สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบฐานข้อมูลเชิงวัตถุและการพัฒนาโปรแกรมประยุกต์สำหรับเอกสาร HTML

Object Oriented Database Design and Develop Application Program for HTML Document

โดย

นาย ธีระชัย อดิเรกพูลลาภ

นาย ธีระเดช พิสุทธิสกุลชัย

เลขหมู่.....
เลขทะเบียน 46406
วัน, เดือน, ปี 1 เม.ย. 2546

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การออกแบบฐานข้อมูลเชิงวัตถุและการพัฒนาโปรแกรมประยุกต์สำหรับเอกสาร HTML
Title	Object Oriented Database Design and Develop Application Program for HTML Document
โดย	นาย ธีระชัย อดิเรกพูลลาภ รหัสประจำตัว 41014197
	นาย ธีระเดช พิสุทธิสกุลชัย รหัสประจำตัว 41014198
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์ สุธีรา พันธุ์ธีรานุรักษ์
ปีการศึกษา	2544

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

.....

(อาจารย์สุธีรา พันธุ์ธีรานุรักษ์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การออกแบบฐานข้อมูลเชิงวัตถุและการพัฒนาโปรแกรมประยุกต์สำหรับเอกสาร HTML
นักศึกษา	นาย ธีระชัย อดิเรกพูลลาภ 41014197
	นาย ธีระเดช พิสุทธิสกุลชัย 41014198
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์ สุธีรา พันธุ์ธีรานุรักษ์
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

โครงงานการออกแบบฐานข้อมูลเชิงวัตถุและการพัฒนาโปรแกรมประยุกต์สำหรับเอกสาร HTML เป็นการประยุกต์โดยทำการแปลงข้อมูลเอกสาร HTML ให้อยู่ในรูปของข้อมูลเชิงวัตถุเพื่อให้ง่ายต่อการวิเคราะห์และการจัดการต่างๆเกี่ยวข้อมูล

วัตถุประสงค์ของโครงงานนี้ เพื่อศึกษาถึงลักษณะของฐานข้อมูลเชิงวัตถุที่มีความสามารถในการจัดการข้อมูลในรูปแบบพิเศษที่ฐานข้อมูลเชิงความสัมพันธ์ไม่สามารถจัดการได้ โดยได้ทำการพัฒนาโปรแกรมที่มีความสามารถในการแปลงข้อมูลเอกสาร HTML ให้อยู่ในรูปของข้อมูลเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT TITLE	**OBJECT ORIENTED DATABASE DESIGN AND DEVELOP APPLICATION PROGRAM FOR HTML DOCUMENT**		
STUDENT	**Mr.Teerachai**	**Adirekpullap**	**41014197**
	Mr.Teeradate	**Pisutthisakunchai**	**41014198**
ADVISOR	**Ms.Sutheera**	**Puntheeranurak**	
COURSE	**Bachelor of Information Engineering**		
DEPARTMENT	**Information Engineering**		
YEAR	**2001**		

ABSTRACT

This project is an application that used object oriented analysis and design for developing software application to manage HTML files to store , delete , update and retrieve by using object oriented database instead of relational database. The main objective of this project is to understand characteristics , advantages and disadvantages of object oriented database compared to relational database by creating the application. It is more flexible but still hard to understand. It has demonstrated that there are advantages more than disadvantages compared to relation systems.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ขอขอบพระคุณอาจารย์สุธีราที่เป็นอาจารย์ที่ปรึกษา และอาจารย์ภุชงค์ที่ให้คำแนะนำและความช่วยเหลือที่ดีเสมอมาตลอด อีกทั้งยังได้ชี้แนะแนวทางในการแก้ไขปัญหาต่างๆจนสามารถสำเร็จผ่านมาได้ด้วยดี

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยห่วงใยและให้การสนับสนุนในการศึกษา รวมทั้งขอขอบคุณญาติสนิททุกคนที่เป็นกำลังใจพร้อมทั้งให้ความช่วยเหลือในด้านต่างๆมาโดยตลอด

สุดท้ายนี้ขอขอบคุณเพื่อน ห้อง 4F รุ่นที่หนึ่งของวิศวกรรมสารสนเทศทุกๆคนและเพื่อนต่างห้องต่างภาควิชาคนอื่นๆที่ช่วยเหลือ ให้คำปรึกษาและเป็นกำลังใจที่ดีให้แก่กันมาโดยตลอด

นาย ธีระชัย อดิเรกพูลลาภ

นาย ธีระเดช พิสุทธิสกุลชัย

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

เนื่องจากในปัจจุบันการจัดเก็บข้อมูลเอกสาร HTML จะทำการจัดเก็บไว้บนเซิร์ฟเวอร์ (Server) ซึ่งจะใช้รูปแบบในการจัดเก็บเป็นไฟล์ข้อมูลต่างๆ และทำการเชื่อมโยงโดยอาศัยลักษณะความสัมพันธ์ภายในข้อมูลที่มีอยู่ภายในไฟล์ต่างๆ และยังต้องทำการเก็บข้อมูลของการเชื่อมต่อหรือเรียกอีกอย่างหนึ่งว่า ลิงค์ (Link) ของแต่ละไฟล์อีกด้วย ทำให้เกิดปัญหาต่างๆ ทั้งในเรื่องของประสิทธิภาพในการจัดเก็บข้อมูล การแก้ไขข้อมูล การลบข้อมูล และการจัดการต่างๆเกี่ยวกับข้อมูลกระทำได้ยาก จึงได้มีการนำเอาแนวความคิดในการจัดการฐานข้อมูลเชิงวัตถุเข้ามาประยุกต์ใช้ โดยลักษณะของการจัดการฐานข้อมูลเชิงวัตถุนั้นจะทำการแบ่งแยกส่วนประกอบต่างๆ (Component) ให้เป็น วัตถุ (Object) เพื่อให้ง่ายในการจัดเก็บลงไปใน ฐานข้อมูล (Database) ซึ่งจะทำการจัดเก็บผ่านทางระบบการจัดการฐานข้อมูล (DBMS) และง่ายต่อการ วิเคราะห์และการจัดการต่างๆเกี่ยวข้อมูลได้ง่ายยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาถึงลักษณะของฐานข้อมูลเชิงวัตถุ (Object Database) ที่มีความสามารถในการจัดการกับข้อมูลมัลติมีเดีย (Multimedia) ซึ่งฐานข้อมูลเชิงความสัมพันธ์ (Relational Database) ไม่สามารถกระทำได้โดยจะนำมาประยุกต์ใช้สำหรับการจัดเก็บเอกสาร HTML เข้าไปใน ระบบการจัดการฐานข้อมูล

1.2.2 เพื่อศึกษาการออกแบบและการพัฒนาระบบ โดยการใช้วิธีการวิเคราะห์และออกแบบข้อมูลเชิงวัตถุ (Object Oriented Analysis and Design)

1.2.3 เพื่อทำการพัฒนาโปรแกรมที่จะทำการแปลงรูปแบบของเอกสาร HTML ให้อยู่ในรูปแบบของข้อมูลเชิงวัตถุ เพื่อให้สามารถทำการจัดเก็บเอกสารไว้ในฐานข้อมูลเชิงวัตถุ แทนที่จะการจัดเก็บไว้บน เวปเซิร์ฟเวอร์ (Web Server) ทั่วๆไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถทำการออกแบบฐานข้อมูลเชิงวัตถุให้ครอบคลุมการใช้งานของเอกสาร HTML ทั้งหมด โดยใช้หลักการวิเคราะห์และออกแบบโดยการใช้ UML เข้ามาประยุกต์

1.3.2 สามารถเขียนโปรแกรมเพื่อทำการแปลงเอกสาร HTML ในรูปทั่วๆไป ให้อยู่ในรูปของข้อมูลเชิงวัตถุ

1.3.3 สามารถนำโปรแกรมที่สร้างขึ้นมาทำการเชื่อมต่อกับฐานข้อมูลเชิงวัตถุเพื่อทำการเก็บข้อมูลเชิงวัตถุนั้น

1.3.4 สามารถเขียนโปรแกรมเพื่อทำการแปลงข้อมูลเชิงวัตถุที่อยู่ในฐานข้อมูลเชิงวัตถุกลับไปเป็นเอกสาร HTML ได้

1.4 ผลที่คาดว่าจะได้รับจากโครงงาน

1.4.1 สามารถทำการวิเคราะห์และออกแบบระบบโดยใช้หลักการเชิงวัตถุได้

1.4.2 สามารถใช้แบบจำลอง UML เพื่อใช้ในการพัฒนาระบบและสื่อสารกับบุคคลที่เกี่ยวข้องกับระบบนั้นได้

1.4.3 สามารถนำโปรแกรมที่สร้างขึ้นไปใช้งานได้จริง โดยสามารถนำไปใช้เพื่อทำการจัดเก็บข้อมูลเอกสาร HTML ให้อยู่ในรูปของข้อมูลเชิงวัตถุที่สามารถทำการวิเคราะห์และจัดการได้ง่ายยิ่งขึ้น

1.5 ส่วนประกอบของโครงงาน

จะเป็นซอฟต์แวร์ (Software) ที่มีส่วนประกอบหลัก 2 ส่วนคือ

1.5.1 ซอฟต์แวร์ที่เป็นระบบจัดการฐานข้อมูลเพื่อใช้ในการติดต่อระหว่างผู้ใช้งานกับฐานข้อมูลเชิงวัตถุ

1.5.2 ซอฟต์แวร์ที่พัฒนาขึ้นเพื่อใช้ในการแปลงเอกสาร HTML ให้มีลักษณะเป็นข้อมูลเชิงวัตถุ และทำการแปลงข้อมูลเชิงวัตถุที่มีอยู่ในฐานข้อมูลเชิงวัตถุกลับไปเป็นเอกสาร HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่ใช้ในโครงงาน

2.1 หลักการของ HTML ที่ใช้ในโครงงาน

จะทำการแยกส่วนประกอบ (Component) ต่างๆที่อยู่ภายในเอกสาร HTML ให้แยกออก โดยมีลักษณะที่อยู่ในรูปเชิงวัตถุ โดยจะทำการวิเคราะห์จากอิลลิเมนต์ (Element) ของเอกสาร HTML โดยภายในแต่ละ อิลลิเมนต์จะมีความหมายที่ชัดเจน โดยรูปแบบของอิลลิเมนต์จะมีการกำหนดรูปแบบมาตราฐานเอาไว้ตามหลักขององค์กร W3C

2.2 ลักษณะโครงสร้างของเอกสาร HTML

ในที่นี้จะยึดตามหลักของเอกสาร HTML 4.0 ที่ใช้งานกันอยู่ทั่วๆไป จะประกอบไปด้วย

2.2.1 อิลลิเมนต์และแทค (Element and Tag)

อิลลิเมนต์จะเป็นส่วนที่ใช้อธิบายส่วนที่อยู่ภายในเอกสาร HTML ยกตัวอย่างเช่น อิลลิเมนต์ *P* เป็นส่วนที่ใช้แสดงพารากราฟ (Paragraph) โดย อิลลิเมนต์จะประกอบไปด้วยส่วนประกอบ 3 ส่วนคือแทคเริ่มต้น (Start Tag) , ส่วนประกอบภายใน (Content) , แทคปิดท้าย (End Tag)

โดยแทค (Tag) คือ ข้อความที่ถูกขั้นด้วย “<” และ “>” และแทคปิดท้ายจะมี “/” ตามหลัง “<” ยกตัวอย่างเช่น อิลลิเมนต์ *EM* จะมีแทคเริ่มต้นเป็น <EM> และมีแทคปิดท้ายเป็น </EM> และส่วนที่อยู่ระหว่างแทคเริ่มต้นกับแทคปิดท้ายจะเป็นส่วนที่เรียกว่าส่วนประกอบภายใน

อิลลิเมนต์จะไม่มีการเหลื่อมทับกัน ยกตัวอย่างเช่น ถ้าแทคเริ่มต้นของ อิลลิเมนต์ *EM* อยู่ภายในอิลลิเมนต์ *P* ส่วนที่เป็นแทคปิดท้ายของ อิลลิเมนต์ *EM* ก็ต้องอยู่ภายในอิลลิเมนต์ *P* เช่นกัน

ชื่อของอิลลิเมนต์จะมีความหมายเดียวกัน โดยไม่ขึ้นอยู่กับตัวอักษร ยกตัวอย่างเช่น <title> , <Title> , <TITLE> และบาง อิลลิเมนต์วนที่เป็นแทคปิดท้าย อาจจะได้รับการยกเว้น ยกตัวอย่างเช่น อิลลิเมนต์ *LI* หรือบาง อิลลิเมนต์ไม่มีส่วนที่เป็นแทคปิดท้าย เพราะว่าไม่มีส่วนประกอบภายในยกตัวอย่างเช่นอิลลิเมนต์ *BR*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 แอททริบิวต์ (Attribute)

แอททริบิวต์จะเป็นคุณสมบัติต่างๆที่อยู่ภายในอิลลิเมนต์นั้นๆ ยกตัวอย่าง เช่น อิลลิเมนต์ IMG จะมีแอททริบิวต์ SRC ที่ใช้ในการบอกตำแหน่งของภาพ (Image) โดยที่ แอททริบิวต์จะรวมอยู่ภายในแทคเริ่มต้นและแทคปิดท้าย โดยจะอยู่ในรูปของ Attribute-Name = "Attribute-Value"

2.2.3 อักขระพิเศษ (Special Characters)

ในการสร้างเอกสาร HTML รูปแบบของอักขระบางตัวจะไม่อยู่ในรูปแบบทั่วๆไปที่สามารถใช้งานได้ แต่จะอยู่ในรูปของอักขระที่เป็น รหัสแอสกี (ASCII) เช่น "<" จะถูกเขียนแทนด้วยเอนทิตี้ (Entity) < หรือ "&"จะถูกเขียนแทนด้วย & หรือใช้ตัวเลขในการแทนค่าตัวอักขระ เช่น ตัวเลข 1 ใช้ &#169 เขียนแทนเป็นต้น

2.2.4 คอนเม็นต์ (Comment)

คอนเม็นต์เป็นส่วนที่ใช้สำหรับเขียนความคิดเห็นเพิ่มเติม เพื่อบอกความหมายของการใช้งานภายในเอกสาร HTML โดยมีหลักการใช้งาน คือ จะเริ่มต้นด้วย "<! - - " ลงท้ายด้วย "- - >" และต้องไม่มีการใช้ " - - " ภายในข้อความคอนเม็นต์นั้นๆ

2.2.5 ลักษณะของเอกสาร HTML ที่สมบูรณ์ (A Complete HTML Document)

