สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้ EJB 1

Component-Based Software Development base on EJB 1

T049904

นาย พีรยสถ์ ศุภจรูญวงศ์
นาย ไพโรจน์ แซ่เล้า
นางสาว ภัทราภรณ์ อริยปรีชากุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน.....4990.4
วัน,เดือน,ปี.....2 เม.ย. 2547
.b.....
.i.....
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้ EJB 1

Component-Based Software Development base on EJB 1

โดย

นาย พีรยสถ์ ศุภจรูญวงศ์

นาย ไพโรจน์ แซ่เล้า

นางสาว ภัทราภรณ์ อริยปรีชากุล

อาจารย์ที่ปรึกษา

ดร. วรวัฒน์ ลิ้มโภคา

ดร. ชุติเมษฎ์ ศรีนิลทา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้ EJB 1

Component-Based Software Development base on EJB 1

ผู้จัดทำ

1. นายพีรยสถ์ ศุภจรูญวงศ์ รหัสประจำตัว 42010239
2. นายไพโรจน์ แซ่เล้า รหัสประจำตัว 42010250
3. นางสาวภัทราภรณ์ อริยปรีชากุล รหัสประจำตัว 42010255



_____ อาจารย์ที่ปรึกษา

(ดร. วรวัฒน์ ลิ้มโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้ EJB 1

นาย พีรยสถ์ ศุภจรูญวงศ์ 42010239
นาย ไพโรจน์ แซ่เล้า 42010250
นางสาว ภัทราภรณ์ อริยปรีชากุล 42010255
ดร. วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษา
ดร.ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์ (Component-based Software Development) เป็นกระบวนการในการสร้างซอฟต์แวร์ โดยนำคอมโพเน็นต์ที่มีอยู่มาใช้ (reuse) เพื่อช่วยลดเวลาและค่าใช้จ่ายในการพัฒนา รวมไปถึงความน่าเชื่อถือในการใช้งาน

ในปัจจุบันซอฟต์แวร์ที่พัฒนาขึ้นมามักมีปัญหาทางด้านการแก้ไขข้อบกพร่อง การบำรุงรักษา จึงทำให้เกิดแนวความคิดใหม่ ในรูปแบบของเว็บเซอร์วิส เป็นแนวคิดที่มีความสามารถในการรองรับการพัฒนาแอพพลิเคชันเชิงคอมโพเน็นต์ ช่วยให้ซอฟต์แวร์ที่พัฒนาสามารถที่จะเรียกใช้งานข้ามเทคโนโลยีได้ ผ่านทาง SOAP ซึ่งเป็นโปรโตคอลในใช้งานร่วมกัน

ตัวอย่างการพัฒนาระบบงานที่ปริญญานิพนธ์ฉบับนี้นำเสนอคือ ระบบเอ็นเตอร์เทนเมนต์ ซึ่งใช้หลักการพัฒนาเชิงคอมโพเน็นต์ ตามโครงสร้างของ EJB และเว็บเซอร์วิส โดยระบบตรงกลางจะเป็นระบบ รับจัดงานแต่งงาน ซึ่งพัฒนาโดยเทคโนโลยี J2EE เว็บเซอร์วิส ทำการเรียกใช้เซอร์วิสจากระบบเอ็นเตอร์เทนเมนต์ ซึ่งพัฒนาโดย J2EE และเรียกใช้ ระบบทัวร์กับธนาคารที่พัฒนาโดย J2EE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Component-Based Software Development base on EJB 1

Perayos Supajaroonwong
Pairod Saelou
Pattraporn Ariyapreechakul
Dr. Worawat Limpoka Advisor
Dr. Chutimet Srinilta Advisor

ABSTRACT

Component-based software development is a software process that reused software components to reduce time and cost of development.

Recently, software development has a problem such as updating application, maintenance. So the concept of "Web Services" was used. This concept can be supported component based application development, provided ability to call services across platforms using SOAP.

This thesis is to study in developing system by component-based software development Methodology using EJB and Web services. The central application is wedding planner system that consist of many called services such as entertainment agency that are implemented by J2EE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้เสร็จลงได้ คือ ดร.วรวัฒน์ ลิ้มโภคา และ ดร.ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และความช่วยเหลือเสมอมา ซึ่งต้องขอขอบคุณเป็นอย่างมาก

ขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสทางการศึกษาอย่างเต็มที่ และคอยให้กำลังใจเอาใจใส่เสมอมา ในทุกๆด้าน อันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นาย พีรยสถ์ ศุภจรูญวงศ์ 42010239
นาย ไพโรจน์ แซ่เล้า 42010250
นางสาว ภัทราภรณ์ อริยปรีชากุล 42010255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	2
1.3 ขอบเขตของโครงงาน	2
1.4 วิธีการดำเนินงานของโครงงาน	2
บทที่ 2 เว็บเซอร์วิส	3
2.1 XML คืออะไร	4
2.1.1 แนะนำ XML	4
2.1.2 ไวยากรณ์ของ XML	4
2.1.3 XML Element	6
2.1.4 XML Attribute	9
2.1.5 XML Validation	11
2.1.6 แนะนำ DTD	11
2.1.6.1 โครงสร้าง XML	13
2.1.6.2 การกำหนด Element ใน DTD	13
2.1.6.3 การกำหนด Attribute ใน DTD	16
2.1.6.4 การกำหนด Entity ใน DTD	20
2.1.7 XML Namespace	20
2.1.8 XML PCDATA และ CDATA	23
2.1.8.1 PCDATA	23
2.1.8.2 CDATA	24
2.1.9 XML Parser	25
2.1.9.1 DOM	25
2.1.9.2 SAX	27
2.1.10 ประโยชน์ของ XML	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.11 บทสรุปของ XML 29
2.2 SOAP 29
2.2.1 แนะนำ SOAP 29
2.2.2 ส่วนประกอบของ SOAP 29
2.2.2.1 Envelope 32
2.2.2.2 Body 32
2.2.2.3 Header 32
2.2.3 SOAP Fault Element 33
2.2.4 SOAP Encoding 34
2.2.4.1 ตัวอย่าง Struct 35
2.2.4.2 ตัวอย่าง Array 36
2.2.5 SOAP ใน HTTP 38
2.2.6 SOAP ใน RPC 39
2.2.7 SOAP Toolkit 40
2.3 UDDI 41
2.4 WSDL 41
บทที่ 3 เว็บลอจิก เว็บเซอร์วิส 44
3.1 คอนเวอร์เซชั่น 44
3.1.1 การทำงานของคอนเวอร์เซชั่น 44
3.1.1.1 การรักษาสถานะโดยใช้การซีเรียวไลเซชั่น (serialization) 44
3.1.1.2 การจัดการข้อมูลผ่านการระบุแบบไม่ซ้ำ 44
3.1.2 วงจรชีวิตของคอนเวอร์เซชั่น 45
3.2 ทรานแซ็กชัน 45
3.2.1 เหตุผลที่ต้องใช้ทรานแซ็กชัน 45
3.2.2 ประโยชน์ของทรานแซ็กชัน 48
3.2.2.1 ACID Properties 49
3.2.3 ทรานแซ็กชันโมเดล 50
3.2.4 การใช้งานทรานแซ็กชันกับ EJB 51
3.2.4.1 วิธีการกำหนดการจัดการทรานแซ็กชัน 51
3.2.4.2 การจัดการทรานแซ็กชันของคอนเทนเนอร์ 53
3.2.4.3 การจัดการทรานแซ็กชันของบีน 55
3.2.5 Transaction isolation 55
3.2.5.1 ระดับของ Isolation 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 ทรานแซ็กชันแบบกระจาย 57
3.2.6.1 Two-phase commit 58
3.3 Asynchronous and Synchronous 60
3.3.1 การใช้คอลแบ็กเพื่อที่จะระบุเหตุการณ์กับไคลเอ็นต์ 60
3.3.2 การใช้การติดต่อแบบสอบถามแทนที่การใช้คอลแบ็ก 61
บทที่ 4 ความปลอดภัยในการใช้งานเว็บเซอร์วิส 63
4.1 Cipher Suite 63
4.2 Public Key Algorithms 63
4.3 Symmetic Key Algorithms 63
4.4 Message Digest Algorithms 64
4.5 ลายมือชื่อดิจิตอล 65
4.6 SSL Protocol 67
4.7 เอกสารสิทธิ์ 68
4.8 การอิมพลีเมนต์ Security ใน Web Logic 74
4.8.1 Security Realms 74
4.8.2 User 74
4.8.3 Group 74
4.8.4 Access Control Mechanisms 75
4.8.5 Authentication Mechanisms 75
บทที่ 5 เปรียบเทียบ .NET เว็บเซอร์วิส กับ Weblogic เว็บเซอร์วิส 76
5.1 แพล็ตฟอร์มการพัฒนา 76
5.1.1 แพล็ตฟอร์มของ .NET 76
5.1.2 แพล็ตฟอร์มของจาวา 77
5.2 เปรียบเทียบเทคโนโลยีของทั้งสองค่าย 78
5.2.1 Vendor Solution 79
5.2.2 การสนับสนุนระบบเดิม 79
5.2.3 การยอมรับทางการตลาด 79
5.2.4 ภาษาที่รองรับ 79
5.2.5 การสนับสนุนแพล็ตฟอร์มเดิม 80
5.2.6 การทำงานข้ามแพล็ตฟอร์ม (Portability) 80
5.2.7 เครื่องมือที่ใช้ในการพัฒนา 80
5.2.8 การสนับสนุนการใช้งานเว็บเซอร์วิส 81
5.2.9 การขยายการรองรับการใช้งานของระบบ (Scalability) 81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.10 สรุปการเลือกใช้งาน 81
5.3 เปรียบเทียบจาวากับ .NET 81
5.4 การพัฒนาเว็บเซอร์วิสจากทั้งสองแพล็ตฟอร์ม 82
5.4.1 การเรียกใช้งานข้ามแพล็ตฟอร์ม 82
5.4.2 ปัญหาที่พบ 82
บทที่ 6 แอพพลิเคชัน 86
6.1 Overview OLALA Wedding Planner 86
6.2 รายละเอียดของแอพพลิเคชัน 88
6.3 อีอาร์ไดอะแกรม 89
6.4 ยูสเคสไดอะแกรม 90
6.5 ซีเควนซ์ไดอะแกรม 93
6.6 คลาสไดอะแกรม 124
6.7 คอมโพเนนต์ไดอะแกรม 129
6.8 รายละเอียดของฟังก์ชัน 130
6.8.1 ฟังก์ชันของส่วนกลาง(Olala Wedding Planner) 130
6.8.2 ฟังก์ชันของบริษัทเอ็นเตอร์เทนเมนต์ 131
6.8.3 ฟังก์ชันของบริษัททัวร์ 134
6.8.4 ฟังก์ชันของโฟโตสตูดิโอ 138
6.8.5 ฟังก์ชันของร้านของชำร่วย 145
6.8.6 ฟังก์ชันของโรงแรม 148
6.9 คู่มือการใช้งานแอพพลิเคชั่น 158
บทที่ 7 บทวิจารณ์และสรุป 167
7.1 สรุปการดำเนินงาน 167
7.2 แนวทางการพัฒนาต่อ 168
ภาคผนวก ก การหาระยะทางระหว่างเขตในกรุงเทพฯ 169
บรรณานุกรม 172

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 ชนิดของแอตทริบิวต์ใน XML 17

ตารางที่ 2-2 ค่าดีฟอลต์ของแอตทริบิวต์ใน XML 17

ตารางที่ 2-3 เอ็นติตี้อ้างอิงมาตรฐาน 24

ตารางที่ 2-4 XML พาร์เซอร์ 28

ตารางที่ 2-5 SOAP Fault Element 34

ตารางที่ 2-6 ค่าฟอลต์โค้ด 34

ตารางที่ 2-7 SOAP Toolkit 40

ตารางที่ 3-1 ผลลัพธ์ที่เกิดจาก Transaction Attribute 55

ตารางที่ 3-2 ระดับของ Isolation 57

ตารางที่ 5-1 ตารางแสดงการสนับสนุนการทำงานด้านต่างๆของ 2 แพล็ตฟอร์ม 78

ตารางที่ 5-2 สรุปข้อเปรียบเทียบระหว่างสองแพล็ตฟอร์ม 84

ตารางที่ ก-1 แสดงพิกัด (x, y) ของเขตต่างๆในกรุงเทพฯ 142

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้าที่

รูปที่ 2-1 โครงสร้างของ DOM 26

รูปที่ 2-2 ความสัมพันธ์ของการแปลด้วย DOM 27

รูปที่ 2-3 โครงสร้างของ SOAP 30

รูปที่ 3-1 วงจรชีวิตของคอนเวอร์เซชั่น 45

รูปที่ 3-2 แอพพลิเคชันที่ทำงานในระบบแบบกระจาย 47

รูปที่ 3-3 การใช้งานฐานข้อมูลพร้อม ๆ กัน 48

รูปที่ 3-4 Flat Transaction 51

รูปที่ 3-5 Nested Transaction 51

รูปที่ 3-6 bean ที่ทำงานกับ Programmatic Transaction 52

รูปที่ 3-7 bean ที่ทำงานใน Declarative Transaction 53

รูปที่ 3-8 ทรานแซ็กชันแบบกระจายที่มีการใช้โพรโตคอล Two-Phase Commit 59

รูปที่ 4-1 แสดงการใช้คีย์คู่ในการติดต่อ 64

รูปที่ 4-2 แสดงการได้มาซึ่งคีย์เดี่ยวในการติดต่อ 64

รูปที่ 4-3 แสดงการนำ Message Digest ไปใช้ใน public-private key encyption 65

รูปที่ 4-4 แสดงวิธีการทำลายมือชื่อโดยใช้แฮชฟังก์ชัน และคีย์ต่างๆ 66

รูปที่ 4-5 แสดงกรณีที่มีผู้แอบอ้าง 67

รูปที่ 4-6 แสดงทรีเวย์แฮนด์เชค 68

รูปที่ 4-7 แสดงส่วนประกอบต่างๆของเอกสารสิทธิ์ 69

รูปที่ 4-8 แสดงเอกสารสิทธิ์ที่ตรวจสอบโดยบราวเซอร์ 73

รูปที่ 5-1 แพล็ตฟอร์มการพัฒนาระบบภายใต้เทคโนโลยี .NET 76

รูปที่ 5-2 แพล็ตฟอร์มการพัฒนาระบบภายใต้เทคโนโลยี J2EE 77

รูปที่ 6-1 ระบบ OLALA Wedding Planner 87

รูปที่ 6-2 แสดงอีอาร์ไดอะแกรมของระบบ Olala Wedding Planner 89

รูปที่ 6-3 แสดง use case ของแอพพลิเคชันในส่วนของบริษัททัวร์ 90

รูปที่ 6-4 แสดง use case ของแอพพลิเคชันในส่วนของบริษัทของชำร่วย กับ เอ็นเตอร์เทนเมนต์ 90

รูปที่ 6-5 แสดง use case ของแอพพลิเคชันในส่วนของโรงแรม 91

รูปที่ 6-6 แสดง use case ของแอพพลิเคชันในส่วนของโฟโตสตูดิโอ 91

รูปที่ 6-7 แสดง use case ของแอพพลิเคชันในส่วนของ Olala Wedding Planner 92

รูปที่ 6-8 แสดงการค้นหามัลติมีเดีย 93

รูปที่ 6-9 แสดงการค้นหาการถ่ายวีดีโอที่งาน 94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6-10 แสดงการค้นหาการถ่ายภาพที่งานวิวาห์ 94
รูปที่ 6-11 แสดงการค้นหาการถ่ายภาพที่ร้าน 95
รูปที่ 6-12 แสดงการค้นหาการแสดง 96
รูปที่ 6-13 แสดงการค้นหาของชำร่วย 97
รูปที่ 6-14 แสดงการค้นหาเค้ก 98
รูปที่ 6-15 แสดงการค้นหาห้องจัดเลี้ยง 99
รูปที่ 6-16 แสดงการค้นหาแพคเกจดอกไม้ 100
รูปที่ 6-17 แสดงการค้นหาห้องพัก 101
รูปที่ 6-18 แสดงการค้นหาอาหาร 102
รูปที่ 6-19 แสดงการค้นหาแพคเกจทัวร์ 103
รูปที่ 6-20 แสดงการจองการถ่ายวีดีโอที่งาน 103
รูปที่ 6-21 แสดงการจองการแสดง 104
รูปที่ 6-22 แสดงการจองของชำร่วย 104
รูปที่ 6-23 แสดงการจองเค้ก 105
รูปที่ 6-24 แสดงการจองการถ่ายภาพวีดีโอที่งาน 105
รูปที่ 6-25 แสดงการจองการทำมัลติมีเดีย 106
รูปที่ 6-26 แสดงการจองห้องจัดงาน 106
รูปที่ 6-27 แสดงการจองการถ่ายภาพที่ร้าน 107
รูปที่ 6-28 แสดงการจองแพคเกจดอกไม้ 108
รูปที่ 6-29 แสดงการจองแพคเกจทัวร์ 108
รูปที่ 6-30 แสดงการจองห้องพัก 109
รูปที่ 6-31 แสดงการจองอาหาร 109
รูปที่ 6-32 แสดงการยกเลิกแพคเกจดอกไม้ 110
รูปที่ 6-33 แสดงการยกเลิกการจองห้อง 111
รูปที่ 6-34 แสดงการยกเลิกการถ่ายวีดีโอที่งาน 112
รูปที่ 6-35 แสดงการยกเลิกการทำมัลติมีเดีย 113
รูปที่ 6-36 แสดงการยกเลิกการถ่ายภาพวีดีโอที่งาน 114
รูปที่ 6-37 แสดการยกเลิกการแสดง 115
รูปที่ 6-38 แสดงการยกเลิกของชำร่วย 116
รูปที่ 6-39 แสดงการยกเลิกการสั่งจองเค้ก 117
รูปที่ 6-40 แสดงการยกเลิกการถ่ายภาพที่ร้าน 118
รูปที่ 6-41 แสดงการยกเลิกการจองห้องจัดงาน 119
รูปที่ 6-42 แสดงการยกเลิกแพคเกจทัวร์ 120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6-43 แสดงการยกเลิกอาหาร 121
รูปที่ 6-44 แสดงการยืนยันการจองและจ่ายเงิน 122
รูปที่ 6-45 แสดงการลงทะเบียน 122
รูปที่ 6-46 แสดงการเข้าสู่ระบบ 123
รูปที่ 6-47 แสดง คลาสไดอะแกรมของทัวร์ 124
รูปที่ 6-48 แสดงคลาสไดอะแกรมของโรงแรม 125
รูปที่ 6-49 แสดงคลาสไดอะแกรมของบริษัทเอ็นเตอร์เทนเมนต์ 126
รูปที่ 6-50 แสดงคลาสไดอะแกรมของบริษัทของชำร่วย 126
รูปที่ 6-51 แสดงคลาสไดอะแกรมของโฟโตสตูดิโอ 127
รูปที่ 6-52 แสดงคลาสไดอะแกรมของ Olala Wedding Planner 128
รูปที่ 6-53 แสดง คอมโพเน็นต์ไดอะแกรม 129
รูปที่ 6-54 แสดงขั้นตอนการสมัคร 130
รูปที่ 6-55 แสดงหน้าต่างล็อกอิน 130
รูปที่ 6-56 แสดงรายละเอียดสัญญาของผู้ใช้ มีกำหนดวันที่ต้องจ่ายเงิน,ราคา สามารถยกเลิก หรือ สร้างใหม่ได้ 131
รูปที่ 6-57 แสดงรายละเอียดของสัญญา บอกว่าจองอะไรไปบ้าง สามารถค้นหา หรือยกเลิกรายการได้ 131
รูปที่ 6-58 แสดงหน้าต่างค้นหาของร้านโฟโตสตูดิโอ โดยสามารถเลือกความต้องการต่างๆ ได้ 132
รูปที่ 6-59 แสดงผลการค้นหา(ไม่ได้เลือกแบบถูกสุด) 132
รูปที่ 6-60 แสดงหน้าต่างหลังจากกดปุ่มจอง โดยต้องมีการใส่รายละเอียดเพิ่มก่อนที่จะจอง 133
รูปที่ 6-61 แสดงผลลัพธ์หลังในหน้ารายละเอียดหลังจากการจองแล้ว 133
รูปที่ 6-62 เมื่อมีการค้นหาครั้งที่สองจะไม่สามารถเลือกร้านได้ เนื่องจากจะสั่งของได้จากร้านเดียว 134
รูปที่ 6-63 แสดงกรณีที่ต้องการค้นหาแบบถูกสุด 134
รูปที่ 6-64 แสดงผลลัพธ์จากการค้นหาเมื่อเลือกเงื่อนไข “แบบถูกสุด” 135
รูปที่ 6-65 แสดงหน้าต่างค้นหาของบริษัททัวร์ 135
รูปที่ 6-66 แสดงหน้าต่างค้นหาห้องจัดเลี้ยงของโรงแรม สามารถค้นหาแบบใกล้สุด(เฉพาะในกรุงเทพ) หรือ ขอบเขต(ภาค) นอกเหนือจากแบบถูกสุดได้ 136
รูปที่ 6-67 แสดงหน้าต่างค้นหาห้องพักของโรงแรม 136
รูปที่ 6-68 แสดงหน้าต่างค้นหาดอกไม้ของโรงแรม 137
รูปที่ 6-69 แสดงหน้าต่างค้นหาเค้กของโรงแรม 137
รูปที่ 6-70 แสดงหน้าต่างค้นหาอาหารของโรงแรม 138
รูปที่ ก-1 แอพพลิเคชันของ thai map guide 141

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก-2 เมื่อนำเมาส์ไปวางเหนือตำแหน่งใดๆบนแผนที่ จะมีพิกัดบอกอยู่ที่ตำแหน่งล่างซ้าย ของหน้าต่าง 142

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์ (Component-Based Software Development หรือ CBSD) เป็นกระบวนการในการสร้างซอฟต์แวร์ โดยนำคอมโพเน็นต์ที่มีอยู่มาใช้ (reuse) เพื่อลดเวลาและค่าใช้จ่ายในการพัฒนา

และการพัฒนาสถาปัตยกรรมคอมโพเน็นต์บนฝั่งเซิร์ฟเวอร์ (Server-side component architecture) ยังช่วยให้การพัฒนาแอพพลิเคชันระดับเอ็นเตอร์ไพรส์ (Enterprise application) ทำได้ง่าย นอกจากนั้นยังมีความน่าเชื่อถือ (Reliable) , มีระบบรักษาความปลอดภัย (Security) ที่ถูกต้องมากขึ้น ทั้งยังช่วยลดความซับซ้อนเนื่องจากโครงสร้างของระบบ ในปัจจุบันได้มีบริษัทต่างๆ ได้พัฒนาสถาปัตยกรรมที่ทำงานบนฝั่งเซิร์ฟเวอร์ (Server-side architecture) 2 สถาปัตยกรรมที่ได้รับการยอมรับมากที่สุดคือ

- Microsoft's Distributed interNet Applications Architecture (DNA) เป็นสถาปัตยกรรมของบริษัทไมโครซอฟท์ (Microsoft) ที่นำเสนอสถาปัตยกรรมคอมโพเน็นต์บนฝั่งเซิร์ฟเวอร์คือ COM+ (Component Object Model plus)
- Sun Microsystem's Java 2 Platform, Enterprise Edition (J2EE) เป็นสถาปัตยกรรมของบริษัทซัน ไมโครซิสเต็มส์ (Sun Microsystems) ที่นำเสนอสถาปัตยกรรมคอมโพเน็นต์บนฝั่งเซิร์ฟเวอร์คือ EJB (Enterprise JavaBeans)

สถาปัตยกรรมของทั้ง 2 บริษัทมีความแตกต่างที่น่าสนใจคือ J2EE นำเสนอความสามารถในการทำงานข้ามแพล็ตฟอร์ม (portability) และ DNA นำเสนอความสามารถในการทำงานข้ามภาษา (interoperability) เนื่องจากทางบริษัทซันไมโครซิสเต็มส์ได้ออกแบบภาษาจาวา (Java) ให้สามารถทำงานข้ามแพลตฟอร์ม การพัฒนาคอมโพเน็นต์ด้วยภาษาจาวาทำให้คอมโพเน็นต์มีความสามารถนี้อยู่ด้วย ในขณะที่ทางบริษัทไมโครซอฟท์ได้สนับสนุนการสร้างคอมโพเน็นต์จากภาษาใดก็ได้เช่น Visual Basic และ Visual C++ คอมโพเน็นต์ที่สร้างด้วย Visual Basic สามารถนำมาใช้กับ Visual C++ ได้ ความแตกต่างอีกอย่างหนึ่งคือ J2EE เป็น specification เท่านั้น แอพพลิเคชันเซิร์ฟเวอร์ (Application Server) จะถูกพัฒนาโดยผู้ผลิตรายอื่น เช่น BEA WebLogic Server , IBM WebSphere , Oracle 9i Application Server , SilverStream Application Server , iPlanet , Sybase Enterprise Application Server , Borland AppServer เป็นต้น

ทั้ง 2 สถาปัตยกรรมนี้มีข้อจำกัดที่ไม่สามารถนำกลับมาใช้ใหม่ (reuse) ข้ามสถาปัตยกรรมได้ แต่ภายหลังการเกิดของ XML , SOAP และแนวคิดของ Web Service ทำให้สามารถทำงานร่วมกันได้ผ่านทาง SOAP ซึ่งเป็นโพรโตคอลที่ใช้สำหรับ Remote Procedure Call (RPC) ผ่าน HTTP โพรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

ภายในโครงงานนี้ได้ทำการศึกษาถึงทฤษฎี การสร้าง และการนำ เว็บเซอร์วิสไปใช้งานโดยมีจุดประสงค์ดังนี้

1. ศึกษาทฤษฎีพื้นฐานของคอมโพเนนต์
2. เพื่อศึกษาทำความเข้าใจ j2EE ของ Java
3. ศึกษาเรื่องเว็บเซอร์วิส และเทคโนโลยีที่เกี่ยวข้อง ได้แก่ SOAP, XML, WSDL, UDDI เป็นต้น
4. ศึกษาการสร้างเว็บเซอร์วิส โดยใช้ Enterprise Java Bean และ Bea Weblogic 7.0 ลงลึกในเทคโนโลยีของเว็บเซอร์วิส เช่น ทรานแซกชั่น , การจัดการเสตท ,ความปลอดภัยในการเรียกใช้เว็บเซอร์วิส
5. ศึกษาวิธีการใช้งาน Bea Weblogic 7.0 เพื่อใช้สร้าง เว็บเซอร์วิส(Web service)
6. ทำการขยายการรองรับการใช้งานของระบบ (Scalability)โดยใช้เทคโนโลยี JMS
7. ศึกษาการเรียกใช้เว็บเซอร์วิสข้ามค่ายรวมทั้งเปรียบเทียบเทคโนโลยีของทั้งสองค่าย.NET และจาวา

1.3 ขอบเขตของโครงงาน

1. ออกแบบและสร้างแอพพลิเคชั่นโดยใช้สถาปัตยกรรม แบบ 3 เทียร์ ร่วมกับการใช้หลักการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ด้วย Enterprise Java Bean
2. ใช้ทฤษฎีของ JMS เพื่อเข้ามาเพิ่มขนาด (Scalability) ให้กับแอพพลิเคชั่น
3. สร้างแอพพลิเคชั่นทำการเรียกใช้เว็บเซอร์วิสคอมโพเน็นต์ข้ามแพลตฟอร์มได้
4. ระบบที่พัฒนาเป็นระบบอีคอมเมิร์ซที่เกี่ยวกับ การจัดการแต่งงาน (Wedding Planner) โดยใช้ Enterprise Java Bean โดยกลุ่มจะทำเป็นเว็บเซอร์วิสที่เป็นส่วนกลางโดยจะมีการ ติดต่อกับเว็บเซอร์วิสของแต่ละค่ายให้ทำงาน โดยนำผลลัพธ์ที่ได้จากแต่ละเวปเซอร์วิส ซึ่งได้แก่ บริษัททัวร์(Tour),บริษัทเอ็นเตอร์เทนเมนต์(Entertainment),ร้านถ่ายรูป(Photo Studio),ร้านขายของชำร่วย(Gift Shop) และ โรงแรม(Hotel) มาจัดการวางแผนการแต่งงานอีกครั้งหนึ่ง

1.4 วิธีการดำเนินงานของโครงงาน

1. ศึกษาทฤษฎีพื้นฐานของ Enterprise Java Bean
2. ศึกษาทฤษฎีของ XML และ SOAP และส่วนประกอบในเว็บเซอร์วิส
3. ศึกษาการพัฒนาเว็บเซอร์วิสใน Bea Weblogic 7.0
4. ออกแบบระบบแอพพลิเคชั่นในลักษณะงานเชิงคอมโพเนนต์
5. ศึกษาการทำงานของ JMS
6. ศึกษาการปรับแต่งเพื่อให้เว็บเซอร์วิสมีความปลอดภัยในการสื่อสาร
7. ศึกษาการใช้งานร่วมกันรวมทั้งเปรียบเทียบเทคโนโลยีจากทั้งสองค่าย .NET และ จาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
เว็บเซอร์วิส

เว็บเซอร์วิส คือ เว็บแอพพลิเคชันยุคใหม่ที่ประกอบด้วยส่วนย่อยๆมีความสมบูรณ์ในตัวเอง สามารถติดตั้ง ค้นหา เริ่มทำงานได้ผ่านเว็บ เว็บเซอร์วิสสามารถทำอะไรก็ได้ตั้งแต่งานง่ายๆ เช่นดึงข้อมูล จนถึงกระบวนการทางธุรกิจที่ซับซ้อน เมื่อเว็บเซอร์วิสตัวใดตัวหนึ่งเริ่มเว็บเซอร์วิสตัวอื่นก็สามารถรับรู้และเริ่มทำงานได้อีกด้วย

หลายคนอาจจะถามว่าทำไมต้องเป็นเว็บเพราะโดยปกติมีมิดเดิลแวร์อื่นๆมากมายเช่น RMI Jini CORBA DCOM ฯลฯ แม้มิดเดิลแวร์เหล่านี้จะสามารถรองรับได้ แต่ไม่มีตัวใดตัวหนึ่งที่เด่นจริง แต่ในเมื่อเว็บมีจุดเด่นในเรื่องของการให้บริการข้อมูลที่สะดวก ใช้งานง่าย จึงกลายเป็นตัวประสานมิดเดิลแวร์ต่างๆ เข้าด้วยกันซึ่งจะให้คุยกันเองคงยากยิ่ง เว็บได้ทำหน้าที่เป็นตัวกลางให้ มิดเดิลแวร์เหล่านี้สามารถคุยกันได้และมีประสิทธิภาพกว่าวิธีการเดิมๆ มาก

หากเรามองจากกรณีของเอ็นเทียร์(n-tier)แอพพลิเคชัน จะพบว่า เว็บเซอร์วิสคือกลไกในการเข้าถึงบริการที่แต่ละมิดเดิลแวร์ให้บริการ การเข้าถึงจะอาศัย Listener และส่วนประกอบที่ระบุถึงบริการต่างๆ ที่รองรับการทำงาน โดยการทำงานจริงๆ นั้นก็ใช้วิธีการปกติของ มิดเดิลแวร์ นั้นๆ
พื้นฐานของเว็บเซอร์วิส คือ XML กับ HTTP ซึ่งจะพบว่า HTTP ก็เป็นที่รู้จักกันดี และไปได้ทั่วทุกแห่งที่มีอินเทอร์เน็ต ส่วน XML คือภาษาสากลที่สามารถปรับแต่งได้ตามใจชอบ เพื่อให้เกิดกิจกรรมระว่างไคลเอนต์ และบริการ หรือระหว่างส่วนประกอบต่างๆ เบื้องหลังเว็บเซอร์วิสก็คือ ข้อความ XML จะถูกแปลงให้การขอบริการจากมิดเดิลแวร์ และผลที่ได้ก็จะแปลงกลับมาในรูป XML โดยมาตรฐาน XML นี้ทำให้เราไม่ต้องกังวลเรื่องของการเชื่อมโยงอีกต่อไป และโพรโตคอล ที่ส่งก็คือ HTTP นั่นเอง ถ้าท่านเชื่อมโยงกับ HTTP (หรือเว็บ) ได้ ท่านก็ใช้บริการทุกอย่างใด

แต่อย่างไรก็ตามการเข้าถึงและการสั่งงานนั้นยังเป็นเพียงโครงสร้างพื้นฐาน แต่ในความเป็นจริงยังมีอะไรมากกว่านั้น เช่น การค้นหา การทำธุระกรรม ความปลอดภัย การพิสูจน์ตัวตน และอื่นๆ อันเป็นบริการที่ทำให้เป็นบริการพื้นฐานจริงๆ ระบบเพิ่มเติมที่ต้องมีและต้องรักษาความสะดวกและใช้งานง่ายไว้ด้วย พื้นฐานของเว็บเซอร์วิส เต็มรูปแบบคือ XML + HTTP + SOAP + WSDL + UDDI หรือในระดับสูงกว่านั้น แต่ไม่ได้ถือเป็นสิ่งจำเป็นเสมอไปคือต้องเพิ่มเทคโนโลยี XAML, XLANG, XKMS, XFS เป็นต้น สำหรับรายละเอียดในบางหัวข้อมีดังต่อไปนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 XML คืออะไร

2.1.1 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>
```

จะใช้แสดงข้อมูลของหนังสือที่ประกอบด้วยอีลีเมนต์ (element) BOOK , NAME , ISDN และ PAGE ข้อมูลที่อยู่ในรูปเท็กซ์นี้เป็นอิสระต่อซอฟต์แวร์และฮาร์ดแวร์ จึงเหมาะที่จะนำมาใช้ในการแลกเปลี่ยนข้อมูลระหว่างระบบบนอินเทอร์เน็ต การเปลี่ยนข้อมูลให้อยู่ในรูป XML เป็นการลดความซับซ้อน และสามารถสร้างข้อมูลที่ถูกอ่านโดยแอพพลิเคชันใด ๆ ก็ได้ นอกจากนั้นการเพิ่มเติมอีลีเมนต์จะไม่ทำให้แอพพลิเคชันเสียหาย เนื่องจากการอ่านข้อมูลโดย XML พาร์เซอร์ (Parser) จะใช้วิธีแยกข้อมูลจากอีลีเมนต์ที่ต้องการ จาก XML ข้างต้นนี้จะแยกข้อมูลอีลีเมนต์ NAME , ISDN และ PAGE หากเพิ่มอีลีเมนต์ AUTHOR เข้าไป แอพพลิเคชันก็ยังคงดึงข้อมูลในอิลิเมนต์เดิมโดยจะมองข้ามอีลีเมนต์ที่ไม่ได้กำหนดให้แยกข้อมูลออกมาในตอนพัฒนาแอพพลิเคชัน

2.1.2 ไวยากรณ์ของ XML

กฎไวยากรณ์ของ XML นั้นง่ายและเคร่งครัดมาก กฎของมันง่ายต่อการเรียนรู้และง่ายต่อการใช้ ด้วยเหตุนี้การสร้างซอฟต์แวร์ในการอ่านและจัดการกับข้อมูลจึงเป็นเรื่องง่าย ตัวอย่างของเอกสาร XML ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ในบรรทัดแรกคือการประกาศเอกสาร XML ที่บอกถึง XML เวอร์ชัน

บรรทัดต่อมาคือรูทอีลีเมนต์ (root element) ซึ่งคือ <note> อีก 4 บรรทัดถัดมาเป็นการอธิบายถึง 4 อีลีเมนต์ลูก (child element) ของรูท (to , from , heading , body) และบรรทัดสุดท้ายเป็นการกำหนดจุดสิ้นสุดของรูท

กฎของเอกสาร XML มีดังนี้

- **XML อีลีเมนต์ (XML element) ทุกตัวต้องมีการปิดแท็ก**

ใน HTML บางอีลีเมนต์ไม่จำเป็นต้องมีการปิดแท็ก แต่ใน XML ทุกอีลีเมนต์จำเป็นที่จะต้องมีการปิดแท็ก

This is a paragraph	X
This is a paragraph	√

- **XML แท็กมีความแตกต่างระหว่างตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ (case sensitive)**

ใน XML แท็ก <Message> นั้นต่างจาก <message> เนื่องจาก XML นั้นมีคุณสมบัติที่แยกความแตกต่างระหว่างตัวพิมพ์เล็กและตัวพิมพ์ใหญ่

<Message>This is incorrect</message>	X
<message>This is correct</message>	√

- **XML อีลีเมนต์ทุกตัวต้องมีการซ้อนกันอย่างเหมาะสม**

ใน HTML บางอีลีเมนต์สามารถเหลื่อมกันได้ แต่ใน XML นั้นต้องซ้อนกันเป็นขั้นโดยสมบูรณ์

This text is bold and italic	X
This text is bold and italic	√

- **ทุกเอกสาร XML ต้องมีรูทแท็ก**

แท็กแรกในเอกสาร XML นั้นจะเป็นรูทแท็กและจะมีได้เพียงหนึ่งเดียวเท่านั้น อีลีเมนต์อื่น ๆ ทั้งหมดจะต้องถูกซ้อนอยู่ภายในรูทอีลีเมนต์เท่านั้น และอีลีเมนต์เหล่านั้นก็สามารถมีอีลีเมนต์ย่อย ได้อีก ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

- **ค่าของแอตทริบิวต์ (Attribute) ต้องมีอยู่ภายในเครื่องหมายคำพูด (“”)**

XML อีลีเมนต์สามารถมีแอตทริบิวต์ได้ โดยที่ค่าของแอตทริบิวต์จะต้องอยู่ภายในเครื่องหมายคำพูด (“”) เอกสาร XML ข้างล่างนี้ อันแรกผิด อันที่สองถูก

```
<note date=12/11/99>          X
<note date="12/11/99">        √
```

2.1.3 XML อีลีเมนต์

XML อีลีเมนต์มีคุณสมบัติดังนี้

- **XML อีลีเมนต์สามารถขยายได้**

ตามตัวอย่างนี้

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

ถ้าเราจะสร้างแอพพลิเคชันที่สามารถแยกอีลีเมนต์ 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 แอพพลิเคชันที่สร้างไว้จะไม่เสียหาย เนื่องจากแอพพลิเคชันจะยังคงค้นหาอีลีเมนต์ to , from และ body ในเอกสาร XML และยังคงให้ผลลัพธ์เหมือนเดิม

- **XML อีลีเมนต์มีความสัมพันธ์กัน**

ในการที่จะเข้าใจถึงภาพรวมของ XML จะต้องเข้าใจถึงความสัมพันธ์ระหว่างชื่อของ XML อีลีเมนต์ที่ตั้งและคอนเทนต์ (content) ของอีลีเมนต์

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>
```

เอกสารนี้มีไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า 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 เป็นรูทอีลีเมนต์ title และ chapter เป็นอีลีเมนต์ย่อยของ book book เป็นอีลีเมนต์หลักของทั้ง title และ chapter title และ chapter มีความเกี่ยวข้องกัน (sibling) เนื่องจากมีอีลีเมนต์หลักเดียวกัน

- **อีลีเมนต์มีคอนเทนต์**

XML อีลีเมนต์คือทั้งหมดตั้งแต่เปิดแท็กจนถึงปิดแท็ก อีลีเมนต์หนึ่งสามารถมีอีลีเมนต์คอนเทนต์ (element content) , มิกซ์คอนเทนต์ (mixed content) , ซิมเปิ้ลคอนเทนต์ (simple content) หรือ คอนเทนต์ว่างเปล่า (empty content) และยังมี แอตทริบิวต์ได้ด้วย

จากตัวอย่างข้างบน book มีอีลีเมนต์คอนเทนต์เพราะมันประกอบด้วยอีลีเมนต์อื่น chapter เป็นมิกซ์คอนเทนต์เพราะมันประกอบด้วยเท็กซ์และอีลีเมนต์อื่น ๆ para เป็นซิมเปิ้ลคอนเทนต์ หรือเท็กซ์คอนเทนต์ (text content) เพราะมันประกอบด้วยเท็กซ์เท่านั้น prod เป็นเอ็มตี้อีลีเมนต์เนื่องจากมันไม่มีข้อมูลใดๆ และเฉพาะ prod อีลีเมนต์เท่านั้นที่มีแอตทริบิวต์ แอตทริบิวต์ที่ชื่อ id มีค่าเท่ากับ “33-657” แอตทริบิวต์ที่ชื่อ media มีค่าเท่ากับ “paper”

- **การตั้งชื่ออีลีเมนต์**

XML อีลีเมนต์มีกฎการตั้งชื่อดังนี้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 XML แอตทริบิวต์

XML แอตทริบิวต์มีคุณสมบัติดังนี้

- **รูปแบบของโค้ท (Quote) , “female” or ‘female’**

ค่าของแอตทริบิวต์ต้องอยู่ในเครื่องหมายคำพูดเสมอ แต่อาจเป็น ‘ ซิงเกิ้ลโค้ท (single quote) หรือ “ อัญประกาศ(double quote) ก็ได้

```
<person sex=“female”>          √
หรือ
<person sex=‘female’>          √
```

อัญประกาศจะเป็นที่ใช้โดยทั่วไปมากกว่า แต่ในบางครั้ง (ถ้าค่าของแอตทริบิวต์จำเป็นที่จะต้องมีโค้ท) จึงจำเป็นที่จะต้องใช้ซิงเกิ้ลโค้ทดังนี้

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

- **การใช้อีลีเมนต์และแอตทริบิวต์**

ข้อมูลสามารถที่จะถูกเก็บในอีลีเมนต์ย่อยหรือแอตทริบิวต์ ดังนี้

```
<person sex="female">
   <firstname>Anna</firstname>
   <lastname>Smith</lastname>
</person>
```

หรือ

```
<person>
   <sex>female</sex>
   <firstname>Anna</firstname>
   <lastname>Smith</lastname>
</person>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างแรก sex เป็นแอตทริบิวต์ ตัวอย่างที่ 2 sex เป็นอีลีเมนต์ย่อยซึ่งทั้งสองตัวอย่างให้ความหมายเดียวกัน มันไม่มีกฎว่าจะใช้แอตทริบิวต์หรืออีลีเมนต์ย่อยเมื่อไร แต่ควรจะหลีกเลี่ยงการใช้แอตทริบิวต์ ควรจะใช้อีลีเมนต์ย่อย ถ้าอินฟอร์เมชันนั้นดูเหมือนจะเป็นข้อมูล (data)

- **หลีกเลี่ยงการใช้ แอตทริบิวต์**

  บางปัญหาของการใช้แอตทริบิวต์

  - แอตทริบิวต์ไม่สามารถมีหลายค่า (อีลีเมนต์ย่อยสามารถมีได้)
  - แอตทริบิวต์ไม่ง่ายที่จะขยาย (ในอนาคต)
  - แอตทริบิวต์ไม่สามารถอธิบายโครงสร้าง (อีลีเมนต์ย่อยสามารถทำได้)
  - แอตทริบิวต์ยากที่จะจัดการด้วยโค้ดของโปรแกรม
  - ค่าของแอตทริบิวต์ไม่ง่ายในการจะตรวจสอบกับ DTD ที่มีอยู่

แต่อย่างไรก็ตาม จะสามารถใช้ แอตทริบิวต์ก็ต่อเมื่อ อินฟอร์เมชันนั้นไม่สัมพันธ์กับข้อมูล ตัวอย่างเช่น ในบางครั้ง ID อ้างอิง (reference) สามารถถูกใช้ในการเข้าถึง XML อีลีเมนต์

```
<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 ในตัวอย่างนี้เป็นตัวนับ (counter) หรือค่าที่ไม่ซ้ำกัน (unique identifier) ในการอ้างถึง note ที่ต่างกันของไฟล์ XML ไม่ใช่ส่วนหนึ่งของข้อมูล note

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 ความถูกต้อง XML (XML Validation)

เอกสาร XML แบ่งการตรวจสอบความถูกต้องของเอกสารเป็น 2 ชนิดคือ

1) **"Well Formed" XML Document**

"Well Formed" XML document เป็นเอกสารที่เป็นไปตามกฎไวยากรณ์ XML ที่ได้กล่าวมาข้างต้น

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 จะต้องมี element NAME เสมอ อาจจะมี element PAGE หรือไม่ก็ได้ เป็นต้น

- **โครงสร้างของ XML (XML Schema)**

โครงสร้างของ XML มีวัตถุประสงค์เหมือนกับ DTD จะถูกนำมาใช้แทน DTD เนื่องจาก

- ง่ายในการเรียนรู้มากกว่า DTD
- ใช้ได้ดีกว่า DTD
- โครงสร้าง (schema) เขียนด้วย XML
- รองรับดาต้าไทป์ (datatype)
- รองรับเนมสเปซ

2.1.6 แนะนำ DTD

1) **การประกาศ DOCTYPE ภายใน**

ถ้า DTD ถูกรวมอยู่ในไฟล์ XML นั้น มันจะต้องอยู่ภายในการให้คำจำกัดความ DOCTYPE ตามไวยากรณ์ นี้

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

ตัวอย่างดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    <!ELEMENT from    (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body    (#PCDATA)>
]>
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend</body>
</note>
```

DTD ข้างบนนี้สามารถแปลความหมายได้ดังนี้

!DOCTYPE note (บรรทัดที่ 2) กำหนดว่าเป็นเอกสารของไทป์ (type) note

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

!ELEMENT to (บรรทัดที่ 4) กำหนดว่า to อีลีเมนต์เป็นข้อมูลชนิด #PCDATA

!ELEMENT from (บรรทัดที่ 5) กำหนดว่า from อีลีเมนต์เป็นข้อมูลชนิด #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>
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ของ DTD ชื่อ note.dtd เป็นดังนี้

note.dtd

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

2.1.6.1 โครงสร้างของ XML

เอกสาร XML จะถูกสร้างจากส่วนต่างๆ ดังนี้

- อีลีเมนต์ – เป็นโครงสร้างหลักของ XML
- แท็ก - ใช้ในการกำหนดมาร์กอัพอีลีเมนต์ (markup element)
- แอตทริบิวต์ - ให้ข้อมูลเพิ่มเติมกับอีลีเมนต์
- เอ็นติตี้ (entity) – เป็นตัวแปรที่ใช้ในการกำหนดเท็กซ์ทั่ว ๆ ไป เอ็นติตี้อ้างอิง (entity reference) เป็นการอ้างถึงเอ็นติตี้ เช่น เอ็นติตี้ชื่อ domain มีค่าเท่ากับ www.w3.org entity reference คือ &domain หากภายในเอกสาร XML มี &domain ค่าของมันจะมีค่าเท่ากับ www.w3.org
- PCDATA -หมายถึง parse character data PCDATA เป็นเท็กซ์ที่จะถูกแปล (parse) โดยพาร์เซอร์
- CDATA –หมายถึง character data เป็นเท็กซ์ที่จะไม่ถูกแปลโดยพาร์เซอร์

2.1.6.2 การกำหนดอีลีเมนต์ใน DTD

วิธีการในการประกาศอีลีเมนต์จะเป็นดังนี้

- **การประกาศอีลีเมนต์**

ประกาศตามไวยากรณ์นี้

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

หรือ

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

- **อีลีเมนต์ว่างเปล่า (Empty)**

ถูกประกาศโดยคีย์เวิร์ด EMPTY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

DTD example:

<!ELEMENT br EMPTY>

XML example:

- **อีลีเมนต์ที่เป็นข้อมูลชนิดตัวอักษร**

  อีลีเมนต์ที่เป็นตัวอักษรเท่านั้นจะถูกประกาศด้วยคีย์เวิร์ด PCDATA ที่เปิดและปิดด้วย "(" ")"

<!ELEMENT element-name (#PCDATA)>

DTD example:

<!ELEMENT note (#PCDATA)>

- **อีลีเมนต์ที่เป็นข้อมูลใด ๆ**

  ประกาศกับคีย์เวิร์ด ANY คอนเท็นต์จะสามารถเป็นข้อมูลชนิดใดก็ได้

<!ELEMENT element-name ANY>

DTD example:

<!ELEMENT note ANY>

- **อีลีเมนต์และอีลีเมนต์ลูก (Sequences)**

  ประกาศโดยใช้ "," ในการแยกอีลิเมนต์ลูกแล้วเปิดและปิดด้วย "(" ")"

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

หรือ

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

DTD example:

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

ถ้ามีการประกาศอีลีเมนต์ลูกจะต้องประกาศอีลีเมนต์ลูกให้สมบูรณ์ด้วยดังตัวอย่างข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

- **การประกาศเพียงครั้งเดียวของอีลีเมนต์เดียวกัน**

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

DTD example:

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

จากตัวอย่างหมายความว่าอีลีเมนต์ note จะประกอบด้วยอีลีเมนต์ลูก message 1 ค่าเท่านั้น

- **การประกาศอย่างน้อย 1 ครั้งของอีลีเมนต์เดียวกัน**

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

DTD example:

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

เครื่องหมาย + ถูกใช้ในความหมายหนึ่งหรือมากกว่า จึงหมายความว่าอีลีเมนต์ note จะประกอบด้วยอีลีเมนต์ลูก message ตั้งแต่ 1 ค่าขึ้นไป

- **การไม่ประกาศหรือประกาศเท่าไรก็ได้ของอีลีเมนต์เดียวกัน**

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

DTD example:

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

เครื่องหมาย * ถูกใช้ในความหมายว่าไม่มีหรือมีเท่าไรก็ได้ จึงหมายความว่าอีลีเมนต์ note จะประกอบด้วยอีลีเมนต์ลูก message กี่ค่าก็ได้ หรือไม่มีเลยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การไม่ประกาศหรือประกาศเพียงครั้งเดียวของอีลีเมนต์เดียวกัน**

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

DTD example:

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

เครื่องหมาย ? ถูกใช้ในความหมายไม่มีหรือมีเพียงหนึ่ง จึงหมายความว่าอีลีเมนต์ note จะประกอบด้วยอีลีเมนต์ลูก message หนึ่งค่า หรือไม่มีก็ได้

- **การประกาศคอนเท็นต์แบบให้เลือก**

  จะใช้เครื่องหมาย | แทนความหมายหรือ

DTD example:

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

ดังนั้นตัวอย่างนี้จึงหมายถึงอีลีเมนต์ note จะต้องประกอบด้วยอีลีเมนต์ to, อีลีเมนต์ from, อีลีเมนต์ header และอีลีเมนต์ message หรืออีลีเมนต์ body อย่างใดอย่างหนึ่งเท่านั้น

- **การประกาศคอนเท็นต์แบบผสม**

  เป็นการผสมการประกาศหลาย ๆ อย่างเข้าด้วยกัน

DTD example:

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

ตัวอย่างนี้หมายถึงอีลีเมนต์ note จะประกอบด้วยตัวอักษรที่ไม่ถูกแปลโดยพาร์เซอร์ หรืออีลีเมนต์ to , from , header , และ message จำนวนเท่าไรก็ได้

2.1.6.3 การกำหนดแอตทริบิวต์ใน DTD

วิธีการประกาศแอตทริบิวต์จะเป็นดังนี้

- **การประกาศแอตทริบิวต์**

  แอตทริบิวต์ต้องถูกประกาศตามไวยากรณ์ นี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

DTD example:

<!ATTLIST payment type CDATA "check">

XML example:

<payment type="check">

ชนิดของแอตทริบิวต์สามารถมีค่าได้ดังตารางที่ 2-1

ค่า	คำอธิบาย
CDATA	ค่าต้องเป็นตัวอักษรที่จะไม่ถูกแปลโดยพาร์เซอร์
(en1|en2|..)	ค่าต้องเป็นอันใดอันหนึ่งในรายการนั้น
ID	ค่าต้องเป็น id ที่ไม่ซ้ำกัน
IDREF	ค่าต้องเป็น id ของอีลีเมนต์อื่น
IDREFS	ค่าต้องเป็นรายการของ id อื่น โดยจะเว้นแต่ละรายการด้วยช่องว่าง
NMTOKEN	ค่าต้องเป็นชื่อที่ประกอบด้วยตัวอักษร , ตัวเลข , เครื่องหมายจุด (.) , เครื่องหมายขีดกลาง (-) , เครื่องหมายขีดล่าง (_) และสามารถมีเครื่องหมายโคลอน (:) ยกเว้นในตำแหน่งแรกได้
NMTOKENS	ค่าต้องเป็นรายการของ NMTOKEN โดยจะเว้นแต่ละรายการด้วยช่องว่าง
ENTITY	ค่าต้องเป็นเอ็นติตี้
ENTITIES	ค่าต้องเป็นรายการของเอ็นติตี้
NOTATION	ค่าต้องเป็นชื่อของ notation (notation คือชนิดของข้อมูลภายนอกที่กำหนดขึ้นเอง เช่น GIF เพื่อบอกว่าเป็นไฟล์รูปภาพ)
XML:	ค่าต้องขึ้นต้นด้วย XML

ตารางที่ 2-1 ชนิดของ แอตทริบิวต์ใน XML

ค่าดีฟอลต์ (default-value) สามารถมีค่าได้ดังตารางที่ 2-2

ค่า	คำอธิบาย
#DEFAULT value	ค่าดีฟอลต์ของแอตทริบิวต์
#REQUIRED	อีลีเมนต์ต้องมีแอตทริบิวต์

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

49904

#IMPLIED	อีลีเมนต์จะมีแอตทริบิวต์หรือไม่ก็ได้
#FIXED value	แอตทริบิวต์ต้องเหมือนกับที่กำหนดไว้

ตารางที่ 2-2 ค่าดีฟอลต์ของ แอตทริบิวต์ใน XML

- **ตัวอย่างของการประกาศ แอตทริบิวต์**

DTD example:

<!ELEMENT square EMPTY>

<!ATTLIST square width CDATA "0">

XML example:

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

ตัวอย่างนี้หมายความว่าสแควร์อีลีเมนต์ (square element) เป็นอีลีเมนต์ว่างกับแอตทริบิวต์ width ซึ่งเป็นชนิดซีดาต้า ถ้าไม่ระบุจะยึดค่าดีฟอลต์เป็น 0

- **ค่าของ แอตทริบิวต์แบบดีฟอลต์**

<!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

- **แอตทริบิวต์โดยนัย**

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

DTD example:

<!ATTLIST contact fax CDATA #IMPLIED>

XML example:

<contact fax="555-667788">

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะใช้ #IMPLIED ในกรณีจะระบุแอตทริบิวต์หรือไม่ก็ได้ ถ้าไม่ระบุก็จะไม่มีค่าดีฟอลต์ให้

- **แอตทริบิวต์ที่ต้องระบุเสมอ**

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

DTD example:

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

XML example:

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

ใช้ในการบังคับให้ต้องระบุแอตทริบิวต์ถ้าในเอกสาร XML ไม่ระบุ แอตทริบิวต์เอกสารนั้นจะไม่ถูกต้อง

- **ค่าของแอตทริบิวต์คงที่**

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

DTD example:

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

XML example:

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

ในแบบนี้ถ้าไม่ได้ระบุค่าแอตทริบิวต์จะใช้ค่าดีฟอลต์ แต่ถ้าจะระบุต้องระบุให้ตรงกับค่าดีฟอลต์มิฉะนั้นจะถือว่าผิด

- **ค่าของแอตทริบิวต์กำหนดเอง**

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

DTD example:

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

XML example:

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

หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

จะใช้เมื่อเราต้องการกำหนดค่าของแอตทริบิวต์เอง

2.1.6.4 การกำหนดเอ็นติตี้ใน DTD

- **การประกาศเอ็นติตี้ภายใน**

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

DTD Example:

```
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">
```

XML example:

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

- **การประกาศเอ็นติตี้ภายนอก**

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

DTD Example:

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

XML example:

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

2.1.7 XML เนมสเปซ

XML เนมสเปซจะหาวิธีหลีกเลี่ยงความสับสนที่เกิดจากชื่ออิลิเมนต์ (Name Conflicts)

- **ความสับสนที่เกิดจากชื่อ**

ความสับสนที่เกิดจากชื่ออีลีเมนต์จะเกิดขึ้นบ่อยเมื่อเอกสารที่แตกต่างกันใช้ชื่อเดียวกันในการอธิบายชนิดของอีลีเมนต์ที่แตกต่างกัน เช่น ตารางใน 2 ตัวอย่างต่อไปนี้

```
<table>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  <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 มีอีลีเมนต์ (table)ซึ่งมีใช้ในความหมายที่ต่างกัน

- **การแก้ปัญหาความสับสนที่เกิดจากชื่อโดยใช้คำเติมหน้า (Prefix)**

  เอกสาร XML ทั้ง 2 คือ

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

ความสับสนที่เกิดจากชื่อจะถูกกำจัดไปโดยการเพิ่มคำเติมหน้าทำให้ได้อีลีเมนต์ที่ต่างกัน 2 ชนิด (<h:table> และ <f:table>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การใช้เนมสเปซ**

จากเอกสาร XML ทั้ง2 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>
```

เราสามารถเพิ่มแอตทริบิวต์ xmlns ลงในแท็ก เพื่อบอกชื่อที่ระบุไว้ ที่เกี่ยวข้องกับเนมสเปซ

- **แอตทริบิวต์เนมสเปซ**

แอตทริบิวต์เนมสเปซจะวางในแท็กเริ่มต้น (start tag) ตามไวยากรณ์ ดังนี้

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

จากตัวอย่างข้างบนเนมสเปซจะสามารถกำหนดด้วยที่อยู่ในอินเตอร์เน็ต (Internet address)

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

จากเนมสเปซเจาะจง (namespace specification) ของ W3C กล่าวไว้ว่าเนมสเปซสามารถเป็น Uniform Resource Identifier (URI)

ถ้าเนมสเปซถูกกำหนดในแท็กเริ่มต้นแล้วอีลีเมนต์ลูกทุกตัวที่มีคำเติมหน้าเหมือนกันจะต้อง มีความสัมพันธ์กับเนมสเปซเดียวกันด้วย

และที่อยู่ที่ใช้บอกเนมสเปซใช้เพื่อให้ชื่อมีความเป็นเอกลักษณ์ไม่ได้ใช้เป็นที่อยู่เพื่ออ้างอิงหาข้อมูล แต่หลายบริษัทใช้เนมสเปซเพื่อเป็นตัวชี้ไปยังหน้าเว็บที่มีอยู่จริงซึ่งเก็บข้อมูลเกี่ยวกับเนมสเปซไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ดีฟอลต์เนมสเปซ (Default Namespaces)**

การกำหนดดีฟอลต์เนมสเปซจะป้องกันการใช้คำเติมหน้าในอีลีเมนต์ลูกทุกตัว มีไวยากรณ์ดังนี้

