

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบจัดการข้อมูลสินค้าคงคลังแบบฝากขาย  
Information System for Consignment Inventory System

โดย

นายปกครอง รอดบุญ

รหัส 43067160



\*H001943\*

อาจารย์ที่ปรึกษา

ดร.จันทร์บูรณ์ สถิตวิริวงศ์

วัน เดือน ปี.....	23 ธ.ค. 2555
เลขทะเบียน.....	01943
เลขเรียกหนังสือ.....	วท. ป115 ร 2545
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2545

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ระบบจัดการข้อมูลสินค้าคงคลังแบบฝากขาย
นักศึกษา	นายปกครอง รอดบุญ
อาจารย์ที่ปรึกษา	ดร.จันทร์บูรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

### บทคัดย่อ

ระบบจัดการข้อมูลสินค้าคงคลังแบบฝากขาย เป็นการศึกษาถึงปัญหาและความต้องการของเจ้าหน้าที่ หรือผู้ที่เกี่ยวข้องกับการดำเนินงาน ทั้งพนักงานดูแลสินค้าคงคลัง พนักงานขายตามจุดขาย ผู้ดูแลด้านการตลาด ซึ่งในปัจจุบันจะเก็บข้อมูลโดยการใช้แฟ้มข้อมูล เอกสารบันทึกข้อมูล อันจะก่อให้เกิดปัญหาในเรื่องความล่าช้า ผิดพลาด ดังนั้นจึงควรนำคอมพิวเตอร์เข้ามาใช้งานในระบบนี้

ระบบจัดการข้อมูลสินค้าคงคลังแบบฝากขาย จะถูกจัดการพัฒนาขึ้นเพื่อแก้ไขปัญหาดังกล่าวข้างต้น และเพิ่มประสิทธิภาพในการทำงาน เพื่อตอบสนองตามความต้องการของผู้ใช้งาน

<b>Title</b>	Information System for Consignment Inventory System
<b>Student</b>	Mr. Pokkhrong Rodboon
<b>Advisor</b>	Dr. Chanboon Satitwiriya Wong
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2002

## ABSTRACT

This development project is to studying the problems of the Consignment Inventory System, requirement of its and solve the problems and redesign procedure to help the system increase efficient correctness and speed the current system which is still using manual system to a more efficient computerize system.

Information System for Consignment Inventory System was developed to solve the problems in management of Consign and Sales department and make satisfied to the person with involve the system.

## กิตติกรรมประกาศ

ขอขอบพระคุณ บิดา มารดา และเพื่อนๆของข้าพเจ้า ที่คอยเป็นกำลังใจ ในการส่งเสริม ซึ่ง  
แนะแนวทาง การศึกษาการเรียนรู้ รวมทั้งอาจารย์ทุกท่านที่ได้ถ่ายทอดความรู้ เพื่อใช้เป็นพื้นฐานใน  
การเรียนรู้สิ่งใหม่ต่อไปในอนาคต

ขอขอบพระคุณอาจารย์ที่ปรึกษา คร.จันทร์บูรณ์ สถิตวิริยวงศ์ ซึ่งช่วยในการตรวจอ่าน  
และตั้งคำถาม เพื่อให้ข้าพเจ้าได้นำมาปรับปรุงเนื้อหา และรูปแบบการนำเสนอให้สมบูรณ์ยิ่งขึ้น

สุดท้ายหากรายงานฉบับนี้มีความผิดพลาดประการใด ต้องขออภัยไว้ ณ ที่นี้ด้วย

นายปกครอง รอดบุญ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่	
1. บทนำ.....	1
1.1 เป้าหมายของโครงการ.....	2
1.2 วัตถุประสงค์ของโครงการ.....	3
1.3 ขอบเขตของการศึกษา.....	3
1.4 ขั้นตอนการดำเนินงาน.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
2. ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ภาษาแบบจำลองสำหรับการพัฒนา โปรแกรมเชิงวัตถุ.....	5
2.2 การวิเคราะห์และออกแบบ.....	20
2.3 JAVA SERVER PAGES.....	26
2.4 JDBC.....	30
3. วิธีดำเนินการและการรวบรวมข้อมูล.....	34
3.1 กรณีศึกษา.....	34
3.2 เครื่องมือที่ใช้ในโครงการ.....	34
3.3 การเก็บรวบรวมข้อมูล.....	35
3.4 การวิเคราะห์และขั้นตอนการพัฒนา.....	35
4. การวิเคราะห์และออกแบบระบบ.....	37
4.1 การดำเนินการในปัจจุบัน.....	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.2 การวิเคราะห์และออกแบบ.....	45
5. การติดตั้งระบบ.....	70
5.1 การติดตั้ง JAVA 2 SDK.....	70
5.2 การแก้ไขตัวแปร Path และ Class Path ของ JAVA 2 SDK .....	73
5.3 การติดตั้งเซิร์ฟเวอร์ TOMCAT .....	74
6. การพัฒนาระบบงาน.....	78
7. สรุป.....	87
7.1 บทสรุป.....	87
7.2 ข้อเสนอแนะ.....	87
บรรณานุกรม.....	88
ประวัติผู้เขียน.....	89

## สารบัญตาราง

	หน้า
ตารางที่	
4.1 เทเบิล p_size ใช้เก็บข้อมูลขนาดของเสื้อผ้า.....	63
4.2 เทเบิล p_color ใช้เก็บข้อมูลสีของเสื้อผ้า.....	63
4.3 เทเบิล p_group ใช้เก็บข้อมูลกลุ่มสินค้า.....	63
4.4 เทเบิล products ใช้เก็บข้อมูลสินค้า.....	64
4.5 เทเบิล outlet ใช้เก็บข้อมูลร้านค้า.....	64
4.6 เทเบิล dept ใช้เก็บข้อมูลแผนกต่างๆในองค์กร.....	65
4.7 เทเบิล employee ใช้เก็บข้อมูลพนักงานในองค์กร.....	65
4.8 เทเบิล user-prg ใช้เก็บข้อมูลสิทธิในการใช้งานของพนักงาน.....	66
4.9 เทเบิล product_color ใช้เก็บข้อมูลความสัมพันธ์ระหว่างสินค้าและสี.....	66
4.10 เทเบิล product_size ใช้เก็บข้อมูลความสัมพันธ์ระหว่างสินค้าและขนาด.....	66
4.11 เทเบิล stock ใช้เก็บข้อมูลยอดคงเหลือสินค้า.....	67
4.12 เทเบิล tx_bill ใช้เก็บข้อมูลรายการขายของแต่ละสินค้าในแต่ละวัน.....	67
4.13 เทเบิล tx_cn ใช้เก็บข้อมูลรายการคืนสินค้ากลับโรงงานขายของแต่ละสินค้าในแต่ละวัน.....	68
4.14 เทเบิล tx_return ใช้เก็บข้อมูลรายการรับคืนสินค้าจากลูกค้าของแต่ละสินค้าในแต่ละวัน.....	68
4.15 เทเบิล tx_order ใช้เก็บข้อมูลรายการรับสินค้าเข้าคลังตามสาขาของแต่ละสินค้าในแต่ละวัน.....	69
4.16 เทเบิล order_detail ใช้เก็บข้อมูลย่อยการรับสินค้าเข้าคลังตามสาขาของแต่ละสินค้าในแต่ละวัน.....	69
5.1 แสดงไคเร็กทอรี่และความหมายของ TOMCAT .....	74

## สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างแบบจำลอง Use Case .....	8
2.2 ตัวอย่างแบบจำลอง Class.....	10
2.3 ตัวอย่างแบบจำลอง Object.....	11
2.4 สัญลักษณ์แบบจำลอง State.....	12
2.5 ตัวอย่างแบบจำลอง State.....	13
2.6 ตัวอย่างแบบจำลอง Sequence.....	14
2.7 แบบจำลอง Collaboration.....	15
2.8 ตัวอย่างแบบจำลอง Activity.....	17
2.9 ตัวอย่างแบบจำลอง Component.....	18
2.10 ตัวอย่างแบบจำลอง Deployment.....	19
2.11 การเพิ่ม Classes จาก ขั้นตอน Analysis ในขั้นตอนการ Design.....	23
2.12 การทำงานของ JSP.....	28
2.13 สถาปัตยกรรมของ JSP.....	29
2.14 JDBC.....	30
4.1 แสดงระบบงานปัจจุบัน.....	39
4.2 แสดงขั้นตอนการสั่งซื้อจากโรงงาน.....	40
4.3 แสดงขั้นตอนการจัดการใบรายงานยอดขาย และ ใบ PO.....	40
4.4 แสดงขั้นตอนการตรวจนับสต็อก.....	41
4.5 แสดงภาพรวมของระบบในรูปแบบ PACKAGE.....	45
4.1 แสดงระบบงานปัจจุบัน.....	39
4.2 แสดงขั้นตอนการสั่งซื้อจากโรงงาน.....	40
4.3 แสดงขั้นตอนการจัดการใบรายงานยอดขาย และ ใบ PO.....	40
4.4 แสดงขั้นตอนการตรวจนับสต็อก.....	41
4.5 แสดงภาพรวมของระบบในรูปแบบ PACKAGE.....	45

## สารบัญรูป(ต่อ)

รูปที่	หน้า
4.6 แสดง USE CASE DIAGRAM ของ PACKAGE OUTLET SALES.....	46
4.7 แสดง USE CASE DIAGRAM ของ PACKAGE ORDER.....	47
4.8 แสดง USE CASE DIAGRAM ของ PACKAGE INVENTORY.....	47
4.9 แสดง USE CASE DIAGRAM ของ PACKAGE SITE.....	48
4.10 แสดง CLASS DIAGRAM ของระบบ.....	50
4.11 แสดงองค์ประกอบของ CLASS PRODUCTS.....	51
4.12 แสดงองค์ประกอบของ CLASS P_SIZE.....	51
4.13 แสดงองค์ประกอบของ CLASS P_COLOR.....	52
4.14 แสดงองค์ประกอบของ CLASS OUTLET.....	52
4.15 แสดงองค์ประกอบของ CLASS STOCK.....	53
4.16 แสดงองค์ประกอบของ CLASS TX_BILL.....	53
4.17 แสดงองค์ประกอบของ CLASS TX_CN.....	54
4.18 แสดงองค์ประกอบของ CLASS TX_RETURN.....	54
4.19 แสดงองค์ประกอบของ CLASS EMPLOYEE.....	55
4.20 แสดงองค์ประกอบของ CLASS USER_PRG.....	55
4.21 แสดงองค์ประกอบของ CLASS P_GROUP.....	56
4.22 แสดง Sequence Diagram ของการบันทึกการขาย.....	57
4.23 แสดง Sequence Diagram ของการบันทึกการขายทำสินค้า.....	58
4.24 แสดง Sequence Diagram ของการบันทึกการขายการรับคืนสินค้าจากลูกค้า.....	59
4.25 แสดง Sequence Diagram ของการบันทึกการปรับปรุงยอดสต็อก.....	60
4.26 แสดง Sequence Diagram ของการบันทึกการปรับปรุงยอดสต็อก.....	61
4.27 แสดง E-R Diagram.....	62
5.1 เริ่มติดตั้ง Java 2SDK.....	70
5.2 เลือก Directory สำหรับติดตั้ง Java 2SDK.....	71
5.3 เลือก Components สำหรับติดตั้ง Java 2SDK.....	72

## สารบัญรูป(ต่อ)

รูปที่	หน้า
5.4	ติดตั้งเสร็จสมบูรณ์..... 72
5.5	แสดง System Properties..... 73
5.6	แสดงการ Set path และ Class path..... 74
5.7	แสดงทดสอบการเรียก Tomcat จาก C:\Tomcat\Bin\StarUp.bat..... 76
5.8	แสดงทดสอบการเรียก HOME PAGES ของ Tomcat..... 77
6.1	แสดงหน้าแรกของระบบงาน..... 78
6.2	แสดง Menu หลักของระบบงาน..... 79
6.3	แสดง Manage Stock Routine Transaction Menu..... 80
6.4	แสดง Manage Saies Transaction Menu..... 80
6.5	แสดงหน้า Sub Menu หลักของระบบงาน..... 81
6.6	แสดงหน้า Outlet Control Menu ของระบบงาน..... 82
6.7	แสดงหน้าการเพิ่มข้อมูลจุดขายของระบบงาน..... 83
6.8	แสดงหน้าเมื่อการเพิ่มข้อมูลสำเร็จ..... 83
6.9	แสดงหน้าจอเมื่อต้องการการแก้ไขข้อมูล..... 84
6.10	แสดงหน้าจอการแก้ไขข้อมูล..... 85
6.11	แสดงหน้าจอเมื่อการแก้ไขข้อมูลเรียบร้อย..... 85
6.12	แสดงหน้าจอเมื่อต้องการลบข้อมูล..... 86
6.13	แสดงหน้าจอเมื่อการลบข้อมูลเรียบร้อย..... 86

# บทที่ 1

## บทนำ

ในปัจจุบันได้มีการพัฒนาเทคโนโลยีให้มีความก้าวหน้าเพื่ออำนวยความสะดวกแก่มนุษย์มากขึ้น เทคโนโลยีคอมพิวเตอร์ เป็นส่วนหนึ่งของเทคโนโลยีที่ได้มีการพัฒนา และได้เข้ามาเป็นส่วนหนึ่งในการดำเนินชีวิตของมนุษย์ โดยการนำคอมพิวเตอร์เข้ามาประยุกต์ใช้งานในระบบงานต่าง ๆ เพื่อเพิ่มประสิทธิภาพในการทำงาน ทั้งในด้านความรวดเร็วถูกต้อง แม่นยำ ลดความผิดพลาด และประหยัดงบประมาณ ทั้งนี้จะต้องมีโปรแกรมที่เหมาะสมกับการใช้งานในด้านต่าง ๆ โดย อาจจะเป็นโปรแกรมสำเร็จรูป หรือเป็น โปรแกรมที่พัฒนาขึ้นเพื่อใช้ในงานเฉพาะด้านนั้น ๆ

ซึ่งในการบริหารงานสมัยใหม่ที่มีเทคโนโลยีไอทีเป็นแกนนำจึงมิได้สนใจเฉพาะตนเองหรือองค์กรของตนเองเท่านั้น แต่จะต้องเชื่อมโยงออกไปยังนอกองค์กร มีการสร้างกิจกรรมร่วมกันต่าง ๆ มากมาย ทำให้ผู้บริหารสมัยใหม่ต้องสร้างแนวคิดในการดำเนินการที่กว้างออกไป และสร้างกิจกรรมร่วมกับองค์กรอื่น เช่นการสร้างพันธมิตร การสร้างธุรกิจร่วมในลักษณะคู่ค้า เป็นต้น

ระบบเครือข่ายอินเทอร์เน็ต ก็เป็นสื่อทำให้การสื่อสารระหว่างองค์กรในหลายด้านเปลี่ยนแปลงไป โดยมีการใช้ระบบคอมพิวเตอร์ในการติดต่อสื่อสารแลกเปลี่ยนกันหรือมีการดำเนินการร่วมกันในลักษณะคู่ค้าหรือดำเนินธุรกิจแบบพึ่งพาซึ่งกันและกัน เพื่อเพิ่มขีดความสามารถในการทำงาน และลดการดูแลหรือการติดตามการทำงาน โดยมอบหน้าที่ให้ระบบได้ติดตามงานและดำเนินการบางอย่างแทน

โดยได้มีการนำบริการบนเว็บมาใช้ในการบริการ ทำให้มีการนำไปประยุกต์ใช้งานในรูปแบบต่างๆ อีกมากมาย โดยเฉพาะการดำเนินการภายในองค์กรแบบอิเล็กทรอนิกส์ และทั้งการทำงานภายในองค์กรเดียวกันหรือระหว่างองค์กร เช่นควบคุมสินค้าคงคลังระหว่างหน่วยงาน การจัดซื้อแบบอิเล็กทรอนิกส์ ซึ่งเราจะเห็นได้อย่างชัดเจนว่า การเชื่อมโยงข้อมูลข่าวสารระหว่างหน่วยงาน เพื่อเข้าสู่การบริการผ่านเว็บในปัจจุบัน จะใช้ HTML อย่างเดียวไม่ได้ การจัดการข้อมูลและวิธีการแลกเปลี่ยนข้อมูลจำเป็นต้องมีมาตรฐานการจัดการบริหารข้อมูลที่ดี เพื่อการแลกเปลี่ยนข่าวสารใหม่กันและกัน รวมทั้งการติดต่อกับระบบฐานข้อมูลเพื่อเพิ่มประโยชน์ของการทำงานผ่านเว็บแอปพลิเคชัน ซึ่งการพัฒนาโปรแกรมบนผ่านเว็บเป็นที่นิยมมากขึ้นก็ได้มีการคิดค้นเทคโนโลยีใหม่ๆ ออกมามากมาย ภาษาจาวาเองก็เช่นกันได้มีการปรับปรุงตัวเอง และได้ออกมาเป็นเทคโนโลยีที่เราเรียกว่า JSP (Java Server Pages) โดยพัฒนาอยู่บนพื้นฐานของภาษาจาวาเพื่อเพิ่มประสิทธิภาพให้

หน้าเว็บเพจมีความยืดหยุ่นสูงขึ้น ซึ่งการพัฒนาโปรแกรมบนเว็บนี้ได้เข้ามามีบทบาทอย่างมากใน  
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อคุณผู้ใดเห็นนำไปเผยแพร่จะถือว่าผิดกฎหมาย  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบัน จึงมีการนำบริการบนเว็บมาใช้เพื่อการบริหาร ทำให้มีการนำไปประยุกต์ใช้งานในแบบต่างๆ อีกมากมาย โดยเฉพาะการดำเนินการภายในองค์กรแบบอิเล็กทรอนิกส์ และการทำงานระหว่างองค์กร เช่นควบคุมสินค้าคงคลังระหว่างหน่วยงาน การจัดซื้อแบบอิเล็กทรอนิกส์ เป็นต้น

ระบบการจัดการข้อมูลสินค้าคงคลังแบบฝากขาย เป็นระบบหนึ่งที่จะนำคอมพิวเตอร์เข้ามาใช้เพื่อเพิ่มประสิทธิภาพในการทำงาน เนื่องจากระบบดังกล่าว จำเป็นที่จะต้องจัดเก็บข้อมูลเป็นจำนวนมากไม่ว่าจะเป็น ข้อมูลบุคลากรของระบบ ข้อมูลจุดขายและรายละเอียด ข้อมูลสินค้าและรายละเอียด เช่น สี ขนาด เป็นต้น ข้อมูลการขายสินค้ารายวัน และข้อมูลการเคลื่อนไหวสินค้าแต่ละจุดขาย ซึ่งข้อมูลที่กล่าวมานั้นล้วนแล้วแต่มีความสำคัญยิ่ง ซึ่งปัจจุบันข้อมูลมีปริมาณมาก เพราะมีชนิดสินค้าหลายสิบชนิด หลายขนาด และหลายสี รวมทั้งมีจุดขายมากถึง 50 จุดขายและอยู่กระจายกันทั่วกรุงเทพ ซึ่งถ้ายังคงเก็บและทำงานในรูปของเอกสาร อาจขาดต่อการค้นหาและเกิดการชำรุด สูญหายได้ง่าย รวมทั้งข้อมูลคงคลังสินค้า จากการที่ผู้ใช้การจดบันทึกข้อมูล ถ้าไม่มีการจัดการที่ดี อาจจะไม่ทราบได้ว่าสินค้าชนิดใดเหลือจำนวนที่แน่นอนเท่าใด ซึ่งอาจทำให้สินค้าที่ต้องการขายไม่เพียงพอต่อการจำหน่าย อันจะทำให้สูญเสียรายได้และอาจสร้างความไม่พอใจในบริการให้กับลูกค้าได้

ระบบการจัดการข้อมูลสินค้าคงคลังแบบฝากขาย เป็นระบบหนึ่งซึ่งนำคอมพิวเตอร์เข้ามาใช้ในการจัดการข้อมูลในด้านต่าง ๆ อันได้แก่ การเก็บข้อมูลบุคลากรของระบบ ข้อมูลจุดขายและรายละเอียด ข้อมูลสินค้าและรายละเอียด เช่น สี ขนาด เป็นต้น ข้อมูลการขายสินค้ารายวัน และข้อมูลการเคลื่อนไหวสินค้าแต่ละจุดขาย นอกจากนี้ยังช่วยในด้านการออกเอกสารต่าง ๆ จากเดิมซึ่งต้องบันทึกด้วยมือ ซึ่งช่วยอำนวยความสะดวกและเพิ่มประสิทธิภาพให้แก่กระบวนการทำงานภายในระบบมากยิ่งขึ้น

สำหรับระบบงานนี้ในขั้นตอนของการวิเคราะห์และออกแบบระบบเชิงวัตถุ จะเลือกใช้ Rational Rose เป็น Case Tool, ใช้ใช้เทคโนโลยี JSP มาใช้ในการพัฒนา เป็นภาษาในการพัฒนาโปรแกรม และใช้ Microsoft SQL Server 2000 เป็นฐานข้อมูลของระบบงาน

## 1.1 เป้าหมายของโครงการ

- 1.1.1 เพื่อเป็นแนวทางการศึกษา การนำระบบ Windows Web Solution มาใช้งาน (ว่าต้องใช้อะไรบ้าง)
- 1.1.2 เพื่อฝึกฝนการสร้างซอฟต์แวร์เว็บแอปพลิเคชัน โดยการใช้ UML (Unified Modeling Language) ซึ่งเป็นภาษาสัญลักษณ์รูปภาพ ในการวิเคราะห์และออกแบบระบบงาน เพื่อให้ทำงานของเว็บแอปพลิเคชันมีประสิทธิภาพสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.1.3 เพื่อเป็นแนวทางในการออกแบบระบบแบบ N-Tier และนำไปประยุกต์พัฒนาระบบพาณิชย์อิเล็กทรอนิกส์ต่อไป
- 1.1.4 เพื่อเป็นการเสริมสร้างพื้นฐานการเรียนรู้เทคโนโลยีใหม่ๆ เพื่อใช้สำหรับการพัฒนาระบบที่กำลังจะเกิดในอนาคต

## 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อพัฒนาระบบการจัดการข้อมูลสินค้าคงคลังแบบฝากขายให้มีประสิทธิภาพ และดำเนินการทำงานได้รวดเร็วยิ่งขึ้น
- 1.2.2 เพื่อช่วยในการสืบค้นข้อมูล ลดความผิดพลาดในการจัดเก็บข้อมูล
- 1.2.3 เพื่อเป็นแนวทางเบื้องต้นในการพัฒนาระบบพาณิชย์อิเล็กทรอนิกส์สมัยใหม่ ให้มีความน่าเชื่อถือ มีความปลอดภัย และสามารถลด หรือ ขยายระบบให้เหมาะสมกับการเติบโตของธุรกิจที่เป็นอยู่ได้

## 1.3 ขอบเขตของการศึกษา

ระบบการจัดการข้อมูลสินค้าคงคลังแบบฝากขาย เป็นการพัฒนาโดยใช้ UML (Unified Modeling Language) ซึ่งเป็นภาษาลัญลักษณ์รูปภาพเข้ามาช่วยในการวิเคราะห์และออกแบบระบบ ซึ่งการทำงานของระบบจะแบ่งส่วนของการทำงานเป็นดังนี้

- 1.3.1 เป็นการเก็บรวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกับระบบการจัดการข้อมูลสินค้าคงคลังแบบฝากขาย โดยใช้เอกสารของระบบเดิมที่มีอยู่หรือการสัมภาษณ์ผู้เกี่ยวข้องกับระบบงาน
- 1.3.2 วิเคราะห์และออกแบบระบบงานใหม่ โดยใช้ข้อมูลในข้อ 1.4.1 เป็นพื้นฐาน ทั้งนี้ระบบงานใหม่จะต้องสามารถรองรับการทำงานของระบบงานเดิมได้

## 1.4 ขั้นตอนการดำเนินงาน

ในการพัฒนาระบบการจัดการข้อมูลสินค้าคงคลังแบบฝากขาย ให้มีประสิทธิภาพที่ดี จำเป็นที่จะต้องศึกษาหลักการและทฤษฎีต่าง ๆ ดังนี้

- 1.4.1 ศึกษาหลักการวิเคราะห์และออกแบบระบบงานโดยการใช้ UML (Unified Modeling Language) เพื่อให้รู้ถึงหลักการงาน คุณสมบัติ และมาตรฐานโดยทั่วไป
- 1.4.2 นำเอาสิ่งที่ได้ศึกษาจาก ข้อ 1.5.1 มาเข้าสู่กระบวนการวิเคราะห์และออกแบบระบบเชิงวัตถุ โดยนำเสนอแนวความคิดในการวิเคราะห์และออกแบบระบบ ในรูปแบบของ UML (Unified Modeling Language) และใช้ Rational Rose เป็น Case Tool ออกแบบระบบ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนลิขสิทธิ์อื่น ๆ กรุณาอย่าให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.4.3 ศึกษาวิธีการพัฒนาโปรแกรมโดยใช้เทคโนโลยี JSP (Java Server Pages) และ Servlet
- 1.4.4 ศึกษาวิธีการทำงานของระบบฐานข้อมูล Microsoft SQL Server 2000 DBMS
- 1.4.5 ศึกษาวิธีการทำงานของ Apache Web Server โดยใช้ Tomcat Jakarta 4.0

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการพัฒนาระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ จำแนกออกเป็น 3 หัวข้อหลัก ๆ ดังนี้

#### 1.5.1 ประโยชน์ต่อผู้ทำการพัฒนาระบบ

- เพื่อเพิ่มความรู้ ความเข้าใจ และประสบการณ์ในการวิเคราะห์และออกแบบระบบงานโดยใช้ UML (Unified Modeling Language)
- เพื่อประโยชน์ต่อการทำงานในอนาคต

#### 1.5.2 ประโยชน์ต่อองค์กร

- ปรับปรุงการทำงานจากเดิมที่เป็น Manual โดยการนำเอาคอมพิวเตอร์เข้ามาใช้เพื่อเพิ่มประสิทธิภาพในการทำงาน
- ลดความผิดพลาด บกพร่องในการทำงาน
- สร้างความน่าเชื่อถือของข้อมูลต่อผู้บริหาร
- เป็นแนวทางในการปรับปรุงขั้นตอนการทำงานในระบบเดิมให้มีประสิทธิภาพมากยิ่งขึ้น

#### 1.5.3 ประโยชน์ทั่วไป

- สามารถนำไปศึกษาเพื่อเป็นแนวทางในการพัฒนาระบบงานอื่น ๆ ได้ต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### แนวความคิดและทฤษฎีที่เกี่ยวข้อง

#### 2.1 ภาษาแบบจำลองสำหรับการพัฒนาโปรแกรมเชิงวัตถุ (Unified Modeling Language)

การที่จะนำวิธีการพัฒนาโปรแกรมเชิงวัตถุมาใช้งานได้อย่างมีประสิทธิภาพนั้น (Object-Oriented Programming) จะต้องมีกระบวนการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis and Design : OOAD) เพื่อที่จะสามารถพัฒนาโปรแกรมได้อย่างมีประสิทธิภาพ นอกจากนี้ยังสามารถเอื้ออำนวยให้มีการพัฒนาโปรแกรมได้อย่างรวดเร็ว และง่ายต่อการบำรุงรักษา จึงจำเป็นต้องกำหนดภาษาในการสร้างแบบจำลองเชิงวัตถุ

เนื่องจากแบบจำลองอื่น เช่น Data Flow Diagram, Context Diagram เป็นแบบจำลองที่นำเสนอในรูปแบบของการส่งผ่านการทำงานภายในระบบ ซึ่งเป็นแบบจำลองที่เหมาะสมในการนำเสนอแนวความคิดการออกแบบระบบเชิงโครงสร้างมากกว่าระบบเชิงวัตถุ ประกอบกับแนวความคิดการวิเคราะห์และออกแบบระบบเชิงวัตถุเป็นที่นิยมเพิ่มขึ้นอย่างรวดเร็ว จึงทำให้ภาษาแบบจำลองเชิงวัตถุมีมากมายแตกต่างกันไป และผู้ใช้งานจำนวนมากที่ใช้วิธีดังกล่าวก็ยังไม่สามารถพบวิธีที่ตนเองพอใจได้ ดังนั้นจึงต้องมีการคิดค้นภาษารูปแบบจำลองขึ้นมาเป็นมาตรฐาน ได้แก่ แบบจำลอง Unified Modeling Language (UML)

##### 2.1.1 ความเป็นมาของ UML (UML ภาษามาตรฐานเพื่อผู้พัฒนาซอฟต์แวร์.2544 :33)

UML เป็นภาษาภาพเพื่อเป็นสื่อแสดงถึง ความคิด ระบุรายละเอียด และวางโครงสร้างในการออกแบบระบบเชิงวัตถุ เริ่มมีการคิดค้นเมื่อปลายศตวรรษที่ 1980 โดย Grandy Booch, James Rumbaugh และ Ivar jacobson UML 1.0 เป็นเวอร์ชันที่สามารถใช้ได้และมีเสถียรภาพ ได้รับการรับรองจาก Object Management Group (OMG) เมื่อเดือนมกราคม 1997 ว่าเป็นภาษาสากลที่เปรียบเสมือนภาพพิมพ์เขียวของโปรแกรม หลาย ๆ องค์กรได้มีการสนับสนุนมาตรฐานของ UML ตั้งแต่พื้นฐานรูปแบบทางภาษา (Modeling Language) ซึ่งเป็นวิธีการเชิงวัตถุ ในปัจจุบันเวอร์ชันล่าสุดของ UML คือ UML 1.3 ซึ่งได้มีการพัฒนา UML เพื่อแก้ไขข้อผิดพลาด และเพิ่มเติมในส่วนที่ขาดหายไปให้ตรงกับความต้องการของผู้ใช้งานมากที่สุด

สิ่งที่ทำให้ UML ประสบความสำเร็จ คือ

- UML เป็นรูปแบบที่แสดงให้เห็นถึงการวิเคราะห์และออกแบบระบบเชิงวัตถุอย่างจน และเป็นขั้นตอน
- UML เป็นการรวมรูปแบบที่ใช้ได้ในหลาย ๆ ระบบ (Business Versus Software), ขั้นตอนการพัฒนาระบบ (Development Phase)

## 2.1.2 แบบจำลองต่าง ๆ ใน UML

มีแบบจำลอง (Diagrams) ต่าง ๆ ดังนี้

1. Use Case Diagram
2. Class Diagram
3. Object Diagram
4. State Diagram
5. Sequence Diagram
6. Collaboration Diagram
7. Activity Diagram
8. Component Diagram
9. Deployment Diagram

เราสามารถแบ่งแบบจำลอง ใน UML ออกเป็น Model ต่าง ๆ ได้ดังนี้

### 1. Functional Model

เป็น Model ที่ใช้แสดงความต้องการของระบบทั้งหมด ช่วยในการอธิบายรายละเอียดหลัก ๆ ภายใน Object แสดงถึงการส่งผ่านข้อมูลในการทำงาน โดยจะสนใจเพียงแค่ว่าต้องทำงานอะไรบ้าง แต่ยังไม่ต้องสนใจว่าจะทำงานนั้นอย่างไร โดยแบบจำลองที่แนะนำเสนอความต้องการของระบบทั้งหมดคือ Use Case Diagram โดยจะถูกนำไปในการวิเคราะห์และออกแบบระบบใน Phase ต่อ ๆ ไป

### 2. Object Model

เป็น Model ที่ใช้แสดงโครงสร้างของระบบ ดังนั้นจึงเป็น Model ที่คงที่ จึงสามารถเรียกได้อีกอย่างหนึ่งว่า Static Model ซึ่งพิจารณาจากความต้องการของระบบจาก Functional Model นำเสนอในรูปของ Class ต่าง ๆ โดยจะนำเสนอด้วยแบบจำลอง Class Diagram และ Component Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. Dynamic Model

