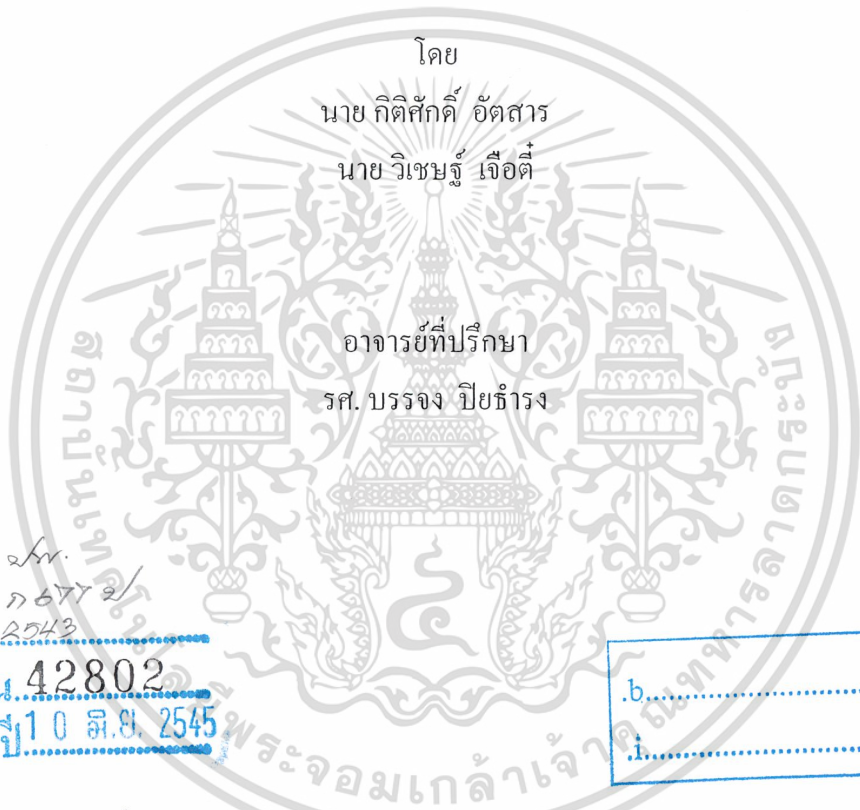


โปรแกรมประยุกต์แบบไคลเอนต์/เซิร์ฟเวอร์ 1  
CLIENT/SERVER APPLICATION DELVELOPMENT 1



โดย  
นาย กิตติศักดิ์ อัดสาร  
นาย วิเชษฐ์ เจือดี  
อาจารย์ที่ปรึกษา  
รศ. บรรจง ปิยะธำรง

5

๔พ.  
๓๖๗๔/  
๒๕๔๓  
เลขที่.....  
เลขทะเบียน 42802  
วัน, เดือน, ปี 10 ส.ย. 2545

.b.....  
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมประยุกต์แบบไคลเอนต์/เซิร์ฟเวอร์ 1

CLIENT/SERVER APPLICATION DELVELOPMENT 1

ผู้จัดทำ

1. นาย กิตติศักดิ์ อุตตสาร รหัสประจำตัว 41013521

2. นาย วิเศษฐ์ เจือดี รหัสประจำตัว 41013549



พจน

(รศ. บรรจง ปิยะธำรง)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมประยุกต์แบบไคลเอนต์/เซิร์ฟเวอร์ 1

นาย กิติศักดิ์ อัครสาร 41013521

นาย วิเชษฐ์ เจือดี 41013549

รศ. บรรจง ปิยธำรง อาจารย์ที่ปรึกษา  
ปีการศึกษา 2543

### บทคัดย่อ

ในปัจจุบันนี้เทคโนโลยีทางการสื่อสารข้อมูลทางอินเทอร์เน็ต มีการนำมาใช้ประโยชน์ทางด้านธุรกิจการค้าขายสินค้าต่างๆ กันอย่างมากเนื่องจากการค้าบนอินเทอร์เน็ต มีความสะดวกและรวดเร็ว นอกจากนี้ การค้าบนอินเทอร์เน็ต ยังสามารถหาลูกค้าได้ทั่วโลก

วิทยานิพนธ์ฉบับนี้ได้นำเอาเทคโนโลยีอินเทอร์เน็ตกับการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Object-Oriented Analysis and Design) มารวมกันทำให้เกิดเป็นระบบพาณิชย์อิเล็กทรอนิกส์ วิธีการที่ใช้ในการวิเคราะห์และออกแบบระบบเชิงวัตถุก็คือภาษา UML (Unified Modeling Language) เพราะว่าเป็นภาษาที่อธิบายการทำงานของระบบต่างออกมาได้เป็นอย่างดีทำให้การวิเคราะห์และออกแบบเชิงวัตถุ เป็นไปอย่างมีประสิทธิภาพ โดยโครงการนี้ได้นำมาประยุกต์สร้างร้านเช่าการ์ตูนบนอินเทอร์เน็ต ซึ่งลูกค้าสามารถเข้ามาเช่า และ อ่านหนังสือการ์ตูนได้จากหน้าจอคอมพิวเตอร์ของลูกค้าเอง นอกจากนี้ร้านเช่าการ์ตูนบนอินเทอร์เน็ต ได้นำเอาหลักการของระบบ Client/Server แบบ 3--tier มาประยุกต์ใช้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Client/Server Application Development 1

Kitisak Auttasam

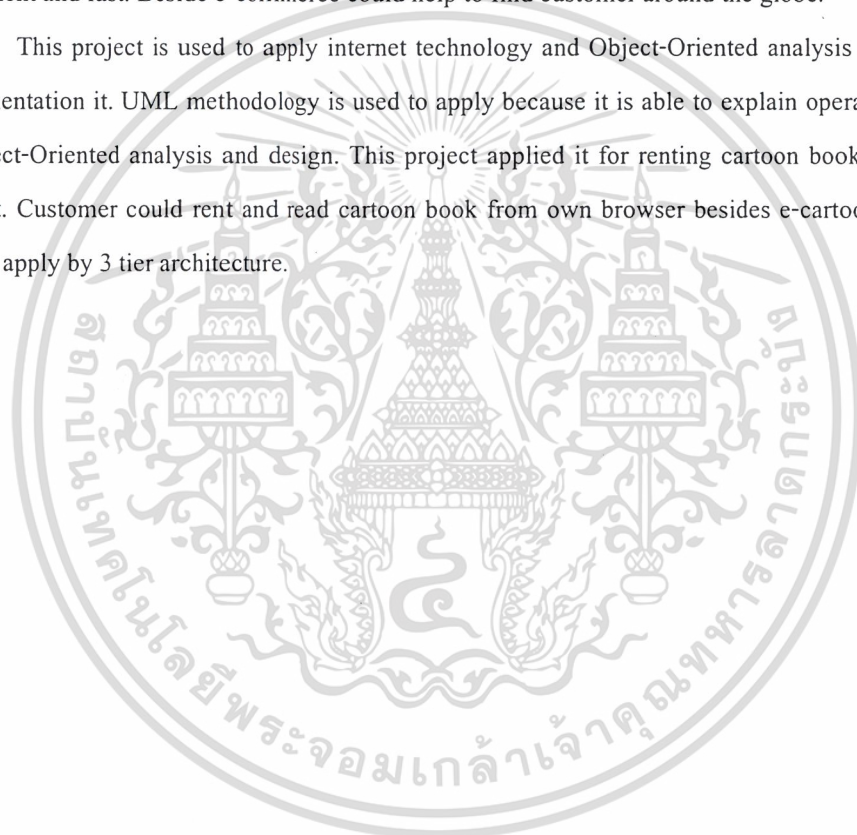
Vichet Juatee

Assoc.Prof. Bunjong Piyathumrong Advisor

### ABSTRACT

Recently, internet technology is used to apply in many e-business because e-commerce is convenient and fast. Beside e-commerce could help to find customer around the globe.

This project is used to apply internet technology and Object-Oriented analysis and design for implementation it. UML methodology is used to apply because it is able to explain operation of system in Object-Oriented analysis and design. This project applied it for renting cartoon book that based on internet. Customer could rent and read cartoon book from own browser besides e-cartoon shop also is used to apply by 3 tier architecture.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์บรรจง ปิยธำรง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

กิตติศักดิ์ อุตตสาร

วิเศษฐ์ เจือดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีและความรู้พื้นฐาน	4
2.1 UML(Unified Modeling Language)	4
2.1.1 ประโยคปัญหา(Problem Statement)	4
2.1.2 Use Case Model	4
2.1.3 ส่วนติดต่อผู้ใช้(User Interface)	5
2.1.4 Object Model	5
2.1.4.1 ออบเจกต์ (Object)	6
2.1.4.2 คลาส (Class)	6
2.1.4.3 สเตอริโอไทป์ (Stereotype)	7
2.1.4.4 แพ็กเกจ (Packages)	7
2.1.4.5 ความสัมพันธ์	7
2.1.4.5.1 Generalization	7
2.1.4.5.2 Aggregation	8
2.1.4.5.3 Association	8
2.1.5 Dynamic Model	8
2.1.5.1 Sequence Diagram	9
2.1.5.2 Collaboration Diagram	9
2.1.5.3 State Diagram	10
2.1.6 การวิเคราะห์ระบบซึ่งทำงานบนเว็บ	11
2.1.6.1 Robus Diagram	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้าที่
2.1.6.1.1 Boundary objects	11
2.1.6.1.2 Entity objects	12
2.1.6.1.3 Control objects	12
2.1.6.2 Overview Class Diagram	12
2.1.6.2.1 Server Page	12
2.1.6.2.2 Client Page	13
2.1.6.2.3 Frameset	13
2.1.6.2.4 Form	13
2.1.6.2.5 Target	14
2.2 Jservlet	14
2.3 เจดีบีซี	16
2.3.1 การทำงานของเจดีบีซี	16
2.3.2 JDBC-ODBC Bridge	17
2.4 Applet	18
2.4.1 Applet Life Cycle	19
2.4.2 Applet Tag	22
2.5 การแปลคลาสเป็นตาราง	23
บทที่ 3 การวิเคราะห์และออกแบบระบบ	25
3.1 การวิเคราะห์ระบบ	25
3.1.1 ความต้องการของระบบ	25
3.1.2 แบบจำลอง USE CASE	26
3.1.3 Sequence Diagram ขั้นตอนวิเคราะห์ระบบ	27
3.1.3.1 Sequence Diagram แสดงการทำงานของกรซื้อ Package	28
3.1.3.2 Sequence Diagram แสดงการทำงานของกรดูรายละเอียด	29
3.1.3.3 Sequence Diagram แสดงการทำงานของกรเพิ่มหนังสือลงตะกร้า	30
3.1.3.4 Sequence Diagram แสดงการทำงานของกร CheckOut	31
3.1.3.5 Sequence Diagram แสดงการทำงานของกรอ่านหนังสือการ์ตูน	32
3.1.3.6 Sequence Diagram การ Login ของผู้ดูแลระบบ	33
3.1.3.7 Sequence Diagram การลบ User ของผู้ดูแลระบบ	33
3.1.3.8 Sequence Diagram การเพิ่มหนังสือของผู้ดูแลระบบ	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

หน้าที่

3.1.4 แผนภาพวัตถุ (Object Diagram)	35
3.2 การออกแบบระบบ	36
3.2.1 Sequence Diagram ขั้นตอนออกแบบระบบ	36
3.2.1.1 Sequence Diagram การอ่านรายละเอียดของหนังสือการ์ตูน	36
3.2.1.2 Sequence Diagram การเพิ่มหนังสือลงตะกร้า	37
3.2.1.3 Sequence Diagram การ Checkout	38
3.2.1.4 Sequence Diagram การ Search	39
3.2.1.5 Sequence Diagram การซื้อ Package ใหม่	40
3.2.1.6 Sequence Diagram การเข้ามาอ่านของลูกค้า	41
3.2.1.7 Sequence Diagram การลบ User ของผู้ดูแลระบบ	42
3.2.2 Overview Class Diagram	43
3.2.3 Class Diagram	44
3.2.3.1 Class Diagram ในส่วนของการเช่า	44
3.2.3.2 Class Diagram ส่วนของการ Search	45
3.2.3.3 Class Diagram ส่วนของการซื้อ Package ใหม่	45
3.2.3.4 Class Diagram ส่วนของการ อ่านของลูกค้า	46
3.2.3.5 Class Diagram ในส่วนของผู้ดูแลระบบ	47
3.2.3.6 ความสัมพันธ์ของ Class ที่เป็น Entity Class	48
3.2.4 การแปลงคลาสเป็นตาราง	49
3.2.5 การออกแบบหน้าจอ	51
3.2.6 หน้าจอของระบบจริง	52
3.2.6.1 หน้าจอ Home	52
3.2.6.2. หน้าจอการดูรายละเอียดการ์ตูน	53
3.2.6.3 หน้าจอ Basket	53
3.2.6.4 หน้าจออ่าน	54
3.2.6.5 หน้าจอการซื้อ Package ใหม่	56
3.2.6.6 หน้าจอของผู้ดูแลระบบ	57
บทที่ 4 สรุป	58
4.1 ปัญหาต่างๆที่เกิดขึ้นในการพัฒนาระบบ	58
4.2 เปรียบเทียบกับการพัฒนาระบบในแนวทางเดิม	58
4.3 แนวทางในการพัฒนาระบบ	58

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้าที่
ภาคผนวก	60
ภาคผนวก ก การสร้าง JServlet Application และ Cartridge	60
ภาคผนวก ข การใช้งาน Oracle8I	65
ภาคผนวก ค การจัดการ Libraries ใน JBuilder เพื่อใช้เขียน JAVAServlets	68
ภาคผนวก ง ตัวอย่าง Source Code JAVAServlets	70
ภาคผนวก จ การติดตั้ง OAS (Oracle Application Server 4.0)	77
ภาคผนวก ฉ JavaScript	84
ภาคผนวก ช ปัญหาต่างๆ และ คำแนะนำ	90
บรรณานุกรม	93



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญภาพ

	หน้าที่
รูปที่ 2-1 USE CASE	5
รูปที่ 2-2 สัญลักษณ์แสดงออบเจกต์	6
รูปที่ 2-3 สัญลักษณ์แสดงคลาส	6
รูปที่ 2-4 ความสัมพันธ์แบบ Generalization	7
รูปที่ 2-5 ความสัมพันธ์แบบ Aggregation	8
รูปที่ 2-6 Association	8
รูปที่ 2-7 Sequence Diagram	8
รูปที่ 2-8 Collaboration Diagram	10
รูปที่ 2-9 State Diagram	10
รูปที่ 2-10 รูปแบบของสถานะ	11
รูปที่ 2-11 แสดง Boundary objects	12
รูปที่ 2-12 แสดง Entity objects	12
รูปที่ 2-13 แสดง Control objects	12
รูปที่ 2-14 Server Page	13
รูปที่ 2-15 Client Page	13
รูปที่ 2-16 Frameset	13
รูปที่ 2-17 Form	14
รูปที่ 2-18 Target	14
รูปที่ 2-19 การทำงานของ JServlet	15
รูปที่ 2-20 รูปแบบที่ใช้ในเจดีพีซี	16
รูปที่ 2-21 สถานะของ Applet life Cycle	19
รูปที่ 2-22 แสดงขั้นตอนที่เกิดขึ้น เมื่อ Applet ทำงาน	21
รูปที่ 2-23 วิธีที่แนะนำในการแปลง many – to – many association	23
รูปที่ 2-24 วิธีที่แนะนำในการแปลง one – to – many association	24
รูปที่ 2-25 วิธีที่เป็นทางเลือกในการแปลง one – to – many association	24
รูปที่ 3-1 USE CASE Diagram ของระบบเช่าการ์ตูนบนอินเทอร์เน็ต	26
รูปที่ 3-2 Sequence Diagram การซื้อ Package ด้วยบัตรเครดิต	28
รูปที่ 3-3 Sequence Diagram การการดูรายละเอียดของหนังสือการ์ตูน	29
รูปที่ 3-4 Sequence Diagram แสดงการทำงานของการทำงานเพิ่มหนังสือลงตะกร้า	30
รูปที่ 3-5 Sequence Diagram แสดงการทำงานของการทำงานการCheckOut	31
รูปที่ 3-6 Sequence Diagram การอ่านหนังสือการ์ตูน	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ(ต่อ)

	หน้าที่
รูปที่ 3-7 Sequence Diagram การ Login ของผู้ดูแลระบบ	33
รูปที่ 3-8 Sequence Diagram การลบ User ของผู้ดูแลระบบ	33
รูปที่ 3-9 Sequence Diagram การเพิ่มหนังสือของผู้ดูแลระบบ	34
รูปที่ 3-10 แผนภาพวัตถุ	35
รูปที่ 3-11 แสดง Sequence Diagram ของการอ่านรายละเอียด	36
รูปที่ 3-12 แสดง Sequence Diagram ของการเพิ่มหนังสือลงตะกร้า	37
รูปที่ 3-13 แสดง Sequence Diagram ของ การ Checkout	38
รูปที่ 3-14 แสดง Sequence Diagram ของการ Search	39
รูปที่ 3-15 แสดง Sequence Diagram ของ การซื้อ Package ใหม่	40
รูปที่ 3-16 แสดง Sequence Diagram การการเข้ามาอ่านของลูกค้า	41
รูปที่ 3-17 Sequence Diagram การลบ User ของผู้ดูแลระบบ	42
รูปที่ 3-18 แสดง Overview Class Diagram	43
รูปที่ 3-19 Class Diagram ในส่วนของการเช่า	44
รูปที่ 3-20 แสดง Class Diagram ส่วนของการ Search	45
รูปที่ 3-21 แสดง Class Diagram ส่วนของการซื้อ Package ใหม่	45
รูปที่ 3-22 แสดง Class Diagram ส่วนของการ อ่านของลูกค้า	46
รูปที่ 3-23 แสดง Class Diagram ในส่วนของผู้ดูแลระบบ	47
รูปที่ 3-24 ความสัมพันธ์ของ Class ที่เป็น Entity Class	48
รูปที่ 3-25 แสดงองค์ประกอบของหน้าจอหลัก	51
รูปที่ 3-26 แสดงหน้าจอ Home	52
รูปที่ 3-27 แสดงหน้าจอรายละเอียดการ์ตูน	53
รูปที่ 3-28 แสดงหน้าจอ Basket	54
รูปที่ 3-29 แสดงการ Login เข้ามาอ่าน	54
รูปที่ 3-30 แสดงรายการเช่าและเวลา Expire	55
รูปที่ 3-31 แสดงหน้าจอการอ่าน	55
รูปที่ 3-32 แสดงหน้าจอการซื้อ Package ใหม่	56
รูปที่ 3-33 แสดงหน้าจอผู้ดูแลระบบ	57

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

เนื่องจากปัจจุบันเป็นยุคของข่าวสาร และ ข้อมูลที่รู้ข่าวสารและข้อมูลได้รวดเร็วย่อมหา ยถึงความได้เปรียบคู่แข่ง อินเทอร์เน็ตนับได้ว่าเป็นปัจจัยหนึ่งที่มีความสำคัญต่อการดำรงชีวิตของเราใน ปัจจุบันเป็นอย่างมาก ใครไม่รู้จักอินเทอร์เน็ตนั้นย่อมจะเสียเปรียบคู่แข่งอย่างแน่นอน และในอนาคตที่ยัง ใช้บริการทางอินเทอร์เน็ตเพิ่มขึ้นอย่างมากมา ประเทศไทยของเราก็กำลังออกกฎหมายการค้าพาณิชย์ อิเล็กทรอนิกส์ เหตุผลก็เพื่อจะก้าวให้ทันกับอารยประเทศที่มีความเจริญก้าวหน้าทางเทคโนโลยี จุด ประสงค์หลักก็เพื่อทำการค้าขายกันทางอินเทอร์เน็ตนั่นเอง

E-Commerce หรือ การค้าขายบนอินเทอร์เน็ตนับว่าเป็นที่รู้จักกันพอสมควรในบ้านเรา แต่ก็ยัง ต้องการการพัฒนาให้ก้าวหน้าต่อไปอีกในด้านต่าง ๆ เช่น ความปลอดภัยบนอินเทอร์เน็ตซึ่งต้องให้ความสำคัญ อย่างมาก และเรื่องอื่น ๆ ที่ต้องการพัฒนาต่อไป

การออกแบบ และ การพัฒนาระบบ E-Commerce นั้นเหมาะสมที่จะพัฒนาในรูปแบบเชิงวัตถุ เพราะเป็นระบบที่มีการเปลี่ยนแปลงอยู่เสมอ การพัฒนาเชิงวัตถุ (Object – Oriented Development) เป็น แนวทางในการพัฒนาที่ได้รับความนิยมอย่างแพร่หลายในปัจจุบัน วิธีนี้ผู้พัฒนาจะมองระบบเป็น ออบเจกต์ (Object) โดยใช้นุ่มมองจากโลกแห่งความเป็นจริงทำให้สามารถเข้าใจในตัวระบบที่ทำการ พัฒนาได้ชัดเจนยิ่งขึ้น นอกจากนี้การพัฒนาเชิงวัตถุยังมีข้อดีเหนือแนวทางการพัฒนารูปแบบอื่นดังนี้

- โมเดลมีความสอดคล้องกันในทุกขั้นตอนการพัฒนา ทั้งในขั้นตอนการวิเคราะห์,การออกแบบไปจน ถึงการสร้าง โมเดลที่ได้จากขั้นตอนการวิเคราะห์สามารถนำมาเขียนเป็นโค้ดได้โดยตรง จึงช่วยให้ผู้ พัฒนาเห็นถึงความสัมพันธ์ระหว่างปัญหาท้กับวิธีการแก้ปัญหาได้ง่าย
- โมเดลเชิงวัตถุมีการแยกแยะเอกลักษณ์ (Abstraction) โดยเป็นการแยกเอาเฉพาะสิ่งที่สนใจออกมา แสดงคุณลักษณะที่สำคัญของออบเจกต์ และการซ่อนรายละเอียด (Encapsulation) ซึ่งเป็นการซ่อนข้อมูลและความซับซ้อนภายในออบเจกต์ เป็นลักษณะที่ดีในการพัฒนาซอฟต์แวร์
- เมื่อมีการเปลี่ยนแปลงความต้องการของระบบ ก็เป็นเพียงการเปลี่ยนแปลงภายในออบเจกต์หรือการ เพิ่มออบเจกต์ โดยไม่ต้องเปลี่ยนแปลงโครงสร้างของระบบทั้งหมด
- สนับสนุนการนำกลับมาใช้ใหม่ (Reuse)
- สนับสนุนการปรับเปลี่ยนขนาดของระบบ เนื่องจากมีการแยกแยะเอกลักษณ์และซ่อนรายละเอียด ของแต่ละออบเจกต์
- สนับสนุนการออกแบบระบบที่เชื่อถือได้และปลอดภัย เนื่องจากออบเจกต์แต่ละตัวสื่อสารกันผ่าน ส่วนติดต่อที่ผู้พัฒนากำหนดให้ไว้เท่านั้นจึงสามารถควบคุมการติดต่อระหว่างคอมโพเนนท์ได้
- สนับสนุนการทำงานพร้อมๆกัน (Concurrency)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการพัฒนาเชิงวัตถุประกอบไปด้วยขั้นตอนต่างๆ คือขั้นตอนการวิเคราะห์และออกแบบ โดยจะมองภาพรวมของฟังก์ชันการทำงานของระบบเป็นหลัก จากนั้นก็จะมากำหนดรายละเอียดของออบเจกต์, คลาส, ความสัมพันธ์และคุณลักษณะที่จะมีในระบบ และจะเข้าสู่ขั้นตอนการพัฒนาโดยใช้ภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object – Oriented Programming Language) ซึ่งเป็นภาษาที่สนับสนุนการสร้างคลาสและออบเจกต์ ทำให้สิ่งที่ได้จากการวิเคราะห์และออกแบบสามารถนำมาใช้ในการเขียนโปรแกรมได้เกือบจะทันที

## 1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาและประยุกต์ใช้ระบบ Client/Server แบบ 3-เทียร์ เพื่อสร้างระบบพาณิชย์อิเล็กทรอนิกส์
2. ศึกษาการวิเคราะห์และออกแบบเชิง Object โดยใช้ UML
3. ศึกษาการออกแบบและสร้างระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ (ORDBMS) โดยใช้ Oracle8i

## 1.3 ขอบเขตของโครงการ

โครงการนี้ได้ศึกษาถึง ความสามารถต่าง ๆ ของระบบฐานข้อมูลแบบ Relational ร่วมกับความสามารถของภาษา JAVA เพื่อนำมาใช้ประโยชน์ กับระบบงานที่มีการทำงานกับข้อมูลแบบมัลติมีเดีย บน Internet และ นอกจากนี้ยังมีการใช้ Application Server ซึ่งเป็นการใช้ระบบ Client/Server แบบ 3-tier มีการใช้ Servlet ซึ่งเป็น Service ตัวหนึ่งที่อยู่ใน Application Server และมีการเขียน โปรแกรมขึ้นมาภายใน Servlet เพื่อเป็นตัวจัดการระบบงานทั้งหมดที่มีใน ซึ่งมีอยู่ใน Application Server ทั้งหมดนี้จะถูกวิเคราะห์ และ ออกแบบบนพื้นฐานของ Object-Oriented โดยใช้ UML ในการวิเคราะห์ และ ออกแบบ

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ความเข้าใจถึงทฤษฎีต่างๆเกี่ยวกับ Object Oriented
2. ได้รับความรู้และประสบการณ์ในการเขียน โปรแกรมด้วยภาษาจาวา
3. ทราบถึงวิธีการเขียนโปรแกรมติดต่อระหว่างเซิร์ฟเวอร์ กับ ไคลเอนต์
4. ได้ทราบถึงปัญหาที่เกิดขึ้นในการปฏิบัติงานจริง และแนวทางในการแก้ไขปัญหาต่างๆ

## 1.5 วิธีการดำเนินงาน

โครงการนี้เริ่มทำการศึกษาถึงหลักการ และ แนวคิดของ Object-Oriented อันจะประกอบไปด้วย การวิเคราะห์และออกแบบในเชิงวัตถุ โดยศึกษา และ ทดลองสร้างจาก UML และ ศึกษาถึงฐานข้อมูลของ Oracle ที่ได้รวมความสามารถของ JAVA เข้าไปด้วยเป็น Oracle8i และ นอกจากนี้ก็ศึกษาถึง Oracle Application Server พร้อมกับลองใช้งานจริงซึ่งประกอบด้วย Servlet และ ระบบรักษาความปลอดภัยบน Application Server

1. ทำการศึกษาถึงทฤษฎี และแนวความคิดของการเขียนโปรแกรมในเชิงวัตถุ
2. ทำการศึกษา และทำความเข้าใจในภาษาจาวา
3. ศึกษา และ ทดลองใช้ tools ต่างๆทั้งในการออกแบบ และ การเขียนโปรแกรม
4. กำหนดหัวข้อของโปรแกรมประยุกต์
5. กำหนดขอบเขตของการทำงานของโปรแกรมประยุกต์
6. ทำการวิเคราะห์และออกแบบโปรแกรม ทั้งในด้านการทำงานและการออกแบบหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนสิทธิ์ในเนื้อหา หากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูง และขอแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทำการเขียนโปรแกรมในส่วนต่างๆ
8. ทดลองใช้ เพื่อหาข้อบกพร่อง และ ทำการแก้ไขข้อบกพร่องนั้นๆ
9. รวบรวมข้อมูลและเอกสารต่างๆ เพื่อเรียบเรียงเป็นปฏิญญานิพนธ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและความรู้พื้นฐาน

### 2.1 UML(Unified Modeling Language)

ขั้นตอนการวิเคราะห์เป็นขั้นตอนหนึ่งของการพัฒนาระบบ ผลลัพธ์ที่ได้จากขั้นตอนนี้คือการอธิบายว่าระบบถูกสร้างขึ้นมาเพื่อทำอะไร ไม่ได้เป็นการระบุว่าระบบควรจะทำอย่างไร ในเทคนิคเชิงวัตถุ ผลลัพธ์จากการวิเคราะห์นี้คือโมเดล ซึ่งโดยปกติจะเรียกขั้นตอนนี้ว่า Modeling Phase

ขั้นตอนของการวิเคราะห์มีดังนี้

1. กำหนดประโยคปัญหา (Problem Statement) ของระบบ
2. กำหนดผู้กระทำ (Actor) และ Use Case จากประโยคปัญหา
3. ร่างภาพส่วนติดต่อผู้ใช้ของหน้าที่พื้นฐานที่ผู้ใช้ติดต่อกับระบบ
4. สร้าง Object Model
5. สร้าง Dynamic Model

#### 2.1.1 ประโยคปัญหา (Problem Statement)

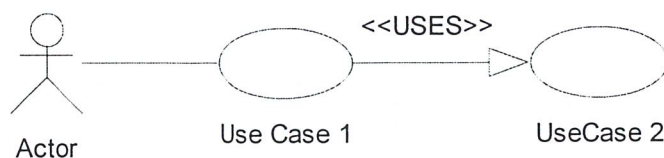
เป็นบทความที่บรรยายถึงลักษณะปัญหาของระบบ หน้าที่การทำงานต่างๆ ที่ระบบต้องมีความสามารถในการติดต่อกันระหว่างผู้ใช้กับระบบ ขอบเขตการทำงานของระบบและรายละเอียดหน้าที่การทำงานที่มีในระบบ

#### 2.1.2 Use Case Model

Use case model ประกอบด้วยผู้กระทำ (Actor) และ Use Case ผู้กระทำจะอยู่นอกระบบ แสดงถึงบุคคลหรือสิ่งของที่มีความสัมพันธ์กับระบบ โดยผู้กระทำอาจจะ

- ให้ข้อมูลแก่ระบบเพียงอย่างเดียว
- รับข้อมูลจากระบบเพียงอย่างเดียว
- ทั้งให้และรับข้อมูลกับระบบ

ผู้กระทำอาจจะพบได้ในประโยคปัญหาและการสนทนาระหว่างลูกค้ากับผู้ดูแลระบบ



รูปที่ 2-1 USES CASE

Use cases คือ บทความสนทนาระหว่างผู้กระทำกับระบบ ซึ่งแสดงถึงหน้าที่การทำงานที่ถูกจัดเตรียมขึ้นโดยระบบ การกำหนด Use Case ควรจะแสดงถึงงานหลักๆที่เกิดขึ้นตั้งแต่ต้นจนจบและต้องมีการส่งผลบางอย่างให้แก่ผู้กระทำ

ความสัมพันธ์ระหว่าง Use Case มีอยู่ 2 แบบ คือ *uses* และ *extends* ความสัมพันธ์แบบ *uses* เกิดจาก Use Case A ถูกเรียกใช้จาก Use Case B และ C นั่นคือ Use Case B และ C *uses* Use Case A

ส่วนความสัมพันธ์แบบ *extends* เกิดจาก Use Case B ถูกสร้างเริ่มต้นมาจาก Use Case A และเพิ่มรายละเอียดบางส่วน เราเรียกว่า Use Case B *extends* Use Case A

### 2.1.3 ส่วนติดต่อผู้ใช้ (User Interface)

เพื่อความเข้าใจความต้องการของผู้ใช้ในการทำงานโปรแกรม จึงควรมีการออกแบบส่วนการติดต่อระหว่างผู้ใช้กับระบบ โดยอาจจะเป็นการสร้างส่วนติดต่อของโปรแกรมในลักษณะคร่าวๆเพื่อให้ผู้ใช้สามารถเห็นภาพของการทำงานโปรแกรมได้ดีขึ้น ซึ่งการสร้างส่วนติดต่อผู้ใช้นี้จะช่วยให้ผู้พัฒนาสามารถแสดงลักษณะการติดต่อภายใน Use Case ได้และเป็นการช่วยในการออกแบบโปรแกรมในขั้นตอนการออกแบบ

### 2.1.4 Object Model

Object Model เป็นแบบจำลองที่สำคัญที่ได้จากขั้นตอนการวิเคราะห์ระบบและเป็นจุดเริ่มต้นในส่วนของการออกแบบและการสร้างระบบ ในโครงงานนี้เราจะสร้าง Object Model ตามรูปแบบของ UML ขั้นตอนการสร้าง Object Model มีดังนี้

1. กำหนดออบเจกต์ที่เป็นไปได้ทั้งหมดของระบบจากประโยคปัญหาและ Use Case
2. กำจัดออบเจกต์ที่ไม่ตรงกับปัญหาหรือมีความหมายที่ซ้ำกัน
3. จัดรวมออบเจกต์ที่มีลักษณะเหมือนกันเข้าเป็นคลาส
4. กำหนดแอททริบิวต์และโอเปอเรชันของแต่ละคลาส
5. กำหนดความสัมพันธ์ระหว่างออบเจกต์ของคลาส ซึ่งมีลักษณะความสัมพันธ์ได้แก่

Generalization , Association และ Aggregation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.4.1 ออบเจกต์ (Object)

ออบเจกต์เป็นกลุ่มก้อนของเอนทิตี (Entity) ที่มีคุณสมบัติ, พฤติกรรมและสถานะ ออบเจกต์แสดงถึงบางอย่างที่มีตัวตน เช่น รถยนต์, เครื่องคอมพิวเตอร์ หรือแสดงถึงสิ่งที่มีตัวตนในความคิด เช่น บัญชีธนาคาร, เครื่องหมายการค้า, พืชแต่งงานหรือรายการสินค้า

ObjectName:ClassName

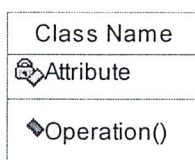
รูปที่ 2-2 สัญลักษณ์แสดงออบเจกต์

สิ่งที่เราสนใจในตัวออบเจกต์ประกอบไปด้วย

- แอททริบิวต์ (Attributes) เป็นข้อมูลที่แสดงคุณลักษณะของวัตถุ เช่น รถยนต์ จะมีแอททริบิวต์คือ ความกว้าง, ความยาว, สี, น้ำหนัก
- โอเปอเรชัน (Operations) คือการกระทำที่วัตถุนั้นสามารถกระทำได้ เช่น รถยนต์ จะมี operations คือ ออกตัว, เบรก, เลี้ยวขวา, เลี้ยวซ้าย
- สถานะ (State) สถานะของวัตถุคือหนึ่งในเงื่อนไขที่เป็นไปได้และอาจเกิดขึ้นได้จริง เช่น รถยนต์ จะมี state คือ กำลังวิ่ง, กำลังชะลอ
- เอกลักษณ์ (Identity) เป็นสิ่งที่ระบุว่าวัตถุนั้นมีอยู่เพียงหนึ่งเดียว เช่น รถยนต์มีเอกลักษณ์คือ ทะเบียนรถ
- หน้าที่ (Responsibility) คือ หน้าที่และบทบาทของวัตถุต่อระบบ

### 2.1.4.2 คลาส (Class)

คลาส คือการอธิบายถึงกลุ่มของออบเจกต์ซึ่งจะมีคุณสมบัติทั่วไป (attributes), พฤติกรรมทั่วไป (operations) และมีความสัมพันธ์ทั่วไปกับออบเจกต์อื่นๆ ในลักษณะเดียวกัน ดังนั้นคลาสจึงเป็นรูปแบบที่ใช้สร้างออบเจกต์



รูปที่ 2-3 สัญลักษณ์แสดงคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.4.3 สเตอริโอไทป์ (StereoType )

คือ การสร้างองค์ประกอบชนิดใหม่ของแบบจำลองหรือการสร้างคลาสใหม่ ซึ่งองค์ประกอบชนิดใหม่จะสืบทอดคุณสมบัติขององค์ประกอบเดิมมาทุกอย่าง(รวมทั้งความสัมพันธ์ต่างๆที่ใช้กับองค์ประกอบนั้นๆ ได้) โดยมีจุดประสงค์การใช้งานที่ต่างออกไป

รูปแบบของสเตอริโอไทป์ใน UML จะใช้เครื่องหมาย << และ >> คร่อมชื่อสเตอริโอไทป์ เช่น << Entity >> แนวคิดสเตอริโอไทป์ของโครงการนี้จะแบ่งประเภทของคลาส เป็น เอนตีตี้คลาส (Entity Class) , คลาสขอบเขต (Boundary Class) และ คลาสควบคุม (Control Class)

