

การพัฒนาเฟรมเวิร์กระบบการจำหน่ายสินค้าบนเว็บเซอร์วิส

FRAMEWORK SALE SYSTEM BASED ON WEB SERVICE



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....

51776

เลขทะเบียน.....

ปีการศึกษา 2546

วันเดือนปี 29 ก.ค. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....
i.....

FRAMEWORK SALE SYSTEM BASED ON WEB SERVICE



A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIRMENT 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 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การพัฒนาเฟรมเวิร์กระบบการจำหน่ายสินค้าบนเว็บเซอร์วิส
 FRAMEWORK FOR SALE SYSTEM BASED ON WEB SERVICE

ชื่อนักศึกษา นายเกริกเกียรติ วรวิสุทธิกุล 43050359
 นางสาววงศ์ทิพา วิเศษเกษม 43050406
 นายสิทธิกร คุณวิโรจน์พานิช 43050426

ปริญญา วิทยาศาสตรบัณฑิต
 ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
 สาขาวิชา วิทยาการคอมพิวเตอร์
 ปีการศึกษา 2546
 อาจารย์ที่ปรึกษา อ.วิสันต์ ตั้งวงษ์เจริญ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้รับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2546

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	รศ.ไพโรบลย์ พันธรัักษ์พงษ์	
กรรมการ	อ.รัชต์ธร อาชีวะ	
กรรมการและอาจารย์ที่ปรึกษา	อ.วิสันต์ ตั้งวงษ์เจริญ	



(ผู้ช่วยศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การพัฒนาเฟรมเวิร์กระบบการจำหน่ายสินค้าบนเว็บเซอร์วิส	
	FRAMEWORK SALE SYSTEM BASED ON WEB SERVICE	
ชื่อนักศึกษา	นายเกริกเกียรติ วรวิสุทธิกุล	43050359
	นางสาววงศัทธิพา วิเศษเกษม	43050406
	นายสิทธิกร คุณวิโรจน์พานิช	43050426
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2546	
อาจารย์ที่ปรึกษา	อ.วิสันต์ ตั้งวงษ์เจริญ	

บทคัดย่อ

ปัญหาพิเศษนี้เป็นการสร้าง Framework ระบบการจำหน่ายสินค้า เพื่อนำไปใช้กับเทคโนโลยี Web Service โดยได้ทำการเขียน WSDL ของตนเองและนำไปฝากไว้ยัง UDDI ที่อนุญาตเพื่อเป็นการประกาศบริการของตนเองให้กับผู้ที่สนใจ ในส่วนของ Framework ระบบการจำหน่ายสินค้านี้ได้แบ่งออกเป็น Component ย่อยจำนวน 5 Component ได้แก่ Component ผู้ดูแลระบบ, Component การอนุญาตให้เข้าใช้ระบบ, Component ฝ่ายขาย, Component ฝ่ายบัญชี และ Component ฝ่ายจัดการสินค้า ซึ่ง Component เหล่านี้จะอยู่ผู้ให้บริการเมื่อผู้ขอใช้บริการต้องการขอใช้บริการก็สามารถติดต่อไปยังผู้ให้บริการ และเมื่อได้รับอนุญาตให้ใช้บริการได้แล้ว ผู้ขอใช้บริการจะสามารถทำการเรียกใช้เรียกใช้ผ่านระบบอินเทอร์เน็ต โดยใช้ Protocol SOAP ในการส่งและรับพารามิเตอร์ ระหว่างทั้ง 2 ฝ่าย การใช้บริการ Web Service จะช่วยอำนวยความสะดวกให้กับผู้ขอใช้บริการ และผู้ให้บริการอย่างมากเนื่องจาก ผู้ขอใช้บริการจะสามารถดึง Component ที่มีอยู่แล้วของผู้ให้บริการมาใช้ได้ทันทีโดยที่ไม่จำเป็นต้องเขียนโปรแกรมเอง โดยอาจจะนำไปใช้ทั้งระบบ หรือ ดึงไปใช้เพียงบางส่วนร่วมกับระบบเดิมที่ผู้ขอใช้บริการมีอยู่ก็ได้ อีกทั้งทางฝั่งผู้ให้บริการสามารถนำ Component ที่ตนให้บริการไปบริการผู้ขอใช้บริการได้หลายราย ซึ่งเป็นประโยชน์กับทั้ง 2 ฝ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	FRAMEWORK FOR SALE SYSTEM BASED ON WEB SERVICE	
Student	Mr.Kroekkiat Voravisuttikul	43050359
	MissWongtipa Wisetkasem	43050406
	Mr.Sittikorn Khunvirojpanich	43050426
Degree	Bachelor of Science	
Department	Mathematics and Computer Science, Faculty of Science	
Programme	Computer Science	
Academic Year	2003	
Special Project Advisor	Wisai Tangwongcharoen	

ABSTRACT

This special project proposes the way to develop Sale System Framework to merge with web service technology. Implementing WSDL bring to embed to permitted UDDI for interested people. There are five components in this framework which have administrator component , Authentication component , Sale component , Accountant component and Stock component . They are all in provider side. When users would like to request for services, they can connect to provider side and receive the permission , then users can call for services from internet. SOAP protocol is use to send and receive parameters between two sides. Web service make both users and providers comfortable because users can call for provider's existing whole system or some components to use immediately. In addition the providers can provide their component to many users. It is very useful for both providers and users.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องเฟรมเวิร์กระบบการจำหน่ายสินค้าบนเว็บไซต์ สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบคุณ อาจารย์วิสันต์ ตั้งวงษ์เจริญ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ ที่กรุณาให้คำแนะนำ และเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้การสนับสนุนในด้านกำลังใจ รวมถึงเพื่อนๆ พี่ๆ น้องๆ ทุกคนที่ได้ให้ความช่วยเหลือ จนทำให้ปัญหาพิเศษนี้เสร็จสมบูรณ์ได้

คณะผู้จัดทำ
มีนาคม 2546



สารบัญ

หน้า

บทคัดย่อปัญหาพิเศษภาษาไทย.....	I
บทคัดย่อปัญหาพิเศษภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์ของการทำปัญหาพิเศษ.....	1
1.3 ขอบเขตของปัญหาพิเศษ.....	1
1.4 ขั้นตอนในการดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 อุปกรณ์ที่ใช้ทำปัญหาพิเศษ.....	2
บทที่ 2 ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง.....	4
2.1 ความหมายของ Web Service.....	4
2.1.1 การพัฒนาเทคโนโลยี World Wide Web.....	4
2.1.2 แนวคิดพื้นฐาน และ โครงสร้างของ Web Service.....	6
2.1.3 เทคโนโลยีที่เกี่ยวข้องกับ Web Service.....	7
2.1.3.1 องค์ประกอบของ XML (The Extensible Markup Language).....	7
2.1.3.2 องค์ประกอบของ SOAP (Simple Object Access Protocol).....	8
2.1.3.3 องค์ประกอบของ WSDL (Web Service Description Language).....	11
2.1.3.4 องค์ประกอบของ Universal Description Discovery and Integration (UDDI).....	12
2.1.4 การพัฒนาเวปเซอร์วิสของบริษัท Microsoft และ Sun One.....	12
2.1.4.1 การพัฒนาเทคโนโลยี Web Service ของ Microsoft. Net.....	12
2.1.4.2 การพัฒนาเทคโนโลยี Web Service ของ Sun One.....	13
2.1.5 การเปรียบเทียบความแตกต่างระหว่างการพัฒนาของ ทั้ง สองบริษัท.....	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.2 การใช้งาน Servlets.....	14
2.2.1 ความหมาย และหน้าที่ของ Servlets.....	14
2.2.2 ข้อดีของ Servlets.....	16
2.2.2.1 Servlets มีการทำงานที่เร็วกว่า CGI Scripts เนื่องจากขั้นตอนการทำงานที่แตกต่างกัน.....	16
2.2.2.2 Servlets ใช้ Standard API ที่ได้รับการสนับสนุนจากหลายๆ Web Servers..	16
2.2.2.3 Servlets สามารถใช้ประโยชน์จาก Java Programming Languages ในเรื่องของความสะดวก ในการพัฒนา และความเป็นอิสระจาก Platform ใด ๆ.....	17
2.2.3 การติดตั้งเครื่องมือสำหรับการพัฒนา.....	17
2.2.3.1 การติดตั้ง Server.....	17
2.2.4 ความรู้พื้นฐานเกี่ยวกับ Java Servlet API.....	18
2.2.5 โครงสร้างของ Servlets.....	18
2.3 ความรู้เบื้องต้นเกี่ยวกับ RMI.....	21
2.3.1 ความหมายของ RMI.....	21
2.3.2 ที่มา ของ Java RMI.....	21
2.3.2.1 Protocol ของ Java RMI.....	21
2.3.2.2 ปัญหาของ RMI.....	23
2.4 Domain ของระบบการจำหน่ายสินค้า.....	25
บทที่ 3 การออกแบบระบบการจำหน่ายสินค้า.....	27
3.1 Use Case Diagram ของระบบการจำหน่ายสินค้า.....	27
3.1.1 พฤติกรรมหลักของระบบการจำหน่ายสินค้า.....	27
3.1.2 พฤติกรรมย่อยที่เกี่ยวข้องกับพฤติกรรมหลัก.....	28
3.1.2.1 การสั่งซื้อสินค้า.....	28
3.1.2.2 การออกใบ Invoice.....	29
3.1.2.3 การจัดการสินค้า.....	29
3.1.2.4 การชำระเงิน.....	30
3.2 การออกแบบ Framework สำหรับระบบการจำหน่ายสินค้า.....	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.2.1 แผนภาพ Class ต่างๆในส่วนของผู้ใช้บริการ Web Service.....	32
3.2.1.1 รายละเอียดส่วนประกอบภายใน Class MarketingService	32
3.2.1.2 รายละเอียดส่วนประกอบภายใน Class StockService.....	33
3.2.1.3 รายละเอียดส่วนประกอบภายใน Class AccountService.....	34
3.2.2 แผนภาพ Class ต่างๆในส่วนของผู้ใช้บริการ Web Service.....	36
3.2.2.1 รายละเอียดส่วนประกอบภายใน Class MarketingManagement.....	37
3.2.2.2 รายละเอียดส่วนประกอบภายใน Class StockManagement.....	39
3.2.2.3 รายละเอียดส่วนประกอบภายใน Class AccountManagement.....	40
3.2.2.4 รายละเอียดส่วนประกอบภายใน Class DatabaseManagement.....	43
3.2.2.5 รายละเอียดส่วนประกอบภายใน Class CustomerManagement.....	44
3.2.2.6 รายละเอียดส่วนประกอบภายใน Class EmployeeManagement.....	45
3.2.2.7 รายละเอียดส่วนประกอบภายใน Class AdminManagement.....	46
3.2.2.8 รายละเอียดส่วนประกอบภายใน Class CheckLogin.....	48
3.2.2.9 รายละเอียดส่วนประกอบภายใน Class CheckUser.....	49
3.3 Sequence Diagram ของระบบการจำหน่ายสินค้า.....	50
3.3.1 Sequence Diagram ของผู้ใช้ที่เป็น Administrator.....	50
3.3.1.1 Sequence Diagram การตรวจสอบสิทธิ์การเข้าใช้ระบบของ Admin.....	50
3.3.1.2 Sequence Diagram การเพิ่มข้อมูลของพนักงานใหม่.....	51
3.3.1.3 Sequence Diagram การแก้ไขข้อมูลของพนักงาน.....	51
3.3.1.4 Sequence Diagram การลบข้อมูลพนักงาน.....	52
3.3.1.5 Sequence Diagram การเปลี่ยน Password เข้าระบบของ Admin.....	52
3.3.2 Sequence Diagram ของผู้ใช้ที่เป็นพนักงาน.....	53
3.3.2.1 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานทุกแผนก.....	53
3.3.2.2 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Marketing.....	54
3.3.2.3 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Stock.....	59
3.3.2.4 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Account.....	62
บทที่ 4 การทดสอบและการพัฒนาระบบ.....	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.1 รายละเอียดของการพัฒนาระบบ.....	65
4.1.1 ภาษาที่ใช้ในการพัฒนา.....	65
4.1.2 เครื่องมือที่ใช้ในการพัฒนา.....	65
4.2 การพัฒนาระบบให้สามารถเป็น Web Service.....	65
4.2.1 การพัฒนาระบบให้เป็น Web Service ฟังก์ชันเซอร์เวอร์.....	65
4.2.1.1 การเพิ่ม Package และ throw Exception.....	65
4.2.1.2 การเตรียม Tomcat.....	65
4.2.1.3 การจัดการเก็บ file soap.war.....	66
4.2.1.4 การเก็บ file ที่ compile แล้ว.....	67
4.2.1.5 การทำ Deployment Descriptor.....	67
4.2.2 การพัฒนาระบบให้เป็น Web Service ฟังก์ชันแอนตี้.....	70
4.3 การทดสอบการใช้งานโปรแกรม.....	72
4.3.1 การจัดการข้อมูลพนักงาน.....	72
4.3.2 การใช้งานเมื่อลูกค้าต้องการสั่งซื้อสินค้า.....	75
4.3.3 การใช้งานเมื่อต้องการออกไป Invoice และตัดจำนวนสินค้าในคลัง.....	82
4.3.4 การแสดงรายงาน ใบ Invoice ที่ใกล้ถึงเวลาชำระค่าสินค้า.....	85
บทที่ 5 สรุปผลการทำปัญหาพิเศษและข้อเสนอแนะ.....	89
5.1 สรุปผลการทำงานของ โปรแกรม.....	89
5.2 ปัญหาและข้อจำกัดของปัญหาพิเศษ.....	92
5.3 ข้อเสนอแนะ	92
บรรณานุกรม.....	93
ภาคผนวก ก คู่มือการติดตั้งโปรแกรม.....	94
ภาคผนวก ข คู่มือการใช้งาน.....	112

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดงหน้าที่องค์ประกอบหลักของเทคโนโลยี SOA.....	6
2.2 ตัวอย่าง Packet API ที่ JAVA ได้จัดเตรียมไว้เพื่อความสะดวกในการสร้างเว็บเซอร์วิส.....	13
3.1 Attribute ภายใน Class MarketingService.....	32
3.2 Method ภายใน Class MarketingService.....	33
3.3 Attribute ภายใน Class StockService.....	33
3.4 Method ภายใน Class StockService.....	34
3.5 Attribute ภายใน Class AccountService.....	35
3.6 Method ภายใน Class AccountService.....	35
3.7 Attribute ภายใน Class MarketingManagement.....	37
3.8 Method ภายใน Class MarketingManagement.....	38
3.9 Attribute ภายใน Class StockManagement.....	39
3.10 Method ภายใน Class StockManagement.....	39
3.11 Attribute ภายใน Class AccountManagement.....	41
3.12 Method ภายใน Class AccountManagement.....	41
3.13 Attribute ภายใน Class DatabaseManagement.....	43
3.14 Method ภายใน Class DatabaseManagement.....	43
3.15 Attribute ภายใน Class CustomerManagement.....	44
3.16 Method ภายใน Class CustomerManagement.....	44
3.17 Attribute ภายใน Class EmployeeManagement.....	45
3.18 Method ภายใน Class EmployeeManagement.....	45
3.19 Attribute ภายใน Class AdminManagement.....	46
3.20 Method ภายใน Class AdminManagement.....	47
3.21 Attribute ภายใน Class CheckLogin.....	48
3.22 Method ภายใน Class CheckLogin.....	48
3.23 Method ภายใน Class CheckUser.....	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การพัฒนาเว็บยุคที่ 1.....	4
2.2 การพัฒนาเว็บยุคที่ 2.....	5
2.3 ภาพแสดงแนวคิดของเทคโนโลยี SOA	6
2.4 แผนภาพแสดงโครงสร้างของ Web Service.....	7
2.5 แพลตฟอร์มของโปรโตคอล SOAP.....	9
2.6 การทำงานของโปรโตคอล SOAP.....	10
2.7 การทำงานของ SOAP Listener.....	11
2.8 แพลตฟอร์มของ .Net.....	12
2.9 การเปรียบเทียบการทำงานระหว่าง .Net กับ Java.....	14
2.10 รูปแบบการทำงานของ Servlets.....	15
2.11 แสดง Protocol ของ JAVA RMI.....	22
2.12 แผนภาพแสดงการส่งข้อมูล "sayHello" ผ่านทาง RMI.....	24
3.1 แสดง Use Case Diagram หลักของระบบการจำหน่ายสินค้า.....	28
3.2 แสดง Use Case Diagram การสั่งซื้อสินค้า.....	29
3.3 แสดง Use Case Diagram การออกไป Invoice.....	29
3.4 แสดง Use Case Diagram การจัดการสินค้า.....	30
3.5 แสดง Use Case Diagram การชำระเงิน.....	30
3.6 แผนภาพการแสดง Framework ของระบบการจำหน่ายสินค้า.....	31
3.7 Diagram แสดง Class ในส่วนของผู้ให้บริการ.....	32
3.8 ภาพแสดงรายละเอียดภายใน Class MarketingService.....	32
3.9 ภาพแสดงรายละเอียดภายใน Class StockService.....	33
3.10 ภาพแสดงรายละเอียดภายใน Class AccountService	34
3.11 Diagram แสดง Class ในส่วนของผู้ขอใช้บริการ.....	36
3.12 ภาพแสดงรายละเอียดภายใน Class MarketingManagement.....	37
3.13 ภาพแสดงรายละเอียดภายใน Class StockManagement.....	39
3.14 ภาพแสดงรายละเอียดภายใน Class AccountManagement.....	40
3.15 ภาพแสดงรายละเอียดภายใน Class DatabaseManagement.....	43
3.16 ภาพแสดงรายละเอียดภายใน Class CustomerManagement.....	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.17 ภาพแสดงรายละเอียดภายใน Class EmployeeManagement.....	45
3.18 ภาพแสดงรายละเอียดภายใน Class AdminManagement.....	46
3.19 ภาพแสดงรายละเอียดภายใน Class CheckLogin.....	48
3.20 ภาพแสดงรายละเอียดภายใน Class CheckUser.....	49
3.21 แสดง Sequence Diagram การตรวจสอบสิทธิ์การเข้าใช้ระบบของ Admin.....	50
3.22 แสดง Sequence Diagram การเพิ่มข้อมูลของพนักงานใหม่.....	51
3.23 แสดง Sequence Diagram การแก้ไขข้อมูลของพนักงาน.....	51
3.24 แสดง Sequence Diagram การลบข้อมูลพนักงาน.....	52
3.25 แสดง Sequence Diagram การเปลี่ยน Password เข้าสู่ระบบของ Admin.....	52
3.26 แสดง Sequence Diagram การตรวจสอบสิทธิ์การเข้าใช้ระบบของพนักงาน.....	53
3.27 แสดง Sequence Diagram การเปลี่ยน Password ของพนักงาน.....	54
3.28 แสดง Sequence Diagram การเพิ่มข้อมูลของลูกค้าใหม่.....	55
3.29 แสดง Sequence Diagram การเปลี่ยนแปลงข้อมูลของลูกค้า.....	55
3.30 แสดง Sequence Diagram การลบข้อมูลของลูกค้า.....	56
3.31 แสดง Sequence Diagram การสร้างใบสั่งสินค้า.....	57
3.32 แสดง Sequence Diagram การยกเลิกใบสั่งสินค้า.....	58
3.33 แสดง Sequence Diagram การเพิ่มจำนวนสินค้าในคลัง.....	59
3.34 แสดง Sequence Diagram การเพิ่มรายการสินค้าใหม่.....	60
3.35 แสดง Sequence Diagram การเปลี่ยนแปลงราคาสินค้า.....	61
3.36 แสดง Sequence Diagram การออกใบ Invoice.....	62
3.37 แสดง Sequence Diagram การเปลี่ยนสถานะของใบ Invoice.....	63
3.38 แสดง Sequence Diagram การออกใบวางบิล.....	64
4.1 หน้าจอหน้าจอแสดง package ที่ใส่ลงใน folder lib.....	66
4.2 หน้าจอแสดงไฟล์ soap.war.....	66
4.3 หน้าจอเก็บ file ที่ compile เรียบร้อยแล้ว.....	67
4.4 เอกสาร XML เพื่อประกาศบริการให้โปรโตคอล SOAP รู้จัก.....	68
4.5 หน้าจอแสดงการ Deploy บริการ.....	68
4.6 หน้าจอแสดงผลว่ามีบริการใดให้บริการอยู่บ้าง.....	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.7 หน้าจอแสดงการ Undeploy บริการ.....	69
4.8 หน้าจอแสดงผลว่ามีบริการใดให้บริการอยู่บ้างหลังการ Undeploy.....	69
4.9 หน้าจอ login ของผู้ดูแลระบบ.....	72
4.10 หน้าจอแสดงรายการกิจกรรมหลักในฝ่ายดูแลระบบ.....	73
4.11 หน้าจอแสดงรายชื่อพนักงานทั้งหมดในระบบ.....	73
4.12 หน้าจอกรอกข้อมูลพนักงานใหม่.....	74
4.13 หน้าจอแสดงรหัสประจำตัวพนักงาน และ password เริ่มต้น.....	75
4.14 หน้าจอหลักของฝ่ายขาย.....	76
4.15 หน้าจอแสดงรายชื่อลูกค้าทั้งหมด.....	76
4.16 หน้าจอกรอกข้อมูลลูกค้าใหม่.....	77
4.17 หน้าจอแสดงการบันทึกข้อมูลลูกค้าใหม่.....	78
4.18 หน้าจอเลือกรหัสลูกค้าเพื่อทำการสั่งซื้อสินค้า.....	78
4.19 หน้าจอแสดงรายละเอียดของพนักงานและลูกค้าที่ต้องการสั่งซื้อสินค้า.....	79
4.20 หน้าจอแสดงรายการสินค้าและจำนวนที่ลูกค้าต้องการสั่งซื้อ.....	80
4.21 หน้าจอแสดงให้ทราบว่าสินค้าในคลังเพียงพอหรือไม่.....	81
4.22 หน้าจอแสดงการบันทึกข้อมูลลงฐานข้อมูล.....	81
4.23 หน้าจอกิจกรรมหลักของฝ่ายบัญชี.....	82
4.24 หน้าจอแสดงใบสั่งซื้อสินค้าเพื่อการออกไป Invoice.....	83
4.25 หน้าจอแสดงรายงานใบ Invoice จากใบสั่งซื้อสินค้า.....	84
4.26 หน้าจอแสดงรายงานใบ Invoice จากใบสั่งซื้อสินค้า (ต่อ).....	84
4.27 หน้าจอแสดงการสร้างใบ Invoice สำเร็จแล้ว.....	85
4.28 หน้าจอแสดงกิจกรรมหลักของฝ่ายบัญชี.....	86
4.29 หน้าจอแสดงรหัสประจำตัว และรายชื่อลูกค้าเพื่อตรวจสอบการชำระเงิน.....	86
4.30 หน้าจอแสดงรายการใบ Invoice ที่เตรียมชำระค่าสินค้า.....	87
4.31 หน้าจอแสดงรายละเอียดของใบ Invoice ที่เลือก.....	88

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

แนวทางการทำงานของโปรแกรมระบบการจำหน่ายในปัจจุบัน มีองค์ประกอบส่วนใหญ่ที่คล้ายคลึงกัน จึงควรที่จะใช้ประโยชน์จากส่วนนี้มาทำเป็น Framework ซึ่งเป็นโครงร่างในการให้กำเนิด Application โดยผู้ออกแบบ Application จะเป็นผู้กำหนดรายละเอียดหรือเงื่อนไขเพียงเล็กน้อยเท่านั้น และ Application ที่สร้างขึ้นจะไม่ขึ้นอยู่กับ Platform จึงสามารถทำงานประสานกันระหว่าง Platform ของ JAVA, Windows, Browser และ PDA/WAP เพื่อที่ผู้พัฒนา Application การจำหน่ายสินค้า จะได้มีความสะดวกในการสร้าง Application โดยใช้เทคโนโลยี Web Service เข้ามาประยุกต์ให้สามารถทำการเลือก Component ที่ต้องการซึ่งอยู่ใน Framework ที่เราได้ทำการสร้างและ/หรือ Framework ที่มีอยู่ก่อนแล้วได้ จากปัญหา และเทคโนโลยีข้างต้น เป็นที่มาของการทำปัญหาพิเศษ

1.2 วัตถุประสงค์ของการทำปัญหาพิเศษ

ในการจัดทำปัญหาพิเศษนี้มีความมุ่งหมายและวัตถุประสงค์ดังต่อไปนี้

- 1) เพื่อสร้าง Framework ระบบการจำหน่ายสินค้า
- 2) เพื่อให้ Framework นี้เป็นตัวต้นแบบในการพัฒนา Application โดยใช้เทคโนโลยี Web Service
- 3) เพื่อแนวทางในการนำเทคโนโลยี Web Service ไปประยุกต์ใช้ร่วมกับระบบงานอื่น โดยเฉพาะกับระบบงานที่เกี่ยวข้องกับการดำเนินการทางธุรกิจได้
- 4) เพื่อให้ผู้ใช้บริการหลายรายสามารถเรียกใช้ Framework ระบบการจำหน่ายสินค้านี้ได้พร้อมกัน และใช้งานได้เหมือนกัน

1.3 ขอบเขตของปัญหาพิเศษ

โครงการพิเศษนี้เป็นการสร้าง Framework ระบบการจำหน่ายสินค้าโดย นำเทคโนโลยี Web Service เข้ามาเป็นตัวช่วย เพื่อให้สามารถนำไปพัฒนา หรือนำไปใช้ได้อย่างสะดวก รวมถึงสามารถนำไปใช้ร่วมกับระบบอื่นได้โดยง่าย

- 1) สร้าง Framework ระบบการขาย
- 2) สร้าง Component ของ Framework การขาย ดังนี้

ผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การอนุญาตให้เข้าใช้ระบบ
 - การจัดการข้อมูลพนักงาน
 - ฝ่ายจัดการสินค้า
 - ฝ่ายขาย
 - ฝ่ายบัญชี
- 3) สามารถเรียกใช้ Framework การซื้อขายผ่านเทคโนโลยี Web Service ได้

1.4 ขั้นตอนในการดำเนินงาน

ในการทำปัญหาพิเศษนี้มีขั้นตอนในการดำเนินงานดังต่อไปนี้

- 1) ศึกษาเทคโนโลยี Web Service, Framework และระบบการจำหน่ายสินค้า
- 2) ออกแบบ Framework สำหรับระบบการซื้อขาย
- 3) ศึกษา และเรียนรู้ Program ที่ช่วยในการสร้าง Framework และเทคโนโลยี Web Service
- 4) เขียนโปรแกรม และทดสอบการทำงาน
- 5) ศึกษาปัญหาที่เกิดขึ้น และแนวทางการแก้ไข พร้อมทั้งจะแก้ไข Program
- 6) รวบรวมข้อมูลต่างๆ และทำรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการทำปัญหาพิเศษนี้ สามารถแบ่งออกได้ดังต่อไปนี้

- 1) มี Framework ระบบการซื้อขายในการใช้งาน
- 2) เป็น Component ตัวยึดแบบในการพัฒนาต่อไป
- 3) ทำให้เกิดความสะดวกในการใช้งาน
- 4) สามารถนำ Component กลับมาใช้งานใหม่ได้โดย สามารถนำไปใช้ร่วมกับระบบอื่นๆ ได้

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

สามารถทำการแบ่งอุปกรณ์ที่ใช้ในการทำปัญหาพิเศษนี้ได้เป็น 2 ส่วน คือ

1. เครื่องเซิร์ฟเวอร์

ประกอบด้วย

- ด้านฮาร์ดแวร์ ได้แก่
 - Pentium III 800 MHz
 - Ram 256 MB
 - Harddisk 5 GB
 - ระบบ High Speed Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ด้านซอฟต์แวร์ ได้แก่
 - Window 2000 Server, Window XP Professional
 - Java2SDK version 1.4.x
 - Java Web Service Developer Pack version 1.2 ขึ้นไป
 - MySQL

2. Client

ประกอบด้วย

- ด้านฮาร์ดแวร์ ได้แก่
 - Pentium II 400 MHz
 - Ram 64 MB
 - Harddisk 1.5 GB
 - ระบบ Internet 56 Kbps
- ด้านซอฟต์แวร์ ได้แก่
 - Web Browser



บทที่ 2

ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง

2.1 ความหมายของ Web Service

2.1.1 การพัฒนาเทคโนโลยี World Wide Web

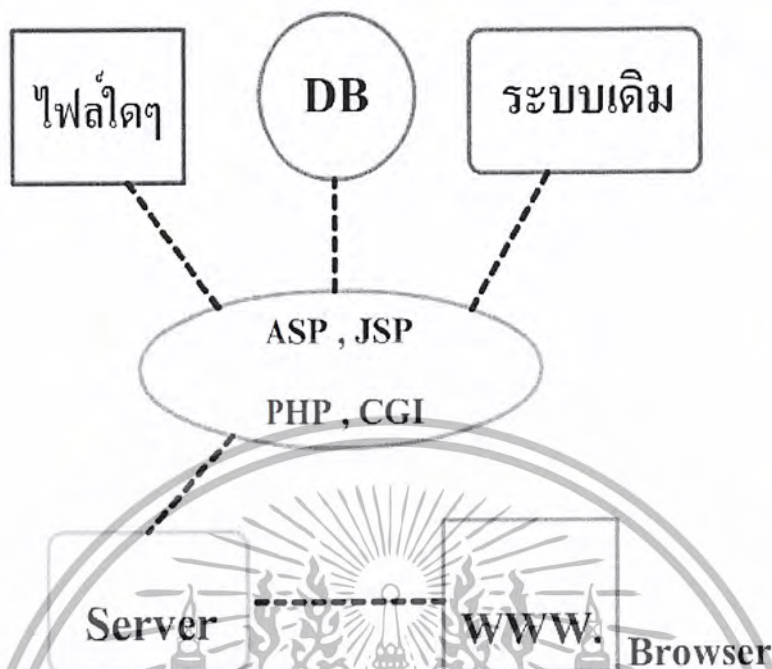
เทคโนโลยีเว็บ เป็นที่แพร่หลายอย่างมากในปัจจุบันเป็น จะสามารถแบ่ง ยุคของการพัฒนา ออกได้เป็น 3 ยุคดังนี้
ยุคที่ 1



รูปที่ 2.1 การพัฒนาเว็บยุคที่ 1

เป็นยุคที่เราเรียกว่า Static เนื่องจากข้อมูลที่แสดงในเว็บมีลักษณะตายตัว โดยการท่องเว็บไซด์ หนึ่งๆ จะใช้เบราว์เซอร์ เพื่อเรียก เว็บเพจ ที่สร้างด้วยภาษา HTML อย่างเดียว หรืออาจมีการใช้สคริปต์ ทางเครื่องด้านไคลแอนท์ (Client – Side Script) ร่วมด้วย อย่างเช่น ภาษา JavaScript , ภาษา VBScript เป็นต้น ซึ่งโปรแกรมเหล่านี้เป็น โปรแกรมที่ทำงานในฝั่ง browser ทำให้ file เหล่านั้นไม่สามารถติดต่อกับองค์ประกอบอื่นๆ ทางฝั่งเซิร์ฟเวอร์ได้ เช่น การติดต่อฐานข้อมูล เป็นต้น

ยุคที่ 2



รูปที่ 2.2 การพัฒนาเว็บยุคที่ 2

ยุคที่ 2 นี้เป็นยุคของ Dynamic Web ที่พัฒนามาจากยุคแรก โดยมีการใช้สคริปต์ทางฝั่งเซิร์ฟเวอร์ (server-side script) มาช่วยเพิ่มความสามารถของเอกสาร HTML ในการติดต่อกับองค์ประกอบอื่นๆ ทางเครื่องด้านเซิร์ฟเวอร์ เช่น การติดต่อกับฐานข้อมูล การสร้างห้องสนทนา (chat room) กระดานถามตอบ (webboard) เป็นต้น เทคโนโลยีสำหรับ โดนามิกเว็บ ในช่วงแรกซึ่งเป็นที่รู้จักกันมากก็คือ CGI (Common Gateway Interface) เป็นโปรแกรมที่ทำงานอยู่ทางเครื่องด้านเซิร์ฟเวอร์ เมื่อผู้ใช้เรียกใช้ CGI จากนั้น CGI ก็จะทำงานตามหน้าที่ ที่ถูกเขียนสคริปต์ขึ้นมา ซึ่งและจะส่งผลลัพธ์ตอบกลับไปยังผู้ใช้หรือไม่ก็ได้ ขึ้นอยู่กับว่ามีการระบุหน้าที่นี้ไว้ในสคริปต์ CGI หรือไม่ หลังจากนั้นได้มีการนำเทคโนโลยีหลายอย่าง ที่มีหลักการคล้าย CGI เพื่อทำงานทางฝั่งเว็บเซิร์ฟเวอร์ เช่น ASP (Active Server Pages) ของไมโครซอฟต์, JSP (JavaServer Pages) ที่ใช้ภาษา Java ในการพัฒนารวมถึง PHP (Professional Home Page) ด้วย

ยุคที่ 3

ก็คือยุคของ Web Service หรือยุคที่ซอฟต์แวร์คือบริการที่มีอยู่ในเว็บ โดยแนวคิดของ Web Service ก็คือ เว็บ ที่สามารถทำงานอะไรบางอย่างหรือ ก็คือ ให้บริการบางอย่างจากการร้องขอ จากต่าง เซิร์ฟเวอร์ ด้วยเหตุนี้ทำให้เทคโนโลยี Web Service จึงเอื้อต่อแนวคิด Distributed Processing

มากกว่า เว็บแอปพลิเคชัน และเมื่อประกอบกับการที่ Web Service มี UDDI ทำให้ Web Service เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถค้นหาบริการต่างๆ ที่ต้องการได้จากทั่วทุกมุมโลก โดยเทคโนโลยีต่างๆ ที่เกี่ยวข้องกับ Web Service จะได้กล่าวถึงในหัวข้อถัดไป

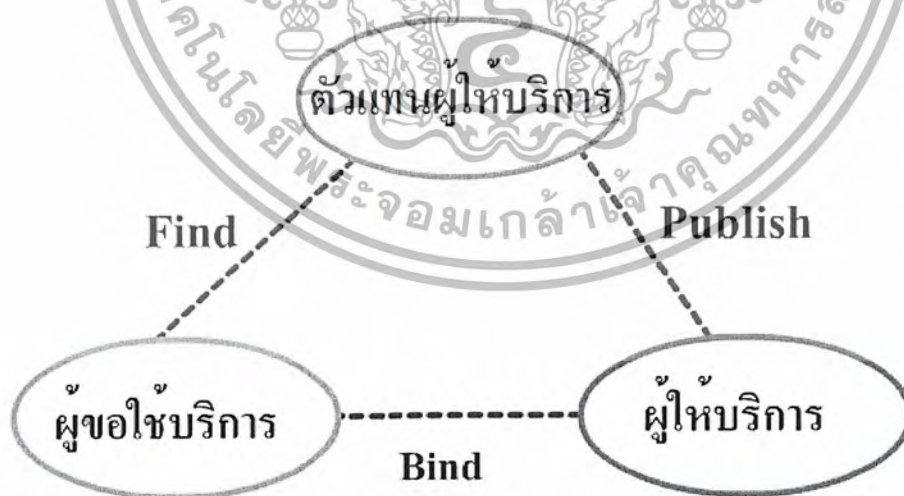
2.1.2 แนวคิดพื้นฐาน และโครงสร้างของ Web Service

Web Service มีแนวคิดมาจากโครงสร้างของ เทคโนโลยี SOA (Service Oriented Architecture) โดยจะประกอบไปด้วย 3 ส่วนหลัก ในการติดต่อถึงกันคือ

1. ตัวแทนผู้ให้บริการ (Registry)
2. ผู้ให้บริการ (Provider)
3. ผู้ขอใช้บริการ (Requestor)

ตารางที่ 2.1 ตารางแสดงหน้าที่ขององค์ประกอบหลักของเทคโนโลยี SOA

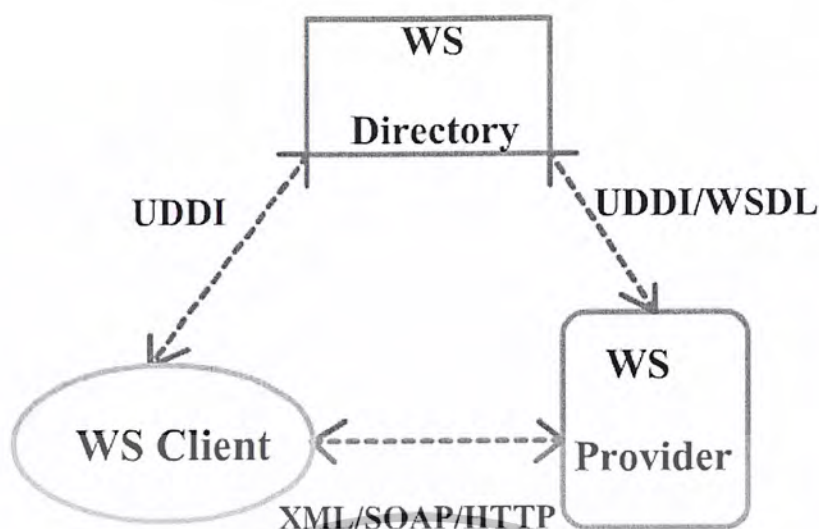
ผู้ขอใช้บริการ	เป็นใครก็ตามที่ต้องการเรียกใช้บริการจาก ผู้ให้บริการ ซึ่งสามารถค้นหาบริการที่ต้องการได้จาก UDDI ที่อยู่ตัวแทนผู้ให้บริการ
ตัวแทนผู้ให้บริการ	ทำหน้าที่เป็นตัวกลางให้ ผู้ให้บริการ มาลงทะเบียนไว้ โดยใช้ WSDL ไฟล์ บอกรายละเอียดของบริษัทและบริการที่มีให้ ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้
ผู้ให้บริการ	เป็นผู้ให้บริการ มีหน้าที่ในการเปิดบริการเพื่อรองรับการขอใช้บริการจาก ผู้ขอใช้บริการ ที่เรียกเข้ามาขอใช้



รูปที่ 2.3 ภาพแสดงแนวคิดของเทคโนโลยี SOA

และจากแนวคิดนี้ เมื่อนำมาผนวกเข้ากับเทคโนโลยีที่จะกล่าวถึงในหัวข้อถัดไป ทำให้เกิดโครงสร้างของ Web Service ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แผนภาพแสดงโครงสร้างของ Web Service

2.1.3 เทคโนโลยีที่เกี่ยวข้องกับ Web Service

เทคโนโลยีสำคัญ ๆ ที่เกี่ยวข้องกับ Web Service ประกอบด้วย

2.1.3.1 องค์ประกอบของ XML (The Extensible Markup Language)

XML (The Extensible Markup Language) เป็นภาษา Markup ที่เป็น text-based ซึ่งทำให้เป็น มาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตอย่างรวดเร็ว ผู้ที่ทำหน้าที่รับผิดชอบ และ กำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) ความแตกต่างระหว่าง XML กับ HTML คือ HTML ถูกนำมาใช้ในการสร้าง เว็บไซต์ ที่สามารถแสดงผลได้โดยโปรแกรม บราวเซอร์ แต่ XML จะสามารถใส่ tags ได้อย่างอิสระ จากนั้นจะทำการส่ง XML ชุดนี้ไปประมวล ผลยังแอปพลิเคชันใด ๆ ที่สามารถใช้ข้อมูลใน XML นี้

XML เป็นภาษาที่มีลักษณะเป็น tag คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อ ความหมายโดยอนุญาตให้ผู้ใช้สามารถกำหนด tag ขึ้นได้เองเพื่อให้สื่อความหมายทางภาษาของ มนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกันทำให้ข้อมูลระหว่างtag สามารถนำไปประมวลผลต่อได้ เช่น

```

<ComputerBook>
  <book>
    <name>Web Service</name>
    <price>10.00$</price>
  </book>
  <book>
    <name>xml</name>
    <price>10.00$</price>
  </book>
</ComputerBook>

```

จากตัวอย่าง จะเห็นได้ว่ารูปแบบเอกสาร XML ไม่ได้บอกวิธีแสดงผลไว้ แต่เอกสารสามารถสื่อความหมายให้คอมพิวเตอร์เข้าใจได้ และนำค่าไปประมวลผลต่อได้ แต่ในกรณีที่ต้องการจะแสดงผลจำเป็นจะต้องใช้ควบคู่กับ HTML

2.1.3.2 องค์ประกอบของ SOAP (Simple Object Access Protocol)

เนื่องจากจุดประสงค์หลักของการใช้งาน Web Service คือ ต้องการให้แอปพลิเคชัน มีการทำงานกับแอปพลิเคชันที่ทำงานอยู่ในเครื่องอื่น โดยผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่าง Object ในระยะไกล (Remote Procedure Calls : RPC) เช่น DCOM , EJB หรือ CORBA นั้นไม่ได้ถูกออกแบบมาใช้สำหรับโปรโตคอล HTTP (เครือข่ายที่ปัจจุบันใช้งานแพร่หลาย และสะดวกมากที่สุดได้แก่เครือข่าย อินเทอร์เน็ต ซึ่งให้การสื่อสารผ่านโปรโตคอล HTTP เป็นส่วนใหญ่)

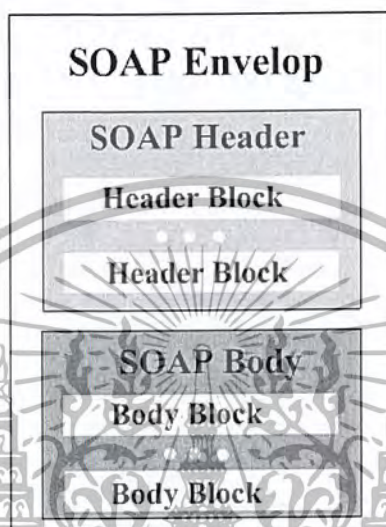
เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้นต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย ยกเว้น CORBA ผู้พัฒนาระบบจะต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของ Firewalls และ Proxy Server เนื่องจากโดยปกติ เซิร์ฟเวอร์จะปิดการสื่อสารที่ไม่ใช่โปรโตคอล HTTP ออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก

ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำบริการ Web Service ก็คือให้ทำงานอยู่บน โปรโตคอล HTTP ซึ่ง SOAP นอกจากจะทำงานบนโปรโตคอล HTTP แล้วยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ, เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนๆของเอกสารได้ เป็น 3 ส่วนหลักดังนี้คือ

1. SOAP Envelop : เนื้อหาสาระ (Content) ของเอกสารทั้งหมด
2. SOAP Header : ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือ ไม่มีก็ได้
3. SOAP Body : ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้จากเซอร์วิส



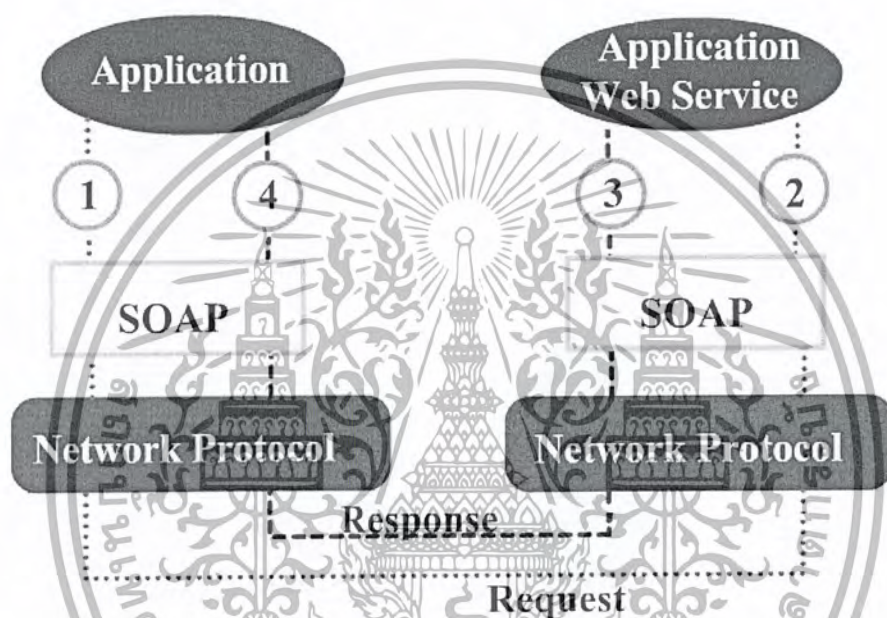
รูปที่ 2.5 แพตฟอร์มของโปรโตคอล SOAP

SOAP กลายเป็นสิ่งที่มีความสำคัญสำหรับ Web Service อย่างรวดเร็ว เป็นโปรโตคอลที่ผู้จัดทำ Web Service เลือกใช้ที่จะส่ง แอสเสท ระหว่าง Web Service SOAP เป็น ทรานสปอร์ตโปรโตคอล (Transport Protocol) ที่มี XML เป็นพื้นฐานและใช้ HTTP เป็นโปรโตคอลร่วมในการส่งผ่านเครือข่าย SOAP จะระบุวิธีในการเข้ารหัสส่วนหัว (Header Encoding) ของทั้ง HTTP และไฟล์ XML ใว้อย่างชัดเจนทั้งใน ส่วนของการติดต่อไปยังคอมพิวเตอร์เครื่องหนึ่งและส่งผ่านข้อมูลไปให้ รวมถึงระบุวิธีที่โปรแกรมซึ่งถูกเรียกนั้นจะส่งค่าคืนกลับมาด้วย

SOAP (Simple Object Access Protocol) เป็น XML-based โปรโตคอล (lightweight protocol) และใช้ HTTP เป็นโปรโตคอลร่วม สำหรับการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (decentralized, distributed environment) SOAP ได้ กำหนดเมสเสจจิงโปรโตคอล (Messaging Protocol) ระหว่างผู้ขอบริการ (requestor) กับผู้ให้บริการ (provider) เช่น ผู้ขอบริการสามารถติดต่อแลกเปลี่ยนข้อมูลกับผู้ให้บริการโดยใช้ RMI (Remote Method Invocation) ตามวิธีการของ โปรแกรมแบบออปเจ็ค บริษัทไมโครซอฟท์, โอบีเอ็ม, โลตัส, ยูสเซอร์แลนด์ (UserLand) และ ดีเวลลอปเปอร์เมนเตอร์ (DeveloperMenter) ได้ร่วมกันกำหนดมาตรฐานของ SOAP ขึ้น ซึ่งต่อมาได้มีบริษัทอีก 30 กว่าบริษัทเข้าร่วมและ จัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

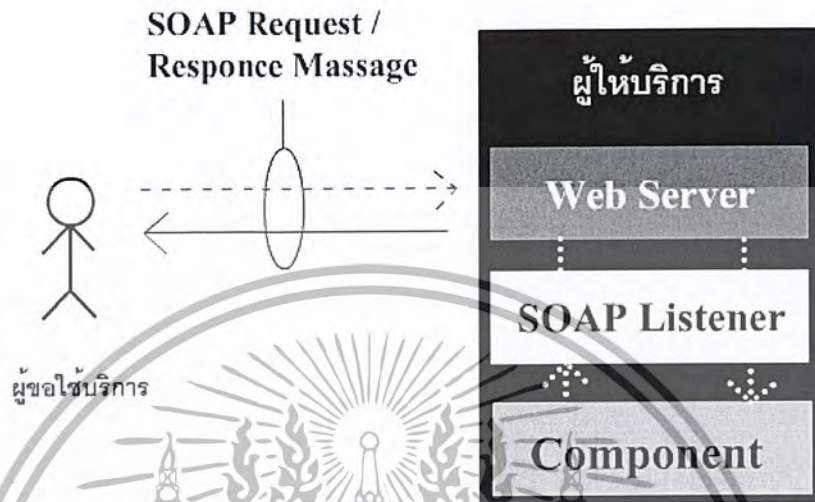
SOAP ได้กำหนดรูปแบบพื้นฐานของการสื่อสารแบบกระจายขึ้น โดย การพัฒนา SOA แม้ว่า SOA จะไม่ได้กำหนดเมจเสจจิง โพรโตคอล (Messaging Protocol) ไว้ แต่ SOAP ได้ถูกกำหนด ให้เป็น Services-Oriented Architecture Protocol เรียบร้อยแล้ว เนื่องจากมันได้ถูกใช้ในการพัฒนา SOA อย่างแพร่หลายแล้วนั่นเอง จุดเด่นของ SOAP ก็คือเป็นโพรโตคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของและเป็นโพรโตคอล ที่ทำงานกับโพรโตคอลอื่นหลายชนิด การพัฒนาก็อนุญาตให้ทำได้ อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการ แบบจำลองทางวัตถุ (Object model) และภาษา โปรแกรมของผู้ที่ทำการพัฒนา



รูปที่ 2.6 การทำงานของโพรโตคอล SOAP

1. ผู้ขอใช้บริการ สร้าง SOAP message เพื่อเรียกใช้บริการของ Web Service แล้วส่งผ่านโพรโตคอล เครือข่าย ไปยังผู้ให้บริการ
2. ผู้ให้บริการได้รับ SOAP message จากผู้ขอใช้บริการซึ่งอยู่ในรูปแบบ XML จึงแปลข้อความนั้นกลับมาอยู่ ในรูปแบบที่เว็บเซิร์ฟเวอร์เข้าใจ แล้วตรวจสอบว่าผู้ขอใช้บริการต้องเรียกใช้ Web Service ชื่ออะไร เมธอดอะไร และส่งพารามิเตอร์อะไรมา จากนั้นจึงส่งไปให้แก่คอมโพเนนต์ที่ให้บริการ Web Service นั้นๆ ดำเนินการต่อไป
3. หลังจากคอมโพเนนต์ที่ให้บริการ Web Service ส่งผลลัพธ์มาแล้ว ผู้ให้บริการก็จะสร้าง SOAP message ที่มีผลลัพธ์นั้นออกมา แล้วจึงส่งผ่านทางโพรโตคอลเครือข่ายไปยังผู้ขอใช้บริการ

4. ผู้ขอใช้บริการได้รับ SOAP message ที่อยู่ในรูปแบบ XML จึงแปลข้อความนั้นกลับมาในรูปแบบที่โปรแกรมของผู้ขอใช้บริการเข้าใจ แล้วนำผลลัพธ์ไปใช้งาน เช่น แสดงผล หรือนำไปทำอย่างอื่น ขึ้นอยู่กับการเขียนโปรแกรมรองรับไว้ว่าให้ทำอย่างไร



รูปที่ 2.7 การทำงานของ SOAP Listener

ภายในเครื่องเว็บเซิร์ฟเวอร์ที่ให้บริการ Web Service จะต้องมีโปรแกรมชนิดหนึ่งเพื่อทำหน้าที่คอยรับฟังว่า มีการเรียกใช้ Web Service หรือยัง โดยบริการ Web Service แต่ละบริการก็จะมีไฟล์ SOAP Listener จำนวน 1 ไฟล์เมื่อใดที่มีการเรียกใช้ Web Service ไฟล์โปรแกรมที่เป็น SOAP Listener ก็จะไปปลุกให้ Web Service นั้นทำงาน

2.1.3.3 องค์ประกอบของ WSDL (Web Service Description Language)

WSDL (Web Services Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของ Web Services และวิธีการติดต่อกับ Web Services ความต้องการของนิยามนี้เกี่ยวข้องกับความต้องการของ distributed system ที่จะกำหนด Interface Definition Language (IDL) โดยใช้ภาษา XML, WSDL เกิดจากการรวมแนวคิดของ NASSL (The Network Accessible Service Specification Language), WDS (Well-Defined Services) ของบริษัทไอบีเอ็ม, SDL (The Service Description Language) และ SCL (the SOAP Contract Language) ของบริษัทไมโครซอฟท์ ปัจจุบัน WSDL เป็นภาษา ที่อยู่ในการดูแลของ W3C (World Wide Web Consortium) โดยใช้ไวยากรณ์ภาษา XML

2.1.3.4 องค์ประกอบของ Universal Description Discovery and Integration (UDDI)

UDDI เป็นมาตรฐานที่จัดตั้งขึ้นโดยบริษัทไอบีเอ็ม , ไมโครซอฟต์ และบริษัทยักษ์ใหญ่ทางธุรกิจ B2B คือ Ariba การที่มีบริษัท Ariba มาเกี่ยวข้องด้วย เป็นการแสดงด้วยว่ามาตรฐานนี้จัดตั้งขึ้นมาเพื่อรองรับการทำธุรกิจอย่างใดอย่างหนึ่งแน่นอน

UDDI ถูกสร้างขึ้นมาเพื่อเป็นมาตรฐานในการค้นหาบริการ Web Service สำหรับคู่ค้าทางธุรกิจ (Business partner) โดย UDDI เปรียบเหมือนฐานข้อมูลขนาดใหญ่ซึ่งมีข้อมูลของ Web Service ที่ให้บริการทางธุรกิจ บางครั้งหากหาข้อมูลใน อินเทอร์เน็ต จะพบคำว่า UDDI Business Registry ซึ่งหมายถึง ฐานข้อมูล Web Service ของบริษัทคู่ค้าทางธุรกิจนั่นเอง

2.1.4 การพัฒนาเว็บเซอร์วิสของบริษัท Microsoft และ Sun One

2.1.4.1 การพัฒนาเทคโนโลยี Web Service ของ Microsoft .Net

.Net เป็นแพลตฟอร์มในการพัฒนาเว็บ และ Web Service โดยองค์ประกอบของแพลตฟอร์ม .Net เป็นดังรูปที่ 2.7



รูปที่ 2.8 แพลตฟอร์มของ .Net

ชั้นล่างสุด คือ ระบบปฏิบัติการ ในเครื่องเซิร์ฟเวอร์ , เดสก์ทอป หรืออุปกรณ์ใดๆ ที่โปรแกรมทำงาน ถัดมาในชั้นที่ 2 แบ่งองค์ประกอบออกเป็น 3 ส่วนคือ .Net Enterprise Server , .Net Framework , .Net Building Block Service และในชั้นสุดท้ายคือ Visual Studio .Net ซึ่งเป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรม หรือ แอปพลิเคชันให้เป็นไปอย่างง่ายดาย

.Net Enterprise Server ที่อยู่ในชั้นที่สองก็คือ ผลิตภัณฑ์ต่างๆ เช่น ฐานข้อมูล SQL Server 2000 เป็นต้น ผลิตภัณฑ์เหล่านี้ถูกออกแบบมาเพื่อรองรับการพัฒนา แอปพลิเคชัน ในระดับองค์กร โดยรองรับข้อมูลในรูปแบบ ภาษา XML ส่วน .Net Building Block Service คือ บริการ Web Service ที่ทางไมโครซอฟต์ สร้างขึ้นมาเพื่อให้นักพัฒนาใช้ ในช่วงระยะแรกๆ มีบริการที่เตรียมไว้ เช่น Microsoft Passport ก็คือการใช้ (user) กรอกข้อมูลบางอย่างเช่น username และ password

