเหมือนเดิมที่ใช้บน ASP แต่เพิ่มบรรทัด Runat = “server” ขึ้นมา ก็บอกแล้วว่าคอนโทรลนี้ใช้กับ ASP.net เช่นเดียวกันหากในฟอร์มนี้ต้องการให้คอนโทรลบางตัวไม่ทำงานในแบบของ ASP.net เพียงแต่เอาคำว่า Runat “server” ออกเท่านั้นเอง

(System.Web.UI.HtmlControls)

```
<input type=button id=button1
```

```
OnServerClick = “Button1_Click” runat=“server”/>
```



รูปที่ 6-2 คอนโทรล HTML ที่เราสามารถเลือกสร้างได้จาก Tool Box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.2.2 คอนโทรล Web Forms

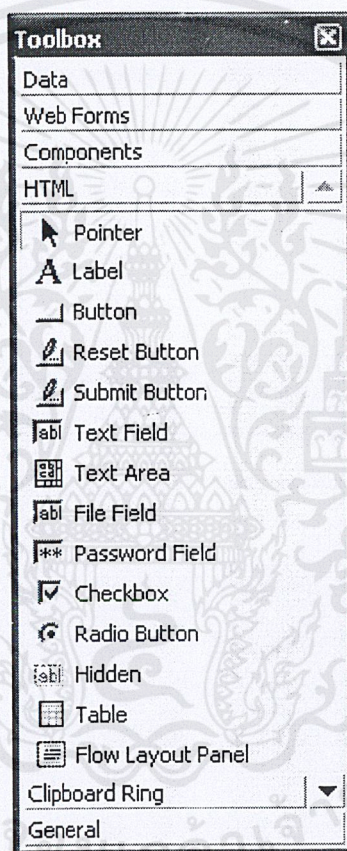
คอนโทรล Web Forms เป็นคอนโทรลพิเศษ ที่มีเฉพาะใน ASP.net คอนโทรลพวกนี้จะมีคำว่า ASP อยู่ข้างหน้าและมีคำว่า Runat “Server”

(System.Web.UI.WebControls)

```
<asp: label id= “ label “ runat =”server”
```

```
backcolor = “red” ForeColor =”white”>
```

```
Hello World </label>
```



รูปที่ 6-3 คอนโทรล Web Forms ที่เราสามารถเลือกสร้างได้จาก Tool Box

### 6.2.3คอนโทรล Customer

คอนโทรลนี้มีนามสกุลเป็น .ascx เหมือน ActiveX control บน Windows Forms นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 คอนโทรลที่ทำงานฝั่งเซิร์ฟเวอร์

ตัวคอนโทรลที่ใช้สำหรับออกแบบบนเว็บเพจที่ทำงานทางฝั่งเซิร์ฟเวอร์นั้น มีกว่า 45 คอนโทรล ตั้งแต่คอนโทรลมาตรฐานอย่าง TextBox, ChecBox จนถึงคอนโทรลพิเศษที่พัฒนาขึ้นมาเพื่อใช้กับ ASP.net

ตัวอย่างโค้ดของคอนโทรลพื้นฐานที่ทำงานบนเซิร์ฟเวอร์

```
<asp:textbox id =text1 runat=server/>
text1.text ="Hello World"

<asp:checkbox id=check1 runat=server/>
check1.checked=True

< asp:button id=button1 runat=server/>
button1_onClick()

<asp:DropDownList id=DropDownList1 runat=server>
DropDownList1.SelectedItem.Text = "Hello"
```

คอนโทรลพื้นฐานมีวิธีการใช้งานเหมือนกับคอนโทรล HTML ทั่วไป เพียงแต่เพิ่มคำว่า ASP และ : (โคล่อน) ตามด้วยชนิดของคอนโทรล กับคำว่า runat = "server" รวมทั้งการอ้างถึงค่าอื่นๆ เช่นข้อความที่ต้องการให้แสดงบนคอนโทรล

ตัวอย่างของคอนโทรลพิเศษเช่น Data Grid ใช้แสดงข้อมูล สามารถรวมเข้ากับ ADO เพื่อนำข้อมูลจากฐานข้อมูลมาแสดง นอกจากนี้มีคอนโทรลประเภท Validation เพื่อตรวจสอบการป้อนข้อมูลของเรา

คอนโทรลประเภท Validation เป็นคอนโทรลที่เราพบบ่อยๆ เช่น การสร้างฟอร์มให้ผู้ใช้กรอกข้อมูล ในปัจจุบันหากเราสร้างคอนโทรลประเภทนี้เอง คงต้องมาทำการเขียนโค้ดเอง แต่สิ่งที่เกิดขึ้นใน ASP.net ก็มีคอนโทรลชื่อ RequiredFieldValidator ให้เราสามารถตรวจสอบได้

```
<asp:RequiredFieldValidator ControlToValidate ="txtName"
ErrorMessage="Please Enter Your Name" runat ="server"/>
```

โดยเราสามารถกำหนดคัพรีออฟเพอร์ดีที่ Error message หากไม่มีการกรอกข้อมูล ให้แสดงข้อความที่อยู่ต่อท้ายคัพรีออฟเพอร์ดีนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 6.4 การเขียนโปรแกรมใน ASP.net

ASP นั้นมีการเขียนโปรแกรมเพื่อคักอิเวนต์ (Event Driven) ด้วย VBScript หรือ JavaScript โดยจะมี Event Driven 4 ตัวดังนี้

- Application\_OnStart เป็นอิเวนต์ที่เกิดขึ้น เมื่อมีการเรียกใช้งานเว็บเพจใดๆ ก็ได้ในเว็บไซด์เป็นเว็บเพจแรก
- Application\_OnEnd เกิดขึ้นเมื่อมีการปิดเครื่องเซิร์ฟเวอร์ หรือทำการ Stop Service ของเว็บเซิร์ฟเวอร์
- Session\_OnStart เกิดขึ้นเมื่อมีการปิดเครื่องเซิร์ฟเวอร์ หรือทำการ Stop Services ของเว็บเซิร์ฟเวอร์
- Session\_OnEnd เกิดขึ้นเมื่อไคลเอ็นต์จบการทำงานบนเว็บไซด์

#### 6.5 สรุปตารางเปรียบเทียบระหว่าง ASP กับ ASP.net

คุณลักษณะ	ASP	ASP.net
นามสกุลของไฟล์	.asp	.aspx , .ascx , .asmx
การทำงานของโปรแกรม	คอมไพล์โค้ดด้วยตัว Interpreter	คอมไพล์โค้ดด้วยตัว JIT Compiler
การออกแบบและการเขียนโปรแกรม	รวมส่วนการเขียนโปรแกรมและส่วนการออกแบบด้วยกัน	แยกส่วนการเขียนโปรแกรมและส่วนการออกแบบออกจากกัน
การจัดการเรื่องสถานะของโปรแกรม	มีการใช้ Cookies ที่ไคลเอ็นต์ไม่สามารถทำงานร่วมกันผ่านแพลตฟอร์มได้	สามารถฝังลง URL ได้เลยโดยไม่ต้องใช้ Cookies สามารถบันทึกข้อมูลที่ใช้ร่วมกันลงในฐานข้อมูล SQL ภายนอกได้
อัปเดตไฟล์ DLL Library	ต้องปิดเว็บเซิร์ฟเวอร์ก่อน	สามารถคัดลอกไฟล์ DLL ลงในไดเรกทอรี bin ได้ทันที (ไม่ต้องมีการลงทะเบียน)

ตารางที่ 6-1 สรุปตารางเปรียบเทียบระหว่าง ASP กับ ASP.net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

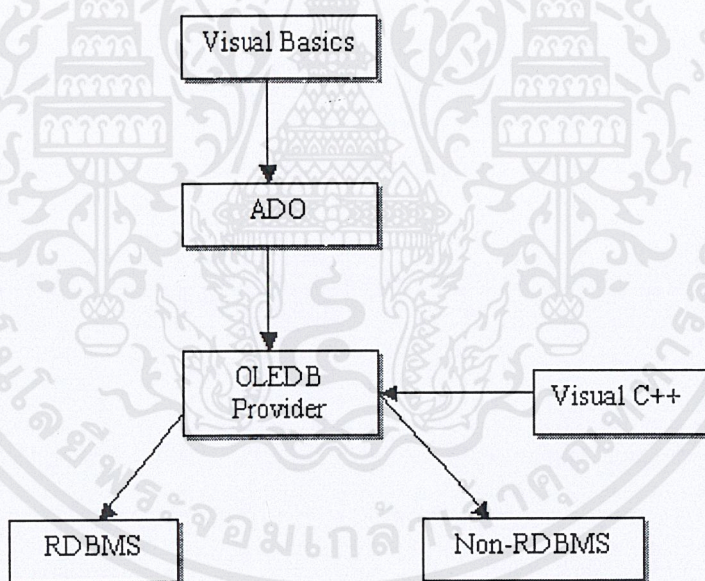
### ADO.net

#### 7.1 รู้จัก ADO.net

ตัว ADO.net เป็นเทคโนโลยีอื่นหนึ่งซึ่งช่วยในการติดต่อกับ ฐานข้อมูลต่างๆ และ ADO.net ยังมีโครงสร้างเหมือน ADO เพียงแต่การออกแบบมีความแตกต่างกันบ้าง

การเก็บข้อมูลที่เก็บใน ADO เมื่อมีการ select ข้อมูลนั้น ผลลัพธ์ที่ออกมาจะเก็บข้อมูลในโครงสร้างของ COM (Common Object Model) แต่ตัว ADO.net เมื่อเรา select ข้อมูลออกมาแล้วต่างๆที่ select ออกมาจะเก็บไว้ในรูปของ XML เพราะฉะนั้น ADO.net จึงเหมาะกับการทำงานที่มีเป็นแอปพลิเคชันบนอินเทอร์เน็ต

ถ้าเป็น ADO ปกติ จะเหมาะกับการทำงานที่เป็นไคลเอนต์ / เซิร์ฟเวอร์ ที่อยู่ในเครือข่าย แลน หรือเครือข่ายอินทราเน็ต(Intranet) ในองค์กรของเรามากกว่า



รูปที่ 7-1 โครงสร้างของ ADO

การเชื่อมต่อฐานข้อมูลบนเครือข่ายภายในองค์กรนั้น เราสามารถกำหนดการเชื่อมต่อได้ คือ เมื่อเปิดแอปพลิเคชัน เราค่อยมาเปิดการเชื่อมต่อฐานข้อมูล ซึ่งเรียกว่าเป็น Connection Oriented แต่เมื่อเอาไปใช้กับการทำงานที่เป็นแอปพลิเคชันบนอินเทอร์เน็ต ซึ่งเป็นระบบที่จำนวนผู้ใช้ไม่แน่นอน และสูงกว่าระบบเครือข่ายในบริษัทแน่นอน เพราะฉะนั้นการที่เรามาเชื่อมต่อกับฐานข้อมูลทั้งไว้จึงดูไม่เหมาะกับการทำงานในเครือข่ายอินเทอร์เน็ตนัก ADO.net จึงมีคุณสมบัติที่สนับสนุน Disconnected Connection ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

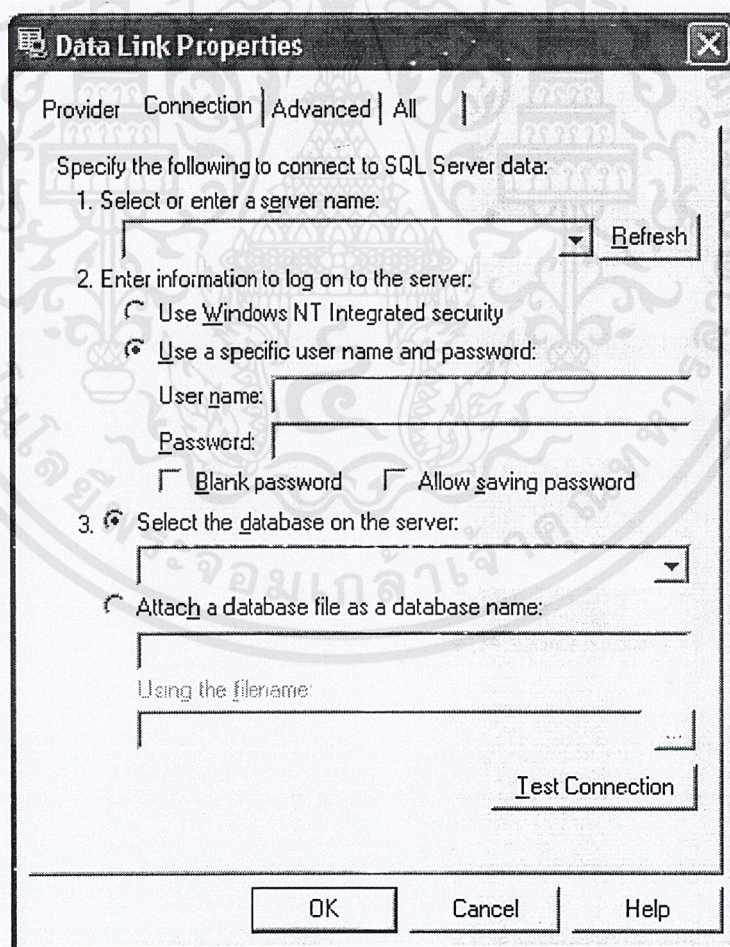
นอกจากนั้นยังสามารถใช้กับ Data Source หลายๆ แบบเหมือนเดิม เช่น Access กับ OLE DB provider ของหลายๆ แบบ หรือกระทั่งข้อมูลใน Outlook Exchange เราก็สามารถใช้ ADO.net ไปselect ออกมาได้เช่นเดียวกัน

## 7.2 ออบเจกต์ที่สำคัญใน ADO.net

ออบเจกต์ที่ใช้บ่อยๆ ใน ADO.net ซึ่งเราจะกล่าวถึงในส่วนนี้ ก็คือ Connection ,Command ,dataset และ DataAdapter

### 7.2.1 ออบเจกต์ Connection

ออบเจกต์ตัวแรกคือ Connection เป็น ออบเจกต์ที่อยู่ใน ADO.net หน้าที่คือ การเปิดการเชื่อมต่อ (Connectuion ) ระหว่างตัวไคลเอนต์ของเรากับ DataSource เราสามารถกำหนดรูปแบบการเชื่อมต่อไปยังฐานข้อมูลที่ต้องการ พร้อมค่ารายละเอียดต่างๆ เช่น รหัสผ่าน เมื่อจะเชื่อมต่อกับฐานข้อมูล ซึ่งยังคงคล้ายกับ ADO แบบเดิมนั่นเอง



รูปที่ 7- 2 Data Link Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง โค้ดการใช้ออบเจกต์ Connection เชื่อมต่อฐานข้อมูล

```
Using System.Data.SqlClient
```

```
SqlConnection Cnn = new SqlConnection();
```

```
Cnn.ConnectionString = " server= localhost ; uid= sa; database =pubs";
```

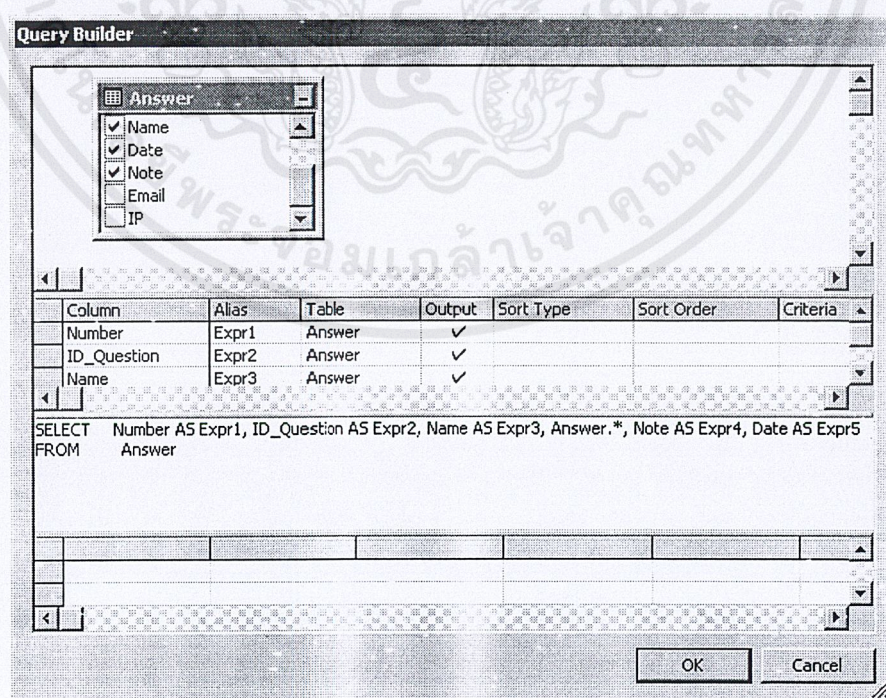
```
Cnn.Open();
```

ตัวอย่างเป็นการเขียนโค้ดด้วย Visual C# บรรทัดแรก คือการเรียกใช้ Name Space ของ System.data.SQL.Client จากนั้นก็สร้าง Instant ของมันขึ้น แล้วกำหนด ConnectionString คือระบุชื่อ เซิร์ฟเวอร์ ชื่อของฐานข้อมูล ชื่อของฐานข้อมูล ชื่อของ User พร้อมกำหนดชนิด ของ OLE DB คือชื่อของ Data Source ซึ่งต้องกำหนดในนี้เหมือนกัน จากนั้น ก็สั่ง open เพื่อเปิดการเชื่อมต่อ

### 7.2.2 ออบเจกต์ Command

ออบเจกต์ ตัวนี้มีหน้าที่เอาไว้เขียนคำสั่ง SQL จากไคลเอ็นต์ไปทำงานบน Data Source เช่นเดียวกับออบเจกต์ Connection คือใน ADO ปกติ ยังมีออบเจกต์นี้อยู่

การใช้งานออบเจกต์ Command ใน ADO.net นี้ก็เหมือนกันกับการทำงานใน ADO เดิม คือเวลาที่เราจะส่งคำสั่ง SQL ที่ไม่ใช่ Select เช่น Insert ,Update หรือ Delect ที่ไม่มีการ Return ค่ากลับมาในส่วนใหญ่ในทุกฐานข้อมูลจะมีโครงสร้างภาษา SQL ให้เราสามารถสร้างฟังก์ชันฝังในตัวฐานข้อมูลเลย ซึ่งส่วนใหญ่ในทุกฐานข้อมูลจะมีโครงสร้างภาษา SQL ให้เราสามารถสร้างฟังก์ชันฝังไว้ในตัวฐานข้อมูลได้



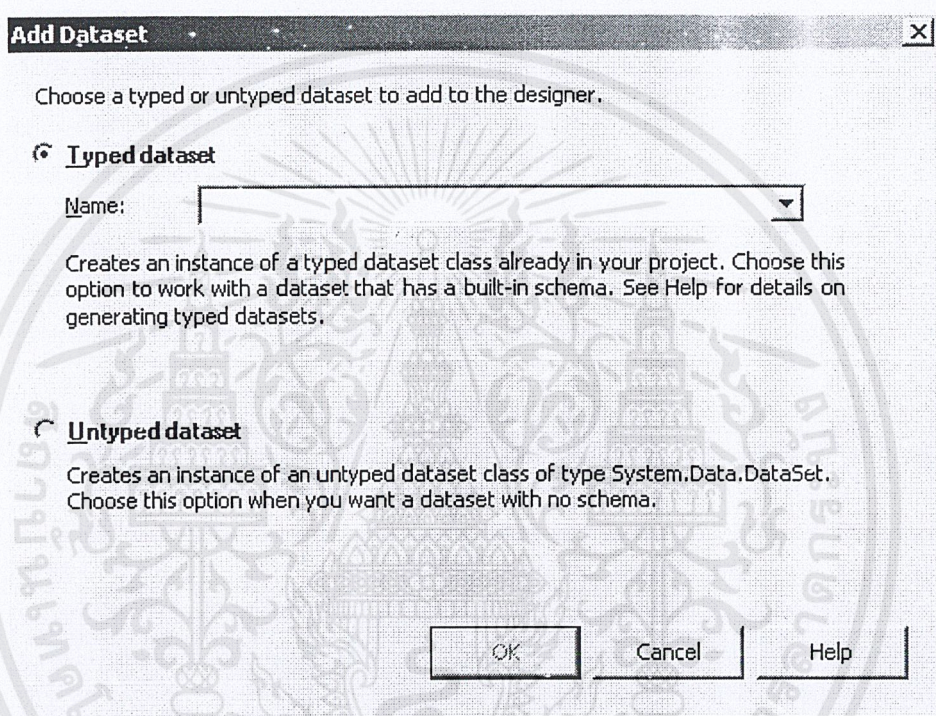
รูปที่ 7-3 Query Builder แสดงการใช้รูปแบบคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 7.2.3 ออบเจกต์ Dataset

