

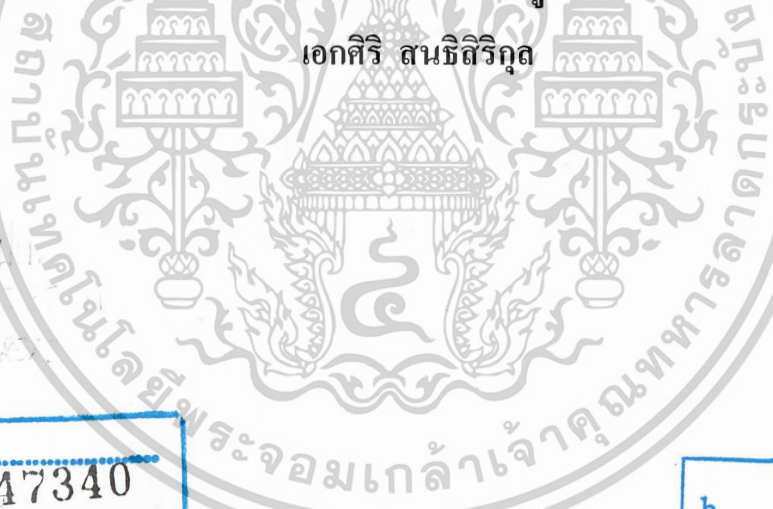
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสาธิตการใช้งานเว็บเซอร์วิส

WEB SERVICES DEMONSTRATION



อรรณีย์ อินทรนางกุล
เอกศิริ สนิธิศิริกุล



เลขหม.....
เลขทะเบียน..... 47340
วัน, เดือน, ปี..... 30 ส.ย. 2546

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

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WEB SERVICES DEMONSTRATON



A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การสาธิตการใช้งานเว็บเซอร์วิส
 WEB SERVICES DEMONSTRATION

ชื่อนักศึกษา อรรถนีย์ อินทรนางกุล 42050465
 เอกศิริ สนธิศิริกุล 42050474

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา รศ.ธีรวัฒน์ ประกอบผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์สถาบันเทคโนโลยี-
 พระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตาม
 หลักสูตร วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2545

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	อ.วีระชัย ต้นยะสิทธิ์	
กรรมการ	อ.รัชต์ธร อาชีวะ	 ๓๓๖
กรรมการและอาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	 ๓๓๖



(ผู้ช่วยศาสตราจารย์ไพโรบลย์ พันธรักษ์พงษ์)
 หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การสาธิตการใช้งานเว็บเซอร์วิส	
ชื่อนักศึกษา	อรรถนีย์ อินทรนางกุล	42050465
	เอกศิริ สนธิศิริกุล	42050474
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2545	
อาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	

บทคัดย่อ

เว็บเซอร์วิส โมเดล (The Web Services model) เป็น โมเดลที่ถูกนำมาใช้ในการพัฒนา distributed applications) อย่างแพร่หลาย เนื่องจากมีความสามารถที่ช่วยให้ นักพัฒนาโปรแกรม สามารถพัฒนาและเรียกใช้แอปพลิเคชันร่วมกันได้อย่างอิสระ ไม่ขึ้นกับแพลตฟอร์ม, ระบบปฏิบัติการ (OS) และ อุปกรณ์ปลายทาง

เนื่องจากการศึกษาเว็บเซอร์วิส โมเดลนั้นมีประโยชน์อย่างมาก เราจึงได้ทำการค้นคว้า และ รวบรวมความรู้เกี่ยวกับโมเดลและได้ทดลองสร้างระบบบริการจำหน่ายตั๋วชมภาพยนตร์ (Ticket agent) ขึ้น เพื่อทดสอบความเข้าใจ และสาธิตการทำงานของโมเดล

ระบบบริการจำหน่ายตั๋วชมภาพยนตร์ (Ticket Agent System) นั้น เป็นแหล่งอำนวยความสะดวกของโรงภาพยนตร์ต่างๆและลูกค้า โดยจะทำการรวบรวมข้อมูลต่างๆของโรงภาพยนตร์มาไว้ในที่แห่งเดียว รวมทั้งสามารถทำการจองตั๋วผ่านเว็บแอปพลิเคชันนี้ได้ด้วย

ในขณะเดียวกัน สำหรับด้านโรงภาพยนตร์ก็สามารถเปิดให้บริการต่างๆผ่าน เว็บเซอร์วิส ของตนซึ่งสร้างด้วยภาษาอะไรก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	WEB SERVICES DEMONSTRATION	
Students	Miss Auttanee Intaranangkul	42050465
	Mr.Ekasiri Sontisirikul	42050474
Degree	Bachelor of Science	
Department	Mathematics and Computer Science, Faculty of Science	
Programme	Computer Science	
Academic Year	2002	
Special Project Advisor	Assoc. Prof. Teerawat Prakobphon	

ABSTRACT

The Web Services model has been more rapidly and widely adopted than any other approach to building distributed applications to date. The phenomenal success of the Web model can be attributed to one core characteristic: it is more loosely coupled than traditional distributed programming models which provide a platform for building distributed applications using software running on different operating systems and devices, written using different programming languages and tools from multiple vendors, all potentially developed and deployed independently.

At this point, Learning about Web Services model is usefully. We had searched and comprehended about the Model, then demonstrated by building The Ticket Agent System. The Ticket Agent System is a dealer system that brings customer and Theatre's Business meets. They provide a customer to booking and find all theater information in one place. In the other hand, because of the loosely coupled model, Businesses can join this system efficiently by building the web services on their server side.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการสาธิตการใช้งานเว็บเซอรัวิสสามารถลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ รองศาสตราจารย์ธีรวัฒน์ ประกอบผล อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งเพื่อนๆ และ น้องๆทุกคน ที่ได้ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้



คณะผู้จัดทำ
มีนาคม 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VIII
สารบัญตาราง.....	X

บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา.....	2
1.5 ขอบเขตการศึกษา.....	2
1.6 ขั้นตอนและวิธีการดำเนินการศึกษา.....	2
1.7 ข้อจำกัดของการศึกษา.....	2
บทที่ 2 ทฤษฎีหรือหลักการ.....	4
2.1 ความรู้พื้นฐานและการเขียนโปรแกรมด้วยภาษา C# เบื้องต้น.....	4
2.1.1 ตัวแปร (Variable).....	4
2.1.1.1 ตัวแปรแบบมีค่า.....	4
2.1.1.2 ตัวแปรแบบอ้างค่า.....	5
2.1.2 โอเปอเรเตอร์ (Operator).....	5
2.1.2.1 Assignment Operator 5	5
2.1.2.2 Arithmetic Operator.....	5
2.1.2.3 Concatenating Operator.....	5
2.1.2.4 Comparison Operator.....	5
2.1.2.5 Logical Operator.....	6
2.1.3 เงื่อนไขควบคุม(Decision).....	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

2.1.3.1	if...else.....	6
2.1.3.2	switch...case.....	6
2.1.4	การวนลูป (Looping).....	6
2.1.4.1	ลูป For.....	7
2.1.4.2	ลูป While.....	7
2.1.4.2	ลูป Do...While.....	7
2.1.4.3	ลูป foreach.....	7
2.1.5	ฟังก์ชัน (Function).....	7
2.1.6	ขอบเขตของตัวแปร (Scope).....	8
2.1.6.1	ตัวแปรGlobal.....	8
2.1.6.2	ตัวแปรLocal.....	8
2.2	ความรู้พื้นฐานและการเขียนโปรแกรมด้วยภาษา VB.NET เบื้องต้น.....	8
2.2.1	ตัวแปร (Variable).....	9
2.2.2	โอเปอเรเตอร์(Operator).....	9
2.2.2.1	Comparison Operator.....	9
2.2.2.2	Arithmetic Operator.....	9
2.2.2.3	Logical Operator.....	10
2.2.3	เงื่อนไขควบคุม(Decision).....	10
2.2.3.1	If...Then...Else.....	10
2.2.3.2	Select...Case.....	10
2.2.4	การวนลูป (Looping).....	10
2.2.4.1	ลูป For.....	10
2.2.4.2	ลูป While.....	11
2.2.4.3	ลูป Do...Loop.....	11
2.2.5	โปรแกรมย่อย.....	11
2.2.5.1	Sub Routine.....	11
2.2.5.2	Function.....	11
2.2.6	ขอบเขตของตัวแปร (Scope).....	11
2.2.6.1	Global.....	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

2.2.6.2 Module.....	12
2.2.6.3 Procedure.....	12
2.2.6.4 Block.....	12
2.3 ความรู้พื้นฐานและโครงสร้างของADO.Net.....	12
2.3.1 โครงสร้างของ ADO.Net.....	12
2.3.2 System.Data.....	13
2.3.3 System.Data.OleDb และ System.Data.SqlClient.....	14
2.3.4 หลักการติดต่อกับฐานข้อมูลมีขั้นตอนดังต่อไปนี้.....	14
2.3.5 Namespace ที่ต้องใช้.....	15
2.3.6 สร้าง Connection ไปที่ฐานข้อมูล.....	15
2.3.7 เปิด และ ปิด Connection.....	16
2.3.8 สร้าง SQL Command.....	16
2.4 ความรู้เบื้องต้นเกี่ยวกับเว็บเซอร์วิส.....	17
2.4.1 แนวคิดหลักของการดำเนินธุรกิจผ่านทางเว็บเซอร์วิส.....	17
2.4.2 ปัจจัยพื้นฐานของเว็บเซอร์วิส.....	17
2.4.3 เทคโนโลยีมาตรฐาน สำหรับการพัฒนาเว็บเซอร์วิส.....	18
2.4.3.1 XML (The Extensible Markup Language 1.0).....	18
2.4.3.2 SOAP (Simple Object Access Protocol).....	19
2.4.3.3 WSDL (Web Services Description Language).....	22
2.4.3.4 UDDI (Universal Description, Discovery and Integration).....	22
2.4.4 สรุปความรู้เบื้องต้นเกี่ยวกับเว็บเซอร์วิส.....	26
บทที่ 3 ขั้นตอนและวิธีดำเนินการวิจัย.....	28
3.1 ขั้นตอนการวิเคราะห์ และ ออกแบบระบบ (Specification Analysis).....	28
3.1.1 Domainระบบตัวแทนจำหน่ายตัวชมภาพยนตร์.....	28
3.1.2 USE CASE ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์.....	29
3.1.3 Statechart Diagram ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์.....	31
3.1.4 Activity Diagram ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์.....	33
3.2 ขั้นตอนการดำเนินงาน (Implementing).....	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

3.2.1	ขั้นตอนการสร้างฐานข้อมูลของโรงพยาบาลนตร์ และ ระบบตัวแทน เจ้าหน้าที่วชมภพยนตร์.....	34
3.2.2	ขั้นตอนการสร้าง Web services ของโรงพยาบาลนตร์.....	41
3.2.3	ขั้นตอนการสร้าง Web page ของระบบตัวแทนเจ้าหน้าที่วชมภพยนตร์ ...	44
3.3	ขั้นตอนการทดสอบการทำงาน(Testing).....	45
<hr/>		
บทที่ 4	สรุปผลการศึกษาและข้อเสนอแนะ.....	51
4.1	การทำงานของระบบ	51
4.2	ผลของปัญหาพิเศษ	51
4.3	ข้อจำกัดของปัญหาพิเศษ	51
4.4	ปัญหาในการทำงาน	51
<hr/>		
	เอกสารอ้างอิง.....	53
	ภาคผนวก ก. การติดตั้งโปรแกรมและการทำงานเบื้องต้น.....	54
	ภาคผนวก ข. สถาปัตยกรรมของโปรแกรม.....	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
2.1 DataSet.....	15
2.2 ตัวอย่างการทำงานของ SOAP.....	23
2.3 โครงสร้างของเอกสาร SOAP	25
2.4 ภาพแสดงความสัมพันธ์ระหว่างผู้ขอบริการกับผู้ให้บริการและ UDDI.....	26
2.5 ภาพการทำงานของ Web Services ที่เกิดจากการทำงานร่วมกันของ SOAP,UDDI.....	27
3.1 USE CASE ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์.....	30
3.2 USE CASE ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์(ต่อ)	31
3.3 Statechart Diagram ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์.....	32
3.4 Statechart Diagram ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์.....	33
3.5 Activity Diagram ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์เมื่อทำการจอง.....	34
3.6 การสร้างฐานข้อมูลด้วยโปรแกรม Microsoft SQL Server 2000.....	35
3.7 การสร้างฐานข้อมูลด้วยโปรแกรม Microsoft SQL Server 2000(ต่อ).....	35
3.8 การสร้างฐานข้อมูลด้วยโปรแกรม Microsoft SQL Server 2000(ต่อ).....	36
3.9 Table และ Relationของฐานข้อมูลโรงภาพยนตร์ที่1 (Theater1).....	38
3.10 Table และ Relationของฐานข้อมูลโรงภาพยนตร์ที่2 (Theater2).....	40
3.11 Table และ Relationของฐานข้อมูลตัวแทนจำหน่ายตัวชมภาพยนตร์.....	41
3.12 การ coding web Service ในการให้ข้อมูลจำนวนที่นั่งในโรงภาพยนตร์.....	42
3.13 การเรียก Web Service เพื่อทดสอบการทำงาน โดยจะมีเอกสาร SOAP กำกับ.....	42
3.14 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของServiceนี้ได้กลับมาเป็น Integer.....	43
3.15 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของServiceนี้ได้กลับมาเป็น DataSet.....	43
3.16 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของServiceนี้ได้กลับมาเป็น String.....	44
3.17 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของServiceนี้ได้กลับมาเป็น String.....	44
3.18 การสร้าง Web page REVIEW โดยโปรแกรม Dreamwever.....	45
3.19 การ running server.....	45
3.20 แสดงหน้าจอการเข้าสู่ web site AI Entertainment.....	46
3.21 แสดง Review web page บอกรายละเอียดของภาพยนตร์ที่กำลังฉาย.....	46
3.22 แสดงหน้าจอ Trailers แสดงตัวอย่างของภาพยนตร์.....	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.23 ShowTime web page แสดงตารางเวลาของภาพยนตร์ที่กำลังเข้าฉาย.....	47
3.24 แสดงการเข้าสู่ web page การจองตั๋ว.....	48
3.25 แสดงการเลือกโรงภาพยนตร์.....	48
3.26 แสดงการเลือกเก้าอี้ประเภท normal.....	49
3.27 แสดง web page การจอง.....	49
3.28 แสดงการจองที่นั่ง F3 – F7 จะทำให้ที่นั่งที่เลือกเปลี่ยนเป็นสีน้ำเงิน.....	50
3.29 แสดงผลสรุปการจองที่นั่งที่สำเร็จ โดยจะแสดงรหัสในการจอง ที่นั่งที่จอง.....	51
3.30 แสดงผลสรุปการจองที่นั่งที่ไม่สำเร็จ.....	51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่

2.1 ตัวแปรแบบมีค่าในภาษา C#.NET และค่าที่เก็บ.....	4
2.2 โอเปอเรเตอร์ชนิด Arithmetic Operator ในภาษา C#.NET.....	5
2.3 โอเปอเรเตอร์ชนิด Comparison Operator ในภาษา C#.NET.....	6
2.4 โอเปอเรเตอร์ชนิด Logical Operator ในภาษา C#.NET.....	6
2.5 ตัวแปรในภาษา VB.NET และค่าที่เก็บ.....	9
2.6 โอเปอเรเตอร์ชนิด Comparison Operator ในภาษา VB.NET.....	10
2.7 โอเปอเรเตอร์ชนิด Arithmetic Operator ในภาษา VB.NET.....	10
2.8 โอเปอเรเตอร์ชนิด Logical Operator ในภาษา VB.NET.....	11
2.9 การเปรียบเทียบ class OleDb และ System.Data.SqlClient.....	16
3.1 รายละเอียดของฐานข้อมูลของ โรงพยาบาลที่ 1.....	36
3.2 รายละเอียดของฐานข้อมูลของ โรงพยาบาลที่ 2.....	38
3.3 รายละเอียดของฐานข้อมูลของตัวแทนจำหน่ายตัวหมภาพยนต์.....	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

การดำเนินงานของผู้ใช้อินเทอร์เน็ตเริ่มเปลี่ยนรูปแบบ โดยเฉพาะการโต้ตอบแบบสองทางได้ จึงสร้างบทบาทที่สำคัญหลายอย่างตามมา โดยเฉพาะการดำเนินงานทางธุรกิจ เริ่มตั้งแต่การโฆษณาประชาสัมพันธ์ผ่านทางเว็บ การให้เป็นสถานที่สำหรับติดต่อหรือบริการข้อมูลข่าวสาร การทำธุรกรรมการซื้อขายในรูปแบบอีคอมเมิร์ซ (E-commerce) การทำธุรกรรมอิเล็กทรอนิกส์ ซึ่งเป็นการแลกเปลี่ยนข้อมูลข่าวสารระหว่างกันบนเครือข่ายหรือที่เรียกว่า อีบิสซิเนส (e-business) โอกาสของการดำเนินงานบนเครือข่ายจึงมีอีกมาก และมีแนวโน้มที่จะพัฒนาการทำงานรูปแบบใหม่ ๆ อีกหลายอย่าง โดยเฉพาะเทคโนโลยีการสื่อสารแบบไร้สาย และระบบโทรศัพท์เคลื่อนที่ กำลังเข้ามาแพร่หลายสู่เครือข่ายอินเทอร์เน็ต การดำเนินกิจกรรมต่าง ๆ จึงกระทำได้ทุกหนทุกแห่ง ไม่ว่าจะอยู่ที่ใดก็ได้

ปัจจุบันการพัฒนาเว็บ ได้ก้าวหน้าขึ้นไปอีกมาก เพื่อรองรับการใช้งานที่มีความต้องการสูงขึ้น โดยเฉพาะการดำเนินการทางธุรกิจการค้าต่าง ๆ ที่เพิ่มมากขึ้นบนเครือข่าย ลักษณะงานที่ต้องการมีลักษณะการทำงานร่วมกันระหว่างองค์กร (interoperability) โดยให้โปรแกรมประยุกต์ขององค์กรหนึ่งส่งคำขอผ่านเครือข่ายอินเทอร์เน็ตด้วยโปรโตคอล http ไปยังเว็บบริการของอีกองค์กรหนึ่ง มีการโต้ตอบเพื่อรับส่งข้อมูลระหว่างกันแบบอัตโนมัติได้

การเรียกใช้บริการต่าง ๆ มีลักษณะการเรียกใช้บริการโดยมีตัวกลางช่วยและเชื่อมโยงเข้ากับเว็บเซิร์ฟเวอร์ แลกเปลี่ยนข้อมูลตามคำเรียกขอโดยใช้มาตรฐาน XML-eXtensible Mark Up Language การบริการบนเว็บจึงเป็นการก้าวเข้ามาในระบบที่ทำให้เกิดการแลกเปลี่ยนข้อมูลระหว่างกัน ซึ่งแต่เดิมมีวิธีการแลกเปลี่ยนข้อมูลกันตามรูปแบบที่เรียกว่า EDI (Electronic Data Interchange) แต่จากการที่มีการใช้ XML ทำให้เพิ่มขีดความสามารถในการทำงานยิ่งขึ้น

ปัจจุบันบริษัทต่าง ๆ ได้ขยายคุณสมบัติประโยชน์ของการดำเนินการแบบอัตโนมัติไปสู่ลูกค้าของบริษัท ด้วยการสร้างโปรแกรมสำหรับให้บริการทางด้านข้อมูลและให้บริการทางด้านการประมวลผลโดยใช้โปรแกรมของบริษัท ให้แก่ลูกค้าของบริษัท และอนุญาตให้เครื่องคอมพิวเตอร์ของลูกค้าของบริษัทสามารถติดต่อสื่อสาร ขอบริการข้อมูลและบริการงานโปรแกรมจากเครื่องคอมพิวเตอร์ของบริษัทที่ติดตั้งโปรแกรมที่ให้บริการโดยตรง เพื่อดำเนินกิจกรรมต่างๆ ตามขั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนและกระบวนการทางของธุรกิจที่ได้ออกแบบไว้ ซึ่งการให้บริการในรูปแบบดังกล่าว ก็คือ “เว็บเซอร์วิส (Web Services)” นั่นเอง

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่ออำนวยความสะดวกแก่ผู้ใช้งานในการทำงานข้ามแพลตฟอร์ม (platform)

1.2.2 เพื่อสนับสนุนการทำธุรกิจแบบB2B (Business to Business) หรือแบบพาณิชย์อิเล็กทรอนิกส์ (E-Commerce)

1.2.3 เพื่อความสะดวกในเรื่องสถานที่ใช้งาน โดยจะสามารถเข้าถึงข้อมูลได้ทั้งแบบออฟไลน์ (โดยApplication ภายในเครื่อง) และแบบออนไลน์ (โดยApplicationที่อยู่บนWeb) ทั้งทางเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ และทางอุปกรณ์แบบไร้สาย เช่น โทรศัพท์มือถือ

1.2.4 เพื่อประโยชน์ในการใช้และการจัดการข้อมูลหรือซอฟต์แวร์ร่วมกัน

1.3 สมมติฐานของการศึกษา

1.3.1 สามารถสร้างเซอร์วิสให้เรียกใช้งาน เพื่อทดสอบการใช้งานตามแนวคิดเว็บเซอร์วิส ได้

1.3.2 สามารถสร้าง เว็บแอปพลิเคชันที่ทำงานโดยเรียกใช้เว็บเซอร์วิสได้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา

1.4.1 แนวคิดการทำงานแบบ XML-Web services

1.4.2 XML (The Extensible Markup Language 1.0)

1.4.3 SOAP (Simple Object Access Protocol) เป็น XML-based โปรโตคอล (lightweight protocol)

1.4.4 ภาษาโปรแกรม Visual Basic.NET

1.4.5 ภาษาโปรแกรม C#.NET

1.4.6 การสร้างระบบฐานข้อมูล

1.5 ขอบเขตของการศึกษา

1.5.1 สร้างระบบบริการจำหน่ายตั๋วชมภาพยนตร์ (Ticket Agent System) ที่สามารถติดต่อกับโรงภาพยนตร์ทุกเครื่องที่มีส่วนให้บริการเว็บเซอร์วิส และมีเซอร์วิสพื้นฐานที่ระบบกำหนดไว้ เช่น เซอร์วิสในการจอง เซอร์วิสในการสอบถามที่นั่ง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5.2 สร้างระบบโรงภาพยนตร์รวมทั้งระบบจัดการฐานข้อมูลและ เซอร์วิสที่ให้ตัวแทนจำหน่ายตั๋ว (ticket agent) นำไปใช้

1.6 ขั้นตอนและวิธีดำเนินการศึกษา

1.6.1 ศึกษาการทำงานของเว็บเซอร์วิสโดยศึกษาทั้งภาษาและโปรโตคอลที่ใช้

1.6.2 วิเคราะห์และออกแบบการทำงานของโปรแกรม โดยออกแบบฐานข้อมูล (Database) และแอปพลิเคชัน (Application บนserver) จากนั้นสร้างเว็บแอปพลิเคชัน (Web Application) ของระบบบริการจำหน่ายตั๋วชมภาพยนตร์ (ticket agent) เพื่อสาธิตการทำงานกับดาตาเบสเซิร์ฟเวอร์ (Database Server) ผ่านเซอร์วิสที่สร้างไว้

1.6.3 ทำการทดสอบโปรแกรมและปรับปรุงแก้ไข

1.6.4 จัดทำเอกสารประกอบการใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานและการเขียนโปรแกรมด้วยภาษา C# เบื้องต้น

2.1.1 ตัวแปร (Variable)

2.1.1.1 ตัวแปรแบบมีค่า คือ ตัวแปรที่สามารถระบุค่าที่แน่นอนเพียงค่าเดียวได้

1. ตัวแปรชนิดตัวเลข (Numeric)
2. ตัวแปรชนิดอักขระ (Character)
3. ตัวแปรบูลีน (Boolean)
4. ตัวแปรสตรักเจอร์ (Structure)
5. ตัวแปรอินัมเมอเรชัน (Enumeration)

