

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนาระบบการจัดซื้อโดยใช้ JAX

The Development of a Procurement System Using JAX



วัน เดือน ปี.....	03 ก.พ. 2550
เลขทะเบียน.....	02147
เลขเรียกหนังสือ.....	วท. ส 335 ท 2546
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

๕๑๑๔

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบการจัดซื้อโดยใช้ JAX
นักศึกษา	นางสาวมีณทนา ถนนอมพงษ์ชาติ
อาจารย์ที่ปรึกษา	ดร. ภัทรชัย ลลิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

เว็บเซอร์วิสเป็นตัวกลางในการติดต่อระหว่างแอปพลิเคชัน ด้วยการตอบสนองคำร้องขอที่ส่งมาจากระบบอื่น ทั้งจากระบบภายใน หรือภายนอก ผ่านทางอินเทอร์เน็ต และช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอปพลิเคชันที่ต่างกันเป็นไปโดยง่าย และเพื่อให้ธุรกิจการซื้อ-ขาย มีความยืดหยุ่น และทำงานร่วมกันได้ง่ายขึ้น จึงได้นำเว็บเซอร์วิสเข้ามาพัฒนาระบบสั่งซื้อ โดยเริ่มตั้งแต่การค้นหาข้อมูลผู้ขาย ข้อมูลรายการวัตถุดิบ การสั่งซื้อวัตถุดิบ จนถึงการตอบรับจากฝั่งผู้ขาย ด้วยการประยุกต์ใช้ Java API for XML (JAX) เขียนโปรแกรมจัดการกับข้อมูลทางธุรกิจ และเว็บแอปพลิเคชันที่เขียนด้วยภาษาจาวาให้ทำงานร่วมกันได้

Title The Development of a Procurement System Using JAX
Student Miss Muntana Thanompongchat
Advisor Dr. Pattarachai Latitrojwong
Level of Study Master of Science in Information Technology
Major Information Science
Academic Year 2003

ABSTRACT

Web Services help applications to be able to communicate to each other. They can respond to the request sent from a local or global system so that it can easily access information from different application. The development of purchase orders and sale system with web services help to flexibility and coordinate between global business. First, searching vendor data, material list, making purchase order document until to response to purchase order by using Java API for XML for programming business data management and Web application written in Java so that they can work together.

กิตติกรรมประกาศ

1. ขอขอบพระคุณ บิดา มารดา ที่ให้โอกาสในการศึกษาเล่าเรียนอย่างเต็มที่ รวมทั้งคอยให้กำลังใจ ช่วยเหลือ และให้คำปรึกษาต่าง ๆ
2. ขอขอบพระคุณ ดร.ภัทรชัย สถิตโรจน์วงศ์ อาจารย์ที่ปรึกษาเป็นอย่างมาก ที่ได้กรุณาให้คำปรึกษา คำแนะนำ และแก้ไขในสิ่งที่บกพร่องในการพัฒนาโครงการนี้
3. ขอขอบพระคุณ คณาจารย์ทุกท่านที่ให้ความรู้ต่าง ๆ เพื่อนำความรู้มาใช้ในการพัฒนาโครงการนี้
4. ขอขอบพระคุณ เจ้าหน้าที่ของคณะเทคโนโลยีสารสนเทศทุกคนที่ให้ความช่วยเหลือในด้านบริการต่าง ๆ เป็นอย่างดี
5. ขอขอบคุณ คุณสุเมธ กาญจ โนภาส เพื่อน ๆ และญาติ ๆ ที่ให้กำลังใจ และช่วยเหลือในการพัฒนาโครงการนี้



บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญตาราง	V
สารบัญรูป	VI
บทที่	
1. บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 ปัญหา	1
1.3 วัตถุประสงค์ของโครงการ	2
1.4 ขอบเขตของระบบ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 รายละเอียดของแต่ละบท	2
2. ทฤษฎีเทคโนโลยีที่เกี่ยวข้อง	4
2.1 เว็บเซอร์วิส	4
2.1.1 ความหมายของเว็บเซอร์วิส	4
2.1.2 สถาปัตยกรรมของเว็บเซอร์วิส	5
2.2 เครื่องมือที่ใช้พัฒนาเว็บเซอร์วิส	5
2.2.1 เทคโนโลยีหลักของเว็บเซอร์วิส [2]	5
2.2.2 เครื่องมือในการพัฒนาเว็บเซอร์วิส	6
2.3 สถาปัตยกรรมและเอพีไอของจาวาสำหรับเอ็กซ์เอ็มแอล	7
2.3.1 JAXP	8
2.3.2 JAXB	13
2.3.3 JAXM	17
2.3.4 JAXR	20

2.3.5 JAX-RPC	23
3. การวิเคราะห์และออกแบบระบบ	26
3.1 ระบบการจัดซื้อ โดยใช้เว็บเซอร์วิส	26
3.2 One Supplier Web Services และ Two Supplier Web Services	28
3.3 Plastic Server	29
3.4 ยูสเคสไคอะแกรม	30
3.5 คลาสไคอะแกรม	31
3.6 ซีเควนซ์ไคอะแกรม	33
4. การออกแบบฐานข้อมูล	40
4.1 ฐานข้อมูลของซัพพลายเออร์รายที่ 1	40
4.2 ฐานข้อมูลของซัพพลายเออร์รายที่ 2	40
4.3 ฐานข้อมูลของ Plastic Server	42
5. การพัฒนาระบบ	45
5.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ	45
5.2 หน้าจอการทำงานของระบบ	46
6. บทสรุป	50
6.1 สรุปผลการออกแบบและพัฒนาระบบ	50
6.2 ประโยชน์ที่ได้รับจากการพัฒนาระบบ	50
6.3 ปัญหาและอุปสรรค	50
6.4 ข้อเสนอแนะ	51
บรรณานุกรม	52
ประวัติผู้เขียน	53

สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงรายละเอียดของตาราง CUSTOMER	41
4.2 แสดงรายละเอียดของตาราง ADDRESS	41
4.3 แสดงรายละเอียดของตาราง ORDER	41
4.4 แสดงรายละเอียดของตาราง ORDER_ITEM	42
4.5 แสดงรายละเอียดของตาราง MATERIAL	42
4.6 แสดงรายละเอียดของตาราง CUSTOMER	43
4.7 แสดงรายละเอียดของตาราง PURCHASE_ORDER	43
4.8 แสดงรายละเอียดของตาราง PURCHASE_ORDER_ITEM	44

สารบัญรูป

รูปที่	หน้า
2.1 สถาปัตยกรรมเว็บเซอร์วิส	5
2.2 แผนภาพการแปลงของ XSLT	13
2.3 แผนภาพแสดงตัวอย่างบริการแจ้งราคาหุ้น โดยใช้แนวคิดของ JAX-RPC	24
3.2 ยูสเคสไดอะแกรม	30
3.3 คลาสไดอะแกรม	33
3.4 ซีควেনซ์ไดอะแกรมของ Search Supplier Web Service	34
3.5 ซีควেনซ์ไดอะแกรมของ Customer Register	35
3.6 ซีควেনซ์ไดอะแกรมของ Validate User	36
3.7 ซีควেনซ์ไดอะแกรมของ Request Material List	37
3.8 ซีควেনซ์ไดอะแกรมของ Purchase Material	38
3.9 ซีควেনซ์ไดอะแกรมของ Purchase Tracking	39
5.1 หน้าจอขึ้นทะเบียนลูกค้าใหม่	47
5.2 หน้าจอแสดงรายการวัตถุดิบ	47
5.3 หน้าจอการสั่งซื้อวัตถุดิบ	48
5.4 หน้าจอกรอกข้อมูลในการจัดส่งสินค้า	49
5.5 หน้าจอยืนยันการสั่งซื้อวัตถุดิบ	49
5.6 หน้าจอแสดงรายการสั่งซื้อวัตถุดิบหลังยืนยัน	49
5.7 หน้าจอสอบถามการสั่งซื้อวัตถุดิบ	50

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

เนื่องจากโรงงานผลิตชิ้นส่วนต่าง ๆ จากวัตถุดิบพลาสติก ต้องการสั่งซื้อวัตถุดิบจากซัพพลายเออร์ต่าง ๆ เพื่อให้ได้วัตถุดิบตรงตามความต้องการมากที่สุด ในระบบเดิมเมื่อมีรายการสั่งซื้อพนักงานจัดซื้อต้องเข้าไปค้นหาหลาย ๆ เว็บไซต์ เพื่อให้ได้ข้อมูลวัตถุดิบและนำมาพิจารณาเลือกรายการที่ต้องการสั่งซื้อให้มีคุณสมบัติตามที่ต้องการ เช่น ราคาถูกที่สุด จึงเป็นการสิ้นเปลืองเวลาสำหรับการค้นหา ฉะนั้นจึงนำแนวคิดของเว็บเซอร์วิสมาใช้ในการพัฒนาให้บริการค้นหารายการสั่งซื้อที่ต้องการ และสั่งซื้อสินค้าได้ผ่านทางเว็บไซต์เดียว ก็สามารถเปรียบเทียบกับรายการของซัพพลายเออร์ต่าง ๆ แล้วตัดสินใจสั่งซื้อได้เลย

เว็บเซอร์วิสช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอปพลิเคชันที่ต่างกันเป็นไปได้โดยง่าย และสามารถถูกเรียกใช้ภายในองค์กรเอง หรือจากภายนอกองค์กร โดยผ่านไฟร์วอลล์ ดังนั้น จึงมีองค์กรขนาดใหญ่มากมาย กำลังพยายามพัฒนาระบบที่มีอยู่ของตนให้เข้ากับเว็บเซอร์วิส ซึ่งนับเป็นการลงทุนที่คุ้มค่า เนื่องจากเว็บเซอร์วิสสามารถเพิ่มศักยภาพในการทำงานขององค์กร อีกทั้งลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กร นอกจากนี้ ยังสามารถใช้ร่วมกับเว็บแอปพลิเคชัน โดยส่งผ่านข้อมูลทางอินเทอร์เน็ต ซึ่งนับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้า หรือหุ้นส่วนทางธุรกิจ

1.2 ที่มาของปัญหา

1. การค้นหาเอกสารต่าง ๆ ทำให้ใช้เวลานาน ทำให้เสียเปรียบทางธุรกิจ
2. ไม่ทราบสถานะของเอกสารสั่งซื้อ
3. การค้นหาข้อมูลซัพพลายเออร์ต้องเข้าไปค้นหาจากหลาย ๆ เว็บไซต์
ดังนั้นจึงเกิดแนวคิดที่จะพัฒนาระบบสารสนเทศสั่งซื้อโดยใช้เว็บเซอร์วิส เพื่อเพิ่มประสิทธิภาพในการทำงานและสะดวกต่อการค้นหาข้อมูลสินค้าที่ต้องการสั่งซื้อ

1.3 วัตถุประสงค์ของโครงการ

1. เพื่อค้นหาข้อมูลรายการสินค้า และวัตถุดิบที่พนักงานต้องการสั่งซื้อจากซัพพลายเออร์ต่าง ๆ และสามารถเปรียบเทียบราคาของสินค้าได้ เพื่อหาราคาสินค้าที่ถูกที่สุด และสามารถตัดสินใจสั่งซื้อได้เลย
2. เพื่ออำนวยความสะดวกในการค้นหาข้อมูลวัตถุดิบ และซัพพลายเออร์ให้ฝ่ายจัดซื้อ
3. เพื่อการตัดสินใจสั่งซื้อวัตถุดิบตรงตามความต้องการมากที่สุด
4. สามารถบอกสถานะของการสั่งซื้อ ตามความต้องการของผู้ใช้ได้ทันที
5. ลดเวลาในการจัดทำเอกสารสั่งซื้อ
6. เป็นตัวแทนในการสั่งซื้อสินค้าและวัตถุดิบ โดยติดต่อกับเว็บเซอร์วิสของซัพพลายเออร์ที่ต้องการ

1.4 ขอบเขตของระบบ

1. สามารถค้นหาข้อมูลสินค้าต่าง ๆ จากซัพพลายเออร์ในธุรกิจที่เกี่ยวข้องได้ และขอรายการราคาสินค้า จากซัพพลายเออร์ที่สนใจได้ โดยใช้เว็บเซอร์วิส
2. เมื่อได้เอกสารข้อมูลรายการราคาวัตถุดิบจากซัพพลายเออร์ต่าง ๆ แล้ว สามารถเปรียบเทียบได้ว่าสินค้าของซัพพลายเออร์รายไหน เป็นไปตามความต้องการของผู้ที่ต้องการสั่งซื้อ
3. สามารถตัดสินใจสั่งซื้อสินค้าได้จากการเปรียบเทียบราคาสินค้าโดยใช้เว็บเซอร์วิส
4. สามารถลงทะเบียนและทำธุรกรรมต่าง ๆ ผ่านทางเว็บเซอร์วิสได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อช่วยอำนวยความสะดวกในการค้นหาข้อมูลวัตถุดิบ และซัพพลายเออร์ให้แก่ฝ่ายจัดซื้อ
2. ช่วยเปรียบเทียบราคาของวัตถุดิบตามความต้องการได้อย่างรวดเร็วและถูกต้อง
3. ช่วยในการตัดสินใจสั่งซื้อวัตถุดิบตรงตามความต้องการมากที่สุด
4. สามารถสอบถามสถานะของเอกสารและลดเวลาจัดทำเอกสารได้

1.6 รายละเอียดของแต่ละบท

1. บทที่ 1 อธิบายถึงความเป็นมาของปัญหา ปัญหาของการระบบเดิม วัตถุประสงค์และขอบเขตของการพัฒนาระบบ และอธิบายรายละเอียดของแต่ละบท

2. บทที่ 2 อธิบายถึงทฤษฎีและเทคโนโลยีที่เกี่ยวข้องกับการพัฒนาระบบ
3. บทที่ 3 การวิเคราะห์ระบบและออกแบบ โดยใช้ยูสเคสไดอะแกรม คลาสไดอะแกรม ซีเควนซ์ไดอะแกรม
4. บทที่ 4 การออกแบบฐานข้อมูล
5. บทที่ 5 การพัฒนาระบบ
6. บทที่ 6 บทสรุป



บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 เว็บเซอร์วิส

2.1.1 ความหมายของเว็บเซอร์วิส

เว็บเซอร์วิสเป็นตัวกลางในการติดต่อระหว่างแอปพลิเคชัน ด้วยการตอบสนองคำร้องขอที่ส่งต่อมาจากระบบอื่น ทั้งจากระบบภายใน (Local) หรือภายนอก (ผ่านทางอินเทอร์เน็ต) ด้วยเทคโนโลยีสื่อสารที่ใช้ง่าย และไม่ยึดติดกับผู้ขายรายใดรายหนึ่ง คุณสมบัติของเว็บเซอร์วิสมีรายละเอียดดังนี้ [Sun Microsystem. 2002]

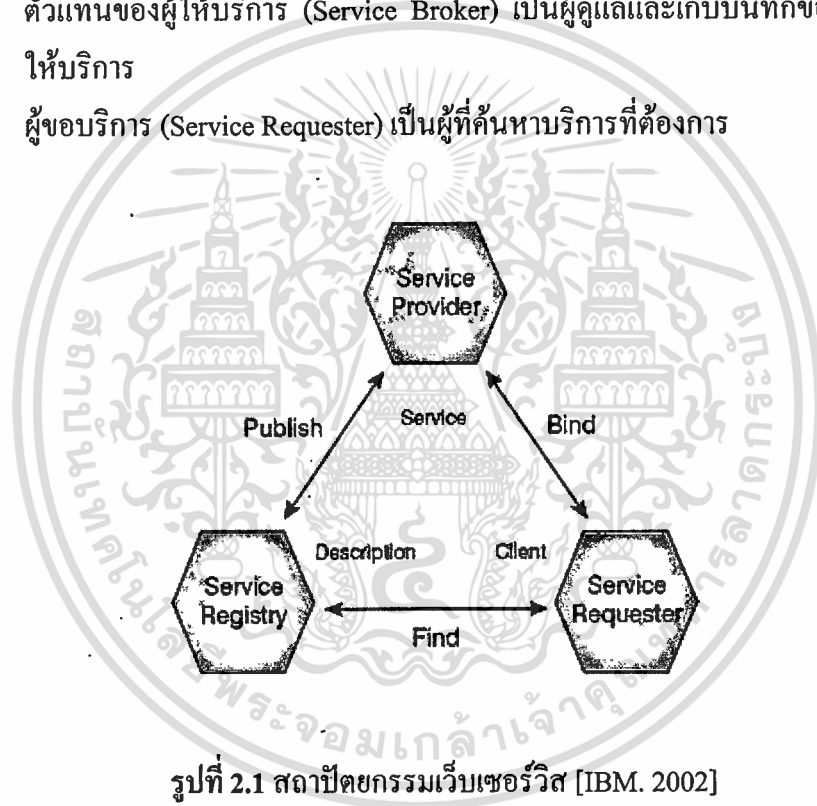
1. รายละเอียดในการสร้างและพัฒนาเว็บเซอร์วิสจะถูกซ่อนไว้ (Encapsulated) เพื่อไม่ให้มองเห็นได้จากภายนอก ผู้เรียกใช้เว็บเซอร์วิสจะรู้จักเพียงอินเทอร์เน็ตเฟสที่ผู้ให้บริการประกาศเอาไว้เท่านั้น กล่าวคือ เว็บเซอร์วิสจะเป็นประตูกันระบบงานภายในกับผู้ใช้จากภายนอก (คล้ายกับ XML ที่แยกตัวข้อมูล ออกจากส่วนที่ให้ความหมายว่าข้อมูลนั้นคืออะไร) ด้วยกรรมวิธีทางออบเจกต์ (Object-Oriented) อย่างเช่น การใช้ภาษา C++ ภาษาจาวา และคอมโพเนนต์ COM ดังนั้น จึงง่ายต่อการเรียกใช้งานเป็นอย่างยิ่ง
2. ซอฟต์แวร์ที่ทำงานบนระบบเว็บเซอร์วิส สามารถนำมาแก้ไขรายละเอียดภายในได้ โดยไม่ส่งผลกระทบต่อออกไปเป็นลูกโซ่ ทำให้การออกแบบซอฟต์แวร์เป็นไปโดยง่าย และผู้ใช้ที่ปลายทาง ก็ไม่จำเป็นต้องโหลดซอฟต์แวร์ติดตั้งไว้มากเกินไปจนเกิดความจำเป็น
3. โปรแกรมที่เรียกใช้เว็บเซอร์วิสจะรับรู้ได้เองว่าเซอร์วิสที่กำลังจะเรียกใช้นั้นมีการกำหนดพารามิเตอร์อินพุตและเอาต์พุตอย่างไร
4. ความเป็นโพรโทคอลมาตรฐาน นับเป็นคุณสมบัติสำคัญที่สุดของเว็บเซอร์วิส เนื่องจากมีพื้นฐานอยู่บนภาษา XML และ HTTP ซึ่งนักพัฒนาซอฟต์แวร์ทั่วโลกให้การยอมรับ โดยไม่ว่าจะเป็น SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) และ UDDI (Universal Description, Discovery, and Integration) ล้วนแต่มีรากฐานมาจากภาษา XML ทั้งสิ้น
5. เว็บเซอร์วิสสนับสนุนการค้นหาและเรียกใช้แบบไดนามิก (Dynamic Discovery and Invocation) ด้วยเทคโนโลยี UDDI แอปพลิเคชันจึงค้นหาและเรียกใช้เว็บเซอร์วิสได้ในขณะรันไทม์