- เอนตีตี้คลาส สิ่งที่มีอยู่จริง หรือการกระทำภายในระบบ
- คลาสขอบเขต กล่าวถึงการติดต่อระหว่างสิ่งที่อยู่รอบๆระบบกับภายในระบบ (การติดต่อไปยังผู้กระทำ)
- คลาสควบคุม ส่งผลให้เกิดพฤติกรรมเฉพาะใน Use Case หรือเรียกว่าเป็นการ “ running ” Use Case

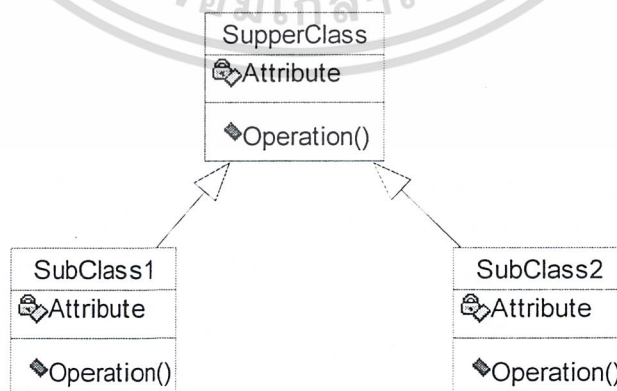
### 2.1.4.4 แพ็กเกจ (Packages)

เป็นการรวมคลาสที่มีความสัมพันธ์ใกล้ชิดกันเข้าไว้ในแพ็กเกจ เพื่อประโยชน์สำหรับระบบที่มีคลาสเป็นจำนวนมาก ในระบบที่มีความซับซ้อนเราอาจจะสร้างแพ็กเกจขึ้นมาก่อน แล้วจึงบรรจุคลาสและความสัมพันธ์เข้าไว้ในแพ็กเกจ

### 2.1.4.5 ความสัมพันธ์

#### 2.1.4.5.1 Generalization

เป็นความสัมพันธ์แบบการให้กำเนิดคลาสร้อย (Subclass) จากคลาสแม่ (Superclass) โดยคลาสร้อยจะสืบทอดแอททริบิวต์, โอเปอเรชันและความสัมพันธ์ทั้งหมดจากคลาสแม่ ทั้งของตัวเองและจากคลาสแม่ของคลาสแม่ขึ้นไปเรื่อยๆ คลาสร้อยอาจจะมีการเพิ่มแอททริบิวต์และ โอเปอเรชันในคลาสของมันเอง ความสัมพันธ์แบบนี้อาจถูกเรียกได้เป็น is-a หรือ kind-of



รูปที่ 2-4 ความสัมพันธ์แบบ Generalization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.5.2 Aggregation

เป็นความสัมพันธ์แบบเป็นส่วนประกอบ หรือที่เรียกว่า part-of คำว่าเป็นส่วนประกอบบางครั้งอาจมีความหมายที่ต่างกัน เช่น เครื่องยนต์และล้อเป็นส่วนประกอบของรถยนต์ในขณะที่เรากำลังขับรถอยู่ แต่ถ้ารถยนต์คันนั้นอยู่ในระหว่างการซ่อมที่อู่ซ่อมรถ นั่นคือเครื่องยนต์และล้อก็ไม่ได้เป็นส่วนประกอบของรถยนต์แล้ว หรืออาจเรียกอีกแบบหนึ่งว่ามีความสัมพันธ์แบบ has-a-relationship คือ class A จำเป็นต้องมีการเรียกใช้ class B ในการทำงาน ดังนั้น class A จะขาด class B ไม่ได้แต่ class B สามารถขาด class A ได้จึงเรียกว่า class B มีความสัมพันธ์เป็นแบบ Aggregation กับ class A



รูปที่ 2-5 ความสัมพันธ์แบบ Aggregation

2.1.4.5.3 Association

เป็นความสัมพันธ์แบบสองทิศทาง ซึ่งใช้ในการติดต่อกันระหว่างคลาส ความสัมพันธ์แบบ association ระหว่างคลาสหมายถึงมีการเชื่อมต่อระหว่างออบเจกต์ในคลาสทั้งสอง ที่ปลายของเส้นความสัมพันธ์จะมีการระบุชื่อบทบาทเพื่อบอกว่าคลาสนั้นถูกมองว่าเป็นอย่างไรจากคลาสนอื่น



รูปที่ 2-6 Association

2.1.5 Dynamic Model

เป็นแบบจำลองที่อธิบายถึงพฤติกรรมของระบบและลำดับของการทำงาน ซึ่งแบ่งเป็น

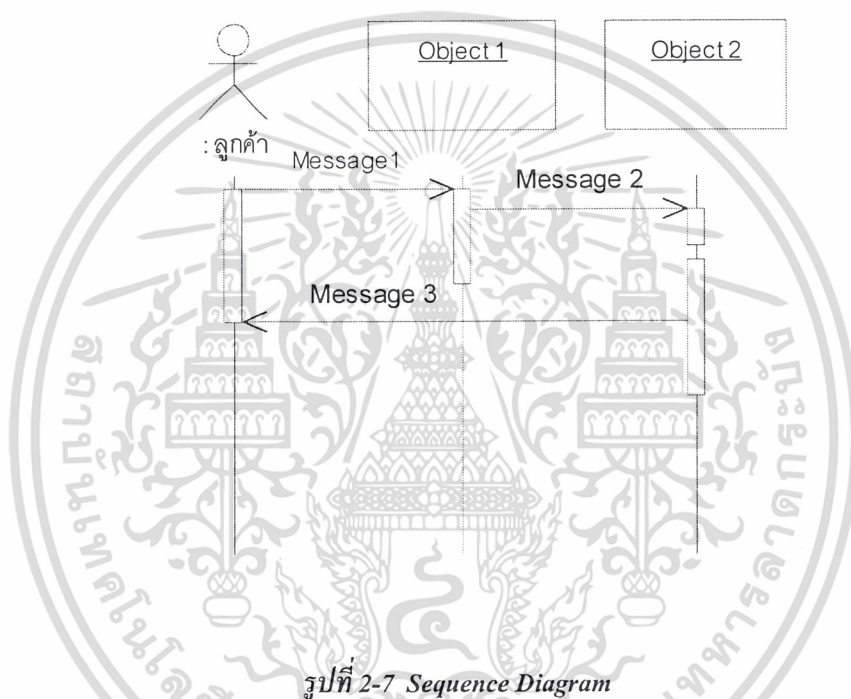
- 1. Sequence Diagram
- 2. Collaboration Diagram
- 3. State Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.5.1 Sequence Diagram

Sequence Diagram เป็นการแสดงข้อความและลำดับของการส่งข้อความระหว่างออบเจกต์ข้อความ คือ หน่วยที่ออบเจกต์ใช้ในการสื่อสารกัน ตัวอย่างของข้อความที่เกิดขึ้น เช่น เมื่อผู้ใช้ร้องขอบริการ , ออบเจกต์ร้องขอบริการจากออบเจกต์อื่น หรือออบเจกต์ส่งคืนข้อมูลจากบริการที่ถูกร้องขอ

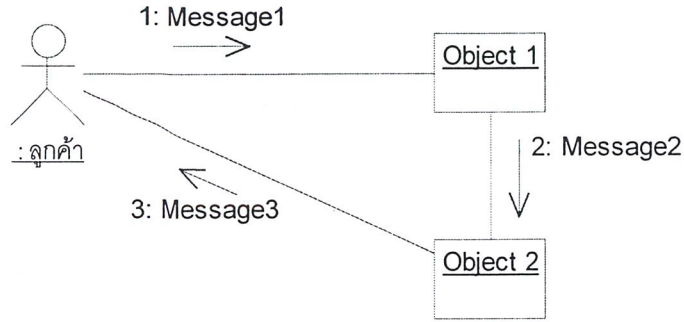
Sequence Diagram จะถูกพัฒนามาจาก Use Case โดยแสดงถึงลำดับของข้อความที่เกิดขึ้นของแต่ละ Use Case ตั้งแต่ต้นจนจบ Sequence Diagram มีจุดประสงค์เพื่อช่วยให้ผู้ใช้และผู้ออกแบบระบบสามารถเข้าใจการทำงานของระบบได้ง่ายขึ้น



จากรูปไดอะแกรมข้างต้น เส้นในแนวตั้งจะหมายถึงออบเจกต์โดยมีชื่อของออบเจกต์เขียนอยู่ด้านบนหรือเป็นผู้กระทำที่ติดต่อกับระบบ ส่วนเส้นในแนวนอนนั้นจะแสดงถึงข้อความซึ่งมีชื่อของข้อความนั้นอยู่ ข้อความที่เกิดขึ้นจะเป็นลำดับตามแนวตั้งจากบนลงล่าง

### 2.1.5.2 Collaboration Diagram

Collaboration Diagram จะแสดงข้อมูลพื้นฐานเช่นเดียวกับ Sequence Diagram แต่ความแตกต่างอยู่ที่ Sequence Diagram จะแสดงลำดับของข้อความเป็นหลัก ส่วน Collaboration Diagram จะแสดงข้อความและโครงสร้างของออบเจกต์ที่ทำงานร่วมกันเป็นหลัก



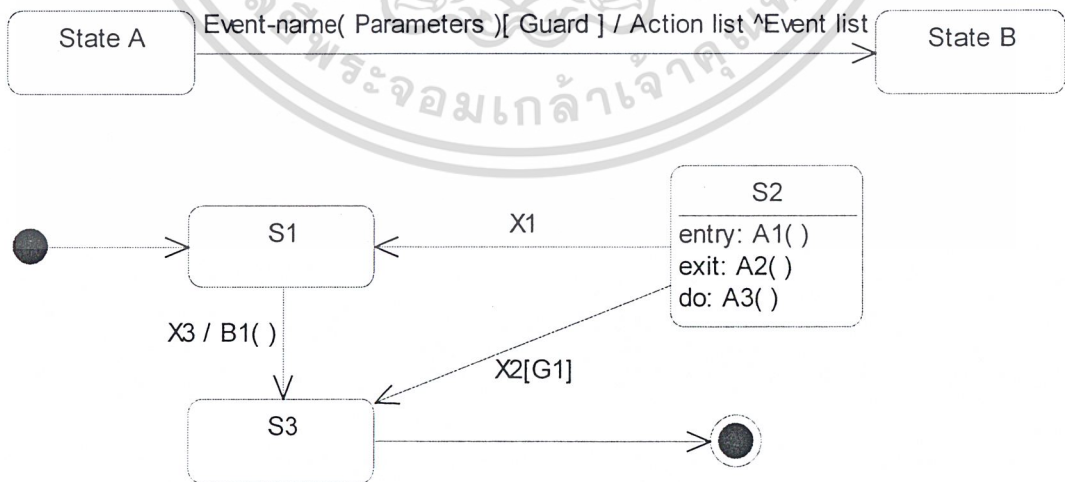
รูปที่ 2-8 Collaboration Diagram

จากรูปไดอะแกรมข้างต้น จะแสดงการไหลของข้อความระหว่างออบเจกต์ด้วยกันหรือระหว่างออบเจกต์กับผู้กระทำ โดยข้อความที่ใช้ในการติดต่อกันจะประกอบด้วยหมายเลขลำดับ, ชื่อข้อความและทิศทางของข้อความ

2.1.5.3 State Diagram

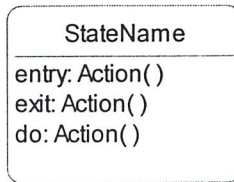
สถานะของออบเจกต์ปกติจะถูกกำหนดโดยค่าของแอททริบิวต์หรือค่าตัวแปรภายใน เช่น ออบเจกต์หลอดไฟจะมีสถานะคือเปิดและปิด ซึ่งสถานะเปิดและปิดนี้จะเป็ค่าของแอททริบิวต์สถานะของออบเจกต์หลอดไฟ

State Diagram แสดงถึงสถานะต่างๆของออบเจกต์และการส่งผ่าน (transition) ที่ทำให้เกิดการเปลี่ยนแปลงของสถานะเหล่านั้น และแสดงถึงกิจกรรมที่ออบเจกต์กระทำเมื่ออยู่ในสถานะหนึ่งๆ



รูปที่ 2-9 State Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-10 รูปแบบของสถานะ

State Diagram สามารถมีสถานะซ้อนภายในสถานะอื่นก็ได้ ตัวสถานะที่อยู่ภายนอกนั้นเราจะเรียกว่า Superstate ส่วนสถานะที่อยู่ภายในเราจะเรียกว่า Substate ในแต่ละสถานะจะมีการระบุถึงการทำงานที่ออกแบบเจ็คต์จะกระทำเมื่ออยู่ในสถานะนั้น โดยสามารถแบ่งการทำงานได้เป็น

- entry action                      จะทำงานเมื่อมีการเข้าสู่สถานะนั้น
- exit action                        จะทำงานเมื่อมีการออกจากสถานะนั้น
- do action                          จะทำงานเมื่อสถานะนั้นๆ แยกที่ฟ

รูปแบบของการส่งผ่านคือ Event-name( Parameters )[ Guard ] / Action list ^Event list โดยมีรายละเอียดดังนี้

Field	คำอธิบาย
Event-name	เป็นชื่อของ event ที่ทำให้เกิดการเปลี่ยนสถานะ
Parameters	เป็นรายการข้อมูลพารามิเตอร์ ที่ต้องการส่งไปพร้อมกับ event
Guard	เป็นเงื่อนไขในรูปของประโยคบูลีน ซึ่งจะต้องมีค่าเป็นจริงจึงจะทำให้เกิดการเปลี่ยนสถานะได้
Action list	เป็นรายการของ โอเปอเรชันที่จะกระทำเมื่อเกิดการเปลี่ยนสถานะโดย โอเปอเรชันเหล่านี้ อาจจะเป็นของออบเจ็คต์เองหรือของออบเจ็คต์อื่นก็ได้
Event list	เป็นรายการของ events ที่ถูกสร้างขึ้นเมื่อมีการเปลี่ยนสถานะ โดย events เหล่านี้สามารถกระจายไปยัง state machines อื่นๆ เพื่อเป็นการทำงานแบบพร้อมกันได้

ตารางที่ 2-1 รูปแบบการส่งผ่าน

## 2.1.6 การวิเคราะห์ระบบที่ทำงานบนเว็บ

### 2.1.6.1 Robus Digram

#### 2.1.6.1.1 Boundary objects แสดงการสื่อสารกันระหว่าง Actor และ ระบบ Instance ของ object จะ

แสดงหน้าจอกับอินเทอร์เฟซกับผู้ใช้ ในโปรแกรมประยุกต์บนเว็บก็คือเว็บเพจนั่นเอง  
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในโปรแกรมประยุกต์บนเว็บก็คือเว็บเพจนั่นเอง  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-11 แสดง Boundary objects

2.1.6.1.2 Entity objects ใช้ในการเก็บข้อมูล หรือ ติดต่อกับฐานข้อมูล



รูปที่ 2-12 แสดง Entity objects

2.1.6.1.3 Control objects เป็นออปเจกต์ที่แสดงการทำงาน และ ควบคุมการทำงานของออปเจกต์ตัวอื่น

รูปที่ 2-13 แสดง Control objects

การกำหนดการทำงาน หรือ ความสัมพันธ์มีกฎดังนี้

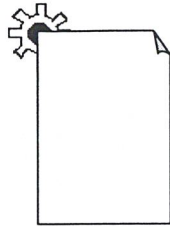
- Actor สามารถทำงานร่วมกับ Boundary object ได้เพียงอย่างเดียวเท่านั้น
- Entity object สามารถทำงานร่วมกับ Control object ได้เพียงอย่างเดียวเท่านั้น
- Control object สามารถทำงานร่วมกับ object ใดก็ได้

#### 2.1.6.2 Overview Class Diagram

ซึ่งจะใช้อธิบายคลาสไดอะแกรมอีกรูปแบบหนึ่งที่มีความเหมาะสมมากขึ้นในการออกแบบระบบที่ทำงานบนเว็บ โดยใช้ UML ซึ่งจะประกอบไปด้วย Stereo Types ดังนี้

##### 2.1.6.2.1 Server Page

จะใช้อธิบายการทำงานที่เกิดขึ้นทางฝั่ง Server โดยทั่วไปจะเป็นการเขียน Script ด้วยภาษาต่าง ๆ เช่น Perl , ASP เป็นต้น หรืออาจจะเป็น Servlet ก็ได้ซึ่งในที่นี้จะใช้แทน Servlet ที่ทำหน้าที่ต่าง ๆ เช่น การติดต่อกับฐานข้อมูล การพิมพ์ Tag ต่าง ๆ ไปยัง Client เป็นต้น ซึ่งจะสามารถแสดงรูปภาพแทนได้ดังรูปที่ 2-14



รูปที่ 2-14 Server Page

#### 2.1.6.2.2 Client Page

เป็นเว็บเพจที่อยู่ในรูปแบบของภาษา HTML และประกอบไปด้วยข้อมูลต่าง ๆ ที่ใช้แสดงในส่วน  
ของ Client ภายในเพจจะประกอบด้วย Frame, Form ต่างๆ หรือแม้กระทั่ง Script ต่าง ๆ ที่ใช้ทำงานบนเว็บ  
เพจ เช่น Java Script เป็นต้น



รูปที่ 2-15 Client Page

#### 2.1.6.2.3 Frameset

ใช้แสดงว่าภายในเป็น Frameset และใน Frameset นั้น ๆ อาจจะประกอบด้วยเว็บเพจต่าง ๆ หรือ  
Frameset ก็ได้

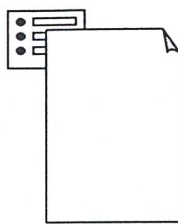


รูปที่ 2-16 Frameset

#### 2.1.6.2.4 Form

โดยส่วนใหญ่จะใช้กับการรับอินพุตเข้าสู่ระบบ ซึ่งฟอร์มนี้จะเป็นส่วนหนึ่งของ Client Page  
และภายใน Form จะประกอบไปด้วยส่วนต่าง ๆ เช่น Button, Check Box, Text เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-17 Form

#### 2.1.6.2.5 Target

เป็นชื่อปลายทางในการแสดงเว็บเพจบนเบราว์เซอร์ของ Client ซึ่งจะใช้ระบุว่าจะให้เพจหนึ่งไปปรากฏบนเฟรมใด



รูปที่ 2-18 Target

## 2.2 Jservlet

Jservlet cartridge ประกอบไปด้วย Java Virtual Machine และ Java Class Libraries ซึ่งมันจะทำงาน Java application ทางด้าน Server-side ซึ่งจะถูกเขียนด้วย Java Servlet API Specification

ข้อดีของการใช้ Jservlet cartridge คือ

- Jservlet cartridge มีประสิทธิภาพของการทำงานมากกว่า CGI เพราะว่ามันไม่ต้องการ start-up และ shutdown ทุก ๆ ครั้งที่มีการเรียกใช้งานเหมือนกับ CGI และในขณะเดียวกันในการติดต่อกับ Database ก็ยังคงใช้งานได้ตลอดเวลาซึ่งก็จะลด Overhead ในการที่ต้องมีการติดต่อกับ Database ใหม่ทุกครั้งที่มีการร้องขอจาก Client
- Jservlet cartridge ได้รับข้อดีมาจาก Oracle Application Server ในด้านของ load-balancing ,scalability ,monitoring,logging และความสามารถในการทำ session management
- มีการใช้งานทรัพยากรของระบบน้อยด้วยการใช้งานแบบ multiple Jservlet cartridge บนเครื่องเดียวกันที่ใช้งาน Application เดียวกัน
- Jservlet cartridge จะมีมาพร้อมกับ Jservlet Toolkit ซึ่งก็จะประกอบไปด้วยกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้ของ Class ซึ่งสามารถใช้ในการสร้าง HTML และ การ access database บนด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JServlet นั้นทำงานอยู่ทางด้าน Server side ซึ่ง JServlet cartridge นี้จะไม่นำไปทำงานบน Java applet ซึ่งจะถูก Download และทำงานอยู่บนฝั่ง Client

#### The init() Method

Init() method ของ Servlet จะถูกเรียกเมื่อ Servlet ได้เริ่มต้นทำงาน ซึ่ง Method นี้ถูกเรียกให้ทำงานเพียงครั้งเดียวในแต่ละครั้งที่ Servlet ทำงาน และจะไม่มีการตอบสนองต่อการร้องขอจนกว่าการทำงานของ init() เสร็จเรียบร้อยแล้วเท่านั้น

#### Entry Point Methods

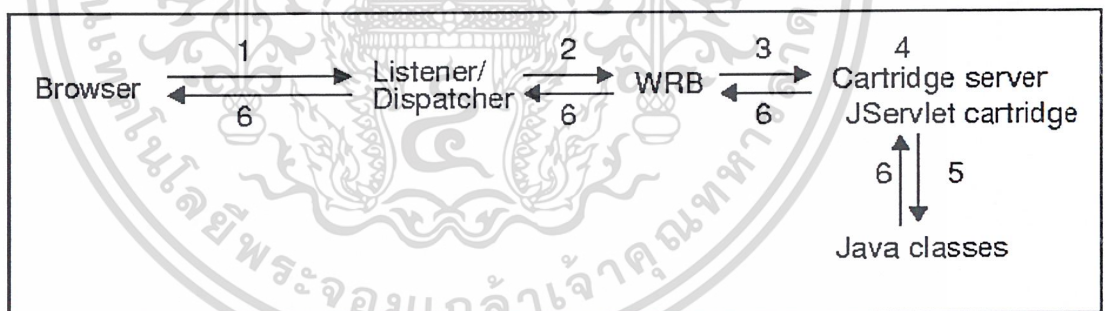
Service() method จะถูกส่งเพื่อร้องขอไปยัง Servlet entry method ซึ่งก็ขึ้นอยู่กับชนิดของการร้องขอมา โดยปกติแล้วจะทำงานทั้งสองแบบ คือ doGet() หรือ doPost() method ทั้งสอง Method นี้สามารถถูกกำหนดหรือถูกใช้เป็น Entry point ไปยัง Servlet

#### The destroy() Method

เมื่อ JServlet ต้องการจบการทำงานหรือ Shutdown แล้วจะเรียก destroy() method ซึ่งก็ จะไม่มีการทำงานใดๆ เกิดขึ้นอีกเมื่อมีการเรียก Method นี้แล้ว Method destroy() นี้จะไม่ถูกเรียกให้ทำงานจนกระทั่ง instance มีการทำงานเสร็จแล้วหรือ Time out

#### Control Flow

รูปข้างล่างจะแสดงว่า Oracle Application Server มีการควบคุมการร้องขอสำหรับ JServlet cartridge



รูปที่ 2-19 การทำงานของ JServlet

1. Listener จะรับสัญญาณร้องขอจาก Client ไปให้กับ JServlet cartridge ยกตัวอย่าง เช่น มีการร้องขอ ไปที่ <http://karla-node/servlets/HelloServlet>
2. เมื่อ Dispatcher เห็นการร้องขอสำหรับ cartridge แล้วก็ส่งต่อการร้องขอไปยัง WRB
3. เมื่อมาถึง WRB แล้วตัว WRB ก็จะทำการตรวจสอบ URL ว่าเป็นการร้องขอนั้น เป็นของ JServlet cartridge เพราะว่ามี Visual path(/servlets) ที่แมปไปยัง JServlet cartridge ดังนั้นก็จะส่งการร้องขอไปยัง JServlet cartridge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Jservlet cartridge ก็จะทำงานอยู่บน cartridge server process ซึ่งจะรับการร้องขอ แล้วตรวจสอบ URL แล้วก็ไปหาชื่อของ Java application ที่ร้องขอมาโดยชื่อของ Class ส่วนใหญ่แล้วก็จะอยู่ส่วนท้ายของ URL
5. จากนั้น Jservlet cartridge ก็จะโหลด class แล้วก็ร้องขอการทำงานไปยัง Entry point method
6. ตัว Java application ก็จะสร้างการตอบสนองซึ่งก็จะประกอบด้วยทั้ง HTTP response header และ response body จากนั้นก็ส่งกลับไปเป็น HtmlStream โดยส่งต่อไปจนถึง Client

### 2.3 เจดีบีซี

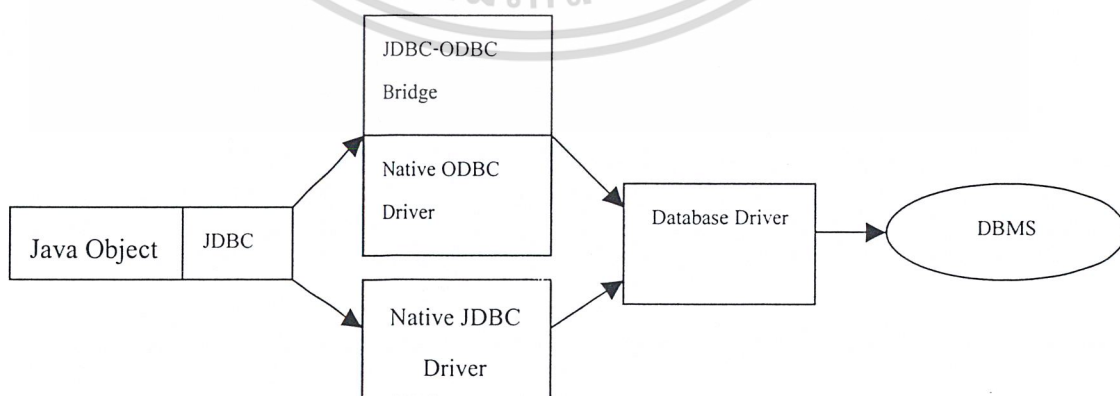
เจดีบีซี หรือ Java Database Connection เป็นมาตรฐานที่ช่วยในการติดต่อ ระหว่างตัวจาวากับระบบจัดการฐานข้อมูล (DBMS : Database Management System) โดยตัวที่ใช้ทำงานจริงๆระหว่างจาวากับระบบจัดการฐานข้อมูลก็คือ เจดีบีซีเอพีไอ ( JDBC API : Java Database Connectivity Application Programming Interface ) สำหรับใช้ในการปฏิบัติตามคำสั่งเอสคิวแอล ซึ่งประกอบด้วยคลาสและอินเทอร์เฟซที่เขียนขึ้นด้วยเจดีบีซี ซึ่งจะจัดหาเอพไอมาตรฐานสำหรับผู้ทำการพัฒนา และทำให้เป็นไปได้ในการเขียนแอปพลิเคชัน เกี่ยวกับฐานข้อมูลโดยใช้ภาษาจาวา

#### 2.3.1 การทำงานของเจดีบีซี

เจดีบีซีจะทำหน้าที่หลักๆ 3 อย่างด้วยกันคือ

1. สร้างจะเชื่อมต่อกับฐานข้อมูลที่ต้องการติดต่อด้วย
2. ส่งคำสั่งเอสคิวแอล (SQL Statement) ไปยังฐานข้อมูลนั้นๆ
3. รับผลลัพธ์จากคำสั่งที่ส่งไปให้ในข้อ 2

รูปแบบที่ใช้ในเจดีบีซี จะเป็นดังนี้



รูปที่ 2-20 รูปแบบที่ใช้ในเจดีบีซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปแบบนี้จะสังเกตเห็นได้ว่า จาวาออบเจ็กต์ (Java Object) จะร้องขอไปยังเจดีย์ช็อคโกแลต ซึ่งจะ เป็นตัวเรียกไปยังตัว เจดีย์ช็อคโกแลตไฟเวอร์ หรือตัวเจดีย์ช็อคโกแลตไฟเวอร์บรีดจ์ ที่กำหนด จากนั้นตัวไฟเวอร์ จะ เรียกตัวไฟเวอร์ของฐานข้อมูลจริงๆ เพื่อที่จะทำการเชื่อมต่อไปยังฐานข้อมูลให้สมบูรณ์

### 2.3.2 JDBC-ODBC Bridge

ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลโดยแปลงฟังก์ชันของเจดีย์ช็อคโกแลตให้อยู่ในรูปแบบฟังก์ชันของ โอดีบีซี ทำให้เจดีย์ช็อคโกแลตสามารถติดต่อกับฐานข้อมูลได้โดยผ่านการทำงานของโอดีบีซี

ข้อดี คือไม่ต้องติดตั้งและปรับเปลี่ยนไฟเวอร์ใหม่ทั้งหมด เพราะโอดีบีซีได้ถูกติดตั้งอยู่บน เครื่องคอมพิวเตอร์ส่วนมากที่ใช้งานในปัจจุบันอยู่แล้ว (Windows95/98/NT, Macintosh และ UNIX บางระบบ) ทำเพียงแค่นำเอาเจดีย์ช็อคโกแลตไฟเวอร์มาต่อเชื่อมเพิ่มประสิทธิภาพให้กับโอดีบีซีเท่านั้น

ข้อเสีย คือ เจดีย์ช็อคโกแลตไฟเวอร์บรีดจ์ไฟเวอร์ถูกเขียนขึ้นมาจากภาษา C/C++ ในบางส่วนจึงทำให้ ไม่สามารถดาวน์โหลดข้ามเครือข่ายอินเทอร์เน็ตและแปลบนเว็บเบราว์เซอร์ได้ เพราะภาษา C/C++ ต้อง รันเฉพาะแต่ละระบบเท่านั้น

ตัวอย่าง Code JDBC

```
// Import
import java.sql.*;
// กำหนดตัวแปร
Connection conn = null;
Statement stmt = null;
String s [] = new String[100];
//(1) เปิดการเชื่อมต่อกับฐานข้อมูล
try {
    Class.forName ("oracle.jdbc.driver.OracleDriver");
    conn = DriverManager.getConnection("jdbc:oracle:thin:@jim:1521:project","scott","tiger");
}
catch ( SQLException e ) {
    System.err.println("Could not establish connection.");
}
catch ( ClassNotFoundException e ) {
    System.err.println("Could not load database driver.");
}
//(2) ส่งคำสั่ง SQL ไปทำงานกับฐานข้อมูล
try{
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ResultSet rset = stmt.executeQuery("select book_name_t from cartoon_details");
// (3) เก็บผลลัพธ์ลงตัวแปร
    int i = 0;
    while(rset.next()){
        s[i]=rset.getString(1);
        i++;
    }
} catch ( SQLException e ) {
    System.err.println("A database error occurred.");
}
// (4) ปิดการเชื่อมต่อ
try {
    stmt.close();
    conn.close();
} catch (SQLException e) {
    System.err.println("Error closing database connection.");
}

```

## 2.4 Applet

Applet คือ โปรแกรมขนาดเล็กที่สร้างขึ้นด้วยภาษา Java สามารถถูกเรียกจากใน HTML page ให้ทำงานเป็นส่วนหนึ่งของ Web page นั้น กล่าวอีกอย่างได้ว่าเป็นโปรแกรมที่ถูกส่งไปกับ HTML page เพื่อไปทำงานภายใต้เบราว์เซอร์ (ที่มี Java Interpreter) บนเครื่องที่รับ HTML page นั้นไป สังเกตว่า โปรแกรมภาษา Java ที่เป็น applications จะทำงานภายใต้ Java interpreter โดยไม่จำเป็นต้องอาศัย โปรแกรมอื่น (จึงเรียกว่า standalone โปรแกรม) และสามารถควบคุมการดำเนินการของตนเอง แต่โปรแกรมที่เป็น Applets จะทำงานภายใต้เบราว์เซอร์ โดยไม่สามารถควบคุมการดำเนินการของตนเองได้ทั้งหมด

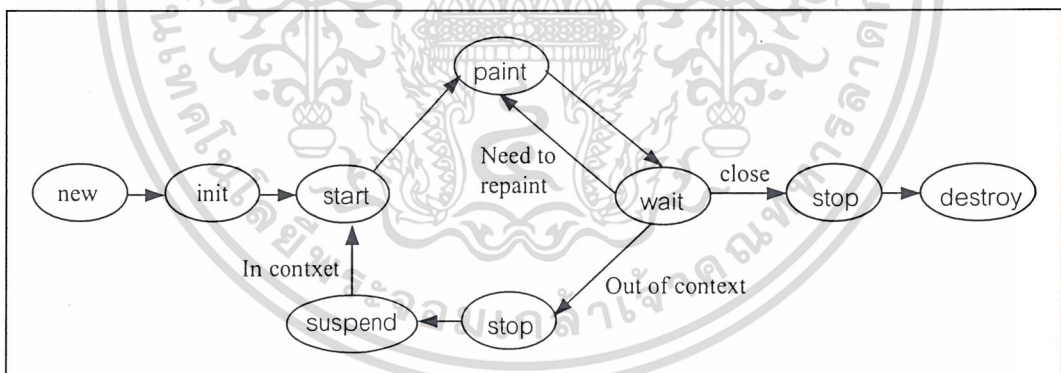
คลาส Applet เก็บอยู่ใน package java.applet ที่มากับ JDK ใช้สำหรับสร้าง applet ทุกโปรแกรมของ applet ต้องมีประโยค import java.applet.\*; เพื่อให้คอมไพเลอร์มองเห็นคลาส Applet และประโยค import java.awt.\*; เพื่อให้คอมไพเลอร์มองเห็น package ที่เรียกว่า Abstract Window Toolkit (AWT) ซึ่งเป็นที่รวบรวมของคลาสเกี่ยวกับ graphics และ graphic user interface (GUI) โดยทั่วไปเราจะไม่สร้าง instance ของคลาส Applet ขึ้นใช้โดยตรง เพราะจะได้ applet ที่ไม่ทำอะไรเลย แต่จะใช้งานคลาส Applet โดยขยายไปเป็นคลาสลูก โดยทำการ override method บางตัวให้ applet ที่ได้นั้นทำงานตามที่เรากำลังต้องการ สังเกตว่าโปรแกรมของ applet ไม่จำเป็นต้องมี main() เป็นจะเริ่มต้นการทำงาน (แม้มีก็ได้ แต่จะไม่ถูกเรียก) เพราะ applets มีช่วงชีวิตที่ถูกควบคุมโดยเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของ applet เมื่อถูกคอมไพล์แล้วจะได้หนึ่งไฟล์เป็น class แต่จะไม่สามารถทำงานได้โดย Java interpreter (java.exe) การเรียกให้ applet ทำงานนั้นต้องทำจากภายใน HTML Page โดยใช้ Applet Tag และ applet นั้นจะทำงานในสถานะแวดล้อมของบราวเซอร์ อย่างเช่น Netscape Navigator, Microsoft Internet Explorer, หรือ Hot Java ซึ่งทั้งหมดเป็นบราวเซอร์ ที่มี Java interpreter อยู่ภายใน หรือ โปรแกรมสำหรับใช้ทดสอบ applet ที่มากับ JDK คือ appletviewer ในหนังสือเล่มนี้เราจะใช้ appletviewer เป็นส่วนใหญ่เนื่องจากทำงานได้เร็วกว่าบราวเซอร์ ทั่วไป และสนับสนุนกลไกของภาษา Java ที่เพิ่มขึ้นใหม่อย่างครบถ้วน ถูกต้องตามมาตรฐานของภาษาที่กำหนดโดยบริษัท Javasoft อย่างไรก็ตาม appletviewer ไม่ได้ถูกสร้างขึ้นเพื่อเป็นบราวเซอร์ มันจึงไม่จำกัดการกับข้อความส่วนอื่นๆ ใน HTML Page และจะทำงานเฉพาะ applet เท่านั้น อีกทั้งยังไม่สามารถติดต่อกับระบบ network ได้เหมือนกับบราวเซอร์ ทั่วไป ดังนั้นในบางกรณีเราจำเป็นต้องใช้บราวเซอร์ จริงๆ ในการตรวจสอบการทำงานของ applet บางตัว ที่ต้องการสถานะแวดล้อมของ บราวเซอร์

#### 2.4.1 Applet Life Cycle

ช่วงชีวิตของ applet แบ่งออกเป็นสถานะต่างๆ ได้ดังนี้ init, start, paint, wait, suspended, stop, destroy ทุกๆ applet จะดำเนินการผ่านสถานะเหล่านี้ โดยบางส่วนจะเป็นไปอย่างอัตโนมัติและบางส่วนอยู่ในการควบคุมของผู้ใช้ที่เรียกดู applet นั้น ช่วงชีวิตของ applet แสดงเป็นรูปการเปลี่ยนสถานะได้ดังนี้



รูปที่ 2-21 สถานะของ Applet Life Cycle

จากรูปเมื่อเริ่มต้น applet ถูกสร้างขึ้นโดยการ new() เหมือน instances ทั่วไป ก็จะเริ่มทำงานทันทีคือ