เป็น Model ที่ใช้แสดงถึงการทำงานระหว่างวัตถุ (Object) ต่าง ๆ ตามการส่งข้อความ (Message) หรือเมื่อมีเหตุการณ์ (Event) ต่าง ๆ เกิดขึ้น โดย Object ในที่นี้หมายถึงตัวอย่าง (Instance) ของ Class ที่ได้ออกแบบไว้ใน Class Model ดังนั้น Object จะมีคุณลักษณะ (Attribute) และพฤติกรรม (Method) เช่นเดียวกับ Class ที่เป็นต้นแบบของ Object นั้น

การทำงานของระบบเกิดจากการส่ง Message ระหว่าง Object เหล่านั้น เมื่อมีการทำงาน Object จะเปลี่ยนสถานะ (State) ไปตามเหตุการณ์ที่เกิดขึ้นได้ เพื่อให้เป็นไปตามที่กำหนดไว้ใน Functional Model ซึ่งแบบจำลองที่จะใช้นำเสนอคือ Sequence Diagram และ State Diagram

#### 2.1.3 ความหมายของแบบจำลองต่าง ๆ ใน UML

##### 2.1.3.1 แบบจำลอง Use Case (Use Case Diagram)

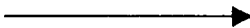
เป็นแบบจำลองที่แสดงถึงความสัมพันธ์ระหว่างผู้ใช้กับระบบต่าง ๆ โดยนำเสนอถึงความต้องการของระบบทั้งหมดในรูปแบบที่เข้าใจได้ง่าย

1) Actor ใช้สัญลักษณ์รูปคน แสดงถึงบทบาท (Roles) ที่ผู้ใช้จะเกี่ยวข้องกับระบบ ซึ่งผู้ใช้แต่ละคนอาจจะมีได้หลายบทบาท แต่ละบทบาทจะมีความสัมพันธ์กับกิจกรรม (Use Case) ได้เพียงหนึ่งกิจกรรมเท่านั้น

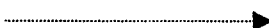
2) Use Case ใช้สัญลักษณ์ วงรี แสดงถึงแต่ละกิจกรรมที่ทำในระบบ ซึ่งจะเกิดจากการแลกเปลี่ยนข้อความ (Message) ระหว่างผู้ใช้กับระบบ หรือระหว่างระบบด้วยกันเอง

3) ความสัมพันธ์ระหว่าง Actor กับ Use Case หรือ ความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง ใช้สัญลักษณ์ ลูกศร ในการเชื่อมโยงโดยหัวลูกศรจะชี้ตามทิศทางของการส่งข้อความ (Message) โดยมีลักษณะความสัมพันธ์ คือ

- Association คือการเชื่อมโยงความสัมพันธ์ระหว่าง Actor กับ Use Case ซึ่งแสดงถึงการติดต่อสื่อสารกัน อาจจะเป็นทางเดียวหรือสองทางก็ได้ โดยหัวลูกศรแสดงถึงทิศทางของการส่งข้อความ

สัญลักษณ์ที่ใช้คือ 

- Extend คือการเชื่อมโยงความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง โดยความสัมพันธ์จะเป็นแบบการขยายหรือเพิ่มการทำงานให้กับ Use Case หัวลูกศรจะชี้ไปที่ Use Case ที่ถูกเรียกใช้งานไปยัง Use Case ที่ต้องการเพิ่มเติมการทำงาน และมีข้อความ <<extend>> ระบุอยู่เหนือเส้น

สัญลักษณ์ที่ใช้คือ 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Generalization คือความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง โดยเป็นความสัมพันธ์ในลักษณะการกำหนดขอบเขต เช่น Use Case A เป็นตัวกำหนดขอบเขต Use Case B หัวลูกศรจะชี้จาก Use Case A ไปที่ Use Case B

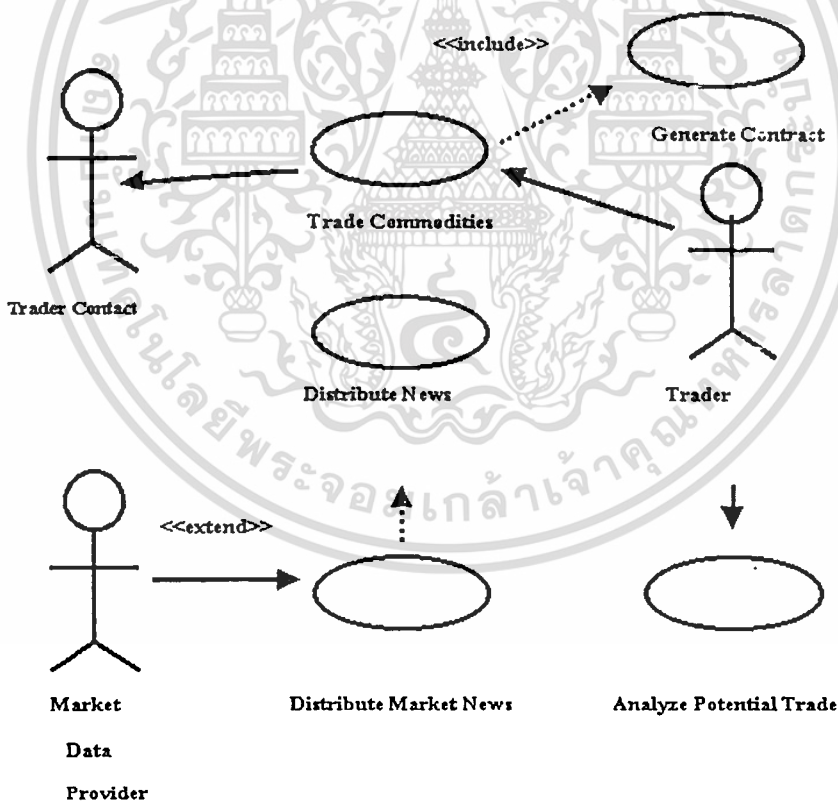
สัญลักษณ์ที่ใช้คือ 

- Include คือความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง โดยจะเป็นการถ่ายทอด คุณสมบัติ (Inherit) จาก Use Case หนึ่ง ไปยังอีก Use Case หนึ่ง ซึ่งหมายถึงความสัมพันธ์แบบ Uses ใน UML เวอร์ชัน 1.1 โดยหัวลูกศรจะชี้ไปยัง Use Case หลัก ที่จะถูกถ่ายทอดคุณสมบัติออกไป และมีข้อความ <<include>> อยู่เหนือเส้น

สัญลักษณ์ที่ใช้คือ 

4) ความสัมพันธ์ระหว่าง Actor ด้วยกันเอง มีลักษณะความสัมพันธ์คือ

- Generalization คือ ความสัมพันธ์ระหว่าง Actor ด้วยกันเอง โดยเป็นความสัมพันธ์ในลักษณะ Actor หนึ่ง สามารถสื่อสารกับ Use Case ได้ในลักษณะเดียวกันกับอีก Actor หนึ่ง



รูปที่ 2.1 ตัวอย่างแบบจำลอง Use Case (Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.2 แบบจำลอง Class (Class Diagram)

จัดว่าเป็นแบบจำลองที่สำคัญที่สุดในการออกแบบระบบเชิงวัตถุ แสดงให้เห็นถึง

- 1) คุณลักษณะ (Attribute) คือ ข้อมูล (Data) หรือ ตัวแปร (variable)
- 2) หน้าที่การทำงาน (Operation) คือ วิธีการ (Method) หรือ พฤติกรรม (Behavior) ภายใน Class และ
- 3) ความสัมพันธ์ของ Class ต่าง ๆ ภายในระบบ

สัญลักษณ์ของ Class ที่ใช้ คือ รูปสี่เหลี่ยมซึ่งอธิบายถึงกลุ่ม Object ต่าง ๆ โดยมีรายละเอียดพื้นฐาน 3 ส่วน คือ

Name Compartment : แสดงชื่อ Class ภายในระบบ


Attribute : แสดงถึงโครงสร้างและคุณลักษณะทั้งหมดที่มีภายใน Class ซึ่งภายใน Class เดียวกันจะไม่สามารถมีคุณลักษณะที่ซ้ำกันได้ แต่ถ้าต่าง Class กันจะสามารถมีคุณลักษณะที่ซ้ำกับคุณลักษณะใน Class อื่นได้


Operation : เป็นส่วนที่ใช้อธิบายถึงวิธีการ (Method) หรือพฤติกรรม (Behavior) ภายใน Class รวมทั้งการรับค่าตัวแปร (Argument) หรือส่งค่าตัวแปรออกไป การเชื่อมโยงความสัมพันธ์ภายใน Class จะนำเสนอโดย

- 1) บทบาทของความสัมพันธ์ (Role Names) แสดงถึงบทบาทที่ Class หนึ่ง สัมพันธ์กับอีก Class หนึ่ง ตัวอย่างเช่น จากรูปที่ 2 Class Trade Administrator มีความสัมพันธ์กับ Class Trade โดย มีบทบาทคือ เป็น Manager

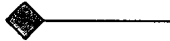
- 2) รูปแบบของความสัมพันธ์ แบ่งออกเป็น 4 แบบ คือ

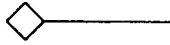
- Association : เป็นการแสดงความสัมพันธ์ระหว่าง Class โดยอาจจะเป็นทางเดียว หรือ สองทาง สัญลักษณ์ที่ใช้ มีดังนี้

ความสัมพันธ์แบบทางเดียว 

ความสัมพันธ์แบบสองทาง 

- Aggregation : เป็นรูปแบบพิเศษของ Association คือเป็นความสัมพันธ์ที่แสดงว่า Class หนึ่ง เป็นส่วนหนึ่งของอีก Class หนึ่ง เช่นจากรูปที่ 2 Class Administrator Position (ตำแหน่งผู้จัดการฝ่ายขาย) เป็นส่วนหนึ่งของ Class Slate (ผู้ที่ได้รับการเสนอชื่อ) สัญลักษณ์ที่ใช้ มีดังนี้

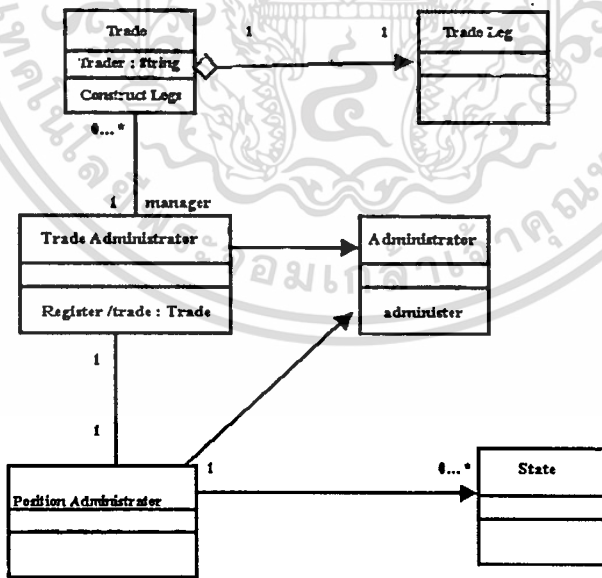
ความสัมพันธ์แบบ By Value 

ความสัมพันธ์แบบ By Reference 

- Depends on : เป็นรูปแบบความสัมพันธ์แบบหนึ่งที่ใช้แสดงความสัมพันธ์ระหว่าง 2 Class ในแง่ที่ Class หนึ่งเรียกใช้บริการของอีก Class หนึ่ง กล่าวคือ Class ของผู้ขอบริการ ขึ้นอยู่กับบริหารของ Class ของผู้ให้บริการ แต่ไม่มีการขึ้นต่อกันภายในโครงสร้างของ Class  
สัญลักษณ์ที่ใช้คือ  $\longrightarrow$

- Generalization : เป็นความสัมพันธ์ระหว่าง Class กับ Class ในแง่ของการถ่ายทอดคุณสมบัติ (Inherit) จาก Class หนึ่งไปยังอีก Class หนึ่ง โดย Class ที่เป็นต้นแบบเรียกว่า Super Class และ Class ที่ได้รับการถ่ายทอด เรียกว่า Sub Class เช่น จากรูปที่ 2 Class Trade Administrator และ Class Position Administrator เป็น Class ที่ได้รับการถ่ายทอดคุณสมบัติจาก Class Administrator โดยหัวลูกศรชี้ไปที่ Class ต้นแบบ  
สัญลักษณ์ที่ใช้คือ  $\triangle$

3) จำนวนความสัมพันธ์ (Multiplicity) แต่ละ Class จะมีความสัมพันธ์ระหว่างกันโดยสามารถกำหนดจำนวนความสัมพันธ์เป็นตัวเลข หรือช่วงของตัวเลข ในรูปแบบค่าต่ำสุด...ค่าสูงสุดไว้ที่ปลายของเส้นแสดงความสัมพันธ์ ถ้าเป็นเลขจำนวนเดียว หมายถึงค่าที่แน่นอน ถ้าเป็นช่วง หมายถึงค่าที่เป็นไปได้ ถ้าหมายถึงจำนวนใดก็ได้จะใช้เครื่องหมายดอกจัน (\*) เช่น จากรูปที่ 2 Class Trade มีความสัมพันธ์กับ Class Trade Administrator ซึ่งมีบทบาทเป็น Manager โดยที่ 1 Trade Administrator จะทำหน้าที่เป็น Manager แก่ Class Trade เป็นจำนวน 0 หรือมากกว่า

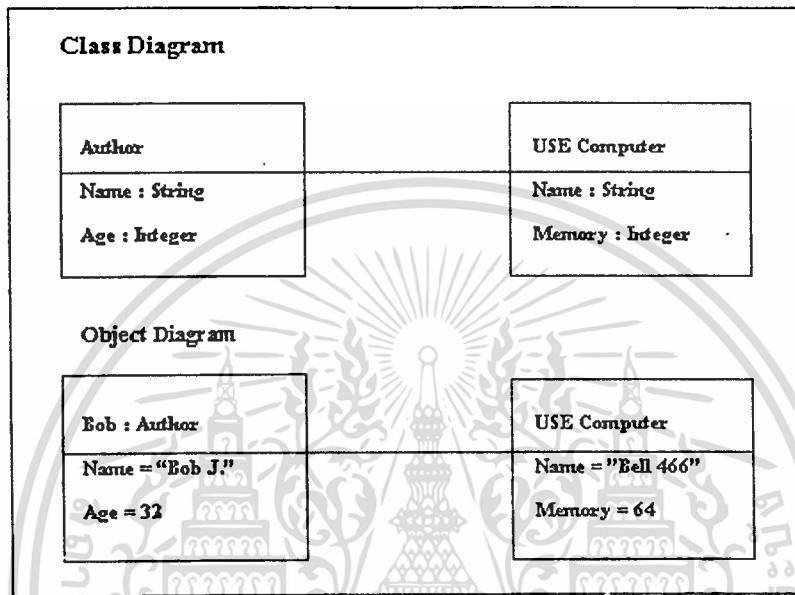


รูปที่ 2.2 ตัวอย่างแบบจำลอง Class  
(Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.3 แบบจำลอง Object (Object Diagram)

เป็นแบบจำลองที่ขึ้นอยู่กับ Class Diagram ที่มีโครงสร้างมาก่อนแล้ว โดย Object Diagram จะนำเสนอในเชิงการบันทึกข้อมูลตามโครงสร้างของ Class Diagram ทำให้เราสามารถมองเห็นภาพรวมอย่างกว้าง ๆ ของ Class Diagram ได้

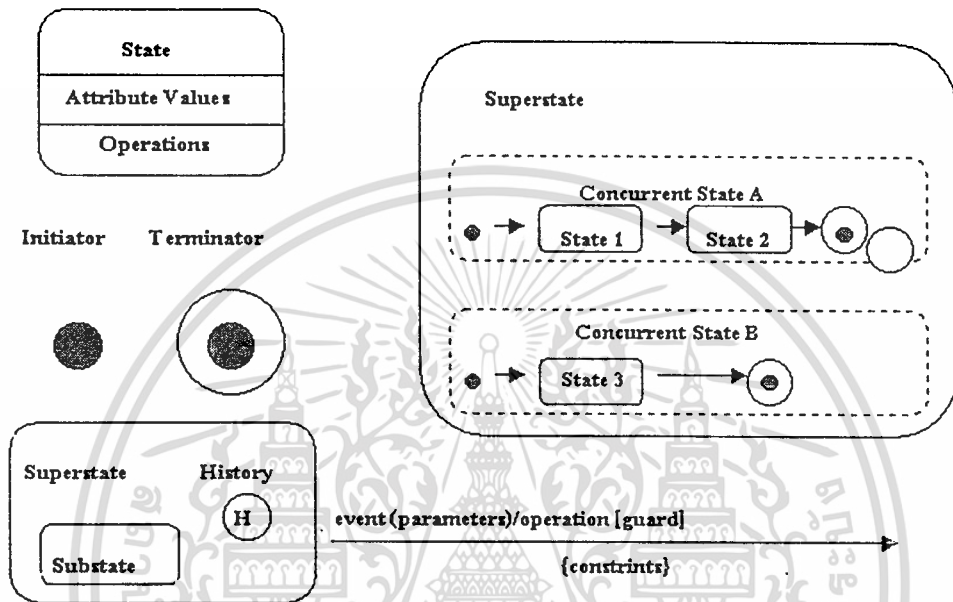


รูปที่ 2.3 ตัวอย่างแบบจำลอง Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.4 แบบจำลอง State (State Diagram)

เป็นแบบจำลองที่อธิบายถึงวงจรชีวิต (Life Cycle) ของ Object ซึ่งจะเป็นตัวบอกว่าเกิดอะไรขึ้นบ้าง State diagram จะเป็นตัวเชื่อมทุก Class ที่มีพฤติกรรมซับซ้อนให้ชัดเจนขึ้นโดยมีสัญลักษณ์ต่าง ๆ ดังนี้

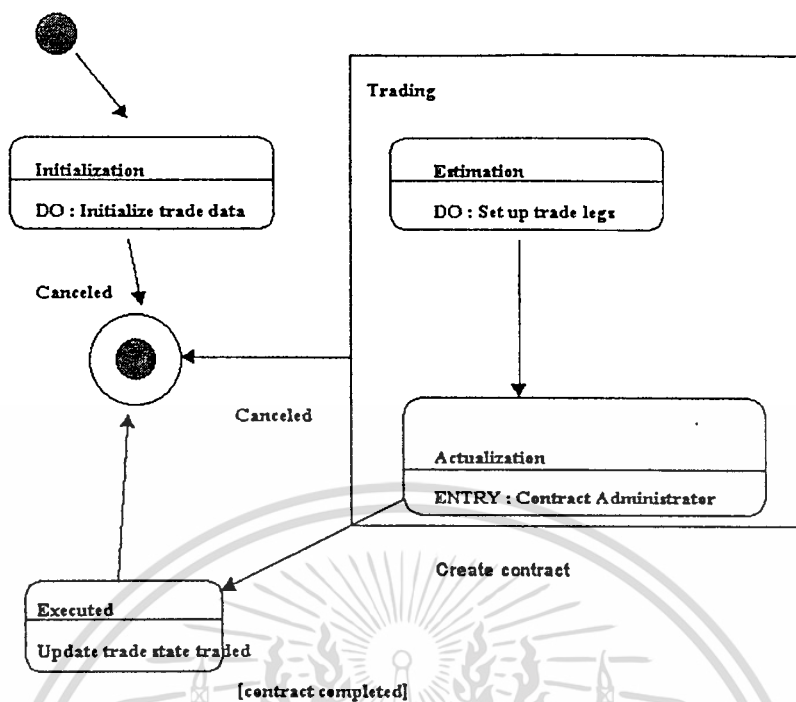


รูปที่ 2.4 สัญลักษณ์แบบจำลอง State  
(Rational Software Corporation, 1999)

State Diagram จะมีส่วนประกอบอยู่ 3 ส่วน

- State's Name : ชื่อของ State
- Optional State : เป็นคุณสมบัติของคลาสใน State Diagram
- Optional Activity State จะเป็นรายละเอียดของเหตุการณ์และกิจกรรม (Action) ต่าง ๆ State ที่ใช้เป็นมาตรฐาน ในกิจกรรมต่าง ๆ มี 3 สถานะคือ
- Entry คือการเข้าไปทำกิจกรรมต่าง ๆ ภายใน State
- Exit จะใช้ในกรณีที่ต้องการออกจาก State
- Do ใช้ในการปฏิบัติงานต่าง ๆ ใน State เช่น การส่งข้อความ (message), การรอหรือการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

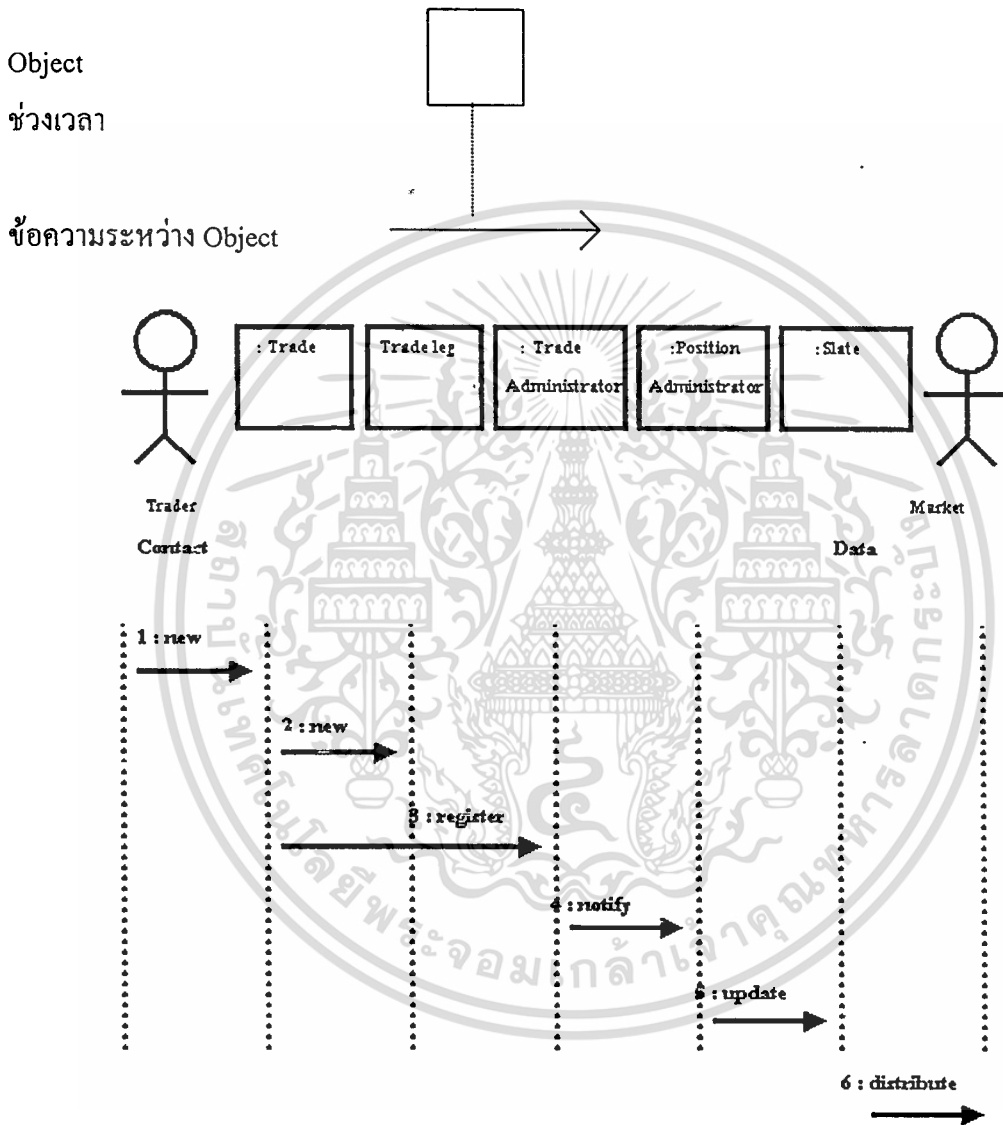


รูปที่ 2.5 ตัวอย่างแบบจำลอง State  
(Rational Software Corporation, 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.5 แบบจำลอง Sequence (Sequence Diagram)

เป็นแบบจำลองที่แสดงให้เห็นถึงการทำงานระหว่าง Object ต่าง ๆ ตามการส่งข้อความและเมื่อเหตุการณ์ต่าง ๆ ได้เกิดขึ้น ซึ่งผู้เขียนโปรแกรมจะใช้ Diagram ตัวนี้เป็นตัวช่วยเพื่อที่จะได้เขียนโปรแกรมให้ได้ตามที่ได้ออกแบบไว้ สัญลักษณ์ที่ใช้คือ



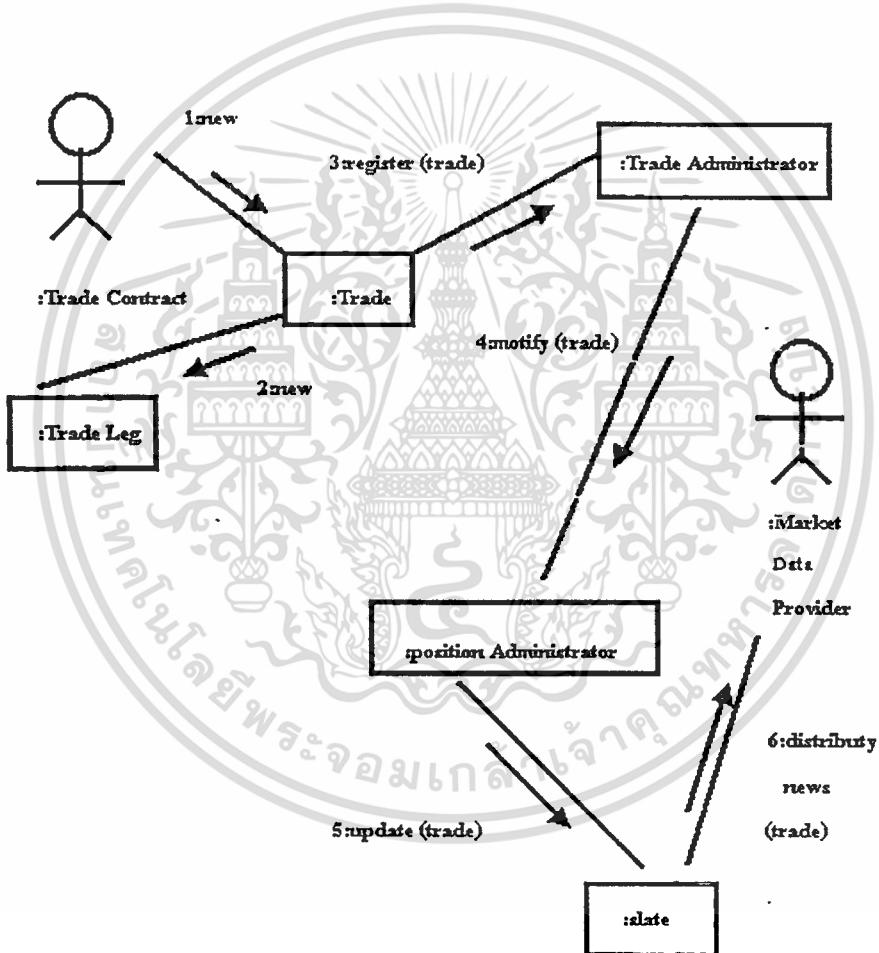
รูปที่ 2.6 ตัวอย่างแบบจำลอง Sequence (Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.6 แบบจำลอง Collaboration (Collaboration Diagram)

เป็นแบบจำลองแสดงการติดต่อสื่อสารระหว่าง Object ต่าง ๆ ที่เกี่ยวข้องกับแต่ละ Object ที่สนใจ รวมทั้งความสัมพันธ์ระหว่างแต่ละ Object ในการติดต่อสื่อสารด้วย สัญลักษณ์ที่ใช้จะเหมือนกับ Sequence Diagram ซึ่งสามารถเลือกแสดงโดยแบบจำลองอันใดอันหนึ่งได้ กล่าวคือ ทั้งสองแบบสามารถปรับเปลี่ยนรูปแบบระหว่างกันได้ หลักเกณฑ์เลือกจะใช้แบบจำลองใดคือ

- 1) ถ้าเป็นการกำหนดช่วงเวลาที่แน่นอนและใช้เวลาเป็นสิ่งสำคัญควรเลือกใช้ Sequence Diagram
- 2) ถ้าให้ความสำคัญกับ Context ควรเลือกใช้ Collaboration Diagram



รูปที่ 2.7 แบบจำลอง Collaboration (Collaboration Diagram)  
(Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.7 แบบจำลอง Activity (Activity Diagram)

เป็นแบบจำลองที่อธิบายถึงกิจกรรม (Action) และผลลัพธ์ของกิจกรรมนั้น เพื่อแสดงถึงความเปลี่ยนแปลงที่เกิดจากการประมวลผลภายใน ซึ่งสถานะการทำงานนั้นเมื่อเริ่มต้นทำก็ต้องทำให้เสร็จก่อนที่จะเริ่มทำงานในสถานะต่อไปได้ ดังนั้นสิ่งที่จะใช้ในการควบคุมการทำงานในขั้นตอนต่าง ๆ ก็คือการติดต่อสื่อสารกันหรือการเชื่อมโยงกันของ State ต่าง ๆ ซึ่งสามารถกระทำได้ทั้งการส่งและรับข้อมูลข่าวสารได้พร้อม ๆ กัน โดยนำมาใช้ประโยชน์ในการเชื่อมต่อกับ flow และรายละเอียดของพฤติกรรมที่มีการประมวลผลแบบขนาน (Parallel) คือสามารถทำงานได้หลายอย่างในเวลาเดียวกันได้เป็นจำนวนมาก ลักษณะของ Activity Diagram จะคล้ายกับ State Diagram แต่ Activity Diagram จะใช้แสดงเหตุการณ์ต่าง ๆ อันเป็นผลมาจากการกระทำกิจกรรมภายในระบบเอง ในขณะที่ State Diagram จะใช้ในการแสดงการกระทำที่เกิดขึ้นในลักษณะที่ไม่ต่อเนื่องกัน

สัญลักษณ์ที่ใช้ใน Activity Diagram

- กิจกรรม (Action)



- การตัดสินใจ (Decision)

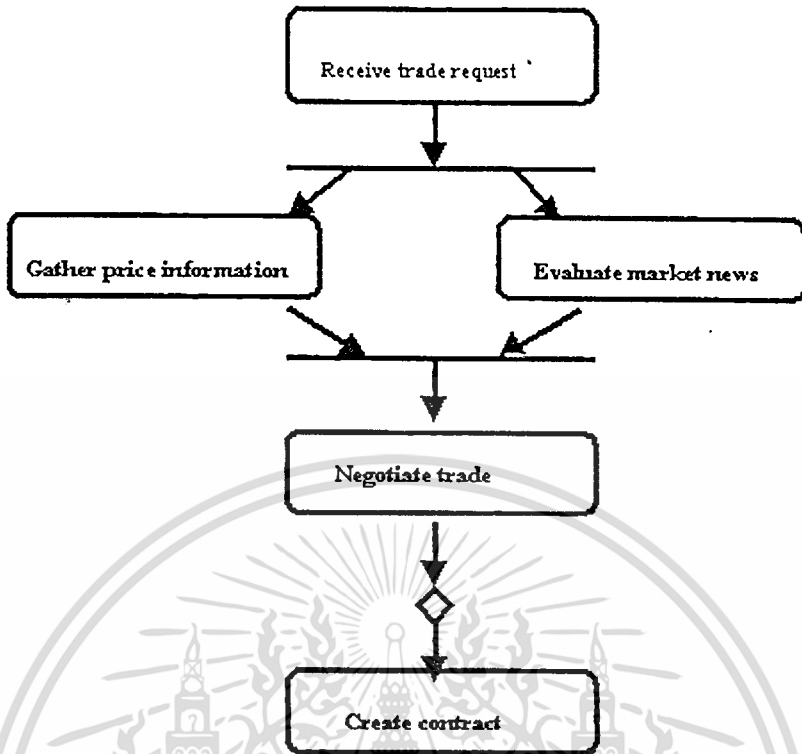


- ลู่วางเขตความรับผิดชอบ (Swimlane)