ซึ่งเพิ่มความยืดหยุ่นให้กับนักพัฒนาซอฟต์แวร์ เนื่องจากไม่จำเป็นต้องกำหนดการเรียกใช้เว็บเซอร์วิสไว้ก่อน

2.1.2 สถาปัตยกรรมของเว็บเซอร์วิส

แนวความคิดของเว็บเซอร์วิสอาศัยหลักการของสถาปัตยกรรมการบริการในลักษณะ Service-Oriented Architecture (SOA) ดังรูปที่ 2.1 ซึ่งประกอบด้วย 3 ส่วนดังนี้ [IBM. 2002]

1. ผู้ให้บริการ (Service Provider) เป็นผู้สร้างเว็บเซอร์วิส
2. ตัวแทนของผู้ให้บริการ (Service Broker) เป็นผู้ดูแลและเก็บบันทึกข้อมูลเกี่ยวกับการให้บริการ
3. ผู้ขอบริการ (Service Requester) เป็นผู้ที่ค้นหาบริการที่ต้องการ



รูปที่ 2.1 สถาปัตยกรรมเว็บเซอร์วิส [IBM. 2002]

จากรูปที่ 2.1 ส่วนประกอบหลักทั้ง 3 ส่วนนี้สามารถติดต่อถึงกันโดยใช้ฟังก์ชันพื้นฐาน คือ การประกาศ (Publish) การค้นหา (Find) และการเรียกใช้ (Bind) ซึ่งฟังก์ชันการทำงานทั้ง 3 มีการทำงานดังนี้ ผู้ให้บริการจะทำการประกาศบริการที่ตนเองให้บริการไปยังตัวแทนของผู้ให้บริการ ซึ่งตัวแทนผู้ให้บริการจะบันทึกเก็บข้อมูลนั้นไว้ และคอยให้ผู้ขอบริการมาทำการค้นหาบริการที่ต้องการ และเมื่อพบบริการที่ต้องการ ผู้ให้บริการและผู้ขอบริการจะทำการติดต่อกัน โดยผู้ขอบริการทำการเรียกใช้บริการไปยังผู้ให้บริการนั้น

2.2 เครื่องมือที่ใช้พัฒนาเว็บเซอร์วิส

2.2.1 เทคโนโลยีหลักของเว็บเซอร์วิส [2]

1. XML (Extensible Markup Language) เป็นภาษาเมคอัพที่ถูกกำหนดมาตรฐานโดย World Wide Web Consortium (W3C) ซึ่งอยู่ในรูปข้อความ (text-based) ที่เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตในปัจจุบัน เนื่องจากเป็นภาษาเมคอัพ จึงมีลักษณะคล้ายกับภาษา HTML แต่ XML มีคุณสมบัติที่แตกต่างจาก HTML คือเป็นภาษาที่สามารถขยายได้ (Extensible) ผู้ใช้สามารถกำหนดแท็ก (tag) ได้เอง เพื่อใช้อธิบายสิ่งที่อยู่ในเอกสาร XML นั้น นอกจากนี้เทคโนโลยีเว็บเซอร์วิสได้ใช้ไวยากรณ์ของ XML เป็นพื้นฐานด้วย
2. SOAP (Simple Object Access Protocol) เป็นโพรโทคอลสำหรับสื่อสารระหว่างเว็บเซอร์วิสและแอปพลิเคชัน โดยระบบจะใช้ SOAP เพื่อให้แอปพลิเคชันและเว็บเซอร์วิสทำงานแลกเปลี่ยนข้อมูลได้ ซึ่งเป็น XML-based Protocol และ Messaging Protocol สำหรับแลกเปลี่ยนข้อมูลในสถานะแวดล้อมแบบกระจาย (Distributed Environment) จุดเด่นของ SOAP คือ เป็นโพรโทคอลที่เป็นกลาง สามารถทำงานกับโพรโทคอลอื่นหลายชนิด การพัฒนาก่อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการตามแบบจำลองทางวัตถุ และภาษาโปรแกรมที่ใช้พัฒนา
3. WSDL (Web Services Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของเว็บเซอร์วิสและวิธีการติดต่อขอรับบริการจากเว็บเซอร์วิส WSDL สร้างขึ้นโดยใช้ภาษา XML เวอร์ชันที่ใช้งานอยู่ปัจจุบันคือ WSDL 1.1
4. UDDI (Universal Description, Discovery, and Integration) เป็นวิธีการมาตรฐานสำหรับจัดเก็บและรวบรวมบริการต่าง ๆ ที่ให้บริการในรูปแบบของใคร่ครวญของบริการ หรือกล่าวได้ว่าเป็นมาตรฐานที่ให้ชุดพื้นฐาน API ของ SOAP สามารถนำมาใช้ในการพัฒนา เพื่อเป็นตัวแทนของผู้ให้บริการ โดย UDDI อาจเปรียบเสมือนฐานข้อมูลที่เกี่ยวข้องรายละเอียดของเว็บเซอร์วิสไว้ และรอให้ผู้ใช้บริการมาค้นหาบริการ บทบาทนี้เรียกว่า Service Discovery ส่วนในกรณีของผู้ให้บริการก็ต้องนำข้อมูลเกี่ยวกับเว็บเซอร์วิสของตนไปเก็บไว้ใน UDDI บทบาทนี้เรียกว่า Service Publication

2.2.2 เครื่องมือในการพัฒนาเว็บเซอร์วิส

Sun ONE (Sun Open Net Environment) เป็นโครงสร้างแบบเปิดที่สนับสนุนเว็บเซอร์วิสทำงานบนแพลตฟอร์มของ J2EE (Java 2 Platform Enterprise Edition) ซึ่ง Sun ONE สามารถสร้างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเซอร์วิสโดยใช้ XML, Servlets, JSP, EJB (Enterprise Java Beans) และเทคโนโลยีของจาวา ได้แก่ Java API for XML (JAX)

2.3 สถาปัตยกรรมและเอพีไอของจาวาสำหรับเอ็กซ์เอ็มแอล

Java APIs สำหรับ XML ใช้กับเว็บแอปพลิเคชันที่เขียนด้วยภาษาจาวา API เหล่านี้แบ่งเป็น 2 ประเภทคือ ที่เกี่ยวกับเอกสาร XML และที่เกี่ยวกับกระบวนการงาน (procedure) โดยมีรายละเอียดดังนี้ [Sun Microsystem. 2002]

- Document-Oriented

1. Java API for XML Processing (JAXP) : ประมวลผลเอกสาร XML โดยใช้พาร์เซอร์ (parser) ได้หลายชนิด
2. Java Architecture for XML Binding (JAXB) : เชื่อมโยงเอเลเมนต์ของ XML ไปสู่คลาสในภาษาจาวา

- Procedure-oriented

1. Java API for XML Messaging (JAXM) : ส่งข้อความ SOAP ด้วยวิธีมาตรฐานผ่านทางอินเทอร์เน็ต
2. Java API for XML Registries (JAXR) : จัดให้มีมาตรฐานเข้าถึงการจดทะเบียนทางธุรกิจและใช้สารสนเทศร่วมกัน
3. Java API for XML-based RPC (JAX-RPC) : ส่งเมธอด SOAP เรียกใช้และรับปาร์ตี้ระยะไกล (remote parties) บนอินเทอร์เน็ต

ลักษณะสำคัญที่สุดของ Java APIs สำหรับ XML คือ ได้สนับสนุนมาตรฐานทางอุตสาหกรรม ดังเช่น เครื่องมือหลากหลายชนิดที่ปฏิบัติงานเป็นกลุ่มที่มีมาตรฐานร่วมกัน เช่น เวิลด์ไวด์เว็บ (W3C) และ the Organization for the Advancement of Structured Information Standards (OASIS) ได้กำหนดมาตรฐานของการทำสิ่งต่าง ๆ ทางธุรกิจที่ทำให้ข้อมูลทางธุรกิจและแอปพลิเคชันทำงานร่วมกันได้

ลักษณะอีกอย่างหนึ่งของ Java APIs สำหรับ XML คือ มีความยืดหยุ่นในการใช้งาน เช่น โค้ดของ JAXP สามารถใช้เครื่องมือได้หลายชนิดในการประมวลผลเอกสาร XML และโค้ดของ JAXM สามารถใช้โพรโทคอลส่งข้อความได้หลายชนิดบน SOAP

2.3.1 JAXP [Sun Microsystem. 2002]

Java API for XML Processing (JAXP) ทำให้การประมวลผลข้อมูล XML ด้วยแอปพลิเคชันที่เขียนด้วยภาษาจาวาง่ายขึ้น JAXP ได้นำเอา SAX (Simple API for XML Parsing) และ DOM (Document Object Model) ที่เป็นมาตรฐานของพาร์เซอร์ให้เลือก parse ข้อมูลตามอีเวนต์ หรือ สร้างตัวแทนของออบเจกต์ ใน JAXP เวอร์ชัน 1.1 ได้สนับสนุนมาตรฐานของ XSLT (XML Stylesheet Language Transformations) ให้ควบคุมการนำเสนอข้อมูลและแปลงข้อมูลไปเป็นเอกสาร XML ตัวอื่น หรือรูปแบบอื่นได้ดังเช่น HTML นอกจากนี้ JAXP จัดให้มี namespace ที่ใช้ทำงานกับ DTD ซึ่งอาจจะมีการตั้งชื่อซ้ำซ้อนกันได้ ตัวอย่างของ JAXP ได้แก่

1. SAX API

SAX เป็น API ที่ทำงานโดยใช้อีเวนต์เป็นหลัก เมื่อพาร์เซอร์อ่านข้อมูลเอกสาร XML ตั้งแต่เริ่มต้นจนถึงส่วนท้ายของเอกสาร ในแต่ละครั้งที่อ่านได้จดจำโครงสร้างไวยากรณ์ของเอกสาร XML และแจ้งแอปพลิเคชันว่ากำลังดำเนินการอยู่ โดยเรียกเมธอดจากอินเทอร์เฟซ ContentHandler ดังตัวอย่าง

```
<priceList> [parser calls startElement]
  <coffee> [parser calls startElement]
    <name>Mocha Java</name> [parser calls startElement, characters, and endElement]
    <price>11.95</price> [parser calls startElement, characters, and endElement]
  </coffee> [parser calls endElement]
</priceList>
```

เมื่อพาร์เซอร์มาถึงเครื่องหมาย “<” จะเรียกเมธอด startElement เมื่อมาถึงข้อมูล (character data) จะเรียกเมธอด characters เมื่อมาถึงเครื่องหมาย “</” จะเรียกเมธอด endElement และทำอย่างนี้ต่อไปเรื่อย ๆ

การอิมพลีเมนต์หรือการทำให้เกิดผล (implementation) ประการแรกจะต้องสร้างออบเจกต์ SAXParser จากออบเจกต์ SAXParserFactory แล้วเรียกใช้เมธอด parse ส่งไฟล์ priceList.xml และอินสแตนซ์ของคลาส handler (เป็นการอิมพลีเมนต์ของเมธอด startElement และ characters)

```
SAXParserFactory factory = SAXParserFactory.newInstance();
```

```
SAXParser saxParser = factory.newSAXParser();
```

```
saxParser.parse("priceList.xml", handler);
```

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SAX Parser จะเข้าไปอ่านไฟล์ priceList.xml ทีละบรรทัด เรียกใช้เมธอดตามความเหมาะสม เมื่อมีเมธอดที่อ้างถึงแล้วพาร์เซอร์จะเรียกใช้เมธอดอื่นเช่น startDocument, endDocument, ignorableWhiteSpace และ processingInstructions

ตามที่ได้แสดงวิธีการอิมพลีเมนต์เมธอด characters และ startElement ทำให้ค้นหาราคาของ Mocha Java และพิมพ์ออกมาได้ ด้วยวิธีที่ SAX parser ทำงานใน 2 เมธอดนี้ร่วมกันค้นหาเอลิเมนต์ name ที่มี characters เป็น Mocha Java และเอลิเมนต์ price เมธอดเหล่านี้ใช้ตัวชี้บ่ง (flag) เก็บสถานะของเงื่อนไขที่พบดังนี้

```
public void startElement(..., String elementName, ...){
    if(elementName.equals("name")){
        inName = true;
    } else if(elementName.equals("price") && inMochaJava){
        inPrice = true;
        inName = false;
    }
}

public void characters(char [] buf, int offset, int len) {
    String s = new String(buf, offset, len);
    if(inName && s.equals("Mocha Java")) {
        inMochaJava = true;
        inName = false
    } else if (inPrice) {
        System.out.println("The price of Mocha Java is: " + s);
        inMochaJava = false;
        inPrice = false;
    }
}
```

SAX Parser ได้ตรวจสอบข้อมูลตามข้อกำหนดที่กำหนดไว้ใน DTD ของเอกสาร XML โดยสร้างจากออบเจกต์ SAXParserFactory ที่มีการเซตเป็น validate ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าคุณคิดอย่างไร สิ่งนี้ช่วยเพิ่มขีดความสามารถ และช่วยให้คุณเข้าถึงข้อมูลเอกสารที่ซ่อนอยู่ที่มีประโยชน์

```
factory.setValidating(true);
```

ดังนั้นพาร์เซอร์จะรู้ว่าใช้ DTD ตัวไหนตรวจสอบเอกสาร XML โดยการประกาศที่ DOCTYPE ดังนี้

```
<!DOCTYPE PriceList SYSTEM "priceList.DTD">
```

2. DOM API

DOM API (Document Object Model API) ถูกกำหนดโดยกลุ่มทำงานของ W3C เป็นชุดของอินเทอร์เฟซสำหรับการสร้างตัวแทนของออบเจกต์ในรูปแบบต้นไม้ สามารถจัดการกับเอกสาร XML ด้วยเมธอดของ DOM เช่น insert และ remove ในรูปโครงสร้างข้อมูลแบบต้นไม้ ต่างกับ SAX Parser ตรงที่ DOM Parser ยอมให้เข้าถึงข้อมูลแบบสุ่มหาข้อมูลบางส่วนในเอกสาร XML ได้ ความแตกต่างอีกอย่างหนึ่งคือ SAX Parser ใช้สำหรับอ่านเอกสาร XML เท่านั้น แต่ DOM Parser สามารถสร้างตัวแทนออบเจกต์ของเอกสาร XML และจัดการเอกสารในหน่วยความจำ โดยเพิ่มเอลิเมนต์ใหม่ หรือลบเอลิเมนต์ที่มีอยู่ออกไปได้

ในตัวอย่างที่แล้ว เราได้ใช้ SAX Parser ค้นหาส่วนหนึ่งของข้อมูลในเอกสาร การใช้ DOM Parser จะต้องมีการนำเอา DOM ทั้งหมดไปไว้ในหน่วยความจำ ซึ่งมีประสิทธิภาพต่ำสำหรับการค้นหาข้อมูลแค่ 2-3 รายการ โดยเฉพาะในเอกสารที่มีขนาดใหญ่ ในตัวอย่างถัดไป เป็นการเพิ่ม new coffee ลงใน price list โดยใช้ DOM Parser ไม่สามารถใช้ SAX Parser สำหรับแก้ไข price list ได้ เพราะอ่านข้อมูลได้อย่างเดียว สมมุติว่าต้องการเพิ่ม Kona coffee ใน price list เราจะอ่านไฟล์ price list เข้าไปใน DOM และ insert เอลิเมนต์ new coffee ด้วย name และ price ดังตัวอย่างข้างล่าง โค้ดนี้สร้างออบเจกต์ DocumentBuilderFactory ขึ้นมา ซึ่งใช้สร้าง builder ที่เป็นออบเจกต์ของ DocumentBuilder โดยเรียกใช้เมธอด parse และส่งไฟล์ priceList.xml เข้าไป

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
```

```
DocumentBuilder builder = factory.newDocumentBuilder();
```

```
Document document = builder.parse("priceList.xml");
```

จากตัวอย่างนี้ document เป็นตัวแทน DOM ของ price list ที่อยู่ในหน่วยความจำ ส่วนของโค้ดที่เพิ่ม new coffee ในเอกสาร price list (ด้วย name "Kona" และ price 13.50) อยู่หน้า coffee ที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ชื่อ “Mocha Java” ขึ้นแรกให้ get รายการเอลิเมนต์ name และเข้าไปค้นหา “Mocha Java” การใช้งาน อินเทอร์เฟซ Node ที่รวมอยู่ในแพ็คเกจ org.w3c.dom สร้างออบเจกต์ Node ให้กับเอลิเมนต์ new coffee และ โหนดใหม่สำหรับเอลิเมนต์ name และ price ดังนั้นสร้างออบเจกต์ TextNode (TextNode) ให้กับเอลิเมนต์เหล่านี้ ดังตัวอย่าง

```

NodeList list = document.getElementsByTagName("name");
Node thisNode = list.item(0);
// loop through list
Node thisChild = thisNode.getChildNode();
if(thisNode.getFirstChild() instanceof org.w3c.dom.TextNode) {
    String data = thisNode.getFirstChild().getData();
}
if (data.equals("Mocha Java")) {
    // โหนดใหม่จะถูกริ insert ก่อน Mocha Java
    Node newNode = document.createElement("coffee");
    Node nameNode = document.createElement("name");
    TextNode textNode = document.createTextNode("Kona");
    nameNode.appendChild(textNode);