ตารางที่ 2.1 ตัวแปรแบบมีค่าในภาษาC#.NET และค่าที่เก็บ

C# Type	ค่าที่เก็บ	.NET Type
bool	True/False	System.Boolean
byte	0 ถึง 255	System.Byte
Sbyte	-128 ถึง -127	System.SByte
char	ตัวอักษรแบบ Unicode 16 bits	System.Char
decimal	1.0e-28 ถึง 7.9e28	System.Decimal
double	-5.0x10e-324 ถึง 1.7e10e308	System.Double
Float	-1.5x10e-45 ถึง 3.4e38	System.Single
Int	-2,147,483,648 ถึง -2,147,483,647	System.Int32
uint	0 ถึง 4,294,967,295	System.UInt32
Long	-9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807	System.Int64
ulong	0 ถึง 18,446,744,073,709,551,615	System.UInt64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Short	-32,768 ถึง 32,767	System.Int16
ushort	0 ถึง 65,535	System.UInt16
Enum	enumeration	
Struct	structure	

2.1.1.2 ตัวแปรแบบอ้างค่า คือ ตัวแปรที่ถูกเก็บไว้เป็นชุดของข้อมูล เมื่อเวลาใช้งานต้องกำหนดตำแหน่งอ้างอิงของตัวแปรนั้น (Reference Position)

1. ตัวแปรสตริง (String)
2. ตัวแปรชนิดอาร์เรย์ (Array)
3. ตัวแปรคลาส (Classes)
4. ตัวแปรอินเตอร์เฟซ (Interface)
5. ตัวแปรออบเจกต์ (Objects)

2.1.2 โอเปอเรเตอร์ (Operator)

2.1.2.1 Assignment Operator

รูปแบบการใช้งาน : ชื่อตัวแปร = ค่าที่กำหนด ;

2.1.2.2 Arithmetic Operator

ตารางที่ 2.2 โอเปอเรเตอร์ชนิด Arithmetic Operator ในภาษาC#.NET

สัญลักษณ์	ชื่อโอเปอเรเตอร์	ความหมาย
a+b	บวก	หาผลรวม
a-b	ลบ	หาผลลบ
a*b	คูณ	หาผลคูณ
a/b	หาร	หาผลหาร
a%b	หารเอาเศษ	หาเศษที่เหลือเมื่อนำ a หาร b

2.1.2.3 Concatenating Operator

ใช้เชื่อมต่อข้อความแบบสตริง โดยจะนำข้อความที่สองไปต่อท้ายข้อความแรกโดยใช้เครื่องหมาย +

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.4 Comparison Operator

ตารางที่ 2.3 โอเปอเรเตอร์ชนิด Comparison Operator ในภาษา C#.NET

สัญลักษณ์	ชื่อโอเปอเรเตอร์	ความหมาย
$a==b$	เท่ากับ	เป็นจริงเมื่อ a เท่ากับ b
$a<b$	น้อยกว่า	เป็นจริงเมื่อ a น้อยกว่า b
$a>b$	มากกว่า	เป็นจริงเมื่อ a มากกว่า b
$a<=b$	น้อยกว่าหรือเท่ากับ	เป็นจริงเมื่อ a น้อยกว่าหรือเท่ากับ b
$a>=b$	มากกว่าหรือเท่ากับ	เป็นจริงเมื่อ a มากกว่าหรือเท่ากับ b
$a!=b$	ไม่เท่ากับ	เป็นจริงเมื่อ a ไม่เท่ากับ b

2.1.2.5 Logical Operator

ตารางที่ 2.4 โอเปอเรเตอร์ชนิด Logical Operator ในภาษา C#.NET

สัญลักษณ์	ชื่อโอเปอเรเตอร์	ความหมาย
$a\&\&b$	และ	เป็นจริงเมื่อ a และ b มีค่าเป็นจริง
$a\ \ b$	หรือ	เป็นจริงเมื่อ a หรือ b มีค่าเป็นจริง
$a!\ b$	ตรงกันข้าม	เป็นจริงเมื่อ a ตรงกันข้ามกับ b มีค่าเป็นจริง

2.1.3 เงื่อนไขควบคุม (Decision)

2.1.3.1 if...else...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

if (Logical Operator)

```
{
  ...
}
```

else

```
{
  ...
}
```

2.1.3.2 switch...case

switch(statement)

```
{
  case option1:
    ...
    break;
  case option2:
    ...
    break;
  default:
    ...
  break;
}
```

2.1.4 การวนลูป (Looping)

2.1.4.1 ลูป For

for (ค่าเริ่มต้น;เงื่อนไขของค่าสิ้นสุด;ค่าที่เปลี่ยนไปในแต่ละรอบ)

```
{
  ...
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.2 รูป While

```
while (เงื่อนไขของค่าสิ้นสุด)
{
...
}
```

2.1.4.3 รูป Do...While

```
do
{
...
} while (เงื่อนไขของค่าสิ้นสุด);
```

2.1.4.4 รูป foreach

```
foreach(ตัวแปรออบเจกต์ ออบเจกต์ in ตัวแปรคอลเลกชัน)
{
...
}
```

2.1.5 ฟังก์ชัน (Function)

```
return-type function-name ( type param1, type param2, ...)
{
...
return var;
}
```

การส่งค่าเข้าไปในฟังก์ชันที่จะใช้งานมี 2 ชนิด คือ

1. การส่งค่าแบบผ่านค่า (Pass by value) คือการ copy ค่าจากผู้เรียกมาเป็นค่าอีกค่าหนึ่ง ดังนั้นเมื่อใน function แก้ parameter ตัวนั้นมันจะไม่มีผลกระทบต่อค่าที่ผู้เรียก function ส่งมา
2. การส่งค่าแบบอ้างอิง (Pass by reference) คือการส่งตำแหน่งของตัวแปรมาให้ ดังนั้นเมื่อ function แก้ parameter ตัวนั้น ผู้เรียก function ก็จะถูกแก้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 ขอบเขตของตัวแปร (Scope)

2.1.6.1 ตัวแปร Global มีขอบเขตการทำงานได้ทุกส่วนของโปรแกรม เมื่อมีการเปลี่ยนแปลงค่า ทำให้ตัวแปรที่มีค่าเปลี่ยนไปทันที

2.1.6.2 ตัวแปร Local คือตัวแปรที่ประกาศไว้ในฟังก์ชันเพื่อใช้งานภายใน โดยจะมีขอบเขตแค่ภายในฟังก์ชันเท่านั้น และถึงแม้ชื่อที่ประกาศจะเหมือนกับที่ประกาศไว้เป็นตัวแปรGlobal เมื่อมีการเปลี่ยนแปลงภายใน จะไม่มีผลต่อตัวแปรGlobal

2.2 ความรู้พื้นฐานและการเขียนโปรแกรมด้วยภาษา VB.NET เบื้องต้น

2.2.1 ตัวแปร (Variable) แบ่งออกเป็น 5 ประเภทคือ

1. Number เป็นกลุ่มข้อมูลที่เก็บตัวเลขชนิดต่างๆ ทั้งจำนวนเต็ม และเลขทศนิยม
2. String เป็นกลุ่มข้อมูลที่เก็บตัวอักษร ทั้งอักษรตัวเดียว และข้อความ
3. Date เป็นกลุ่มข้อมูลที่เก็บข้อมูลเกี่ยวกับวัน
4. Boolean เป็นข้อมูลที่เก็บค่าทางตรรกศาสตร์
5. Object เป็นกลุ่มข้อมูลที่เก็บข้อมูลชนิดอื่นๆ ไว้ภายใน และมีความสามารถในการทำงานอยู่ภายในด้วย

ตารางที่ 2.5 ตัวแปรในภาษาVB.NET และค่าที่เก็บ

ชนิดข้อมูล	ขนาดหน่วยความจำ	คำอธิบาย
Boolean	1 byte	True/False
Byte	1 byte	เก็บจำนวนเต็ม 0 ถึง 255
Char	2 byte	ตัวอักษรแบบ Unicode 16 bits
Decimal	16 byte	เก็บจำนวนเต็มและทศนิยม ซึ่งมีความละเอียดแล้วแต่กำหนด
Double	8 byte	เก็บเลขทศนิยมระหว่าง-3.402823E ถึง -1.401298E-45 สำหรับค่าลบ และ 1.401298E-45 ถึง 1.79769313486232E08 สำหรับค่าบวก
Integer	4 byte	เก็บจำนวนเต็ม -2,147,483,648 ถึง -

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		2,147,483,647
Long	8 byte	เก็บจำนวนเต็ม -263 ถึง 263-1
Short	2 byte	เก็บจำนวนเต็ม -32,768 ถึง 32,767

2.2.2 โอเปอเรเตอร์ (Operator)

2.2.2.1 Comparison Operator

ตารางที่ 2.6 โอเปอเรเตอร์ชนิด Comparison Operator ในภาษา VB.NET

สัญลักษณ์	ชื่อโอเปอเรเตอร์	ความหมาย
a=b	เท่ากับ	เป็นจริงเมื่อ a เท่ากับ b
a<b	น้อยกว่า	เป็นจริงเมื่อ a น้อยกว่า b
a>b	มากกว่า	เป็นจริงเมื่อ a มากกว่า b
a<=b	น้อยกว่าหรือเท่ากับ	เป็นจริงเมื่อ a น้อยกว่าหรือเท่ากับ b
a>=b	มากกว่าหรือเท่ากับ	เป็นจริงเมื่อ a มากกว่าหรือเท่ากับ b
a<>b	ไม่เท่ากับ	เป็นจริงเมื่อ a ไม่เท่ากับ b
Is	เป็นออบเจกต์ชนิดเดียวกัน	เป็นจริงเมื่อ a เป็นออบเจกต์ชนิดเดียวกันกับ b
Like	รูปแบบเหมือนกัน	เป็นจริงเมื่อ a มีรูปแบบเหมือนกันกับ b

2.2.2.2 Arithmetic Operator

ตารางที่ 2.7 โอเปอเรเตอร์ชนิด Arithmetic Operator ในภาษา VB.NET

สัญลักษณ์	คำอธิบาย
a+b	บวก
a-b	ลบ
a*b	คูณ
a/b	หาร
&	String Concatenation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$a \setminus b$	ผลหารที่เป็นจำนวนเต็ม
A^{\wedge}	ยกกำลัง
$A \text{ Mod } b$	หารเอาเศษ

2.2.2.3 Logical Operator

ตารางที่ 2.8 โอเปอเรเตอร์ชนิด Logical Operator ในภาษาVB.NET

สัญลักษณ์	ชื่อโอเปอเรเตอร์	ความหมาย
A and B	และ	เป็นจริงเมื่อ a และ b มีค่าเป็นจริง
A Or B	หรือ	เป็นจริงเมื่อ a หรือ b มีค่าเป็นจริง
A Not B	ตรงกันข้าม	เป็นจริงเมื่อ a ตรงกันข้ามกับ b มีค่าเป็นจริง
A Xor B	Exclusive Or	เป็นจริงเมื่อ a และ b มีค่าความเป็นจริงต่างกัน

2.2.3 เงื่อนไขควบคุม (Decision)

2.2.3.1 If...Then...Else...

if (Logical Operator)Then

...

Else

...

End if

2.2.3.2 Select...Case

Select Case statement

case option1: ...

case option2: ...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

...

Case Else: ...

End Select

2.2.4 การวนลูป (Looping)

2.2.4.1 ลูป For

For ตัวแปรใช้นับจำนวนรอบ = จำนวนรอบเริ่มต้น To จำนวนรอบสุดท้าย [Step ขั้นของการนับ]

...

Next ตัวแปรที่ใช้วัดจำนวนรอบ

2.2.4.2 ลูป While

While < ทดสอบเงื่อนไข >

< ทำงานเมื่อเงื่อนไขเป็นจริง >

End While

2.2.4.3 ลูป Do...Loop

Do While < ทดสอบเงื่อนไข >

< ทำงานเมื่อเงื่อนไขยังเป็นจริงอยู่ >

Loop

หรือ

Do Until < ทดสอบเงื่อนไข >

< ทำงานเมื่อเงื่อนไขยังเป็นเท็จ >

Loop

หรือ

Do

< ทำงานตามคำสั่ง >

Loop While < ทดสอบเงื่อนไข ถ้าเป็นจริงให้กลับขึ้นไปทำอีกรอบ >

หรือ

Do

< ทำงานตามคำสั่ง >

Loop Until < ทดสอบเงื่อนไข ถ้าเป็นเท็จให้กลับขึ้นไปทำอีกรอบ >

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 โปรแกรมย่อย

2.2.5.1 Sub Routine เป็นโปรแกรมย่อยที่เมื่อทำงานเสร็จแล้วจะไม่มีการคืนค่ากลับมายังผู้เรียกใช้

Sub ชื่อของซบรูทีน (รายการพารามิเตอร์ จะมีหรือไม่มีก็ได้)

...

End Sub

2.2.5.2 Function เป็นโปรแกรมย่อยที่เมื่อทำงานเสร็จแล้วจะทำการคืนค่ากลับมายังผู้เรียกใช้

Function ชื่อของฟังก์ชัน (รายการพารามิเตอร์ จะมีหรือไม่มีก็ได้) As ชนิดข้อมูลที่คืนให้ผู้เรียก

...

ชื่อฟังก์ชัน = ค่าที่คืนกลับ

End Sub

2.2.6 ขอบเขตของตัวแปร (Scope)

2.2.6.2 Global เป็นขอบเขตที่มองเห็นได้ทั้งโปรเจกต์

2.2.6.3 Module เป็นขอบเขตระดับของแต่ละไฟล์ในโปรเจกต์

2.2.6.4 Procedure เป็นขอบเขตระดับโปรแกรมย่อย (ทั้งในฟังก์ชันและซบรูทีน)

2.2.6.5 Block เป็นขอบเขตตัวแปรที่จะมีผลอยู่ระหว่างบล็อกของคำสั่งหนึ่งเท่านั้น

2.3 ความรู้พื้นฐานและโครงสร้างของ ADO.Net

เริ่มแรกนั้น ไมโครซอฟท์ต้องการพัฒนา DAO (Data Access Objects) ขึ้นมาเพื่อให้โปรแกรมเมอร์ที่เขียน Visual Basic สามารถเขียนโปรแกรมติดต่อกับฐานข้อมูล Microsoft Access (หรือที่รู้จักกันในชื่อของ Jet Engine) ได้ แต่อย่างไรก็ดีการใช้ DAO ติดต่อกับฐานข้อมูลในระบบที่ใหญ่ขึ้น เช่น SQL Server หรือ Oracle นั้นไม่เป็นที่น่าพอใจนักเนื่องจากประสิทธิภาพของ DAO ไม่ได้เท่าที่ควร ดังนั้นไมโครซอฟท์จึงพัฒนาโปรแกรมตัวใหม่ที่เรียกว่า RDO (Remote Data Objects) ออกมาเพื่อสนองความต้องการในระดับที่สูงขึ้นของฐานข้อมูลขนาดใหญ่ และเพื่อสามารถติดต่อกับฐานข้อมูลหลาย ๆ ประเภทจึงทำให้ต้องมี ADO (ActiveX Data Objects) ออกมา โดย ADO นั้นใช้ OLE DB หรือ ODBC ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อกับฐานข้อมูลด้วย ADO นั้นเป็นแบบที่เรียกว่า Connected Database ซึ่งก็หมายความว่า ในขณะที่เราต้องเปิดการติดต่อกับฐานข้อมูลอยู่ตลอดเวลาที่ใช้ ถ้าโปรแกรมส่วนอื่น หรือ session อื่น ต้องการใช้ฐานข้อมูลด้วย connection อันใหม่ก็ต้องถูกสร้างขึ้นมา ส่วนนี้จะเป็นปัญหาในกรณีที่มีผู้เข้าใช้โปรแกรมมาก ๆ เพราะฐานข้อมูลอย่าง Access นั้นรองรับการใช้งานพร้อม ๆ กันได้จำกัด

สำหรับ ADO.Net นั้น เราสามารถติดต่อกับฐานข้อมูลได้ 2 รูปแบบหลัก ๆ ก็คือ connect กับ disconnect การติดต่อแบบ connect โดยหลักการแล้วก็เหมือนกันกับ RecordSet ของ ADO ซึ่ง class หลักที่ใช้ใน ADO.Net นั้นคือ DataReader ส่วนการติดต่อแบบ disconnect นั้นเป็นรูปแบบใหม่ที่แทนที่จะต้องเปิด connection เข้าเอาไว้ ADO.Net จะทำ snapshot ของฐานข้อมูลที่เรากำลังติดต่อยู่เก็บไว้ในหน่วยความจำของเครื่อง ในส่วนนี้ไม่ใช่การทำ snapshot ทั้งฐานข้อมูลนะครับ ทำเฉพาะกับ query ที่เราใช้ดึงข้อมูลเท่านั้น ดังนั้นการอ่านข้อมูลก็จะเป็นการดึงผลที่ได้จากการ query มาทั้งก่อนเอามาเก็บเอาไว้ พอได้ข้อมูลมาหมดก็ disconnect ฐานข้อมูลได้ทันที ในขณะที่แบบ connect นั้นเราจะต้องเรียกข้อมูลมาทีละรายการ class หลัก ๆ ที่เกี่ยวข้องกับการติดต่อกับฐานข้อมูลแบบ disconnect ก็คือ DataAdapter และ DataSet

สำหรับ DataReader นั้นจะเหมาะกับการดึงข้อมูลไม่กี่รายการ เพราะว่าไม่เสียเวลาและทรัพยากรในการสร้าง DataAdapter กับ DataSet ขึ้นมาเพื่อดึงข้อมูลแค่ไม่กี่รายการ แต่สำหรับ DataSet นั้นจะเหมาะกับการประมวลผลข้อมูลมาก ๆ

2.3.1 โครงสร้างของ ADO.Net

ใน ADO.Net นั้นจะมี namespace ที่เกี่ยวข้องอยู่ 5 ตัวด้วยกันได้แก่ System.Data, System.Data.Common, System.Data.OleDb, System.Data.SqlClient, และ System.Data.SqlTypes

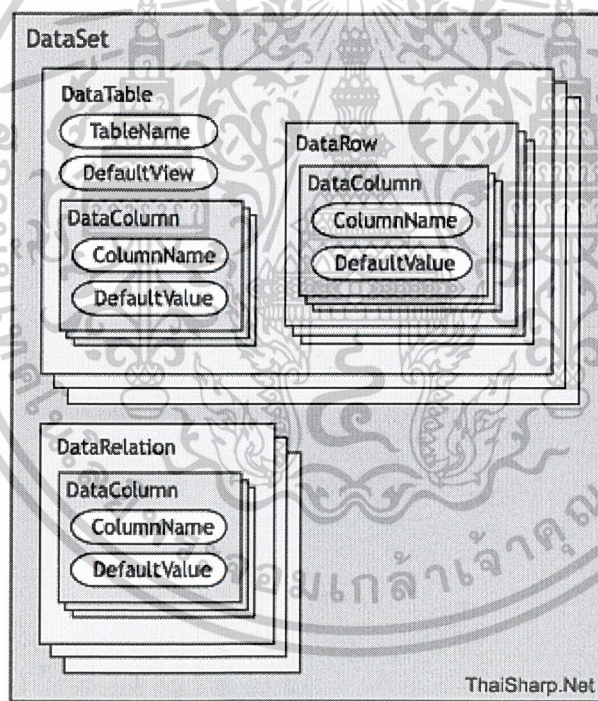
1. System.Data เป็น namespace ที่รับผิดชอบพวก class พื้นฐานที่เกี่ยวกับฐานข้อมูล เช่น DataSet, DataTable, DataRow, DataColumn, ฯลฯ
2. System.Data.Common จะมีพวก class พื้นฐานที่ใช้สำหรับการสร้าง DataAdapter สำหรับฐานข้อมูลแบบต่าง ๆ ซึ่งเราจะไม่ได้ใช้ class เหล่านี้โดยตรง แต่จะไปใช้ class ที่ derive ไปจาก class เหล่านี้อีกที
3. System.Data.OleDb รวบรวม class ที่เกี่ยวข้องกับการติดต่อกับฐานข้อมูลด้วย OLE DB ซึ่งจะเป็น provider-specific
4. System.Data.SqlClient ก็เป็นพวก class ที่ใช้ติดต่อกับ SQL Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. System.Data.SqlTypes เป็น namespace ที่รวบรวมข้อมูลเกี่ยวกับ data type ของ SQL Server

2.3.2 System.Data

สำหรับ System.Data นั้น class ที่สำคัญที่สุดคือ DataSet เพราะเป็น class หลักที่รวบรวมทุกอย่างที่เกี่ยวข้องเข้าไว้ด้วยกัน ในหนึ่ง DataSet นั้นจะประกอบไปด้วยสองส่วนหลัก ๆ คือ DataTable กับ DataRelation สำหรับ DataTable นั้นเราสามารถเก็บไว้ได้มากกว่าหนึ่ง Table ภายใน DataSet ตัวเดียวกัน Table ที่ว่าเป็น Table ที่เกิดจากการ query ฐานข้อมูล หรือเราจะสร้างขึ้นมาก็ได้เพื่อทำการ update กลับไปที่ฐานข้อมูลเมื่อพร้อม ใน DataTable ก็จะมีหลาย ๆ DataRow และ DataColumn ดังแสดงในภาพ



รูปที่ 2.1 DataSet

DataRelation นั้นเป็นการกำหนดความสัมพันธ์ระหว่าง DataTable เพื่อสร้างข้อมูลที่มีความสัมพันธ์กันอย่างเช่น เรามี DataTable ที่ชื่อ Customer กับ Order เราสามารถสร้าง DataRelation ขึ้นมาเพื่อเชื่อม Customer กับ Order ได้ โดยกำหนด key ที่จะใช้เช่น CustomerID เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 System.Data.OleDb และ System.Data.SqlClient

โดยโครงสร้างแล้ว namespace ทั้งสองตัวนี้เหมือนกัน ส่วนที่ต่างกันก็คือ System.Data.OleDb นั้นเอาไว้ใช้ติดต่อฐานข้อมูลด้วย OLE DB เช่น Jet Engine ของ MS Access ส่วน System.Data.SqlClient นั้นเอาไว้ใช้กับ SQL Server ดังตารางที่ 2.9

ตารางที่ 2.9 การเปรียบเทียบ class OleDb และ System.Data.SqlClient

OleDb	SqlCommand
OleDbConnection	SqlConnection
OleDbCommand	SqlCommand
OleDbDataAdapter	SqlDataAdapter
OleDbDataReader	SqlDataReader
OleDbParameter	SqlParameter
OleDbTransaction	SqlTransaction
OleDbException	SqlException

2.3.4 หลักการติดต่อกับฐานข้อมูล มีขั้นตอนดังต่อไปนี้

1. เปิด connection ไปที่ ฐานข้อมูล ซึ่งทำได้โดยการสร้าง object จาก OleDbConnection ขึ้นมาโดยระบุ connection string เข้าไป แล้วเรียก method ที่ชื่อ Open
2. สร้าง OleDbCommand และกำหนด CommandText ซึ่งก็คือพวก SQL statement ที่เคยใช้ ากัน แล้วก็กำหนด connection ที่จะให้ command นี้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Query, Update, หรือ Insert ข้อมูล ในขั้นนี้ทำได้ 2 วิธี อย่างที่ได้พูดถึงไว้ตอนต้น คือ connect หรือ disconnect ถ้าต้องการแบบ connect ก็ใช้ OleDbDataReader ถ้าต้องการแบบ disconnect ก็ใช้ DataSet และ OleDbDataAdapter

4. ในกรณีที่ไม่ได้ใช้ DataAdapter นั้น เราสามารถใช้ OleDbCommand ได้อยู่ 4 วิธี คือ ถ้าเป็นคำสั่งที่ไม่ใช่ SELECT ก็ใช้ ExecuteNonQuery ซึ่งผลลัพธ์ที่ได้จะเป็นจำนวน record ที่ถูกเพิ่ม หรือแก้ไขไปที่ฐานข้อมูล ส่วนถ้าเป็นคำสั่ง SELECT นั้นจะใช้ได้ 3 วิธีคือ

- ExecuteReader ซึ่งจะได้ผลออกมาเป็น DataReader ที่เราสามารถเอาไปใช้ดึงข้อมูลทีละรายการได้ เหมือนกันกับ RecordSet ของ ADO

- ExecuteScalar ซึ่งจะให้ค่าที่ได้จาก row แรก column แรก ออกมาเป็นค่า ๆ เดียว

เท่านั้น

- ExecuteXmlReader ซึ่งจะให้ผลออกมาเป็น XmlReader เพื่อใช้เอาข้อมูลที่จะได้ออกมาจากฐานข้อมูลในรูปของ XML คำสั่งนี้ก็มีประโยชน์มากครับ อย่างเช่นเราสามารถดึงข้อมูลมาเป็น XML แล้วส่งต่อไปให้ XSLT ทำการจัดผลลัพธ์ ออกมาเป็น HTML ที่เราต้องการ ทำให้ผลลัพธ์ของเราสามารถแสดงได้หลายแบบจากข้อมูลชุดเดียวกัน

5. ในกรณีที่ใช้ DataSet นั้น สามารถทำได้โดยสร้าง DataSet ขึ้นมาแล้วสั่งให้ DataAdapter นำข้อมูลมาใส่ไว้ใน DataSet

6. ปิด connection ทำได้โดยการเรียก method ที่ชื่อ Close

2.3.5 Namespace ที่ต้องใช้

สำหรับการติดต่อกับฐานข้อมูลนั้น namespace ที่ต้องใช้โดยมากแล้วจะมีอยู่ 2 ตัว คือ System.Data ซึ่งเป็นตัวพื้นฐานไม่ว่าจะติดต่อกับฐานข้อมูลยี่ห้ออะไร และอีกตัวก็คือ namespace ที่เป็นตัวแทนเฉพาะของฐานข้อมูลที่เราจะติดต่อกับ เช่นถ้าเป็น MS Access ก็ใช้

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

หรือถ้าเป็น MS SQL Server ก็ใช้

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

2.3.6 สร้าง Connection ไปที่ฐานข้อมูล

คำสั่งที่ใช้ในการสร้าง connection ไปที่ฐานข้อมูล ทำโดยสร้าง object ที่เป็น connection ที่เราต้องการขึ้นมา แล้วส่งค่า connection string เข้าไปเป็นค่าเริ่มต้น