```
<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>
```

2.1.8 XML PCDATA และ CDATA

Parsed Character Data (PCDATA) คือข้อมูลแบบเท็กซ์ที่จะถูกแปลโดยพาร์เซอร์

Character Data (CDATA) คือข้อมูลแบบเท็กซ์ที่ไม่ถูก parse โดยพาร์เซอร์

2.1.8.1 PCDATA

- **XML พาร์เซอร์จะมองทุกเท็กซ์เป็น Parsed Characters (PCDATA)**

เมื่อ XML ถูกแปลเท็กซ์ที่อยู่ระหว่าง XML แท็กจะถูกแปลไปด้วย

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

พาร์เซอร์ทำเช่นนี้เนื่องจาก XML อีลีเมนต์สามารถมีหลายอีลีเมนต์อยู่ภายใน ดังในตัวอย่าง

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

และพาร์เซอร์จะแบ่งเป็นซับอีลีเมนต์ (sub-element) ดังนี้

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

- **ตัวอักษรเอสเคป (Escape Character)**

ตัวอักษร XML ที่ไม่สามารถใช้ได้จะถูกแทนที่ด้วยเอ็นติตี้อ้างอิง

ถ้าใส่ตัวอักษร "<" ใน XML อีลีเมนต์จะทำให้เกิดความผิดพลาด เพราะพาร์เซอร์จะแปลเป็นจุดเริ่มต้นของอีลีเมนต์ใหม่ ดังนั้นจึงไม่สามารถเขียน อย่างนี้ได้

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

การหลีกเลี่ยง สามารถทำได้โดย แทนที่ "<" ด้วยเอ็นติตี้อ้างอิงดังนี้

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

เอ็นติตี้อ้างอิงมาตรฐานที่ได้กำหนดมาให้ก่อนแล้วใน XML มีทั้งหมด 5 ตัว ดังตารางที่ 2-3

Entity Reference	ค่า	ความหมาย
<	<	น้อยกว่า
>	>	มากกว่า
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

ตารางที่ 2-3 เอ็นติตี้อ้างอิงมาตรฐาน

เอ็นติตี้อ้างอิงจะขึ้นต้นด้วย & และลงท้ายด้วย ;

เฉพาะ "<" และ "&" เท่านั้นที่ไม่สามารถใช้ได้ใน XML ส่วนตัวที่เหลือควรทำให้เป็นนิสัย

2.1.8.2 CDATA

พาร์เซอร์จะไม่แปลทุกอย่างที่อยู่ในส่วนของ CDATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเท็กซ์มี “<” หรือ “&” อยู่ภายใน XML element นั้นจะถูกกำหนดเป็นส่วนของ 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 จะถูกเพิกเฉยโดยพาร์เซอร์

2.1.9 XML พาร์เซอร์

XML พาร์เซอร์คือโปรแกรมที่ใช้ในการอ่าน สร้าง และจัดการกับเอกสาร XML รวมถึงตรวจความถูกต้องของเอกสาร XML พาร์เซอร์สามารถเขียนได้ด้วยภาษาใด ๆ ก็ได้ ในบราวเซอร์ IE 5.0 ขึ้นไปก็มี Microsoft XML parser ทำให้ IE สามารถแสดงผลเอกสาร XML ได้ XML พาร์เซอร์ของภาษาจาวาอย่างเช่น XML พาร์เซอร์ของซัน (http://java.sun.com/xml) หากเราต้องการสร้างแอพพลิเคชันที่สามารถจัดการกับ XML เราก็ต้องมีพาร์เซอร์ของภาษานั้นและเขียนโค้ดติดต่อกับ API ของมัน

XML พาร์เซอร์มี 2 ชนิดคือ

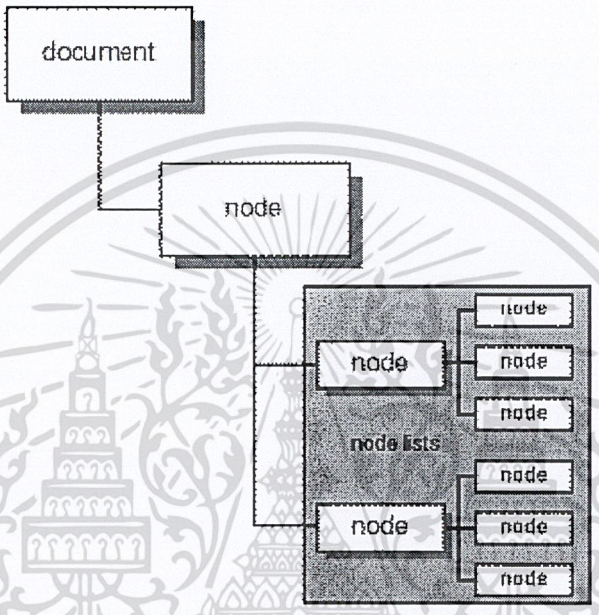
- DOM (Document Object Model)
- SAX (Simple API for XML)

2.1.9.1 DOM

DOM จะใช้วิธีแปลเอกสาร XML เป็นลำดับชั้น (hierachy) ของออบเจ็กต์ดังนี้ (ดังรูปที่ 2-1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เอกสารอ็อบเจ็ค (Document Object) – แหล่งข้อมูล XML
2) โหนดอ็อบเจ็ค (Node Object) – โหนดพ่อ(Parent node) ของโหนดลูก (child node)
3) โหนดลิสต์อ็อบเจ็ค (Node List Object) – ลิสต์ของ sibling node
4) แปลผิดพลาด (Parse Error) – ใช้ในการรับค่าผิดพลาดที่เกิดจากการแปล



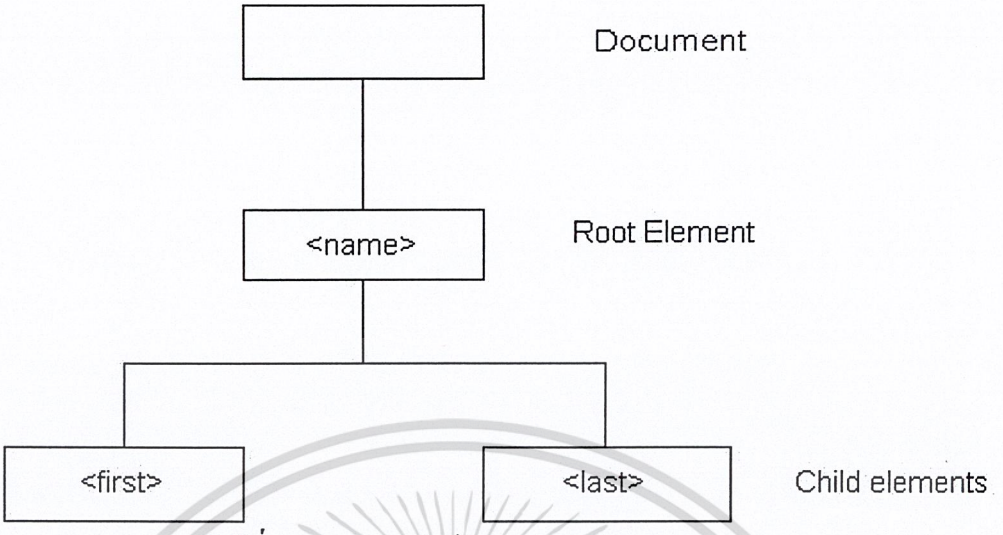
รูปที่ 2-1 โครงสร้างของ DOM

หากเอกสาร XML เป็นดังนี้

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

จะได้ความสัมพันธ์ดังรูปที่ 2-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 ความสัมพันธ์ของการแปลด้วย DOM

2.1.9.2 SAX

SAX (Simple API for XML) วิธีนี้จะเป็นแบบตอบสนองเหตุการณ์ event-driven โดยเหตุการณ์จะประกอบด้วย

1) อีลีเมนต์เริ่มต้น (Start Element) – จะตอบสนองกับเหตุการณ์นี้เมื่อพบจุดเริ่มต้นของแท็ก XML
2) อีลีเมนต์สิ้นสุด (End element) – จะตอบสนองกับเหตุการณ์นี้เมื่อพบจุดสิ้นสุดของแท็ก
3) ตัวอักษร (Character) – จะตอบสนองกับเหตุการณ์นี้เมื่อพบกับตัวอักษร
4) เริ่มต้นเอกสาร (Start Document) – จะตอบสนองกับเหตุการณ์นี้ในตอนเริ่มแปล
5) สิ้นสุดเอกสาร (End Document) - จะตอบสนองกับเหตุการณ์นี้ในเมื่อจบการแปล

ดังนั้นจากโค้ดข้างบนจะเกิดเหตุการณ์ดังนี้

	→	เริ่มต้นเอกสาร
<name>	→	อีลีเมนต์เริ่มต้น name
<first>	→	อีลีเมนต์เริ่มต้น first
Jim	→	ตัวอักษร Jim
</first>	→	อีลีเมนต์สิ้นสุด first
<last>	→	อีลีเมนต์เริ่มต้น last
Jones	→	ตัวอักษร Jones
</last>	→	อีลีเมนต์สิ้นสุด last
</name>	→	อีลีเมนต์สิ้นสุด name
	→	สิ้นสุดเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้ง 2 วิธีมีข้อดีข้อเสียต่างกันคือ DOM จะใช้เมโมรีและมีการประมวลผลมาก แต่สามารถจัดการได้ง่าย สามารถที่จะแก้ไขค่าได้ ไม่เหมาะกับเอกสาร XML ที่มีขนาดใหญ่ ถูกนำไปใช้ในบราวเซอร์ ในขณะที่ SAX ไม่ต้องแปลเอกสาร XML ทั้งหมดจึงใช้โมเมรีและมีการประมวลผลน้อยกว่า เหมาะกับการสร้างและรับค่าจากเอกสาร จะถูกใช้ในการประมวลผลทางฝั่งเซิร์ฟเวอร์ ตัวอย่าง XML พาร์เซอร์ดังตารางที่ 2-4

ชื่อ	แพล็ตฟอร์ม	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 2-4 XML พาร์เซอร์

2.1.10 ประโยชน์ของ XML

สำหรับประโยชน์ของ XML นั้น เป็นด้านความยืดหยุ่นในการใช้งานสำหรับแอพพลิเคชันที่อิงกับเว็บที่ใช้ง่ายในการค้นหาข้อมูล มีความยืดหยุ่นในการพัฒนาเว็บ สามารถผสมผสานข้อมูลจากหลายแหล่งจากแอพพลิเคชันที่ต่างกัน สามารถแสดงข้อมูลแบบต่างๆ และสามารถปรับปรุงข้อมูลให้ทันสมัยเสมอ และคาดว่าจะเป็นมาตรฐานใหม่ของระบบเปิด ซึ่งนับเป็นรูปแบบใหม่สำหรับการส่งข้อมูลบนเว็บที่มากด้วยข้อมูลหลายแบบ แต่ส่งผ่านด้วยเทคโนโลยีที่บีบอัดข้อมูลที่ให้ความเร็วได้รับการสนับสนุนจากผลิตภัณฑ์ค่ายไมโครซอฟท์

สถาปัตยกรรม XML

ภาษา XML ได้รับการสนับสนุนจาก W3C ให้นักพัฒนาเว็บแอพพลิเคชันได้หันมาใช้เป็นส่วนประกอบของการพัฒนาเว็บไซต์ เพราะ XML มีประสิทธิภาพและมีความน่าเชื่อสูงในการแปลงข้อมูลและโครงสร้างข้อมูลให้สามารถนำไปใช้งาน สถาปัตยกรรม 3 เทียร์ ที่ XML สามารถสร้างขึ้นจากระบบข้อมูลที่ใช้โมเดลของ 3-เทียร์ โครงสร้าง ของข้อมูลต่างๆ สามารถนำมาแสดงตามข้อกำหนด หรือรูปแบบที่ต้องการตามการใช้งานได้

XML เป็นการทำงานในระดับกลาง มิดเดิลเทียร์ที่สามารถเรียกใช้ฐานข้อมูลได้หลากหลายระบบฐานข้อมูลและโอนข้อมูลให้อยู่ในรูปแบบของ XML และมีการให้รายละเอียดเกี่ยวกับตัวข้อมูล โครงสร้างต่างๆ ของระบบฐานข้อมูลได้ XML เป็นระบบเปิดที่นำเสนอข้อมูลในรูปแบบเท็กซ์ผ่านทาง HTTP เหมือนกับ HTML แต่จะมีคุณสมบัติในการให้ข้อมูลแบบตามความเป็นจริง (real time) อัพเดทหรือเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงได้ตามข้อกำหนด การแสดงข้อมูลจาก XML ใน HTML จะเป็นการเพิ่มในส่วนของรายละเอียดข้อมูล ที่มีการเรียกใช้จากแหล่งหรือฐานข้อมูลที่เชื่อมโยงกันในหลายแหล่งเพื่อให้ HTML มีความสมบูรณ์มากขึ้น ในอนาคตการพัฒนาเว็บหรือการเขียน และสร้าง HTML ไม่จำเป็นต้องมีการเขียนชุดคำสั่งที่ยุ่งยากซับซ้อนมากก็สามารถทำงานร่วมกับระบบข้อมูลได้อย่างมีประสิทธิภาพ XML จะทำการกำหนดค่าสำหรับโครงสร้างข้อมูลที่จะนำไปแสดงใน HTML นอกจากนั้นยังสามารถนำไปสนับสนุนระบบการแลกเปลี่ยนข้อมูลข่าวสารทางอิเล็กทรอนิกส์ได้อย่างดีอีกด้วย

2.1.11 บทสรุปของ XML

XML มีความยืดหยุ่นพอในการนำเสนอข้อมูลแบบต่างๆ ที่สามารถให้รายละเอียดโครงสร้างข้อมูลตามระดับและความต้องการในการนำไปใช้งาน

2.2 SOAP

2.2.1 แนะนำ SOAP

SOAP (Simple Object Access Protocol) เป็นโพรโตคอลที่ใช้ XML เป็นพื้นฐาน เพื่อให้ซอฟต์แวร์คอมโพเนนต์ และแอพพลิเคชันสามารถติดต่อกันผ่าน HTTP ซึ่งเป็นมาตรฐานอินเตอร์เน็ตโพรโตคอลได้

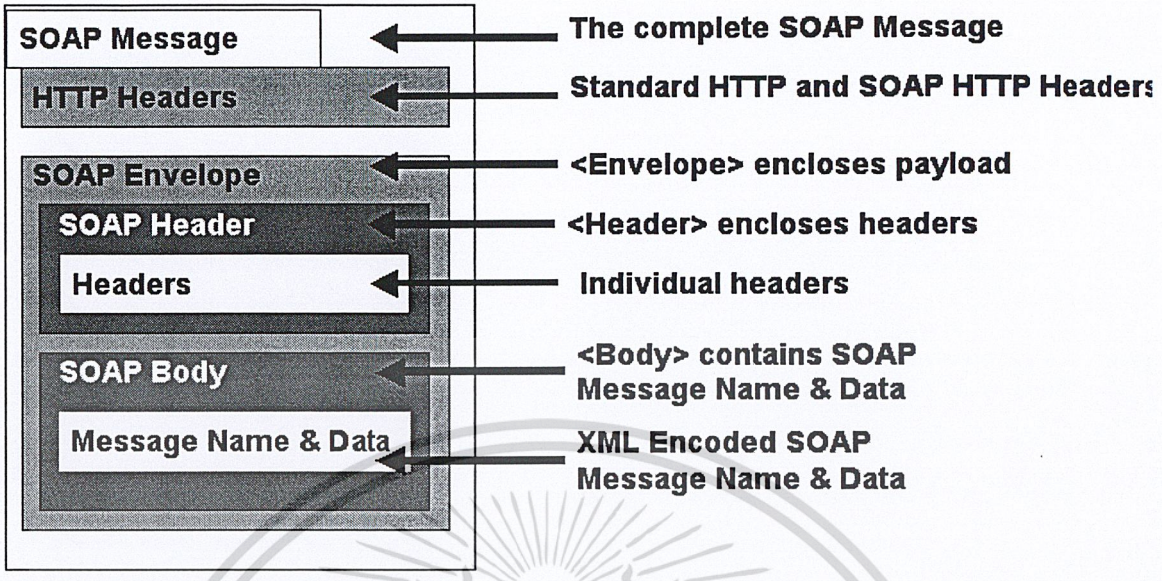
เนื่องจากการสื่อสารระหว่างแอพพลิเคชันจำเป็นต่อการพัฒนาแอพพลิเคชันบนอินเตอร์เน็ต เป็นอย่างมาก แต่แอพพลิเคชันแบบกระจายกันทำงาน (distributed application) ในปัจจุบันใช้ Remote Procedure Call (RPC) สื่อสารกันระหว่างออบเจ็กต์ เช่น DCOM และ CORBA ซึ่งไม่ได้ใช้พอร์ตเดียวกับ HTTP ที่เป็นพอร์ตมาตรฐานสำหรับให้บริการเว็บ ดังนั้น RPC จึงนำมาปรับใช้กับอินเตอร์เน็ตได้ยาก และมีปัญหาทางด้านความปลอดภัย ไฟร์วอลล์และพร็อกซีเซิร์ฟเวอร์จะไม่ยอมให้ส่งข้อมูลชนิดนี้ได้ตามปกติ วิธีที่ดีกว่าคือใช้ HTTP เพราะเป็นที่ยอมรับโดยอินเตอร์เน็ตบราวเซอร์ และเซิร์ฟเวอร์ทุกชนิด ซึ่ง SOAP ถูกสร้างมาเพื่อใช้ในกรณีนี้

ข้อดีของ SOAP ก็คือ SOAP ไม่ขึ้นกับคอมโพเนนต์เทคโนโลยี และภาษาการเขียนโปรแกรมใดๆ สามารถเขียนได้ง่ายและขยายเพิ่มเติมได้

2.2.2 ส่วนประกอบของ SOAP

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 โครงสร้างของ SOAP

SOAP เมสเซจ จะต้องเป็นไปตามกฎนี้

- Envelope เป็นอีลีเมนต์ที่อยู่บนสุด ต้องมีอีลีเมนต์นี้เสมอ
- Header อาจจะมีหรือไม่มีก็ได้ แต่ถ้ามีต้องเป็นอีลีเมนต์ลูกอีลีเมนต์แรกของenvelope
- Body ต้องเป็นอีลีเมนต์ลูกอีลีเมนต์แรกของenvelopeหรือheader

ตัวอย่างของ SOAP เมสเซจ ตัวอย่างนี้คือ GetLastTradePrice SOAP ร้องขอ (request) ที่ส่งไปยัง StockQuote เซอร์วิสประกอบด้วยสตริงพารามิเตอร์และคืนค่าโฟลท (float)กลับมากับ SOAP ตอบสนอง SOAP Envelope อีลีเมนต์จะอยู่ที่จุดบนสุดของเอกสาร XML XML เนมสเปซใช้เพื่อป้องกันการสับสนของ SOAP identifier

SOAP ร้องขอจะเป็นดังนี้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        <symbol>DIS</symbol>
      </m:GetLastTradePrice>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP ตอบสนองจะเป็นดังนี้

```
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 เมสเซจ จะมีการใช้ XML เนมสเปซ ทุกๆ อีลีเมนต์ในเอกสารจะขึ้นต้นด้วยเนมสเปซ เนมสเปซจะถูกกำหนดโดยใช้ xmlns แอตทริบิวต์ดังนี้

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

แต่หากไม่ต้องการให้ขึ้นต้นด้วยเนมสเปซก็สามารถทำได้ ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.1 Envelope

envelope อีลีเมนต์เป็นอีลีเมนต์บนสุดของเอกสาร XML ที่ใช้แสดงเมสเซจ อาจประกอบด้วยแอตทริบิวต์ คือ envelope เนมสเปซ และ encodingStyle

1. envelope เนมสเปซ – SOAP เมสเซจจะแสดงเวอร์ชันโดยใช้ เนมสเปซ ของ envelope อีลีเมนต์เนมสเปซ จะถูกนำไปใช้ขึ้นต้น envelope, headerและ Body อีลีเมนต์
2. EncodingStyle แอตทริบิวต์ – ใช้แสดงวิธี serialization ของ SOAP เมสเซจ แอตทริบิวต์นี้สามารถมีได้ในทุกอีลีเมนต์และจะมีผลกับ คอนเท็นต์ของอีลีเมนต์นั้นและอีลีเมนต์ลูกทั้งหมดของมันที่ไม่ได้ประกาศแอตทริบิวต์

2.2.2.2 Body

เป็นส่วนของข้อมูลที่ใช้ในการแลกเปลี่ยน โดยทั่ว ๆ ไปข้อมูลจะเป็นการ marshall RPC call และ error report อีลีเมนต์ลูกของBody อีลีเมนต์จะถูกเรียกว่า Body entry

Body entry จะต้องเป็นไปตามกฎนี้

- Body entry จะต้องแสดงด้วยชื่อเต็มที่ประกอบด้วย เนมสเปซ URI และ ชื่อของมัน (local name)
- SOAP encodingStyle แอตทริบิวต์อาจจะมีได้ เพื่อแสดงถึงวิธีเข้ารหัส (encode) ของ Body entry นั้น

2.2.2.3 header (Header)

เป็นส่วนเพิ่มเติมของเมสเซจสามารถประกอบด้วยอินฟอร์เมชันที่ระบุไปยังแอพพลิเคชัน ส่วนเพิ่มเติมนี้อาจนำไปใช้อิมพลีเม็นต์เป็น authentication, transaction management เป็นต้น ทุก ๆ อีลีเมนต์ลูกของ header อีลีเมนต์จะถูกเรียกว่า Header entry

Header entry จะต้องเป็นไปตามกฎดังนี้

- Header entry จะต้องแสดงด้วยชื่อเต็มที่ประกอบด้วย เนมสเปซ URI และ ชื่อของมัน (local name)
- อาจมี SOAP encodingStyle แอตทริบิวต์ที่ใช้สำหรับHeader entry
- SOAP mustUnderStand แอตทริบิวต์และ SOAP actor แอตทริบิวต์จะมีหรือไม่ก็ได้ เพื่อใช้บอกว่าจะจัดการกับเอ็นทรีนั้นอย่างไรและโดยใคร

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    </m:local>
</soap:Header>
```

1) SOAP Actor แอตทริบิวต์ – SOAP เมสเซจสามารถถูกส่งไปยังปลายทางผ่านตัวกลางของ SOAP (SOAP intermediary) ตัวกลางของ SOAP คือแอพพลิเคชันที่มีความสามารถทั้งรับและส่งต่อ SOAP เมสเซจ ในการส่งต่อตัวกลางจะต้องไม่ส่งต่อเฮดเดอร์อีลีเมนต์ไปให้กับแอพพลิเคชัน โดยตัวกลางนี้จะถูกกำหนดเป็น URI ถ้าไม่มี แอตทริบิวต์นี้หมายถึงผู้รับ SOAP เมสเซจเป็นปลายทางสุดท้าย
2) SOAP MustUnderStand แอตทริบิวต์ – ใช้แสดงว่า header entry นี้จำเป็นหรือเป็นเพียงอ็อปชันสำหรับผู้รับที่จะนำไปประมวลผล ค่าของมันคือ “1” หรือ “0” ถ้าไม่มีการระบุจะมีค่าเท่ากับ “0” โดย “0” หมายถึงเป็นอ็อปชัน และ “1” หมายถึงจำเป็น โดยจะต้องประมวลผลให้ถูกต้องตามกฎ (semantic) หรือต้องผิดพลาด (fail) เช่น เมสเซจเป็นส่วนหนึ่งของทรานแซคชั่น ถ้าปลายทางไม่รองรับทรานแซคชั่นจะต้องไม่ประมวลผลและส่งผลผิดพลาดกลับไปในกรณีที่ mustUnderStand แอตทริบิวต์เป็น “1” แต่ถ้าเป็น “0” จะยังคงประมวลผลต่อไป

2.2.3 SOAP ฟอลต์อีลีเมนต์ (SOAP Fault Element)

ข้อความแสดงความผิดพลาดจากแอพพลิเคชันของ SOAP จะเก็บอยู่ในฟอลต์อีลีเมนต์ ซึ่งถ้ามีจะต้องปรากฏใน body อีลีเมนต์เพียงครั้งเดียวใน 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>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</soap:Body>
</soap:Envelope>
```

SOAP ฟอลต์อีลีเมนต์มีอีลีเมนต์ย่อย ๆ ดังตารางที่ 2-5

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

ตารางที่ 2-5 SOAP Fault Element

ค่าของฟอลต์โค้ด (faultcode0 สามารถมีค่าได้ดังตารางที่ 2-6

Error	Description
VersionMismatch	เนมสเปซ ภายใน SOAP เอ็นเวโลปอีลีเมนต์ไม่ถูกต้อง
MustUnderstand	อีลีเมนต์ลูกของเฮดเดอร์อีลีเมนต์กับ mustUnderstand แอตทริบิวต์ที่มีค่า "1" ผู้รับไม่รองรับ
Client	เมสเซจมีรูปแบบไม่ถูกต้อง หรือมีอินฟอร์เมชันที่ไม่ถูกต้อง
Server	เกิดปัญหากับเซิร์ฟเวอร์ ไม่สามารถประมวลผลได้

ตารางที่ 2-6 ค่าฟอลต์โค้ด

2.2.4 SOAP เอ็นโค้ดดิ้ง

SOAP เอ็นโค้ดดิ้งมีวิธีการ map จากไทป์ของโปรแกรมมิ่งไปเป็น XML 2 วิธี คือจากภายนอกโดยใช้ WSDL (Web Services Description Language) ที่บอกถึงไทป์ของข้อมูลที่รับหรือส่ง หรือใช้ xsi:type แอตทริบิวต์ในกรณีที่ภาษาที่ใช้ไม่รองรับ WSDL โดยทั้ง 2 วิธีจะใช้โครงสร้างของ XML ในการระบุไทป์ ตัวอย่างของการใช้ xsi จะเป็นดังนี้

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
        soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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 รองรับไทป์ของข้อมูลได้ทั้งไทป์พื้นฐาน (simple type) และไทป์ประกอบ (compound type) เช่นสตรัค (struct) และอาร์เรย์ (array)

2.2.4.1 ตัวอย่างสตรัค (Struct)

สตรัค book ที่เขียนด้วยภาษา c++ จะเป็นดังนี้

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

Soap เมสเซจที่ไม่ใช้ WSDL แต่ใช้ xsi จะเป็นดังนี้

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<ns1:searchBook xmlns:ns1="http://soapinterop.org/" 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>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<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 ในส่วนของการประกาศโครงสร้างจะต้องประกอบด้วยโครงสร้างดังนี้

```
<types>
  <schema target namespace='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>
```

2.2.4.2 ตัวอย่างอาร์เรย์

ตัวอย่างนี้เป็นอาร์เรย์ของสตรัค 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">
<SOAP-ENV:Body>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<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 ในส่วนของการประกาศโครงสร้างจะต้องประกอบด้วยโครงสร้างดังนี้

```
<types>
  <schema target namespace='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'>
```

เอกสารนี้มีไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    <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>
```

2.2.5 SOAP ใน HTTP

ในการส่ง SOAP ผ่านทาง HTTP จะต้องใช้ content-type เป็น text/xml แต่ใน SOAP ร้องขอจะต้องมีเฮดเดอร์ SOAP Action ภายใน HTTP เฮดเดอร์ SOAP Action จะเป็นตัวบอกให้เซิร์ฟเวอร์รู้ว่า HTTP Post นั้นเป็น SOAP เมสเซจ และค่าของ เฮดเดอร์ คือ URI ที่แสดงถึงจุดหมายของ SOAP เมสเซจ ส่วน SOAP ตอบสนองจะต้องมี status code ตามมาตรฐานของ HTTP โดย 200-299 แสดงว่าสำเร็จ แต่ถ้าตอบสนองเมสเซจ เป็นการฟอลต์ แล้ว status code จะต้องเป็น 500 ซึ่งแสดงถึง internal server error ตัวอย่างของตอบสนองที่มี 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 ไม่ว่ากรณีใดๆทั้งสิ้น 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>
```

2.2.6 SOAP สำหรับ RPC

จุดประสงค์ของการออกแบบ SOAP คือทำ RPC โดยใช้ XML การเรียก RPC นั้นจะ map เข้ากับ HTTP ร้องขอ และ RPC ตอบสนองจะ map กับ HTTP ตอบสนอง

ในการทำ mechod call จำเป็นที่จะต้องมีอินฟอร์เมชันดังนี้

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

พิจารณาเมธอดดังนี้

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

ถ้าเนมสเปซของเมธอดคือ “http://www.worxstox.com/” แล้วการเรียกเมธอดโดยร้องขอ stock quote ใช้สัญลักษณ์ OU812 จะเป็นดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อเมธอดและชื่อของอีลีเมนต์จะต้อง match กันเช่นเดียวกับพารามิเตอร์

สำหรับ response จะต้องชื่ออีลีเมนต์จะต้องเป็นชื่อเมธอดตามด้วยตอบสนอง ดังนี้

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

ชื่ออีลีเมนต์ของค่าที่คืนมาสามารถเปลี่ยนได้ โดยจะกำหนดได้จากโปรแกรมที่เขียนหรือจาก WSDL

2.2.7 SOAP Toolkit

SOAP toolkit คือเครื่องมือที่จะทำหน้าที่ในการประมวลผลการร้องขอ SOAP และส่งการตอบสนองกลับไปให้ไคลเอนต์ โดย SOAP toolkit แต่ละตัวใช้ได้บนแพลตฟอร์มที่ต่างกัน รองรับเทคโนโลยีเว็บเซอร์วิสต่างกัน บางตัวอาจจะรองรับ WSDL หรือ UDDI ในขณะที่บางตัวรองรับอย่างใดอย่างหนึ่งหรือไม่รองรับทั้งสองอย่าง และถึงแม้ว่า SOAP จะเป็นอิสระต่อแพลตฟอร์ม แต่ใน SOAP toolkit บางตัวยังไม่สามารถทำงานข้ามผลิตภัณฑ์ (interoperability) ได้ เนื่องจากรองรับเทคโนโลยีของเว็บเซอร์วิส ไม่เหมือนกัน เช่น xsi ในบางผลิตภัณฑ์จะบังคับให้ SOAP เมสเซจจะต้องระบุไทป์ด้วย xsi หากไม่มีก็จะความผิดพลาดซึ่งในปัจจุบันแต่ละผลิตภัณฑ์ก็พยายามพัฒนาให้สามารถทำงานด้วยกันได้

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 2-7 SOAP Toolkit

2.3 UDDI

UDDI (Universal Description, Discovery and Integration) เป็นมาตรฐานที่จัดตั้งขึ้น โดยบริษัทไอบีเอ็ม, ไมโครซอฟท์และบริษัทยักษ์ใหญ่ทางธุรกิจ B2B (Business-to-Business) อื่น ๆ UDDI ถูกสร้างขึ้นมาเป็นมาตรฐาน ในการค้นหาบริการเว็บเซอร์วิสสำหรับคู่ค้าทางธุรกิจ ซึ่งเปรียบได้กับฐานข้อมูลขนาดใหญ่ ซึ่งมีข้อมูลของเว็บเซอร์วิสที่เปิดให้บริการ โดยที่เว็บไซต์สำหรับค้นหาเว็บเซอร์วิสมีอยู่หลายที่อย่างเช่น

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

2.4 WSDL

WSDL (Web Services Description Language) เกิดจากความร่วมมือระหว่างบริษัทไอบีเอ็มและไมโครซอฟท์ WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะของเว็บเซอร์วิสและวิธีการติดต่อกับเว็บเซอร์วิสนั้น ๆ โดยใช้ไวยากรณ์ของภาษา XML ซึ่ง WSDL อยู่ในความดูแลของ W3C เวอร์ชั่นล่าสุด คือ WSDL 1.1 รายละเอียด สามารถหาอ่านเพิ่มเติมได้จากเว็บไซต์ของ W3C ที่ http://www.w3.org/TR/wsdl ส่วนในทางปฏิบัติ หากเราต้องการ สร้างเว็บเซอร์วิสขึ้นมาเป็นของตนเอง 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"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      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>

<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>
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<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 3
เว็บลอจิกเว็บเซอร์วิส

3.1 คอนเวอร์เซชั่น (Conversation)

สำหรับเว็บเซอร์วิส สามารถติดต่อกับหลายผู้ใช้ (Client) ในการทำงานหนึ่งการทำงานเว็บเซอร์วิส สามารถติดต่อแบบ อะซิงโครนัส (asynchronous) โดยใช้ คอนเวอร์เซชั่นในการรักษาสถานะของข้อมูลที่ติดต่อได้พร้อมกัน คอนเวอร์เซชั่น มีปัญหาต้องจัดการ 2 อย่าง

1. รักษาสถานะ หรือข้อมูล โดยระบุการทำงานแบบ เว็บเซอร์วิส
2. จัดการ การเรียกใช้บริการ และ ตอบกลับไปยังเจ้าของ

3.1.1 การทำงานของคอนเวอร์เซชั่น

คอนเวอร์เซชั่น ทำงาน กับ คอนเทค (context) ที่ได้สร้างขึ้น โดย เว็บลอจิก เซิร์ฟเวอร์ (weblogic server) ในการสร้างคอนเวอร์เซชั่น จะสร้างโดยไม่มีการซ้ำกัน เพื่อรักษา สถานการณ์ทำงานของ เว็บเซอร์วิส การสร้างคอนเทค เพื่อใช้ในการติดต่อผู้ใช้

3.1.1.1 การรักษาสถานะโดยใช้การซีเรียวไลเซชั่น (serialization)

สถานะหมายถึง เหตุการณ์ ที่เว็บเซอร์วิส ณ เวลาหนึ่ง ได้สร้างข้อมูล ที่มันเก็บไว้ ตัวอย่าง เซอร์วิสได้รับหมายเลขเสียภาษี จากผู้ใช้ และใช้หมายเลขนั้นในการหาข้อมูล เซอร์วิสนั้นเก็บหมายเลขผู้เสียภาษี เมื่อเวลาที่ต้องการใช้ หมายเลขผู้เสียภาษี ดังนั้น สามารถได้รับค่าหมายเลขผู้เสียภาษี เมื่อเวลาที่ต้องการใช้ หมายเลขผู้เสียภาษี เมื่อเวลาที่ต้องการใช้ หมายเลขผู้เสียภาษี เป็นตัวอย่างของข้อมูลที่มีสถานะมาเกี่ยวข้อง ลองพิจารณา คอมพิวเตอร์ที่มีการปิดบริการทันที ถ้าหมายเลขผู้เสียภาษี เก็บเป็นเพียงตัวแปรในบริการในบริการนั้น จะเกิดการสูญหายของค่านั้นถ้ามีการสูญหายของค่านั้นทันทีจากเหตุการณ์ ปิดคอมพิวเตอร์

3.1.1.2 จัดการข้อมูลผ่านการระบุแบบไม่ซ้ำ

คอนเทคถูกสร้างเพื่อใช้การบริการ โดยใช้ในการติดต่อกัน รวมถึงข้อมูลที่เก็บในตัวแปร รวมไปถึงการระบุผู้ใช้ในการใช้บริการตั้งแต่เริ่มเรียกใช้ข้อมูล เว็บเซอร์วิสอาจมีการเรียกใช้การทำงานมากมาย ณ เวลาเดียวกัน จากผู้ใช้คนเดียวหรือหลายคน ในการติดต่อ เซอร์วิสต้องการ จดจำสถานะเพื่อใช้ในการจัดการการร้องขอและการตอบกลับอย่างมีประสิทธิภาพ

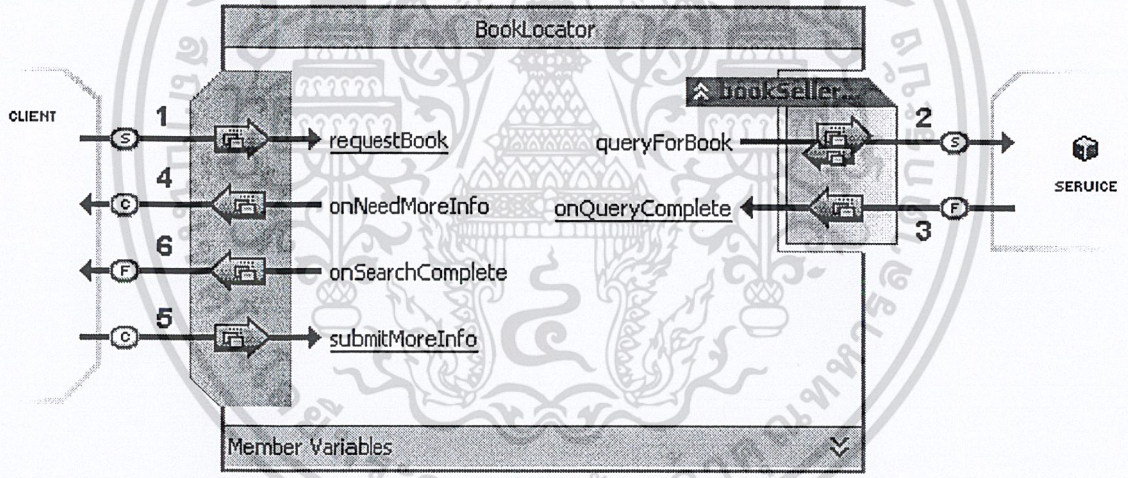
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง

- แต่ละคอนเวอร์เซชั่น เริ่มการทำงานโดย เว็บลอจิกเซิร์ฟเวอร์ สร้าง หมายเลขการติดต่อเฉพาะที่ใช้เรียก คอนเวอร์เซชั่น หมายเลขคอนเวอร์เซชั่นจะสัมพันธ์กับผู้ใช้ (เจ้าของการเรียกใช้)
- แต่ละการทำงานจะมีผลกับข้อมูลที่ถูกเรียกโดยใช้ หมายเลขคอนเวอร์เซชั่น ในการระบุเจ้าของการเรียกใช้ เพราะฉะนั้น ผู้ใช้จะใช้หมายเลขคอนเวอร์เซชั่นในการติดต่อ

3.1.2 วงจรชีวิตของคอนเวอร์เซชั่น

คอนเวอร์เซชั่นมีวงจรชีวิตของมันเอง จะเริ่มต้นเมื่อผู้ใช้เรียกบิการทางเว็บโดยใช้เมธอด "เริ่ม" (start) และทำงานโดยใช้เมธอด "ต่อไป"(continue) และจบโดยใช้เมธอด "จบ" (finish) ในการเรียกใช้เมธอดครั้งแรกจะมีการสร้างคอนเทค เพื่อใช้ในการติดต่อ และการเรียกใช้อื่นอื่น มีการทำซีเรียวไลเซชั่นเพื่อแก้ไขข้อมูลและเมื่อจบการทำงานของ คอนเวอร์เซชั่น เว็บลอจิกจะมีการคืนทรัพยากรและทำลายสถานะที่จัดการโดยเว็บเซอร์วิส



รูปที่ 3-1 วงจรชีวิตของคอนเวอร์เซชั่น

3.2 ทรานแซ็กชัน

3.2.1 เหตุผลที่ต้องใช้ทรานแซ็กชัน

ตัวอย่างปัญหาที่ต้องใช้ทรานแซ็กชันในการแก้

- **Atomic Operation**

กรณีที่ต้องการทำคำสั่งหลาย ๆ คำสั่ง โดยมองทั้งหมดเป็นเหมือนคำสั่งเดียว (atomic operation) เช่น กรณีบัญชีธนาคาร เมื่อต้องการโอนเงินจากบัญชีหนึ่งไปยังอีกบัญชีหนึ่ง ต้องถอนเงินออกจากบัญชีหนึ่ง จากนั้นจึงฝากเงินเข้าอีกบัญชีหนึ่ง ถ้ามีคำสั่งใดคำสั่งหนึ่งล้มเหลว ต้องให้คำสั่งทั้งคู่ล้มเหลว ไม่เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นยอดเงินคงเหลือในบัญชีจะไม่ถูกต้อง จึงต้องไม่เกิดสถานการณ์ที่คำสั่งหนึ่งทำงานสำเร็จ ในขณะที่อีกคำสั่งทำงานล้มเหลว เนื่องจากทั้งสองคำสั่งนั้นเป็นทรานแซ็กชันเดียวกัน

การแก้ปัญหานี้วิธีหนึ่งคือการใช้ฟังก์ชั่นในการดักจับข้อผิดพลาด (exception) ดังตัวอย่างต่อไปนี้

```
try {
// Withdraw funds from account 1
}
catch (Exception e) {
// If an error occurred, do not proceed.
return;
}
try {
// Otherwise, deposit funds into account 2
}
catch (Exception e) {
// If an error occurred, do not proceed,
// and redeposit the funds back into account 1.
return;
}
```

ตัวอย่างโค้ดข้างต้น ถอนเงินจากบัญชีที่ 1 ก่อน หากเกิดข้อผิดพลาดขึ้น แอพพลิเคชันจะจบการทำงาน และไม่มีผลของการทำงานใด ๆ เกิดขึ้น หากไม่เกิดข้อผิดพลาดก็จะฝากเงินเข้าไปยังบัญชีที่ 2 ในขณะฝากเงินนั้น หากเกิดข้อผิดพลาด จะต้องทำการฝากเงินกลับเข้าไปในบัญชีที่ 1 ใหม่ จากนั้นจึงจบการทำงาน

ปัญหาที่เกิดขึ้นจากการแก้ปัญหาด้วยวิธีนี้คือ

- โปรแกรมมีความซับซ้อน
- ต้องตรวจสอบทุกความเป็นไปได้ที่จะทำให้เกิดการทำงานผิดพลาดในทุก ๆ ขั้นตอน และต้องเขียนโค้ดเพื่อยกเลิกการเปลี่ยนแปลงที่ได้ทำไปก่อนที่จะเกิดความผิดพลาดนั้นเอง
- ในการตรวจสอบความผิดพลาด แทบจะไม่สามารถควบคุมได้หากกระบวนการทั้งหมดมีความซับซ้อน เช่นการปรับปรุงข้อมูลทางการเงิน ที่ต้องอาศัยการทำงานทั้งหมด 10 ขั้นตอนด้วยกัน จะต้องหาวิธีการในการรองรับการทำงานที่ผิดพลาดในทุกขั้นตอนที่เป็นไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

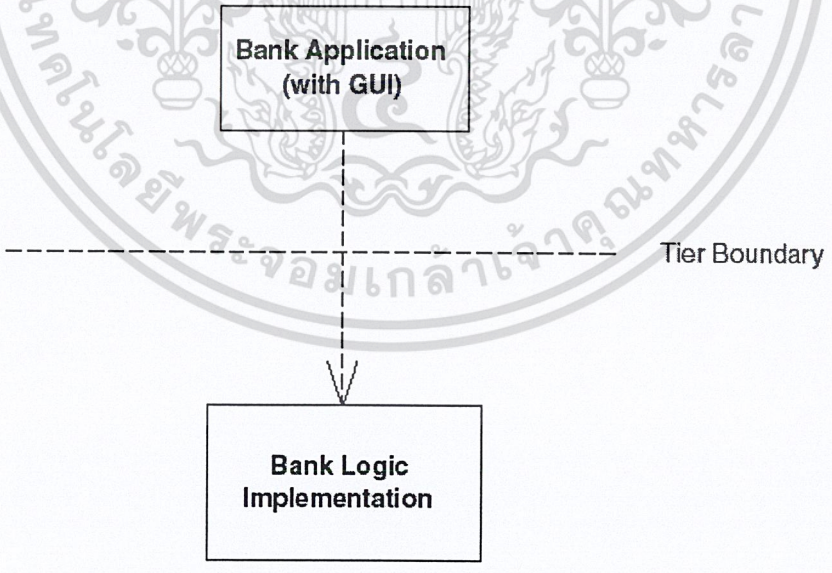
- การตรวจสอบการทำงานได้ยาก เพราะต้องทำการจำลองการทำงานที่ผิดพลาดที่เกิดขึ้นได้ในทุกขั้นตอน

ในทางอุดมคตินั้นต้องการวิธีการที่สามารถทำคำสั่งทั้งสองได้ โดยมองเป็นคำสั่งใหญ่เพียงคำสั่งเดียว และรับประกันว่าคำสั่งทั้งสองจะต้องทำงานสำเร็จทั้งคู่ หรือทำงานผิดพลาดทั้งคู่เท่านั้น

- **การทำงานผิดพลาดของเครื่องคอมพิวเตอร์ หรือระบบเครือข่าย**

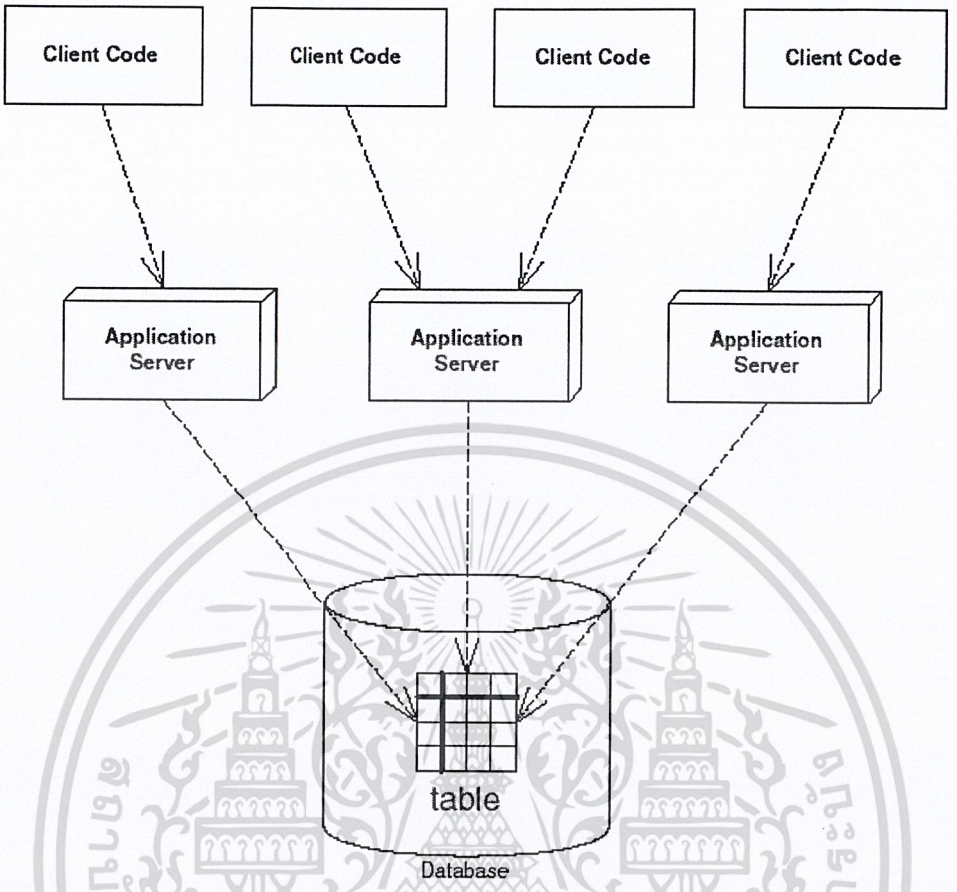
จากตัวอย่างข้างต้น ถ้าลอจิกต่าง ๆ ที่ทำงานกับบัญชีธนาคารถูกกระจายบนเครือข่ายโดยดีพลอยแบบมัลติเทียร์ เพื่อเหตุผลทางด้านความปลอดภัย , การเพิ่มขยาย หรือประโยชน์จากการแบ่งโปรแกรมออกเป็นโมดูล ในการดีพลอยแบบมัลติเทียร์ ไคลเอ็นต์ที่ต้องการใช้งานแอพพลิเคชันบัญชีธนาคารนี้จะต้องกระทำผ่านทาง remote method invocation แสดงในรูปที่ 3-2 ทำให้ต้องคำนึงถึงเรื่องของการทำงานผิดพลาดและความน่าเชื่อถือ เช่น ระบบเครือข่ายแครชระหว่างที่โปรแกรมบัญชีธนาคารกำลังทำคำสั่งใด ๆ อยู่ แม้ว่าเมื่อระบบเครือข่ายแครชจะมีการสร้าง exception ขึ้นมา แต่การแครชของระบบเครือข่ายนั้นอาจจะเกิดขึ้นก่อนหน้าการถอนเงิน หรืออาจเกิดขึ้นภายหลังการถอนเงินก็ได้ ซึ่งไม่อาจแยกความแตกต่างระหว่าง exception ที่เกิดขึ้นจากกรณีทั้งสองนี้ได้เลย

ในความเป็นจริง ระบบเครือข่ายไม่ได้เป็นจุดเดียวที่ทำให้เกิดปัญหานี้ขึ้น เนื่องจากข้อมูลบัญชีธนาคารก็คือข้อมูลแบบ persistent ที่อยู่ในฐานข้อมูล มีความเป็นไปได้เสมอที่ระบบฐานข้อมูลจะแครช หรือเครื่องคอมพิวเตอร์ที่ดีพลอยระบบฐานข้อมูลไว้จะเกิดแครช ถ้าหากการแครชนั้นเกิดขึ้นในระหว่างที่กำลังเขียนข้อมูลลงไปในฐานข้อมูล อาจทำให้ฐานข้อมูลอยู่ในสถานะที่ไม่ consistent ได้



รูปที่ 3-2 แอพพลิเคชันที่ทำงานในระบบแบบกระจาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 การใช้งานฐานข้อมูลพร้อม ๆ กัน

- **การที่ผู้ใช้หลายคนสามารถใช้งานข้อมูลเดียวกันพร้อมกันได้**

ในระบบแบบกระจายในระดับเอนเตอร์ไพรซ์ (enterprise-level distributed system) เป็นรูปแบบที่ไคลเอ็นต์หลาย ๆ ตัวติดต่อกับแอพพลิเคชันเซิร์ฟเวอร์หลาย ๆ ตัว ถ้าแอพพลิเคชันเซิร์ฟเวอร์ทั้งหมดใช้ฐานข้อมูลเดียวกัน (ดังรูปที่ 3-3) เซิร์ฟเวอร์สามารถที่จะแก้ไขข้อมูลตัวเดียวกันที่อยู่ในฐานข้อมูลได้

ถ้ามีไคลเอ็นต์สองตัวพยายามแก้ไขข้อมูลเดียวกันที่อยู่ในฐานข้อมูลในเวลาเดียวกันแล้ว คำสั่งของไคลเอ็นต์ทั้งสองอาจทำงานซ้อนกัน และได้ผลลัพธ์สุดท้ายผิดพลาด ซึ่งการมีข้อมูลที่ไม่ถูกต้องในฐานข้อมูล ในระบบทั่วไปถือว่าเป็นสิ่งที่ยอมรับไม่ได้ (unacceptable) และอาจสร้างความเสียหายอย่างมากให้กับองค์กรดังนั้นจึงจำเป็นต้องหาวิธีการที่จะจัดการกับการแก้ไขข้อมูลพร้อมกันของไคลเอ็นต์หลาย ๆ ตัว โดยจะต้องไม่ทำให้ผลลัพธ์สุดท้ายที่ได้นั้นเป็นผลลัพธ์ที่ไม่ถูกต้อง

3.2.2 ประโยชน์ของทรานแซ็กชัน

ทรานแซ็กชัน คือเซ็ตของคำสั่งที่ถูกเอ็กซิคิวต์โดยมองว่าเป็นคำสั่งใหญ่เพียงอันเดียว ทรานแซ็กชันรับประกันว่าคำสั่งทั้งหมดทำงานสำเร็จ มิฉะนั้นจะต้องไม่มีผลการทำงานจากคำสั่งใดเกิดขึ้นเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานแซ็กชันสามารถแก้ปัญหาที่เกิดจากการทำงานผิดพลาดของระบบเครือข่ายได้ อนุญาตให้ไคลเอ็นต์หลาย ๆ ตัวใช้งานข้อมูลเดียวกันพร้อมกันได้ และรับประกันว่าข้อมูลทั้งหมดที่คำสั่งเหล่านี้ทำการแก้ไขจะถูกต้องสมบูรณ์ โดยไม่ถูกรบกวนจากการแก้ไขข้อมูลโดยไคลเอ็นต์อื่น ๆ

การใช้ทรานแซ็กชันอย่างถูกต้องทำให้สามารถมีหลายไคลเอ็นต์ทำงานกับฐานข้อมูลได้โดยไม่ขึ้นต่อกัน เช่น ถ้ามีไคลเอ็นต์สองตัวอ่านและเขียนข้อมูลลงในฐานข้อมูล ไคลเอ็นต์ทั้งสองจะ mutual exclusive กัน ระบบจัดการฐานข้อมูลจะจัดการเกี่ยวกับการควบคุมการทำงานพร้อมกันที่จำเป็นบนฐานข้อมูล เช่น การล็อคข้อมูล เพื่อไม่ให้การทำงานของแต่ละเธรดมีผลกระทบซึ่งกันและกัน

3.2.2.1 ACID Properties

เมื่อใช้งานทรานแซ็กชันอย่างถูกต้อง คำสั่งจะถูกเอ็กซิคิวต์ โดยมีการรับประกันผลการทำงานทั้งสิ้น 4 อย่างด้วยกัน เรียกว่า ACID properties โดยคำว่า ACID ย่อมาจาก Atomicity , Consistency , Isolation และ Durability

- **Atomicity** - รับประกันว่าคำสั่งทั้งหมดจะถูกรวมเข้าไว้ด้วยกัน และมองเป็นหน่วยของการทำงานเพียงอันเดียว การทำงานของทรานแซ็กชันจะเป็นไปในลักษณะของ All-or-Nothing คือ การอัพเดตข้อมูลในฐานข้อมูลทั้งหมดจะต้องเสร็จสิ้น หรือถ้าเกิดข้อผิดพลาดใด ๆ ขึ้นระหว่างการทำงาน การอัพเดตฐานข้อมูลทั้งหมดนั้นจะต้องไม่เกิดขึ้นเลย โดยการอัพเดตฐานข้อมูลที่เกิดขึ้นก่อนแล้ว จะต้องถูกยกเลิก
- **Consistency** - รับประกันว่าทรานแซ็กชันจะต้องทำให้ระบบอยู่ในสถานะที่ consistent เมื่อทรานแซ็กชันทำงานเสร็จเรียบร้อย เช่น การโอนเงินจะต้องไม่มีเงินหายไปจากระบบ แต่ในระหว่างการทำงานของทรานแซ็กชันนี้ ขณะที่เงินถูกถอนออกมาจากบัญชีแรก ข้อมูลอยู่ในสถานะที่ไม่ consistent แต่เนื่องจากทรานแซ็กชันจะมองเห็นคำสั่งทั้งหมดเปรียบเสมือนเป็นคำสั่งเดียว ดังนั้นถ้าทรานแซ็กชันสำเร็จก็คือคำสั่งทั้งหมดจะต้องสำเร็จ แต่ถ้าทรานแซ็กชันทำงานไม่สำเร็จคำสั่งทั้งหมดก็จะต้องไม่สำเร็จ และการเปลี่ยนแปลงใด ๆ ที่ได้กระทำไประหว่างนั้นจะต้องถูกยกเลิก ตามคุณสมบัติข้อแรกของทรานแซ็กชัน ดังนั้นข้อมูลจะอยู่ในสถานะที่ consistent เสมอ
- **Isolation** - เป็นการป้องกันทรานแซ็กชันที่ทำงานพร้อมกัน ไม่ให้เห็นผลลัพธ์ที่ยังไม่สมบูรณ์ที่เกิดจากทรานแซ็กชันอื่น isolation อนุญาตให้หลายทรานแซ็กชันสามารถอ่านหรือเขียนฐานข้อมูลโดยไม่รู้ถึงการมีอยู่ของทรานแซ็กชันอื่น ถ้าไม่มีคุณสมบัติ isolation สถานะของข้อมูลอาจจะอยู่ในสถานะที่ไม่ consistent ได้ ซึ่งมีประโยชน์มากในกรณีไคลเอ็นต์หลายตัวแก้ไขฐานข้อมูลในเวลาเดียวกัน โดยแต่ละไคลเอ็นต์จะทำงานโดยเปรียบเสมือนว่ามีเพียงตัวมันเองเท่านั้นที่กำลังใช้งานฐานข้อมูลอยู่ ทรานแซ็กชันสร้างคุณสมบัติ isolation โดยการใช้ synchronization โพรโตคอลต่าง ๆ กับข้อมูลในฐานข้อมูล เช่น ระหว่างการทำงานของทรานแซ็กชัน ข้อมูลจะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล็อคและปลดล็อคโดยอัตโนมัติตามความจำเป็น และการอัพเดตข้อมูลของทรานแซ็กชันอื่น จะไม่มีผลกับข้อมูลที่ทรานแซ็กชันนี้กำลังทำงานอยู่

- **Durability** - รับประกันว่าการเปลี่ยนแปลงข้อมูลใด ๆ ในฐานข้อมูลที่เสร็จสิ้น (commit) แล้วจะต้องมีความถาวร คือต้องไม่สูญหายเมื่อเกิดความผิดพลาดใด ๆ ขึ้น เช่น เครื่องคอมพิวเตอร์แครช ,ระบบเครือข่ายแครช , ฮาร์ดดิสก์แครช หรือไฟฟ้าดับ โดยต้องมีการเก็บ log file ของการอัพเดตข้อมูลใด ๆ ในทรานแซ็กชัน เมื่อระบบเกิดแครชขึ้น ข้อมูลสามารถสร้างขึ้นมาใหม่โดยการทำงานตามลำดับใน log file อีกครั้งหนึ่ง

3.2.3 ทรานแซ็กชันโมเดล

ทรานแซ็กชันโมเดลเป็นวิธีการที่ต่างกันในการทำทรานแซ็กชัน มีหลายโมเดลที่ใช้จัดการทรานแซ็กชัน แต่ละโมเดลจะมีความซับซ้อนและฟีเจอร์ที่ต่างกัน 2 โมเดลที่ได้รับความนิยมมากที่สุดคือ flat transaction และ nested transaction

- **Flat Transactions**

flat transaction เป็นทรานแซ็กชันโมเดลที่เข้าใจได้ง่าย flat transaction เป็นกลุ่มของ operation ที่ทำงานเป็นหน่วยเดียว ถ้า operation ไม่สำเร็จทั้งหมดจะเกิด rollback ขึ้น แต่ถ้าสำเร็จจะเกิด commit กระบวนการของ flat transaction ดังรูปที่ 3-4

- **Nested Transactions**

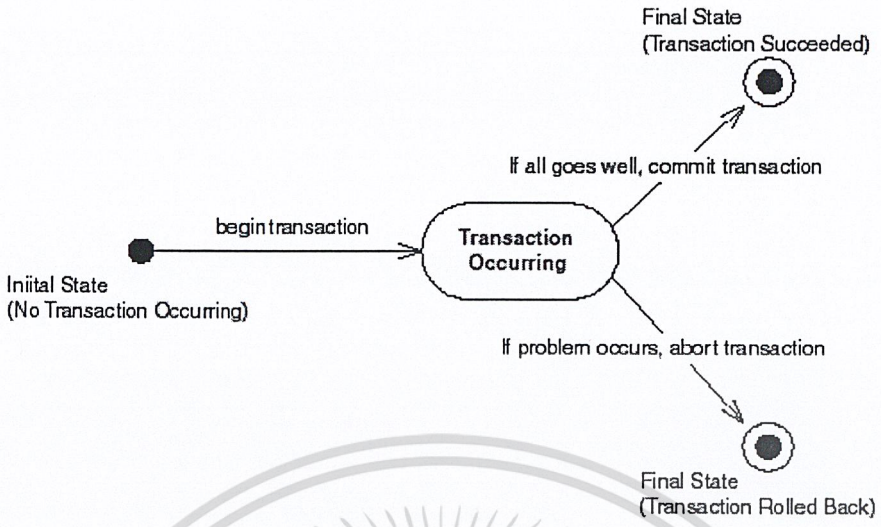
ตัวอย่างของงานที่ต้องใช้ nested transaction คือปัญหา trip-planning โดยแอพพลิเคชันจะมี operation ดังนี้