เพียงครั้งเดียวก็สามารถเข้าออก ในเว็บไซต์ ต่างๆ ที่ใช้ระบบ Microsoft Passport ได้ทันที และใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่สำคัญก็คือ .Net Framework ซึ่งภายในมีองค์ประกอบที่จะเอื้อให้ผู้ใช้สามารถใช้งานเพื่อพัฒนา Web Service ให้เกิดประโยชน์สูงสุดได้โดยง่าย

2.1.4.2 การพัฒนาเทคโนโลยี Web Service ของ Sun One

ภาษาจาวามีแนวคิดในการพัฒนาโปรแกรมให้ทำงานในแพลตฟอร์มอะไรก็ได้ โดยที่แพลตฟอร์มนั้นๆ จะต้องมีการ Virtual Machine เพื่อที่จะให้โค้ดเหล่านั้นทำงาน สำหรับภาษาจาวาก็มี Virtual Machine ที่มีชื่อว่า JVM (Java Virtual Machine) โดยการพัฒนาจะเริ่มจากการเขียนโปรแกรมด้วยภาษาจาวาเมื่อทำการบันทึกข้อมูลและทำการคอมไพล์เป็นไบต์โค้ด (ไฟล์ .class) แล้วจะสามารถนำไปใช้ในแพลตฟอร์มใดก็ได้ที่มี JVM

ทางบริษัท Sun ONE ได้พัฒนามาตรฐาน API ใหม่ๆ มาชุดหนึ่งซึ่งจะใช้จัดการกับข้อมูลที่อยู่ในรูป XML ประกอบไปด้วย JAXP (Java API for XML Parsing), JAXB (Java API for XML Binding), JAXR (Java API for XML Registries) และ JAX-WS โดยที่แต่ละ Package ทำหน้าที่แตกต่างกันออกไป

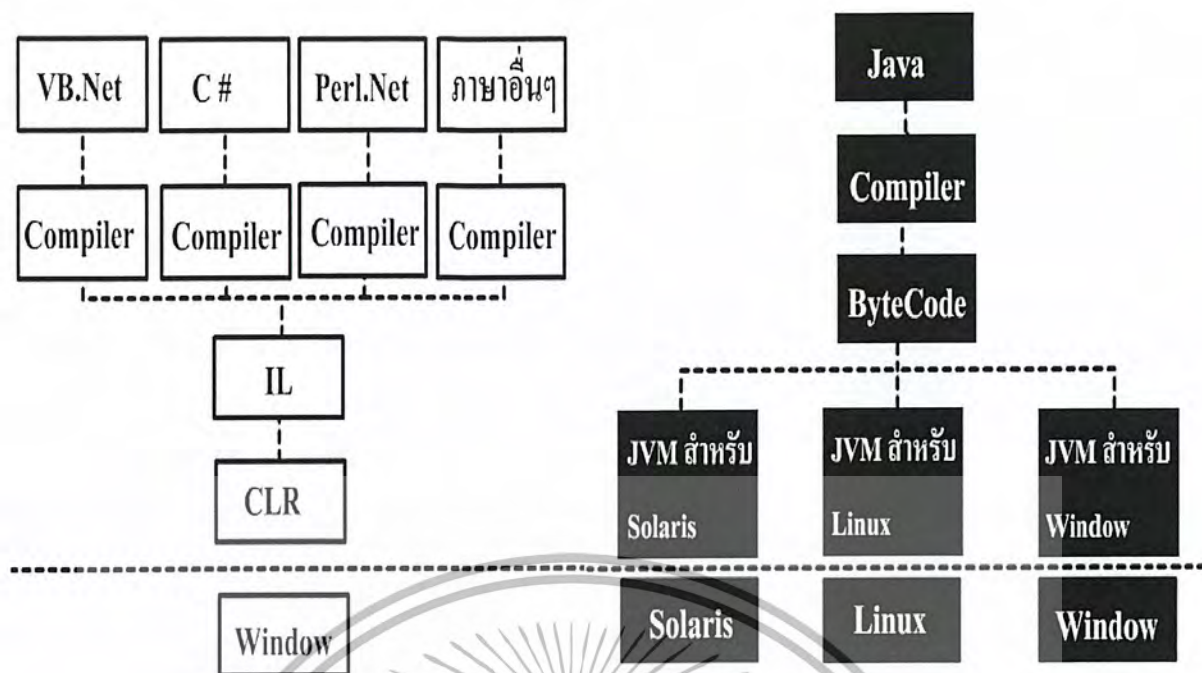
ตารางที่ 2.2 ตัวอย่าง Packet API ที่ Java ได้จัดเตรียมไว้เพื่อความสะดวกในการสร้าง Web Service

ชื่อ API	ชื่อเต็ม	หน้าที่การใช้งาน
JAXP	Java API for XML Prasing	ใช้ในการเข้าถึงข้อมูลแบบ XML
JAXB	Java API for XML Binding	เป็นกลไกในการแปลงข้อมูล XML มาอยู่ในรูปแบบของ Class ใน Java
JAXM	Java API for XML – base Massaging	ใช้เป็นมาตรฐานในการสื่อสารข้อมูล
JAXR	Java API for XML Registries	เป็นมาตรฐานในการเข้าถึง UDDI และ ebXML registries
JAX-WS RPC	Java API for XML – base Remote Process Communication	เป็นมาตรฐานในการร้องขอเมธอดของบริการที่อยู่ในเซิร์ฟเวอร์อื่นๆ

2.1.5 การเปรียบเทียบความแตกต่างระหว่างการพัฒนาของ ทั้ง สองบริษัท

ความแตกต่างกันของทั้ง 2 ค่าเราสามารถสรุปเป็นแผนภาพได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การเปรียบเทียบการทำงานระหว่าง .Net กับ Java

โดยจะสังเกตได้ว่า ส่วนการพัฒนาโปรแกรมในแพลตฟอร์ม .Net ผู้ใช้จะสามารถเลือกใช้ภาษาใด ก็ได้ในการพัฒนา แต่ภาษานั้นต้องมีคอมไพเลอร์เฉพาะ ที่คอมไพล์ โปรแกรมจากแต่ละภาษา มาอยู่ในโปรแกรมอีกรูปแบบหนึ่งที่เรียกว่า IL (Intermedia Language) ซึ่งเป็นรูปแบบเฉพาะที่ทำงานใน CLR บนแพลตฟอร์ม Windows ในขณะที่ทาง Sun ออกแบบในแนวทางกลับกัน คือ การพัฒนาจะอิงอยู่กับภาษา Java เท่านั้นแต่โปรแกรมที่พัฒนาออกมาจะสามารถทำงานได้ในหลายแพลตฟอร์ม

2.2 การใช้งาน Servlets

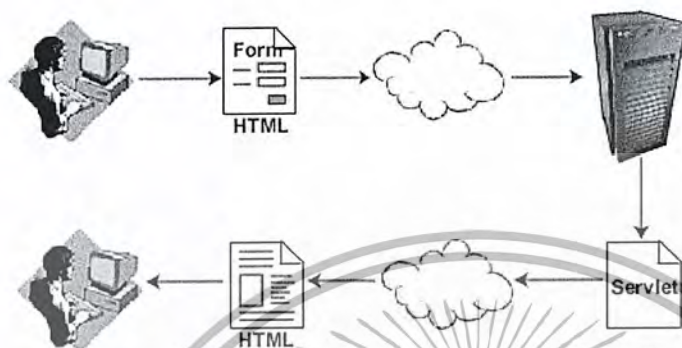
2.2.1 ความหมาย และหน้าที่ของ Servlets

Servlets คือ component ที่พัฒนาขึ้นด้วย Java Technology ให้มีความสามารถในการจัดการกับ Dynamic Content ของ Web Applications ซึ่ง Servlets นั้นจะเป็น โปรแกรมที่ทำงานอยู่บน Web Server ทำหน้าที่เป็นตัวกลางระหว่างคำสั่งที่ได้รับจาก Web Browser หรือ HTTP Client กับฐานข้อมูล หรือ Applications ต่าง ๆ บน HTTP Server

Servlets ได้รับการออกแบบให้ทำงานภายใต้ Request/Response Processing Model ดังแสดงในรูปที่ 1 ภายใน Request/Response Model นั้น ไคลเอนท์ จะส่ง Request Message ไปที่ จะตอบกลับด้วย Reply Message โดยที่ Request สามารถอยู่ในรูปแบบของ HTTP URL, FTP URL หรือ Custom Protocol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Request และ Response แสดงให้เห็นถึงสถานะของ ไคลเอนท์ และ เซิร์ฟเวอร์ ในขณะที่มีการรับส่งข้อมูลระหว่างกัน โดยทั่วไปแล้ว สถานะของ ไคลเอนท์/เซิร์ฟเวอร์ Connection จะไม่ถูกเก็บไว้ภายใต้ Request/Response ในช่วงเวลาที่แตกต่างกัน แต่หากเก็บค่าของ Request/Response เหล่านั้นไว้ใน Session แล้ว ข้อมูลจะได้รับการเก็บไว้ใน Servlets



รูปที่ 2.10 รูปแบบการทำงานของ Servlets

Servlets จะมี Servlet Container หรือที่เรียกว่า “Servlet Engines” ซึ่งเป็นส่วนหนึ่งของ Web Server เป็นตัวคอยจัดการการโต้ตอบระหว่าง Web Clients กับ Servlets ที่อยู่ทาง Web Server ตัวอย่างขั้นตอนการทำงานของ Servlets มีดังนี้

1. ไคลเอนท์ เช่น Web Browser เข้า Web Server และส่ง HTTP Request
2. Web Server ได้รับ Request และส่งต่อไปที่ Servlet Container
3. Servlet Container จะพิจารณาว่า ควรจะใช้งาน Servlet ตัวใด โดยพิจารณาจาก Configuration ของ Servlet นั้น ๆ และจะเรียกใช้งาน Servlet โดยพิจารณาจาก Request ที่ได้จาก ไคลเอนท์ และ Response ที่ได้รับจาก เซิร์ฟเวอร์
4. Servlet ทรอบถึงข้อมูลต่างๆ ผ่านทาง Request Object ที่ได้รับจาก เซิร์ฟเวอร์ หลังจากนั้น Servlet จะทำการประมวลผลและส่งผลลัพธ์กลับไปยัง ไคลเอนท์ ผ่านทาง Response Object
5. หลังจากที่ Servlet ประมวลผลเสร็จเรียบร้อย Servlet Container จะทำการตรวจสอบว่า Response มีความเรียบร้อยสมบูรณ์ แล้วจึงส่งหน้าที่กลับไปให้ Web Server ทำการส่ง Response กลับไปยัง ไคลเอนท์ ต่อไป

โดยสรุปหน้าที่หลักของ Servlet มีดังนี้

1. อ่านข้อมูลที่ได้รับจาก ไคลเอนท์

ข้อมูลส่วนใหญ่จะได้รับมาจากฟอร์มบน Web Page นอกจากนี้ยังสามารถนำมาจาก Java Applet หรือ โปรแกรม HTTP Client อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตรวจสอบข้อมูลต่างๆ ที่เกี่ยวกับ Request ที่อยู่ใน HTTP Request ตัวอย่างของข้อมูลเหล่านี้ได้แก่ ความสามารถของ Browser, Cookies, Host Name ของ โคลแอนท์ และข้อมูลอื่นๆ
3. ประมวลผล
ขั้นตอนนี้ Servlet อาจจะต้องติดต่อกับฐานข้อมูล หรือ โปรแกรมอื่นๆ หรืออาจจะคำนวณผลลัพธ์โดยตรง
4. จัดการกับรูปแบบของผลลัพธ์
โดยทั่วไปแล้วขั้นตอนนี้เป็นการจัดการกับข้อมูลใน HTML Page
5. กำหนด HTTP Response Parameters ที่เหมาะสม
ขั้นตอนนี้ Servlet จะบอก Browser ถึงชนิดของเอกสารที่ส่งกลับ การ Set Cookies และ Cache Parameters รวมไปถึงงานอื่นๆ ที่เกี่ยวข้อง
6. ส่งเอกสารกลับไปยัง โคลแอนท์
เอกสารที่ได้รับการส่งกลับไปยัง โคลแอนท์ นั้น อาจจะเป็น HTML Format หรือรูปภาพก็ได้

2.2.2 ข้อดีของ Servlets

เมื่อพิจารณาถึงหน้าที่ของ Servlets แล้วพบว่า Servlets มีการทำงานในลักษณะเดียวกับ Common Gateway Interface (CGI) Programs และ เซิร์ฟเวอร์ Extensions อื่นๆ เช่น Netscape Server API (NSAPI) หรือ Apache Modules อย่างไรก็ตาม Servlets มีข้อดีเหนือเทคโนโลยีดังกล่าวต่อไปนี้

2.2.2.1 Servlets มีการทำงานที่เร็วกว่า CGI Scripts เนื่องจากขั้นตอนการทำงาน ที่แตกต่างกัน

การทำงานของ CGI Programs นั้น จะมีการเริ่ม Process ใหม่ทุกครั้ง ที่มี HTTP Request และ ถ้าหากมี Request จำนวน N ครั้งไปที่ CGI Programs นั้น Code ของ CGI Programs นั้นจะถูก Load เข้าสู่ Memory เป็นจำนวน N ครั้ง เช่นเดียวกัน ซึ่งเป็นการสิ้นเปลือง Resource มาก สำหรับ Servlets นั้น ทุกๆ Request จะเป็น Lightweight Java Thread ซึ่งได้รับการควบคุมโดย Java Virtual Machine และถ้ามี Request จำนวน N ครั้งไปที่ Servlets นั้น Servlet Class จะถูก load เพียงครั้งเดียว ถึงแม้ว่าจะมี N Threads

2.2.2.2 Servlets ใช้ Standard API ที่ได้รับการสนับสนุนจากหลายๆ Web Servers

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันมีบริษัท Third Party หลายบริษัทที่นำเสนอ Web Server ที่สนับสนุนการทำงานของ Servlet และ JSP ตัวอย่างเช่น Apache Web Server , iPlanet Web Server และ Microsoft IIS เป็นต้น นอกจากนี้ Servlet Container ยังสามารถนำไปผนวกเข้ากับ Web-Enabled Application Server เช่น BEA Web Logic Application Server, IBM WebSphere และ iPlanet Application Server เป็นต้น Servlets สามารถพูดคุยกับ Web Server ได้โดยตรง ในขณะที่ CGI Programs ไม่สามารถทำได้ถ้าไม่ใช่ Server-specific API การสื่อสารกับ Web Server โดยตรงมีข้อดีหลายประการ เช่น ทำให้การแปลง Relative URLs ไปเป็น Path Names ที่ถูกต้องได้ง่ายขึ้น Servlets หลายๆ โปรแกรมยังสามารถใช้ข้อมูลร่วมกัน ทำให้การพัฒนา Database Connection Pooling และ Resource Sharing สะดวกขึ้น นอกจากนี้ Servlets ยังสามารถรักษาข้อมูลจาก Request หนึ่งไปยังอีก Request หนึ่งได้โดยการใช้เทคนิคของ Session Tracking และ Computation Caching

2.2.2.3 Servlets สามารถใช้ประโยชน์จาก Java Programming Languages ในเรื่องของความสะดวกในการพัฒนา และความเป็นอิสระจาก Platform ใดๆ

Servlets ถูกเขียนขึ้นมาจากภาษา Java และตรงกับ Standard API ทำให้ Servlets สามารถเข้าถึง APIs ที่มีมากมายของ Java Platforms ด้วยเหตุนี้ทำให้ Servlets สามารถ run อยู่บน Web Server ต่างชนิดกันโดยไม่ต้องมีการแก้ไข Code ตัวอย่างเช่น Servlets ที่ run อยู่บน Java Web Server สามารถ run บน Apache Tomcat ได้ โดยไม่ต้องมีการแก้ไขเปลี่ยนแปลง Code ในปัจจุบันมี Web Server จำนวนมากมายที่สนับสนุน Java 2 Platform, Enterprise Edition (J2EE) ซึ่งจะเห็นแนวโน้มของการใช้ Servlets ที่เพิ่มขึ้นด้วย

2.2.3 การติดตั้งเครื่องมือสำหรับการพัฒนา

ก่อนที่จะทำการพัฒนา Servlets หรือ JSP Application ได้ั้น ต้องมีการติดตั้งซอฟต์แวร์ต่างๆ ที่จำเป็นให้ครบก่อน รวมทั้งกำหนดค่าต่างๆ ให้ถูกต้องเหมาะสม เพื่อให้การทำงานของโปรแกรมเป็นไปอย่างถูกต้อง

เริ่มแรกนั้น จะต้องมีติดตั้ง Java 2 SDK เพื่อใช้ในการพัฒนา Java Application โดย Version ที่ใช้ต้องเป็น Version 1.2 ขึ้นไป

2.2.3.1 การติดตั้งเซิร์ฟเวอร์

ผู้พัฒนาสามารถเลือก Download Software ที่สนับสนุน Java Servlet 2.1, 2.2 หรือ 2.3 และ Java Server Pages 1.0, 1.1 หรือ 1.2 Specifications ปัจจุบันมีบริษัท Third Party หลายบริษัทที่นำเสนอ Web Server ที่สนับสนุนการทำงานของ Servlets และ JSP ดังที่ได้กล่าวถึงไปแล้วในส่วน

ของข้อดีของ Servlets ผู้พัฒนาสามารถตรวจสอบว่ามี เซิร์ฟเวอร์ ไคบ้างที่สนับสนุนการพัฒนา Servlets และ JSP ได้ที่ <http://java.sun.com/products/index.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้ง เซิร์ฟเวอร์ ขอเลือกใช้ Apache Tomcat ซึ่งเป็น Free Software ที่สนับสนุนการพัฒนา Java Servlet และ JSP Specifications ดังกล่าว สำหรับทำปัญหาพิเศษนี้

Tomcat เป็น Servlet Container ที่ใช้สำหรับ Java Servlet และ Java Server Pages ผู้พัฒนาสามารถใช้ Tomcat บน Stand-Alone Computer เพื่อใช้ในการทดสอบ Servlets และ JSP Pages สามารถติดตามรายละเอียดได้ที่ <http://jakarta.apache.org>.

2.2.4 ความรู้พื้นฐานเกี่ยวกับ Java Servlet API

API หรือ Application Programming Interface เป็นรูปแบบมาตรฐานที่ผู้พัฒนาใช้อ้างอิงในการพัฒนา Application ซึ่ง API จะเกี่ยวข้องกับการเข้าถึงลักษณะ และสถานะของ Classes และ Objects ต่างๆ ที่มีในโปรแกรม

Java Application Programming Interface (Java API) เป็นมาตรฐาน JDK (Java Development Kit) ที่ผู้พัฒนาใช้อ้างอิงในการพัฒนา Java Program โดยจะถูกเก็บรวบรวมไว้ภายใต้ Package ของ java เช่น java.io.* หรือ java.util.* เป็นต้น สำหรับ Java Servlet API นั้น เป็นส่วนที่เพิ่มเติมขึ้นจากมาตรฐาน JDK หรือที่เรียกว่า “JDK Extensions” โดย Interface ต่าง ๆ จะถูกเก็บรวบรวมไว้ภายใต้ Package ของ javax

Java Servlet API ประกอบด้วย Package ดังต่อไปนี้

1. Package javax.servlet
2. Package javax.servlet.http

Servlet API สนับสนุนการทำงานหลัก 4 ข้อ คือ

1. การจัดการกับ Servlet Life Cycle
2. การเข้าถึง Servlet Context
3. การใช้งาน Utility Classes
4. การใช้งาน HTTP-Specific Support Classes

2.2.5 โครงสร้างของ Servlets

Servlet Interface เป็นองค์ประกอบที่สำคัญของ Servlet API ทุก Servlets จะ implement Interface นี้โดยตรง หรือโดยการ extend Class ที่ implement Interface นี้ Class 2 Class ที่ implement Servlet Interface ได้แก่ Generic Servlet และ HttpServlet โดยส่วนใหญ่ผู้พัฒนาจะสามารถใช้ HTTP Servlet ในการพัฒนา Servlets

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Servlet Interface มี service Method สำหรับจัดการกับ Client Requests ซึ่ง Method นี้จะถูกเรียกใช้สำหรับทุก ๆ Request ที่ Servlet Container (ซึ่งมีหน้าที่จัดการกับ Request และ Response) จะจัดการกับ Instance ของ Servlet

HttpServlet Abstract Subclass มี Method อื่นๆ นอกเหนือจากที่มีใน Servlet Interface ซึ่ง Method เหล่านี้จะถูกเรียกโดย service Method ใน HttpServlet class เพื่อจัดการกับการประมวลผล HTTP Requests Method เหล่านี้ได้แก่

1. doGet สำหรับการจัดการกับ HTTP GET Requests
2. doPost สำหรับการจัดการกับ HTTP POST Requests
3. doPut สำหรับการจัดการกับ HTTP PUT Requests
4. doDelete สำหรับการจัดการกับ HTTP DELETE Requests
5. doOptions สำหรับการจัดการกับ HTTP OPTIONS Requests
6. doTrace สำหรับการจัดการกับ HTTP TRACE Requests

โดยส่วนใหญ่แล้ว ผู้พัฒนาจะใช้ doGet และ doPost Method ในการพัฒนา HTTP Based Servlets

GET และ PUT Method เป็นวิธีที่ Browser ใช้ในการส่งข้อมูลจาก โคลแอนท์ ไปยัง เซิร์ฟเวอร์ โดยผู้พัฒนาจะสามารถกำหนด Method ที่ต้องการที่ METHOD Attribute ใน HTML Form เช่น

```
<FORM ACTION="requestServlet" METHOD = "POST" Name = "Form1">
.....
.....
</FORM>
```

เมื่อข้อมูลถูกส่งกลับไปยัง Form แล้ว Get จะนำข้อมูลที่ โคลแอนท์ ต้องการส่งไปยัง เซิร์ฟเวอร์ ไปต่อท้าย URL เมื่อข้อมูลไปถึง เซิร์ฟเวอร์ แล้ว Servlets จะทำการตัดเอาข้อมูลเหล่านั้นไปประมวลผล แต่ POST จะใส่ข้อมูลที่ได้รับจาก โคลแอนท์ ลงในข้อมูลอีกส่วนหนึ่ง ทำให้ข้อมูลที่ต้องการส่งไม่ปรากฏให้เห็นใน URL และถูกส่งโดยตรงไปยัง Servlets

Servlets จะ extend HttpServlet และ override doGet หรือ doPost ขึ้นอยู่กับว่าข้อมูลจะถูกส่งด้วยวิธี GET หรือ POST ถ้าผู้พัฒนาต้องการให้ Servlets จัดการทั้ง GET และ POST และใช้ action เดียวกัน ผู้พัฒนาสามารถใช้ doGet เรียก doPost หรือทำในทางกลับกันคือ ใช้ doPost เรียก doGet

ต่อไปจะเป็นลักษณะโครงสร้างของ Servlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/** โปรแกรมง่าย ๆ ที่แสดงให้เห็นถึง Servlets Structure
**/

// การ import Java Packages
import java.io.*;

// การ import Servlet Packages
import javax.servlet.*;
import javax.servlet.http.*;

public class ServletStructure extends HttpServlet
{
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        //รับ parameters จาก request

        //ประมวลผล

        // รับ print writer object
        PrintWriter out = response.getWriter();