    Node priceNode = document.createElement("price");
    TextNode tpNode = document.createTextNode("13.50");
    priceNode.appendChild(tpNode);

    newNode.appendChild(nameNode);
    newNode.appendChild(priceNode);
    thisNode.insertBefore(newNode, thisNode);
}

```

DOM Parser ตรวจสอบความสมเหตุสมผล (validate) วิธีเดียวกันกับ SAX Parser โดย

เรียกใช้ setValidating(true) สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไปว่ากรณีโดยหนังสือ อีกข้างหนึ่งเปิดให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. XSLT API

XSLT (XSL Transformations) ถูกกำหนดโดยกลุ่มทำงานของ W3C เป็นภาษาที่กำหนดรายละเอียดการแปลงเอกสารต่าง ๆ ของ XML ไปเป็นเอกสารอื่น เช่น เอกสาร HTML ซึ่งการแปลงรูปแบบนี้จะต้องมีสไตลชีตที่เขียนด้วยภาษา XSL (XML Stylesheet Language) ในการแสดงข้อมูล XML

JAXP สนับสนุน XSLT ด้วยแพ็คเกจ `javax.xml.transform` ซึ่งยอมให้ใช้ XSLT ในการแปลงรูปแบบเอกสารได้ ตามตัวอย่างต่อไปนี้ แสดงวิธีสร้างเอกสาร XML จาก DOM tree

จากตัวอย่างโค้ดข้างล่างได้สร้างออบเจกต์ `Transformer (transformer)` เพื่อแปลงรูปแบบดังนี้

```
TransformerFactory transFactory = TransformerFactory.newInstance();
```

```
Transformer transformer = transFactory.newTransformer();
```

ใช้รูทโหนดของ DOM tree สร้างออบเจกต์ `DOMSource (source)` ของการแปลงรูปแบบดังนี้

```
DomSource source = new DOMSource(document);
```

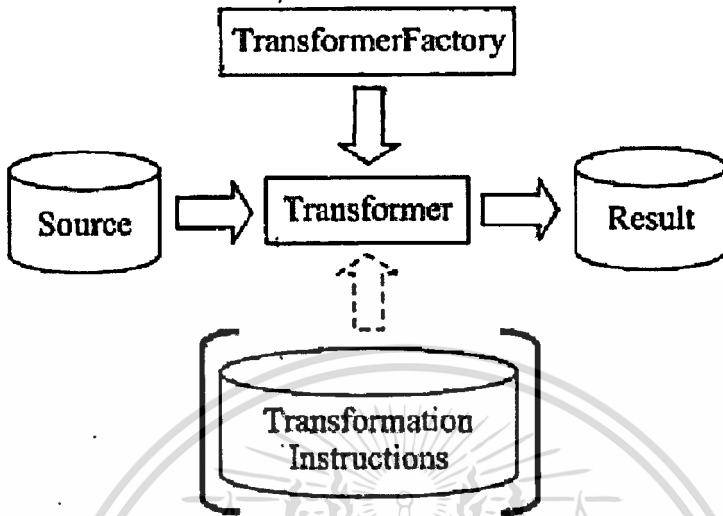
ตามโค้ดข้างล่างสร้างออบเจกต์ `StreamResult (result)` เพื่อเก็บผลลัพธ์ที่ได้ และแปลงทรี (`source`) ไปเป็นเอกสาร XML เก็บไว้ใน `result` ดังรูปที่ 2.2 [Myerson Judith M. 2001]

```
File newXML = new File("newXM.xml");
```

```
FileOutputStream os = new FileOutputStream(newXML);
```

```
StreamResult result = new StreamResult(os);
```

```
transformer.transform(source, result);
```



รูปที่ 2.2 แผนภาพการแปลงของ XSLT

2.3.2 JAXB

JAXB จัดหาวิธีที่สะดวกรวดเร็วในการเชื่อมโยงระหว่างเอกสาร XML กับจาวาออบเจกต์ โดยมีคอมไพเลอร์ของ JAXB ได้สร้างชุดของจาวาคลาสที่ประกอบด้วยโค้ดต่าง ๆ ที่ parse เอกสาร XML ตามที่กำหนดในเค้าร่าง (schema) ผู้พัฒนาที่ใช้คลาสต่าง ๆ ที่สร้างขึ้นมา สามารถสร้างจาวาออบเจกต์ทรี (Java Object Tree) แทนเอกสาร XML ได้โดยใช้โครงสร้างของต้นไม้และสร้างเอกสาร XML กลับมาจากทรีได้

เมื่อเริ่มใช้งานแอปพลิเคชัน JAXB จำเป็นต้องมีเค้าร่าง (schema) ซึ่งเวอร์ชันปัจจุบันของ JAXB เป็น DTD อาจจะต้องสร้าง DTD ขึ้นมาเองหรือเรียกใช้ผ่านทาง JAXR ก็ได้ การผูก DTD ไว้กับชุดของคลาสมีขั้นตอนดังนี้ [Sun Microsystem. 2002]

1. เขียนเค้าร่างยึดเหนี่ยว (Binding Schema) ขึ้นมา ภายในมีคำสั่งวิธีการผูกเค้าร่าง (schema) กับคลาสเหล่านั้น เค้าร่างยึดเหนี่ยวถูกเขียนด้วยภาษา XML-based binding language ซึ่งได้รวบรวมอยู่ใน JAXB

2. ทำการรันคอมไพเลอร์ของเค้าร่าง (schema) ซึ่งจะนำ DTD กับเค้าร่างยึดเหนี่ยว และสร้างคลาสต่าง ๆ ขึ้นมา ในแต่ละคลาส คอมไพเลอร์จะสร้างเมธอด get และ set

เมื่ออินสแตนซ์ของคลาสถูกสร้างขึ้นและถูกกำหนดด้วยข้อมูล สามารถใช้เมธอด accessor เข้าถึงข้อมูลได้ (ชุดของเมธอด accessor เรียกว่า property)

การสร้างคลาสจาก DTD

ตัวอย่างของการสร้างคลาสจาก DTD พิจารณา DTD ต่อไปนี้ (อยู่ในไฟล์ priceList.dtd)

```
<!ELEMENT priceList (coffee)+ >
<!ELEMENT coffee (name, price) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT price (#PCDATA) >
```

คอมไพเลอร์ของ JAXB schema จะทำการแปลง DTD ไปเป็นเค้าร่างชนิดเหนียว (binding schema) โดยกำหนดเพียงเอลิเมนต์รากของเอกสาร ดังนี้

```
...
<element name="priceList" type="class" class="PriceList" root="true" />
<element name="price" type="value" convert="BigDecimal" />
<conversion name="BigDecimal" type="java.math.BigDecimal" />
...
```

จาก DTD และเค้าร่างชนิดเหนียวนี้ คอมไพเลอร์ของเค้าร่าง (schema) ได้สร้างคลาส PriceList และ Coffee ขึ้นมา คลาส PriceList ได้รวบรวม constructor และ List property ที่มีเอลิเมนต์ coffee ผูกติดอยู่ คลาส Coffee ประกอบด้วย constructor และ property แทนชื่อของ coffee และ property แทนราคา เมล็ดต่าง ๆ ของ price มีดังนี้

```
BigDecimal getPrice();
```

```
Void setPrice(BigDecimal x);
```

ทั้ง PriceList และ Coffee ประกอบไปด้วยเมล็ดดังต่อไปนี้

- Unmarshalling : เป็นโพรเซสของการสร้างตัวแทนออบเจกต์ของข้อมูล XML
- Validation : เป็นโพรเซสของการตรวจสอบออบเจกต์ต่างๆ ให้ตรงกับข้อกำหนดของ DTD
- Marshalling : เป็นโพรเซสของการสร้างข้อมูล XML จากตัวแทนของออบเจกต์

การสร้างตัวแทนออบเจกต์ของข้อมูล XML

หลังจากสร้างคลาสขึ้นมาแล้ว เราสามารถเขียนจาวาแอปพลิเคชัน โดยใช้คลาสเหล่านั้นและสร้างตัวแทนออบเจกต์ของเอกสาร XML ให้ถูกต้องตาม DTD ออบเจกต์แต่ละตัวที่มีลักษณะเดียวกับเอลิเมนต์ในเอกสาร XML เป็นอินสแตนซ์ ของคลาส ๆ หนึ่งจากชุดของคลาสที่ถูกสร้างขึ้นมาดังกล่าว เนื่องจากออบเจกต์ที่เชื่อมโยงทั้งเอกสารและคลาสต่าง ๆ จะมีวิธีที่แตกต่างกันทั้ง 2 วิธีในการสร้างจาวาออบเจกต์ทรีโดย unmarshalling เอกสาร XML ให้ถูกต้อง หรืออินสแตนซ์ (สร้างตัวแทน) ออบเจกต์จากคลาส ด้วยวิธีนี้เอง JAXB ได้อนุญาตให้ประมวลผลได้ทั้งเอกสาร XML ที่มีอยู่แล้วและสร้างข้อมูล XML ใหม่โดยการอินสแตนซ์คลาสที่ถูกสร้างขึ้นมา สมมุติให้มีเอกสาร XML ดังต่อไปนี้

```
<priceList>
  <coffee>
    <name>Arabica</name>
    <price>13.50</price>
  </coffee>
  <coffee>
    <name>Mocha Java</name>
    <price>11.95</price>
  </coffee>
</priceList>
```

เราสามารถทำ unmarshal กับเอกสาร XML นี้โดยสร้าง input stream จากเอกสาร XML นี้ และอ้างถึงเมธอด unmarshal ของคลาส PriceList ได้ดังนี้

```
FileInputStream fis = new FileInputStream("priceList.xml");
PriceList myPrices = PriceList.unmarshal(fis);
```

ขณะนี้เรามีจาวาออบเจกต์ทรีกับออบเจกต์ myPrices ที่เป็นรูทของทรี สมมุติว่าต้องการสร้างราคากาแฟในเอกสาร XML อันดับแรกต้องสร้างออบเจกต์ทรี โดยการอินสแตนซ์แล้ว marshal ทรีไปเป็นเอกสาร XML อันดับสอง สร้างออบเจกต์ทรีโดยการอินสแตนซ์ออบเจกต์ PriceList และ get

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการของออบเจ็กต์ Coffee จากออบเจ็กต์ทีรีนั้น และสร้างออบเจ็กต์ Coffee (zCoffee) ขึ้นมาเซตค่า Name และ Price แล้วเพิ่มเข้าไปใน Coffee ดังนี้

```
PriceList myNewPrices = new PriceList();
List listOfCoffees = myNewPrices.getCoffees();
Coffee zCoffee = new Coffee();
zCoffee.setName("Zapoteca");
zCoffee.setPrice("15.00");
listOfCoffees.add(zCoffee);
```

การเข้าถึงข้อมูลจากออบเจ็กต์ทีรี

สมมติว่าต้องการที่จะเปลี่ยนราคาของ Mocha Java ในออบเจ็กต์ทีรีตัวแรกที่สร้างขึ้นมา จะต้องทำการค้นหา Mocha Java ในรายการของ coffee และกำหนดราคาใหม่โดยอ้างถึงเมธอด setPrice ในออบเจ็กต์ Coffee ดังนี้

```
List coffees = myPrices.getCoffees();
For (ListIterator i = coffees.listIterator(); i.hasNext(); ) {
    Coffee myCoffee = (Coffee) i.next();
    If (myCoffee.getName().equals("Mocha Java")) {
        MyCoffee.setPrice("12.50");
    }
}
```

การเขียนเอกสาร XML จากออบเจ็กต์ทีรี

จากการที่ใช้ unmarshalling สร้างตัวแทนออบเจ็กต์จาก XML เราก็สามารถ marshal ออบเจ็กต์เหล่านั้นออกไปเป็นเอกสาร XML ได้นั่นคือ JAXB อนุญาตให้คุณสร้างเอกสาร XML ใหม่ที่ถูกต้องตามข้อกำหนดในเค้าร่าง (schema)

นำออบเจ็กต์ทีรีที่ถูกคัดแปลงในรูปของเอกสาร XML โดยสร้างไฟล์ XML และ output stream ขึ้นมา แล้วเรียกใช้เมธอด marshal กับออบเจ็กต์ myNewPrices ดังนี้

```
File newPrices = new File("newPriceList.xml");
```

```
FileOutputStream fos = new FileOutputStream(newPrices);
```

```
MyNewPrices.marshal(fos);
```

JAXB ได้จัดให้มีตัวเชื่อมโยงระหว่างเทคโนโลยี XML และ Java เข้าด้วยกัน เพียงแค่มีเอกสาร XML อยู่ในรูป

อินสแตนซ์ของ schema และ Java object อยู่ในรูปอินสแตนซ์ของคลาส ดังนั้นเราสามารถสร้าง Java objects ที่มี concept เดียวกันกับข้อมูล XML ด้วย JAXB ซึ่งทำให้การสร้างแอปพลิเคชันที่ประมวลผลข้อมูล XML ง่ายขึ้น

ความแตกต่างระหว่าง JAXP และ JAXB

เราจำเป็นต้องพิจารณาเมื่อเลือกใช้สถาปัตยกรรมหรือ API ที่ประมวลผลข้อมูล XML จากข้อดีของ JAXB และ JAXP เพื่อตัดสินใจได้ถูกต้องดังนี้ [Sun Microsystem, 2002]

ใช้ JAXB เมื่อคุณต้องการ :

- เข้าถึงข้อมูลในหน่วยความจำ แต่ไม่จำเป็นต้องใช้ความสามารถในการจัดการกับทรี
- ประมวลผลเฉพาะข้อมูลให้ถูกต้อง.
- แลกเปลี่ยนข้อมูลที่มีชนิดแตกต่างกัน
- สร้างคลาสต่าง ๆ ตามที่กำหนดใน DTD
- สร้างตัวแทนออบเจกต์ของข้อมูล XML

ใช้ JAXP เมื่อคุณต้องการ :

- มีความยืดหยุ่นกับการพิจารณาวิธีการเข้าถึงข้อมูลแบบเป็นลำดับด้วย SAX หรือแบบสุ่มในหน่วยความจำด้วย DOM
- ใช้โค้ดที่ประมวลผลเหมือนกันกับเอกสารที่มี DTD แตกต่างกัน
- ประยุกต์ใช้ XSLT ในการแปลงรูปแบบ
- แทรกหรือลบออบเจกต์ต่าง ๆ จากออบเจกต์ทรีที่เป็นตัวแทนข้อมูล XML

2.3.3 JAXM [Sun Microsystem, 2002]

Java API for XML Messaging (JAXM) ได้จัดหาวิธีการมาตรฐานในการส่งเอกสาร XML

บนอินเทอร์เน็ตจากแพลตฟอร์มที่เป็นจาวา โดยใช้ SOAP 1.1 และข้อกำหนดที่แนบมากับ SOAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติไปใช้ประโยชน์ด้านการค้า

และถูกขยายให้ทำงานกับโพรโทคอลสำหรับส่งข้อความในระดับสูง อย่างเช่น ebXML

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของธุรกิจที่ใช้บริการผู้ให้บริการข้อความ (messaging provider) ซึ่งทำงานอยู่เบื้องหลังต้องการที่จะขนส่งและหาเส้นทางของข้อความ เมื่อผู้ให้บริการข้อความถูกเรียกใช้งานข้อความทุกตัวของ JAXM ก็จะถูกเรียกใช้ด้วย โดยส่งข้อความแรกไปที่ผู้ให้บริการของผู้ส่ง และผู้ให้บริการของผู้รับ ถ้าไม่ถึงปลายทาง ผู้ให้บริการข้อความได้ใช้เทคโนโลยี JAXM บนไคลเอนต์โดยไม่ต้องรู้ว่าผู้ให้บริการทำงานอยู่เบื้องหลัง JAXM client ทำให้มีการเรียกใช้เมธอด JAXM และ messaging provider ร่วมกันกับโครงสร้างพื้นฐานของข้อความ

ข้อความ JAXM ประกอบไปด้วย 2 ส่วน คือส่วนของ SOAP กับ optional attachment

1. ส่วนของ SOAP ประกอบด้วยออบเจกต์ SOAPEnvelope ที่บรรจุออบเจกต์ SOAPHeader และ SOAPBody ซึ่ง SOAPBody ในเอกสาร XML อยู่ในรูปเนื้อหาข้อความ (message content)
2. ส่วนของ Attachment อาจจะมีจำนวนเท่าใดก็ได้ หรืออาจจะไม่มีก็ได้ แต่ถ้ามีก็ไม่ใช่เอกสาร XML และประกอบไปด้วยตัวอักษร หรือไฟล์รูปภาพ

การรับการเชื่อมต่อ

สิ่งแรกที่ JAXM client จำเป็นต้องทำคือรับการเชื่อมต่อ (connection) ทั้งออบเจกต์

SOAPConnection และ ProviderConnection

1. การรับการเชื่อมต่อแบบจุดต่อจุด (Point-to-Point)

เครื่องไคลเอนต์ ประเภท Standalone ถูกจำกัดการใช้ออบเจกต์ SOAPConnection ด้วยการติดต่อแบบจุดต่อจุด คือไปอย่างมีทิศทางจากผู้ส่งไปยังผู้รับ การติดต่อทั้งหมดของ JAXM ถูกสร้างโดย connection factory ในกรณีนี้ออบเจกต์ SOAPConnection ถูกสร้างจาก factory ที่เป็นออบเจกต์ของ SOAPConnectionFactory

```
SOAPConnectionFactory factory = SOAPConnectionFactory.newInstance();
```

```
SOAPConnection con = factory.createConnection();
```

2. การรับการเชื่อมต่อจากผู้ให้บริการข้อความ

การใช้ผู้ให้บริการข้อความ แอปพลิเคชันจะต้องมีออบเจกต์ ProviderConnection สร้างการติดต่อจากผู้ให้บริการข้อความมี 2 วิธีดังนี้

วิธีที่ 1 รับอินสแตนซ์ของ ProviderConnectionFactory ที่ใช้สร้างการติดต่อ ตัวแปร pcCon เป็นการอิมพลิเมนต์การติดต่อของ JAXM messaging provider

```
ProviderConnectionFactory pcFactory = ProviderConnectionFactory.newInstance();
```

```
ProviderConnection pcCon = pcFactory.createConnection();
```

วิธีที่ 2 สร้างออบเจ็กต์ *ProviderConnection* ให้ดึงข้อมูลจากออบเจ็กต์

ProviderConnectionFactory ที่ถูกอิมพลีเมนต์ เพื่อสร้างการติดต่อกับ messaging provider ตัวอย่างโค้ดข้างล่าง สาธิตการ get ออบเจ็กต์ *ProviderConnectionFactory* และใช้สร้างการติดต่อ ใน 2 บรรทัดแรกใช้ JNDI API ดึงข้อมูลออบเจ็กต์ *ProviderConnectionFactory* จาก naming service ที่จดทะเบียนด้วยชื่อ “*CoffeeBreakProvider*” เมื่อถูกส่งผ่านไปในรูปแบบอาร์กิวเมนต์ (argument) เมธอด *lookup* ได้ส่งค่าออบเจ็กต์ *ProviderConnectionFactory* กลับไปที่ชื่อนั้นอยู่

