

การพัฒนาเว็บเซอร์วิสแอปพลิเคชัน โดยใช้ดอทเน็ตเทคโนโลยี 1
Web Service Application Development by .NET Technology 1



นาย พิสุทธิพงษ์ แสงอ่อน
นาย เมธา ไพบูลย์ภิญญาเลิศ



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....

เลขทะเบียน.....49979

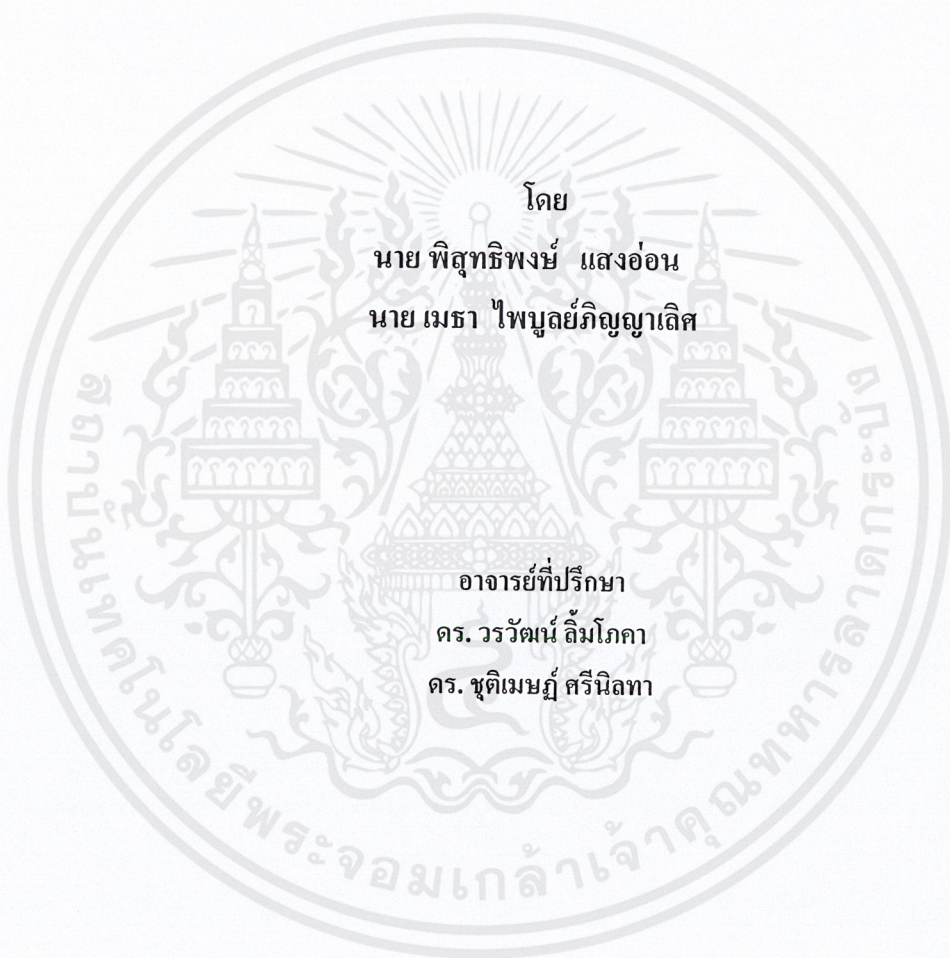
วัน,เดือน,ปี.16 เม.ย. 2547

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเว็บเซอร์วิสแอปพลิเคชัน โดยใช้ดีเอ็นทีเทคโนโลยี 1

Web Service Application Development by .NET Technology 1



โดย

นาย พิสุทธิพงษ์ แสงอ่อน

นาย เมธา ไพบุญย์ภิญญาเลิศ

อาจารย์ที่ปรึกษา

ดร. วรวัฒน์ ลิ้มโกคา

ดร. ชุตติเมษฎ์ ศรีนิลทา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

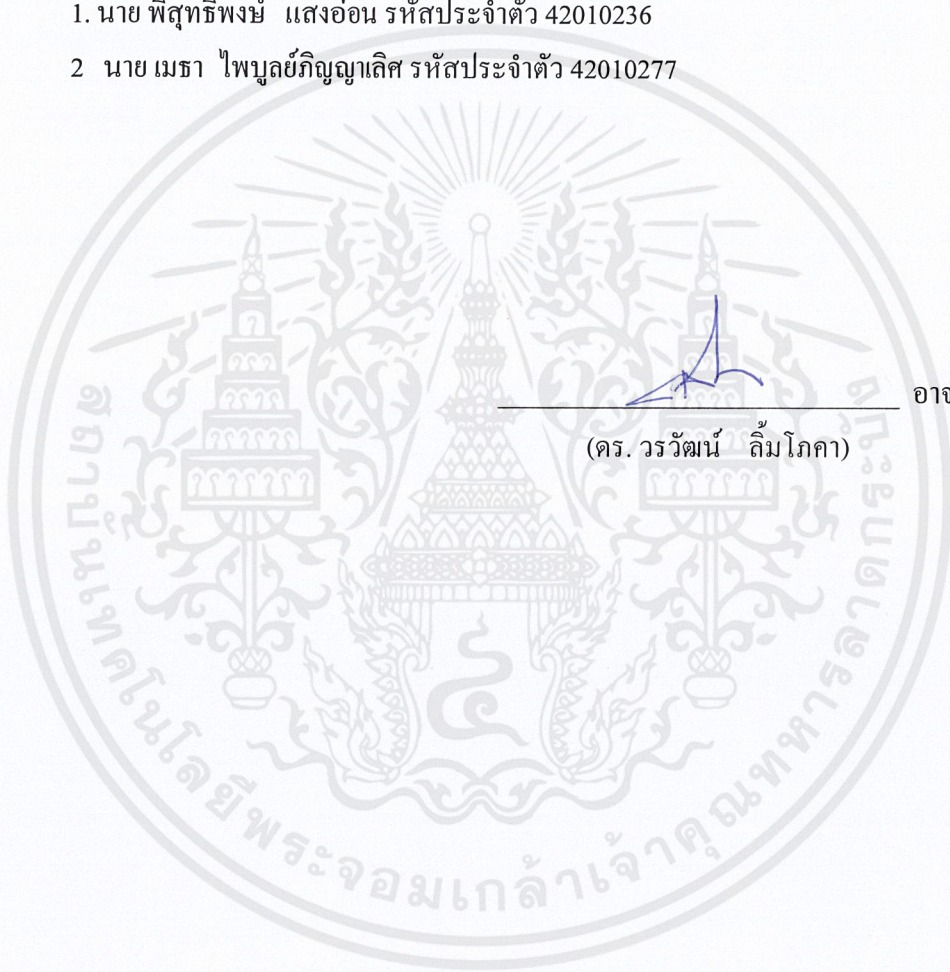
เรื่อง การพัฒนาเว็บเซอร์วิสแอปพลิเคชัน โดยใช้ดีเอ็นทีเทคโนโลยี 1

Web Service Application Development by .NET Technology 1

ผู้จัดทำ

1. นาย พิสุทธิพงษ์ แสงอ่อน รหัสประจำตัว 42010236

2. นาย เมธา ไพบุลย์ภิญญาเลิศ รหัสประจำตัว 42010277



(ดร. วรวัฒน์ ตี๋มโกคา)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเว็บเซอร์วิสแอปพลิเคชัน โดยใช้ค็อดเน็ตเทคโนโลยี

พิศุทธิพงษ์ แสงอ่อน

เมธา ไพบูลย์ภิญญาเลิศ

ดร. วรวัฒน์ ถิรมโกศา อาจารย์ที่ปรึกษา

ดร. ชุติเมษย์ ศรีนิลทา อาจารย์ที่ปรึกษา

ปีการศึกษา 2545

บทคัดย่อ

ในปัจจุบันซอฟต์แวร์ที่พัฒนาขึ้นมามีปัญหาทางด้าน การแก้ไขข้อบกพร่อง การบำรุงรักษา รวมไปถึงความน่าเชื่อถือในการใช้งาน ถึงแม้ว่าจะมีการปรับเปลี่ยนมาเป็นการพัฒนาเป็นการพัฒนาเชิงวัตถุแล้วก็ตาม แต่ยังไม่สามารถตอบสนองความต้องการพัฒนาซอฟต์แวร์ที่ต้องการความรวดเร็วในการพัฒนา ซึ่งทำให้เกิดแนวความคิดใหม่ของการนำกลับมาใช้ใหม่ ซึ่งมีประสิทธิภาพสูงกว่า และเชื่อถือได้มากกว่ารูปแบบเดิมเทคโนโลยี .NET เป็นเทคโนโลยีใหม่ที่มีความสามารถในการรองรับการพัฒนาแอปพลิเคชันเชิงคอมโพเนนต์ ทั้งในรูปแบบของ .NET คอมโพเนนต์และ เว็บเซอร์วิส ซึ่งเว็บเซอร์วิสช่วยให้ซอฟต์แวร์ที่พัฒนาสามารถที่จะเรียกใช้งานข้ามแพลตฟอร์มได้ นอกจากนี้ยังรวมถึงการเรียกใช้งานบริการต่างๆที่มีมาพร้อมใน .NET เฟรมเวิร์ก เช่น การจัดการทรานแซกชัน การจัดการด้านความปลอดภัย การเรียกใช้เมสเสจคิว เพื่อเพิ่มประสิทธิภาพให้แก่ระบบ และในที่สุดท้ายที่ทำการเปรียบเทียบเว็บเซอร์วิสที่พัฒนาจากทั้งสองแพลตฟอร์ม

ตัวอย่างการพัฒนาระบบงานที่ปฏิญาณพันธันฉบับนี้นำเสนอคือ ระบบการจองงานถ่ายรูป และระบบสั่งของข้าวสวย ซึ่งใช้หลักการพัฒนาเชิงคอมโพเนนต์ ตามโครงสร้างของ เอเอสพีค็อดเน็ต เว็บเซอร์วิส(ASP.NET Web Services) โดยจะระบบตรงกลางเป็นระบบ รับจัดงานแต่งงาน ซึ่งพัฒนาโดยเทคโนโลยี J2EE เว็บเซอร์วิส ทำการเรียกใช้เซอร์วิสจากระบบโรงแรม ระบบงานถ่ายรูป ระบบสั่งจองของข้าวสวย ซึ่งพัฒนาโดยค็อดเน็ต และเรียกใช้ ระบบตัวร่วมกับธนาคารที่พัฒนาโดย J2EE

Web Services Application Development by .NET Technology

Pisuttipong Sangon

Metha Paiboonpinyalert

Dr. Worawat Limpoka Advisor

Dr. Shutimet Srinilta Advisor

ABSTRACT

Recently, software development has a problem such as updating application, maintenance and reliability. Object-oriented development cannot solve these problems completely. It also cannot meet requirement of rapid software development. So the concept of “Reusability” that is more efficiency and reliability than legacy system was used.

.NET Technology is a new technology that support component based application and Web services, provided ability to call services across platforms. Additionally, Enterprise services(COM+) in .NET Framework has other services that help developer can develop easier such as Microsoft Transaction Server (MTS), Security and Microsoft Message Queue (MSMQ) in order to scalability web services .Comparison of implementation web services from two platforms .NET and Java is also included in this thesis.

This thesis is study in developing system by Component Based Development Methodology using .NET Web services. The central application is wedding planner system that consist of many called services such as tour agency ,bank that are implemented by J2EE and three Web services by .NET ,Photo Studio , Hotel ,Gift Registries.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์วรวัฒน์ ลิ้ม โทคา และอาจารย์ชุตินเมษณ์ ศรีนิลทา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา นอกจากนั้นยังได้ให้ความรู้และแนวคิดในการทำงานรวมถึงการใช้ชีวิต อันมีค่าอย่างยิ่งแก่ผู้เขียน ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณที่ ๆ เพื่อนๆ ห้อง OLALA และพี่ๆปริญญาโท ที่ช่วยเป็นกำลังใจเป็นที่ปรึกษาและให้ประสบการณ์ในการใช้ชีวิตในมหาวิทยาลัยที่มีค่าอย่างยิ่งให้กับผู้เขียน

สุดท้ายนี้ ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอ กราบขอบพระคุณมา ณ ที่นี้

นายพิสุทธิพงษ์ แสงอ่อน

นายนายเมธา ไพบุญย์ภิญญาเลิศ

สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | X |
| สารบัญรูป | XI |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความเป็นมาของโครงการ | 1 |
| 1.2 วัตถุประสงค์ของโครงการ | 1 |
| 1.3 ขอบเขตของโครงการ | 2 |
| 1.4 วิธีการดำเนินโครงการ | 2 |
| บทที่ 2 ทฤษฎีคอมพิวเตอร์ | 3 |
| 2.1 การพัฒนาเชิงคอมพิวเตอร์ | 3 |
| 2.1.1 เฟรมเวิร์ก | 4 |
| 2.1.2 ส่วนประกอบของคอมพิวเตอร์ | 4 |
| 2.1.3 คุณสมบัติของคอมพิวเตอร์ | 4 |
| 2.2 งานที่เหมาะสมกับการพัฒนาเชิงคอมพิวเตอร์ | 7 |
| 2.3 ประโยชน์ของการพัฒนาเชิงคอมพิวเตอร์ | 7 |
| 2.4 แนวโน้มของระบบในอนาคต | 7 |
| บทที่ 3 วิวัฒนาการของสถาปัตยกรรมแบบกระจาย | 8 |
| 3.1 สถาปัตยกรรมแบบหนึ่ง-tier (One Tier Architecture/Centralize Host) | 8 |
| 3.2 เครือข่ายภายในและการแบ่งปันทรัพยากรภายในเครือข่าย | 8 |
| 3.3 สถาปัตยกรรมแบบสอง-tier (ไคลเอนต์/เซิร์ฟเวอร์) | 9 |
| 3.3.1 ดาต้าเบสเซิร์ฟเวอร์ (Database Server) | 9 |
| 3.3.2 แอปพลิเคชันเซิร์ฟเวอร์ | 9 |
| 3.4 สถาปัตยกรรมแบบสาม-tier | 9 |
| 3.4.1 วินโดวส์ดีเอ็นเอ (Windows DNA) | 10 |
| 3.4.1.1 ฟริเอนต์-เอนด์-ทีเยอร์ | 10 |
| 3.4.1.2 บิสิเนส โลจิกทีเยอร์ (Business Logic Tier) | 10 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | หน้า |
|--|------|
| 3.4.1.3 ดาต้าแอ็กเซสเลเยอร์ (Data Access Layer) | 11 |
| 3.5 สถาปัตยกรรมแบบหลายเทียร์(N-Tier) | 11 |
| 3.5.1 แอปพลิเคชันแบบหลายเทียร์ทางตรง | 11 |
| 3.5.2 แอปพลิเคชัน N-Tier โดยใช้ XML เว็บเซอร์วิส | 12 |
| 3.5.3 แอปพลิเคชัน N-Tier โดยใช้รีโมต .NET | 14 |
| บทที่ 4 เว็บเซอร์วิส | 15 |
| 4.1 XML คืออะไร | 16 |
| 4.1.1 แนะนำ XML | 16 |
| 4.1.2 ไวยากรณ์ของ XML | 16 |
| 4.1.3 XML Element | 18 |
| 4.1.4 XML Attribute | 21 |
| 4.1.5 XML Validation | 22 |
| 4.1.6 แนะนำ DTD | 23 |
| 4.1.6.1 โครงสร้างของ XML | 25 |
| 4.1.6.2 การกำหนด Element ใน DTD | 25 |
| 4.1.6.3 การกำหนด Attribute ใน DTD | 28 |
| 4.1.6.4 การกำหนด Entity ใน DTD | 32 |
| 4.1.7 XML Namespace | 33 |
| 4.1.8 XML PCDATA and CDATA | 36 |
| 4.1.8.1 PCDATA | 36 |
| 4.1.8.2 CDATA | 37 |
| 4.1.9 XML Parser | 38 |
| 4.1.9.1 DOM | 38 |
| 4.1.9.2 SAX | 39 |
| 4.1.10 ประโยชน์จาก XML | 40 |
| 4.1.11 บทสรุปของ XML | 41 |
| 4.2 SOAP | 41 |
| 4.2.1 แนะนำ SOAP | 41 |
| 4.2.2 ส่วนประกอบของ SOAP | 42 |

| | หน้า |
|--|------|
| 4.2.2.1 Envelope | 44 |
| 4.2.2.2 Body | 44 |
| 4.2.2.3 Header | 44 |
| 4.2.3 SOAP Fault Element | 45 |
| 4.2.4 SOAP Encoding | 47 |
| 4.2.4.1 ตัวอย่าง Struct | 47 |
| 4.2.4.2 ตัวอย่าง Array | 49 |
| 4.2.5 SOAP ใน HTTP | 50 |
| 4.2.6 SOAP สำหรับ RPC | 51 |
| 4.2.7 SOAP Toolkit | 52 |
| 4.2.8 โครงสร้างภายในของ SOAP Toolkit ของ Microsoft | 53 |
| 4.2.8.1 การสร้าง SoapClientObject | 54 |
| 4.2.8.2 การประมวลผลภายใน SoapClientObject | 55 |
| 4.2.8.3 การสร้าง SoapServerObject | 56 |
| 4.2.8.4 การประมวลผลภายใน SoapSeverObject | 57 |
| 4.2.9 การสร้าง SOAP Client และ SOAP Server ด้วย .NET Framework | 58 |
| 4.2.9.1 การใช้ ASP.NET ทำ SOAP Web Service | 58 |
| 4.2.9.2 การใช้ .NET ทำ SOAP Client | 58 |
| 4.3 UDDI | 59 |
| 4.4 WSDL | 59 |
| บทที่ 5 Microsoft.NET Technology | 62 |
| 5.1.NET PLATFORM | 62 |
| 5.1.1 ชั้นระบบปฏิบัติการ(Operating System) | 62 |
| 5.1.2 ชั้นถัดมาแบ่งออกเป็น 3 ส่วน ได้แก่ | 62 |
| 5.1.3 ชั้น Visual Studil.NET เครื่องมือในการพัฒนา.NET | 62 |
| 5.2 .NETFramework | 62 |
| 5.2.1 Common Language Runtime (CLR) | 64 |
| 5.2.1.1 Memory Management | 65 |
| 5.2.1.2 Common Type System | 65 |
| 5.2.2 .NET Framework Class Library | 67 |

| | หน้า |
|--|------|
| 5.2.3 Presentation Layer | 67 |
| 5.3 .NET Compact Framework | 67 |
| 5.4 .NET Framework on Linux | 68 |
| 5.4.1 .NET Framework Runtime (CLR) | 68 |
| 5.4.2 NET Framework based class Library | 68 |
| 5.4.3 C#Compiler | 69 |
| 5.5 มาตรฐาน CLI และ C# ของ ECMA | 72 |
| บทที่ 6 คอมโพเนนต์ .NET (Assembly) | 73 |
| 6.1 วิวัฒนาการของคอมโพเนนต์ | 73 |
| 6.2 Assemblies | 74 |
| 6.2.1 Unique Identities | 74 |
| 6.2.2 Working .NET Component | 75 |
| 6.2.3 Deployment | 78 |
| 6.2.3.1 Private Components | 78 |
| 6.2.3.2 Shared Components | 79 |
| 6.2.4 ขั้นตอนการลงทะเบียนคอมโพเนนต์ | 80 |
| 6.2.5 Building and testing the drive.exe | 81 |
| 6.3 Enterprise Services (COM+) in .NET | 83 |
| 6.3.1 Transaction | 83 |
| 6.3.2 Lazy registration | 84 |
| 6.4 .NET Interoperability | 85 |
| 6.4.1 การเรียก COM คอมโพเนนต์จาก .NET | 85 |
| 6.4.2 การเรียก .NET คอมโพเนนต์จาก COM | 86 |
| 6.5 .NET Componet VS COM | 89 |
| บทที่ 7 แอปพลิเคชัน | 90 |
| 7.1 Overview ของระบบ OLALA Wedding Planner | 90 |
| 7.2 รายละเอียดของแอปพลิเคชัน | 92 |
| 7.2.1ระบบสตูดิโอถ่ายภาพ | 92 |
| 7.2.2 ระบบร้านของชำร่วย | 92 |
| 7.3 ER diagram | 94 |

| | หน้า |
|--|------|
| 7.3.1 ER diagram ของระบบสตูดิโอถ่ายภาพ | 94 |
| 7.3.2 ER diagram ของระบบร้านของชำร่วย | 95 |
| 7.4 Use case diagram | 96 |
| 7.4.1 Use case diagram ระบบสตูดิโอถ่ายภาพ | 96 |
| 7.4.2 Use case diagram ของระบบร้านของชำร่วย | 97 |
| 7.5 Sequence diagram | 98 |
| 7.5.1 Sequence diagram ระบบสตูดิโอถ่ายภาพ | 98 |
| 7.5.1.1 ระบบสมาชิก | 98 |
| 7.5.1.2 เข้าสู่ระบบ | 99 |
| 7.5.1.3 สร้าง contact ใหม่ | 100 |
| 7.5.1.4 เรียกดู contact ทั้งหมด | 101 |
| 7.5.1.5 ค้นหาบริการจากช่วงราคา | 102 |
| 7.5.1.6 ตรวจสอบวันเวลาที่ว่างของแพ็คเกจที่ต้องการจากวันเวลาที่ต้องการ | 103 |
| 7.5.1.7 เรียกดูตารางเวลาล่วงหน้าตามจำนวนวันที่ต้องการของบริการที่ต้องการ | 104 |
| 7.5.1.8 จองเวลาถ่ายภาพในสตูดิโอ | 105 |
| 7.5.1.9 จองเวลาถ่ายภาพนอกสถานที่ | 106 |
| 7.5.1.10 จองเวลาถ่ายภาพทำวีดีโอมีลคิมิเดียฟรีเซนเตอร์ | 107 |
| 7.5.1.11 ดูข้อมูลการจองบริการ | 108 |
| 7.5.1.12 เปลี่ยนแปลงเวลาจองบริการ | 109 |
| 7.5.1.13 ชำระเงินผ่านบัตรเครดิต | 110 |
| 7.5.1.14 ยกเลิกการจองที่รายการ | 111 |
| 7.5.1.15 ยกเลิกการจองทั้ง contact | 112 |
| 7.5.2 Sequence diagram ระบบร้านของชำร่วย | 113 |
| 7.5.2.1 สมัครสมาชิก | 113 |
| 7.5.2.2 เข้าสู่ระบบ | 114 |
| 7.5.2.3 สร้าง contact ใหม่ | 115 |
| 7.5.2.4 ค้นหาสินค้าตามประเภทและรูปแบบ | 116 |
| 7.5.2.5 สั่งจองสินค้า | 117 |
| 7.5.2.6 ดูข้อมูลการจองสินค้า | 118 |
| 7.5.2.7 ชำระเงินผ่านบัตรเครดิต | 119 |

| | หน้า |
|---|------|
| 7.5.2.8 เปลี่ยนสถานที่และเวลาส่งสินค้า | 120 |
| 7.5.2.9 ยกเลิกการจองที่ละรายการ | 121 |
| 7.6 Class diagram | 123 |
| 7.6.1 Class diagram ของระบบสตูดิโอถ่ายภาพ | 124 |
| 7.6.2 Class diagram ของระบบร้านขายของชำร่วย | 125 |
| 7.7 component diagram | 126 |
| 7.7.1 component diagram ระบบสตูดิโอถ่ายภาพ | 126 |
| 7.7.2 component diagram ของระบบร้านขายของชำร่วย | 126 |
| 7.8 คู่มือการใช้งานระบบร้านขายของชำร่วย | 127 |
| 7.8.1 ส่วนบริการสมาชิก | 127 |
| 7.8.1.1 การสมัครสมาชิก | 127 |
| 7.8.1.2 การล็อกอินเข้าสู่ระบบ | 128 |
| 7.8.1.3 การสร้าง contact ID ใหม่ | 129 |
| 7.8.1.4 การออกจากระบบ | 130 |
| 7.8.2 การตั้งสินค้า | 130 |
| 7.8.3 การค้นหาสินค้าตามเงื่อนไขที่ต้องการ | 132 |
| 7.8.4 การดูข้อมูลการจอง | 133 |
| 7.8.5 การยกเลิกการจอง | 134 |
| 7.8.5.1 การยกเลิกที่ละรายการจอง | 134 |
| 7.8.5.2 การยกเลิกการจองทั้ง contact | 136 |
| 7.8.6 การเปลี่ยนแปลงข้อมูลการสั่งซื้อสินค้า | 136 |
| 7.8.7 การชำระเงินผ่านบัตรเครดิต | 138 |
| 7.9 คู่มือการใช้งานสตูดิโอถ่ายภาพ | 138 |
| 7.9.1 ส่วนบริการสมาชิก | 139 |
| 7.9.1.1 การสมัครสมาชิก | 139 |
| 7.9.1.2 การล็อกอินเข้าสู่ระบบ | 140 |
| 7.9.2 การค้นหาสินค้าตามเงื่อนไขที่ต้องการ | 140 |
| บทที่ 8 บทวิจารณ์และสรุป | 142 |
| 8.1 สรุปผลและการดำเนินงาน | 142 |
| 8.2 แนวทางการพัฒนาต่อ | 143 |

สารบัญตาราง

| ตาราง | หน้า |
|--|------|
| 4-1 ชนิดของ Attribute ใน XML | 29 |
| 4-2 ค่าดีฟอลต์ของ Attribute ใน XML | 30 |
| 4-3 Entity Reference มาตรฐาน | 37 |
| 4-4 XML Parser | 41 |
| 4-5 SOAP Fault Element | 47 |
| 4-6 ค่า faultcode | 47 |
| 4-7 SOAP Toolkit | 53 |
| 6-1 ตารางเปรียบเทียบระหว่าง .NET Component กับ COM | 89 |



สารบัญภาพ

| ภาพที่ | หน้า |
|--|------|
| 2-1 คอมโปเนนต์และส่วนประกอบต่างๆของมัน | 4 |
| 2-2 ตัวอย่างของอินเทอร์เฟซ | 5 |
| 2-3 คอมโปเนนต์และการแทนที่ | 6 |
| 2-4 การเพิ่มเติมคอมโปเนนต์ | 6 |
| 3-1 สถาปัตยกรรมแบบหนึ่งเทียร์ | 8 |
| 3-2 เครือข่ายภายในของเครื่องคอมพิวเตอร์ส่วนบุคคล | 8 |
| 3-3 สถาปัตยกรรมแบบ N-Tier | 11 |
| 3-4 สถาปัตยกรรมแบบ N-Tier ผ่านเว็บเซอร์วิส | 13 |
| 3-5 สถาปัตยกรรมแบบ N-Tier ผ่าน .NET Remoting | 14 |
| 4-1 โครงสร้างของ DOM | 39 |
| 4-2 ความสัมพันธ์ของการ parse ด้วย DOM | 40 |
| 4-3 โครงสร้างของ SOAP | 43 |
| 4-4 แสดงการ Client-Side Data Flow | 54 |
| 4-5 แสดงโครงสร้างภายในของ SOAP | 55 |
| 4-6 แสดง Sever-Side Data Flow | 57 |
| 4-7 แสดงโครงสร้างภายใน SOAP Server | 57 |
| 4-8 การใช้ ASP.NET ทำ SOAP web service | 58 |
| 4-9 Runtime Callable Wrapper | 58 |
| 4-10 การใช้ .NET ทำ SOAP client | 59 |
| 4-11 COM callable wrapper ที่ทำให้ COM client สามารถเห็น .NET object | 59 |
| 5-1 องค์ประกอบ .NET Platform | 62 |
| 5-2 องค์ประกอบ .NET Framework | 63 |
| 5-3 รูปแบบการคอมไพล์โค้ดไปเป็น IL Code | 64 |
| 5-4 Platano Media Player Startup | 69 |
| 5-5 Platano playing the Code Monkey At Work Video | 70 |
| 5-6 GSIRC is shown with the preferences dialog and chatting on some channels (rooms) | 70 |
| 5-7 SQL# For GTK# แสดงการดึงข้อมูลจาก Microsoft SQL Server 2000 database | 71 |
| 5-8 ตัวอย่างแอปพลิเคชัน SkyNet | 71 |
| 6-1 การทำ Digital Signature | 75 |

| | หน้า |
|--|------|
| 6-2 Multiple-directory tree of component | 79 |
| 6-3 แชร์คอมโพเนนต์ที่เก็บใน GAC | 80 |
| 6-4 แสดงการทำงานของ Lazy Registration | 85 |
| 6-5 แสดงการเรียก COM คอมโพเนนต์จาก .NET | 85 |
| 6-6 แสดงการเรียก .NET คอมโพเนนต์จาก COM | 86 |
| 6-7 in-process activation | 87 |
| 6-8 cross-process activation | 88 |
| 7-1 การศึกษาการเรียกใช้คอมโพเนนต์ข้ามสถาปัตยกรรม | 91 |
| 7-2 ER Diagram ระบบร้านของชำร่วย | 94 |
| 7-3 ER Diagram ระบบสตูดิโอถ่ายภาพ | 95 |
| 7-4 Use Case Diagram ระบบร้านของชำร่วย | 96 |
| 7-5 Use Case Diagram ระบบสตูดิโอถ่ายภาพ | 97 |
| 7-6 Sequence Diagram การสมัครสมาชิก ของระบบสตูดิโอถ่ายภาพ | 98 |
| 7-7 Sequence Diagram การเข้าสู่ระบบ ของระบบสตูดิโอถ่ายภาพ | 99 |
| 7-8 Sequence Diagram การสร้าง contact ใหม่ ของระบบสตูดิโอถ่ายภาพ | 100 |
| 7-9 Sequence Diagram การเรียกดู contact ทั้งหมด ของระบบสตูดิโอถ่ายภาพ | 101 |
| 7-10 Sequence Diagram การค้นหาบริการของระบบสตูดิโอถ่ายภาพ จากช่วงเวลาที่ต้องการจอง | 102 |
| 7-11 Sequence Diagram การตรวจสอบวันเวลาว่างของแพคเกจ ของระบบสตูดิโอถ่ายภาพ | 103 |
| 7-12 Sequence Diagram การเรียกดู ตารางจองเวลาล่วงหน้า ของระบบสตูดิโอถ่ายภาพ | 104 |
| 7-13 Sequence Diagram การจองเวลาถ่ายภาพในสตูดิโอ ของระบบสตูดิโอถ่ายภาพ | 105 |
| 7-14 Sequence Diagram การจองเวลาถ่ายภาพนอกสถานที่ ของระบบสตูดิโอถ่ายภาพ | 106 |
| 7-15 Sequence Diagram การจองเวลาถ่ายทำวิดีโออัตโนมัติฟรีเซนเตอร์ ของระบบสตูดิโอถ่ายภาพ | 107 |
| 7-16 Sequence Diagram การดูข้อมูลการจองบริการของระบบสตูดิโอถ่ายภาพ | 108 |
| 7-17 Sequence Diagram การเปลี่ยนแปลงเวลาจองบริการ ของระบบสตูดิโอถ่ายภาพ | 109 |
| 7-18 Sequence Diagram การชำระเงินผ่านบัตรเครดิต ของระบบสตูดิโอถ่ายภาพ | 110 |
| 7-19 Sequence Diagram การยกเลิกการจองที่รายการของระบบสตูดิโอถ่ายภาพ | 111 |
| 7-20 Sequence Diagram การยกเลิกการจองทั้ง contact ของระบบสตูดิโอถ่ายภาพ | 112 |
| 7-21 Sequence Diagram การสมัครสมาชิก ของระบบร้านขายของชำร่วย | 113 |
| 7-22 Sequence Diagram การเข้าสู่ระบบร้านขายของชำร่วย | 114 |
| 7-23 Sequence Diagram การสร้าง contact ใหม่ ของระบบร้านขายของชำร่วย | 115 |

| | หน้า |
|--|------|
| 7-24 Sequence Diagram การค้นหาสินค้า ของระบบร้านขายของชำร่วยตามประเภท | 116 |
| 7-25 Sequence Diagram การตั้งจองสินค้า ของระบบร้านขายของชำร่วย | 117 |
| 7-26 Sequence Diagram การดูข้อมูลการจองสินค้า ของระบบร้านขายของชำร่วย | 118 |
| 7-27 Sequence Diagram การชำระเงินผ่านบัตรเครดิต ของระบบร้านขายของชำร่วย | 119 |
| 7-28 Sequence Diagram การเปลี่ยนแปลงสถานที่และเวลาส่งสินค้า ของระบบร้านขายของชำร่วย | 120 |
| 7-29 Sequence Diagram การยกเลิกการจองที่ละรายการ ของระบบร้านขายของชำร่วย | 121 |
| 7-30 Sequence Diagram การยกเลิกทั้ง contact ของระบบร้านขายของชำร่วย | 122 |
| 7-31 Class Diagram ระบบสตูดิโอถ่ายภาพ ส่วนที่1 | 123 |
| 7-32 Class Diagram ระบบสตูดิโอถ่ายภาพ ส่วนที่2 | 124 |
| 7-33 C lass Diagram ระบบร้านขายของชำร่วย | 125 |
| 7-34 Component Diagram ของระบบสตูดิโอถ่ายภาพ | 126 |
| 7-35 Component Diagram ของระบบร้านขายของชำร่วย | 126 |
| 7-36 แสดงหน้าการสมัครสมาชิก | 127 |
| 7-37 แสดงส่วนการล็อกอินเข้าสู่ระบบ | 128 |
| 7-38 แสดงรายการต่างๆที่สมาชิกสามารถใช้งานได้ | 129 |
| 7-39 แสดงหน้าจอในการขอ contact ใหม่ | 129 |
| 7-40 แสดงหน้าเลือกสินค้าที่ต้องการสั่งซื้อ | 130 |
| 7-41 แสดงหน้ากรอกรายละเอียดการซื้อสินค้า | 131 |
| 7-42 หน้าจอแสดงการสั่งซื้อสินค้าเรียบร้อยแล้ว | 131 |
| 7-43 แสดงหน้ากรอกรายละเอียดการค้นหาสินค้า | 132 |
| 7-44 แสดงหน้าผลการค้นหาสินค้า | 132 |
| 7-45 แสดงหน้ารายการ contact ทั้งหมดของผู้ใช้ | 133 |
| 7-46 แสดงการเลือกรายการย่อยที่ต้องการดูข้อมูลการจอง | 133 |
| 7-47 แสดงหน้ารายละเอียดการสั่งซื้อสินค้าของรายการจองที่เลือก | 134 |
| 7-48 แสดงหน้าการเลือก edit ของcontact ที่ต้องการแก้ไขข้อมูลการสั่งซื้อ | 134 |
| 7-49 แสดงการเลือกคำสั่ง edit ของรายการจองที่ต้องการยกเลิก | 135 |
| 7-50 แสดงการเลือกคำสั่ง cancel เพื่อทำการยกเลิกรายการสั่งซื้อสินค้า | 135 |
| 7-51 แสดงการเลือกคำสั่ง cancel ของ contact ที่ต้องการเพื่อยกเลิกcontact นั้น | 136 |
| 7-52 แสดงการเลือกคำสั่ง edit ของรายการสั่งซื้อสินค้าที่ต้องการแก้ไขข้อมูลการสั่งซื้อ | 137 |
| 7-53 แสดงหน้าเปลี่ยนแปลงข้อมูลการสั่งซื้อสินค้า | 137 |

| | หน้า |
|---|------|
| 7-54 แสดงหน้าหลัก | 138 |
| 7-55 ตัวอย่างบริการที่มีให้ของเว็บ | 139 |
| 7-56 แสดงหน้าการสมัครสมาชิก | 139 |
| 7-57 แสดงการ login | 140 |
| 7-58 แสดงหน้ากรอกรายละเอียดการค้นหาสินค้า | 140 |
| 7-59 แสดงหน้าผลการค้นหาสินค้า | 141 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

การพัฒนาโปรแกรมคอมพิวเตอร์นั้นได้มีการพัฒนาจากการใช้การพัฒนาโดยรูปแบบโครงสร้าง (Structural Development) ไปเป็นการพัฒนาเชิงวัตถุ (Object Oriented Development) ซึ่งก็มีข้อดีที่สามารถจัดการกับระบบที่มีความซับซ้อนสูงได้อย่างมีประสิทธิภาพมากขึ้นแต่ในขณะเดียวกัน การพัฒนาเชิงวัตถุก็มีจุดบกพร่องอยู่เป็นจำนวนมาก เช่น จุดบกพร่องของหลักการสืบทอดคุณสมบัติ (Inheritance) การนำกลับมาใช้ใหม่ ซึ่งยังไม่มีประสิทธิภาพเพียงพอเป็นต้น ซึ่งทำให้เกิดแนวความคิดใหม่ของการนำกลับมาใช้ใหม่ ซึ่งมีประสิทธิภาพสูงกว่า และเชื่อถือได้มากกว่ารูปแบบเดิม การพัฒนาโดยสถาปัตยกรรม COM เป็นพื้นฐาน (COM based Architecture) เป็นอีกแนวทางหนึ่งซึ่งได้ถูกนำมาใช้

แต่ปัจจุบันแนวโน้มของการพัฒนาแอปพลิเคชันนั้นเป็นไปในรูปแบบที่เป็นแอปพลิเคชันกระจายมากขึ้น และระบบอินเทอร์เน็ตเข้ามามีบทบาทในการสื่อสารรับส่งข้อมูลมากขึ้น เทคโนโลยีที่น่าสนใจและเหมาะสมกว่าคือเว็บเซอร์วิสโดยโครงการฉบับนี้ได้นำเสนอทั้งทฤษฎี และตัวอย่างการสร้างเว็บเซอร์วิสโดยใช้ .NET เฟรมเวิร์กในการพัฒนา การย้ายการรองรับการใช้งานโดยใช้ MSMQ มีการใช้งานข้ามแพลตฟอร์มร่วมกันระหว่าง จาวาคีย์ และทำการเปรียบเทียบเทคโนโลยีทั้งสองค่าย

1.2 วัตถุประสงค์ของโครงการ

ภายในโครงการนี้ได้ทำการศึกษาถึงทฤษฎี การสร้าง และการนำ ASP.NET เว็บเซอร์วิสไปใช้งานโดยมีจุดประสงค์ดังนี้

1. ศึกษาทฤษฎีพื้นฐานของคอมโพเนนต์
2. เพื่อศึกษาทำความเข้าใจ .NET Technology และ Enterprise Service (COM+) ใน .NET Technology
3. ศึกษาเรื่องเว็บเซอร์วิส และเทคโนโลยีที่เกี่ยวข้อง ได้แก่ SOAP, XML, WSDL, UDDI เป็นต้น
4. ศึกษาการสร้างเว็บเซอร์วิส โดยใช้ .NET Technology ลงลึกในเทคโนโลยีของเว็บเซอร์วิส เช่น ทรานแซกชัน , การจัดการเซต , ความปลอดภัยในการเรียกใช้เว็บเซอร์วิส
5. ทำการขยายการรองรับการใช้งานของระบบ (Scalability) โดยใช้เทคโนโลยี MSMQ
6. ศึกษาการเรียกใช้เว็บเซอร์วิสข้ามค่าย รวมทั้งเปรียบเทียบเทคโนโลยีของทั้งสองค่าย .NET และ จาวา

1.3 ขอบเขตของโครงการงาน

1. ออกแบบและสร้างแอปพลิเคชันโดยใช้สถาปัตยกรรม แบบ 3 เทียร์ส ร่วมกับการใช้หลักการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ด้วยเทคโนโลยี .NET เว็บเซอร์วิส
2. ใช้ทฤษฎีของ MSMQ เพื่อเข้ามาเพิ่ม เพิ่ม Scalability ให้กับแอปพลิเคชัน
3. สร้างแอปพลิเคชันทำการเรียกใช้เว็บเซอร์วิสคอมโพเนนต์ข้ามแพลตฟอร์มได้
4. ระบบที่พัฒนาเป็นระบบอีคอมเมิร์ซที่เกี่ยวกับ การจัดการแต่งงาน (Wedding Planner) โดยใช้ .NET Technology โดยกลุ่มจะทำการเป็นเว็บเซอร์วิสที่เป็นระบบร้านค้าถ่ายรูป และร้านขายของชำร่วย โดยจะมีการ ติดต่อกับเว็บเซอร์วิสส่วนกลาง โดยส่วนกลางจะทำหน้าที่ติดต่อเว็บเซอร์วิสของแต่ละค่ายให้ทำงาน โดยนำผลลัพธ์ที่ได้จากแต่ละเว็บเซอร์วิส มาจัดการวางแผนการแต่งงานอีกครั้งหนึ่ง

1.4 วิธีการดำเนินงานของโครงการงาน

1. ศึกษาทฤษฎีพื้นฐานของ COM และ COM+
2. ศึกษาทฤษฎีของ XML และ SOAP และส่วนประกอบในเว็บเซอร์วิส
3. ศึกษาการพัฒนาคอมโพเนนต์และเว็บเซอร์วิสใน .NET
4. ออกแบบระบบแอปพลิเคชันในลักษณะงานเชิงคอมโพเนนต์
5. ศึกษาการทำงานของ MSMQ ที่มีอยู่ใน Windows 2000 Advanced Server
6. ศึกษาการปรับแต่งเพื่อให้เว็บเซอร์วิสมีความปลอดภัยในการสื่อสาร
7. ศึกษาการใช้งานร่วมกันรวมทั้งเปรียบเทียบเทคโนโลยีจากทั้งสองค่าย .NET และ จาวา

บทที่ 2

ทฤษฎีคอมโพเนนต์

2.1 การพัฒนาเชิงคอมโพเนนต์

คอมโพเนนต์ คือ ส่วนย่อยของระบบที่ไม่ขึ้นอยู่กับส่วนอื่นๆ และถูกซ่อนรายละเอียดไว้ภายใน ซึ่งหน้าที่ของคอมโพเนนต์จะถูกกำหนดไว้อย่างเหมาะสมเท่าที่คลเ็นต์ต้องการใช้งานเท่านั้น ส่วนการจำกัดหน้าที่บางอย่างของคอมโพเนนต์นั้นจะกระทำโดยสถาปัตยกรรม (Architecture) หรือ คอนเทนเนอร์ (container) ที่คอมโพเนนต์นั้นดีพลอย (deploy) อยู่

คอมโพเนนต์อาจพัฒนา ทดสอบ และดีพลอยในสภาพแวดล้อมที่แยกจากส่วนอื่นๆ ของระบบ ถึงแม้ว่าในความเป็นจริงนั้น คอมโพเนนต์จะถูกใช้งานร่วมกับคอมโพเนนต์อื่นๆ ก็ตาม แต่การทำแบบนี้มีความสำคัญต่อการทดสอบและคุณภาพของคอมโพเนนต์

เราอาจพิจารณาว่าส่วนใดๆ ของระบบซอฟต์แวร์เป็นคอมโพเนนต์ได้ เช่น ออบเจกต์ เมธอด เพราะระบบถูกสร้างขึ้นจากการประกอบของส่วนต่างๆ นี้ แต่การพิจารณาแบบนี้ไม่เหมาะกับการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ เพราะหมายความว่าผู้พัฒนาจะต้องเกี่ยวข้องกับคอมโพเนนต์เสมอ

เพื่อความหมายที่สมบูรณ์ของคอมโพเนนต์จะต้องพูดถึงอินเทอร์เฟซ เพราะคลเ็นต์จะใช้อินเทอร์เฟซในการทำงานกับคอมโพเนนต์

การพัฒนาเชิงคอมโพเนนต์นั้น มีความหมายกับคนแต่ละกลุ่มแตกต่างกันไปตามความต้องการ ซึ่งมีลักษณะเด่นบางอย่างที่นิยามความเป็นระบบคอมโพเนนต์ดังต่อไปนี้

1. คอมโพเนนต์มีลักษณะของออบเจกต์ คือ การซ่อนรายละเอียด (Encapsulated), โพลิมอร์ฟิก (polymorphic), การกำหนดหน้า, การกำหนดอินเทอร์เฟซ
2. คอมโพเนนต์ออกแบบภายใต้เฟรมเวิร์ก (Framework) ซึ่งได้สร้างข้อจำกัดบางอย่างไว้ เช่น ต้องไม่มีหลายเธรด, ไม่มีการติดต่อกับภายนอกโดยไม่ผ่านบริการของเฟรมเวิร์ก
3. คอมโพเนนต์สามารถอยู่ได้โดยไม่พึ่งพาคอมโพเนนต์อื่นๆ ยกเว้นคอมโพเนนต์ของเฟรมเวิร์กที่คอมโพเนนต์ดังกล่าวใช้อยู่
4. ทุกคอมโพเนนต์มีอินเทอร์เฟซสามัญที่แน่นอน (Fix and common) และอินเทอร์เฟซนี้เปลี่ยนแปลงไม่ได้
5. คอมโพเนนต์สามารถอธิบายตนเองได้ โดยอินเทอร์เฟซของคอมโพเนนต์จะต้องมีข้อมูลมากพอที่สามารถทำให้คลเ็นต์สามารถเข้าใจวิธีใช้คอมโพเนนต์นั้นได้

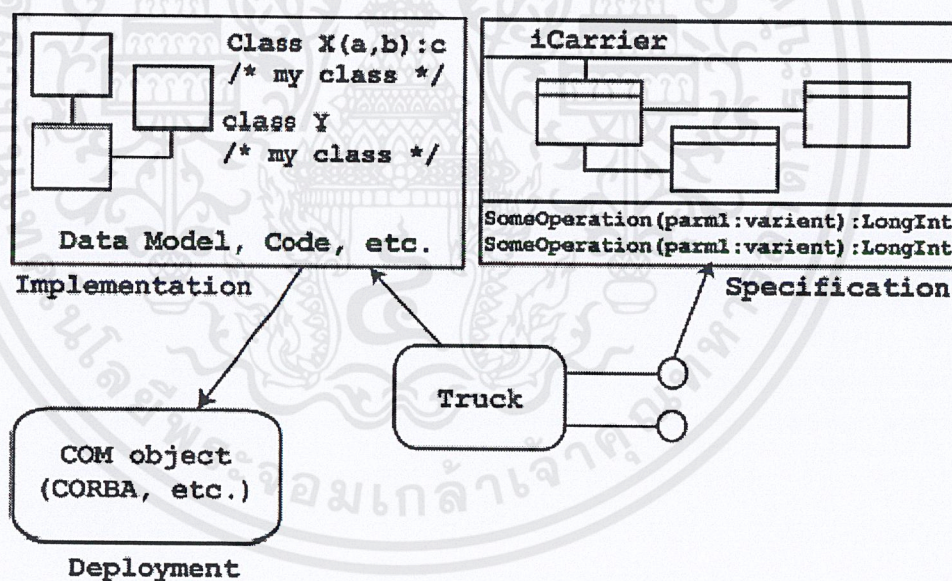
2.1.1 เฟรมเวิร์ก คือ สภาพแวดล้อมที่มีลักษณะดังนี้

1. สร้างอินสแตนซ์ของคอมโพเนนต์คอนรันไทม์
2. ทำให้คอมโพเนนต์ค้นพบคอมโพเนนต์อื่นๆ ได้
3. ทำให้คอมโพเนนต์ต่างๆ สามารถติดต่อกันได้
4. จัดหาบริการสามัญ เช่น เพอร์ซิสเทนซ์ (persistence), ทรานส์แอ็กชัน (transaction), ความไม่ขึ้นกับสถานที่, การรักษาความปลอดภัย, มอนิเตอร์ริง (monitoring)

2.1.2 ส่วนประกอบของคอมโพเนนต์

คอมโพเนนต์จะประกอบด้วยส่วนสำคัญ 3 ส่วนดังนี้

- อินเทอร์เฟซ (interface) - คอมโพเนนต์จะถูกเรียกใช้งานผ่านทางอินเทอร์เฟซ
- อิมพลีเมนต์เตชัน (implementation) - เป็นโค้ดที่กำหนดการทำงานของคอมโพเนนต์
- ดีพลอยเมนต์ (deployment) - เป็นเอ็กซิคิวต์ไฟล์จะใช้ในการทำให้คอมโพเนนต์ทำงานได้ ทำหน้าที่จัดหารันไทม์เอ็นไวรอนเมนต์ในการควบคุมการทำงานของคอมโพเนนต์และจัดหาเซอร์วิสที่จำเป็น



