

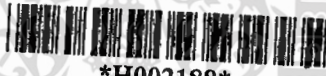
ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบขายหนังสือและสื่อการเรียนโดยใช้จาวาเว็บเซอร์วิส
The Development of a Learning Material Selling System
using Java Web Services

โดย

นางสาวดวงกมล อุทยานวิทยา

รหัส 45066077



H002188

อาจารย์ที่ปรึกษา

ดร. ภัทรชัย ลลิตโรจน์วงศ์

วัน เดือน ปี.....	0 8 ก.พ. 2550
เลขทะเบียน.....	02188
เลขเรียกหนังสือ.....	วท. ตา 53 ร 2547
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

611698142

1124 70987

รายงานนี้เป็นส่วนหนึ่งของวิชา โครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2547
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์ ดร. ภัทรชัย ลลิตโรจน์วงศ์ ซึ่งเป็นอาจารย์ที่ปรึกษา ที่ช่วยดูแลและให้คำปรึกษาต่างๆมาโดยตลอด ไม่ว่าจะเป็น สัมมนา 1 สัมมนา 2 ตลอดจนมาถึงการทำโครงการนี้

ขอขอบคุณ อาจารย์ทุกท่านของคณะเทคโนโลยีสารสนเทศ สจล. ที่ได้ให้ความรู้ในแขนงวิชาต่างๆ ภายในระยะ 2 ปีที่ข้าพเจ้าได้รับการศึกษาที่นี่ ข้าพเจ้าได้รับความรู้เพิ่มขึ้นเป็นอย่างมาก

ขอขอบคุณ คุณพ่อ คุณแม่ ที่คอยดูแลและเป็นกำลังใจในทุกๆเรื่องตลอดมา ทุกๆความสำเร็จของข้าพเจ้ามีคุณพ่อ คุณแม่คอยสนับสนุนอยู่เบื้องหลังทุกครั้งไป

ขอขอบคุณ น้องเบงค์ น้องใหม่ และน้องพิม ที่ช่วยเหลือข้าพเจ้าในเรื่องต่างๆสำหรับการทำโครงการนี้

ขอขอบคุณคุณ วีระพล และคุณเลอชาติ เจ้าของเอกสาร โครงการเว็บเซอร์วิสของคณะเทคโนโลยีสารสนเทศ สจล. ที่ข้าพเจ้าได้นำมาศึกษา

สุดท้ายนี้ขอขอบคุณทุกคนที่มีส่วนเกี่ยวข้องในการทำโครงการครั้งนี้ รวมถึงผู้ที่ถามตอบในกระทู้ต่างๆในอินเทอร์เน็ต ทำให้ข้าพเจ้าได้รู้ว่าไม่มีปัญหาอะไรที่เราจะแก้ไม่ได้ถ้าเรามีความพยายามมากพอ

ดวงกมล อุทยานวิทยา

ผู้จัดทำ

ชื่อหัวข้อ	ระบบขายหนังสือและสื่อการเรียน โดยใช้งานเว็บเซอร์วิส
นักศึกษา	นางสาวดวงกมล อุทยานวิทยา
อาจารย์ที่ปรึกษา	ดร. ภัทรชัย สถิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ระบบขายหนังสือและสื่อการเรียน โดยใช้งานเว็บเซอร์วิสนี้จัดทำขึ้นเพื่อให้บริการสำหรับการสั่งซื้อหนังสือและสื่อการเรียนผ่านทางเครือข่ายอินเทอร์เน็ต โดยมีลักษณะการทำงานเป็นแบบเว็บเซอร์วิส ซึ่งประกอบด้วยเซิร์ฟเวอร์ของผู้ให้บริการขายหนังสือ ผู้ให้บริการขายสื่อการเรียน ผู้ให้บริการขายบทเรียนออนไลน์ และผู้ให้บริการหลักทำงานร่วมกัน ผู้รับบริการนั้นจะได้รับความสะดวก เนื่องจากไม่ต้องเสียเวลาติดต่อกับแต่ละผู้ให้บริการด้วยตนเอง โดยผู้ให้บริการหลักจะทำการติดต่อให้ผู้รับบริการตั้งแต่ต้นจนจบ หัวข้อโครงการนี้จะทำการวิเคราะห์ ออกแบบ และพัฒนาระบบดังกล่าวโดยใช้ความสามารถของ XML และภาษาจาวาเพื่อให้เกิดประโยชน์สูงสุดในการทำงานข้ามแพลตฟอร์ม ซึ่งเป็นจุดสำคัญของเว็บเซอร์วิส

Title	The Development of a Learning Material Selling System using Java Web Services
Student	Miss. Duangkamol Uttayanwittaya
Advisor	Dr. Pattarachai Lalitrojwong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2547



Abstract

The goal of a learning material selling system is to support the way of selling learning materials on the Internet. The system contains book service, learning material service, and online course service which, interoperate using web services. The system facilitates customers for leveraging overhead in contacting several sub-services. The main service will operate this process for the customers. This project reports the analysis, design, and the system using the ability of XML and Java to create a powerful solution for platform-independent operations.

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
สารบัญ.....	III
สารบัญตาราง.....	V
สารบัญรูป.....	VI
บทที่	
1. บทนำ	
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนการดำเนินงานของโครงการ.....	3
2. ทฤษฎีที่เกี่ยวข้อง	
2.1 Web Services.....	5
2.2 Java 2 Platform Enterprise Edition	11
2.3 Java API of XML.....	18
3. การวิเคราะห์ความต้องการของระบบ	
3.1 ความต้องการทางฟังก์ชันของระบบรวม.....	28
3.2 การวิเคราะห์ความต้องการของระบบ โดยแยกแต่ละบริการ.....	33
4. การออกแบบระบบ	
4.1 สถาปัตยกรรมของระบบ.....	36
4.2 คลาสไดอะแกรม.....	37
4.3 ซีเควนซ์ไดอะแกรม.....	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

5. การออกแบบฐานข้อมูล	
5.1 การออกแบบฐานข้อมูลของผู้ให้บริการหลัก.....	53
5.2 การออกแบบฐานข้อมูลของผู้ให้บริการหนังสือ.....	55
5.3 การออกแบบฐานข้อมูลของผู้ให้บริการธนาคาร.....	56
5.4 ข้อสมมติ	57
6. การพัฒนาระบบ	
6.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ.....	58
6.2 การพัฒนาระบบของผู้ให้บริการหนังสือ (JAX-RPC Book Service).....	59
6.3 การพัฒนาระบบของผู้ให้บริการหลัก (Main Service).....	63
6.4 การพัฒนาระบบของผู้ให้บริการย่อยในแบบ (SAAJ Bank Service).....	66
7. สรุปผลการดำเนินงาน	
7.1 ประโยชน์ที่ได้รับ.....	69
7.2 ข้อจำกัดของระบบ.....	69
7.3 ปัญหาและอุปสรรคที่พบ.....	70
7.4 แนวทางในการพัฒนาต่อ.....	70
บรรณานุกรม	71
ประวัติผู้เขียน	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่	
2.1 ตารางชุดคำสั่งใน J2EE.....	12
3.1 ตารางสรุปบริการที่ต้องมีสำหรับแต่ละระบบย่อย.....	29
4.1 ตารางอธิบายรายละเอียดประกอบยูสเคส RequestProductInfo.....	42
4.2 ตารางอธิบายรายละเอียดประกอบยูสเคส RequestPlaceOrder.....	43
4.3 ตารางอธิบายรายละเอียดประกอบยูสเคส RequestCheckoutOrder.....	44
4.4 ตารางอธิบายรายละเอียดประกอบยูสเคส AuthenticateCustomer.....	45
4.5 ตารางอธิบายรายละเอียดประกอบ CreateNewCustomer.....	46
4.6 ตารางอธิบายรายละเอียดประกอบยูสเคส RequestPriceList.....	47
4.7 ตารางอธิบายรายละเอียดประกอบยูสเคส PlaceOrder.....	48
4.8 ตารางอธิบายรายละเอียดประกอบยูสเคส PerformPaymentTransaction.....	49
4.9 ตารางอธิบายรายละเอียดประกอบยูสเคส RegisterService.....	50
4.10 ตารางอธิบายรายละเอียดประกอบยูสเคส WithdrawService.....	51
4.11 ตารางอธิบายรายละเอียดประกอบยูสเคส FindService.....	52
5.1 พจนานุกรมข้อมูลของตาราง CUSTOMER.....	54
5.2 พจนานุกรมข้อมูลของตาราง ORDER.....	54
5.3 พจนานุกรมข้อมูลของตาราง ORDERDETAIL.....	54
5.4 พจนานุกรมข้อมูลของตาราง DISTRIBUTOR.....	54
5.5 พจนานุกรมข้อมูลของตาราง BOOK.....	55
5.6 พจนานุกรมข้อมูลของตาราง ORDER.....	55
5.7 พจนานุกรมข้อมูลของตาราง ORDERDETAIL.....	56
5.8 พจนานุกรมข้อมูลของตาราง CARD.....	56
5.9 พจนานุกรมข้อมูลของตาราง TRANSICTION.....	57
6.1 ข้อมูลที่ต้องใช้ในการลงทะเบียนบริการกับ Registry Server.....	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การทำงานของ Web Services.....	5
2.2 SOAP Message.....	7
2.3 SOAP Listener.....	7
2.4 WSDL.....	9
2.5 UDDI.....	10
2.6 ตัวอย่างไฟล์ DTD (confirm.dtd).....	10
2.7 ตัวอย่างไฟล์ XML (confirm.xml).....	11
2.8 โครงสร้างของ Java Application Server.....	14
2.9 วงจรชีวิตของ Servlets.....	15
2.10 การเรียกใช้งาน EJB.....	17
2.11 SAX APIs.....	19
2.12 DOM APIs.....	19
2.13 XSLT APIs.....	20
2.14 ขั้นตอนการ Binding.....	21
2.15 ตัวอย่างการ Binding.....	22
2.16 SOAP Message with No Attachments.....	23
2.17 SOAP Message with Attachments.....	24
2.18 การทำงานของ JAX-RPC.....	25
2.19 JAXR Architecture.....	27
3.1 Actor ของระบบ.....	30
3.2 Activity Diagramของระบบทั้งหมด.....	31
3.3 ยูสเคสไดอะแกรมของผู้ให้บริการหลัก.....	33
3.4 ยูสเคสไดอะแกรมของผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน.....	34
3.5 ยูสเคสไดอะแกรมของผู้ให้บริการธนาคาร.....	34
3.6 ยูสเคสไดอะแกรมของผู้ให้บริการบัญชีรายชื่อเว็บเซอร์วิส.....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 สถาปัตยกรรมของระบบ.....	36
4.2 คลาสไดอะแกรมของผู้ให้บริการหลัก.....	37
4.3 คลาสไดอะแกรมของผู้ให้บริการหนังสือและสื่อการเรียน.....	39
4.4 คลาสไดอะแกรมของผู้ให้บริการธนาคาร.....	41
4.5 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส RequestProductInfo.....	42
4.6 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส RequestPlaceOrder.....	43
4.7 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส RequestCheckoutOrder.....	44
4.8 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส AuthenticateCustomer.....	45
4.9 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส CreateNewCustomer.....	46
4.10 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส RequestPriceList.....	47
4.11 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส PlaceOrder.....	48
4.12 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส PerformPaymentTransaction.....	49
4.13 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส RegisterService.....	50
4.14 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส WithdrawService.....	51
4.15 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส FindService.....	52
5.1 ER Diagram ของผู้ให้บริการหลัก.....	53
5.2 ER Diagram ของผู้ให้บริการหนังสือ.....	55
5.3 ER Diagram ของผู้ให้บริการธนาคาร.....	56
6.1 การค้นหาบริการใน Registry Server.....	60
6.2 ไฟล์ WSDL ของ JAX-RPC Book Service.....	61
6.3 หน้าจอตรวจสอบการสั่งซื้อจากผู้ให้บริการหลักของผู้ให้บริการหนังสือ.....	62
6.4 หน้าจอตรวจสอบการสั่งซื้อจากผู้ให้บริการหลักของผู้ให้บริการซีดี.....	62
6.5 หน้าจอเข้าสู่ระบบของผู้ให้บริการหลัก.....	63
6.6 หน้าจอเลือกรายการสินค้าของผู้ให้บริการหลัก.....	64

สารบัญรูป (ต่อ)

รูปที่	หน้า
6.7 หน้าจอลงชื่อสินค้าของผู้ให้บริการหลัก.....	65
6.8 หน้าจอกรอกข้อมูลการชำระเงินของผู้ให้บริการหลัก.....	65
6.9 หน้าจอยืนยันการสั่งซื้อและชำระเงินของผู้ให้บริการหลัก.....	66
6.10 รูปแบบข้อมูลการชำระเงิน.....	67
6.11 รูปแบบข้อมูลยืนยันการชำระเงิน.....	67
6.12 หน้าจอตรวจสอบการชำระเงินจากผู้ให้บริการหลัก.....	68



บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันการค้นหาข้อมูลและการสั่งซื้อสินค้าผ่านทางเครือข่ายอินเทอร์เน็ตกำลังเป็นที่นิยมเป็นอย่างมากเนื่องจากสะดวกสบาย ประหยัดเวลา และสามารถให้บริการได้ตลอดเวลา ในปัจจุบันมีร้านค้าทางอินเทอร์เน็ตเปิดให้บริการมากมายและลูกค้าเองก็ต้องการรับบริการจากหลายๆร้านในขณะหนึ่งเช่นกัน ยกตัวอย่างเช่น หากลูกค้าต้องการซื้อหนังสือเกี่ยวกับการสนทนาภาษาอังกฤษ ก็อาจต้องการดูข้อมูลของสื่อการเรียนที่เกี่ยวข้องกันเช่น เทปหรือซีดีรวมไปถึงข้อมูลของบทเรียนออนไลน์ต่างๆด้วย ซึ่งหากจะให้ลูกค้าไปติดต่อกับแต่ละบริการดังกล่าวเองแต่ละครั้ง ก็จะเป็นการสิ้นเปลืองเวลา และลูกค้าเองก็ต้องรู้เว็บไซต์ที่ให้บริการดังกล่าวตั้งแต่ต้น

การใช้เทคโนโลยีเว็บเซอร์วิสมาใช้งานจะช่วยแก้ไขข้อไม่สะดวกดังกล่าวได้ เนื่องจากเว็บเซอร์วิสสามารถทำให้แต่ละผู้ให้บริการสามารถติดต่อกันได้ ถึงแม้ว่าจะพัฒนาขึ้นจากแพลตฟอร์มที่แตกต่างกัน ซึ่งจะเป็นการประหยัดเวลาสำหรับลูกค้า เนื่องจากติดต่อมาที่เดียวแล้วแต่ละผู้ให้บริการจะไปติดต่อกันเอง จนกว่าจะได้ข้อมูลที่ต้องการ แล้วจึงส่งกลับมาให้ลูกค้า นอกจากนี้ลูกค้าก็ไม่จำเป็นต้องรู้ถึงตำแหน่งของผู้ให้บริการ เนื่องจากจะมีฐานข้อมูลขนาดใหญ่ที่ใช้เก็บรายการเว็บเซอร์วิสไว้อยู่แล้วซึ่ง ผู้ให้บริการจะเป็นผู้ติดต่อให้ลูกค้าเช่นกัน

ในปัจจุบันมีการกำเนิดของภาษาใหม่คือ Extensible Markup Language หรือ XML ซึ่งมีความสามารถในการเคลื่อนย้ายข้อมูลได้ง่าย (Data Portable) คือสามารถใช้งานข้อมูลได้หลากหลาย มีความเป็นมาตรฐานและโปรแกรมส่วนใหญ่สามารถตีความข้อมูลได้ คุณสมบัติเหล่านี้ทำให้ภาษา XML สามารถใช้ร่วมกับเทคโนโลยีเว็บเซอร์วิสที่กล่าวมาข้างต้นได้เป็นอย่างดี นอกจากนี้สำหรับภาษาจาวานั้นก็มีลักษณะเด่นในการเคลื่อนย้ายโค้ดได้ง่าย (Code Portable) คือสามารถนำโปรแกรมไปใช้งานที่ใดก็ได้ (ที่มี JVM) ซึ่งเมื่อใช้ความสามารถของ XML และภาษาจาวาร่วมกัน จะทำให้การออกแบบระบบงานแบบเว็บเซอร์วิสสามารถใช้ประโยชน์ได้มาก เนื่องจากเว็บเซอร์วิสเป็นการทำงานร่วมกันของเซอร์วิสจำนวนมากที่ต่างแพลตฟอร์มกัน

1.2 วัตถุประสงค์ของโครงการ

ระบบขายหนังสือและสื่อการเรียน โดยใช้จาวาเว็บเซอร์วิสนี้จัดทำขึ้นเพื่อ

1. ให้บริการ การแสดงผลมูลราคา ของสื่อการเรียนประเภท หนังสือ เทป และซีดีรอม
2. ให้บริการ การสั่งซื้อ สื่อการเรียนประเภท หนังสือ เทป และซีดีรอม
3. ให้บริการ การชำระเงิน สื่อการเรียนประเภท หนังสือ เทป และซีดีรอม
4. ลักษณะการทำงานโดยรวมของระบบเป็นการทำงานร่วมกันของระบบเว็บแอปพลิเคชันของผู้ให้บริการหลัก ระบบเว็บเซอร์วิสของผู้ให้บริการหนังสือ ระบบเว็บเซอร์วิสของผู้ให้บริการสื่อการเรียน และระบบเว็บเซอร์วิสของผู้ให้บริการธนาคาร

1.3 ขอบเขตของโครงการ

ระบบขายหนังสือและสื่อการเรียน โดยใช้จาวาเว็บเซอร์วิสนี้ประกอบด้วยส่วนงานต่างๆ ดังต่อไปนี้

1. ระบบเว็บเซอร์วิสของผู้ให้บริการหนังสือ (Book Service)
ให้บริการหนังสือภาษาไทยและหนังสือต่างประเทศ
2. ระบบเว็บเซอร์วิสของผู้ให้บริการสื่อการเรียน (Learning Material Service)
ให้บริการเทปและซีดีรอม
3. ระบบเว็บเซอร์วิสของผู้ให้บริการธนาคาร (Bank Service)
ให้บริการการชำระเงินผ่านบัตรเครดิต
4. ระบบเว็บแอปพลิเคชันของผู้ให้บริการหลัก (Main Service)
ให้บริการการแสดงผลมูลราคา สั่งซื้อและชำระเงินแก่ลูกค้า โดยการเรียกใช้งานไปยังระบบเว็บเซอร์วิสของผู้ให้บริการหนังสือ ระบบเว็บเซอร์วิสของผู้ให้บริการสื่อการเรียน และระบบเว็บเซอร์วิสของผู้ให้บริการธนาคาร

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ลูกค้า

1. สะดวกสบาย ประหยัดเวลา ทำการติดต่อ ณ. จุดเดียว
2. ได้ข้อมูลหลากหลายและทันสมัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน

1. เพิ่มช่องทางการจำหน่ายสินค้า เพิ่มรายได้

ผู้ให้บริการธนาคาร

1. เพิ่มช่องทางการชำระเงิน เพิ่มรายได้

ผู้พัฒนาระบบ

1. ได้รับความรู้เกี่ยวกับเทคโนโลยีเว็บเซอร์วิสและภาษาจาวา
2. ได้นำทฤษฎีเกี่ยวกับการการโปรแกรมเชิงวัตถุและ Unified Modeling Language มาประยุกต์ใช้ในการออกแบบและพัฒนาระบบงานจริง
3. เป็นแนวทางในการพัฒนาระบบเว็บเซอร์วิสและเว็บแอปพลิเคชันที่มีความซับซ้อนมากขึ้นในอนาคต

1.5 ขั้นตอนการดำเนินงานในโครงการ

1. ศึกษาความเป็นไปได้ ประกอบด้วยขั้นตอนย่อยต่างๆดังต่อไปนี้
 - 1.1 ศึกษาเทคโนโลยีจาวาเว็บเซอร์วิสและจาวาเว็บแอปพลิเคชัน
 - 1.2 ศึกษาการทำงานของร้านขายหนังสือและร้านขายหนังสือออนไลน์ที่มีอยู่ในปัจจุบัน
2. วิเคราะห์ระบบ
 - 2.1 วิเคราะห์ความต้องการของระบบโดยรวม
 - 2.2 วิเคราะห์ความต้องการสำหรับแต่ละส่วนคือ ผู้ให้บริการหลัก ผู้ให้บริการหนังสือ ผู้ให้บริการสื่อการเรียน และผู้ให้บริการธนาคาร
 - 2.3 ออกแบบยูสเคสไดอะแกรม
3. ออกแบบระบบ
 - 3.1 ออกแบบสถาปัตยกรรมของระบบ
 - 3.2 ออกแบบโครงสร้างพื้นฐานของระบบและเขียนคลาสไดอะแกรม
 - 3.3 ออกแบบพฤติกรรมการทำงาน
 - 3.4 เขียนซีควเอนซ์ไดอะแกรม
 - 3.5 ออกแบบส่วนเก็บข้อมูล
 - 3.6 ออกแบบส่วนติดต่อกับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. พัฒนาระบบ

- 4.1 พัฒนาระบบเว็บเซอร์วิสของผู้ให้บริการหนังสือ
- 4.2 พัฒนาระบบเว็บเซอร์วิสของผู้ให้บริการสื่อการเรียน
- 4.3 พัฒนาระบบเว็บเซอร์วิสของผู้ให้บริการธนาคาร
- 4.4 พัฒนาระบบเว็บแอปพลิเคชันของผู้ให้บริการหลัก โดยการเรียกใช้งานไปยังระบบเว็บเซอร์วิสของผู้ให้บริการหนังสือ ระบบเว็บเซอร์วิสของผู้ให้บริการสื่อการเรียน และระบบเว็บเซอร์วิสของผู้ให้บริการธนาคาร

5. ทดสอบระบบย่อย

6. ทดสอบระบบรวม

7. ทำคู่มือการใช้งาน



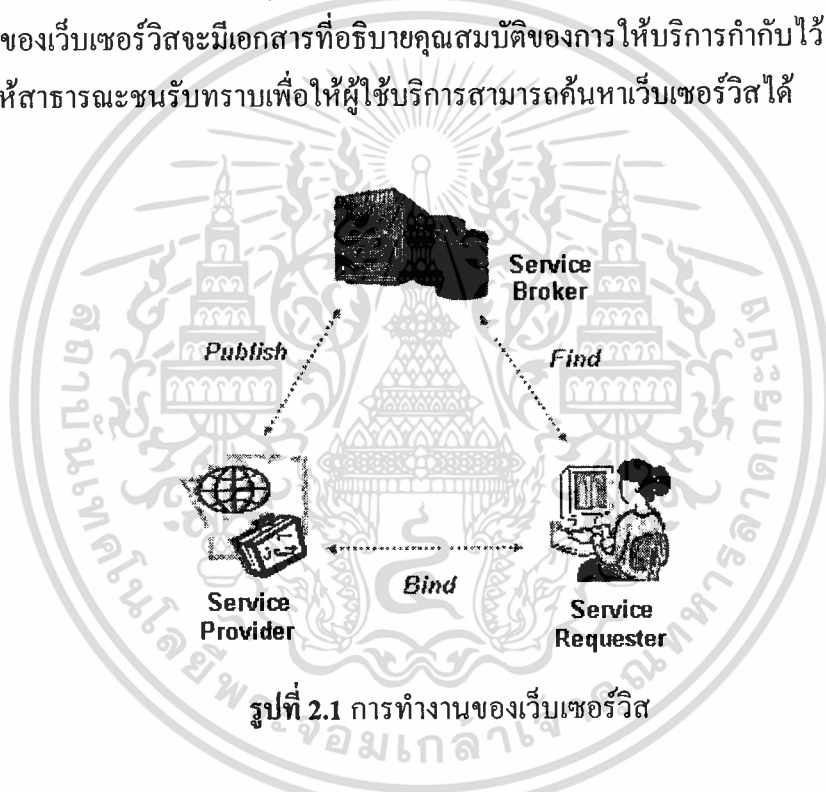
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 เว็บเซอร์วิส (สราวุธ อ้อยศรีสกุล. 2544:36)

เว็บเซอร์วิส (Web Services) คือ แอปพลิเคชันหรือโปรแกรมซึ่งทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากแอปพลิเคชันหรือโปรแกรมอื่นๆผ่านเว็บ การให้บริการของเว็บเซอร์วิสจะมีเอกสารที่อธิบายคุณสมบัติของการให้บริการกำกับไว้ และมีการนำเสนอให้สาธารณะชนรับทราบเพื่อให้ผู้ใช้บริการสามารถค้นหาเว็บเซอร์วิสได้



รูปที่ 2.1 การทำงานของเว็บเซอร์วิส

การทำงานโดยรวมของเว็บเซอร์วิสประกอบด้วยการทำงานร่วมกันของ 3 องค์ประกอบ ดังนี้

- ผู้ให้บริการ (Service provider) ทำหน้าที่เป็นเซิร์ฟเวอร์ให้บริการเว็บเซอร์วิสคอมพิวเตอร์ โดยที่คอมพิวเตอร์ที่จะให้บริการ จะมีการเผยแพร่ (Publish) ลักษณะการให้บริการและวิธีการเรียกใช้ เช่น เอกสาร WSDL ไปลงทะเบียนไว้กับนายหน้าซื้อขาย (Service broker)
- ผู้ขอบริการ (Service requester) ทำหน้าที่เป็น Client เรียกใช้บริการจากผู้ให้บริการ ถ้าหากยังไม่รู้วิธีการเรียกใช้ หรือลักษณะการให้บริการ ก็สามารถทำการค้นหา (Find) ได้จากนายหน้าซื้อขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- นายหน้าซื้อขาย (Service broker) ทำหน้าที่เป็นตัวแทน หรือเครื่องมือในการค้นหาเว็บเซอร์วิสที่มีการลงทะเบียนไว้เพื่อให้บริการ

เว็บเซอร์วิสจะทำให้การทำงานมีประโยชน์และมีประสิทธิภาพมากขึ้น โดยโปรแกรมของเว็บเซอร์วิสจะมีลักษณะหน่วยย่อยที่มีการกำหนดโครงสร้างไว้อย่างดี และซ่อนฟังก์ชันการทำงานภายในไว้และเปิดให้โปรแกรมอื่นเรียกใช้งานได้เฉพาะบางส่วน ซึ่งทำให้เกิดการทำงานร่วมกันระหว่างโปรแกรมหรือระบบที่เป็นไปในลักษณะต่อกันหลวมๆ (Loosely Coupled)