ออบเจกต์ Dataset นี้มีเฉพาะใน ADO .net เท่านั้น Dataset เปรียบเสมือนฐานข้อมูลทั้งก้อนเลย คือเมื่อเรา select ออกมา เราไม่ได้เฉพาะแถวของข้อมูลออกมา แต่ได้ข้อมูลอันใหม่อันหนึ่งโดยเป็นฐานข้อมูลที่อยู่ในหน่วยความจำของเรา

ภายใน Dataset นั้นจะมองว่าเป็นเหมือนฐานข้อมูลหนึ่งก้อนในนั้นมีทั้งตาราง (table) คอลัมน์ (columns) และแถว (rows) ต่างๆ อยู่ภายใน มีความสัมพันธ์ระหว่างตารางได้ มีการกำหนด constrain ,index key ,public key ได้ โดยการเก็บข้อมูลของ Dataset จะเก็บทั้ง Schema และ ข้อมูลของตัวมันด้วย



รูปที่ 7-4 Dataset

เราสามารถ select ข้อมูลออกมาเก็บไว้ใน Dataset จากนั้นก็สามารถลอกจากการเชื่อมต่อได้โดยเราก็จะมี Dataset เอาไว้ใช้งาน จากภายในโปรแกรมเราก็สามารถใช้คำสั่ง insert,update,delete ลงไปใน dataset ได้ จนเสร็จเรียบร้อยแล้วเราสามารถกลับมาเชื่อมต่อกับฐานข้อมูลอีกที่หนึ่งแล้วทำการซิงค์ (Synch) ข้อมูลกลับขึ้นไปได้ โดยตัว ADO.net จะคอยจัดการความแตกต่างระหว่างข้อมูลด้วยตัวเอง

การซิงค์ข้อมูลกลับขึ้นไปในลักษณะนี้ คือการเขียนโปรแกรมแบบ Disconnection Programming โดยไม่จำเป็นต้องเชื่อมต่อฐานข้อมูลตลอดเวลา เพราะแอปพลิเคชันบนอินเทอร์เน็ตนั้น ถ้าเราเชื่อมต่อฐานข้อมูลไว้ตลอดเวลา จะเป็นการสิ้นเปลืองทรัพยากรของเซิร์ฟเวอร์มาก

นอกจากนี้ เนื่องจากใน Dataset มีหลายตาราง และแต่ละตารางอาจจะมาจากคนละ Data Source เช่น

1. ตารางแรกมาจากฐานข้อมูล SQL
2. ตารางที่ 2 อาจ Select มาจากฐานข้อมูล Oracle ตาราง Active memory ก็ได้

นี่คือความสามารถของ ADO.net ซึ่ง Dataset จะมองเหมือนฐานข้อมูลทั้งก้อนในหน่วยความจำของเรานั่นเอง

#### 7.2.4 ออบเจกต์ DataAdapter

ออบเจกต์ DataAdapter มีเฉพาะใน ADO.net เช่นกัน ออบเจกต์ DataAdapter ก็เหมาะสำหรับการ Select ทั่วๆ ไป ออบเจกต์ DataAdapter เปรียบเสมือนเป็นสะพานเชื่อมระหว่าง DataSource ที่มีข้อมูลกับตัว Dataset ของเรา โดยลำเลียงข้อมูลจาก Data Source เอามาเก็บใน DataSet จากนั้นจึงสามารถใช้คำสั่ง SQL ใน Dataset ทำการเปลี่ยนแปลงข้อมูลในแถวต่างๆ ได้ และหากต้องการซิงค์ข้อมูลจาก Dataset กลับไปใน Data Source ใหม่ เราก็ใช้ DataAdapter ทำการอัปเดตข้อมูล กลับขึ้นไปใน Data Source

DataAdapter มีเมธอด อยู่ 2 ตัว คือ fill กับ update

- Fill : ใช้อ่านข้อมูลจาก Data Source กลับมาไว้ใน Dataset
- Update: ใช้ดึงข้อมูลกลับ

หากเปรียบเทียบของ DataAdapter ก็เหมือนสะพานเชื่อมโยงระหว่างฐานข้อมูลของเรากับ Dataset นั่นเอง

ตัวอย่างการสร้าง DataAdapter ในภาษา visual C#

```
Dataset pubs;
SQLDataAdapter objDataAdapter = new SQLDataAdapter("Select * from authors ",cnn);

ObjDataAdapter.Fill(pubs,"authors");
Pubs.Tables["Authors"].Rows[0]["au_lname"]='smith';
ObjDataAdapter.update(pubs,"authors");
```

จากตัวอย่าง การที่จะเอาข้อมูลมาเก็บไว้ใน Dataset เราต้องสร้างออบเจกต์ขึ้นมา 2 ตัว ตัวแรกเป็น Dataset ชื่อ pubs ตัวที่ 2 เป็น Dataadapter หลังสร้างเสร็จแล้ว ให้ใช้คำสั่ง SQL เพื่อดึงข้อมูลมา จากนั้นใช้คำสั่ง Fill เพื่อนำข้อมูลที่ได้อ่านเข้ามาใส่ใน Dataset

เวลาที่เราอ้างถึงแถวหรือคอลัมน์ต่างๆ นั้น ต้องทำตามรูปแบบโครงสร้างดังนี้

ชื่อ Dataset ตามด้วย Tables แล้วก็ใช้ชื่อของตาราง ตามด้วย Rows และอินเด็กซ์ของแถวที่ต้องการ และตามด้วยชื่อของคอลัมน์ เช่น

```
Pubs.Tables["Authors"].Rows[0]["au_lanme"]
```

จากตัวอย่างเป็นการอ้างถึง Dataset ชื่อ pubs ที่ตาราง Athors ในแนวแรกของตาราง และคอลัมน์ชื่อว่า au\_lanme

### 7.3 การสนับสนุนมาตรฐาน XML

การสนับสนุนมาตรฐาน XML เป็นคุณลักษณะหนึ่งของ ADO.net ที่เพิ่มขึ้นมาจาก ADO ซึ่ง XML เป็นเทคโนโลยีพื้นฐานอยู่เบื้องหลังไมโครซอฟท์อย่างมาก เพราะ Web Service ก็ใช้ XML รับส่งข้อมูล มีฟังก์ชันอยู่ 4 ตัว ที่เกี่ยวกับ XML คือ

- Read XML ()
- Read XML Schema ()
- Write XML ()
- Write XML Schema ()

เพื่อใช้ในการอ่านข้อมูลไฟล์ XML มาเก็บใน Dataset โดยรวมกับ Schema ไว้ใน Dataset หรือว่าจะเขียนลงเป็น XML ใน DataSEt ก็ได้

### 7.4 สรุปตารางเปรียบเทียบ ระหว่าง ADO กับ ADO.net

คุณลักษณะ	ADO	ADO.net
สถาปัตยกรรมที่ใช้	บนพื้นฐาน COM การเชื่อมโยงแบบ connection orientd	XML กับ HTTP การเชื่อมโยงแบบ Connectionless
การเข้าถึงระบบป้องกัน Firewall	ไม่ผ่านระบบ Firewall	ใช้ HTTP และสามารถ ผ่าน Firewall ได้เป็นอย่างดี
ออบเจ็กต์ที่ใช้	Connection Command Recordset	Connection Command Dataset DataSetview DataView เป็นต้น

ตารางที่ 7-1 สรุปตารางเปรียบเทียบ ระหว่าง ADO กับ ADO.net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

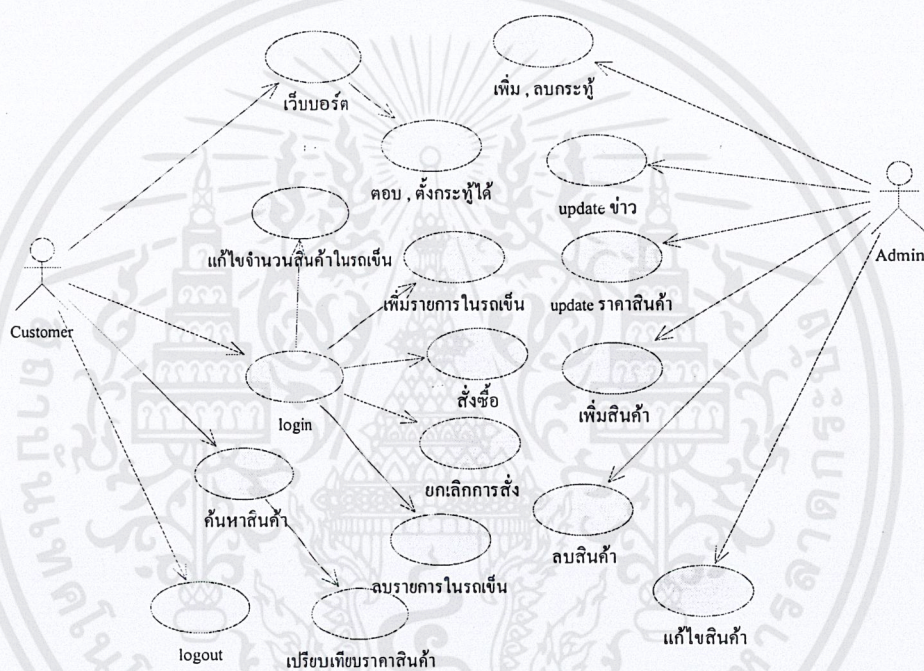
## บทที่ 8

### การออกแบบ

การออกแบบแบ่งออกเป็น 2 ส่วนใหญ่ๆคือ

- 1 ระบบร้านค้า
- 2 ระบบพาสพอร์ด

#### 8.1 ระบบร้านค้าออนไลน์(Use case diagram)



รูปที่ 8-1 แสดง Use Case Diagram ระบบร้านค้า online

**Login** : การขอเข้าใช้บริการเพื่อการเพิ่ม ลบ แก้ไขสินค้าในรถเข็นซึ่งจะต้องเป็นสมาชิกเท่านั้นจะมีสิทธิ

**Logout** : การสิ้นสุดการขอเข้าใช้งานของทางระบบพาสพอร์ด

**ค้นหาสินค้า** : คือการสืบสินค้าตามหมวดของสินค้าที่ทางระบบได้กำหนดไว้

**เว็บบอร์ด** : คือลูกค้าสามารถเข้ามาใช้บริการกระดานถาม-ตอบของทางระบบได้

**เปรียบเทียบราคาสินค้า** : คือ การแสดงการเปรียบเทียบราคาสินค้าที่หน้าเว็บไซต์

**สั่งซื้อ** : คือการตัดสินใจในการสั่งซื้อโดยจะบันทึกเป็นรายการสั่งซื้อ

**ยกเลิกการสั่งซื้อ** : เป็นการตัดสินใจยกเลิกการสั่งซื้อทุกรายการหรือบางรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เพิ่มรายการสินค้าในรถเข็น :** คือการนำรายการสินค้าบันทึกลงในฐานข้อมูลที่เก็บข้อมูลส่วนตัวของลูกค้าย่อในรถเข็น

**ลบรายการสินค้าในรถเข็น :** คือการนำรายการสินค้าที่ถูกบันทึกแล้วลบออกจากฐานข้อมูลของลูกค้าย่อในรถเข็น

**แก้ไขรายการสินค้าในรถเข็น :** คือการนำรายการสินค้าที่ถูกบันทึกแล้วนำมาแก้ไขปรับปรุงจำนวนสินค้าที่จะตั้งชื่อจากฐานข้อมูลของลูกค้าย่อในรถเข็น

**update ข่าว :** เพิ่มเติมข่าวของรายการสินค้าใหม่ในฐานข้อมูลของระบบ

**update รายการสินค้า :** เพิ่มเติมรายการสินค้าใหม่จะเก็บในฐานข้อมูลของระบบ

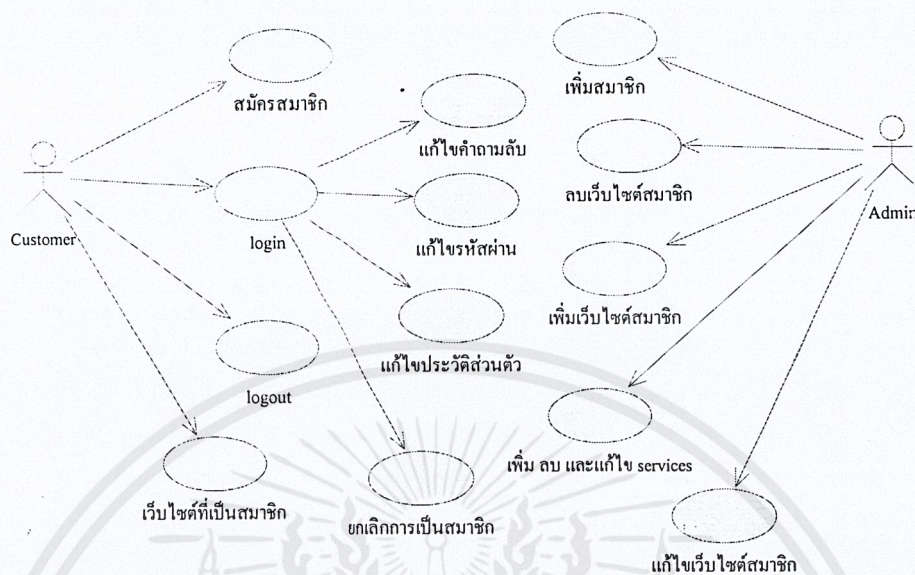
**เพิ่มสินค้า :** ผู้ดูแลระบบสามารถจะเพิ่มสินค้าในระบบได้

**ลบสินค้า :** ผู้ดูแลระบบสามารถจะลบสินค้าในระบบได้

**แก้ไขสินค้า :** ผู้ดูแลระบบสามารถจะแก้ไขสินค้าในระบบได้



## 8.2 ระบบพาสพอร์ท(Use case diagram)



รูปที่ 8-2 แสดง Use case diagram ระบบ passport

**Login** : การขอเข้าใช้บริการเกี่ยวกับการเพิ่ม ลบ แก้ไขสินค้าในรูดเงินซึ่งจะต้องเป็นสมาชิกเท่านั้นจะมีสิทธิ

**Logout** : การสิ้นสุดการขอเข้าใช้งานของทางระบบพาสพอร์ท

**เว็บไซต์ที่เป็นสมาชิก** : คือผู้เข้ามาใช้งานในระบบทั้งเป็นสมาชิกหรือไม่เป็นสมาชิกสามารถรับรู้เว็บไซต์ต่างๆที่เป็นสมาชิกกับทางระบบพาสพอร์ทของเรา

**สมัครสมาชิก** : เป็นการดำเนินการขั้นตอนแรกก่อนจะเข้าใช้ในระบบ และเข้าใช้เว็บไซต์ที่เป็นสมาชิกของทางระบบพาสพอร์ท

**แก้ไขคำถามลับ** : เป็นขั้นตอนในการที่สมาชิกมีความต้องการที่จะแก้ไขและเปลี่ยนแปลงคำถามลับโดยต้องทำการ authentication กับระบบก่อน

**แก้ไขรหัสผ่าน** : เป็นขั้นตอนในการที่สมาชิกมีความต้องการที่จะแก้ไขและเปลี่ยนแปลงรหัสผ่านโดยต้องทำการ authentication กับระบบก่อน

**แก้ไขประวัติส่วนตัว** : เป็นขั้นตอนในการที่สมาชิกมีความต้องการที่จะแก้ไขและเปลี่ยนแปลงประวัติส่วนตัวโดยต้องทำการ authentication กับระบบก่อน

**ยกเลิกการเป็นสมาชิก** : เป็นขั้นตอนในการที่สมาชิกมีความต้องการที่จะยกเลิกการเป็นสมาชิกโดยต้องทำการ authentication กับระบบก่อน

**เพิ่มสมาชิก** : ผู้ดูแลระบบสามารถที่จะเพิ่มสมาชิกเข้าไปกับระบบได้

เพิ่มเว็บไซต์สมาชิก : เป็นขั้นตอนในการที่ผู้ดูแลระบบมีความต้องการที่เพิ่มเว็บไซต์สมาชิกลงในระบบฐานข้อมูลของทางระบบ

ลบเว็บไซต์สมาชิก : เป็นขั้นตอนในการที่ผู้ดูแลระบบมีความต้องการที่ลบเว็บไซต์สมาชิกลงในระบบฐานข้อมูลของทางระบบ

แก้ไขเว็บไซต์สมาชิก : เป็นขั้นตอนในการที่ผู้ดูแลระบบมีความต้องการที่แก้ไขเว็บไซต์สมาชิกลงในระบบฐานข้อมูลของทางระบบ

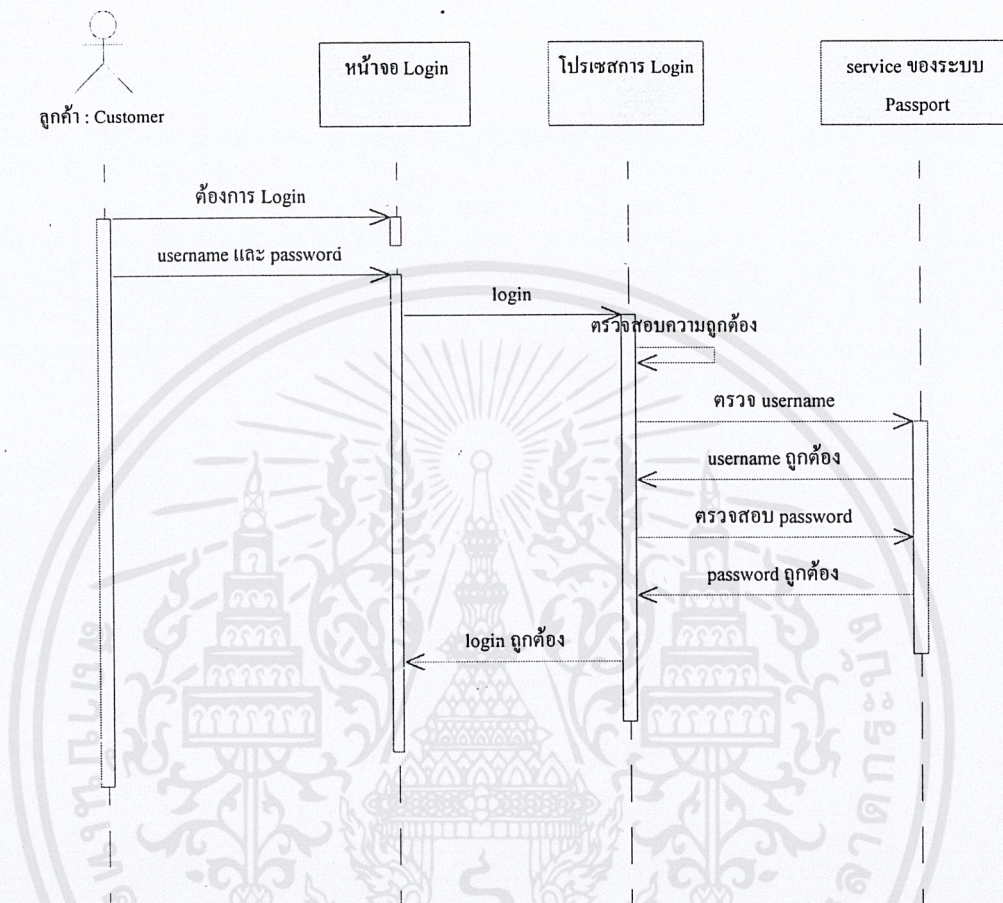
เพิ่ม ลบ และแก้ไข service : เป็นขั้นตอนในการที่ผู้ดูแลระบบมีความต้องการที่จะเพิ่ม หรือ ลบ หรือ แก้ไข service ต่างที่ทางระบบ เปิดให้ใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 8.3 Sequence diagram