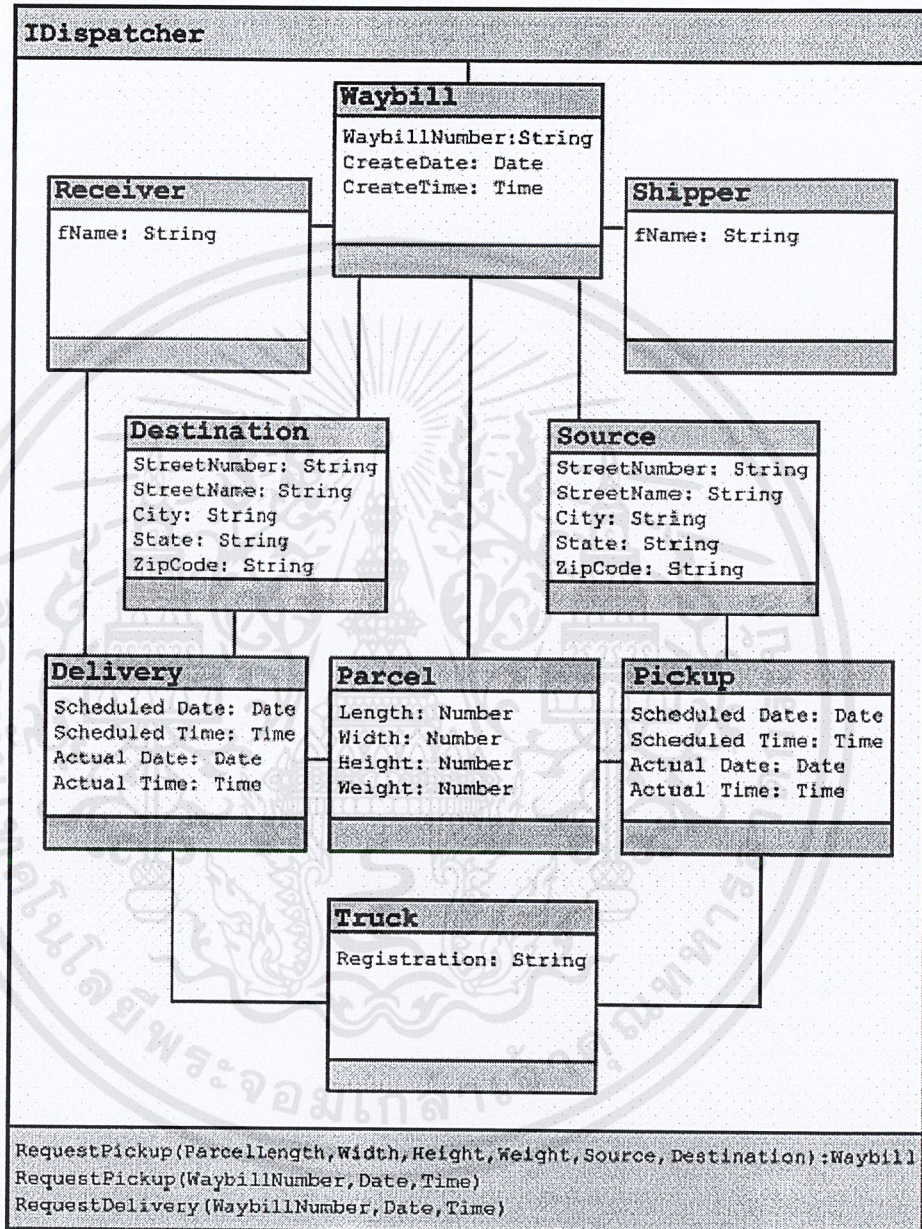
รูปที่ 2-1 คอมโพเนนต์และส่วนประกอบต่างๆ ของมัน

2.1.3 คุณสมบัติของคอมโพเนนต์

- **Encapsulated** - เป็นกระบวนการในการซ่อนโค้ดหรืออิมพลีเมนต์เตชันของคอมโพเนนต์ ผู้ใช้จะรู้เพียง อินเทอร์เฟซและจะใช้งานผ่านมัน โดยไม่จำเป็นต้องรู้ถึงอิมพลีเมนต์เตชัน ทำให้การเปลี่ยนแปลงอิมพลีเมนต์เตชันไม่มีผลกระทบต่อผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

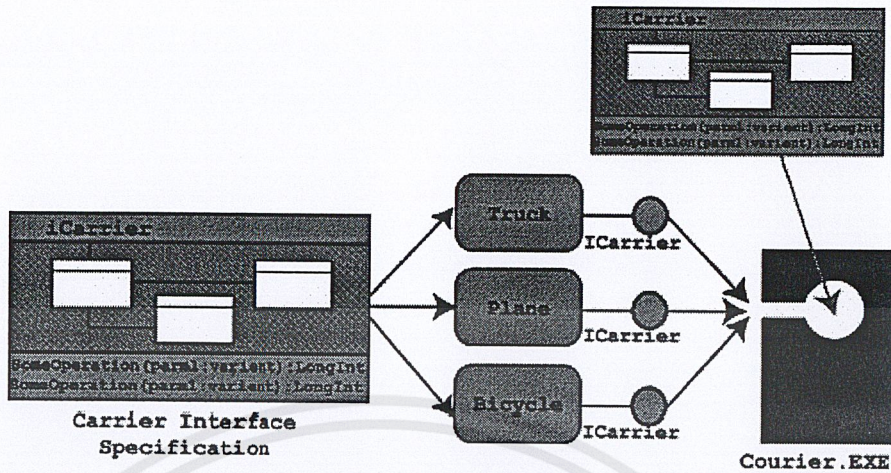
- **Descriptive** - เนื่องจากคอมพิวเตอร์นั้นจะต้องติดต่อผ่านอินเทอร์เน็ตเท่านั้น มันจะต้องมีอินฟอร์เมชันของมันเองที่ผู้ใช้สามารถจะเข้าใจได้ โดยอินฟอร์เมชันนั้นจะต้องอธิบายถึงอินเทอร์เน็ต , อิมพลีเมนต์เตชัน และคิพลอยเมนต์



รูปที่ 2-2 ตัวอย่างของอินเทอร์เน็ตเฟซ

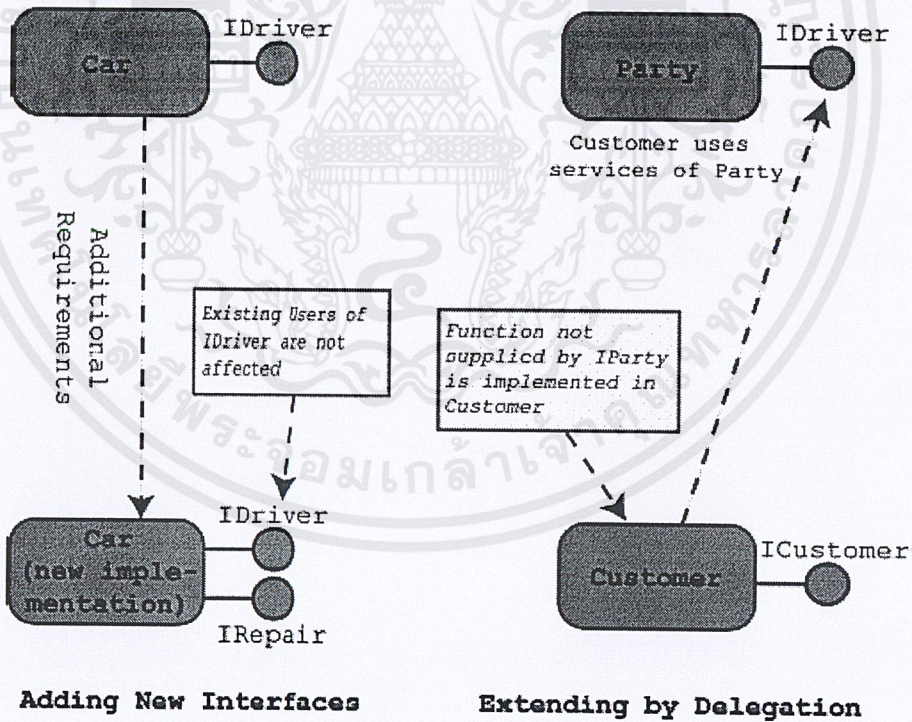
- **Replaceable** - คอมพิวเตอร์มีความสามารถในการเปลี่ยนคอมพิวเตอร์หนึ่งกับคอมพิวเตอร์ใด ๆ ที่มีอินเทอร์เน็ตเฟซเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 คอมโพเนนต์และการแทนที่

- Extensible - สามารถที่จะเพิ่มความสามารถได้ มี 2 วิธีคือ การเพิ่มอินเทอร์เฟซและมอบหมายหน้าที่ (Delegating Responsibility) การสร้างคอมโพเนนต์ใหม่สามารถมอบหมายหน้าที่ให้กับบริการที่มีอยู่ในคอมโพเนนต์ที่มีอยู่แล้ว



รูปที่ 2-4 การเพิ่มเติมคอมโพเนนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 งานที่เหมาะสมกับการพัฒนาเชิง คอมโพเนนต์

การพัฒนาเชิงคอมโพเนนต์ เป็นวิธีการพัฒนาคอมโพเนนต์โดยใช้คอมโพเนนต์เฟรมเวิร์ก ซึ่งการคงหน้าที่ของคอมโพเนนต์ไว้อย่างถูกต้อง (Well defined responsibility) กระทำได้โดยให้คอมโพเนนต์เชิงธุรกิจ หลีกเลียงโค้ดที่ไม่สามารถทำงานกับอินเทอร์เฟซที่แน่นอนได้

เฟรมเวิร์กคอมโพเนนต์จะซ่อนบริการต่างๆ ของเอ็นไวรอนเมนต์ ซึ่งบริการต่างๆ เหล่านี้เป็นมาตรฐานแล้ว เช่น message oriented middle-ware, Transaction Monitors, การรักษาความปลอดภัย และ พิสูจน์สิทธิ์

2.3 ประโยชน์ของการพัฒนาเชิงคอมโพเนนต์

การพัฒนาเชิงคอมโพเนนต์มีประโยชน์อย่างมากด้านเกี่ยวกับเทคนิค และประโยชน์เหล่านี้ยังนำไปสู่ประโยชน์ด้านธุรกิจด้วย

ประโยชน์ด้านเทคนิค

1. จัดการความซับซ้อนได้ดีขึ้น ทำให้คุณภาพของการแก้ปัญหาดีขึ้น
2. งานซับซ้อนที่ไม่เกี่ยวกับฟังก์ชันธุรกิจ ถูกรวมไว้ในเฟรมเวิร์ก
3. การออกแบบ, อิมพลีเมนต์, ทดสอบ ส่วนต่างๆอย่างอิสระ ทำให้สามารถพัฒนาส่วนต่างๆ ได้พร้อมกัน
4. การไม่ขึ้นต่อกันของคอมโพเนนต์ลดผลกระทบของการเปลี่ยนแปลง requirement ไม่ให้กระจายไปทั้งระบบ

ประโยชน์ด้านธุรกิจ

1. ผลิภัณฑ์มีคุณภาพมากขึ้น
2. เร่งเวลาออกสู่ตลาด
3. ใช้ทรัพยากรบุคคลได้คุ้มค่าขึ้น
4. ลดค่าใช้จ่าย
5. เพิ่มปริมาณการนำกลับมาใช้ ในโครงการต่อไป

2.4 แนวโน้มของระบบในอนาคต

- ความต้องการระบบที่ซับซ้อนยิ่งขึ้น
- ความต้องการให้ระบบติดต่อกันจากธุรกิจสู่ธุรกิจ (B2B)
- ความต้องการให้ระบบ โดยด้านเครือข่าย ดีขึ้น

คอมโพเนนต์สามารถเติมเต็มความต้องการเหล่านี้ได้ เพราะระบบสามารถสร้างจากระบบย่อยที่อิสระต่อกัน ทำให้ระบบรวมมีความแข็งแกร่งขึ้น เพราะชิ้นส่วนย่อยหนึ่งไม่ยึดติดกับความยุ่งยากภายในชิ้นส่วนย่อยอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

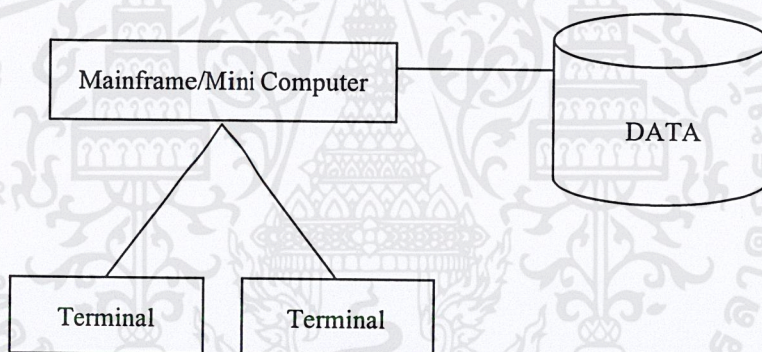
บทที่ 3

วิวัฒนาการของสถาปัตยกรรมแบบกระจาย

3.1 สถาปัตยกรรมแบบหนึ่งเทียร์ (One Tier Architecture/Centralize Host)

สถาปัตยกรรมแบบหนึ่งเทียร์ คือระบบผู้ใช้หลายคนที่เรียบง่ายที่สุด โดยจะมีการเก็บข้อมูลและประมวลผลอยู่ที่คอมพิวเตอร์ส่วนกลางทั้งหมด (เช่น เมนเฟรมคอมพิวเตอร์หรือมินิคอมพิวเตอร์) ผู้ใช้จะติดต่อกับคอมพิวเตอร์ส่วนกลางโดยผ่านทางเทอร์มินอล ซึ่งเป็นเพียงแค่อุปกรณ์อินพุตและเอาต์พุตเพียงอย่างเดียว ไม่มีการประมวลผล ในระบบเช่นนี้มีข้อเสียคือ

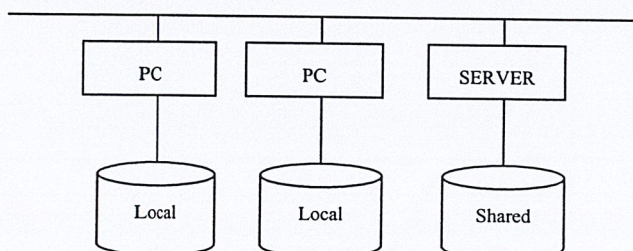
- การที่ทุกๆ กระบวนการทำงานทำในคอมพิวเตอร์ส่วนกลางทั้งหมด ทำให้ต้องแบ่งกำลังการประมวลผลไปให้ผู้ใช้แต่ละคน อาจมีผลกระทบต่อเวลาการตอบสนอง ทำให้ตอบสนองได้ช้าลง
- ถ้าเกิดความขัดข้องที่คอมพิวเตอร์ส่วนกลาง ผู้ใช้ทุกคนจะไม่สามารถใช้งานได้
- ค่าใช้จ่ายในการบำรุงรักษาทางด้านฮาร์ดแวร์ และซอฟต์แวร์สูงมาก



รูปที่ 3-1 สถาปัตยกรรมแบบหนึ่งเทียร์

3.2 เครื่องข่ายภายในและการแบ่งปันทรัพยากรภายในเครือข่าย (PC LANs/ทรัพยากร Sharing LANs)

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



รูปที่ 3-2 เครื่องข่ายภายในของเครื่องคอมพิวเตอร์ส่วนบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนที่เป็นเซิร์ฟเวอร์นั้น อาจเป็นเซิร์ฟเวอร์ที่ให้บริการเพิ่มข้อมูล หรือเซิร์ฟเวอร์ที่ให้ บริการเครื่องพิมพ์ (File/Print Server) ก็ได้ แต่แอปพลิเคชันจะยังคงทำงานอยู่บนพีซีของตัวเอง การที่ข้อมูลอยู่บนคิสก์ที่ใช้ร่วมกันที่เซิร์ฟเวอร์ และแอปพลิเคชันทำงานอยู่ที่พีซี ทำให้เกิดการถ่ายโอนข้อมูล จำนวนมากและเป็นไปอย่างต่อเนื่อง ยกตัวอย่างเช่น การคิวรีข้อมูลจากข้อมูลทั้งหมด 10,000 เรคคอร์ด เซิร์ฟเวอร์จะต้องส่งข้อมูลทั้ง 10,000 เรคคอร์ดนั้นมาประมวลผลคำสั่ง SQL ที่เครื่องไคลเอนต์

3.3 สถาปัตยกรรมแบบสองเทียร์ (ไคลเอนต์/เซิร์ฟเวอร์)

3.3.1 ดาต้าเบสเซิร์ฟเวอร์ (Database Server)

ในระบบดาต้าเบสเซิร์ฟเวอร์นั้นดาต้าและดาต้าเบสเอ็นจิน (DBMS) จะรันอยู่ที่เซิร์ฟเวอร์ ส่วน หน้าจอใช้งานจะรันอยู่บนพีซี ซึ่งเรียกว่า ไคลเอนต์ ในการสืบค้นข้อมูลนั้น จะส่งคำสั่ง SQL statement ไป สืบค้นที่เซิร์ฟเวอร์และเซิร์ฟเวอร์จะส่งเฉพาะผลลัพธ์กลับไปไคลเอนต์ ซึ่งจะช่วยลดความหนาแน่นของ ข้อมูลในระบบเครือข่ายลงได้ ไคลเอนต์กับเซิร์ฟเวอร์จะเป็นอิสระต่อกัน แต่จะมีแอปพลิเคชันทำงานร่วม กันอยู่ แอปพลิเคชันฝั่งเซิร์ฟเวอร์คือ ดาต้าเบสเอ็นจิน ส่วนแอปพลิเคชันฝั่งไคลเอนต์ ก็คือแอปพลิเคชัน ต่างๆ ที่ต้องการติดต่อกับดาต้าเบส แอปพลิเคชันทั้งสองส่วนจะติดต่อสื่อสารกันผ่านเน็ตเวิร์กโพรโตคอล ซึ่งเป็นความรับผิดชอบของ DBMS โดย DBMS จะจัดเตรียมโครงสร้างพื้นฐานทั้งหมดให้

ข้อเสียของไคลเอนต์/เซิร์ฟเวอร์ ก็คือการที่แอปพลิเคชันนั้นรันอยู่บนไคลเอนต์ การแก้ไขแอป พลิเคชันจึงต้องทำทุกๆไคลเอนต์ ทำให้มีค่าใช้จ่ายในการบำรุงรักษาสูงมาก

3.3.2 แอปพลิเคชันเซิร์ฟเวอร์

แอปพลิเคชันเซิร์ฟเวอร์จะมีความสามารถมากกว่าการแชร์ไฟล์หรือเครื่องพิมพ์ในเครือข่ายภายใน หรือการจัดการดาต้าเบสในดาต้าเบสเซิร์ฟเวอร์ แอปพลิเคชันเซิร์ฟเวอร์นั้นสามารถใส่แอปพลิเคชันที่มี ฟังก์ชันการทำงานต่างๆ ไว้ และเรียกใช้เมื่อต้องการได้ โดยแอปพลิเคชันที่จะเรียกใช้ฟังก์ชันที่อยู่บน เซิร์ฟเวอร์จะต้องใช้เน็ตเวิร์กโพรโตคอล เช่น TCP/IP, SPX/IPX หรือโพรโตคอลในระดับสูง เช่น DCOM, COBRA, RMI

3.4 สถาปัตยกรรมแบบสามเทียร์

ในระบบสถาปัตยกรรมแบบสองเทียร์นั้น ทุกๆ ไคลเอนต์ติดต่อโดยตรงกับดาต้าเบส ซึ่งการทำ เช่นนี้เหมาะสมกับงานขนาดเล็ก แต่ไม่เหมาะสมกับองค์กรใหญ่ๆ ที่มีผู้ใช้จำนวนมาก โดยเฉพาะอย่างยิ่ง ในระบบอินเตอร์เน็ตจะมีผู้ใช้ที่ใช้งานพร้อมกันเป็นจำนวนมากในช่วงเวลาที่การใช้งานมากๆ (peak time) จะทำให้เกิดปัญหาการขาดแคลนทรัพยากรได้ ด้วยเหตุนี้ เทียร์ที่สามคือแอปพลิเคชันเทียร์หรือบิสิเนส เทียร์ จึงเกิดขึ้นมาคั่นระหว่างพรีเซนเตชันเทียร์และดาต้าเบสเทียร์เพื่อเพิ่มความยืดหยุ่น และมีความ สามารถในการกระจายโหลดได้มากขึ้น ยกตัวอย่างเช่นบิสิเนสเลเยอร์สามารถจัดการพูล (Pool) การเชื่อมต่อของดาต้าเบสได้ โดยจะไม่ยกเลิกการเชื่อมต่อกับดาต้าเบสเมื่อไม่ใช้งาน แต่จะเก็บลงพูลแทน ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รองรับโคลนเอ็นดีได้มากขึ้น บิสิเนสเลเยอร์นั้นจะแยกโคลนเอ็นดีออกจากคาค้าเบส ไม่ให้ขึ้นต่อกันโดยตรง ดังนั้นคาค้าเบสสามารถเปลี่ยนแปลงได้ โดยไม่มีผลกระทบต่อโค้ดในพีริเซนต์เลเยอร์เลย บิสิเนสเลเยอร์ก็สามารถสร้าง Business Rule ที่เป็นสามัญซึ่งจะใช้ได้ในหลายๆ โคลนเอ็นดีแอปพลิเคชันทำให้เกิดการนำโค้ดต้นฉบับมาใช้ใหม่ ถ้าฟังก์ชันทางธุรกิจเปลี่ยนแปลงไป ก็ปรับปรุงเฉพาะบิสิเนสเลเยอร์เท่านั้น

3.4.1 วินโดวส์ดีเอ็นเอ (Windows DNA)

ไมโครซอฟต์สนับสนุนการกระจายแอปพลิเคชันด้วยสถาปัตยกรรมวินโดวส์ดีเอ็นเอ (Windows Distributed Internet Application Architecture: Windows DNA)

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

วินโดวส์ดีเอ็นเอไม่ใช่ผลิตภัณฑ์ของบริษัทใดๆ แต่เป็นแนวทางการพัฒนาแอปพลิเคชันแบบสามเทียร์โดยทางไมโครซอฟต์ได้จัดเตรียมแพลตฟอร์ม คอมโพเนนต์เทคโนโลยี โครงสร้างภายในแอปพลิเคชัน บริการและเครื่องมือต่างๆ ให้ หัวใจของวินโดวส์ดีเอ็นเอเน้น คือการรวมโมเดลการเขียนโปรแกรมที่อยู่บนพื้นฐานของ COM เข้าด้วยกัน

3.4.1.1 พีริเซนต์เลเยอร์

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

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

3.4.1.2 บิสิเนสโลจิกเทียร์ (Business Logic Tier)

บิสิเนสโลจิกเลเยอร์หรือ Middle Tier นั้น เป็นส่วนที่สำคัญที่สุดของสถาปัตยกรรมแบบสามเทียร์ บริการใหม่ ๆ ของไมโครซอฟต์จะอยู่ที่เลเยอร์นี้ เช่น

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

3.4.1.3 ดาต้าแอ็กเซสเลเยอร์ (Data Access Layer)

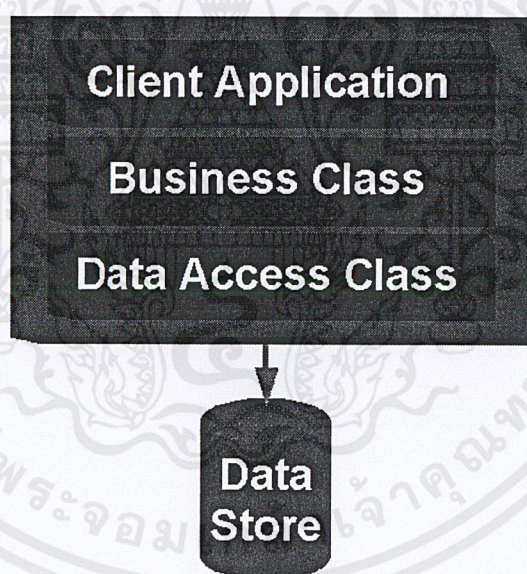
ในดาต้าแอ็กเซสเลเยอร์นี้ ไมโครซอฟต์มีผลิตภัณฑ์ที่เป็นตัวจัดการฐานข้อมูลคือ Microsoft SQL Server 2000 แต่ก็ยังกำหนด Data Access Interface ซึ่งสนับสนุนผลิตภัณฑ์อื่น ๆ ด้วย เช่น Open Database Connectivity (ODBC), OLE DB

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

3.5 สถาปัตยกรรมแบบหลายเทียร์(N-Tier)

3.5.1 แอปพลิเคชันแบบหลายเทียร์ทางตรรก

วิธีที่ดีที่สุดในการพัฒนาแอปพลิเคชันโดยใช้ .NET นั้น คือการแบ่งลอจิกคอลลอกเป็นคลาสที่ไม่ขึ้นต่อกัน โดยทั่วไปแอปพลิเคชันในทางธุรกิจ จะเกี่ยวข้องกับกฎทางธุรกิจ ส่วนประกอบของชั้นข้อมูล และโค้ดที่ใช้ส่วนประกอบนั้นๆ



รูปที่ 3-3 สถาปัตยกรรมแบบ เอ็น เทียร์

สำหรับเทคนิคนี้ การใช้ลจิกคอลลอก n-tier ในการพัฒนาแอปพลิเคชัน สามารถใช้ได้ทั้งในแอปพลิเคชันขนาดเล็ก กลาง และใหญ่ ร่วมกับเดสก์ท็อปและเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางเลือกโดยทั่วไปในการพัฒนาแอปพลิเคชัน

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

ข้อดี

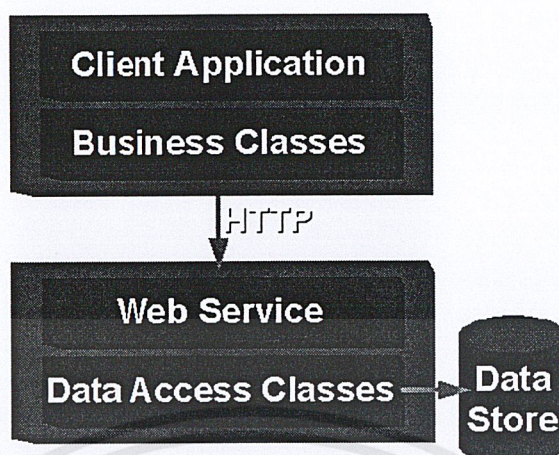
- การรวมกฎทางธุรกิจเป็นคอมโพเนนต์จะทำให้ง่ายต่อการสร้าง การนำไปใช้ และการนำกลับมาใช้ใหม่ ทำให้ง่ายต่อการพัฒนาและการดูแล
- สามารถใช้ภาษาระดับสูง ในการพัฒนากฎทางธุรกิจ ซึ่งตรงข้ามกับการใช้สโตรโปรซิเจอร์ (Stored procedure) และการตรวจสอบการจำกัดภาษา SQL
- รวมการเข้าถึงข้อมูลเป็นคอมโพเนนต์ ทำให้ลดการเขียนโค้ดซ้ำๆกันในแอปพลิเคชัน แต่ละฟอร์มที่ต้องการติดต่อกับฐานข้อมูลจะใช้คอมโพเนนต์ตัวเดียวกัน
- หากใช้ดาต้าเซต(DataSet) จะทำให้สะดวกต่อการค้นหาอัตโนมัติโดยใช้ อินเทลลิเซนซ์ (IntelliSense) แทนที่จะต้องจำ
- รวบรวม (Routines) ในการเข้าถึงข้อมูลจะช่วยให้การดูแล เนื่องจากการเปลี่ยนแปลงข้อมูลในการเข้าถึงข้อมูลจะต้องทำเพียงครั้งเดียว
- มีความยืดหยุ่นในการแบ่งสรรคอมโพเนนต์ให้กับเครื่องที่ทำงานเฉพาะอย่างได้ไม่ว่าเวลาใดก็ตาม ช่วยให้การขยายระบบให้รองรับการใช้งานได้มากขึ้น และง่ายต่อการรวบรวมโค้ด

ข้อเสีย

- การพัฒนายาวนานเนื่องจากต้องเสียเวลาในการแบ่งคอมโพเนนต์
- มีคอมโพเนนต์ที่ต้องติดตาม ทำให้การพัฒนาเมธอด (method) มีความยุ่งยากมากขึ้นเล็กน้อยที่จะทำให้นักพัฒนาใหม่จะเข้าใจ

3.5.2 แอปพลิเคชัน N-Tier โดยใช้ XML เว็บเซอร์วิส

สถาปัตยกรรมนี้พัฒนาการออกแบบลจิกคอล n-tier แอปพลิเคชันขึ้น โดยแบ่งออกเป็นหลายๆ เครื่อง XML เว็บเซอร์วิสที่ติดต่อไปยังชั้นข้อมูล



รูปที่ 3-4 สถาปัตยกรรมแบบ เอ็น เทียร์ ผ่านเว็บเซอร์วิส

โดยใช้ค่าค่าเซิร์ฟเวอร์จะส่งข้อมูลกลับมาจากชั้น HTTP มายังชั้นของธุรกิจ แอปพลิเคชันทางฝั่งไคลเอนท์จะรับข้อมูลมาเพื่อใช้ในการแสดงผล

สำหรับเทคนิคนี้ การออกแบบเว็บเซอร์วิสโดยใช้ XML นั้นจะทำกับวินโดวส์ฟอร์มแอปพลิเคชัน แต่ผู้ใช้จะต้องติดต่อไปยังเครื่องที่ต้องการ และรับข้อมูลผ่านทางชั้น HTTP การเก็บกฎทางธุรกิจบนฝั่งไคลเอนท์นั้น จะทำให้ลดปัญหาแทรกฟิคบนเน็ตเวิร์ก แต่จะทำให้การดูแลลำบากขึ้นเมื่อมีการเปลี่ยนแปลงกฎบ่อยครั้ง

การทำงานในลักษณะนี้ จะใช้ไค้กับเว็บเบสแอปพลิเคชันได้เช่นกัน โดยที่ข้อมูลจะต้องถูกเตรียมโดยเว็บเซิร์ฟเวอร์ และแสดงผลบนเว็บแอปพลิเคชันโดยเว็บเซิร์ฟเวอร์อีกเครื่องหนึ่ง ทางเลือกทั่วไปในการพัฒนา

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

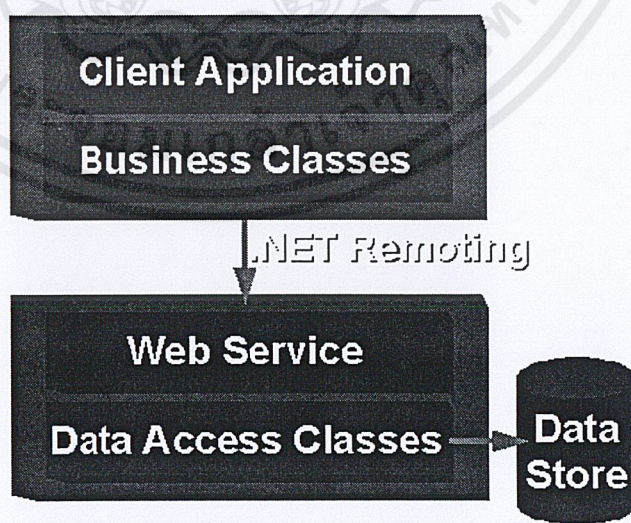
ข้อดี

- การรวมกฎทางธุรกิจเป็นคอมโพเนนต์จะทำให้ง่ายต่อการสร้าง การนำไปใช้ และการนำกลับมาใช้ใหม่ ทำให้ง่ายต่อการพัฒนาและการดูแล
- สามารถใช้ภาษาระดับสูง ในการพัฒนากฎทางธุรกิจ ซึ่งตรงข้ามกับการใช้สโตรโปรซิเจอร์ (Stored procedure) และการตรวจสอบการจำกัดภาษา SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รวมการเข้าถึงข้อมูลเป็นคอมโพเนนต์ ทำให้ลดการเขียนโค้ดซ้ำๆกันในแอปพลิเคชัน แต่
ละฟอร์มที่ต้องการติดต่อกับฐานข้อมูลจะใช้คอมโพเนนต์
ตัวเดียวกัน
 - หากใช้ดาต้าเซ็ท จะทำให้สะดวกต่อการค้นหาคอลัมน์โดยใช้ อินเทลลิเซนซ์ (IntelliSense)
แทนที่จะต้องจำ
 - รวบรวมทีน (Routines) ในการเข้าถึงข้อมูลจะช่วยในการดูแล เนื่องจากการเปลี่ยนแปลงข้อมูลใน
การเข้าถึงข้อมูลจะต้องทำเพียงครั้งเดียว
 - มีความยืดหยุ่นในการแบ่งสรรคอมโพเนนต์ให้กับเครื่องที่ทำงานเฉพาะอย่างได้ไม่ว่าเวลาใดก็
ตาม ช่วยให้การขยายระบบให้รองรับการใช้งานได้มากขึ้น
และง่ายต่อการรวบรวมโค้ด
 - ชั้นการเข้าถึงข้อมูลเป็นศูนย์กลางในการเข้าถึงข้อมูล
 - สามารถขยายระบบให้รองรับการใช้งานได้มากขึ้น โดยการเพิ่มเว็บฟอร์มเพื่อรองรับการร้องขอ
เพื่อเข้าถึงฐานข้อมูลของผู้ใช้
 - ผู้ใช้สามารถติดต่อผ่านทางอินเทอร์เน็ตและยังคงเข้าถึงข้อมูลได้จากเดสก์ท็อปแอปพลิเคชัน
หรือเว็บเบสแอปพลิเคชัน
- ข้อเสีย
- การพัฒนายาวนานเนื่องจากต้องเสียเวลาในการแบ่งคอมโพเนนต์
 - มีคอมโพเนนต์ที่ต้องติดตาม ทำให้การพัฒนาเมธอด (method) มีความยุ่งยากมากขึ้นเล็กน้อยที่
จะทำให้ให้นักพัฒนาใหม่จะเข้าใจ

3.5.3 แอปพลิเคชัน N-Tier โดยใช้รีโมต .NET



รูปที่ 3-5 สถาปัตยกรรมแบบ เอ็น เทียร์ผ่าน .NET Remoting

การทำงาน ข้อดีและข้อเสียจะเหมือนกับ n-tier เว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

เว็บเซอร์วิส

เว็บเซอร์วิส คือ เว็บแอปพลิเคชันยุคใหม่ ที่ประกอบด้วยส่วนย่อยๆที่มีความสมบูรณ์ในตัวเอง สามารถติดตั้ง ค้นหา เริ่มทำงานได้ผ่านเว็บ เว็บเซอร์วิส สามารถทำอะไรก็ได้ตั้งแต่งานง่ายๆ เช่นดึงข้อมูล จนถึงกระบวนการทางธุรกิจที่ซับซ้อน เมื่อ เว็บเซอร์วิสตัวใดตัวหนึ่งเริ่มเว็บเซอร์วิสตัวอื่นก็สามารถรับรู้ และเริ่มทำงานได้อีกด้วย

หลายคนอาจจะถามว่าทำไมต้องเป็น เว็บ เพราะโดยปกติมี Middle Ware อื่นๆมากมายเช่น RMI Jini CORBA DCOM ฯลฯ แม้ Middle Ware เหล่านี้จะสามารถรองรับได้ แต่ไม่มีตัวใดตัวหนึ่งที่เด่นจริง แต่ในเมื่อ Web มีจุดเด่นในเรื่องของการให้บริการข้อมูลที่สะดวก ใช้งานง่าย จึงกลายเป็นตัวประสาน Middle Ware ต่างๆ เข้าด้วยกันซึ่งจะให้คุยกันเองคงยากยิ่ง Web ได้ทำหน้าที่เป็นตัวกลางให้ Middle Ware เหล่านี้สามารถคุยกันได้ และมีประสิทธิภาพกว่าวิธีการเดิมๆ มาก

หากเรามองจากกรณีของ N-Tier แอปพลิเคชัน จะพบว่า เว็บเซอร์วิส คือกลไกในการเข้าถึง บริการที่แต่ละ Middle Ware ให้บริการ การเข้าถึงจะอาศัย Listener และส่วนประกอบที่ระบุถึงบริการ ต่างๆ ที่รองรับการทำงาน โดยการทำงานจริงๆ นั้นก็ใช้วิธีการปกติของ Middle Ware นั้นๆ พื้นฐานของเว็บเซอร์วิสก็คือ XML กับ HTTP ซึ่งจะพบว่า HTTP ก็เป็นที่รู้จักกันดี และไปได้ทั่วทุกแห่งที่มี อินเทอร์เน็ต ส่วน XML คือภาษาสากลที่คุณสามารถปรับแต่งได้ตามใจชอบ เพื่อให้เกิดกิจกรรมระหว่าง ไคลเอนต์ และบริการ หรือระหว่างส่วนประกอบต่างๆ เมื่อหลังเว็บเซอร์วิสก็คือ ข้อความ XML จะถูก แปลงให้การขอบริการจาก Middle ware และผลที่ได้ก็จะแปลงกลับมาในรูป XML โดยมาตรฐาน XML นี้ ทำให้เราไม่ต้องกังวลเรื่องของการเชื่อมโยงอีกต่อไป และโปรโตคอล ที่ส่งก็คือ HTTP นั่นเอง ถ้าท่าน เชื่อมโยงกับ HTTP (หรือเว็บ) ได้ ท่านก็ใช้บริการทุกอย่างได้

แต่อย่างไรก็ตามการเข้าถึงและการส่งงานนั้นยังเป็นเพียงโครงสร้างพื้นฐาน แต่ในความเป็นจริง ยังมีอะไรมากกว่านั้น เช่น การค้นหา การทำธุรกรรม ความปลอดภัย การพิสูจน์ตัวตน และอื่นๆ อันเป็น บริการที่ทำให้เป็นบริการพื้นฐานจริงๆ ระบบเพิ่มเติมที่ต้องมีและต้องรักษาความสะดวกและใช้งานง่าย ไว้ด้วย พื้นฐานของเว็บเซอร์วิส เต็มรูปแบบคือ XML + HTTP + SOAP + WSDL + UDDI หรือในระดับ สูงกว่านั้น แต่ไม่ได้ถือเป็นสิ่งจำเป็นเสมอไปคือต้องเพิ่มเทคโนโลยี XAML, XLANG, XKMS, XFS เป็นต้น สำหรับรายละเอียดในบางหัวข้อมีดังต่อไปนี้คือ

4.1 XML

4.1.1 แนะนำ XML

XML ย่อมาจาก eXtensible Markup Language XML ถูกออกแบบมาเพื่อใช้อธิบายข้อมูล ใน XML จะไม่มีแท็ก (tag) ที่กำหนดไว้ล่วงหน้า เราจะต้องสร้างแท็กของเราขึ้นมาเอง และจะใช้ Document Type Definition (DTD) ในการอธิบายรูปแบบของข้อมูล

ข้อแตกต่างที่สำคัญระหว่าง XML และ HTML คือ XML ถูกออกแบบมาเพื่อใช้แลกเปลี่ยนข้อมูล ไม่ใช่มาแทนที่ HTML XML กับ HTML ถูกออกแบบมาเพื่อจุดประสงค์ที่ต่างกัน HTML จะเกี่ยวข้องกับการแสดงข้อมูล XML จะเกี่ยวข้องกับการอธิบายข้อมูล แท็กของ HTML เช่น <H1></H1> เป็นแท็กสำหรับบอกรูปแบบการแสดงผล ในขณะที่แท็กของ XML ต้องสร้างขึ้นมาเองเช่น

```
<BOOK>
  <NAME>Beginning XML</NAME>
  <ISDN>1-861003-41-2</ISDN>
  <PAGE>800</PAGE>
</BOOK>
```

จะใช้แสดงข้อมูลของหนังสือที่ประกอบด้วย element BOOK , NAME , ISDN และ PAGE ข้อมูลที่อยู่ในรูปแท็กขึ้นนี้เป็นอิสระต่อซอฟต์แวร์และฮาร์ดแวร์ จึงเหมาะที่จะนำมาใช้ในการแลกเปลี่ยนข้อมูลระหว่างระบบบนอินเทอร์เน็ต การเปลี่ยนข้อมูลให้อยู่ในรูป XML เป็นการลดความซับซ้อนและสามารถสร้างข้อมูลที่ถูกรับโดยแอปพลิเคชันใด ๆ ก็ได้ นอกจากนั้นการเพิ่มเติม element จะไม่ทำให้แอปพลิเคชันเสียหาย เนื่องจากการอ่านข้อมูลโดย XML parser จะใช้วิธีแยกข้อมูลจาก element ที่ต้องการ จาก XML ข้างต้นนี้จะแยกข้อมูล element NAME , ISDN และ PAGE หากเพิ่ม element AUTHOR เข้าไป แอปพลิเคชันก็ยังคงดึงข้อมูลใน element เดิมโดยจะมองข้าม element ที่ไม่ได้กำหนดให้แยกข้อมูลออกมาในตอนพัฒนาแอปพลิเคชัน

4.1.2 ไวยากรณ์ของ XML

กฎไวยากรณ์ของ XML นั้นง่ายและเคร่งครัดมาก กฎของมันง่ายต่อการเรียนรู้และง่ายต่อการใช้ ด้วยเหตุนี้การสร้างซอฟต์แวร์ในการอ่านและจัดการกับข้อมูลจึงเป็นเรื่องง่าย ตัวอย่างของเอกสาร XML ดังนี้

```
<?XML version="1.0"?>
<note>
<to>Tove</to>
```

```
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ในบรรทัดแรกคือการประกาศเอกสาร XML ที่บอกถึง XML เวอร์ชัน
บรรทัดต่อมาคือ root element (note) อีก 4 บรรทัดถัดมาเป็นการอธิบายถึง 4 child element ของ root (to , from , heading , body) และบรรทัดสุดท้ายเป็นการกำหนดจุดสิ้นสุดของ root
กฎของเอกสาร XML มีดังนี้

▪ XML Element ทุกตัวต้องมีการปิดแท็ก

ใน HTML บาง element ไม่จำเป็นต้องมีการปิดแท็ก แต่ใน XML ทุก element จำเป็นที่จะต้องมีการปิดแท็ก

| | |
|----------------------------|---|
| <p>This is a paragraph | X |
| <p>This is a paragraph</p> | ✓ |

▪ XML แท็กมีความแตกต่างระหว่างตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ (case sensitive)

ใน XML แท็ก <Letter> นั้นต่างจาก <letter> เนื่องจาก XML นั้นมีคุณสมบัติที่แยกความแตกต่างระหว่างตัวพิมพ์เล็กและตัวพิมพ์ใหญ่

| | |
|--------------------------------------|---|
| <Message>This is incorrect</message> | X |
| <message>This is correct</message> | ✓ |

▪ XML Element ทุกตัวต้องมีการซ้อนกันอย่างเหมาะสม

ใน HTML บาง element สามารถเหลื่อมกันได้ แต่ใน XML นั้นต้องซ้อนกันเป็นชั้น โดยสมบูรณ์

| | |
|--|---|
| <i>This text is bold and italic</i> | X |
| <i>This text is bold and italic</i> | ✓ |

▪ ทุกเอกสาร XML ต้องมี root แท็ก

แท็กแรกในเอกสาร XML นั้นจะเป็น root แท็กและจะมีได้เพียงหนึ่งเดียวเท่านั้น element อื่น ๆ ทั้งหมดจะต้องถูกซ้อนอยู่ภายใน root element เท่านั้น และ element เหล่านี้ก็สามารถมี element ย่อย (child element) ได้อีก ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>

```

- ค่าของ Attribute ต้องมีอยู่ภายในเครื่องหมายคำพูด ("")

XML element สามารถมี attribute ได้ โดยที่ค่าของ attribute จะต้องอยู่ภายในเครื่องหมายคำพูด
 (") เอกสาร XML ข้างล่างนี้ อันแรกผิด อันที่สองถูก

```

<note date=12/11/99>
<note date="12/11/99">

```

X

✓

4.1.3 XML Element

XML element มีคุณสมบัติดังนี้

- XML Element สามารถขยายได้ตามตัวอย่างนี้

```

<note>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>

```

ถ้าเราจะสร้างแอปพลิเคชันที่สามารถแยก element to , from และ body จาก เอกสาร XML เพื่อแสดงผลลัพธ์ดังนี้

MESSAGE

To:Tove

From: Jani

Don't forget me this weekend!

