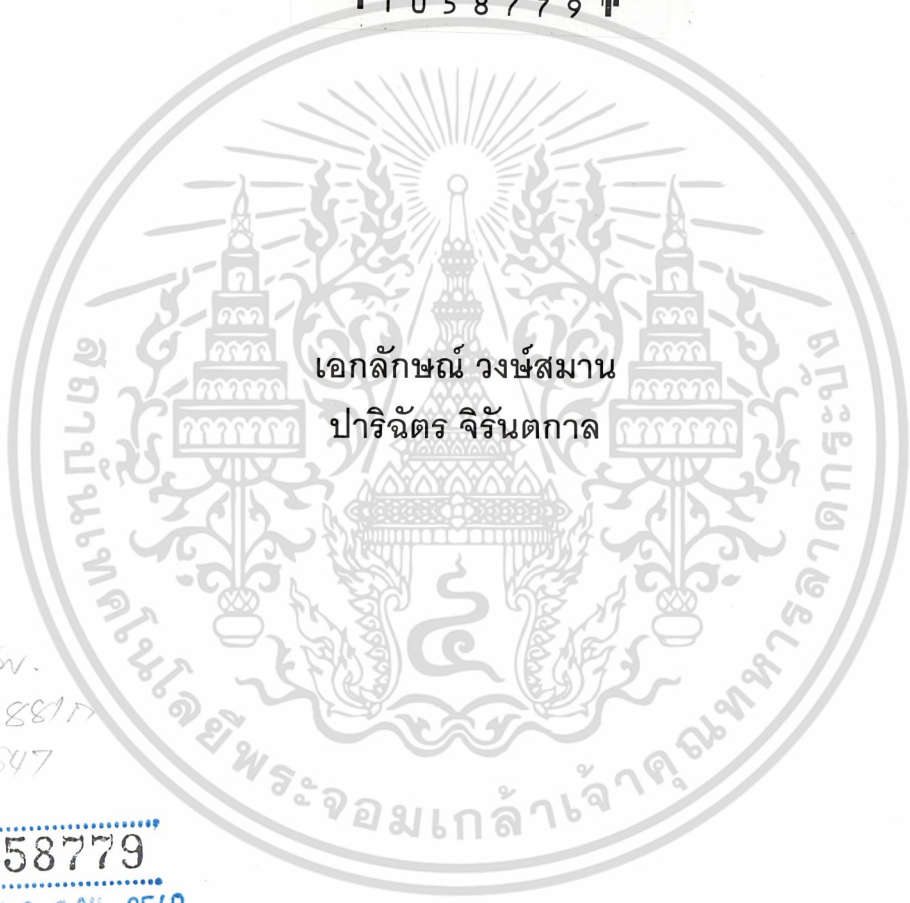


การพัฒนาไมดเดิลแวร์โดยใช้วิธีการส่งผ่านข้อมูลแบบออบเจ็กต์พาสซิ่ง

MIDDLEWARE DEVELOPING USING OBJECT PASSING
TECHNIQUE



เอกลักษณ์ วงษ์สมาน
ปาริฉัตร จิรันตกาล

รฟว.
ค 881 ก
2547

เลขหมู่.....
เลขทะเบียน..... 58779
วัน,เดือน,ปี..... 10 ก.พ. 2549

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11199002
5.....
1.....

MIDDLEWARE DEVELOPING USING OBJECT PASSING
TECHNIQUE



EKKALUK WONGSAMAN
PARICHAT JIRUNTAGARN

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การพัฒนาไมดเดิลแวร์โดยใช้วิธีการส่งผ่านข้อมูลแบบออบเจกต์พาสซิง	
ชื่อนักศึกษา	นายเอกลักษณ์ วงษ์สมาน	44050477
	นางสาวปาริฉัตร จิรันตกาล	44050432
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2547	
อาจารย์ที่ปรึกษา	อ.กฤษฎา บุศรา	
	อ.วิสันต์ ตั้งวงษ์เจริญ	

บทคัดย่อ

ซอฟต์แวร์ในปัจจุบันเน้นพัฒนาโปรแกรมในลักษณะเชิงวัตถุมากขึ้น ในขณะที่ระบบฐานข้อมูลเริ่มมีการจัดเก็บออบเจกต์ได้โดยตรง แต่โปรแกรมที่ติดต่อกับฐานข้อมูลนั้นยังคงใช้วิธีการส่งข้อความที่เป็นภาษาเอสคิวแอลเป็นหลัก ทำให้เป็นภาระแก่ผู้พัฒนาโปรแกรม เนื่องจากโปรแกรมซอฟต์แวร์ต่างๆ มีกลไกภายในแตกต่างกัน ก่อให้เกิดความผิดพลาด และยุ่งยากต่อการติดต่อกับฐานข้อมูล

จากปัญหาดังกล่าวทำให้การพัฒนาไมดเดิลแวร์ซึ่งทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างโปรแกรมซอฟต์แวร์กับระบบฐานข้อมูลบนเครือข่ายไคลเอนท์และเซิร์ฟเวอร์ต้องการวิธีการที่จะส่งผ่านข้อมูลในรูปแบบโครงสร้างเชิงวัตถุ ซึ่งในการแก้ปัญหานี้ทางฝั่งของไคลเอนท์จะเป็นฝั่งที่ทำการร้องขอการทำงานไมดเดิลแวร์ซึ่งจะติดต่อไปยังส่วน Servlet ที่มีอยู่ในตัวไมดเดิลแวร์ ซึ่งตัว Servlet นี้จะรอรับ request อยู่แล้ว โดยจะส่งผ่านข้อมูลของออบเจกต์ในรูปแบบของเอกสารเอ็กซ์เอ็มแอล ซึ่งจะใช้ Hibernate ช่วยในการเรียกใช้และส่งผ่านข้อมูลไปยังระบบฐานข้อมูลเพื่อทำการจัดการกับข้อมูลตามที่ไคลเอนท์ร้องขอ ทำให้การส่งผ่านข้อมูลทำได้ง่ายและสะดวกสบายขึ้น อีกทั้งเป็นการเพิ่มประสิทธิภาพให้แก่ระบบงาน

Special Project Title	MIDDLEWARE DEVELOPING USING OBJECT PASSING TECHNIQUE
Students	Ekkaluk Wongsaman 44050477 Parichat Jiruntagarn 44050432
Degree	Bachelor of Science
Programme	Computer Science
Academic Year	2004
Special Project Advisor	Kridsada Budsara Wisat Tangwongcharoen

ABSTRACT

Software development emphasized in Object-Oriented concept but when we work with DBMS by using SQL command which is a non-Object Oriented way. Also, most commercial DBMS in marketplace is not Object-Oriented DBMS. This is difficulty of software developer to work with non-Object Oriented DBMS in Object-Oriented way.

This special problem is the program that acts as middleware between Object-Oriented software and non-Object Oriented DBMS. Middleware is requested by Client which contact into middleware servlet. It use XML document for its mechanism to transfer the object between Client and the middleware program. And use hibernate help for map data pass to database for manage data follow Client's request that increase defiantly of data transfer.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษหัวข้อ การพัฒนามิตเดิลแวร์โดยใช้วิธีการส่งผ่านข้อมูลแบบ ออบเจกต์พาสซึ่ง สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ อาจารย์กฤษฎา บุศรา และอาจารย์วิสันต์ ตั้งวงษ์เจริญ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ ที่กรุณาให้ คำแนะนำ และเป็นທີ່ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหา พิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้าน กำลังใจและทุนทรัพย์จนการทำปัญหาพิเศษนี้สำเร็จด้วยดี ขอขอบพระคุณอาจารย์ทุกท่านที่ให้ความรู้ทั้งในภาคทฤษฎีและภาคปฏิบัติ รวมทั้งเพื่อนๆ ทุกคนที่มีส่วนช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ
มีนาคม 2548



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อปัญหาพิเศษภาษาไทย.....	I
บทคัดย่อปัญหาพิเศษภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหา.....	2
1.2 วัตถุประสงค์ของการทำปัญหาพิเศษ.....	2
1.3 ขอบเขตของปัญหา.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนในการดำเนินงาน.....	3
1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ.....	3
1.7 งบประมาณที่ใช้ในการทำปัญหาพิเศษ.....	3
บทที่ 2 หลักการและทฤษฎี.....	4
2.1 ความรู้ทั่วไปเกี่ยวกับระบบจัดการฐานข้อมูลเชิงวัตถุ.....	4
2.1.1 คุณสมบัติพื้นฐานของออบเจกต์.....	4
2.1.2 ความรู้ทั่วไปเกี่ยวกับระบบจัดการฐานข้อมูล.....	5
2.1.3 โครงสร้างข้อมูลของฐานข้อมูลแบบออบเจกต์โอเรียนเตด.....	5
2.1.4 คุณสมบัติต่างๆ ของระบบจัดการฐานข้อมูลเชิงวัตถุ.....	6
2.2 ภาษาเอสคิวแอล1999.....	8
2.2.1 คุณสมบัติของภาษาเอสคิวแอล.....	8
2.2.2 คุณสมบัติของภาษาเอสคิวแอล1999.....	8
2.2.3 แผนผังแสดงการเพิ่มเติมชนิดของข้อมูลในเอสคิวแอล1999.....	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.3 ความรู้เกี่ยวกับการแมปโป้งข้อมูลลงฐานข้อมูล.....	10
2.3.1 เทเบิลเบสด์แมปโป้ง.....	10
2.3.2 ออปเจ็กต์รีเลชันนอลแมปโป้ง.....	12
2.3.3 การแมปโป้งข้อมูลโดยใช้ไฮเบอร์เน็ต.....	14
2.4 ความรู้ทั่วไปเกี่ยวกับมิดเดิลแวร์.....	16
2.4.1 รูปแบบของระบบไคลเอนท์เซิร์ฟเวอร์.....	17
2.4.1.1 สแตนด์อโลนไคลเอนท์เซิร์ฟเวอร์.....	17
2.4.1.2 ดีพาร์ทเมนต์ไคลเอนท์เซิร์ฟเวอร์.....	18
2.4.1.3 เอนเตอร์ไพรส์ไคลเอนท์เซิร์ฟเวอร์.....	18
2.5 ความรู้ทั่วไปเกี่ยวกับเจดีบีซี.....	19
2.5.1 การทำงานของเจดีบีซีแอปพลิเคชัน.....	19
2.5.2 ข้อดีของเทคโนโลยีเจดีบีซี.....	19
2.5.3 ลักษณะพิเศษของเทคโนโลยีเจดีบีซี.....	19
2.6 ความรู้ทั่วไปเกี่ยวกับเอ็กซ์เอ็มแอล.....	21
2.6.1 ลักษณะโครงสร้างของเอ็กซ์เอ็มแอล.....	21
2.6.1.1 ความหมายของแท็กกับอิลีเมนต์.....	22
2.6.1.2 ความหมายของคอนเท้นท์กับแอททริบิวต์.....	23
2.6.2 วัตถุประสงค์หลักของเอ็กซ์เอ็มแอล.....	23
2.6.3 ภาษาลำดับจัดการโครงสร้างเอกสาร.....	23
2.6.4 การประยุกต์ใช้เกี่ยวกับเอ็กซ์เอ็มแอล.....	24
2.6.5 ดีทีดีข้อกำหนดโครงสร้างสำหรับเกี่ยวกับเอ็กซ์เอ็มแอล.....	26
2.6.5.1 กฎและคำศัพท์ที่เกี่ยวข้อง.....	29
บทที่ 3 การวิเคราะห์และออกแบบ.....	31
3.1 การวิเคราะห์การทำงานของมิดเดิลแวร์ที่จะทำการพัฒนา.....	31
3.2 รายละเอียดการออกแบบส่วนประกอบต่างๆ เพื่อใช้ในการทำงานของมิดเดิลแวร์...32	32
3.2.1 การออกแบบการติดต่อระหว่างส่วนประกอบต่างๆ เพื่อใช้ในการทำงานของมิดเดิลแวร์ที่สามารถส่งผ่านข้อมูลในรูปแบบเชิงวัตถุ.....	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.2.2 การออกแบบคอมโพเนนต์ต่างๆ ในระบบ.....	34
3.3 การออกแบบขั้นตอนการทำงานของมิดเดิลแวร์.....	35
3.3.1 การออกแบบขั้นตอนการทำงานของมิดเดิลแวร์สำหรับการเพิ่มเติม ลบ และเปลี่ยนแปลงข้อมูล.....	35
3.3.1.1 ซีควเอนซ์ไดอะแกรมของการเพิ่มเติม ลบ และเปลี่ยนแปลงข้อ- มูล.....	36
3.3.2 การออกแบบขั้นตอนการทำงานของมิดเดิลแวร์สำหรับการทำการค้นหา ข้อมูล.....	37
3.3.2.1 ซีควเอนซ์ไดอะแกรมของการค้นหาข้อมูล.....	38
บทที่ 4 การพัฒนาและการทดสอบ.....	39
4.1 การพัฒนาโปรแกรมเพื่อนำไปทดสอบกับมิดเดิลแวร์.....	39
4.1.1 แผนผังแสดงขั้นตอนการพัฒนามิดเดิลแวร์.....	39
4.2 รายละเอียดข้อมูลต่างๆ ที่จะทำการจัดเก็บในฐานข้อมูล.....	40
4.2.1 การออกแบบกรณีศึกษาในรูปแบบคำสั่งต่างๆ.....	40
4.2.1.1 การเพิ่มเติมข้อมูล.....	40
4.2.1.2 การค้นหาข้อมูล.....	40
4.2.1.3 การลบข้อมูล.....	40
4.2.1.4 การเปลี่ยนแปลงข้อมูล.....	40
4.2.2 รายละเอียดข้อมูลโดยแสดงในอ็อบเจกต์ไดอะแกรม.....	42
4.2.3 ตารางรายละเอียดข้อมูลต่างๆ ในฐานข้อมูล.....	43
4.3 การทดสอบการทำงานของโปรแกรม.....	46
4.3.1 การทดสอบการเพิ่มเติมข้อมูล.....	46
4.3.2 การทดสอบการค้นหาข้อมูล.....	50
4.3.3 การทดสอบการเปลี่ยนแปลงข้อมูล.....	53
4.3.4 การทดสอบการลบข้อมูล.....	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปผลการพัฒนาโปรแกรม.....	59
5.1 การออกแบบและผลที่ได้จากการพัฒนาโปรแกรม.....	59
5.1.1 การออกแบบและพัฒนาโปรแกรม.....	59
5.1.2 คุณสมบัติของโปรแกรม.....	59
5.2 ข้อจำกัดของโปรแกรม.....	60
5.3 ข้อเสนอแนะในการพัฒนาโปรแกรม.....	60
บรรณานุกรม.....	61
ภาคผนวก ก. การติดตั้งระบบ.....	62
ภาคผนวก ข. การติดตั้งค่าสำหรับการทำงาน.....	82



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงหน้าที่ของส่วนประกอบในมิดเดิลแวร์.....	33
4.2.2.1 แสดงรายละเอียดตารางอาจารย์.....	43
4.2.2.2 แสดงรายละเอียดตารางผลงานวิจัยของอาจารย์.....	43
4.2.2.3 แสดงรายละเอียดตารางอาจารย์-สาขาที่เชี่ยวชาญและสนใจ.....	44
4.2.2.4 แสดงรายละเอียดตารางสาขาที่เชี่ยวชาญและสนใจ.....	44
4.2.2.5 แสดงรายละเอียดตารางหลักสูตร.....	44
4.2.2.6 แสดงรายละเอียดตารางสาขาวิชา.....	45
4.2.2.7 แสดงรายละเอียดตารางภาควิชา.....	45
4.2.2.8 แสดงรายละเอียดตารางคณะ.....	45
4.2.2.9 แสดงรายละเอียดตารางวิชา.....	45



สารบัญภาพ

ภาพที่	หน้า
2.2.3 แผนผังแสดงชนิดของข้อมูลในเอสคิวแอล1999.....	9
2.3.1.1 ตัวอย่างเอกสารอิเล็กทรอนิกส์เอ็มแอลที่ถูกแปลงเป็นเทเบิล.....	10
2.3.1.2 โครงสร้างของเอกสารอิเล็กทรอนิกส์เอ็มแอลที่ถูกแปลงเป็นเทเบิล.....	11
2.3.2.1 ตัวอย่างของการแมปข้อมูลระหว่างเอกสารอิเล็กทรอนิกส์เอ็มแอล ตัวอย่างอบเจกต์ และตาราง.....	12
2.3.2.2 ตัวอย่างของการแมปข้อมูลระหว่างดีทีดีคลาสและตารางข้อมูล.....	12
2.3.2.3 ตัวอย่างของเอกสารอิเล็กทรอนิกส์เอ็มแอลที่มีความซับซ้อน.....	13
2.3.2.4 ตัวอย่างของเอกสารอิเล็กทรอนิกส์เอ็มแอลที่มีความซับซ้อนซึ่งถูกแปลงไปยังตัวอย่างอบเจกต์.....	13
2.3.2.5 ตัวอย่างตัวอย่างอบเจกต์ต่างๆ ซึ่งถูกแปลงให้อยู่ในรูปของตาราง.....	14
2.4 การดำเนินการประมวลผลในสถาปัตยกรรมซอฟต์แวร์ไคลเอนต์/เซิร์ฟเวอร์.....	17
2.4.1.1 สเตนด์ออลไคลเอนต์/เซิร์ฟเวอร์.....	17
2.4.1.2 ดีพาร์ทเมนต์ไคลเอนต์/เซิร์ฟเวอร์.....	18
2.4.1.3 เอนเตอร์ไพรส์ไคลเอนต์/เซิร์ฟเวอร์.....	18
3.2.1 แสดงไดอะแกรมส่วนประกอบต่างๆ ของมิดเดิลแวร์.....	34
3.2.2 การออกแบบคอมโพเนนต์ไดอะแกรม.....	35
3.3.1.1 ซีควเอนซ์ไดอะแกรมของการเพิ่มเติม เปลี่ยนแปลง และลบข้อมูล.....	36
3.3.2.1 ซีควเอนซ์ไดอะแกรมของการค้นหา.....	38
4.1.1 โพลีชาร์ตขั้นตอนการพัฒนาส่วนต่างๆ เพื่อทดสอบการทำงานของมิดเดิลแวร์.....	39
4.2 แสดงไดอะแกรมส่วนประกอบต่างๆ ของมิดเดิลแวร์.....	41
4.2.2 ตารางอีอาร์ไดอะแกรม.....	42
4.3.1.1 หน้าจอสำหรับเลือกรายการที่ต้องการกระทำ.....	46
4.3.1.2 หน้าจอสำหรับเลือกรูปแบบข้อมูลที่ต้องการเพิ่มเติมข้อมูล.....	47
4.3.1.3 หน้าจอสำหรับเพิ่มเติมข้อมูลภาควิชา.....	48
4.3.1.4 หน้าจอสำหรับกรอกค่าให้แก่คลาสที่มีแอตทริบิวต์ขึ้นกับคลาสที่ต้องการเพิ่มเติมข้อมูล.....	49
4.3.1.5 หน้าจอสำหรับเพิ่มเติมข้อมูลภาควิชา.....	49
4.3.2.1 หน้าจอสำหรับเลือกรายการที่ต้องการกระทำ.....	50
4.3.2.2 หน้าจอสำหรับค้นหาข้อมูล.....	51
4.3.2.3 หน้าจอแสดงผลการค้นหาข้อมูล.....	51
4.3.2.4 หน้าจอแสดงผลแอตทริบิวต์ขึ้นต่างๆ ของคลาสที่ทำการค้นหา.....	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

ภาพที่	หน้า
4.3.3.1 หน้าจอสำหรับเลือกรายการที่ต้องการกระทำ.....	53
4.3.3.2 หน้าจอสำหรับค้นหาข้อมูลเพื่อทำการเปลี่ยนแปลงข้อมูล.....	54
4.3.3.3 หน้าจอแสดงผลการค้นหาข้อมูลเพื่อทำการเปลี่ยนแปลงข้อมูล.....	55
4.3.3.4 หน้าจอแสดงผลแอสโซซิเอชันต่างๆ ของคลาสที่ทำการค้นหาเพื่อเปลี่ยนแปลงข้อมูล....	55
4.3.3.5 หน้าจอแสดงผลการค้นหาข้อมูลที่ต้องการเปลี่ยนแปลงข้อมูล.....	56
4.3.3.6 หน้าจอสำหรับบรรยายละเอียดที่ต้องการเปลี่ยนแปลงข้อมูล.....	56
4.3.3.7 หน้าจอสำหรับกรอกค่าให้แก่คลาสที่มีแอสโซซิเอชันกับคลาสที่ต้องการเปลี่ยนแปลง....	57
4.3.3.8 หน้าจอสำหรับบรรยายละเอียดที่ต้องการเปลี่ยนแปลงข้อมูล.....	57
4.3.4.1 หน้าจอสำหรับการลบข้อมูล.....	58
1-ก หน้าจอเริ่มต้นในการติดตั้งอปาเซ่ทอมแคท.....	63
2-ก การเลือกซอฟต์แวร์ทอมแคทเพื่อติดตั้งระบบ.....	64
3-ก การเลือกสถานที่สำหรับการติดตั้งระบบอปาเซ่ทอมแคท.....	65
4-ก หน้าจอแสดงสัญญาที่ผู้ใช้ต้องยอมรับก่อนทำการติดตั้งจาวาทูซอฟต์แวร์ดีเวลอปमेंท์คิท.....	66
5-ก หน้าจอสำหรับเลือกโปรแกรมที่ต้องการทำการติดตั้งจาวาทูซอฟต์แวร์ดีเวลอปमेंท์คิท.....	67
6-ก หน้าจอสำหรับเลือกเบเวียร์ที่ต้องการสำหรับการติดตั้งจาวาทูซอฟต์แวร์ดีเวลอปमेंท์- คิท.....	68
7-ก หน้าจอแสดงผลการติดตั้งจาวาทูซอฟต์แวร์ดีเวลอปमेंท์คิท.....	69
8-ก หน้าจอต้อนรับเข้าสู่การติดตั้งออราเคิล.....	70
9-ก หน้าจอเลือกสถานที่สำหรับติดตั้งโปรแกรมออราเคิล.....	71
10-ก หน้าจอสำหรับเลือกรูปแบบผลิตภัณฑ์ของออราเคิลที่ต้องการ.....	72
11-ก หน้าจอสำหรับเลือกชนิดของออราเคิลที่ต้องการ.....	73
12-ก หน้าจอสำหรับเลือกรูปแบบของฐานข้อมูลออราเคิลสำหรับร้องขอความต้องการ.....	74
13-ก หน้าจอสำหรับใส่หมายเลขของพอร์ตสำหรับออราเคิล.....	75
14-ก หน้าจอสำหรับกำหนดชื่อและเอสไอดีสำหรับฐานข้อมูลออราเคิล.....	76
15-ก หน้าจอสำหรับกำหนดสถานที่ของไฟล์ฐานข้อมูลออราเคิล.....	77
16-ก หน้าจอสำหรับกำหนดกลุ่มของภาษาที่สามารถทำการจัดเก็บลงฐานข้อมูลออราเคิล.....	78
17-ก หน้าจอแสดงผลสรุปของส่วนต่างๆ ของออราเคิลที่ต้องการติดตั้ง.....	79
18-ก หน้าจอสำหรับสร้างระบบป้องกันของระบบออราเคิล.....	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

ภาพที่

หน้า

19-ก หน้าจอแสดงผลการติดตั้งออราเคิล.....81



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ปัจจุบันระบบฐานข้อมูลเชิงสัมพันธ์และซอฟต์แวร์ต่างๆถูกนำมาใช้อย่างกว้างขวาง เช่น ในเชิงธุรกิจ เพื่อช่วยสนับสนุนให้ธุรกิจสามารถดำเนินไปได้อย่างมีประสิทธิภาพ และในปัจจุบัน การพัฒนาซอฟต์แวร์จะเน้นพัฒนาโปรแกรมในลักษณะเชิงวัตถุมากขึ้น เนื่องจากจุดเด่นของการพัฒนาเชิงวัตถุเน้นอยู่ที่การพัฒนาให้ออบเจกต์ต่างๆ ซึ่งประกอบไปด้วย Data และ Method ทำงานร่วมกัน ในขณะที่ตัวฐานข้อมูลเองเริ่มมีการสนับสนุนการเก็บออบเจกต์ลงในฐานข้อมูลได้โดยตรง แต่การพัฒนาโปรแกรมที่ติดต่อกับฐานข้อมูลนั้นใช้วิธีการส่ง message ที่เป็น SQL message เป็นหลัก ซึ่งทำให้เป็นภาระแก่ผู้พัฒนาโปรแกรม อีกทั้งยังก่อให้เกิดความผิดพลาด ซึ่งการที่เราติดต่อกับฐานข้อมูลโดยวิธี message passing ที่เป็นรูปแบบของ SQL message นั้นทำให้เราได้รับผลกระทบจากความผิดพลาดและปัญหาที่เกิดขึ้น อีกทั้งเนื่องจากระบบฐานข้อมูลก็มี interface ในการติดต่อกับโปรแกรม และกลไกภายในที่แตกต่างกันออกไปตามแต่ละผู้ผลิต ทำให้เกิดความยุ่งยาก และไม่สะดวกในการติดต่อกับฐานข้อมูล จากปัญหาดังกล่าวการพัฒนา Middleware ตัวหนึ่งทำงานบนเครือข่ายในรูปแบบของ Client-Server โดยมี Middleware ซึ่งเป็น software components ตัวหนึ่งที่มีหน้าที่เป็นส่วนที่เชื่อมต่อระหว่าง functions ของเครือข่ายและระบบฐานข้อมูล โดยที่เราต้องการจะใช้วิธีการส่งโครงสร้างของ Object และข้อมูลไปแทนที่จะส่งเป็นคำสั่ง SQL โดยจะใช้ XML หรือ Extensible Markup Language ช่วยในการส่งผ่านข้อมูล ซึ่ง XML จะสามารถบอกหน้าที่และประเภทของข้อมูลของส่วนต่างๆ หรือเป็นการบอกโครงสร้างในเอกสารนั้น การติดต่อและทำงานกับฐานข้อมูล Oracle9i โดยในการติดต่อและทำงานกับฐานข้อมูลนั้นจะทำผ่าน Hibernate ซึ่งมีความยืดหยุ่นและรองรับการเข้าถึงในหลายวิธี ซึ่งการพัฒนาวิธีการส่งผ่านข้อมูลดังกล่าว อาจจะสามารถทำให้การส่งผ่านข้อมูลทำได้ง่าย และสะดวกสบายขึ้น และเป็นการเพิ่มประสิทธิภาพให้แก่ระบบงาน

1.2 วัตถุประสงค์ของการทำปัญหาพิเศษ

1. เพื่อพัฒนามิดเดิลแวร์ที่เป็นตัวกลางในการติดต่อระหว่างโปรแกรมที่ใช้ภาษา Java กับฐานข้อมูล โดยใช้การส่งข้อมูลแบบออบเจกต์พาสซิง
2. เพื่อลดขั้นตอนการพัฒนาโปรแกรมในการติดต่อกับฐานข้อมูล
3. เพื่อลดขั้นตอนในการปรับปรุงโปรแกรมในภายหลัง เนื่องจากการเปลี่ยนแปลงใน Storage Layer ในอนาคต ซึ่งอาจเปลี่ยนเป็นฐานข้อมูลแบบ OODBMS แทนการใช้ RDBMS

1.3 ขอบเขตของปัญหา

1. ออกแบบพัฒนาโปรแกรมในฝั่งเซิร์ฟเวอร์ ซึ่งสามารถรองรับการส่งผ่านข้อมูลแบบออบเจกต์พาสซิงได้
2. ออกแบบพัฒนาโปรแกรมฝั่งไคลเอนท์ ซึ่งทำหน้าที่เป็นโปรแกรมทดสอบการทำงานของมิดเดิลแวร์
3. ออกแบบพัฒนาการติดต่อและทำงานกับฐานข้อมูล Oracle9i โดยในการติดต่อและทำงานกับฐานข้อมูลนั้นจะผ่าน Hibernate โดยจำกัดการทำงานกับฐานข้อมูลเพียงบางคำสั่งคือ Insert Delete Update และ Select เท่านั้น และคำสั่งเหล่านี้จะรองรับการทำงานในขั้นพื้นฐานเท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้มิดเดิลแวร์ที่เป็นตัวกลางติดต่อกับฐานข้อมูลที่พัฒนา โดยใช้วิธีการออบเจกต์พาสซิง
2. ช่วยลดขั้นตอนในการติดต่อกับฐานข้อมูล โดยใช้รูปแบบการติดต่อเชิงวัตถุ
3. ช่วยลดขั้นตอนในการปรับปรุง หรือแก้ไขโปรแกรมได้ง่ายขึ้นในภายหลัง เนื่องมาจากแนวโน้มในอนาคตที่จะมีการพัฒนาระบบการจัดการใน Storage Layer โดยอาจเปลี่ยนมาใช้ OODBMS แทนระบบการจัดการฐานข้อมูลแบบอื่นๆ

1.5 ขั้นตอนในการดำเนินงาน

1. ศึกษาวิธีการทำงานของระบบฐานข้อมูลเชิงวัตถุและวิธีการติดต่อกับฐานข้อมูลโดยใช้วิธีการเชิงวัตถุแทนที่เป็นการติดต่อโดยใช้ SQL ธรรมดา
2. ศึกษากลไกและวิธีการทำ Object passing ตลอดจน protocol และ technology ที่เกี่ยวข้อง และเลือกใช้รูปแบบที่เหมาะสมที่สุดในการพัฒนา
3. เก็บรวบรวมเอกสารข้อมูลต่างๆ โดยนำเอาเอกสารและข้อมูลที่เกี่ยวข้องมารวบรวมและใช้ประกอบการทำงาน
4. ออกแบบกรณีศึกษาในรูปแบบคำสั่งต่างๆ พร้อมทั้งกำหนดรูปแบบและตัวแปรต่างๆ ของข้อมูลที่จะจัดเก็บในฐานข้อมูล และส่วนประกอบต่างๆ ของ Middleware
5. พัฒนาระบบงานตามทีออกแบบไว้ในขั้นตอนการออกแบบ
6. ขั้นตอนการทดสอบโปรแกรมและแก้ไขโปรแกรม Middleware เพื่อทดสอบการทำงาน และแก้ไขปรับปรุงข้อผิดพลาด
7. จัดทำเอกสารคู่มือการใช้งานของ Middleware และเอกสารอ้างอิงในการศึกษาเพื่อทำปัญหาพิเศษ

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1.6.1 เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server 1 เครื่อง

- CPU Pentium4 1.8 Ghz. ขึ้นไป
- Ram 256 Mb ขึ้นไป
- Harddisk 40 GB ขึ้นไป

1.6.1.1 Software ที่ใช้ในเครื่อง Server

1.6.1.1.1 ฐานข้อมูล Oracle9i

1.6.1.1.2 ระบบปฏิบัติการ Window2000 Server

1.6.1.1.3 Apache Tomcat

1.6.2 เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client 1 เครื่อง

- spec เดียวกันกับเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server

1.6.2.1 Software ที่ใช้ในเครื่อง Client

1.6.2.1.1 J2EE ,J2SE

1.7 งบประมาณที่ใช้ในการทำปัญหาพิเศษ

ค่าใช้จ่ายในการจัดทำปัญหาพิเศษ 4000 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎี

2.1 ความรู้ทั่วไปเกี่ยวกับระบบจัดการฐานข้อมูลเชิงวัตถุ (OODBMS)

Object-Oriented Database Management System หรือ OODBMS คือ ระบบฐานข้อมูลที่เก็บ Object ตามหลักการคิด และมุมมองข้อมูลในรูปแบบเชิงวัตถุ การพัฒนาโปรแกรมส่วนใหญ่ในยุคปัจจุบัน ล้วนแต่ยึดหลักการโปรแกรมเชิงวัตถุมากขึ้น ทำให้เริ่มมีการพัฒนาระบบฐานข้อมูลเชิงวัตถุ (Object-Oriented Database หรือ OODB) ซึ่งจะสามารถรองรับการใช้งานข้อมูลที่มีความซับซ้อนได้มากกว่าฐานข้อมูลเชิงสัมพันธ์ (Relational Database System) ออบเจกต์ (Object) ที่เราสร้างในหน่วยความจำ เราสามารถนำไปเก็บใน OODBMS ได้โดยความสามารถที่เรียกว่า Persistent หรือ ความสามารถในการเอาออบเจกต์ที่มีอยู่แล้วนำไปใช้ในการดำเนินการอื่น ๆ ได้ซ้ำอีก

การเขียนโปรแกรมเชิงวัตถุแตกต่างจากการเขียนโปรแกรมแบบดั้งเดิมที่มีลักษณะโปรแกรมที่เป็นโครงสร้าง (Structure Programming) ในส่วนของข้อมูลกับโปรแกรมแยกอยู่กันคนละไฟล์ มาเป็นการกำหนดให้ออบเจกต์เป็นศูนย์รวมของข้อมูลและเมธอด โดยมีคลาสทำหน้าที่เป็นตัวกำหนดคุณสมบัติของออบเจกต์ ซึ่งเป็นเครื่องมือสำคัญสำหรับการพัฒนาโปรแกรมต้นแบบ (Prototyping) รวมถึงการนำออบเจกต์ที่ต้องการกลับมาใช้งานอีกครั้ง

2.1.1 คุณสมบัติพื้นฐานของออบเจกต์

พื้นฐานของออบเจกต์ คือ การนำเอาโครงสร้างข้อมูลมารวมกับการประมวลผลข้อมูลที่สามารถกระทำบนโครงสร้างข้อมูลนั้น ทำให้สามารถอธิบายถึงคุณสมบัติและพฤติกรรมของออบเจกต์ได้ เราเรียกโครงสร้างของออบเจกต์นี้ว่าพรอพเพอร์ตี้ (Properties) และเรียกการกระทำกับข้อมูลนี้ว่าเมธอด (Method) โดยออบเจกต์แต่ละตัวจะมีคุณสมบัติ ดังนี้

1. Encapsulation หมายถึง การแทนคุณสมบัติที่สำคัญของบางสิ่งบางอย่างโดยตัดรายละเอียดที่ไม่จำเป็นออกไป ในทางกรโปรแกรมนี้แล้ว มีความหมายว่าออบเจกต์นั้นรวมข้อมูลและกระบวนการที่เกี่ยวข้องเข้าด้วยกันเพื่อทำหน้าที่เป็นตัวแทนของออบเจกต์นั้น มีกลุ่มของแอททริบิวต์ที่รวมอยู่ด้วยกัน ที่เรียกว่า Instance variable หรือ Class variable นอกจากนี้ยังมีการกำหนดเมธอดที่สามารถนำมาใช้งานร่วมกันได้อีกด้วย โดยปกติแล้วไม่สามารถเข้าถึงแอททริบิวต์ของออบเจกต์ได้จากภายนอก ยกเว้นการใช้เมธอดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Inheritance เป็นการสืบทอด หรือกระบวนการในการจัดการคลาส เป็นการสร้างส่วนของคลาสใหม่ขึ้นมาโดยมีพื้นฐานมาจากคลาสเดิม แต่มีข้อมูลหรือฟังก์ชันพิเศษเป็นของตัวเองเพิ่มขึ้นมา ทำให้เกิดการแตกสายของออบเจกต์เป็นออบเจกต์ใหม่ที่มีคุณสมบัติของออบเจกต์เดิม

3. Polymorphism เป็นการจัดการกับออบเจกต์ต่างชนิดกันด้วยเมธอดเดียว

2.1.2 ความรู้ทั่วไปเกี่ยวกับ DBMS

ระบบจัดการฐานข้อมูล หรือ DBMS คือ ซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล ซึ่งมีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่าย สะดวกและมีประสิทธิภาพ ซึ่งช่วยในด้านของการเข้าถึงข้อมูลโดยผู้ใช้ ซึ่งอาจจะเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการค้นหาข้อมูลจากฐานข้อมูล การกระทำต่าง ๆ นั้นระบบจัดการฐานข้อมูลจะจัดการให้ โดยที่ผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูลเลย

2.1.3 โครงสร้างข้อมูลแบบ Object-Oriented Database

การใช้งานข้อมูลในฐานข้อมูลเชิงวัตถุจะอยู่ในรูปแบบของ Class ซึ่งเป็นแนวความคิดพื้นฐานของทฤษฎีทาง Object-Oriented โดยที่ใน Class จะประกอบด้วย Attribute และ Method รวมไปถึง Integrity Constrain ของ Class นั้นๆ ซึ่งจะมีการกำหนด Object Identifier หรือ OID เอาไว้ เพื่อแบ่งแยกความแตกต่างระหว่าง Class นอกจากนี้ยังสนับสนุนการทำงานแบบ Encapsulation และ Inheritance ด้วย สำหรับชนิดของข้อมูลนั้นสามารถรองรับชนิดข้อมูลแบบ Abstract Data Type ได้

เนื่องจากฐานข้อมูลเชิงวัตถุนี้เกิดขึ้นจากการรวมเอาพื้นฐานแนวความคิดเชิงวัตถุกับหลักการด้านภาษาสำหรับการโปรแกรมเชิงวัตถุ และความสามารถด้านฐานข้อมูลเข้าด้วยกัน ผลลัพธ์ที่ได้คือทำให้เกิดความเข้ากันได้เป็นอย่างดีระหว่างโครงสร้างข้อมูลภายในฐานข้อมูล กับโครงสร้างข้อมูลที่ใช้ในการพัฒนาโปรแกรม ซึ่งอำนวยความสะดวกในการใช้ภาษาสำหรับการโปรแกรมเชิงวัตถุต่างๆ เพื่อพัฒนาโปรแกรมที่ใช้งานร่วมกับฐานข้อมูลเชิงวัตถุ ทำให้สามารถลดจำนวน Code ที่ใช้ในการเขียนโปรแกรมต่างๆลงได้

2.1.4 คุณสมบัติต่างๆ ของ OODBMS

OODBMS ควรเป็นระบบการจัดการฐานข้อมูลและเป็นระบบ Object-Oriented ในเวลาเดียวกัน ดังนั้นคุณสมบัติย่อยของระบบจัดการฐานข้อมูลเชิงวัตถุ ได้แก่

2.2.1 Persistence คือ สามารถนำข้อมูลที่ผ่านการ execute จาก process หนึ่งมาแล้ว และยังสามารถนำข้อมูลนั้นกลับมาใช้ได้อีกใน process อื่นๆ

2.2.2 Concurrency คือสามารถรองรับการให้สิทธิ์แก่ผู้ใช้ในการเข้าถึงข้อมูลได้พร้อมกันหลายๆ คน

2.2.3 Recovery คือ สามารถกู้ข้อมูลคืนได้ เมื่อเกิดความผิดพลาดขึ้นกับระบบฐานข้อมูล

2.2.4 Secondary Storage Management เป็นกลไกการจัดการที่ทำหน้าที่เป็นหน่วยความจำชั่วคราว ทำให้สามารถปรับปรุงประสิทธิภาพของระบบให้ดีขึ้น เช่น การสร้าง Index เป็นต้น

2.2.5 Ad hoc Query Mechanism คือ กลไกการ query เฉพาะทาง ซึ่ง OODBMS ต้องจัดเตรียมไว้ ซึ่งอาจจะเป็น ภาษาสำหรับการ query ภาษาการโปรแกรมที่สามารถรองรับการเพิ่มเติมข้อมูลและการดำเนินการได้ในภายหลัง หรือ เครื่องมือที่เกี่ยวกับ graphic บางตัว

2.2.6 Composite Object ส่วนประกอบของ object ซึ่งสามารถสร้างได้จาก type ที่มีอยู่ในภาษาการโปรแกรมทั้งหมดใน object ทั่วไป เช่น integers characters string boolean เป็นต้น ซึ่งจะอยู่ในโครงสร้างที่ประยุกต์ใช้ เช่น list set bag array tuple เป็นต้น โดยโครงสร้างที่ OODBMS ต้องมีในระบบ ได้แก่ Set เพื่อแสดงการจัดเก็บแบบไม่เรียงลำดับข้อมูล List เพื่อแสดงการจัดเก็บแบบเรียงลำดับข้อมูล และ Tuple เพื่อแสดงการเก็บข้อมูลในแต่ละแถวข้อมูล

2.2.7 Object-identity คือลักษณะเด่นของ object ที่แสดงความแตกต่างจาก object อื่นๆ

2.2.8 Encapsulation เป็นกลไกการรวมส่วนโปรแกรมและข้อมูลเข้าไว้ด้วยกันใน object

2.2.9 Inheritance คือ การสืบทอดคุณสมบัติทั้ง method และ attribute ของ superclass ไปยัง subclass โดย subclass สามารถเพิ่มคุณสมบัติบางอย่างเข้าไปได้ รวมทั้งสามารถเปลี่ยนการทำงานของ method ที่กำหนดไว้ใน superclass ได้

2.2.10 Late Binding คือ code ของ method ควรถูก execute อยู่ในเขตจำกัดของ run-time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.11 Extensibility คือ ความสามารถในการเพิ่มเติมข้อมูลและการดำเนินการ
ได้ในภายหลัง

2.2.12 Computationally complete database language คือ ภาษาของ
ฐานข้อมูล ซึ่งได้แก่ Definition language และ Manipulate language ต้องมีความสามารถใน
ด้านการคำนวณได้อย่างครบถ้วนสมบูรณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ภาษา SQL:1999

ภาษา SQL สามารถอ่านออกเสียงได้ 2 แบบ คือ SQL เอสคิวแอล หรือ Sequel ที่แผลงมาจาก Structured Query Language หรือ ภาษาที่เกี่ยวข้องกับการประมวลผลข้อมูล เป็นภาษาทางด้านฐานข้อมูล

2.2.1 คุณสมบัติของภาษา SQL

ภาษา SQL ตามมาตรฐาน ANSI มีคุณสมบัติดังนี้

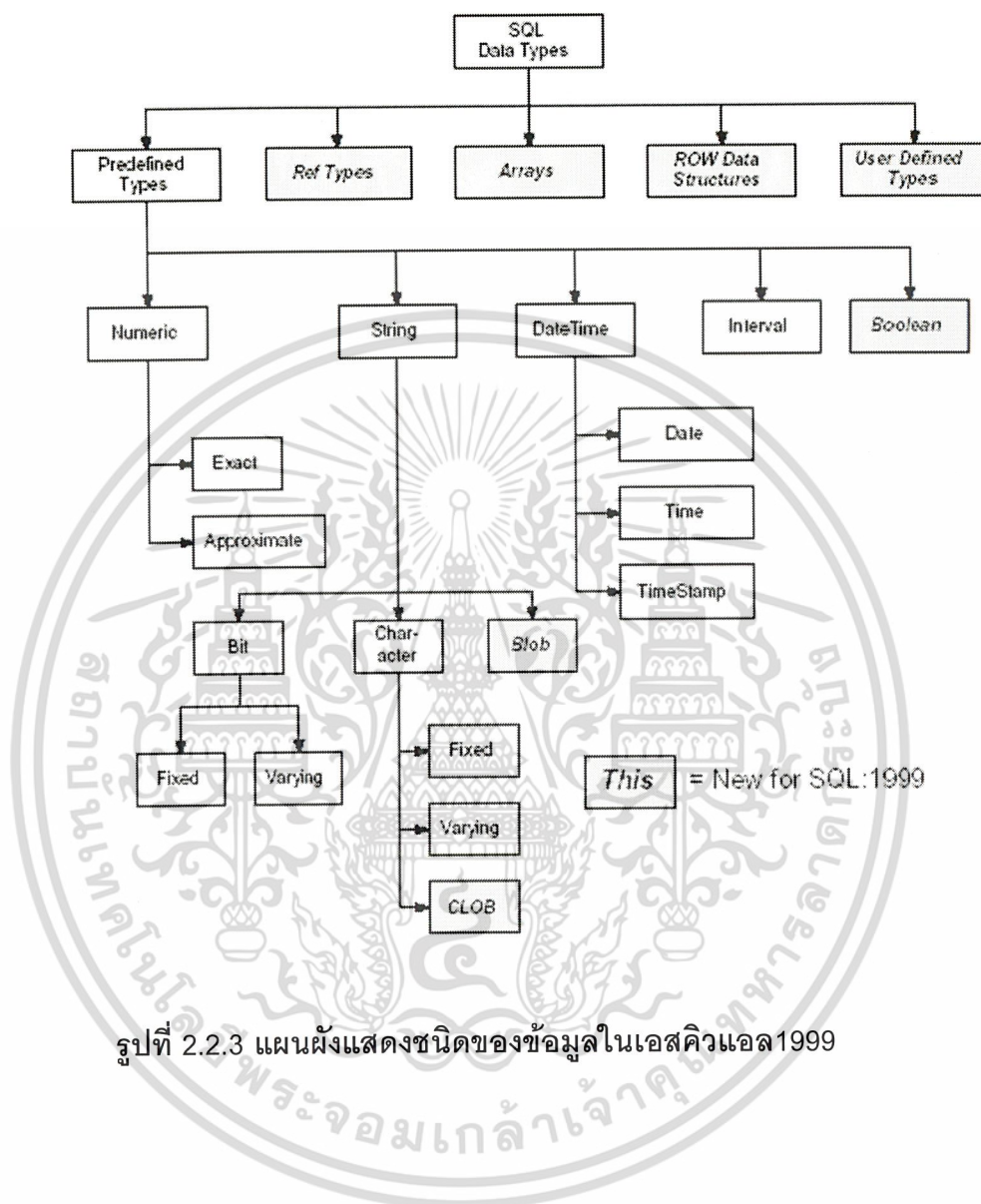
1. โครงสร้างของภาษาคัดลอกภาษาอังกฤษ สามารถเรียกดูข้อมูลที่ระบุได้ตามความต้องการ เปลี่ยนแปลง เพิ่มเติม และลบข้อมูลออกจากระบบได้
2. มีโครงสร้างไม่แน่นอน เพียงระบุความต้องการก็สามารถใช้งานได้แล้ว
3. สามารถประมวลผลข้อมูลเป็นกลุ่มได้
4. ผู้ใช้ทุกกลุ่มสามารถใช้ได้ไม่ว่าจะเป็น Database Administrator ซึ่งเรียกย่อๆ ว่า ผู้ดูแลระบบฐานข้อมูล (DBA) โปรแกรมเมอร์ (Programmer) หรือ ผู้ใช้ทั่วไป (End User)

2.2.2 คุณสมบัติของภาษา SQL:1999

จะระบุถึงรูปแบบ object model ซึ่งได้เพิ่มเติม UDTs หรือ User Defined Types ให้แก่ SQL โดย UDTs จะมี type 2 types คือ type เฉพาะ (distinct type) และ type โครงสร้าง (structured type) ซึ่ง distinct type มีพื้นฐานจาก โครงสร้างภายในของ data type เช่น INTEGER แต่ ค่าทั้งหมดไม่สามารถถูกรวมเข้าด้วยกันกับ operation ที่สร้างใน type ได้โดยตรง ส่วน structured type โครงสร้างภายในเป็น address type ที่ อาจประกอบด้วย attribute ถนน รัฐ และรหัสไปรษณีย์ structured type ถูกระบุ functions และเป็นส่วนหนึ่งในการ hierachy ด้วย type โครงสร้างอื่นๆ การ inheritance เดียวนั้น SQL:1999 รองรับแต่ mutiple inheritance ของ interface เช่นใน Java นั้นไม่ถูกรองรับ

ทั้ง distinct type และ structured type ถูกใช้ในฐานะที่เป็น data type สำหรับ column ตัวแปร SQL field ของ row หรือ เป็น attribute ของ UDT อื่นๆ object ของ java ถูกเก็บไว้ในแบบ byte stream ที่ต่อเป็นชุดๆ ที่ SQL BLOB (Binary Large Object) หรือ map เป็น row ใน table ด้วย SQL BLOB ตัว object ถูกเก็บไว้ใน cell ของ table หรือ เก็บเป็น UDT ใน cell ใน table

2.2.3 แผนผังแสดงการเพิ่มเติมชนิดของข้อมูลใน SQL:1999



รูปที่ 2.2.3 แผนผังแสดงชนิดของข้อมูลในเอสคิวแอล1999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ความรู้เกี่ยวกับการแมปข้อมูลลงฐานข้อมูล

การ map ข้อมูลมี 2 อย่างคือ

1. Table-based Mapping
2. Object-Relational Mapping

ทั้ง 2 อย่างเป็นการ map ที่ถูกใช้เป็นที่พื้นฐานสำหรับการแปลงข้อมูลระหว่างเอกสาร XML และฐานข้อมูล โดยสามารถที่จะใช้แปลงข้อมูลจากเอกสาร XML ไปยังฐานข้อมูลและแปลงจากฐานข้อมูลไปเป็นเอกสาร XML ก็ได้

2.3.1 Table-based Mapping

เป็นการ map ระหว่างเอกสาร XML กับ table จากตัวอย่างดังรูปที่ 2.3.1.1

```

<A>
  <B>
    <C>ccc</C>
    <D>ddd</D>
    <E>eee</E>
  </B>
</A>
  
```

Table A		
C	D	E
ccc	ddd	eee
fff	ggg	hhh

รูปที่ 2.3.1.1 ตัวอย่างเอกสาร XML ถูกแปลงเป็น table

จากรูปเอกสาร XML นั้นจะถูกมองเป็นตารางเดี่ยว หรือกลุ่มของตาราง โครงสร้างของเอกสาร XML จะต้องประกอบไปด้วย

```

<Table>
  <Row>
    <Column_1>...</Column_1>
    ...
    <Column_n>...</Column_n>
  </Row>
  ...
  <Row>
    <Column_1>...</Column_1>
    ...
    <Column_n>...</Column_n>
  </Row>
</Table>

```

or

```

<Tables>
  <Table_1>
    <Row>
      <Column_1>...</Column_1>
      ...
      <Column_n>...</Column_n>
    </Row>
    ...
  </Table_1>
  ...
  <Table_n>
    <Row>
      <Column_1>...</Column_1>
      ...
      <Column_m>...</Column_m>
    </Row>
    ...
  </Table_n>
</Tables>

```

รูปที่ 2.3.1.2 โครงสร้างของเอกสาร XML ที่จะถูกแปลงเป็น table

ข้อดีของการ map ข้อมูลแบบนี้คือ มันมีรูปแบบง่ายๆ เพราะโครงสร้างของตารางและผลลัพธ์ที่เกิดกับฐานข้อมูลเชิงสัมพันธ์นั้นมีความสอดคล้องกัน ทำให้ง่ายในการเขียน code ซึ่ง code จะค่อนข้างใช้ได้ดีสำหรับการแปลงข้อมูลระหว่างฐานข้อมูลกับตารางเพียง 1 ตารางต่อ 1 ครั้ง

ข้อเสียของการ map ข้อมูลแบบนี้คือ สามารถทำงานกับเอกสาร XML ที่มีขนาดเล็กมากๆ หรือไม่ได้จัดเตรียมไว้ใช้กับโครงสร้างยากๆ

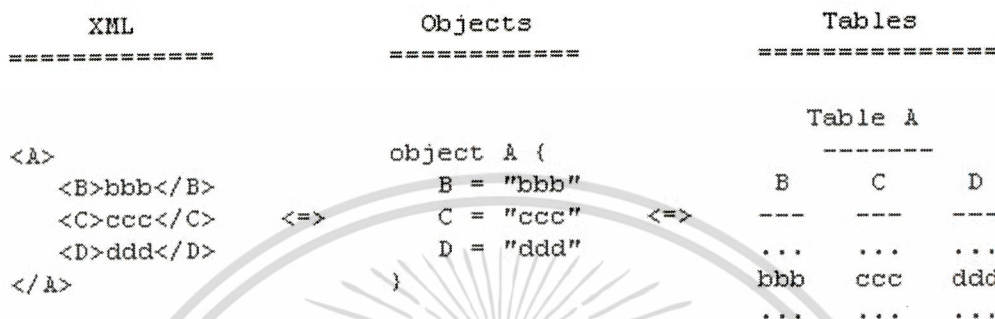
table-based mapping จะถูกใช้โดยมิดเดิลแวร์ที่แปลงข้อมูลระหว่าง

เอกสาร XML และฐานข้อมูลเชิงสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

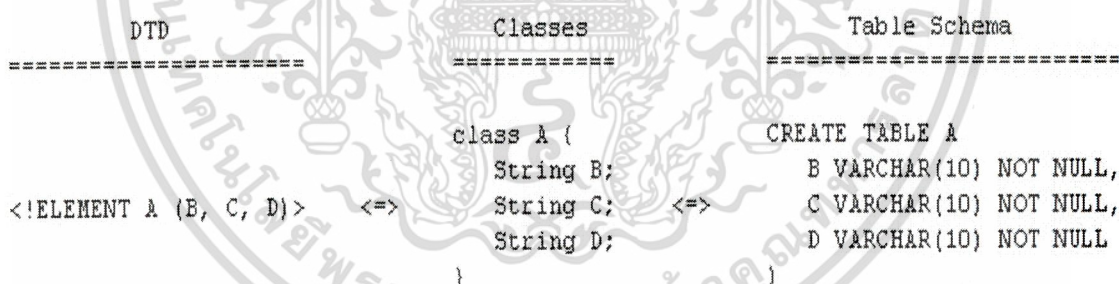
2.3.2 Object-Relational Mapping

รูปแบบการ map นี้สามารถที่จะ map ข้อมูลที่อยู่ในรูปแบบของเอกสาร XML ที่เป็น tree ของ object ต่างๆ ซึ่ง Object-Relational Mapping สามารถ map ตัว object ไปยังฐานข้อมูลได้ทั้งฐานข้อมูลเชิงสัมพันธ์ และฐานข้อมูลเชิงวัตถุ



รูปที่ 2.3.2.1 ตัวอย่างของการ Map ข้อมูลระหว่างเอกสาร XML ตัว Object และตาราง

นอกจากนี้การ map ข้อมูลยังสามารถทำในรูปแบบอื่นได้อีกนั่นคือแปลงเอกสาร DTD ไปยังฐานข้อมูลเชิงสัมพันธ์ได้ดังรูปที่ 2.3.2.2



รูปที่ 2.3.2.2 ตัวอย่างของการ Map ข้อมูลระหว่าง DTD Class และตารางข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของการ Map ข้อมูลระหว่าง DTD Class และตารางข้อมูล

```

<SalesOrder>
  <Number>1234</Number>
  <Customer>Gallagher Industries</Customer>
  <Date>29.10.00</Date>
  <Item Number="1">
    <Part>A-10</Part>
    <Quantity>12</Quantity>
    <Price>10.95</Price>
  </Item>
  <Item Number="2">
    <Part>B-43</Part>
    <Quantity>600</Quantity>
    <Price>3.99</Price>
  </Item>
</SalesOrder>

```

รูปที่ 2.3.2.3 ตัวอย่างของเอกสาร XML ที่มีความซับซ้อน

จากรูปที่ 2.3.2.3 ซึ่งเป็นตัวอย่างเอกสาร XML ที่มีความซับซ้อนจะสามารถ map ไปยังตัว object ต่างๆ ดังรูปที่ 2.3.2.4

```

object SalesOrder {
  number = 1234;
  customer = "Gallagher Industries";
  date = 29.10.00;
  items = (ptrs to Item objects);
}

object Item {
  number = 1;
  part = "A-10";
  quantity = 12;
  price = 10.95;
}

object Item {
  number = 2;
  part = "B-43";
  quantity = 600;
  price = 3.95;
}

```

รูปที่ 2.3.2.4 ตัวอย่างเอกสาร XML ที่มีความซับซ้อนซึ่งถูกแปลงไปยังตัว Object ต่างๆ

และจากตัวอย่างตัว Object ต่างๆ จะสามารถถูกแปลงให้อยู่ในรูปของตารางได้ดังรูปที่

2.3.2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SaleOrders

Number	Customer	Date
1234	Gallagher Industries	29.10.00
...
...

Items

SONumber	Item	Part	Quantity	Price
1234	1	A-10	12	10.95
1234	2	B-43	600	3.99
...

รูปที่ 2.3.2.5 ตัวอย่างตัว Object ต่างๆ ซึ่งถูกแปลงให้อยู่ในรูปของตาราง

ใน object-relational mapping มีรูปแบบในการแปลงข้อมูลจาก XML ที่เป็นเอกสาร DTD แปลงให้อยู่ในรูปตัว Object ต่างๆ แล้วจึงแปลงไปยังฐานข้อมูล หรือการ map ข้อมูลทั้ง 2 ครั้งนี้อาจสามารถทำการแปลงจากเอกสาร DTD ไปยังฐานข้อมูลได้โดยตรง โดยใช้ Software ที่ถูกสร้างขึ้นในปัจจุบันนี้ ซึ่งในที่นี้จะอธิบายเกี่ยวกับการ Map ข้อมูลโดยใช้ Hibernate

2.3.3 การแมปปิงข้อมูลโดยใช้ไฮเบอร์เนต

Hibernate คือ Software ที่ได้รับความนิยมในการ Map ข้อมูลสำหรับ Object-Relational mapping ซึ่งไฮเบอร์เนตได้จัดเตรียมการโต้ตอบกับฐานข้อมูลไว้หลายระดับ

ไฮเบอร์เนตมีความยืดหยุ่นและรองรับการเข้าถึงในหลายวิธี ไฮเบอร์เนตถูกใช้ได้ง่ายในการติดต่อกับฐานข้อมูล ซึ่งตัวไฮเบอร์เนตจะมีการจัดการการติดต่อด้วยตัวมันเอง ในกรณีที่เราจัดหาไฮเบอร์เนตที่ภายนอกฐานข้อมูลซึ่งจะไม่เพียงสร้างและจัดการการติดต่อให้เท่านั้น แต่ไฮเบอร์เนตยังจัดการเชื่อมต่อโดยการส่งไปยังฐานข้อมูลพื้นฐาน

ในการใช้ไฮเบอร์เนตจะต้องสร้าง class ของ Java ที่แสดง table ต่างๆ ในฐานข้อมูลแล้ว map ตัวแปลใน column ต่างๆ และยังสามารถเรียก method ในไฮเบอร์เนตเพื่อการ Select Insert Update และ Delete ที่บันทึกใน table ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมของไฮเบอร์เนตประกอบด้วย 3 ส่วน คือ

1. Connection management

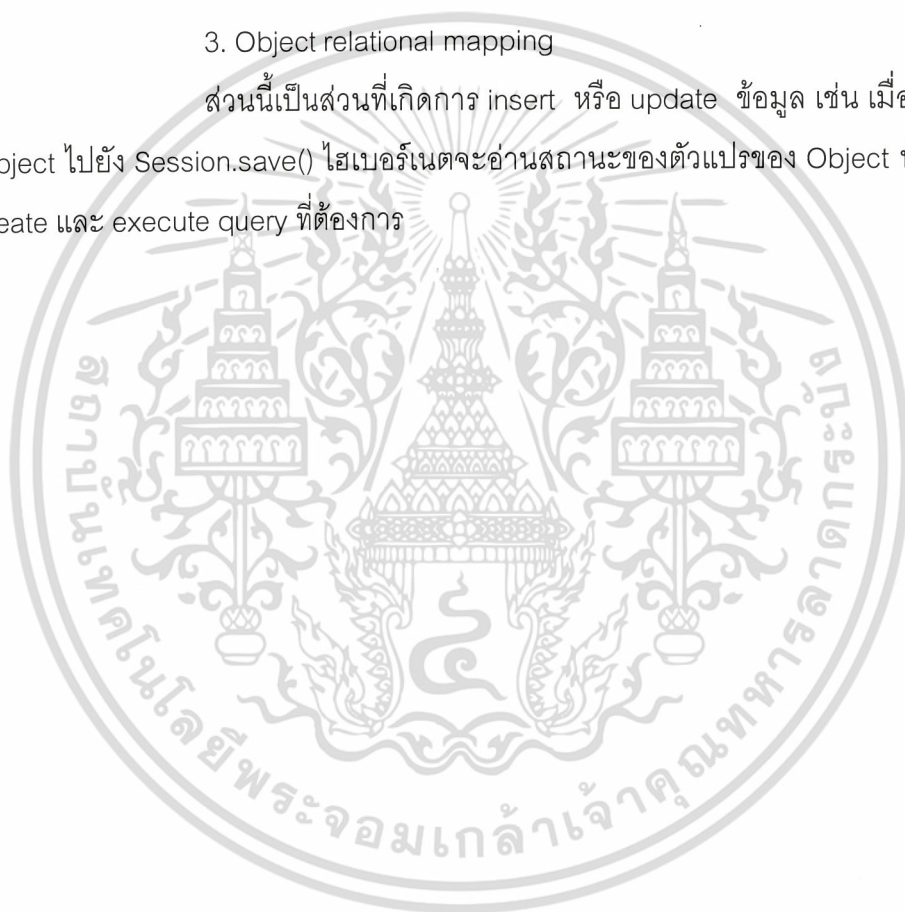
การเปิดและการปิดการติดต่อกับฐานข้อมูลเป็นส่วนที่ถูกใช้บ่อยมากที่สุด สำหรับการโต้ตอบกับฐานข้อมูล และเรายังต้องมีการติดต่อกับฐานข้อมูลอีก

2. Transaction management

Transaction ถูกใช้เมื่อต้องการที่จะประมวลผลมากกว่าหนึ่ง query ในหนึ่งครั้ง ซึ่งผลที่เกิดขึ้นภายใน Transaction อาจจะไม่สำเร็จทั้งหมดหรือล้มเหลวทั้งหมดได้อย่างไรอย่างหนึ่งก็ได้

3. Object relational mapping

ส่วนนี้เป็นส่วนที่เกิดการ insert หรือ update ข้อมูล เช่น เมื่อส่งตัวแปรของ Object ไปยัง Session.save() ไฮเบอร์เนตจะอ่านสถานะของตัวแปรของ Object นั้น แล้วทำการ create และ execute query ที่ต้องการ



2.4 ความรู้ทั่วไปเกี่ยวกับมิดเดิลแวร์

Middleware คือ ซอฟต์แวร์ประเภทหนึ่งที่ทำหน้าที่เป็นตัวกลาง เป็นตัวประสานและ บริการ Service ต่างๆ ให้กับ Application เช่น การทำ Load Balancing การทำ Transaction การทำ Resource Pooling และการทำ Message Oriented Service เป็นต้น ซึ่งในระบบงาน ขนาดใหญ่ หรือ Enterprise Application จำเป็นต้องใช้งาน Service เหล่านี้เป็นอย่างมาก ซึ่งในอดีต องค์กรต่างๆ ได้พัฒนาซอฟต์แวร์ประเภท Middleware ขึ้นมาใช้งานเอง โดยจะมี Service ต่างๆ แล้วแต่แต่ละองค์กรจะพัฒนาขึ้นมา