- จุดแยกให้กิจกรรมก่อนหน้าเสร็จก่อน (Sync States)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ตัวอย่างแบบจำลอง Activity  
(Rational Software Corporation. 1999)

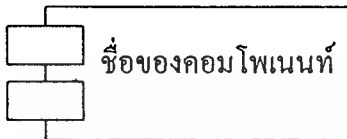
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.8 แบบจำลอง Component (Component Diagram)

เป็นแบบจำลองที่แสดงถึงโครงสร้างภายนอกและความสัมพันธ์ระหว่างองค์ประกอบต่างๆ ของโปรแกรม File ที่มีในระบบในสภาวะแวดล้อมของการพัฒนาระบบ

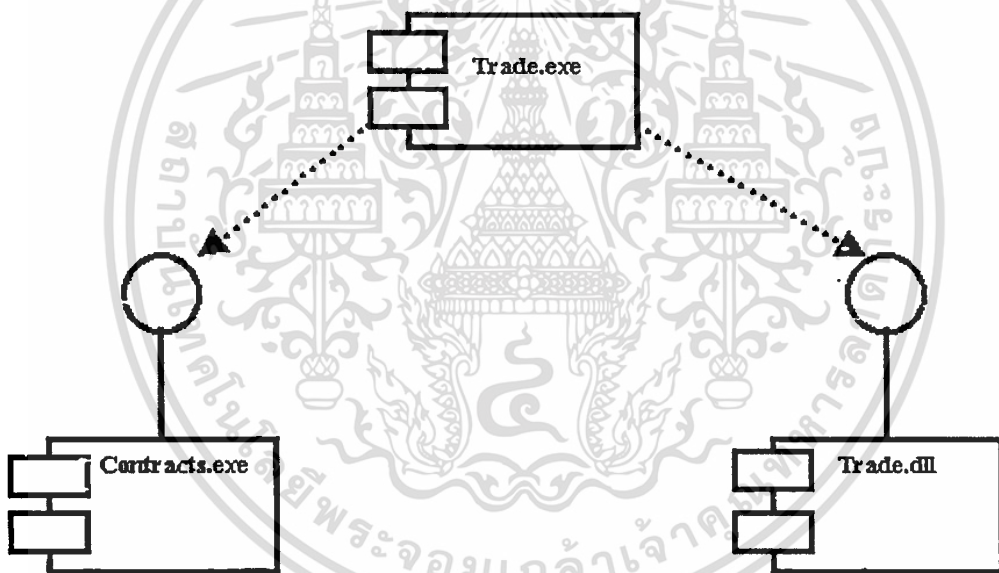
สัญลักษณ์ที่ใช้ใน Component Diagram

- คอมโพเนนต์ต่าง ๆ ของซอฟต์แวร์



- อินเตอร์เฟซ (Interface)

ชื่ออินเตอร์เฟซ

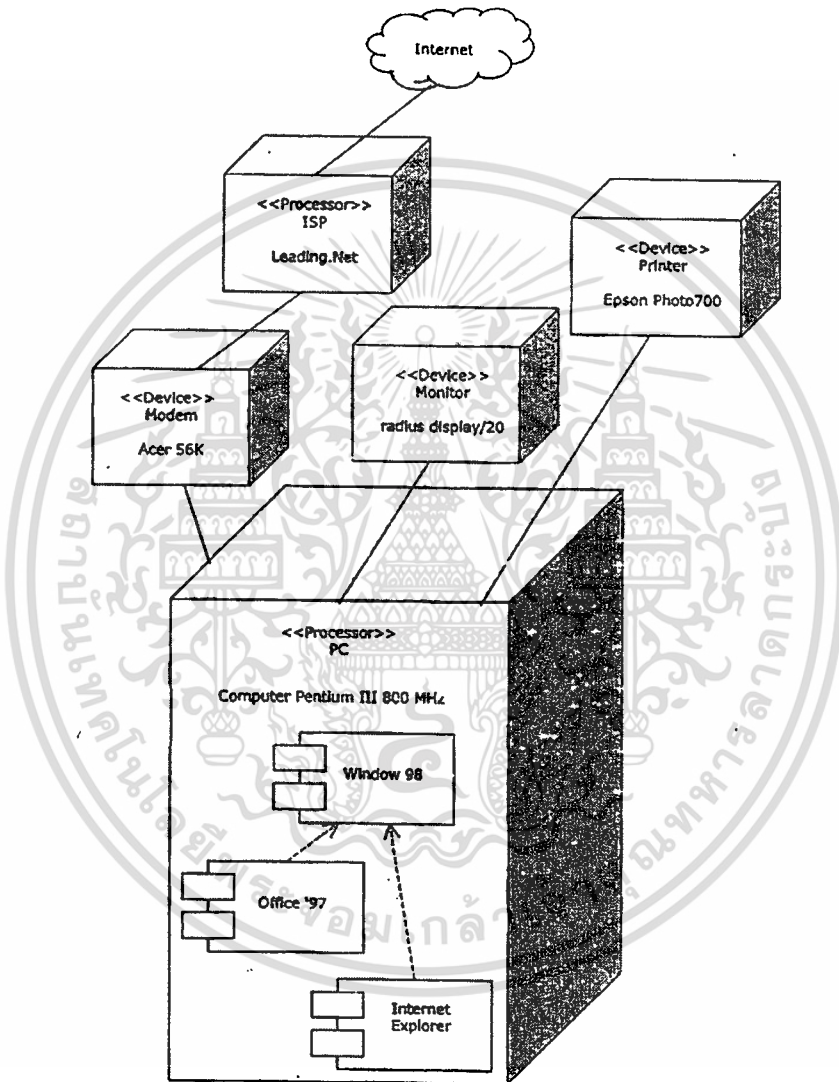


รูปที่ 2.9 ตัวอย่างแบบจำลอง Component  
(Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.9 แบบจำลอง Deployment (Deployment Diagram)

เป็นแบบจำลองที่แสดงถึงส่วนประกอบต่างหรือสถาปัตยกรรมของระบบขณะโปรแกรมทำงาน รวมถึงโปรโตคอล (Protocol) ในการเชื่อมต่อระหว่างโนด (Node) โดยที่ Node แสดงถึงหน่วยประมวลผล อันได้แก่อุปกรณ์คำนวณต่าง ๆ หรืออุปกรณ์ต่าง ๆ ที่ใช้ในการประมวลผลด้วย



รูปที่ 2.10 ตัวอย่างแบบจำลอง Deployment  
(Rational Software Corporation. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 การวิเคราะห์และออกแบบระบบเชิงวัตถุ

จุดประสงค์ที่สำคัญในการวิเคราะห์และออกแบบระบบเชิงวัตถุ คือ การที่จะมองระบบเป็นวัตถุให้ใกล้เคียงกับวัตถุในโลกแห่งความเป็นจริง เพื่อที่จะนำมาแปลงเป็นแบบจำลองของระบบซึ่งจะเป็นแนวทางในการพัฒนาโปรแกรมต่อไป โดยแบบจำลองที่ได้จะเสมือนการสื่อสารระหว่างสิ่งที่ ผู้วิเคราะห์ระบบ (System Analyst) มีความเข้าใจระบบตรงกับผู้ใช้งานระบบ (User) หรือไม่ว่าอย่างไร และปรับปรุงให้มีความเข้าใจตรงกัน จากนั้นแบบจำลองนี้จะนำเสนอแนวความคิดของผู้ออกแบบระบบ เพื่อให้ผู้พัฒนาระบบ (Programmer) สามารถดำเนินการพัฒนาระบบได้ต่อไป

การวิเคราะห์ระบบเชิงโครงสร้าง (Structure Analysis and Design) จะนำเสนอโดย Entity Relationship Diagram (ERDs) และ Data Flow Diagram (DFDs) ซึ่งในส่วนของ DFDs จะเป็นแบบจำลองที่นำเสนอถึงกระบวนการส่งผ่านข้อมูลภายในระบบ และ ERDs จะเป็นแบบจำลองที่แสดงถึงโครงสร้างของข้อมูลหรือสิ่งต่าง ๆ ภายในระบบ โดยที่ DFDs กับ ERDs จะเชื่อมโยงกันโดยเพิ่มข้อมูล (Data Store) จาก DFDs โดยที่หลาย ๆ กระบวนการใน DFDs อาจเกี่ยวข้องกับ Data Store เดียวกัน ซึ่งก่อให้เกิดความซ้ำซ้อน นอกจากนี้การวิเคราะห์และออกแบบระบบเชิงโครงสร้างไม่สามารถนำเสนอถึงคุณสมบัติเชิงวัตถุ เช่น การนำกลับมาใช้ใหม่ เป็นต้น

จากข้อจำกัดของการวิเคราะห์และออกแบบระบบเชิงโครงสร้างที่ได้กล่าวมาแล้วข้างต้น จะเห็นว่า การวิเคราะห์และออกแบบระบบเชิงวัตถุจึงเป็นวิธีการที่เหมาะสมกว่าในการวิเคราะห์และออกแบบระบบเพื่อพัฒนาระบบเชิงวัตถุ กล่าวคือ การวิเคราะห์และออกแบบระบบเชิงวัตถุจะนำเสนอในรูปแบบที่เสมือนโลกแห่งความเป็นจริงมากกว่าการวิเคราะห์ระบบเชิงโครงสร้าง ดังนั้น User จึงสามารถทำความเข้าใจได้ง่าย นอกจากนี้การวิเคราะห์และออกแบบระบบเชิงวัตถุนี้ยังสนับสนุนหลักการทำงานและคุณสมบัติเชิงวัตถุ จึงทำให้แบบจำลองที่ได้จากการวิเคราะห์และออกแบบระบบสามารถแสดงถึงแนวความคิดเชิงวัตถุได้อย่างชัดเจน ทำให้มองเห็นเค้าโครงของระบบและพัฒนาโปรแกรมเชิงวัตถุได้โดยสะดวกขึ้น

### 2.2.1 ขั้นตอนในการพัฒนาระบบเชิงวัตถุ

ในการพัฒนาระบบเชิงวัตถุ จะแบ่งขั้นตอนในการทำงานเป็น Phase หลัก 6 Phase คือ

- 1) Requirement Analysis
- 2) Analysis
- 3) Design
- 4) Programming
- 5) Testing

## 6) Maintenance

### 2.2.1.1 12 ขั้นตอนการวิเคราะห์ ้ระความต้องการของรบบ (Requirement Analysis)

เป็นขั้นตอนที่พยายามหาความต้องการของผู้ใช้งานและวัตถุประสงค์ของระบบ อันเป็นตัวช่วยในการกำหนดว่าระบบที่จะพัฒนาขึ้นนี้ต้องมีอะไรบ้าง โดยจะอธิบายความต้องการของผู้ใช้งานว่าอะไรคือสิ่งที่ระบบควรจะเป็น เพื่อที่จะเรียบเรียงเป็นความต้องการของระบบ และทำให้สามารถพัฒนาระบบที่บรรลุตามวัตถุประสงค์ที่วางไว้ได้ ดังนั้นจึงควรอธิบายความต้องการของระบบให้ชัดเจน เพื่อที่จะพัฒนาระบบได้ถูกต้อง ตรงตามความต้องการต่อไป

โดยขั้นตอนนี้จะนำเสนอด้วย แบบจำลอง Use Case ซึ่งจะนำเสนอให้เห็นว่าระบบมีความเกี่ยวข้องกับสิ่งใด ๆ หรือวัตถุใด ๆ บ้าง และระบบต้องมีกิจกรรมอะไรบ้าง สิ่งที่ได้จากการวิเคราะห์ความต้องการของระบบ นอกจากจะเป็นการทำความเข้าใจระบบเพื่อนำไปพัฒนาโปรแกรมเท่านั้น แต่ยังเป็นการทำความเข้าใจเกี่ยวกับระบบการทำงานในเชิง Business Process ด้วย

### 2.2.1.2 ขั้นตอนในการวิเคราะห์ระบบเชิงวัตถุ (Analysis)

ขั้นตอนการวิเคราะห์ระบบถือว่าเป็นส่วนที่มีความสำคัญ เนื่องจากเป็นขั้นตอนที่ช่วยในการกำหนดว่าระบบที่จะพัฒนาขึ้นนี้ต้องมีอะไรบ้าง เพื่อที่จะพัฒนาระบบได้ตรงตามความต้องการของผู้ใช้งานระบบ โดยจะนำเอาความต้องการของผู้ใช้งานที่ได้ศึกษา มาทำการวิเคราะห์ว่าเราจะพัฒนาระบบได้อย่างไร

ในขั้นตอนนี้ จะเป็นการระบุถึง Class ของวัตถุ (Object) ที่มีอยู่ในระบบ โดยรวมถึง คุณสมบัติ (Attribute) และพฤติกรรม (Operation) ของ Class รวมทั้งระบุความสัมพันธ์ระหว่าง Class ว่าเป็น One-to-One, One-to-Many หรือ Many-to-One และแต่ละ Class มีความสัมพันธ์แบบใด นอกจากนี้ จะนำเสนอถึงขั้นตอนการทำงานของระบบเพื่อเป็นแนวทางในการพัฒนาระบบต่อไป ซึ่งในขั้นตอนนี้จะเป็นการวิเคราะห์ปัญหา และหาทางเพื่อให้บรรลุตามวัตถุประสงค์ เป็นการกำหนดว่าระบบที่จะพัฒนาขึ้นมา จะต้องต้องมีอะไรบ้าง โดยที่ยังไม่อธิบายถึงเทคนิคในการที่จะนำมาใช้

ขั้นตอนนี้จะนำเสนอด้วยแบบจำลอง UML คือ

- 1) Class Diagram เพื่อนำเสนอถึง Class ต่าง ๆ ภายในระบบ
- 2) Activity Diagram หรือ State Diagram เพื่อนำเสนอถึงขั้นตอน กระบวนการทำงานของระบบ ซึ่งทั้ง 2 Diagram สามารถนำเสนอได้เหมือนกัน แต่ Activity Diagram จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงกระบวนการ ซึ่งเป็นการกระทำที่ต่อเนื่องกันและสามารถแสดงให้เห็นว่าในกระบวนการทำงานนั้นจำเป็นที่จะต้องให้ขั้นตอนใดเสร็จก่อนจึงจะทำขั้นตอนต่อไปได้ โดยใช้เส้นทึบเป็นตัวแบ่ง

### 2.2.1.3 ขั้นตอนในการออกแบบระบบเชิงวัตถุ (Design)

ในขณะทำการวิเคราะห์ระบบจะเป็นการกำหนดว่าระบบที่จะพัฒนาขึ้นนั้นต้องมีอะไรบ้าง การออกแบบระบบจะเป็นการกำหนดว่าจะทำการพัฒนาระบบได้อย่างไร

ในมุมมองเชิงวัตถุ การวิเคราะห์และการออกแบบระบบจะมีความแตกต่างกันน้อยกว่า เนื่องจากความสัมพันธ์ของคุณสมบัติและพฤติกรรมของวัตถุ และแบบจำลองที่ใช้ในขั้นตอนของการวิเคราะห์และออกแบบจะใช้หลักการเดียวกัน ดังนั้นกระบวนการของการออกแบบจึงสามารถกระทำต่อได้จากการวิเคราะห์ระบบได้อย่างราบรื่น

ช่วงของการออกแบบระบบจะเริ่มเมื่อได้ข้อมูลจากการวิเคราะห์ระบบเพียงพอที่จะพัฒนาระบบได้แล้ว จาก Classes ต่าง ๆ ที่ได้จากช่วงของการวิเคราะห์ระบบ อาจจะต้องมีการเพิ่มเติมบาง Classes เพื่อให้ระบบสมบูรณ์ขึ้น และสามารถพัฒนาระบบต่อไปได้ตาม Hardware และ Software ที่ได้พิจารณาเลือกไว้ ในขั้นนี้จะมอง Classes เป็นกลุ่มของวัตถุ (Objects) ซึ่งจะช่วยให้เห็นมุมมองในการสร้างหน้าจอของโปรแกรมเพื่อผู้ใช้งานระบบได้อย่างชัดเจนขึ้น

ในขั้นตอนของการออกแบบระบบนี้ อาจแบ่งเป็นส่วนในการพิจารณาเลือกเทคนิคเพื่อใช้พัฒนาระบบออกเป็น 2 ส่วนคือ การเลือก Hardware และ Software ที่จะนำมาพัฒนาระบบ โดยอาจมีเกณฑ์ในการพิจารณาดังต่อไปนี้

#### 1) การพิจารณาในส่วนของ Hardware

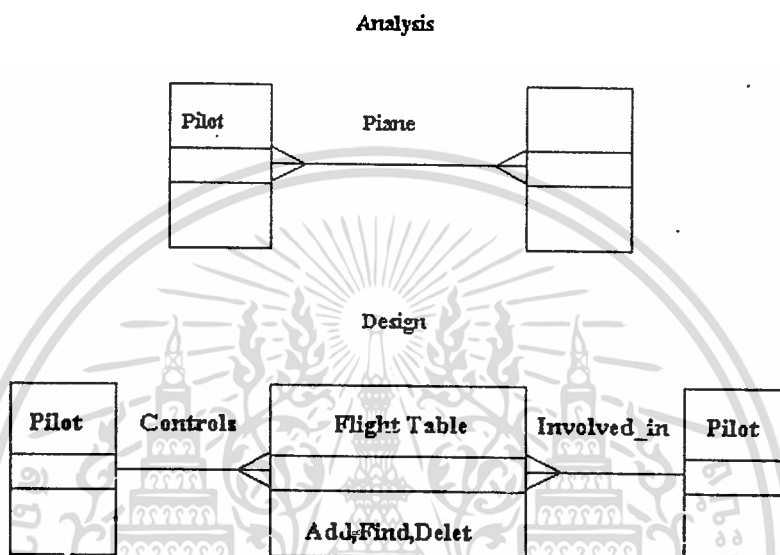
- ระบบควรจะเป็นแบบ Centralized หรือ Distribute
- ถ้าเป็นระบบ Distributed การจะจัดตำแหน่ง Object ควรจะนำ Object ไปไว้ที่ใดจึงจะสามารถทำงานได้ตรงตามความต้องการและได้ประสิทธิภาพสูงสุดและจะใช้อะไรเป็นตัวติดต่อสื่อสาร
- พิจารณาประสิทธิภาพของเครื่องที่จะนำมาใช้ในการทำงานของโปรแกรม

#### 2) การพิจารณาในส่วนของ Software

- ระบบที่จะพัฒนาขึ้นนี้ ควรที่จะแยกออกเป็น โปรแกรมมากกว่า 1 โปรแกรมหรือไม่
- Database ใดที่ควรนำมาใช้ในระบบ
- ควรจะเพิ่ม Classes ใดขึ้นมาเพื่อทำให้ระบบสมบูรณ์ขึ้น และสนับสนุนการสร้างหน้าจอโปรแกรมใช้งานตามที่ต้องการได้ หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่ม operation ใน Class เพื่อเพิ่มความเข้าใจ และสะดวกในการเขียนโปรแกรม
- ควรเพิ่มเติม Classes เพื่อทำให้ความสัมพันธ์ระหว่าง Classes เข้าใจได้ง่ายขึ้นหรือไม่ เช่น จากการวิเคราะห์ อาจได้ Classes ซึ่งมีความสัมพันธ์แบบ Many-to-Many ดังนั้น อาจจะเพิ่มอีก Class หนึ่งเพื่อทำให้เป็นความสัมพันธ์แบบ One-to-Many ดังรูปที่ 2.14



รูปที่ 2.11 การเพิ่ม Classes จาก ขั้นตอน Analysis ในขั้นตอนการ Design

ขั้นตอนนี้จะนำเสนอด้วยแบบจำลอง UML คือ

1) Class Diagram ซึ่งจะนำ Class ที่ได้ในขั้นตอนของการวิเคราะห์ระบบเชิงวัตถุมาเพิ่มเติม เพื่อให้เข้าในการทำงานของระบบง่ายขึ้นและสามารถพัฒนาโปรแกรมได้ง่ายขึ้นด้วย

2) Sequence Diagram หรือ State Diagram โดยจะเป็นการแสดงให้เห็นถึงการทำงานระหว่าง Object ต่าง ๆ ตามการส่งข้อความ (Message) และเมื่อเหตุการณ์ (Event) ต่าง ๆ ได้เกิดขึ้น จะแสดงถึงการส่ง Message เพื่อเป็นการติดต่อสื่อสารระหว่าง Object ต่าง ๆ ที่เกี่ยวข้อง ซึ่งสามารถนำเสนอได้เช่นเดียวกับ Collaboration Diagram แต่การนำเสนอด้วย Sequence Diagram นี้ จะแสดงให้เห็นถึงลำดับช่วงเวลาที่แน่นอนของการส่ง Message

#### 2.2.1.4 ขั้นตอนการเขียนโปรแกรม (Programming)

ในขั้นตอนของการเขียน โปรแกรมเชิงวัตถุ จะเป็นการนำเอา Class และ Model ต่าง ๆ ที่ได้ จากขั้นตอนของการวิเคราะห์และออกแบบระบบเชิงวัตถุ มาเขียนเป็น Code Program

จาก UML ซึ่งเป็น Model ที่ใช้เป็นสื่อกลางในการนำเสนอแนวความคิดในการวิเคราะห์ และออกแบบระบบ จะทำให้เราสามารถเขียนโปรแกรมได้สะดวกขึ้น โดยที่ผู้เขียนโปรแกรมไม่ ต้องกังวลในการแปลงจาก Model มาเป็น Code Program เนื่องจากรูปแบบของ UML ที่กำหนดนั้น มีประโยชน์ในการช่วยให้ผู้ใช้งานได้ทำความเข้าใจถึงโครงสร้างของระบบได้ดียิ่งขึ้น ดังนั้นการคว่ นสรุปถึงตัว Code นั้น จะไม่เกิดประโยชน์ต่อการพัฒนา เมื่อเทียบกับวิธีแบบมาตรฐานที่ถูกต้อง นอก จากนี้ Rational Rose ซึ่งเป็น Case Tools ในการเขียน Model UML ยังมีฟังก์ชันช่วยในการสร้าง เป็น Code โปรแกรมอย่างง่าย ตาม Model ที่ได้ออกแบบไว้

#### 2.2.1.5 ขั้นตอนการทดสอบระบบ (Testing)

ในขั้นตอนของการทดสอบระบบ จะแบ่งการทดสอบออกเป็นระดับต่าง ๆ คือ Unit Test, Integration Test, System Test และ Acceptance Test

1) **Unit Test** จะเป็นการทดสอบการทำงานในส่วนของ Class หรือกลุ่มของ Class ซึ่งการ ทดสอบนี้จะกระทำโดย Programmer

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ Class Diagram

2) **Integration Test** จะเป็นการทดสอบการทำงานโดยรวมของ Component หรือ Class เพื่อที่จะบรรลุตามความต้องการบางอย่างของระบบ

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ Components Diagram และ Collabolation/Sequence Diagrams

3) **System Test** จะเป็นการทดสอบทั้งระบบ โดยทดสอบความถูกต้องของการทำงานของ ระบบ ซึ่งจะทดสอบโดย End User

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ User Case Diagram อันแสดงถึงการ ทำงานของระบบโดยรวม

4) **Acceptance test** จะเป็นการทดสอบระบบโดย User ว่าระบบที่ได้นั้น ตรงตามความ ต้องการหรือไม่

Diagram ที่เกี่ยวข้องกับการทดสอบระดับนี้คือ User Case Diagrams เช่นเดียวกับระดับ System Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1.6 ขั้นตอนการบำรุงรักษา (Maintenance)

ในการ Maintenance ระบบเชิงวัตถุจะสามารถทำได้สะดวกกว่าระบบเชิงโครงสร้าง เช่น ถ้ามีการเปลี่ยนแปลงโครงสร้างของ Class ก็จะมีการเปลี่ยนแปลงเฉพาะ Class ที่ต้องการเปลี่ยนแปลงเท่านั้น โดยที่จะไม่ส่งผลกระทบต่อ Class อื่น

### 2.2.2 ข้อดี-ข้อเสีย ของการวิเคราะห์และออกแบบระบบเชิงวัตถุเปรียบเทียบกับเชิงโครงสร้าง ข้อดี

1) ผลลัพธ์ที่ได้จากการวิเคราะห์ระบบเชิงโครงสร้าง จะได้แบบจำลอง DFDs ซึ่งสามารถนำเสนอได้เฉพาะมุมมองในแง่การทำงานของระบบ แต่ไม่ได้ระบุถึงมุมมองในแง่การวิเคราะห์เพื่อพัฒนาโปรแกรม ในขณะที่การวิเคราะห์เชิงวัตถุมองระบบในมุมมองเชิงวัตถุ จึงทำให้การวิเคราะห์และออกแบบระบบทำได้ต่อเนื่องกัน โดยตรง และง่ายต่อการพัฒนาระบบโดยใช้ภาษาโปรแกรมเชิงวัตถุ (Object-Oriented Programming Language) และ/หรือระบบจัดการฐานข้อมูลเชิงวัตถุ

2) จากหลักการของการวิเคราะห์ระบบเชิงวัตถุที่มองระบบในแง่เชิงวัตถุเสมือนโลกแห่งความจริง จึงทำให้สามารถอธิบายถึงภาพรวมของระบบได้โดยง่าย และสามารถสื่อสารระหว่างผู้วิเคราะห์ระบบกับผู้ใช้งานระบบได้เข้าใจดีกว่าการวิเคราะห์ระบบเชิงโครงสร้าง

3) จากข้อดีต่าง ๆ ของการพัฒนาระบบเชิงวัตถุ เช่นการนำกลับมาใช้ใหม่ (Reusability) เป็นต้น จึงทำให้การพัฒนาระบบเชิงวัตถุเริ่มได้รับความนิยมมากกว่าการพัฒนาระบบเชิงโครงสร้าง ซึ่งการวิเคราะห์ระบบเชิงวัตถุก็สนับสนุนการพัฒนาระบบเชิงวัตถุมากกว่าการวิเคราะห์ระบบเชิงโครงสร้าง

4) การมองระบบในมุมมองเชิงวัตถุช่วยให้การสร้างหน้าจอใช้งานสามารถทำได้สะดวกขึ้น เนื่องจากการมองระบบเชิงวัตถุทำให้ทราบว่าแต่ละวัตถุจะต้องมีคุณสมบัติและพฤติกรรมอะไรบ้าง ในขณะที่การวิเคราะห์และออกแบบระบบเชิงโครงสร้างสามารถช่วยให้เห็นมุมมองของหน้าจอการใช้งานระบบได้เพียงเล็กน้อย

### ข้อเสีย

- 1) ผู้วิเคราะห์ระบบจะต้องมีความชำนาญในหลักการเชิงวัตถุเป็นอย่างดี
- 2) ภาษาโปรแกรมเชิงวัตถุ เป็นภาษาที่ค่อนข้างซับซ้อนในการที่จะสร้างระบบเชิงวัตถุให้บรรลุได้ตามวัตถุประสงค์
- 3) เนื่องจากระบบเชิงวัตถุค่อนข้างมีหลักการที่ซับซ้อน จึงทำให้การทดสอบระบบทำได้ยากด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการวิเคราะห์ระบบเชิงวัตถุอาจจะต้องใช้เวลานานกว่าการวิเคราะห์ระบบเชิงโครงสร้าง เนื่องจากจะต้องเก็บรายละเอียดให้ได้ทั้งหมด หรือมากที่สุด

## 2.3 Java Server Pages (เรียนรู้ JSP พร้อมใช้งาน. 2545: 45)

### 2.3.1 ความหมายของ Java Server Pages

Java Server Page หรือที่เรียกกันง่ายๆว่า JSP เป็นเทคโนโลยีที่เป็นผลรวม ของการนำเอาหลักการสร้างเว็บแบบสแตติกด้วย HTML มารวมกับการสร้างเว็บแบบไดนามิก ก่อให้เกิดการแสดงผลเว็บ ที่มีการเปลี่ยนแปลงข้อมูลได้ตามการใช้งานของผู้ใช้ เห็นได้ว่าหากพูดถึงหลักการนำเสนอเว็บแล้ว เป็นหลักการที่ทำให้เว็บมีการเปลี่ยนแปลงข้อมูล เช่นเดียวกับ ที่ใช้โปรแกรมจําพวก CGI ทั่วไป แต่ในความหมายของ JSP แล้ว เป็นขั้นตอนและวิธีการสร้างที่ทำงานร่วมกับคำสั่ง HTML และเก็บบันทึกไว้ในไฟล์เดียวกัน แต่ในขณะที่ CGI เป็น โปรแกรมที่ผลิต HTML ให้เมื่อถูกสั่งให้ทำงาน ตัวอย่างคำสั่ง JSP กับคำสั่ง HTML ดังนี้

```
<HTML><HEAD><TITLE>Show JSP Page</TITLE></HEAD>
<BODY>
<% out.println("Hello Java Server Page"); %>
</BODY>
</HTML>
```

จากตัวอย่างด้านบนเป็นเพียงส่วนหนึ่งของ JSP ที่ถูกใส่ไว้ร่วมกับคำสั่ง HTML สังเกตจากตัวหนาที่อยู่ใน แท็กพิเศษ <% ... %> ซึ่งเป็นคำสั่งในรูปแบบ JSP ทำให้ส่วนดังกล่าวสร้างข้อมูลในแบบไดนามิก ไว้ในเว็บเพจตามคำสั่ง ไฟล์ดังกล่าวจะบันทึกไว้ในนามสกุล .jsp จาก HTML เดิมรูปแบบดังกล่าวไม่สามารถทำได้สำเร็จตามเทคโนโลยี JSP หากไฟล์ตัวอย่างด้านบน ไม่ได้ถูกเก็บไว้ในเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงานของ JSP ข้อความที่เป็นไดนามิกนั้น ถูกสำรวจเพื่อประมวลผล และใส่ลงในตำแหน่งดังกล่าว ด้วยความสามารถของ เซิร์ฟเวอร์ที่สนับสนุน JSP ก่อนส่งกลับไปให้ผู้ใช้งาน ซึ่งผู้ใช้โดยผู้ใช้งานจะได้รับข้อความ Hello Java Server Page เท่านั้น โดยไม่เห็นคำสั่ง JSP เลย

ตัวอย่างที่ผู้ใช้งานมองเห็นจากบราวเซอร์

Hello Java Server Page

### ข้อแตกต่างของ JSP เมื่อเทียบกับเทคโนโลยีอื่นๆ

JSP ไม่ได้เป็นเทคโนโลยีเดียว ในปัจจุบันที่สามารถทำให้เว็บ แสดงข้อมูลในแบบไดนามิกได้ มีเทคโนโลยีอื่นๆที่ทำงานในลักษณะนี้หลายแบบ เนื้อหาส่วนนี้ เป็นการแนะนำให้อ่านได้เห็นข้อแตกต่างบางประการ เมื่อเทียบกับ JSP

