

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบจัดซื้อวัสดุประเภทพลาสติกโดยใช้เว็บเซอร์วิส
Plastic Material Ordering System Using Web Services



H002385



อาจารย์ที่ปรึกษา

ผศ. ดร. ภัทรชัย ลลิตโรจน์วงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2548

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาติให้นำไปเผยแพร่โดยไม่ขออนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส
Plastic Material Ordering System Using Web Services



ผศ. ดร. ภัทรชัย สถิตโรจน์วงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2548
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ระบบจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส
นักศึกษา	นายสุเมธ กาญจนภาส
อาจารย์ที่ปรึกษา	ผศ.ดร. ภัทรชัย สถิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

ในการจัดซื้อวัตถุดิบประเภทพลาสติก ผู้ผลิตจำเป็นต้องสั่งซื้อวัตถุดิบที่มีราคาต่ำ เพื่อลดต้นทุนในการผลิต และสามารถเพิ่มความรวดเร็วในการสั่งซื้อวัตถุดิบ จากผู้แทนจำหน่าย ดังนั้นจึงได้นำเว็บเซอร์วิสมาช่วยในการสั่งซื้อ และเพิ่มช่องทางในการแลกเปลี่ยนข้อมูลกัน ระหว่างผู้ผลิต และผู้แทนจำหน่าย ในการสั่งซื้อและจัดส่งวัตถุดิบ เนื่องจากในบางครั้งทั้งผู้ผลิต และผู้แทนจำหน่ายแต่ละราย ต่างมีระบบงานของตน ซึ่งต่างแพลตฟอร์มกัน เทคโนโลยีเว็บเซอร์วิสจึงถูกออกแบบมาเพื่อช่วยในการจัดการปัญหาดังกล่าว และอำนวยความสะดวกในการแลกเปลี่ยนข้อมูลกันได้

Title	Plastic Material Ordering System Using Web Services
Student	Mr. Sumate Kanjanophas
Advisor	Asst.Prof.Dr. Pattarachai Latitrojwong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2005

ABSTRACT

The purpose of the purchasing process is to identify the qualified suppliers that provide the best pricing goods consistent with quality, delivery and service. The purchasing process requires the cooperation of all suppliers to exchange data. However, each of them has its own information system running on the different platforms in heterogeneous environment. In the past few years, Web Service technology has been developed for interoperating between applications in distributed environment. In this project, we adopt the Web Service to solve the heterogeneity problem in data exchange

กิตติกรรมประกาศ

1. ขอขอบพระคุณ บิดา มารดา ที่ให้โอกาสในการศึกษาเล่าเรียนอย่างเต็มที่ รวมทั้งคอยให้กำลังใจช่วยเหลือ และให้คำปรึกษาต่าง ๆ
2. ขอขอบพระคุณ ผศ.ดร.ภัทรชัย ลลิตโรจน์วงศ์ อาจารย์ที่ปรึกษาเป็นอย่างมาก ที่ได้กรุณาให้คำปรึกษา คำแนะนำ และแก้ไขในสิ่งที่บกพร่องในการพัฒนาโครงการนี้
3. ขอขอบพระคุณ คณาจารย์ทุกท่านที่ให้ความรู้ต่าง ๆ เพื่อนำความรู้มาใช้ในการพัฒนาโครงการนี้
4. ขอขอบพระคุณ เจ้าหน้าที่ของคณะเทคโนโลยีสารสนเทศทุกคนที่ให้ความช่วยเหลือในด้านบริการต่าง ๆ เป็นอย่างดี

สุเมธ กาญจนภาค



สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่	
1. บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 ที่มาของปัญหา	1
1.3 วัตถุประสงค์ของโครงการ	2
1.4 ขอบเขตของระบบ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 รายละเอียดของแต่ละบท	3
2. ทฤษฎีเทคโนโลยีที่เกี่ยวข้อง	
2.1 เว็บเซอร์วิส	4
2.2 เครื่องมือที่ใช้พัฒนาเว็บเซอร์วิส	6
2.3 สถาปัตยกรรมและเอพีไอของจาวาสำหรั้บเอ็็กซ์เอ็มแอล	7
3. การวิเคราะห์และออกแบบระบบ	
3.1 ระบบการจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส	24
3.2 Supplier 1 Web Services และ Supplier 2 Web Services	26
3.3 การออกแบบระบบงานสั่งซื้อของโรงงานผลิตวัตถุดิบประเภทพลาสติก	27
3.4 การออกแบบระบบงานเว็บเซอร์วิสของ Plastic Supplier	37
3.5 การออกแบบเว็บเซอร์วิส	40

สารบัญ (ต่อ)

หน้า

4. การออกแบบฐานข้อมูล	
4.1 ฐานข้อมูลของซัพพลายเออร์รายที่ 1	41
4.2 ฐานข้อมูลของซัพพลายเออร์รายที่ 2	42
4.3 ฐานข้อมูลของโรงงานผลิตวัตถุดิบประเภท	45
5. การพัฒนาระบบ	
5.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ	48
5.2 หน้าจอการทำงานของระบบ	49
6. บทสรุป	
6.1 สรุปผลการออกแบบและพัฒนาระบบ	54
6.2 ประโยชน์ที่ได้รับจากการพัฒนาระบบ	54
6.3 ปัญหาและอุปสรรค	55
6.4 ข้อเสนอแนะ	55
บรรณานุกรม	56
ประวัติผู้เขียน	57

สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงรายละเอียดของตาราง CUSTOMER	43
4.2 แสดงรายละเอียดของตาราง ADDRESS	43
4.3 แสดงรายละเอียดของตาราง ORDER	44
4.4 แสดงรายละเอียดของตาราง ORDER_DETAIL	44
4.5 แสดงรายละเอียดของตาราง MATERIAL	45
4.6 แสดงรายละเอียดของตาราง CUSTOMER	46
4.7 แสดงรายละเอียดของตาราง ADDRESS	46
4.8 แสดงรายละเอียดของตาราง PURCHASE_ORDER	47
4.9 แสดงรายละเอียดของตาราง PURCHASE_ORDER_DETAIL	47

สารบัญรูป

รูปที่	หน้า
2.1 สถาปัตยกรรมเว็บเซอร์วิส	5
2.2 แผนภาพการแปลงของ XSLT	12
2.3 แผนภาพแสดงตัวอย่างการให้บริการโดยใช้แนวคิดของ JAX-RPC	22
3.1 ภาพรวมของระบบการจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส	25
3.2 ยูสเคสไดอะแกรมของ Plastic Server	28
3.3 คลาสไดอะแกรมของ Plastic Server	30
3.4 ซีเควนซ์ไดอะแกรมของ Search Supplier Web Service	32
3.5 ซีเควนซ์ไดอะแกรมของ Customer Register	33
3.6 ซีเควนซ์ไดอะแกรมของ Validate User	34
3.7 ซีเควนซ์ไดอะแกรมของ Request Material List	35
3.8 ซีเควนซ์ไดอะแกรมของ Purchase Order Material	36
3.9 ซีเควนซ์ไดอะแกรมของ Tracking Order	37
3.10 ยูสเคสไดอะแกรมของ Plastic Supplier	38
3.11 คลาสไดอะแกรมของ Plastic Supplier	39
3.12 ซีเควนซ์ไดอะแกรมของ Create Order Service	40
4.1 ER Diagram ของซัพพลายเออร์รายที่ 1	41
4.2 ER Diagram ของซัพพลายเออร์รายที่ 2	42
4.3 ER Diagram ของโรงงานผลิตวัตถุดิบประเภทพลาสติก	45
5.1 หน้าจอขึ้นทะเบียนสำหรับลูกค้าใหม่	50
5.2 หน้าจอแสดงรายการวัตถุดิบ	50
5.3 หน้าจอการสั่งซื้อวัตถุดิบ	51
5.4 หน้าจอกรอกข้อมูลในการจัดส่งสินค้า	51
5.5 หน้าจอยืนยันการสั่งซื้อวัตถุดิบ	52
5.6 หน้าจอแสดงรายการสั่งซื้อวัตถุดิบหลังยืนยัน	52
5.7 หน้าจอสอบถามการสั่งซื้อวัตถุดิบ	53

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

เนื่องจากโรงงานผลิตวัตถุดิบประเภทพลาสติก มีความต้องการสั่งซื้อวัตถุดิบจากซัพพลายเออร์รายต่างๆ เพื่อให้ได้วัตถุดิบตรงตามความต้องการมากที่สุด ซึ่งในระบบงานเดิมเมื่อโรงงานมีความต้องการสั่งซื้อวัตถุดิบ พนักงานจัดซื้อจะทำการเข้าไปค้นหาข้อมูลวัตถุดิบจากหลายๆ เว็บไซต์ เพื่อให้ได้ข้อมูลวัตถุดิบ และนำมาพิจารณาเลือกรายการวัตถุดิบที่ต้องการสั่งซื้อ เพื่อให้ได้วัตถุดิบที่มีคุณภาพ และตรงตามความต้องการ เช่น วัตถุดิบที่มีราคาต่ำ เพื่อช่วยลดต้นทุนในการผลิต เป็นต้น จึงเป็นการสิ้นเปลืองเวลาสำหรับการค้นหาข้อมูลวัตถุดิบจากซัพพลายเออร์รายต่างๆ ฉะนั้นจึงนำแนวคิดของเว็บเซอร์วิสมาใช้ในการพัฒนาให้บริการค้นหารายการสั่งซื้อที่ต้องการ และสั่งซื้อสินค้าได้ผ่านทางเว็บไซต์ สามารถเปรียบเทียบราคาสินค้าของซัพพลายเออร์ต่างๆ แล้วตัดสินใจสั่งซื้อได้เลย ตลอดจนมีการแลกเปลี่ยนข้อมูลกันระหว่างผู้ผลิตและซัพพลายเออร์ เพื่ออำนวยความสะดวกติดต่อสื่อสารกัน

เว็บเซอร์วิสช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอปพลิเคชันที่ต่างกัน เป็นไปโดยง่าย และสามารถถูกเรียกใช้ภายในองค์กรเอง หรือจากภายนอกองค์กร โดยผ่านไฟร์วอลล์ ดังนั้น จึงมีองค์กรขนาดใหญ่มากมาย กำลังพยายามพัฒนาระบบที่มีอยู่ของตน ให้เข้ากับเว็บเซอร์วิส ซึ่งนับเป็นการลงทุนที่คุ้มค่า เนื่องจากเว็บเซอร์วิสสามารถเพิ่มศักยภาพในการทำงานขององค์กร อีกทั้งลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กร นอกจากนี้ ยังสามารถใช้ร่วมกับเว็บแอปพลิเคชัน โดยส่งผ่านข้อมูลทางอินเทอร์เน็ต ซึ่งนับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้า หรือหุ้นส่วนทางธุรกิจ

1.2 ที่มาของปัญหา

1. การค้นหาเอกสารต่างๆ ทำให้ใช้เวลานาน ทำให้เสียเปรียบทางธุรกิจ
2. ไม่ทราบสถานะของเอกสารสั่งซื้อ
3. การค้นหาข้อมูลซัพพลายเออร์ต้องเข้าไปค้นหาจากหลายๆ เว็บไซต์
4. การแลกเปลี่ยนข้อมูลกันระหว่างผู้ผลิตและซัพพลายเออร์ เนื่องจากต่างมีระบบงานตนเองที่ต่างแพลตฟอร์มกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงเกิดแนวคิดที่จะพัฒนาระบบสารสนเทศสั่งซื้อโดยใช้เว็บเซอร์วิส เพื่อเพิ่มประสิทธิภาพในการทำงาน และสะดวกต่อการค้นหาข้อมูลสินค้าที่ต้องการสั่งซื้อ ตลอดจนใช้สำหรับแลกเปลี่ยนข้อมูลกัน

1.3 วัตถุประสงค์ของโครงการ

1. เพื่อค้นหาข้อมูลรายการสินค้า และวัตถุดิบที่พนักงานต้องการสั่งซื้อจากซัพพลายเออร์ต่างๆ และสามารถเปรียบเทียบราคาของสินค้าได้ เพื่อหาราคาสินค้าที่ถูกที่สุด และสามารถตัดสินใจสั่งซื้อได้เลย
2. เพื่ออำนวยความสะดวกในการค้นหาข้อมูลวัตถุดิบ และซัพพลายเออร์ให้ฝ่ายจัดซื้อ
3. เพื่อการตัดสินใจสั่งซื้อวัตถุดิบตรงตามความต้องการมากที่สุด
4. สามารถบอกสถานะของการสั่งซื้อ ตามความต้องการของผู้ใช้ได้ทันที
5. ช่วยลดเวลาในการจัดทำเอกสารสั่งซื้อ
6. เป็นตัวแทนในการสั่งซื้อสินค้าและวัตถุดิบ โดยติดต่อกับเว็บเซอร์วิสของซัพพลายเออร์ที่ต้องการ
7. สามารถแลกเปลี่ยนข้อมูลกันระหว่างผู้ผลิตและซัพพลายเออร์

1.4 ขอบเขตของระบบ

1. สามารถค้นหาข้อมูลสินค้าต่างๆ จากซัพพลายเออร์ในธุรกิจที่เกี่ยวข้องได้ และขอรายการราคาสินค้า จากซัพพลายเออร์ที่สนใจได้ โดยใช้เว็บเซอร์วิส
2. เมื่อได้เอกสารข้อมูลรายการราคาวัตถุดิบจากซัพพลายเออร์ต่างๆ แล้ว สามารถเปรียบเทียบได้ว่าสินค้าของซัพพลายเออร์รายไหน เป็นไปตามความต้องการของผู้ที่ต้องการสั่งซื้อ
3. สามารถตัดสินใจสั่งซื้อสินค้าได้จากการเปรียบเทียบราคาสินค้าโดยใช้เว็บเซอร์วิส
4. สามารถแลกเปลี่ยนข้อมูลกันระหว่างผู้ผลิตและซัพพลายเออร์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อช่วยอำนวยความสะดวกในการค้นหาข้อมูลวัตถุดิบ และซัพพลายเออร์ให้แก่ฝ่ายจัดซื้อ
2. ช่วยเปรียบเทียบราคาของวัตถุดิบตามความต้องการได้อย่างรวดเร็วและถูกต้อง
3. ช่วยในการตัดสินใจสั่งซื้อวัตถุดิบตรงตามความต้องการมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สามารถสอบถามสถานะของเอกสารและลดเวลาจัดทำเอกสารได้
5. สามารถแลกเปลี่ยนข้อมูลกันระหว่างผู้ผลิตและซัพพลายเออร์ ซึ่งใช้ระบบงานซึ่งต่างแพลตฟอร์มกันได้

1.6 รายละเอียดของแต่ละบท

- บทที่ 1 อธิบายถึงความเป็นมาของปัญหา ปัญหาของการระบบเดิม วัตถุประสงค์และขอบเขตของการพัฒนาระบบ และอธิบายรายละเอียดของแต่ละบท
- บทที่ 2 อธิบายถึงทฤษฎีและเทคโนโลยีที่เกี่ยวข้องกับการพัฒนาระบบ
- บทที่ 3 การวิเคราะห์ระบบและออกแบบ โดยใช้ยูสเคสไดอะแกรม คลาสไดอะแกรม ซีเควนซ์ไดอะแกรม
- บทที่ 4 การออกแบบฐานข้อมูล
- บทที่ 5 การพัฒนาระบบ
- บทที่ 6 บทสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 เว็บเซอร์วิส

2.1.1 ความหมายของเว็บเซอร์วิส

เว็บเซอร์วิสเป็นตัวกลางในการติดต่อระหว่างแอปพลิเคชัน ด้วยการตอบสนองคำร้องขอที่ส่งต่อมาจากระบบอื่น ทั้งจากระบบภายในและภายนอก (ผ่านทางอินเทอร์เน็ต) ด้วยเทคโนโลยีสื่อสารที่ใช้ง่าย และไม่ยึดติดกับผู้ขายรายใดรายหนึ่ง คุณสมบัติของเว็บเซอร์วิสมีรายละเอียดดังนี้ (Sun Microsystem. 2002)

1. รายละเอียดในการสร้าง และพัฒนาเว็บเซอร์วิสจะถูกซ่อนไว้ (Encapsulated) เพื่อไม่ให้มองเห็นได้จากภายนอก ผู้เรียกใช้เว็บเซอร์วิสจะรู้จักเพียงอินเทอร์เน็ตเฟสที่ผู้ให้บริการประกาศเอาไว้เท่านั้น กล่าวคือ เว็บเซอร์วิสจะเป็นประจักษ์ระบบงานภายในกับผู้ใช้จากภายนอก (คล้ายกับ XML ที่แยกตัวข้อมูล ออกจากส่วนที่ให้ความหมายว่าข้อมูลนั้นคืออะไร) ด้วยกรรมวิธีทางอ้อมเจตต์ อย่างเช่น การใช้ภาษา C++ ภาษาจาวา และคอมโพเนนต์ COM ดังนั้น จึงง่ายต่อการเรียกใช้งานเป็นอย่างยิ่ง

2. ซอฟต์แวร์ที่ทำงานบนระบบเว็บเซอร์วิส สามารถนำมาแก้ไขรายละเอียดภายในได้ โดยไม่ส่งผลกระทบต่อออกไปเป็นลูกโซ่ ทำให้การออกแบบซอฟต์แวร์เป็นไปโดยง่าย และผู้ใช้ที่ปลายทาง ก็ไม่จำเป็นต้องโหลดซอฟต์แวร์ติดตัวไว้มากเกินไปจนเกิดความจำเป็น