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ ถ้าเพิ่มข้อมูลเพิ่มเติมลงไปดังนี้

```
<note>
  <date>1999-08-01</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

เมื่อเอกสาร XML เป็นดังนี้แล้ว แอปพลิเคชันที่สร้างไว้จะไม่เสียหาย เนื่องจาก แอปพลิเคชันจะยังคงค้นหา element to , from และ heading ในเอกสาร XML และยังคงให้ผลลัพธ์เหมือนเดิม

- XML Element มีความสัมพันธ์กัน

ในการที่จะเข้าใจถึงภาพรวมของ XML จะต้องเข้าใจถึงความสัมพันธ์ระหว่างชื่อของ XML element ที่ตั้งและ content ของ element คำอธิบายของหนังสือดังนี้

```
Book Title: My First XML
Chapter 1: Introduction to XML
  What is HTML
  What is XML
Chapter 2: XML Syntax
  Elements must have a closing tag
  Elements must be correctly nested
```

สามารถมีเอกสาร XML ที่อธิบายหนังสือนั้น ได้ดังนี้

```
<book>
  <title>My First XML</title>
  <prod id="33-657" media="paper"></prod>
  <chapter>Introduction to XML
    <para>What is HTML</para>
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<para>What is XML</para>
</chapter>
<chapter>XML Syntax
  <para>Elements must have a closing tag</para>
  <para>Elements must be properly nested</para>
</chapter>
</book>

```

book เป็น root element title และ chapter เป็น child element ของ book book เป็น parent element ของทั้ง title และ chapter title และ chapter จึงเป็น sibling กันเนื่องจากมี parent เดียวกัน

■ Element มี Content

XML element คือทั้งหมดตั้งแต่เปิดแท็กจนถึงปิดแท็ก element หนึ่งสามารถมี element content , mixed content , simple content หรือ empty content และยังมี attribute ได้ด้วย

จากตัวอย่างข้างบน book มี element content เพราะมันประกอบด้วย element อื่น chapter เป็น mixed content เพราะมันประกอบด้วยแท็กและ element อื่น ๆ para เป็น simple content (text content) เพราะมันประกอบด้วยแท็กเท่านั้น prod เป็น empty element เนื่องจากมันไม่มีข้อมูลใดๆ และเฉพาะ prod element เท่านั้นที่มี attribute attribute ที่ชื่อ id มีค่าเท่ากับ “33-657” attribute ที่ชื่อ medis มีค่าเท่ากับ “paper”

■ การตั้งชื่อ Element

XML element มีกฎการตั้งชื่อดังนี้

- ชื่อประกอบด้วย ตัวอักษร , ตัวเลข และอักขระอื่นๆ
- ชื่อต้องไม่ขึ้นต้นด้วยตัวเลขหรือ “_” (underscore)
- ชื่อต้องไม่ขึ้นด้วยคำว่า XML (หรือ Xml หรือ xml)
- ชื่อไม่สามารถมีช่องว่างได้ (space)
- ชื่อต้องไม่ควรประกอบด้วย “:” เพราะมันถูกสงวนไว้กับการใช้ namespace

ทุกๆชื่อสามารถใช้ได้ ไม่มีค่าใดเป็นคำสงวน แต่จะเป็นการดีในการที่จะหลีกเลี่ยงการใช้ “_” หรือ “:” แล้วใช้เฉพาะ “_” เท่านั้น เนื่องจากอาจเกิดการสับสนของซอฟต์แวร์ในกรณีพยายามที่จะตัดคำ (first-name) หรือคิดว่า “name” เป็นคุณสมบัติของ object “first” (first.name)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 XML Attribute

XML attribute มีคุณสมบัติดังนี้

- รูปแบบของ Quote , “female” or ‘female’

ค่าของ attribute ต้องอยู่ในเครื่องหมายคำพูดเสมอ แต่อาจเป็น ‘ (single quote) หรือ “ (double quote) ก็ได้

```
<person sex="female">
```

✓

หรือ

```
<person sex='female'>
```

✓

Double quote จะเป็นที่ใช้โดยทั่วไปมากกว่า แต่ในบางครั้ง (ถ้าค่าของ attribute จำเป็นที่จะต้องมี quote) จึงจำเป็นต้องใช้ single quote ดังนี้

```
<gangster name='George "Shotgun" Ziegler'>
```

- การใช้ Element และ Attribute

ข้อมูลสามารถที่จะถูกเก็บใน child element หรือ attribute ดังนี้

```
<person sex="female">
```

```
  <firstname>Anna</firstname>
```

```
  <lastname>Smith</lastname>
```

```
</person>
```

หรือ

```
<person>
```

```
  <sex>female</sex>
```

```
  <firstname>Anna</firstname>
```

```
  <lastname>Smith</lastname>
```

```
</person>
```

ตัวอย่างแรก sex เป็น attribute ตัวอย่างที่ 2 sex เป็น child element ซึ่งทั้งสองตัวอย่างให้ความหมายเดียวกัน มันไม่มีกฎว่าจะใช้ attribute เมื่อไร หรือ child element เมื่อไร แต่แนะนำว่าควรหลีกเลี่ยงการใช้ attribute ควรจะใช้ child element ถ้าอินฟอร์เมชันนั้นดูเหมือนจะเป็นข้อมูล (data)

▪ หลีกเลี่ยงการใช้ Attribute

บางปัญหาของการใช้ attribute

- attribute ไม่สามารถมีหลายค่า (child element สามารถทำได้)
- attribute ง่ายที่จะขยาย (ในอนาคต)
- attribute ไม่สามารถอธิบายโครงสร้าง (child element สามารถทำได้)
- attribute ยากที่จะจัดการด้วยโค้ดของโปรแกรม
- ค่าของ attribute ง่ายในการจะตรวจสอบกับ DTD ที่มีอยู่

แต่อย่างไรก็ตาม จะสามารถใช้ attribute ก็ต่อเมื่อ อินฟอร์เมชันนั้นไม่สัมพันธ์กับข้อมูล ตัวอย่างเช่น ในบางครั้ง ID reference สามารถถูกใช้ในการเข้าถึง XML element

```
<messages>
  <note ID="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note ID="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not!</body>
  </note>
</messages>
```

ID ในตัวอย่างนี้เป็น counter หรือ unique identifier ในการอ้างอิงถึง note ที่ต่างกันของไฟล์ XML ไม่ใช่ส่วนหนึ่งของข้อมูล note

4.1.5 XML Validation

เอกสาร XML แบ่งการตรวจสอบความถูกต้องของเอกสารเป็น 2 ชนิดคือ

1) “Well Formed” XML Document

“Well Formed” XML document เป็นเอกสารที่เป็นไปตามกฎไวยากรณ์ XML ที่ได้กล่าวมาข้างต้น

เอกสารนี้เป็นเอกสารที่สว่นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) “Valid” XML Document

“Valid” XML document เป็นเอกสาร XML ที่มีรูปแบบที่เป็นไปตามกฎของ DTD โดยจะต้องตรงตามโครงสร้างข้อมูลของเอกสารที่ได้ประกาศโดย DTD

▪ XML DTD

วัตถุประสงค์ของ DTD คือกำหนดโครงสร้างของเอกสาร XML เพื่อใช้ในการตรวจสอบว่าเอกสาร XML นั้นถูกต้องตามรูปแบบหรือไม่ เช่น element BOOK จะต้องมีการมี element NAME เสมอ อาจจะมี element PAGE หรือไม่ก็ได้ เป็นต้น

▪ XML Schema

XML schema มีวัตถุประสงค์เหมือนกับ DTD จะถูกนำมาใช้แทน DTD เนื่องจาก

- ง่ายในการเรียนรู้มากกว่า DTD
- ใช้ได้ดีกว่า DTD
- schema เขียนด้วย XML
- รองรับ datatype
- รองรับ namespace

4.1.6 แนะนำ DTD

1) การประกาศ DOCTYPE ภายใน

ถ้า DTD ถูกรวมอยู่ในไฟล์ XML นั้น มันจะต้องอยู่ภายในการให้คำจำกัดความ DOCTYPE ตามไวยากรณ์นี้

```
<!DOCTYPE root-element [element-declarations]>
```

ตัวอย่างดังนี้

```
<?XML version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<to>Tove</to>

<from>Jani</from>

<heading>Reminder</heading>

<body>Don't forget me this weekend</body>

</note>

```

DTD ข้างบนนี้สามารถแปลความหมายได้ดังนี้

!DOCTYPE note (บรรทัดที่ 2) กำหนดว่าเป็นเอกสารของ type note

!ELEMENT note (บรรทัดที่ 3) กำหนดว่า note element มี 4 element คือ to , from , heading และ body

!ELEMENT to (บรรทัดที่ 4) กำหนดว่า to element เป็นข้อมูลชนิด #PCDATA

!ELEMENT from (บรรทัดที่ 5) กำหนดว่า from element เป็นข้อมูลชนิด #PCDATA

2) การประกาศ DOCTYPE ภายนอก

ถ้า DTD อยู่ภายนอกไฟล์ XML มันจะต้องอยู่ภายในการให้คำจำกัดความ DOCTYPE ตามไวยากรณ์นี้

```

<!DOCTYPE root-element SYSTEM "filename">

```

ตัวอย่างไฟล์ XML เป็นดังนี้

```

<?XML version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>

```

ไฟล์ของ DTD ชื่อ note.dtd เป็นดังนี้

note.dtd

```

<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

4.1.6.1 โครงสร้างของ XML

เอกสาร XML จะถูกสร้างจากส่วนต่างๆ ดังนี้

- Element – เป็นโครงสร้างหลักของ XML
- Tag - ใช้ในการกำหนด markup element
- Attribute - ให้ข้อมูลเพิ่มเติมกับ element
- Entity – เป็นตัวแปรที่ใช้ในการกำหนดเท็กซ์ทั่ว ๆ ไป entity reference เป็นการอ้างถึง entity เช่น entity ชื่อ domain มีค่าเท่ากับ www.w3.org entity reference คือ &domain หากภายในเอกสาร XML มี &domain ค่าของมันจะมีค่าเท่ากับ www.w3.org
- PCDATA - หมายถึง parse character data PCDATA เป็นเท็กซ์ที่จะถูก parse โดย parser
- CDATA - หมายถึง character data เป็นเท็กซ์ที่จะไม่ถูก parse โดย parser

4.1.6.2 การกำหนด Element ใน DTD

วิธีการในการประกาศ element จะเป็นดังนี้

- การประกาศ Element
ประกาศตามไวยากรณ์นี้

```
<!ELEMENT element-name category>
```

หรือ

```
<!ELEMENT element-name (element-content)>
```

- Element ว่างเปล่า (Empty)
ถูกประกาศโดยคีย์เวิร์ด EMPTY

```
<!ELEMENT element-name EMPTY>
```

DTD example:

```
<!ELEMENT br EMPTY>
```

XML example:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<br />
```

- **Element ที่เป็นข้อมูลชนิดตัวอักษร**
element ที่เป็นตัวอักษรเท่านั้นจะถูกประกาศด้วยคีย์เวิร์ด PCDATA ที่เปิดและปิดด้วย “(“ ”)”

```
<!ELEMENT element-name (#PCDATA)>
```

DTD example:

```
<!ELEMENT note (#PCDATA)>
```

- **Element ที่เป็นข้อมูลใด ๆ**
ประกาศกับคีย์เวิร์ด ANY content จะสามารถเป็นข้อมูลชนิดใดก็ได้

```
<!ELEMENT element-name ANY>
```

DTD example:

```
<!ELEMENT note ANY>
```

- **Element และ Children (Sequences)**
ประกาศโดยใช้ “;” ในการแยก child element แล้วเปิดและปิดด้วย “(“ ”)”

```
<!ELEMENT element-name (child-element-name)>
```

หรือ

```
<!ELEMENT element-name (child-element-name,child-element-name,.....)>
```

DTD example:

```
<!ELEMENT note (to,from,heading,body)>
```

ถ้ามีการประกาศ child element จะต้องประกาศ child element ให้สมบูรณ์ด้วยคั้งตัวอย่างข้างล่างนี้

```
<!ELEMENT note (to,from,heading,body)>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

- การประกาศเพียงครั้งเดียวของ Element เดียวกัน

```
<!ELEMENT element-name (child-name)>
```

DTD example:

```
<!ELEMENT note (message)>
```

จากตัวอย่างหมายความว่า note element จะประกอบด้วย child element message เท่านั้น

- การประกาศอย่างน้อย 1 ครั้งของ Element เดียวกัน

```
<!ELEMENT element-name (child-name+)>
```

DTD example:

```
<!ELEMENT note (message+)>
```

เครื่องหมาย + ถูกใช้ในความหมายหนึ่งหรือมากกว่า จึงหมายความว่า note element จะประกอบด้วย child element message ตั้งแต่ 1 คำขึ้นไป

- การไม่ประกาศหรือประกาศเท่าไรก็ได้ของ Element เดียวกัน

```
<!ELEMENT element-name (child-name*)>
```

DTD example:

```
<!ELEMENT note (message*)>
```

เครื่องหมาย * ถูกใช้ในความหมายว่าไม่มีหรือมีเท่าไรก็ได้ จึงหมายความว่า note element จะประกอบด้วย child element message ก็ค่าก็ได้ หรือไม่มีเลยก็ได้

- การไม่ประกาศหรือประกาศเพียงครั้งเดียวของ Element เดียวกัน

```
<!ELEMENT element-name (child-name?)>
```

DTD example:

```
<!ELEMENT note (message?)>
```

เครื่องหมาย ? ถูกใช้ในความหมาย ไม่มีหรือมีเพียงหนึ่ง จึงหมายความว่า note element จะประกอบด้วย child element message หนึ่งค่า หรือไม่มีก็ได้

- การประกาศ Content แบบให้เลือก
จะใช้เครื่องหมาย | แทนความหมายหรือ

DTD example:

```
<!ELEMENT note (to,from,header,message|body)>
```

ดังนั้นตัวอย่างนี้จึงหมายถึง note element จะต้องประกอบด้วย to element , from element , header element แล้วก็ message element หรือ body element ใดอย่างหนึ่งเท่านั้น

- การประกาศ Content แบบผสม
เป็นการผสมการประกาศหลาย ๆ อย่างเข้าด้วยกัน

DTD example:

```
<!ELEMENT note (#PCDATA|to|from|header|message)*>
```

ตัวอย่างนี้หมายถึง note element จะประกอบด้วย character data หรือ element to , from , header , และ message จำนวนเท่าไรก็ได้

4.1.6.3 การกำหนด Attribute ใน DTD

วิธีการประกาศ attribute จะเป็นดังนี้

- การประกาศ Attribute
attribute ต้องถูกประกาศตามไวยากรณ์ นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<!ATTLIST element-name attribute-name attribute-type default-value>
```

DTD example:

```
<!ATTLIST payment type CDATA "check">
```

XML example:

```
<payment type="check">
```

ชนิดของ attribute สามารถมีค่าได้ดังตารางที่ 4-1

| ค่า | คำอธิบาย |
|--------------|---|
| CDATA | ค่าต้องเป็น character ที่จะไม่ถูก parse โดย parser |
| (en1 en2 ..) | ค่าต้องเป็นอันใดอันหนึ่งในรายการนั้น |
| ID | ค่าต้องเป็น unique id |
| IDREF | ค่าต้องเป็น id ของ element อื่น |
| IDREFS | ค่าต้องเป็นรายการของ id อื่น โดยจะเว้นแต่ละรายการด้วยช่องว่าง |

ตารางที่ 4-1 ชนิดของ Attribute ใน XML

| ค่า | คำอธิบาย |
|----------|---|
| NMTOKEN | ค่าต้องเป็นชื่อที่ประกอบด้วยตัวอักษร , ตัวเลข , เครื่องหมายจุด (.) , เครื่องหมายขีดกลาง (-) , เครื่องหมายขีดล่าง () และสามารถมีเครื่องหมายโคลอน (:) ยกเว้นในตำแหน่งแรกได้ |
| NMTOKENS | ค่าต้องเป็นรายการของ NMTOKEN โดยจะเว้นแต่ละรายการด้วยช่องว่าง |
| ENTITY | ค่าต้องเป็น entity |
| ENTITIES | ค่าต้องเป็นรายการของ entity |
| NOTATION | ค่าต้องเป็นชื่อของ notation (notation คือชนิดของข้อมูลภายนอกที่กำหนดขึ้นเอง เช่น GIF เพื่อบอกว่าเป็นไฟล์รูปภาพ) |
| XML: | ค่าต้องขึ้นต้นด้วย XML |

ตารางที่ 4-1 ชนิดของ Attribute ใน XML (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

default-value สามารถมีค่าได้ดังตารางที่ 4-2

| ค่า | คำอธิบาย |
|----------------|-------------------------------------|
| #DEFAULT value | ค่าดีฟอลต์ของ attribute |
| #REQUIRED | element ต้องมี attribute |
| #IMPLIED | element จะมี attribute หรือไม่ก็ได้ |
| #FIXED value | attribute ต้องเหมือนกับที่กำหนดไว้ |

ตารางที่ 4-2 ค่าดีฟอลต์ของ Attribute ใน XML

- ตัวอย่างของการประกาศ Attribute

DTD example:

```
<!ELEMENT square EMPTY>
<!ATTLIST square width CDATA "0">
```

XML example:

```
<square width="100"></square>
```

ตัวอย่างนี้หมายความว่า square element เป็น empty element กับ attribute width ซึ่งเป็นชนิด CDATA ถ้าไม่ระบุจะยึดค่า default เป็น 0

- ค่าของ Attribute แบบดีฟอลต์

```
<!ATTLIST element-name attribute-name attribute-type "default-value">
```

DTD example:

```
<!ATTLIST payment type CDATA "check">
```

XML example:

```
<payment type="check">
```

ถ้าในเอกสาร XML ไม่มีการระบุจะใช้ค่าดีฟอลต์ แต่ถ้าระบุก็จะใช้ตามที่ระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Attribute โดยนัย

```
<!ATTLIST element-name attribute-name attribute-type #IMPLIED>
```

DTD example:

```
<!ATTLIST contact fax CDATA #IMPLIED>
```

XML example:

```
<contact fax="555-667788">
```

จะใช้ #IMPLIED ในกรณีจะระบุ attribute หรือไม่ก็ได้ ถ้าไม่ระบุก็จะมีค่า default ให้

- Attribute ที่ต้องระบุเสมอ

```
<!ATTLIST element-name attribute_name attribute-type #REQUIRED>
```

DTD example:

```
<!ATTLIST person number CDATA #REQUIRED>
```

XML example:

```
<person number="5677">
```

ใช้ในการบังคับให้ต้องระบุ attribute ถ้าในเอกสาร XML ไม่ระบุ attribute เอกสารนั้นจะไม่ถูกต้อง

- ค่าของ Attribute คงที่

```
<!ATTLIST element-name attribute-name attribute-type #FIXED "value">
```

DTD example:

```
<!ATTLIST sender company CDATA #FIXED "Microsoft">
```

XML example:

```
<sender company="Microsoft">
```

ในแบบนี้ถ้าไม่ได้ระบุค่า attribute จะใช้ค่าดีฟอลต์ แต่ถ้าจะระบุต้องระบุให้ตรงกับค่าดีฟอลต์ มิฉะนั้นจะถือว่าผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ค่าของ Attribute กำหนดเอง

```
<!ATTLIST element-name attribute-name (en1|en2|..) default-value>
```

DTD example:

```
<!ATTLIST payment type (check|cash) "cash">
```

XML example:

```
<payment type="check">
```

หรือ

```
<payment type="cash">
```

จะใช้เมื่อเราต้องการกำหนดค่าของ attribute เอง

4.1.6.4 การกำหนด Entity ใน DTD

- การประกาศ Entity ภายใน

```
<!ENTITY entity-name "entity-value">
```

DTD Example:

```
<!ENTITY writer "Donald Duck.">
```

```
<!ENTITY copyright "Copyright W3Schools.">
```

XML example:

```
<author>&writer;&copyright;</author>
```

- การประกาศ Entity ภายนอก

```
<!ENTITY entity-name SYSTEM "URI/URL">
```

DTD Example:

```
<!ENTITY writer
```

```
SYSTEM "http://www.w3schools.com/entities/entities.XML">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<!ENTITY copyright
SYSTEM "http://www.w3schools.com/entities/entities.dtd">
```

XML example:

```
<author>&writer;&copyright;</author>
```

4.1.7 XML Namespace

XML namespace จะหาวิธีหลีกเลี่ยงความสับสนที่เกิดจากชื่อ element (Name Conflicts)

- ความสับสนที่เกิดจากชื่อ

ความสับสนที่เกิดจากชื่อ element จะเกิดขึ้นบ่อยเมื่อเอกสารที่แตกต่างกัน ใช้ชื่อเดียวกันในการอธิบายชนิดของ element ที่แตกต่างกัน เช่น table ใน 2 ตัวอย่างต่อไปนี้

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

และ

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

ถ้านำเอกสาร XML ทั้ง 2 มารวมกัน จะเกิดความสับสนจากชื่อขึ้นเพราะทั้ง 2 มี element (table) ซึ่งมีใช้ในความหมายที่ต่างกัน

- การแก้ปัญหาความสับสนที่เกิดจากชื่อโดยใช้ Prefix

เอกสาร XML ทั้ง 2 คือ

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<h:td>Bananas</h:td>
</h:tr>
</h:table>
```

และ

```
<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

ความสับสนที่เกิดจากชื่อจะถูกกำจัดไปโดยการเพิ่ม prefix ทำให้ได้ element ที่ต่างกัน 2 ชนิด (<h:table> และ <f:table>)

- การใช้ Namespaces
จากเอกสาร XML ทั้ง 2 คือ

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

และ

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

เราสามารถเพิ่ม attribute xmlns ลงในแท็ก เพื่อบอกชื่อที่ระบุไว้ ที่เกี่ยวข้องกับ namespace

- **Attribute Namespace**
attribute namespace จะวางในแท็กเริ่มต้น (start tag) ตามไวยากรณ์ ดังนี้

```
Xmlns:namespace-prefix="namespace"
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างข้างบน namespace จะสามารถกำหนดด้วย Internet address

```
Xmlns:f="http://www.w3schools.com/furniture">
```

จาก namespace specification ของ W3C กล่าวไว้ว่า namespace สามารถเป็น Uniform Resource Identifier (URI)

ถ้า namespace ถูกกำหนดในแท็กเริ่มต้นแล้ว child element ทุกตัวที่มี prefix เหมือนกันจะต้อง มีความสัมพันธ์กับ namespace เดียวกันด้วย

และ address ที่ใช้บอก namespace ใช้เพื่อให้ชื่อมีความเป็นเอกลักษณ์ไม่ได้ใช้เป็น address เพื่อ อ้างอิงหาข้อมูล แต่หลายบริษัทใช้ namespace เพื่อเป็นตัวชี้ไปยัง web page ที่มีอยู่จริงซึ่งเก็บข้อมูลเกี่ยว กับ namespace ไว้

■ Default Namespaces

การกำหนด default namespace จะป้องกันการใช้ prefix ใน child element ทุกตัว มีไวยากรณ์ดัง
นี้

```
<element xmlns="namespace">
```

ซึ่งจากตัวอย่างที่ผ่านมาจะได้ดังนี้

```
<table xmlns="http://www.w3.org/TR/html4/">
```

```
<tr>
```

```
<td>Apples</td>
```

```
<td>Bananas</td>
```

```
</tr>
```

```
</table>
```

และ

```
<table xmlns="http://www.w3schools.com/furniture">
```

```
<name>African Coffee Table</name>
```

```
<width>80</width>
```

```
<length>120</length>
```

```
</table>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.8 XML PCDATA and CDATA

Parsable Character Data (PCDATA) คือข้อมูลแบบเท็กซ์ที่จะถูก parse โดย parser

Character Data (CDATA) คือข้อมูลแบบเท็กซ์ที่ไม่ถูก parse โดย parser

4.1.8.1 PCDATA

- XML Parser จะมองทุกเท็กซ์ที่เป็น Parsable Characters (PCDATA) เมื่อ XML ถูก parse เท็กซ์ที่อยู่ระหว่าง XML แท็กจะถูก parse ไปด้วย

```
<message>This text is also parsed</message>
```

parser ทำเช่นนี้เนื่องจาก XML element สามารถมีหลาย element อยู่ภายใน ดังในตัวอย่าง

```
<name><first>Bill</first><last>Gates</last></name>
```

และ parser จะแบ่งเป็น sub-element ดังนี้

```
<name>
  <first>Bill</first>
  <last>Gates</last>
</name>
```

- Escape Character

XML character ที่ไม่สามารถใช้ได้จะถูกแทนที่ด้วย entity reference

ถ้าใส่ตัวอักษร "<" ใน XML element จะทำให้เกิดความผิดพลาด เพราะ parser จะแปลเป็นจุดเริ่มต้นของ element ใหม่ ดังนั้นจึงไม่สามารถเขียน อย่างนี้ได้

```
<message>if salary < 1000 then</message>
```

การหลีกเลี่ยง สามารถทำได้โดย แทนที่ "<" ด้วย entity reference ดังนี้

```
<message>if salary &lt; 1000 then</message>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

entity reference มาตรฐานที่ได้กำหนดมาให้ก่อนแล้วใน XML มีทั้งหมด 5 ตัว ดังตารางที่ 4-3

| Entity Reference | ค่า | ความหมาย |
|------------------|-----|----------------|
| < | < | less than |
| > | > | greater than |
| & | & | ampersand |
| ' | ' | apostrophe |
| " | " | quotation mark |

ตารางที่ 4-3 Entity Reference มาตรฐาน

entity reference จะขึ้นต้นด้วย & และลงท้ายด้วย ;

เฉพาะ “<” และ “&” เท่านั้นที่ไม่สามารถใช้ได้ใน XML ส่วนตัวที่เหลือควรทำให้เป็นนิสย

4.1.8.2 CDATA

parser จะไม่ parse ทุกอย่างที่อยู่ในส่วนของ CDATA

ถ้าเท็กซ์มี “<” หรือ “&” อยู่ภายใน XML element นั้นจะถูกกำหนดเป็นส่วนหนึ่งของ CDATA ส่วนของ CDATA จะเริ่มต้นด้วย “<![CDATA[“ และลงท้ายด้วย “]]>”

```
<script>
<![CDATA[
function matchwo(a,b)
{
if (a < b && a < 0) then
{
return 1
}
else
{
return 0
}
}
]
```

```
]]>
</script>
```

จากตัวอย่างข้างต้น ทุกอย่างที่อยู่ในส่วนของ CDATA จะถูกเพิกเฉยโดย parser

4.1.9 XML Parser

XML parser คือโปรแกรมที่ใช้ในการอ่าน สร้าง และจัดการกับเอกสาร XML รวมถึงตรวจสอบความถูกต้องของเอกสาร XML parser สามารถเขียนได้ด้วยภาษาใด ๆ ก็ได้ ในบราวเซอร์ IE 5.0 ขึ้นไปก็มี Microsoft XML parser ทำให้ IE สามารถแสดงผลเอกสาร XML ได้ XML parser ของภาษาจาวาอย่างเช่น XML parser ของซัน (<http://java.sun.com/xml>) หากเราต้องการสร้างแอปพลิเคชันที่สามารถจัดการกับ XML เราก็ต้องมี parser ของภาษานั้นและเขียนโค้ดติดต่อกับ API ของมัน

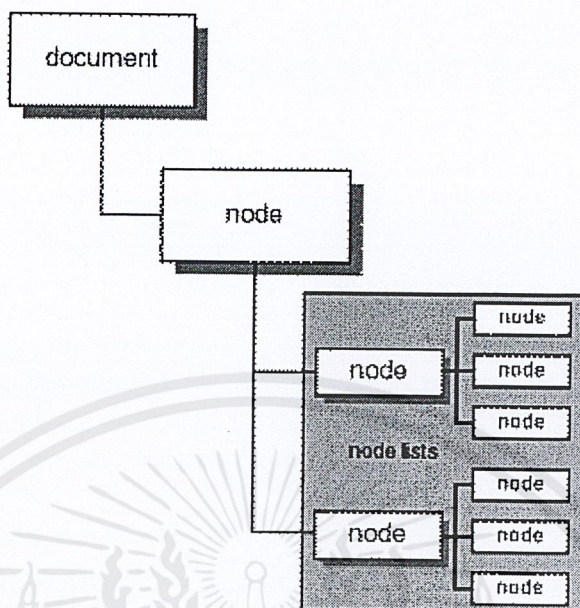
XML parser มี 2 ชนิดคือ

- DOM (Document Object Model)
- SAX (Simple API for XML)

4.1.9.1 DOM

DOM จะใช้วิธี parse เอกสาร XML เป็นลำดับชั้น (hierachy) ของออบเจ็กต์ดังนี้ (ดังรูปที่ 4-1)

- 1) Document Object – แหล่งข้อมูล XML
- 2) Node Object – Parent node ของ child node
- 3) Node List Object – list ของ sibling node
- 4) Parse Error – ใช้ในการรับค่าผิดพลาดที่เกิดจากการ parse



รูปที่ 4-1 โครงสร้างของ DOM

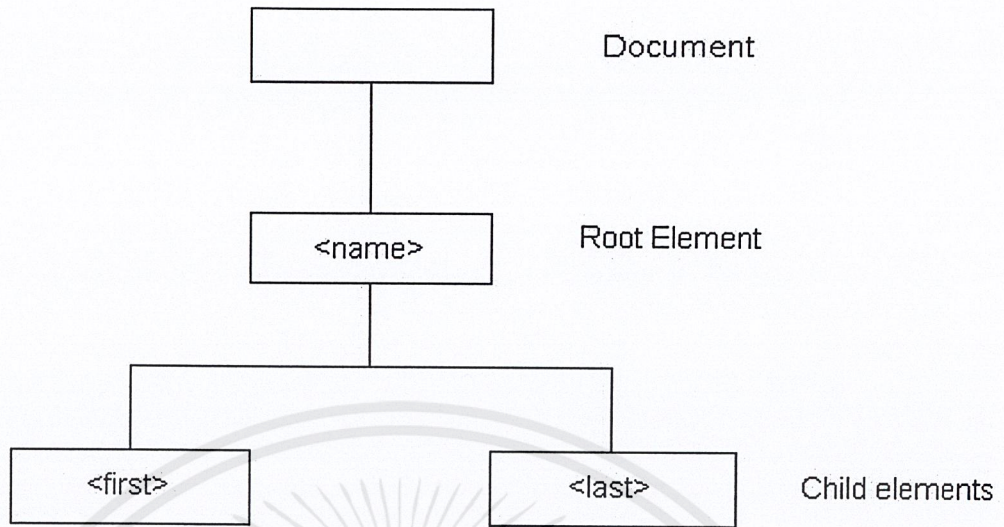
หากเอกสาร XML เป็นดังนี้

```

<name>
  <first>Jim</first>
  <last>Jones</last>
</name>
  
```

จะได้ความสัมพันธ์ดังรูปที่ 4-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-2 ความสัมพันธ์ของการ parse ด้วย DOM

4.1.9.2 SAX

SAX (Simple API for XML) วิธีนี้จะเป็นแบบ event-driven โดย event จะประกอบด้วย

- 1) Start Element – จะตอบสนองกับ event นี้เมื่อพบจุดเริ่มต้นของแท็ก XML
- 2) End element – จะตอบสนองกับ event นี้เมื่อพบจุดสิ้นสุดของแท็ก
- 3) Character – จะตอบสนองกับ event นี้เมื่อพบกับตัวอักษร
- 4) Start Document – จะตอบสนองกับ event นี้ในตอนเริ่ม parse
- 5) End Document - จะตอบสนองกับ event นี้ในเมื่อจบการ parse

ดังนั้นจากโค้ดข้างบนจะเกิด event ดังนี้

| | | |
|----------|---|---------------------|
| | → | Start Document |
| <name> | → | Start Element name |
| <first> | → | Start Element first |
| Jim | → | Character Jim |
| </first> | → | End Element first |
| <last> | → | Start Element last |
| Jones | → | Character Jones |
| </last> | → | End Element last |
| </name> | → | Start Element name |
| | → | End Document |

ทั้ง 2 วิธีมีข้อดีข้อเสียต่างกันคือ DOM จะใช้เมมโมรี่และมีการประมวลผลมาก แต่สามารถจัดการได้ง่าย สามารถที่จะแก้ไขค่าได้ ไม่เหมาะกับเอกสาร XML ที่มีขนาดใหญ่ ถูกนำไปใช้ในเบราว์เซอร์ ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะที่ SAX ไม่ต้อง parse เอกสาร XML ทั้งหมดจึงใช้โมเมรีและมีการประมวลผลน้อยกว่า เหมาะกับการสร้างและรับค่าจากเอกสาร จะถูกใช้ในการประมวลผลทางฝั่งเซิร์ฟเวอร์ ตัวอย่าง XML parser ดังตารางที่ 4-4

| ชื่อ | แพลตฟอร์ม | คำบรรยาย |
|----------------------------|---------------------|---|
| ActiveDOM | Microsoft Window | http://www.vivid-creations.com |
| ActiveSAX | Microsoft Window | http://www.vivid-creations.com |
| JavaTM API for XML | Java | http://java.sun.com/xml |
| MS XML | Microsoft Window | http://www.microsoft.com |
| Oracle XML Developer's Kit | Java , C++ , PL/SQL | http://technet.oracle.com/tech/xml |
| Xerces | Java , C++ | http://xml.apache.org |
| XP | Java | http://www.jclark.com/xml |

ตารางที่ 4-4 XML Parser

4.1.10 ประโยชน์จาก XML

สำหรับประโยชน์ของ XML นั้น เป็นด้านความยืดหยุ่นในการใช้งานสำหรับแอปพลิเคชันที่อิงกับ Web Base ที่ใช้ง่ายในการค้นหาข้อมูล มีความยืดหยุ่นในการพัฒนาเว็บ สามารถผสมผสานข้อมูลจากหลายแหล่ง จากแอปพลิเคชันที่ต่างกัน สามารถแสดงข้อมูลแบบต่างๆ และสามารถ update ข้อมูลให้ทันสมัยเสมอ และคาดว่าจะจะเป็นมาตรฐานใหม่ของระบบเปิด ซึ่งนับเป็น format ใหม่สำหรับการส่งข้อมูลบนเว็บที่มากด้วยข้อมูลหลายแบบ แต่ส่งผ่านด้วยเทคโนโลยีที่มีบีบอัดข้อมูลที่ให้ความเร็วได้รับการสนับสนุนจากผลิตภัณฑ์ค่ายไมโครซอฟท์

สถาปัตยกรรม XML

ภาษา XML ได้รับการสนับสนุนจาก W3C ให้นักพัฒนาเว็บแอปพลิเคชันได้หันมาใช้เป็นส่วนประกอบของการพัฒนาเว็บไซด์ เพราะ XML มีประสิทธิภาพและมีความน่าเชื่อถือสูงในการแปลงข้อมูลและโครงสร้างข้อมูลให้สามารถนำไปใช้งาน สถาปัตยกรรม 3 Tier ที่ XML สามารถสร้างขึ้นจากระบบข้อมูลที่ใช้โมเดลของ 3-Tier โครงสร้าง ของข้อมูลต่างๆ สามารถนำมาแสดงตามข้อกำหนด หรือรูปแบบที่ต้องการตามการใช้งานได้

XML เป็นการทำงานในระดับกลาง middle tier ที่สามารถเรียกใช้ฐานข้อมูลได้หลากหลายระบบ ฐานข้อมูลและโอนข้อมูลให้อยู่ในรูปแบบของ XML และมีการให้รายละเอียดเกี่ยวกับตัวข้อมูล โครงสร้างต่างๆ ของระบบฐานข้อมูลได้ XML เป็นระบบเปิดที่นำเสนอข้อมูลในรูปแบบ text ผ่านทาง HTTP เหมือนกับ HTML แต่จะมีคุณสมบัติในการให้ข้อมูลแบบ real time อัปเดตหรือเปลี่ยนแปลงได้ตามข้อกำหนด การแสดงข้อมูลจาก XML ใน HTML จะเป็นการเพิ่มในส่วนของการรายละเอียดข้อมูล ที่มีการเรียกใช้จากแหล่งหรือฐานข้อมูลที่เชื่อมโยงกันในหลายแหล่งเพื่อให้ HTML มีความสมบูรณ์มากขึ้น ในอนาคตเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเว็บหรือการเขียน และสร้าง HTML ไม่จำเป็นต้องมีการเขียนชุดคำสั่งที่ยุ่งยากซับซ้อนมากก็สามารถทำงานร่วมกับระบบข้อมูลได้อย่างมีประสิทธิภาพ XML จะทำการกำหนดค่าสำหรับโครงสร้างข้อมูลที่จะนำไปแสดงใน HTML นอกจากนั้นยังสามารถนำไปสนับสนุนระบบการแลกเปลี่ยนข้อมูลข่าวสารทาง Electronic ได้อย่างดีอีกด้วย

4.1.11 บทสรุปของ XML

XML มีความยืดหยุ่นพอในการนำเสนอข้อมูลแบบต่างๆ ที่สามารถให้รายละเอียดโครงสร้างข้อมูลตามระดับและความต้องการในการนำไปใช้งาน

4.2 SOAP

4.2.1 แนะนำ SOAP

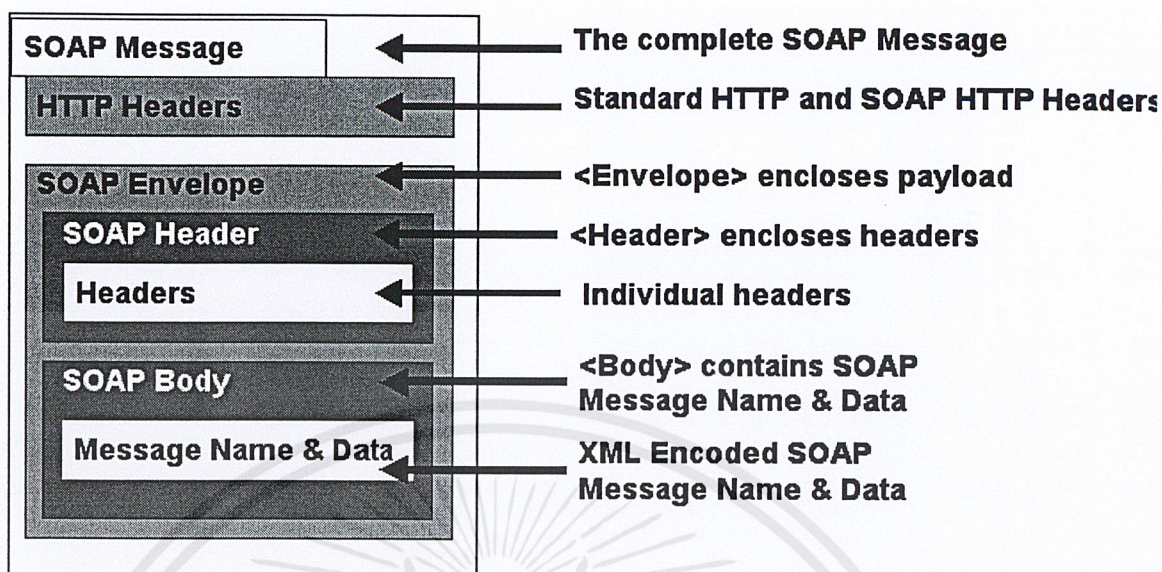
SOAP (Simple Object Access Protocol) เป็นโพรโทคอลที่ใช้ XML เป็นพื้นฐาน เพื่อให้ซอฟต์แวร์คอมพิวเตอร์ และแอปพลิเคชันสามารถติดต่อกันผ่าน HTTP ซึ่งเป็นมาตรฐานอินเทอร์เน็ตโพรโทคอลได้

เนื่องจากการสื่อสารระหว่างแอปพลิเคชันจำเป็นต้องการพัฒนาแอปพลิเคชันบนอินเทอร์เน็ต เป็นอย่างมาก แต่แอปพลิเคชันแบบกระจายกันทำงาน (distributed application) ในปัจจุบันใช้ Remote Procedure Call (RPC) สื่อสารกันระหว่างออบเจกต์ เช่น DCOM และ CORBA ซึ่งไม่ได้ใช้พอร์ตเดียวกับ HTTP ที่เป็นพอร์ตมาตรฐานสำหรับให้บริการเว็บ ดังนั้น RPC จึงนำมาปรับใช้กับอินเทอร์เน็ตได้ยาก และมีปัญหาทางด้านความปลอดภัย ไฟร์วอลล์และพร็อกซีเซิร์ฟเวอร์จะไม่ยอมให้ส่งข้อมูลชนิดนี้ได้ตามปกติวิธีที่ดีกว่าคือใช้ HTTP เพราะเป็นที่ยอมรับโดยอินเทอร์เน็ตบราวเซอร์ และเซิร์ฟเวอร์ทุกชนิด ซึ่ง SOAP ถูกสร้างมาเพื่อใช้ในกรณีนี้

ข้อดีของ SOAP ก็คือ SOAP ไม่ขึ้นกับคอมพิวเตอร์เทคโนโลยี และภาษาการเขียนโปรแกรมใดๆ สามารถเขียนได้ง่ายและขยายเพิ่มเติมได้

4.2.2 ส่วนประกอบของ SOAP

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



รูปที่ 4-3 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 และคืนค่า float กลับมากับ SOAP response SOAP envelope element อยู่ที่จุดบนสุดของเอกสาร XML XML เนมสเปซ 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>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<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>

```

นอกจากนั้นใน SOAP เมสเสจ จะมีการใช้ XML เนมสเปซ ทุกๆ อีลีเมนต์ในเอกสารจะขึ้นต้นด้วย เนมสเปซ เนมสเปซ จะถูกกำหนดโดยใช้ xmlns attribute ดังนี้

```

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

```

แต่หากไม่ต้องการให้ขึ้นต้นด้วย เนมสเปซ ก็สามารถทำได้ ดังนี้

```

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

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับหน่วยงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.1 Envelope

envelope element เป็นอีลีเมนต์บนสุดของเอกสาร XML ที่ใช้แสดงเมสเซจ อาจประกอบด้วย แอตทริบิวต์ คือ envelope เนมสเปซ และ encodingStyle

1. Envelope เนมสเปซ – SOAP เมสเซจจะแสดงเวอร์ชันโดยใช้ เนมสเปซ ของ envelope element เนมสเปซ จะถูกนำไปใช้ขึ้นต้น envelope, header และ body element
2. EncodingStyle Attribute – ใช้แสดงวิธี serialization ของ SOAP เมสเซจ แอตทริบิวต์นี้ สามารถมีได้ในทุกอีลีเมนต์และจะมีผลกับ content ของอีลีเมนต์นั้นและ child element ทั้งหมดของมันที่ไม่ได้ประกาศแอตทริบิวต์

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

4.2.2.3 Header

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

header entry จะต้องเป็นไปตามกฎดังนี้

- header entry จะต้องแสดงด้วยชื่อเต็มประกอบด้วย เนมสเปซ 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" />
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<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 นี้จำเป็นหรือเป็นเพียงอ็อปชันสำหรับผู้รับที่จะนำไปประมวลผล ค่าของมันคือ “1” หรือ “0” ถ้าไม่มีการระบุจะมีค่าเท่ากับ “0” โดย “0” หมายถึงเป็นอ็อปชัน และ “1” หมายถึงจำเป็น โดยจะต้องประมวลผลให้ถูกต้องตาม semantic หรือต้อง fail เช่น เมสเซจเป็นส่วนหนึ่งของ transaction ถ้าปลายทางไม่รองรับ transaction จะต้องไม่ประมวลผลและส่งผลผิดพลาดกลับไปในกรณีที่ mustUnderStand attribute เป็น “1” แต่ถ้าเป็น “0” จะยังคงประมวลผลต่อไป

4.2.3 SOAP Fault Element

ข้อความแสดงความผิดพลาดจากแอปพลิเคชันของ SOAP จะเก็บอยู่ใน fault element ซึ่งถ้ามีจะต้องปรากฏใน 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>

```

```

</detail>
</soap:Fault>
</soap:Body>
</soap:Envelope>

```

SOAP fault element มี element ย่อย ๆ ดังตารางที่ 4-5

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

ตารางที่ 4-5 SOAP Fault Element

ค่าของ faultcode สามารถมีค่าได้ดังตารางที่ 4-6

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

ตารางที่ 4-6 ค่า faultcode

4.2.4 SOAP Encoding

SOAP encoding มีวิธีการ map จาก type ของโปรแกรมมิ่งไปเป็น XML 2 วิธี คือจากภายนอก โดยใช้ WSDL (Web Services Description Language) ที่บอกถึง type ของข้อมูลที่รับหรือส่ง หรือใช้ xsi:type attribute ในกรณีที่ใช้ภาษาที่ใช้ไม่รองรับ WSDL โดยทั้ง 2 วิธีจะใช้ XML schema ในการระบุ type ตัวอย่างของการใช้ xsi จะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<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

4.2.4.1 ตัวอย่าง Struct

struct book ที่เขียนด้วยภาษา c++ จะเป็นดังนี้

```

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

```

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:searchBook xmlns:ns1="http://soapinterop.org/" SOAP-
  ENV: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/"
    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>

```

4.2.4.2 ตัวอย่าง Array

ตัวอย่างนี้เป็น array ของ struct book ในตัวอย่างก่อน 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">

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่บนสื่อออนไลน์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<SOAP-ENV:Body>
  <ns1:BookArrayResult xmlns:ns1="http://soapinterop.org/" SOAP-
  ENV: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>
      </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'>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<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>
</schema>
</types>

```

4.2.5 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 เป็นการ fault 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>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<soap:Fault>
  <faultcode>soap:VersionMismatch</faultcode>
  <faultstring>The SOAP เนมสเปซ is incorrect.</faultstring>
  <faultactor>http://www.wrox.com/endpoint.asp</faultactor>
  <detail>
    <w:errorinfo xmlns:w="http://www.wrox.com/">
      <desc>The SOAP เนมสเปซ was blank.</desc>
    </w:errorinfo>
  </detail>
</soap:Fault>
</soap:Body>
</soap:Envelope>

```

4.2.6 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 method call จำเป็นที่จะต้องมียูเอิลโฟร์เมชันดังนี้

1. URI ของออบเจกต์ปลายทาง (target object)
2. ชื่อของเมธอด
3. คำอธิบายของเมธอด (method signature) ส่วนนี้เป็นอ็อปชัน
4. พารามิเตอร์สำหรับเมธอด
5. ข้อมูลส่วนหัว (header data) เป็นอ็อปชัน

พิจารณาเมธอดดังนี้

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

ถ้า เนมสเปซ ของเมธอดคือ "http://www.worxstox.com/" แล้วการเรียกเมธอดโดย 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

4.2.7 SOAP Toolkit

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

| ชื่อ | แพลตฟอร์ม | แหล่งที่มา |
|---------------------------|------------|---|
| 4s4c | COM | http://www.4s4c.com |
| A SOAP for RPC NT Service | COM | http://www.whitemesa.com |
| Apache Axis | Java | http://xml.apache.org/axis |
| Apache SOAP | Java | http://xml.apache.org/soap |
| CapeConnect | Java | http://www.capeclear.com |
| Glue | Java | http://www.themindelectric.com |
| IBM Web Services Toolkit | Java | http://www.alphaworks.ibm.com |
| IdooXoap for Java and C++ | Java , C++ | http://www.zvon.org |
| Microsoft SOAP Toolkit | COM | http://www.microsoft.com |
| PocketSOAP | COM | http://www.pocketsoap.com |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|--------------------|------|--------------------------|
| SOAP::Lite | Perl | http://www.soaplite.com |
| Visual Studio .NET | COM | http://www.microsoft.com |

ตารางที่ 4-7 SOAP Toolkit

4.2.8 โครงสร้างภายในของ SOAP Toolkit ของ Microsoft

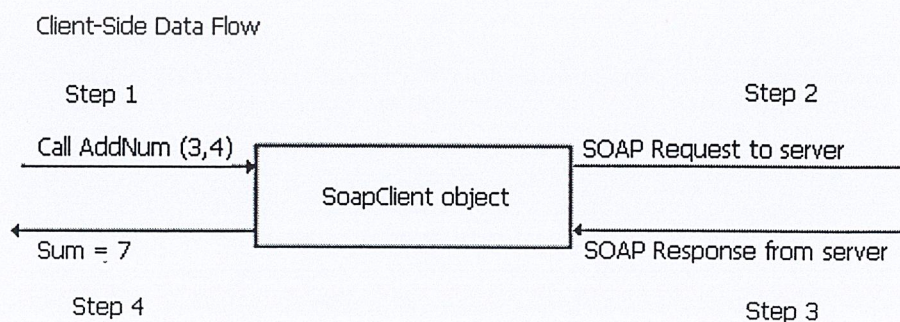
The Web Services Description Language (WSDL) เป็นรูปแบบหนึ่งของ XML เพื่อใช้อธิบายว่า เซิร์ฟเวอร์นั้นมีเซอร์วิส (Services) และ operation ภายในเซอร์วิสอะไรให้ใช้บ้าง ซึ่ง operation ภายใน เซิร์ฟเวอร์นั้นจะอธิบายรูปแบบที่ไคลเอนต์ (client) ต้องใช้ในการเรียก operation

The Web Services Meta Language (WSML) file เป็นส่วนที่เพิ่มขึ้นมาจากไฟล์ WSDL เพื่อใช้ในการเชื่อมโยงแต่ละ operation ในเซอร์วิสกับ COM object โดย WSML file จะพิจารณาว่า COM object ตัวใดที่จะต้องถูกโหลด (load) ขึ้นมาเมื่อมีการเรียกใช้ operation ในเซอร์วิส

ทั้งไฟล์ WSDL และ WSML จะต้องอยู่ที่เซิร์ฟเวอร์ โดยไคลเอนต์ที่ต้องการที่จะส่ง SOAP request ไปที่เซิร์ฟเวอร์ จะต้องได้รับ WSDL จากเซิร์ฟเวอร์ก่อน เพื่อหารูปร่างของ SOAP request ในการติดต่อ

4.2.8.1 การสร้าง SoapClient Object

ทางฝั่งไคลเอนต์ เมื่อแอปพลิเคชันส่ง message ไปที่ Soap Client object เพื่อเรียกใช้ operation หนึ่งๆ นั้น(ตามรูปนี้คือการบวกเลข 2 ตัว) SoapClient object จะทำการสร้าง SOAP request ส่งไปที่ เซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ทำการประมวลผลเสร็จเรียบร้อยแล้ว ก็จะส่งผลลัพธ์ในรูปแบบของ SOAP response กลับมาให้ไคลเอนต์ หลังจากนั้น SoapClient object จึงจะทำการแปลและส่ง message ที่เก็บผลลัพธ์ของ operation นั้นกลับไปให้แอปพลิเคชัน ตามไคอะแกรมด้านล่าง



รูปที่ 4-4 แสดงการ Client-Side Data Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