1. ซื้อตั๋วรถไฟจากบอสตันไปยังนิวยอร์ค
2. ซื้อตั๋วเครื่องบินจากนิวยอร์คไปยังลอนดอน
3. ซื้อตั๋วบอลลูนจากลอนดอนไปยังปารีส
4. แอพพลิเคชันพบว่าไม่มีไฟลท์ไปยังปารีส

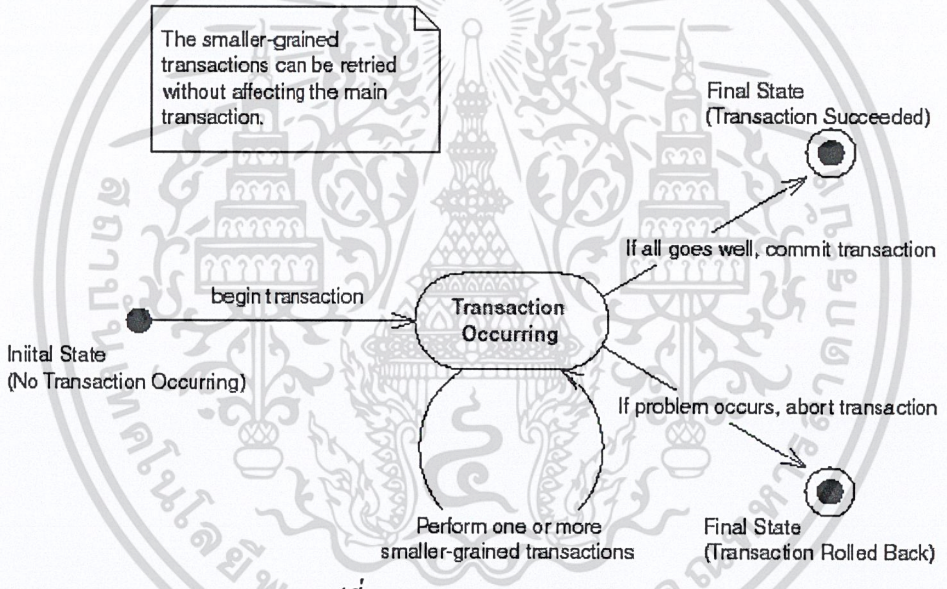
ถ้าลำดับของ operation นี้ทำกับ flat transaction จะเกิด rollback เท่านั้น เนื่องจากไม่มีไฟลท์ไปยังปารีส แอพพลิเคชันจะต้องไม่เก็บการจองใด ๆ ไว้เลย แต่ตรงจุดนี้อาจมีทางในการเปลี่ยนการเดินทางด้วยบอลลูนไปเป็นทางอื่น ทำให้เก็บข้อมูลการจองไว้ได้

nested transaction ช่วยในการแก้ปัญหานี้ มันจะยอมให้เกิดทรานแซ็กชันซ้อนกันได้ รูปที่ 3-5 แสดงแนวคิดของ nested transaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 Flat Transaction



รูปที่ 3-5 Nested Transaction

- โมเดลอื่น ๆ

ทรานแซ็กชันโมเดลยังมี chained transaction และ sagas แต่จะไม่อธิบายถึง เพราะ EJB specification ไม่รองรับโมเดลเหล่านั้น รวมถึง nested transaction ด้วย มันจะรองรับเฉพาะ flat transaction เท่านั้น

3.2.4 การใช้งานทรานแซ็กชันกับ EJB

3.2.4.1 วิธีการกำหนดการจัดการทรานแซ็กชัน

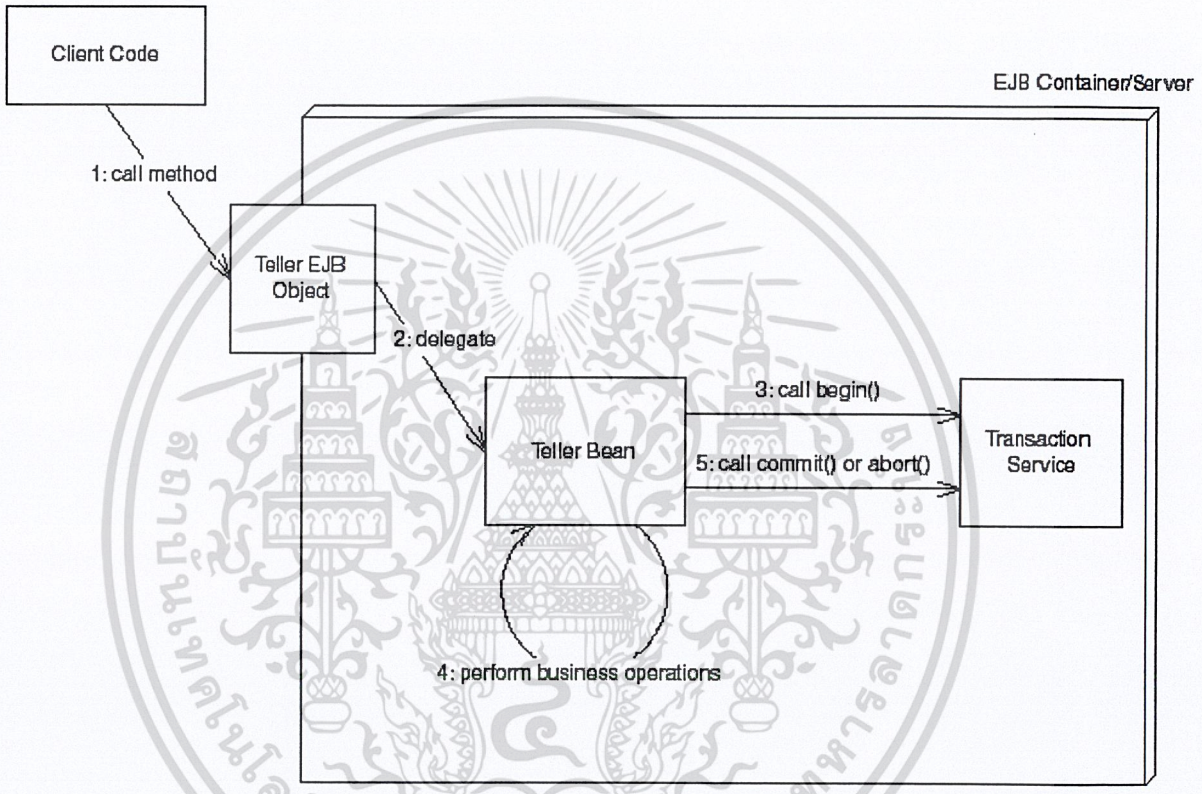
ทรานแซ็กชันใด ๆ ผลการทำงานของทรานแซ็กชันจะจบลงด้วยการ commit หรือ abort อย่างใดอย่างหนึ่งเท่านั้น โดยวิธีการกำหนดขอบเขตของทรานแซ็กชันสำหรับ EJB มี 2 วิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) Programmatic Transaction

ผู้พัฒนาแอพพลิเคชันเป็นผู้เขียนโปรแกรมลอจิกเกี่ยวกับการควบคุมทรานแซ็กชันลงไปในโค้ดของแอพพลิเคชันเอง ต้องเป็นผู้สั่งคำสั่ง begin , commit หรือ abort เอง

การทำงานของ programmatic transaction แสดงดังรูปที่ 3-6



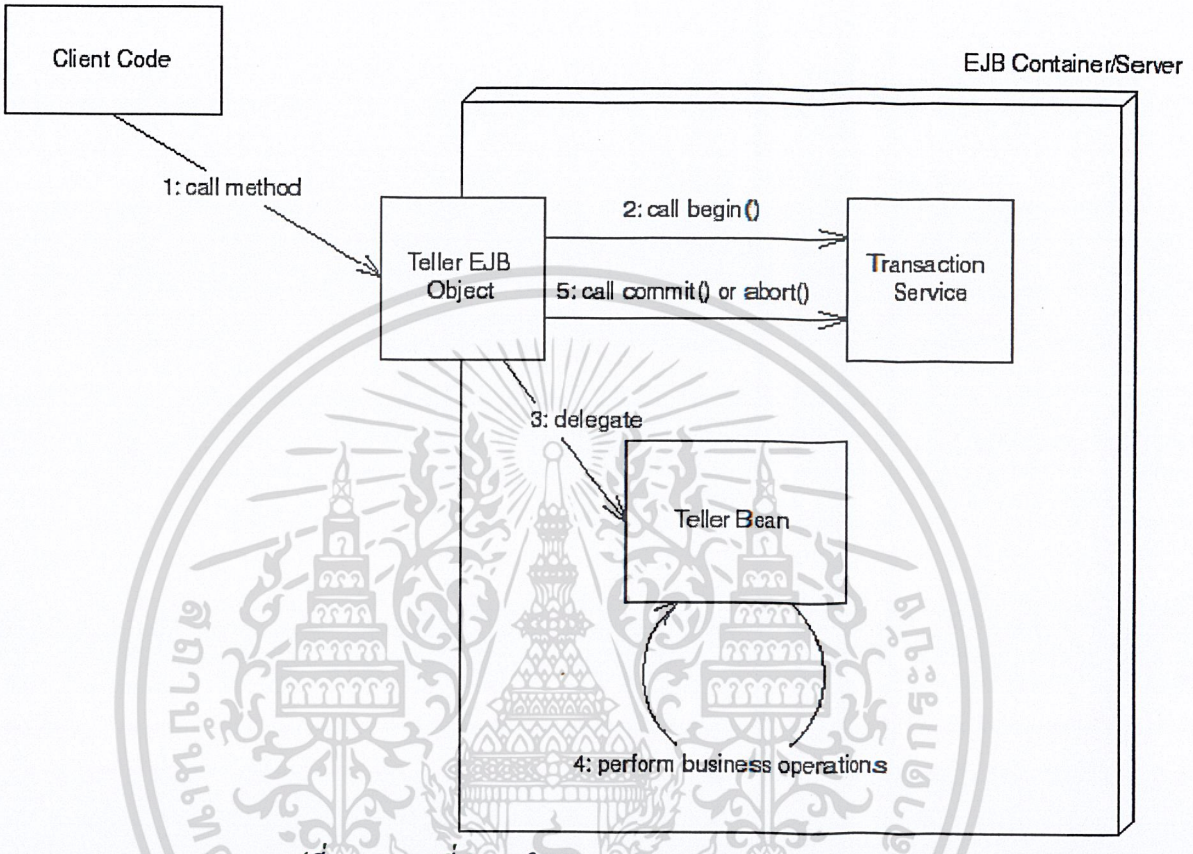
รูปที่ 3-6 bean ที่ทำงานกับ Programmatic Transaction

2) Declarative Transaction

declarative transaction อนุญาตให้คอมโพเน็นต์ทำงานในรูปแบบทรานแซ็กชันโดยอัตโนมัติ enterprise bean ไม่จำเป็นต้องสั่ง begin , commit หรือ abort เอง คอนเทนเนอร์จัดการให้อัตโนมัติ EJB คอนเทนเนอร์จะดักจับการร้องขอที่มีเข้ามาและเริ่มต้นทรานแซ็กชันอัตโนมัติโดยการสั่ง begin หลังจากนั้นคอนเทนเนอร์ก็จะส่งต่อการร้องขอไปยัง enterprise bean ซึ่งจะทำงานอยู่ในทรานแซ็กชันที่เกิดขึ้น ถ้าหากมีข้อผิดพลาดใด ๆ bean จะส่งไปบอกคอนเทนเนอร์ให้ยกเลิกทรานแซ็กชัน เมื่อการทำงานเสร็จสิ้น bean จะส่งการควบคุมกลับมาให้คอนเทนเนอร์ คอนเทนเนอร์จะสั่ง commit หรือ abort โดยขึ้นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับว่ามีข้อผิดพลาดเกิดขึ้นระหว่างการทำงานหรือไม่ การทำงานของ declarative transaction แสดงใน รูปที่ 3-7



รูปที่ 3-7 bean ที่ทำงานใน Declarative Transaction

declarative transaction ช่วยให้ bean ไม่จำเป็นต้องติดต่อกับ API ของทรานแซ็กชันโดยตรง โดยที่โค้ดของ bean และไคลเอ็นต์จะไม่รู้เลยว่ามันทำงานอยู่ภายใต้ทรานแซ็กชัน

การระบุว่าต้องการทำงานในแบบ declarative หรือ programmatic transaction นั้น EJB กำหนดให้ระบุไว้ในแท็ก persistence-type ที่อยู่ใน ejb-jar.xml ถ้ามีค่าเท่ากับ Bean จะเป็นการระบุให้ bean ควบคุมเอง แต่ถ้าเป็น Container จะให้คอนเทนเนอร์เป็นผู้ควบคุมจะมีค่าดังนี้

<persistence-type>Bean</persistence-type>

<persistence-type>Container</persistence-type>

3.2.4.2 การจัดการทรานแซ็กชันของคอนเทนเนอร์

transaction attribute เป็นค่าที่ระบุว่าคอนเทนเนอร์จะควบคุมทรานแซ็กชันของ bean อย่างไร โดย สามารถระบุ transaction attribute ที่แตกต่างกันสำหรับแต่ละ bean ในระบบได้โดยไม่ต้องคำนึงถึงว่าจะมี bean ทั้งหมดกี่ตัวที่ทำงานอยู่ร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

transaction attribute เป็นส่วนหนึ่งของ deployment descriptor ของแต่ละ bean deployment descriptor ที่มากับ bean จะระบุคุณสมบัติต่าง ๆ ในการทำงานของ bean คอนเทนเนอร์จะจัดการกับทรานแซ็กชันของ bean ตามที่ได้ดูจาก transaction attribute ใน deployment descriptor และสามารถกำหนด transaction attribute สำหรับทุก ๆ เมธอดของ bean หรือจะระบุแตกต่างกันไปในแต่ละเมธอดก็ได้ โดยถ้ามีการระบุค่าทั้งสองแบบ คอนเทนเนอร์จะยึดถือค่าที่ระบุให้แต่ละเมธอดเป็นหลัก ตัวอย่างของการกำหนด transaction attribute ใน ejb-jar.xml ดังนี้

```
<assembly-descriptor>
    <container-transaction>
      <method>
        <ejb-name>Product</ejb-name>
        <method-name>*</method-name>
      </method>
      <!—
      Transaction attribute. Can be “NotSupported” , “Supports” , “Required” ,
      “RequiresNew” , “Mandatory” , or “Never”
      -->
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
</assembly-descriptor>
```

ค่า transaction attribute ที่เป็นไปได้ มีดังนี้

- **Required**

จะใช้งานโหมด Required เมื่อต้องการให้ bean ทำงานในทรานแซ็กชันเสมอ ถ้ามีทรานแซ็กชันกำลังทำงานอยู่แล้ว bean จะทำงานร่วมกับทรานแซ็กชันเดิม แต่ถ้าไม่มีทรานแซ็กชันกำลังทำงานอยู่ bean จะเริ่มต้นการทำงานในทรานแซ็กชันใหม่

- **RequiresNew**

จะใช้ค่านี้เมื่อต้องการให้มีการเริ่มต้นทรานแซ็กชันใหม่ทุกครั้งเมื่อ bean ถูกเรียกใช้งาน ถ้ามีทรานแซ็กชันเดิมกำลังทำงานอยู่ก่อนแล้ว ทรานแซ็กชันนั้นจะถูกหยุดไว้ชั่วคราว คอนเทนเนอร์จะเริ่มต้นการทำงานของทรานแซ็กชันใหม่และส่งต่อการร้องขอมายัง bean ใหม่ที่ถูกเรียกใช้งานนี้ เมื่อ bean ทำงานเสร็จสิ้น คอนเทนเนอร์ก็จะสั่ง commit หรือ abort ทรานแซ็กชันนี้ และกลับไปทำงานต่อในทรานแซ็กชันเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Supports**

เมื่อ bean ถูกเรียกใช้งานด้วย Supports มันจะทำงานในทรานแซ็กชันเมื่อไคลเอ็นต์นั้นทำงานอยู่ในทรานแซ็กชันก่อนแล้ว โดยทำงานในทรานแซ็กชันเดิม แต่ถ้าไคลเอ็นต์ไม่ได้ทำงานในทรานแซ็กชัน bean ก็จะไม่เริ่มต้นการทำงานในทรานแซ็กชันใหม่

- **Mandatory**

ระบุว่าต้องมีทรานแซ็กชันกำลังทำงานอยู่แล้วเมื่อเมธอดของ bean ถูกเรียกใช้งาน ถ้าหากไม่มีทรานแซ็กชันทำงานอยู่ก่อน exception *javax.ejb.TransactionRequired* จะถูกส่งกลับไปยังผู้เรียก

- **NotSupported**

bean จะไม่สามารถทำงานในทรานแซ็กชันได้ ยกตัวอย่างเช่น สมมติว่าเรามี enterprise bean A และ B เมื่อ A เริ่มต้นทรานแซ็กชัน และเรียกใช้งาน B หาก B มีการตั้งค่า transaction attribute เป็น NotSupported แล้ว ทรานแซ็กชันที่ A เริ่มต้นไว้จะหยุดชั่วคราว คำสั่งของ B ทั้งหมดจะทำงานโดยไม่อยู่ในทรานแซ็กชัน เมื่อ B ทำงานเสร็จเรียบร้อย ทรานแซ็กชันของ A จะกลับมาทำงานต่อ

- **Never**

ใช้ระบุว่า bean จะต้องไม่มีทรานแซ็กชัน ถ้าไคลเอ็นต์ที่เรียก bean อยู่ในทรานแซ็กชันจะเกิด exception คืนมาให้ไคลเอ็นต์

ตารางที่ 3-1 แสดงผลลัพธ์ที่เกิดจาก transaction attribute

3.2.4.3 การจัดการทรานแซ็กชันของบีน

การระบุการทำงานทรานแซ็กชันเป็นแบบ bean ควบคุมทรานแซ็กชันของ bean เองนั้น EJB กำหนดให้ระบุไว้ในแท็ก persistence-type ที่อยู่ใน ejb-jar.xml หรือกำหนดในตอนเริ่มสร้าง bean มีค่าเท่ากับ Bean จะเป็นการระบุให้ bean ควบคุมเอง ทำให้ผู้พัฒนาแอพพลิเคชันสามารถเขียนโปรแกรมลอจิกเกี่ยวกับการควบคุมทรานแซ็กชันลงไปในโค้ดของแอพพลิเคชันเอง และต้องเป็นผู้สั่งคำสั่ง begin , commit หรือ rollback เอง

3.2.5 Transactional Isolation

คุณสมบัติ isolation รับประกันว่า ผู้ใช้ที่ทำงานอยู่พร้อม ๆ กันจะไม่รู้สึกและได้รับผลกระทบจากผู้ใช้คนอื่น ๆ แม้ว่าผู้ใช้ทั้งหมดนั้นกำลังทำงานกับข้อมูลเดียวกันในฐานข้อมูลก็ตาม โดยสามารถระบุระดับของ isolation ที่ต้องการได้

TRANSACTION ATTRIBUTE	CLIENT'S TRANSACTION	BEAN'S TRANSACTION
Required	none	T2
	T1	T1
	none	T2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RequiresNew	none	T2
RequiresNew	T1	T2
	T1	T1
Mandatory	none	error
	T1	T1
NotSupported	none	none
	T1	none
	none	none
	T1	error

ตารางที่ 3-1 ผลลัพธ์ที่เกิดจาก Transaction Attribute

3.2.5.1 ระดับของ Isolation

การระบุระดับของ isolation สามารถควบคุมได้ว่าจะให้แต่ละทรานแซ็กชัน isolate จากกันมากน้อยเพียงใด ถ้าตั้งระดับของ isolation ที่สูง ก็มั่นใจได้ว่าแต่ละทรานแซ็กชันนั้น จะถูก isolate ออกจากทรานแซ็กชันอื่น ๆ ทั้งหมด แต่การตั้งระดับของ isolation สูงก็ทำให้ประสิทธิภาพในการทำงานที่ได้ลดลง เนื่องจากในการทำ isolation นั้นใช้วิธีการล็อคค่าในฐานข้อมูลเป็นหลัก แต่ถ้าตั้งระดับของ isolation ไว้ต่ำ ทรานแซ็กชันก็จะไม่ isolate แต่ประสิทธิภาพในการทำงานร่วมกันของหลาย ๆ ทรานแซ็กชันจะดีขึ้น

ดังนั้น ต้องพิจารณาให้รอบคอบว่า ต้องการความเป็น isolation สูงต่ำเพียงใดในการทำงาน การกำหนด transaction isolation มีอยู่ทั้งหมด 4 ระดับด้วยกัน ดังนี้

- **READ UNCOMMITTED**

เป็นระดับการทำงานที่ต่ำที่สุด ไม่มีการรับประกันคุณสมบัติ isolation เลย แต่จะให้ประสิทธิภาพในการทำงานสูงสุด การทำงานในระดับนี้อาจเกิดปัญหา dirty read ขึ้นได้ โดยหากทรานแซ็กชันหนึ่งทำงานพร้อมกันไปกับทรานแซ็กชันอื่น แล้วอีกทรานแซ็กชันนั้นเขียนข้อมูลโดยยังไม่ได้ commit ทรานแซ็กชันเดิมจะอ่านค่านั้นเข้ามาได้ทำให้ได้ค่าที่ผิดพลาดไป โดยไม่ขึ้นกับระดับของ isolation ที่อีกทรานแซ็กชันหนึ่งใช้งานอยู่ นอกจากนี้ยังเกิดปัญหา unrepeatable read และ phantom phenomenon ได้อีกด้วย

- **READ COMMITTED**

การทำงานในระดับนี้คล้ายกับระดับ READ UNCOMMITTED แตกต่างกันที่ในระดับนี้โค้ดจะอ่านค่าข้อมูลที่ commit แล้วเท่านั้น จะไม่อ่านค่าที่ถูกเขียนลงไปยังฐานข้อมูลแล้วแต่ยังไม่ commit ดังนั้นระดับของ isolation ระดับนี้จึงสามารถแก้ปัญหา dirty read ได้ แต่ยังไม่สามารถแก้ปัญหาที่ซับซ้อน เช่นปัญหา unrepeatable read และ phantom phenomenon

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **REPEATABLE READ**

เป็นการรับประกันที่สูงขึ้นไปจาก READ COMMITTED เมื่อใดก็ตามที่อ่านค่าข้อมูลที่ commit แล้วจากฐานข้อมูล สามารถจะอ่านค่าข้อมูลนั้นซ้ำได้และค่าที่ได้จะเป็นค่าที่เหมือนกับค่าที่เราอ่านขึ้นมาได้ครั้งแรก ซึ่งระดับของ isolation ที่ READ COMMITTED หรือต่ำกว่า ทรานแซ็กชันอื่นที่ทำงานอยู่พร้อมกันอาจจะ commit ข้อมูลในระหว่างการอ่านข้อมูลแต่ละครั้งได้ ทำให้ค่าของข้อมูลเดียวกันที่อ่านได้ในแต่ละครั้งไม่ตรงกัน การทำงานในโหมดนี้แก้ปัญหาที่เรียกว่า unrepeatable read

- **SERIALIZABLE**

สามารถแก้ปัญหา phantom phenomenon คือการที่เซตของข้อมูลใหม่ถูกเพิ่มเข้ามาในฐานข้อมูลระหว่างการอ่านข้อมูลของทรานแซ็กชัน รวมถึงแก้ปัญหาอื่น ๆ ที่กล่าวมาก่อนหน้านี้ทั้งหมด รับประกันว่าทุก ๆ ทรานแซ็กชันจะทำงานเป็นลำดับกับทรานแซ็กชันอื่น ๆ และมีคุณสมบัติ ACID properties ของฐานข้อมูลครบถ้วน หมายความว่าแต่ละทรานแซ็กชันจะทำงานแยกกับทรานแซ็กชันอื่น ๆ อย่างแท้จริง

ตารางที่ 3-2 จะแสดงถึงปัญหาที่อาจเกิดในแต่ละระดับของ isolation

ISOLATION LEVEL	DIRTY READ?	UNREPEATABLE READS?	PHANTOM READ?
READ UNCOMMITTED	Yes	Yes	Yes
READ COMMITTED	No	Yes	Yes
REPEATABLE READ	No	No	Yes
SERIALIZABLE	No	No	No

ตารางที่ 3-2 ระดับของ Isolation

การกำหนด isolation ใน EJB ในกรณีของ bean-managed persistent จะกำหนด isolation โดยใช้ resource manager API เช่น JDBC โดยการเรียก *java.sql.Connection.SetTransactionIsolation(...)* ส่วนในกรณีของ container-managed persistent จะไม่สามารถกำหนดระดับของ isolation ใน deployment descriptor ได้ การกำหนดระดับของ isolation จะขึ้นอยู่กับผลิตภัณฑ์ที่ใช้เป็นคอนเทนเนอร์หรือฐานข้อมูล

3.2.6 ทรานแซ็กชันแบบกระจาย

นอกจากทรานแซ็กชันพื้นฐาน คือแอพพลิเคชันเซิร์ฟเวอร์หนึ่งตัวเชื่อมต่อกับฐานข้อมูลหนึ่งตัวแล้ว ยังสามารถที่จะทำ distributed transaction ได้เช่นเดียวกัน โดยขึ้นอยู่กับการรองรับของแอพพลิเคชันเซิร์ฟเวอร์ ซึ่ง distributed transaction ยึดหลักการเดียวกับทรานแซ็กชันธรรมดา โดยที่หากมีคอมโพเน็นต์หนึ่งบนเครื่องคอมพิวเตอร์เครื่องหนึ่ง abort ทรานแซ็กชัน ทรานแซ็กชันทั้งหมดจะต้องถูก abort ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นกัน distributed transaction สามารถทำงานได้กับหลายแอพพลิเคชันเซิร์ฟเวอร์และหลายฐานข้อมูล หมายความว่าคอมโพเน็นต์ที่ดีพลอยในแอพพลิเคชันเซิร์ฟเวอร์หนึ่ง ทำงานในทรานแซ็กชันเดียวกันร่วมกับอีกคอมโพเน็นต์ซึ่งถูกดีพลอยในอีกแอพพลิเคชันเซิร์ฟเวอร์ ในการทำงานบิซิเนสเมธอดหนึ่ง ๆ เครื่องหลายเครื่องอาจจำเป็นต้องทำงานร่วมกัน distributed transaction จึงอนุญาตให้หลายแอพพลิเคชันเซิร์ฟเวอร์ซึ่งอาจเป็นซอฟต์แวร์จากผู้พัฒนาคนละรายกันทำงานร่วมกันในทรานแซ็กชันได้

3.2.6.1 Durability และโพรโตคอล Two-Phase Commit

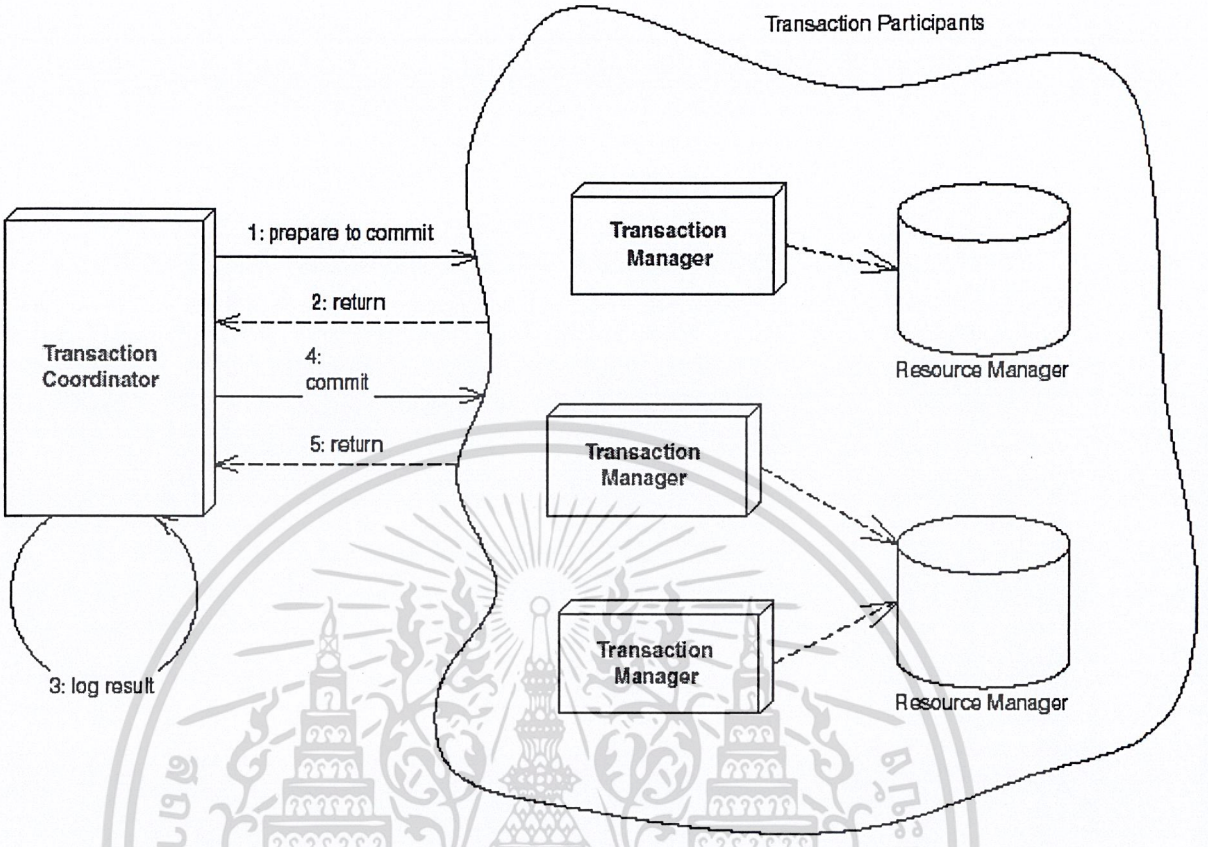
durability รับประกันว่า การอัพเดต resource ที่ commit แล้วทุกครั้งจะต้องมีความถาวร สามารถอิมพลีเมนต์ได้ไม่ยากถ้าทำงานกับฐานข้อมูลเพียงฐานข้อมูลเดียว แต่กรณีที่มีหลาย resource manager เข้ามาเกี่ยวข้องด้วย หากมี resource หนึ่งเกิดความผิดพลาดในการทำงานใด ๆ ขึ้น เช่น ฐานข้อมูลแครชจะต้องมีวิธีที่จะกู้ข้อมูลกลับคืนมาให้ได้

วิธีหนึ่งในการแก้ปัญหานี้คือ การบันทึก log ของคำสั่งที่ทำงานกับฐานข้อมูลทั้งหมดก่อนที่คำสั่งนั้นจะทำงานจริง จะช่วยให้สามารถกู้ข้อมูลคืนมาได้ โดยดูข้อมูลได้จาก log file และทำการอัพเดตข้อมูลต่าง ๆ ใหม่อีกครั้ง ด้วยวิธีการนี้ทรานแซ็กชันจะเสร็จสิ้นในสองเฟส

- **เฟสที่หนึ่ง** - เริ่มต้นโดยส่งข้อความ before commit ไปยังทุก resource ที่เกี่ยวข้องในทรานแซ็กชัน ณ จุดนี้ resource ที่เกี่ยวข้องในทรานแซ็กชันจะมีโอกาสที่จะยกเลิกทรานแซ็กชันเป็นครั้งสุดท้าย ถ้าหากมี resource ใดตัดสินใจยกเลิกทรานแซ็กชัน ทรานแซ็กชันทั้งหมดก็จะถูกยกเลิก และการอัพเดต resource ต่าง ๆ ก็จะไม่เกิดขึ้น มิฉะนั้นทรานแซ็กชันก็จะทำงานต่อไปและไม่มีการหยุดนอกจากจะมีความผิดพลาดเกิดขึ้น เพื่อป้องกันความผิดพลาด การอัพเดต resource ใด ๆ ในแต่ละครั้งจะถูกบันทึกลงใน log หรือ journal ที่มีความคงทน (persistent) โดยข้อมูลเหล่านี้จะไม่สูญหายไปเมื่อมีการแครชเกิดขึ้น และสามารถใช้ข้อมูลเหล่านี้ในการอัพเดต resource ต่าง ๆ ได้อย่างถูกต้อง
- **เฟสที่สอง** - จะเกิดขึ้นก็ต่อเมื่อเฟสที่หนึ่งจบลงโดยไม่มีการยกเลิกการทำงาน ณ จุดนี้ทุก resource manager ซึ่งอาจถูกควบคุมจากคนละที่ จะทำการอัพเดตข้อมูลจริง ๆ

การแบ่งจุดที่ทรานแซ็กชันทำงานสำเร็จออกเป็นสองเฟสนี้ เรียกว่าโพรโตคอล two-phase commit หรือ *2PC* มันมีประโยชน์มากเนื่องจากทำให้หลาย transaction manager และ resource manager ทำงานร่วมกันในทรานแซ็กชันแบบกระจายได้ หากมีใครก็ตามระบุว่าทรานแซ็กชันควรถูกยกเลิก ทุก resource ก็จะต้องถูก roll back ดังรูปที่ 3-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 ทรานแซ็กชันแบบกระจายที่มีการใช้โพรโตคอล Two-Phase Commit

ในระบบ two-phase commit แบบกระจาย มี transaction manager หลักอยู่หนึ่งตัว เรียกว่า distributed transaction coordinator ทำหน้าที่ควบคุมทรานแซ็กชันทั้งหมด และจัดการกับแต่ละคำสั่งที่มาจากหลาย ๆ transaction manager ในเครือข่าย ขั้นตอนต่อไปนี้เป็นขั้นตอนที่เกิดขึ้นในทรานแซ็กชัน two-phase commit แบบกระจาย

1) transaction coordinator ส่งข้อความ prepare to commit ไปยังทุก transaction manager ที่เกี่ยวข้อง
2) แต่ละ transaction manager อาจส่งผ่านข้อความนี้ไปยังแต่ละ resource manager ที่สัมพันธ์กับ transaction manager นั้น
3) แต่ละ transaction manager รายงานกลับไปยัง transaction coordinator หากทุก transaction manager ยินยอมที่จะ commit คำสั่ง commit ที่กำลังจะเกิดขึ้นจะถูกบันทึกลง log เพื่อใช้ในกรณีที่เกิดความผิดพลาดขึ้น
4) สุดท้าย transaction coordinator จะบอกแต่ละ transaction manager ให้ทำการ commit แต่ละ transaction manager จะเรียก resource manager ให้อัพเดตข้อมูล ถ้าหากมีข้อผิดพลาดใด ๆ เกิดขึ้น log จะถูกใช้ในการทำขั้นตอนนี้ซ้ำอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Asynchronous and Synchronous

ถ้า เว็บเซอร์วิส ที่คุณกำลัง ออกแบบ มีกระบวนการที่อาจจะใช้เวลานานในการดำเนินการคุณจะต้องออกแบบส่วนติดต่อผู้ใช้ ของ เว็บเซอร์วิส โดยที่ไคลเอ็นต์ไม่ต้องหยุดรอกระบวนการ(method)นั้นเสร็จ โดยทำให้เป็นเว็บเซอร์วิสของคุณเป็นอะซิงโครนัส(asynchronous) โดยในการออกแบบคุณสามารถทำได้โดยให้กระบวนการนั้นๆ(ที่ใช้เวลานาน) ส่งค่าผลการรับคำร้องขอกลับไปให้ไคลเอ็นต์ก่อนในทันที หลังจากนั้นค่อยส่งผลลัพธ์ไปให้ไคลเอ็นต์เมื่อกระบวนการนั้นทำงานเสร็จแล้ว

มันเป็นเรื่องสำคัญที่จะแยกแยะระหว่าง เว็บเซอร์วิสที่เป็นอะซิงโครนัส กับ กระบวนการที่เป็นอะซิงโครนัส(กระบวนการที่ ส่งค่ากลับทันที และ ส่งค่าเป็น วอย(void)) แต่เว็บเซอร์วิส ที่เป็นอะซิงโครนัสนั้นไม่จำเป็นที่กระบวนการต่างๆของมันต้องเป็น อะซิงโครนัส ยกตัวอย่าง เช่นเว็บเซอร์วิสที่ใช้กระบวนการเป็นอะซิงโครนัส เมื่อได้รับการร้องขอจากไคลเอ็นต์จะส่งค่า การยอมรับ(acknowledgement) กลับไปให้ไคลเอ็นต์ แต่ไม่ใช่ผลลัพธ์หลังจากนั้นจะเรียกซิงโครนัส คอลแบ็ก(synchronous callback ของไคลเอ็นต์โดยส่งค่าผลลัพธ์กลับไปและรอรับการยืนยันจากไคลเอ็นต์ โดยทั้งกระบวนการของเว็บเซอร์วิสและ คอลแบ็กต่างก็เป็นซิงโครนัสทั้งคู่ แต่เว็บเซอร์วิสเป็น อะซิงโครนัส

มี 2 วิธีในการออกแบบ อะซิงโครนัส เว็บเซอร์วิส

1. สร้างกระบวนการที่เริ่มต้นการร้องขอ และ คอลแบ็ก เพื่อส่งส่งค่าผลลัพธ์กลับ
2. สร้างกระบวนการที่เริ่มต้นการร้องขอ และ เพิ่มกระบวนการ ที่ส่งค่าผลลัพธ์กลับ สถานะของการร้องขอนั้น(ยังไม่เสร็จ/เสร็จแล้ว) วิธีการนี้เรียกว่า การติดต่อแบบสอบถาม(polling interface)

คุณอาจจะต้อง สร้าง โดยใช้ทั้ง 2 วิธี ใน เว็บเซอร์วิส ของคุณ เนื่องจากคุณจะสามารถอำนวยความสะดวก ของทั้ง ไคลเอ็นต์ ที่สามารถจัดการกับ คอลแบ็ก ได้ และ ที่ไม่สามารถจัดการได้ (โดยใช้การติดต่อแบบสอบถาม)

3.3.1 การใช้คอลแบ็กเพื่อที่จะระบุเหตุการณ์กับไคลเอ็นต์

คอลแบ็ก ใช้ในการเตือน ไคลเอ็นต์ ของ เว็บเซอร์วิส ว่ามีบางเหตุการณ์เกิดขึ้น เช่น คุณต้องการจะบอกกับ ไคลเอ็นต์ ว่า ผลลัพธ์ที่ ไคลเอ็นต์ร้องขอมาเสร็จแล้ว

การเพิ่ม คอลแบ็ก ให้กับส่วนติดต่อของ เว็บเซอร์วิส นั้นง่าย แต่คุณต้องตระหนักว่า ไม่ใช่ว่าทุกไคลเอ็นต์ จะรองรับการติดต่อแบบคอลแบ็กได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การใช้การติดต่อแบบสอบถามแทนที่การใช้คอลแบ็ก

ไคลเอ็นต์ บางรายของ เว็บเซอร์วิส อาจไม่สามารถจัดการกับ คอลแบ็ก ได้ คุณสามารถทำให้ไคลเอ็นต์ ที่ไม่สามารถรับ คอลแบ็ก ใช้ เว็บเซอร์วิสของคุณได้โดยการเพิ่มการติดต่อแบบสอบถามให้เป็นอีกทางเลือหนึ่งของ ไคลเอ็นต์ โดยในการติดต่อแบบสอบถามจะสร้าง กระบวนการ ที่ ไคลเอ็นต์สามารถใช้เป็นตัวคอยตรวจสอบได้ว่า ผลลัพธ์ของการร้องขอก่อนหน้าของตนพร้อมแล้วหรือยัง ดังโปรแกรมตัวอย่างข้างล่าง

```
public class Conversation  {
   /**
    * @jws:operation
    * @jws:conversation phase="start"
    */
   public void startRequest()
   {
      ...
   }


   /**
    * @jws:operation
    * @jws:conversation phase="continue"
    */
   public String getRequestStatus()
   {
      ...


   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    /**
     * @jws:operation
     * @jws:conversation phase="finish"
     */
    public void terminateRequest()
    { }
}
```

ไคลเอ็นต์ จะเรียก กระบวนการ startRequest เพื่อเริ่มการร้องขอหลังจากนั้น ไคลเอ็นต์ สามารถเรียก getRequestStatus เป็นระยะๆ เพื่อตรวจสอบผลลัพธ์ ระหว่างการเรียก getRequestStatus ไคลเอ็นต์จะสามารถทำกระบวนการอื่นๆได้ โดย getRequestStatus จะส่งค่ากลับว่าการร้องขอนั้นกำลังทำงานอยู่จนกว่าการร้องขอจะเสร็จโดยครั้งถัดไปถ้า ไคลเอ็นต์ เรียก getRequestStatus หลังจากที่การร้องขอเสร็จแล้วผลลัพธ์ จะถูกส่งกลับมาที่ ไคลเอ็นต์ โดย ไคลเอ็น ต์สามารถหยุดการติดต่อได้โดยใช้ กระบวนการ terminateRequest

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ความปลอดภัยในการใช้งานเว็บเซอร์วิส

Security หมายถึง เทคนิคในการที่จะมั่นใจได้ว่าข้อมูลที่ถูกเก็บอยู่ในคอมพิวเตอร์ หรือข้อมูลที่ถูกส่งผ่านระหว่างเครื่องคอมพิวเตอร์จะไม่ถูกเปลี่ยนแปลง โดยหลักการวัด security โดยทั่วไปนั้นส่วนใหญ่จะเกี่ยวกับการ พิสูจน์ข้อความ และการเข้ารหัส

การส่งข้อมูลใน e-comerce นั้นจะมี access point หลายจุดซึ่งผู้ประสงค์ร้ายจะสามารถดักจับ และเปลี่ยนแปลงข้อมูลได้ โดยเฉพาะยิ่งธุรกิจมีการกระจายตัวมากเท่าไรก็จะยิ่งเพิ่มโอกาสที่จะโดนโจมตีได้มากขึ้นจึงมีความต้องการซอร์ฟแวร์ที่จะมาช่วยจัดการในด้านนี้

โดยในตอนต้นของบทนี้จะพูดถึงทฤษฎีที่เกี่ยวข้องกับการทำ Security ก่อน

4.1 Cipher Suite

Cipher Suite คือ กระบวนการเข้ารหัส (encryption) ของ SSL(Secure Socket Layer) ที่รวม key exchange algorithm, symmetric encryption และ secure hash algorithm เข้าด้วยกัน ตัวอย่างเช่น Cipher Suite ที่เรียกว่า RSA_WITH_RC4_128MD5 ใช้ RSA สำหรับ key exchange , RC4 ที่ใช้ 128-bit key ในการ encryption และ MD5 สำหรับการทำเมสเซจไดเจสต์ (Message digest)

4.2 Public Key Algorithms

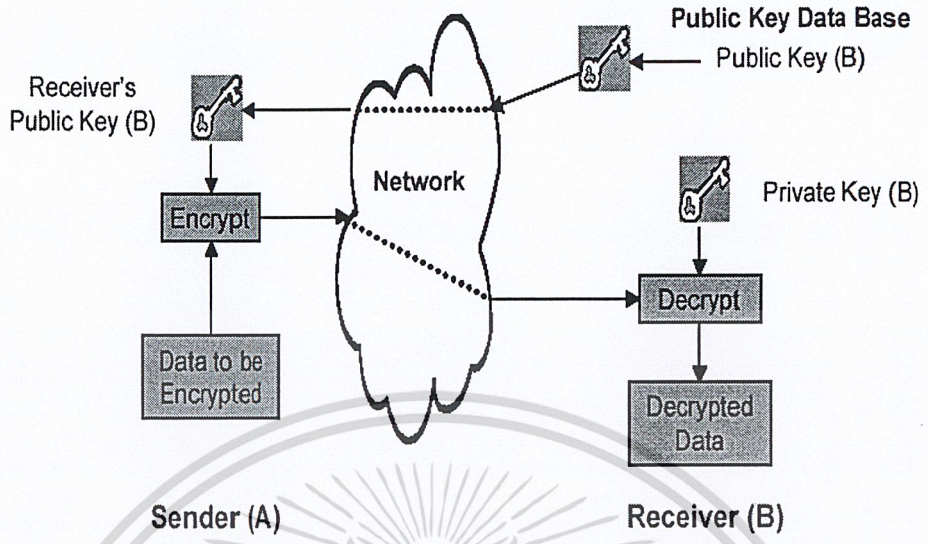
Public key(or asymmetric key)algorithms เป็นการใช้คีย์ 2 คีย์ที่แตกต่างกัน แต่มีความสัมพันธ์กันทางคณิตศาสตร์

- คีย์สาธารณะ (A public key)(ที่ซึ่งถูกแจกจ่ายโดยทั่วไป) ใช้สำหรับการยืนยันลายมือชื่อดิจิตอล (digital signature) ว่าเป็นของผู้ส่งจริง หรือ ใช้ในการเปลี่ยนข้อมูลให้อยู่ในรูปที่ไม่สามารถเข้าใจได้
- คีย์ส่วนตัว(A private key)(ที่ซึ่งถูกเก็บเป็นความลับ) ใช้ในการสร้างลายมือชื่อดิจิตอลหรือเปลี่ยนข้อมูลที่ถูกส่งมาให้กลับอยู่ในรูปเดิม

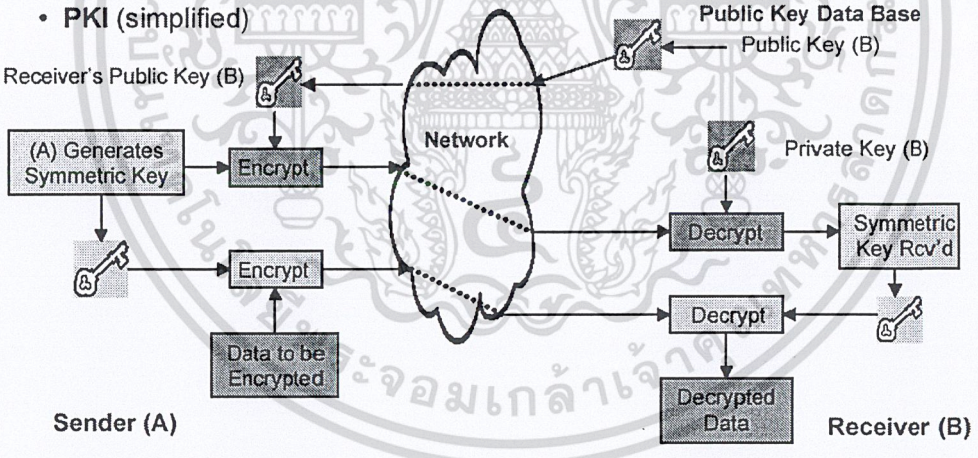
4.3 Symmetric Key Algorithms

ใน symmetric key algorithms จะใช้คีย์เดียวกันทั้งการเข้ารหัสและการถอดรหัสเมสเซจโดยการใช้วิธีนี้จะเร็วกว่าแบบ public key cryptography ประมาณ 1,000 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-1 แสดงการใช้คีย์คู่ในการติดต่อ



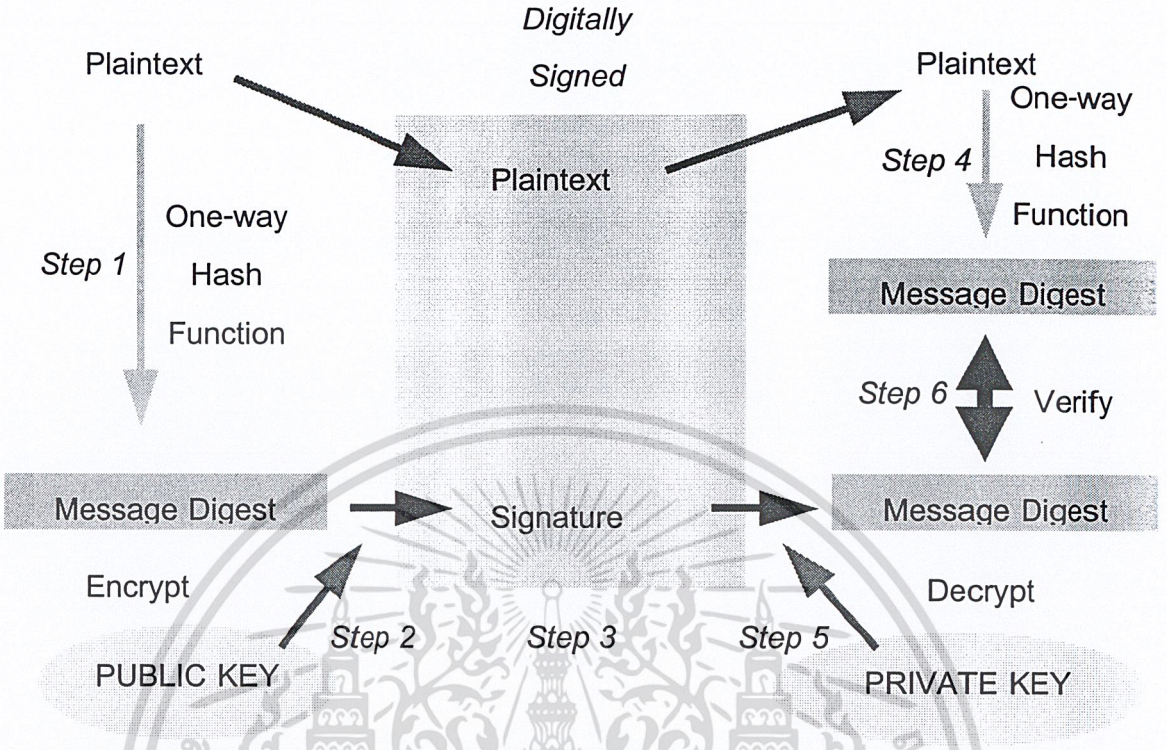
รูปที่ 4-2 แสดงการได้มาซึ่งคีย์เดี่ยวในการติดต่อ

4.4 Message Digest Algorithms

สนับสนุนอัลกอริทึมเมสเซจไดเจสต์ แบบ MD5 และ SHA(Secure Hash Algorithm) ซึ่งเป็น one-way hash algorithm โดย one-way hash algorithm จะแปลงเมสเซจให้อยู่ในรูปแบบสตริงที่มีขนาดคงที่ซึ่งเรียกว่า เมสเซจไดเจสต์ (message digest) หรือ hash value

MD5 จะเป็น high-speed 128-bit hash ส่วน SHA จะมีความปลอดภัยมากกว่าโดยใช้ 160-bit hash แต่จะช้ากว่า MD5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-3 แสดงการนำ Message Digest ไปใช้ใน public-private key encyption

4.5 ลายมือชื่อดิจิตอล

ลายมือชื่อดิจิตอลใช้เมื่อต้องการความมั่นใจในแหล่งที่มาของเอกสาร เปรียบเสมือนลายมือชื่อซึ่งเฉพาะเจ้าของจริงที่สามารถคำนวณขึ้นได้ แต่ลายมือชื่อนี้สามารถพิสูจน์ได้ กล่าวคือบุคคลอื่นสามารถตรวจสอบได้ว่าลายมือชื่อนั้นมาจากผู้สร้างจริงๆ วิธีที่ใช้ในการทำลายมือชื่อดิจิตอลที่มีความปลอดภัยค่อนข้างสูงก็คือ การใช้ public-private key โดยเข้ารหัสค่าลายมือชื่อด้วยคีย์ส่วนตัว และคนอื่นสามารถใช้คีย์สาธารณะพิสูจน์ได้ว่าลายมือชื่อมาจากคีย์ส่วนตัวที่ตรงกัน

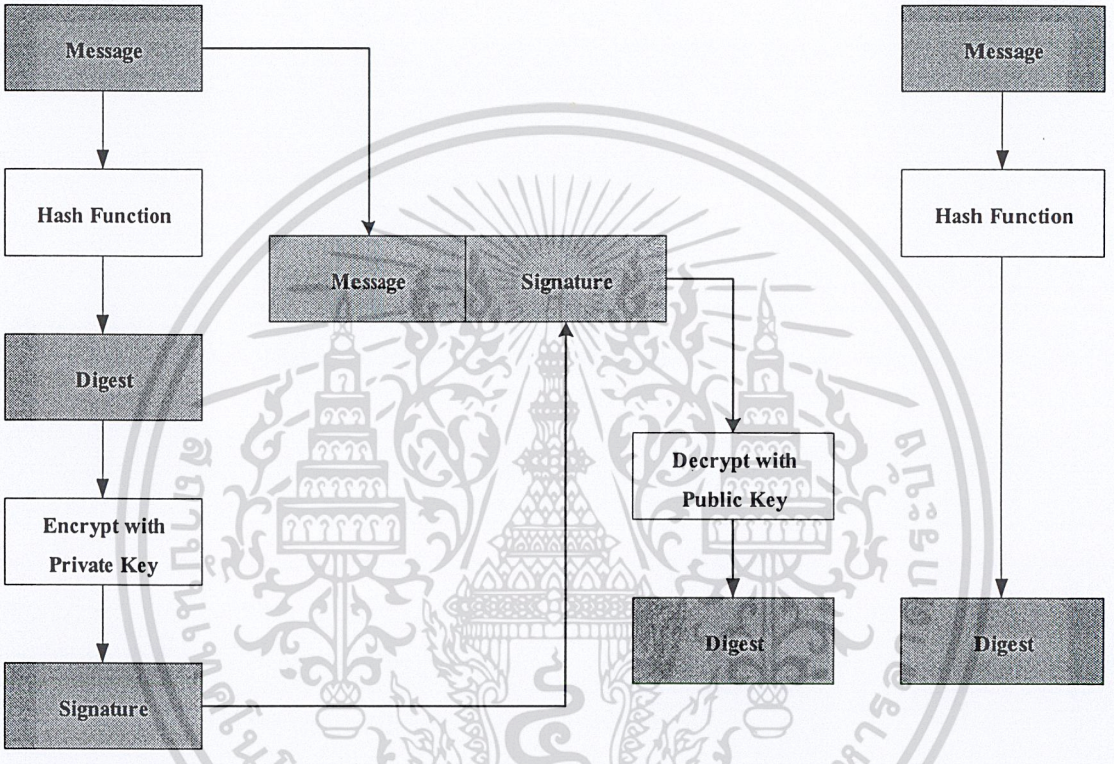
คุณสมบัติที่สำคัญของลายมือชื่อดิจิตอลที่สำคัญนั้นจะต้องประกอบด้วย 2 ประการคือ

- สามารถยืนยันได้ว่าข้อมูลที่รับมานั้นไม่มีการเปลี่ยนแปลงระหว่างการส่ง
- สามารถยืนยันได้ว่าข้อมูลนั้นได้รับการยืนยันจากผู้ลงลายมือชื่อจริงๆ

การทำลายมือชื่อดิจิตอลนั้นทำได้โดยนำข้อความแรกเริ่มไปเข้าแฮชฟังก์ชันจะได้เป็นเมสเซจไดเจสต์เพื่อเป็นการยืนยันความถูกต้องของข้อมูลก่อน จากนั้นจะนำเมสเซจไดเจสต์ที่ได้ส่งไปเข้ารหัสกับคีย์ส่วนตัวของผู้ส่งเพื่อเป็นการยืนยันว่าเป็นเจ้าของเอกสารนั้นจริง และจะได้สิ่งที่เรียกว่าลายมือชื่อดิจิตอลออกมาส่งไปพร้อมกับข้อมูลด้วย ทางด้านรับนั้นจะเป็นวิธีการที่คล้ายกับด้านส่งโดยจะต้องเริ่มจากการหาเมสเซจไดเจสต์ของข้อมูลที่ได้รับมาเก็บไว้ก่อน จากนั้นจะนำลายมือชื่อดิจิตอลที่ได้รับมาด้วย มาทำการถอดรหัสด้วยคีย์สาธารณะของผู้ส่ง ในขั้นตอนนี้เองที่จะสามารถยืนยันบุคคลผู้ส่งได้ เพราะถ้าหากผู้ที่ลงลายมือชื่อไม่ได้เป็นบุคคลที่เราคาดว่าเป็นเจ้าของ ก็จะไม่สามารถถอดลายมือดิจิตอลได้ ถ้าหากการยืนยัน

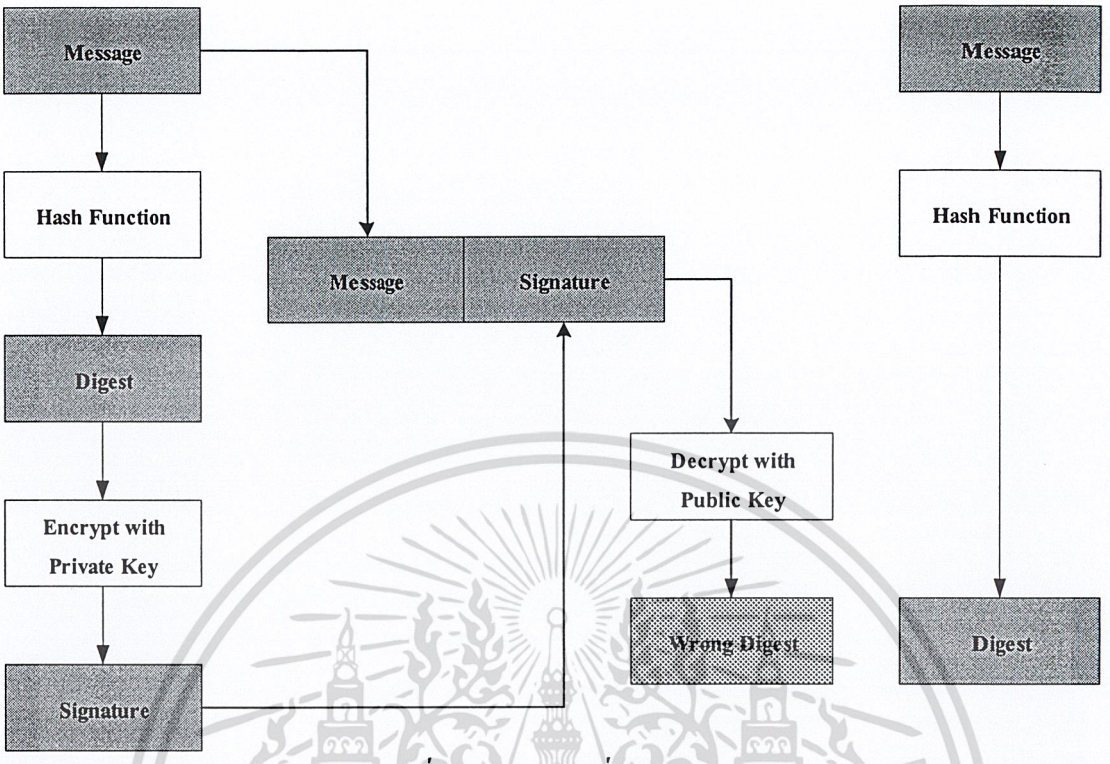
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บุคคลผู้ส่งสำเร็จแล้วจะได้รับเมสเซจไดเจสต์ของข้อมูลก่อนจะส่งออกมา ด้านรับจะต้องทำการตรวจสอบความถูกต้องของข้อมูลโดยการนำเมสเซจไดเจสต์ที่คำนวณได้ในตอนรับข้อมูล กับเมสเซจไดเจสต์ที่ได้หลังจากการถอดรหัสลายมือชื่อดิจิตอลออกมา นำมาเปรียบเทียบกัน ถ้าหากเมสเซจไดเจสต์ที่ได้ออกมามีค่าเท่ากันนั่นแสดงว่าข้อมูลที่ได้รับนั้นเป็นข้อมูลเดียวกันจากทางด้านส่ง และไม่มีการเปลี่ยนแปลงข้อมูลระหว่างการส่ง ขั้นตอนการส่ง และการรับของวิธีการนี้ แสดงไว้ในรูปที่ 4-4 โดยในกรณีที่มีการปลอมแปลงจะเป็นดังรูปที่ 4-5