3. โปรแกรมที่เรียกใช้เว็บเซอร์วิสจะรับรู้ได้เองว่า เซอร์วิสที่กำลังจะเรียกใช้นั้นมีการกำหนดพารามิเตอร์อินพุต และเอาท์พุตอย่างไร

4. ด้วยความเป็นโพรโทคอลมาตรฐาน 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. เว็บเซอร์วิสสนับสนุนการค้นหา และเรียกใช้แบบไดนามิก (Dynamic Discovery and Invocation) ด้วยเทคโนโลยี UDDI แอปพลิเคชันจึงค้นหา และเรียกใช้เว็บเซอร์วิสได้ในขณะรันไทม์

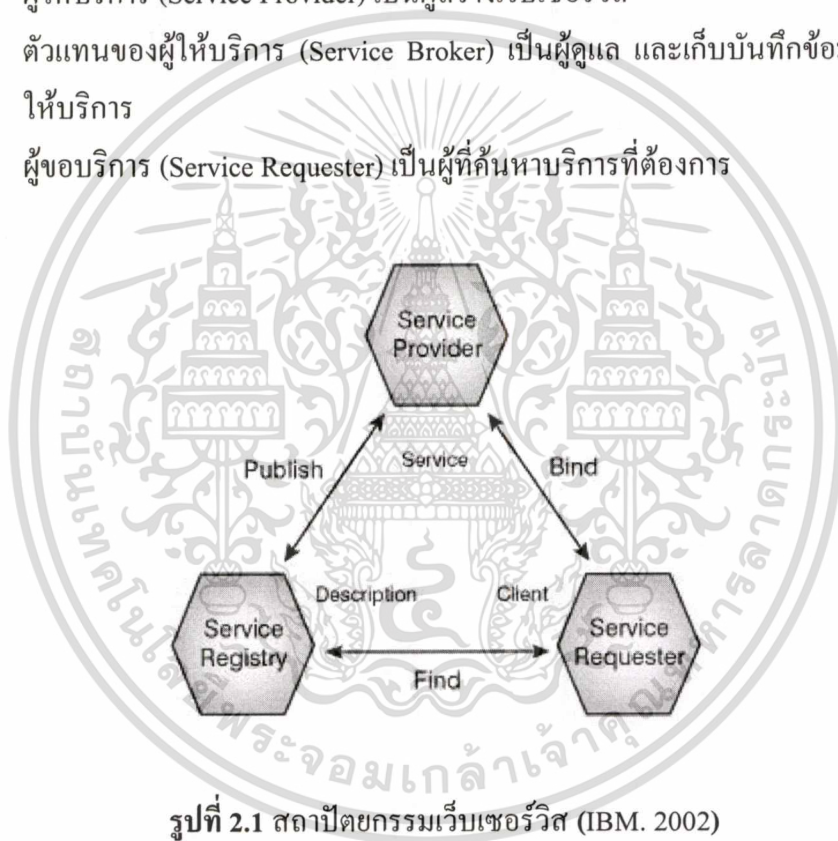
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเพิ่มความยืดหยุ่นให้กับนักพัฒนาซอฟต์แวร์ เนื่องจากไม่จำเป็นต้องกำหนดการเรียกใช้เว็บเซอร์วิสไว้ก่อน

2.1.2 สถาปัตยกรรมของเว็บเซอร์วิส

แนวความคิดของเว็บเซอร์วิสอาศัยหลักการของสถาปัตยกรรม การบริการในลักษณะของ Service-Oriented Architecture (SOA) ดังรูปที่ 2.1 ซึ่งประกอบด้วย 3 ส่วนดังนี้ (IBM. 2002)

1. ผู้ให้บริการ (Service Provider) เป็นผู้สร้างเว็บเซอร์วิส
2. ตัวแทนของผู้ให้บริการ (Service Broker) เป็นผู้ดูแล และเก็บบันทึกข้อมูลเกี่ยวกับการให้บริการ
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

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 นอกจากนี้ เทคโนโลยีเว็บเซอร์วิสได้ใช้ไวยากรณ์ของ XML เป็นพื้นฐานด้วย
2. SOAP (Simple Object Access Protocol) เป็นโพรโทคอลสำหรับสื่อสารระหว่างเว็บเซอร์วิส และแอปพลิเคชัน โดยระบบจะใช้ SOAP เพื่อให้แอปพลิเคชัน และเว็บเซอร์วิสทำงานแลกเปลี่ยนข้อมูลได้ ซึ่งเป็นทั้ง XML-based Protocol และ Messaging Protocol สำหรับแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจาย (Distributed Environment) จุดเด่นของ SOAP คือ เป็นโพรโทคอลที่เป็นกลาง สามารถทำงานร่วมกับโพรโทคอลอื่นๆ ได้หลายชนิด การพัฒนาก็อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการ ตามแบบจำลองทางวัตถุ และภาษาโปรแกรมที่ใช้พัฒนา
3. WSDL (Web Services Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของเว็บเซอร์วิส และวิธีการติดต่อขอรับบริการจากเว็บเซอร์วิส WSDL สร้างขึ้นโดยใช้ภาษา XML เวอร์ชันที่ใช้งานอยู่ปัจจุบันคือ WSDL 1.1
4. UDDI (Universal Description, Discovery, and Integration) เป็นวิธีการมาตรฐานสำหรับจัดเก็บ และรวบรวมบริการต่างๆ ที่ให้บริการในรูปแบบของไดเรกทอรีของบริการ หรือกล่าวได้ว่าเป็นมาตรฐานที่ให้ชุดพื้นฐาน API ของ SOAP สามารถนำมาใช้ในการพัฒนา เพื่อเป็นตัวแทนของผู้ให้บริการ โดย UDDI อาจเปรียบเสมือนฐานข้อมูลที่เก็บรายละเอียดของเว็บเซอร์วิสไว้ 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 สามารถสร้างเว็บเซอร์วิสโดยใช้ XML, Servlets, JSP (Java Server Pages), EJB (Enterprise Java Beans) และเทคโนโลยีของจาวา ซึ่งได้แก่ Java API for XML

2.3 สถาปัตยกรรมและเอพีไอของจาวาสำหรับเอ็กซ์เอ็มแอล

Java APIs สำหรับ XML ใช้กับเว็บแอปพลิเคชันที่เขียนด้วยภาษาจาวา API เหล่านี้แบ่งเป็น 2 ประเภทคือ ที่เกี่ยวกับเอกสาร XML และที่เกี่ยวกับกระบวนการ (Procedure) โดยมีรายละเอียดดังนี้ (Sun Microsystem. 2002)

- Document-oriented
 1. Java API for XML Processing (JAXP) ประมวลผลเอกสาร XML โดยใช้แปลเอกสาร (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 เรียกใช้ และรับปาร์ตี้ระยะไกล บนอินเทอร์เน็ต

ลักษณะสำคัญที่สุดของ Java API สำหรับ XML คือ ได้สนับสนุนมาตรฐานทางอุตสาหกรรม ดังเช่นเครือข่ายหลากหลายชนิดที่ปฏิบัติงานเป็นกลุ่มที่มีมาตรฐานร่วมกัน เช่น เวิลด์ไวด์เว็บ (WWW) และ the Organization for the Advancement of Structured Information Standards (OASIS) ได้กำหนดมาตรฐานของการทำสิ่งต่างๆ ทางธุรกิจที่ทำให้ข้อมูลทางธุรกิจ และแอปพลิเคชันทำงานร่วมกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะอีกอย่างหนึ่งของ Java API 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) ซึ่งเป็นมาตรฐานของพาร์เซอร์ ให้เลือกแปลเอกสารข้อมูลตามเหตุการณ์ หรือสร้างตัวแทนของอ็อบเจกต์ ใน JAXP เวอร์ชัน 1.1 ได้สนับสนุนมาตรฐานของ XSLT (XML Stylesheet Language Transformations) ให้สามารถควบคุมการนำเสนอข้อมูล และแปลงข้อมูลไปเป็นเอกสาร XML ตัวอื่น หรือรูปแบบอื่นได้ ดังเช่น HTML นอกจากนี้ JAXP จัดให้มีเนมสเปซ ที่ใช้ทำงานร่วมกับ DTD ซึ่งอาจจะมีการตั้งชื่อซ้ำซ้อนกันได้ ตัวอย่างของ JAXP ได้แก่

1. SAX API

SAX เป็น API ที่ทำงานโดยใช้เหตุการณ์เป็นหลัก 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 และทำอย่างนี้ต่อไปเรื่อยๆ

การอิมพลิเมนต์หรือการทำให้เกิดผล (Implementation) ประการแรกจะต้องสร้างอ็อบเจกต์ SAXParser จากอ็อบเจกต์ SAXParserFactory แล้วเรียกใช้เมธอด parse ส่งไฟล์ pricelist.xml และอินสแตนซ์ของคลาส Handler (เป็นการอิมพลิเมนต์ของเมธอด startElement และ Characters)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SAXParserFactory factory = SAXParserFactory.newInstance();
SAXParser saxParser = factory.newSAXParser();
SaxParser.parse("priceList.xml", handler);
```

SAX Parser จะเข้าไปอ่านไฟล์ pricelist.xml ที่ละบรรทัด เรียกใช้เมธอดตามความเหมาะสม เมื่อมีเมธอดที่อ้างถึงแล้ว พาร์เซอร์จะเรียกใช้เมธอดอื่นเช่น startDocument, endDocument, ignorableWhiteSpace และ processingInstructions

ตามที่ได้แสดงวิธีการอิมพลิเมนต์เมธอด 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 โดยสร้างจากอ็อบเจกต์ SAXParserFactory ที่มีการเซตเป็น validate ดังนี้

```
factory.setValidation(true);
```

ดังนั้น พาร์เซอร์จะรู้ว่าใช้ DTD ตัวไหนตรวจสอบเอกสาร XML โดยการประกาศที่ DOCTYPE ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2. DOM API

DOM API (Document Object Model API) ถูกกำหนดโดยกลุ่มทำงานของ W3C เป็นชุดของอินเทอร์เฟซสำหรับการสร้างตัวแทนของอ็อบเจกต์ในรูปแบบต้นไม้ 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 สามารถสร้างตัวแทนอ็อบเจกต์ของเอกสาร 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 และเพิ่มเอลิเมนต์ new coffee ด้วย name และ price ดังตัวอย่างข้างล่าง โค้ดนี้สร้างอ็อบเจกต์ DocumentBuilderFactory ขึ้นมา ซึ่งใช้สร้าง builder ที่เป็นอ็อบเจกต์ของ 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 ที่ชื่อ "Mocha Java" ขึ้นแรกให้ get รายการเอลิเมนต์ name และเข้าไปค้นหา "Mocha Java" การใช้งานอินเทอร์เฟซ Node ที่รวมอยู่ในแพ็คเกจ org.w3c.dom สร้างอ็อบเจกต์ Node ให้กับเอลิเมนต์ new coffee และโหนดใหม่สำหรับเอลิเมนต์ name และ price ดังนั้นสร้างอ็อบเจกต์ TextNode (TextNode) ให้กับเอลิเมนต์เหล่านี้ ดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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")) {
    // new node is insert before 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) วิธีเดียวกับ SAX Parser โดยเรียกใช้ `setValidating(true)`

3. XSLT API

XSLT (XSL Transformations) ถูกกำหนดโดยกลุ่มทำงานของ W3C เป็นภาษาที่กำหนดรายละเอียดการแปลงเอกสารต่างๆ 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

จากตัวอย่างโค้ดข้างล่างได้สร้างอ็อบเจกต์ Transformer (*transformer*) เพื่อแปลงรูปแบบดังนี้

```

TransformerFactory transFactory = TransformerFactory.newInstance();
Transformer transformer = transFactory.newTransformer();

```

ใช้รหัสโหนดของ DOM Tree สร้างอ็อบเจกต์ `DOMSource (source)` ของการแปลงรูปแบบดังนี้

```

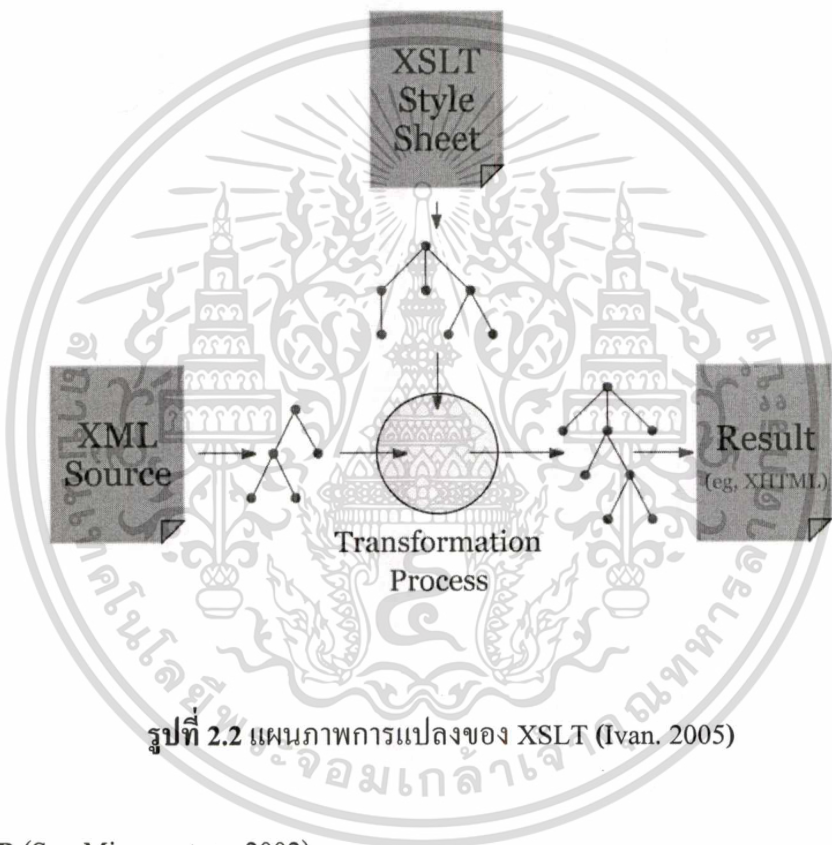
DomSource source = new DomSource(document);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามโค้ดข้างล่างสร้างอ็อบเจกต์ `StreamResult (result)` เพื่อเก็บผลลัพธ์ที่ได้ และแปลงทรี (`source`) ไปเป็นเอกสาร XML เก็บไว้ใน `result` ดังรูปที่ 2.2 (Ivan Herman. 2005)

```
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 (Ivan. 2005)

2.3.2 JAXB (Sun Microsystems. 2002)

JAXB จัดหาวิธีที่สะดวกรวดเร็วในการเชื่อมโยงระหว่างเอกสาร XML กับจาวาอ็อบเจกต์ โดยมีคอมไพเลอร์ของ JAXB ได้สร้างชุดของจาวาคลาสที่ประกอบด้วยโค้ดต่างๆ ที่วิเคราะห์เอกสาร XML ตามที่กำหนดในเค้าร่าง (schema) ผู้พัฒนาที่ใช้คลาสต่างๆ ที่สร้างขึ้นมา สามารถสร้างจาวาอ็อบเจกต์ทรี (Java Object Tree) แทนเอกสาร XML ได้โดยใช้โครงสร้างของต้นไม้ และสร้างเอกสาร XML กลับมาจากทรีได้

เมื่อเริ่มใช้งานแอปพลิเคชัน JAXB จำเป็นต้องมีเค้าร่าง ซึ่งเวอร์ชันปัจจุบันของ JAXB เป็น DTD อาจจะทำ DTD ขึ้นมาเองหรือเรียกใช้ผ่านทาง JAXR ก็ได้ การผูก DTD ไว้กับชุดของคลาสมีขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เขียนเค้าร่างซีกเหนือข (Binding Schema) ขึ้นมา ภายในมีคำสั่งวิธีการผูกเค้าร่าง กับคลาส เหล่านั้น เค้าร่างซีกเหนือขถูกเขียนด้วยภาษา XML-based Binding Language ซึ่งได้รวบรวมอยู่ใน JAXB

2. ทำการรันคอมไพเลอร์ของเค้าร่าง ซึ่งจะนำ DTD กับเค้าร่างซีกเหนือข และสร้าง คลาส ต่างๆ ขึ้นมา ในแต่ละคลาส คอมไพเลอร์จะสร้างเมธอด get และ set

เมื่ออินสแตนซ์ของคลาสถูกสร้างขึ้น และถูกกำหนดด้วยข้อมูล สามารถใช้เมธอด accessor เข้าถึงข้อมูลได้ (ชุดของเมธอด accessor เรียกว่าพร็อพเพอร์ตี้)

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 ไปเป็นเค้าร่างซีกเหนือข (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 และเค้าร่างซีกเหนือขนี้ คอมไพเลอร์ของเค้าร่าง ได้สร้างคลาส PriceList และ Coffee ขึ้นมา คลาส PriceList ได้รวบรวม constructor และ List property ที่มีเอลิเมนต์ coffee ผูกติด อยู่ คลาส Coffee ประกอบด้วย constructor และ property แทนชื่อของ coffee และ property แทน ราคา เมธอดต่างๆ ของ price มีดังนี้

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

ทั้ง PriceList และ Coffee ประกอบไปด้วยเมธอดดังต่อไปนี้

- Unmarshalling เป็น โพรเซสของการสร้างตัวแทนอ็อบเจกต์ของข้อมูล XML
- Validation เป็น โพรเซสของการตรวจสอบอ็อบเจกต์ต่างๆ ให้ตรงกับข้อกำหนดของ DTD
- Marshalling เป็น โพรเซสของการสร้างข้อมูล XML จากตัวแทนของอ็อบเจกต์

1. การสร้างตัวแทนอ็อบเจกต์ของข้อมูล XML

หลังจากสร้างคลาสขึ้นมาแล้ว เราสามารถเขียนจาวาแอปพลิเคชัน โดยใช้คลาสเหล่านั้น และสร้างตัวแทนอ็อบเจกต์ของเอกสาร XML ให้ถูกต้องตาม DTD อ็อบเจกต์แต่ละตัวที่มีลักษณะเดียวกับเอลิเมนต์ในเอกสาร XML เป็นอินสแตนซ์ของคลาสฯ หนึ่งจากชุดของคลาสที่ถูกสร้างขึ้นมดังกล่าว เนื่องจากอ็อบเจกต์ที่เชื่อมโยงทั้งเอกสาร และคลาสต่างๆ จะมีวิธีที่แตกต่างกันทั้ง 2 วิธีในการสร้างจาวาอ็อบเจกต์ทรีโดย Unmarshalling เอกสาร XML ให้ถูกต้อง หรืออินสแตนซ์ (สร้างตัวแทน) อ็อบเจกต์จากคลาส ด้วยวิธีนี้เอง JAXB ได้อนุญาตให้ประมวลผลได้ทั้งเอกสาร XML ที่มีอยู่แล้ว และสร้างข้อมูล 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 Unmarshalling กับเอกสาร XML นี้โดยสร้าง Input Stream จากเอกสาร XML นี้ และอ้างถึงเมธอด unmarshal ของคลาส PriceList ได้ดังนี้

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

ขณะนี้เรามีจาวาอ็อบเจกต์ทรีกับอ็อบเจกต์ myPrices ที่เป็นรูปของทรี สมมุติว่าต้องการสร้างเอกสารราคากาแฟในเอกสาร XML อันดับแรกต้องสร้างอ็อบเจกต์ทรี โดยการอินสแตนซ์แล้ว Marshalling ถ้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรีไปเป็นเอกสาร XML อันดับสอง สร้างอ็อบเจกต์ทรีโดยการอินสแตนซ์อ็อบเจกต์ PriceList และ get รายการของอ็อบเจกต์ Coffee จากอ็อบเจกต์ทรีนั้น และสร้างอ็อบเจกต์ Coffee (zCoffee) ขึ้นมา เซตค่า 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);
```