ในปัจจุบันความต้องการ Service ของ Middleware มีมากขึ้น เพื่อรองรับการดำเนินงาน ทางธุรกิจที่มีความสลับซับซ้อนมากขึ้น จึงต้องการระบบงานที่ยืดหยุ่นสามารถรองรับการ เปลี่ยนแปลงได้อย่างรวดเร็ว เพื่อให้สามารถแข่งขันกับคู่แข่งทางธุรกิจได้ ประกอบกับการที่จะ พัฒนา Middleware ขึ้นมาใช้เองนั้น เป็นเรื่องที่ยาก มีโอกาสที่จะล้มเหลวสูง ด้วยสาเหตุดังกล่าว จึงได้มีผู้ผลิต Middleware ออกมาจำหน่าย โดยเรียกว่า Application Server ซึ่งจะบรรจุ Service ต่างๆ ไว้ ทำให้องค์กรไม่ต้องพัฒนา Service ขึ้นใช้เอง ซึ่งจะทำให้องค์กรมุ่งคิด หรือทำเฉพาะที่เป็น Business Logic ใน Business Domain ของตัวเองเพียงอย่างเดียวเท่านั้น วิธีการก็คือพัฒนา ซอฟต์แวร์ขึ้นมา แล้วนำไป Deploy บน Application Server ที่เตรียมสถานะแวดล้อมขณะ Runtime ไว้ให้ โดยมีการประกาศความต้องการ Service ให้ Application Server เตรียม Service ต่างๆ ไว้บริการ

Middleware เป็น Software ที่ทำหน้าที่ให้มาตรฐานในการแปลงระหว่าง Software จาก ผู้ผลิตต่าง ๆ ทำหน้าที่คล้ายล้าม เช่น จัดการการส่งคำขอต่าง ๆ ระหว่าง Client กับ Server โดย มันจะเป็นตัวกลางในการรับส่ง เมื่อมีการเปลี่ยนแปลง ทั้ง Client และ Server จะไม่ได้รับ ผลกระทบใด ๆ เพราะการเปลี่ยนแปลงเกิดขึ้นที่ Middleware เช่น ODBC หรือ JDBC เป็นต้น

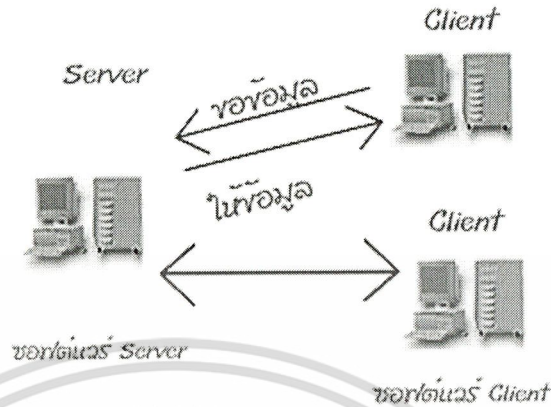
ในปัจจุบันเครื่องคอมพิวเตอร์ได้มีการเชื่อมต่อเครือข่ายในลักษณะรูปแบบของการแบ่ง พื้นที่ได้ 3 แบบ ได้แก่

1. LAN หรือ Local Area Network
2. WAN หรือ Wide Area Network
3. Internet

และได้มีการพัฒนาสถาปัตยกรรมซอฟต์แวร์ Client/Server โดยได้รับการออกแบบ แบ่งเป็น 2 ส่วน คือส่วนที่ให้บริการ หรือ Server และส่วนที่ผู้ให้บริการ หรือ Client โดยผู้ใช้ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีซอฟต์แวร์ในการติดต่อไปยังผู้ให้บริการ Server ก็จะต้องสนองโดยการดูความต้องการของผู้ใช้ว่าต้องการอะไร เมื่อดำเนินการประมวลผล ดังรูป

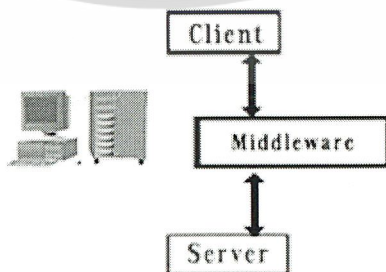


รูปที่ 2.4 การดำเนินการประมวลผลในสถาปัตยกรรมซอฟต์แวร์ Client/Server

ระบบ Client/Server ได้ถูกพัฒนาขึ้นมาเพื่อตอบสนองแนวคิดการ Downsizing เป็นการลดค่าใช้จ่ายระบบ Time Sharing ของเครื่อง Mainframe ซึ่งระบบ Client/Server เป็นระบบประมวลแบบกระจายหรือ Distributed Processing โดยจะแบ่งการประมวลผลระหว่าง Client และ Server โปรแกรมประยุกต์หรือ Application Program จะประมวลผลบางส่วนที่ Client และบางส่วนก็ประมวลที่ server

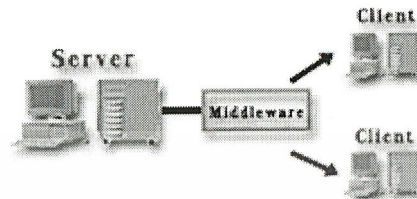
2.4.1 รูปแบบของ Client/Server

2.4.1.1 Stand alone Client/Server เป็นการที่มีผู้ใช้บริการให้อยู่ในเครื่องเดียวกัน



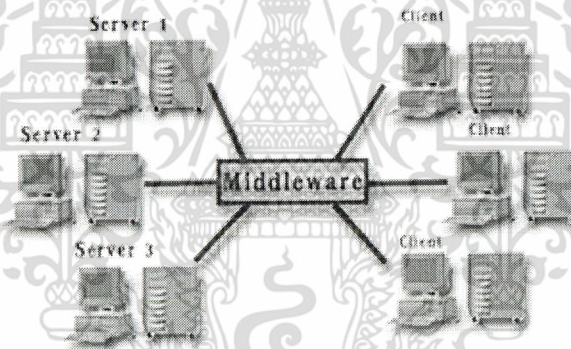
รูปที่ 2.4.1.1 Stand alone Client/Server

2.4.1.2 Department Client/Server เป็นการทำงานที่ผู้ให้บริการจะบริการฐานข้อมูล และโปรแกรมประยุกต์ต่างๆที่อยู่บน Server โดยจะเชื่อมต่อกันด้วยระบบเครือข่ายท้องถิ่นหรือ LAN และ Middleware ประสิทธิภาพการประมวลผลจะช้ากว่า Stand alone เพราะต้องผ่านเครือข่าย



รูปที่ 2.4.1.2 Department Client/Server

2.4.1.3 Enterprise Client/Server เป็นการทำงานที่เชื่อมโยงเครื่อง Server ต่าง Platform เข้าด้วยกัน สามารถใช้ทรัพยากรร่วมกันได้ โดยผ่าน Middleware



รูปที่ 2.4.1.3 Enterprise Client/server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ความรู้ทั่วไปเกี่ยวกับเจตียซี

JDBC คือ Java™ API หรือ Java Application Programming Interface คือตัวเชื่อมต่อระหว่างโปรแกรมกับฐานข้อมูล โครงสร้างของ JDBC ประกอบด้วย Class และ Interface ที่เขียนด้วยภาษา Java ที่เรียกว่าไดรเวอร์ฐานข้อมูล เพื่อใช้ในการประมวลผลคำสั่ง SQL โดยเพียงแค่วิธีเขียนโปรแกรมๆ เดียวก็สามารถประมวลผลคำสั่ง SQL ที่ใช้กับ DBMS ต่าง ๆ กัน ไม่ว่าจะเป็น Oracle InterBase หรือ Sybase

Java เป็นภาษาที่มีประสิทธิภาพ ความปลอดภัยสูง ใช้งานง่าย เข้าใจง่าย และสามารถ download ได้ เป็นภาษาที่มีประสิทธิภาพมากสำหรับการเขียนโปรแกรมติดต่อกับ database และที่สำคัญ Java สามารถติดต่อกับ database ต่างๆ กันได้

JDBC ก็มีความสามารถเหมือนกับ Java ยกตัวอย่างเช่น Java และ JDBC API ซึ่งสามารถเขียนเว็บเพจโดยใช้ applet ติดต่อและใช้ข้อมูลด้วย remote database หรือสามารถใช้ JDBC ติดต่อกับลูกค้า ที่ใช้ OS ต่างๆ กันได้โดยเราเขียนโปรแกรมเพียงโปรแกรมเดียวเท่านั้น

2.5.1 การทำงานของ JDBC Application

2.5.1.1 สร้าง connection ติดต่อกับ database

2.5.1.2 ส่งคำสั่ง SQL ไปให้ DBMS

2.5.1.3 ประมวลผลผลลัพธ์

2.5.2 ข้อดีของ JDBC Technology

2.5.2.1 JDBC Technology สามารถปรับเปลี่ยนสถาปัตยกรรมได้ สามารถติดตั้งฐานข้อมูลและการเข้าถึงข้อมูลได้ง่าย มีการจัดการระบบฐานข้อมูลที่แตกต่างกันได้

2.5.2.2 การรวมกันของ Java API และ JDBC API ทำให้การพัฒนาของ Application ง่ายและประหยัด JDBC จะซ่อนการทำงานที่ซับซ้อนของการ Access ข้อมูลซึ่งเป็นงานหนักสำหรับ Programmer และ JDBC API ยัง ง่ายต่อการเรียนรู้ และค่าใช้จ่ายในการดูแลบำรุงรักษาต่ำ

2.5.3 ลักษณะพิเศษของ JDBC Technology

2.5.3.1 JDBC API สามารถพัฒนาวิธีการใช้ที่สลับซับซ้อนให้เข้าใจได้ง่ายบนพื้นฐานของความสะดวกและความสามารถในการติดต่อกับฐานข้อมูล

2.5.3.2 JDBC technology-based driver จะดาวน์โหลดส่วนที่ใช้ในการทำงานอัตโนมัติโดยไม่ต้องมีการติดตั้งใดๆ

2.5.3.3 JDBC Technology สามารถระบุการติดต่อกับฐานข้อมูลได้ โดย JDBC API จะประกอบด้วยกระบวนการและติดต่อกับแหล่งข้อมูล โดยใช้ Datasource Object สร้างโค้ดที่ง่ายต่อการบำรุงรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.4 JDBC API ถูกรวมอยู่ใน Standard Edition หรือ J2SE™ และ Enterprise Edition หรือ J2EE™

2.5.3.5 ความต้องการ Software เพื่อการใช้ JDBC ได้แก่

- Java 2 Platform
- SQL Database
- JDBC Technology – based driver สำหรับฐานข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ความรู้ทั่วไปเกี่ยวกับ XML

Extensible Markup Language หรือ XML เป็นภาษาใหม่ซึ่งกลุ่ม World Wide Web (W3C) กำหนดให้เป็นภาษามาตรฐานสำหรับการอธิบายเอกสาร (Description Language) หรือภาษาที่ใช้สำหรับการเขียน เอกสาร markup (markup document) โดยที่เอกสาร markup นั้นมีการใช้ metadata หรือ tags เพื่อบอกหน้าที่และประเภทของข้อมูลของส่วนต่างๆของ เอกสารนั้นได้โดยชัดเจน การเพิ่ม metadata เข้าไปในเอกสารสามารถทำให้โครงสร้างของเอกสารชัดเจนขึ้น และทำให้การประมวลผลเอกสารเป็นไปโดยง่าย และไม่จำเป็นที่จะต้องอาศัยมนุษย์เพื่อตีความเอกสาร XML เป็นภาษาที่เหมาะสมกับการแลกเปลี่ยนข้อมูลผ่านเครือข่ายคอมพิวเตอร์ เนื่องจาก XML ไม่ได้ขึ้นอยู่กับโปรแกรมประยุกต์หรือระบบปฏิบัติการใด นอกจากนี้ XML ยังเป็นภาษาที่มีความยืดหยุ่น ผู้ใช้สามารถที่จะกำหนดและตั้งค่า metadata ให้เหมาะกับ เอกสารเฉพาะที่ตนต้องการได้ และยังสามารถเพิ่มเติม metadata ได้ในภายหลังโดย ไม่มีผลกระทบต่อโปรแกรมที่มีอยู่แล้วด้วย

2.6.1 ลักษณะโครงสร้างของ XML

XML เป็นการใช้ข้อความเพื่อบ่งบอกโครงสร้างของเอกสาร ยกตัวอย่างรูปแบบโครงสร้างของหนังสือ หากหนังสือประกอบด้วยจำนวนบท 2 บท ซึ่งในแต่ละบทจะประกอบด้วยเนื้อความ (Text)

ตัวอย่างคำสั่ง

```

Begin Book
  Begin Chapter 1
    Text for Chapter 1
  End Chapter 1
  Begin Chapter 2
    Text for Chapter 2
  End Chapter 2
End Book

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างลักษณะของเอกสาร XML

```
<?xmlversion="1.0"encoding="windows-874"?>
<mali>
  <malisorn>ขึ้นต้นด้วยมะลิซ้อน</malisorn>
  <malila>พอแตกใบอ่อนเป็นมะลิลา</malila>
</mali>
```

สามารถอธิบายโดยได้ ดังนี้

บรรทัดที่ 1 นั้นหมายความว่าเราประกาศเอกสารนี้เป็นเอกสาร XML และมีการเข้ารหัสอักขระเป็น windows-874 เพื่อให้ใช้ภาษาไทยได้ จากนั้นเราจะเห็น สิ่งที่เราเรียกว่า tag แต่จริงๆแล้วในภาษา XML จะแบ่งโครงสร้างเป็น 2 ส่วนใหญ่ ๆ คือ tag และ element สามารถอธิบายเพิ่มเติมได้จาก

```
<root>
  <element>
    <tag></tag>
  </element>
</root>
```

2.6.1.1 ความหมายของ Tag กับ Element

Tag สำหรับใน XML แล้ว tag มีความหมายในลักษณะเดียวกับที่ใช้ใน HTML tag คือข้อความที่อยู่ระหว่างสัญลักษณ์ < และ > ประกอบด้วย

1. Tag เปิด (Start tag) จากตัวอย่าง <book>ถูกเรียกว่า tag เปิด ดังนั้น tag เปิดจึงมีสัญลักษณ์คือ <...>

2. Tag ปิด (End Tag) tag จะถูกเรียกว่า tag ปิดก็ต่อเมื่อใน tag มีเครื่องหมาย / อยู่หลังสัญลักษณ์ < ดังนั้นลักษณะของ tag ปิดจึง มีรูปแบบคือ </...>

หากพิจารณาระหว่าง tag เปิดกับ tag ปิดแล้ว ข้อแตกต่างอีกข้อหนึ่งคือ tag เปิด เป็น tag ที่สามารถใส่ข้อมูล attribute ลงไปภายใน tag ได้ แต่ tag ปิดจะไม่ทำกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Element ในที่นี้คือ โครงสร้างหลักของ XML ซึ่งอยู่ในรูปของ tag เช่นเดียวกัน จะมีลักษณะซ้อนกันเป็นชั้นๆ โดยที่ Root element เป็น element แรกสุดของเอกสาร XML

element เป็นส่วนประกอบของ Content และ attribute หรือถ้ามองง่าย ๆ คือ element เริ่มต้นที่ tag เปิด และสิ้นสุดที่ tag ปิดใน tag คำสั่งเดียวกัน

2.6.1.2 ความหมายของ Content กับ Attribute

Content เนื้อความ หรือ Content ถือได้ว่าเป็นข้อมูลเพื่อใช้ในการแสดง หรือกล่าวอีกนัยหนึ่งคือ Content อยู่หลัง tag เปิด และจบก่อนที่จะถึง tag ปิดนั่นเอง

Attribute ถูกบรรจุอยู่ใน tag เปิด และ attribute นี้ในบางครั้ง อาจจะมีหรือไม่มีก็ได้ รวมถึงถ้ามีอาจมีได้มากกว่าหนึ่งตัวใน tag เปิด

Attribute คือข้อมูลความหมายเพิ่มเติม เช่น ถ้าพูดถึง บท ในหนังสือ แค่นี้เป็นความหมายโดยรวม แต่ถ้าบอกว่า บทที่ 1 ในหนังสือ เลข 1 ในที่นี้คือความหมายเพิ่มเติมให้กับบท

2.6.2 วัตถุประสงค์หลักของ XML

จุดประสงค์หลักของ XML คือการแยกส่วนข้อมูลเพื่อประโยชน์ในการแสดงผล นั่นคือ เอกสาร XML ใช้สำหรับควบคุมตัวอักษร (plain text) โดยยึดเอาข้อมูลที่มีใจความเหมือนกัน แต่สามารถนำไปแสดงผล ให้ผู้ใช้หรือผู้อ่านหลายคนได้รับรู้ข้อมูลใจความเดียวกัน เมื่อมองผ่านอุปกรณ์แสดงผลที่ต่างกัน ซึ่งลักษณะเหล่านี้ ไม่เกิดขึ้นจริงกับ เอกสารประเภทประมวลผลคำอื่นๆ word processing หลากหลายคอมพิวเตอร์และหลากหลายระบบปฏิบัติการ นี่ไม่ใช่สิ่งที่ควรจะได้เลย เห็นได้ชัดว่าหลายปีที่ผ่านมา คอมพิวเตอร์หลากหลายชนิดถูกผลิตออกมาเป็นจำนวนมาก เพื่อใช้งาน รวมถึงระบบปฏิบัติการ (Operation System) ก็ยังมีความแตกต่างกันออกไปด้วย นั่นคือคอมพิวเตอร์รุ่นใหม่ ๆ จึงมีความหลากหลาย และยุ่งยากในการทำให้เข้าใจข้อมูลเดียวกัน ซึ่งเปรียบแล้วเหมือนกับภาษามนุษย์ที่ใช้สื่อสารกัน มีมากมายหลายร้อยภาษา

W3C ได้กล่าวถึงจุดมุ่งหมายหลักไว้ 10 หัวข้อ ดังนี้

1. XML มีการใช้งานโดยตรงบนเครือข่าย Internet ซึ่ง XML จะถูกออกแบบมาสำหรับจัดเก็บและจัดส่งข้อมูลบนเว็บ
2. XML มีการสนับสนุนโปรแกรมที่หลากหลาย ซึ่ง XML จะถูกออกแบบมาเพื่อใช้กับโปรแกรมที่มีรูปแบบต่าง ๆ ตัวอย่างเช่น การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมทางการเงิน การเผยแพร่และปรับปรุงโปรแกรมให้ทันสมัยและการเขียน Voice Script ให้สื่อสารได้ด้วยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทรศัพท์

3. XML จะต้องเข้ากันได้กับ SGML
4. XML จะต้องง่ายต่อการเขียนโปรแกรมเพื่อประมวลผลเอกสาร
5. จำนวนของทางเลือกเฉพาะของ XML ควรมีจำนวนน้อยที่สุดหรือไม่ควรมีเลย
6. เอกสาร XML จะต้องอ่านเข้าใจง่ายและมีความชัดเจน
7. XML ออกแบบมาเพื่อให้พัฒนาโปรแกรมได้อย่างรวดเร็ว
8. การออกแบบ XML ต้องมีรูปแบบที่เหมาะสมและกะทัดรัด
9. สามารถสร้างเอกสาร XML ได้ง่าย
10. Markup ของ XML ต้องไม่รวบรัดมากเกินไป

2.6.3 ภาษาสำหรับจัดการโครงสร้างเอกสาร

XML พยายามลดข้อจำกัดของความแตกต่าง ทางด้านระบบคอมพิวเตอร์ เพื่อสร้างเป็นภาษาที่ใช้กำหนดโครงสร้างของเอกสาร ให้สามารถเข้าใจกันได้ในทุก ๆ ระบบ

XML คือภาษาที่มีลักษณะเมตา (meta language) หมายถึงรูปแบบการรวบรวมข้อมูลข่าวสาร ที่นำมาจากแหล่งข้อมูลข่าวสารอื่นๆ เช่น บทอ้างอิงในหนังสือสามารถบอกผู้อ่านได้ ถ้าถูกรวบรวมเนื้อหาไว้ด้วย XML ผู้อ่านจะสามารถรู้ว่าหัวข้อที่ตัวเองสนใจ จะไปอยู่ที่หน้าที่เท่าไรของหนังสือ ซึ่งเมื่อเปิดไปยังหน้าดังกล่าวแล้ว จะได้รับข้อมูลที่อ้างไปถึงจากหัวข้อในบทอ้างอิงนั่นเอง ดังนั้นแต่ละหัวข้อในบทอ้างอิงจึงเป็น ข้อมูล ที่ได้รวบรวมมาจากข้อมูลในเนื้อหา บทอ้างอิงจะสามารถบอกข้อมูลเกี่ยวกับ element (elements) และแอตทริบิวต์ (attributes) ซึ่งจะมีเนื้อหาต่อไปได้ เราจึงใช้ XML เป็นภาษาเมตา

2.6.4 การประยุกต์ใช้ XML

ตัวอย่างการประยุกต์ใช้งาน XML

```
<book>
  <chap>
    Text for Chapter 1
  </chap>
  <chap>
    Text for Chapter 2
  </chap>
</book>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น ไวยากรณ์ที่ถูกใช้ใน XML เมื่อเปรียบเทียบตัวอย่างนี้กับเอกสารที่เห็นในส่วนต้นของบทความ สำหรับโครงสร้างของหนังสือ โดยเปรียบเทียบทีละส่วน โดยพิจารณาที่ส่วนของ element (สังเกตกลุ่ม element จากสัญลักษณ์ <...> จนถึง </...>) ดังนั้นโครงสร้างของ XML จะมี element เป็นส่วนประกอบ ใช้กำหนดโครงสร้างของเอกสาร

ตัวอย่างคำสั่ง สำหรับปรับปรุงตัวอย่าง XML ก่อนหน้าที่มี element อยู่ โดยเพิ่มส่วนที่เรียกว่า attribute ลงใน element คือตัวเลขบอกเลขที่ของบท โดยชื่อ attribute ใช้คำว่า number และ ตามด้วยตัวเลขบอกความหมายของบท

```
<book>
  <chap number="1">
    Text for Chapter 1
  </chap>
  <chap number="2">
    Text for Chapter 2
  </chap>
</book>
```

ตัวอย่างของคำสั่ง สำหรับการกำหนดโครงสร้างสำหรับหนังสือ

```
<book>
  <chap number="1">
    Text for Chapter 1
  </chap>
  <chap number="2">
    Text for Chapter 2
  </chap>
</book>
```

จากตัวอย่างโครงสร้าง เมื่อพิจารณาจะเห็นได้ว่า มีรูปแบบของจำนวนบทอยู่ 2 บท ในแต่ละบทมีข้อความอยู่บางส่วน เช่น บทที่ 1 มีข้อความ Text for Chapter 1 ที่อยู่ระหว่าง element chap จะมี attribute ชื่อ number โดยบทที่ 1 ใช้ number="1" เป็นข้อมูลของ attribute

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโครงสร้างเอกสาร XML สามารถนำไปผ่านขบวนการ render เพื่อให้ได้เอกสารใช้งานจริงๆ ได้ในหลากหลายรูปแบบ ข้อมูลอยู่ใน attribute number มีความหมายที่จะสามารถทำให้กลไก render สามารถแยกแยะเอกสารตามรายละเอียด ของเนื้อหาในแต่ละบทได้ ดังนั้นหากต้องการให้มุมมองของการมองเห็นแตกต่างกัน ก็ย่อมทำได้เช่นเดียวกัน ยกตัวอย่างเช่น หากนำเอกสาร XML ผ่านขบวนการ render สำหรับหนังสือที่เป็นกระดาษ การจัดพิมพ์ย่อจะเกิดหน้าว่างหนึ่งหน้า ของแต่ละบทเสมอ ในขณะที่เอกสารตัวเดียวกัน หากนำไปผ่านขบวนการ render ของมอนิเตอร์แล้ว ก็ไม่จำเป็นต้องทำให้เกิดหน้าว่างก่อนบท

เอกสาร XML เพียงบอกแค่ ข้อมูลในส่วนที่เป็น Content ซึ่งถูกแยกแยะข้อมูล ออกเป็นส่วนๆ ในลักษณะเอกสารที่มีโครงสร้าง หรือกล่าวอีกนัยหนึ่งคือ XML ใช้สำหรับแยกแยะ เอกสารออกเป็นส่วนๆ เท่านั้น โดยที่ข้อมูลในเอกสาร ไม่มีส่วนที่ใช้บอกว่าควรแสดง Content ในลักษณะการมองเห็นแบบไหน

2.6.5 DTD ข้อกำหนดโครงสร้างสำหรับ XML

DTD คือเพิ่มข้อมูลหรือหลายเพิ่มข้อมูลที่ใช้งานร่วมกัน ซึ่งบรรจุข้อกำหนด และ กฎเกณฑ์ของเอกสาร ชุดข้อกำหนดเหล่านี้ สำหรับการกำหนดรูปแบบ element ตัวอย่างเช่น หาก ต้องการเอกสารที่มี element <LIST> ที่มี element <ITEM> บรรจุอยู่ใน ข้อกำหนดในเพิ่ม ข้อมูล DTD จะมีรูปแบบดังนี้

```
<!ELEMENT item (#pcdata)>
<!ELEMENT list (item)+>
```

ซึ่งอธิบายความหมายคือ element items บรรจุข้อความใดๆ และ element list บรรจุ element item อื่นๆ ดังนั้น DTD เป็นรูปแบบภาษา ซึ่งทำให้สามารถตรวจสอบเอกสาร ที่ นำเอาข้อกำหนด DTD ไปใช้ ว่าถูกจัดสร้างตามความต้องการหรือไม่ ทำให้ระบบการ rendering สามารถเข้าใจตัวเอกสารได้ดี และดึงไปใช้งานได้ถูกต้อง

การสร้าง DTD มีรายละเอียดและขั้นตอนที่ยุ่งยากพอสมควร ถึงแม้ DTD จะทำให้การสร้างเอกสารมีความยุ่งยากก็ตาม ซึ่ง XML ไม่จำเป็นต้องใช้ DTD เสมอไป แม้จะต้องใช้ DTD แต่ก็จะมีกรสร้าง DTD ที่เป็นมาตรฐานให้ใช้อยู่แล้ว เพื่อให้เกิดรูปแบบมาตรฐานที่ชัดเจน และเข้าใจข้อมูลในเอกสารประเภทเดียวกัน เอกสาร XML มีส่วนประกอบที่สำคัญอยู่สามส่วน สอง ส่วนเป็นสิ่งที่จำเป็น ในขณะที่อีกส่วนเป็นทางเลือก ที่จะมีหรือไม่มีก็ได้

1. ส่วนแรก คือส่วนของเนื้อหา หรือ Content นั้นเอง ซึ่งทำให้เอกสารมี ข้อมูลสำหรับดูโดยผู้อ่าน Content นี้เป็นได้ทั้งข้อความ รูปภาพ ส่วนนี้ถูกสร้างขึ้นมาจาก element ในที่นี้คือไฟล์ XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **ส่วนที่สอง** คือกฎเกณฑ์และข้อกำหนด โครงสร้างของเอกสาร ในที่นี้คือไฟล์ DTD ส่วนนี้ถือเป็นทางเลือก ซึ่งจะเลือกใช้หรือไม่ก็ได้

3. **ส่วนสุดท้าย** คือ StyleSheet คือ ลักษณะข้อกำหนดสำหรับการแสดงผลพอร์ทัลเอง ในที่นี้คือไฟล์ XSL

DTD คือ ขอบข่ายในขณะที่ยัง StyleSheet ไม่ใช่ เอกสาร XML สามารถถูกบังคับโครงสร้างด้วย DTD ซึ่งจะใช้หรือไม่ใช้ก็ได้ แต่ StyleSheet คือสิ่งที่จำเป็นที่จะสามารถนำข้อมูลในเอกสาร มาแสดงด้วยการ rendering StyleSheet ในบางครั้งไม่ต้องการ หากเรามีกลไก rendering ที่จัดสร้างไว้พร้อมแล้ว สำหรับเอกสาร XML ที่ต้องผ่านขบวนการแปลงด้วย XSL เพื่อให้เป็น HTML ในการแสดงผ่านบราวเซอร์ อย่างนี้ต้องจำเป็นในการใช้งาน StyleSheet

เอกสาร DTD ในการจัดสร้าง DTD ทำให้เอกสารมีความเป็นรูปแบบ well-formed เรียกว่ากันว่า เอกสารที่ถูกต้อง ในบางครั้งหากเรา ต้องการสร้างเอกสาร XML ที่สามารถแลกเปลี่ยนกับคนอื่น ๆ หรือส่วนอื่นๆ และให้สามารถเข้าใจโครงสร้างเดียวกัน นั่นก็คือเอกสาร XML ของเรา และส่วนอื่นๆ จำเป็นต้องมีรูปแบบโครงสร้างเดียวกัน นั่นก็คือควรที่จะเลือกใช้ DTD เดียวกัน เพื่อบังคับโครงสร้างให้เหมือนกัน

เอกสารที่ถูกต้อง เอกสารที่อยู่ในรูปแบบ well-formed เป็นเอกสารที่เรียกว่า ถูกต้องตามข้อกำหนดของ XML แต่เอกสารที่ไม่ถูกต้องตามรูปแบบ XML ก็สามารถเป็นเอกสารที่ดีได้เช่นกัน เนื่องจากในปัจจุบันข้อมูล บนระบบอินเทอร์เน็ตมีอยู่มากมาย ที่ไม่ได้สร้างเป็นเอกสารที่ถูกต้องตามแบบ XML นั่นคือความต้องการส่วนหนึ่ง ที่ DTD จะสามารถบังคับเอกสาร XML ที่สร้างมาจาก DTD เดียวกัน ให้เป็นเอกสารที่ถูกต้องตามแบบ XML

เอกสารรูปแบบ well-formed แนวคิดของเอกสารแบบ well-formed ถูกนำมาใช้สำหรับ XML ในกรณีที่ ไม่สามารถสร้างเอกสาร XML ด้วยข้อกำหนด DTD จากตัวอย่างของ tag ใน HTML เป็น tag ที่เรียกว่า element แปลว่า เนื่องจากว่า tag ดังกล่าว ไม่จำเป็นต้องอาศัย tag ปิด เช่นเดียวกัน หาก XML มี element ที่มี tag ปิด จึงไม่เป็นเอกสารในรูปแบบ 'well-formed' แต่เอกสาร XML ต้องเป็นเอกสารรูปแบบ well-formed ซึ่งมีลักษณะดังนี้

1. element ที่ถูกต้อง element ต้อง tag เปิด และต้องปิดด้วยด้วย tag ปิดเสมอ เช่น <...> </...> ยกเว้น element ว่าง

2. ข้อมูล attribute ต้องอยู่ภายในเครื่องหมาย '...' หรือ "..." หาก attribute ไม่ได้อยู่ในสัญลักษณ์ดังกล่าว เอกสารจะไม่ถือว่าเป็นแบบ well-formed