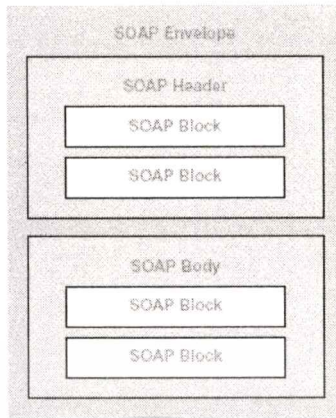
2.1.1 Simple Object Access Protocol (SOAP) (สราวุธ อ้อยศรีสกุล. 2544:48)

SOAP คือ โพรโทคอลที่ใช้ในการเข้าถึงอ็อบเจกต์ด้วยวิธีการง่ายๆ โดยใช้ไวยากรณ์ของภาษา XML ซึ่งเป็นข้อความธรรมดาไม่ขึ้นกับแพลตฟอร์ม โดยเราเรียกสิ่งที่ส่ง-รับระหว่างผู้รับและผู้ส่งนี้ว่า SOAP Message มีการนำ SOAP ไปใช้ในระบบหลายรูปแบบตั้งแต่ระบบส่งข้อความ (Messaging System) จนถึง การเรียกกระบวนการระยะไกล (Remote Procedure Call หรือ RPC)

SOAP ประกอบด้วยส่วนต่างๆดังต่อไปนี้

- SOAP Envelope เปรียบเสมือนซองจดหมาย ประกอบด้วย SOAP Header และ SOAP Body อยู่ภายใน โดยสิ่งที่อยู่ใน SOAP Body คือข้อมูลที่ต้องใช้ในการประมวลผลเว็บเซอร์วิสหรือเปรียบเทียบอาร์กิวเมนต์
- SOAP Encoding Rules คือข้อกำหนดการเข้ารหัสลับของข้อมูล SOAP เป็นส่วนของกลไกที่ใช้ในการแลกเปลี่ยนชนิด โครงสร้างของข้อมูลที่เรียกใช้
- SOAP RPC เป็นการนิยามรูปแบบการเรียกใช้งานการเรียกกระบวนการระยะไกล และการตอบกลับ
- SOAP Binding เป็นข้อกำหนดการแลกเปลี่ยน SOAP Envelope ระหว่างกันภายใต้โพรโทคอลที่ใช้ในการสื่อสารที่มีอยู่เดิมในระบบปัจจุบัน ซึ่งในปัจจุบันทำงานอยู่บนโพรโทคอล HTTP

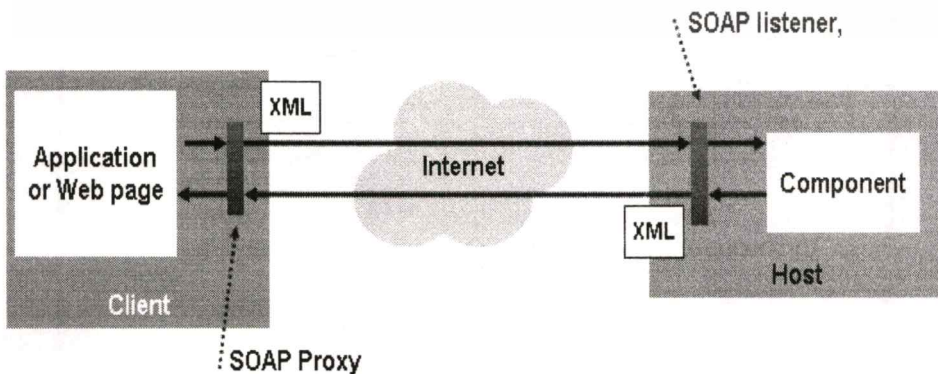
SOAP ทำให้เราสามารถเรียกใช้งานเว็บเซอร์วิสหรือคอมโพเนนต์ข้ามเครื่องได้ SOAP ยังสามารถใช้งานร่วมกับโพรโทคอลอื่นๆได้ด้วยเช่น HTTP และ SMTP ซึ่งจากการที่ใช้งานร่วมกับ HTTP ได้นั้นทำให้ SOAP Message ผ่านระบบไฟร์วอลล์ได้ การทำงานของ SOAP จะมี SOAP Listener เป็นตัวคอยรับฟัง หากมีการเรียกใช้งานเข้ามาก็จะไปเรียกให้เว็บเซอร์วิสหรือคอมโพเนนต์นั้นทำงาน



```

<!-- SOAP request packet -->
<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:GetProductPrice xmlns:m="MyCompany-URI">
<productId>123456</productId>
</m:GetProductPrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
-----
<!-- SOAP response packet -->
<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:GetProductPriceResponse xmlns:m="MyCompany-URI">
<Price>99.95</Price>
</m:GetProductPriceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
    
```

รูปที่ 2.2 ตัวอย่าง SOAP Message



รูปที่ 2.3 SOAP Listener

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Web Services Description Language (WSDL) (สราวุธ อ้อยศรีสกุล. 2544:48)

WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะของเว็บเซอร์วิสและวิธีการติดต่อกับเว็บเซอร์วิส โดยใช้ไวยากรณ์ของภาษา XML ที่ได้รับการควบคุมมาตรฐานโดย W3C มีลักษณะคล้ายเป็นภาษานิยามส่วนต่อประสาน (Interface Definition Language) เพื่อแสดงคุณสมบัติของเว็บเซอร์วิสว่าเป็นของเซอร์วิสใด ให้บริการอะไรบ้าง เรียกใช้งานอย่างไร โดยในไฟล์แต่ละไฟล์ประกอบด้วยจุดสำคัญที่ควรรู้อย่างนี้

- จุดที่ 1 บอกชื่อเว็บเซอร์วิส โดยจะประกาศอยู่ในแอตทริบิวต์ name ของเอลิเมนต์ <service> ซึ่งในที่นี้คือ StockQuoteService
- จุดที่ 2 บอกชื่อเมธอดของเว็บเซอร์วิส ซึ่งเว็บเซอร์วิสจะมีให้บริการกี่เมธอดก็ได้ ประกาศอยู่ในแอตทริบิวต์ name ของเอลิเมนต์ <operation> ในตัวอย่างนี้คือ getQuote
- จุดที่ 3 และ จุดที่ 4 บอกข้อมูลพารามิเตอร์ของเมธอดและค่าที่ส่งกลับมาของเว็บเซอร์วิส โดยปกติแล้วในไฟล์ WSDL จะมีเอลิเมนต์ <message> เป็นคู่ๆ ซึ่งก็คือข้อมูลพารามิเตอร์ของเมธอดที่ส่งไปให้เว็บเซอร์วิสประมวลผล และข้อมูลผลลัพธ์ที่ส่งกลับมา สำหรับในตัวอย่างนี้ข้อมูลพารามิเตอร์เป็นชนิด string (เพราะเมธอด getQuote จะให้ใส่พารามิเตอร์เป็นชื่อย่อของบริษัทที่ต้องการทราบราคาหุ้น) ส่วนข้อมูลที่ส่งกลับเป็นชนิด float (ตัวเลขดัชนีราคาหุ้นซึ่งเป็นเลขทศนิยม)
- จุดที่ 5 บอกตำแหน่งของไฟล์ซึ่งทำหน้าที่เป็น SOAP Listener โดยไฟล์ SOAP Listener จะทำหน้าที่คอยรับฟังว่ามีการเรียกใช้เว็บเซอร์วิสหรือไม่ ถ้ามีก็จะไปปลุกให้เว็บเซอร์วิสนั้นทำงาน โดยตำแหน่งของไฟล์นี้จะบอกด้วยแอตทริบิวต์ location ในเอลิเมนต์ <address> ตามทฤษฎีแล้วไฟล์เอกสาร WSDL แต่ละไฟล์สามารถอธิบายคุณลักษณะของบริการเว็บเซอร์วิสได้มากกว่า 1 บริการ โดยแต่ละเว็บเซอร์วิสจะมีพอร์ตสื่อสารเฉพาะตัว

```

<?xml version="1.0" ?>
<definition name="StockQuoteService" targetNamespace="http://www.xmethods.net/sd/StockQuoteService.wsdl"
xmlns:tns="http://www.xmethods.net/sd/StockQuoteService.wsdl" xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">
- <message name="getQuoteRequest">
  <part name="symbol" type="xsd:string" />
</message>
- <message name="getQuoteResponse">
  <part name="Result" type="xsd:float" />
</message>
- <portType name="StockQuotePortType">
  - <operation name="getQuote">
    <input message="tns:getQuoteRequest" name="getQuote" />
    <output message="tns:getQuoteResponse" name="getQuoteResponse" />
  </operation>
</portType>
- <binding name="StockQuoteBinding" type="tns:StockQuotePortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  - <operation name="getQuote">
    <soap:operation soapAction="" />
    - <input>
      <soap:body use="encoded" name="urn:xmethods-delayed-quotes"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding" />
    </input>
    - <output>
      <soap:body use="encoded" namespace="urn:xmethods-delayed-quotes"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding" />
    </output>
  </operation>
</binding>
- <service name="StockQuoteServices">
  <documentation>Obtains 20-minute delayed quotes from Yahoo</documentation>
  - <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://services.xmethods.net:80/soap" />
  </port>
</service>
</definition>

```

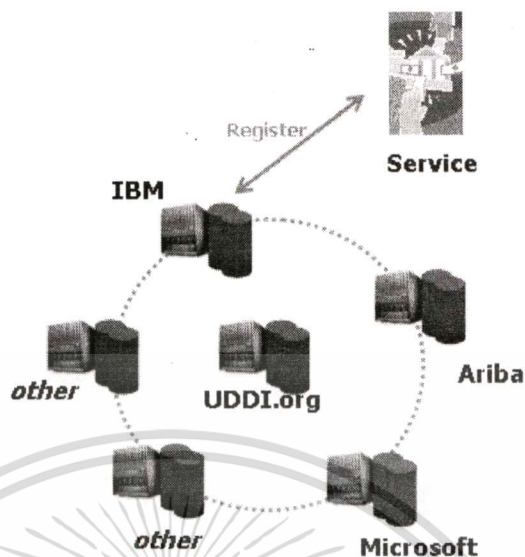
รูปที่ 2.4 WSDL

2.1.3 Universal Description Discovery and Integration (UDDI) (สราวุธ อ้อยศรีสกุล,

2544:52)

UDDI เป็นฐานข้อมูลขนาดใหญ่ที่เก็บรายชื่อของเซอร์วิสต่างๆไว้ เปรียบเสมือนเป็นแหล่งค้นหาข้อมูล โดยผู้ที่ต้องการเปิดให้บริการของตัวเองก็จะมาลงทะเบียนไว้ที่นี้ ส่วนผู้ที่ต้องการเรียกใช้บริการอะไรก็จะเข้ามาค้นหารายชื่อจากที่นี้เช่นกัน ซึ่งจะเป็ประโยชน์แก่ผู้พัฒนาแอปพลิเคชัน เนื่องจากจะได้ไม่ต้องพัฒนาสิ่งที่มีอยู่แล้ว แต่จะสามารถเรียกใช้งานได้ตามข้อตกลงวิธีการทำงานของ UDDI นั้น ก็จะใช้วิธีการแลกเปลี่ยนข้อมูล หาข้อมูลและที่อยู่ของเว็บเซอร์วิสกับ UDDI ด้วยกัน โดยอาศัยความสามารถในการทำงานข้ามระบบปฏิบัติการของ SOAP และ XML เป็นพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 UDDI

2.1.4 Extensible Markup Language (XML) (Sun Microsystems, 2004b)

XML คือภาษาที่ใช้แสดงผลข้อมูล ซึ่งเป็นมาตรฐานและไม่ขึ้นกับแพลตฟอร์ม โดยจะเป็นข้อความธรรมดาล้อมด้วยแท็กที่มีลักษณะเป็นโครงสร้างและเราสามารถกำหนดแท็กต่างๆ เหล่านี้ได้ ซึ่งแต่ละแท็กนั้นสามารถตีความได้โดยใช้ XML Parser

เอกสารที่ใช้กำหนดว่าภายในไฟล์ XML ประกอบด้วยแท็กอะไรได้บ้างนั้นเราเรียกว่าเค้าร่าง XML (XML Schema) โดยภาษาที่ใช้เขียนเค้าร่างนั้นคือ DTD (Document Type Definition Language) ดังรูป 2.6 และรูป 2.7 ซึ่งจะแสดงให้เห็นว่าใน confirm.dtd มีการกำหนดไว้ว่า Confirmation เป็นเอลิเมนต์ราก ซึ่งจะประกอบด้วยเอลิเมนต์ย่อยคือ OrderID และ ship-date

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT confirmation (orderID, ship-date)>
<!ELEMENT orderID (#PCDATA)>
<!ELEMENT ship-date (#PCDATA)>
```

รูปที่ 2.6 ตัวอย่างไฟล์ DTD (confirm.dtd)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<!--
  Copyright 2003 Sun Microsystems, Inc. All rights reserved.
  SUN PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.
-->

<!DOCTYPE confirm
  PUBLIC "-//Sun Microsystems, Inc.//DTD confirmation//EN"
  "http://java.sun.com/webservices/dtds/confirm.dtd">

<confirmation>
  <orderID>
    "1234"
  </orderID>

  <ship-date>
    "01-01-2001"
  </ship-date>
</confirmation>

```

รูปที่ 2.7 ตัวอย่างไฟล์ XML (confirm.xml)

การกำหนดเค้าร่างไว้จะทำให้สามารถทำไฟล์ XML ไปใช้ที่ไหนก็ได้ (Data Portable) เช่น ดังรูป 2.6 และ 2.7 ถ้าแอปพลิเคชันส่งไฟล์ confirm.xml ไปและมีไฟล์ confirm.dtd เป็นตัวกำหนด โครงสร้างไว้ จะทำให้ XML Parser ผู้รับสามารถตีความข้อมูลได้

2.2 Java 2 Platform, Enterprise Edition (J2EE™) (นรินทร์ โอพารากิจนันต์. 2547:9)

Java 2 Platform, Enterprise Edition (J2EE™) คือ กลุ่มของเทคโนโลยีภาษาจาวาที่จำเป็น สำหรับการพัฒนาโปรแกรมประยุกต์ด้วยภาษาจาวาให้ใช้งานได้ในระดับองค์กร

แต่เดิมนั้นภาษาจาวาถูกใช้ในการสร้างโปรแกรมประยุกต์ส่วนบุคคลเป็นหลัก ต่อมาเมื่อจาวาถูกนำไปใช้งานในองค์กรมากขึ้น ผู้พัฒนาจาวาจำเป็นต้องเพิ่มชุดคำสั่งใหม่ๆเข้าไปเพื่อตอบสนองความต้องการใช้งานระดับองค์กร เมื่อชุดคำสั่งที่เพิ่มเข้าไปมีจำนวนมากขึ้นเรื่อยๆ จึงได้มีการจัดระเบียบชุดคำสั่งภาษาจาวาเสียใหม่ โดยแบ่งเทคโนโลยีภาษาจาวาออกเป็น 3 ระดับดังนี้

1. J2SE (Java 2 Standard Edition) หรือ จาวาระดับมาตรฐาน คือ จาวาดั้งเดิมที่มี แต่ชุดคำสั่งปกติสำหรับการสร้างโปรแกรมประยุกต์ส่วนบุคคล
2. J2EE (Java 2 Enterprise Edition) หรือ จาวาระดับองค์กร คือ จาวาที่มีชุดคำสั่งการสร้างโปรแกรมประยุกต์ให้ใช้งานได้ในระดับองค์กร
3. J2ME (Java 2 Micro Edition) หรือ จาวาระดับจิ๋ว คือ จาวาที่มีชุดคำสั่งเพื่อการสร้างโปรแกรมประยุกต์สำหรับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

J2EE ถือได้ว่าเป็นส่วนขยายของ J2SE ด้วย กล่าวคือ J2EE รวมเอาทั้งหมดของ J2SE เข้าไปในตัวมันแล้วเพิ่มชุดคำสั่งที่จำเป็นสำหรับการพัฒนาโปรแกรมประยุกต์ให้ใช้งานได้ในระดับองค์กรเข้าไป

ตารางที่ 2.1 ชุดคำสั่งใน J2EE

ชื่อชุดคำสั่ง	หน้าที่และประโยชน์ใช้สอย
RMI-IIOP	ใช้สำหรับการสื่อสารหรือการเรียกเมธอดทางไกลผ่านระบบเครือข่าย
Java IDL	ภาษากลางสำหรับสื่อสารกับโปรแกรมประยุกต์ระดับองค์กรที่พัฒนาขึ้นด้วยเทคโนโลยีอื่น
JNDI	ระบบการเรียกชื่อของบริการหรือวัตถุที่มีอยู่บนระบบเครือข่าย
JDBC	ใช้เชื่อมการติดต่อกับฐานข้อมูล
JMS	ใช้ประสานงานหรือสื่อสารระหว่างโปรแกรมประยุกต์ด้วยกัน
Java Mail	ใช้ติดต่อกับเมลเซิร์ฟเวอร์ในองค์กร
Java Servlets	บริการเว็บบนฝั่งเซิร์ฟเวอร์
JSP	บริการเว็บบนฝั่งเซิร์ฟเวอร์ที่ตัดแปลงมาจาก Java Servlets อีกทีหนึ่งโดยมีความง่ายในการพัฒนามากกว่า
JTA	ชุดคำสั่งสำหรับการบริหารจัดการธุรกรรม
JTS	บริการธุรกรรม
EJB	โครงสร้างโปรแกรมประยุกต์แบบคอมโพเนนท์
JAXP	ชุดคำสั่งสำหรับจัดการไฟล์ XML
JAAS	ชุดคำสั่งสำหรับระบบรักษาความปลอดภัย
JCA	ใช้ติดต่อกับระบบอื่นๆที่มีอยู่ในเครือข่ายขององค์กร

2.2.1 Java Application Server (นรินทร์ โอพาทกิจอนันต์. 2547:10)

การนิยามคำว่า J2EE ขึ้นมาไม่ได้เป็นแค่การจัดกลุ่มชุดคำสั่งให้เป็นระบบเท่านั้น แต่ได้มีการสร้างนวัตกรรมของการพัฒนาโปรแกรมระดับองค์กรด้วย กล่าวคือมีการสร้างแนวคิดเรื่อง Application Server ขึ้นมา แต่เดิมการเขียนโปรแกรมประยุกต์ให้ใช้งานได้ระดับองค์กรเป็นเรื่องยากมาก เพราะนอกจากจะต้องเขียนคำสั่งให้โปรแกรมทำงานได้อย่างที่เราต้องการได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเมอร์ระดับองค์กรยังต้องเขียนคำสั่งให้โปรแกรมทำงานได้ในระบบเครือข่ายอีกด้วย ซึ่งเป็นเรื่องที่ยากมาก

แนวคิดเรื่องการพัฒนา Application Server ทำให้การพัฒนาโปรแกรมประยุกต์ระดับองค์กรทำได้ง่ายขึ้น เพราะ Application Server จะรับผิดชอบเกี่ยวกับการจัดการเครือข่ายทั้งหมดแทนเรา ทำให้โปรแกรมเราทำงานบนเครือข่ายคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ ถือได้ว่า Application Server เป็นหัวใจสำคัญของ J2EE

โครงสร้างของ Java Application Server

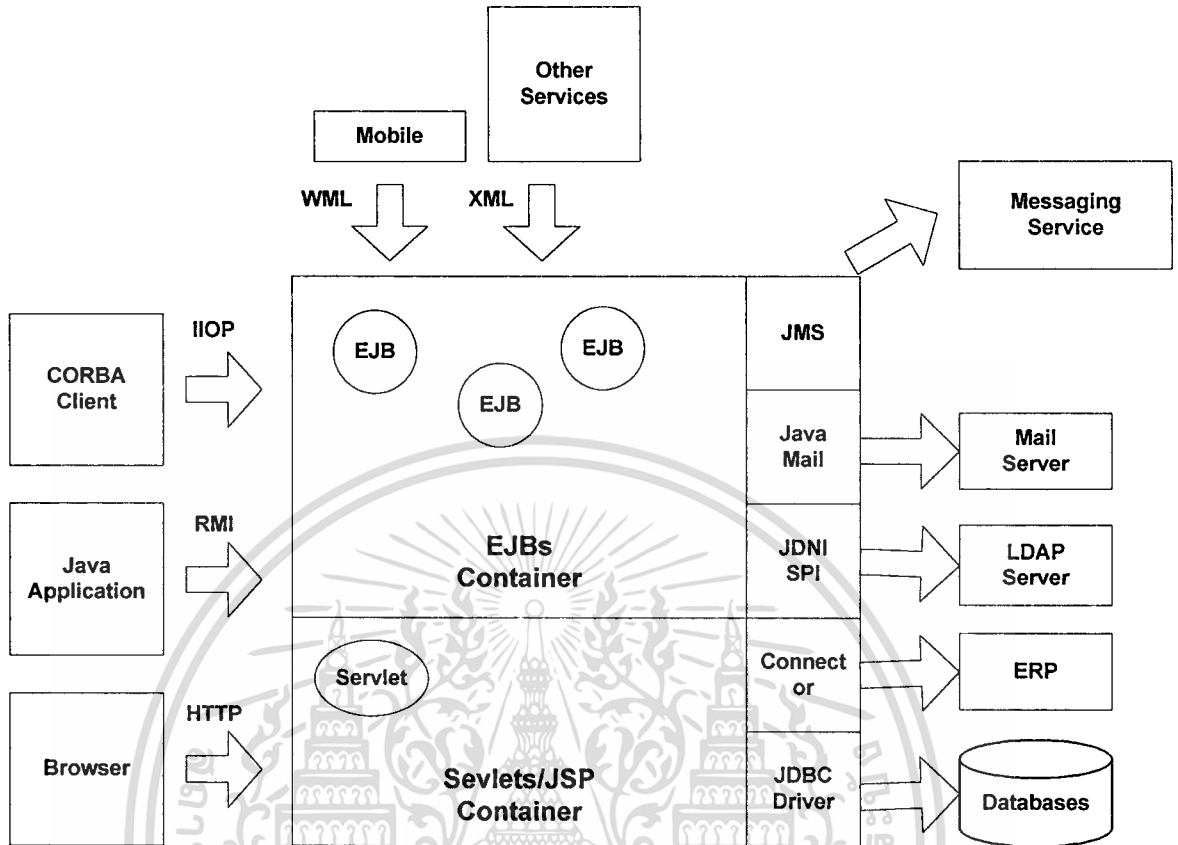
Java Application Server มีลักษณะคล้ายกับ Java Virtual Machine ในจาวาระดับมาตรฐาน กล่าวคือเป็นสิ่งที่โปรแกรมภาษาจาวาที่เราเขียนขึ้นทำงานบนตัวมันอีกที อาจกล่าวได้ว่า Java Application Server เป็นตัวที่ห่อหุ้มโปรแกรมภาษาจาวาระดับองค์กรที่เราเขียนเอาไว้จากสภาวะแวดล้อมภายนอก มันทำหน้าที่เป็นตัวกลางในการติดต่อสื่อสารกับภายนอกแทนโปรแกรมของเราแบบเดียวกับที่ Java Virtual Machine ที่คั่นกลางระหว่างโปรแกรมภาษาจาวาของเรากับระบบปฏิบัติการของเครื่อง

จาวาแอปพลิเคชันเซิร์ฟเวอร์ สามารถทำงานได้มากกว่า Java Virtual Machine เพราะในระดับองค์กร นอกจากโปรแกรมประยุกต์จะต้องติดต่อกับระบบปฏิบัติการแล้ว มันยังต้องติดต่อกับสิ่งอื่นอีกมากมาย ซึ่งส่วนมากเป็นการติดต่อผ่านทางระบบเครือข่ายขององค์กร เช่น บางครั้งโปรแกรมต้องดึงข้อมูลจากฐานข้อมูลที่อยู่บนเครื่องคอมพิวเตอร์อีกหนึ่งที่ห่างออกไปบนเครือข่ายองค์กร เป็นต้น ตัวอย่างของผลิตภัณฑ์สำเร็จรูปที่ผลิตออกมา เช่น WebLogic ของ BEA, WebSphere ของ IBM, Sun Java System Application Server ของ Sun

คุณสมบัติเด่นของ Java Application Server

- สนับสนุนมาตรฐาน COBRA
- สนับสนุนแนวคิดของ Web Application
- ใช้หลักการของคอม โพนেন্ট
- สนับสนุนแนวคิดของบริการผ่านเว็บ (Web Services)
- มีการบริหารหน่วยความจำที่มีประสิทธิภาพ
- มีการจัดการด้านธุรกรรมข้อมูล
- มีระบบรักษาความปลอดภัยและระบบจัดการผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 โครงสร้างของ Java Application Server

2.2.2 Java Servlets (นรินทร์ โอพารกิจอนันต์. 2547:215)

Java Servlets เป็นชุดคำสั่งของ J2EE ซึ่งใช้ในการสร้างโปรแกรมบนฝั่ง Server ที่ใช้ Browser เป็น Client เป็นการใช้ภาษาจาวาในการสร้างโปรแกรมที่มีลักษณะเป็น Web Application นั่นเอง

โปรแกรมบนฝั่ง Server ที่เขียนด้วย Java Servlets เราเรียกว่า Servlet คำว่า Servlet มีความหมายว่า Server ขนาดเล็ก ซึ่งเป็นการตั้งชื่อให้สอดคล้องกับ Java Applet ซึ่งเป็น โปรแกรมภาษาจาวาขนาดเล็กที่ทำงานบนฝั่ง Client

Servlet ไม่ได้รันอยู่บน Java Virtual Machine แบบที่โปรแกรมภาษาจาวาทัวไปใช้ แต่มันรันอยู่บน Java Virtual Machine พิเศษที่เรียกว่า Servlet Container ซึ่งมีคุณสมบัติพิเศษคือทำหน้าที่เกี่ยวกับการติดต่อสื่อสารกับ Browser ให้กับ Servlets Servlet Container มีทั้งแบบที่สมบูรณ์ในตัวมันเองและแบบที่ต้องเกาะอยู่บน Web Server อีกที ตัวอย่างที่ได้รับความนิยมมากที่สุดคือ Tomcat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