```
String connectionString = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=";
connectionString += Server.MapPath("Northwind.mdb");
OleDbConnection myConnection = new OleDbConnection(connectionString);
```

จากตัวอย่างข้างต้น เราระบุให้ใช้ Provider เป็น Microsoft Jet Engine และให้ Data Source เป็นฐานข้อมูลที่มีชื่อ Northwind.mdb สำหรับคำสั่ง Server.MapPath นั้นเป็นการเติม path ไปต่อหน้าชื่อฐานข้อมูลที่เรากำหนด เพื่อให้ server หาฐานข้อมูลที่เราต้องการเจอ เช่นถ้า "c:/myweb" คือ path ผลลัพธ์ที่ได้ก็จะเป็น "c:/myweb/Northwind.mdb"

2.3.7 เปิด และ ปิด Connection

การเปิด connection ทำได้ด้วยการเรียก method ที่ชื่อ Open() ของ OleDbConnection object เช่น

```
myConnection.Open();
```

ส่วนการปิด connection ทำได้ด้วยการเรียก method ที่ชื่อ Close() ดังนี้

```
myConnection.Close();
```

การสั่งเปิด และปิด connection นั้นไม่จำเป็นต้องส่งไปที่เราต้องส่งไปในโปรแกรมของเรา อันนี้ขึ้นอยู่กับวิธีการดึงข้อมูลที่เราจะใช้ในลำดับต่อไป ถ้าเป็นการดึงข้อมูลด้วย DataReader เราจำเป็นต้องสั่งให้เปิด และปิด connection ด้วยคำสั่งดังกล่าว แต่ถ้าเป็นกรณีของ DataSet นั้นเราไม่จำเป็นต้องสั่งเปิด (แต่ถ้าจะสั่งก็ได้) เพราะตัว DataSet จะเปิด connection ให้เราเองถ้าถึงเวลา ในกรณีของการปิดก็เช่นกันเราไม่จำเป็นต้องสั่งปิด โดยจะขึ้นอยู่กับระบบ Garbage Collection ของ .Net การเปิด connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในช่วงเวลาที่สั้นที่สุดเท่าที่จะทำได้นั้น จะมีผลดีต่อความเร็วของโปรแกรมเรา เนื่องจาก connection นั้นสามารถนำไปใช้ใหม่ได้ ที่เราเรียกว่าการทำ Connection Pooling การทำ Connection Pooling นั้น .Net จะทำให้เราโดยอัตโนมัติ โดยดูที่ connection string เป็นหลัก ถ้าเป็น connection string ที่เหมือนกัน .Net จะพยายามสร้าง connection ให้น้อยที่สุด และใช้ connection ที่มีอยู่ให้เกิดประโยชน์มากที่สุด

2.3.8 สร้าง SQL Command

ขั้นตอนต่อมาหลังจากได้ connection มาแล้ว ก็คือการสร้าง Command Object สำหรับ OLE DB นั้น เราจะใช้ class ที่ชื่อ System.Data.OleDb.OleDbCommand ยกตัวอย่างเช่น

```
OleDbCommand myCommand = new OleDbCommand();
myCommand.CommandText =
"SELECT ProductID, ProductName, UnitPrice, Discontinued FROM Products WHERE UnitPrice < 15";
myCommand.Connection = myConnection;
```

ในตัวอย่างข้างต้น เราสร้าง OleDbCommand ด้วย default constructor -- หมายถึงตัวสร้าง object ของ class นั้น ๆ ที่ไม่รับค่า parameter เลย ซึ่งในการใช้งานนั้น เราสามารถเรียกใช้ได้อีกหลายแบบแล้วแต่คนที่สร้าง class นั้น ๆ จะกำหนดเอาไว้ให้ สำหรับ OleDbCommand นั้น ยังมี constructor อีก 3 ตัวที่เรียกใช้ได้คือ

1. ตัวที่รับเฉพาะ CommandText ซึ่งเป็น String
2. ตัวที่รับ CommandText กับ OleDbConnection
3. ตัวที่รับ CommandText, OleDbConnection, และ OleDbTransaction

เขียนอีกแบบได้ดังนี้