3. element ว่าง หากต้องการใช้งาน element ว่าง ต้องสร้าง element เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่างให้ถูกต้อง เอกสาร XML ไม่จำเป็นต้องมี element ที่มี tag เปิดและ tag ปิดเสมอ เรายังสามารถสร้าง element ว่าง หรือ Empty Element บรรจุลงในเอกสารได้ด้วย element ว่าง จะต้องไม่บรรจุตัวอักษรใด ๆ ลงภายใน element ดังกล่าว สามารถเขียนได้ในสองรูปแบบ element ที่มี tag เปิดและ tag ปิด โดยที่ไม่มีข้อความ หรือ Content อยู่ระหว่าง tag (ยกเว้นตัวอักษรขึ้นบรรทัดใหม่) เช่น `<book></book>` หรือ element เฉพาะ ที่เรียกว่า element ว่างตามรูปแบบ `<.../>` เช่น `<book/>` หรือ element ว่างที่มี attribute แน่นอน ซึ่ง element ว่าง สามารถบรรจุข้อมูล attribute ได้ เช่น `<book author="baldwin" price="$9.95" />`

4. ต้องไม่มีตัวอักษรประเภท Markup อยู่ในส่วน Content คือข้อมูล < หรือ & เนื่องจากตัวอักษร < ถูกใช้สำหรับการสร้าง tag และ & ถูกใช้สำหรับการแสดงสัญลักษณ์พิเศษ หากต้องการแสดงข้อมูลตัวอักษรดังกล่าว ในส่วน Content จริงๆ ให้ใช้สัญลักษณ์ใช้งาน `<` สำหรับแสดงตัวอักษร < และ `&` สำหรับแสดงตัวอักษร &

5. Element ต้องซ้อนกันอย่างมีรูปแบบ หากมีการสร้างเอกสาร XML ที่มี การวางซ้อนกันของ element ในหลายๆชั้น การวางซ้อนกันต้องมีรูปแบบ เป็นขั้นๆ ห้ามวาง element ในลักษณะที่ขึ้นไขว้กัน

ตัวอย่างเอกสารที่ element วางซ้อนกันอย่างถูกต้อง

```
<book>
  <chapter number="1">
    <paragraph>
    </paragraph>
  </chapter>
</book>
```