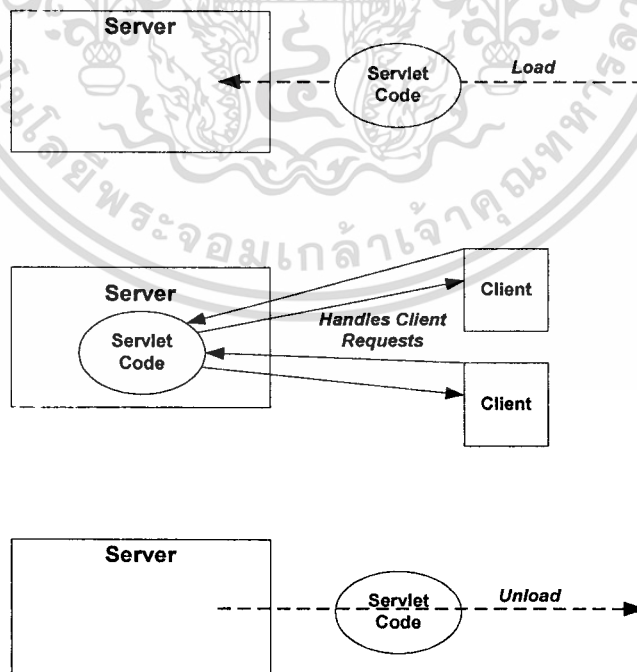
หลักการการทำงานของ Servlet

หลังจากที่เราติดตั้ง Servlet ไว้ใน Servlet Container แล้ว มันจะยังไม่ถูกโหลดเข้าไปในหน่วยความจำหลักของเครื่อง การโหลด Servlet ลงหน่วยความจำจะเกิดขึ้นก็ต่อเมื่อมี Browser เรียก Servlet นั้นเป็นครั้งแรก และเมื่อมีการโหลด Servlet ลงหน่วยความจำแล้ว มันจะค้างอยู่ในหน่วยความจำตลอดเวลาที่ Servlet Container ยังทำงานอยู่ เวลาที่มี Browser เรียก Servlet นั้นซ้ำ มันจะทำงานได้เลยไม่ต้องมีการโหลดใหม่ ซึ่งทำให้การเข้าถึง Servlet ครั้งต่อไปทำงานได้เร็วขึ้น

2.2.3 JavaServer Pages (JSP)(นรินทร์ โอพารากิจอนันต์. 2547:235)

JavaServer Pages คือการใช้ภาษาจาวาในการสร้าง Web Application ที่มีเนื้อหาไม่ตายตัว โดยใช้กลไกสำหรับแทรก Code ภาษาจาวาลงใน html page ซึ่งเมื่อตอนถูกเรียกใช้ page นั้นจะถูกเปลี่ยนไปเป็น Servlet ดังนั้น JSP จะสามารถรับ HTTP requests ได้เช่นเดียวกับ Servlet แต่การสร้าง JSP นั้นสามารถสร้างได้ง่ายกว่าการสร้าง Servlet

โดยทั่วไปเรามักให้ JSP ทำหน้าที่ presentation logic ส่วนงาน business logic ถูกส่งต่อไปให้ enterprise javabeans ทำ



รูปที่ 2.9 วงจรชีวิตของ Servlets

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 Enterprise JavaBean (EJB) (นรินทร์ โอพารกิจอนันต์. 2547:28)

Enterprise JavaBean (EJB) หรือ บีน คือโปรแกรมภาษาจาวาขนาดเล็ก (คอมโพเนนต์) บนเครื่อง Server ที่เขียนขึ้นมาเพื่อนำมาใช้งานเฉพาะเจาะจงอย่างใดอย่างหนึ่ง บีนมีความเป็นอิสระในตัวเอง สามารถทำงานอยู่อย่างโดดเดี่ยวบน Server หรือทำงานประสานกับบีนด้วยกันเองก็ได้

ประเภทของบีน

1. บีนบริการ (Session Bean)

ถูกสร้างขึ้นมาเพื่อทำงานบางอย่างให้กับผู้ใช้ฝั่ง Client ที่ติดต่อเข้ามายัง Server แบ่งเป็น

- บีนบริการแบบคงสถานะ (Stateful Bean) คืออินสแตนซ์ของบีนแบบนี้ จะถูกสร้างขึ้นเพื่อจัดการกับผู้ใช้ที่เข้ามาแบบตัวต่อตัว สามารถจดจำสถานะต่างๆของผู้ใช้ได้
- บีนบริการแบบไม่คงสถานะ (Stateless Bean) คือ บีนบริการที่ไม่มีการจดจำสถานะต่างๆของผู้ใช้ที่เข้ามา ดังนั้นในเวลาเดียวกันบีนบริการแบบไม่คงสถานะสามารถให้บริการผู้ใช้ได้มากกว่า 1 คน

2. บีนวัตถุ (Entity Bean)

บีนวัตถุ คือ บีนที่สามารถคงสถานะของบีนไว้ได้ตลอดไป แม้บีนจะถูกทำลายไปแล้ว

โดย Server จะอาศัยฐานข้อมูลในการเก็บสถานะของบีนเหล่านี้ แบ่งออกเป็น 2 แบบคือ

- บีนวัตถุแบบที่บีนเป็นผู้จัดการ (Bean-Managed Persistence) โดยโปรแกรมเมอร์จะต้องเป็นผู้เขียนคำสั่งติดต่อกับฐานข้อมูลเอง
- บีนวัตถุแบบที่คอนเทนเนอร์เป็นผู้จัดการ (Container-Managed Entity Bean)

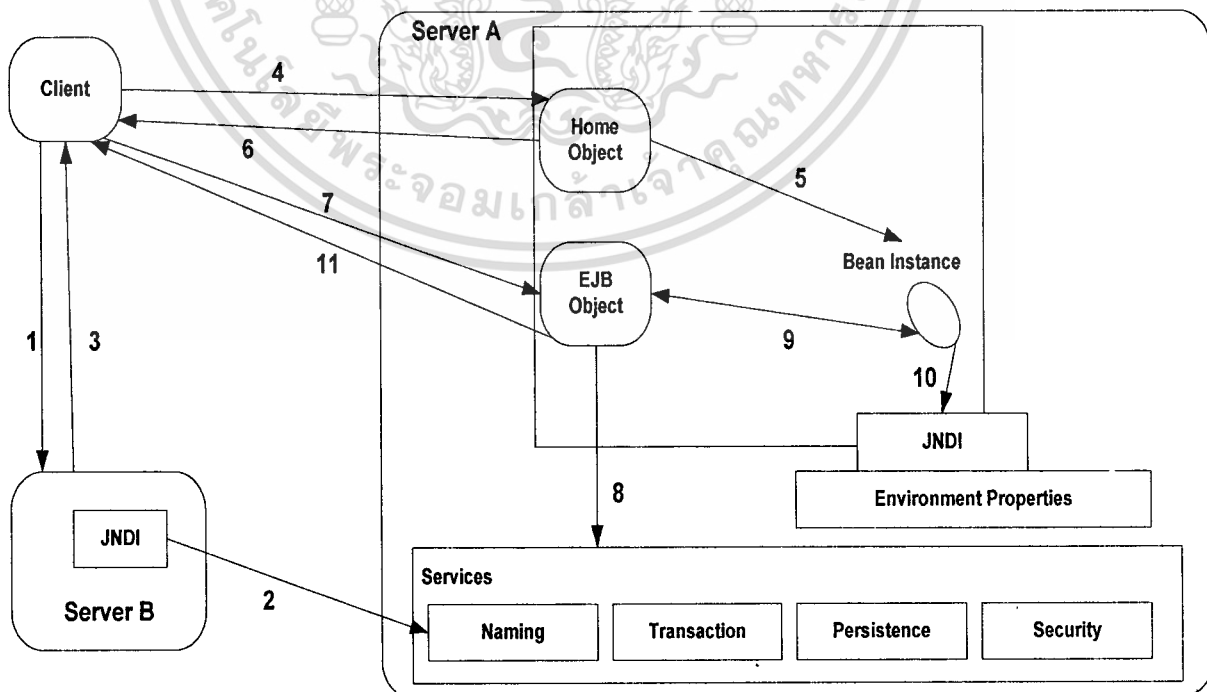
3. บีนที่ทำงานด้วยแมสเสจ (Message-Driven Bean) โดยจะทำงานก็ต่อเมื่อได้รับ Message จาก Message-oriented-Middleware (ตัวกลางที่ช่วยในการทำให้คอมพิวเตอร์ต่างระบบสามารถคุยกันได้)

สำหรับบีนบริการและบีนวัตถุ ทุกๆคอมโพเนนต์จะต้องประกอบด้วย

1. Home Interface ใช้ประกาศเมธอดมาตรฐานที่ Client เรียกใช้เวลาติดต่อเข้ามา
2. Interface ใช้เป็นตัวกลางให้ Client เรียก business ของบีน

ขั้นตอนการเรียกใช้

1. Client ติดต่อ Server B เพื่อขอ reference ของ home object ของ ejb ที่จะใช้งาน
2. Server B ติดต่อ Server A เพื่อขอ reference ของ home object ซึ่งถูกทำ binding ไว้กับ JNDI ตอนที่ deploy
3. Server B ส่ง reference ไปให้ Client
4. Client ใช้ reference ที่ได้มาเรียกไปที่ home object เพื่อสั่งให้สร้าง bean instance
5. home object ทำการสร้าง bean instance
6. home object ส่ง reference ของ ejb object สำหรับ bean instance นั้นให้ Client
7. Client เรียก business method ไปที่ ejb object
8. ejb object แทรกบริการของ ejb container เข้าไปในการเรียกเมธอดนั้น
9. ejb object เรียกไปที่เมธอดของ bean instance นั้น
10. ระหว่างที่ bean instance ทำงานอาจต้องการสอบถามค่าเกี่ยวกับสถานะแวดล้อมการทำงานจาก name server
11. ejb object ส่งผลลัพธ์ที่ได้ไปให้ Client



รูปที่ 2.10 การเรียกใช้งาน EJB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Java API for XML (Sun Microsystems. 2004b)

Java API for XML เป็น API ภาษาจาวาที่ใช้ในการติดต่อกับเอกสาร XML แบ่งออกเป็น 2 ประเภทคือ

- Document-Oriented คือ API ที่ใช้ในการจัดการ โดยตรงกับเอกสาร XML เช่น JAXP, JAXB และ SAAJ
- Procedure-Oriented คือ API ที่ใช้ในการจัดการกระบวนการ เช่น JAX-RPC และ JAXR

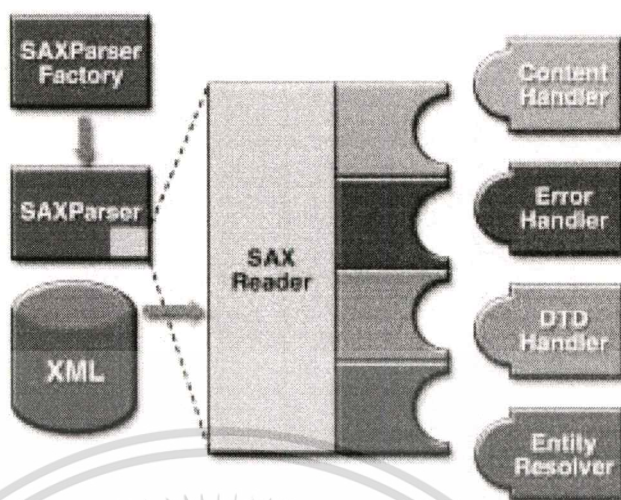
ลักษณะเด่นของ Java API for XML คือสามารถทำงานร่วมกับมาตรฐานอุตสาหกรรม ซึ่งได้ถูกกำหนดขึ้นมาก่อนแล้ว โดยกลุ่มหน่วยงานที่กำหนดมาตรฐานต่างๆ เช่น World Wide Web Consortium (W3C) และ Organization for the Advancement of Structured Information Standards (OASIS) นอกจากนี้แล้ว Java API for XML จะมีความยืดหยุ่นในการใช้งาน เช่น โค้ด JAXP สามารถถูกเรียกใช้งานได้หลากหลายจากเครื่องมือประมวลผล XML ซึ่งทำให้ผู้พัฒนาสามารถเขียนโปรแกรมได้เหมาะสมพอดีกับระบบของตน

2.3.1 JAXP

Java API for XML Processing เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาให้ทำการประมวลผลข้อมูลของเอกสาร XML ได้ โดยสามารถใช้งานร่วมกับ SAX, DOM และ XSLT ดังนี้

- SAX APIs

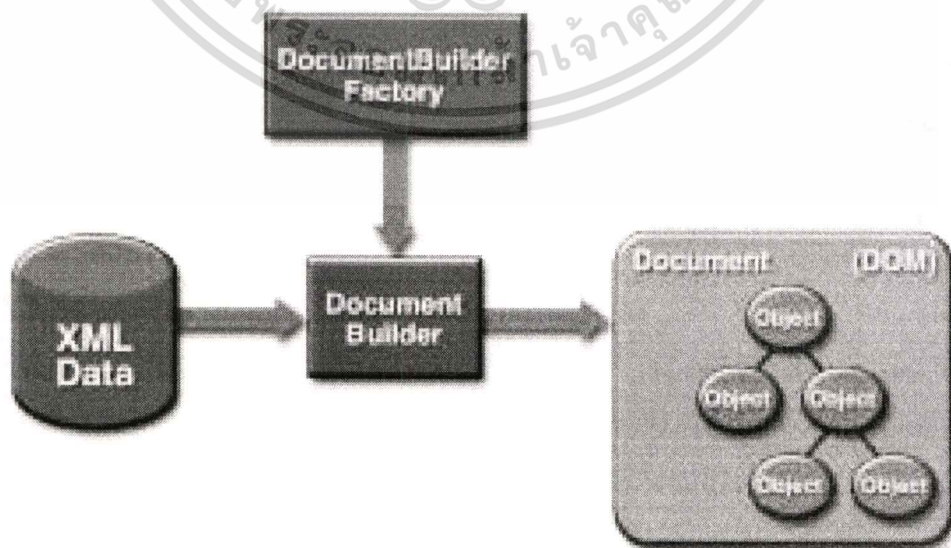
เริ่มจากการสร้าง SAXParser Object จาก SAXParserFactory Object ซึ่งตัว SAXParser Object นี้จะห่อหุ้ม SAXReader Object ไว้ภายใน และเมื่อเรียก SAXParser ทำการเรียกเมธอด parse() ก็จะทำให้ SAXReader Object ไปเรียกเมธอดที่ได้ทำการอิมพลีเมนต์ไว้ ซึ่งเมธอดต่างๆนั้น อิมพลีเมนต์ขึ้นจาก ContentHandler Interface, ErrorHandler Interface, DTDHandler Interface และ EntityResolver Interface ดังรูป 2.10



รูปที่ 2.11 SAX APIs

- DOM APIs
เริ่มจากการใช้ **DocumentBuilderFactory** Object ไป get **DocumentBuilder** Object ออกมา

เพื่อที่จะไปใช้ในการสร้างเอกสารที่สอดคล้องกับข้อกำหนดของ **DOM** (คือการนำ **DocumentBuilder** มาสร้างเอกสารใหม่ที่ทำกรอิมพลีเมนต์ **org.w3c.dom.Document** Interface นั้นเอง) หลังจากนั้นก็เรียกเมธอด **parse()** เพื่อสร้างเป็นเอกสาร โดยใช้ข้อมูล **XML** ที่มีอยู่ ผลที่ได้คือ **DOM Tree** ทั้งนี้ **DOM APIs** ดีกว่า **SAX APIs** ตรงที่เราสามารถแก้ไขและเพิ่ม **XML Node** ได้

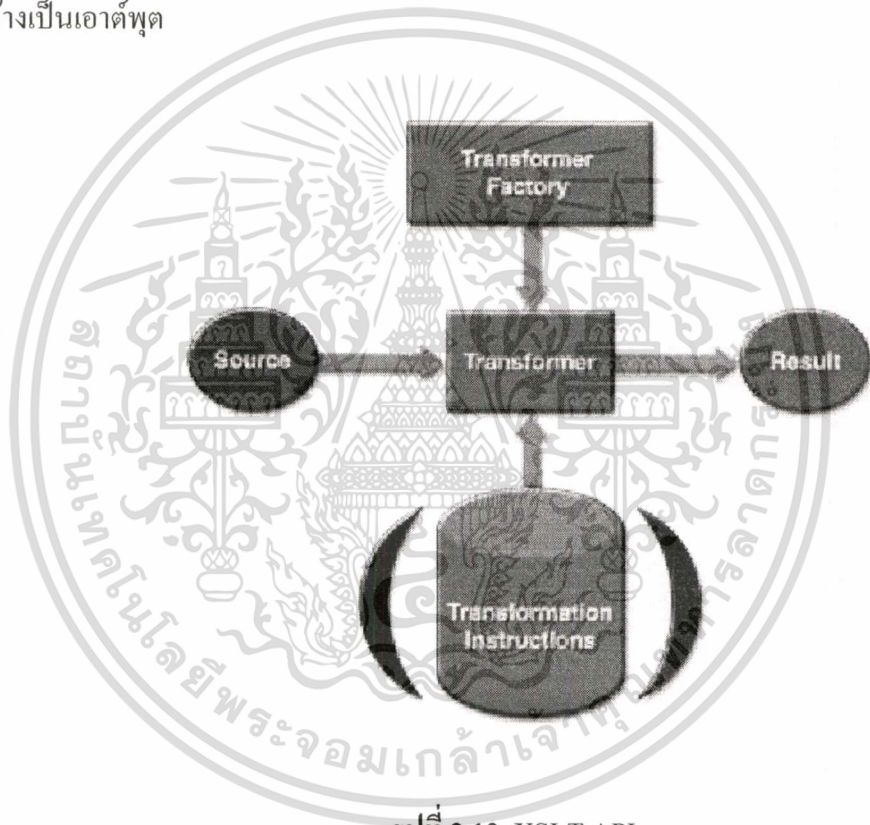


รูปที่ 2.12 DOM APIs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- XSLT APIs

Extensible Stylesheet Language for Transformation หรือ XSLT เป็นภาษาที่ใช้ในการแปลงเอกสาร XML ให้เป็นรูปแบบอื่นๆ การทำงานของ XSLT APIs ของภาษาจาวาเริ่มจากการสร้าง Transformer Object จาก TransformationFactory Object และรับอินพุตเข้ามา ซึ่ง Input นี้ อาจจะมาจากการอ่านของ SAX หรือ DOM ก็ได้ รวมทั้งอาจเป็น Inputstream ก็ได้เช่นกัน เมื่อ Transformer Object เกิดขึ้นมานั้น ก็จะมีการสร้าง Transformation Instructions ด้วย เพื่อใช้ในการทำงานสร้างเป็นเอาต์พุต



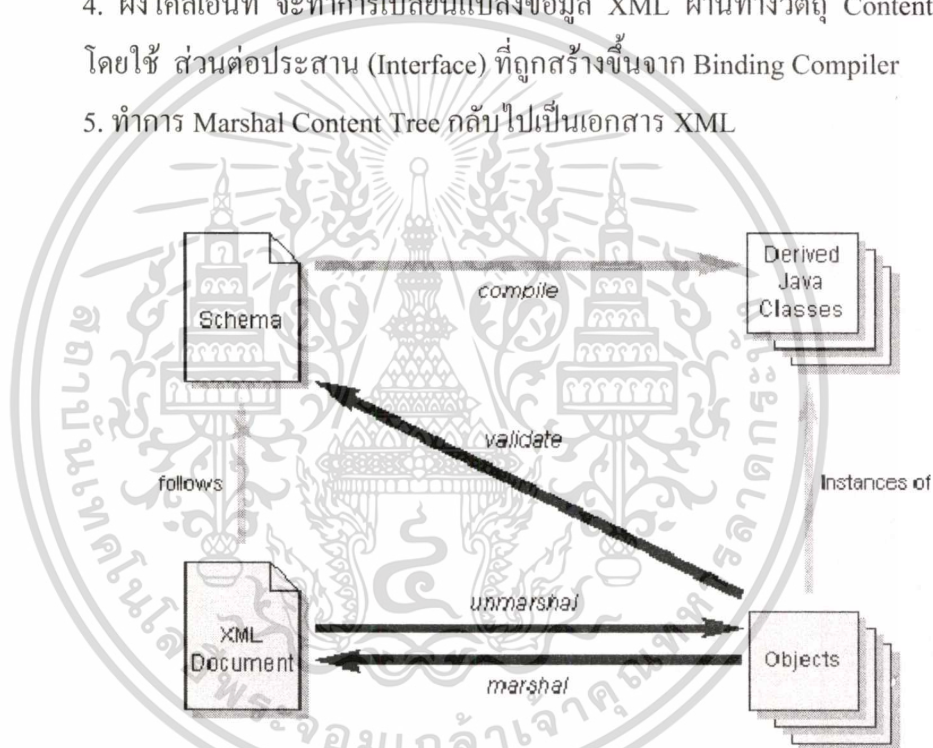
รูปที่ 2.13 XSLT APIs

2.3.2 JAXB

Java APIs for XML Bindings เป็น APIs ที่ใช้ในการเขียนโปรแกรมภาษาจาวาเพื่อทำการยึดเหนี่ยว (binding) ซึ่งก็คือการสร้างจาวาคลาสจากเค้าร่าง XML นั้นเอง โดยการเรียกใช้ เมธอด Unmarshalling และ เมธอด Marshalling ซึ่งเมธอด Unmarshalling คือการเฝ้า จากเอกสาร XML ไปเป็นวัตถุจาวาซึ่งมีลักษณะเป็น Content Tree ส่วนเมธอด Marshalling ก็คือการเฝ้า จากวัตถุ Java Content Tree ไปเป็นเอกสาร XML สำหรับการทำงานของ XSLT APIs นี้จะทำงานโดยซ่อนรายละเอียดการติดต่อกับ SAX และ DOM ทำให้ผู้พัฒนาได้รับความสะดวกเป็นอย่างมาก เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับขั้นตอนในการ bindings มีดังรูป 2.13

1. สร้างจาวาคลาสโดยใช้เค้าร่าง XML และคอมไพล์คลาสนั้นได้เป็นคลาส JAXB
2. ทำการ Unmarshal เอกสาร XML ที่สอดคล้องกับเค้าร่าง XML ในข้อ 1 ไปเป็นวัตถุ Java Content Tree ที่ถูก Instantiated ขึ้นจากคลาส JAXP ในข้อ 1
3. ทำการ Unmarshal ส่วนอื่นๆที่จำเป็น เช่น ส่วนที่เกี่ยวข้องกับการตรวจสอบเอกสาร XML โดยในกรณีที่เรามีการเปลี่ยนแปลงวัตถุ Content Tree จะได้นำเอาข้อมูลส่วนนี้มาใช้ในการตรวจสอบในขั้นตอนการ Marshal ไปเป็นเอกสาร XML
4. ฟังก์ชันโคลเอนท์ จะทำการเปลี่ยนแปลงข้อมูล XML ผ่านทางวัตถุ Content Tree โดยใช้ ส่วนต่อประสาน (Interface) ที่ถูกสร้างขึ้นจาก Binding Compiler
5. ทำการ Marshal Content Tree กลับไปเป็นเอกสาร XML



รูปที่ 2.14 ขั้นตอนในการ Binding

รูปที่ 2.14 แสดงตัวอย่างการ Binding โดยจะเห็นว่ามีการแมปจากเอลิเมนต์ในเค้าร่าง XML ไปเป็นแฟ้มจาวาคลาส เช่น เอลิเมนต์ PurchaseOrder ถูกสร้างเป็น PurchaseOrder.java และ เอลิเมนต์ PurchaseOrderType ถูกสร้างเป็น PurchaseOrderType.java เป็นต้น

โดยภายในแฟ้มจาวาคลาสนั้นจะถูกสร้างเป็นส่วนต่อประสาน และนอกจากนี้แต่ละเอลิเมนต์ย่อยก็จะถูกนำไปสร้างเป็น accessors และ mutators อีกด้วย เช่นเอลิเมนต์ย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

name ที่มี type เป็น string ก็จะถูกนำไปสร้างเป็น เมธอด getName() ที่คืนค่าเป็น String เมธอด setName() ที่รับพารามิเตอร์เป็น String ใน USAddress Interface

XML Schema	Java Class Files
<pre><xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"></pre>	
<pre><xsd:element name="purchaseOrder" type="PurchaseOrderType"/></pre>	PurchaseOrder.java
<pre><xsd:element name="comment" type="xsd:string"/></pre>	Comment.java
<pre><xsd:complexType name="PurchaseOrderType"> <xsd:sequence> <xsd:element name="shipTo" type="USAddress"/> <xsd:element name="billTo" type="USAddress"/> <xsd:element ref="comment" minOccurs="0"/> </xsd:sequence> <xsd:attribute name="orderDate" type="xsd:date"/> </xsd:complexType></pre>	PurchaseOrder- Type.java
<pre><xsd:complexType name="USAddress"> <xsd:sequence> <xsd:element name="name" type="xsd:string"/> <xsd:element name="street" type="xsd:string"/> <xsd:element name="city" type="xsd:string"/> <xsd:element name="state" type="xsd:string"/> <xsd:element name="zip" type="xsd:decimal"/> </xsd:sequence> <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/> </xsd:complexType></pre>	USAddress.java
<pre></xsd:schema></pre>	

```
public interface USAddress {
    String getName();        void setName(String);
    String getStreet();     void setStreet(String);
    String getCity();       void setCity(String);
    String getState();      void setState(String);
    int getZip();           void setZip(int);
    static final String COUNTRY="USA";
};
```

รูปที่ 2.15 ตัวอย่างการ Binding

2.3.3 SAAJ

SOAP with Attachments API for Java เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาเพื่อทำการส่งข้อความ SOAP ไปในเครือข่ายอินเทอร์เน็ต ซึ่ง SAAJ นี้เป็นการทำงานเบื้องหลังของการอิมพลีเมนต์ JAX-RPC และ JAXR โดยทำให้ผู้พัฒนาสามารถเขียนโปรแกรมเพื่อส่งข้อความเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOAP ได้โดยตรง โดยไม่ต้องผ่านกระบวนการ JAX-RPC โดย SAAJ APIs นี้สอดคล้องกับข้อกำหนดของ SOAP เวอร์ชัน 1.1

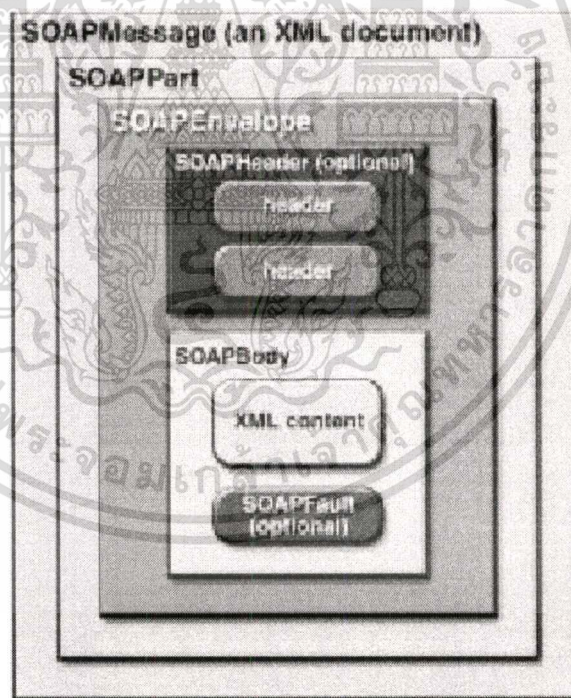
ข้อกำหนดของ SAAJ เวอร์ชัน 1.2 นี้ได้กำหนดไว้ในแพ็คเกจ `javax.xml.soap` ซึ่งประกอบด้วย APIs ที่ใช้ในการสร้างและนำข้อมูลใส่ข้อความ SOAP และการส่ง-รับ สำหรับการอิมพลีเม้นต์นั้นมีคลาสที่ต้องใช้บ่อยๆ 2 ตัว คือ `SOAPConnection` กับ `SOAPMessage`