รูปที่ 4-4 แสดงวิธีการทำลายมือชื่อโดยใช้แฮชฟังก์ชัน และคีย์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 แสดงกรณีที่มีผู้แอบอ้าง

เมื่อมีผู้แอบอ้าง ผู้แอบอ้างไม่รู้คีย์ส่วนตัวของผู้ส่งจริงๆ จึงใช้คีย์ของตัวเอง ซึ่งไม่เข้าคู่กับคีย์สาธารณะทางฝั่งผู้รับ ทำให้ทางฝั่งผู้รับไม่สามารถถอดรหัสลายมือชื่อดิจิตอลได้ จึงรู้ว่าไม่ใช่ผู้ส่งตัวจริงเป็นคนส่ง

4.6 SSL Protocol

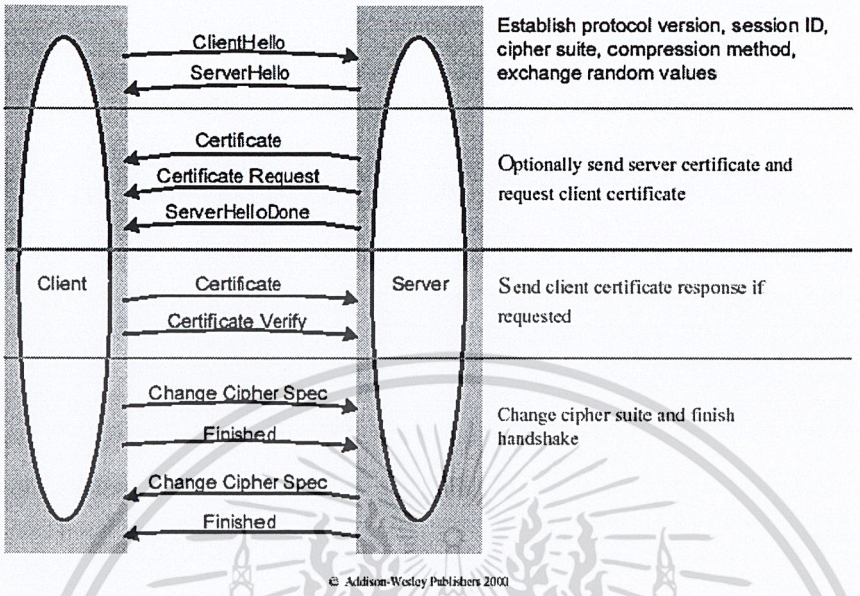
SSL Protocol สร้างความปลอดภัยให้กับแอพพลิเคชั่นที่ทำการติดต่อผ่านระบบเครือข่ายซึ่งมีรายละเอียดดังนี้

- กลไกในการที่แต่ละแอพพลิเคชั่นสามารถใช้ยืนยันหลักฐานของแอพพลิเคชั่นอื่นได้
- เข้ารหัสข้อมูลที่ใช้ในการแลกเปลี่ยนระหว่างแอพพลิเคชั่น

เมื่อ SSL Protocol ถูกใช้ ผู้รับการติดต่อจะทำการยืนยันตัวเองกับผู้เริ่มการติดต่อ ถ้าผู้รับการติดต่อร้องขอการติดต่อ ผู้เริ่มการติดต่อจะสามารถยืนยันตัวเองกับผู้รับการติดต่อได้ SSL connection จะเริ่มด้วยการ handshake ระหว่างแอพพลิเคชั่นจะ แลกเปลี่ยนเอกสารสิทธิ์ (digital certificate) กัน ตกลงอัลกอริทึมในการเข้ารหัสที่จะใช้ และ สร้างคีย์ที่ใช้ในการเข้ารหัสสำหรับใช้ตลอดช่วงการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

♦ **SSL Protocol Handshake Phases**



รูปที่ 4-6 แสดงทรีเวย์แฮนด์เชค

4.7 เอกสารสิทธิ์ (Digital Certificates)

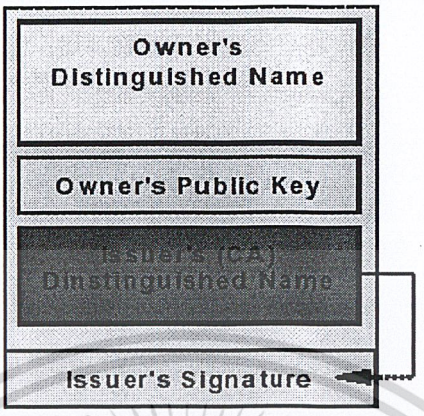
ลักษณะของเอกสารสิทธิ์ดิจิตอล

เอกสารสิทธิ์เป็นเสมือนกับบัตรประจำตัวประชาชน ซึ่งจะบ่งบอกรายละเอียดของบุคคล เพื่อใช่ในการยืนยันตัวตน เราสามารถส่งเอกสารสิทธิ์ไปควบคู่กับลายมือชื่อดิจิตอลเพื่อใช่ในการยืนยันว่าเอกสารสิทธิ์นั้นไม่มีการเปลี่ยนแปลง การสร้างเอกสารสิทธิ์ทำโดยผู้ใช้ทุกคนทำการขอเอกสารสิทธิ์กับองค์กรพิสูจน์สิทธิ์ (Certificate Authority : CA) โดยการส่งคีย์สาธารณะและข้อมูลตามที่องค์กรพิสูจน์สิทธิ์นั้นกำหนดไปและทำการขอเอกสารสิทธิ์มา การติดต่อต้องทำเป็นการส่วนตัวหรือติดต่อผ่านระบบการพิสูจน์ตัวตนที่ปลอดภัย เราสามารถกำหนดสิ่งที่จำเป็นในการสื่อสารได้ดังต่อไปนี้

1. ผู้ใช้ทุกคนสามารถคำนวณหาชื่อและคีย์สาธารณะของเจ้าของเอกสารสิทธิ์ได้
2. ผู้ใช้ทุกคนสามารถตรวจสอบได้ว่าเอกสารสิทธิ์มาจากองค์กรพิสูจน์สิทธิ์จริงๆไม่ได้ถูกปลอมแปลงมา
3. ผู้ใช้สามารถตรวจสอบได้ว่าเอกสารสิทธิ์นั้นหมดอายุหรือไม่
4. ผู้ที่สามารถอ้างและอัพเดต(update)เอกสารสิทธิ์ได้มีเพียงองค์กรมีอำนาจในการรับรองสิทธิ์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ผู้ใช้ทุกคนสามารถตรวจสอบเอกสารสิทธิ์ได้เป็นประจำ



รูปที่ 4-7 แสดงส่วนประกอบต่างๆของเอกสารสิทธิ์

ความสำคัญของเอกสารสิทธิ์

ในการเข้ารหัสแบบคีย์ต่างๆนั้นเราจะต้องมีการสร้างทั้งคีย์ส่วนตัวและคีย์สาธารณะ ซึ่งโดยทั่วไปการสร้างคีย์จะทำโดยโปรแกรมที่จะใช้คีย์นั้น เช่น โปรแกรมเว็บบราวเซอร์หรือโปรแกรมติดต่ออิเล็กทรอนิกส์เมล์ หลังจากที่สร้างคีย์ทั้งสองเรียบร้อยแล้วการเก็บรักษาคีย์เป็นเรื่องสำคัญ เราจะต้องเก็บรักษาคีย์ส่วนตัวไว้ให้ดีอย่าให้ใครแอบมาเห็นหรือขโมยไปได้ จากนั้นจะเป็นการตัดสินใจว่าจะทำการแจกจ่ายคีย์สาธารณะของเราไปสู่ผู้อื่นด้วยวิธีใด เช่นการแจกคีย์โดยส่งอิเล็กทรอนิกส์เมล์ไปให้เพื่อนหรือบุคคลที่ติดต่อกับเรา แต่วิธีนี้เราอาจส่งคีย์ไปให้ไม่ได้ครบทุกคน และยังต้องเป็นภาระการจัดส่งคีย์ให้กับบุคคลใหม่ๆที่ต้องการติดต่อด้วย นอกจากนี้ยังไม่สามารถทำให้ผู้รับมั่นใจได้ว่าคีย์ที่ส่งไปนั้นเป็นคีย์ของเราจริงๆ เนื่องจากอาจมีผู้อื่นแอบสร้างคีย์โดยใช้ชื่อเราและแอบอ้างส่งคีย์ดังกล่าวให้กับผู้อื่นเพื่อให้เข้าใจว่าเป็นคีย์ของเราได้

วิธีที่ดีกว่าและใช้อยู่ในปัจจุบันก็คือการใช้ระบบแจกจ่ายคีย์ที่เชื่อถือได้โดยจะมีองค์กรที่ทำหน้าที่เฉพาะเป็นองค์กรที่สาม(Third Party) ในการรับรองและระงับการรับรองคีย์ที่เรียกว่าองค์กรพิสูจน์สิทธิ์(Certificate Authority - CA) โดยองค์กรพิสูจน์สิทธิ์นี้จะตรวจสอบคีย์สาธารณะของเราด้วยหลักฐานว่าคีย์ของเรานั้นๆเป็นของจริง พร้อมทั้งตรวจสอบข้อมูลส่วนตัวของเรา(ข้อมูลที่ตรวจสอบจะมากน้อยแค่ไหนขึ้นอยู่กับระดับชั้นของการรับรอง) เมื่อผู้อื่นได้รับคีย์ของเราก็สามารถที่จะตรวจสอบกับผู้ออกเอกสารสิทธิ์นี้ว่าคีย์ที่ได้รับเป็นของจริงหรือไม่ ซึ่งตัวเอกสารสิทธิ์นี้จะเปรียบเสมือนบัตรประชาชนดิจิตอลของเราที่ใช้บอกได้ว่าเราเป็นบุคคลที่อ้างจริงๆในระบบเครือข่ายหรือการส่งข้อมูลทางอิเล็กทรอนิกส์

ในปัจจุบันบริษัทที่ออกเอกสารสิทธิ์ดิจิตอล(Digital Certificate) คือบริษัท Verisign , Cyvertrust , Global Sign และ Nortel โดยในเอกสารสิทธิ์ดิจิตอลจะประกอบด้วยข้อมูลต่างๆ ได้แก่ ชื่อของผู้ถือเอกสารสิทธิ์ ชื่อของบริษัทผู้ออกเอกสารสิทธิ์ คีย์สาธารณะของผู้ถือเอกสารสิทธิ์ วันหมดอายุของผู้ถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารสิทธิ์(โดยทั่วไปจะกำหนดระยะเวลา 6 เดือนหรือ 1 ปี) ระดับชั้นของเอกสารสิทธิ์ และเลขหมายของตัวเอกสารสิทธิ์ดิจิตอล

เอกสารสิทธิ์ดิจิตอลแบ่งออกได้เป็นสี่ระดับชั้นตามระดับการตรวจสอบข้อมูลของเจ้าของเอกสารสิทธิ์

1. ระดับที่หนึ่ง เป็นชั้นที่ออกเอกสารสิทธิ์ได้ง่ายที่สุดเนื่องจากมีการตรวจสอบน้อยที่สุด โดยจะตรวจแค่ชื่อผู้ถือเอกสารสิทธิ์ และที่อยู่อิเล็กทรอนิกส์เมล์(e-mail address) ว่าถูกต้องจริงเท่านั้น
2. ระดับชั้นที่สอง จะตรวจสอบเลขประจำตัวประชาชน เลขประจำตัวของระบบสวัสดิการหรือประกันสังคม(social security number) และวันเดือนปีเกิด
3. ระดับชั้นที่สาม จะมีการตรวจสอบเพิ่มเติมเกี่ยวกับประวัติการใช้เครดิต และการชำระเงิน
4. ระดับชั้นที่สี่ ยังไม่มีการออกมาเป็นมาตรฐานอย่างแน่ชัด แต่จะเป็นการตรวจสอบข้อมูลเพิ่มเติมเกี่ยวกับตำแหน่งงานในองค์กรด้วย

ในการขอเอกสารสิทธิ์นั้นจะต้องกระทำบนกระบวนการที่ปลอดภัย ซึ่งในปัจจุบันมี 2 วิธีคือ

1. การขอเอกสารสิทธิ์ผ่านโปรแกรมประเภทบราวเซอร์ เช่น อินเตอร์เน็ตเอ็กซ์พลอเรอร์, เนสเคปท์ เป็นต้น โดยผู้ใช้เข้าไปยังโฮสต์ที่ให้บริการการขอเอกสารสิทธิ์หรือเว็ปไซต์ที่เปิดให้บริการดังกล่าวอยู่ ตัวโปรแกรมประเภทบราวเซอร์จะทำการสร้างคู่คีย์ คีย์ส่วนตัวจะเก็บไว้ที่เครื่องของเรา จากนั้นจะส่งคีย์สาธารณะและข้อมูลส่วนตัวของเราไปยังโฮสต์หรือไซต์ที่เป็นขององค์กรพิสูจน์ จากนั้นองค์กรพิสูจน์สิทธิ์จะทำการรับรองข้อมูลแล้วจึงส่งเอกสารสิทธิ์กลับมาที่เครื่องของเรา
2. การขอเอกสารสิทธิ์อีกรูปแบบหนึ่งคือ การส่งเอกสารสิทธิ์ที่ไม่ผ่านระบบเครือข่าย กล่าวคือ ผู้ขอต้องไปทำการขอที่สำนักงานขององค์กรพิสูจน์สิทธิ์โดยตรง โดยทำเรื่องของเอกสารสิทธิ์และกรอกข้อมูลส่วนตัว องค์กรพิสูจน์สิทธิ์จะทำการรับรองและบันทึกข้อมูลของผู้ขอแล้วทำการส่งเอกสารสิทธิ์พร้อมทั้งคีย์ส่วนตัวให้กับผู้ขอในรูปแบบของไฟล์ในแผ่นดิสก์ให้ผู้ขอนำไปบันทึกในเครื่องของตนเองต่อไป

บริการพิสูจน์สิทธิ์แบบ X.509 (X.509 Authentication Service)

ระบบ X.509 เป็นระบบพิสูจน์สิทธิ์ที่สำคัญมากในระบบเครือข่าย โดย X.509 เป็นอนุกรมย่อยของ X.500 ซึ่งกำหนดมาตรฐาน ITU-T โดยขณะที่ X.500 เป็นตัวกำหนดโครงสร้างในลักษณะที่เป็นไดเรกทอรีหรือโครงสร้างนั้น X.509 จะทำหน้าที่ในการพิสูจน์สิทธิ์ให้กับส่วนต่าง ๆ ของไดเรกทอรี่นั้น สำหรับรูปแบบการใช้งานจะเน้นไปที่การพิสูจน์บุคคล เพื่อยืนยันการติดต่อเป็นสำคัญ

การทำงานของ X.509 จะมีโครงสร้างการทำงานที่เป็นไดเรกทอรี่ โดยในที่นี้ไดเรกทอรี่

เอกสารนี้เป็น... ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำหน้าที่เป็นที่เก็บข้อมูลที่ใช้ในการยืนยัน ซึ่งโดยทั่วไปจะอยู่ในรูปของเอกสารสิทธิ์ซึ่งในเอกสารสิทธิ์จะบรรจุคีย์สาธารณะของผู้ใช้ที่เข้ารหัสโดยคีย์ส่วนตัวขององค์กรที่จ่ายใบเอกสารสิทธิ์มาให้ สำหรับการทำงานของ X.509 นั้นจะมีขอบเขตการนำไปใช้งานที่กว้างขวางมาก เช่น ใช้ในการทำ Mail Security ใช้ในการทำ IP Security ใช้ในการทำ Web Security หรือหากจะกล่าวว่า เมื่อใดที่ต้องการพิสูจน์บุคคลหรือยืนยันเครื่องคอมพิวเตอร์แล้ว ก็มักจะอยู่ในขอบข่ายการทำงานของ X.509 เสมอ

X.509 ได้ถูกนำเสนอเมื่อปี 1988 จากนั้นได้ผ่านการปรับปรุงเป็นลำดับขั้น ในประเด็นต่าง ๆ รวมทั้งเรื่องความปลอดภัยด้วย จากนั้นก็ได้ออกมาเป็นข้อเสนอที่ปรับปรุงแล้วในปี 1993 และปรับปรุงอีกครั้งในปี 1995 โดยการทำงานของ X.509 จะใช้การเข้ารหัสแบบคีย์สาธารณะแล้วใช้มาตรฐานลายมือชื่อดิจิตอลในการรับรองข้อมูล สำหรับอัลกอริทึมนั้น ไม่ได้ระบุแน่นอนโดยสามารถเลือกใช้ได้หลายตัวแต่ที่แนะนำคือ RSA

รูปแบบทั่วไปของเอกสารสิทธิ์

- เวอร์ชัน (Version) แสดงหมายเลขเวอร์ชัน เพราะในแต่ละเวอร์ชันจะมีรูปแบบของข้อมูลที่ไม่เหมือนกันก็ได้ โดยปกติจะเป็นเวอร์ชัน 1 แต่หากในเอกสารสิทธิ์มีการใช้
- หมายเลขลำดับ (Serial Number) เป็นเลขจำนวนเต็ม โดยจะต้องไม่ซ้ำกันในองค์กรที่ออกเอกสารสิทธิ์ โดยเลขนี้จะเป็นเลขที่จะใช้อ้างถึงแต่ละเอกสารสิทธิ์ที่สร้างขึ้นมา
- อัลกอริทึมที่ใช้สร้าง (Signature Algorithm Identifier) เป็นฟิลด์ที่ระบุอัลกอริทึมที่ใช้ในการสร้างเอกสารสิทธิ์
- ชื่อผู้ออกเอกสารสิทธิ์ (Issue Name) เป็นชื่อขององค์กรที่ออกเอกสารสิทธิ์
- ช่วงเวลาที่รับรองเอกสารสิทธิ์ (Period of Validity) เป็นตัวบอกว่า ให้ใช้เอกสารสิทธิ์นี้ตั้งแต่วันที่เท่าไหร่และสิ้นสุดวันที่เท่าไหร่
- ชื่อเจ้าของเอกสารสิทธิ์ (Subject Name) เป็นชื่อของบุคคลที่เอกสารสิทธิ์ใบนี้อ้างถึงหรือแทนตัวบุคคลนั้น
- ข้อมูลของคีย์สาธารณะ (Subject's Public Key Information) เป็นฟิลด์ที่เก็บคีย์สาธารณะและระบุถึงอัลกอริธึมที่ใช้กับคีย์นี้ขึ้นมา รวมถึงพารามิเตอร์อื่น ๆ ด้วย
- ตัวระบุผู้ออกเอกสารสิทธิ์ (Issuer Unique Identifier) เป็นฟิลด์ออปชันที่ใช้ในการระบุถึงองค์กรที่ออกเอกสารสิทธิ์ ในกรณีที่ชื่อ X.500 มีการนำไปใช้กับส่วนอื่น ๆ
- ตัวระบุผู้ขอเอกสารสิทธิ์ (Subject Unique Identifier) เป็นฟิลด์ที่ใช้ในการระบุถึงตัวบุคคลที่เป็นเจ้าของเอกสารสิทธิ์ ในกรณีที่ชื่อ X.500 มีการนำไปใช้กับส่วนอื่น ๆ
- ส่วนขยาย (Extension) เป็นกลุ่มของฟิลด์ที่เพิ่มเติมข้อมูลอื่น ๆ เข้ามาด้วย
- ลายมือชื่อ (Signature) จะบรรจุเมสเซจไดเจสต์ของข้อมูลในทุกฟิลด์ที่เข้ารหัสด้วยคีย์ส่วนตัวขององค์กรพิสูจน์สิทธิ์เพื่อเป็นการยืนยันว่า เอกสารสิทธิ์นี้สร้างมาจากองค์กรดังกล่าวจริง ๆ โดยจะมีข้อมูลที่ระบุวิธีการแฮชและวิธีการเข้ารหัสด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

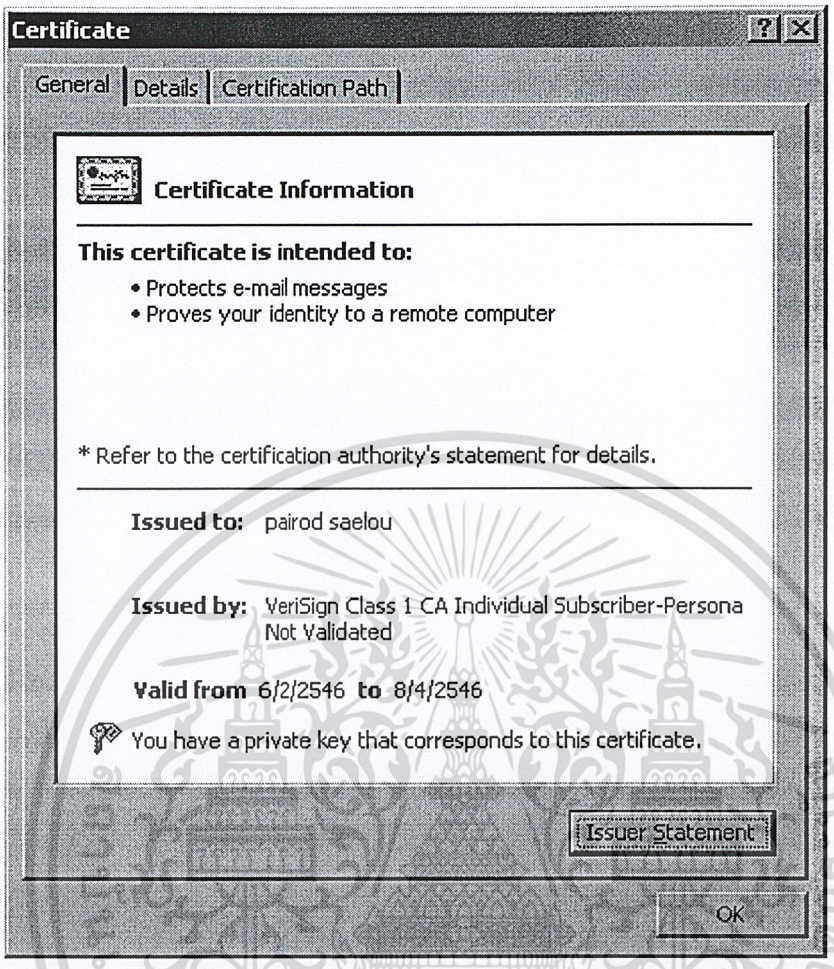
ในการใช้งานเอกสารสิทธิ์จะมีช่วงเวลาใช้งานที่จำกัดแน่นอน ดังนั้นหากผู้ใช้ต้องการใช้เอกสารสิทธ์ต่อไป ก็ต้องขอต่ออายุเอกสารสิทธิ์ก่อนที่จะหมดอายุ แต่หากจะมีการหมดอายุโดยที่ไม่ขอต่อหรือมีการเลิกใช้เอกสารสิทธิ์อาจจะเนื่องมาจากพนักงานลาออก หรืออาจจะเนื่องจากเอกสารสิทธินี้ไม่ปลอดภัยแล้วก็ต้องทำการเรียกคืน (Revoke) และองค์กรพิสูจน์สิทธิ์จะต้องมีการจัดทำรายการเอกสารสิทธิ์ที่ถูกเรียกคืน (Certificate Revocation List - CRL) ซึ่งจะเก็บไว้ในไดเรกทอรีและรับรองโดยองค์กรพิสูจน์สิทธิ์ ซึ่งผู้ที่ต้องการตรวจสอบเอกสารสิทธิ์ว่าเป็นเอกสารสิทธิ์ที่ไม่ใช้งานแล้วหรือไม่ ก็ต้องขอรายการเอกสารสิทธิ์ที่ถูกเรียกคืนไปตรวจสอบ

และเนื่องจากเอกสารสิทธิ์ไม่สามารถปลอมได้ ดังนั้นการเก็บเอกสารสิทธิ์ไว้ที่องค์กรพิสูจน์สิทธิ์จึงไม่ต้องมีกลไกพิเศษมาป้องกันแต่อย่างใด กล่าวคือผู้ใช้คนใดที่เป็นสมาชิกขององค์กรพิสูจน์สิทธิ์ก็สามารถเข้าถึงเอกสารสิทธ์ของผู้ใช้คนอื่น ๆ ได้ทุกคน โดยเอกสารสิทธิจะเก็บอยู่ในไฟล์เพียงไฟล์เดียวที่มีขนาดเล็ก นอกจากจะสามารถขอเอกสารสิทธ์จากองค์กรพิสูจน์สิทธิ์แล้วผู้ใช้ยังสามารถส่งเอกสารสิทธ์ไปให้กันเองได้อีกด้วย โดยผ่านทางสื่อต่าง ๆ เช่น จดหมายอิเล็กทรอนิกส์, ส่งผ่านแผ่นดิสก์เก็ต เป็นต้น

อย่างไรก็ตาม เนื่องจากระบบเครือข่ายในปัจจุบันมีขนาดใหญ่โตกว้างขวางมาก และการติดต่อสื่อสารก็ไม่ได้มีลักษณะเฉพาะกลุ่มอีกแล้ว ดังนั้นการที่จะให้ผู้ใช้ทุกคนมาใช้เอกสารสิทธิ์ที่รับรองโดยองค์กรพิสูจน์สิทธิ์เดียวกันทั้งหมดก็อาจเป็นเรื่องยาก หากผู้ใช้ 2 คนใช้เอกสารสิทธิ์ที่รับรองจากองค์กรพิสูจน์สิทธิ์คนละแห่งก็จะไม่สามารถตรวจสอบเอกสารสิทธิ์ของอีกฝ่ายได้ว่าเป็นฉบับจริงหรือไม่ ซึ่งในกรณีเช่นนี้ ก็อาจจะใช้วิธีสำเนาคีย์สาธารณะขององค์กรพิสูจน์สิทธิ์ของผู้ใช้อีกคนหนึ่งมาทำการตรวจสอบเองก็สามารถทำได้ เช่น กำหนดให้มี CA-A และ CA-B โดยให้บริการกับผู้ใช้ A และ B แต่เนื่องจากในครั้งแรกที่ A สำเนาคีย์สาธารณะของ CA-B มานั้นอาจเกิดการปลอมแปลงได้ เพราะสิ่งที่เรามีอยู่ก็คือคีย์สาธารณะของ CA-A ของเรา แต่เอกสารสิทธิ์ของ CA-B ทำให้เราไม่สามารถตรวจสอบว่าเอกสารสิทธิ์ที่ได้รับมานั้นเป็นฉบับที่ถูกต้องหรือไม่ ดังนั้นวิธีดังกล่าวจึงถือว่า ไม่มีความปลอดภัยเพียงพอ

สำหรับอีกวิธีการอีกแบบ คือให้ CA-A เก็บเอกสารสิทธิ์ของ CA-B เอาไว้ด้วยและ CA-B ก็เก็บเอกสารสิทธิ์ของ CA-A เอาไว้เช่นกัน ด้วยวิธีนี้เราก็สามารถให้องค์กรพิสูจน์สิทธิ์ตรวจสอบเอกสารสิทธิ์ได้ไม่ว่าเอกสารสิทธิ์นั้นจะรับรองจาก CA-A หรือ CA-B ก็ตาม เช่น ผู้ใช้ A ต้องการตรวจสอบเอกสารสิทธิ์ที่รับรองจาก CA-A ก็สามารถทำได้เลยเพราะรู้คีย์สาธารณะของ CA-A อยู่แล้วเนื่องจากเป็นสมาชิก CA-A และหากผู้ใช้ A ต้องการตรวจสอบเอกสารสิทธิ์ที่รับรองโดย CA-B ผู้ใช้ A ก็ขอเอกสารสิทธิ์ของ CA-B จาก CA-A โดยเอกสารสิทธิ์ดังกล่าวจะรับรองโดย CA-A ดังนั้นจึงแน่ใจได้ว่าเอกสารสิทธิ์ของ CA-B ที่ได้รับนั้นเป็นของจริงและคีย์สาธารณะของ CA-B ก็เป็นของจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 แสดงเอกสารสิทธิ์ที่ตรวจสอบโดยบราวเซอร์

จากนั้นจึงนำเอาคีย์สาธารณะของ CA-B ไปตรวจสอบเอกสารสิทธิ์อีกทีก็จะทราบได้ว่าเอกสารสิทธิ์นั้นเป็นของจริงหรือไม่ และนอกเหนือจาก CA-A และ CA-B แล้วการเชื่อถือ(Trust)กันเช่นนี้ยังสามารถกระทำกับองค์กรพิสูจน์สิทธิ์อื่นๆไปเรื่อยๆ อย่างไรก็ตามหากองค์กรพิสูจน์สิทธิ์มีเป็นจำนวนมากแล้ว ก็มีความจำเป็นที่จะต้องจัดโครงสร้างการเชื่อถือกันขององค์กรพิสูจน์สิทธิ์ให้เป็นระบบ ไม่เช่นนั้นก็อาจมีการเชื่อถือกันแบบยุ่งเหยิงและทำให้การทำงานเป็นไปอย่างไม่มีประสิทธิภาพได้ ซึ่งมาตรฐาน X.509 ก็ได้แนะนำให้มีการใช้ระบบเชื่อถือกันในรูปของความสัมพันธ์แบบระดับชั้น(Hierachial)

องค์กรพิสูจน์สิทธิ์

เอกสารสิทธิ์จะถูกออกโดยองค์กรพิสูจน์สิทธิ์ที่ซึ่งเป็นองค์กร(บุคคลที่สาม)ที่เชื่อถือได้ เมื่อ องค์กรพิสูจน์สิทธิ์สร้างเอกสารสิทธิ์แล้ว จะทำการเข้ารหัสด้วยคีย์ส่วนตัวขององค์กรพิสูจน์สิทธิ์นั้น แล้วจะทำการส่งไปให้กับผู้ที่ขอเอกสารสิทธิ์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยผู้ขอเอกสารสิทธิ์จะยืน ยัน ตัวขององค์กรพิสูจน์สิทธิ์นั้น ด้วยคีย์สาธารณะซึ่งถูกออกด้วย high-level certificate authority ซึ่งมีความน่าเชื่อถือกว่า

4.8 การอิมพลีเมนต์ Security ในเว็บลอจิก

เว็บลอจิกจะมีส่วนของ authentication, authorization และการเข้ารหัสให้บริการเพื่อที่จะป้องกันความปลอดภัยในทรัพยากรของคุณ แต่อย่างไรก็ดีวิธีการเหล่านี้ก็ยังไม่สามารถแก้ปัญหาในกรณีที่มีคนค้นพบช่องโหว่ หรือ จุดอ่อน ในสภาพแวดล้อมการดีพลอย (deployment environment) ของคุณได้

ทางที่ดีคือ หลังจากที่คุณดีพลอยเว็บลอจิกเซิร์ฟเวอร์แล้ว คุณควรจะให้ผู้ที่เชี่ยวชานในด้านความปลอดภัยมาตรวจสอบ และแนะนำในการปรับปรุงให้มีความปลอดภัยยิ่งขึ้น นอกจากนี้คุณอาจจะติดตามวารสารต่างๆที่เกี่ยวกับความปลอดภัย ,ตรวจสอบเรื่องของบั๊กและคอยปรับปรุง service pack ต่างๆ

4.8.1 Security Realms

Security Realms ประกอบไปด้วยกลไกในการป้องกันทรัพยากรของเว็บลอจิกเซิร์ฟเวอร์โดย resorce ของเว็บลอจิกเซิร์ฟเวอร์จะถูกป้องกันภายใต้ Security realms เพียงอันเดียวเท่านั้น และโดยนโยบายหนึ่งเดียวของ Security realms เมื่อ user ต้องการที่จะเข้าถึงทรัพยากรใดๆที่เป็นของ realms นั้นๆ เซิร์ฟเวอร์จะพยายามที่จะพิสูจน์ตัวตนของผู้ใช้คนนั้น

4.8.2 ผู้ใช้

คือเอ็นติตี้ที่มาเรียกใช้ เว็บลอจิกเซิร์ฟเวอร์เช่นแอพพลิเคชั่น end users หรือเว็บลอจิกเซิร์ฟเวอร์ด้วยกันเอง

เมื่อผู้ใช้ต้องการที่จะเข้าถึงเว็บลอจิกเซิร์ฟเวอร์ ผู้ใช้คนนั้นจะต้องแสดงหลักฐาน(เช่น password หรือ digital certificate) ผ่าน JAAS เพื่อที่จะ authentication ต่อ security realm เมื่อเว็บลอจิกเซิร์ฟเวอร์สามารถยืนยันผู้ใช้คนนั้นได้จะทำการเข้าไปมีส่วนร่วมกับผู้ใช้ด้วย thead ที่ทำการ execute code ของผู้ใช้แต่ก่อนที่ thread จะเริ่มทำการ execute code เซิร์ฟเวอร์จะทำการตรวจสอบบทบาท และสิทธิของผู้ใช้อีกทีเพื่อยืนยันว่าผู้ใช้สามารถทำต่อได้

การระบุผู้ใช้ในเว็บลอจิกนั้น คุณสามารถใช้การระบุรหัสผ่านสำหรับผู้ใช้นั้นๆ โดยเว็บลอจิก จะทำการแฮชทุกรหัสผ่าน หลังจากได้รับการร้องขอรหัสผ่านจากไคลเอ็นต์จะถูกแฮชและ เว็บเซิร์ฟเวอร์จะทำการเปรียบเทียบกับรหัสผ่านที่ถูกแฮชแล้ว

4.8.3 Group

Group คือ กลุ่มของผู้ใช้ การจัดการกับ group นั้นมีประสิทธิภาพมากกว่าการจัดการเป็นทีละผู้ ใช้ยกตัวอย่างเช่น administrator สามารถกำหนดสิทธิ (permission) สำหรับ 50 ผู้ใช้ได้ในครั้งเดียวโดยการกำหนดบทบาท และสิทธิให้กับ group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8.4 Access Control Mechanisms

กลไก 2 แบบที่มีอยู่ในเว็บลอจิกเซิร์ฟเวอร์ซึ่งทำการด้วยกันเพื่อที่จะควบคุม และป้องกันการเข้าถึงทรัพยากร คือ Security Policies และ Roles

SSL Protocol กำหนดคุณสมบัติของ security ดังนี้

- Server authentication – เว็บลอจิกเซิร์ฟเวอร์ใช้ เอกสารสิทธิ์ของมันเพื่อที่จะใช้ในการยืนยันกับไคลเอ็นต์
- Client Identify Verification – ไคลเอ็นต์ก็จะทำการยืนยันตัวเองกับเว็บลอจิกเซิร์ฟเวอร์โดยใช้เอกสารสิทธิ์ของมันเอง โดยเซิร์ฟเวอร์จะทำการตรวจสอบเอกสารสิทธิ์ของไคลเอ็นต์และ เริ่ม SSL connection ถ้าเอกสารสิทธิ์ได้รับการยืนยันแล้วว่าถูกต้อง วิธีนี้เรียกว่า two-way SSL เนื่องจากผู้ใช้มีได้หลายคนดังนั้น ไคลเอ็นต์ต้องใช้ทั้ง username และรหัสผ่าน(or เอกสารสิทธิ์) ในการ authentication และ authorization
- Confidentiality – ทั้งการร้องขอของไคลเอ็นต์และการตอบสนองของเซิร์ฟเวอร์จะถูกเข้ารหัสเพื่อที่จะทำให้ การแลกเปลี่ยนข้อมูลเป็นส่วนตัวมากขึ้นตลอดเส้นทางในระบบเครือข่าย
- Data Integrity – ข้อมูลที่ส่งระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์จะถูกป้องกันจากกลุ่มบุคคลที่3

4.8.5 Authentication Mechanisms

ผู้ใช้จะต้องทำการยืนยันตัวเองทุกครั้งที่ร้องขอ ผู้ใช้ต้องการที่จะเข้าถึงทรัพยากรที่ถูกป้องกันไว้โดยเว็บลอจิกเซิร์ฟเวอร์ด้วยเหตุนี้เอง ผู้ใช้จะต้องมีหนังสือแนะนำตัว(username/password หรือเอกสารสิทธิ์) ต่อ เว็บลอจิกเซิร์ฟเวอร์ ซึ่งเว็บลอจิกเซิร์ฟเวอร์จะมีกลไกในการยืนยันตัวตน 2 แบบ คือ

- Username/Password authentication – user ID และรหัสผ่านจากไคลเอ็นต์จะถูกส่งไปยังเซิร์ฟเวอร์ เพื่อที่จะทำการตรวจสอบข้อมูลที่มี และถ้าถูกต้องก็จะอนุญาตให้เข้าถึงทรัพยากรได้ SSL(หรือ HTTPS) โปรโตคอลจะสามารถถูกใช้เพื่อเพิ่มระดับของความปลอดภัยในการส่ง username และรหัสผ่านเนื่องจากโปรโตคอลนี้ จะทำการเข้ารหัสข้อมูลที่ถูกส่งระหว่างไคลเอ็นต์และเซิร์ฟเวอร์
- Certificate authentification – เมื่อ SSL หรือ HTTPS ผู้ใช้เริ่มต้นการร้องขอ เว็บลอจิกเซิร์ฟเวอร์จะตอบกลับโดยส่งเอกสารสิทธิ์ ของมันไปให้ไคลเอ็นต์ โดยไคลเอ็นต์จะทำการตรวจสอบ แล้วส่งเอกสารสิทธิ์ของมัน เซิร์ฟเวอร์จะทำการตรวจสอบเอกสารสิทธิ์ของไคลเอ็นต์โดยใช้ CertAuthenticator class ของมันทำการดึงเอาข้อมูลออกมาจากเอกสารสิทธิ์ของไคลเอ็นต์เพื่อตรวจสอบว่าเป็นของผู้ใช้คนไหน แล้วทำการยืนยันผู้ใช้จาก security realm การทำ mutual authentication นี้มีประโยชน์ในกรณีที่ เราต้องการจะจำกัดผู้ใช้ให้อยู่ในวงผู้ใช้ที่เราเชื่อถือได้เท่านั้น

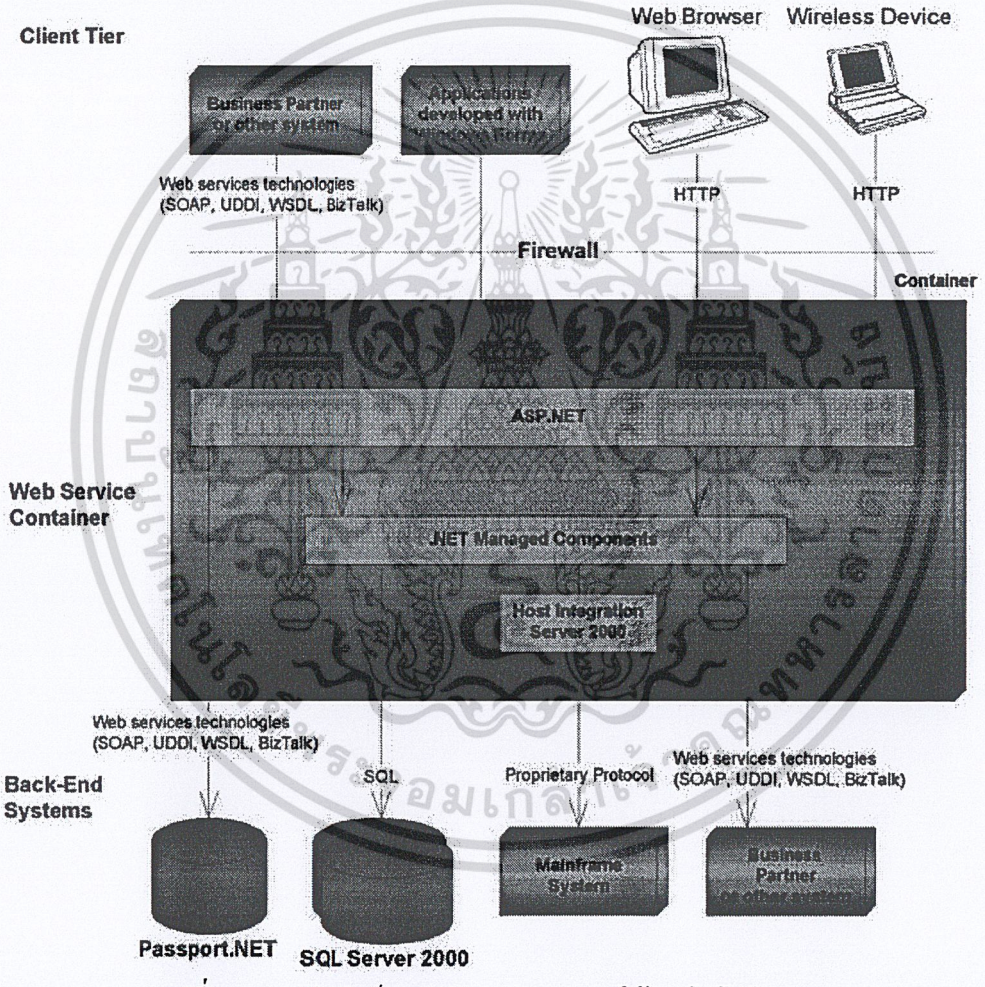
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
เปรียบเทียบ .NET เว็บเซอร์วิส และจาวาเว็บเซอร์วิส

ในบทนี้เราจะกล่าวถึงเทคโนโลยีในการพัฒนาโซลูชั่นจากทั้งสองค่าย ตั้งแต่แพล็ตฟอร์มในการพัฒนา และในส่วนของการสร้างเว็บเซอร์วิส

5.1 แพล็ตฟอร์มการพัฒนา

5.1.1 แพล็ตฟอร์มของ .NET



รูปที่ 5 - 1 แพล็ตฟอร์มการพัฒนาระบบภายใต้เทคโนโลยี .NET

<u>.NET แอพพลิเคชั่น</u>

ถูกดีพลอยอยู่ภายในคอนเทนเนอร์ ที่มีเซอร์วิสที่จำเป็นให้บริการสำหรับเอนเตอร์ไพรส์ แอพพลิเคชั่น เช่น ทรานแซ็กชัน ความปลอดภัย และ เมสเสจจิ้งเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ชั้นของธุรกิจ</u>

ของ.NET แอพพลิเคชั่นถูกสร้างโดยใช้ .NET เมเนจคอมโพเน็นต์(managed component) ชั้นนี้ทำในส่วนของการประมวลผลทางธุรกิจและการจัดการข้อมูล การติดต่อไปยังฐานข้อมูลโดยผ่าน ADO.NET มีการติดต่อกับระบบเก่า(legacy system) โดยผ่าน Microsoft Host Integration Server 2000 เช่นติดต่อกับ COM Transaction Integrator (COM TI) หรือติดต่อคู่ค้าทางธุรกิจผ่านเว็บเซอร์วิส

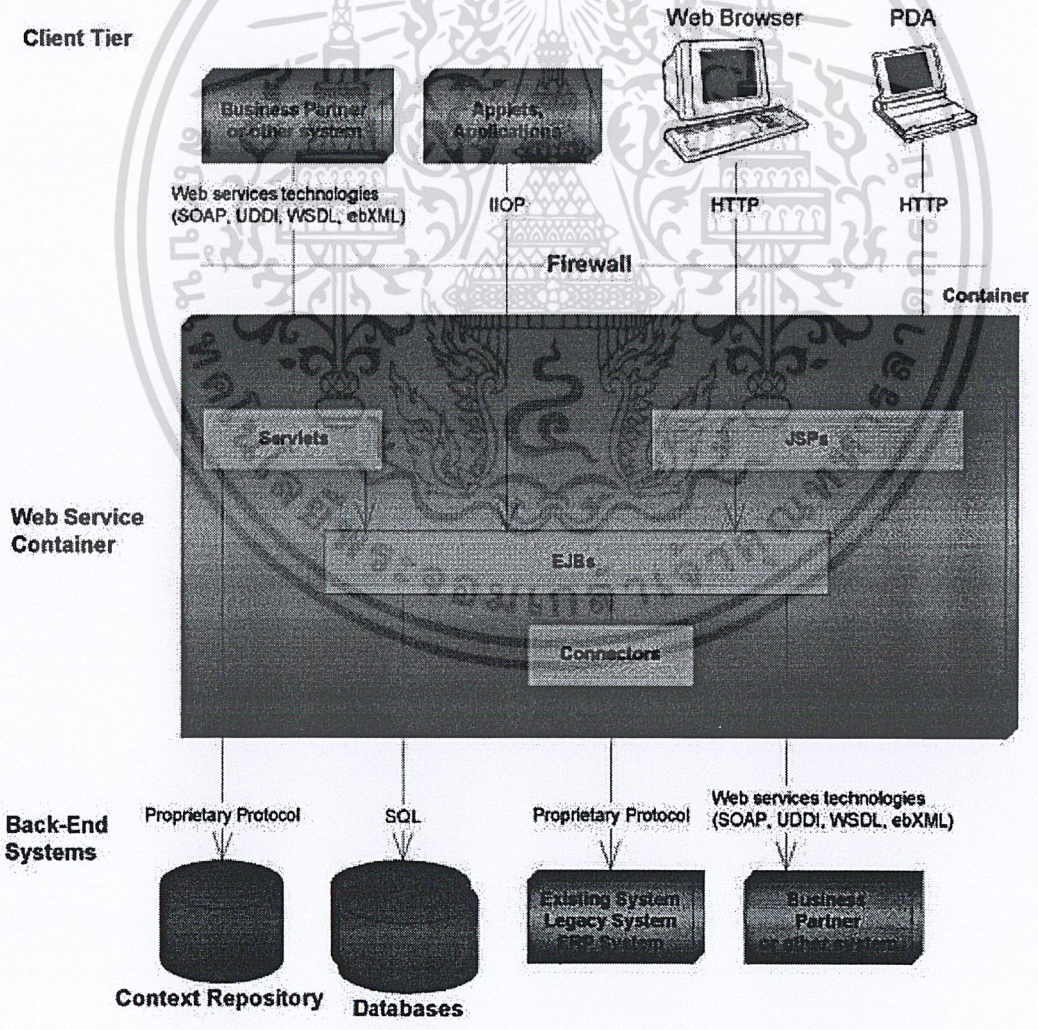
<u>คู่ค้าทางธุรกิจ</u>

สามารถใช้ .NET แอพพลิเคชั่นติดต่อโดยผ่านเว็บเซอร์วิส

<u>ไคลเอนต์ที่เรียกใช้งาน</u>

ก็เป็นได้ทั้งผ่านเว็บ ASP.NET ซึ่งให้ผลได้ในรูปของ HTML, XHTML, WML และรูปแบบของวินโดว์ฟอร์ม

5.1.2 แพล็ตฟอร์มของจาวา



รูปที่ 5 - 2 แพล็ตฟอร์มการพัฒนาระบบภายใต้เทคโนโลยี J2EE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

J2EE แอพพลิเคชั่น

ถูกดีพลอยอยู่ภายในคอนเทนเนอร์ ที่มีเซอร์วิสที่จำเป็นให้บริการสำหรับเอนเตอร์ไพรส์ แอพพลิเคชั่น เช่น ทรานแซ็กชัน ความปลอดภัย

ชั้นของธุรกิจ

ในแอพพลิเคชั่นขนาดใหญ่ถูกสร้างโดยใช้ Enterprise JavaBeans (EJB) คอมโพเน็นต์ ติดต่อฐานข้อมูลโดย JDBC หรือ SQL/J หรือติดต่อกับระบบเดิมผ่าน Java Connector Architecture (JCA) ติดต่อผ่านเว็บเซอร์วิสโดยใช้ จาวา API สำหรับXML (JAX APIs)

คู่ค้าทางธุรกิจ

ติดต่อผ่านเว็บเซอร์วิส ตัวเซิร์ฟเลท(servlet) เป็นตัวคอยรับการร้องขอที่เข้ามาละส่งผลลัพธ์กลับไปโดยใช้ JAX APIs เป็นตัวจัดการเกี่ยวกับเว็บเซอร์วิส

ไคลเอนต์ที่เรียกใช้งาน

นอกเหนือจากการผ่านเว็บ และเว็บเซอร์วิสแล้ว ยังมีแอพเพลตที่สามารถจะติดต่อกับจาวาบีนผ่านยังโปรโตคอล Internet Inter-ORB Protocol (IIOP)ได้ด้วย

5.2 เปรียบเทียบเทคโนโลยีของทั้งสองค่าย

เริ่มจากแสดงให้เห็นถึงความคล้ายคลึงกันของทั้งสองแพล็ตฟอร์มดังตารางข้างล่าง และจะทำการพิจารณาเปรียบเทียบให้เห็นในแต่ละหัวข้อ

Feature	J2EE	.NET
Type of technology	Standard	Product
Middleware Vendors	30+	Microsoft
Interpreter	JRE	CLR
Dynamic Web Pages	JSP	ASP.NET
Middle-Tier Components	EJB	.NET Managed Components
Database access	JDBC SQL/J	ADO.NET
SOAP, WSDL, UDDI	Yes	Yes
Implicit middleware (load-balancing, etc)	Yes	Yes

ตารางที่ 5 – 1 ตารางแสดงการสนับสนุนการทำงานด้านต่างๆของ 2 แพล็ตฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.1 Vendor Solution

ทางด้านของ J2EE นั้นเป็นเสปกที่ออกมาโดยมีเวนเดอร์ (vendor) หลายเจ้าที่นำเอามาตรฐานเหล่านั้นไปพัฒนาผลิตภัณฑ์ (product) ออกมารองรับ รวมทั้งมีเครื่องมือให้ใช้ค่อนข้างหลากหลาย ข้อดีคือมีตัวเลือกในการพัฒนาโซลูชั่นค่อนข้างหลากหลาย แต่ข้อเสียก็มาจากข้อดีของมันนั่นเองคือ ความไม่เข้ากันของแต่ละผลิตภัณฑ์ ไม่สามารถที่จะย้ายโค้ดโปรแกรมที่พัฒนาข้ามผลิตภัณฑ์ได้อย่างครบถ้วนซึ่งการพัฒนาระบบจริงนั้นเหมาะกับการเลือกใช้ผลิตภัณฑ์จากค่ายเดียวกันมากกว่า

ทางฝั่ง .NET ออกผลิตภัณฑ์ออกมานำ เนื่องจากเป็นผู้ออกเสปกและพัฒนาเองทั้งหมด รวมทั้งระบบที่ทำการสนับสนุนการทำงานส่วนมากก็ปเป็นของไมโครซอฟท์เอง

5.2.2 การสนับสนุนระบบเดิม

ในระบบเดิมที่มีการใช้งานโดยหลากหลายภาษาหลายระบบในส่วนของจาวานั้นก็มีหลากหลายทางในการทำให้สามารถใช้งานร่วมกันกับระบบเดิมเหล่านั้นได้

- ใช้ Java Message Service (JMS) เชื่อมกับระบบเก่า
- ใช้เว็บเซอร์วิสในการอินทิเกรตกับระบบต่างๆ
- ติดต่อผ่าน CORBA

รวมทั้งยังมี J2EE Connector Architecture (JCA) ทำหน้าที่เป็น ตัวติดต่อกับทรัพยากรอื่นๆ เช่น SAP R/3 ,CICS/COBOL และอื่นๆ

ฝั่ง .NET มีตัว Host Integration Server 2000 เป็นตัวช่วยจัดการเกี่ยวกับทรานแซ็กชันไปยังระบบเมนเฟรม หรือตัว Microsoft Message Queue (MSMQ) ในการทำงานร่วมกันกับระบบเดิมที่ใช้ IBM MQSeries และสุดท้ายคือตัว BizTalk Server2000 ทำการอินทิเกรตระบบบนโปรโตคอลของ B2B

5.2.3 การยอมรับทางการตลาด

ทางด้านจาวามีมากกว่า 50 บริษัทที่ให้การยอมรับในแพล็ตฟอร์ม

อีกด้านนั้นทีมการตลาดของไมโครซอฟท์ทำงานด้านนี้ได้ค่อนข้างดี รวมทั้งวินโดว์ก็มีผู้ใช้อยู่มากมาย

รวมทั้งทางด้านไมโครซอฟท์ประกาศตัวเกี่ยวกับเรื่องเว็บเซอร์วิสมาก่อนจาวา ในข้อนี้สรุปได้ว่าไมโครซอฟท์มีการทำการตลาดที่ดีกว่า

5.2.4 ภาษาที่รองรับ

จาวามีจุดเด่นที่เป็นภาษากลางที่สามารถนำไปใช้งานได้หลากหลายแพล็ตฟอร์ม

.NET มีจุดเด่นตรงที่รองรับได้หลายภาษาที่เป็นไปตามข้อกำหนดของ .NET เฟรมเวิร์ก โดยข้อดีของมันคือ ภายใต้โปรเจคเดียวกันสามารถพัฒนาได้จากหลายภาษา แต่ข้อเสียก็เยอะเช่นกัน เช่น ความเสี่ยงของระบบใหม่นี้ที่ทางไมโครซอฟท์พัฒนาขึ้นมา ของใหม่ย่อมมีความเสี่ยงเกิดขึ้น

ใช้ประโยชน์ด้านการค้า ครั้งที่มีการนำไปใช้

- การดูแลรักษาระบบ เนื่องจากเขียนด้วยหลากหลายภาษา ถ้าบุคคลผู้เชี่ยวชาญในบางภาษาลาออกไป ทำให้คนอื่นๆที่ดูแลลำบากในการทำความเข้าใจโค้ดภาษานั้นๆ

- การสร้างองค์ความรู้ การใช้งานหลากหลายภาษาส่งผลให้โปรแกรมเมอร์ไม่สามารถแชร์ประสบการณ์หลายๆอย่างในการพัฒนาแก่กันได้

ในการพัฒนาระบบนั้นก็ต้องพิจารณาให้เหมาะสม รวมถึงคำนึงถึงผลกระทบของการตัดสินใจด้วย

5.2.5 การสนับสนุนแพล็ตฟอร์มเดิม

J2EE ไม่มีปัญหาเรื่องการเชื่อมต่อกับระบบเดิมมากนัก แต่ตัว JCA และ เว็บเซอร์วิสก็ยังต้องการการเขียนโค้ดใหม่เพิ่มเติม ถึงแม้ว่าไมโครซอฟท์ .NET ได้รับการพัฒนามาจาก MTS และ COM+ แต่อย่างไรก็ตามรูปแบบการพัฒนาก็เปลี่ยนไปโดยสิ้นเชิง มีคอมมอนรันไทม์(CLR)เข้ามาจัดการ มีการจัดการเกี่ยวกับคอมโพเน็นต์ที่เปลี่ยนไป ทำให้อาจต้องมีการเขียนโค้ดใหม่เกือบ 60% เพื่อให้สามารถใช้ประโยชน์จากรันไทม์เอนเวอรอนเมนต์ใหม่นี้ได้อย่างเต็มที่

5.2.6 การทำงานข้ามแพล็ตฟอร์ม (Portability)

J2EE เป็นมาตรฐานสำหรับหลายๆแพล็ตฟอร์ม และรองรับการอิมพลีเมนต์ได้หลากหลาย แต่ถึงอย่างไรก็ตาม ยังมีปัญหาเรื่องของการนำโค้ดไปใช้ใหม่กับต่างผลิตภัณฑ์ เนื่องจากแต่ละผลิตภัณฑ์ก็ยังมีการเพิ่มความสามารถในผลิตภัณฑ์ของตัวเองด้วยที่ไม่เป็นไปตามมาตรฐาน J2EE ทางซันเองก็มีซอฟท์แวร์ที่ให้การทดสอบว่าผลิตภัณฑ์เป็นไปตามมาตรฐาน J2EE ไหม

ถึงวันนี้ ตัวผลิตภัณฑ์ที่รองรับการใช้งาน .NET ก็มาจากไมโครซอฟท์แห่งเดียวจึงไม่มีปัญหาส่วนนี้ แต่มีการพัฒนาเฟรมเวิร์กบนลินุกซ์ด้วยแต่ยังไม่สมบูรณ์นักการพัฒนาคลาสไลบารี่ มีการทดสอบการรันโค้ดโปรแกรมง่ายข้ามจาก วินโดว์ไปยังลินุกซ์ซึ่งก็ได้ผลถูกต้อง

ถึงแม้ว่าไมโครซอฟท์จะรับปากถึงการทำให้ .NET สามารถใช้งานข้ามระบบได้ก็ตาม แต่จากที่ผ่านมาเช่น COM คอมโพเน็นต์ ไมโครซอฟท์ก็สามารถทำให้ใช้ได้บนระบบปฏิบัติการอื่นเช่นกัน แต่ไม่มีการพัฒนาเซอร์วิสอื่นๆขึ้นมารองรับการใช้งานเลย ดังนั้นในส่วนของ .NET เราก็ต้องติดตามกันต่อไปว่าจะเป็นอย่างไร

5.2.7 เครื่องมือที่ใช้ในการพัฒนา

จาวามีเครื่องมือให้เลือกใช้ค่อนข้างเยอะ เช่น Forte, Visual Cafe, Visaul Age, JBulider

ไมโครซอฟท์ ใช้เครื่องมือ Visual Studio.NET ในการพัฒนาซึ่งมีเอนไวรอนเมนต์การใช้งานคล้ายกับ Visual Studio ตัวเดิมซึ่งโปรแกรมเมอร์คุ้นเคยอยู่แล้ว รวมทั้งเครื่องมือของไมโครซอฟท์เองก็มีประสิทธิภาพในการพัฒนาค่อนข้างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.8 การสนับสนุนการใช้งานเว็บเซอร์วิส