```
OleDbCommand myCommand = new OleDbCommand("SELECT ProductID, ProductName,
UnitPrice,
Discontinued FROM Products WHERE UnitPrice < 15", myConnection);
```

2.4 ความรู้เบื้องต้นเกี่ยวกับเว็บเซอร์วิส

2.4.1 แนวคิดหลักของการดำเนินธุรกิจผ่านทางเว็บเซอร์วิส คือกลยุทธ์การเพิ่มประสิทธิภาพของการดำเนินธุรกิจโดยใช้อินเทอร์เน็ต อินทราเน็ตในทุกระดับอย่างต่อเนื่อง และตอบสนองความต้องการในการเชื่อมโยงสารสนเทศของธุรกิจข้ามองค์กร ซึ่งในยุคที่สามของอินเทอร์เน็ตได้มาถึงจุดที่จะมุ่งความสนใจไปยังการยกระดับ โครงสร้างพื้นฐานและการบูรณาการระบบต่างๆเข้าด้วยกัน เพื่อให้การดำเนินธุรกิจโดยเฉพาะอย่างยิ่งทางด้านพาณิชย์อิเล็กทรอนิกส์ แบบเชื่อมโยงธุรกิจกับธุรกิจ (Business-to-Business: B2B) ได้ใช้ประโยชน์จากมาตรฐานเปิดบนอินเทอร์เน็ต และโครงสร้างพื้นฐานของอินเทอร์เน็ตให้เกิดประสิทธิภาพสูงสุดทั้งภายในองค์กรและระหว่างองค์กร เว็บเซอร์วิสเกิดขึ้นโดยมีความมุ่งหวังว่าธุรกิจที่ดำเนินการผ่านทางอินเทอร์เน็ต จะสามารถทำการติดต่อสื่อสารกันแบบอัตโนมัติ มีความยืดหยุ่น และสามารถปรับเปลี่ยนได้ทันกับการเปลี่ยนแปลงของการดำเนินธุรกิจในปัจจุบัน โดยการทำงานของเว็บเซอร์วิสจะเป็นการติดต่อสื่อสารกันระหว่างโปรแกรมกับโปรแกรม (P2P) หรือระหว่างแอปพลิเคชันกับแอปพลิเคชัน (A2A)

2.4.2 ปัจจัยพื้นฐานของเว็บเซอร์วิส

ซึ่งมีความเกี่ยวข้องกันทั้งทางด้านเทคนิค และทางด้านธุรกิจ ไว้ดังต่อไปนี้

- การรวมหรือการบูรณาการของซอฟต์แวร์ต่างระบบกันนั้น จะต้องอนุญาตให้แต่ละระบบมีความเป็นอิสระจากกัน (Loosely Coupled)
 - เมสเสจ (Message) ที่ใช้ติดต่อกันของการทำงานแบบแอปพลิเคชันกับแอปพลิเคชัน (A2A) ต้องสอดคล้องกับมาตรฐานเปิดบนอินเทอร์เน็ต
 - แอปพลิเคชันสามารถสร้างได้จากการใช้ซอฟต์แวร์คอมโพเนนต์ (Software Component) ทั้งจากภายในและภายนอกองค์กร โดยสร้างตามกระบวนการดำเนินธุรกิจหลักขององค์กร
 - มีแหล่งซอฟต์แวร์คอมโพเนนต์ (Software Component) ที่หาได้ง่าย ซึ่งช่วยเพิ่มความยืดหยุ่นในการสร้างแอปพลิเคชันที่มีคุณลักษณะเฉพาะตามกระบวนการทางธุรกิจขององค์กร
 - การนำซอฟต์แวร์จากภายนอกองค์กรกลับมาใช้ใหม่ ช่วยให้เกิดการลดต้นทุนและ/หรือช่วยปรับปรุงประสิทธิภาพในการบริหารแก่ลูกค้า
 - ซอฟต์แวร์สามารถขายเป็นบริการได้
- แนวคิดของเว็บเซอร์วิสนั้นไม่ใช่ของใหม่อย่างที่เข้าใจกันโดยทั่วไป แต่ได้มีการนำมาใช้งานในรูปแบบต่างๆ กันในกระบวนการพัฒนาระบบ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การพัฒนาโปรแกรมแบบซอฟต์แวร์คอมโพเนนต์ (Software Component) ตามแนวคิดของการออกแบบและพัฒนาโปรแกรมเชิงวัตถุ (Object-Oriented Concept)
- การออกแบบระบบแบบกระจายศูนย์ (Distributed computing) ซึ่งเป็นเป้าหมายสำคัญของการพัฒนาระบบตามสถาปัตยกรรมแบบไคลเอนต์/เซิร์ฟเวอร์
- การทำ EDI (Electronic Data Interchange) ซึ่งสร้างขึ้นโดยการกำหนดรูปแบบและมาตรฐานของข้อมูลสำหรับการทำธุรกิจ
- การบูรณาการของซอฟต์แวร์ต่างระบบ (Enterprise application integration: EAI) ที่อยู่บนพื้นฐานของความต้องการใช้ข้อมูลร่วมกัน รวมทั้งการแลกเปลี่ยนข้อมูลระหว่างแอปพลิเคชันให้สามารถทำงานที่ต้องการได้อย่างถูกต้องเหมาะสม
- รูปแบบการให้บริการซอฟต์แวร์ของ ASP (Application Service Provider)
- แนวคิดการทำเหมืองข้อมูล (Data Mining) ซึ่งต้องการนำข้อมูลที่ถูกจัดเก็บในรูปแบบที่แตกต่างกันตามแหล่งต่างๆ มาใช้งานร่วมกัน

จะเห็นว่าในทางเทคนิคแล้ว เว็บเซอร์วิสไม่ได้เป็นของใหม่แต่อย่างใด มันเกิดขึ้นจากหลักการพื้นฐานของเทคโนโลยีต่างๆ ที่ได้กล่าวไว้ในข้างต้น แต่ได้ดำเนินการปรับปรุง แก้ไขข้อจำกัดต่างๆ โดยการคิดค้นเทคโนโลยีใหม่ที่มีคุณลักษณะที่สำคัญคือ เป็นเทคโนโลยีที่เป็นมาตรฐานแบบเปิด (Opened Standard) และไม่มีใครเป็นเจ้าของ (Neural Vendor)

2.4.3 เทคโนโลยีมาตรฐาน สำหรับการพัฒนาเว็บเซอร์วิส

2.4.3.1 XML (The Extensible Markup Language 1.0) เป็นภาษา Markup ที่เป็น text-based ที่ใช้เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตในปัจจุบัน ผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) และ อย่างที่เราทราบกันดีอยู่แล้วว่า XML คือมาตรฐานในการระบุโครงสร้างข้อมูลในรูปแบบเท็กซ์ ด้วยความยืดหยุ่นที่สูงมากจึงก่อให้เกิดทั้งประโยชน์และโทษ เพราะในงานแบบเดียวกัน XML อาจจะมีฟอร์แมตได้หลายรูปแบบ การทำงานร่วมกันจึงยากขึ้น

แต่ถ้าฟัง XML อย่างเดียวก็ไม่สามารถสร้าง web services ได้ เพราะ XML Schema ทำหน้าที่เพียงอธิบายและกำหนดรูปแบบเนื้อหาของเอกสาร XML แต่ละประเภทเท่านั้น

XML ได้มีการพัฒนาเพิ่มขึ้นเป็นอย่างมาก ในยุคแรก XML จะถูกใช้งานในลักษณะของข้อมูลตามมาตรฐาน XML 1.0 เช่น WDDX (Web Distributed Data Exchange) , XML-RPC ต่อมาในยุค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สองได้มีการเพิ่มความสามารถของ XML ในเรื่องของ XML Namespaces และ Schema ซึ่ง SOAP จัดว่าอยู่ในยุคที่สองของ XML โพรโทคอล

ในยุคที่สองของ XML นั้นได้กำจัดจุดอ่อนของยุคแรกในสองข้อหลักคือ

1. ในยุคแรกจะมีจุดอ่อนในด้านการเปลี่ยนแปลงโครงสร้างเอกสาร เช่นใน XML-RPC หรือ WDDX ถ้าต้องการเพิ่มให้โปรโตคอลรองรับข้อมูลใหม่ เช่น ข้อมูลไบนารี ทั้ง XML-RPC และ WDDX จะต้องทำการอัปเดตมาตรฐานของตนใหม่ ทำให้ผู้ที่เขียนโปรแกรมในภาษาต่างๆ จะต้องมีการเปลี่ยนแปลงตามไปด้วยซึ่งทำให้การเปลี่ยน Version ของมาตรฐานไม่คล่องตัว แต่ในยุคที่สองของ XML โพรโทคอล จะใช้ Namespaces เข้ามาช่วยในการแก้ปัญหา

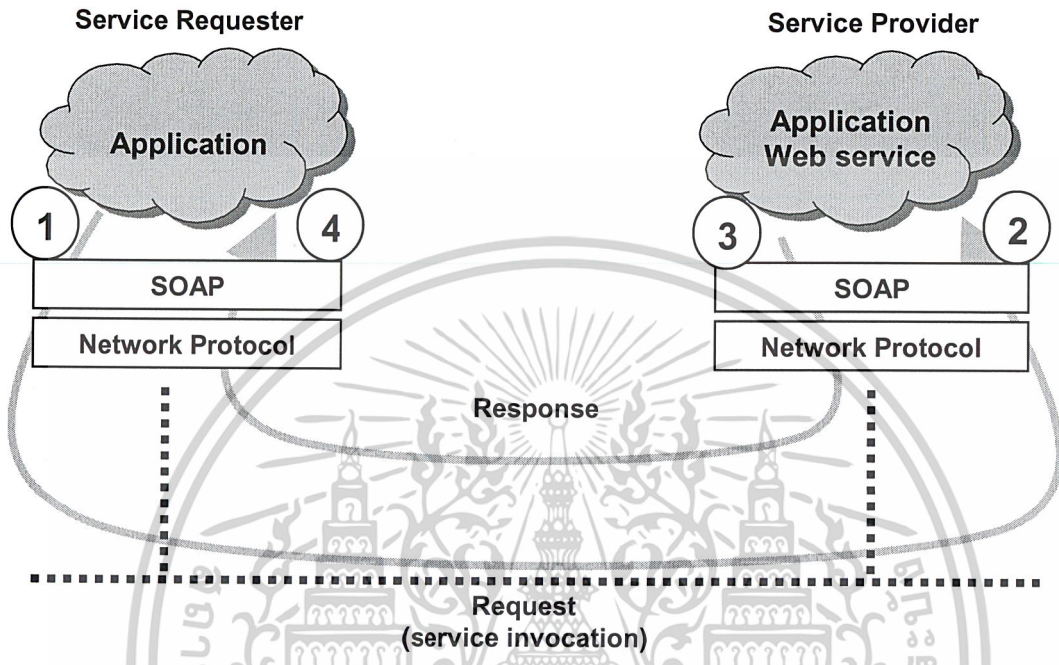
2. เรื่องที่สองคือการทำงานกับข้อมูลที่มีหลายชนิด (Data type) เนื่องจากในยุคแรกจะใช้ Document Type Definition (DTD) อธิบายโครงสร้างของเอกสาร XML ซึ่ง DTD จะไม่มีส่วนของการอธิบายความหมายของชนิดข้อมูล แต่ในยุคที่สองจะใช้ XML Schema แทนซึ่งจะมีการอธิบายทั้งรูปแบบโครงสร้างของลำดับ (syntax) และข้อมูลของชนิดของข้อมูลด้วย

2.4.3.2 SOAP (Simple Object Access Protocol) เป็น XML-based โพรโทคอล

เป็นเมสเสจจิงโพรโทคอล (Messaging Protocol) สำหรับใช้ในการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (decentralized, distributed environment) SOAP ได้กำหนดเมสเสจจิงโพรโทคอลระหว่างผู้ขอบริการ (requestor) กับผู้ให้บริการ (provider) ในการติดต่อสื่อสารกัน เช่น กำหนดให้ผู้ขอบริการต้องส่งข้อมูล เช่น ข้อมูลที่ระบุฟังก์ชัน และค่าพารามิเตอร์ต่างๆ ที่ต้องใช้ในแอปพลิเคชันที่ร้องขอ ส่งไปให้กับผู้ให้บริการ ซึ่งแอปพลิเคชันของผู้ให้บริการอาจจะทำงานโดยใช้ RMI (Remote Method Invocation) ตามวิธีการของโปรแกรมแบบออบเจกต์ เป็นต้น ในช่วงเริ่มต้น บริษัท ไมโครซอฟท์, ไอบีเอ็ม, ล็อตัส, ยูสเซอร์แลนด์ (UserLand) และ ดีเวลลอปเปอร์เมนเตอร์ (DeveloperMenter) ได้ร่วมกันกำหนดมาตรฐานของ SOAP ขึ้น ซึ่งต่อมาได้มีบริษัทอีก 30 กว่าบริษัท เข้าร่วมและจัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น SOAP ได้กำหนดรูปแบบพื้นฐานของการสื่อสารแบบกระจายขึ้นโดยใช้ SOA เป็นสถาปัตยกรรมพื้นฐาน ซึ่งในปัจจุบัน SOAP ได้ถูกกำหนดให้เป็น Services-Oriented Architecture Protocol เรียบร้อยแล้ว เนื่องจาก SOAP ได้ถูกนำมาใช้งานอย่างแพร่หลายแล้วนั่นเอง แม้ว่า SOAP จะไม่ได้กำหนดการทำเมสเสจจิงโพรโทคอลก็ตาม จุดเด่นของ SOAP ก็คือเป็นโพรโทคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของและเป็นโพรโทคอลที่สามารถทำงานกับโพรโทคอลอื่นหลายชนิด การพัฒนาก็อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการ ตามแบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา ตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ SOAP ได้แสดงไว้ในภาพด้านล่าง (รายละเอียดเพิ่มเติมเกี่ยวกับ SOAP สามารถศึกษาเพิ่มเติมได้ที่ www.w3.org/TR/SOAP)



รูปที่ 2.2 ตัวอย่างการทำงานของ SOAP

จากรูปที่ 2.4 แสดงลำดับขั้นตอนการทำงานของ soap คือ

1. แอปพลิเคชันของผู้ขอบริการสร้าง SOAP message เพื่อเรียกใช้บริการของเว็บเซอร์วิส (Web Services)
2. เว็บเซอร์วิส (Web Services) ของผู้ให้บริการ ผู้ให้บริการได้รับ SOAP message จากผู้ร้องขอ ซึ่งอยู่ในรูปแบบ XML
3. เว็บเซอร์วิส (Web Services) ประมวลผลตามคอมพิวเตอร์ที่ให้บริการ เว็บเซอร์วิสส่งผลลัพธ์มาแล้วผู้ให้บริการก็จะสร้าง SOAP message ที่มีผลลัพธ์นั้นส่งกลับมายังผู้ร้องขอบริการ
4. แอปพลิเคชันของผู้ขอบริการได้รับผลลัพธ์ที่เป็น SOAP message แล้วทำการแปลงให้อยู่ในรูปแบบที่ต้องการ เพื่อนำไปประมวลผลต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส เราต้องการให้แอปพลิเคชัน มีการทำงานกับแอปพลิเคชันที่ทำงานอยู่ในเครื่องอื่นโดยผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่าง Object ในระยะไกล (Remote Procedure Calls : RPC) เช่น DCOM,EJB หรือ CORBA นั้นไม่ได้ถูกออกแบบมาใช้สำหรับโปรโตคอล HTTP (เครือข่ายที่ปัจจุบันใช้งานแพร่หลาย และสะดวกมากที่สุดได้แก่เครือข่าย Internet ซึ่งให้การสื่อสารผ่านโปรโตคอล HTTP เป็นส่วนใหญ่)

ทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำบริการเว็บเซอร์วิส ก็คือให้ทำงานอยู่บนโปรโตคอล HTTP ไปเลย โดย SOAP จะอาศัย XML Schema ในการอธิบายองค์ประกอบทั้ง 3 ส่วนดังนี้

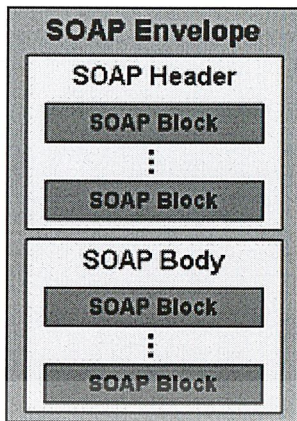
1. Envelopที่จะระบุว่า “ส่งข้อมูลอะไร” และ “ส่งไปให้ใคร” พร้อมกับบอกว่าเป็นส่วนสำคัญหรือตัวเล็ก
2. Encoding Rule คือระเบียบวิธีการในการแลกเปลี่ยนข้อมูลแต่ละประเภทในแอปพลิเคชันต่างๆ กัน
3. RPC (Remote Procedure Call) ระบุกฎสำหรับเรียกใช้ฟังก์ชันระยะไกลและMethod call รวมไปถึงการตอบรับผ่านอินเทอร์เน็ต

ซึ่ง SOAP นอกจากจะทำงานบนโปรโตคอล HTTP แล้วยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ, เทคโนโลยีรวมไปถึงภาษาที่ใช้ในการพัฒนาด้วยก็ได้

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

1. SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด
2. SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือไม่มีก็ได้
3. SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้จากเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โครงสร้างของเอกสาร SOAP

อย่างไรก็ดี SOAP ก็ยังมีข้อเสียอยู่บ้าง ได้แก่

- เนื่องจาก SOAP message นั้นเก็บอยู่ในรูปแบบ XML ทำให้เสียเวลาในการแปลเอกสาร XML กลับมาเป็นรูปแบบที่โปรแกรมเราเข้าใจ