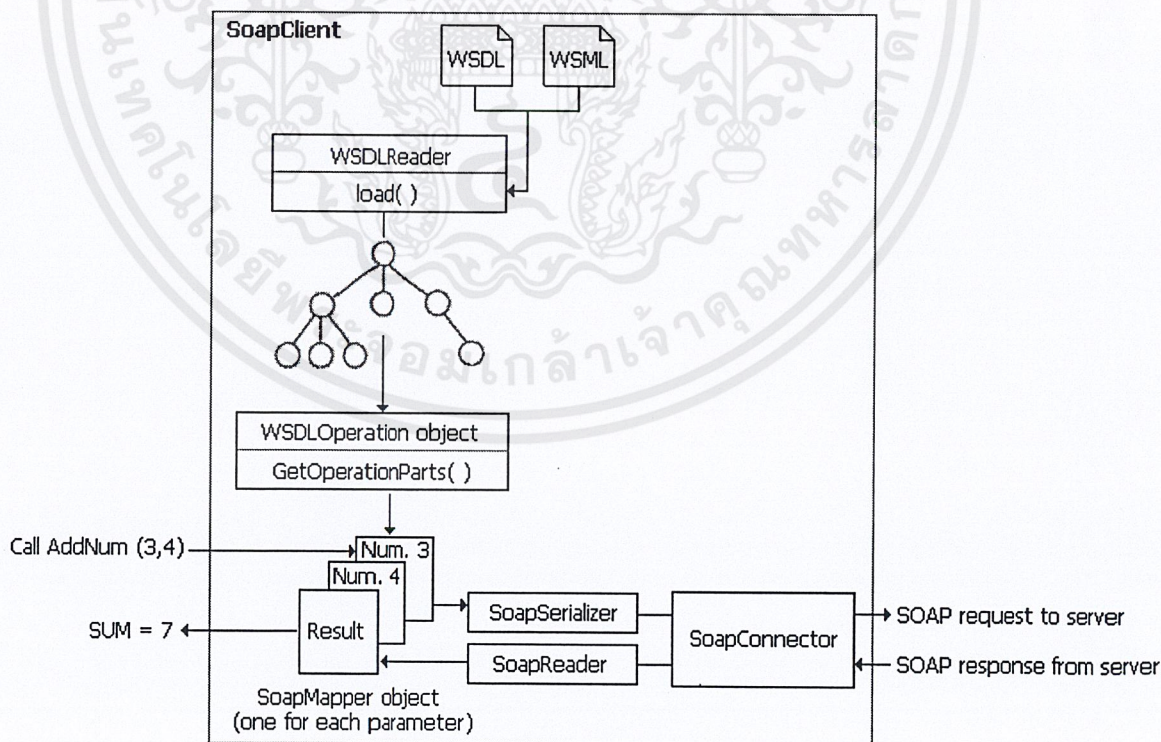
แอปพลิเคชันต้อง instantiate Soap object ก่อนที่จะทำการเรียกใช้ operation ใดๆ (SoapClient object นี้ถูกสร้างเก็บไว้ในไฟล์ MSSOAP1.dll) โดยใช้เมธอด SoapClient.mssoapinit (ชื่อไฟล์ WSDL, ชื่อเซอร์วิส, ข้อมูลพอร์ต) โดยข้อมูลพอร์ตจะใช้ในการแบ่งแยกช่องทางที่ใช้ในการส่ง SOAP message

ในระหว่าง initialize SoapClient object จะตรวจสอบเมธอดทั้งหมดที่อยู่ในไฟล์ WSDL ด้วยการ ทำงานแบบไดนามิกนี้ทำให้เราสามารถที่จะเรียกใช้เมธอดใดก็ได้ที่ได้อธิบายไว้ในไฟล์ WSDL ดังแสดง ตาม 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)
```

4.2.8.2 การประมวลผลภายใน SoapClient Object

การทำงานภายใน SoapClient object แสดงได้ดังรูปที่ 4-5



รูปที่ 4-5 แสดงโครงสร้างภายในของ SOAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 ของอินพุตและเอาท์พุทของ 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 ที่เหมาะสมและส่งไปที่เซิร์ฟเวอร์

เมื่อเซิร์ฟเวอร์ประมวลผลเสร็จแล้วก็จะส่ง SOAP response กลับไปที่ไคลเอนต์ SoapClient object จะทำการโหลดผลลัพธ์ของ operation ไปเก็บใน SoapMapper object ที่เหมาะสมแล้วส่งผลลัพธ์กลับไปให้แอปพลิเคชันของผู้ใช้

4.2.8.3 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 ที่ระบุว่าจะใช้ไฟล์ใดบนเซิร์ฟเวอร์ในการรับ SOAP request เพื่อใช้ในการพิจารณาว่าจะรับ SOAP request อย่างไรบนเซิร์ฟเวอร์ ตามโค้ดด้านล่างนี้ ตัวพิมพ์หนาจะเป็นส่วนที่ใช้ในการเรียกไปที่ ISAPI listener

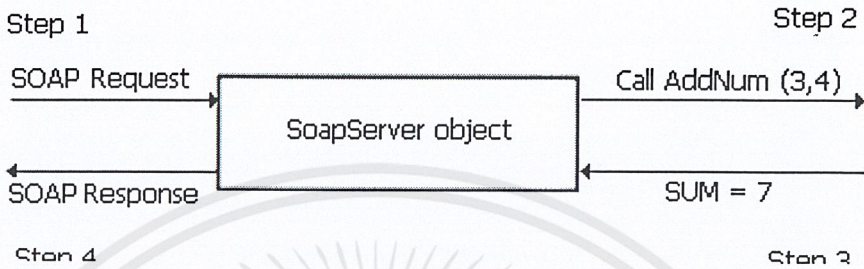
```
<definitions>
...
<service name='DocSample1' >
  <port name='DocSample1PortType' binding='tns:DocSample1Binding' >
    <soap:address
      location='http://localhost/DocSample1Test/DocSample1.wsdl' />
  </port>
</service>
...
</definitions>
```

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) หลังจากนั้นเมธอดที่ถูกเรียกใช้งานก็เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งผลลัพธ์กลับมาให้ SoapServer object ซึ่งมันก็จะแปลงผลลัพธ์ให้อยู่ในรูปของ SOAP response แล้วจึงส่งกลับไปที่ไคลเอนต์

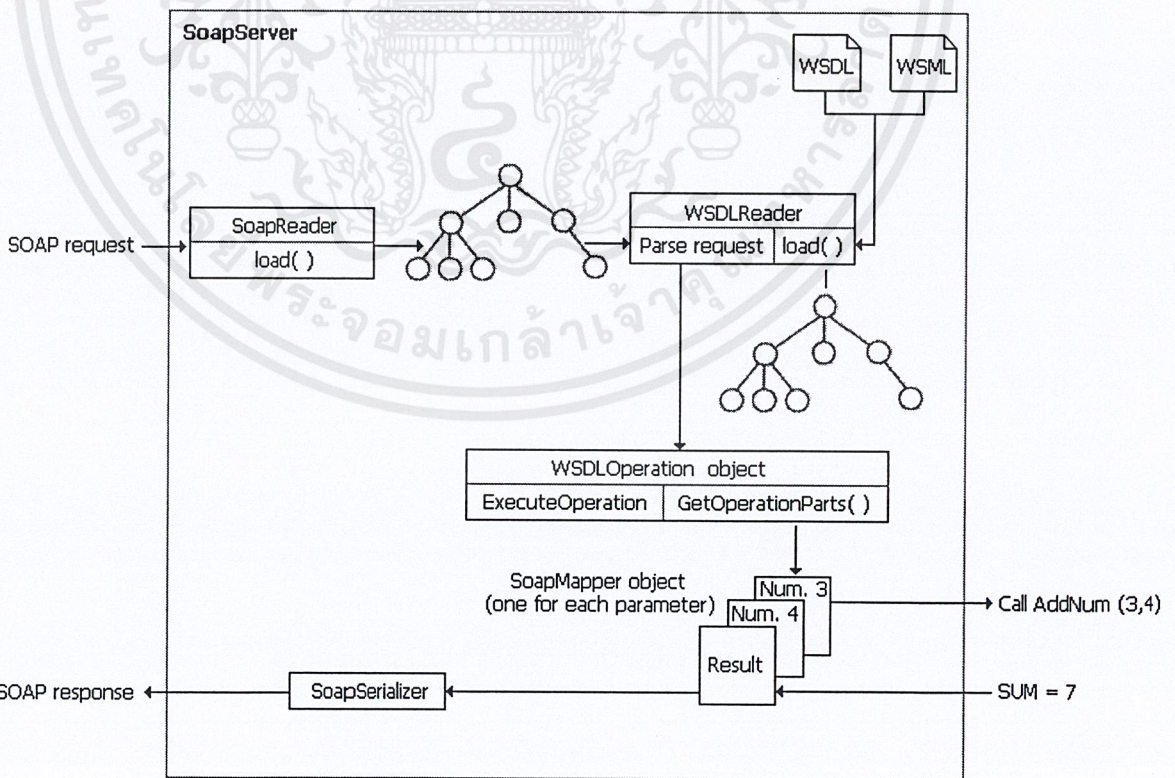
Server-Side Data Flow



รูปที่ 4-6 แสดง Server-Side Data Flow

4.2.8.4 การประมวลผลภายใน SoapServer Object

การทำงานภายใน SoapServer object แสดงได้ดังรูปด้านล่าง



รูปที่ 4-7 แสดงโครงสร้างภายใน SOAP Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

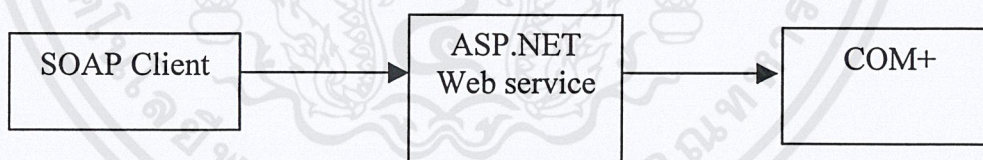
เมื่อได้รับ SOAP request จากไคลเอนต์แล้ว SoapReader object ก็จะโหลด request message ไปไว้ใน DOM structure หนึ่ง ในขณะที่ WSDLReader จะโหลดไฟล์ WSDL และไฟล์ WSML ไปไว้ในอีก DOM structure หนึ่ง ตัว WSDLReader จะแปล request และสร้าง WSDLOperation object ขึ้นเพื่อใช้ใน requested operation หลังจากนั้น WSDLOperation object จะเรียกใช้เมธอด GetOperationParts เพื่อดึงรูปแบบ message ของอินพุตและเอาที่พุทของ operation ออกมา แล้ว SoapServer จะสร้าง Soap Mapper objects ให้กับแต่ละส่วนที่ได้จากเมธอด GetOperationParts และโหลดค่าที่ได้มาจากการเรียกใช้ operation เข้าไว้ใน objects เหล่านี้ พอถึงขั้นตอนนี้ SoapServer object จะทำการเรียกไปที่ COM method ที่ตรงกับที่ร้องขอเข้ามา

เมื่อ COM method ประมวลผลเสร็จแล้วก็จะส่งผลลัพธ์กลับไป SoapServer object ซึ่ง SoapServer object ก็จะทำการโหลดผลลัพธ์ของ operation ไปเก็บใน SoapMapper object ที่เหมาะสม แล้วจึงใช้ SoapSerializer object ส่ง SOAP response message กลับไปให้ไคลเอนต์แอปพลิเคชันของผู้ใช้

4.2.9 การสร้าง SOAP Client และ SOAP Server ด้วย .NET Framework

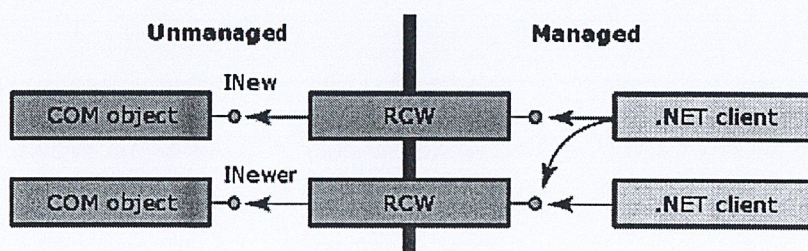
4.2.9.1 การใช้ ASP.NET ทำ SOAP เว็บเซอร์วิส

ใช้ความสามารถในการทำ COM+ Interop โดยสร้างคลาสของ ASP.NET มาเรียกใช้งาน business logic ที่สร้างด้วย Visual Basic 6



รูปที่ 4-8 การใช้ ASP.NET ทำ SOAP เว็บเซอร์วิส

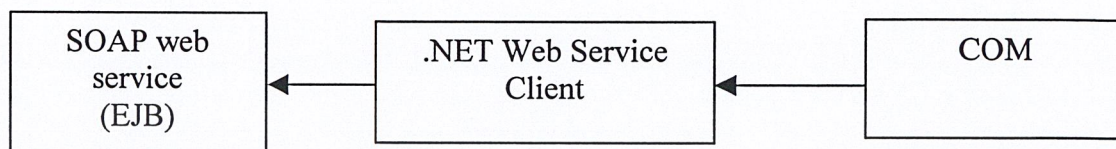
อาศัยความสามารถในการทำ marshall / unmarshall object ให้อยู่ในรูปแบบ SOAP ของ ASP.NET โดย ASP.NET ต้องทำ Interop โดยอาศัย Runtime Callable Wrapper ช่วยจำลอง COM object ให้ .NET มองเห็นเป็น .NET object ธรรมดา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4-9 Runtime Callable Wrapper
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

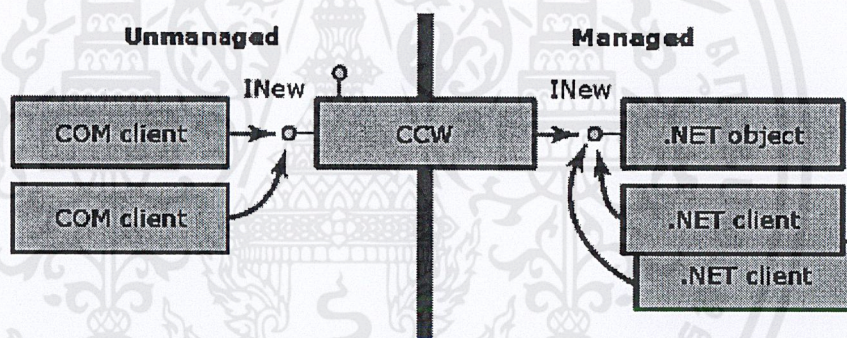
4.2.9.2 การใช้ .NET ทำ SOAP Client

ในทางกลับกันที่ COM ต้องการใช้บริการจาก SOAP ก็กระทำผ่าน .NET เว็บเซอร์วิส client



รูปที่ 4-10 การใช้ .NET ทำ SOAP client

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



รูปที่ 4-11 COM callable wrapper ที่ทำให้ COM client สามารถเห็น .NET object

4.3 UDDI

UDDI (Universal Description, Discovery and Integration) เป็นมาตรฐานที่จัดตั้งขึ้น โดยบริษัท IBM, Microsoft และบริษัทยักษ์ใหญ่ทางธุรกิจ B2B (Business-to-Business) อื่น ๆ UDDI ถูกสร้างขึ้นมาเป็นมาตรฐาน ในการค้นหาบริการ เว็บเซอร์วิส สำหรับคู่ค้าทางธุรกิจ ซึ่งเปรียบได้กับฐานข้อมูล ขนาดใหญ่ ซึ่งมีข้อมูลของ เว็บเซอร์วิส ที่เปิดให้บริการ โดยที่ web site สำหรับค้นหา เว็บเซอร์วิส มีอยู่หลายที่อย่างเช่น

- <http://uddi.microsoft.com/search.aspx>
- <http://www-3.ibm.com/services/uddi/testregistry/find>
- <http://uddi.org>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 WSDL

WSDL (Web Services Description Language) เกิดจากความร่วมมือระหว่าง IBM และ Microsoft WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะของ เว็บเซอร์วิส และวิธีการติดต่อกับ เว็บเซอร์วิส นั้น ๆ โดยใช้ไวยากรณ์ของภาษา XML ซึ่ง WSDL อยู่ในความดูแลของ W3C version ล่าสุด คือ WSDL 1.1 รายละเอียด สามารถหาอ่านเพิ่มเติมได้จากเว็บไซต์ของ W3C ที่ <http://www.w3.org/TR/wsdl> ส่วนในทางปฏิบัติ หากเราต้องการ สร้างเว็บเซอร์วิส ขึ้นมาเป็นของตนเอง ก็สามารถสร้างเอกสาร WSDL ได้โดยอัตโนมัติ เราจึงไม่ต้องไปกังวลในรายละเอียด ในข้อกำหนดใน WSDL มากนักโดยตัวอย่างเอกสาร WSDL เช่น

```
<?xml version="1.0"?>
<definitions name="StockQuote" target namespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd1="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
  <schema target namespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>
</definitions>

```

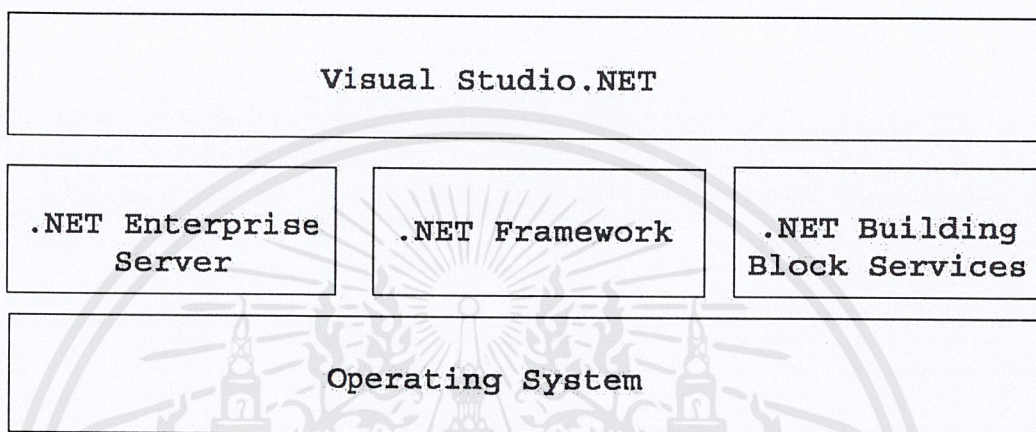
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

Microsoft .NET Technology

5.1 .NET PLATFORM

องค์ประกอบของแพลตฟอร์ม .NET แบ่งออกได้เป็น 3 ชั้น



รูปที่ 5-1 องค์ประกอบ .NET Platform

5.1.1 ชั้นระบบปฏิบัติการ (Operating System)

5.1.2 ชั้นถัดมาแบ่งออกเป็น 3 ส่วน ได้แก่

- .NET Enterprise Server
 - SQL Server 2000, BizTalk Server, Exchange 2000
- .NET Framework
 - เป็น Framework ที่สำคัญในการพัฒนา .NET
- .NET Building Block Services

Microsoft passport, Notification and Messaging, Personalization , XML Store, Calendar, Directory and Search

5.1.3 ชั้น Visual Studio .NET เครื่องมือในการพัฒนา .NET

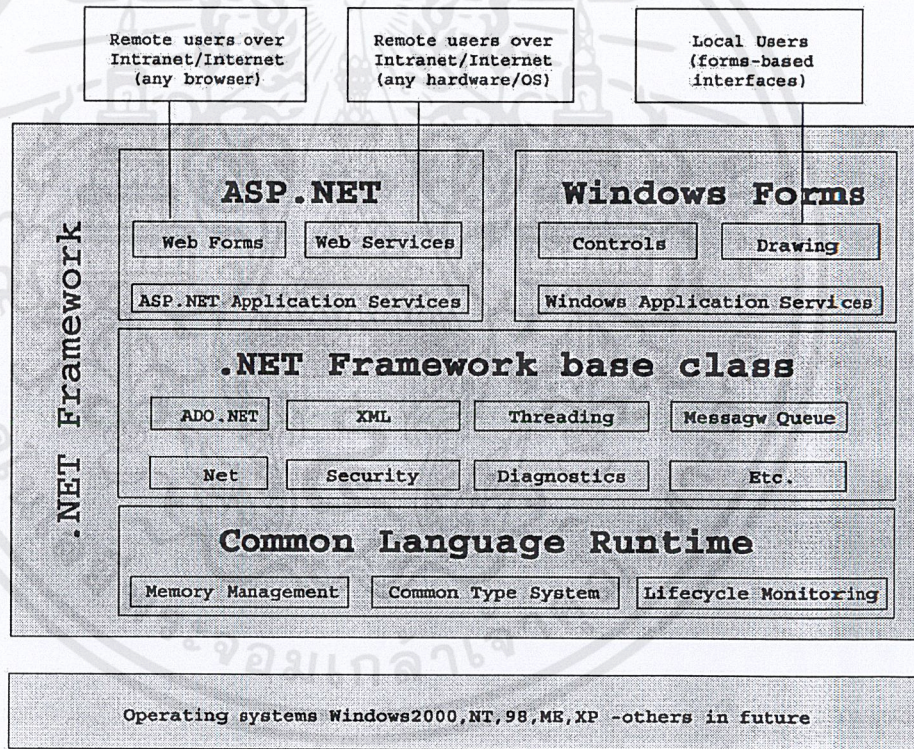
5.2 .NET Framework

.NET Framework มีองค์ประกอบ ดังนี้

1. Common Language Runtime (CLR) มีหน้าที่ในการจัดเตรียมบริการต่างๆที่เกี่ยวข้อง เพื่อให้โปรแกรมที่ขอใช้บริการสามารถทำงานได้

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

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



รูปที่ 5-2 องค์ประกอบ .NET Framework

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

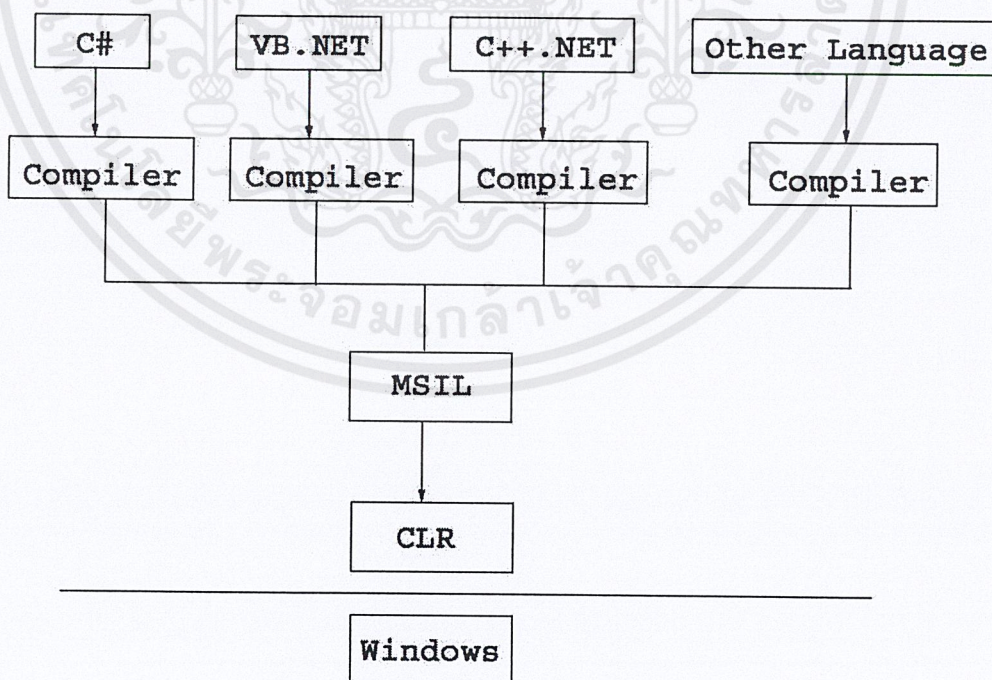
5.2.1 Common Language Runtime (CLR)

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

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

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

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



รูปที่ 5-3 รูปแบบการคอมไพล์โค้ดไปเป็น IL Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Managed Code เป็นโค้ดที่คอมไพล์ให้ทำงานใน .NET Framework

Native Code เป็นโค้ดที่คอมไพล์เป็นภาษาเครื่องโดยตรง ๆ ไม่ต้องผ่านตัว CLR ในการรันโปรแกรม

Unmanaged Code เป็นโค้ดที่ไม่ได้คอมไพล์เพื่อใหทำงานใน .NET Framework เช่น COM Object ใน version ก่อนหน้านี้ ซึ่ง .NET ก็ยังสามารถรันโค้ดพวกนี้ได้ แต่ไม่สามารถใช้ความสามารถของ .NET ได้อย่างเต็มที่

5.2.1.1 Memory Management

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

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

5.2.1.2 Common Type System

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

Equal(Object) = Boolean

GetHashCode() = Int32

GetType () = Type

Tostring () =String

ข้อดีของ Common Language Runtime

- ไม่มีปัญหาเรื่องการรัน บน MS Windows Platform ต่างๆ เนื่องจาก CLR จะทำการตรวจสอบระบบปฏิบัติการให้โดยอัตโนมัติ และทำการจำลองสภาพให้เหมาะสมกับการทำงานของโปรแกรมโดยอัตโนมัติเช่นเดียวกัน

- ไม่ต้องสนใจเรื่อง Registry การเข้าถึง คอมโพเนนต์ ต่างๆที่เกี่ยวข้องจะอยู่ในความดูแล

ของ CLR ทั้งหมด เราจำเป็นต้องรู้แค่การใช้ Namespace ตัวใดในการอ้างอิงถึงเท่านั้น

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

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

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

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

6. ระบบจัดสรรทรัพยากรที่ดีขึ้น การดำเนินการที่เกี่ยวข้องกับทรัพยากรต่างๆ ระหว่างโปรแกรมไม่ว่าจะเป็นระบบการจัดการไฟล์ระบบการเชื่อมต่อ หรือพื้นที่ว่างของหน้าจอ เป็นต้น จะถูกดูแลจัดการอย่างใกล้ชิด โดย CLR ไม่สามารถมีโปรแกรมตัวใดตัวหนึ่งครอบครองทรัพยากรไว้แต่เพียงผู้เดียว

7. การแจกจ่ายโปรแกรมง่ายขึ้น การติดตั้งโปรแกรมแบบเดิม โปรแกรมหนึ่งๆมีไฟล์เป็นจำนวนมาก ,ต้องมีการลงทะเบียนใน registry ,การสร้าง short cut , การจัดเตรียมระบบที่เกี่ยวข้อง ใน .NET การติดตั้งโปรแกรมเป็นไปโดยง่าย เนื่องจาก CLR จะทำการแยกโปรแกรมออกเป็นส่วนๆ การติดตั้งโปรแกรมจึงเป็นการ คัดลอก ไฟล์ที่เกี่ยวข้องมาไว้บน hard disk เท่านั้น ในทางกลับกัน การยกเลิกการติดตั้ง ก็เพียงลบไฟล์เหล่านั้นออก

8. ระบบความปลอดภัยที่ดีขึ้น ระบบรักษาความปลอดภัยใหม่นี้จะทำการตรวจสอบพฤติกรรมการทำงานของโปรแกรมอย่างใกล้ชิด อย่างในระบบเดิมนั้น การทำงานของ ภาษาสคริปต์ต่างๆหรือโปรแกรมที่มาจาก Internet อาจเป็นโปรแกรมที่สร้างช่องโหว่ให้กับระบบได้

5.2.2 .NET Framework Class Library

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

5.2.3 Presentation Layer

การแสดงผลสามารถทำได้ 2 รูปแบบหลัก คือในรูปแบบ Windows form และ Web form ซึ่งก่อนนี้ทั้ง คำสั่ง และกระบวนการเขียนโปรแกรม ของทั้ง 2 แบบ แต่ใน .Net แล้วได้รวมทั้งสองอย่างมาทำให้ เหมือนกันที่สุด คือเราเขียนโปรแกรมบน Windows อย่างไร คุณก็เขียน Web Application ด้วยวิธีเดียวกัน โดยเรียกใช้ คลาส ด้วยคำสั่งที่เหมือนกัน บน Windows จะอยู่ใน คลาส ที่ชื่อ system.windows แต่ Web จะอยู่ใน คลาส ชื่อ system.web.ui ซึ่งวิธีใช้เหมือนกันมาก ที่พิเศษคือ มี Object สำหรับ Web ให้คุณเขียน โปรแกรม แบบ visual basic ธรรมดา แต่เมื่อรันบน browser แล้วจะสร้าง tag html และ javascript ให้ อัตโนมัติ ทำให้ browser ทั่วๆ ไปก็สามารถรันได้ ไม่จำเป็นต้องสนับสนุน .net โดยเฉพาะ ตัวอย่างเช่น เราสร้างเว็บเพจที่มี ปฏิทิน ด้วย(นึกถึง calendar ใน VB) ในส่วนของ รูปแบบ web จะแสดง ในรูปของ html code ส่วนปฏิทินจะอยู่ในรูปของ java script ซึ่งทาง โปรแกรมจะทำการแปลงให้เอง

5.3 .NET Compact Framework

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

คุณสมบัติหลักของ .NET Compact Framework

- .NET Compact Framework ถูกออกแบบให้เป็น platform ในการพัฒนา สำหรับการเขียน และเรียกใช้ XML เว็บเซอร์วิส
- เนื่องจาก .NET Compact Framework ใช้ programming model เดียวกันกับ desktop .NET Framework จึงง่ายต่อ developer ในการเขียน application ใหม่บน .NET Compact Framework
- ใช้ Visual Studio .NET ตัวเดียวกับที่ใช้ใน desktop .NET Framework ในการเขียน program device application

- ใช้ระบบ evidence-base security ที่อนุญาตให้เฉพาะ application ที่ผ่านการรับรองจากผู้ผลิต device นั้น จึงจะสามารถ access secure resource ได้ ถ้าเป็น application ที่ยังไม่ผ่านการรับรองความปลอดภัย จะ access ได้เพียง basic system resource เท่านั้น
- ออกแบบสำหรับอุปกรณ์ที่มี resource จำกัด เช่น โทรศัพท์มือถือ โดยที่ตัว framework เองใช้ memory อย่างมีประสิทธิภาพ และมีระบบดึง resource ที่ไม่ได้ใช้แล้วกลับมาจาก application อย่างมีประสิทธิภาพ
- ใช้ Just-in-time compiler ซึ่งมีประสิทธิภาพดีกว่า ระบบ device programming อื่นๆ ที่ใช้ code interpreter
- เพิ่มประสิทธิภาพและลดต้นทุนในการพัฒนา เนื่องจากใช้ programming model และ toolset ของ Visual Studio .NET อันเดียวกันกับของ desktop .NET framework ทำให้นักพัฒนาสามารถใช้ความรู้ความชำนาญที่มีอยู่แล้วมาพัฒนา application บน อุปกรณ์ได้

5.4 .NET Framework on Linux

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

5.4.1 .NET Framework Runtime (CLR)

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

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

5.4.2 NET Framework based class Library

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

5.4.3 C# Compiler

Ximian ทำคอมไพเลอร์ภาษา C# ที่ชื่อว่า mcs (Mono CSharp compiler)

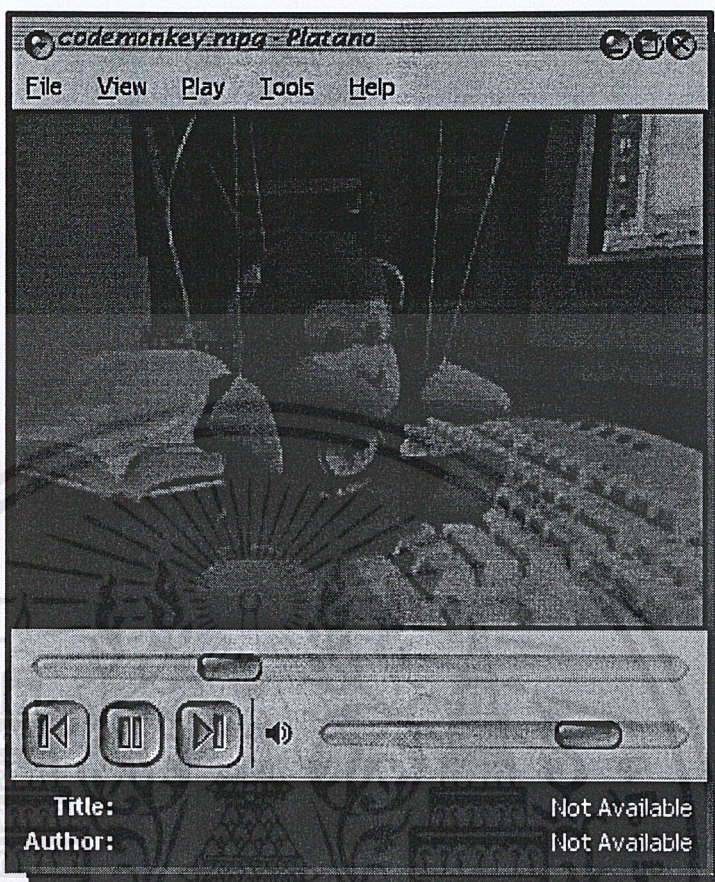
ตัวอย่างแอปพลิเคชัน .NET Framework บนลินุกซ์

- Platano Media Player เขียนโดยใช้ Gtk#



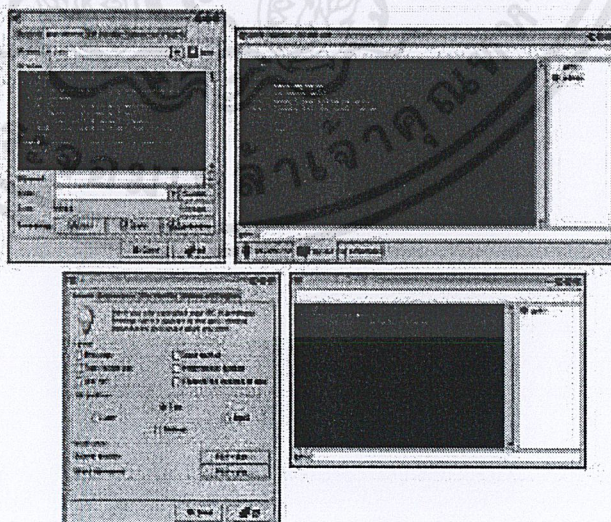
รูปที่ 5-4 Platano Media Player Startup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-5 Platano playing the Code Monkey At Work video.

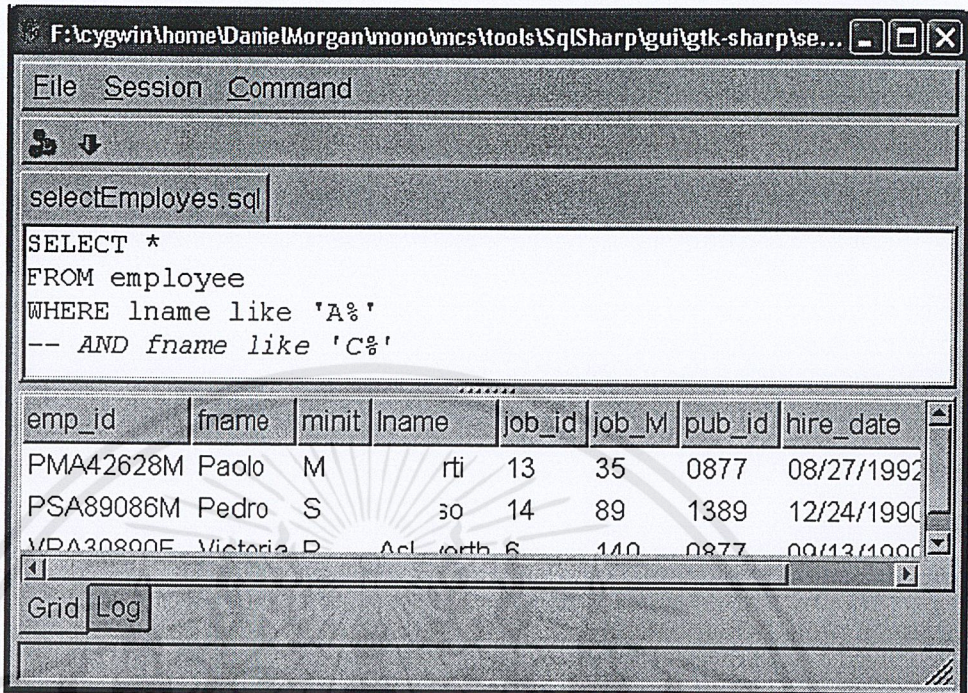
- GSIRC เป็น โปรแกรมแชทที่พัฒนาจาก Gtk#



รูปที่ 5-6 GSIRC is shown with the preferences dialog and chatting on some channels (rooms).

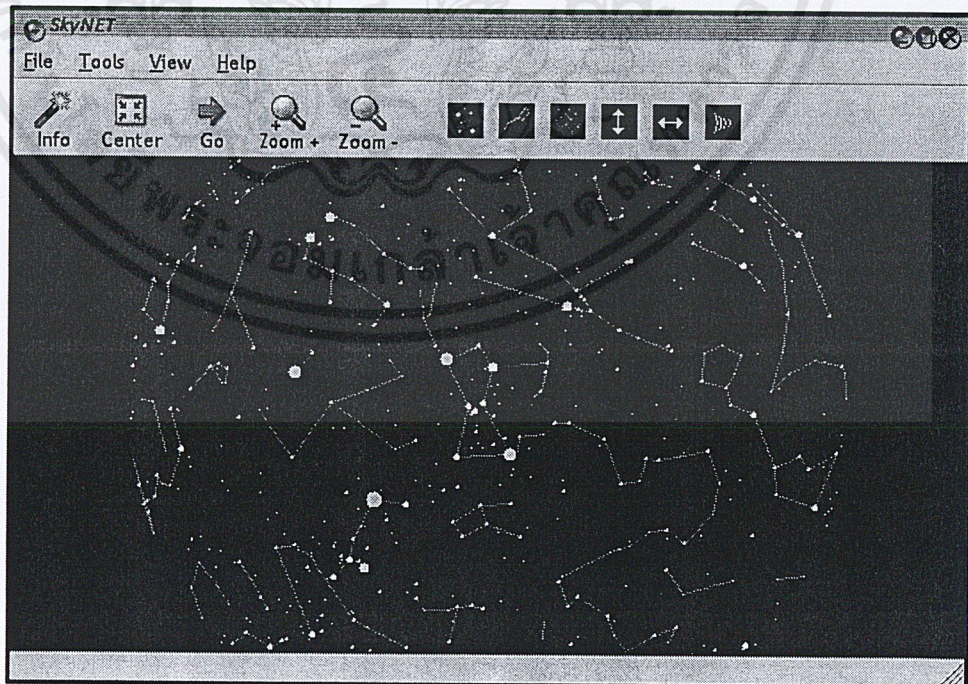
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SQL# สำหรับ Gtk# พัฒนาจากภาษา C# สามารถติดต่อกับ ADO.NET และคำสั่ง SQL



รูปที่ 5-7 SQL# For GTK# is shown retrieving data from a Microsoft SQL Server 2000 database

- SkyNet พัฒนาจากภาษา C# และ ใช้ Gtk# ในการแสดงผลดวงดาว



รูปที่ 5-8 ตัวอย่างแอปพลิเคชัน SkyNet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 มาตรฐาน CLI และ C# ของ ECMA

เดือนสิงหาคม ปี2000 ไมโครซอฟท์ ฮีเลดต์แพกการ์ด และ อินเทล ได้ร่วมเป็นผู้สนับสนุนร่วม เสนอการกำหนดรายละเอียดสำหรับ CLI และ ภาษา C# ให้กับองค์การจติมาตรฐานระหว่างประเทศ ECMA ผลคือ ECMA ได้ตั้งมาตรฐานกลุ่มการทำงานขึ้นมา 2 กลุ่ม สำหรับ CLI และ C# คือ TC39-TG3 และ TC39-TG2 ตามลำดับซึ่งเป็นคณะกรรมการเทคนิคซึ่งรับผิดชอบเกี่ยวกับภาษาในการเขียนโปรแกรม และการพัฒนาแอปพลิเคชัน

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

อ้างอิงจาก

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

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

<http://gtk-sharp.sourceforge.net/>

บทที่ 6

คอมโพเนนต์ .NET (Assembly)

6.1 วิวัฒนาการของคอมโพเนนต์

ในยุคสมัยที่ระบบปฏิบัติการยังใช้ดอส (DOS) กันอยู่นั้น โปรแกรมที่เขียนขึ้นเป็นเพียงโปรแกรมเล็ก ๆ เท่านั้น แต่ในยุคสมัยที่วินโดวส์ 16 บิต ได้ออกมา การเขียนโปรแกรมบนวินโดวส์นั้นจะมีขนาดใหญ่กว่าการเขียนโปรแกรมบนดอสมาก จึงได้มีการพัฒนาแชร์ไลบรารี (Share Library) ขึ้นมา เพื่อลดขนาดของโปรแกรมที่พัฒนาบนวินโดวส์ลง ซึ่งแชร์ไลบรารีนี้ก็คือ ไดนามิกลิงค์ไลบรารี (Dynamic Linked Library - DLL) นั่นเอง

มาถึงยุคสมัยวินโดวส์ 95 วินโดวส์ 95 ได้กลายเป็นระบบปฏิบัติการหลักที่มาแทนที่ดอส ทำให้ทางผู้พัฒนาแอปพลิเคชันพัฒนาแอปพลิเคชันต่างๆ ที่ใช้กับวินโดวส์ 95 ขึ้นมามากมาย ซึ่งจากการที่มีบริษัทที่พัฒนาแอปพลิเคชันอยู่หลายบริษัท ต่างฝ่ายต่างทำการพัฒนาแอปพลิเคชันของตน ทำให้มีการใช้ชื่อ DLL ที่เหมือนกัน เมื่อผู้ใช้ลงแอปพลิเคชันที่มี DLL เหมือนกัน แต่คนละบริษัทที่พัฒนา ทำให้เกิดการทับกันของ DLL ซึ่งอาจทำให้ไม่สามารถใช้งานแอปพลิเคชันแรกที่ได้ลงไว้แล้วได้เลย หรืออาจจะเกิดการที่บริษัทผู้พัฒนาได้มีการพัฒนา DLL เวอร์ชันใหม่ขึ้นมา แอปพลิเคชันเก่าที่ใช้ DLL เวอร์ชันเก่าอาจจะไม่สามารถใช้งาน DLL เวอร์ชันใหม่ได้

ทางไมโครซอฟท์จึงได้ทำการพัฒนา DLL ใหม่ขึ้น ซึ่งรู้จักกันในนามคอมโพเนนต์ (COM Component) ซึ่งจะมีความสามารถในการทำอินเทอร์เฟซอินเฮอริเทนต์ (Interface Inheritance) โดยใน DLL ชนิดเดียวกันจะมีหลายเวอร์ชันได้ โดยที่ DLL เวอร์ชันเก่าจะต้องเข้ากันได้กับ DLL เวอร์ชันใหม่ด้วย

จากการที่ไมโครซอฟท์ได้พัฒนา DLL รุ่นที่ 2 ขึ้นมา แต่กลับแก้ปัญหาได้ไม่ดีเท่าไรนัก ทางไมโครซอฟท์จึงได้พัฒนา DLL รุ่นที่ 3 ขึ้นมาโดยใช้เทคโนโลยีที่เรียกว่า ฟิวชันเทคโนโลยี (Fusion Technology) โดยมีแนวทางที่ว่า DLL แต่ละตัวจะถือว่าเป็นคนละแอมเบิลกัน เพื่อไม่ให้เกิดการทับกัน ซึ่งในรุ่นที่ 3 นี้ DLL จะไม่เพียงแต่เก็บโค้ด แต่จะเก็บข้อมูลอธิบายถึงองค์ประกอบของ DLL รวมทั้งเวอร์ชันของ DLL แต่ละตัวด้วย เรียกว่าเมตาดาต้า (Metadata) ซึ่ง DLL รุ่นที่ 3 นี้ ทางไมโครซอฟท์เรียกว่า แอสเซมบลี (Assembly) โดยแอสเซมบลีนี้จะมีอยู่ 2 ชนิดด้วยกัน คือ

1. ไพรเวท แอสเซมบลี (Private Assembly) ซึ่งจะเก็บอยู่ในโฟลเดอร์ที่ติดตั้งแอปพลิเคชัน โดยจะไม่ไปยุ่งเกี่ยวกับโฟลเดอร์อื่นๆ
2. แชร์แอสเซมบลี (Shared Assembly) แอสเซมบลีชนิดนี้จะถูกเก็บอยู่ใน โกลบอลแอสเซมบลีแคช (Global Assembly Cache - GAC) ซึ่งจะอยู่ใน WINNT\Assembly\GAC โดยเพื่อให้ไม่เกิดการทับกันขึ้น ทางไมโครซอฟท์ใช้หลักการเข้ารหัสแบบ RSA โดยที่กำหนดให้บริษัททุกบริษัทที่จะนำเอาแอส

แชมป์ลี ไปเก็บไว้ใน GAC ได้ นั่นต้องสร้างคีย์ (Key) 2 ตัว นั่นก็คือคีย์สาธารณะ (Public Key) และคีย์ส่วนตัว (Private Key) โดยทางไมโครซอฟท์จะนำเอา 8 ไบต์หลังของคีย์สาธารณะของบริษัทนั้นมาสร้างเป็นชื่อไคเรกทอรี เช่น 1.0.2411.0_b77a5c561934e089 ส่วนแรก 1.0.2411.0 คือเวอร์ชัน และส่วนที่สอง b77a5c561934e089 นั้นเป็น 8 ไบต์สุดท้ายของคีย์สาธารณะของไมโครซอฟท์ โดยชื่อไคเรกทอรีที่เกิดจากคีย์สาธารณะนี้ ทางไมโครซอฟท์ได้เรียกว่า สตรองเนม (Strong Name)

6.2 Assemblies

ในระหว่างยุคของ COM เอกสารของไมโครซอฟท์ ใช้คำว่า คอมโพเนนต์ ในความหมายของ COM คลาส หรือ COM โมดูล (DLLs หรือ EXEs) ใน .NET ไมโครซอฟท์ ได้เตรียมพร้อมกับความสับสนนี้โดยแนะนำนิยามใหม่ที่เรียกว่า assembly ซึ่ง คือ ซอฟต์แวร์ คอมโพเนนต์ ที่สนับสนุน plug-and-play คล้าย ฮาร์ดแวร์ คอมโพเนนต์ .NET assembly นั้นมีความหมายเหมือนกับ COM โมดูล ในทางปฏิบัติ assembly สามารถบรรจุ หรือ อ้างถึง จำนวนของ type หรือ ไฟล์ทางกายภาพ (รวมทั้ง bitmap file, .NET PE files) ที่จำเป็นขณะรันไทม์ สำหรับการเอคซิคว ให้สมบูรณ์ นอกจากนี้ยังเป็นที่อยู่ของ IL โค้ด assembly นั้น เป็น หน่วยพื้นฐาน สำหรับการกำหนด version การ deployment การบริหารความปลอดภัย การเอคซิควแบบ side-by-side การแชร์ การนำกลับมาใช้ใหม่

6.2.1 Unique Identities

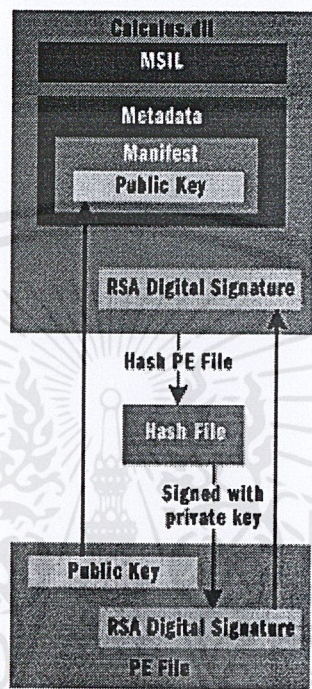
type ที่มีความเป็นเอกลักษณ์ มีความสำคัญ ใน COM และ .NET การให้จำนวนหลากหลายของ GUIDs ใน COM การพัฒนา และการตีผลอย นำเบื้อเพราะ จะต้องใช้ตัวเลขเหล่านี้ในโค้ด ใน .NET จะต้องอ้างอิงถึง type ที่เจาะจงโดยมันสามารถอ่านชื่อและ namespaces ของมันได้ เพราะว่า ชื่อที่อ่านได้ของมัน และ namespaces ไม่สามารถที่จะเป็นเอกลักษณ์ได้ .NET จึงใช้ unique public/private key pairs ทำให้ assembly ทั้งหมดนั้นสามารถแชร์ได้ โดยที่แอปพลิเคชัน ทั้งหมดต้องสร้างด้วย public/private key pair โดย assembly creator สามารถ sign assembly ด้วย private key และ ใครก็ตามสามารถ ตรวจสอบ digital signature นั้นโดยใช้ public key ของ assembly creator

ในการ sign assembly นั้น คุณจะต้องใช้ public/private key pair ขณะการสร้าง assembly คอมไพเลอร์สร้าง hash ของ assembly files แล้ว sign hash นั้นด้วย private key และ เก็บผลของ digital signature ในส่วนที่จองไว้ของ PE ไฟล์ public key ก็จะทำกรเก็บไว้ใน assembly ด้วย

ในการตรวจสอบ digital signature ของ assembly CLR จะใช้ public key ของ assembly ในการถอดรหัส digital signature โดย CLR จะใช้ข้อมูลใน manifest ของ assembly ในการสร้าง hash แบบเปลี่ยนแปลงตลอดเวลา โดยค่าของ hash นี้จะเปรียบเทียบกับค่าของ hash ดั้งเดิม ค่าเหล่านี้จะต้องตรงกัน

เมื่อคุณหรือใครบางคน สร้าง แอปพลิเคชันขึ้นมาต้องการใช้แชร์ assembly assembly manifest ของ แอปพลิเคชันจะต้องรวม 8-byte hash ของ public key ของ แชร์ assembly ด้วย เมื่อทำการรันแอปพลิเคชัน

เครื่อง CLR ก็จะอนุพันธ์ 8-byte hash จาก public key ของแชร assembly และทำการเปรียบค่านี้นี้ด้วย ค่า hash ที่ทำการเก็บไว้ใน assembly manifest ของคุณถ้าตรงกัน CLR ก็จะโหลด assembly ที่ถูกต้อง สำหรับคุณ



รูปที่ 6-1 การทำ Digital Signature

Interfaces

Interface สนับสนุนคอนเซปต์เดียวกับ C++ abstract base class (ABC) ด้วย pure virtual function เท่านั้น อย่างแท้จริง ABC คือ คลาสที่ประกาศเพียงหนึ่งหรือมากกว่านั้น ของ pure virtual function คุณ ต้องกำหนด แต่ไม่ต้อง อิมพลีเมนต์ คลาสที่อนุพันธ์จากอินเทอร์เฟส ของคุณต้องทำการอิมพลีเมนต์อิน เทอร์เฟสของคุณ โดยอินเทอร์เฟสจะบรรจุ เมธอด คุณสมบัติ indexers และ อีเวนต์ ใน .NET คลาส สามารถอนุพันธ์จาก หลายอินเทอร์เฟสได้

6.2.2 Working .NET Component

ตัวอย่างการเขียน Assembly ในภาษา c++