J2EE สนับสนุนการใช่งานเว็บเซอร์วิสโดยใช้ Java APIs สำหรับการแปลภาษา XML แต่อย่างไรก็ตามในส่วนนี้ยังอยู่ระหว่างการพัฒนา ยังต้องใช้ปรับแต่งเขียนโค้ดเพิ่มเติมด้วยตัวเองเพื่อให้เป็นเว็บเซอร์วิสได้ ผลิตภัณฑ์ของแต่ละแห่งก็มีการพัฒนาไลบรารี่ในส่วนนี้เพิ่มเติมขึ้นเองเพื่อให้การใช้งานเป็นไปอย่างอัตโนมัติและรวดเร็ว ซึ่งส่งผลต่อการผูกติดซอร์สโค้ดกับระบบ

เฟรมเวิร์กของ .NET ถูกพัฒนามาเพื่อสนับสนุนเว็บเซอร์วิสอย่างเต็มตัวตั้งแต่แรกเริ่มอยู่แล้ว

ไม่ว่าอย่างไรก็ตามมาตรฐานของเว็บเซอร์วิสยังต้องพัฒนาอีกยาวไกล เช่นเรื่องของการจัดการ ทรานแซ็กชัน เป็นต้นดังนั้นทั้งสองค่ายก็จะต้องเร่งออกผลิตภัณฑ์มารองรับในส่วนนี้ด้วย

5.2.9 การขยายการรองรับการใช้งานของระบบ(Scalability)

การเพิ่มประสิทธิภาพของระบบโดยทั่วไปเราสามารถที่จะเพิ่มได้การเลือกใช้ฮาร์ดแวร์ใหม่ๆที่มีประสิทธิภาพ ทาง J2EE สามารถใช้งานได้หลากหลายแพล็ตฟอร์มซึ่งแพล็ตฟอร์มนั้นก็สามารถเลือกใช้โพรเซสเซอร์ได้หลายตัวด้วย

ทาง .NET นั้นใช้งานบนวินโดส์เท่านั้นจึงมีข้อจำกัดทางด้านการสนับสนุนโพรเซสเซอร์

5.2.10 สรุปการเลือกใช้งาน

ทั้งสองแพล็ตฟอร์มสามารถนำไปสู่จุดมุ่งหมายที่เดียวกันได้ การที่จะเลือกใช้ผลิตภัณฑ์จากค่ายใดก็ต้องพิจารณาจากหลากหลายองค์ประกอบได้แก่ นักพัฒนาในองค์กรของคุณมีความถนัดระบบใด , ระบบเดิมที่ใช้งาน, ความสัมพันธ์กับบริษัทผู้ผลิต, และลูกค้าของบริษัทคุณ

5.3 เปรียบเทียบจาวาและ .NET

มีความพยายามเอาจาวาและ .NET มาเปรียบเทียบกัน โดยมีการเขียนโปรแกรมเลียนแบบแซมเปิลแอพพลิเคชั่น (Sample Application) ตัวหนึ่ง ชื่อ เพ็ทสโตร์ (Petstore) ของจาว่าใช้ชื่อว่า เพ็ทช็อป (Petshop) แล้วนำมาวัดประสิทธิภาพกัน ผลออกมาว่า .NET ให้ประสิทธิภาพในการทำงานรวดเร็วกว่าเพ็ทสโตร์ที่พัฒนาด้วยจาวาถึง 17 เท่า ซึ่งคุณสามารถไปดาวน์โหลดโค้ดได้ฟรี การเปรียบเทียบครั้งนี้ได้รับการประชาสัมพันธ์กันขนานใหญ่ แต่ก็มีโปรแกรมเมอร์ออกมาวิจารณ์กันมาก ถึงความเป็นไปได้ที่อาจจะมีการลำเอียงในการเปรียบเทียบ โดยประเด็นที่มีการโต้แย้งหลักๆ ดังนี้

1.เพ็ทสโตร์ (Petstore) ถูกออกแบบมาบนพื้นฐานของการแสดงคุณลักษณะของจาว่าเป็นหลัก ไม่ได้เน้นเรื่องของความเร็ว การที่เพ็ทช็อปจะมาโจมตีในส่วนของความเร็วและเหมารวมไปหมดว่า .NET เหนือกว่านั้น ไม่น่าจะถูกต้องนัก

2. เพ็ทช็อป (Pet Shop) ใช้สโตร์โพรซิเดอร์ในฐานข้อมูลในการทำงาน ซึ่งแน่นอน ย่อมให้ความเร็วในการทำงานเหนือกว่าเพ็ทสโตร์อยู่แล้ว และจะไม่สามารถนำไปทำงานบนฐานข้อมูลระบบอื่น ได้ทันที

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมือนเพ็ทสโตร์ เพราะสโตร์โพรซิเดอร์ไม่เหมือนกัน ถ้าจะเปรียบเทียบกันอย่างเป็นธรรมก็ต้องเขียนโปรแกรมติดต่อฐานข้อมูลโดยไม่ใช้สโตร์โพรซิเดอร์

3. จำนวนบรรทัดของโปรแกรมนั้นเพ็ทช็อปใช้จำนวนบรรทัดเพียง 1 ใน 3 ของเพ็ทสโตร์ ส่วนหนึ่งเป็นเพราะโครงสร้างภาษา C# ที่พัฒนามาให้กระชับกว่าอยู่แล้ว แต่อีกส่วนหนึ่งก็เกิดจากการเขียนโปรแกรมโดยมีจำนวนบรรทัดของ Error Exception น้อยกว่า

4. การทดสอบนั้น ทำบนแพล็ตฟอร์มของวินโดวส์ ซึ่งแน่นอนว่า .NET ย่อมจะทำงานได้เป็นอย่างดีกับผลิตภัณฑ์จากบริษัทเดียวกัน แต่ถ้านำเอาไปทดสอบบนระบบอื่นๆ เช่นยูนิกซ์ (UNIX) อาจจะไม่ให้ผลแตกต่างกันมากถึงขนาดนี้

5. มีการใช้แคช (Cache) จำนวนมากกับเพจ (Page) ที่พัฒนาบนเพ็ทช็อป ซึ่งแน่นอน ย่อมทำให้เพ็ทช็อปทำงานได้เร็วขึ้นอยู่แล้ว ขณะที่เพ็ทสโตร์ไม่ได้เน้นในเรื่องนี้

6. ประเด็นอื่นๆ มีอีกมากมาย เช่น การวางโมเดล (Model) ที่ไม่เหมือนกัน ความเสถียร

5.4 การพัฒนาเว็บเซอร์วิสจากทั้งสองแพล็ตฟอร์ม

5.4.1 การเรียกใช้งานข้ามแพล็ตฟอร์ม

จากโปรโตคอล SOAP ที่ใช้ในการติดต่อสื่อสารเว็บเซอร์วิสมีการอธิบายการทำงานแค่เพียงสามารถให้มีการเรียกใช้งานข้ามระบบได้เท่านั้น โดยแลกเปลี่ยนบริการข้อมูลต่างๆในรูปของ XML จากการเรียกใช้งานจริงก็สามารถทำการเรียกใช้เมธอดได้อย่างถูกต้อง

5.4.2 ปัญหาที่พบ

ดังที่กล่าวไว้ในหัวข้อที่แล้วว่า สามรถเรียกเปลี่ยนเซอร์วิสข้อมูลได้เพียงเบื้องต้นเท่านั้น จากการพยายามที่จะให้ทั้งสองค่าย มีการใช้เซสชันระหว่างกัน ไม่สามารถทำได้ เพราะเป็นเพียงการพัฒนาในระบบของตัวเองเท่านั้น

การใช้งานในธุรกิจ ปัญหาเรื่องของทรานแซ็กชัน ที่ต้องมีการทำ two phrase commit การใช้งานเว็บเซอร์วิสไม่สามารถเรียกสนับสนุนการทำงานในส่วนนี้ได้ เนื่องจากโปรโตคอล, ภาษา XML ที่ใช้ ไม่ได้รองรับการจัดการทรานแซ็กชันเลย ยังต้องรอมาตรฐานที่คาดว่าจะออกมาในประมาณปี 2547

ความปลอดภัยในการรับส่งข้อมูลข้ามเครือข่ายก็เป็นปัญหาสำคัญในทางธุรกิจเช่นกัน โดยปัจจุบันการใช้งานที่มีให้ใช้คือ การใช้การพิสูจน์ตัว(Authentication) ผ่านเว็บเซิร์ฟเวอร์ของระบบตนเอง รวมด้วยการปรับแต่งระบบให้ใช้โปรโตคอล SSL แต่ก็ทำให้ประสิทธิภาพของระบบลดลงไปด้วย มาตรฐานของความปลอดภัยในเว็บเซอร์วิสเพิ่งจะออกมาได้ไม่นาน ออกโดยองค์กร W3C คือมาตรฐาน WS-Security ทางด้าน .NET นั้นออกปลั๊กอินของ ASP.NET เพื่อให้ใช้งานตามมาตรฐานได้ ส่วนฝั่ง J2EE นั้น BEA Web Logic ในเวอร์ชัน 2 ถึงจะมีการสนับสนุนซึ่งอยู่ระหว่างการพัฒนา

เอกสารนี้มีไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางด้านจาวายังมีปัญหาที่เกิดขึ้นอีกอย่างคือ การพอร์ตหรือนำโค้ดที่เขียนเพื่อดีพลอยบนเว็บเซิร์ฟเวอร์ตัวหนึ่งไปใช้บนเว็บเซิร์ฟเวอร์ของบริษัทอื่นนั้นไม่สามารถทำได้ เนื่องจากมีการผูกติดกับไลบรารี่ของเว็บเซิร์ฟเวอร์ของแต่ละบริษัทด้วย

ข้อมูลเปรียบเทียบ	.NET	Java
1.โครงสร้างภาษา การจัดการหน่วยความจำ JIT เครื่องเสมือน(Virtual Machine) มีหลักการคล้ายๆกัน	แนวคิดแบบ OOP ลอกแบบมาจากจาว่า C++ เพิ่มเติมแนวคิดบางประการจากจาว่า เช่น เมตาดาต้า (Meta Data) มี Intermediate Language(IL) เป็นภาษากลาง ทำงานบน Common Language Runtime (CLR)	ต้นแบบมาจาก C/C++ มี CORBA IDL และ ORB ช่วยทำให้โค้ดหลายๆภาษาใช้ออบเจกต์ (Object) ร่วมกันได้ ทำงานบน J2EE แต่ไม่ใกล้ชิดกันเหมือนกับ IL ทำงานกับ CLR บน .NET
2. ภาษาที่สนับสนุน	C#,VB.NET เป็นหลัก ภาษาอื่นๆที่สามารถสนับสนุน แต่อาจไม่สะดวกในการใช้งาน เช่น C++,COBOL,Python,Delphi,J#	จาวาเป็นหลัก สามารถใช้ภาษาอื่นได้บ้าง แต่ไม่ดี ต้องการเครื่องมืออื่นเพิ่มเติม เช่น C++,C
3. ราคา	Visual Studio .NET ซึ่งมีราคาแพง และฟรี สำหรับ SDK และ Web Matrix ซึ่งเป็นเครื่องมือพื้นฐานในการพัฒนาที่ไม่เก่งนัก	มีเครื่องมือฟรีมากมายแจกจ่าย เครื่องมือที่เก่งกว่าได้แก่ J-Builder ราคาถูกกว่า .NET
4. ความเร็วในการพัฒนา ในงานขนาดเท่ากัน	เร็วกว่าจาวา (ประมาณ 2-5 เท่ากับงานที่เท่ากัน โดยเฉพาะงานเว็บหรือฐานข้อมูล) เพราะเครื่องมือที่มีมาให้ฉลาด และอำนวยความสะดวกได้มาก	
5. ความเร็วในการทำงานของแอพพลิเคชัน	เร็วกว่า 17 เท่าตามการเปรียบเทียบแอพพลิเคชันที่คล้ายกัน Petstore/Petshop	น่าจะช้ากว่า แต่ไม่น่าจะถึงกับที่ถูกกล่าวอ้างในการเปรียบเทียบระหว่าง PetStore/PetShop
6. การทำงานข้ามแพล็ทฟอร์ม	- ทำงานได้เฉพาะระบบวินโดวส์ แต่ในอนาคต จะสนับสนุนระบบอื่น เช่น ลีนุกซ์ (Linux) - สามารถทำงานข้ามแพล็ตฟอร์ม	ทำงานได้หลากหลายแทบทุกแพล็ตฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ได้ ถ้าเขียนโปรแกรมในลักษณะของเว็บแอพพลิเคชัน (Web Application) ASP.NET แต่การแสดงผลและการทำงานบางอย่าง เช่น Validation Control ก็ไม่สมบูรณ์ 100% บนบราวเซอร์อื่น	
7. คลาสที่สนับสนุน	.NET Framework SDK- มีหลากหลายกว่า 1400 คลาส ที่พัฒนาโดยไมโครซอฟท์ ระบบค่อนข้างผูกขาดบนไมโครซอฟท์	จาวาคอร์เอพีไอ (Java core API) เป็นระบบเปิด มีพัฒนาโค้ดเปิด (Open source) และแจกฟรีมากมาย
8. ความเข้ากันได้กับ Application/Solution อื่นๆ	เข้ากันได้ดีกับผลิตภัณฑ์ของไมโครซอฟท์เอง เช่น ไมโครซอฟท์ออฟฟิศ (MS Office) Exchange, MS SQL	ได้รับการยอมรับจากระบบที่หลากหลายกว่า เช่น IBM , Notes , SAP
9. ขนาดที่เหมาะสมของโครงการและการลงทุน	เหมาะสำหรับโครงการระยะสั้น-กลาง ขนาด ระยะเวลาสั้น-กลาง เช่นเดียวกัน เหมาะกับโครงการขนาดเล็กที่ต้องแข่งขันกันด้านราคา หรือทำเป็นแอพพลิเคชันสำเร็จรูป	เหมาะสมกับโครงการขนาดกลาง-ใหญ่ ระยะเวลาโครงการยาวนานกว่า และมีรายละเอียดข้อจำกัดทางเทคนิค และระบบค่อนข้างมาก
10. การเปิดกว้างทางเทคโนโลยี	เทคโนโลยีค่อนข้างอยู่ในวงจำกัด ข้อมูลเชิงลึกไม่เปิดเผย โครงการในลักษณะโค้ดเปิดค่อนข้างน้อย โค้ดตัวอย่างฟรี หาดีๆได้ยากกว่า	ค่อนข้างเปิดเผยอย่างกว้างขวาง มีโครงการโค้ดเปิด จำนวนมาก แจกจ่ายให้โค้ดดีๆที่ค่อนข้างสมบูรณ์แบบให้ดาวน์โหลดฟรี
11. การสนับสนุนเกี่ยวกับเว็บ	เก่งมากสำหรับงานบนเว็บ มีเทคโนโลยี ASP.NET เขียนเว็บแอพพลิเคชันได้ง่าย และคล้ายคลึงกับการเขียนโปรแกรมบนวินโดวส์ทั่วไป มีเครื่องมือที่เก่งด้านเว็บเซอร์วิส (Webservices) ,SOAP มีคลาสสำเร็จที่สร้างเอกสาร XML ให้เลย โดยใช้เพียง ADO.NET	จาวาเซิร์ฟเวอร์เพจ (Java Server Pages) เครื่องมือในการพัฒนายังค่อนข้างพื้นๆ ต้องเข้าใจ HTML Tag ต่างๆเอง เครื่องมือยังไม่เก่งมากนักเกี่ยวกับงานเว็บเซอร์วิส ต้องเขียนโปรแกรมเพื่อจัดการกับเว็บเซอร์วิสเองเป็นส่วนใหญ่ โดยใช้ JDBC, EJB, JMS, Java XML Libraries

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ หาประโยชน์ในเชิงการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. สนับสนุนกับวินโดวส์ฟอร์ม	ใช้เครื่องมือพัฒนา Visual Studio.NET ในการพัฒนา	ใช้จาวาสวิงส์ (Java Swing) ซึ่งได้รับการสนับสนุนจากเครื่องมือพัฒนาจาวาที่หลากหลาย
13. Inter-Operatibility	เน้นการทำงานข้ามภาษา	เน้นการทำงานแพล็ตฟอร์ม

ตารางที่ 5 – 2 สรุปข้อเปรียบเทียบระหว่างสองแพล็ตฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

แอพพลิเคชัน

6.1 Overview OLALA Wedding Planner

ระบบ OLALA Wedding Planner เป็นระบบจัดการการแต่งงาน ที่เป็นเว็บเซอร์วิสให้บริการจัดหาโรงแรม , ของที่ระลึก , ถ่ายรูปแต่งงาน , การจองแพ็คเกจทัวร์ , การจองวงดนตรีหรือการแสดง ที่เป็นส่วนหนึ่งในงานแต่งงาน โดยผู้ใช้บริการสามารถค้นหาข้อมูลต่างๆได้ตามต้องการ โดยกรอกรายละเอียดที่จำเป็นในการค้นหาลงไป เช่น ต้องการจองห้องจัดเลี้ยง ข้อมูลที่ต้องใส่ในการค้นหาอย่างน้อย คือ สไตล์การตกแต่งสถานที่ห้องจัดเลี้ยงแบบไหน , การจองการถ่ายรูปแต่งงาน ข้อมูลที่ต้องใส่อย่างน้อย คือ ประเภทของการถ่ายรูป (ในสถานที่ , นอกสถานที่) และช่วงราคา เป็นต้น จากนั้นทางระบบจะเรียกใช้เซอร์วิสต่างๆ ที่ เว็บเซอร์วิสมีให้เรียกใช้

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

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

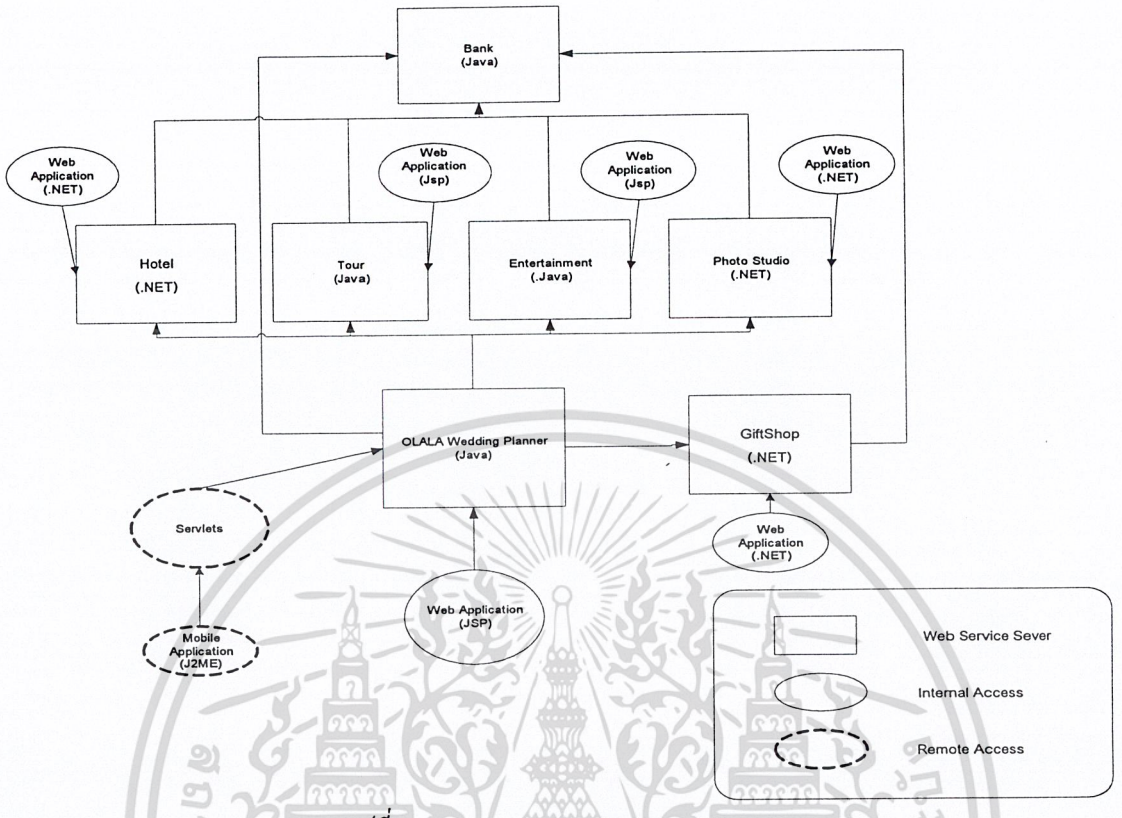
เทคโนโลยีของจาวาจะประกอบด้วย

- ธนาคาร
- ระบบทัวร์
- ระบบเอ็นเตอร์เทนเมนต์
- ระบบ OLALA Wedding Planner

เทคโนโลยีของ .NET เฟรมเวิร์คประกอบด้วย

- ระบบโรงแรม
- ระบบสตูดิโอถ่ายภาพ
- ระบบขายของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-1 ระบบ OLALA Wedding Planner

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 รายละเอียดของแอพพลิเคชัน

ระบบ Olala Wedding planner ประกอบด้วยบริการส่วนต่างๆดังนี้

1. การลงทะเบียน
2. การค้นหาข้อมูลต่างๆของ
   - ทัวร์
   - โรงแรม
   - ร้านถ่ายภาพ
   - ร้านของที่ระลึก
   - การแสดง
3. การจอง
4. การแก้ไขข้อมูลที่จอง
5. การยกเลิกการจอง
6. การชำระค่าบริการ

การลงทะเบียน

สามารถลงทะเบียนผ่านทางหน้าเว็บ โดยจะต้องกรอกรายละเอียดต่างๆรวมทั้ง username และรหัสผ่าน

การค้นหาข้อมูล

โดยแต่ระระบบจะจำลองโดยมีอย่างละ 10 บริษัท สามารถค้นหาแบบถูกสุดได้โดยจะทำการเปรียบเทียบราคากับบริษัทอื่นๆ นอกจากนี้ยังสามารถค้นหาแบบใกล้สุด(ได้เฉพาะในกรุงเทพ) โดยการเลือกเขตที่ต้องการระบบจะหาบริษัทที่อยู่ใกล้กับเขตนั้นที่สุดให้ หรืออาจจะเลือกเป็นภาคโดยผลการค้นหาจะได้เฉพาะบริษัทที่อยู่ในภาคที่ต้องการ โดยเงื่อนไขทั้งหมดสามารถนำมาผสมกันได้ในการค้นหาเพื่อให้ได้ผลลัพธ์ที่ตรงตามความต้องการมากที่สุด นอกเหนือจากนี้จะเป็นค่าพารามิเตอร์ที่ใช้ในการค้นหาของเซอร์วิสของบริษัทต่างๆซึ่งจะแตกต่างกันไป

การจอง

สามารถทำการจองโดยเลือกจากข้อมูลที่ค้นหาได้ซึ่งเส้นตายในการจ่ายเงินนั้นจะขึ้นอยู่กับบริษัทต่างๆที่กำหนดขึ้น

การยกเลิกการจอง

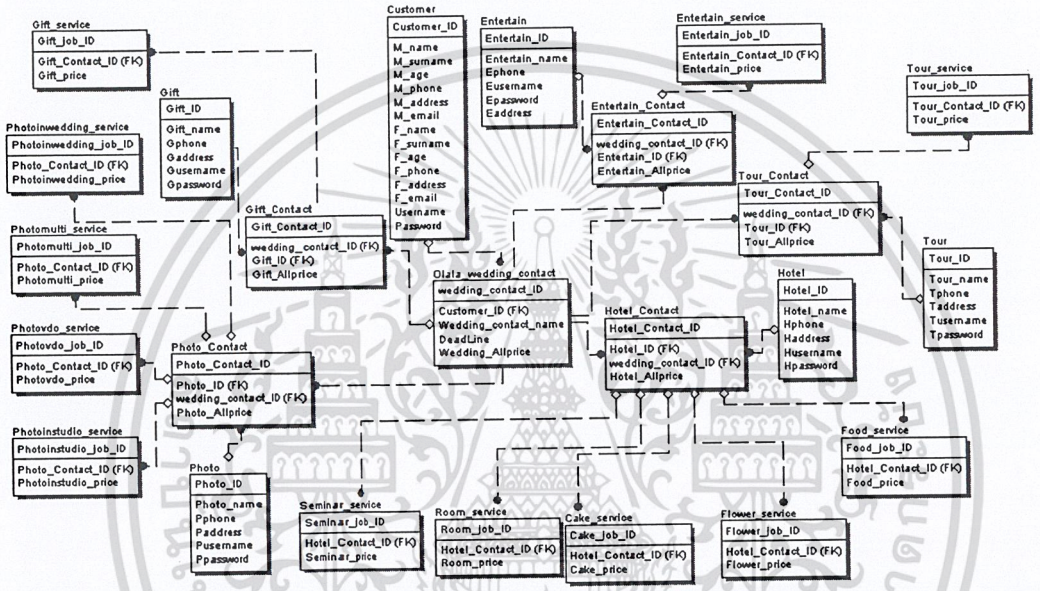
ลูกค้าสามารถทำการยกเลิกการจองได้ตลอดเวลาแต่ในกรณีที่ชำระเงินแล้วจะไม่ได้รับเงินคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การชำระค่าบริการ

ชำระโดยใช้บัตรเครดิต โดยชำระกับทาง Olala Wedding Planner โดยไม่ต้องจ่ายกับบริษัทอื่นๆโดยตรง โดยเส้นตายในการชำระเงินของบริษัทต่างๆจะไม่เท่ากัน โดยทาง Olala Wedding Planner จะยึดเส้นตายของบริษัทที่เร็วที่สุดเป็นหลัก โดยลูกค้าจะต้องชำระเงินในเวลาดังกล่าว

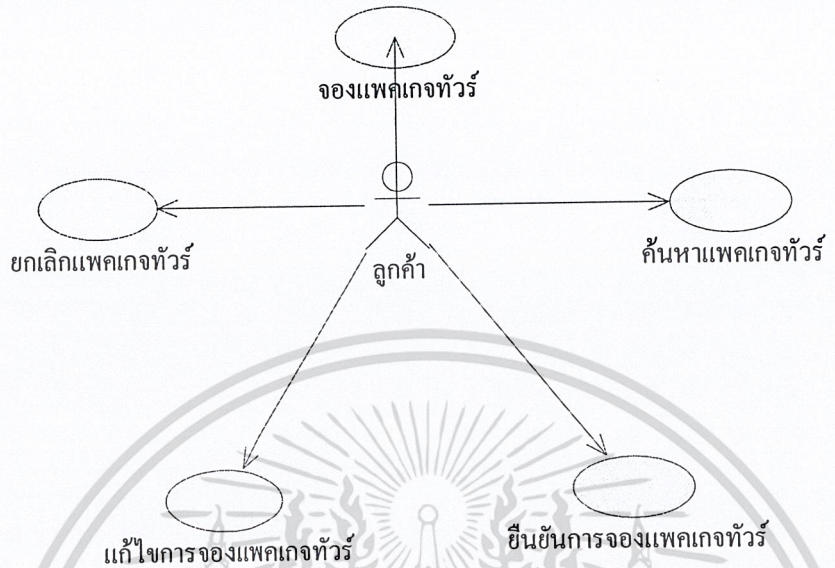
6.3 อีอาร์ไดอะแกรม (ER Diagram)



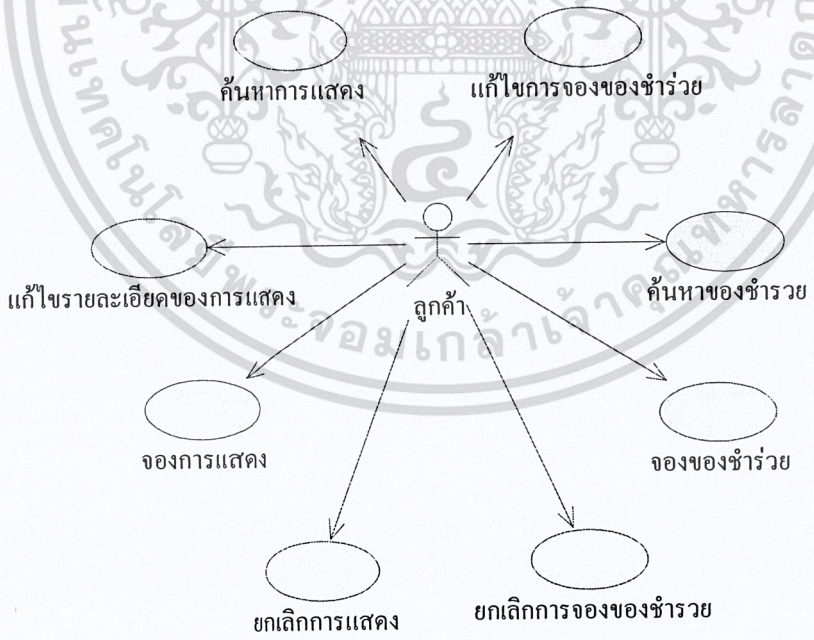
รูปที่ 6-2 แสดงอีอาร์ไดอะแกรมของระบบ Olala Wedding Planner

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 Use Case ไดอะแกรม

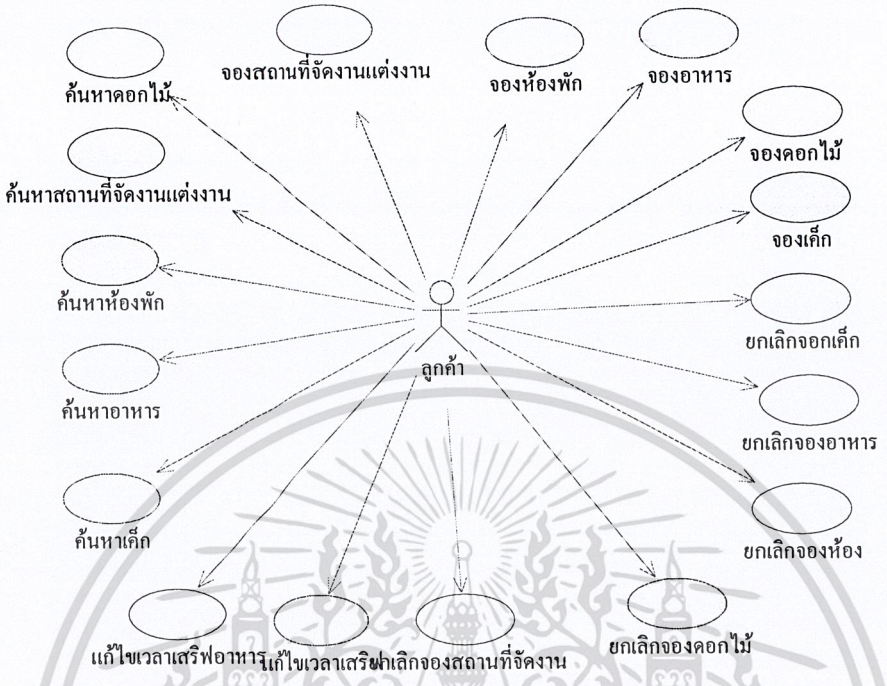


รูปที่ 6-3 แสดง use case ของแอพพลิเคชันในส่วนของบริษัททัวร์

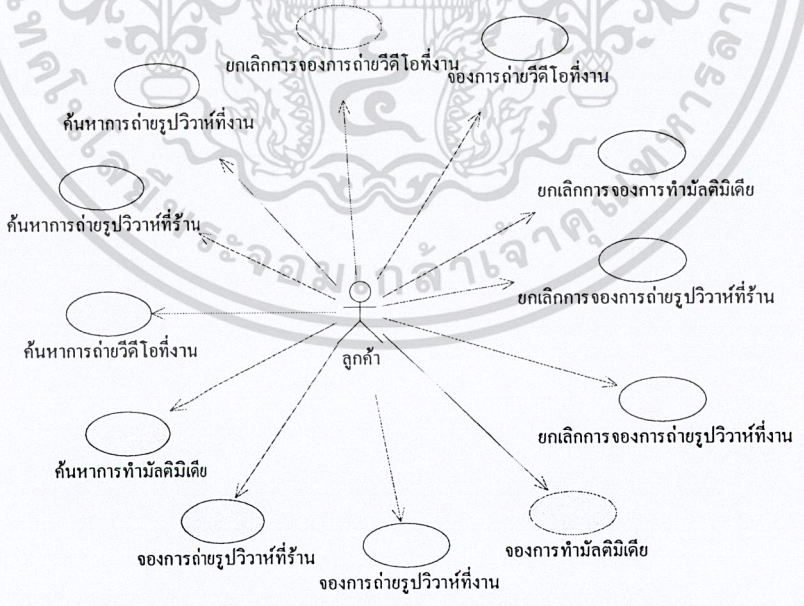


รูปที่ 6-4 แสดง use case ของแอพพลิเคชันในส่วนของบริษัทของชำร่วย กับ เอ็นเตอร์เทนเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

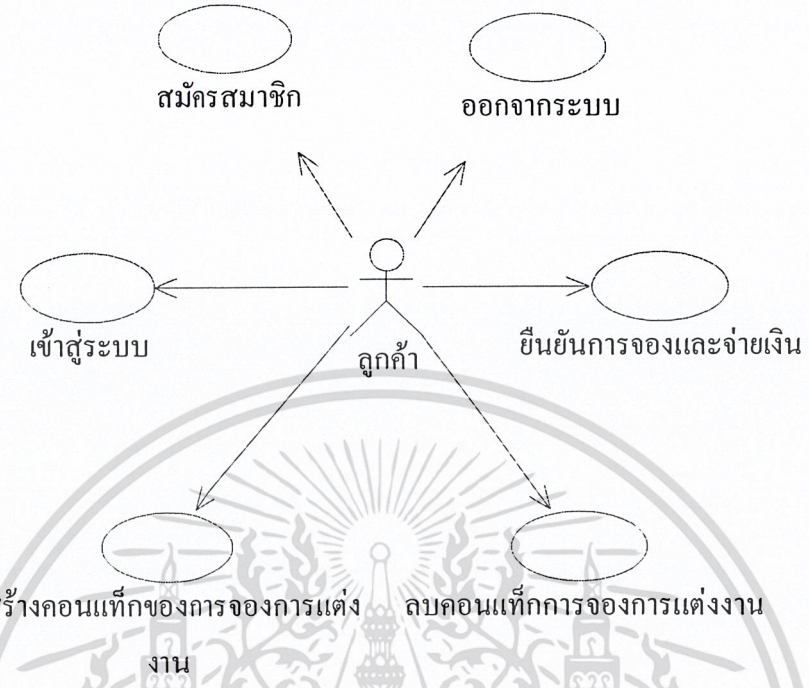


รูปที่ 6-5 แสดง use case ของแอพพลิเคชันในส่วนของโรงแรม



รูปที่ 6-6 แสดง use case ของแอพพลิเคชันในส่วนของโฟโตสตูดิโอ

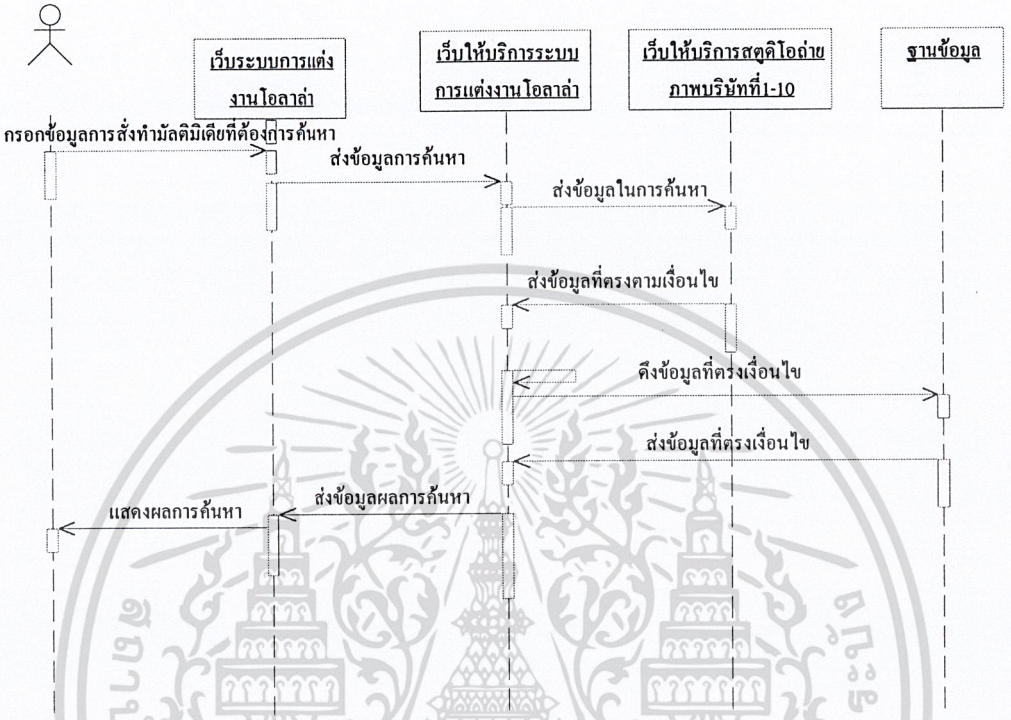
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-7 แสดง use case ของแอพพลิเคชันในส่วนของ Olala Wedding Planner

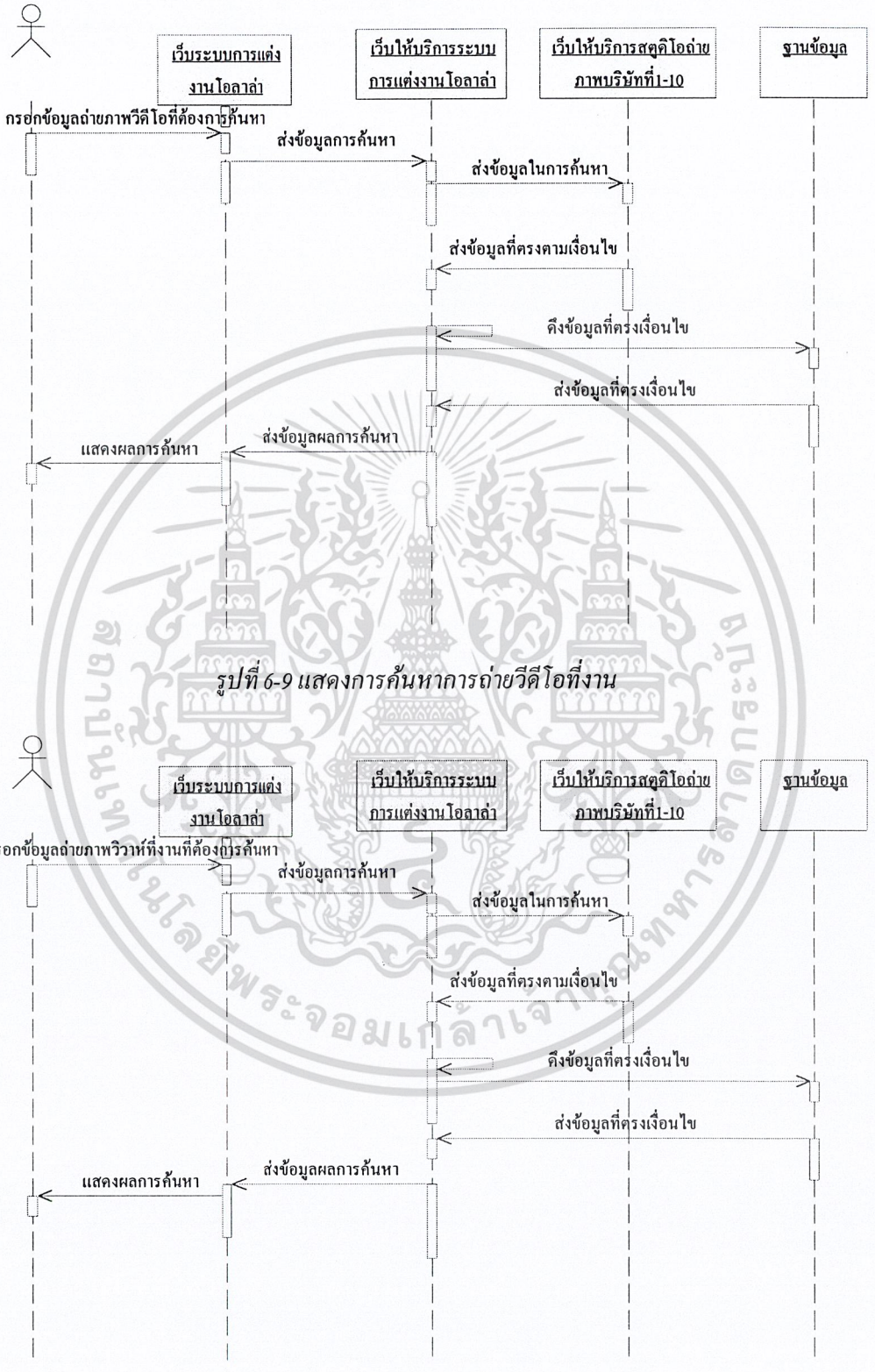
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 ซีเควนซ์ ไดอะแกรม (Sequence Diagram)



รูปที่ 6-8 แสดงการค้นหามัลติมีเดีย

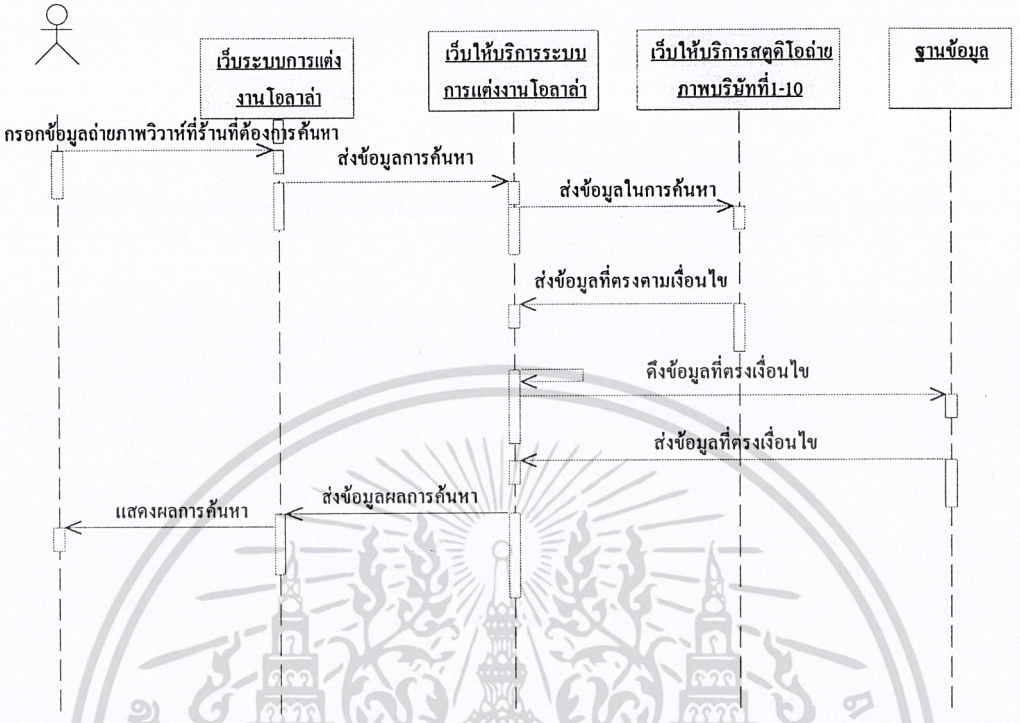
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-9 แสดงการค้นหาการถ่ายวีดีโอที่งาน

รูปที่ 6-10 แสดงการค้นหาการถ่ายภาพที่งานวิวาห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



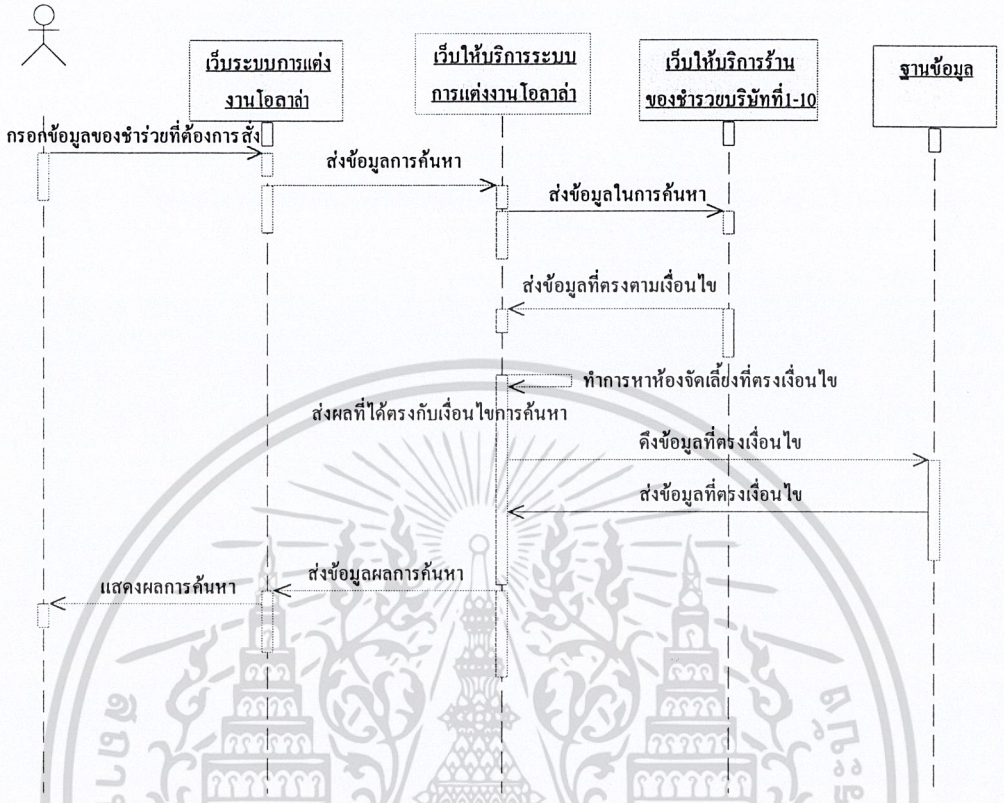
รูปที่ 6-11 แสดงการค้นหาการถ่ายภาพที่ร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



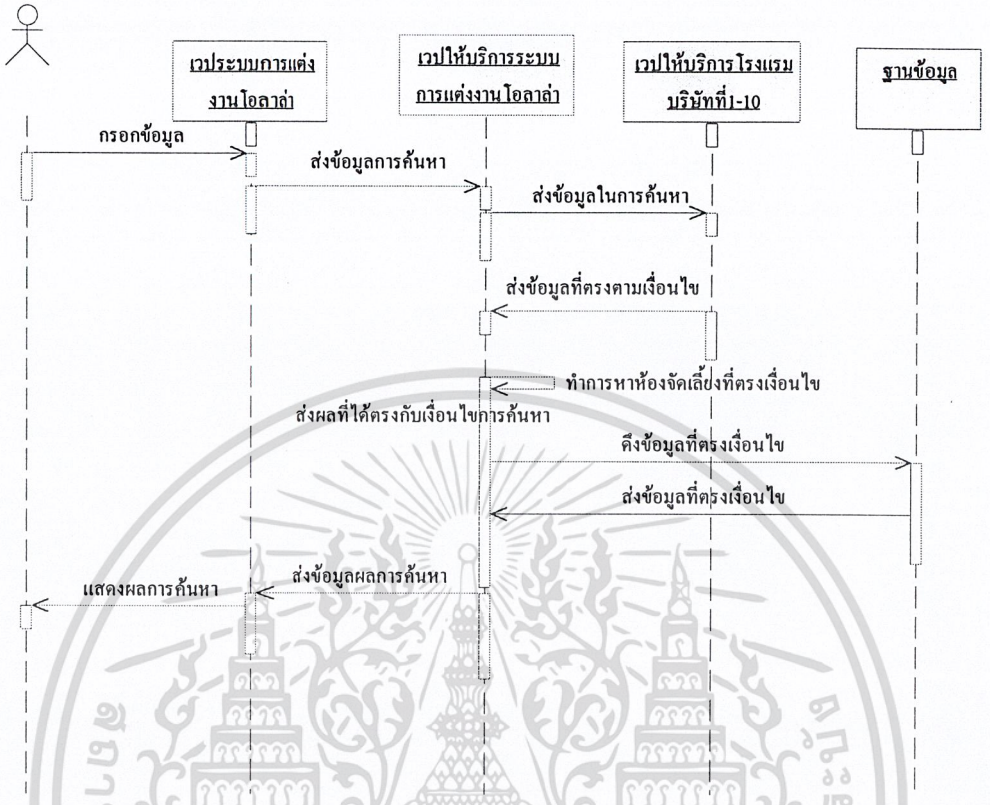
รูปที่ 6-12 แสดงการค้นหาการแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



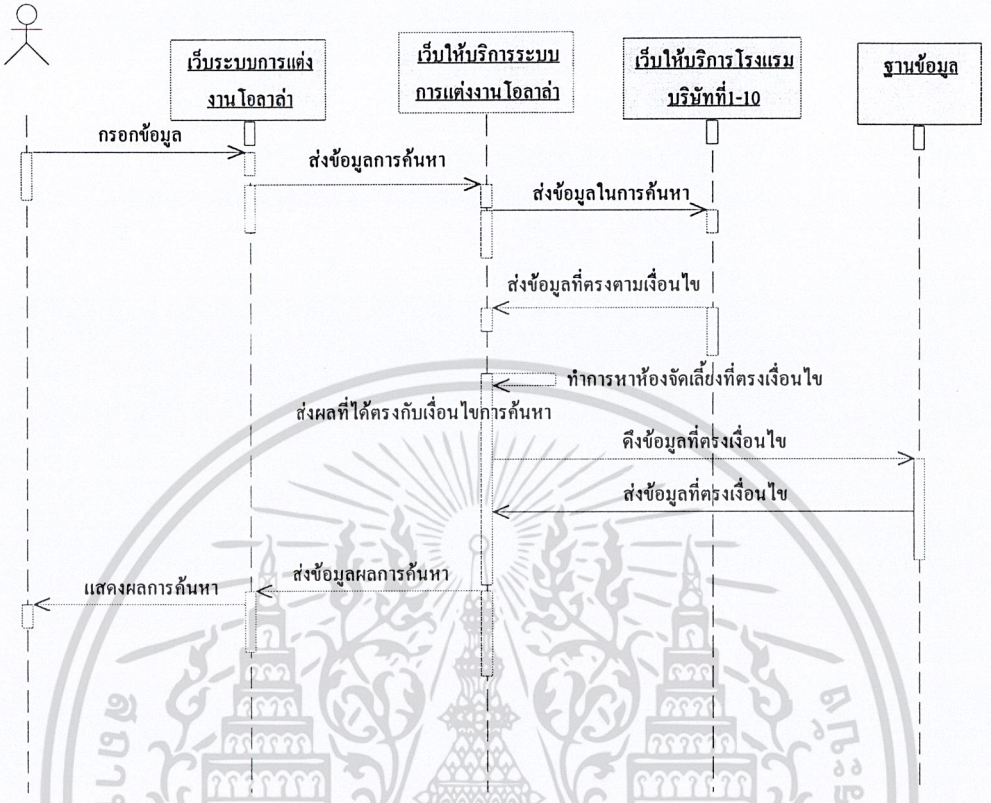
รูปที่ 6-13 แสดงการค้นหาของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



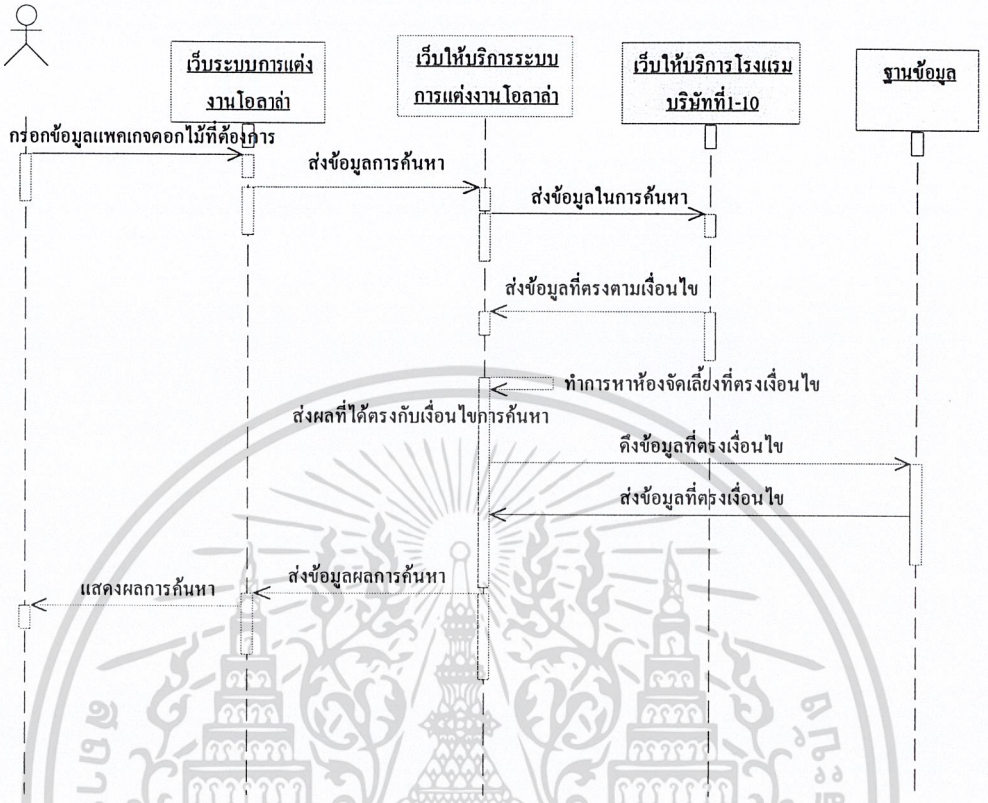
รูปที่ 6-14 แสดงการค้นหาเค้ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



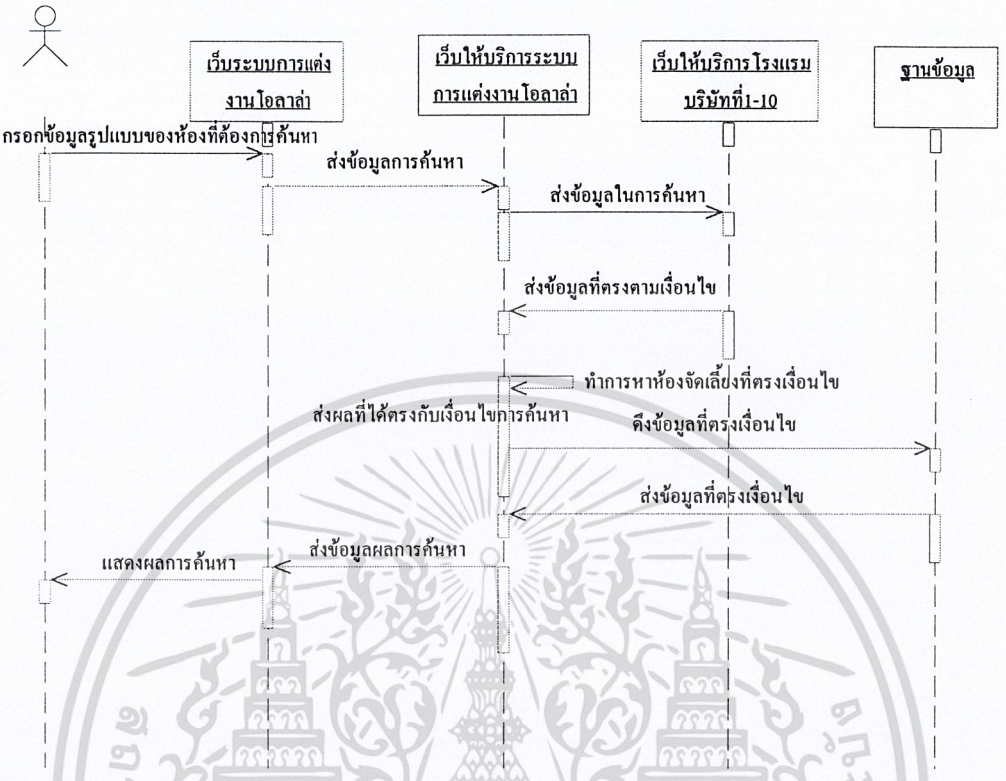
รูปที่ 6-15 แสดงการค้นหาห้องจัดเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



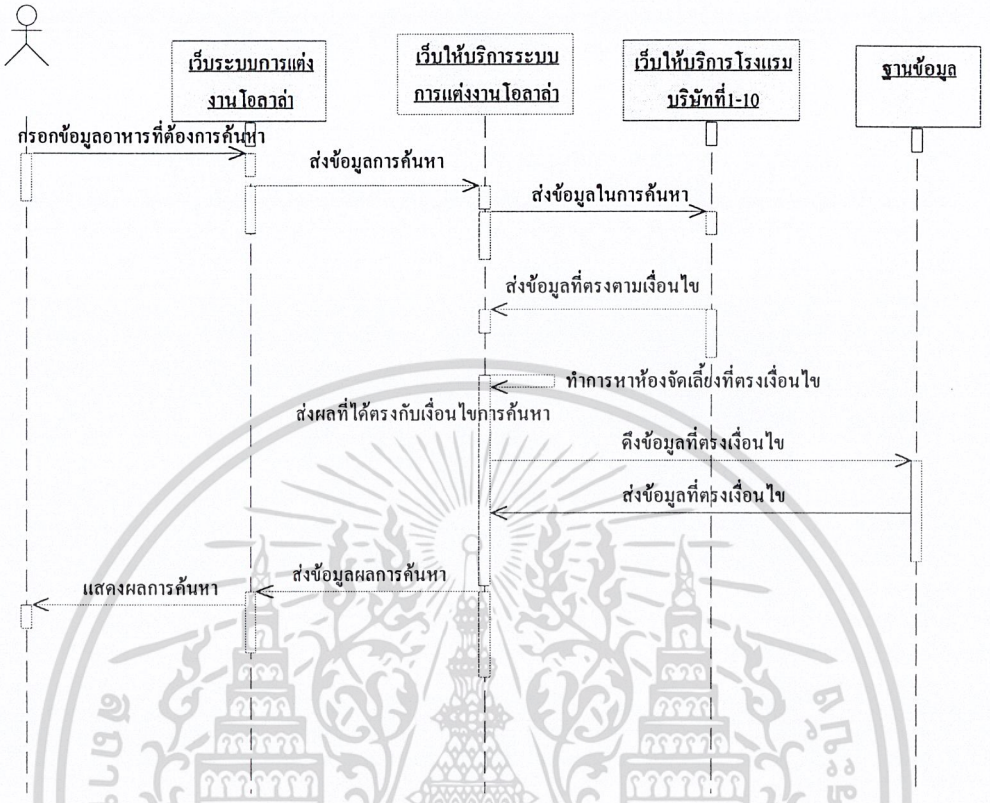
รูปที่ 6-16 แสดงการค้นหาแพคเกจดอกไม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