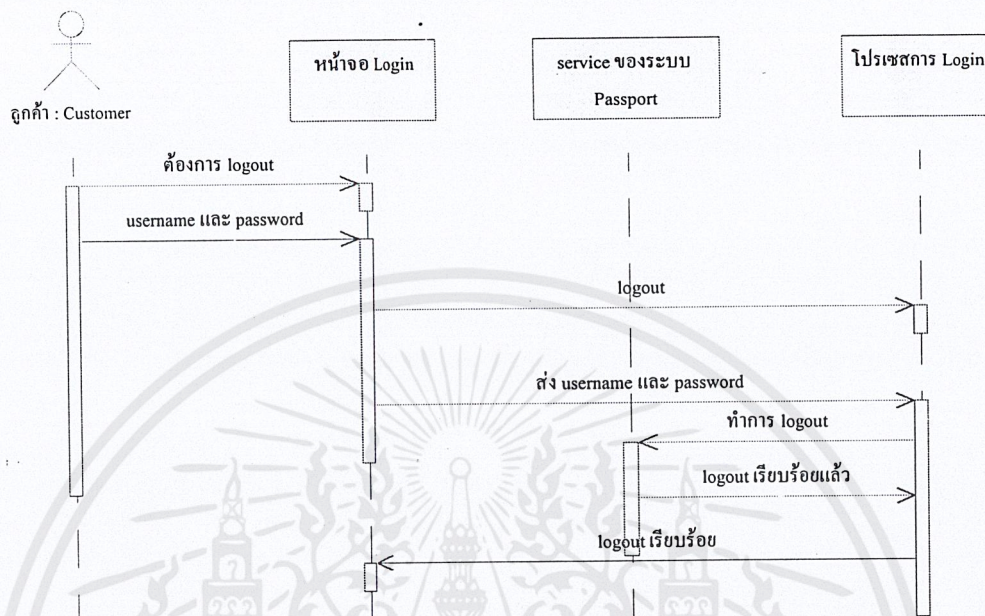
#### Login (Sequence diagram)



รูปที่ 8-3 แสดง Sequence diagram ของ Login

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

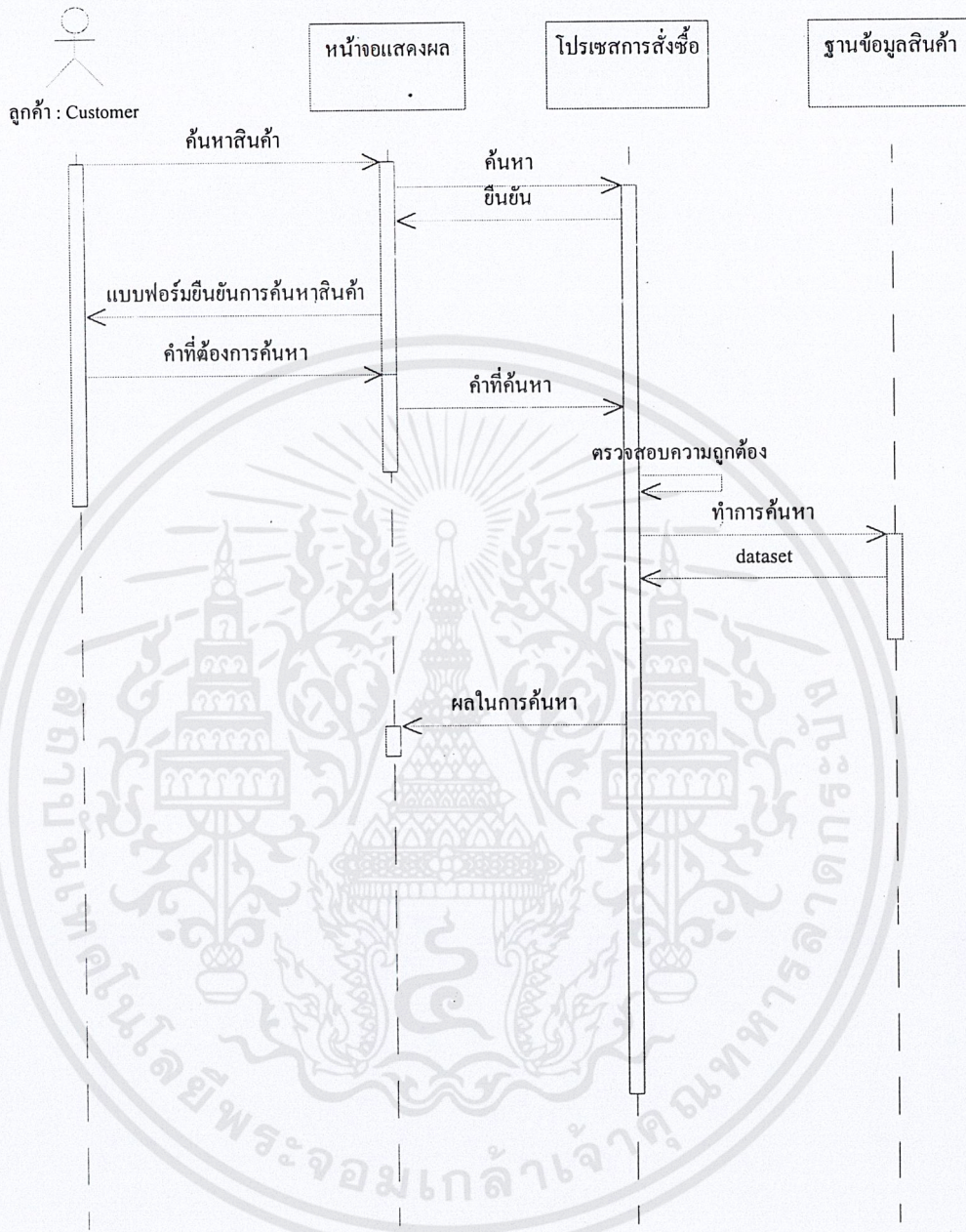
## Logout (Sequence diagram)



รูปที่ 8-4 แสดง Sequence diagram ของ Logout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

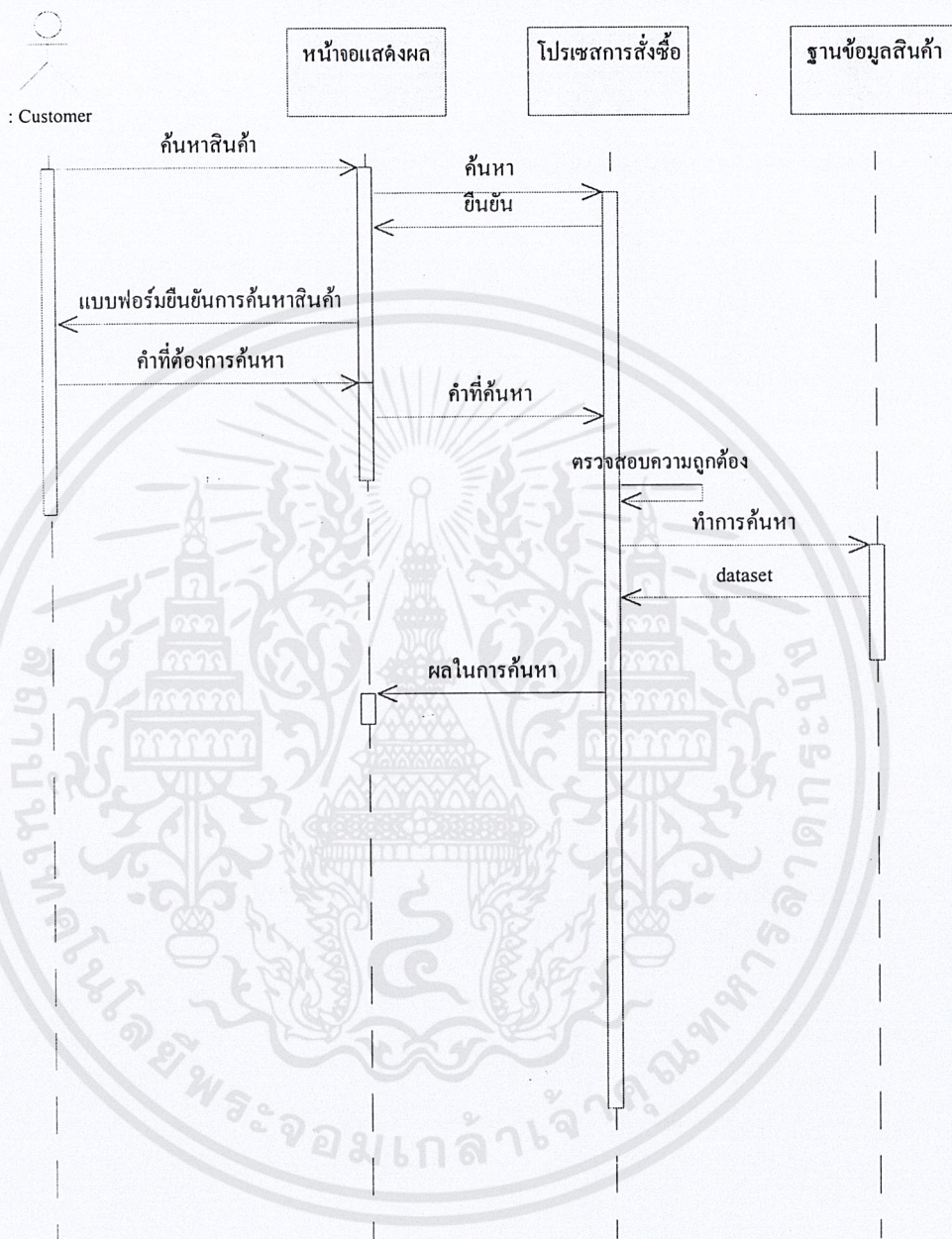
### การค้นหาสินค้า (Sequence diagram)



รูปที่ 8-5 แสดง Sequence diagram ของ การค้นหาสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

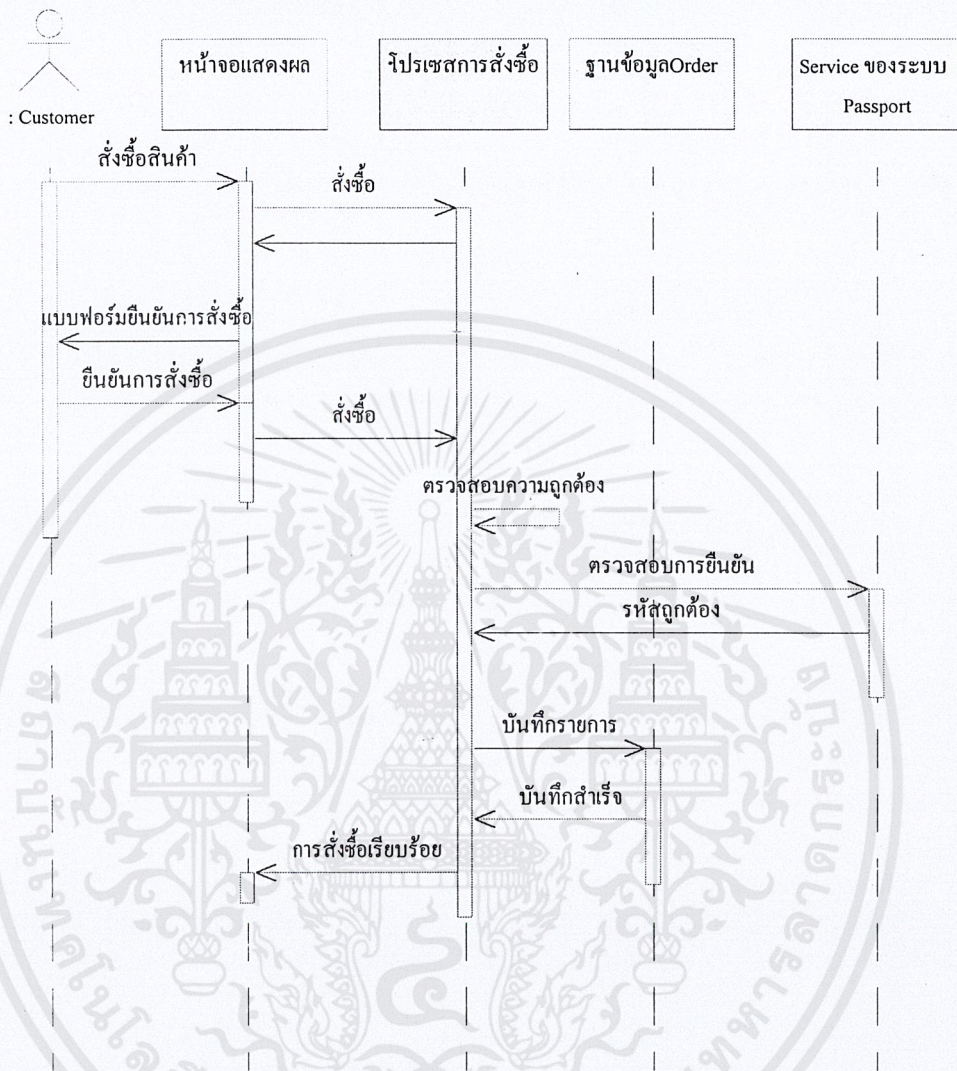
## การเปรียบเทียบสินค้า (Sequence diagram)



รูปที่ 8-6 แสดง Sequence diagram ของ การเปรียบเทียบสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

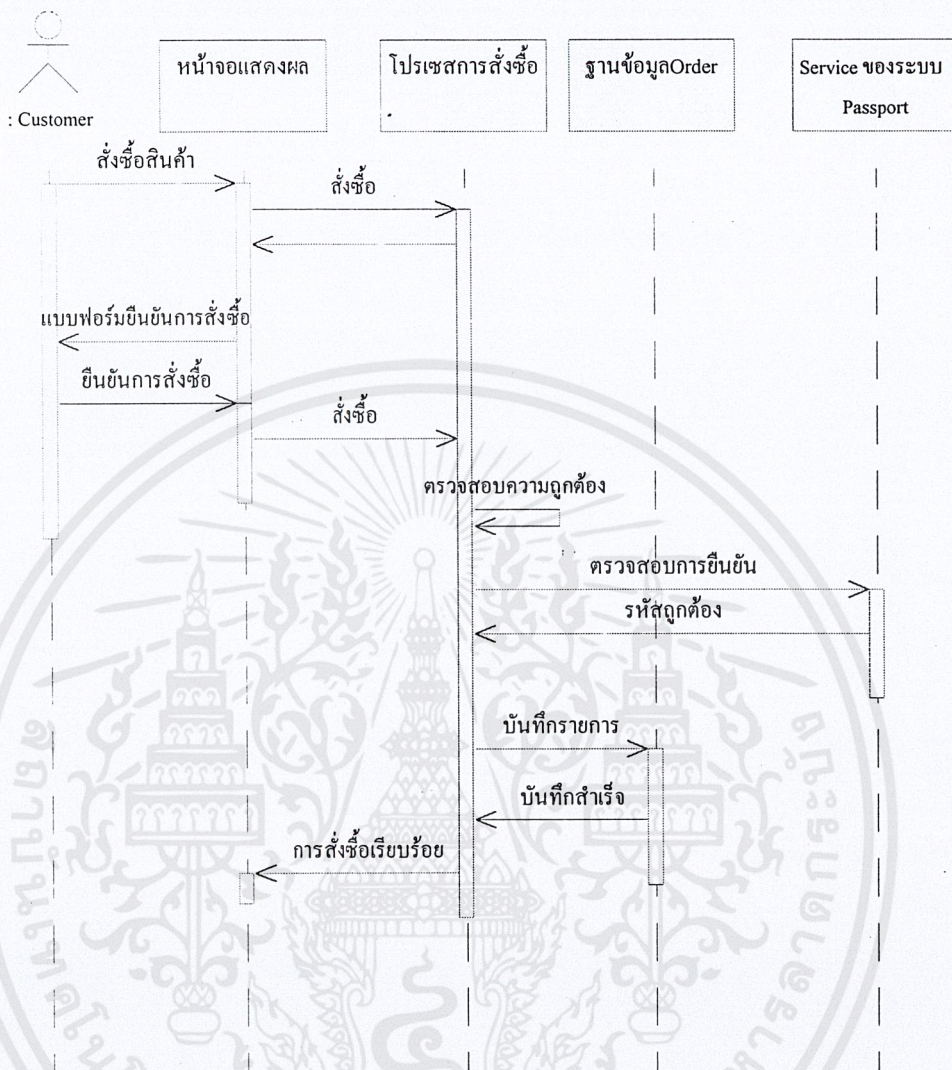
### การสั่งซื้อสินค้า (Sequence diagram)



รูปที่ 8-7 แสดง Sequence diagram ของ การสั่งซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

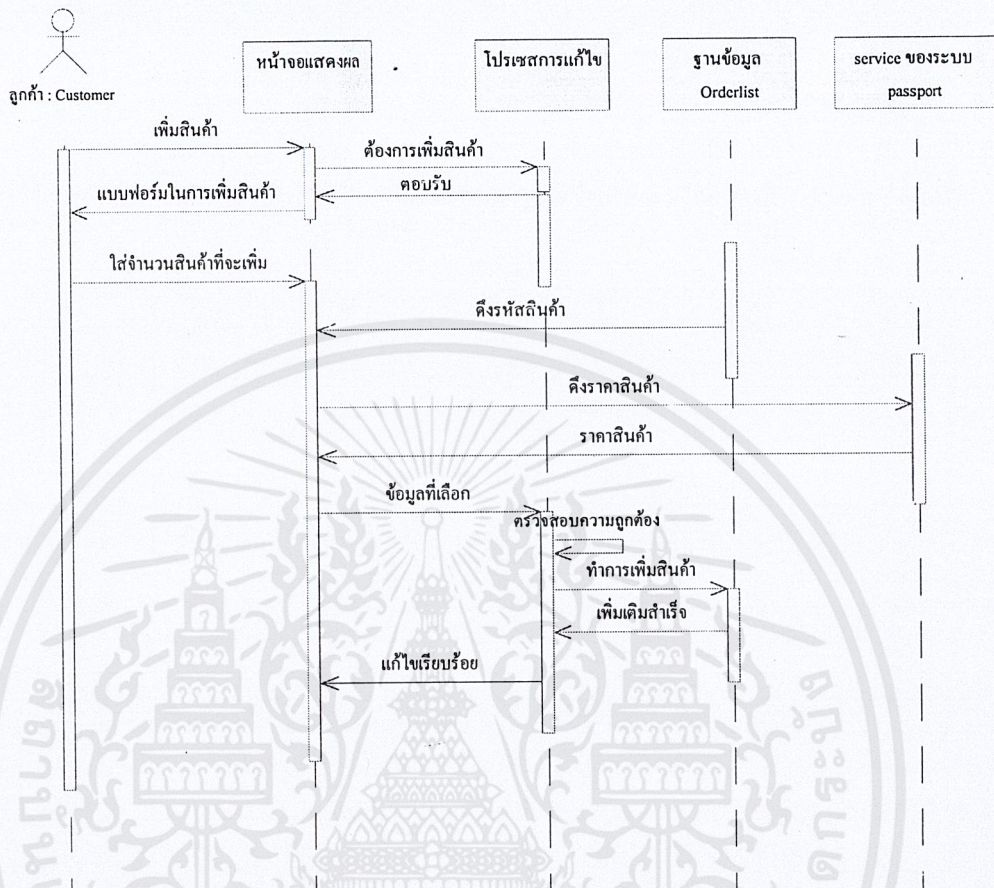
## การสั่งซื้อสินค้า (Sequence diagram)



รูปที่ 8-8 แสดง Sequence diagram ของ การสั่งซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

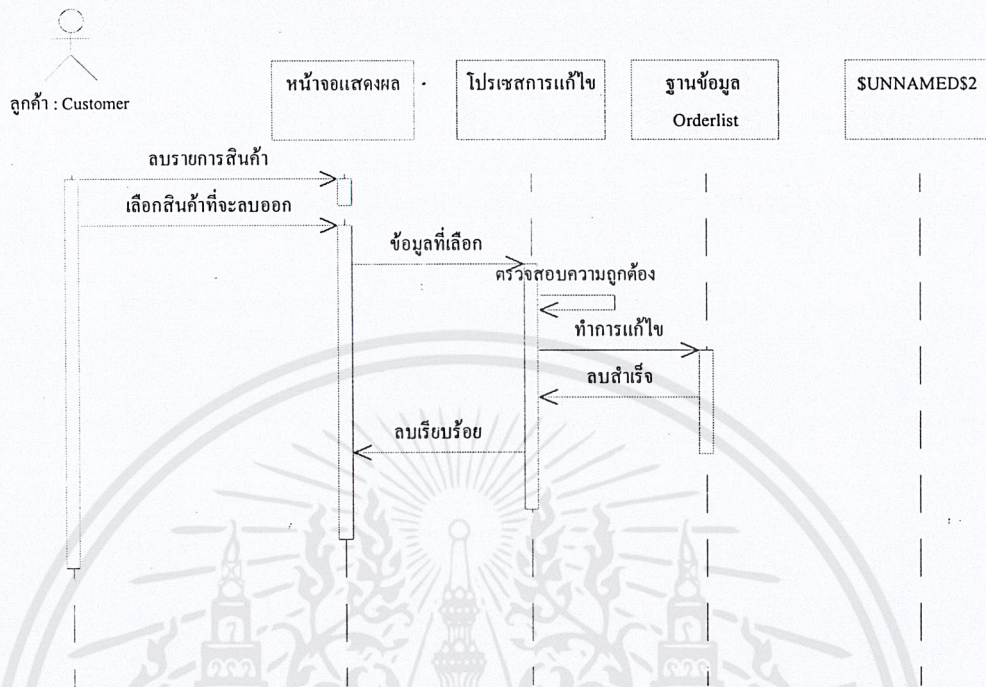
การเพิ่มรายการสินค้า (Sequence diagram)