จะเริ่มต้นด้วย DOCTYPE เพื่อเป็นการประกาศเวอร์ชั่น (Version) ของเอกสาร HTML ที่สร้างขึ้น จากนั้นตามด้วย อิลลิเมนต์ HTML ซึ่งภายในจะบรรจุ อิลลิเมนต์ HEAD และ BODY ซึ่ง HEAD จะบรรจุด้วยข้อมูลเกี่ยวกับเอกสาร HTML จำพวก ชื่อหัวเรื่องขณะที่ BODY จะบรรจุข้อมูลที่ต้องการใช้งานในเอกสาร HTML นั้นๆ ซึ่งตัวอย่างด้านล่างจะแสดงรูปแบบพื้นฐานของเอกสาร HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
        "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
        <HEAD>
                <TITLE>The document title</TITLE>
        </HEAD>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<BODY>
    <H1>Main heading</H1>
    <P>A paragraph.</P>
    <P>Another paragraph.</P>
    <UL>
        <LI>A list item.</LI>
        <LI>Another list item.</LI>
    </UL>
</BODY>
</HTML>
```

2.3 หน้าที่ของอิลลิเมนต์ต่างๆที่มีอยู่ในเอกสาร HTML

อิลลิเมนต์ที่มีอยู่ในเอกสาร HTML ถ้านำมาจำเรียงตามลักษณะของหน้าที่การใช้งานจะสามารถแยกออกเป็น 10 ส่วน ซึ่งประกอบไปด้วย

2.3.1 Top-Level Elements (HTML , HEAD , BODY , FRAMESET)

- HTML ใช้ในการกำหนดให้เบาว์เซอร์ (Browser) รู้ว่ารูปแบบของเอกสารอยู่ในรูปของ HTML
- HEAD ใช้ในการบรรจุข้อมูลเนื้อหาที่เกี่ยวกับเอกสาร HTML
- BODY ใช้ในการกำหนดส่วนที่เป็น BODY ของเอกสาร HTML ซึ่งจะประกอบไปด้วยข้อมูลจำพวก ข้อความ (Text) , ภาพ (Images) , สี (Colors) , ภาพเคลื่อนไหว(Graphics) และข้อมูลอื่นๆ

2.3.2 Head Elements(BASE , ISINDEX , LINK , META , SCRIPT , STYLE , TITLE)

ซึ่งเป็นอิลลิเมนต์ที่บรรจุอยู่ภายในส่วนที่เป็น HEAD

- BASE ใช้ในการอ้างอิงไปยังข้อมูลที่อยู่ภายนอก ยกตัวอย่าง เช่น ถ้าแอดเดรส (Address) ของภาพ ถูกจัดเก็บอยู่บนเวปไซค์ (Web Site) ที่ < img src = http://www.w3schools.com/images/back40.gif > ถ้าเราใช้ BASE เป็นตัวอ้างอิงในส่วนที่เป็น HEAD จะได้เป็น < base href = http://www.w3schools.com/ > หลังจากนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเพิ่มในส่วนที่เป็นภาพ ก็จะใช้แอตเดรสที่อ้างอิงจาก BASE แทนที่ใช้แอตเดรสจริงๆ จะได้เป็น < img src = " images/smile.gif " >

- ISINDEX ใช้ในการกำหนด Single Line Text Input แต่โดยส่วนมากแล้วจะนิยมใช้ INPUT แทน
- LINK ใช้ในการเชื่อมความสัมพันธ์ระหว่างเอกสาร HTML
- META ใช้ในการกำหนดข้อมูล META ของเอกสาร HTML เช่น คำจำกัดความต่างๆ และคำที่ใช้ในการค้นหาข้อมูลภายในเอกสาร HTML ที่ใช้กับเซิร์สเอ็นจิน(Search Engine) ต่างๆ
- SCRIPT ใช้ในการเรียกใช้สคริปต์ (Script) ต่างๆ เช่นจาวาสคริปต์ (Java Script)
- STYLE ใช้ในการกำหนดเอกสาร HTML ให้เป็นรูปแบบเฉพาะตัว
- TITLE ใช้ในการกำหนดหัวเรื่องของเอกสาร HTML

2.3.3 Generic Block-Level Elements (ADDRESS , BLOCKQUOTE , CENTER , DEL , DIV , H1 , H2 , H3 , H4 , H5 , H6 , HR , INS , ISINDEX , NOSCRIPT , P , PRE)

- ADDRESS ใช้ในการกำหนดแอตเดรสเริ่มต้น
- BLOCKQUOTE ใช้ในการกำหนดที่ว่างด้านซ้ายและขวาของเอกสาร HTML
- CENTER ใช้ในการกำหนดข้อความให้อยู่ตรงกลางตามแนวนอน
- DEL ใช้ในการกำหนดข้อความที่ถูกลบของเอกสาร HTML
- DIV ใช้ในการกำหนดส่วนที่ถูกแบ่งออกของเอกสาร HTML
- H1 – H6 ใช้ในการกำหนดขนาดของข้อความ
- HR ใช้ในการแสดงเส้นตามแนวนอน แต่โดยส่วนมากแล้วจะนิยมใช้ Style Sheet แทน
- INS ใช้ในการเพิ่มข้อความเข้าไปในเอกสาร HTML
- ISINDEX ใช้ในการกำหนด Single Line Text Input แต่โดยส่วนมากแล้วจะนิยมใช้ INPUT แทน
- NOSCRIPT ใช้ในการแสดงข้อความ เมื่อเกิดการเรียกใช้ สคริปต์แต่บาว์เซอร์ไม่สามารถแสดงผลได้
- P ใช้ในการกำหนดย่อหน้าของเอกสาร HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PRE ใช้ในการกำหนดข้อความภายในเอกสาร HTML โดยจะแสดงข้อความที่อยู่ภายใน PRE ในลักษณะที่เป็น Fix-pitch Font คือ มีลักษณะเหมือนกับข้อความที่พิมพ์ เช่น จำนวนช่องว่าง และ คำสิ้นสุดของประโยค ยกตัวอย่างเช่น

2.3.4 Lists (DIR , DL , DT , DD , LI , MENU , OL , UL)

ซึ่งเป็นอิลลิเมนต์ที่ใช้แสดงส่วนที่เป็นลิสต์ที่ปรากฏอยู่ในเอกสาร HTML

- DIR ใช้ในการแสดง ไดเรอเทอรี่ลิสต์ (Directory List) แต่โดยส่วนมากแล้วจะนิยมใช้ UL แทน
- DL ใช้ในการกำหนดส่วนเริ่มต้นของเดฟฟินิชั่นลิสต์ (Definition List)
- DTใช้ในการกำหนดเริ่มต้นของเทอม (Term) ที่อยู่ในเดฟฟินิชั่นลิสต์
- DD ใช้ในการกำหนดส่วนที่คำบรรยาย ของเทอมที่อยู่ในเดฟฟินิชั่นลิสต์
- LI ใช้ในการกำหนดส่วนเริ่มต้นของลิสต์ไอเท็ม (List Item)
- MENU ใช้ในการกำหนดเมนูลิสต์ (Menu List)แต่โดยส่วนมากแล้วจะนิยมใช้ UL แทน
- OL ใช้ในการกำหนดส่วนเริ่มต้นของออเดอร์ลิสต์ (Order List) โดยจะมีลักษณะเป็นข้อมูลเรียงตามลำดับ เช่น ข้อมูลตัวเลข (1,2,3,4,5,…) ข้อมูลตัวอักษร (a,b,c,d,e,…z)
- UL ใช้ในการกำหนดส่วนเริ่มต้นของ อันออเดอร์ลิสต์ (Unorder List) โดยจะมีลักษณะเป็นข้อมูลที่ไม่มีการเรียงลำดับ โดยจะใช้ รูปวงกลม จุด หรือ สี่เหลี่ยมแทน

2.3.5 Tables (TABLE , CAPTION , COLGROUP , COL , THEAD , TBODY , TFOOT , TR , TD , TH)

ซึ่งเป็นอิลลิเมนต์ที่ใช้แสดงส่วนที่เป็นเทเบิล (Table) ที่ปรากฏอยู่ในเอกสาร HTML

- TABLE ใช้ในการกำหนดส่วนเริ่มต้นของ เทเบิล
- CAPTION ใช้ในการกำหนดคำอธิบายที่ใช้กำกับใน เทเบิล
- COLGROUP ใช้ในการรวมกลุ่มของคอลัมท์ (Column) ให้อยู่ในส่วนเดียวกัน
- COL ใช้ในการกำหนดค่าคุณสมบัติภายใน คอลัมท์ที่มีอยู่ในเทเบิล
- THEAD ใช้ในการกำหนดส่วนบนสุดของ เทเบิลทำให้สามารถรวมกลุ่มของแถว(Row) ที่มีอยู่ในเทเบิลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TBODY ใช้ในการกำหนดส่วนถัดจากส่วนบนสุดของ เทเบิลทำให้สามารถรวมกลุ่มของแถวที่มีอยู่ในเทเบิลได้
- TFOOT ใช้ในการกำหนดส่วนล่างสุดของ เทเบิลทำให้สามารถรวมกลุ่มของแถวที่มีอยู่ในเทเบิลได้
- TR ใช้ในการกำหนดแถวที่มีอยู่ใน เทเบิล
- TD ใช้ในการเริ่มต้นใส่ข้อมูลลงในเซล (Cell) ที่มีอยู่ในเทเบิลโดยเซลคือช่องสี่เหลี่ยมที่เกิดจากการตัดกันของแถวและคอลัมท์
- TH ใช้ในการกำหนดส่วนหัวของเซลที่มีอยู่ในเทเบิลโดยข้อมูลจะแสดงเป็นตัวเข้ม

2.3.6 Forms (FORM , BUTTON , FIELDSET , LEGEND , INPUT , LABEL , SELECT , OPTGROUP , OPTION , TEXTAREA)

- FORM ใช้ในการสร้างแบบฟอร์มที่ใช้สำหรับรับค่าอินพุท (Input) จากผู้ใช้
- BUTTON ใช้ในการสร้างปุ่ม โดยสามารถเพิ่มส่วนที่เป็นรูปภาพหรือข้อความลงไปในปุ่มได้
- FIELDSET ใช้ในการสร้างกล่องข้อมูล (Box) ที่ใช้สำหรับใส่ค่าของข้อความหรือค่าที่เป็นข้อมูลอื่นๆได้
- LEGEND ใช้ในการกำหนดคำอธิบายที่ใช้กำกับใน FIELDSET
- INPUT ใช้ในการกำหนดส่วนเริ่มต้นของพื้นที่ อินพุทที่ให้ผู้ใช้ใส่ข้อมูลในไปในพื้นที่นั้นๆ
- LABEL ใช้ในการกำหนดค่าลาเบิว (Label) ของพื้นที่อินพุท
- SELECT ใช้ในการสร้าง Drop-Down Box โดยจะมีลักษณะเป็นแถบข้อมูลที่สามารถแสดงข้อมูลที่ให้เลือกทั้งหมด หรือแสดงเฉพาะข้อมูลเริ่มต้น ก็ได้
- OPTGROUP ใช้ในการกำหนด Option Group เพื่อจัดกลุ่มของข้อมูลที่ให้เลือก
- OPTION ใช้ในการกำหนดค่าของข้อมูลที่อยู่ภายใน Drop-Down Box
- TEXTAREA ใช้ในการสร้าง Text area ไว้สำหรับให้ผู้ใช้ใส่ข้อมูล โดยข้อมูลภายใน Text area จะมีลักษณะเป็น Fix-pitch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7 Special Inline Element (A , APPLET , BASEFONT , BDO , BR , FONT , IFRAME , IMG , MAP , AREA , OBJECT , PARAM , Q , SCRIPT , SPAN , SUB , SUP)

- A ใช้ในการกำหนดการเชื่อมต่อไปยังเอกสารอื่นหรือว่าภายในเอกสารเดียวกัน
- APPLET ใช้ในการกำหนดการเรียกใช้งาน จาวาแอปเปิต (Java Applet) แต่โดยส่วนมากแล้วจะนิยมใช้ OBJECT แทน
- BASEFONT ใช้ในการกำหนดส่วนประกอบต่างๆที่เกี่ยวข้องกับ ฟอร์น (Font) แต่โดยส่วนมากแล้วจะนิยมใช้ Style Sheet แทน
- BDO ใช้ในการกลับทิศทางการแสดงข้อความ จากซ้ายไปขวา เปลี่ยนเป็นจากขวาไปซ้าย
- BR ใช้ในการขึ้นบรรทัดข้อความใหม่
- FONT โดยส่วนมากแล้วจะนิยมใช้ Style Sheet แทน
- IFRAME ใช้ในการสร้าง เฟรม (Frame) ที่บรรจุเอกสาร HTML อื่นแต่สามารถนำมาแสดงผลบนเอกสาร HTML ที่ต้องการใช้งาน
- IMG ใช้ในการแทรกรูปภาพลงบนเอกสาร HTML
- MAP ใช้ในการสร้าง ลิงค์ที่อยู่ในรูปภาพ โดยสามารถกำหนดว่าลิงค์อยู่บนพิกัดใดบนรูปภาพได้
- AREA ใช้ในการกำหนดพิกัดที่อยู่ในรูปภาพ
- OBJECT ใช้ในการแทรกมัลติมิเดีย เช่น ภาพ , เสียง , VRML ลงบนเอกสาร HTML
- PARAM ใช้ในการกำหนดเงื่อนไขต่างๆลงใน OBJECT และ APPLET เช่น เวลาที่ใช้ในการแสดงเสียง เป็นต้น
- Q ใช้ในการกำหนดค่าเริ่มต้นของ Short Quotation
- SCRIPT ใช้ในการเรียกใช้สคริป์ ต่างๆเช่นจาวาสคริป์
- SPAN ใช้งานร่วมกันกับ LANG , ID , DIR , CLASS
- SUB ใช้ในการกำหนดข้อความให้อยู่ในรูปของตัวห้อย
- SUP ใช้ในการกำหนดข้อความให้อยู่ในรูปของตัวยก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.8 Phrase Element (ABBR , ACRONYM , CITE , CODE , DEL , DFN , EM , INS , KBD , SAMP , STRONG , VAR)

- ABBR ใช้ในการกำหนดอักษรย่อ
- ACRONYM ใช้ในการกำหนดอักษรย่อที่มีความยาวมากกว่าแบบ ABBR
- CITE ใช้ในการกำหนด Citation
- CODE ใช้ในการกำหนดว่า ข้อความอยู่ในรูปของภาษาคอมพิวเตอร์
- DEL ใช้ในการกำหนดข้อความที่ถูกลบของเอกสาร HTML
- DFN ใช้ในการกำหนด Definition Term
- EM ใช้ในการกำหนด Emphasis Text หรือข้อความที่มีความสำคัญ
- INS ใช้ในการเพิ่มข้อความเข้าไปในเอกสาร HTML
- KBD ใช้ในการกำหนดข้อความที่มาจาก Keyboard
- SAMP ใช้ในการแสดง Sample Output เช่น มาจากโปรแกรมหรือมาจากส่วนที่เป็น Script
- STRONG ใช้ในการกำหนด Strong Emphasis Text หรือข้อความที่มีความสำคัญมาก
- VAR ใช้ในการกำหนด Variable ต่างๆ

2.3.9 Font Style Elements (B , BIG , I , S , SMALL , STRIKE , TT , U)

- B ใช้ในการกำหนดรูปแบบของข้อความ โดยแสดงผลเป็นตัวเข้ม
- BIG ใช้ในการกำหนดรูปแบบของข้อความ โดยแสดงผลเป็นตัวขนาดใหญ่
- I ใช้ในการกำหนดรูปแบบของข้อความ โดยแสดงผลเป็นตัวเอียง
- S ใช้งานเหมือนกับคำสั่ง DEL ดังนั้นจึงนิยมใช้ DEL แทน
- SMALL ใช้ในการกำหนดรูปแบบของข้อความ โดยแสดงผลเป็นตัวเล็ก
- STRIKE ใช้งานเหมือนกับคำสั่ง DEL ดังนั้นจึงนิยมใช้ DEL แทน
- TT ใช้ในการกำหนดข้อความให้อยู่ในรูปของ Teletype หรือ Monospace Text
- U ใช้ในการกำหนดข้อความให้อยู่ในรูปของ Underline Text แต่โดยส่วนมากแล้วจะนิยมใช้ Style Sheet แทน

2.3.10 Frames (FRAMESET , FRAME , IFRAME)

- FRAMESET ใช้ในการจัดการเกี่ยวกับการกำหนดคุณสมบัติต่างๆของเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- FRAME ใช้ในการกำหนดเฟรมที่ใช้ในการแสดงข้อมูลต่างๆผ่านทางจอภาพ