1. เข้าสู่สถานะ init แล้วผ่านเข้าสู่สถานะ start จากนั้นจะเข้าสู่สถานะ paint และเข้าสู่สถานะ wait ซึ่งจะหยุดรอเหตุการณ์ (Event) หรือสัญญาณจากผู้ใช้ (คืออินพุตที่รับเข้าไปทาง mouse, keyboard หรือ graphic user interface) ที่จะเข้ามาให้เกิดการทำงานของ method ที่กำหนดไว้สำหรับจัดการกับเหตุการณ์นั้น เมื่อเสร็จแล้วจะกลับเข้าสู่สถานะ wait เพื่อรอเหตุการณ์อื่นต่อไป

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของโครงการวิจัยเพื่อการพัฒนา เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ระหว่างที่ applet อยู่ในสถานะ wait อยู่ หากผู้ใช้เปลี่ยนแปลง Web page จนทำให้

2.a. พื้นที่แสดงผลของ applet นั้นต้องมีการถูกวาดใหม่ เช่นเมื่อ applet ถูกย้าย (move) เปลี่ยนขนาด (resize) ขยายขนาด (maximize) หรือมี Windows อื่นมาทับ applet แล้วนำ Window นั้นออกไป บรรดาเซอร์จะออกคำสั่งให้ applet ทำการวาดอีกครั้งโดยเรียก repaint() ซึ่งจะส่งผลให้ applet เข้าสู่สถานะ paint และทำการวาดพื้นที่แสดงผลของ applet นั้นอีกครั้ง เมื่อเสร็จแล้วจะกลับเข้าสู่สถานะ wait เพื่อรอเหตุการณ์อื่นต่อไป

2.b. พื้นที่แสดงผลของ applet นั้นไม่ปรากฏใน Web page ปัจจุบัน เช่น เมื่อผู้ใช้ทำการเลื่อน page ไปดูหน้าอื่น หรือกดปุ่ม minimize จะทำให้ บรรดาเซอร์ออกคำสั่งให้ applet ผ่านเข้าสู่สถานะ stop และไปสู่สถานะ suspended ซึ่ง applet จะหยุดรออยู่ในสถานะนี้โดยไม่ยอมรับอินพุตจากผู้ใช้ และไม่ได้ตอบใดๆ ไปจนกว่าพื้นที่แสดงผลของ applet นั้นจะถูกทำให้กลับมาปรากฏอีกครั้งบรรดาเซอร์ ก็จะออกคำสั่งให้ applet นั้นเข้าสู่สถานะ start, paint และ wait เพื่อเริ่มทำงานอีก

3.ระหว่างที่ applet อยู่ในสถานะ wait หากผู้ใช้กดปุ่ม close ของบรรดาเซอร์ เพื่อหยุดการทำงาน จะทำให้ applet เข้าสู่สถานะ stop และไปสู่สถานะ destroy แล้วสิ้นสุดการทำงาน

ข้อสังเกตเกี่ยวกับวงจรชีวิตของ applet พอสรุปได้ดังนี้

- เมื่อเริ่มต้นทำงาน applet จะเข้าสู่สถานะ init, start, และ paint
- ตอนถูกทำลายก็จะเข้าสู่สถานะ stop, และ destroy
- applet หนึ่งจะผ่านสถานะ init และสถานะ destroy ได้ครั้งเดียว นั่นคือ ผ่านการ init เมื่อตอนเริ่มต้นทำงานครั้งแรกและเมื่อผ่านการ destroy แล้วจะกลับมาทำงานอีกไม่ได้
- applet หนึ่งจะผ่านสถานะ start หรือ stop ได้หลายครั้งขึ้นกับการควบคุมของผู้ใช้ที่ดู applet นั้น
- applet จะหยุดรอเหตุการณ์อยู่ในสถานะ wait หรือ suspended ได้เป็นเวลานานๆ ส่วนสถานะอื่นๆ นั้น applet มีการเข้าไปทำงานแล้วจะผ่านออกไปทันที

ในการสร้างโปรแกรมของ applet เราสามารถกำหนดรูปแบบการทำงานในแต่ละสถานะต่างๆ ที่กล่าวมา เมื่อ applet ดำเนินการผ่านคลาส Applet ก็จะมี methods ชื่อเหมือนกับสถานะคือ init(), start(), paint(), stop(), และ destroy() สำหรับใช้กำหนดการทำงานเมื่อตอน applet นั้นเข้าสู่สถานะนั้นๆ ถ้าตรวจสอบดูจะพบว่า paint() ไม่ได้ถูกกำหนดไว้ในคลาส Applet แต่ได้รับถ่ายทอดมาจากคลาส Component และสังเกตว่าไม่มี method สำหรับสถานะ suspended และ wait เนื่องจากการเข้าสู่และออกจากสถานะ suspended นั้นอยู่ในการควบคุมของบรรดาเซอร์ ส่วนการโปรแกรม applet ในสถานะ wait นั้นเป็นเรื่องใหญ่อีกเรื่องหนึ่งที่จะอธิบายในบทที่เกี่ยวกับ Event Handling

Methods ทั้งหมดที่จะกล่าวถึงในที่นี้คือ init(), start(), paint(), stop(), และ destroy() ถูกกำหนดไว้เป็น method ที่ว่างเปล่าในคลาส Applet ในการสร้างคลาสของ applet เราจะทำการ override methods เหล่านี้ เพื่อระบุการทำงานของ applet นั้นในตอนที่ผ่านมาเข้าไปในสถานะต่างๆ ให้เป็นไปตามที่ต้องการ

```
// LifeTime.java
import java.applet.*;
import java.awt.*;

public class LifeTime extends Applet {
    public void init() { System.out.println("init"); }
    public void start() { System.out.println("start"); }
    public void paint(Graphics g) { System.out.println("paint"); }
    public void stop() { System.out.println("stop"); }
    public void destroy() { System.out.println("destroy"); }
}

<<applet code="LifeTime.class" width=200 height=200>> </applet>
```

### รูปที่ 2-22 แสดงขั้นตอนที่เกิดขึ้น เมื่อ Applet ทำงาน

ในรูปที่ 2-22 คลาส LifeTime นี้ขยายออกจากคลาส Applet มีการ override methods: init(), start(), paint(), stop(), และ destroy() ด้วย methods ที่ แสดงว่า applet อยู่ในสถานะใด

การสร้างคลาสของ applet เราควรทำการ override methods: init(), start(), paint(), stop() และ destroy() แต่ไม่ควรเขียนโปรแกรมที่เรียก methods เหล่านี้โดยตรง แม้ว่าที่จริงแล้วสามารถทำได้ แต่โดยปกติเราจะปล่อยให้ บราวเซอร์ เป็นผู้เรียก methods เหล่านี้ เงื่อนไขที่แต่ละ method นี้จะถูกเรียก มีดังนี้

- init() จะถูกเรียกเมื่อ applet เริ่มต้นทำงานเป็นครั้งแรก และครั้งเดียวเท่านั้น จึงเหมาะสำหรับใช้ในการกำหนดค่าเริ่มต้นให้แก่ตัวแปรใน applet นั้น
- start() จะถูกเรียกหลังการทำงานของ init() เป็นการเริ่มทำงานตามที่เรากำหนดไว้ และจะถูกเรียกทุกครั้งเมื่อพื้นที่แสดงผลของ applet ถูกทำให้ปรากฏใน Web page
- paint() โดยปกติ paint() จะถูกเรียกหลังจากการทำงานของ start() และจะถูกเรียกเมื่อ บราวเซอร์ ต้องการวาดในพื้นที่แสดงผลของ applet นั้นใหม่ ซึ่งอาจเกิดขึ้นได้หลายกรณีเช่น พื้นที่แสดงผลของ applet นั้นถูก window อื่นมาทับ และเมื่อ window นั้นถูกนำออกไปแล้ว paint() ก็จะถูกเรียกให้ทำงาน หรือในกรณีที่ Web page นั้นถูก minimized (หรือ maximized) และเมื่อถูกนำกลับมาอีก paint() ก็จะถูกเรียกเช่นกัน paint() จะมี Graphics g เป็นพารามิเตอร์ ซึ่งเป็นข้อมูลที่เกี่ยวข้องกับสถานะแวดล้อมทาง graphic ของ applet ที่กำลังทำงานนั้นอยู่ สังเกตว่า บราวเซอร์ จะมี Graphics g ซึ่ง paint() สามารถวาดลงใน Graphic g นี้ แต่ในโปรแกรมของ applet ไม่มีข้อมูล Graphics g นี้จึงไม่สามารถเรียก paint() ได้โดยตรง หากเราต้องการให้มีการวาดใหม่เกิดขึ้นก็ต้องเรียก repaint() ซึ่งจะกล่าวถึงในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเรียงรายที่ขอให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- stop() จะถูกเรียกเมื่อ Web page ถูกเลื่อนไปหน้าอื่นที่ไม่มีพื้นที่แสดงผลของ applet นั้นปรากฏ หรือ Web page นั้นถูก minimized เมื่อใดที่ applet ถูก stop() จะทำให้ไม่มีการแสดงผลและไม่สามารถโต้ตอบกับผู้ใช้ แต่ถ้า applet นั้นมี threads อยู่ด้วยก็จะยังมีการทำงานต่อไป หากเราต้องการให้ threads เหล่านั้นหยุดทำงานด้วย ก็ต้องเรียก stop () ของ threads เหล่านั้นด้วย
- destroy() จะถูกเรียกเมื่อ applet นั้นสิ้นสุดการทำงานแล้ว ซึ่งจะเกิดขึ้นเมื่อ Web page นั้นถูกยกเลิกการทำงาน เช่น ถูกกดปุ่ม x ที่อยู่ด้านขวาบนสุด หรือถูกกดค^c จาก Dos prompt ก็จะทำให้ browser เรียก stop() ก่อนจะเรียก destroy() เมื่อ applet ทำงาน destroy พื้นที่หน่วยความจำและทรัพยากรของมันจะถูกเก็บกลับคืนไป ไม่สามารถถูก start() ได้อีก

#### 2.4.2 APPLETTAG

APPLET tag คือข้อความที่อยู่ระหว่าง <applet> กับ </applet> สำหรับใช้เรียก applet ให้ทำงานจากใน HTML page ตัวอย่างที่ผ่านมาใช้แต่เพียงรูปย่อของ APPLETTAG ในตอนนี้เราจะศึกษา รูปแบบเต็มของ APPLETTAG ดังนี้

```
<applet
  code=applet_program
  width=pixels
  height=pixels
  [codebase=URL]
  [name=applet_name] >
</applet>
```

สิ่งที่อยู่ระหว่าง [ กับ ] คือ option ซึ่งจะมีหรือไม่ก็ได้ หากถามด้วย \* จะหมายถึง มีซ้ำอีกกี่ครั้งก็ได้ชื่อที่อยู่ทางด้านซ้ายของ = คือชื่อ attribute ของ APPLETTAG และค่าที่อยู่ด้านขวาของ = (แสดงไว้เป็นตัวเอียง) คือค่าของ attribute นั้น จะเห็นว่าทุกๆ APPLETTAG จะต้องมี attributes อย่างน้อยคือ code, width, และ height โดยที่

- code ใช้ระบุโปรแกรมของ applet ที่จะถูกทำงาน ซึ่งเป็นไฟล์ .class หากเป็น applet ที่อยู่ใน Web site อื่นจะต้องอ้างถึงโดยใช้ URL ถ้าอ้างถึงแต่เพียงชื่อไฟล์จะเป็นชื่อที่ relative กับ URL ของ Web site ที่มีไฟล์ HTML นั้นอยู่
- width ใช้ระบุความยาวของพื้นที่ในแนวนอนที่จะแสดง applet ใน Web page
- height ใช้ระบุความกว้างของพื้นที่ในแนวตั้งที่จะแสดง applet ใน Web page ส่วน attributes ของ APPLETTAG ที่เป็น options มีรายละเอียดดังนี้

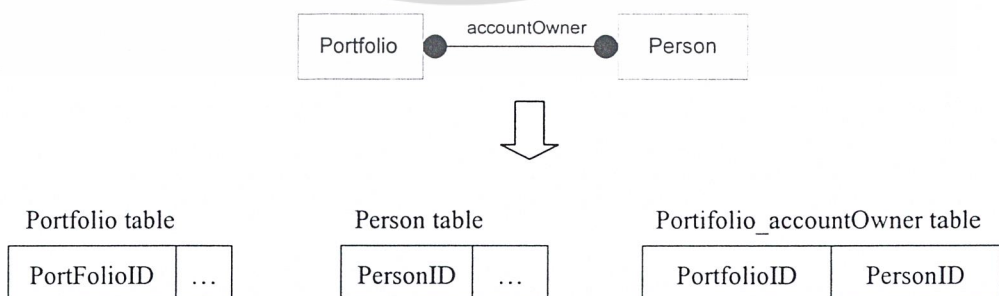
- codebase ใช้ระบุ base URL ของ applet class ที่จะถูกเรียกมาทำงาน ซึ่งก็คือ โดเมนเทอร์มินัลที่จะถูกค้นหา applet นั้น หากไม่ระบุ attribute นี้ base URL ของไฟล์ HTML จะถูกใช้เป็น base URL ของ applets
- name ใช้ระบุชื่อของ applet มีประโยชน์ในกรณีที่บาง Web page อาจมีหลายๆ applet ทำงานพร้อมกัน ซึ่งจะทำให้ applet หนึ่งสามารถอ้างอิงถึงอีก applet หนึ่งได้ ในการที่ applet จะได้ชื่อของ applet หนึ่งจะต้องใช้ getApplet() ที่กำหนดไว้ใน AppletContext interface

## 2.5 การแปลงคลาสเป็นตาราง

หลักการแปลงคลาสไปเป็นตารางคือเราจะแปลงแต่ละคลาสไปเป็นหนึ่ง(หรืออาจมากกว่าหนึ่ง) ตารางโดยแต่ละแอททริบิวต์ของออบเจกต์จะแปลงไปเป็นคอลัมน์ของตาราง ซึ่งชนิดของแอททริบิวต์นั้นจะต้องสามารถเก็บไว้ได้ในฐานข้อมูลแบบรีเลชันแนลได้

ในแต่ละตารางจะมี primary key ซึ่งเหมือนกับเป็นตัว unique identifier ของออบเจกต์ที่เก็บอยู่ในตาราง โดย primary key อาจจะไม่ใช่เป็นเลข ID แต่จะเป็นชื่อหรือวันที่หรือข้อมูลชนิดอื่นๆ ในบางโปรแกรมจะมีการใช้ artificial ID เป็นตัวแทนของ attribute ID ข้อดีของการใช้ artificial ID อยู่ที่ค่าของมันจะไม่มีการเปลี่ยนแปลงจึงมีความอิสระในการเปลี่ยนแปลงค่าข้อมูลของออบเจกต์และทำให้เกิดความเสถียรของ ID สำหรับความสัมพันธ์ที่ถูกอ้างอิงจากออบเจกต์อื่น เช่น หากเราอ้างอิงออบเจกต์โดยใช้ชื่อเราจำเป็นต้องมีการแก้ไขความสัมพันธ์ทั้งหลายที่อ้างอิงเมื่อออบเจกต์มีการเปลี่ยนแปลงค่าของชื่อนั้น ข้อเสียของการใช้ artificial ID คือมีความลำบากในการสร้าง ID สำหรับ RDBMS ที่ไม่มีการสนับสนุน เช่น การหา ID ก่อนหน้า และการนำ ID ที่ถูกลบไปกลับมาใช้ใหม่

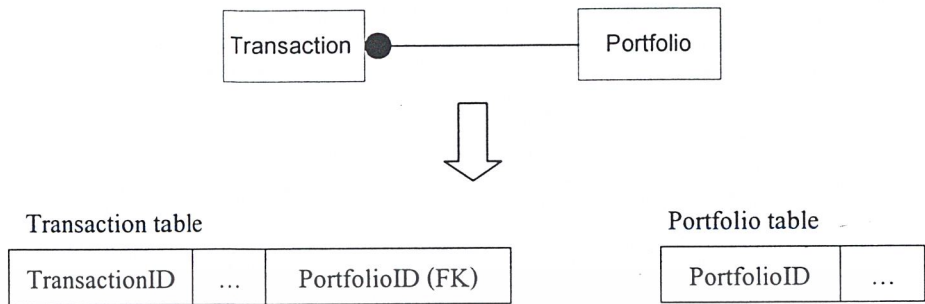
- แยกตารางสำหรับ many - to - many association ควรแปลงแต่ละ many - to - many association เป็นตารางต่างหาก primary key ของความสัมพันธ์นี้ได้จากการรวม primary key จากแต่ละตาราง โดยลำดับของคลาสไม่มีผลกับ primary key และอาจมี non key ที่เป็นแอททริบิวต์ของความสัมพันธ์มาประกอบด้วย



รูปที่ 2-23 วิธีที่แนะนำในการแปลง many - to - many association

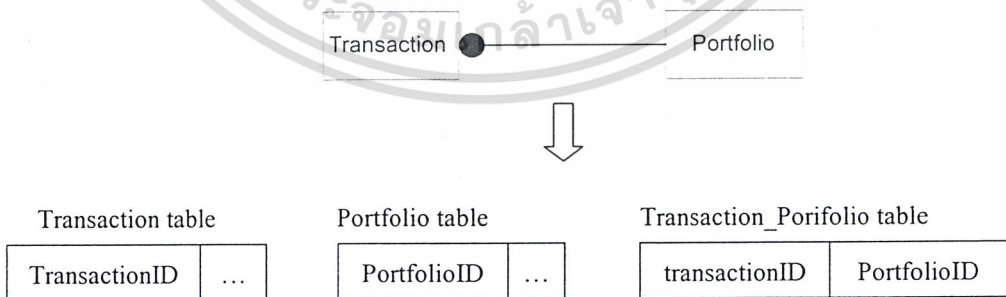
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รวม one - to - many association นำ primary key ของ class ที่มี multiplicity เป็น 1 ไปเป็น foreign key ในตารางของคลาสที่มี multiplicity เป็น many



รูปที่ 2-24 วิธีที่แนะนำในการแปลง one - to - many association

- รวม zero - or - one - exactly one association นำ primary key ของ 1 ไปเป็น foreign key ร่วมกับตารางของคลาสที่เป็น 0 หรือ 1 โดยกำหนดให้ foreign key นั้นห้ามเป็น null
- รวม one - to - one association นำ primary key ของคลาสใดคลาสหนึ่งไปเป็น foreign key ในตารางของอีกคลาสหนึ่ง foreign key จะเป็น null หรือไม่นั้นขึ้นอยู่กับ minimum multiplicity ตามที่กำหนดไว้
- แยกตาราง วิธีที่แนะนำคือให้รวมตารางสำหรับ one - to - one และ one - to - many association แต่ก็สามารถสร้างเป็นตารางต่างหากได้ โดยใน one - to - many association เลื่อน foreign key ของ many มาเป็น primary key ของตารางความสัมพันธ์ ใน one - to - one association สามารถเลือกจากคลาสใดก็ได้ ข้อดีของการแยกตารางคือช่วยให้การออกแบบเป็นไปในทางเดียวกันและสามารถขยายเพิ่มเติมได้หากมีการเปลี่ยน multiplicity ของความสัมพันธ์ สิ่งที่ต้องทำคือกำหนดขอบบังคับใหม่เท่านั้น ไม่จำเป็นต้องเปลี่ยนโครงสร้างของตาราง แต่ข้อเสียคือทำให้ฐานข้อมูลกระจาย



รูป 2-25 วิธีที่เป็นทางเลือกในการแปลง one - to - many association

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การวิเคราะห์และออกแบบระบบ

#### 3.1 การวิเคราะห์ระบบ

##### 3.1.1 ความต้องการของระบบ

- ลูกค้าที่เข้ามาเช่าจะเข้ามาเลือกประเภทของหนังสือการ์ตูนที่สนใจ จากนั้นก็จะเลือกตัวอย่างของหนังสือการ์ตูนที่สนใจจะอ่าน ถ้าลูกค้าต้องการที่จะเช่าหนังสือเล่มใดก็จะเลือกใส่ตะกร้า จากนั้นก็จะ Check Out เพื่อชำระค่าเช่า
    - ในการเช่าหนังสือการ์ตูนลูกค้าจะชำระด้วย Package ที่ได้ซื้อไว้
    - ลูกค้าสามารถซื้อ Package ได้โดยการชำระเงินด้วยบัตรเครดิตผ่านธนาคาร
    - ลูกค้าสามารถกำหนด User Name และ Password ของ Package ที่ซื้อได้เอง
    - ลูกค้าสามารถเลือกราคาของ Package ที่ต้องการซื้อได้จาก menu
    - ลูกค้าสามารถเข้ามาอ่านหนังสือการ์ตูนที่เช่าไว้โดยการแสดง User Name และ Password ของ Package ที่ได้ซื้อไว้ จากนั้นระบบก็จะแสดงรายการของหนังสือการ์ตูนที่ลูกค้าเช่าไว้ เพื่อให้ลูกค้าเลือกอ่าน
      - การอ่านการ์ตูนของลูกค้าจะอ่านจากหน้าจอ
      - ลูกค้าสามารถซื้อ Package เพิ่มจากที่ซื้อไว้ได้โดยไม่ต้องกำหนด User Name และ Password ใหม่
      - ระบบจะแสดงวันและเวลาหมดอายุการเช่า (กำหนดวันที่ต้องคืนหนังสือ) ของหนังสือที่เช่าไว้ในกรเข้ามาอ่านแต่ละครั้ง
      - ในการเช่าแต่ละครั้งถ้ายอด Package คงเหลือไม่เพียงพอ สามารถคืนหนังสือหรือขอขยาย Package ได้
      - ลูกค้าสามารถค้นหาการ์ตูนที่ต้องการได้ โดยการใส่ชื่อการ์ตูนลงในระบบค้นหา
  - Package ที่ลูกค้าซื้อไว้จะถูกสับออกถ้าลูกค้าขาดการติดต่อกันนานมากกว่า 90 วัน
  - การเช่าลูกค้าจะต้องชำระค่าเช่าด้วย Package เท่านั้น
- ลักษณะกิจกรรมต่างๆของระบบเช่า และ อ่านการ์ตูน**
- การเช่า
    - 1.ลูกค้าเข้ามาเลือกตัวอย่างของหนังสือการ์ตูน
    - 2.ถ้าลูกค้าต้องการที่จะเช่าหนังสือเล่มใดก็จะเลือกใส่ตะกร้า
    - 3.จากนั้นก็ Check Out เพื่อชำระค่าเช่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การซื้อ Package
  - 1.ใส่รายละเอียดของบัตรเครดิต
  - 2.กรอกรายละเอียดของPackageที่ต้องการ เช่น User/Password , ราคา
- การเข้ามาอ่าน
  - 1.ลูกค้าใส่ USER/PASSWORD ของ Package
  - 2.ระบบแสดงรายการการ์ตูนที่ได้เข้าไว้
  - 3.ลูกค้าเลือกอ่านการ์ตูนจากรายการ

### 3.1.2 แบบจำลอง USES CASE

การที่จะได้ Actor และ Use Case มานั้นจะต้องผ่านการศึกษาความต้องการของระบบ และผ่านการทำซ้ำหลาย ๆ รอบซึ่งก็เป็นการศึกษาความต้องการของระบบ พร้อมกับกำหนด Actor และ Use Case ไปด้วยกัน และในการศึกษารอบต่อไปอาจจะต้องมีการปรับปรุงเปลี่ยนแปลง Actor และ Use Case ที่ค้นพบด้วย จนในที่สุดได้ Actor และ Use Case ดังรูปที่ 3-1



รูปที่ 3-1 USES CASE Diagram ของระบบเช่าการ์ตูนบนอินเทอร์เน็ต

#### Actor

ลูกค้า (Customer)	ผู้ที่เข้ามาใช้บริการต่างๆ ในร้าน
ผู้ดูแลระบบ (Admin)	ผู้ที่ทำหน้าที่ดูแลระบบภายในร้าน

#### Use Case

##### เช่าการ์ตูน

ผู้เริ่มต้น : ลูกค้า(Customer)

หน้าที่ : ให้บริการในการแสดงรายการการ์ตูนทั้งหมดแก่ลูกค้าและให้ลูกค้า

ได้เลือกเช่าการ์ตูนที่ต้องการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ่านการ์ตูนที่เช่า

ผู้เริ่มต้น : ลูกค้า(Customer)

หน้าที่ : บริการการอ่านหนังสือการ์ตูนที่ลูกค้าได้ทำการเช่าเอาไว้

ซื้อ Package

ผู้เริ่มต้น : ลูกค้า(Customer)

หน้าที่ : ให้บริการแก่ลูกค้าในการซื้อ Package เพื่อนำไปใช้ในการเช่าการ์ตูน และ ทำหน้าที่ในการติดต่อกับธนาคารเพื่อตรวจสอบบัตร Credit และทำ การโอนเงินจากบัญชีของลูกค้าไปยังบัญชีของร้าน

ดูแลระบบ

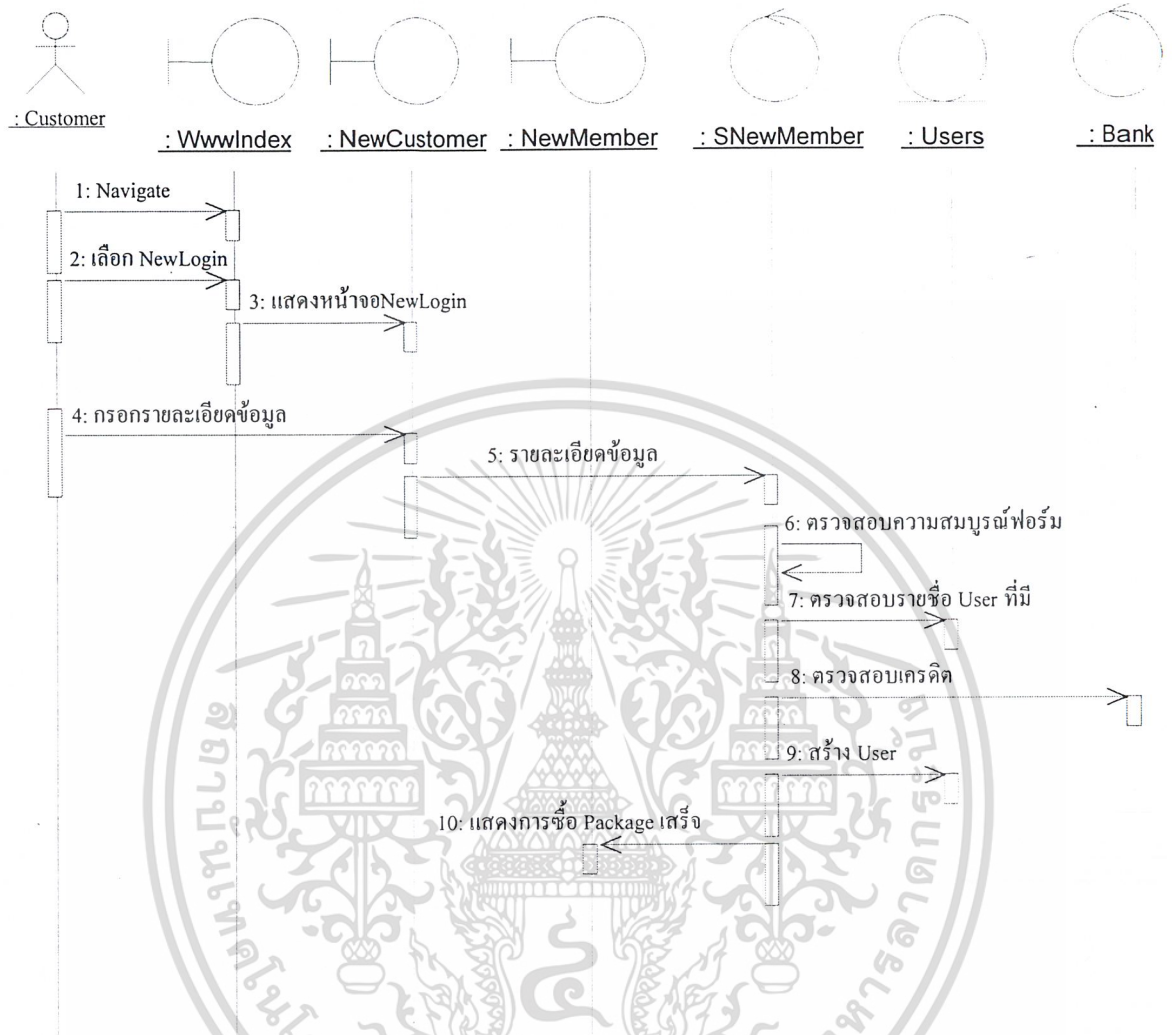
ผู้เริ่มต้น : ผู้ดูแลระบบ (Admin)

หน้าที่ : ทำหน้าที่ดูแลความเรียบร้อยของระบบต่าง ๆ ทั้งหมดและนอกจาก นั้นยังทำหน้าที่ปรับปรุงHome Page ให้มีความทันสมัยอยู่ตลอดเวลา

**3.1.3 Sequence Diagram**

เราจะใช้ Sequence Diagram เป็นหลักในการแสดงเหตุการณ์ที่เกิดขึ้นเพื่อต้องการแสดงให้เห็นถึงลำดับของเหตุการณ์ที่เกิดขึ้น โดยจะไม่แสดงทุก ๆ เหตุการณ์ในที่นี้ แต่จะยกมาแต่เฉพาะเหตุการณ์ที่สำคัญเท่านั้น

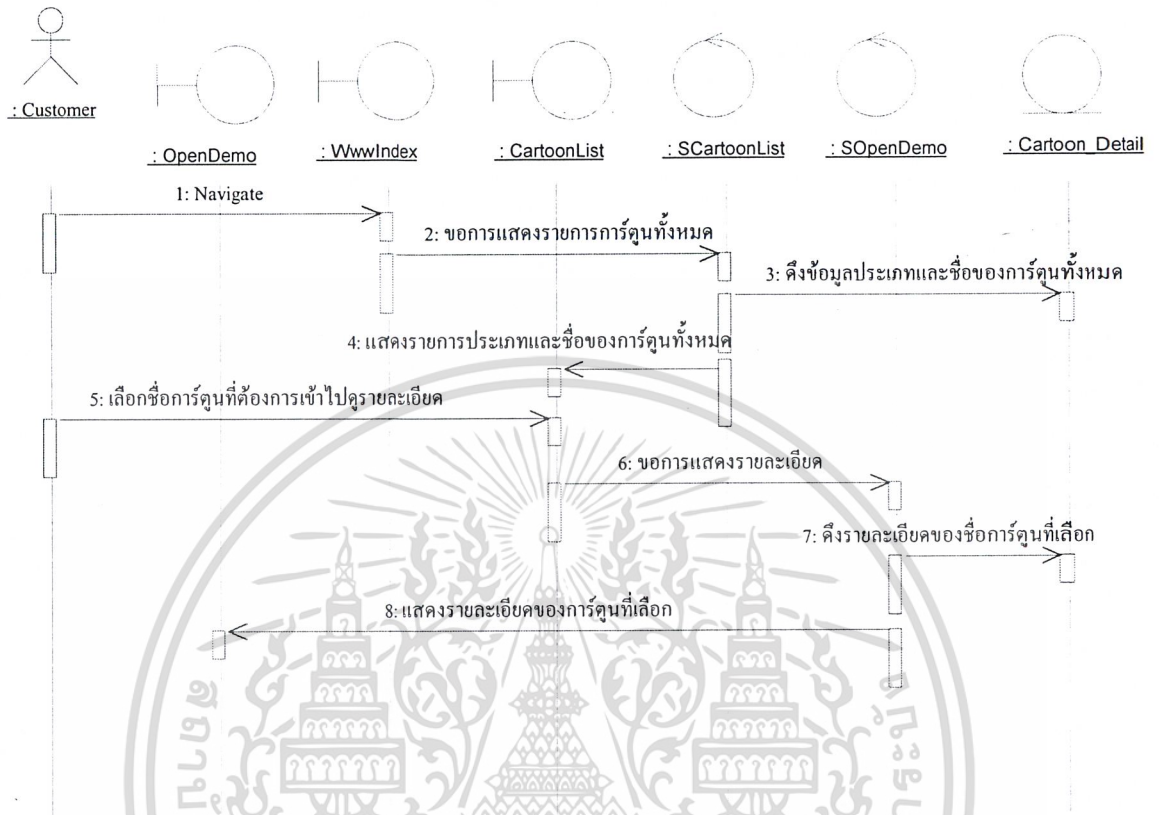
### 3.1.3.1 Sequence Diagram แสดงการทำงานของ การซื้อ Package



รูปที่ 3-2 Sequence Diagram การซื้อ Package ด้วยบัตรเครดิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.2 Sequence Diagram แสดงการทำงานของ การดูรายละเอียด



รูปที่ 3-3 Sequence Diagram การดูรายละเอียดของหนังสือการ์ตูน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

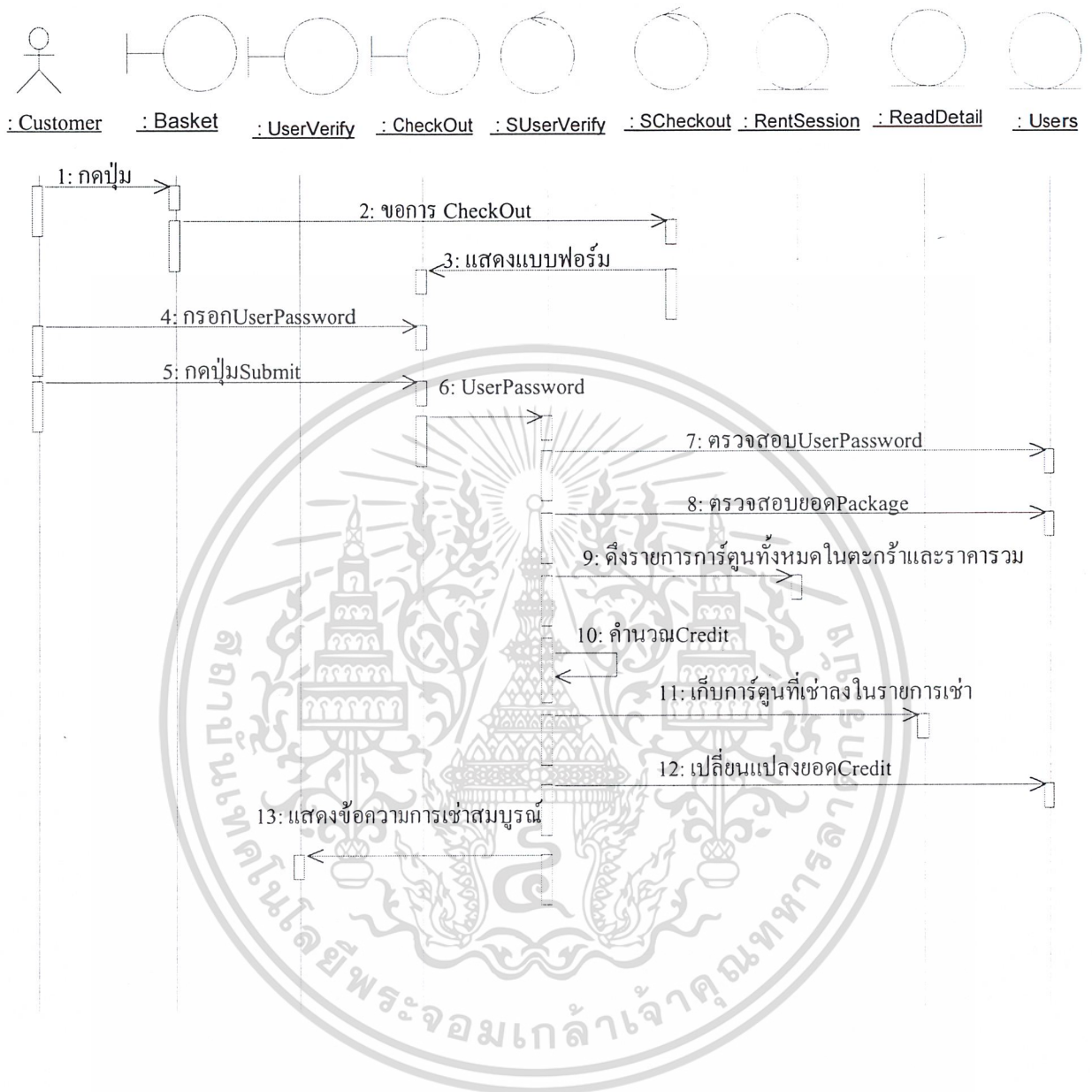
### 3.1.3.3 Sequence Diagram แสดงการทำงานของการทำงานของการเพิ่มหนังสือลงในตะกร้า