รูปที่ 8-9 แสดง Sequence diagram ของ การเพิ่มรายการสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

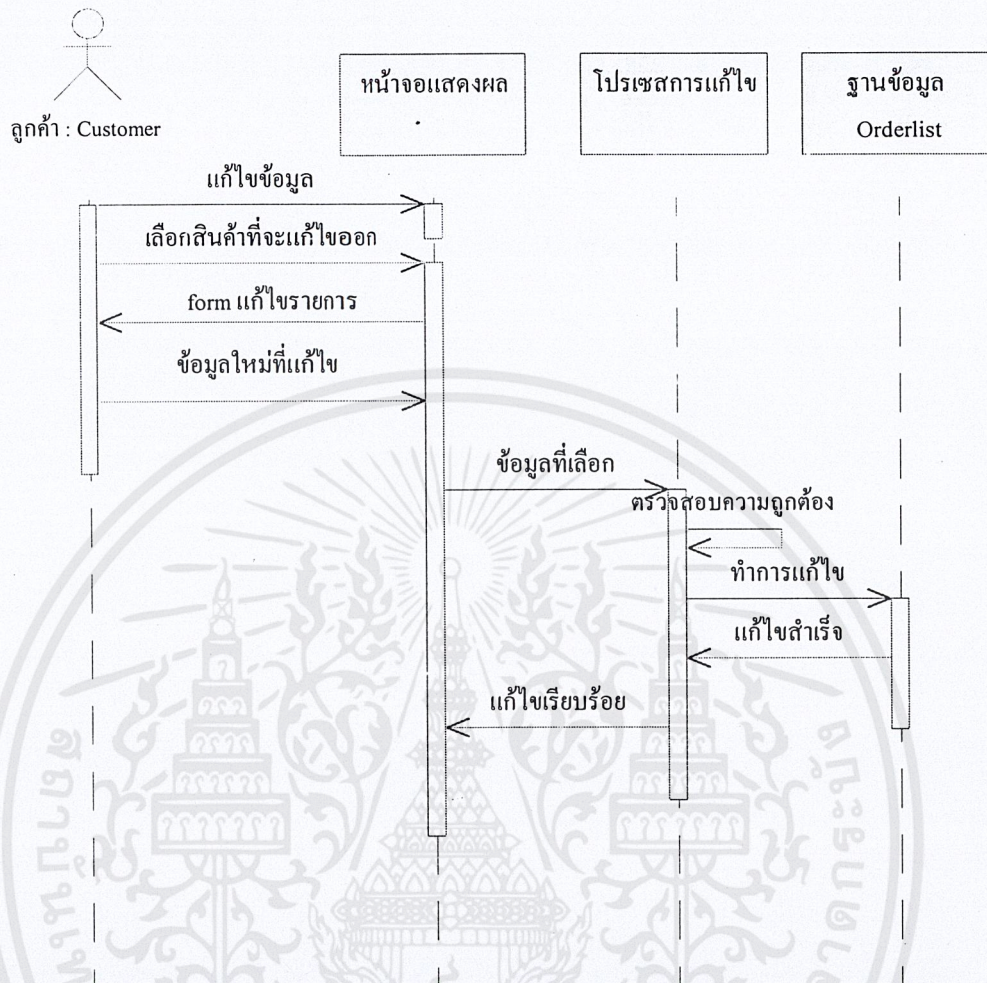
## การลบบรรายการสินค้า (Sequence diagram)



รูปที่ 8-10 แสดง Sequence diagram ของ การลบบรรายการสินค้า

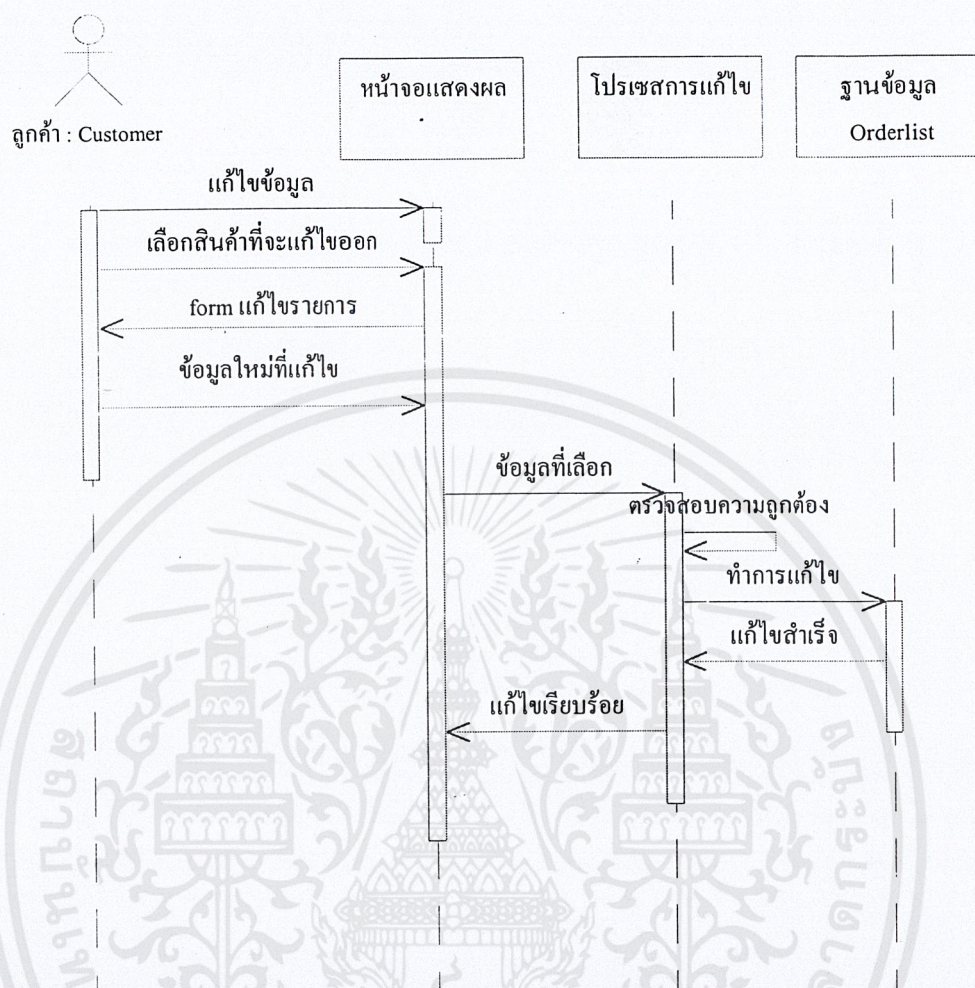
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การแก้ไขรายการสินค้า (Sequence diagram)



รูปที่ 8-11 แสดง Sequence diagram ของ การแก้ไขรายการสินค้า

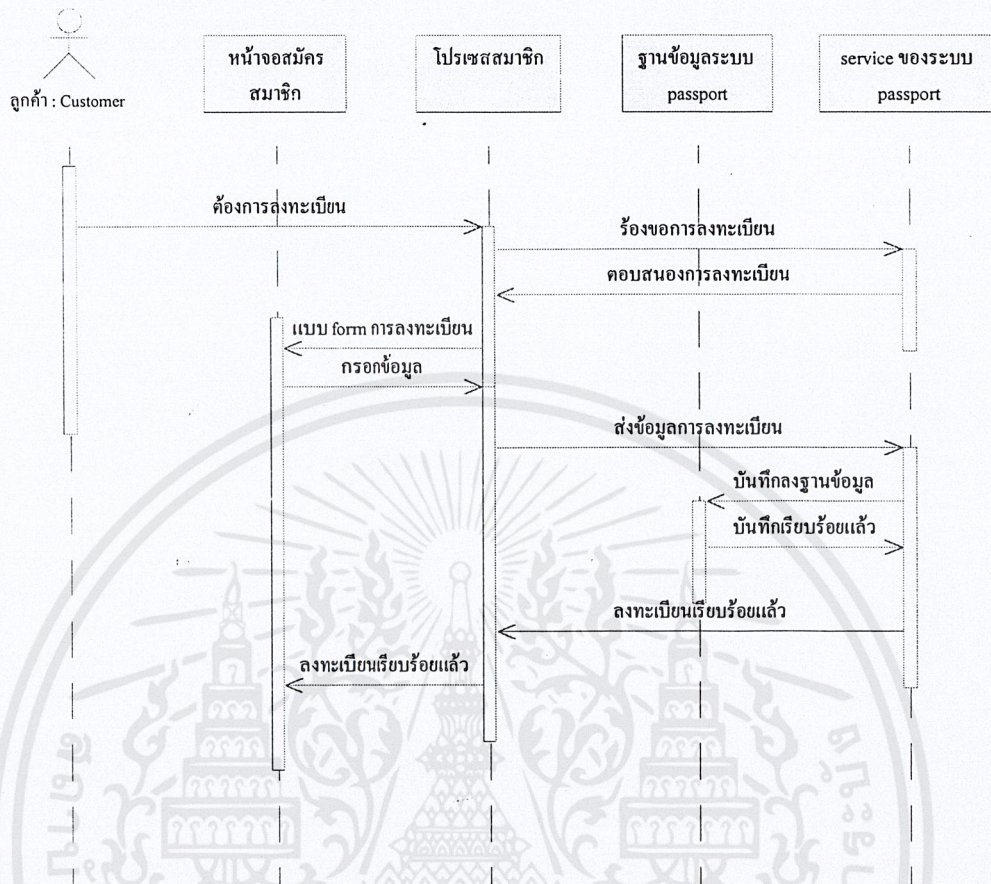
## การแก้ไขรายการสินค้า (Sequence diagram)



รูปที่ 8-12 แสดง Sequence diagram ของ การแก้ไขรายการสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

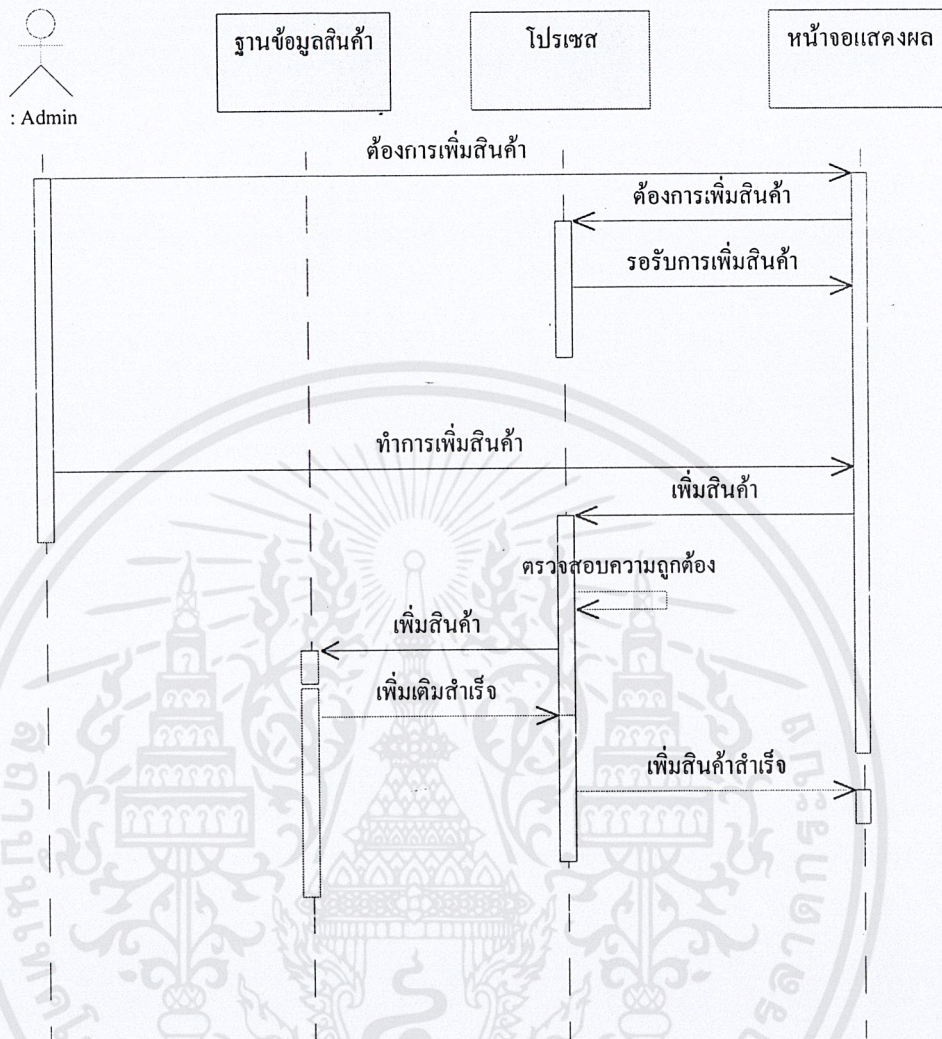
### การสมัครสมาชิก (Sequence diagram)



รูปที่ 8-13 แสดง Sequence diagram ของ การสมัครสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

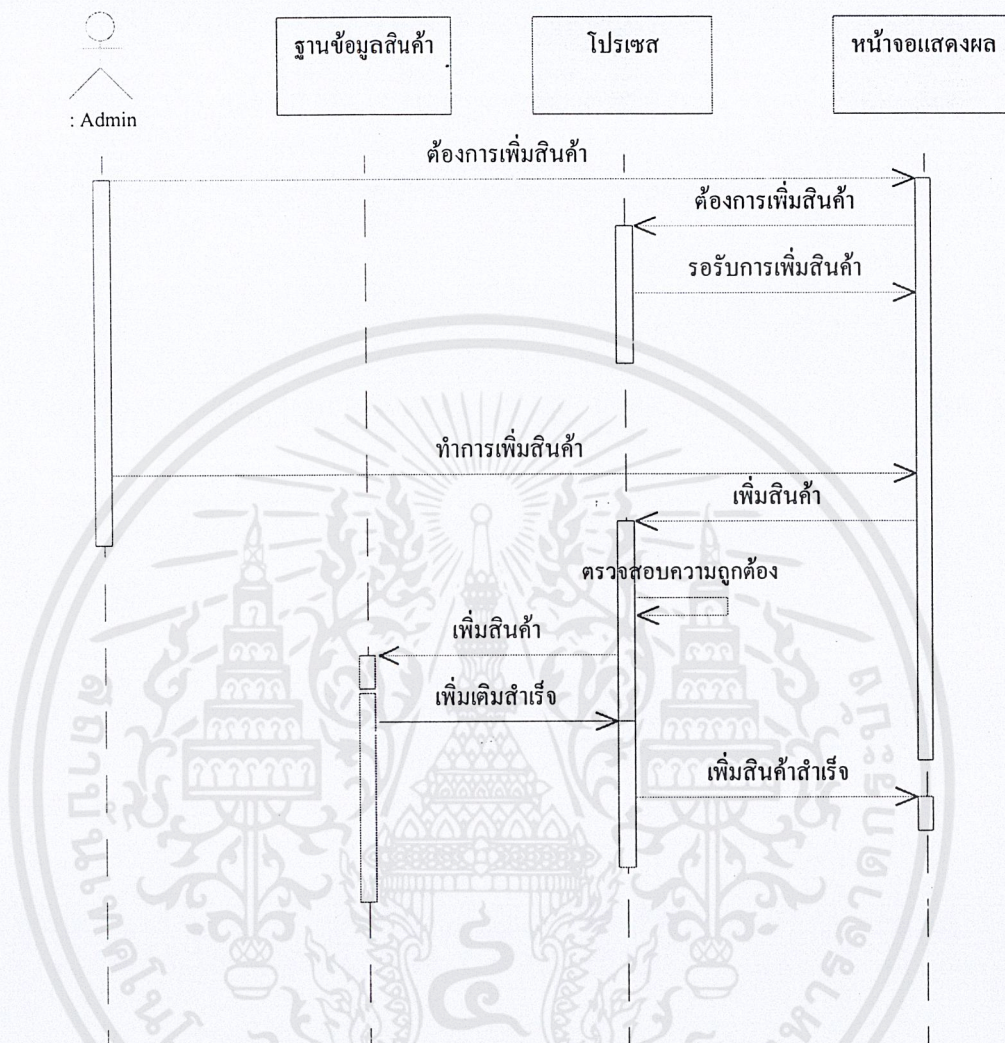
### การเพิ่มเติมสินค้า (Sequence diagram)



รูปที่ 8-14 แสดง Sequence diagram ของ การเพิ่มเติมสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

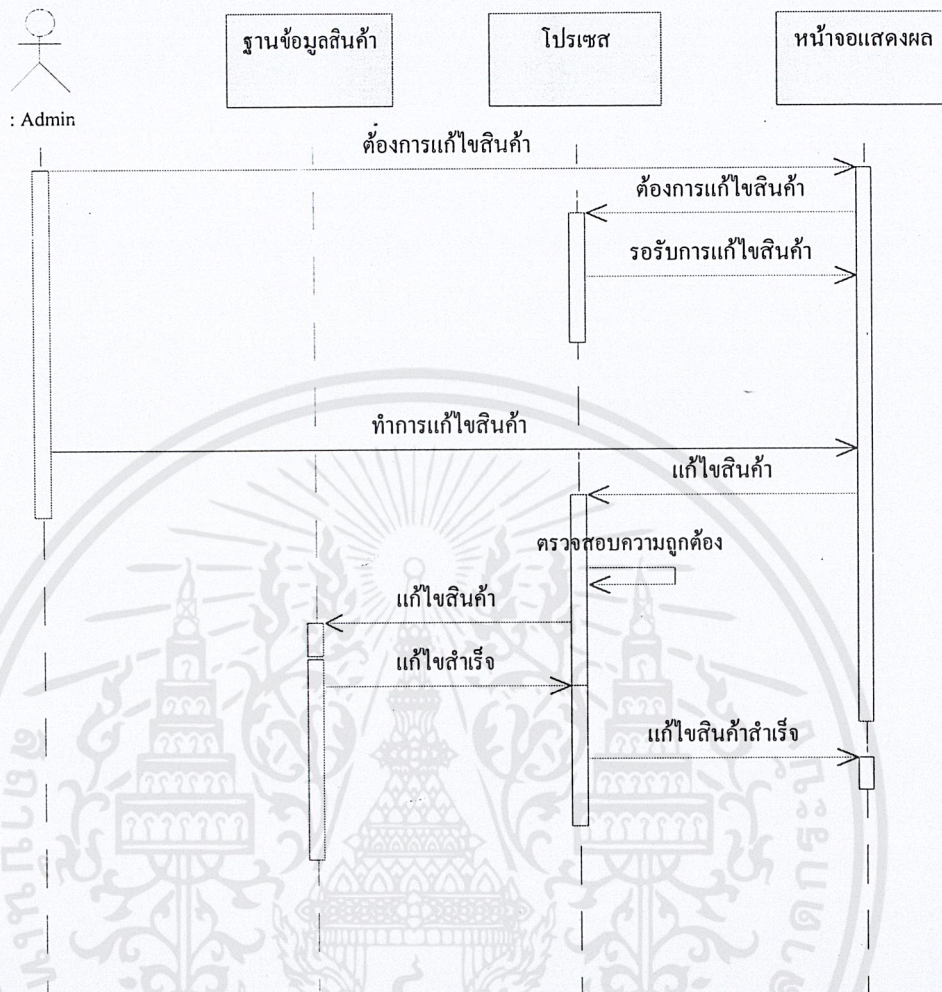
## การลบสินค้า (Sequence diagram)



รูปที่ 8-15 แสดง Sequence diagram ของ การลบสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