- IFRAME ใช้ในการสร้างเฟรมที่บรรจุเอกสาร HTML อื่นแต่สามารถนำมาแสดงผลบนเอกสาร HTML ที่ต้องการใช้งาน

2.4 การพัฒนาระบบเชิงวัตถุ

การพัฒนาระบบซอฟต์แวร์หรือระบบสารสนเทศในอดีตนั้นจะใช้วิธีการพัฒนาแบบโครงสร้าง (Structural) ซึ่งการพัฒนาระบบแบบโครงสร้างนี้มีข้อดีเหนือกว่าการพัฒนาระบบโดยไม่ใช้โมเดลแบบใดๆช่วยอย่างชัดเจน แต่อย่างไรก็ดีการพัฒนาระบบแบบโครงสร้างนี้ก็ใช่ว่าจะไม่มีปัญหาเลย ปัญหาหลักๆของการพัฒนาระบบแบบนี้ก็คือ โมเดล (Model) ในการวิเคราะห์และออกแบบระบบจะไม่เชื่อมต่อไปถึงรายละเอียดในการพัฒนาระบบ คือบ่งบอกแต่เพียงว่าระบบมีลักษณะการทำงานอย่างไรแต่ไม่ได้บอกถึงวิธีในการพัฒนาระบบหรือรูปแบบในการเขียน โ ปรแกรมเลย ทำให้หลายๆครั้งระบบที่พัฒนาขึ้นมามีปัญหาเนื่องจากว่าการตรวจสอบความถูกต้องของระบบในเชิงพฤติกรรมนั้นทำได้ยากและไม่มีโมเดลใดๆที่รองรับ จึงได้มีผู้คิดค้นและพัฒนาวิธีการหรือโมเดลในการพัฒนาระบบเชิงวัตถุ (Object-Oriented) ขึ้นมา

การพัฒนาระบบเชิงวัตถุเป็นวิธีการในการพัฒนาซอฟต์แวร์รูปแบบใหม่ ซึ่งการพัฒนาระบบด้วยวิธีการเชิงวัตถุนั้นผู้พัฒนาระบบจะมองระบบและส่วนประกอบหรือระบบย่อย (Subsystem) เป็นวัตถุ เนื่องจากสิ่งต่างๆที่มีอยู่ในโลกความเป็นจริงนั้นก็ถือได้ว่าเป็นวัตถุได้อยู่แล้ว และด้วยการมองวัตถุเป็นระบบทำให้ผู้พัฒนาระบบสามารถมองระบบได้ชัดเจนมากยิ่งขึ้น ส่งผลให้การวิเคราะห์และออกแบบระบบทำได้ง่ายและมีความถูกต้องมากขึ้น

การพัฒนาระบบเชิงวัตถุนั้นประกอบไปด้วยขั้นตอนสำคัญ 2 ขั้นตอนคือ ขั้นตอนของการวิเคราะห์และการออกแบบระบบเชิงวัตถุ และขั้นตอนของการพัฒนาระบบ ในการวิเคราะห์และออกแบบระบบเชิงวัตถุนี้ผู้พัฒนาจะมองภาพของระบบออกเป็นยูสเคส (Use Case) ซึ่งก็คือการมองฟังก์ชันการทำงานหลักของระบบและซีนาริโอ (Scenario) ซึ่งเป็นรายละเอียดการทำงานตาม ยูสเคส จากนั้นผู้พัฒนาก็จะมาออกแบบวัตถุและคลาสพร้อมคุณลักษณะที่จะมีในระบบโดยขั้นตอนของการออกแบบและพัฒนาระบบนี้ ผู้พัฒนาจะเขียนออกมาเป็นแผนภาพยูสเคส (Use Case Diagram), แผนภาพแสดงลำดับขั้นตอนการทำงาน (Sequence Diagram) , แผนภาพแสดงการทำงานร่วมกันของวัตถุ (Collaboration Diagram) , แผนภาพวัตถุ (Object Diagram) , แผนภาพคลาส (Class Diagram) และแผนภาพสถานะ (State Diagram) โดยหลังจากที่ผู้พัฒนาได้วิเคราะห์และออกแบบระบบเป็นที่เรียบร้อยแล้วก็จะเข้ามาสู่ขั้นตอนของการพัฒนาระบบ ซึ่งในขั้นตอนของการพัฒนาระบบนี้ก็มีภาษาในการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming Language) ซึ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาในการเขียนโปรแกรมที่สนับสนุนการสร้างคลาสและวัตถุ ส่งผลให้สิ่งที่ได้จากการวิเคราะห์และออกแบบนั้นสามารถนำมาใช้ในการเขียนโปรแกรมได้

นอกจากนั้นการวิเคราะห์และออกแบบระบบเชิงวัตถุมีโมเดลในการช่วยทำงานรองรับอยู่ ซึ่งโมเดลที่เป็นมาตรฐานอยู่ในขณะนี้มีชื่อว่า UML (Unified Modeling Language) ซึ่งเป็นกระบวนการความคิด (Methodology) และเป็นโมเดลที่สนับสนุนการวิเคราะห์และออกแบบระบบเชิงวัตถุที่สามารถเข้าใจได้ง่ายและสามารถใช้ประกอบในขั้นตอนการเขียนโปรแกรมได้ด้วย

2.5 ความซับซ้อนของซอฟต์แวร์

ซอฟต์แวร์ทุกประเภทมีความซับซ้อนอยู่ภายในตัวแม้กระทั่งซอฟต์แวร์ขนาดเล็ก ซึ่งความซับซ้อนเหล่านี้ผู้พัฒนาซอฟต์แวร์อาจจะสามารถจัดการได้ด้วยความรวดเร็ว แต่ซอฟต์แวร์ที่มีขนาดใหญ่ที่มีความซับซ้อนมากขึ้น บางครั้งนักพัฒนาหรือกลุ่มของนักพัฒนาจำเป็นต้องมีเครื่องมือช่วยในการจัดการกับความซับซ้อนเหล่านี้ โดยทั่วไปสามารถแบ่งความซับซ้อนของซอฟต์แวร์ออกได้เป็น 4 ประเภทใหญ่ๆคือ

2.5.1 ความซับซ้อนของโดเมนปัญหา

โดเมนของปัญหานั้นมีความซับซ้อนอยู่ในตัว ผู้ที่จะพัฒนาซอฟต์แวร์ ให้กับระบบเหล่านี้ต้องมีความเข้าใจในโดเมนของปัญหาซึ่งก็คือ ความต้องการของระบบนั่นเอง และผู้พัฒนายังจะต้องทราบความต้องการของระบบแบบ Nonfunctional เช่น ประสิทธิภาพ ราคา และความเชื่อถือได้ด้วย

บางครั้งความซับซ้อนของระบบก็มาจากความไม่เข้าใจระหว่างตัวผู้ใช้ระบบกับผู้พัฒนาระบบ ผู้ใช้ระบบจะมีมุมมองในแง่ที่ว่าระบบจะต้องทำอะไรให้กับตัวผู้ใช้ได้บ้าง แต่ผู้พัฒนาระบบจะมองระบบเป็นโครงสร้างใหญ่ๆ และในหลายครั้งตัวผู้ใช้เองก็ยังไม่ทราบว่าตัวเองต้องการอะไร ซึ่งสิ่งเหล่านี้เป็นข้อมูลที่นักพัฒนาระบบต้องการ

2.5.2 ความยากในการจัดการกระบวนการพัฒนา

งานหลักของนักพัฒนาระบบคือจะต้องทำให้การพัฒนาและการใช้งานระบบเป็นเรื่องง่ายอย่างไรก็ดีเนื่องจากในแต่ละระบบก็จะมีความเฉพาะเจาะจงที่แตกต่างกันออกไป ส่งผลให้ผู้พัฒนาระบบจะเป็นต้องลงมือเขียนโปรแกรมด้วยตัวเองเป็นจำนวนมาก ถึงแม้ว่าในปัจจุบันจะมีเครื่องมือในการพัฒนาที่ครบถ้วนและไว้ใจได้มากกว่าในอดีต แต่ถ้าซอฟต์แวร์ที่พัฒนาขึ้นมีจำนวนโค้ด (Code) มากมายหลายบรรทัดแล้ว ปัญหาจะตกมาอยู่ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวผู้พัฒนา เนื่องจากว่าไม่มีใครที่จะสามารถเข้าใจโค้ดจำนวนมหาศาลได้ทั้งหมด ดังนั้นจึงจำเป็นต้องแบ่งงานให้กับลูกทีมช่วยกันพัฒนาส่วนต่างๆของซอฟต์แวร์ขึ้นมา ในการดูแลทีมนักพัฒนาจำเป็นต้องมีการสื่อสารระหว่างกันเพื่อที่จะทำงานร่วมกัน สิ่งที่ยากก็คือการจัดการให้การออกแบบและพัฒนามีความถูกต้องครบถ้วน และเป็นหนึ่งเดียวกันตลอดจนกว่าการพัฒนาเสร็จสมบูรณ์

2.5.3 ซอฟต์แวร์มีความยืดหยุ่น

โครงการซอฟต์แวร์นั้นแตกต่างจากโครงการในการสร้างตึกหรือถนนที่มีตัวตนจับต้องได้ ซอฟต์แวร์ นั้นมีความยืดหยุ่น(Flexibility)ต่อความต้องการของผู้ใช้สูงเนื่องจากนักพัฒนาและผู้ใช้แต่ละคนต่างก็มีมุมมองในปัญหาที่แตกต่างกัน ดังนั้นจึงจำเป็นต้องมีโมเดลบางชนิดเพื่อใช้แทนมุมมองหรือความต้องการของระบบที่ทำให้นักพัฒนาและผู้ใช้ทุกๆคนเข้าใจเหมือนกัน

2.5.4 ปัญหาที่เกิดการเปลี่ยนแปลงคุณลักษณะของระบบ

ในซอฟต์แวร์ประยุกต์ขนาดใหญ่นั้นจะประกอบด้วยตัวแปรจำนวนมากที่ถูกควบคุมและใช้งาน ซึ่งกลุ่มของตัวแปรเหล่านี้ก็มีค่าแอดเดรสและตำแหน่งในสแต็ก (Stack) เป็นของตัวเอง เนื่องจากซอฟต์แวร์ นั้นทำงานบนคอมพิวเตอร์ซึ่งเป็นการประมวลผลแบบไม่ต่อเนื่อง (Discrete Computing) ซึ่งจะไม่เหมือนกับระบบอิเล็กทรอนิกส์ที่ทำงานแบบต่อเนื่อง (Continuous) กล่าวคือเมื่ออินพุทเปลี่ยนแปลงเอาท์พุทก็จะเปลี่ยนแปลงตาม ในการประมวลผลแบบไม่ต่อเนื่องจำเป็นต้องมีตัวแปรจำนวนมากเหล่านี้เพื่อเป็นตัวบ่งชี้สถานะของซอฟต์แวร์ ทำให้ซอฟต์แวร์ที่พัฒนาขึ้นมีสถานะที่แตกต่างกันจำนวนมหาศาล และเมื่อมีการเปลี่ยนแปลงซอฟต์แวร์ส่งผลให้สถานะของซอฟต์แวร์มีมากขึ้น ซึ่งบางครั้งนักพัฒนาก็ไม่สามารถบ่งบอกได้ว่าในขณะนั้นซอฟต์แวร์ที่พัฒนาขึ้นอยู่ในสถานะใด

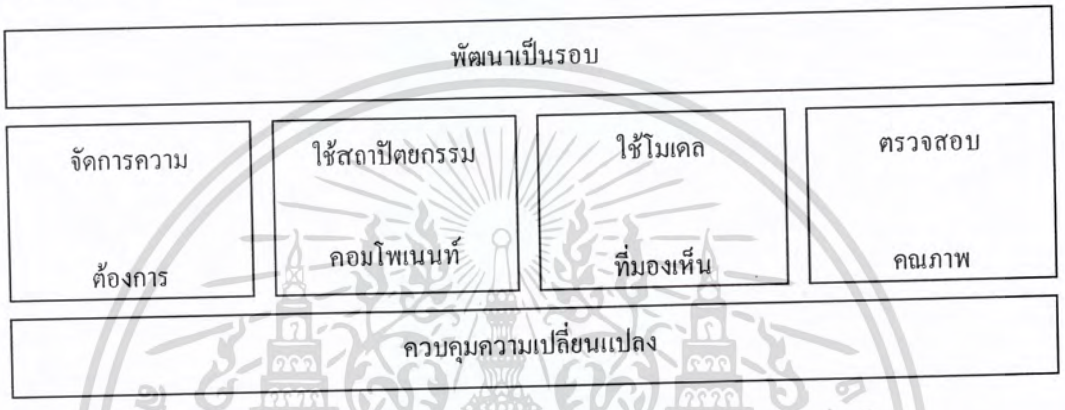
2.6 วิศวกรรมซอฟต์แวร์ (Software Engineering)

นักพัฒนาซอฟต์แวร์จำเป็นต้องใช้หลักการทางวิศวกรรมในการพัฒนาซอฟต์แวร์แต่ว่าโครงสร้างของซอฟต์แวร์นั้นไม่เหมือนกับผลงานทางวิศวกรรมอื่นๆ ดังนั้นการพัฒนาจึงมีกระบวนการทางวิศวกรรมเป็นของตัวเอง สำหรับกระบวนการทางวิศวกรรมซอฟต์แวร์ที่ดีนั้น จะต้องสามารถทำให้การพัฒนาซอฟต์แวร์เสร็จได้ทันเวลา โดยใช้งบประมาณตามที่ตั้งไว้ และที่สำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์จะต้องตรงตามความต้องการของผู้ใช้ ถ้าซอฟต์แวร์ใดมีคุณสมบัติเหล่านี้จะถือว่าซอฟต์แวร์นั้นเป็นซอฟต์แวร์ที่ดี

ในการพัฒนาผู้พัฒนาจำเป็นต้องใช้หลักการต่างๆประกอบกันเพื่อให้ได้ซอฟต์แวร์ที่ดี ซึ่งหลักการในการพัฒนาเหล่านี้จะประกอบไปด้วย การพัฒนาเป็นรอบ , จัดการความต้องการ , การใช้สถาปัตยกรรมของคอมโพเนนท์, ใช้โมเดลที่มองเห็น , มีการตรวจสอบคุณภาพ และต้องมีการควบคุมการเปลี่ยนแปลง

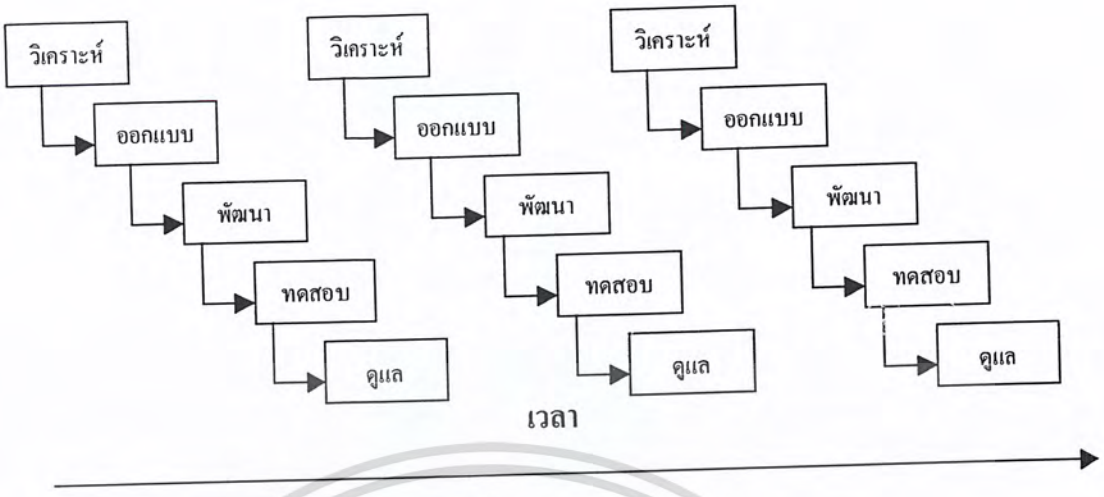


2.6.1 การพัฒนาเป็นรอบ (Develop Iteratively)

เป็นการใช้โมเดลทางวิศวกรรมซอฟต์แวร์ที่นักพัฒนาส่วนใหญ่ใช้ คือ โมเดลน้ำตก (Water fall Model) ให้มีการวนพัฒนา กระบวนการพัฒนาซอฟต์แวร์แบบนี้จะกระทำตามกระบวนการ โดยเริ่มตั้งแต่การวิเคราะห์ความต้องการของระบบ ออกแบบระบบ ลงมือพัฒนา ทดสอบและติดตามระบบ ซึ่งในโมเดลนี้จะถือว่าหลังจากที่ได้วิเคราะห์ความต้องการเรียบร้อยแล้วจะต้องไม่มีการเปลี่ยนแปลงความต้องการอีก แต่ในความเป็นจริงก็คือการเปลี่ยนแปลงความต้องการนั้นมีได้ตลอดเวลา ทำให้การทำตามโมเดลนี้ทำให้ซอฟต์แวร์ที่พัฒนาด้วยโมเดลนี้มีความผิดพลาด ใช้เวลาและงบประมาณเกินกำหนดและซอฟต์แวร์ที่พัฒนาขึ้นนั้นอาจใช้งานไม่ได้เลย

ดังนั้นจึงมีการพัฒนาให้มีการใช้งานในแบบวนรอบพัฒนาขึ้นเพื่อลดความผิดพลาดและความเสียหายที่จะเกิดขึ้น โดยในการพัฒนาแต่ละครั้งก็จะเพิ่มส่วนประกอบของระบบเข้าไปเรื่อยๆจนกระทั่งครบทั้งระบบ และเมื่อพัฒนาในส่วนใดเสร็จแล้วก็จะทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ส่วนนั้นให้เรียบร้อย จากนั้นจึงเริ่มพัฒนาในส่วนต่อไป ซึ่งการที่จะพัฒนาในส่วนใดก่อนนั้นโดยทั่วไปจะวัดจากความเสี่ยง คือถ้าฟังก์ชัน (Function) ใดมีความเสี่ยงน้อยที่สุดก็จะได้รับการพัฒนาก่อน ส่วนฟังก์ชันที่มีความเสี่ยงมากขึ้นก็จะได้รับการพัฒนาตามไป การพัฒนาเป็นรอบแบบนี้ทำให้ระบบค่อยๆขยายตัวจนเป็นระบบที่สมบูรณ์

ด้วยกระบวนการพัฒนาเป็นรอบช่วยให้ผู้พัฒนาระบบได้รับการตอบรับจากผู้ใช้ระบบอยู่เป็นระยะๆ เพราะว่ามีต้นแบบ (Prototype) ให้ผู้ใช้ได้ทดลองใช้งานอยู่ตลอดเวลา ส่งผลให้ผู้พัฒนาทราบความต้องการที่แท้จริงของผู้ใช้ นอกจากนั้นยังช่วยเพิ่มความถูกต้องของระบบได้อีกด้วย เนื่องจากระบบที่พัฒนาออกมาในขั้นต่างๆจะได้รับการทดสอบให้ถูกต้องก่อนที่จะได้รับการพัฒนาต่อ และเมื่อพัฒนาในรอบต่อไปก็ทดสอบเพียงส่วนที่เพิ่มเข้าไปเท่านั้น

2.6.2 จัดการความต้องการ (Manage Requirement)

ความต้องการในที่นี้คือเงื่อนไขหรือความสามารถที่ระบบควรจะต้องมี ส่วนการจัดการความต้องการ(Requirement Management) คือ การค้นหา จัดการ ทำเอกสารความต้องการของระบบ และการสร้างและดูแลข้อตกลงระหว่างผู้ใช้กับผู้พัฒนาระบบในการเปลี่ยนแปลงความต้องการของระบบ

ดังนั้นผู้พัฒนาระบบกับผู้ใช้จะต้องมีข้อตกลงสำหรับการกำหนดความต้องการของระบบ เนื่องจากผู้พัฒนาระบบนั้นไม่สามารถที่จะทราบความต้องการทั้งหมดของผู้ใช้ได้ และในบางครั้งผู้ใช้เองก็อาจจะเกิดการเปลี่ยนแปลงความต้องการขึ้นมา ความต้องการของระบบจึงควรมาจากความเข้าใจที่ตรงกันของผู้ใช้ระบบและผู้พัฒนาระบบ ทั้งนี้เพื่อให้การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาระบบเป็นไปตามเป้าหมายหลัก คือ ผู้ใช้ยอมรับระบบ เพราะผู้พัฒนาไม่สามารถบังคับความเปลี่ยนแปลงความต้องการของระบบได้แต่ผู้พัฒนาสามารถจัดการได้

2.6.3 การใช้สถาปัตยกรรมของคอมโพเนนท์ (Use Component Architecture)