รูปที่ 3-4 Sequence Diagram แสดงการทำงานของการทำงานของการเพิ่มหนังสือลงในตะกร้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

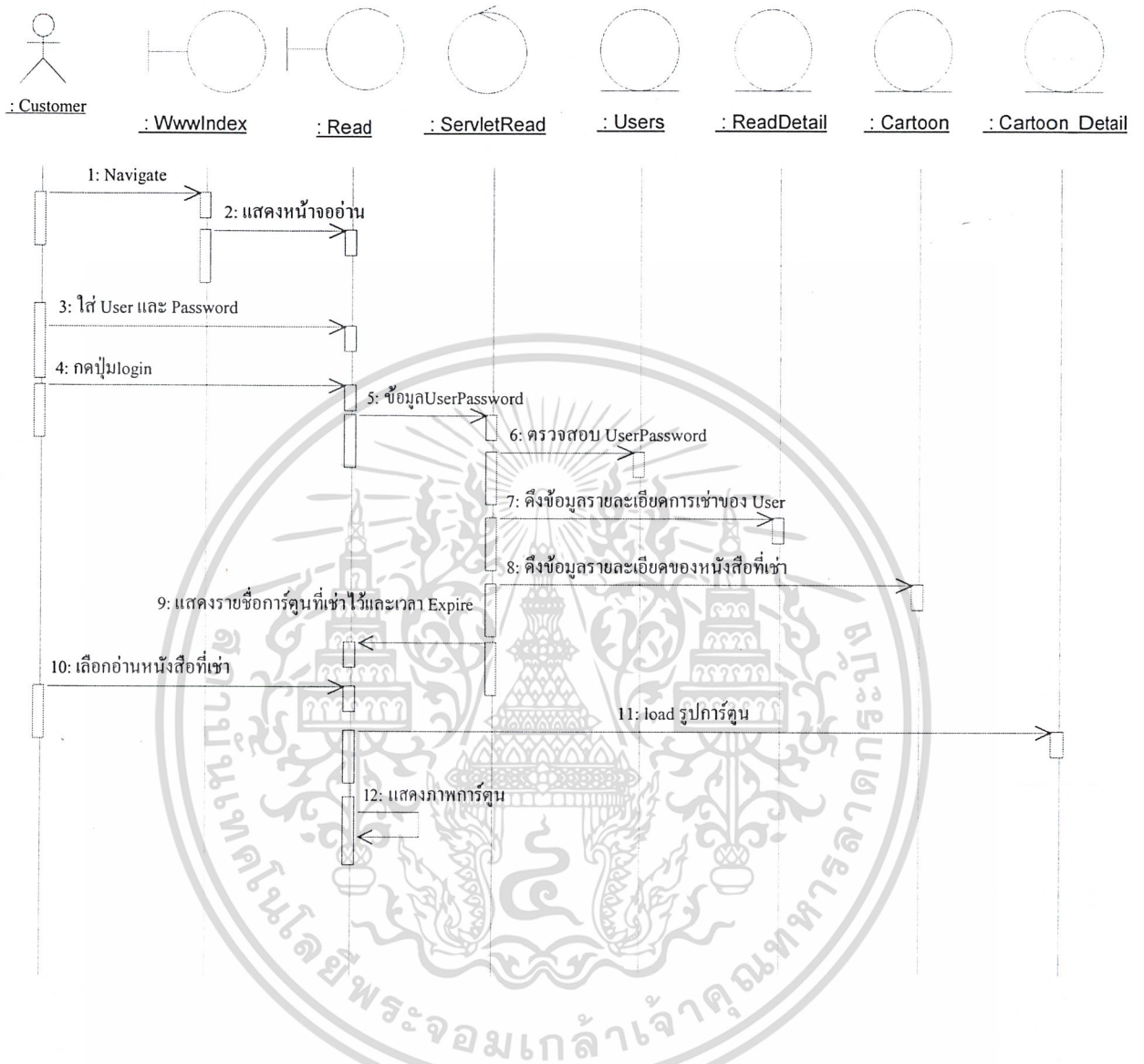
### 3.1.3.4 Sequence Diagram แสดงการทำงานของ การ CheckOut



รูปที่ 3-5 Sequence Diagram แสดงการทำงานของ การ CheckOut

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

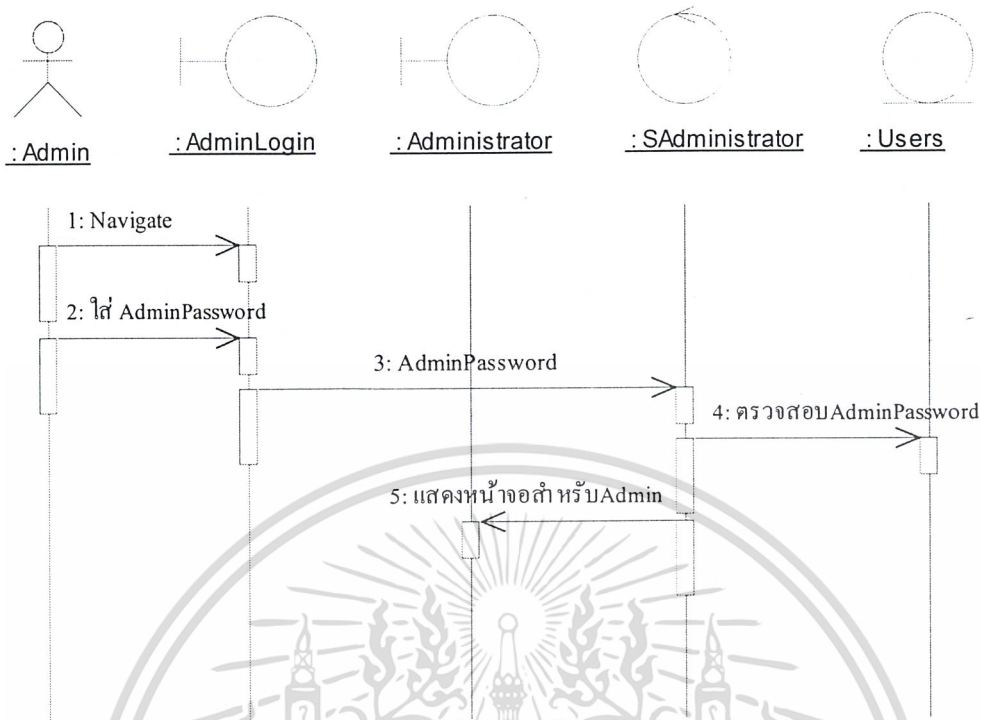
### 3.1.3.5 Sequence Diagram แสดงการทำงานของกรอ่านหนังสือการ์ตูน



รูปที่ 3-6 Sequence Diagram การอ่านหนังสือการ์ตูน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.6 Sequence Diagram การ Login ของผู้ดูแลระบบ



รูปที่ 3-7 Sequence Diagram การ Login ของผู้ดูแลระบบ

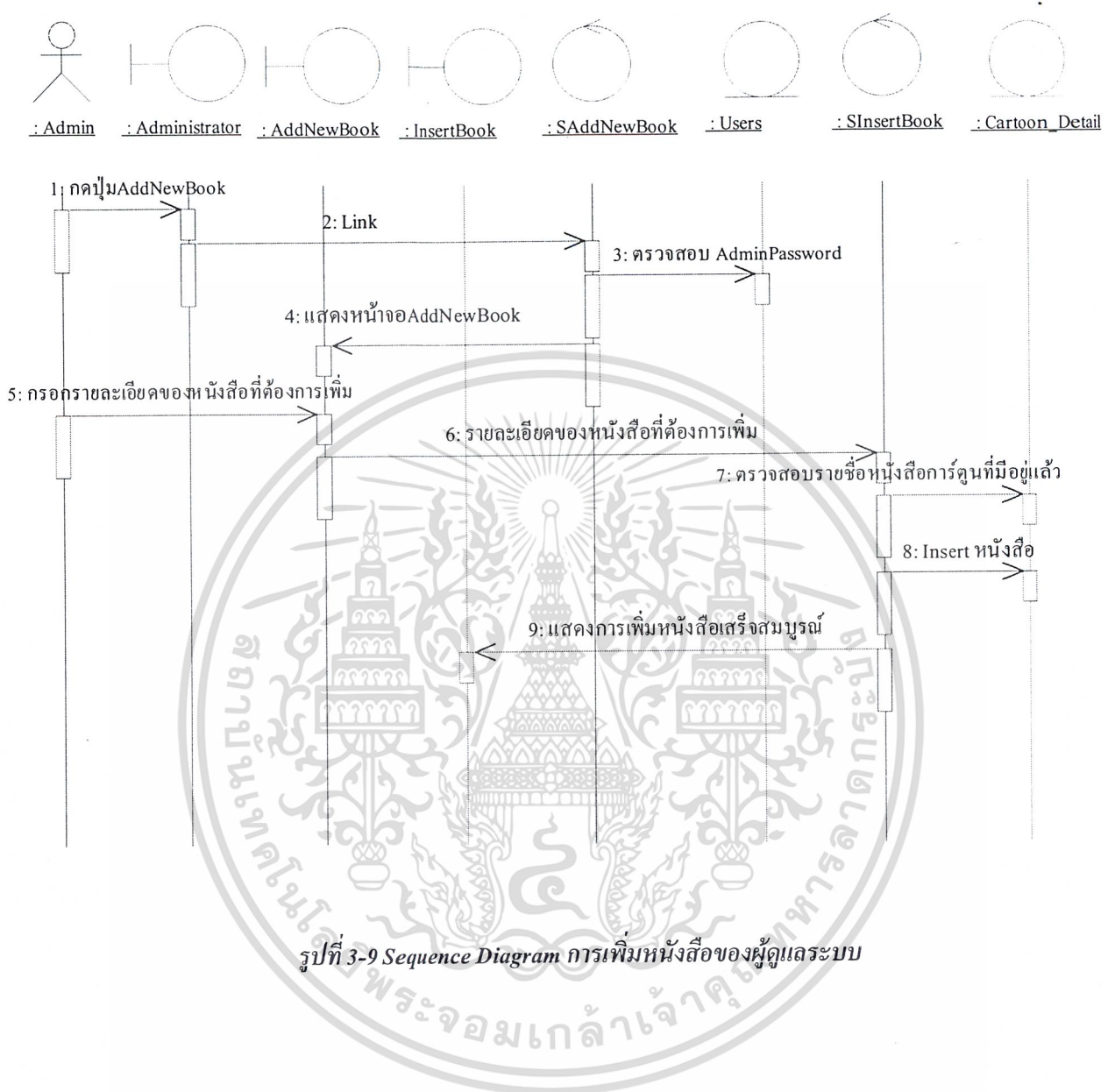
3.1.3.7 Sequence Diagram การลบ User ของผู้ดูแลระบบ



รูปที่ 3-8 Sequence Diagram การลบ User ของผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

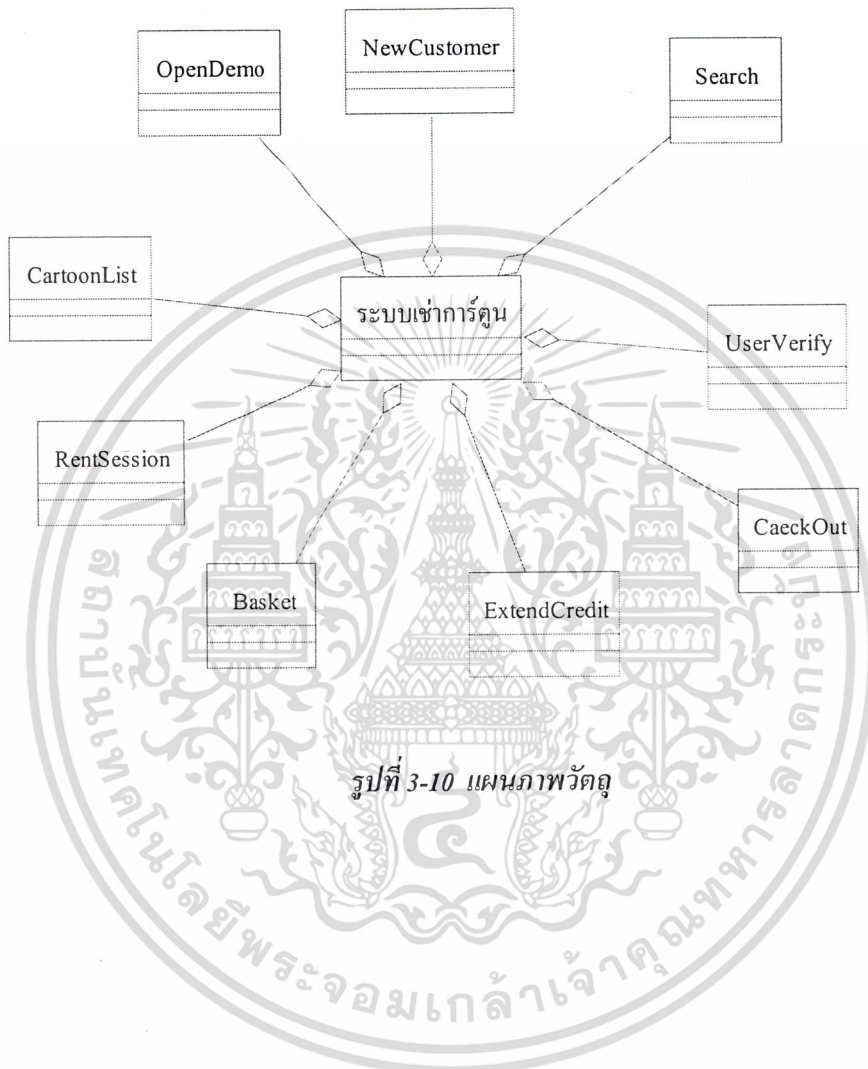
### 3.1.3.8 Sequence Diagram การเพิ่มหนังสือของผู้ดูแลระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4 แผนภาพวัตถุ (Object Diagram)

จากการวิเคราะห์และจากแบบจำลองการทำงาน Use Case Diagram และ Sequence Diagram ที่ได้มาในตอนต้นเพื่อให้เข้าใจส่วนประกอบต่างๆ ของระบบมากขึ้นสามารถจำแนกออกเป็นแผนภาพวัตถุได้ดังรูปที่3-10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

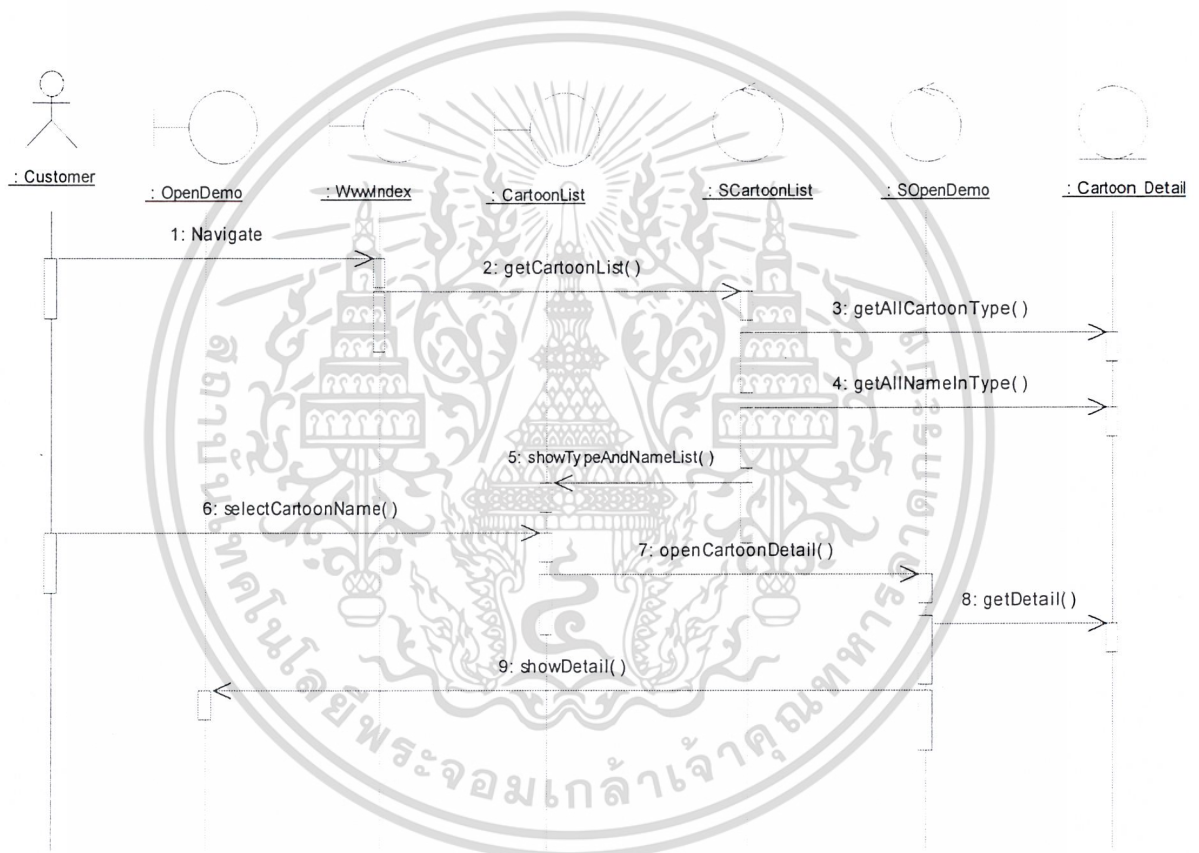
### 3.2 การออกแบบระบบ

หลังจากที่ได้ทำการวิเคราะห์ระบบทำให้มีความเข้าใจระบบมากขึ้นขั้นตอนต่อไปจะเป็นการออกแบบระบบโดยเริ่มจากการออกแบบ Method ต่างๆด้วย Sequence Diagram จากนั้นจะได้ภาพรวมของ Class Diagram (Overview Class Diagram) แล้วนำ Overview Class Diagram มาออกแบบความสัมพันธ์ของ Class Diagram ในแต่ละ Class โดยการออกแบบในแต่ละขั้นตอนไม่จำเป็นต้องทำงานสมบูรณ์ก่อนที่จะทำขั้นตอนต่อไป

#### 3.2.1 Sequence Diagram

ในขั้นตอนนี้จะทำความเข้าใจเกี่ยวกับการกำหนด เมธอด และ แอตทริบิว ของ Class ต่างๆ

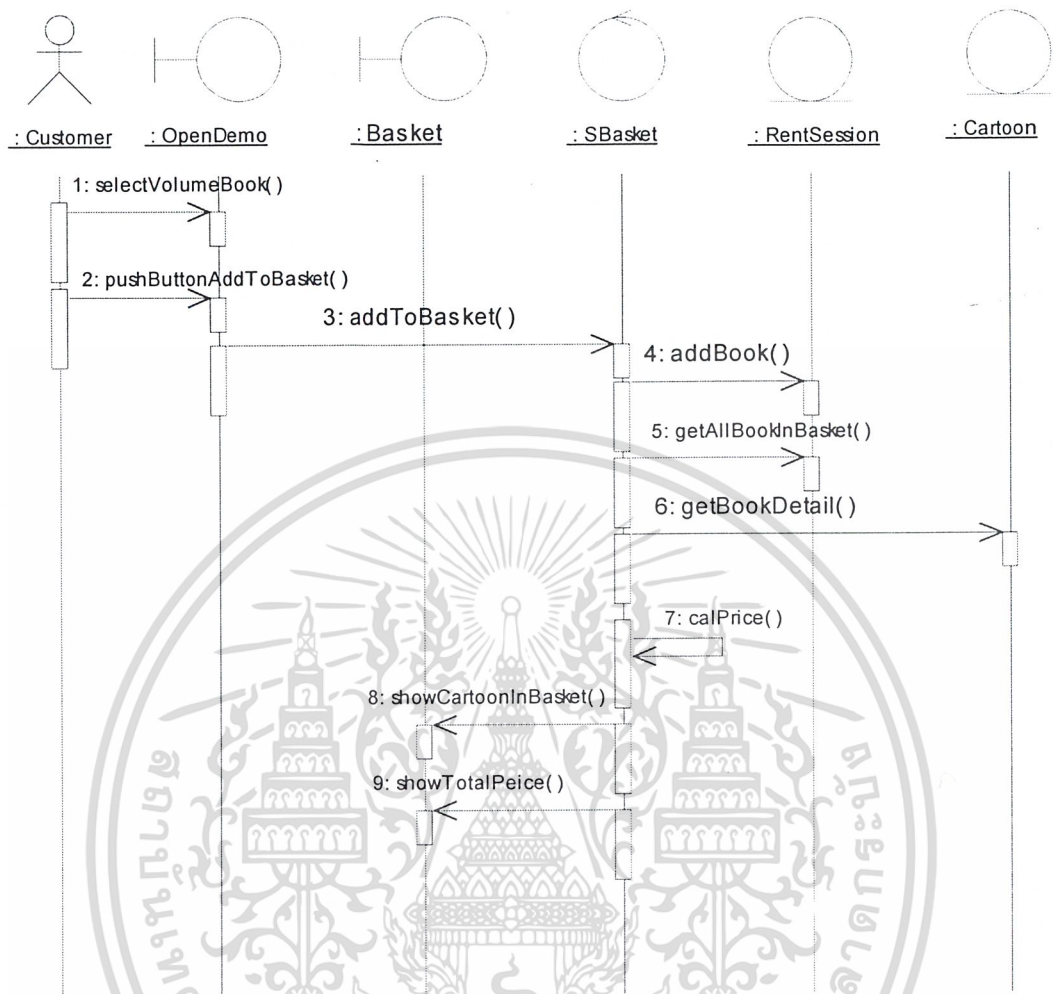
##### 3.2.1.1 Sequence Diagram การอ่านรายละเอียดของหนังสือการ์ตูน



รูปที่ 3-11 แสดง Sequence Diagram ของการอ่านรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

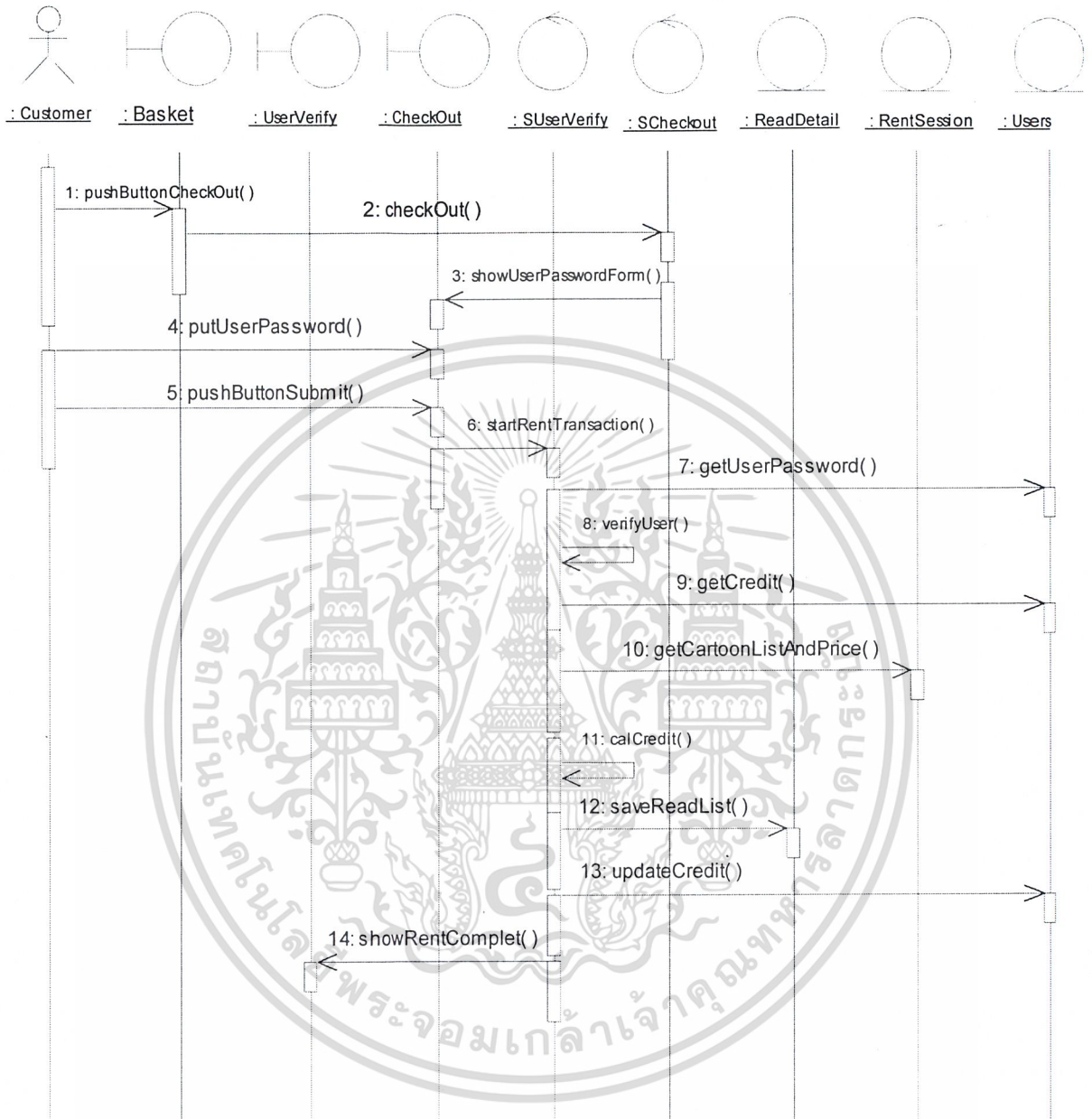
### 3.2.1.2 Sequence Diagram การเพิ่มหนังสือลงตะกร้า



รูปที่ 3-12 แสดง Sequence Diagram ของการเพิ่มหนังสือลงตะกร้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

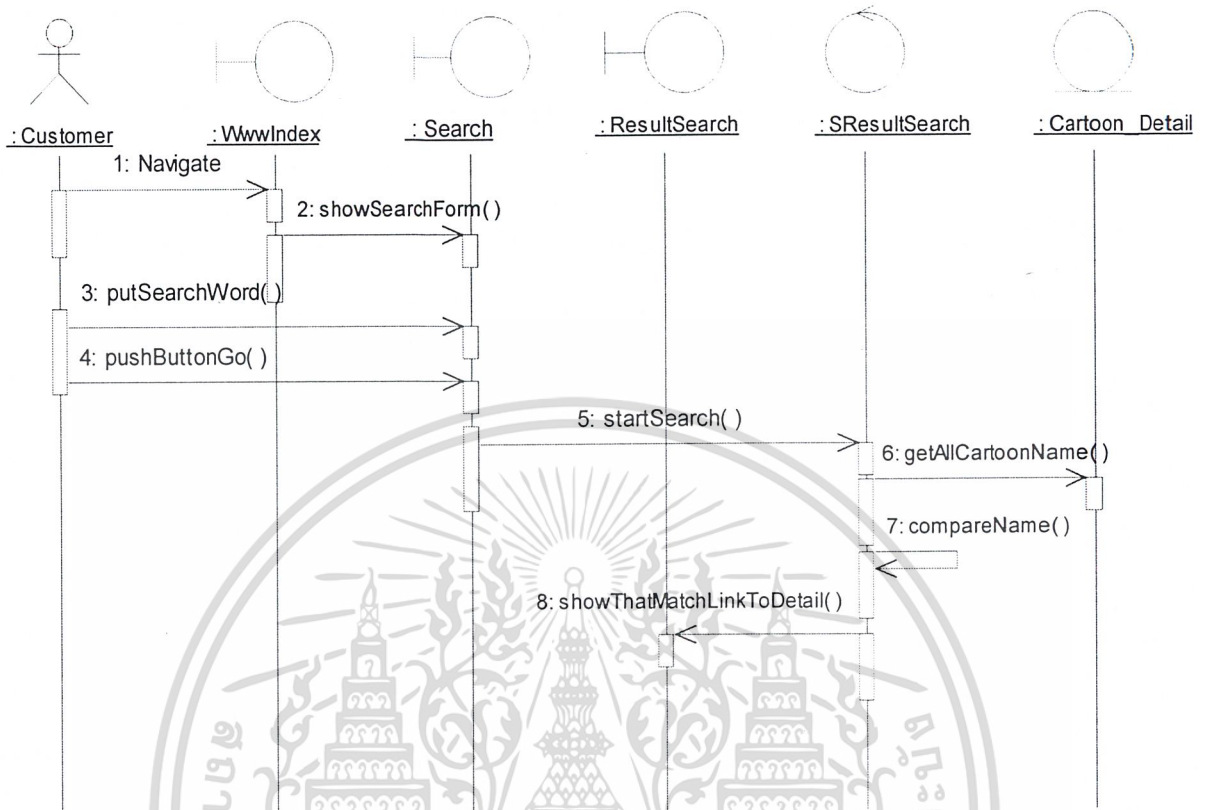
### 3.2.1.3 Sequence Diagram การ Checkout



รูปที่ 3-13 แสดง Sequence Diagram ของ การ Checkout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

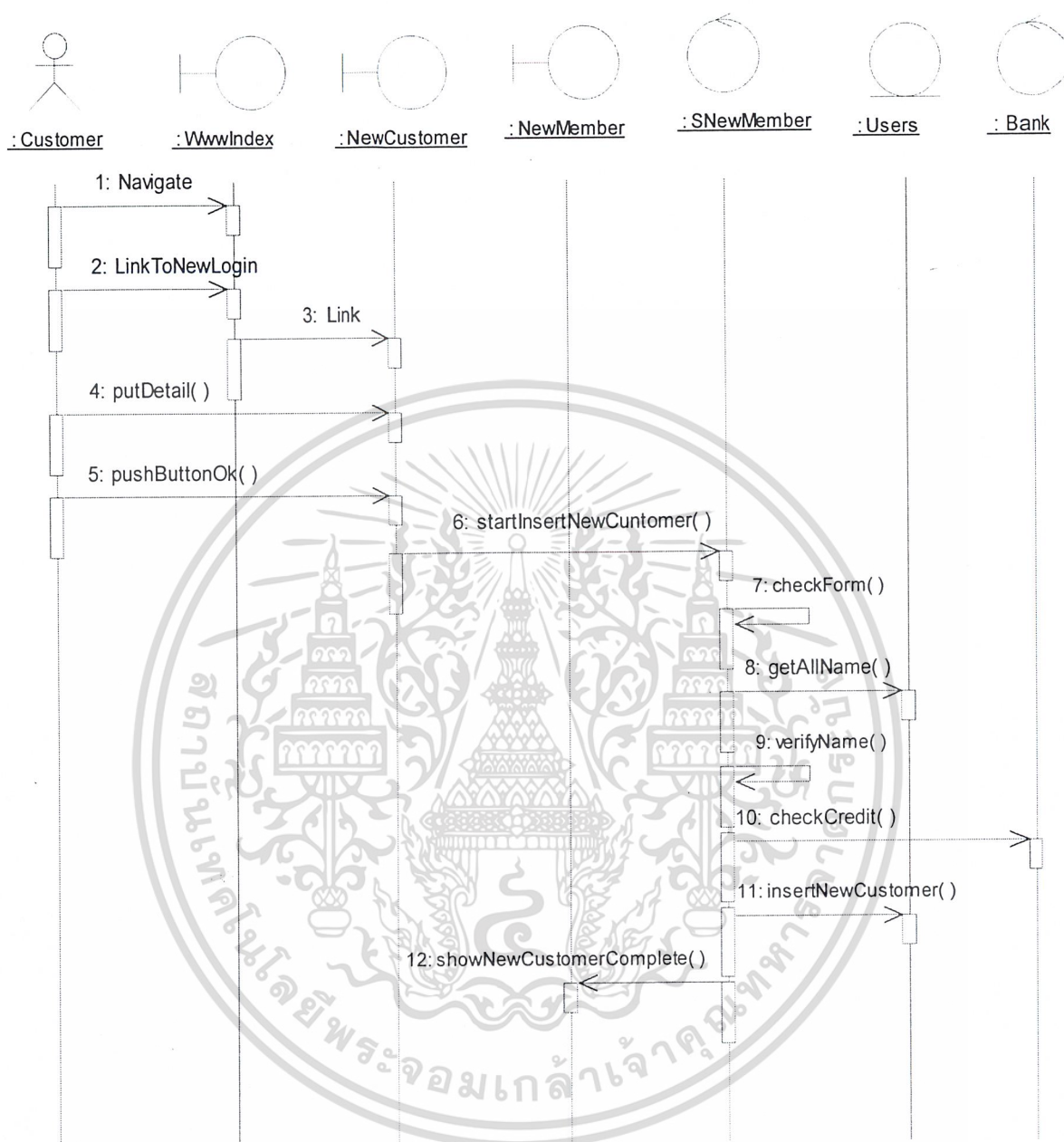
### 3.2.1.4 Sequence Diagram การ Search



รูปที่ 3-14 แสดง Sequence Diagram ของการ Search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

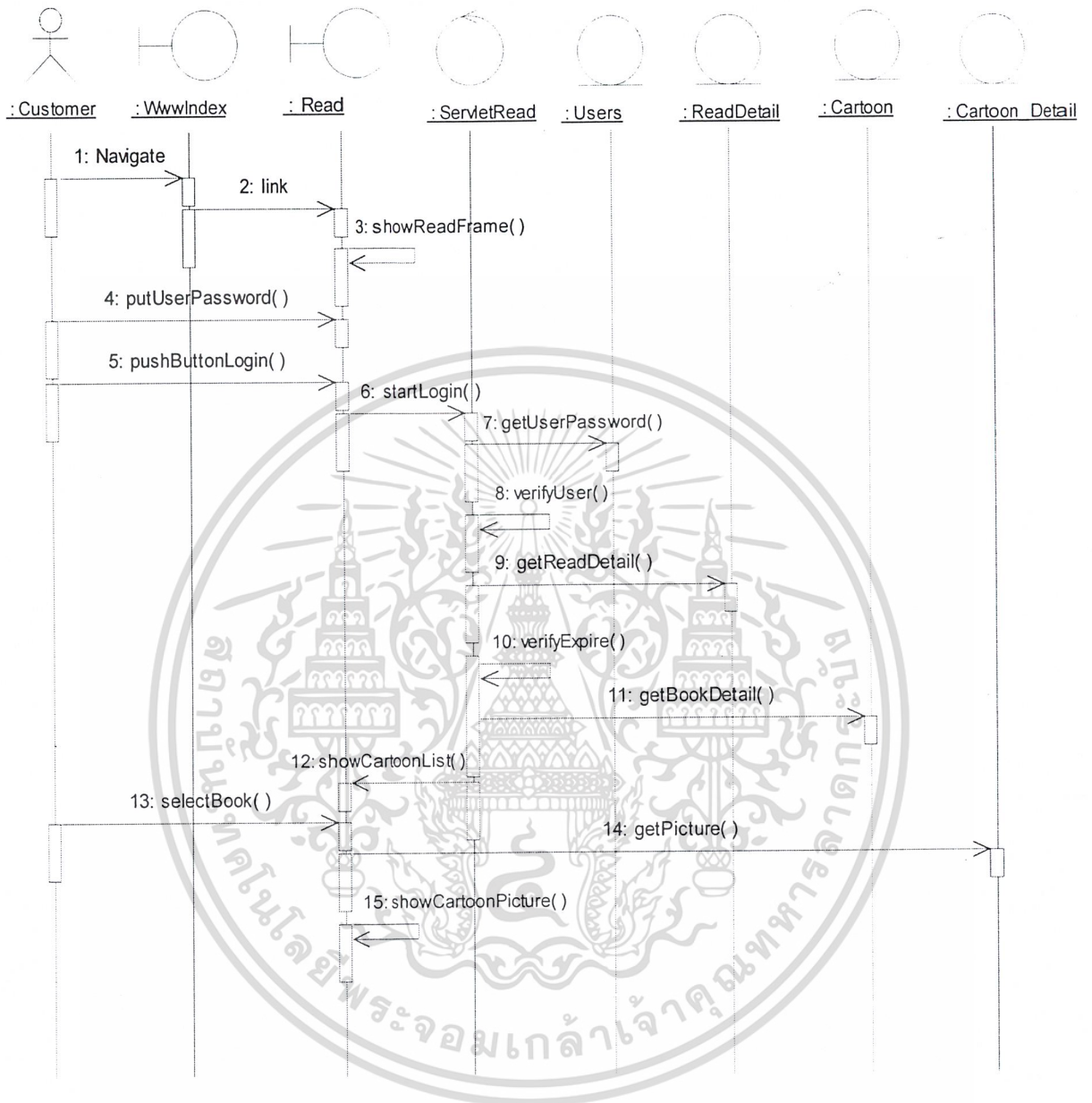
### 3.2.1.5 Sequence Diagram การซื้อ Package ใหม่