```
Context ctx = getInitialContext();
```

```
ProviderConnectionFactory pcFactory =
```

```
(ProviderConnectionFactory)ctx.lookup("CoffeeBreakProvider");
```

```
ProviderConnection con = pcFactory.createConnection();
```

อินสแตนซ์ของ *ProviderConnection (con)* เป็นการเชื่อมต่อของ *Coffee Break*

การสร้างข้อความ

การติดต่อกันจะมีข้อความที่ถูกสร้างโดย factory ออบเจ็กต์ *MessageFactory* สามารถนำไปใช้ได้อยู่ 2 วิธี

วิธีที่ 1. รับค่าอินสแตนซ์ของคลาส *MessageFactory* โดยที่อินสแตนซ์นี้ถูกใช้สร้างออบเจ็กต์ *SOAPMessage*

```
MessageFactory messageFactory = MessageFactory.newInstance();
```

```
SOAPMessage m = messageFactory.createMessage();
```

ออบเจ็กต์ *SOAPMessage* ทุกตัวที่ *messageFactory* สร้างขึ้นมารวมทั้ง *m* จากบรรทัดบน จะเป็นข้อความของ SOAP ซึ่งข้อความเหล่านี้จะไม่ได้กำหนดไว้ก่อนใน Header

ความยืดหยุ่นของ JAXM API อนุญาตให้ใช้ SOAP header ได้ ดังเช่น โพรโทคอล ebXML หรือ BizTalk ถูกสร้างอยู่ในส่วนบนสุดของข้อความ SOAP การใช้วิธีนี้ของ SOAP โดยกลุ่มมาตรฐานหรืออุตสาหกรรมถูกเรียกว่า *profile*

วิธีที่ 2 สร้างออบเจกต์ของ MessageFactory จากเมธอด createMessageFactory ของ ProviderConnection และกำหนด profile ส่วนออบเจกต์ SOAPMessage สร้างจากผลลัพธ์ของออบเจกต์ MessageFactory ซึ่งจะสนับสนุน profile ที่กำหนด ดังตัวอย่างข้างล่าง schemaURI เป็น URI ของ schema สำหรับ profile ที่ต้องการ และ m2 จะสนับสนุน profile ของข้อความ (messaging profile) ที่ถูกสร้างจาก createMessageFactory

```
MessageFactory messageFactory2 = con.createMessageFactory(<schemaURI>);
SOAPMessage m2 = messageFactory2.createMessage();
```

ทั้ง m และ m2 ได้บรรจุอยู่ในเอลิเมนต์ SOAPPart, SOAPEnvelope และ SOAPBody ส่วนใน SOAPHeader เป็นทางเลือกเพิ่มเติม

การส่งข้อความ

เครื่องไคลเอนต์ประเภท standalone ใช้เมธอด call ของ SOAPConnection ส่งข้อความ โดยอาร์กิวเมนต์ในเมธอด call ประกอบด้วยข้อความที่จะส่งและออบเจกต์ Endpoint ที่บรรจุ URL ของฝั่งรับ

```
SOAPMessage response = soapConnection.call(message, urlEndpoint);
```

แอปพลิเคชันที่ใช้เมธอด send ของ ProviderConnection ส่งข้อความทีละข้อความ ซึ่งเมธอดนี้ส่งข้อความแบบอะซิงโครนัส นั่นคือส่งข้อความและส่งค่ากลับ (return) ในทันที เมธอดนี้ใช้พารามิเตอร์แค่ตัวเดียวในการส่งข้อความ

```
providerConnection.send(message);
```

2.3.4 JAXR [Sun Microsystem. 2002]

Java API for XML Registries (JAXR) ได้จัดหาวิธีที่จะเข้าถึงมาตรฐานการจดทะเบียนทางธุรกิจบนอินเทอร์เน็ต การจดทะเบียนทางธุรกิจเปรียบเสมือนสมุดหน้าเหลืองที่บรรจุรายการต่าง ๆ ของธุรกิจและผลิตภัณฑ์หรือบริการของธุรกิจนั้น JAXR ได้ให้ผู้พัฒนาเขียนแอปพลิเคชันในภาษาจาวาด้วยวิธีการที่ไม่ซ้ำแบบในการจดทะเบียนทางธุรกิจ ที่อยู่บนมาตรฐานเปิด (ดังเช่น ebXML) หรือข้อกำหนดของ industry consortium-led (เช่น UDDI)

การจดทะเบียนเหล่านี้กลายเป็นองค์ประกอบที่สำคัญมากขึ้นของเว็บเซอร์วิส เพราะว่าการจดทะเบียนอนุญาตให้ธุรกิจต่าง ๆ ทำงานร่วมกันได้ อย่างไรก็ตาม ความจำเป็นของ JAXR ทำให้องค์กรเข้าถึงการจดทะเบียนที่เป็นมาตรฐานทางธุรกิจ จากภาษาจาวาที่กำลังก้าวหน้ามากขึ้น

การจดทะเบียนธุรกิจ

องค์กรที่ใช้แพลตฟอร์มจาวาสำหรับธุรกิจทางอิเล็กทรอนิกส์ จะใช้ JAXR รีจิสเตอร์ด้วยตัวเองในการจดทะเบียนที่เป็นมาตรฐาน โดยจะจัดหาชื่อและคำอธิบายด้วยตัวเอง และจัดแยกประเภท concepts ให้ง่ายต่อการค้นหา ดังตัวอย่างโค้ดข้างล่าง ในบรรทัดแรก สร้างออบเจกต์ RegistryService (rs) และใช้ rs สร้างออบเจกต์ BusinessLifeCycleManager (lcm) ธุรกิจกาแฟลูกโซ่ที่เรียกว่า The Coffee-Break ได้ถูกแสดงโดยออบเจกต์ Organization (org) ซึ่ง Coffee Break ได้เพิ่มชื่อและคำอธิบาย ด้วยตัวเองและแบ่งแยกประเภทภายใน North American Industry Classification System (NAICS) ในขณะนี้ org บรรจุ properties และ classifications สำหรับ Coffee Break ที่ถูกเพิ่มให้กับออบเจกต์ Collection (org) ในบรรทัดสุดท้าย orgs ถูก save โดย lcm ซึ่งจะจัดการกับวงจรชีวิตของออบเจกต์ Organization ที่บรรจุนอยู่ใน orgs

```
RegistryService rs = connection.getRegistryService();
```

```
BusinessLifeCycleManager lcm = rs.getBusinessLifeCycleManager();
```

```
Organization org = lcm.createOrganization("The Coffee Break");
```

```
Org.setDescription("Purveyor of only the finest coffees.Established 1895");
```

```
ClassificationSchema cSchema = lcm.createClassificationScheme("ntis-gov:naics",  
"North American Industry Classification System");
```

```
javax.xml.registry.infomodel.Key cKey = lcm.createKey("uuid:C0B9FE13-179F-413D-8A5B-5004DB8E5BB2");
```

```
cScheme.setKey(cKey);
```

```
Classification classification = (Classification)lcm.createClassification(cScheme, "Snack and Nonalcoholic Beverage Bars", "722213");
```

```
Collection classifications = new ArrayList();
```

```
classifications.add(classification);
```

```
org.addClassifications(classifications);
```

```
Collection orgs = new ArrayList();
```

```
orgs.add(org);
```

```
lcm.saveOrganizations(orgs);
```

การค้นหาการจดทะเบียน

ธุรกิจหนึ่ง ๆ สามารถใช้ JAXR ค้นหาการจดทะเบียน (registry) ของธุรกิจอื่น ๆ ได้ดังตัวอย่างโค้ดข้างล่าง ใช้ออบเจกต์ BusinessQueryManager (bqm) ค้นหา The Coffee Break ก่อนที่ bqm อ้างถึงเมธอด findOrganizations ในโค้ดนี้จำเป็นต้องกำหนดเกณฑ์ในการค้นหา ในกรณีนี้ตัวแปร 3 จาก 6 ตัวที่ใช้ค้นหาในเมธอด findOrganizations เพราะว่าตัวแปรตัวที่ 3, 5 และ 6 มีค่าเป็น null ซึ่งเกณฑ์นี้ไม่ถูกใช้จำกัดในการค้นหาอาร์กิวเมนต์ที่ 1, 2 และ 4 เป็นออบเจกต์ Collection ทั้งหมด ซึ่งมี findQualifiers และ namePatterns ที่กำหนดที่นี้ และมีเฉพาะเอลิเมนต์ใน findQualifiers เป็นการกำหนดค่า String ที่ไม่มีการส่งค่ากลับ ถ้าไม่เช่นนั้น name เป็น case-sensitive match จับคู่กับส่วนหนึ่งของ name ในตัวแปร namePatterns ตัวแปรนี้เป็นของออบเจกต์ Collection ที่มีเพียงหนึ่งเอลิเมนต์ หรือเรียกในทางธุรกิจว่า "Coffee" ออบเจกต์ Collection ตัวอื่น คือ classifications ถูกกำหนดเมื่อ The Coffee Break ได้จดทะเบียนด้วยตัวเอง (ดูตัวอย่างจากโค้ดข้างบน)

```
BusinessQueryManager bqm = rs.getBusinessQueryManager();
```

//กำหนดคุณสมบัติการค้นหา

```
Collection findQualifiers = new ArrayList();
```

```
findQualifiers.add(FindQualifier.CASE_SENSITIVE_MATCH);
```

```
Collection namePatterns = new ArrayList();
```

```
namePatterns.add("%Coffee%"); // ค้นหาชื่อที่มี 'Coffee'
```

// ค้นหาโดยใช้ name และ classifications

```
BulkResponse response = bpm.findOrganizations(findQualifiers, namePatterns, null,
    classifications, null, null);
```

```
Collection orgs = response.getCollection();
```

JAXR สนับสนุนการใช้คิวรีของ SQL ค้นหาการจดทะเบียน มีตัวอย่างการใช้ขอบเจ็ทต์

DeclarativeQueryManager ดังนี้

```
DeclarativeQueryManager dqm = rs.getDeclarativeQueryManager();
```

```
Query query = dqm.createQuery(Query.QUERY_TYPE_SQL,
```

```
“SELECT id FROM RegistryEntry WHERE name LIKE %Coffee% “ + “AND
    majorVersion >= 1 AND “ + “(majorVersion >= 2 OR minorVersion >= 3)”;
```