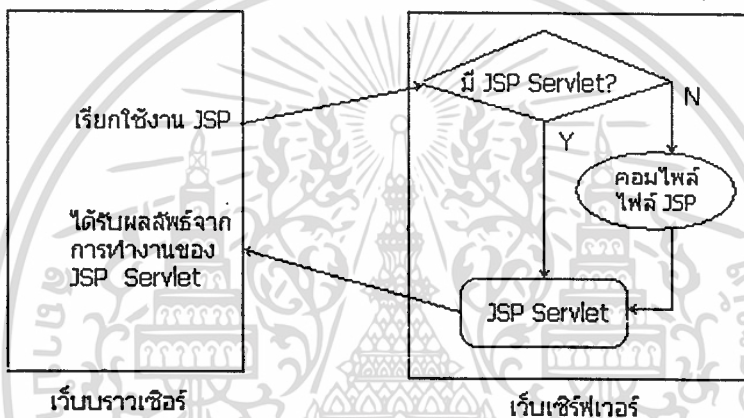
- **Active Server Page(ASP)** ASP เป็นเทคโนโลยีที่เหมือนกับ JSP แต่เป็นเทคโนโลยีที่พัฒนามาจาก บริษัทไมโครซอฟต์ โดยทั่วไป ลักษณะ JSP มีรูปแบบที่แตกต่างกันเด่นชัด 2 ประการคือ ประการแรก JSP สามารถสร้างได้จาก ภาษาจาวา ต่างจากภาษา VB หรือภาษาใดๆ ซึ่งเป็นข้อกำหนดเฉพาะของ ไมโครซอฟต์ ประการที่สอง JSP ซึ่งเป็นผลพวงมาจาก ประการแรกคือ สามารถโยกย้ายการทำงาน ไปใช้งานระบบปฏิบัติการใดๆก็ได้ ที่มีการใช้งานกันอยู่ในปัจจุบัน โดยไม่ต้องแก้ไขหรือเปลี่ยนแปลงใดๆ ในขณะที่ ASP ก็โยกย้ายได้เช่นเดียวกัน แต่ต้องอยู่ในระบบปฏิบัติการที่ ไมโครซอฟต์กำหนดขึ้นเท่านั้น
- **Servlet** JSP สามารถทำงานได้เช่นเดียวกับที่ Servlet ทำได้ แต่มีจุดเด่นมากกว่าที่ JSP มีความสะดวกในการสร้าง และเปลี่ยนแปลง มากกว่า เพราะทำโดยตรงที่ไฟล์ HTML มากกว่าการที่ต้องลงมือเขียนคำสั่งภาษาจาวา โดยตรงเหมือนกันสร้าง servlet ซึ่งต้องมีการนำไปคอมไพล์ ก่อนการนำไปใช้งาน
- **Server-Side Includes(SSI)** SSI คือรูปแบบการสร้างไดนามิกเว็บเช่นเดียวกัน แต่รูปแบบการทำงานคือ การนำเอาข้อมูลที่มีอยู่แล้ว เท่านั้น ต่างจาก JSP ซึ่งสามารถนำข้อมูลมาใส่ แต่มีรูปแบบในการทำงาน มีรูปแบบในการประมวลผล หรือการเรียกโปรแกรมภายนอก เช่น Servlet มาช่วยในการทำให้ข้อมูล มีรูปแบบต่างๆ เช่น ดึงข้อมูลจากระบบฐานข้อมูล (Database) นอกจากข้อมูลที่มีอยู่บนเซิร์ฟเวอร์เช่นที่ใช้ใน SSI เท่านั้น
- **JavaScript** เป็นการทำเนื้อหาไดนามิกให้กับเว็บ แต่รูปแบบการทำงานเกิดขึ้นของ JavaScript เกิดจากการประมวลผลและดึงข้อมูล ที่มีอยู่บนเครื่องผู้ใช้ หรือไคลเอนท์ เท่านั้น มารวบรวมเนื้อหาในเว็บ ต่างกับ JSP ที่เป็นการทำไดนามิก แต่ข้อมูลถูกสร้าง และดึงมาจากระบบเซิร์ฟเวอร์ ที่มีหลากหลายมากกว่า และทำให้เนื้อข้อมูลเป็นเนื้อเดียวกัน เมื่อผู้ใช้เรียกดู แต่ JavaScript เนื้อข้อมูลเป็นของเครื่องผู้ใช้เอง ซึ่งผู้ใช้ต่างคน(ต่างเครื่อง) ก็จะทำให้ข้อมูลที่คล้ายกัน คือไม่เหมือนกันทั้งหมด
- **Static HTML หรือสแตติก HTML** แน่นนอนข้อนี้ คงเห็นได้ชัดเจนที่ว่า สแตติก HTML ให้เนื้อข้อมูลที่คงที่ตลอดเวลา ไม่ว่าจะถูกเรียกดูในเวลาใดๆ (ยกเว้นมีคนมาแก้ไขคำสั่ง HTML) ในขณะที่ JSP ทำให้เนื้อข้อมูลมีการเปลี่ยนแปลงตามการใช้งาน หรือเครื่องผู้ใช้ โดยไม่ต้องเปลี่ยนแปลงแก้ไขคำสั่ง JSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 หลักการทำงานของ JSP

Java Servlet คือเซิร์ฟเล็ตที่เกิดขึ้น จากชุดคำสั่ง JSP ที่ถูกคอมไพล์ไปเป็น JSP Servlet โดยขบวนการนี้ เกิดโดยอัตโนมัติภายในเว็บเซิร์ฟเวอร์ที่สนับสนุน การทำงานของ JSP และ JSP Servlet นี้เองที่เป็นส่วนที่เกิดการประมวลผลบนเซิร์ฟเวอร์ ดังนั้นไฟล์นามสกุล JSP ที่ถูกเรียกใช้งาน จะต้องถูกคอมไพล์ไปเป็น JSP Servlet ก่อนในครั้งแรกของการเรียกใช้งาน หรือเมื่อมีการเปลี่ยนแปลงชุดคำสั่ง

การทำงานของ JSP

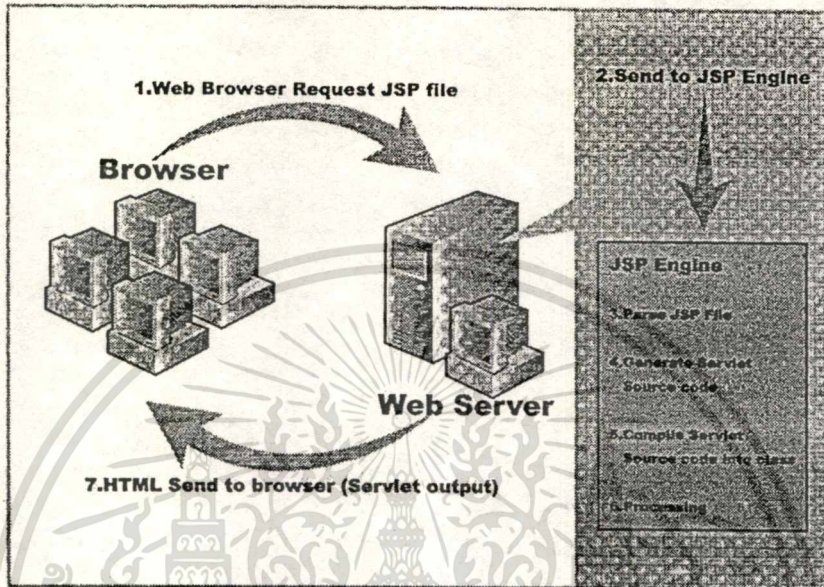


รูปที่ 2.12 การทำงานของ JSP

เมื่อเว็บเบราว์เซอร์ มีการร้องขอโดยผู้ใช้เพื่อเรียกข้อมูล JSP ที่อยู่บนเซิร์ฟเวอร์ ซึ่งก็หมายถึงไฟล์ที่มีนามสกุล JSP เว็บเซิร์ฟเวอร์จะทำการตรวจสอบว่า ไฟล์นั้นมีการคอมไพล์ไปเป็น JSP Servlet แล้วหรือยัง ถ้ายังไฟล์ JSP จะถูกคอมไพล์ โดยตัวเว็บเซิร์ฟเวอร์ ไปเป็น JSP Servlet แต่ถ้าไฟล์ถูกคอมไพล์แล้ว (ซึ่งเกิดขึ้นจากการใช้งานในครั้งแรก) เว็บเซิร์ฟเวอร์จะนำ JSP Servlet มาใช้งาน โดยการเรียกเพื่อประมวลผล ได้ผลลัพธ์อย่างไรก็จะเป็นข้อมูล ซึ่งเว็บเซิร์ฟเวอร์ จะส่งกลับไปให้ผู้เรียกใช้งาน และการประมวลผลของ JSP Servlet นี้เอง ที่ก่อให้เกิด ข้อมูลเว็บในแบบไดนามิก

### 2.3.3 สถาปัตยกรรมของ JSP

ทุกครั้งที่มีการเรียกเอกสาร JSP ที่เราสร้างขึ้นมาทางบราวเซอร์ หากเป็นการเรียกครั้งแรก จะช้ากว่าครั้งถัดไปเสมอ ซึ่งสามารถอธิบายได้ดังนี้



รูปที่ 2.13 สถาปัตยกรรมของ JSP

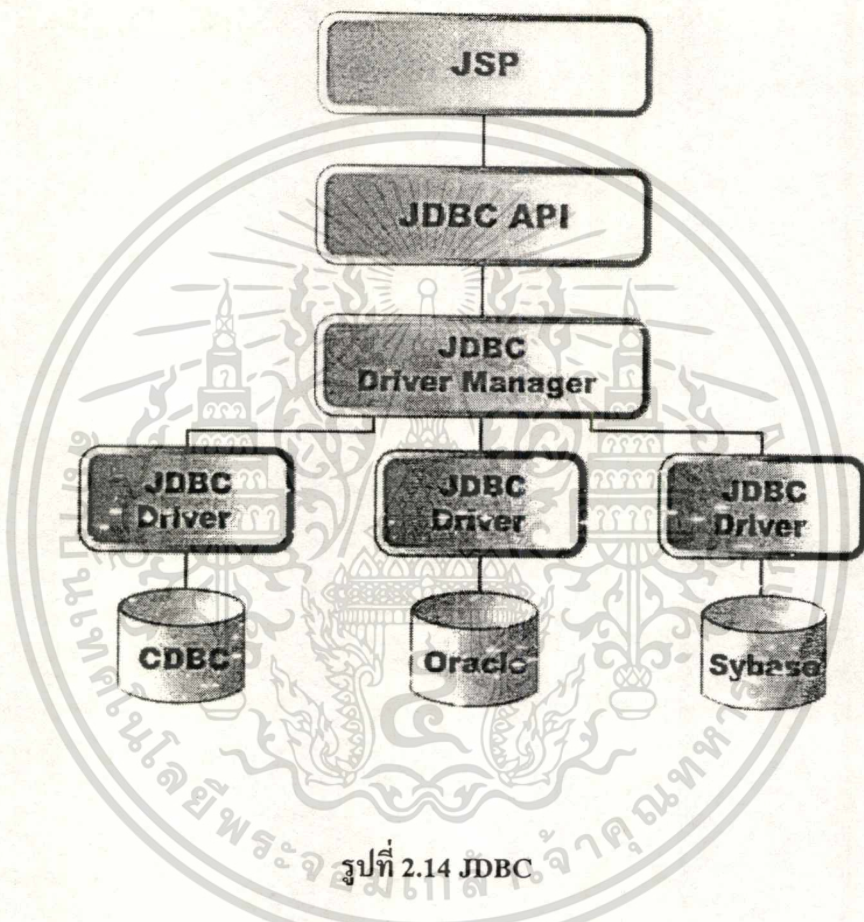
จากรูปเป็นการแสดงขั้นตอนที่เกิดขึ้นเมื่อมีการเรียกเอกสาร JSP ในการเรียกครั้งแรกจะมีขั้นตอนดังนี้

- 1.บราวเซอร์ร้องขอเอกสาร JSP มายังเซิร์ฟเวอร์
- 2.เซิร์ฟเวอร์จะส่งผ่านการร้องขอนั้น ไปให้กับ JSP Engine
- 3.JSP Engine ก็จะคอมไพล์เอกสาร JSP นั้น
- 4.เมื่อการคอมไพล์ผ่าน โดยไม่มีข้อผิดพลาด JSP Engine ก็จะแปลงเอกสาร JSP นั้นให้  
เป็นชุดคำสั่งของ Servlet
- 5.ชุดคำสั่งของ Servlet เหล่านั้นก็จะถูกแปลงเป็นคลาส
- 6.เริ่มทำงานตามคำสั่งที่เขียนมา
- 7.เมื่อได้ผลลัพธ์แล้วก็จะส่งผลลัพธ์ในลักษณะเอกสาร HTML นั้นกลับไปให้เว็บ  
เซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ก็จะส่งผลลัพธ์ต่อไปให้กับบราวเซอร์อีกที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 JDBC

JDBC คือ ตัวเชื่อมต่อระหว่างโปรแกรมกับฐานข้อมูล โครงสร้างของ JDBC เป็นกลุ่มอินเทอร์เฟซของจาวาที่เรียกว่า (Driver) ฐานข้อมูลแต่ละประเภทก็มีไดรเวอร์ต่างชนิดกันออกไป แต่เราไม่จำเป็นต้องรู้ถึงโครงสร้างของฐานข้อมูลที่เราต้องการจะติดต่อ เพียงแค่รู้ชนิดของไดรเวอร์ที่ใช้ในฐานข้อมูลชนิดนั้นก็พอ ลองพิจารณารูปดังต่อไปนี้



รูปที่ 2.14 JDBC

จากรูปจะเห็นว่าเมื่อ JSP ต้องการติดต่อกับฐานข้อมูล ก็จะผ่านทาง JDBC API และ JDBC Driver Manager ส่วนไดรเวอร์จะเป็น ODBC, Oracle, Sybase หรือ ฐานข้อมูลชนิดอื่นก็ได้ สำหรับไดรเวอร์ของฐานข้อมูลอื่นๆ นั้น ผู้อ่านต้องติดตั้งเอง วิธีการก็คือ เมื่อได้ไดรเวอร์มาแล้ว (ไดรเวอร์จะอยู่ในรูปแบบของคลาสหรือแพ็คเกจ) ก็นำไดรเวอร์มาติดตั้งในไคลเรทอรีที่มีอยู่ในตัวแปร classpath สำหรับในบทนี้จะใช้ไดรเวอร์ ODBC ซึ่งมีอยู่แล้วในคลาส "java.sql.\*" สามารถใช้ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดต่อกับฐานข้อมูล

ไม่ว่าเราจะเลือกติดต่อกับฐานข้อมูลชนิดไหน ขั้นตอนติดต่อก็ใช้ JDBC นั้น ประกอบด้วย 5 ขั้นตอนดังนี้

1. กำหนดชนิดของไดรเวอร์
2. สร้างการเชื่อมต่อไปยังฐานข้อมูล
3. สร้างสเตทเมนต์ (Statement)
4. ประมวลผล SQL และจัดการกับข้อมูล
5. ปิดการเชื่อมต่อ

ขั้นตอนที่ 1 – กำหนดชนิดของไดรเวอร์

การกำหนดชนิดของไดรเวอร์เพื่อเชื่อมต่อกับฐานข้อมูลนั้น ใช้โค้ดดังต่อไปนี้

```
Class.forName (<Driver Type>)
```

เวลาเราเลือกชนิดของไดรเวอร์ ออบเจกต์ DriverManager จะตรวจสอบก่อนว่าระบบของเรานั้นได้ติดตั้งไดรเวอร์ไว้หรือยัง วิธีการตรวจสอบก็มี 2 วิธี แต่วิธีที่ผมขอกล่าวคือ วิธีไดรเวอร์รีจิสตรชัน (Driver Registration) ครับ สำหรับวิธีนี้ก็คือเมื่อเรากำหนดชนิดของไดรเวอร์แล้ว เราก็จะบอกกับออบเจกต์ DriverManager ว่าต้องการตรวจสอบไดรเวอร์ว่ามีอยู่ในระบบหรือไม่ ด้วยการเรียกเมธอด newInstance () ยกตัวอย่างเช่น หากเราต้องการติดต่อกับ ODBC ก็กำหนดดังนี้

```
Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver").newInstance ()
```

หากระบบของเรามีไดรเวอร์ จะทำให้ออบเจกต์ DriverManager สามารถสร้างอินสแตนซ์ของไดรเวอร์นั้นได้ แต่ถ้าไม่มีจะเกิด Exception ที่บอกถึงชนิดไดรเวอร์ที่เราเรียกใช้ เช่น หากผมพยายามเรียกใช้ไดรเวอร์ของ Oracle ซึ่งไม่มีลงอยู่ในเครื่องผมจะมีข้อผิดพลาดคือ

```
Unable to load class oracle.jdbc.driver.OracleDriver
```

ขั้นตอนที่ 2-สร้างการเชื่อมต่อไปยังฐานข้อมูล

หลังจากกำหนดไดรเวอร์ที่จะใช้เชื่อมต่อไปยังฐานข้อมูลแล้ว ขั้นตอนต่อไป เราจะเชื่อมต่อไปยังฐานข้อมูลโดยใช้ออบเจกต์ที่ชื่อว่า Connection มีรูปแบบดังนี้

```
Connection conn = DriverManager.getConnection  
(db_url, "login", "pwd")
```

ออบเจกต์ Connection นั้นสร้างได้จากการเรียกเมธอด getConnection () ของออบเจกต์ DriverManager โดยหากิวเมนต์ที่ผ่านให้เมธอดตามปกติจะมี 3 ตัวแต่ก็ขึ้นอยู่กับฐานข้อมูลที่เรา

ต้องการใช้ด้วยครับ อากิวเมนต์ตัวแรกคือ db\_url โดยทั่วไปแล้วการเชื่อมต่อจะทำโดยใช้ db\_url ซึ่งมีรูปแบบดังนี้

**Jdbc:driver\_name:database\_name**

หรือ

**Jdbc:driver\_name:// host/database\_name**

รูปแบบของ db\_url จะมีสามส่วน ส่วนแรกจะต้องขึ้นต้นด้วย jdbc ส่วนที่สองคือไดรเวอร์ที่จะใช้ และส่วนสุดท้ายคือข้อมูลเพิ่มเติมเพื่อใช้ในการเชื่อมต่อ ยกตัวอย่างเช่น

**Jdbc:odbc:jdbc\_registration**

จากตัวอย่างหากเราต้องการใช้ไดรเวอร์เป็น odbc เราต้องระบุชื่อ Data Source ด้วย หากเราต้องการเชื่อมต่อกับ odbc รูปแบบการประกาศคือ

```
Connection con = DriverManager.getConnection
("jdbc:odbc: jdbc_registration")
```

### ขั้นตอนที่ 3-สร้างสเตทเมนต์ (Statement)

เมื่อไดรเวอร์โหลดและเชื่อมต่อไปยังฐานข้อมูลแล้ว ขั้นตอนต่อไปคือ ต้องส่งคำสั่ง SQL เข้าไปยังฐานข้อมูลเพื่อบันทึกข้อมูลกับข้อมูลภายใน วิธีการคือ ส่งผ่านออบเจกต์ statement ออบเจกต์นี้สร้างจากเมธอด createStatement () ของออบเจกต์ Connection มีรูปแบบคือ

```
Statement stmt = con.createStatement ();
```

### ขั้นตอนที่ 4-ประมวลผล SQL และจัดการกับข้อมูล

เมื่อเราสร้างออบเจกต์ statement แล้ว เราจะส่งคำสั่ง SQL เข้าไปครับ วิธีการส่งต้องเรียกเมธอดของออบเจกต์ Statement สองเมธอดคือ executeQuery () และ executeUpdate () ทั้งสองเมธอดมีวิธีใช้ต่างกันขึ้นอยู่กับคำสั่ง SQL ที่เราส่งเข้าไปรูปแบบการประกาศคือ

```
stmt.executeQuery(sql); หรือ stmt.executeUpdate (sql);
```

เมธอด executeQuery () จะใช้กับคำสั่ง SQL ที่ได้กลุ่มผลลัพธ์กลับมาคือ SELECT ส่วนคำสั่ง SQL ที่ไม่ส่งผลลัพธ์กลับมาคือ UPDATE, INSERT และ DELETE นั้น จะใช้เมธอด executeUpdate () เมธอดนี้จะผ่านค่าของจำนวนแถวที่ถูกคำสั่ง SQL จัดการไป ตัวอย่างเช่น

```
string sql = "Select * from address"
```

```
ResultSet rs = stmt.executeQuery (sql)
```

และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
String sql = "Delete * from address"
```

```
Int retVal = stmt.executeUpdate (sql)
```

หากเราใช้คำสั่ง SQL ไม่ตรงกับเมธอดจะเกิด Exception คือ

No ResultSet was produced

เมื่อเราเรียกเมธอด `executeQuery ( )` ผลลัพธ์ที่ได้กลับมาจะเป็นออบเจกต์ `ResultSet` ซึ่งออบเจกต์ตัวนี้จะมีคอลัมน์และแถวของข้อมูล และตัวชี้ตำแหน่ง (Cursor) อยู่ โดยตัวชี้ตำแหน่งทำหน้าที่ชี้ไปยังแถวปัจจุบันของข้อมูล โดยค่าเริ่มต้นจะอยู่ที่แถวแรกเสมอ หากเราต้องการดำเนินการกับข้อมูลก็ใช้วิธีเลื่อนตัวชี้ตำแหน่งไปที่ละหนึ่งแถวโดยเรียกเมธอด `next ( )` ของออบเจกต์ `ResultSet` ดังนี้

```
while (rs.next ( ) ) {
}
```

ผลลัพธ์ที่ได้จากการเรียกเมธอด `next ( )` จะมีค่าเป็นบูลีน เพื่อบอกว่ายังมีแถวของข้อมูลอีกหรือไม่ สำหรับการเข้าถึงข้อมูลของแถวในแต่ละคอลัมน์ ก็ทำโดยใช้เมธอดของออบเจกต์ `ResultSet` คือ `getString ( )`, `getInt ( )` หรือ `getDate ( )` เราต้องเรียกแต่ละเมธอดให้ตรงกับชนิดของข้อมูลในคอลัมน์ด้วย สำหรับอักขระที่ผ่านไปที่เมธอดนี้คือสตริงที่แทนชื่อของคอลัมน์ หรือเราจะส่งตำแหน่งของคอลัมน์เข้าไปก็ได้ โดยคอลัมน์แรกจะเริ่มที่ 1 และถัดไปเรื่อย ๆ จนหมด (ผมมักจะผิดพลาดบ่อย เนื่องจากสับสนกับการอ้างตำแหน่งของอาร์เรย์ คืออาร์เรย์ในจาวาจะเริ่มที่ 0 ท่านผู้อ่านต้องระวังด้วยนะครับ

### ขั้นตอนที่ 5-ปิดการเชื่อมต่อ

หลังจากที่เรานำผลลัพธ์ที่ได้จากคำสั่ง SQL มาจัดการแล้วก็ถือว่าจบการติดต่อกับฐานข้อมูล เราต้องปิดการเชื่อมต่อเพื่อให้แน่ใจว่า ออบเจกต์ `Connection` และออบเจกต์ `Statement` ได้จบการทำงาน โดยมีวิธีการคือ