รูปที่ 3-15 แสดง Sequence Diagram ของ การซื้อ Package ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

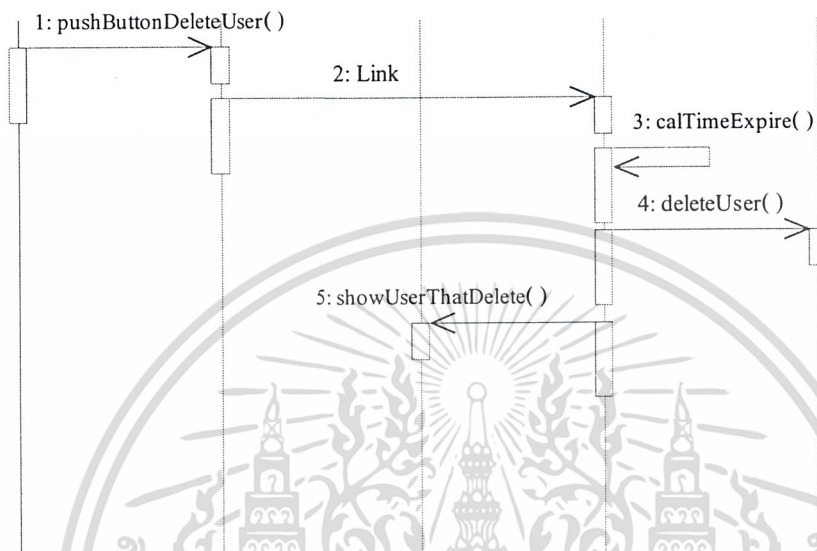
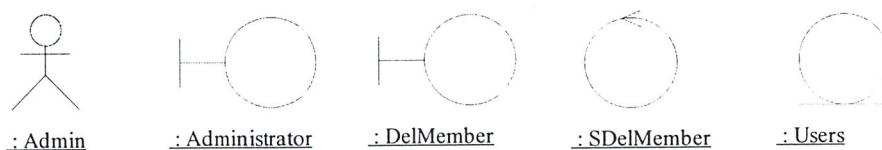
### 3.2.1.6 Sequence Diagram การเข้ามาอ่านของลูกค้า



รูปที่ 3-16 แสดง Sequence Diagram การการเข้ามาอ่านของลูกค้า

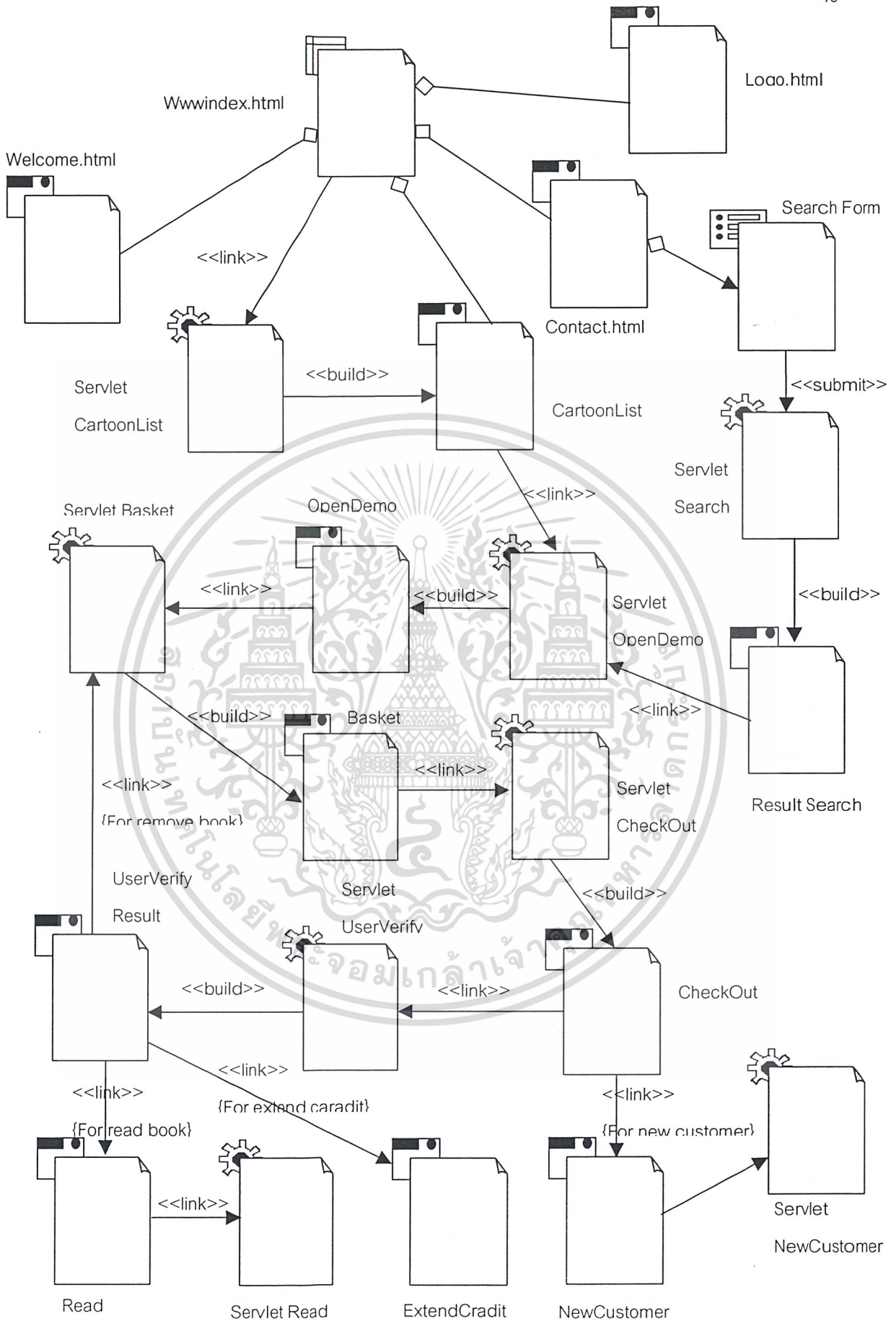
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.7 Sequence Diagram การลบ User ของผู้ดูแลระบบ



รูปที่ 3-17 Sequence Diagram การลบ User ของผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-18 แสดง Overview Class Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

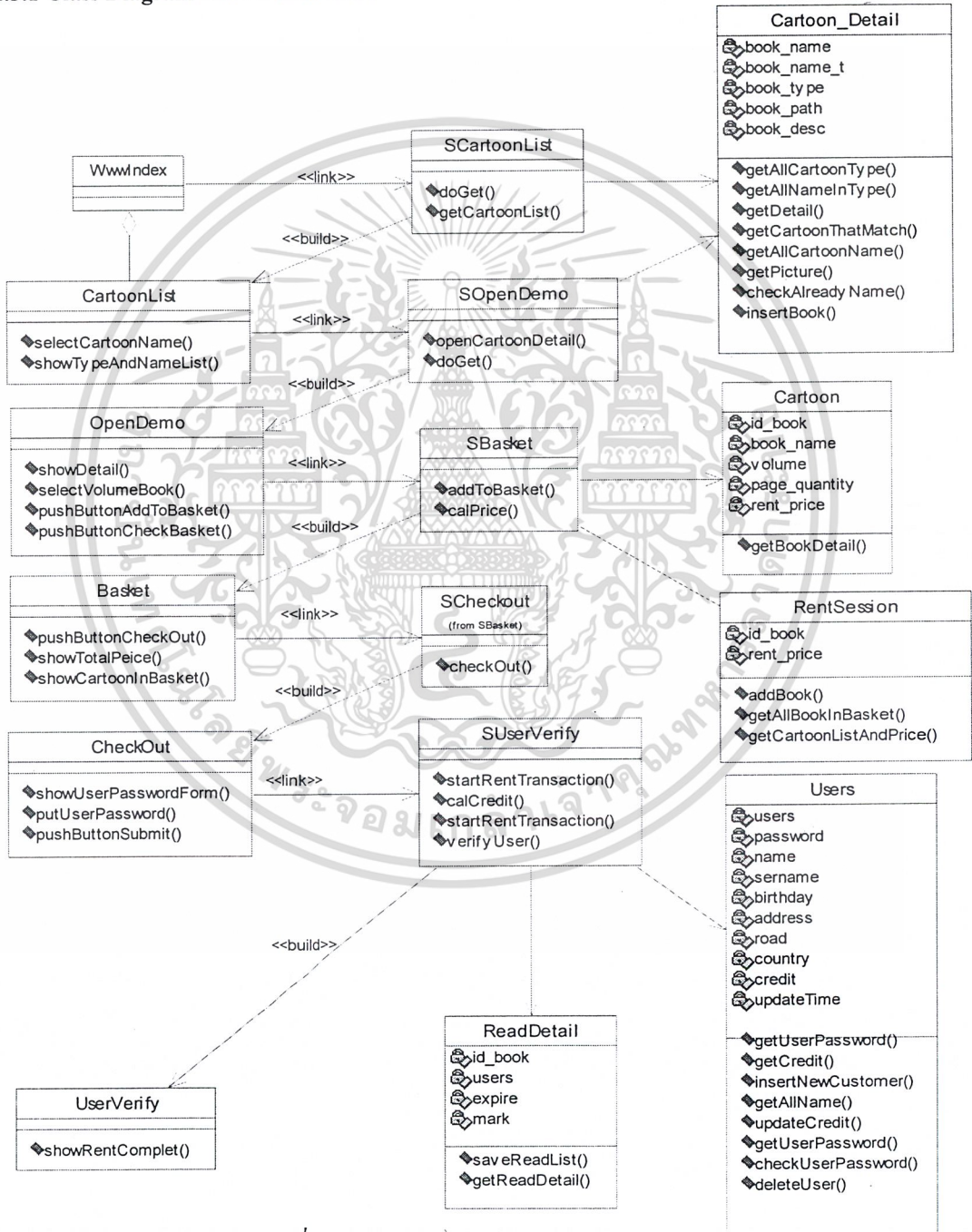
### 3.2.2. Overview Class Diagram

จากรูปที่ 3-18 เป็นการแสดงความสัมพันธ์โดยรวมของ Class และ แสดงถึงลำดับการทำงานของหน้าจอ Interface รวมไปถึงการแสดงลักษณะการทำงานว่าส่วนใดเป็น Server Page และ ส่วนใดที่เป็น Client Page หรือ เป็น Form

### 3.2.3 Class Diagram

จาก Over View Class Diagram นำมาแสดง รายละเอียด ของ Method , Attribute และ ความสัมพันธ์

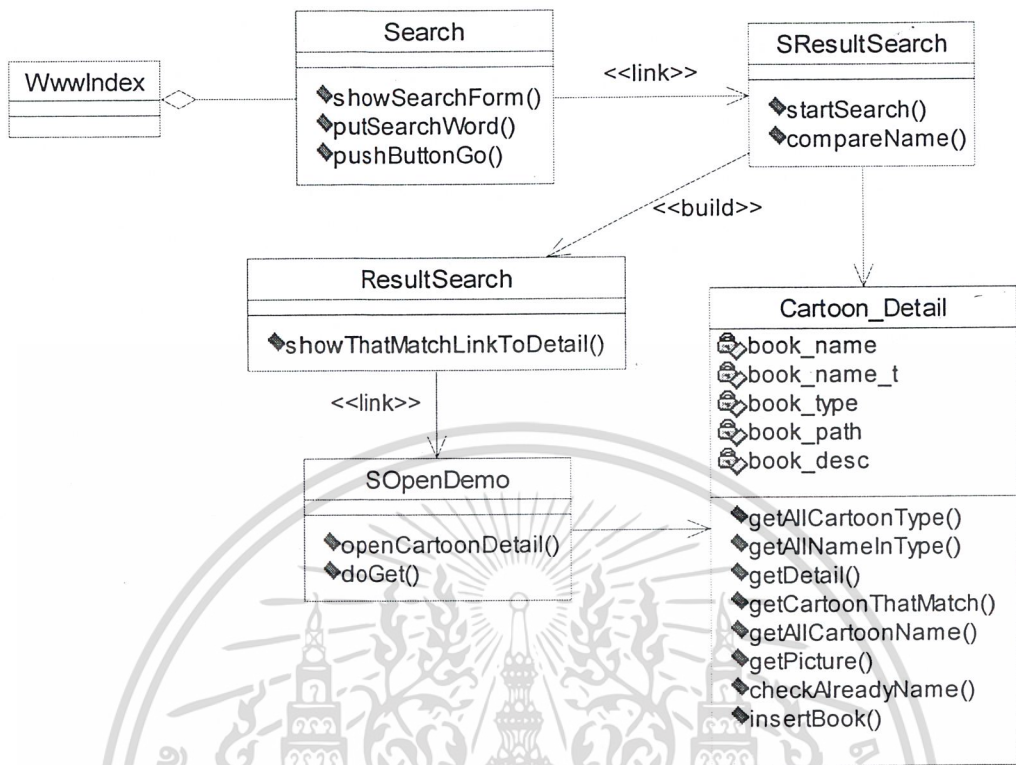
#### 3.2.3.1 Class Diagram ในส่วนของการเช่า



รูปที่ 3-19 Class Diagram ในส่วนของการเช่า

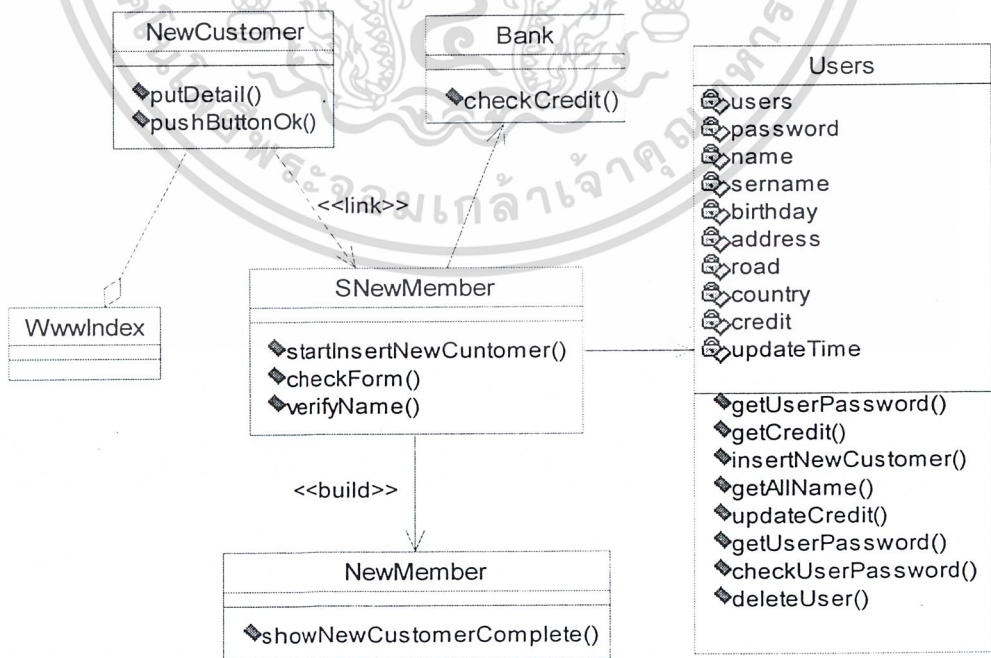
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.2 Class Diagram ส่วนของการ Search



รูปที่ 3-20 แสดง Class Diagram ส่วนของการ Search

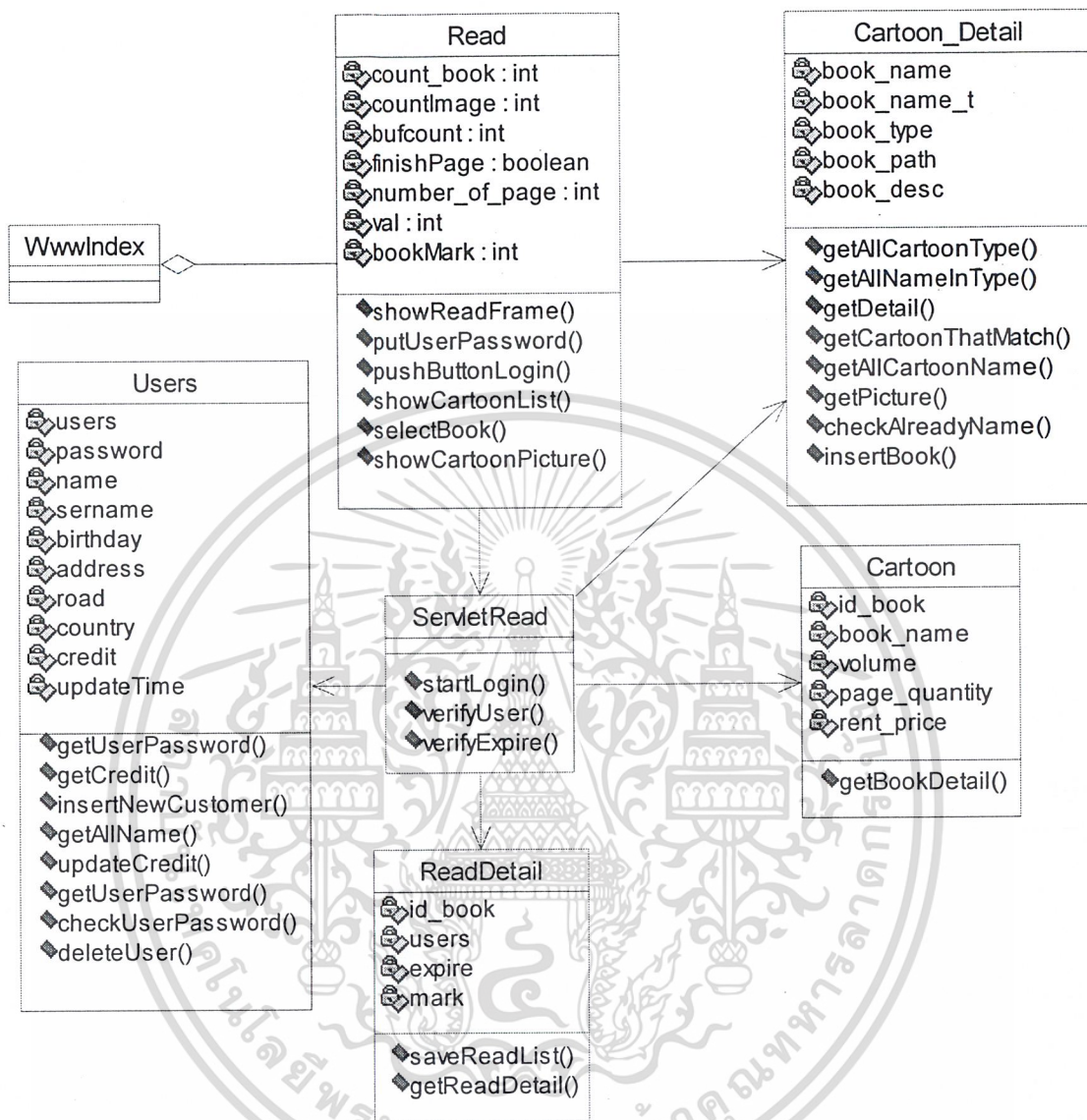
### 3.2.3.3 Class Diagram ส่วนของการซื้อ Package ใหม่



รูปที่ 3-21 แสดง Class Diagram ส่วนของการซื้อ Package ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

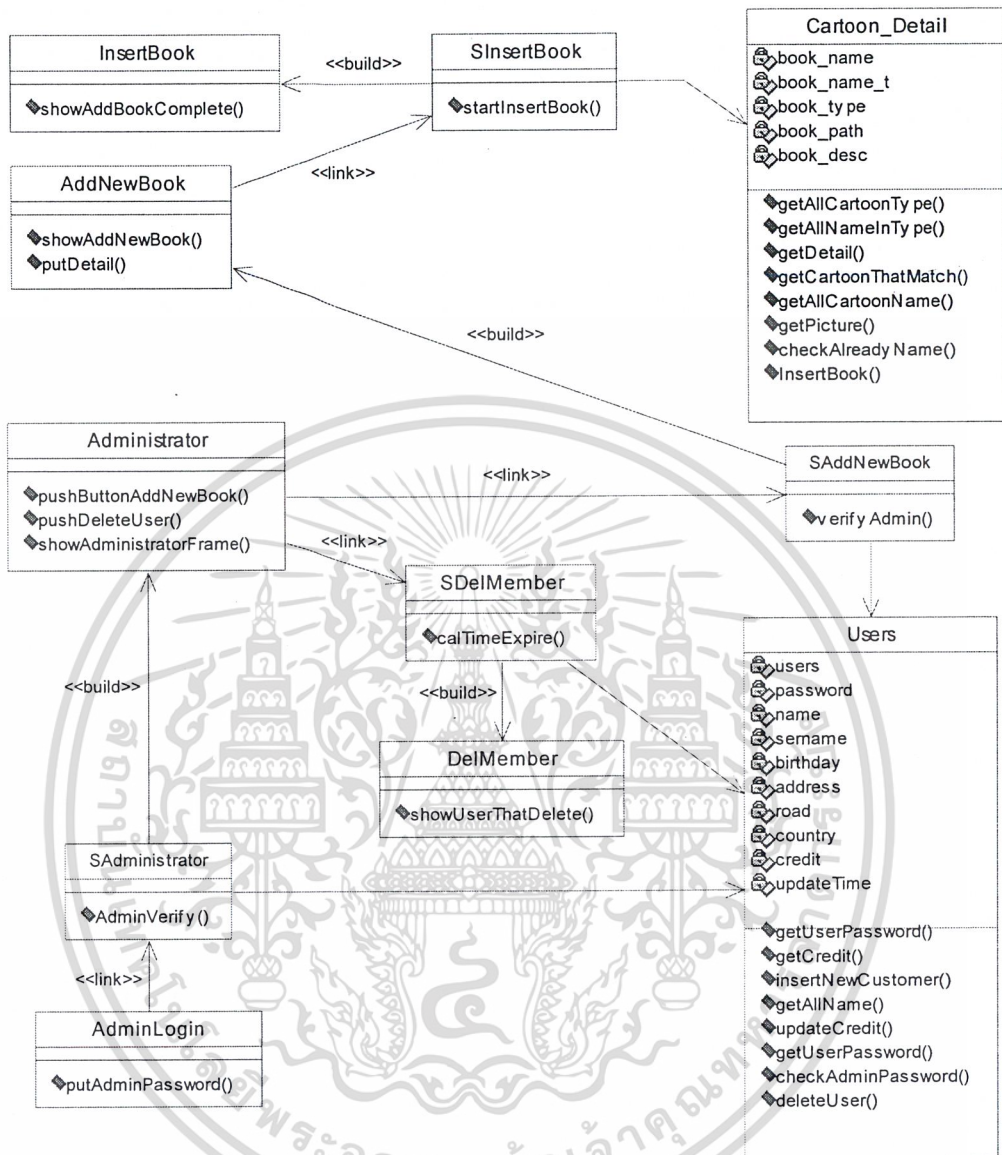
### 3.2.3.4 Class Diagram ส่วนของการ อ่านของลูกค้า



รูปที่ 3-22 แสดง Class Diagram ส่วนของการ อ่านของลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

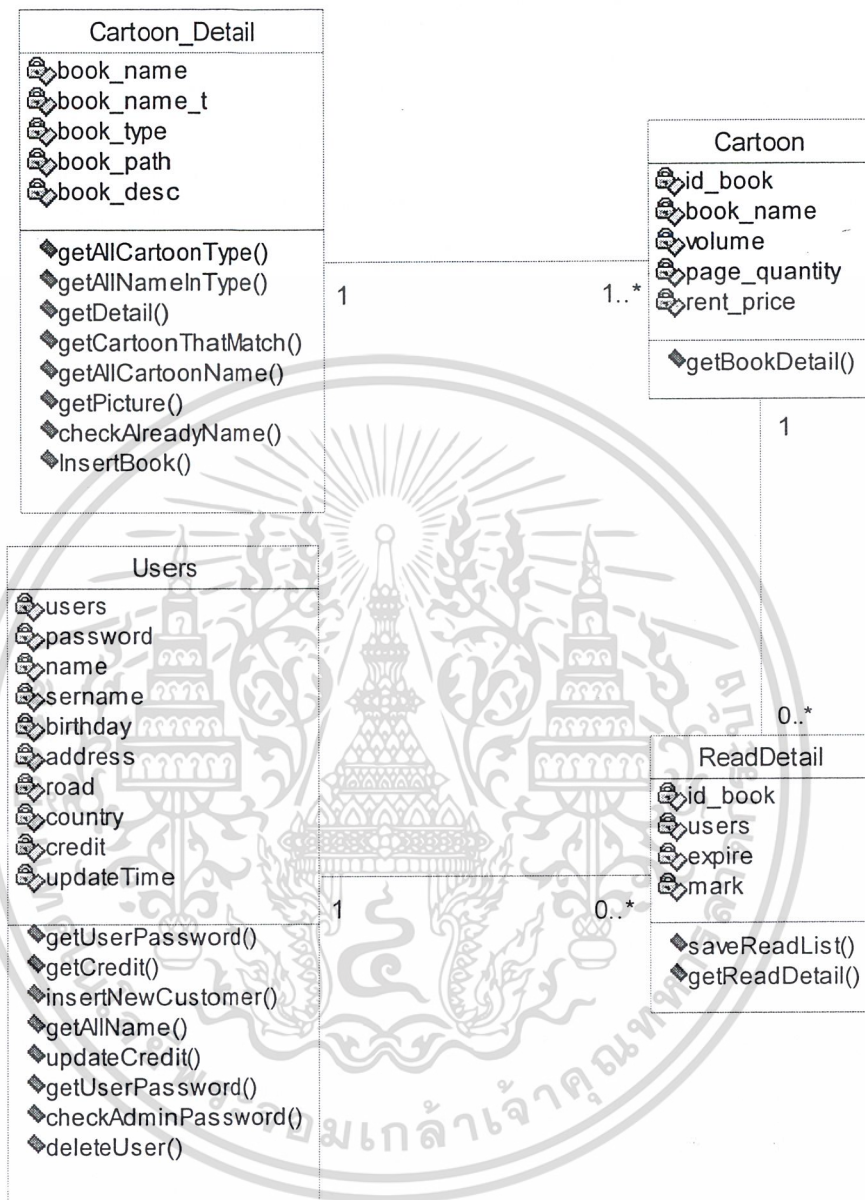
### 3.2.3.5 Class Diagram ในส่วนของผู้ดูแลระบบ



รูปที่ 3-23 แสดง Class Diagram ในส่วนของผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3.6 ความสัมพันธ์ของ Class ที่เป็น Entity Class



รูปที่ 3-24 ความสัมพันธ์ของ Class ที่เป็น Entity Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4 การแปลงคลาสเป็นตาราง

การแปลงคลาสเป็นตารางจะนำเอา Class ที่เป็น Entity Class มาแปลง เป็นตาราง Relational โดยการแปลงจะแปลงแต่ละคลาสไปเป็นหนึ่งตาราง หรือ มากกว่า 1 ตารางก็ได้ และในบางกรณีทีคลาสได้จากการสืบทอดคุณสมบัติคลาสเหล่านั้นก็จะนำมารวมเป็น ตารางเดียวกัน

การแปลงคลาสที่มีความสัมพันธ์กันแบบ one-to-many association จะนำ primary key ของคลาสที่มี multiplicity เป็น 1 ไปเป็น foreign key ของตารางของคลาสที่เป็น many

จากคลาส Diagram และ ความสัมพันธ์ของ คลาสที่เป็น Entity Class สามารถแปลงได้เป็นตารางด้านล่างนี้

ตาราง Cartoon

คอลัมน์	ชนิดข้อมูล	ข้อบังคับ
id_book	NUMBER(5)	Primary Key
Book_name	CHAR (30)	Foreign Key อ้างอิง Cartoon_details
Volume	NUMBER(5)	Not Null
Page_guantity	NUMBER(5)	Not Null
Rent_price	NUMBER(5)	Not Null

ตารางที่ 3-1 แสดงตารางที่ได้จากคลาส Cartoon

ตาราง Cartoon\_details

คอลัมน์	ชนิดข้อมูล	ข้อบังคับ
Book_name	char(50)	Primary Key
Book_name_t	char (50)	Not Null
Book_type	char(50)	Not Null
Book_path	char(50)	Not Null
Book_desc	char(200)	Not Null

ตารางที่ 3-2 แสดงตารางที่ได้จากคลาส Cartoon\_details

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง Users

คอลัมน์	ชนิดข้อมูล	ข้อบังคับ
Users	char(50)	Primary Key
Password	char(50)	Not Null
Name	char(50)	-
Surname	char(50)	-
Birthday	char(50)	-
Address	char(50)	-
Road	char(50)	-
Country	char(50)	-
Credit	NUMBER(15)	Not Null
Updatetime	NUMBER(5)	Not Null

ตารางที่ 3-3 แสดงตารางที่ได้จากคลาส Users

ตาราง Read

คอลัมน์	ชนิดข้อมูล	ข้อบังคับ
id_book	NUMBER(5)	Primary Key(concatenate key)
Users	char(50)	Primary Key(concatenate key) , Foreign Key อ้างอิง Users
Expire	char(50)	Not Null
Mark	char(50)	Not Null

ตารางที่ 3-4 แสดงตารางที่ได้จากคลาส ReadDetail

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.5 การออกแบบหน้าจอ

#### 3.2.5.1 องค์ประกอบของหน้าจอหลัก

หน้าจอหลักประกอบไปด้วย 4 ส่วน คือ

- **Logo**

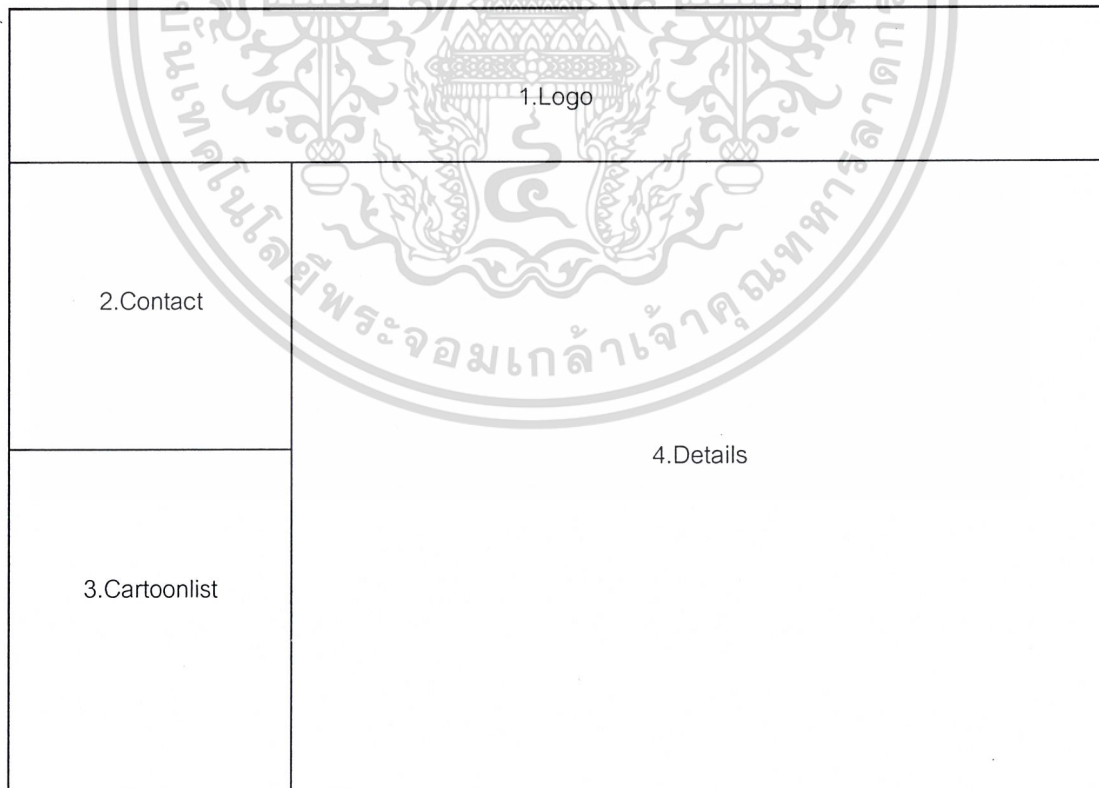
เป็นส่วนแสดง Logo ของร้าน

- **Contact**

เป็นส่วนที่ใช้แสดงรายการของกิจกรรมต่างๆ สำหรับลูกค้าในส่วนนี้จะเป็น Menu ที่มี Link ไปยังหน้า จออื่นซึ่งส่วนของ Contact ประกอบไปด้วย

- About us ข้อมูลเกี่ยวกับผู้สร้าง หรือ ผู้ดูแลระบบ
- Contact us ติดต่อส่ง e-mail ให้กับผู้สร้าง หรือ ผู้ดูแลระบบ
- Login การเข้าไปที่หน้าจอ Login เพื่ออ่านหนังสือที่เช่าไว้
- New login ไปที่หน้าจอซื้อ Package ใหม่
- Extend Credit ลิงค์ไปที่การขยาย Package เพื่อเพิ่ม Credit
- WebBoard ลิงค์ไปที่การแสดงความคิดเห็น
- Search ค้นหาหนังสือการ์ตูนที่ต้องการ

ส่วนของหน้าจอ Contact จะอยู่คงที่ตลอดเพื่อให้ ผู้ใช้สามารถ Link ไปหน้าจอใดก็ได้ตลอดเวลา



รูปที่ 3-25 แสดงองค์ประกอบของหน้าจอหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Cartoonlist**

ส่วนนี้เป็นส่วนที่ใช้แสดงประเภท และ ชื่อของการ์ตูน ในประเภทนั้นๆ โดยจะต้องทำการ กด Click ไปที่ประเภทของการ์ตูนที่ต้องการจะดูรายชื่อจากนั้นรายชื่อของการ์ตูนในประเภทที่เลือกก็จะแสดงเป็น รายการขึ้นมาเพื่อให้ลูกค้าเลือกเพื่อดูรายละเอียดของการ์ตูนเล่มนั้น

- **Details**

หน้าจส่วนนี้ใช้ในการแสดงรายละเอียดของหน้าจอหรือกิจกรรมต่างๆ ของผู้ใช้ เช่น การ Shop ,การเพิ่มหรือเอาของออกจากตะกร้า

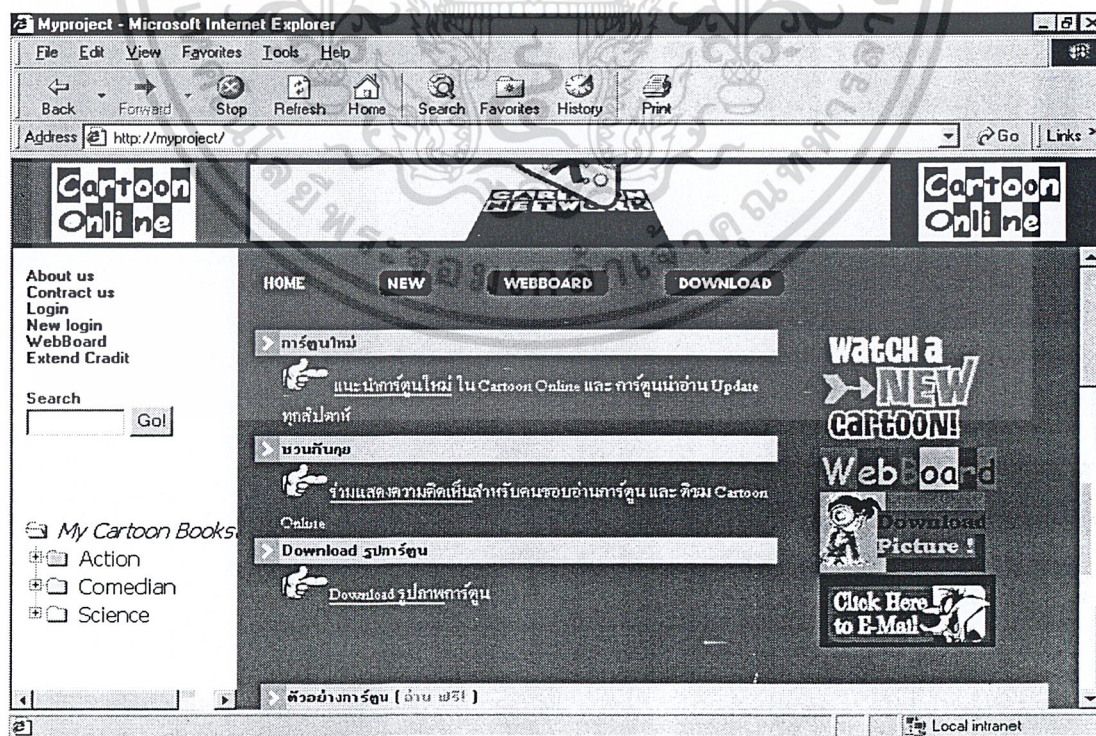
### 3.2.6 หน้าจอของระบบจริง

#### 3.2.6.1 หน้าจอ Home

แบ่งเป็น 4 ส่วน ประกอบไปด้วย Logo, CartoonList, Details, Contact หน้าจอนี้เป็นหน้าจอแรก ในการเข้ามาใช้งานระบบของลูกค้า ส่วนของ Details ใน หน้าจอ Home นี้จะแสดงเพื่อดึงดูดลูกค้าให้เข้ามาดูซึ่งจะประกอบไปด้วย