```
#using <mscorlib.dll>
using namespace System;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

namespace Lang
{
    __gc __interface ISteering
    {
        void TurnLeft();
        void TurnRight ();
    };

    __gc class Vehicle : public ISteering
    {
        public:
        void TurnLeft( )
        {
            Console::WriteLine("Vehicle turns left.");
        }
        void TurnRight( )
        {
            Console::WriteLine("Vehicle turns right.");
        }
        virtual void ApplyBrakes( ) = 0;
    };

    public __gc class Car : public Vehicle
    {
        public:
        void ApplyBrakes( )
        {
            Console::WriteLine("Car trying to stop.");
            throw new Exception("Brake failure!");
        }
    };
} // This brace ends the Lang namespace.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเขียน Assembly ในภาษา VB.NET

```

Imports System

Namespace Lang
Interface ISteering
    Sub TurnLeft( )
    Sub TurnRight( )
End Interface

MustInherit Class Vehicle
    Implements ISteering
    Public Sub TurnLeft( ) Implements ISteering.TurnLeft
        Console.WriteLine("Vehicle turns left.")
    End Sub
    Public Sub TurnRight( ) Implements ISteering.TurnRight
        Console.WriteLine("Vehicle turn right.")
    End Sub
    Public MustOverride Sub ApplyBrakes( )
End Class

Class Car
    Inherits Vehicle
    Public Overrides Sub ApplyBrakes( )
        Console.WriteLine("Car trying to stop.")
        throw new Exception("Brake failure!")
    End Sub
End Class
End Namespace

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเขียน Client Application เรียกใช้ ซึ่งต้องทำการ add DLL reference ใน tool VS.NET

```
using System;
namespace ConsoleApplication2
{
    class Drive
    {
        [STAThread]
        static void Main(string[] args)
        {
            Lang.Car bank = new Lang.Car();
            bank.ApplyBrakes();
        }
    }
}
```

6.2.3 Deployment

จากที่เราได้เห็นตัวอย่างในหัวข้อที่ผ่านมา .NET Assembly ทุกตัว เป็น binary component ซึ่งคุณสามารถนำไป plug กับ component หรือ application อื่นๆได้ โดยที่ไม่ต้องการ source code และไม่ต้องทำ plumbing code ด้วย

ใน .NET คอมโพเนนต์ สามารถเป็นได้ 2 แบบ ทั้ง private ที่ไม่ประกาศ(unpublished) และถูกใช้เพียงไคลเอนต์บางตัวเท่านั้น

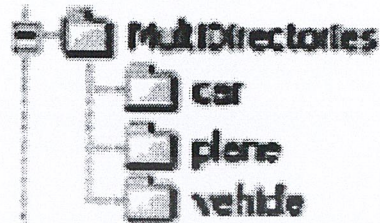
6.2.3.1 Private Components

private component ถูกใช้กับ ไคลเอนต์ที่เจาะจงเท่านั้น ซึ่งมีสองวิธีในการทำ คือ

1. One-directory deployment ใช้ คอมโพเนนต์ เหล่านี้จากไดเรกทอรีเดียวกันกับ ไคลเอนต์

ถ้าเราทำการลบหรือย้ายไฟล์คอมโพเนนต์ออกไป เมื่อเรารันโปรแกรม จะทำให้เกิด exception ออกมา

2. Multiple-directory deployment ใช้คอมโพเนนต์จากโคเร็กทอรีที่ไคลเอนต์สามารถเข้าถึงได้ การทำแบบนี้ขั้นตอนสร้างต้องทำการอ้างอิง path ของ dll ที่อ้างอิงด้วย



รูปที่ 6-2 Multiple-directory tree of component

```
csc /r:..\vehicle\vehicle.dll /t:library /out:plane.dll plane.cs
```

ในขณะที่ทำการอ้างอิงคอมโพเนนต์เหล่านี้ตอน build time CLR จะไม่ทำการเช็ค เวอร์ชัน ของ private คอมโพเนนต์

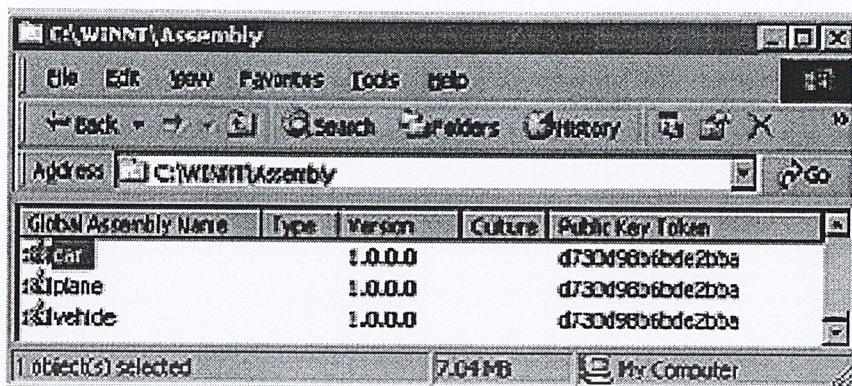
การ Install เพียงทำการ xcopy คือ ก๊อปปี้ไฟล์จากโคเร็กทอรีของต้นฉบับ ไปยัง โคเร็กทอรีที่ต้องการ

การ Remove เพียงทำการลบไฟล์(หรือโคเร็กทอรี)เหล่านั้นทิ้งไปก็พอ เราไม่ต้องกังวลเกี่ยวกับการเขียนโค้ดเพื่อทำการเก็บข้อมูลต่างๆลงไปใน registry และเรื่องอื่นๆเกี่ยวกับการ registry เลย

6.2.3.2 Shared Components

shared Components เรียกอีกอย่างว่า system component ซึ่งสามารถถูกเรียกใช้ได้จาก โปรแกรม ไคลเอนต์ใดๆก็ได้ แต่ว่า เราต้องทำการประกาศหรือทำการ ลงทะเบียน(register) ลงใน system Global Assembly Cache (GAC) เมื่อเราทำการลงทะเบียนเสร็จ มันก็เป็นเหมือน system dll ซึ่ง process ใดๆในระบบสามารถเรียกใช้ได้

ข้อสำคัญในการลงทะเบียนใน GAC คือ คอมโพเนนต์ต้องทำเกี่ยวกับ originator และ version ด้วย และด้วยสองส่วนนี้ ทำให้เราสามารถสร้าง คอมโพเนนต์เดียวกันในหลายๆเวอร์ชันและรันบนเครื่องเดียวกันได้ โดยไม่มีปัญหา



รูปที่ 6-3 แชร่คอมโพเนนต์ที่เก็บใน GAC

6.2.4 ขั้นตอนการลงทะเบียนคอมโพเนนต์

1. ใช้ share named utility (sn.exe) ในการสร้าง random public/private key และบันทึกเก็บไว้ในไฟล์ เช่น originator.key

```
sn -k originator.key
```

2. สร้าง assembly(คอมโพเนนต์) พร้อมด้วย assembly version number และ key information จาก originator.key

```
Imports System
```

```
Imports System.Reflection
```

```
<Assembly:AssemblyVersion("1.0.0.0")>
```

```
<assembly:AssemblyKeyFile("../key\originator.key")>
```

```
Public Class Car
```

```
    Inherits Vehicle
```

```
    Overrides Public Sub TurnLeft( )
```

```
        Console.WriteLine("Car turns left.")
```

```
    End Sub
```

```
    Overrides Public Sub TurnRight( )
```

```
        Console.WriteLine("Car turns right.")
```

```
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Overrides Public Sub ApplyBrakes( )
    Console.WriteLine("Car trying to stop.")
    Console.WriteLine("ORIGINAL VERSION - 1.0.0.0.")
    throw new Exception("Brake failure!")
End Sub

```

End Class

3. ใช้ .NET Global Assembly Cache Utility(gacutil.exe) ในการลงทะเบียน assembly ลงใน GAC ก็เป็นอันเสร็จเป็น shared component สามารถเรียกใช้จาก client ใดๆก็ได้

gacutil /i car.dll

การ uninstall shared component เหล่านี้ ก็เพียงแค่ทำการ ลบไฟล์ assembly ที่เหมาะสม ก็เสร็จ มันก็จะออกจาก GAC

6.2.5 Building and testing the drive.exe

เราสร้างแอปพลิเคชัน drive.exe เราต้องมีการ include คอมโพเนนต์ที่ต้องการเรียกใช้เข้าไปด้วย เราเรียกตอนนี้ว่า build time ซึ่งเมื่อสร้างแล้ว ได้ผลลัพธ์ไฟล์ drive.exe

ORIGINAL VERSION - 1.0.0.0.

Adding new versions

ในการใช้ shared assembly นั้นจะมีการใช้ประโยชน์เกี่ยวกับเรื่อง version policies ในขณะที่ โหลด shared assemblies ทั้งหมด ซึ่ง CLR จะทำการโหลด assembly version ที่เราใช้ในขณะ build time แต่เราก็สามารถที่จะสั่งให้ CLR โหลด เวอร์ชันที่ต้องการได้ โดยสร้าง application configuration ขึ้นมา เพื่อบอกให้เรียกใช้ เวอร์ชันที่เราต้องการ

แก้ไข code ใหม่เป็น

```
<Assembly:AssemblyVersion("1.0.1.0")>
```

```
Overrides Public Sub ApplyBrakes( )
```

```
    Console.WriteLine("Car trying to stop.")
```

```
    Console.WriteLine("BUILD NUMBER change - 1.0.1.0.")
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราคอมไพล์ออกมาก็จะได้ไฟล์ที่อาจจะมีชื่อซ้ำกันก็ได้ แต่ยังสามารถใช้ได้เนื่องจากเวอร์ชันไม่เหมือนกัน ถ้าเราทำการรันโปรแกรม drive.exe ก็จะทำได้ผลลัพธ์ คือ

ORIGINAL VERSION - 1.0.0.0.

ถ้าต้องการให้โหลด assembly เวอร์ชันใหม่ เราต้องทำการสร้าง config ให้มัน ในชื่อของ drive.exe.config ไว้ในไดเรกทอรีเดียวกันกับ drive.exe

```
<?xml version="1.0"?>
<configuration>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="car"
          publicKeyToken="D730D98B6BDE2BBA"
          culture="" />
        <bindingRedirect oldVersion="1.0.0.0"
          newVersion="1.0.1.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

เมื่อเราทำการรัน drive.exe อีกครั้ง ผลลัพธ์ที่ได้จะเปลี่ยนไป จะไปเรียกใช้ dll เวอร์ชัน 1.0.1.0 แทน BUILD NUMBER change - 1.0.1.0.

มีส่วนที่ควรรู้คู่สองส่วนในเรื่องของ config คือ 1.ชื่อของ assemblyIdentity tag จะเป็น ชื่อของ assembly ที่ใช้สำหรับคนอ่าน (human-readable) ซึ่งเก็บใน GAC และ 2.publicKeyToken attribute จะเก็บค่าของ public key ซึ่งเป็นเลข 8 byte hash ของ public key ในการสร้าง(build) คอมไพล์

ในการดูค่า 8-byte hash นี้ สามารถดูได้จากหลายวิธี เช่นจาก Shell Cache Viewer , IL dump of component หรือใช้ sn.exe
sn -T car.dll

6.3 Enterprise Services (COM+) in .NET

.NET สามารถใช้งานทุกบริการของ COM+ ที่มีให้ใช้งานได้ เช่น

- 1.Transaction
- 2.Just-In-Time (JIT)
- 3.Object Pooling
- 4.Security

6.3.1 Transaction

ในการเขียน managed แอปพลิเคชันนั้นในการใช้บริการ คลาสต่างๆต้องทำการ derive จาก ServicedComponent ระบุแอททริบิวต์ ต่างๆเพื่อระบุ บริการที่ต้องการจริงๆ

```
BankComponentClient
using system;
using BankComponent'
namespace BankComponentClient
{
    class Client
    {
        public static int Main()
        {
            Account act = new Account ();
            act.Post (5,100);
            act.Dispose ();
            return 0;
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BankComponent Server

```

using System.EnterpriseServices

[assembly: ApplicationName ("BankComponent")]
[assembly: AssemblyKeyFileAttribute("Demos.snk")]

namespace BankComponentServer
{
    [Transaction(TransactionOption.Required)]
    public class Account : ServicedComponent
    {
        [AutoComplete]
        public bool Post (int accountNum, double amount)
        {
            // Updates the database, no need to call SetComplete.
            //Calls SetComplete automatically if no exception is thrown.
        }
    }
}

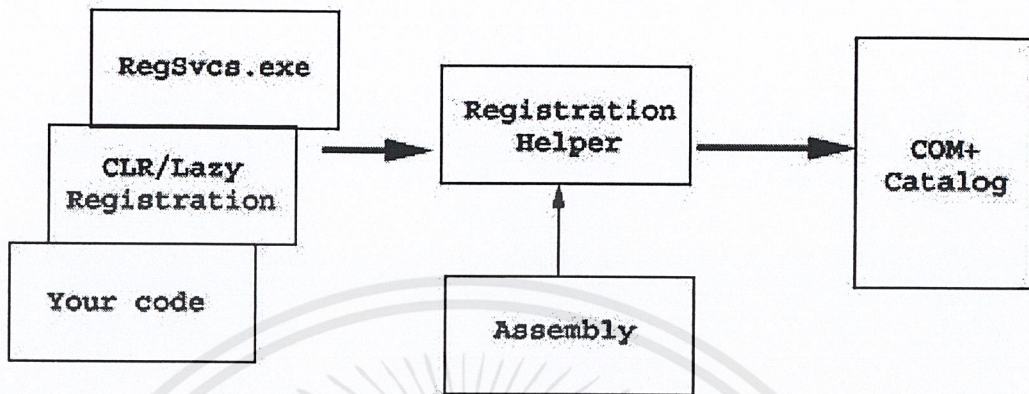
```

BankComponent Server code แสดงการใช้งานของ Account คลาสใน .NET ซึ่งใช้ การ ทรานแซกชัน โดยคลาส Account จะ derive จากคลาส System.Enterprise.ServicedComponent ค่าแอททริบิวต์ของการทรานแซกชัน ก็จะถูกกำหนด ให้เป็นต้องการ การซิงโครไนซ์ และ JIT ก็จะถูกกำหนดให้ใช้งานโดยอัตโนมัติเพราะได้กำหนดค่าแอททริบิวต์ของทรานแซกชันแล้ว

จากโค้ดจะเห็นว่าใน BankComponent Server namespace แสดงให้เห็นว่าเป็นการง่ายในการที่จะใช้ บริการของ COM+ ใน .NET

6.3.2 Lazy Registration

เมื่อไหร่ก็ตามไคลเอนต์แอปพลิเคชันทำการสร้างอินสแตนซ์ของคลาสที่ derive มาจาก ServicedComponent runtime จะทำการค้นหาว่าได้ทำการริจิสเตอร์ assembly เรียบร้อยหรือยังถ้าไม่พบ เซอร์วิสคอมโพเนนต์ทั้งหมดใน assembly ก็จะถูกริจิสเตอร์ลงใน COM+ แอปพลิเคชันซึ่งเรียกว่า lazy registration ซึ่ง lazy registration มีประโยชน์ในช่วงระหว่างการพัฒนา



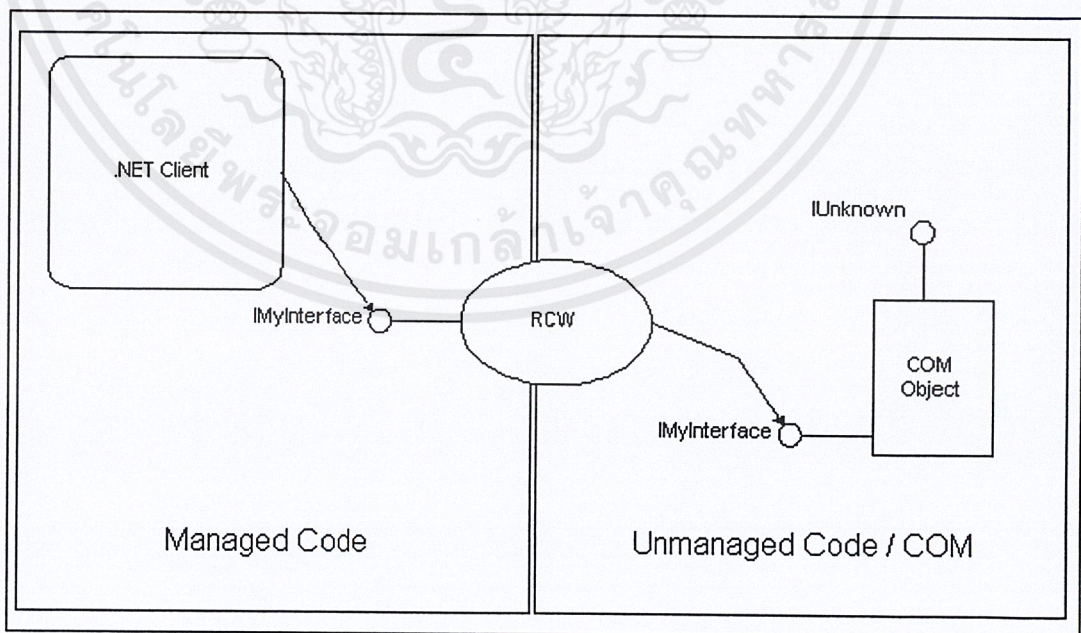
รูปที่ 6-4 แสดงการทำงานของ Lazy Registration

6.4 .NET Interoperability

คือความสามารถที่ทำให้ .NET สามารถใช้งานร่วมกับ COM คอมโพเนนต์ แบบเดิมได้ดังนี้

6.4.1 การเรียก COM คอมโพเนนต์จาก .NET

เมื่อ COM ออบเจ็กต์ถูกเรียกจาก .NET รันไทม์ก็จะทำการสร้าง Runtime Callable Wrapper (RCW) โดยที่ RCW ก็จะทำหน้าที่เหมือนพรีอ็อกซีสำหรับ unmanage ออบเจ็กต์



รูปที่ 6-5 แสดงการเรียก COM คอมโพเนนต์จาก .NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RCW จะรับผิดชอบหน้าที่ในการติดต่อระหว่าง .NET ไคลเอนต์ โค้ด และ COM คอมโพเนนต์รวมทั้ง

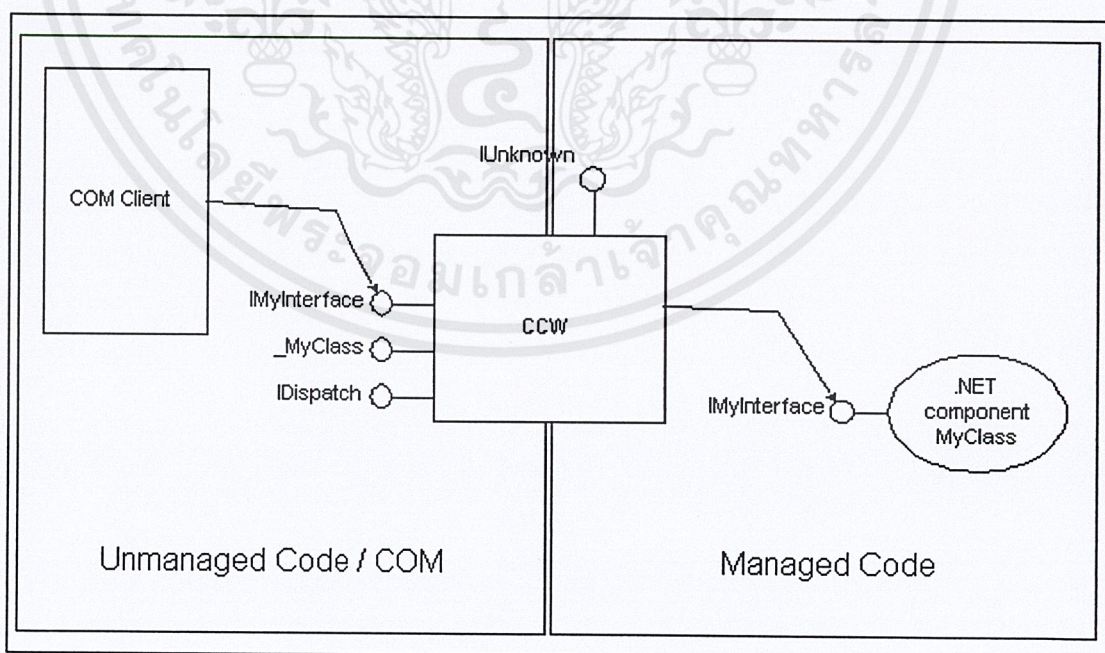
- สร้างและการ binding ที่มี COM คอมโพเนนต์ เป็นรากฐาน
- การใช้ อินเตอร์เฟซ และ องค์กรประกอบขอ Synthesizing อินเตอร์เฟซไปเป็น managed form
- ทำการแปลและทำการมาแชล ระหว่างสภาพแวดล้อม
- บริหาร ไลฟไทม์ ของ wrapped COM object
- ทำการแปลค่า COM HRESULT ไปเป็น .NET exception

RCW เป็น manage ออบเจกต์และอยู่ใน heap ซึ่งดูแลโดย CLR เพราะกับ manage ออบเจกต์อื่นๆก็จะอ้างไปที่ RCW ซึ่งจะถูกตามรอยโดย รันไทม์ และ RCW ก็คือหัวข้อ ไปสู่ garbage collection

คือความสามารถที่ทำให้ .NET สามารถใช้งานร่วมกัน COM คอมโพเนนต์ แบบเดิมได้ดังนี้

6.4.2 การเรียก .NET คอมโพเนนต์จาก COM

เมื่อ .NET คอมโพเนนต์ถูกเรียกจาก COM รันไทม์ก็จะสร้าง wrapper ออบเจกต์เพื่อเชื่อมต่อระหว่างสภาพแวดล้อมซึ่งในกรณีนี้รันไทม์จะสร้าง COM callable wrapper (CCW) รันไทม์ก็จะอ่านรายละเอียด type สำหรับคอมโพเนนต์จาก assembly metadata และสร้าง CCW ที่เข้ากันได้ คล้ายกับ RCW แต่ CCWจะทำตัวเป็นพรีอ็อกซี unmanaged COM code กับ managed .NET code



รูปที่ 6-6 แสดงการเรียก .NET คอมโพเนนต์จาก COM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CCW รับผิดชอบทั้งหมดระหว่าง COM ไคลเอนต์กับ managed อ็อบเจ็กต์รวมทั้ง

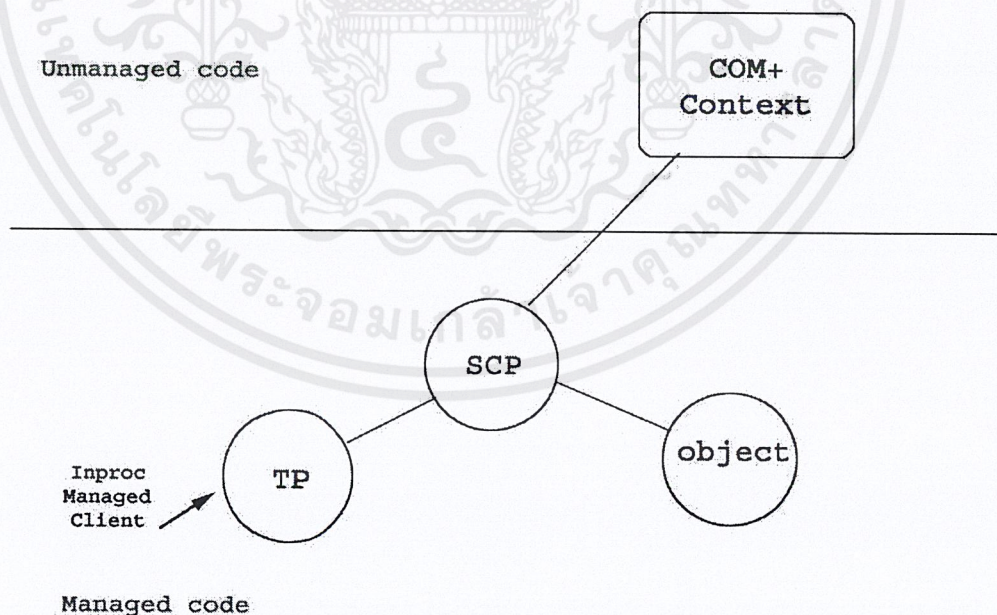
- สร้างและมีการ binding ที่มี managed อ็อบเจ็กต์เป็นรากฐาน
- ทำการสังเคราะห์อินเทอร์เฟซที่สำคัญหลายตัวของ COM (เช่น IUnknown และ Idispach) ขึ้นกับข้อมูล type ของอ็อบเจ็กต์

- ทำการมาแชนและทำการแปลระหว่างสภาพแวดล้อม
- บริหารไลฟ์ไทม์ของ .NET คอมโพเนนต์
- แปล .NET exceptions ไปเป็นค่า COM HRESULT

CCW ก็คือ unmanaged COM อ็อบเจ็กต์ซึ่งอยู่ที่ heap และ เป็น reference-counted COM อ็อบเจ็กต์เดิม CCW ไม่ใช่ garbage collected แต่เป็นการทำลาย โดยปล่อย ไคลเอนต์ที่ทำการอ้างถึงล่าสุด เมื่อ CCW ถูกทำลายแล้วก็จะทำการ wrap ก็จะถูก mark สำหรับ garbage collection

- Activation

Enterprise Services สร้างจากคอนเซปต์ ของ คอนเท็กซ์ ซึ่ง คอนเท็กซ์คือสภาพแวดล้อมสำหรับ อ็อบเจ็กต์ที่มีความต้องการสำหรับการเอ็กซีคิวทีฟที่คล้ายคลึงกัน เซอร์วิสสามารถถูกบังคับให้ปฏิบัติตาม ระหว่าง activation และ/หรือ เมธอด เรียกว่า interception ถึงแม้ว่าบริการของ COM+ จะถูกเขียนเป็น unmanaged code การผนวก COM+ services กับ .NET ลึกเกินกว่าแค่การใช้ COM interop technology in .NET ที่ปราศจากการ derive จาก ServicedComponent การ registration โปรเซสซึ่งจะไม่เกิดผลตามที่ปรารถนา



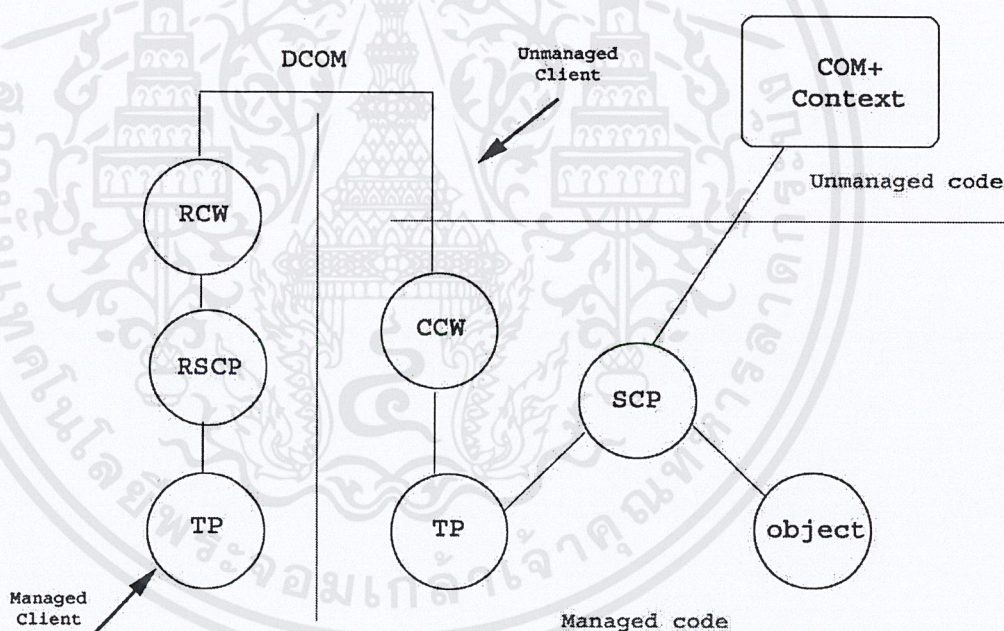
รูปที่ 6-7 in-process activation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

activation มี 3 กรณีดังต่อไปนี้แบบแรก in-process activation คือมีการข้าม context boundary แบบที่สอง cross-domain เป็นการข้ามทั้ง context และ app domain แบบที่สาม cross-process ข้ามเครื่อง/ข้ามprocess และ ข้าม context boundaries

ในกรณี in-process activation จากรูปแสดงให้เห็นว่า reference เป็นแบบunmarshaled ซึ่งเป็น direct reference ไปที่ transparent proxy (TP) กรณี cross app-domain reference เป็นแบบ unmarshaled ซึ่งคือ .NET remotine proxy

ในกรณี Cross process หรือ Cross machine reference ต้องการมากกว่าการ unmarshaled โดย COM interop ทำการเรียก IManagedObject ถูกอิมพลิเมนต์โดย ServicedComponent ระหว่าง activation และการมาแชล remote serviced component proxy (RSCP) ทำการเรียก IServicedComponentInfo ระหว่าง activation



รูปที่ 6-8 cross-process activation

- Contexts

คลาส ContextUtil ใช้ในการเข้าถึง COM+ object context ที่เกี่ยวข้อง และ คุณสมบัติต่างๆของมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 .NET Component VS COM

| | .NET Component | COM |
|--------------|---|---|
| Architecture | Object oriented to the core of language | Just the wrapper of the binary component |
| Deployment | <ul style="list-style-type: none"> - Just copy the file - Varied versions of same component can exist in the system | <ul style="list-style-type: none"> - Require registration - Incompatibility problem |
| Language | All language are 1 st class | Only c++ can be low -level COM programming |
| Code | Eliminate plumbing code | Plumbing code to support IUnknow, class factory , component lifetime |

ตาราง 6-1 ตารางเปรียบเทียบ ระหว่าง .NET Component กับ COM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

แอปพลิเคชัน

7.1 Overview ของระบบ OLALA Wedding Planner

ระบบ OLALA Wedding Planner เป็นระบบจัดการการแต่งงาน ที่เป็นเว็บเซอร์วิสให้บริการจัดหาโรงแรม , ของที่ระลึก , ถ่ายรูปแต่งงาน , การจองแพ็คเกจทัวร์ , การจองวงดนตรีหรือการแสดง ที่เป็นส่วนหนึ่งในงานแต่งงาน โดยผู้ใช้บริการสามารถค้นหาข้อมูลต่างๆได้ตามต้องการ โดยกรอกรายละเอียดที่จำเป็นในการค้นหาไป เช่น ต้องการจองห้องจัดเลี้ยง ข้อมูลที่ต้องใช้ในการค้นหาอย่างน้อยคือ สไตล์การตกแต่ง สถานที่ห้องจัดเลี้ยงแบบไหน , การจองการถ่ายรูปแต่งงาน ข้อมูลที่ต้องใส่อย่างน้อยคือ ประเภทของการถ่ายรูป (ในสถานที่ , นอกสถานที่) และช่วงราคา เป็นต้น จากนั้นทางระบบจะเรียกใช้เซอร์วิสต่างๆ ที่เว็บเซอร์วิสมีให้เรียกใช้

เมื่อผู้ใช้บริการทำการจองเรียบร้อยแล้ว จะสามารถเข้าตรวจสอบดูรายละเอียด , เปลี่ยนแปลง และยกเลิกการจองครั้งนั้นๆได้ ส่วนเรื่องการชำระค่าบริการจะเป็นการชำระผ่านบัตรเครดิต เมื่อชำระค่าบริการเรียบร้อยแล้ว หากผู้ใช้บริการต้องการยกเลิกรายการที่จองไว้ ผู้ใช้บริการจะไม่ได้ค่าบริการคืน ซึ่งเป็นกฎของทางระบบ OLALA Wedding Planner

ในปฏิญญาพันธบัตรฉบับนี้จะแบ่งการออกแบบบริการต่างๆเป็น 2 เทคโนโลยี คือ เทคโนโลยีของ .NET เฟรมเวิร์ค และ เทคโนโลยีของจาวา ดังรูปที่ 11-1 เพื่อทำการศึกษาการเรียกใช้คอมโพเนนต์ข้ามสถาปัตยกรรม โดยทำการติดต่อสื่อสารกันผ่านโปรโตคอล SOAP เทคโนโลยีของจาวาจะประกอบด้วย

ธนาคาร

ระบบทัวร์

ระบบเอ็นเตอร์เทนเมนต์

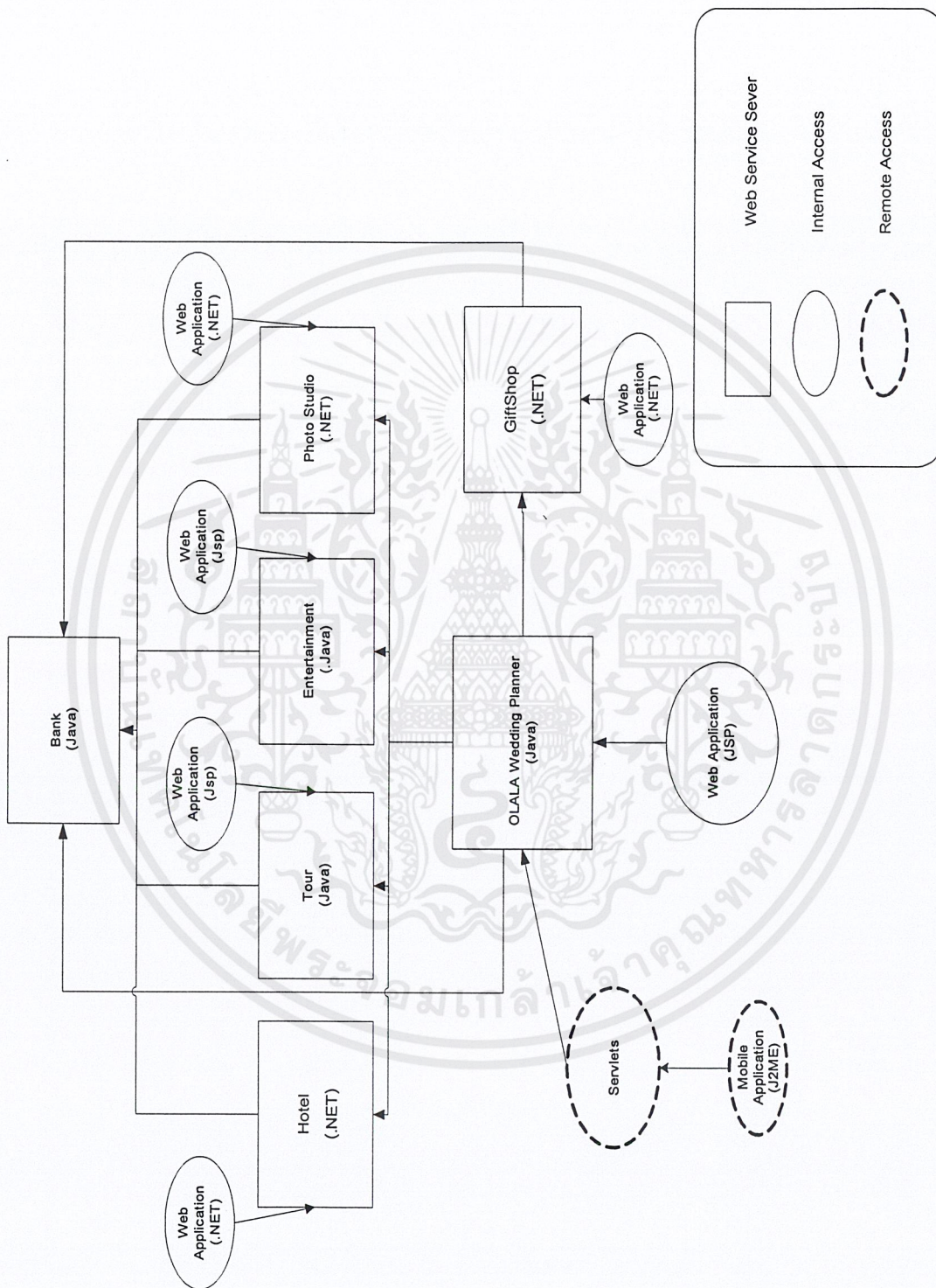
ระบบ OLALA Wedding Planner

เทคโนโลยีของ .NET เฟรมเวิร์คประกอบด้วย

ระบบโรงแรม

ระบบสตูดิโอถ่ายภาพ

ระบบขายของชำร่วย



รูปที่ 7-1 ระบบ OLALA Wedding Planner

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 รายละเอียดของแอปพลิเคชัน

7.2.1 ระบบสตูดิโอถ่ายภาพ

ระบบสตูดิโอถ่ายภาพ ซึ่งสามารถค้นหาบริการตามเงื่อนไขที่ต้องการ ทำการจอง ชำระเงิน ดูรายละเอียดการจอง เปลี่ยนแปลงข้อมูลหรือยกเลิกการจองได้ สำหรับบริการที่มี แบ่งเป็น 6ประเภทหลัก คือ

- 1-บริการถ่ายภาพถ่ายภาพวิวาห์ ในสตูดิโอ
- 2-บริการถ่ายภาพถ่ายภาพครอบครัว,ภาพแฟชั่น ในสตูดิโอ
- 3-บริการถ่ายภาพในวันแต่งงาน (นอกสตูดิโอ)
- 4-บริการถ่ายภาพวันรับปริญญา,ในวันพิธี หรือ งานสังสรรค์ต่างๆ (นอกสตูดิโอ)
- 5-บริการถ่ายวีดีโอวันงานพิธี หรืองานเลี้ยงสังสรรค์ (นอกสตูดิโอ)
- 6-บริการถ่ายวีดีโอจัดทำวีดีโอ มัลติมีเดีย พิธีเซ่นเซ่น ในงานแต่งงาน

แต่ละประเภทจะแบ่งออกเป็นหลายแพ็คเกจ ผู้ใช้สามารถค้นหาแพ็คเกจที่ต้องการได้โดยกำหนดเงื่อนไขทางด้าน วันที่ต้องการจองเวลาบันทึกภาพ ราคาสูงสุด ต่ำสุดของแต่ละแพ็คเกจ โดยที่ผู้ใช้ไม่จำเป็นต้องเป็นสมาชิกหรือล็อกอินเข้าระบบก่อน

สำหรับบริการอื่นๆ ได้แก่บริการจอง แก้วไขหรือยกเลิกการจอง ชำระเงิน ผู้ใช้จะต้องทำการสมัครสมาชิก และล็อกอินเข้าระบบก่อน หลังจากนั้น จะต้องทำการเลือกหรือสร้างชื่อของงานในการติดต่อที่ต้องการจองบริการ เพื่อความสะดวกในกรณีที่ต้องการบริการหลายๆแพ็คเกจ สำหรับคนละงานกัน ซึ่งชื่อของงานในการติดต่อนี้ เราจะเรียกว่า คอนแท็ก เมื่อลูกจะทำการจอง จะต้องเลือกว่าการจองบริการรายการนี้ เป็นของงานใด แล้วจึงทำการจองได้ เมื่อจองแล้ว ระบบจะแจ้งวันสุดท้ายที่จะทำการชำระเงินเพื่อยืนยันการจองได้ ลูกค้าจะต้องทำการชำระค่าบริการรวมทั้งคอนแท็กผ่านบัตรเครดิต ก่อนวันที่ระบบแจ้งมิฉะนั้น รายการจองทั้งหมดจะถูกยกเลิกโดยอัตโนมัติ เมื่อชำระเงินแล้ว ไม่สามารถเปลี่ยนแปลงข้อมูลการจองได้ แต่ยังสามารถทำการจองเพิ่มในคอนแท็กเดียวกันได้ จนกว่าจะถึงวันสุดท้ายที่สามารถชำระเงินเพื่อยืนยันการจอง หลังจากเลยวันดังกล่าวไปแล้ว ไม่สามารถทำรายการจองได้เพิ่มเติมในคอนแท็กได้อีก

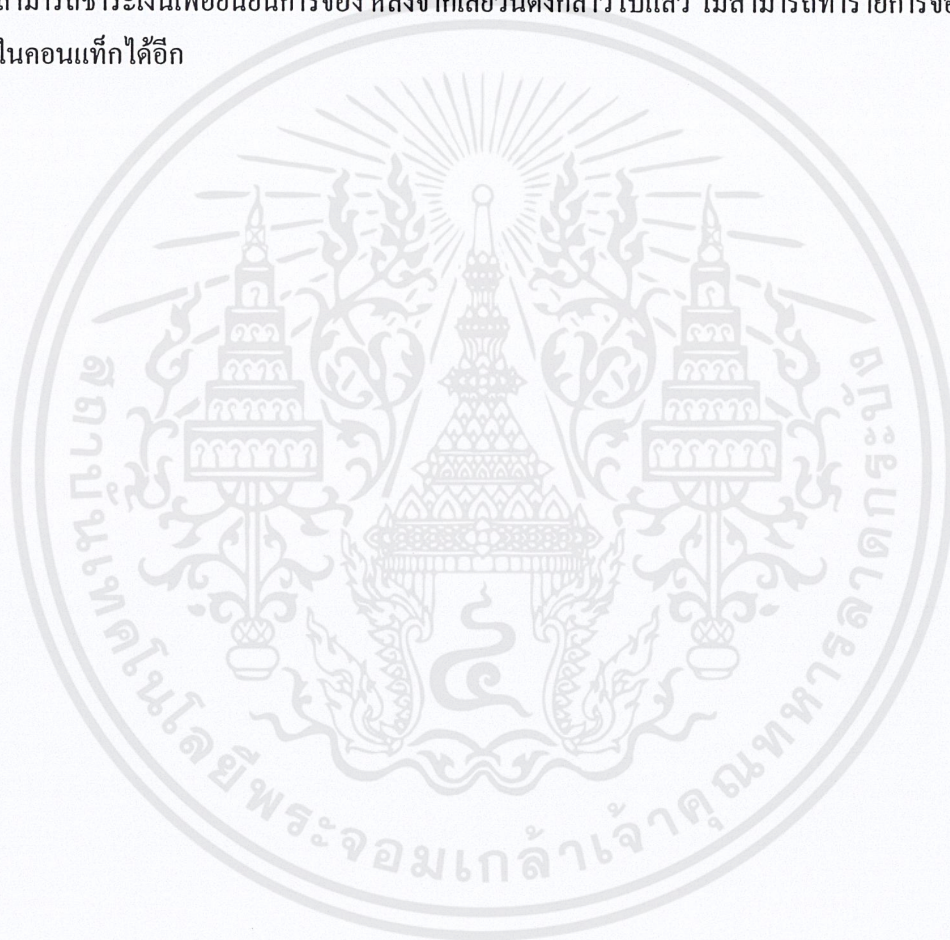
7.2.2 ระบบร้านของชำร่วย

ระบบร้านของชำร่วย ให้บริการส่งของสินค้าที่จำเป็นในงานพิธีวิวาห์ โดยมีสินค้าหลายประเภทให้เลือก นอกจากนั้น ยังสามารถพิมพ์ข้อความลงบนสินค้า โดยเลือกข้อความ รูปแบบตัวอักษรได้ตามความพอใจของลูกค้า

สำหรับบริการที่มี ได้แก่การเลือกสินค้าตามความต้องการของลูกค้า โดยสามารถกำหนดเงื่อนไขทางราคา ประเภทสินค้า และรูปแบบของสินค้าว่าต้องการสินค้าในรูปแบบ ไทย จีน หรือสากล และยัง สามารถทำการจอง โดยระบุจำนวน วันที่ส่งสินค้า สถานที่ส่งสินค้า และข้อความที่จะพิมพ์ลงบนสินค้า สามารถดูข้อมูลการจอง ยกเลิกการจองและแก้ไขข้อมูลการจองได้ สำหรับบริการส่งของสินค้า ลูกค้าน่าจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

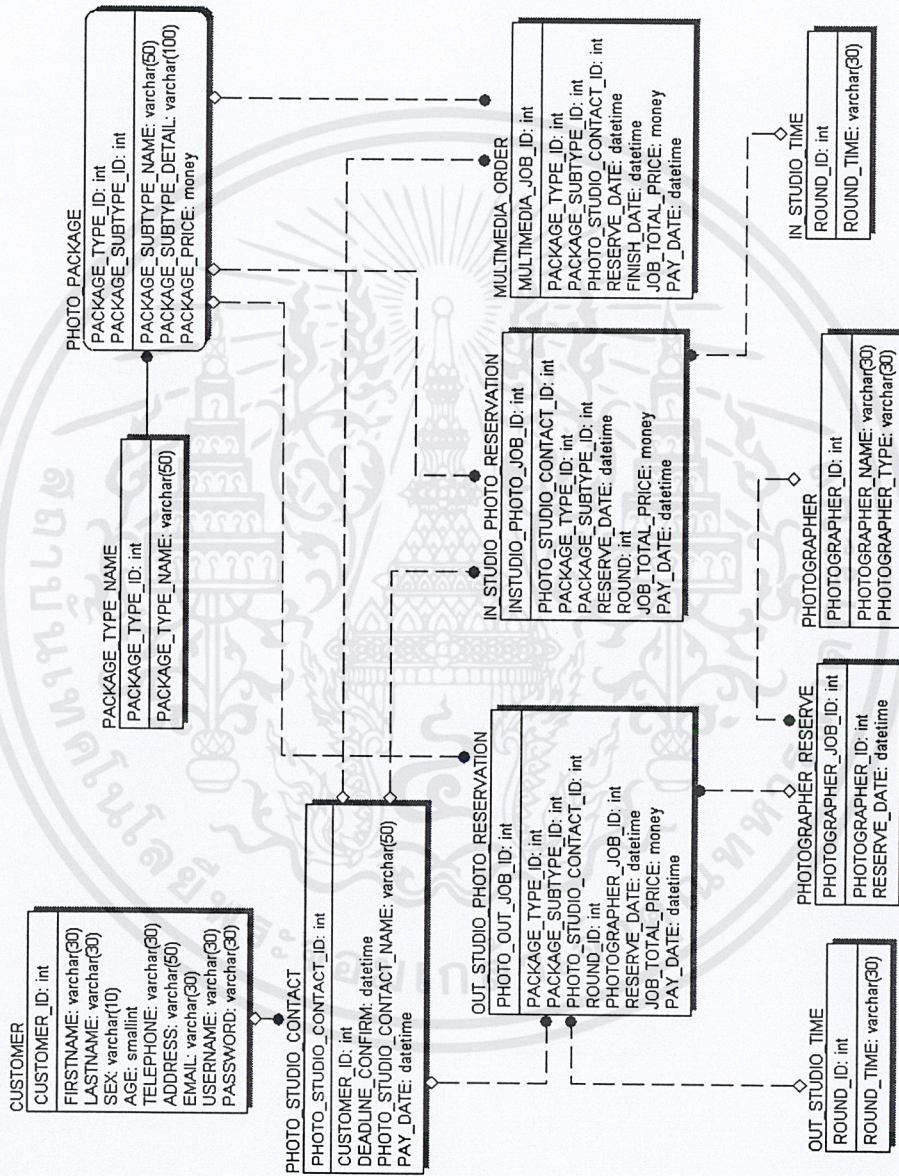
ต้องสมัครเป็นสมาชิก และทำการล็อกอินเข้าระบบ หลังจากนั้น จะต้องทำการเลือกหรือสร้างชื่อของงานในการติดต่อที่ต้องการส่งออกไป เพื่อความสะดวกในกรณีที่ตั้งสินค้าหลายๆชนิด สำหรับคนละงานกัน ซึ่งชื่อของงานในการติดต่อนี้ เราจะเรียกว่า คอนแท็ก เมื่อลูกจะทำการจอง จะต้องเลือกว่าการจองบริการรายการนี้ เป็นของงานใด แล้วจึงทำการจองได้ เมื่อจองแล้ว ระบบจะแจ้งวันสุดท้ายที่จะทำการชำระเงินเพื่อยืนยันการจองได้ ลูกค้าจะต้องทำการชำระค่าบริการรวมทั้งคอนแท็กผ่านบัตรเครดิต ก่อนวันที่ระบบแจ้ง มิฉะนั้น รายการจองทั้งหมดจะถูกยกเลิกโดยอัตโนมัติ เมื่อชำระเงินแล้ว ไม่สามารถเปลี่ยนแปลงข้อมูลการจองได้ แต่ยังสามารถทำการจองเพิ่มในคอนแท็กเดียวกันได้ จนกว่าจะถึงวันสุดท้ายที่สามารถชำระเงินเพื่อยืนยันการจอง หลังจากเลขวันดังกล่าวไปแล้ว ไม่สามารถทำรายการจองได้เพิ่มเติมในคอนแท็กได้อีก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 ER diagram