คอมโพเนนท์ในการออกแบบและพัฒนาระบบหมายถึง สิ่งที่ส่วนของระบบทำงานได้เกือบเป็นอิสระและสามารถใช้คอมโพเนนท์อื่นแทนได้ การออกแบบและพัฒนาระบบแบบคอมโพเนนท์ทำให้สถาปัตยกรรมของระบบมีความยืดหยุ่นและสนับสนุนการนำกลับมาใช้ (Reuse) การพัฒนาโดยมีพื้นฐานอยู่บนคอมโพเนนท์ทำให้ผู้พัฒนาสามารถใช้หรือปรับแต่งคอมโพเนนท์ ที่มีอยู่หรือคอมโพเนนท์ที่ได้รับจากผู้พัฒนาคนอื่น การออกแบบระบบเป็นคอมโพเนนท์ยังช่วยเพิ่มการแยกแยะความสัมพันธ์(Modularity)ให้กับระบบ ส่งผลให้การทดสอบและดูแลระบบทำได้ง่ายขึ้น สำหรับคอมโพเนนท์ที่ได้รับการออกแบบมาดีนั้น จะทำให้ผู้พัฒนาสามารถจัดการและควบคุมส่วนคอมโพเนนท์ที่เป็นส่วนหนึ่งของระบบได้

2.6.4 ใช้โมเดลที่มองเห็น (Model Visually)

การใช้โมเดลที่มองเห็นได้จะทำให้ผู้พัฒนาระบบสามารถวางโครงสร้าง วางรูปแบบการติดต่อและคุณลักษณะของระบบได้ดีขึ้น โมเดลที่ได้จากการออกแบบทำให้ผู้พัฒนาสามารถใช้ประกอบระหว่างการพัฒนาได้อีกด้วย และสำหรับทีมพัฒนาที่มีนักพัฒนาหลายๆคนหรือหลายๆทีมโมเดลยังช่วยเป็นเครื่องมือในการสื่อสารระหว่างผู้พัฒนาระบบแต่ละคนหรือแต่ละทีมด้วย

2.6.5 การตรวจสอบคุณภาพ (Verify Quality)

เป็นที่รับรู้และยอมรับกันดีแล้วว่า ค่าใช้จ่ายสำหรับการแก้ไขระบบนั้นมากกว่าค่าใช้จ่ายในการพัฒนาระบบหลายเท่าตัว ด้วยเหตุผลเพียงเท่านี้ก็เพียงพอที่จะทำให้นักพัฒนาระบบและผู้ใช้จึงจำเป็นต้องมีการทดสอบและตรวจสอบความถูกต้องของระบบ ซึ่งความถูกต้องของระบบในที่นี้คือ ระบบมีฟังก์ชันการทำงานตรงตามที่ผู้ใช้ต้องการนั่นเอง

ด้วยการพัฒนาเป็นรอบทำให้นักพัฒนาระบบสามารถสร้างเครื่องมือในการทดสอบระบบแบบอัตโนมัติไปได้พร้อมๆกัน นอกจากการทดสอบความถูกต้องของฟังก์ชันการทำงานของระบบแล้ว ผู้พัฒนายังอาจจะต้องทดสอบคุณภาพของระบบในเรื่องของความเชื่อถือได้ ประสิทธิภาพของซอฟต์แวร์ประยุกต์ และประสิทธิภาพของระบบทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หมดด้วย โดยการทดสอบความน่าเชื่อถือของระบบ ผู้พัฒนาอาจจะใช้เครื่องมือที่มีอยู่ในท้องตลาดทดสอบแทนการพัฒนาขึ้นมาแทน แต่ทั้งนี้ผู้พัฒนาก็ควรจะทดสอบประสิทธิภาพโดยรวมของระบบด้วย เนื่องจากในซอฟต์แวร์ประยุกต์หนึ่งๆอาจจะมีประสิทธิภาพที่ยอมรับได้ แต่เมื่อนำมาทำงานร่วมกันเป็นระบบใหญ่แล้วประสิทธิภาพโดยรวมของระบบอาจจะต่ำเกินกว่าที่ผู้ใช้จะยอมรับก็เป็นได้

2.6.6 การควบคุมการเปลี่ยนแปลง (Control Changes)

เนื่องด้วยในการพัฒนาระบบนั้นอาจจะมีนักพัฒนาหลายๆคนที่ร่วมงานกันหรืออาจจะมีหลายทีมพัฒนา และระบบที่ออกมาก็มีหลายรุ่นและหลายแพลตฟอร์ม (Platform) ดังนั้นจึงมีความจำเป็นที่ผู้ควบคุมการพัฒนาระบบจะต้องมีการควบคุมการเปลี่ยนแปลงของระบบ

วิธีควบคุมการเปลี่ยนแปลงก็คือ การแบ่งระบบออกเป็นระบบย่อยๆและให้ทีมหรือนักพัฒนารับผิดชอบในการพัฒนาระบบย่อยไป และการกำหนดขอบเขตของการพัฒนาของระบบย่อย ทั้งนี้เพื่อไม่ให้ระบบย่อยหนึ่งได้รับผลกระทบเมื่ออีกระบบย่อยหนึ่งมีการเปลี่ยนแปลง และยังช่วยให้การสร้างโมเดล โค้ด และเอกสารเป็นอิสระจากระบบย่อยอื่นๆด้วย ทีมพัฒนาควรจะมีการกำหนดรูปแบบในการนำระบบย่อยต่างๆมารวมกัน ทั้งนี้เพื่อไม่ให้เกิดปัญหาการทำงานระหว่างระบบย่อย ทีมพัฒนาควรจะจัดทำเอกสารไว้ว่า ในระบบแต่ละรุ่นนั้นมีสิ่งใดเปลี่ยนแปลงไป เพื่อให้สามารถสร้างรูปแบบในการทดสอบได้อย่างเหมาะสม

2.7 ข้อดีของการพัฒนาเชิงวัตถุ

2.7.1 ทำให้นักพัฒนาสามารถโมเดลระบบได้อย่างครบถ้วนมากยิ่งขึ้น

ปัญหาหนึ่งในการโมเดลระบบแบบโครงสร้าง (Structural Model) คือ การเปลี่ยนจากมุมมองของการวิเคราะห์เป็นการออกแบบนั้นทำได้ยาก ถึงแม้ว่ามุมมองทั้งสองจะมาจากระบบเดียวกัน แต่การโมเดล ระบบแบบโครงสร้างไม่ได้แสดงถึงความสัมพันธ์ระหว่างการวิเคราะห์และผังงานโครงสร้าง เนื่องจากตัวแผนภาพที่ได้จากการโมเดลระบบแบบโครงสร้างนี้ไม่ได้ถูกออกแบบมาสำหรับการเขียนโปรแกรมนั่นเอง ด้วยเหตุผลนี้ทำให้โปรแกรมที่เขียนขึ้นอาจจะไม่ตรงกับสิ่งที่ได้จากการวิเคราะห์ก็เป็นได้

ในระบบที่พัฒนาด้วยวิธีเชิงวัตถุนั้น ผู้พัฒนาจะใช้โมเดลเดียวกันตลอดการพัฒนา การกำหนดคลาส (Class) และวัตถุในโมเดลที่ได้จากการวิเคราะห์สามารถเปลี่ยนเป็นโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46406

ของโปรแกรมได้โดยตรง ดังนั้นผู้พัฒนาระบบจึงสามารถดูความสัมพันธ์ระหว่างการกำหนดปัญหากับวิธีการแก้ปัญหาและลงมือพัฒนาได้ง่าย

2.7.2 ช่วยเพิ่มความเข้าใจในโดเมนของปัญหา

โมเดลของการพัฒนาเชิงวัตถุนั้นจะมีการแสดงความสัมพันธ์ระหว่างข้อมูลกับการกระทำหรือโอเปอเรชัน (Operation) ที่กระทำกับข้อมูลนั้นอย่างใกล้ชิด เนื่องจากโมเดลนี้มีการจำลองมาจากโลกจริงๆ การแยกแยะเอกลักษณ์(Abstraction)ของการพัฒนาเชิงวัตถุนั้นให้ความเข้าใจและมีความสามารถที่ดีกว่า รวมทั้งมีการตั้งชื่อสำหรับวัตถุนั้นที่นำมาจากชื่อที่มีอยู่จริงในโดเมนของปัญหา ส่งผลให้ผู้ที่เกี่ยวข้องสามารถเข้าใจความต้องการจากโมเดลได้ง่าย มุมมองของวัตถุจะเป็นการมองจากระดับสูงและใกล้กับโดเมนของปัญหามากกว่า แต่ที่สำคัญคือ การโมเดลในการพัฒนาระบบเชิงวัตถุนั้นแทบจะไม่มีมุมมองทางด้านคอมพิวเตอร์เข้าไปเกี่ยวข้องเลย ส่งผลให้ระบบที่ได้รับการพัฒนามีการแยกแยะระหว่างสิ่งที่เป็นอิสระต่อกัน เช่น วัตถุจะแยกออกจากกันอย่างชัดเจน แต่ก็แสดงความสัมพันธ์ที่ใกล้ชิดระหว่างสิ่งที่ต้องเกี่ยวข้องกัน เช่น ข้อมูลกับโอเปอเรชันของข้อมูลนั้นๆอย่างใกล้ชิด

2.7.3 ช่วยเพิ่มเสถียรภาพของการเปลี่ยนแปลง

เนื่องจากว่าการแยกแยะเอกลักษณ์ของการพัฒนาระบบด้วยวิธีเชิงวัตถุนั้นมีพื้นฐานมาจากโลกแห่งความจริง ดังนั้นระบบจึงมีความเสถียรมากกว่า การเปลี่ยนแปลงความต้องการในระบบก็เพียงแต่เพิ่มวัตถุหรือเปลี่ยนแปลงภายในวัตถุหนึ่งๆเท่านั้น โดยที่ไม่ต้องเปลี่ยนแปลงโครงสร้างของระบบทั้งหมด

2.7.4 มีโมเดลที่สนับสนุนการนำกลับมาใช้ (Reuse)

ในโมเดลของการพัฒนาระบบเชิงวัตถุนั้นมีคุณสมบัติที่สนับสนุนการนำกลับมาใช้ใหม่โดยตรงคือ การสืบทอดคุณสมบัติแบบที่สามารถเพิ่มเติมและขยายคุณสมบัติของคอมโพเนนท์โดยที่ไม่ต้องมีการเปลี่ยนแปลงซอร์สโค้ด (Source Code)เก่าเลย คือสามารถเพิ่มโค้ดในส่วนที่แตกต่างจากคอมโพเนนท์เดิมเท่านั้น และการสืบทอดคุณสมบัติแบบที่สามารถเพิ่มหรือแก้ไขคุณสมบัติของคอมโพเนนท์ที่มีอยู่ได้ (Refinement) ตามที่ตนเองต้องการ

2.7.5 สนับสนุนการปรับเปลี่ยนขนาดของระบบ (Scalability)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดประสงค์หลักของวิธีการในการพัฒนาซอฟต์แวร์ก็คือ จะต้องสามารถจัดการกับความซับซ้อนได้ แต่ว่าการพัฒนาระบบแบบโครงสร้างนั้นไม่เหมาะสมกับระบบที่มีขนาดใหญ่ที่มีความซับซ้อนมากกว่าได้ และเนื่องจากระบบแบบโครงสร้างนั้นมีการแยกแยะเอกลักษณ์และการซ่อนรายละเอียด (Encapsulation) ที่ไม่ดี ทำให้มีข้อจำกัดในการปรับเปลี่ยนขนาดของระบบเพราะระบบจะมีโครงสร้างที่ซับซ้อนมากขึ้นเมื่อมีปัญหากับระบบมากขึ้น ในกรณีนี้การพัฒนาระบบด้วยวิธีเชิงวัตถุมีข้อได้เปรียบกว่า เนื่องจากมีการแยกแยะเอกลักษณ์และการซ่อนรายละเอียดที่ดี ส่งผลให้คอมโพเนนท์แต่ละส่วนแยกออกจากกันอย่างชัดเจน และการใช้เครื่องหมายเดียวกันตลอดกระบวนการของการพัฒนา ทำให้นักพัฒนาสามารถเปลี่ยนจากการวิเคราะห์เป็นการออกแบบได้ง่าย

2.7.6 สนับสนุนการออกแบบระบบที่เชื่อถือได้และมีความปลอดภัย (Reliability and Safety)

เนื่องจากการพัฒนาด้วยวิธีเชิงวัตถุมีการแยกแยะเอกลักษณ์และการซ่อนรายละเอียดที่ดีกว่าทำให้การสื่อสารระหว่างวัตถุที่ไม่เกี่ยวข้องกันจำกัดอยู่ด้วยอินเทอร์เฟซ (Interface) ที่ผู้พัฒนาได้กำหนดไว้เท่านั้น ส่งผลให้ระบบที่พัฒนาระบบด้วยวิธีเชิงวัตถุมีความเชื่อถือได้มากกว่าเพราะผู้พัฒนาสามารถควบคุมการติดต่อระหว่างคอมโพเนนท์ได้ นอกจากนี้ยังสามารถเพิ่มเงื่อนไขก่อนหน้าและท้าย (Pre and Post Condition) เพื่อให้ระบบสามารถทำงานได้อย่างถูกต้องด้วย

ในขณะที่ภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุมีคุณสมบัติของการเอ็กเซพชัน (Exception) เพื่อให้ผู้พัฒนาสามารถตรวจสอบและรองรับกับความผิดพลาดที่อาจจะ เกิดขึ้นได้ และการที่การพัฒนาเชิงวัตถุสนับสนุนการนำสิ่งที่มีอยู่กลับไปใช้ ดังนั้นคอมโพเนนท์ได้รับการทดสอบว่าถูกต้องแล้วก็สามารถนำไปใช้กับระบบใหม่ได้โดยที่ไม่ก่อให้เกิดปัญหา

2.7.7 สนับสนุนการทำงานแบบพร้อมกัน (Concurrency)

การทำงานไปพร้อมกันนั้นเป็นรูปแบบการทำงานของสิ่งที่มีตัวตนในโลกของความเป็นจริง และยังเป็นคุณสมบัติสำคัญที่มีอยู่ในระบบคอมพิวเตอร์ยุคใหม่ด้วย การพัฒนาแบบโครงสร้างไม่มีสัญลักษณ์สำหรับการทำงานพร้อมกัน การจัดการงาน (Task) หรือการซิงโครไนเซชัน (Synchronization)ระหว่างงาน ดังนั้น โครงสร้างที่สำคัญๆของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบอาจจะไม่สามารถพัฒนาขึ้นได้โดยการใช้รูปแบบการพัฒนาแบบโครงสร้างมาตรฐานได้

การพัฒนาระบบด้วยวิธีเชิงวัตถุสนับสนุนการทำงานแบบพร้อมกันในตัวเอง และรายละเอียดของงาน การทำซิงโครไนเซชันระหว่างงานก็สามารถแสดงออกมาในแผนภาพได้ ซึ่งแผนภาพเหล่านี้เป็นเครื่องมือที่มีความเหมาะสมสำหรับการสร้างระบบที่มีเงื่อนไขทางด้านประสิทธิภาพ

2.8 หลักการพัฒนาระบบด้วยวิธีเชิงวัตถุ

เทคโนโลยีการพัฒนาระบบด้วยวิธีเชิงวัตถุเป็นหลักการทางวิศวกรรม ซึ่งบรรจุไว้ด้วยองค์ประกอบต่างๆที่เรียกว่า โมเดลของวัตถุ (Object Model) ในโมเดลของวัตถุนั้นก็แบ่งออกเป็นการแยกแยะเอกลักษณ์ (Abstraction) การซ่อนรายละเอียด (Encapsulation) การรวมกลุ่มความสัมพันธ์ (Modularity) ลำดับชั้น (Hierarchy) ชนิด (Typing) การทำงานพร้อมกัน(Concurrency) และการรักษาสถานะ (Persistence)

2.9 หลักการของวัตถุ

การพัฒนาระบบแบบโครงสร้างนั้นจะพยายามให้นักพัฒนาระบบแก้ปัญหาด้วยการแบ่งปัญหาออกเป็นส่วนๆแล้วแก้ปัญหาในแต่ละส่วนด้วยอัลกอริทึม (Algorithm) และโครงสร้างข้อมูลเฉพาะสำหรับการพัฒนาระบบด้วยวิธีเชิงวัตถุก็เช่นเดียวกันแต่จะมีมุมมองต่อปัญหาเป็นวัตถุประกอบกับการเขียนโปรแกรมด้วยภาษาในการเขียนโปรแกรมเชิงวัตถุ และยังมีข้อดีจากโมเดลของวัตถุ ซึ่งสามารถนำไปใช้ประยุกต์กับการออกแบบส่วนติดต่อกับผู้ใช้ ระบบฐานข้อมูลหรือแม้แต่การออกแบบสถาปัตยกรรมฮาร์ดแวร์ของคอมพิวเตอร์ เนื่องจากหลักการของการพัฒนาระบบด้วยวิธีเชิงวัตถุนั้นเป็นแนวความคิดในการจัดการกับความซับซ้อนของสิ่งต่างๆอย่างมีระบบ จึงสามารถนำไปใช้ประยุกต์ในงานต่างๆได้อย่างกว้างขวาง

การวิเคราะห์และออกแบบระบบด้วยวิธีเชิงวัตถุนั้นยังคงมีข้อดีของการพัฒนาแบบเก่า แต่ก็ได้เพิ่มข้อดีใหม่ๆเข้าไปด้วย โดยหลักการของวัตถุนั้นจะมีการแบ่งปัญหาออกเป็นวัตถุ ซึ่งมีความสัมพันธ์กันน้อย ทำให้การดูแลและแก้ไขส่วนต่างๆของระบบหรือวัตถุสามารถทำได้ง่าย และมีผลกระทบต่อส่วนอื่นๆของระบบน้อยที่สุด

2.9.1 ภาษาในการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming , OOP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมด้วยภาษาเชิงวัตถุเป็นวิธีในการสร้างโปรแกรม โปรแกรมจะถูกจัดให้มีโครงสร้างในการทำงานร่วมกันของวัตถุ ซึ่งวัตถุนั้นจะเป็นตัวตนหรือเป็นอินสแทนซ์ (Instance) ของคลาส และคลาสก็เป็นส่วนหนึ่งของลำดับชั้นของคลาสที่มีความสัมพันธ์ในการสืบทอดคุณสมบัติ(Inheritance)โดยภาษาที่จะเป็นภาษาในการเขียนโปรแกรมเชิงวัตถุได้นั้นจะต้องมีการสนับสนุนคุณสมบัติ 3 ข้อดังนี้คือ