2.3.3.1 SOAPMessage

มีการแบ่ง ข้อความ SOAP ออกเป็น 2 ประเภทคือ

1. ข้อความ SOAP ที่ไม่มีส่วนผูกติด (SOAPMessage with No Attachments)

โครงสร้างของ SOAPMessage with No Attachments ประกอบด้วยส่วนต่างๆ ดังต่อไปนี้



รูปที่ 2.16 SOAP Message with No Attachments

SAAJ APIs ได้จัดเตรียมคลาสต่างๆเหล่านี้ไว้ให้เรียกใช้งาน ในขั้นตอนแรกจะเป็นการสร้าง วัตถุ SOAPMessage ขึ้นมาก่อน ซึ่ง SOAPMessage นี้ก็จะประกอบด้วย SOAPPart และ SOAPPart เองก็จะประกอบด้วย SOAPEnvelope ตามลำดับ และสำหรับ SOAPEnvelope เองก็

จะมี SOAPHeader กับ SOAPBody อยู่ภายใน ถ้าเราไม่ต้องการใช้ SOAPHeader ก็ลบทิ้งได้

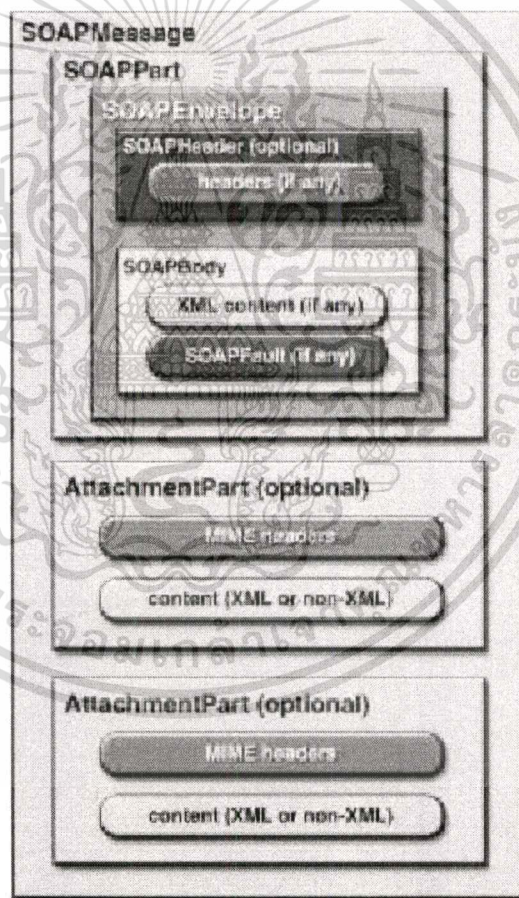
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น เมื่อผู้ดูแลระบบเห็นประโยชน์ในการนำเอกสารนี้ไปใช้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOAPHeader เองก็จะประกอบด้วยส่วนหัว (header) ซึ่งจะเก็บข้อมูลการส่ง-รับข้อความ ส่วน SOAPBody ก็จะบรรจุข้อความที่จะส่งให้ผู้รับข้อความจริงๆ ในกรณีที่มี SOAPFault Message ก็จะถูกบรรจุอยู่ใน SOAPBody นี้เช่นกัน

2. ข้อความ SOAP ที่มีส่วนผูกติด (SOAPMessage with Attachments)

SOAPMessage อาจจะประกอบด้วย AttachmentPart (มีได้มากกว่า 1) โดยภายในอาจบรรจุเนื้อหาที่เป็น XML หรือไฟล์ประเภทอื่น เช่นไฟล์ไบนารี ดังรูปที่ 2.16 SAAJ APIs ได้จัดเตรียมคลาส AttachmentPart ไว้ให้เรียกใช้ โดยจะต้องการสร้าง AttachmentPart และทำการใส่มันลงไป SOAPMessage



รูปที่ 2.17 SOAP Message with Attachements

2.3.3.2 SOAPConnection

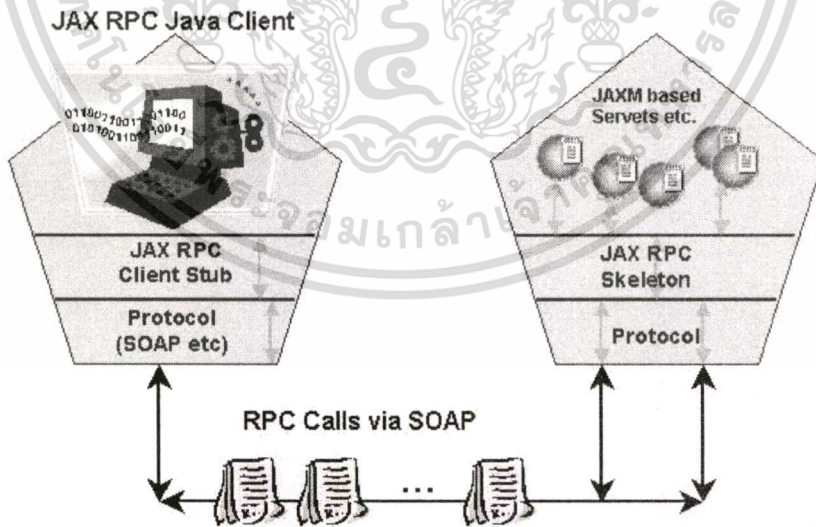
การส่ง SOAPMessage ทำโดยเรียกใช้งาน SOAPConnection Object โดยจะเป็นการรับส่งแบบจุดต่อจุด (Point-to-Point) โดยตรงกับเว็บเซอวิซและเป็นแบบ Synchronous คือส่งคำร้องขอเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปแล้วจะ ทำการบล็อกไว้ไม่ส่งไปอีกรอจนกระทั่งได้รับคำตอบกลับมา การส่งข้อความ ใช้ SAAJ API ที่เรียกว่า Request-Response Messages โดยการเรียกเมธอด call ของ SOAPConnection Object ซึ่งถูกสร้างโดย SOAPConnectionFactory Object

2.3.4 JAX-RPC

Java API for XML-based RPC เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาเพื่อทำการเรียกใช้งานเว็บเซอร์วิส โดยเป็นการเรียกใช้งานกระบวนการระยะไกล (Remote Procedure Calling) โดย JAX-RPC นี้เป็นส่วนที่สำคัญหลักของการพัฒนาจาวาเว็บเซอร์วิสเนื่องจากเป็นตัวสำคัญที่ทำให้เกิดการทำงานร่วมกันข้ามระบบ (Interoperability) ในการทำงานข้ามแพลตฟอร์ม โดยอาศัยพื้นฐานของ XML, WSDL และ SOAP

โดยปกติแล้ว การพัฒนาโปรแกรมเพื่อเรียกใช้งาน Procedure ข้ามเครื่องจะมีขั้นตอนที่ยุ่งยากซับซ้อน แต่ JAX-RPC ได้ทำการอิมพลิเมนต์ ขั้นตอนเหล่านั้นไว้แล้ว เช่น การทำ Marshal/Unmarshal การสร้างคลาสในระดับล่างสำหรับไคลเอนต์ และเซิร์ฟเวอร์ (Stubs และ Ties) การสร้างหรือตีความ SOAPMessage (SAAJ เองก็เป็นการทำงานเบื้องหลังของ JAX-RPC เช่นกัน)



รูปที่ 2.18 การทำงานของ JAX-RPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีก้นำไปใช้

การสร้างเว็บเซอร์วิสที่สามารถเรียกใช้งานผ่าน RPC ได้นั้นมี 2 ขั้นตอนคือสร้างส่วนต่อประสาน และสร้างคลาสที่อิมพลีเมนต์ส่วนต่อประสาน นั้นขึ้นมา และคลาสตัวนี้ก็จะถูกไคลเอนต์มาเรียกใช้เมธอดของมันในภายหลัง ซึ่ง JAX-RPC ไม่ได้บังคับว่าไคลเอนต์ที่ติดต่อเข้ามาเรียกใช้เว็บเซอร์วิสดังกล่าว ต้องรันอยู่บนแพลตฟอร์มของจาวาเท่านั้น

ในการทำงานร่วมกับ XML และ WSDL นั้นในการทำงานเบื้องหลังของ JAX-RPC จะมีการแมประหว่างแบบชนิดข้อมูลของ XML กับ แบบชนิดข้อมูลของภาษาจาวา ซึ่งผู้พัฒนาไม่จำเป็นต้องรู้วิธีการแมป นั้น เพียงแต่ต้องระวังว่าแบบชนิดข้อมูลประเภทใดบ้างในภาษาจาวา ที่สามารถใช้เป็นพารามิเตอร์และค่าที่คืนกลับมาของเมธอดที่สนับสนุนการทำงานของ JAX-RPC โดยแบบชนิดข้อมูลที่สนับสนุนมีดังนี้

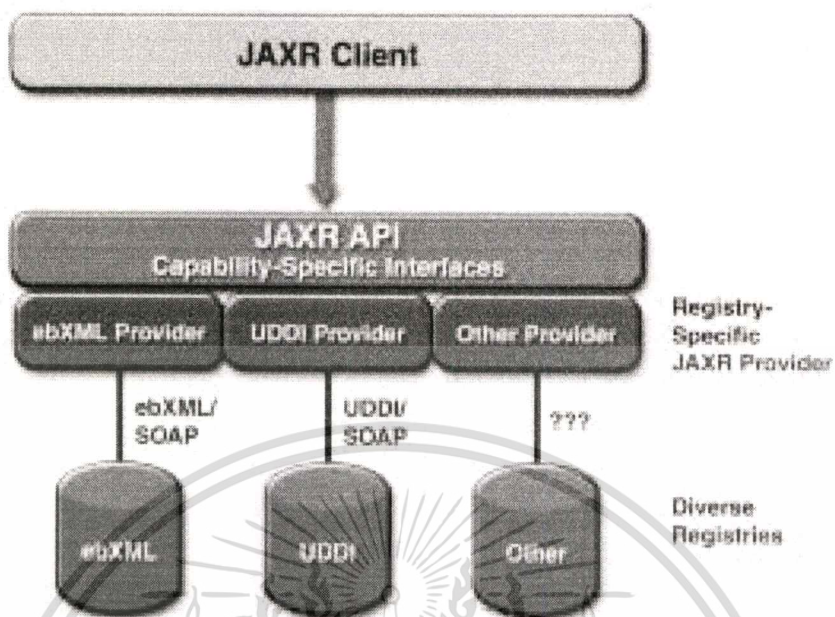
- J2SE SDK Classes
- Primitives
- Arrays
- Value Types
- JavaBeans Components

2.3.5 JAXR

Java API for XML Registries เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาเพื่อทำการติดต่อกับ Registries Server เช่นการเขียนโปรแกรมเพื่อ Publish บริการของตัวเองไปยัง Registries Server หรือการเขียนโปรแกรมเพื่อค้นหาบริการของคนอื่น ตัวอย่างของ Registries Server เช่น UDDI และ ebXML Registries

สถาปัตยกรรมของ JAXR ประกอบด้วย

- JAXR Client เป็นโปรแกรมฝั่งไคลเอนต์ที่มีการเรียกใช้งาน Registry โดยผ่านทาง JAXR Provider (โดยโปรแกรมฝั่งไคลเอนต์นั้นมีการเรียกใช้ JAXR)
- JAXR Provider เป็นการอิมพลีเมนต์ JAXR API เพื่อเรียกใช้งาน JAXR Provider หรือกลุ่มของ JAXR Provider ที่เป็นไปตามมาตรฐานใดมาตรฐานหนึ่ง



รูปที่ 2.19 สถาปัตยกรรม JAXR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์ความต้องการของระบบ

3.1 ความต้องการทางฟังก์ชันของระบบรวม

ระบบโดยรวมต้องสามารถทำงานตามขั้นตอนต่อไปนี้ได้อย่างถูกต้องตามลำดับ

การลงทะเบียนบริการ

1. ผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียนทำการลงทะเบียนกับ Registry Server (UDDI)

การดูรายละเอียดสินค้า

2. ผู้ให้บริการหลักค้นหาผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียนจาก Registry Server
3. ผู้ให้บริการหลักขอข้อมูลราคาจากผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน
ผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียนส่งข้อมูลราคากลับมายังผู้ให้บริการหลัก
4. ผู้ให้บริการหลักประมวลผลและแสดงผลข้อมูลราคาให้แก่ลูกค้า

การสั่งซื้อสินค้า

5. ลูกค้าสั่งซื้อสินค้าจากหน้าจอของผู้ให้บริการหลัก
6. ผู้ให้บริการหลักขอสั่งซื้อสินค้าไปยังผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน
7. ผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียนส่งข้อมูลยืนยันการสั่งซื้อสินค้ากลับมายังผู้
ให้บริการหลัก
8. ผู้ให้บริการหลักประมวลผลและแสดงผลข้อมูลยืนยันการสั่งซื้อสินค้าให้แก่ลูกค้า

การชำระเงิน

9. ลูกค้ากรอกข้อมูลการชำระเงินจากหน้าจอของผู้ให้บริการหลัก
10. ผู้ให้บริการหลักส่งข้อมูลการชำระเงินไปยังผู้ให้บริการธนาคาร
11. ผู้ให้บริการธนาคารส่งข้อมูลยืนยันการชำระเงินกลับมายังผู้ให้บริการหลัก
12. ผู้ให้บริการหลักประมวลผลและแสดงผลข้อมูลยืนยันการชำระเงินให้แก่ลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

การยกเลิกลงทะเบียนบริการ

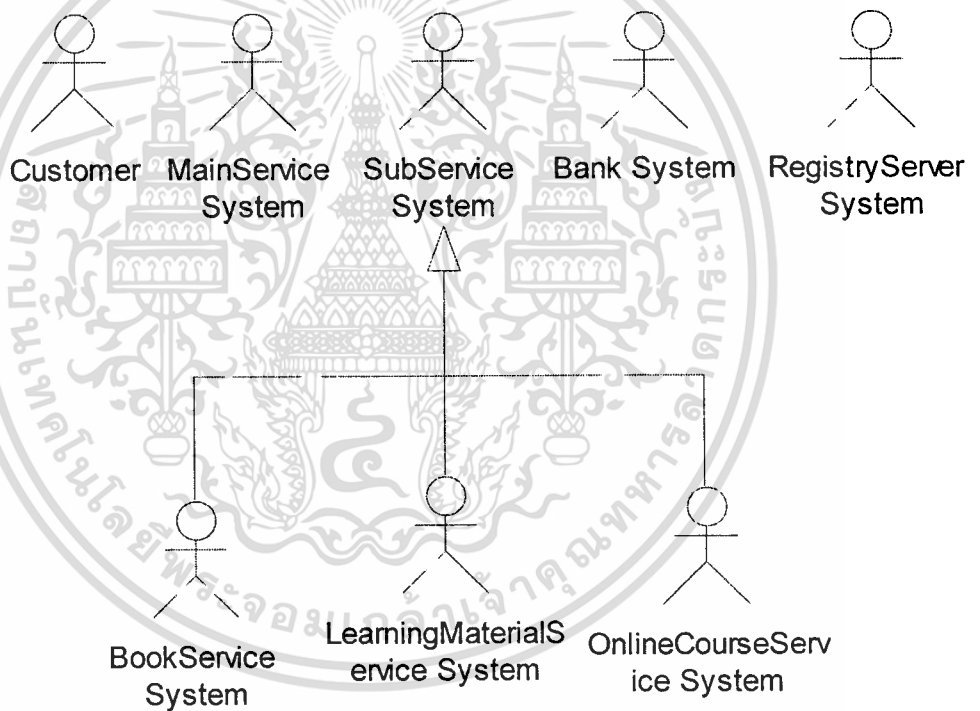
13. ผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียนยกเลิกการลงทะเบียนกับ Registry Server

ตารางที่ 3.1 สรุปบริการที่ต้องมีสำหรับแต่ละระบบย่อย

ผู้ให้บริการ	บริการ
1. ผู้ให้บริการหลัก	<ul style="list-style-type: none"> ● ให้บริการข้อมูลราคาสินค้า ● ติดต่อขอข้อมูลราคาสินค้าจากผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน ● ให้บริการสั่งซื้อสินค้า ● ติดต่อขอสั่งซื้อสินค้าไปยังผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน ● ให้บริการชำระเงิน ● ติดต่อขอชำระเงินไปยังผู้ให้บริการธนาคาร
2. ผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน	<ul style="list-style-type: none"> ● ทำการลงทะเบียนกับ Registry Server เพื่อนำเสนอบริการของตนออกสู่สาธารณะ ● ส่งข้อมูลราคาไปยังผู้ให้บริการหลัก ● ส่งข้อมูลยืนยันการส่งของไปยังผู้ให้บริการหลัก ● ยกเลิกการลงทะเบียนกับ Registry Server
3. ผู้ให้บริการธนาคาร	<ul style="list-style-type: none"> ● ส่งข้อมูลยืนยันการชำระเงินไปยังผู้ให้บริการหลัก
4. ผู้ให้บริการบัญชีรายชื่อเว็บเซิร์ฟเวอร์	<ul style="list-style-type: none"> ● ลงทะเบียนบริการ ● ยกเลิกลงทะเบียนบริการ ● คั่นหาบริการ

จากการวิเคราะห์ข้างต้นสามารถกำหนดแอกเตอร์ของระบบได้ดังนี้

1. Customer คือลูกค้า
2. MainService System คือผู้ให้บริการหลัก
3. SubService System คือ
 - ผู้ให้บริการหนังสือ (BookService System)
 - ผู้ให้บริการสื่อการเรียน (LearningMaterialService System)
4. Bank System คือผู้ให้บริการธนาคาร
5. RegistryServer System คือผู้ให้บริการบัญชีรายชื่อเว็บเซอร์วิส

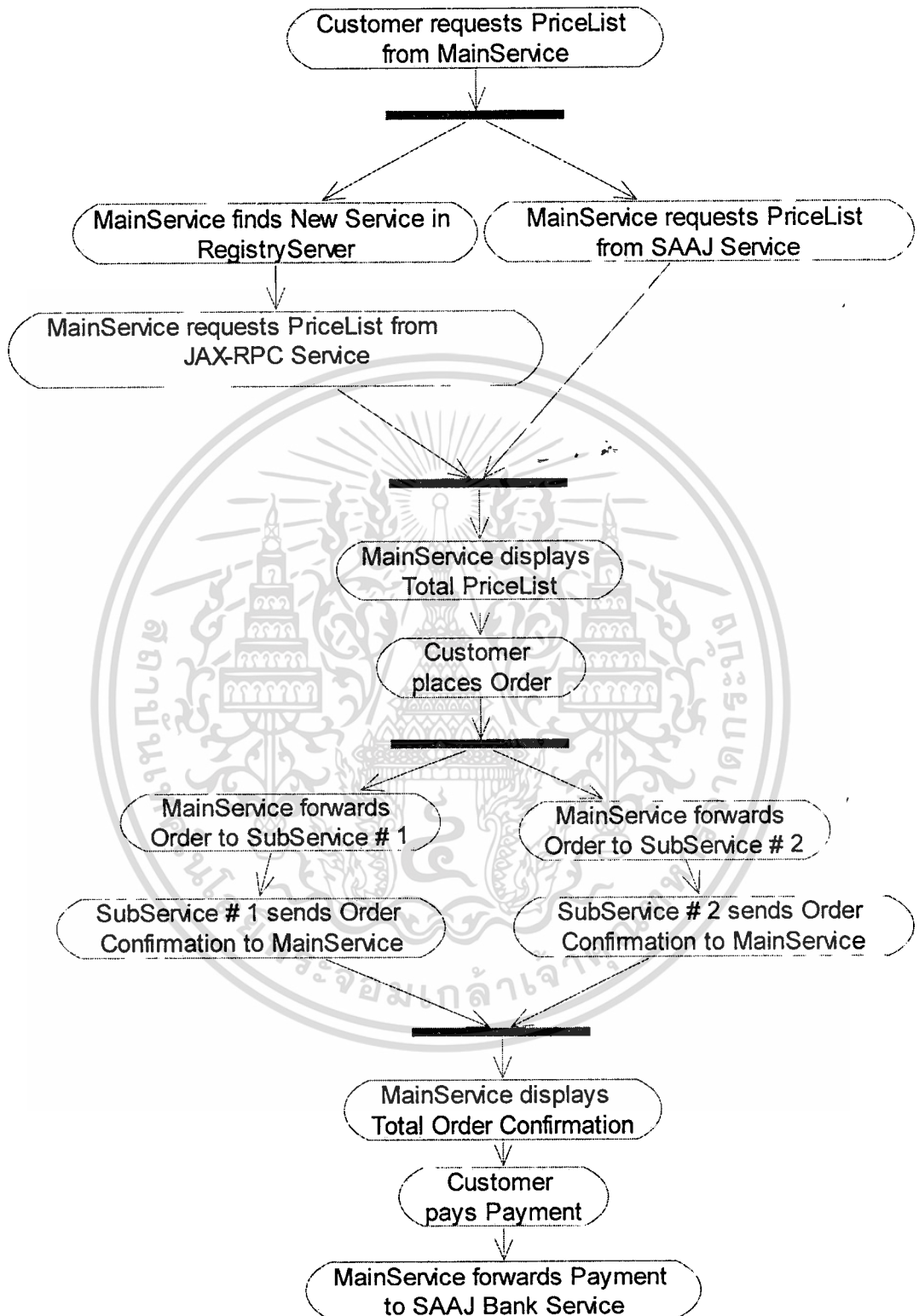


รูปที่ 3.1 แอกเตอร์ของระบบ

Activity Diagram ของระบบทั้งหมด

กิจกรรมต่างๆของระบบทั้งหมดสามารถเขียนเป็น Activity Diagram ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 Activity Diagram ของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Customer requests PriceList from Main Service
คือ ลูกค้าขอข้อมูลราคา
- Main Service finds New Service in Registry Server
คือ ผู้ให้บริการหลักค้นหาบริการใหม่จาก Registry Server
- Main Service requests PriceList from SAAJ Service
คือ ผู้ให้บริการหลักขอข้อมูลราคาไปยังผู้ให้บริการหนังสือหรือผู้ให้บริการสื่อการเรียนที่เคยได้ทำการติดต่อกันมาแล้ว
- Main Service requests PriceList from JAX-RPC Service
คือ ผู้ให้บริการหลักขอข้อมูลราคาไปยังผู้ให้บริการหนังสือหรือผู้ให้บริการสื่อการเรียน รายใหม่
- Main Service displays Total PriceList
คือ ผู้ให้บริการหลักแสดงข้อมูลราคาสินค้าทั้งหมด
- Customer places Order
คือ ลูกค้าขอสั่งซื้อของ
- Main Service forwards Order to Sub Service #1
คือ ผู้ให้บริการหลักส่งของไปยังผู้ให้บริการหนังสือหรือผู้ให้บริการสื่อการเรียนรายที่ 1
- Main Service forwards Order to Sub Service #2
คือ ผู้ให้บริการหลักส่งของไปยังผู้ให้บริการหนังสือหรือผู้ให้บริการสื่อการเรียนรายที่ 2
- Sub Service#1 sends Order Confirmation to Main Service
คือ ผู้ให้บริการหนังสือหรือผู้ให้บริการสื่อการเรียนรายที่ 1 ส่งข้อมูลยืนยันการสั่งซื้อกลับมายังผู้ให้บริการหลัก
- Sub Service#2 sends Order Confirmation to Main Service
คือ ผู้ให้บริการหนังสือหรือผู้ให้บริการสื่อการเรียนรายที่ 2 ส่งข้อมูลยืนยันการสั่งซื้อกลับมายังผู้ให้บริการหลัก
- Main Service displays Total Order Confirmation
คือ ผู้ให้บริการหลักแสดงข้อมูลยืนยันการสั่งซื้อทั้งหมด
- Customer pays Payment
คือ ลูกค้าขอชำระเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

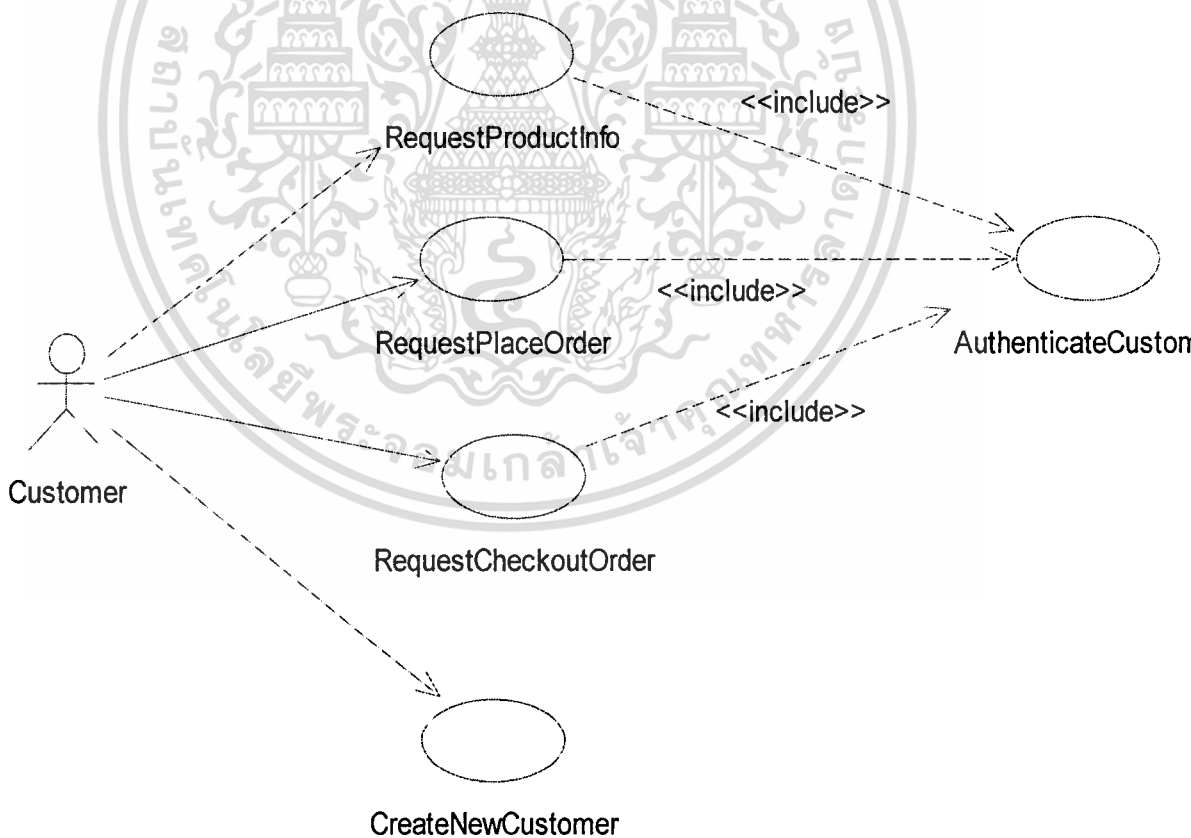
- Main Service forwards Payment to SAAJ Bank Service