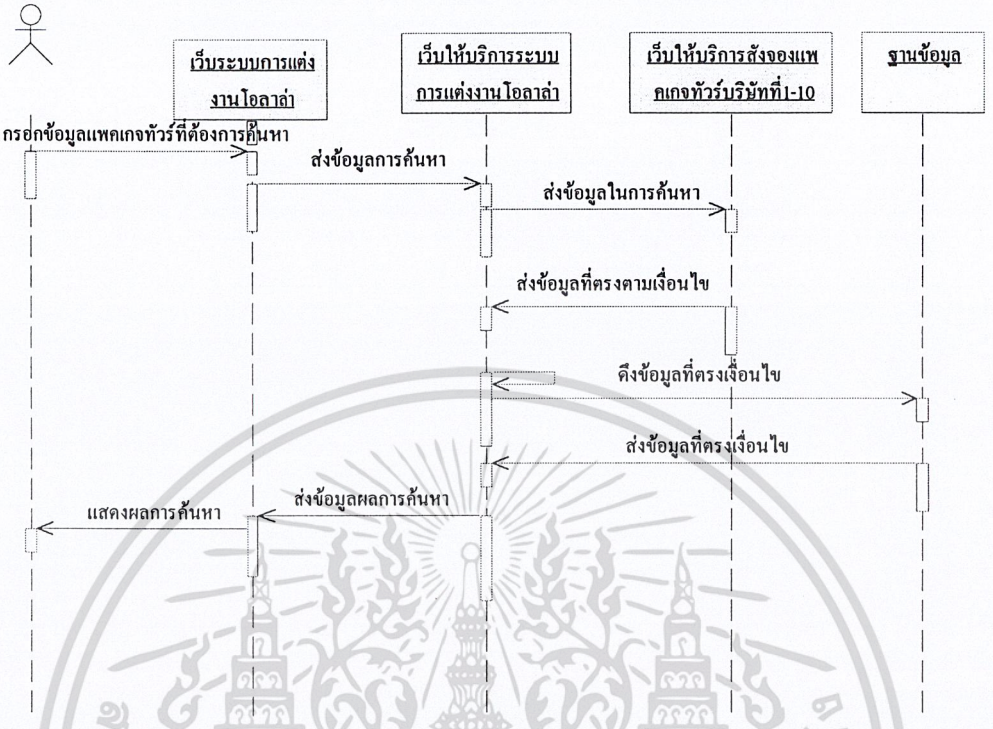
รูปที่ 6-17 แสดงการค้นหาห้องพัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

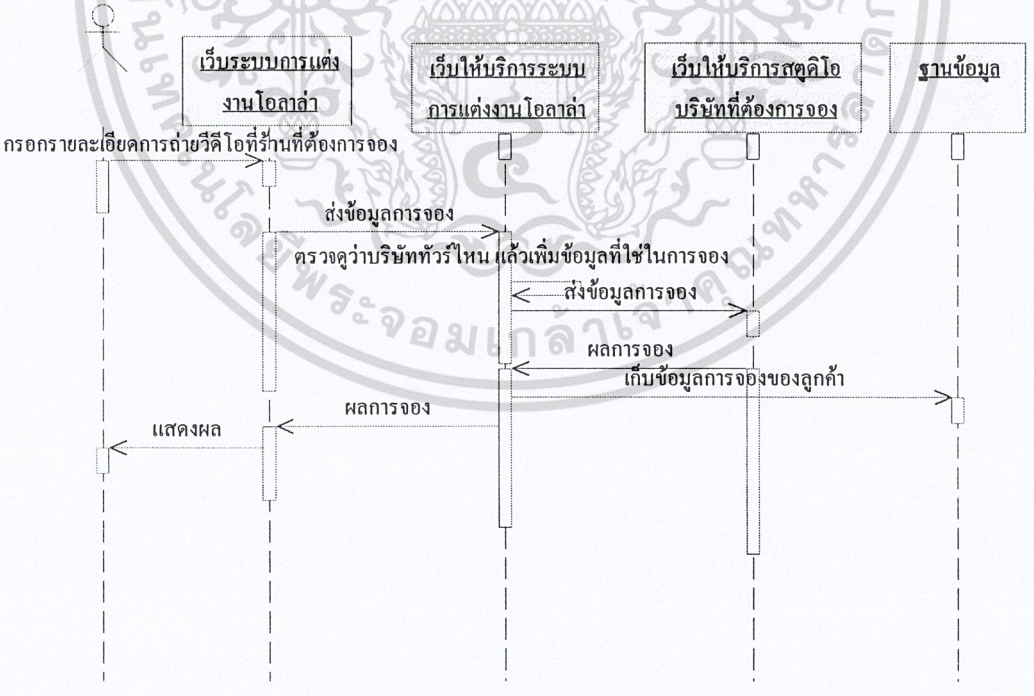


รูปที่ 6-18 แสดงการค้นหาอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

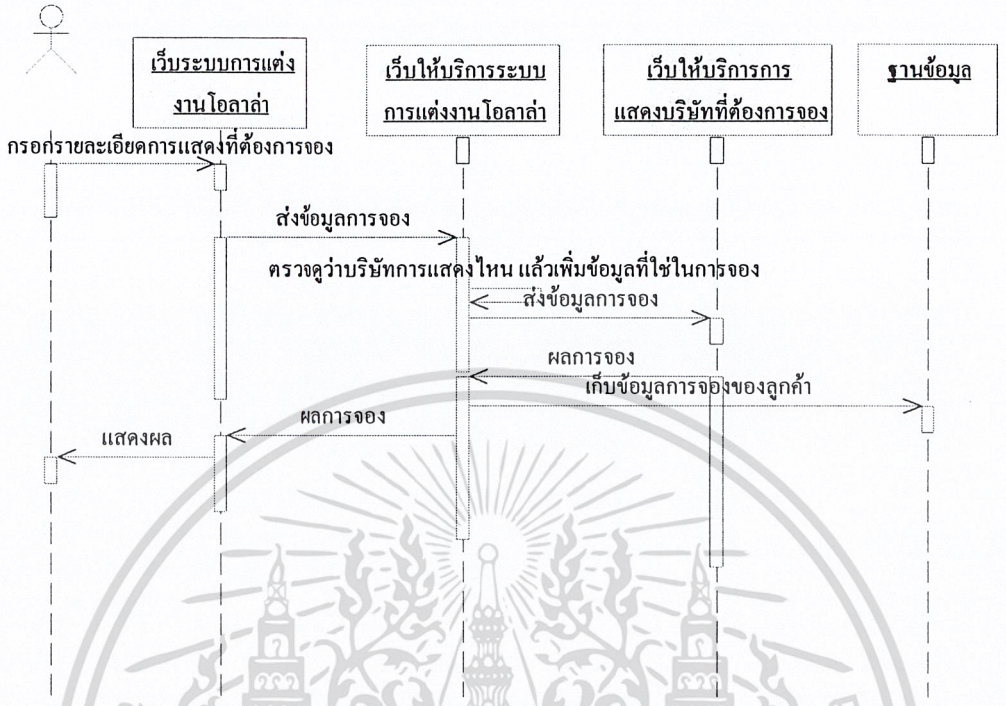


รูปที่ 6-19 แสดงการค้นหาแพคเกจทัวร์

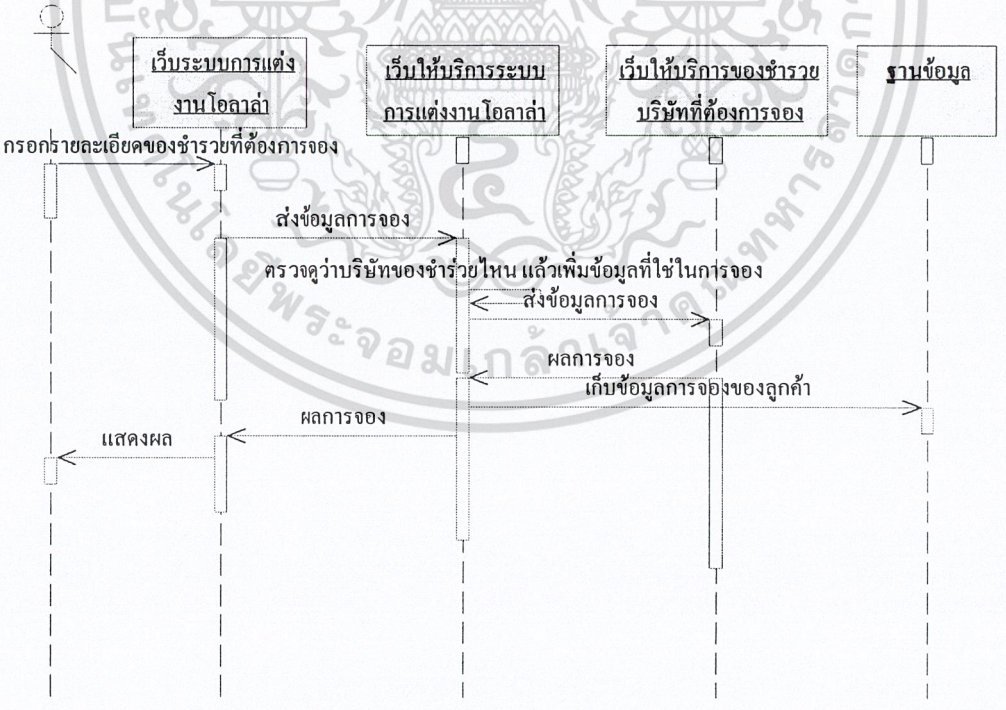


รูปที่ 6-20 แสดงการจองการถ่ายวีดีโอที่งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

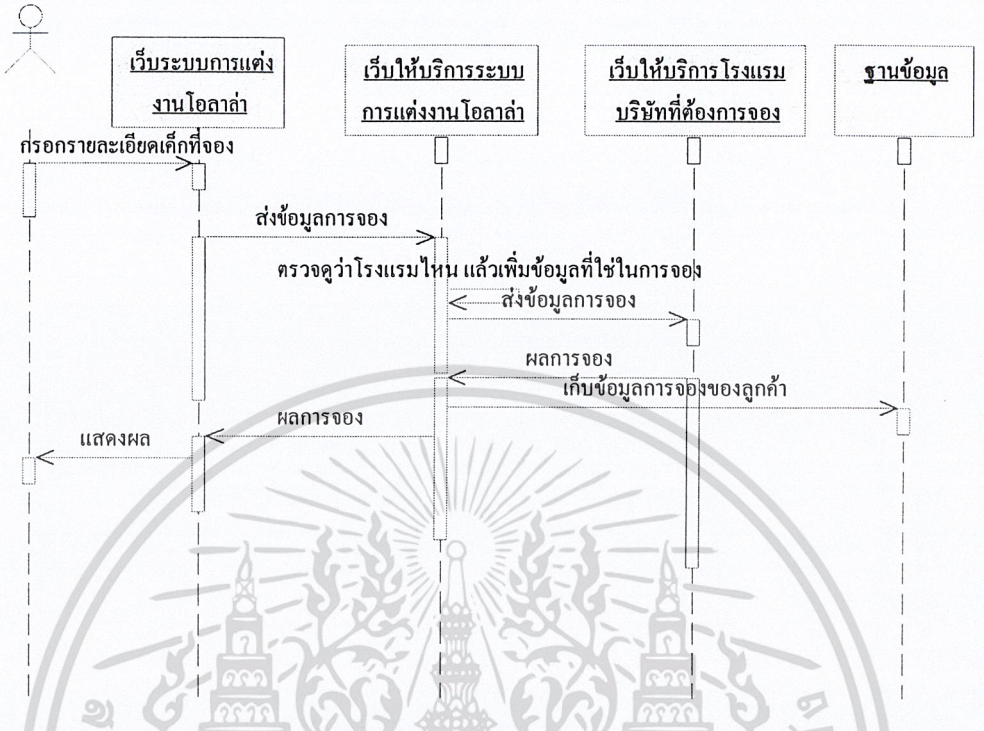


รูปที่ 6-21 แสดงการจองการแสดง

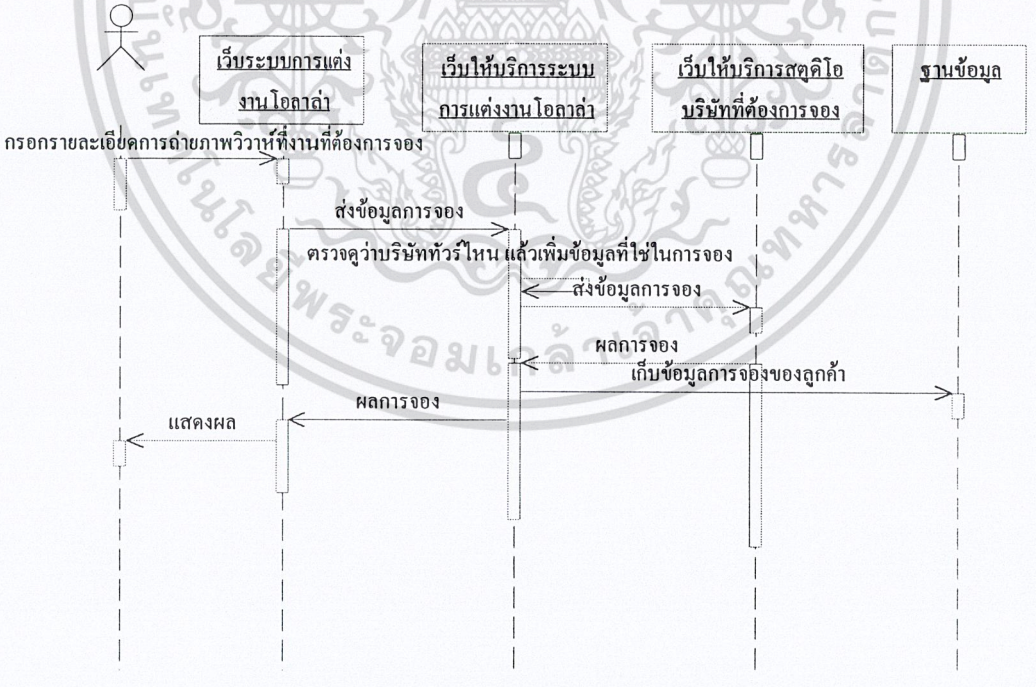


รูปที่ 6-22 แสดงการจองของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

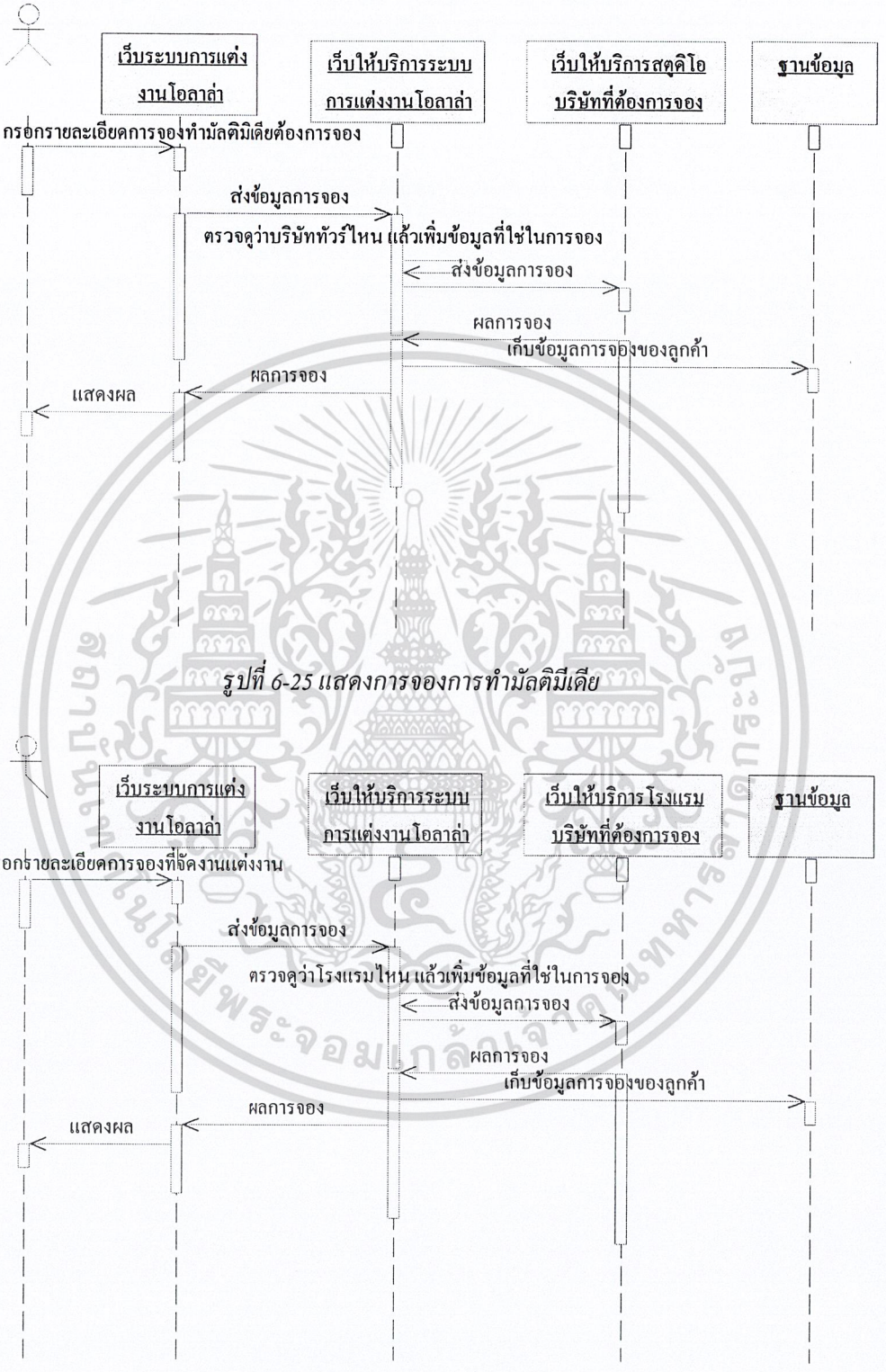


รูปที่ 6-23 แสดงการจองเค้ก



รูปที่ 6-24 แสดงการจองการถ่ายภาพวีดีโอที่งาน

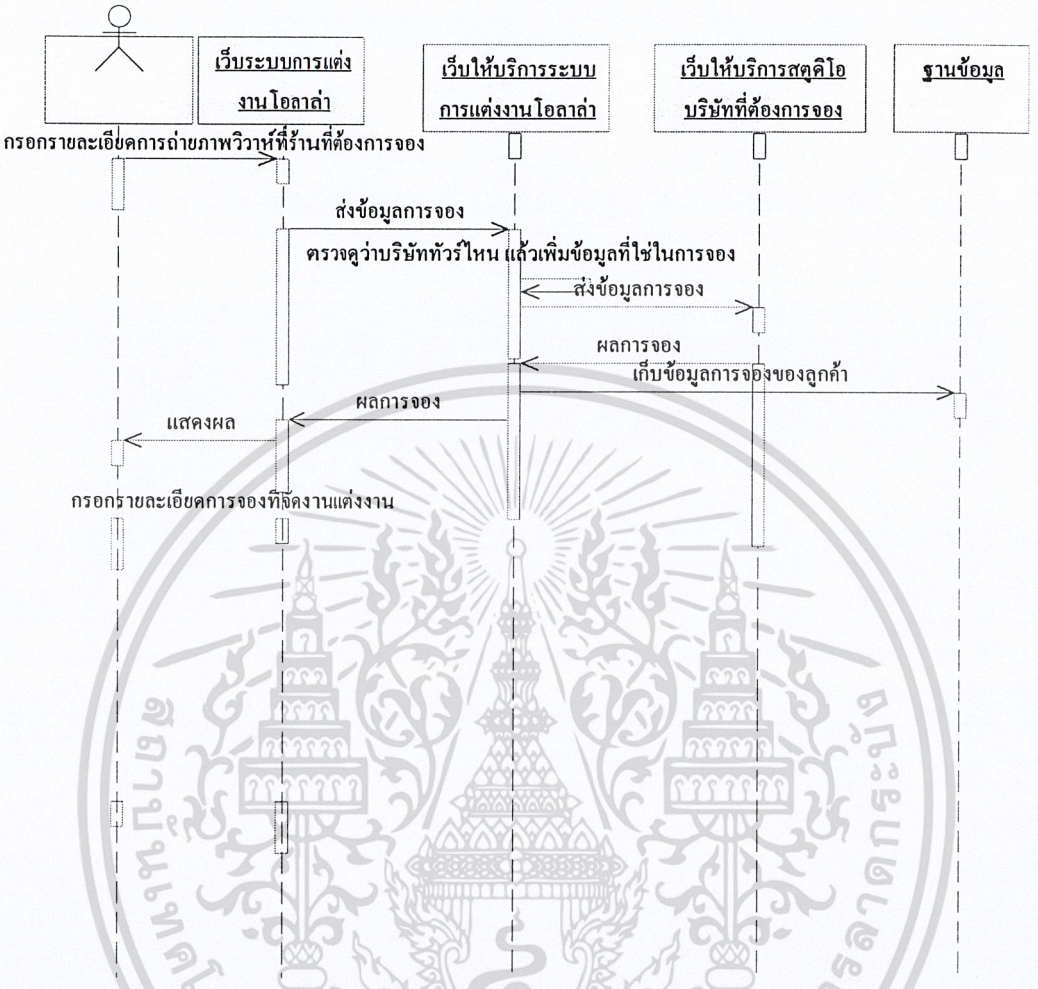
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-25 แสดงการจองการทำมัลติมีเดีย

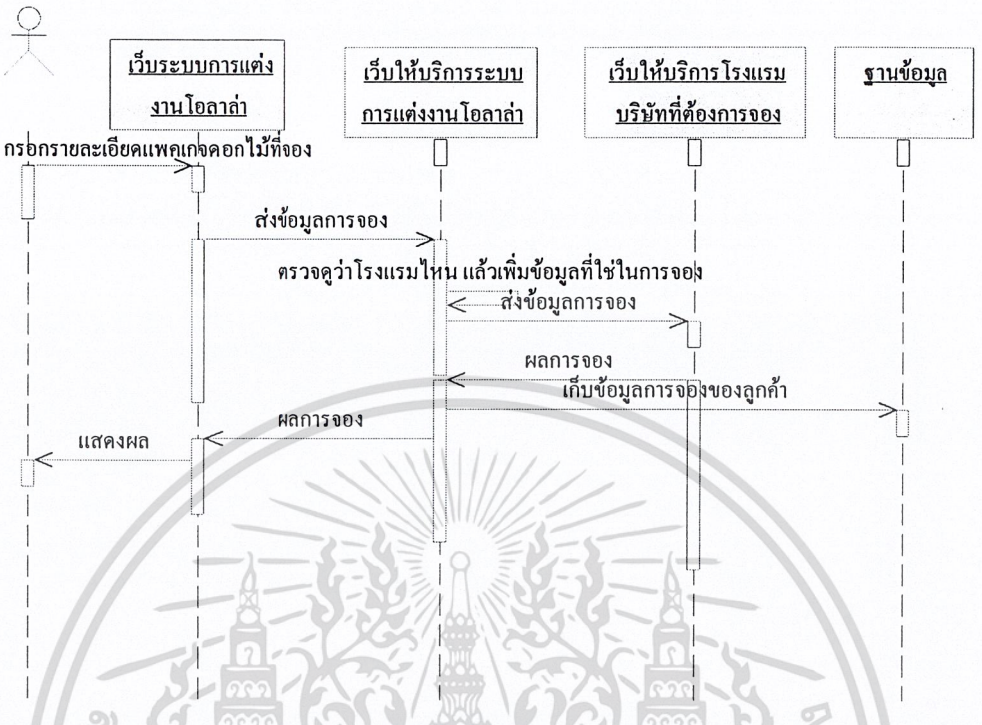
รูปที่ 6-26 แสดงการจองห้องจัดงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

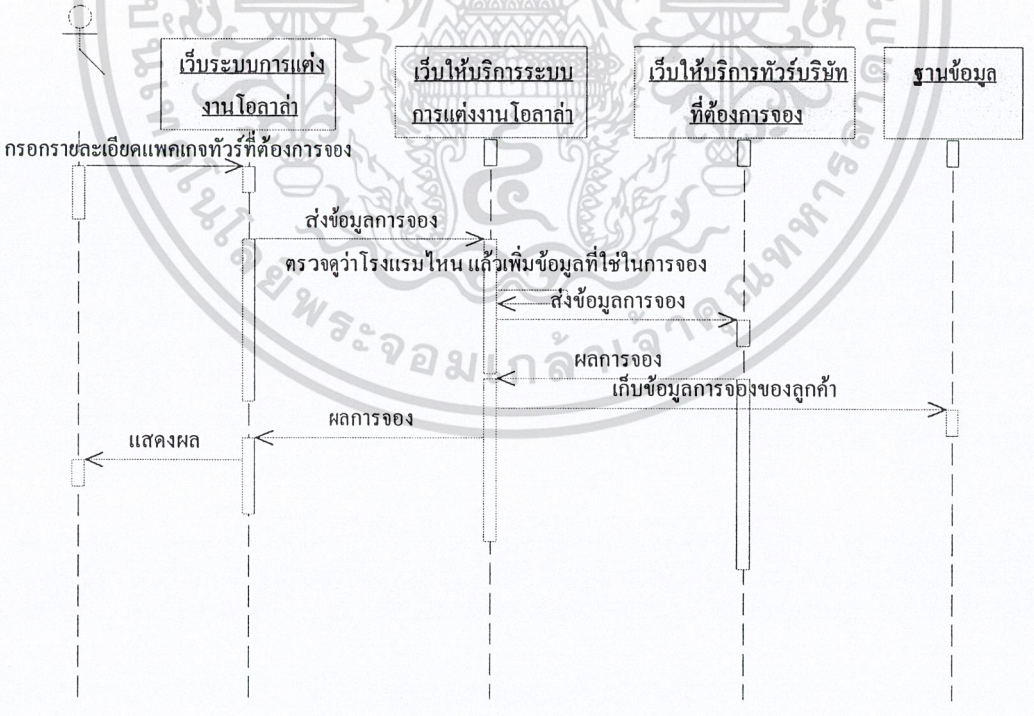


รูปที่ 6-27 แสดงการจองการถ่ายภาพที่ร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

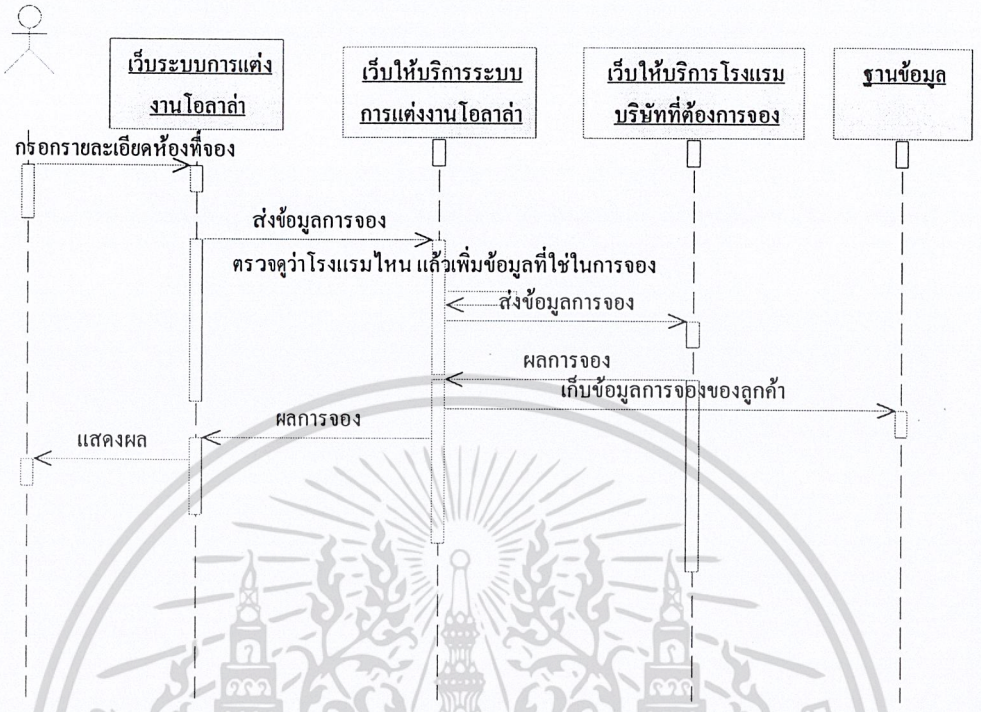


รูปที่ 6-28 แสดงการจองแพคเกจดอกไม้



รูปที่ 6-29 แสดงการจองแพคเกจทัวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

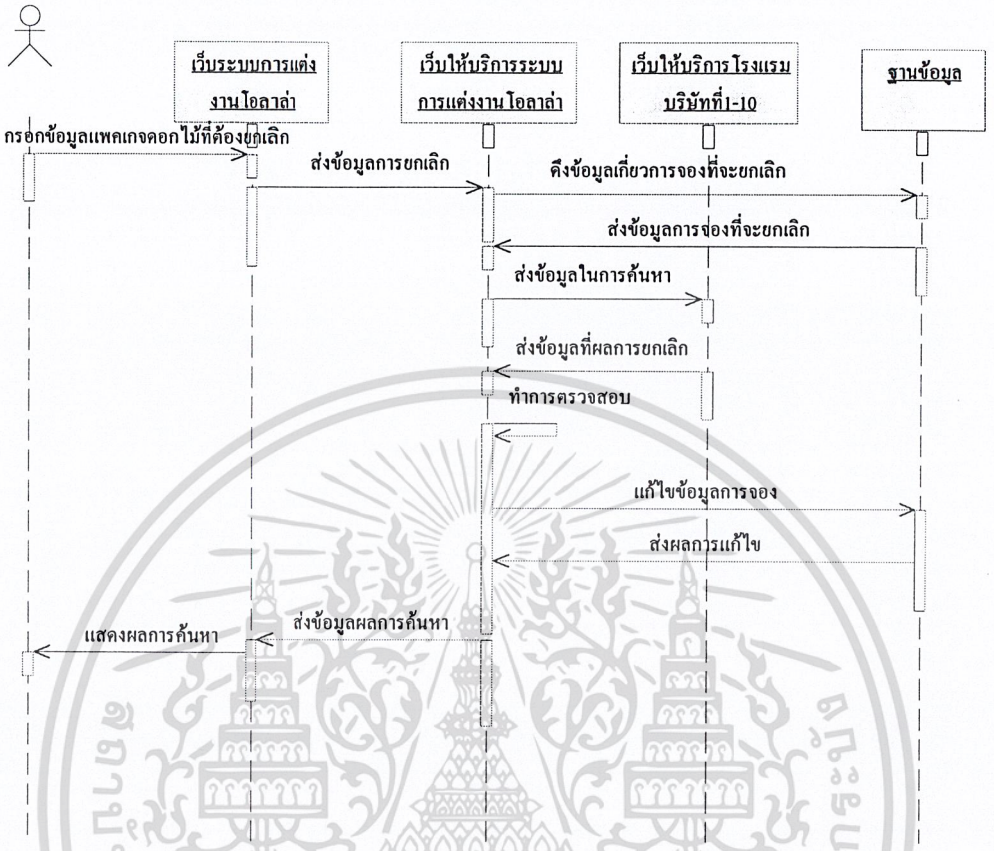


รูปที่ 6-30 แสดงการจองห้องพัก



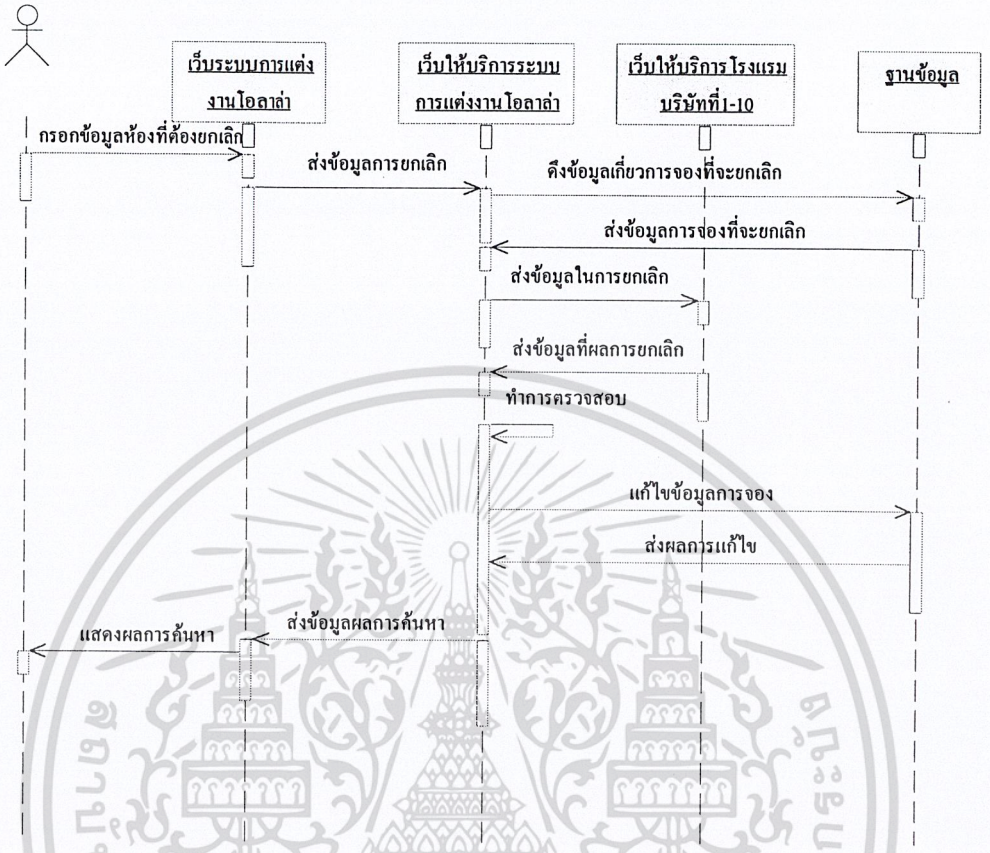
รูปที่ 6-31 แสดงการจองอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



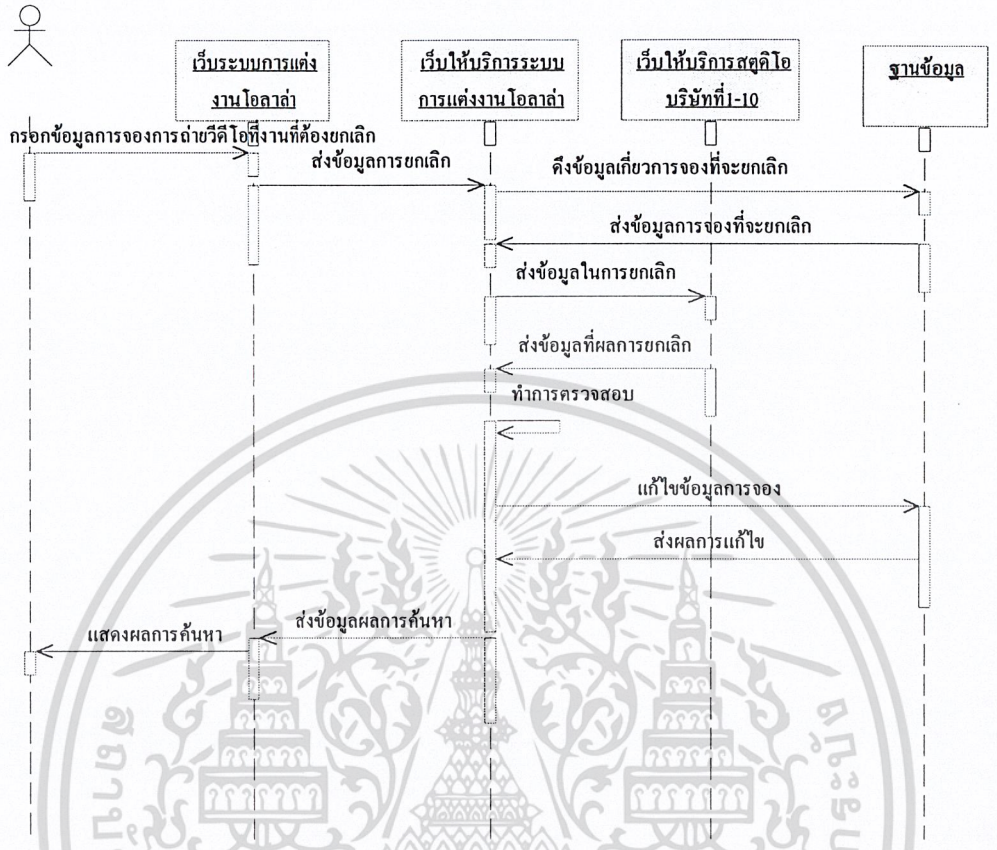
รูปที่ 6-32 แสดงการยกเลิกแพคเกจดอกไม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



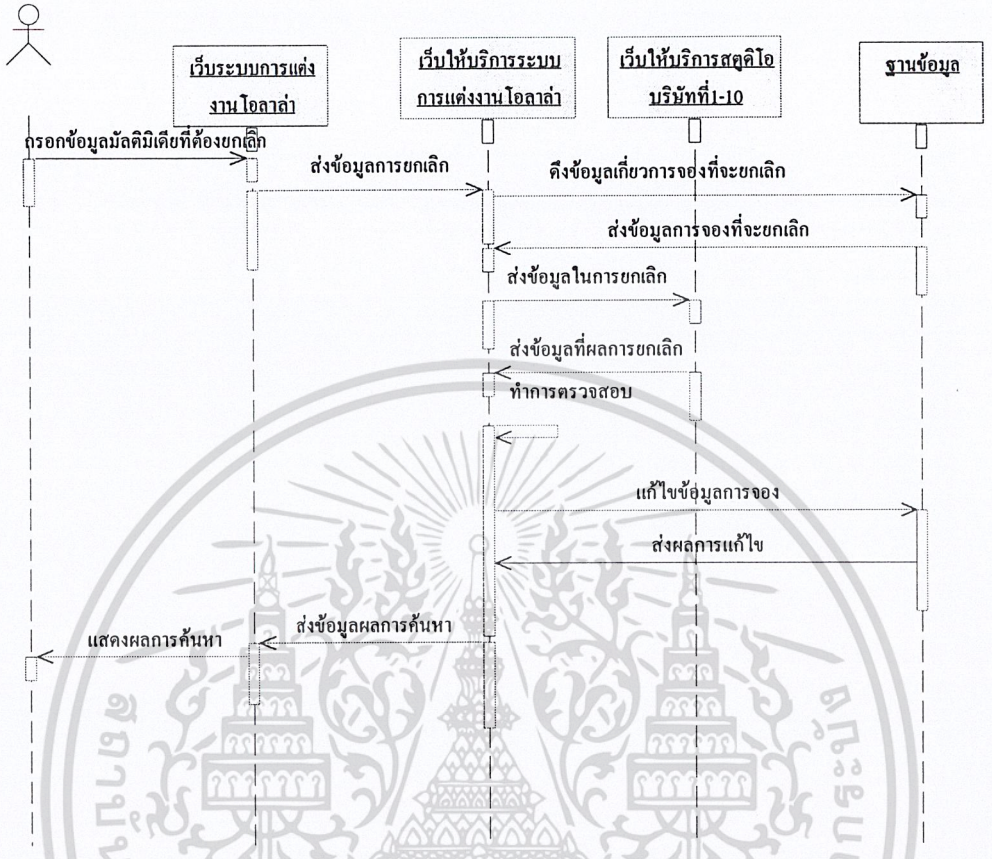
รูปที่ 6-33 แสดงการยกเลิกการจองห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



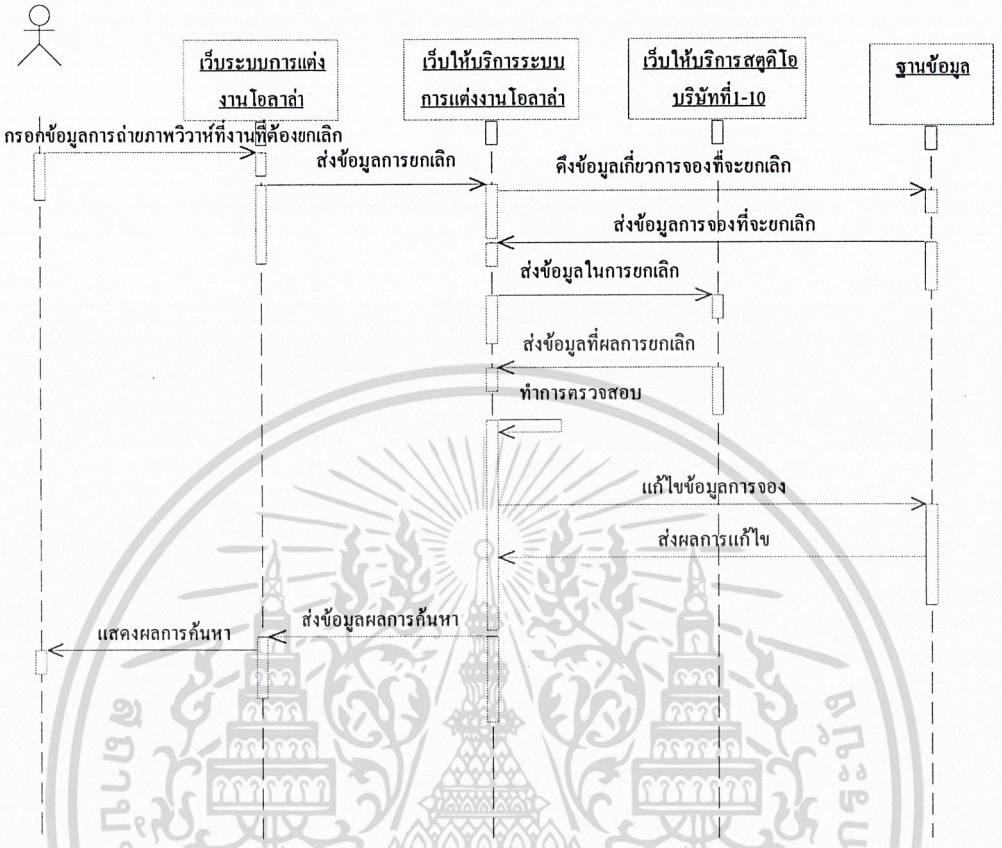
รูปที่ 6-34 แสดงการยกเลิกการถ่ายวีดีโอที่งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



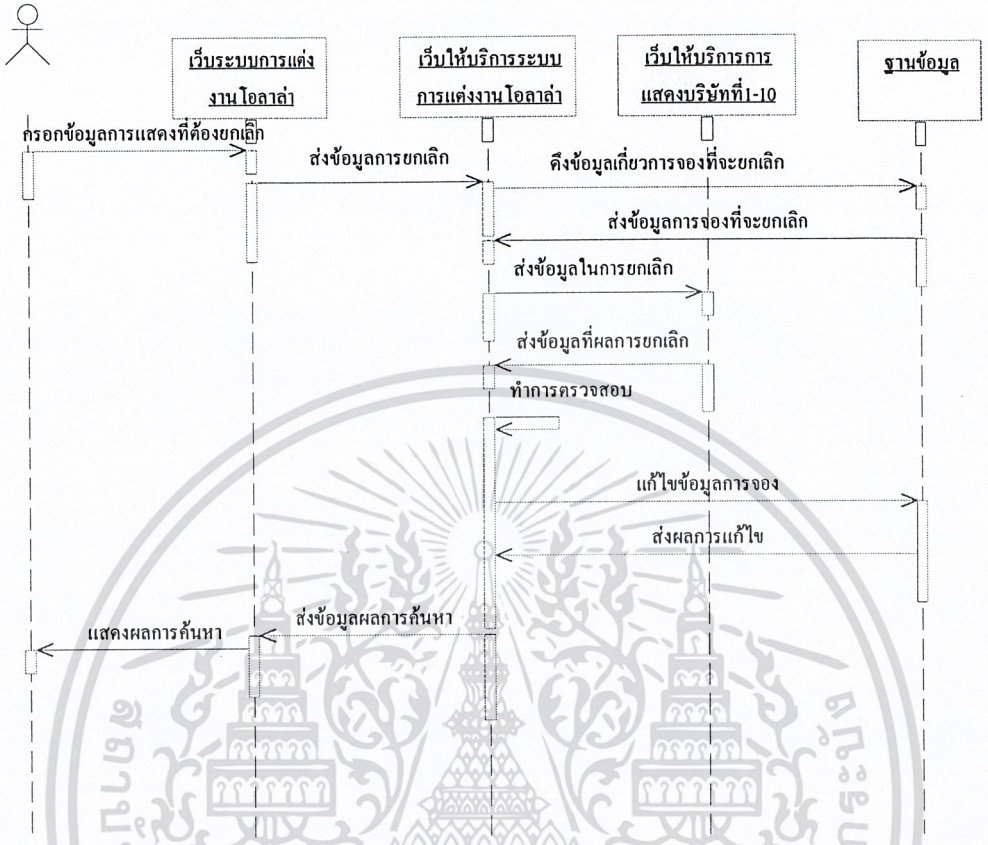
รูปที่ 6-35 แสดงการยกเลิกการทำมัลติมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



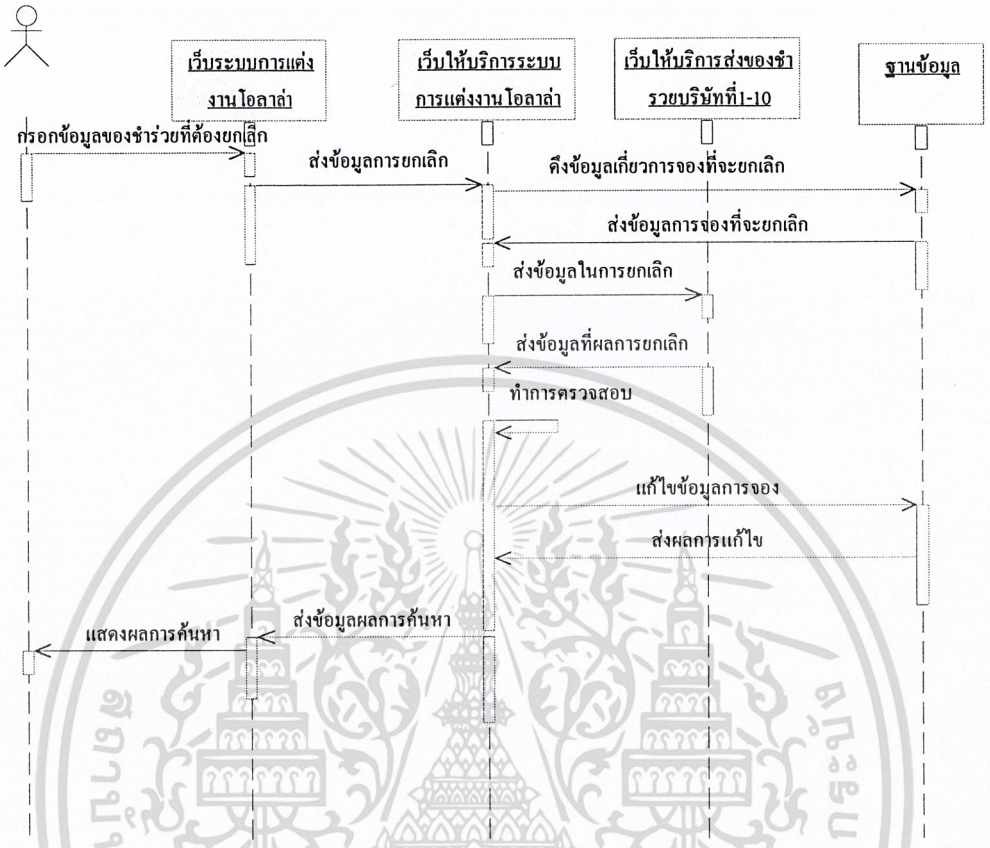
รูปที่ 6-36 แสดงการยกเลิกการถ่ายภาพวีดีโอที่งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



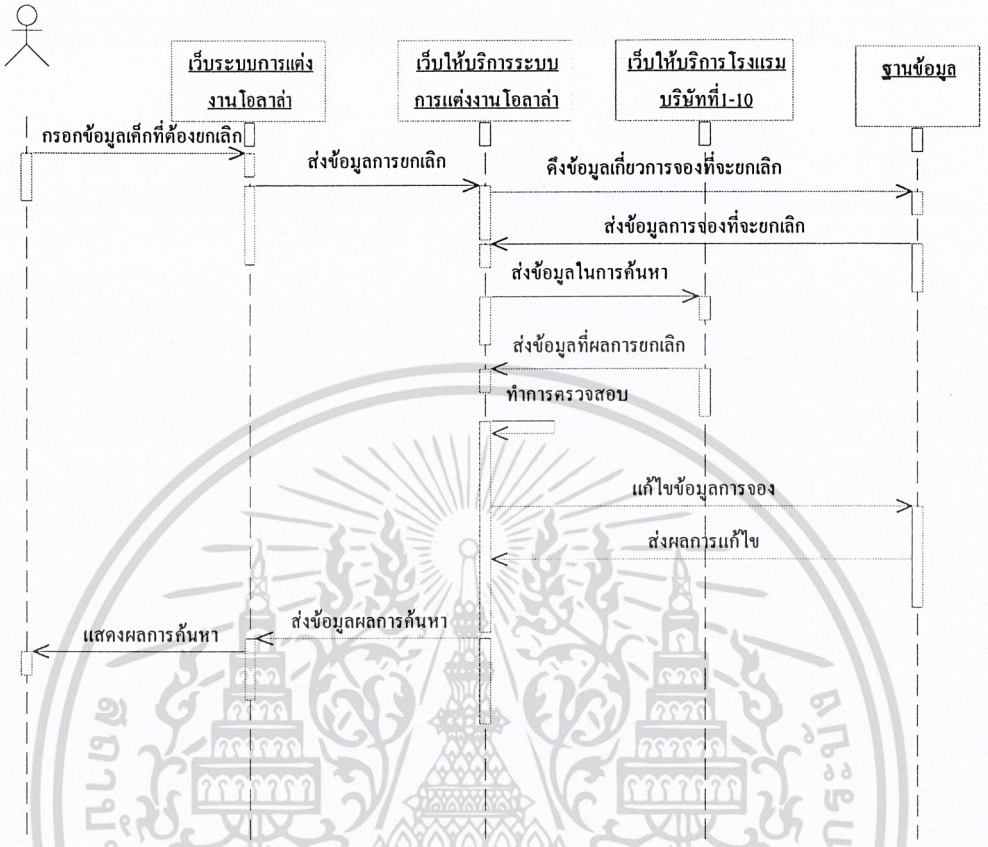
รูปที่ 6-37 แสดการยกเลิกการแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



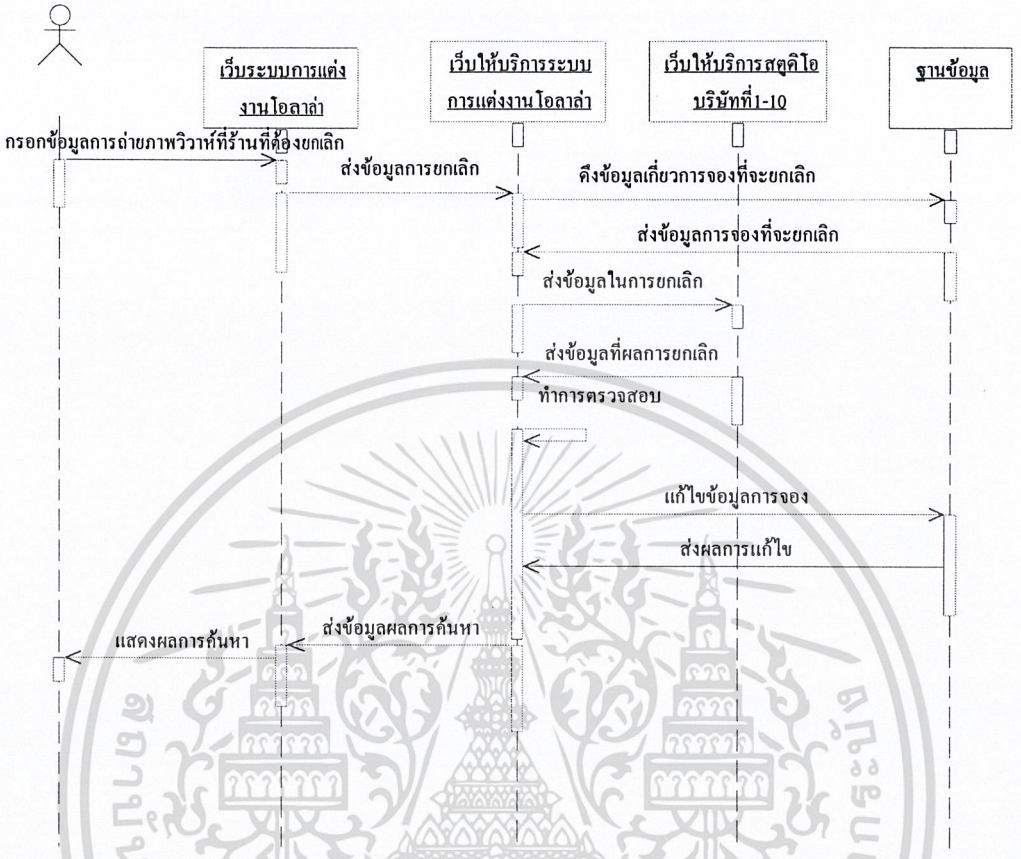
รูปที่ 6-38 แสดงการยกเลิกของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



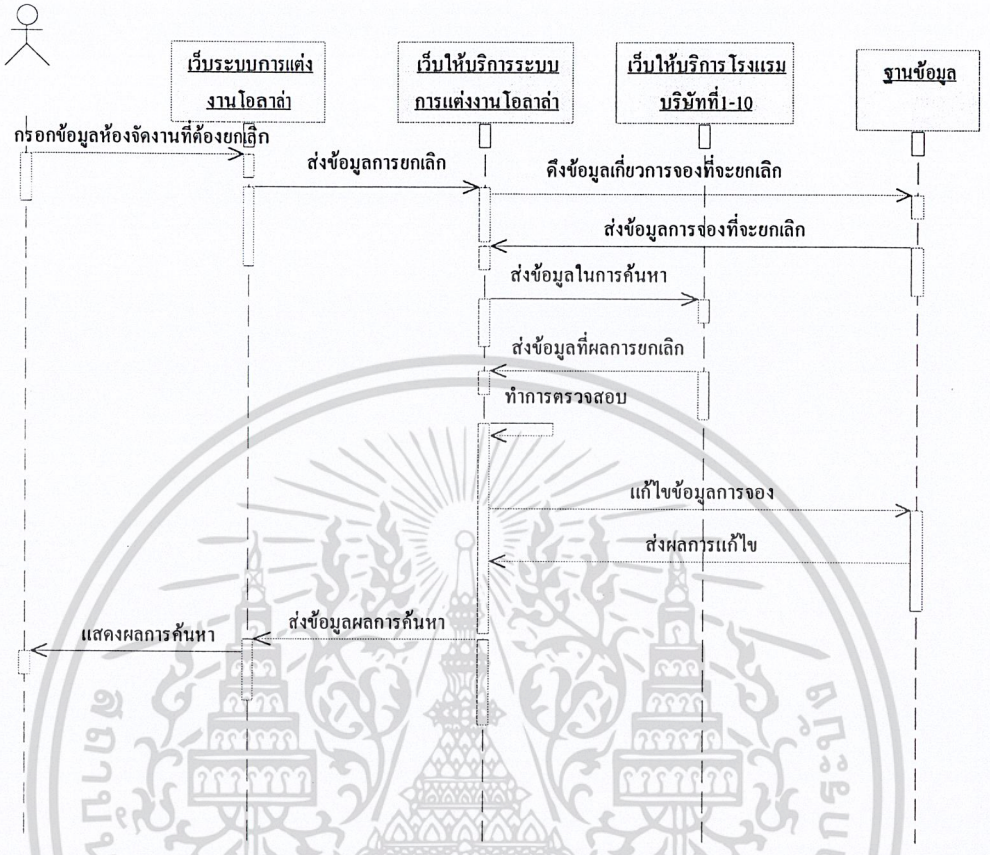
รูปที่ 6-39 แสดงการยกเลิกการสั่งจองเด้ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



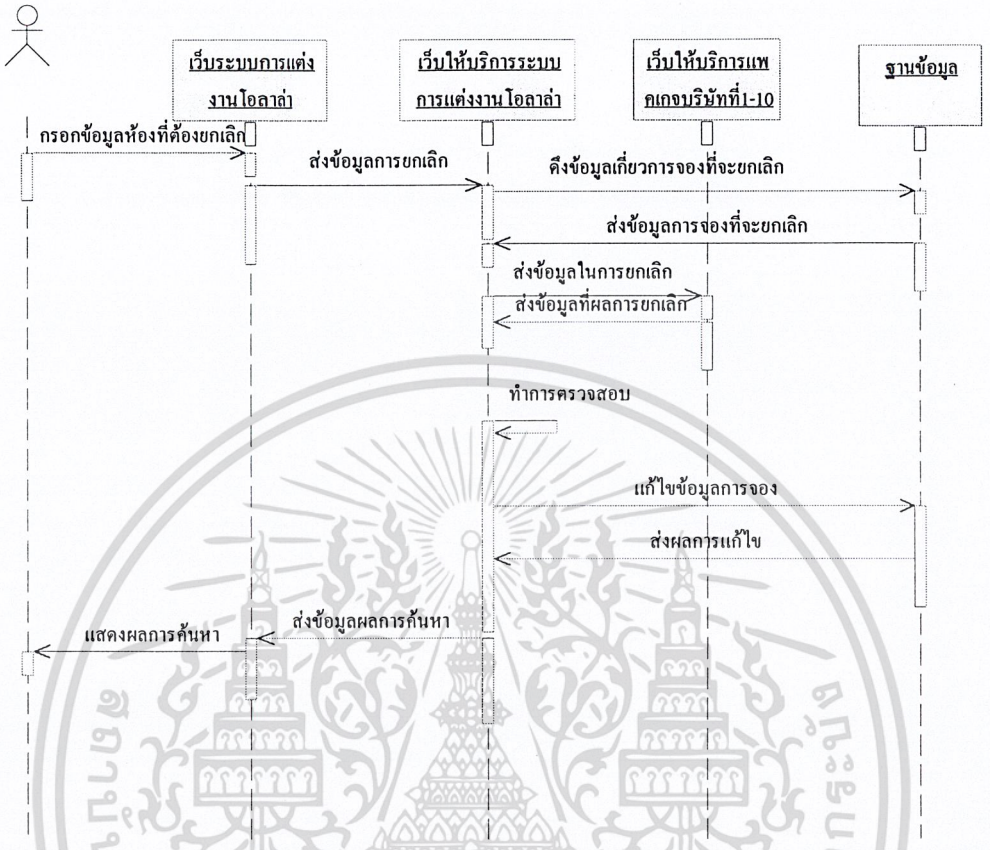
รูปที่ 6-40 แสดงการยกเลิกการถ่ายภาพที่ร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