2. การเข้าถึงข้อมูลจากอ็อบเจกต์ทรี

สมมุติว่าต้องการที่จะเปลี่ยนราคาของ Mocha Java ในอ็อบเจกต์ทรีตัวแรกที่สร้างขึ้นมา จะต้องทำการค้นหา Mocha Java ในรายการของ coffee และกำหนดราคาใหม่โดยอ้างถึงเมธอด setPrice ในอ็อบเจกต์ Coffee ดังนี้

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

3. การเขียนเอกสาร XML จากอ็อบเจกต์ทรี

จากการที่ใช้ Unmarshalling สร้างตัวแทนอ็อบเจกต์จาก XML เราก็สามารถ Marshalling อ็อบเจกต์เหล่านั้นออกไปเป็นเอกสาร XML ได้ นั่นคือ JAXB อนุญาตให้คุณสร้างเอกสาร XML ใหม่ที่ถูกต้องตามข้อกำหนดในเค้าร่าง

นำอ็อบเจกต์ทรีที่ถูกดัดแปลงในรูปของเอกสาร XML โดยสร้างไฟล์ XML และ Output Stream ขึ้นมา แล้วเรียกใช้เมธอด marshal กับอ็อบเจกต์ myNewPrices ดังนี้

```
File newPrices = new File("newPriceList.xml");
FileOutputStream fos = new FileOutputStream(newPrices);
MyNewPrices.marshall(fos);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JAXB ได้จัดให้มีตัวเชื่อมโยงระหว่างเทคโนโลยี XML และ Java เข้าด้วยกัน เพียงแคมีเอกสาร XML อยู่ในรูปอินสแตนซ์ของโครงร่าง และจาวาอ็อบเจกต์ อยู่ในรูปอินสแตนซ์ของคลาส ดังนั้น เราสามารถสร้างจาวาอ็อบเจกต์ ที่มีหลักการเดียวกันกับข้อมูล XML ด้วย JAXB ซึ่งทำให้การสร้างแอปพลิเคชันที่ประมวลผลข้อมูล XML ง่ายขึ้น

4. ความแตกต่างระหว่าง JAXP และ JAXB

เราจำเป็นต้องพิจารณาเมื่อเลือกใช้สถาปัตยกรรมหรือ API ที่ประมวลผลข้อมูล XML จากข้อดีของ JAXB และ JAXP เพื่อตัดสินใจได้ถูกต้องดังนี้ (Sun Microsystem. 2002)

ใช้ JAXB เมื่อคุณต้องการ

- เข้าถึงข้อมูลในหน่วยความจำ แต่ไม่จำเป็นต้องใช้ความสามารถในการจัดการกับทรี
- ประมวลผลเฉพาะข้อมูลให้ถูกต้อง
- แลกเปลี่ยนข้อมูลที่มีชนิดแตกต่างกัน
- สร้างคลาสต่างๆ ตามที่กำหนดใน DTD
- สร้างตัวแทนอ็อบเจกต์ของข้อมูล XML

ใช้ JAXP เมื่อคุณต้องการ

- มีความยืดหยุ่นกับการพิจารณาวิธีการเข้าถึงข้อมูลแบบเป็นลำดับด้วย SAX หรือแบบสุ่มในหน่วยความจำด้วย DOM
- ใช้โค้ดที่ประมวลผลเหมือนกันกับเอกสารที่มี DTD แตกต่างกัน
- ประยุกต์ใช้ XSLT ในการแปลงรูปแบบ
- แทรกหรือลบอ็อบเจกต์ต่างๆ จากอ็อบเจกต์ทรีที่เป็นตัวแทนข้อมูล XML

2.3.3 JAXM (Sun Microsystem. 2002)

Java API for XML Messaging (JAXM) ได้จัดหาวิธีการมาตรฐานในการส่งเอกสาร XML บนอินเทอร์เน็ตจากแพลตฟอร์มที่เป็นจาวา โดยใช้ SOAP 1.1 และข้อกำหนดที่แนบมากับ SOAP และถูกขยายให้ทำงานกับโพรโทคอลสำหรับส่งข้อความในระดับสูง อย่างเช่น ebXML

ตัวอย่างของธุรกิจที่ใช้บริการผู้ให้บริการข้อความ (Messaging Provider) ซึ่งทำงานอยู่เบื้องหลังต้องการที่จะขนส่ง และหาเส้นทางของข้อความ เมื่อผู้ให้บริการข้อความถูกเรียกใช้งานข้อความทุกตัวของ JAXM ก็จะถูกเรียกใช้ด้วย โดยส่งข้อความแรกไปที่ผู้ให้บริการของผู้ส่ง และผู้ให้บริการของผู้รับ ถ้าไม่ถึงปลายทาง ผู้ให้บริการข้อความได้ใช้เทคโนโลยี JAXM บนไคลเอนต์โดยไม่ต้องรู้ว่าผู้ให้บริการทำงานอยู่เบื้องหลัง JAXM Client ทำให้มีการเรียกใช้เมธอด JAXM และเอกสาร Messaging Provider ร่วมกันกับ โครงสร้างพื้นฐานของข้อความไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความ JAXM ประกอบไปด้วย 2 ส่วน คือส่วนของ SOAP กับ Optional Attachment

1. ส่วนของ SOAP ประกอบด้วยอ็อบเจกต์ SOAPEnvelope ที่บรรจุอ็อบเจกต์ SOAPHeader และ SOAPBody ซึ่ง SOAPBody ในเอกสาร XML อยู่ในรูปเนื้อหาข้อความ (Message Content)
2. ส่วนของ Attachment อาจจะมีจำนวนเท่าใดก็ได้ หรืออาจจะไม่มีก็ได้ แต่ถ้ามีก็ไม่ใช่อเอกสาร XML และประกอบไปด้วยตัวอักษร หรือไฟล์รูปภาพ

1. การรับการเชื่อมต่อ

สิ่งแรกที่ JAXM Client จำเป็นต้องทำคือรับการเชื่อมต่อ (Connection) ทั้งอ็อบเจกต์ SOAPConnection และ ProviderConnection

1.1 การรับการเชื่อมต่อแบบจุดต่อจุด (Point-to-Point)

เครื่องไคลเอนต์ประเภท Standalone ถูกจำกัดการใช้อ็อบเจกต์ SOAPConnection ด้วยการติดต่อแบบจุดต่อจุด คือไปอย่างมีทิศทางจากผู้ส่งไปยังผู้รับ การติดต่อทั้งหมดของ JAXM ถูกสร้างโดย Connection Factory ในกรณีนี้อ็อบเจกต์ SOAPConnection ถูกสร้างจาก Factory ที่เป็นอ็อบเจกต์ของ SOAPConnectionFactory

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

1.2. การรับการเชื่อมต่อจากผู้ให้บริการข้อความ

การใช้ผู้ให้บริการข้อความ แอปพลิเคชันจะต้องมีอ็อบเจกต์ ProviderConnection สร้างการติดต่อจากผู้ให้บริการข้อความมี 2 วิธีดังนี้

วิธีที่ 1 รับอินสแตนซ์ของ ProviderConnectionFactory ที่ใช้สร้างการติดต่อตัวแปร pcCon เป็นการอิมพลิเมนต์การติดต่อของ JAXM Messaging Provider

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

วิธีที่ 2 สร้างอ็อบเจกต์ ProviderConnection ให้ดึงข้อมูลจากอ็อบเจกต์ ProviderConnectionFactory ที่ถูกอิมพลิเมนต์ เพื่อสร้างการติดต่อกับ Messaging

เอกสารนี้เป็นเอกสารที่ส Provider ตัวอย่างโค้ดข้างล่าง สาธิตการ get อ็อบเจกต์ ProviderConnectionFactory ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และใช้สร้างการติดต่อ ใน 2 บรรทัดแรกใช้ JNDI API ดึงข้อมูลอ็อบเจกต์ ProviderConnectionFactory จาก Naming Service ที่จดทะเบียนด้วยชื่อ "CoffeeBreakProvider" เมื่อถูกส่งผ่านไปในรูปแบบอาร์กิวเมนต์ (Argument) เมธอด lookup ได้ส่งค่าอ็อบเจกต์ ProviderConnectionFactory กลับไปที่ชื่อนั้นอยู่

```
Context ctx = getInitialContext();
ProviderConnectionFactory pcFactory =
    (ProviderConnectionFactory)ctx.lookup("CoffeeBreakProvider");
ProviderConnection con = pcFactory.createConnection();
```

อินสแตนซ์ของ ProviderConnection (con) เป็นการเชื่อมต่อของ Coffee Break

2. การสร้างข้อความ

การติดต่อกันจะมีข้อความที่ถูกสร้างโดย Factory อ็อบเจกต์ MessageFactory สามารถนำไปใช้ได้อยู่ 2 วิธี

วิธีที่ 1 รับค่าอินสแตนซ์ของคลาส MessageFactory โดยที่อินสแตนซ์นี้ถูกใช้สร้างอ็อบเจกต์ SOAPMessage

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

อ็อบเจกต์ SOAPMessage ทุกตัวที่ MessageFactory สร้างขึ้นมารวมทั้ง m จากบรรทัดบนจะเป็นข้อความของ SOAP ซึ่งข้อความเหล่านี้จะไม่ได้กำหนดไว้ก่อนใน Header

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

วิธีที่ 2 สร้างอ็อบเจกต์ของ MessageFactory จากเมธอด createMessageFactory ของ ProviderConnection และกำหนด profile ส่วนอ็อบเจกต์ SOAPMessage สร้างจากผลลัพธ์ของอ็อบเจกต์ MessageFactory ซึ่งจะสนับสนุน profile ที่กำหนด ดังตัวอย่างข้างล่าง schemaURI เป็น URI ของ schema สำหรับ profile ที่ต้องการ และ m2 จะสนับสนุน profile ของข้อความ (Messaging Profile) ที่ถูกสร้างจาก

เอกสารนี้เป็นเอกสารที่ `createMessageFactory` เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

3. การส่งข้อความ

เครื่องไคลเอนต์ประเภท Standalone ใช้เมธอด call ของ SOAPConnection ส่งข้อความ โดยอาร์กิวเมนต์ในเมธอด call ประกอบด้วยข้อความที่จะส่ง และอ็อบเจกต์ Endpoint ที่บรรจุ URL ของฝั่งรับ

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

แอปพลิเคชันที่ใช้เมธอด send ของ ProviderConnection ส่งข้อความที่ละข้อความ ซึ่งเมธอดนี้ส่งข้อความแบบอะซิงโครนัส นั่นคือ ส่งข้อความ และส่งค่ากลับในทันที เมธอดนี้ใช้พารามิเตอร์แค่ตัวเดียวในการส่งข้อความ

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

2.3.4 JAXR (Sun Microsystem. 2002)

Java API for XML Registries (JAXR) ได้จัดหาวิธีที่จะเข้าถึงมาตรฐานการจดทะเบียนทางธุรกิจบนอินเทอร์เน็ต การจดทะเบียนทางธุรกิจเปรียบเสมือนสมุดหน้าเหลืองที่บรรจุรายการต่างๆ ของธุรกิจ และผลิตภัณฑ์หรือบริการของธุรกิจนั้น JAXR ได้ให้ผู้พัฒนาเขียนแอปพลิเคชันในภาษาจาวาด้วยวิธีการที่ไม่ซ้ำแบบในการจดทะเบียนทางธุรกิจ ที่อยู่บนมาตรฐานเปิด (ดังเช่น ebXML) หรือข้อกำหนดของสมาคมอุตสาหกรรม (เช่น UDDI)

การจดทะเบียนเหล่านี้กลายเป็นองค์ประกอบที่สำคัญมากขึ้นของเว็บเซอร์วิส เพราะว่าการจดทะเบียนอนุญาตให้ธุรกิจต่างๆ ทำงานร่วมกันได้ อย่างไรก็ตาม ความจำเป็นของ JAXR ทำให้องค์กรเข้าถึงการจดทะเบียนที่เป็นมาตรฐานทางธุรกิจ จากภาษาจาวาที่กำลังก้าวหน้ามากขึ้น

1. การจดทะเบียนธุรกิจ

องค์กรที่ใช้แพลตฟอร์มจาวาสำหรับธุรกิจทางอิเล็กทรอนิกส์ จะใช้ JAXR รีจิสเตอร์ด้วยตัวเองในการจดทะเบียนที่เป็นมาตรฐาน โดยจะจัดหาชื่อและคำอธิบายด้วยตัวเอง และจัดแยกประเภท concepts ให้ง่ายต่อการค้นหา ดังตัวอย่างโค้ดข้างล่าง ในบรรทัดแรก สร้างอ็อบเจกต์ RegistryService (rs) และใช้ rs สร้างอ็อบเจกต์ BusinessLifeCycleManager (lcm) ธุรกิจกาแฟลูกโซ่ที่เรียกว่า The Coffee Break ได้ถูกแสดงโดยอ็อบเจกต์ Organization (org) ซึ่ง Coffee Break ได้เพิ่มชื่อและคำอธิบายด้วยตัวเอง และแบ่งแยกประเภทภายใน North American Industry Classification System (NAICS) ในขณะนี้ org บรรจุ Properties และ Classifications สำหรับ Coffee Break ที่ถูกเพิ่มให้กับอ็อบเจกต์ Collection (org) ในบรรทัดสุดท้าย orgs ถูก save โดย lcm ซึ่งจะจัดการกับวงจรชีวิตของอ็อบเจกต์ Organization ที่บรรจุอยู่ใน 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("ntls-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);
```

2. การค้นหาการจดทะเบียน

ธุรกิจหนึ่งๆ สามารถใช้ JAXR ค้นหาการจดทะเบียน (Registry) ของธุรกิจอื่นๆ ได้ดังตัวอย่างโค้ดข้างล่าง ใช้อ็อบเจกต์ 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 เป็นอ็อบเจกต์ Collection ทั้งหมด ซึ่งมี findQualifiers และ namePatterns ที่กำหนดที่นี้ และมีเฉพาะเอลิเมนต์ใน findQualifiers เป็นการกำหนดค่า String ที่ไม่มีการส่งค่ากลับ ถ้าไม่เช่นนั้น name เป็น case-sensitive match จับคู่กับส่วนหนึ่งของ name ในตัวแปร namePatterns ตัวแปรนี้เป็นของอ็อบเจกต์ Collection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้โดยไม่ได้รับอนุญาตเห็นว่าเป็นการละเมิดลิขสิทธิ์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีเพียงหนึ่งเอลิเมนต์ หรือเรียกในทางธุรกิจว่า “Coffee” อ็อบเจกต์ Collection ตัวอื่น คือ classifications ถูกกำหนดเมื่อ The Coffee Break ได้จัดทะเบียนด้วยตัวเอง (ดูตัวอย่างจากโค้ดข้างบน)

```
BusinessQueryManager bqm = rs.getBusinessQueryManager();
// define searching qualification
Collection findQualifiers = new ArrayList();
findQualifiers.add(FindQualifier.CASE_SENSITIVE_MATCH);
Collection namePatterns = new ArrayList();
// search name equals to 'Coffee'
namePatterns.add("%Coffee%");
// search by name and classifications
BulkResponse response = bqm.findOrganizations(findQualifiers, namePatterns, null,
        classifications, null, null);
Collection orgs = response.getCollection();
```