- จะต้องเป็นภาษาที่ใช้วัตถุไม่ใช่อัลกอริทึม
- วัตถุที่ปรากฏในการเขียนโปรแกรมต้องเป็นอินสแทนซ์ของคลาส
- คลาสจะต้องมีความสัมพันธ์กับคลาสอื่นด้วยการสืบทอดคุณสมบัติ

ถ้าภาษาใดขาดคุณสมบัติหนึ่งในสามข้อนี้ จะไม่เรียกภาษานั้นว่าเป็นภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ

2.9.2 การออกแบบระบบด้วยวิธีเชิงวัตถุ (Object Oriented Design , OOD)

การออกแบบด้วยวิธีเชิงวัตถุ เป็นวิธีในการออกแบบกระบวนการในการแยกย่อยและเป็นการใช้สัญลักษณ์แบบการพัฒนาระบบด้วยวิธีเชิงวัตถุสำหรับโมเดลระบบทั้งในทางตรรก (Logic) และทางกายภาพ (Physical) รวมถึงโมเดการทำงานแบบสแตติก (Static) และไดนามิก (Dynamic) สำหรับระบบที่ต้องการออกแบบ

ด้วยการแยกย่อยปัญหาด้วยวิธีเชิงวัตถุ ทำให้การออกแบบระบบด้วยวิธีเชิงวัตถุต่างจากการออกแบบระบบแบบโครงสร้าง เนื่องจากในการออกแบบระบบด้วยวิธีเชิงวัตถุนั้น ผู้ออกแบบระบบ จะมีการแยกแยะเอกลักษณ์โดยยึดวัตถุเป็นหลัก ในขณะที่การออกแบบระบบแบบโครงสร้าง จะมีการแยกแยะเอกลักษณ์ โดยดูจากอัลกอริทึมการทำงานเป็นหลัก

2.9.3 การวิเคราะห์ระบบด้วยวิธีเชิงวัตถุ (Object Oriented Analysis , OOA)

การวิเคราะห์ระบบด้วยวิธีเชิงวัตถุนั้น จะเน้นที่การโมเดล (Model) ระบบในโลกความเป็นจริงโดยใช้มุมของการพัฒนาระบบด้วยวิธีเชิงวัตถุ การวิเคราะห์ด้วยวิธีเชิงวัตถุเป็นวิธีวิเคราะห์และตรวจสอบความต้องการโดยใช้มุมมองจากคลาสและวัตถุจากคำศัพท์ที่อยู่ในโดเมนของปัญหา

โดยทั้งภาษาในการเขียนโปรแกรมเชิงวัตถุ การออกแบบและการวิเคราะห์ระบบด้วยวิธีเชิงวัตถุนั้นจะเกี่ยวข้องกันดังนี้คือ หลังจากที่ได้วิเคราะห์ระบบแล้วก็จะได้โมเดลของระบบเชิงวัตถุ ซึ่งจะใช้เป็นหลักในการออกแบบ และเมื่อออกแบบเสร็จก็จะได้โครง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างหลักหรือพิมพ์เขียว(Blueprint)สำหรับการเขียนโปรแกรมด้วยภาษาในการเขียนโปรแกรมเชิงวัตถุต่อไป

2.9.4 การทำงานพร้อมกัน (Concurrency)

ระบบโดยทั่วไปจะต้องสามารถตอบสนองต่อเหตุการณ์หลายๆ เหตุการณ์ได้ในเวลาเดียวกัน ซึ่งคุณสมบัตินี้จะเป็นปัญหาสำหรับระบบที่มีหน่วยประมวลผลเพียงตัวเดียว แต่สำหรับระบบที่มีหน่วยประมวลผลหลายๆตัวแล้ว ระบบสามารถทำงานหลายๆงานบนหน่วยประมวลผลต่างๆไปได้พร้อมกัน โดยการทำงานพร้อมกันนั้นแบ่งออกเป็น 2 แบบ คือ เฮฟวี่เวต (Heavyweight Concurrency) และ ไลท์เวต (Lightweight Concurrency) สำหรับเฮฟวี่เวตนั้นจะทำงานแต่ละงานเป็นอิสระ และอยู่บนพื้นที่ของหน่วยความจำของตัวเอง ซึ่งเรียกว่าโพรเซส (Process) ส่วนไลท์เวตนั้น การทำงานของแต่ละงานเกี่ยวข้องกับงานอื่นและอาจจะมีการแชร์พื้นที่ของหน่วยความจำร่วมกันก็ได้ ซึ่งเรียกว่า เธรด (Thread) สำหรับภาษาในการเขียนโปรแกรมนั้น จะเน้นที่การแยกแยะเอกลักษณ์ข้อมูล การซ่อนรายละเอียดและการสืบทอดคุณสมบัติ ส่วนการพร้อมกันนั้นจะเกี่ยวข้องกับการแยกแยะเอกลักษณ์ของโพรเซสและการซิงโครไนเซชันสามารถแบ่งวัตถุออกได้เป็น 2 ชนิดคือ วัตถุ ซึ่งส่วนใหญ่จะจำลองมาจากวัตถุที่มีอยู่ในโลกความจริง ทำงานได้ด้วยตัวเองจะเรียกว่าวัตถุที่ทำงานแบบ แอกทีฟ (Active Object) ในระบบที่ออกแบบด้วยวิธีเชิงวัตถุสามารถกำหนดให้มีวัตถุที่ทำงานแบบแอกทีฟ ทำงานไปพร้อมๆกันได้ โดยวัตถุที่ทำงานแบบแอกทีฟ ในระบบอาจจะเป็นอิสระต่อกัน หรืออาจจะต้องมีการสื่อสารกันได้

2.9.5 การคงสถานะ (Persistence)

การคงสถานะเป็นคุณสมบัติของวัตถุคือ วัตถุจะยังสามารถอยู่ได้ถึงแม้ว่าเวลาจะล่วงเลยไปแล้วก็ตาม และรวมถึงพื้นที่ของวัตถุที่อยู่ด้วย เช่น ถ้าตำแหน่งของวัตถุย้ายที่จากหน่วยความจำส่วนที่เคยอยู่เมื่อตอนถูกสร้างขึ้นมา วัตถุซอฟต์แวร์นั้นจะอยู่ในพื้นที่ของหน่วยความจำและในช่วงเวลาที่กำหนดเท่านั้น อย่างไรก็ตามมีภาษาในการเขียนโปรแกรมเชิงวัตถุเพียงไม่กี่ตัวเท่านั้นที่สนับสนุนคุณสมบัติการคงสถานะ ตัวอย่าง เช่น ภาษาจาวา

2.10 วัตถุและส่วนประกอบต่างๆ

การพัฒนาระบบแบบโครงสร้างนั้น จะมองระบบเป็นกลุ่มของข้อมูลและฟังก์ชัน โดยแบ่งฟังก์ชัน ออกเป็นฟังก์ชันย่อยๆที่มีการทำงานง่ายๆส่วนการพัฒนาระบบด้วยมุมมองของการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาระบบด้วยวิธีเชิงวัตถุนั้นจะต่างออกไป โดยวัตถุทั้งหมดนั้นจะมีสิ่งที่ผู้พัฒนาสนใจซึ่งประกอบไปด้วย แอททริบิวต์ (ข้อมูล) เมธอด (การกระทำ) สถานะ (หน่วยความจำ) สิ่งบ่งชี้ (Identity) ความรับผิดชอบ (Responsibility) ตัวอย่างวัตถุที่พบเห็นอยู่บ่อยๆ เช่น รถยนต์ ซึ่งวัตถุรถยนต์ก็มีแอททริบิวต์เช่น ความกว้าง ความยาว สี น้ำหนัก และก็มีเมธอด เช่น ออกตัว เบรก เลี้ยวซ้าย หรือเลี้ยวขวา สำหรับสถานะของรถยนต์ ก็เช่นกำลังออกตัว กำลังลดความเร็ว ส่วนสิ่งบ่งชี้ของรถยนต์คือทะเบียนรถ เป็นต้น และความรับผิดชอบของรถยนต์ก็คือนำพาผู้โดยสารไปให้ถึงที่หมายโดยปลอดภัย

2.10.1 แอททริบิวต์ (Attribute)

วัตถุจำเป็นต้องมีการเก็บข้อมูลของตัวเอง โดยข้อมูลที่วัตถุเก็บไว้นั้นอาจจะเก็บไว้เฉยๆเก็บไว้เพื่อที่จะให้ผู้อื่นเรียกไปใช้งาน เก็บไว้ใช้ในการประมวลผล เพื่อตอบสนองกับเหตุการณ์ภายนอก หรืออาจจะเก็บไว้เพื่อเป็นสถานะของตัววัตถุเอง สรุปว่าวัตถุโดยส่วนใหญ่จะต้องเก็บข้อมูล เพื่อให้วัตถุสามารถทำหน้าที่รับผิดชอบได้อย่างถูกต้อง

โดยทั่วไปแอททริบิวต์ของวัตถุจะถูกซ่อนไว้จากผู้ที่จะใช้งานวัตถุ ทั้งนี้เนื่องจากว่าการเข้าใช้งานแอททริบิวต์โดยตรงเปรียบเสมือนเป็นการเข้าถึงโครงสร้างของวัตถุ ดังนั้นการออกแบบวัตถุที่ดี นักออกแบบจึงควรที่จะซ่อนแอททริบิวต์ของวัตถุไว้เพื่อไม่ให้ผู้ใช้เห็น

บางครั้งการออกแบบวัคถุที่ไม่ดีก็อาจจะทำให้วัตถุนั้นมีแต่แอททริบิวต์โดยไม่มีเมธอดได้ ซึ่งในกรณีนี้นักออกแบบจะต้องย้อนกลับไปดูว่าวัตถุนั้นควรจะมีเมธอดอะไร เพราะวัตถุที่มีแต่แอททริบิวต์ นั้นก็ไม่ต่างจากโครงสร้างของข้อมูลเลย

2.10.2 เมธอดหรือพฤติกรรม

วัตถุที่ทำงานแบบพาสซิฟ (Passive Object) จะมีบริการไว้ให้วัตถุอื่นๆเรียกใช้ส่วนวัตถุที่ทำงานแบบแอกทีฟนั้นจะเป็นวัตถุหลักของเธรดซึ่งจะทำหน้าที่เรียกใช้บริการจากวัตถุที่ทำงานพาสซีฟ

เมธอดสามารถแบ่งออกเป็น 3 แบบคือ แบบไม่ซับซ้อน (Simple) , แบบอัตโนมัติ (Automation) และ แบบต่อเนื่อง (Continuous) เมธอดทั้ง 3 แบบนี้มีความสำคัญที่แตกต่างกันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบไม่ซับซ้อน วัตถุจะให้บริการกับการร้องขอเข้ามาและไม่ค้างค่าในหน่วยความจำไว้สำหรับให้บริการครั้งต่อไป การกระทำเป็นหนึ่งเดียวและสมบูรณ์ในตัว วัตถุพื้นฐานจะมีข้อมูลชนิดพื้นฐานและโอเปอเรชันที่เกี่ยวข้องกับข้อมูลเหล่านั้น

แบบอัตโนมัติ หรือ Finite State Machine (FSM) วัตถุชนิดนี้จะมีกลุ่มของสถานะที่มีขอบเขตวัตถุหนึ่งอาจจะมีสถานะ 1 หรือมากกว่าในเวลาใดๆ การที่วัตถุจะอยู่ในสถานะใดๆนั้น ก็จะขึ้นอยู่กับอีเวนต์ (Event) หรือเหตุการณ์ที่เข้ามาจากสภาวะแวดล้อมภายนอก และเนื่องจากว่าวัตถุแบบนี้ทำงานแบบ State Machine เพื่อตอบสนองกับอีเวนต์ ภายนอกจึงเรียกวัตถุแบบนี้ว่าวัตถุแบบรีแอกทีฟ (Reactive Object)

แบบต่อเนื่อง วัตถุจะมีกลุ่มของสถานะที่ไม่จำกัดและไม่มีขอบเขต การทำงานของวัตถุปัจจุบันจะขึ้นอยู่กับการทำงานของเมธอดและอินพุทก่อนหน้า ซึ่งความเกี่ยวข้องระหว่างสถานะนี้อาจจะเป็นความสัมพันธ์แบบต่อเนื่อง

2.10.3 เมสเซส (Message)

การสื่อสารระหว่างวัตถุสามารถทำได้โดยการส่งแมสเซเป็นการแยกแยะเอกลักษณ์ของข้อมูล (Data) หรือข้อมูลสารสนเทศ (Information) การควบคุมที่ส่งมา จากวัตถุหนึ่งไปยังวัตถุอื่นๆในเชิงการเขียนโปรแกรมแมสเซจ สามารถทำได้หลายแบบ ตัวอย่างเช่น

- การเรียกฟังก์ชัน
- เมล์บน Real time Operating System (RTOS)
- อีเวนต์ บน RTOS
- อินเทอร์รัพท์ (Interrupt)
- การแชร์ทรัพยากรโดยใช้ Semaphore – Protection
- Rendezvous ของ Ada
- Remote Procedure Call (RPC) ในระบบแบบกระจาย

ในขั้นตอนการวิเคราะห์ระบบจะมีการกำหนดแมสเซจส่วนในขั้นตอนของการออกแบบนั้นจึงจะมากำหนดรูปแบบการซิงโครไนเซชันและความต้องการทางด้านเวลาของแต่ละแมสเซจ และเมื่อวัตถุได้รับ แมสเซจมา วัตถุก็จะเปลี่ยนแมสเซจที่เข้าให้เป็นโอเปอเรชันการเปลี่ยนสถานะ คำสั่งหรือข้อมูลตามที่เหมาะสม

การใช้แมสเซจทำให้แต่ละวัตถุมีความเกี่ยวข้องกันน้อยลงในขั้นตอนของการวิเคราะห์ผู้พัฒนาระบบจะไม่ได้กำหนดรายละเอียดในการติดต่อของแมสเซจว่า จะมีราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละเอียดของการเรียกฟังก์ชัน Rendezvous ไทม์เอาท์ (Time Out) และเมื่อถึงขั้นตอนของการออกแบบจึงจะจัดการกับรายละเอียดเหล่านี้

ส่วนอินเทอร์เฟซ ของวัตถุเป็นส่วนที่วัตถุใช้ติดต่อกับโลกภายนอก ซึ่งอินเทอร์เฟซจะทำหน้าที่กำหนดเซ็ตของโพรโตคอล (Protocol) เพื่อสื่อสารกับวัตถุอื่นๆ โพรโตคอลของส่วนอินเทอร์เฟซนั้นประกอบไปด้วย 3 สิ่งดังนี้คือ

- เงื่อนไขก่อนหน้า (Precodition)
- ซิกเนเจอร์ (Signature)
- เงื่อนไขท้าย (Postcondition)

เงื่อนไขก่อนหน้าเป็นเงื่อนไขที่ต้องมีค่าเป็นจริงก่อนที่จะส่งหรือรับ แมสเซจโดยปกติเงื่อนไขก่อนหน้า นั้นจะเป็นเงื่อนไขที่ต้องมีค่าเป็นจริงหลังจากที่วัตถุได้ประมวลผลแมสเซจ เสร็จเรียบร้อยแล้ว และเป็นหน้าที่ของวัตถุที่รับ แมสเซจ สำหรับแมสเซจซิกเนเจอร์นั้นเป็นรูปแบบในการส่งแมสเซจ ซึ่งอาจจะเป็นการเรียกฟังก์ชัน พร้อมพารามิเตอร์และค่าที่คืนกลับมา (Returm Type) หรือเป็นแมสเซจ post/pend ของ RTOS หรือข้อตกลงของแมสเซจ กับ บัส (Bus)

2.10.4 ความรับผิดชอบ (Responsibility)

คือหน้าที่และบทบาทของวัตถุต่อระบบ โดยส่วนอินเทอร์เฟซ และเมธอดของวัตถุนั้นสามารถใช้เพื่อให้วัตถุทำหน้าที่รับผิดชอบได้ ความรับผิดชอบของวัตถุควรจะเป็นสิ่งแรกที่ผู้พัฒนาระบบกำหนดให้กับวัตถุ เพราะถ้าไม่รู้ว่าวัตถุนั้นมีหน้าที่อย่างไร ก็จะไม่สามารถกำหนดคุณลักษณะอื่นๆให้กับวัตถุได้อย่างถูกต้องและครบถ้วน

2.11 ความสัมพันธ์ระหว่างวัตถุ

โดยปกตินั้นวัตถุเพียงวัตถุเดียวไม่สามารถที่จะรับผิดชอบงานทั้งหมดของระบบได้วัตถุจะต้องมีความสัมพันธ์และต้องทำงานร่วมกันเพื่อให้ระบบสามารถทำหน้าที่รับผิดชอบได้ โดยเราแบ่งความสัมพันธ์ระหว่างวัตถุออกเป็น 2 แบบคือ ลิงค์ และการเป็นส่วนหนึ่งของ(Aggregation)

2.11.1 ลิงค์ (Link)

เป็นการเชื่อมต่อระหว่างวัตถุในทางกายภาพหรือในทางตรรก การที่วัตถุซึ่งมาประกอบกันเป็นระบบจะทำงานร่วมกันได้นั้น วัตถุเหล่านี้จะต้องการสื่อสารระหว่างกัน วัตถุจะสื่อสารระหว่างกันโดยวัตถุที่เรียกใช้บริการ (Service) จากวัตถุอื่นเรียกว่า วัตถุไคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็นต์ (Client) ส่วนวัตถุที่มีบริการให้วัตถุอื่นเรียกนั้นจะเรียกว่า วัตถุเซิรฟ์เวอร์ (Server) หรือ Supplier

2.11.2 ความสัมพันธ์แบบเป็นส่วนหนึ่งของ (Aggregation)