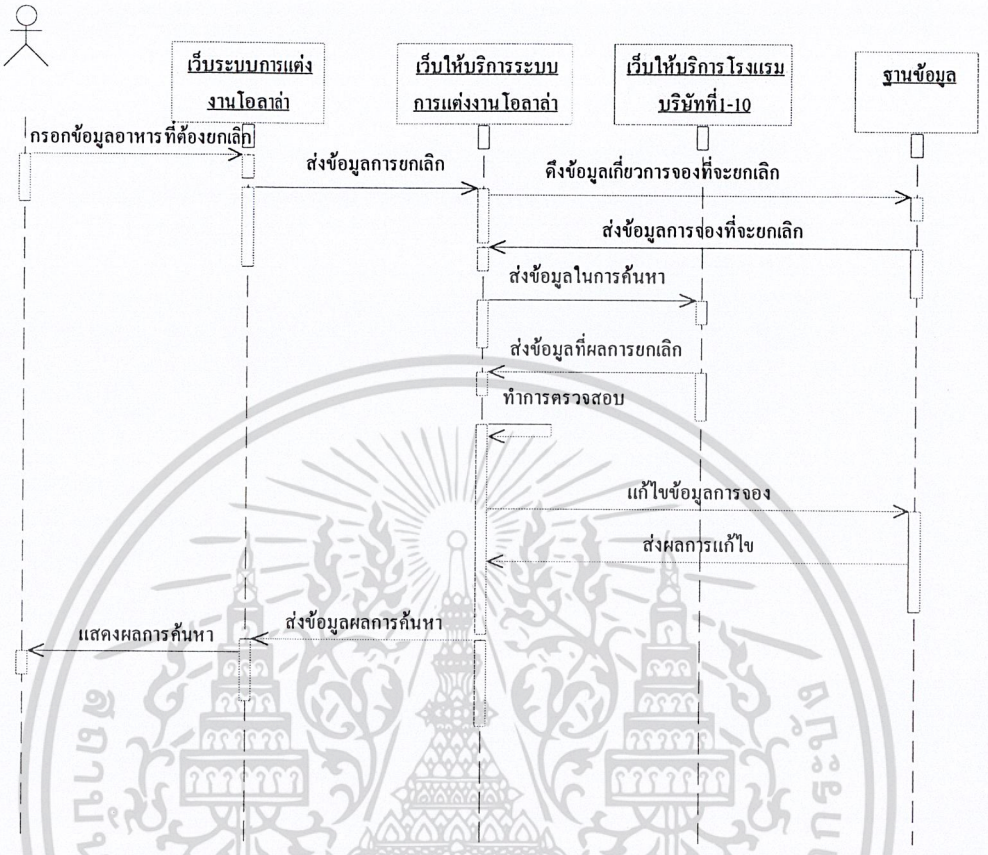
รูปที่ 6-41แสดงการยกเลิกการจองห้องจัดงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



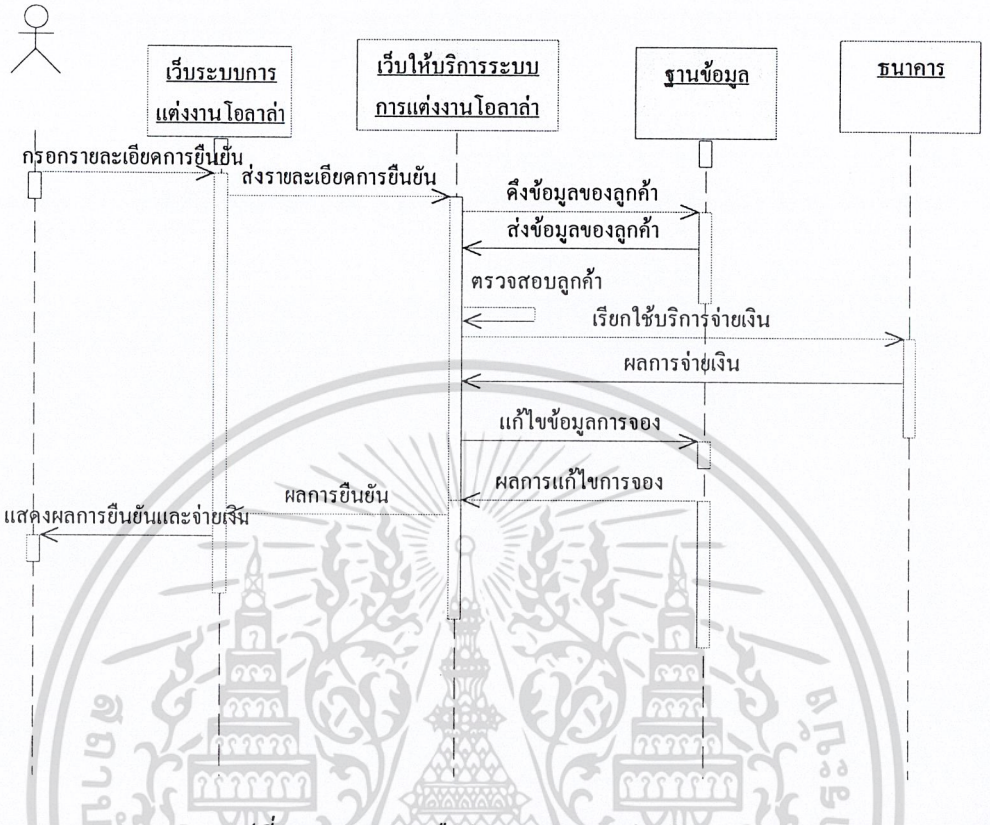
รูปที่ 6-42 แสดงการยกเลิกแพคเกจทัวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

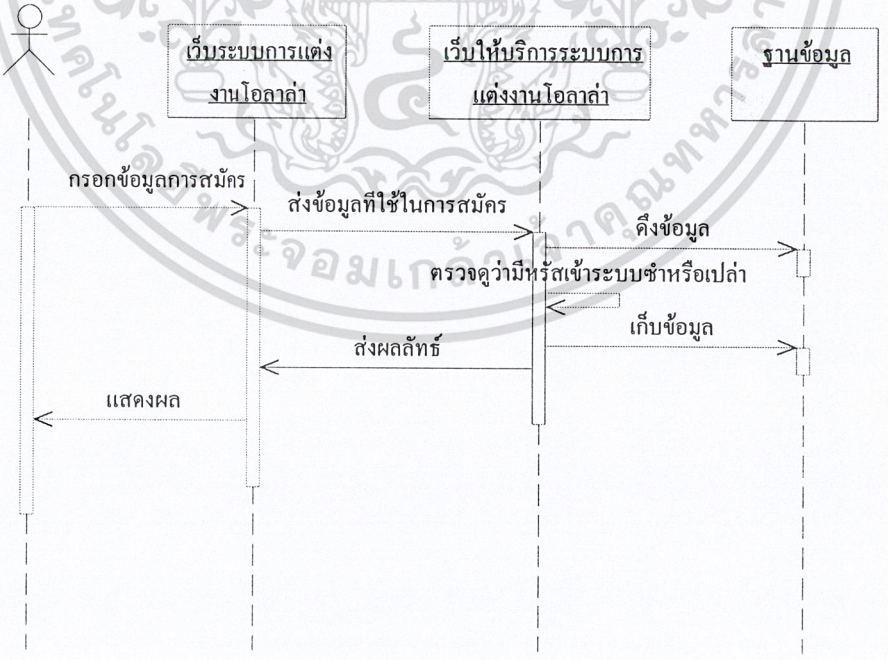


รูปที่ 6-43 แสดงการยกเลิกอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

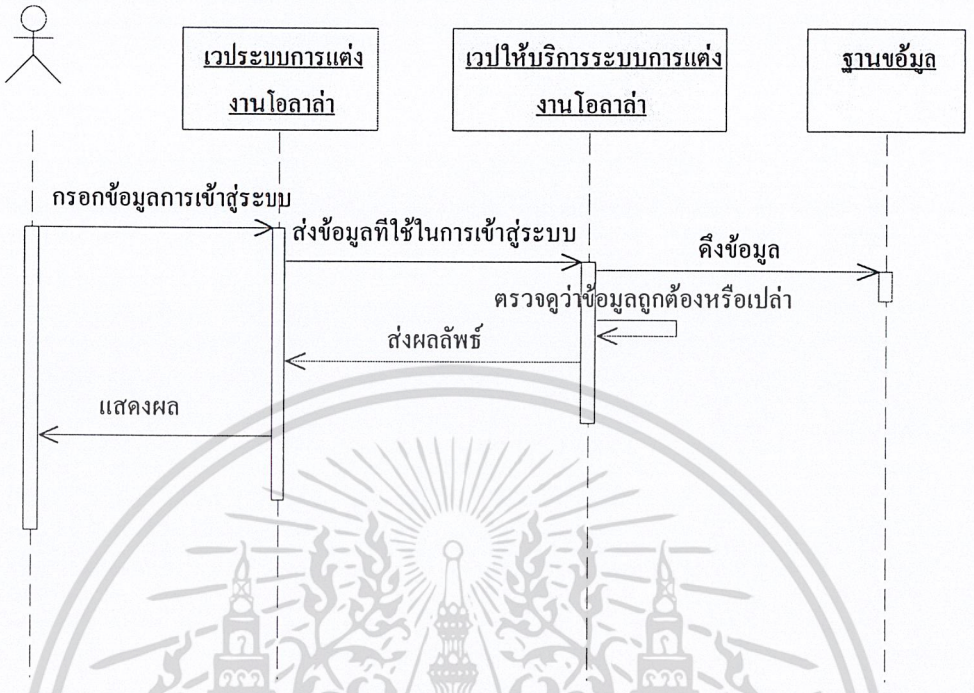


รูปที่ 6-44 แสดงการยืนยันการจองและจ่ายเงิน



รูปที่ 6-45แสดงการลงทะเบียน

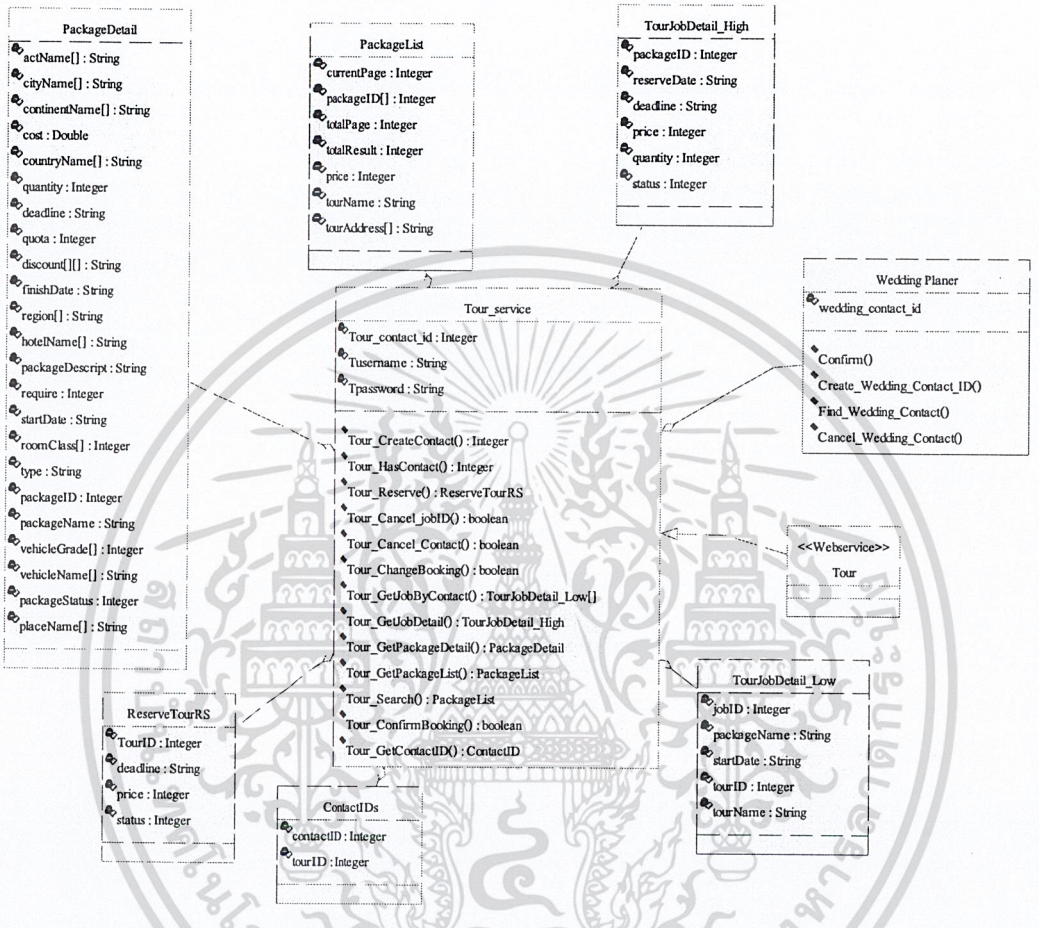
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-46 แสดงการเข้าสู่ระบบ

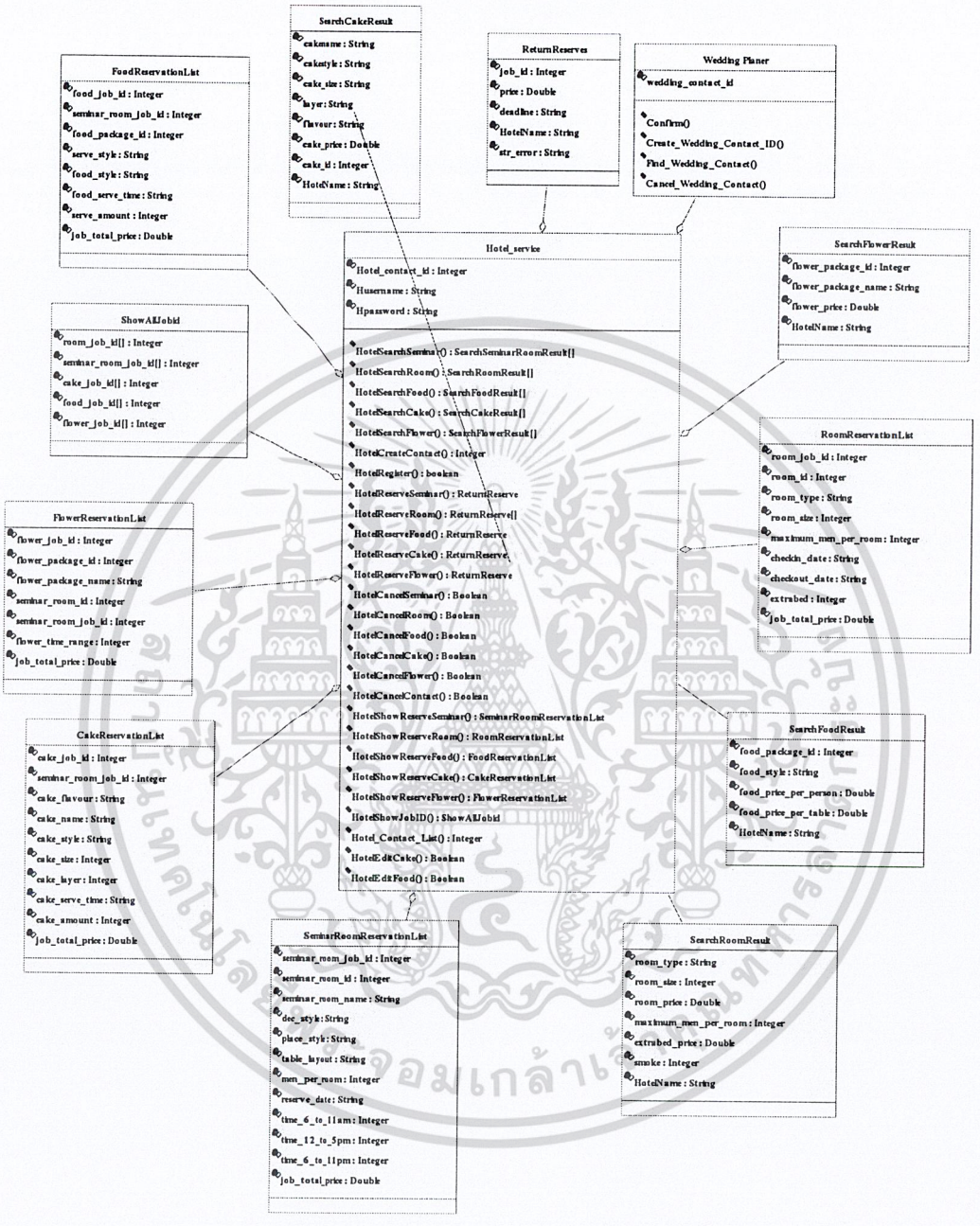
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 คลาสไดอะแกรม (class diagram)



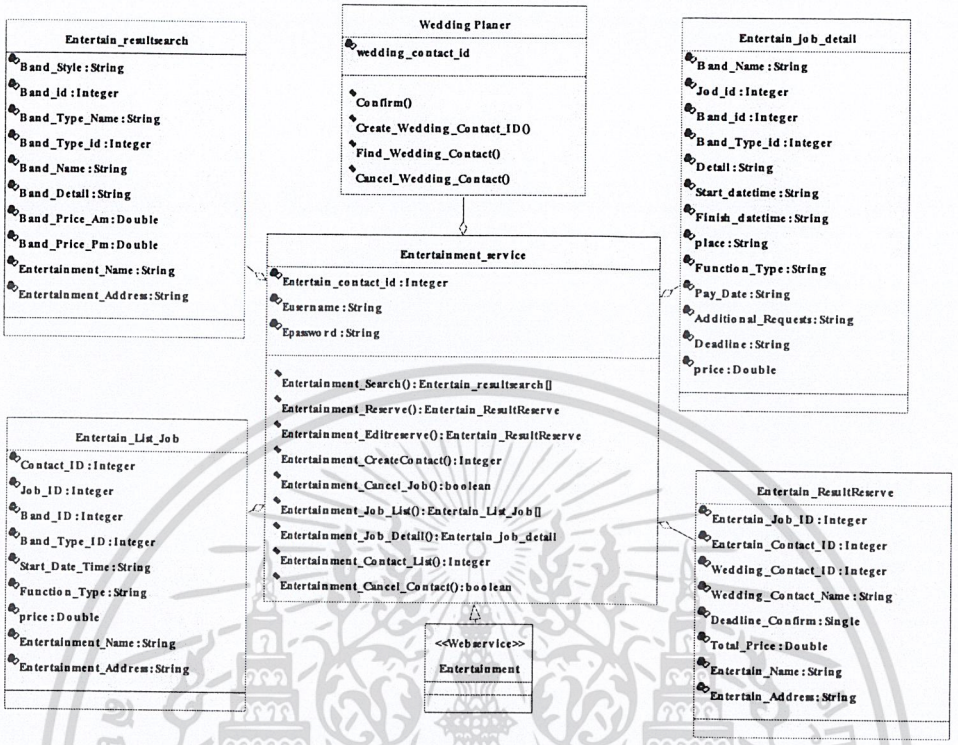
รูปที่ 6-47แสดง คลาสไดอะแกรมของทัวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

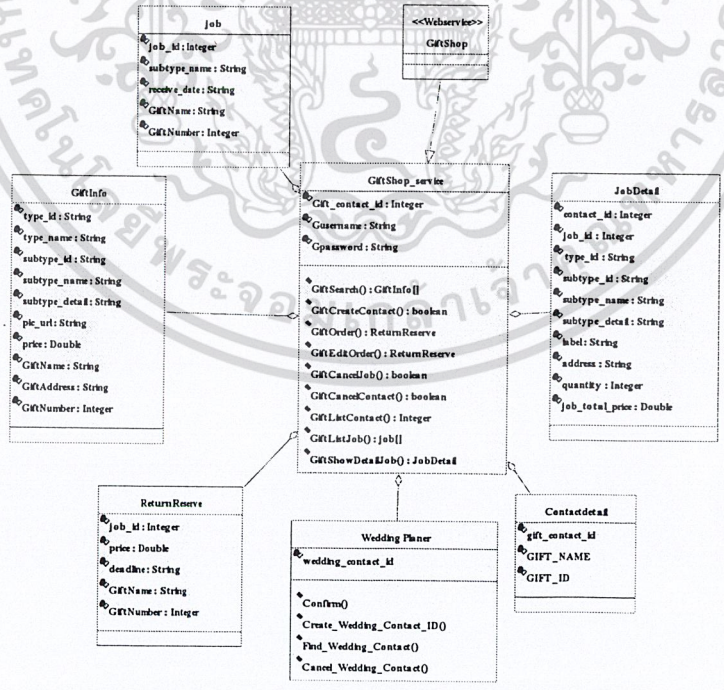


รูปที่ 6-48 แสดงคลาสไดอะแกรมของโรงแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

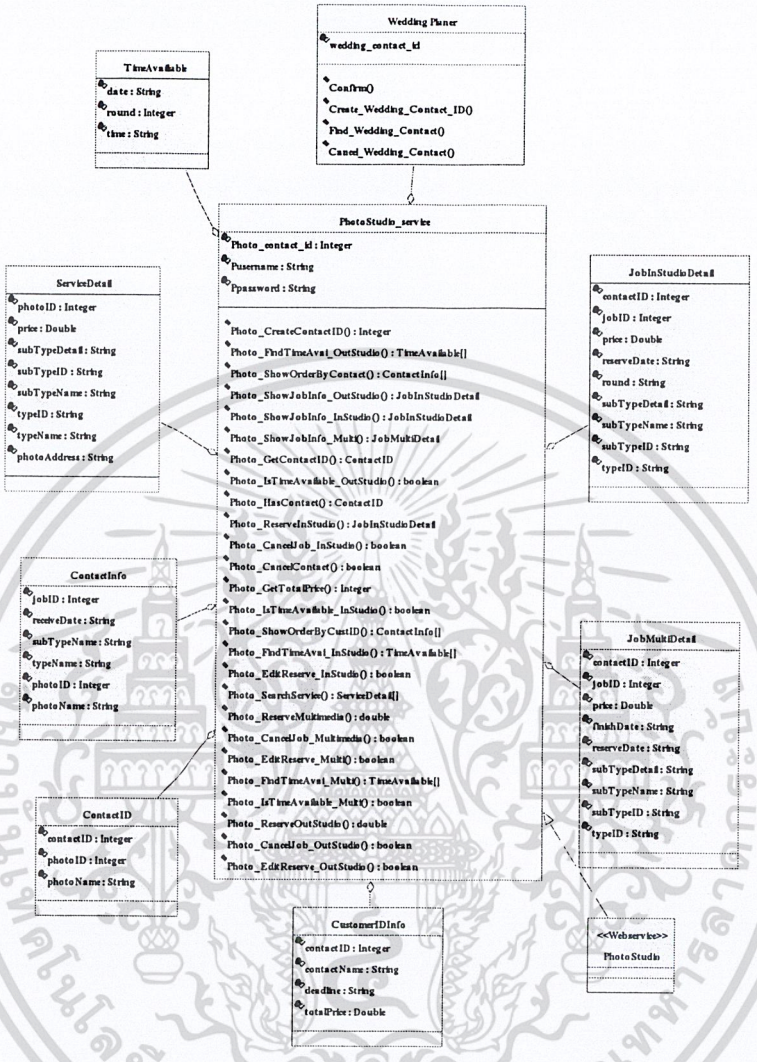


รูปที่ 6-49 แสดงคลาสไดอะแกรมของบริษัทเอ็นเตอร์เทนเมนต์



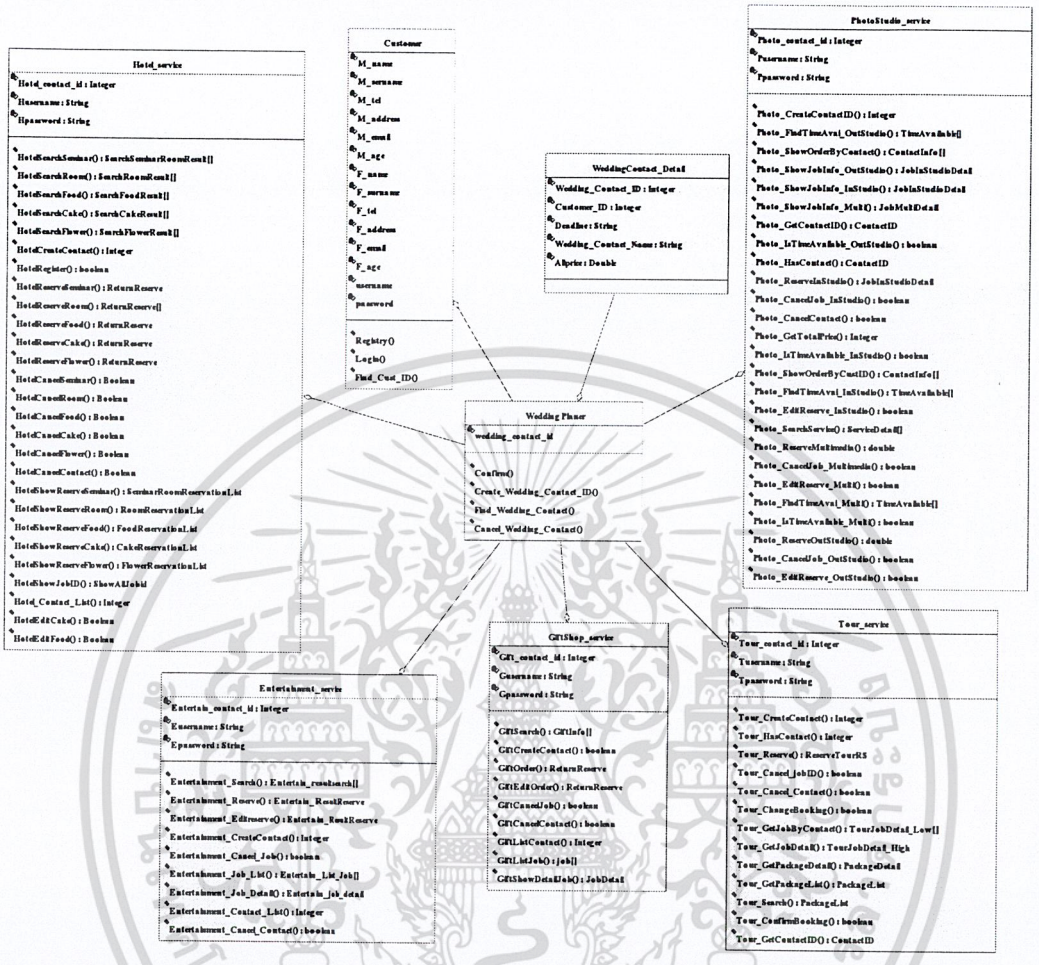
รูปที่ 6-50 แสดงคลาสไดอะแกรมของบริษัทของชำร่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-51 แสดงคลาสไดอะแกรมของโฟโตสตูดิโอ

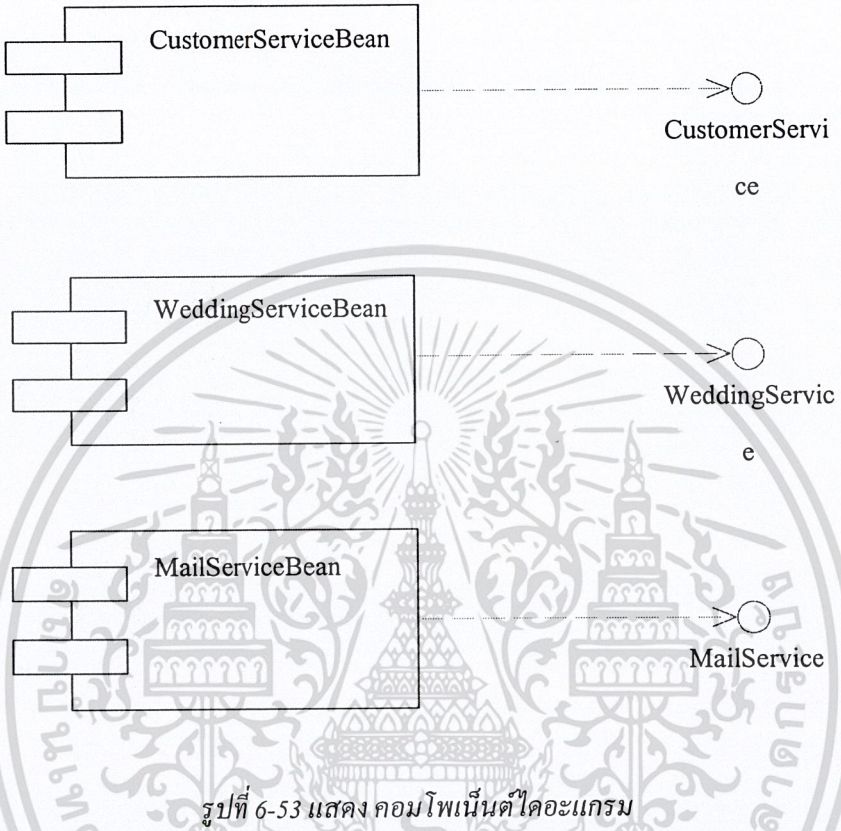
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-52 แสดงคลาสไดอะแกรมของ Olala Wedding Planner

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 คอมโพเนนต์ไดอะแกรม (Component Diagram)



รูปที่ 6-53 แสดง คอมโพเน็นต์ไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8 รายละเอียดของฟังก์ชัน

6.8.1 ฟังก์ชันของส่วนกลาง(Olala Wedding Planner)

1. ฟังก์ชันสมัครสมาชิก

Function : Register(String M_name, String M_surname, String M_birthdate, String M_phone, String M_address, String M_email, String F_name, String F_surname, String F_birthdate, String F_phone, String F_address, String F_email, String username, String password)

Return : Customer ID ถ้าไม่สามารถสมัครได้จะมีค่า -1

2. ฟังก์ชันสร้างเว็ดดิ้งคอนแทค(Wedding Contact ID)

Function : Create_Wedding_Contact_ID(int Customer_ID,String Wedding_Contact_name)

Return : Wedding Contact ID ถ้าไม่สามารถสมัครได้จะมีค่า -1

3. ฟังก์ชันแสดง รหัสสมาชิก(Customer ID)

Function : Find_Cust_ID(String username,String password)

Return : Customer ID ถ้าผิดพลาดจะมีค่า -1

4. ฟังก์ชันแสดงรายละเอียดของเว็ดดิ้งคอนแทค

Function : Find_Wedding_Contact(int CustID)

Return : public static class WeddingContact_Detail implements Serializable{

```
public int Wedding_Contact_ID;
public int Customer_ID;
public String Deadline;
public String Wedding_Contact_Name;
public double Allprice;
}
```

5. ฟังก์ชันล็อกอิน

Function : Login(String username,String password)

Return : Customer ID ถ้าไม่สามารถล็อกอินได้จะมีค่า -1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ฟังก์ชันยกเลิกเว็ดดิ้งคอนแทค

Function : cancel_wedding_contact(int coustomer_id,int wedding_contact_id)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

6.8.2 ฟังก์ชันของบริษัทเอ็นเตอร์เทนเมนต์

1. ฟังก์ชันสมัครสมาชิก

Function : Entertainment_Register(int EntertainNumber,String username , String password,String EntertainName)

Return : ผลลัพธ์ว่าสมัครสำเร็จหรือไม่(“success”,”fail”)

2. ฟังก์ชันค้นหาการแสดง

Function : Entertainment_Search(String style,int type,double minprice,double maxprice,String startdate,String finishdate,boolean cheap,String near,int EntertainmentNumber)

Return : public static class Entertain_resultsearch implements Serializable{

```
public String Band_Style;
public int Band_id;
public String Band_Type_Name;
public int Band_Type_id;
public String Band_Name;
public String Band_Detail;
public double Band_Price_Am;
public double Band_Price_Pm;
public int Entertain_id;
public String Entertainment_Name;
public String Entertainment_Address;
}
```

3. ฟังก์ชันแสดงคอนแทคของลูกค้า

Function : Entertainment_Contact_List(int Customer_id,int Wedding_contact_id)

Return : Contact ID ถ้าไม่มีมีค่า -1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ฟังก์ชันสร้างคอนแทคใหม่

Function : Entertainment_CreateContact(int wedding_contact_id,int EntertainNumber,int Customer_id)

Return : Contact ID ถ้าไม่สำเร็จมีค่า -1

5. ฟังก์ชันจองการแสดง

Function : Entertainment_Reserve(int Customer_ID, int Wedding_Contact_ID, int Entertain_Contact_ID, int Band_id, int Band_Type_ID, String startdate, String finishdate, String place, String function_type, double Price_AM, double Price_PM, String Additional_Requests, int EntertainmentNumber,String Band_Contact_Name)

Return : public static class Entertain_ResultReserve implements Serializable{

```
public int Entertain_Job_ID;
public int Entertain_Contact_ID;
public int Wedding_Contact_ID;
public String Wedding_Contact_Name;
public String Deadline_Confirm;
public double Total_Price;
public String Entertain_Name;
public String Entertain_Address;
}
```

6. ฟังก์ชันแก้ไขข้อมูลการจอง

Function : Entertainment_Editreserve(int Customer_ID,int Wedding_Contact_ID,int Entertain_Contact_ID,int Entertain_Job_ID,String Start_Date,String Finish_Date,String Place,String Function_Type,String Additional_Requests,int Band_id,int Band_type_id)

Return : public static class Entertain_ResultReserve implements Serializable{

```
public int Entertain_Job_ID;
public int Entertain_Contact_ID;
public int Wedding_Contact_ID;
public String Wedding_Contact_Name;
public String Deadline_Confirm;
public double Total_Price;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public String Entertain_Name;

public String Entertain_Address;

}

7. ฟังก์ชันแสดงรายการ job ทั้งหมด

Function : Entertainment_Job_List(int Customer_id,int Wedding_contact_id)

Return : public static class Entertain_List_Job implements Serializable{

public int Contact_ID;

public int Job_ID;

public int Band_ID;

public int Band_Type_ID;

public String Start_Date_Time;

public String Function_Type;

public double price;

public String Entertainment_Name;

public String Entertainment_Address;

}

8. ฟังก์ชันแสดงรายละเอียดของ job

Function : Entertainment_Job_Detail(int Wedding_contact_id,int Entertain_contact_id,int Entertain_job_id)

Return : public static class Entertain_job_detail implements Serializable{

public String Band_Name;

public int Jod_id;

public int Band_id;

public int Band_Type_id;

public String Detail;

public String Start_datetime;

public String Finish_datetime;

public String place;

public String Function_Type;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    public String Pay_Date;
    public String Additional_Requests;
    public String Deadline;
    public double price;
}
```

9. ฟังก์ชันยกเลิกการจองเฉพาะ job

Function : Entertainment_Cancel_Job(int Customer_ID,int Wedding_Contact_ID,int Entertain_Contact_ID,int Entertain_Job_ID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

10. ฟังก์ชันยกเลิกการจองทั้งคอนแทค

Function : Entertainment_Cancel_Contact(int Customer_ID, int Entertain_Contact_id , int Wedding_contactID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

6.8.3 ฟังก์ชันของบริษัททัวร์

1. ฟังก์ชันสมัครสมาชิก

Function : public boolean Tour_Subscribe(String tourName,int TourID,String Name,String Surname,String Email,String Username,String Password,String Address,String TAddress,String TPhone)

Return : boolean บอกว่าสามารถสมัครได้หรือไม่

2. ฟังก์ชันค้นหาแพคเกจที่ตรงกับความต้องการ

Function : public PackageList Tour_Search(int tourID,String placeID,String region,String cityID,String countryID,String continentID,String vehicleID,String vehicleGrade,String hotelID,String HotelGrade,String actID,String type,int minQuota,int maxQuota,int minRequire,int maxRequire,int minCost,int maxCost,int minStartDay,int minStartMonth,int minstartYear,int maxStartDay,int maxStartMonth,int maxStartYear,int minFinishDay,int minFinishMonth,int minFinishYear,int maxFinishDay,int maxFinishMonth,int maxFinishYear,int currentPage,int resultPerPage,boolean cheap)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Return : public static class PackageList implements Serializable{

```
public int currentPage;
public int packageID[];
public int totalPage;
public int totalResult;
public int price[];
public String tourName[];
public String tourAddress[];
};
```

3. ฟังก์ชันแสดงรายละเอียดของ package

Function : Tour_GetPackageDetail(int packageID,int TourID)

Return : public static class PackageDetail implements Serializable{

```
public String actName[];
public String cityName[];
public String continentName[];
public int cost;
public String countryName[];
public String deadline;
public int discount[][];
public String finishDate;
public String hotelName[];
public String packageDescript;
public int packageID;
public String packageName;
public int packageStatus;
public String placeName[];
public int quantity;
public int quota;
public String region[];
public int require;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        public int roomClass[];
        public String startDate;
        public String type;
        public int vehicleGrade[];
        public String vehicleName[];
    };
```

4 ฟังก์ชันแสดงแพคเกจทั้งหมดของบริษัททัวร์

Function : Tour_GetPackageList(int tourID,int currentPage,int resultPage)

Return : public static class PackageList implements Serializable{

```
        public int currentPage;
        public int packageID[];
        public int totalPage;
        public int totalResult;
        public int price[];
        public String tourName[];
        public String tourAddress[];
    };
```

5. ฟังก์ชันหา Contact ID ของลูกค้าคนนั้นๆ

Function : Tour_GetContactID(int weddingContactID)

Return : public static class ContactID implements Serializable{

```
        public int contactID;
        public int tourID;
    };
```

6. ฟังก์ชันตรวจสอบว่าลูกค้ามีคอนแทคอยู่หรือยัง

Function : Tour_HasContact(int Wedding_ContactID)

Return : หมายเลขคอนแทค ถ้ามีจะเป็น -1

7. ฟังก์ชันสร้างคอนแทค(Contact)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function : Tour_CreateContact(int Wedding_Contact_ID,int Cust_ID,int TourID)

Return : ค่า Contact ID ถ้ามีค่า <1 แสดงว่าไม่สามารถสร้างได้

8. ฟังก์ชันจองแพคเกจทัวร์

Function : public ReserveTourRS Tour_Reserve(int CustID,int Wedding_ContactID,int Tour_ContactID,int TourID,int PackageID,int Quantity)

Return : public static class ReserveTourRS implements Serializable{

```
public int TourID;
public String deadline;
public int price;
public int status;
}
```

9. ฟังก์ชันเปลี่ยนแปลงการจอง

Function : Tour_ChangeBooking(int Wedding_ContactID,int contactID,int jobID,int packageID,int Quantity)

Return : ราคาใหม่

10. ฟังก์ชันหา job ทั้งหมดของ Contact ID นั้น

Function : Tour_GetJobByContact(int ContactID,int Wedding_ContactID)

Return : public static class TourJobDetail_Low implements Serializable{

```
public int jobID;
public String packageName;
public String startDate;
public int tourID;
public String tourName;
};
```

11. ฟังก์ชันแสดงรายละเอียดของ job ID นั้น

Function : Tour_GetJobDetail(int weddingContactID,int jobID)

Return : public static class TourJobDetail_High implements Serializable{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        public int packageID;
        public String reserveDate;
        public String deadline;
        public int price;
        public int quantity;
        public int status;
    };
```

12. ฟังก์ชันยกเลิกการจองเฉพาะ job

Function : Tour_Cancel_jobID(int Wedding_ContactID,int contactID,int jobID)

Return :boolean บอกว่ายกเลิกได้หรือไม่

13. ฟังก์ชันยกเลิกการจองทั้งหมด

Function :Tour_Cancel_Contact(int Wedding_ContactID,int contactID)

Return :boolean บอกว่ายกเลิกได้หรือไม่

14. ฟังก์ชันยืนยันการจอง

Function : Tour_ConfirmBooking(int TourID,int contactID)

Return : boolean บอกว่าสำเร็จหรือไม่

6.8.4 ฟังก์ชันของโฟโตสตูดิโอ

1. ฟังก์ชันยกเลิกการจองคอนแทค(Contact)

Function : Photo_CancelContact(int contactID,int weddingContactID)

Return : บอกว่า cancel ได้หรือไม่

2. ฟังก์ชันยกเลิกการจองการถ่ายภาพในสตูดิโอ

Function : Photo_CancelJob_InStudio(int contactID,int jobID,int weddingContactID)

Return : บอกว่า cancel ได้หรือไม่

3. ฟังก์ชันยกเลิกการจองการทำมัลติมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function : Photo_CancelJob_Multimedia(int contactID,int jobID,int weddingContactID)

Return : บอกว่า cancel ได้หรือไม่

4. ฟังก์ชันยกเลิกการจองการการถ่ายภาพนอกสถานที่

Function : Photo_CancelJob_OutStudio(int contactID,int jobID,int weddingContactID)

Return : บอกว่า cancel ได้หรือไม่

5. ฟังก์ชันสร้างคอนแทค

Function : Photo_CreateContactID(int photoID,String contactName,int weddingContactID,int custID)

Return : หมายเลขคอนแทคที่ได้

6. ฟังก์ชันแก้ไขการจองในสตูดิโอ

Function : Photo_EditReserve_InStudio(int weddingContactID,int contactID,int jobID,String reserveDate,int round)

Return : บอกว่าจองได้หรือไม่ เป็นboolean

7. ฟังก์ชันแก้ไขการจองการทำมัลติมีเดีย

Function : Photo_EditReserve_Multi(int weddingContactID,int contactID,int jobID,String reserveDate,String finishDate)

Return : บอกว่าจองได้หรือไม่ เป็นboolean

8. ฟังก์ชันแก้ไขการจองนอกสถานที่

Function : Photo_EditReserve_OutStudio(int weddingContactID,int contactID,int jobID,String reserveDate,int round)

Return : บอกว่าจองได้หรือไม่ เป็นboolean

9. ฟังก์ชันหาเวลาว่างที่สามารถจองการถ่ายภาพในสตูดิโอ

Function : Photo_FindTimeAvai_InStudio(int photoID,int typeID,int numDay)

Return : คลาส TimeAvailable บอก วัน เวลา

```
public static class TimeAvailable implements Serializable
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    public String date;
    public int round;
    public String time;
};
```

10. ฟังก์ชันหาเวลาว่างที่สามารถจองการทำมัลติมีเดีย

Function : Photo_FindTimeAvai_Multi(int photoID,int numDay)

Return : คลาส TimeAvailable บอก วัน เวลา

```
public static class TimeAvailable implements Serializable
{
    public String date;
    public int round;
    public String time;
};
```

11. ฟังก์ชันหาเวลาว่างที่สามารถจองนอกสถานที่

Function : Photo_FindTimeAvai_OutStudio(int photoID,int typeID,int numDay)

Return : คลาส TimeAvailable บอก วัน เวลา

```
public static class TimeAvailable implements Serializable
  {
    public String date;
    public int round;
    public String time;
  };
```

12. ฟังก์ชันหาคอนแทคไอดีของลูกค้า

Funciotn : Photo_GetContactID(int weddingContactID)

Return : บอกรายละเอียดของ Contact

```
public static class ContactID implements Serializable
  {
    public int contactID;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    public int photoID;
    public String photoName;
};
```

13. ฟังก์ชันหาราคารวมการจองทั้งหมดของลูกค้า

Function : Photo_GetTotalPrice(int weddingContactID)

Return : ราคาของการจองกับบริษัทถ่ายภาพ

14. ฟังก์ชันตรวจสอบว่าจะสามารถจองการถ่ายภาพในสตูดิโอได้หรือไม่

Function : Photo_IsTimeAvailable_InStudio(int photoID,int typeID,String checkDate,int round)

Return : บอกว่าเวลาที่ตรวจสอบนั้นว่าหือไม่โดยใช้เป็น boolean

15. ฟังก์ชันตรวจสอบว่าจะสามารถจองการทำมัลติมีเดียได้หรือไม่

Function : Photo_IsTimeAvailable_Multi(int photoID,String checkDate)

Return : บอกว่าเวลาที่ตรวจสอบนั้นว่าหือไม่โดยใช้เป็น boolean

16. ฟังก์ชันตรวจสอบว่าจะสามารถจองการถ่ายภาพนอกสตูดิโอได้หรือไม่

Function : Photo_IsTimeAvailable_OutStudio(int photoID,int typeID,String checkDate,int round)

Return : บอกว่าเวลาที่ตรวจสอบนั้นว่าหือไม่โดยใช้เป็น boolean

17. ฟังก์ชันบอกที่อยู่ของบริษัท

Function : Photo_PlaceDetail(int photoID)

Return : ที่อยู่ของบริษัท

18. ฟังก์ชันจองการถ่ายภาพในสตูดิโอ

Function : Photo_ReserveInStudio(int photoID,int weddingContactID,String typeID,String subtypeID,String reserveDate,int round)

Return : ราคาการจองการถ่ายภาพในสตูดิโอของการจองครั้งนั้น

19. ฟังก์ชันจองการทำมัลติมีเดีย

Function : Photo_ReserveMultimedia(int photoID,int weddingContactID,String subtypeID,String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

reserveDate,String showDate)

Return : ราคาการจองการมัลติมีเดียของการจองครั้งนั้น

20. ฟังก์ชันจองการถ่ายภาพนอกสตูดิโอ

Function : Photo_ReserveOutStudio(int photoID,int weddingContactID,String typeID,String subtypeID,String reserveDate,int round)

Return : ราคาการจองการถ่ายภาพนอกสตูดิโอของการจองครั้งนั้น

21. ฟังก์ชันใช้ในการค้นหาบริการของบริษัทถ่ายภาพ

Function : Photo_SearchService(int photoID,String typeID,int min_Price,int maxPrice,boolean cheap,int district,int region)

Return : เป็นอาเรย์ของคลาสบอกรายละเอียดของบริการที่ตรงกับเงื่อนไข

```
public static class ServiceDetail implements Serializable
  {
    public int photoID;
    public double price;
    public String subTypeDetail;
    public String subTypeID;
    public String subTypeName;
    public String typeID;
    public String typeName;
    public String photoAddress;
  };
```

22. ฟังก์ชันบอกรายละเอียดการจองการถ่ายภาพในสตูดิโอ

Function : Photo_ShowJobInfo_InStudio(int photoID,int contactID,int jobID)

Return : คลาสบอกรายละเอียดการจองการถ่ายภาพในสตูดิโอ

```
public static class JobInStudioDetail implements Serializable
  {
    public int contactID;
    public int jobID;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public double price;
public String reserveDate;
public String round;
public String subTypeDetail;
public String subTypeName;
public String subTypeID;
public String typeID;
};
```

23. ฟังก์ชันบอกรายละเอียดการจองการทำมัลติมีเดีย

Fuction : Photo_ShowJobInfo_Multi(int photoID,int contactID,int jobID)

Return : คลาสบอกรายละเอียดการจองการทำมัลติมีเดีย

```
public static class JobMultiDetail implements Serializable
{
public int contactID;
public int jobID;
public double price;
public String finishDate;
public String reserveDate;
public String subTypeDetail;
public String subTypeName;
public String subTypeID;
public String typeID;
};
```

24. ฟังก์ชันบอกรายละเอียดการจองการถ่ายภาพนอกสตูดิโอ

Function : Photo_ShowJobInfo_OutStudio(int photoID,int contactID,int jobID)

Return : คลาสบอกรายละเอียดการจองการถ่ายภาพนอกสตูดิโอ

```
public static class JobInStudioDetail implements Serializable
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    public int contactID;
    public int jobID;
    public double price;
    public String reserveDate;
    public String round;
    public String subTypeDetail;
    public String subTypeName;
    public String subTypeID;
    public String typeID;
};
```

25. ฟังก์ชันแสดงรายการจองทั้งหมดของคอนแทคของลูกค้า

Function : Photo_ShowOrderByContact(int weddingContactID,int contactID)

Return : อาเรย์ของคลาสที่บอกรายละเอียดของงานที่ถูกจองโดยคอนแทคนั้น

```
public static class ContactInfo implements Serializable
    {
    public int jobID;
    public String receiveDate;
    public String subTypeName;
    public String typeName;
    public int photoID;
    public String photoName;
    public String typeID;
    };
```

26. ฟังก์ชันไว้ดูว่าลูกค้ามีคอนแทคอะไรบ้าง

Function : Photo_ShowOrderByCustID(int photoID,int custID)

Return : public static class CustomerIDInfo implements Serializable{

```
public int contactID;
 public String contactName;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        public String deadline;
        public double totalPrice;
    };
```

6.8.5 ฟังก์ชันของร้านของชำร่วย

1. ฟังก์ชันสมัครสมาชิก

Function : GiftRegister(java.lang.String username, java.lang.String password, java.lang.String firstname, java.lang.String lastname, java.lang.String sex, int age, java.lang.String tel, java.lang.String address, java.lang.String email,int GiftNumber,String GiftName,String GiftAddress)

Return : Customer ID ถ้าสมัครไม่ได้จะ <1

2. ฟังก์ชันค้นหาของชำร่วย

Function : GiftSearch(java.lang.String type_id, java.lang.String style, double min_price, double max_price,boolean cheap,int GiftNumber)

Return : public static class GiftInfo implements Serializable{

```
        public String type_id;
        public String type_name;
        public String subtype_id;
        public String subtype_name;
        public String subtype_detail;
        public String pic_url;
        public double price;
        public String GiftName;
        public String GiftAddress;
        public int GiftNumber;
    }
```

3. ฟังก์ชันแสดงคอนแทคของลูกค้า

Function : GiftListContact(int Customer_id,int Wedding_contact_id)

Return : Contact ID (ถ้าไม่มีจะเป็น –1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ฟังก์ชันสร้างคอนแทคใหม่

Function : GiftCreateContact(int customer_id, java.lang.String contact_name,int GiftNumber,int wedding_contact_id)

Return : หมายเลขคอนแทค ถ้าไม่สามารถสร้างได้จะมีค่าเป็น -1

5. ฟังก์ชันแสดงรายละเอียดของคอนแทค

Function : GiftListContact1(int Customer_id,int Wedding_contact_id)

Return : public static class Contactdetail implements Serializable{

```
public int gift_contact_id;
public String GIFT_NAME;
public int GIFT_ID;
}
```

6. ฟังก์ชันสั่งจองของชำร่วย

Function : GiftOrder(int customer_id,java.lang.String type_id, java.lang.String subtype_id, java.lang.String font_style, java.lang.String font_color, java.lang.String label, java.lang.String receivedateS, java.lang.String address, int quantity,int GiftNumber,int Wedding_Contact_ID,int Gift_contact_id)

Return : public static class ReturnReserve implements Serializable{

```
public int job_id;
public double price;
public String deadline;
public String GiftName;
public int GiftNumber;
}
```

7. ฟังก์ชันสั่งแก้ไขการจองจองของชำร่วย

Function : GiftEditOrder(int customer_id, int job_id, java.lang.String receivetimeS, java.lang.String address, java.lang.String label,int GiftNumber,int Wedding_Contact_ID,int Gift_contact_id)

Return : public static class ReturnReserve implements Serializable{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        public int job_id;
        public double price;
        public String deadline;
        public String GiftName;
        public int GiftNumber;
    }
```

8. ฟังก์ชันแสดง job ทั้งหมดของคอนแทค

Function : GiftListJob(int Custeomer_id,int Wedding_contact_id,int Gift_contact_id)

Return : public static class job implements Serializable{

```
        public int job_id;
        public String subtype_name;
        public String receive_date;
        public String GiftName;
        public int GiftNumber;
    }
```

9. ฟังก์ชันแสดงรายละเอียดทั้งหมดของ job

Function : GiftShowDetailJob(int wedding_contact_id,int gift_contact_id, int gift_job_id)

Return : public static class JobDetail implements Serializable{

```
        public int contact_id;
        public int job_id;
        public String  type_id;
        public String subtype_id;
        public String  subtype_name;
        public String subtype_detail;
        public String label;;
        public String receive_date;
        public String address;
        public int quantity;
        public double job_total_price ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

10. ฟังก์ชันยกเลิก job ของลูกค้า

Function : GiftCancelJob(int customer_id, int Gift_contact_id, int job_id,int Wedding_Contact_ID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

11. ฟังก์ชันยกเลิกคอนแทคของลูกค้า

Function : GiftCancelContact(int customer_id,int gift_contact_id,int Wedding_ContactID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

6.8.6 ฟังก์ชันของโรงแรม

1. ฟังก์ชันสร้างคอนแทค

Function : HotelCreateContact(java.lang.String contact_name,int HotelNumber,int customer_id,int wedding_contact_id)

Return : Contact ID ถ้าสร้างไม่ได้จะเป็น -1

2. ฟังก์ชันสมัครสมาชิก

Function : HotelRegister(java.lang.String fname, java.lang.String lname, java.lang.String sex, int age, java.lang.String tel, java.lang.String addr, java.lang.String email, java.lang.String username, java.lang.String password,int HotelNumber)

Return : boolean บอกว่าสามารถสมัครได้หรือไม่

3. ฟังก์ชันจองห้องจัดเลี้ยง

Function : HotelReserveSeminar(int customer_id, int hotel_contact_id, int seminar_room_id, java.lang.String dec_style, java.lang.String Sreserve_date,int t6to11ams,int t12to5pms,int t6to11pms,int men_per_rooms, java.lang.String table_layout,int HotelID,int Wedding_Contact_ID)

Return : public static class ReturnReserve implements Serializable{

public int job_id;

public double price;

public String deadline;

public String HotelName;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public String str_error; }

4. ฟังก์ชันจองห้อง

Function : HotelReserveRoom(int customer_id, int hotel_contact_id, int amount_room, java.lang.String room_type, java.lang.String Scheckin_date, java.lang.String Scheckout_date, int amount_extrabed, int smoke,int HotelID,int Wedding_Contact_ID)

Return : public static class ReturnReserve implements Serializable{

public int job_id;

public double price;

public String deadline;

public String HotelName;

public String str_error;

}

5. ฟังก์ชันจองอาหาร

Function : HotelReserveFood(int customer_id, int hotel_contact_id, int seminar_room_job_id, int food_package_id, java.lang.String Sfood_serve_time, short serve_amount,int HotelID,int Wedding_Contact_ID)

Return : public static class ReturnReserve implements Serializable{

public int job_id;

public double price;

public String deadline;

public String HotelName;

public String str_error;

}

6. ฟังก์ชันจองเค้ก

Function : HotelReserveCake(int customer_id, java.lang.String cake_flavour, int cake_id, int seminar_room_job_id, int hotel_contact_id, java.lang.String Scake_serve_time, short cake_amount,int HotelID,int Wedding_Contact_ID)

Return : public static class ReturnReserve implements Serializable{

public int job_id;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public double price;

public String deadline;

public String HotelName;

public String str_error; }

7. ฟังก์ชันจองดอกไม้

Function : ReturnReserve HotelReserveFlower(int customer_id, int flower_package_id, int seminar_room_id, int seminar_room_job_id, int hotel_contact_id, int flower_time_range,int HotelID,int Wedding_Contact_ID)

Return : public static class ReturnReserve implements Serializable{

public int job_id;

public double price;

public String deadline;

public String HotelName;

public String str_error;

}

8. ฟังก์ชันยกเลิกการจองห้องจัดเลี้ยง

Function : HotelCancelSeminar(int customer_id, int hotel_contact_id, int seminar_room_job_id,int Wedding_ContactID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

9. ฟังก์ชันยกเลิกการจองห้อง

Function : HotelCancelRoom(int customer_id, int hotel_contact_id, int room_job_id,int Wedding_ContactID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

10. ฟังก์ชันยกเลิกการจองอาหาร

Function : HotelCancelFood(int customer_id, int hotel_contact_id, int food_job_id,int Wedding_ContactID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ฟังก์ชันยกเลิกการจองเค้ก

Function : HotelCancelCake(int customer_id, int hotel_contact_id, int cake_job_id,int Wedding_ContactID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

12. ฟังก์ชันยกเลิกการจองดอกไม้

Function : HotelCancelFlower(int customer_id, int hotel_contact_id, int flower_job_id,int Wedding_ContactID)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

13. ฟังก์ชันยกเลิกคอนแทค

Function : HotelCancelContact(int Wedding_ContactID, int hotel_contact_id)

Return : boolean บอกว่าสามารถยกเลิกได้หรือไม่

14. ฟังก์ชันแสดงห้องจัดเลี้ยงที่ได้จองไว้ทั้งหมด

Function : HotelShowReserveSeminar(int seminar_room_job_id,int HotelNumber)

Return : public static class SeminarRoomReservationList implements Serializable{

```
public int seminar_room_job_id;
public int seminar_room_id;
public String seminar_room_name;
public String dec_style;
public String place_style;
public String table_layout;
public int men_per_room;
public String reserve_date;
public int time_6_to_11am;
public int time_12_to_5pm;
public int time_6_to_11pm;
public double job_total_price;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. ฟังก์ชันแสดงห้องที่ได้จองไว้ทั้งหมด

Function : HotelShowReserveRoom(int room_job_id,int HotelNumber)

Return : public static class RoomReservationList implements Serializable{

```
public int room_job_id;
public int room_id;
public String room_type;
public int room_size;
public int maximum_men_per_room;
public String checkin_date;
public String checkout_date;
public int extrabed;
public double job_total_price;
}
```

16. ฟังก์ชันแสดงรายละเอียดการจองอาหารของลูกค้า

Function : HotelShowReserveFood(int food_job_id,int HotelNumber)

Return : รายละเอียดการจองอาหารเป็นคลาส

```
public static class FoodReservationList implements Serializable
{
    public int food_job_id;
    public int seminar_room_job_id;
    public int food_package_id;
    public String serve_style;
    public String food_style;
    public String food_serve_time;
    public int serve_amount;
    public double job_total_price;
}
```

17. ฟังก์ชันแสดงรายละเอียดการจองเค้กของลูกค้า

Function : HotelShowReserveCake(int cake_job_id,int HotelNumber)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Return : รายละเอียดการจองเค้กเป็นคลาส

```
public static class CakeReservationList implements Serializable
    {   public int cake_job_id;
        public int seminar_room_job_id;
        public String cake_flavour;
        public String cake_name;
        public String cake_style;
        public int cake_size;
        public int cake_layer;
        public String cake_serve_time;
        public int cake_amount;
        public double job_total_price;
    }
```

18. ฟังก์ชันแสดงรายละเอียดการจองดอกไม้ของลูกค้า

Function : HotelShowReserveFlower(int flower_job_id,int HotelNumber)

Return : รายละเอียดการจองดอกไม้เป็นคลาส

```
public static class FlowerReservationList implements Serializable
    {   public int flower_job_id;
        public int flower_package_id;
        public String flower_package_name;
        public int seminar_room_id;
        public int seminar_room_job_id;
        public int flower_time_range;
        public double job_total_price;
    }
```