JAXR สนับสนุนการใช้คิวรีของ SQL ค้นหาการจดทะเบียน มีตัวอย่างการใช้อ็อบเจกต์ 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);
```

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

2.3.5 JAX-RPC (Sun Microsystem. 2002)

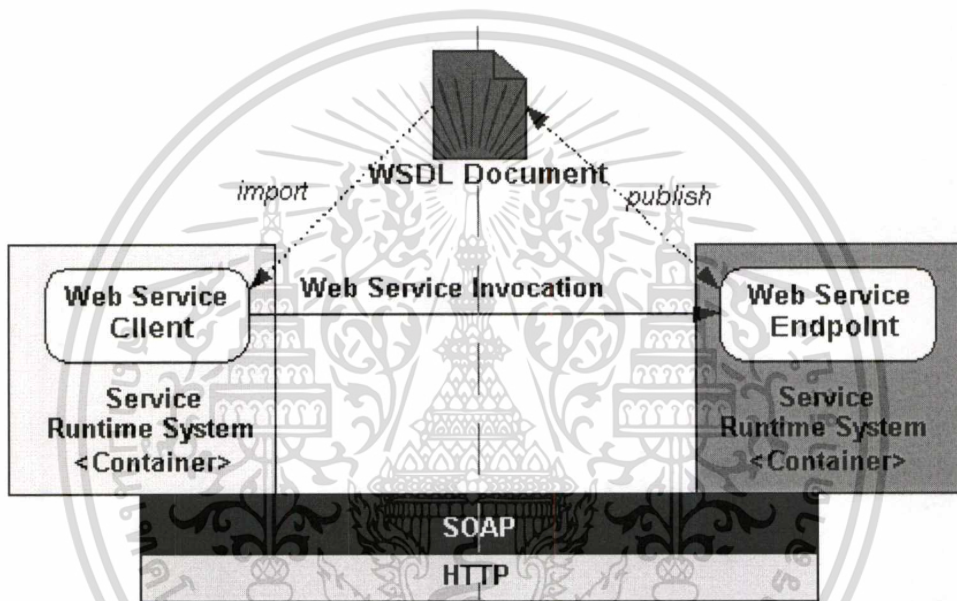
Java API for XML-based RPC (JAX-RPC) ทำให้การเขียนแอปพลิเคชันด้วยภาษาจาวา มีความเป็นไปได้ที่จะใช้ XML มีการเรียกใช้โพรซีเจอร์ระยะไกล (RPC)

ภาษาจาวามี API สำหรับสร้างการเรียกใช้โพรซีเจอร์ระยะไกลอยู่ 2 ตัวคือ Java IDL และ Remote Method Invocation (RMI) ความแตกต่างที่ JAX-RPC ขึ้นอยู่กับ XML และทำงานอยู่กับเว็บเซอร์วิส ส่วน JAVA IDL ขึ้นอยู่กับ CORBA (Common Object Request Broker Architecture) และใช้ภาษาการนิยามอินเทอร์เฟซของกลุ่มการจัดการอ็อบเจกต์ (OMG IDL) ส่วน RMI ขึ้นอยู่กับ RPC Sun จะสนับสนุน CORBA และ RMI ต่อไปเพื่อพัฒนา JAX-RPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ JAX-RPC

JAX-RPC ในแพลตฟอร์ม J2EE ทำให้ใช้เว็บเซอร์วิสง่ายขึ้น เว็บเซอร์วิสที่ใช้ RPC เป็นการรวมโพรซีเจอร์ที่ถูกเรียกใช้จากไคลเอนต์ระยะไกลบนอินเทอร์เน็ต บริการนี้อยู่ในแอปพลิเคชันฝั่งเซิร์ฟเวอร์ เช่น การสอบถามราคา โดยใช้เว็บเซอร์วิสแบบ RPC-base โดยไคลเอนต์จะร้องขอข้อมูลราคาที่ต้องการผ่านโพรโทคอล SOAP และเซิร์ฟเวอร์จะส่งราคากลับมาโดยใช้โพรโทคอล SOAP ดังรูปที่ 2.3 (Rahul. 2002)



รูปที่ 2.3 แผนภาพแสดงตัวอย่างการให้บริการโดยใช้แนวคิดของ JAX-RPC (Rahul. 2002)

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

ถึงแม้ว่า JAX-RPC อิมพลีเมนต์การเรียกใช้โพรซีเจอร์ระยะไกลเหมือนกับข้อความ SOAP ที่มีการร้องขอ-ตอบรับ แต่ JAXM เป็นรูปแบบที่แข็งแกร่งกว่าของข้อความ SOAP ซึ่งผู้พัฒนายอมรับกัน ในความเป็นจริงโพรโทคอลระดับที่สูงขึ้นไป เช่น ebXML ถูกกำหนดอยู่ชั้นบนสุดของ SOAP

รายการรวบรวมคุณลักษณะของ JAXM และ RPC ที่จัดทำให้ รวมทั้ง JAX-RPC ที่ไม่ได้จัดทำให้ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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



บทที่ 3

การวิเคราะห์และออกแบบระบบ

เนื่องจากโรงงานผลิตวัตถุดิบประเภทพลาสติก มีความต้องการสั่งซื้อวัตถุดิบจากซัพพลายเออร์รายต่างๆ เพื่อให้ได้วัตถุดิบตรงตามความต้องการมากที่สุด ซึ่งในระบบงานเดิมเมื่อโรงงานมีความต้องการสั่งซื้อวัตถุดิบ พนักงานจัดซื้อจะทำการเข้าไปค้นหาข้อมูลวัตถุดิบจากหลายๆ เว็บไซต์ เพื่อให้ได้ข้อมูลวัตถุดิบ และนำมาพิจารณาเลือกรายการวัตถุดิบที่ต้องการสั่งซื้อ เพื่อให้ได้วัตถุดิบที่มีคุณภาพ และตรงตามความต้องการ เช่น วัตถุดิบที่มีราคาต่ำ เพื่อช่วยลดต้นทุนในการผลิต เป็นต้น จึงเป็นการสิ้นเปลืองเวลาสำหรับการค้นหาข้อมูลวัตถุดิบจากซัพพลายเออร์รายต่างๆ ฉะนั้นจึงนำแนวคิดของเว็บเซอร์วิสมาใช้ในการพัฒนาให้บริการค้นหารายการสั่งซื้อที่ต้องการ และสั่งซื้อสินค้าได้ผ่านทางเว็บไซต์ สามารถเปรียบเทียบราคาสินค้าของซัพพลายเออร์ต่างๆ แล้วตัดสินใจสั่งซื้อได้เลย ตลอดจนมีการแลกเปลี่ยนข้อมูลกันระหว่างผู้ผลิต และซัพพลายเออร์ เพื่ออำนวยความสะดวกติดต่อสื่อสารกัน

3.1 ระบบการจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส

ระบบการจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส เป็นการนำเทคโนโลยีเว็บเซอร์วิสมาประยุกต์ใช้งานกับเว็บแอปพลิเคชันการสั่งซื้อสินค้าผ่านทางอินเทอร์เน็ต โดยมีระบบงานแบ่งออกเป็น 2 ส่วนหลักๆ ดังนี้

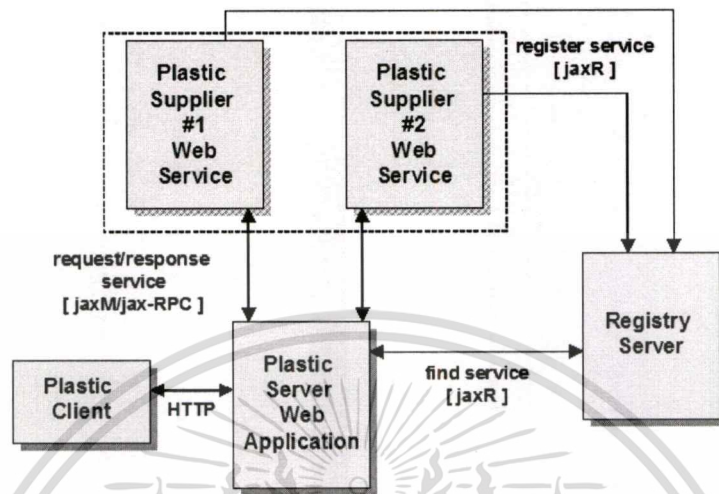
1. การสั่งซื้อสินค้าบนอินเทอร์เน็ต จะมีการติดต่อสื่อสาร โรงงานผลิตวัตถุดิบประเภทพลาสติก เพื่อสั่งซื้อสินค้าในลักษณะออนไลน์ โดยมีจุดประสงค์ดังนี้

- เพื่อค้นหาข้อมูลรายการสินค้าและวัตถุดิบ ที่พนักงานต้องการสั่งซื้อจากซัพพลายเออร์ และนำข้อมูลรายการสินค้าและวัตถุดิบที่ได้มาแสดงบนหน้าจอ เพื่อให้พนักงานสั่งซื้อทำการเปรียบเทียบ และเลือกสั่งซื้อสินค้าและวัตถุดิบจากซัพพลายเออร์ที่ต้องการ
- เป็นตัวแทนในการสั่งซื้อสินค้าและวัตถุดิบ โดยรับคำสั่งซื้อจากพนักงานสั่งซื้อแล้วส่งต่อไปให้กับเว็บเซอร์วิสของซัพพลายเออร์ที่ต้องการสั่งซื้อนั้นๆ

2. งานในระบบขายฝั่งซัพพลายเออร์ ในที่นี้จะกล่าวถึง การจัดทำรายการวัตถุดิบจากการร้องขอของผู้ซื้อ และจะยืนยันการสั่งซื้อกลับไปให้ผู้ซื้อ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การแจ้งในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ภาพรวมของระบบการจัดการข้อมูลประเภทพลาสติกโดยใช้เว็บเซอร์วิส

จากรูปที่ 3.1 การทำงานของระบบจัดการข้อมูลประเภทพลาสติกโดยใช้เว็บเซอร์วิส เริ่มจาก Plastic Client ร้องขอข้อมูลไปยัง Plastic Server Web Application ในรูปของเมสเสจ SOAP ส่งผ่านทางโพรโทคอล HTTP จากนั้น Plastic Server Web Application ใช้ JAX-RPC เพื่อร้องขอการบริการที่มีอยู่ใน Plastic Supplier Web Services อื่นๆ กรณีที่ไม่ทราบข้อมูลของ Service ของซัพพลายเออร์ สามารถค้นหาได้จาก Registry Server โดยใช้ JAXR ค้นหา Service ที่ต้องการ สำหรับ Plastic Supplier Web Services ที่ยังไม่ได้ลงทะเบียนกับ Registry Server ต้องลงทะเบียนบริการนั้นก่อนโดยใช้ JAXR

การทำงานของ Plastic Server Web Application มีรายละเอียดดังนี้

- มีการร้องขอเมธอดเพื่อขอใช้บริการจากซัพพลายเออร์อื่นๆ โดยใช้ JAX-RPC
- มีการส่งเมสเสจไปยัง Plastic Supplier Web Services ได้แก่ คิวรีของรายการราคาวัตถุดิบ และรายการสั่งซื้อ
- Plastic Supplier Web Services มีการตอบสนองด้วยเมสเสจข้อมูลวัตถุดิบ ราคาของวัตถุดิบ และการยืนยันการสั่งซื้อ
- Plastic Server Web Application มีการตอบสนองข้อมูลรายการราคาพลาสติก เพื่อนำรายการราคาพลาสติกที่ได้รับจากซัพพลายเออร์ มาแสดงให้ผู้ซื้อได้เปรียบเทียบราคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Plastic Server Web Application ใช้ Servlet ที่ทำงานบน Tomcat และ JavaBean Component ในการเก็บข้อมูลต่างๆ ที่ได้รับมาจาก Plastic Supplier Web Service และการคำนวณราคาต่างๆ เช่น การคำนวณราคาสินค้าที่ต้องการขายให้กับผู้ซื้อ เป็นต้น
- Plastic Server Web Application และ Plastic Supplier Web Services ต้องมี SOAP Listener ไว้สำหรับทำหน้าที่คอยฟังการเรียกใช้ และโปรแกรมที่ทำหน้าที่ติดต่อกับโพรโทคอล SOAP เพื่อใช้ในการสร้างและอ่านเมสเสจ SOAP ซึ่งส่วนใหญ่แล้วโปรแกรมนี้จะมี XML Parser เพื่อแปลงข้อมูลให้อยู่ในรูปที่เข้าใจได้ง่าย และนำไปใช้งานได้

จากการทำงานข้างต้นสามารถนำมาออกแบบการทำงานได้เป็น 3 ระบบ ดังนี้

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

3.2 Supplier 1 Web Services และ Supplier 2 Web Services

ระบบเว็บเซอร์วิสของซัพพลายเออร์รายที่ 1 และรายที่ 2 เป็นเว็บเซอร์วิสที่เป็นผู้ให้บริการ โดยจะรอรับการร้องขอ และส่งผลการตอบรับกลับไปยังผู้ร้องขอ (Plastic Server) โดยใช้เทคโนโลยี JAX-RPC ซึ่งมีการให้บริการดังนี้

- การค้นหาข้อมูลวัตถุดิบ
- การสั่งซื้อวัตถุดิบ
- การตรวจสอบสถานะของการจัดส่งวัตถุดิบ

การนำระบบดังกล่าวไปใช้งาน จำเป็นต้องมีการติดตั้งสภาพแวดล้อมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Supplier 1 Web Services ติดตั้งอยู่บนระบบปฏิบัติการ Windows XP และระบบฐานข้อมูล Postgres
- Supplier 2 Web Service ติดตั้งอยู่บนระบบปฏิบัติการ Windows XP และระบบฐานข้อมูล MySQL

3.3 การออกแบบระบบงานสั่งซื้อของโรงงานผลิตวัตถุดิบประเภทพลาสติก