เป็นความสัมพันธ์ที่วัตถุหนึ่งเป็นส่วนหนึ่งของอีกวัตถุ โดยอาจจะเป็นความสัมพันธ์แบบเป็นส่วนหนึ่งของธรรมดาหรือคอมโพซิชัน(Composition)โดยคลาสที่ประกอบไปด้วยคลาสอื่นจะไม่สามารถแชร์ความเป็นเจ้าของได้ ส่วนคลาสที่เป็นเจ้าของก็จะมีหน้าที่สร้างและทำลายวัตถุของคลาสที่ประกอบอยู่ภายในด้วย ในการวาดโมเดลจะเขียนคลาสที่เป็นส่วนประกอบของคลาสอื่นให้อยู่ในคลาสที่เป็นคลาสเจ้าของ หรือสามารถแทนความสัมพันธ์ของคอมโพซิชันโดยใช้สัญลักษณ์เดียวกับความสัมพันธ์แบบเป็นส่วนหนึ่งของคือใช้เส้นและหัวลูกศรที่เป็นสี่เหลี่ยมระบายสีทึบ

2.12 คลาส

คลาสคือ การแยกแยะเอกลักษณ์ของคุณสมบัติพื้นฐานจากกลุ่มของวัตถุที่เหมือนกัน เช่น คลาสของสัตว์เลี้ยงลูกด้วยนม เช่น แมว หนู หมี จะมีคุณสมบัติพื้นฐานใช้ร่วมกันอยู่โดยค่าของคุณสมบัติก็จะแตกต่างกันไปแล้วแต่อินสแทนซ์

คลาสจะกำหนดแอททริบิวต์และเมธอดของวัตถุ ซึ่งเป็นอินสแทนซ์ แต่ว่าคลาสไม่มีความรับผิดชอบ คุณสมบัติทั้งหมดของคลาสอยู่ในรูปของชนิด ไม่ใช่ค่า (Value) คลาสใน UML นั้นจะเขียนเป็นรูปสี่เหลี่ยม โดยมีชื่อของคลาสอยู่ด้านใน โดยอาจจะแบ่งสี่เหลี่ยมออกเป็น 3 ส่วน โดยส่วนบนสุดเป็นชื่อของคลาส ส่วนตรงกลางใช้แสดงรายการของแอททริบิวต์ และส่วนล่างใช้แสดงรายการของโอเปอเรชัน สำหรับ แอททริบิวต์ และโอเปอเรชัน นั้นอาจจะไม่ต้องแสดงไว้ก็ได้

2.13 ความสัมพันธ์ระหว่างคลาส

ความสัมพันธ์ระหว่างคลาสแบบออกเป็น 3 แบบใหญ่ๆคือ ความสัมพันธ์ในการสืบทอดคุณสมบัติหรือ Generalization/Specialization ซึ่งเป็นความสัมพันธ์แบบเป็นหรือ " is a " ตัวอย่าง เช่น ดอกกุหลาบเป็นดอกไม้ หมายความว่า ดอกกุหลาบเป็นชนิดหนึ่งที่เฉพาะของดอกไม้ ส่วนความสัมพันธ์ระหว่างคลาสแบบที่สองคือ ความสัมพันธ์แบบเป็นส่วนหนึ่งหรือ " whole/pare " หรือ " part of " เช่น เกสรดอกไม้ไม่ได้เป็นชนิดหนึ่งของดอกไม้ แต่ว่าเกสรดอกไม้เป็นส่วนหนึ่งของดอกไม้ ส่วนความสัมพันธ์ที่สามคือ แอซโซซิเอต (Associate) ซึ่งเป็นความสัมพันธ์ของคลาสที่บ่งบอกถึงความเกี่ยวเนื่องของคลาสที่ไม่มีความหมายเหมือนกันเลย ตัวอย่างเช่น ดอกกุหลาบกับผึ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งไม่ได้มีความหมายเหมือนกันเลย แต่ ดอกกุหลาบกับผึ้งมีความเกี่ยวข้องกัน คือ แมลงช่วยผสมเกสรดอกไม้ให้ดอกกุหลาบ และดอกกุหลาบก็เป็นแหล่งอาหารให้กับผึ้ง เป็นต้น

2.13.1 ความสัมพันธ์แบบแอซโซซิเอชัน (Association)

เมื่อวัตถุหนึ่งใช้บริการของอีกวัตถุแต่ว่าไม่ได้เป็นเจ้าของวัตถุนั้น จะเรียกความสัมพันธ์นี้ว่าแอสโซซิเอชันซึ่งความสัมพันธ์แบบนี้จะใช้ได้อย่างเหมาะสมก็ต่อเมื่อ

- วัตถุใช้บริการของอีกวัตถุหนึ่ง แต่ไม่ได้มีความสัมพันธ์แบบเป็นส่วนหนึ่งของ
- วงจรชีวิตของคลาสที่ใช้ไม่ได้เป็นหน้าที่ความรับผิดชอบของคลาสที่เรียกใช้ทั้งการสร้างและการทำลายวัตถุ
- ความสัมพันธ์ระหว่างวัตถุนั้นน้อยกว่าความสัมพันธ์แบบเป็นส่วนหนึ่งของ
- ความสัมพันธ์เป็นคุณลักษณะของไคลเอ็นต์-เซิร์ฟเวอร์ (Client – Server)
- วัตถุมีการถูกเรียกใช้บริการจากหลายๆวัตถุ และถูกใช้มากเท่าๆกับที่วัตถุอื่นๆเรียกใช้

2.13.2 ความสัมพันธ์ในการสืบทอดคุณสมบัติ (Inheritance)

เมื่อคลาสหนึ่งเป็นความเฉพาะของอีกคลาสหนึ่งจะใช้ความสัมพันธ์ที่เรียกว่า การสืบทอดคุณสมบัติหรือ Generalization หรือ Inheritance หมายความว่า คลาสลูกจะมีคุณสมบัติทุกอย่างที่คลาสพ่อแม่มี ถึงแม้ว่าคลาสลูกจะมีความเฉพาะมากกว่าก็ตาม คลาสลูกอาจจะเพิ่มคุณสมบัติของคลาสพ่อแม่ โดยการเพิ่มแอททริบิวต์ และเมธอดเข้าไป เรียกความสัมพันธ์นี้ว่า "เป็นชนิดหนึ่งของ" เช่น สัตว์เลี้ยงลูกด้วยนมเป็นชนิดหนึ่งของสัตว์

2.13.3 ความสัมพันธ์แบบเป็นส่วนหนึ่งของ (Aggregation)

ความสัมพันธ์แบบเป็นส่วนหนึ่งของนั้น จะใช้ก็ต่อเมื่อมีวัตถุหนึ่งบรรจุอีกวัตถุหนึ่งไว้ทั้งในทางตรรกและทางกายภาพ คลาสที่ใหญ่กว่าเรียกว่าเจ้าของหรือ Owner หรือ Whole ซึ่งเป็นคลาสที่มีหัวลูกศรรูปสี่เหลี่ยมอยู่ ส่วนคลาสที่เล็กกว่านั้นเรียกว่า คลาสส่วนประกอบหรือคลาสคอมโพเนนท์ หรือ Owned หรือ Part โดยทั่วไปคลาสเจ้าของมีหน้าที่สร้างและทำลายคลาสคอมโพเนนท์ใน UML ยอมให้คลาสส่วนประกอบสามารถมีคลาสเจ้าของได้หลายคลาส และเมื่อวัตถุมีหลายเจ้าของ ผู้พัฒนาก็จะพิจารณาว่าคลาสใดจะทำหน้าที่สร้างและทำลายคลาสคอมโพเนนท์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14 แบบจำลอง Unified Modeling Language

Unified Modeling Language หรือ UML เป็นหนึ่งในเครื่องมือที่ใช้ในกระบวนการพัฒนาและสร้างระบบใดๆขึ้นมา โดยที่ UML ช่วยให้ผู้พัฒนาระบบสามารุที่จะสร้างแบบจำลองให้มีความเป็นมาตราฐาน เข้าใจได้ง่ายและสามารถที่จะสื่อสารกับบุคคลอื่นได้

แบบจำลอง UML ประกอบไปด้วยองค์ประกอบย่อยที่เป็นรูปภาพแล้วนำมารวมกันตามกฎเกณฑ์ที่กำหนดไว้ เพื่อสร้างแผนภาพที่ใช้ในการอธิบายถึงส่วนประกอบของระบบที่ทำการพัฒนา มีจุดประสงค์เพื่อที่จะทำการนำเสนอมุมมองในหลายๆมุมมองของระบบที่ทำการพัฒนานั้นมีลักษณะและรูปแบบการทำงานเป็นอย่างไร ไม่ได้บอกถึงขั้นตอนในการสร้างระบบว่าทำอย่างไร และไม่จำเป็นที่การออกแบบและพัฒนาระบบทุกระบบจะต้องมีแผนภาพทุกชนิดที่มีอยู่ในแบบจำลอง UML นี้

แบบจำลอง UML ประกอบไปด้วยแผนภาพหลายชนิด แต่แผนภาพโดยทั่วไปที่นิยมใช้ในการแสดงและอธิบายโครงสร้างของระบบที่ทำการพัฒนานั้นมีดังนี้

2.14.1 แผนภาพของคลาส (Class Diagram)

แผนภาพของคลาสเป็นแผนภาพที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบโดยใช้วิธีเชิงวัตถุ เนื่องจากแผนภาพของคลาสจะแสดงโครงสร้างของวัตถุและคลาสที่มีในระบบรวมทั้งแสดงความสัมพันธ์ด้วย แผนภาพของคลาสจะเป็นโครงสร้างหลักของระบบและยังใช้ในการแยกย่อยรายละเอียด (Decomposite) ด้วยวิธีเชิงวัตถุ อย่างไรก็ดี แผนภาพของคลาสนั้นอาจจะมีหน้าตาคล้ายกับแผนภาพของวัตถุแต่ว่าแผนภาพของคลาสจะแสดงเฉพาะคลาส โดยไม่ได้แสดงอินสแทนซ์หรือวัตถุทั้งหมด และจะเน้นที่การแสดงโครงสร้างของคลาสมากกว่าความสัมพันธ์ระหว่างคลาส

2.14.2 แผนภาพของวัตถุ (Object Diagram)

เป็นแผนภาพที่ใช้แสดงอินสแทนซ์ของคลาส โดยทำการกำหนดค่าให้กับแอททริบิวต์

2.14.3 แผนภาพของการติดต่อระหว่างระบบกับผู้กระทำภายนอก (Use Case Diagram)

เป็นแผนภาพที่แสดงการติดต่อกันระหว่างระบบกับตัวผู้กระทำภายนอกระบบ แผนภาพยูสเคสนั้นจะแสดงมุมมองต่อฟังก์ชันการทำงานหลักๆของระบบซึ่งผู้พัฒนาระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถใช้ในเนื่องจากตัวแผนภาพยูสเคสนั้น ได้รวมความต้องการและรายละเอียดการทำงานหรือกรณีที่เป็นไปได้หรือ ซีนาริโอ ของระบบไว้ภายในแล้ว สำหรับตัวยูสเคสเองนั้นก็สามารถที่จะแบ่งเป็นยูสเคสย่อยๆลงไปได้อีก หรืออาจจะใช้ในการสร้างซีนาริโอจะแสดงรายละเอียดลำดับในการติดต่อสื่อสารระหว่างวัตถุในระบบ เนื่องจากแผนภาพยูสเคสนั้นเป็นมุมมองต่อระบบในระดับสูง และมีการใช้ภาษาอธิบายระบบทั่วๆไป ทำให้ผู้ใช้ในทุกๆระดับสามารถตรวจสอบแผนภาพยูสเคสได้ง่ายขึ้น และสามารถใช้ในการสื่อสารกันระหว่างทีมผู้พัฒนาเองได้ด้วย เนื่องจากสัญลักษณ์ต่างๆที่ใช้ในแผนภาพยูสเคสนี้เป็นมาตรฐานที่ทุกคนรู้จัก และแผนภาพก็มีรายละเอียดของระบบที่ครบถ้วน

แผนภาพยูสเคสในระดับบนสุดนั้นจะแสดงภาพรวมของระบบทั้งหมดซึ่งเป็นมุมมองจากฟังก์ชันการทำงานไม่ใช่มุมมองต่อคลาส และวัตถุที่มีในระบบ โดยฟังก์ชันการทำงานของระบบจะแทนด้วยยูสเคสและจะแสดงความสัมพันธ์กับวัตถุที่อยู่ภายนอกระบบโดยผ่านทางโพรโตคอลของระบบ

สำหรับการสร้างยูสเคส ขึ้นมานั้นโดยส่วนใหญ่จะใช้วิธีกำหนด ซีนาริขึ้นมาก่อนจากนั้นก็จะทำการเลือกซีนาริโอที่สำคัญต่อระบบขึ้นมาเป็นยูสเคสซึ่งก็มีหลายๆวิธีในการค้นหารายละเอียดของระบบเพื่อนำมาใช้ในการกำหนดซีนาริโอจากนั้นก็จะทำการสร้าง ยูสเคสขึ้นมาจากบทบาทและการติดต่อสื่อสารระหว่างวัตถุภายนอกต่อระบบ ลำดับการไหลเวียนของเหตุการณ์และข้อมูลในซีนาริโอและต้องคำนึงถึงการเปลี่ยนแปลงที่อาจเกิดขึ้นกับซีนาริโอด้วย

ผู้พัฒนาระบบจะใช้ยูสเคสเป็นเครื่องมือหลักในการพัฒนาโครงการซอฟต์แวร์ทั้งนี้เนื่องจากสามารถใช้ยูสเคสสื่อสารกับทั้งลูกค้าและทีมพัฒนาได้ นอกจากนั้นยูสเคสยังได้รวมฟังก์ชัน การทำงานและ ซีนาริโอของระบบไว้ด้วย จึงทำให้ยูสเคสนั้นสามารถใช้ได้ตลอดกระบวนการพัฒนาซอฟต์แวร์

2.14.4 แผนภาพของการแสดงเปลี่ยนสถานะของแต่ละวัตถุ (State Diagram)

เป็นแผนภาพที่ใช้แสดงการเปลี่ยนสถานะของแต่ละวัตถุ ที่มีอยู่ในระบบ โดยจะแสดงลำดับการเปลี่ยนสถานะและแมสเซจที่ใช้ในการเปลี่ยนสถานะของวัตถุ เพื่อแสดงให้เห็นถึงการทำงานของแต่ละวัตถุ

2.14.5 แผนภาพของการแสดงซีนาริโอของระบบ (Sequence Diagram)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นแผนภาพที่ใช้แสดงซีนาริโอของระบบโดยจะแสดงลำดับของแมสเซจระหว่างวัตถุที่ไหลเวียนอยู่ในระบบโดยจะแสดงเป็นแต่ละเหตุการณ์

2.14.6 แผนภาพของการแสดงลำดับของกิจกรรมทั้งหมด (Activity Diagram)

เป็นแผนภาพที่ใช้แสดงลำดับของกิจกรรมที่จัดทำทั้งหมด ที่เกิดขึ้นตามยูสเคสหรือเกิดจากกิจกรรมของวัตถุเองตามปกติ

2.14.7 แผนภาพของการแสดงโครงสร้างของวัตถุที่ทำงานร่วมกัน (Collaboration Diagram)

เป็นแผนภาพที่ใช้แสดงซีนาริโออีกตัวหนึ่งที่มีประโยชน์โดยจะแสดงข้อมูลพื้นฐานเช่นเดียวกับแผนภาพแสดงลำดับขั้นตอนการทำงาน (Sequence Diagram) แต่มีความแตกต่างกันคือ แผนภาพแสดงลำดับขั้นตอนการทำงานนั้นจะแสดงลำดับของแมสเซจเป็นหลัก ส่วนแผนภาพแสดงการทำงานร่วมกันของวัตถุนั้นจะแสดงโครงสร้างของวัตถุที่ทำงานร่วมกันเป็นหลัก

2.14.8 แผนภาพของแบบจำลองทางกายภาพของระบบ (Component Diagram)

เป็นแผนภาพที่ใช้จำลองลักษณะทางกายภาพของระบบเชิงวัตถุ โดยจะแสดงให้เห็นถึงส่วนประกอบทางซอฟต์แวร์ต่างๆของระบบ รวมถึงความสัมพันธ์ระหว่างส่วนประกอบต่างๆด้วย

2.14.9 แผนภาพของการแสดงโครงสร้างของซอฟต์แวร์และฮาร์ดแวร์ (Deployment Diagram)

เป็นแผนภาพที่ใช้ในการแทนระบบในระดับสถาปัตยกรรมคือจะแสดงเกี่ยวกับโครงสร้างของซอฟต์แวร์และฮาร์ดแวร์ที่มีความสัมพันธ์กัน

2.15 ระบบการจัดการฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Management System)

ระบบจัดการฐานข้อมูลเชิงวัตถุ (OODBMS) บางครั้งเรียกสั้นๆได้ว่า ODBMS มาจาก Object Database management system เป็นระบบจัดการฐานข้อมูลที่สนับสนุนการออกแบบและการสร้างข้อมูลในรูปของวัตถุ ซึ่งประกอบชนิดของการสนับสนุนจาก คลาสของวัตถุและลักษณะที่ถ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทอดของคลาส(Inheritance Class) , คุณสมบัติ (Properties) และ กระบวนการ (Method) ของคลาสย่อยในวัตถุ เหล่านั้น

ในปัจจุบันยังไม่มีข้อตกลงเกี่ยวกับมาตรฐานในการจัดทำ OODBMS และผลิตภัณฑ์ที่มีลักษณะของ OODBMS ยังอยู่ในระยะเริ่มต้น ในขณะเดียวกันระบบจัดการฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Management Systems : ORDBMS) มีแนวคิดที่ว่าสามารถเสริมเพิ่มเติมลงไปบนฐานข้อมูลเชิงความสัมพันธ์ ซึ่งถูกพบเห็นได้ในผลิตภัณฑ์ทั่วๆไป มาตรฐานของการเชื่อมต่อของฐานข้อมูลเชิงวัตถุ ได้ถูกพัฒนาโดยกลุ่มองค์กรหนึ่ง ภายใต้ชื่อว่า The Object Data Management Group (ODMG) ซึ่งได้ทำการกำหนดมาตรฐานของการเชื่อมต่อข้อมูลเชิงวัตถุระหว่างระบบในเครือข่ายเน็ตเวิร์ค (Network)