        // แสดงผล
    }
}

```

doGet และ doPost Method จะรับ 2 Arguments ได้แก่ HttpServletRequest และ HttpServletResponse โดย HttpServletRequest จะมี Methods ที่บอกให้ทราบถึงข้อมูลที่เข้ามา ได้แก่ ข้อมูลจาก Form, HTTP Request Headers และชื่อ Host ของ ไคลเอนท์ ส่วน HttpServletResponse ช่วยให้ผู้พัฒนาสามารถระบุข้อมูลที่ส่งกลับไป เช่น HTTP Status Codes (200, 404, และอื่นๆ) และ Response Headers (Content-Type, Set-Cookie, และอื่นๆ) นอกจากนั้นผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนายังสามารถรับตัวแปร PrintWriter สำหรับส่งข้อมูลกลับไปยัง ไคลแอนท์ ได้ เป็นที่ทราบกันว่า Servlets เปรียบเสมือนการนำเอา HTML ไปใส่ไว้ใน Java File

ดังนั้น ผู้พัฒนาจะต้องใช้คำสั่ง println ในการจัดการกับ Page ที่เป็นผลลัพธ์มากพอสมควร ซึ่งจะได้เห็นในตัวอย่างโปรแกรมที่จะนำเสนอต่อไป

doGet และ doPost จะ throw 2 Exceptions คือ ServletException และ IOException ดังนั้นผู้พัฒนาจะต้องใส่ Exceptions ทั้งสองไว้ด้วย

นอกจากนี้ผู้พัฒนาจะต้อง

- import Class จาก java.io สำหรับ PrintWriter Object และอื่นๆ
- import Class javax.servlet สำหรับ HttpServlet และอื่นๆ
- import Class javax.http สำหรับ HttpServletRequest และ HttpServletResponse

2.3 ความรู้เบื้องต้นเกี่ยวกับ RMI

2.3.1 ความหมายของ RMI

RMI เป็นเทคโนโลยีที่ช่วยนักพัฒนาในการเขียน Distributed Object Computing โดยจะอนุญาตให้โปรแกรมเมอร์สามารถเรียกใช้ method และ object จากเครื่องคอมพิวเตอร์เครื่องอื่นได้ โดยโครงสร้างที่ได้รับความนิยม คือ CORBA และ Java RMI ซึ่งเมื่อเปรียบเทียบกันแล้ว CORBA จะเด่นในเรื่องของการไม่ขึ้นกับภาษาใดภาษาหนึ่ง แต่ Java RMI จะมีข้อดีในเรื่องที่มีความยืดหยุ่นมากกว่า CORBA เนื่องจากหลาย platform ใช้ Java RMI ในการติดต่อกันระหว่าง framework แต่ข้อเสียก็คือ จะต้องใช้ร่วมกับภาษา Java เท่านั้น และเป็นที่คาดการณ์ว่าในอนาคต อุปกรณ์ไร้สายจะใช้ Java RMI กันมากยิ่งขึ้น

2.3.2 ที่มาของ Java RMI

2.3.2.1 Protocol ของ Java RMI

Java RMI ถูกออกแบบมาให้ง่ายต่อการติดต่อสื่อสารระหว่าง Object ที่มี virtual machines ต่างกัน เนื่องจากว่า Remote Object อยู่ใน virtual machines ที่ต่างกัน ดังนั้นจะมี RMI Registry เป็นตัวช่วยในการจัดการกับการอ้างอิงในระยะไกล โดยเมื่อ RMI Server ต้องการสร้าง method ที่จะใช้ในการติดต่อไปหา Remote Object ตัว register ของมันจะเรียก method ภายในตัวเอง เพื่อให้ทำหน้าที่ในการติดต่อ และจากนั้น Remote Object จะทำการเชื่อมต่อกับ Remote Register ของตัวเซิร์ฟเวอร์ ซึ่งจะทำหน้าที่รองรับ socket และรับข้อมูลที่ถูกส่งมา

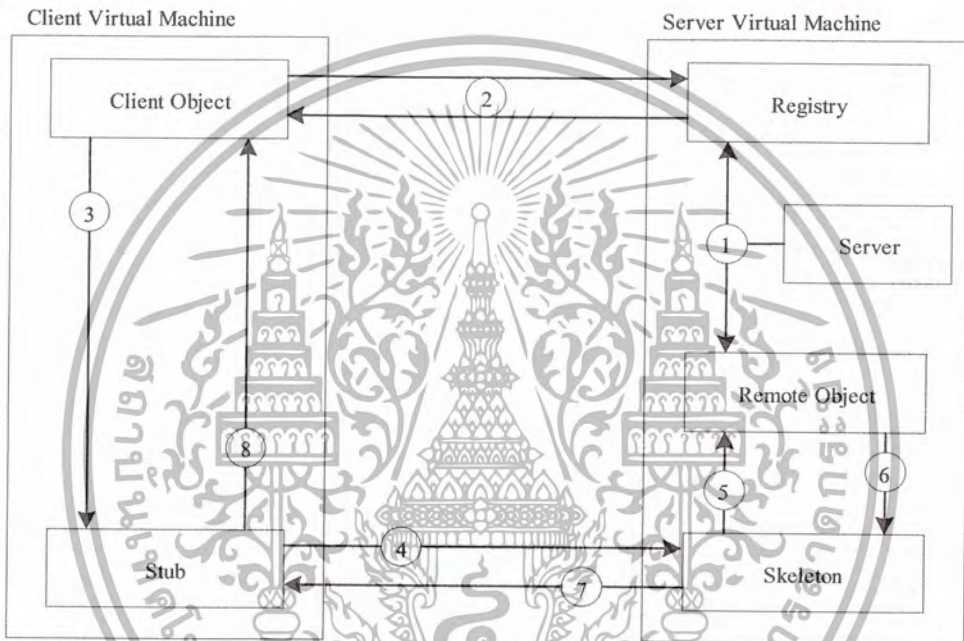
Java RMI ถูกสร้างอยู่ที่ส่วนบนสุดของ Transport Layer ซึ่งทำหน้าที่ในการรับ และกรองการติดต่อแบบ RMI ที่อยู่บนการติดต่อถึงกันของ TCP โดยเมื่อเกิดการเชื่อมต่อ RMI ขึ้น

Transport Layer อาจจะเปิดการติดต่อ TCP ให้ใหม่ 1 การติดต่อ หรืออาจจะนำการติดต่อเดิมซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปเผยแพร่โดยไม่ผิดเงื่อนไขใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนนี้ว่างอยู่มาใช้แทนอย่างใดอย่างหนึ่งก็ได้ และถ้าการติดต่อเดิมที่จะนำมาใช้นั้นถูกปล่อยให้อยู่เป็นเวลานานมากกว่า 1 รอบเวลาการทำงาน (the time of a round-trip) Transport Layer จะทำการส่งสัญญาณไป ping packet นั้น เพื่อให้แน่ใจว่าการติดต่อนั้นจะยังคงใช้งานได้อยู่ ซึ่งเมื่อมีการตอบรับโดยการส่งสัญญาณ Ack (Acknowledgment) กลับมาหลังจากการ ping ไปของ Transport Layer ก็จะมีการติดต่อ RMI โดยผ่านการติดต่อเดิมที่ยังว่างอยู่นั้นได้ทันที ในขณะที่การติดต่อทาง TCP ไม่ได้ถูกใช้เพื่อเชื่อมการติดต่อ RMI เป็นช่วงเวลาหนึ่ง TCP ตัวนั้นจะถูกปิด

สถาปัตยกรรมโดยทั่วไปของ RMI สามารถศึกษาได้ตามรูป



รูปที่ 2.14 แสดง Protocol ของ Java RMI

1. เซิร์ฟเวอร์ จะสร้าง Remote Object และ Register เพื่อให้เป็น Local Registry
2. ไคลเอนท์ จะทำการเชื่อมต่อไปยัง Remote Registry และรอรับข้อมูล ในขั้นตอนนี้ Stub ของ Remote Object จะถูกส่งมาจาก Remote Virtual Machines ไปยัง Client Virtual Machines ในกรณีที่ Stub ยังไม่ถูกสร้าง
3. เมื่อ ไคลเอนท์ ทำการร้องขอ Method ก็จะไปร้องขอกับ Local Stub ก่อน
4. ตัว Stub จะจัดการเอนคริป parameter และส่ง message ไปยัง Skeleton ที่จะอยู่ทางฝั่ง เซิร์ฟเวอร์
5. Skeleton จะดีคริป parameter และจัดการร้องขอ Method ที่เหมาะสมกับ parameter นั้น
6. Remote Object จะเรียก Method และทำการส่งค่ากลับไปยัง Skeleton
7. ที่ Skeleton จะเอนคริป ค่าที่ได้จาก 6. และส่ง message ไปยัง Stub ที่อยู่ทางฝั่ง ไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ทำที่สุด stub จะตีคริปนำค่าออกมา และส่งค่านี้ไปยัง ไคลแอนท์

2.3.2.2 ปัญหาของ RMI

RMI ที่ใช้ในการติดต่อผ่าน TCP

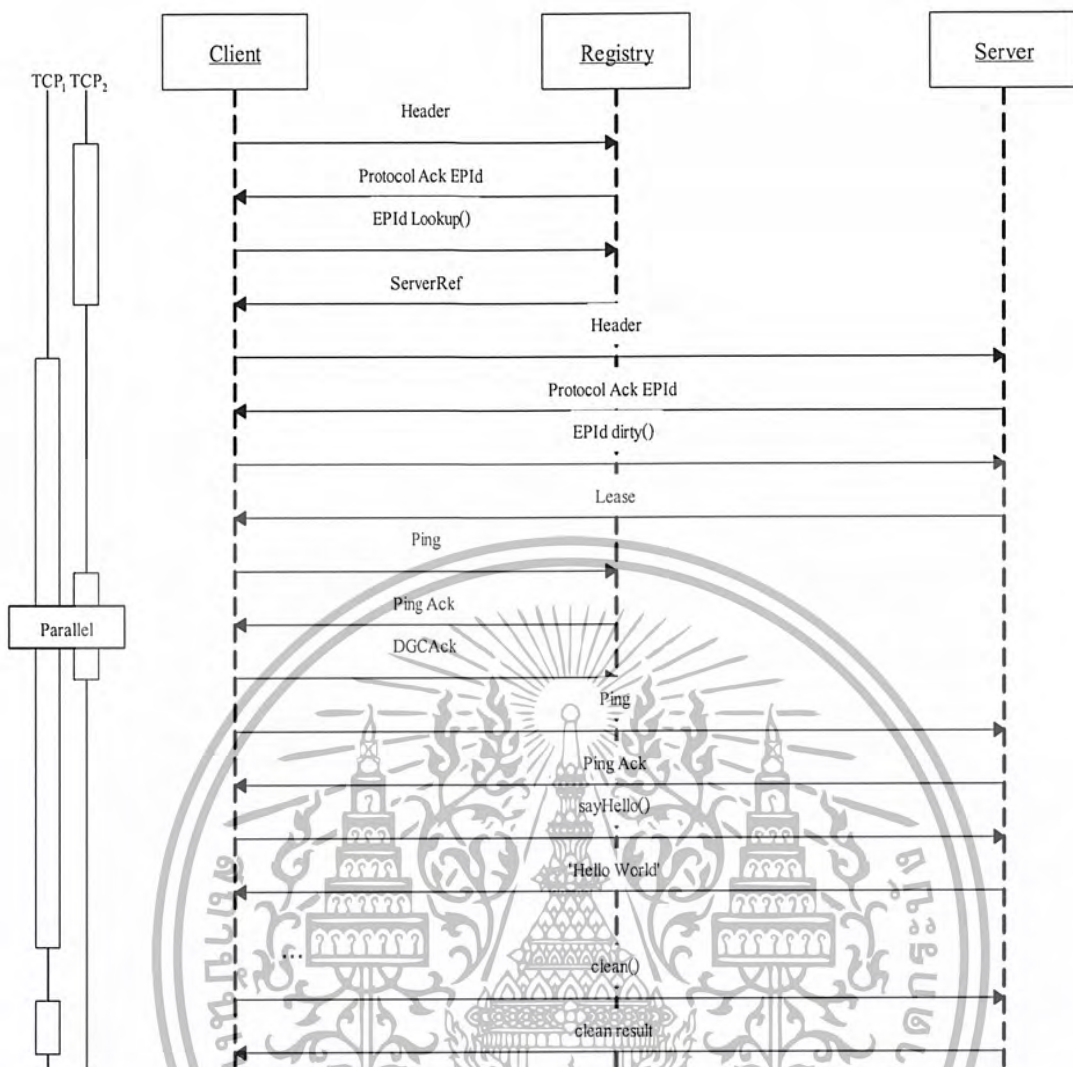
การที่มีกรนำ TCP ที่ใช้ไปแล้วกลับมาใช้ใหม่ใน Transport Layer ทำให้ประหยัดทรัพยากร อย่างไรก็ตาม การที่จะต้องส่งสัญญาณไป ping ตรวจสอบนั้น ก็อาจทำให้เกิดปัญหาในกรณีที่เป็น high-round-trip การติดต่อไร้สาย

ขณะที่ Java RMI ไม่เหมาะสมกับระบบเครือข่ายไร้สาย และการใช้ TCP เพื่อสร้าง Java RMI ซึ่งปัญหาเกี่ยวกับ TCP เป็นปัญหาที่สำคัญมาก โดยเฉพาะอย่างยิ่งใน JDK 1.1 ที่ RMI และ TCP ทำงานร่วมกันทำให้ผลลัพธ์ออกมาไม่ดีนัก เนื่องจาก RMI จะทำการเขียนหัวของข้อมูล byte ต่อ byte และเป็นเพราะว่ามี algorithm ที่ทำงานช้า ทำให้การที่ TCP ส่งสัญญาณไป และกลับไปได้ใช้เวลาเต็ม หรือหนึ่ง round-trip ใน JDK 1.2 (Java™ 2) ข้อมูลที่มีขนาดไม่ยาวเกินไป จะถูกเขียน byte ต่อ byte และผลการทำงานดีขึ้น แต่ protocol ยังคงใช้ round-trip หลายรอบการทำงานในการส่ง 1 คำสั่งอยู่ดี

วิเคราะห์การทำงานของ RMI

จะทำการวิเคราะห์การทำงานของ RMI โดยผ่านการเรียก method “sayHello” โดยผลลัพธ์จะมีการส่งค่า String “Hello World” ออกมา

จากตัวอย่างนี้ ตัว stub class จะอยู่ทางฝั่ง ไคลแอนท์ จึงไม่จำเป็นต้องทำการ download stub class



รูปที่ 2.15 แผนภาพแสดงการส่งข้อมูล “sayHello” ผ่านทาง RMI

1. Round-trip แรกจะอยู่ระหว่าง ไคลแอนท์ กับ Registry บน Remote Side โดยใช้การติดต่อกับ TCP ที่ถูกสร้างขึ้นใหม่ (TCP1) จะเห็นว่า Registry จะส่งค่า Ack พร้อมทั้ง IP address ของ ไคลแอนท์ (EPId) กลับ จะมีการทำงานตามขั้นตอนนี้ทุกครั้งที่มีการสร้างการติดต่อใหม่
2. ในการติดต่อ Round-trip ที่ 2 ทางฝั่ง ไคลแอนท์ จะร้องขอข้อมูลจาก class ที่ตนเองต้องการ ซึ่งในตอนนี้ การติดต่อ RMI จะถูกปิด และในทางเทคนิคก็จะทำการปิดการติดต่อ TCP1 ด้วย
3. จะมีการเปิดการติดต่อ TCP ชื่อ TCP2 โดย ไคลแอนท์ จะเรียกไปยัง เซิร์ฟเวอร์ เนื่องจากการเรียกผ่าน RMI เป็นครั้งแรกจึงมีการแลกเปลี่ยน header และ สัญญาณ Ack
4. ทางฝั่ง ไคลแอนท์ เป็น Distributed Garbage Collection (DGC) จะทำการร้องขอ Dirty() จาก เซิร์ฟเวอร์ จากนั้น การติดต่อ RMI จะถูกปิดส่งผลให้ TCP2 ถูกปิดไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ในจุดนี้ทางฝั่ง ไคลแอนท์ จะต้องบอก DGC ของ registry ว่าทาง ไคลแอนท์ ได้ข้อมูลแล้วดังนั้นจะมีการเปิดการติดต่อ RMI ใหม่ทำให้ Round-trip ระหว่าง ไคลแอนท์ และ registry ถูก ping ทำให้การติดต่อ TCP (TCP1) ที่เคยถูกปิดไปกลับมาทำงานอีกครั้ง การติดต่อของ ไคลแอนท์ ไปยัง registry จะได้รับการยืนยันว่าจะสามารถใช้ข้อมูลจาก เซิร์ฟเวอร์ ได้ จากการส่ง DGCAck message

6. ในช่วงการทำงานที่ขนานกันตามรูป ของ TCP ทางฝั่ง ไคลแอนท์ จะสามารถร้องขอ remote method จากเซิร์ฟเวอร์ ได้แต่ถ้า RMI ถูกปิดอยู่ จะมีการ ping เพื่อตรวจสอบค่าหลังจากนี้

7. เมื่อ ไคลแอนท์ ไม่จำเป็นต้องใช้ข้อมูลต่อไป ทาง ไคลแอนท์ จะส่ง message “ clean ” ไปหา เซิร์ฟเวอร์

2.4 Domain ของระบบการจำหน่ายสินค้า

ระบบนี้เป็นระบบการขายสินค้า โดยเมื่อพนักงานต้องการเข้าใช้โปรแกรมจะต้องทำการ log in จากนั้นระบบจะทำการตรวจสอบเองว่า username และ password ถูกต้องหรือไม่ ถ้าถูกต้องจะตรวจสอบต่อไปว่า username นี้เป็นพนักงานของฝ่ายใด ซึ่งในบริษัทแบ่งพนักงานออกเป็น 4 ฝ่ายคือ ฝ่ายขาย, ฝ่ายบัญชี, ฝ่ายจัดการสินค้า หรือ ฝ่ายดูแลระบบ ทางโปรแกรมจะทำการแสดงหน้าจอหลักของฝ่ายนั้นขึ้นมา ในส่วนของการทำงานจะอธิบายขอบเขตงานของแต่ละฝ่ายดังนี้

ฝ่ายขาย

ฝ่ายขายจะทำงานในส่วนที่เกี่ยวข้องกับลูกค้า คือ การสมัครเป็นลูกค้าใหม่, การปรับปรุงข้อมูลลูกค้า, การลบข้อมูลลูกค้ารายนั้น, การแสดงรายงานรายละเอียดลูกค้าทั้งหมด และการสั่งซื้อสินค้า โดยเมื่อลูกค้าต้องการจะทำการติดต่อสั่งซื้อสินค้า จะติดต่อผ่านทางฝ่ายขาย โดยอาจจะ แฟกซ์ , อีเมลล์ หรือ โทรศัพท์ มายังบริษัท จากนั้นฝ่ายขายจะทำการกรอกข้อมูล และ โปรแกรมจะทำการตรวจสอบดูว่ามีสินค้าเพียงพอต่อการจัดส่งหรือไม่ กรณีที่มีสินค้าไม่เพียงพอ แม้จะเป็นเพียงชนิดเดียว ก็จะถูกเลิกใบสั่งซื้อสินค้านั้น ลูกค้าจะต้องทำการเปิดใบสั่งซื้อสินค้าใบใหม่มายังบริษัท เช่น ลูกค้าคนที่ หนึ่ง ต้องการซื้อสินค้าชนิด A , ชนิด B และชนิด C ในจำนวน 10, 15, 20 ตามลำดับ แต่เมื่อพนักงานกรอกข้อมูลเพื่อตรวจสอบดูแล้ว ปรากฏว่า มีสินค้า ชนิด C ไม่เพียงพอ คือมีอยู่เพียง 8 ชิ้น ก็จะมีการแจ้ง กลับไปยังลูกค้าว่ามีสินค้าไม่เพียงพอ และยกเลิกใบสั่งซื้อสินค้าใบนั้นไป แต่ถ้าใบสั่งซื้อสินค้าที่ลูกค้าส่งมาให้กับบริษัทนั้น มีจำนวนสินค้าเพียงพอทุกชนิด ทางพนักงานก็จะทำการบันทึกข้อมูลใบสั่งซื้อสินค้าใบนั้นลงในฐานข้อมูล ในส่วนของใบสั่งซื้อสินค้าจะมีสถานะได้ 4 สถานะ คือ ใบสั่งซื้อสินค้าที่รับมาแล้วแต่ยังไม่ได้ทำและการออกไป Invoice, ใบสั่งซื้อสินค้าที่ออกไป Invoice แล้วแต่ยังไม่มีการชำระเงิน, ใบสั่งซื้อสินค้าที่ออกไป Invoice แล้ว

และลูกค้าทำการชำระเงินแล้ว และใบสั่งซื้อสินค้าที่ถูกยกเลิก(อาจถูกยกเลิกเนื่องจาก มีสินค้าไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพียงพอดตามรายการในใบสั่งซื้อสินค้านั้น หรือ มีการขอให้ยกเลิกจากลูกค้าเอง) โดยทางบริษัทจะถือว่า 2 ใบแรกเป็นใบสั่งซื้อสินค้าที่ยังทำไม่สำเร็จ แต่ 2 แบบหลังเป็นแบบที่ทำสำเร็จแล้ว

ฝ่ายบัญชี

ฝ่ายบัญชีจะทำหน้าที่ในการแสดงรายงาน, การออกใบ Invoice และการตรวจสอบใบ Invoice ที่ยังไม่ได้ชำระเงินแต่ใกล้ถึงวันที่บริษัทกำหนดให้ชำระเงินแล้ว(Check Due) ในส่วนของการแสดงรายงาน ประกอบด้วย การแสดงรายงานใบสั่งซื้อสินค้าทั้งหมด, การแสดงรายงานใบสั่งซื้อสินค้าที่ยังอยู่ระหว่างการดำเนินการ, การแสดงรายงานรายละเอียดของใบสั่งซื้อสินค้า, การแสดงรายงานรายละเอียดของใบ Invoice, การแสดงรายงานจำนวนใบสั่งซื้อสินค้าที่ทำสำเร็จ และไม่สำเร็จ, การแสดงรายงานใบ Invoice ที่ทำสำเร็จและไม่สำเร็จ และ ในส่วนการทำการออกใบ Invoice จะทำเมื่อฝ่ายบัญชีเข้าไปตรวจสอบว่ามีใบสั่งซื้อสินค้าใบบ้างที่ต้องออกใบ Invoice ก็จะทำการออกใบ Invoice ใบนั้น โดยจัดทำสำเนาเอาไว้ 2 ชุด จากนั้นจะส่งทั้ง 3 ใบไปให้ฝ่ายจัดการสินค้า นอกจากนี้ในส่วนของสถานะของใบ Invoice ทางบริษัทได้แบ่งไว้ 2 สถานะคือ ใบ Invoice ที่ยังไม่ได้ชำระค่าสินค้า และใบ Invoice ที่ชำระค่าสินค้าแล้ว ในส่วนสุดท้ายคือการตรวจใบ Invoice ที่ยังไม่ได้ทำการชำระค่าสินค้า และใกล้ถึงวันที่บริษัทกำหนดให้ชำระเงินแล้ว (ทางบริษัทกำหนดให้ค้างการชำระเงินตามใบ Invoice แต่ละใบได้ไม่เกิน ที่กำหนดในช่วง Check Due ซึ่งลูกค้าแต่ละคนอาจจะแตกต่างกันไป) พนักงานฝ่ายบัญชีจะสามารถเลือกได้ว่าจะต้องส่งใบเรียกเก็บเงินของใบ Invoice ใบบ้าง ซึ่งระบบจะทำการคำนวณ และแสดง ให้เองว่ามีใบ Invoice ใบใบบ้างที่อยู่ในกรณีนี้

ฝ่ายจัดการสินค้า

ฝ่ายจัดการสินค้าจะทำหน้าที่เกี่ยวกับสินค้าคือ การแสดงรายงานรายละเอียดของสินค้าทั้งหมด, การเพิ่มรายการสินค้าใหม่, การเพิ่มจำนวนสินค้าที่มีอยู่แล้ว และการจัดส่งสินค้าตามใบ Invoice ที่ได้รับมาจากฝ่ายบัญชี โดยเมื่อนำสินค้าไปส่งยังลูกค้าก็จะให้ลูกค้าลงชื่อในใบ Invoice และสำเนาอีก 2 ใบ จากนั้นจึงจะให้สำเนาใบ Invoice กับลูกค้าไป 1 ใบ เพื่อเป็นการยืนยันว่าได้ทำการส่งสินค้าเรียบร้อยแล้ว และถ้าลูกค้าชำระเงินทันทีก็จะให้ใบ Invoice ฉบับจริงกับลูกค้าเพื่อเป็นใบเสร็จรับเงิน จากนั้นพนักงานส่งสินค้าจะนำสำเนาใบ Invoice อีกใบกลับมายังบริษัทเพื่อให้ฝ่ายบัญชีเก็บไว้เป็นหลักฐานในการส่งสินค้า และรับสินค้า แต่ถ้าลูกค้ายังไม่ทำการชำระค่าสินค้าในทันทีพนักงานส่งสินค้าก็จะ นำใบ Invoice ฉบับจริงและสำเนาใบ Invoice กลับมาให้กับทางฝ่ายบัญชี

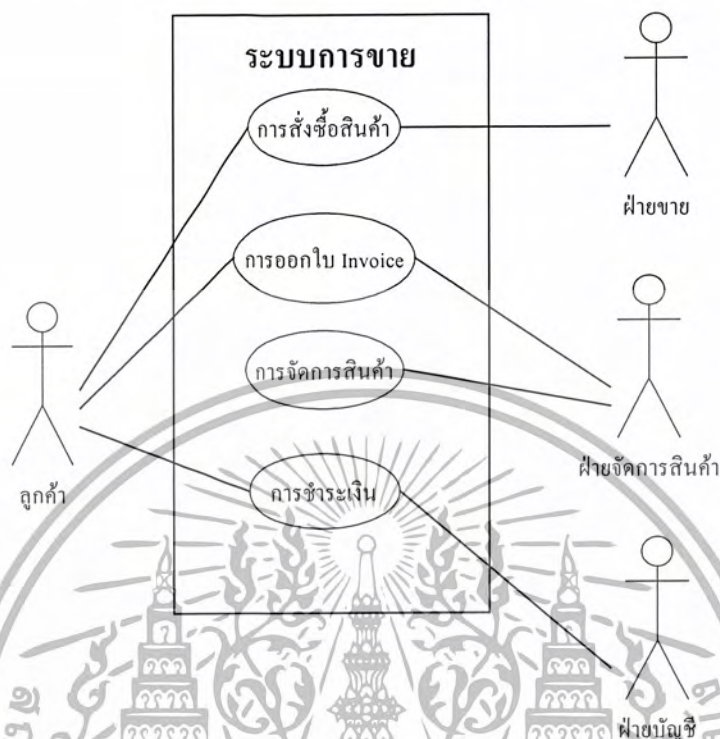
ฝ่ายดูแลระบบ

ผู้ดูแลระบบจะมีหน้าที่ในการจัดการข้อมูลพนักงาน เพิ่มข้อมูล เปลี่ยนแปลงข้อมูล และลบข้อมูลพนักงาน โดยระบบจะทำการสร้างรหัสประจำตัวพนักงาน และ Password ของพนักงานแต่ละคนให้โดยอัตโนมัติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การชำระเงิน เป็นพฤติกรรมที่เกี่ยวข้องกับลูกค้าและฝ่ายบัญชี เมื่อลูกค้าได้รับสินค้าครบถ้วนแล้ว ก็จะต้องทำการชำระเงินค่าสินค้าที่ตนสั่ง



รูปที่ 3.1 แสดง Use Case Diagram หลักของระบบการจำหน่ายสินค้า

3.1.2 พฤติกรรมย่อยที่เกี่ยวข้องกับพฤติกรรมหลัก

พฤติกรรมหลักยังมีพฤติกรรมย่อยอีก ดังนี้

3.1.2.1 การสั่งซื้อสินค้า

- สมัครสมาชิก
- รับใบสั่งซื้อสินค้า
- ตรวจสอบว่ามีสินค้าเพียงพอหรือไม่
- ยกเลิกในกรณีมีสินค้าไม่เพียงพอ
- บันทึกใบสั่งซื้อสินค้ากรณีมีสินค้าพอ
- แก้ไขข้อมูลลูกค้า
- รายงานรายละเอียดลูกค้า

บทที่ 3

การออกแบบระบบการจำหน่ายสินค้า

ผู้ใช้งานระบบจะสามารถนำ Framework ของโปรแกรมระบบการจำหน่ายสินค้าไปใช้ได้โดยเลือกเพียงหนึ่ง Component แล้วนำไปใช้ร่วมกับระบบเดิมของตนเอง หรือสามารถนำไปใช้หลาย Component ก็สามรถทำได้ โดย ระบบการจำหน่ายสินค้านี้ได้ถูกออกแบบให้พนักงานในระบบแบ่งออกเป็น 4 ฝ่ายคือ ฝ่ายขาย ฝ่ายจัดการสินค้า ฝ่ายบัญชี และฝ่ายผู้ดูแลระบบ โดยแต่ละฝ่ายจะมีขอบเขตการทำงานที่แบ่งแยกกัน พนักงานสามารถผ่านเข้าใช้ระบบในฝ่ายของตนเองได้โดย การ Login และใส่ username และpassword ของตนเองเท่านั้น จึงจะสามารถทำกิจกรรมหลักในฝ่ายของตนได้ เมื่อพนักงานกระทำการใดก็ตามที่เกี่ยวกับการประมวลผล ระบบจะส่งค่า parameter ไปยังเครื่องเซิร์ฟเวอร์โดยใช้ โปรโตคอล SOAP จากนั้นเครื่องเซิร์ฟเวอร์จะทำการนำค่า parameter ที่ได้มาทำการประมวลผล เมื่อได้คำตอบก็จะส่งผลลัพธ์นั้นกลับไปยังเครื่องไคลเอนท์ ที่ส่งค่ามาตามกระบวนการการทำงานของ Web Service และเพื่อให้ผู้ใช้บริการสามารถเลือกใช้บริการ บริการใดบริการหนึ่ง หรือ บริการทั้งหมดได้จึงจำเป็นต้องแบ่ง โปรแกรมออก Component และจัด Method ในแต่ละ Class ให้เหมาะสมดังจะ ได้กล่าวต่อไป

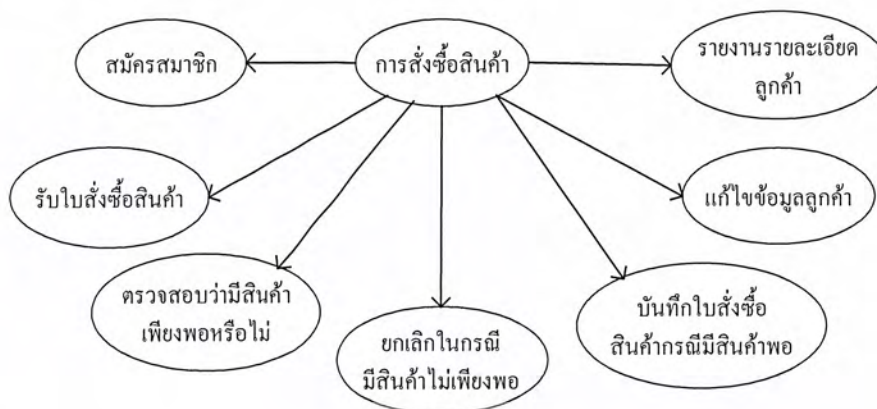
3.1 Use Case Diagram ของระบบการจำหน่ายสินค้า

3.1.1 พฤติกรรมหลักของระบบการจำหน่ายสินค้า

พฤติกรรมในระบบการจำหน่ายสินค้าจะแบ่งเป็นพฤติกรรมใหญ่ๆ ได้ 4 พฤติกรรม ดังนี้ การสั่งซื้อสินค้า เป็นพฤติกรรมที่เกี่ยวข้องกับลูกค้าและฝ่ายขาย พฤติกรรมนี้หลังจากลูกค้าติดต่อมายังฝ่ายขายแล้ว ฝ่ายขายจะทำการตรวจสอบก่อนว่ามีสินค้าเพียงพอหรือไม่ ถ้ามีสินค้าเพียงพอครบทุกรายการจึงจะสามารถสั่งซื้อสินค้าได้

การออกใบ Invoice เป็นพฤติกรรมที่เกี่ยวข้องกับลูกค้าและฝ่ายขาย พฤติกรรมนี้จะออกใบ Invoice เมื่อมีใบสั่งซื้อสินค้าในฐานะข้อมูล

การจัดการสินค้า เป็นพฤติกรรมที่เกี่ยวข้องกับฝ่ายขาย พฤติกรรมนี้ฝ่ายขายจะดำเนินการจัดการกับสินค้า เช่น การส่งสินค้า การตัดจำนวนสินค้าในคลังสินค้า การเพิ่มรายการสินค้า เป็นต้น



รูปที่ 3.2 แสดง Use Case Diagram การสั่งซื้อสินค้า

3.1.2.2 การออกไป Invoice

- ออกใบ Invoice จำนวน 3 ชุด
- ตรวจสอบว่ามีใบสั่งซื้อสินค้าใดบ้าง ที่ยังไม่ออกไป Invoice

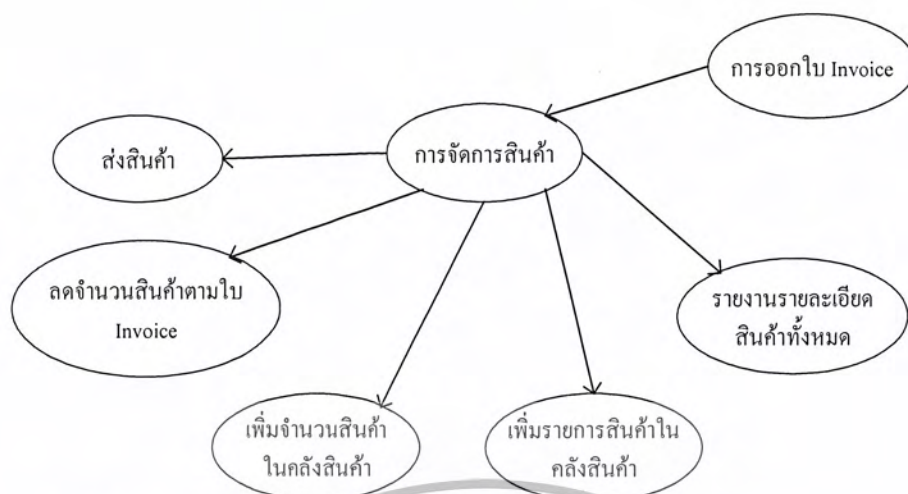


รูปที่ 3.3 แสดง Use Case Diagram การออกไป Invoice

3.1.2.3 การจัดการสินค้า

- ส่งสินค้าให้ลูกค้า
- ลดจำนวนสินค้าตามใบ Invoice
- เพิ่มจำนวนสินค้าในคลังสินค้า
- เพิ่มรายการสินค้าในคลังสินค้า
- รายงานรายละเอียดสินค้าทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดง Use Case Diagram การจัดการสินค้า

3.1.2.4 การชำระเงิน

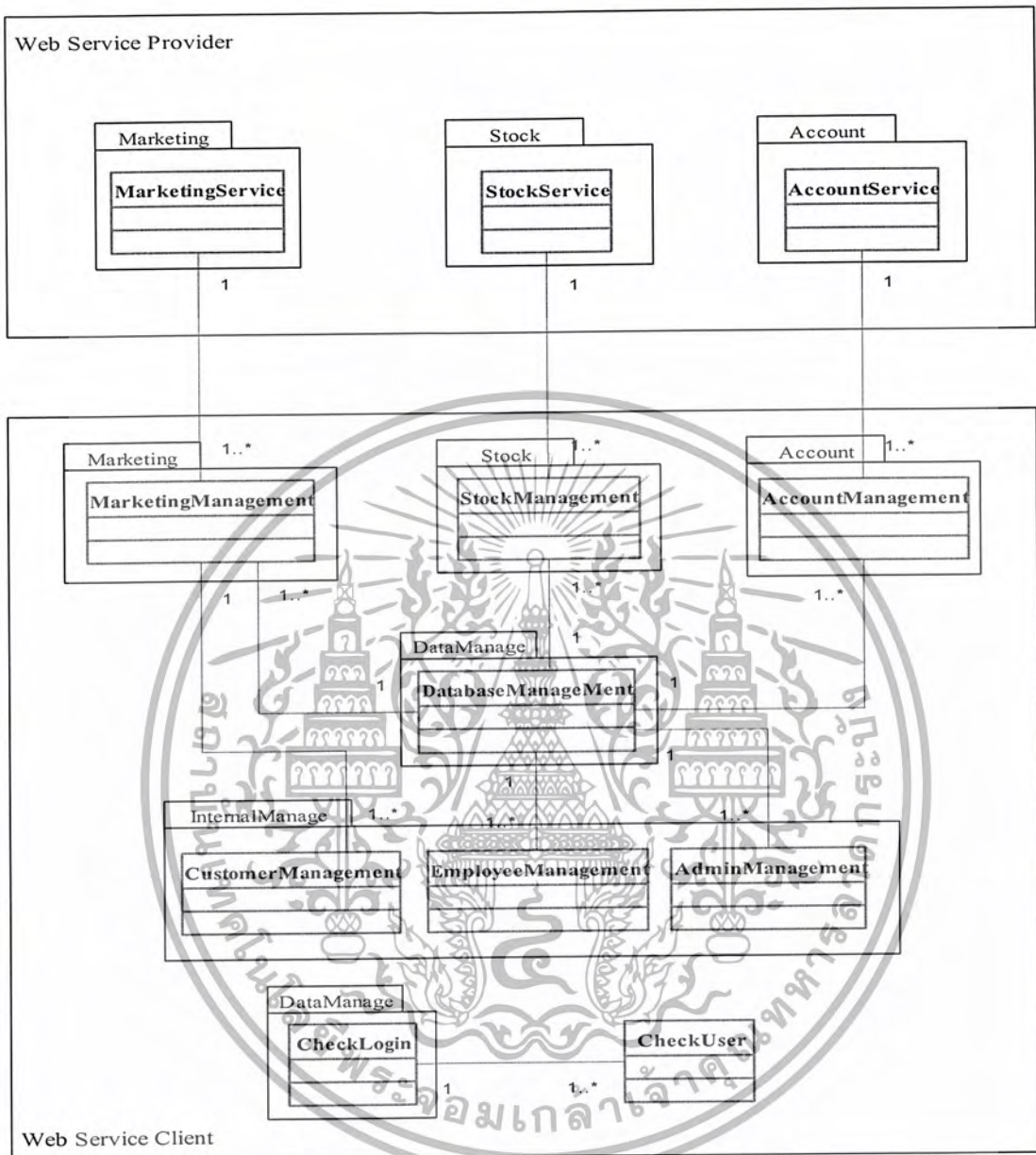
- การตรวจหาใบ Invoice ที่ใกล้ถึงเวลาชำระค่าสินค้า
- ตรวจหาใบ Invoice ที่ใกล้ถึงเวลาชำระค่าสินค้า
- เรียกเก็บเงินจากลูกค้า
- ให้ใบเสร็จลูกค้า



รูปที่ 3.5 แสดง Use Case Diagram การชำระเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบ Framework สำหรับระบบการจำหน่ายสินค้า

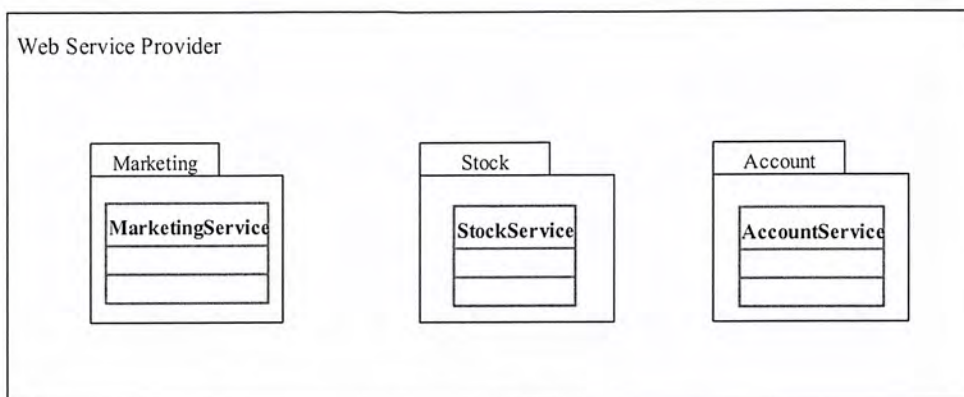


รูปที่ 3.6 แผนภาพการแสดงผล Framework ของระบบการจำหน่ายสินค้า

จากรูปที่ 3.6 เป็น Diagram ภาพรวมของระบบเท่านั้น ดังนั้นจะขออธิบายรายละเอียดของ Diagram ต่างๆ ของระบบซึ่งจะแยกเป็น 2 ส่วนใหญ่ๆ คือ ฟังก์ชันให้บริการ Web Service และฟังก์ชันของผู้ใช้บริการ Web Service ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 แผนภาพแสดง Class ต่างๆในส่วนของผู้ให้บริการ Web Service



รูปที่ 3.7 Diagram แสดง Class ในส่วนของผู้ให้บริการ

จากรูปที่ 3.7 เป็น Diagram ที่แสดง Package และ Class ที่อยู่ทางฝั่งผู้ให้บริการ Web Service ซึ่งเป็นผู้ให้บริการ Class ที่ทำการสร้าง String ที่ใช้ในการดึงข้อมูลหรือทำการใส่ข้อมูลลงฐานข้อมูลในระบบขายสินค้า

3.2.1.1 รายละเอียดส่วนประกอบภายใน Class MarketingService

เป็น Class ที่จัดการเกี่ยวกับการสร้างในคลังสินค้าที่ถูกคำสั่ง สามารถทำการสร้างใบสั่งสินค้าใหม่ และสามารถทำการยกเลิกใบสั่งสินค้า

MarketingService	
-SQLCommand	: String
+MarketingService()	
+addNewOrder(in customerOrderNumber : String, in employeeld : String, in customerId : String, in sendDate : String) : String	
+addOrderDetail(in customerOrderNumber : String, in productId : String, in orderQuantity : String) : String	
+listAllNewCustomerOrder() : String	
+listCustomerOrderDetail(in customerOrderNumber : String) : String	
+findCustomerOrder(in customerOrderNumber : String) : String	
+cancelCustomerOrder(in customerOrderNumber : String) : String	

รูปที่ 3.8 ภาพแสดงรายละเอียดภายใน Class MarketingService

ตารางที่ 3.1 Attribute ภายใน Class MarketingService

ชื่อ	หน้าที่
SQLCommand	เป็น Attribute ที่เก็บค่า String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 Method ภายใน Class MarketingService

ชื่อ	หน้าที่
MarketingService()	เป็น constructor method ของ Class
addNewOrder(String customerOrderNumber, String employeeId, String customerId, String sendDate)	ทำการสร้าง String เพิ่มข้อมูลใบสั่งซื้อสินค้า
addOrderDetail(String customerOrderNumber, String productId, String orderQuantity)	ทำการสร้าง String เพิ่มข้อมูลสินค้าของใบสั่งซื้อสินค้า
listAllNewCustomerOrder()	ทำการสร้าง String ที่ใช้ดึงข้อมูลใบสั่งที่ยังไม่ได้สร้าง Invoice
listCustomerOrderDetail(String customerOrderNumber)	ทำการสร้าง String ที่ใช้ดึงข้อมูลรายละเอียดของใบสั่งซื้อสินค้า
findCustomerOrder(String customerOrderNumber)	ทำการสร้าง String ที่ใช้ดึงข้อมูลใบสั่งซื้อสินค้าที่ผู้ใช้ต้องการ
cancelCustomerOrder(String customerOrderNumber)	ทำการสร้าง String ที่ใช้ยกเลิกใบสั่งซื้อสินค้า

3.2.1.2 รายละเอียดส่วนประกอบภายใน Class StockService

เป็น Class ที่จัดการเกี่ยวกับสินค้าในคลังสินค้าของระบบ สามารถทำการเพิ่มชนิดสินค้า เพิ่มจำนวนสินค้าในคลัง และสามารถเปลี่ยนราคาของสินค้า

StockService
-SQLCommand : String
+StockService()
+listAllProduct() : String
+searchProduct(in productId : String) : String
+changeProductPrice(in productId : String, in newPrice : String) : String
+addProductStock(in employeeId : String, in productId : String, in addQuantity : String) : String
+addNewProduct(in employeeId : String, in productId : String, in productName : String, in productDetail : String, in productQuantity : String, in productPrice : String) : String

รูปที่ 3.9 ภาพแสดงรายละเอียดภายใน Class StockService

ตารางที่ 3.3 Attribute ภายใน Class StockService

ชื่อ	หน้าที่
SQLCommand	เป็น Attribute ที่เก็บค่า String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class

ตารางที่ 3.4 Method ภายใน Class StockService

ชื่อ	หน้าที่
StockService()	เป็น constructor method ของ Class
listAllProduct()	ทำการสร้าง String ที่ใช้ดึงข้อมูลของสินค้าทั้งหมด
searchProduct(String productId)	ทำการสร้าง String ที่ใช้ดึงข้อมูลของสินค้าที่ผู้ใช้ต้องการ
changeProductPrice(String productId, String newPrice)	ทำการสร้าง String ที่ใช้เปลี่ยนข้อมูลราคาของสินค้า 1 ชนิดที่ผู้ใช้ต้องการ
addProductStock(String employeeId, String productId, String addQuantity)	ทำการสร้าง String เพิ่มจำนวนสินค้าที่ผู้ใช้ต้องการ 1 ชนิด
addNewProduct(String employeeId, String productId, String productName, String productDetail, String productQuantity, String productPrice)	ทำการสร้าง String เพิ่มข้อมูลสินค้าชนิดใหม่

3.2.1.3 รายละเอียดส่วนประกอบภายใน Class AccountService

เป็น Class ที่จัดการเกี่ยวกับการออกใบ Invoice เปลี่ยน status ของใบ Invoice และสร้างใบ

วางบิล

AccountService
-SqlCommand : String
+AccountService()
+createInvoice(in invoiceNumber : String, in customerOrderNumber : String, in employeeId : String, in invoiceTotal : String, in invoiceTotalQuantity : String) : String
+addInvoiceDetail(in invoiceNumber, in productId, in orderQuantity) : String
+changeInvoiceStatus(in invoiceNumber : String) : String
+createNewBill(in billNumber : String, in employeeId : String, in billTotalPrice : String) : String
+addBillDetail(in billNumber : String, in invoiceNumber : String, in invoicePrice : String, in customerOrderNumber : String) : String
+reportAllProduct() : String
+reportAllCustomerOrder() : String
+reportAllInvoice() : String

รูปที่ 3.10 ภาพแสดงรายละเอียดภายใน Class AccountService

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 Attribute ภายใน Class AccountService

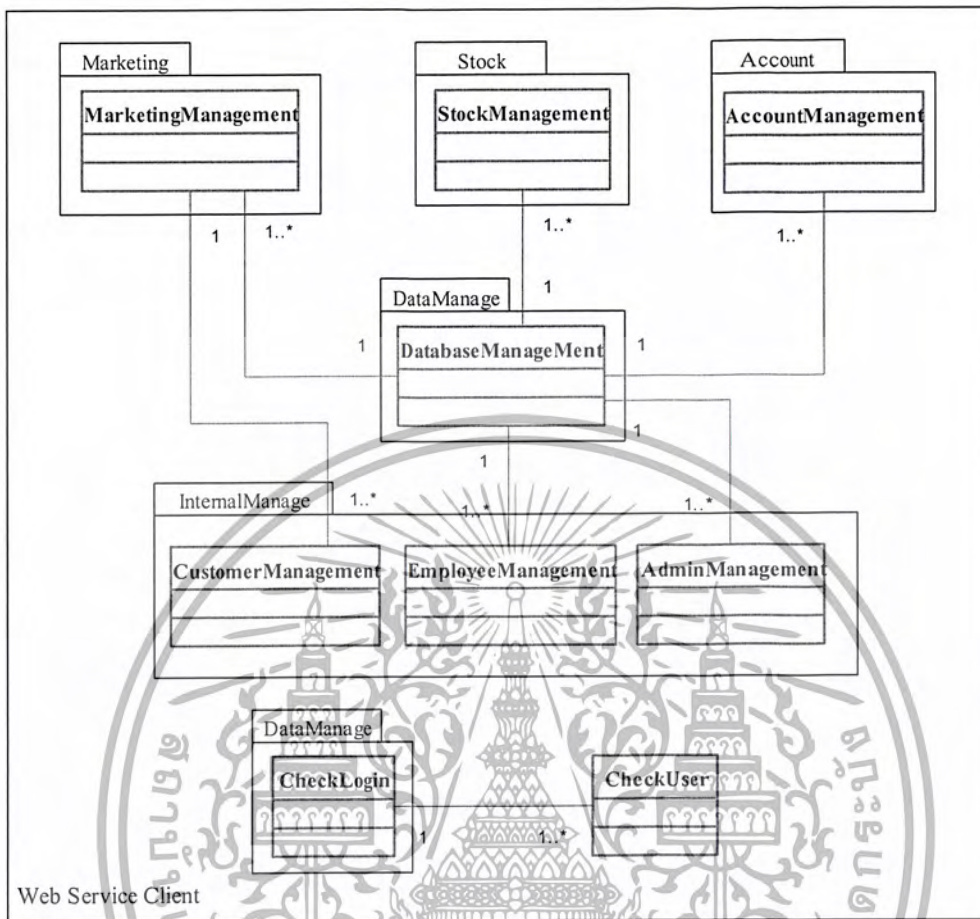
ชื่อ	หน้าที่
SQLCommand	เป็น Attribute ที่เก็บค่า String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class

ตารางที่ 3.6 Method ภายใน Class AccountService

ชื่อ	หน้าที่
AccountService()	เป็น constructor method ของ Class
createinvoice(String invoiceNumber, String customerOrderNumber, String employeeId, String invoiceTotal, String InvoiceTotalQuantity)	ทำการสร้าง String เพิ่มข้อมูลใบ Invoice
addinvoiceDetail(String invoiceNumber, String productId, String orderQuantity)	ทำการสร้าง String เพิ่มข้อมูลสินค้าของใบ Invoice
changeInvoiceStatus(String invoiceNumber)	ทำการสร้าง String เปลี่ยนสถานะของใบ Invoice ให้เป็นสถานะเสร็จสมบูรณ์
createNewBill(String billNumber, String employeeId, String billTotalPrice)	ทำการสร้าง String เพิ่มข้อมูลใบวางบิล
addBillDetail(String billNumber, String invoiceNumber, String invoicePrice, String customerOrderNumber)	ทำการสร้าง String เพิ่มรายละเอียดของใบวางบิล
reportAllProduct()	ทำการสร้าง String ที่ใช้ดึงข้อมูลสินค้าทั้งหมด
reportAllCustomerOrder()	ทำการสร้าง String ที่ใช้ดึงข้อมูลใบสั่งซื้อสินค้าทั้งหมด
reportAllInvoice()	ทำการสร้าง String ที่ใช้ดึงข้อมูลใบ Invoice ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 แผนภาพแสดง Class ต่างๆในส่วนของผู้ใช้บริการ Web Service



รูปที่ 3.11 Diagram แสดง Class ในส่วนของผู้ขอใช้บริการ

จากรูปที่ 3.11 เป็น Diagram ที่แสดง Package และ Class ที่อยู่ฝั่งผู้ใช้บริการ Web Service ซึ่งใน Diagram นี้จะมีหน้าที่ทำการติดต่อกับ Class ที่อยู่ทางฝั่งผู้ให้บริการ Web Service ติดต่อกับฐานข้อมูลและจัดการข้อมูลของบุคคลที่เกี่ยวข้องกับระบบ เช่น ข้อมูลพนักงาน ข้อมูลลูกค้า รวมทั้งการตรวจสอบสิทธิ์ของผู้ใช้งานในระบบ

3.2.2.1 รายละเอียดส่วนประกอบภายใน Class MarketingManagement

เป็น Class ที่จัดการเกี่ยวกับการสร้างใบสั่งสินค้า ยกเลิกใบสั่งสินค้า และจัดการข้อมูลของลูกค้า

MarketingManagement
-SQLCommand : String -customerOrderNumber : String -m_hostURL : String -DBManage : DataManage.DatabaseManagement -call : org.apache.soap.rpc.Call -resp : org.apache.soap.rpc.Response -CustomerManage : InternalManage.CustomerManagement
+MarketingManagement() +createNewCustomerOrderNumber() +queryAllProduct() : Object +queryAllNewCustomerOrder() : Object +queryCustomerOrderDetail(in customerOrderNumber : String) : Object +queryAllCustomer() : Object +queryCustomerDetail(in cusId : String) : Object +insertNewOrder(in empld : String, in cusId : String, in sendDate : String) : Integer +insertNewOrderDetail(in productId : String, in quantity : String) +insertNewCustomer(in cusId : String, in cusName : String, in cusLastName : String, in cusAddress : String, in cusTel : String, in cusFax : String, in due : String) : Integer +findStockQuantity(in productId : String) : Integer +findCustomerDetailFromCustomerOrderNumber(in customerOrderNumber : String) : Object +cancelOrder(in customerOrderNumber : String) : Integer +deleteCustomerDetail(in cusId : String) : Integer +updateCustomerDetail(in cusId : String, in cusName : String, in cusLastName : String, in cusAddress : String, in cusTel : String, in cusFax : String) : Integer

รูปที่ 3.12 ภาพแสดงรายละเอียดภายใน Class MarketingManagement

ตารางที่ 3.7 Attribute ภายใน Class MarketingManagement

ชื่อ	หน้าที่
SQLCommand	เป็น Attribute ที่ใช้เก็บ String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class หรือรับมาจาก method ผู้ให้บริการ Web Service
customerOrderNumber	เป็น Attribute ที่ใช้เก็บหมายเลขใบสั่งสินค้า
m_hostURL	เป็น Attribute ที่ใช้เก็บ host ที่ใช้ทำการติดต่อไปยังผู้ให้บริการ Web Service
DBManage	เป็นออบเจกต์ที่ใช้จัดการกับฐานข้อมูล
call	เป็นออบเจกต์ที่ใช้สำหรับติดต่อกับผู้ให้บริการ Web Service
resp	เป็นออบเจกต์ที่ใช้สำหรับรับผลจากออบเจกต์ call
CustomerManage	เป็นออบเจกต์ที่ใช้จัดการกับข้อมูลลูกค้า

ตารางที่ 3.8 Method ภายใน Class MarketingManagement

ชื่อ	หน้าที่
MarketingManagement()	เป็น constructor method ของ Class
createNewCustomerOrderNumber()	ทำการสร้างหมายเลขใบสั่งสินค้าสำหรับใบสั่งสินค้าใบใหม่
queryAllProduct()	ทำการดึงข้อมูลสินค้าทั้งหมดจากฐานข้อมูล
queryAllNewCustomerOrder()	ทำการดึงข้อมูลใบสั่งสินค้าทั้งหมดจากฐานข้อมูล
queryCustomerOrderDetail(String customerOrderNumber)	ทำการดึงข้อมูลรายละเอียดของใบสั่งสินค้าจากฐานข้อมูล
queryAllCustomer()	ทำการดึงข้อมูลลูกค้าทั้งหมดจากฐานข้อมูล
queryCustomerDetail(String cusId)	ทำการดึงข้อมูลรายละเอียดของลูกค้าจากฐานข้อมูล
insertNewOrder(String empId, String cusId, String sendDate)	ทำการเพิ่มข้อมูลใบสั่งสินค้าใบใหม่เข้าไปในฐานข้อมูล
insertNewOrderDetail(String productId, String quantity)	ทำการเพิ่มข้อมูลรายละเอียดสินค้าเข้าไปในฐานข้อมูล
insertNewCustomer(String cusId, String cusName, String cusLastName, String cusAddress, String cusTel, String cusFax, String due)	ทำการเพิ่มข้อมูลลูกค้าใหม่เข้าไปในฐานข้อมูล
findStockQuantity(String productId)	ทำการค้นหาจำนวนสินค้าที่สามารถให้ลูกค้าสั่งซื้อได้จากฐานข้อมูล
findCustomerDetailFromCustomerOrderNumber (String customerOrderNumber)	ทำการค้นหาข้อมูลลูกค้าจากหมายเลขใบสั่งสินค้า
cancelOrder(String customerOrderNumber)	ทำการยกเลิกใบสั่งสินค้า
deleteCustomerDetail(String cusId)	ทำการลบข้อมูลลูกค้าออกจากฐานข้อมูล
updateCustomerDetail(String cusId, String cusName, String cusLastName, String cusAddress, String cusTel, String cusFax)	ทำการเปลี่ยนแปลงข้อมูลของลูกค้าในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 รายละเอียดส่วนประกอบภายใน Class StockManagement

เป็น Class ที่จัดการเกี่ยวกับข้อมูลสินค้าในคลังสินค้า

StockManagement
-SQLCommand : String
-m_hostURL : String
-DBManage : DataManage.DatabaseManagement
-call : org.apache.soap.rpc.Call
-resp : org.apache.soap.rpc.Response
+StockManagement()
+queryProductDetail(in productId : String) : Object
+queryAllProduct() : Object
+addStock(in empId : String, in productId : String, in quantity : String) : Integer
+addNewProduct(in empId : String, in productId : String, in productName : String, in productDetail : String, in productPrice : String, in quantity : String) : Integer
+changeProductPrice(in productId : String, in productPrice : String) : Integer

รูปที่ 3.13 ภาพแสดงรายละเอียดภายใน Class StockManagement

ตารางที่ 3.9 Attribute ภายใน Class StockManagement

ชื่อ	หน้าที่
SQLCommand	เป็น Attribute ที่ใช้เก็บ String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class หรือรับมาจาก method ผู้ให้บริการ Web Service
m_hostURL	เป็น Attribute ที่ใช้เก็บ host ที่ใช้ในการติดต่อไปยังผู้ให้บริการ Web Service
DBManage	เป็นออบเจกต์ที่ใช้จัดการกับฐานข้อมูล
call	เป็นออบเจกต์ที่ใช้สำหรับติดต่อกับผู้ให้บริการ Web Service
resp	เป็นออบเจกต์ที่ใช้สำหรับรับผลจากออบเจกต์ call

ตารางที่ 3.10 Method ภายใน Class StockManagement

ชื่อ	หน้าที่
StockManagement()	เป็น constructor method ของ Class
queryProductDetail(String productId)	ทำการดึงรายละเอียดสินค้าจากฐานข้อมูล
queryAllProduct()	ทำการดึงข้อมูลสินค้าทั้งหมดจากฐานข้อมูล
addStock(String empId, String productId, String quantity)	ทำการเพิ่มจำนวนสินค้า 1 ชนิดลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

addNewProduct(String empId, String productId, String productName, String productDetail, String productPrice, String quantity)	ทำการเพิ่มสินค้าใหม่เข้าไปในฐานข้อมูล
changeProductPrice(String productId, String productPrice)	ทำการเปลี่ยนแปลงราคาสินค้าในฐานข้อมูล

3.2.2.3 รายละเอียดส่วนประกอบภายใน Class AccountManagement

เป็น Class ที่จัดการเกี่ยวกับข้อมูลใบ Invoice ใบวางบิล และการออกรายงานต่างๆ

AccountManagement
-SQLCommand : String -billNumber : String -invoiceNumber : String -m_hostURL : String -DBManage : DataManage.DatabaseManagement -call : org.apache.soap.rpc.Call -resp : org.apache.soap.rpc.Response
+AccountManagement() +queryAllNewCustomerOrder() : Object +queryCustomerDetailByCustomerOrderNumber(in customerOrderNumber : String) : Object +queryCustomerOrderDetail(in customerOrderNumber : String) : Object +findProduct(in productId : String) : Object -createNewInvoiceNumber() +createInvoice(in customerOrderNumber : String, in empId : String, in invoiceTotalPrice : String, in invoiceTotalQuantity : String) : Integer +addInvoiceDetail(in productId : String, in orderQuantity : String) +updateInvoiceStatus(in invoiceNumber : String) : Integer +queryAllCustomer() : Object +queryAllNotCompleteCustomerOrderByCustomerId(in cusId : String) : Object +queryInvoice(in invoiceNumber : String) : Object +queryInvoiceByCustomerOrderNumber(in customerOrderNumber : String) : Object +checkDue(in cusId : String, in invoiceNumber : String) : String -createNewBillNumber() +createNewBill(in empId : String, in billTotalPrice : String) : Integer +addBillDetail(in invoiceNumber : String, in invoicePrice : String, in customerOrderNumber : String) +reportAllProduct() : Object +reportAllCustomerOrder() : Object +reportAllInvoice() : Object

รูปที่ 3.14 ภาพแสดงรายละเอียดภายใน Class AccountManagement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 Attribute ภายใน Class AccountManagement

ชื่อ	หน้าที่
SQLCommand	เป็น Attribute ที่ใช้เก็บ String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class หรือรับมาจาก method ผู้ให้บริการ Web Service
billNumber	เป็น Attribute ที่ใช้เก็บหมายเลขใบวางบิล
invoiceNumber	เป็น Attribute ที่ใช้เก็บหมายเลขใบ Invoice
m_hostURL	เป็น Attribute ที่ใช้เก็บ host ที่ใช้ทำการติดต่อไปยังผู้ให้บริการ Web Service
DBManage	เป็นออบเจกต์ที่ใช้จัดการกับฐานข้อมูล
call	เป็นออบเจกต์ที่ใช้สำหรับติดต่อกับผู้ให้บริการ Web Service
resp	เป็นออบเจกต์ที่ใช้สำหรับรับผลจากออบเจกต์ call

ตารางที่ 3.12 Method ภายใน Class AccountManagement

ชื่อ	หน้าที่
AccountManagement()	เป็น constructor method ของ Class
queryAllNewCustomerOrder()	ทำการดึงข้อมูลใบสั่งซื้อสินค้าทั้งหมดที่ยังไม่ได้ออกไป Invoice
queryCustomerDetailByCustomerOrderNumber(String customerOrderNumber)	ทำการดึงรายละเอียดของลูกค้าโดยใช้หมายเลขใบสั่งซื้อสินค้าเป็นตัวช่วยค้นหาจากฐานข้อมูล
queryCustomerOrderDetail(String customerOrderNumber)	ทำการดึงรายละเอียดของใบสั่งซื้อสินค้าจากฐานข้อมูล
findProduct (String productId)	ทำการดึงข้อมูลสินค้าที่ต้องการจากฐานข้อมูล
createNewInvoiceNumber()	ทำการสร้างหมายเลขใบ Invoice สำหรับใบ Invoice ใบใหม่
createInvoice(String customerOrderNumber, String empId, String invoiceTotalPrice, String invoiceTotalQuantity)	ทำการเพิ่มข้อมูลใบ Invoice เข้าไปในฐานข้อมูล
addInvoiceDetail(String productId, String orderQuantity)	ทำการเพิ่มข้อมูลรายละเอียดของใบ Invoice เข้าไปในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

updateInvoiceStatus(String invoiceNumber)	ทำการเปลี่ยนแปลงสถานะของใบ Invoice ในฐานข้อมูล
queryAllCustomer()	ทำการดึงข้อมูลลูกค้าทั้งหมดจากฐานข้อมูล
queryAllNotCompleteCustomerOrderByCustomerId(String cusId)	ทำการดึงข้อมูลใบสั่งซื้อสินค้าทั้งหมดที่สร้างใบ Invoice แล้วแต่ยังไม่ได้ชำระเงิน
queryInvoice(String invoiceNumber)	ทำการดึงข้อมูลใบ Invoice จากฐานข้อมูล
queryInvoiceByCustomerOrderNumber(String customerOrderNumber)	ทำการดึงข้อมูลใบ Invoice โดยใช้หมายเลขใบสั่งซื้อสินค้าเป็นตัวค้นหา
checkDue(String cusId, String invoiceNumber)	ทำการตรวจสอบใบ Invoice ว่าถึงกำหนดจ่ายเงินที่ค้างชำระไว้หรือยัง
createNewBillNumber()	ทำการสร้างหมายเลขใบวางบิลสำหรับใบวางบิลใบใหม่
createNewBill(String empId, String billTotalPrice)	ทำการเพิ่มข้อมูลใบวางบิลเข้าไปในฐานข้อมูล
addBillDetail(String invoiceNumber, String invoicePrice, String customerOrderNumber)	ทำการเพิ่มข้อมูลรายละเอียดของใบวางบิลเข้าไปในฐานข้อมูล
reportAllProduct()	ทำการดึงข้อมูลสินค้าทั้งหมดจากฐานข้อมูล
reportAllCustomerOrder()	ทำการดึงข้อมูลใบสั่งซื้อสินค้าทั้งหมดจากฐานข้อมูล
reportAllInvoice()	ทำการดึงข้อมูลใบ Invoice ทั้งหมดจากฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.4 รายละเอียดส่วนประกอบภายใน Class DatabaseManagement

เป็น Class ที่จัดการติดต่อกับฐานข้อมูล ดึงข้อมูล ใส่อข้อมูลและเปลี่ยนแปลงข้อมูลในฐานข้อมูล

DatabaseManagement
-con : Connection
-stmt : Statement
-rs : ResultSet
+DatabaseManagement()
+connectDatabase()
+queryData(in SQLCommand : String) : Object
+updateData(in SQLCommand : String) : Integer
+deleteData(in SQLCommand : String) : Integer
+disconnectDatabase()

รูปที่ 3.15 ภาพแสดงรายละเอียดภายใน Class DatabaseManagement

ตารางที่ 3.13 Attribute ภายใน Class DatabaseManagement

ชื่อ	หน้าที่
con	เป็นออบเจกต์ที่ใช้สำหรับติดต่อกับฐานข้อมูล
stmt	เป็นออบเจกต์ที่ใช้สำหรับการจัดการข้อมูลกับฐานข้อมูล
rs	เป็นออบเจกต์ที่รับผลของการดึงข้อมูลจากฐานข้อมูล

ตารางที่ 3.14 Method ภายใน Class DatabaseManagement

ชื่อ	หน้าที่
DatabaseManagement()	เป็น constructor method ของ Class
connectDatabase()	ทำการเชื่อมต่อกับฐานข้อมูลที่ต้องการใช้
queryData(String SQLCommand)	ทำการดึงข้อมูลจากฐานข้อมูล โดยใช้ คำสั่ง Select ของ ชุดคำสั่ง SQL
updateData(String SQLCommand)	ทำการเพิ่มหรือปรับปรุงข้อมูลในฐานข้อมูลที่เราต้องการ
deleteData(String SQLCommand)	ทำการลบข้อมูลในฐานข้อมูลที่เราต้องการ
disconnectDatabase()	ทำการตัดการเชื่อมต่อกับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.5 รายละเอียดส่วนประกอบภายใน Class CustomerManagement

เป็น Class ที่ทำการสร้างStringคำสั่งที่เกี่ยวกับการจัดการข้อมูลของลูกค้า

CustomerManagement
-SQLCommand : String
+CustomerManagement() +listAllCustomer() : String +findCustomerDetail(in cusId : String) : String +deleteCustomer(in cusId : String) : String +updateCustomer(in cusId : String, in cusName : String, in cusLastName : String, in cusAddress : String, in cusTel : String, in cusFax : String) : String +addNewCustomer(in cusId : String, in cusName : String, in cusLastName : String, in cusAddress : String, in cusTel : String, in cusFax : String, in due : String) : String

รูปที่ 3.16 ภาพแสดงรายละเอียดภายใน Class CustomerManagement

ตารางที่ 3.15 Attribute ภายใน Class CustomerManagement

ชื่อ	หน้าที่
SQLCommand	เป็น Attribute ที่ใช้เก็บ String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class

ตารางที่ 3.16 Method ภายใน Class CustomerManagement

ชื่อ	หน้าที่
CustomerManagement()	เป็น constructor method ของ Class
listAllCustomer()	ทำการสร้าง String ที่ใช้ดึงข้อมูลของลูกค้าทั้งหมด
findCustomerDetail(String cusId)	ทำการสร้าง String ที่ใช้ดึงรายละเอียดของลูกค้าที่ต้องการ
deleteCustomer(String cusId)	ทำการสร้าง String ที่ใช้ลบข้อมูลลูกค้าที่ต้องการ
updateCustomer(String cusId, String cusName, String cusLastName, String cusAddress, String cusTel, String cusFax)	ทำการสร้าง String ที่ทำการเปลี่ยนแปลงข้อมูลลูกค้า
addNewCustomer(String cusId, String cusName, String cusLastName, String cusAddress, String cusTel, String cusFax, String due)	ทำการสร้าง String ที่เพิ่มข้อมูลลูกค้าใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.6 รายละเอียดส่วนประกอบภายใน Class EmployeeManagement

เป็น Class ที่ทำการสร้าง String คำสั่งที่เกี่ยวข้องกับการจัดการข้อมูลของพนักงาน

EmployeeManagement
-employeeId : String
-employeeName : String
-employeeUsername : String
-employeePassword : String
-DBManage : DataManager.DatabaseManagement
+EmployeeManagement()
+setEmployeeId(in employeeId : String)
+getEmployeeId() : String
+setEmployeeName(in employeeName : String)
+getEmployeeName() : String
+setEmployeeUsername(in employeeUsername : String)
+getEmployeeUsername() : String
+setEmployeePassword(in employeePassword : String)
+getEmployeePassword() : String
+changePassword(in newPassword : String) : String

รูปที่ 3.17 ภาพแสดงรายละเอียดภายใน Class EmployeeManagement

ตารางที่ 3.17 Attribute ภายใน Class EmployeeManagement

ชื่อ	หน้าที่
employeeId	เป็น Attribute ที่ใช้เก็บหมายเลขของพนักงาน
employeeName	เป็น Attribute ที่ใช้เก็บชื่อของพนักงาน
employeeUsername	เป็น Attribute ที่ใช้เก็บรหัสที่ใช้เข้าระบบ
employeePassword	เป็น Attribute ที่ใช้เก็บรหัสลับที่ใช้เข้าระบบ
DBManage	เป็นออบเจกต์ที่ใช้จัดการกับฐานข้อมูล

ตารางที่ 3.18 Method ภายใน Class EmployeeManagement

ชื่อ	หน้าที่
EmployeeManagement()	เป็น constructor method ของ Class
setEmployeeId(String employeeId)	ทำการใส่ค่าให้กับ Attribute employeeId
getEmployeeId()	ทำการคืนค่าที่อยู่ใน Attribute employeeId
setEmployeeName(String employeeName)	ทำการใส่ค่าให้กับ Attribute employeeName
getEmployeeName()	ทำการคืนค่าที่อยู่ใน Attribute employeeName
setEmployeeUsername(String employeeUsername)	ทำการใส่ค่าให้กับ Attribute employeeUsername

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

getEmployeeUsername()	ทำการคืนค่าที่อยู่ใน Attribute employeeUsername
setEmployeePassword(String employeePassword)	ทำการใส่ค่าให้กับ Attribute employeePassword
getEmployeePassword()	ทำการคืนค่าที่อยู่ใน Attribute employeePassword
changePassword(String newPassword)	ทำการเปลี่ยนรหัสลับของพนักงานที่ใช้เข้าระบบในฐานข้อมูล

3.2.2.7 รายละเอียดส่วนประกอบภายใน Class AdminManagement

เป็น Class ที่ทำการสร้าง String คำสั่งที่เกี่ยวกับการจัดการข้อมูลของผู้ดูแลระบบ

AdminManagement
-adminUsername : String -adminPassword : String -SQLCommand : String -DBManage : DataManage.DatabaseManagement
+AdminManagement() +getAdminUsername() : String +getAdminPassword() : String +authorizeAdmin(in username : String, in password : String) : Boolean +listAllEmployee() : Object +listDepartment() : Object +findEmployeeDetail(in empId : String) : Object +createNewEmployeePassword() : String +addNewEmployee(in empId : String, in empName : String, in empLastName : String, in empNickName : String, in empAddress : String, in empTel : String, in empMobile : String, in username : String, in deptId : String) : String +updateEmployeeDetail(in empId : String, in empName : String, in empLastName : String, in empNickName : String, in empAddress : String, in empTel : String, in empMobile : String, in deptId : String) : Integer +deleteEmployee(in empId : String) : Integer +changePassword(in newPassword : String) : Integer

รูปที่ 3.18 ภาพแสดงรายละเอียดภายใน Class AdminManagement

ตารางที่ 3.19 Attribute ภายใน Class AdminManagement

ชื่อ	หน้าที่
adminUsername	เป็น Attribute ที่เก็บชื่อที่ใช้เข้าระบบของ administrator
adminPassword	เป็น Attribute ที่เก็บรหัสลับที่ใช้เข้าระบบของ administrator
SQLCommand	เป็น Attribute ที่ใช้เก็บ String คำสั่ง SQL ที่สร้างขึ้นจาก method ใน Class
DBManage	เป็นออบเจกต์ที่ใช้จัดการกับฐานข้อมูล

ตารางที่ 3.20 Method ภายใน Class AdminManagement

ชื่อ	หน้าที่
AdminManagement()	เป็น constructor method ของ Class
getAdminUsername()	ทำการคืนค่าที่อยู่ใน Attribute adminUsername
getAdminPassword()	ทำการคืนค่าที่อยู่ใน Attribute adminPassword
authorizeAdmin(String username, String password)	ทำการตรวจสอบสิทธิ์การเข้าใช้ระบบ admin
listAllEmployee()	ทำการดึงข้อมูลพนักงานทั้งหมดจากฐานข้อมูล
listDepartment()	ทำการดึงข้อมูลแผนกทั้งหมดจากฐานข้อมูล
findEmployeeDetail(String empId)	ทำการดึงข้อมูลพนักงานที่ต้องการจากฐานข้อมูล
createNewEmployeePassword()	ทำการสร้างรหัสลับที่ใช้ในระบบของพนักงานใหม่
addNewEmployee(String empId, String empName, String empLastName, String empNickName, String empAddress, String empTel, String empMobile, String username, String depId)	ทำการเพิ่มข้อมูลพนักงานใหม่เข้าไปในฐานข้อมูล
updateEmployeeDetail(String empId, String empName, String empLastName, String empNickName, String empAddress, String empTel, String empMobile, String depId)	ทำการเปลี่ยนแปลงข้อมูลพนักงานในฐานข้อมูล
deleteEmployee(String empId)	ทำการลบข้อมูลพนักงานออกจากฐานข้อมูล
changePassword(String newPassword)	ทำการเปลี่ยนรหัสลับที่ใช้ในระบบของ admin ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.8 รายละเอียดส่วนประกอบภายใน Class CheckLogin

เป็น Class ที่ช่วยจัดการเกี่ยวกับการตรวจสอบสิทธิ์การเข้าใช้ระบบของพนักงาน

CheckLogin
-con : Connection
-stmt : Statement
-rs : ResultSet
+connectMySQL()
+disconnectDatabase()
+validateUser(in username : String, in password : String) : Object
+checkDepartment(in depid : String) : Object

รูปที่ 3.19 ภาพแสดงรายละเอียดภายใน Class CheckLogin

ตารางที่ 3.21 Attribute ภายใน Class CheckLogin

ชื่อ	หน้าที่
con	เป็นออบเจกต์ที่ใช้สำหรับติดต่อฐานข้อมูล
stmt	เป็นออบเจกต์ที่ใช้สำหรับการจัดการข้อมูลกับฐานข้อมูล
rs	เป็นออบเจกต์ที่ไว้รับผลของการดึงข้อมูลจากฐานข้อมูล

ตารางที่ 3.22 Method ภายใน Class CheckLogin

ชื่อ	หน้าที่
connectMySQL()	ทำการเชื่อมต่อกับฐานข้อมูลที่ต้องการ
disconnectDatabase()	ทำการตัดการเชื่อมต่อกับฐานข้อมูลที่ต้องการ
validateUser(String username, String password)	ทำการตรวจสอบสิทธิ์ของพนักงานที่เข้าใช้ระบบ
checkDepartment(String depid)	ทำการดึงข้อมูลแผนกที่พนักงานสังกัดอยู่

3.2.2.9 รายละเอียดส่วนประกอบภายใน Class CheckUser

เป็น Class ที่จัดการตรวจสอบสิทธิ์การเข้าใช้ระบบของพนักงาน

CheckUser
+init()
+doPost(in req : HttpServletRequest, in res : HttpServletResponse)
+gotoPage(in path : String, in req : HttpServletRequest, in res : HttpServletResponse)

รูปที่ 3.20 ภาพแสดงรายละเอียดภายใน Class CheckUser

ตารางที่ 3.23 Method ภายใน Class CheckUser

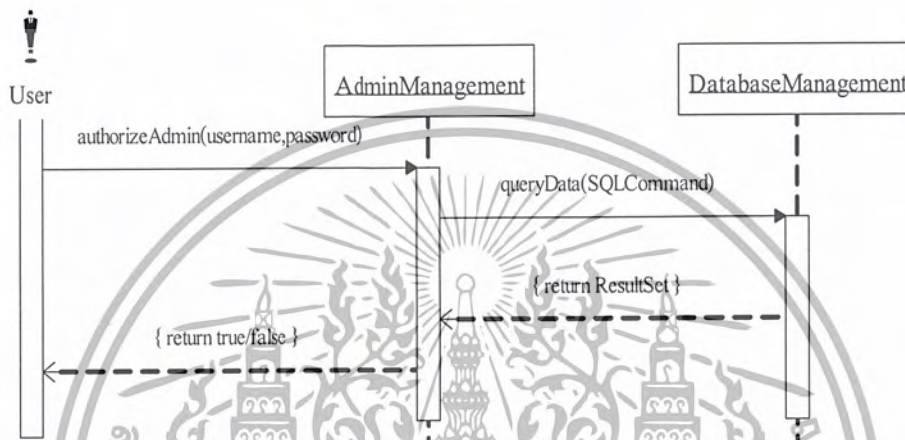
ชื่อ	หน้าที่
init()	ทำการกำหนดค่าเริ่มต้นของ Servlet
doPost(HttpServletRequest req, HttpServletResponse res)	ทำการรองรับการร้องขอจากผู้ใช้ให้ตรวจสอบสิทธิ์การเข้าใช้งานของระบบการขายสินค้า
gotoPage(String path, HttpServletRequest req, HttpServletResponse res)	ทำการส่งออบเจกต์ HttpServletRequest และ HttpServletResponse ไปยังหน้าที่กำหนด

3.3 Sequence Diagram ของระบบการจำหน่ายสินค้า

Sequence Diagram จะแบ่งเป็นกลุ่มใหญ่ๆ ตามลักษณะงานของผู้ใช้ ได้ 2 กลุ่ม คือ Sequence Diagram ของผู้ใช้ที่เป็น Administrator, Sequence Diagram ของผู้ใช้ที่เป็นพนักงาน

3.3.1 Sequence Diagram ของผู้ใช้ที่เป็น Administrator

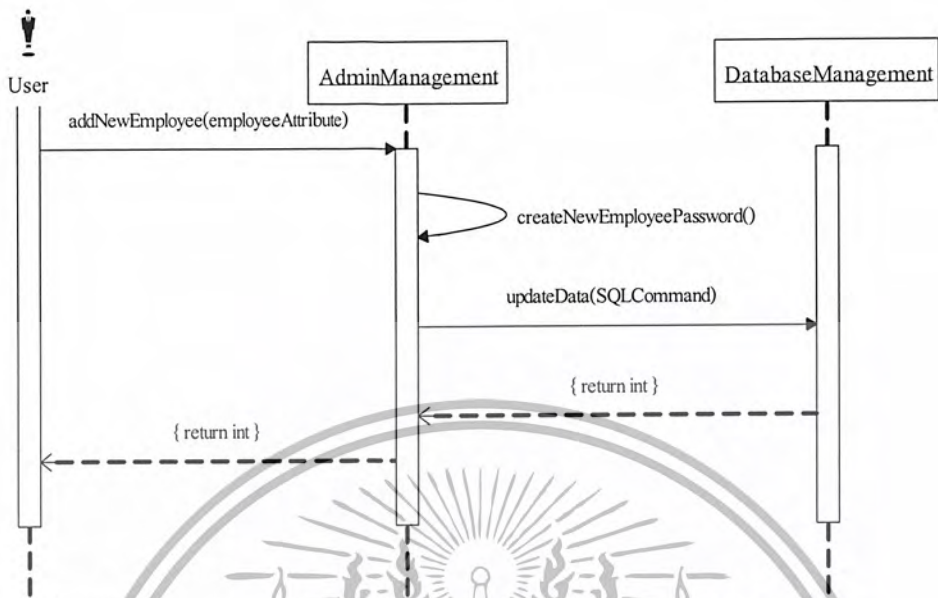
3.3.1.1 Sequence Diagram การตรวจสอบสิทธิ์การเข้าใช้ระบบของ Admin



รูปที่ 3.21 แสดง Sequence Diagram การตรวจสอบสิทธิ์การเข้าใช้ระบบของ Admin

ผู้ใช้จะทำการเรียก method `authorizeAdmin` ของ Class `AdminManagement` โดยจะส่ง `username` และ `password` ที่ใช้เข้าไปในระบบเป็น parameter จากนั้น method `authorize` จะทำการดึงข้อมูลออกมา ถ้ามีข้อมูลก็จะทำการคืนค่า `true` ไปให้กับผู้ใช้ แต่ถ้าไม่มีข้อมูล ก็จะทำการคืนค่า `false` ให้กับผู้ใช้

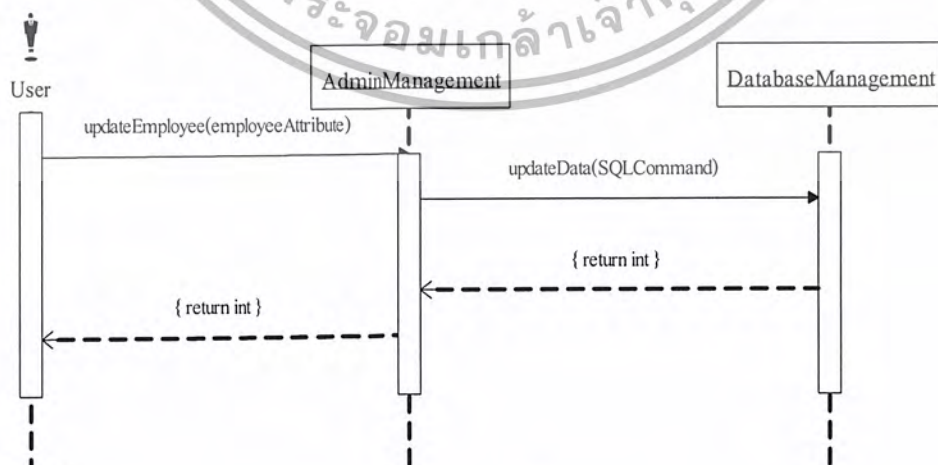
3.3.1.2 Sequence Diagram การเพิ่มข้อมูลของพนักงานใหม่



รูปที่ 3.22 แสดง Sequence Diagram การเพิ่มข้อมูลของพนักงานใหม่

ผู้ใช้งานจะทำการเรียก method `addNewEmployee` ของ Class `AdminManagement` โดยจะส่งข้อมูลของพนักงานที่ต้องการเพิ่มเป็น parameter แล้วจะทำการสร้าง password เพื่อให้พนักงานคนนั้นสามารถเข้าระบบได้ และจะทำการบันทึกข้อมูล ลงฐานข้อมูลโดยผ่านทาง method `updateData` ของ Class `DatabaseManagement`

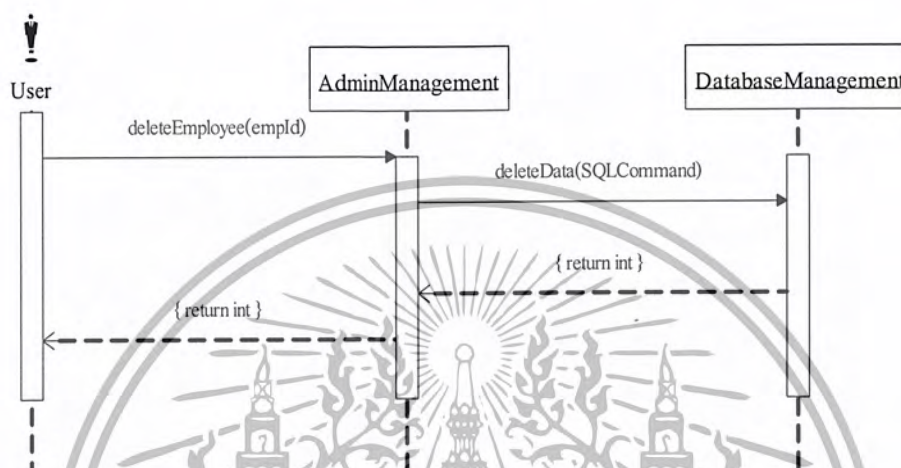
3.3.1.3 Sequence Diagram การแก้ไขข้อมูลของพนักงาน



รูปที่ 3.23 แสดง Sequence Diagram การแก้ไขข้อมูลของพนักงาน

ผู้ใช้จะทำการเรียก method updateEmployee ของ Class AdminManagement โดยจะส่งข้อมูลพนักงานที่ได้เปลี่ยนแปลงแล้วเป็น parameter จากนั้นทำการบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method updateData ของ Class DatabaseManagement

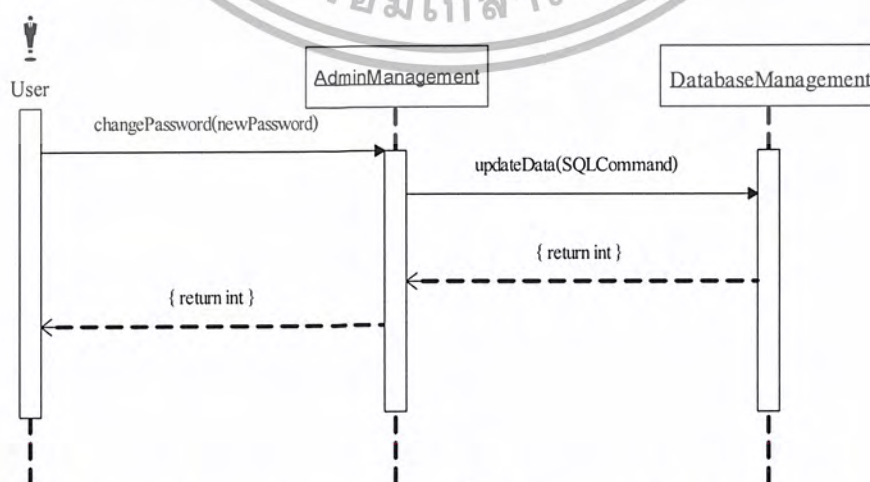
3.3.1.4 Sequence Diagram การลบข้อมูลพนักงาน



รูปที่ 3.24 แสดง Sequence Diagram การลบข้อมูลพนักงาน

ผู้ใช้จะทำการเรียก method deleteEmployee ของ Class AdminManagement โดยส่งหมายเลขประจำตัวพนักงานเพื่อเป็น parameter แล้วจะทำการลบข้อมูลออกจากฐานข้อมูลโดยผ่านทาง method deleteData ของ Class DatabaseManagement

3.3.1.5 Sequence Diagram การเปลี่ยน Password เข้าสู่ระบบของ Admin



รูปที่ 3.25 แสดง Sequence Diagram การเปลี่ยน Password เข้าสู่ระบบของ Admin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

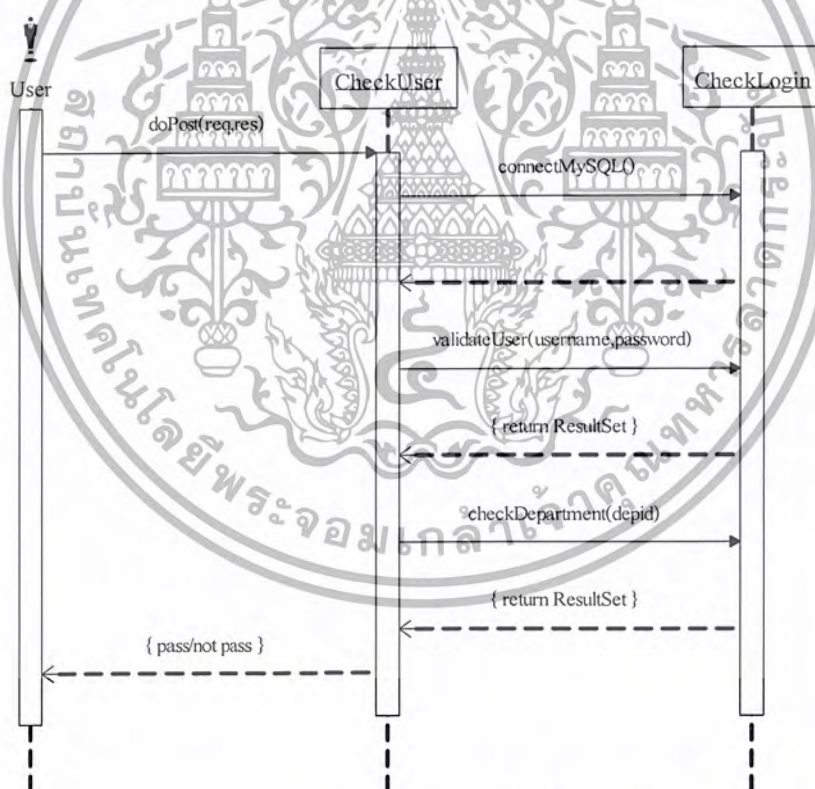
ผู้ใช้จะทำการเรียก method `changePassword` ของ Class `AdminManagement` โดยส่ง password ตัวใหม่ไป เพื่อให้เป็น parameter แล้วจะทำการบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

3.3.2 Sequence Diagram ของผู้ใช้ที่เป็นพนักงาน

Sequence Diagram ในกลุ่มนี้ยังแบ่งย่อยออกได้อีกเป็น 4 กลุ่มตามลักษณะแผนกงานของผู้ใช้ คือ Sequence Diagram ของผู้ใช้ที่เป็นพนักงานทุกแผนก, Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Marketing, Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Stock และ Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Account

3.3.2.1 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานทุกแผนก

1) Sequence Diagram การตรวจสอบสิทธิ์การเข้าใช้ระบบของพนักงาน



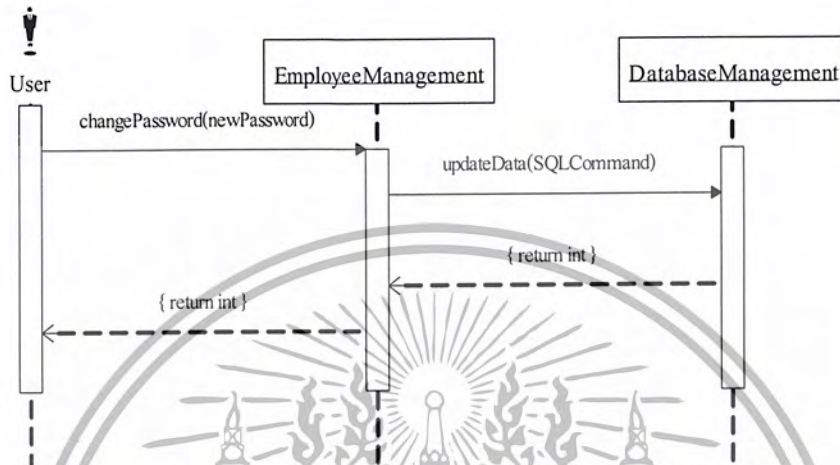
รูปที่ 3.26 แสดง Sequence Diagram การตรวจสอบสิทธิ์การเข้าใช้ระบบของพนักงาน

ผู้ใช้จะทำการเรียก method `doPost` ของ Class `CheckUser` ซึ่งเป็น Servlet เรียบร้อยแล้ว จากนั้นจะใช้ method `doPost` เพื่อทำการเชื่อมต่อกับฐานข้อมูล และทำการตรวจสอบ username และ password โดยเรียก method `validateUser` ของ Class `CheckLogin` เมื่อดำเนินการสำเร็จแล้วจะเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

method `checkDepartment` เพื่อตรวจสอบว่าพนักงานที่จะเข้าใช้ระบบอยู่แผนกใด จากนั้นจะทำการสร้าง link ไปหน้าหลักของแต่ละแผนกเพื่อให้ผู้ใช้งานสามารถเข้าใช้งานได้

2) Sequence Diagram การเปลี่ยน Password ของพนักงาน

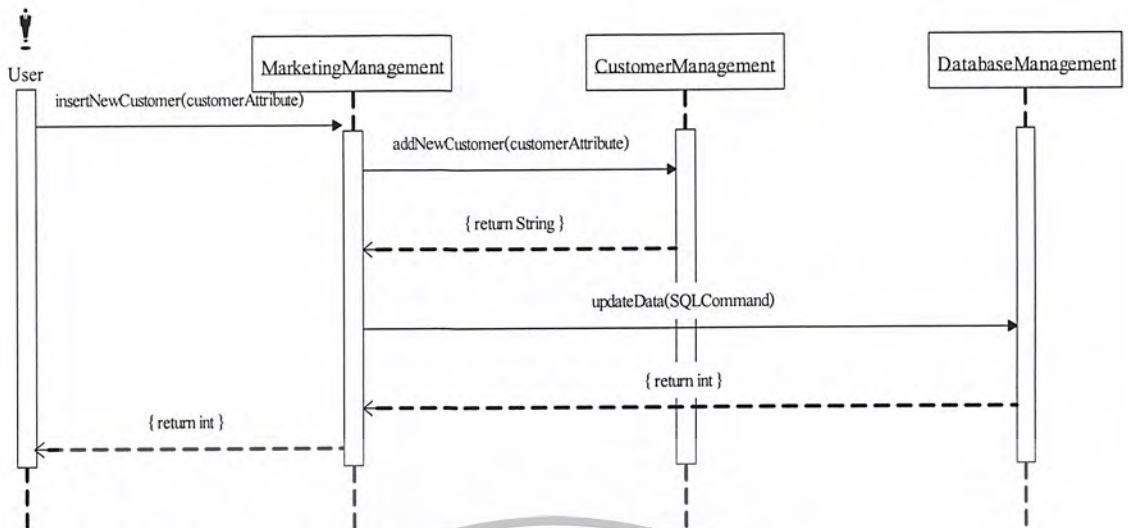


รูปที่ 3.27 แสดง Sequence Diagram การเปลี่ยน Password ของพนักงาน

ผู้ใช้งานจะทำการเรียก method `changePassword` ของ Class `EmployeeManagement` โดยส่ง password ใหม่เป็น parameter จากนั้น method `changePassword` จะทำการบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

3.3.2.2 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Marketing

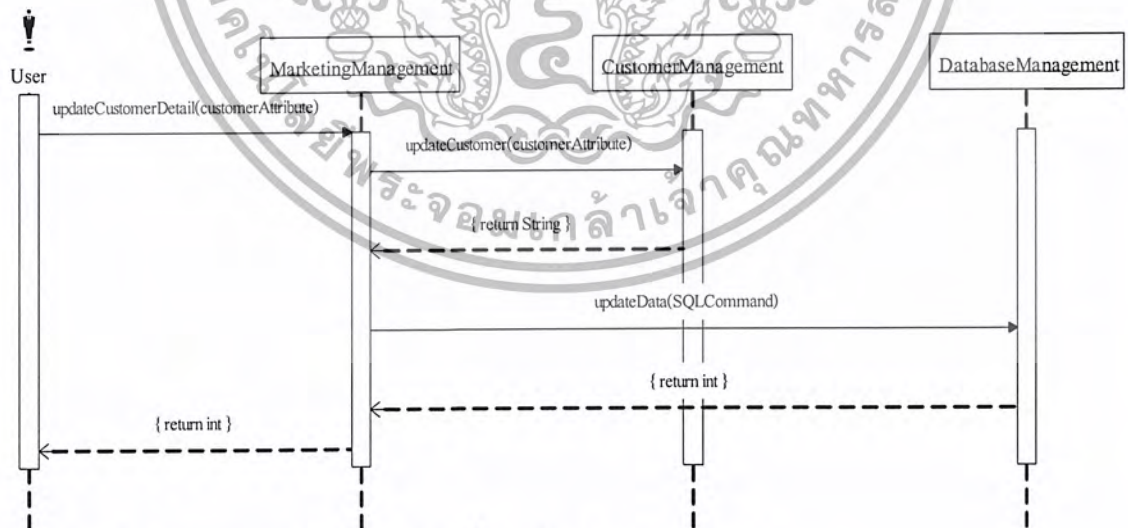
1) Sequence Diagram การเพิ่มข้อมูลของลูกค้าใหม่



รูปที่ 3.28 แสดง Sequence Diagram การเพิ่มข้อมูลของลูกค้าใหม่

ผู้ใช้จะทำการเรียก method insertNewCustomer ของ Class MarketingManagement โดยส่งข้อมูลลูกค้า เพื่อเป็น parameter จากนั้น method insertNewCustomer จะทำการเรียก method addNewCustomer ของ Class CustomerManagement เพื่อให้ทำการสร้างคำสั่ง SQL แล้วทำการบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method updateData ของ Class DatabaseManagement

2) Sequence Diagram การเปลี่ยนแปลงข้อมูลของลูกค้า

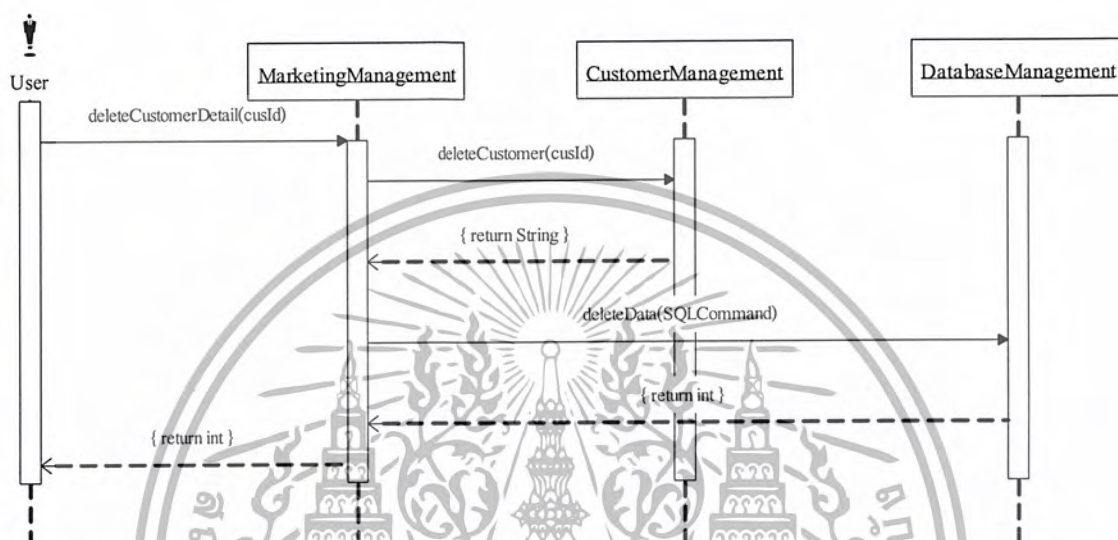


รูปที่ 3.29 แสดง Sequence Diagram การเปลี่ยนแปลงข้อมูลของลูกค้า

ผู้ใช้จะทำการเรียก method updateCustomerDetail ของ Class MarketingManagement โดยส่งข้อมูลที่ทำการเปลี่ยนแปลงแล้วเพื่อเป็น parameter จากนั้น method updateCustomerDetail จะทำเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียก method `updateCustomer` ของ Class `CustomerManagement` เพื่อทำการสร้างคำสั่ง SQL แล้วทำการบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

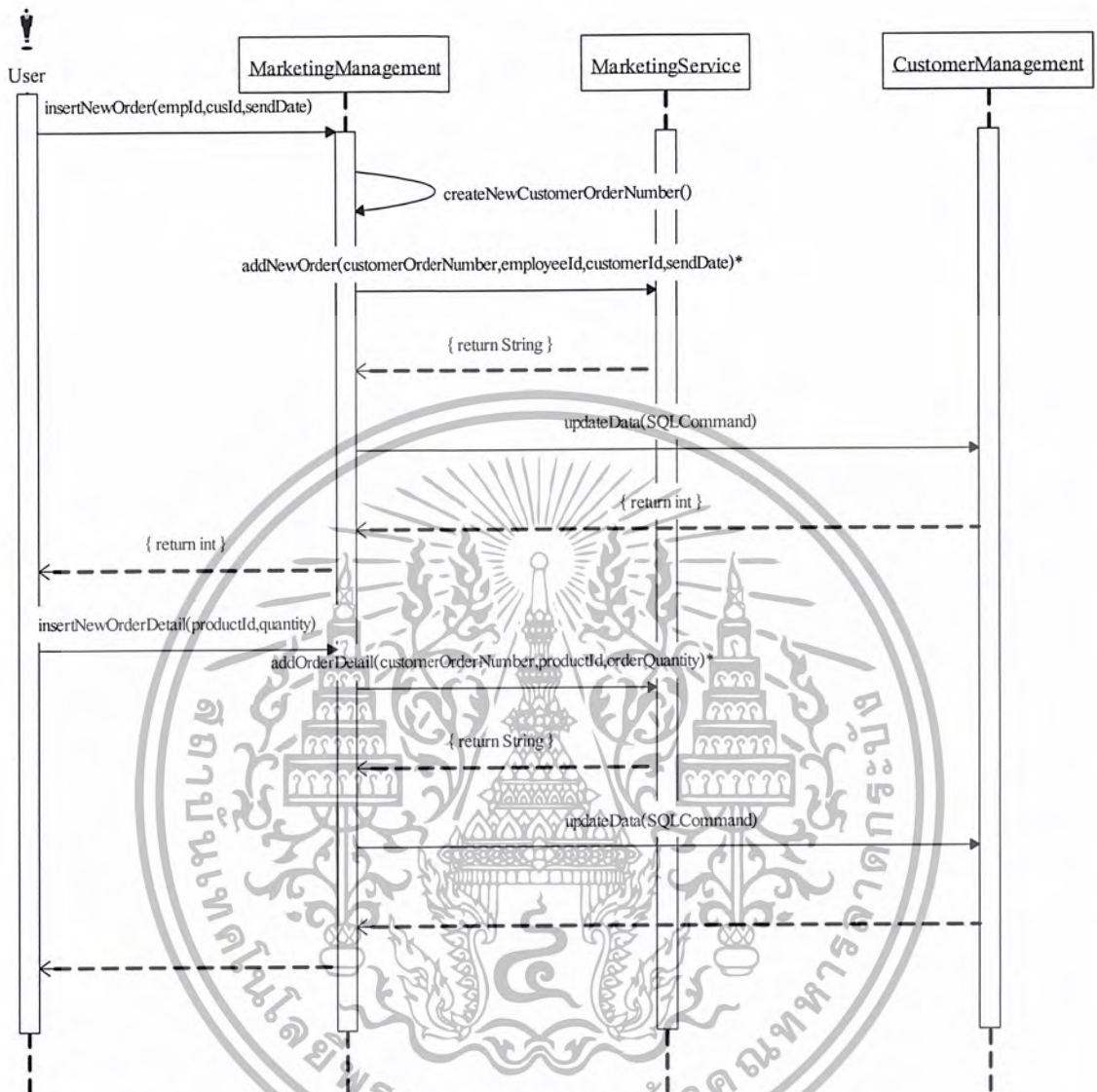
3) Sequence Diagram การลบข้อมูลของลูกค้า



รูปที่ 3.30 แสดง Sequence Diagram การลบข้อมูลของลูกค้า

ผู้ใช้จะทำการเรียก method `deleteCustomerDetail` ของ Class `MarketingManagement` โดยส่งหมายเลขลูกค้าเพื่อเป็น parameter จากนั้น method `deleteCustomerDetail` จะทำการเรียก method `deleteCustomer` ของ Class `CustomerManagement` เพื่อทำการสร้างคำสั่ง SQL แล้วทำการลบข้อมูลออกจากฐานข้อมูลผ่านทาง method `deleteData` ของ Class `DatabaseManagement`

4) Sequence Diagram การสร้างใบสั่งซื้อสินค้า



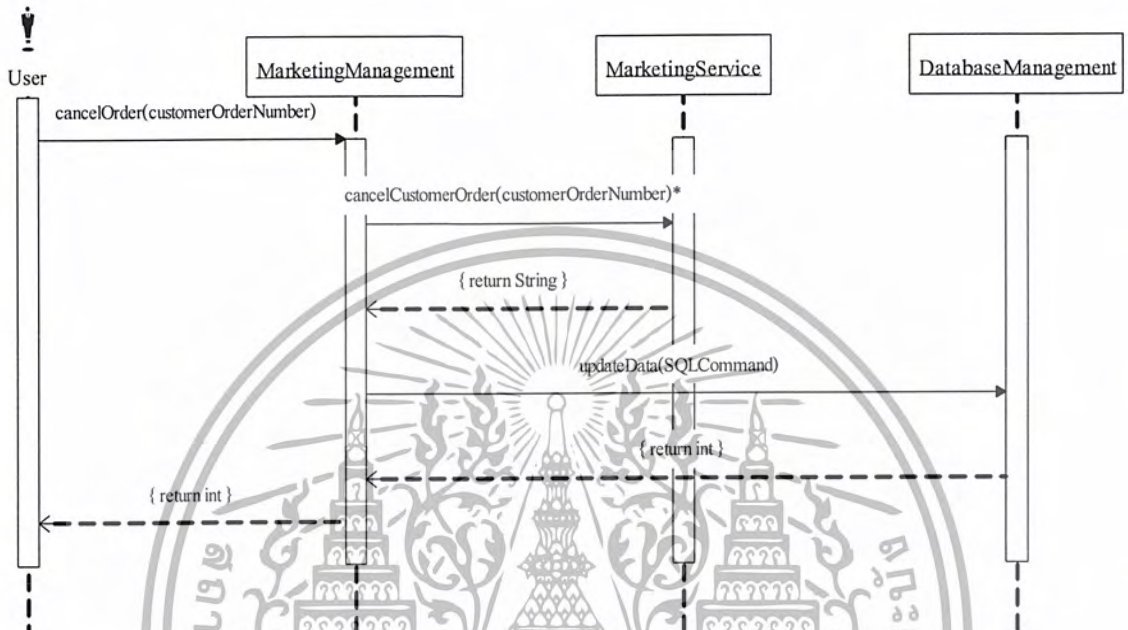
รูปที่ 3.31 แสดง Sequence Diagram การสร้างใบสั่งซื้อสินค้า

ผู้ใช้จะทำการเรียก method `insertNewOrder` ของ Class `MarketingManagement` โดยส่งหมายเลขพนักงาน, หมายเลขลูกค้า และวันสั่งซื้อสินค้าเพื่อเป็น parameter จากนั้น method `insertNewOrder` จะทำการเรียก method `createNewCustomerOrderNumber` เพื่อทำการสร้างหมายเลขใบสั่งซื้อสินค้าอันใหม่ แล้วทำการเรียก method `addNewOrder` ของ Class `MarketingService` เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement` จากนั้นผู้ใช้จะต้องเรียก method `insertNewOrderDetail` ของ Class `MarketingManagement` โดยส่งหมายเลขสินค้า และจำนวนที่สั่งซื้อเพื่อเป็น parameter ในการบันทึกรายละเอียดของสินค้าที่ถูกสั่งซื้อจากใบสั่งซื้อสินค้าใบนี้ จากนั้น method `insertNewOrderDetail` จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการเรียก method `addOrderDetail` ของ Class `MarketingService` เพื่อสร้างคำสั่ง SQL แล้วทำการบันทึกลงฐานข้อมูล

5) Sequence Diagram การยกเลิกใบสั่งสินค้า

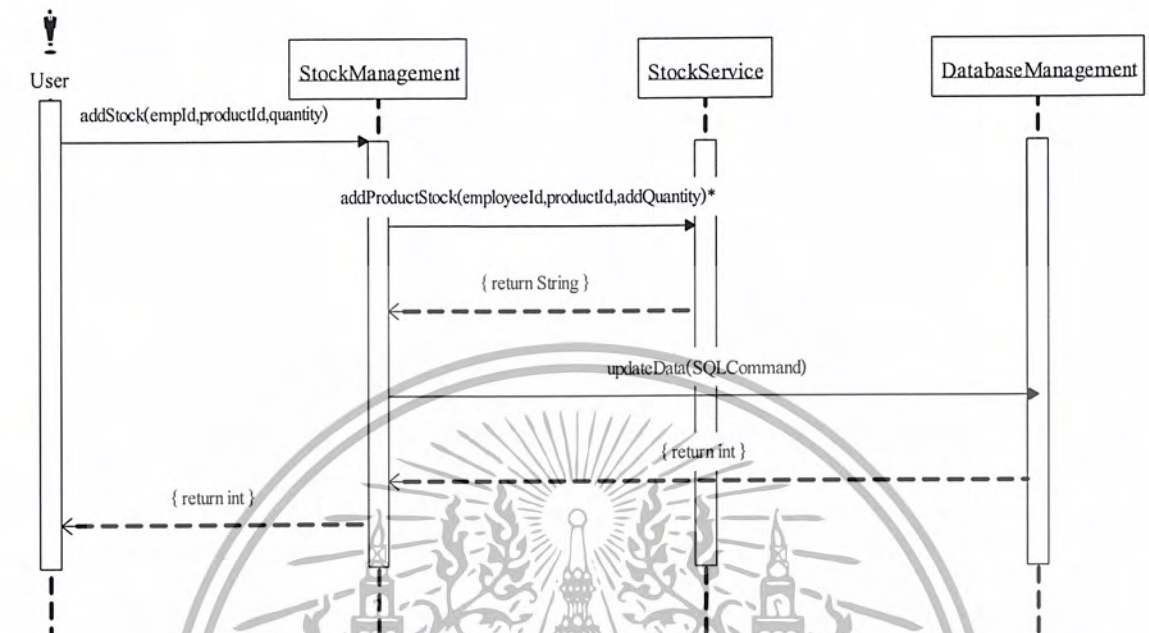


รูปที่ 3.32 แสดง Sequence Diagram การยกเลิกใบสั่งสินค้า

ผู้ใช้งานจะทำการเรียก method `cancelOrder` ของ Class `MarketingManagement` โดยส่งหมายเลขใบสั่งสินค้าไปเป็น parameter จากนั้น method `cancelOrder` จะทำการเรียก method `cancelCustomerOrder` ของ Class `MarketingService` เพื่อทำการสร้างคำสั่ง SQL แล้วทำการบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

3.3.2.3 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Stock

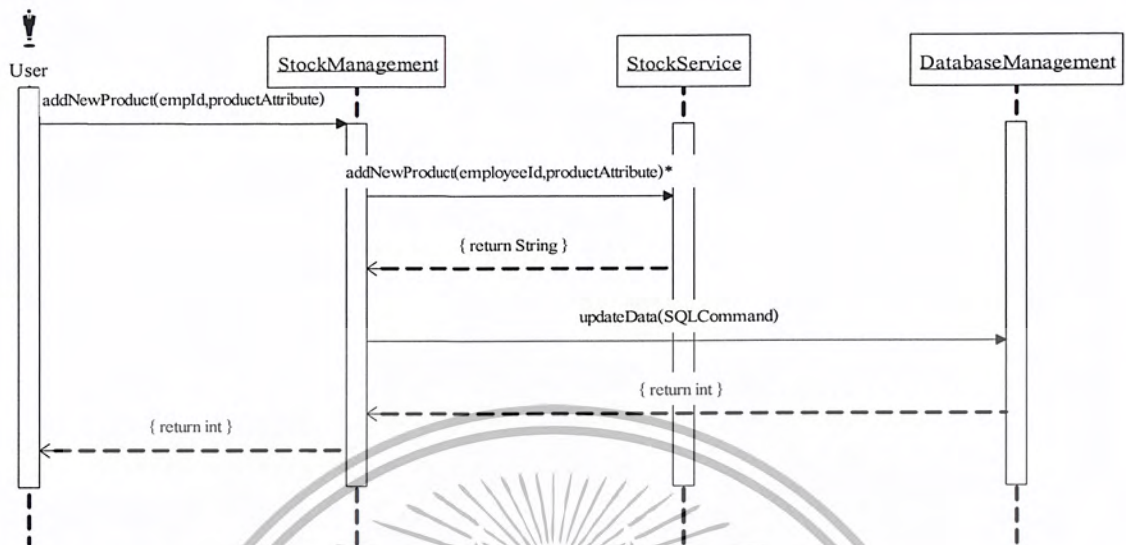
1) Sequence Diagram การเพิ่มจำนวนสินค้าในคลัง



รูปที่ 3.33 แสดง Sequence Diagram การเพิ่มจำนวนสินค้าในคลัง

ผู้ใช้จะทำการเรียก method `addStock` ของ Class `StockManagement` โดยส่งหมายเลขสินค้าและจำนวนที่จะทำการเพิ่มไปเป็น parameter จากนั้น method `addStock` จะทำการเรียก method `addProductStock` ของ Class `StockService` เพื่อทำการสร้างคำสั่ง SQL แล้วทำการบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

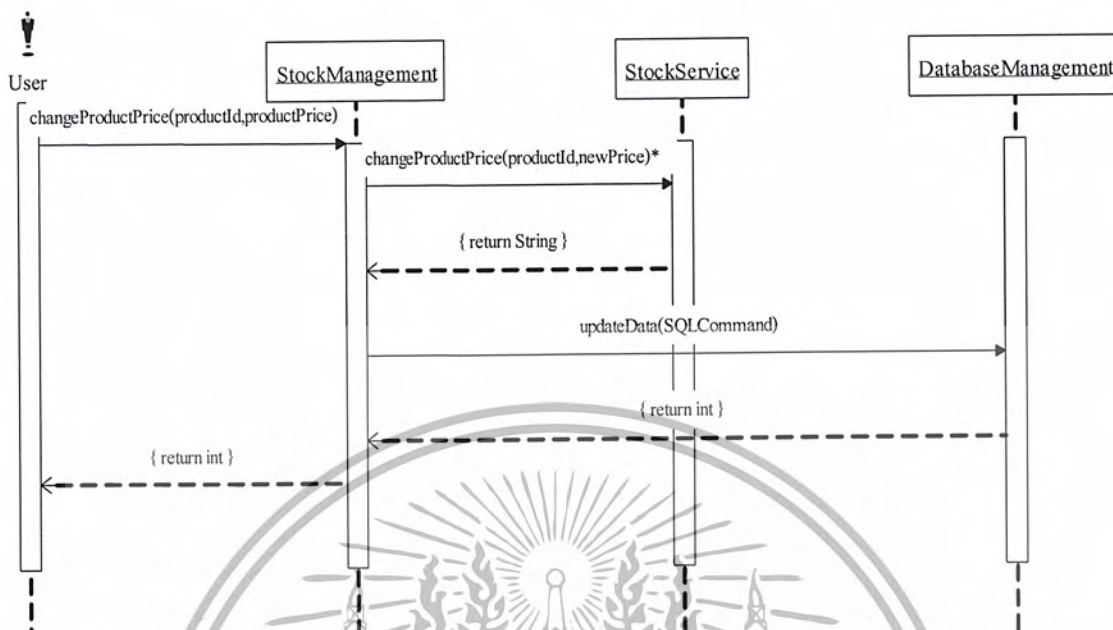
2) Sequence Diagram การเพิ่มรายการสินค้าใหม่



รูปที่ 3.34 แสดง Sequence Diagram การเพิ่มรายการสินค้าใหม่

ผู้ใช้งานจะทำการเรียก method `addNewProduct` ของ Class `StockManagement` โดยส่ง หมายเลขพนักงานและข้อมูลของสินค้าเพื่อเป็น parameter จากนั้น method `addNewProduct` จะทำการเรียก method `addNewProduct` ของ Class `StockService` เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

3) Sequence Diagram การเปลี่ยนแปลงราคาสินค้า

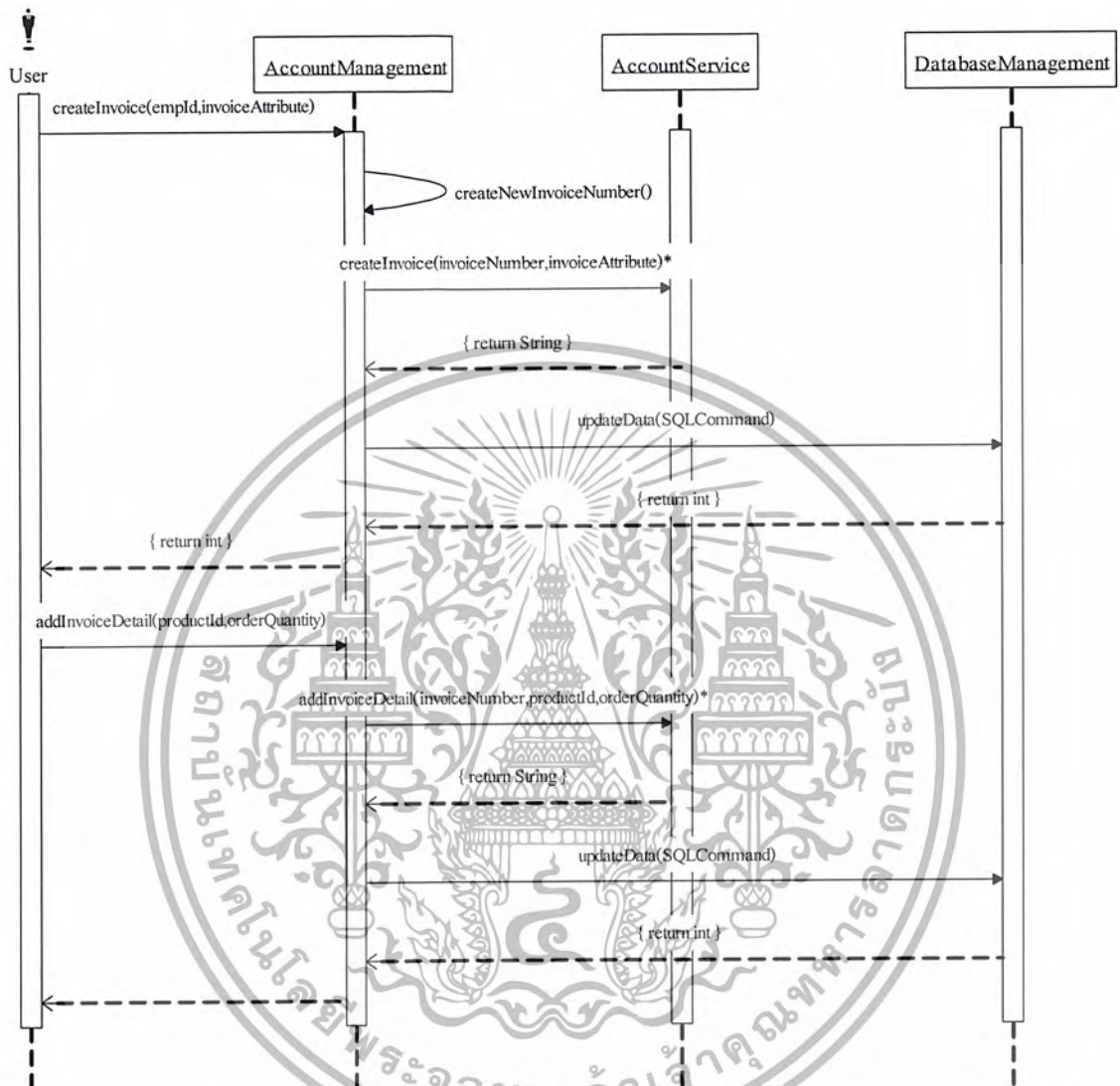


รูปที่ 3.35 แสดง Sequence Diagram การเปลี่ยนแปลงราคาสินค้า

ผู้ใช้งานทำการเรียก method `changeProductPrice` ของ Class `StockManagement` โดยส่งหมายเลขสินค้าและราคาสินค้าไปเป็น parameter จากนั้น method `changeProductPrice` จะทำการเรียก method `changeProductPrice` ของ Class `StockService` เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

3.3.2.4 Sequence Diagram ของผู้ใช้ที่เป็นพนักงานแผนก Account

1) Sequence Diagram การออกไป Invoice

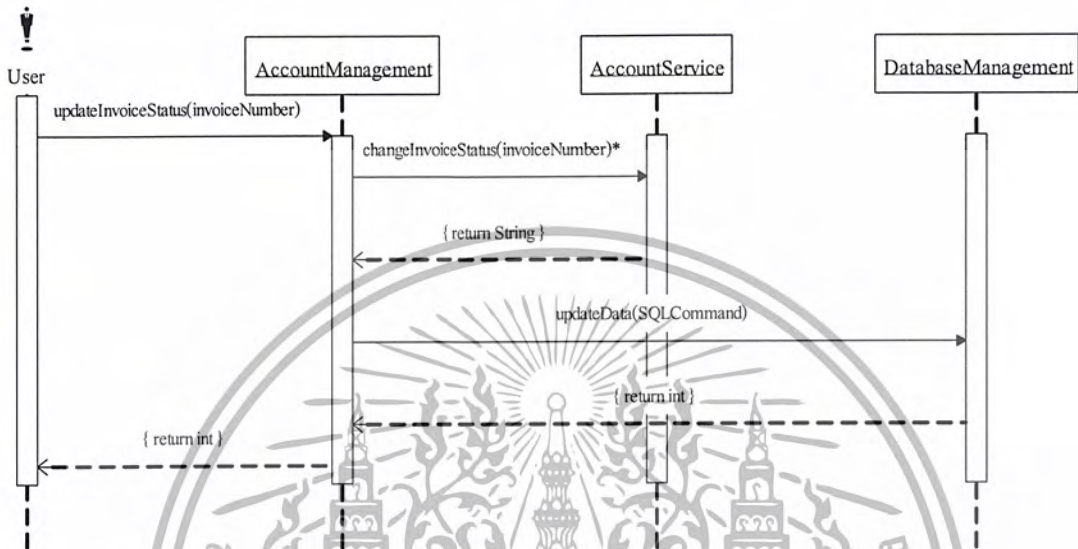


รูปที่ 3.36 แสดง Sequence Diagram การออกไป Invoice

ผู้ใช้จะทำการเรียก method createInvoice ของ Class AccountManagement โดยส่งหมายเลขพนักงานและข้อมูลใบ Invoice เพื่อเป็น parameter จากนั้น method createInvoice จะทำการเรียก method createNewInvoiceNumber เพื่อสร้างหมายเลขใบ Invoice ใหม่แล้วทำการเรียก method createInvoice ของ Class AccountService เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method updateData ของ Class DatabaseManagement จากนั้นผู้ใช้จะต้องเรียก method addInvoiceDetail ของ Class AccountManagement เพื่อทำการบันทึกรายละเอียดของใบ Invoice โดยส่งหมายเลขสินค้าและจำนวนที่สั่งเป็น parameter แล้ว method addInvoiceDetail จะทำการ

เรียก method `addInvoiceDetail` ของ Class `AccountService` เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData`

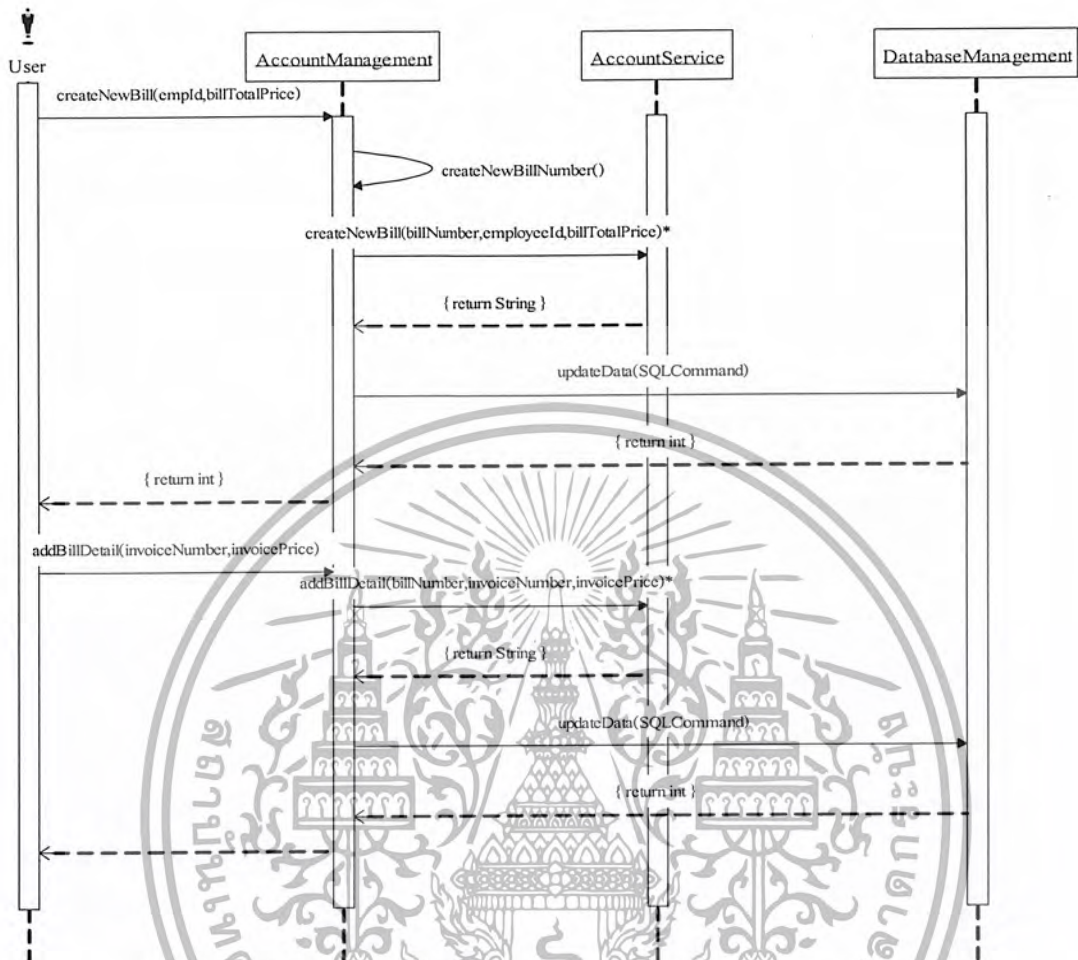
2) Sequence Diagram การเปลี่ยนสถานะของใบ Invoice



รูปที่ 3.37 แสดง Sequence Diagram การเปลี่ยนสถานะของใบ Invoice

ผู้ใช้งานทำการเรียก method `updateInvoiceStatus` ของ Class `AccountManagement` โดยส่งหมายเลข Invoice เป็น parameter จากนั้น method `updateInvoiceStatus` จะทำการเรียก method `changeInvoiceStatus` ของ Class `AccountService` เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement`

3) Sequence Diagram การออกใบวางบิล



รูปที่ 3.38 แสดง Sequence Diagram การออกใบวางบิล

ผู้ใช้งานจะทำการเรียก method `createNewBill` ของ Class `AccountManagement` โดยส่งหมายเลขพนักงานและราคารวมของใบวางบิลเพื่อเป็น parameter จากนั้น method `createNewBill` จะทำการเรียก method `createNewBillNumber` เพื่อสร้างหมายเลขใบวางบิลใหม่แล้วทำการเรียก method `createNewBill` ของ Class `AccountService` เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData` ของ Class `DatabaseManagement` จากนั้นผู้ใช้งานจะต้องเรียก method `addBillDetail` ของ Class `AccountManagement` เพื่อทำการบันทึกรายละเอียดของใบวางบิล โดยส่งหมายเลขใบ Invoice และราคาของใบ Invoice เป็น parameter แล้ว method `addBillDetail` จะทำการเรียก method `addBillDetail` ของ Class `AccountService` เพื่อทำการสร้างคำสั่ง SQL แล้วบันทึกข้อมูลลงฐานข้อมูลผ่านทาง method `updateData`

หมายเหตุ * ที่อยู่ต่อท้าย method ในรูปแสดงถึงการเรียกใช้งานผ่าน Web Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดสอบและการพัฒนาระบบ

4.1 รายละเอียดของการพัฒนาระบบ

4.1.1 ภาษาที่ใช้ในการพัฒนา

ใช้ภาษา JAVA Version 1.4.1_01 , JSP, Enterprise JAVA Bean และ XML ในการพัฒนา

4.1.2 เครื่องมือที่ใช้ในการพัฒนา

1. J Creator
2. Macromedia Dreamwever MX
3. Sun One
4. JAVA Web Service

4.2 การพัฒนาระบบให้สามารถเป็น Web Service

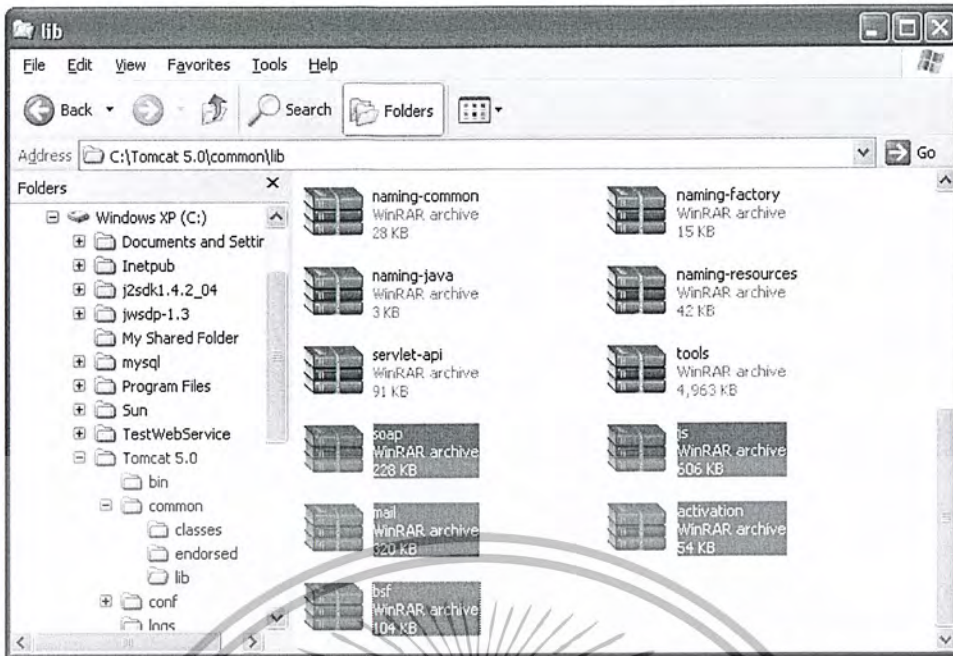
4.2.1 การพัฒนาระบบให้เป็น Web Service ฟังก์ชันเซอร์เวอร์

4.2.1.1 การเพิ่ม package และ throw exception

ภายในโปรแกรมที่อยู่ทางฝั่ง เซิร์ฟเวอร์ จะต้องทำการดึงบาง Package ของ Java เข้ามา เพื่อใช้งานร่วม ได้แก่ Package java.net.* , java.util.* และ org.apache.SoaException รวม ถึงจะต้องเพิ่มส่วน throw SOAPException ในบริเวณที่เป็น constructor method ทุกตัว

4.2.1.2 การเตรียม Tomcat

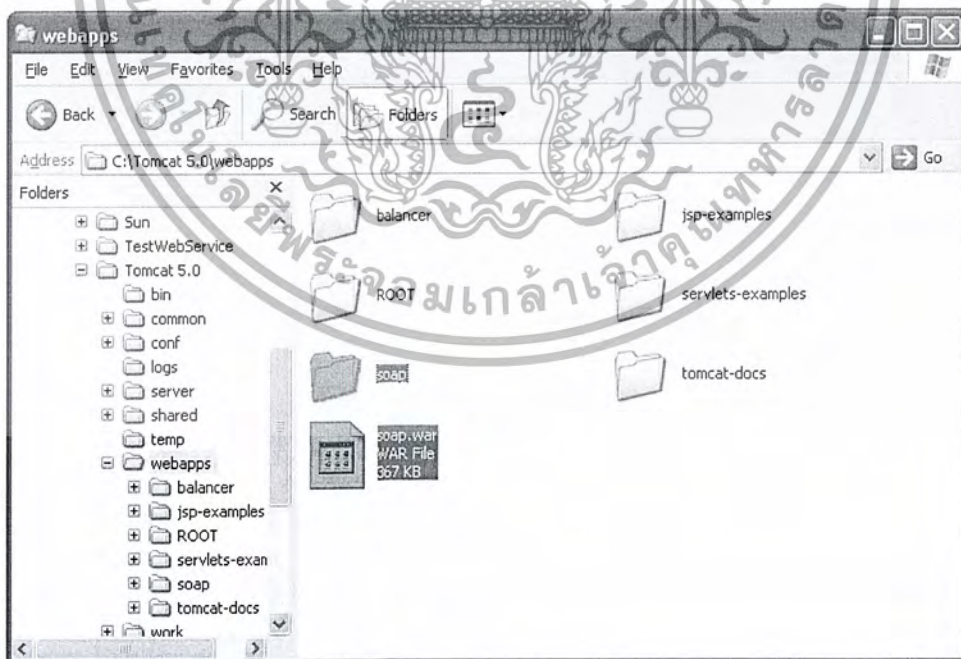
สามารถเตรียม Tomcat ให้พร้อมสำหรับการเรียกใช้โดย นำไฟล์ activation , bsf , soap และ mail มาใส่ลงใน folder lib ซึ่งอยู่ใน c:\Tomcat 5.0\common\lib ดังรูป



รูปที่ 4.1 หน้าจอแสดง package ที่ติดตั้งใน folder lib

4.2.1.3 การจัดการเก็บไฟล์ soap.war

จะต้องนำไฟล์ soap.war ซึ่งอยู่ใน Soap Package ไปเก็บไว้ที่ c:\Tomcat 5.0\webapps โดยหลังจากที่เรา run Tomcat แล้วไฟล์ จะทำการสร้าง folder soap ให้เอง ดังรูป

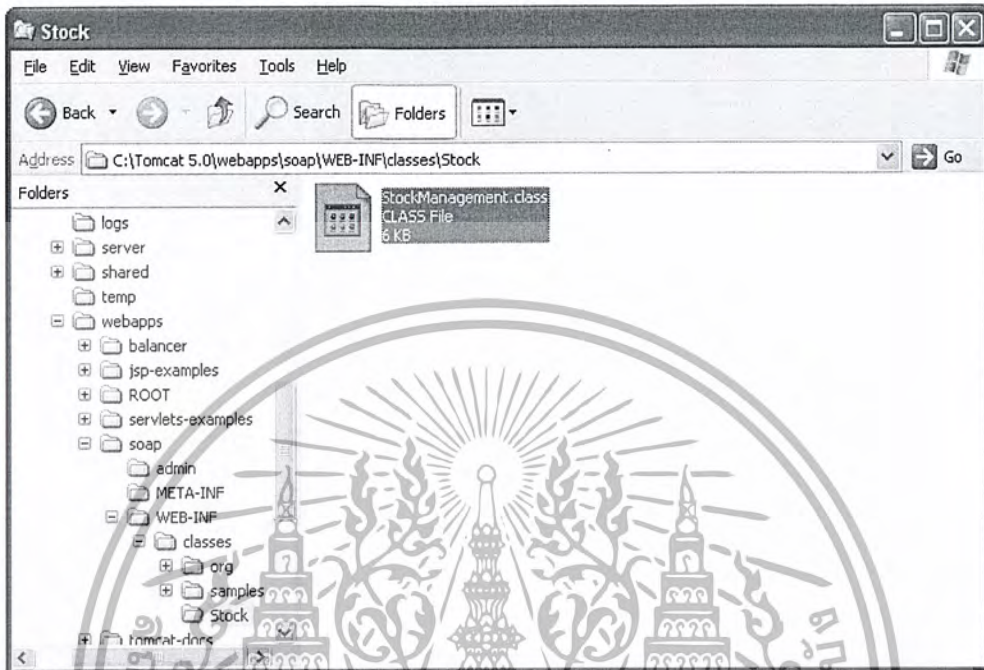


รูปที่ 4.2 หน้าจอแสดงไฟล์ soap.war

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.4 การเก็บไฟล์ที่ Compile แล้ว

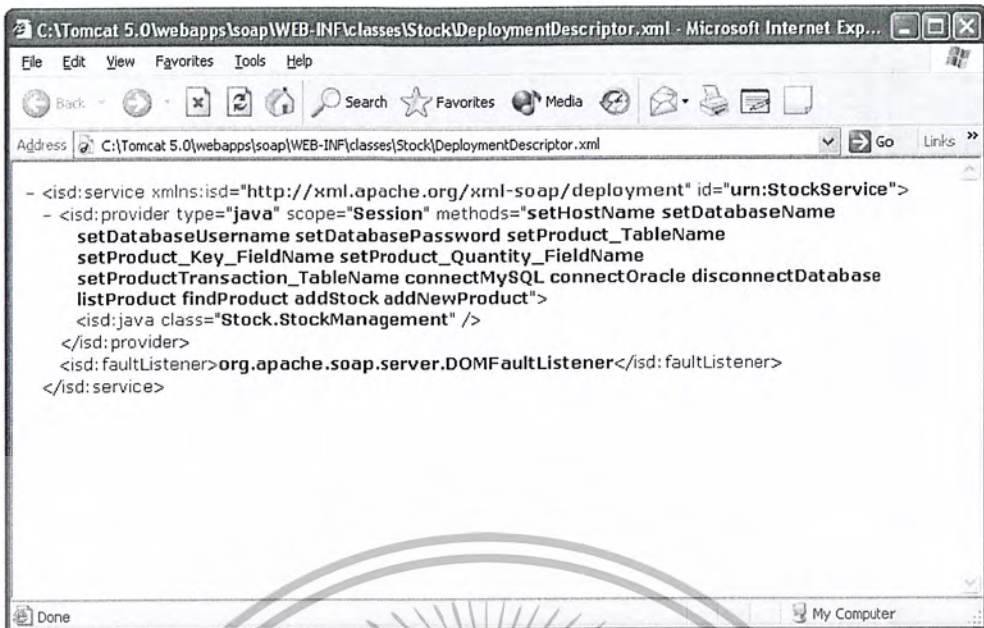
การเก็บไฟล์ที่เราได้ Compile แล้วจะต้องนำไปเก็บไว้ยัง folder soap ที่ Java สร้างให้ จากข้อ 4.2.1.3



รูปที่ 4.3 หน้าจอเก็บ file ที่ compile เรียบร้อยแล้ว

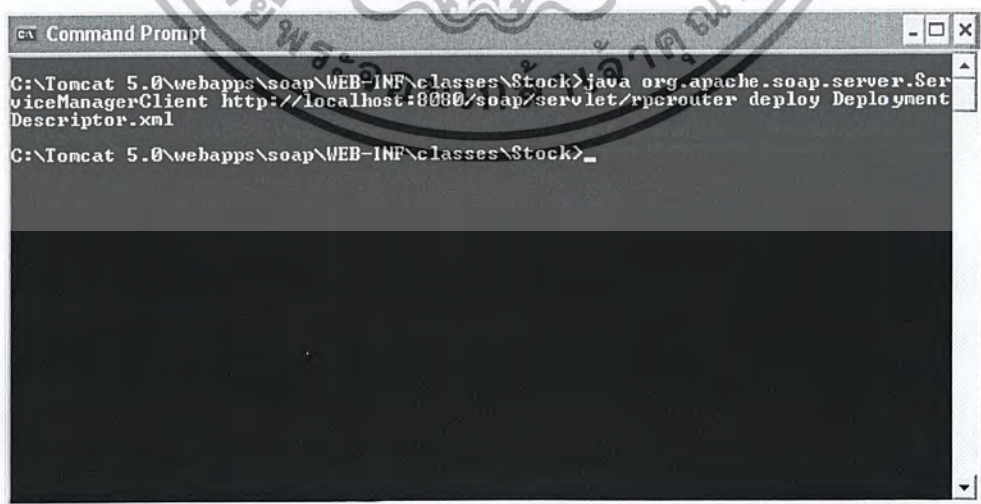
4.2.1.5 การทำ Deployment Descriptor เพื่อให้โปรโตคอล SOAP รู้จัก

จะต้องประกาศค่าต่างๆ เพื่อให้ โปรโตคอล SOAP รู้จักกับบริการที่ต้องการจะเปิดให้บริการ โดยค่า id จะบอกถึงชื่อของ Web Service , type จะบอกว่าบริการนี้ถูกสร้างด้วยภาษาใด , scope จะบอกรายละเอียดในตามกรกำหนดใช้บริการนี้ , method จะระบุว่าต้องการให้บริการ method ใดบ้างและถ้าต้องการให้บริการมากกว่า 1 ชนิดสามารถเว้นวรรค ระหว่างชื่อบริการได้ , java class เป็นการระบุว่าเรียกใช้ class ใด โดยถ้า class นั้นอยู่ใน Package จะระบุชื่อของ Package ก่อนแล้วกันด้วยจุด . ตามด้วยชื่อ class ดังรูป



รูปที่ 4.4 เอกสาร XML เพื่อประกาศบริการให้โปรโตคอล SOAP รู้จัก

การกำหนด path ของโปรโตคอล SOAP และการ deploy และ undeploy บริการ จากนั้นทำการกำหนด class path ของ SOAP โดย พิมพ์ `org.apache.soap.server.ServiceManagerClient` จากนั้นตามด้วยคำสั่งต่อไปนี้ list ถ้าต้องการตรวจสอบว่ามีบริการใดให้บริการอยู่บ้าง, deploy ถ้าต้องการลงทะเบียน Web Service และ undeploy เมื่อต้องการยกเลิกการให้บริการนั้นๆ และสุดท้ายตามด้วย ชื่อไฟล์นามสกุล) XML (ที่ได้ทำเป็น Deployment Description เรียบร้อยแล้วตามตัวอย่างข้อ 4.2.1.5) ดังรูป



รูปที่ 4.5 หน้าจอแสดงการ Deploy บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

C:\Tomcat 5.0\webapps\soap\WEB-INF\classes\Stock>java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter list
Deployed Services:
urn:StockService
C:\Tomcat 5.0\webapps\soap\WEB-INF\classes\Stock>_