- 1.การแนะนำการ์ตูนใหม่ และ การ์ตูนที่น่าอ่านหรือเป็นที่นิยม
- 2.Webboard เพื่อให้ลูกค้าได้มี โอกาสเข้ามาแสดงความคิดเห็น หรือ ดิชม
- 3.ส่วนของการ Download รูปภาพการ์ตูน
- 4.ตัวอย่างหนังสือการ์ตูนที่ให้อ่านฟรี

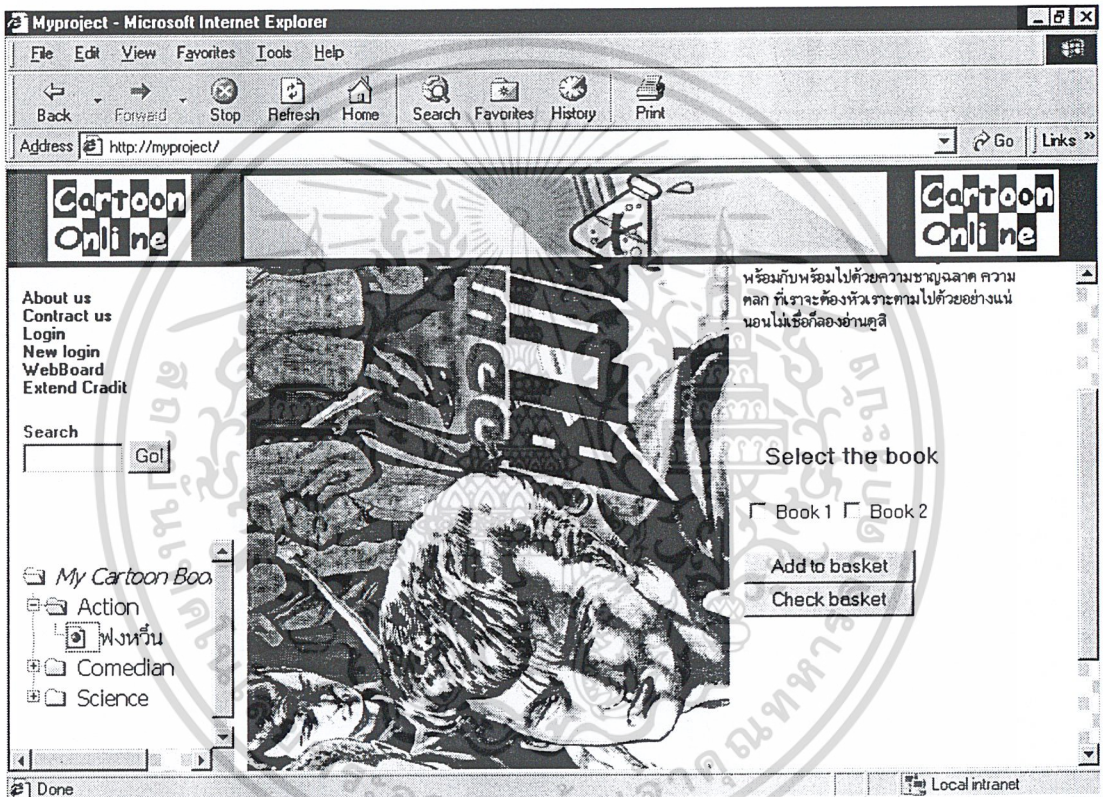
ในส่วนบนของหน้าจอเป็น Menu ให้ลูกค้าสามารถเลือกไปที่หน้าจอต่างๆ ได้สะดวก หน้าจอ Home แสดงในรูปที่ 3-26



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการรูปที่ 3-26 แสดงหน้าจอ Home อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.6.2. หน้าจอการดูรายละเอียดการ์ตูน

หน้าจอนี้ใช้แสดงรายละเอียดของการ์ตูนแต่ละเรื่อง ซึ่งเกิดขึ้นจากการที่ลูกค้า Click ชื่อของการ์ตูนที่ต้องการดูรายละเอียด หน้าจอนี้จะประกอบไปด้วย เนื้อเรื่องย่อของการ์ตูนเล่มนั้นๆ และ CheckBox สำหรับเลือกเช่าการ์ตูนเล่มที่ต้องการ นอกจากนี้ผู้ใช้งานสามารถไปที่หน้าจอตรวจสอบตะกร้า และสามารถเพิ่มการ์ตูนลงในตะกร้าได้โดยการกดปุ่ม Add to basket หน้าจอการดูรายละเอียดการ์ตูนแสดงในรูปแบบที่ 3-27

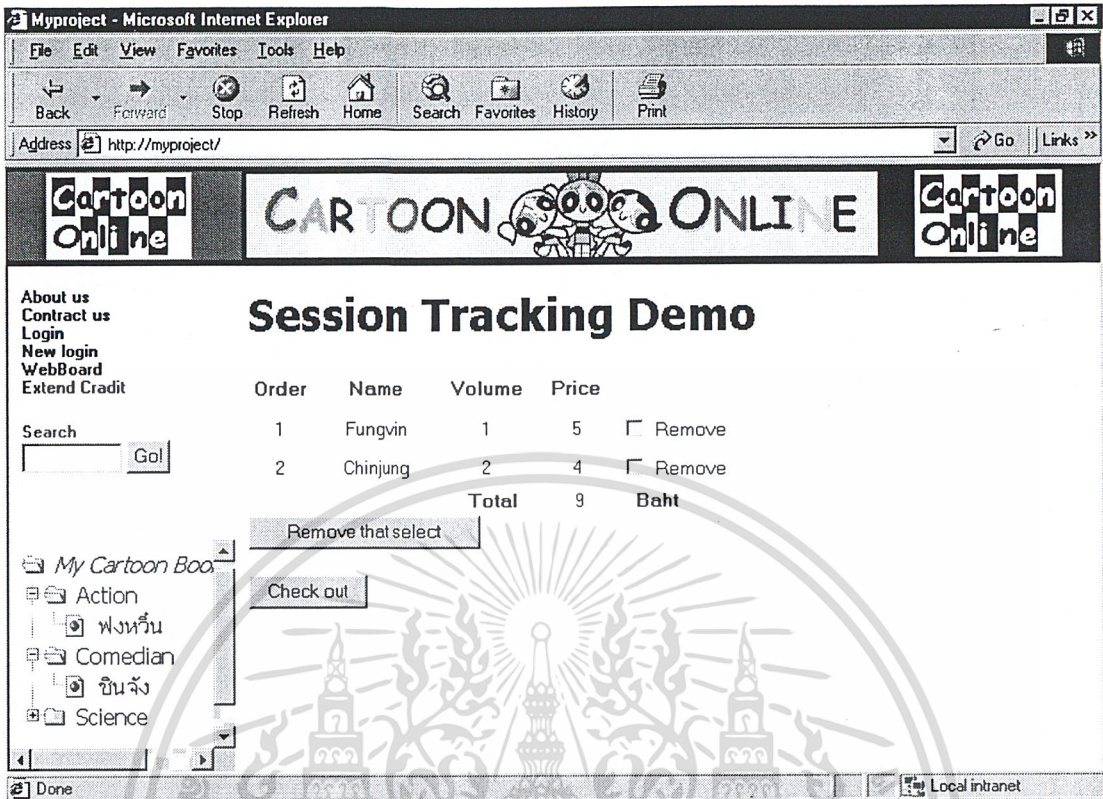


รูปที่ 3-27 แสดงหน้าจอรายละเอียดการ์ตูน

### 3.2.6.3 หน้าจอ Basket

หน้าจอนี้เกิดจากการที่ลูกค้ากดปุ่ม เพิ่มหนังสือลงในตะกร้า หรือ กดปุ่มตรวจสอบตะกร้า หน้าจอ Basket ประกอบไปด้วยรายการของหนังสือการ์ตูนที่ลูกค้าเลือกใส่ตะกร้าไว้ ซึ่งมีรายละเอียดราคาเช่าของหนังสือการ์ตูนแต่ละเล่ม และ ราคารวม พร้อมทั้ง Input CheckBox เพื่อเลือกเอาหนังสือออกจากตะกร้า ถ้าลูกค้าต้องการเอาหนังสือเล่มใดออกจากตะกร้าก็จะทำการ Check Box ที่หนังสือเล่มนั้น แล้วกดปุ่ม Remove ถ้าต้องการชำระค่าเช่าลูกค้าก็จะกดปุ่ม Check Out หน้าจอ Basket แสดงในรูปแบบที่ 3-28

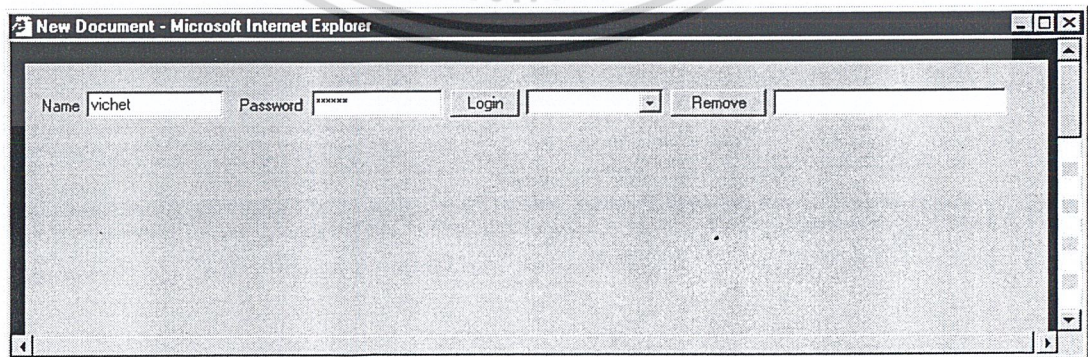
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-28 แสดงหน้าจอบasket

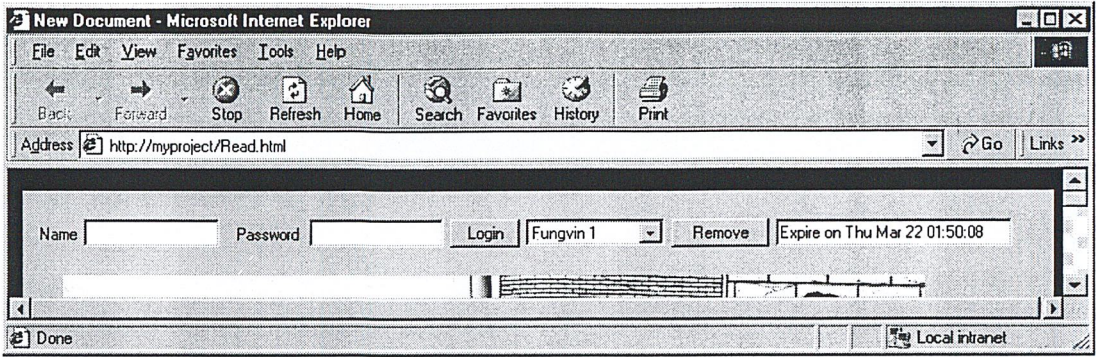
#### 3.2.6.4 หน้าจออ่าน

เป็นหน้าจอการ login ของลูกค้าเพื่อเข้ามาอ่านหนังสือการ์ตูนที่ได้เช่าไว้ในหน้าจอนี้ประกอบไปด้วย รายละเอียดในการอ่านของลูกค้าแต่ละคน เมื่อลูกค้าใส่ User และ Password แล้วกด Login ระบบจะแสดงรายการการ์ตูนที่ User คนนี้เช่าไว้ โดยจะแสดงเวลา Expire ของหนังสือแต่ละเล่มไว้ด้วย ถ้าต้องการที่จะ Remove หนังสือออกจากรายการก็สามารถกด Remove ได้



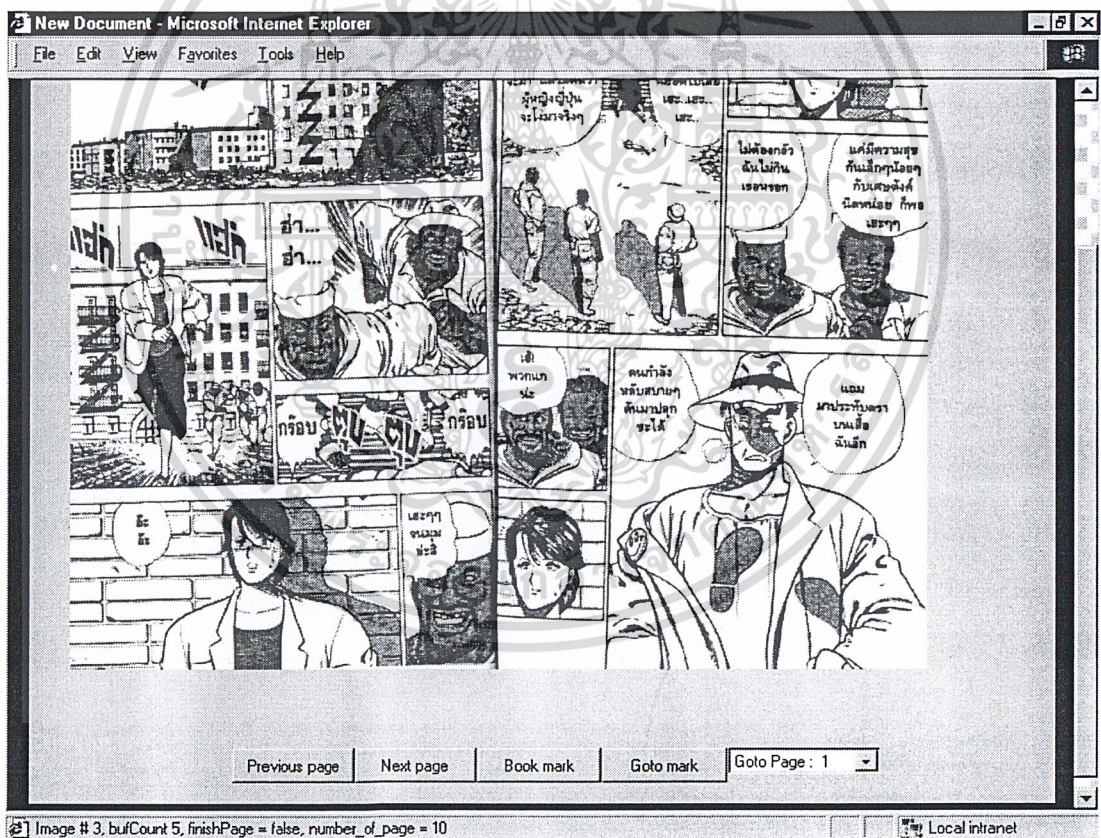
รูปที่ 3-29 แสดงการ Login เข้ามาอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-30 แสดงรายการเข้าและเวลา Expire

ในส่วนล่างของหน้าจออ่านจะเป็นส่วนของการควบคุมการเปลี่ยนรูปภาพในการอ่านของ User



รูปที่ 3-31 แสดงหน้าจอการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.6.5 หน้าจอการซื้อ Package ใหม่

หน้าจอนี้ประกอบไปด้วยส่วนประกอบต่าง ๆ ดังนี้ ส่วนบนสุดเป็นส่วนที่แสดงเงื่อนไขของการเช่า ตัวอย่างเช่น ในการเช่าลูกค้าจะมีเวลาในการอ่านครั้งละ 3 วันส่วนต่อมาเป็นส่วนให้ลูกค้ากรอกรายละเอียด UserName และ Password ของ Package ที่ต้องการซื้อ ต่อจากนั้นเป็นส่วนของการใส่รายละเอียดของบัตรเครดิตที่ใช้ในการซื้อ Package ส่วนล่างสุดเป็นราคาของ Package ที่ต้องการซื้อ

กรณการกรอกรายละเอียด

เงื่อนไขในการใช้บริการ

1. ผู้ใช้บริการจะมีเวลาในการอ่านหนังสือแต่ละเล่ม 3 วัน หลังจากนั้นหนังสือจะถูก Expire ไม่สามารถอ่านได้อีก
2. ถ้าหากผู้ให้บริการขาดการติดต่อหรือขาดการใช้งานไปเกิน 3 เดือน จะหมดสิทธิ์ในการใช้บริการ

Login name  \*ชื่อและรหัสผ่านนี้ใช้เข้าและอ่านการ์ตูนครับ

Password (รหัสผ่าน)  \*

Confirm password (พิมพ์รหัสผ่านซ้ำ)  \*

ชื่อผู้ใช้นับบัตรเครดิต  \*

เลขที่บัตรเครดิต  \*

วันหมดอายุของบัตรเครดิต (mm/dd/yyyy) Jan / 1 / 2001 \*

ขนาดของ Package ที่ต้องการ  บาท \*

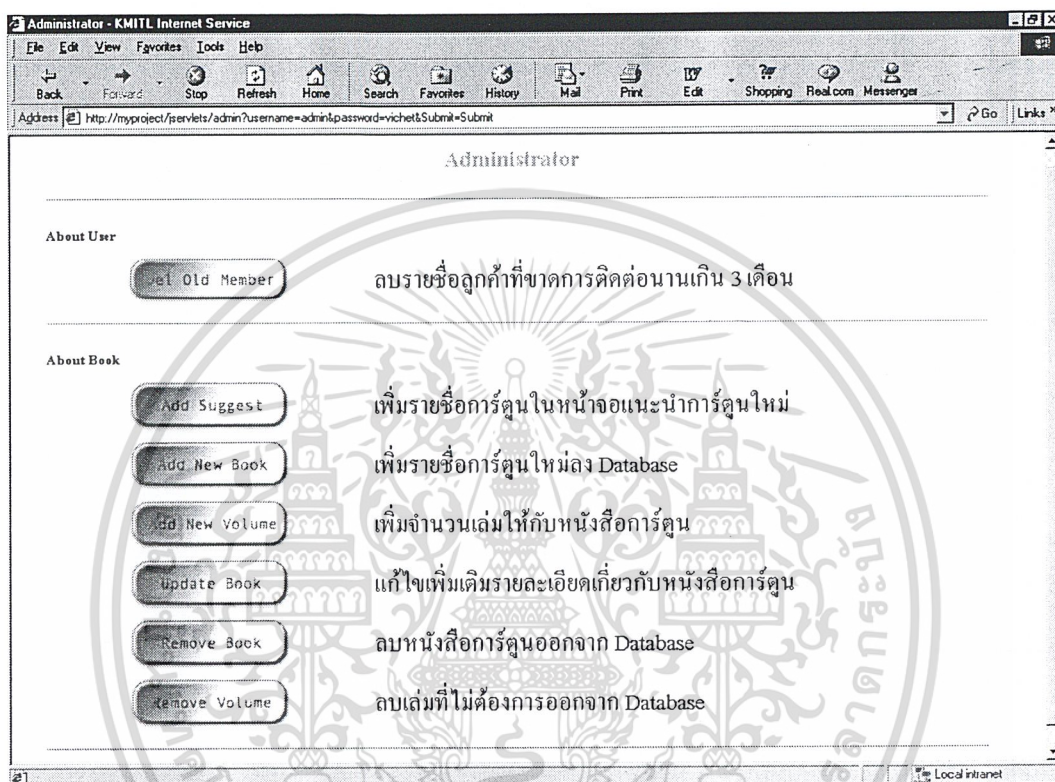
Done Local intranet

รูปที่ 3-32 แสดงหน้าจอการซื้อ Package ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.6.6 หน้าจอของผู้ดูแลระบบ

หน้าจอของผู้ดูแลระบบประกอบด้วยส่วนสำคัญ ๆ คือ การลบรายชื่อลูกค้า หรือ Package ที่ขาดการติดต่อมานานเกิน 3 เดือน ส่วนของการเพิ่มรายชื่อและรายละเอียดของหนังสือการ์ตูนเล่มใหม่ลงในฐานข้อมูล นอกจากนี้ยังมีส่วนของการลบการ์ตูนออกจากฐานข้อมูล หรือการเปลี่ยนแปลงรายละเอียด เช่น การเปลี่ยนแปลงราคาเช่า



รูปที่ 3-33 แสดงหน้าจอผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### สรุป

#### 4.1 ปัญหาต่างๆ ที่เกิดขึ้นในการพัฒนาระบบ

ในการพัฒนาระบบแบบเชิงวัตถุ(Object-Oriented) จะประกอบไปด้วยการมองระบบทั้งหมดเป็นแบบเชิงวัตถุตั้งแต่ขั้นตอนแรกของการพัฒนาหรือที่เรียกว่า Object-Oriented Analysis หรือการวิเคราะห์ระบบในเชิงวัตถุ จากนั้นเมื่อทำการวิเคราะห์แล้วก็จะมาทำขั้นตอนของการออกแบบเรียกว่า Object-Oriented Design หลังจากการออกแบบผ่านพ้นไปก็จะมาทำการสร้างระบบตามที่ได้ออกแบบไว้ เพราะว่าถ้าจะให้ดำเนินการเขียนโปรแกรมของระบบทั้งหมดในเชิงวัตถุโดยไม่ผ่านกระบวนการของการวิเคราะห์และออกแบบตามที่ได้ออกไปแล้วจะเป็นไปไม่ได้อย่างแน่นอน แม้กระทั่งผ่านการวิเคราะห์และออกแบบมาแล้วก็ตามบางครั้งเมื่อจะเขียนโปรแกรมขึ้นมาจริงก็ยังคงต้องมีการกลับไปทำกระบวนการต่างๆ ซ้ำใหม่อยู่เหมือนกัน เพราะว่าในการวิเคราะห์และออกแบบเชิงวัตถุนั้นจะต้องอาศัยการพัฒนาแบบ Iterative คือเป็นไปในลักษณะซ้ำๆ ต้องมีการแก้ไขปรับปรุงให้ดีขึ้นอยู่ตลอดเวลา ดังนั้นจึงเป็นการยากอยู่พอสมควรในการพัฒนาระบบเชิงวัตถุ

เนื่องจากโครงการนี้เป็นการพัฒนาเว็บซึ่งจะเป็นการยากอย่างมากในการมองระบบให้เป็นวัตถุเพราะจะมีความแตกต่างจากที่ได้เคยศึกษามา และที่สำคัญคือเครื่องมือที่ใช้ในการพัฒนานี้ยังไม่สนับสนุนการออกแบบบนเว็บโดยตรงทีเดียว เช่น ไม่มีไดอะแกรมให้เห็นถึงการออกแบบในแต่ละเพจจึงต้องมีการใช้เครื่องมือในการวาดรูปอื่นๆ เข้ามาใช้งานแทนเพื่อที่จะสื่อให้เห็นถึงการเปลี่ยนแปลงหน้าจอต่างๆ ของระบบทั้งหมดได้

#### 4.2 เปรียบเทียบกับการพัฒนาระบบในแนวทางเดิม

การพัฒนาระบบเชิงวัตถุจะพยายามมองทุกอย่างให้ออกมาเป็นวัตถุ ภายในตัวของวัตถุก็จะประกอบไปด้วยคุณสมบัติของวัตถุนั้นๆ และยังมีส่วนของพฤติกรรมของวัตถุ ซึ่งการมองทุกอย่างออกเป็นวัตถุจะทำให้เรามองระบบได้กว้างมากขึ้นเหมาะสำหรับการพัฒนาระบบใหญ่ๆ มากกว่าการพัฒนาในระบบเก่าซึ่งจะมีความซับซ้อนอย่างมาก แต่ในขณะเดียวกันการพัฒนางานเล็กๆ นั้นการมองเป็นวัตถุก็จะดูยากเกินไปที่จะทำได้ การมองเป็นโครงสร้างยังคงมีความเหมาะสมอยู่กับงานเล็กๆ เท่านั้น

#### 4.3 แนวทางในการพัฒนาระบบ

การพัฒนาระบบพหุขั้วเล็กทรอนิกส์โดยใช้แนวทางของ Object-Oriented นั้นนับว่ามีความสะดวกและมีความยืดหยุ่น ยกตัวอย่างเช่นเมื่อเราได้ขายสินค้าชนิดใดชนิดหนึ่งออกไปแล้วเกิดขายไม่ได้เป็นไปตามเป้าหมายที่ตั้งไว้และอยากจะเปลี่ยนสินค้าขึ้นมา จะเห็นได้ชัดเจนว่าถ้าเป็นการเขียนโปรแกรมแบบเก่าแล้วจะต้องมีการเขียนขึ้นมาใหม่ทั้งหมด แต่ถ้าเป็นแบบเชิงวัตถุแล้วการเปลี่ยนสินค้าไปนั้นจะไม่มีผลกระทบกับการออกแบบระบบใหม่ทั้งหมดอย่างแน่นอนสามารถนำเอาการออกแบบเก่ามาใช้งานได้ทันที กรณีนี้เป็นการนำของเก่ากลับมาใช้ประโยชน์(Reusable) ซึ่งเป็นข้อดีของการออกแบบเชิงวัตถุ

เอกสารนี้ออกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ UML เองนั้นมีส่วนช่วยในการพัฒนาเป็นอย่างมาก เนื่องจากมีแบบจำลองที่สามารถสนับสนุนสถานะแวดล้อมในการพัฒนาระบบเชิงวัตถุทุกๆ ด้าน ไม่ว่าจะเป็นด้านการรวบรวมความต้องการ ด้านการวิเคราะห์ ด้านการออกแบบ หรือแม้แต่ด้านการสร้างระบบ และยิ่งในปัจจุบันมีเครื่องมือที่สนับสนุนการพัฒนาระบบเชิงวัตถุเพิ่มขึ้นเรื่อยๆ และมีผลทำให้การพัฒนาระบบเชิงวัตถุมีแนวโน้มเป็นการพัฒนาระบบในอนาคตต่อไปได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

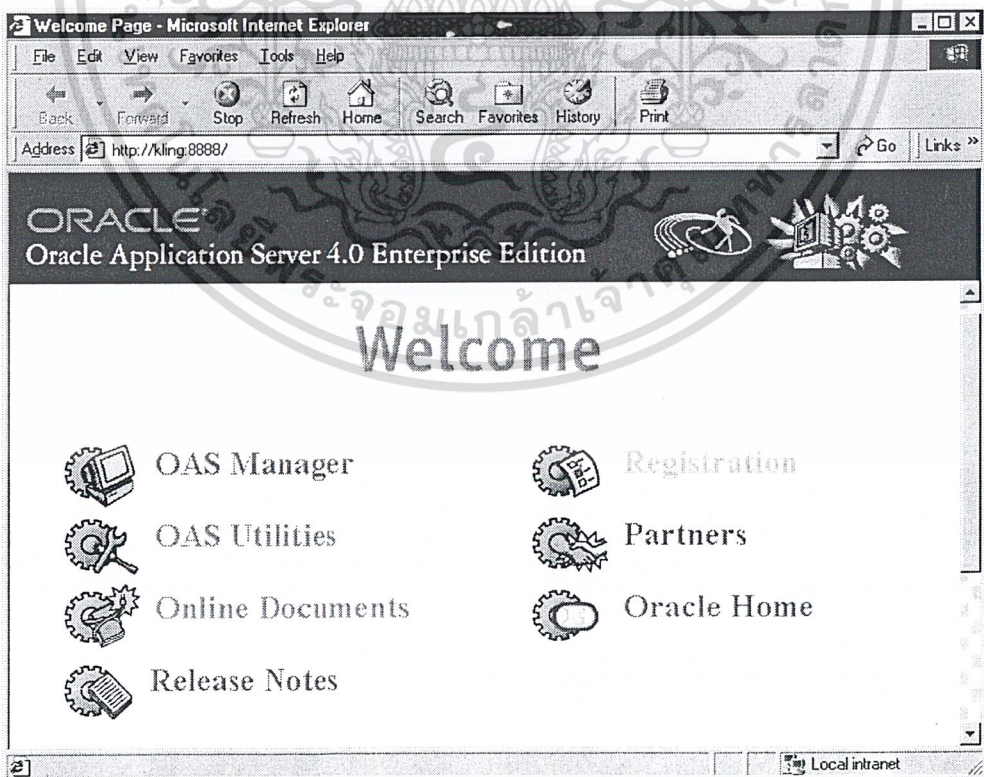
## การสร้าง JServlet application และ Cartridge

ในส่วนนี้จะเป็นการอธิบายถึงการสร้าง JServlet application และ Cartridge ที่อยู่บน ส่วนของ Oracle Application Server (OAS) ก่อนที่จะดำเนินการขั้นตอนต่อไปจะต้องทำการเรียกโปรแกรม OAS ขึ้นมาก่อนโดยการล็อกอิน Admin จากนั้นจึงดำเนินการเป็นขั้นตอนต่างๆ ต่อไป



รูปที่ ก-1 แสดงการล็อกอินเข้า OAS 4.0.8.1

1. เมื่อ Login เข้ามาแล้วจะเข้าสู่หน้าจอ Welcome Page

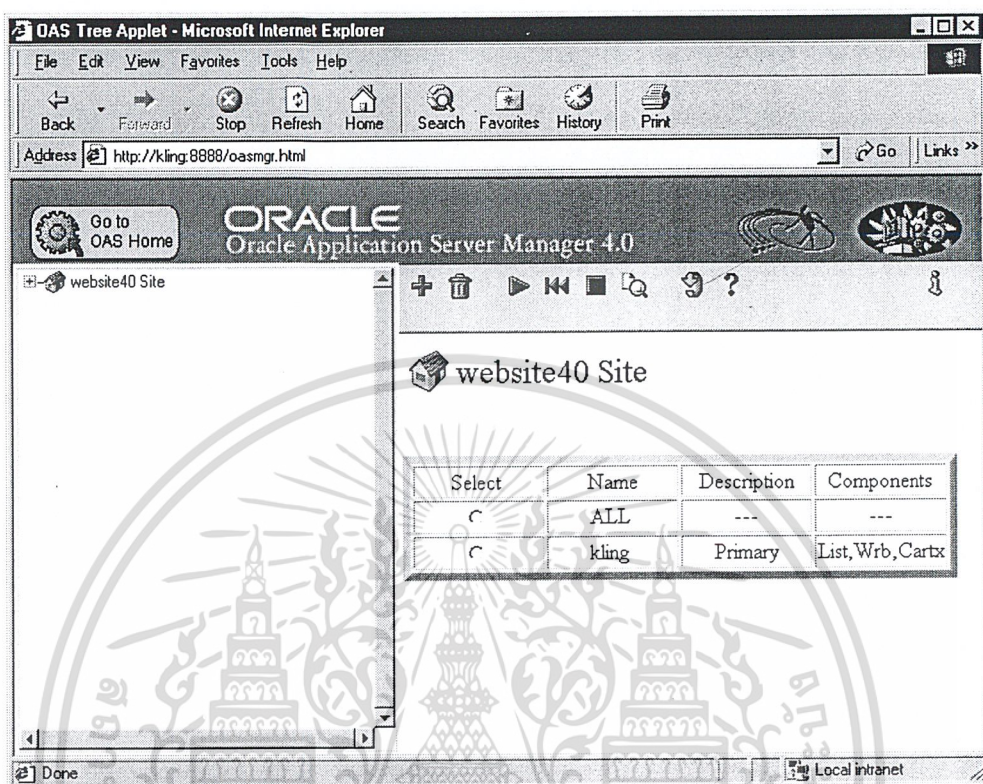


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเชิงพาณิชย์เท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. นำ Mouse ไปคลิกที่



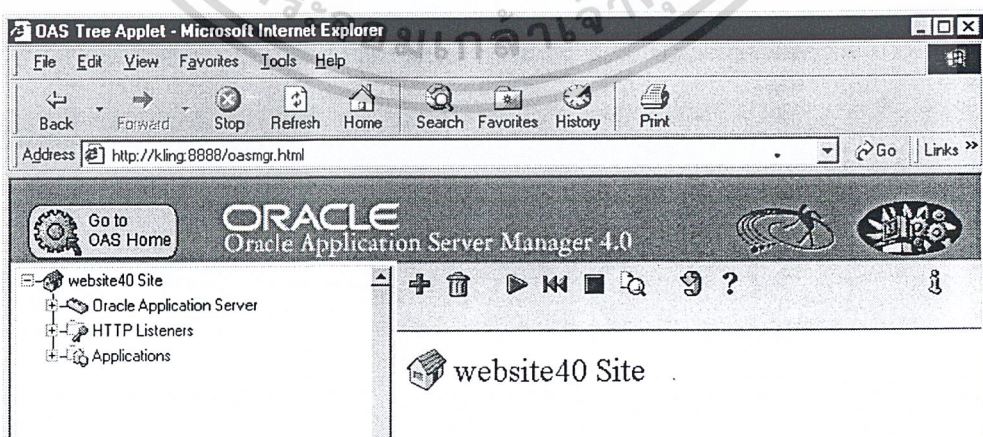
OAS Manger จะปรากฏหน้าจอดังรูป



รูปที่ ก-3 แสดงหน้าจอ OAS Manager


3. จากนั้นให้นำ Mouse ไปคลิกที่เครื่องหมาย  ก็จะเกิดหัวข้อรายการขึ้นมา 3

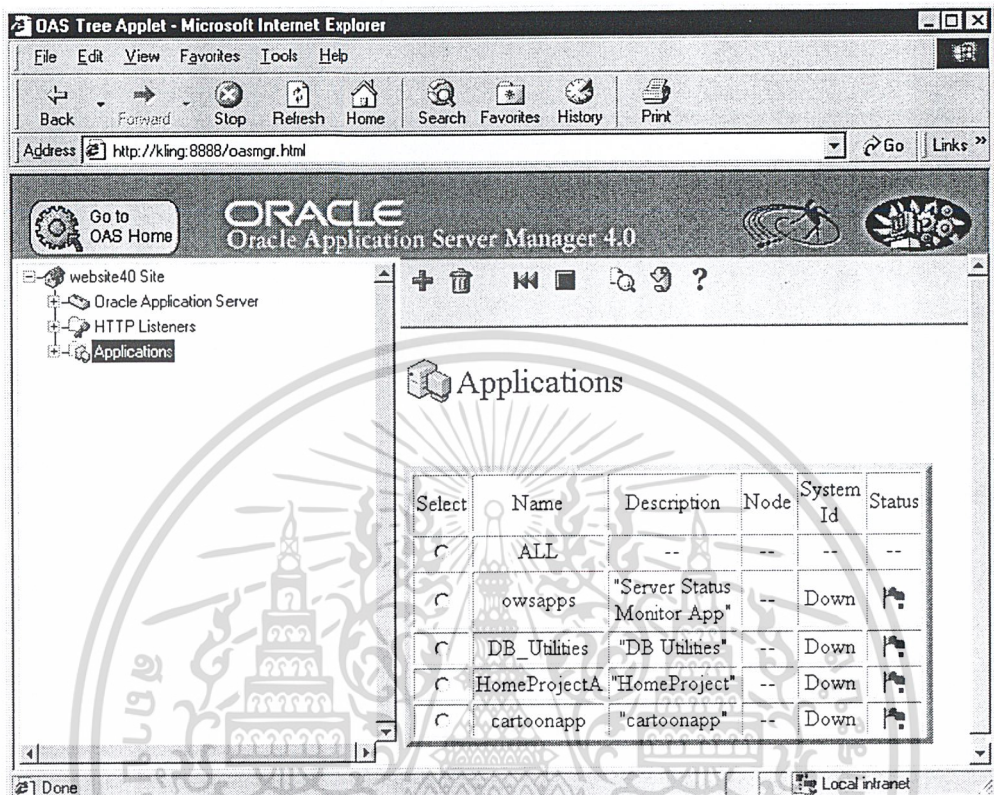
หัวข้อ คือ Oracle Application Server , HTTP Listeners , Applications