- SOAP ทำงานอยู่กับ HTTP Protocol ซึ่งมีความเร็วในการรับและส่งข้อมูลค่อนข้างต่ำ ด้วยเหตุนี้ทำให้ความเร็วของ SOAP อยู่ในระดับเดียวกับ HTTP ในขณะที่ Protocol อื่น เช่น FTP มีความเร็วสูงกว่า

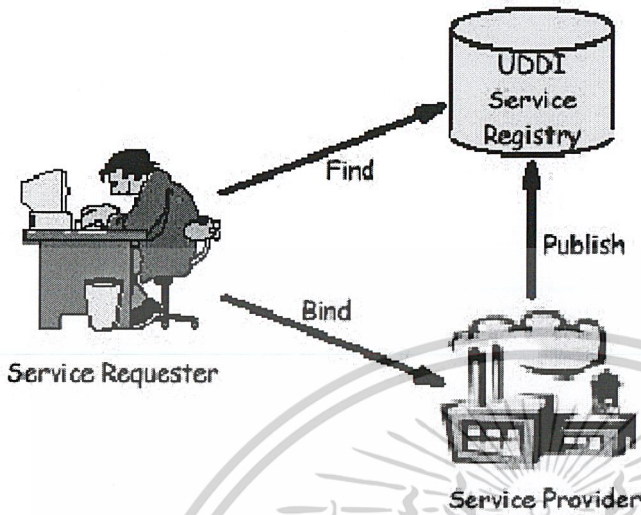
2.4.3.3 WSDL (Web Services Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของเว็บเซอร์วิส และวิธีการติดต่อขอรับบริการจากเว็บเซอร์วิส โดย WSDL จะบอกเซิร์ฟเวอร์อื่น ๆ ว่า จะจัดรูปแบบของรีเควสต์ (Request) และการตอบรับ (Response) มายังเว็บเซอร์วิสอย่างไร นอกจากนี้ยังระบุการเชื่อมโยง (Bind) ตัวเซอร์วิสมาที่โปรโตคอลอีกด้วย (ปกติคือ HTTP) WSDL สร้างขึ้นโดยใช้ภาษา XML WSDL เกิดจากการรวมแนวคิดของ NASSL (The Network Accessible Service Specification Language) และ WDS (Well-Defined Services) ของบริษัทไอบีเอ็ม กับ SDL (The Service Description Language) และ SCL (the SOAP Contract Language) ของบริษัทไมโครซอฟท์ ปัจจุบัน WSDL เป็นภาษาที่อยู่ในการดูแลของ W3C (World Wide Web Consortium) ซึ่งยังไม่เป็นมาตรฐานที่สมบูรณ์ เวอร์ชันที่ใช้งานอยู่ในปัจจุบันคือ WSDL 1.1 (รายละเอียดเพิ่มเติมเกี่ยวกับ WSDL สามารถศึกษาเพิ่มเติมได้ที่ <http://www.w3c.org/TR/wsdl>)

2.4.3.4 UDDI (Universal Description, Discovery and Integration) เป็นมาตรฐานที่ให้ชุด
พื้น

ฐาน APIs (Application Programming Interface) ของ SOAP สามารถนำมาใช้ในการพัฒนาเพื่อเป็นตัวแทนของผู้ให้บริการ (Service broker) UDDI เป็นมาตรฐานที่จัดตั้งขึ้นโดยบริษัท ไอบีเอ็ม บริษัทไมโครซอฟท์ และบริษัทอริบา (Ariba) ปัจจุบันมีบริษัทที่ร่วมกันกำหนดมาตรฐานของ UDDI มากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

70 บริษัท ซึ่งมาตรฐานของ UDDI ถูกกำหนดให้เป็นมาตรฐานสำหรับ B2B interoperability (รายละเอียดเพิ่มเติมเกี่ยวกับ UDDI สามารถศึกษาเพิ่มเติมได้ที่ <http://uddi.org>)

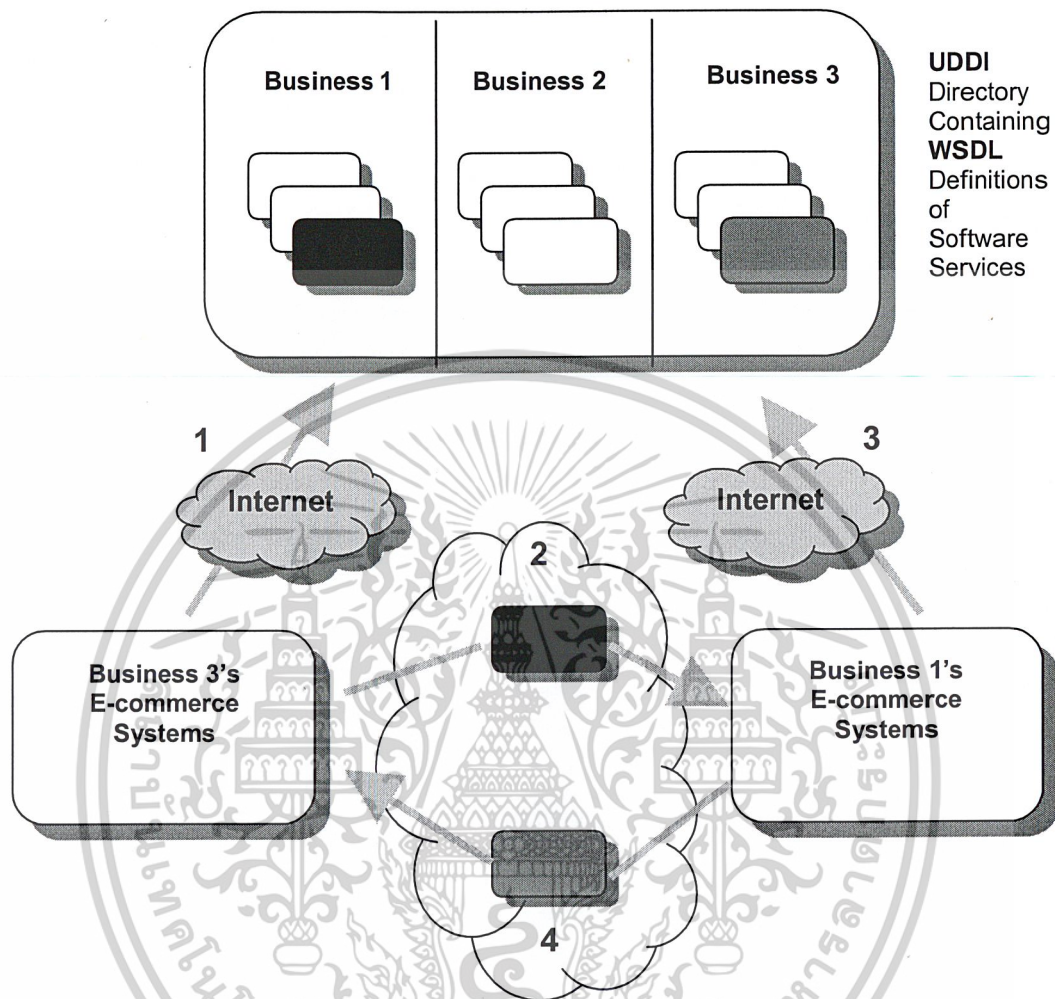


รูปที่ 2.4 ภาพแสดงความสัมพันธ์ระหว่างผู้ขอบริการ (Service Requester) กับผู้ให้บริการ (Service Provider) และ UDDI

2.4.4 สรุปความรู้เบื้องต้นเกี่ยวกับ Web Services

กล่าวโดยสรุปคือ การให้บริการทางด้านซอฟต์แวร์ต่างๆ ของผู้ให้บริการ (Service providers) บนอินเทอร์เน็ตนี้ รู้จักกันในนามของ “เว็บเซอร์วิส (Web Service)” กล่าวคือ เว็บเซอร์วิส คือ ซอฟต์แวร์คอมโพเนนต์ (Software Component) ที่ผู้ให้บริการสามารถนำมาสร้างเป็นแอปพลิเคชันสำหรับให้บริการการทำงานๆ หนึ่งให้แก่ผู้ขอบริการบนอินเทอร์เน็ต หรือผู้ขอบริการสามารถที่จะขอบริการจากผู้ให้บริการเว็บเซอร์วิสจากหลายๆ ที่ นำมาประกอบเข้าด้วยกันตามกระบวนการทางธุรกิจ เพื่อร่วมกันทำงานในลักษณะ Interoperability ซึ่งสามารถจำลองภาพการทำงานของ Web Services ที่เกิดจากการทำงานร่วมกันของ SOAP, UDDI และ WSDL ในดังภาพต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ภาพการทำงานของ Web Services ที่เกิดจากการทำงานร่วมกันของ SOAP, UDD และ WSDL

พนักงาน ไอทีของ Business 3:

1. ค้นหาข้อกำหนดการให้บริการใน WSDL ผ่านอินเทอร์เน็ตของ Business 1
2. ติดต่อผ่านอินเทอร์เน็ตของแอปพลิเคชันที่ให้บริการ (ด้วย SOAP) ผ่านทางอินเทอร์เน็ต
- 3,4. Business 1 รวมหรือบูรณาการกับ Business 3 ด้วยวิธีเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนและวิธีดำเนินการวิจัย

วิธีดำเนินการวิจัยปัญหาพิเศษ เรื่องการสาธิตการใช้งานเว็บเซอร์วิส (WEB SERVICES DEMONSTRATION) ด้วยระบบตัวแทนจำหน่ายตัวชมภาพยนตร์ มีขั้นตอนกิจกรรมการดำเนินงาน ซึ่งแบ่งเป็น 3 ขั้นตอน คือ การวิเคราะห์และออกแบบระบบ การดำเนินงานและ ทดสอบการทำงาน

3.1 ขั้นตอนการวิเคราะห์ และ ออกแบบระบบ (Specification Analysis)

3.1.1 Domainระบบตัวแทนจำหน่ายตัวชมภาพยนตร์ มีการทำงานหลักๆ คือ

-ให้ข้อมูลภาพยนตร์

โดยระบบจะทำการเรียกใช้เว็บเซอร์วิสเพื่อสอบถามข้อมูลของภาพยนตร์จากโรงภาพยนตร์ เพื่อให้ลูกค้าสอบถามผ่านทางเว็บไซต์ (website) ซึ่งจะมีทั้งข้อมูลของภาพยนตร์ทั้งหมดที่ฉาย ข้อมูลของภาพยนตร์แต่ละเรื่อง ข้อมูลของภาพยนตร์ทั้งหมดที่กำลังจะฉาย และโปรมืออื่นต่างๆ

รวมทั้งสามารถสอบถามข้อมูลภาพยนตร์ทุกเรื่องที่มีในระบบได้ด้วย โดยระบบจะทำการเรียกใช้เว็บเซอร์วิสของทุกโรงภาพยนตร์มาเก็บที่ฐานข้อมูล และจะทำการquery แล้วแสดงผลเมื่อลูกค้าเรียกใช้

-ให้รายละเอียดในการจองซื้อตัวชมภาพยนตร์

โดยระบบจะทำการแสดงข้อมูล โรงภาพยนตร์ รูปแบบเก้าอี้ ภาพยนตร์ และรอบ ของภาพยนตร์ทั้งหมดจากโรงภาพยนตร์ทุกโรงเพื่อให้ลูกค้าเลือก และเมื่อลูกค้าเลือกรอบระบบจะทำการเรียกใช้เว็บเซอร์วิส ขอแผนผังที่นั่งจากโรงภาพยนตร์ ซึ่งจะมีข้อมูลการซื้อ ,จองที่นั่งบางส่วนแล้วณ. ขณะนั้น

-บริการจองซื้อตัวชมภาพยนตร์

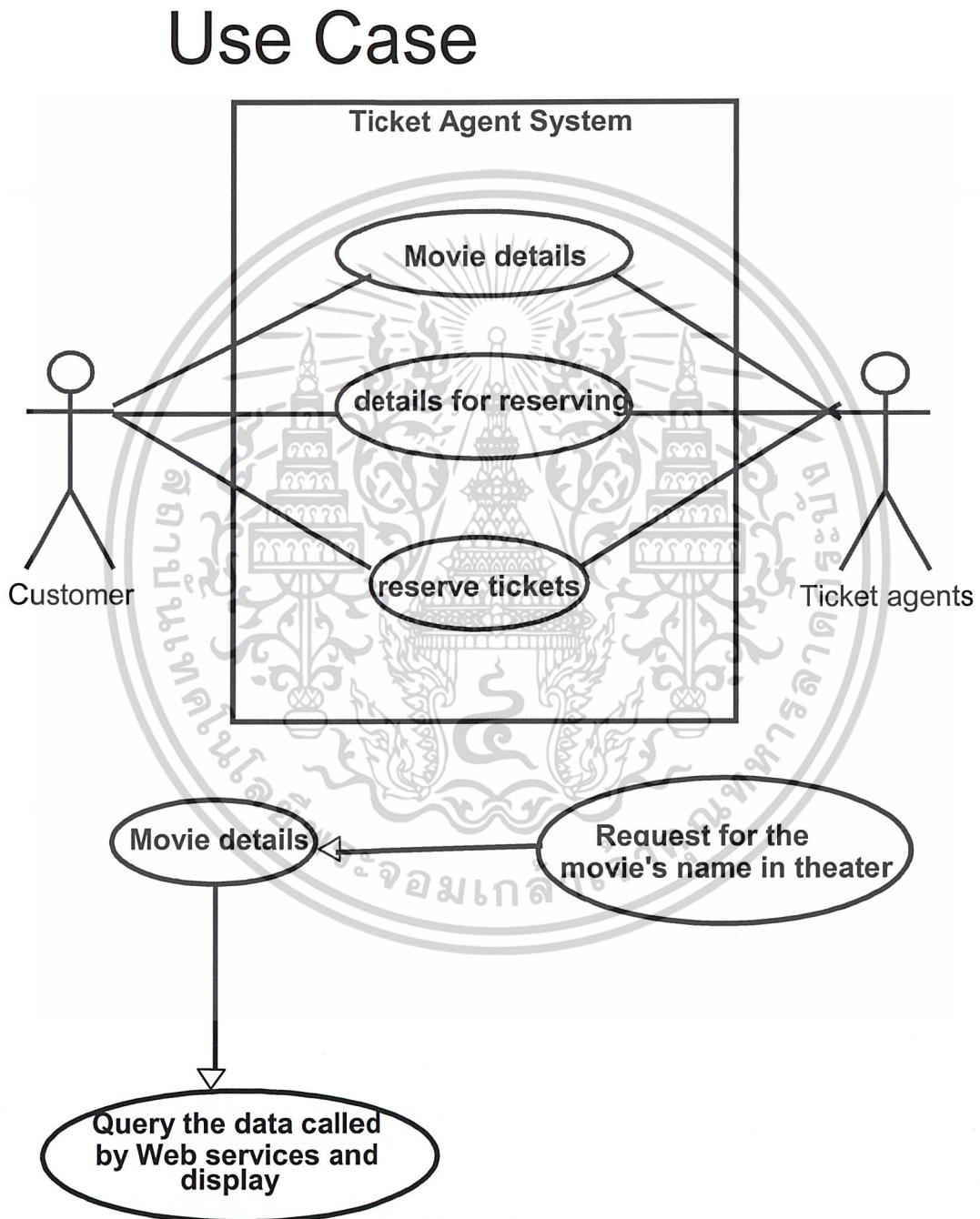
โดยเมื่อลูกค้าเลือกที่นั่งจากแผนผังที่แสดงผ่านทางเว็บไซต์ของระบบแล้ว ระบบจะทำการเรียกใช้เว็บเซอร์วิสเพื่อจองที่นั่งโดยส่งชุดของหมายเลขที่นั่งที่ลูกค้าเลือกไป

โรงภาพยนตร์จะทำการตรวจสอบข้อมูลการจองที่ระบบส่งมา แล้วตอบรับ (accept) หรือยกเลิกการจอง (reject) ซึ่งจะส่งแผนผังที่นั่งมาupdateข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

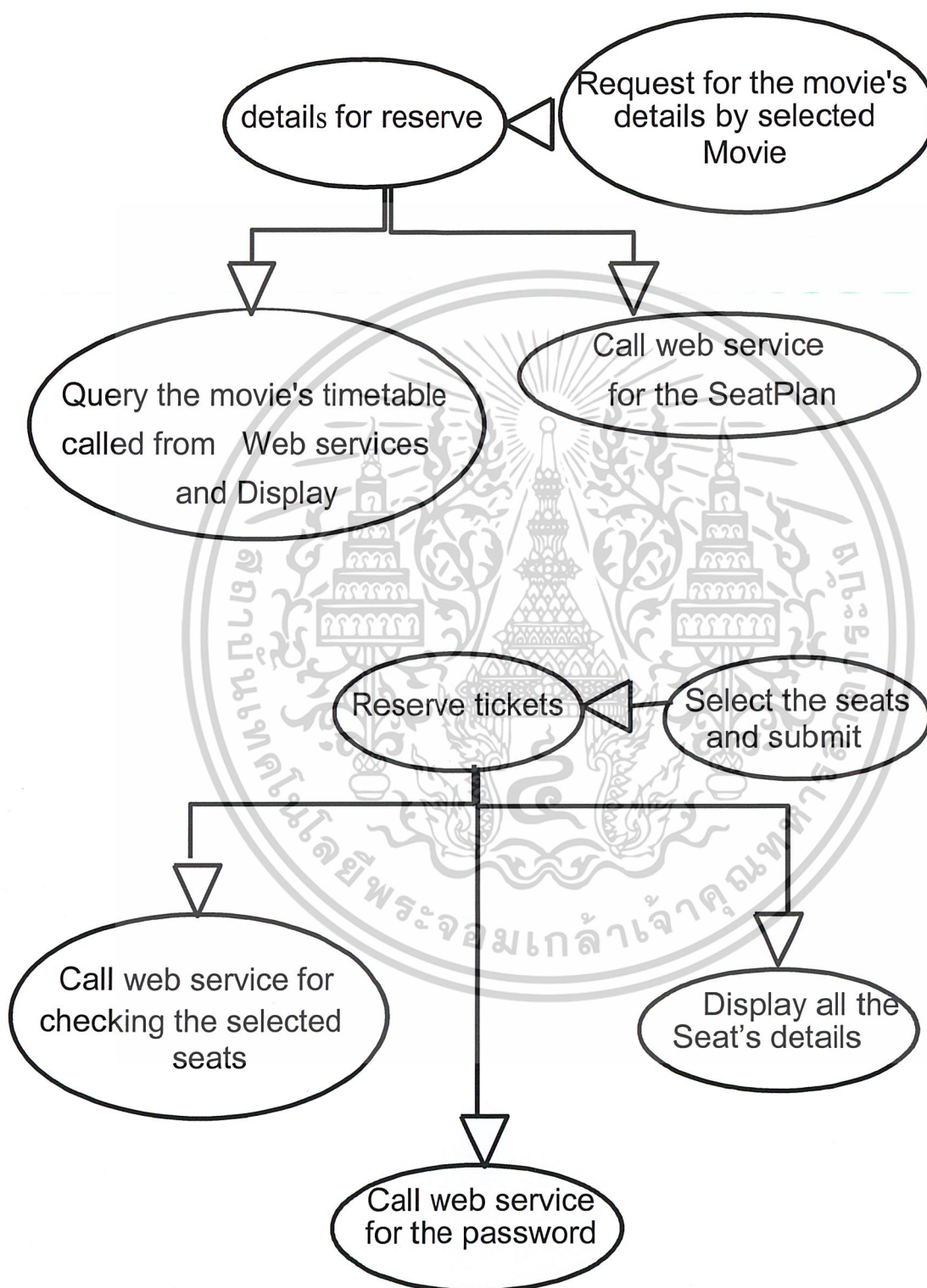
จากนั้นระบบจะทำการรายงานการตอบรับจากของภาพยนตร์ให้ลูกค้าทราบ โดยโรงภาพยนตร์จะทำการ Generate password กลับมานำไปใช้รับ ,ชื่อตัวชมภาพยนตร์ที่โรงภาพยนตร์ หลังจากนั้นระบบจะทำการสรุปข้อมูลการจองทั้งหมด แสดงบนหน้าจอ

3.1.2 USE CASE ของระบบตัวแทนจำหน่ายตั๋วชมภาพยนตร์



รูปที่ 3.1 USE CASE ของระบบตัวแทนจำหน่ายตั๋วชมภาพยนตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

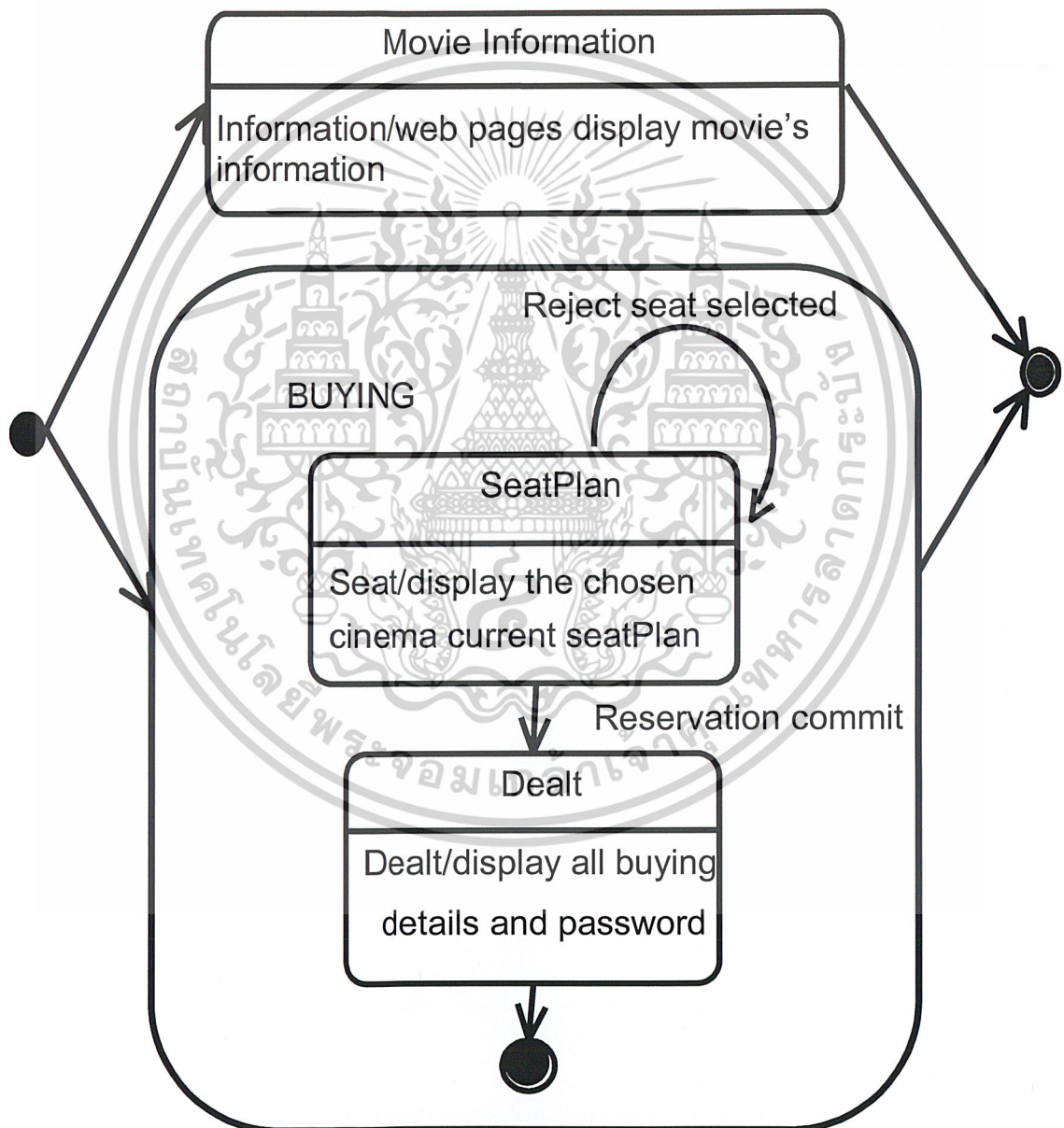


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 USE CASE ของระบบตัวแทนจำหน่ายตั๋วชมภาพยนตร์(ต่อ)

3.1.3 Statechart Diagram ของระบบตัวแทนจำหน่ายตั๋วชมภาพยนตร์

Statechart Diagram : Ticket agent System

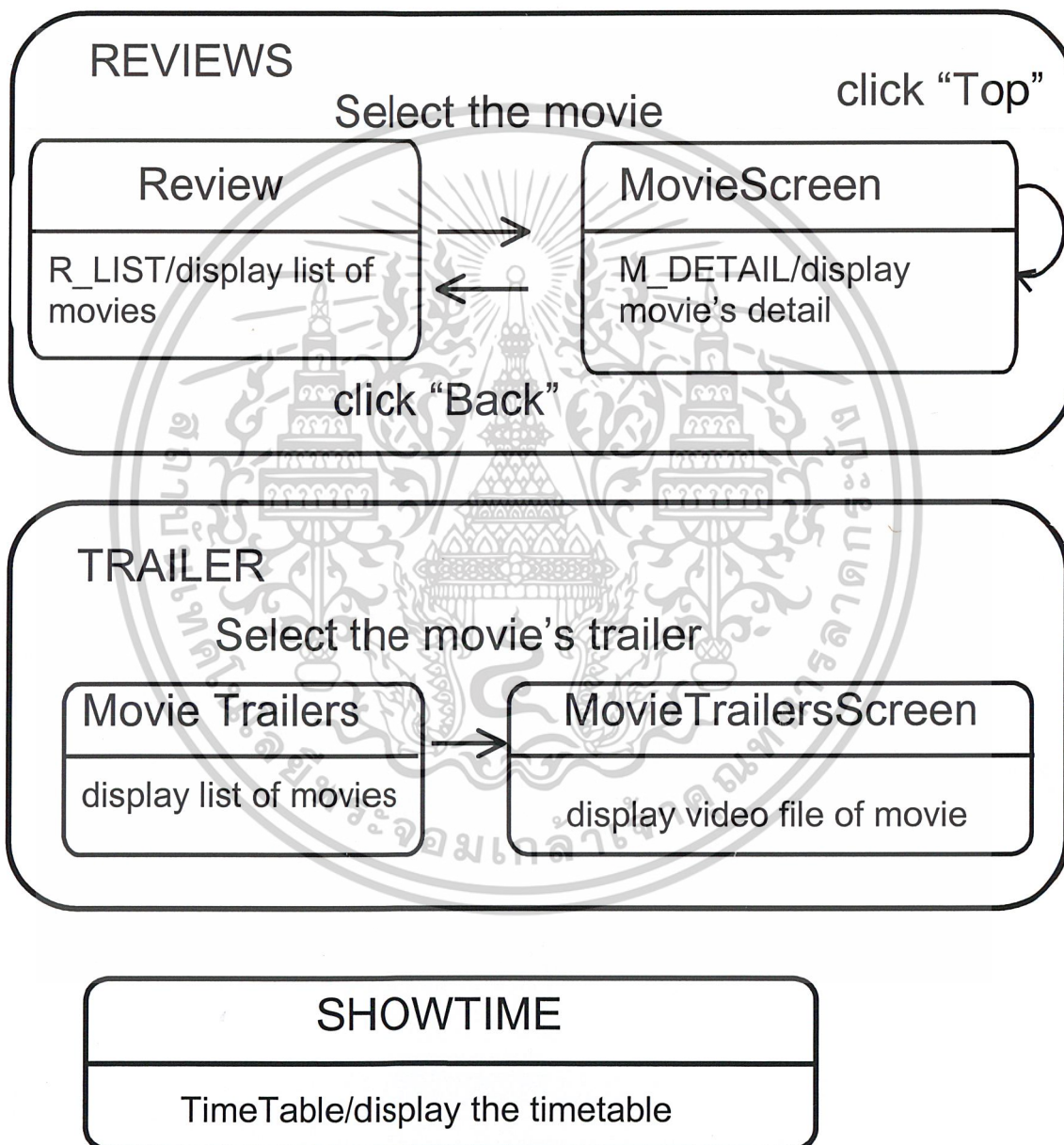


รูปที่ 3.3 Statechart Diagram ของระบบตัวแทนจำหน่ายตั๋วชมภาพยนตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Statechart Diagram : Extends Movie Information

INFORMATION

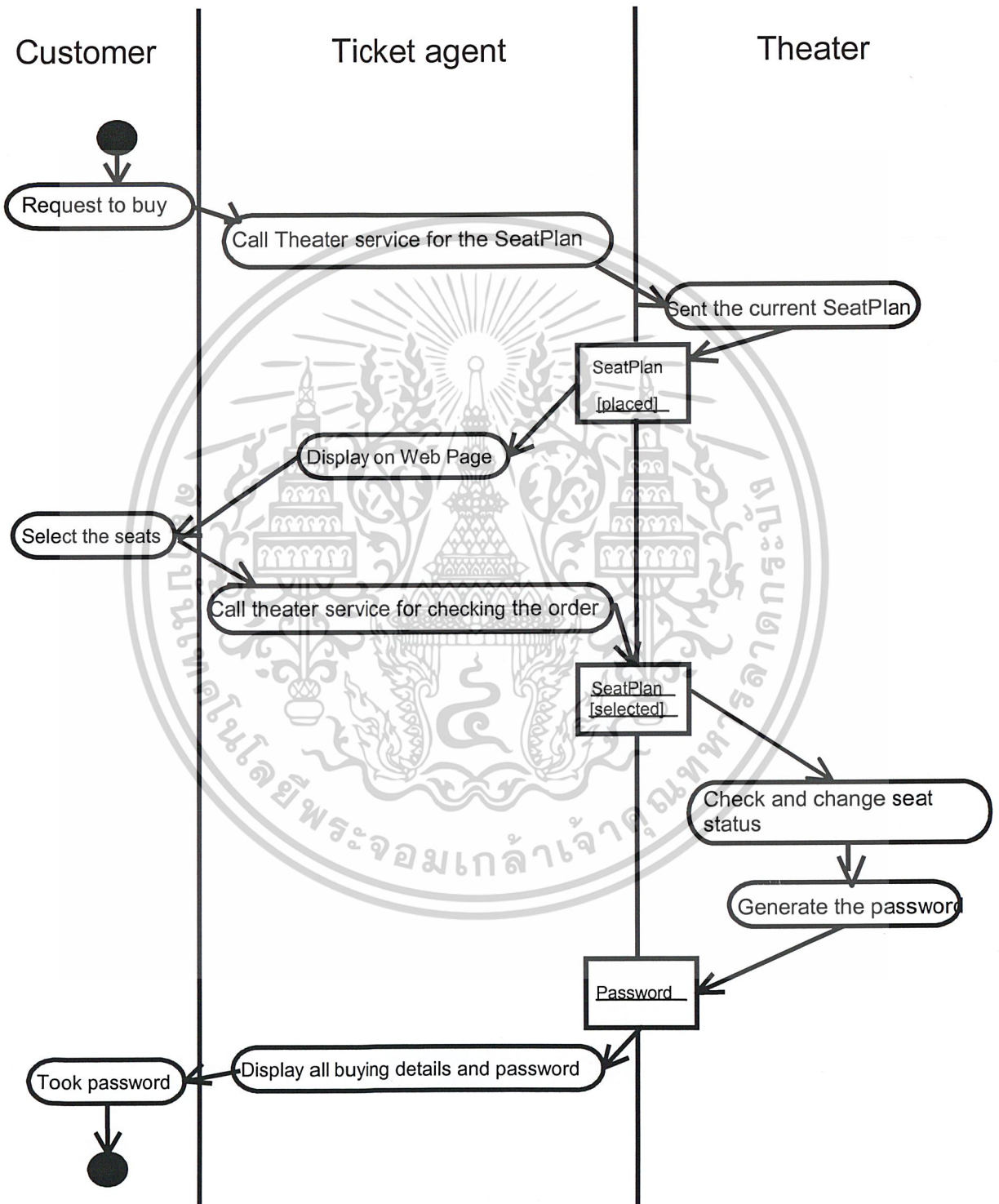


รูปที่ 3.4 Statechart Diagram ของระบบตัวแทนจำหน่ายตัวชมภาพยนตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 Activity Diagram ของระบบตัวแทนจำหน่ายตั๋วชมภาพยนตร์

Activity Diagram : Reserve the ticket



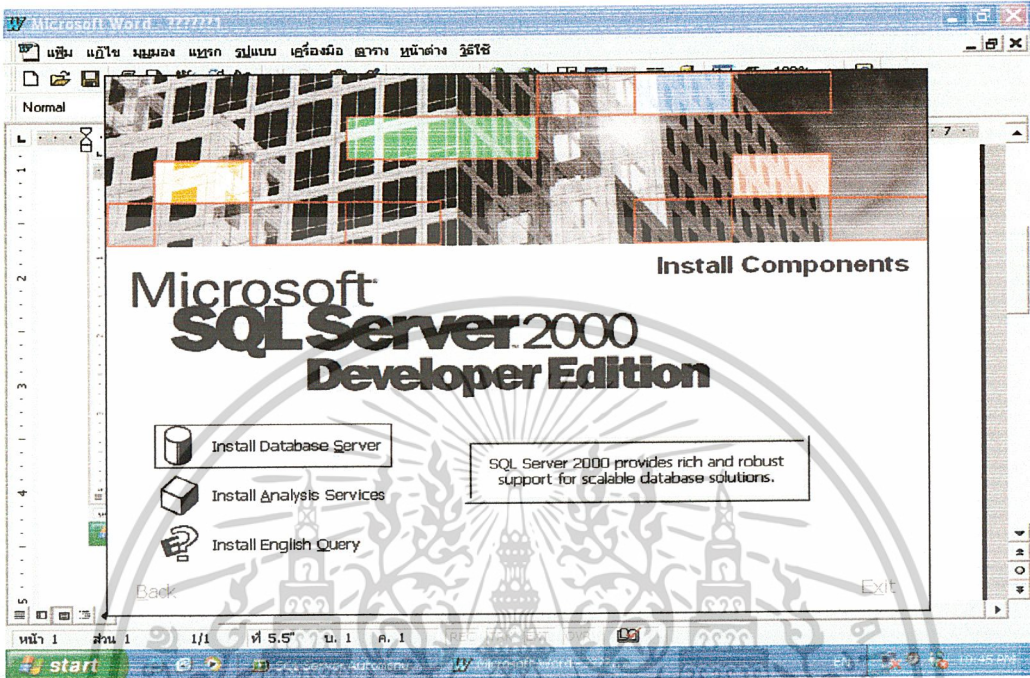
รูปที่ 3.5 Activity Diagram ของระบบตัวแทนจำหน่ายตั๋วชมภาพยนตร์เมื่อทำการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการดำเนินงาน (Implementing)

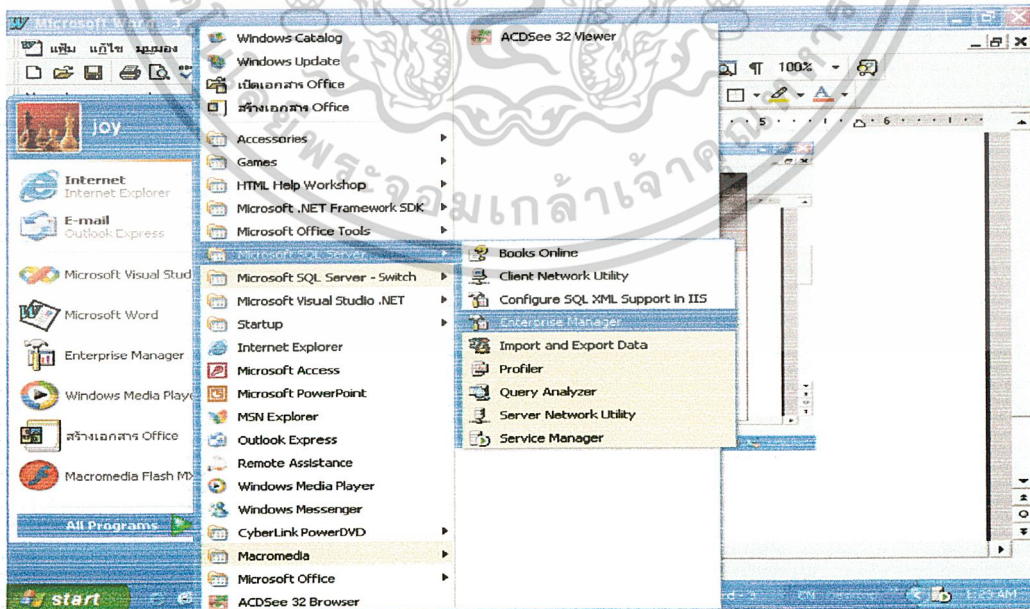
3.2.1 ขั้นตอนการสร้างฐานข้อมูลของโรงพยาบาลนคร และ ระบบตัวแทนจำหน่ายตัวหมวกภาพยนตร์

1. ทำการติดตั้งโปรแกรม Microsoft SQLServer2000 (Developer Edition)



รูปที่ 3.5 การสร้างฐานข้อมูลด้วยโปรแกรม Microsoft SQLServer2000

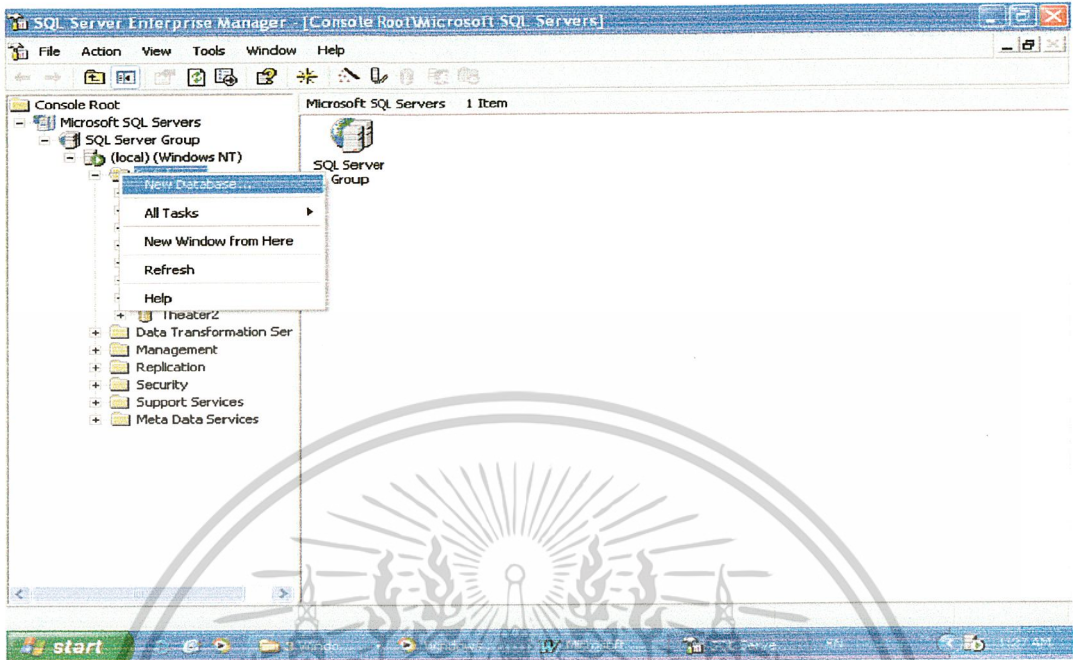
2. เปิดโปรแกรม All Program → Microsoft SQL Server → Enterprise Manager



รูปที่ 3.6 การสร้างฐานข้อมูลด้วยโปรแกรม Microsoft SQLServer2000(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สร้างฐานข้อมูล โดยclickขวาที่ Database แล้วเลือก New Database



รูปที่ 3.7 การสร้างฐานข้อมูลด้วยโปรแกรม Microsoft SQLServer2000(ต่อ)

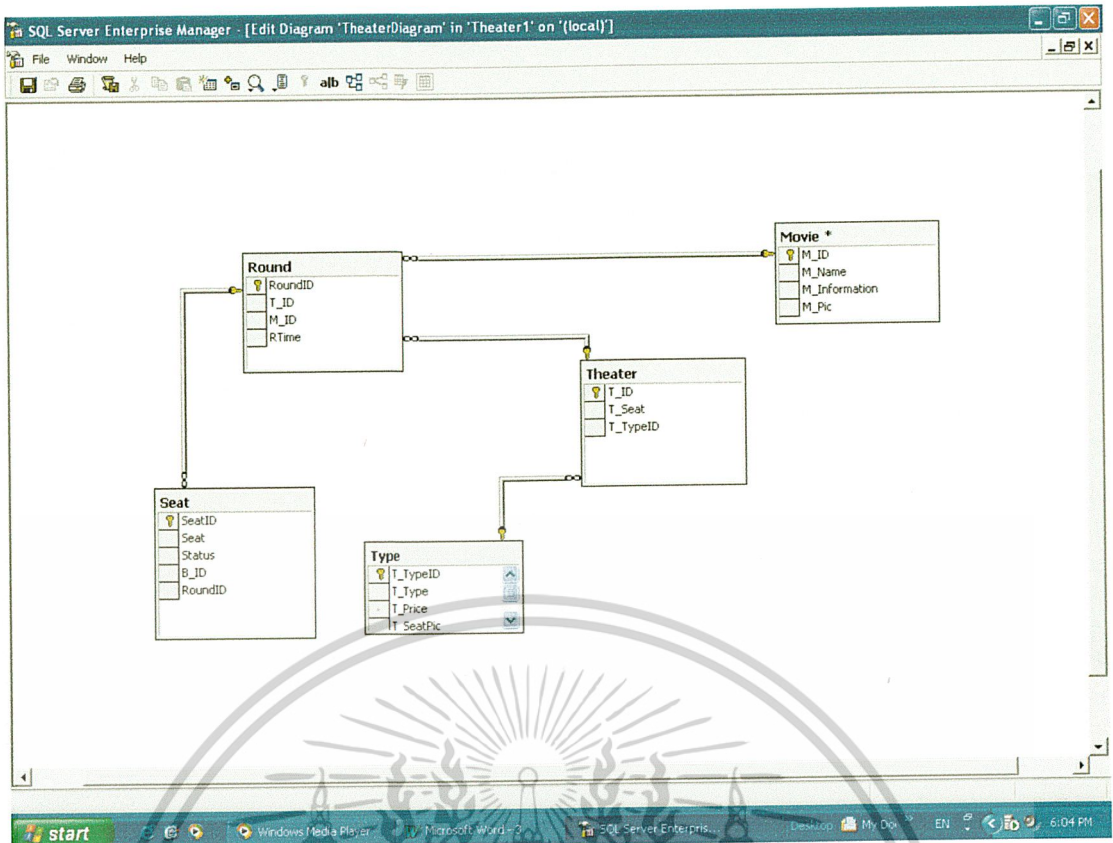
4. สร้างฐานข้อมูลของโรงภาพยนตร์ที่1 (Theater1) โดยทำการสร้างTable และ Relation ระหว่างแต่ละtable ซึ่งแต่ละTableมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 รายละเอียดของฐานข้อมูลของโรงภาพยนตร์ที่ 1 (Theater1)

Column Name	Data Type
Theater	
T_ID	Char
T_Seat	Char
T_TypeID	integer
Type	
T_TypeID	Integer
T_Type	Char
T_Price	Integer
T_SeatPic	VarChar
Seat	
SeatID	Char
Seat	Char
Status	Char
B_ID	Char
RoundID	Char
Round	
RoundID	Char
T_ID	Char
M_ID	Char
RTime	DateTime
Movie	
M_ID	Char
M_Name	Char
M_Pic	Varchar
M_Informatuon	Varchar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Table และ Relationของฐานข้อมูลโรงภาพยนตร์ที่1 (Theater1)

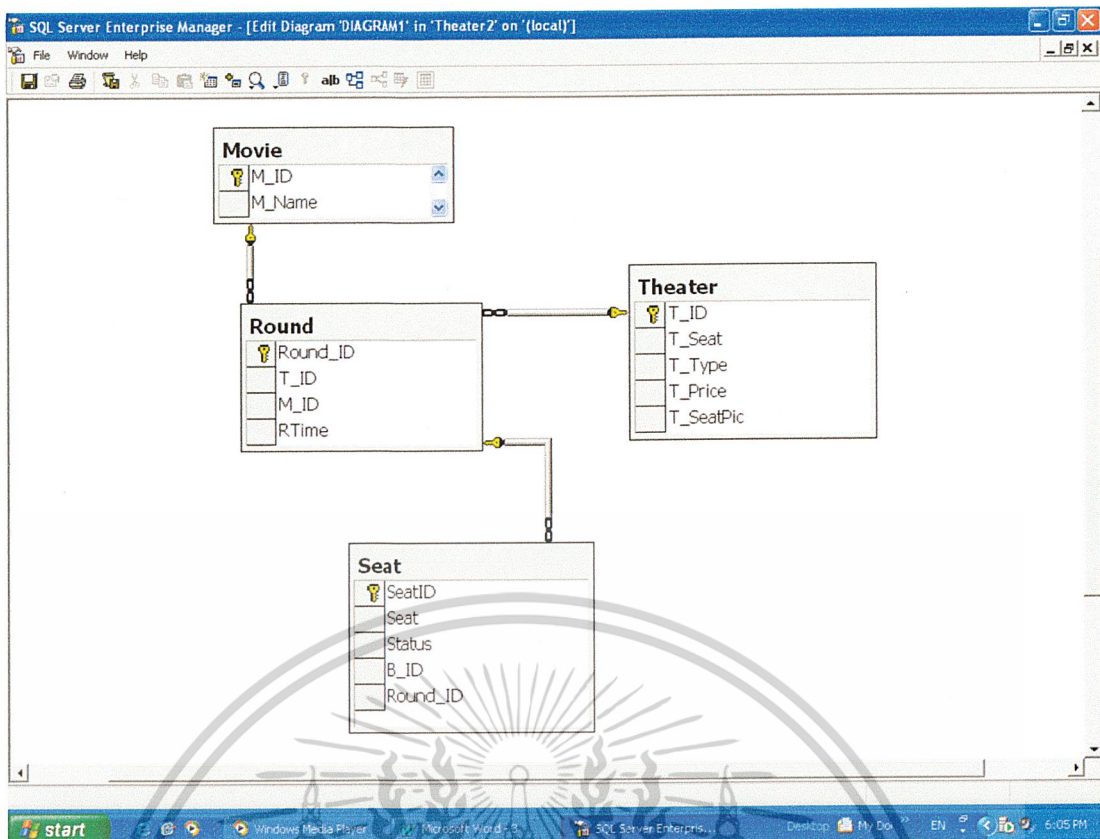
5. สร้างฐานข้อมูลของโรงภาพยนตร์ที่2 (Theater2) โดยทำการสร้างTable และ Relation ระหว่างแต่ละtable ซึ่งแต่ละTableมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 รายละเอียดของฐานข้อมูลของโรงภาพยนตร์ที่ 2 (Theater2)

Column Name	Data Type
Theater	
T_ID	Char
T_Seat	Char
T_Type	Char
T_Price	Integer
T_SeatPic	VarChar
Seat	
SeatID	Integer
Seat	Char
Status	Char
B_ID	Char
RoundID	Char
Round	
RoundID	Char
T_ID	Char
M_ID	Char
RTime	Data Time
Movie	
M_ID	Char
M_Name	Char
M_Pic	Varchar
M_Informatuon	Varchar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 Table และ Relationของฐานข้อมูลโรงภาพยนตร์ที่2 (Theater2)

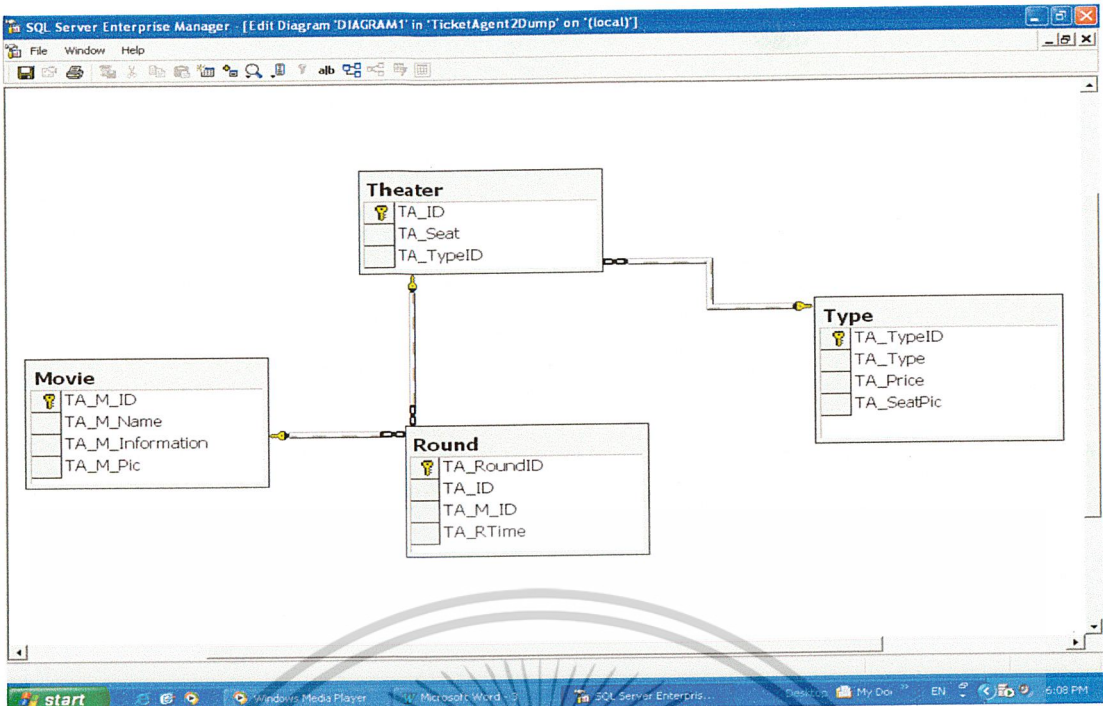
6. สร้างฐานข้อมูลของตัวแทนจำหน่ายตั๋วชมภาพยนตร์(Ticket agent) โดยทำการสร้าง Table และ Relation ระหว่างแต่ละtable ซึ่งแต่ละTableมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 รายละเอียดของฐานข้อมูลของคิวแทนจำหน่ายตั๋วชมภาพยนตร์

Column Name	Data Type
Theater	
TA_ID	Integer
TA_Seat	Char
TA_TypeID	Integer
Type	
TA_TypeID	Integer
TA_Type	Char
T_Price	Integer
TA_SeatPic	VarChar
Round	
TA_RoundID	Integer
TA_ID	Integer
TA_M_ID	Integer
TA_RTime	Data Time
Movie	
TA_M_ID	Integer
TA_M_Name	Varchar
TA_M_Pic	Varchar
TA_M_Informatuon	Varchar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Table และ Relation ของฐานข้อมูลตัวแทนจำหน่ายตั๋วชมภาพยนตร์ (Ticket agent)

3.2.2 ขั้นตอนการสร้าง Web services ของโรงภาพยนตร์ โดยจะทำการimplementด้วย C#.NET สำหรับ Theater1 และ VB.NET สำหรับ Theater2 ซึ่งจะเห็นว่าการใช้หลักการของ Web services นั้นเป็นอิสระต่อภาษาที่ใช้พัฒนา

มี Web Service ที่จำเป็นอยู่ 4 web services คือ

1. Service ในการให้ข้อมูลของภาพยนตร์ และ รอบในแต่ละวัน
2. Service ในการให้ข้อมูลจำนวนที่นั่งในโรง ของรอบภาพยนตร์ที่เลือก