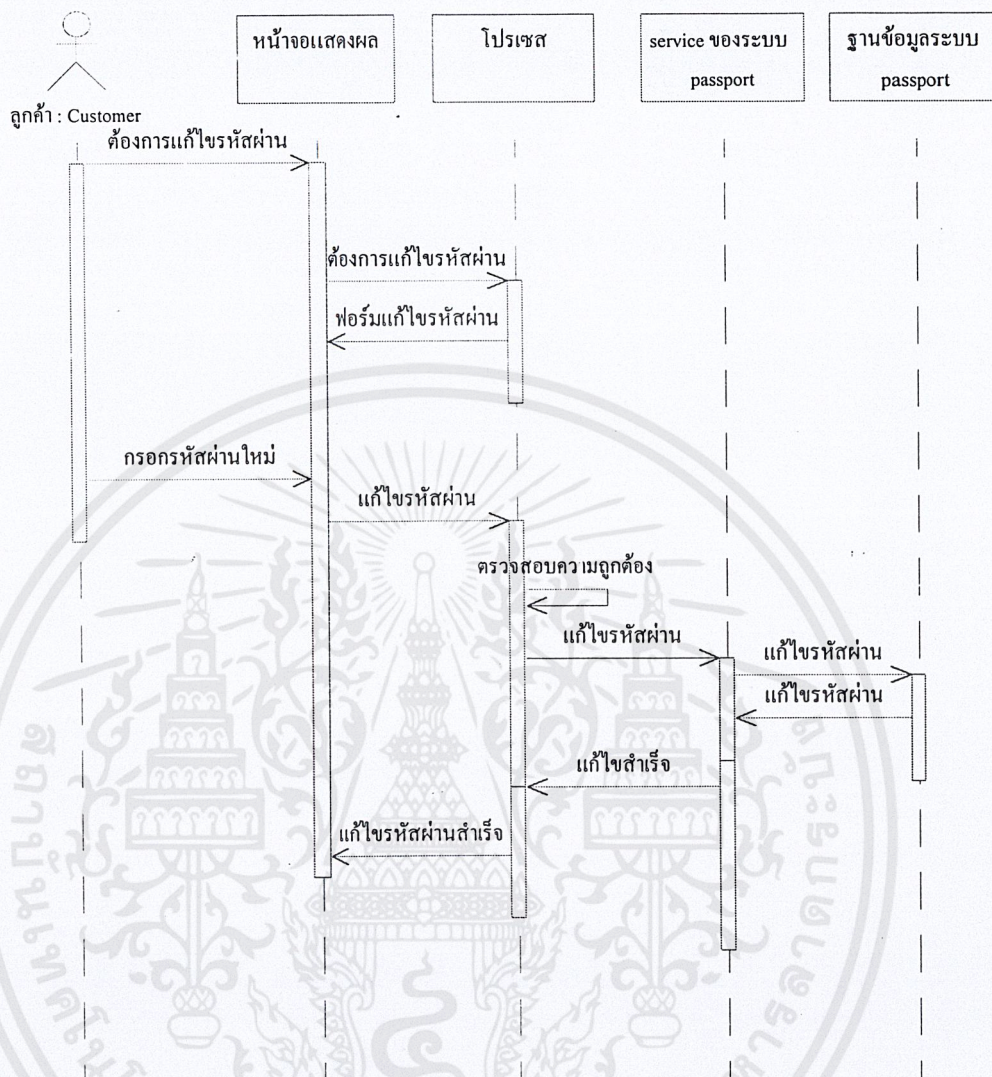
### การแก้ไขสินค้า (Sequence diagram)



รูปที่ 8-16 แสดง Sequence diagram ของ การแก้ไขสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

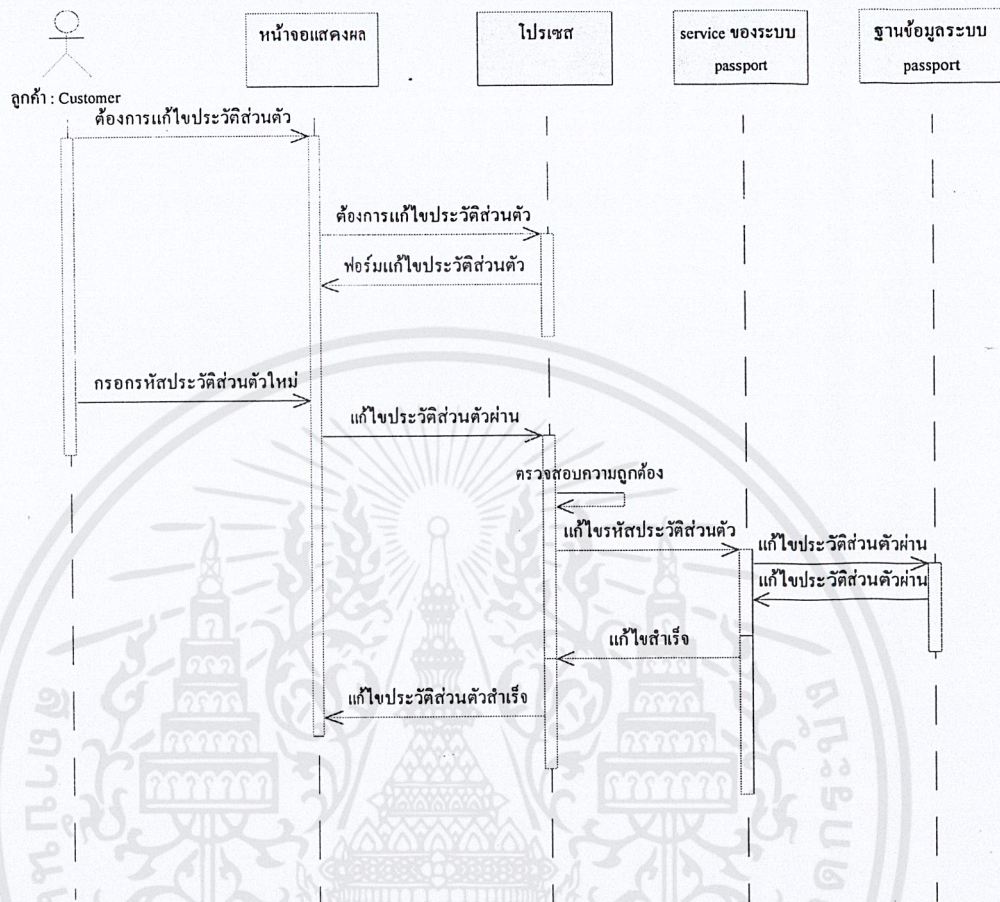
## การแก้ไขรหัสผ่าน (Sequence diagram)



รูปที่ 8-17 แสดง Sequence diagram ของ การแก้ไขรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การแก้ไขประวัติส่วนตัว (Sequence diagram)



รูปที่ 8-18 แสดง Sequence diagram ของ การแก้ไขประวัติส่วนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8.4 การออกแบบฐานข้อมูล

### 8.4.1 ระบบ passport

ได้มีการ ออกแบบตาราง ฐานข้อมูลไว้ 4 ตาราง คือ

- 1.biography ทำหน้าที่เก็บประวัติของสมาชิกของสมาชิกทั้งหมด
- 2.member ทำหน้าที่เก็บรหัสผ่าน
- 3.secret ทำหน้าที่เก็บคำถามคำตอบ
- 4.web\_member ทำหน้าที่เก็บรายชื่อ เว็บไซต์ ที่เป็นสมาชิก

biography

email	name	surname	address	province	zipcode	country	timezone	sex	birthday	job
-------	------	---------	---------	----------	---------	---------	----------	-----	----------	-----

member

Email	Password	status
-------	----------	--------

secret

Email	Q_secret	A_secret
-------	----------	----------

Web\_member

Id	Host	detail
----	------	--------

### 8.4.2 ระบบการซื้อขาย อุปกรณ์คอมพิวเตอร์ บนอินเทอร์เน็ต

ได้มีการออกแบบตารางฐานข้อมูล

1. cpu ทำหน้าที่เก็บรายละเอียดของ ซีพียู
2. ram ทำหน้าที่เก็บรายละเอียดของ หน่วยความจำ
3. hdd ทำหน้าที่เก็บรายละเอียดของ ฮาร์ดดิสก์
4. mbd ทำหน้าที่เก็บรายละเอียดของเมนบอร์ด
5. vga ทำหน้าที่เก็บรายละเอียดของการ์ดแสดงผล
6. pda ทำหน้าที่เก็บรายละเอียดของ พีดีเอ
7. cdd ทำหน้าที่เก็บรายละเอียดของ ซีดีรอม
8. mon ทำหน้าที่เก็บรายละเอียดของ จอภาพ
9. snd ทำหน้าที่เก็บรายละเอียดของ ซาวด์การ์ด
- 10.spk ทำหน้าที่เก็บรายละเอียดของลำโพง
- 11.mdm ทำหน้าที่เก็บรายละเอียดของ โมเด็ม
- 12.pnt ทำหน้าที่เก็บรายละเอียดของ ปริ้นเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 13.scn ทำหน้าที่เก็บรายละเอียดของสแกนเนอร์
- 14.ups ทำหน้าที่เก็บรายละเอียดของ ยูพีเอส
- 15.scs ทำหน้าที่เก็บรายละเอียดของ เอสซีเอสไอ
- 16.bup ทำหน้าที่เก็บรายละเอียดของ หน่วยความจำสำรอง
- 17.cam ทำหน้าที่เก็บรายละเอียดของ กล้องดิจิทัล
18. hub ทำหน้าที่เก็บรายละเอียดของ ฮับและอุปกรณ์เน็ตเวิร์ก
19. lan ทำหน้าที่เก็บรายละเอียดของ การ์ดแลน
20. hdc ทำหน้าที่เก็บรายละเอียดของ กล้องพกพา
21. nbk ทำหน้าที่เก็บรายละเอียดของ โน้ตบุค
22. mcd ทำหน้าที่เก็บรายละเอียดของ เมมโมรี่การ์ด
23. opt ทำหน้าที่เก็บรายละเอียดของอุปกรณ์เสริมต่างๆเช่น จอยสติคค์
24. group ทำหน้าที่เก็บหมวดสินค้า
25. Orders ทำหน้าที่เก็บบันทึกรายการสินค้าที่จะสั่งซื้อ
26. OrderDetail ทำหน้าที่เก็บบันทึกรายละเอียดสินค้าของแต่ละสมาชิก
27. topten ทำหน้าที่เก็บรายละเอียดสินค้าใน10 อันดับที่ซื้อขายสูงสุด
28. m\_update ทำหน้าที่เก็บรายละเอียดข่าวของสินค้าใหม่ที่จะนำเสนอ
- 29.Answer ทำหน้าที่เก็บคำตอบของกระทู้ต่างๆ
30. Question ทำหน้าที่เก็บหัวข้อของกระทู้ต่างๆ

### ส่วนหัวข้อมูล

cpu

ID	Brand	Model	Speed	FSB	Multipier	Vcore	Interface	Feature
----	-------	-------	-------	-----	-----------	-------	-----------	---------

ram

ID	Brand	Module	Type	chip	zipcode	Pin	Size
----	-------	--------	------	------	---------	-----	------

hdd

ID	Brand	Model	Size	Interface	More
----	-------	-------	------	-----------	------

mbd

ID	Brand	Model	Chipset	Interface	Form	Slot	More
----	-------	-------	---------	-----------	------	------	------

vga

ID	Brand	Model	GraphicsChip	Ram	More
----	-------	-------	--------------	-----	------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pda

ID	Brand	Model	Speed	OS	Memory	More
----	-------	-------	-------	----	--------	------

cdd

ID	Brand	X	Interface	More
----	-------	---	-----------	------

mon

ID	Brand	Model	Size	More
----	-------	-------	------	------

snd

ID	Brand	Model	Size	More
----	-------	-------	------	------

spk

ID	Brand	Model	More
----	-------	-------	------

mdm

ID	Brand	Speed	Type	Interface	More
----	-------	-------	------	-----------	------

pnt

ID	Brand	Model	More
----	-------	-------	------

scn

ID	Brand	Model	More
----	-------	-------	------

ups

ID	Brand	Model	More
----	-------	-------	------

scs

ID	Brand	Model	Type	More
----	-------	-------	------	------

bup

ID	Brand	Interface	More
----	-------	-----------	------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cam

ID	Brand	Model	Resolution	Memeroy	More
----	-------	-------	------------	---------	------

Hub

ID	Brand	Model	Port	More
----	-------	-------	------	------

lan

ID	Brand	Interface	Model	More
----	-------	-----------	-------	------

hdc

ID	Brand	Model	More
----	-------	-------	------

nbc

ID	Brand	Model	Cpu	Ram	Hdd	CD Drive	Floppy Disk	Display	Modem	Weight
----	-------	-------	-----	-----	-----	----------	-------------	---------	-------	--------

mcd

ID	Brand	Type	Size
----	-------	------	------

opt

ID	Brand	Type	Interface	More
----	-------	------	-----------	------

Groups

GroupID	GroupName
---------	-----------

Orders

OrderID	OrderDate	name	surname	Address	province	zipcode	CreditType	CreditNo
---------	-----------	------	---------	---------	----------	---------	------------	----------

OrderDetail

ID	Brand	Type	Interface	More
----	-------	------	-----------	------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

topten

ID	Product	Coount
----	---------	--------

M\_update

M_ID	Product	Detail	Picture
------	---------	--------	---------

Answer

Number	ID question	Name	Date	Note	province	Email	CreditType	ip
--------	----------------	------	------	------	----------	-------	------------	----

Question

ID question	Topic	Name	Date	Note	Email	Ip	NAnswer
----------------	-------	------	------	------	-------	----	---------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 9

### Passport System

#### 9.1 หลักการทำงาน

Passport เป็นบริการที่ช่วยให้ผู้ใช้สามารถสร้างชื่อและรหัสผ่านสำหรับลงชื่อเข้าใช้แบบครั้งเดียว สำหรับใช้กับทุกบริการและทุกไซต์ที่มีบริการของ Passport



รูปที่ 9-1 โครงสร้างพื้นฐาน Web service

#### 9.2 การเก็บรวบรวมข้อมูลส่วนตัวโดย PASSPORT

##### 9.2.1 ข้อมูลชนิดใดบ้างที่ Passport เก็บรวบรวม

เมื่อใดที่คุณลงทะเบียนขอใช้ Passport ก็จะปรากฏข้อความแจ้งให้กรอกข้อมูลส่วนตัวที่จะถูกเก็บไว้ใน "แฟ้มประวัติ" สำหรับ Passport ตามที่ได้อธิบายไว้ด้านล่างนี้ ปริมาณข้อมูลที่มีการขอร้องให้กรอกนั้นจะแตกต่างกันไปตามแต่ละไซต์ที่ทำการลงทะเบียนขอใช้บริการ แต่ส่วนใหญ่แล้วแฟ้มประวัติสำหรับ Passport จะประกอบไปด้วยข้อมูลดังต่อไปนี้คือ (1) ที่อยู่อีเมล (2) ชื่อต้นและชื่อสกุล (3) ประเทศหรือภูมิภาค (4) รัฐหรือดินแดน (5) รหัส ZIP หรือรหัสไปรษณีย์ (6) ภาษา (7) โซนเวลา (8) เพศ (9) วันเกิด และ (10) อาชีพ

##### 9.2.2 การกำหนดข้อมูลที่จะเก็บรวบรวมโดยไซต์ที่ทำการลงทะเบียน

คุณสามารถลงทะเบียนขอใช้ Passport จากบริการหรือไซต์ที่มีบริการของ Passport หรือจาก เว็บไซต์ Passport ได้ ซึ่งไซต์ที่ทำการลงทะเบียนจะเป็นตัวกำหนดปริมาณข้อมูลที่ Passport จะเก็บรวบรวมไว้ขณะที่มีการลงทะเบียน

### 9.3 การบริหารจัดการข้อมูลส่วนตัวของคุณ

#### 9.3.1 การเข้าไปใช้ข้อมูลส่วนตัวของคุณ

คุณสามารถเพิ่ม ปรับปรุง หรือทำการเปลี่ยนแปลงอื่น ๆ กับข้อมูลใน ของคุณได้ตลอดเวลา

#### 9.3.2 การปิดบัญชี Passport ของคุณ

ผู้ให้บัญชี Passport จะเป็นผู้กำหนดขั้นตอนการปิดบัญชี Passport อันนั้นของคุณ ดังอธิบายไว้

#### 9.3.3 ความปลอดภัยและแหล่งเก็บข้อมูลส่วนตัว

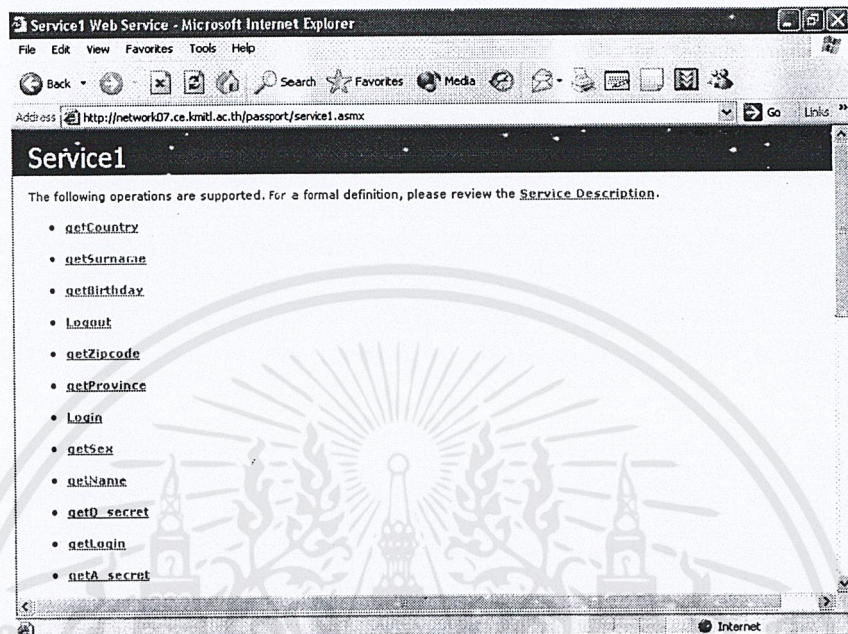
เมื่อคุณแสดงความจำนงให้มีการส่งข้อมูลสำหรับ Passport ของคุณ ไปยังไซต์ที่มีบริการของ Passport นั้น Passport จะใช้เทคโนโลยีด้านการรักษาความปลอดภัยมาตรฐานอุตสาหกรรมในการเข้ารหัสลับข้อมูลเพื่อส่งผ่านทางอินเทอร์เน็ตอย่างปลอดภัย

#### 9.3.4 การใช้งานคุกกี้

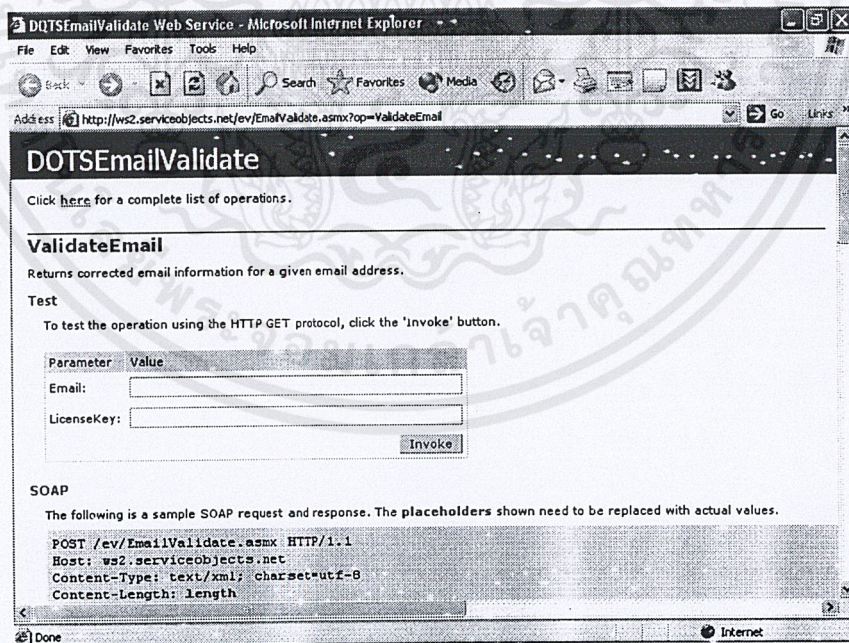
คุกกี้ คือแฟ้มข้อความขนาดเล็กมากที่เว็บไซต์บนที่กลบบนฮาร์ดดิสก์ของคอมพิวเตอร์เพื่อเก็บข้อมูลที่คุณกรอกเกี่ยวกับตัวเอง หรือเพื่อเก็บข้อมูลสิ่งที่คุณชอบของคุณ Passport จะใช้คุกกี้ทุกครั้งที่คุณลงชื่อเข้าใช้ไซต์ที่มีบริการของ Passport โดยจะเก็บตัวบ่งชี้เอกลักษณ์ของคุณ เวลาที่คุณเคยลงชื่อเข้าใช้ และข้อมูลใดก็ตามในแฟ้มประวัติสำหรับ Passport ที่คุณได้เลือกไว้เพื่อแบ่งให้ใช้ร่วมกับ ไซต์ที่มีบริการดังกล่าวจากในคุกกี้ที่เข้ารหัสลับอย่างรัดกุมบนฮาร์ดดิสก์