รูปที่ ก-4 แสดงหัวข้อ 3 รายการ

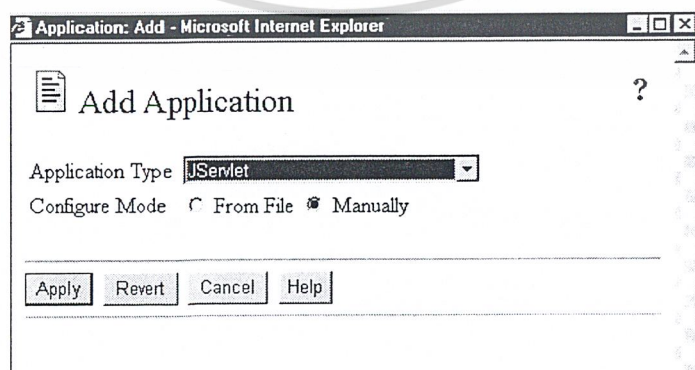
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ให้คลิกที่ Applications อย่าคลิกที่เครื่องหมาย  เพราะจะทำให้ไม่เห็นรายการต่างๆทางด้านขวามือดังรูป



รูปที่ ก-5 แสดงรายการใน Applications

5. ในเฟรมทางด้านขวามือให้คลิกที่  จะปรากฏ Add dialog ขึ้นมาดังรูป โดยเลือก Application Type = JServlet , Configure Mode = Manually จากนั้นคลิก Apply



รูปที่ ก-6 แสดง Add Dialog

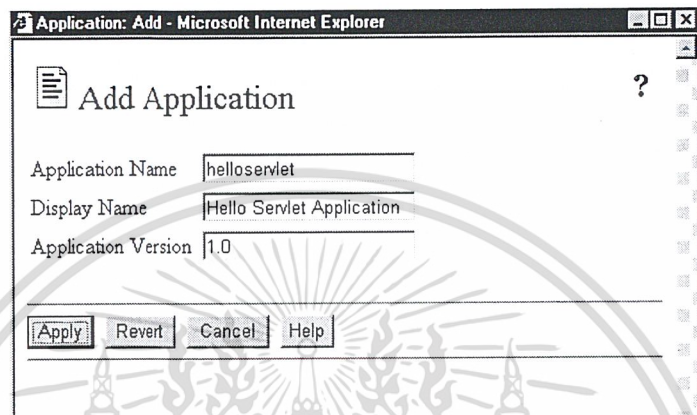
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. จากนั้นจะปรากฏ Dialog มาให้ใส่รายการต่างๆ ดังนี้ แล้วคลิก Apply

Application Name = helloservlet เป็นชื่อที่ใช้อ้างถึง Application

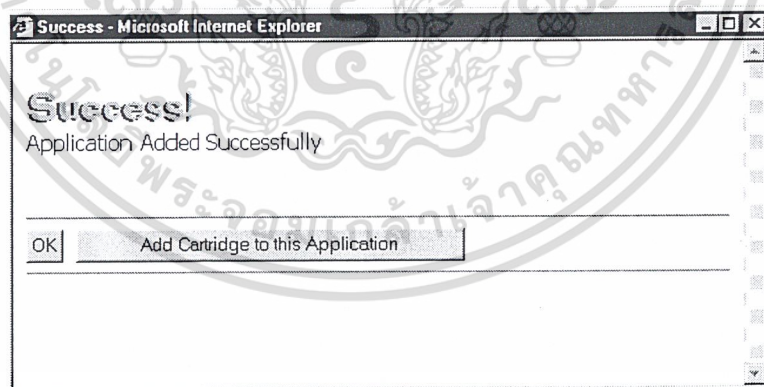
Display Name = Hello Servlet Application เป็นชื่อที่ใช้จัดการภายใน Form

Application Version = 1.0 ระบุ Version ของ Application ของเรา



รูปที่ ก-7 แสดง Add Application

7. แสดงหน้าจอ Success! จากนั้นให้ทำการเพิ่ม Cartridge โดยคลิกที่ “Add Cartridge to this Application”



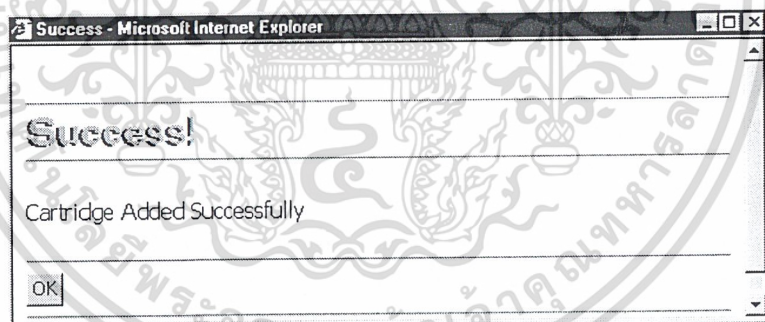
รูปที่ ก-8 แสดง Success Application Dialog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. จะปรากฏหน้าจอการ Add A Cartridge เพื่อที่เราจะได้เพิ่ม Cartridge ให้กับ Application โดยจะระบุค่าต่างๆ ตามต้องการดังรูป

รูปที่ ก-9 แสดง Add A Cartridge

9. เมื่อคลิก Apply แล้วจะปรากฏ Success ดังรูป



รูปที่ ก-10 แสดง Cartridge Added Successfully

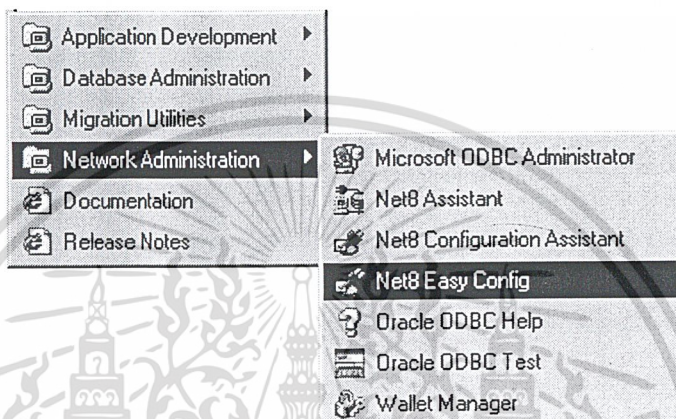
และขณะนี้เราก็ได้มี helloservlet ซึ่งบรรจุอยู่ใน JServlet Cartridge

## ภาคผนวก ข การใช้งาน Oracle8i

### 1. การทดสอบการเชื่อมต่อกับฐานข้อมูล

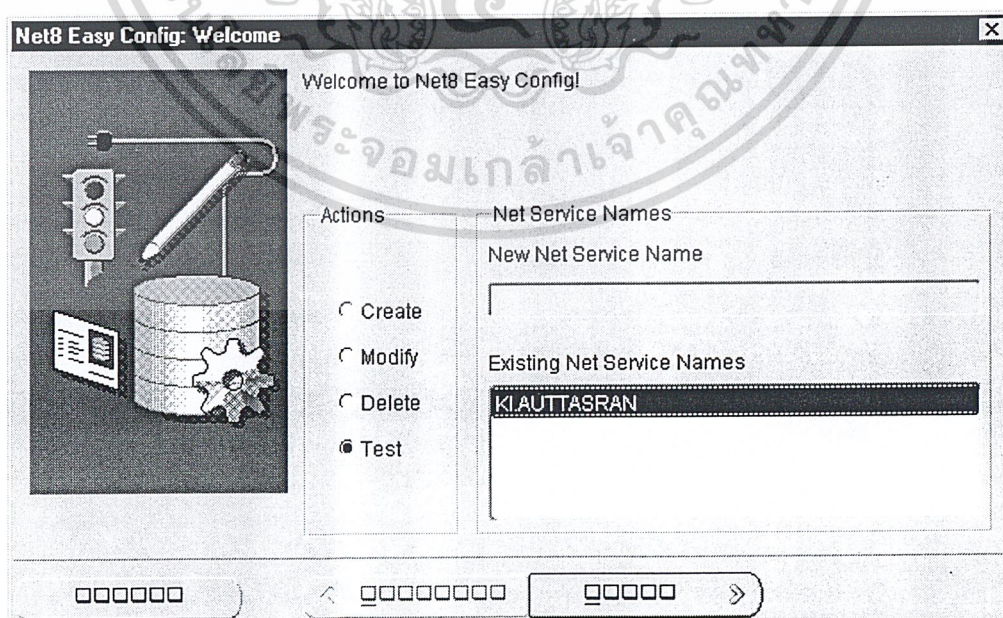
ทดสอบเพื่อดูว่าฐานข้อมูลที่จะใช้งานมีการทำงาน และ การเชื่อมต่อปกติดี อยู่หรือไม่ถ้าปกติก็จะแสดงข้อความ The connection was successful.

#### 1.1 เลือก Menu Net8 Easy Config



รูปที่ ข-1 แสดงการเลือก Net8 Easy Config

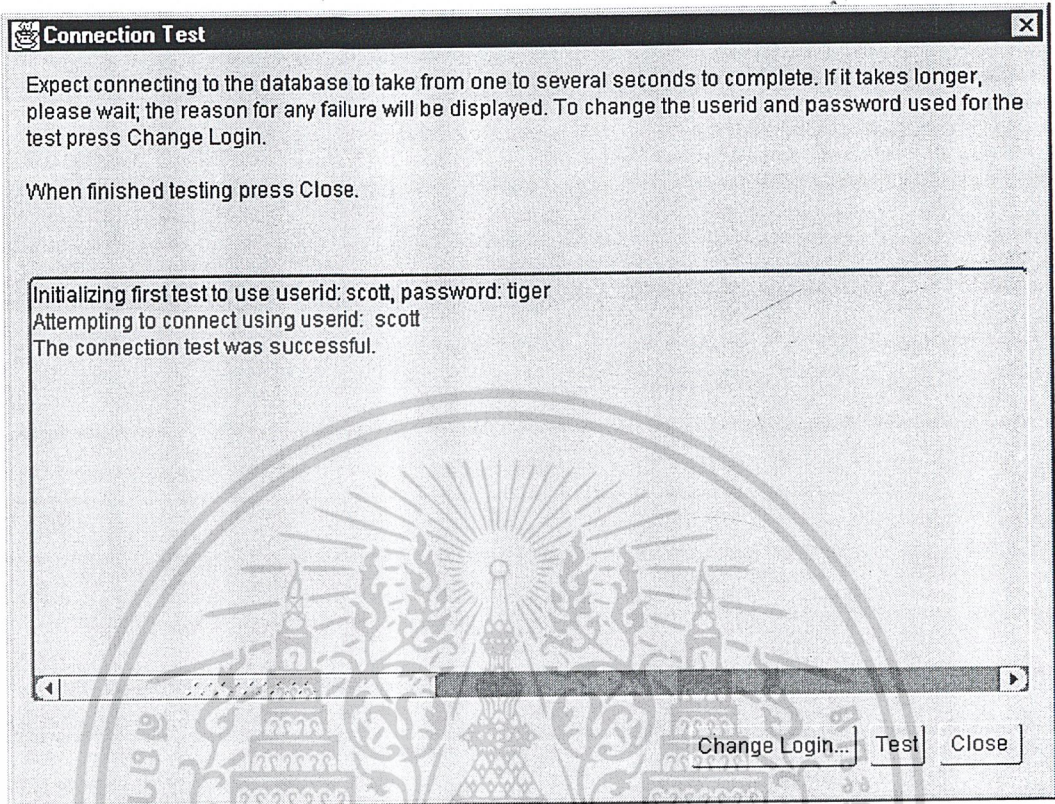
#### 1.2 เลือกทำการ Test และ เลือกชื่อฐานข้อมูลที่ต้องการ Test



รูปที่ ข-2 แสดงการเลือกฐานข้อมูลที่ต้องการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อคุณได้เห็นนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

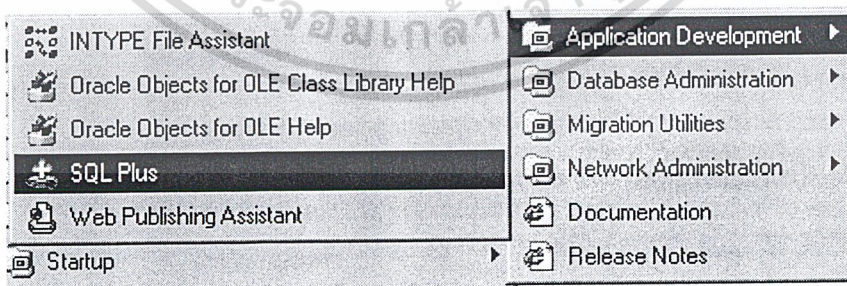
### 1.3 แสดงการทดสอบการเชื่อมต่อสมบูรณ



รูปที่ ข-3 แสดงการทดสอบสมบูรณ

## 2.การใช้งาน SQL Plus ใน Oracle8i

### 2.1 เลือก Menu SQL Plus



รูปที่ ข-4 แสดงการเลือก SQL Plus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 การ ใส่ User Name ,Password และ Host String ก่อนเข้า SQL Plus

**Log On**

User Name:

Password:

Host String:

OK Cancel

รูปที่ ข-5 แสดงการ Login เข้า SQL Plus

## 2.3 ตัวอย่างการใช้งานคำสั่ง SQL ใน SQL Plus

```

Oracle SQL*Plus
File Edit Search Options Help
SQL*Plus: Release 8.1.5.0.0 - Production on ศ. เม.ย. 7 11:18:03 2000
(c) Copyright 1999 Oracle Corporation. All rights reserved.

Connected to:
Oracle8i Release 8.1.5.0.0 - Production
With the Java option
PL/SQL Release 8.1.5.0.0 - Production

SQL> select * from tab;

TNAME                                TABTYPE CLUSTERID
-----                                -
BONUS                                 TABLE
CARTOON                               TABLE
CARTOON_DETAILS                       TABLE
CARTOON_NEW                           TABLE
DEPT                                   TABLE
EMP                                    TABLE
READ                                   TABLE
REPLY                                  TABLE
SALGRADE                              TABLE
SUBJECT_POST                          TABLE
USERS                                  TABLE

11 rows selected.

SQL> |

```

รูปที่ ข-6 แสดงการใช้งาน SQL Plus

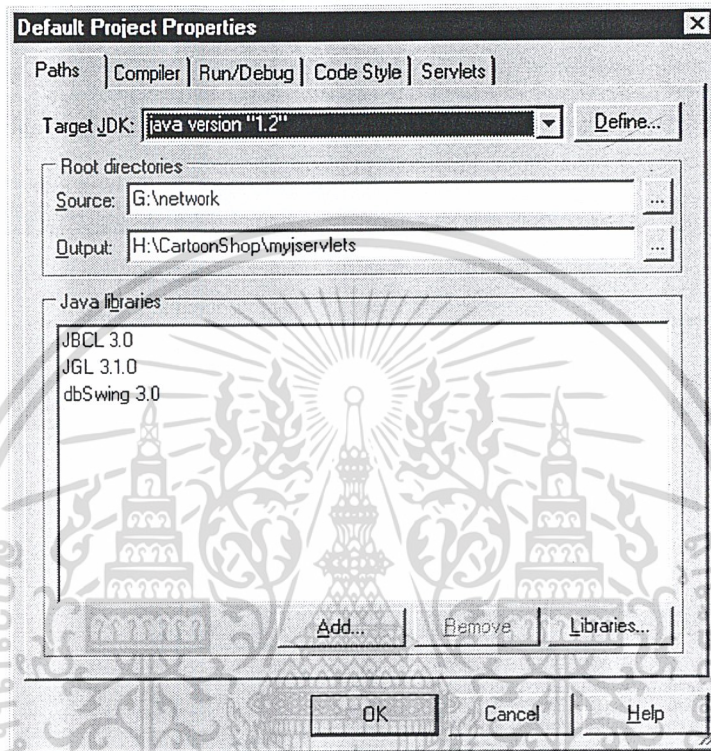
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

## การจัดการ libraries ใน JBuilder เพื่อใช้เขียน JAVAServlets

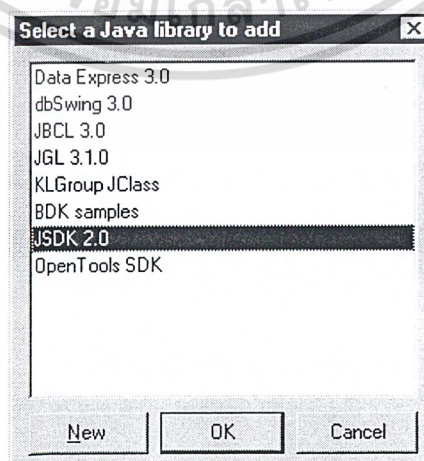
## 1. การเพิ่ม Libraries ที่มีอยู่แล้ว

1.1 เข้าโปรแกรม JBuilder แล้วเลือก Project -&gt; Default Properties



รูปที่ ก-1 แสดงรายการ Libraries

1.2. เลือก Add แล้วเลือก JSDK 2.0 จากนั้นกด OK

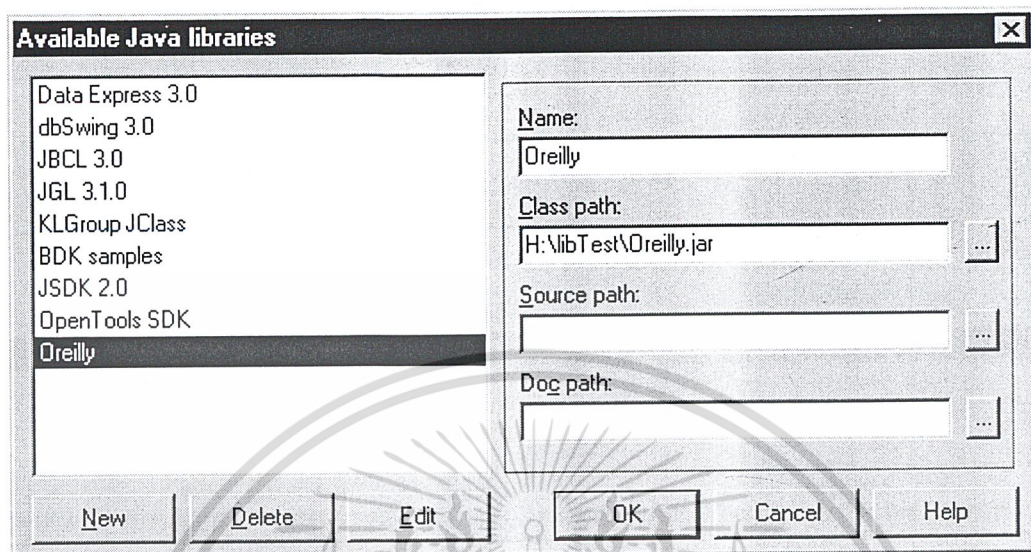


รูปที่ ก-2 แสดงการเลือก Libraries

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

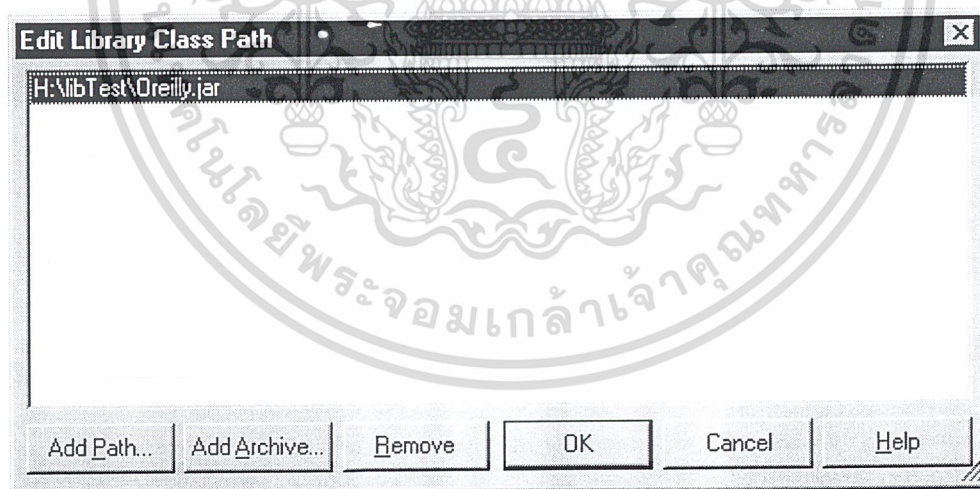
## 2. การเพิ่ม Libraries ใหม่

2.1 เข้าโปรแกรม JBuilder แล้วเลือก Project -> Default Properties -> Libraries...



รูปที่ ก-3 แสดงการเพิ่ม Libraries ใหม่

2.2 เลือกปุ่ม ... ในส่วนของ Class path:



รูปที่ ก-4 แสดงการเลือก Add Archive...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ง

## ตัวอย่าง Source Code JAVAServlets

## 1. ตัวอย่างการแสดงผลข้อความ Hello World

```

import java.io.* ;
import javax.servlet.* ;
import javax.servlet.http.* ;

public class HelloServlet extends HttpServlet {

    public void doGet (
        HttpServletRequest request,
        HttpServletResponse response
    ) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HTML><BODY>");
        out.println("<center><H1>Hello World</H1></center>");
        out.println("</BODY></HTML>");
    } // doGet
} // HelloServlet

```

รูปที่ ง-1 แสดง Source Code ของ Hello World



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ ง-2 แสดงผลการทำงานของ Hello World หน้าไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. ตัวอย่าง Source Code การติดต่อฐานข้อมูล

### 2.1 การอ่านข้อมูลจากฐานข้อมูล

```

import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class DBTest extends HttpServlet
implements SingleThreadModel {
    private Connection conn = null;
    public void init( ServletConfig config ) throws ServletException {
        super.init(config);
        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            conn = DriverManager.getConnection("jdbc:oracle:thin:@jim:1521:PROJECT","scott","tiger");
        } catch ( SQLException e ) {
            System.err.println("Could not establish connection.");
        }
        catch ( ClassNotFoundException e ) {
            System.err.println("Could not load database driver.");
        }
    } // init
    //-----Get Method-----
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        //-----Data Base-----
        try {
            Statement stmt = conn.createStatement();
            ResultSet rset = stmt.executeQuery("select NAME from TEST");

```

รูปที่ ๓-3 แสดง Source Code การอ่านข้อมูลจากฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----Response User-----
res.setContentType("text/html;charset=windows-874");
PrintWriter out = res.getWriter();
out.println("<HTML>");
out.println("<BODY>");

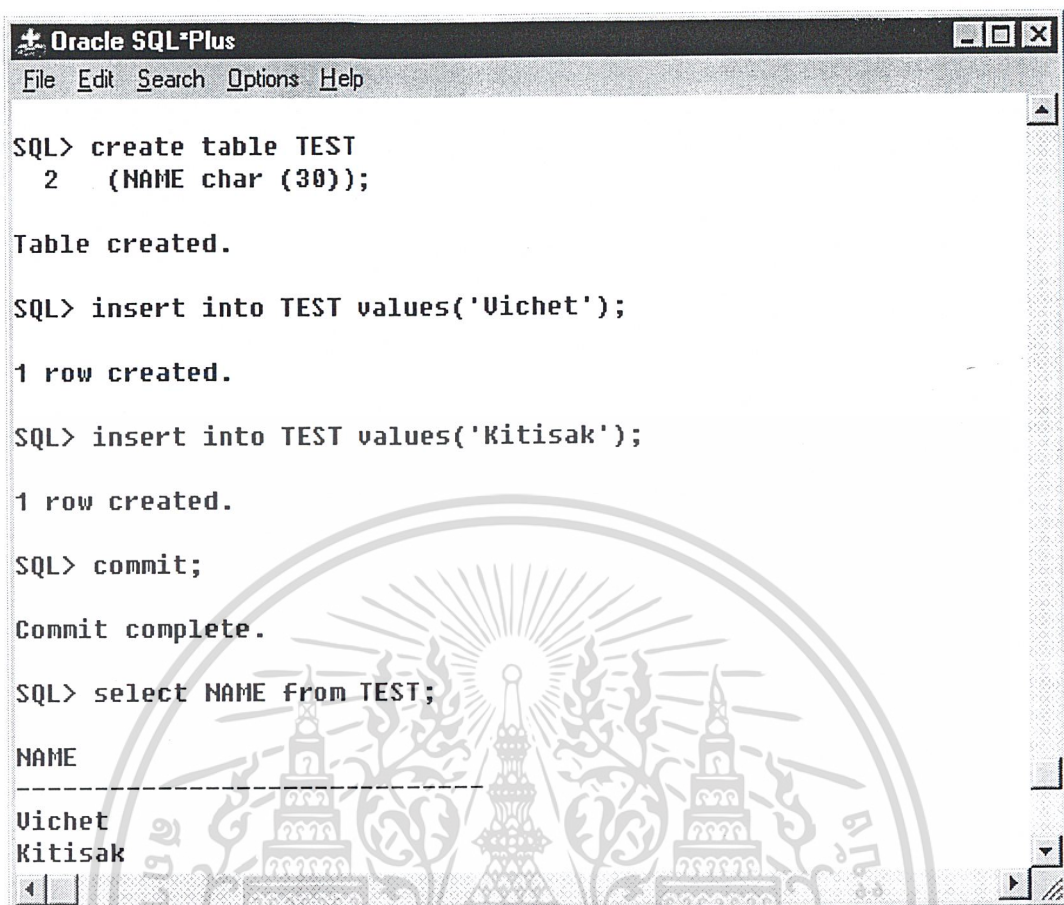
while(rset.next())
    out.println("<center><H1>"+rset.getString("NAME")+"</H1></center><br>");
out.println("</body>");
out.println("</html>");
} catch ( SQLException e ) {
    System.err.println("A database error occurred.");
}
}

//-----Distroy-----
public void destroy() {
try {
// close the database connection
if ( conn != null) conn.close();
} catch (SQLException e) {
    System.err.println("Error closing database connection.");
}
} // destroy
}

```

รูปที่ ง-4 แสดง Source Code การอ่านข้อมูลจากฐานข้อมูล  
(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

Oracle SQL*Plus
File Edit Search Options Help

SQL> create table TEST
2 (NAME char (30));

Table created.

SQL> insert into TEST values('Vichet');

1 row created.

SQL> insert into TEST values('Kitisak');

1 row created.

SQL> commit;

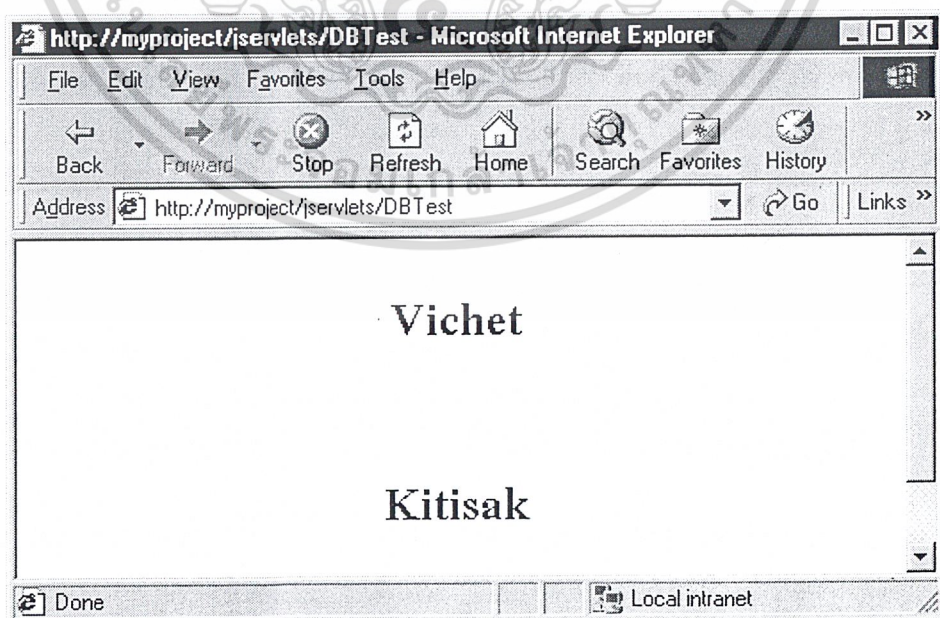
Commit complete.

SQL> select NAME from TEST;

NAME
-----
Vichet
Kitisak

```

รูปที่ 3-5 แสดงการสร้างตารางเพื่อทดสอบการติดต่อฐานข้อมูล



รูปที่ 3-6 แสดงผลการทำงาน การติดต่อฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 การ Update ฐานข้อมูล

```

import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class update extends HttpServlet
implements SingleThreadModel {
private Connection conn = null;
public void init( ServletConfig config ) throws ServletException {
super.init(config);
try {
Class.forName ("oracle.jdbc.driver.OracleDriver");
conn = DriverManager.getConnection("jdbc:oracle:thin:@jim:1521:project","scott","tiger");
} catch ( SQLException e ) {
System.err.println("Could not establish connection.");
} catch ( ClassNotFoundException e ) {
System.err.println("Could not load database driver.");
}
} // init
//-----Get Method-----
public void doGet(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException {
//-----get Data Session-----
res.setContentType("text/html;charset=windows-874");

```

รูปที่ ง-7 แสดง Source Code การ Update ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

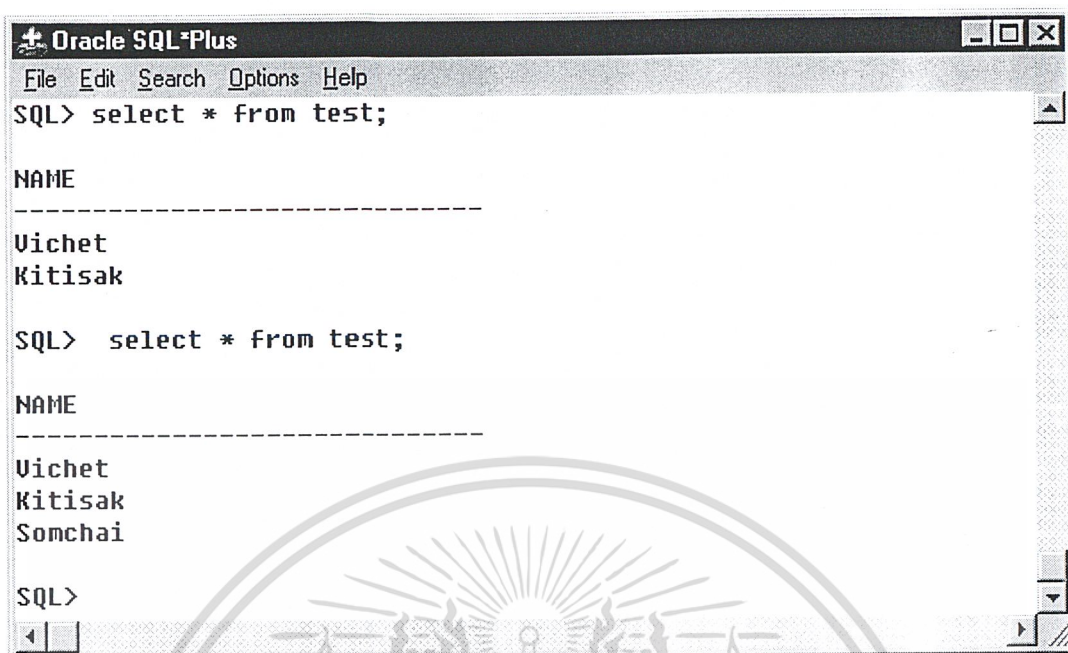
//-----Data Base-----
// Create a Statement
try{
    Statement stmt = conn.createStatement ();
    // insert row
    stmt.executeUpdate("INSERT INTO TEST VALUES ('Somchai')");
} catch ( SQLException e ) {
    System.err.println("A database error occurred.");
}
//-----Response User-----
PrintWriter out = res.getWriter();

    out.println("<HTML><HEAD><TITLE>TestUpdate</TITLE></HEAD>");
    out.println("<BODY>");
    out.println("Complete");
    out.println("</BODY></HTML>");
}
//-----Destroy-----
public void destroy() {
try {
// close the database connection
if ( conn != null) conn.close();
} catch (SQLException e) {
    System.err.println("Error closing database connection.");
}
} // destroy
}

```

รูปที่ ง-8 แสดง Source Code การ Updateฐานข้อมูล(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select * from test;

NAME
-----
Uichet
Kitisak

SQL> select * from test;

NAME
-----
Uichet
Kitisak
Somchai

SQL>
```

รูปที่ ๙-๑ แสดงผลการทำงาน การUpdateฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาคผนวก จ**  
**การติดตั้ง OAS (Oracle Application Server 4.0)**

**1. Hardware ที่ต้องการ**

Hardware Item	Required
CPU	An Intel compatible 486 or higher processor
Memory	128 MB
Disk Space	170 MB
Swap Space	256 MB

ตารางที่ จ-1 แสดง Hardware ที่ต้องการ

**2. Software ที่ต้องการ**

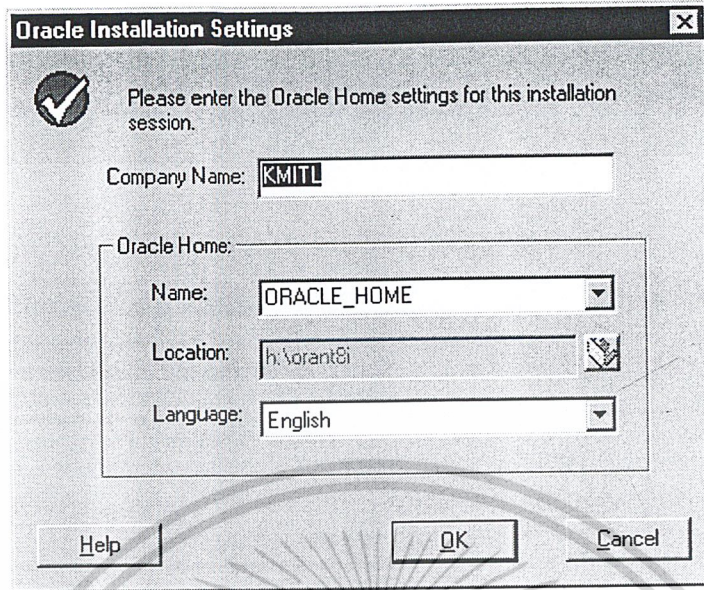
Software Item	Required
Operating System	NT Server 4.0 with version 1280 of Service Pack 5 (certified for deployment) NT Workstation 4.0 with Service Pack 5 (certified for development only)
Browser	Microsoft Internet Explorer: 5.0 Netscape Communicator: 4.51, 4.6

ตารางที่ จ-2 แสดง Software ที่ต้องการ

**3. การติดตั้ง OAS แต่ละ Step**

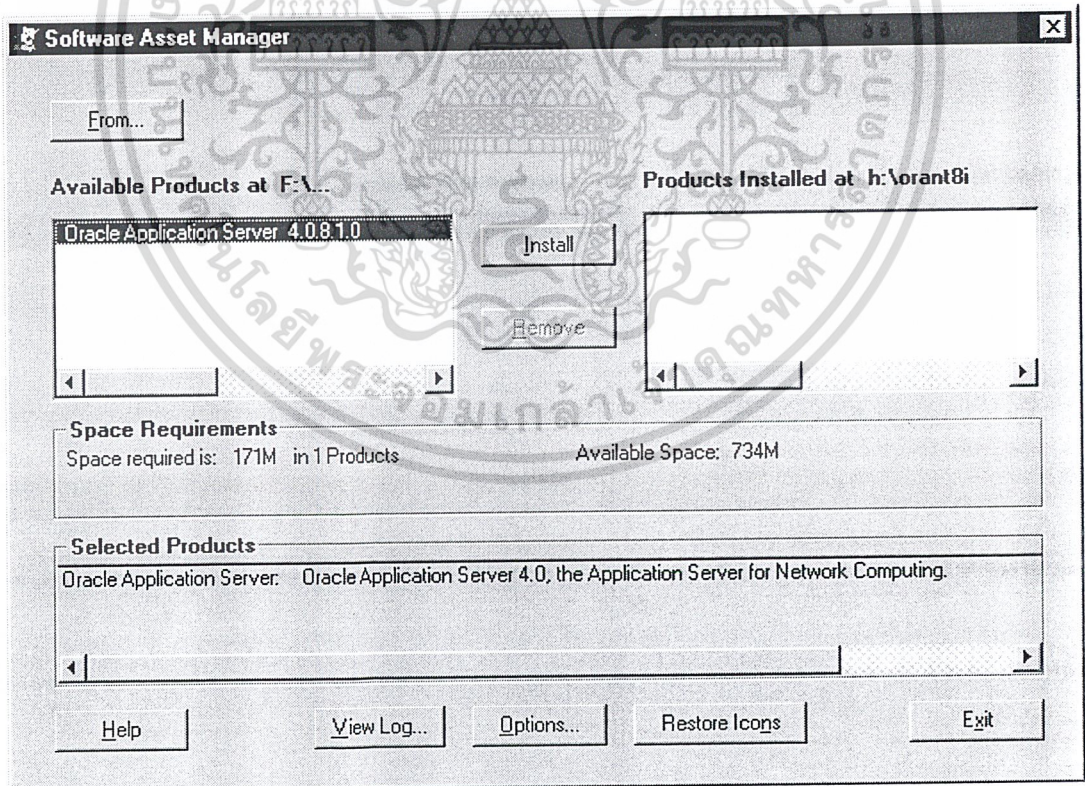
3.1 Run Setup จากแผ่น CD จากนั้นกำหนด Directory ,Name และ ภาษา ดังแสดงในรูปที่ จ-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-1 แสดงการกำหนด Directory