19. ฟังก์ชันแสดงงานทั้งหมดของคอนแท็คของลูกค้า

Function : HotelShowJobID(int customer_id, int hotel_contact_id,int HotelNumber)

Return : งานทั้งหมดของคอนแท็คของลูกค้าเป็นคลาส

```
public static class ShowAllJobid implements Serializable
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{       public int[] room_job_id;
        public int[] seminar_room_job_id;
        public int[] cake_job_id;
        public int[] food_job_id;
        public int[] flower_job_id;
}
```

20. ฟังก์ชันแสดงรายละเอียดของคอนแท็คของลูกค้า

Function : Hotel_Contact_List(int Customer_id,int Wedding_contact_id)

Return : รายละเอียดของคอนแท็คของลูกค้า

```
public static class hotelcontact implements Serializable
  {   public int hotel_id;
      public String hotel_name;
      public int hotelcontact;
  }
```

21. ฟังก์ชันแก้ไขการจองเค้ก

Function : HotelEditCake(int customer_id, int cake_job_id, java.lang.String Scake_serve_time,int HotelNumber)

Return : บอกว่าแก้ไขได้หรือไม่เป็นรูปแบบ boolean

22. ฟังก์ชันแก้ไขการจองอาหาร

Function : HotelEditFood(int customer_id, int food_job_id, java.lang.String Sfood_serve_time,int HotelNumber)

Return : บอกว่าแก้ไขได้หรือไม่เป็นรูปแบบ boolean

23. ฟังก์ชันแสดงรายชื่ออาหารในแพ็คเกจ

Function : HotelShowFoodPackage(int HotelID,int food_package_id)

Return : รายชื่อของอาหารที่อยู่ในแพ็คเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

24. ฟังก์ชันแสดงรายการงานทั้งหมดที่เกี่ยวกับห้องจัดเลี้ยงนั้นๆ

Function : HotelShowJobInSeminar(int customer_id, int hotel_contact_id, int seminar_room_job_id,int HotelID,int Wedding_ContactID)

Return : รายการงานทั้งหมดที่เกี่ยวกับห้องจัดเลี้ยงนั้นๆ

```
public static class ShowAllJobid implements Serializable
    {       public int[] room_job_id;
            public int[] seminar_room_job_id;
            public int[] cake_job_id;
            public int[] food_job_id;
            public int[] flower_job_id;
    }
```

25. ฟังก์ชันค้นหาห้องจัดงาน

Function : HotelSearchSeminar(java.lang.String place_style, java.lang.String dec_style, int seminar_room_size_min, int seminar_room_size_max, double minprice, double maxprice, java.lang.String Sreserve_date, short t6to11_am, short t12to5_pm, short t61011_pm,boolean cheap,int HotelNumber,int district,int region)

Return : รายละเอียดห้องจัดเลี้ยงที่ตรงกับเงื่อนไขทั้งหมดอยู่ในรูปอเรย์ของคลาส

```
public static class SearchSeminarRoomResult implements Serializable
    {       public int seminar_room_id;
            public String seminar_room_name;
            public String dec_style;
            public String place_style;
            public int seminar_room_size;
            public double seminar_room_price;
            public int max_men_per_room_table;
            public int max_men_per_room_buffet;
        public String HotelName;
        public String HotelAddress;
        public  int HoteIID;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

26. ฟังก์ชันค้นหาห้องพัก

Function : HotelSearchRoom(int amount_room, java.lang.String room_type, java.lang.String Scheckin_date, java.lang.String Scheckout_date, int smoke, double minprice, double maxprice,boolean cheap,int HotelNumber,int region,int district)

Return : รายละเอียดห้องพักที่ตรงกับเงื่อนไขทั้งหมดอยู่ในรูปอเรย์ของคลาส

```
public static class SearchRoomResult implements Serializable
  {   public String room_type;
      public int room_size;
      public double room_price;
      public int maximum_men_per_room;
      public double extrabed_price;
      public int smoke;
      public String HotelName;
      public String HotelAddress;
      public  int HotelID;
  }
```

27. ฟังก์ชันค้นหาแพ็คแกจอาหาร

Function : HotelSearchFood(double minprice, double maxprice, java.lang.String food_style, java.lang.String serve_type, java.lang.String food_name,boolean cheap,int HotelNumber,int food_per)

Return : รายการแพ็กเกจอาหารที่ตรงกับเงื่อนไขทั้งหมดอยู่ในรูปอเรย์ของคลาส

```
public static class SearchFoodResult implements Serializable
    {       public int food_package_id;
            public String food_style;
            public double food_price_per_person;
            public double food_price_per_table;
      public String HotelName;
      public  int HotelID;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

28. ฟังก์ชันค้นหาเค้ก

Function : HotelSearchCake(java.lang.String cake_style, double minprice, double maxprice, int layer, java.lang.String flavour, int cake_size,boolean cheap,int HotelNumber,int region,int district)

Return : รายละเอียดเค้กที่ตรงกับเงื่อนไขทั้งหมดอยู่ในรูปอเรย์ของคลาส

```
public static class SearchCakeResult implements Serializable
        {       public String cakename;
                public String cakestyle;
                public String cake_size;
                public String layer;
                public String flavour;
                public double cake_price;
         public int cake_id;
         public String HotelName;
         public  int HotelID;
         }
```

29. ฟังก์ชันค้นหาแพ็คเกจดอกไม้

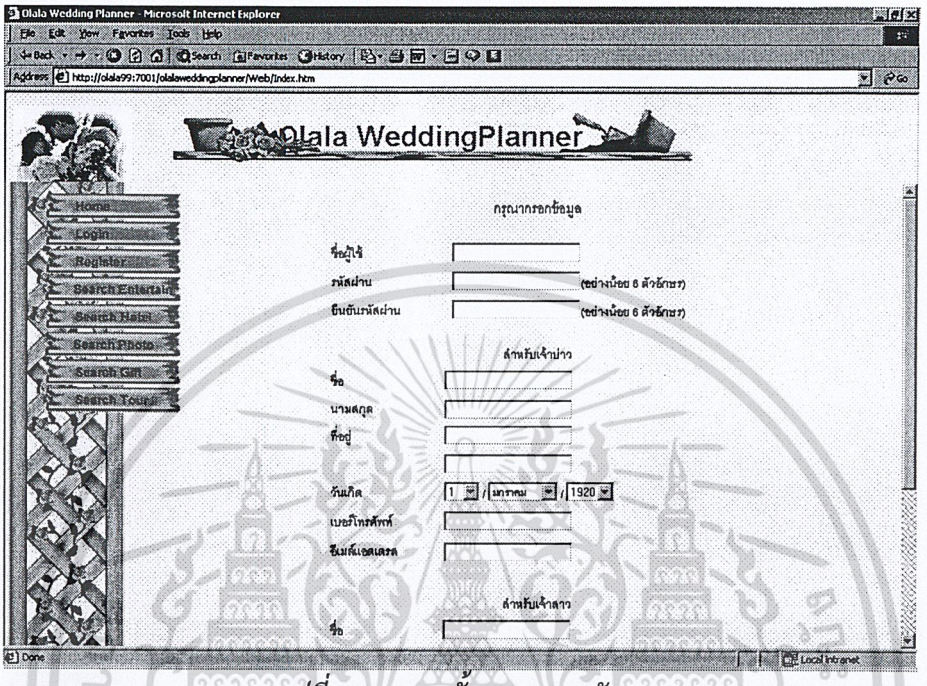
Function : HotelSearchFlower(java.lang.String seminar_room_name, double minprice, double maxprice,boolean cheap,int HotelNumber,int region,int district)

Return : รายละเอียดของแพ็กเจกดอกไม้ที่ตรงกับเงื่อนไขทั้งหมดอยู่ในรูปอเรย์ของคลาส

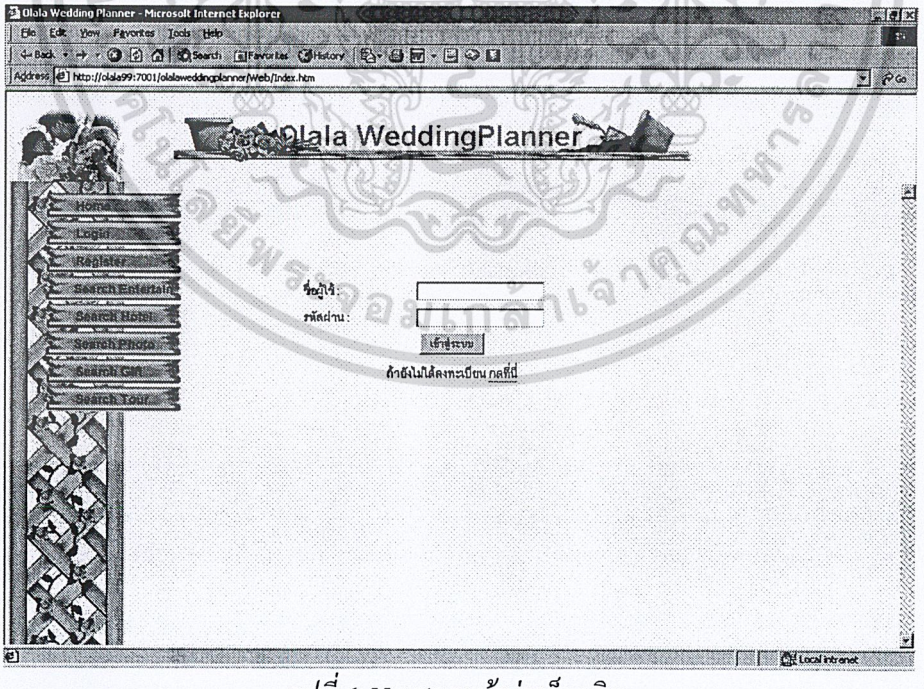
```
public static class SearchFlowerResult implements Serializable
        {       public int flower_package_id;
                public String flower_package_name;
                public double flower_price;
         public String HotelName;
         public  int HotelID;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.9 คู่มือการใช้งานแอพพลิเคชั่น

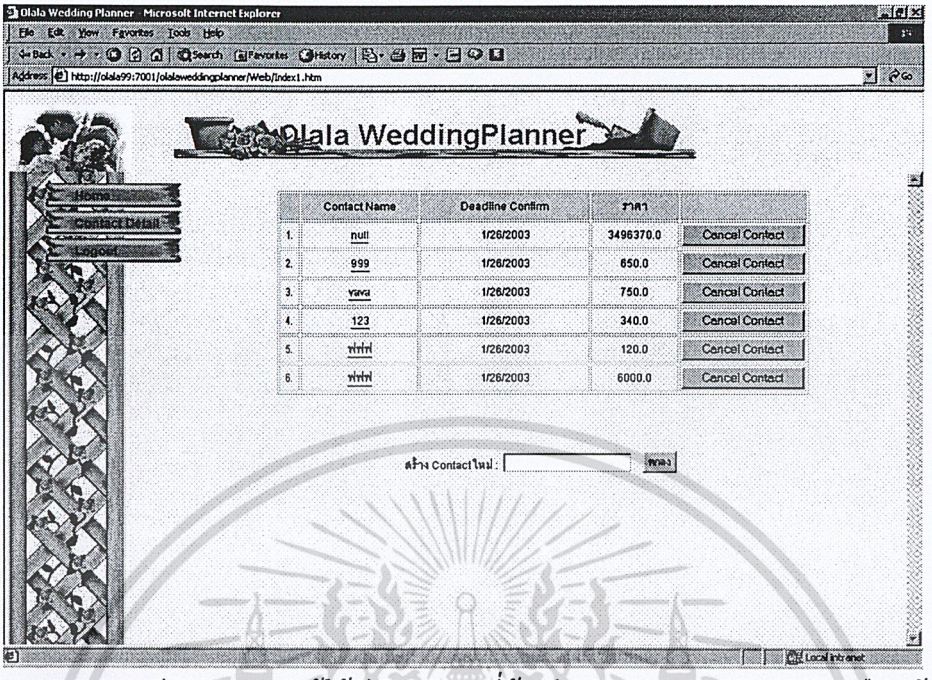


รูปที่ 6-54 แสดงขั้นตอนการสมัคร

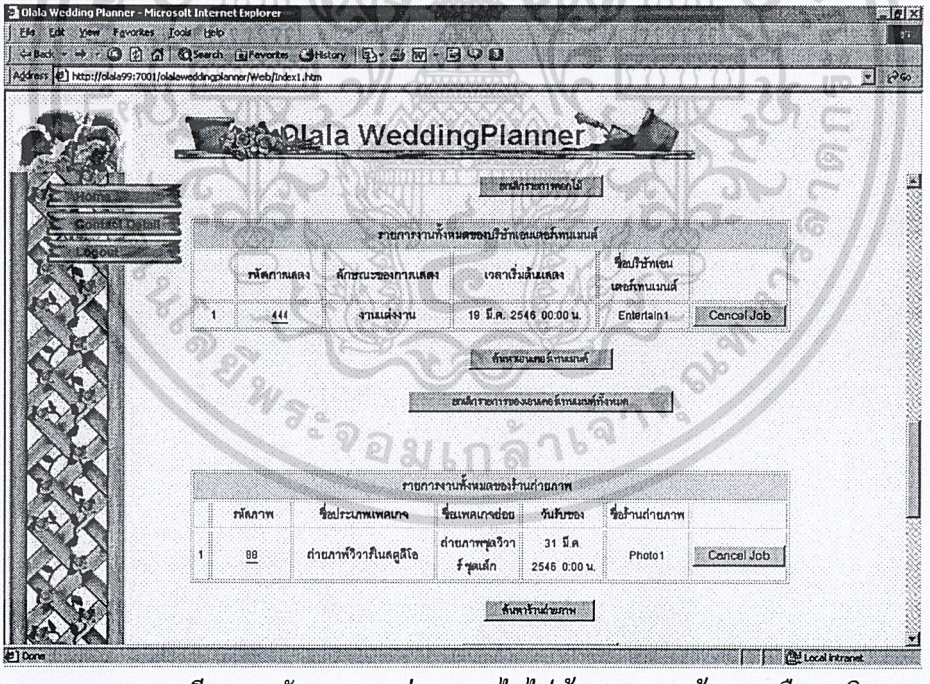


รูปที่ 6-55 แสดงหน้าต่างล็อกอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

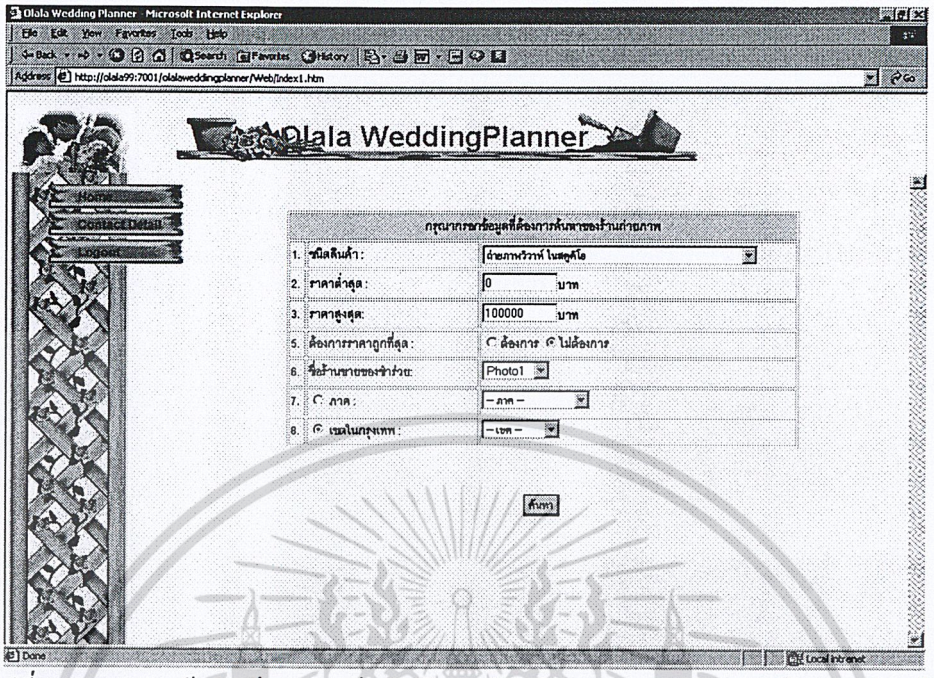


รูปที่ 6-56 แสดงรายละเอียดสัญญาของผู้ใช้ มีกำหนดวันที่ต้องจ่ายตัง,ราคา สามารถยกเลิก หรือ สร้างใหม่ได้

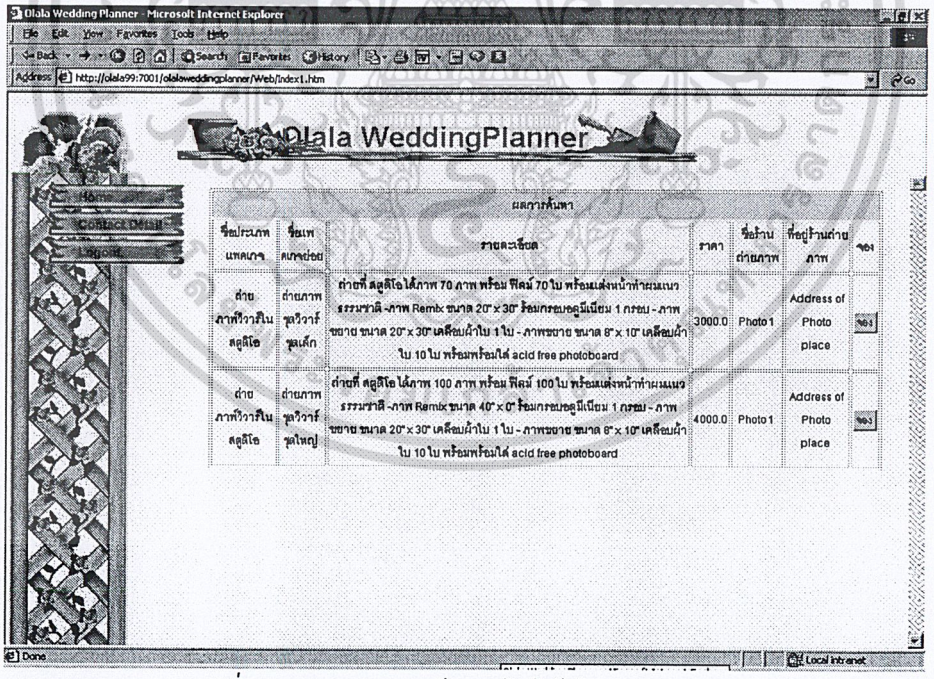


รูปที่ 6-57 แสดงรายละเอียดของสัญญา บอกว่าจองอะไรไปบ้าง สามารถค้นหา หรือยกเลิกรายการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

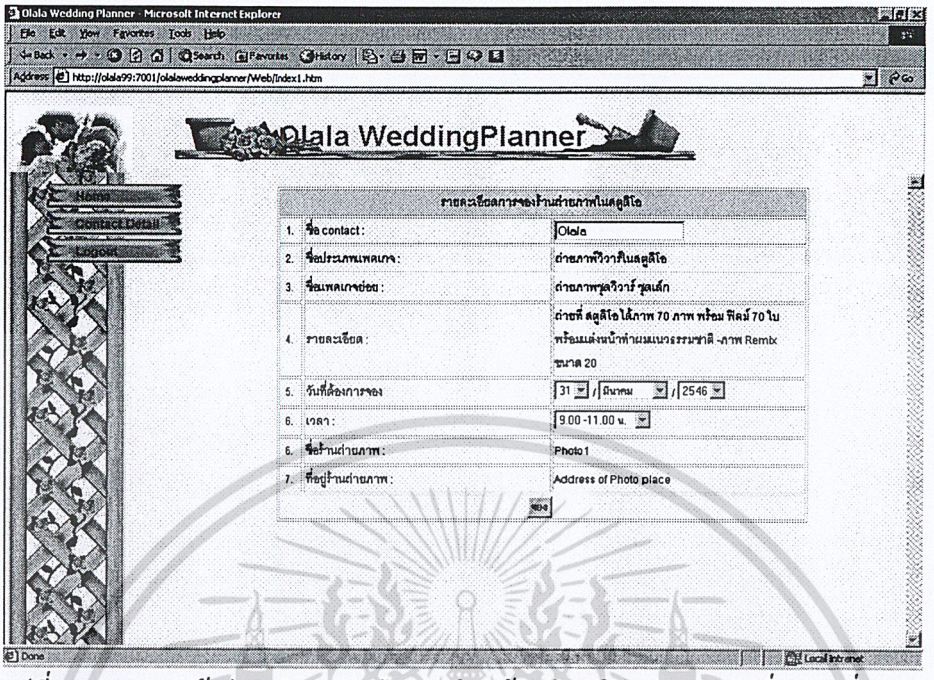


รูปที่ 6-58 แสดงหน้าต่างค้นหาของร้านโฟโตสตูดิโอ โดยสามารถเลือกความต้องการต่างๆ ได้

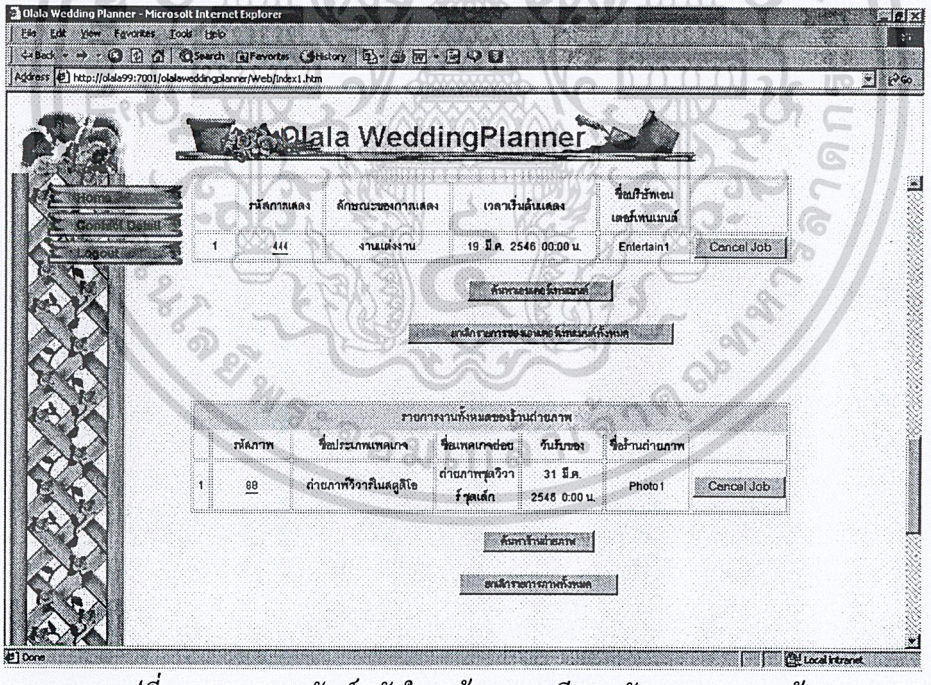


รูปที่ 6-59 แสดงผลการค้นหา(ไม่ได้เลือกแบบถูกสุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

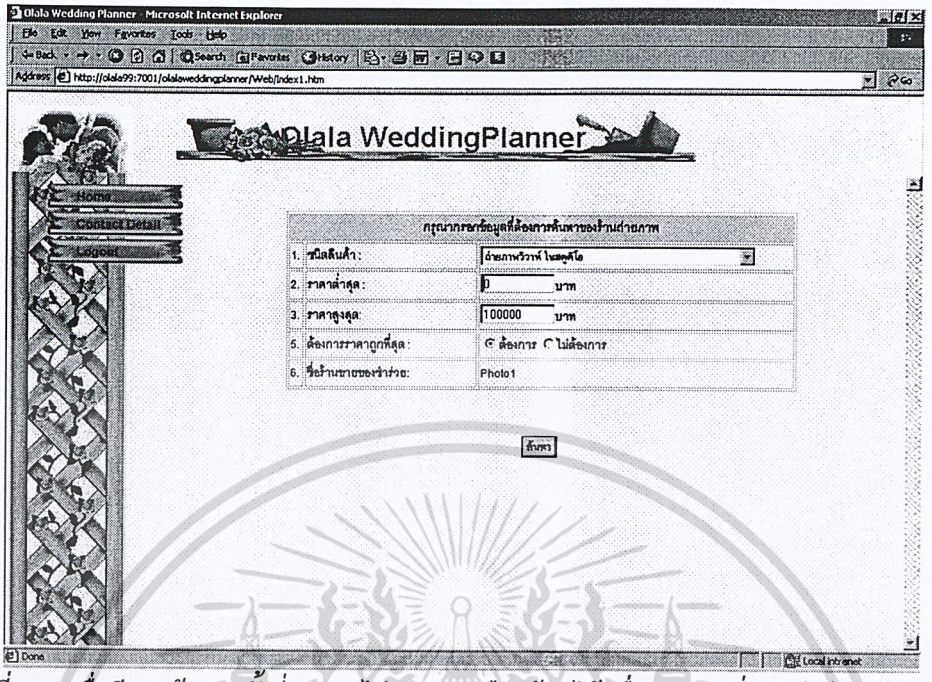


รูปที่ 6-60 แสดงหน้าต่างหลังจากกดปุ่มจอง โดยต้องมีการใส่รายละเอียดเพิ่มก่อนที่จะจอง

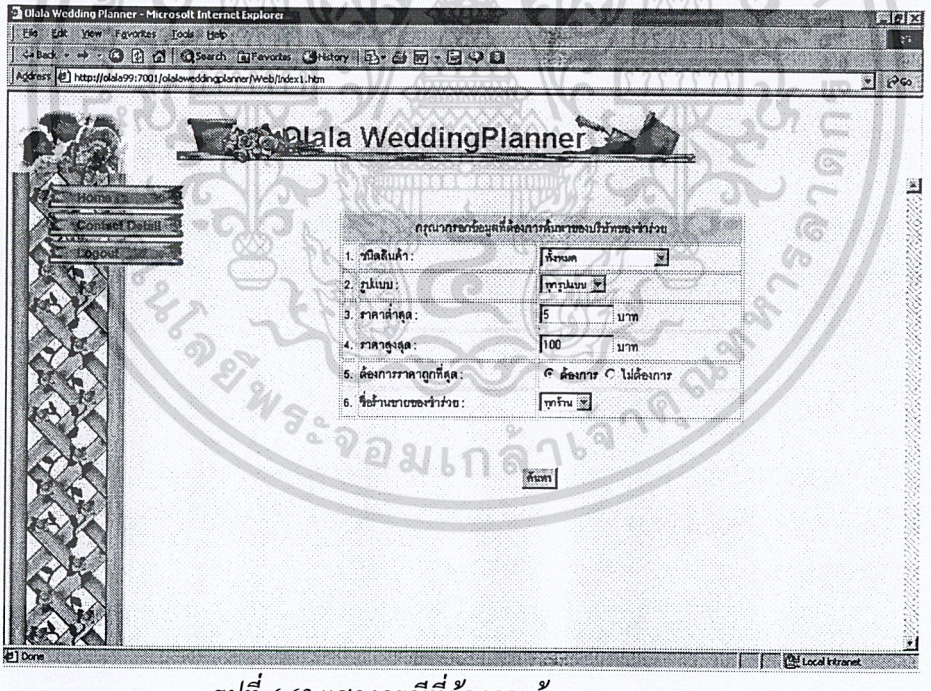


รูปที่ 6-61 แสดงผลลัพธ์หลังในหน้ารายละเอียดหลังจากการจองแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

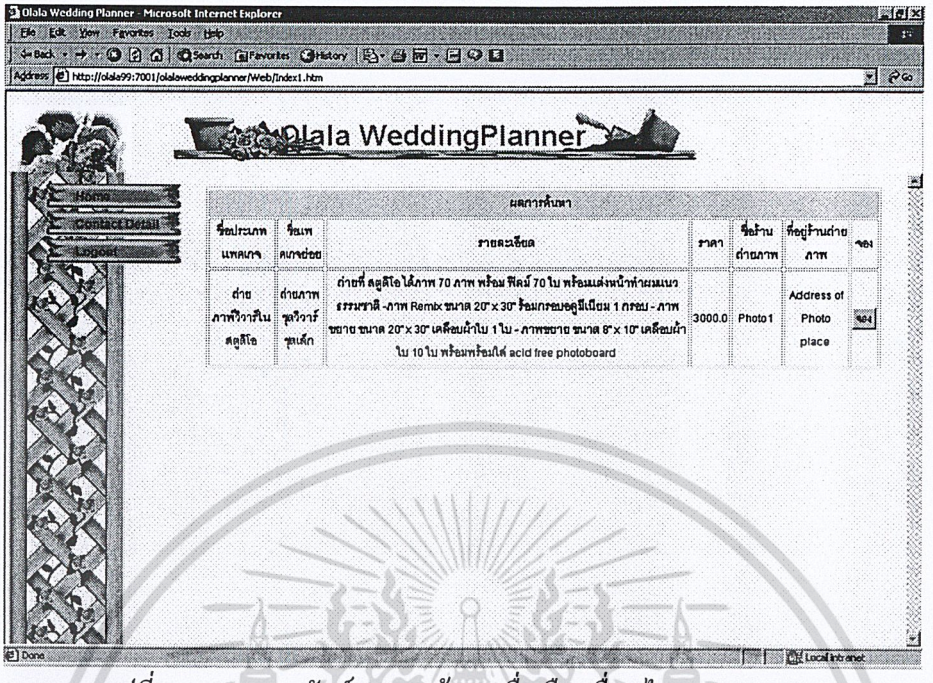


รูปที่ 6-62 เมื่อมีการค้นหาครั้งที่สองจะไม่สามารถเลือกร้านได้ เนื่องจากจะสั่งของได้จากร้านเดียว

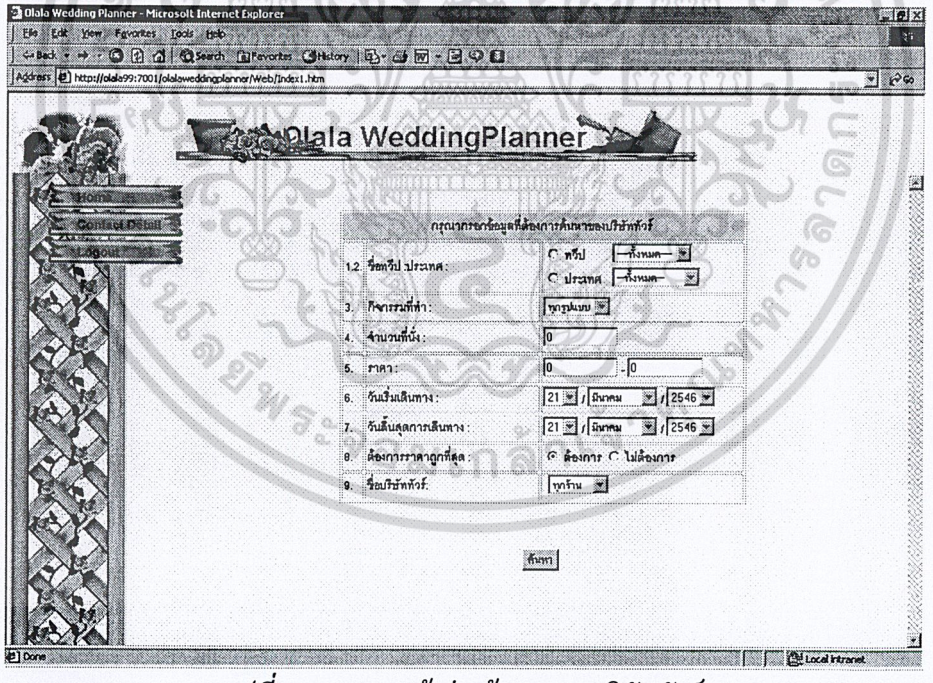


รูปที่ 6-63 แสดงกรณีที่ต้องการค้นหาแบบถูกสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

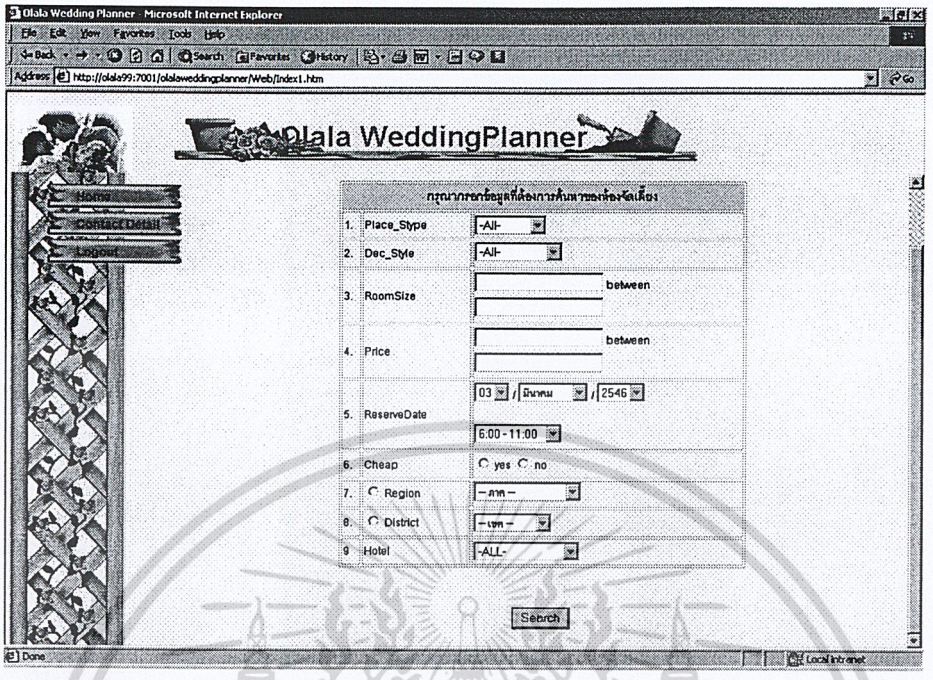


รูปที่ 6-64 แสดงผลลัพธ์จากการค้นหาเมื่อเลือกเงื่อนไข "แบบถูกสุด"

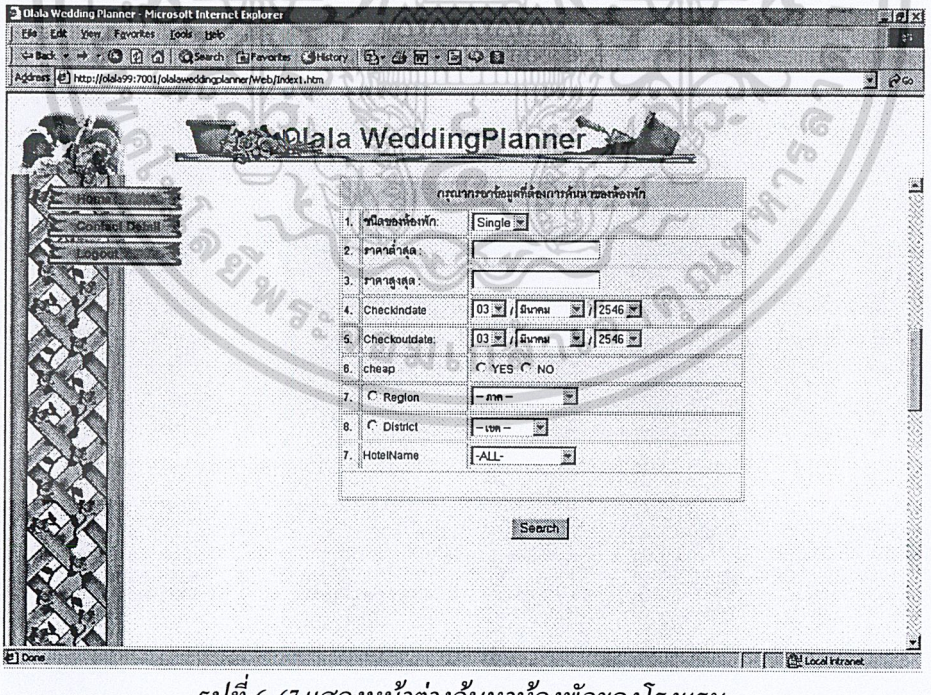


รูปที่ 6-65 แสดงหน้าต่างค้นหาของบริษัททัวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

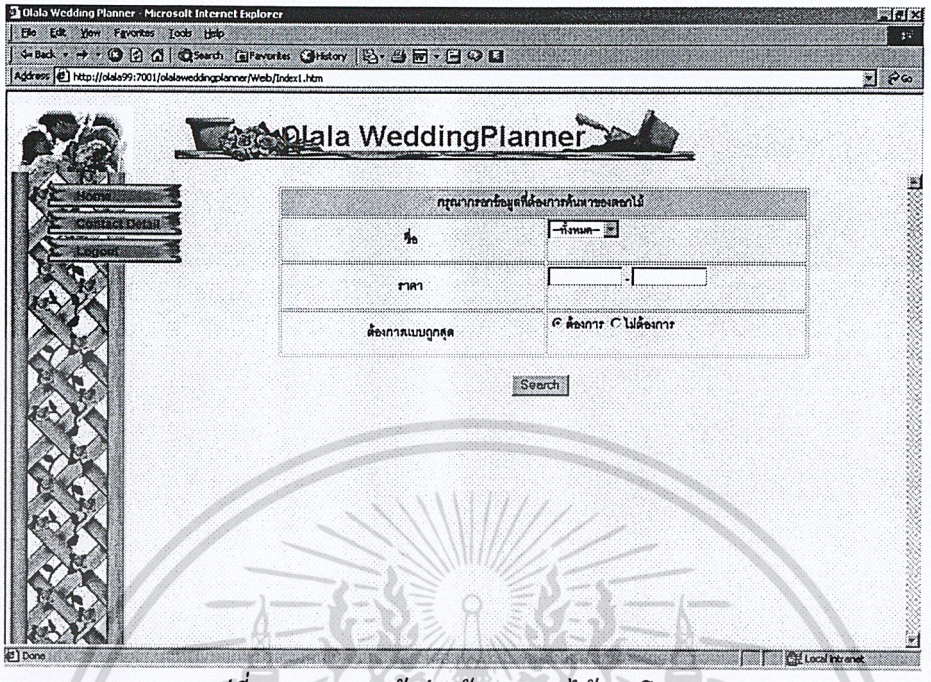


รูปที่ 6-66 แสดงหน้าต่างค้นหาห้องจัดเลี้ยงของโรงแรม สามารถค้นหาแบบใกล้สุด(เฉพาะในกรุงเทพ) หรือขอบเขต(ภาค) นอกเหนือจากแบบถูกสุดได้

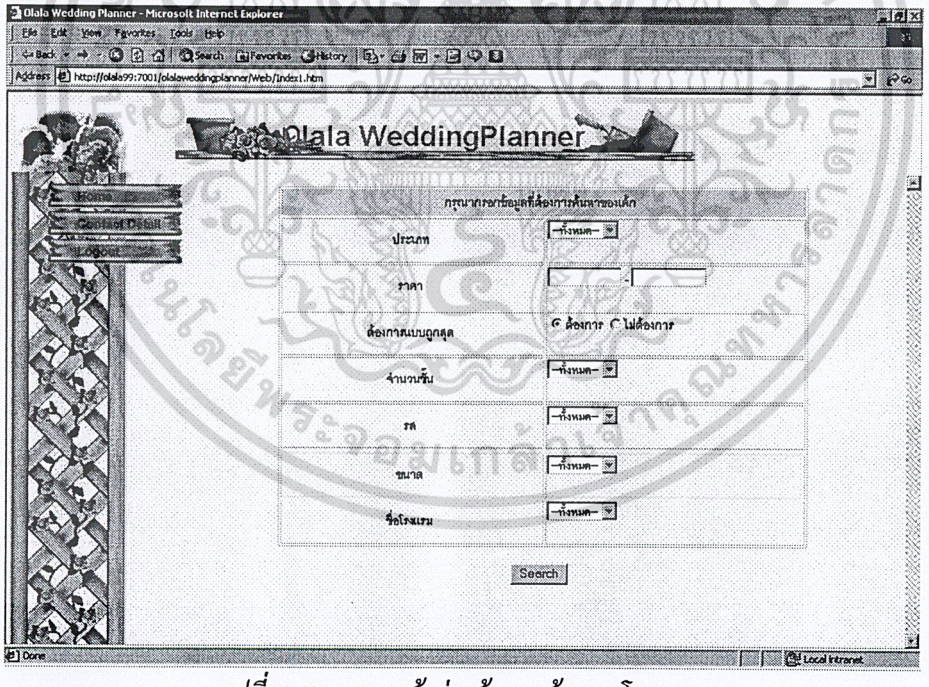


รูปที่ 6-67 แสดงหน้าต่างค้นหาห้องพักของโรงแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

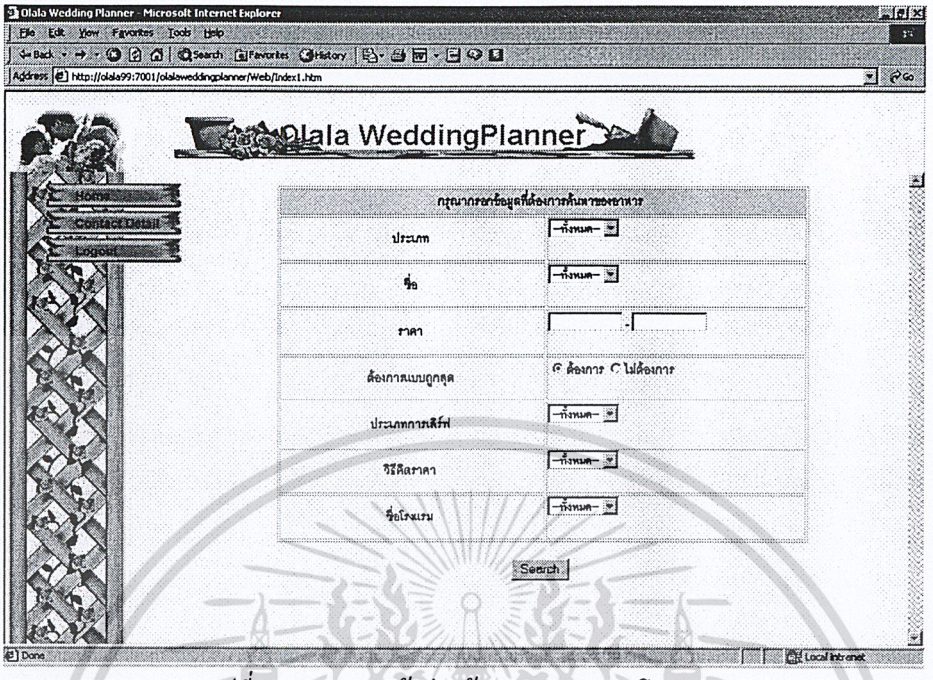


รูปที่ 6-68 แสดงหน้าต่างค้นหาดอกไม้ของโรงแรม



รูปที่ 6-69 แสดงหน้าต่างค้นหาเค้กของโรงแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-70 แสดงหน้าต่างค้นหาอาหารของโรงแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทวิจารณ์ และสรุป

7.1 สรุปผลการดำเนินงาน

ในโครงงานนี้ได้แบ่งการดำเนินงานออกเป็น 5 ส่วนหลัก คือ

1. ศึกษาทฤษฎีพื้นฐานของ คอมโพเน็นต์ , จาวาเว็บเซอร์วิส และเทคโนโลยีของจาวา
2. ออกแบบระบบงาน เพื่อทำการทดสอบทฤษฎีต่างๆ ที่ได้ศึกษามา โดยตัวอย่างระบบงานที่สร้างขึ้นมา คือ ระบบการจองแพ็กเกจทัวร์ , ระบบการจองเอ็นเตอร์เทนเมนต์ และระบบ OLALA Wedding Planner ที่เป็นตัวกลางในการติดต่อข้ามเทคโนโลยี
3. ทำการอิมพลีเมนต์ระบบงานที่ได้ออกแบบไว้ โดยใช้ภาษา จาวา , JSP พัฒนาตามมาตรฐานของเว็บเซอร์วิส และอิงตามเทคโนโลยีของจาวา
4. ทำการอิมพลีเมนต์ระบบงานในส่วนที่เรียกใช้ระบบโรงแรม , ระบบสตูดิโอถ่ายภาพ และระบบร้านขายของชำร่วย ที่พัฒนาจากเทคโนโลยีเว็บเซอร์วิสของ .NET ผ่านโพรโตคอลมาตรฐานที่เรียกว่า SOAP
5. เพิ่มประสิทธิภาพโดยการขยายการรองรับของระบบให้มากขึ้น โดยการนำเทคโนโลยีของ JMS เข้ามาช่วย

ระบบงานที่ทดลองสร้างขึ้น มีการใช้เทคโนโลยีต่างๆ ที่ได้ศึกษามา โดยได้ทำการสร้างระบบงานเป็น 3 เทียร์ คือ

1. *พรีเซนเตชั่นเทียร์* มีไคลเอ็นต์เป็นเว็บบราวเซอร์ที่เรียกใช้บริการจากระบบและแอพพลิเคชันที่ทำหน้าที่เป็นเครื่องมือในการจัดการระบบ
2. *บิสิเนสเทียร์* มีเว็บเซิร์ฟเวอร์ Weblogic ติดตั้ง โดยในส่วนของเว็บแอพพลิเคชั่นและเว็บเซอร์วิสมีการเรียกใช้บริการของ EJB ด้วย
3. ดาต้าเทียร์ ใช้ดาตาเบสเซิร์ฟเวอร์เป็น Oracle 9iAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 แนวทางการพัฒนาต่อ

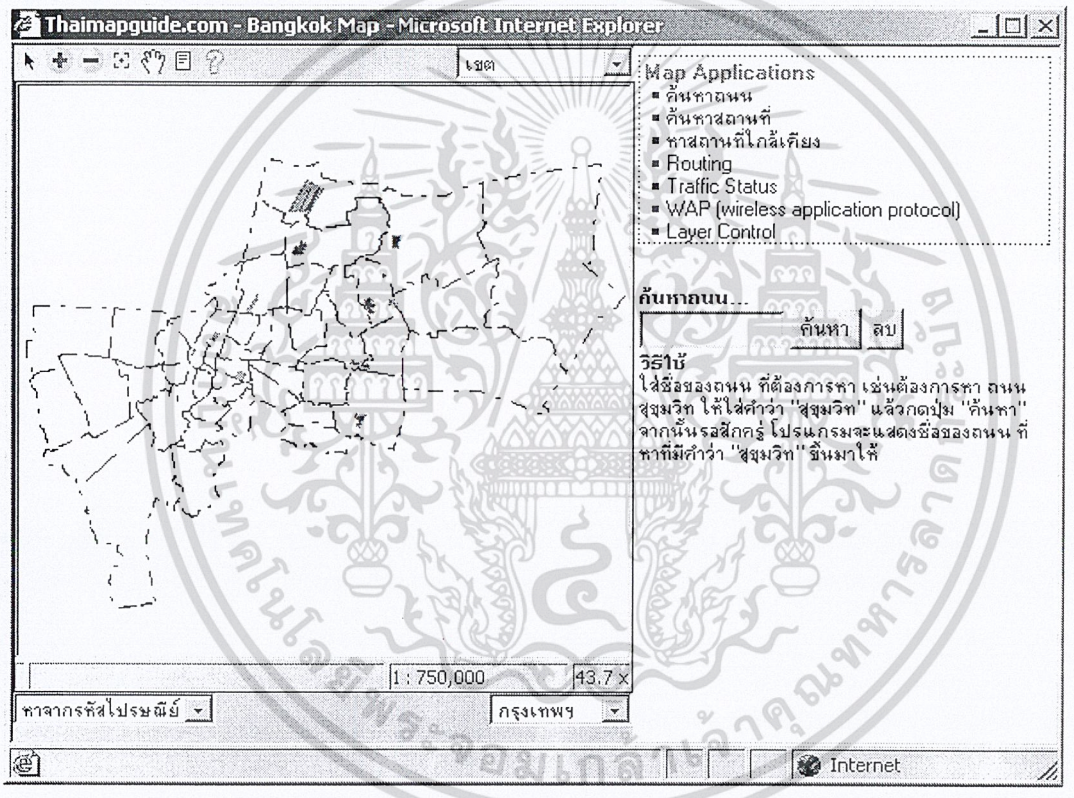
ถึงแม้ว่า การทดลองสร้างระบบงานเว็บเซอร์วิสให้สามารถเรียกใช้บริการจากระบบงานอื่นที่พัฒนามาจากเทคโนโลยีที่แตกต่างกันได้ (คือ .NET และ EJB) โดยผ่านทางโพรโตคอล SOAP นั้น สามารถทำงานได้เป็นอย่างดีก็ตาม แต่ก็ยังเป็นเพียงการแลกเปลี่ยนข้อมูลและเรียกใช้เซอร์วิสบริการแบบง่ายๆ เพียงแค่รับส่งข้อมูลผ่านกันได้รูปแบบของ XML เท่านั้น แต่การใช้งานความสามารถที่เพิ่มขึ้น เช่น การทำสเตทฟูลเว็บเซอร์วิสสามารถทำได้ภายในแพล็ตฟอร์มเดียวกัน หรือถ้าเป็นจาวาต้องภายในผลิตภัณฑ์ตัวเดียวกันด้วย การจัดการทรานแซ็กชันที่ไคลเอนต์ไม่สามารถทำ two phrase commit ระหว่างสองเว็บเซอร์วิสได้ รวมไปถึงหัวข้อสำคัญคือ ความปลอดภัยระหว่างการสื่อสารในขณะนี้ก็ทำโดยนำ โปรโตคอล SSL เข้ามาช่วยแต่ก็ส่งผลให้ความเร็วลดลงเป็นอย่างมาก

การนำเว็บเซอร์วิสเพื่อไปใช้งานนั้น ยังคงต้องมีการพัฒนามาตรฐานในเรื่องต่างๆออกมารองรับอีกมากมาย ดังเช่นเรื่องของทรานแซ็กชัน ที่คาดว่าจะมีมาตรฐานเปิดตัวออกมาในปี 2547 มาตรฐานทางด้านความปลอดภัยซึ่งขณะนี้ได้ออกมาแล้ว โดยทางไมโครซอฟท์เพิ่งมีปลั๊กอินเข้ามาสนับสนุนการทำงานเมื่อไม่นานมานี้ ทางฝั่งจาวาก็ต้องรอผลิตภัณฑ์ของหลายๆบริษัทที่จะออกมารองรับ ซึ่งมาตรฐานเหล่านี้ก็เป็นสิ่งที่ควรศึกษาและนำมาพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การหาระยะทางระหว่างเขตในกรุงเทพฯ

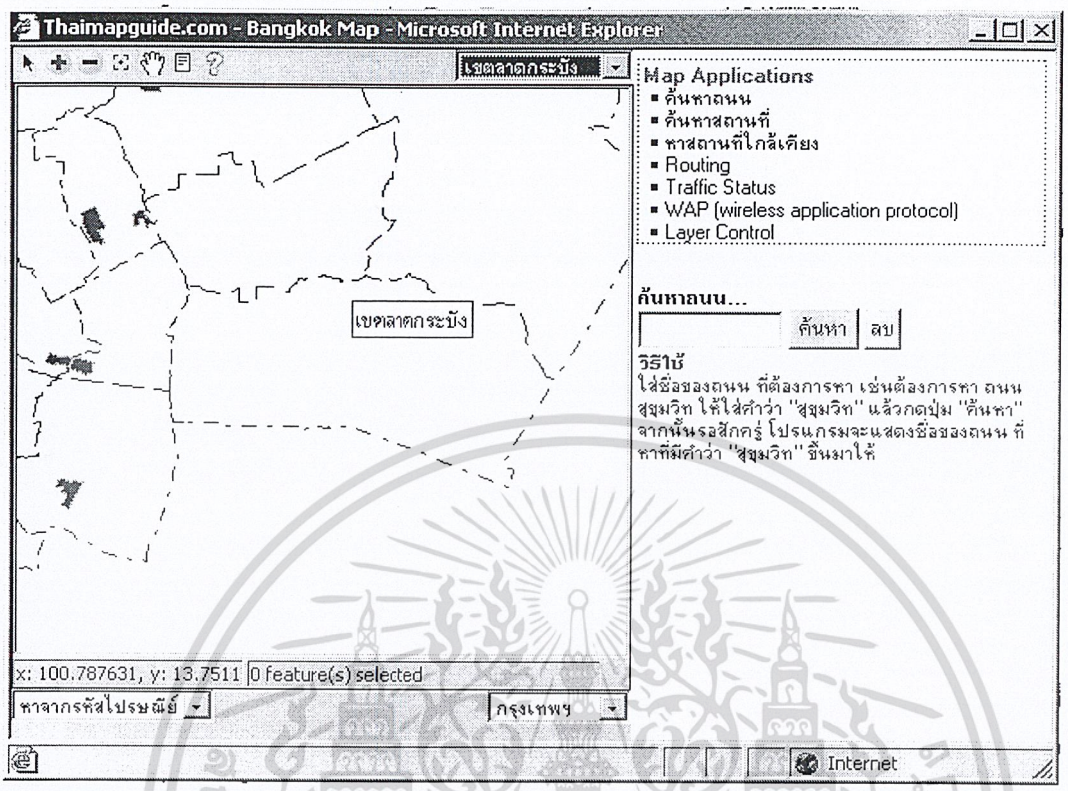
ในการหาว่าบริษัท หรือ ร้านค้าไหนอยู่ใกล้กับเขตที่ระบุไว้มากที่สุดนั้นเราจะต้องทราบว่าระยะทางระหว่างเขตต่างๆในกรุงเทพนั้นเป็นเท่าไรโดยในที่นี้ได้ใช้แอพพลิเคชันของ thaimapguide โดยเข้าไปที่ url (http://www.thaimapguide.com) โดยจะเป็นโปรแกรมที่สามารถระบุพิกัด(x,y)ของเขตต่างๆได้ ดังรูปที่ ก-1



รูปที่ ก-1 แอพพลิเคชันของ thaimapguide

โดยการหาระยะทางระหว่างเขตต่างๆนั้นเราจะหาโดยใช้พิกัดของเขตต่างๆ บนแผนที่โดยยึดพิกัด ณ จุดกึ่งกลางของเขตนั้นๆเป็นหลัก เราสามารถหาพิกัดได้โดยนำเมาส์(mouse) ไปวางเหนือตำแหน่งนั้น ดังรูปที่ ก-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-2 เมื่อนำเมาส์ไปวางเหนือตำแหน่งใดๆบนแผนที่ จะมีพิกัดบอกอยู่ที่ตำแหน่งล่างซ้ายของหน้าต่าง

เราจะทำอย่างนี้เช่นเดียวกับทุกเขตทั้ง 50 เขต ได้พิกัด(x,y) ของเขตต่างดังตารางที่ 1

เขต	x	y
หนองแขม	3583	6931
ทวีวัฒนา	3677	7732
บางบอน	3995	6538
บางแค	4006	7096
บางขุนเทียน	4307	5830
ตลิ่งชัน	4374	7698
ภาษีเจริญ	4465	7209
จอมทอง	4685	6833
บางกอกน้อย	4758	7582
บางกอกใหญ่	4791	7320

เขต	x	y
ราชเทวี	5438	7524
พญาไท	5476	7784
จตุจักร	5687	8215
หลักสี่	5719	8744
ดินแดง	5666	7714
วัฒนา	5873	7286
คลองเตย	5758	7111
ดอนเมือง	5930	9208
ห้วยขวาง	5870	7605
พระโขนง	6154	6911

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางพลัด	4992	7833
ธนบุรี	4877	7093
ทุ่งครุ	4995	6235
ราษฎร์บูรณะ	5011	6689
พระนคร	5006	7523
บางคอแหลม	5124	6932
คลองสาน	5059	7197
ดุสิต	5217	7761
สัมพันธวงศ์	5114	7366
ป้อมปราบ	5146	7480
สาธร	5335	7102
บางซื่อ	5296	8146
บางรัก	5259	7233
ปทุมวัน	5359	7371
ยานนาวา	5430	6862

บางเขน	6434	8649
ลาดพร้าว	6104	8207
วังทองหลาง	6136	7728
บางนา	6179	6673
สวนหลวง	6292	7211
บางกะปิ	6460	7660
สายไหม	6467	9082
บึงกุ่ม	6544	8033
ประเวศ	6801	6970
คันนายาว	6809	8174
สะพานสูง	6928	7568
คลองสามวา	7435	8707
มีนบุรี	7577	8013
ลาดกระบัง	7871	7428
หนองจอก	8587	8554

ตารางที่ ก-1 แสดงพิกัด (x,y) ของเขตต่างๆในกรุงเทพฯ

เราสามารถหาระยะทางระหว่างเขตได้จากสูตร

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

โดยเราจะใช้ค่าพิกัดหลังจุดทศนิยม(ใช้ 4 ตัว) เนื่องจากค่าจำนวนเต็ม(หน้าจุดทศนิยม)ของทุกเขตมีค่าเท่ากันหมดทั้ง x และ y คือ (x,y) = (100,13) ที่เราสามารถนำค่าหลังทศนิยมมาคิดได้เนื่องจากว่าเราไม่ได้ต้องการหาระยะทางจริงเพียงแต่ต้องการนำมาเปรียบเทียบกันว่าตำแหน่งไหนใกล้กว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Ed Roman : "*Mastering Enterprise JavaBeans*" , Wiley Computer Publishing , 1999

[2] Ed Roman , Scott Ambler , Tyler Jewell : "*Mastering Enterprise JavaBeans Second Edition*" , Wiley Computer Publishing , 2002

[3] Danny Ayers , Hans Bergsten , Michael Bogovich , Jason Diamond , Matthew Ferris , Marc Fleury , Ari Halberstadt , Paul Houle , Piroz Mohseni , Andrew Patzer , Ron Phillips , Sing Li , Krishna Vedati , Mark Wilcox and Stefan Zeiger : "*Professional Java Server Programming*" , Wrox Press Ltd , 1999

[4] Sun Microsystems : "*The Java™ 2 Enterprise Edition Developer's Guide*" , 2000

[5] Deepak Alur , John Crupi , Dan Malks : "*core J2EE Patterns Best Practices and Design Strategies*" , Prentice Hall PTR , 2001

[6] BEA Systems , Inc : "*Programming WebLogic Enterprise JavaBeans*" , 2001

[7] BEA Systems , Inc : "*BEA WebLogic Server Administration Guide*" , 2001

[8] Patrick Cauldwell, Rajesh Chawla, Vivek Chopra, Gary Damschen, Chris Dix, Tony Hong, Francis Norton, Uche Ogbuji, Glenn Olander, Mark A. Richman, Kristy Saunders, and Zoran Zaev : "*Professional XML Web Services*" , Wrox Press Ltd , 2001

[9] Alan W. Brown : "*Large-Scale , Component-Based Development*" , Prentice Hall PTR , 2000

[10] BEA Systems,Inc. : "BEA Weblogic Server : Programming Weblogic JTA" , BEA e-docs , 2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้