#### 9.4 ส่วนตัวระบบ

สามารถใช้บริการของ web service ตามรูป โดยออกแบบให้สามารถเรียกใช้ service เพิ่มประวัติของลูกค้าได้ไม่ว่าจะเป็น วันเดือน ปีเกิด ที่อยู่ สถานะ อื่นๆ ที่ได้กล่าวมาแล้ว



รูปที่ 9-2 แสดง service ต่างๆใน รูปแบบ asmx ของ ระบบ passport



รูปที่ 9-3 แสดง service ตรวจสอบ e-mail ใน รูปแบบ asmx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.4.1 ความปลอดภัย

เราได้เพิ่มส่วนการทำงานจากการทำงานปกติคือ

1. การทำ input & output validation
2. การใช้ HTML forms
3. การใช้ Cookies
4. การใช้ HTTP REFERER Header
5. การใช้ POST & GET method
6. มีกระบวนการในการทำ logout(logout mechanism)
7. Error Handling

#### การทำ input & output validation

1. ในการทำงานของเว็บแอปพลิเคชัน จะมีการรับข้อมูลบางอย่างจาก user เพื่อใช้ในการทำงาน และทำการส่งผลลัพธ์ไปให้ user ซึ่งอินพุตจากยูสเซอร์ เป็นสาเหตุหลักที่ทำให้ระบบเกิดความไม่ปลอดภัย โดยผู้เขียนระบบต้องพึงระลึกถึงว่าไม่ควรเชื่อถือข้อมูลใดๆที่ส่งมาจากฝั่ง ไคลเอนต์
2. สำหรับ Client Side Script เช่น JavaScript ,VBScript อื่นๆ สามารถเปลี่ยนแปลงได้จึงไม่ควรเชื่อผลลัพธ์มากนัก
3. ทำ Sanity Checking โดยการตรวจสอบ input ทุกอย่าง que เข้ามาในระบบ เพื่อตรวจว่าข้อมูลที่เรากำลังต้องการมีอะไรบ้าง เช่นถ้า input ที่ต้องการ เป็น yes หรือ no เท่านั้นจะต้องทำการ drop ข้อมูลอื่นๆที่รับเข้ามาทิ้งไป
4. ควรมีการตรวจสอบอักขระพิเศษต่างๆด้วย เพื่ออักขระพิเศษต่างมักเป็นต้นกำเนิดของปัญหาการใช้ฟังก์ชัน หรือ system call ที่ผิดปกติ
5. ถ้าเว็บแอปพลิเคชันไม่จำเป็นต้องใช้งานตัวอักษร HTML ต่างๆ ก็ควรทำการกรองตัวอักษร HTML ก่อนและเปลี่ยนรูปแบบตัวอักษรให้รูปอื่น เช่น > เปลี่ยนเป็น &gt;
6. การดึงข้อมูลจากฐานข้อมูลแล้วแสดงผลให้ กับยูสเซอร์ควรมีการกรองข้อมูล และเปลี่ยนแปลงค่าตัวอักษรที่อยู่ในข้อมูล HTML ก่อน เพื่อป้องกันการส่งคำสั่งมาทำงานที่เซิร์ฟเวอร์

### 9.4.2 การใช้ html forms

การใช้ hidden form element โดยจะมอง hidden เป็นเหมือนกับข้อมูลที่รับมาจากUser แต่ไม่ควรใช้กับข้อมูลที่มีความสำคัญมาก

### 9.4.3 การใช้ Cookies

Cookies เป็นเนื้อที่ในการเก็บข้อมูลในการทำงานบางส่วนไว้ที่ ไคลเอนต์ โดยมี สองประเภท คือ

1. persistent : เป็น Cookie ที่ไม่มีการลบข้อมูลออกแม้ว่าจะปิดแอปพลิเคชันไปแล้วก็ตาม
2. non-persistent : เป็น Cookie ที่ลบข้อมูลออกจาก ไคลเอนต์เมื่อหมดเวลา หรือเมื่อปิดแอปพลิเคชันไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 9.4.4 การใช้ HTTP REFERER Header

การป้องกัน script attack อีกทางหนึ่งที่สามารถป้องกัน script attack ได้บ้าง แต่ป้องกันได้ไม่

หมด

#### 9.4.5 การใช้ POST \$ GET method

ไม่ควรใช้ method GET ในการส่งข้อมูลที่มีความสำคัญ ในกรณี เว็บแอปพลิเคชันควรใช้ POST เท่านั้น และควรตั้งค่าให้ เซิร์ฟเวอร์ตอบสนองเฉพาะ POST เท่านั้น

มีกระบวนการในการทำ logout (logout mechanism)

การเพิ่มกระบวนการในการ Logout ในการทำงานต่างๆ ในเว็บแอปพลิเคชัน นั้น มีประโยชน์ในการ

1. ลบ Cookies หรือทำให้ Cookies ที่ฝั่งไคลเอนต์ไม่สามารถทำงานได้
2. จัดการทับ session ทางฝั่งเซิร์ฟเวอร์เพื่อป้องกันการขโมย session ในกรณี Cookie ที่ไคลเอนต์ไม่

หมดอายุ

#### 9.5 Error Handling Mechanism

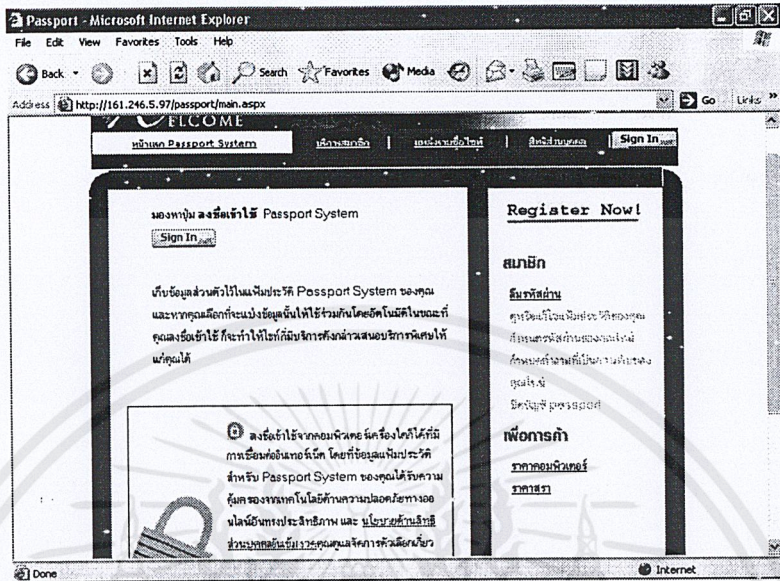
การทำ Error Handling ที่มีการแจ้ง Error Description นั้น เป็นประโยชน์ในการแก้ปัญหาที่มัก ไม่ควรส่ง Error Description ต่างๆ ไปให้ user เนื่องจากจะทำให้ผู้บุกรุกรู้รายละเอียดได้

#### 9.6 การติดตั้ง

ไม่ต้องทำการติดตั้งเพราะการทำงานอยู่ในรูปของ Web สามารถนำไฟล์ไปเก็บไว้ที่ Web Server แล้วสร้างตารางฐานข้อมูลที่มีชื่อตารางและชื่อคอลัมภ์เหมือนกับที่แสดงไว้ในส่วนของการออกแบบก็สามารถใช้งานได้เลย

## 9.7 การใช้งาน

เมื่อเข้ามาที่ web passport ดังรูป



รูปที่ 9-4 แสดงหน้า Web Passport

ประกอบด้วยเมนูใหญ่

1. หน้าแรก passport system
2. บริการสมาชิก
3. สิทธิส่วนบุคคล

### 9.7.1 หน้าแรก passport system

สำหรับผู้ที่ยังไม่ได้ทำการสมัครเป็นสมาชิกของ Passport จะต้องทำการสมัครสมาชิกก่อน โดยคลิกที่ Register Now! จะทำการเข้าสู่ หน้าการลงทะเบียนผู้ใช้ใหม่

Step 1: กรอกที่อยู่อีเมลและรหัสผ่านของคุณ เพื่อทำการลงทะเบียนเป็นสมาชิก ดังรูปที่ 9-5 ซึ่งในขั้นตอนนี้จะมีการเรียกใช้บริการจากภายนอก เพื่อทำการตรวจสอบเมล ว่ามีอยู่จริงหรือไม่ ตามรูปที่ 9-3(ข) ซึ่ง ถ้ามีจะสามารถทำงานต่อไปได้ เลือกทำต่อไป ถ้าไม่ได้กรอกข้อมูลหรือป้อนรหัสไม่เหมือนกันจะทำการแจ้งเตือนให้กรอกข้อมูลให้ครบ

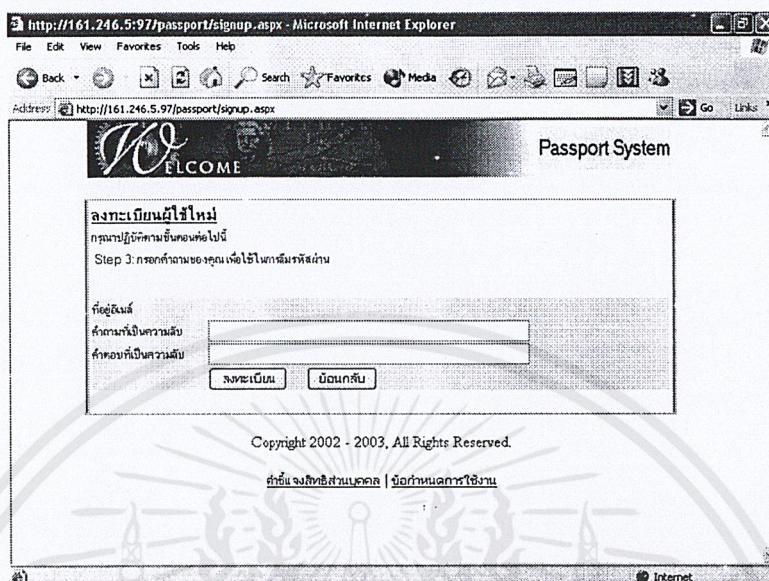
รูปที่ 9-5 แสดงหน้าให้กรอกที่อยู่อีเมลและรหัสผ่านของคุณ เพื่อทำการลงทะเบียนเป็นสมาชิก

Step 2: กรอกข้อมูลส่วนตัวของคุณ เพื่อทำการลงทะเบียนเป็นสมาชิก ดังรูป

รูปที่ 9-6 แสดงหน้ากรอกข้อมูลส่วนตัว เพื่อทำการลงทะเบียนเป็นสมาชิก

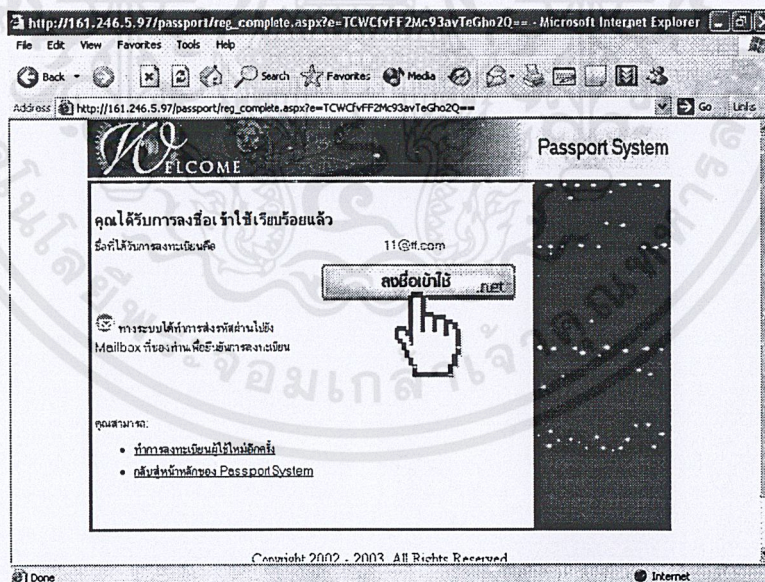
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step 3: กรอกคำถามของคุณ เพื่อใช้เมื่อลืมรหัสผ่าน ดังรูป



รูปที่ 9-7 แสดงหน้ากรอกคำถาม เพื่อใช้เมื่อลืมรหัสผ่าน

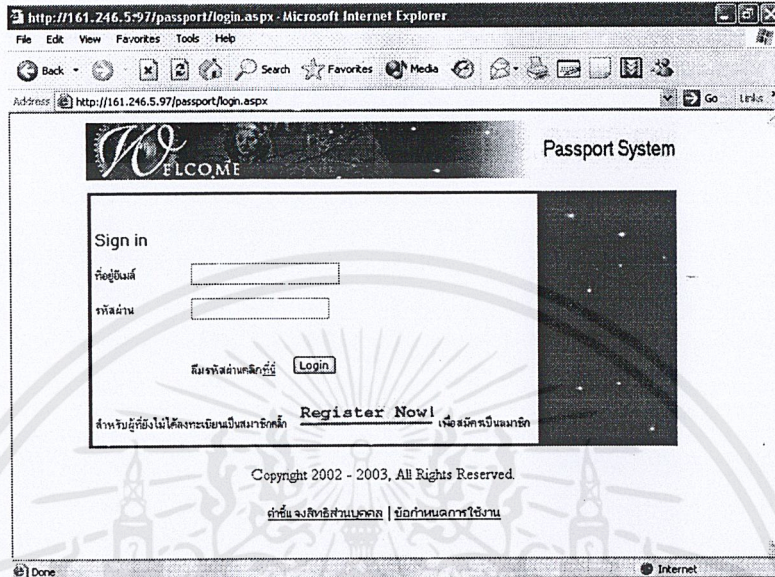
Step 4: เสร็จสิ้นการสมัคร



รูปที่ 9-8 แสดงหน้าเสร็จสิ้นการสมัคร

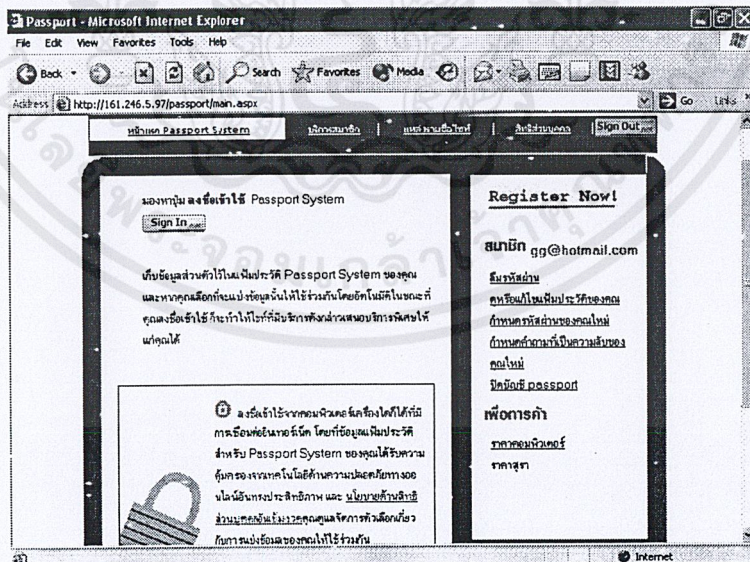
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการสมัครเป็นสมาชิกแล้ว สามารถที่จะ login เข้าสู่ระบบ ที่ปุ่ม **Sign In** net ก็จะเข้าสู่หน้า การ login ดังรูป



รูปที่ 9-9 หน้าการ login เพื่อเข้าสู่ระบบ passport system

โดยทำการกรอก e-mail และ password ถ้าถูกต้องจะเข้าสู่การทำงานของ passport จะสามารถใช้งาน ต่างๆ ได้ ดังรูป



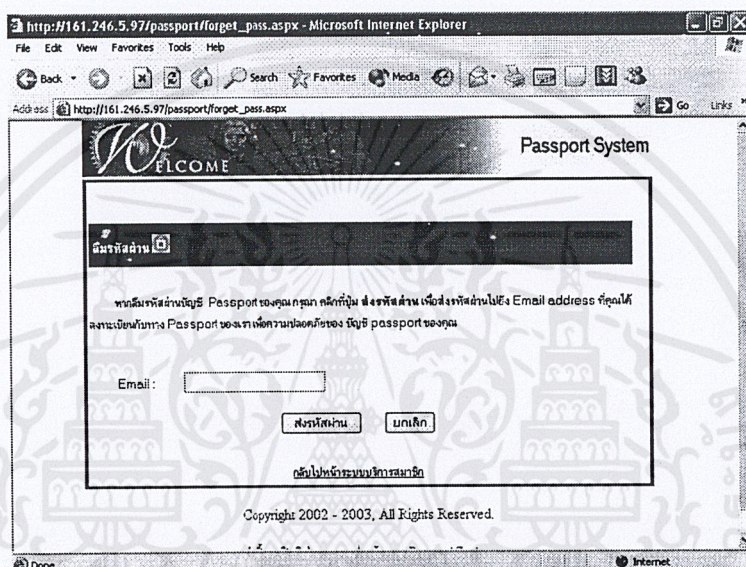
รูปที่ 9-10 หน้าการทำงานของ passport

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถทำการแก้ไขข้อมูลดังต่อไปนี้

1. เข้าไปแก้ไขเพิ่มประวัติได้
2. กำหนดรหัสผ่านใหม่ จะต้องทำการใส่ password เก่า ก่อนแล้วใส่ password ใหม่
3. กำหนดคำถามที่เป็นความลับ
4. สามารถทำการยกเลิกการใช้งาน passport ได้

สำหรับสมาชิกที่ลืม password ลืมรหัสผ่านก็จะหน้า ลืมรหัสผ่าน จะให้ทำการ ใส่ข้อมูล e-mail แล้วคลิก ส่งรหัสผ่าน ระบบจะทำการตรวจสอบ e-mail แล้วทำการส่ง password นั้น ไปที่ e-mail นั้น

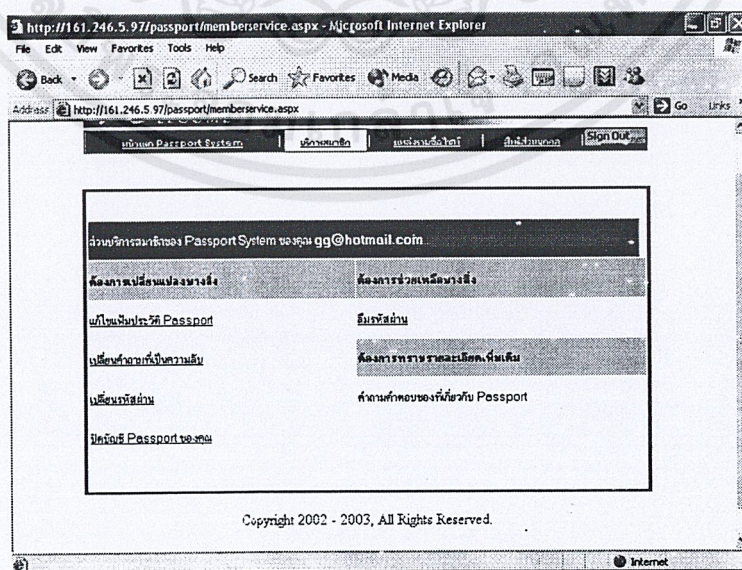


รูปที่ 9-11 แสดงหน้าสำหรับกรอก e-mail เมื่อ ลืมรหัสผ่าน

### 9.7.2 บริการสมาชิก

เป็นหน้าที่ให้บริการสำหรับสมาชิกโดยต้องทำการ login ก่อนถึงจะเข้าสู่การทำงานหน้านี้ได้ ดัง

รูป



รูปที่ 9-12 เป็นหน้าที่ให้บริการสำหรับสมาชิก

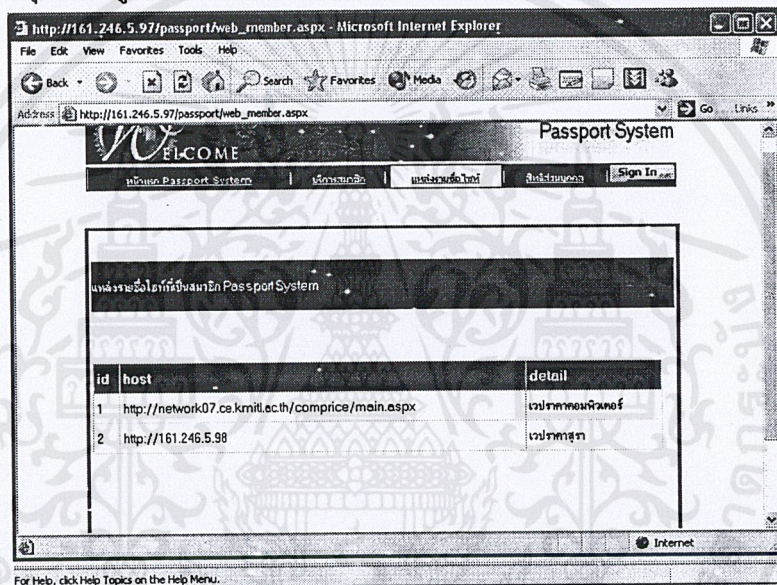
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วย