ในรายงานที่มีชื่อเสียงเรื่องหนึ่ง ที่มีชื่อว่า The Object-Oriented Database Manifesto, Malcolm Atkinson and others define an OODBMS ได้กล่าวไว้ว่า

Object-Oriented Database Systems ต้องเป็นไปตามหลักการ 2 ข้อคือ

1 จะต้องเป็นระบบจัดการฐานข้อมูล (DBMS)

2 จะต้องเป็นระบบที่มีความสัมพันธ์เชิงวัตถุ (Object-Oriented System)

ตัวอย่างเช่น จะต้องยึดหลักของการเขียนโปรแกรมเชิงวัตถุ(Object-Oriented Programming Language) โดยหลักการข้อแรก จะเปลี่ยนไปเป็นคุณสมบัติ 5 ข้อคือ persistence , secondary storage management , concurrency , recovery และ ad hoc query facility ส่วนในหลักการข้อสอง จะเปลี่ยนไปเป็นคุณสมบัติ 8 ข้อคือ complex object , object identity , encapsulation , types or classes , inheritance , overriding combined with late binding , extensibility และ computational completeness

2.15.1 คำจำกัดความของระบบการจัดการฐานข้อมูลเชิงวัตถุ (Object-Oriented Database Management System Definition)

เมื่อทำการรวมสมรรถภาพของฐานข้อมูลเข้ากับความสามารถของการเขียนโปรแกรมเชิงวัตถุ ผลลัพธ์ที่ได้คือ ODBMS ซึ่งเป็นการทำให้ฐานข้อมูลมีลักษณะเป็นข้อมูลเชิงวัตถุ การเขียนโปรแกรมเชิงวัตถุ ที่มีการใช้งานอยู่ทั่วๆไป ODBMS เป็นการขยายภาษา (Language) โดยใช้หลักการของ ข้อมูลที่ต่อเนื่องอย่างชัดเจน (Transparently Persistent Data) , การควบคุมที่สอดคล้อง (Concurrency Control) , การคืนข้อมูลขึ้นมาใหม่ (Data Recovery) ,การเรียกดูข้อมูลที่สอดคล้องกัน (Associative queries) และความสามารถอื่นๆของฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการจัดการฐานข้อมูลเชิงวัตถุสามารถเขียนชื่อย่อในรูปแบบอื่นได้หลากหลาย ยกตัวอย่างเช่น OODBMS , ODB , OODB , และ OODMS ซึ่งทั้งหมดนี้ สามารถเรียกได้ว่าเป็น Object-Oriented Database Systems

2.15.2 Transparent Persistence

Transparent Persistence ในฐานข้อมูลเชิงวัตถุ เป็นการกล่าวถึงความสามารถโดยตรงการจัดการกับข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลที่มีการใช้หลักการของการเขียนโปรแกรมเชิงวัตถุ ซึ่งมีความแตกต่างกับฐานข้อมูล Sub Language ที่มีการใช้ embedded SQL หรือการเชื่อมต่อโดยใช้ ODBC หรือ JDBC ซึ่งหมายความว่าการใช้งานฐานข้อมูลเชิงวัตถุจะทำให้เกิดประสิทธิภาพมากขึ้น และ จำนวนโค้ด ที่ใช้ก็น้อยลง

ใน Transparent persistence การจัดการและการตรวจสอบ (Traversal) ของ persistent object เป็นการกระทำโดยตรงโดยการ หลักการของการเขียนโปรแกรมเชิงวัตถุในลักษณะเดียวกันกับที่ใช้ในหน่วยความจำ (Memory) ซึ่งก็คือ non-persistent ซึ่งเป็นการทำให้สำเร็จโดยการใช้ intelligent caching

2.15.3 Caching

คือการกักเก็บข้อมูล โดยส่วนมากในโปรแกรมการใช้งาน เช่น โปรแกรมที่ใช้ดูความหนาแน่นของข้อมูลทางเครือข่ายหรือในการเข้าถึงข้อมูล ด้วย Transparent Persistence Caching จะทำการติดตั้งลงไปในบางส่วนของเนื้อที่ใน โปรแกรมการใช้งานด้วยแคส (Cache) ทำให้ไม่ต้องเคลื่อน วัตถุจากดิสต์ (Disk) โดยตรง แต่สามารถเคลื่อนแบบอัตโนมัติจากที่เก็บข้อมูล (Disk Storage) ไปยัง หน่วยความจำของโปรแกรม (Program Memory)

การแคสชิง (Caching) ใน ODBMS มีความคล้ายคลึงกับ แคสชิงของ Object-Relational Mapping แต่ปัญหานี้จะไม่เกิดขึ้นกับ ODBMS เพราะว่าแคสจะถูกรวมเข้าไปอยู่ใน ODBMS เซิร์ฟเวอร์

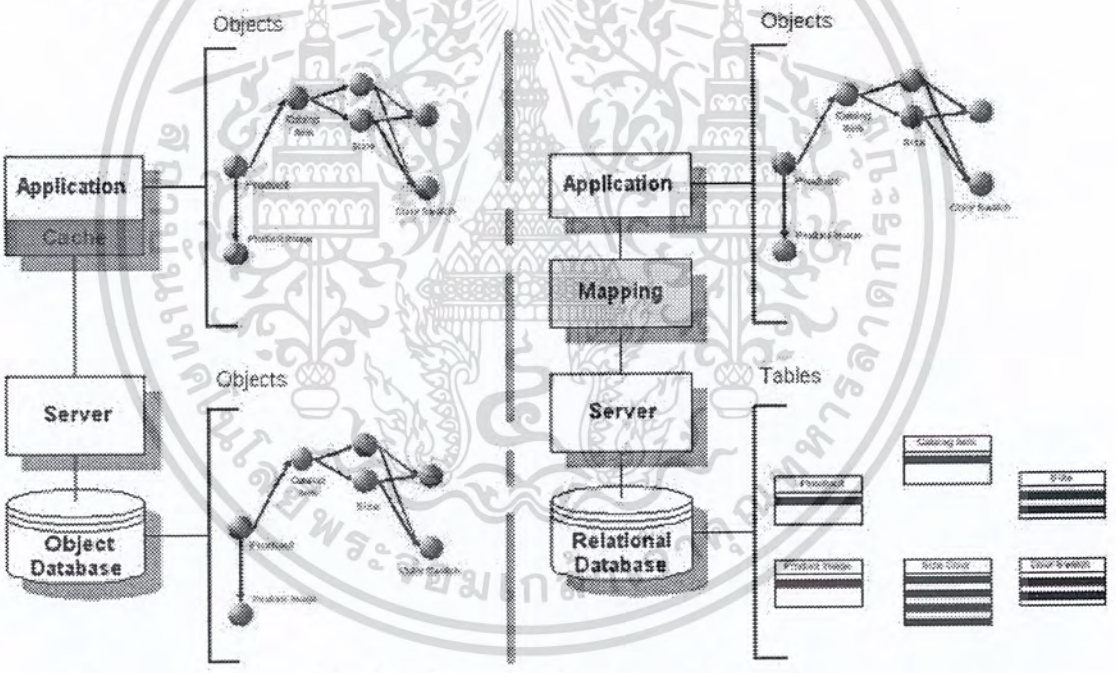
ODBMS จะเกิดประสิทธาภาพสูงและค่าใช้จ่ายที่ใช้ในการพัฒนาถูก เมื่อ ใน วัตถุ เพราะว่า จะเก็บข้อมูลวัตถุ บนดิสต์ และมี Transparent Program Integration โดยใช้หลักการของการเขียนโปรแกรมเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพจะเพิ่มขึ้นเมื่อทำการจัดเก็บวัตถุ บนดิสต์ โดยตรง เพราะปราศจาก Impedance Mismatch ส่วนค่าใช้จ่ายที่ใช้การพัฒนาจะลดลงเพราะว่าไม่ต้องการโปรแกรมที่ใช้ในการจัดการเกี่ยวกับการแคสชิงของโปรแกรม

2.15.4 Lack of Impedance mismatch

ODBMSอนุญาตในเราทำการเก็บข้อมูลได้โดยตรงโดยปราศจากการแมปปิ่ง (Mapping)ของข้อมูลที่มีโครงสร้างที่แตกต่างกันระบบการจัดการฐานข้อมูลเชิงความสัมพันธ์ต้องการ การแมปปิ่ง จากข้อมูล ไปเป็นตาราง (Table) ซึ่งเป็นการการแมปปิ่งของข้อมูลที่มีโครงสร้างแตกต่างกันที่เรียกว่า Impedance Mismatch โดยรูปด้านล่างแสดงการเปรียบเทียบถึงการเก็บข้อมูลโดยตรงทางด้านซ้ายกับการเกิด Impedance Mismatch ทางด้านขวา



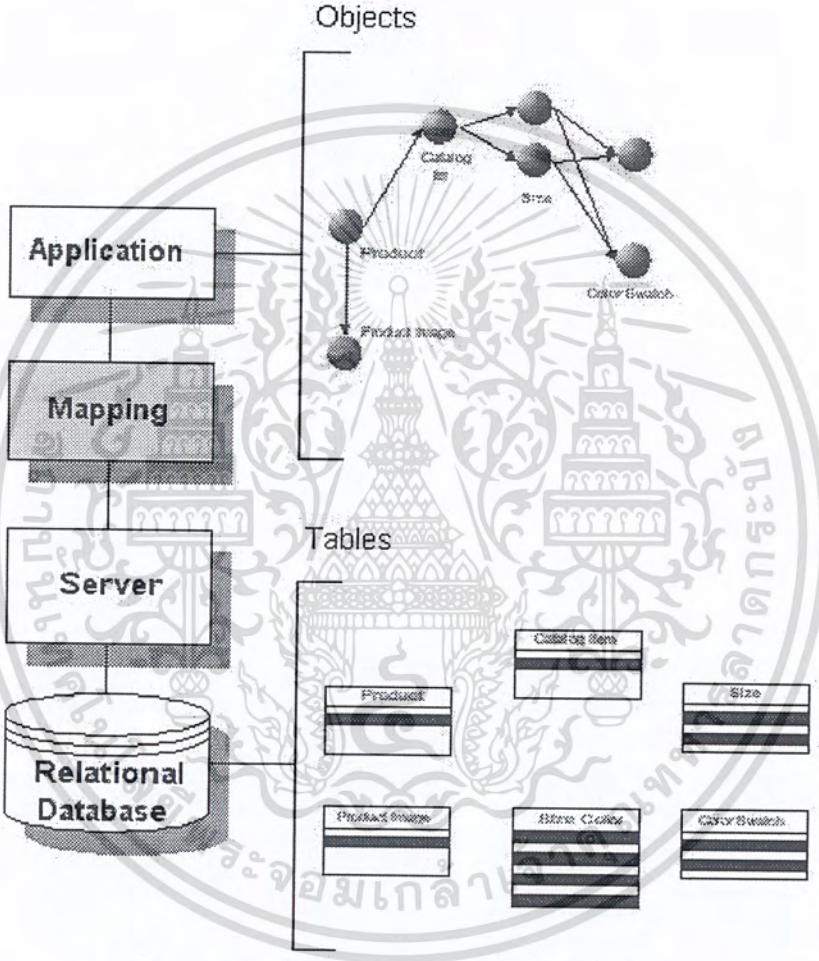
รูปที่ 2-15 ก. แสดงการเปรียบเทียบถึงการเก็บข้อมูลโดยตรงกับการเกิด Impedance Mismatch

การไม่มี Impedance Mismatch ใน Object DBMS ทำให้ได้เปรียบกว่าระบบการจัดการฐานข้อมูลเชิงความสัมพันธ์โดยเฉพาะในฐานข้อมูลที่ซับซ้อน Impedance Mismatch จะทำให้ประสิทธิภาพลดลง เนื่องจากมีขั้นตอนการเปลี่ยน (Map)โครงสร้างจากโครงสร้างข้อมูลหนึ่ง จากตาราง ไปอีกโครงสร้างหนึ่ง วัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15.5 Impedance Mismatch

เกิดจากการที่เราต้องทำการเปลี่ยนวัตถุ ไปเป็นตาราง เพื่อทำการเก็บลงในฐานข้อมูลเชิงวัตถุ โดยรูปด้านล่างแสดงตัวอย่าง การแมปปิ่งวัตถุ เป็น ตาราง และประสิทธิภาพที่ลดลงเมื่อใช้กับข้อมูลที่มีความซับซ้อน



รูปที่ 2-15 ข. แสดงรูปการแมปปิ่งจากวัตถุเป็นตาราง

2.15.6 Complex Data ข้อมูลที่มีความซับซ้อน

จะต้องมีคุณสมบัติดังนี้

1 ไม่เป็น unique หรือ natural identification การบ่งชี้ที่เป็นธรรมชาติ

2 มีความสัมพันธ์แบบ many to many เป็นจำนวนมาก

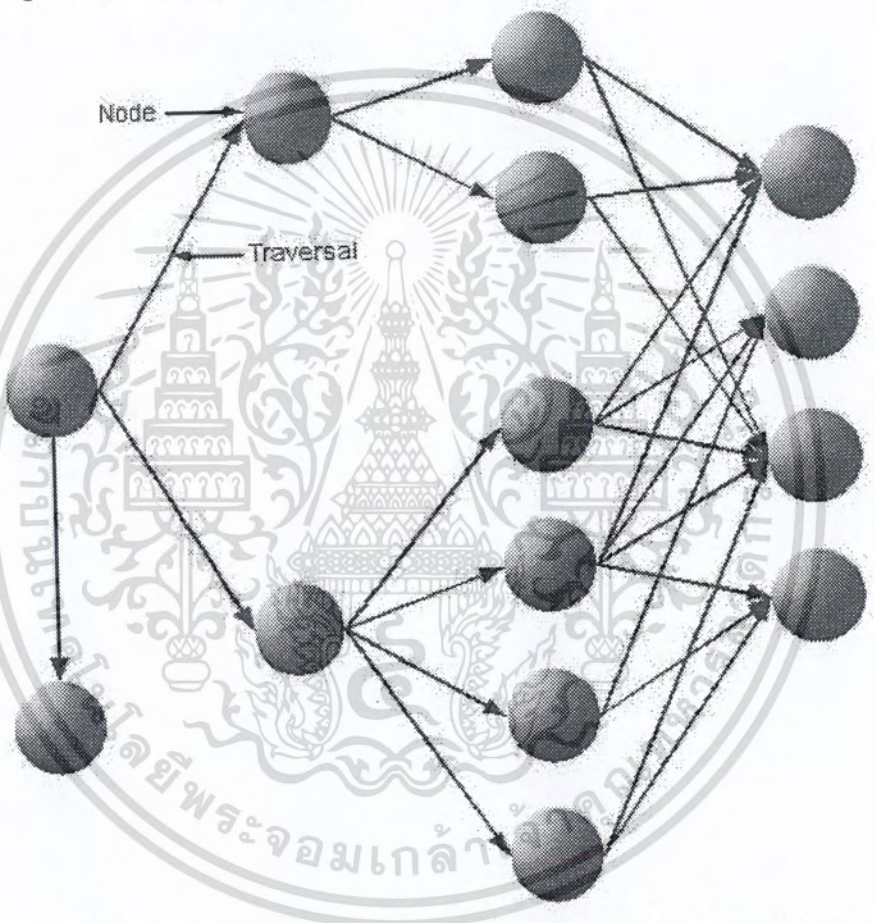
3 การเข้าถึงต้องใช้การตรวจสอบ (Traversals)

4 ใช้ชนิดของโค้ดที่เกิดขึ้นเสมอ เหมือนที่พบใน Relation Schema

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15.7 Navigation with an object database

โดยทั่วไปแล้วการเข้าถึงข้อมูลเชิงวัตถุ จะเรียกว่าการ Navigation หรือที่รู้จักกันในอีกชื่อหนึ่งว่า Traversal ซึ่งเป็นคำศัพท์ที่ได้มาจากการเข้าถึงรูปแบบของโครงสร้างข้อมูลแบบทรี (Tree) หรือ กราฟ (Graph) ถ้าทำการวาดโครงสร้างโดยดูจากรูปทางด้านล่าง จะเห็นได้ว่าการเคลื่อนที่จากโหนดหนึ่งไปอีกโหนดหนึ่งภายในโครงสร้างนั่นก็คือ Navigation หรือ Traversing นั่นเอง



รูปที่ 2-15 ค. แสดงโครงสร้างของทรี และลักษณะของการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

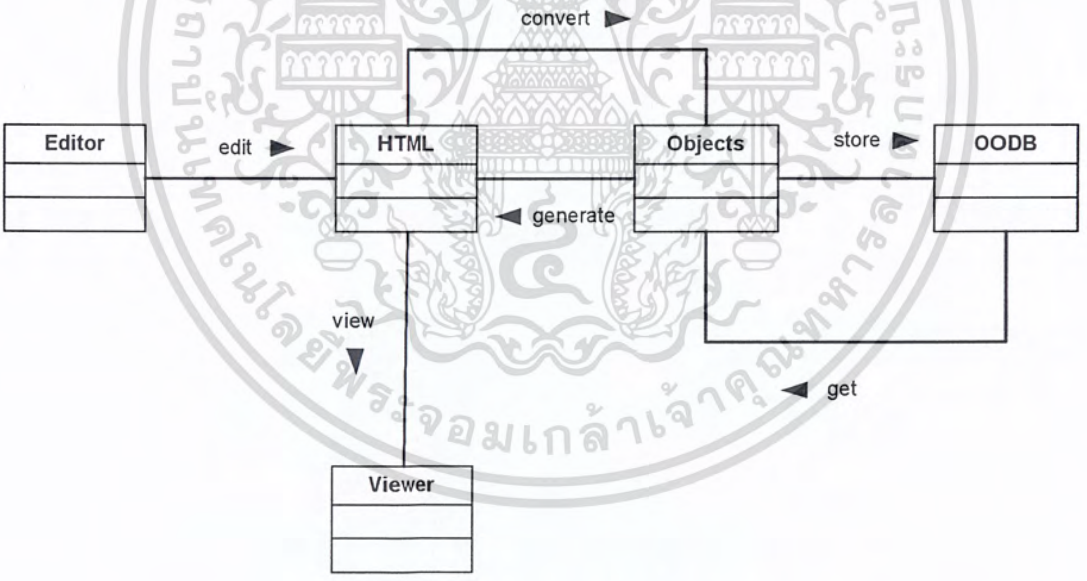
บทที่ 3
การออกแบบซอฟต์แวร์ (Software) ที่ใช้ในโครงงาน

การทำโครงงานในเทอมการศึกษานี้เป็นการดำเนินการในด้านต่างๆได้แก่

1. การศึกษาถึงวิธีการพัฒนาและออกแบบระบบด้วยวิธีเชิงวัตถุ
2. การศึกษาโครงสร้างและส่วนประกอบของเอกสาร HTML เพื่อใช้ในการออกแบบการจัดเก็บเอกสาร HTML ที่ทำการแปลงเป็นวัตถุแล้วเก็บลงในฐานข้อมูล
3. การออกแบบระบบโดยใช้หลักการเชิงวัตถุและใช้เครื่องมือในการจัดทำแบบจำลองของระบบขึ้นมา

จากการทำงานในส่วนต่างๆดังนี้ ทำให้ได้ผลการทำงานออกมาแบ่งเป็นส่วนต่างๆได้ดังนี้

3.1 Client Interview หรือ Context Diagram

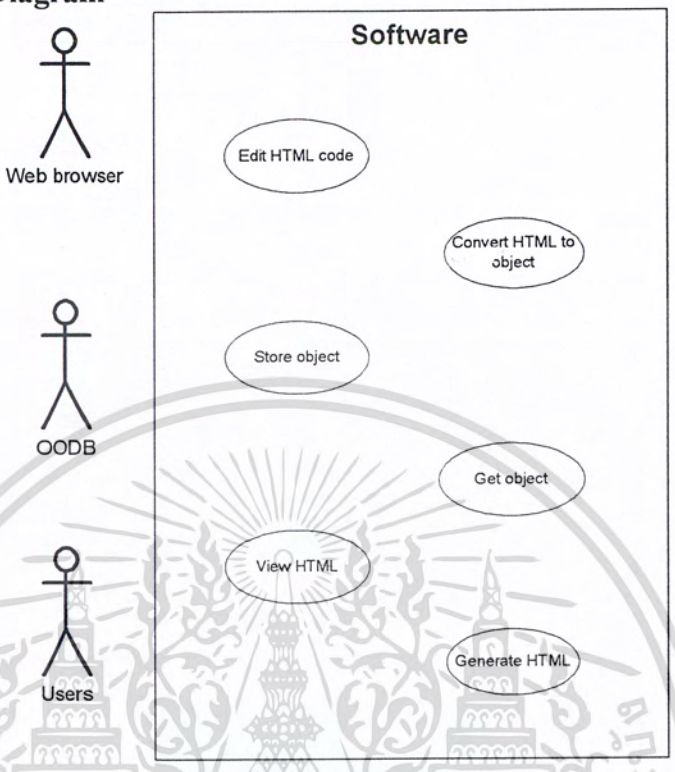


รูปที่ 3-1 แสดง Client Interview ที่ทำการออกแบบ

จากแผนภาพที่ได้ทำการออกแบบสามารถอธิบายถึงโครงสร้างและการกระทำต่างๆที่สามารถเกิดขึ้นได้กับวัตถุนั้นๆ ได้แก่ กระบวนการในการจัดเก็บเอกสาร HTML จะเริ่มจากการที่ตัวแก้ไขเอกสาร HTML ใช้ในการแก้ไขเอกสาร HTML และเอกสาร HTML ถูกแปลงเป็นวัตถุแล้วทำการเก็บลงในฐานข้อมูล ซึ่งจะถูกดึงออกมาเป็นวัตถุแล้วทำการสร้างเอกสาร HTML ขึ้นมาเพื่อทำการแสดงผลที่โปรแกรมที่ใช้ในการแสดงผลในกระบวนการเรียกดูเอกสาร HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

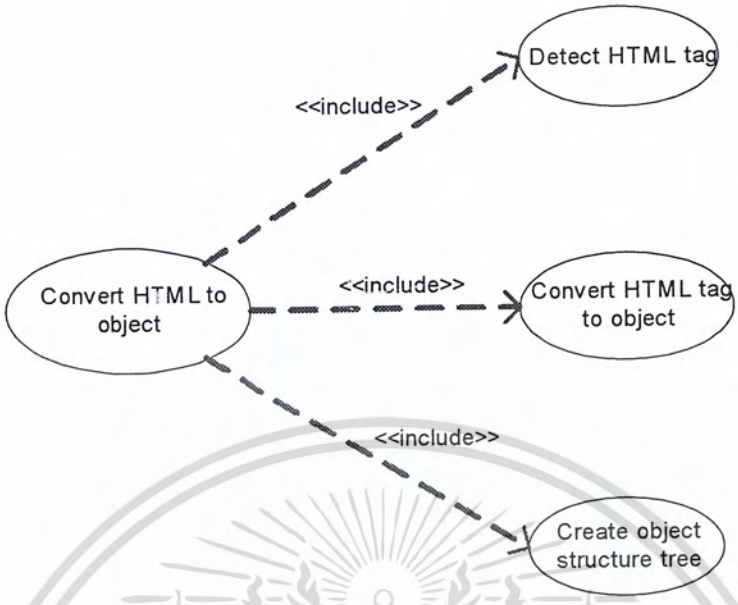
3.2 Use Case Diagram



รูปที่ 3-2 ก. แสดง High Level Use Case Diagram

แผนภาพนี้เป็นการออกแบบระบบแบบโครงสร้างโดยรวม เพื่อแสดงให้เห็นว่าในระบบที่จะทำการพัฒนาขึ้นนั้นมีวัตถุใดๆที่มีความสัมพันธ์เกี่ยวข้องกับระบบที่พัฒนาขึ้นมาบ้าง และสามารถกระทำหรือจัดการกับระบบที่พัฒนาขึ้นมาได้อย่างไรบ้าง โดยมีผลมาจากการออกแบบในขั้นตอนของการทำไคลเอนต์อินเทอร์วิวที่ผ่านมา

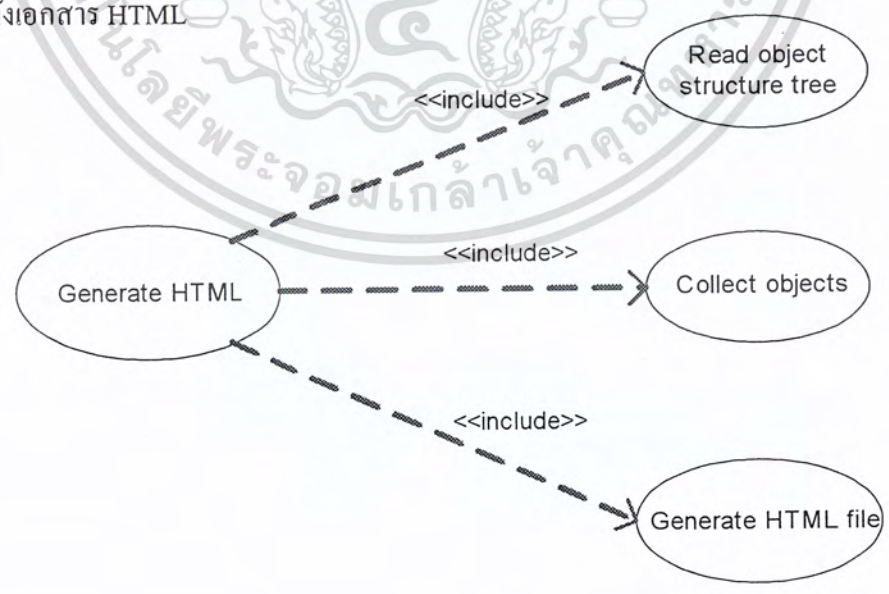
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



The "Convert HTML to object" use case diagram

รูปที่ 3-2 ข. แสดง Convert HTML to Object Use Case Diagram

แผนภาพนี้แสดงลำดับการแปลงจากเอกสาร HTML ให้เป็นวัตถุเพื่อจัดเก็บลงในฐานข้อมูลเชิงวัตถุ โดยเริ่มจากการตรวจหารหัสของเอกสาร HTML แล้วทำการแปลงจากรหัสของเอกสาร HTML นั้นให้เป็นวัตถุ แล้วทำการสร้างโครงสร้างต้นไม้เพื่อใช้ในการเก็บวัตถุนั้นลงในฐานข้อมูลขึ้นมา ถือเป็นการเสร็จสิ้นการแปลงรหัสเอกสาร HTML ให้เป็นวัตถุใน 1 รอบ ทำกระบวนการนี้จนเสร็จสิ้นทั้งเอกสาร HTML

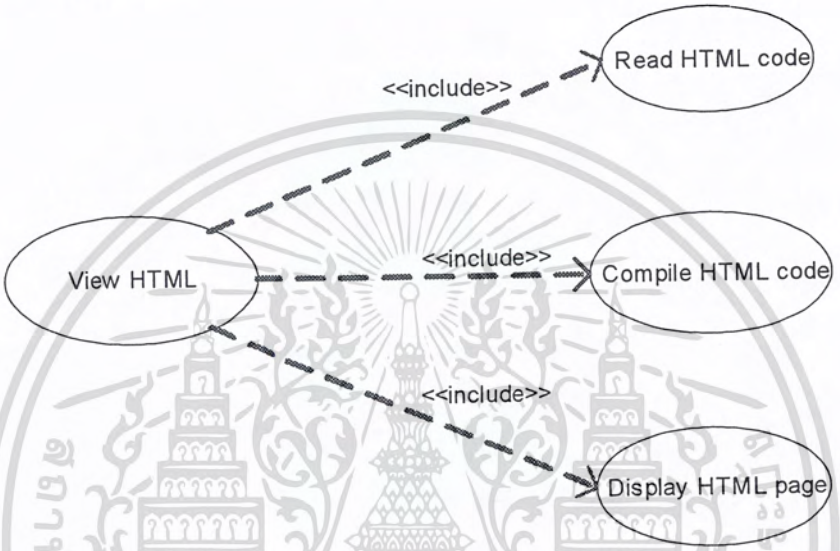


The "Generate HTML" use case diagram

รูปที่ 3-2 ค. แสดง Generate HTML Use Case Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพในการสร้างเอกสาร HTML แสดงลำดับในขั้นตอนการสร้างเอกสาร HTML จากวัตถุที่ดึงมาจากฐานข้อมูล โดยมีลำดับขั้นตอนดังนี้คือ การอ่านข้อมูลจากโครงสร้างต้นไม้ของเอกสาร HTML ที่ต้องการ จากนั้นทำการดึงวัตถุจากภายในฐานข้อมูลออกมาตามลำดับของโครงสร้างต้นไม้ของเอกสาร HTML แล้วทำการสร้างเอกสาร HTML ขึ้นมาเพื่อใช้ในการเรียกดูผ่านทางหน้าจอแสดงผลที่ต้องการ

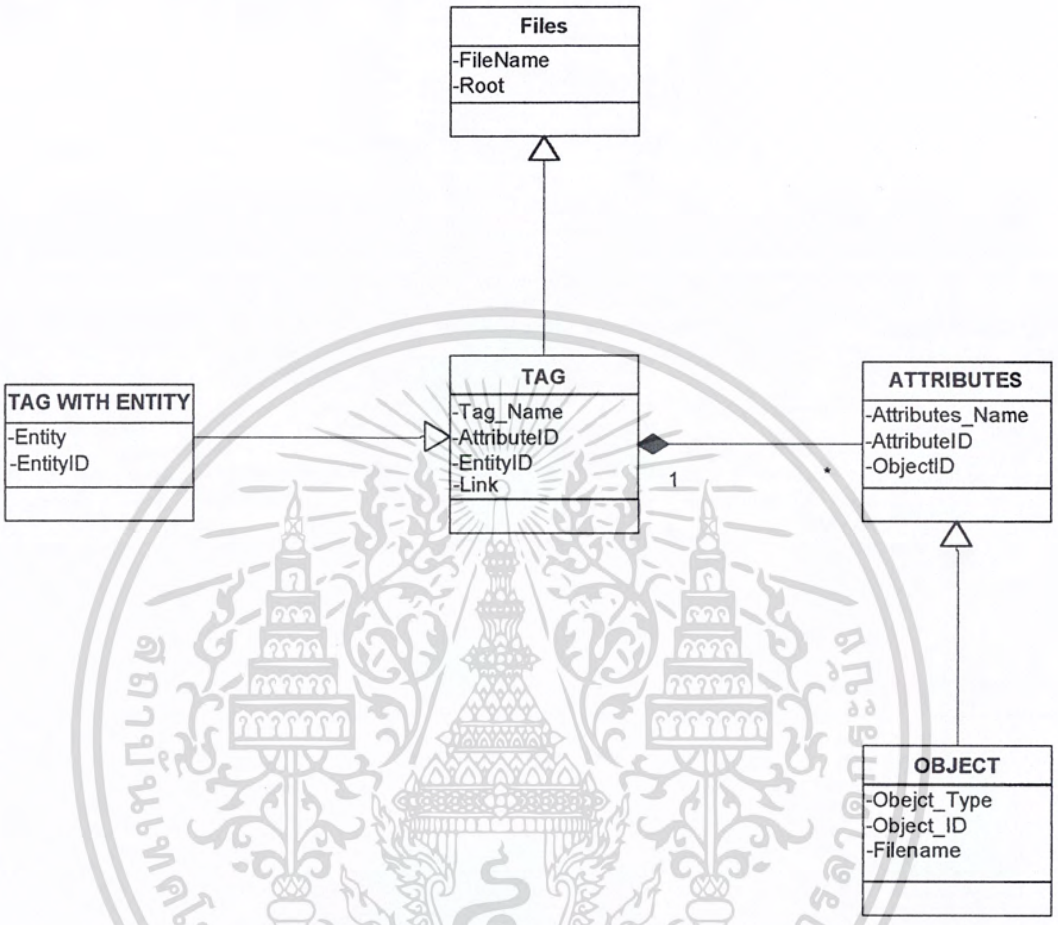


รูปที่ 3-2 ง. แสดง View HTML Use Case Diagram

แผนภาพนี้แสดงลำดับขั้นตอนในการเรียกดูเอกสาร HTML ผ่านทางโปรแกรมที่ใช้แสดงเอกสาร HTML โดยเริ่มจากการอ่านข้อมูลที่เป็นรหัสของเอกสาร HTML แล้วทำการแปลรหัสที่อ่านมาได้ และทำการแสดงผลผ่านทางหน้าจอของโปรแกรมที่ใช้แสดงเอกสาร HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Database Schema Class Diagram



รูปที่ 3-3 ก. แสดง Class Diagram ที่ทำการออกแบบ

จะทำการแปลงเอกสาร HTML ออกเป็น 2 ส่วนคือ

1. ส่วนที่เป็น วัตถุข้อความ (Object Text) จะสามารถแยกเป็น Class Diagram ได้ดังภาพด้านบน
2. ส่วนที่เป็น โครงสร้างของวัตถุ (Object Structure) เพื่อที่จะสามารถบ่งบอกถึงลำดับในการเรียงข้อมูลของ แทค (Tag) ที่อยู่ในเอกสาร HTML โดยจะมีลักษณะโครงสร้างในการจัดเก็บเป็น Binary Balance Tree Structure

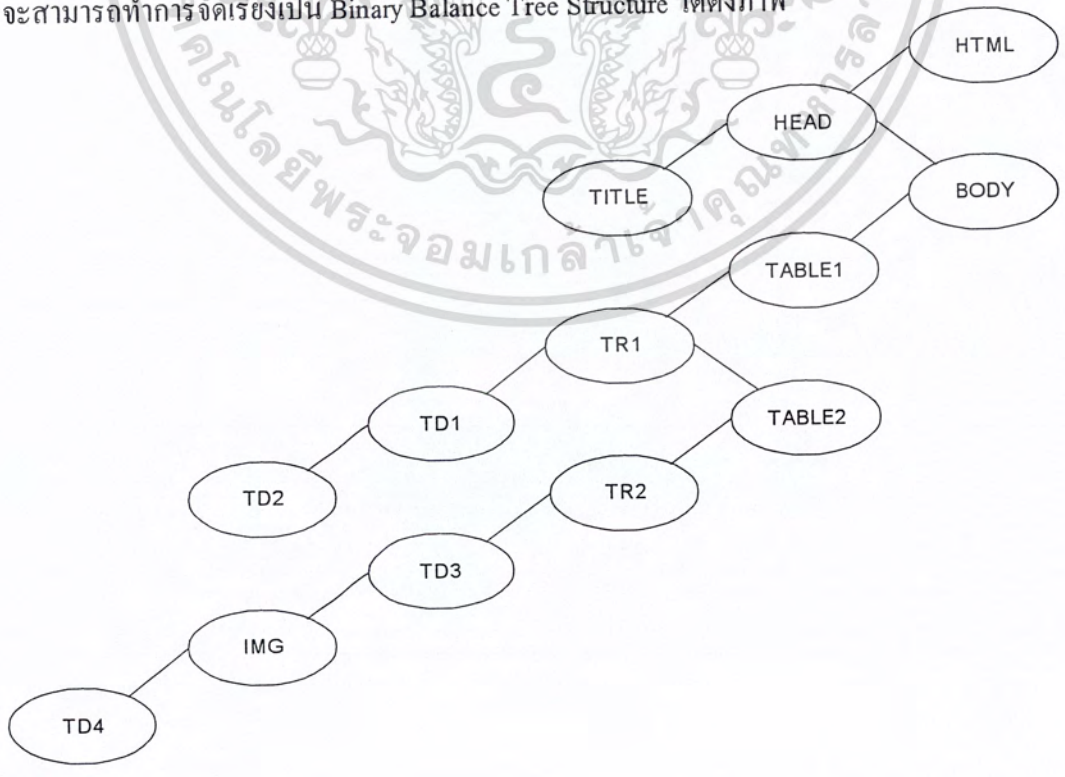
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่าง Source Code HTML อย่างง่ายเช่น