ตัวอย่างเอกสารที่วางซ้อน element อย่างไม่ถูกต้อง

```
<book>
  <chapter number="1">
    <paragraph>
  </chapter>
  </paragraph>
</book>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5.1 กฎและคำศัพท์ที่เกี่ยวข้อง

อันดับแรกการที่เราจะใช้งาน XML นั้น เราต้องอาศัยส่วนของการอ้างอิงความหมาย ที่เรียกว่า DTD (Document Type Definition) ในการกำหนดลักษณะและส่วนประกอบต่างๆ ของ element เรียกว่าเป็น grammar ของภาษา XML นั้นแหละ ซึ่ง DTD นี้เราสามารถเขียนขึ้นมาใช้เองได้ง่ายๆ หรือ ถ้ามีความซับซ้อนมากๆ ของเอกสาร XML เราอาจจะใช้ DTD ที่มีผู้เขียนไว้แล้วสำหรับเอกสารเฉพาะอย่าง ใ้ก็ได้

Well-formed และ valid ศัพท์ 2 คำนี้จะใช้ตรวจสอบว่า เอกสาร XML ที่เราเขียนขึ้นนั้น ถูกต้องหรือไม่ โดยใช้โปรแกรม EditML Pro 2.1.1 ถ้าถูกต้อง จะต้องประกอบด้วย 2 ประการคือ

1. เอกสาร xml นั้น tag ต่างๆ ต้องมีรูปแบบที่ถูกต้องเช่น ขึ้นต้น tag ด้วยตัวเล็กต้องจบ tag ด้วยตัวเล็กเป็นต้น และ หากมี tag เปิดต้องมี tag ปิดเสมอ เช่น

<boy></Boy> ---- ผิด , <BOY></boy> --> ผิด

<boy> ----> ผิดเพราะไม่มี tag ปิด

<girl></girl> ----> ถูกเพราะมี tag ปิด

แบบนี้เราจะเรียกว่า well-formed คือ โครงสร้างถูกต้องทั้งหมด

2. เอกสาร xml นั้น ต้องเป็นไปตามกฎของ DTD ของเอกสารนั้นๆ ด้วย จึงจะเรียกได้ว่า valid หากลองเพิ่ม code เพื่ออ้างอิง DTD กัน จะได้เป็น

<?xml version="1.0" encoding="windows-874"?>

<!DOCTYPE mali [

<!ELEMENT mali (malisorn,malila)>

<!ELEMENT malisorn (#PCDATA)>

<!ELEMENT malila (#PCDATA)>]>

<mali>

<malisorn>ขึ้นต้นด้วยมะลิซ้อน

</malisorn>

<malila>พอแตกใบอ่อนเป็นมะลิลา

</malila>

</mali>

<?xmlstylesheet type="text/xsl" href=""?>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปคือ สำหรับเอกสารที่มีรูปแบบ well-formed คือเอกสาร XML ที่สร้างถูกต้อง แต่หากต้องการให้เกิดบังคับโครงสร้างแล้ว มักนิยมสร้างเอกสาร XML โดยการอาศัย DTD มาช่วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบ

3.1 การวิเคราะห์การทำงานของมิดเดิลแวร์ที่จะทำการพัฒนา

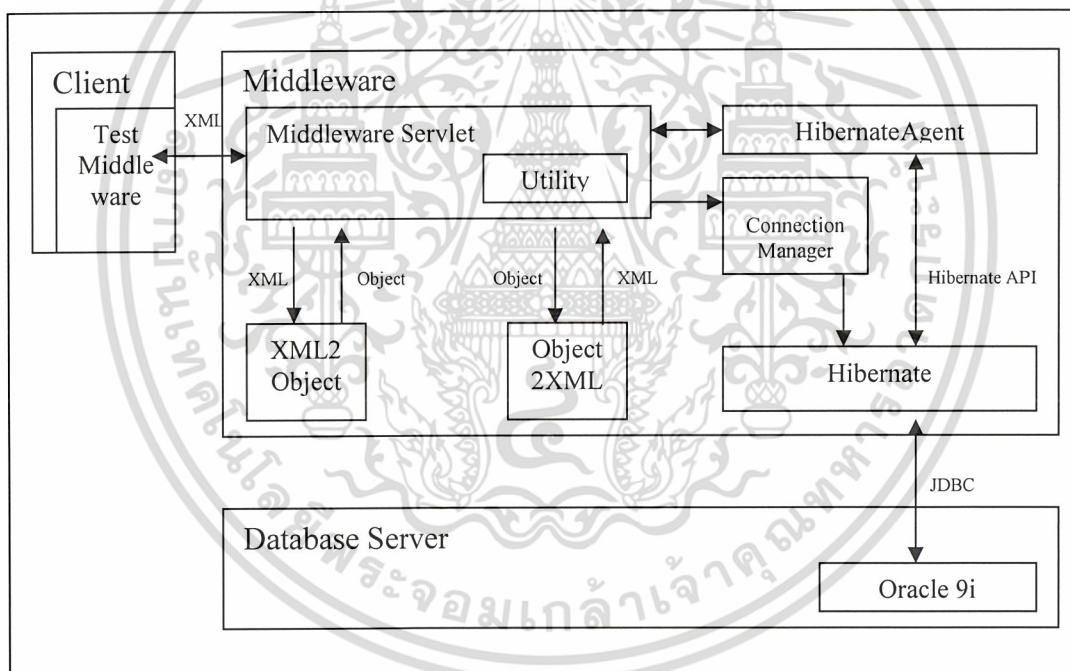
มิดเดิลแวร์ คือส่วนที่เป็นตัวเชื่อมการติดต่อเพื่อส่งผ่านข้อมูลระหว่าง Client/Server ซึ่งในปัญหาพิเศษนี้มีจุดประสงค์ที่จะทำการพัฒนาในส่วนของ Middleware ซึ่งเดิมการเรียกใช้งาน Middleware จะอยู่ในรูปของภาษาเอสคิวแอล ซึ่งเรียกรูปการส่งผ่านข้อมูลนี้ว่า SQL Message แต่ความต้องการของปัญหาพิเศษนี้คือทำให้การส่งผ่านข้อมูลเป็นแบบออบเจกต์พาสซึ่ง ดังนั้นการที่จะสามารถส่งผ่านข้อมูลให้อยู่ในรูปแบบเชิงวัตถุนั้นการใช้ภาษา Java ซึ่งเป็นภาษาที่มีรูปแบบเชิงวัตถุสูงนั้น น่าจะเหมาะแก่การพัฒนา Middleware ที่ต้องการที่สุด

โดยทางฝั่งของ Client จะเป็นฝั่งที่ทำการร้องขอการทำงาน Middleware ซึ่งจะติดต่อไปยังส่วน Servlet ที่มีอยู่ในตัวมิดเดิลแวร์ ซึ่งตัว Servlet นี้จะรอรับ request อยู่แล้ว และจะเรียกใช้ method เพื่อติดต่อและทำงานกับฐานข้อมูลที่มีมิดเดิลแวร์เตรียมไว้ให้ โดยจะส่งผ่านข้อมูลของออบเจกต์ ในรูปของเอกสารเอ็กซ์เอ็มแอล โดยจะใช้ Hibernate ช่วยในการเรียกใช้และส่งผ่านข้อมูลไปยังระบบฐานข้อมูลเพื่อทำการจัดการกับข้อมูลตามที่ไคลเอนต์ร้องขอ

3.2 รายละเอียดการออกแบบส่วนประกอบต่างๆ เพื่อใช้ในการทำงานของมิดเดิลแวร์

3.2.1 การออกแบบการติดต่อระหว่างส่วนประกอบต่างๆ เพื่อใช้ในการทำงานของมิดเดิลแวร์ ที่สามารถส่งผ่านข้อมูลในรูปแบบเชิงวัตถุ

ส่วนประกอบของมิดเดิลแวร์ที่ต้องการพัฒนาตามจุดประสงค์ตามปัญหาพิเศษนี้สามารถแสดงเป็นแผนผังโปรแกรมได้ดังรูปที่ 3.2.1 ซึ่งประกอบไปด้วยส่วนต่างๆ ที่เกี่ยวข้องในการทำงานของมิดเดิลแวร์ พร้อมทั้งแสดงแผนผังการทำงาน ในแผนผังจะแสดงให้เห็นว่าจะมีไคลเอนท์ถูกสร้างขึ้นมาสำหรับทดสอบการทำงานของมิดเดิลแวร์ โดยไคลเอนท์จะติดต่อไปยังส่วน Servlet ของมิดเดิลแวร์ ซึ่งทำการรองรับ request อยู่แล้ว และจะเรียกใช้ method ที่มีอยู่ในมิดเดิลแวร์เพื่อติดต่อกับฐานข้อมูลที่เตรียมไว้ โดยจะส่งผ่านข้อมูลของออบเจกต์ในรูปแบบของเอกสารเอกซ์เอ็มแอล โดยจะใช้ Hibernate ช่วยในการเรียกใช้และส่งผ่านข้อมูลไปยังระบบฐานข้อมูลเพื่อทำการจัดการกับข้อมูลตามที่ไคลเอนท์ร้องขอ



รูปที่ 3.2.1 แสดง Diagram ส่วนประกอบต่างๆ ของ Middleware

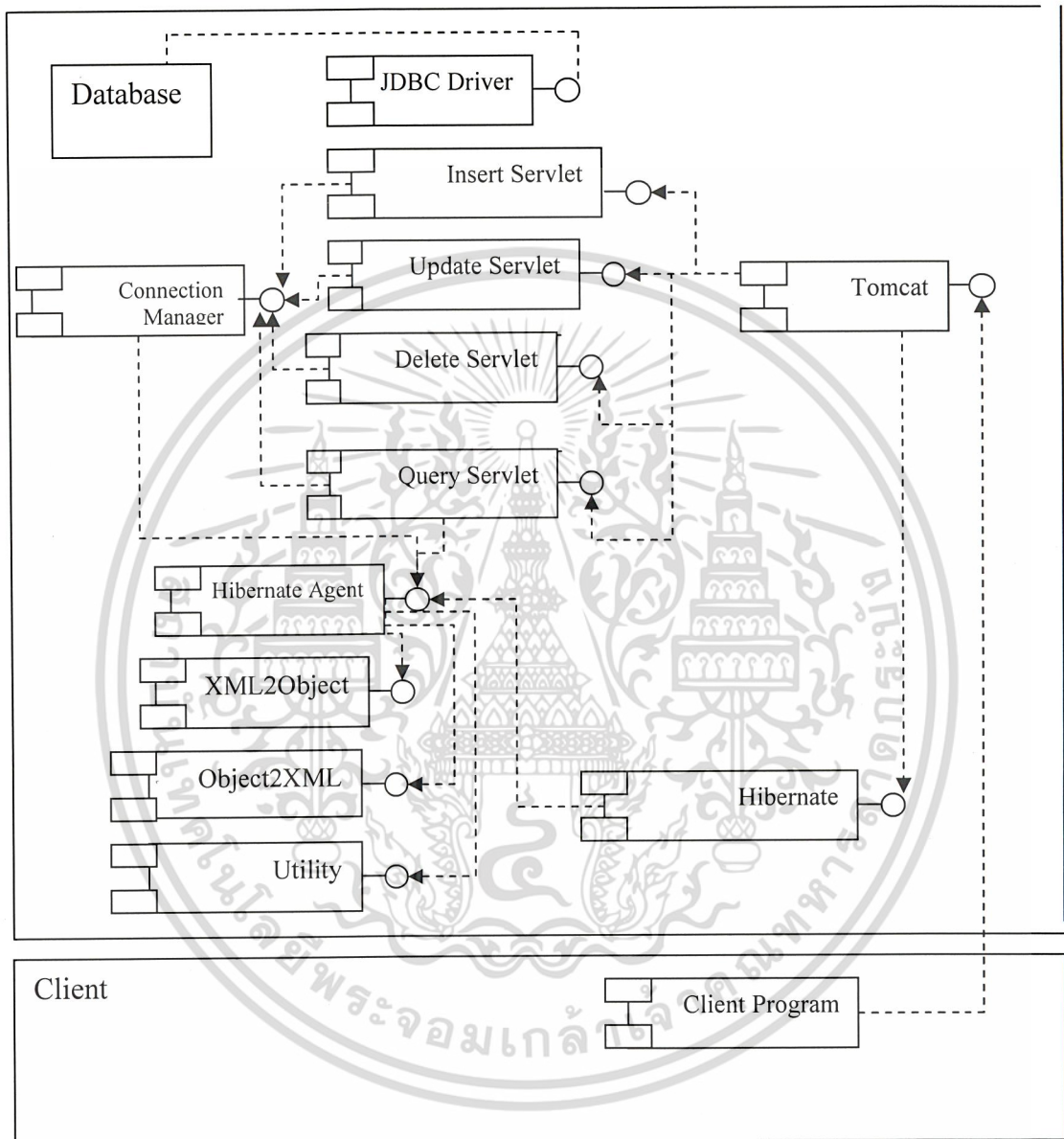
ตารางที่ 3.2.1 แสดงหน้าที่ของส่วนประกอบใน Middleware

ส่วนประกอบ	หน้าที่
Middleware Servlet	เป็นส่วนที่ทำหน้าที่รองรับ Request จาก ไคลเอนท์ ซึ่งจะทำการ Insert Update Delete หรือ Query
Object2XML	เป็นส่วนที่ทำหน้าที่แปลง Object ให้อยู่ในรูปของเอกสารเอ็็กซ์เอ็มแอล ซึ่งส่วนนี้จะถูกเรียกใช้เวลาที่ต้องการส่งข้อมูลที่ได้จากการ Query จากฐานข้อมูลกลับไปยัง ไคลเอนท์
XML2Object	เป็นส่วนที่ทำหน้าที่แปลงเอกสารเอ็็กซ์เอ็มแอล ให้อยู่ในรูปของ Object ซึ่งส่วนนี้จะถูกเรียกใช้เวลาที่รับข้อมูลจาก ไคลเอนท์ มาแล้วต้องการส่งต่อข้อมูลนี้ไปให้ Hibernate จะต้องแปลงข้อมูลให้อยู่ในรูปของ Object ก่อน
HibernateAgent	เป็นส่วนที่ทำหน้าที่ติดต่อกับ Hibernate เพื่อทำการ Insert Update Delete หรือ Query
Hibernate	เป็นส่วนที่ทำหน้าที่ติดต่อกับฐานข้อมูล โดย Hibernate จะทำหน้าที่แปลงข้อมูล Object ให้อยู่ในรูปที่สามารถจัดเก็บในฐานข้อมูลแบบ RDBMS หรือ ORDBMS ได้ ทำให้เราสามารถติดต่อกับฐานข้อมูลในรูปแบบ Object ได้ แม้ว่าฐานข้อมูลนั้นจะไม่ใช่แบบ OODBMS ก็ตาม
Connection Manager	เป็นส่วนที่ทำหน้าที่ในการ Initialize ค่าที่จำเป็นต่อการทำงานกับ Hibernate เพื่อทำการเชื่อมต่อกับ Database
Utility	เป็น Component ที่สนับสนุนการทำงานต่างๆของ Middleware Servlet เช่น การรับส่ง File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การออกแบบคอมโพเนนต์ต่างๆ ในระบบ

จากการวิเคราะห์เพื่อสร้างมิดเดิลแวร์ที่ส่งผ่านข้อมูลเชิงวัตถุได้นั้น แต่ละส่วนประกอบภายในมิดเดิลแวร์จะสามารถอธิบายได้ตามรูปที่ 3.2.2



รูปที่ 3.2.2 Component Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบขั้นตอนการทำงานของมิดเดิลแวร์

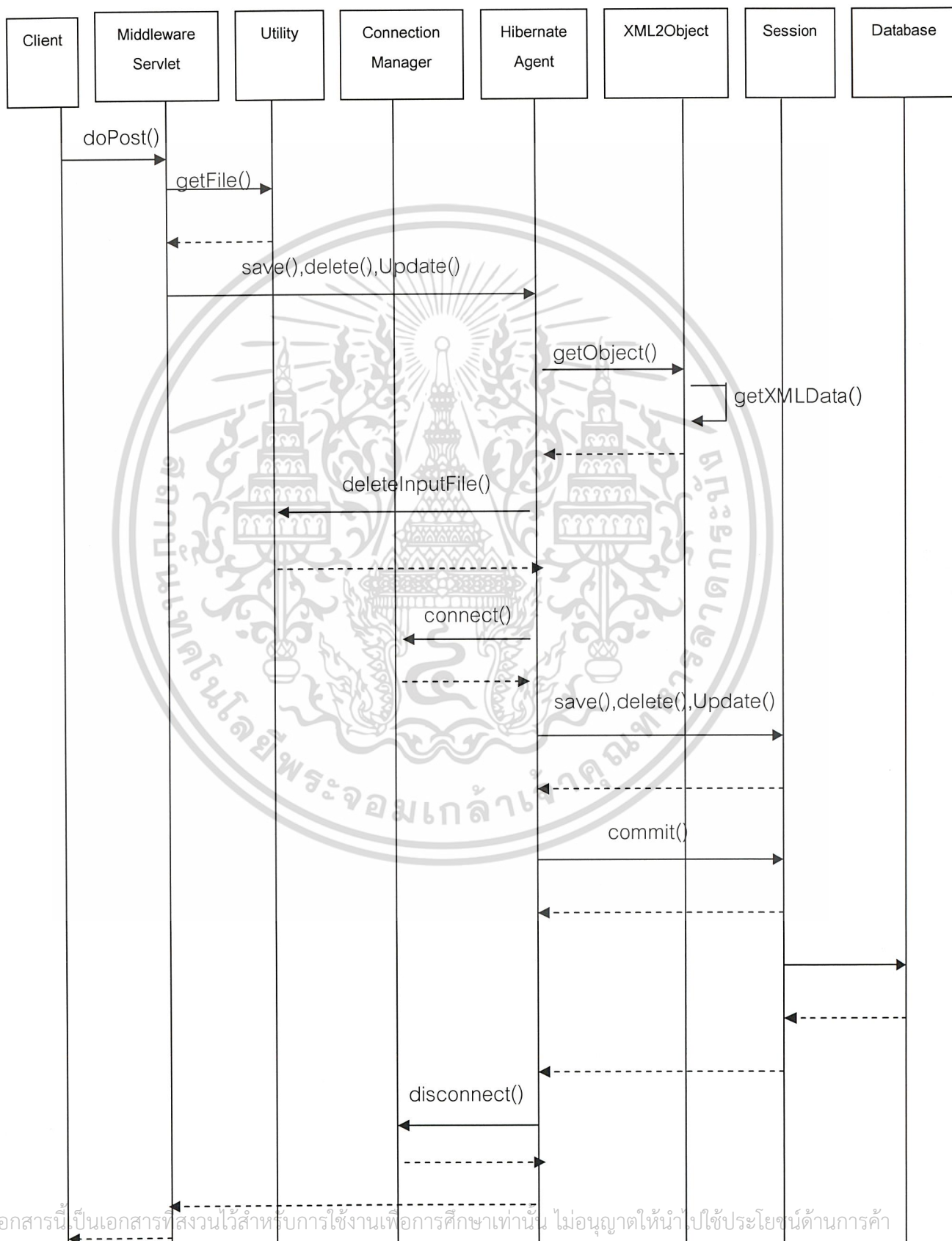
ในส่วนของการทำงานของ Middleware จะแบ่งลักษณะที่แตกต่างกันของขั้นตอนการทำงานออกเป็น 2 แบบ เนื่องจากการ Insert Update และ Delete จะมีลักษณะขั้นตอนการทำงานที่เหมือนกัน ส่วนการ Query นั้นจะมีขั้นตอนการทำงานที่ต่างออกไป

3.3.1 การออกแบบขั้นตอนการทำงานของ Middleware สำหรับการ Insert Update และ Delete

1. เมื่อไคลเอนต์ต้องการจะทำการ Insert Update หรือ Delete นั้นจะต้องทำการสร้างเอกสารเอ็กซ์เอ็มแอล ที่บรรจุข้อมูลของออบเจกต์ที่ต้องการ Insert Update Delete จากนั้นจึงทำการสร้างออบเจกต์ของ XMLContainer ขึ้นมาห่อหุ้มเอกสารเอ็กซ์เอ็มแอลนี้
2. ทำการ request ไปยัง servlet ที่ต้องการ โดยต้องทำการส่ง request ในแบบ POST method โดยในการ request นี้ให้ส่งออบเจกต์ของ XMLContainer ไปเป็น input เมื่อ servlet ได้รับ request แล้วจะทำการดึงข้อมูลเอกสารเอ็กซ์เอ็มแอลจากออบเจกต์ของ XMLContainer และจะเรียกใช้งาน component Utility เพื่อทำการสร้างไฟล์เอกสารเอ็กซ์เอ็มแอล
3. เมื่อ component Utility ทำการสร้างไฟล์เอกสารเอ็กซ์เอ็มแอลเสร็จแล้ว จะทำการเรียกใช้งาน component HibernateAgent
4. ซึ่ง component HibernateAgent จะเรียกใช้งาน component XML2Object เพื่อทำการสร้างออบเจกต์ที่ต้องการ Insert Update หรือ Delete
5. จากนั้นเรียกใช้งาน component ConnectionManager เพื่อให้ Hibernate ทำการเชื่อมต่อกับฐานข้อมูล
6. จากนั้นจึงทำการสั่งให้ Insert Update หรือ Delete

3.3.1.1 ซีควেনซ์ไดอะแกรมของการ Insert Update และ Delete

จากการออกแบบการทำงานของ Middleware สำหรับการ Insert Update และ Delete ในหัวข้อ 3.3.2 สามารถเขียนเป็นซีควেনซ์ไดอะแกรมได้ดังรูปที่ 3.3.2.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

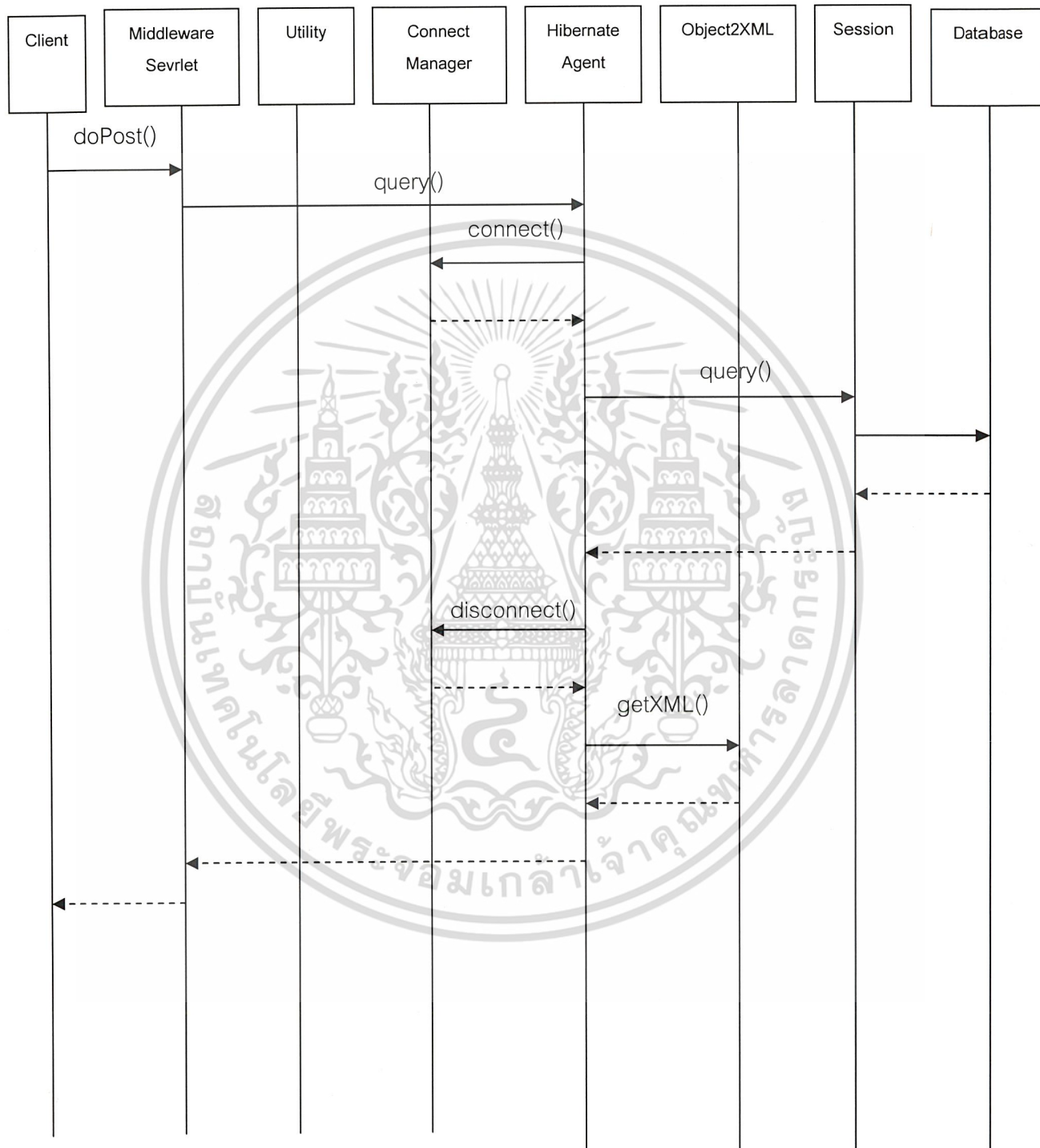
รูปที่ 3.3.1.1 ซีควেনซ์ไดอะแกรมของการ Insert Update และ Delete

3.3.2 การออกแบบขั้นตอนการทำงานของ Middleware สำหรับการทำการ Query

1. การ Query นั้น เมื่อไคลเอนท์จะทำการ Query ให้ทำการสร้าง ออบเจกต์ ของ คลาส XMLContainer ซึ่งห่อหุ้มสตริงที่แทนคลาสและเงื่อนไขที่จะทำการ Query จากนั้นให้
2. ทำการ request แบบ POST method ไปยัง servlet โดยส่ง ออบเจกต์ของ XMLContainer ไปเป็น input เมื่อ servlet ได้รับ request แล้วจะทำการ แกะข้อมูลสตริงซึ่งถูกห่อหุ้มไว้ใน ออบเจกต์ของ XMLContainer ออกมาแล้ว
3. จากนั้นจะทำการเรียกใช้งาน component HibernateAgent โดยจะส่งสตริงที่แทนคลาสและเงื่อนไขที่ต้องการค้นหาไปเป็นพารามิเตอร์
4. ซึ่ง HibernateAgent จะเรียกใช้งาน component ConnectionManager เพื่อให้ Hibernate ทำการเชื่อมต่อกับฐานข้อมูล จากนั้นก็จะทำการ Query
5. เมื่อได้ผลลัพธ์จากการ Query มาแล้ว ก็จะเรียกใช้งาน component Object2XML เพื่อแปลง ออบเจกต์ที่ได้จากการค้นหาให้เป็นเอกสารเอ็กซ์เอ็มแอล
6. จากนั้นจะสร้างออบเจกต์ของคลาส XMLContainer ขึ้นมาเพื่อห่อหุ้มเอกสารเอ็กซ์เอ็มแอลนี้ แล้วส่ง output กลับไปยัง ไคลเอนท์

3.3.2.1 ซีควেনซ์ไดอะแกรมของการ Query

จากการออกแบบการทำงานของ Middleware สำหรับการ Query ในหัวข้อ 3.3.3 สามารถเขียนเป็นซีควেনซ์ไดอะแกรมได้ดังรูปที่ 3.3.3.2



รูปที่ 3.3.2.1 ซีควেনซ์ไดอะแกรมของการ Query

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

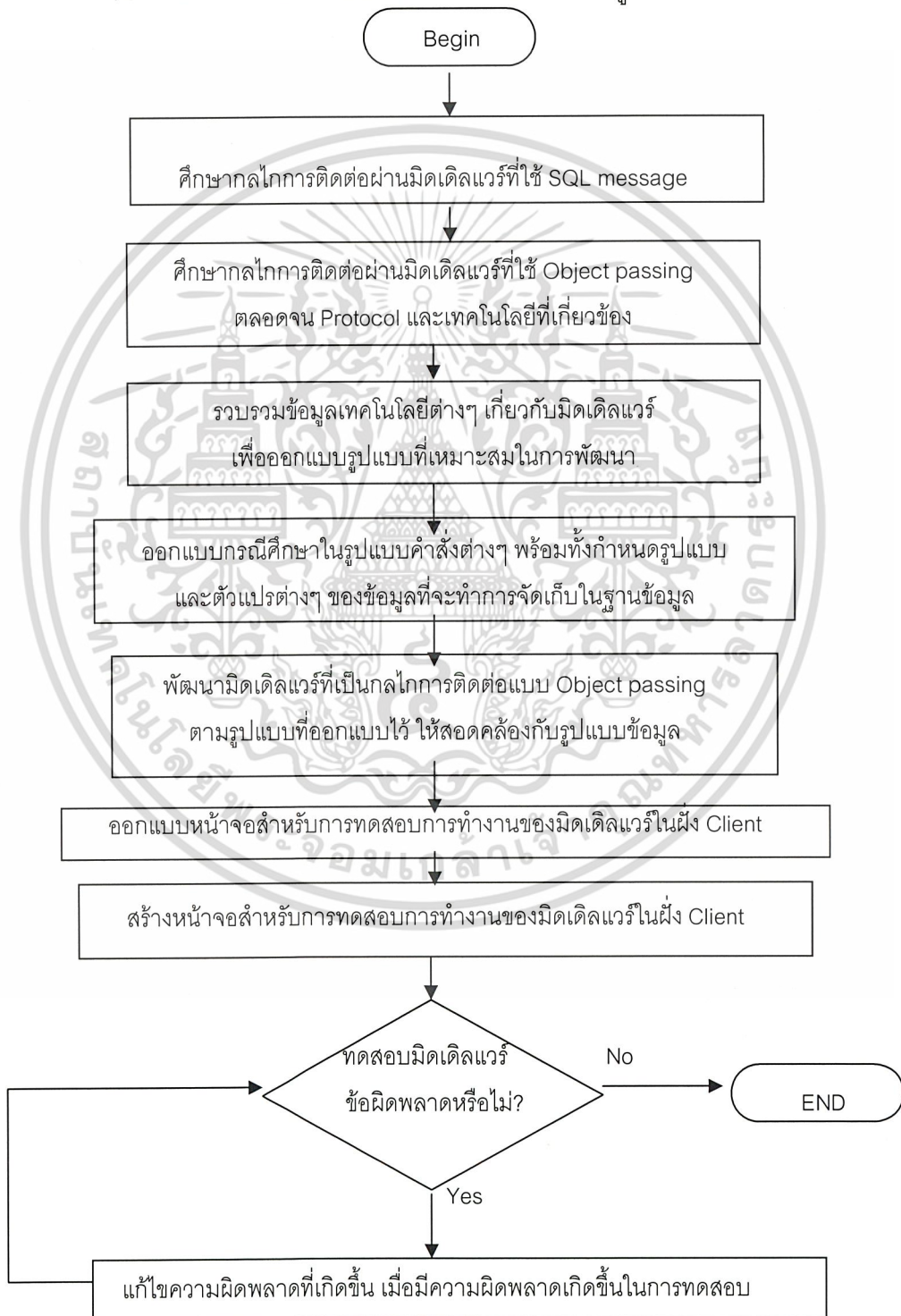
บทที่ 4

การพัฒนาและทดสอบ

4.1 การพัฒนาโปรแกรมเพื่อนำไปทดสอบ

4.1.1 แผนผังแสดงขั้นตอนการพัฒนาไมดเดิลแวร์

ในการพัฒนาสามารถอธิบายเป็นขั้นตอนต่างได้ดังรูปที่ 4.1.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.1.1 Flow Chart ขั้นตอนการพัฒนาส่วนต่างๆ เพื่อทดสอบมิดเดิลแวร์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 รายละเอียดข้อมูลต่างๆ ที่จะทำการจัดเก็บในฐานข้อมูลฝั่ง Server

จากการสร้างและพัฒนาเว็บที่ผ่านมา ส่วนที่มีความสำคัญในการทดสอบการทำงานของ Middleware นั้น จะต้องมีการสร้างข้อมูลในฐานข้อมูล ซึ่งในส่วนนี้เราจะหมายถึงข้อมูลที่ถูกเก็บบันทึกใน Database Server เพื่อจะได้สามารถทดสอบได้ว่า Middleware ที่สร้างขึ้นนั้น มีความสามารถในการทำงาน Insert Delete Update และ Query ได้จริง

4.2.1 การออกแบบกรณีศึกษาในรูปแบบคำสั่งต่างๆ

ในการทดสอบการทำงาน จะประกอบด้วย การ Select Insert Update และ Delete โดยจะยกตัวอย่างในแต่ละกรณีดังต่อไปนี้

4.2.1.1 Insert

สมมุติว่าต้องการทำการ insert ข้อมูล class Teacher ซึ่งมี relation กับ class วิชาที่สอนแบบ one-many และมี relation กับ class Department แบบ one-one เมื่อทำการ insert จะ insert Teacher และ relation กับ class Department โดยอัตโนมัติ แต่ไม่มีการ insert วิชาที่สอน หรือในกรณีเดียวกันกับการ insert วิชาที่สอน ก็จะมี insert relation กับ Teacher ด้วยโดยอัตโนมัติ

4.2.1.2 Select

สมมุติให้ class Teacher มี relation แบบ one-many กับ class วิชาที่สอน และมี relation แบบ one-one กับ class Major เมื่อทำการ select เพื่อค้นหารายชื่ออาจารย์ทุกคนที่สอนในภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ จะได้ Object Teacher ทุก Object ที่เป็นไปตามเงื่อนไขที่กำหนดในการ Select และแต่ละ Object ที่ได้มานั้นจะ Create ทุก Object ที่มี relation กัน แต่จะทำการ initialize ค่าไว้เฉพาะกับค่าที่เป็น Primary key

4.2.1.3 Delete

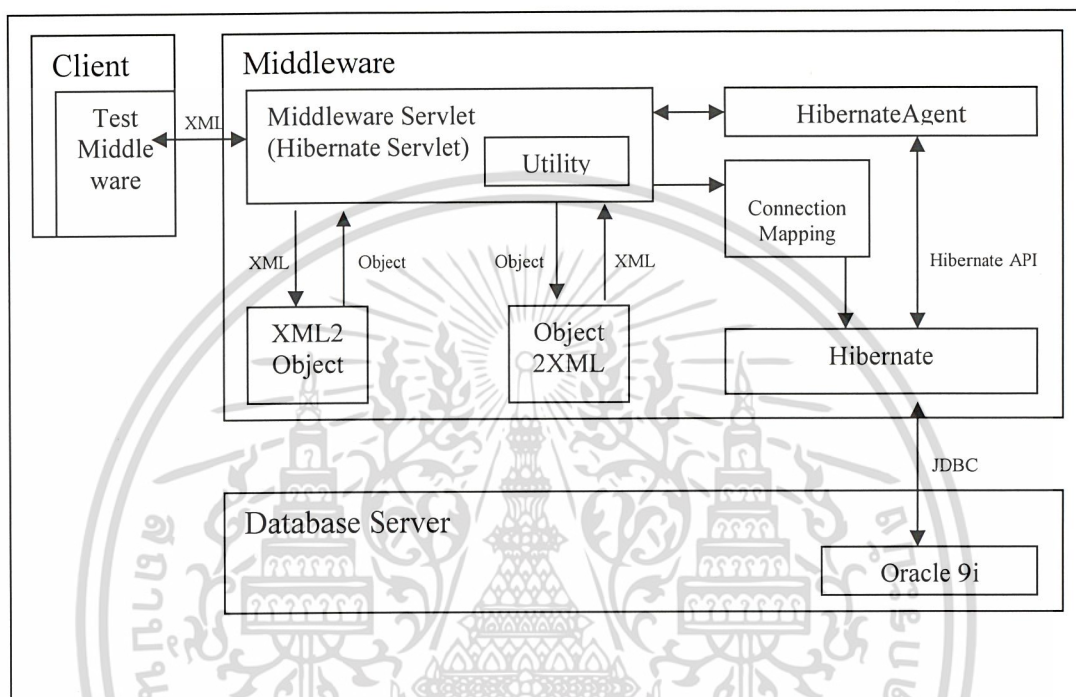
สมมุติว่าต้องการลบ Department ออก ซึ่งมี relation กับ Teacher และ Faculty ดังนั้น Teacher ทั้งหมดที่สังกัดใน Department ที่ทำการลบจะโดนลบด้วย แต่ Faculty ยังอยู่

4.2.1.4 Update

สมมุติให้ทำการ update class Teacher การทำการ update นี้ก็จะมีผลเฉพาะกับ class Teacher ส่วน Object ต่างๆ ที่มี relation กับ Teacher จะไม่มีการเปลี่ยนแปลงใดๆ ใน database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างข้อมูลในฐานข้อมูล ซึ่งในส่วนนี้เราจะหมายถึงข้อมูลที่ถูกเก็บบันทึกใน Database Server จากรูปที่ 4.2

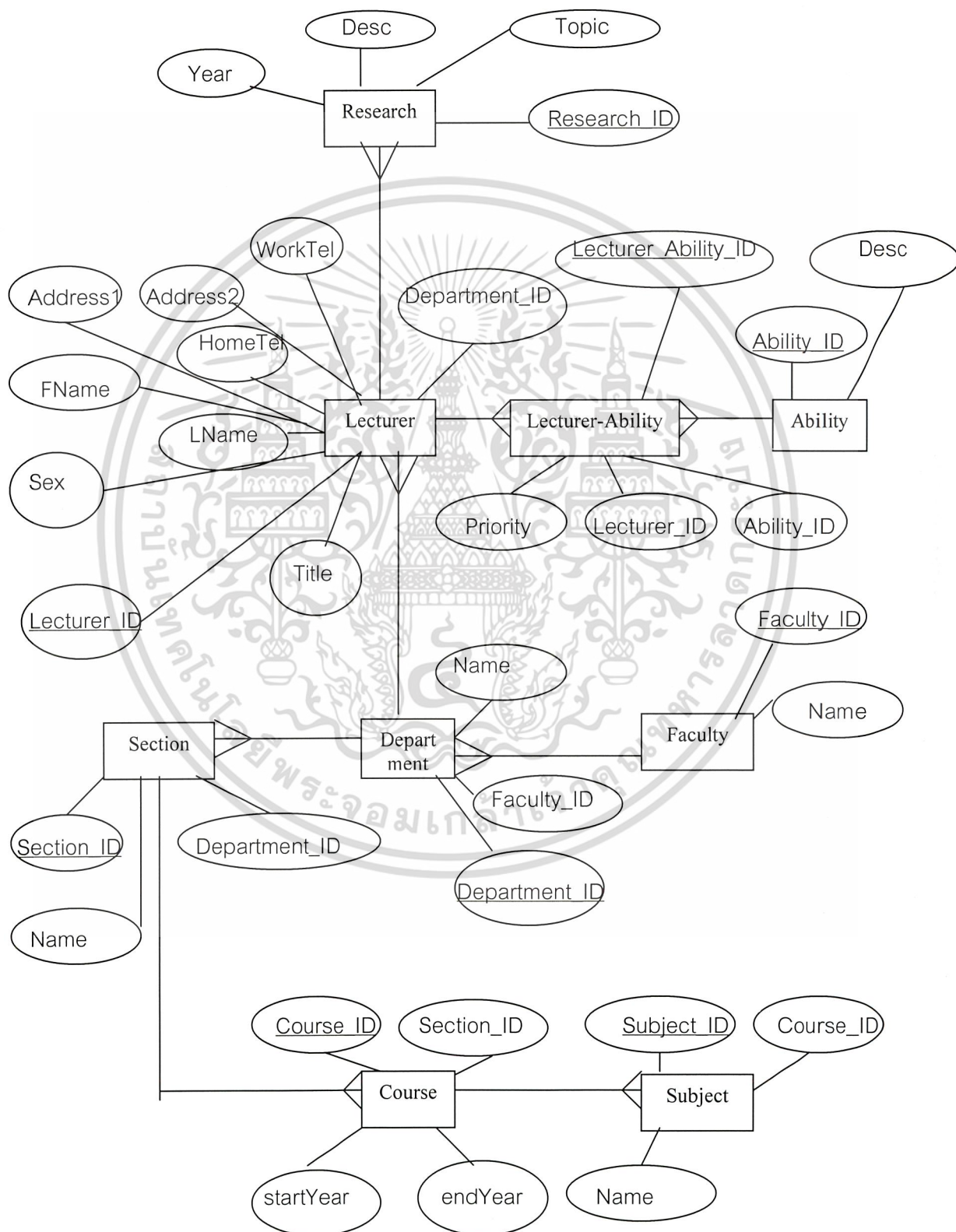


รูปที่ 4.2 แสดง Diagram ส่วนประกอบต่างๆ ของ Middleware

ข้อมูลที่เราทำการสร้างขึ้นเพื่อทดสอบมิตเดิลแวร์นั้นจะต้องมีลักษณะเชิงวัตถุ นั่นคือต้องมีหลาย class และแต่ละ class จะต้องประกอบไปด้วย Attribute ต่างๆ โดยจะสามารถอธิบายได้ในรูปแบบดังต่อไปนี้

4.2.2 รายละเอียดข้อมูลโดยแสดงใน E-R Diagram

E-R Diagram จะอธิบายถึงส่วนประกอบต่างๆ ที่ จะทำการจัดเก็บไว้ในฐานข้อมูล ซึ่งได้แก่ Class Attribute และ Relation ต่างๆ ดังรูปที่ 4.2.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2.2 ตาราง E-R Diagram

4.2.3 ตารางรายละเอียดข้อมูลต่างๆ ในฐานข้อมูล

จากตาราง E-R Diagram จะสามารถอธิบายในรูปตารางเพื่อบอกถึงชื่อ ประเภท ลักษณะของ Key และคำอธิบายข้อมูลได้ ดังต่อไปนี้

ตารางที่ 4.2.3.1 แสดงรายละเอียดตารางอาจารย์

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Lecturer_ID	VARCHAR2(8)	PK		รหัสอาจารย์
2	Department_ID	VARCHAR2(2)	FK		รหัสภาควิชาที่สังกัด
3	Title	VARCHAR2(10)	N.N.		คำนำหน้าชื่อ เช่น รศ.ดร.
4	FName	VARCHAR2(30)	N.N.		ชื่ออาจารย์
5	LName	VARCHAR2(30)	N.N.		นามสกุล
6	Sex	VARCHAR2(1)	N.N.	F,M	เพศ
7	Address1	VARCHAR(50)	N.N.		ที่อยู่
8	Address2	VARCHAR(50)			ที่อยู่
9	HomeTel	VARCHAR(9)			เบอร์โทรศัพท์ที่บ้าน
10	WorkTel	VARCHAR(9)			เบอร์โทรศัพท์ที่ทำงาน

ตารางที่ 4.2.3.2 แสดงรายละเอียดตารางผลงานวิจัยของอาจารย์

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Research_ID	VARCHAR2(8)	PK		รหัสอาจารย์
2	Topic	VARCHAR(50)	PK		หัวข้อผลงานวิจัยของ อาจารย์
3	Year	VARCHAR2(4)	N.N.		ปีพ.ศ.ที่อาจารย์ทำ ผลงานวิจัย
4	Desc	VARCHAR2(50)			รายละเอียดงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2.3.3 แสดงรายละเอียดตารางอาจารย์-สาขาที่เชี่ยวชาญและสนใจ

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Lecturer_Ability_ID	VARCHAR2(8)	PK		รหัสอาจารย์
2	Ability_ID	VARCHAR2(2)	PK,FK		รหัสสาขาวิชาที่ เชี่ยวชาญและสนใจ
3	Lecturer_ID	VARCHAR2(8)	PK,FK		รหัสอาจารย์
4	Priority	NUMBER(2)	N.N.		ลำดับความ เชี่ยวชาญและสนใจ

ตารางที่ 4.2.3.4 แสดงรายละเอียดตารางสาขาที่เชี่ยวชาญและสนใจ

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Ability_ID	VARCHAR2(2)	PK		รหัสสาขาที่เชี่ยวชาญ และสนใจ
2	SabDesc	VARCHAR2(50)	N.N.		วิชาที่เชี่ยวชาญและ สนใจ

ตารางที่ 4.2.3.5 แสดงรายละเอียดตารางหลักสูตร

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Course_ID	VARCHAR2(2)	PK		รหัสหลักสูตร
2	startYear	VARCHAR2(4)	N.N.		ปีการศึกษาที่เริ่มใช้ หลักสูตร
3	endYear	VARCHAR2(4)	N.N.		ปีการศึกษาที่สิ้นสุด หลักสูตร
4	Section_ID	VARCHAR2(2)	FK		รหัสสาขาวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2.3.6 แสดงรายละเอียดตารางสาขาวิชา

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Section_ID	VARCHAR2(2)	PK		รหัสภาควิชา
2	Name	VARCHAR2(50)	N.N.		ชื่อสาขาวิชา
3	Department_ID	VARCHAR2(2)	FK		สังกัดภาควิชา

ตารางที่ 4.2.3.7 แสดงรายละเอียดตารางภาควิชา

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Department_ID	VARCHAR2(2)	PK		รหัสสาขาวิชา
2	Name	VARCHAR2(50)	N.N.		ชื่อภาควิชา
3	Faculty_ID	VARCHAR2(2)	FK		สังกัดคณะ

ตารางที่ 4.2.3.8 แสดงรายละเอียดตารางคณะ

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Faculty_ID	VARCHAR2(2)	PK		รหัสคณะ
2	Name	VARCHAR2(30)	N.N.		ชื่อคณะ

ตารางที่ 4.2.3.9 แสดงรายละเอียดตารางวิชา

ลำดับ	ชื่อฟิลด์	ประเภทข้อมูล	Key	Domain	Description
1	Subject_ID	VARCHAR2(8)	PK		รหัสวิชา
2	Course_ID	VARCHAR2(2)	PK,FK		รหัสหลักสูตร
3	Name	VARCHAR2(50)	N.N.		ชื่อวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการทำงานของโปรแกรม

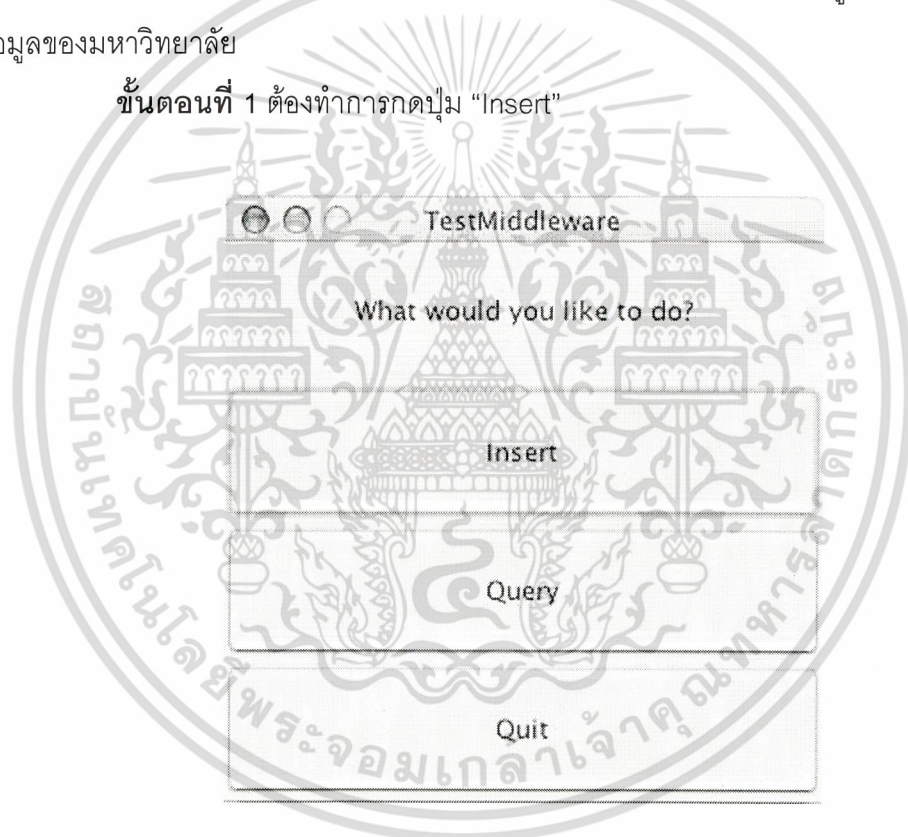
ในการทดสอบการทำงานของคำสั่งต่างๆ จะเริ่มต้นที่ส่วนของหน้าจอสำหรับผู้ใช้ในการติดต่อเพื่อเรียกใช้งานมิดเดิลแวร์

4.3.1 การทดสอบการเพิ่มเติมข้อมูล

เมื่อต้องการเพิ่มข้อมูลเข้าไปยังฐานข้อมูล โดยที่อาจเป็นการเพิ่มข้อมูลเกี่ยวกับภาควิชา เช่น

สมมติให้มหาวิทยาลัย A จัดเพิ่มภาควิชาคณิตศาสตร์ขึ้นมา โดยภาควิชาคณิตศาสตร์นั้นสังกัดในคณะวิทยาศาสตร์ เมื่อฝ่ายทะเบียนต้องการเพิ่มข้อมูลดังกล่าวลงในฐานข้อมูลของมหาวิทยาลัย

ขั้นตอนที่ 1 ต้องทำการกดปุ่ม "Insert"



รูปที่ 4.3.1.1 หน้าจอสำหรับเลือกรายการที่ต้องการกระทำ

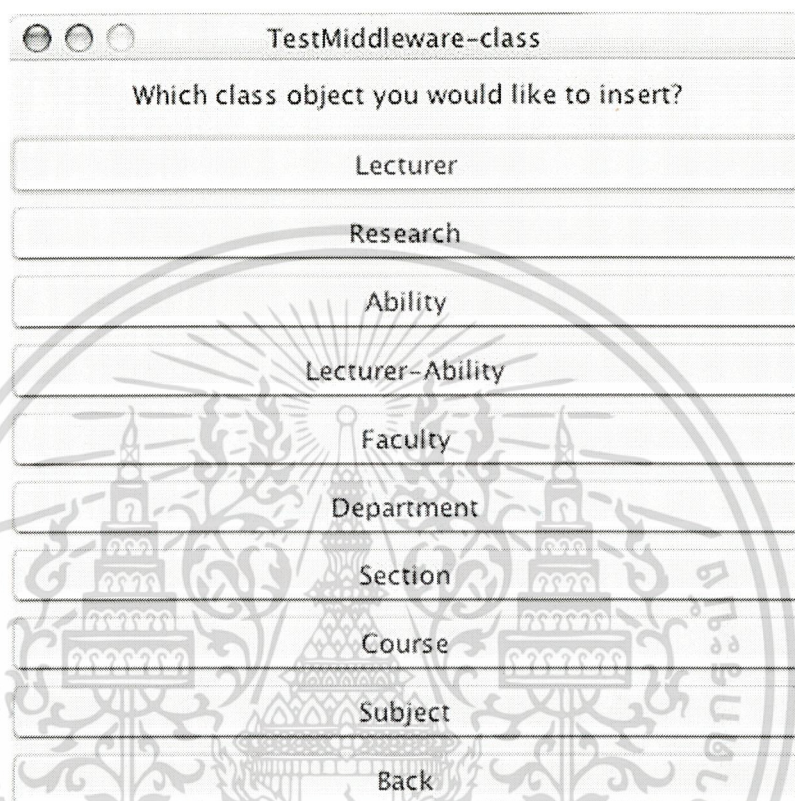
ซึ่งในหน้าจอแรกจะมีส่วนประกอบต่างๆ ดังนี้

ส่วนที่ 1 เป็นส่วนหัว ในส่วนนี้จะแสดงถึงชื่อระบบที่ใช้งาน ในที่นี้คือการทดลองใช้งานมิดเดิลแวร์

ส่วนที่ 2 เป็นส่วนของรายการหลักให้ผู้ใช้เลือกการทำงานที่ต้องการกระทำ ได้แก่ Insert Query และ Quit โดยเมื่อผู้ใช้กดที่ชื่อรายการที่ต้องการ หน้าจอจะเปลี่ยนไปยังหน้าจอที่ต้องการกระทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 จากรูป 4.3.1.1 เมื่อผู้ใช้ทำการกดที่คำสั่ง Insert หน้าจอจะเปลี่ยนไป ในกรณีนี้เราต้องการเพิ่มเติมข้อมูลภาควิชา ดังนั้นจึงต้องกดปุ่มคำสั่ง “Department”



รูปที่ 4.3.1.2 หน้าจอสำหรับเลือกรูปแบบข้อมูลที่ต้องการเพิ่มเติมข้อมูล

ในหน้าจอของการเพิ่มเติมข้อมูลจะประกอบไปด้วยปุ่มต่างๆ เพื่อให้ผู้ใช้สามารถเลือกชนิดของ class ที่ต้องการเพิ่มเติมข้อมูล ได้แก่

ปุ่ม Lecturer สำหรับเพิ่มเติมข้อมูลทั่วไปเกี่ยวกับอาจารย์ เช่น ชื่อนามสกุล เป็นต้น

ปุ่ม Research สำหรับเพิ่มเติมข้อมูลผลงานวิจัยต่างๆ ของอาจารย์

ปุ่ม Ability สำหรับเพิ่มเติมข้อมูลเกี่ยวกับสาขาวิชาต่างๆ ที่อาจารย์มีความเชี่ยวชาญและความสนใจ

ปุ่ม Lecturer-Ability สำหรับเพิ่มเติมข้อมูลลำดับความเชี่ยวชาญและสนใจในสาขาวิชาต่างๆ

ปุ่ม Faculty สำหรับเพิ่มเติมข้อมูลคณะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น มิใช่ของลิขสิทธิ์ในเชิงพาณิชย์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้โดยไม่ได้รับอนุญาตจากผู้นิพนธ์ทุกครั้งที่มีการนำไปใช้

ปุ่ม Course สำหรับเพิ่มเติมข้อมูลรายละเอียดเกี่ยวกับหลักสูตรการ
สอนต่างๆ

ปุ่ม Subject สำหรับเพิ่มเติมข้อมูลวิชาต่างๆที่เปิดสอน

ปุ่ม Back สำหรับผู้ใช้ที่เปลี่ยนใจไม่ต้องการเพิ่มเติมข้อมูล หรือ
ผู้ใช้ต้องการจะกลับไปเลือกการทำงานในหน้าจอแรกใหม่

ขั้นตอนที่ 3 เมื่อเลือกรูปแบบการกรอกข้อมูลที่ต้องการเพิ่มเติมจากหน้าจอรูปที่
4.3.1.2 แล้ว หน้าจอจะเปลี่ยนไปในแต่ละช่องให้ผู้ใช้กรอกข้อมูลลงใน Text Box ซึ่งในช่องที่มี
เครื่องหมาย Star (*) หมายถึงต้องกรอกข้อมูลลงไป ห้ามว่าง ส่วนช่องอื่นๆ ไม่จำเป็นต้องกรอก
ข้อมูลลงไปก็ได้ ดังนั้นในการเพิ่มข้อมูลภาควิชาก็ต้องพิมพ์ชื่อภาควิชาและชื่อย่อภาควิชาดังรูป
4.3.1.3

รูปที่ 4.3.1.3 หน้าจอสำหรับเพิ่มเติมข้อมูลภาควิชา

จากนั้นให้กดปุ่ม “Link to persist object” เพื่อใส่ค่าในกรณีที่มี Association กับ
class อื่นๆ แบบ Many – One ซึ่งมีรายละเอียดดังรูป 4.3.1.4

ขั้นตอนที่ 4 จากนั้นให้ผู้ใช้ระบุรายละเอียดต่อไปนี้

ช่องแรก Class name ให้ผู้ใช้ระบุชื่อ class ที่มี association ในที่นี้คือ
Faculty

ช่องที่ 2 Value ให้ผู้ใช้ระบุค่าสำหรับ Attribute ที่เป็น Foreign Key
ตัวอย่างรูปแบบสำหรับการกรอกข้อมูล คือ Fac.name = ‘Sci’

จากนั้นให้ผู้ใช้กดปุ่ม Set to Reference หากต้องการเพิ่มข้อมูล หรือ
กดปุ่ม Back หากผู้ใช้ต้องการกลับไปยังหน้าจอหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TestMiddleware-persist

Enter class name and value

Class name Faculty

Value fac.name = 'sci'

Back Set to Reference

รูป 4.3.1.4 หน้าจอสำหรับกรอกค่าให้แก่ Class ที่มี Association กับ Class ที่ต้องการเพิ่มเติมข้อมูล

เมื่อทำตามขั้นตอนที่ 4 แล้ว หน้าจอจะย้อนกลับไปยังรูปที่ 4.3.1.3

ขั้นที่ 5 กดปุ่ม "Save" เพื่อทำการบันทึกข้อมูลที่เพิ่มเติมลงฐานข้อมูล

TestMiddleware-lecturer

Enter value of attribute <*attribute is a must>

*ชื่อภาควิชา Math

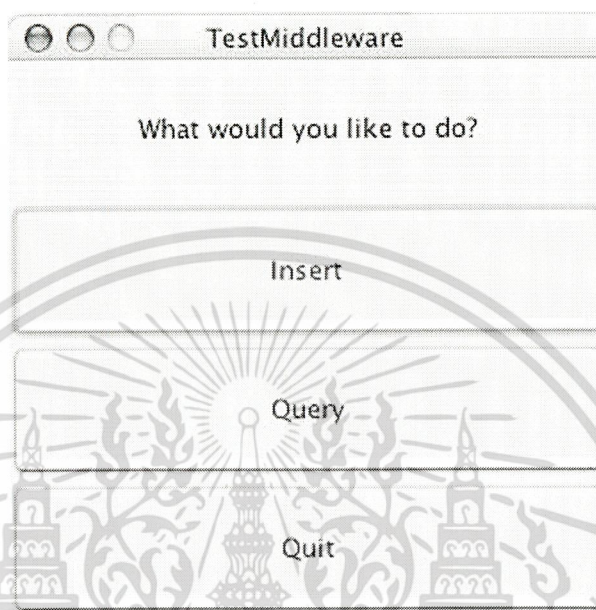
*ชื่อภาควิชา Mat

Back Link to persist... Save

รูปที่ 4.3.1.5 หน้าจอสำหรับเพิ่มเติมข้อมูลภาควิชา

4.3.2 การทดสอบการค้นหาข้อมูล

เมื่อต้องการค้นหาข้อมูลเกี่ยวกับคณะต่างๆ ที่มีในมหาวิทยาลัย
ขั้นตอนที่ 1 ให้ทำการกดที่ปุ่ม “Query” จากรูป 4.3.2.1



รูปที่ 4.3.2.1 หน้าจอสำหรับเลือกรายการที่ต้องการกระทำ

ขั้นตอนที่ 2 เมื่อผู้ใช้ทำการกดที่คำสั่ง Query หน้าจอจะเปลี่ยนไป โดยมีลักษณะ

ดังนี้

ให้กรอกชื่อ class ที่ต้องการทำการค้นหาข้อมูล ตัวอย่างการกรอกข้อมูลคือ

ช่องแรก เป็นช่องให้ระบุชื่อ class ที่ต้องการค้นหาข้อมูลในที่นี้คือค้นหา

Class ชื่อ "Faculty"

ช่องที่ 2 เป็นช่องสำหรับระบุ class ที่มี Association แบบ Many – One

เช่นหากช่องแรกเป็น Department ตัวอย่างของการกรอกช่องที่ 2 คือ Fac.name = 'Sci'

TestMiddleware-query

Enter class name you would like to query Faculty

Enter condition Fac.name = 'Sci'

OK

Chose Object you would like to work with 1

Delete Show Reference key Get Reference Edit Back

รูปที่ 4.3.2.2 หน้าจอสำหรับค้นหาข้อมูล

หลังจากกรอกข้อมูลในส่วนที่ 1 เรียบร้อยแล้ว ให้ทำการกดปุ่ม “OK” เมื่อกดเรียบร้อยแล้ว ข้อมูลจะถูกดึงขึ้นมาปรากฏดังรูปที่ 4.3.2.3

TestMiddleware-query

Enter class name you would like to query Faculty

Enter condition Fac.name = 'Sci'

OK

id	name	shortName		
1:	ff80808102e5a0420102e5a22008000c	Physic	Physic	
2:	ff80808102e5a0420102e5a159710006	null	null	
3:	ff80808102e5a3ae0102e5a3f2ad0003	Chem	Chem	
4:	ff80808102e5a3ae0102e5a45d090006	Stat	Stat	
5:	ff80808102e5a0420102e5a1cdad0009	null	null	
6:	ff80808102e5a0420102e5a22008000c	null	null	
7:	ff80808102e5a0420102e5a1cdad0009	Bio	Bio	
8:	ff80808102e5a3ae0102e5a3f2ad0003	null	null	
9:	ff80808102e5a3ae0102e5a45d090006	null	null	
10:	ff80808102e5a0420102e5a090c50003	Math	Math	
11:	ff80808102e5a0420102e5a159710006	Com	Com	
12:	ff80808102e5a0420102e5a090c50003	null	null	

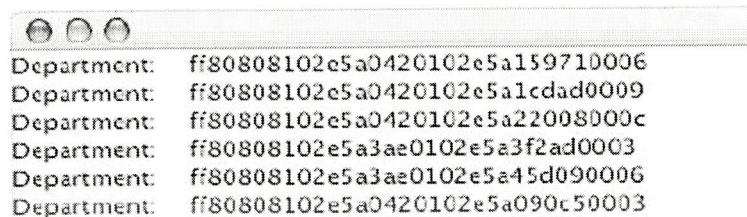
Chose Object you would like to work with 1

Delete Show Reference key Get Reference Edit Back

รูปที่ 4.3.2.3 หน้าจอแสดงผลการค้นหาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งหลังจากขั้นตอนนี้หากต้องการทราบ Associate ต่างๆ ของ class ที่เราทำการค้นหา ก็สามารถทำการกดปุ่ม Get Reference ได้ หน้าจอจะแสดงผลดังรูปที่ 4.3.2.4