```

<WebMethod(> Public Function SeatedNow(ByVal R_id As String) As DataSet
    Dim strconn As String = "initial catalog=Theater2;persist security info=False;"

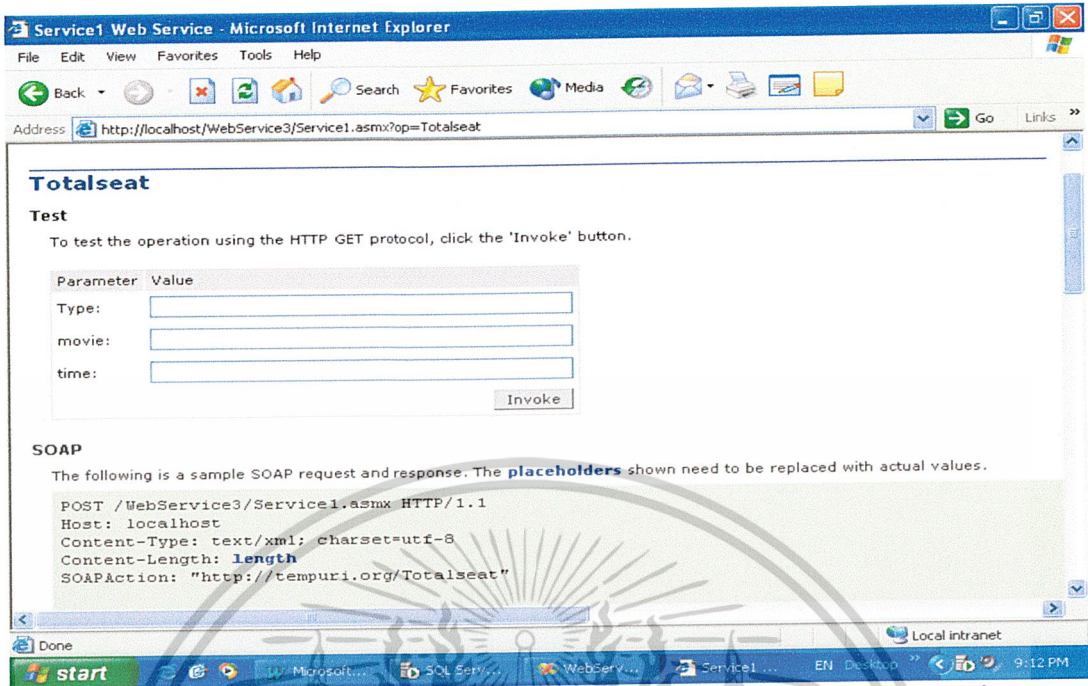
    Dim sql1 As String = "SELECT Seat,Status FROM Seat WHERE Round_ID = " & R_id & ";"
    Dim conn As SqlConnection = New SqlConnection()
    With conn
        .ConnectionString = strconn
        .Open()
    End With

    Dim da As SqlDataAdapter = New SqlDataAdapter(sql1, conn)
    Dim ds1 As DataSet = New DataSet()
    'da.Fill(ds1, "temp")
    da.Fill(DataSet1, "temp")
    SeatedNow = DataSet1

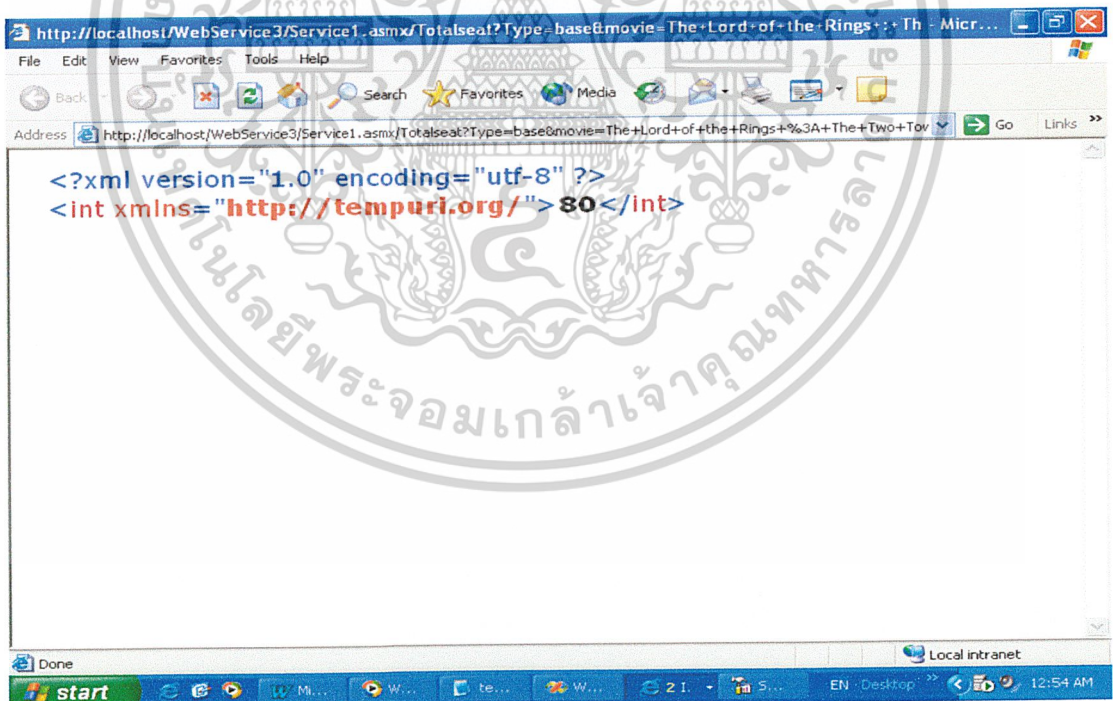
```

รูปที่ 3.11 การcoding web Service ในการให้ข้อมูลจำนวนที่นั่งในโรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 การเรียก Web Service เพื่อทดสอบการทำงาน โดยจะมีเอกสาร SOAP กำกับ เพื่อบอกรายละเอียดของ service



รูปที่ 3.13 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของ Service นี้จะได้กลับมาเป็น Integer ซึ่งแสดงในรูปแบบ XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Service ในการให้ข้อมูลแผนผังที่นั่งและสถานะของที่นั่งในขณะนั้น

```

<?xml version="1.0" encoding="utf-8" ?>
- <DataSet xmlns="http://tempuri.org/">
- <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
- <xs:element name="NewDataSet" msdata:IsDataSet="true">
- <xs:complexType>
- <xs:choice maxOccurs="unbounded">
- <xs:element name="temp">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="Seat" type="xs:string" minOccurs="0" />
  <xs:element name="Status" type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
  xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1" />
</DataSet>
  
```

รูปที่ 3.14 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของ Service นี้จะได้ออกมาเป็น Dataset ซึ่งแสดงในรูปแบบ XML

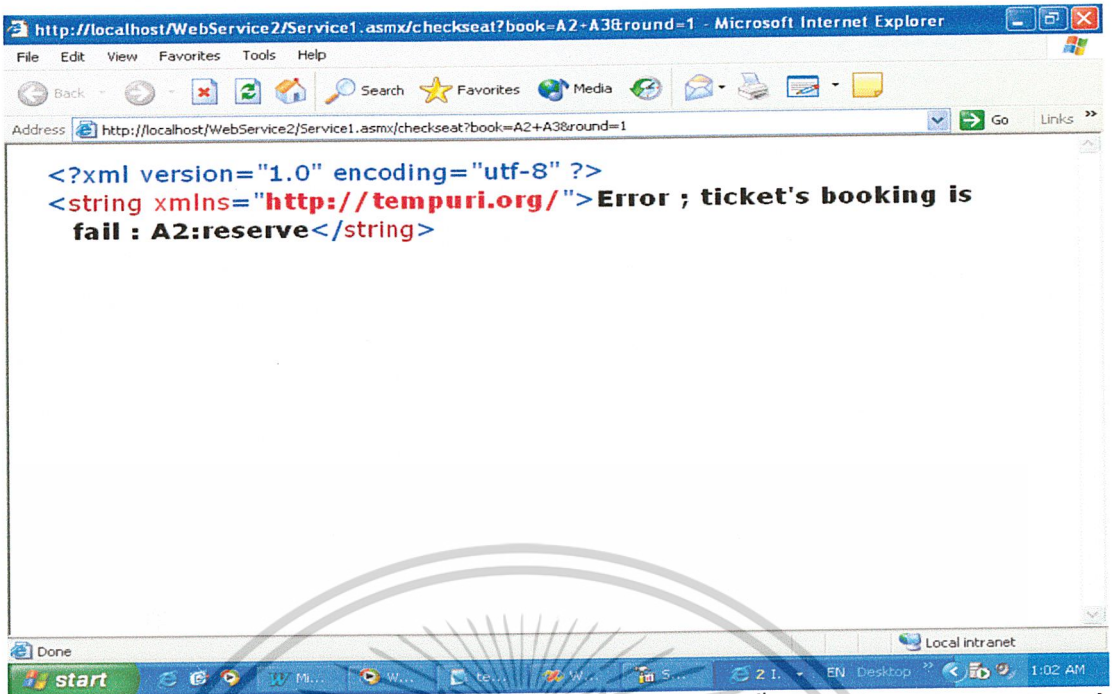
4. Service ในการทำการจอง โดยจะส่ง string ของที่นั่งที่ต้องการ ไปเป็นพารามิเตอร์ และจะได้ผลการจองกลับมา