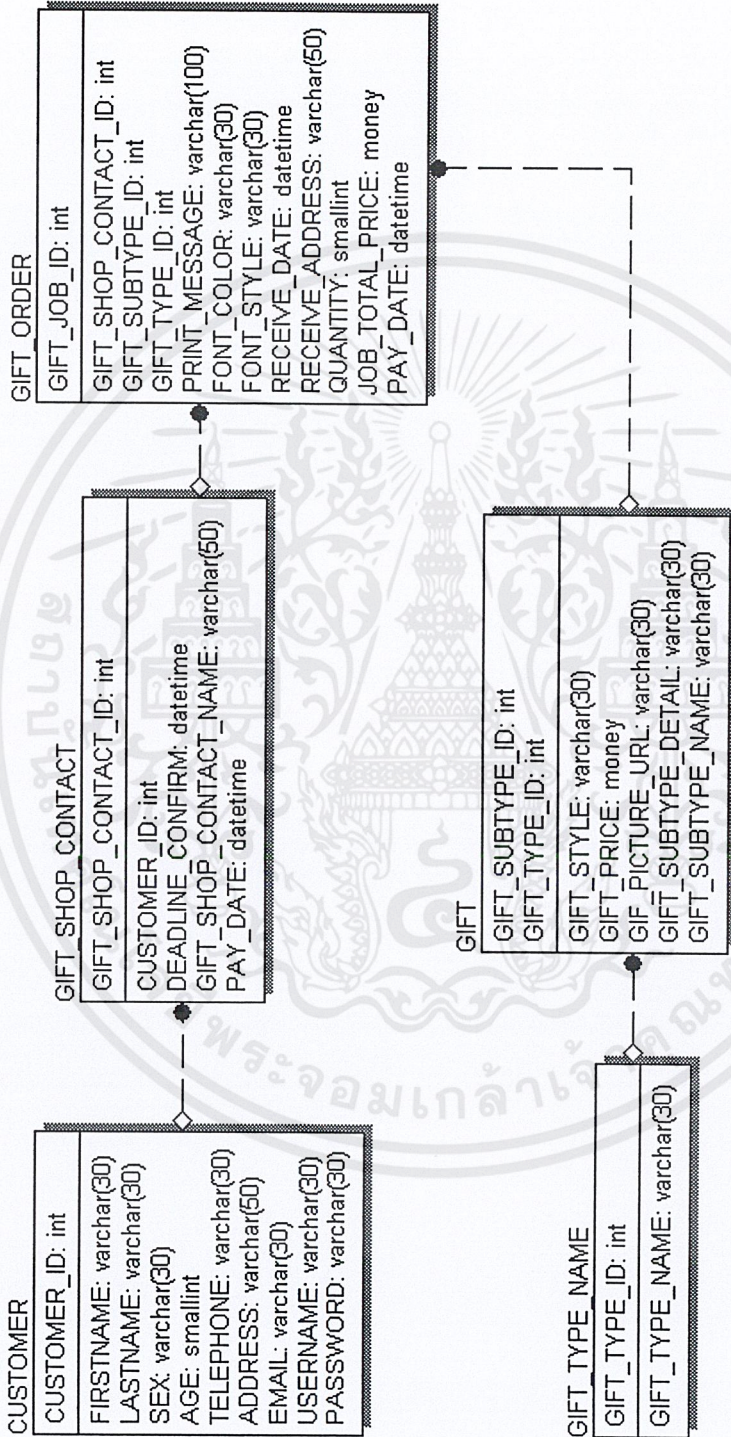
7.3.1 ER diagram ของระบบสตูดิโอถ่ายภาพ



รูปที่ 7-2 ER Diagram ระบบ สตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.2 ER diagram ของระบบร้านของขวัญ

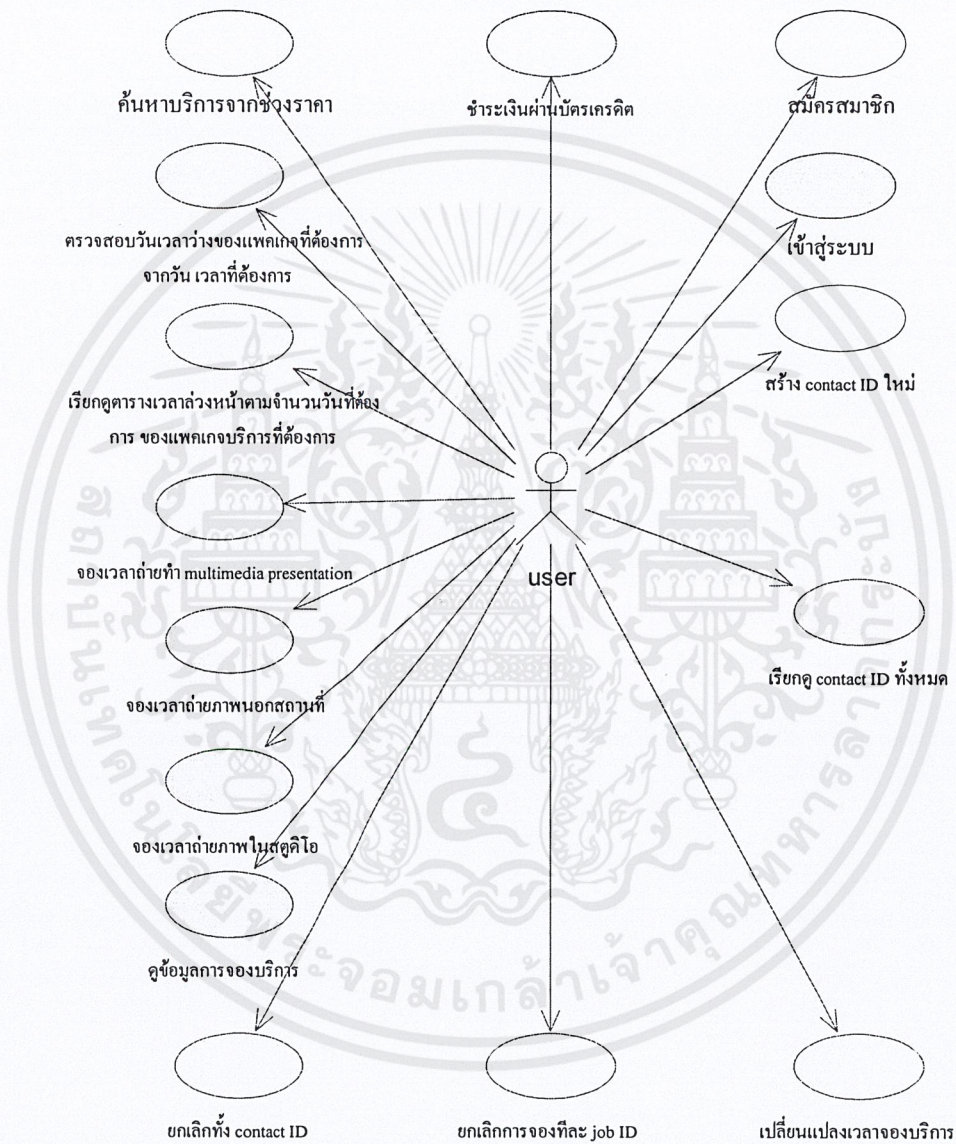


รูปที่ 7-3 ER Diagram ระบบ ร้านขายของขวัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 Use case diagram

7.4.1 Use case diagram ระบบสตูดิโอถ่ายภาพ



รูปที่ 7-4 Use Case Diagram ระบบร้านของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4.2 Use case diagram ระบบร้านของชำราย

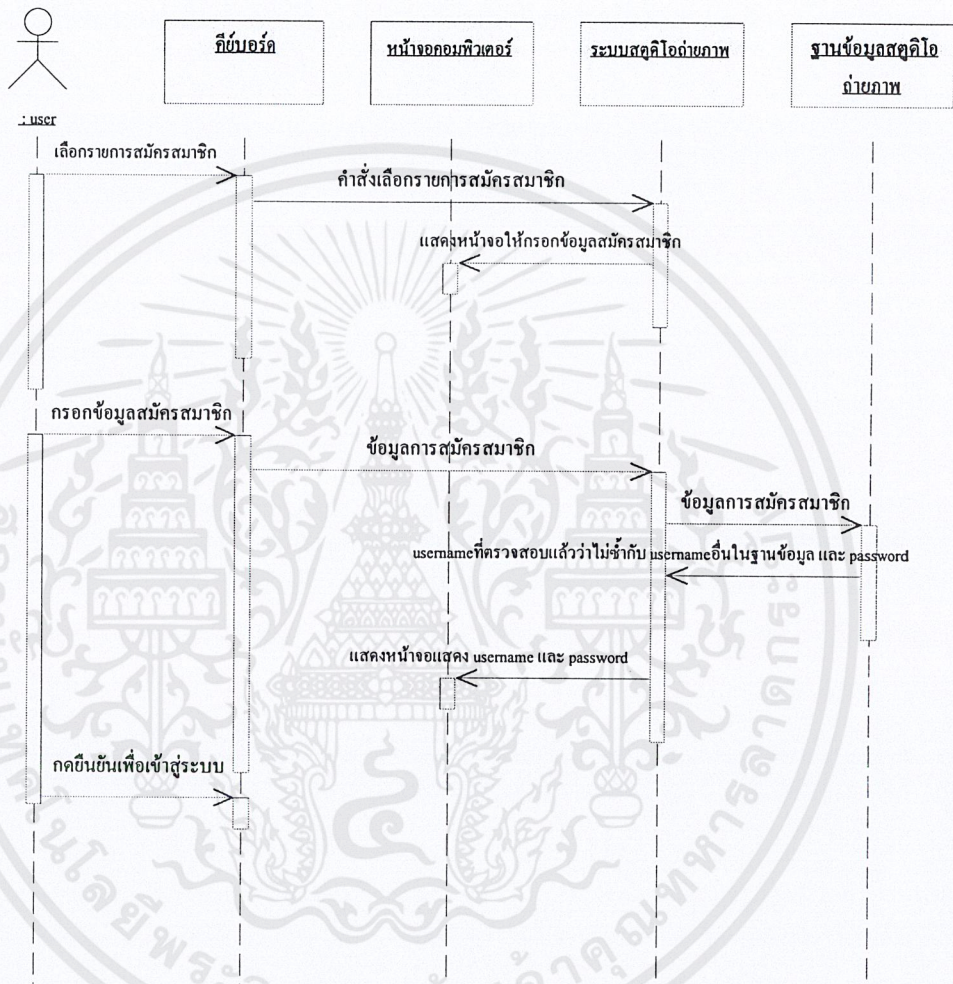


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 Sequence diagram

7.5.1 Sequence diagram ระบบสตูดิโอถ่ายภาพ

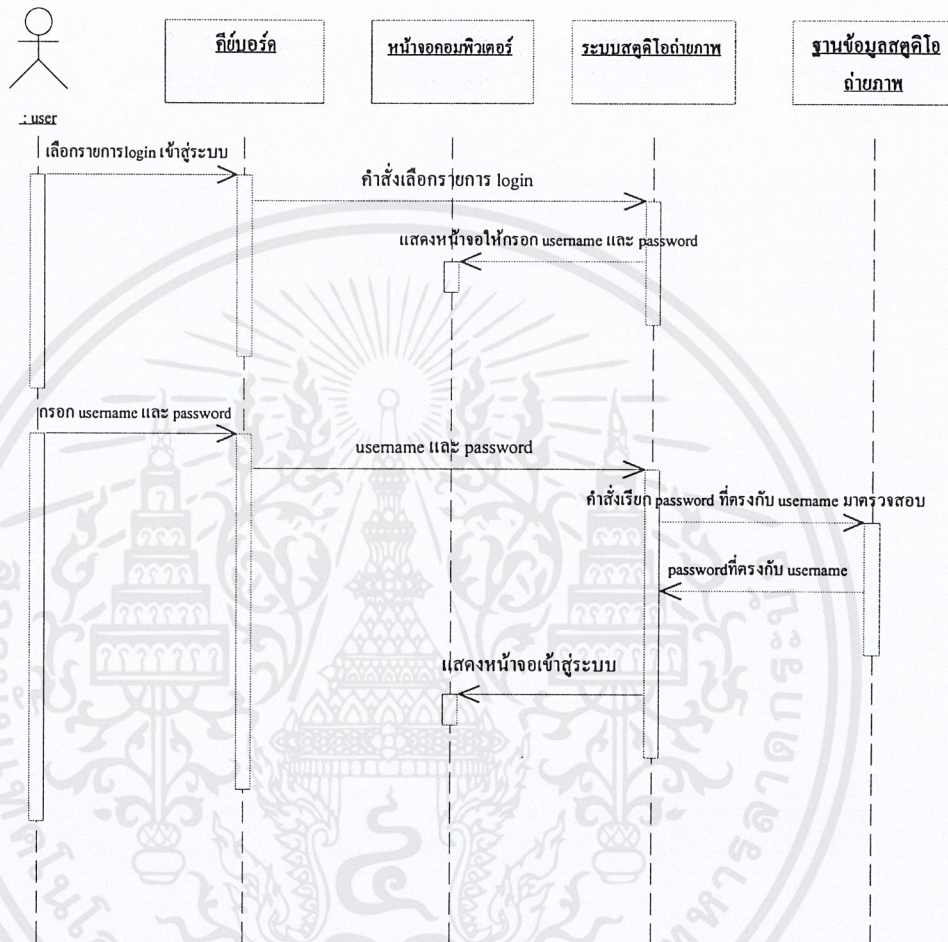
7.5.1.1 สมัครสมาชิก



รูปที่ 7-6 Sequence Diagram การสมัครสมาชิก ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

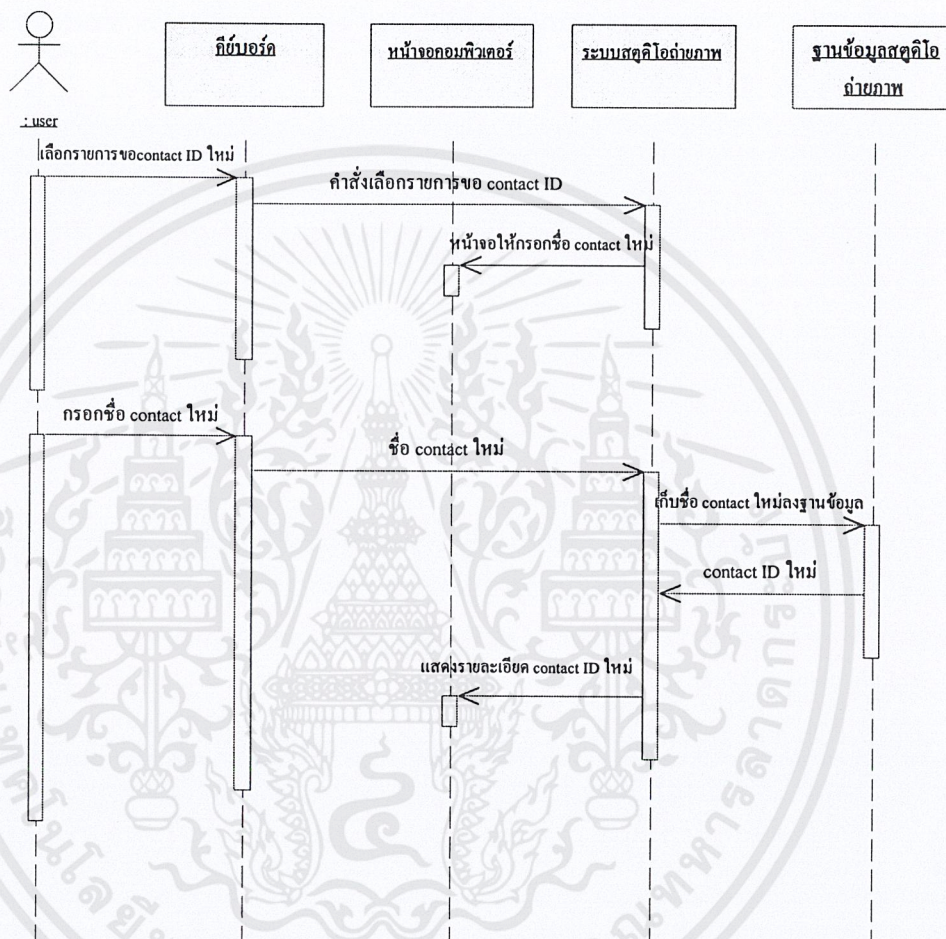
7.5.1.2 เข้าสู่ระบบ



รูปที่ 7-7 Sequence Diagram การเข้าสู่ระบบ ของระบบสติ๊กเกอร์ถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

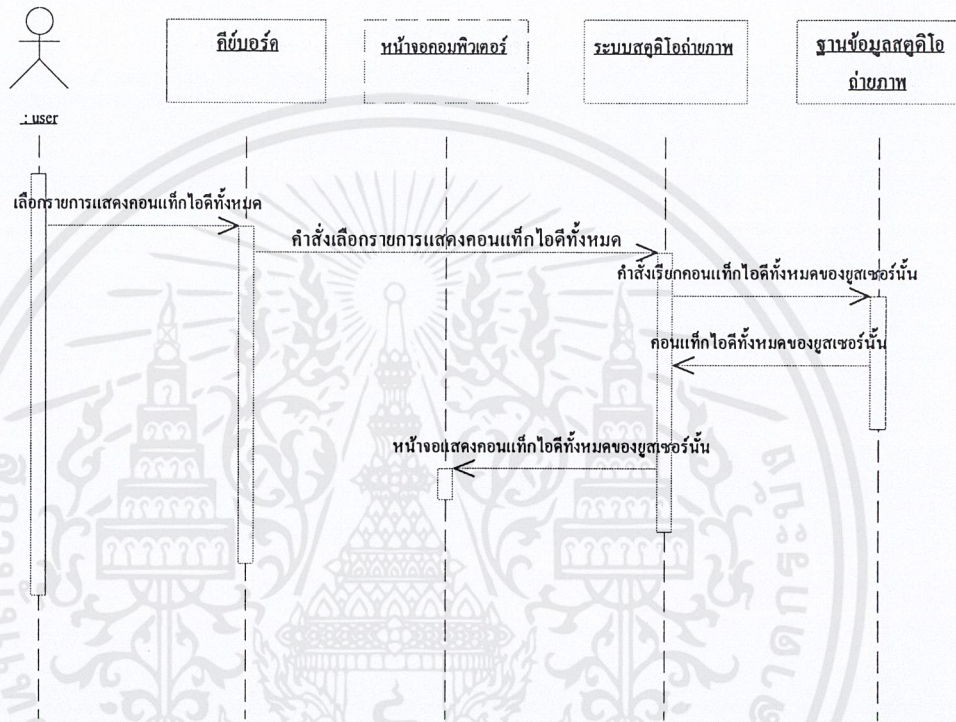
7.5.1.3 สร้าง contact ใหม่



รูปที่ 7-8 Sequence Diagram การสร้าง contact ใหม่ ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

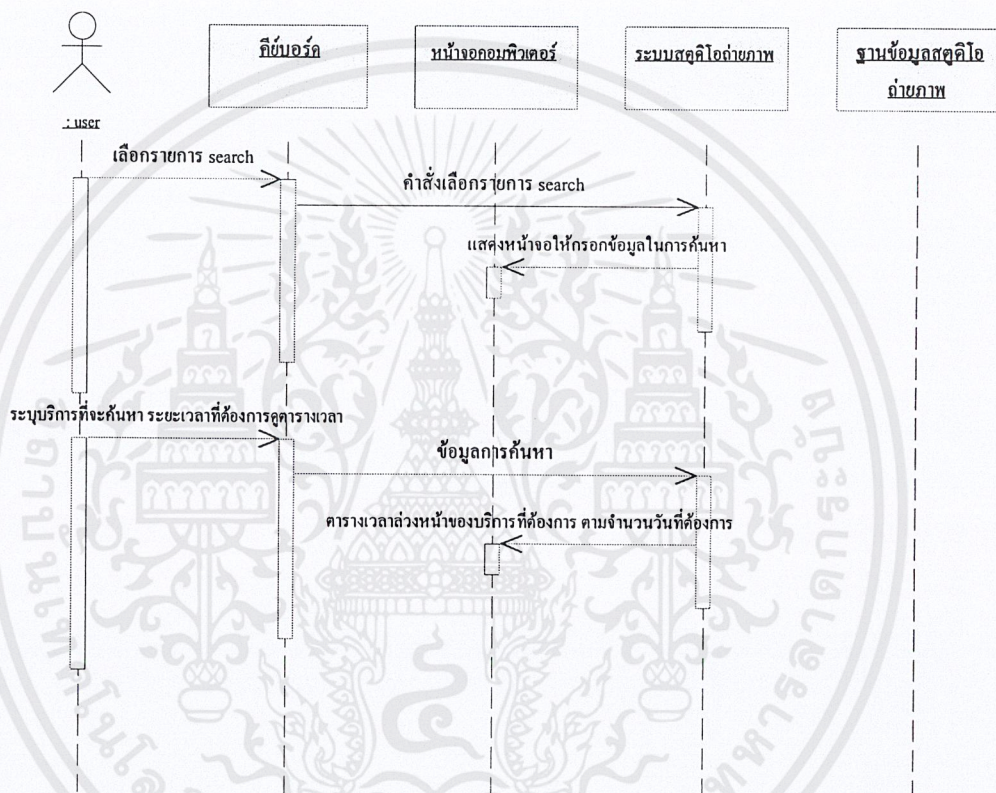
7.5.1.4 เรียกดู contact ทั้งหมด



รูปที่ 7-9 Sequence Diagram การเรียกดู contact ทั้งหมด ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

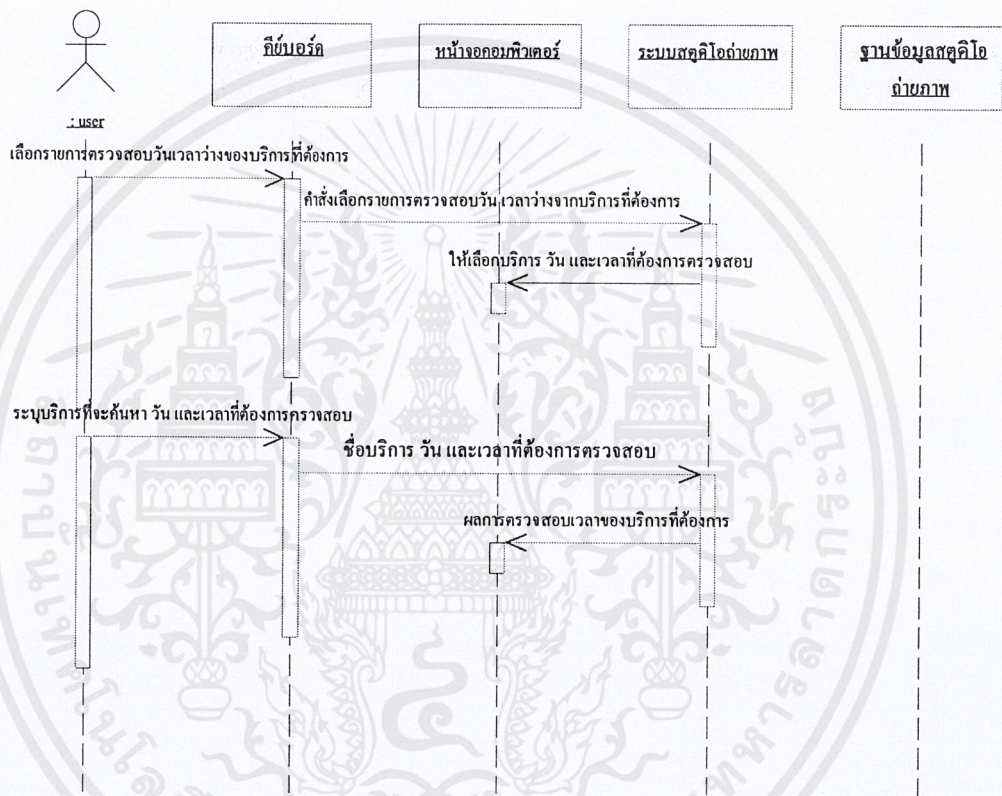
7.5.1.5 ค้นหาบริการจากช่วงราคา



รูปที่ 7-10 Sequence Diagram การค้นหาบริการของระบบสตูดิโอถ่ายภาพ จากช่วงเวลาที่ต้องการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

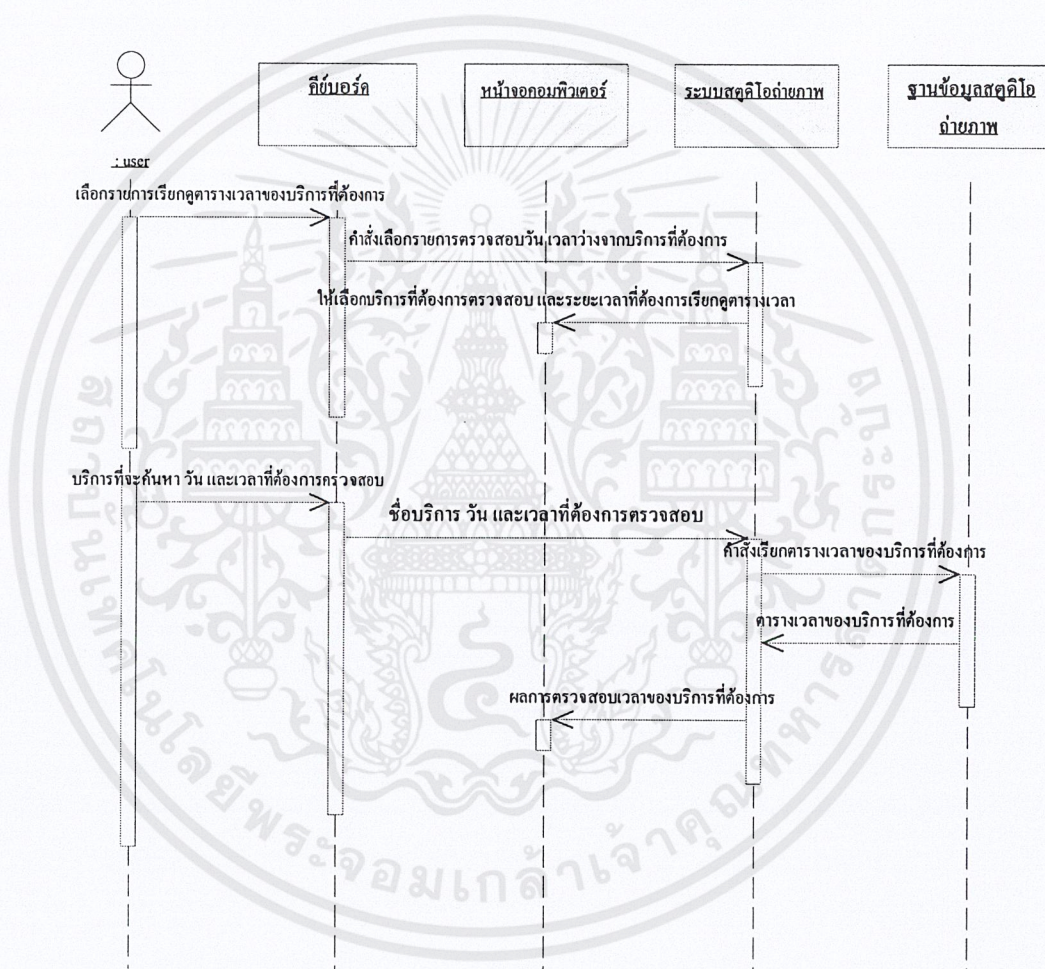
7.5.1.6 ตรวจสอบวันเวลาว่างของแพ็คเกจที่ต้องการ จากวัน เวลาที่ต้องการ



รูปที่ 7-11 Sequence Diagram การตรวจสอบวันเวลาว่างของแพ็คเกจ ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

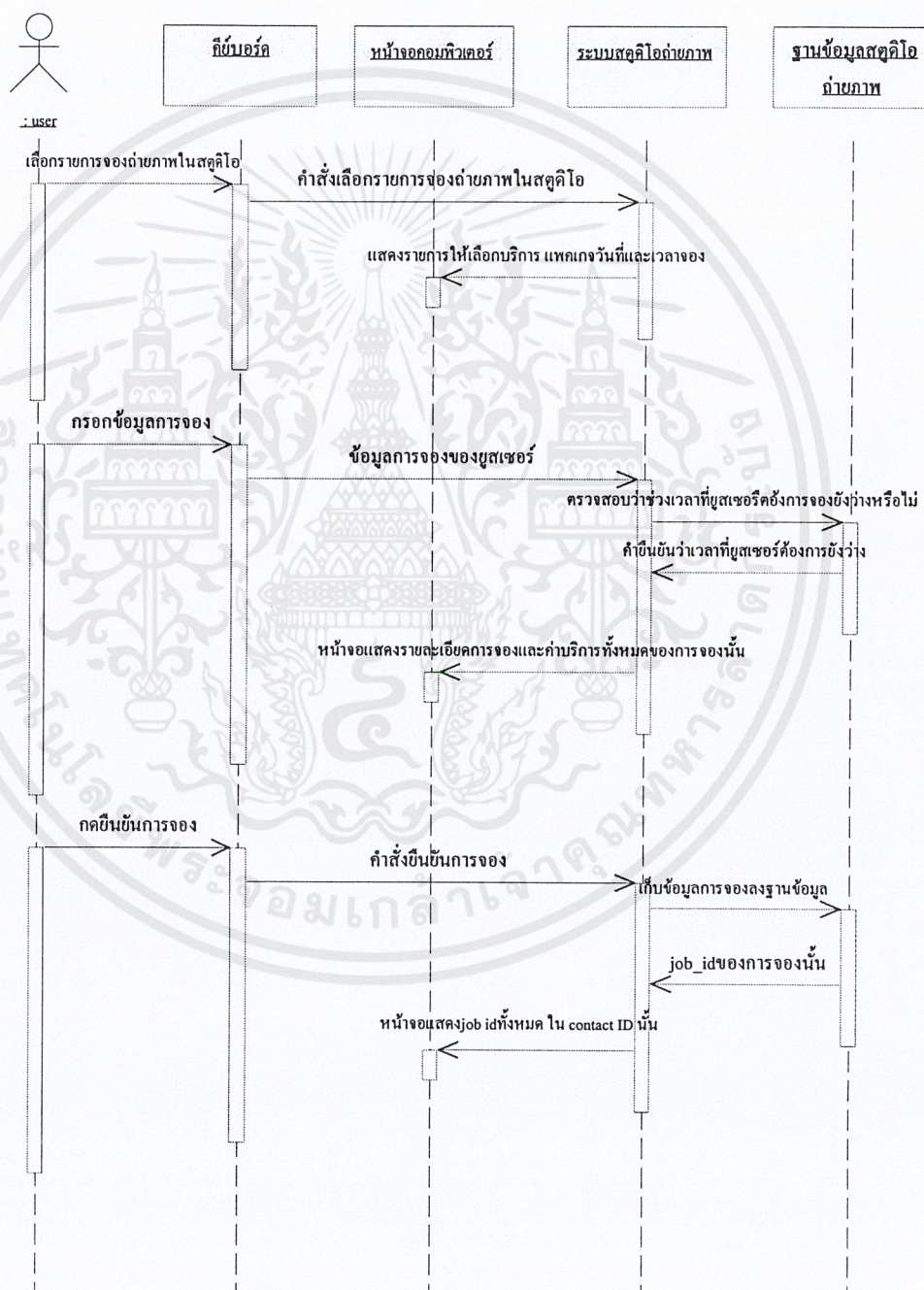
7.5.1.7 เรียกดูตารางเวลาล่วงหน้าตามจำนวนวันที่ต้องการ ของบริการที่ต้องการ



รูปที่ 7-12 Sequence Diagram การเรียกดู ตารางจองเวลาล่วงหน้า ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

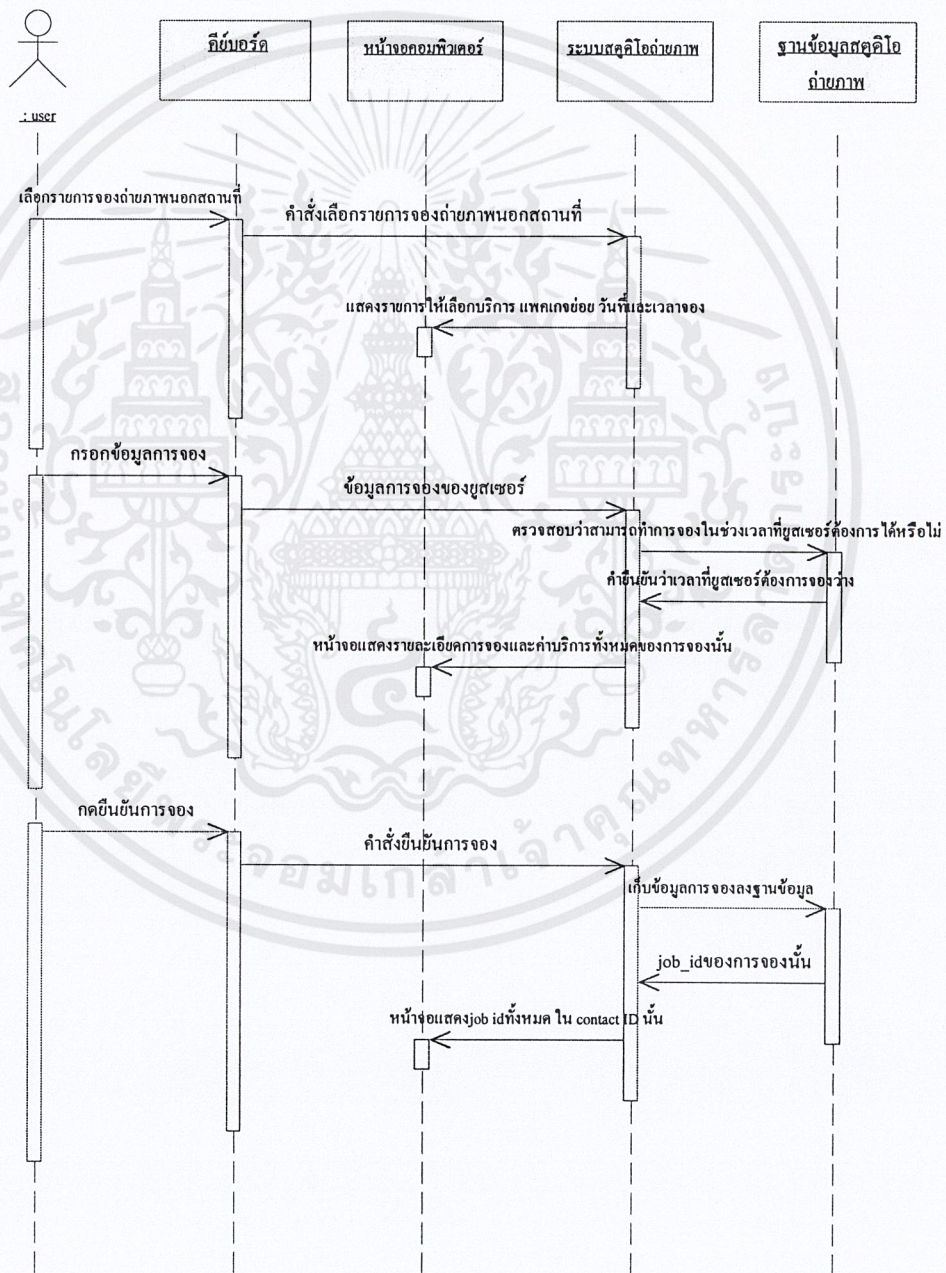
7.5.1.8 จองเวลาถ่ายภาพในสตูดิโอ



รูปที่ 7-13 Sequence Diagram การจองเวลาถ่ายภาพในสตูดิโอ ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

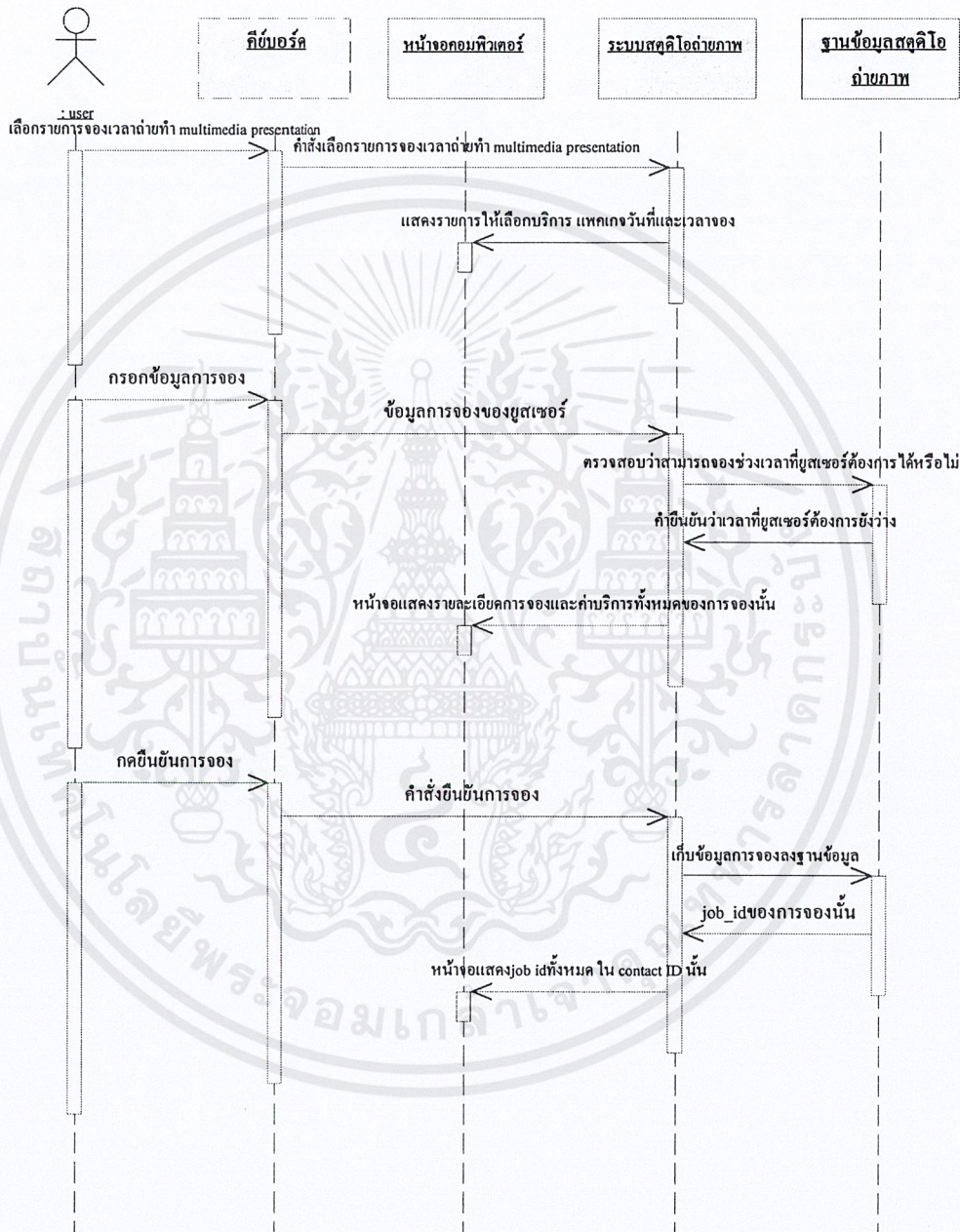
7.5.1.9 จองเวลาถ่ายภาพนอกสถานที่



รูปที่ 7-14 Sequence Diagram การจองเวลาถ่ายภาพนอกสถานที่ ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5.1.10 จองเวลาถ่ายทำวีดีโอมีดติมีเดียฟรีเซนต์ซัน

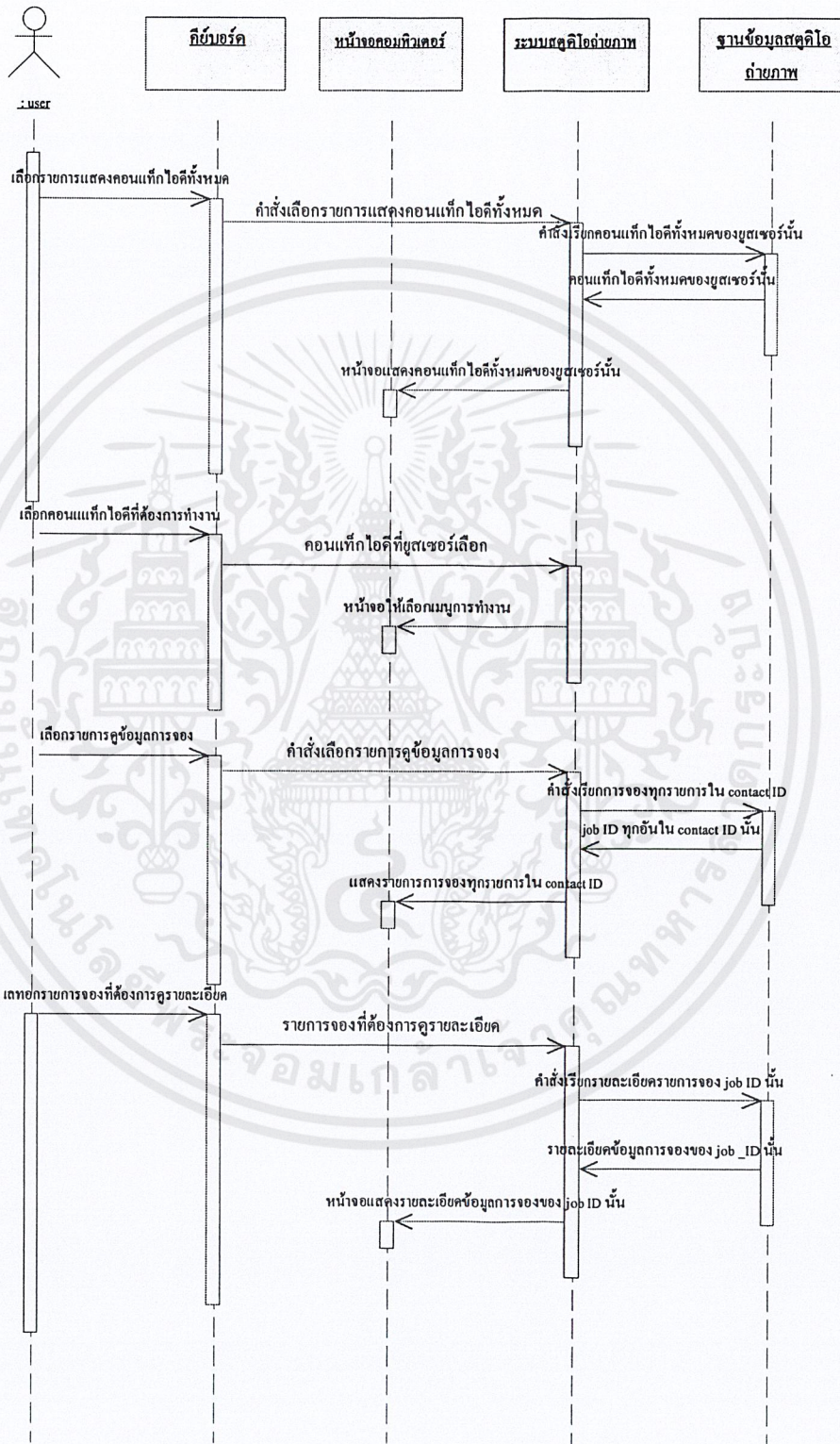


รูปที่ 7-15 Sequence Diagram การจองเวลาถ่ายทำวีดีโอมีดติมีเดีย ฟรีเซนต์ซัน ของระบบสตูดิโอถ่าย

ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

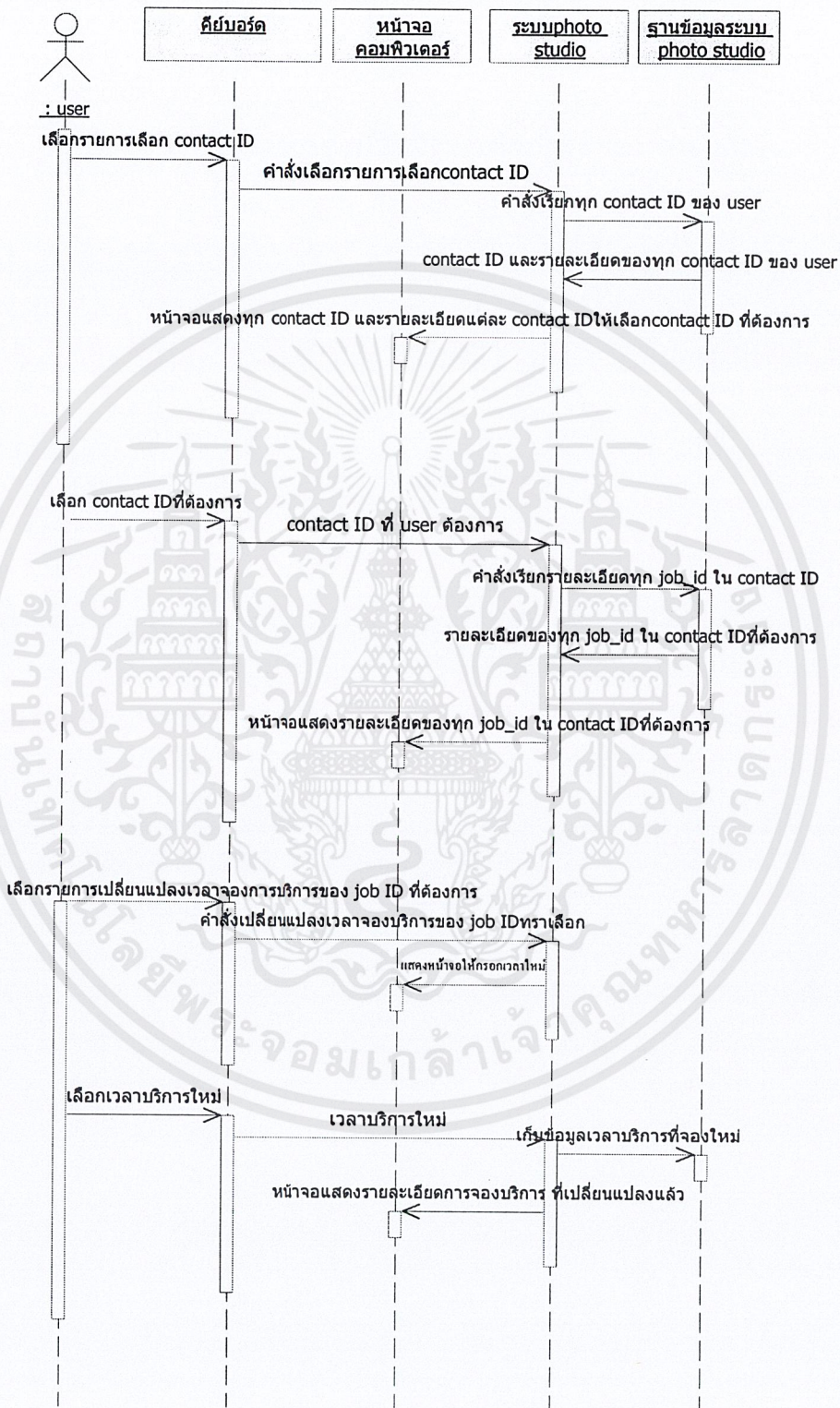
7.5.1.11 ดูข้อมูลการจองบริการ



รูปที่ 7-16 Sequence Diagram การดูข้อมูลการจองบริการของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