```

Department: ff80808102e5a0420102e5a159710006
Department: ff80808102e5a0420102e5a1cdad0009
Department: ff80808102e5a0420102e5a22008000c
Department: ff80808102e5a3ae0102e5a3f2ad0003
Department: ff80808102e5a3ae0102e5a45d090006
Department: ff80808102e5a0420102e5a090c50003
  
```

รูปที่ 4.3.2.4 หน้าจอแสดงผล Associate ต่างๆ ของ class ที่เราทำการค้นหา

จากหน้าจอที่ 4.3.2.4 เมื่อทำการกดปุ่ม "OK" แล้วหน้าจอจะกลับไปยังหน้าจอรูปที่ 4.3.2.3 ซึ่งจากหน้าจอนี้ผู้ใช้สามารถเลือกที่จะกระทำการ Delete หรือ Update ข้อมูลต่อไปได้

4.3.3 การทดสอบการเปลี่ยนแปลงข้อมูล

ในรูปแบบการกระทำลักษณะเดียวกับการลบข้อมูลผู้ใช้สามารถที่จะทำการเปลี่ยนแปลงข้อมูลได้เช่นกัน เพียงแต่เปลี่ยนจากกดปุ่มคำสั่ง Delete เป็น Edit แทน

หากเราต้องการที่จะเปลี่ยนแปลงชื่อภาควิชาคณิตศาสตร์ ที่สังกัดในคณะวิทยาศาสตร์ เราสามารถทำได้ดังขั้นตอนต่อไปนี้

ขั้นตอนที่ 1 ให้ทำการกดที่ปุ่ม “Query” จากรูป 4.3.3.1



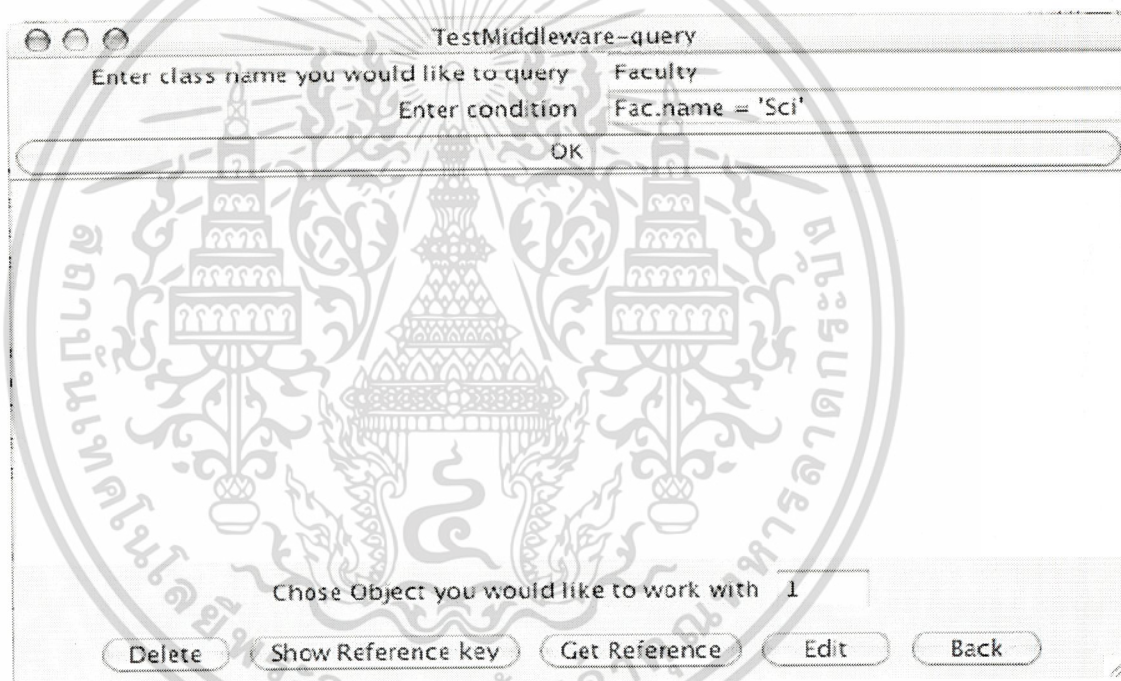
รูปที่ 4.3.3.1 หน้าจอสำหรับเลือกรายการที่ต้องการกระทำ

ขั้นตอนที่ 2 เมื่อผู้ใช้ทำการกดที่คำสั่ง Query หน้าจอจะเปลี่ยนไป โดยมีลักษณะดังนี้

ให้กรอกชื่อ class ที่ต้องการทำการค้นหาข้อมูลเพื่อทำการเปลี่ยนแปลง ตัวอย่างการกรอกข้อมูลคือ

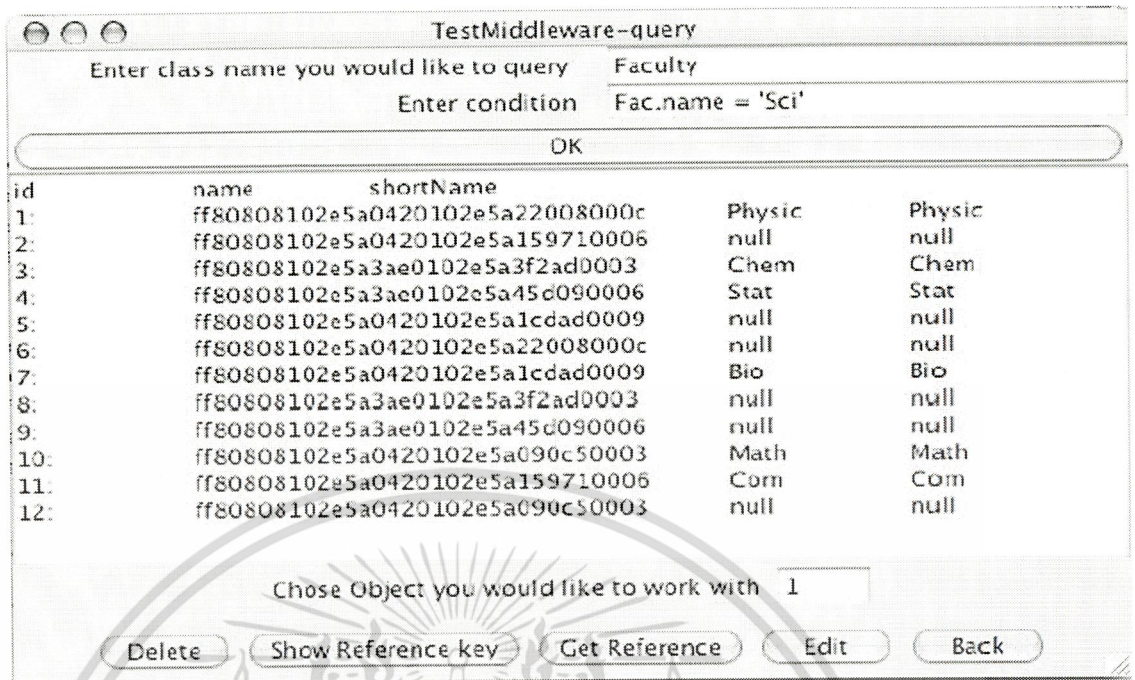
ช่องแรก เป็นช่องให้ระบุชื่อ class ที่ต้องการเปลี่ยนแปลงข้อมูล ซึ่งในที่นี้เราต้องการเปลี่ยนแปลงข้อมูลในคณะวิทยาศาสตร์ ดังนั้น Class ที่ต้องระบุคือ “Faculty”

ช่องที่ 2 เป็นช่องสำหรับระบุ class ที่มี Association แบบ Many – One เช่นหากช่องแรกเป็น Department ตัวอย่างของการกรอกช่องที่ 2 คือ Fac.name = 'Sci'



รูปที่ 4.3.3.2 หน้าจอสำหรับค้นหาข้อมูลเพื่อทำการเปลี่ยนแปลงข้อมูล

หลังจากกรอกข้อมูลในส่วนที่ 1 เรียบร้อยแล้ว ให้ทำการกดปุ่ม “OK” เมื่อกดเรียบร้อยแล้ว ข้อมูลจะถูกดึงขึ้นมาปรากฏดังรูปที่ 4.3.3.3



รูปที่ 4.3.3.3 หน้าจอแสดงผลการค้นหาข้อมูลที่ต้องการทำการเปลี่ยนแปลง

นอกจากนี้หากต้องการดูเพียง Reference key ของ class ที่เราต้องการทำการเปลี่ยนแปลงให้กดที่ปุ่ม "Show Reference key" หน้าจอจะเปลี่ยนไปดังรูปที่ 4.3.3.4



รูปที่ 4.3.3.4 หน้าจอแสดงผล Associate ต่าง ๆ ของ class ที่เราทำการค้นหาเพื่อเปลี่ยนแปลงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3 หลังจากกดปุ่ม “OK” ที่หน้าจอรูปที่ 4.3.3.4 แล้ว หน้าจอจะกลับไปยังรูปที่ 4.3.3.3 เพื่อที่จะทำการเปลี่ยนแปลงข้อมูลให้ทำการกดที่ปุ่ม “Edit”

id	name	shortName	Fac.name	Fac.name
1:	ff80808102e5a0420102e5a22008000c	Physic	Physic	null
2:	ff80808102e5a0420102e5a159710006	null	null	null
3:	ff80808102e5a3ae0102e5a3f2ad0003	Chem	Chem	Chem
4:	ff80808102e5a3ae0102e5a45c090006	Stat	Stat	Stat
5:	ff80808102e5a0420102e5a1cdad0009	null	null	null
6:	ff80808102e5a0420102e5a22008000c	null	null	null
7:	ff80808102e5a0420102e5a1cdad0009	Bio	Bio	Bio
8:	ff80808102e5a3ae0102e5a3f2ad0003	null	null	null
9:	ff80808102e5a3ae0102e5a45c090006	null	null	null
10:	ff80808102e5a0420102e5a090c50003	Math	Math	Math
11:	ff80808102e5a0420102e5a159710006	Com	Com	Com
12:	ff80808102e5a0420102e5a090c50003	null	null	null

Chose Object you would like to work with 1

Buttons: Delete, Show Reference key, Get Reference, Edit, Back

รูปที่ 4.3.3.5 หน้าจอแสดงผลการค้นหาข้อมูลที่ต้องการเปลี่ยนแปลง

ขั้นตอนที่ 4 เมื่อทำการกดที่ปุ่ม “Edit” หน้าจอจะเปลี่ยนไปดังรูปที่ 4.3.3.6

Enter value of attribute <attribute is a must>

*ชื่อภาควิชา Mr.

*ชื่อภาควิชา Sutipan

Buttons: Back, Link to persist..., Update

รูปที่ 4.3.3.6 หน้าจอสำหรับระบุรายละเอียดที่ต้องการเปลี่ยนแปลงข้อมูล

จากนั้นให้กดปุ่ม “Link to persist object” เพื่อใส่ค่าในกรณีที่มี Association กับ class อื่นๆ แบบ Many – One ซึ่งมีรายละเอียดดังรูป 4.3.3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 5 จากนั้นให้ผู้ใช้ระบุรายละเอียดต่อไปนี้

ช่องแรก Class name ให้ผู้ใช้ระบุชื่อ class ที่มี association ในที่นี้คือ Faculty

ช่องที่ 2 Value ให้ผู้ใช้ระบุค่าสำหรับ Attribute ที่เป็น Foreign Key ตัวอย่างรูปแบบสำหรับการกรอกข้อมูล คือ Fac.name = 'Sci'

จากนั้นให้ผู้ใช้กดปุ่ม Set to Reference หากต้องการเปลี่ยนแปลงข้อมูลหรือกดปุ่ม Back หากผู้ใช้ต้องการกลับไปยังหน้าจอหลัก

รูป 4.3.3.7 หน้าจอสำหรับกรอกค่าให้แก่ Class ที่มี Association กับ Class ที่ต้องการเปลี่ยนแปลงข้อมูล

เมื่อทำตามขั้นตอนที่ 5 แล้ว หน้าจอจะย้อนกลับไปยังรูปที่ 4.3.3.6

ขั้นตอนที่ 6 กดปุ่ม "Update" เพื่อทำการบันทึกข้อมูลที่เปลี่ยนแปลงลง

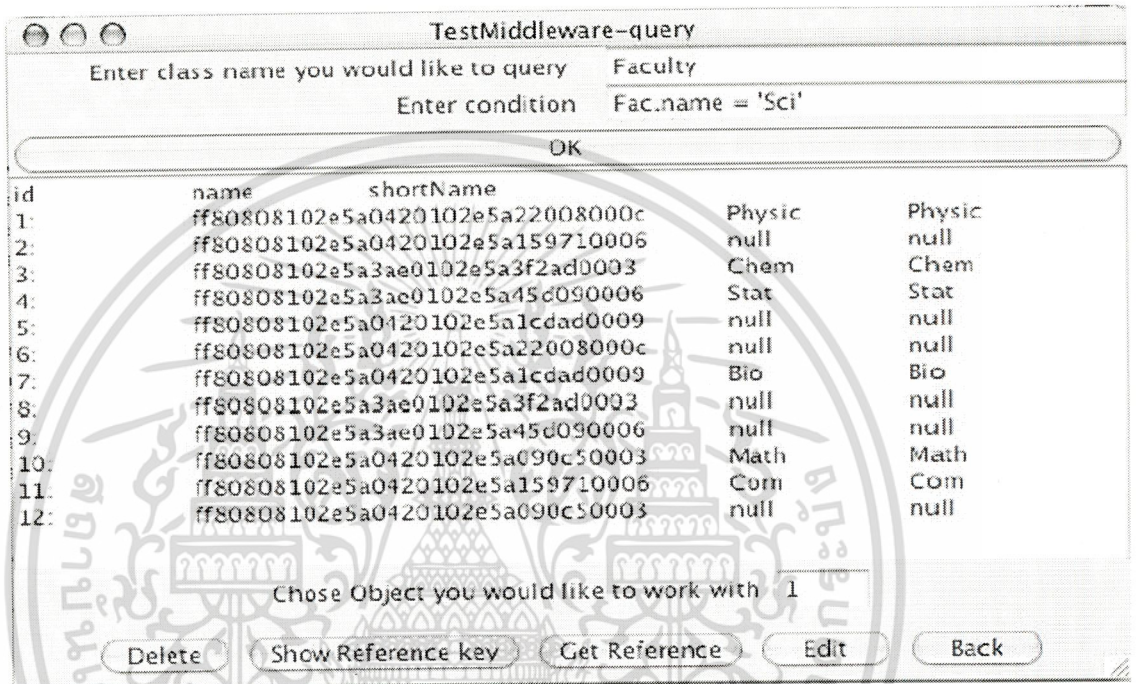
ฐานข้อมูล

รูปที่ 4.3.3.8 หน้าจอสำหรับระบุรายละเอียดที่ต้องการเปลี่ยนแปลงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การทดสอบการลบข้อมูล

หลังจากการทำ query ในหัวข้อ 4.3.2 เมื่อผู้ใช้ต้องการทำการลบข้อมูลให้ผู้ใช้ทำการเลือกตำแหน่งของข้อมูล ตามภาพ 4.3.3.1 ที่ช่อง "Chose Object you would like to work with" จากนั้นให้กดที่ปุ่มคำสั่ง "Delete" ข้อมูลจะถูกลบออกจากฐานข้อมูล



รูปที่ 4.3.4.1 หน้าจอสำหรับการลบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการพัฒนาโปรแกรม

5.1 การออกแบบและผลที่ได้จากการพัฒนาโปรแกรม

จากการศึกษาเพื่อออกแบบและพัฒนาไมดเดิลแวร์สำหรับส่งผ่านข้อมูลในระบบ Client/Server โดยใช้วิธีการส่งผ่านข้อมูลแบบออบเจกต์พาสซึ่ง สามารถสรุปได้ดังนี้

5.1.1 การออกแบบและพัฒนาโปรแกรม

ในส่วนของการพัฒนาองค์ประกอบต่างๆของไมดเดิลแวร์โดยในส่วนของการติดต่อระหว่างไคลเอนท์ และเซิร์ฟเวอร์นั้น ใช้วิธีการส่งออบเจกต์สตรีม หรือกลุ่มข้อมูลที่ถูกรวมกันที่ถูกส่งไปอย่างต่อเนื่อง เพื่อที่จะทำการจัดการกับข้อมูลตามที่ไคลเอนท์ร้องขอ โดยข้อมูลจะอยู่ในรูปของเอกสารเอ็กซ์เอ็มแอล ซึ่งการจัดการกับข้อมูลอาจเป็นคำสั่ง Insert Update Delete หรือ Select ก็ได้ โดยในการติดต่อกับโปรแกรมไมดเดิลแวร์เพื่อทำการ Insert Update Delete หรือ Select นั้น จะใช้การติดต่อโดยส่งผ่านข้อมูลในรูปแบบของออบเจกต์แทนการติดต่อในรูปแบบของการใช้ภาษา SQL และได้พัฒนาให้สามารถเปลี่ยนรูปแบบข้อมูลได้ทั้งจากออบเจกต์ให้เปลี่ยนเป็นรูปแบบของเอกสารเอ็กซ์เอ็มแอล และเปลี่ยนรูปแบบของเอกสารเอ็กซ์เอ็มแอลให้เปลี่ยนเป็นรูปแบบออบเจกต์ได้ รวมทั้งได้มีการพัฒนาในส่วนของการติดต่อกับ Hibernate ซึ่งจะช่วยให้สามารถทำการ Insert Update Delete หรือ Select ไปยังฐานข้อมูลได้ และพัฒนาโปรแกรมในส่วนของฝั่งไคลเอนท์สำหรับทดสอบการทำงานของไมดเดิลแวร์ที่พัฒนาขึ้น

5.1.2 คุณสมบัติของโปรแกรม

ในส่วนของการพัฒนาไมดเดิลแวร์ซึ่งทำหน้าที่เป็นส่วนเชื่อมต่อเพื่อส่งผ่านข้อมูลระหว่างฝั่ง Client และ Server โดยใช้การส่งผ่านข้อมูลแบบออบเจกต์พาสซึ่ง ทำให้โปรแกรมนี้มีคุณสมบัติดังนี้

1. การส่งผ่านข้อมูลระหว่างไคลเอนท์และเซิร์ฟเวอร์ จะใช้การส่งข้อมูลในรูปของออบเจกต์สตรีม ซึ่งเป็นรูปแบบที่มีการรวมข้อมูลให้เป็นกลุ่มไว้ในรูปของเอกสารเอ็กซ์เอ็มแอล สำหรับใช้ในการ Insert Update Delete และ Select
2. การจัดการกับข้อมูลทั้งคำสั่ง Insert Update Delete และ Select จะใช้รูปแบบเชิงวัตถุ แทนการใช้ภาษา SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ผลลัพธ์ที่ได้จากการจัดการหรือค้นหาข้อมูล จะประกอบไปด้วยค่าแอททริบิวต์ต่างๆ ที่มีอยู่ในออบเจกต์ รวมทั้งค่าที่ใช้ในการอ้างอิงในการค้นหา หรือเป็น primary key ของออบเจกต์ที่มีความสัมพันธ์กันกับออบเจกต์ที่ถูกทำการจัดการหรือค้นหาข้อมูล

5.2 ข้อจำกัดของโปรแกรม

จากการออกแบบและพัฒนาโปรแกรมมิดเดิลแวร์ขึ้นนั้น ในส่วนของโปรแกรมายังมีบางส่วนที่เป็นข้อจำกัดของการใช้โปรแกรม ได้แก่

1. ในส่วนของการ Insert และ Update จะสามารถทำได้เฉพาะกับออบเจกต์หลักที่สั่ง Insert และ Update เท่านั้น จะไม่สามารถ cascade การ Insert และ Update นี้ไปยังออบเจกต์อื่นๆ แม้ว่าจะมีความสัมพันธ์กับออบเจกต์นั้นก็ตาม

2. ในส่วนของการลบข้อมูล หรือ Delete นั้น เมื่อทำการสั่งลบค่าใดค่าหนึ่งภายในออบเจกต์ จะเกิดการ cascade การ Delete ไปยังออบเจกต์ทุกออบเจกต์ที่มีความสัมพันธ์ต่อกัน กล่าวคือ หากออบเจกต์หนึ่งถูกสั่งให้ลบค่าใดค่าหนึ่งภายในออบเจกต์นั้น ออบเจกต์อื่นๆที่มีความสัมพันธ์ต่อกันกับออบเจกต์ที่ถูกสั่งให้ลบค่านั้นจะถูกทำการลบค่านั้นๆ ทั้งหมดไปด้วย

3. โปรแกรมที่พัฒนายังมิได้คำนึงถึงประสิทธิภาพของอัลกอริทึมที่ใช้

5.3 ข้อเสนอแนะในการพัฒนาโปรแกรม

จากการออกแบบและพัฒนาโปรแกรมในส่วนที่จัดทำขึ้นนั้น ยังสามารถที่จะพัฒนาโปรแกรมให้มีขีดความสามารถเพิ่มขึ้นได้อีก โดยสามารถนำไปพัฒนาส่วนต่างๆ ได้แก่

1. สามารถนำ DTDs ที่เป็นมาตรฐานมาทำให้ระบบมีศักยภาพเพิ่มขึ้น สามารถรองรับข้อมูลได้ครบถ้วนและสมบูรณ์มากขึ้น ซึ่ง DTDs ที่ใช้จะมีความซับซ้อนมากขึ้น

2. สามารถทำการเพิ่มคุณสมบัติของมิดเดิลแวร์ ให้สามารถรองรับความสามารถของ OODMBS ได้ อย่างเช่น สามารถนำข้อมูลที่ผ่านการดำเนินการแล้วในขบวนการหนึ่ง ให้สามารถนำมาใช้กับขบวนการอื่นได้อีกโดยไม่ต้องทำการ Query ใหม่อีกครั้ง เป็นต้น

3. สามารถขยายคุณสมบัติในการ Insert และ Update ให้สามารถ cascade การ Insert และ Update นั้นไปยังออบเจกต์ทั้งหมดที่มีความสัมพันธ์ต่อกันได้

4. สามารถจำกัดให้การ Delete ไม่มีผลต่อออบเจกต์ที่มีความสัมพันธ์เป็น parent ของออบเจกต์นั้นได้

บรรณานุกรม