คือ ผู้ให้บริการหลักส่งข้อมูลการชำระเงิน ไปยังผู้ให้บริการธนาคารซึ่งเคยทำการติดต่อกันมาก่อนแล้ว

3.2 การวิเคราะห์ระบบโดยแยกแต่ละบริการ

3.2.1 ยูสเคสไดอะแกรมของผู้ให้บริการหลัก

- RequestProductInfo ถูกค้าขอรายละเอียดสินค้า
- RequestPlaceOrder ถูกค้าสั่งซื้อสินค้า
- RequestCheckOutOrder ถูกค้าชำระเงิน
- AuthenticateCustomer ถูกค้าล็อกอินเข้าระบบ
- CreateNewCustomer ถูกค้าสมัครสมาชิกใหม่

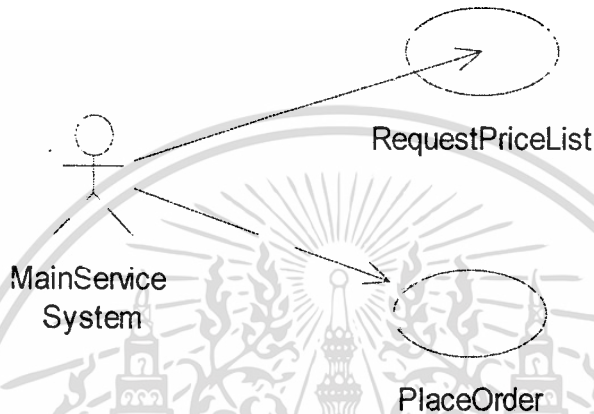


รูปที่ 3.3 ยูสเคสไดอะแกรมของผู้ให้บริการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ยูสเคสไดอะแกรมของผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน

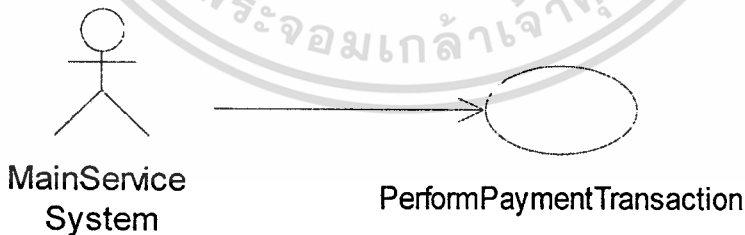
- RequestPriceList ผู้ให้บริการหลักขอราคาสินค้า
- PlaceOrder ผู้ให้บริการหลักทำการสั่งซื้อสินค้า



รูปที่ 3.4 ยูสเคสไดอะแกรมของผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน

3.2.3 ยูสเคสไดอะแกรมของผู้ให้บริการธนาคาร

- PerformPaymentTransaction ผู้ให้บริการหลักทำการชำระเงิน

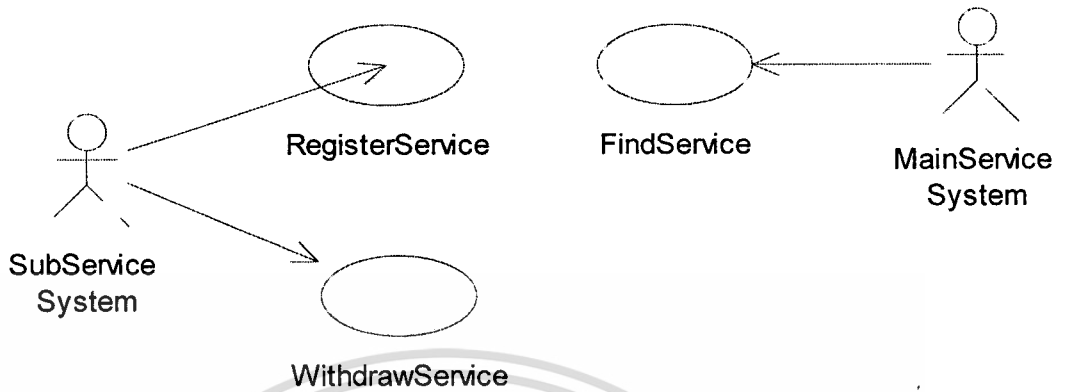


รูปที่ 3.5 ยูสเคสไดอะแกรมของผู้ให้บริการธนาคาร

3.2.4 ยูสเคสไดอะแกรมของผู้ให้บริการบัญชีรายชื่อเว็บเซอร์วิส

- RegisterService ผู้ให้บริการย่อยลงทะเบียนกับ RegistryServer
- WithdrawService ผู้ให้บริการย่อยถอนบริการออกจาก RegistryServer
- FindService ผู้ให้บริการหลักค้นหาบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ยูสเคสไดอะแกรมของผู้ให้บริการบัญชีรายชื่อเว็บเซอร์วิส



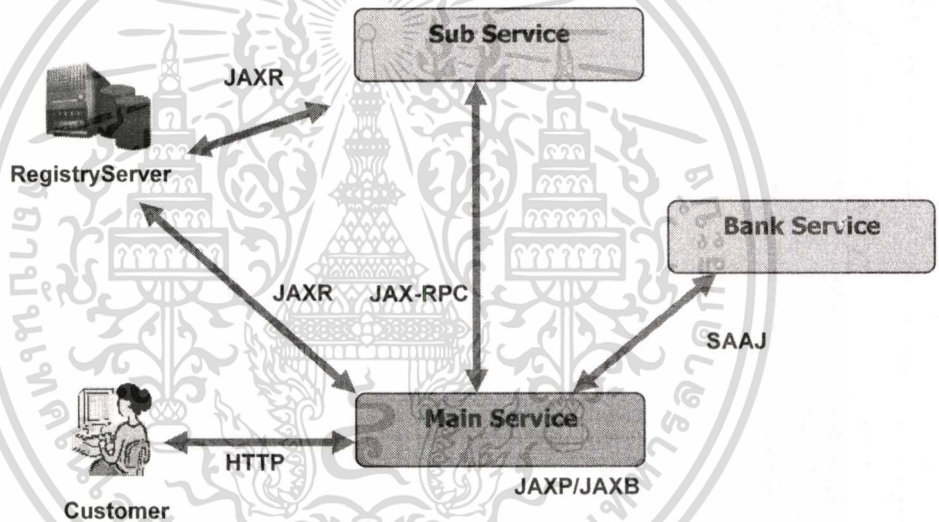
บทที่ 4

การออกแบบระบบ

4.1 สถาปัตยกรรมของระบบ

ระบบขายหนังสือและสื่อการเรียน โดยใช้จาวาเว็บเซอร์วิส ประกอบด้วยการทำงานร่วมกันของผู้ให้บริการหลัก ผู้ให้บริการหนังสือ ผู้ให้บริการสื่อการเรียน และผู้ให้บริการธนาคาร ดังรูปที่

4.1



รูปที่ 4.1 สถาปัตยกรรมของระบบ

ในขั้นตอนแรกผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียนต้องทำการลงทะเบียนกับ Registry Server เพื่อนำเสนอบริการของตนเองให้ออกสู่สาธารณะก่อน (โดยใช้ JAXR)

ลูกค้าจะติดต่อเข้ามาที่ผู้ให้บริการหลักโดยเป็นการเรียกหน้าเว็บผ่านโปรโตคอล HTTP ผู้ให้บริการหลักก็จะทำการค้นหาบริการจาก Registry Server (โดยใช้ JAXR) ซึ่งทาง Registry Server จะส่งข้อมูลรายการบริการตอบกลับมาให้

หลังจากนั้นผู้ให้บริการหลักจะทำการติดต่อกับผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน ในรูปแบบของการเรียกใช้เมธอดแบบทางไกล (โดยใช้ JAX-RPC) และเมื่อได้ข้อมูลตอบกลับมา ผู้ให้บริการหลักจะทำการประมวลผลข้อมูลนั้นเพื่อแสดงผลให้กับลูกค้า (โดยใช้ JAXB และ JAXP)

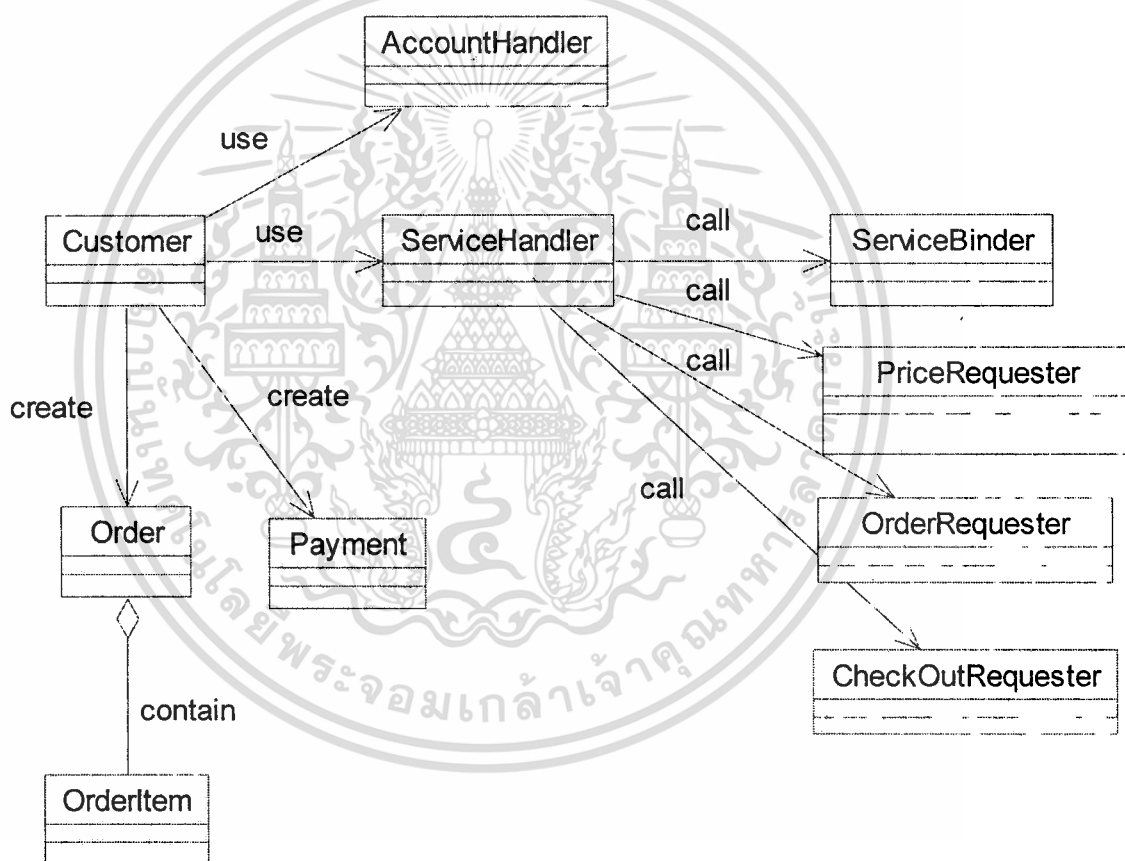
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการติดต่อระหว่างผู้ให้บริการหลักกับผู้ให้บริการธนาคารนั้นเป็นแบบที่เคยติดต่อกันมาก่อนแล้ว ซึ่งมีการตกลง DTD สำหรับเอกสาร XML ไว้เรียบร้อยแล้ว ข้อมูลที่ส่งรับระหว่างกันจึงเป็นเอกสาร XML ที่ส่งมาใน SOAP Message (โดยใช้ SAAJ) หลังจากที่ลูกค้าสั่งชำระเงินแล้วก็สามารถส่งข้อมูลได้เลย ไม่ต้องค้นหาบริการใหม่จาก Registry Server

4.2 คลาสไดอะแกรม

4.2.1 คลาสไดอะแกรมของผู้ให้บริการหลัก

คลาสไดอะแกรมของผู้ให้บริการหลักจะประกอบด้วยคลาสต่างๆดังรูปที่ 4.2



รูปที่ 4.2 คลาสไดอะแกรมของผู้ให้บริการหลัก

1. Customer ใช้ธิบายอ็อบเจกต์ลูกค้าที่เข้ามาติดต่อกับระบบ ประกอบด้วยแอตทริบิวต์ดังนี้
 - userid รหัสผู้ใช้
 - password รหัสผ่าน
2. AccountHandler ใช้ธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการดูแลจัดการบัญชีรายชื่อผู้ใช้ระบบ ประกอบด้วยเมธอดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

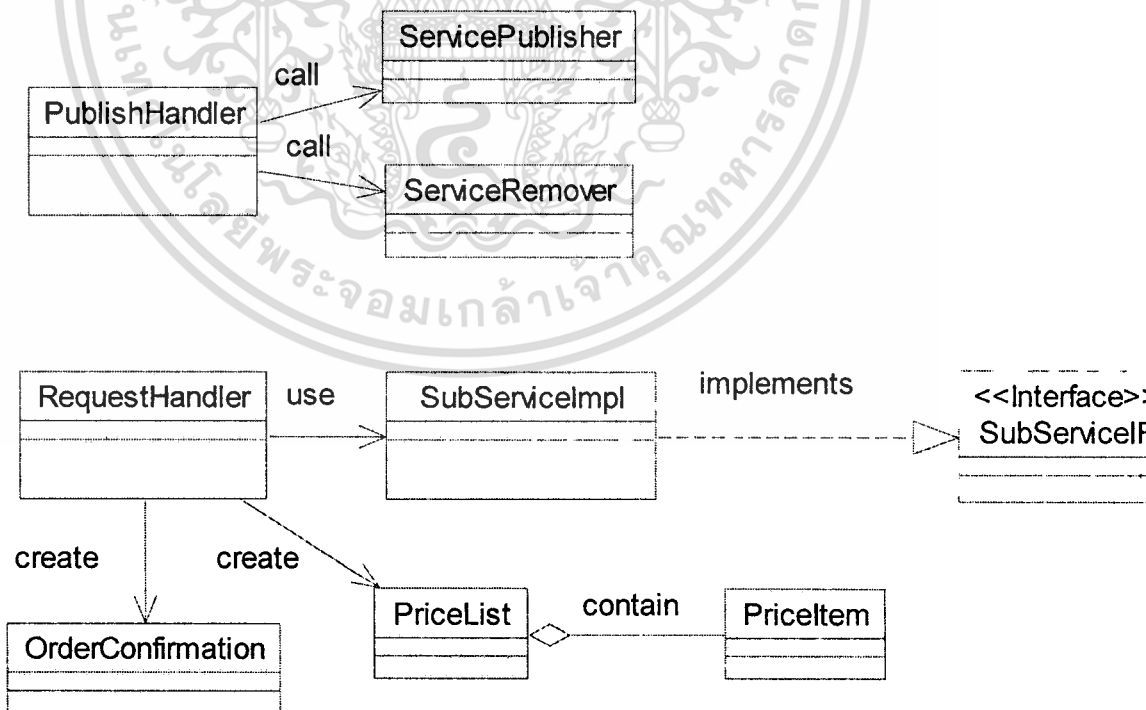
- addCustomer เพิ่มรายชื่อสมาชิกใหม่ให้กับระบบ
 - validateCustomer ตรวจสอบสิทธิการใช้งานระบบ (การล็อกอิน)
 - displayResult รายงานผลการตรวจสอบสิทธิการใช้งานระบบ
3. ServiceHandler ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการดูแลจัดการบริการต่างๆ ให้กับลูกค้า เช่น ลูกค้าขอราคาสินค้า ลูกค้าขอสั่งซื้อ และลูกค้าขอชำระเงิน เป็นต้น โดย ServiceHandler จะเป็นส่วนที่ติดต่อกับลูกค้า ซึ่งเมื่อรับคำร้องขอมาแล้วจะทำการเรียกใช้ไปยังอ็อบเจกต์อื่นๆซึ่งทำหน้าที่เฉพาะ เช่น ServiceBinder, PriceRequester, OrderRequester และ CheckoutRequester
- โดย ServiceHandler ประกอบด้วยเมธอดต่างๆดังนี้
- performRequest ดำเนินการกับคำร้องขอ โดยเรียกใช้ไปยังอ็อบเจกต์อื่นๆ
 - createOrder สร้าง อ็อบเจกต์ Order เพื่อเตรียมส่งไปใช้งานกับอ็อบเจกต์ OrderRequester เพื่อทำการสั่งซื้อสินค้าให้กับลูกค้า
 - createPayment สร้างอ็อบเจกต์ Payment เพื่อเตรียมส่งไปใช้งานกับอ็อบเจกต์ CheckoutRequester เพื่อทำการชำระเงินให้กับลูกค้า
4. ServiceBinder ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการค้นหาบริการใน Registry Server ประกอบด้วยเมธอดต่างๆดังนี้
- makeConnection ทำการติดต่อไปยัง Registry Server
 - executeQuery ทำการประมวลผลข้อความที่ส่งไป
 - getServiceName รับชื่อบริการมาจาก Registry Server ในกรณีที่ยังค้นหาเจอ
5. PriceRequester ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการส่งคำร้องขอ การร้องขอราคาไปยังผู้ให้บริการย่อย ประกอบด้วยเมธอด sendRequest ซึ่งเป็นการส่ง Request ไปยังผู้ให้บริการย่อย จากนั้นทางผู้ให้บริการย่อยก็จะตอบกลับมาเป็นคำตอบกลับซึ่งภายในประกอบด้วยอ็อบเจกต์ PriceList ผู้ให้บริการหลักก็จะทำการประมวลผลและตีความข้อมูลดังกล่าวรายงานผลไปยังหน้าจอให้ลูกค้า
6. OrderRequester ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการส่ง Request การขอสั่งซื้อสินค้าไปยังผู้ให้บริการย่อย ประกอบด้วยเมธอด sendRequest ซึ่งใช้ส่งไปยังผู้ให้บริการย่อย แต่ก่อนจะส่งไปนั้นจะต้องมีการส่งอ็อบเจกต์ Order ไปด้วย (อ็อบเจกต์ ServiceHandler เป็นผู้จัดเตรียม) และผู้ให้บริการย่อยจะตอบกลับมาเป็น Response ซึ่งภายในประกอบด้วยอ็อบเจกต์ OrderConfirmation ถือเป็นการยืนยันการสั่งซื้อสินค้าที่ส่งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. CheckoutRequester ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการส่ง Request การขอชำระเงินไปยังผู้ให้บริการธนาคาร ประกอบด้วยเมธอด sendRequest ทำการส่ง แต่ก่อนจะส่งไปนั้นจะต้องมีการส่งอ็อบเจกต์ Payment ไปด้วย (อ็อบเจกต์ServiceHandler เป็นผู้จัดเตรียม) การทำงานเบื้องหลังจะไปติดต่อชำระเงินกับผู้ให้บริการธนาคาร
8. Order ใช้อธิบายอ็อบเจกต์รายการสั่งซื้อสินค้า โดย Order จะประกอบด้วย OrderItem คือรายการสินค้าย่อย อ็อบเจกต์ Order นี้จะสร้างขึ้นโดย ServiceHandler เพื่อไปใช้งานกับ OrderRequester โดยอ็อบเจกต์ Order มีเมธอด fill Order List เพื่อทำการรวบรวมรายการสั่งซื้อสินค้าย่อย
9. OrderItem ใช้อธิบายอ็อบเจกต์รายการสินค้าย่อยแต่ละรายการ ประกอบด้วยแอตทริบิวต์ item (สินค้า) และ quantity (จำนวนที่สั่ง)
10. Payment ใช้อธิบายอ็อบเจกต์รายการชำระเงิน โดยอ็อบเจกต์ Payment นี้สร้างขึ้นโดย ServiceHandler เพื่อไปใช้งานกับ CheckOutRequester

4.2.2 คลาสไดอะแกรมของผู้ให้บริการย่อย

คลาสไดอะแกรมของผู้ให้บริการย่อยจะประกอบด้วยคลาสต่างๆดังรูปที่ 4.3



รูปที่ 4.3 คลาสไดอะแกรมของผู้ให้บริการหนังสือและสื่อการเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **PublishHandler** ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการดูแลจัดการการลงทะเบียนบริการ โดยจะติดต่อกับ Registry Server ให้กับผู้ให้บริการย่อย โดยจะทำการเรียกใช้ไปยังอ็อบเจกต์ ServicePublisher และ ServiceRemover เพื่อทำหน้าที่เฉพาะในการลงทะเบียนบริการและยกเลิกลงทะเบียนบริการตามลำดับ อ็อบเจกต์ PublishHandler ประกอบด้วยเมธอด performRequest ใช้สำหรับจัดการคำร้องขอ ที่รับมาซึ่งจะทำการแยกติดต่อไปยังอ็อบเจกต์ที่ทำงานเฉพาะตามที่กล่าวมา
2. **ServicePublisher** ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการลงทะเบียนบริการกับ Registry Server ประกอบด้วยเมธอดต่างๆดังต่อไปนี้
 - makeConnection ทำการติดต่อไปยัง Registry Server
 - executePublish ทำการประมวลผลลงทะเบียนบริการ โดยทาง Registry Server จะส่งคีย์ที่มีค่าไม่ซ้ำใคร (Unique Key) กลับมาให้ แสดงว่าลงทะเบียนสำเร็จแล้ว
3. **ServiceRemover** ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการยกเลิกลงทะเบียนบริการกับ Registry Server ประกอบด้วยเมธอดต่างๆดังต่อไปนี้
 - makeConnection ทำการติดต่อไปยัง Registry Server
 - executeRemove ทำการประมวลผลยกเลิกลงทะเบียนบริการ โดยในการทำเมธอดนี้ ทางผู้ให้บริการต้องส่งคีย์ที่มีค่าไม่ซ้ำใคร (Unique Key) ไปทำการยกเลิกด้วย
4. **RequestHandler** ใช้อธิบายอ็อบเจกต์ซึ่งทำหน้าที่ในการจัดการคำร้องขอ ที่รับมาจากผู้ให้บริการหลัก โดยจะทำการเรียกใช้งานเมธอดของอ็อบเจกต์ SubServiceImpl อีกต่อหนึ่ง
5. **SubServiceIF** เป็นส่วนต่อประสานหรืออินเทอร์เฟซที่ทางผู้ให้บริการย่อยทำการประกาศไว้ เนื่องจากเป็นการเรียกกระบวนการข้ามเครื่อง จำเป็นต้องมีการกำหนดไว้เพื่อให้เป็นมาตรฐานเดียวกัน หากใครจะใช้บริการก็ต้องปฏิบัติตาม จึงจะใช้งานร่วมกันได้ โดยจะทำการเป็นโครงเมธอดไว้ให้อิมพลีเมนต์ภายหลัง ในที่นี้อินเทอร์เฟซ SubServiceIF กำหนด Abstract Method ไว้ดังนี้
 - placeOrder ใช้ในการสั่งซื้อสินค้า ส่งพารามิเตอร์เป็นอ็อบเจกต์ Order และค่าที่ส่งกลับเป็น อ็อบเจกต์ OrderConfirmation
 - getPriceList ใช้สอบถามราคาสินค้า ค่าที่ส่งกลับเป็นอ็อบเจกต์ PriceList
6. **SubServiceImpl** เป็นคลาสที่ทำการอิมพลีเมนต์อินเทอร์เฟซ SubServiceIF ที่กล่าวข้างต้น โดยจะมีรายละเอียดภายในของเมธอดละเอียดสำหรับการทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. PriceList ใช้อธิบายอ็อบเจกต์รายการราคาสินค้า โดยจะประกอบขึ้นจากPriceListItem โดยอ็อบเจกต์ PriceList มีเมธอด fillPriceList เพื่อทำการรวบรวมรายการราคาสินค้าย่อย
8. PriceListItem ใช้อธิบายอ็อบเจกต์รายการราคาสินค้าย่อยประกอบด้วยแอตทริบิวต์ item(สินค้า) และ price(ราคา)
9. OrderConfirmation ใช้อธิบายอ็อบเจกต์ยืนยันการสั่งซื้อสินค้า ประกอบด้วยแอตทริบิวต์
 - orderid รหัสใบสั่งซื้อสินค้า
 - shipdate วันนัดส่งสินค้า

4.2.3 คลาสไดอะแกรมของผู้ให้บริการธนาคาร

คลาสไดอะแกรมของผู้ให้บริการธนาคารจะประกอบด้วยคลาสต่างๆดังรูปที่ 4.4

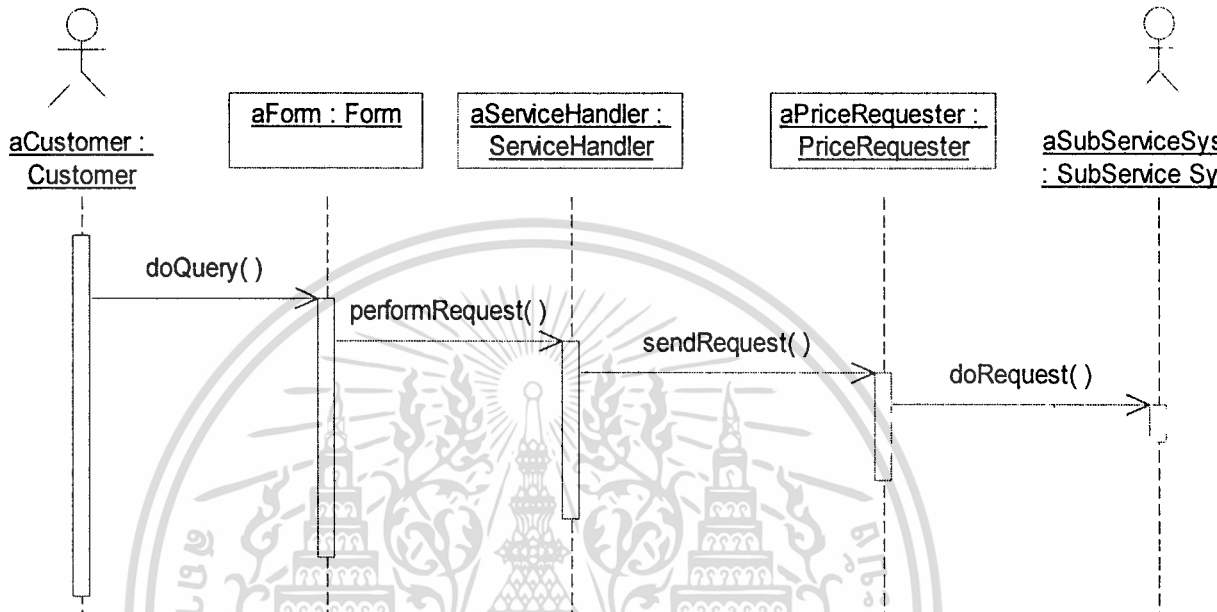


รูปที่ 4.4 คลาสไดอะแกรมของผู้ให้บริการธนาคาร

1. CreditCardHandler ใช้อธิบายอ็อบเจกต์ที่ทำหน้าที่เกี่ยวกับการชำระเงินผ่านบัตรเครดิต โดยจะรับคำร้องขอจากผู้ให้บริการหลัก ประกอบด้วย
 - เมธอด validate สำหรับตรวจสอบบัตร
 - เมธอด performPaymentTransaction สำหรับทำการตัดบัญชี
2. Payment ใช้อธิบายอ็อบเจกต์รายการชำระเงิน ประกอบด้วย attribute ต่างๆดังนี้
 - cardno หมายเลขบัตรเครดิต
 - cardtype ประเภทบัตร
 - expiredate วันที่หมดอายุบัตรเครดิต
 - total จำนวนเงิน
 - shopid รหัสร้านค้า
 - ordereid รหัสใบสั่งซื้อของร้านค้า
 - checkdigit รหัส 3 ตัว