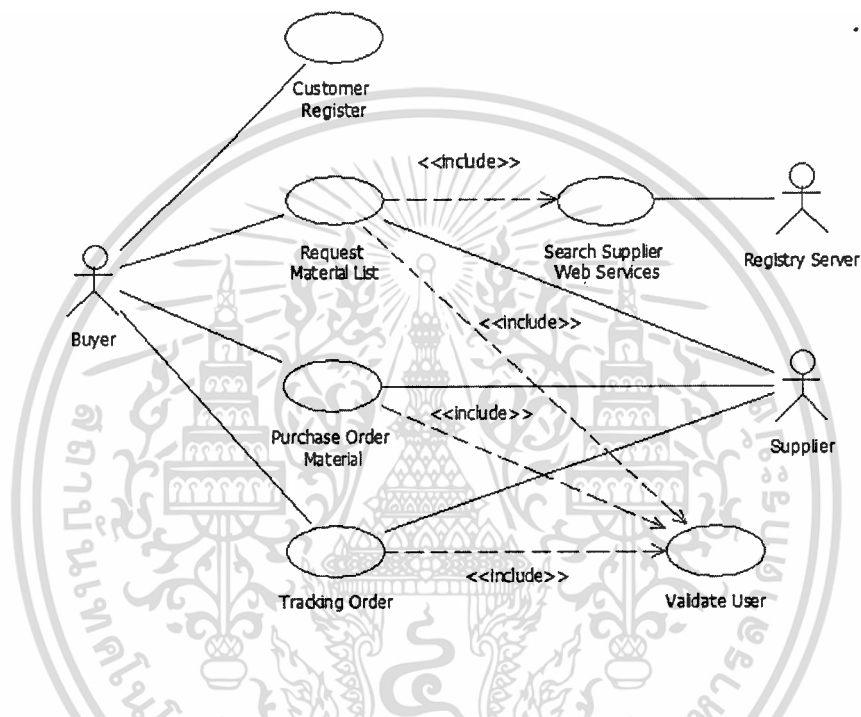
ระบบงานสั่งซื้อของโรงงานผลิตวัตถุดิบประเภทพลาสติก เป็นเว็บแอปพลิเคชัน ที่ทำหน้าที่เรียกใช้งานเว็บเซอร์วิสของซัพพลายเออร์ต่างๆ โดยใช้ Java Servlets , JSP และ JavaBeans ซึ่งติดตั้งอยู่บนระบบปฏิบัติการ Windows XP และใช้ระบบฐานข้อมูล Postgres

ขั้นตอนการทำงานของ Plastic Server มีดังนี้

1. Plastic Server จะใช้เทคโนโลยี JAXR ในการค้นหาข้อมูลการให้บริการ ของซัพพลายเออร์ ที่ลงทะเบียนไว้กับ Registry Server
2. เมื่อพบรายชื่อและข้อมูลต่างๆของซัพพลายเออร์ทั้ง 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

3.3.1 ยูสเคสไดอะแกรม

การวิเคราะห์และออกแบบระบบ จะเริ่มจากการเก็บรวบรวมความต้องการต่างๆ ของระบบ ทำให้ทราบถึงฟังก์ชันการทำงานของระบบ และแอกเตอร์ที่เกี่ยวข้องกับระบบ ที่จะนำมาใช้เขียนเป็นยูสเคสไดอะแกรม ดังรูปที่ 3.2



รูปที่ 3.2 ยูสเคสไดอะแกรมของ Plastic Server

แอกเตอร์ที่เกี่ยวข้องกับระบบ ได้แก่

- Buyer เป็นพนักงานจัดซื้อที่ทำหน้าที่จัดซื้อสินค้าที่ต้องการเข้าสู่โรงงาน
- Registry Server เป็นผู้ให้บริการค้นหาและลงทะเบียนบริการเว็บเซอร์วิสในเครือข่าย
- Supplier ผู้จัดจำหน่ายวัตถุดิบให้กับโรงงาน

ยูสเคสที่เกี่ยวข้องกับระบบ ได้แก่

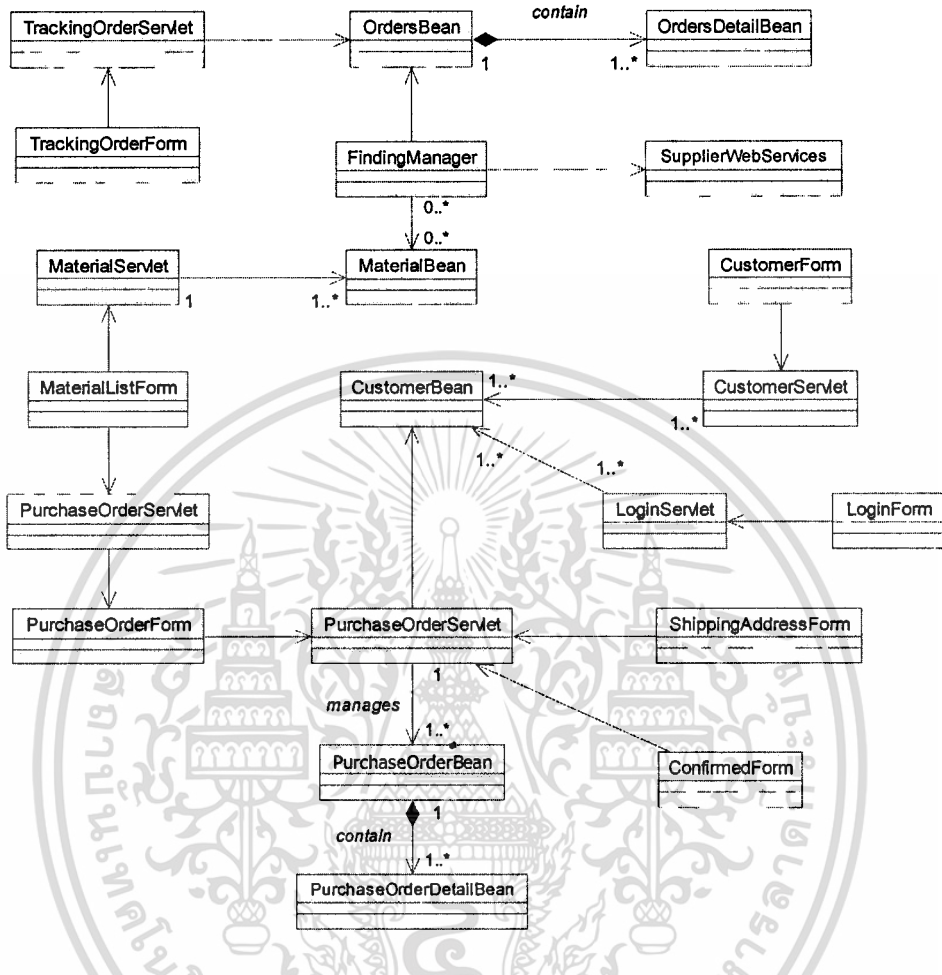
- Customer Register เป็นการให้ลูกค้า หรือผู้สั่งซื้อลงทะเบียนกับระบบในฝั่ง Plastic Server โดยโปรแกรมจะเก็บข้อมูลต่างๆ ของลูกค้าลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Request Material List พนักงานจัดซื้อร้องขอข้อมูลรายการวัสดุที่ต้องการ ไปยังซัพพลายเออร์ต่างๆ ผ่านเว็บเซอร์วิส ซึ่งเมื่อได้รับข้อมูลจากซัพพลายเออร์ โปรแกรมจะทำการประมวลผลรายการราคาวัสดุ และจัดเรียงรายการวัสดุ ให้พนักงานจัดซื้อเปรียบเทียบราคาได้ง่ายขึ้น
- Search Supplier Web Services ใช้สำหรับค้นหาซัพพลายเออร์ ที่ต้องการจาก Registry Server ซึ่งจะได้ข้อมูลผู้ให้บริการจำหน่ายสินค้าที่ต้องการจะสั่งซื้อ
- Purchase Order Material พนักงานจัดซื้อจะกำหนดรายการสั่งซื้อสินค้าต่างๆ เพื่อจัดซื้อเข้าบริษัท โดยเลือกรายการสินค้าและวัสดุได้ตามความต้องการ พร้อมกับระบุจำนวนสินค้า และป้อนที่อยู่สำหรับการจัดส่งสินค้า เมื่อทำการเก็บข้อมูลที่สั่งซื้อแล้ว โปรแกรมจะแสดงหน้าจอ ให้ผู้ใช้ยืนยันการสั่งซื้อสินค้า หลังจากนั้นจะจัดส่งรายการสั่งซื้อดังกล่าว ให้กับซัพพลายเออร์
- Purchase Tracking ใช้สำหรับตรวจสอบสถานะของการสั่งซื้อวัสดุของลูกค้าว่าดำเนินการถึงขั้นตอนนี้แล้ว โดยที่โปรแกรมจะตรวจสอบว่า ลูกค้าเคยลงทะเบียนมาก่อนไหม แล้วระบบจะเรียกใช้เว็บเซอร์วิสของซัพพลายเออร์ และแจ้งสถานะของการสั่งซื้อให้ทราบผ่านทางเบรเซอร์
- Validate User ตรวจสอบรหัสผ่านในการเข้าใช้บริการสั่งซื้อสินค้าผ่านเว็บเซอร์วิส

3.3.2 คลาสไดอะแกรม

เมื่อผ่านขั้นตอนวิเคราะห์เข้าสู่ขั้นตอนออกแบบ ทำให้สามารถออกแบบแบบคลาสไดอะแกรมสำหรับอธิบายรายละเอียดของคลาสภายในโปรแกรมที่ต้องสร้างขึ้น และความสัมพันธ์ต่างๆ ระหว่างคลาส โดยคลาสใช้สัญลักษณ์เป็นสี่เหลี่ยมผืนผ้า ภายในแบ่งเป็น ชื่อคลาส คุณสมบัติ และเมธอดเรียงจากบนลงล่าง ดังรูปที่ 3.3



รูปที่ 3.3 คลาสไดอะแกรมของ Plastic Server

คลาสภายในโปรแกรมที่ต้องสร้างขึ้น ได้แก่

1. FindingManager เป็นคลาสสำหรับควบคุมการค้นหา ข้อมูล URL และที่อยู่สำหรับการติดต่อขอบริการจาก Supplier Web Services
2. SupplierWebservices เป็นคลาสสำหรับจัดเก็บ ข้อมูล URL ของผู้จัดจำหน่ายวัตถุดิบ และสินค้าต่างๆ ที่ต้องการสั่งซื้อ
3. MaterialBean เป็นคลาสสำหรับจัดเก็บ ข้อมูลวัตถุดิบและสินค้าต่างๆ ที่ได้จากการค้นหาจากซัพพลายเออร์
4. MaterialServlet เป็นคลาสสำหรับควบคุม ข้อมูลวัตถุดิบและสินค้าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. MaterialListForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอค้นหาและเลือกรายการวัตถุดิบต่างๆ
6. PurchaseOrderBean เป็นคลาสสำหรับจัดเก็บ ข้อมูลรายการสั่งซื้อส่วนหลัก
7. PurchaseOrderItemBean เป็นคลาสสำหรับจัดเก็บ ข้อมูลรายการสั่งซื้อส่วนรายละเอียด
8. PurchaseOrderServlet เป็นคลาสสำหรับควบคุม การสั่งซื้อรายการวัตถุดิบต่างๆ
9. PurchaseOrderForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอบันทึกรายการสั่งซื้อวัตถุดิบต่างๆ
10. ShippingAddressForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอแสดงข้อมูลสถานที่จัดส่งและรายละเอียดการชำระเงิน
11. ConfirmedForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอยืนยันการสั่งซื้อวัตถุดิบต่างๆ
12. OrdersBean เป็นคลาสสำหรับจัดเก็บ ข้อมูลรายการสั่งซื้อส่วนหลัก จากซัพพลายเออร์
13. OrdersDetailBean เป็นคลาสสำหรับจัดเก็บ ข้อมูลรายการสั่งซื้อส่วนรายละเอียด จากซัพพลายเออร์
14. TrackingOrderServlet เป็นคลาสสำหรับควบคุม การสั่งซื้อรายการวัตถุดิบต่างๆ จากซัพพลายเออร์
15. TrackingOrderForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอสอบถามสถานะของการสั่งซื้อวัตถุดิบต่างๆ จากซัพพลายเออร์
16. CustomerBean เป็นคลาสสำหรับจัดเก็บ ข้อมูลลูกค้า
17. CustomerServlet เป็นคลาสสำหรับควบคุม การลงทะเบียนลูกค้าใหม่ เพื่อใช้กำหนดชื่อผู้ใช้ระบบและรหัสผ่าน ในการสั่งซื้อสินค้าจากระบบ
18. CustomerForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอลงทะเบียนลูกค้าใหม่
19. LoginServlet เป็นคลาสสำหรับควบคุม การเข้าใช้งานระบบ
20. LoginForm เป็นคลาสอินเทอร์เฟซสำหรับจัดการกับ หน้าจอตรวจสอบการเข้าใช้งานระบบ

3.3.3 ซีเควนซ์ไคอะแกรม

เมื่อผ่านขั้นตอนการออกแบบและมีความต้องการอธิบายความสัมพันธ์ระหว่างยูสเคส และคลาสไคอะแกรม ทำให้สามารถออกแบบซีเควนซ์ไคอะแกรม โดยใช้สัญลักษณ์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

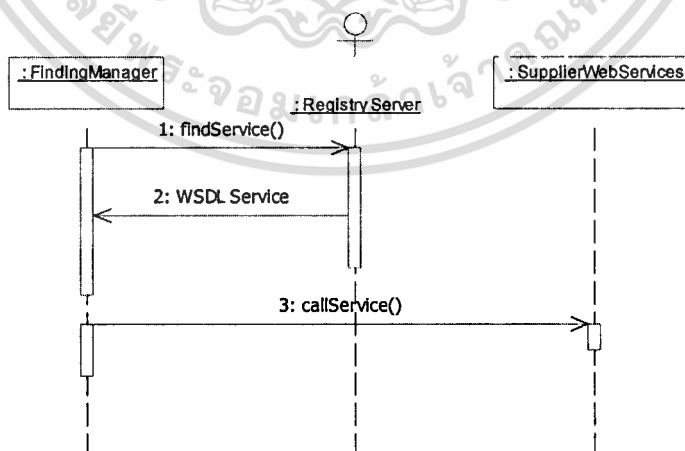
- สีเหลี่ยมผืนผ้าใช้แทนคลาสหรืออ็อบเจกต์ ภายในสีเหลี่ยมจะมีชื่อคลาสหรืออ็อบเจกต์
- เส้นประแนวตั้งใช้แสดงลำดับเวลาจากบนลงล่าง
- เส้นตรงแนวนอนมีหัวลูกศร แสดงกิจกรรมที่เกิดขึ้นจากคลาส หรืออ็อบเจกต์ในไคอะแกรม
- สีเหลี่ยมใสที่วางบนเส้นแสดงเวลา หมายถึงช่วงเวลาให้อ็อบเจกต์ยัง active อยู่

ในการพัฒนาระบบจัดซื้อวัตถุดิบประเภทพลาสติก มีด้วยซีควენซ์ไคอะแกรม ดังนี้

3.3.3.1 ซีควেনซ์ไคอะแกรมของ Search Supplier Web Services แสดงดังรูปที่ 3.4

อธิบายการทำงาน ได้ดังนี้

1. Registry Server ให้บริการค้นหาข้อมูลการบริการของซัพพลายเออร์ เคมมาลงทะเบียน และประกาศการให้บริการไว้
2. Plastic Server ใช้ FindingManager เพื่อทำการค้นหาบริการเกี่ยวกับสินค้าที่ต้องการจาก Register Server
3. ผลลัพธ์ที่ได้ถูกอธิบายในรูปแบบของ WSDL และส่งกลับมายัง FindingManager
4. Plastic Server ใช้ FindingManager ติดต่อไปยังซัพพลายเออร์ เพื่อขอใช้บริการเกี่ยวกับสินค้านั้นโดยตรง



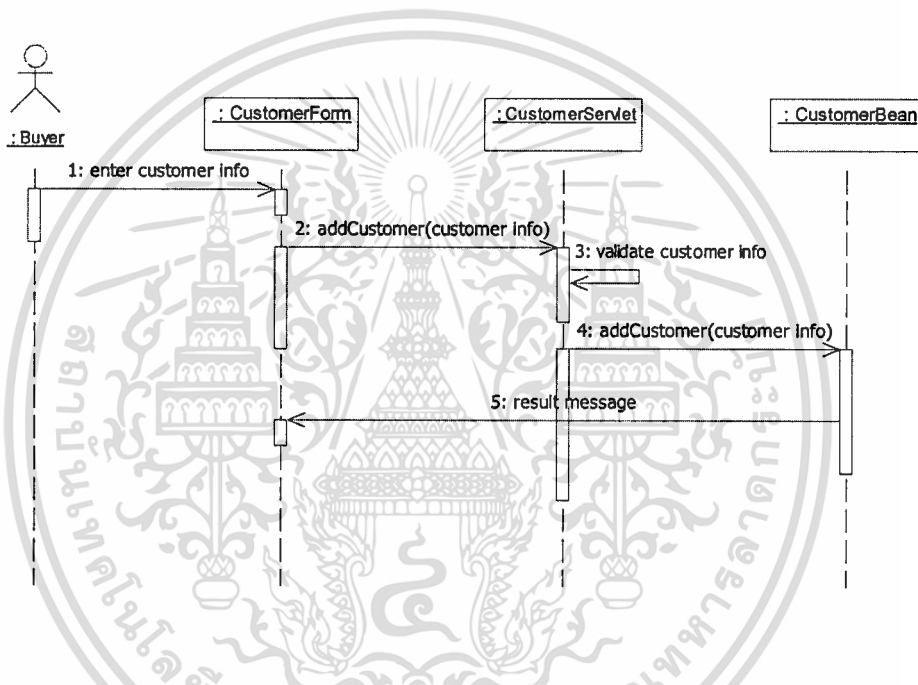
รูปที่ 3.4 ซีควเอนซ์ไคอะแกรมของ Search Supplier Web Services

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.2 ซีควেনซ์ไดอะแกรมของ Customer Register แสดงดังรูปที่ 3.5

อธิบายการทำงานได้ดังนี้

1. ผู้ซื้อป้อนข้อมูลชื่อและที่อยู่ สำหรับการลงทะเบียนเป็นสมาชิก เพื่อเข้าใช้ระบบงาน
2. CustomerServlet จะทำการรับข้อมูลการลงทะเบียน และนำไปตรวจสอบว่าถูกต้องหรือไม่ ก่อนทำการบันทึกข้อมูลลงฐานข้อมูลของลูกค้า