สมนึก เจียมเจริญเดช. "เจาะลึก Oracle9i รีลีส 2" คณะบุคคล เอส.พี.บุ๊ค, 2546.

Michael V. Mannino. "Database Design, Application Development, & Administration"
The McGraw-Hill Companies, 2004.

Robert Orfali , Dan Harkey , Jeri Edwards. "Client/Server Survival Guide Third Edition"
John Wiley & Sons, Inc. 1999.

E. Reed Doke , John W. Satzinger , Susan Rebstock Williams. "Object-Oriented
Application Development Using Java" Course Technology, 2002

Joe Salemi , "Client/Server Computing with Oracle" Ziff-Davis Press, 1993.



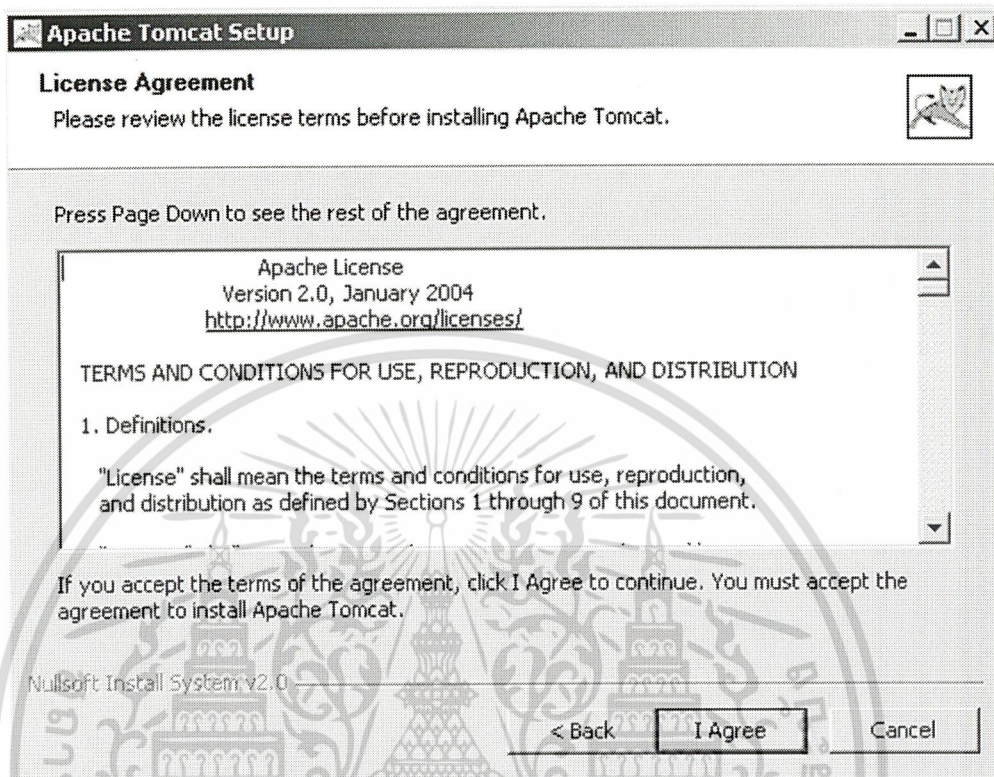
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.
การติดตั้งระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

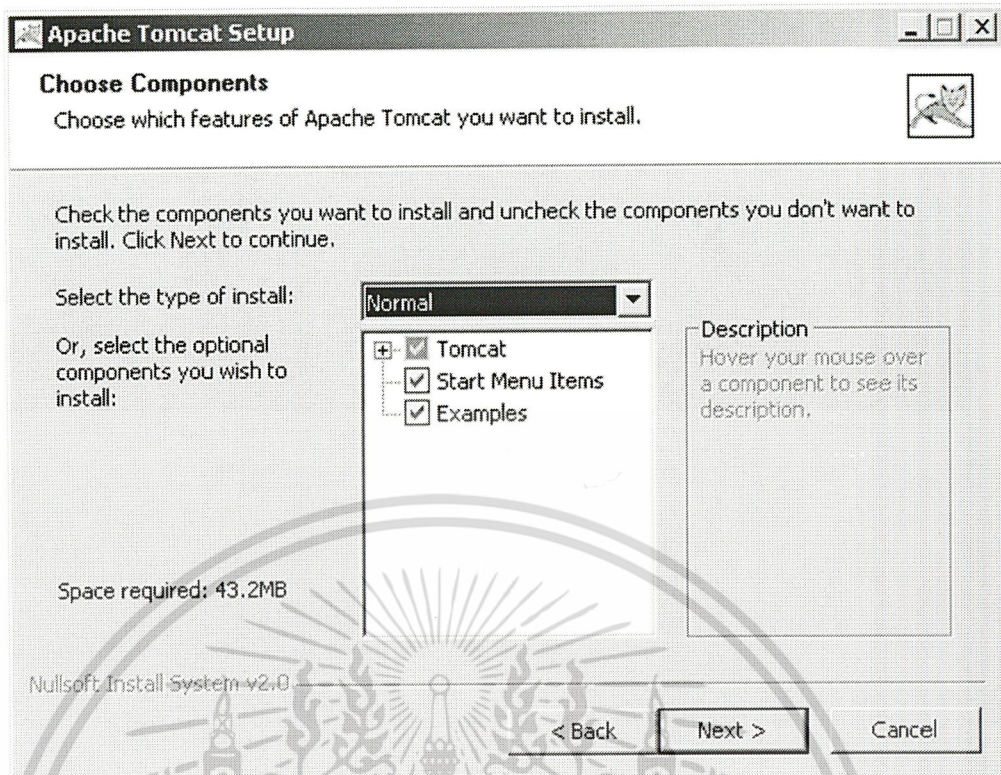
การติดตั้ง Apache Tomcat ลงเครื่องคอมพิวเตอร์ฝั่ง Server



รูปที่ 1-ก หน้าจอเริ่มต้นในการติดตั้ง Apache Tomcat

1. ในหน้าจอเริ่มต้นของการติดตั้งระบบ Apache Tomcat จะมีเนื้อหาสัญญาที่ให้ผู้ใช้ได้อ่านเพื่อรับรู้ส่วนต่างๆ ของสัญญาตามที่เจ้าของผู้ผลิตกำหนด โดยหากผู้ใช้อ่านแล้วยอมรับตามเงื่อนไขให้ผู้ใช้กดปุ่ม "I Agree"

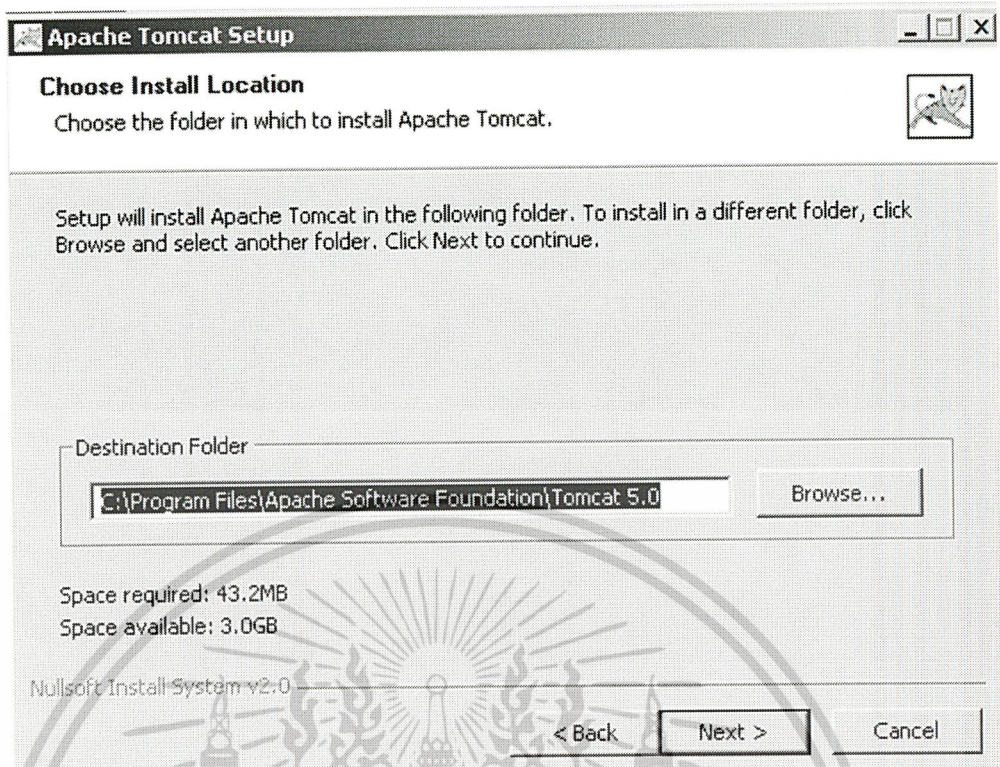
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-ก การเลือกซอฟต์แวร์ Tomcat เพื่อติดตั้งระบบ

2. หลังจากนั้นหน้าจอจะเปลี่ยนให้ผู้ใช้เลือกรูปแบบที่มีใน Apache Tomcat ที่ต้องการทำการติดตั้งลงไปยังเครื่องคอมพิวเตอร์ เมื่อเลือกแล้วผู้ใช้ต้องทำการกดปุ่ม "Next"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

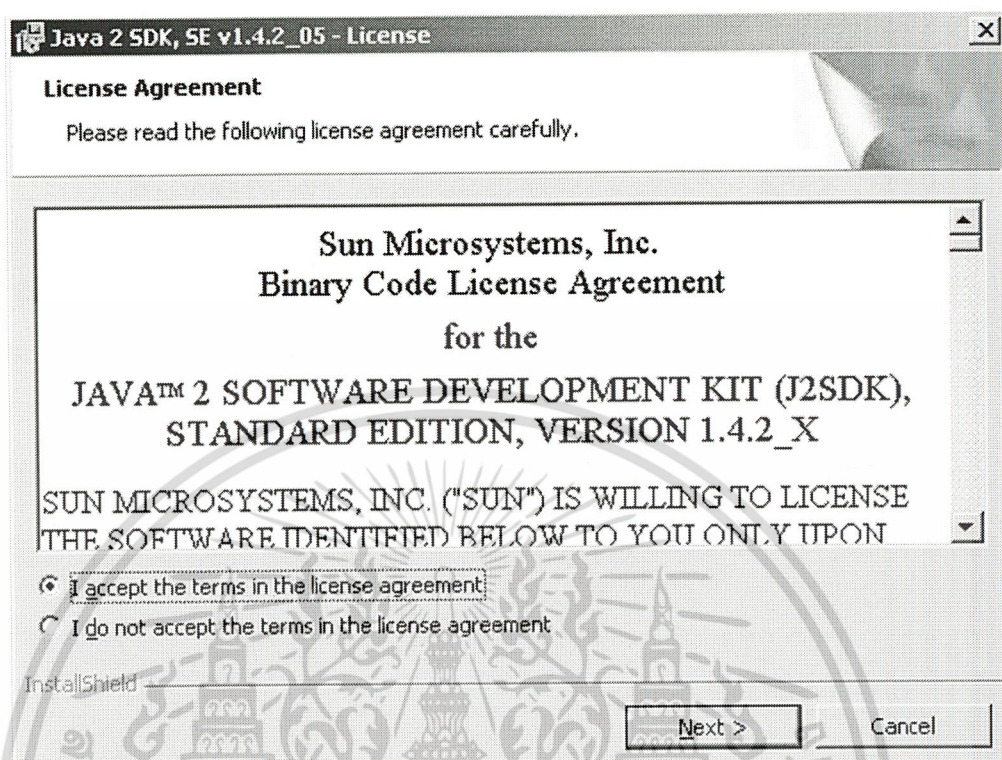


รูปที่ 3-ก การเลือกสถานที่สำหรับการติดตั้งระบบ

3. เมื่อเลือกรูปแบบในการติดตั้งเรียบร้อยแล้วหน้าจอจะเปลี่ยนเป็นส่วนที่ให้ผู้ใช้งานทำการเลือกสถานที่ที่ผู้ใช้ต้องการติดตั้งระบบลง โดยทำการกดปุ่ม "Browse..." และเลือกสถานที่ที่ต้องการ เมื่อทำการกำหนดสถานที่เรียบร้อยแล้วจึงกดปุ่ม "Next"

เมื่อกดปุ่ม "Next" แล้วระบบก็就会被ทำการติดตั้งโดยอัตโนมัติ

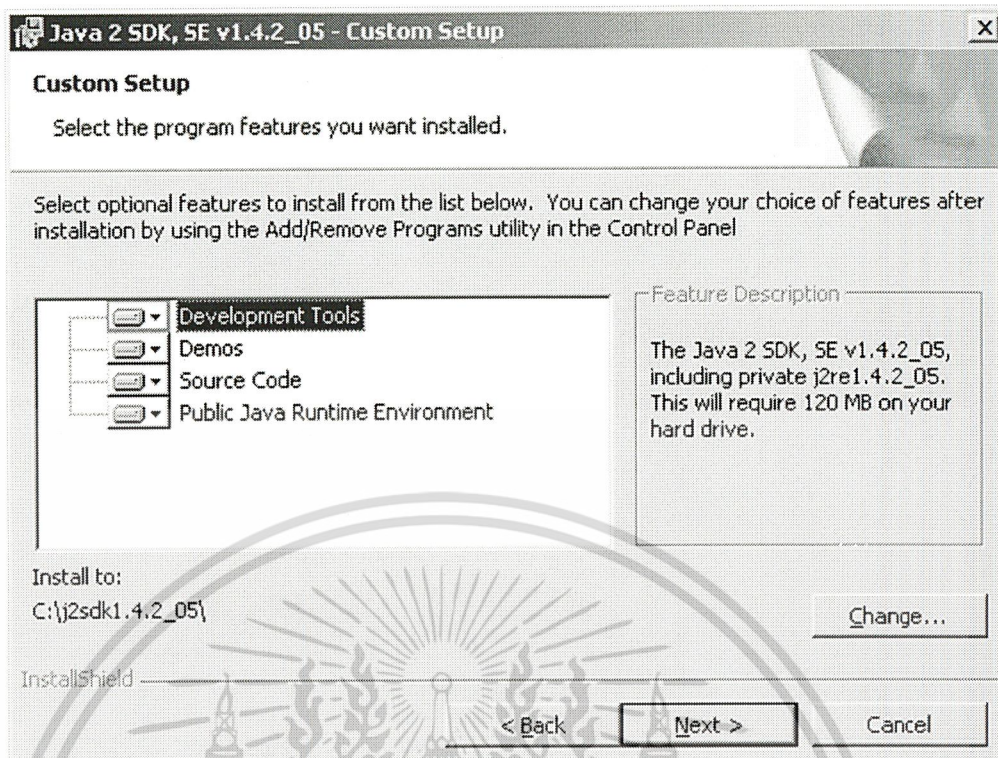
การติดตั้งระบบ Java 2 Software Development Kit ลงยังเครื่อง Server



รูปที่ 4-ก หน้าจอแสดงสัญญาที่ผู้ใช้ต้องยอมรับก่อนทำการติดตั้ง

1. ในหน้าจอเริ่มต้นของการติดตั้งระบบ Java 2 Software Development Kit จะมีเนื้อหาสัญญาที่ให้ผู้ใช้อ่านเพื่อรับรู้ส่วนต่างๆ ของสัญญาตามที่เจ้าของผู้ผลิตกำหนด โดยหากผู้ใช้อ่านแล้วยอมรับตามเงื่อนไขให้ผู้กดให้ในส่วนวงกลมของประโยค "I accept the the teams in the license agreement" จากนั้นให้กดปุ่ม "Next"

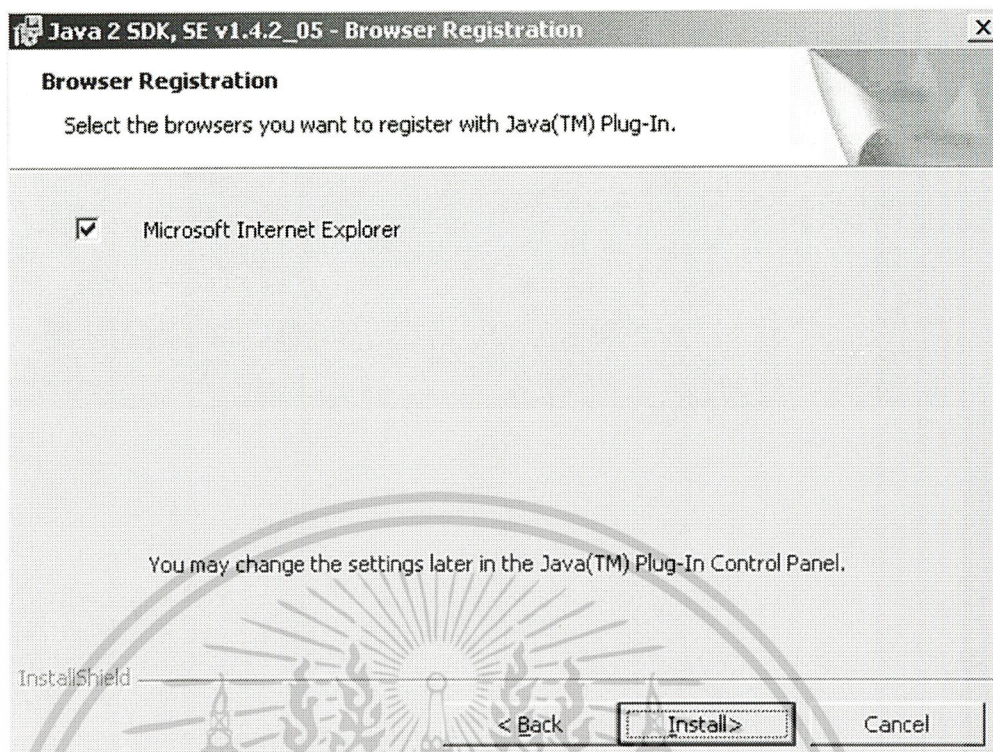
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-ก หน้าจอสำหรับเลือกโปรแกรมที่ต้องการทำการติดตั้ง

2. สำหรับส่วนนี้ให้ผู้ใช้คลิกเลือกโปรแกรมที่ต้องการติดตั้ง จากนั้นกดปุ่ม "Next"

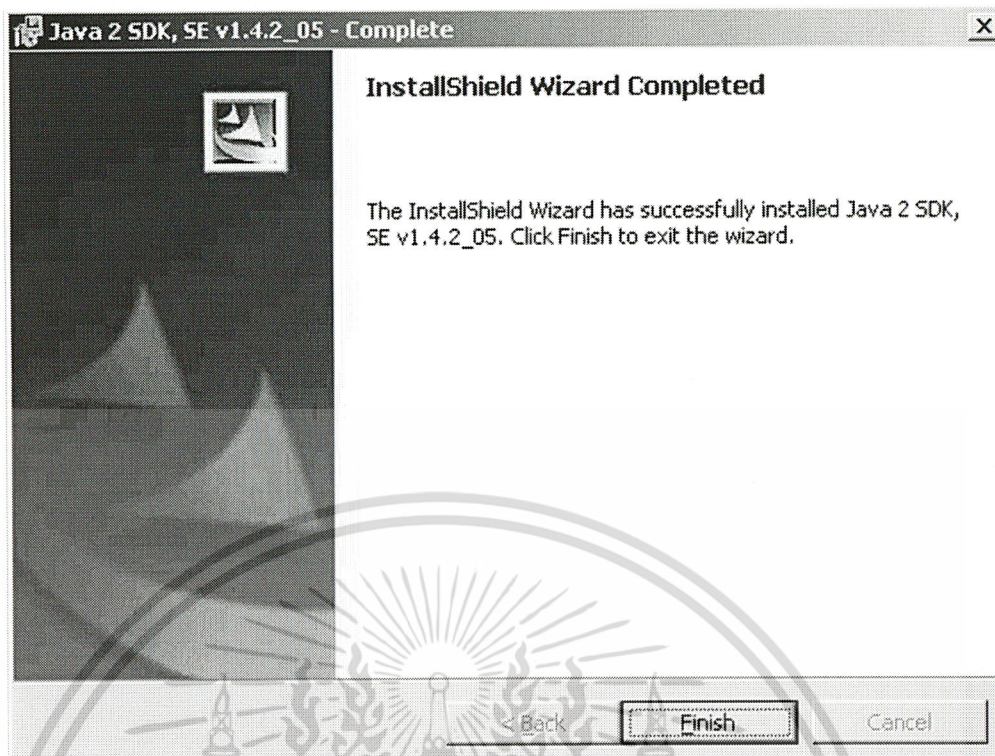
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-ก หน้าจอสำหรับเลือกเบราว์เซอร์ที่ต้องการ

3. ในหน้าจอนี้จะแสดงเพื่อให้ผู้ใช้เลือกเบราว์เซอร์ที่ต้องการ โดยกดที่ช่องสี่เหลี่ยมก่อนหน้าประโยค "Microsoft Internet Explorer" แล้วจึงกดปุ่ม "Install"

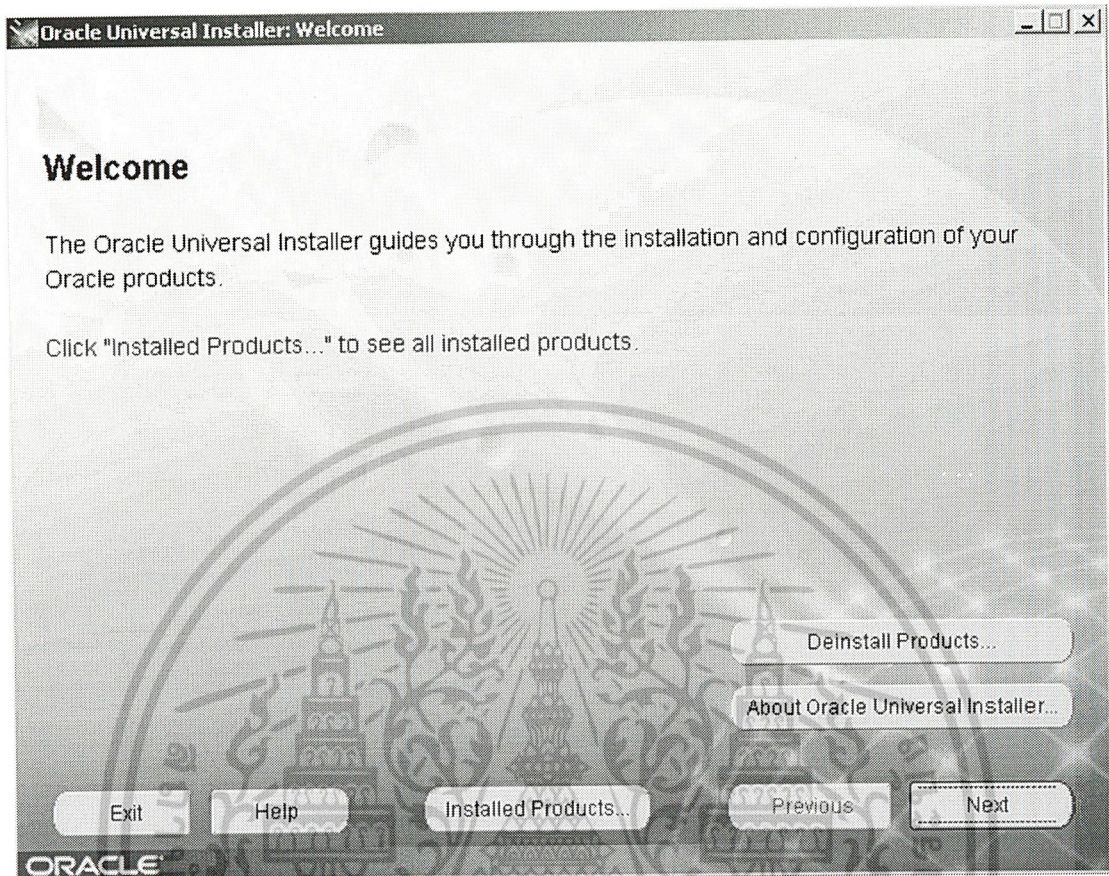
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-ก หน้าจอแสดงผลการติดตั้ง Java 2 Software Development Kit

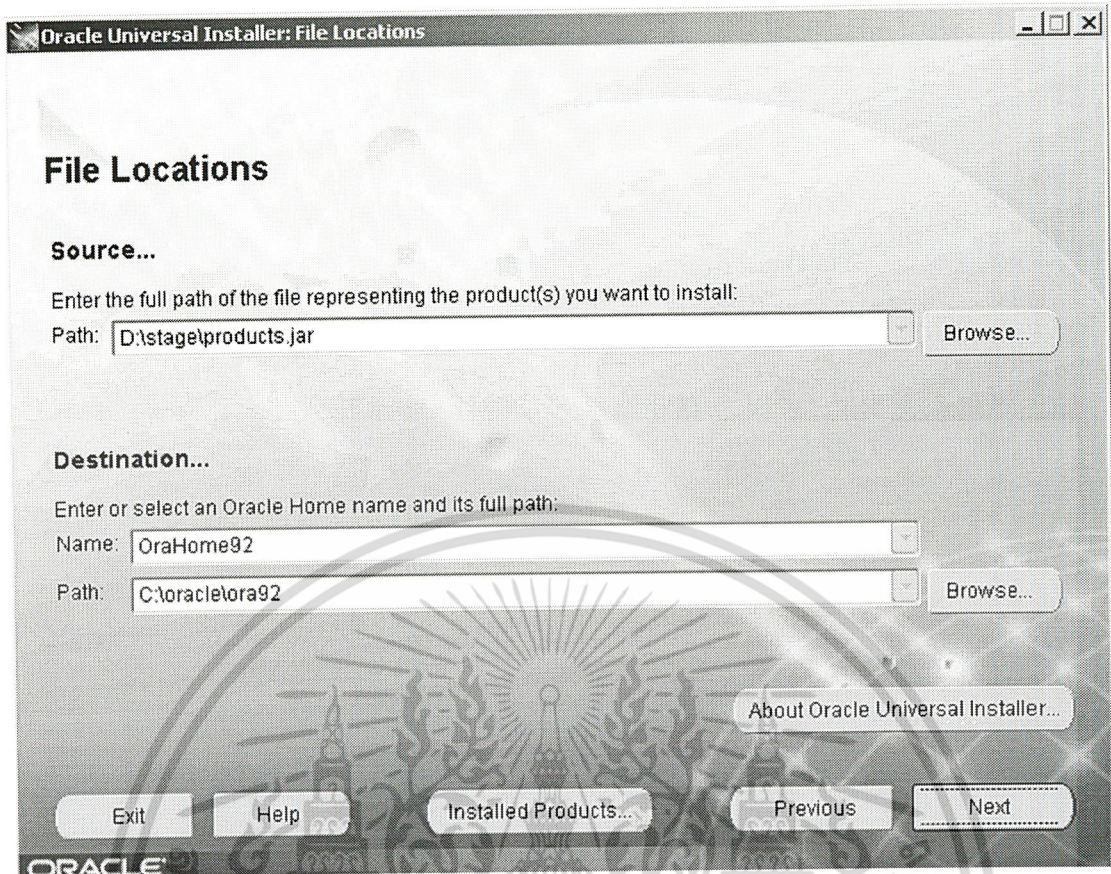
4. หน้าจอนี้จะบอกถึงสถานการณ์ติดตั้งว่าเรียบร้อยแล้ว เพื่อที่จะออกจากการติดตั้งนี้ให้กดปุ่ม "Finish"

การติดตั้งระบบ Oracle ลงยังเครื่อง Server



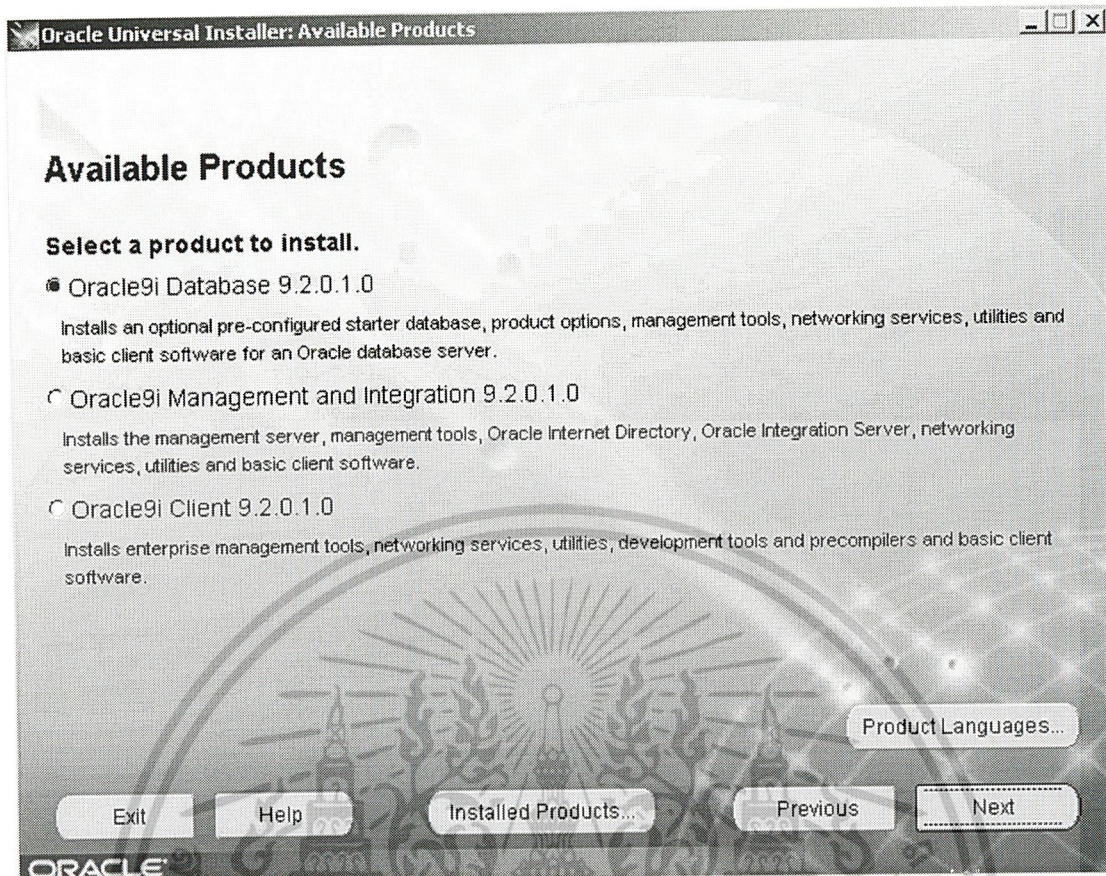
รูปที่ 8-ก หน้าจอต้อนรับเข้าสู่การติดตั้ง Oracle

1. หน้าจอนี้เป็นหน้าจอเริ่มต้นในการติดตั้ง Oracle เพื่อที่จะดำเนินการติดตั้งต่อไปให้กดปุ่ม "Next"



รูปที่ 9-ก หน้าจอเลือกสถานที่สำหรับติดตั้งโปรแกรม Oracle

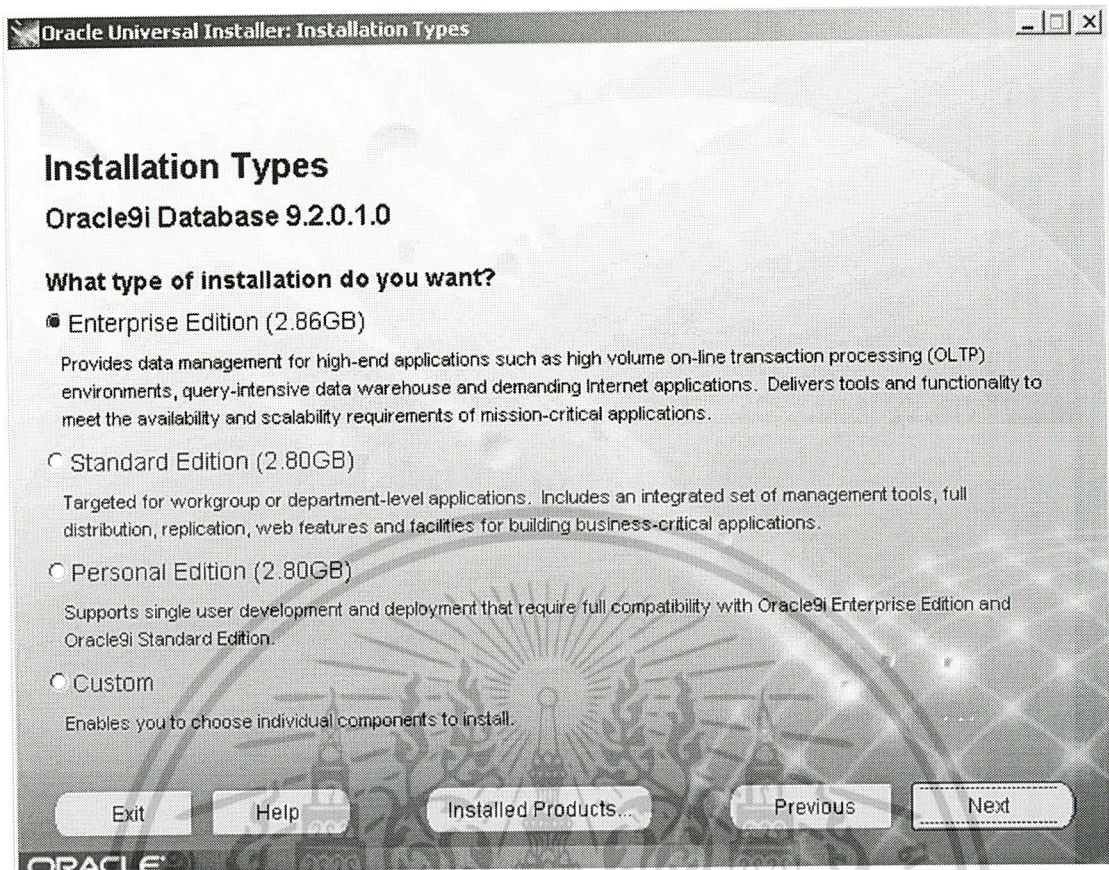
2. หน้าจอนี้จะให้ผู้ใช้ทำการกำหนดสถานที่ที่เป็นส่วนของต้นทางของโปรแกรมและกำหนดสถานที่ที่เป็นส่วนปลายทางที่ต้องการสำหรับติดตั้งโปรแกรม Oracle ลงยังเครื่อง



รูปที่ 10-ก หน้าจอสำหรับเลือกรูปแบบผลิตภัณฑ์ของ Oracle ที่ต้องการ

3. ให้ผู้ใช้เลือกผลิตภัณฑ์ที่ต้องการติดตั้งโดยกดที่วงกลมหน้าประโยค "Oracle9i Database 9.2.0.1.0" จากนั้นให้กดปุ่ม "Next"

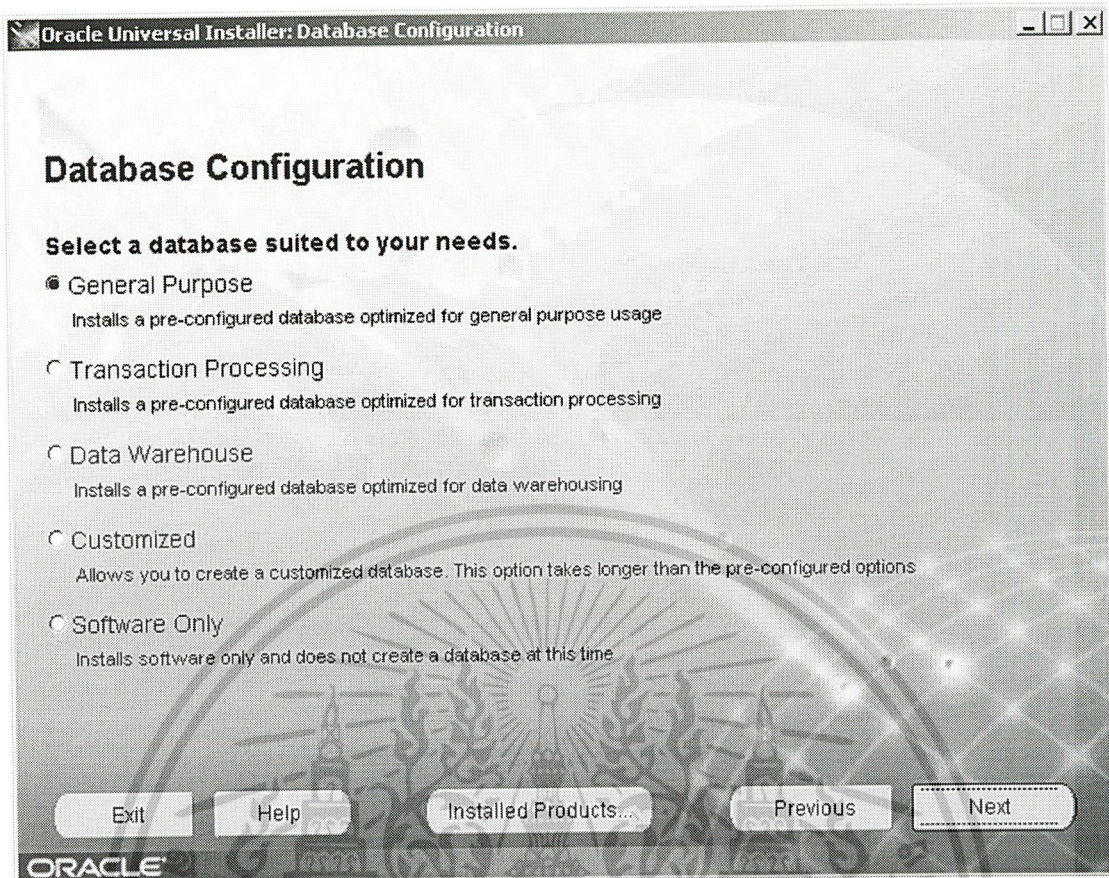
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11-ก หน้าจอสำหรับเลือกชนิดของ Oracle ที่ต้องการ

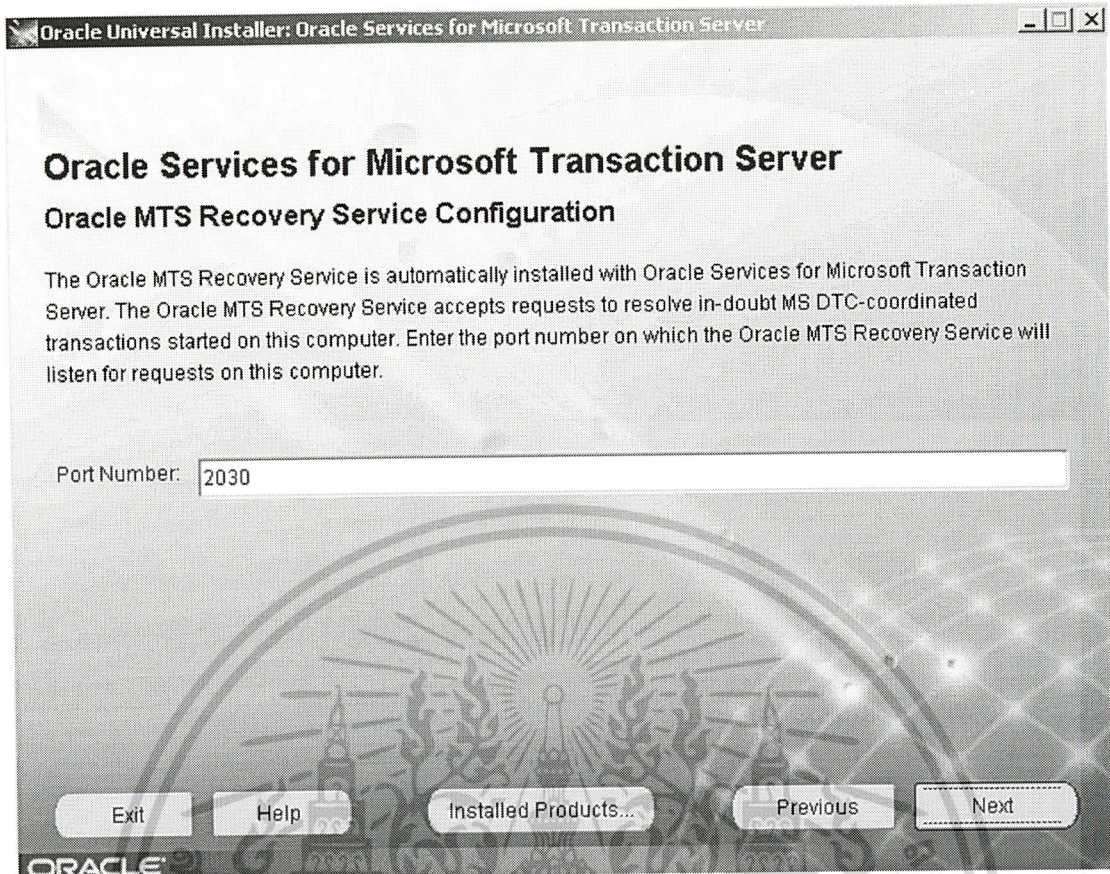
4. ให้ผู้ใช้เลือกชนิดของ Oracle ที่ต้องการติดตั้ง โดยในส่วนนี้จะกำหนดให้เป็น Enterprise Edition โดยกดที่วงกลมหน้าประโยค "Enterprise Edition" จากนั้นให้กดปุ่ม "Next"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



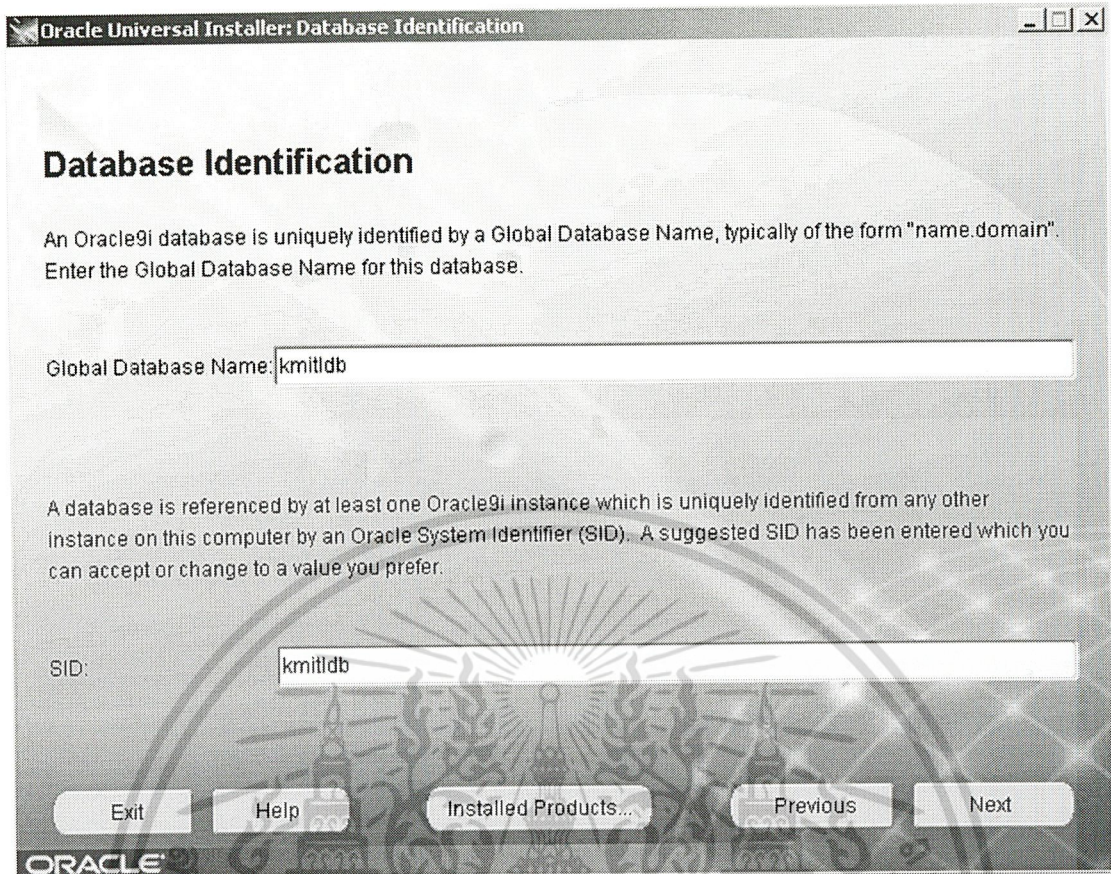
รูปที่ 12-ก หน้าจอสำหรับเลือกรูปแบบของฐานข้อมูลสำหรับร้องขอความต้องการ

5. ให้ผู้ใช้เลือกรูปแบบของฐานข้อมูลสำหรับร้องขอความต้องการ โดยในส่วนี้จะกำหนดให้เป็น General Purpose โดยกดที่วงกลมหน้าประโยค "General Purpose" จากนั้นให้กดปุ่ม "Next"



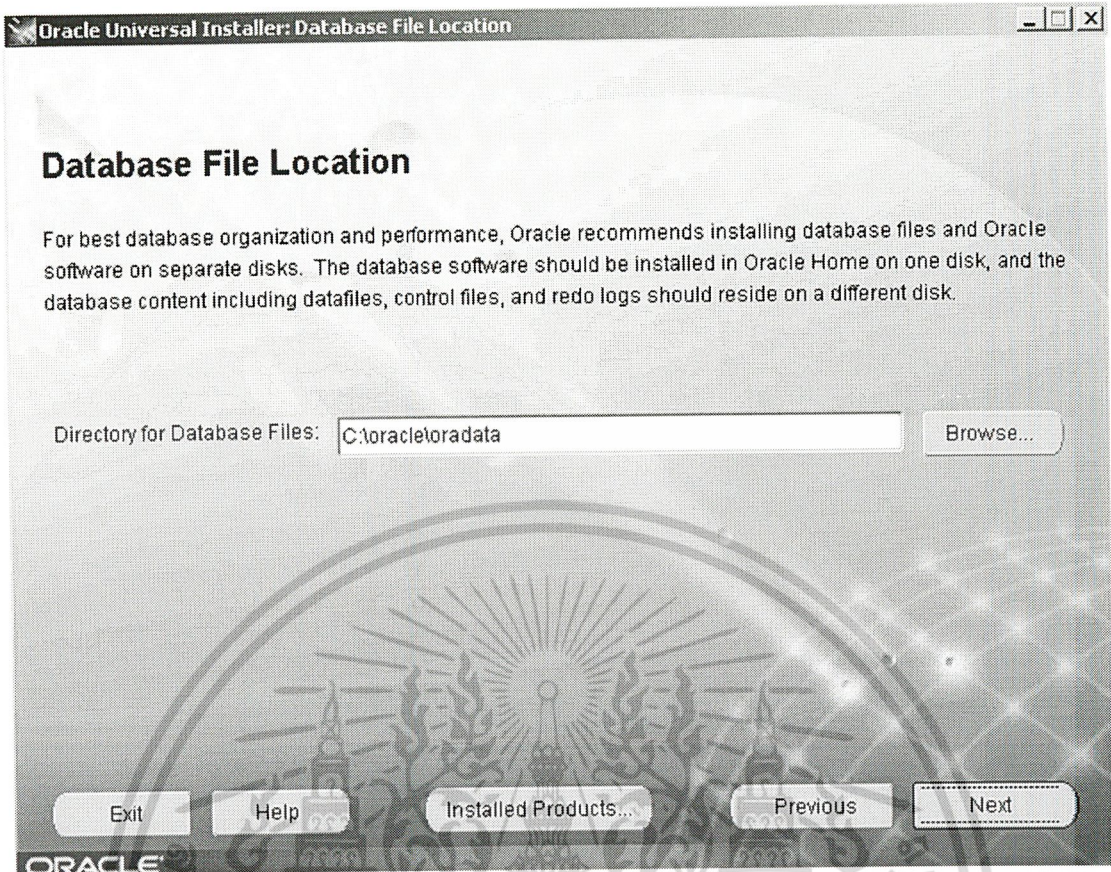
รูปที่ 13-ก หน้าจอสำหรับใส่หมายเลขของพอร์ต

6. หน้าจอนี้ให้ผู้ใช้ระบุหมายเลขพอร์ตที่จะมีหน้าที่รับคำสั่งบนเครื่อง Server ในหน้าจอนี้จะระบุพอร์ตหมายเลข 2030 จากนั้นกดปุ่ม “Next”



รูปที่ 14-ก หน้าจอสำหรับกำหนดชื่อและเอสไอดีสำหรับฐานข้อมูล

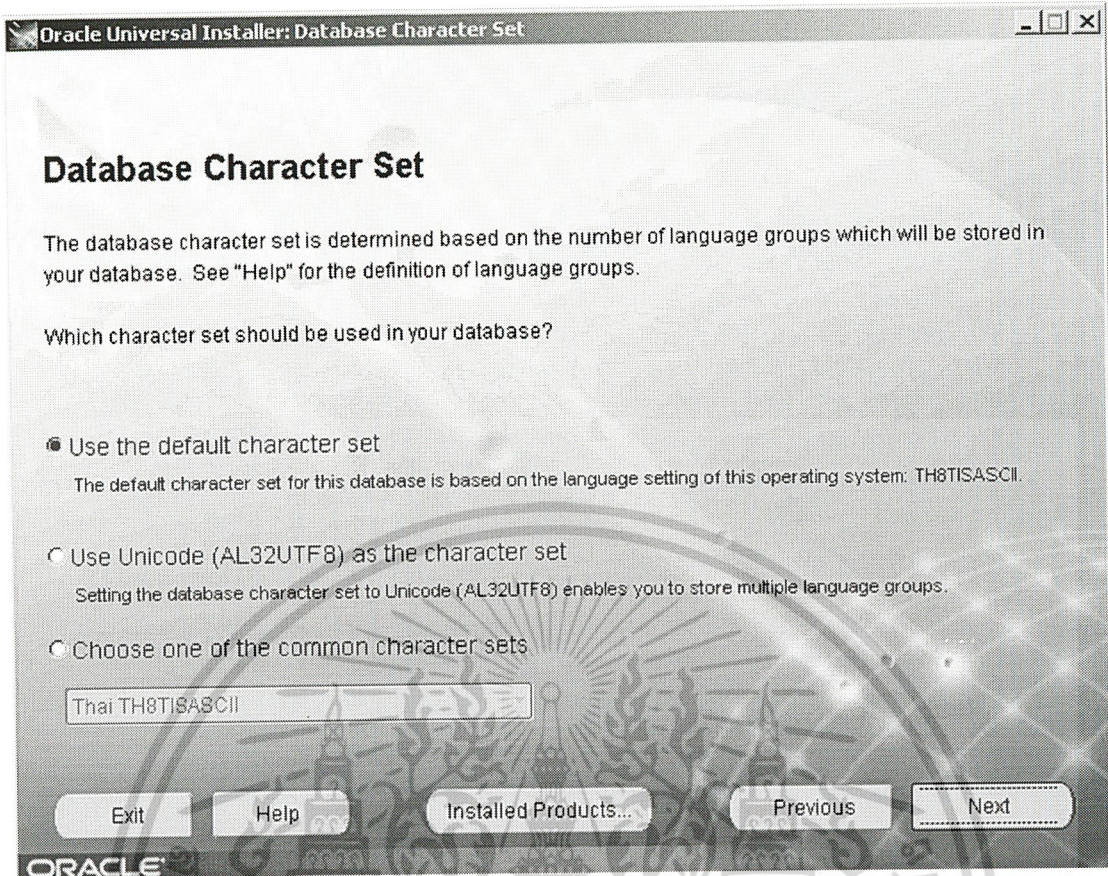
7. หน้าจอสำหรับกำหนดชื่อและเอสไอดีสำหรับฐานข้อมูล โดยในหน้าจอนี้จะกำหนดให้ทั้งชื่อและเอสไอดีเป็น "kmitldb" จากนั้นกดปุ่ม "Next"



รูปที่ 15-ก หน้าจอสำหรับกำหนดสถานที่ของไฟล์ฐานข้อมูล

8. ในหน้าจอนี้ให้ผู้ใช้กำหนดตำแหน่งสถานที่ของไฟล์ฐานข้อมูล แล้วกดปุ่ม "Next"

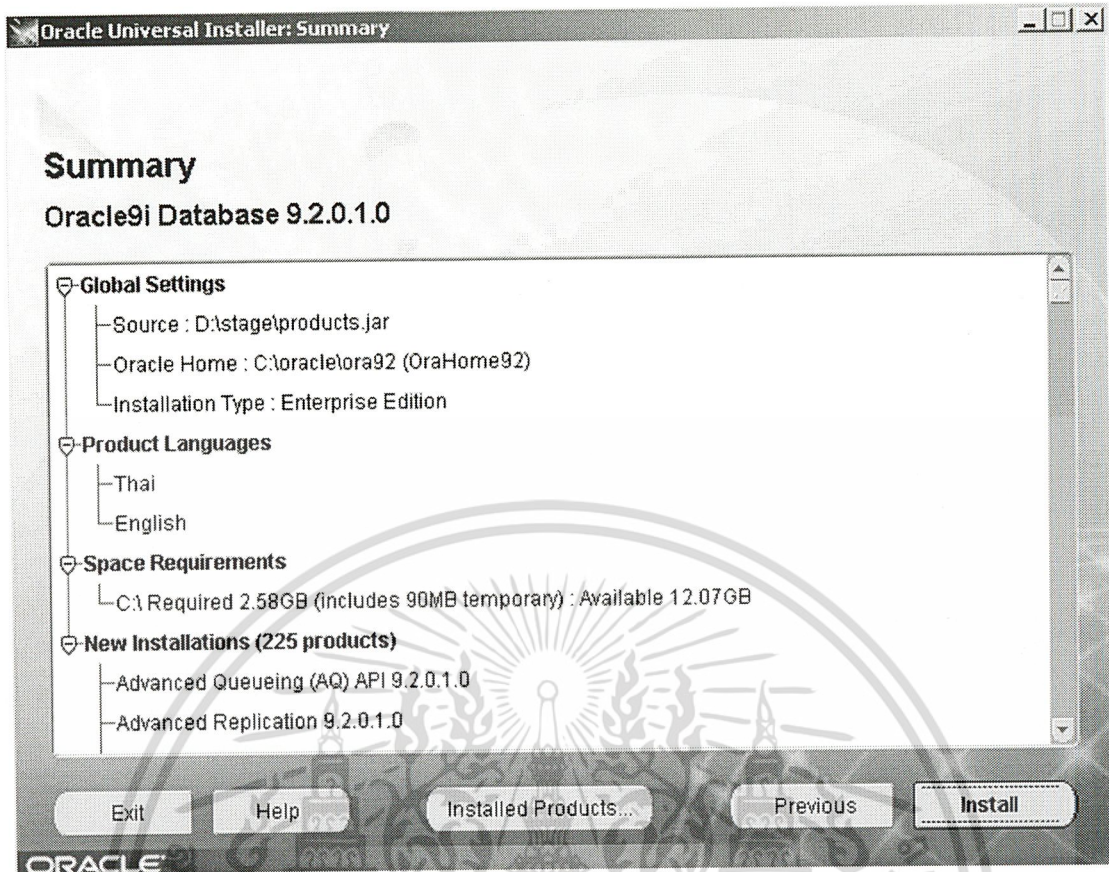
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 16-ก หน้าจอสำหรับกำหนดกลุ่มของภาษาที่สามารถทำการจัดเก็บลงฐานข้อมูล

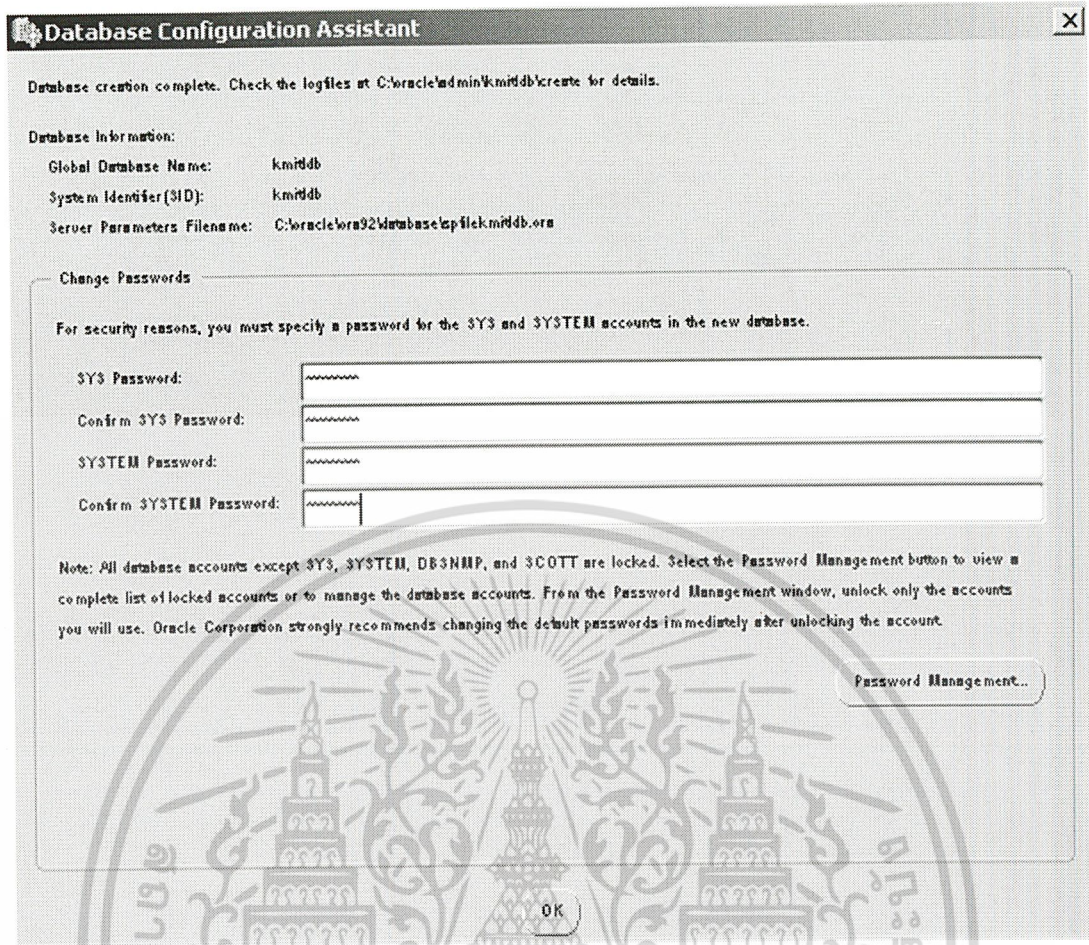
9. หน้าจอสำหรับกำหนดกลุ่มของภาษาที่สามารถทำการจัดเก็บลงฐานข้อมูลได้ โดยใน ส่วนนี้จะกำหนดได้โดยกดที่วงกลมหน้าประโยค "Use the default character set" "General Purpose" จากนั้นให้กดปุ่ม "Next"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



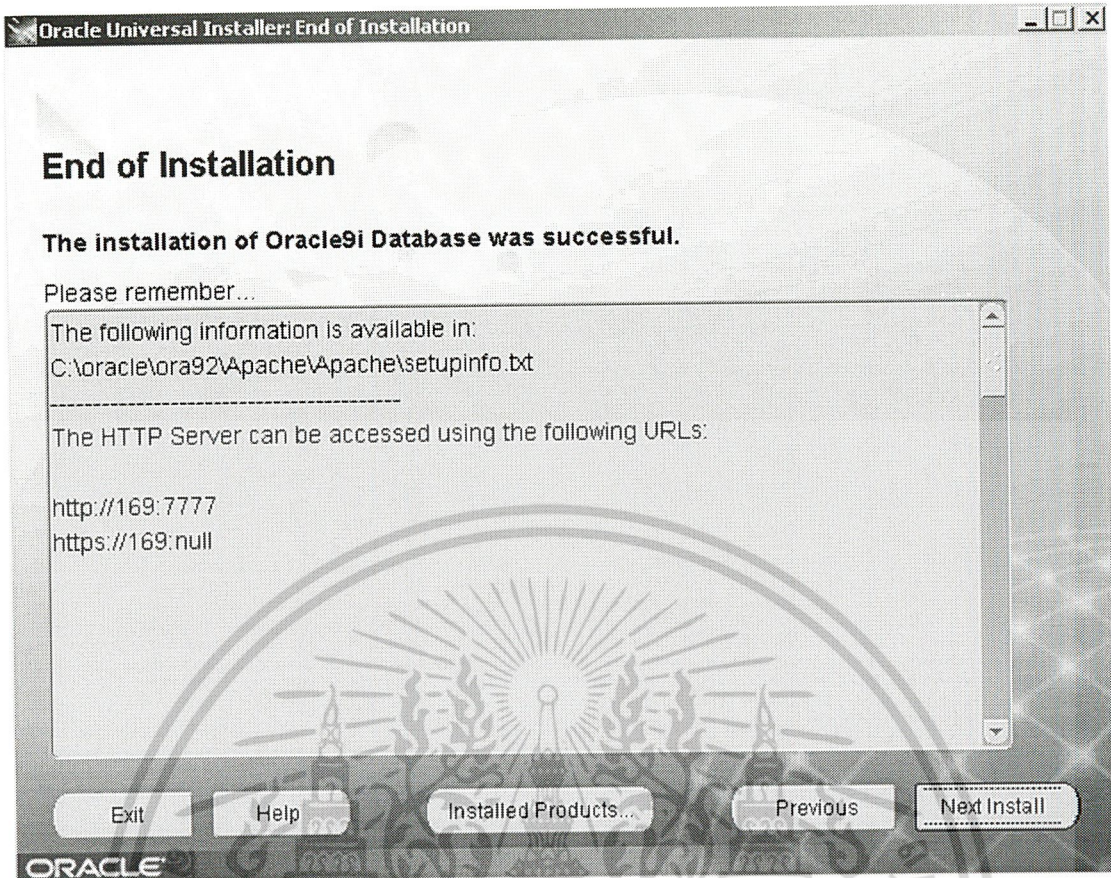
รูปที่ 17-ก หน้าจอแสดงผลสรุปของส่วนต่างๆ ของ Oracle ที่ต้องการติดตั้ง

10. ในหน้าจอนี้จะแสดงผลสรุปของส่วนต่างๆ ของ Oracle ที่ต้องการติดตั้ง ซึ่งผู้ใช้ต้องทำการตรวจสอบความถูกต้องก่อนทำการกดปุ่ม "Install"



รูปที่ 18-ก หน้าจอสำหรับสร้างระบบป้องกันของระบบ Oracle

11. หน้าจอนี้เป็นการสร้างระบบป้องกันโดยให้ผู้ใช้กำหนดรหัส SYS และ SYSTEM เมื่อกำหนดแล้วให้กดปุ่ม "OK"



รูปที่ 19-ก หน้าจอแสดงผลการติดตั้ง Oracle

12. หน้าจอนี้จะบอกถึงสถานการณ์ติดตั้งว่าเรียบร้อยแล้ว เพื่อที่จะออกจากการติดตั้งนี้ให้กดปุ่ม "Exit" หรือถ้าต้องการติดตั้งระบบอื่นอีกก็ให้กดปุ่ม "Next Install"



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ Set JDBC Data Source

web application นั้นจำเป็นต้องติดต่อกับฐานข้อมูลผ่าน JDBC driver เพื่อสนับสนุนการทำงาน ที่จำเป็นของ web application ด้วยเหตุนี้ J2EE specification จึงกำหนดให้มีการ implement Data Source ซึ่งใน tomcat นั้น default Data Source จะใช้ DBCP connection pool โดยมีขั้นตอนการติดตั้งดังนี้

1. ติดตั้ง JDBC driver

2. การประกาศ Resource requirement โดยประกาศไว้ในไฟล์ `web.xml` ของ web application นั้นๆ โดยประกาศไว้ใน `<resource-ref>` element ซึ่งมีรายละเอียดดังนี้

`<res-ref-name>` ชื่อ resource ที่ใช้ในการอ้างอิง

`<res-type>` ชนิดของ resource ในที่นี้คือ `javax.sql.DataSource`

`<res-auth>` การ authentication ในที่นี้เป็น Container ซึ่งหมายถึง Container เป็นผู้จัดการ authentication

3. การ config resource factory โดยจะกระทำใน `<context>` element ของ web application นั้นๆ ซึ่งอยู่ในไฟล์ `/conf/server.xml` โดย resource name ต้องตรงกันกับที่ระบุในไฟล์ `web.xml` ของ web application นั้น โดย property ที่สำคัญมีดังนี้

`driverClassName` เป็นชื่อของ java class ในรูปแบบเต็มของ JDBC driver

`url` เป็น url connection string ของ JDBC driver

`user` เป็น username ที่ใช้ในการ ติดต่อกับฐานข้อมูล

`password` เป็น password สำหรับการติดต่อกับฐานข้อมูล

`maxIdle` จำนวนของ idle connection มากที่สุดที่มีได้ใน connection pool

`maxWait` จำนวนของเวลาเป็น millisecond มากที่สุดที่ connection pool

จะรอก่อนเกิด Exception

`maxActive` จำนวนมากที่สุดของ active instance ที่อยู่ใน connection pool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ config ไฟล์ web application deployment descriptor

มีรายละเอียดในส่วนที่สำคัญดังนี้

<servlet> element จะประกอบไปด้วย element สำคัญๆ คือ

<servlet-name> เป็นชื่อของ servlet

<description> เป็น comment อธิบาย servlet

<servlet-mapping> ซึ่งประกอบไปด้วย element สำคัญคือ

<servlet-name> ชื่อ servlet

<url-pattern> เป็น url ที่ใช้เรียก servlet

<session-config> ประกอบไปด้วย element สำคัญดังนี้

<session-timeout> กำหนดค่า timeout ของ servlet session

และส่วนหัวของไฟล์จะต้อง เป็นไปตามข้อกำหนดคือ