```
BulkResponse response2 = dqm.executeQuery(query);
```

ขอบเจ็ทต์ BulkResponse (response2) ประกอบด้วย id(uuid) จาก RegistryEntry ที่ชื่อคำว่า “Coffee” เป็นส่วนหนึ่งของ name และมีจำนวนเวอร์ชันเป็น 1.3 หรือมากกว่า

2.3.5 JAX-RPC [Sun Microsystem. 2002]

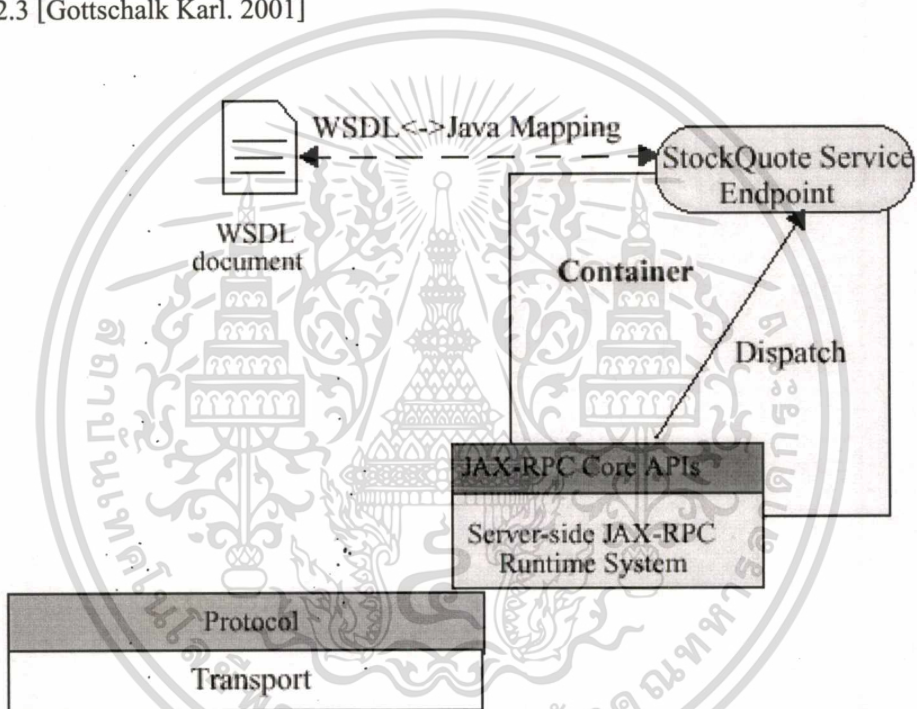
Java API for XML-based RPC (JAX-RPC) ทำให้การเขียนแอปพลิเคชันด้วยภาษาจาวา มีความเป็นไปได้ที่จะใช้ XML มีการเรียกใช้โพรซีเยอร์ระยะไกล (RPC)

ภาษาจาวามี API สำหรับสร้างการเรียกใช้โพรซีเยอร์ระยะไกลอยู่ 2 ตัวคือ Java IDL และ Remote Method Invocation (RMI) ความแตกต่างที่ว่า JAX-RPC ขึ้นอยู่กับ XML และทำงานคู่กับ เว็บเซอร์วิส ส่วน JAVA IDL ขึ้นอยู่กับ CORBA (Common Object Request Broker Architecture)

และใช้ภาษาการนิยามอินเทอร์เฟซของกลุ่มการจัดการออบเจกต์ (OMG IDL) ส่วน RMI ขึ้นอยู่กับ RPC Sun จะสนับสนุน CORBA และ RMI ต่อไปเพื่อพัฒนา JAX-RPC

การใช้ JAX-RPC

JAX-RPC ในแพลตฟอร์ม J2EE ทำให้ใช้เว็บเซอร์วิสง่ายขึ้น เว็บเซอร์วิสที่ใช้ RPC เป็นการรวมพร็อกซีเซอร์วิสที่ถูกเรียกใช้จากไคลเอนต์ระยะไกลบนอินเทอร์เน็ต บริการนี้อยู่ในแอปพลิเคชันฝั่งเซิร์ฟเวอร์ เช่น การสอบถามราคา โดยใช้เว็บเซอร์วิสแบบ RPC-base โดยไคลเอนต์จะร้องขอข้อมูลราคาที่ต้องการผ่านโพรโทคอล SOAP และเซิร์ฟเวอร์จะส่งราคากลับมาโดยใช้โพรโทคอล SOAP ดังรูปที่ 2.3 [Gottschalk Karl. 2001]



รูปที่ 2.3 แผนภาพแสดงตัวอย่างบริการแจ้งราคาหุ้น โดยใช้แนวคิดของ JAX-RPC

เว็บเซอร์วิสมีรายละเอียดการให้บริการที่ถูกกำหนดโดย Web Service Description Language (WSDL) ผู้ขอบริการ (Web Client) สามารถดูเอกสารของ WSDL เพื่อเรียกใช้บริการที่ต้องการและใช้โปรแกรมภาษาจาวา (JAX-RPC) หรือภาษาอื่นขอใช้บริการก็ได้ ในทางกลับกัน ไคลเอนต์ที่ใช้ภาษาอื่นสามารถร้องขอบริการที่ถูกกำหนดไว้บนแพลตฟอร์มจาวาก็ได้

ถึงแม้ว่า JAX-RPC อิมพลีเมนต์การเรียกใช้พร็อกซีเซอร์วิสระยะไกลเหมือนกับข้อความ SOAP ที่มีการร้องขอ-ตอบรับ แต่ JAXM เป็นรูปแบบที่แข็งแกร่งกว่าของข้อความ SOAP ซึ่งผู้พัฒนานิยมใช้กัน ในความเป็นจริงโพรโทคอลระดับที่สูงขึ้นไป เช่น ebXML ถูกกำหนดอยู่ชั้นบนสุดของ SOAP เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการรวบรวมคุณลักษณะของ JAXM และ RPC ที่จัดทำให้ รวมทั้ง JAX-RPC ที่ไม่ได้จัดทำให้ มีดังนี้

- การส่งข้อความแบบอะซิงโครนัส (Asynchronous Messaging)
- การจัดเส้นทาง (Routing) ของข้อความที่มีมากกว่า 1 ปาร์ตี้
- ลักษณะของข้อความที่น่าเชื่อถือ เช่นการรับประกันการส่ง

JAX-RPC เป็นทางเลือกที่ดีกว่าสำหรับแอปพลิเคชันที่หลีกเลี่ยงความซับซ้อนของข้อความ SOAP และที่มีการสื่อสารโดยใช้โมเดลของ RPC สิ่งที่สำคัญคือ ใช้ JAXM หรือ JAX-RPC ทำให้ใช้ข้อความ XML ได้อย่างสะดวกโดยใช้ภาษาจาวา



บทที่ 3

การวิเคราะห์และออกแบบระบบ

เนื่องจากโรงงานผลิตชิ้นส่วนต่าง ๆ จากวัตถุดิบพลาสติก ต้องการสั่งซื้อวัตถุดิบจากซัพพลายเออร์ต่าง ๆ เพื่อให้ได้วัตถุดิบตรงตามความต้องการมากที่สุด ในระบบเดิมเมื่อมีรายการสั่งซื้อ พนักงานจัดซื้อต้องเข้าไปค้นหาหลาย ๆ เว็บไซต์ เพื่อให้ได้ข้อมูลวัตถุดิบและนำมาพิจารณาเลือกรายการที่ต้องการสั่งซื้อให้มีคุณสมบัติตามที่ต้องการ เช่น ราคาถูกที่สุด จึงเป็นการสิ้นเปลืองเวลาสำหรับการค้นหา และมีโอกาสเสียเปรียบคู่แข่งในทางธุรกิจ ฉะนั้นจึงนำแนวคิดของเว็บเซอร์วิส มาใช้ในการพัฒนาระบบการจัดซื้อ โดยให้บริการค้นหารายการสั่งซื้อที่ต้องการ และสั่งซื้อสินค้าได้ผ่านทางเว็บไซต์เดียว ก็สามารถเปรียบเทียบราคาวัตถุดิบจากซัพพลายเออร์ต่าง ๆ แล้วตัดสินใจสั่งซื้อได้เลย

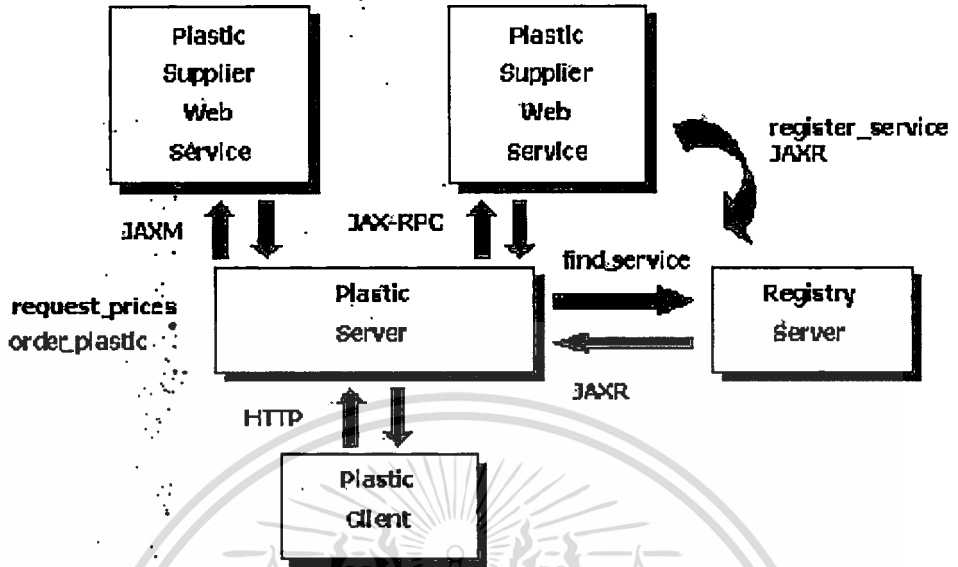
3.1 ระบบการจัดซื้อโดยใช้เว็บเซอร์วิส

เป็นการนำเทคโนโลยี Java API for XML มาประยุกต์ใช้งานกับ เว็บแอปพลิเคชันการสั่งซื้อสินค้าผ่านทางอินเทอร์เน็ต แสดงภาพรวมของระบบดังรูปที่ 3.1 โดยมีระบบงานแบ่งออกเป็น 2 ส่วนหลัก ๆ ดังนี้

1. การสั่งซื้อสินค้าบนอินเทอร์เน็ต จะมีการติดต่อสื่อสารกับ Plastic Server เพื่อสั่งซื้อสินค้าในลักษณะออนไลน์ โดยมีจุดประสงค์ดังนี้

- เพื่อค้นหาข้อมูลรายการสินค้าและวัตถุดิบที่พนักงานต้องการสั่งซื้อจากซัพพลายเออร์ต่าง ๆ และนำข้อมูลรายการสินค้าและวัตถุดิบที่ได้มาแสดงบนหน้าจอ เพื่อให้พนักงานสั่งซื้อทำการเปรียบเทียบ และเลือกสั่งซื้อสินค้าและวัตถุดิบจากซัพพลายเออร์ที่ต้องการ
- เป็นตัวแทนในการสั่งซื้อสินค้าและวัตถุดิบ โดยรับคำสั่งซื้อจากพนักงานสั่งซื้อ แล้วส่งต่อไปให้กับเว็บเซอร์วิสของซัพพลายเออร์ที่ต้องการสั่งซื้อนั้น ๆ

2. งานในระบบขาย ฝั่งซัพพลายเออร์ในที่นี้จะกล่าวถึง การจัดทำรายการวัตถุดิบจากร่องขอของผู้ซื้อ และจะยืนยันการสั่งซื้อกลับไปให้ผู้ซื้อ



รูปที่ 3.1 ภาพรวมของระบบการจัดซื้อโดยใช้ JAX [Sun Microsystem. 20002]

จากรูปที่ 3.1 การทำงานของระบบจัดซื้อโดยใช้ JAX เริ่มจาก Plastic Client ร้องขอข้อมูลไปยัง Plastic Web Services ในรูปของ SOAP เมสเสจส่งผ่านทางโปรโตคอล HTTP จากนั้น Plastic Web Services ใช้ JAX-RPC เพื่อร้องขอการบริการที่มีอยู่ใน Plastic Supplier Web Services อื่น ๆ หลังจากนั้นใช้ JAXM ส่ง SOAP เมสเสจไปยัง Plastic Web Service กรณีที่ไม่ทราบข้อมูลของ Service ของซัพพลายเออร์ สามารถค้นหาได้จาก Registry Server โดยใช้ JAXR ค้นหา Service ที่ต้องการ สำหรับ Plastic Supplier Web Services ที่ยังไม่ได้ลงทะเบียนกับ Registry Server ต้องลงทะเบียนบริการนั้นก่อนโดยใช้ JAXR

การทำงานของ Plastic Web Services ที่อยู่ใน Plastic Server มีรายละเอียดดังนี้

- มีการร้องขอเมธอดเพื่อขอใช้บริการจากซัพพลายเออร์อื่น ๆ โดยใช้ JAX-RPC
- มีการส่งเมสเสจไปยัง Plastic Supplier Web Services ได้แก่คิวรีของรายการราคา วัตถุดิบ และรายการสั่งซื้อ โดยใช้ JAXM
- Plastic Supplier Web Services มีการตอบสนองด้วยเมสเสจข้อมูลวัตถุดิบ ราคาของ วัตถุดิบ และการยืนยันการสั่งซื้อ โดยใช้ JAXM
- Plastic Web Services มีการตอบสนองข้อมูลรายการราคาพลาสติก เพื่อนำรายการราคา พลาสติกที่ได้รับจาก Supplier มาแสดงให้ผู้ซื้อ ได้เปรียบเทียบราคา

- Plastic Web Services ใช้ Servlet ที่ทำงานบน Tomcat และ JavaBean Component ในการเก็บข้อมูลต่าง ๆ ที่ได้รับมาจาก Plastic Supplier Web Service และการคำนวณต่าง ๆ เช่น การคำนวณราคาสินค้าที่ต้องการขายให้กับผู้ซื้อ เป็นต้น
- Plastic Web Services และ Plastic Supplier Web Services ต้องมี SOAP Listener ไว้สำหรับทำหน้าที่คอยฟังการเรียกใช้ และโปรแกรมที่ทำหน้าที่ติดต่อกับโปรโตคอล SOAP เพื่อใช้ในการสร้างและอ่าน SOAP เมสเซจซึ่งส่วนใหญ่แล้วโปรแกรมนี้จะมี XML Parser เพื่อแปลงข้อมูลให้อยู่ในรูปที่เข้าใจได้ง่าย และนำไปใช้งานได้

จากการทำงานข้างต้นสามารถนำมาออกแบบการทำงานได้เป็น 3 ระบบ ดังนี้

1. ระบบเว็บเซอร์วิสของซัพพลายเออร์รายที่ 1 (Supplier 1) ให้บริการการร้องขอข้อมูลวัตถุดิบพลาสติก การสั่งซื้อวัตถุดิบ และการตรวจสอบสถานะการจัดส่งวัตถุดิบ โดยจะพัฒนาระบบดังกล่าวโดยใช้เทคโนโลยี JAX-RPC และใช้ระบบฐานข้อมูล Postgres
2. ระบบเว็บเซอร์วิสของซัพพลายเออร์รายที่ 2 (Supplier 2) ให้บริการการร้องขอข้อมูลวัตถุดิบพลาสติก การสั่งซื้อวัตถุดิบ และการตรวจสอบสถานะการจัดส่งวัตถุดิบ โดยจะพัฒนาระบบดังกล่าวโดยใช้เทคโนโลยี JAX-RPC และใช้ระบบฐานข้อมูล MySQL
3. ระบบเว็บเซอร์วิสของงานสั่งซื้อ (Plastic Server) ทำหน้าที่ขอรับบริการ โดยการเรียกใช้งานเว็บเซอร์วิสของซัพพลายเออร์ต่างๆ ซึ่งจะพัฒนาด้วย Java Servlets, JSP และ JavaBeans และใช้ระบบฐานข้อมูล Postgres

3.2 Supplier 1 Web Services และ Supplier 2 Web Services

เป็นระบบเว็บเซอร์วิสของซัพพลายเออร์ ที่เป็นผู้ให้บริการ โดยจะรองรับการร้องขอและส่งผลการตอบรับกลับไปยังผู้ร้องขอ (Plastic Server) โดยใช้เทคโนโลยี JAX-RPC ซึ่งมีการให้บริการดังนี้

- การค้นหาข้อมูลวัตถุดิบ
- การสั่งซื้อวัตถุดิบ
- การตรวจสอบสถานะของการจัดส่งวัตถุดิบ

การนำระบบดังกล่าวไปใช้งาน จำเป็นต้องมีการติดตั้งสภาพแวดล้อมดังนี้

- One Supplier Web Services ติดตั้งอยู่บนระบบปฏิบัติการ Windows XP , ระบบฐานข้อมูล Postgres (ตั้งอยู่บนแบบจำลอง Unix : Cygwin for Windows XP)
- Two Supplier Web Service ติดตั้งอยู่บนระบบปฏิบัติการ Windows XP , ระบบฐานข้อมูล MySQL

3.3 Plastic Server

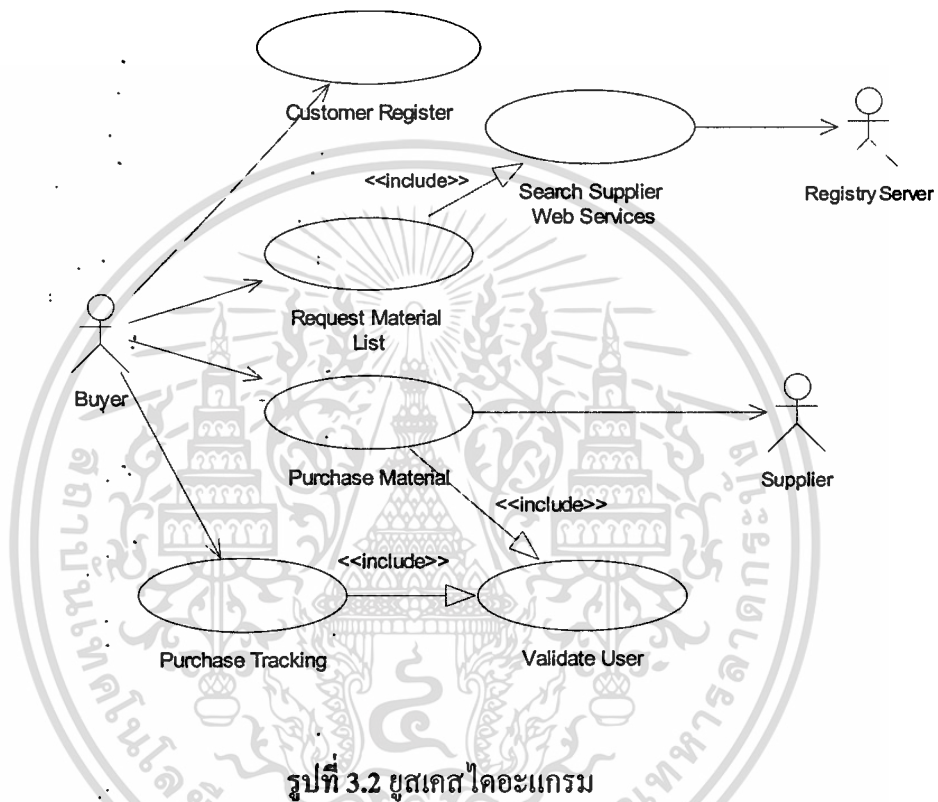
เป็นเว็บแอปพลิเคชันของระบบงานสั่งซื้อ โดยทำหน้าที่เรียกใช้งานเว็บเซอร์วิสของซัพพลายเออร์ต่างๆ โดยใช้ Java Servlets , JSP และ JavaBeans ติดตั้งอยู่บนระบบปฏิบัติการ Windows XP และใช้ระบบฐานข้อมูล Postgres (ตั้งอยู่บนแบบจำลอง Unix : Cygwin for Windows XP)

ขั้นตอนการทำงานของ Plastic Server มีดังนี้

1. Plastic Server จะใช้เทคโนโลยี JAXR ในการค้นหาข้อมูลการบริการ ของซัพพลายเออร์ ที่ลงทะเบียนไว้กับ Registry Server
2. เมื่อพบรายชื่อและข้อมูลต่าง ๆ ของซัพพลายเออร์ทั้ง 2 ราย คือ ซัพพลายเออร์รายที่ 1 และซัพพลายเออร์รายที่ 2 แล้ว จะทำการบันทึกข้อมูลลงบนฐานข้อมูลของ Plastic Server
3. Plastic Server จะใช้เทคโนโลยี JAX-RPC ในการติดต่อกับซัพพลายเออร์รายที่ 1 และซัพพลายเออร์รายที่ 2
4. เมื่อลูกค้าทำการร้องขอข้อมูลวัตถุดิบ ต้องการสั่งซื้อวัตถุดิบ หรือตรวจสอบสถานะของการสั่งซื้อวัตถุดิบ Plastic Server จะทำการติดต่อไปยังซัพพลายเออร์ เพื่อทำการตอบรับการร้องขอตามที่ลูกค้าต้องการ
5. ลูกค้าจะสามารถทราบรายการวัตถุดิบ พร้อมราคา จากซัพพลายเออร์รายต่างๆ แล้วสามารถเปรียบเทียบราคาจากซัพพลายเออร์รายต่างๆ ได้
6. เมื่อลูกค้าทำการสั่งซื้อวัตถุดิบ Plastic Server จะทำการบันทึกข้อมูลการสั่งซื้อไว้ภายในฐานข้อมูล พร้อมกับส่งรายการสั่งซื้อไปยังซัพพลายเออร์รายนั้นๆ
7. เมื่อซัพพลายเออร์ตอบรับการสั่งซื้อกลับมา Plastic Server จะรับการยืนยันรายการสั่งซื้อนั้น แล้วแจ้งให้ลูกค้าทราบ

3.4 ยูสเคสไดอะแกรม

การวิเคราะห์และออกแบบระบบ จะเริ่มจากการเก็บรวบรวมความต้องการต่าง ๆ ของระบบ ทำให้ทราบถึงฟังก์ชันการทำงานของระบบ และแอกเตอร์ที่เกี่ยวข้องกับระบบ ที่จะนำมาใช้เขียนเป็นยูสเคสไดอะแกรม ดังรูปที่ 3.2



แอกเตอร์ที่เกี่ยวข้องกับระบบ ได้แก่

- Buyer เป็นพนักงานจัดซื้อที่ทำหน้าที่จัดซื้อสินค้าที่ต้องการเข้าสู่โรงงาน
- Registry Server เป็นผู้ให้บริการค้นหาและลงทะเบียนบริการเว็บเซอร์วิสในเครือข่าย
- Supplier ผู้จัดจำหน่ายวัตถุดิบให้กับ โรงงาน

ยูสเคสที่เกี่ยวข้องกับระบบ ได้แก่

- Request Material List พนักงานจัดซื้อร้องขอข้อมูลรายการวัตถุดิบที่ต้องการ ไปยังซัพพลายเออร์ต่าง ๆ ผ่านเว็บเซอร์วิส ซึ่งเมื่อได้รับข้อมูลจากซัพพลายเออร์ โปรแกรมจะทำการประมวลผลรายการราคาวัตถุดิบ และจัดเรียงรายการวัตถุดิบ ให้พนักงานจัดซื้อเปรียบเทียบราคาได้ง่ายขึ้น

- Search Supplier Web Services ใช้สำหรับค้นหาซัพพลายเออร์ที่ต้องการจาก Registry Server ซึ่งจะ ได้ข้อมูลผู้ให้บริการจำหน่ายสินค้าที่ต้องการจะสั่งซื้อ
- Purchase Material พนักงานจัดซื้อจะกำหนดรายการสั่งซื้อสินค้าต่าง ๆ เพื่อจัดซื้อเข้าบริษัท โดยเลือกรายการสินค้าและวัตถุดิบได้ตามความต้องการ พร้อมกับระบุจำนวนสินค้า และป้อนที่อยู่สำหรับการจัดส่งสินค้า เมื่อทำการเก็บข้อมูลที่สั่งซื้อแล้ว โปรแกรมจะแสดงหน้าจอ ให้ผู้ใช้ยืนยันการสั่งซื้อสินค้า หลังจากนั้นจะจัดส่งรายการสั่งซื้อดังกล่าว ให้กับซัพพลายเออร์ ผ่าน Plastic Server
- Validate User ตรวจสอบรหัสผ่านในการเข้าใช้บริการสั่งซื้อผ่านเว็บเซอร์วิส
- Customer Register เป็นการให้ลูกค้า หรือผู้สั่งซื้อลงทะเบียนกับระบบในฝั่ง Plastic Server โดยโปรแกรมจะเก็บข้อมูลต่าง ๆ ของลูกค้าลงฐานข้อมูล