```

รูปที่ 4.6 หน้าจอแสดงผลว่ามีบริการใดให้บริการอยู่บ้าง

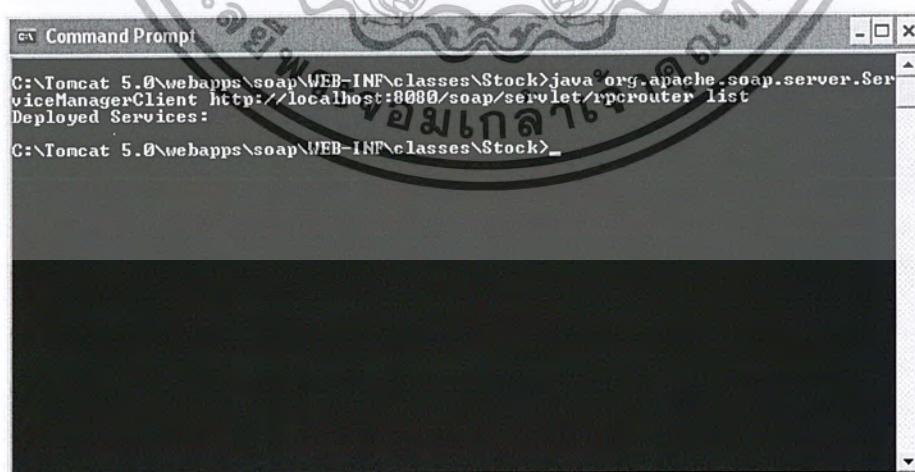


```

C:\Tomcat 5.0\webapps\soap\WEB-INF\classes\Stock>java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter undeploy urn:StockService
C:\Tomcat 5.0\webapps\soap\WEB-INF\classes\Stock>