3. เมื่อลงทะเบียนเรียบร้อยแล้ว จะแสดงข้อความให้ผู้ใช้งานทราบ

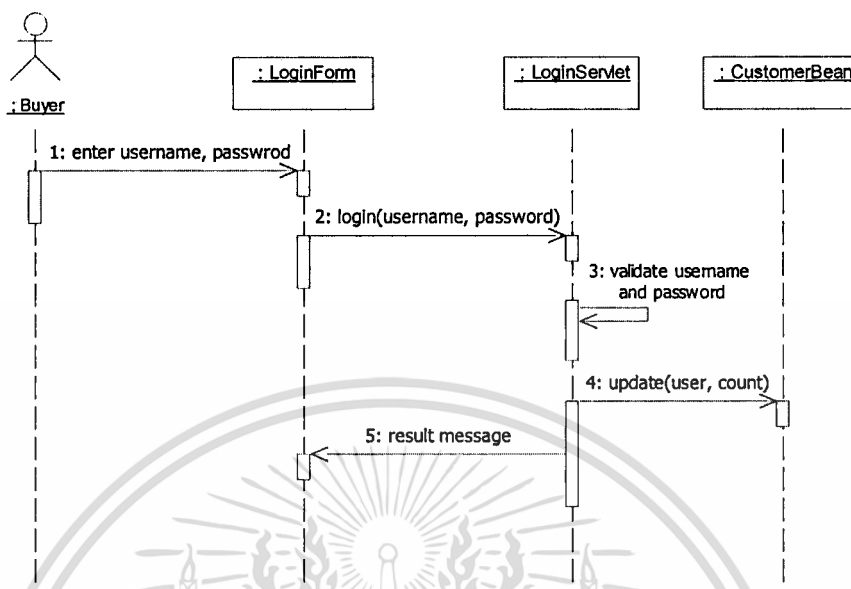


รูปที่ 3.5 ซีควেনซ์ไดอะแกรมของ Customer Register

3.3.3.3 ซีควেনซ์ไดอะแกรมของ Validate User แสดงดังรูปที่ 3.6

อธิบายการทำงานได้ดังนี้

1. ผู้ซื้อเลือกเมนูล็อกอินเข้าสู่ระบบ พร้อมทั้งใส่ชื่อและรหัสผ่าน
2. LoginServlet จะรับข้อมูลชื่อและรหัสผ่าน มาตรวจสอบกับข้อมูลในฐานข้อมูลของลูกค้า ว่าถูกต้องหรือไม่ ก่อนให้เข้าใช้ระบบงาน
3. ถ้าไม่ถูกต้องระบบจะทำการเก็บจำนวนครั้งในการล็อกอิน ถ้าเกินจำนวนครั้งที่กำหนด ระบบจะแสดงข้อความเตือนให้ผู้ใช้งานทราบ
4. ถ้าถูกต้องระบบจะอนุญาตให้เข้าสู่หน้าจอการทำงานของระบบ

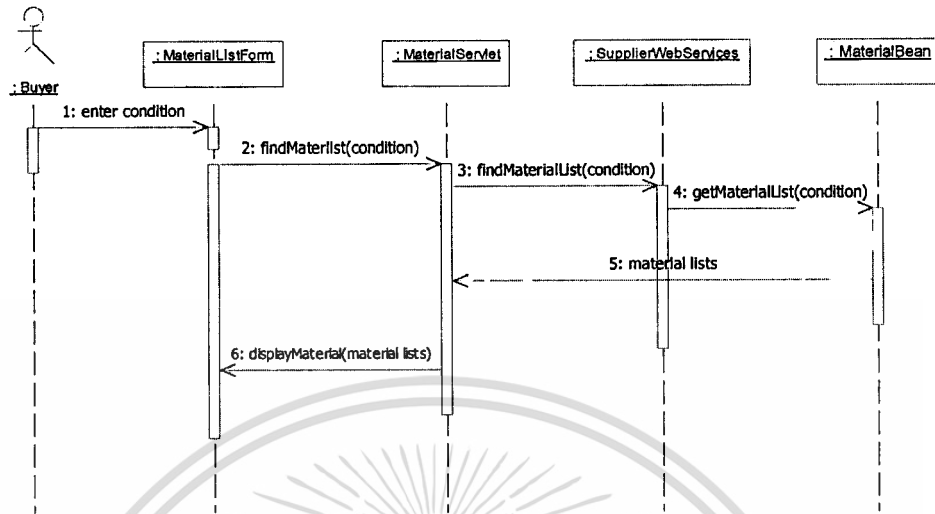


รูปที่ 3.6 ซีเควนซ์ไดอะแกรมของ Validate User

3.3.3.4 ซีเควนซ์ไดอะแกรมของ Request Material List แสดงดังรูปที่ 3.7

อธิบายการทำงานได้ดังนี้

1. การค้นหาข้อมูลวัตถุดิบที่ต้องการสั่งซื้อ เริ่มจากผู้ซื้อใส่เงื่อนไขสำหรับการค้นหา เช่น ระบุชื่อซัพพลายเออร์ หรือรายชื่อบริการที่ต้องการจะเข้าไปขอราคาและสั่งซื้อ
2. โปรแกรมจะทำการค้นหาข้อมูลรายการวัตถุดิบและราคา ตามเงื่อนไขที่ระบุไปยัง เว็บไซต์วิศของซัพพลายเออร์นั้นๆ
3. โดยผลลัพธ์ที่ได้จะถูกจัดการ โดย MaterialServlet แล้วประมวลผลมาแสดงให้ผู้ซื้อบนหน้าจอได้เปรียบเทียบราคา แล้วตัดสินใจสั่งซื้อสินค้าได้

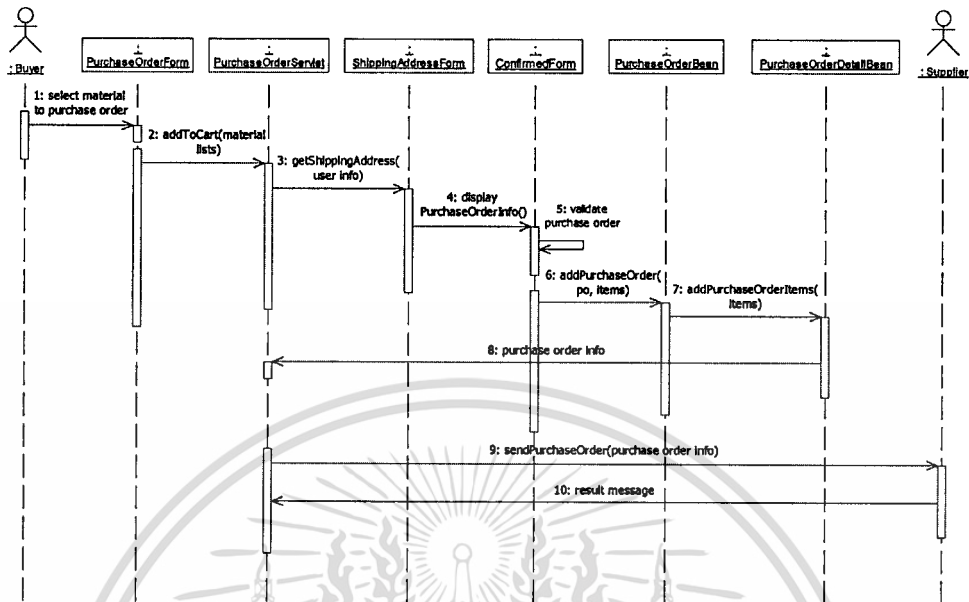


รูปที่ 3.7 ซีควเอนซ์ไดอะแกรมของ Request Material List

3.3.3.5 ซีควเอนซ์ไดอะแกรมของ Purchase Order Material แสดงดังรูปที่ 3.8

อธิบายการทำงานได้ดังนี้

1. ผู้ซื้อเข้าสู่ระบบการใช้งานแล้ว ทำการเลือกและระบุจำนวนวัตถุดิบที่ต้องการจะสั่งซื้อ
2. เมื่อผู้ซื้อกดปุ่มให้คำนวณราคา PurchaseOrderServlet จะทำการตรวจสอบและคำนวณราคาสินค้าที่สั่งซื้อ
3. เมื่อผู้ซื้อกดปุ่มสั่งซื้อ PurchaseOrderServlet จะทำการตรวจสอบรายการที่สั่งซื้อ แล้วแสดงหน้าจอให้ผู้ซื้อได้ป้อนข้อมูลชื่อและที่อยู่ในการจัดส่งสินค้า
4. เมื่อผู้ซื้อยืนยันการสั่งซื้อสินค้าแล้ว ระบบจะทำการจัดเก็บข้อมูลการสั่งซื้อลงฐานข้อมูลของ Plastic Server และจะส่งไปให้ซัพพลายเออร์อื่นๆ อีกทอดหนึ่ง
5. เมื่อทางฝั่งซัพพลายเออร์ได้รับรายการสั่งซื้อสินค้าแล้ว จะทำการตอบรับกลับมา

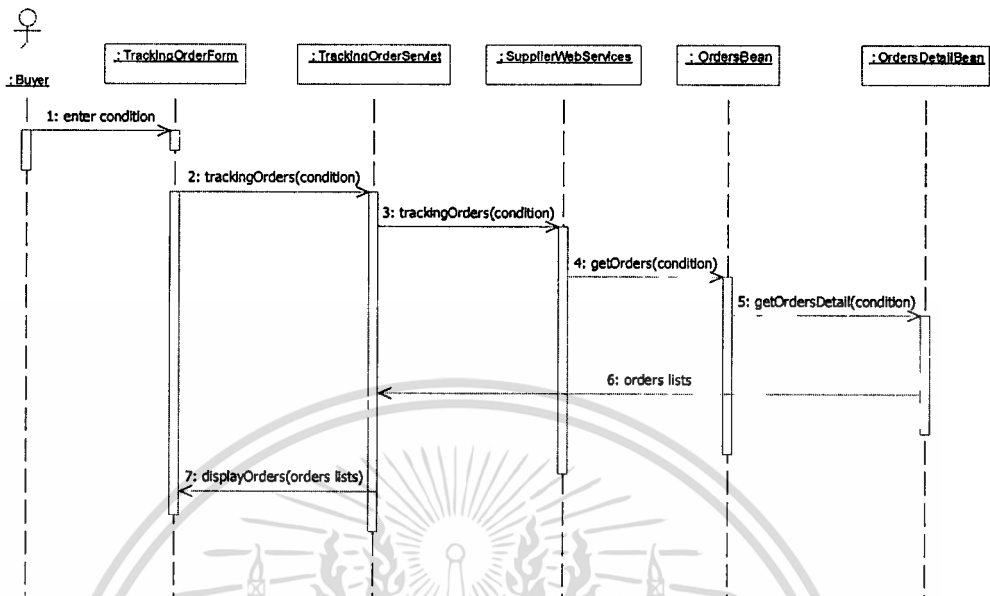


รูปที่ 3.8 ซีเควนซ์ไดอะแกรมของ Purchase Order Material

3.3.3.6 ซีเควนซ์ไดอะแกรมของ Tracking Order แสดงดังรูปที่ 3.9

อธิบายการทำงานได้ดังนี้

1. เมื่อผู้ซื้อใส่วันที่ใบสั่งซื้อสินค้า ที่ต้องการสอบถาม
2. ระบบจะทำการตรวจสอบวันที่ใบสั่งซื้อ และดึงข้อมูลใบสั่งซื้อจากซัพพลายเออร์
3. เมื่อมีการเปลี่ยนแปลงสถานะของใบสั่งซื้อนั้นระบบจะทำการ ปรับปรุงข้อมูลสถานะของใบสั่งซื้อสินค้านั้น แล้วดึงสถานะ ใบสั่งซื้อสินค้าให้ผู้ใช้ทราบผ่านทาง Browser



รูปที่ 3.9 ซีควเอนซ์ไดอะแกรมของ Tracking Order

3.4 การออกแบบระบบงานเว็บเซอร์วิสของ Plastic Supplier

ระบบงานเว็บเซอร์วิสของ Plastic Supplier ทำหน้าที่ให้บริการต่างๆ ให้กับโรงงานผลิตวัตถุดิบประเภทพลาสติก โดยใช้ Java Application , JavaBeans และ AXIS ซึ่งติดตั้งอยู่บนระบบปฏิบัติการ Windows XP และใช้ระบบฐานข้อมูล Postgres และ MySQL

ขั้นตอนการทำงานของระบบงานเว็บเซอร์วิสของ Plastic Supplier มีดังนี้

1. Plastic Supplier ทำการลงทะเบียนบริการต่างๆ กับ Registry Server ซึ่งได้แก่
 - MaterialListService สำหรับให้บริการข้อมูลรายการวัตถุดิบ
 - CreateOrderService สำหรับให้บริการจัดเก็บข้อมูลรายการสั่งซื้อสินค้า
 - TrackingOrderService สำหรับให้บริการตรวจสอบสถานะของการสั่งซื้อสินค้า
2. จากนั้น Plastic Supplier จะทำการรอกการขอเรียกใช้บริการจากโรงงานผลิตวัตถุดิบประเภทพลาสติก ที่ทำการค้นหาข้อมูลการให้บริการ ของซัพพลายเออร์ ที่ลงทะเบียนไว้กับ Registry Server

3.4.1 ยูสเคสไดอะแกรม

การวิเคราะห์และออกแบบระบบ จะเริ่มจากการเก็บรวบรวมความต้องการต่างๆ ของระบบ ทำให้ทราบถึงฟังก์ชันการทำงานของระบบ และแอกเตอร์ที่เกี่ยวข้องกับระบบ ที่จะนำมาใช้เขียนเป็นยูสเคสไดอะแกรม ดังรูปที่ 3.10



รูปที่ 3.10 ยูสเคสไดอะแกรมของ Plastic Supplier

แอกเตอร์ที่เกี่ยวข้องกับระบบได้แก่

- Supplier ผู้จัดจำหน่ายวัตถุดิบให้กับโรงงาน

ยูสเคสที่เกี่ยวข้องกับระบบได้แก่

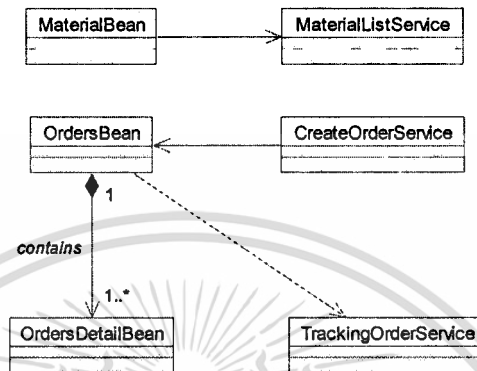
- Material List Service เป็นบริการสำหรับให้ข้อมูลรายการวัตถุดิบของซัพพลายเออร์
- Create Order Service เป็นบริการสำหรับการจัดเก็บข้อมูลรายการสั่งซื้อสินค้า ให้กับซัพพลายเออร์
- Tracking Order Service เป็นบริการในการตรวจสอบสถานะรายการสั่งซื้อสินค้าของซัพพลายเออร์

3.4.2 คลาสไดอะแกรม

เมื่อผ่านขั้นตอนวิเคราะห์เข้าสู่ขั้นตอนออกแบบ ทำให้สามารถออกแบบคลาสไดอะแกรมสำหรับอธิบายรายละเอียดของคลาสภายในโปรแกรมที่ต้องสร้างขึ้น และความสัมพันธ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างคลาส โดยคลาสใช้สัญลักษณ์เป็นสี่เหลี่ยมผืนผ้า ภายในแบ่งเป็น ชื่อคลาส คุณสมบัติ และ เมธอดเรียงจากบนลงล่าง ดังรูปที่ 3.11



รูปที่ 3.11 คลาสไดอะแกรมของ Plastic Supplier

คลาสภายในโปรแกรมที่ต้องสร้างขึ้น ได้แก่

1. MaterialListService เป็นคลาสสำหรับให้บริการข้อมูลวัตถุดิบ
2. MaterialBean เป็นคลาสสำหรับจัดเก็บข้อมูลวัตถุดิบ
3. CreateOrderService เป็นคลาสสำหรับให้บริการจัดเก็บข้อมูลรายการสั่งซื้อสินค้าให้กับซัพพลายเออร์
4. TrackingOrderService เป็นคลาสสำหรับให้บริการตรวจสอบสถานะรายการสั่งซื้อสินค้าของซัพพลายเออร์
5. OrdersBean เป็นคลาสสำหรับจัดเก็บข้อมูลรายการสั่งซื้อส่วนหลัก ของซัพพลายเออร์
6. OrdersDetailBean เป็นคลาสสำหรับจัดเก็บข้อมูลรายการสั่งซื้อส่วนรายละเอียด ของซัพพลายเออร์

3.4.3 ซีเควนซ์ไดอะแกรม

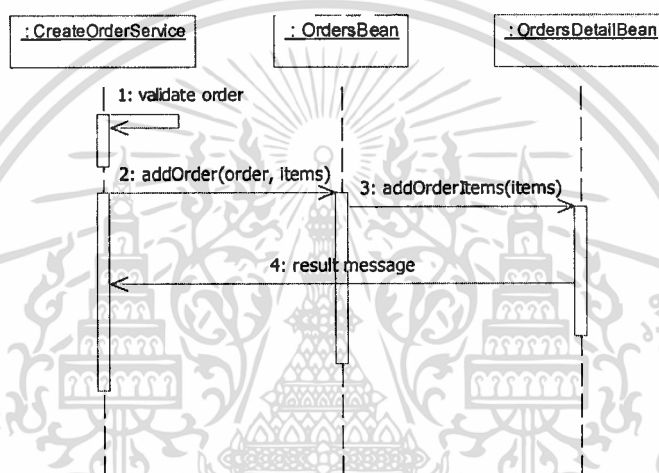
เมื่อผ่านขั้นตอนการออกแบบและมีความต้องการอธิบายความสัมพันธ์ระหว่างยูสเคส และ คลาสไดอะแกรม ทำให้สามารถออกแบบซีเควนซ์ไดอะแกรม โดยในส่วนการออกแบบระบบงาน เว็บไซต์ervis ของ Plastic Supplier จะขอยกตัวอย่างเฉพาะซีเควนซ์ไดอะแกรมที่สำคัญ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3.1 ซีควเอนซ์ไดอะแกรมของ Create Order Service แสดงดังรูปที่ 3.12

อธิบายการทำงาน ได้ดังนี้

1. เมื่อมีการร้องขอการให้บริการจัดเก็บข้อมูลรายการสั่งซื้อสินค้า CreateOrderService จะทำการตรวจสอบข้อมูลที่ได้รับ
2. เมื่อตรวจสอบความถูกต้องของข้อมูลที่ได้รับแล้ว จะทำการสร้างรายการสั่งซื้อ
3. หลังจากสร้างรายการสั่งซื้อแล้ว จึงทำการส่งคืนค่าสถานะของการสร้างรายการสั่งซื้อ



รูปที่ 3.12 ซีควเอนซ์ไดอะแกรมของ Create Order Service

3.5 การออกแบบเว็บเซอร์วิส

การออกแบบเว็บเซอร์วิสของซัพพลายเออร์ เพื่อให้บริการต่างๆ กับโรงงานผลิตวัตถุดิบประเภทพลาสติกนั้น ซัพพลายเออร์แต่ละรายจะต้องทำการลงทะเบียนกับ Registry Server เพื่อประกาศการให้บริการต่างๆ ไว้ ซึ่งรายละเอียดของการให้บริการอยู่ในรูปแบบที่ถูกกำหนดโดย Web Service Description Language (WSDL)

รายละเอียดของการให้บริการเว็บเซอร์วิสของซัพพลายเออร์ ได้แก่

1. คลาส MaterialListService ใช้เมธอด findByName ในการค้นหาข้อมูลวัตถุดิบ โดยมีการรับค่าพารามิเตอร์ name เป็นชื่อวัตถุดิบที่ต้องการค้นหา และส่งคืนค่ากลับ materialList เป็นรายการวัตถุดิบที่ค้นหาได้