4.3 ซีควเอนซ์ไดอะแกรม

4.3.1 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestProductInfo ของผู้ให้บริการหลัก



รูปที่ 4.5 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestProductInfo ของผู้ให้บริการหลัก

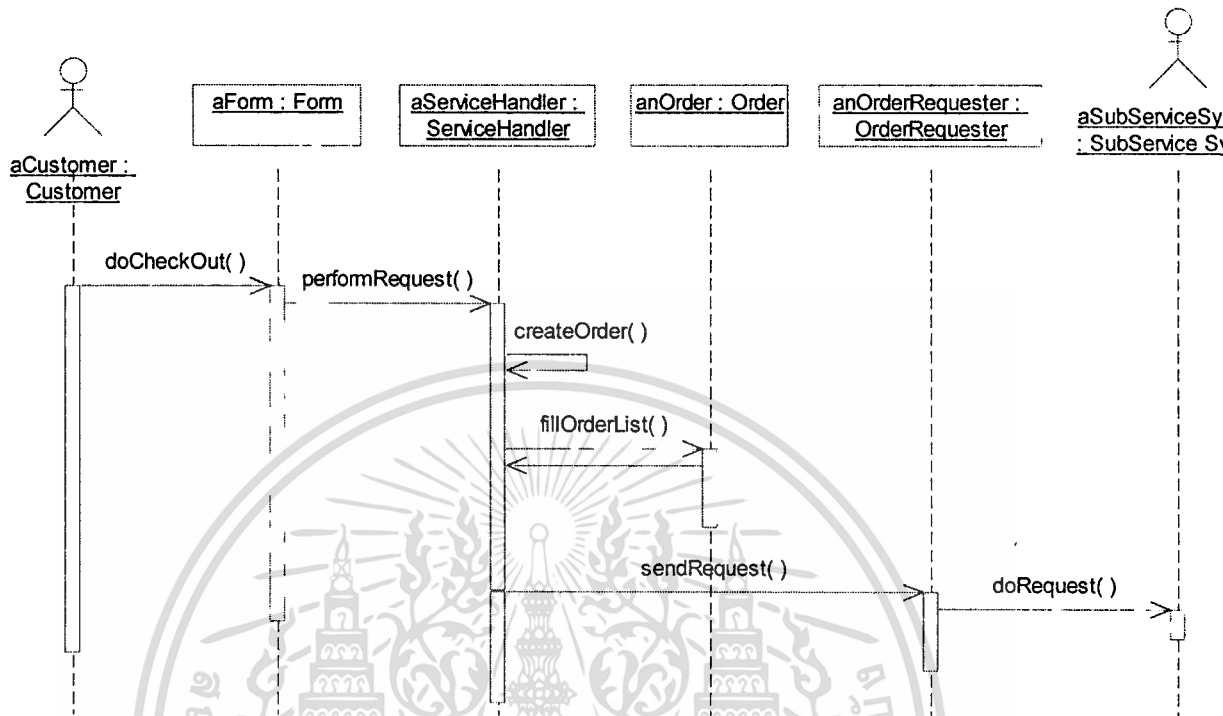
ตารางที่ 4.1 ตารางอธิบายรายละเอียดประกอบซีควเอนซ์ไดอะแกรมสำหรับยูสเคส

RequestProductInfo

คำอธิบาย	ลูกค้าร้องขอราคาสินค้ามายังผู้ให้บริการหลัก
Actor หลัก	Customer
เงื่อนไขเริ่มต้น	<ol style="list-style-type: none"> 1. ลูกค้าต้องทำการล็อกอินเข้าระบบ 2. ผู้ให้บริการหลักต้องทราบชื่อบริการย่อย
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. ลูกค้าใส่ข้อมูลสินค้าที่ต้องการค้นหาและคลิกปุ่ม Search 2. ServiceHandler เรียกใช้ PriceRequester 3. PriceRequester ส่ง Request Message ไปยังผู้ให้บริการย่อย 4. ผู้ให้บริการย่อยรับ Request Message และส่ง Response Message กลับมาเป็น PriceList
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำงานสำเร็จจะแสดงข้อมูลสินค้าของแต่ละผู้ให้บริการย่อย 2. หากทำงานไม่สำเร็จจะแสดงรายงานความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestPlaceOrder ของผู้ให้บริการหลัก



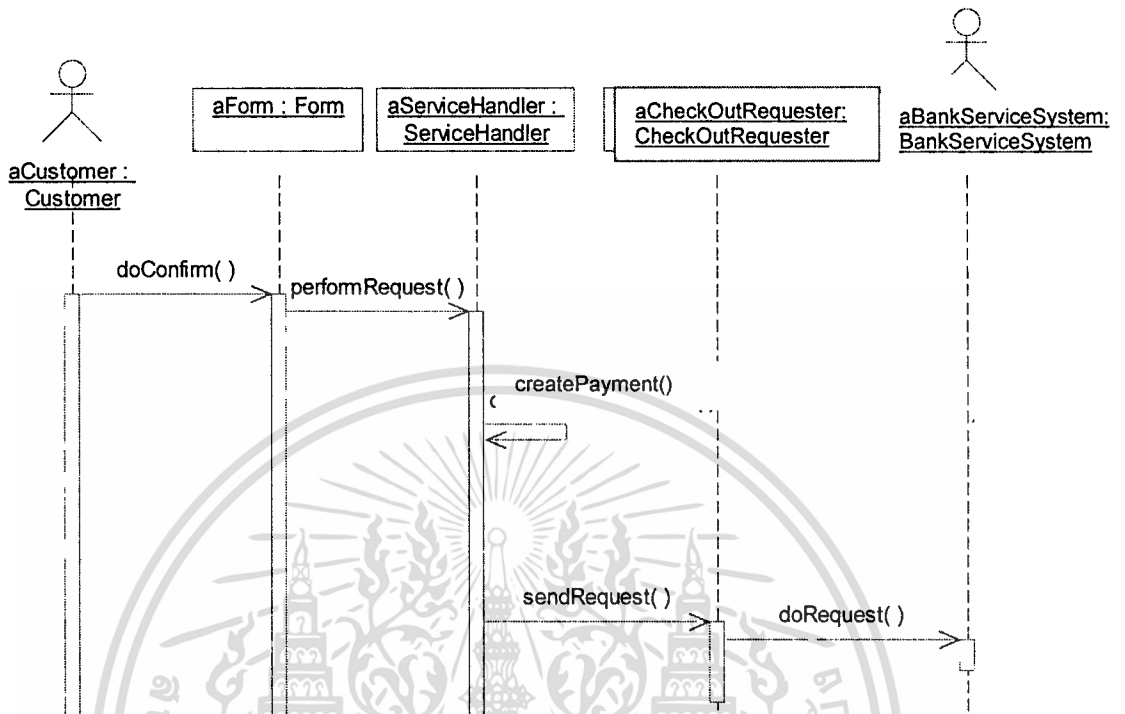
รูปที่ 4.6 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestPlaceOrder ของผู้ให้บริการหลัก

ตารางที่ 4.2 ตารางอธิบายรายละเอียดประกอบซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestPlaceOrder

คำอธิบาย	ลูกค้าร้องขอสั่งซื้อสินค้าจากผู้ให้บริการหลัก
Actor หลัก	Customer
เงื่อนไขเริ่มต้น	<ol style="list-style-type: none"> 1. ลูกค้าต้องทำการล็อกอินเข้าระบบ 2. ผู้ให้บริการหลักต้องทราบชื่อบริการย่อย
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. ลูกค้าคลิกปุ่ม Checkout 2. ServiceHandler สร้าง Order 3. ServiceHandler เรียกใช้ OrderRequester 4. OrderRequester ส่ง Request Message ไปยังผู้ให้บริการย่อย 5. ผู้ให้บริการย่อยรับ Request Message และตอบกลับมาเป็น OrderConfirmation
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำงานสำเร็จจะแสดงข้อมูลยืนยันการสั่งซื้อสินค้า 2. หากไม่สำเร็จจะรายงานความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestCheckoutOrder ของผู้ให้บริการหลัก



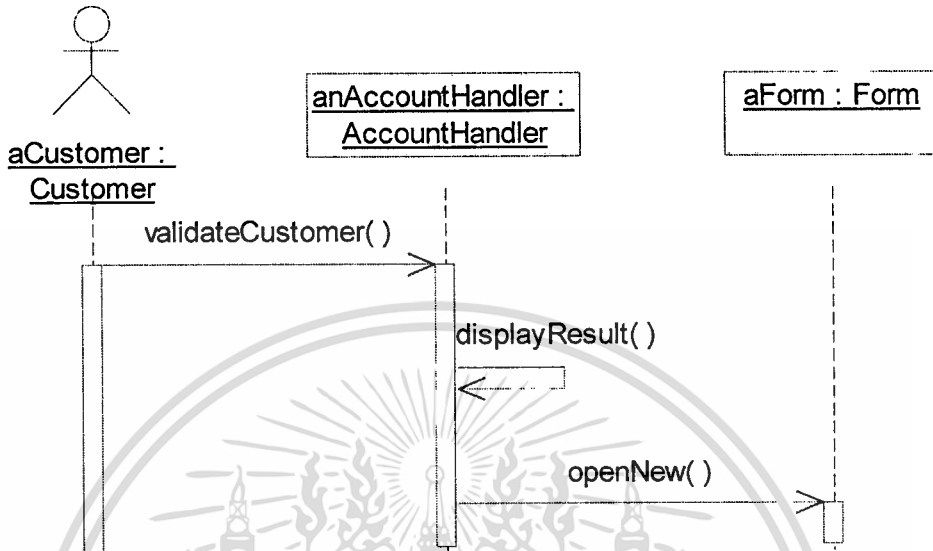
รูปที่ 4.7 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestCheckoutOrder ของผู้ให้บริการหลัก

ตารางที่ 4.3 ตารางอธิบายรายละเอียดประกอบซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestCheckoutOrder

คำอธิบาย	ลูกค้าร้องขอชำระเงินมายังผู้ให้บริการหลัก
Actor หลัก	Customer
เงื่อนไขเริ่มต้น	ลูกค้าต้องทำการเลือกสินค้าอย่างน้อย 1 รายการ
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. ลูกค้าคลิกปุ่ม Confirm 2. ServiceHandler สร้าง Payment 3. ServiceHandler เรียกใช้ CheckOutRequester CheckoutRequester ส่ง Request Message ไปยังผู้ให้บริการธนาคาร
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำงานสำเร็จจะแสดงข้อมูลยืนยันการชำระเงิน 2. หากไม่สำเร็จจะรายงานความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส AuthenticateCustomer ของผู้ให้บริการหลัก

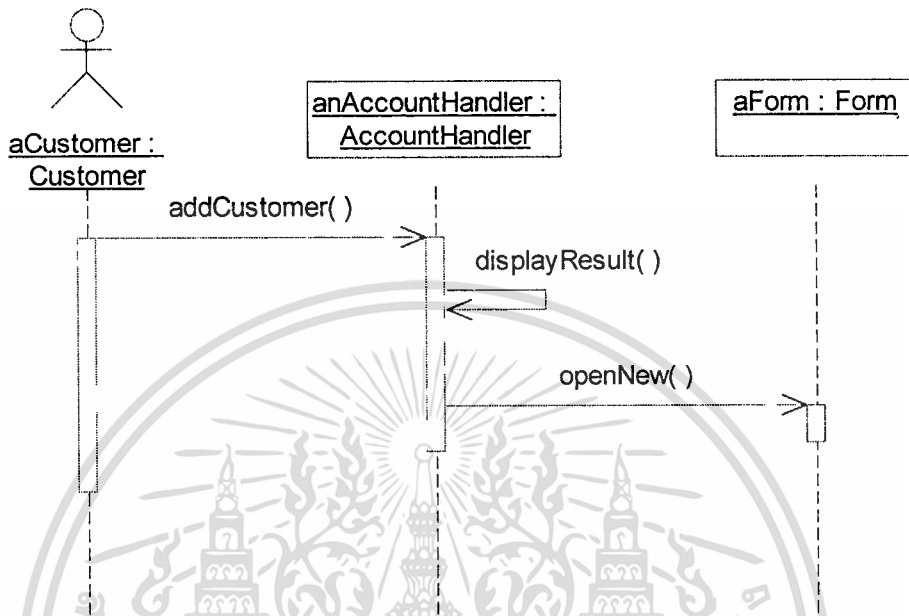


รูปที่ 4.8 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส AuthenticateCustomer ของผู้ให้บริการหลัก

ตารางที่ 4.4 ตารางอธิบายรายละเอียดประกอบซีควเอนซ์ไดอะแกรมสำหรับยูสเคส AuthenticateCustomer

คำอธิบาย	ลูกค้าทำการล็อกอินเข้าระบบ
Actor หลัก	Customer
เงื่อนไขเริ่มต้น	ลูกค้าต้องเป็นสมาชิกของระบบ
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. ลูกค้าป้อนข้อมูลรหัสผู้ใช้และรหัสผ่าน 2. ลูกค้าคลิกปุ่ม Login
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำการล็อกอินสำเร็จจะเข้าสู่ระบบ 2. หากทำการล็อกอินไม่สำเร็จจะแสดงข้อผิดพลาดกลับไปยัง Customer

4.3.5 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส CreateNewCustomer ของผู้ให้บริการหลัก



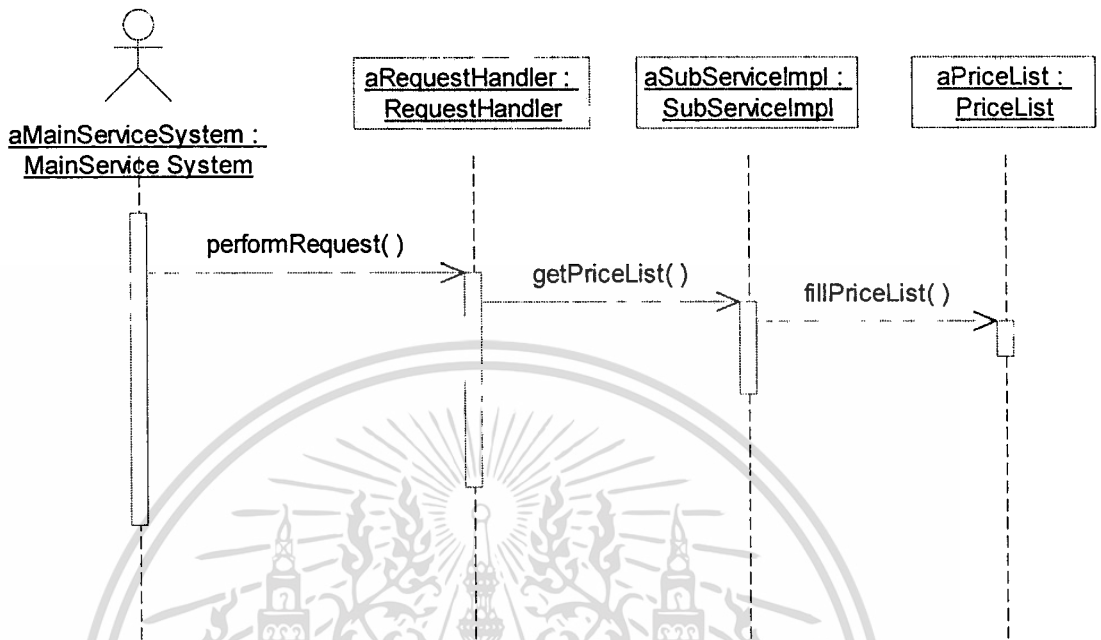
รูปที่ 4.9 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส CreateNewCustomer ของผู้ให้บริการหลัก

ตารางที่ 4.5 ตารางอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรมสำหรับยูสเคส CreateNewCustomer

คำอธิบาย	ลูกค้าทำการสมัครสมาชิกใหม่
Actor หลัก	Customer
เงื่อนไขเริ่มต้น	
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. ลูกค้ากรอกข้อมูลส่วนตัว 2. ลูกค้าคลิกปุ่ม Register
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำการลงทะเบียนสำเร็จจะรายงานผลและทำการเข้าสู่ระบบ 2. หากทำการลงทะเบียนไม่สำเร็จจะรายงานความผิดพลาดไปยังลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestPriceList ของผู้ให้บริการย่อย

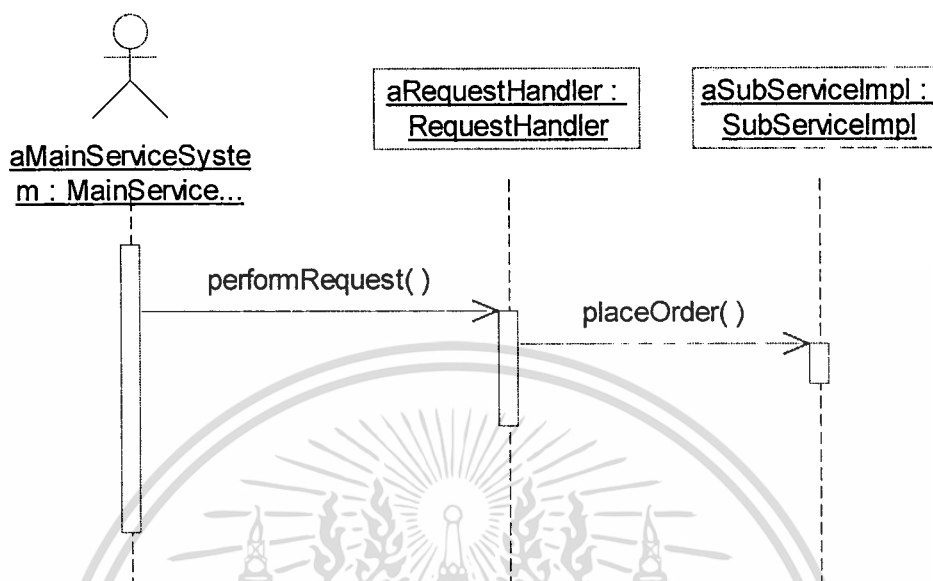


รูปที่ 4.10 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestPriceList ของผู้ให้บริการย่อย

ตารางที่ 4.6 ตารางอธิบายรายละเอียดประกอบซีควเอนซ์ไดอะแกรมสำหรับยูสเคส RequestPriceList

คำอธิบาย	ผู้ให้บริการหลักขอราคาสินค้าจากผู้ให้บริการย่อย
Actor หลัก	MainService System
เงื่อนไขเริ่มต้น	ผู้ให้บริการหลักทราบชื่อผู้ให้บริการย่อย
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. RequestHandler รับ Request เข้ามา และเรียกใช้งาน SubServiceImpl 2. เรียกใช้เมธอด getPriceList ของ SubServiceImpl
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำงานสำเร็จจะ Return ค่าออกมาเป็น PriceList 2. หากทำงานไม่สำเร็จจะ Return NULL

4.3.7 ซีควენซ์ไดอะแกรมสำหรับยูสเคส PlaceOrder ของผู้ให้บริการย่อย



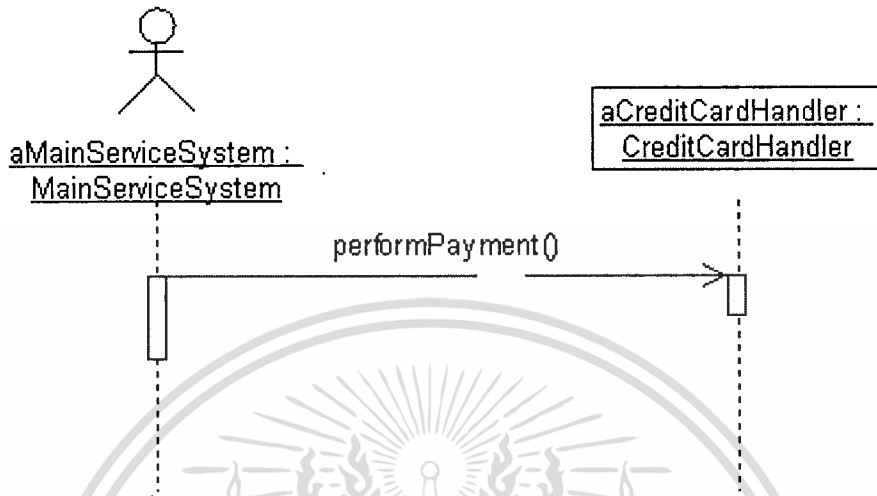
รูปที่ 4.11 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส PlaceOrder ของผู้ให้บริการย่อย

ตารางที่ 4.7 ตารางอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรมสำหรับยูสเคส PlaceOrder

คำอธิบาย	ผู้ให้บริการหลักส่งสินค้าจากผู้ให้บริการย่อย
Actor หลัก	MainService System
เงื่อนไขเริ่มต้น	ผู้ให้บริการหลักทราบชื่อผู้ให้บริการย่อย
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. RequestHandler รับ Request เข้ามาและเรียกใช้งาน SubServiceImpl 2. เรียกใช้เมธอด placeOrder ของ SubServiceImpl
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำงานสำเร็จจะ Return ค่าออกมาเป็น OrderConfirmation 2. หากทำงานไม่สำเร็จจะ Return NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.8 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส PerformPaymentTransaction ของผู้ให้บริการธนาคาร

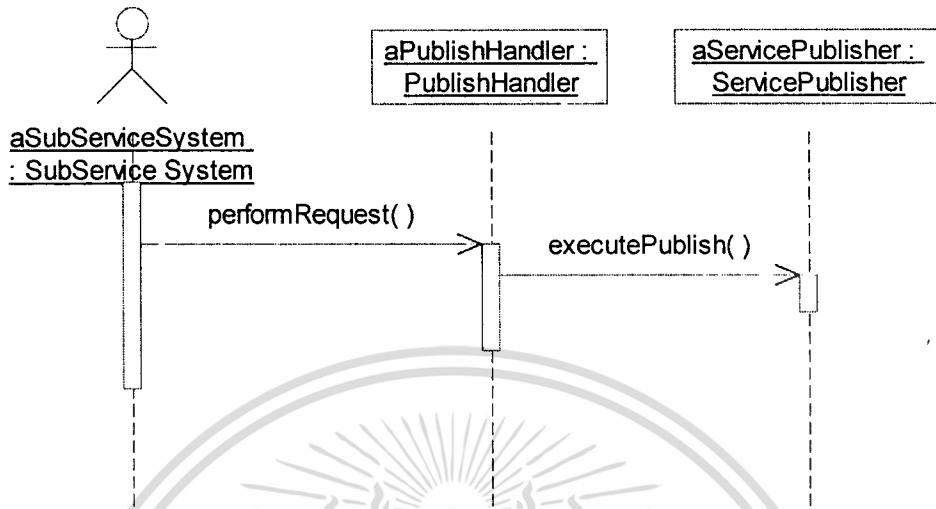


รูปที่ 4.12 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส PerformPaymentTransaction ของผู้ให้บริการธนาคาร

ตารางที่ 4.8 ตารางอธิบายรายละเอียดประกอบควเอนซ์ไดอะแกรมสำหรับยูสเคส PerformPaymentTransaction

คำอธิบาย	ผู้ให้บริการหลักทำการชำระเงินไปยังธนาคาร
Actor หลัก	MainService System
เงื่อนไขเริ่มต้น	ผู้ให้บริการหลักต้องสั่งชำระเงินไปยังผู้ให้บริการย่อยเสียก่อน
ลำดับการทำงานหลัก	<ol style="list-style-type: none"> 1. CreditCardHandler รับ Request เข้ามา 2. เรียกใช้เมธอด performPayment() เพื่อทำการตัดบัตรเครดิต
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากชำระเงินสำเร็จจะมีข้อความแสดง Success จากทางธนาคาร 2. หากชำระเงินไม่สำเร็จ ทางธนาคารจะแสดงรายงานความผิดพลาดกลับมา

4.3.9 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส RegisterService

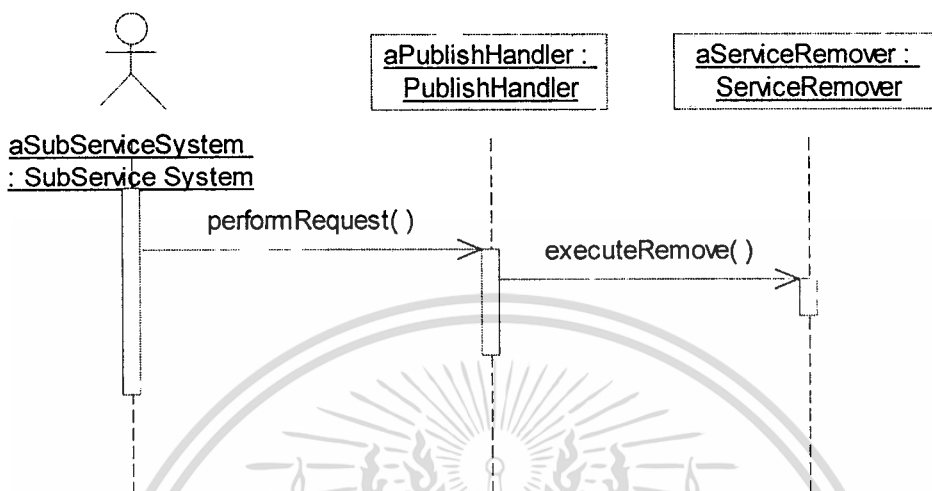


รูปที่ 4.13 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส RegisterService

ตารางที่ 4.9 ตารางอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรมสำหรับยูสเคส RegisterService

คำอธิบาย	ผู้ให้บริการย่อยทำการลงทะเบียนกับ Registry Server
Actor หลัก	SubService System
เงื่อนไขเริ่มต้น	-
ลำดับการทำงานหลัก	1. PublishHandler เรียกใช้เมธอด executePublish ของ ServicePublisher
เงื่อนไขสิ้นสุด	1. หากทำการลงทะเบียนสำเร็จทาง Registry Server จะรายงานความสำเร็จและส่ง Unique Key มาให้ 2. หากทำการลงทะเบียนไม่สำเร็จทาง Registry Server จะแจ้งรายงานความผิดพลาดกลับมา