```

รูปที่ 4.7 หน้าจอแสดงการ Undeploy บริการ



```

C:\Tomcat 5.0\webapps\soap\WEB-INF\classes\Stock>java org.apache.soap.server.ServiceManagerClient http://localhost:8080/soap/servlet/rpcrouter list
Deployed Services:
urn:StockService
C:\Tomcat 5.0\webapps\soap\WEB-INF\classes\Stock>_

```

รูปที่ 4.8 หน้าจอแสดงผลว่ามีบริการใดให้บริการอยู่บ้างหลังการ Undeploy

เมื่อทำครบทุกขั้นตอนจึงจะสามารถเป็น Web Service ในเครื่องทางเซิร์ฟเวอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การพัฒนาระบบให้เป็น Web Service ฟังก์ชัน ไลอ้อน

ทางฟังก์ชัน ไลอ้อน จะต้องทำการ เพิ่มเพื่อให้สามารถเรียกใช้ บริการจากผู้ให้บริการได้ ดังตัวอย่าง

```
//Build the call.
org.apache.soap.rpc.Call call = new org.apache.soap.rpc.Call ();

//This service uses standard SOAP encoding
String encodingStyleURI = org.apache.soap.Constants.NS_URI_SOAP_ENC;
call.setEncodingStyleURI(encodingStyleURI);

//Set the target URI
call.setTargetObjectURI ("urn:StockService");

//Set the method name to invoke
call.setMethodName ("search Product ");

//Create the parameter objects
Vector params = new Vector ();
params.addElement (new org.apache.soap.rpc.Parameter("value",
String.class, productId, null));

//Set the parameters
call.setParams (params);

//Invoke the service
call.invoke ( new java.net.URL ("http://localhost:5555/soap/servlet/rpcrouter"),
"");
```

จากตัวอย่างสามารถอธิบายส่วนประกอบต่างๆ ได้ดังนี้

เริ่มจากการสร้าง Object Call เพื่อขอเข้าใช้บริการ คือ

```
org.apache.soap.rpc.Call call = new org.apache.soap.rpc.Call ();
```

เพื่อเป็น Object ตัวกลางที่ใช้ในการติดต่อกับผู้ให้บริการ Web Service ผ่าน SOAP

```
String encodingStyleURI = org.apache.soap.Constants.NS_URI_SOAP_ENC;
```

```
call.setEncodingStyleURI(encodingStyleURI);
```

เป็นลักษณะการ encode ข้อมูลที่ต้องการจะส่งผ่าน SOAP

```
call.setTargetObjectURI ("urn:StockService");
```

เป็นส่วนให้ผู้ขอใช้บริการกำหนดชื่อของ Web Service ที่ตนเองต้องการร้องขอ

```
call.setMethodName ("search Product");
```

เป็นการกำหนดชื่อของ method ที่ผู้ขอใช้บริการต้องการเรียกใช้บริการ

```
Vector params = new Vector ();
```

```
params.addElement (new org.apache.soap.rpc.Parameter("value", String.class,  
productId, null));
```

เป็นการกำหนด parameter ที่จะส่งไปให้ผู้ให้บริการ Web Service

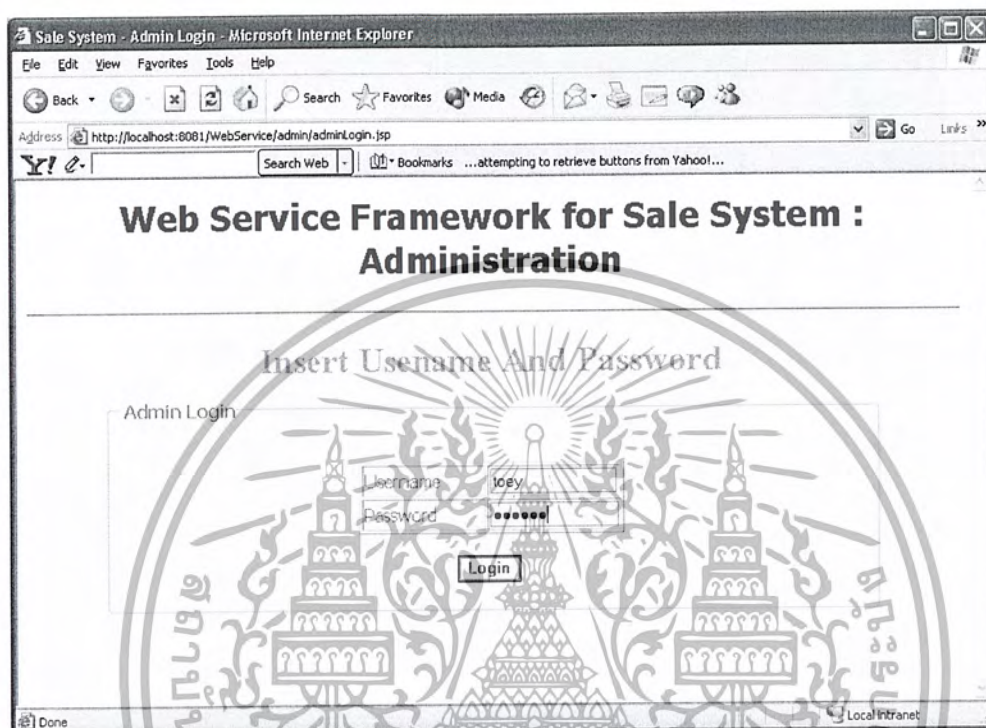
```
call.setParams (params); เา parameter ที่เรากำหนดได้ใน object call, call.invoke ( new  
java.net.URL("http://localhost:8080/soap/servlet/rpcrouter"), "");
```

เป็นการสั่งให้ทำการติดต่อกับผู้ให้บริการ Web Service

4.3 การใช้งานโปรแกรม

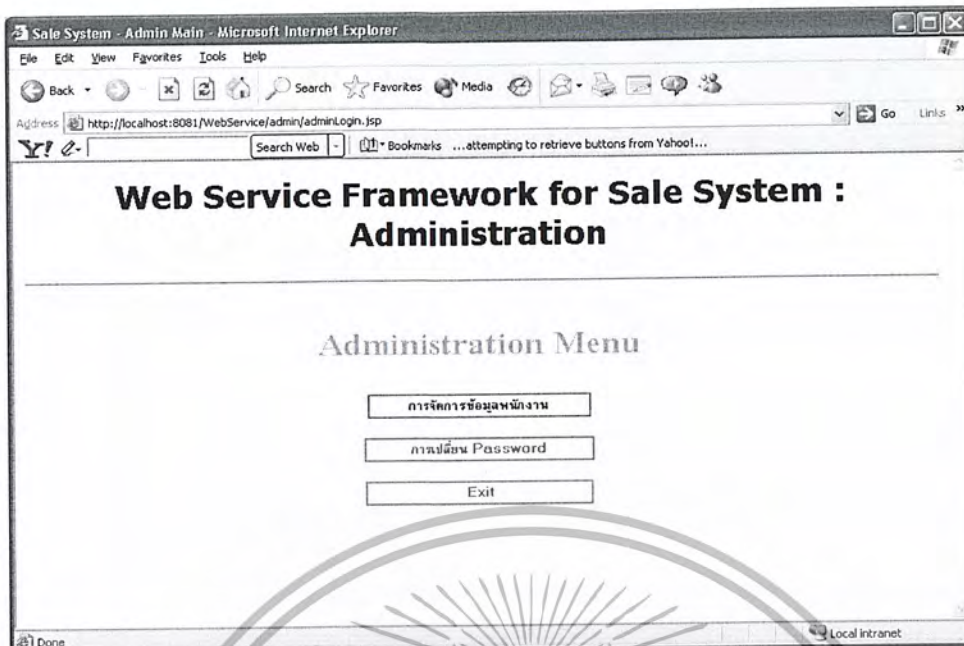
4.3.1 การจัดการข้อมูลพนักงาน

คือ การที่ผู้ดูแลระบบทำการเพิ่มข้อมูลพนักงานใหม่ และระบบจะทำการสร้าง password และ รหัสประจำตัวให้โดยอัตโนมัติ ดังนี้



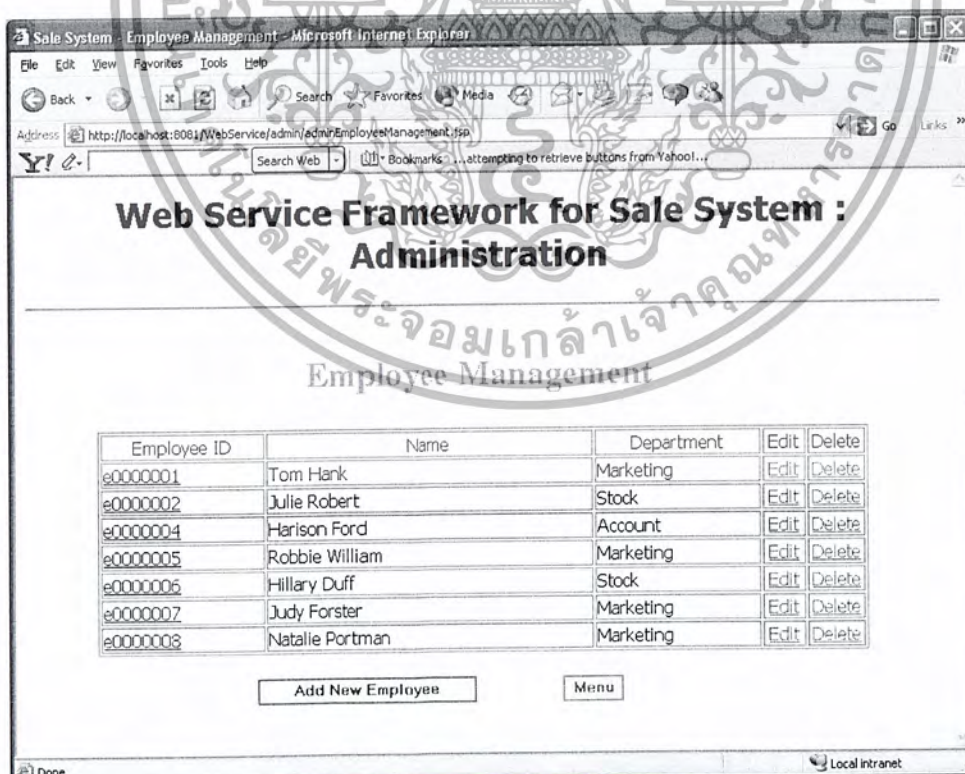
รูปที่ 4.9 หน้าจอ login ของผู้ดูแลระบบ

เมื่อใส่ username และ password แล้วก็สามารถเข้าใช้ระบบฝ่ายผู้ดูแลระบบได้ โดยปรากฏหน้าจอดังนี้



รูปที่ 4.10 หน้าจอแสดงรายการกิจกรรมหลักในฝ่ายดูแลระบบ

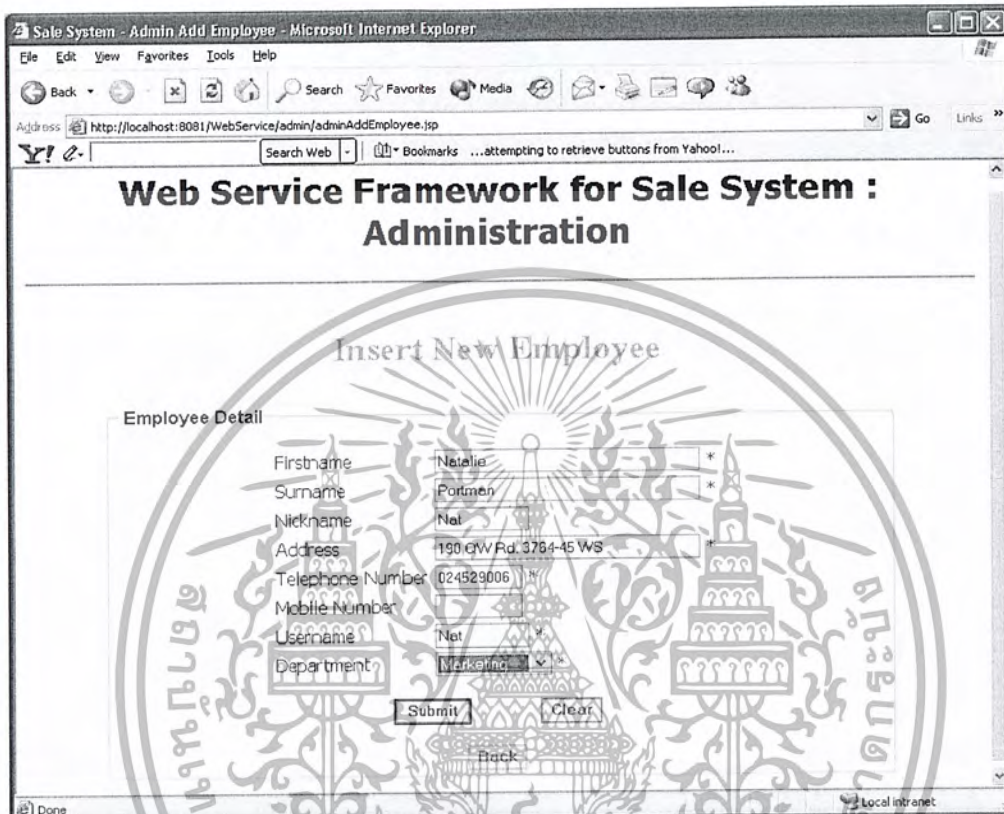
เมื่อเลือกการจัดการข้อมูลพนักงาน จะแสดงรายชื่อของพนักงานที่มีอยู่ในระบบทั้งหมด ดังนี้



รูปที่ 4.11 หน้าจอแสดงรายชื่อพนักงานทั้งหมดในระบบ

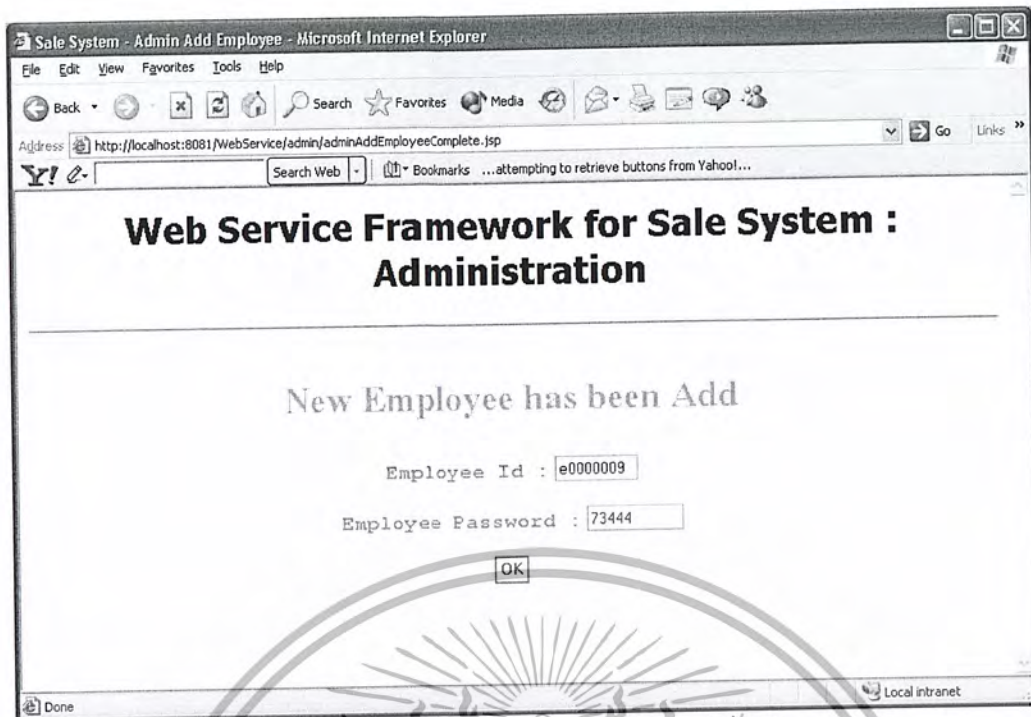
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่หน้าจอนี้ ผู้ดูแลระบบจะสามารถ เพิ่มข้อมูลพนักงานใหม่ , เปลี่ยนแปลงข้อมูล หรือลบข้อมูลของพนักงานแต่ละคนได้ และเมื่อต้องการเพิ่มข้อมูลพนักงานใหม่ กดปุ่ม Add New Employee จะปรากฏหน้าจอ ดังรูป



รูปที่ 4.12 หน้าจอกรอกข้อมูลพนักงานใหม่

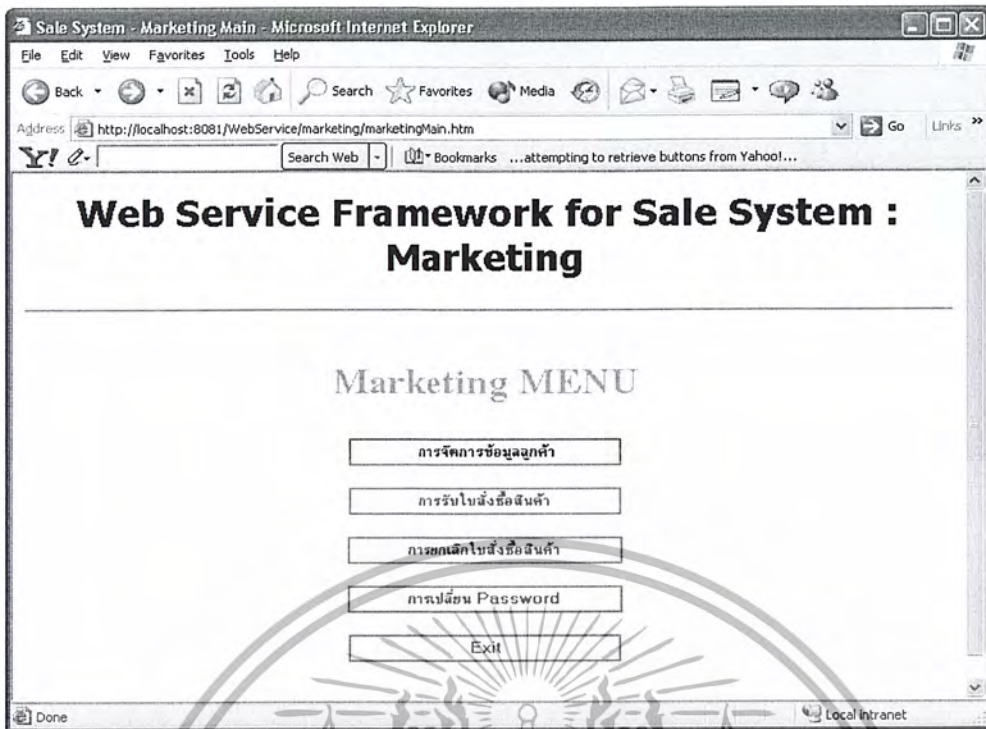
จากนั้นเลือกปุ่ม Submit เพื่อเข้าสู่หน้าจอถัดไป ซึ่งจะแสดงข้อมูลว่าได้มีการกรอกข้อมูลอะไรไปบ้าง แล้วจึงเข้าสู่หน้าจอแสดงรหัสประจำตัวพนักงาน และ password เริ่มต้นของพนักงานคนนั้น



รูปที่ 4.13 หน้าจอแสดงรหัสประจำตัวพนักงาน และ password เริ่มต้น

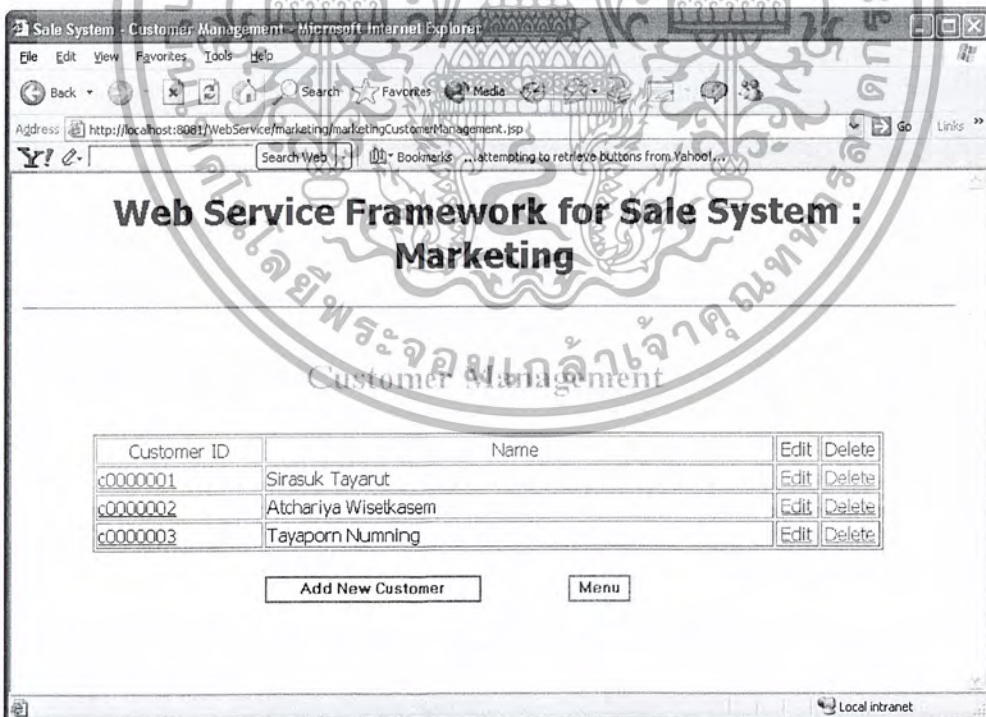
4.3.2 การใช้งานเมื่อลูกค้าต้องการสั่งซื้อสินค้า

คือ การที่พนักงานฝ่ายขายจะรับคำสั่งซื้อสินค้าจากลูกค้า ที่เป็นสมาชิก โดยลูกค้าคนใดที่ยังไม่ได้เป็นสมาชิกจะต้องทำการให้ข้อมูลกับพนักงาน เพื่อกรอกข้อมูลลูกค้าใหม่ จากนั้นจึงสามารถสั่งซื้อสินค้าได้ โดยพนักงานจะเลือกรหัสสินค้า และจำนวนสินค้าตามที่ลูกค้าต้องการ และระบบจะตรวจสอบโดยอัตโนมัติ ว่าในคลังสินค้ามีสินค้าเพียงพอที่จะส่งสินค้าหรือไม่ ถ้ามีสินค้าชนิดใดชนิดหนึ่งไม่เพียงพอ ระบบจะทำการยกเลิก รายการสั่งซื้อสินค้านั้น แต่ถ้าระบบมีสินค้าเพียงพอก็จะทำการบันทึกข้อมูลลงในฐานข้อมูลต่อไป ดังหน้าจอต่อไปนี้



รูปที่ 4.14 หน้าจอหลักของฝ่ายขาย

จากนั้นเลือก ปุ่ม การจัดการข้อมูลลูกค้าเพื่อเพิ่มข้อมูลลูกค้าใหม่



รูปที่ 4.15 หน้าจอแสดงรายชื่อลูกค้าทั้งหมด

จากนั้นกด Add New Customer เพื่อกรอกรายละเอียดของลูกค้าใหม่ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Service Framework for Sale System :
Marketing

New Customer

New Customer Detail

Firstname *

Surname *

Address *

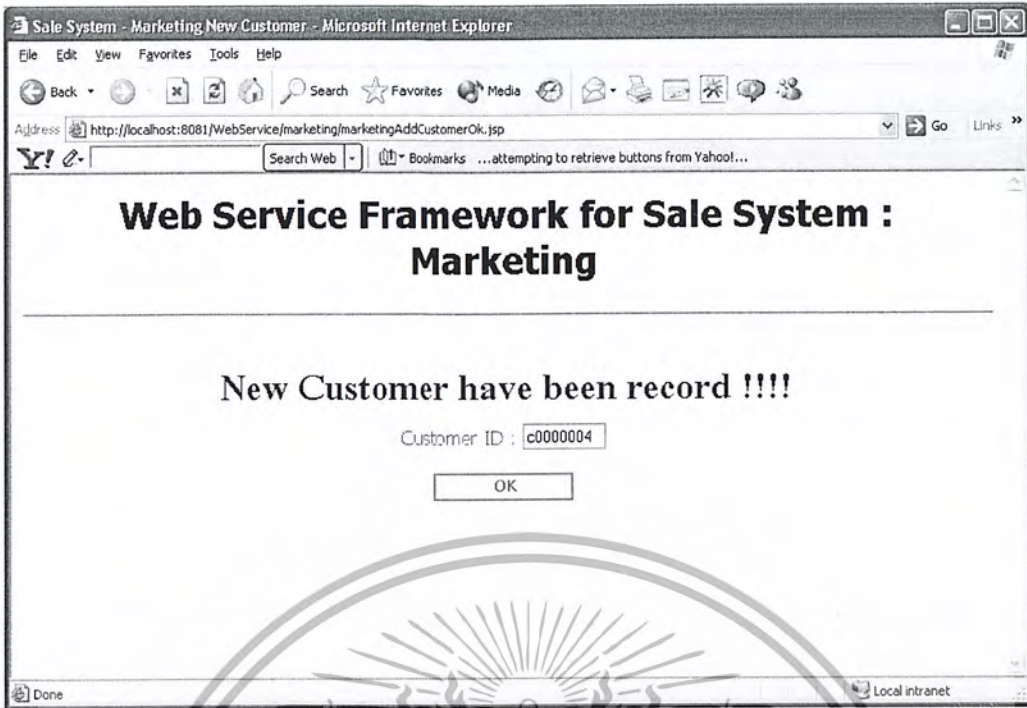
Telephone Number *

Fax Number

Due

รูปที่ 4.16 หน้าจอกรอกข้อมูลลูกค้าใหม่

เมื่อกรอกข้อมูลเรียบร้อยแล้วทำการคลิกปุ่ม Submit เพื่อบันทึกข้อมูล จากนั้นระบบจะทำการแสดงข้อความที่ได้กรอกไปในหน้าจอถัดไป ซึ่งถ้าตรวจสอบดูแล้วพบว่ากรอกข้อมูลผิดพลาดก็สามารถแก้ไขได้ หลังจากนั้นระบบจะแสดงรหัสประจำตัวลูกค้าดังนี้



รูปที่ 4.17 หน้าจอแสดงการบันทึกข้อมูลลูกค้าใหม่

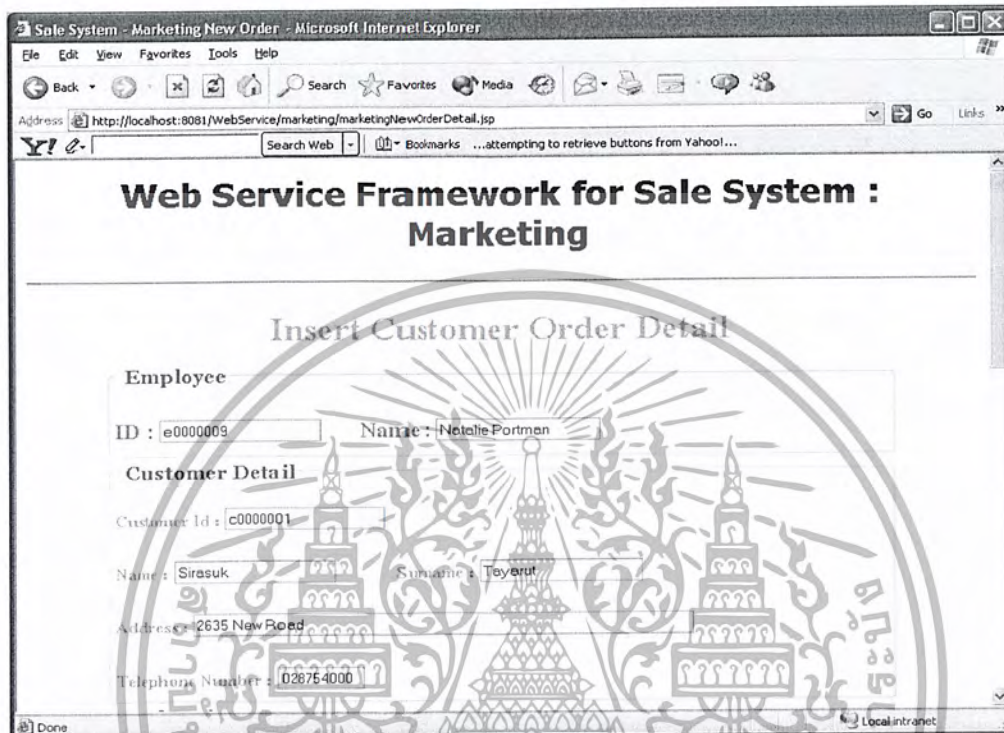
เมื่อบันทึกข้อมูลลูกค้าใหม่แล้วก็สามารถ สั่งซื้อสินค้าได้ โดยที่หน้ากิจกรรมหลัก เลือก การรับสั่งซื้อสินค้า จะปรากฏหน้าจอ



รูปที่ 4.18 หน้าจอเลือกรหัสลูกค้าเพื่อทำการสั่งซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือกได้แล้วจึงกดปุ่ม OK เพื่อสั่งซื้อสินค้าต่อไป โดยในส่วนบนจะแสดงรายละเอียดของ รหัส และชื่อของพนักงานที่ทำการรับ ใบสั่งซื้อสินค้า รวมถึง รายละเอียดของลูกค้ารายนั้น ที่ต้องการสั่งซื้อสินค้า ดังรูป



รูปที่ 4.19 หน้าจอแสดงรายละเอียดของพนักงานและลูกค้าที่ต้องการสั่งซื้อสินค้า

และส่วนล่างจะเป็นส่วนของการสั่งซื้อสินค้า โดยพนักงานฝ่ายขายจะเลือกรายการสินค้า และ กรอกจำนวนสินค้าที่ลูกค้าต้องการ ตั้งแต่ละชนิดและเสือกปุ่ม Order เพื่อดำเนินการต่อไป

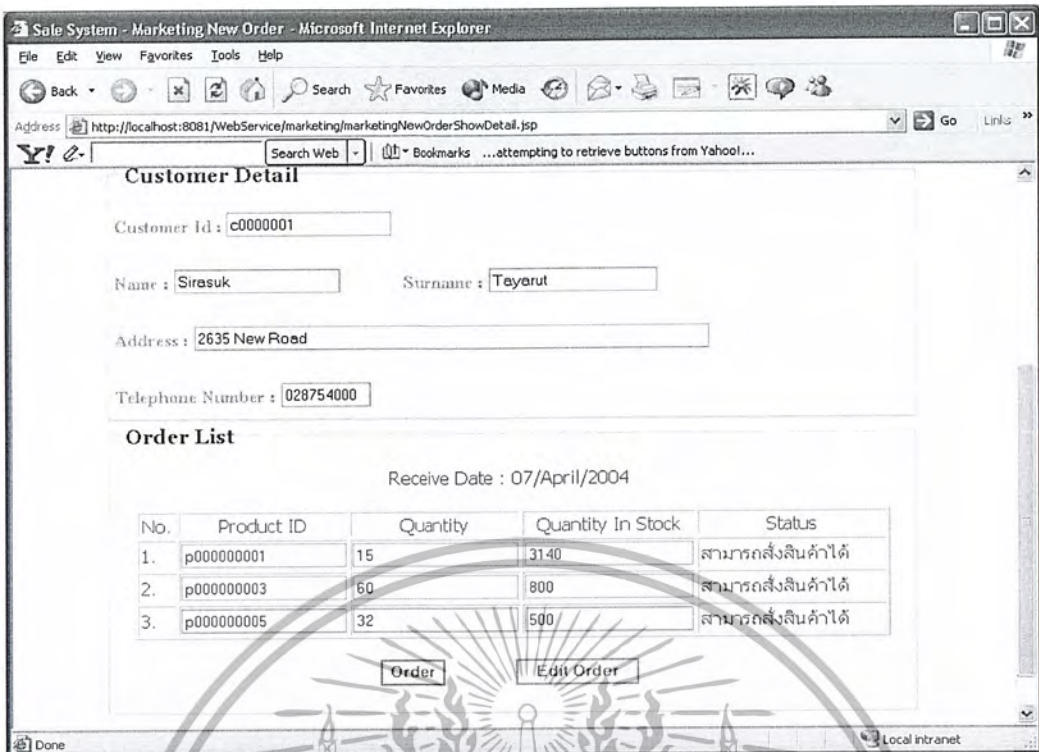
Order List

Send Date : 7 / April / 2004

No.	Product	Quantity
1.	p000000001 - Finding Nemo	15
2.	p000000003 - The Phoneboot	60
3.	p000000005 - You got mail	32
4.	Choose One	
5.	Choose One	
6.	Choose One	
7.	Choose One	
8.	Choose One	
9.	Choose One	
10.	Choose One	
11.	Choose One	
12.	Choose One	

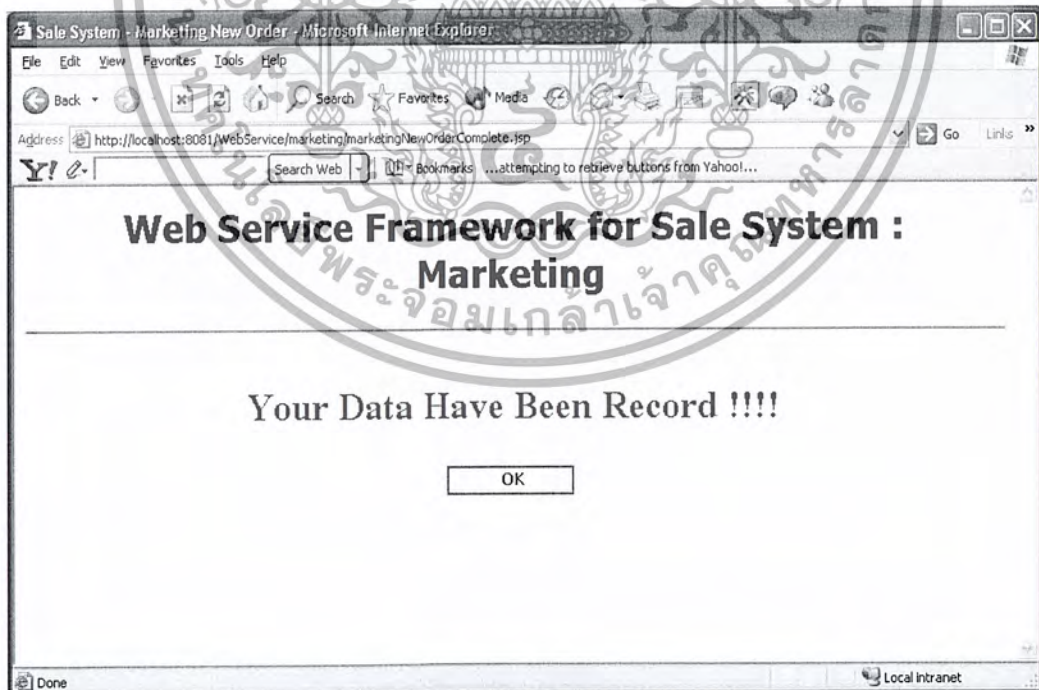
รูปที่ 4.20 หน้าจอแสดงรายการสินค้าและจำนวนที่ลูกค้าต้องการสั่งซื้อ

หลังจากนั้นเลือกปุ่ม Order เพื่อสั่งซื้อสินค้า แต่ถ้าต้องการกรอกข้อมูลใหม่ก็สามารถเลือกปุ่ม Clear ได้เช่นกัน จากนั้น จะแสดงหน้าจอว่ามีสินค้าแต่ละชนิดตามที่ลูกค้าสั่งเพียงพอหรือไม่ ดังรูป



รูปที่ 4.21 หน้าจอแสดงให้ทราบว่ามีสินค้าในคลังเพียงพอหรือไม่

ลูกค้าจะสั่งซื้อสินค้าได้ก็ต่อเมื่อ สินค้าทุกชนิดมีสินค้าพอที่จะส่งได้ จากนั้นเลือกปุ่ม Order



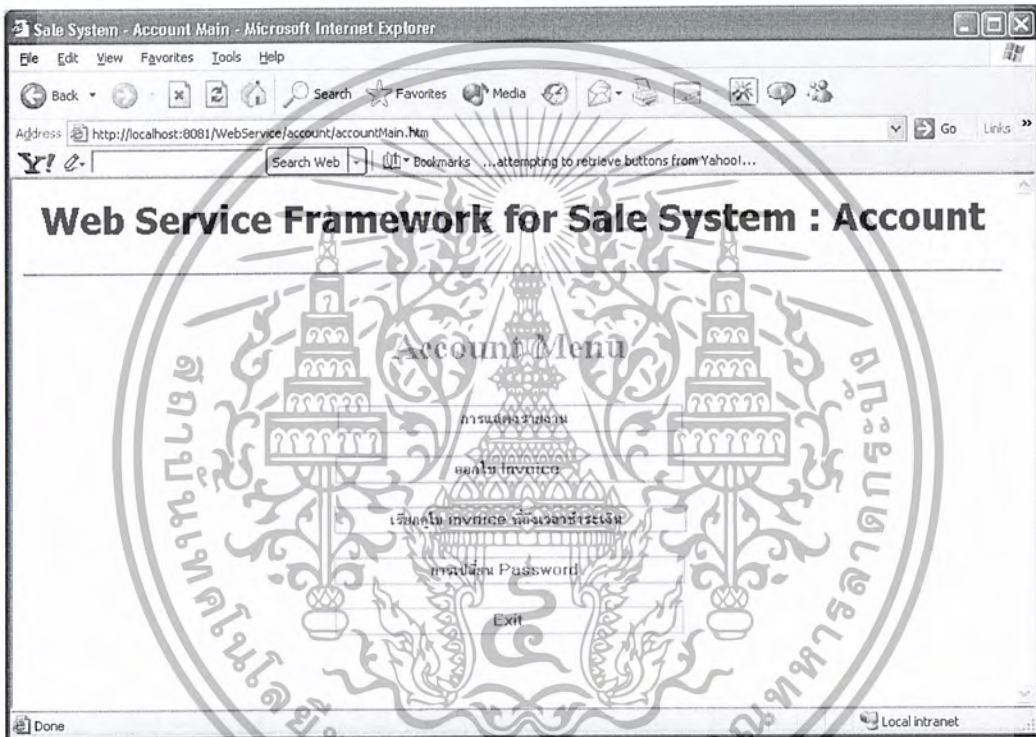
รูปที่ 4.22 หน้าจอแสดงการบันทึกข้อมูลลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะปรากฏข้อความว่าข้อมูลได้ถูกบันทึกลงในฐานข้อมูลเรียบร้อยแล้ว เป็นอันเสร็จ
กระบวนการสั่งซื้อสินค้า

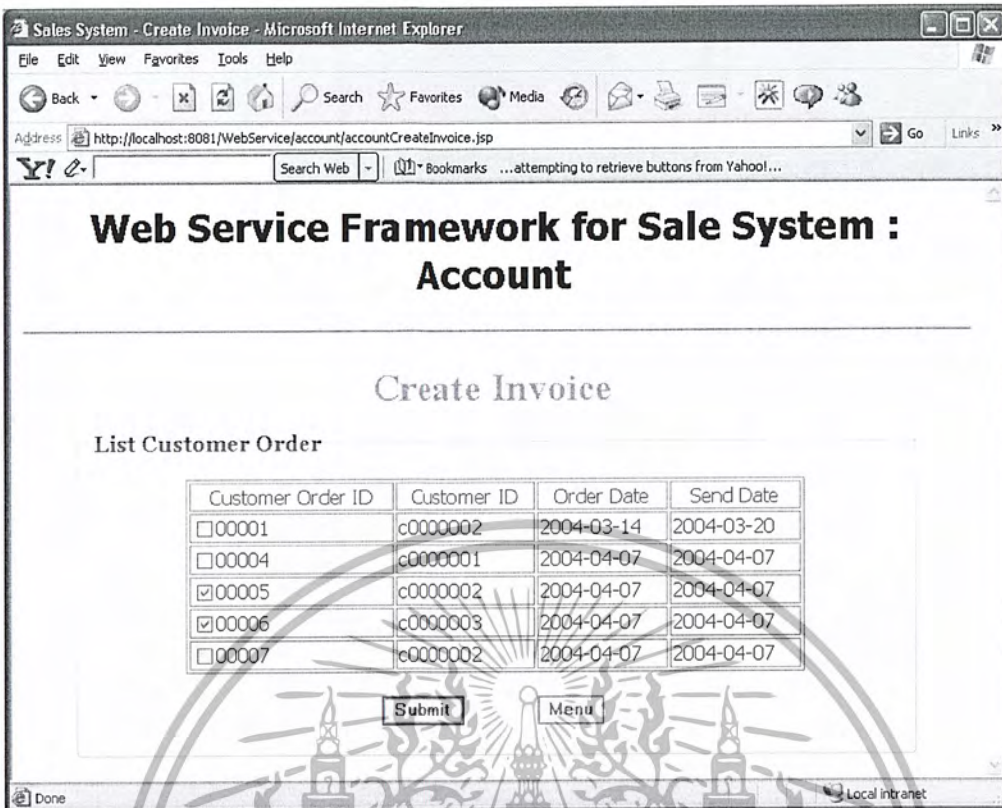
4.3.3 การใช้งานเมื่อต้องการออกไป Invoice และตัดจำนวนสินค้าในคลัง

คือ การที่พนักงานฝ่ายบัญชี จะทำการเลือกว่าใบสั่งซื้อสินค้าใบใดบ้างที่จะต้อง ออกไป Invoice และตัดจำนวนสินค้าในคลังสินค้า โดยในการเลือกสามารถเลือก ออกไป Invoice ตาม ใบสั่งซื้อสินค้าได้หลายใบในครั้งเดียวกัน ที่หน้าจอกิจกรรมหลักของฝ่ายบัญชีเลือกปุ่มออกไป Invoice



รูปที่ 4.23 หน้าจอกิจกรรมหลักของฝ่ายบัญชี

ระบบจะแสดงหน้าจอที่รายงานว่าจะขณะนี้มิใบสั่งซื้อสินค้าใบใดบ้างที่ยังไม่ได้ ทำการออกไป Invoice โดยพนักงานจะสามารถเลือกออกไป Invoice ได้หลายใบในคราวเดียวกัน



รูปที่ 4.24 หน้าจอแสดงใบสั่งซื้อสินค้าเพื่อการออกไป Invoice

เมื่อเลือกได้แล้วว่าต้องการออก Invoice จากใบสั่งซื้อสินค้าใด ก็ทำการกดปุ่ม Submit จากนั้น ระบบจะทำการแสดงรายละเอียดของใบสั่งซื้อสินค้าแต่ละใบดังรูป

Web Service Framework for Sale System : Account

Show Invoice Detail

Customer Order No.00005

Employee
 ID : e0000009 Name : nAccount sAccount

Customer
 ID : c0000002 Name : Atchariya Wisetkasem
 Address : 19 Jelearnkrung

Product Order

NO.	Product ID	Product Name	Amount
1	p000000001	Finding Nemo	300
2	p000000006	abc	40
		Total Amount	340
		Total Price	140000.0

รูปที่ 4.25 หน้าจอแสดงรายงานใบ Invoice จากใบสั่งซื้อสินค้า

Customer Order No.00006

Employee
 ID : e0000009 Name : nAccount sAccount

Customer
 ID : c0000003 Name : Teyaporn Numning
 Address : 123 Bangmod

Product Order

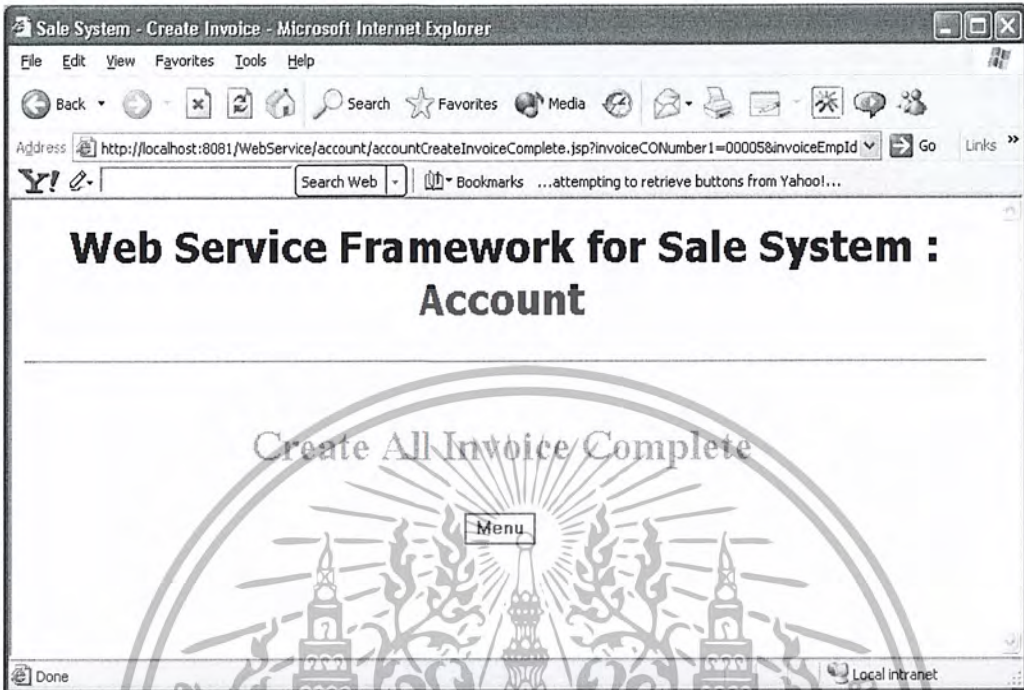
NO.	Product ID	Product Name	Amount
1	p000000002	Peter Pan	60
2	p000000005	You got mail	45
		Total Amount	105
		Total Price	36000.0

Create Edit

รูปที่ 4.26 หน้าจอแสดงรายงานใบ Invoice จากใบสั่งซื้อสินค้า (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

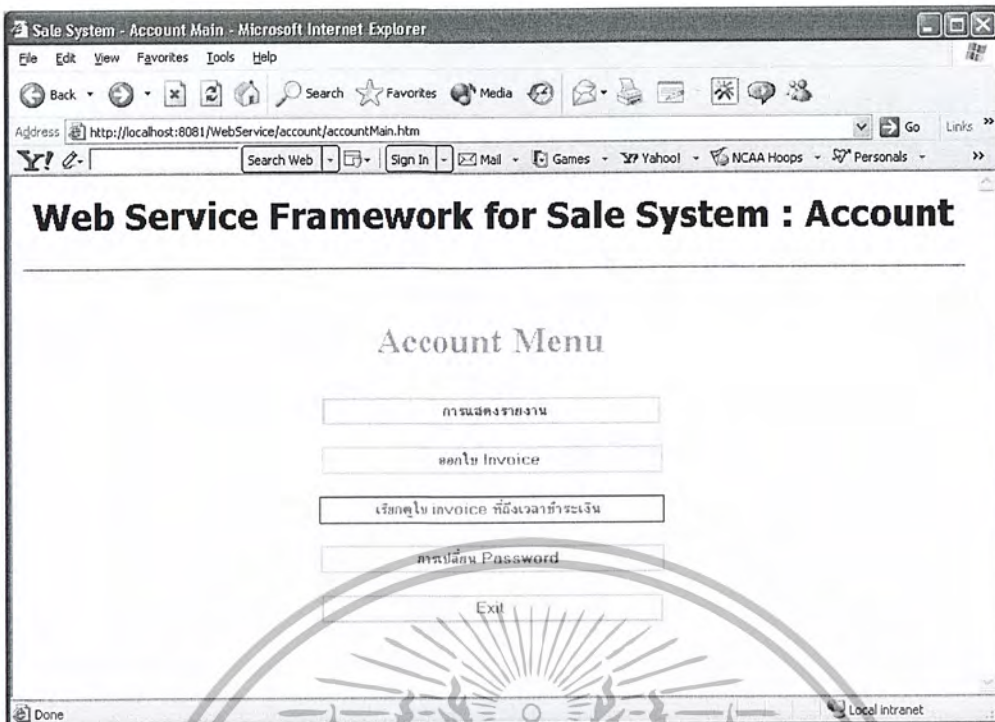
เลือกปุ่ม Create เพื่อสร้างใบ Invoice หลังจากนั้นระบบจะทำการ เก็บข้อมูลลงในฐานข้อมูล โดยแสดงผล ดังนี้



รูปที่ 4.27 หน้าจอแสดงการสร้างใบ Invoice สำเร็จแล้ว

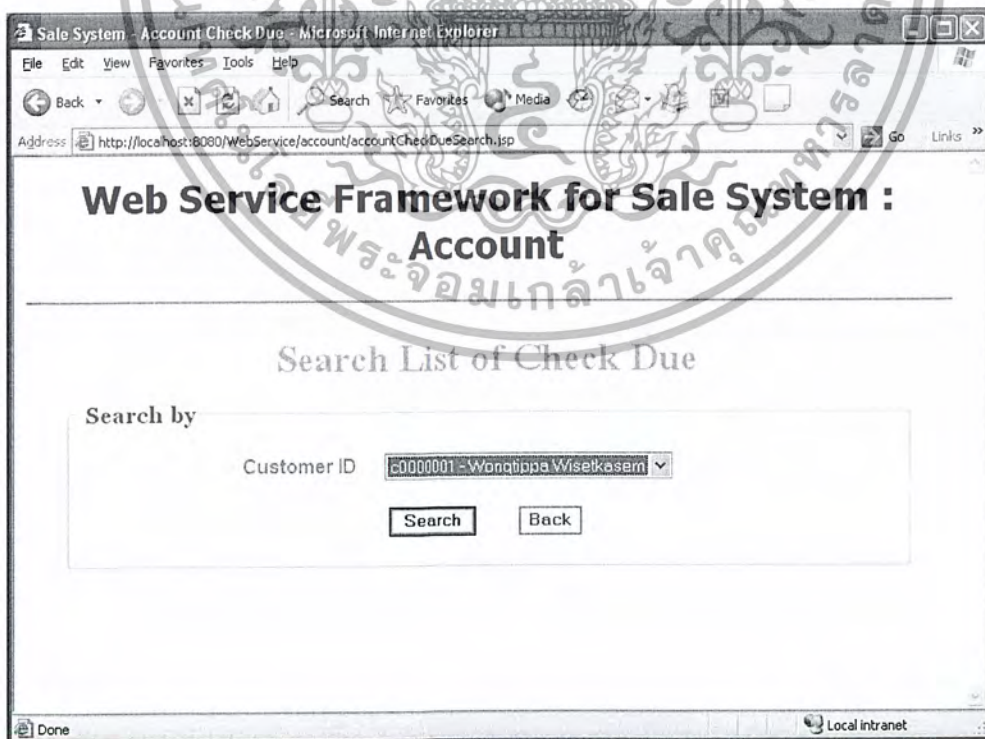
4.3.4 การแสดงรายงานใบ Invoice ที่ใกล้ถึงเวลาชำระค่าสินค้า

เป็นการเลือกใบ Invoice ว่าใบใดบ้างที่ถึงกำหนดเวลาชำระเงินให้กับทางบริษัทแล้ว โดยพนักงานจะเลือกดูตามรายชื่อ และ รหัสประจำตัวลูกค้าแต่ละราย จากนั้นระบบจะแสดงรายการใบ Invoice ของลูกค้ารายนั้นทั้งหมด และพนักงานสามารถตรวจสอบได้ว่ามีใบ Invoice ใดบ้างที่ถึงกำหนดชำระเงินแล้ว โดยดูได้จากการแสดงสถานะในช่องสุดท้ายของใบ Invoice ถ้าหากขึ้นว่า Not Due หมายถึง ยังไม่ครบกำหนดจ่ายเงิน แต่ถ้าขึ้น Due หมายถึง ใกล้ครบกำหนดจ่ายเงินแล้ว