```
con.close ( )
```

```
stmt.close ( )
```

## บทที่ 3

### วิธีดำเนินการและการรวบรวมข้อมูล

#### 3.1 กรณีศึกษา

โครงการระบบจัดการข้อมูลสินค้าคงคลังแบบฝากขาย (Information System for Consignment Inventory System) จะเป็นการศึกษาถึงกระบวนการทำงานของระบบงานที่ทำอยู่ในปัจจุบัน โดยการใช้เพิ่มเอกสารในการจัดการกับข้อมูลสินค้าคงคลัง ซึ่งจะต้องปรับเปลี่ยน เพิ่มเติม โดยการนำคอมพิวเตอร์เข้ามาช่วยในการเก็บข้อมูลบุคลากรของระบบ ข้อมูลจุดขายและรายละเอียด ข้อมูลสินค้าและรายละเอียด เช่น สี ขนาด เป็นต้น ข้อมูลการขายสินค้ารายวัน และข้อมูลการเคลื่อนไหวสินค้าแต่ละจุดขาย ซึ่งข้อมูลดังกล่าวมานั้นล้วนแล้วแต่มีความสำคัญยิ่ง ซึ่งปัจจุบันข้อมูลมีปริมาณมาก เพราะมีชนิดสินค้าหลายสิบชนิด หลายขนาด และหลายสี รวมทั้งมีจุดขายมากถึง 50 จุดขายและอยู่กระจายกันทั่วกรุงเทพฯ ซึ่งการศึกษางานของระบบนี้จะทำการสอบถามข้อมูล จากความต้องการของผู้ใช้งานระบบและผู้เกี่ยวข้องอันได้แก่ พนักงานฝ่ายขาย พนักงานฝ่ายตรวจสอบ เพื่อที่ทำให้เราสามารถดำเนินการวิเคราะห์และออกแบบระบบได้ตรงตามความต้องการของผู้ใช้งาน และมีประสิทธิภาพยิ่ง

#### 3.2 เครื่องมือที่ใช้ในโครงการ

ในโครงการระบบจัดการข้อมูลภายในคลินิกโดยหลักการพัฒนาระบบเชิงวัตถุ เราจะใช้เครื่องมือช่วยในการทำงาน ซึ่งประกอบด้วย

1. อุปกรณ์ Hardware ดังนี้
  - 1.1 คอมพิวเตอร์จำนวน 1 เครื่อง เป็นพิมพ์ข้อมูล (Keyboard) พร้อม Mouse จำนวน 1 ตัว
  - 1.2 เครื่องพิมพ์ (Printer) 1 เครื่อง
2. Software ในการพัฒนาระบบ ดังนี้
  - 2.1 ระบบปฏิบัติการ (Operating System) จะใช้ Window 98 Thai Edition
  - 2.2 Rational Rose 2000 เป็น Case Tools ในการนำเสนอแนวความคิดการออกแบบ และวิเคราะห์ระบบเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.1 ใช้ Microsoft SQL Server 2000 DBMS Database เป็นระบบจัดการฐานข้อมูล
- 1.2 ใช้ Apache Web Server เป็นเว็บเซิร์ฟเวอร์
- 1.3 ใช้ JSP (Java Server Pages) ในการพัฒนาระบบงาน

### 3.3 การเก็บรวบรวมข้อมูล

เก็บรวบรวมข้อมูลทั้งหมดที่ใช้ในระบบงาน โดยจะศึกษาถึงกระบวนการขั้นตอนต่าง ๆ ในการดำเนินการภายในของการฝากขายสินค้าตามจุดขายต่างๆ และข้อมูลต่าง ๆ ที่จำเป็นต้องจัดเก็บในระบบและจะต้องดูว่าความต้องการในระบบที่จะพัฒนามีอะไรบ้าง เช่น จากเดิมที่บันทึกการจ่ายแบบ Manual ซึ่งอาจมีข้อผิดพลาดในการจดบันทึกได้ ก็อาจจะต้องการระบบที่สามารถเตือนได้เมื่อจำนวนสินค้าแต่ละจุดขายเหลือน้อยกว่าที่กำหนด รวมทั้งเก็บข้อมูลว่าข้อมูลใดจะต้องเป็นข้อมูลนำเข้าหรือข้อมูลออกจากกระบวนการใด ๆ และต้องมีเอกสารหรือรายงานใดบ้างจากระบบ

ในขั้นตอนของการเก็บรวบรวมข้อมูลนี้จะศึกษาจากระบบเก่าที่ทางเจ้าหน้าที่กำลังดำเนินการอยู่และสอบถามความต้องการเพิ่มเติมจากผู้ใช้จากระบบ เช่นว่าอะไรคือสิ่งที่ระบบควรจะเป็นเพื่อที่จะเรียบเรียงเป็นความต้องการของระบบ และทำให้สามารถพัฒนาระบบที่บรรลุตามวัตถุประสงค์ที่วางไว้ได้ ดังนั้นจึงควรอธิบายความต้องการของระบบให้ชัดเจน เพื่อที่จะพัฒนาระบบได้ถูกต้อง ตรงตามความต้องการต่อไป โดยในขั้นตอนนี้ อาจจะนำเสนอด้วยแบบจำลอง Use Case Diagram เพื่อแสดงถึงแนวความคิดตามที่ผู้รวบรวมข้อมูลเข้าใจ ว่าถูกต้องตรงกับการทำงานของระบบหรือไม่

### 3.4 การวิเคราะห์และขั้นตอนการพัฒนาระบบ

การวิเคราะห์และการพัฒนาระบบจัดการข้อมูลสินค้าคงคลัง โดยหลักการพัฒนาระบบเชิงวัตถุ จะนำเอาข้อมูลที่ได้จากขั้นตอนของการรวบรวมข้อมูลมาวิเคราะห์และออกแบบระบบ เพื่อให้ระบบเป็นไปตามความต้องการของผู้ใช้งาน มีประสิทธิภาพ ลดความซ้ำซ้อน ข้อมูลปราศจากความขัดแย้งกันและกัน

การวิเคราะห์สามารถแบ่งออกเป็นขั้นตอนดังนี้

#### 3.4.1 การวิเคราะห์ขั้นตอนและวิธีการในการดำเนินการ

- 1) ศึกษาและวิเคราะห์ข้อมูลทางด้านการจัดการข้อมูลที่เกิดขึ้นกับระบบ โดยรวบรวมข้อมูลการทำงานของระบบเดิม และสำรวจความต้องการของผู้ใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) วิเคราะห์ถึงปัญหาที่มีในระบบเดิม รวมทั้งของกำหนดต่าง ๆ และวิเคราะห์ถึงโครงสร้างของข้อมูลที่ได้ทำการรวบรวมมา เพื่อเป็นแนวทางในการออกแบบระบบที่จะพัฒนาขึ้นมาใหม่
- 3) ออกแบบและพัฒนาระบบใหม่ โดยนำข้อมูลที่ได้จากการวิเคราะห์มาทำการออกแบบระบบให้ได้ตรงตามความต้องการ และออกแบบให้ข้อมูลที่รวบรวมมานั้นมีความสัมพันธ์กันอย่างถูกต้อง

#### 3.4.2 การพัฒนาระบบสารสนเทศเพื่อจัดการข้อมูลสินค้าคงคลังแบบฝากขาย

เมื่อได้ทำการวิเคราะห์ปัญหาและออกแบบระบบจัดการข้อมูลสินค้าคงคลังแบบฝากขายแล้ว จะดำเนินการพัฒนาโปรแกรมต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การวิเคราะห์และออกแบบระบบงาน

#### 4.1 การดำเนินการในปัจจุบัน

เนื่องมาจากการดำเนินงานของบริษัท Football Thai Factory Sporting Goods Co., Ltd. ซึ่งเป็นการผลิตและจัดจำหน่ายสินค้าภายใต้ตรา FBT โดยมีโรงงานเป็นผู้ผลิตป้อนสินค้าเพื่อจำหน่าย โดยการจัดจำหน่ายจะกระทำได้หลายรูปแบบ การจัดจำหน่ายโดยการนำสินค้าของบริษัทไปฝากขายตามห้างสรรพสินค้าในกรุงเทพ ก็เป็นรูปแบบหนึ่ง ซึ่งร้านค้าที่อยู่ตามห้างต่างๆ จะเรียกว่า สาขา โดยที่ทรัพย์สินและพนักงานทั้งหมดในสาขายังคงอยู่ภายใต้การดำเนินงานของบริษัท Football Thai Factory Sporting Goods Co., Ltd. ซึ่งบริษัทจะมีหน่วยงานที่เป็นฝ่ายที่ควบคุมดูแลสาขาจากสำนักงานใหญ่ โดยการทำงานจะดูแลผ่านทางพนักงานของบริษัทที่อยู่แต่ละสาขาอีกทีหนึ่ง โดยที่พนักงานจะเป็นจัดจำหน่าย โดยมีการควบคุมสินค้าคงคลังที่สาขาเองด้วย และการทำงานต่างๆที่สาขาจะมีการควบคุม โดยเอกสารต่างๆ เช่น รายงานยอดขาย, การสั่งสินค้า, ใบรายงานยอด Stock ฯลฯ จากเอกสารต่างๆที่เกิดขึ้นจากการดำเนินงานของแต่ละสาขา จะถูกนำมาบันทึกข้อมูลที่สำนักงานใหญ่ เพื่อที่จะสามารถควบคุมดูแลความเคลื่อนไหวของสินค้าทั้งหมด

จากการทำงานมีส่วนที่เกี่ยวข้องกับข้อมูลหลายๆส่วนทั้งในส่วนที่เกี่ยวข้องกับ การสั่งสินค้าจากทางสาขา, การส่งมอบสินค้าจากทางโรงงาน, การบริหารสินค้าคงคลังที่แต่ละสาขา รวมทั้งการบันทึกข้อมูลที่สำนักงานใหญ่ ซึ่งจากการทำงานในปัจจุบันที่มีการทำงานเป็นระบบอยู่แล้วนั้น ยังไม่สามารถที่จะควบคุมการทำงานดังกล่าวให้มีประสิทธิภาพได้ เพื่อผู้บริหารจะได้นำข้อมูลไปใช้ในการบริหารงานต่างๆ ได้ตามสมควร ทางบริษัทจึงมีความประสงค์ที่จะพัฒนาระบบการจัดการดำเนินงานในปัจจุบัน ให้มีความเป็นระบบมากยิ่งขึ้น เพื่อที่จะสามารถขจัดปัญหาที่เกิดขึ้น หรืออาจจะเกิดขึ้นในอนาคต ซึ่งจุดที่เป็นหัวใจสำคัญของการดำเนินการดังกล่าวมานี้ จะเกี่ยวข้องกับการสั่งสินค้า และ จัดส่งสินค้านี้ระหว่างโรงงานกับสาขา

จากการทำงานของระบบงานดังกล่าวจะเป็นการดำเนินงานที่เกี่ยวข้องกันระหว่างหน่วยงานดังนี้

1. สำนักงานใหญ่
2. สาขาต่างๆ
3. โรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากความสัมพันธ์ของทั้ง 3 หน่วยงานดังกล่าวข้างต้น จะมีการดำเนินงานที่เกิดขึ้นในส่วนต่างๆซึ่งสามารถจำแนกตามหน่วยงานต่างๆ ได้ดังนี้

1. สำนักงานใหญ่
  - 1.1 การอนุมัติ
  - 1.2 การบันทึก
  - 1.3 การตรวจสอบ
2. สาขาต่างๆ
  - 2.1 การขาย
  - 2.2 การบันทึกข้อมูล
  - 2.3 การตรวจนับ
  - 2.4 การตั้งของ
  - 2.5 การจัดโปรโมชั่น
3. โรงงาน
  - 3.1 การจัดสินค้า
  - 3.2 การส่งสินค้า
  - 3.3 การตรวจนับ
  - 3.4 การจัดทำเอกสาร

#### **ระบบปัจจุบัน (Existing System)**

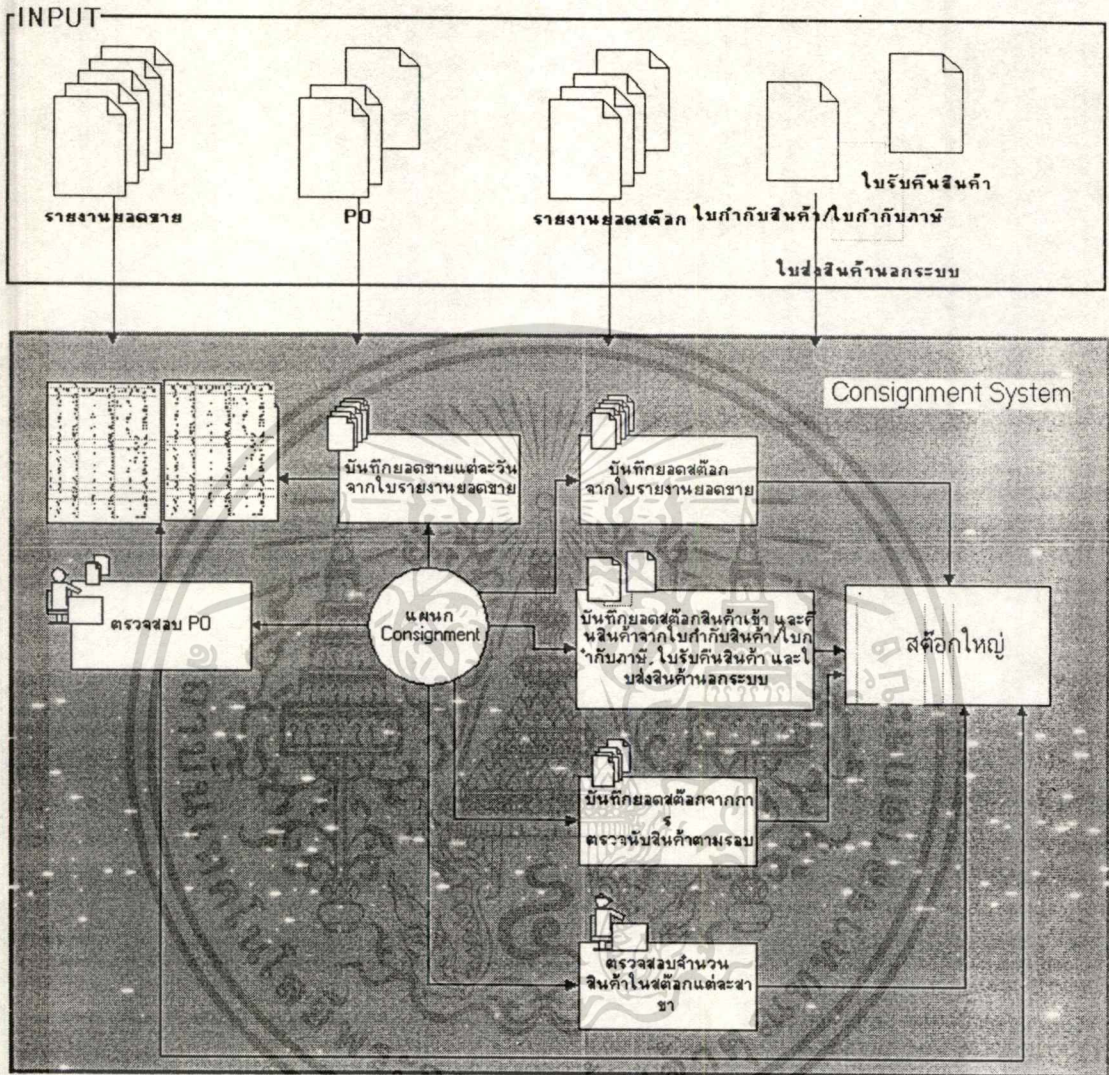
#### **ระบบการทำงานในปัจจุบัน (Existing Business System)**

ระบบ Consignment เป็นส่วนหนึ่งของกระบวนการทำงานทั้งหมดของการดำเนินงานของบริษัท FBT Sport Complex Co., Ltd. เพราะการทำงานของบริษัท FBT Sport Complex Co., Ltd. จะเกี่ยวข้องกับการกระจายสินค้า ไปยังสาขาต่างๆที่มีอยู่ ดังนั้นการที่จะสามารถควบคุม และ ทราบถึงความต้องการที่เหมาะสมของสินค้าจากแต่ละสาขาได้ จำเป็นต้องมีวิธีการที่ใช้ในการตรวจสอบ และติดตาม เพื่อให้มีการจัดสรรการใช้ทรัพยากรอย่างคุ้มค่าที่สุด ซึ่งวิธีการทำงานดังกล่าวเรียกว่า ระบบ Consignment ซึ่งปัจจุบันนี้ก็ได้อาศัย Concept หรือ วิธีการของระบบ Consignment อยู่แล้ว เพียงแต่ต้องการนำเทคโนโลยีสารสนเทศเข้ามาปรับใช้กับการดำเนินงานดังกล่าวให้มีประสิทธิภาพมากยิ่งขึ้น และ สอดคล้องกับวัตถุประสงค์ของโครงการ

สามารถสรุปการดำเนินการของระบบ Consignment Inventory ที่เป็นอยู่ในปัจจุบัน โดย นำเสนอด้วยรูปภาพ เพื่อให้เห็นถึงการทำงานที่ชัดเจนยิ่งขึ้นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

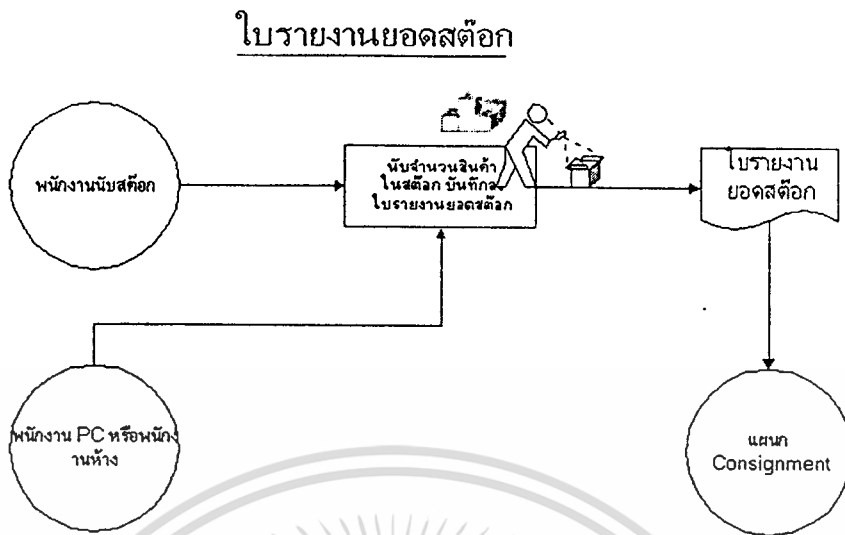
### ระบบงาน Consignment



รูปที่ 4.1 แสดงระบบงานปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 4.4 แสดงขั้นตอนการตรวจนับสต็อก

จากระบบงานปัจจุบันจะพบว่าขั้นตอนการทำงานทั้งหมดจะกระทำโดยเอกสารทั้งหมด โดยที่ยังไม่มีการนำระบบคอมพิวเตอร์เข้ามาช่วย ซึ่งจากการทำการศึกษาจะได้ระบบงานที่พอจะนำคอมพิวเตอร์เข้ามาช่วยได้หลายจุดด้วยกัน เช่น จุดขายแต่ละจุด (ประมาณ 50 จุดขาย) ควรมีระบบการควบคุมสินค้าคงคลังและระบบ POINT OF SALES เข้ามาช่วย และการรวบรวมข้อมูลส่วนกลางที่สำนักงานใหญ่ก็ควรจะนำระบบการควบคุมสินค้าคงคลังเข้ามาใช้เช่นกัน

แต่จากแนวคิดดังกล่าวมีความเป็นไปได้ น้อยมาก เพราะสาเหตุดังนี้

- การมีอุปกรณ์คอมพิวเตอร์และอุปกรณ์การสื่อสาร (สายโทรศัพท์และMODEM) ในห้างสรรพสินค้าที่ FBT มีจุดขายอยู่นั้นทำได้ยากและ/หรือมีเฉพาะบางแห่งเท่านั้นที่กระทำได้
- งบประมาณของโครงการจะต้องใช้งบประมาณจำนวนมากในการกระทำความดังกล่าว ดังนั้นแนวทางของระบบใหม่ที่จะนำคอมพิวเตอร์มาช่วยก็คือ

#### ระบบที่ต้องการ (Proposed System)

##### ข้อกำหนดของระบบที่ออกแบบ (System Specification)

ในการพัฒนาระบบ Consignment Inventory เพื่อแก้ปัญหาที่เกิดขึ้นในการจัดการเดิมที่มีอยู่ โดยการนำเทคโนโลยีสารสนเทศเข้ามาประยุกต์ใช้งานนั้น จำเป็นต้องมีการระบุถึงความสามารถ หรือ สิ่งที่ระบบใหม่จะสามารถตอบสนองได้ เพื่อลดปัญหาที่เกิดขึ้นในระบบเดิมให้ได้มากที่สุด โดยคำนึงถึงปัจจัยต่างๆ ในการพัฒนาระบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เวลา
2. งบประมาณ
3. ความพร้อมของบุคคลกร
4. วิธีการที่สามารถแก้ปัญหาได้จริง

ซึ่งเราได้สรุปเป็นข้อกำหนดของระบบที่จะออกแบบดังนี้

1. ระบบต้องสามารถแสดงความเคลื่อนไหวของ Stock ที่สาขา ณ.เวลาใดเวลาหนึ่งได้ โดยการนำข้อมูล ไปใช้ที่สำนักงานใหญ่ เกือบจะในทันที (Real Time) เช่น วันต่อวัน
2. ระบบต้องสามารถตรวจสอบยอดขายของแต่ละสาขาได้ และสามารถที่จะนำข้อมูลไปใช้ที่สำนักงานใหญ่ เกือบจะในทันที (Real Time) เช่น วันต่อวัน
3. ระบบจะต้องสามารถให้ข้อมูล หรือ รายงาน หรือ กราฟที่แสดงความสัมพันธ์ในรูปแบบต่างๆ ที่ผู้บริหารสามารถนำไปใช้ในการตัดสินใจ และบริหารงานได้อย่างมีประสิทธิภาพ
4. ระบบจะต้องกำหนดวิธีการที่แน่นอนในการส่งข้อมูลระหว่าง สาขาและสำนักงานใหญ่ รวมทั้งวิธีการสำรองในกรณีที วิธีแรกเกิดปัญหาเฉพาะหน้า
5. ระบบจะต้องสามารถลดการทำงานที่ดำเนินการอยู่ในปัจจุบันลงได้ เพื่อขจัดปัญหาความซ้ำซ้อนของการดำเนินงาน
6. ระบบจะต้องสามารถ จัดการข้อมูลของการ จัดโปร โมชั่นสินค้าได้อย่างมีประสิทธิภาพ และสามารถตรวจสอบได้

การทำงานของระบบ Consignment ประกอบด้วยระบบย่อย 2 ระบบดังนี้

1. ระบบสินค้าคงคลัง (Inventory Control: IC)
2. ระบบส่งเสริมการขาย (Promotion: PR)

#### ระบบสินค้าคงคลัง (Inventory Control: IC)

ในระบบสินค้าคงคลังจะประกอบด้วย

- การรับ (Received)
- การขาย (Sale)
- การปรับปรุงยอดสต็อก (Adjustment Stock on hand)
- สินค้าคงเหลือ (Stock On hand)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### กระบวนการรับสินค้าจะมีการทำงานดังนี้

การสั่งของจากทางห้าง โดยพนักงาน PC หรือ ผ่านทาง DC หรือ Distribution Center

- พนักงาน PC จะสั่งของมาที่สำนักงานใหญ่โดยตรง หรือ ผ่านทาง DC ซึ่งจะมีเอกสารที่เป็นแบบฟอร์มการสั่งของมาตรฐาน (เป็นเอกสารของทาง FBT ที่มาขยายเพิ่มเติม)
- ที่สำนักงานใหญ่จะบันทึกเข้าสู่ระบบ (Key In) โดยหน้าจอจะใกล้เคียง หรือ เหมือนกันกับแบบฟอร์มที่รับเข้ามาเพื่อเป็นรูปแบบ และ วิธีการมาตรฐานในการบันทึกข้อมูล (ข้อมูลได้รับการบันทึกเข้าสู่ระบบฐานข้อมูล) คือ Key ไม่ต้องยุ่งยาก
- ระบบจะออกรายงานสรุปการสั่งสินค้า เพื่อการพิจารณาอนุมัติ (จะมีโปรแกรมที่ตรวจสอบว่า ควรมีการสั่งซื้อหรือไม่)
- จะส่งรายงานการสั่งสินค้าที่ได้รับการอนุมัติแล้วไปยังโรงงาน โดยวิธี
  - Document by Messenger
  - Fax (Document)
- ทางโรงงานจัดเตรียมสินค้าตามขั้นตอน และ ออกใบส่งสินค้า และส่งสินค้าไปยัง Outlet ต่างๆ เช่น ห้าง หรือ DC (Distribution Center)
- พนักงาน PC จะตรวจนับและกรอกแบบฟอร์มการรับสินค้าซึ่งเป็นแบบฟอร์มมาตรฐาน และส่งไปยังสำนักงานใหญ่ โดยการนับสินค้าอาจจะครบ หรือ ไม่ก็ได้ โดย
  - ถ้าไม่ครบจะสามารถบันทึกหมายเหตุได้ (บันทึกหรือไม่ก็ได้)
  - ถ้าสินค้าไม่ถูกต้อง ก็สามารถทำการคืนสินค้าได้เช่นกัน โดยการกรอกแบบฟอร์ม
  - ถ้าสินค้าเกิน ก็สามารถทำใบขอซื้อเพิ่ม เข้ามา หรือ ทำการคืนก็ได้
- สำนักงานใหญ่ บันทึกข้อมูลจากเอกสารที่ได้รับจาก Outlet ต่างๆ เข้าสู่ระบบ ซึ่งวิธีการส่งเอกสารจะ อธิบายเพิ่มเติมต่อไป
- สิ้นสุดกระบวนการรับเข้า