- 2.1 เข้าไปแก้ไขประวัติได้
- 2.2 กำหนดรหัสผ่านใหม่
- 2.3 กำหนดคำถามที่เป็นความลับ
- 2.4 สามารถทำการยกเลิกการใช้งาน passport ได้

### 9.7.3 แหล่งรวมเว็บไซต์

ประกอบด้วย web ที่เป็นสมาชิกของ passport การใช้งานของต้องคลิก ปุ่มเพื่อทำการ Sign In เข้าระบบ ป้อน e-mail กับ password เว็บนั้นก็สามารถใช้ข้อมูลของสมาชิกที่สมัครไว้ที่ passport ตามเงื่อนไขสิทธิส่วนบุคคลดังรูป



รูปที่ 9-13 web ที่เป็นสมาชิกของ passport

### 9.7.4 สิทธิส่วนบุคคล

เป็นรายละเอียดสิทธิส่วนบุคคลของ passport ข้อกำหนดการใช้งานต่างๆ เมื่อต้องการออกจากระบบ passport ต้องทำการกด **Sign Out** net ก็จะออกจากระบบ passport

## บทที่ 10

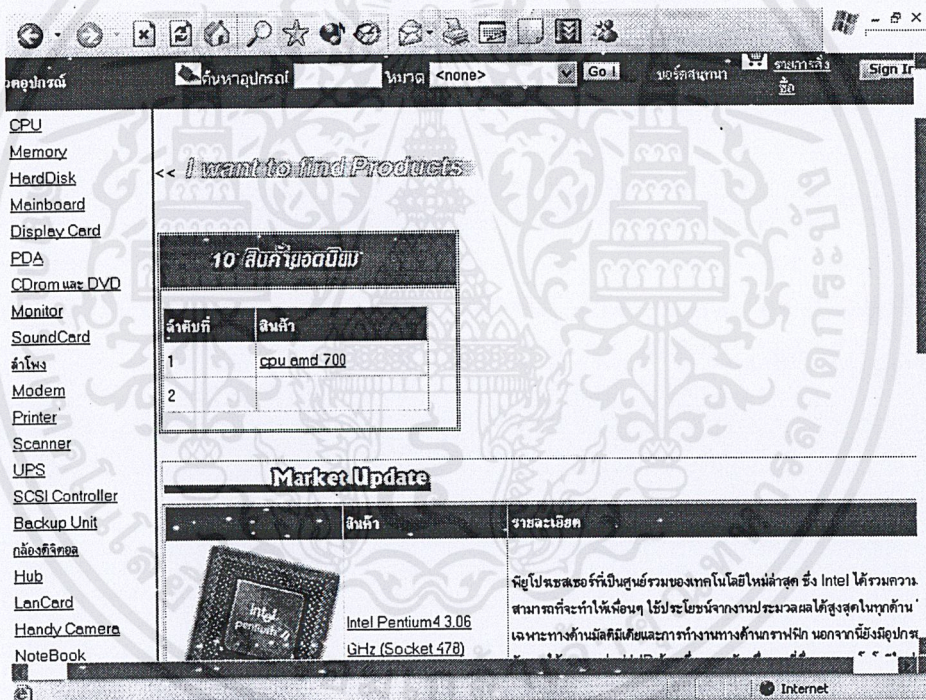
### ระบบซื้อขายผ่านอินเทอร์เน็ต

#### 10.1 ระบบซื้อขาย อุปกรณ์ คอมพิวเตอร์ บนอินเทอร์เน็ต

ระบบการซื้อขาย อุปกรณ์ คอมพิวเตอร์ จะเป็นระบบที่ใช้ในการสั่งซื้ออุปกรณ์คอมพิวเตอร์ โดยจะให้บริการผ่านทาง อินเทอร์เน็ต ลูกค้าสามารถทำการตรวจเช็คราคาสินค้า และสั่งซื้ออุปกรณ์คอมพิวเตอร์ แล้วใส่รายการที่ต้องการลงใน รายการสินค้าที่สั่งซื้อ เมื่อได้สินค้าครบตามที่ต้องการแล้ว ก็ทำการสั่งซื้อ และต้องทำการยืนยันการสั่งซื้ออีกที ภายหลัง

#### 10.2 การใช้งาน เว็บ ระบบการซื้อขาย อุปกรณ์ คอมพิวเตอร์ บนอินเทอร์เน็ต

##### หน้าจอหลัก



รูปที่ 10-1 แสดงหน้าจอหลัก ของ ระบบการซื้อขาย อุปกรณ์คอมพิวเตอร์

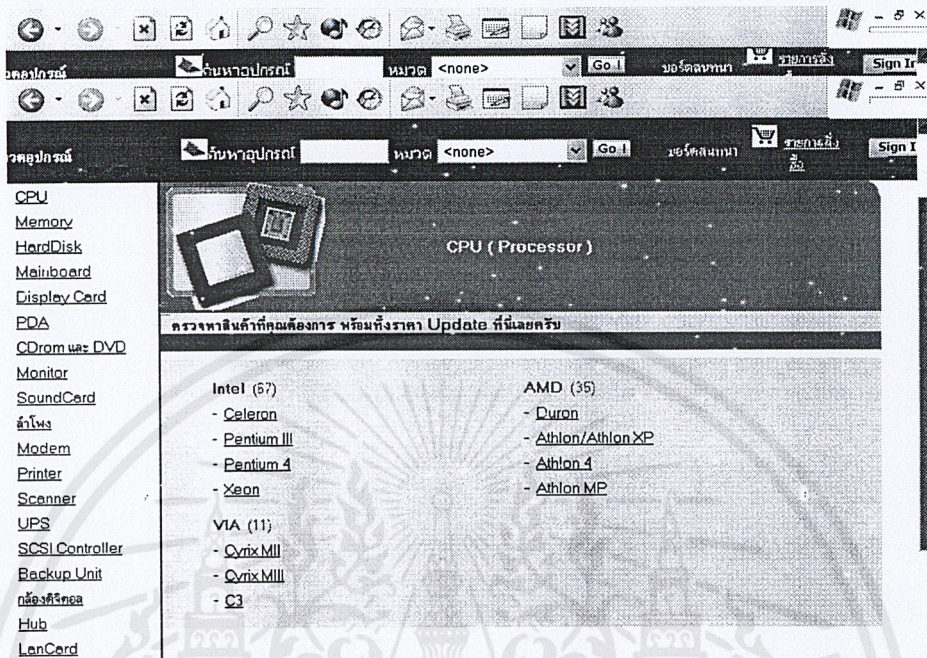
ประกอบด้วยส่วนประกอบหลักคือ

- เมนูรายการสินค้า : แสดงรายการของสินค้าแบ่งตามชนิดประเภทของแต่ละอุปกรณ์ เช่น CPU,Memory เป็นต้น
- ค้นหาอุปกรณ์ : ใช้ในการค้นหาสินค้าที่ต้องการโดยใส่ ชื่อสินค้าที่ต้องการ เลือกประเภทที่ต้องการ
- รายการสั่งซื้อ : ไว้ดูเพื่อตรวจสอบสินค้าที่ได้ทำการสั่งซื้อแล้วและสามารถแก้ไขได้
- ส่วนแสดงผล: แสดงผลรายการต่างๆที่ได้ทำการเลือก นำมาแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

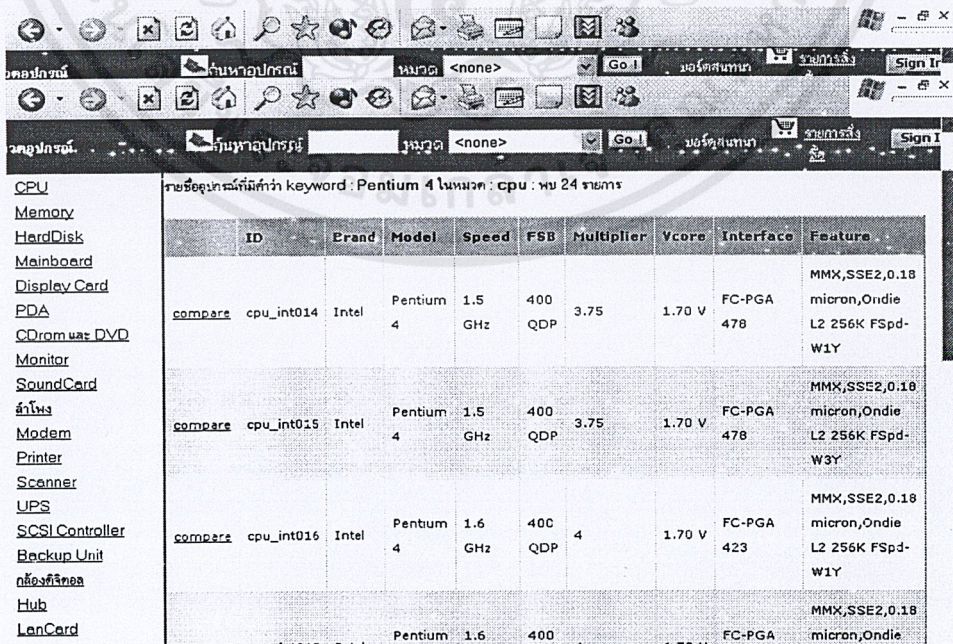
10.2.1 เมนูหลัก

จากหน้าจอหลัก เมื่อเราเลือก ชนิดของสินค้าจาก เมนูหลัก



รูปที่ 10-2 เมื่อเราเลือก ชนิดของอุปกรณ์ที่ต้องการจากเมนูหลัก

จากรูปที่ 10-2 เมื่อเราทำการเลือก CPU จากเมนูหลัก ก็จะทำการแสดงรายการของ CPU ต่างๆที่มี โดยทำการแบ่งตามประเภทยี่ห้อ รุ่น ต่างๆของ CPU จากรูปที่ 10-2 ตัวอย่าง เมื่อเราเลือก CPU Pentium 4 ก็จะเข้าสู่รายการของสินค้าที่เลือกในที่นี้คือ cpu pentium 4 ดังรูปที่ 10 -3

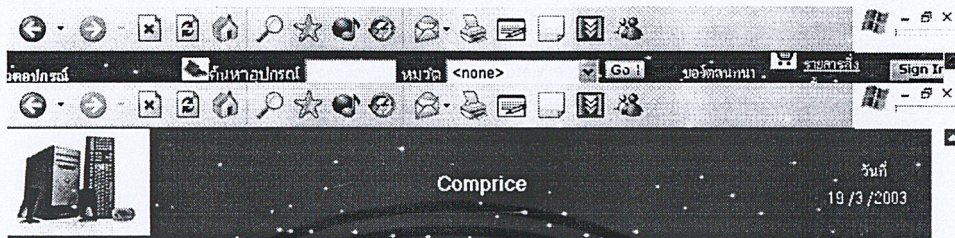


รูปที่ 10-3 แสดงรายการของประเภทสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 10-3 จะแสดงรายการของสินค้าที่ได้ทำการเลือกไว้ ในที่นี้ คือ cpu pentium 4 ก็จะทำการแสดง รุ่น และคุณสมบัติ ต่างๆ ของ cpu ที่ได้ทำการเลือกไว้

โดยเราสามารถ ตรวจสอบราคาได้โดยทำการ เลือกที่ compare ก็จะได้ดังรูปที่ 10-4 ซึ่งจะแสดงการเปรียบเทียบราคาของที่ต่างๆ



รายละเอียดของอุปกรณ์ของร้านต่างๆ

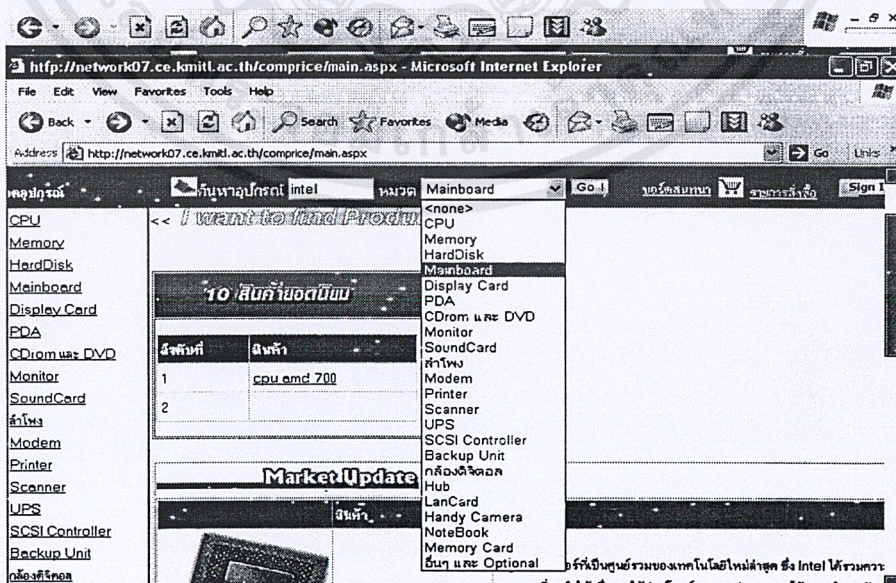
shop	Price	ID	Brand Model	Speed	FSB	Multiplier	Vcore	Interface	Feature
add basket	4800	cpu_int014	Intel Pentium 4	1.5 GHz	400 QDP	3.75	1.70 V	FC-PGA 478	MMX,SSE2,0.18 micron,Ondie L2 256K FSpd-W1Y
add basket	4800	cpu_int014	Intel Pentium 4	1.5 GHz	400 QDP	3.75	1.70 V	FC-PGA 478	MMX,SSE2,0.18 micron,Ondie L2 256K FSpd-W1Y

กลับไปค้นหา  
Copyright 2002 - 2003, All Rights Reserved.

รูปที่ 10-4 แสดงการเปรียบเทียบ ราคาที่ต่างๆ

### 10.2.2 ส่วนการค้นหาอุปกรณ์

จากหน้าจอหลัก เมื่อเราต้องการค้นหาอุปกรณ์ แบบรวดเร็ว



รูปที่ 10-5 แสดงส่วนการค้นหาแบบรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 10-5 ส่วนของการค้นหาแบบรวดเร็ว จะประกอบด้วย 2 ส่วนคือ

- ส่วนที่ใส่ชื่ออุปกรณ์สินค้า
- หมวดของสินค้า

ตัวอย่างเมื่อเราเลือกตามรูปที่10.5 จะได้ ตามรูปที่ 10-6

รูปที่ 10-6 แสดงรายการสินค้าที่ได้จากการค้นหา

ID	Brand	Model	Chipset	Interface	Form	Slot	More
compare mbd_int001	Intel	D815EPEA	i815EP	Socket 370	ATX	?	AGP 4X , ATA/100,Sound
compare mbd_int002	Intel	D815EEA	i815E	Socket 370	ATX	?	AGP 4X , ATA/100, VGA
compare mbd_int003	Intel	D850GB	i850	Socket 423	ATX	?	AGP 4X , ATA/100,RDRAM
compare mbd_int004	Intel	D850MV	i850	Socket 478	ATX	?	AGP 4X , ATA/100,RDRAM
compare mbd_int005	Intel	D845WN	i845S	Socket 478	ATX	?	AGP 4X , ATA/100,SDRAM
compare mbd_int006	Intel	D845GC	i845D	Socket 478	ATX	?	AGP 4X ,

รูปที่ 10-6 แสดงรายการสินค้าที่ได้จากการค้นหา

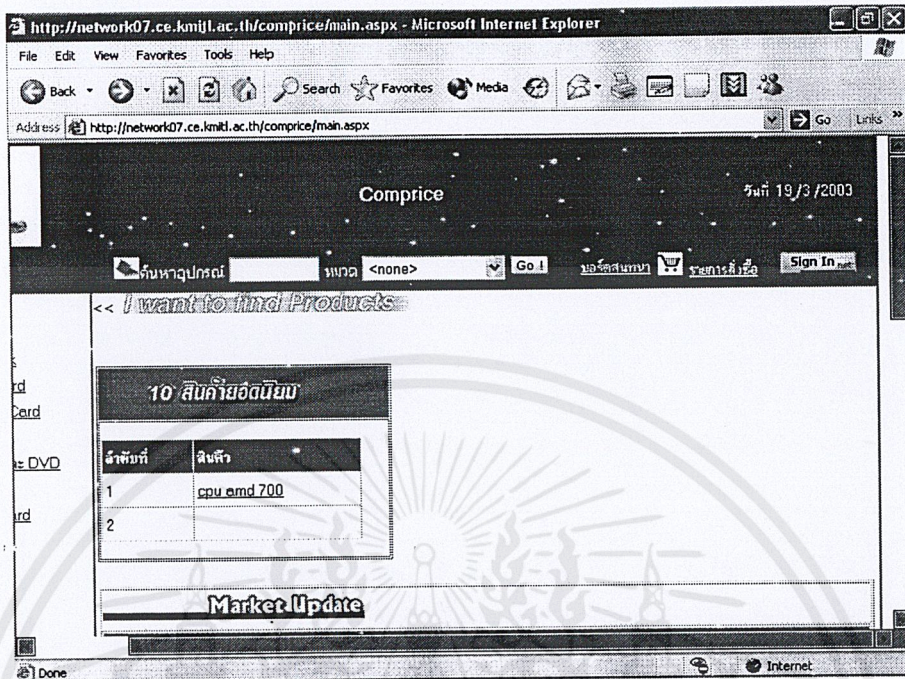
แต่ถ้าไม่พบสินค้าที่ต้องการหาจะแสดงตามรูปที่ 10-7

รูปที่ 10-7 แสดงรายการสินค้าปรากฏว่าไม่พบ

รูปที่ 10-7 แสดงรายการสินค้าปรากฏว่าไม่พบ

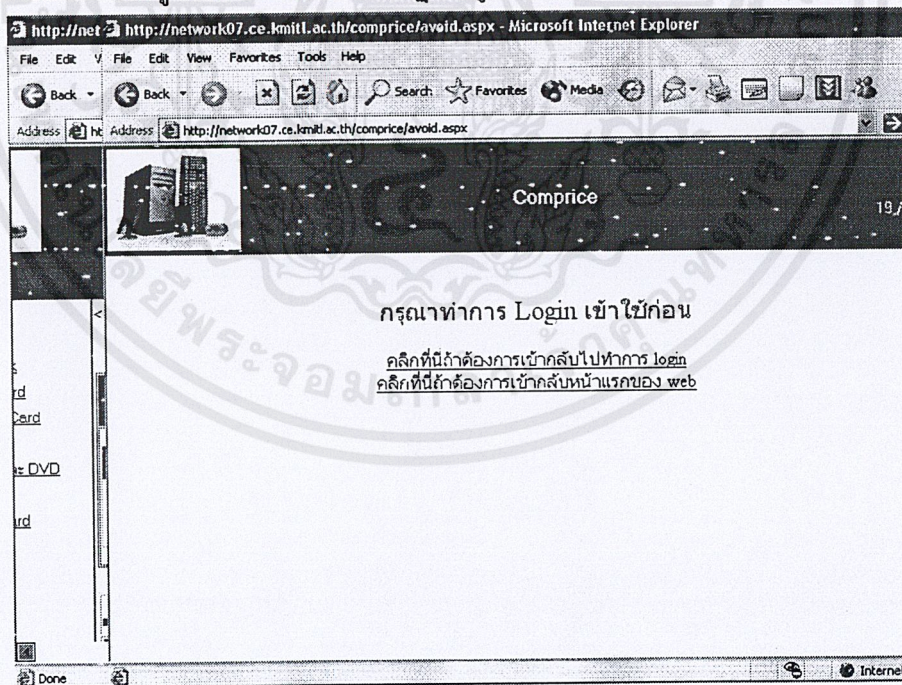
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 10.2.3 รายการสั่งซื้อ