รูปที่ 4.28 หน้าจอแสดงกิจกรรมหลักของฝ่ายบัญชี

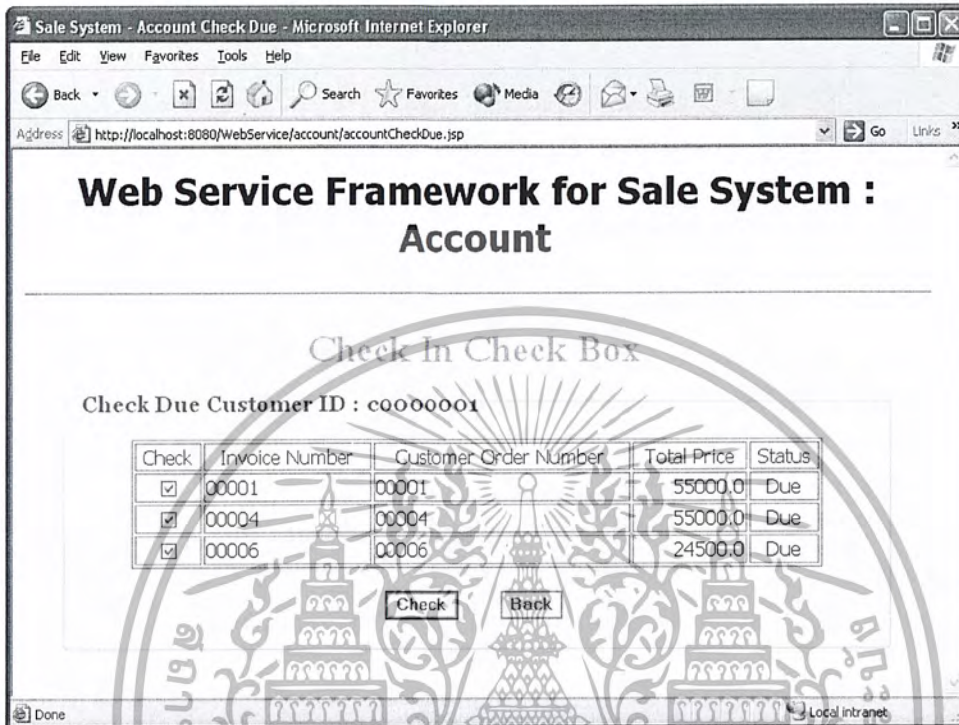
จากนั้นเลือกปุ่ม เรียกดูใบ Invoice ที่ถึงเวลาชำระเงิน จะแสดงหน้าจอต่อมา ให้เลือกว่าจะตรวจสอบของลูกค้านั้นใด



รูปที่ 4.29 หน้าจอแสดงรหัสประจำตัว และรายชื่อลูกค้าเพื่อตรวจสอบการชำระเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือกลูกค้าแล้วก็สามารถคลิกปุ่ม Search เพื่อหาข้อมูลของลูกค้า จากนั้นจะปรากฏหน้าจอที่แสดงรายการว่าลูกค้ารายนี้มีใบ Invoice อะไรบ้าง ทั้งที่ยังไม่ใกล้ถึงเวลาชำระค่าสินค้า และใกล้ถึงเวลาชำระค่าสินค้าแล้ว



รูปที่ 4.30 หน้าจอแสดงรายการใบ Invoice ที่เตรียมชำระค่าสินค้า

หลังจากเลือกรายการใบ Invoice ที่ถึงเวลาชำระค่าสินค้าแล้ว ในช่อง check box ด้านหน้าก็สามารถคลิกปุ่ม Check เพื่อดำเนินการต่อไป

**Web Service Framework for Sale System :
Account**

Report Check Due

List

Invoice Number	Customer Order Number	Amount
00001	00001	55000.0
00004	00004	55000.0
00006	00006	24500.0

Total 134500.0 Bath

รูปที่ 4.31 หน้าจอแสดงรายละเอียดของใบ Invoice ที่เลือก

และเมื่อเลือกปุ่ม OK ก็จะกลับสู่การทำงานในหน้าหน้าเมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทำงานของโปรแกรม และข้อเสนอแนะ

5.1 สรุปผลการทำงานของโปรแกรม

การทำงานของ Framework ระบบการขายโดยใช้ Web Service จะทำได้โดย ผู้ให้บริการ (เครื่องด้าน เซิร์ฟเวอร์) จะทำการเขียน WSDL เพื่อเป็นการประกาศบริการระบบการขายของตนเองไว้ที่ UDDI ที่อนุญาต จากนั้นผู้ขอใช้บริการ (เครื่องทางไคลแอนท์) จะเข้ามาค้นหาบริการที่ตนเองสนใจใน UDDI และทำการติดต่อไปยังผู้ให้บริการโดยตรง ผู้ขอใช้บริการจะสามารถเรียกใช้ Component ของทางฝั่งผู้ให้บริการได้ โดยจะต้องตั้งค่าเริ่มต้นของฐานข้อมูล ตามที่ Component กำหนด ทั้งจำนวนตาราง, ประเภทของแต่ละตาราง, จำนวน Attribute และประเภทของ Attribute ในแต่ละตาราง ผู้ขอใช้บริการจะสามารถนำ Component บางส่วนของระบบการขายที่ให้บริการอยู่ ไปใช้ร่วมกับระบบเดิมที่ตนเองมีอยู่ หรือ จะนำ Component ทั้งหมดของ Framework ไปใช้งานทั้งระบบก็ได้ เมื่อผู้ขอใช้บริการต้องการเรียกใช้ส่วนใดจากผู้ให้บริการก็จะส่งพารามิเตอร์ ไปยังผู้ให้บริการโดยส่งผ่าน Protocol SOAP หลังจากนั้นผู้ให้บริการได้รับแล้วก็จะนำไปประมวลผล และจะนำผลลัพธ์ที่ได้ ส่งกลับมายังผู้ขอใช้บริการโดยอาศัย Protocol SOAP เช่นกัน ในส่วนของ Framework ระบบการขายแบ่งออกเป็น 5 ส่วน ซึ่งจะอธิบายการทำงานของแต่ละ Component อย่างละเอียด ดังต่อไปนี้

1. ผู้ดูแลระบบ

การดำเนินการใน Component นี้จะสามารถแบ่งได้เป็น 3 กิจกรรม ดังต่อไปนี้

- การจัดการข้อมูลพนักงาน คือ การนำข้อมูลพนักงานบันทึกลงฐานข้อมูลเมื่อมีพนักงานใหม่ พร้อมทั้งให้ Password ที่ได้จากระบบแก่พนักงาน เพื่อการเข้าใช้ระบบของพนักงานแต่ละคน, การแก้ไขข้อมูลพนักงานเมื่อข้อมูลที่บันทึกลงในฐานข้อมูลไม่ถูกต้อง และการลบข้อมูลพนักงาน
- การเปลี่ยน Password คือ การเปลี่ยนแปลง Password ของผู้ดูแลระบบเอง

2. การอนุญาตให้เข้าใช้ระบบ

จะเป็น Component ในการตรวจสอบสิทธิการเข้าใช้งานระบบของพนักงานทุกฝ่าย โดยเมื่อพนักงานคนใดต้องการเข้าใช้งานระบบ จะต้องใส่ Username และ Password ของตนเองก่อนทุกครั้ง ระบบจะทำการตรวจสอบสิทธิ และหากพบว่า Username และ Password นั้นไม่ถูกต้องก็จะไม่อนุญาตให้ผู้นั้นเข้าใช้งาน แต่ถ้าระบบตรวจสอบแล้วว่า Username และ Password นั้นถูกต้องก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการแสดงโปรแกรมในฝ่ายที่พนักงานคนนั้นสังกัดอยู่ขึ้นมาทำงาน ดังนั้นพนักงานจะไม่มีสิทธิเข้าใช้งาน โปรแกรมของฝ่ายอื่น ที่ตนเองไม่ได้สังกัด

3. ฝ่ายขาย

การดำเนินการในฝ่ายขายจะแบ่งออกเป็น 4 กิจกรรม ดังนี้

- การจัดการเกี่ยวกับลูกค้า โดยสามารถเพิ่มลูกค้าใหม่โดยระบบจะทำการกำหนดรหัสกับลูกค้าใหม่โดยอัตโนมัติ ให้ต่อจากลูกค้ารายสุดท้ายในฐานข้อมูล, แก้ไขข้อมูลลูกค้าเดิม และการลบรายชื่อลูกค้าออก
- การรับใบสั่งซื้อสินค้า คือ การรับคำสั่งซื้อจากลูกค้าโดยระบบสามารถตรวจสอบได้ทันทีว่า จากใบสั่งซื้อใบนั้น มีสินค้าคงเหลือในคลังข้อมูลเพียงพอหรือไม่ ถ้าไม่พอระบบจะทำการยกเลิกใบสั่งซื้อสินค้าใบนั้น แต่ถ้ามีสินค้าเพียงพอก็จะบันทึกลงฐานข้อมูลถัดไป
- การยกเลิกใบสั่งซื้อสินค้า โดยระบบจะอนุญาตให้มีการยกเลิกในกรณีเดียวเท่านั้นคือ ใบสั่งซื้อสินค้าใบนั้นยังไม่ได้ทำการออกใบ Invoice
- การเปลี่ยน Password ของพนักงานฝ่ายขาย

4. ฝ่ายบัญชี

การดำเนินการในฝ่ายบัญชีจะแบ่งออกเป็น 3 กิจกรรม ดังนี้

- การแสดงรายงาน ซึ่งประกอบด้วยรายงานย่อยจำนวน 4 รายงาน
 - 1) แสดงรายงานใบสั่งซื้อสินค้าทั้งหมด สามารถเลือกช่วงเวลา , เลือกหมายเลขประจำตัวลูกค้า หรือ เลือกหมายเลขใบ Invoice เพื่อแสดงรายงานได้ โดยจะแสดงรายงานใบสั่งซื้อสินค้าตามที่ค้นหาที่เข้ามาในระบบทั้งหมด รายงานจะแสดงวันที่ทำการสั่งใบสั่งซื้อสินค้านี้ , รหัสประจำตัวลูกค้า , ยอดเงินรวมของใบสั่งซื้อสินค้า และสถานะ โดยสามารถแบ่งออกได้เป็น 4 ประเภท ตามสถานะภาพของใบสั่งซื้อสินค้านั้น คือ ใบสั่งซื้อสินค้าได้ออกใบ Invoice แล้วและชำระเงินแล้ว, ใบสั่งซื้อสินค้าที่ออกใบ Invoice แล้วแต่ยังไม่ได้ทำการชำระเงิน, ใบสั่งซื้อสินค้าที่ยังไม่ได้ออกใบ Invoice และใบสั่งซื้อสินค้าที่ถูกยกเลิก โดยถ้าต้องการทราบรายละเอียดของใบสั่งซื้อสินค้าใบใด ก็สามารถเลือกเพื่อดูรายละเอียดได้จากรายงานนี้ ซึ่งรายละเอียดของแต่ละใบสั่งซื้อสินค้าจะแสดงรหัสสินค้า , ชื่อสินค้า , ราคาสินค้าต่อหน่วย , จำนวนที่สั่งซื้อ และราคารวมทั้งหมด
 - 2) แสดงรายงานใบ Invoice โดยสามารถเลือกดูได้ ตามช่วงเวลา , ตามหมายเลขประจำตัวลูกค้า หรือ ตามหมายเลขใบ Invoice จากนั้นรายงานจะแสดงข้อมูลว่า เป็นใบที่ออกตามใบสั่งซื้อสินค้าใบใด , ลูกค้าที่ทำการสั่งมีหมายเลขประจำตัวใด , ราคาสินค้าทั้งหมด และสถานะของใบ Invoice นั้น โดยสามารถเลือกดูรายละเอียด ของรายการสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามใบ Invoice ได้โดยเลือกที่ใบสั่งซื้อสินค้าที่ใบ Invoice นั้นออกตาม เนื่องจากในระบบการจำหน่ายสินค้านี้ ใบ Invoice ทุกใบ จะออกตามใบสั่งซื้อสินค้าทั้งใบ ใบสั่งซื้อสินค้าหนึ่งใบ ต่อใบ Invoice หนึ่งใบ ไม่มีกรณีที่ใบสั่งซื้อสินค้า หนึ่งใบจะสามารถออกใบ Invoice ได้หลายใบ

- 3) การแสดงรายงานลูกค้า จะแสดงรายละเอียดของลูกค้าทั้งหมดในฐานข้อมูล โดยแสดงหมายเลขประจำตัวลูกค้า, ชื่อ, นามสกุล, ที่อยู่ และเบอร์โทรศัพท์ที่สามารถติดต่อได้
- 4) การแสดงรายงานสินค้า จะแสดงรายงานสินค้าทั้งหมดในคลังสินค้า โดยรายงานจะแสดง รหัสสินค้า, ชื่อสินค้า, ราคาต่อหน่วยของสินค้าชนิดนั้น และ จำนวนที่มีในคลังสินค้าขณะนี้

- การเรียกเก็บเงินลูกค้าจากใบ Invoice ที่ใกล้ถึงเวลาชำระเงิน คือ การที่พนักงานจะสามารถตรวจสอบใบ Invoice ได้ว่า ใบ Invoice ใบใดสมควรแก่เวลาชำระค่าสินค้าแล้ว ซึ่งระบบจะทำการแสดงผลใบ Invoice ที่ใกล้ถึงเวลาชำระค่าสินค้าออกมา
- การเปลี่ยน Password ของพนักงานในฝ่ายบัญชี

5. ฝ่ายจัดการสินค้า

การดำเนินการในฝ่ายการจัดการสินค้าแบ่งออกเป็น 5 กิจกรรม คือ

- การเพิ่มจำนวนสินค้าในคลัง คือ การเพิ่มจำนวนสินค้ารายการที่มีอยู่แล้วแต่ จำนวนสินค้าอาจจะไม่เพียงพอกับความต้องการของลูกค้า
- การเพิ่มรายการสินค้าใหม่ คือ การเพิ่มรายการสินค้าชนิดใหม่ที่ยังไม่เคยมีมาก่อน ซึ่งทางระบบจะกำหนดรหัสสินค้าให้โดยอัตโนมัติ และรหัสสินค้านั้นจะต่อจากรหัสสินค้าเดิมที่มีอยู่
- การแสดงรายงานรายการสินค้า คือ การแสดงรายงานที่ระบุรายละเอียดของสินค้า ประกอบด้วยรหัสสินค้า, ชื่อสินค้า และปริมาณที่คงเหลือในคลังสินค้าปัจจุบัน
- การออกใบ Invoice คือ การที่ระบบจะทำการตรวจสอบว่ามีใบสั่งซื้อสินค้าใบใดบ้างที่ยังไม่ได้ทำการออกใบ Invoice จะแสดงผลใบสั่งซื้อสินค้านั้นๆ ขึ้นมาเพื่อให้พนักงานทำการเลือกเพื่อออกใบ Invoice

การเปลี่ยน Password คือ หลังจากที่พนักงานได้รับ Password จากผู้ดูแลระบบแล้ว จะสามารถ Log in เข้าใช้งานโปรแกรมได้และเมื่อ ต้องการเปลี่ยน Password ก็สามารถทำได้

- การเปลี่ยน Password ของพนักงานฝ่ายจัดการสินค้า

5.2 ปัญหา และข้อจำกัดของปัญหาพิเศษ

ข้อจำกัดของปัญหาพิเศษเรื่องการพัฒนา Framework ระบบการขายบน Web Service มีดังนี้

1. ยังไม่สามารถนำ Framework นี้ไปใช้ได้กับทุกระบบการขาย
 2. เมื่อผู้ให้บริการ Web Service ต้องการประกาศการให้บริการของตน ต้องนำ WSDL ไปประกาศบน UDDI ของตัวแทนผู้ให้บริการรายอื่นก่อน เนื่องจากระบบไม่ได้สร้าง UDDI ของตนเอง
 3. ผู้ที่ต้องการนำ Framework ระบบการขายนี้ไปใช้ จำเป็นต้องอ้างอิงฐานข้อมูลตามที่ระบบกำหนดเท่านั้น ทั้งจำนวนตาราง, ประเภทของตาราง, จำนวนของ Attribute ในแต่ละตาราง และ ประเภทของ Attribute ในแต่ละตาราง เช่น Component ของฝ่ายขาย จะต้องเกี่ยวข้องกับ 3 ตารางคือ ตารางระบุรายละเอียดของลูกค้า, ตารางระบุรายละเอียดของใบสั่งซื้อสินค้า และตารางระบุรายละเอียดของสินค้า ซึ่งในส่วนนี้ผู้ใช้ไม่สามารถกำหนดให้อ้างถึงตารางอื่นได้ด้วยตนเอง
- การเรียกใช้ Framework ระบบการขายนี้จะต้องเรียกใช้ผ่านระบบอินเทอร์เน็ต เท่านั้น
4. ใบสั่งซื้อสินค้า หนึ่งใบ จะมีใบ Invoice ได้เพียงแค่ ใบเดียวเท่านั้น

5.3 ข้อเสนอแนะ

1. การนำไปใช้จริงควรเพิ่มในส่วนของการพิมพ์รายงาน หรือ เอกสารสำคัญออกทางเครื่องพิมพ์ได้ เพื่อเป็นหลักฐานในการดำเนินการใดๆ
2. เพิ่ม Component มากขึ้นเพื่อสามารถใช้ได้กับ ระบบการจำหน่ายสินค้าหลากหลายประเภทมากขึ้น

บรรณานุกรม

สาธิต ชัยวิวัฒน์ตระกูล. 2545. **เก่ง JSP ให้ครบสูตร**. พิมพ์ครั้งที่ 1. กรุงเทพฯ: H.N. Group
วันชัย แซ่เตีย และสิทธิชัย ประสานวงศ์. 2542. **สร้างเว็บเพจด้วย HTML 4**. กรุงเทพฯ: ซอฟท์เพรส
พันจันทร์ ธนวัฒน์เสถียร,สิทธิพัฒน์ จำนงศิลป์ และยุทธชัย รุจิรวิมล. **คู่มือการเรียนรู้และเทคนิคการ
สร้างเว็บเพจ Macromedia Dreamweaver MX ฉบับสมบูรณ์**. พิมพ์ครั้งที่ 2. กรุงเทพฯ:
ซัคเซส มีเดีย จำกัด

จิตเกษม พัฒนาศิริ. 2541. **เสริมแต่งโฮมเพจให้มีชีวิตชีวา ด้วย Java Script**. พิมพ์ครั้งที่ 1. กรุงเทพฯ:
วิดีส์ กรุ๊ป จำกัด

David Chappell & Tyler Jewell , **Java Web Service**,First Editing March 2002,ISDN: 0-596-00269-6
Kurt A.Gabrick & David B. Weiss, **Java 2EE and XML Development**, First Editing 2002,
ISDN: 1-931101-30-8, Publisher: Manning Publications Co.



ภาคผนวก ก.

คู่มือการติดตั้งโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

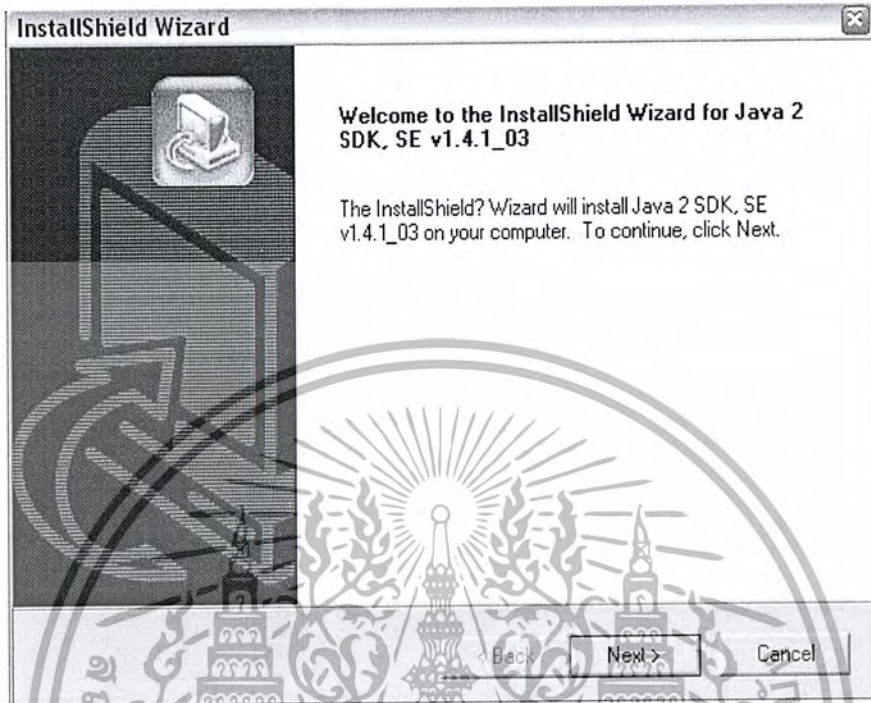
การติดตั้งและการปรับแต่งโปรแกรมที่ใช้ในระบบงาน

ขั้นตอนการติดตั้ง J2SDK

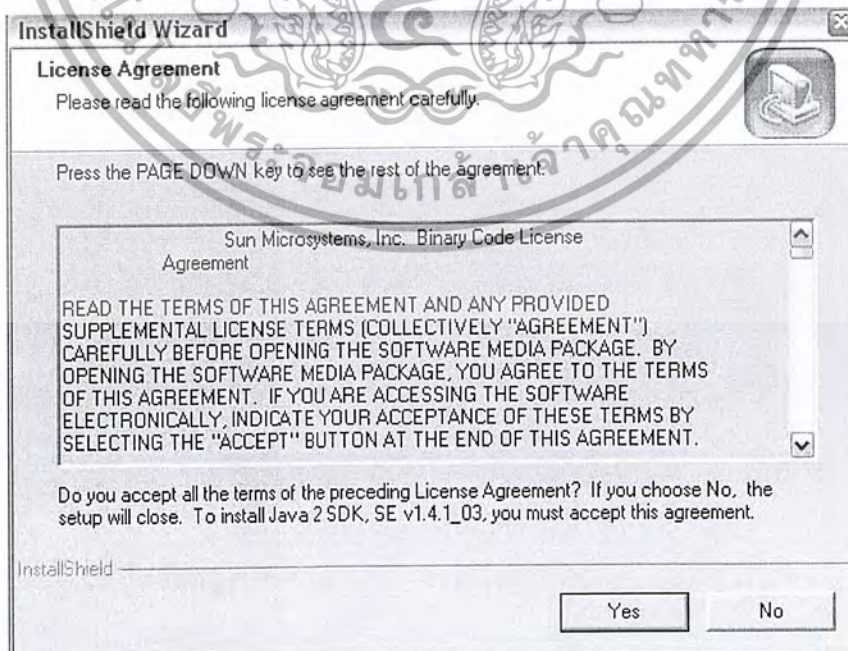
1. ดับเบิลคลิกที่ไฟล์ j2sdk-1_4_1_03-windows-i586.exe
2. โปรแกรมจะ extract ไฟล์เพื่อเตรียมการติดตั้ง ดังรูป



3. จากนั้นจะเข้าสู่หน้าจอแสดงข้อความต้อนรับการติดตั้งโปรแกรม ดังรูป

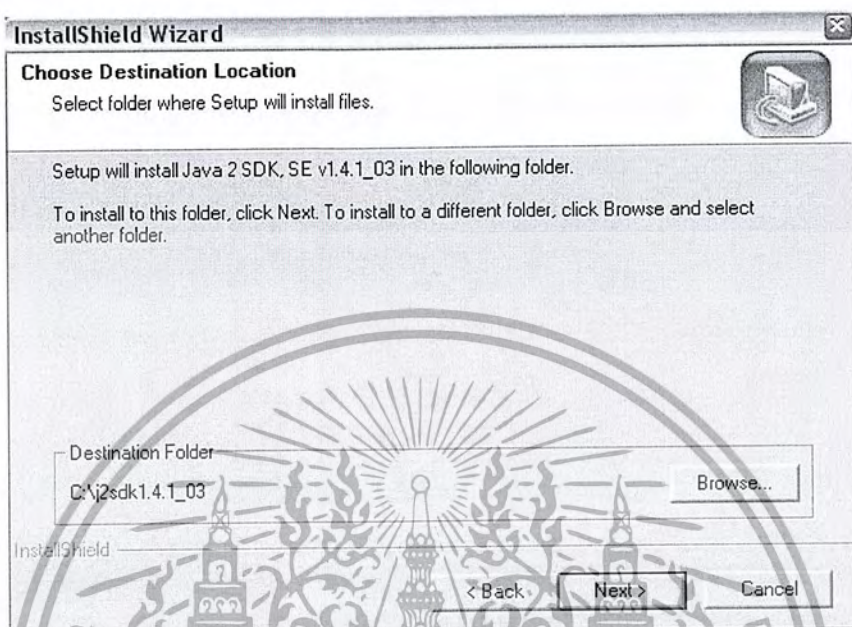


4. หลังจากนั้นทำการคลิกที่ ปุ่ม Next เพื่อจะเข้าสู่หน้าจอแสดงรายละเอียดเงื่อนไข ข้อตกลง เกี่ยวกับการนำโปรแกรมไปใช้งาน ดังรูป

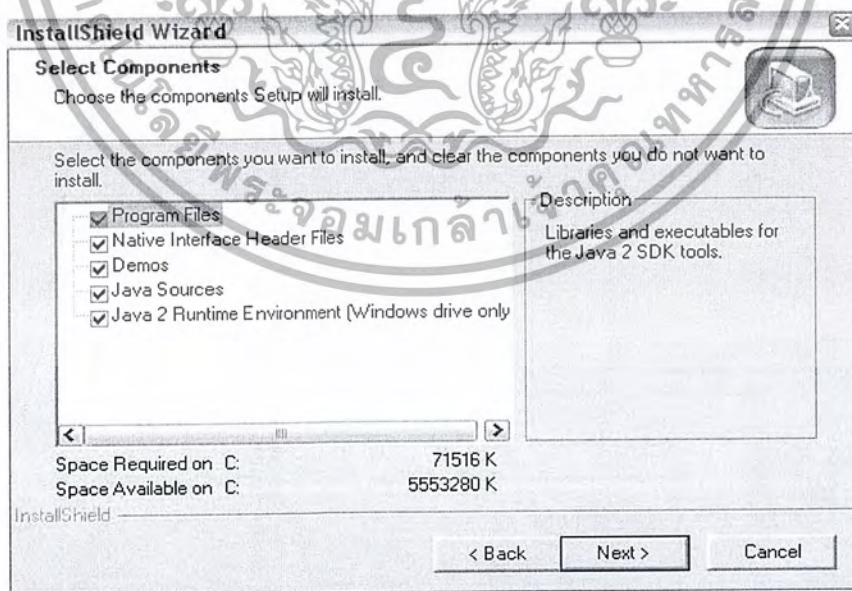


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

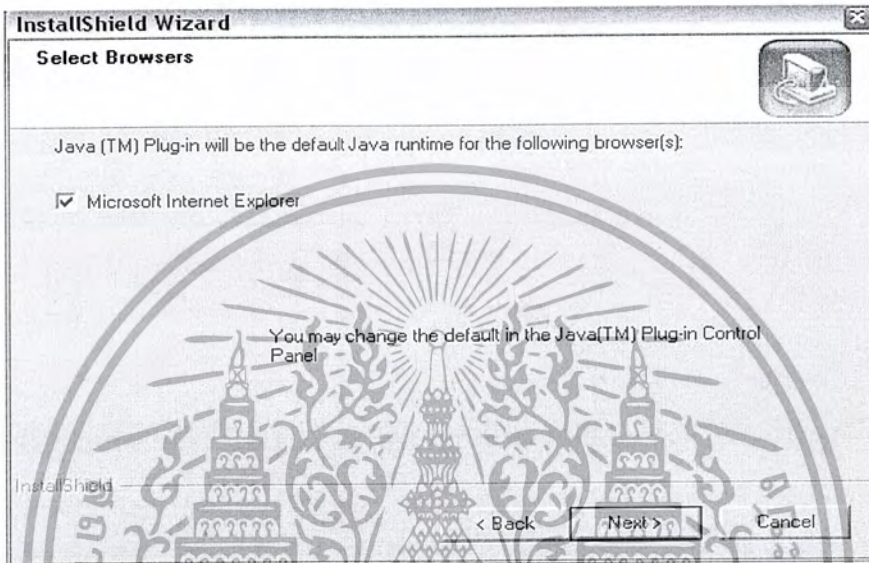
5. คลิกปุ่ม Yes เพื่อตอบรับข้อตกลง หน้าต่างถัดไปจะให้ทำการเลือกไดเรกทอรีที่ต้องการจะติดตั้ง J2SE ลงไป ดังรูป



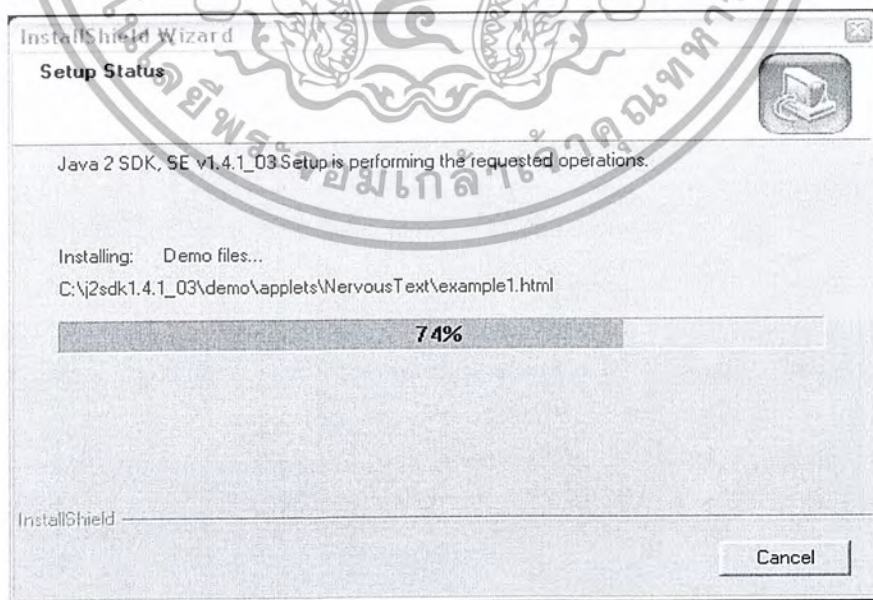
6. เมื่อเลือกไดเรกทอรี สำเร็จแล้ว ทำการคลิกปุ่ม Next ก็จะไปหน้าต่างดังรูป เพื่อให้เลือกส่วนประกอบที่ต้องการติดตั้ง จากนั้นเลือกคลิกปุ่ม Next



7. หลังจากนั้น หน้าจอจะแสดง ดังรูปที่ เพื่ออัปเดต Java Virtual Machine (JVM) ที่มีอยู่ในโปรแกรมเบราว์เซอร์ภายในเครื่อง โดย JVM ทำหน้าที่เป็นตัวช่วยรันโค้ดของ Java แล้วทำการเลือกว่าต้องการอัปเดต JVM ในโปรแกรม IE หรือไม่ ในกรณีที่ต้องการอัปเดต IE ก็ทำการกาเครื่องหมาย ถูกเข้าไปตรงช่องหน้า แล้วคลิกปุ่ม Next

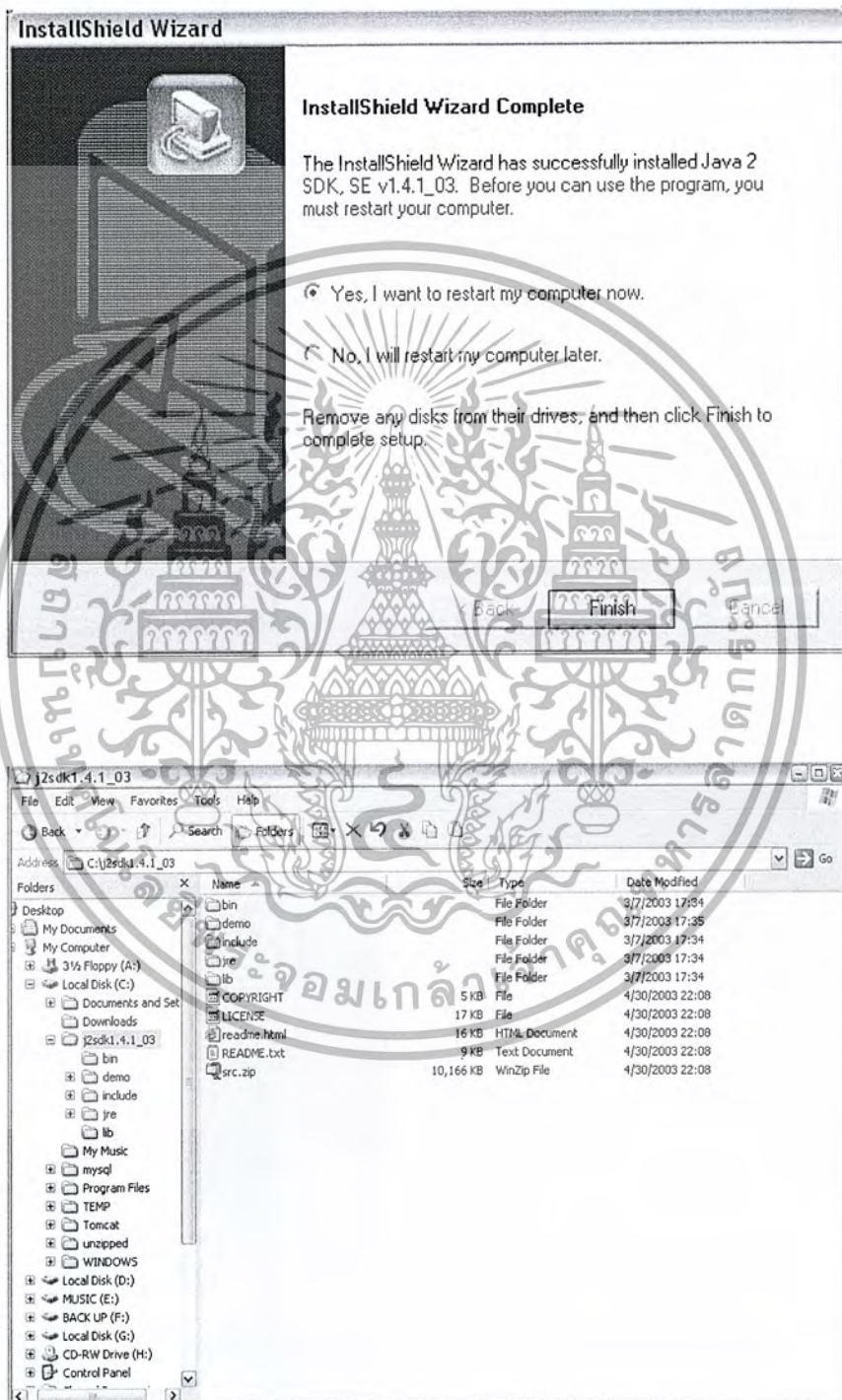


8. หลังจากนั้น โปรแกรมจะเริ่มต้นทำการติดตั้ง พร้อมทั้งแสดงเปอร์เซ็นต์ความคืบหน้า ดังรูปที่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

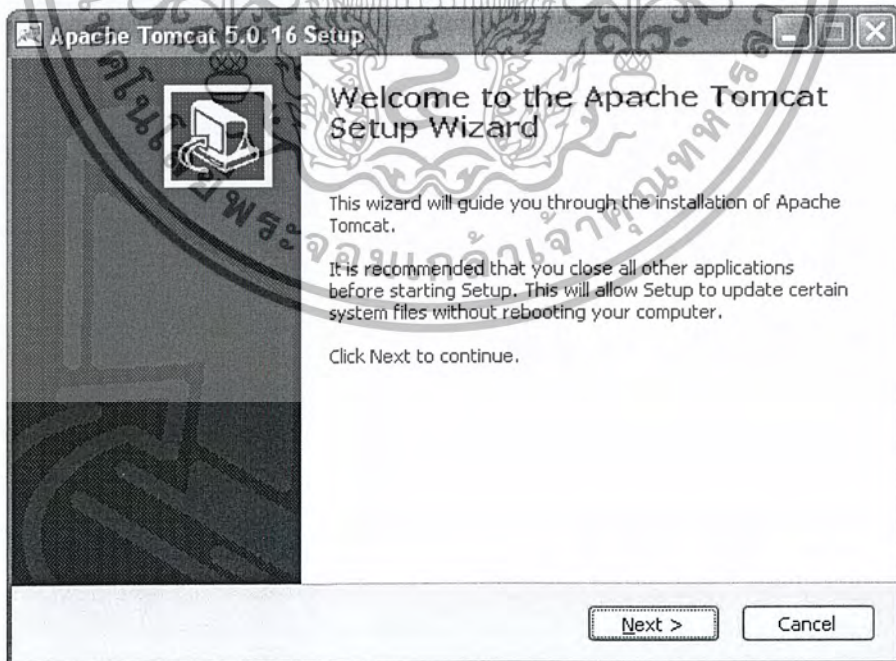
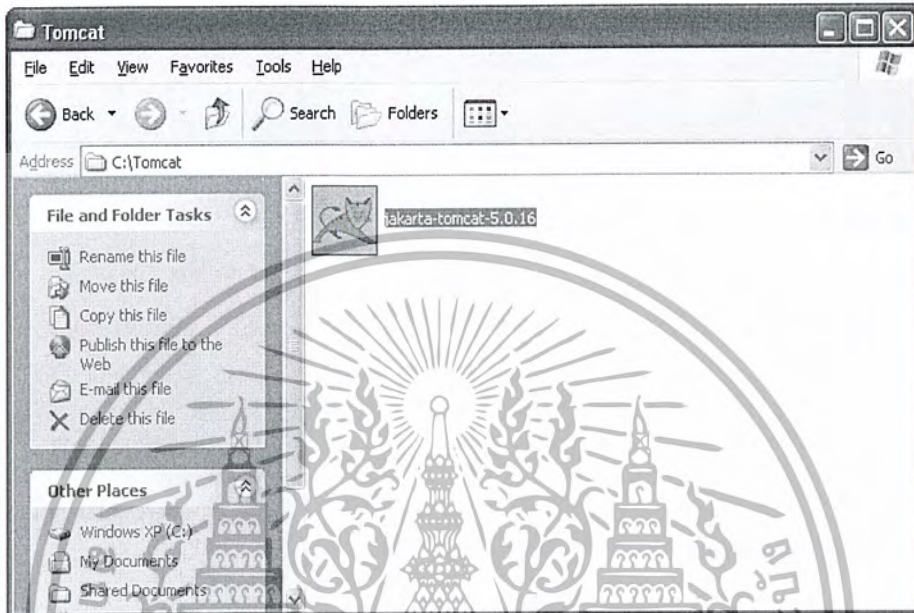
9. เมื่อการติดตั้งเสร็จสมบูรณ์แล้ว หน้าจอจะแสดงคังรูป จากนั้น เมื่อถึงตอนนี้คลิกที่ปุ่ม Finish ซึ่งหมายความว่าได้ทำการติดตั้งเสร็จสมบูรณ์เรียบร้อยแล้ว เราก็จะได้โปรแกรม J2SE ที่ติดตั้งอยู่ในไดเรกทอรี j2sdk1.4.1_03 ซึ่งประกอบด้วยไดเรกทอรีย่อยต่าง ๆ ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

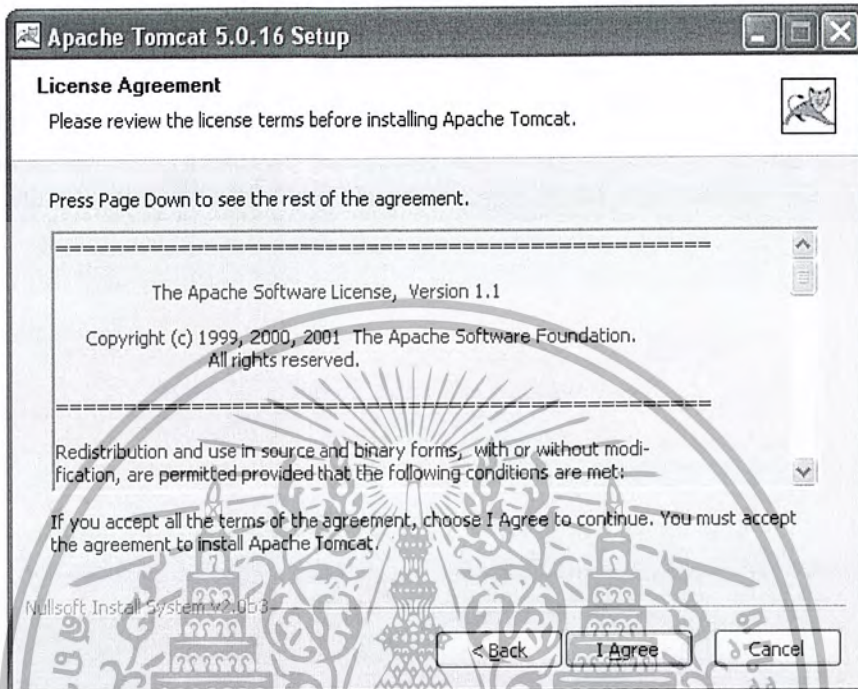
ขั้นตอนการติดตั้ง Tomcat Web Server

1. ทำการดับเบิลคลิกไฟล์ Jakarta-Tomcat-5.0.16 ดังรูป โดยสามารถ download ได้จาก www.apache.org และจะแสดงหน้าการติดตั้ง Tomcat เมื่อทำการคลิกปุ่ม Next

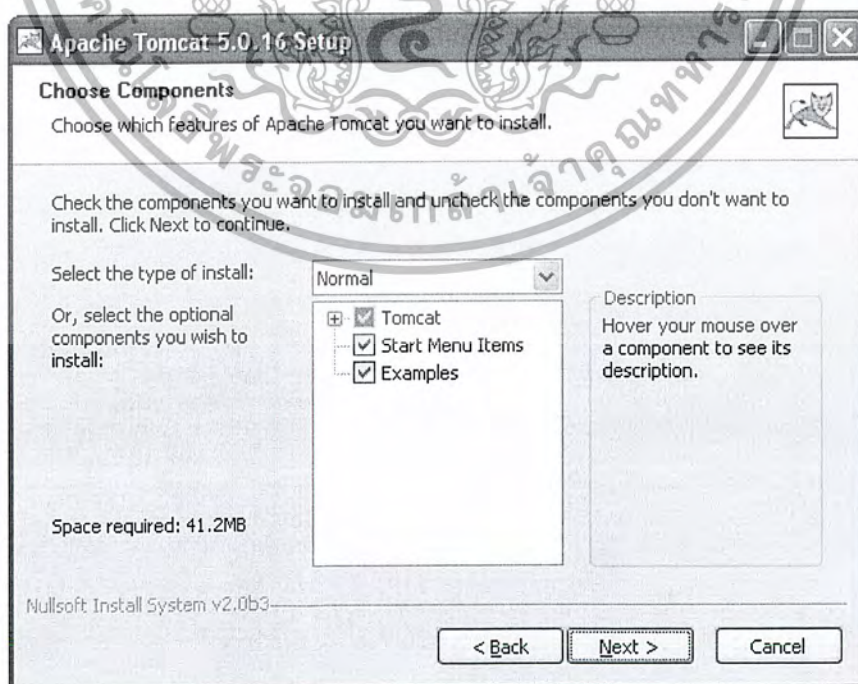


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คลิกปุ่ม I Agree เมื่อทำการอ่านข้อตกลงของ โปรแกรมเรียบร้อยแล้ว และต้องการลงโปรแกรม

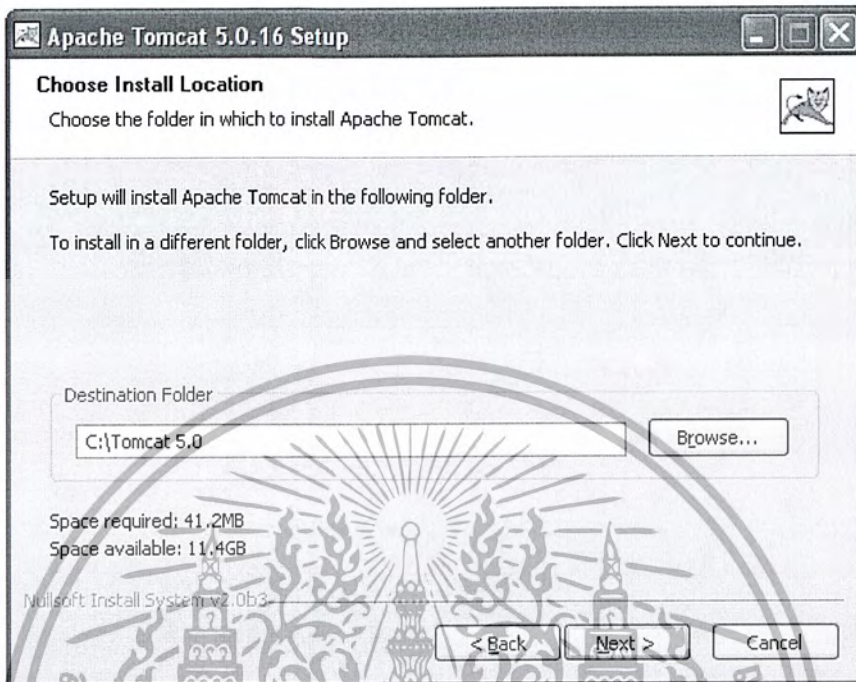


3. เลือกรูปแบบการลงโปรแกรม ในที่นี้ให้เลือกเป็น Normal แล้วทำการคลิกปุ่ม Next

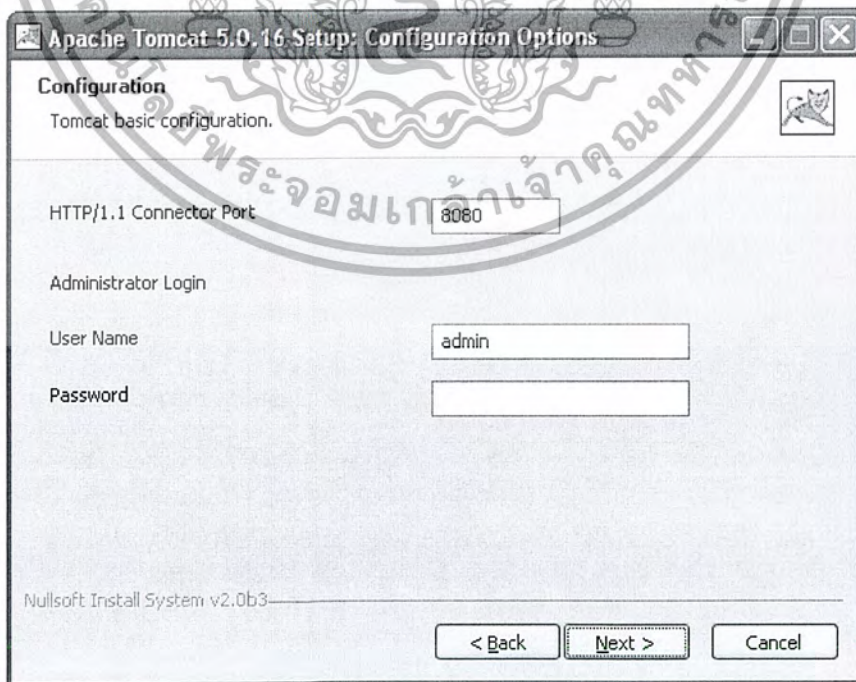


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการเลือกไดเรกทอรีปลายทาง ที่ต้องการติดตั้ง โปรแกรม จากนั้นทำการคลิกปุ่ม Next

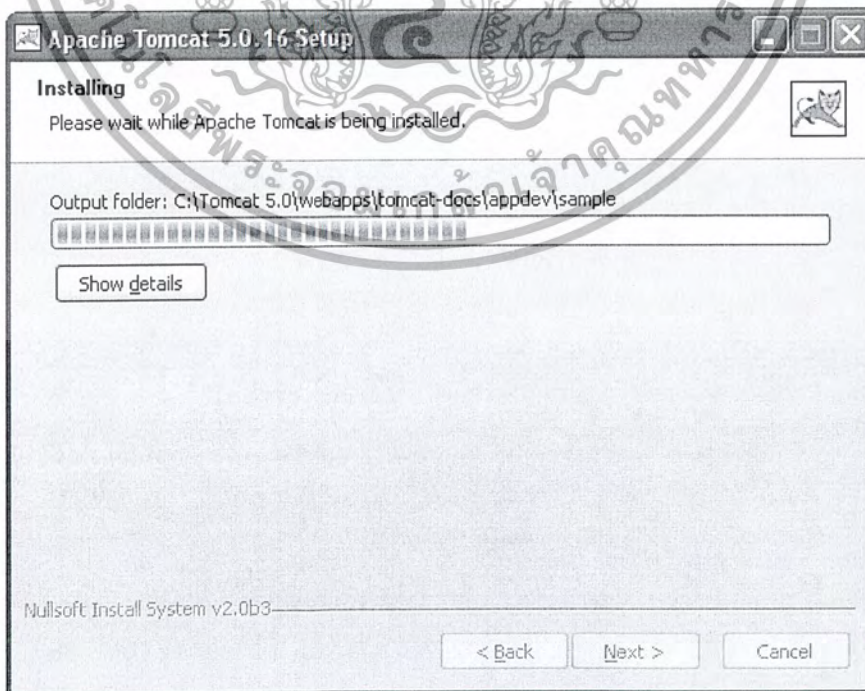
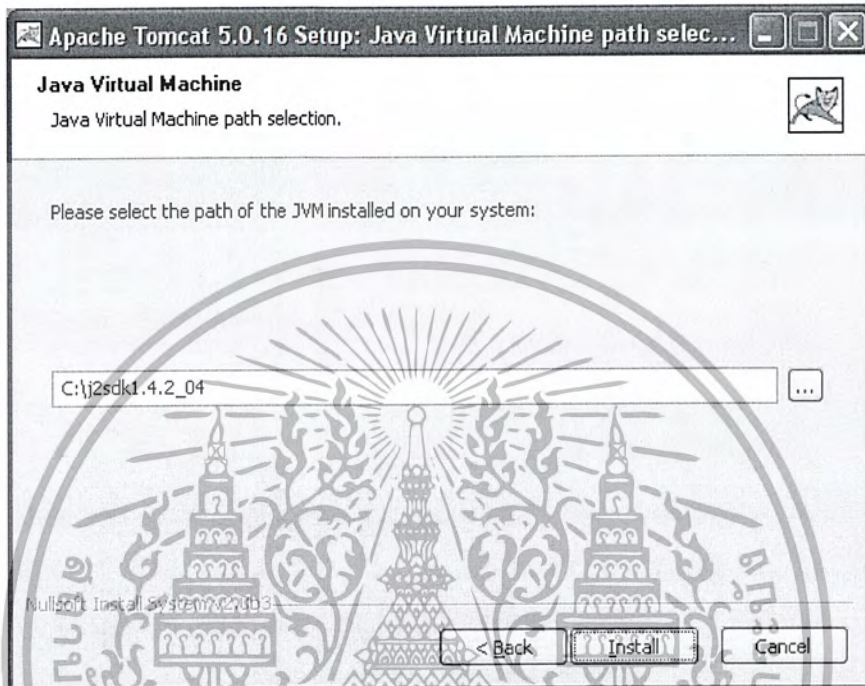


5. ทำการใส่หมายเลข Port ที่ใช้ติดต่อกับ Tomcat โดยให้ใช้ตามที่โปรแกรมตั้งมา ทำการคลิกปุ่ม Next



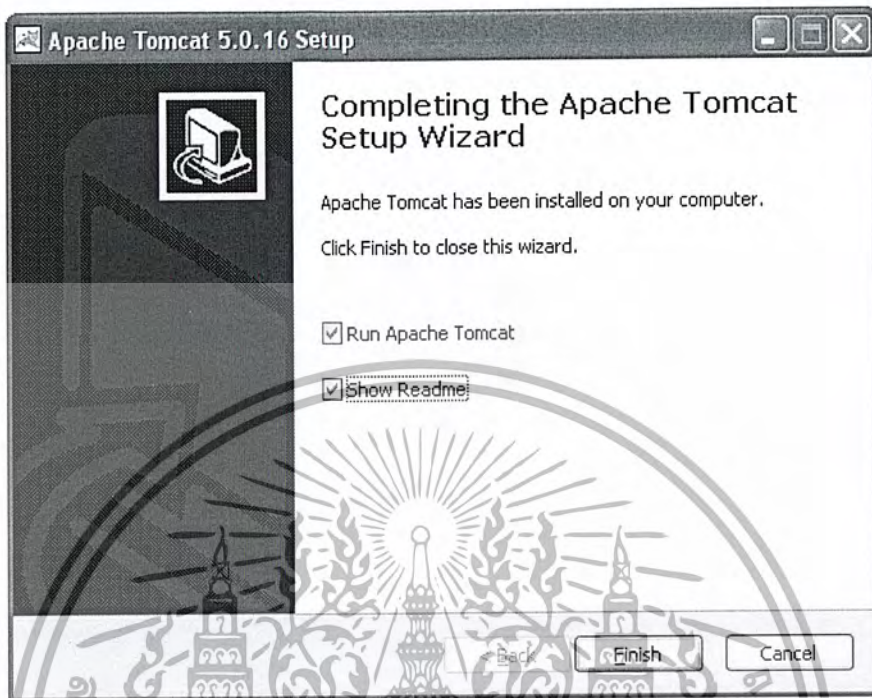
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ทำการกำหนดที่อยู่ของ Java Virtual Machine ซึ่งปกติแล้วถ้าเราทำการติดตั้ง Java2SDK ไว้ก่อนหน้าแล้ว โปรแกรมจะทำการค้นหาให้โดยอัตโนมัติ เราสามารถทำการคลิกปุ่ม Install เพื่อเริ่มติดตั้งโปรแกรมได้ทันทีดังรูป



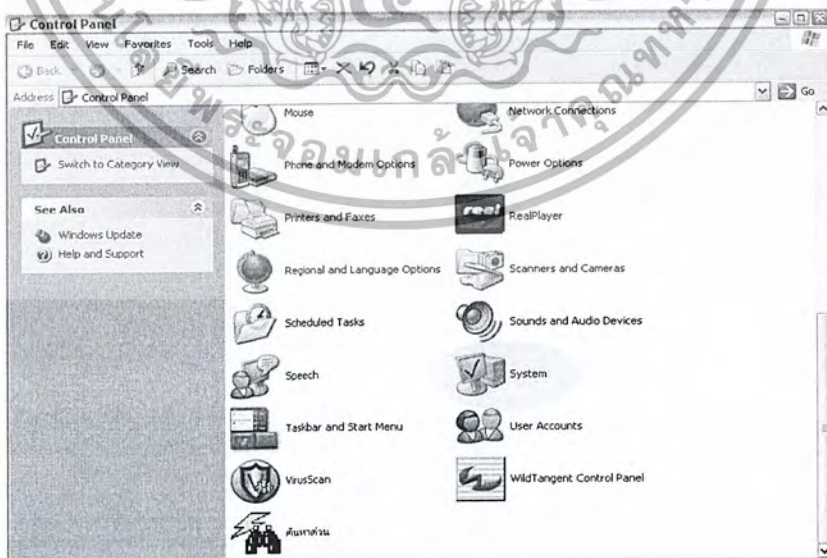
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทำการคลิกปุ่ม Finish เมื่อ โปรแกรมทำการติดตั้งเสร็จสมบูรณ์