### กระบวนการขายสินค้าจะมีการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อพนักงาน PC ขายสินค้า ก็จะมีการบันทึกลงในเอกสารบันทึกการขาย (ซึ่งอาจจะเป็น Stock Card ด้วยเลขก็ได้) ซึ่งทาง FBT มีอยู่แล้ว หรือ เราสามารถเพิ่มเติมได้
- เมื่อมีการปิดสิ้นวัน และ ถึงงวดที่ต้องตรวจนับจริง ก็จะมีการบันทึกลงในแบบฟอร์มมาตรฐาน เช่นกัน
- และแบบฟอร์มจะถูกส่งเข้ามาที่ Head Office พร้อมทั้ง Diskette (กรณีใช้คอมพิวเตอร์ที่สาขา) และ/หรือ Bar Code เพื่อทำการบันทึกเข้าสู่ระบบที่สำนักงานใหญ่

#### กระบวนการปรับปรุงยอดมีการทำงานดังนี้

- เช่น กรณีการคืนสินค้า ก็จะสามารถปรับปรุงรายการได้ และ มีการบันทึกหมายเหตุได้
- หรือ กรณีที่มีการเสียหายหรือ การเสื่อมสินค้าของสินค้า ก็สามารถทำได้เช่นเดียวกัน
- หรือมีการ รับโอน(โอนระหว่างสาขา โดยการ โอนโดยตรง หรือ โอนการเข้า H/O ตามแต่ Business Logic) หรือ โอนจ่าย จะมีการทำคืนสินค้า เข้าโรงงาน โดยเอกสารจะไปที่ Head Office และ โรงงานด้วย

#### กระบวนการตรวจสอบสินค้าคงคลัง

- เมื่อมีการปิดสิ้นวัน และ ถึงงวดที่ต้องตรวจนับจริง ก็จะมีการบันทึกลงในแบบฟอร์มมาตรฐาน เช่นกัน
- และแบบฟอร์มจะถูกส่งเข้ามาที่ Head Office เพื่อทำการบันทึก

### การติดต่อสื่อสาร ระหว่างหน่วยงานต่างๆ ภายในระบบ Consignment Inventory System

#### 1. การติดต่อระหว่าง โรงงานกับ สำนักงานใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Document By Messenger
  - Electronic Document (Fax, Email)
2. การติดต่อระหว่าง สาขา หรือ DC กับ สำนักงานใหญ่
- Document by Mail (Post Office)

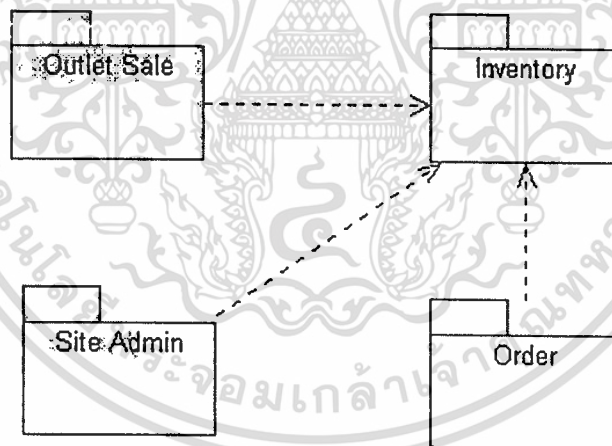
#### 4.2 การวิเคราะห์และออกแบบระบบ

มีขั้นตอนดังนี้

##### 4.2.1 ขั้นตอนการวิเคราะห์ความต้องการ (Requirement Analysis )

โดยขั้นตอนนี้จะเป็นการศึกษาถึงความต้องการของระบบที่จะพัฒนาขึ้นว่าต้องเกี่ยวข้องกับสิ่งใดบ้าง โดยจะเป็นการรวบรวมข้อมูลต่างๆ และนำเสนอด้วย Use Case Diagram เพื่อเป็นการแสดงแนวคิดของผู้วิเคราะห์ว่าตรงและถูกต้องต่อความต้องการหรือไม่ อย่างไร

Use Case Diagram ในรูปที่ 4.5 แสดงให้เห็นถึงสิ่งที่เกี่ยวข้องกับระบบและกิจกรรมกระบวนการต่างๆ ของระบบ



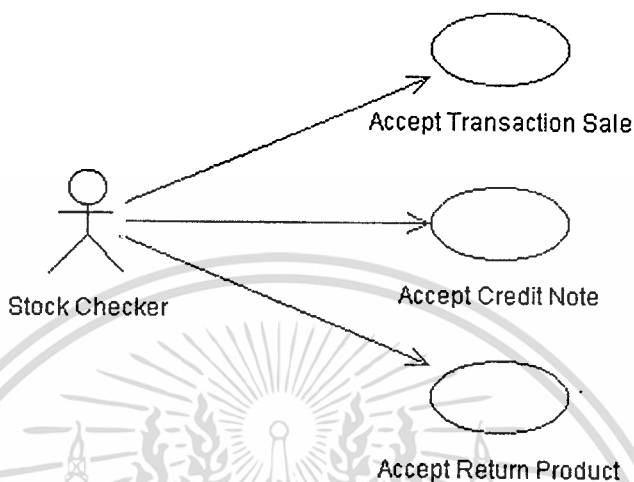
รูปที่ 4.5 แสดงภาพรวมของระบบในรูปแบบ PACKAGE

- Package Outlet Sale จะรวบรวม Use Case Diagram ของกิจกรรมการทำงานของระบบ ณ.จุดขาย
- Package Inventory จะรวบรวม Use Case Diagram ของกิจกรรมการทำงานของ การควบคุมสต็อกจากศูนย์กลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Package Order จะรวบรวม Use Case Diagram ของกิจกรรมการทำงานของกรรับสินค้าเข้าสู่สต็อก

รายละเอียดของ Use Case Diagram ต่างๆมีดังนี้

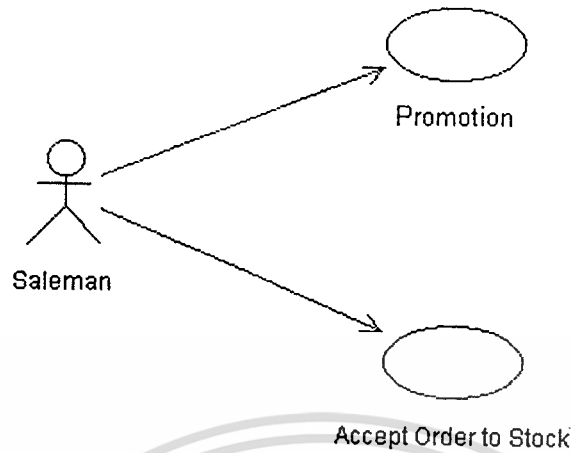


รูปที่ 4.6 แสดง USE CASE DIAGRAM ของ PACKAGE OUTLET SALES

Use Case Accept Transaction Sale – เป็นการนำรายงานยอดขายประจำวันมา Key in เข้าสู่ระบบ โดยพนักงานแผนกสต็อก

Use Case Accept Credit Note – เป็นการนำรายงานการคืนสินค้ากลับไปยังโรงงานมา Key in เข้าสู่ระบบ โดยพนักงานแผนกสต็อก

Use Case Accept Return Products – เป็นการนำรายงานการคืนสินค้าจากลูกค้าที่ซื้อแล้วต้องการเปลี่ยนมา Key in เข้าสู่ระบบ โดยพนักงานแผนกสต็อก



รูปที่ 4.7 แสดง USE CASE DIAGRAM ของ PACKAGE ORDER

Use Case Promotion – เป็นการตั้งโปร โมชั่น ซึ่งจะต้องนำรายการมา Key in เข้าสู่ระบบ โดยพนักงานแผนกขาย

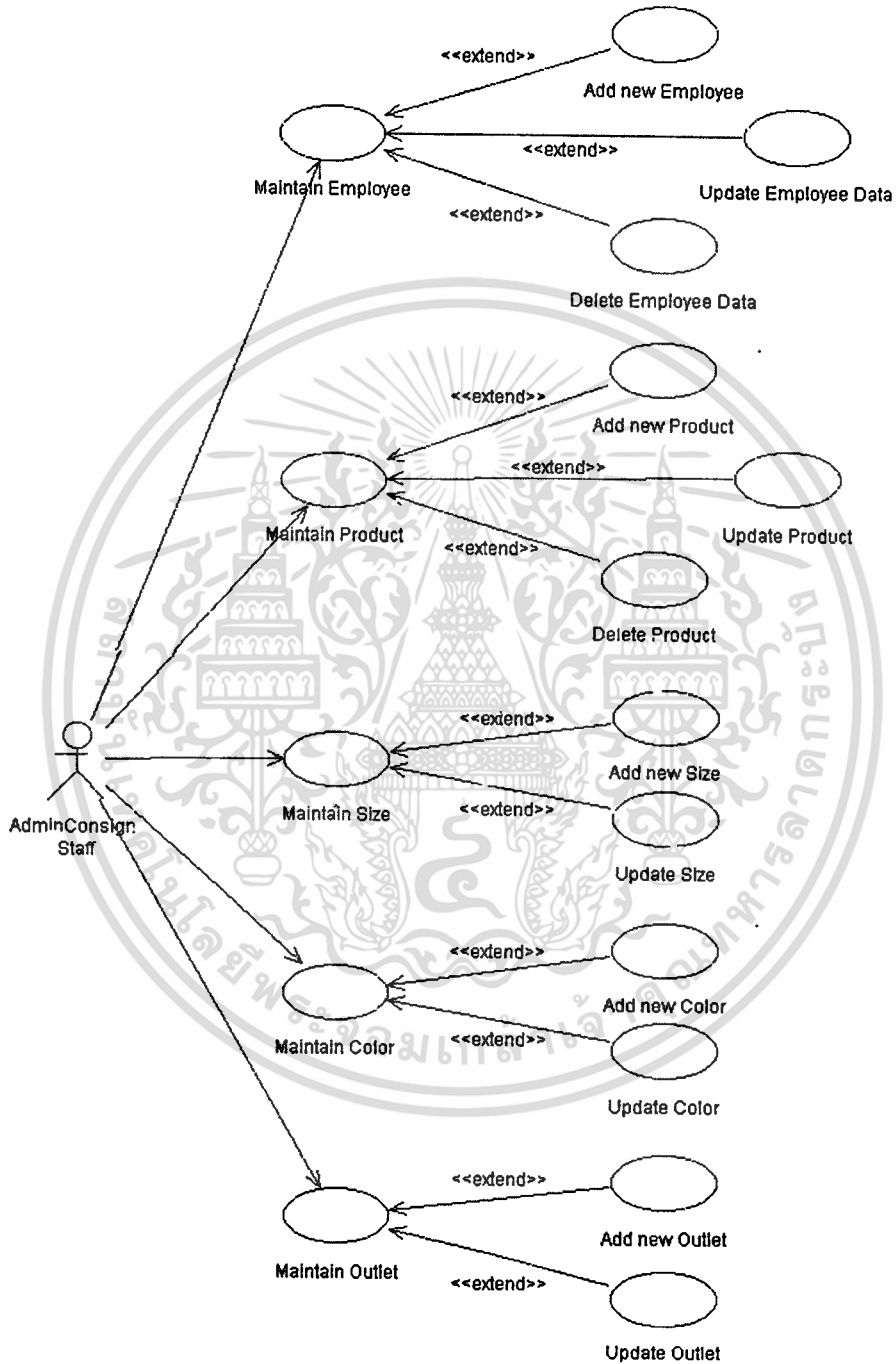
Use Case Accept Order to Stock – เป็นการนำเอกสารใบกำกับภาษีมา Key in เข้าสู่ระบบ เพื่อรับสินค้าเข้าสู่สต็อกโดยพนักงานแผนกขาย



รูปที่ 4.8 แสดง USE CASE DIAGRAM ของ PACKAGE INVENTORY

Use Case Check Stock – เป็นการตรวจนับยอดคงเหลือสินค้าตามรอบการตรวจนับ โดยการนำรายงานการตรวจนับมา Key in เข้าสู่ระบบ โดยพนักงานแผนกสต็อก เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Case Adjust Stock – เป็นการแก้ไขยอดคงเหลือสินค้า โดยการแก้ไขจะต้องมีเอกสารอ้างอิงในการเปลี่ยนแปลงมา Key in เข้าสู่ระบบ โดยพนักงานแผนกสต็อก



รูปที่ 4.9 แสดง USE CASE DIAGRAM ของ PACKAGE SITE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Case Maintain Employee – เป็นการจัดการข้อมูลพื้นฐานของพนักงาน

Use Case Product – เป็นการจัดการข้อมูลพื้นฐานของสินค้า

Use Case Color – เป็นการจัดการข้อมูลพื้นฐานของสีของสินค้า

Use Case Size – เป็นการจัดการข้อมูลพื้นฐานของขนาดของสินค้า

Use Case Outlet – เป็นการจัดการข้อมูลพื้นฐานของจุดขาย

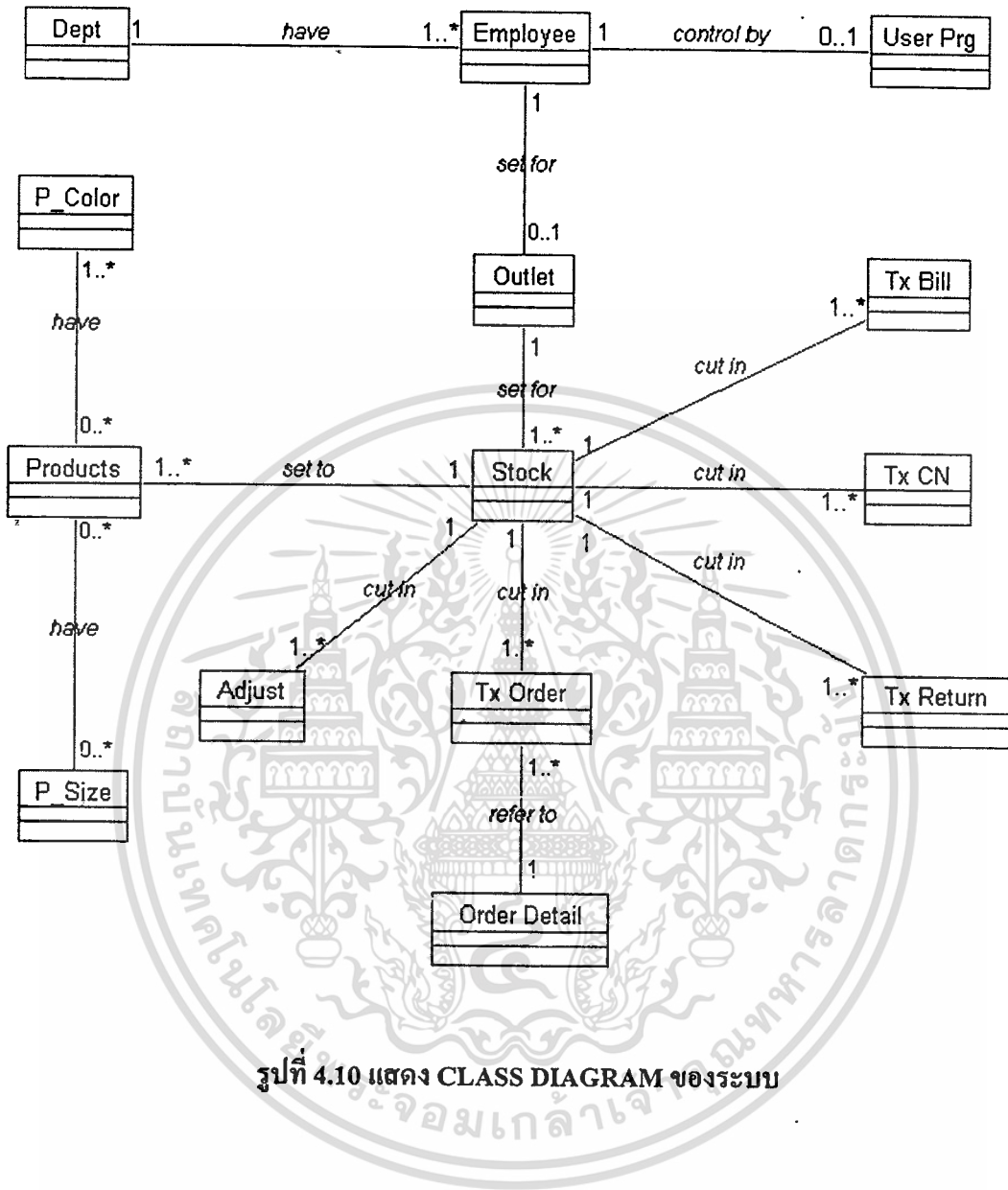
#### 4.2.2 ขั้นตอนการวิเคราะห์และออกแบบระบบ

หลังจากได้ศึกษาและรวบรวมข้อมูลการทำงานภายในระบบแล้ว ขั้นตอนการวิเคราะห์ระบบนี้จะเป็นการนำเอาข้อมูลที่ได้มีการกำหนดว่าระบบจะพัฒนาระบบอย่างไรตาม Use Case Diagram ในหัวข้อที่แล้ว มาทำการวิเคราะห์โดยระบุ Class ที่จะต้องมีในระบบ ซึ่งจะรวมถึงคุณสมบัติและพฤติกรรมของ Class ด้วย จะนำเสนอในรูปแบบของ Class Diagram

##### 1. Class Diagram

จาก Use Case Diagram ที่ได้จากการวิเคราะห์ความต้องการของระบบ เราสามารถวิเคราะห์ Class ต่างๆ ได้ดังนี้

- Class Products – เป็น Class ที่เก็บข้อมูลของสินค้าต่างๆ
- Class Size – เป็น Class ที่เก็บข้อมูลของขนาดของเสื้อผ้า
- Class Color – เป็น Class ที่เก็บข้อมูลของสีของสินค้า
- Class Outlet – เป็น Class ที่เก็บข้อมูลของจุดขาย(สาขา)
- Class Employee – เป็น Class ที่เก็บข้อมูลของพนักงาน
- Class Stock – เป็น Class ที่เก็บข้อมูลของยอดสินค้าที่อยู่ในสต็อก
- Class Bill Transaction – เป็น Class ที่เก็บข้อมูลของยอดขายของแต่ละจุดขาย
- Class Credit Note – เป็น Class ที่เก็บข้อมูลของการลดหนี้(ทำสินค้าคืน โรงงาน)ของแต่ละจุดขาย
- Class Return Product – เป็น Class ที่เก็บข้อมูลของการรับคืนสินค้าจากลูกค้า
- Class User Program – เป็น Class ที่เก็บข้อมูลของการกำหนดระดับการใช้งานของผู้ใช้ระบบ

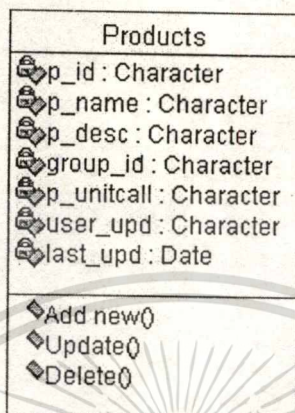


รูปที่ 4.10 แสดง CLASS DIAGRAM ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

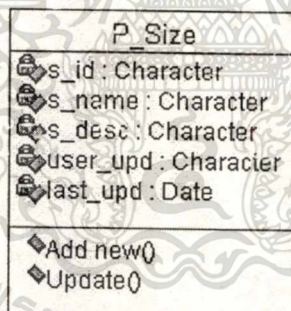
โดยที่คุณสมบัติของแต่ละ Class มีดังนี้

1. Class Products เป็นการแสดงรายละเอียดของสินค้าโดยมีรายละเอียด คือ รหัสสินค้า, ชื่อสินค้า, รหัสกลุ่ม, หน่วยเรียก เป็นต้น



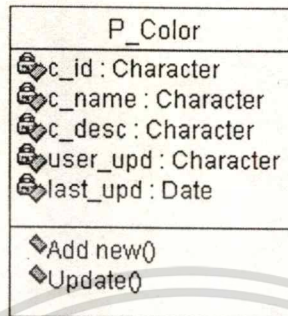
รูปที่ 4.11 แสดงองค์ประกอบของ CLASS PRODUCTS

2. Class P\_Size เป็น Class ที่เก็บรายละเอียดขนาดสินค้า เช่น Size S,M,L,XL เป็นต้น



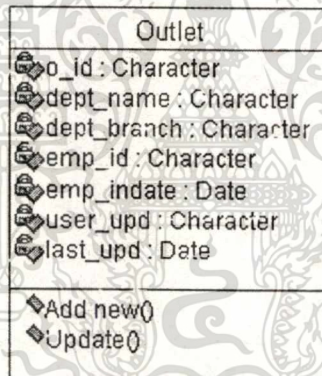
รูปที่ 4.12 แสดงองค์ประกอบของ CLASS P\_SIZE

3. Class P\_Color เป็น Class ที่เก็บข้อมูลสีของสินค้า มี Attribute เช่น c\_id - รหัสสี, c\_name - ชื่อสี, c\_desc - รายละเอียดอื่น



รูปที่ 4.13 แสดงองค์ประกอบของ CLASS P\_COLOR

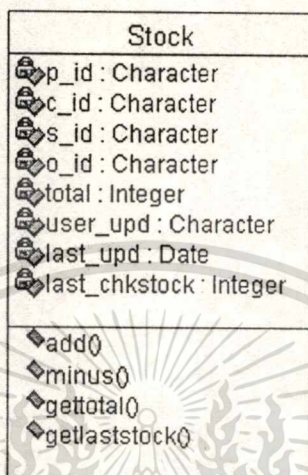
4. Class Outlet เป็น Class ที่จัดการเก็บข้อมูลของจุดขายและพนักงานที่ขาย ณ จุดขาย นั้นๆ



รูปที่ 4.14 แสดงองค์ประกอบของ CLASS OUTLET

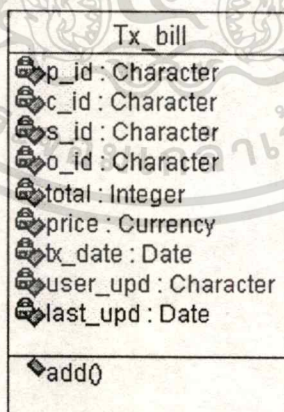
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Class Stock เป็น Class ที่จัดเก็บข้อมูลเกี่ยวกับสต็อกสินค้าของแต่ละสาขา โดยมีคุณสมบัติ คือ เก็บรหัสสินค้า, รหัสสี, รหัสขนาด, รหัสสาขา, จำนวนคงเหลือ และจำนวนคงเหลือจากการตรวจสอบสต็อกครั้งล่าสุด ส่วน Method จะมี การเพิ่ม,ลด จำนวนสินค้า เป็นต้น



รูปที่ 4.15 แสดงองค์ประกอบของ CLASS STOCK

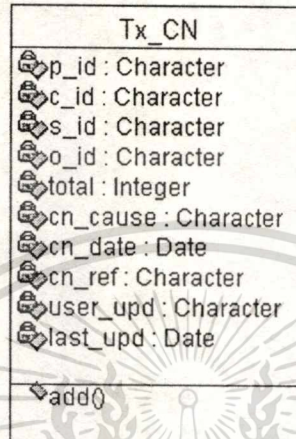
6. Class Tx\_Bill เป็น Class ที่เก็บรายการยอดขายของแต่ละวันแต่ละสาขา ซึ่งมีรายละเอียดคือ เก็บรหัสสินค้า, รหัสสี, รหัสขนาด, รหัสสาขา, จำนวน, ราคา และวันที่ขาย



รูปที่ 4.16 แสดงองค์ประกอบของ CLASS TX\_BILL

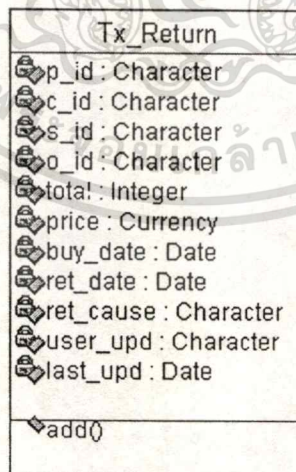
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. Class Tx\_CN เป็น Class ที่เก็บรายการทำคืนสินค้ากลับโรงงานของแต่ละวันแต่ละสาขา ซึ่งมีรายละเอียดคือ เก็บรหัสสินค้า, รหัสสี, รหัสขนาด, รหัสสาขา, จำนวน, เหตุการณ์คืนสินค้า, เอกสารอ้างอิง และวันที่ทำคืนสินค้า



รูปที่ 4.17 แสดงองค์ประกอบของ CLASS TX\_CN.

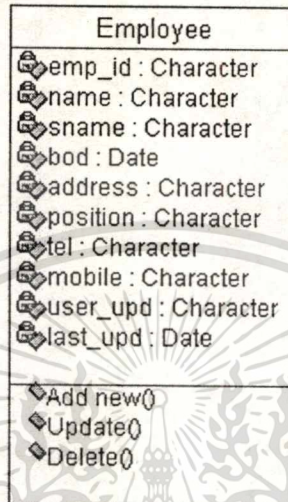
8. Class Tx\_Return เป็น Class ที่เก็บรายการรับคืนสินค้าจากลูกค้าแต่ละวันแต่ละสาขา ซึ่งมีรายละเอียดคือ เก็บรหัสสินค้า, รหัสสี, รหัสขนาด, รหัสสาขา, จำนวน, ราคา, เหตุการณ์คืนสินค้า, เอกสารอ้างอิง, วันที่ซื้อ และวันที่คืนสินค้า



รูปที่ 4.18 แสดงองค์ประกอบของ CLASS TX\_RETURN

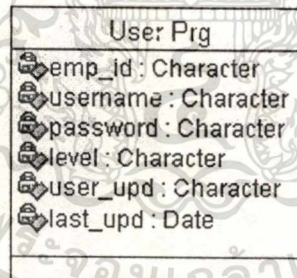
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Class Employee เป็น Class ที่เก็บข้อมูลพนักงาน มีรายละเอียดคือ เก็บรหัสพนักงาน,ชื่อ,สกุล,วันเกิด,ที่อยู่,ตำแหน่ง,โทรศัพท์ และมือถือ



รูปที่ 4.19 แสดงองค์ประกอบของ CLASS EMPLOYEE

10. Class User\_prg เป็น Class ที่เก็บข้อมูลการกำหนดสิทธิให้ผู้ใช้งาน



รูปที่ 4.20 แสดงองค์ประกอบของ CLASS USER\_PRG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. Class P\_Group เป็น Class ที่เก็บข้อมูลกลุ่มสินค้า

P_Group
group_id : Character group_name : Character group_desc : Character user_upd : Character last_upd : Date
Add new() Update()

รูปที่ 4.21 แสดงองค์ประกอบของ CLASS P\_GROUP



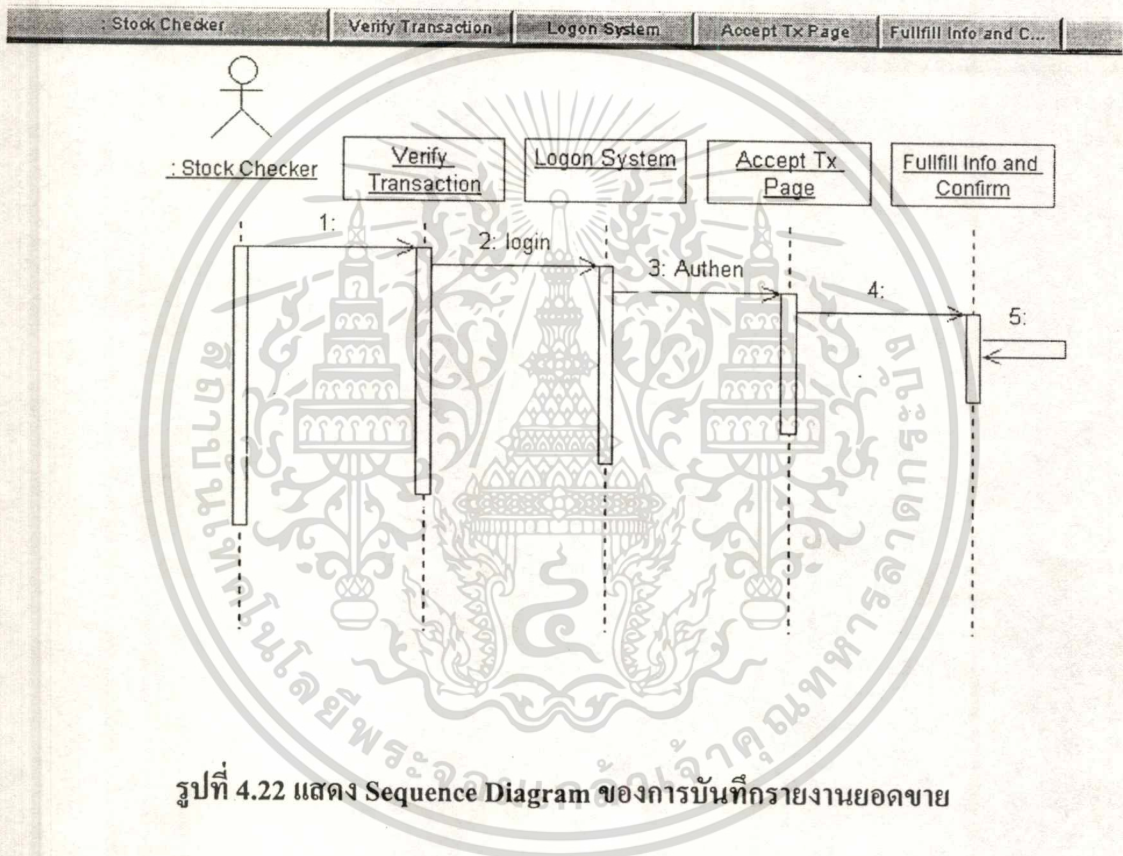
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. Sequence Diagram

จะแสดงถึงการส่ง Message เพื่อการติดต่อสื่อสารระหว่าง Object ต่างๆที่เกี่ยวข้องซึ่งจะนำเสนอในรูปแบบ Sequence Diagram โดยจะเป็นการแสดงถึงลำดับของช่วงเวลาที่แน่นอนของการส่ง Message

### 2.1 กระบวนการบันทึกที่รายงานยอดขาย

เมื่อรายงานยอดขายได้ถูกจัดส่งมาแล้ว ทางแผนก Consignment ก็จะจัดการแยกเอกสารแล้วจึงนำเอกสารรายงานบันทึกยอดขายมาบันทึกลงระบบ

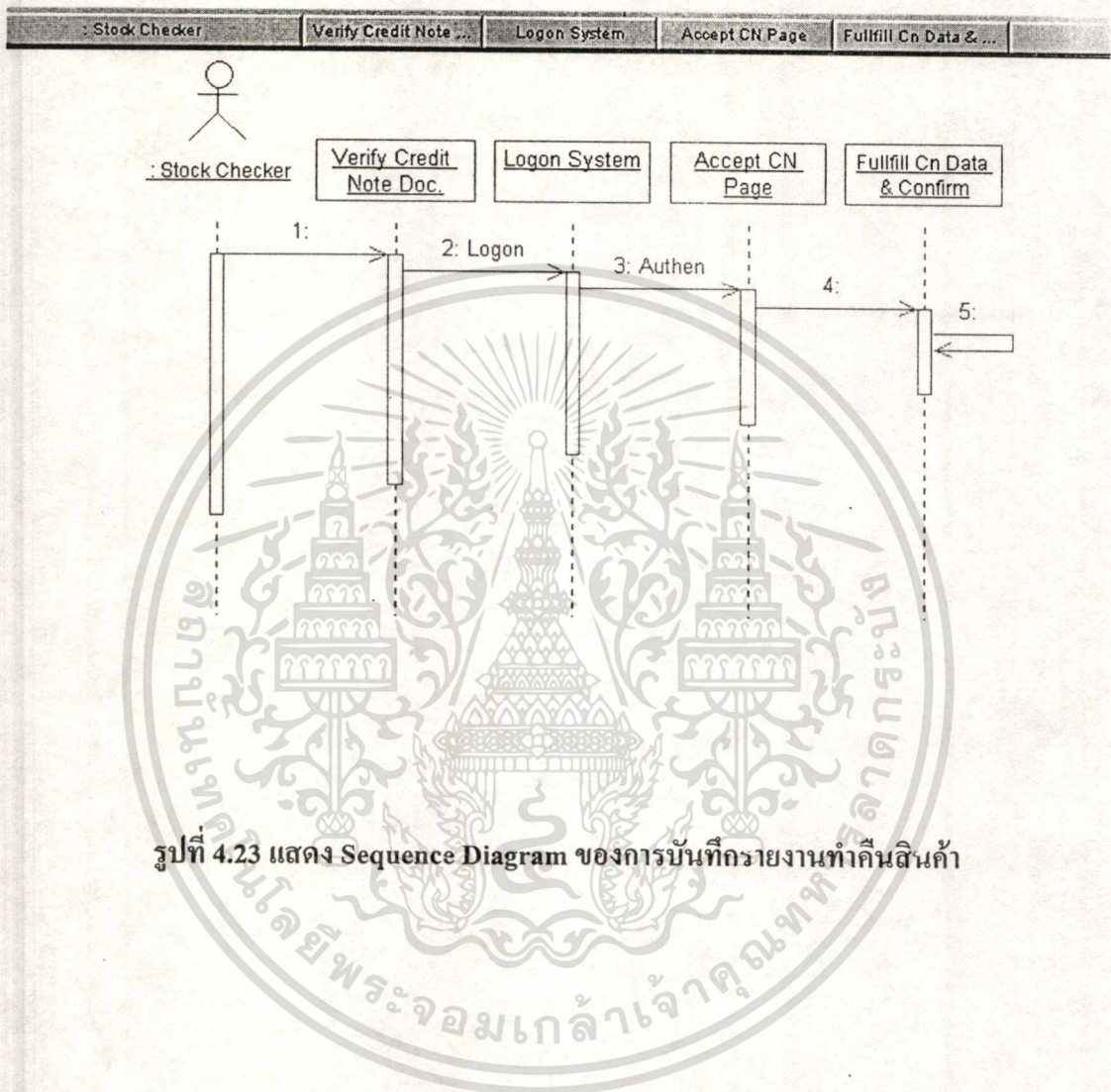


รูปที่ 4.22 แสดง Sequence Diagram ของการบันทึกที่รายงานยอดขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 กระบวนการบันทึกรายงานการทำคืนสินค้า

เมื่อรายงานยอดขายได้ถูกจัดส่งมาแล้ว ทางแผนก Consignment ก็จะจัดการแยกเอกสาร แล้วนำเอกสารทาง Credit Note (CN) มาทำการบันทึกเข้าระบบงาน

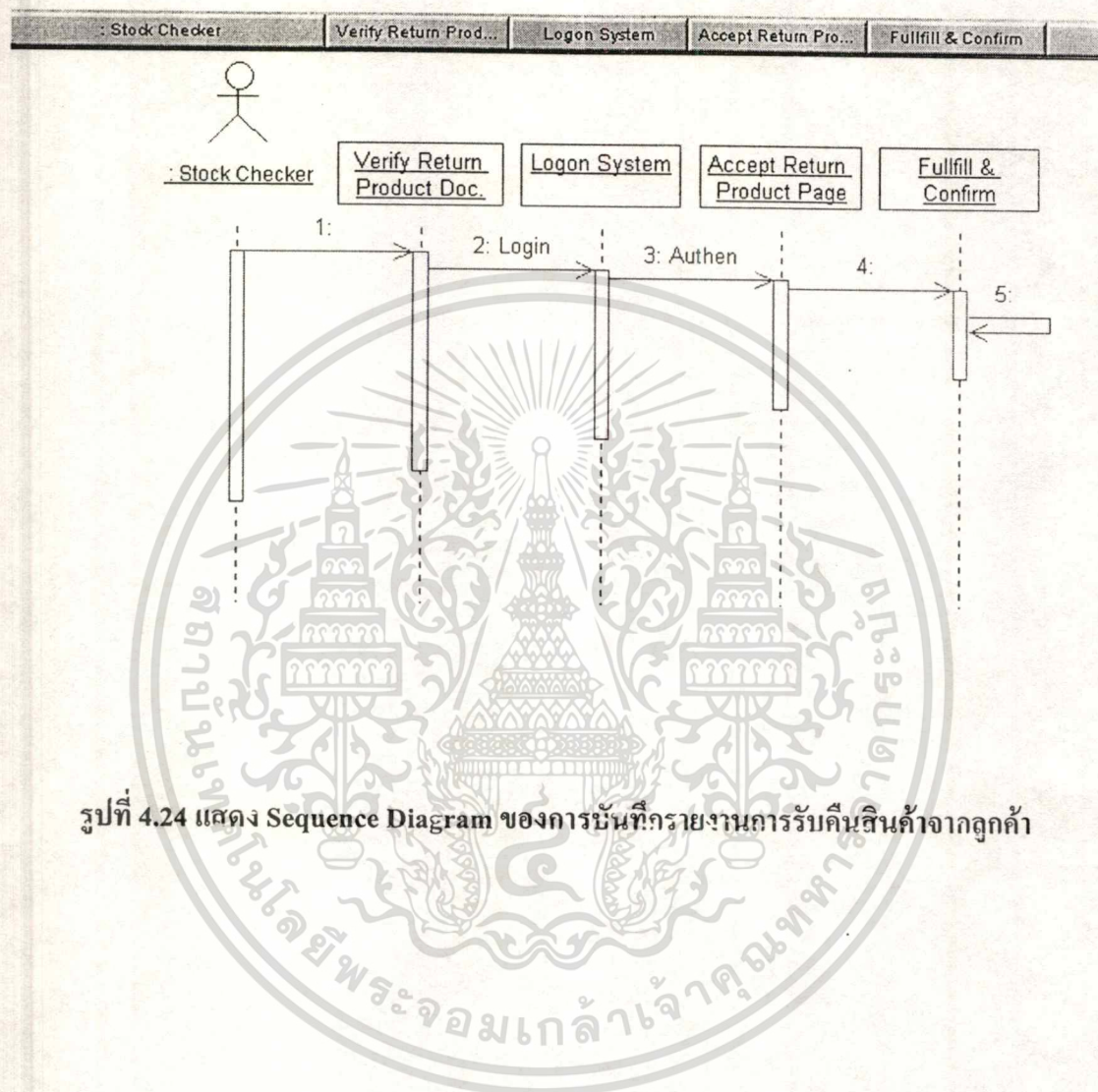


รูปที่ 4.23 แสดง Sequence Diagram ของการบันทึกรายงานทำคืนสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 กระบวนการบันทึกรายงานการคืนสินค้าจากลูกค้า

เมื่อรายงานยอดขายได้ถูกจัดส่งมาแล้ว ทางแผนก Consignment ก็จะจัดการแยกเอกสาร แล้วนำเอกสารการคืนสินค้าจากลูกค้า มาทำการบันทึกเข้าระบบงาน

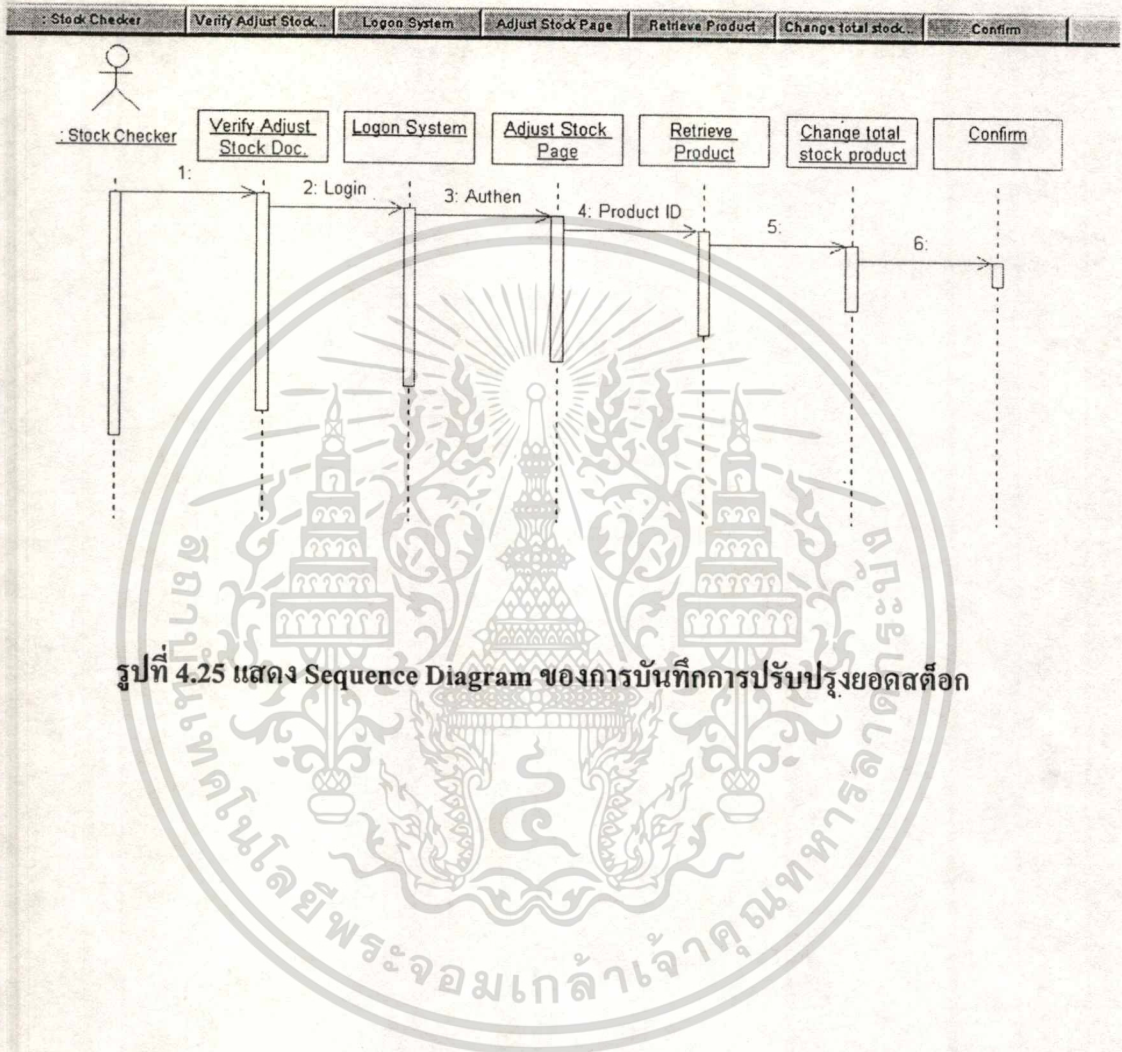


รูปที่ 4.24 แสดง Sequence Diagram ของการบันทึกรายงานการรับคืนสินค้าจากลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 กระบวนการบันทึกการปรับปรุงยอดสต็อก

เมื่อรายงานยอดขายได้ถูกจัดส่งมาแล้ว ทางแผนก Consignment ก็จะจัดการแยกเอกสาร แล้วนำเอกสารการปรับปรุงยอดสต็อกของสาขา พร้อมทั้งเหตุผลการปรับยอดสต็อก จากนั้นก็นำมาเข้าระบบงาน

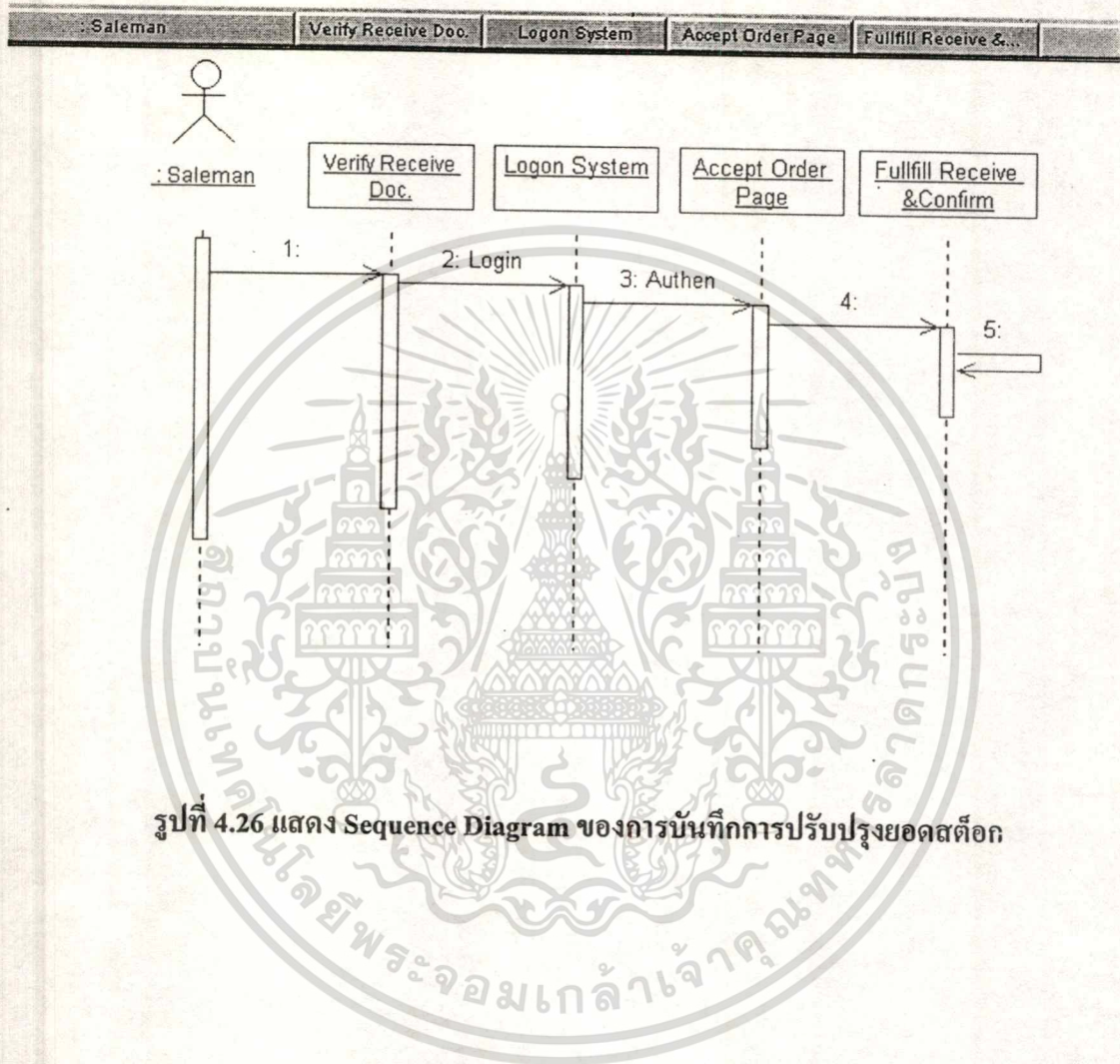


รูปที่ 4.25 แสดง Sequence Diagram ของการบันทึกการปรับปรุงยอดสต็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5 กระบวนการบันทึกการรับสินค้าเข้าสต็อกตามการสั่งสินค้า

เมื่อรายงานยอดขายได้ถูกจัดส่งมาแล้ว ทางแผนก Consignment ก็จะจัดการแยกเอกสาร แล้ว ผู้ควบคุมการขายของแผนกจะนำเอกสารการรับสินค้าเข้าสต็อกตามการสั่งสินค้าสาขา เข้าระบบงาน

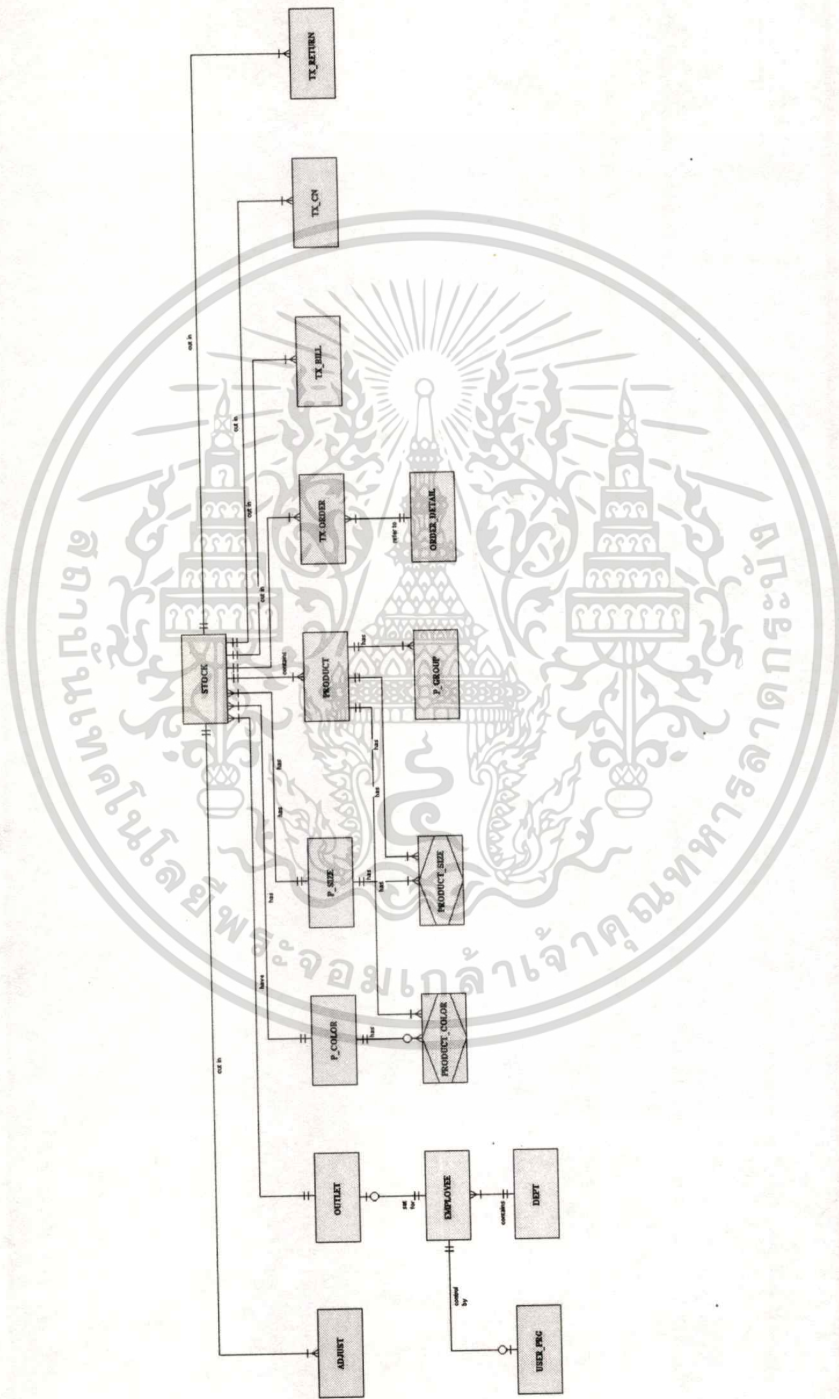


รูปที่ 4.26 แสดง Sequence Diagram ของการบันทึกการปรับปรุงยอดสต็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. E-R DIAGRAM

เป็นการนำ CLASS DIAGRAM ของ UML มาแปลงเป็น E-R DIAGRAM ของ DATA MODELING ซึ่งจะมองในลักษณะ PHYSICAL มากขึ้น



รูปที่ 4.27 แสดง E-R Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. DATA DICTIONARY

แสดง โครงสร้างของระบบฐานข้อมูลซึ่ง ได้ถูกออกแบบเป็นเทเบิลหลายๆเทเบิล และความ สัมพันธ์ในแต่ละเทเบิล สามารถแสดงได้ดังนี้

ตารางที่ 4.1 เทเบิล p\_size ใช้เก็บข้อมูลขนาดของเสื้อผ้า

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
s_id	CHAR(2)	PK		รหัสขนาดของเสื้อผ้า
s_name	CHAR(20)			ชื่อขนาดของเสื้อผ้า
s_desc	CHAR(20)			คำบรรยายอื่นๆ
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.2 เทเบิล p\_color ใช้เก็บข้อมูลสีของเสื้อผ้า

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
c_id	CHAR(4)	PK		รหัสสีของเสื้อผ้า
c_name	CHAR(20)			ชื่อสีของเสื้อผ้า
c_desc	CHAR(20)			คำบรรยายอื่นๆ
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.3 เทเบิล p\_group ใช้เก็บข้อมูลกลุ่มสินค้า

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
group_id	CHAR(3)	PK		รหัสกลุ่มสินค้า
group_name	CHAR(20)			ชื่อกกลุ่มสินค้า
group_desc	CHAR(20)			คำบรรยายอื่นๆ
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 เทเบิล products ใช้เก็บข้อมูลสินค้า

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK		รหัสสินค้า
p_name	CHAR(20)			ชื่อสินค้า
p_unitcall	CHAR(20)			หน่วยนับสินค้า
Price	FLOAT			ราคาต่อหน่วย