รูปที่ 10-8 แสดงเมนู รายการสั่งซื้อ

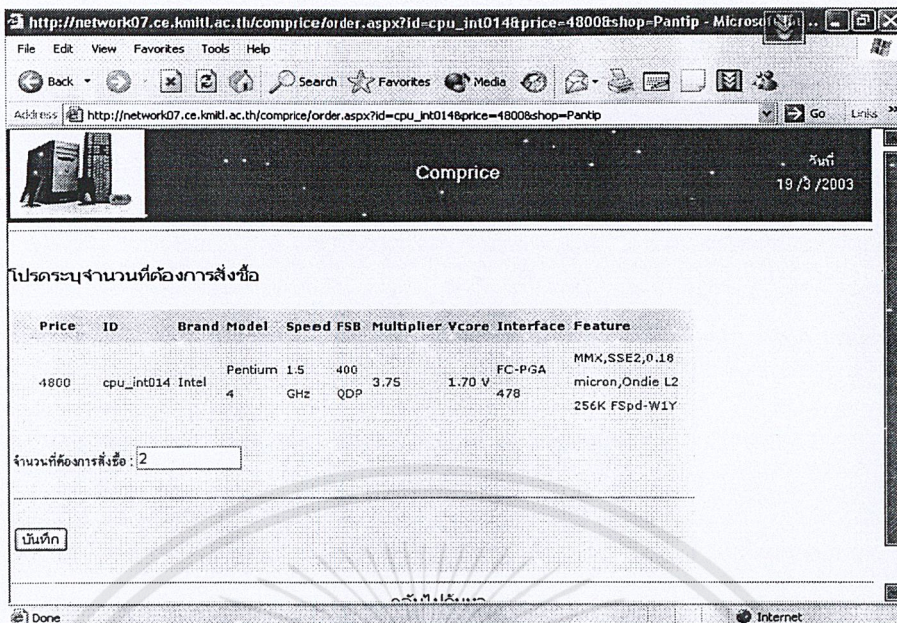
เมื่อเราทำการเลือก ที่ เมนู รายการสั่งซื้อ จะปรากฏตามรูปที่ 10-9



รูปที่ 10-9 แสดงผล เมื่อไม่ได้ทำการ login

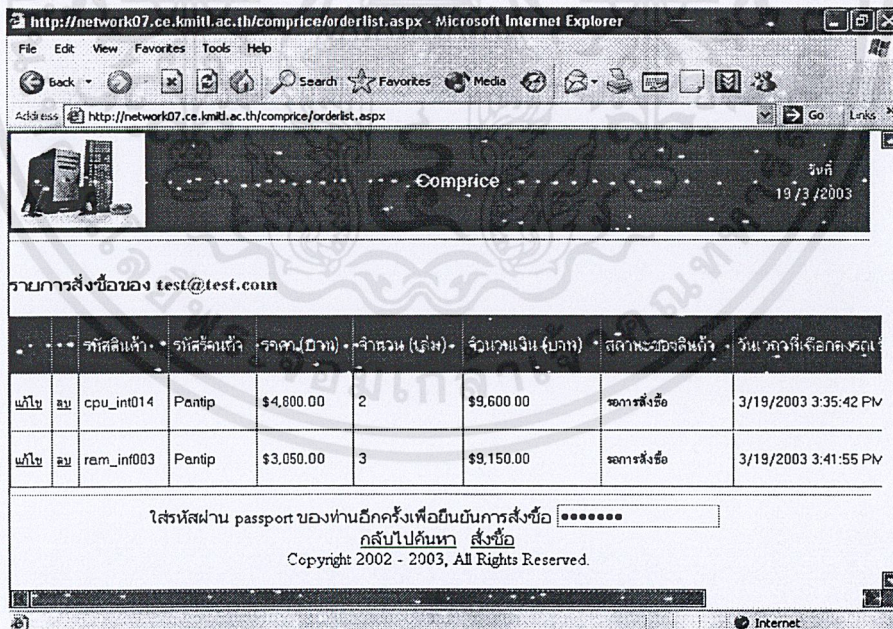
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 10-12 แสดงรายการที่สั่งเพื่อให้ระบุจำนวนที่ต้องการ

จากรูปที่ 10-12 เมื่อเราทำการใส่จำนวนสินค้าที่ต้องการ แล้วทำการกดที่ บันทึก เพื่อทำการบันทึกการซื้อสินค้า เราสามารถเลือกซื้อ สินค้าอื่นได้อีก โดยทำตาม วิธีการเลือกซื้ออีกทีก็ได้ เมื่อได้สินค้าที่ต้องการเรียบร้อยแล้ว เราสามารถ ตรวจสอบ สินค้า และ ราคา ได้ โคนเลือกที่ เมนู รายการสั่งซื้อที่เมนูหลัก จะได้ดังรูปที่ 10-13



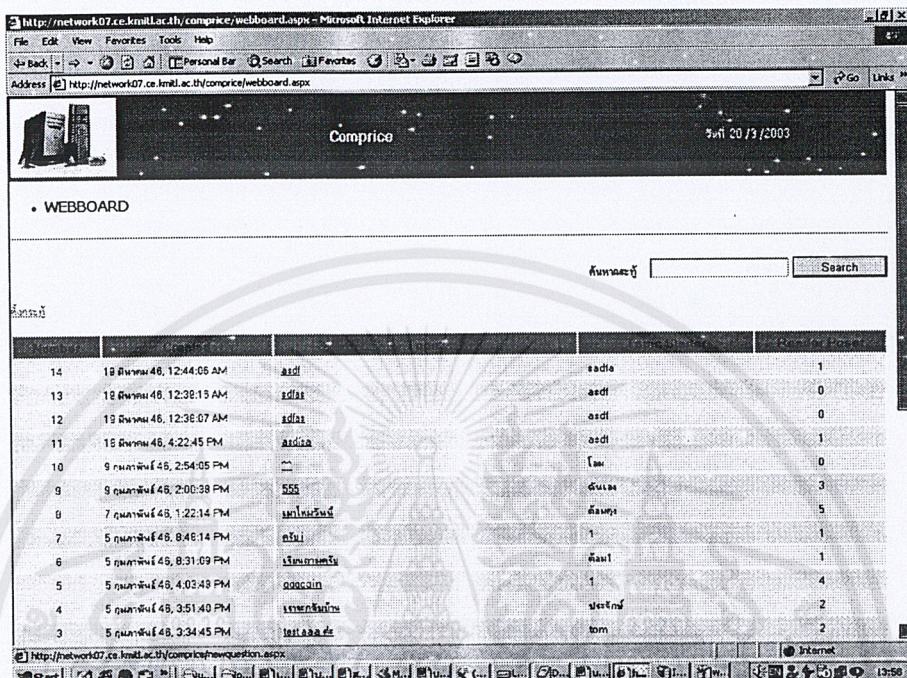
รูปที่ 10-13 แสดงรายการ การสั่งซื้อสินค้าทั้งหมด

จากรูปที่ 10-13 เมื่อเราต้องการสั่งซื้อ สินค้า เราจะต้องทำการใส่ รหัสผ่าน อีกครั้ง เพื่อทำการยืนยันการสั่งซื้อ และในหน้าเราสามารถทำการแก้ไข สินค้าที่ต้องการ ได้ และสามารถ ลบ ได้ เมื่อทำการสั่งซื้อแล้วจะมี อีเมล ส่งไปที่ สมาชิกที่ได้ทำการสั่งซื้อสินค้า เพื่อให้สมาชิกได้ทำการยืนยัน อีกครั้ง ก่อนการสั่งซื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 10.2.4 เว็บบอร์ด

เมื่อเราทำการเลือกที่ บอร์ด สนทนา ที่หน้าจอหลัก จะเข้าสู่หน้า เว็บบอร์ด ดังรูปที่ 10-14

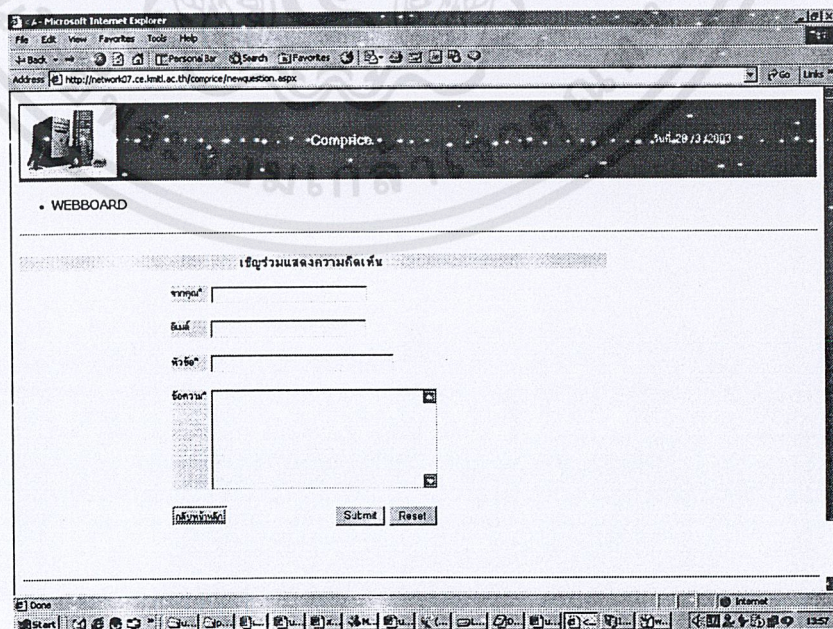


The screenshot shows a web browser window displaying a forum page. The page title is "Comprice" and the date is "วันที่ 20/3/2003". Below the title, there is a search bar and a "Search" button. The main content is a table of posts, with columns for "หัวข้อ" (Topic), "ผู้โพสต์" (Posted by), "เวลา" (Time), "จำนวน" (Count), and "ตอบกลับ" (Reply). The table contains 14 rows of data.

หัวข้อ	ผู้โพสต์	เวลา	จำนวน	ตอบกลับ
14	18 สิงหาคม 48, 12:44:05 AM	asdi	radia	1
13	18 สิงหาคม 48, 12:38:13 AM	asdi	asdi	0
12	19 สิงหาคม 48, 12:36:07 AM	asdi	asdi	0
11	18 สิงหาคม 48, 4:22:45 PM	asdi	asdi	1
10	9 กุมภาพันธ์ 48, 2:54:05 PM	---	โหล	0
9	9 กุมภาพันธ์ 48, 2:00:38 PM	SSS	ค้นหา	3
8	7 กุมภาพันธ์ 48, 1:22:14 PM	มาใหม่	ค้นหา	5
7	5 กุมภาพันธ์ 48, 8:46:14 PM	กรีน	1	1
6	5 กุมภาพันธ์ 48, 8:31:09 PM	เว็บบอร์ด	ค้นหา	1
5	5 กุมภาพันธ์ 48, 4:03:43 PM	อดิวิ	1	4
4	5 กุมภาพันธ์ 48, 3:51:40 PM	เว็บบอร์ด	ค้นหา	2
3	5 กุมภาพันธ์ 48, 3:34:45 PM	โหล	tom	2

รูปที่ 10-14 แสดงหน้าตาของ เว็บบอร์ด

ในหน้านี้จะแสดงรายละเอียดของ ข้อความที่ถูกตั้งเป็นกระทู้ แล้ว และเราสามารถทำการตั้งกระทู้เอง ได้โดยการคลิกที่ ตั้งกระทู้ ก็จะเข้าสู่หน้าการตั้งกระทู้ ดังรูปที่ 10-15



The screenshot shows a web browser window displaying a forum page. The page title is "Comprice" and the date is "วันที่ 20/3/2003". Below the title, there is a search bar and a "Search" button. The main content is a form titled "เชิญร่วมแสดงความคิดเห็น" (Invitation to share your opinion). The form has four input fields: "หัวข้อ" (Topic), "ชื่อ" (Name), "รหัส" (Code), and "ข้อความ" (Message). Below the form, there are buttons for "ยืนยัน" (Confirm), "Submit", and "Reset".

รูปที่ 10-15 แสดงหน้า การตั้งกระทู้ของ เว็บบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการตั้งกระทู้ นี้เราสามารถทำการตั้งกระทู้ โดยจะมีให้ใส่ อีเมล หัวข้อ จากใคร และข้อความ ภายในกระทู้

จากรูปที่ 10-14 ถ้าเราต้องการดูข้อความภายในกระทู้เราก็สามารถทำได้การ คลิกที่ ส่วน Topic ซึ่งจะ ลิงค์ ไป ดังรูปที่ 10-16

รูปที่ 10-16 แสดงข้อความภายในของแต่ละกระทู้

เราสามารถ ตอบ กระทู้ได้ โดยใส่ข้อความลงในข้อความ พร้อมทั้ง ชื่อ และ อีเมล ทำการคลิก ที่ Submit เพื่อทำการส่งข้อความไปที่ เว็บบอร์ด

## บทที่ 11 วิจารณ์และสรุป

สำหรับบทวิจารณ์และสรุปจะเป็นการสรุปการทำงานในขั้นตอนต่าง ๆ โดยกล่าวเป็นส่วน ๆ พร้อมทั้งกล่าวถึงปัญหาและวิธีการแก้ไข รวมทั้งได้สรุปการทำงานโดยรวมของระบบไว้ด้วย

### 11.1 ส่วนการศึกษาระบบที่มีอยู่เดิม

ในขั้นต้น เป็นขั้นตอนของการศึกษาระบบเก่าที่มีอยู่เดิม เพื่อเป็นแนวทางในการวิเคราะห์และออกแบบสร้างระบบต่อไป โดยจากการศึกษาระบบการทำงานของ Web Services ต่าง ๆ เช่น เว็บไซต์ของ ไมโครซอฟท์, เว็บไซต์ .NET Passport, เว็บไซต์ของ MSN เป็นต้น พบว่าระบบ .NET Passport ที่มีอยู่เดิมเป็นระบบที่ปิดจนเกินไปซึ่งผูกขาดทางการค้าซึ่งเว็บไซต์ที่จะได้เป็นสมาชิกรุนั้นจะต้องถูกพิจารณาจากทางผู้ให้บริการซึ่งต้องมีผลประโยชน์ทางการค้ากับเว็บไซต์นั้นๆ จึงได้ทำการ ออกแบบระบบ Passport จำลองการใช้งานขึ้นมาเอง

#### 11.1.1 ส่วนการวิเคราะห์และการออกแบบระบบ

ในส่วนของการวิเคราะห์และออกแบบระบบนั้น ได้เลือกใช้หลักการของออบเจกต์โอเรียนเท็ด (Object Oriented) เป็นหลักในการออกแบบ เนื่องจากมีข้อดีคือ ทำให้ผู้พัฒนาสามารถวิเคราะห์และออกแบบระบบได้อย่างสะดวก และมีประสิทธิภาพ โดยใช้ภาษา UML ช่วยในการออกแบบระบบเนื่องจากเป็นภาษาที่มีมาตรฐาน และมีการใช้งานอย่างกว้างขวาง

#### 11.1.2 ส่วนการพัฒนาระบบ

ในส่วนของการพัฒนาระบบนั้น ได้เลือกใช้ซอฟต์แวร์ Ms SQL Server 2000 เป็นตัวจัดการฐานข้อมูลของระบบ ใช้ซอฟต์แวร์ IIS 5.0 เป็นเว็บเซิร์ฟเวอร์ และเป็นตัวจัดการ Service และ Web application ทั้งหมด และใช้วินโดวส์ 2000 เซิร์ฟเวอร์ (Windows 2000 Server) เป็นระบบปฏิบัติการ ส่วนตัวโปรแกรมบริหารจัดการ Service และ Web application นั้น พัฒนาโดยใช้เทคโนโลยี ASP.NET เป็นตัวพัฒนาเนื่องจากเป็นภาษาที่รองรับมาตรฐาน .NET Framework และใช้ภาษา VB.NET ในการพัฒนา ทำให้ง่ายต่อการทำความเข้าใจ และเป็นภาษาที่ไม่ขึ้นกับแพลตฟอร์ม ทำให้ไม่มีปัญหาในการนำไปใช้งานบนเครื่องต่าง ๆ โดยมี CLR เป็นตัวกลางในการทำงาน และ ใช้ ADO.NET เป็นตัวกลางในการติดต่อกับฐานข้อมูล

#### 11.1.3 ส่วนวิจารณ์และบทสรุปโดยรวมของระบบ

ระบบ พาสพอร์ต ที่ได้ พัฒนาระบบนั้น มีความสามารถครบตามที่ได้ ออกแบบมา และสามารถตอบสนองความต้องการพื้นฐานในการบริหารจัดการสั่งซื้อสินค้าได้ แต่ก็ยังถือว่ามี services ในการให้

บริการน้อยมากหากจะนำไปใช้งานจริง และเนื่องจากเว็บไซต์ที่จะขอให้บริการระบบพาสพอร์ด มีความต้องการข้อมูลที่ต่างกัน ระบบบริหาร พาสพอร์ด ที่พัฒนาขึ้นมานี้อาจไม่เหมาะสำหรับเว็บไซต์แบบอื่น ๆ

## 11.2 ปัญหาและแนวทางในการพัฒนาระบบ

### 11.2.1 ปัญหาเกี่ยวกับแหล่งอ้างอิง

ปัญหาส่วนใหญ่เกิดจากการศึกษาภาษาและเทคโนโลยีในช่วงแรกมีแหล่งเอกสารอ้างอิงเกี่ยวกับ ข้อมูลฟังก์ชันต่าง ๆ ของ .NET ยังมีน้อยมาก เนื่องจากเป็นเทคโนโลยีที่ค่อนข้างใหม่ และประกอบกันแหล่งเอกสารอ้างอิงส่วนใหญ่ที่ได้เป็นเอกสารต่างประเทศจึงลำบากในการทำความเข้าใจ

### 11.2.2 การเพิ่มในส่วนของ Wallet

จะอย่างไรเพื่อให้เว็บไซต์ต่างๆที่เป็นสมาชิกกับทางระบบ เนื่องจากเว็บไซต์ที่จะขอใช้บริการระบบพาสพอร์ด มีความต้องการข้อมูลที่ต่างกัน โดยระบบจะให้ทางเว็บไซต์ที่เป็นสมาชิคนั้นทำการเก็บข้อมูลไว้เองก็ไม่สามารถประกันความปลอดภัยแล้วข้อมูลที่เก็บนั้นอาจจะสำคัญมาก จึงสามารถจะพัฒนาเป็นระบบ wallet ซึ่งจะเป็นการเพิ่มข้อมูลของสมาชิกจากทางเว็บไซต์ที่สมาชิกไปขอใช้บริการ โดยทางระบบจะทำการเพิ่มเติมให้ ตอบสนองความต้องการของแต่ละเว็บไซต์ให้มีความสมบูรณ์แบบที่สุด

### 11.2.3 การรักษาความปลอดภัยของข้อมูล

เนื่องจากทางระบบที่ได้ออกแบบได้ทำการเพิ่มการทำงานที่ปลอดภัยขึ้น ไม่ว่าจะเป็นการตรวจสอบ input output เป็นหลักมีการใช้ SSL ในการเข้าของข้อมูล ใช้ HTTPS ในการส่งข้อมูลที่มีความสำคัญ แทนที่ HTTP มีการใช้ Cookies ในการเก็บสถานะการ login และใช้การเข้ารหัส แบบ DES ในบางกรณี ซึ่งทำให้ระบบมีความปลอดภัยในระดับหนึ่งแต่ก็ไม่ได้หมายความว่าระบบจะไม่มีช่องว่างในการบุกรุกเข้าไปได้ อธิเช่น ตัวของ IIS เองนั้นมีจุดอ่อนในการ โจมตีจากผู้บุกรุกมากทั้งการ Disclosures แล้วการ โจมตีแบบต่างๆที่ทำให้ระบบอยู่ให้บริการผู้พัฒนาต่อจึงต้องมีการพัฒนาให้ทันกับเหตุการณ์ เช่นมีการ update path ต่างๆ ซึ่งจะทำให้ระบบมีความเสถียรมากขึ้น

### บรรณานุกรม

- [1] จักรพันธุ์ โพธิวรรณ , อัมรินทร์ เพ็ชรกุล : “Microsoft Studio.net”, สำนักพิมพ์ บริษัท ชัค เซส มีเดีย จำกัด
- [2] จำลอง ครูอุตสาหะ : “ ASP.net “ , สำนักพิมพ์ บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด , พิมพ์ครั้งที่ 1 กรกฎาคม 2545
- [3] กิตติชัย พิณจิดำ : “ XML และ SOAP สำหรับ BizTalk Server “ , สำนักพิมพ์สามย่าน.com, พิมพ์ครั้งที่1
- [4] ยุทธนา ติลาวัฒนกุล , สัจจะ จรัสรุ่งรวีร บรรณาธิการ : “คู่มือการเขียนโปรแกรมและใช้งาน Visual C++.net ฉบับสมบูรณ์ “ , สำนักพิมพ์อินโฟเพรส , พิมพ์ครั้งที่ 1/2546
- [5] Matthew Reynolds, Richard Blair, Jonathan Crossland, Thearon Willis : “คัมภีร์การใช้ Visual Basic .NET ฉบับสมบูรณ์ “ , สำนักพิมพ์ บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) , พิมพ์ครั้งที่ 1/2545
- [6] สัจจะ จรัสรุ่งรวีร : “Visaul Basic.Net ฉบับสมบูรณ์” , สำนักพิมพ์ บริษัท ด้านสุทธาการ พิมพ์ จำกัด , พิมพ์ครั้งที่1/2545