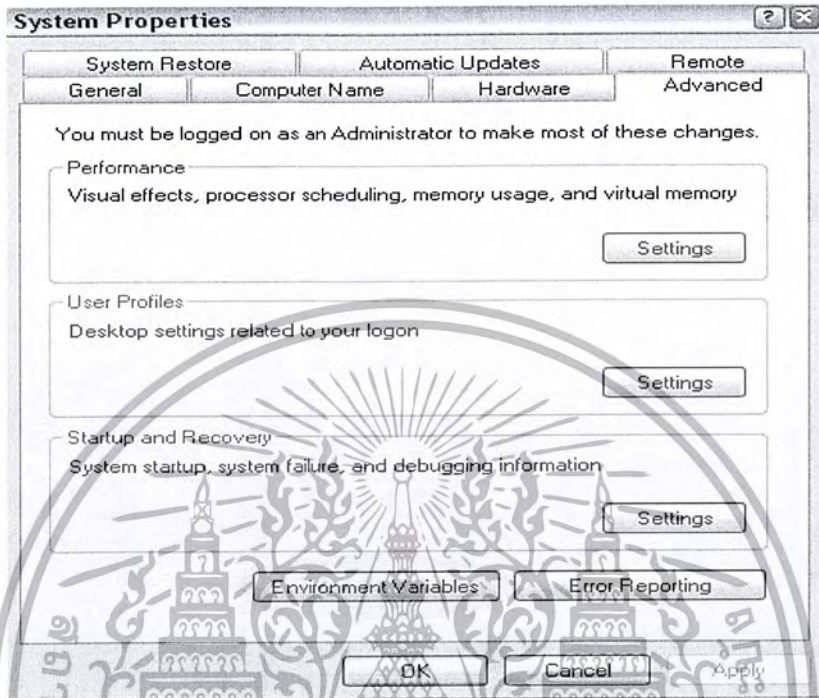
8. หลังจากติดตั้ง Tomcat แล้ว จะต้องทำการเพิ่มตัวแปรเข้าไปในระบบ โดยมีขั้นตอนดังนี้

8.1 คลิกปุ่ม Start ของ Windows แล้วเลือกคำสั่ง Settings > Control Panel จะปรากฏหน้าต่าง Control Panel ดังรูป จากนั้นทำการดับเบิลคลิกที่ไอคอน System ในหน้าต่าง Control Panel นี้

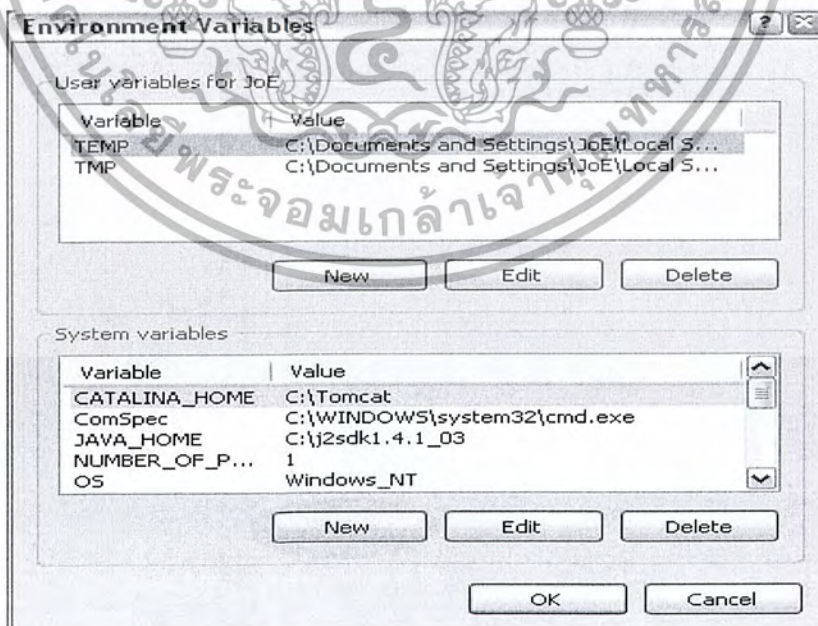


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 หน้าต่าง System Properties จะเปิดออกมา ให้คลิกไปที่แท็บ Advanced ดังรูป จากนั้นคลิกที่ปุ่ม Environment Variables

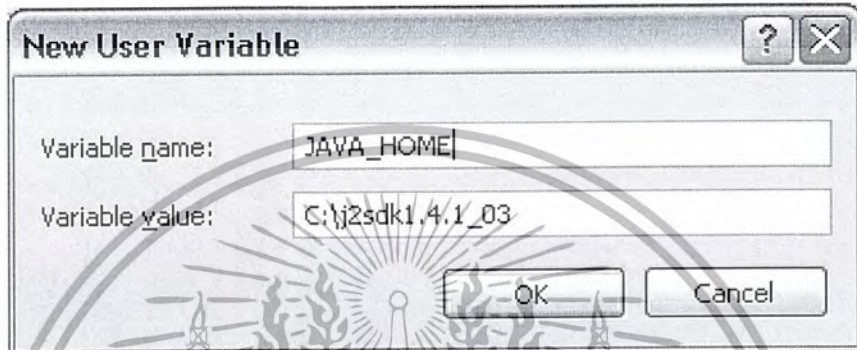


8.3 หน้าต่าง Environment Variables จะเปิดขึ้นมาดังรูปที่ ก.21 แล้วคลิกที่ New



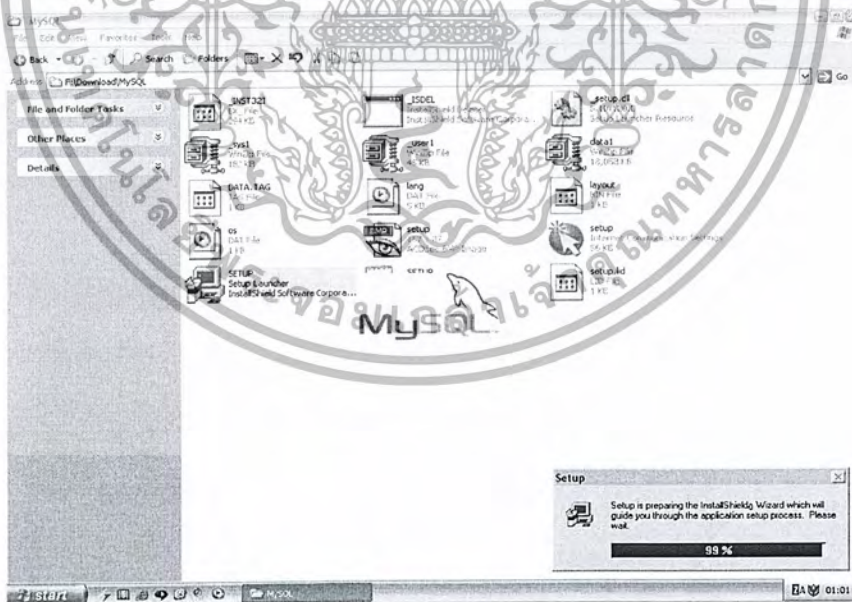
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.4 หลังจากคลิกที่ปุ่ม New หน้าต่าง New System Variable จะเปิดออกมา ให้ทำการใส่ค่าตัวแปรลงในช่อง Variable name และใส่ค่าลงในช่อง Variable value แล้วทำการคลิกปุ่ม OK โดยในที่นี้จะกำหนดค่าตัวแปรใหม่ 2 ตัวชื่อ JAVA_HOME ใส่ลงในช่อง Variable name และใส่ค่า C:\j2sdk1.4.1_3 ลงใน Variable value และตัวแปรอีกตัวหนึ่งชื่อ CATALINA_HOME ใส่ลงในช่อง Variable name และใส่ค่า C:\Tomcat ลงใน Variable value

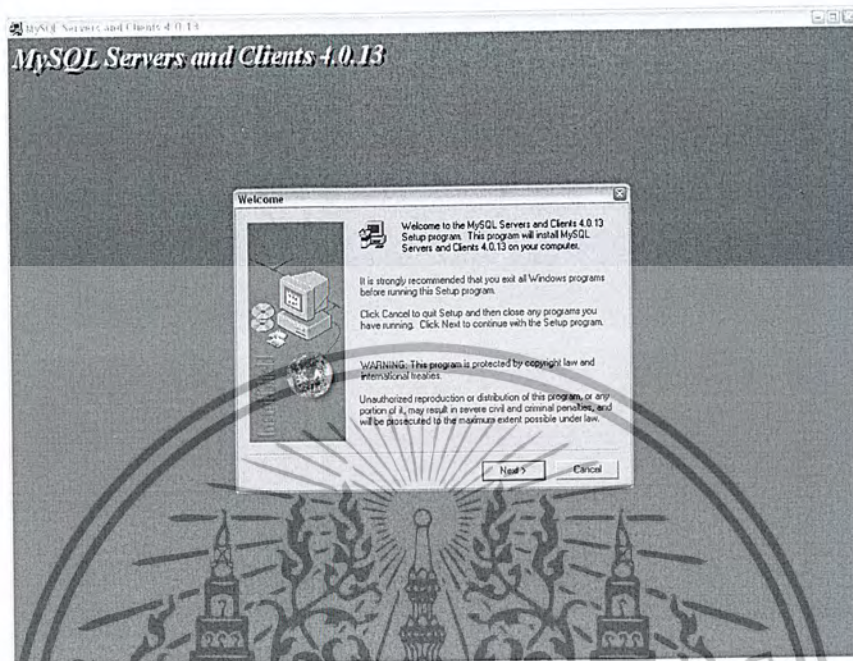


ขั้นตอนการติดตั้ง MySQL

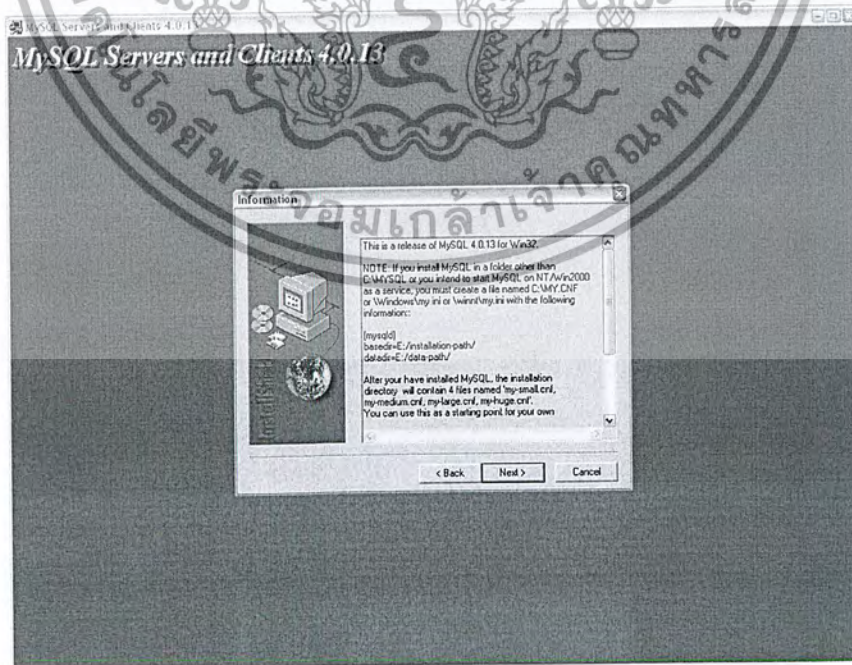
1. ดับเบิลคลิกที่ไฟล์ setup.exe แล้วโปรแกรมจะ extract ไฟล์เพื่อเตรียมการติดตั้ง ดังรูป



2. จากนั้นจะเข้าสู่หน้าจอแสดงข้อความต้อนรับการติดตั้งโปรแกรม ดังรูป

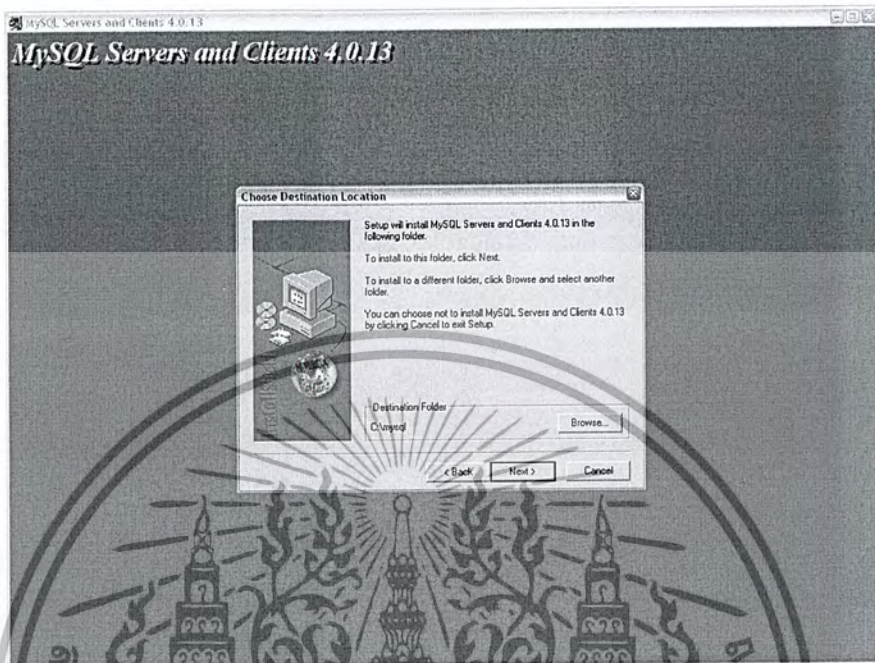


3. หลังจากนั้นทำการคลิกที่ ปุ่ม Next ก็จะเข้าสู่หน้าจอแสดงรายละเอียดเงื่อนไขข้อตกลงเกี่ยวกับการนำโปรแกรมไปใช้งาน ดังรูป

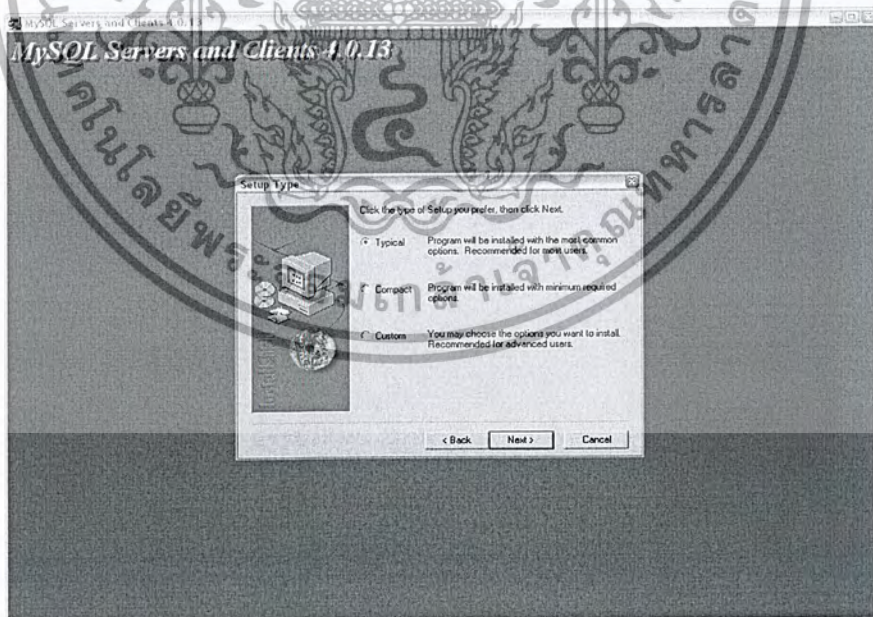


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คลิกปุ่ม Yes เพื่อตอบรับข้อตกลง หน้าต่างถัดไปจะให้ทำการเลือกไดเรกทอรีที่ต้องการจะติดตั้ง MySQL ลงไป ดังรูป

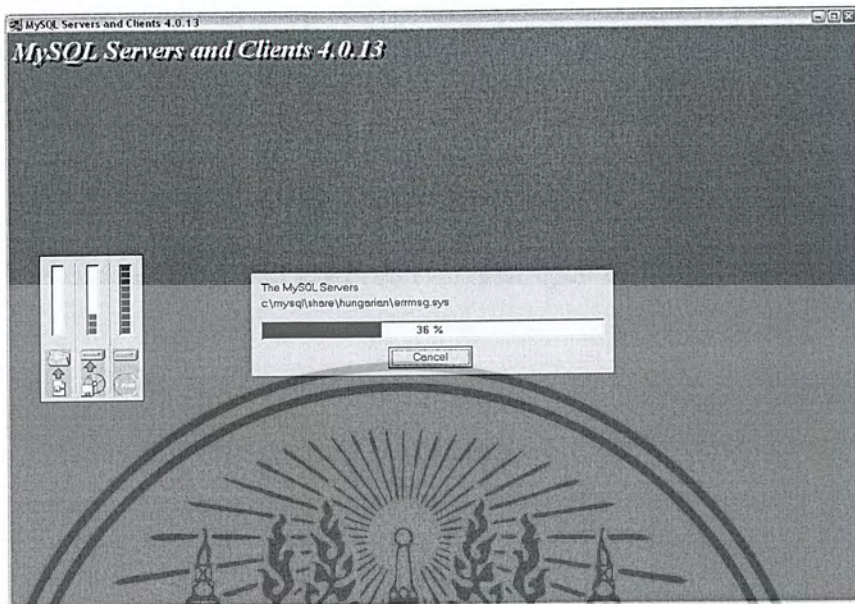


5. ทำการเลือกชนิดของการติดตั้ง ในที่นี้จะเลือก แบบ Typical แล้วคลิกปุ่ม Next ดังรูป

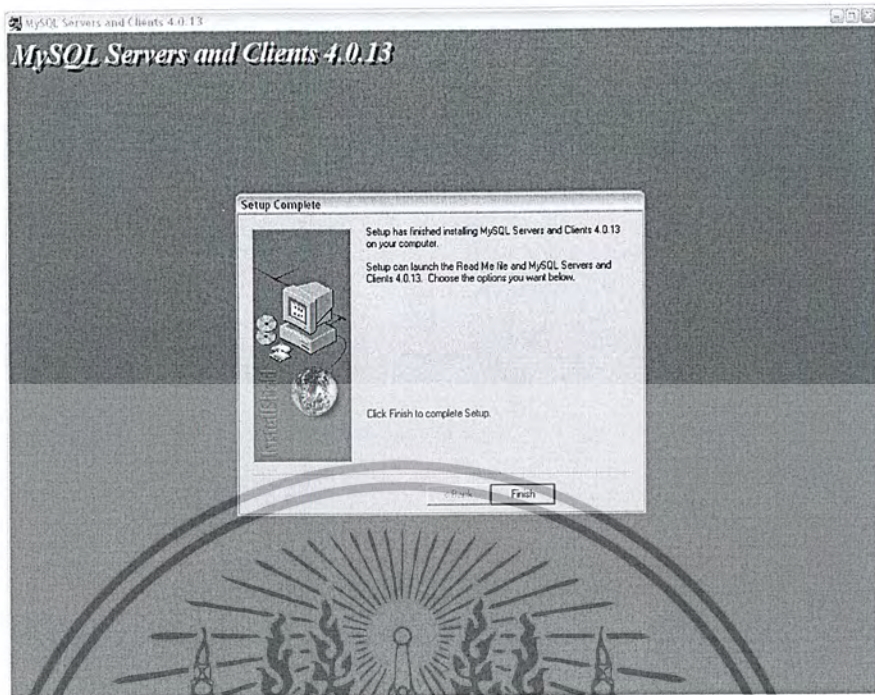


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

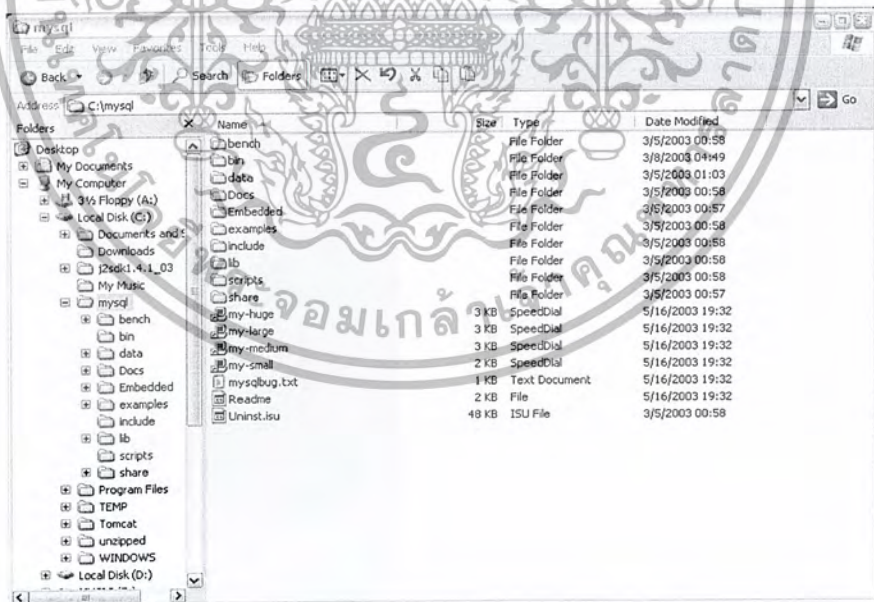
6. หลังจากนั้น โปรแกรมจะเริ่มดำเนินการติดตั้ง พร้อมทั้งแสดงเปอร์เซ็นต์ความคืบหน้าดังรูป



7. ถ้าการติดตั้งสมบูรณ์แล้ว หน้าจอจะแสดงดังรูป จะปรากฏออกมา เมื่อถึงตอนนี้คลิกที่ปุ่ม Finish ซึ่งหมายความว่าเราได้ทำการติดตั้งเสร็จสมบูรณ์เรียบร้อยแล้ว เราก็จะได้โปรแกรม MySQL ที่ติดตั้งอยู่ในไดเรกทอรี mysql ซึ่งประกอบด้วยไดเรกทอรีย่อยต่าง ๆ



เมื่อการติดตั้งสมบูรณ์แล้ว จะปรากฏหน้าต่างดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้งไดรเวอร์ JDBC

การติดตั้งไดรเวอร์ของฐานข้อมูลที่ใช้กับ JSP ให้ก๊อปปี้ไฟล์ mysql-connector-java-3.0.9-stable-bin ในแผ่น CD-ROM ไปไว้ที่ไดเรกทอรี C:\j2sdk1.4.2_03\jre\lib\ext\ ซึ่งเป็นไดเรกทอรีของ J2SE ที่สามารถเก็บไฟล์ .jar ไว้ หรือจะนำไปเก็บในไดเรกทอรี C:\Tomcat_Home\lib หรือ C:\Tomcat_Home\webapps\ROOT\WEB-INF\lib ของ Tomcat ก็ได้หลังจากนั้นเมื่อ shutdown การทำงานของ Tomcat และ restart ใหม่แล้ว ก็สามารถเขียนสคริปต์ JSP ติดต่อกับ MySQL ได้เลย



ภาคผนวก ข.

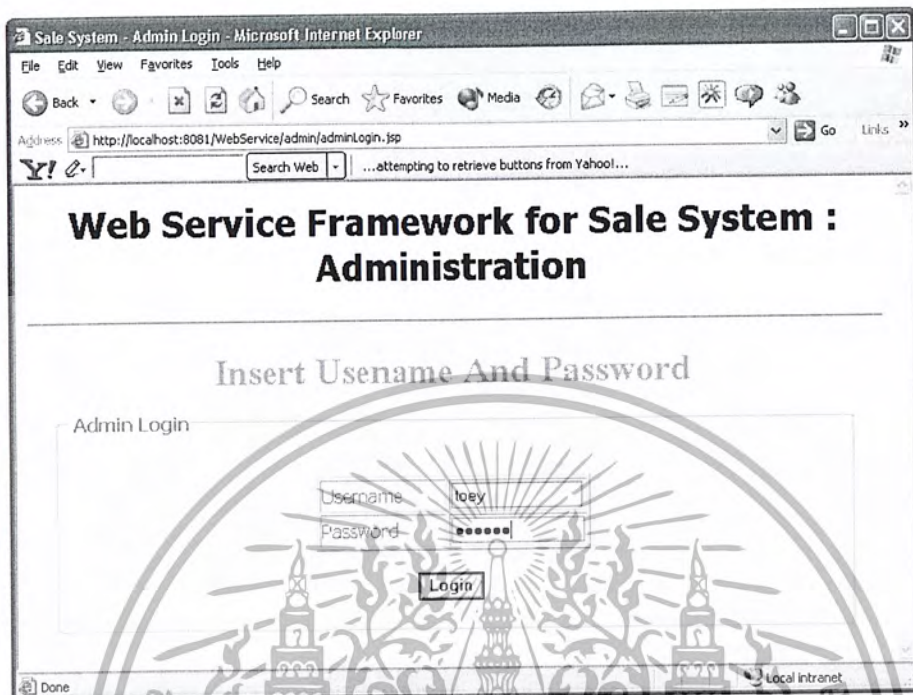
คู่มือการใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การใช้งานในฝ่ายผู้ดูแลระบบ

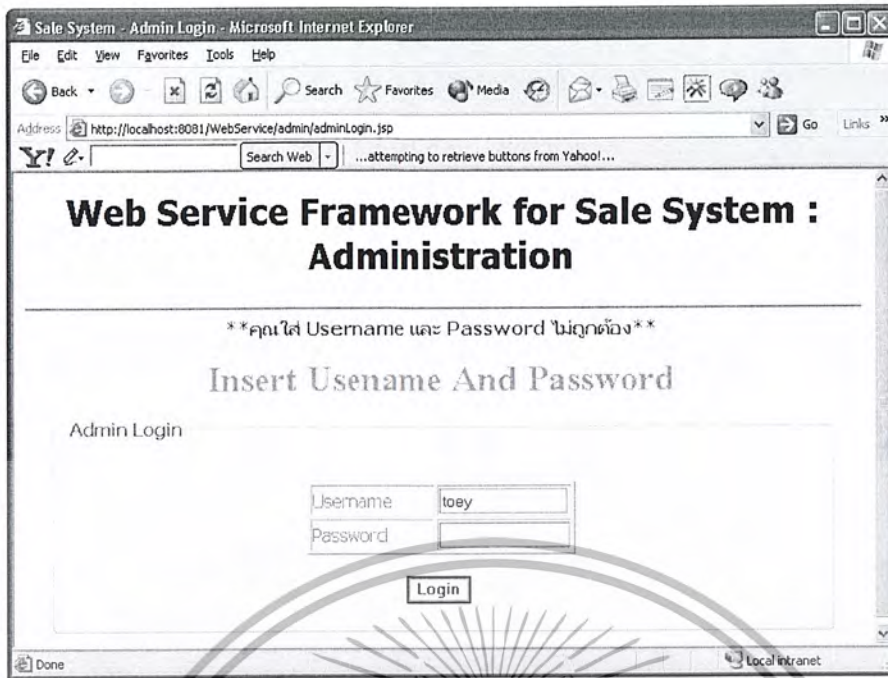
เมื่อเริ่มเข้าทำงานจะแสดงหน้าจอเพื่อทำการ Login เข้าสู่ระบบ ดังนี้



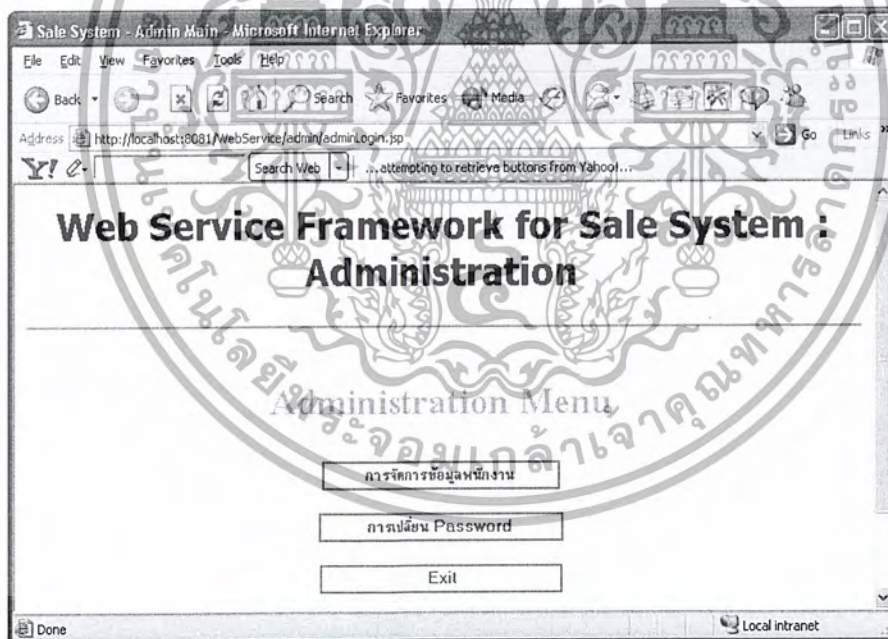
หลังจากกรอก Username และ Password แล้วก็กดปุ่ม Login เพื่อเข้าใช้งานระบบได้ แต่หากยังไม่สามารถกรอกข้อมูลใดๆ แต่เลือกปุ่ม Login ระบบจะทำการแสดงข้อความเตือน



หากกรอกข้อมูลผิดระบบจะทำการเตือนดังนี้



เมื่อเข้าสู่ระบบแล้วจะแสดงหน้าจอกิจกรรมหลัก ของฝ่ายผู้ดูแลระบบ



ผู้ดูแลระบบสามารถเลือก กิจกรรมหลักที่จะทำได้ดังต่อไปนี้ คือ

1.1 การจัดการข้อมูลพนักงาน

เมื่อเลือกปุ่มการจัดการข้อมูลพนักงานจะเข้าสู่หน้าจอ ที่แสดงข้อมูลพนักงานทั้งหมด

โดยในหน้าจอนี้ ผู้ดูแลระบบสามารถแก้ไข เพิ่มข้อมูล หรือ ลบข้อมูลพนักงานในบริษัทได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Service Framework for Sale System : Administration

Employee Management

Employee ID	Name	Department	Edit	Delete
e0000001	Tom Hark	Marketing	Edit	Delete
e0000002	Julie Robert	Stock	Edit	Delete
e0000004	Harison Ford	Account	Edit	Delete
e0000006	Hillary Duff	Stock	Edit	Delete
e0000008	nMarketing sMarketing	Marketing	Edit	Delete
e0000009	nAccount sAccount	Account	Edit	Delete
e0000010	nStock sStock	Stock	Edit	Delete

Buttons: Add New Employee, Menu

1.1.1 การเพิ่มพนักงานใหม่

เมื่อกดปุ่ม Add New Employee จะปรากฏหน้าจอ ให้กรอกข้อมูล

Web Service Framework for Sale System : Administration

Insert New Employee

Employee Detail

Firstname *

Surname *

Nickname

Address *

Telephone Number *

Mobile Number

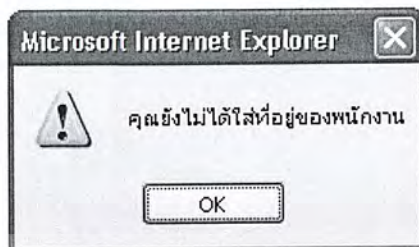
Username *

Department *

Buttons: Submit, Clear, Back

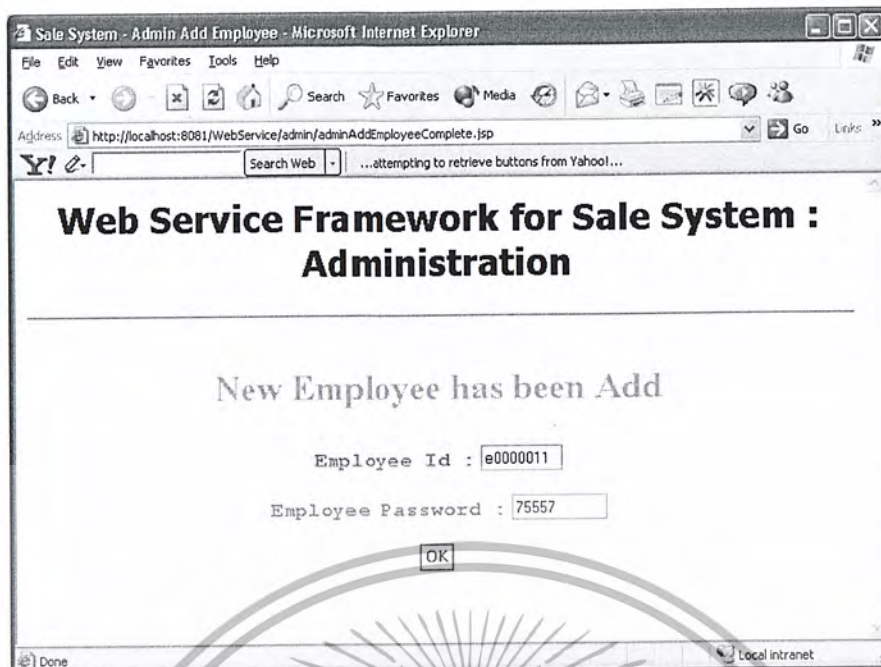
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ดูแลระบบจะต้องกรอก ชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ Username และ ฝ่ายที่พนักงานสังกัด ไม่เช่นนั้นจะถือว่าการกรอกข้อมูลไม่สำเร็จ และขึ้นข้อความเตือนในสิ่งที่ผู้ดูแลระบบใส่ข้อมูลไม่ครบ เช่น

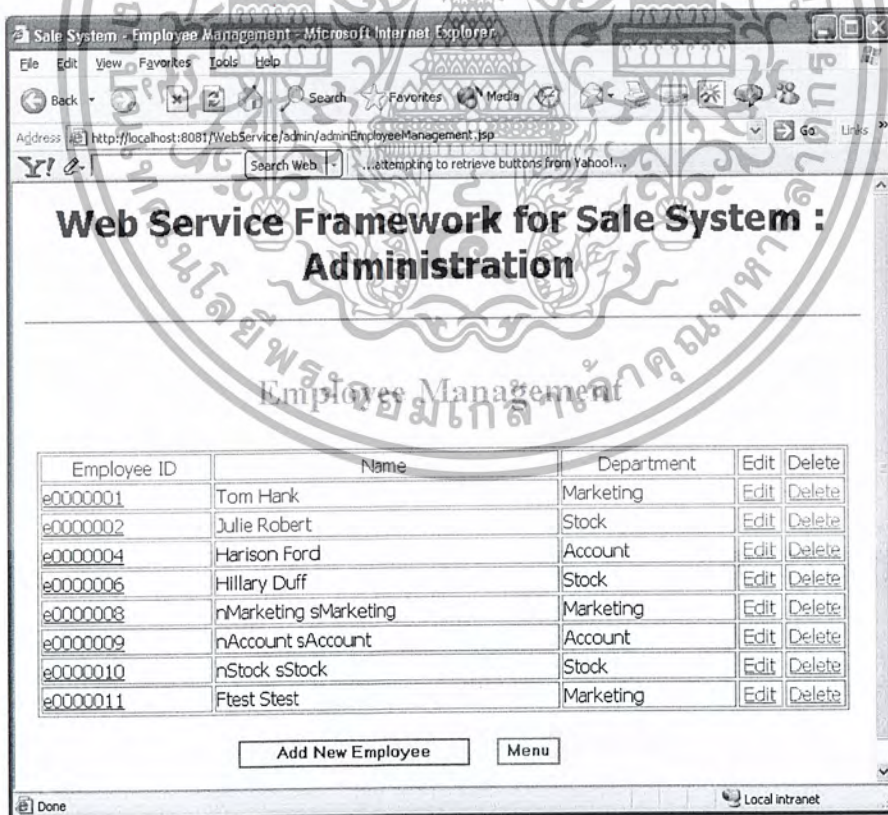


สามารถเลือกปุ่ม Clear เพื่อลบข้อมูลที่ได้ทำการกรอกไปแล้วทั้งหมดได้โดยไม่ต้องลบทีละช่อง และเมื่อกรอกข้อมูลครบแล้วจึงเลือกปุ่ม Submit ระบบจะแสดงข้อมูลที่ได้กรอกไปทั้งหมด เพื่อยืนยันว่าต้องการจะบันทึกข้อมูลนี้

หากไม่ต้องการบันทึกข้อมูลนี้แต่ต้องการแก้ไขข้อมูล สามารถเลือกปุ่ม Back เพื่อกลับไปแก้ไขข้อมูลในหน้าก่อนหน้านี้ได้ และหากต้องการบันทึกข้อมูลนี้เลือกปุ่ม Submit



ระบบจะทำการสร้าง รหัสประจำตัวพนักงาน และ Password ให้กับพนักงานโดยอัตโนมัติ เลือกปุ่ม OK เพื่อดำเนินการต่อไป



จะแสดงข้อมูลทั้งหมดของพนักงาน ซึ่งหมายความว่าข้อมูลถูกบันทึกแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.2 การแก้ไขข้อมูลพนักงาน

สามารถเลือก Edit หลังชื่อพนักงานที่ต้องการแก้ไข

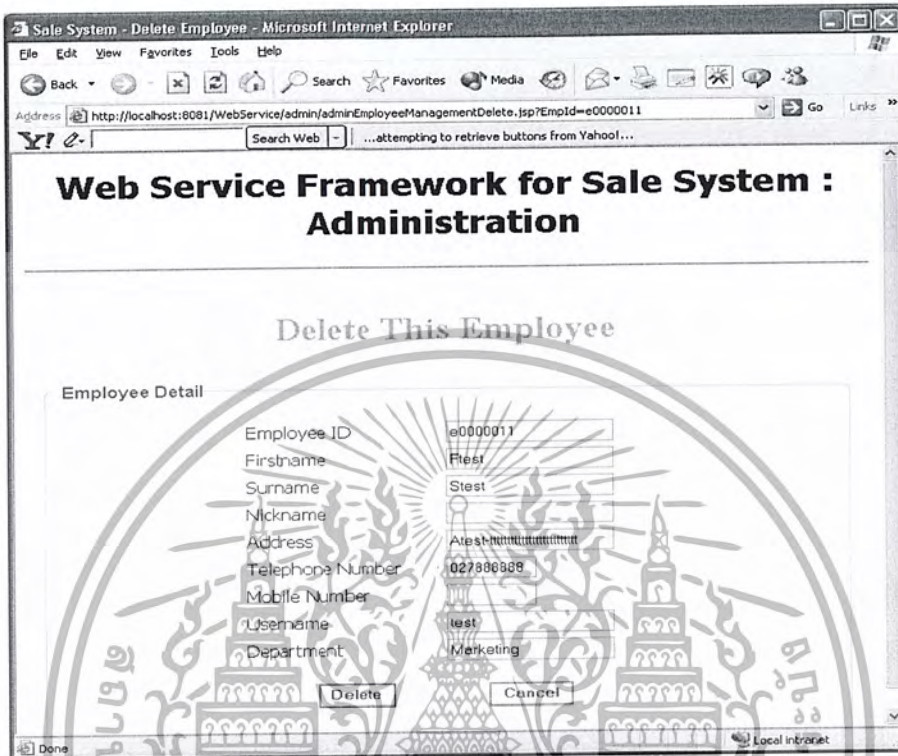
The screenshot shows a web browser window with the following details:

- Browser: Microsoft Internet Explorer
- Title: Sale System - Edit Employee
- Address: http://localhost:8081/WebService/admin/adminEmployeeManagementEdit.jsp?EmpId=e0000011
- Page Title: Web Service Framework for Sale System : Administration
- Section: Edit This Employee
- Form Title: Employee Detail
- Fields:
 - Employee ID: e0000011
 - Firstname: Ftest
 - Surname: Stest
 - Address: Atest
 - Telephone Number: 027800000
 - Mobile Number: [empty]
 - Department: Marketing
- Buttons: OK, Cancel

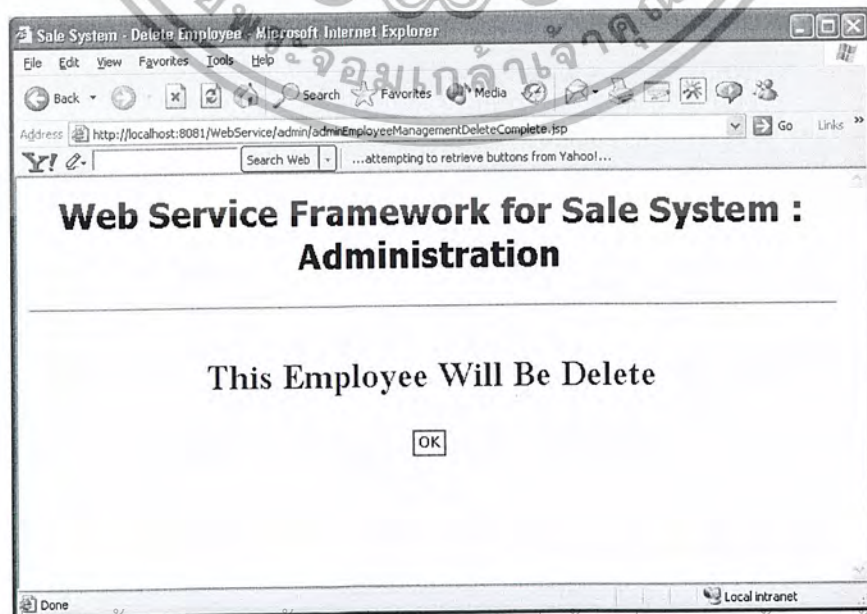
ในการแก้ไขจะไม่สามารถแก้ไขรหัสประจำตัวพนักงานได้ สามารถเลือก Cancel เพื่อยกเลิกการแก้ไขข้อมูลพนักงาน หรือเลือก OK เพื่อทำการบันทึกข้อมูลที่แก้ไขต่อไป

1.1.3 การลบข้อมูลพนักงาน

เลือก **Delete** ที่อยู่หลังชื่อพนักงานที่ต้องการลบข้อมูลจากนั้นระบบจะแสดงข้อมูลของพนักงานคนนั้น



ผู้ดูแลระบบจะสามารถเลือกได้ว่าต้องการลบข้อมูลของพนักงานคนนั้นใช่หรือไม่ โดยถ้าไม่ต้องการลบข้อมูล กด Cancel เพื่อกลับสู่หน้าจอก่อนหน้านี้ และเลือก Delete เพื่อทำการลบข้อมูลของพนักงานคนนั้น

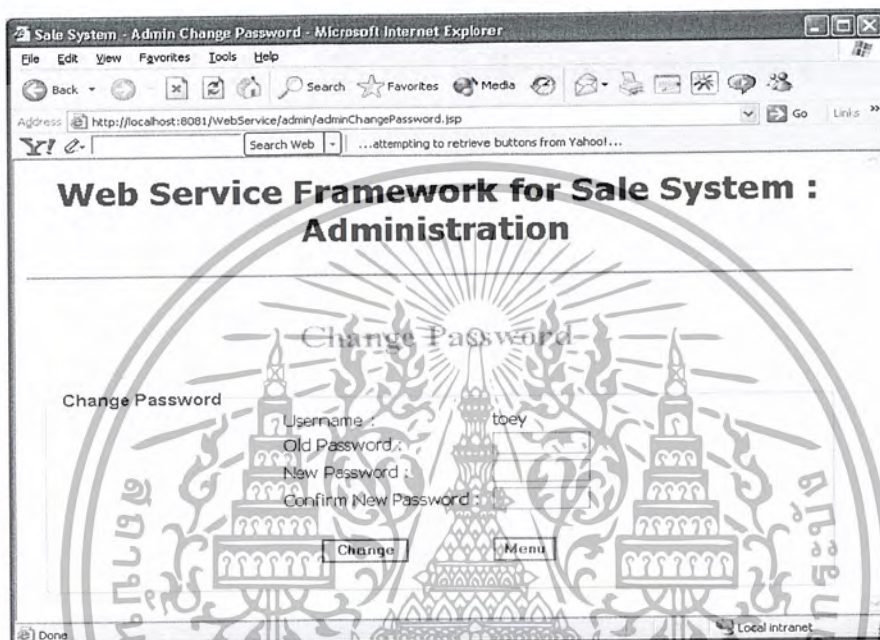


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

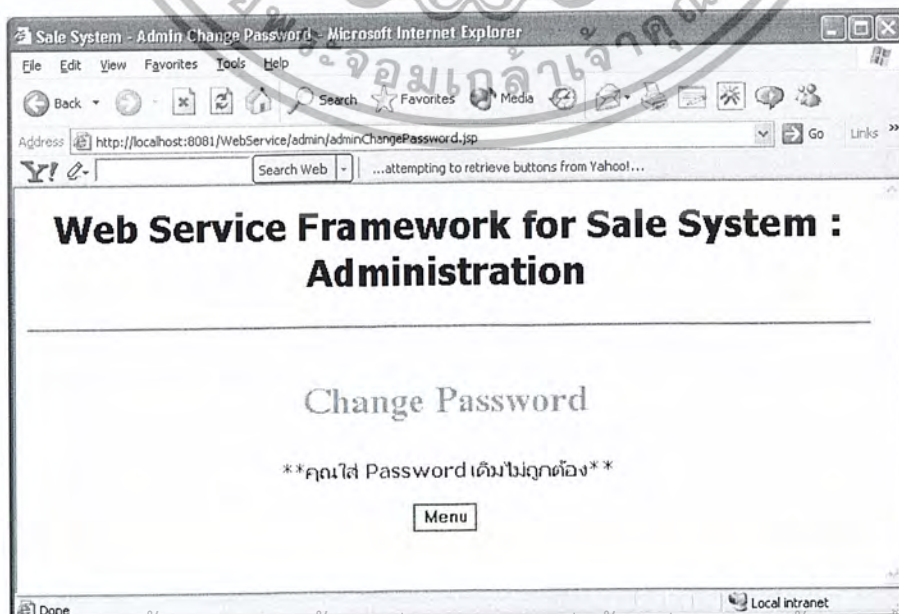
จากนั้นจะแสดงข้อความว่าได้ทำการลบข้อมูลของพนักงานเรียบร้อยแล้วโดยสามารถตรวจสอบได้จาก การกดปุ่ม OK เพื่อกลับหน้าจอแสดงข้อมูลพนักงานทั้งหมด

1.2 การเปลี่ยน Password

ผู้ดูแลระบบสามารถเปลี่ยน Password เพื่อทำการเข้าระบบของตัวเองได้ โดยเลือกปุ่ม การเปลี่ยน Password เพื่อเข้าสู่หน้าจอ



จากนั้นผู้ดูแลระบบสามารถใส่ Password เดิมของตนเองและ กรอก Password ใหม่ที่ต้องการ พร้อมทำการยืนยัน กรณีที่ได้ Password เดิมไม่ถูกต้องจะแสดงข้อความเตือน

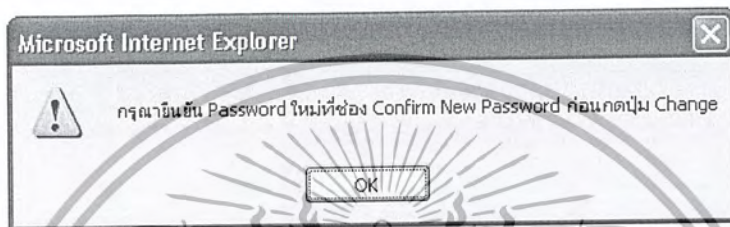


เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

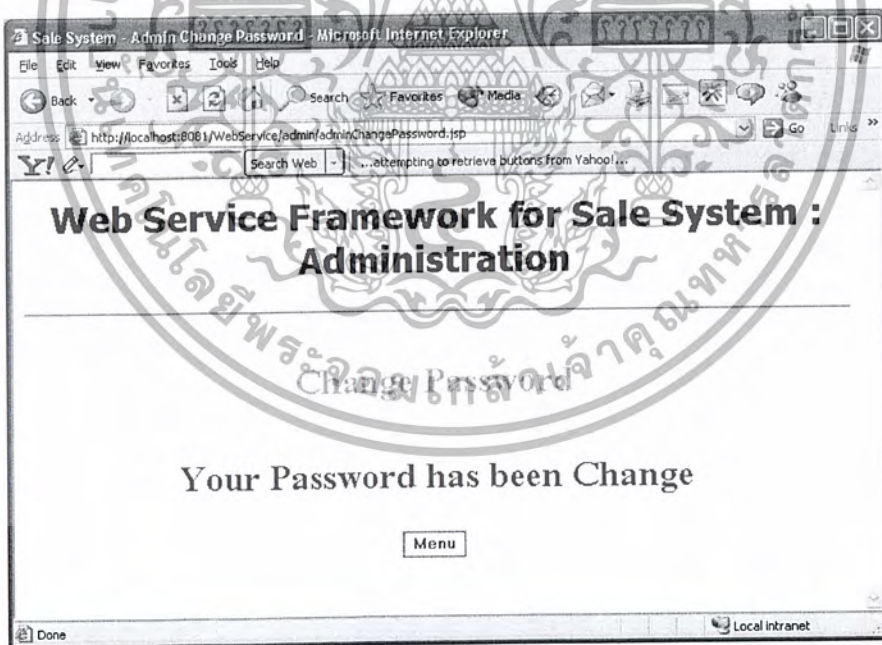
กรณีที่ใส่ Password ใหม่ไม่ตรงกันจะแสดงข้อความเตือน ดังนี้



และกรณีที่กรอกข้อมูลไม่ครบระบบจะแสดงข้อความเตือน ดังนี้



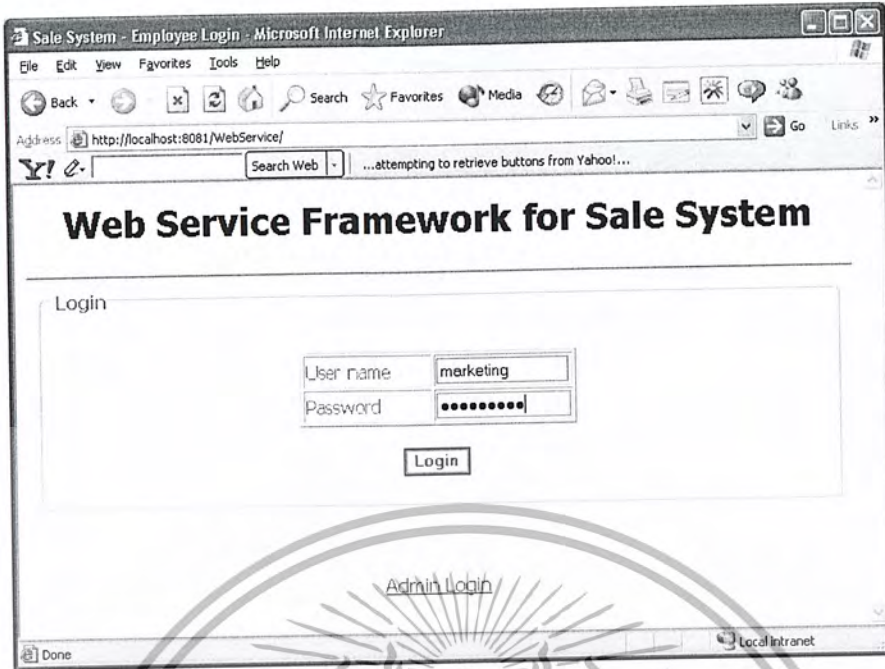
เมื่อกรอกข้อมูลครบเรียบร้อยแล้วจากนั้นเลือก Change เพื่อทำการแก้ไขข้อมูล และเลือก Menu ได้ เมื่อไม่ต้องการแก้ไข Password และกลับสู่หน้าจอการทำงานหลัก



2. การใช้งานในฝ่ายขาย

พนักงานจะต้องทำการกรอก Username และ Password เพื่อเข้าใช้งานในฝ่ายของตนเอง ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

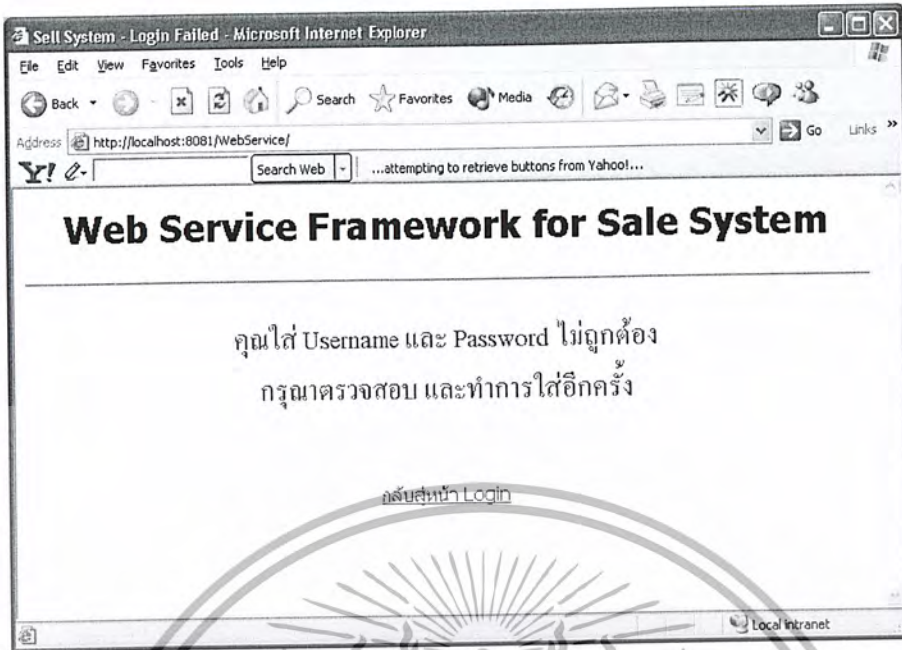


จากนั้นเลือกปุ่ม Login เพื่อเข้าใช้งาน ถ้ายังไม่ได้ทำการกรอก Username และ Password จะแสดงข้อความ ดังนี้