group-id	CHAR(3)	FK	p_group	รหัสกลุ่มสินค้า
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.5 เทเบิล outlet ใช้เก็บข้อมูลร้านค้า

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
o_id	CHAR(3)	PK		รหัสร้านค้า
store_name	CHAR(20)			ชื่อห้างสรรพสินค้า
branch_name	CHAR(20)			สาขา
tax_id	CHAR(11)			เลขที่กำกับภาษี
tax_add	CHAR(35)			ที่อยู่ตามกฎหมาย
emp_id	CHAR(6)	FK	employee	ราคาต่อหน่วย
emp_date	DATETIME			รหัสกลุ่มสินค้า
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 เทเบิล dept ใช้เก็บข้อมูลแผนกต่างๆในองค์กร

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
dept_id	CHAR(2)	PK		รหัสแผนก
dept_name	CHAR(20)			ชื่อแผนก
dept_desc	CHAR(20)			คำบรรยายอื่นๆ
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.7 เทเบิล employee ใช้เก็บข้อมูลพนักงานในองค์กร

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
e_id	CHAR(6)	PK		รหัสพนักงาน
e_name	CHAR(20)			ชื่อพนักงาน
e_sname	CHAR(20)			ชื่อสกุล
Bod	DATETIME			วันเกิด
Address	CHAR(35)			ที่อยู่
Position	CHAR(25)			ตำแหน่ง
Tel	CHAR(20)			โทรศัพท์
Mobile	CHAR(20)			มือถือ
dept_id	CHAR(2)	FK	dept	สังกัดแผนก
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 เทเบิล user-prg ใช้เก็บข้อมูลสิทธิในการใช้งานของพนักงาน

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
e_id	CHAR(6)	PK		รหัสพนักงาน
Username	CHAR(15)			ชื่อผู้ใช้
Password	CHAR(15)			รหัสผ่าน
Level	CHAR(1)			สิทธิในการใช้งาน
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.9 เทเบิล product\_color ใช้เก็บข้อมูลความสัมพันธ์ระหว่างสินค้าและสี

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK	products	รหัสสินค้า
c_id	CHAR(4)	PK	p_color	รหัสสีของเสื้อผ้า
User_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.10 เทเบิล product\_size ใช้เก็บข้อมูลความสัมพันธ์ระหว่างสินค้าและขนาด

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK	products	รหัสสินค้า
s_id	CHAR(2)	PK	p_size	รหัสขนาดของเสื้อผ้า
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 เทเบิล stock ใช้เก็บข้อมูลยอดคงเหลือสินค้า

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK,FK	products	รหัสสินค้า
c_id	CHAR(4)	PK,FK	p_color	รหัสสีของเสื้อผ้า
s_id	CHAR(2)	PK,FK	p_size	รหัสขนาดของเสื้อผ้า
o_id	CHAR(2)	PK,FK	outlet	รหัสร้านค้า
Total	INTEGER			จำนวนคงเหลือสินค้า
last_chkstock	INTEGER			จำนวนคงเหลือสินค้าครั้งล่าสุด
Last_chkdate	DATETIME			วันที่บันทึกแก้ไขสินค้าล่าสุด
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.12 เทเบิล tx\_bill ใช้เก็บข้อมูลรายการขายของแต่ละสินค้าในแต่ละวัน

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK,FK	stock	รหัสสินค้า
c_id	CHAR(4)	PK,FK	stock	รหัสสีของเสื้อผ้า
s_id	CHAR(2)	PK,FK	stock	รหัสขนาดของเสื้อผ้า
o_id	CHAR(2)	PK,FK	stock	รหัสร้านค้า
total	INTEGER			จำนวนสินค้าที่ขาย
price	INTEGER			ราคาที่ยขาย
tx_date	DATETIME			วันที่ขาย
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.13 เทเบิล tx\_cn ใช้เก็บข้อมูลรายการคืนสินค้ากลับโรงงานขายของแต่ละสินค้าในแต่ละวัน

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK,FK	stock	รหัสสินค้า
c_id	CHAR(4)	PK,FK	stock	รหัสสีของเสื้อผ้า
s_id	CHAR(2)	PK,FK	stock	รหัสขนาดของเสื้อผ้า
o_id	CHAR(2)	PK,FK	stock	รหัสร้านค้า
total	INTEGER			จำนวนสินค้าที่คืน
cn_cause	CHAR(30)			เหตุที่คืนสินค้า
cn_date	DATETIME			วันที่คืนสินค้า
cn_ref	CHAR(20)			เอกสารอ้างอิงการคืนสินค้า
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.14 เทเบิล tx\_return ใช้เก็บข้อมูลรายการรับคืนสินค้าจากลูกค้าของแต่ละสินค้าในแต่ละวัน

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK,FK	stock	รหัสสินค้า
c_id	CHAR(4)	PK,FK	stock	รหัสสีของเสื้อผ้า
s_id	CHAR(2)	PK,FK	stock	รหัสขนาดของเสื้อผ้า
o_id	CHAR(2)	PK,FK	stock	รหัสร้านค้า
total	INTEGER			จำนวนสินค้าที่คืน
price	FLOAT			ราคาซื้อ
bill_no	CHAR(11)			เลขที่ใบเสร็จรับเงิน
buy_date	DATETIME			วันที่ซื้อสินค้า
ret_date	DATETIME			วันที่รับคืนสินค้า
ret_cause	CHAR(35)			เหตุผลที่คืนสินค้า
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.15 เทเบิล tx\_order ใช้เก็บข้อมูลรายการรับสินค้าเข้าคลังตามสาขาของแต่ละสินค้าในแต่ละวัน

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
p_id	CHAR(13)	PK,FK	stock	รหัสสินค้า
c_id	CHAR(4)	PK,FK	stock	รหัสสีของเสื้อผ้า
s_id	CHAR(2)	PK,FK	stock	รหัสขนาดของเสื้อผ้า
o_id	CHAR(2)	PK,FK	stock	รหัสร้านค้า
od_id	CHAR(13)	PK,FK	order_detail	รหัสรายละเอียดย่อย
total	INTEGER			จำนวนสินค้าที่รับเข้า
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

ตารางที่ 4.16 เทเบิล order\_detail ใช้เก็บข้อมูลย่อยการรับสินค้าเข้าคลังตามสาขาของแต่ละสินค้าในแต่ละวัน

ชื่อฟิลด์	ชนิดข้อมูล	ลักษณะ	เทเบิล	หมายเหตุ
od_id	CHAR(13)	PK		รหัสรายละเอียดย่อย
o_id	CHAR(2)	PK,FK	tx_order	รหัสร้านค้า
tax_invoice	CHAR(11)			เลขที่ใบกำกับภาษี
tax_date	DATETIME			วันที่ออกใบกำกับภาษี
order_ref	CHAR(6)			เลขที่เอกสารการสั่งสินค้า
order_date	DATETIME			วันที่สั่งสินค้า
grand_total	INTEGER			ยอดเงินรวม
user_upd	CHAR(6)	FK	employee	รหัสผู้บันทึกแก้ไข
upd-date	DATETIME			วันที่บันทึกแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

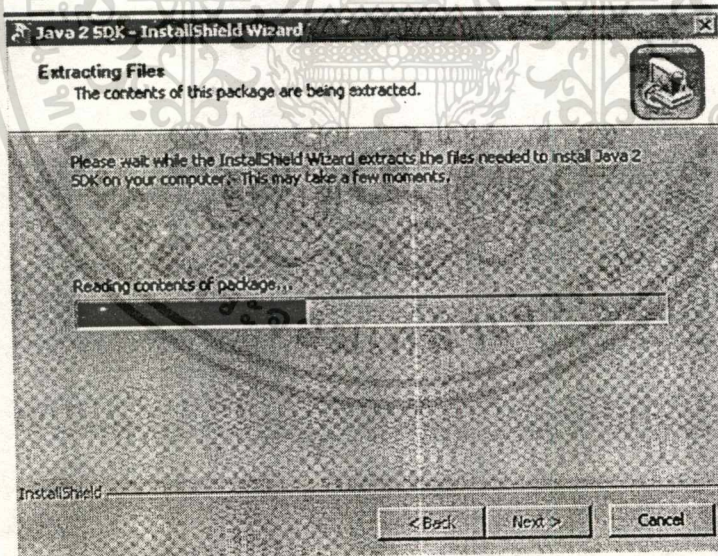
### การติดตั้งระบบ

การติดตั้งซอฟต์แวร์และกำหนดสภาวะแวดล้อมของ JSP

- 5.1 การติดตั้ง JAVA 2 SDK(Java 2 Software Development Kit)
- 5.2 การแก้ไขตัวแปร Path และ Class Path ของ JAVA 2 SDK
- 5.3 การติดตั้งเซิร์ฟเวอร์ TOMCAT (Installing the Tomcat Server)

5.1 การติดตั้ง JAVA 2 SDK(Java 2 Software Development Kit) เนื่องจากการที่เอกสาร JSP ต้องทำงานภายใต้สภาวะแวดล้อม JAVA จึงต้องติดตั้ง Java 2 SDK สำหรับ Windows, Linux และ Solaris ซึ่งสามารถไปดาวน์โหลดได้ที่เว็บไซต์ <http://java.sun.com/j2se> จากนั้นก็จะทำการติดตั้ง ในที่นี้ขอใช้ Java 2 SDK for Windows ซึ่งมีขั้นตอนการติดตั้งดังนี้

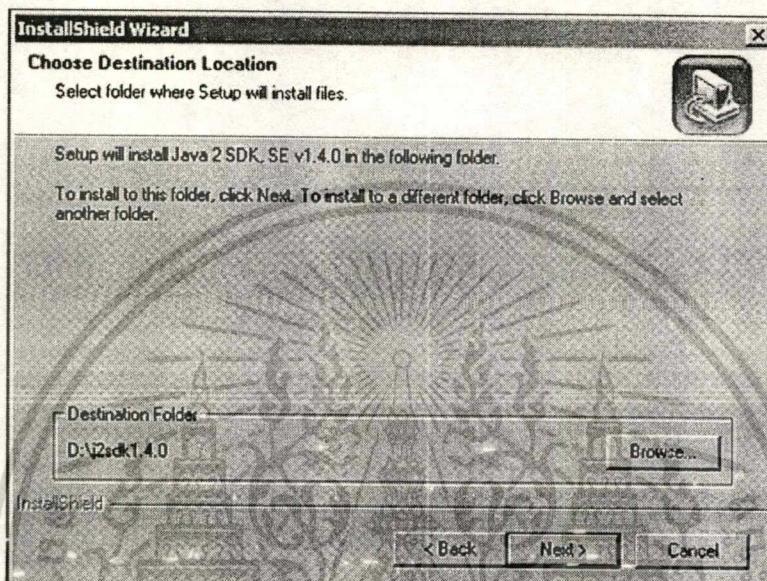
1. ดับเบิลคลิกที่ไฟล์เพื่อเริ่มติดตั้ง Java 2 SDK



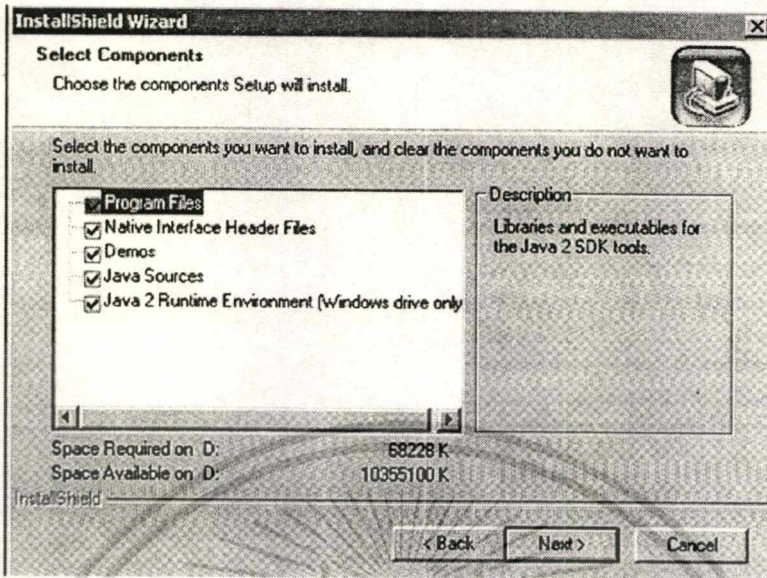
รูปที่ 5.1 เริ่มติดตั้ง Java 2SDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

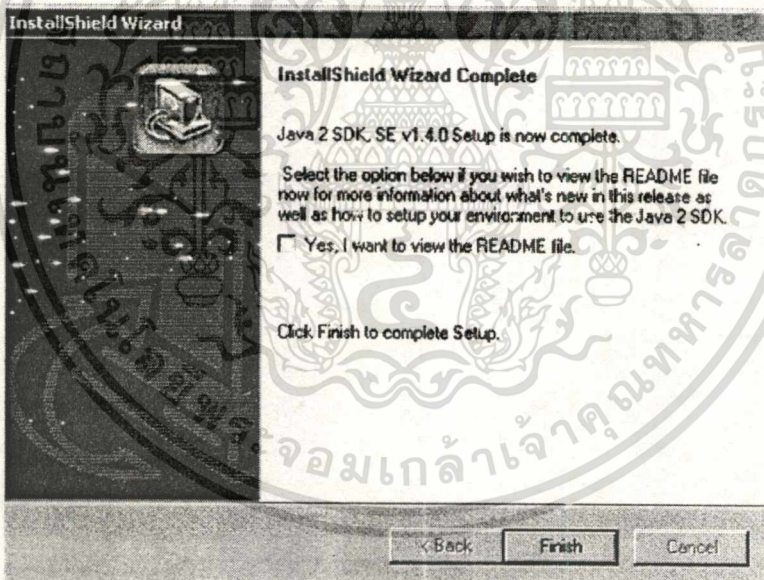
1. ให้เลือกปุ่ม Next ไปเรื่อย ๆ จากนั้นให้เลือกไดเรกทอรีเพื่อทำการติดตั้ง โดยเลือกที่ไดเรกทอรีไหนก็ได้ แต่ขออ้างอิงที่ D:\j2sdk1.4.0 จากนั้นกดปุ่ม Next จะเป็นการแสดงข้อมูลการติดตั้ง JDK เช่น การ copy ไฟล์ต่าง ๆ ไปเก็บไว้ในไดเรกทอรีที่เลือกติดตั้ง เป็นต้น



รูปที่ 5.2 เลือก Directory สำหรับติดตั้ง Java 2SDK



รูปที่ 5.3 เลือก Components สำหรับติดตั้ง Java 2SDK



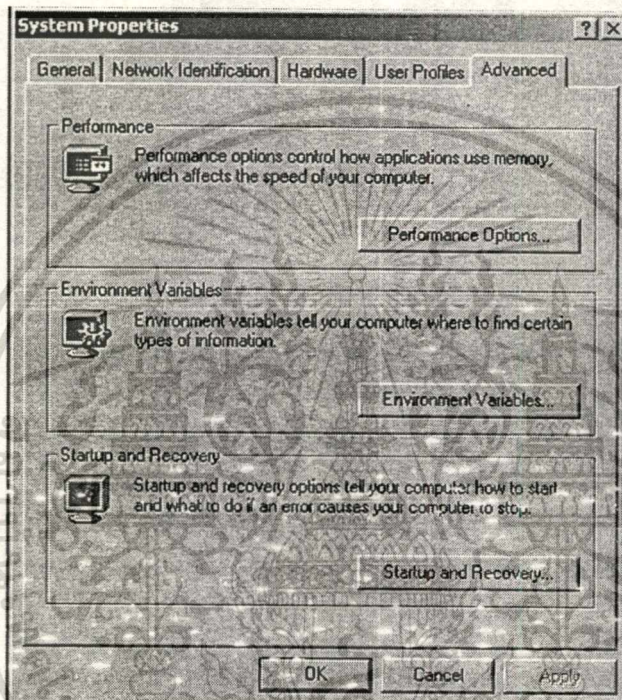
รูปที่ 5.4 ติดตั้งเสร็จสมบูรณ์

ติดตั้งเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

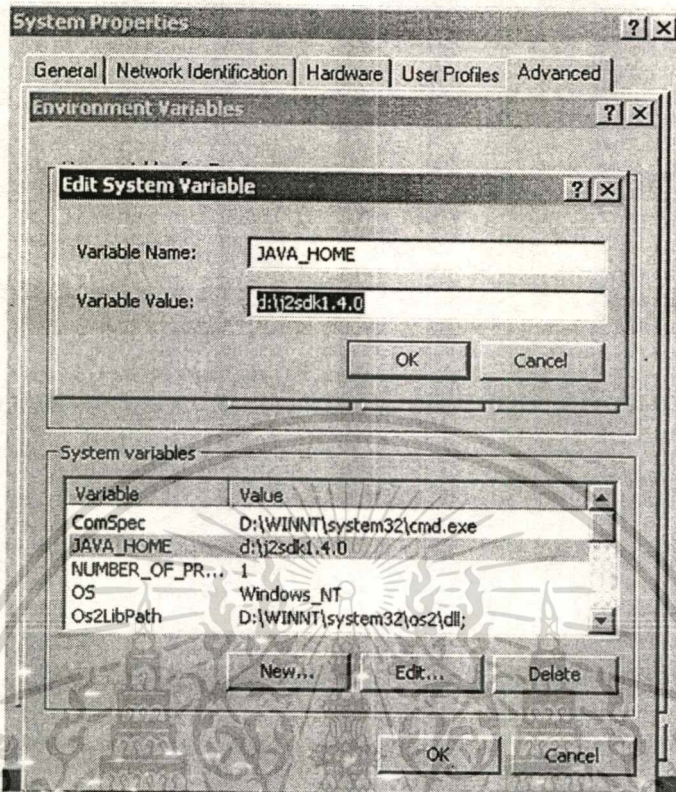
5.2 การแก้ไขตัวแปร Path และ Class Path เมื่อติดตั้ง JDK 2 SDK เสร็จเรียบร้อยแล้ว จะต้องกำหนดตัวแปร Path และ Classpath การกำหนดค่าให้ตัวแปร Path นั้นก็เพื่อให้คอมไพเลอร์จาวา (JAVAC) สามารถเรียกจากที่ใดก็ได้ นอกจากไดเรกทอรีที่ติดตั้งไว้ ส่วนตัวแปร Classpath นั้นช่วยให้โปรแกรมจาวาที่เราพัฒนาขึ้น สามารถเชื่อมโยงกับคลาสไลบรารีอื่น ๆ ที่อยู่คนละไดเรกทอรีกันได้ ขั้นตอนการกำหนดตัวแปรทั้งสองมีดังนี้

1. คลิกขวาที่ My computer เลือกที่เมนู Properties



รูปที่ 5.5 แสดง System Properties

2. จะปรากฏได้อะลือกบ็อกซ์ System Properties ให้คลิกที่แท็บ Advanced แล้วคลิกที่ปุ่ม Environment Variables จากนั้นให้คลิกเลือกที่ตัวแปร path ในช่องรายการ System Variables เพิ่มเติมค่าที่ตัวแปร Path เป็น C:\j2sdk1.4.0\bin\; และจากนั้นให้คลิกเลือกตัวแปร Classpath ให้เพิ่มเติมค่าที่ตัวแปร Classpath เป็น C:\j2sdk1.4.0\jre\lib\;



รูปที่ 5.6 แสดงการ Set path และ Class path

### 5.3 การติดตั้งเซิร์ฟเวอร์ TOMCAT (Installing the Tomcat Server) ซึ่งมีขั้นตอนการติดตั้งดังนี้

1. สามารถดาวน์โหลด Tomcat 4.0 ได้ที่ <http://jakarta.apache.org> เมื่อดาวน์โหลดมาเรียบร้อยแล้วให้ดับเบิลคลิกที่ไฟล์เพื่อเริ่มติดตั้ง Tomcat 4.0 ระบบการติดตั้งจะทำการ Unzip ไฟล์ Tomcat 4.0
2. ให้เลือกไดเรกทอรี C:\jakarta-tomcat-4.0 เพื่อแตกไฟล์ที่ zip มาไปเก็บไว้ เมื่อการก๊อปปี้เสร็จแล้วจะปรากฏไดเรกทอรีและไฟล์ต่าง ๆ ในไดเรกทอรี C:\jakarta-tomcat-4.0 และเป็นอันเสร็จสิ้นการติดตั้ง
3. การแก้ไขตัวแปรต่างๆ เมื่อก๊อปปี้ไฟล์ต่าง ๆ ไปเก็บไว้ในไดเรกทอรีที่กำหนดแล้วขั้นตอนต่อไป จะต้องกำหนดตัวแปร เพื่อให้ Tomcat ทำงานได้ ประกอบไปด้วย 2 ตัวแปรคือตัวแปร Tomcat\_Home กับ Java\_Home ให้ผู้อ่านไปยัง Environment Variable อีกครั้ง แล้วเพิ่มตัวแปร 2 ตัวดังต่อไปนี้ (ขั้นตอนเดียวกับการแก้ไขตัวแปรใน JAVA 2 SDK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือกที่ปุ่ม New เพื่อสร้างตัวแปรใหม่ใน System Variables
- สร้างตัวแปร TOMCAT\_HOME ใส่ค่าให้กับตัวแปรเป็นไคเรกทอรีที่เราเลือกติดตั้ง Tomcat ในที่นี้คือ C:\jakarta-tomcat-4.0
- สร้างตัวแปร JAVA\_HOME ใส่ค่าให้กับตัวแปรเป็นไคเรกทอรีที่เราเลือกติดตั้ง JAVA 2 SDK

### หมายเหตุ

สำหรับระบบที่ใช้ระบบปฏิบัติการ Windows 95/98/Me ให้เพิ่มค่าตัวแปรทั้งสองตัวที่ไฟล์ autoexec.bat ดังนี้

- TOMCAT\_HOME = C:\jakarta-tomcat-4.0;
- JAVA\_HOME = C:\J2DK1.4.0;

แล้วรีสตาร์ทเครื่องเพื่อให้ค่าตัวแปรเปลี่ยนแปลงตามที่เรากำหนดไว้

ความหมายของไคเรกทอรีต่าง ๆ ไคเรกทอรีที่สำคัญใน ไคเรกทอรี C:\jakarta-tomcat-4.0

Directory	Content
Bin	เก็บไฟล์ที่ควบคุมการทำงานของ Tomcat เช่น Startup.bat, Shutdown.bat
Conf	เก็บไฟล์คอนฟิกต่างๆ เช่น server.xml
Doc	เก็บคู่มือการใช้ Tomcat
Lib	เก็บคลาสไลบรารีต่างๆ เช่น Servlet.jar
Logs	เก็บ Logs file หากเกิดข้อผิดพลาดสามารถมาดูได้จากไฟล์เหล่านี้
WebApps	เป็น Home Directory ของเว็บเซิร์ฟเวอร์

### ตารางที่ 5.1 แสดงไคเรกทอรีและความหมาย ของ TOMCAT

ทดสอบการใช้งาน เมื่อได้ติดตั้ง Tomcat เรียบร้อยแล้ว เราสามารถมาทดสอบว่าขั้นตอนที่เราติดตั้งนั้น สามารถทำงานได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือกที่ปุ่ม New เพื่อสร้างตัวแปรใหม่ใน System Variables
- สร้างตัวแปร TOMCAT\_HOME ใส่ค่าให้กับตัวแปรเป็นไดเรกทอรีที่เราเลือกติดตั้ง Tomcat ในที่นี้คือ C:\jakarta-tomcat-4.0
- สร้างตัวแปร JAVA\_HOME ใส่ค่าให้กับตัวแปรเป็นไดเรกทอรีที่เราเลือกติดตั้ง JAVA 2 SDK

#### หมายเหตุ

สำหรับระบบที่ใช้ระบบปฏิบัติการ Windows 95/98/Me ให้เพิ่มค่าตัวแปรทั้งสองตัวที่ไฟล์ autoexec.bat ดังนี้

- TOMCAT\_HOME = C:\jakarta-tomcat-4.0;
- JAVA\_HOME = C:\J2DK1.4.0;

แล้วรีสตาร์ทเครื่องเพื่อให้ค่าตัวแปรเปลี่ยนแปลงตามที่เรากำหนดไว้

ความหมายของไดเรกทอรีต่างๆ ไดเรกทอรีที่สำคัญใน ไดเรกทอรี C:\jakarta-tomcat-4.0

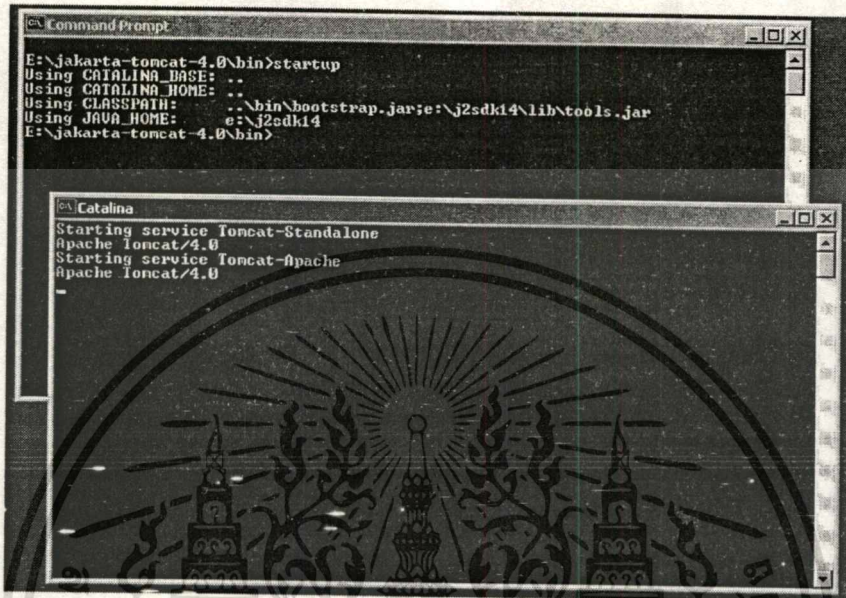
ตารางที่ 5.1 แสดงไดเรกทอรีและความหมาย ของ TOMCAT

Directory	Content
Bin	เก็บไฟล์ที่ควบคุมการทำงานของ Tomcat เช่น Startup.bat, Shutdown.bat
Conf	เก็บไฟล์คอนฟิกต่างๆ เช่น server.xml
Doc	เก็บคู่มือการใช้ Tomcat
Lib	เก็บคลาสไลบรารีต่างๆ เช่น Servlet.jar
Logs	เก็บ Logs file หากเกิดข้อผิดพลาดสามารถมาดูได้จากไฟล์เหล่านี้
WebApps	เป็น Home Directory ของเว็บเซิร์ฟเวอร์

ทดสอบการใช้งาน เมื่อได้ติดตั้ง Tomcat เรียบร้อยแล้ว เราสามารถมาทดสอบว่าขั้นตอนที่เราติดตั้งนั้น สามารถทำงานได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

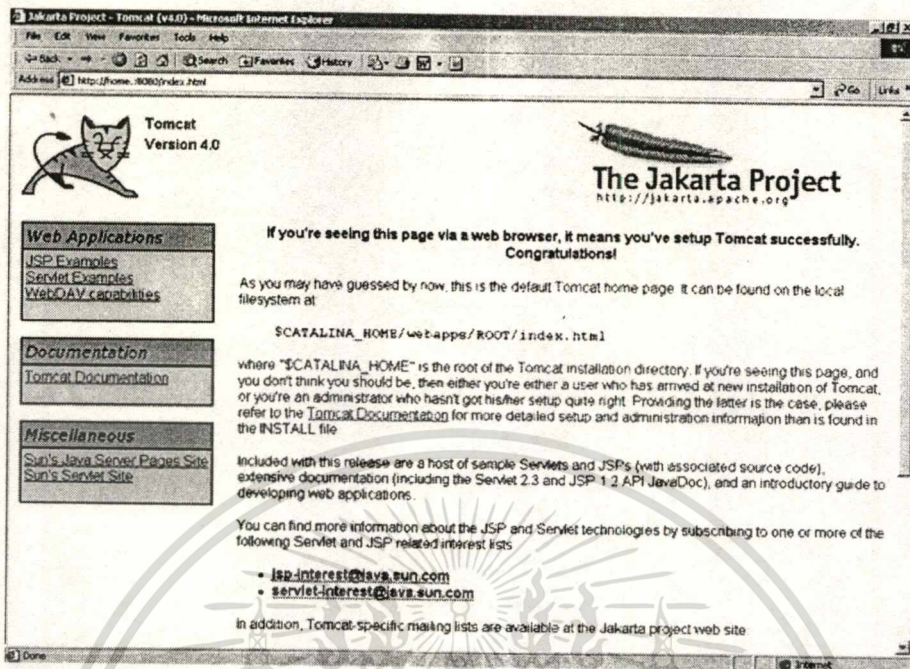
1. ให้เรียก Command Prompt ขึ้นมาแล้วเข้าไปในไดเรกทอรีที่ติดตั้ง Tomcat ไว้ในที่นี้คือ C:\jakarta-tomcat-4.0 จากนั้นให้เข้าไปในไดเรกทอรี bin เพื่อเรียก Tomcat จากไฟล์ StartUp.bat



รูปที่ 5.7 แสดงทดสอบการเรียก Tomcat จาก C:\Tomcat\Bin\StarUp.bat

2. เมื่อเรียกไฟล์ StartUp.bat แล้ว จะปรากฏ Command Prompt อีกหน้าต่างชื่อว่า Catalina ขึ้นมาและ Command Prompt จะค้างอยู่ที่ 2 หน้าต่าง หากมีอันใดปิดไปเอง แสดงว่ามีข้อผิดพลาดเกิดขึ้นให้ ผู้อ่าน ไปดูไฟล์ log ในไดเรกทอรี Logs
3. เมื่อหน้าต่าง Catalina ปรากฏค้างอยู่แสดงว่า การติดตั้งของเราเสร็จสมบูรณ์แล้ว ผู้อ่าน สามารถเปิดเว็บเบราว์เซอร์ไปยัง <http://localhost:8080> ซึ่งจะเป็นที่เก็บเอกสาร Home Pages ของ JSP สำหรับค่าเริ่มต้นนั้น Tomcat จะทำงานที่พอร์ต 8080

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



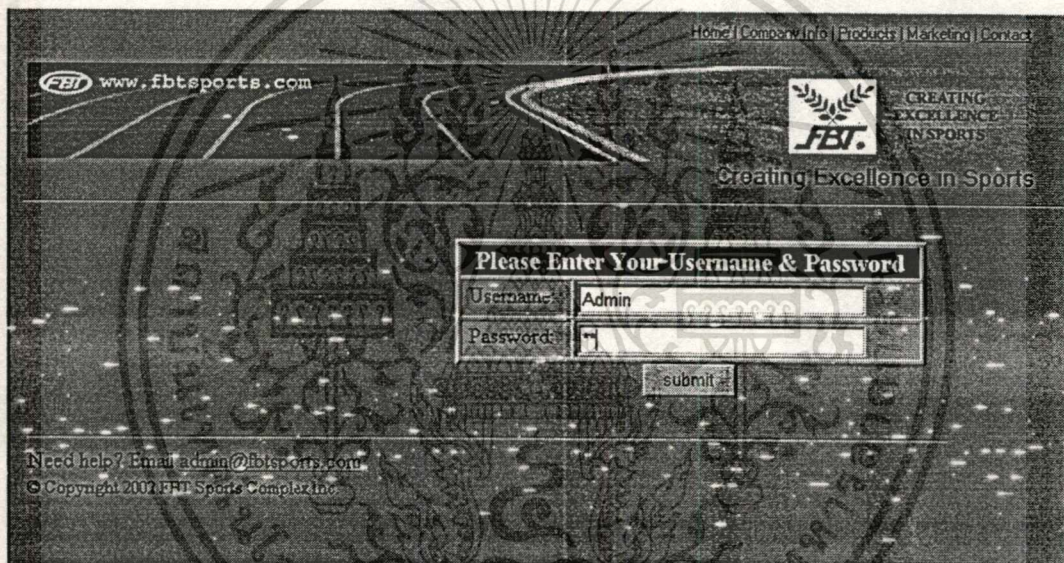
รูปที่ 5.8 แสดงทดสอบการเรียก HOME PAGES ของ Tomcat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### การพัฒนาระบบงาน