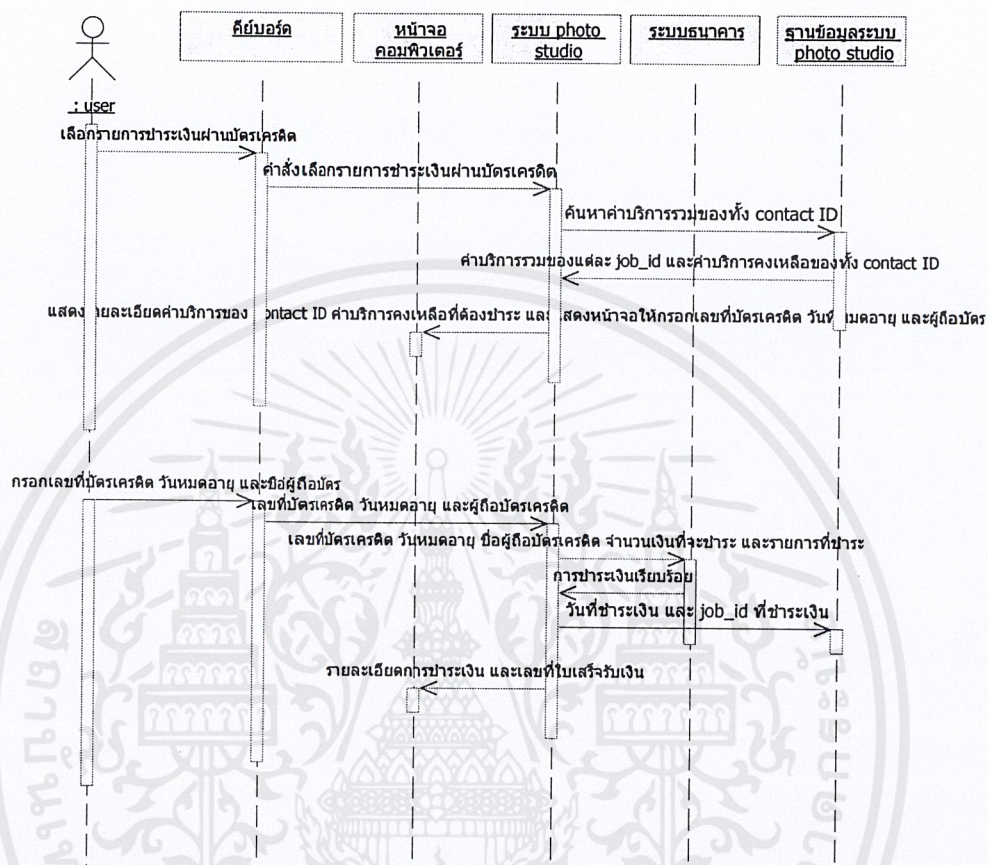
7.5.1.12 เปลี่ยนแปลงเวลาของบริการ



รูปที่ 7-17 Sequence Diagram การเปลี่ยนแปลงเวลาของบริการ ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

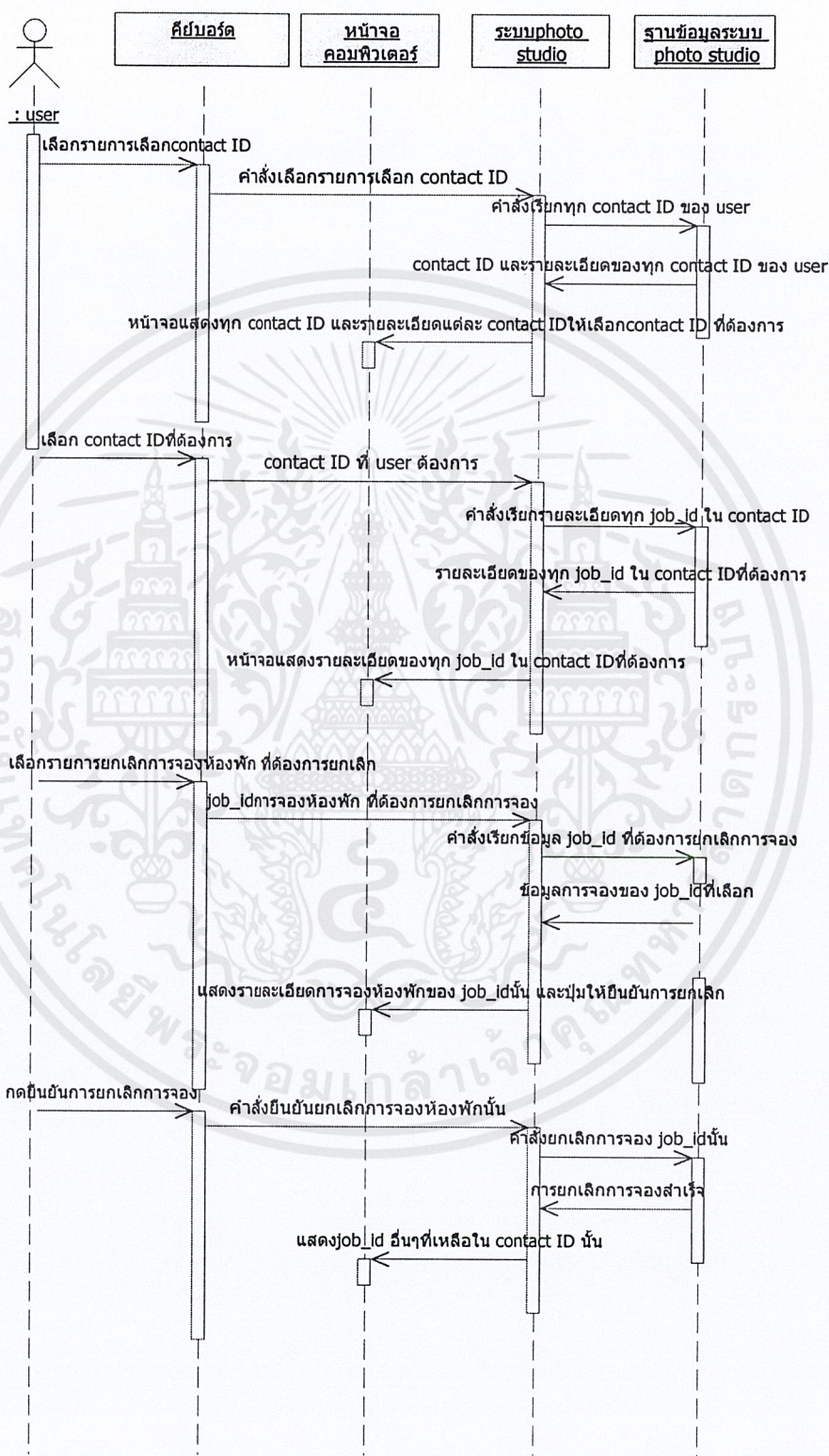
7.5.1.13 ชำระเงินผ่านบัตรเครดิต



รูปที่ 7-18 Sequence Diagram การชำระหนี้ผ่านบัตรเครดิต ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

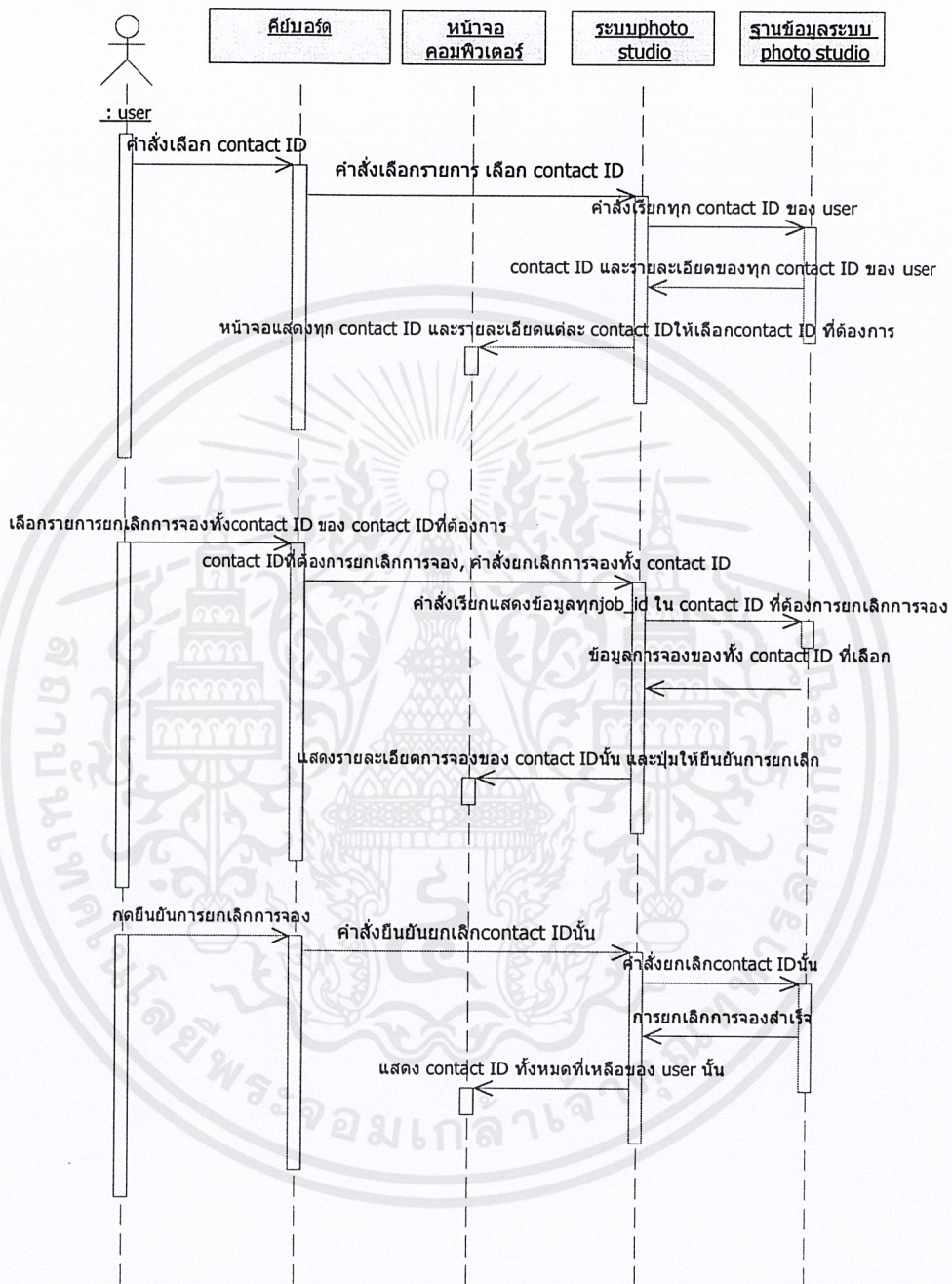
7.5.1.14 ยกเลิกการจองที่ละรายการ



รูปที่ 7-19 Sequence Diagram การยกเลิกการจองที่ละรายการของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5.1.15 ยกเลิกการจองทั้ง contact

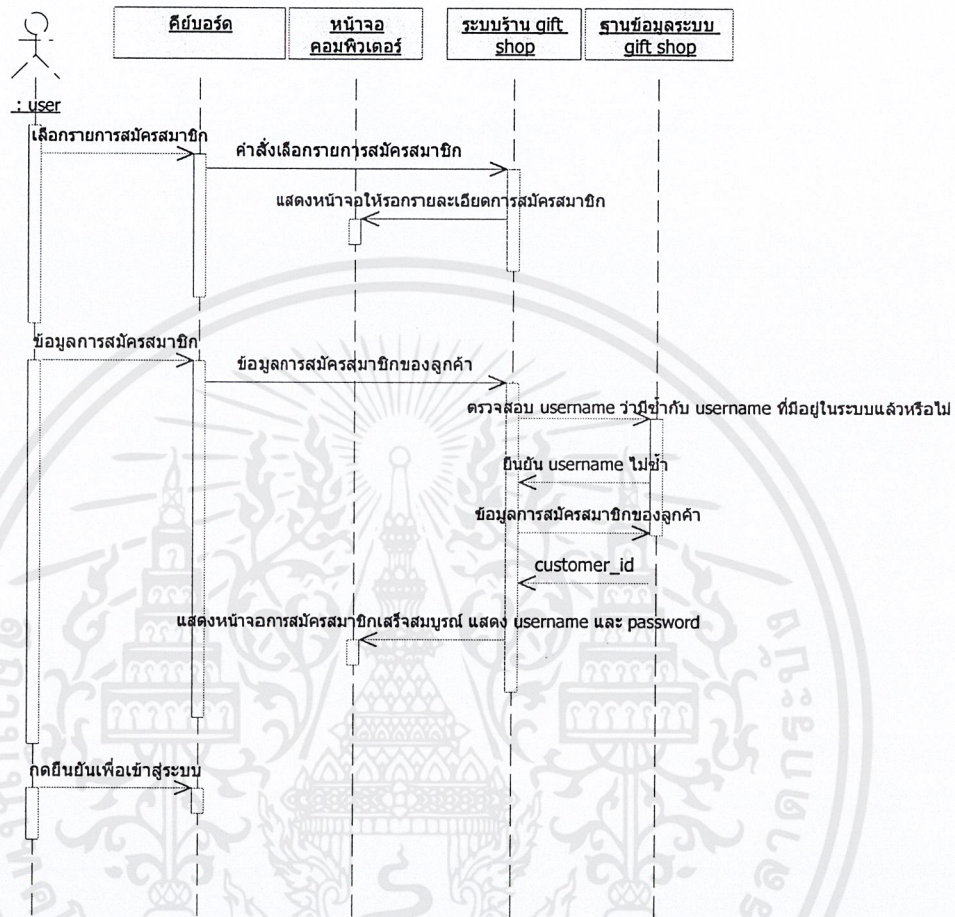


รูปที่ 7-20 Sequence Diagram การยกเลิกการจองทั้ง contact ของระบบสตูดิโอถ่ายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5.2 Sequence diagram ระบบร้านขายของชำร่วย

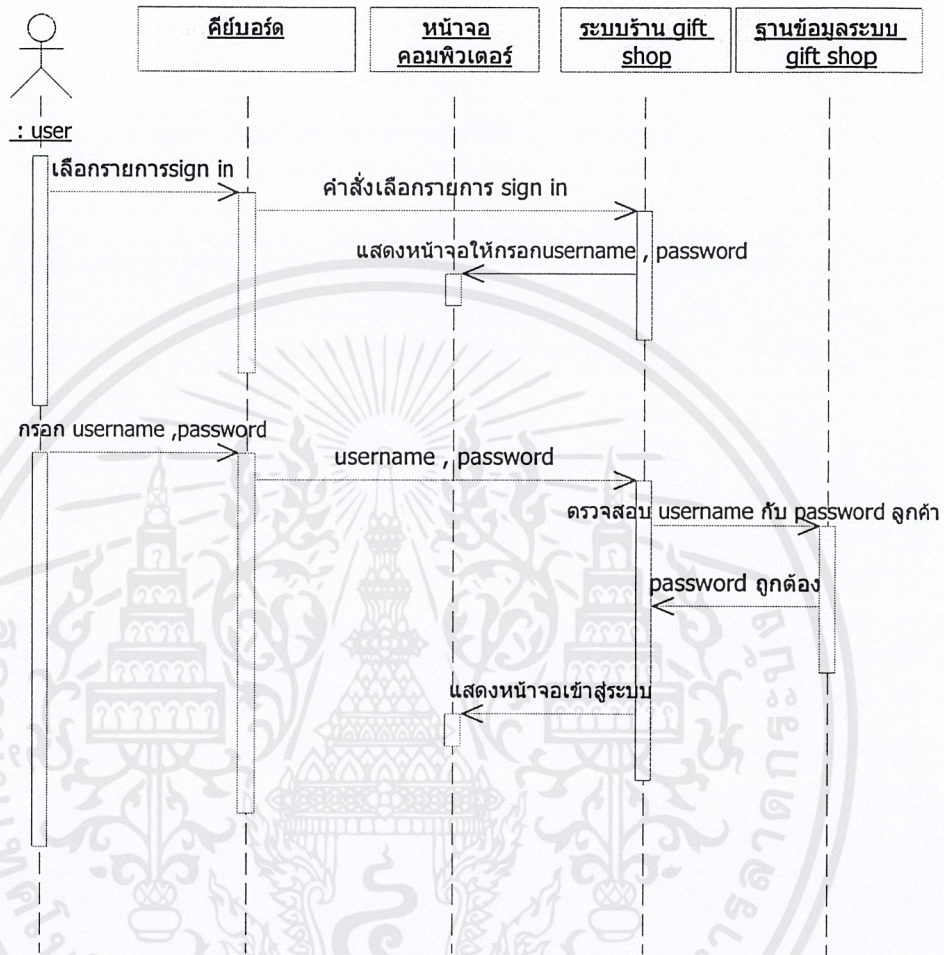
7.5.2.1 สมัครสมาชิก



รูปที่ 7-21 Sequence Diagram การสมัครสมาชิก ของระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

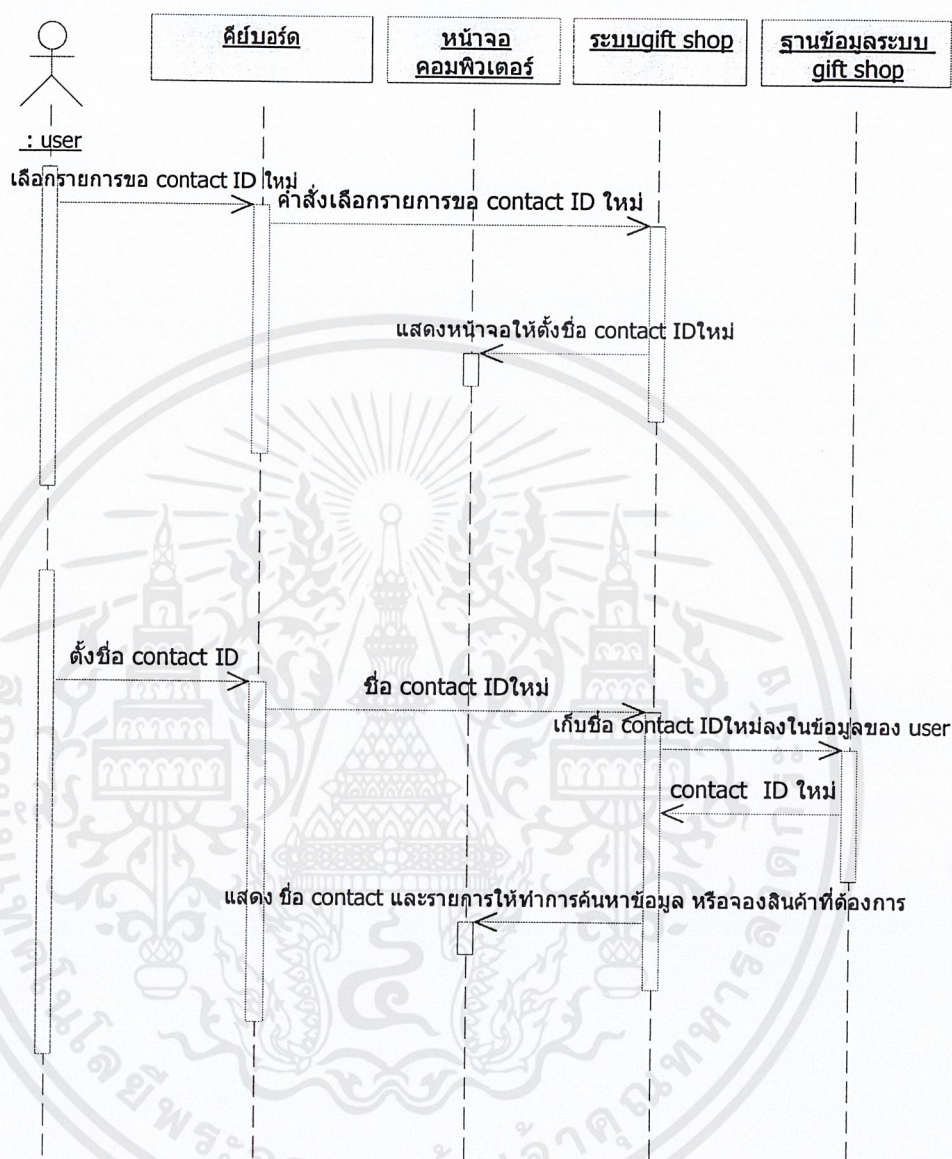
7.5.2.2 เข้าสู่ระบบ



รูปที่ 7-22 Sequence Diagram การเข้าสู่ระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

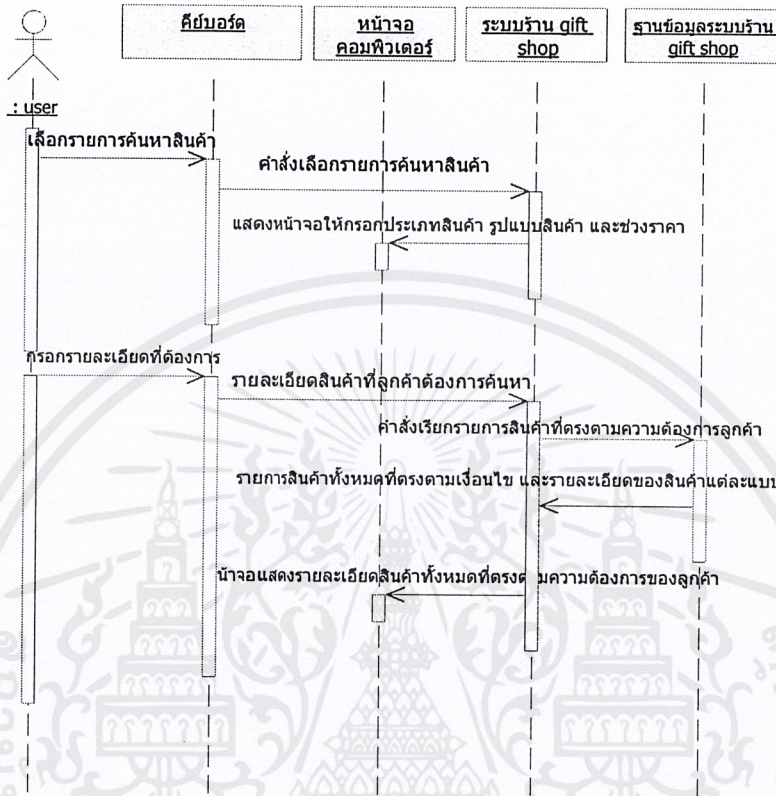
7.5.2.3 สร้าง contact ใหม่



รูปที่ 7-23 Sequence Diagram การสร้าง contact ใหม่ ของระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

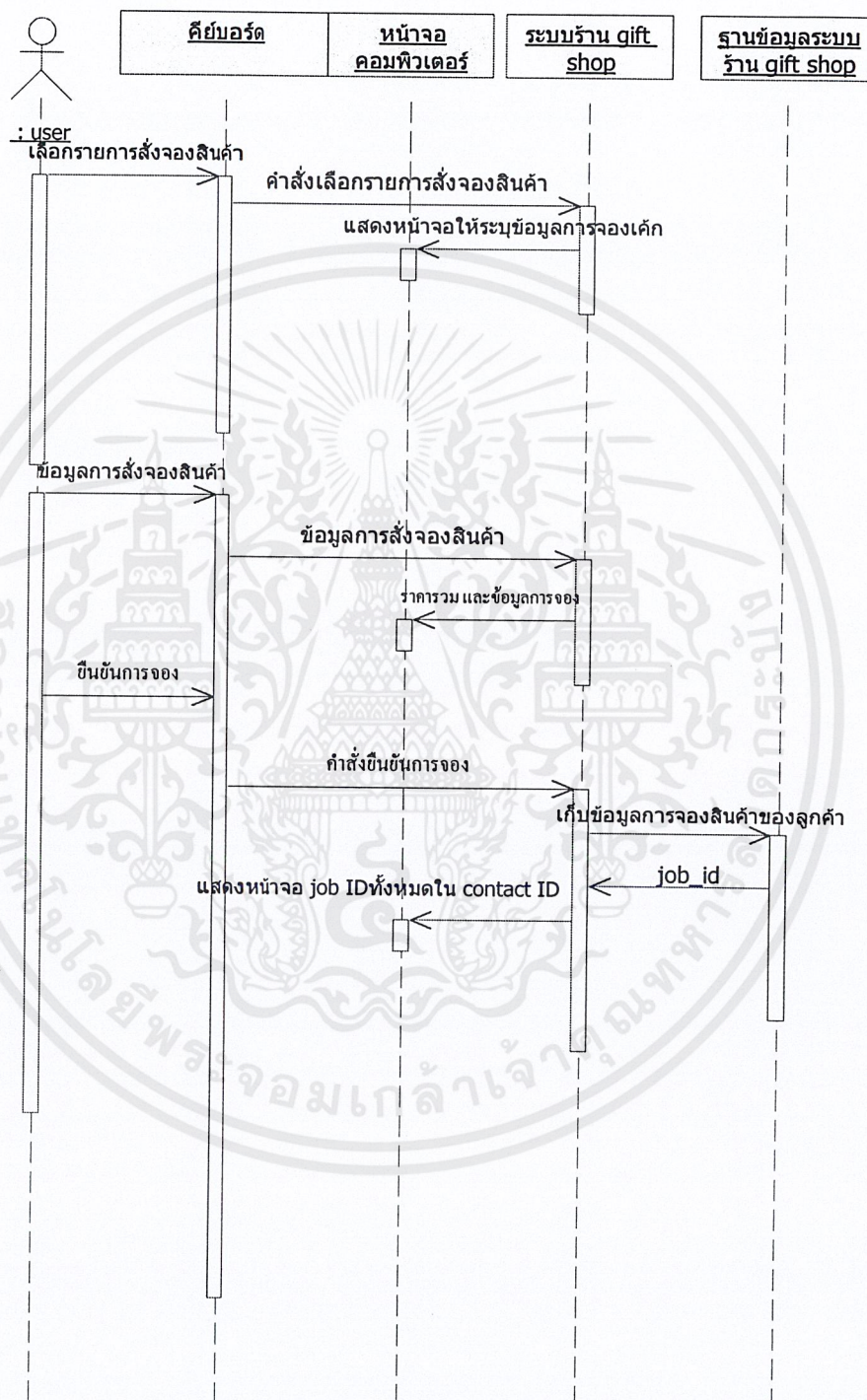
7.5.2.4 ค้นหาสินค้าตามประเภทและรูปแบบ



รูปที่ 7-24 Sequence Diagram การค้นหาสินค้า ของระบบร้านขายของชำร่วยตามประเภทและรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

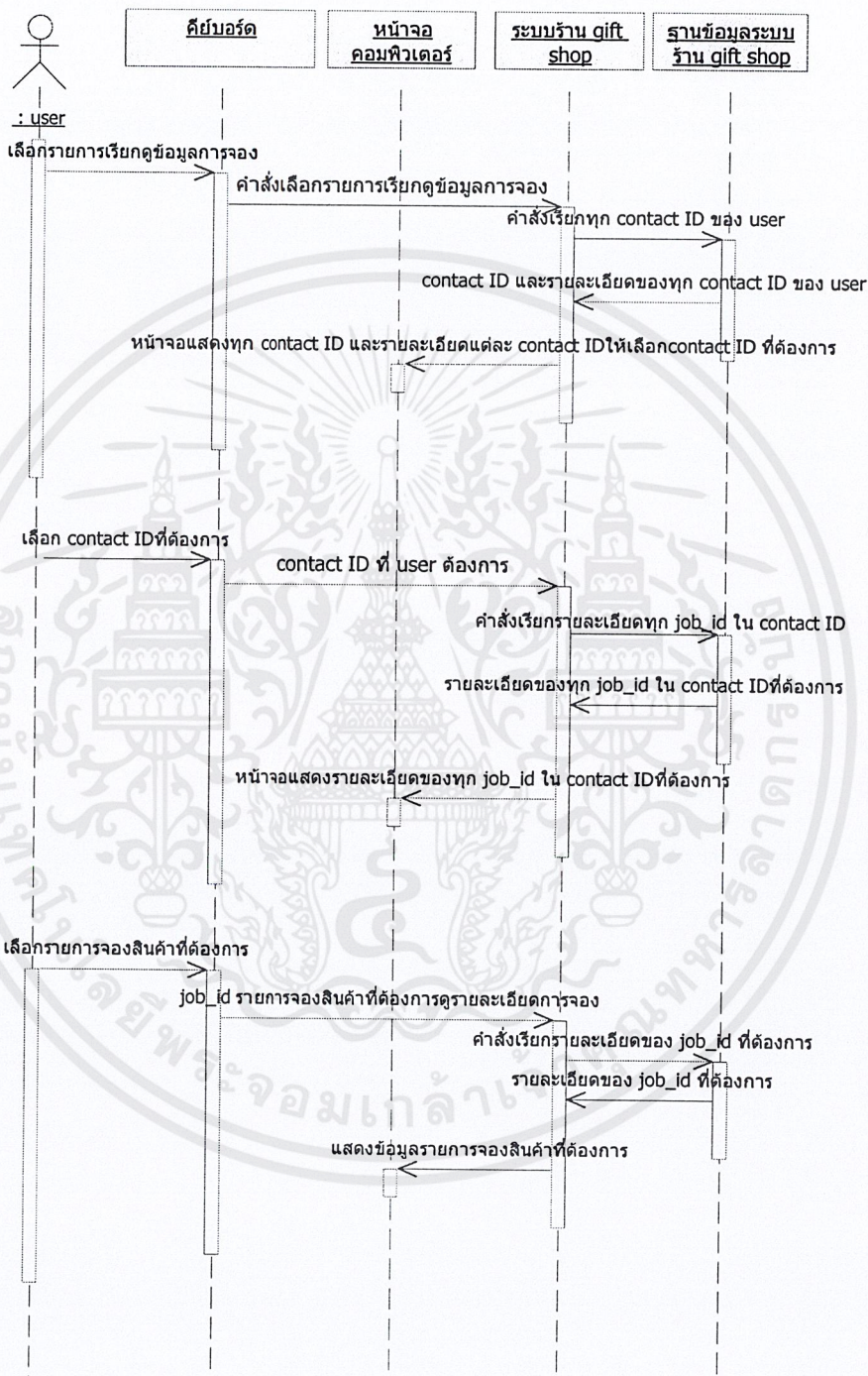
7.5.2.5 ตั้งจองสินค้า



รูปที่ 7-25 Sequence Diagram การตั้งจองสินค้า ของระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

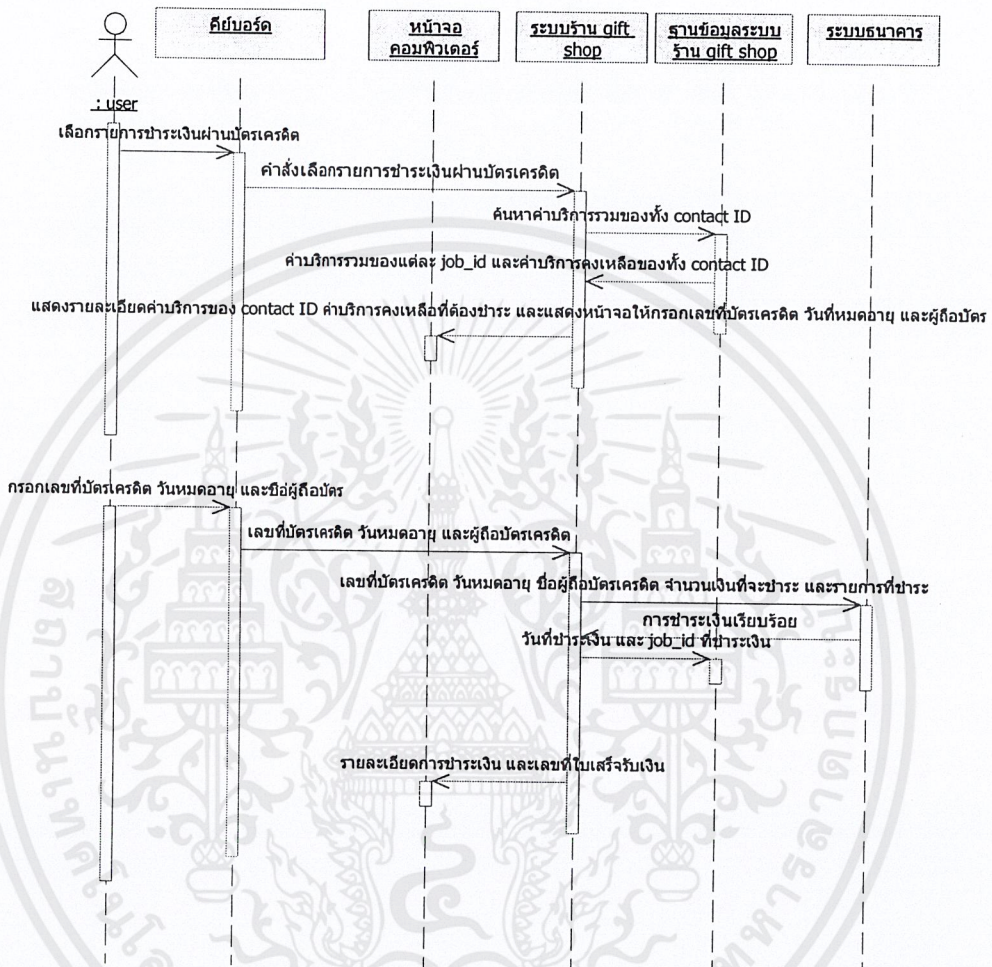
7.5.2.6 ดูข้อมูลการจองสินค้า



รูปที่ 7-26 Sequence Diagram การดูข้อมูลการจองสินค้า ของระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

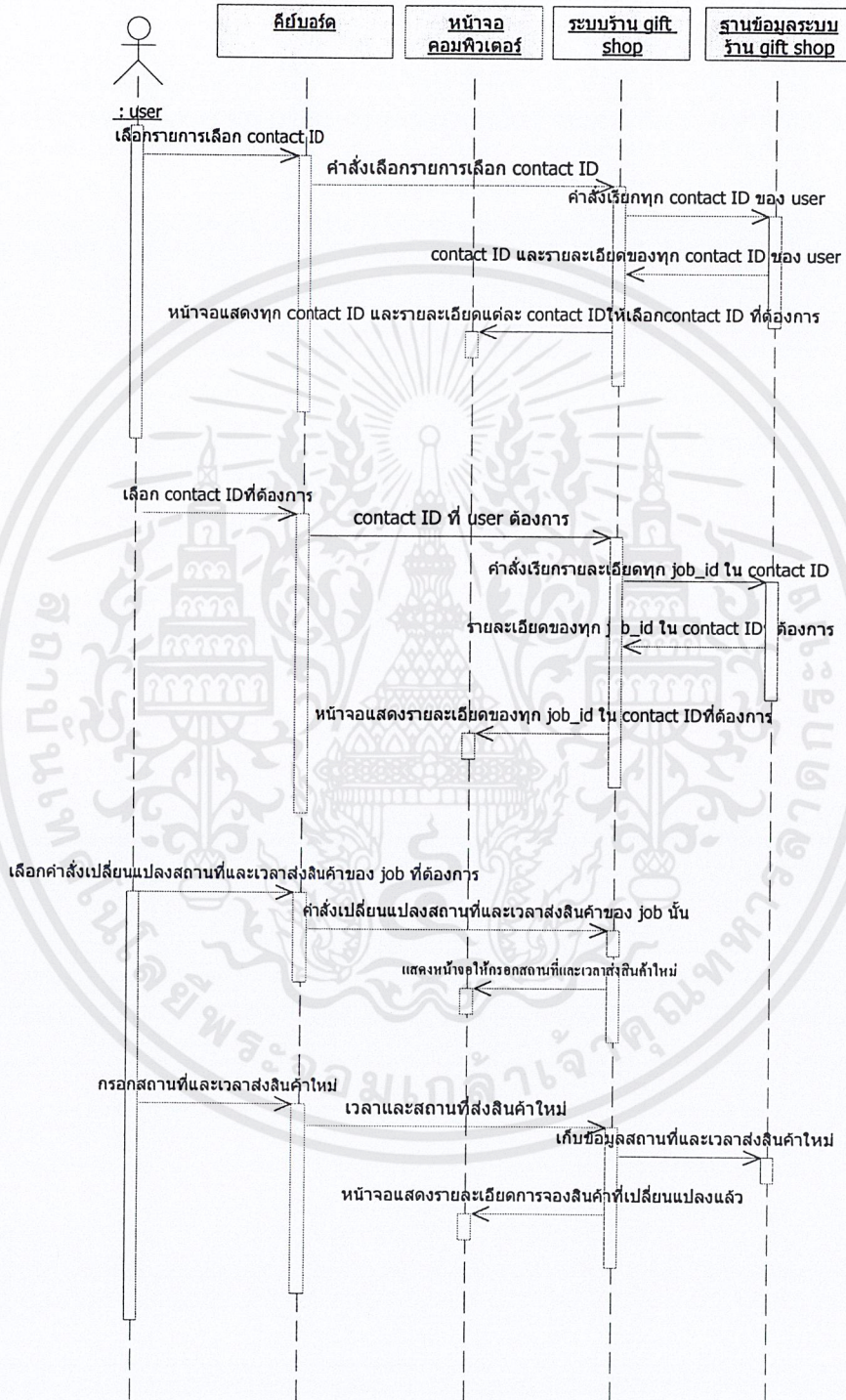
7.5.2.7 ชำระเงินผ่านบัตรเครดิต



รูปที่ 7-27 Sequence Diagram การชำระเงินผ่านบัตรเครดิต ของระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

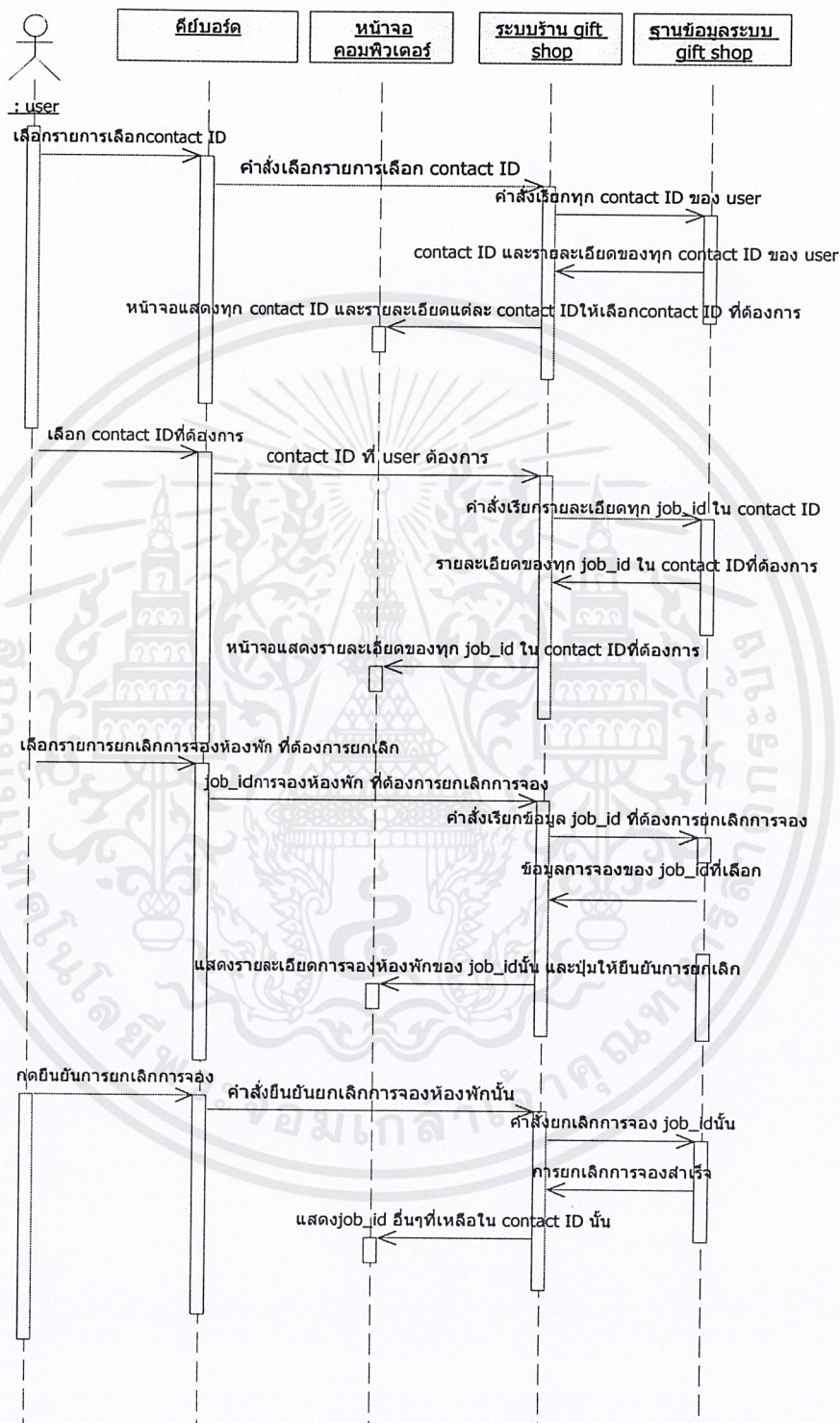
7.5.2.8 เปลี่ยนสถานที่และเวลาส่งสินค้า



รูปที่ 7-28 Sequence Diagram การเปลี่ยนแปลงสถานที่และเวลาส่งสินค้า ของระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

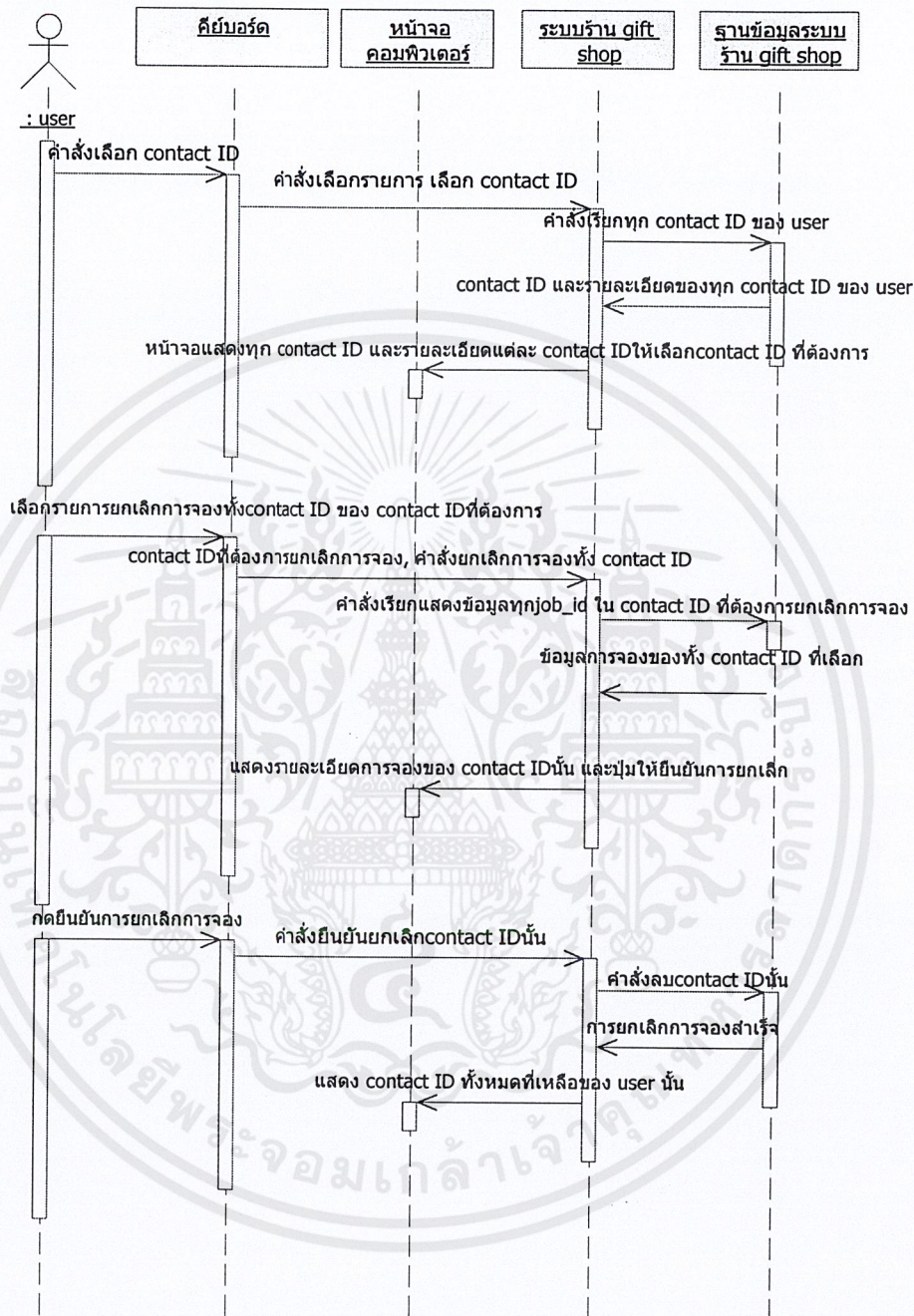
7.5.2.9 ยกเลิกการจองที่ละรายการ



รูปที่ 7-29 Sequence Diagram การยกเลิกการจองที่ละรายการ ของระบบร้านขายของชำราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5.2.10 ยกเลิกทั้ง contact

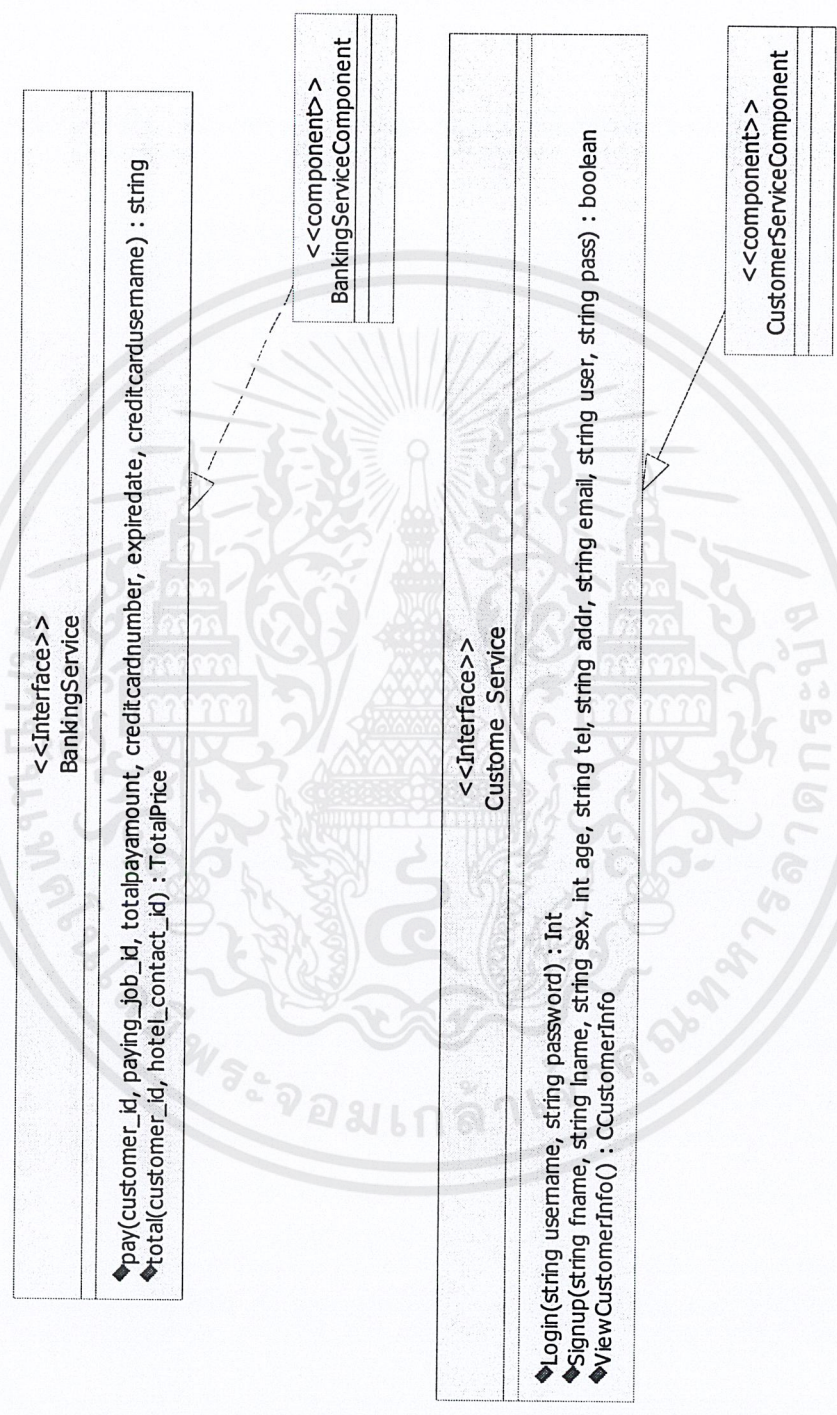


รูปที่ 7-30 Sequence Diagram การยกเลิกทั้ง contact ของระบบร้านขายของชำราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