4.3.10 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส WithdrawService

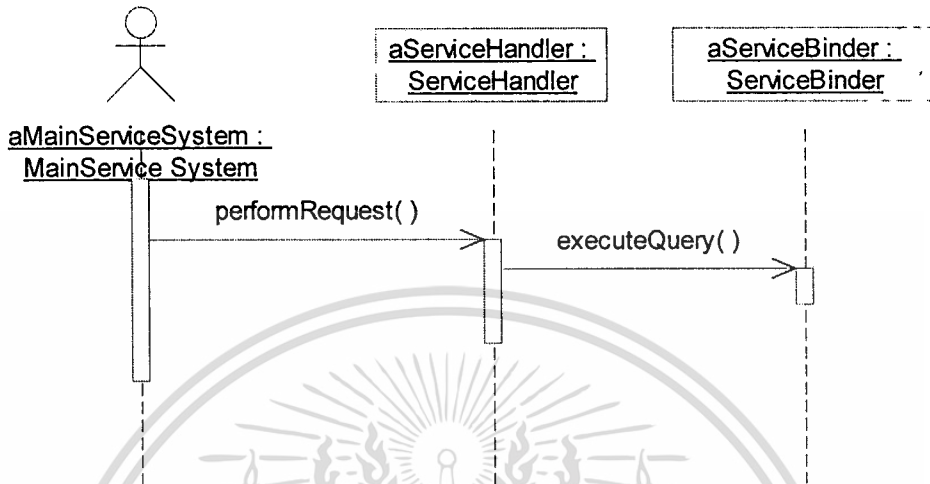


รูปที่ 4.14 ซีควেনซ์ไดอะแกรมสำหรับยูสเคส WithdrawService

ตารางที่ 4.10 ตารางอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรมสำหรับยูสเคส WithdrawService

คำอธิบาย	ผู้ให้บริการย่อยทำการยกเลิกการลงทะเบียนกับ Registry Server
Actor หลัก	SubService System
เงื่อนไขเริ่มต้น	ผู้ให้บริการย่อยต้องเคยทำการลงทะเบียนกับ Registry Server มาก่อน
ลำดับการทำงานหลัก	PublishHandler เรียกใช้เมธอด executeRemove ของ ServiceRemover
เงื่อนไขสิ้นสุด	<ol style="list-style-type: none"> 1. หากทำยกเลิกสำเร็จ Registry Server จะรายงานความสำเร็จกลับมา 2. หากทำการลงทะเบียนไม่สำเร็จ Registry Server จะแจ้งความผิดพลาดกลับมา

4.3.11 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส FindService



รูปที่ 4.15 ซีควเอนซ์ไดอะแกรมสำหรับยูสเคส FindService ของผู้ให้บริการ Registry Server

ตารางที่ 4.11 ตารางอธิบายรายละเอียดประกอบซีควเอนซ์ไดอะแกรมสำหรับยูสเคส FindService

คำอธิบาย	ผู้ให้บริการหลักทำการค้นหาบริการจาก Registry Server
Actor หลัก	MainService System
เงื่อนไขเริ่มต้น	ผู้ให้บริการขอยกข้อเคขทำการลงทะเบียนกับ Registry Server ไว้แล้ว
ลำดับการทำงานหลัก	1. ServiceHandler เรียกใช้เมธอด executeQuery ของ ServiceBinder
เงื่อนไขสิ้นสุด	1. หากการค้นหาสำเร็จทาง Registry Server จะส่งเอกสาร WSDL กลับมา 2. หากการค้นหาไม่สำเร็จทาง Registry Server จะรายงานกลับมา

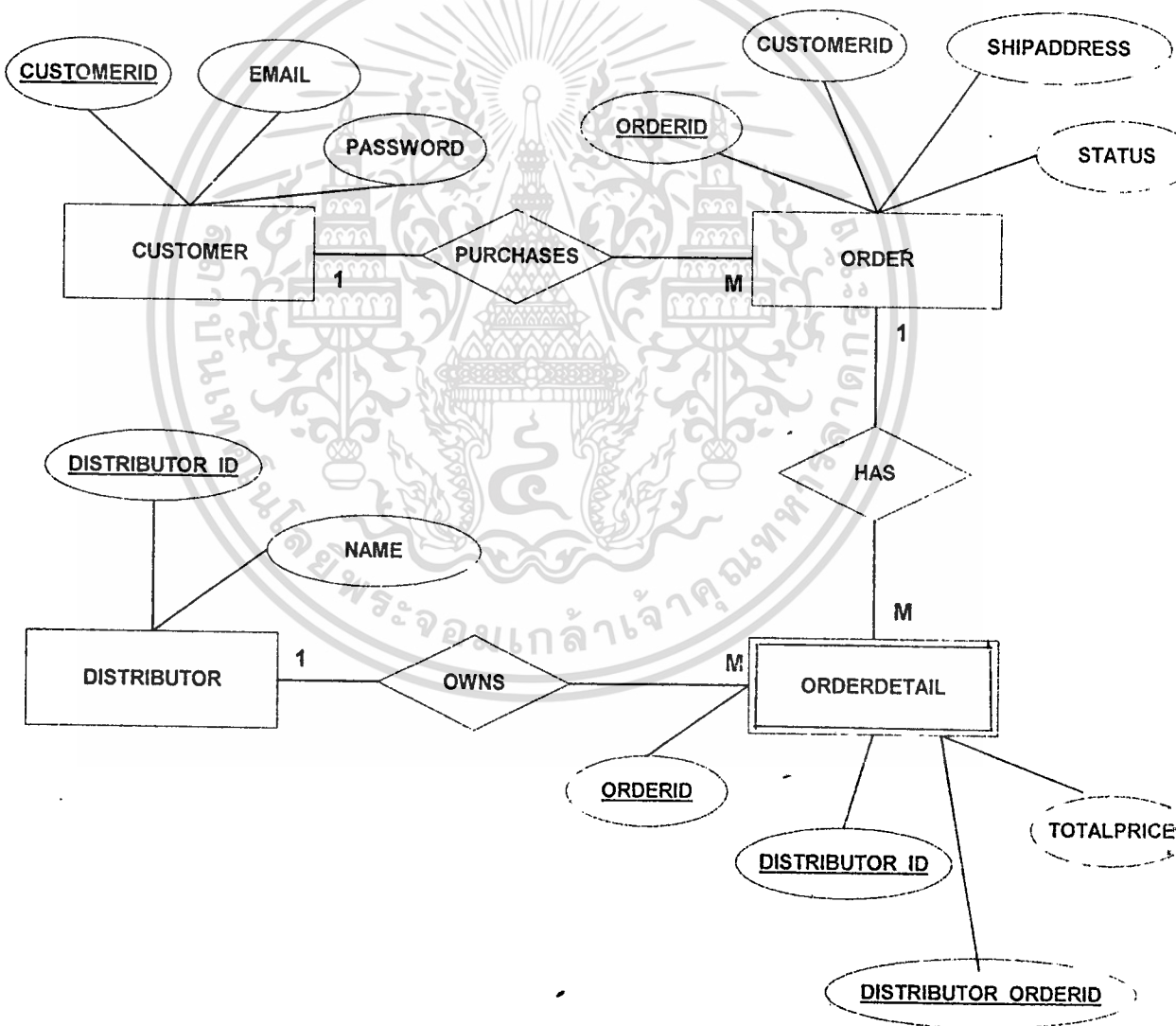
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบฐานข้อมูล

5.1 การออกแบบฐานข้อมูลของผู้ให้บริการหลัก

ฐานข้อมูลของผู้ให้บริการหลักสามารถเขียนเป็น ER Diagram ได้ดังรูป 5.1 โดยรายละเอียดของตารางสามารถแสดงด้วยพจนานุกรมข้อมูลดังตารางที่ 5.1 – 5.4



รูปที่ 5.1 ER Diagram ของผู้ให้บริการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 พจนานุกรมข้อมูล ตาราง CUSTOMER

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างอิงตาราง
CUSTOMERID	รหัสลูกค้า	INTEGER	ไม่ได้	PK	
USERID	รหัสผู้ใช้	VARCHAR	ไม่ได้		
PASSWORD	รหัสผ่าน	VARCHAR	ไม่ได้		

ตารางที่ 5.2 พจนานุกรมข้อมูล ตาราง ORDER

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างอิงตาราง
ORDERID	เลขที่ใบสั่งซื้อ	INTEGER	ไม่ได้	PK	
CUSTOMERID	รหัสลูกค้า	INTEGER	ไม่ได้	FK	CUSTOMER
SHIPADDRESS	ที่อยู่จัดส่งสินค้า	VARCHAR	ไม่ได้		
STATUS (ค่าเริ่มต้นเป็น 'N' คือยังไม่ได้จ่ายเงิน)	สถานะการจ่ายเงิน	CHAR	ไม่ได้		

ตารางที่ 5.3 พจนานุกรมข้อมูล ตาราง ORDERDETAIL

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างอิงตาราง
ORDERID	เลขที่ใบสั่งซื้อ	INTEGER	ไม่ได้	PK,FK	ORDER
DISTRIBUTORID	รหัสร้านค้าย่อย	INTEGER	ไม่ได้	PK,FK	DISTRIBUTOR
DISTRIBUTOR_ ORDERID	เลขที่ใบสั่งซื้อของ ร้านค้าย่อย	INTEGER	ไม่ได้		
TOTALPRICE	ราคารวม	DOUBLE	ไม่ได้		

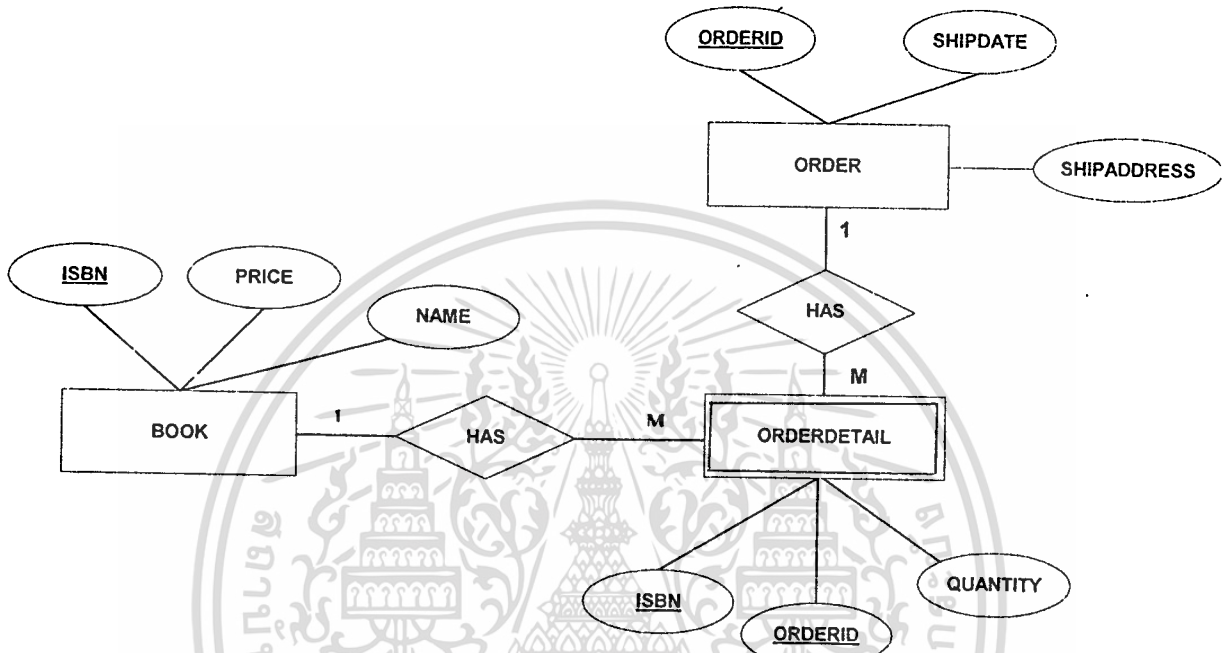
ตารางที่ 5.4 พจนานุกรมข้อมูล ตาราง DISTRIBUTOR

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างอิงตาราง
DISTRIBUTORID	รหัสร้านค้าย่อย	INTEGER	ไม่ได้	PK	
NAME	ชื่อร้านค้าย่อย	VARCHAR	ไม่ได้		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การออกแบบฐานข้อมูลของผู้ให้บริการหนังสือ

ฐานข้อมูลของผู้ให้บริการหนังสือสามารถเขียนเป็น ER Diagram ได้ดังรูป 5.2 โดยรายละเอียดของตารางสามารถแสดงด้วยพจนานุกรมข้อมูลดังตารางที่ 5.5 – 5.7



รูปที่ 5.2 ER Diagram ของผู้ให้บริการหนังสือ

ตารางที่ 5.5 พจนานุกรมข้อมูล ตาราง BOOK

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างถึงตาราง
ISBN	รหัสหนังสือ	VARCHAR	ไม่ได้	PK	
NAME	ชื่อหนังสือ	VARCHAR	ไม่ได้		
PRICE	ราคา	DOUBLE	ไม่ได้		

ตารางที่ 5.6 พจนานุกรมข้อมูล ตาราง ORDER

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างถึงตาราง
ORDERID	เลขที่ใบสั่งซื้อ	INTEGER	ไม่ได้	PK	
SHIPDATE	วันส่งของ	DATETIME	ได้		
SHIPADDRESS	สถานที่ส่งของ	VARCHAR	ได้		

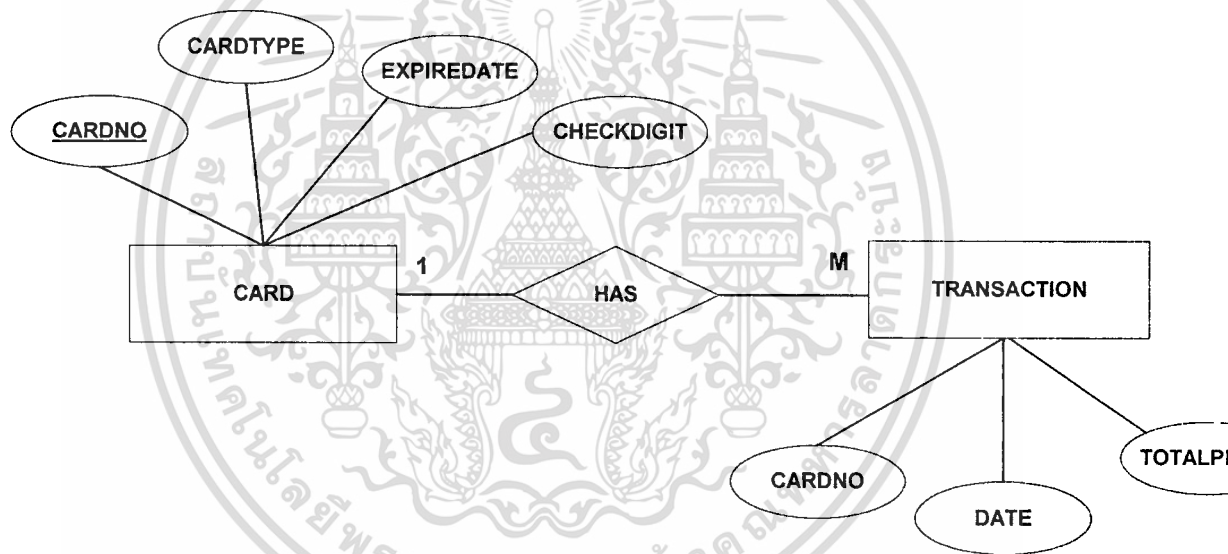
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.7 พจนานุกรมข้อมูล ตาราง ORDERDETAIL

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างอิงตาราง
ORDERID	เลขที่ใบสั่งซื้อ	INTEGER	ไม่ได้	PK,FK	ORDER
ISBN	รหัสหนังสือ	VARCHAR	ไม่ได้	PK,FK	CUSTOMER
QUANTITY	จำนวนที่สั่งซื้อ	INTEGER	ไม่ได้		

5.3 การออกแบบฐานข้อมูลของผู้ให้บริการธนาคาร

ฐานข้อมูลของผู้ให้บริการหลักสามารถเขียนเป็น ER Diagram ได้ดังรูป 5.3 โดยรายละเอียดของตารางสามารถแสดงด้วยพจนานุกรมข้อมูลดังตารางที่ 5.8 – 5.9



รูปที่ 5.3 ER Diagram ของผู้ให้บริการหลักธนาคาร

ตารางที่ 5.8 พจนานุกรมข้อมูล ตาราง CARD

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างอิงตาราง
CARDNO	หมายเลขบัตรเครดิต	VARCHAR	ไม่ได้	PK	
CARDTYPE	ประเภทบัตร	VARCHAR	ไม่ได้		
EXPIREDATE	วันหมดอายุ	DATETIME	ไม่ได้		
CHECKDIGIT	เลข 3 ตัวท้ายของบัตร	VARCHAR	ไม่ได้		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.9 พจนานุกรมข้อมูล ตาราง TRANSACTION

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	เป็นค่าว่าง	KEY	อ้างอิงตาราง
CARDNO	หมายเลขบัตรเครดิต	VARCHAR	ไม่ได้	FK	CARD
DATE	วันที่ทำธุรกรรม	DATETIME	ไม่ได้		
TOTALPRICE	ราคา	DOUBLE	ไม่ได้		

5.4 ข้อสมมติ (Assumption)

1. ผู้ให้บริการหนังสือให้บริการกับผู้ให้บริการหลักเพียงรายเดียว
2. ผู้ให้บริการหนังสือ ไม่มีการตรวจสอบสินค้าคงคลัง เมื่อผู้ให้บริการหลักสั่งของมา จะบันทึกข้อมูลรายการสั่งซื้อนั้นและส่งข้อมูลยืนยันการสั่งซื้อกลับไป ซึ่งถือว่าของมีในคลังเสมอ
3. ผู้ให้บริการหนังสือขายหนังสือเพียงรายการหนังสือละ 1 ตำรา
4. ผู้ให้บริการธนาคารจะไม่มีการตรวจสอบสถานะของบัตรเครดิต เมื่อผู้ให้บริการหลักสั่งชำระเงินมาจะทำกรบันทึกข้อมูลการชำระเงินนั้นและส่งข้อมูลยืนยันการชำระเงินกลับไป ซึ่งถือว่าสถานะของบัตรสามารถจ่ายได้เสมอ

บทที่ 6

การพัฒนาระบบ

6.1 เครื่องมือและภาษาที่ใช้พัฒนาระบบ

1. Java 2 Platform, Enterprise Edition (J2EE) 1.4 SDK
 - Sun Java System Application Server Platform Edition 8 Update 1
 - Java 2 Platform, Standard Edition (J2SE) 1.4.2
 - NetBeans IDE 3.6
2. Java Web Services Developer Pack (JWSDP) 1.4
 - XML and Web Services Security v1.0 EA3
 - XML Digital Signatures 1.0 EA
 - Java Architecture for XML Binding (JAXB) v1.0.3 FCS
 - Java API for XML Processing (JAXP) v1.2.6 FCS
 - Java API for XML Registries (JAXR) v1.0.6 FCS
 - Java API for XML-based RPC (JAX-RPC) v1.1.2 FCS
 - SOAP with Attachments API for Java (SAAJ) v1.2.1 FCS
 - JavaServer Pages Standard Tag Library (JSTL) v1.1 FCS
 - Java WSDP Registry Server v1.0_06 EA
 - Ant Build Tool 1.5.4 FCS
 - WS-I Attachments Sample Application 1.0 EA
3. Rational Rose 2000 Enterprise Edition
4. Macromedia Dreamweaver MX
5. Hardware
 - CPU : Pentium IV 2.4 GHz
 - RAM : 256 MB
 - Hard Disk : 80 GB
 - Network Interface : 100BASE-TX/10BASE-T

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การพัฒนาระบบของผู้ให้บริการหนังสือ (JAX-RPC Book Service)

6.2.1 การลงทะเบียนบริการกับ Registry Server

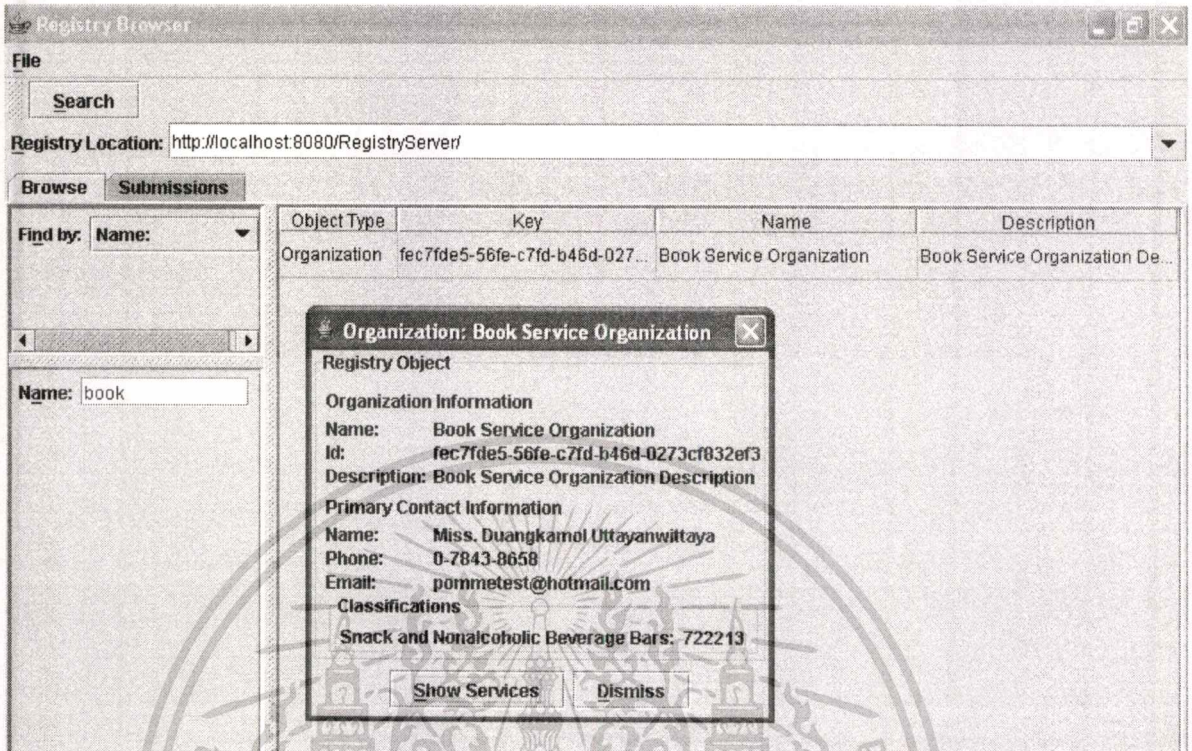
ทำการการเขียน Java Application เพื่อทำหน้าที่ในการลงทะเบียนบริการกับ Registry Server (ServicePublisher.java) โดยอ่านข้อมูลต่างๆที่จำเป็นต้องใช้ในการลงทะเบียนบริการจาก Property File (BookService.properties) สำหรับข้อมูลที่ต้องใช้ในการลงทะเบียนบริการมีดังนี้

ตารางที่ 6.1 ข้อมูลที่ต้องใช้ในการลงทะเบียนบริการกับ Registry Server

ชื่อตัวแปร	ค่า
Query URL	http://localhost:8080/RegistryServer/
Publish URL	http://localhost:8080/RegistryServer/
Registry Server Username	testuser
Registry Server Password	testuser
Organization Name	Book Service Organization
Organization Description	Book Service Organization Description
Contact Person Name	Miss. Duangkamol Uttayanwittaya
Contact Person Phone	0-7843-8658
Contact Person Email Address	pometest@hotmail.com
Classification Scheme	ntis-gov:naics:1997
Classification Name	Snack and Nonalcoholic Beverage Bars
Classification Value	722213
Service Name	Book Service
Service Description	Book Service Description
Service Binding URI	http://localhost:8080/jaxrpc_bookservice
Service Binding Description	Book Service Binding Description

หลังจากการรันโปรแกรม ServicePublisher.java เสร็จสมบูรณ์เราจะได้รับค่า key จาก Registry Server (เช่น fec7fde5-56fe-c7fd-b46d-0273cf832ef3) และ JAX-RPC Book Service จะไปอยู่ใน Registry Server เรียบร้อยแล้ว หากเราทำการค้นหาบริการโดยกำหนด keyword เป็น คำว่า “book” ก็จะเจอบริการที่เราเพิ่งลงทะเบียนไป ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1 การค้นหาบริการใน Registry Server

6.2.2 สร้างคลาสพื้นฐานของบริการ

ทำการเขียน Java Class ที่เป็นคลาสพื้นฐานของบริการ ได้แก่

- คลาส Customer (CustomerBean.java)
- คลาส Order (OrderBean.java)
- คลาส OrderItem (OrderItemBean.java)
- คลาส PriceList (PriceListBean.java)
- คลาส PriceListItem (PriceListItemBean.java)
- คลาส OrderConfirmation (OrderConfirmationBean.java)

6.2.3 การจัดเตรียมเมธอดให้ให้สามารถถูกเรียกใช้แบบทางไกลได้

- ทำการเขียน Service Endpoint Interface (SEI) เพื่อกำหนดรูปแบบในการเรียกใช้งานเมธอดแบบทางไกลสำหรับเครื่องลูกข่าย (BookServiceIF.java) ซึ่งภายในไฟล์จะประกาศเพียงแค่ว่ามีเมธอดอะไรให้เรียกใช้บ้าง ต้องส่งพารามิเตอร์อย่างไร และจะได้ค่ากลับคืนเป็นอะไร โดยตัว SEI ต้องสืบทอดคลาส `java.rmi.Remote` และ โยน `java.rmi.RemoteException`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการเขียน Implementation Class เพื่อ Implement ตัว Service Endpoint Interface ดังกล่าว (BookServiceImpl.java) ซึ่งภายในไฟล์ประกอบด้วย รายละเอียดการทำงานของเมธอด getPriceList และ placeOrder

6.2.4 การสร้างเอกสาร WSDL เพื่อใช้อธิบายบริการ

ใช้ wscompile Tool ช่วยสร้าง WSDL (BookService.wsdl)