- Purchase Tracking ใช้สำหรับตรวจสอบ สถานะของการสั่งซื้อวัตถุดิบ ของลูกค้า ว่าดำเนินการถึงขั้นตอนไหนแล้ว โดยที่โปรแกรมจะตรวจสอบว่า ลูกค้าเคยลงทะเบียนมาก่อนไหม แล้วระบบจะเรียกใช้เว็บเซอร์วิสของซัพพลายเออร์ และแจ้งสถานะของการสั่งซื้อให้ทราบผ่านทางเบรดาเซอร์

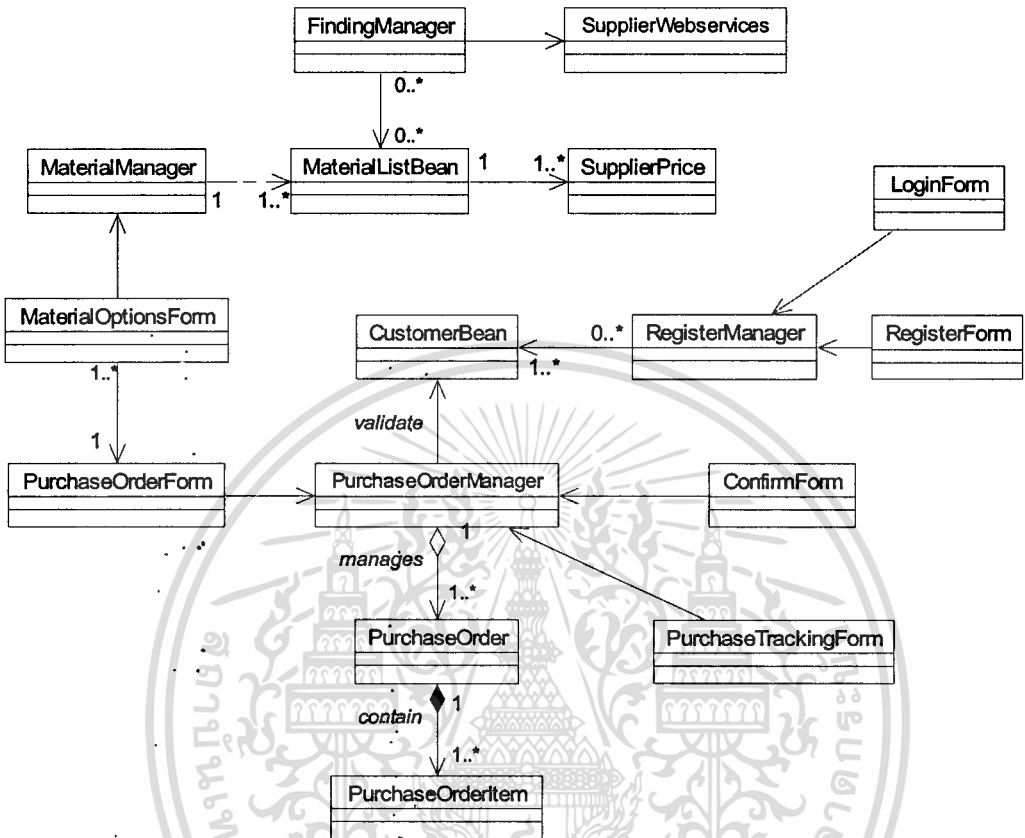
3.5 คลาสไดอะแกรม

เมื่อผ่านขั้นตอนวิเคราะห์เข้าสู่ขั้นตอนออกแบบ ทำให้สามารถออกแบบคลาสไดอะแกรมสำหรับอธิบายรายละเอียดของคลาสภายในโปรแกรมที่ต้องสร้างขึ้น และความสัมพันธ์ต่าง ๆ ระหว่างคลาส โดยคลาสใช้สัญลักษณ์เป็นสี่เหลี่ยมผืนผ้า ภายในแบ่งเป็น ชื่อคลาส คุณสมบัติ และเมธอดเรียงจากบนลงล่าง ดังรูปที่ 3.3

คลาสภายในโปรแกรมที่ต้องสร้างขึ้น ได้แก่

- FindingManager เป็นคลาสสำหรับควบคุมการค้นหา ข้อมูล URL และที่อยู่สำหรับการติดต่อขอบริการจาก Supplier Web Services
- MaterialListBean เป็นคลาสสำหรับจัดการกับข้อมูลวัตถุดิบและสินค้าต่าง ๆ ที่ได้จากการค้นหาจากซัพพลายเออร์
- SupplierPrice เป็นคลาสสำหรับจัดการกับข้อมูลรายการราคาของวัตถุดิบและสินค้าจากซัพพลายเออร์
- SupplierWebservices เป็นคลาสสำหรับจัดการกับข้อมูล URL ของผู้จัดจำหน่ายวัตถุดิบและสินค้าต่าง ๆ ที่ต้องการสั่งซื้อ

- MaterialManager เป็นคลาสสำหรับควบคุมการจัดการข้อมูลวัตถุดิบและสินค้าต่าง ๆ
- MaterialOptionsForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอค้นหาและเลือกรายการวัตถุดิบต่าง ๆ
- PurchaseOrderForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอบันทึกรายการสั่งซื้อวัตถุดิบต่าง ๆ
- PurchaseOrderManager เป็นคลาสสำหรับควบคุมการสั่งซื้อรายการวัตถุดิบต่าง ๆ
- PurchaseOrderBean เป็นคลาสสำหรับจัดการกับข้อมูลรายการสั่งซื้อส่วนหลัก
- PurchaseOrderItemBean เป็นคลาสสำหรับจัดการกับข้อมูลรายการสั่งซื้อส่วนรายละเอียด
- PurchaseTrackingForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอสอบถามสถานะของการสั่งซื้อวัตถุดิบต่าง ๆ
- ConfirmForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอยืนยันการสั่งซื้อวัตถุดิบต่าง ๆ
- CustomerBean เป็นคลาสสำหรับจัดการกับข้อมูลลูกค้า
- RegisterManager เป็นคลาสสำหรับควบคุมการจัดการลงทะเบียนลูกค้า เพื่อใช้เป็นตัวผ่านในการสั่งซื้อสินค้าจากระบบ และตรวจสอบผู้ใช้งานเมื่อมีการสั่งซื้อ หรือสอบถามสถานะของการสั่งซื้อ
- RegisterForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอลงทะเบียนลูกค้าใหม่



รูปที่ 3.3 คลาสไดอะแกรม

3.6 ซีควเอนซ์ไดอะแกรม

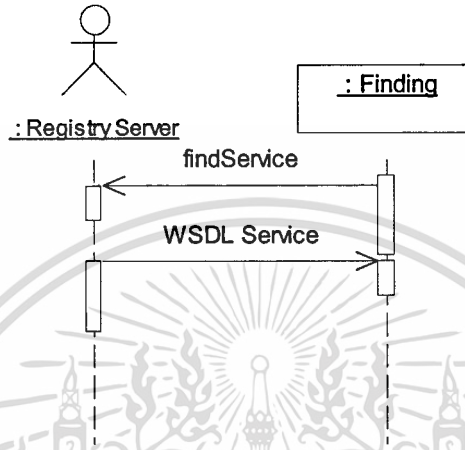
เมื่อผ่านขั้นตอนการออกแบบและมีความต้องการอธิบายความสัมพันธ์ระหว่างยูสเคส และ คลาสไดอะแกรม ทำให้สามารถออกแบบซีควเอนซ์ไดอะแกรม โดยใช้สัญลักษณ์ดังนี้

- สีเหลี่ยมผืนผ้าใช้แทนคลาสหรือออบเจกต์ ภายในสีเหลี่ยมจะมีชื่อคลาสหรือออบเจกต์
- เส้นประแนวตั้งใช้แสดงลำดับเวลาจากบนลงล่าง
- เส้นตรงแนวนอนมีหัวลูกศร แสดงกิจกรรมที่เกิดขึ้นจากคลาส หรือออบเจกต์ในไดอะแกรม
- สีเหลี่ยมสี่ที่วางบนเส้นแสดงเวลา หมายถึงช่วงเวลาที่ยอบเจกต์ยัง active อยู่

ในการพัฒนาระบบการจัดซื้อโดยใช้ JAX มีซีควเอนซ์ไดอะแกรม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.6.1 ซีเควนซ์ไคอะแกรมของ Search Supplier Web Services แสดงดังรูปที่ 3.4



รูปที่ 3.4 ซีเควนซ์ไคอะแกรมของ Search Supplier Web Services

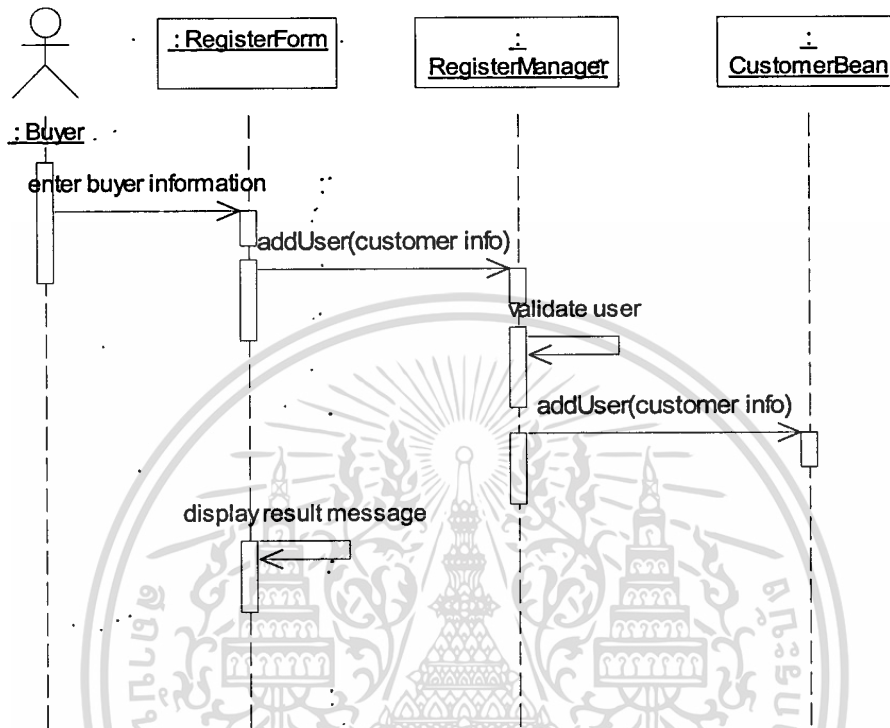
อธิบายการทำงานได้ดังนี้

1. Registry Server ให้บริการค้นหาข้อมูลการบริการของซัพพลายเออร์ เคยมาลงทะเบียน และ ประกาศการให้บริการไว้
2. Plastic Server จะทำค้นหาบริการที่เกี่ยวข้องกับสินค้าที่ผู้ซื้อต้องการ ว่ามีซัพพลายเออร์รายใด ที่ให้บริการเกี่ยวกับสินค้านั้น จาก Registry Server
3. ผลลัพธ์ที่ได้ถูกอธิบายในรูปของ WSDL

3.6.2 ซีเควนซ์ไคอะแกรมของ Customer Register แสดงดังรูปที่ 3.5

อธิบายการทำงานได้ดังนี้

1. ผู้ซื้อป้อนข้อมูลชื่อและที่อยู่ สำหรับการลงทะเบียนเป็นสมาชิก เพื่อเข้าใช้ระบบงาน
2. RegisterManager จะทำการรับข้อมูลการลงทะเบียน และนำไปตรวจสอบว่าถูกต้องหรือไม่ ก่อนทำการบันทึกข้อมูลลงฐานข้อมูลของลูกค้า
3. เมื่อลงทะเบียนเรียบร้อยแล้ว จะแสดงข้อความให้ผู้ใช้ทราบ

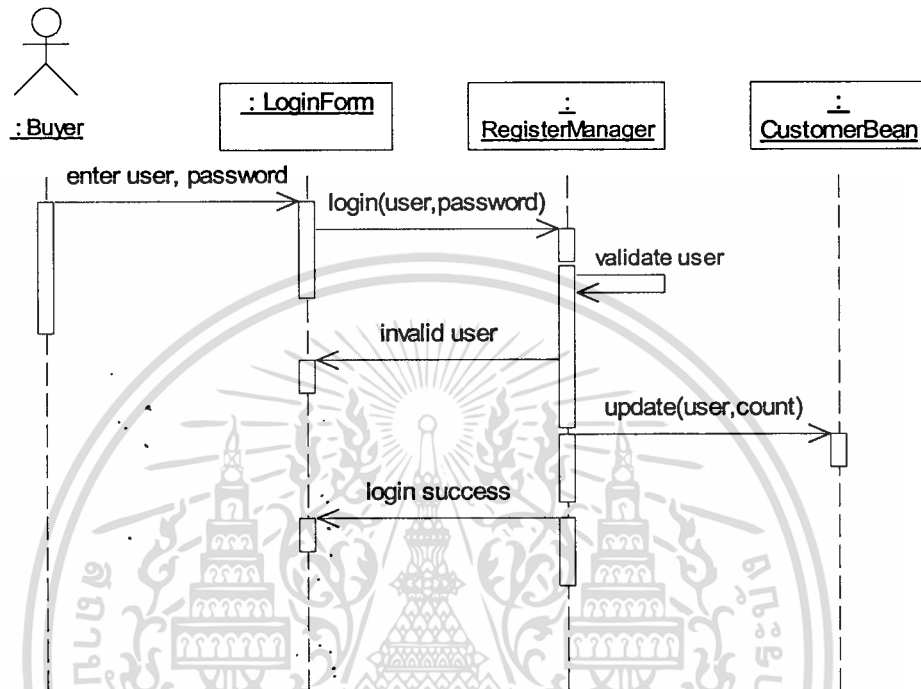


รูปที่ 3.5 ซีเควนซ์ไดอะแกรมของ Customer Register

3.6.3 ซีเควนซ์ไดอะแกรมของ Validate User แสดงดังรูปที่ 3.6

อธิบายการทำงานได้ดังนี้

1. ผู้ซื้อเลือกเมนูล็อกอินเข้าสู่ระบบ พร้อมทั้งใส่ชื่อและรหัสผ่าน
2. RegisterManager จะรับข้อมูลชื่อและรหัสผ่าน มาตรวจสอบกับข้อมูลในฐานข้อมูลของลูกค้า ว่าถูกต้องหรือไม่ ก่อนให้เข้าใช้ระบบงาน
3. ถ้าไม่ถูกต้องระบบจะทำการเก็บจำนวนครั้งในการล็อกอิน ถ้าเกินจำนวนครั้งที่กำหนดระบบจะแสดงข้อความเตือนให้ผู้ใช้งานทราบ
4. ถ้าถูกต้องระบบจะอนุญาตให้เข้าสู่หน้าจอการทำงาน of ระบบ

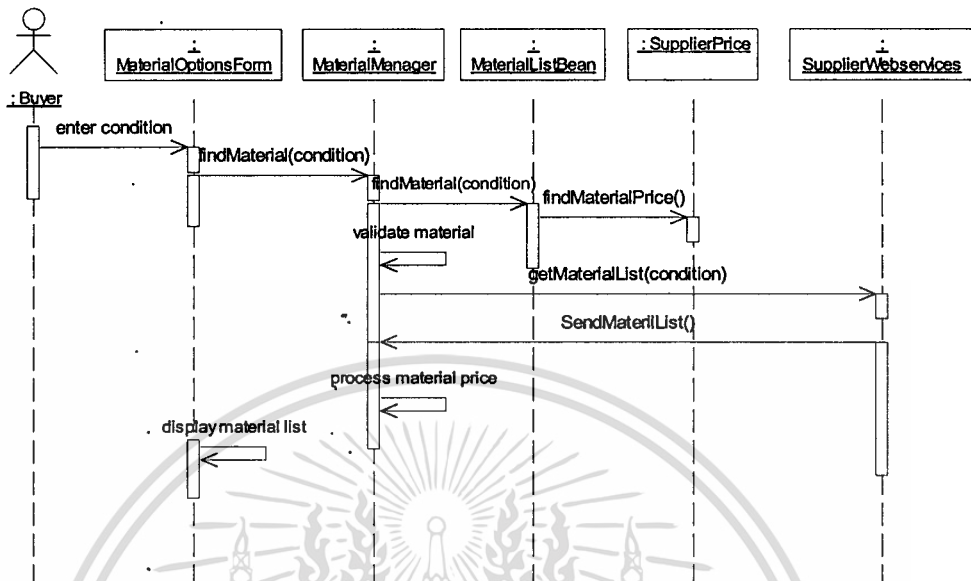


รูปที่ 3.6 ซีควนซ์ไดอะแกรมของ Validate User

3.6.4 ซีควนซ์ไดอะแกรมของ Request Material List แสดงดังรูปที่ 3.7

อธิบายการทำงานได้ดังนี้

1. การค้นหาข้อมูลวัตถุดิบที่ต้องการสั่งซื้อ เริ่มจากผู้ซื้อใส่เงื่อนไขสำหรับการค้นหาเช่น ระบุชื่อซัพพลายเออร์ หรือรายชื่อบริการที่ต้องการจะเข้าไปขอราคาและสั่งซื้อ
2. โปรแกรมจะทำการค้นหาข้อมูลรายการวัตถุดิบและราคา ตามเงื่อนไขที่ระบุไปยังเว็บไซต์ของซัพพลายเออร์นั้น ๆ
3. โดยผลลัพธ์ที่ได้จะถูกจัดการ โดย MaterialManager แล้วประมวลผลมาแสดงให้ผู้ซื้อบนหน้าจอได้เปรียบเทียบราคา แล้วตัดสินใจสั่งซื้อสินค้าได้

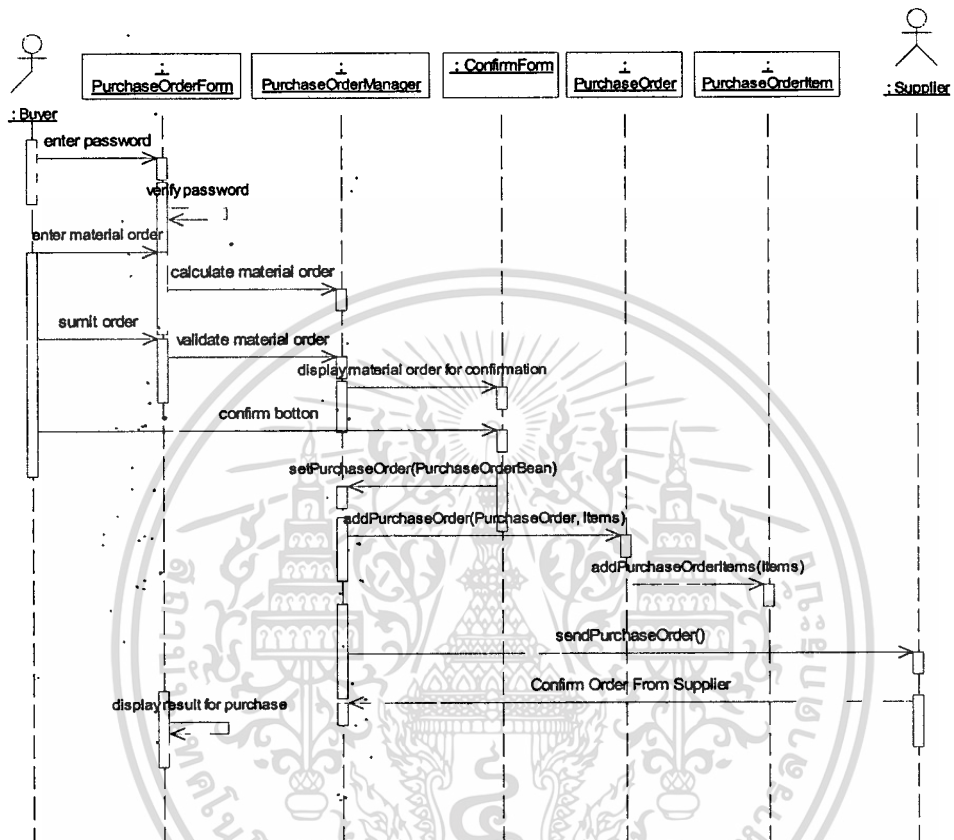


รูปที่ 3.7 ซีเควนซ์ไดอะแกรมของ Request Material List

3.6.5 ซีเควนซ์ไดอะแกรมของ Purchase Material แสดงดังรูปที่ 3.8

การสั่งซื้อวัตถุดิบและสินค้า อธิบายการทำงาน ได้ดังนี้

1. ผู้ซื้อเข้าสู่ระบบการใช้งานแล้ว ทำการเลือกและระบุจำนวนวัตถุดิบที่ต้องการจะสั่งซื้อ
2. เมื่อผู้ซื้อกดปุ่มให้คำนวณราคา PurchaseOrderManager จะทำการตรวจสอบและคำนวณราคาสินค้าที่สั่งซื้อ
3. เมื่อผู้ซื้อกดปุ่มสั่งซื้อ PurchaseOrderManager จะทำการตรวจสอบรายการที่สั่งซื้อ แล้วแสดงหน้าจอให้ผู้ซื้อ ได้ป้อนข้อมูลชื่อและที่อยู่ในการจัดส่งสินค้า
4. เมื่อผู้ซื้อยืนยันการสั่งซื้อสินค้าแล้ว ระบบจะทำการจัดเก็บข้อมูลการสั่งซื้อลงฐานข้อมูลของ Plastic Server และจะส่งไปให้ซัพพลายเออร์นั้น ๆ อีกทอดหนึ่ง
5. เมื่อทางฝั่งซัพพลายเออร์ได้รับรายการสั่งซื้อสินค้าแล้ว จะทำการตอบรับกลับมา

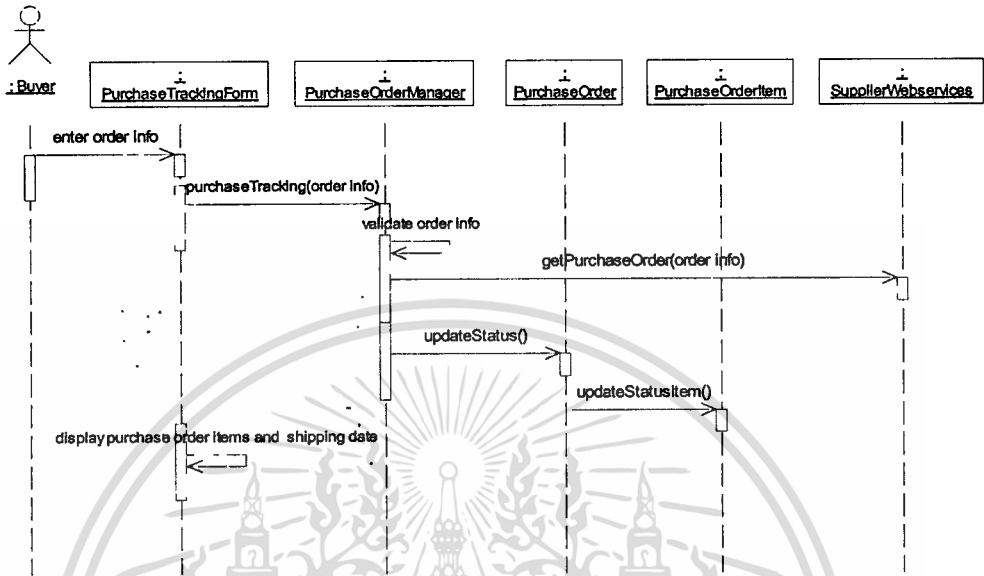


รูปที่ 3.8 ซีควেনซ์ไดอะแกรมของ Purchase Material

3.6.6 ซีควেনซ์ไดอะแกรมของ Purchase Tracking แสดงดังรูปที่ 3.9

การสอบถามสถานะของการสั่งซื้อสินค้า อธิบายการทำงานได้ดังนี้

1. เมื่อผู้ซื้อใส่เลขที่ใบสั่งซื้อสินค้า ที่ต้องการสอบถาม
2. ระบบจะทำการตรวจสอบเลขที่ใบสั่งซื้อ และดึงข้อมูลใบสั่งซื้อจากเซฟฟลายเออร์
3. เมื่อมีการเปลี่ยนแปลงสถานะของใบสั่งซื้อนั้นระบบจะทำการ ปรับปรุงข้อมูลสถานะของใบสั่งซื้อสินค้านั้น แล้วดึงสถานะใบสั่งซื้อสินค้าให้ผู้ใช้ทราบผ่านทาง Browser



รูปที่ 3.9 ซีเควนซ์ไดอะแกรมของ Purchase Tracking

บทที่ 4

การออกแบบฐานข้อมูล

โครงการนี้มีระบบที่ใช้ติดต่อกับฐานข้อมูลอยู่ 3 ระบบ คือฐานข้อมูลของซัพพลายเออร์รายที่ 1, ฐานข้อมูลของซัพพลายเออร์รายที่ 2 และฐานข้อมูลของ Plastic Server โดยมีรายละเอียดดังต่อไปนี้

4.1 ฐานข้อมูลของซัพพลายเออร์รายที่ 1

ฐานข้อมูลของซัพพลายเออร์รายที่ 1 ใช้ Postgres เป็นฐานข้อมูล ประกอบด้วยตารางดังต่อไปนี้

1. CUSTOMER เก็บข้อมูลของลูกค้า แสดงรายละเอียดดังตารางที่ 4.1
2. ADDRESS เก็บข้อมูลที่อยู่ของลูกค้า แสดงรายละเอียดดังตารางที่ 4.2
3. ORDER เก็บข้อมูลการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.3
4. ORDER_ITEM เก็บข้อมูลรายละเอียดของการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.4
5. MATERIAL เก็บข้อมูลวัตถุดิบพลาสติก ลูกค้า แสดงรายละเอียดดังตารางที่ 4.5

4.2 ฐานข้อมูลของซัพพลายเออร์รายที่ 2

ฐานข้อมูลของซัพพลายเออร์รายที่ 2 ใช้ MySQL เป็นฐานข้อมูล ประกอบด้วยตารางดังต่อไปนี้

1. CUSTOMER เก็บข้อมูลของลูกค้า แสดงรายละเอียดดังตารางที่ 4.1
2. ADDRESS เก็บข้อมูลที่อยู่ของลูกค้า แสดงรายละเอียดดังตารางที่ 4.2
3. ORDER เก็บข้อมูลการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.3
4. ORDER_ITEM เก็บข้อมูลรายละเอียดของการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.4
5. MATERIAL เก็บข้อมูลวัตถุดิบพลาสติก ลูกค้า แสดงรายละเอียดดังตารางที่ 4.5

ตารางที่ 4.1 แสดงรายละเอียดของตาราง CUSTOMER

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
CUSTOMER_ID	INT	PK	รหัสลูกค้า
INITIAL	VARCHAR (15)		คำนำหน้าชื่อ
FIRST_NAME	VARCHAR (50)		ชื่อลูกค้า
LAST_NAME	VARCHAR (50)		นามสกุลลูกค้า
ADDRESS_ID	INT	FK	รหัสที่อยู่ [ตาราง ADDRESS]

ตารางที่ 4.2 แสดงรายละเอียดของตาราง ADDRESS

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
ADDRESS_ID	INT	PK	รหัสที่อยู่
ADDRESS	VARCHAR (100)		ที่อยู่
CITY	VARCHAR(40)		เมือง
ZIP_CODE	VARCHAR(10)		รหัสไปรษณีย์
EMAIL	VARCHAR(20)		อีเมล
COUNTRY	VARCHAR(40)		ประเทศ
TELEPHONE	VARCHAR(15)		หมายเลขโทรศัพท์

ตารางที่ 4.3 แสดงรายละเอียดของตาราง ORDER

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
ORDER_ID	INT	PK	รหัสการสั่งซื้อ
ORDER_DATE	DATE		วันที่สั่งซื้อ
CUSTOMER_ID	INT	FK	รหัสลูกค้า [ตาราง CUSTOMER]
SHIP_DATE	DATE		วันที่ส่งสินค้า
ADDRESS_ID	INT	FK	รหัสที่อยู่ [ตาราง ADDRESS]
TOTAL	DOUBLE		ราคาสุทธิ
ORDER_NO	VARCHAR(15)		เลขที่ใบสั่งซื้อ
AGENT_CODE	CHAR(5)		รหัสตัวแทนจำหน่าย

ตารางที่ 4.3 แสดงรายละเอียดของตาราง ORDER (ต่อ)

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
STATUS	CHAR(1)		สถานะของการสั่งซื้อ

ตารางที่ 4.4 แสดงรายละเอียดของตาราง ORDER_ITEM

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
ORDER_ITEM_ID	INT	PK	รหัสรายละเอียดการสั่งซื้อ
ORDER_ID	INT	FK	รหัสการสั่งซื้อ [ตาราง ORDER]
MATERIAL_ID	INT	FK	รหัสสินค้าและวัตถุดิบ [ตาราง MATERIAL]
UNIT_PRICE	DOUBLE		ราคาต่อหน่วย
QUANTITY	DOUBLE		จำนวนที่สั่งซื้อ

ตารางที่ 4.5 แสดงรายละเอียดของตาราง MATERIAL

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
MATERIAL_ID	INT	PK	รหัสสินค้าและวัตถุดิบ
NAME	VARCHAR(50)		ชื่อสินค้าและวัตถุดิบ
UNIT_PRICE	DOUBLE		ราคาต่อหน่วยของสินค้าและวัตถุดิบ
IMAGE_URL	VARCHAR(100)		URL ที่เก็บรูปภาพของสินค้าและวัตถุดิบ
UNIT_NAME	VARCHAR(15)		ชื่อหน่วยของสินค้า

4.3 ฐานข้อมูลของ Plastic Server

ฐานข้อมูลของ Plastic Server ใช้ Postgres เป็นฐานข้อมูล ประกอบด้วยตารางดังต่อไปนี้

1. CUSTOMER เก็บข้อมูลชื่อและที่อยู่ของลูกค้า แสดงรายละเอียดดังตารางที่ 4.6
2. PURCHASE_ORDER เก็บข้อมูลหลักของการสั่งซื้อวัตถุดิบ แสดงรายละเอียดดังตารางที่ 4.7
3. PURCHASE_ORDER_ITEM เก็บข้อมูลรายละเอียดของการสั่งซื้อวัตถุดิบ แสดงรายละเอียดดังตารางที่ 4.8.

ตารางที่ 4.6 แสดงรายละเอียดของตาราง CUSTOMER

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
CUSTOMER_ID	INT	PK	รหัสลูกค้า
USER_NAME	VARCHAR(20)		ชื่อที่ใช้เข้าระบบ
PASSWORD	VARCHAR (20)		รหัสผ่านเข้าใช้ระบบ
INITIAL	VARCHAR (15)		คำนำหน้าชื่อ
FIRST_NAME	VARCHAR (50)		ชื่อลูกค้า
LAST_NAME	VARCHAR (50)		นามสกุลลูกค้า
ADDRESS	VARCHAR (100)		ที่อยู่
CITY	VARCHAR(40)		เมือง
ZIP_CODE	VARCHAR(10)		รหัสไปรษณีย์