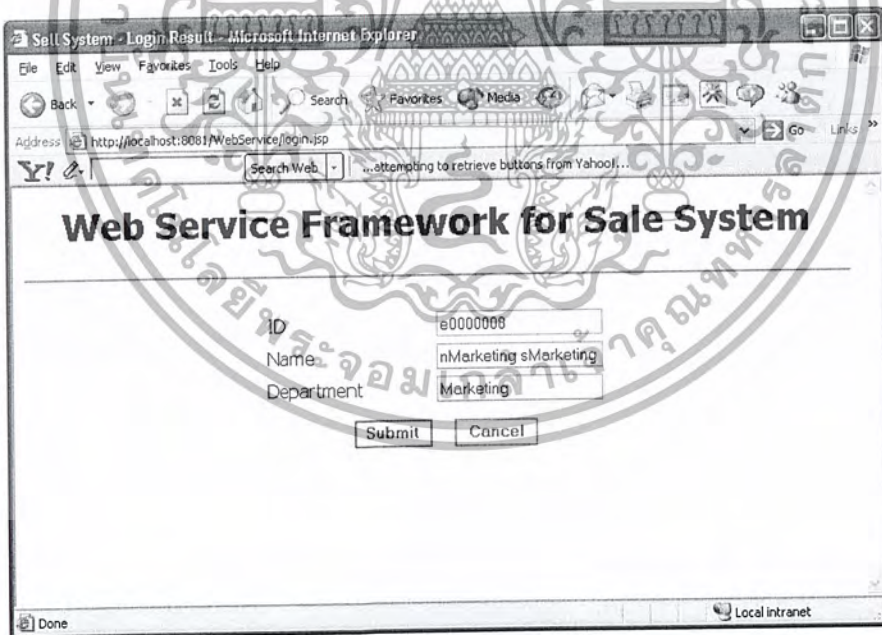


แต่ถ้าทำการกรอกข้อมูล Username และ Password แล้วแต่ข้อมูลที่กรอกไม่ถูกต้องจะแสดงข้อความ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อกรอกข้อมูลถูกต้องเรียบร้อยแล้ว จะเข้าสู่หน้าที่จะแสดงข้อมูลว่าพนักงานที่ทำการ Login เข้ามา มีข้อมูล รหัสประจำตัว ชื่อ นามสกุล และสังกัดฝ่ายใด เลือกปุ่ม Submit เพื่อเข้าใช้งานระบบ

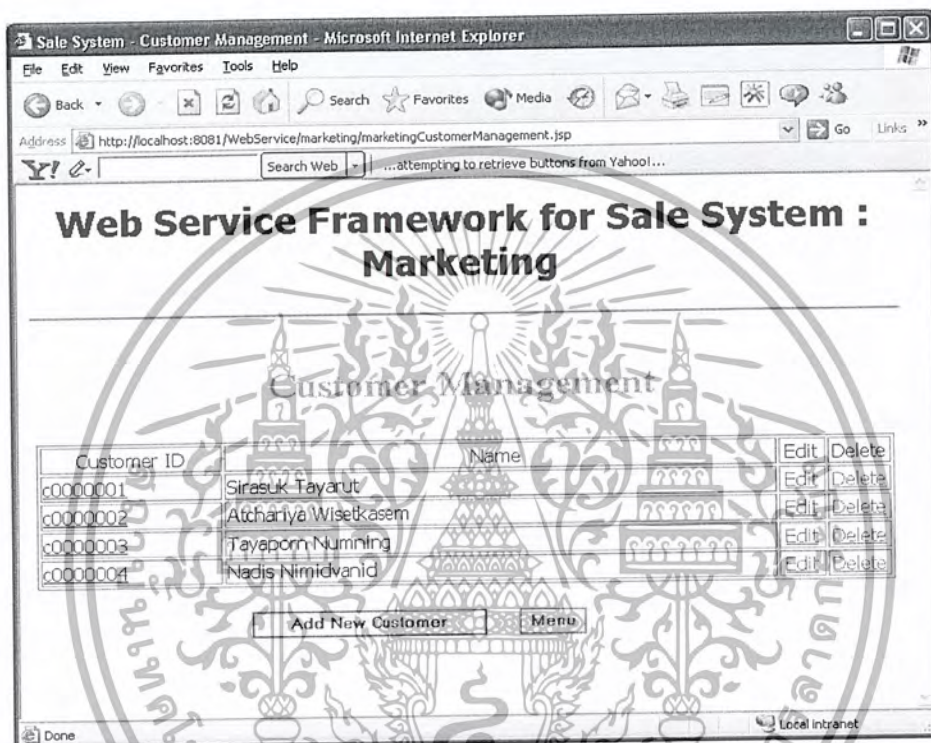


เมื่อเข้าสู่หน้าจอการ กิจกรรมหลักในฝ่าย จะสามารถเลือกได้ว่า ต้องการทำกิจกรรมใดดังต่อไปนี้

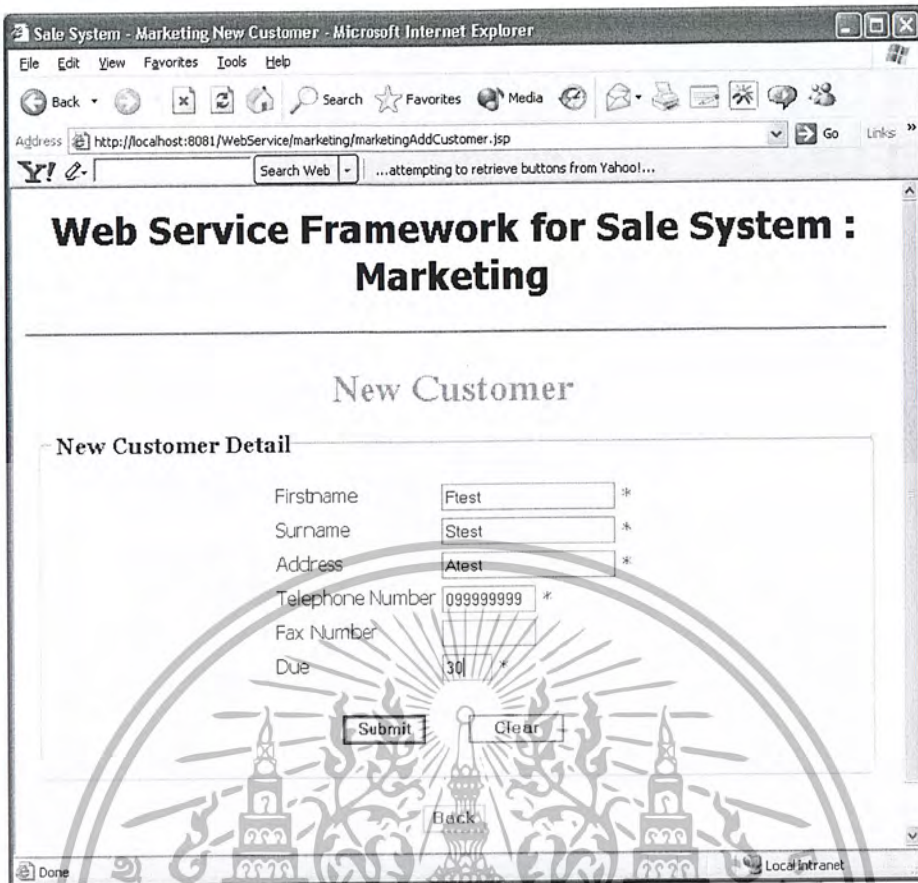
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 การจัดการข้อมูลลูกค้า

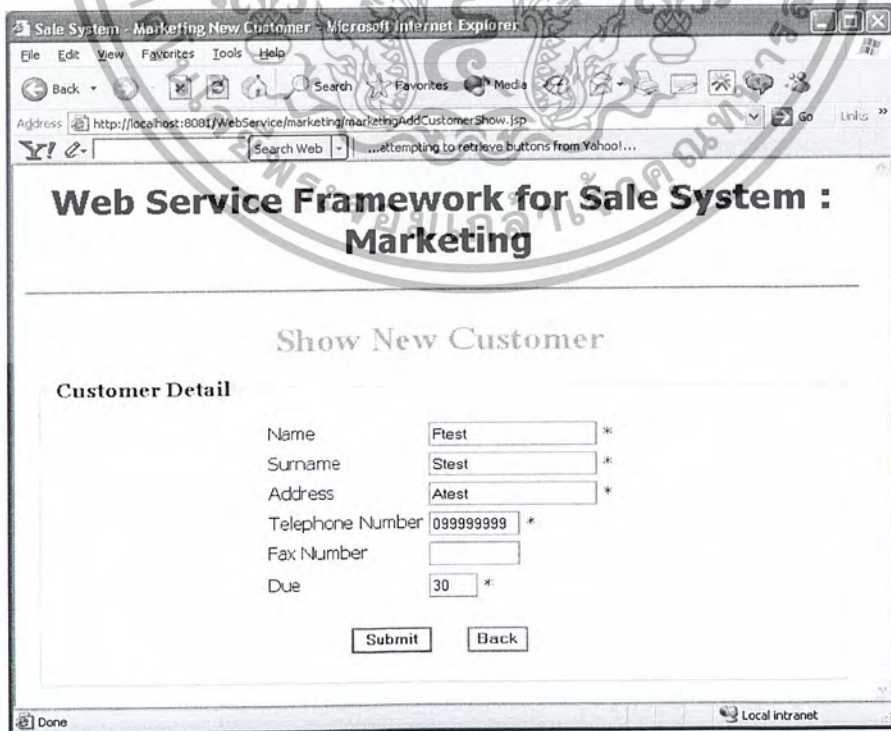
ที่หน้าจอการจัดการข้อมูลลูกค้า จะแสดงรายชื่อลูกค้าของทางบริษัททั้งหมด โดยสามารถจัดการข้อมูลลูกค้าโดยถ้าต้องการ แก้ไขข้อมูล และลบข้อมูลลูกค้าก็สามารถทำได้คล้ายกับ การที่ผู้ดูแลระบบ แก้ไขข้อมูล และลบข้อมูลพนักงาน แต่ในส่วนของการเพิ่มข้อมูลลูกค้าใหม่ จะแตกต่างออกไปเล็กน้อย โดย เลือก Add New Customer จากหน้าจอนี้



จะแสดงหน้าจอให้กรอกข้อมูลลูกค้าใหม่โดยต้องกรอกข้อมูลตามชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ และ วันที่กำหนดให้ลูกค้าคนนี้ค้างชำระกับทางบริษัทได้ โดยถ้าต้องการแก้ไขข้อมูลทั้งหมดสามารถเลือกปุ่ม Clear เพื่อลบข้อมูลทั้งหมดแล้วกรอกใหม่ได้ หรือเลือก Back เพื่อทำการกลับสู่หน้าจอก่อนหน้า และเลือก Submit เพื่อดำเนินการบันทึกข้อมูลลูกค้าใหม่

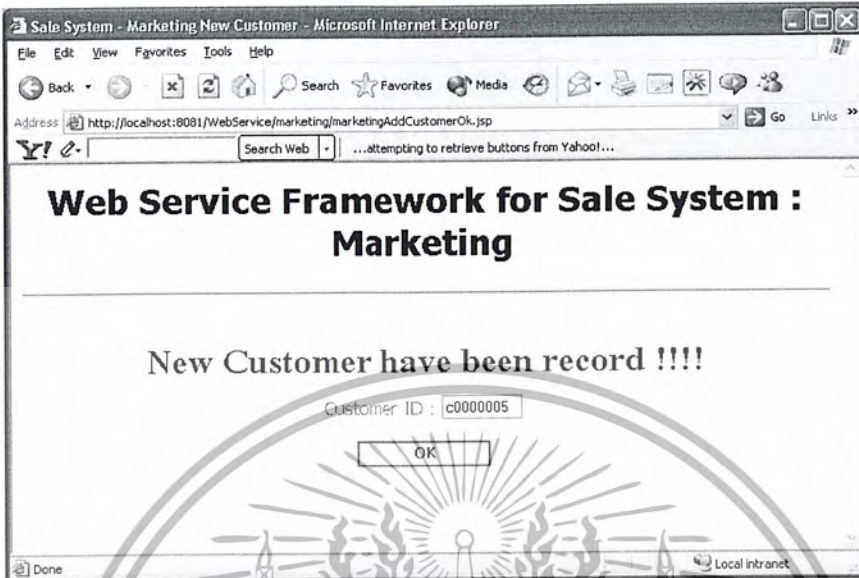


จะแสดงข้อมูลลูกค้าใหม่ที่ได้ทำการกรอกไปในหน้าที่แล้วเพื่อให้พนักงาน ยืนยัน ว่าต้องการจะบันทึกข้อมูลนี้ลงฐานข้อมูลหรือไม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

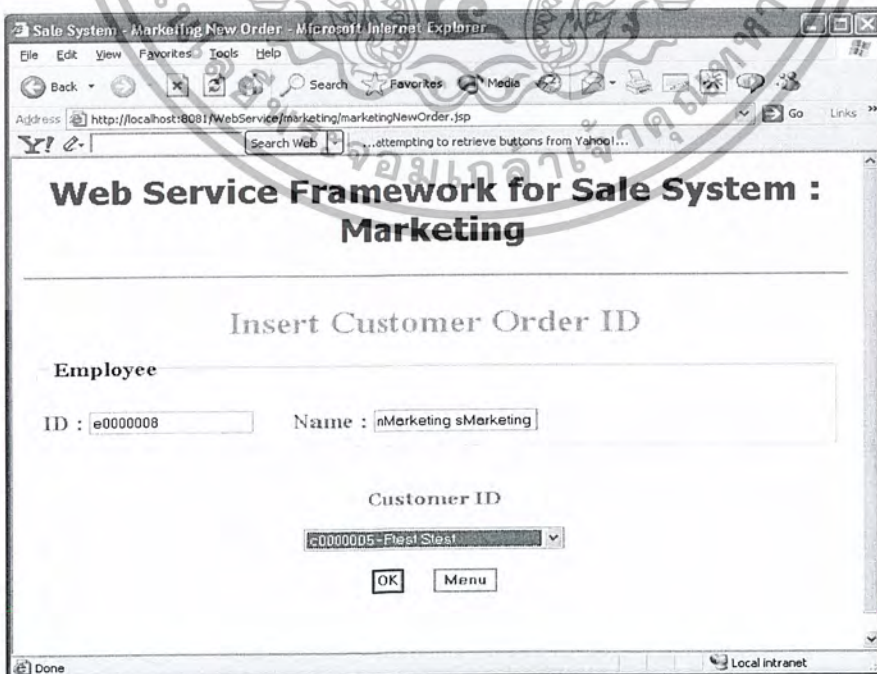
สามารถเลือกปุ่ม Submit เพื่อบันทึกข้อมูลลูกค้าใหม่ได้ และเลือกปุ่ม Back เมื่อยังคงต้องการแก้ไขข้อมูลลูกค้าใหม่



ระบบจะทำการสร้าง รหัสประจำตัวลูกค้าขึ้นให้กับลูกค้าโดยอัตโนมัติ โดยเป็นลำดับที่ต่อจากลูกค้ารายสุดท้ายในฐานข้อมูล

2.2 การรับใบสั่งซื้อสินค้า

เมื่อลูกค้าทำการสั่งซื้อสินค้าเข้ามา พนักงานจะเลือกปุ่ม การรับใบสั่งซื้อสินค้า เพื่อทำการบันทึกข้อมูลการสั่งซื้อ โดยจะแสดงหน้าจอ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้น หน้าจอจะแสดงรหัสประจำตัว และชื่อของพนักงานที่รับคำสั่งซื้อ และพนักงานสามารถเลือกรหัสประจำตัวลูกค้าได้ว่าลูกค้าคนใดต้องการสั่งซื้อสินค้า แล้วเลือกปุ่ม OK หน้าจอถัดไปด้านบนจะแสดงรายละเอียดของลูกค้าที่ต้องการสั่งซื้อ ด้านล่างจะเป็นส่วนในการเลือกสินค้าและกรอกจำนวนที่ต้องการสั่งซื้อ

No.	Product	Quantity
1.	p000000004 - The Lord of The Ring	14
2.	p000000005 - You got mail	60
3.	Choose One	
4.	Choose One	
5.	Choose One	
6.	Choose One	
7.	Choose One	
8.	Choose One	
9.	Choose One	
10.	Choose One	
11.	Choose One	
12.	Choose One	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเลือกปุ่ม Clear ได้เมื่อต้องการลบรายการสั่งซื้อทั้งหมดแล้วทำการเลือกสินค้าและใส่จำนวนที่ต้องการใหม่ และเลือก Order เพื่อดำเนินการสั่งซื้อสินค้าต่อ

The screenshot shows a web browser window titled "Sale System - Marketing New Order - Microsoft Internet Explorer". The address bar shows "http://localhost:8081/WebService/marketing/marketingNewOrderShowDetail.jsp". The page content is divided into two main sections:

Customer Detail

Customer Id : c0000005
 Name : Ftest Surname : Stest
 Address : Atest
 Telephone Number : 099999999

Order List

Receive Date : 11/April/2004

No.	Product ID	Quantity	Quantity In Stock	Status
1.	p000000004	14	645	สามารถส่งสินค้าได้
2.	p000000005	60	413	สามารถส่งสินค้าได้

Below the table are two buttons: "Order" and "Edit Order".

ในหน้าจอก็จะแสดงรายละเอียดว่าสินค้าที่ทำการสั่งซื้อ ทางบริษัทมีเพียงพอหรือไม่ ถ้ามีเพียงพอก็สามารถสั่งซื้อสินค้าได้ โดยกดปุ่ม Order เพื่อดำเนินการต่อไป แต่ถ้ายังคงต้องการแก้ไขการสั่งซื้อสามารถเลือก Edit Order ได้

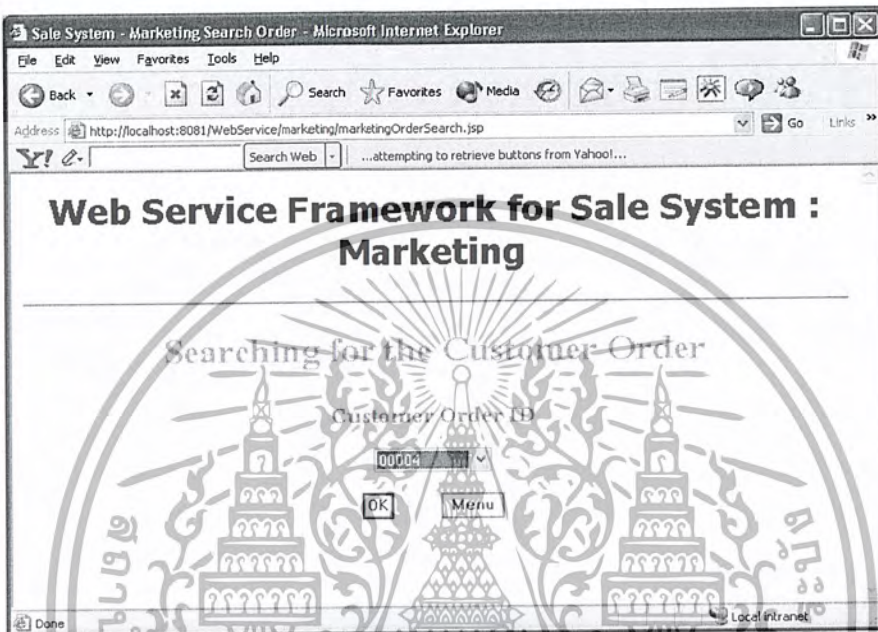
The screenshot shows a confirmation message in a web browser window. The title is "Web Service Framework for Sale System : Marketing". The main text reads "Your Data Have Been Record !!!!". Below the text is an "OK" button.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

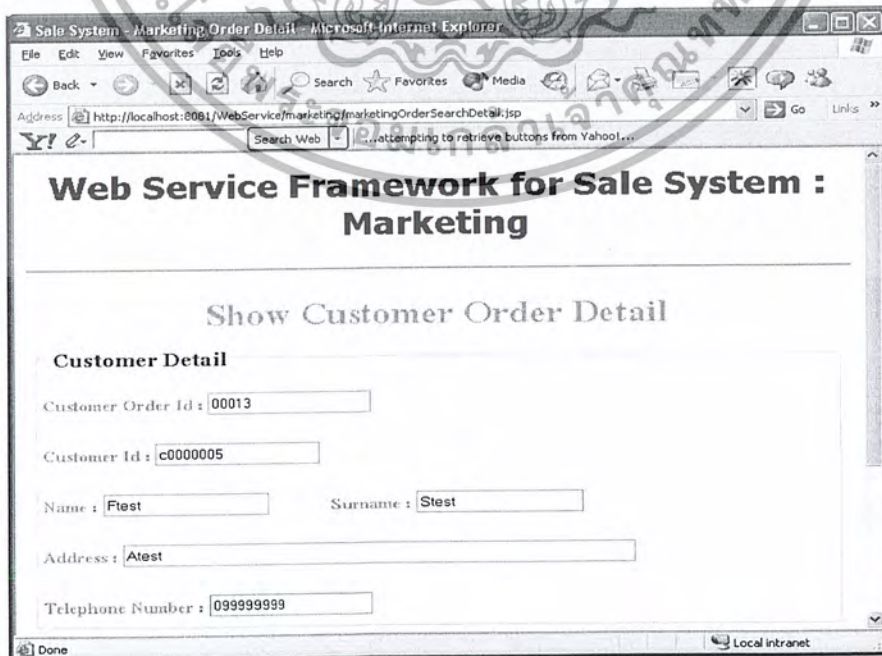
เมื่อเลือก Order แล้วจะแสดงหน้าจอว่า ได้ทำการบันทึกข้อมูลการสั่งซื้อลงในฐานข้อมูลแล้ว

2.3 การยกเลิกใบสั่งซื้อสินค้า

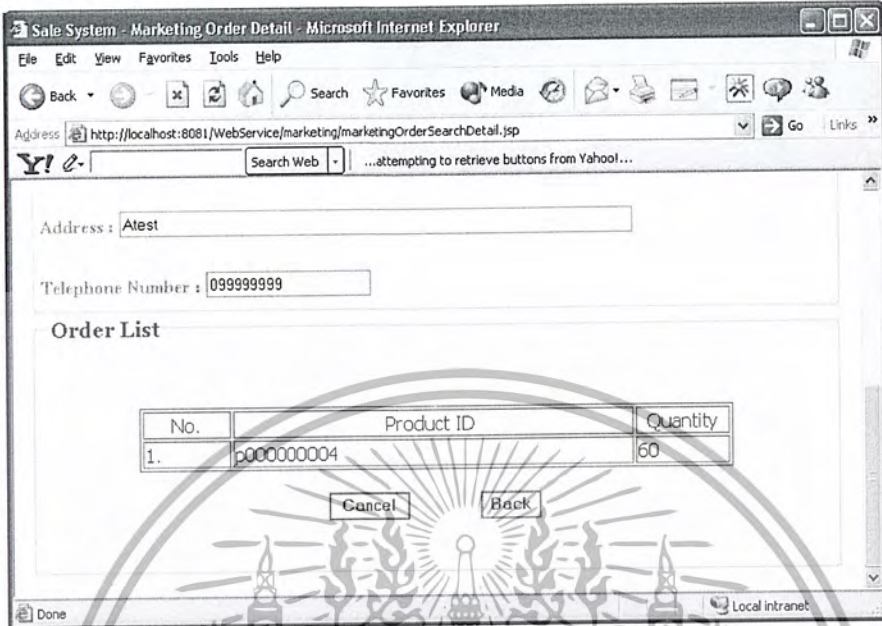
เมื่อเข้าสู่หน้าจอ การยกเลิกใบสั่งซื้อสินค้า พนักงานจะต้องทำการเลือก รหัสประจำตัวลูกค้า จากนั้นกดปุ่ม OK



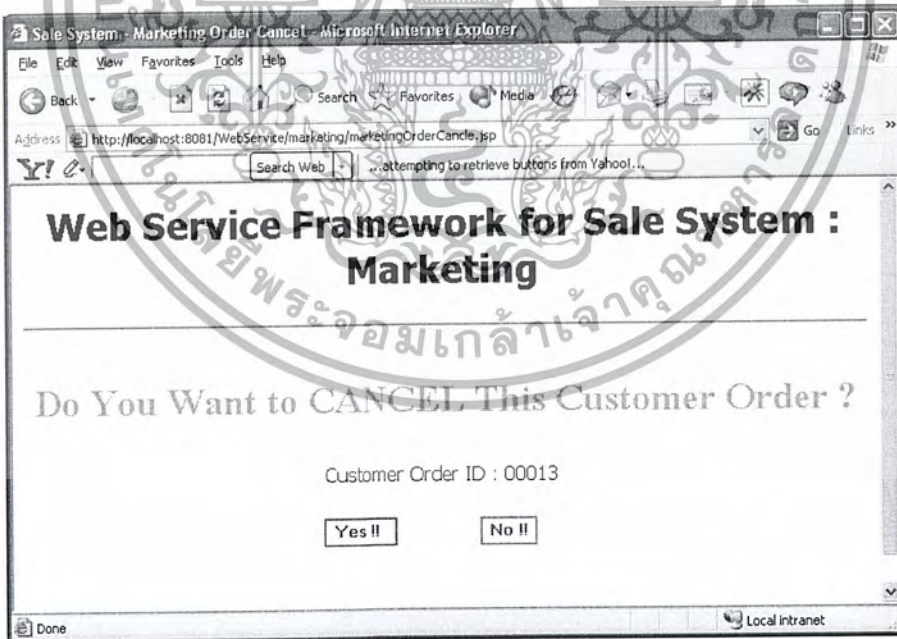
จากนั้นระบบจะแสดงหน้าจอที่ส่วนบนแสดงรายละเอียดของลูกค้า และส่วนล่างแสดงรายละเอียดของใบสั่งซื้อสินค้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



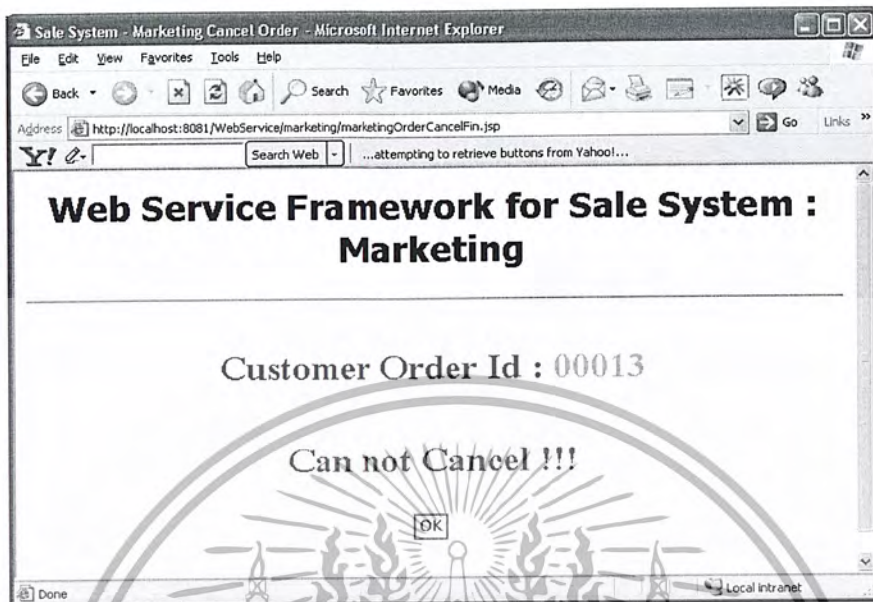
เลือกปุ่ม Cancel เมื่อต้องการลบใบสั่งซื้อสินค้านั้น



ระบบจะถามว่าต้องการจะลบใบสั่งซื้อสินค้า หมายเลขที่ระบุหรือไม่ ถ้าไม่ต้องการก็กด No !! แต่ถ้าต้องการกด Yes !!

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ใบสั่งซื้อสินค้านั้นได้ออกใบ Invoice ไปเรียบร้อยแล้วจะไม่สามารถลบได้
 ดังรูป

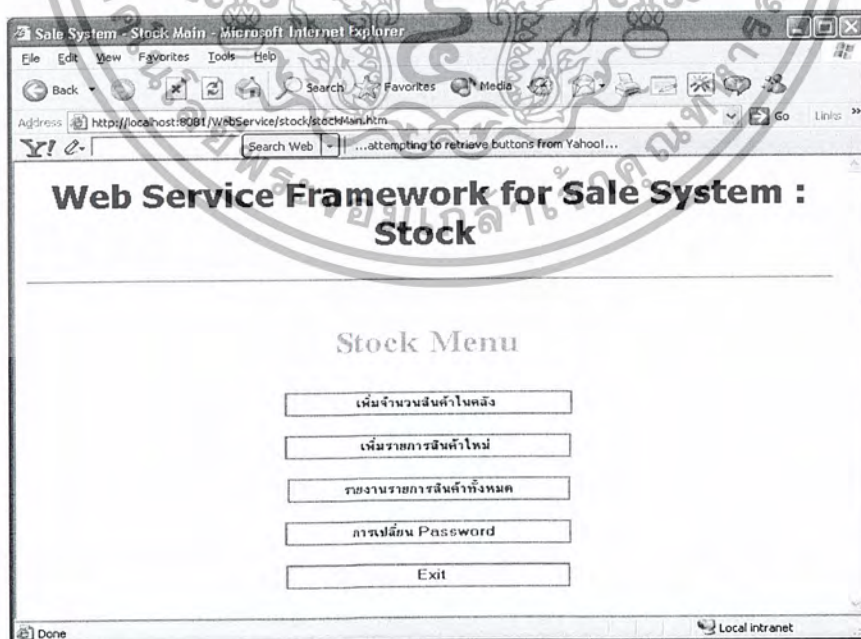


2.4 การเปลี่ยน Password

จะดำเนินการเช่นเดียวกับการเปลี่ยน Password ของผู้ดูแลระบบ

3. การใช้งานในฝ่ายจัดการสินค้า

เมื่อทำการ Login เข้าใช้งานระบบแล้วจะเข้าสู่ หน้ากิจกรรมหลักของฝ่าย ดังนี้

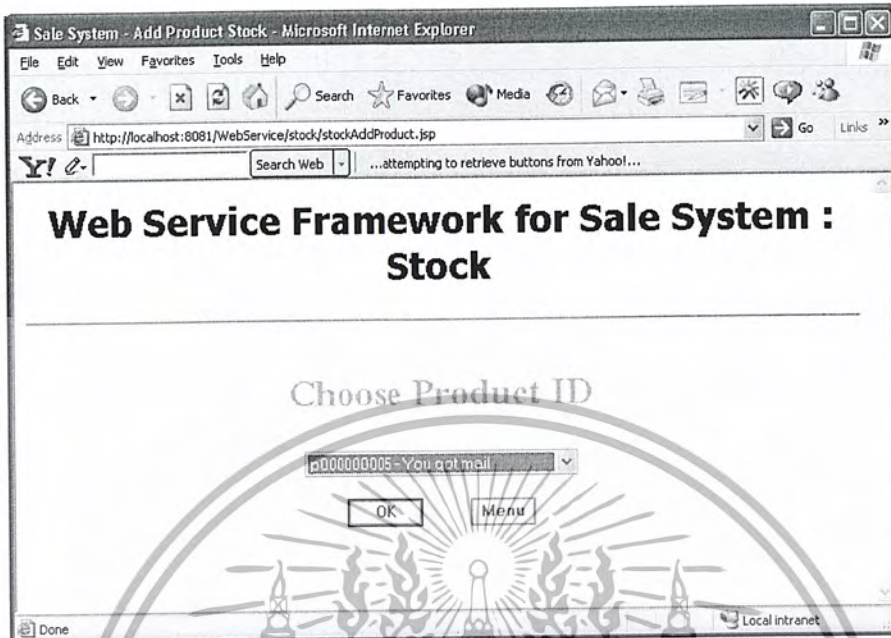


สามารถดำเนินการหลักได้ดังนี้

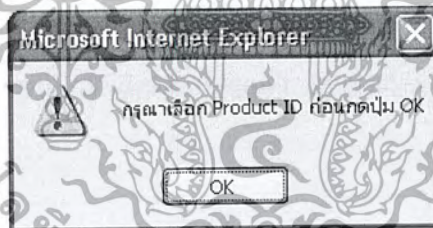
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 เพิ่มจำนวนสินค้าในคลังสินค้า

เป็นการเพิ่มจำนวนสินค้ารายการที่มีอยู่แล้ว ดังนี้



เมื่อเลือกรายการสินค้าได้แล้ว จึงกดปุ่ม OK กรณีที่ยังไม่มีการเลือกรายการสินค้าแต่กดปุ่ม OK ระบบจะแจ้งเตือนดังนี้



เมื่อกดปุ่ม OK แล้วจะแสดงรายการ รหัสประจำตัวสินค้า ชื่อสินค้า และจำนวนที่มีอยู่ในคลังสินค้า พนักงานสามารถกรอกข้อมูลเพื่อเพิ่มสินค้าได้ในช่อง จำนวนสินค้าที่ต้องการใส่เพิ่ม จากนั้นเลือกปุ่ม OK เพื่อดำเนินการต่อไป

**Web Service Framework for Sale System :
Stock**

Add Product Amount

Product Detail

Product ID :

Product Name :

Product Balance :

จำนวนสินค้าที่ต้องการใส่เพิ่ม :

จากนั้นระบบจะแสดงข้อมูลว่าขณะนี้สินค้าทั้งสิ้นจำนวนเท่าใด คือจำนวนของสินค้าเดิมที่เพิ่มจำนวนสินค้าใหม่แล้ว

**Web Service Framework for Sale System :
Stock**

Product Detail

Product ID :

Product Name :

Product Balance :

เลือกปุ่ม OK เพื่อบันทึกข้อมูลต่อไป

3.2 เพิ่มรายการสินค้าใหม่

เป็นการเพิ่มรายการสินค้าใหม่ที่ทางบริษัทไม่เคยมีมาก่อน จะประกอหน้าจอให้กรอดข้อมูลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sale System - Stock Add New Product - Microsoft Internet Explorer

Address: http://localhost:8081/WebService/stock/stockAddNewProduct.jsp

Web Service Framework for Sale System : Stock

New Product

Product Detail

Product Name : *

Product Detail :

Product Amount : *

Product Price : *

เลือก Add เพื่อทำการบันทึกข้อมูล และเลือก Menu เมื่อไม่ต้องการเพิ่มรายการสินค้า เมื่อเลือก Add จะแสดงข้อมูลที่ได้ออกไปทั้งหมด

Sale System - Stock Add New Product - Microsoft Internet Explorer

Address: http://localhost:8081/WebService/stock/stockAddNewProductDetail.jsp

Web Service Framework for Sale System : Stock

New Product Detail

Product Detail

Product ID :

Product Name :

Product Detail :

Product Balance :

Product Price :

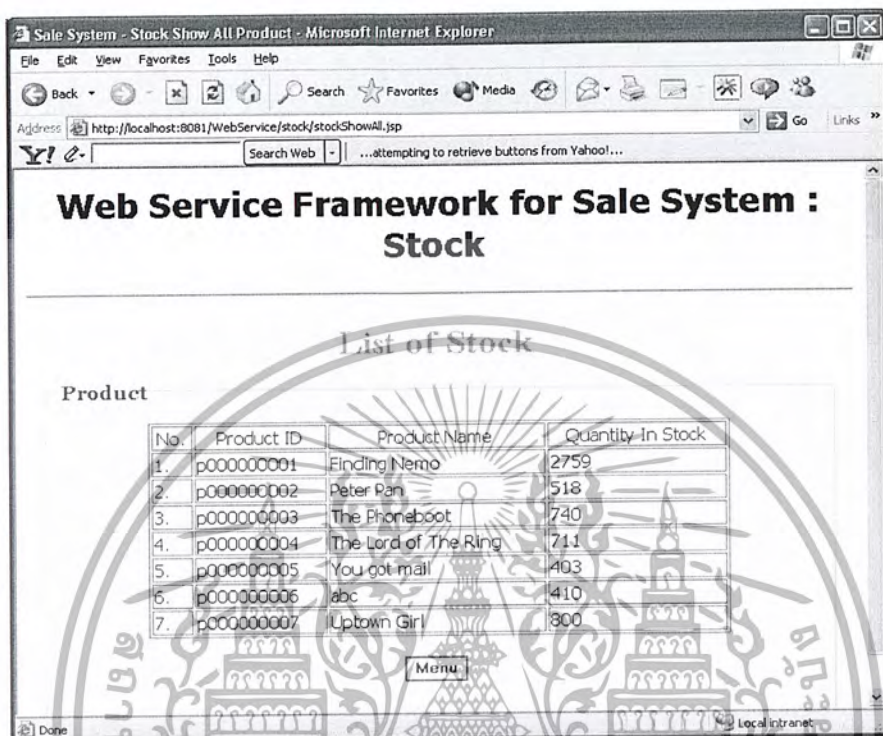
พร้อมกับระบบจะทำการสร้างรหัสสินค้าให้กับสินค้าใหม่ โดยอัตโนมัติโดยสร้าง

ต่อจาก รหัสสินค้าตัวสุดท้ายในฐานะข้อมูล จากนั้นเลือก OK เป็นอันเสร็จเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 รายงานรายการสินค้าทั้งหมด

เป็นการแสดงรายงานรายละเอียดของรายการสินค้าทั้งหมดของบริษัท เพื่อสร้างความสะดวกกับพนักงานในการตรวจสอบจำนวนสินค้าแต่ละชนิดขณะ เวลานั้น ดังนี้



Web Service Framework for Sale System : Stock

List of Stock

Product

No.	Product ID	Product Name	Quantity-In Stock
1.	p000000001	Finding Nemo	2759
2.	p000000002	Peter Pan	518
3.	p000000003	The Phoneboot	740
4.	p000000004	The Lord of The Ring	711
5.	p000000005	You got mall	403
6.	p000000006	abc	410
7.	p000000007	Uptown Girl	800

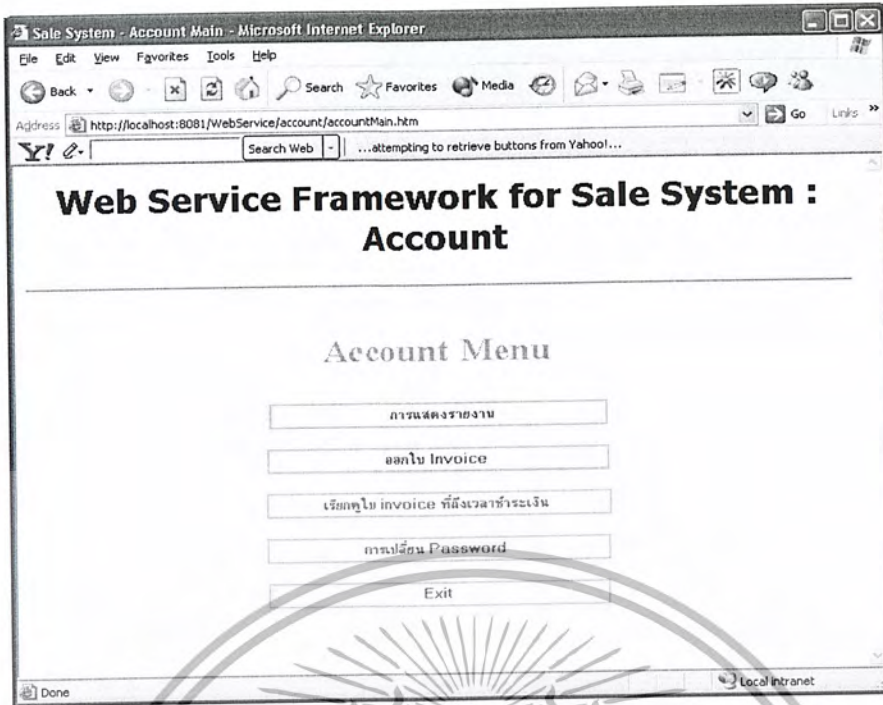
Menu

3.4 การเปลี่ยน Password

จะดำเนินการในลักษณะเดียวกับ การเปลี่ยน Password ของฝ่ายผู้ดูแลระบบ

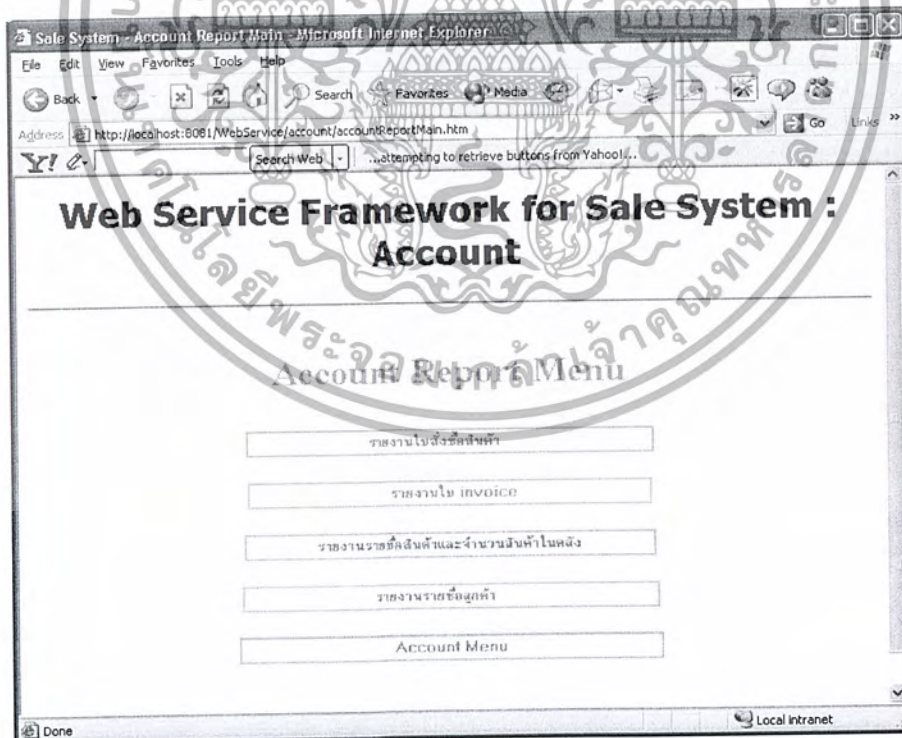
4. การใช้งานในฝ่ายบัญชี

เมื่อเข้าสู่หน้าจอการทำงานหลักจะประกอบด้วยกิจกรรม ดังนี้



4.1 การแสดงรายงาน

เมื่อเรียกดูจะปรากฏเมนูย่อยดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 รายงานซื้อสินค้า และจำนวนสินค้าในคลัง

จะแสดงรหัสสินค้า ชื่อสินค้า ราคาของสินค้าต่อหน่วย และจำนวนของสินค้าแต่ละชนิดในคลังสินค้า ดังรูป

Web Service Framework for Sale System : Account

Product Report

List

Product ID	Product Name	Price Per Unit	Product Amount
p000000001	Finding Nemo	450	3150
p000000002	Peter Pan	300	600
p000000003	The Phoneboot	250	300
p000000004	The Lord of The Ring	500	1000
p000000005	You got mail	400	550
p000000006	abc	125	450
p000000007	Uptown Girl	300	300

Menu

4.1.2 รายงานรายชื่อลูกค้า

จะแสดงรายงานรหัสประจำตัวลูกค้า ชื่อ นามสกุล ที่อยู่ และเบอร์โทรศัพท์ ของลูกค้าแต่ละคน ดังรูป

Web Service Framework for Sale System : Account

Customer Report

List

Customer ID	Name	Surname	Address	Telephone Number
c0000001	Sirasuk	Tayarut	2635 New Road	028754000
c0000002	Atchariya	Wisetkasem	19 Jalearnkrung	067878789
c0000003	Tayaporn	Numning	123 Bangmod	024529006
c0000005	Ftest	Stest	Atest	099999999
c0000004	Nadis	Nimidvanid	PTY Rd 4567-78 Duvall	064333212

Menu

4.1.3 Account Menu

เป็นการกลับไปยัง หน้ากิจกรรมหลักของฝ่ายบัญชี

4.2 ออกใบ Invoice

ระบบจะทำการแสดงใบสั่งซื้อสินค้าทั้งหมดที่ยังไม่ได้ออกใบ Invoice ขึ้นมาเพื่อให้พนักงานเลือกออกใบ Invoice โดยพนักงานสามารถเลือกออกใบ Invoice ได้หลายใบพร้อมๆ กันดังนี้

Web Service Framework for Sale System : Account

Create Invoice

List Customer Order

Customer Order ID	Customer ID	Order Date	Send Date
<input type="checkbox"/> 00001	c0000002	2004-03-14	2004-03-20
<input checked="" type="checkbox"/> 00004	c0000001	2004-04-07	2004-04-07
<input type="checkbox"/> 00007	c0000002	2004-04-07	2004-04-07
<input type="checkbox"/> 00008	c0000004	2004-04-07	2004-04-07
<input checked="" type="checkbox"/> 00009	c0000002	2004-04-08	2006-04-24
<input type="checkbox"/> 00010	c0000002	2004-04-08	2004-01-02
<input type="checkbox"/> 00011	c0000005	2004-04-11	2004-04-11
<input type="checkbox"/> 00012	c0000005	2004-04-11	2004-04-11
<input type="checkbox"/> 00013	c0000005	2004-04-11	2004-04-11

Submit Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเลือกปุ่ม Submit เพื่อทำการออกใบ Invoice ใบสั่งซื้อสินค้าที่เลือกไว้ จากนั้นระบบจะแสดงรายละเอียดของใบสั่งซื้อสินค้าทุกใบที่ต้องการออกใบ Invoice พร้อมทั้งรายละเอียดลูกค้าที่เกี่ยวข้องกับใบสั่งซื้อสินค้านั้น ดังนี้

Sala System - Create Invoice - Microsoft Internet Explorer

Address http://localhost:8081/WebService/account/accountCreateInvoiceDetail.jsp

Web Service Framework for Sale System : Account

Show Invoice Detail

Customer Order No.00004

Employee ID : e0000009 Name : nAccount sAccount

Customer ID : c0000001 Name : Sirasak Teyarut
Address : 2635 New Road

Product Order

NO.	Product ID	Product Name	Amount
1	p000000001	Finding Nemo	15
2	p000000003	The Phoneboot	60
3	p000000005	You got mail	32
Total Amount			107
Total Price			34550.0

Done Local intranet

Sala System - Create Invoice - Microsoft Internet Explorer

Address http://localhost:8081/WebService/account/accountCreateInvoiceDetail.jsp

Customer Order No.00009

Employee ID : e0000009 Name : nAccount sAccount

Customer ID : c0000002 Name : Atchariya Wisetkasem
Address : 19 Jaleankrung

Product Order

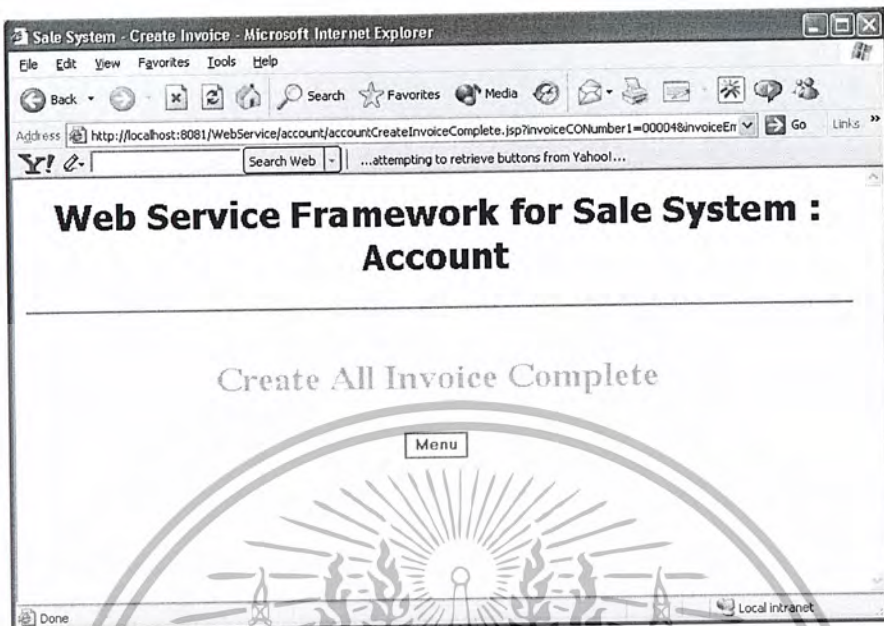
NO.	Product ID	Product Name	Amount
1	p000000001	Finding Nemo	20
Total Amount			20
Total Price			9000.0

Create Edit

Done Local intranet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

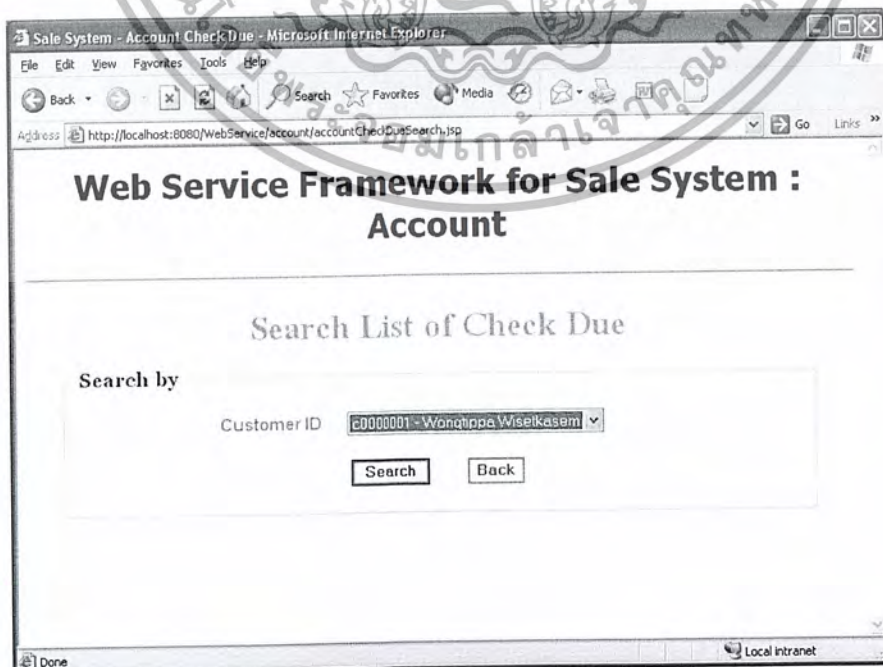
เลือกปุ่ม Create เพื่อทำการออกไป Invoice ตามใบสั่งซื้อสินค้าที่เลือก หรือเลือก Edit เพื่อทำการเลือกออกไป Invoice ตามใบสั่งซื้อสินค้าใหม่ หลังจากกด Create จะแสดงหน้าจอ



เพื่อเป็นการแจ้งให้ทราบว่าได้ทำการออกไป Invoice เรียบร้อยแล้ว

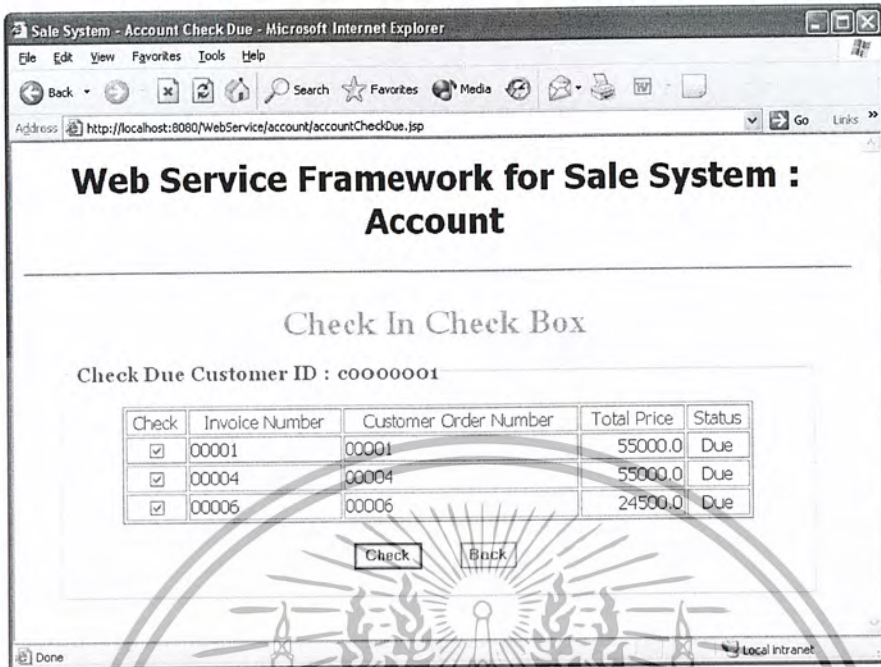
4.3 เรียกดูใบ Invoice ที่ถึงเวลาชำระเงิน

จะแสดงหน้าจอให้พนักงานเลือกว่าต้องการดูว่าลูกค้าคนใดบ้างที่มีใบ Invoice ใกล้เคียงเวลาชำระเงิน โดยเลือกรหัสประจำตัวลูกค้าดังรูป

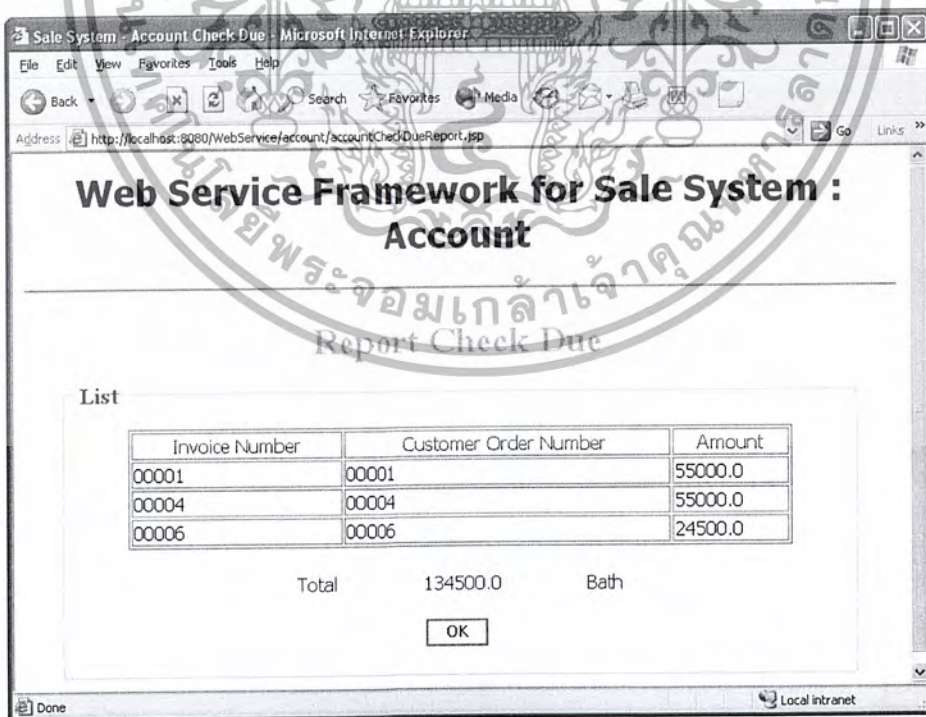


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเลือกปุ่ม Search เพื่อทำการหาข้อมูล



จากนั้นสามารถเลือกได้ว่าต้องการ แจงใบ Invoice ใตบ้างที่ควรจะชำระค่าสินค้า โดยดูได้จาก Status ที่ระบุว่า Due หมายถึงใกล้ถึงเวลาชำระค่าสินค้าแล้ว เมื่อเลือกได้แล้ว กดปุ่ม Check



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าระบบจะทำการแสดงจำนวนเงินตามใบ Invoice นั้นออกมา เลือกปุ่ม OK เพื่อ
ดำเนินการกลับสู่หน้าจอกิจกรรมหลักต่อไป

4.4 การเปลี่ยน Password

การเปลี่ยน Password จะดำเนินการเช่นเดียวกับการเปลี่ยน Password ในฝ่ายผู้ดูแลระบบ