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="BookService" targetNamespace="urn:Foo" xmlns:tns="urn:Foo"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
+ <types>
- <message name="BookServiceIF_getPriceList" />
- <message name="BookServiceIF_getPriceListResponse">
  <part name="result" type="tns:PriceListBean" />
</message>
- <message name="BookServiceIF_placeOrder">
  <part name="OrderBean_1" type="tns:OrderBean" />
</message>
- <message name="BookServiceIF_placeOrderResponse">
  <part name="result" type="tns:OrderConfirmationBean" />
</message>
+ <portType name="BookServiceIF">
+ <binding name="BookServiceIFBinding" type="tns:BookServiceIF">
+ <service name="BookService">
</definitions>
```

รูปที่ 6.2 ไฟล์ WSDL ของ JAX-RPC Book Service

6.2.5 การสร้างไฟล์ .war (BookService.war) และ โหลดโปรแกรมเข้าทำงานใน J2EE Server ในที่นี้เราใช้ Sun Java System Application Server Platform Edition 8 ซึ่งสามารถใช้ Deploytool ช่วยในการทำ Packaging และ Deployment

6.2.6 การสร้าง Test Client สำหรับทดสอบการเรียกเมธอดแบบทางไกล

ปกติเวลาที่ Client คุยกับ Server จะไม่คุยโดยตรง โดยทางฝั่ง Client จะมีการสร้าง Class ที่เรียกว่า Stub ขึ้นมา และเวลา Client ร้องขอบริการก็จะเรียกผ่าน Stub นี้ (BookServiceWebClient.java)

6.2.7 เขียนโปรแกรมเพื่อรอรับ Request การขอข้อมูลราคาจากผู้ให้บริการหลัก ซึ่งเมื่อรับ Request มาแล้วก็จะส่ง SOAP Message ซึ่งภายในบรรจุข้อความรายการราคา (PriceList) ส่งกลับไปให้ผู้ให้บริการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การพัฒนาระบบของผู้ให้บริการหลัก (Main Service)

ส่วนงาน Back End

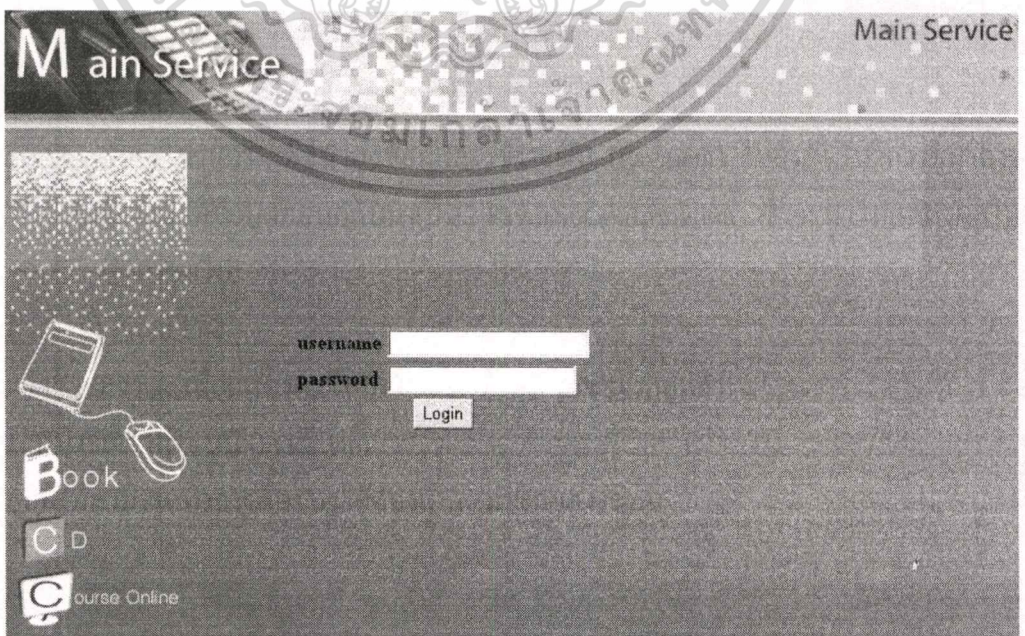
- เขียนโปรแกรมเพื่อค้นหาบริการใหม่จาก Registry Server โดย Registry Server จะส่งตอบกลับมาเป็นกลุ่มของบริการ ซึ่งผู้ให้บริการหลักจะติดต่อไปยังแต่ละบริการต่อไป โดยดูรายละเอียดการขอใช้บริการจาก ไฟล์ WSDL
- เขียนโปรแกรมเพื่อขอใช้บริการเมธอดแบบทางไกลเพื่อขอดูรายการราคาสินค้า โดยจะได้ข้อมูลตอบกลับมาเป็น PriceList ซึ่งจะนำไปแสดงผลให้ผู้ค้าต่อไป
- เขียนโปรแกรมเพื่อขอใช้บริการเมธอดแบบทางไกลเพื่อขอสั่งซื้อสินค้า โดยจะต้องส่งข้อมูล Order ไปให้และจะได้ข้อมูลตอบกลับมาเป็น OrderConfirmation ซึ่งจะนำไปแสดงผลให้ผู้ค้าต่อไป
- เขียนโปรแกรมเพื่อนำส่งข้อมูลการชำระเงิน (Payment) ไปยังผู้ให้บริการธนาคาร โดยจะได้ข้อมูลตอบกลับมาเป็น PaymentConfirmation ซึ่งจะนำไปแสดงผลให้ผู้ค้าต่อไป

ส่วนงาน Front End

สร้าง JSP Page เพื่อแสดงผลข้อมูลและรองรับคำร้องขอจากลูกค้า

1. หน้าจอเข้าสู่ระบบ (index.jsp)

ลูกค้าที่สามารถทำการซื้อของได้ต้องเป็นสมาชิกของระบบ

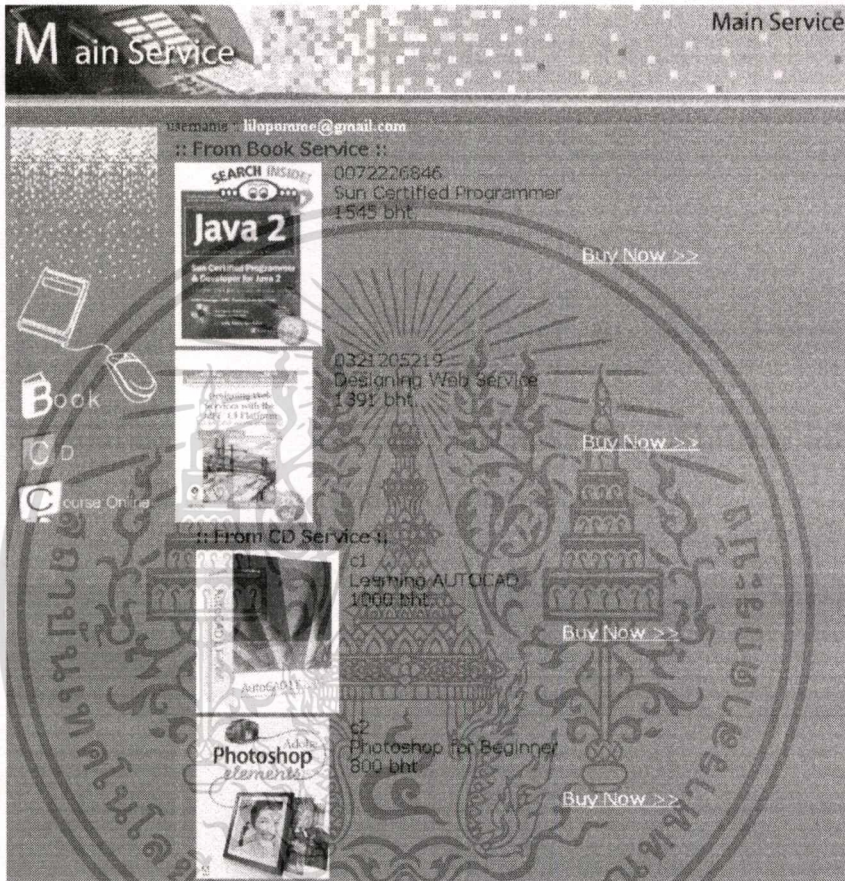


รูปที่ 6.5 หน้าจอเข้าสู่ระบบของผู้ให้บริการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้าจอเลือกรายการสินค้า (main.jsp)

โดยข้อมูลที่นำมาแสดงจะเป็นการดึงข้อมูลจากเว็บเซอร์วิสของผู้ให้บริการหนังสือและผู้ให้บริการสื่อการเรียน (ซีดี) หากต้องการซื้อสินค้าให้คลิกที่ Buy Now

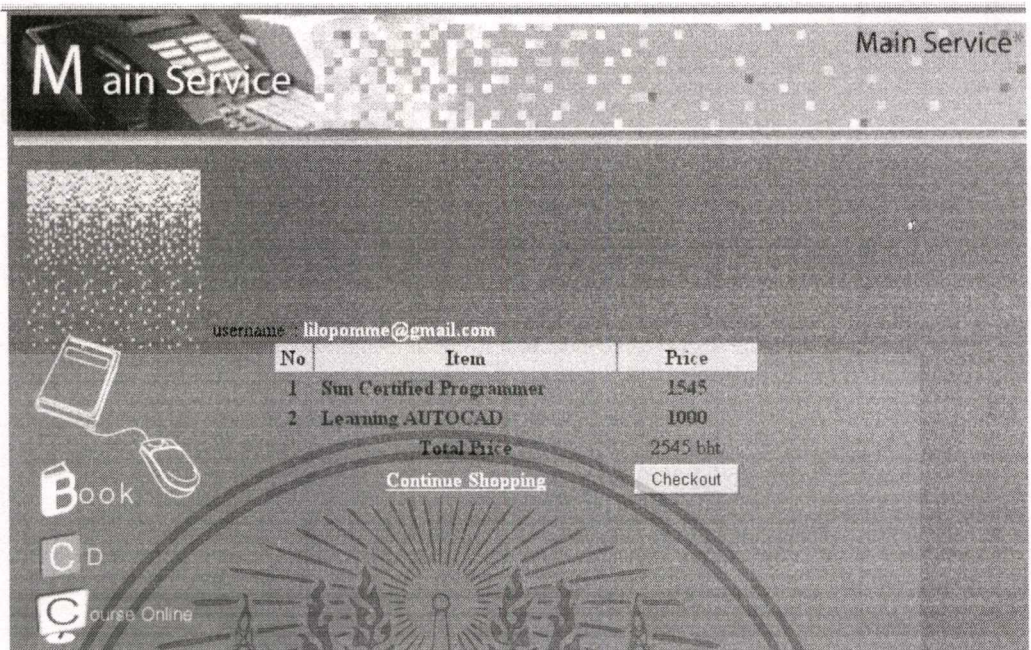


รูปที่ 6.6 หน้าจอเลือกรายการสินค้าของผู้ให้บริการหลัก

3. หน้าจอสั่งซื้อสินค้า (cart.jsp)

เมื่อลูกค้าทำการเลือกสินค้า รายการสินค้าจะไปรวมอยู่ในตะกร้าซื้อของลูกค้า หากลูกค้าต้องการเลือกซื้อสินค้าชนิดอื่นต่อให้คลิกที่ Continue Shopping หากลูกค้าต้องการชำระเงินให้คลิกที่ CheckOut

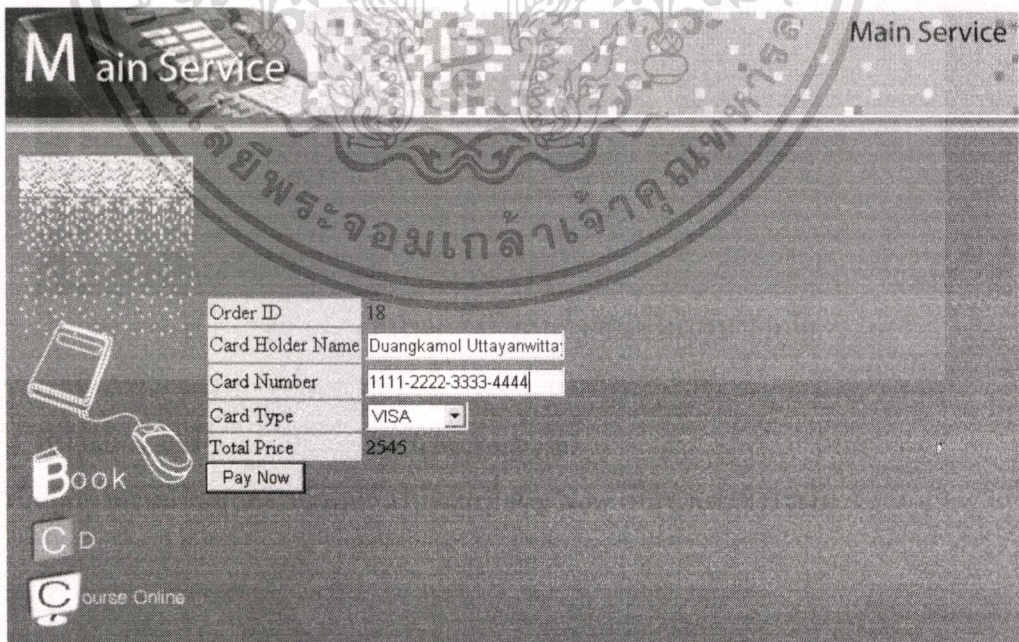
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.7 หน้าจอสั่งซื้อสินค้าของผู้ให้บริการหลัก

4. หน้าจอกรอกข้อมูลการชำระเงิน (checkout.jsp)

เมื่อลูกค้ากรอกข้อมูลครบถ้วนแล้วให้คลิกที่ Pay Now เพื่อทำการชำระเงิน

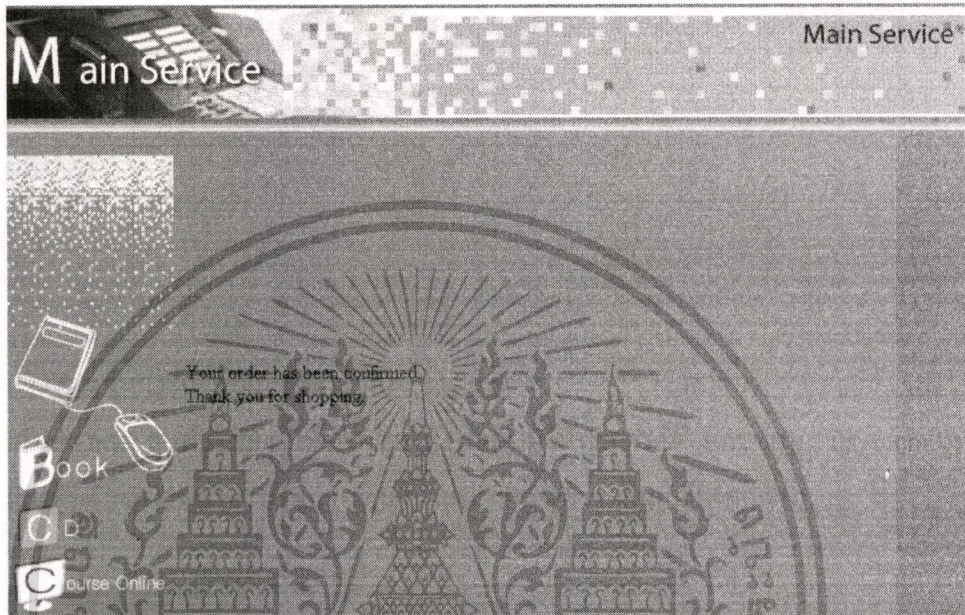


รูปที่ 6.8 หน้าจอกรอกข้อมูลการชำระเงินของผู้ให้บริการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หน้าจอยืนยันการสั่งซื้อและชำระเงิน (pay.jsp)

เมื่อข้อมูล Order ถูกส่งไปยังผู้ให้บริการหนังสือกับผู้ใช้บริการซีดี และข้อมูล Payment ถูกส่งไปยังผู้ให้บริการธนาคารเรียบร้อยแล้ว จะปรากฏหน้าจอยืนยันให้ลูกค้าทราบ



รูปที่ 6.9 หน้าจอยืนยันการสั่งซื้อและชำระเงินของผู้ให้บริการหลัก

6.4 การพัฒนาระบบของผู้ให้บริการธนาคาร (SAAJ Bank Service)

6.4.1 กำหนดรูปแบบข้อมูลที่ผู้ให้บริการหลักจะส่งให้กับผู้ให้บริการธนาคาร

ในที่นี้เราสมมติว่าผู้ให้บริการธนาคารเคยมีการติดต่อกับผู้ให้บริการหลักมาก่อนแล้ว ซึ่งจะมีการกำหนดเป็น DTD สำหรับเอกสาร XML ไว้เรียบร้อยแล้ว และส่งมาในรูปแบบของ SOAPMessage ข้อมูลที่ทางผู้ให้บริการหลักจะส่งมาให้ผู้ให้บริการธนาคารคือข้อมูลการชำระเงิน (Payment) ซึ่งเรากำหนดให้ Payment เป็น Root Element และภายในประกอบด้วย Element ย่อยดังนี้

- cardno คือ หมายเลขบัตรเครดิต
- cardtype คือ ประเภทบัตรเครดิต
- expiredate คือ วันหมดอายุบัตร
- total คือ จำนวนเงิน
- shopid คือ รหัสร้านค้าของผู้ให้บริการหลัก
- orderid คือ รหัสใบสั่งซื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

- <payment>
  <cardno>"1111-2222-3333-4444"</cardno>
  <cardtype>"VISA"</cardtype>
  <expiredate>"09-2005"</expiredate>
  <total>"1000.00"</total>
  <shopid>"1"</shopid>
  <orderid>"1234"</orderid>
</payment>

```

รูปที่ 6.10 รูปแบบข้อมูลการชำระเงิน

6.4.2 กำหนดรูปแบบข้อมูลที่ผู้ให้บริการธนาคารจะส่งตอบกลับผู้ให้บริการหลัก

เมื่อผู้ให้บริการธนาคารทำการประมวลผลการชำระเงินเรียบร้อยแล้วจะทำการส่งข้อมูลยืนยันการชำระเงินกลับมาให้ผู้ให้บริการหลัก (payment-confirmation) ซึ่งเรากำหนดให้ payment-confirmation เป็น Root Element และภายในประกอบด้วย Element ย่อยดังนี้

- shopid คือ รหัสร้านค้าของผู้ให้บริการหลัก
- orderid คือ รหัสใบสั่งซื้อ
- approve คือ ผลการชำระเงิน

```

- <payment-confirmation>
  <shopid>"1"</shopid>
  <orderid>"1234"</orderid>
  <approve>"YES"</approve>
</payment-confirmation>

```

รูปที่ 6.11 รูปแบบข้อมูลยืนยันการชำระเงิน

6.4.3 เขียนโปรแกรมเพื่อรองรับ Request ที่จะส่งมาจากผู้ให้บริการหลัก ซึ่งเมื่อรับ Request มาแล้วก็จะส่ง SOAP Message ซึ่งภายในบรรจุข้อความการยืนยันการชำระเงินส่งกลับไปให้ผู้ให้บริการหลัก

6.4.5 สร้างรายงาน Back Office เพื่อตรวจสอบการชำระเงินจากผู้ให้บริการหลัก

Bank
Service

Bank Service

:: Payments Report ::

Customer Name	Card No	Card Type	Total Price	Shop Name
Donlaporn	1111-1111-1111-1111	VISA	8569	Main Service 1
Nilubon	9999-9999-9999-9999	MASTER	2345	Main Service 1
Pomme	3333-5555-5555-7777	MASTER	2345	Main Service 1
Duangkamol Uttayanwittaya	1111-2222-3333-4444	VISA	2545	Main Service 1

รูปที่ 6.12 หน้าจอตรวจสอบการชำระเงินจากผู้ให้บริการหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการดำเนินงาน

7.1 ประโยชน์ที่ได้รับ

ระบบขายหนังสือและสื่อการเรียนโดยใช้จาวาเว็บเซอร์วิสสามารถให้บริการดังต่อไปนี้ได้

1. ให้บริการ การแสดงผลข้อมูลของ สื่อการเรียนประเภทหนังสือ
2. ให้บริการ การสั่งซื้อ สื่อการเรียนประเภทหนังสือ
3. ให้บริการ การชำระเงิน สื่อการเรียนประเภทหนังสือ
4. ลักษณะการทำงาน โดยรวมของระบบเป็นการทำงานร่วมกันของแต่ละเว็บเซอร์วิสรวมกับเว็บแอปพลิเคชันหลักเพื่อตอบสนองข้อมูลไปยังลูกค้า

ซึ่งระบบขายหนังสือและสื่อการเรียน โดยใช้จาวาเว็บเซอร์วิสดังกล่าวมีประโยชน์ดังต่อไปนี้

1. ลูกค้าได้รับความสะดวกสบาย ประหยัดเวลา ทำการติดต่อ ณ จุดเดียวก็สามารถได้รับบริการจากผู้ให้บริการหลายแห่งเช่น ผู้ให้บริการหนังสือ ผู้ให้บริการสื่อการเรียน และผู้ให้บริการบทเรียนออนไลน์ โดยลูกค้าได้รับบริการต่างๆ เช่น การดูราคาสินค้า การสั่งซื้อสินค้าและการชำระเงิน
2. เป็นการเพิ่มช่องทางการจำหน่ายให้กับผู้ให้บริการย่อย เพิ่มลูกค้า เพิ่มรายได้
3. ผู้พัฒนาระบบได้รับความรู้ในเรื่องต่างๆดังนี้
 - 3.1 เว็บเซอร์วิส
 - 3.2 การเขียนโปรแกรมภาษาจาวาและเทคโนโลยีเชิงวัตถุ
 - 3.3 การวิเคราะห์และออกแบบโปรแกรมโดยใช้ UML

7.2 ข้อจำกัดของระบบ

ระบบขายหนังสือและสื่อการเรียน โดยใช้จาวาเว็บเซอร์วิสที่พัฒนาขึ้นยังมีข้อจำกัดในเรื่องของการใช้งานดังต่อไปนี้

1. การติดต่อกับ Registry Server

การลงทะเบียนบริการของผู้ให้บริการย่อยและการค้นหาบริการของผู้ให้บริการหลัก ทั้ง 2 ส่วนนี้ยังไม่สามารถทำงานแบบอัตโนมัติได้ เป็นเพียงการเรียกใช้งาน Java Application เพื่อทำงานดังกล่าวเท่านั้น ทำให้มีข้อเสียคือข้อมูลที่ได้รับมาจาก Registry Server จะไม่ทันสมัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การให้บริการการชำระเงินแก่ลูกค้า

ในการให้บริการการชำระเงินยังไม่สามารถให้ลูกค้าเลือกธนาคารได้ ระบบที่พัฒนาผูกติดอยู่กับธนาคารเดียว ซึ่งได้มีการตกลงในเรื่องของข้อมูลในการติดต่อไว้ล่วงหน้าแล้ว ทำให้ลูกค้าที่ไม่ได้เป็นสมาชิกของธนาคารดังกล่าวไม่สามารถทำการสั่งซื้อสินค้าได้ (ทำได้เพียงดูข้อมูลและราคาสินค้าเท่านั้น)

3. การตรวจสอบสินค้าคงคลัง

ผู้ให้บริการหนังสือไม่มีการตรวจสอบสินค้าคงคลัง เมื่อผู้ให้บริการหลักส่งของมา จะบันทึกข้อมูลรายการสั่งซื้อนั้นและส่งข้อมูลยืนยันการสั่งซื้อกลับไป ซึ่งถือว่าของมีในคลังเสมอ

4. การตรวจสอบความถูกต้องของบัตรเครดิต

ผู้ให้บริการธนาคารจะไม่มีการตรวจสอบสถานะของบัตรเครดิต เมื่อผู้ให้บริการหลักส่งชำระเงินมา จะทำการบันทึกข้อมูลการชำระเงินนั้นและส่งข้อมูลยืนยันการชำระเงินกลับไป ซึ่งถือว่าสถานะของบัตรสามารถจ่ายได้เสมอ

7.3 ปัญหาและอุปสรรคที่พบ

ระหว่างการดำเนินงานได้พบปัญหาและอุปสรรคในเรื่องของเทคโนโลยีและความรู้พื้นฐานที่จำเป็น ซึ่งเทคโนโลยีในการทำเว็บเซอร์วิสที่ใช้เป็นเรื่องใหม่ รวมทั้งต้องใช้ความรู้พื้นฐานในเรื่องของการเขียนโปรแกรมภาษาจาวาและเทคโนโลยีเชิงวัตถุค่อนข้างมาก ทำให้การออกแบบและพัฒนาระบบเป็นไปได้ยากและใช้เวลานาน

7.4 แนวทางในการพัฒนาต่อ

ระบบขายหนังสือและสื่อการเรียน โดยใช้จาวาเว็บเซอร์วิสยังสามารถนำไปพัฒนาต่อเพื่อเพิ่มประสิทธิภาพในการทำงานได้ดังต่อไปนี้

1. เพิ่มช่องทางการให้บริการเพื่อให้ลูกค้าสามารถรับบริการผ่านทางอุปกรณ์อื่นๆ ได้ เช่น โทรศัพท์มือถือ
2. พัฒนาเพิ่มเติมเชื่อมโยงไปยังระบบเว็บแอปพลิเคชันและระบบดั้งเดิมขององค์กร เพื่อให้เกิดประโยชน์สูงสุดในด้านการบริการข้อมูล

บรรณานุกรม

นรินทร์ โอพารกิจอนันต์. 2547. **J2EE & XML** จาวาระดับองค์กร. กรุงเทพฯ : เดคิซูกิคอมเพนีต.

เลอชาติ ธรรมเสถียร. 2545. “การพัฒนาเว็บเซอร์วิสเพื่อการคำนวณคุณค่าทางโภชนาการของอาหาร.” รายงานโครงการพัฒนาระบบงาน คณะเทคโนโลยีสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

วีระศักดิ์ ชิงฉาวร. 2546. **Enterprise JavaBeans**. กรุงเทพฯ : ซีเอ็ด บุ๊คชั่น.

วีระพล นรکانต์กร. 2546. “การใช้จาวาเว็บเซอร์วิสกับระบบการขายดอกไม้ผ่านทางอินเทอร์เน็ต.” รายงานโครงการพัฒนาระบบงาน คณะเทคโนโลยีสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สรารุช อ้อยศรีสกุล. 2544. **ถอดรหัส .Net + Web Services**. กรุงเทพฯ : วิตตี้ กรู๊ป.

Sun Microsystems. 2004a. **The J2EE™ 1.4 Tutorial**.

[Online]. Available: <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>

Sun Microsystems. 2004b. **The Java™ Web Services Tutorial**.

[Online]. Available: <http://java.sun.com/webservices/docs/1.4/tutorial/doc/index.html>

Sun Microsystems. 2004c. **Java Technology and Web Services**.

[Online]. Available: <http://java.sun.com/webservices/index.jsp>

ประวัติผู้เขียน

ชื่อผู้เขียน

นางสาวดวงกมล อุทยานวิทยา

วุฒิการศึกษาระดับปริญญาตรี

วท. บ (วิทยาการคอมพิวเตอร์)

จุฬาลงกรณ์มหาวิทยาลัย

ประวัติการทำงาน

Professional Computer Co., Ltd.

(Web Programmer)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้