EMAIL	VARCHAR(20)		อีเมล
COUNTRY	VARCHAR(40)		ประเทศ
TELEPHONE	VARCHAR(15)		หมายเลขโทรศัพท์

ตารางที่ 4.7 แสดงรายละเอียดของตาราง PURCHASE_ORDER

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
PO_ID	INT	PK	รหัสการสั่งซื้อ
PO_DATE	DATE		วันที่สั่งซื้อ
CUSTOMER_ID	INT	FK	รหัสลูกค้า
SHIP_DATE	DATE		วันที่ส่งสินค้า
SUPPLIER_URL	VARCHAR(100)		URL ของซัพพลายเออร์ที่สั่งซื้อ
TOTAL	DOUBLE		ราคาสุทธิ
STATUS	CHAR(1)		สถานะของการสั่งซื้อ
ADDRESS	VARCHAR (100)		ที่อยู่
CITY	VARCHAR(40)		เมือง
ZIP_CODE	VARCHAR(10)		รหัสไปรษณีย์
EMAIL	VARCHAR(20)		อีเมล
COUNTRY	VARCHAR(40)		ประเทศ

ตารางที่ 4.7 แสดงรายละเอียดของตาราง PURCHASE_ORDER (ต่อ)

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
TELEPHONE	VARCHAR(15)		หมายเลขโทรศัพท์
SUPP_ORDER_ID	VARCHAR(15)		รหัสการสั่งซื้อฝั่งซัพพลายเออร์

ตารางที่ 4.8 แสดงรายละเอียดของตาราง PURCHASE_ORDER_ITEM

ชื่อฟิลด์	ประเภท	Key	คำอธิบาย
PO_ITEM_ID	INT	PK	รหัสรายละเอียดของการสั่งซื้อ
PO_ID	INT	FK	รหัสการสั่งซื้อ [PURCHASE_ORDER]
RETAIL_PRICE	DOUBLE		ราคาขายปลีกต่อหน่วยของสินค้า
WHOLESALE_PRICE	DOUBLE		ราคาขายส่งต่อหน่วยของสินค้า
QUANTITY	DOUBLE		จำนวนที่สั่งซื้อ
MATERIAL_NAME	VARCHAR(50)		ชื่อสินค้าและวัตถุดิบที่สั่งซื้อ
UNIT_NAME	VARCHAR(15)		ชื่อหน่วยของสินค้า

บทที่ 5

การพัฒนาระบบ

ในการพัฒนาระบบการจัดซื้อโดยใช้ JAX สามารถอธิบายได้เป็นหัวข้อ ดังนี้

5.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ

การพัฒนาระบบในโครงการนี้ ได้ใช้เครื่องมือ และภาษาในการพัฒนา ดังนี้

5.1.1 ฮาร์ดแวร์

เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนา และทดสอบระบบ มีคุณสมบัติดังนี้

- CPU : Pentium 4 2.0 GHz.
- RAM : 512 MB.
- Hard Disk : 20 GB.
- Network Interface : 100BASE-TX/10BASE-T

5.1.2 ซอฟต์แวร์

ซอฟต์แวร์ที่ใช้ในการพัฒนา และทดสอบระบบ มีดังนี้

- Operating System : Microsoft Windows XP Home Edition
- Programming Language : J2SE 1.4.2
- Web Services : Java Web Services Developer Pack Version 1.3
- RDBMS : MySQL 4.0.12 และ Postgres
- Web Browser : Internet Explorer 6.0

5.1.3 เครื่องมือ

เครื่องมือที่ใช้ในการพัฒนา และทดสอบระบบ มีดังนี้

- Java IDE : NetBeans 3.5.1
- Web Development Tool : Macromedia Dreamweaver MX

- UML Tool : Rational Rose 2000 Enterprise Edition
- Build Tool : Apache Ant 1.5

5.2 หน้าจอการทำงานของระบบ

หน้าจอการทำงานของระบบ หรือส่วนที่ติดต่อกับผู้ใช้งาน มีดังต่อไปนี้

1. หน้าจอขึ้นทะเบียนลูกค้าใหม่ แสดงดังรูปที่ 5.1
2. หน้าจอแสดงรายการวัตถุดิบ แสดงดังรูปที่ 5.2 โดย Plastic Server จะทำการร้องขอข้อมูลรายการวัตถุดิบและราคาจากซัพพลายเออร์ต่าง ๆ
3. หน้าจอการสั่งซื้อวัตถุดิบ แสดงดังรูปที่ 5.3
4. หน้าจอกรอกข้อมูลในการจัดส่งสินค้าที่สั่งซื้อ แสดงดังรูปที่ 5.4
5. หน้าจอยืนยันการสั่งซื้อวัตถุดิบ แสดงดังรูปที่ 5.5 โดยที่ Plastic Server จะเก็บข้อมูลการสั่งซื้อ และส่งข้อมูลการสั่งซื้อ ไปให้กับซัพพลายเออร์อีกทอดหนึ่ง
6. หน้าจอแสดงรายการสั่งซื้อวัตถุดิบหลังยืนยัน แสดงดังรูปที่ 5.6 โดยทางฝั่งซัพพลายเออร์ได้รับใบสั่งซื้อแล้ว จะตอบรับการสั่งซื้อด้วยการส่ง วันที่ส่งสินค้ากลับมา
7. หน้าจอสอบถามการสั่งซื้อวัตถุดิบ แสดงดังรูปที่ 5.7 ซึ่งจะแสดงรายการที่สั่งซื้อ สถานะการสั่งซื้อ และที่อยู่ในการจัดส่ง

PlasticAndPolymer.com
Webservices for Shopping Plastic + Polymer

User Name : [None] Home Page

Menu	Customer Registry
	<div style="text-align: right;">Mode : <input type="text" value="A"/></div> <p>Initial : <input type="text" value="Miss"/> First Name : <input type="text" value="Muntana"/> *</p> <p>Last Name : <input type="text" value="Thanompongchat"/> *</p> <p>Address : <input type="text" value="55 Unicon build 6 th Floor Ladyao Chatuchak"/> *</p> <p>City : <input type="text" value="Bangkok"/> * Country : <input type="text" value="Thailand"/> *</p> <p>Zip Code : <input type="text" value="10900"/> * Telephone : <input type="text" value="0-2930-0286-7"/></p> <p>Email : <input type="text" value="kmuntana@yahoo.com"/> *</p> <p>User Name : <input type="text" value="MNTT"/> *</p> <p>Password : <input type="password" value="••••••"/></p> <p>Re-password : <input type="password" value="••••••"/></p> <p style="text-align: right;"><input type="button" value="Submit"/> <input type="button" value="Reset"/></p>

Copyright © 2004

Project Development Subject | Faculty of Information Technology, KMUTT

รูปที่ 5.1 หน้าจอขึ้นทะเบียนลูกค้าใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PlasticAndPolymer.com
Webservices for Shopping Plastic + Polymer

User Name : [MNTT] [Home Page](#)

Menu	Product Catalogue																																				
Product Catalogue Tracking Order Edit Profile About Us Contact Us Sign Out	Product Name : <input type="text" value="Hose"/> <input type="button" value="Search"/> <table border="1"> <thead> <tr> <th>Picture</th> <th>Material Description</th> <th>Price/Unit</th> <th>Quantity</th> <th>Source</th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td>Oil seals</td> <td>\$9.00</td> <td><input type="text" value="5"/> Kgs.</td> <td>Supplier 1</td> <td><input type="button" value="Add to Cart"/></td> </tr> <tr> <td></td> <td>Hose Delta PUR-L</td> <td>\$7.00</td> <td><input type="text" value="0.0"/> Lbs.</td> <td>Supplier 1</td> <td><input type="button" value="Add to Cart"/></td> </tr> <tr> <td></td> <td>O Ring measuring Equipment</td> <td>\$15.00</td> <td><input type="text" value="0.0"/> Kgs.</td> <td>Supplier 1</td> <td><input type="button" value="Add to Cart"/></td> </tr> <tr> <td></td> <td>Hose Delta PUR-L</td> <td>\$5.80</td> <td><input type="text" value="0.0"/> Lbs.</td> <td>Supplier 2</td> <td><input type="button" value="Add to Cart"/></td> </tr> <tr> <td></td> <td>Breaker welding supports</td> <td>\$6.40</td> <td><input type="text" value="0.0"/> Pcs.</td> <td>Supplier 2</td> <td><input type="button" value="Add to Cart"/></td> </tr> </tbody> </table>	Picture	Material Description	Price/Unit	Quantity	Source			Oil seals	\$9.00	<input type="text" value="5"/> Kgs.	Supplier 1	<input type="button" value="Add to Cart"/>		Hose Delta PUR-L	\$7.00	<input type="text" value="0.0"/> Lbs.	Supplier 1	<input type="button" value="Add to Cart"/>		O Ring measuring Equipment	\$15.00	<input type="text" value="0.0"/> Kgs.	Supplier 1	<input type="button" value="Add to Cart"/>		Hose Delta PUR-L	\$5.80	<input type="text" value="0.0"/> Lbs.	Supplier 2	<input type="button" value="Add to Cart"/>		Breaker welding supports	\$6.40	<input type="text" value="0.0"/> Pcs.	Supplier 2	<input type="button" value="Add to Cart"/>
Picture	Material Description	Price/Unit	Quantity	Source																																	
	Oil seals	\$9.00	<input type="text" value="5"/> Kgs.	Supplier 1	<input type="button" value="Add to Cart"/>																																
	Hose Delta PUR-L	\$7.00	<input type="text" value="0.0"/> Lbs.	Supplier 1	<input type="button" value="Add to Cart"/>																																
	O Ring measuring Equipment	\$15.00	<input type="text" value="0.0"/> Kgs.	Supplier 1	<input type="button" value="Add to Cart"/>																																
	Hose Delta PUR-L	\$5.80	<input type="text" value="0.0"/> Lbs.	Supplier 2	<input type="button" value="Add to Cart"/>																																
	Breaker welding supports	\$6.40	<input type="text" value="0.0"/> Pcs.	Supplier 2	<input type="button" value="Add to Cart"/>																																

รูปที่ 5.2 หน้าจอแสดงรายการวัตถุดิบ

PlasticAndPolymer.com
Webservices for Shopping Plastic + Polymer

User Name : [MNTT] [Home Page](#)

Menu	Material Order																												
Product Catalogue Tracking Order Edit Profile About Us Contact Us Sign Out	<ul style="list-style-type: none"> Step 1 : Enter the amount of material and Click Update to update the Grand Total. Step 2 : Click Check Out to proceed with your order. <table border="1"> <thead> <tr> <th>Picture</th> <th>Material Name</th> <th>Price/Unit</th> <th>Quantity</th> <th>Total</th> <th>Source</th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td>Oil seals</td> <td>\$9.00</td> <td><input type="text" value="5"/> Kgs.</td> <td>\$45.00</td> <td>Supplier 1</td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td></td> <td>Pyramid ranks</td> <td>\$3.00</td> <td><input type="text" value="7.0"/> Kgs.</td> <td>\$21.00</td> <td>Supplier 2</td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td></td> <td>Can mandrels</td> <td>\$8.40</td> <td><input type="text" value="5.0"/> Pcs.</td> <td>\$42.00</td> <td>Supplier 2</td> <td><input type="button" value="Delete"/></td> </tr> </tbody> </table> <p style="text-align: right;"><input type="button" value="Update"/> Grand Total \$108.00</p> <p style="text-align: center;"> <input type="button" value="Continue Shopping"/> <input type="button" value="Proceed to Checkout"/> </p>	Picture	Material Name	Price/Unit	Quantity	Total	Source			Oil seals	\$9.00	<input type="text" value="5"/> Kgs.	\$45.00	Supplier 1	<input type="button" value="Delete"/>		Pyramid ranks	\$3.00	<input type="text" value="7.0"/> Kgs.	\$21.00	Supplier 2	<input type="button" value="Delete"/>		Can mandrels	\$8.40	<input type="text" value="5.0"/> Pcs.	\$42.00	Supplier 2	<input type="button" value="Delete"/>
Picture	Material Name	Price/Unit	Quantity	Total	Source																								
	Oil seals	\$9.00	<input type="text" value="5"/> Kgs.	\$45.00	Supplier 1	<input type="button" value="Delete"/>																							
	Pyramid ranks	\$3.00	<input type="text" value="7.0"/> Kgs.	\$21.00	Supplier 2	<input type="button" value="Delete"/>																							
	Can mandrels	\$8.40	<input type="text" value="5.0"/> Pcs.	\$42.00	Supplier 2	<input type="button" value="Delete"/>																							

รูปที่ 5.3 หน้าจอการสั่งซื้อวัตถุดิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PlasticAndPolymer.com
Webservices for Shopping Plastic + Polymer

User Name : [MNTT] [Home Page](#)

Menu	Shipping Address