```

<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://tempuri.org/">1A1</string>
  
```

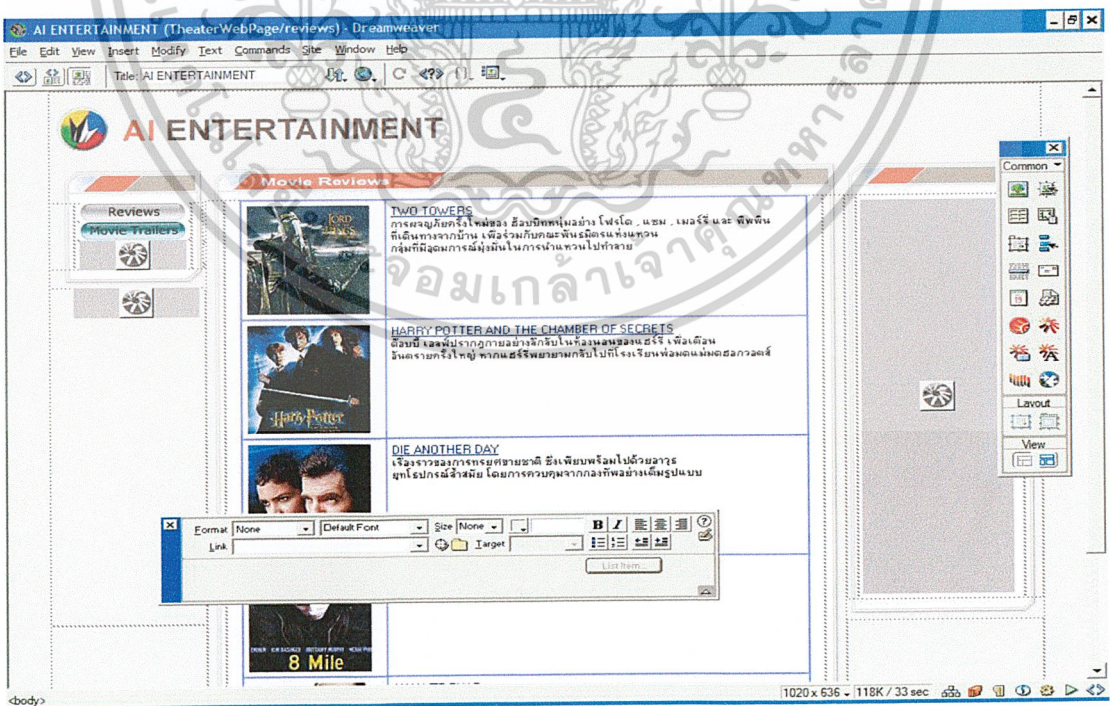
รูปที่ 3.15 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของ Service นี้จะได้ออกมาเป็น String ของ password ถ้าทำการจองสำเร็จ ซึ่งแสดงในรูปแบบ XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ผลจากการเรียกใช้ Web Service โดยผลลัพธ์ของ Service นี้จะ ได้กลับมาเป็น String ของที่
 นิ่งที่จองไม่ได้ ซึ่งแสดงในรูปแบบ XML

3.2.3 ขั้นตอนการสร้าง Web page ของระบบตัวแทนจำหน่ายด้วยมภาพยนตร์ ซึ่งจะมี
 การสร้าง web page ทั้งหมด 6 หน้า คือ Interview , Reviews , Movie Trailers , Timetable และหน้า
 จอสำหรับการจองตั๋วอีก 2 หน้า

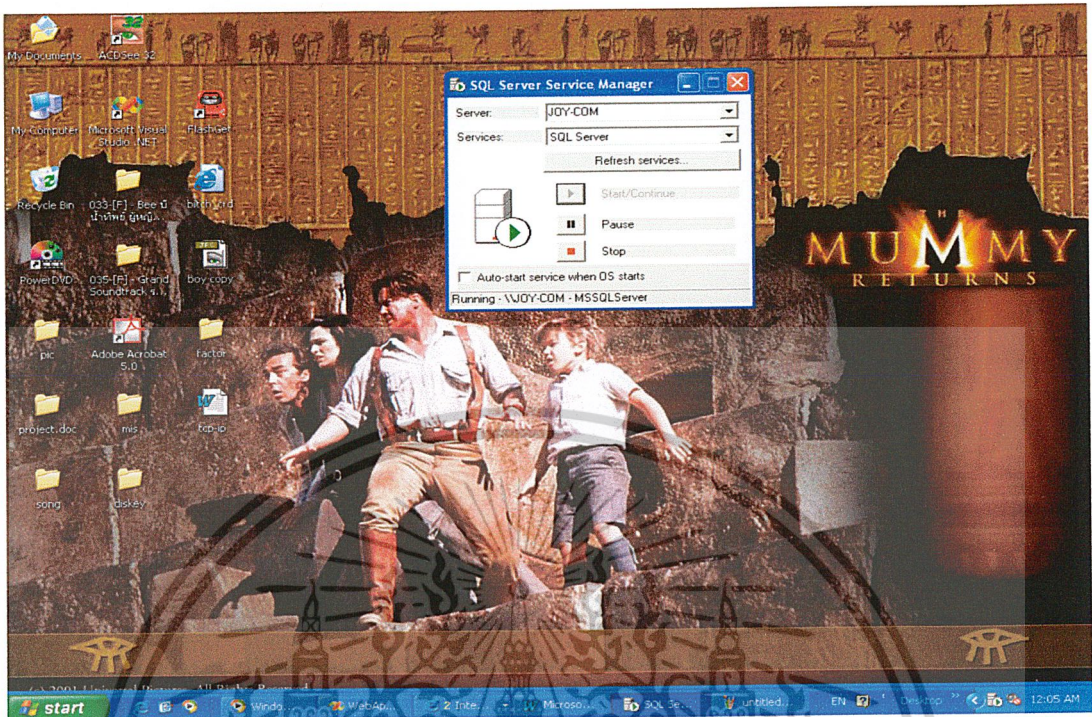


รูปที่ 3.17 การสร้าง Web page REVIEW โดยโปรแกรม Dreamweaver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

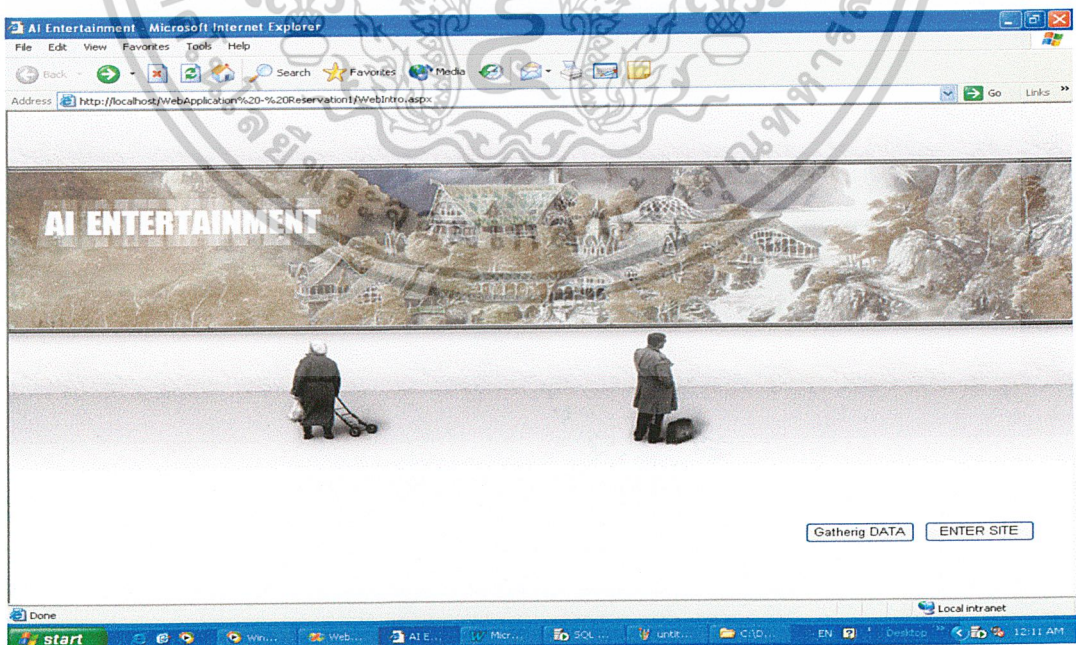
3.3 ขั้นตอนการทดสอบการทำงาน(Testing)

1. ทำการเปิดการใช้งานServer



รูปที่ 3. 18 การ running server

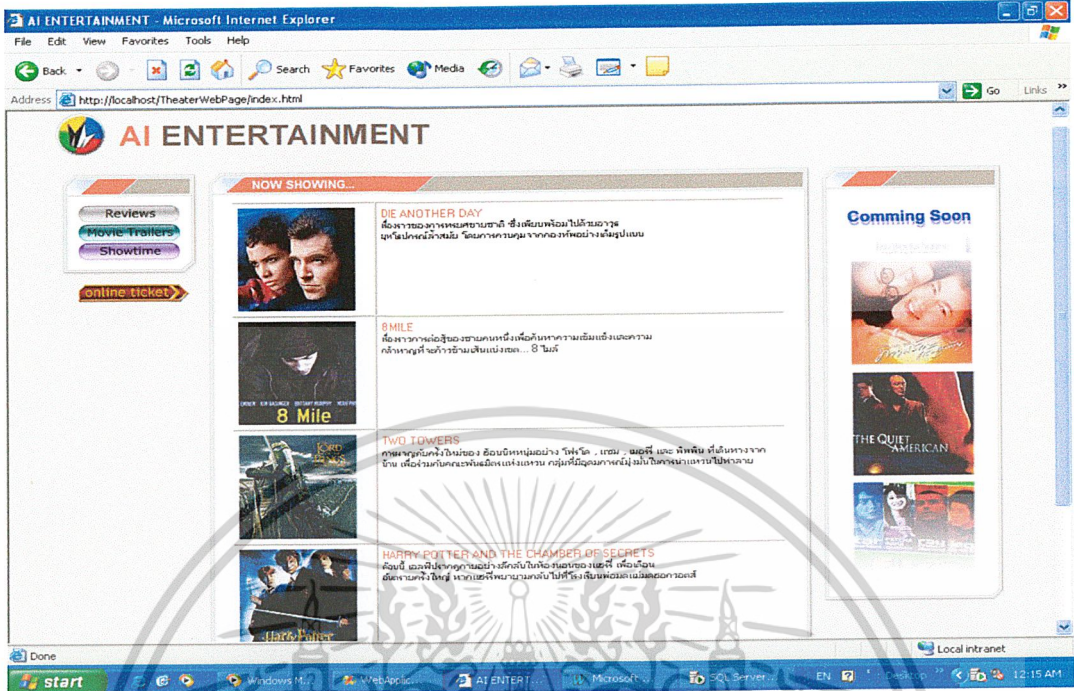
2. เปิด Browser แล้วเข้า <http://localhost/WebApplication-Reservation1/WebIntro> เพื่อเริ่มต้นการใช้งาน web site นี้



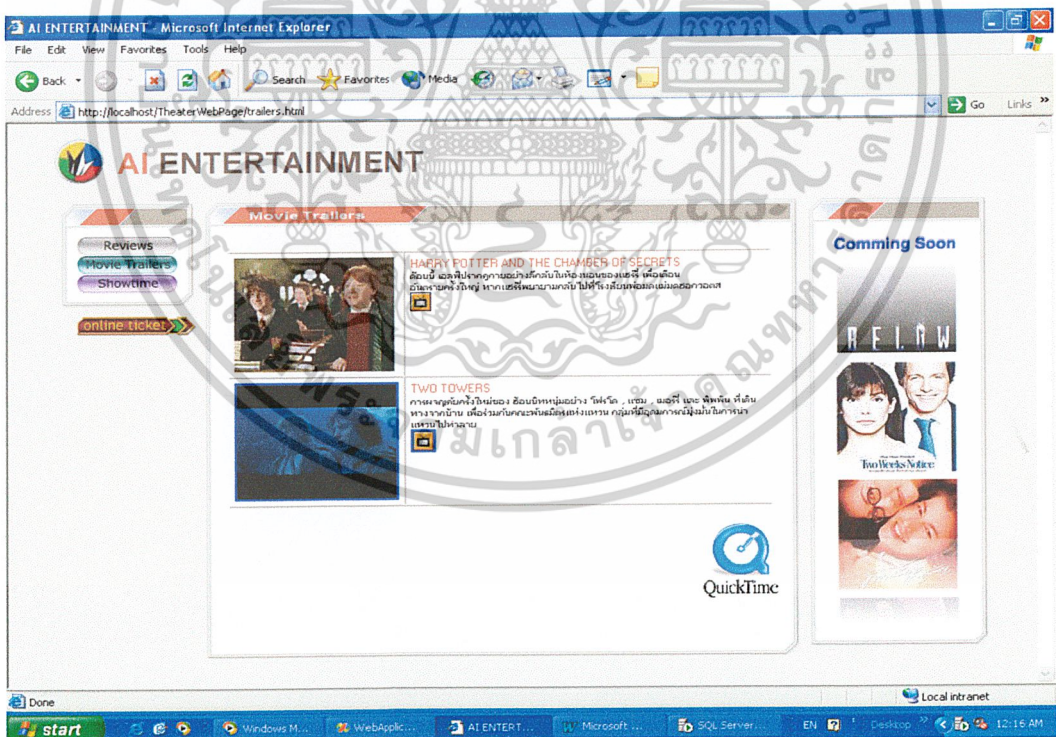
รูปที่ 3. 19 แสดงหน้าจอการเข้าสู่ web site AI Entertainment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อเริ่มเข้าสู่ web site จะมีข้อมูลเกี่ยวกับภาพยนตร์ต่างๆ เช่น ข้อมูลภาพยนตร์ที่กำลังฉายหรือกำลังจะเข้าฉาย ตัวอย่างภาพยนตร์ และ ตารางเวลาฉายภาพยนตร์ของแต่ละโรง

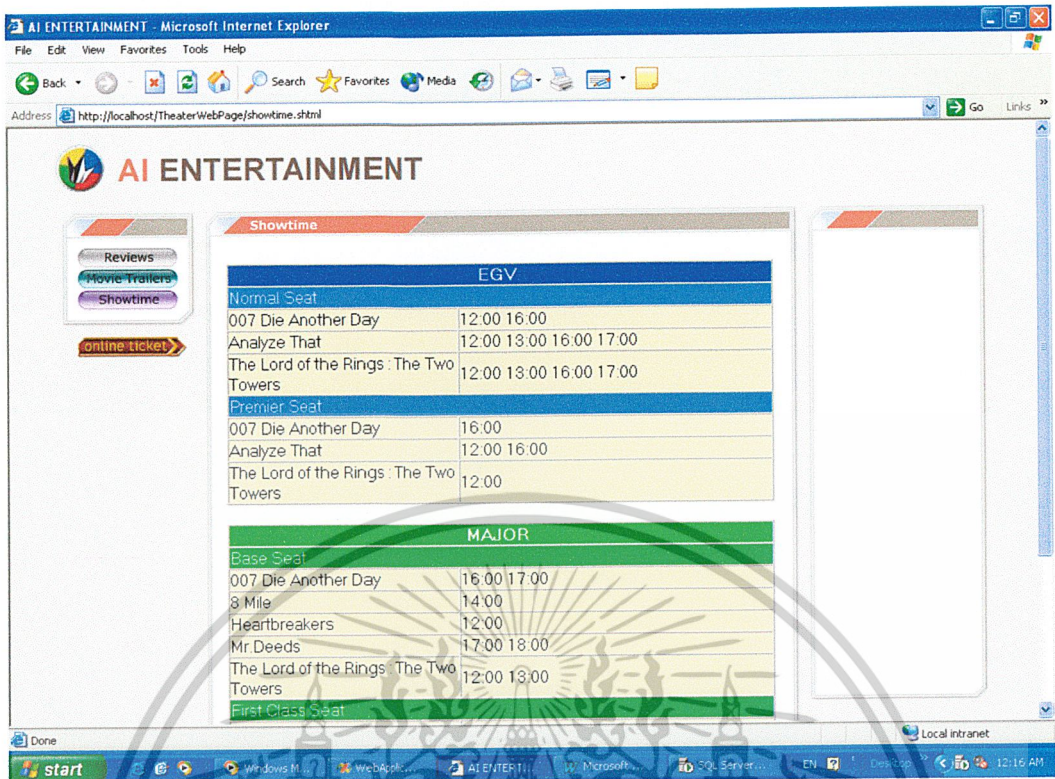


รูปที่ 3. 20 แสดง Review web page บอกรายละเอียดของภาพยนตร์ที่กำลังฉาย



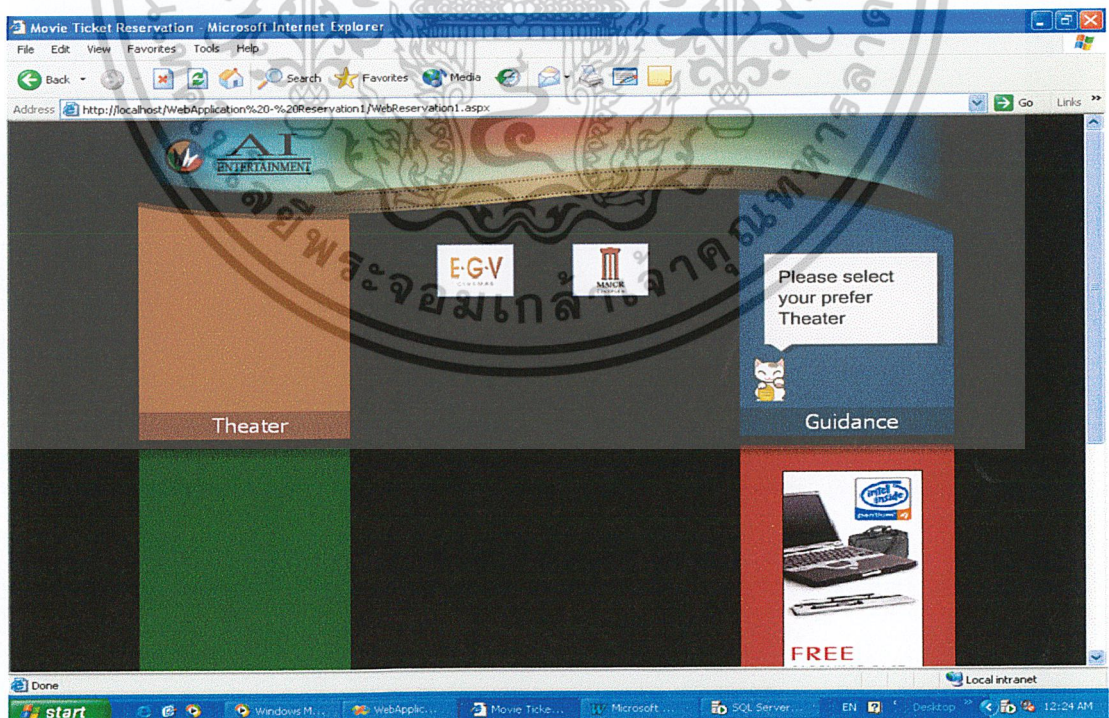
รูปที่ 3. 21 แสดงหน้าจอ Trailers แสดงตัวอย่างของภาพยนตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 ShowTime web page แสดงตารางเวลาของภาพยนตร์ที่กำลังเข้าฉาย

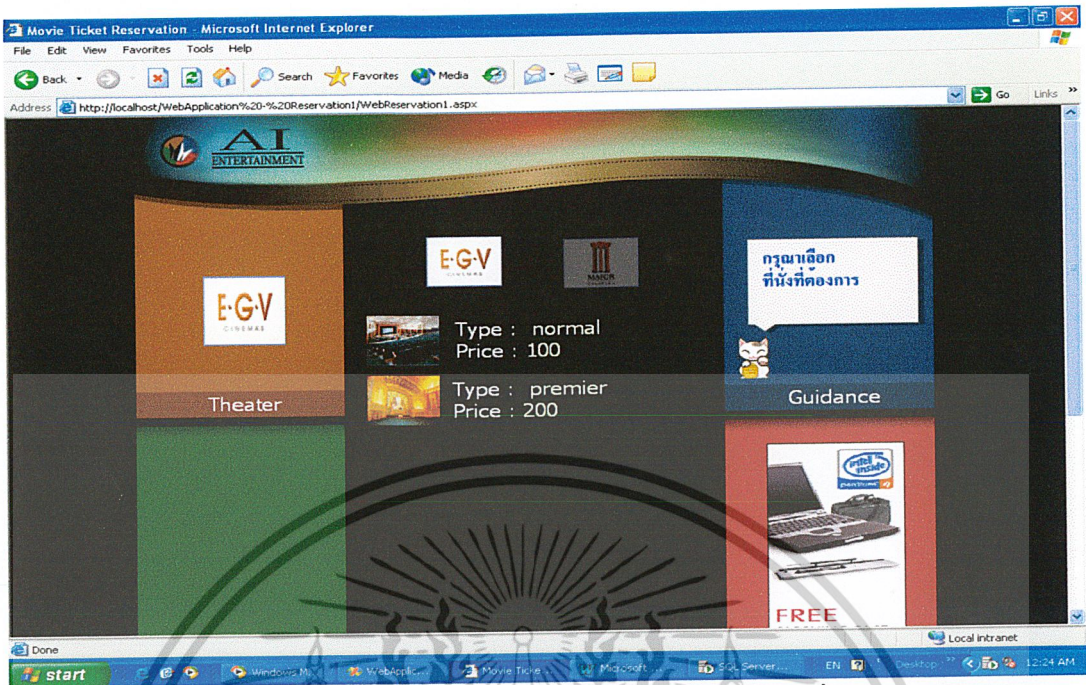
4. จากนั้นให้เข้าสู่การจองตั๋ว โดยเลือกที่ Online tickets



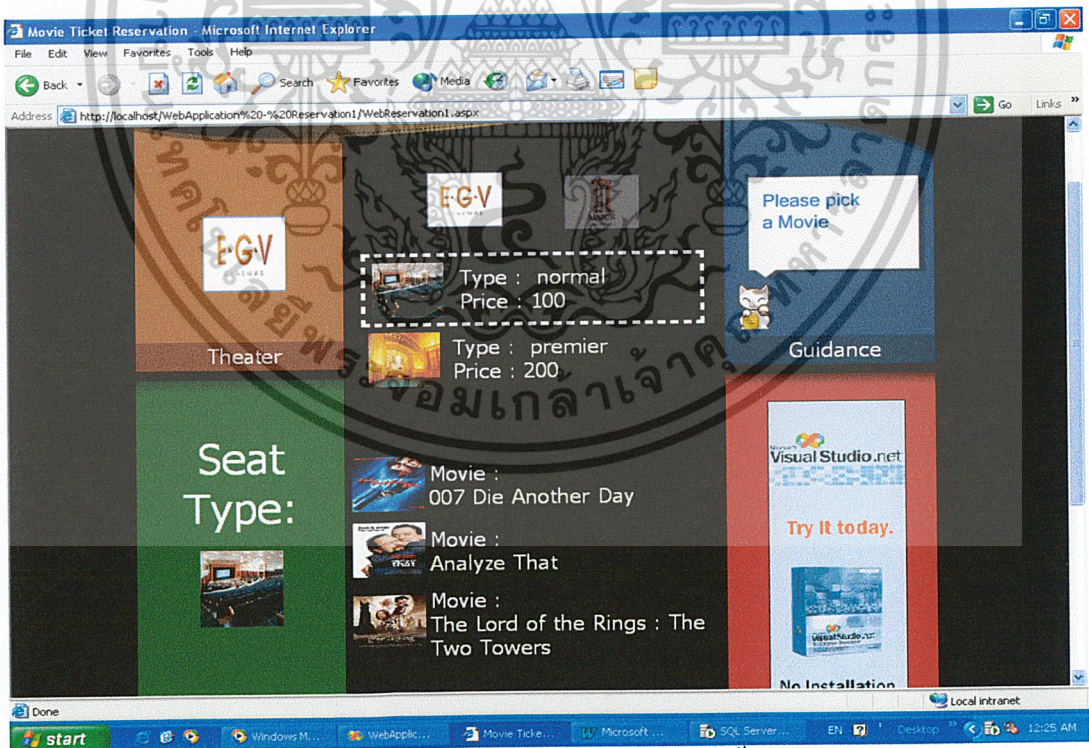
รูปที่ 3.23 แสดงการเข้าสู่ web page การจองตั๋ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เริ่มทำการจอง โดยเลือก โรงภาพยนตร์ ประเภทเก้าอี้ ภาพยนตร์ และ รอบ ที่ต้องการ ซึ่งจะมี ตัวอย่างขั้นตอนการทำงานอยู่ที่ด้านซ้ายของหน้าจอ



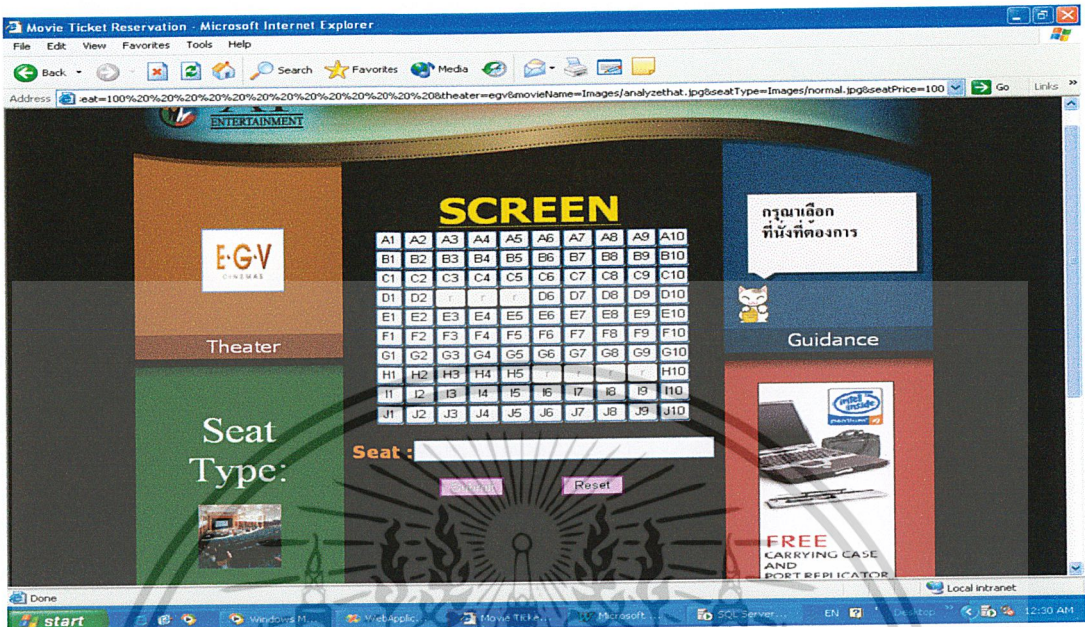
รูปที่ 3. 24 แสดงการเลือก โรงภาพยนตร์ เมื่อเลือกแล้วจะทำการแสดง โรงที่เลือกทางด้านซ้าย และ ทำการแสดงรูปแบบของที่นั่งที่มีในโรงให้เลือก



รูปที่ 3.25 แสดงการเลือกเก้าอี้ประเภทnormal ซึ่งประเภทของเก้าอี้ที่เลือกจะไปแสดงอยู่ที่ด้านซ้าย รวมทั้งทำการล้อมกรอบประเภทเก้าอี้ที่เลือกด้วย

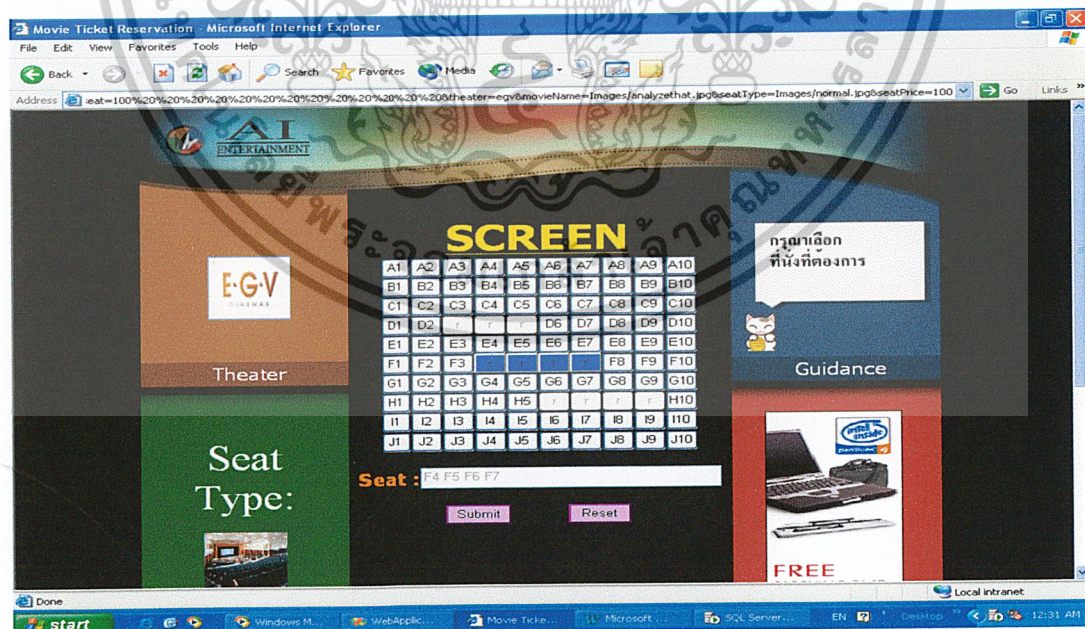
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 เมื่อเลือกรอบเสร็จ จะเข้าสู่หน้าจอการจอง ซึ่งจะแสดงที่นั่งในโรงภาพยนตร์ตามที่ได้เลือกมา(จะสังเกตได้ว่าที่นั่งบางที่นั่งได้ถูกจองไปแล้ว เนื่องจาก จะทำการเรียกwe serviceเพื่อขอสถานะที่นั่งในขณะนั้น)



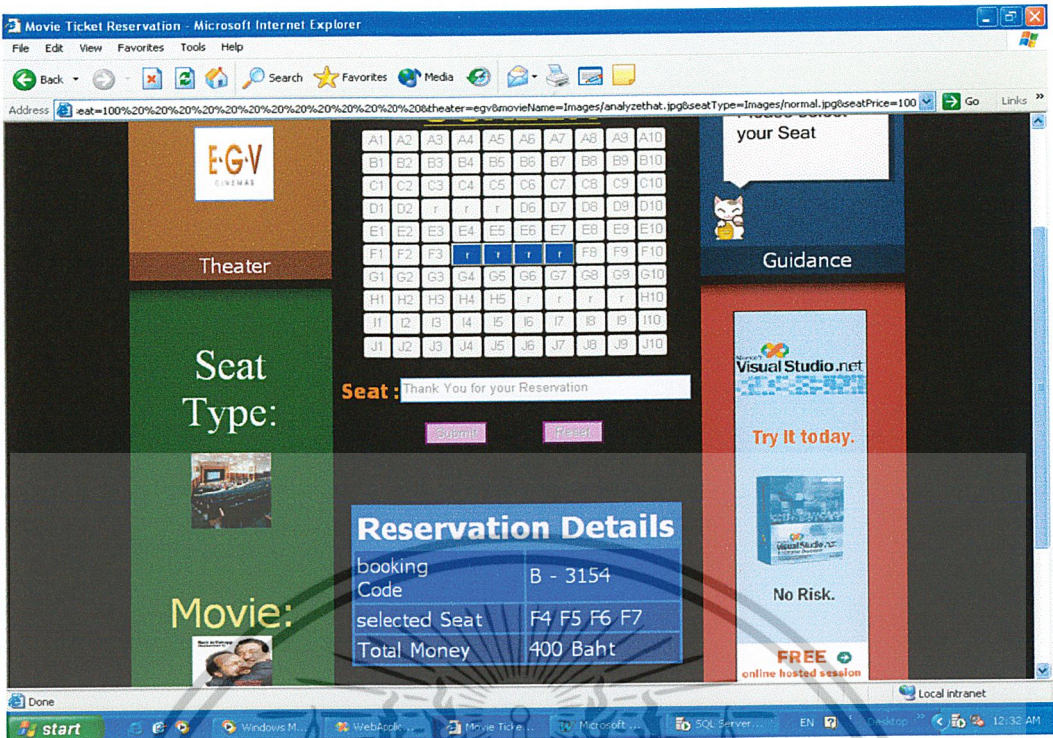
รูปที่ 3. 27 แสดง web page การจอง

7 กดเลือกที่นั่งที่ต้องการ แล้วกด Submit เพื่อทำการจอง โดยสามารถกดReset เพื่อเลือกที่นั่งใหม่อีกครั้งได้ หรือสามารถกดที่นั่งที่เลือกเพื่อยกเลิกสำหรับที่นั่งนั้นๆ หลังจากนั้น program จะทำการคำนวณเงินและแสดงผลการจอง

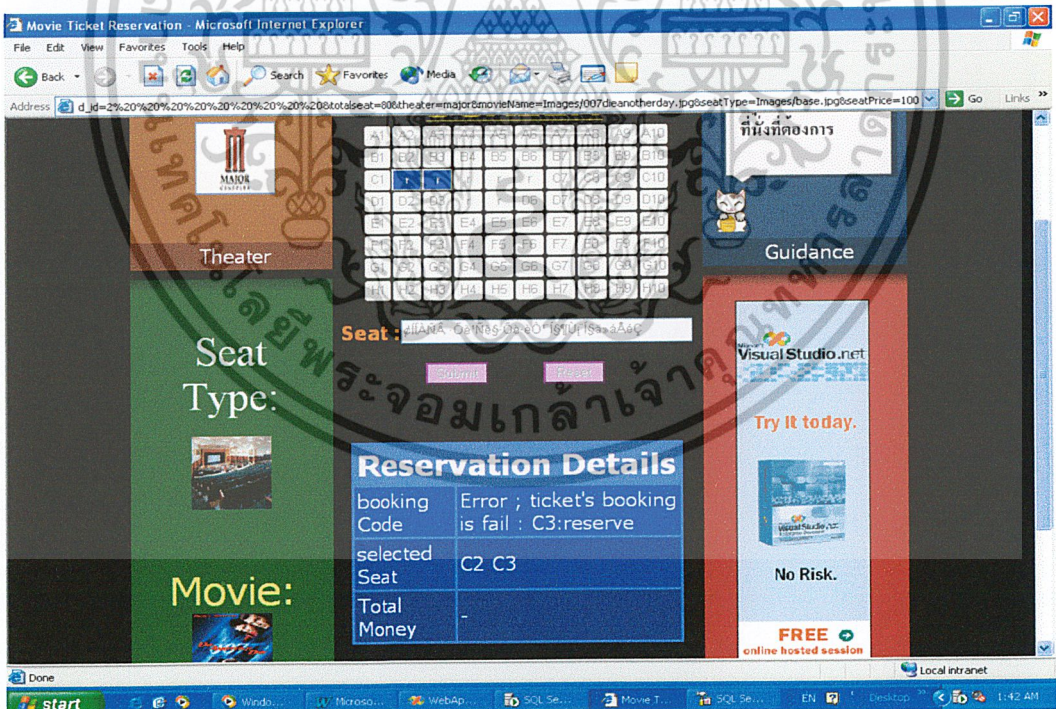


รูปที่ 3. 28 แสดงการจองที่นั่ง F3 - F7 จะทำให้ที่นั่งที่เลือกเปลี่ยนเป็นสีน้ำเงิน และระบบการจอง (reserve) ด้วยอักษร r รวมทั้งแสดงที่นั่งที่ถูกเลือกในFieldด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 29 แสดงผลสรุปการจองที่นั่งที่สำเร็จ โดยจะแสดงรหัสในการจอง ที่นั่งที่จอง และคำนวณเงินทั้งหมด



รูปที่ 3.30 แสดงผลสรุปการจองที่นั่งที่ไม่สำเร็จ โดยจะแสดงสาเหตุการผิดพลาดเนื่องจากที่นั่ง C3 ได้ถูกจองแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

สรุปผลการศึกษาและข้อเสนอแนะ

4.1 การทำงานของระบบ

โครงสร้างของระบบในปัญหาพิเศษนี้จะถูกแบ่งออกเป็น 3 ส่วนใหญ่ นั่นก็คือ ส่วนที่ 1 ส่วนที่เป็น Web Services ของโรงภาพยนตร์ที่ 1 และโรงภาพยนตร์ที่ 2 โดยโรงภาพยนตร์ทั้ง 2 แห่งจะมีฟังก์ชันการให้บริการ (Web Services) ในการขอข้อมูลเกี่ยวกับภาพยนตร์ การขอข้อมูลเกี่ยวกับที่นั่งว่างและไม่ว่างในโรงภาพยนตร์ การจองตั๋วภาพยนตร์ ส่วนที่ 2 ส่วนที่เป็นการเรียกใช้ Web Services ของตัวแทนจำหน่ายภาพยนตร์ (Ticket Agent) ส่วนที่ 3 ส่วนแสดงผล โดยนำข้อมูลต่างๆที่ได้จากโรงภาพยนตร์ทั้ง 2 แห่งมาเปลี่ยนเป็นข้อมูลที่แสดงผลในรูปแบบที่ผู้ใช้ทำความเข้าใจได้

4.2 ผลของปัญหาพิเศษ