เมื่อได้ทำการวิเคราะห์และออกแบบระบบงานเป็นที่เรียบร้อยแล้ว ก็จะนำสิ่งที่ได้มาลงมือพัฒนาระบบงาน โดยใช้ JSP ในการพัฒนา Web Pages ควบคู่กับ Microsoft SQL SERVER 2000 จัดทำฐานข้อมูล ซึ่งสิ่งที่ได้จากการพัฒนาจะนำเสนอรูปหน้าจอการทำงานของระบบงานคร่าวๆ ดังนี้



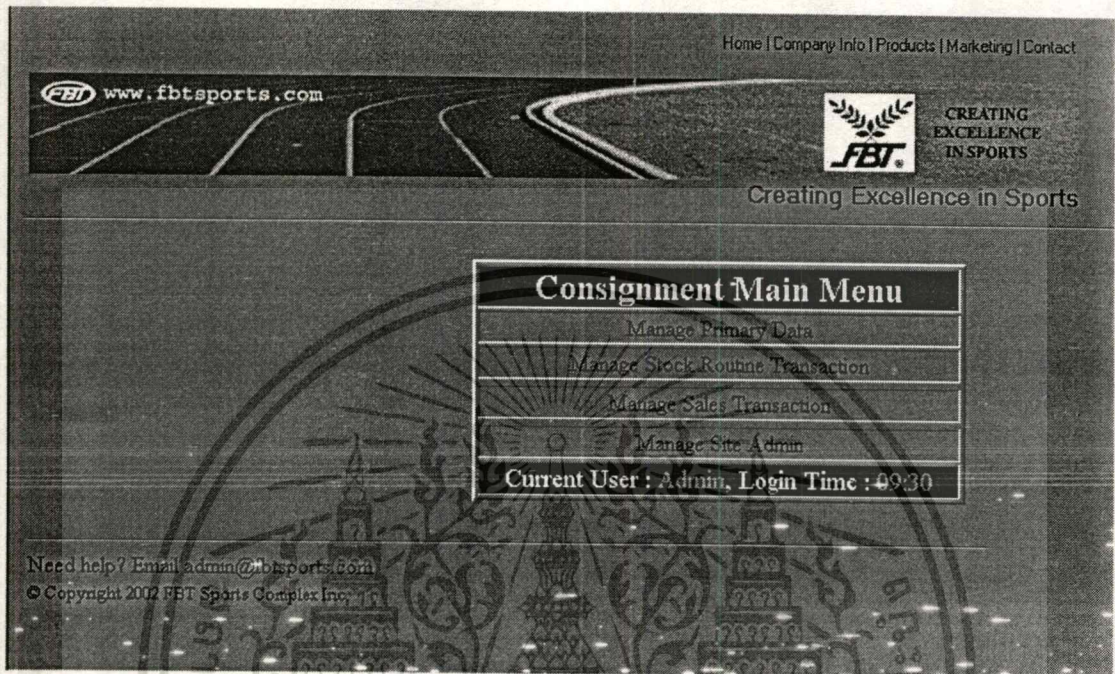
รูปที่ 6.1 แสดงหน้าแรกของระบบงาน

เริ่มต้นการใช้ระบบงาน ได้มีการแบ่งการใช้งานเป็น 4 ระดับซึ่งมองภาพด้านความปลอดภัยของข้อมูลเป็นหลัก คือ

- สำหรับผู้ใช้งานที่ต้องการดูข้อมูลอย่างเดียว สามารถเข้าสู่การใช้งาน โดยการใช้รหัสพนักงานของแต่ละบุคคลเป็น username และ password ได้ เรียกผู้ใช้งานพวกนี้ว่า GUEST
- สำหรับผู้ใช้งานที่ต้องการปรับปรุงแก้ไขข้อมูลทั่วไป สามารถเข้าสู่การใช้งาน โดยการใช้ username และ password ที่ได้รับจากผู้ดูแลระบบ เรียกผู้ใช้งานพวกนี้ว่า USER
- สำหรับผู้ที่เป็นผู้ควบคุมงานของแต่ละแผนกก็จะสามารถเข้าใช้งานของระบบงานใน

เอกสารนี้เป็นเอกสาร MENU ภัทร ใช้งาน และ สามารถกำหนดสิทธิ์การเข้าถึงข้อมูลของผู้ใช้ในระดับล่างด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สำหรับผู้ดูแลระบบงาน จะเป็นผู้กำหนดสิทธิการใช้งานของผู้ใช้และการใช้งานระบบงานข้อมูล เรียกผู้ดูแลระบบงานว่า ADMIN



รูปที่ 6.2 แสดง Menu หลักของระบบงาน

สำหรับการรับข้อมูล user และ password ผู้ใช้จะต้องใส่ข้อมูลให้ครบ ปุ่ม submit จึงจะปรากฏ ดังรูปที่ 6.1 เพื่อทำงานต่อไป ซึ่งการเข้าสู่การใช้งานระบบแต่ละครั้งของผู้ใช้ต่างๆ จะปรากฏ ชื่อผู้ใช้งาน และเวลาการเข้าสู่ระบบงานในทุกๆ MENU การทำงาน

เมื่อระบบรับข้อมูลเบื้องต้นถูกต้องแล้วก็จะเข้าสู่ MENU หลักของระบบ ดังรูปที่ 6.2 โดยจะมี MENU ย่อยๆ ลงไปดังนี้

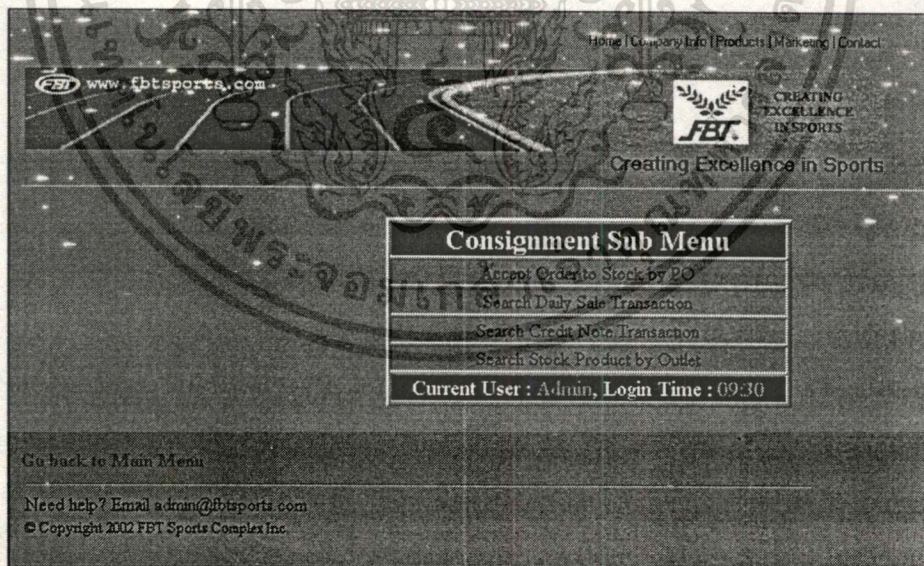
- Manage Primary Data เป็น MENU ย่อยที่แสดงการจัดเก็บ/ปรับปรุง ข้อมูลพื้นฐานของระบบงาน ซึ่งมีรายละเอียดย่อยลงไป แสดงด้วยรูปที่ 6.5
- Manage Stock Routine Transaction เป็น MENU ย่อยที่แสดงการจัดเก็บ/ปรับปรุง ข้อมูลทางด้านสินค้าคงคลัง โดยการทำงานของ MENU นี้ จะแสดงเป็นหัวข้อต่างๆตามรูปที่ 6.3
- Manage Sales Transaction เป็น MENU ย่อยที่แสดงการจัดเก็บ/ปรับปรุง ข้อมูลทางด้านฝ่ายขายและการรับข้อมูลเข้าสู่ สาขาต่างๆ รายละเอียดของ MENU นี้ จะแสดงเป็นหัวข้อต่างๆตามรูปที่ 6.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Manage Site Admin เป็น MENU ย่อยที่แสดงการจัดการจัดเก็บ/ปรับปรุง ข้อมูลทางด้าน การกำหนดความปลอดภัยของระบบงาน เช่นการกำหนดสิทธิการใช้งาน เป็นต้น

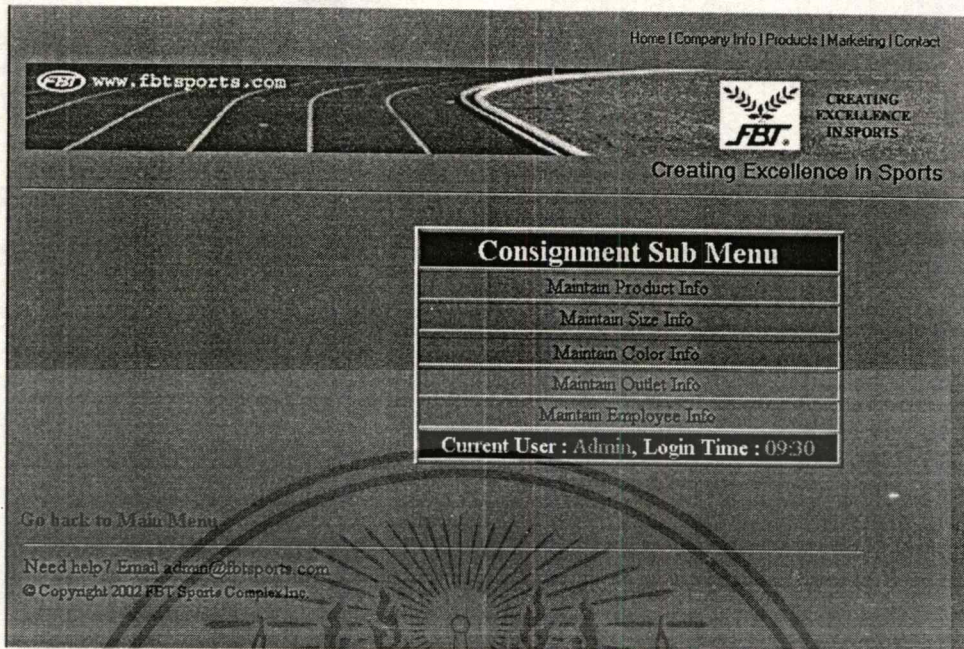


รูปที่ 6.3 แสดง Manage Stock Routine Transaction Menu



รูปที่ 6.4 แสดง Manage Sales Transaction Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### รูปที่ 6.5 แสดงหน้า Sub Menu หลักของระบบงาน

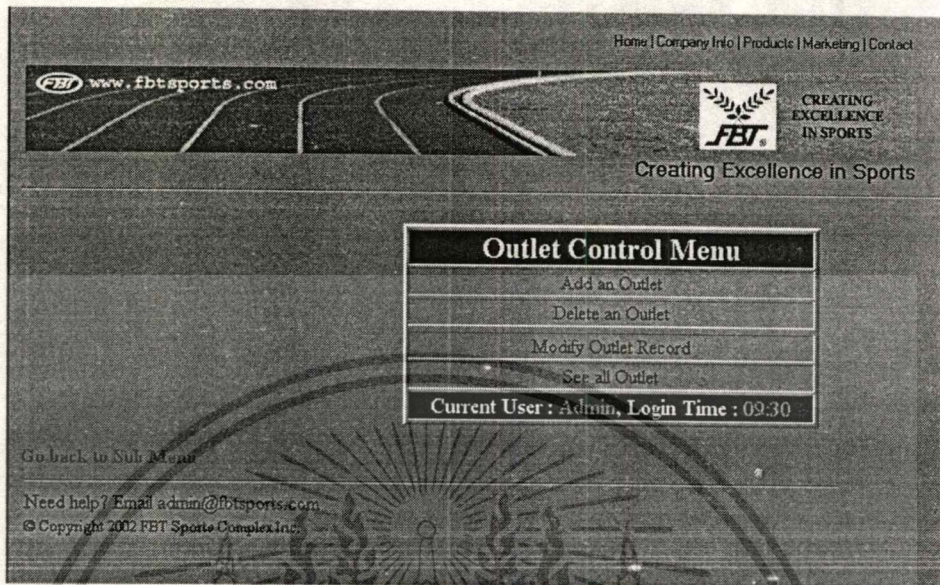
เมื่อผู้ใช้ที่มีสิทธิเข้าแก้ไข/ปรับปรุงข้อมูลพื้นฐานถือคอินเข้าสู่ระบบจะต้องเข้ามาที่ MENU Manage Primary Data ซึ่งที่ MENU นี้จะแสดง Sub Menu ต่างๆ ดังรูปที่ 6.5 โดยมีรายละเอียดดังนี้

- Maintain Product Info เป็นการแก้ไข/ปรับปรุง ข้อมูลพื้นฐานทางด้านรายละเอียดสินค้า ซึ่งสามารถทำการ ADD,UPDATE,DELETE,VIEW ข้อมูลได้ที่ MENU นี้
- Maintain Size Info เป็นการแก้ไข/ปรับปรุง ข้อมูลพื้นฐานทางด้านรายละเอียดขนาดสินค้า ซึ่งสามารถทำการ ADD,UPDATE,DELETE,VIEW ข้อมูลได้ที่ MENU นี้
- Maintain Color Info เป็นการแก้ไข/ปรับปรุง ข้อมูลพื้นฐานทางด้านรายละเอียดสีสินค้า ซึ่งสามารถทำการ ADD,UPDATE,DELETE,VIEW ข้อมูลได้ที่ MENU นี้
- Maintain Outlet Info เป็นการแก้ไข/ปรับปรุง ข้อมูลพื้นฐานทางด้านรายละเอียดจุดขายสินค้าตามห้างสรรพสินค้าต่างๆที่ทางบริษัทมีจุดขายเปิดให้บริการอยู่ ซึ่งสามารถทำการ ADD,UPDATE,DELETE,VIEW ข้อมูลได้ที่ MENU นี้
- Maintain Employee Info เป็นการแก้ไข/ปรับปรุง ข้อมูลพื้นฐานทางด้านรายละเอียดพนักงานของบริษัท ซึ่งสามารถทำการ ADD,UPDATE,DELETE,VIEW ข้อมูลได้ที่ MENU นี้

เมื่อผู้ใช้ที่มีสิทธิเข้าแก้ไข/ปรับปรุงข้อมูลพื้นฐาน เช่นเข้าไปในส่วนของการแก้ไข/ปรับปรุงข้อมูลด้านจุดขาย สามารถแสดงการ ADD,UPDATE,DELETE,VIEW ข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังรูปที่ 6.7 เป็น MENU สำหรับทำงาน ซึ่งเรียกว่า Outlet Control Menu



รูปที่ 6.6 แสดงหน้า Outlet Control Menu ของระบบงาน

จากรูปที่ 6.7 จะมีหัวข้อย่อยสำหรับทำงานในการแก้ไข/ปรับปรุงข้อมูลจุดขาย ดังนี้

- Add an Outlet เป็นหัวข้อเพื่อเข้าสู่การเพิ่มข้อมูลจุดขาย
- Delete an Outlet เป็นหัวข้อเพื่อเข้าสู่การลบข้อมูลจุดขาย
- Modify Outlet Record เป็นหัวข้อเพื่อเข้าสู่การแก้ไขข้อมูลจุดขายที่มีอยู่เดิมแล้ว
- See all Outlet เป็นหัวข้อเพื่อดูข้อมูลจุดขายทั้งหมด

ดังนี้

การเพิ่มข้อมูลจุดขาย เป็นการเพิ่มจุดขายเข้าสู่ระบบ ดังรูปที่ 6.8 โดยข้อมูลที่ผู้ใช้ต้องเพิ่มมี

- รหัสจุดขายตั้ง โดยผู้ควบคุมงานของแผนกสินค้าคงคลัง
- ห้างสรรพสินค้าที่จุดขายนั้นๆตั้ง
- สาขาของห้างสรรพสินค้าที่จุดขายนั้นๆตั้ง
- หมายเลขผู้เสียภาษีของจุดขายนั้นๆ
- ที่อยู่ตามหนังสือบริคณสนธิ
- พนักงานขายประจำจุดขายนั้นๆ

เอกสารนี้เป็นเอกสารที่พนักงานขายคนนั้นๆเข้าประจำจุดขายนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Home | Company Info | Products | Marketing | Contact

FBI www.fbtsports.com

FBI CREATING EXCELLENCE IN SPORTS

Creating Excellence in Sports

### Add An Outlet

Outlet ID: 02

Store Name: The Mall

Branch Name: som-rong

Tax ID: 12345678901

Tax Address: 157/7 Taparat Rd.

Employee: 998211 Pongthon Rudee

Employee Date: 4-1-2003 (Please Select Date from Dropdown)

<	April	>	2003	<	>	
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Go back to Control Menu

Need help? Email admin@fbtsports.com

© Copyright 2002 FBI Sports Complex Inc.

รูปที่ 6.7 แสดงหน้าการเพิ่มข้อมูลจุดขายของระบบงาน

Home | Company Info | Products | Marketing | Contact

FBI www.fbtsports.com

FBI CREATING EXCELLENCE IN SPORTS

Creating Excellence in Sports

### Successful Addition of Outlet

Thank you for adding new outlet.

- See all Outlet

Go back to admin control

Need help? Email admin@fbtsports.com

© Copyright 2002 FBI Sports Complex Inc.

รูปที่ 6.8 แสดงหน้าเมื่อการเพิ่มข้อมูลสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไขข้อมูลเดิมของจุดขาย ซึ่งระบบจะคืนข้อมูลของจุดขายที่มีอยู่แล้วมาให้ผู้ใช้เลือก เพื่อแก้ไข ดังรูปที่ 6.10 โดยข้อมูลที่ผู้ใช้สามารถแก้ไขได้มีดังนี้

- ห้างสรรพสินค้าที่จุดขายนั้นๆตั้ง
- สาขาของห้างสรรพสินค้าที่จุดขายนั้นๆตั้ง
- หมายเลขผู้เสียภาษีของจุดขายนั้นๆ
- ที่อยู่ตามหนังสือบริคณสนธิ
- พนักงานขายประจำจุดขายนั้นๆ
- วันที่ พนักงานขายคนนั้นๆ เข้าประจำ ณ จุดขาย

จะเห็นว่าระบบจะไม่อนุญาต ให้ผู้ใช้แก้ไขรหัสจุดขายได้ ขั้นตอนการแก้ไขสามารถแสดง ด้วยรูปที่ 6.9 รูปที่ 6.10 และ รูปที่ 6.11 โดยขั้นตอนการแก้ไขมีดังนี้

- ระบบคืนข้อมูลให้เลือก
- ผู้ใช้เลือกเพื่อแก้ไข
- ปราบกฏหน้าจอเพื่อการแก้ไขข้อมูล
- บันทึกข้อมูล
- แสดงการบันทึกข้อมูลเรียบร้อย

Outlet ID	Store Name	Branch Name	Tax ID	Address	Employee	Employee Date	
02	The Mall	som-rong	12345678901	1577 Tapara Rd.	998211	2003-04-01	modify
03	Central	Lad-prow	12345678901	1 Ladprowd Rd.	998220	2003-04-02	modify
01	The Mall	Bangkapi	1111	111111	998211	2003-03-27	modify

รูปที่ 6.9 แสดงหน้าจอเมื่อต้องการการแก้ไขข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Home | Company Info | Products | Marketing | Contact

www.fbtsports.com

CREATING EXCELLENCE IN SPORTS

Creating Excellence in Sports

### Modify an Outlet Record

Outlet ID:	02
Store:	The Mall
Branch:	som-rong
Tax ID:	12345678901
Tax Address:	157/7 Taparat Rd.
Employee:	998211 Pongthon Rudee
Employee Date:	2003-04-01 <small>(Please Select Date from Dropdown)</small>

Go back to Control Menu

Need help? Email [admin@fbtsports.com](mailto:admin@fbtsports.com)  
© Copyright 2002 FBT Sports Complex Inc.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Today: April 8, 2003

รูปที่ 6.10 แสดงหน้าจอการแก้ไขข้อมูล

Home | Company Info | Products | Marketing | Contact

www.fbtsports.com

CREATING EXCELLENCE IN SPORTS

Creating Excellence in Sports

**The outlet record has been modified.**

[See all Outlets](#)

Go back to Control Menu

Need help? Email [admin@fbtsports.com](mailto:admin@fbtsports.com)  
© Copyright 2002 FBT Sports Complex Inc.

รูปที่ 6.11 แสดงหน้าจอเมื่อการแก้ไขข้อมูลเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลบข้อมูลจุดขาย ผู้ใช้สามารถลบข้อมูลจุดขาย โดยระบบจะค้นข้อมูลเพื่อให้เลือกว่า ต้องการจะลบข้อมูลใด ดังรูปที่ 6.12 โดยการลบข้อมูลมีขั้นตอนดังนี้

- ระบบค้นข้อมูลเพื่อให้เลือกข้อมูลที่ต้องการลบ
- กดปุ่มเพื่อลบข้อมูล
- แสดงการลบข้อมูลเรียบร้อยแล้วผู้ใช้

ทั้ง 3 ขั้นตอนสามารถแสดงด้วยรูปที่ 6.12 และรูปที่ 6.13

Outlet ID	Store Name	Branch Name	Tax ID	Address	Employee	Employee Date	Delete?
02	The Mall	som-rong	12345678901	157/7 Taparat Rd.	998211	2003-04-01	<a href="#">delete</a>
03	Central	Lad-prow	12345678901	1 Ladprowd Rd.	998220	2003-04-02	<a href="#">delete</a>
01	The Mall	Bangkapi	1111	1111111	998211	2003-03-27	<a href="#">delete</a>

Go back to Control Menu

Need help? Email [admin@fbtsports.com](mailto:admin@fbtsports.com)

© Copyright 2002 FBT Sports Complex Inc.

รูปที่ 6.12 แสดงหน้าจอเมื่อต้องการลบข้อมูล

Outlet Record has been Deleted

[See outlets](#)

Go back to admin control

Need help? Email [admin@fbtsports.com](mailto:admin@fbtsports.com)

© Copyright 2002 FBT Sports Complex Inc.

รูปที่ 6.13 แสดงหน้าจอเมื่อการลบข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### บทสรุปและข้อเสนอแนะ

จากการพัฒนาระบบจัดการสินค้าคงคลังแบบฝากขาย นั้นผลที่ได้สามารถสรุปและมีข้อเสนอแนะดังต่อไปนี้

#### 7.1 บทสรุป

- การวิเคราะห์และออกแบบระบบงานโดยใช้หลักการ UML สามารถช่วยสะท้อนให้เห็นถึงการทำงานในโลกแห่งความเป็นจริง ซึ่งต้องเกี่ยวข้องกับวัตถุต่างๆ การนำมาประยุกต์ใช้งานช่วยให้ผู้พัฒนาระบบงานสามารถวิเคราะห์และออกแบบระบบงาน ได้อย่างมีประสิทธิภาพ
- การพัฒนาระบบงานโดยการใช้ Java Server Pages (JSP) ซึ่งใช้หลักการทำงานของภาษา JAVA และการใช้ Apache Tomcat โครงการ Jakarta เป็น Web Server ซึ่งเป็น การสนับสนุน Open Source อย่างหนึ่ง และทำให้โครงการพัฒนาระบบงานมีค่าใช้จ่ายต่ำ ซึ่งเป็นทางเลือกสำหรับองค์กรที่ต้องการประหยัดค่าใช้จ่ายด้าน IT

#### 7.2 ข้อเสนอแนะ

จากการทดลองใช้งานระบบที่พัฒนาขึ้นนั้น พบว่ายังคงต้องปรับปรุงแก้ไขในส่วนรายละเอียดการทำงานของระบบ เช่น Web Pages (หน้าจอ) การทำงานควรจะต้องปรับปรุงให้มีความน่าใช้งานมากขึ้น เป็นต้น

ข้อเสนอแนะด้านการลงทุน จะพบว่าระบบงานการจัดการข้อมูลสินค้าคงคลังแบบฝากขาย จะสามารถพัฒนาให้มีประสิทธิภาพมาก ถ้าแต่ละจุดขาย จะมีระบบการขาย (Point of Sales) และระบบการดูแลสินค้าคงคลังของตนเอง และเมื่อสิ้นสุดการขายในแต่ละวัน ก็จัดส่งข้อมูลผ่านสายโทรศัพท์มาที่สำนักงานใหญ่ ซึ่งผู้บริหาร ณ จุดศูนย์กลางจะได้ข้อมูลที่ทันสมัยมากที่สุด ซึ่งในอนาคตควรจะต้องปรับปรุงระบบงานให้อยู่ในแนวทางดังกล่าว เพื่อประโยชน์สูงสุดขององค์กร

## บรรณานุกรม

นรินทร์ โอพารกิจอนันต์, 2545. จาวา. กรุงเทพฯ : เคซีซี คอทเนท.

เจนวิทย์-ปิยวิทย์ เหลืองอร่าม, 2545. การพัฒนา Web Application. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

ศิวณัฐ มาศสุรางค์, 2545. เรียนรู้ JSP พร้อมใช้งาน. กรุงเทพฯ :

เอ.อาร์. อินฟอร์เมชัน แอนด์ พับลิเคชัน.

ชาติ วรกุลพิพัฒน์ และเทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2544. UML ภาษามาตรฐานเพื่อผู้พัฒนาซอฟต์แวร์. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

สุนทริน วงศ์ศิริกุล, 2545. พัฒนาโมเดลยุคใหม่ UML. กรุงเทพฯ : ชัคเชส มีเดีย.

Whitten, J. and Bentley, L. 2001. **SYSTEM ANALYSIS AND DESIGN METHODS.**

New York: McGraw-Hill.

Scott McPherson, 2000. **A Developer's Perspective.** [Online]. Available :

<http://developer.java.sun.com/developer/technicalArticles/Programming/jsp/dper.html>

## ประวัติผู้เขียน

### ผู้เขียน

นายปกครอง รอดบุญ

### วันเดือนปีเกิด

7 ตุลาคม พ.ศ. 2514

### การศึกษา

2526-2532

มัธยมศึกษา

โรงเรียนมัธยมวัดบึงทองหลาง

2532-2538

วิทยาศาสตร์บัณฑิต (ศาสตร์คอมพิวเตอร์)

มหาวิทยาลัยรามคำแหง

### การทำงาน :

2538-2539

กองคอมพิวเตอร์ สำนักนโยบายและแผน กรุงเทพมหานคร

2539-2542

บริษัท ซีดีจี ซีเอสเต็ม จำกัด

2542-ปัจจุบัน

บริษัท เอฟ บี ที สปอร์ต คอมเพล็กซ์ จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้