Product Catalogue Tracking Order Edit Profile About Us Contact Us Sign Out	<p>• Your order totals \$108.00. To complete your order, fill in the form and click Submit.</p> <p>Initial : <input type="text" value="Miss"/> First Name : <input type="text" value="Muntana"/> *</p> <p>Last Name : <input type="text" value="Thanompongchat"/> *</p> <p>Address : <input type="text" value="55 Unicorn build 6 th Floor Ladyao Chatuchak"/> *</p> <p>City : <input type="text" value="Bangkok"/> * Country : <input type="text" value="Thailand"/> *</p> <p>Zip Code : <input type="text" value="10900"/> * Telephone : <input type="text" value="0-2930-0286-7"/></p> <p>Email : <input type="text" value="kmuntana@yahoo.com"/> *</p> <p>Credit Card : <input type="text" value="VISA"/> * Card Number : <input type="text" value="1234-4567-8911-4567"/></p> <p style="text-align: right;"><input type="button" value="Submit"/> <input type="button" value="Reset"/></p>

รูปที่ 5.4 หน้าจกรอกข้อมูลในการจัดส่งสินค้า

PlasticAndPolymer.com
Webservices for Shopping Plastic + Polymer

User Name : [MNTT] [Home Page](#)

Menu	Confirm Order																														
Product Catalogue Tracking Order Edit Profile About Us Contact Us Sign Out	<p style="text-align: center;">Do you want to confirm order ?</p> <table border="1"> <thead> <tr> <th>No.</th> <th>Material Name</th> <th>Price/Unit</th> <th>Quantity</th> <th>Total</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Oil seals</td> <td>\$9.00</td> <td>5.0 Kgs.</td> <td>\$45.00</td> <td>Supplier 1</td> </tr> <tr> <td>2</td> <td>Pyramid ranks</td> <td>\$3.00</td> <td>7.0 Kgs.</td> <td>\$21.00</td> <td>Supplier 2</td> </tr> <tr> <td>3</td> <td>Can mandrels</td> <td>\$8.40</td> <td>5.0 Pcs.</td> <td>\$42.00</td> <td>Supplier 2</td> </tr> <tr> <td colspan="4" style="text-align: right;">Grand Total</td> <td>\$108.00</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Shipping Address</p> <p>Miss Muntana Thanompongchat 55 Unicorn build 6 th Floor Ladyao Chatuchak Bangkok 10900 Thailand</p> <p style="text-align: center;">Continue Shopping... <input type="button" value="Confirm Order"/></p>	No.	Material Name	Price/Unit	Quantity	Total	Source	1	Oil seals	\$9.00	5.0 Kgs.	\$45.00	Supplier 1	2	Pyramid ranks	\$3.00	7.0 Kgs.	\$21.00	Supplier 2	3	Can mandrels	\$8.40	5.0 Pcs.	\$42.00	Supplier 2	Grand Total				\$108.00	
No.	Material Name	Price/Unit	Quantity	Total	Source																										
1	Oil seals	\$9.00	5.0 Kgs.	\$45.00	Supplier 1																										
2	Pyramid ranks	\$3.00	7.0 Kgs.	\$21.00	Supplier 2																										
3	Can mandrels	\$8.40	5.0 Pcs.	\$42.00	Supplier 2																										
Grand Total				\$108.00																											

รูปที่ 5.5 หน้าจอยืนยันการสั่งซื้อวัตถุดิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PlasticAndPolymer.com
Webservices for Shopping Plastic + Polymer

User Name : [MNTT] Home Page

Menu	Response Order									
Product Catalogue Tracking Order Edit Profile About Us Contact Us Sign Out	<p style="text-align: center;">Your Order has been submitted successfully. Thank you for your interest in our products and services.</p> <table border="1"> <thead> <tr> <th>Order No.</th> <th>Ship Date</th> <th>Source</th> </tr> </thead> <tbody> <tr> <td>ORD47/482</td> <td>Friday, March 26, 2004</td> <td>Supplier 1</td> </tr> <tr> <td>ORDER47/300</td> <td>Sunday, March 28, 2004</td> <td>Supplier 2</td> </tr> </tbody> </table> <p style="text-align: center;">Continue Shopping...</p>	Order No.	Ship Date	Source	ORD47/482	Friday, March 26, 2004	Supplier 1	ORDER47/300	Sunday, March 28, 2004	Supplier 2
Order No.	Ship Date	Source								
ORD47/482	Friday, March 26, 2004	Supplier 1								
ORDER47/300	Sunday, March 28, 2004	Supplier 2								

รูปที่ 5.6 หน้าจอแสดงรายการสั่งซื้อวัตถุดิบหลังยืนยัน

PlasticAndPolymer.com
Webservices for Shopping Plastic + Polymer

User Name : [MNTT] Home Page

Menu	Tracking Order																																																		
Product Catalogue Tracking Order Edit Profile About Us Contact Us Sign Out	<table border="1"> <thead> <tr> <th>No.</th> <th>Material Name</th> <th>Price/Unit</th> <th>Quantity</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td colspan="5">Supplier 1 : ORD47/482</td> </tr> <tr> <td colspan="2">Date : Sunday, March 21, 2004</td> <td colspan="3">Miss Muntana Thanompongchat 55 Unicorn build 6 th Floor Ledyao Chetuchak Bangkok Thailand 10900</td> </tr> <tr> <td>1</td> <td>Oil seals</td> <td>\$9.00</td> <td>5.00 Kgs.</td> <td>\$45.00</td> </tr> <tr> <td colspan="4">Grand Total :</td> <td>\$45.00</td> </tr> <tr> <td colspan="5">Supplier 2 : ORDER47/300</td> </tr> <tr> <td colspan="2">Date : Sunday, March 21, 2004</td> <td colspan="3">Ship Date : Sunday, March 28, 2004</td> </tr> <tr> <td>1</td> <td>Pyramid ranks</td> <td>\$3.00</td> <td>7.00 Kgs.</td> <td>\$21.00</td> </tr> <tr> <td>2</td> <td>Can mandrels</td> <td>\$8.40</td> <td>5.00 Pcs.</td> <td>\$42.00</td> </tr> <tr> <td colspan="4">Grand Total :</td> <td>\$63.00</td> </tr> </tbody> </table>	No.	Material Name	Price/Unit	Quantity	Total	Supplier 1 : ORD47/482					Date : Sunday, March 21, 2004		Miss Muntana Thanompongchat 55 Unicorn build 6 th Floor Ledyao Chetuchak Bangkok Thailand 10900			1	Oil seals	\$9.00	5.00 Kgs.	\$45.00	Grand Total :				\$45.00	Supplier 2 : ORDER47/300					Date : Sunday, March 21, 2004		Ship Date : Sunday, March 28, 2004			1	Pyramid ranks	\$3.00	7.00 Kgs.	\$21.00	2	Can mandrels	\$8.40	5.00 Pcs.	\$42.00	Grand Total :				\$63.00
No.	Material Name	Price/Unit	Quantity	Total																																															
Supplier 1 : ORD47/482																																																			
Date : Sunday, March 21, 2004		Miss Muntana Thanompongchat 55 Unicorn build 6 th Floor Ledyao Chetuchak Bangkok Thailand 10900																																																	
1	Oil seals	\$9.00	5.00 Kgs.	\$45.00																																															
Grand Total :				\$45.00																																															
Supplier 2 : ORDER47/300																																																			
Date : Sunday, March 21, 2004		Ship Date : Sunday, March 28, 2004																																																	
1	Pyramid ranks	\$3.00	7.00 Kgs.	\$21.00																																															
2	Can mandrels	\$8.40	5.00 Pcs.	\$42.00																																															
Grand Total :				\$63.00																																															

รูปที่ 5.7 หน้าจอสอบถามการสั่งซื้อวัตถุดิบ

บทที่ 6

บทสรุป

ในบทนี้เป็นการสรุปภาพโดยรวมของการพัฒนาระบบ รวมทั้งข้อเสนอแนะในการพัฒนาระบบเพิ่มเติม โดยสรุปได้เป็นหัวข้อ ดังนี้

6.1 สรุปผลการออกแบบและพัฒนาระบบ

โครงการนี้ได้นำความรู้ที่ศึกษามาในเรื่องของ Java API for XML มาตรฐานเทคโนโลยีเว็บเซอร์วิส และทฤษฎีที่เกี่ยวข้อง มาประยุกต์ใช้กับเว็บแอปพลิเคชัน เพื่อให้การทำงานของเว็บแอปพลิเคชันมีประสิทธิภาพมากขึ้น โดยการยกตัวอย่างของการพัฒนาระบบการจัดซื้อโดยใช้ JAX เพื่อทดสอบว่าเทคโนโลยีเว็บเซอร์วิส และ JAX สามารถนำมาช่วยให้ระบบการพาณิชย์อิเล็กทรอนิกส์มีประสิทธิภาพมากขึ้น และรองรับการทำงานในหลากหลายรูปแบบ

ในระหว่างการวิเคราะห์และออกแบบระบบ ได้ศึกษาทฤษฎีการออกแบบและพัฒนาระบบเพิ่มเติม ได้แก่ Unified Modeling Language (UML) เพื่อนำมาใช้ในการออกแบบระบบ ในระหว่างการพัฒนา และทดสอบระบบมีการปรับปรุงโครงสร้าง และฟังก์ชันการทำงานของระบบ จนสามารถใช้งานได้ตามต้องการ

6.2 ประโยชน์ที่ได้รับจากการพัฒนาระบบ

ประโยชน์ที่ได้รับจากการพัฒนาระบบ สรุปได้ดังนี้

1. ได้ศึกษา Java API for XML และเทคโนโลยีเว็บเซอร์วิสต่าง ๆ แล้วนำมาประยุกต์ใช้กับเว็บแอปพลิเคชันการสั่งซื้อสินค้าผ่านอินเทอร์เน็ต เพื่อให้การทำงานของระบบการสั่งซื้อ มีประสิทธิภาพมากขึ้น
2. ได้ความรู้ในเรื่องของการวิเคราะห์และออกแบบระบบด้วย UML และนำไปใช้กับระบบงานอื่น ๆ ได้
3. ได้นำความรู้ที่ได้จากการศึกษาเทคโนโลยีเว็บเซอร์วิส ไปประยุกต์ใช้กับระบบงานอื่นๆ

6.3 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคระหว่างการพัฒนาระบบ สามารถสรุปได้ดังนี้

1. เนื่องจากเทคโนโลยีที่นำมาใช้ในโครงการนี้ เป็นเทคโนโลยีใหม่ ทำให้เสียเวลาในการศึกษาและทำความเข้าใจ
2. ปัญหาของอุปกรณ์คอมพิวเตอร์ และความเร็วในการติดต่อสื่อสารของอินเทอร์เน็ต ทำให้การทดสอบระบบจึงทำบนเครื่องคอมพิวเตอร์เครื่องเดียวกันเท่านั้น

6.4 ข้อเสนอแนะ

ในการพัฒนาระบบนี้ เป็นการจำลองการทำงาน เพื่อการศึกษาและทำความเข้าใจกับ Java API for XML และเทคโนโลยีเว็บเซอร์วิสที่เกี่ยวข้อง หากมีการพัฒนาต่อไปเพื่อให้สามารถนำไปใช้งานจริงได้อย่างสมบูรณ์ จำเป็นต้องปรับปรุง และพัฒนาเพิ่มเติม ดังต่อไปนี้

- ควรมีการชำระเงิน โดยใช้บัตรเครดิตและระบบสามารถตรวจสอบข้อมูลของบัตรเครดิตได้ ผ่านทางอินเทอร์เน็ต
- การยืนยันผลการสั่งซื้อสินค้า อาจมี E-mail ส่งไปให้กับลูกค้า เพื่อเพิ่มความน่าเชื่อถือมากขึ้น
- การคำนวณราคาของสินค้า ควรเพิ่มเรื่องภาษี ส่วนลด และสกุลเงิน เป็นต้น
- การรักษาความปลอดภัยของข้อมูล เพื่อให้ระบบมีความน่าเชื่อถือในการให้บริการลูกค้าได้มากขึ้น

บรรณานุกรม

- Gottschalk, Karl. 2001. **Web Services Architecture Overview**. [Online]. Available:
<http://www-106.ibm.com/developerworks/library/w-ovr/index.html>.
- IBM. 2002. **Web Services – The Web’s Next Revolution**. [Online]. Available:
<ftp://www6.software.ibm.com/software/developer/library/ws-wnxtrev.pdf>.
- Myerson, Judith M. 2001. **Web Services Architectures**. [Online]. Available:
<http://www.webservicesarchitect.com>.
- Orl, Ed and Mandava, Ramesh. 2002. **Java Web Services Developer Pack Part 1 : Registration and the JAXR API**. [Online]. Available:
<http://developer.java.sun.com/developer/technicalArticles/WebServices/WSPack/index.html>.
- Sommers, Frank. 2002. **Publish and Find UDDI Models with JAXR and WSDL**. [Online]. Available: http://www.javaworld.com/javaworld/jw-12-2002/jw-1213-webservices_p.html.
- Sun Microsystem. 2002. **The Java Web Services Tutorial**. [Online]. Available:
<http://java.sun.com/webservices/docs/1.0/tutorial/index.html>.
- W3C. 2000. **Simple Object Access Protocol (SOAP) 1.1**. [Online]. Available:
<http://www.w3.org/TR/SOAP/>.

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวมัณฑนา ถนอมพงษ์ชาติ
วันเกิด	16 มิถุนายน 2517
สถานที่เกิด	ลพบุรี
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (วิทยาการคอมพิวเตอร์) คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร
การทำงาน	Software Engineer บริษัท อิงเกรส (ประเทศไทย) จำกัด