ปัญหาพิเศษหัวข้อนี้ เป็นการสาธิตการใช้งาน Web Services โดยการสร้างระบบขอข้อมูลภาพยนตร์และการจองตั๋วภาพยนตร์ โดยตัวแทนจำหน่ายภาพยนตร์ (Ticket Agent) ดึงข้อมูลจากโรงภาพยนตร์ทั้ง 2 แห่งผ่านฟังก์ชันการให้บริการ (Web Services) ที่โรงภาพยนตร์ได้เตรียมไว้ให้

4.3 ข้อจำกัดของปัญหาพิเศษ

1. เนื่องจากเครื่องคอมพิวเตอร์ที่เป็นเครื่อง Server จะต้องรองรับการทำงานหลายอย่าง ดังนั้นจึงควรเป็นเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพค่อนข้างสูง
2. เนื่องจากคอมพิวเตอร์ที่เป็น Server จะต้องคอยรับและส่งข้อมูลไปยังเครื่อง Client ดังนั้นเครื่องคอมพิวเตอร์ที่เป็นเครื่อง Server จะต้องทำการเชื่อมต่ออินเทอร์เน็ตในกรณีที่ต้องการแสดงผลผ่าน WAP
3. การส่งข้อมูลอาจจะไม่เร็วเพียงพอต่อการต้องการของผู้ใช้ในกรณีที่ขณะนั้นมีการใช้งานเครือข่ายอินเทอร์เน็ตกันหนาแน่น

4.4 ปัญหาในการทำงาน

เนื่องจากในขณะที่ทำการศึกษายูนิ้น .NET Webservices เป็นเรื่องใหม่ ทำให้พบอุปสรรคเล็กน้อยในการศึกษาเนื่องจากข้อมูลทั้งหมดจะมาจากหนังสือต่างประเทศและเว็บไซต์ต่างประเทศ และในขณะที่เริ่มทำการศึกษา .NET Webservices เพิ่งออกรุ่นเต็มออกมา และบางส่วนยังอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวอร์ชันที่กำลังพัฒนา (Beta Version) ทำให้พบปัญหาในการพัฒนาบางส่วนเนื่องจากสภาพแวดล้อมในการพัฒนาไม่อยู่ในขั้นสมบูรณ์ 100 เปอร์เซ็นต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Andrew Krowczyk, Zach Greenvoss. **Professional C# Web Services: Building .NET Web Services with ASP.NET and .NET Remoting**. New Delhi : Wrox Inc. 2001

Harvey M. Deitel, Paul J. Dietel. 2001. **C# How to Program**. New York : Sams Publishing. 2001

Saurabh Nandu. **C# .NET Web Developer's Guide**. New York : Syngress Publishing. 2001



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.
การติดตั้งโปรแกรมและการใช้งานเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งโปรแกรมและการทำงานเบื้องต้น

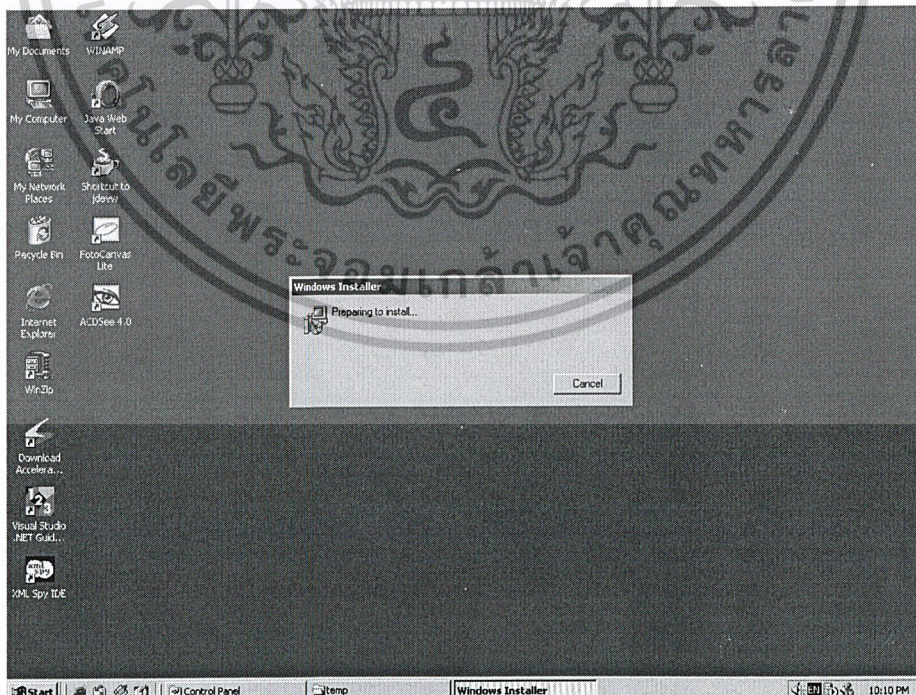
การติดตั้งโปรเจ็กต์

ก่อนทำการติดตั้งโปรเจ็กต์นี้จะต้องทำการตรวจสอบว่าอุปกรณ์ฮาร์ดแวร์ที่ใช้เป็นไปตามอุปกรณ์ขั้นต่ำที่จำเป็นต่อการติดตั้งหรือไม่ โดยอุปกรณ์ฮาร์ดแวร์ขั้นต่ำที่ต้องการดังรายการด้านล่าง

CPU	Intel Pentium 400 MHz หรือมากกว่า
หน่วยความจำ	128 MB หรือ 256 MB สำหรับการติดตั้ง Visual Studio.NET ในเครื่อง
ฮาร์ดดิสก์	กินเนื้อที่ในฮาร์ดดิสก์ 60 MB
ระบบปฏิบัติการ	Microsoft Windows 2000 + Service Pack 2, XP Professional
ระบบ Web Server	Internet Information Service 5.0 (IIS 5.0) + .NET Framework
อุปกรณ์อื่นๆ	ฟลอปปีดิสก์ 3.5 นิ้ว และ CD-ROM Drive

ในการติดตั้งโปรเจ็กต์มีขั้นตอนดังนี้

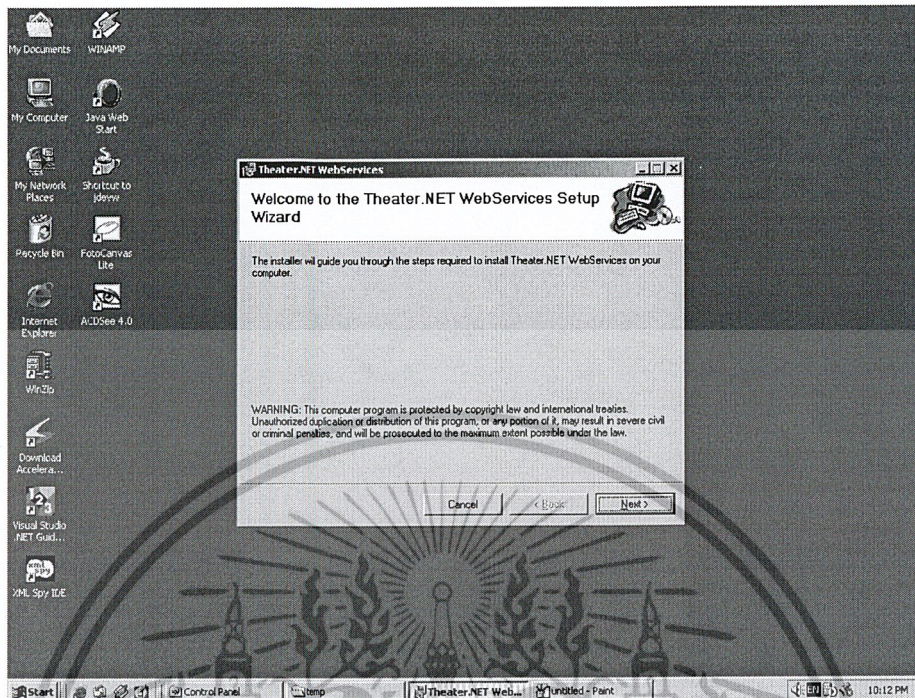
1. นำแผ่น CD-ROM ใส่ใน CD-ROM Drive จากนั้น Windows จะ Autorun CD-ROM ถ้าเกิดกรณีที่ Windows ไม่ Autorun CD-ROM ให้ทำการเรียกไฟล์ Setup ใน CD-ROM จะเกิดดังรูปที่ ก-1



รูปที่ ก-1 หน้าจอแสดงการเริ่มต้นติดตั้งโปรเจ็กต์จาก CD-ROM

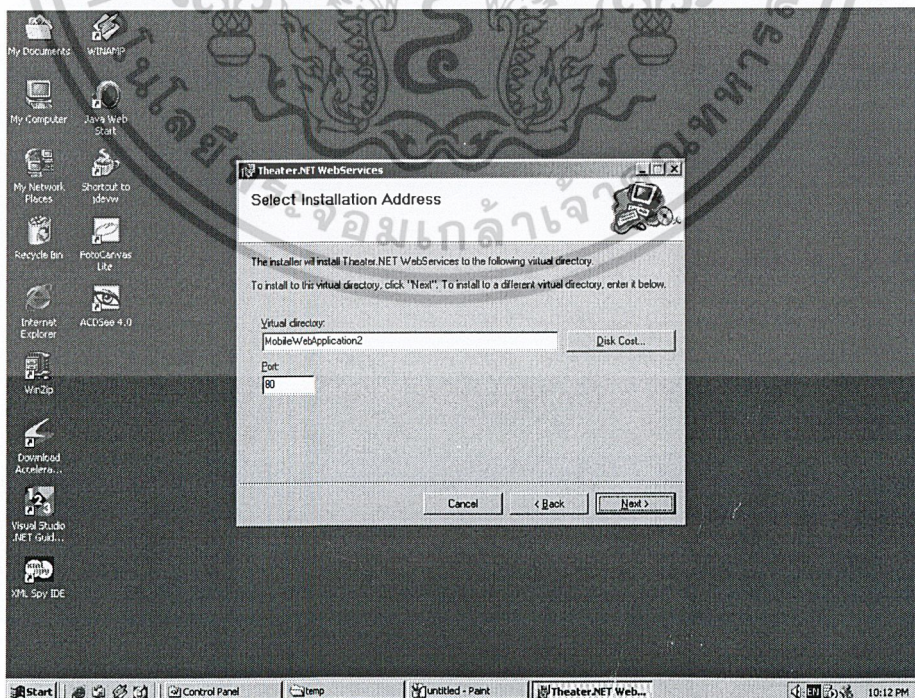
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อทำการอ่านข้อตกลงในการยอมรับเรื่องลิขสิทธิ์ ให้ทำการกดปุ่ม Next ดังรูปที่ ก-2



รูปที่ ก-2 หน้าจอแสดงรายละเอียดเกี่ยวกับเรื่องลิขสิทธิ์

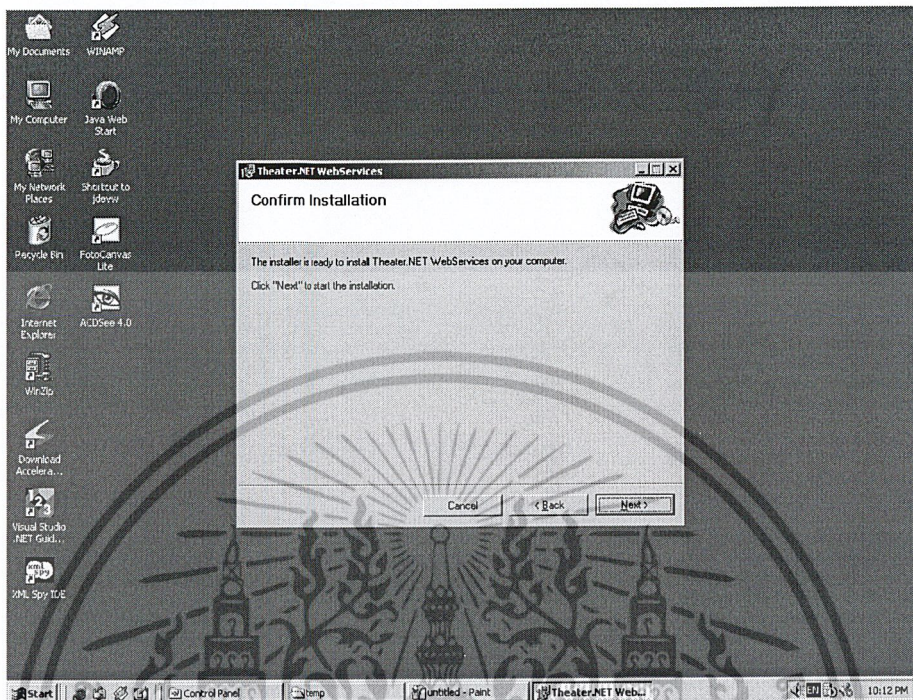
3. เลือก Virtual Directory ที่ต้องการทำการติดตั้งลงไป (ควรจะใช้ตาม default ที่ให้มา) จากนั้นให้ทำการกดปุ่ม Next ดังรูปที่ ก-3



รูปที่ ก-3 หน้าจอแสดงการเลือก Virtual Directory ที่ต้องการทำการติดตั้ง

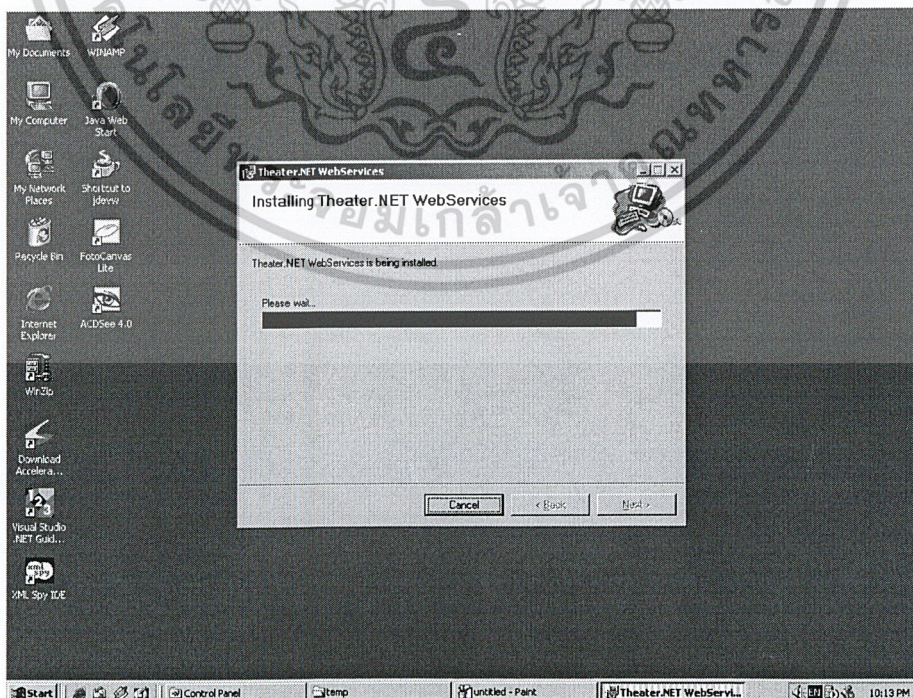
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตรวจสอบความพร้อมในการติดตั้งโปรแกรม จากนั้นกดปุ่ม Next ดังรูปที่ ก-4



รูปที่ ก-4 หน้าจอแสดงความพร้อมในการเริ่มการติดตั้งโปรแกรม

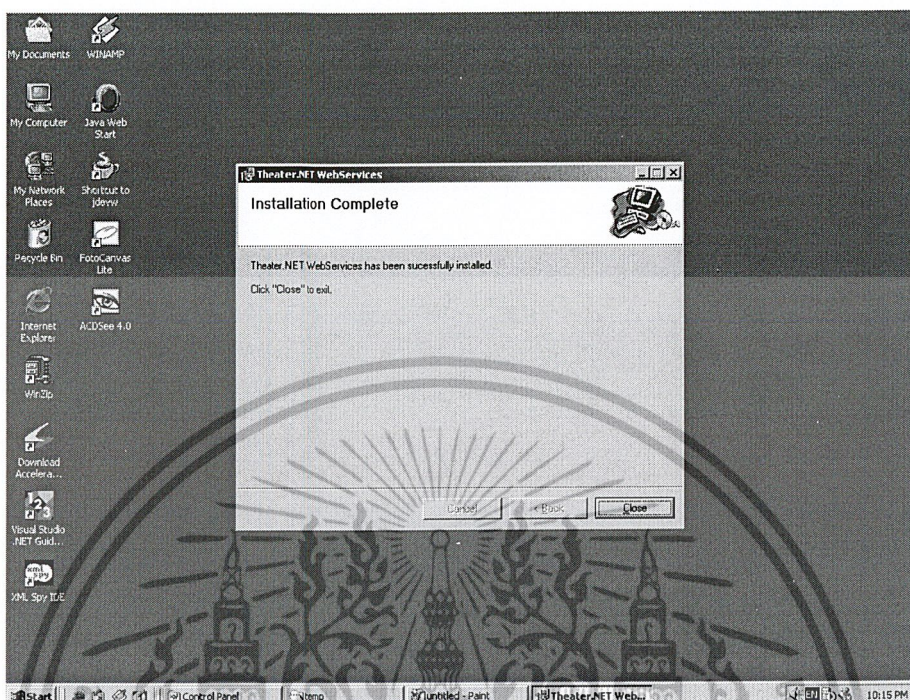
5. แสดงการติดตั้งโปรแกรม ดังรูปที่ ก-5



รูปที่ ก-5 แสดงหน้าจอกำลังติดตั้งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. แสดงหน้าจอเมื่อทำการติดตั้งโดยสมบูรณ์ ดังรูปที่ 6 จากนั้นกดปุ่ม Close



รูปที่ ก-6 แสดงหน้าจอเมื่อติดตั้งโปรแกรมสมบูรณ์

การใช้งานเบื้องต้น

ทำการตรวจสอบความพร้อมในการทำงานของโปรเจกต์โดยทำการเปิด Internet Explorer และพิมพ์ Address เป็น <http://localhost/WebApplication - Reservation1/webintro.aspx> หรือ <http://localhost/WebApplication - Reservation1> ถ้าหากไม่มีอะไรผิดพลาดจะแสดงผลดังรูปที่ ก-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-7 แสดงหน้าจอเริ่มต้นในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

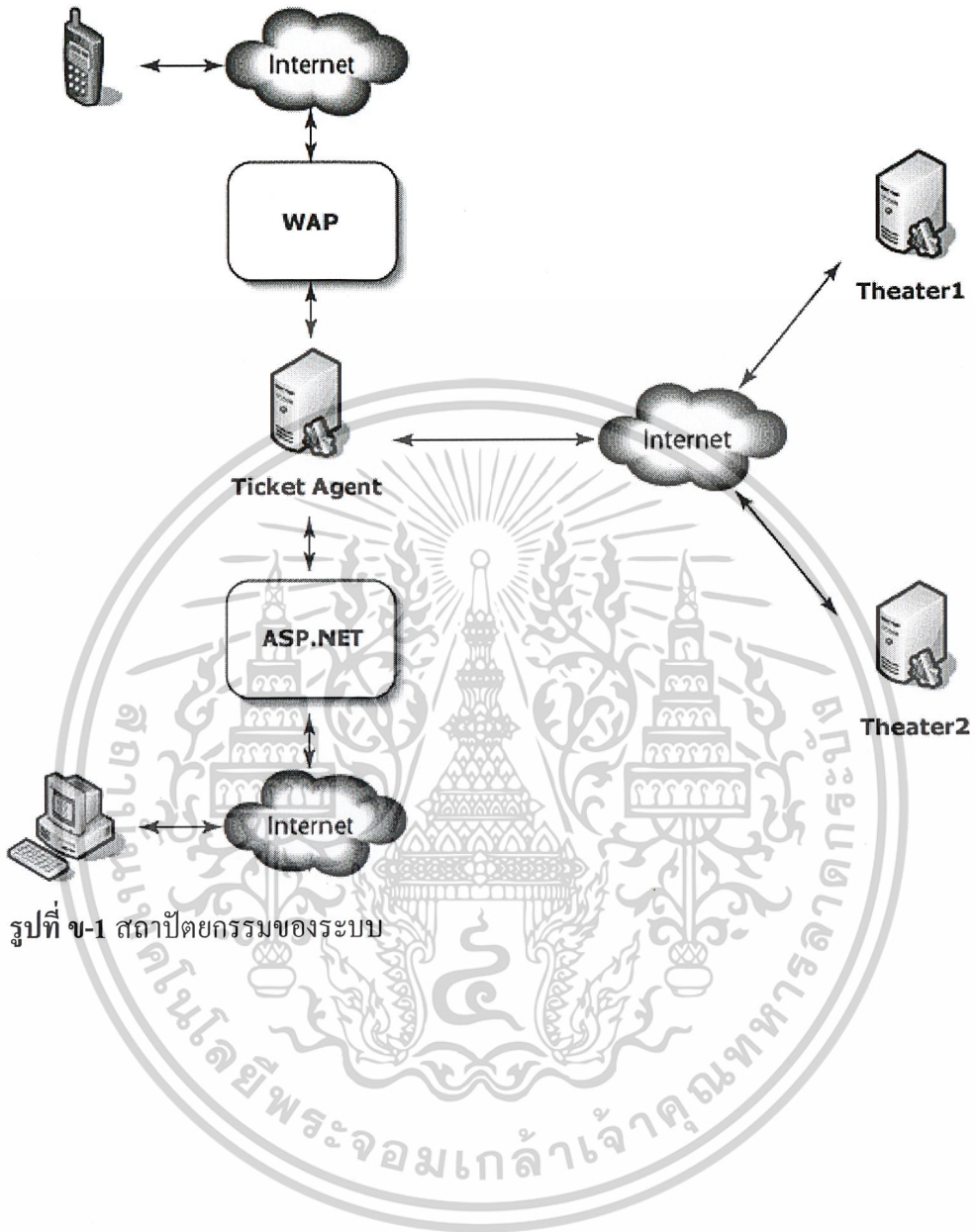


ภาคผนวก ข.

สถาปัตยกรรมของโปรเจ็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมของโปรเจ็ค



รูปที่ ข-1 สถาปัตยกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้