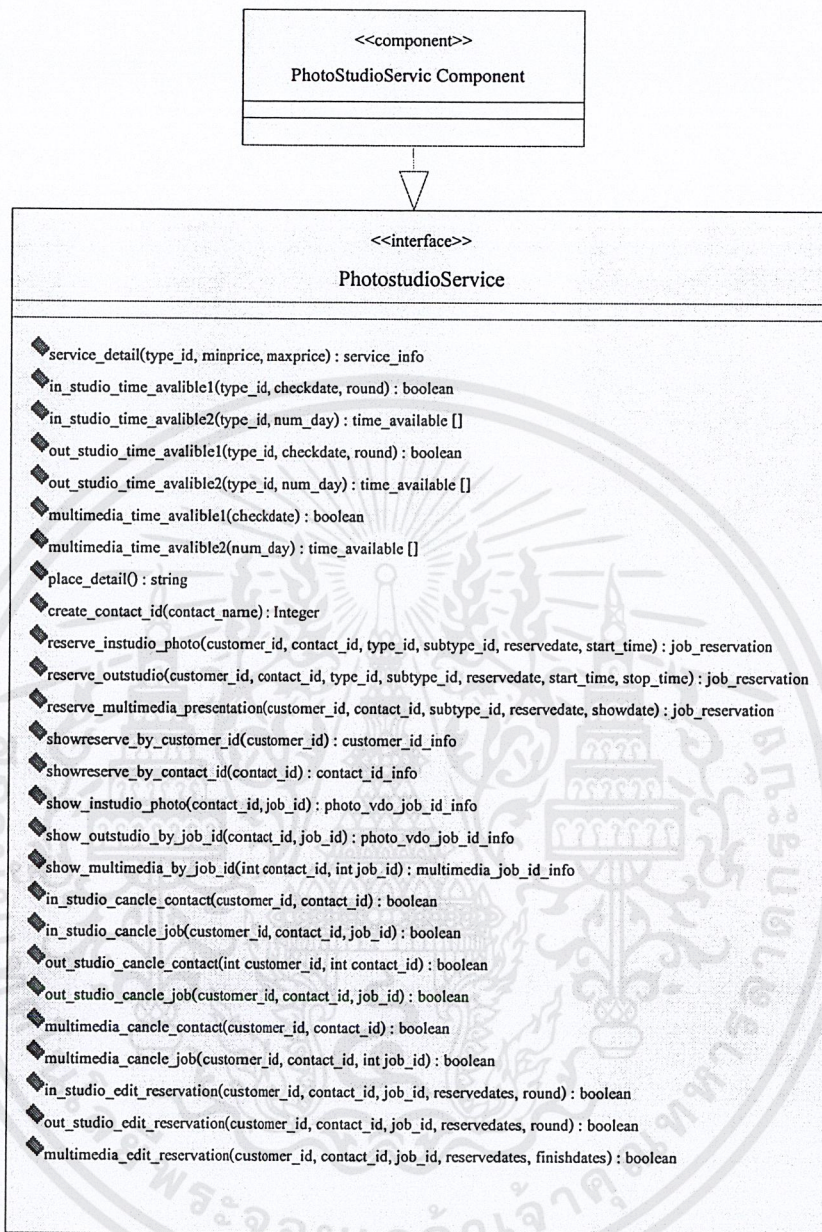
7.6 Class diagram

7.6.1 Class diagram ระบบสตูดิโอถ่ายภาพ



รูปที่ 7-31 Class Diagram ระบบสตูดิโอถ่ายภาพ ส่วนที่ 1

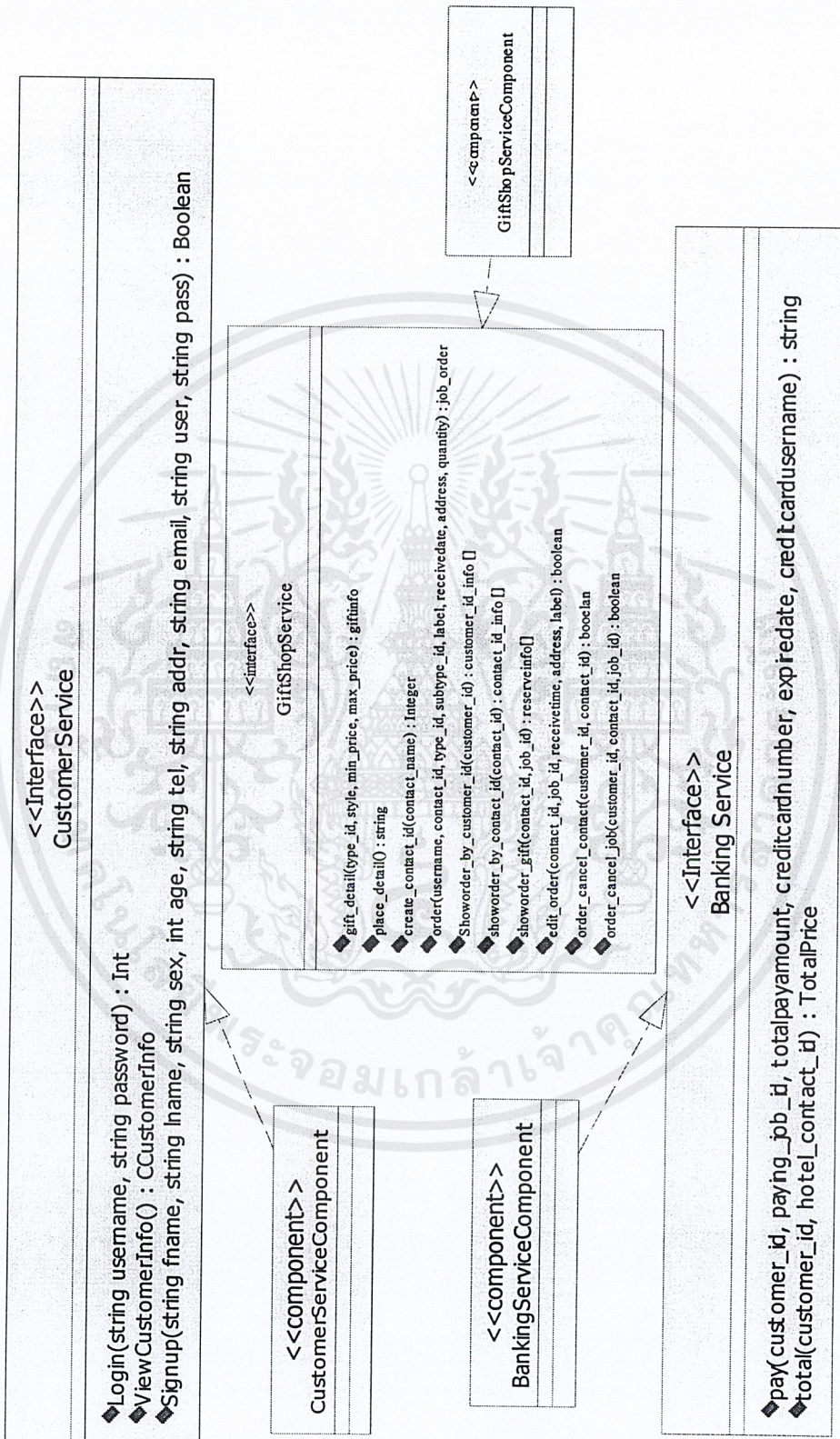
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-32 Class Diagram ระบบสตูดิโอถ่ายภาพ ส่วนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6.3 Class diagram ระบบร้านขายของชำราย

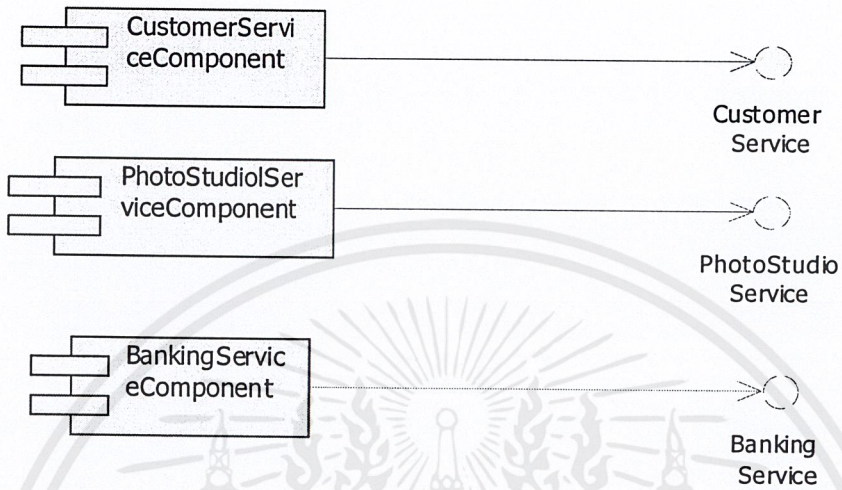


รูปที่ 7-33 Class Diagram ระบบร้านขายของชำราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

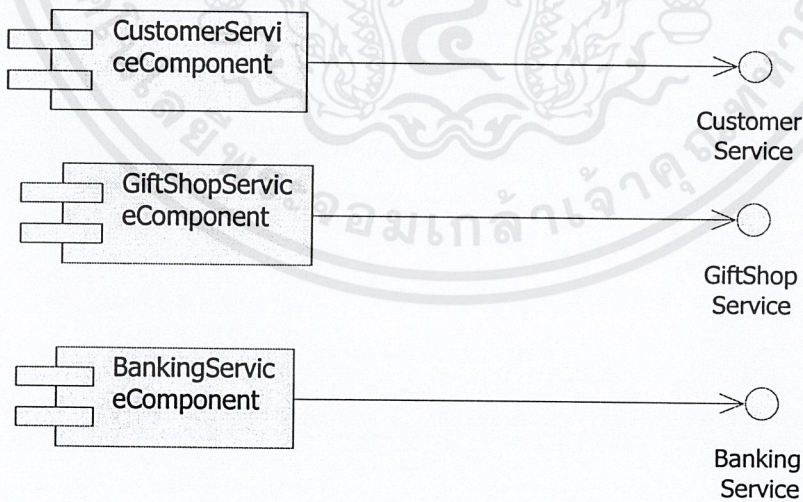
7.7 Component diagram

7.7.1 Component diagram ระบบสตูดิโอถ่ายภาพ



รูปที่ 7-34 Component Diagram ของระบบสตูดิโอถ่ายภาพ

7.7.2 Component diagram ระบบร้านขายของชำร่วย



รูปที่ 7-35 Component Diagram ของระบบร้านขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8 คู่มือการใช้งานระบบร้านขายของชำร่วย

7.8.1 ส่วนบริการสมาชิก

7.8.1.1 การสมัครสมาชิก

1. คลิกที่ “sign up now” บริเวณด้านซ้ายของหน้าเว็บ ระบบจะแสดงหน้าสำหรับกรอกข้อมูลสมัครสมาชิกใหม่

The screenshot shows the registration page for 'Gift Registries Shop'. The page has a header with the shop's logo and navigation links: HOME, Gift Package, Member Zone, SEARCH, and CONTACT. On the left, there is a 'ล็อกอินเข้าสู่ระบบ' (Login) section with fields for Username and Password, and a 'Login' button. Below that is a 'สมัครสมาชิกใหม่' (Sign up now) link and a small image of a gift certificate. The main registration form is titled 'สมัครสมาชิกใหม่ กรุณา กรอกข้อมูลให้' and contains the following fields:

- ชื่อ (Name): นายเมธา
- สกุล (Surname): ไพบูลย์ปัญญาเลิศ
- ที่อยู่ (Address): 11 เจริญกรุง24 สัมพันธวงศ์ กทม10100
- เพศ (Gender): ชาย หญิง
- อายุ (Age): 21 ปี
- โทรศัพท์ (Phone): 0-2864-3446
- อีเมล (Email): bnets@hotmail.co
- รหัสผ่าน (Password): bank
- รหัสผ่าน (Password): ****

At the bottom of the form is a 'ลงทะเบียน' (Register) button.

รูปที่ 7-36 แสดงหน้าการสมัครสมาชิก

2. กรอกข้อมูล ชื่อ นามสกุล เพศ อายุ หมายเลขโทรศัพท์ ที่อยู่ อีเมลแอดเดรส ตั้งชื่อผู้ใช้และพาสเวิร์ดที่จะใช้ในการล็อกอินเข้าสู่ระบบ แล้วคลิกปุ่ม “ลงทะเบียน”
3. ระบบจะแสดงหน้าจอลงทะเบียนเรียบร้อย หากจะทำการจองใดๆ ต้องทำการล็อกอินเข้าสู่ระบบก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.1.2 การล็อกอินเข้าสู่ระบบ

1. กรอกชื่อผู้ใช้และรหัสผ่านที่เมนู sign in ทางด้านซ้ายของหน้าเว็บ

ล็อกอินเข้าสู่ระบบ

username

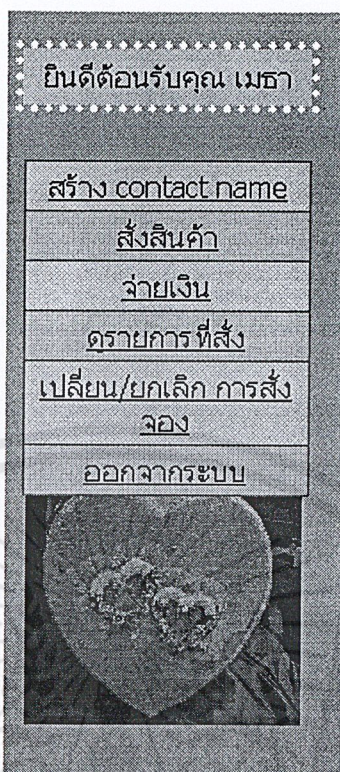
password

Login

สมัครสมาชิกใหม่

รูปที่ 7-37 แสดงส่วนการล็อกอินเข้าสู่ระบบ

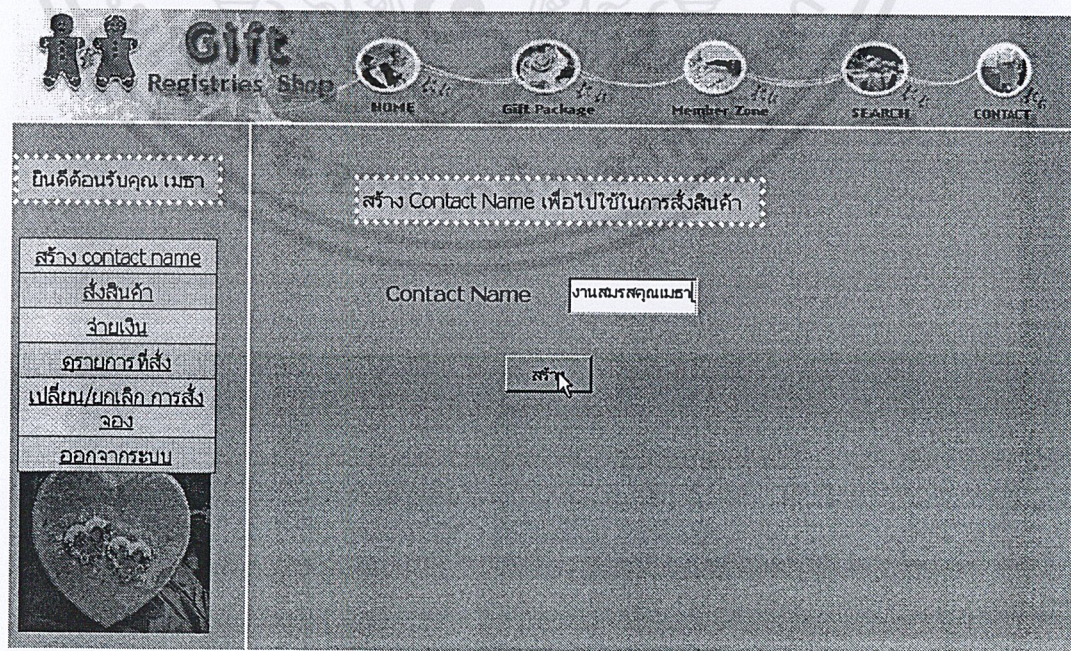
2. กดปุ่ม “sign in”
3. ระบบจะแสดงเมนูของสมาชิกทางด้านซ้ายของหน้าจอ ดังภาพ



รูปที่ 7-38 แสดงรายการต่างๆที่สมาชิกสามารถใช้งานได้

7.8.1.3 การสร้าง contact ID ใหม่

1. หลังจากทำการล็อกอินเข้าสู่ระบบแล้ว ทำการเลือกเมนู “สร้าง Contact ใหม่” ในเมนูทางด้านซ้าย ระบบจะแสดงหน้าเว็บให้กรอกชื่อของ contact ใหม่
2. ตั้งชื่อของ contact ใหม่ที่ต้องการ โดยกรอกลงในช่อง ชื่อcontact



รูปที่ 7-39 แสดงหน้าจอในการขอ contact ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กดปุ่ม “สร้าง”
4. ระบบจะแสดงหน้าส่งของสินค้า

7.8.1.4 การออกจากระบบ

1. หลังจากทำการล็อกอินเข้าสู่ระบบและทำงานเสร็จสิ้นแล้วต้องการออกจากระบบ เลือกเมนู “ออกจากระบบ” ทางด้านซ้ายของหน้าเว็บ
2. ระบบจะแสดงหน้าจอให้ทำการ sign in ใหม่ได้

7.8.2 การสั่งซื้อสินค้า

1. หลังจากทำการล็อกอินเข้าสู่ระบบแล้ว เลือกรายการสั่งซื้อสินค้าที่ปรากฏในเมนูทางซ้ายของหน้าจอ
2. เลือกหมวดสินค้าที่ต้องการจอง หลังจากเลือกหมวดสินค้าแล้ว ระบบจะแสดงรายละเอียดของสินค้าทุกชนิดในหมวดนั้น
3. เลือกสินค้าที่ต้องการจอง

| หมวดสินค้า | ชนิด | รายละเอียด | ราคา | รูปตัวอย่าง | เลือกสินค้า |
|------------|-----------------------|--|------|-------------|-------------|
| ครอบรูป | ครอบรูปสีฟ้าเล็ก สดใส | ครอบรูปสีฟ้าเล็ก สดใส เหมือนเก็กรอคคแต่ง | 12 | | เลือก |
| ครอบรูป | ครอบรูป เล็กน่ารัก | ครอบรูป เล็กน่ารักสีชมพู | 12 | | เลือก |

รูปที่ 7-40 แสดงหน้าเลือกสินค้าที่ต้องการสั่งซื้อ

4. กรอกรายละเอียดการจองโดยเลือกว่าจะทำการจองใน contact ไหน กำหนดแบบอักษร ตัวอักษรและข้อความที่จะพิมพ์ลงบนสินค้า กำหนดสถานที่ส่งสินค้า จำนวนสินค้าที่ต้องการ และวันที่ต้องการรับสินค้า เมื่อเสร็จสิ้นแล้วกดปุ่ม “จอง”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยินดีต้อนรับคุณ เมธา

กรอกรายละเอียดการสั่งซื้อสินค้า

สร้าง contact name

ส่งสินค้า

จ่ายเงิน

ดูรายการที่ส่ง

เปลี่ยน/ยกเลิก การส่ง
ของ

ออกจากระบบ

กรอกกรณขณยบดการส่งซื้อสินค้า

Contact Name

แบบตัวอักษร

สีตัวอักษร

ข้อความ

ที่อยู่

จำนวนชิ้น

วันที่รับสินค้า เดือน ปี

รูปที่ 7-41 แสดงหน้ากรอกรายละเอียดการซื้อสินค้า

ระบบจะแสดงหน้าจอระบุค่าสินค้าทั้งหมดในการจองนี้ พร้อมทั้งวันสุดท้ายที่จะทำการชำระเงินเพื่อยืนยันการจองได้

ยินดีต้อนรับคุณ เมธา

ทำการส่งสินค้าเรียบร้อยแล้ว ขอขอบคุณที่ใช้บริการ

ราคาของสินค้า : 12000

กรุณาชำระภายในวันที่ 21/10/2003

สร้าง contact name

ส่งสินค้า

จ่ายเงิน

ดูรายการที่ส่ง

เปลี่ยน/ยกเลิก การส่ง
ของ

ออกจากระบบ

รูปที่ 7-42 หน้าจอแสดงการสั่งซื้อสินค้าเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.3 การค้นหาสินค้าตามเงื่อนไขที่ต้องการ

1. คลิกที่รายการ “search” บนแถบเมนูหลัก ด้านบนของหน้าจอ ระบบจะแสดงหน้าจอให้ค้นหาสินค้า
2. กำหนดเงื่อนไขสินค้าที่ต้องการค้นหา ได้แก่ชนิดสินค้า รูปแบบสินค้า ราคาต่ำสุด สูงสุดต่อชิ้น

รูปที่ 7-43 แสดงหน้ากรอกรายละเอียดการค้นหาสินค้า

3. กดปุ่ม “ค้นหา” ระบบจะทำการค้นหาห้องพักรับพัสดุที่ตรงตามความต้องการ และแสดงผลการค้นหา

| หมวดสินค้า | ชนิด | รายละเอียด | ราคา | รูปตัวอย่าง |
|------------|-----------------------|---|------|-------------|
| ของขวัญ | ของขวัญสีฟ้าเล็ก สดใส | ของขวัญสีฟ้าเล็ก สดใส เหมาะแก่การตกแต่ง | 12 | |
| ของขวัญ | ของขวัญ เล็กน่ารัก | ของขวัญ เล็กน่ารักสีชมพู | 12 | |

รูปที่ 7-44 แสดงหน้าผลการค้นหาสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.4 การดูข้อมูลการจอง

1. การดูข้อมูลการจอง จะทำได้เมื่อทำการล็อกอินเข้าระบบแล้วเท่านั้น โดยเลือกรายการดูรายการที่ตั้ง บนเมนูทางด้านซ้ายของหน้าเว็บ ระบบจะแสดงหน้าจอรายละเอียดการตั้งจอง ซึ่งประกอบด้วย contact ทั้งหมดที่มีของ ยูสเซอร์นั้น

ยินดีต้อนรับคุณ เมธา

รายละเอียดการสั่งซื้อสินค้าของคุณ

| contact name | ราคาทั้งหมด | วันกำหนดชำระเงิน |
|-----------------------------|-------------|------------------|
| งานสมรสคุณเมธา | 12000 | 2 10 2003 0:0:0 |
| งานเลี้ยงวันสวนศกหลาย | 6000 | 29 6 2003 0:0:0 |
| งานฉลองจบการศึกษาห้อง Qlala | 420 | 22 3 2003 0:0:0 |

สร้าง contact name

ส่งสินค้า

จ่ายเงิน

ดูรายการที่สั่ง

เปลี่ยน/ยกเลิก การสั่งซื้อ

ออกจากระบบ

รูปที่ 7-45 แสดงหน้ารายการ contact ทั้งหมดของผู้ใช้

2. เลือก contact ที่ต้องการ หน้าจอจะแสดงรายการสั่งซื้อสินค้าทุกรายการที่มีอยู่ใน contact นั้น พร้อมทั้งวันที่รับของของรายการตั้งสินค้าแต่ละรายการ

ยินดีต้อนรับคุณ เมธา

รายละเอียดการสั่งซื้อสินค้าของคุณใน contact:

| สินค้าที่สั่งจอง | วันรับของ |
|---------------------|-------------------|
| กรอบรูปฟ้าเล็ก สดใส | 12 10 2003 10:0:0 |

สร้าง contact name

ส่งสินค้า

จ่ายเงิน

ดูรายการที่สั่ง

เปลี่ยน/ยกเลิก การสั่งซื้อ

ออกจากระบบ

รูปที่ 7-46 แสดงการเลือกรายการย่อยที่ต้องการดูข้อมูลการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกรายการสั่งซื้อสินค้าที่ต้องการดูข้อมูล ระบบจะแสดงรายละเอียดการสั่งซื้อสินค้าที่ถูกเลือก

| ชนิด | รายละเอียด | ข้อความ | วันรับของ | จำนวนชิ้น | ราคาทั้งหมด |
|------------------------|--|------------------------|-------------------|-----------|-------------|
| กรอบรูปสี่เหลี่ยม สดใส | กรอบรูปสี่เหลี่ยม สดใส เหนวนท์ การตกแต่ง | แวงค์-ส้ม 21 ค.ค. 2546 | 12 10 2003 10:0:0 | 1000 | 12000 |

รูปที่ 7-47 แสดงหน้ารายละเอียดการสั่งซื้อสินค้าของรายการจองที่เลือก

7.8.5 การยกเลิกการจอง

7.8.5.1 การยกเลิกที่ละรายการจอง

1. การยกเลิกรายการจองย่อยที่ละรายการ จะทำได้เมื่อทำการล็อกอินเข้าระบบแล้วเท่านั้น โดยเลือกรายการ “เปลี่ยน/ยกเลิกการสั่งจอง” ระบบจะแสดงหน้าจอ contact ทั้งหมดที่มีของผู้ใช้
2. เลือก “edit” ที่ contact ที่ต้องการเปลี่ยนแปลงข้อมูล หรือยกเลิกการสั่งจอง หน้าจอจะแสดงรายการจองทุกรายการที่มีอยู่ใน contact นั้น

| contact name | ราคาทั้งหมด | วันกำหนดชำระเงิน | edit | cancel |
|-----------------------------|-------------|------------------|------|--------|
| งานสมรสคุณเมธา | 12000 | 2 10 2003 0:0:0 | edit | cancel |
| งานเลี้ยงรุ่นสวนกุหลาบ | 6000 | 29 6 2003 0:0:0 | edit | cancel |
| งานฉลองจบการศึกษาห้อง Olala | 420 | 22 3 2003 0:0:0 | edit | cancel |

หากต้องการยกเลิกทั้ง contact ให้กดที่ cancel แต่ถ้าต้องการเข้าไปแก้ไขงาน, หรือยกเลิกงาน ให้กดที่ edit เพื่อเข้าไปเลือกดูจากงานทั้งหมดที่มีใน contact นั้น

รูปที่ 7-48 แสดงหน้าการเลือก edit ของ contact ที่ต้องการแก้ไขข้อมูลการสั่งซื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือก “edit” ที่รายการจองย่อยที่ต้องการยกเลิกการจอง ระบบจะแสดงหน้าจอรายละเอียดการจองนั้น

The screenshot shows the website interface with a navigation bar at the top containing icons for HOME, Gift Package, Member Zone, SEARCH, and CONTACT. On the left, there is a sidebar with a menu: ยินดีต้อนรับคุณ เมธา, สร้าง contact name, สั่งสินค้า, จ่ายเงิน, รายการที่สั่ง, เปลี่ยน/ยกเลิก การสั่งจอง, and ออกจากระบบ. Below the menu is a small image of a gift. The main content area has a heading: เลือก งาน ที่ต้องการยกเลิกหรือเปลี่ยนแปลง. Below this is a table with three columns: สินค้าที่สั่งจอง, วันรับของ, and edit. The first row contains: กรอบรูปสี่เหลี่ยม สดใส, 12 10 2003 10:0:0, and edit. A mouse cursor is pointing at the 'edit' link.

| สินค้าที่สั่งจอง | วันรับของ | edit |
|------------------------|-------------------|------|
| กรอบรูปสี่เหลี่ยม สดใส | 12 10 2003 10:0:0 | edit |

รูปที่ 7-49 แสดงการเลือกคำสั่ง edit ของรายการจองที่ต้องการยกเลิก

4. กด cancel เพื่อทำการยกเลิกการจองรายการนั้น

The screenshot shows the website interface with a navigation bar at the top containing icons for HOME, Gift Package, Member Zone, SEARCH, and CONTACT. On the left, there is a sidebar with a menu: ยินดีต้อนรับคุณ เมธา, สร้าง contact name, สั่งสินค้า, จ่ายเงิน, รายการที่สั่ง, เปลี่ยน/ยกเลิก การสั่งจอง, and ออกจากระบบ. Below the menu is a small image of a gift. The main content area has a heading: รายละเอียดการสั่งซื้อสินค้าของคุณ กรุณาเลือก งาน ที่ต้องการยกเลิกหรือเปลี่ยนแปลง. Below this is a table with seven columns: ชนิด, รายละเอียด, ข้อความ, วันรับของ, จำนวน ชิ้น, ราคาทั้งหมด, edit, and cancel. The first row contains: กรอบรูปสี่เหลี่ยม สดใส, กรอบรูปสี่เหลี่ยม สดใส เหมือนกับการตกแต่ง, แบบคัสตม 21 ต. ค. 2546, 12 10 2003 10:0:0, 1000, 12000, edit, and cancel. A mouse cursor is pointing at the 'cancel' link.

| ชนิด | รายละเอียด | ข้อความ | วันรับของ | จำนวน ชิ้น | ราคาทั้งหมด | edit | cancel |
|------------------------|---|------------------------|-------------------|------------|-------------|------|--------|
| กรอบรูปสี่เหลี่ยม สดใส | กรอบรูปสี่เหลี่ยม สดใส เหมือนกับการตกแต่ง | แบบคัสตม 21 ต. ค. 2546 | 12 10 2003 10:0:0 | 1000 | 12000 | edit | cancel |

รูปที่ 7-50 แสดงการเลือกคำสั่ง cancel เพื่อทำการยกเลิกการจองรายการสั่งซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.5.2 การยกเลิกการจองทั้ง contact

1. การยกเลิกทั้ง contact จะทำได้เมื่อทำการล็อกอินเข้าระบบแล้วเท่านั้น โดยเลือกรายการ “เปลี่ยน/ยกเลิกการจอง” บนเมนูทางด้านซ้ายของหน้าเว็บ ระบบจะแสดงหน้าจอ contact ทั้งหมดที่มีของผู้ใช้นั้น
2. เลือก “cancel” ที่ contact ที่ต้องการยกเลิกการจองทั้ง contact

The screenshot shows the 'Gift Registries Shop' interface. On the left, there is a navigation menu with options like 'สร้าง contact name', 'ส่งสินค้า', 'จ่ายเงิน', 'ดูรายการที่ส่ง', 'เปลี่ยน/ยกเลิก การจอง', and 'ออกจากระบบ'. The main content area displays a table of contacts with columns for 'contact name', 'ราคาทั้งหมด', 'วันกำหนดชำระเงิน', 'edit', and 'cancel'. The 'cancel' button for the first contact is highlighted. Below the table, there is a text box explaining the 'cancel' and 'edit' actions.

| contact name | ราคาทั้งหมด | วันกำหนดชำระเงิน | edit | cancel |
|-----------------------------|-------------|------------------|------|--------|
| งานสมรสคุณเมธา | 12000 | 2 10 2003 0:0:0 | edit | cancel |
| งานเลี้ยงรุ่นสวนกุหลาบ | 6000 | 29 6 2003 0:0:0 | edit | cancel |
| งานฉลองจบการศึกษาห้อง Olala | 420 | 22 3 2003 0:0:0 | edit | cancel |

หากต้องการยกเลิกทั้ง contact ให้กดที่ cancel แต่ถ้าต้องการเข้าไปแก้ไขงาน หรือยกเลิกงาน ให้กดที่ edit เพื่อเข้าไปเลือกดูจากงานทั้งหมดที่มีใน contact นั้น

รูปที่ 7-51 แสดงการเลือกคำสั่ง cancel ของ contact ที่ต้องการเพื่อยกเลิกcontact นั้น

3. ระบบจะแสดงหน้าจอการยกเลิกการสั่งซื้อทั้งคอนแทกเรียบร้อยแล้ว

7.8.6 การเปลี่ยนแปลงข้อมูลการสั่งซื้อสินค้า

1. การเปลี่ยนแปลงข้อมูลการสั่งซื้อสินค้า จะทำได้เมื่อทำการล็อกอินเข้าระบบแล้วเท่านั้น โดยเลือกรายการ “เปลี่ยน/ยกเลิกการจอง” บนเมนูทางด้านซ้ายของหน้าเว็บ ระบบจะแสดงหน้าจอ contact ทั้งหมดที่มีของผู้ใช้นั้น
2. เลือก “edit” ที่ contact ที่ต้องการเปลี่ยนแปลงข้อมูล หรือยกเลิกการจอง หน้าจอจะแสดงรายการจองทุกรายการที่มีอยู่ใน contact นั้น
3. เลือก “edit” ที่รายการจองย่อยที่ต้องการยกเลิกการจอง ระบบจะแสดงหน้าจอรายละเอียดการจองนั้น
4. กด edit เพื่อทำการเปลี่ยนแปลงข้อมูลการจองรายการนั้น

Gift Registries Shop

HOME GIFT Package Member Zone SEARCH CONTACT

ยินดีต้อนรับคุณ เมธา

รายละเอียดการสั่งซื้อสินค้าของคุณ กรุณาเลือก งาน ที่ต้องการยกเลิกหรือเปลี่ยนแปลง

| ชนิด | รายละเอียด | ข้อความ | วันรับของ | จำนวนชิ้น | ราคาทั้งหมด | edit | cancel |
|-----------------|---------------------------|--------------------------------|-----------------|-----------|-------------|------|--------|
| สมุดขนาด เล็ก 1 | สมุดขนาด เล็ก 1 สีฟ้าอ่อน | ที่ระลึกงานเลี้ยงรุ่นสวนกุหลาบ | 9 7 2003 10:0:0 | 500 | 6000 | edit | cancel |

สร้าง contact name
ส่งสินค้า
จ่ายเงิน
ดูรายการที่สั่ง
เปลี่ยน/ยกเลิก การสั่ง
จอง
ออกจากระบบ

รูปที่ 7-52 แสดงการเลือกคำสั่ง *edit* ของรายการสั่งซื้อสินค้าที่ต้องการแก้ไขข้อมูลการสั่งซื้อ

5. กรอกข้อมูลใหม่ที่ต้องการเปลี่ยนแปลง เมื่อกรอกข้อมูลเสร็จแล้ว คลิกที่ปุ่ม “เปลี่ยนแปลง”

Gift Registries Shop

HOME GIFT Package Member Zone SEARCH CONTACT

ยินดีต้อนรับคุณ เมธา

ทำการเปลี่ยนแปลงรายการ : งาน ที่ต้องการรับสินค้า, ข้อความ, ที่อยู่

สมุดขนาด เล็ก 1 สีฟ้าอ่อน

ข้อความ

ที่อยู่

วัน ที่รับสินค้า เดือน ปี

สร้าง contact name
ส่งสินค้า
จ่ายเงิน
ดูรายการที่สั่ง
เปลี่ยน/ยกเลิก การสั่ง
จอง
ออกจากระบบ

รูปที่ 7-53 แสดงหน้าเปลี่ยนแปลงข้อมูลการสั่งซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8.7 การชำระเงินผ่านบัตรเครดิต

1. เมื่อทำการล๊อคอินเข้าระบบ หรือหลังจากกดปุ่ม “เปลี่ยน contact ปัจจุบัน” หน้าจอหลักจะแสดง contact ทั้งหมดของผู้ใช้คนนั้น
2. เลือก contact ที่ต้องการ หน้าจอจะแสดงรายละเอียดของcontactนั้น ได้แก่ค่าบริการรวม ค่าบริการที่ชำระแล้ว ค่าบริการคงเหลือ วันสุดท้ายที่จะชำระเงินเพื่อยืนยันการจองได้ และรายการจองย่อยทุกรายการใน contact นั้น
3. เลือกรายการ “ชำระค่าบริการ” จากเมนูทางซ้ายมือของหน้าเว็บ
4. กรอกประเภทบัตรเครดิต หมายเลขบัตรเครดิต และวันหมดอายุลงในหน้าจอชำระเงินผ่านบัตรเครดิต แล้วกด submit

7.9 คู่มือการใช้งานสตูดิโอถ่ายภาพ

การใช้งานสตูดิโอถ่ายภาพนี้โดยส่วนใหญ่จะมีความคล้ายคลึงกับระบบร้านของชำร่วยดังนั้นในหัวข้อนี้จึงนำเสนอแค่ บางตัวอย่างเพื่อให้เห็นการทำงานของระบบโดยจะไม่ลงถึงทุกขั้นตอนของระบบ



รูปที่ 7-54 แสดงหน้าหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-55 ตัวอย่างบริการที่มีให้ของเว็บ

7.9.1 ส่วนบริการสมาชิก

7.9.1.1 การสมัครสมาชิก

1. คลิกที่ “sign up new user!” บริเวณด้านขวาของหน้าเว็บ ระบบจะแสดงหน้าสำหรับกรอกข้อมูลสมัครสมาชิกใหม่

รูปที่ 7-56 แสดงหน้าการสมัครสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรอกข้อมูล ชื่อ นามสกุล เพศ อายุ หมายเลขโทรศัพท์ ที่อยู่ อีเมลแอดเดรส ตั้งชื่อผู้ใช้และพาสเวิร์ดที่จะใช้ในการล็อกอินเข้าสู่ระบบ แล้วกดปุ่ม “ลงทะเบียน”
3. ระบบจะแสดงหน้าจอลงทะเบียนเรียบร้อย หากจะทำการจองใดๆ ต้องทำการล็อกอินเข้าสู่ระบบก่อน

7.9.1.2 การล็อกอินเข้าสู่ระบบ

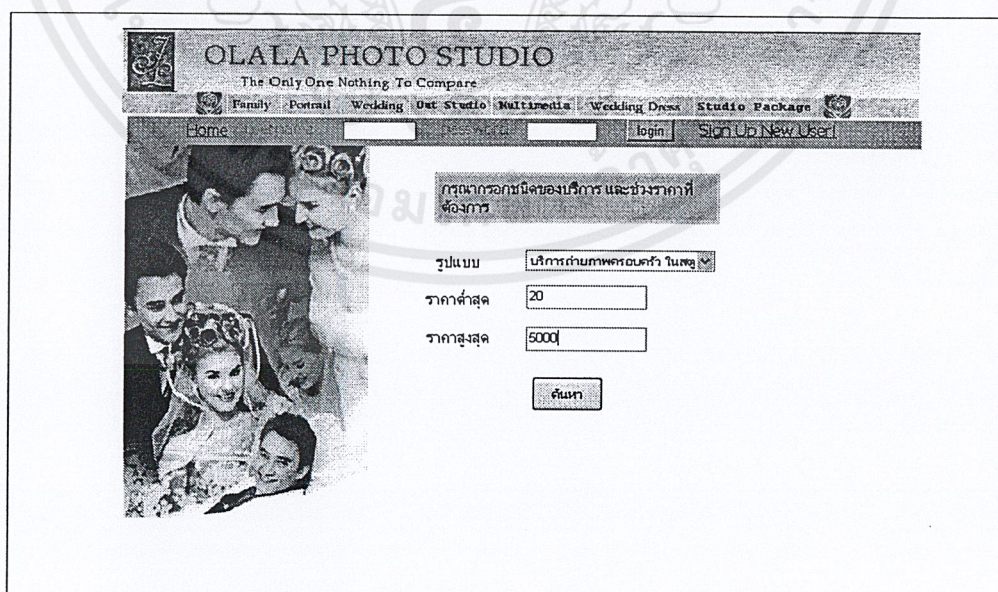
1. กรอกชื่อผู้ใช้และรหัสผ่านที่เมนู sign in ทางด้านซ้ายของหน้าเว็บ



รูปที่ 7-57 แสดงการ login

7.9.2 การค้นหาสินค้าตามเงื่อนไขที่ต้องการ

1. คลิกที่รายการ “search” บนแถบเมนูหลัก ด้านบนของหน้าจอ ระบบจะแสดงหน้าจอให้ค้นหาสินค้า
2. กำหนดเงื่อนไขสินค้าที่ต้องการค้นหา ได้แก่ ชนิดสินค้า รูปแบบสินค้า ราคาต่ำสุด สูงสุดต่อชิ้น



รูปที่ 7-58 แสดงหน้ากรอกรายละเอียดการค้นหาสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กดปุ่ม “ค้นหา” ระบบจะทำการค้นหาสินค้าที่ตรงตามความต้องการ และแสดงผลการค้นหา

หา

OLALA PHOTO STUDIO
The Only One Nothing To Compare

Family Portrait Wedding Out Studio Multimedia Wedding Dress Studio Package

Home login Sign Up New User

ผลการค้นหาตามชนิดของบริการ และช่วงราคาที่ต้องการ

| หมวดบริการ | ชนิด | รายละเอียด | ราคา |
|--------------------------|------------------------------|---|------|
| ถ่ายภาพครอบครัวในสตูดิโอ | ถ่ายภาพครอบครัวในสตูดิโอเล็ก | ถ่ายที่ สตูดิโอ ได้ภาพ 70 ภาพ พร้อมฟิล์ม 70 ใบ พร้อมแต่งหน้าทำผมแนวธรรมชาติ - ภาพ Remix ขนาด 20" x 30" รวมกรอบอลูมิเนียม 1 กรอบ - ภาพขยาย ขนาด 20" x 30" เคลือบผ้าใบ 1 ใบ - ภาพขยาย ขนาด 8" x 10" เคลือบผ้าใบ 10 ใบ พร้อมพร้อมใส่ acid free photoboard | 3500 |
| ถ่ายภาพครอบครัวในสตูดิโอ | ถ่ายภาพครอบครัวในสตูดิโอใหญ่ | ถ่ายที่ สตูดิโอ ได้ภาพ 100 ภาพ พร้อมฟิล์ม 100 ใบ พร้อมแต่งหน้าทำผมแนวธรรมชาติ - ภาพ Remix ขนาด 40" x 0" รวมกรอบอลูมิเนียม 1 กรอบ - ภาพขยาย ขนาด 20" x 30" เคลือบผ้าใบ 1 ใบ - ภาพขยาย ขนาด 8" x 10" เคลือบผ้าใบ 10 ใบ พร้อมพร้อมใส่ acid free photoboard | 3500 |

รูปที่ 7-59 แสดงหน้าผลการค้นหาสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

บทวิจารณ์ และสรุป

8.1 สรุปผลการดำเนินงาน

ในโครงการนี้ได้แบ่งการดำเนินงานออกเป็น 5 ส่วนหลัก คือ

1. ศึกษาทฤษฎีพื้นฐานของ คอมพิวเตอร์ , .NET เฟรมเวิร์ก , ASP.NETเว็บเซิร์ฟเวอร์ และเทคโนโลยีอื่นๆ ที่เกี่ยวข้อง
2. ออกแบบระบบงาน เพื่อทำการทดสอบทฤษฎีต่างๆ ที่ได้ศึกษามา โดยตัวอย่างระบบงานที่สร้างขึ้นมา คือ ระบบการจองแพ็คเกจสตูดิโอถ่ายภาพ,ร้านขายของชำรายและระบบการจองโรงแรม
3. ทำการอิมพลีเมนต์ระบบงานที่ได้ออกแบบไว้ โดยใช้ภาษา C#,VB.NET, HTML พัฒนาตามมาตรฐานของ เว็บเซิร์ฟเวอร์ และอิงตามแพลตฟอร์ม .NET
4. ทำการอิมพลีเมนต์ระบบงานในส่วนที่เรียกใช้ระบบธนาคารจำลองการโอนเงิน ที่พัฒนาจากเทคโนโลยีเว็บเซิร์ฟเวอร์ของ EJB ผ่าน โพรโตคอลมาตรฐานที่เรียกว่า SOAP
5. เพิ่มประสิทธิภาพโดยการขยายการรองรับของระบบให้มากขึ้น โดยการนำเทคโนโลยีของ MSMQ เข้ามาช่วย นอกจากนี้ยังให้บริการต่างๆ ของ COM+ เพื่อเพิ่มความสามารถต่างๆ ให้แก่ระบบงานด้วย

ระบบงานที่ทดลองสร้างขึ้น มีการใช้เทคโนโลยีต่างๆ ที่ได้ศึกษามา โดยได้ทำการสร้างระบบงานเป็น 3 เทียร์ คือ

1. ฟรอนต์เอนด์เทียร์ มีไคลเอ็นต์เป็นเว็บเบราว์เซอร์ที่เรียกใช้บริการจากระบบและแอปพลิเคชันที่ทำหน้าที่เป็นเครื่องมือในการจัดการระบบ
2. บิสิเนสเทียร์ มีเว็บเซิร์ฟเวอร์ IIS และ .NET เฟรมเวิร์กติดตั้ง โดยในส่วนของเว็บแอปพลิเคชันและเว็บเซิร์ฟเวอร์ โดย ASP.NET ที่มีการเรียกใช้บริการของ COM+ ด้วย
3. ดาต้าเทียร์ ใช้ดาตาเบสเซิร์ฟเวอร์เป็น Microsoft SQL Server 2000

8.2 แนวทางการพัฒนาต่อ

ถึงแม้ว่า การทดลองสร้างระบบงานเว็บเซอร์วิสให้สามารถเรียกใช้บริการจากระบบงานอื่นที่พัฒนา มาจากเทคโนโลยีที่แตกต่างกันได้ (คือ .NET และ EJB) โดยผ่านทางโพรโตคอล SOAP นั้น สามารถทำงานได้ เป็นอย่างคึกคักตาม แต่ก็ยังเป็นเพียงการแลกเปลี่ยนข้อมูลและเรียกใช้เซอร์วิสบริการแบบง่ายๆ เพียงแค่รับส่งข้อมูลผ่านกันได้รูปแบบของ XML เท่านั้น แต่การใช้งานความสามารถที่เพิ่มขึ้น เช่น การทำเสตทฟูลเว็บเซอร์วิส สามารถทำได้ภายในแพลตฟอร์มเดียวกัน หรือถ้าเป็นจาวาต้องภายในผลิตภัณฑ์ตัวเดียวกันด้วย การจัดการทรานแซกชันที่โคเลเอนต์ไม่สามารถทำ two phase commit ระหว่างสองเว็บเซอร์วิสได้ รวมไปถึงหัวข้อสำคัญ คือ ความปลอดภัยระหว่างการสื่อสารในขณะนี้ทำโดยนำ โพรโตคอล SSL เข้ามาช่วยแต่ก็ส่งผลให้ความเร็ว ลดลงเป็นอย่างมาก

การนำเว็บเซอร์วิสเพื่อไปใช้นั้น ยังคงต้องมีการพัฒนามาตรฐานในเรื่องต่างๆออกมารองรับอีก มากมาย ดังเช่นเรื่องของทรานแซกชัน ที่คาดว่าจะมีมาตรฐานเปิดตัวออกมาในปี 2547 มาตรฐานทางด้านความปลอดภัยซึ่งขณะนี้ได้ออกมาแล้ว โดยทางไมโครซอฟท์เพิ่งมีปลั๊กอินเข้ามาสนับสนุนการทำงานเมื่อไม่นานมานี้ ทางฝั่งจาวาก็ต้องรอผลิตภัณฑ์ของหลายๆบริษัทที่จะออกมารองรับ ซึ่งมาตรฐานเหล่านี้ก็เป็นสิ่งที่ควรศึกษา และนำมาพัฒนาต่อไป