```
<!DOCTYPE web-app
```

```
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN
```

```
"http://java.sun.com/dtd/web-app_2_3.dtd">
```

การ config tomcat 5.0.28

tomcat server มีโครงสร้างแฟ้มที่สำคัญๆ ที่เกี่ยวข้องในการ deploy web application ดังนี้

1. /common/lib เป็นแฟ้มที่ใช้เก็บไฟล์ library .jar ต่างๆ ที่ใช้ร่วมกันทุก web application ซึ่งรวมถึง JDBC driver ด้วย ให้เรา copy ไฟล์ ojdbc14.jar ซึ่งเป็น oracle JDBC driver ลงในแฟ้มนี้

2. /conf/catalina/localhost/ เป็นแฟ้มที่ใช้เก็บไฟล์ .xml ของแต่ละ web application ที่จะ deploy บน tomcat

โดยในโครงงานนี้จะชื่อไฟล์ middleware.xml มีรายละเอียดดังนี้

```
<Context path="/middleware" docBase="middleware">
  <Resource name="jdbc/middleware" auth="Container"
    scope="Shareable" type="javax.sql.DataSource"/>
  <ResourceParams name="jdbc/middleware">
    <parameter>
      <name>factory</name>
      <value>org.apache.commons.dbcp.BasicDataSourceFactory
    </value>
    </parameter>

    <!-- DBCP database connection settings -->

    <parameter>

      <name>url</name>

      <value>jdbc:oracle:thin:@localhost:1521:kmitldb</value>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</parameter>
```

```
<parameter>
```

```
<name>driverClassName</name>
```

```
<value>oracle.jdbc.OracleDriver</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>username</name>
```

```
<value>kmitluser</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>password</name>
```

```
<value>kmitluser</value>
```

```
</parameter>
```

```
<!-- DBCP connection pooling options -->
```

```
<parameter>
```

```
<name>maxIdle</name>
```

```
<value>100</value>
```

```
</parameter>
```

```
<parameter>
```

```
<name>maxActive</name>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<value>10</value>
```

```
</parameter>
```

```
</ResourceParams>
```

```
</Context>
```

3. /webapps/middleware/WEB-INF/ เป็นแฟ้มที่ใช้เก็บไฟล์ web application deployment descriptor ซึ่งในโครงงานนี้มีรายละเอียดดังนี้

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE web-app
```

```
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
```

```
"http://java.sun.com/dtd/web-app\_2\_3.dtd">
```

```
<web-app>
```

```
<display-name>
```

```
Middleware Web Application
```

```
</display-name>
```

```
<description>
```

```
Middleware web application
```

```
</description>
```

```
<resource-ref>
```

```
<description>
```

```
DB Connection
```

```
</description>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<res-ref-name>
```

```
jdbc/middleware
```

```
</res-ref-name>
```

```
<res-type>
```

```
javax.sql.DataSource
```

```
</res-type>
```

```
<res-auth>
```

```
Container
```

```
</res-auth>
```

```
</resource-ref>
```

```
<servlet>
```

```
<servlet-name>
```

```
Middleware
```

```
</servlet-name>
```

```
<description>
```

```
Middleware servlet
```

```
</description>
```

```
<servlet-class>
```

```
Middleware
```

```
</servlet-class>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<load-on-startup>5</load-on-startup>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
<servlet-name>
```

```
Middleware
```

```
</servlet-name>
```

```
<url-pattern>
```

```
/middleware
```

```
</url-pattern>
```

```
</servlet-mapping>
```

```
<session-config>
```

```
<session-timeout>30</session-timeout>
```

```
</session-config>
```

```
</web-app>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ setup ฐานข้อมูลและ initialize ค่าต่างๆของ Oracle9i

หลังจากติดตั้ง Oracle9i แล้วจะต้องสร้าง tablespace , สร้าง user และ กำหนดสิทธิ์ต่างๆ
ที่จำเป็นในการทำงาน ดังนี้

Create Tablespace project_ts

Datafile 'c:\oracle\oradata\kmitldb\project_1.dbf' size 200M

Minimum Extent 1000k

Default Storage

(Initial 1000k

Next 1000k

Maxextents 1000

Pctincrease 10;

Create User kmitluser

Identified by kmitluser

Default Tablespace project_ts

Quota unlimited on project_ts;

Grant connect to kmitluser;

Grant resource to kmitluser;

Grant sysdba to kmitluser;

Grant create session to kmitluser;

Grant create table to kmitluser;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้