2. คลาส `CreateOrderService` ใช้เมธอด `create` ในการจัดเก็บข้อมูลรายการสั่งซื้อสินค้า โดยมีการรับค่าพารามิเตอร์ `orders` เป็นข้อมูลรายการสั่งซื้อสินค้า และส่งคืนค่ากลับ `result` เป็นผลการจัดเก็บข้อมูลรายการสินค้า
3. คลาส `TrackingOrderService` ใช้เมธอด `findByOrderDate` ในการค้นหารายการสั่งซื้อสินค้าที่ต้องการทราบสถานะ โดยมีการรับค่าพารามิเตอร์ `orderDate` เป็นวันที่สั่งซื้อสินค้า และส่งคืนค่ากลับ `orderList` เป็นรายการสั่งซื้อสินค้าที่ต้องการทราบสถานะ



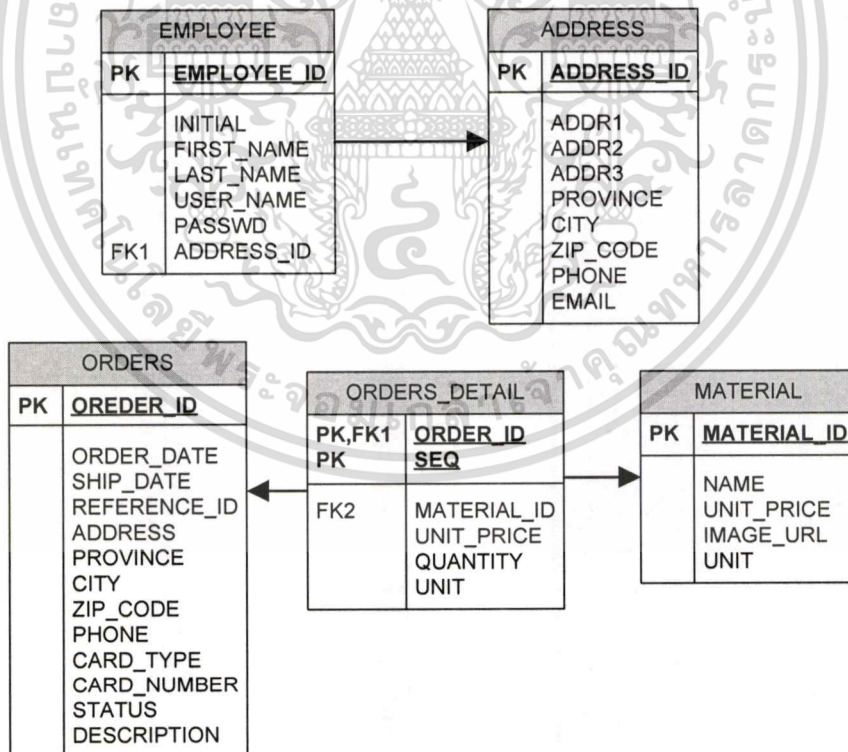
บทที่ 4

การออกแบบฐานข้อมูล

โครงการนี้มีระบบที่ใช้ติดต่อกับฐานข้อมูลอยู่ 3 ระบบ คือฐานข้อมูลของซัพพลายเออร์รายที่ 1, ฐานข้อมูลของซัพพลายเออร์รายที่ 2 และฐานข้อมูลของโรงงานผลิตวัตถุดิบประเภทพลาสติก โดยมีรายละเอียดดังต่อไปนี้

4.1 ฐานข้อมูลของซัพพลายเออร์รายที่ 1

ฐานข้อมูลของซัพพลายเออร์รายที่ 1 ใช้ Postgres เป็นฐานข้อมูล ประกอบด้วย 5 ตารางดังรูปที่ 4.1 คือ

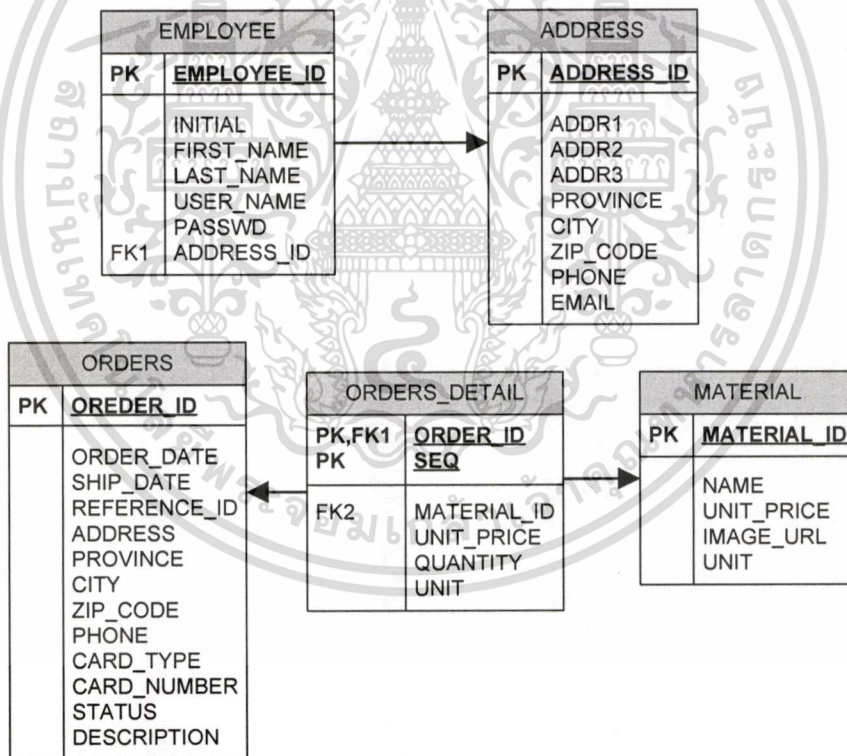


รูปที่ 4.1 ER Diagram ของซัพพลายเออร์รายที่ 1

1. EMPLOYEE เก็บข้อมูลของพนักงาน แสดงรายละเอียดดังตารางที่ 4.1
2. ADDRESS เก็บข้อมูลที่อยู่ของพนักงาน แสดงรายละเอียดดังตารางที่ 4.2
3. ORDERS เก็บข้อมูลการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.3
4. ORDERS_DETAIL เก็บข้อมูลรายละเอียดของการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.4
5. MATERIAL เก็บข้อมูลวัตถุดิบพลาสติก แสดงรายละเอียดดังตารางที่ 4.5

4.2 ฐานข้อมูลของซัพพลายเออร์รายที่ 2

ฐานข้อมูลของซัพพลายเออร์รายที่ 2 ใช้ MySQL เป็นฐานข้อมูล ประกอบด้วย 5 ตารางดังรูปที่ 4.2 คือ



รูปที่ 4.2 ER Diagram ของซัพพลายเออร์รายที่ 2

1. EMPLOYEE เก็บข้อมูลของพนักงาน แสดงรายละเอียดดังตารางที่ 4.1
2. ADDRESS เก็บข้อมูลที่อยู่ของพนักงาน แสดงรายละเอียดดังตารางที่ 4.2
3. ORDERS เก็บข้อมูลการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.3

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของโรงเรียนเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ORDERS_DETAIL เก็บข้อมูลรายละเอียดของการสั่งซื้อวัตถุดิบของลูกค้า แสดงรายละเอียดดังตารางที่ 4.4
5. MATERIAL เก็บข้อมูลวัตถุดิบพลาสติก แสดงรายละเอียดดังตารางที่ 4.5

ตารางที่ 4.1 แสดงรายละเอียดของตาราง EMPLOYEE

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
EMPLOYEE_ID	INT	รหัสพนักงาน	PK	
INITIAL	VARCHAR (15)	คำนำหน้าชื่อ		
FIRST_NAME	VARCHAR (50)	ชื่อพนักงาน		
LAST_NAME	VARCHAR (50)	นามสกุลพนักงาน		
USER_NAME	VARCHAR (20)	ชื่อเข้าใช้ระบบ		
PASSWD	VARCHAR (20)	รหัสผ่าน		
ADDRESS_ID	INT	รหัสที่อยู่	FK	ADDRESS

ตารางที่ 4.2 แสดงรายละเอียดของตาราง ADDRESS

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
ADDRESS_ID	INT	รหัสที่อยู่	PK	
ADDR1	VARCHAR (50)	ที่อยู่ 1		
ADDR2	VARCHAR (50)	ที่อยู่ 2		
ADDR3	VARCHAR (50)	ที่อยู่ 3		
PROVINCE	VARCHAR(40)	เขต/อำเภอ		
CITY	VARCHAR(40)	จังหวัด		
ZIP_CODE	CHAR(5)	รหัสไปรษณีย์		
PHONE	VARCHAR(15)	หมายเลขโทรศัพท์		
EMAIL	VARCHAR(30)	อีเมล		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงรายละเอียดของตาราง ORDERS

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
ORDER_ID	INT	รหัสการสั่งซื้อ	PK	
ORDER_DATE	DATE	วันที่สั่งซื้อ		
SHIP_DATE	DATE	วันที่ส่งสินค้า		
REFERENCE_ID	INT	รหัสอ้างอิง		
ADDRESS	VARCHAR(150)	ที่อยู่		
PROVINCE	VARCHAR(40)	เขต/อำเภอ		
CITY	VARCHAR(40)	จังหวัด		
ZIP_CODE	CHAR(5)	รหัสไปรษณีย์		
PHONE	VARCHAR(15)	หมายเลขโทรศัพท์		
CARD_TYPE	VARCHAR(10)	ประเภทบัตรเครดิต		
CARD_NUMBER	VARCHAR(20)	หมายเลขบัตรเครดิต		
STATUS	CHAR(1)	สถานะสั่งซื้อ		
DESCRIPTION	VARCHAR(80)	รายละเอียดอื่นๆ		

ตารางที่ 4.4 แสดงรายละเอียดของตาราง ORDERS_DETAIL

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
ORDER_ID	INT	รหัสการสั่งซื้อ	PK, FK	ORDER
SEQ	INT	ลำดับที่รายการ	PK	
MATERIAL_ID	INT	รหัสสินค้าและวัตถุดิบ	FK	MATERIAL
UNIT_PRICE	DOUBLE	ราคาต่อหน่วย		
QUANTITY	INT	จำนวนที่สั่งซื้อ		
UNIT	VARCHAR(15)	ชื่อหน่วยของสินค้า		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

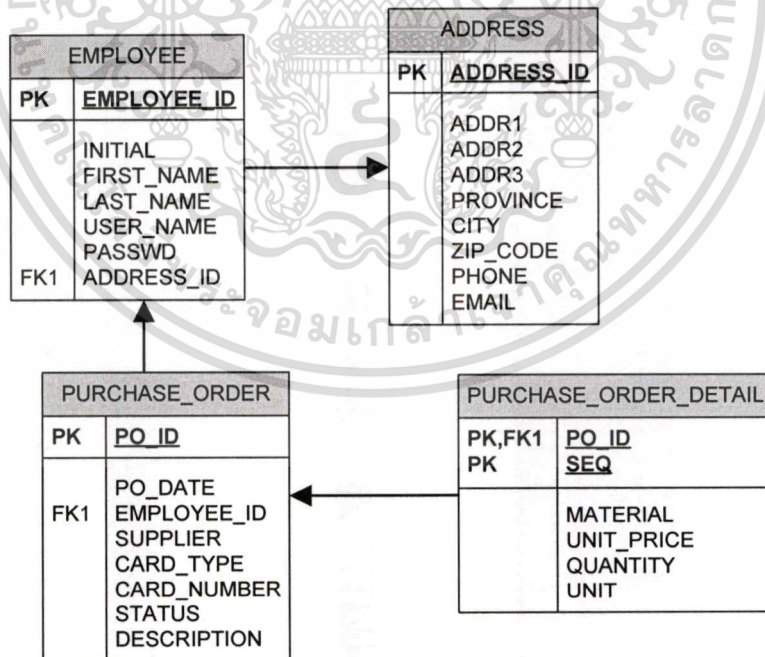
ตารางที่ 4.5 แสดงรายละเอียดของตาราง MATERIAL

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

4.3 ฐานข้อมูลของโรงงานผลิตวัตถุดิบประเภทพลาสติก

ฐานข้อมูลของ Plastic Server ใช้ Postgres เป็นฐานข้อมูล ประกอบด้วย 4 ตารางดังรูปที่

4.3 คือ



รูปที่ 4.3 ER Diagram ของโรงงานผลิตวัตถุดิบประเภทพลาสติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. EMPLOYEE เก็บข้อมูลของพนักงาน แสดงรายละเอียดดังตารางที่ 4.6
2. ADDRESS เก็บข้อมูลที่อยู่ของพนักงาน แสดงรายละเอียดดังตารางที่ 4.7
3. PURCHASE_ORDER เก็บข้อมูลหลักของการสั่งซื้อวัตถุดิบ แสดงรายละเอียดดังตารางที่ 4.8
4. PURCHASE_ORDER_DETAIL เก็บข้อมูลรายละเอียดของการสั่งซื้อวัตถุดิบ แสดงรายละเอียดดังตารางที่ 4.9

ตารางที่ 4.6 แสดงรายละเอียดของตาราง EMPLOYEE

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
EMPLOYEE_ID	INT	รหัสพนักงาน	PK	
INITIAL	VARCHAR (15)	คำนำหน้าชื่อ		
FIRST_NAME	VARCHAR (50)	ชื่อพนักงาน		
LAST_NAME	VARCHAR (50)	นามสกุลพนักงาน		
USER_NAME	VARCHAR (20)	ชื่อเข้าใช้ระบบ		
PASSWD	VARCHAR (20)	รหัสผ่าน		
ADDRESS_ID	INT	รหัสที่อยู่	FK	ADDRESS

ตารางที่ 4.7 แสดงรายละเอียดของตาราง ADDRESS

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
ADDRESS_ID	INT	รหัสที่อยู่	PK	
ADDR1	VARCHAR (50)	ที่อยู่ 1		
ADDR2	VARCHAR (50)	ที่อยู่ 2		
ADDR3	VARCHAR (50)	ที่อยู่ 3		
PROVINCE	VARCHAR(40)	เขต/อำเภอ		
CITY	VARCHAR(40)	จังหวัด		
ZIP_CODE	CHAR(5)	รหัสไปรษณีย์		
PHONE	VARCHAR(15)	หมายเลขโทรศัพท์		
EMAIL	VARCHAR(30)	อีเมล		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 แสดงรายละเอียดของตาราง PURCHASE_ORDER

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
PO_ID	INT	รหัสการสั่งซื้อ	PK	
PO_DATE	DATE	วันที่สั่งซื้อ		
EMPLOYEE_ID	INT	รหัสพนักงาน	FK	EMPLOYEE
SUPPLIER	VARCHAR(50)	ชื่อซัพพลายเออร์ที่สั่งซื้อ		
CARD_TYPE	VARCHAR(10)	ประเภทบัตรเครดิต		
CARD_NUMBER	VARCHAR(20)	หมายเลขบัตรเครดิต		
STATUS	CHAR(1)	สถานะสั่งซื้อ		
DESCRIPTION	VARCHAR(80)	รายละเอียดอื่นๆ		

ตารางที่ 4.9 แสดงรายละเอียดของตาราง PURCHASE_ORDER_DETAIL

ชื่อฟิลด์	ประเภท	คำอธิบาย	คีย์	ตารางอ้างอิง
PO_ID	INT	รหัสการสั่งซื้อ	PK, FK	PURCHASE_ORDER
SEQ	INT	ลำดับที่รายการ	PK	
MATERIAL	VARCHAR(50)	ชื่อสินค้าและวัตถุดิบ		
UNIT_PRICE	DOUBLE	ราคาต่อหน่วย		
QUANTITY	INT	จำนวนที่สั่งซื้อ		
UNIT	VARCHAR(15)	ชื่อหน่วยของสินค้า		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การพัฒนาระบบ

ในการพัฒนาระบบจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส สามารถอธิบายได้
เป็นหัวข้อ 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 Server : Apache Tomcat 5.0.28 และ Web Services - Axis 1.3
- RDBMS : MySQL 4.0.17 และ PostgreSQL 8.0
- Web Browser : Internet Explorer 6.0

5.1.3 เครื่องมือ