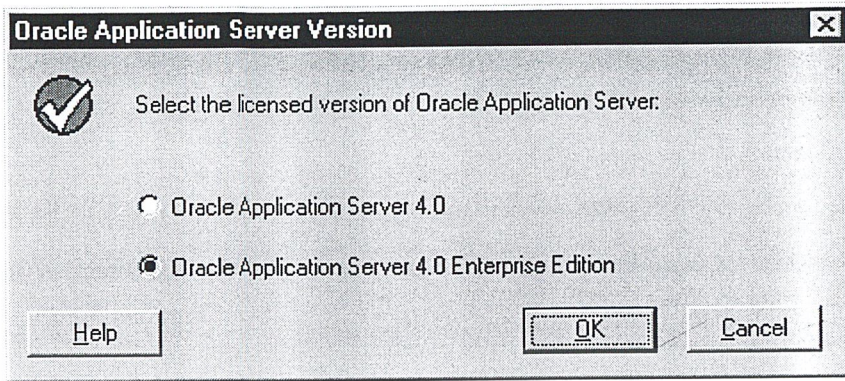
3.2 เลือก Product Oracle Application Server 4.0 แล้วกด Install



รูปที่ จ-2 แสดงการเลือก Product

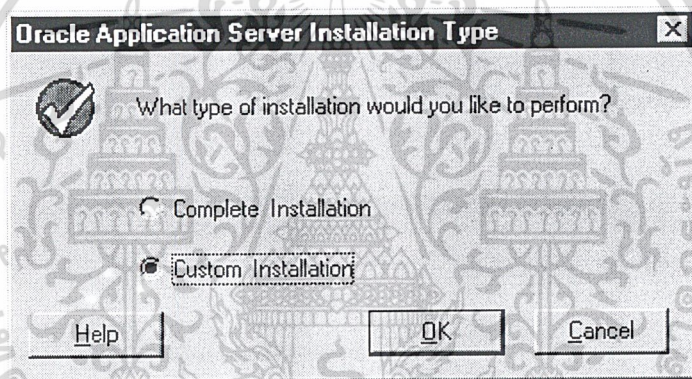
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเลือก Version โดยให้เลือก Enterprise Edition



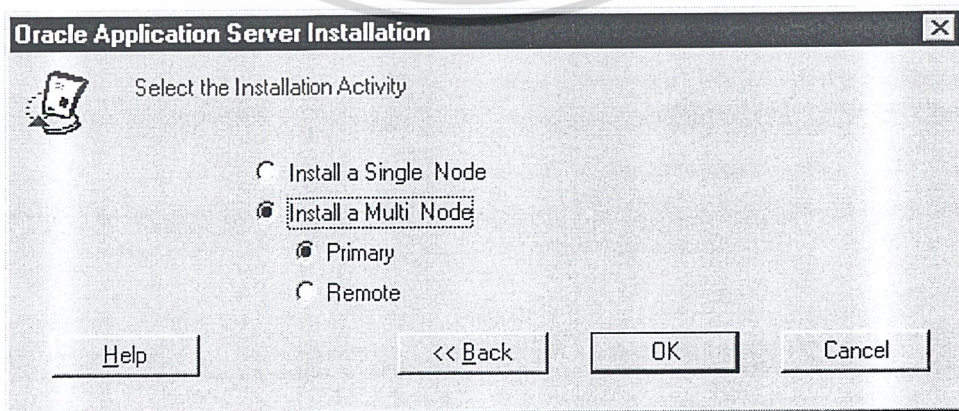
รูปที่ จ-3 แสดงการเลือก Version

3.4 การเลือก Installation Type โดยให้เลือกแบบ Custom



รูปที่ จ-4 แสดงการเลือก Installation Type

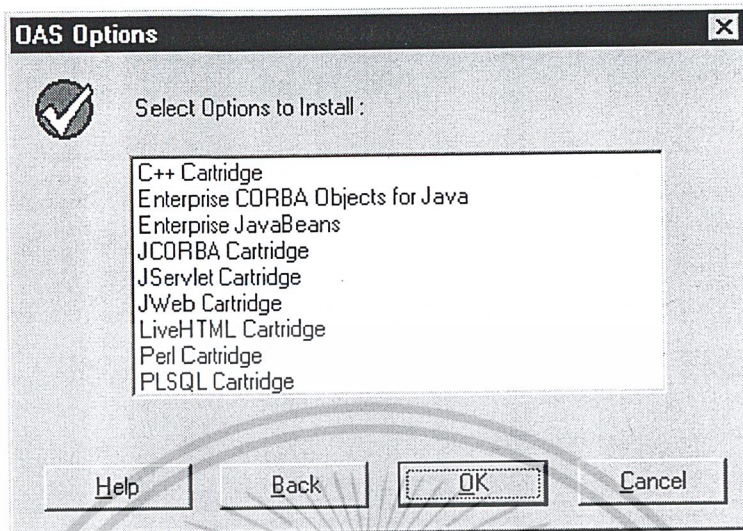
3.5 กำหนด Installation Activity ให้เลือก Multi Node แบบ Primary



รูปที่ จ-5 แสดงการกำหนด Installation Activity

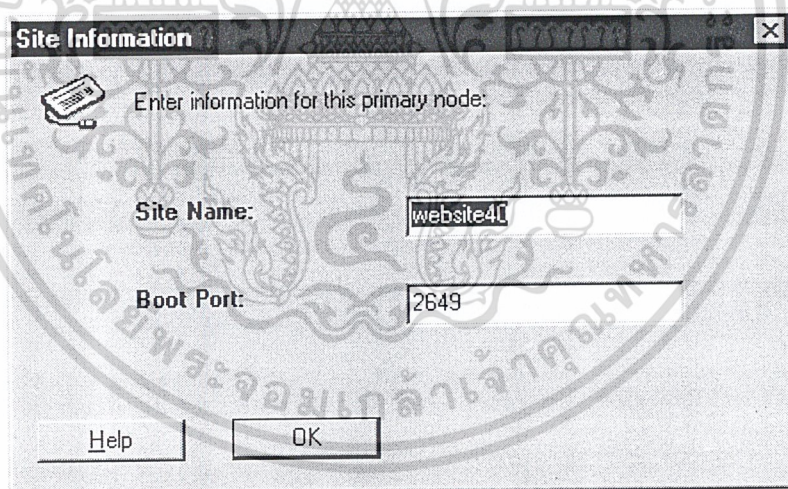
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเชิงวิชาการเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 การเลือก Option ที่ต้องการ



รูปที่ จ-6 แสดงการเลือก Option

### 3.7 การกำหนด Site Name และ Boot Port ให้กำหนดตามรูปที่ จ-7



รูปที่ จ-7 แสดงการกำหนด Site Name และ Boot Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.8 กำหนดข้อมูลเกี่ยวกับการเข้ามาจัดการ Server ของ Administrator

**Node Manager Listener Information** [X]

Node Manager Listener Settings :

Port Number:

User Name:

User Password:

Confirm Password:

Help OK

รูปที่ ๑-8 การกำหนด User Name และ Password

### 3.9 การกำหนด Multicast IP และ Port number ให้กำหนดตาม Default

**Group Messaging Configuration** [X]

Enter the Multicast IP address and Port number to be used by OAS for internal site wide group messaging:

If you are not sure what Multicast IP address and or Port to use, contact your System Administrator

Multicast IP:

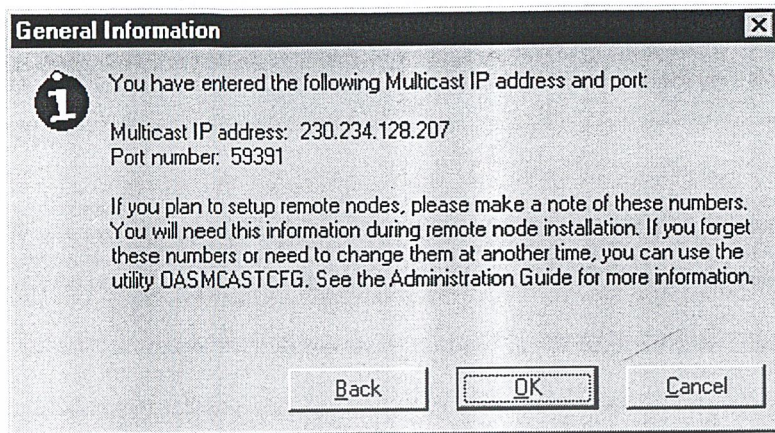
Port Number:

Help OK

รูปที่ ๑-9 การกำหนด Multicast IP และ Port number

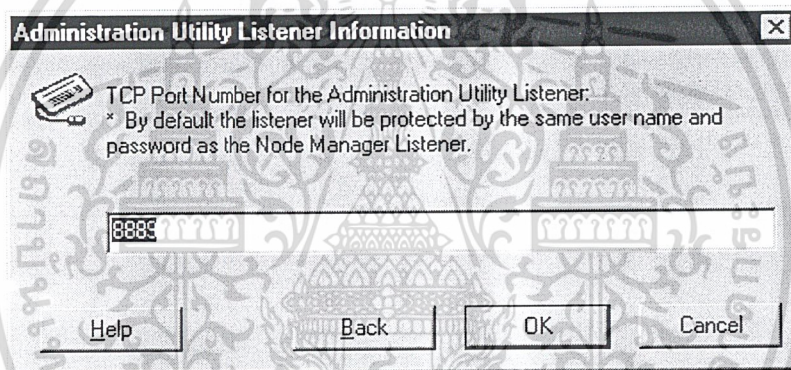
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.10 แสดง General Information ให้กด OK ได้เลย



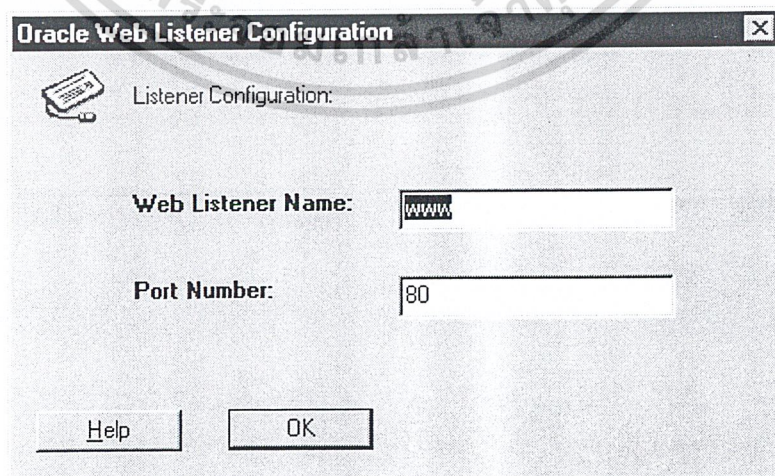
รูปที่ จ-10 แสดง General Information

### 3.11 กำหนด TCP Port Number ให้กำหนดตาม Default



รูปที่ จ-11 แสดงการกำหนด TCP Port Number

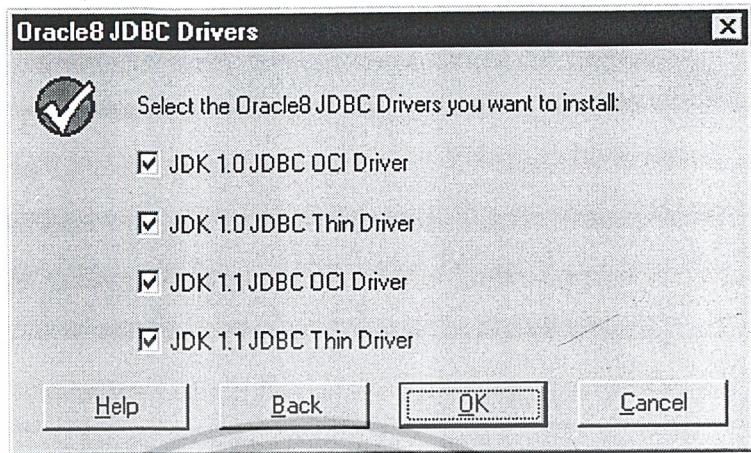
### 3.12 กำหนด Port Number เพื่อใช้ในการเรียก Web



รูปที่ จ-12 การกำหนด Port Number เพื่อใช้ในการเรียก Web

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.13 เลือก JDBC เพื่อใช้ในการติดต่อกับฐานข้อมูล



รูปที่ จ-13 เลือก JDBC เพื่อใช้ในการติดต่อกับฐานข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ฉ

### JavaScript

#### 1. หลักการเขียน JavaScript ใน HTML

จาวาสคริปต์เป็นภาษาซึ่งทำหน้าที่เพิ่มขยายขีดความสามารถและคุณสมบัติต่างๆที่ไม่มีใน HTML ดังนั้นลักษณะคำสั่งของจาวาสคริปต์จึงแตกต่างออกไปจากภาษาคอมพิวเตอร์อื่นๆเราสามารถเพิ่มคำสั่งจาวาสคริปต์เข้าไปในเว็บเพจได้ด้วยวิธีการดังต่อไปนี้

- เขียนคำสั่งหรือกำหนดฟังก์ชันไว้ภายในส่วนหัวของ Web Page
- เรียกใช้ฟังก์ชันจากสคริปต์อื่นๆ ที่เขียนต่อมา
- เรียกใช้ฟังก์ชันโดยค่า attribute ของคำสั่ง HTML

#### 2. การใช้คำสั่ง <SCRIPT>...</SCRIPT>

คำสั่ง <SCRIPT>...</SCRIPT> (เขียนเป็นตัวอักษรใหญ่หรือเล็ก หรือผสมกันได้) เป็นคำสั่งเพิ่มเติมในภาษา HTML ใช้กำหนดขอบเขตที่แน่นอนของคำสั่งภาษาสคริปต์ เพื่อให้เบราว์เซอร์สามารถจำแนกแยกแยะได้ว่าส่วนไหนเป็นชุดคำสั่งของภาษาสคริปต์ ส่วนไหนเป็นชุดคำสั่งของ HTML เพราะการทำงานของ Browser ในชุดคำสั่งทั้งสองแตกต่างกัน การเขียนจาวาสคริปต์มีรูปแบบดังนี้

```
<SCRIPT>
```

```
JavaScript Statements...
```

```
</SCRIPT>
```

คำสั่ง <SCRIPT> เป็นจุดเริ่มต้นที่ Browser จะทำงานตามชุดคำสั่งของ JavaScript ไปเรื่อยๆ จนกระทั่งพบคำว่า </SCRIPT> อันเป็นจุดสิ้นสุดการทำงานในส่วนของสคริปต์ หลังจากนั้นเว็บ Browser ก็จะทำหน้าที่อ่านชุดคำสั่งของ HTML ที่อยู่นอกสคริปต์ต่อไป ซึ่งภายในเว็บหนึ่งๆ จะใช้คำสั่ง <SCRIPT>...</SCRIPT> เท่าใดก็ได้ไม่จำกัดจำนวน

#### 3. ตำแหน่งของ SCRIPT

คำสั่ง <SCRIPT>...</SCRIPT> เป็นการกำหนดบริเวณพื้นที่และขอบเขตของการสั่งงานด้วยชุดคำสั่งของจาวาสคริปต์ซึ่งเราไม่สามารถเขียนชุดคำสั่งของจาวาสคริปต์นอกเหนือขอบเขตดังกล่าวได้ แต่ถ้าแปลนาคำสั่งจาวาสคริปต์ไปเขียนไว้ในนอกขอบเขตที่กำหนด HTML ไปเขียนอยู่ภายในขอบเขตของคำสั่ง <SCRIPT>...</SCRIPT> โปรแกรมจะแสดงข้อความขึ้นมา ยกเว้นกรณีที่น่าเอาชุดคำสั่ง HTML นั้นๆ ไปใช้ร่วมกับชุดคำสั่งตามแบบของจาวาสคริปต์

เราสามารถเขียนชุดคำสั่งจาวาสคริปต์ให้อยู่ในตำแหน่งที่ต้องการทั้งในส่วน <HEAD>...</HEAD> และส่วน <INDEX>...</INDEX> โดยที่โปรแกรมจะแสดงผลของ HTML และปฏิบัติตามคำสั่งจาวาสคริปต์ตามที่เรากำหนด

```

ตัวอย่างการเขียนชุดคำสั่งของจาวาสคริปต์ภายในส่วน <HEAD>...</HEAD>

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
        function pushStack(theForm) {
            theForm.stack.value = theForm.display.value
            theForm.display.value = 0
        }
    </SCRIPT>
</HEAD>
<BODY>
.....
</BODY>
</HTML>

```

หรือ

```

<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
    function pushStack(theForm) {
        theForm.stack.value = theForm.display.value
        theForm.display.value = 0
    }
</SCRIPT>
.....
</BODY>
</HTML>

```

ตัวอย่างการเขียนสคริปต์ทั้งสองชุดให้ผลลัพธ์ออกมาเหมือนกัน

การตัดสินใจเขียนจาวาสคริปต์ไว้ในส่วนใดนั้นขึ้นอยู่กับความพึงพอใจของผู้เขียนเป็นหลัก แต่ถ้าประเมินความนิยมแล้วพบว่านิยมเขียนไว้ในส่วน <HEAD>...</HEAD> มากกว่าเพราะมีข้อดีคือ ชุดคำสั่งและฟังก์ชันต่างที่เขียนนั้น จะถูกโหลดไว้ในหน่วยความจำก่อนการเกิด HTML ส่วนอื่นๆ เมื่อเกิดเหตุการณ์ (Event) ต่างๆตามมาโดยการกำหนด “อีเวนต์แฮนด์เลอร์” (Event Handler) ไว้เช่น onLoad ในส่วน <BODY> ( หมายถึงการกระตุ้นให้สคริปต์ทำงานเมื่อเว็บเพจถูกโหลด ) หากเราเขียนสคริปต์ไว้ในส่วน <BODY>...</BODY> จะไม่สามารถใช้อีเวนต์แฮนด์เลอร์นี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. Document Object

มาถึงตอนนี้ จะขอเริ่มแนะนำ Object อันดับแรก ซึ่งน่าจะสามารใช้งานได้บ่อย ก็คือ Document Object นั้นเอง Document มี properties ต่างๆ ที่น่าสนใจดังนี้

- bgColor : กำหนดค่าของสีของฉากหลัง เป็น RGB Color
- fgColor : กำหนดค่าของสีตัวอักษร เป็น RGB Color เช่นกัน

forms : property นี้จะกล่าวถึงอีกครั้งเมื่อพูดถึง form object

- lastModified : ค่าแสดงเวลาการแก้ไขครั้งสุดท้ายของเอกสารนั้น

และมี method

- write() : เป็นคำสั่งให้มีการเขียนข้อความ
- writeln() : เป็นคำสั่งให้มีการเขียนข้อความโดยขึ้นบรรทัดใหม่

สิ่งเหล่านี้ไม่ใช่ properties หรือ method ทั้งหมด ของ object นี้ เนื่องจาก version ของ JavaScript เอง รวมถึง การอ้างอิงถึง JavaScript จากหลายๆค่าย ทั้งนี้ ผมจะอ้างอิง JavaScript version 2 ของค่าย Sun เป็นหลัก ทั้งนี้เนื่องจากว่า เป็น JavaScript ที่เห็นว่า สามารถ support จาก browser เกือบทั้งหมดที่ support JavaScript ไม่ว่าจะเป็น IE(Internet Explorer 3 ขึ้นไปของ Microsoft) หรือ Netscape และนี่เป็น ตัวอย่างการใช้ประโยชน์จาก Object Document นี้

```
<Script Language="JavaScript">
//<!--
document.write("นี่เป็นข้อความที่เขียนด้วย JavaScript <br>")
document.write("และเอกสารนี้แก้ไขครั้งสุดท้ายเมื่อ " + document.lastModified)
//-->
</Script>
```

#### 5. Window Object

Window มี properties ต่างๆ ที่น่าสนใจดังนี้

- status และ defaultStatus : ข้อความที่ระบุที่ช่องแสดงสถานะด้านล่างของ Window ( ซึ่งในที่นี้คือ browser ) และ ข้อความปกติที่ window แสดง
- name : ชื่อของ window นั้น เพื่อการอ้างอิงภายหลังและมี method
- alert( 'ข้อความที่ต้องการแสดง' ) : เรียกหน้าต่างแสดงข้อความ
- open( URL , ชื่อของ window , ข้อกำหนดย่อย ) : สั่งให้เปิด window ใหม่ โดยมีข้อกำหนดย่อยต่างๆ ดังนี้ directories, location, menubar, resizable, scrollbars, status, toolbar height=pixels, width=pixels ซึ่งจะอธิบายในตัวอย่างอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- `setTimeout` (เวลาเป็นมิลิวินาที) : กำหนดเวลาการ refresh หน้าจอ และนี่เป็น ตัวอย่างการใช้ประโยชน์จาก `window Document` นี้

```
<a href=# onClick="window.open('testwindow.htm','test_window',
'height=50,width=50,status,scrollbars')">TEST-WINDOW</a>
```

## 6. Function และ ตัวแปร

ก่อนจะอธิบายถึง Object อื่นๆ ต่อไป ขออธิบายวิธีการกำหนดตัวแปร และวิธีการกำหนด Function ก่อน เนื่องจากการใช้ JavaScript ส่วนใหญ่หลังจากนี้ใช้ผ่าน Function เป็นส่วนใหญ่

```
< Script Language='JavaScript'>
```

```
//<!--
```

```
var ชื่อตัวแปร
```

```
function ชื่อFunction(ตัวแปรที่ต้องการใช้)
```

```
{
```

```
ชุดการทำงานที่กำหนด
```

```
}
```

```
//-->
```

```
</Script>
```

จากตัวอย่างครั้งที่แล้ว คุณสามารถเขียนในลักษณะนี้ได้เช่นเดียวกัน

```
<Script Language='JavaScript'>
```

```
//<!--
```

```
function openwindow( windowfile )
```

```
{
```

```
window.open( windowfile , 'test_window' , 'height=50,width=50,status,scrollbars' )
```

```
}
```

```
//-->
```

```
</Script>
```

```
<a href=# onClick="openwindow( 'testwindow.htm' )">TEST-WINDOW</a>
```

## 7. Text Object

Text Object มี properties ที่สำคัญดังนี้

- `name` : ชื่อซึ่งใช้อ้างอิง Text Object นั้น
- `Value` : ค่า ใน Text Object นั้น
- `defaultValue` : ค่าที่กำหนดให้ Text Object สว่างหน้า Method

- `focus()` : กำหนด ให้ cursor ไปยัง Text Object นั้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- select() : highlight text ใน Text Object

จากนี้เราจะลองกำหนด เวลา ลงใน Text Object และ highlight text เมื่อ click mouse บน Text object นั้น

```
<Script Language='JavaScript'>
//<!--
function showtime()
{
document.form1.text1.value = Date()
setTimeout('showtime()',6000)
}
//-->
</Script>
```

<body onLoad="showtime()">

และ

```
<form name="form1" method="post">
<input type="text" name="text1" onFocus="document.form1.text1.select()">
</form>
```

Function Date() เป็น Function ของ JavaScript ที่จะกล่าวถึงต่อไป

## 8. Built-in Function

ตัวอย่างการใช้ Function

ก่อนอื่นให้สร้าง form ดังนี้

```
<form method="POST" name="f_test">
<p align="center"><input type="text" name="T_input" size="10"><br>
<input type="button" value="Click Test" name="B_submit" onclick="f_teststr()"></p>
</form>
```

โดยเมื่อ Click "Click Test" จะไปเรียก Function f\_teststr() ตามตัวอย่างด้านล่าง

```
<Script Language='JavaScript'>
//<!--
function f_teststr()
```

```
{
```

```
var v_temp1, v_temp2, v_temp3
```

```
v_temp1 = document.f_test.T_input.value
```

```
v_temp2 = parseFloat( document.f_test.T_input.value )
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (v_temp1 == "")
{
window.alert("Please input Value")
return
}
if (v_temp1 == v_temp2)
{
window.alert("You have input Number")
}
else
{
window.alert("You have input String")
}
}
//-->
</Script>
```

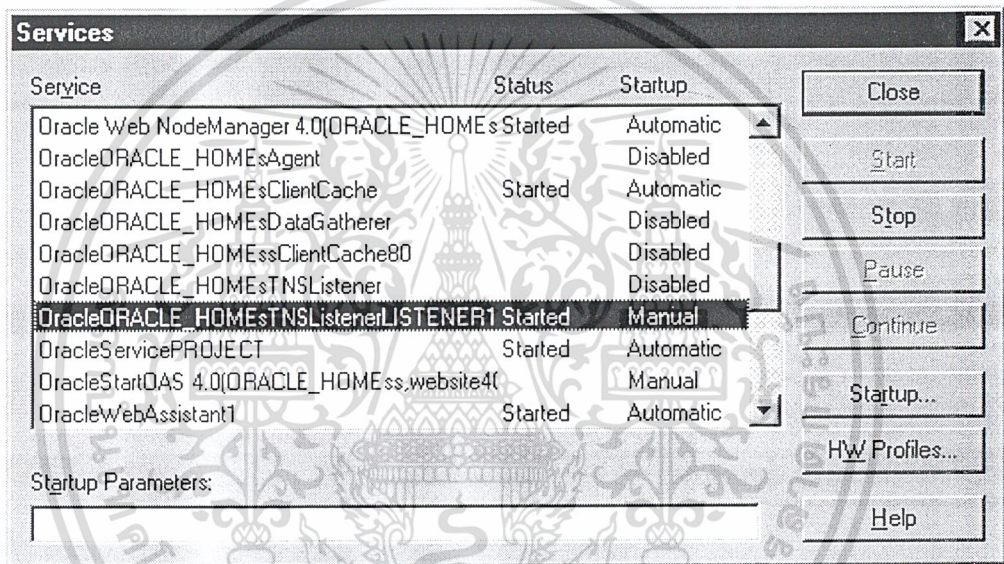


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข

### ปัญหาต่างๆ และ คำแนะนำ

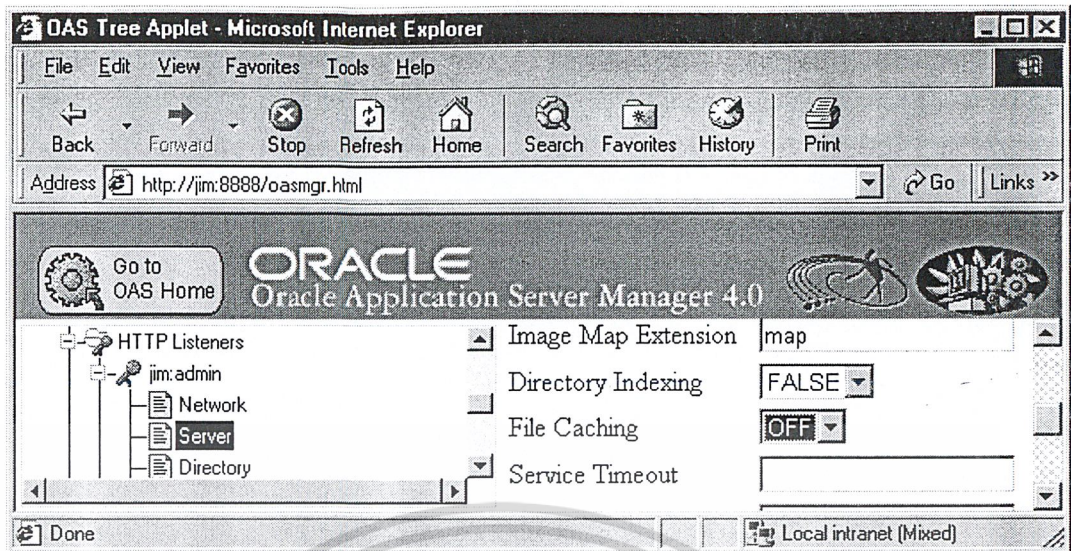
1. การติดตั้ง WindowNT 4.0 จะต้องมีเครื่องที่เป็น Primary เพียงหนึ่งเครื่อง เท่านั้นภายใน Domain เดียวกัน ไม่เช่นนั้นจะมีปัญหาการชนกันและทำให้เกิดปัญหาอื่น ๆ เกิดขึ้น
2. การติดตั้ง Oracle8i และ OAS(Oracle Application Server) ในเครื่องเดียวกัน โดยปกติแล้วไม่ควรทำแต่ถ้ามีความจำเป็นต้องติดตั้งในเครื่องเดียวกัน จะต้องกำหนดการ Startup ของ Oracle TNSListener ของ Oracle8i DBMS ใน Service ของ WindowNT 4.0 ให้เป็น Manual จากนั้นให้มาทำการ Start เองหลังจาก Boot เครื่อง เสร็จแล้วดังรูปที่ ข-1



รูปที่ ข-1 การ Start TNSListener ของ Oracle8i DBMS ในภายหลัง

3. ปัญหาเกี่ยวกับภาษาไทย
  - 3.1 การแสดงภาษาไทยออกที่หน้า Webpage ไม่ได้ในส่วนของ JServlet ให้ใช้ Code ดังด้านล่างนี้
 

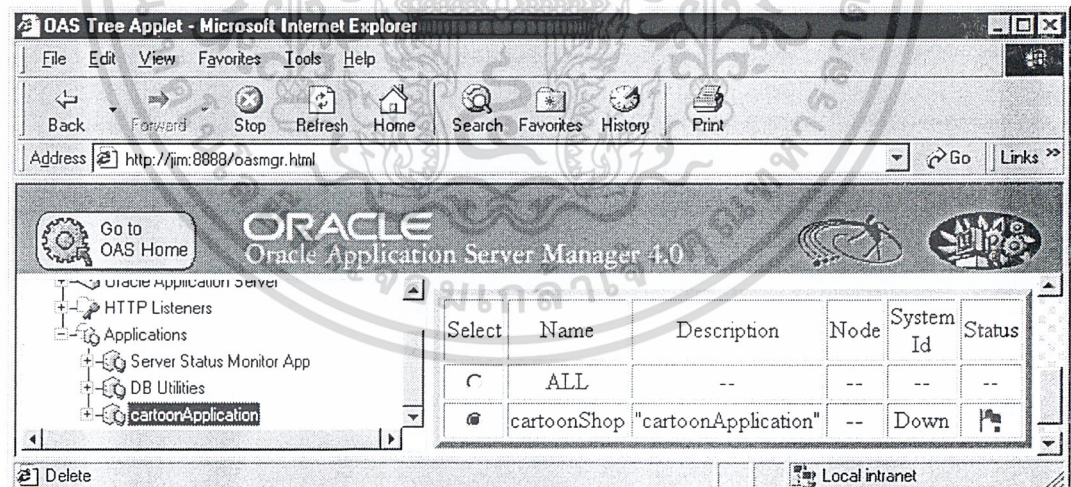
```
res.setContentType("text/html;charset=windows-874");
```
  - 3.2 การทำ Submit Form แล้วมีการนำข้อมูลจากการ Submit มาใช้ในการติดต่อกับฐานข้อมูล จะต้องนำข้อมูลที่ได้มาแปลงเป็น Unicode เสียก่อน
4. เมื่อแก้ไข file.html หรือ file.class แล้วต้อง Stop WWW ใน OAS ก่อนจึงจะ Save ได้มีวิธีการแก้ไขโดยการตั้งค่าของ File Caching ให้เป็น OFF ดังรูปที่ ข-2



รูปที่ ข-2 แสดงการตั้งค่า File Caching

5. เมื่อ Update ข้อมูล ใน SQL Plus จะต้องมี การ Commit; ไม่เช่นนั้นเมื่อมี Application เข้ามาใช้ข้อมูลจะยังเป็นของเดิมอยู่

6. เมื่อแก้ไข File.class ที่เป็น JServlet จะต้อง Stop Application นั้นก่อนไม่เช่นนั้นข้อมูลที่แสดงออกมาจะยังเป็นของเดิมอยู่



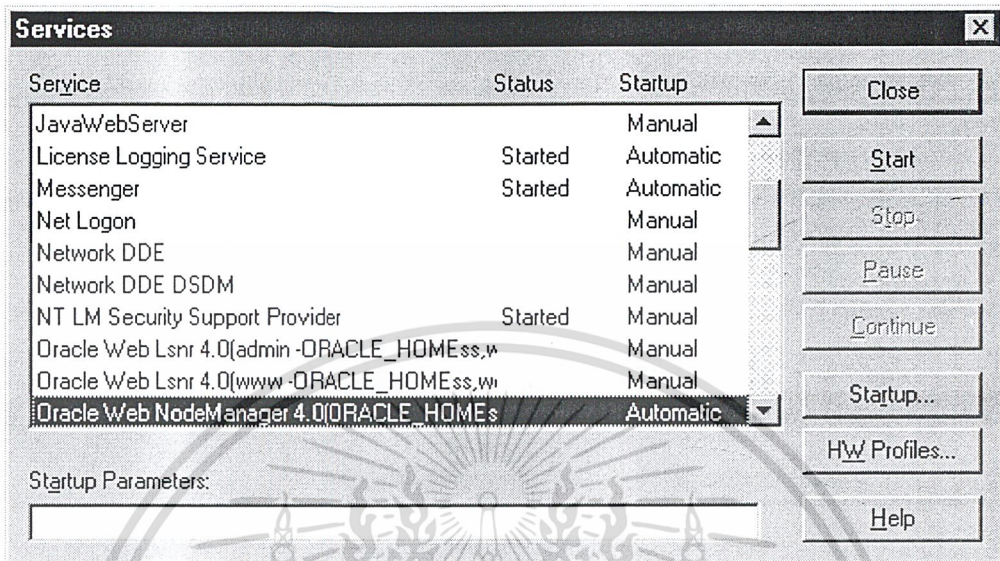
รูปที่ ข-3 แสดงการ Stop Application

7. การเรียกดู Web จากเครื่องที่มีชื่อเครื่องไม่เหมือนกับชื่อ Web ให้หา File ชื่อ Hosts แล้ว Edit แก้ไขดังตัวอย่างด้านล่างนี้

127.0.0.1 localhost

เอกสารนี้เป็นเอกสารที่ 161.246.5.163 myproject เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. การ UnInstall OAS จะต้องทำการ Stop Services Oracle Web ให้หมดเสียก่อนจึงจะทำการ UnInstall ได้ ดังรูปที่ ข-4



รูปที่ ข-4 แสดงการ Stop Oracle Web

9. การนำข้อมูลจากฐานข้อมูล Oracle8i หรือ จาก Submit Form มาใช้จะต้องทำการตัดช่องว่างที่อยู่ก่อนหน้า และ ต่อท้ายออกก่อนเพื่อป้องกันความผิดพลาด

### บรรณานุกรม

- [1] Jim Conallen : “*Building Web Applications with UML : Library of Congress Cataloging-in-Publication Data*”,1999.
- [2] Grady Booch : “*Object-Oriented Analysis and Design With Applications : Addison-Wealey Publishing Co.*,”,1994.
- [3] อภิเนตร อุณาภุท : “*Object Oriented Analysis and Design : แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง* “,2000.
- [4] Jason Hunter with William Crawford : “*JAVA Servlet Programming : O'Reilly & Associates,Inc., 101 MorrisStreet,Sebastopol,CA 95472*”,1998.
- [5] Karl Moss : “*Java Servlets : McGraw-Hill*”,1999.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้