เครื่องมือที่ใช้ในการพัฒนา และทดสอบระบบ มีดังนี้

- Java IDE : Eclipse SDK 3.1.1

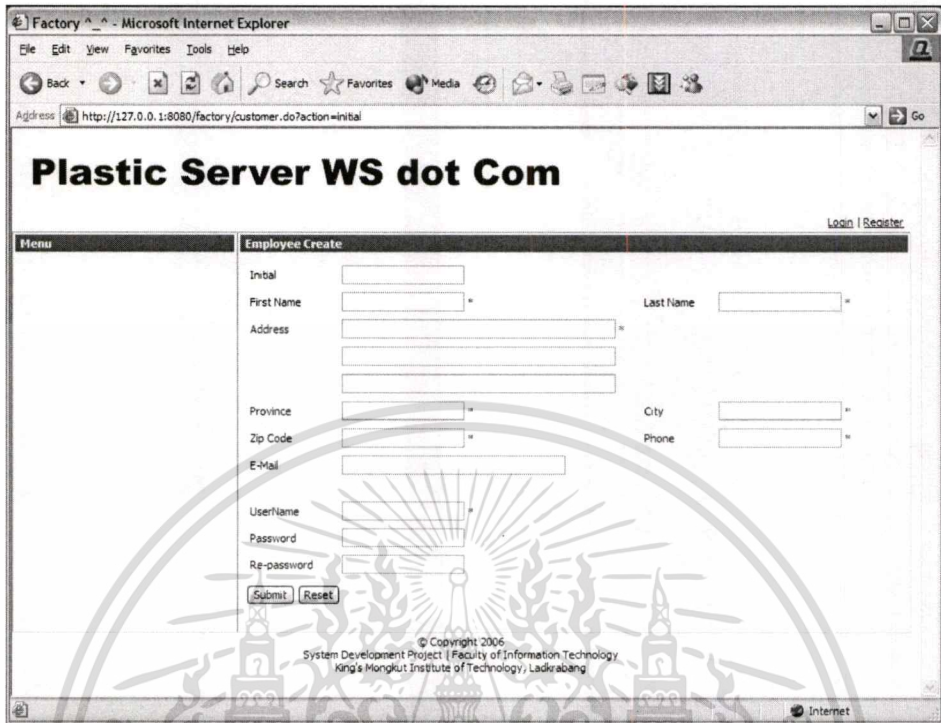
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Web Development Tool : Macromedia Dreamweaver MX
- UML Tool : Rational Rose 2000 Enterprise Edition
- Build Tool : Apache Ant 1.6.2

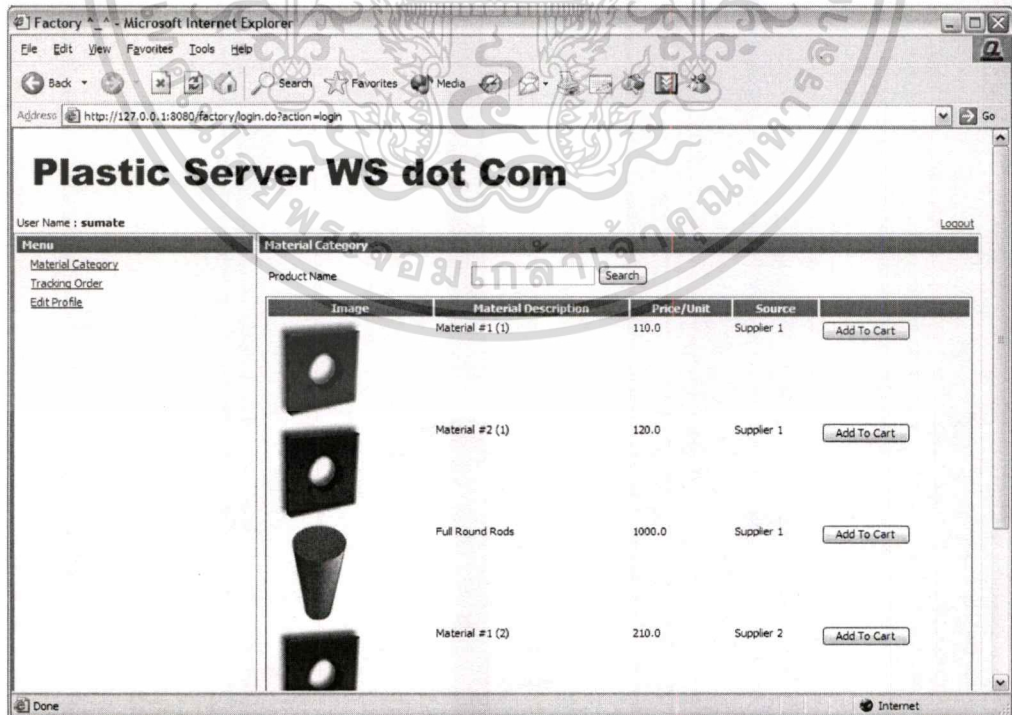
5.2 หน้าจอการทำงานของระบบ

หน้าจอการทำงานของระบบ หรือส่วนที่ติดต่อกับผู้ใช้งาน มีดังต่อไปนี้

1. หน้าจอขึ้นทะเบียนสำหรับพนักงานใหม่ แสดงดังรูปที่ 5.1
2. หน้าจอแสดงรายการวัตถุดิบ แสดงดังรูปที่ 5.2 โดย Plastic Server จะทำการร้องขอข้อมูลรายการวัตถุดิบและราคาจากซัพพลายเออร์ต่างๆ
3. หน้าจอการสั่งซื้อวัตถุดิบ แสดงดังรูปที่ 5.3
4. หน้าจอกรอกข้อมูลในการจัดส่งสินค้าที่สั่งซื้อ แสดงดังรูปที่ 5.4
5. หน้าจอยืนยันการสั่งซื้อวัตถุดิบ แสดงดัง โดยที่รูปที่ 5.5 Plastic Server จะเก็บข้อมูลการสั่งซื้อ และส่งข้อมูลการสั่งซื้อไปให้กับซัพพลายเออร์อีกทอดหนึ่ง
6. หน้าจอแสดงรายการสั่งซื้อวัตถุดิบหลังยืนยัน แสดงดังรูปที่ 5.6 โดยทางฝั่งซัพพลายเออร์ได้รับใบสั่งซื้อแล้ว จะตอบรับการสั่งซื้อด้วยการส่ง วันที่ส่งสินค้ากลับมา
7. หน้าจอสอบถามการสั่งซื้อวัตถุดิบ แสดงดังรูปที่ 5.7 ซึ่งจะแสดงรายการที่สั่งซื้อ สถานะการสั่งซื้อ และที่อยู่ในการจัดส่ง



รูปที่ 5.1 หน้าจอขึ้นทะเบียนสำหรับพนักงานใหม่



รูปที่ 5.2 หน้าจอแสดงรายการวัตถุดิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Factory ^_^ - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://127.0.0.1:8080/factory/material.do

Plastic Server WS dot Com

User Name : sumate Logout

Menu

- Material Category
- Tracking Order
- Edit Profile

Purchase Order

No.	Material Description	Price/Unit	Quantity	Total	Source
1	Material #1 (1)	110.0	<input type="text" value="1"/>	110.0	Supplier 1
Summary			1	110.0	<input type="button" value="Reload"/>

© Copyright 2006
System Development Project | Faculty of Information Technology
King's Mongkut Institute of Technology, Ladkrabang

Done Internet

รูปที่ 5.3 หน้าจอการสั่งซื้อวัตถุดิบ

Factory ^_^ - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://127.0.0.1:8080/factory/purchaseOrder.do

Plastic Server WS dot Com

User Name : sumate Logout

Menu

- Material Category
- Tracking Order
- Edit Profile

Shipping Address

Address

Province * City *

Zip Code * Phone *

Credit Card

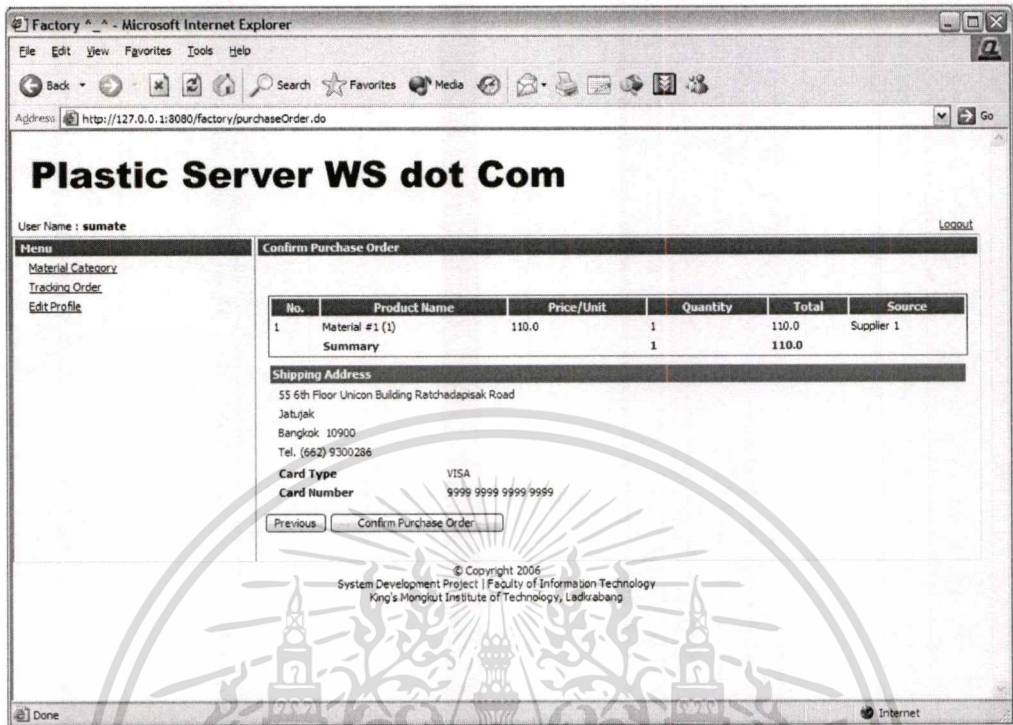
Card Number * Card Digit *

© Copyright 2006
System Development Project | Faculty of Information Technology
King's Mongkut Institute of Technology, Ladkrabang

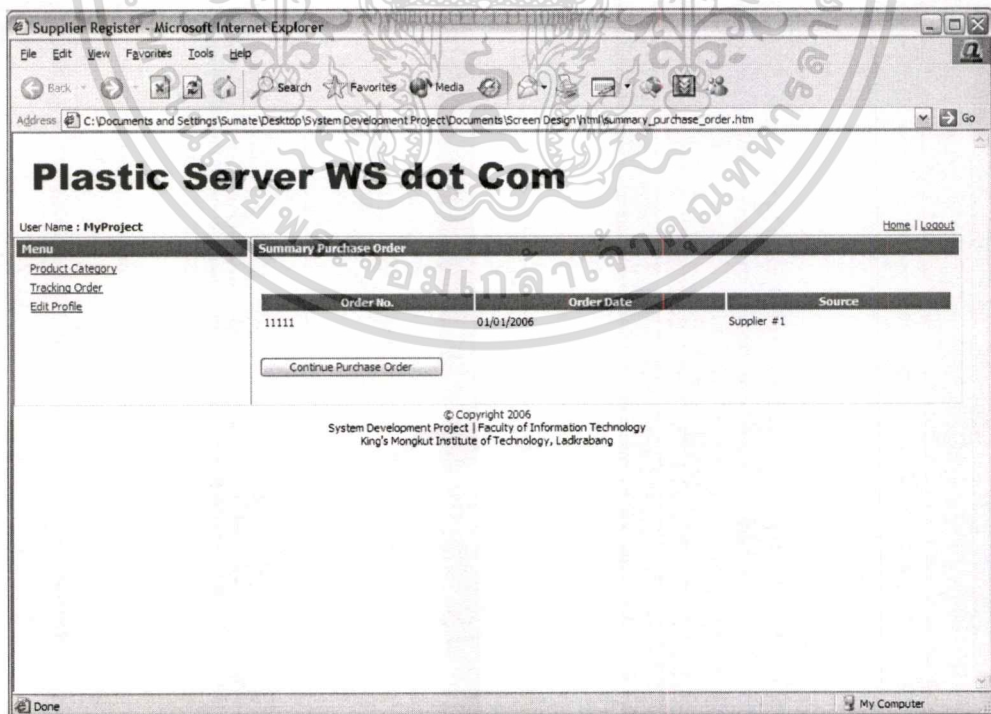
Done Internet

รูปที่ 5.4 หน้าจอรอกข้อมูลในการจัดส่งสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 หน้าจอยืนยันการสั่งซื้อวัตถุดิบ



รูปที่ 5.6 หน้าจอแสดงรายการสั่งซื้อวัตถุดิบหลังยืนยัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Factory ^_^ - Microsoft Internet Explorer

Address: http://127.0.0.1:8080/factory/trackingOrder.do

Plastic Server WS dot Com

User Name : sumate Logout

Menu

- [Material Category](#)
- [Tracking Order](#)
- [Edit Profile](#)

Tracking Order

Order Date:

Order No	26	Order Date	28/02/2006
Ship Date	28/02/2006	Reference ID	64
Address	55 6th Floor Unicon Building Ratchadapisek Road		
Province	Jatujak	City	Bangkok
ZipCode	10900	Phone	(662) 9300286
Card Type	VISA	Card Number	456
Source	Supplier 1	Status	A

No.	Product Name	Price/Unit	Quantity	Total
1	Material #1 (1)	110.0	1	110.0
2	Full Round Rods	1000.0	1	1000.0
Summary			2	1110.0

Order No	27	Order Date	09/03/2006
Ship Date	09/03/2006	Reference ID	65
Address	55 6th Floor Unicon Building Ratchadapisek Road		
Province	Jatujak	City	Bangkok
ZipCode	10900	Phone	(662) 9300286

Done Internet

รูปที่ 5.7 หน้าจอสอบถามการสั่งซื้อวัสดุดิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุป

ในบทนี้เป็นการสรุปภาพโดยรวมของการพัฒนาระบบ รวมทั้งข้อเสนอแนะในการพัฒนาระบบเพิ่มเติม โดยสรุปได้เป็นหัวข้อ ดังนี้

6.1 สรุปผลการออกแบบและพัฒนาระบบ

โครงการนี้ได้นำความรู้ที่ศึกษามาในเรื่องของเทคโนโลยีเว็บเซอร์วิส และทฤษฎีที่เกี่ยวข้องมาประยุกต์ใช้กับเว็บแอปพลิเคชัน เพื่อให้การทำงานของเว็บแอปพลิเคชันมีประสิทธิภาพมากขึ้น โดยการยกตัวอย่างของการพัฒนาระบบจัดซื้อวัตถุดิบประเภทพลาสติกโดยใช้เว็บเซอร์วิส เพื่อทดสอบว่าเทคโนโลยีเว็บเซอร์วิส สามารถนำมาช่วยให้ระบบการพาณิชย์อิเล็กทรอนิกส์มีประสิทธิภาพมากขึ้น และรองรับการทำงานในหลากหลายรูปแบบ

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

6.2 ประโยชน์ที่ได้รับจากการพัฒนาระบบ

ประโยชน์ที่ได้รับจากการพัฒนาระบบ สรุปได้ดังนี้

1. ได้ศึกษาเทคโนโลยีเว็บเซอร์วิสต่างๆ แล้วนำมาประยุกต์ใช้กับเว็บแอปพลิเคชันการสั่งซื้อสินค้าผ่านอินเทอร์เน็ต เพื่อให้การทำงานของระบบการสั่งซื้อ มีประสิทธิภาพมากขึ้น
2. ได้ความรู้ในเรื่องของการวิเคราะห์และออกแบบระบบด้วย UML และนำไปใช้กับระบบงานอื่นๆ ได้
3. นำความรู้ที่ได้จากการศึกษาเทคโนโลยีเว็บเซอร์วิส ไปประยุกต์ใช้กับระบบงานอื่นๆ

6.3 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคระหว่างการพัฒนาระบบ สามารถสรุปได้ดังนี้

1. เนื่องจากเทคโนโลยีที่นำมาใช้ในโครงการนี้เป็นเทคโนโลยีใหม่ ทำให้เสียเวลาในการศึกษาและทำความเข้าใจ
2. ปัญหาของอุปกรณ์คอมพิวเตอร์ และความเร็วในการติดต่อสื่อสารของอินเทอร์เน็ต ทำให้การทดสอบระบบจึงทำบนเครื่องคอมพิวเตอร์เครื่องเดียวกันเท่านั้น

6.4 ข้อเสนอแนะ

ในการพัฒนาระบบงานเว็บแอปพลิเคชัน โดยใช้เทคโนโลยีเว็บเซอร์วิส เป็นการจำลองการทำงาน เพื่อการศึกษาและทำความเข้าใจกับเทคโนโลยีเว็บเซอร์วิสต่างๆ หากมีการพัฒนาระบบงานต่อไปเพื่อให้สามารถนำไปใช้งานจริงได้อย่างสมบูรณ์ จำเป็นต้องปรับปรุง และพัฒนาเพิ่มเติมดังต่อไปนี้

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

บรรณานุกรม

- IBM. 2002. **Web Services – The Web’s Next Revolution**. [Online]. Available:
<ftp://www6.software.ibm.com/software/developer/library/ws-wnextrev.pdf>.
- Ivan, Herman. 2005. **Overview of XSLT and XPath**. [Online]. Available:
http://www.w3.org/Consortium/Offices/Presentations/XSLT_XPATH/.
- Orl, Ed and Mandava, Ramesh. 2002. **Java Web Services Developer Pack Part 1 : Registration and the JAXR API**. [Online]. Available:
- Rahul, Sharma. 2002. **Java API for XML-based RPC (JAX-RPC)**. [Online]. Available:
<http://java.sun.com/developer/technicalArticles/xml/jaxrpc/>.
<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 นายสุเมธ กาญจนภาส
วันเกิด 30 ตุลาคม 2517
สถานที่เกิด กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี วท.บ. (สถิติประยุกต์)
คณะวิทยาศาสตร์ประยุกต์
สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ
การทำงาน System Analyst
บริษัท อิงเกรส (ประเทศไทย) จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้