```
<HTML>
    <HEAD>
        <TITLE>
        </TITLE>
    </HEAD>
    <BODY>
        <TABLE>
            <TR>
                <TD>   </TD>
                <TD>   </TD>
            </TR>
        </TABLE>
        <TABLE>
            <TR>
                <TD> <IMG> </TD>
                <TD>   </TD>
            </TR>
        </TABLE>
    </BODY>
</HTML>
```

รูปที่ 3-3 ข. แสดงตัวอย่างเอกสาร HTML ที่ใช้ในการแปลงเป็น Tree

จะสามารถทำการจัดเรียงเป็น Binary Balance Tree Structure ได้ดังภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-3 ค. แสดง Structure Tree ที่ได้จากการออกแบบโดยใช้ Preorder Traversal

โดยอัลกอริทึมที่ใช้ในการออกแบบโครงสร้างของ Tree คือใช้ Preorder Traversal ซึ่งมีหลักการคือ

1. ทำการคิดที่ รูทโหนด (Root Node) หรือโหนดเริ่มต้นโดยทำการวิซิตโหนด (Visit Node) หรือยึดที่โหนดนั้นเป็นหลักในการคิด
2. ถ้าโหนดทางซ้ายของ ชายน์ (Child) ยังไม่สิ้นสุด ให้ทำการกระจายโหนดนั้นต่อไปในทางซ้าย
3. ถ้าโหนดทางขวาของ ชายน์ ยังไม่สิ้นสุด ให้ทำการกระจายโหนดนั้นต่อไปในทางขวา

อธิบายจากแผนภาพ ยกตัวอย่างเช่น ถ้ายึดที่โหนด HEAD จะเห็นได้ว่า จะมีโหนด TITLE เป็น ชายน์ต่อเข้าไปในทางซ้าย แต่โหนด TITLE สิ้นสุดแล้ว (พบ Eng Tag </TITLE>) ถ้าย้อนกลับมาคิดที่โหนด HEAD แสดงว่าทางซ้ายของโหนด HEAD สิ้นสุดแล้วจึงทำการกระจายโหนดเพิ่มต่อไปในทางขวาจึงได้ว่า โหนด BODY อยู่ทางขวาของโหนด HEAD

จากนั้นให้ยึดหลักของ พรีออเดอร์ ทราเวอร์เซาว์ (Preorder Traversal) จะสามารถทำการออกแบบโครงสร้างของทรีได้ทั้งหมด ซึ่งสามารถนำทรีมาเก็บในฐานข้อมูลเชิงวัตถุได้

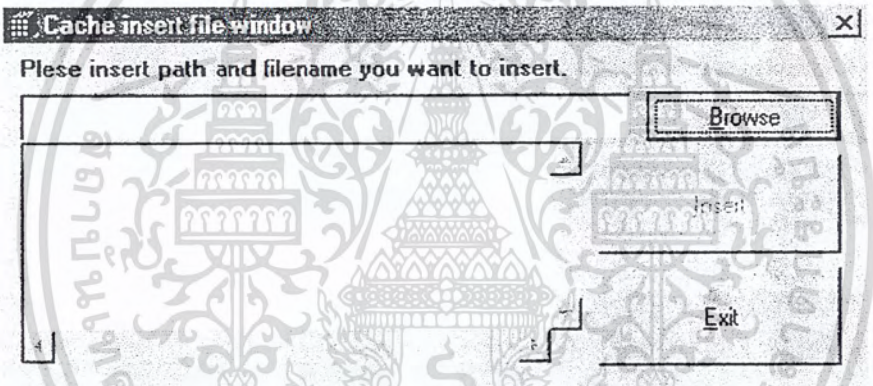
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทำงานของโปรแกรม

กระบวนการทำงานในการจัดการข้อมูลหรือเอกสาร HTML ในฐานข้อมูลนั้นมีอยู่หลายกระบวนการ อธิบายแยกตามกระบวนการทำงานในส่วนต่างๆของโปรแกรมดังนี้

4.1 การจัดเก็บเอกสาร HTML ลงในฐานข้อมูลเชิงวัตถุ (Insert)

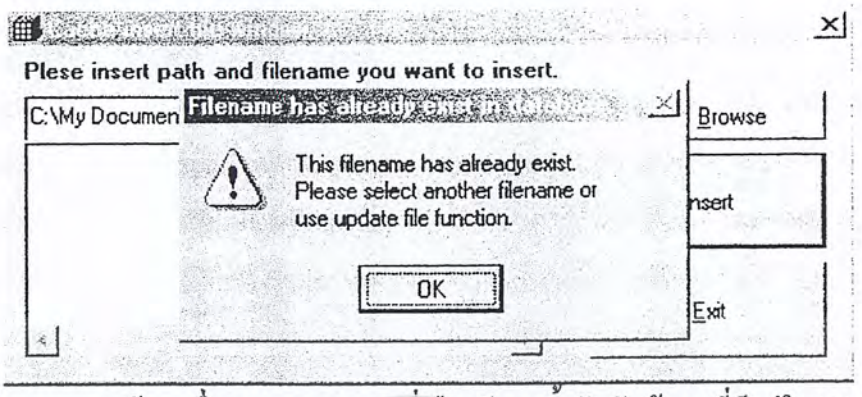
การจัดเก็บข้อมูลลงในฐานข้อมูลเชิงวัตถุจะเริ่มต้นการทำงานโดยการแสดงหน้าจอการทำงานเริ่มต้นดังรูปที่ 4-1 ก.



รูปที่ 4-1 ก. แสดงหน้าจอที่ใช้จัดเก็บเอกสาร HTML ลงในฐานข้อมูลเชิงวัตถุ

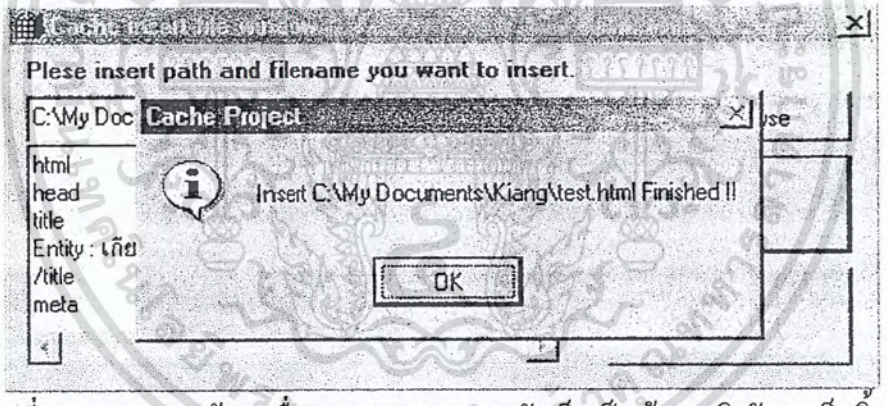
โดยมีขั้นตอนการทำงานเริ่มจากการทำการพิมพ์เส้นทางและชื่อเอกสารหรือทำการเลือกเอกสารที่จะทำการจัดเก็บลงในฐานข้อมูล ซึ่งจะมีการแสดงหน้าจออีกหน้าจอหนึ่งเพื่อใช้ในการเลือกเอกสารปรากฏขึ้นมา จากนั้นเมื่อทำการกดปุ่มเพื่อเริ่มการทำการจัดเก็บเอกสาร โปรแกรมจะทำการตรวจสอบก่อนว่าเส้นทางและชื่อเอกสารนั้นๆได้มีการจัดเก็บไว้ในฐานข้อมูลแล้วหรือไม่ หากตรวจสอบแล้วพบว่าเส้นทางและชื่อเอกสารนั้นได้มีการจัดเก็บไว้ในฐานข้อมูลแล้ว ก็จะปรากฏหน้าจอแสดงการเตือนขึ้นมา ดังรูปที่ 4-1 ข. เนื่องจากโปรแกรมนี้จะไม่อนุญาตให้มีการจัดเก็บข้อมูลที่มีเส้นทางและชื่อเอกสารเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



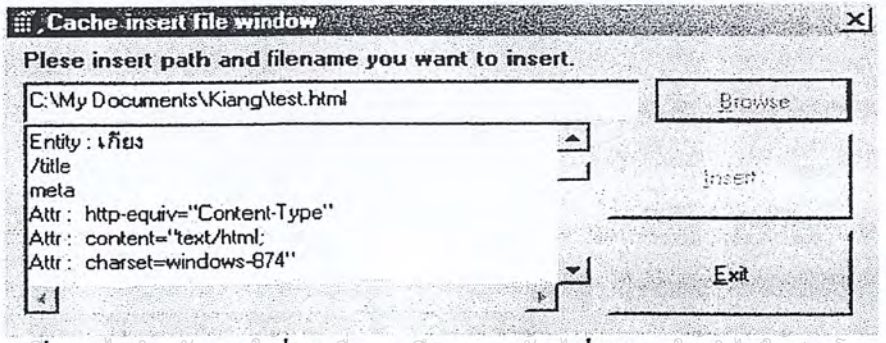
รูปที่ 4-1 ข. แสดงหน้าจอเมื่อเอกสาร HTML ที่เลือกมีการซ้ำกันกับข้อมูลที่มีอยู่ในฐานข้อมูล

หากไม่มีการซ้ำกันของเส้นทางและชื่อเอกสารในฐานข้อมูล โปรแกรมก็จะทำงานไปตามปกติคือจะทำการจัดการแบ่งแยกเอกสาร HTML เหล่านั้นเพื่อให้ข้อมูลภายในเอกสารมีลักษณะที่เป็นเชิงวัตถุ จากนั้นจะทำการจัดเก็บลงฐานข้อมูลในลักษณะเชิงวัตถุ และเมื่อทำการจัดเก็บเอกสารเสร็จเรียบร้อยแล้วก็จะปรากฏหน้าจอแสดงข้อความว่าโปรแกรมนั้นได้ทำงานเสร็จเรียบร้อยแล้ว ดังรูปที่ 4-1 ค. แสดงว่าการทำงานของโปรแกรมในการจัดเก็บเอกสารนั้นสำเร็จเรียบร้อยดี



รูปที่ 4-1 ค. แสดงหน้าจอเมื่อเอกสาร HTML ถูกจัดเก็บเป็นข้อมูลเชิงวัตถุเสร็จสิ้น

หลังจากที่โปรแกรมได้ทำการจัดเก็บข้อมูลลงฐานข้อมูลเรียบร้อยแล้ว ผู้ใช้สามารถที่จะดูการแบ่งแยกข้อมูลในเอกสารที่มีลักษณะเป็นเชิงวัตถุได้ในช่องข้อความที่ได้มีการแสดงผลไว้บนหน้าจอการทำงานของโปรแกรม ดังรูปที่ 4-1 ง.

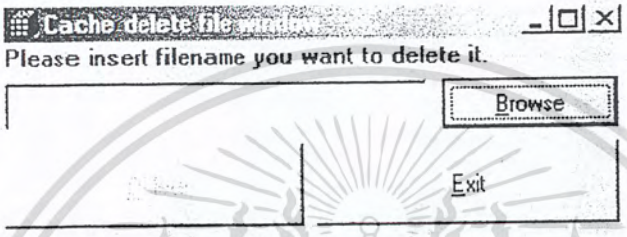


รูปที่ 4-1 ง. แสดงหน้าจอที่ใช้แสดงผลข้อมูลเอกสารที่มีลักษณะเป็นเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

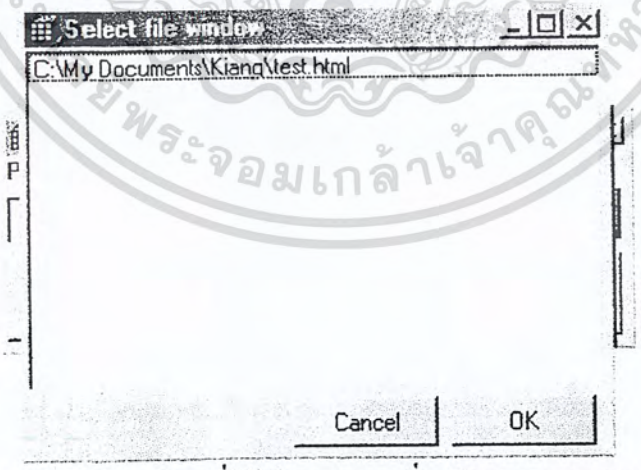
4.2 การลบเอกสาร HTML ที่มีอยู่ในฐานข้อมูลเชิงวัตถุ(Delete)

สำหรับการจัดการลบข้อมูลหรือเอกสารภายในฐานข้อมูลเชิงวัตถุนั้นจะต้องทำการลบ ข้อมูลที่อยู่ในรูปแบบเชิงวัตถุและต้องทำการลบการเชื่อมโยงกันระหว่างวัตถุแต่ละตัวภายใน ฐานข้อมูลด้วย เพื่อมีให้เกิดข้อผิดพลาดในการเรียกใช้งานฐานข้อมูลได้ โดยมีหน้าจอเริ่มต้นการทำงานดังรูปที่ 4-2 ก.



รูปที่ 4-2 ก. แสดงหน้าจอที่ใช้ในการลบเอกสาร HTML ออกจากฐานข้อมูล

การทำงานในส่วนของการลบเอกสารในฐานข้อมูลนั้นจะมีลักษณะเหมือนกับการจัดเก็บลงในฐานข้อมูล โดยเริ่มจากส่วนในการเลือกเอกสารที่จะทำการลบก่อน โดยที่จะมีหน้าจอที่แสดงชื่อเอกสารทั้งหมดที่มีอยู่ในฐานข้อมูลขณะนั้นเพื่อให้ผู้ใช้ได้ทำการเลือกเอกสารที่ต้องการจะลบดังรูปที่ 4-2 ข. จากนั้นโปรแกรมจึงจะสามารถทำการจัดการลบข้อมูลของเอกสารนั้นภายในฐานข้อมูลทิ้งไปได้ และหลังจากที่ได้ทำการลบข้อมูลของเอกสารนั้นเรียบร้อยแล้วก็จะมีหน้าจอข้อความแสดงขึ้นมาเพื่อบอกให้ผู้ใช้ทราบด้วยเช่นกัน



รูปที่ 4-2 ข. แสดงหน้าจอที่ใช้เลือกเอกสารที่จะทำการลบออกจากฐานข้อมูล

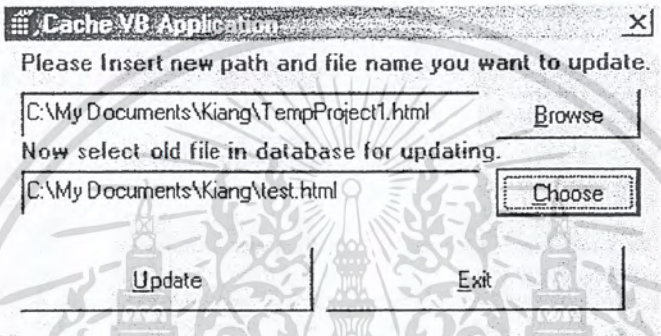
4.3 การแก้ไขเอกสาร HTML ที่มีอยู่ในฐานข้อมูลเชิงวัตถุ(Update)

สำหรับในการจัดการแก้ไขเอกสาร HTML นั้นปกติจะทำโดยการแก้ไขเอกสารภายนอกฐานข้อมูลแล้วทำการจัดเก็บเอกสารลงไปใหม่ในฐานข้อมูล ในโปรแกรมนี้ก็ใช้หลักการนี้เช่นเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กัน ในส่วนเริ่มต้นการทำงานของโปรแกรมจึงปรากฏหน้าจอที่ให้ผู้ใช้ทำการเลือกเอกสารที่ทำการแก้ไขแล้วและเลือกเอกสารที่ต้องการแก้ไขที่อยู่ในฐานข้อมูล ดังรูปที่ 4-3

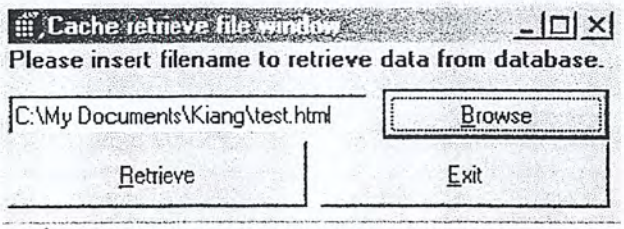
จากนั้นโปรแกรมจะทำการแก้ไขข้อมูลในเอกสารที่อยู่ในฐานข้อมูลให้ตรงกับข้อมูลในเอกสารที่ทำการแก้ไขเรียบร้อยแล้ว ซึ่งกระบวนการในการทำการแก้ไขข้อมูลนี้จะเป็นส่วนที่ใช้เวลาโดยเฉลี่ยในการประมวลผลนานกว่ากระบวนการอื่นๆ เมื่อทำการแก้ไขเอกสารเรียบร้อยแล้วก็จะแสดงหน้าจอข้อความเพื่อบอกให้ผู้ใช้ทราบ



รูปที่ 4-3 แสดงหน้าจอที่ใช้ในการแก้ไขข้อมูลเอกสาร HTML ที่อยู่ในฐานข้อมูล

4.4 การเรียกดูเอกสาร HTML จากฐานข้อมูลเชิงวัตถุ(Retrieve)

การเรียกดูเอกสาร HTML จากฐานข้อมูลเชิงวัตถุนั้น ผู้ใช้จะต้องทำการเลือกเอกสารที่มีอยู่ในฐานข้อมูลที่ต้องการ โดยมีหน้าจอการทำงานดังรูปที่ 4-4 จากนั้นโปรแกรมก็จะทำการดึงเอาข้อมูลเชิงวัตถุของเอกสารนั้นมาทำการเรียบเรียงให้ถูกต้องตามลำดับในการจัดเก็บ เพื่อให้ข้อมูลที่จะถูกแสดงผลออกมามีความถูกต้องตรงกับเอกสารที่จัดเก็บในฐานข้อมูล การทำการเรียกดูข้อมูลนั้นจะเริ่มต้นด้วยหน้าจอที่ให้ทำการเลือกเอกสาร HTML ที่มีอยู่ในฐานข้อมูล เพื่อเลือกเอกสารที่ต้องการเรียกดู จากนั้นเมื่อทำการเรียกดูเอกสารโปรแกรมก็จะทำการประมวลผลในการจัดการดึงข้อมูลเชิงวัตถุของเอกสารที่เลือกไปนั้นออกมา แล้วทำการเรียบเรียงให้ตรงกับข้อมูลในเอกสารที่ถูกจัดเก็บลงไปในฐานข้อมูล แล้วทำการแสดงผลผ่านทางหน้าจอแสดงเอกสาร HTML ของโปรแกรม

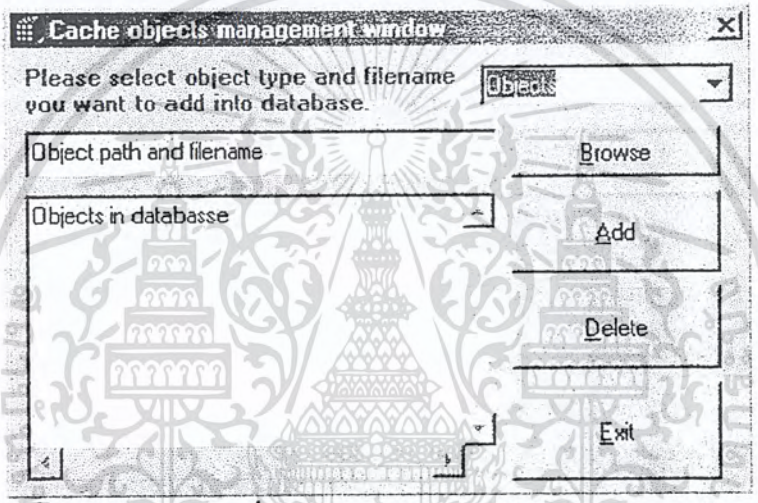


รูปที่ 4-4 แสดงหน้าจอที่ใช้การเรียบเรียงข้อมูลจากฐานข้อมูลให้อยู่ในรูปแบบของเอกสารHTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การใส่ข้อมูลที่อยู่ในรูปแบบของข้อมูลเชิงวัตถุที่ปรากฏอยู่ในเอกสาร HTML ลงไปในฐานข้อมูล

เนื่องจากเอกสาร HTML จะมีลักษณะของข้อมูลหลากหลาย ทั้งข้อมูลที่เป็น ข้อความ รูปภาพ ภาพเคลื่อนไหว หรือข้อมูลเสียง ซึ่งข้อมูลที่มีลักษณะเหล่านี้ จะถูกจัดเก็บอยู่ที่เซิร์ฟเวอร์ เพื่อให้สามารถเรียกดูผ่านทางบาว์เซอร์ (Browser) ได้ ดังนั้นเป็นให้เอกสาร HTML มีความสมบูรณ์ครบถ้วน จึงต้องมีการจัดเก็บข้อมูลที่มีลักษณะดังกล่าวลงในฐานข้อมูลเชิงวัตถุด้วย โดยจะทำการจัดเก็บผ่านทางโปรแกรมที่ทำการออกแบบไว้ ซึ่งมีหน้าจอการทำงานดังรูปที่ 4-5



รูปที่ 4-5 แสดงหน้าจอเริ่มต้นในการจัดการกับวัตถุข้อมูลในฐานข้อมูล

4.6 ผลการทดลอง

4.6.1 การจัดเก็บเอกสาร HTML ลงไปในฐานข้อมูลสามารถทำงานได้ผลดีตามที่ต้องการ โดยสามารถที่จะทำการแยกข้อมูลในเอกสาร HTML ให้เป็นเชิงวัตถุได้อย่างถูกต้องตามที่ได้ออกแบบเอาไว้ มีความเร็วในการประมวลผลในระดับที่น่าพอใจ

4.6.2 การลบเอกสาร HTML ที่มีอยู่ภายในฐานข้อมูลสามารถทำการลบข้อมูลเชิงวัตถุของเอกสาร HTML ที่ต้องการได้ทั้งหมดโดยที่ไม่มีผลต่อข้อมูลเชิงวัตถุของเอกสาร HTML อื่นๆ มีความเร็วในการประมวลผลอยู่ในระดับที่ดี

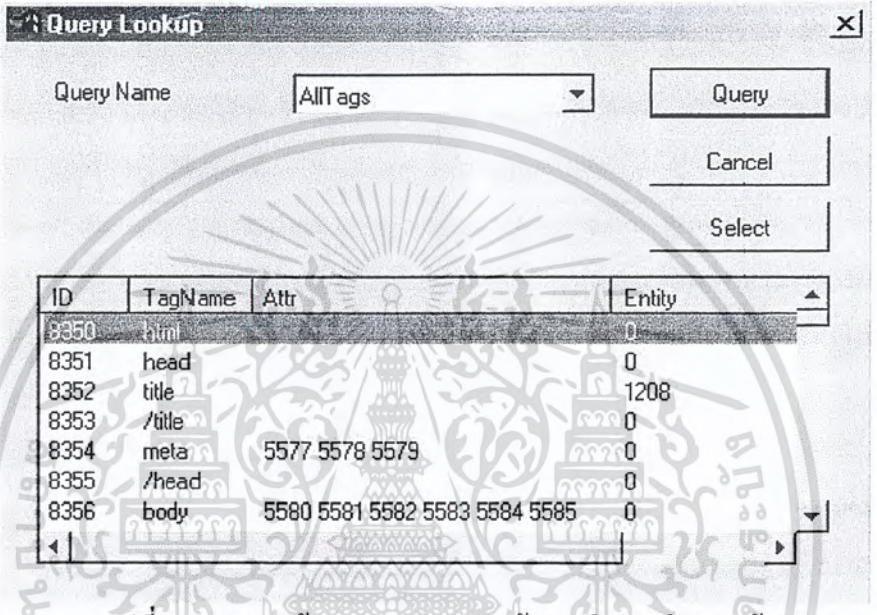
4.6.3 การแก้ไขเอกสาร HTML สามารถทำงานได้เป็นที่น่าพอใจ สามารถทำการแก้ไขเอกสาร HTML ที่ต้องการได้อย่างถูกต้อง ความเร็วในการประมวลผลอยู่ในระดับที่ยอมรับได้

4.6.4 การเรียกดูเอกสาร HTML สามารถเรียกดูเอกสารที่มีความถูกต้องสูงตามที่ได้ทำการจัดเก็บเอกสารลงในฐานข้อมูล และมีความเร็วอยู่ในระดับที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

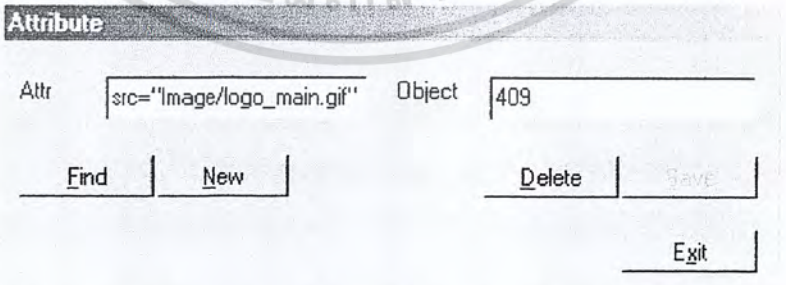
4.6.5 ในส่วนของการทำงานของโปรแกรมฐานข้อมูลมีผลการทำงานโดยเฉพาะในส่วนของความเร็วในการจัดการกับข้อมูลสามารถทำได้อยู่ในระดับที่ดี โปรแกรมมีฟังก์ชันการทำงานให้เลือกมากมายเพียงพอต่อการใช้งานตั้งแต่ในระดับพื้นฐานไปจนถึงการใช้งานในระดับสูง สามารถนำไปประยุกต์ใช้งานได้หลายประเภท

4.7 ตัวอย่างข้อมูลเชิงวัตถุในฐานข้อมูล



รูปที่ 4-6 แสดงหน้าจอการตรวจสอบข้อมูลเชิงวัตถุในฐานข้อมูล

จากรูปที่ 4-6 เป็นหน้าจอในการแสดงการคิวรีที่ได้มาจากการสร้างคิวรีเก็บไว้ในฐานข้อมูล ซึ่งจะต้องสร้างขึ้นเพื่อใช้ในการเรียกดูต่างหาก ในรูปเป็นการคิวรีข้อมูลเชิงวัตถุของคลาสเทค ซึ่งมีข้อมูลของรหัสของวัตถุ ชื่อของวัตถุแทค เลขประจำตัวของวัตถุแอททริบิวต์และเลขประจำตัวของวัตถุเอนติตี้ที่เป็นสมาชิกของวัตถุแทคนั้นๆตามที่ได้ออกแบบไว้



รูปที่ 4-7 แสดงหน้าจอของผู้ใช้ในการคิวรีดูข้อมูลเชิงวัตถุในฐานข้อมูล

จากภาพเป็นการแสดงหน้าจอของผู้ใช้ในการเรียกดูข้อมูลเชิงวัตถุในฐานข้อมูลของคลาสแอททริบิวต์ซึ่งจะมีการเก็บค่าของเลขประจำวัตถุของคลาสออบเจ็คต์ที่มีการเชื่อมโยงข้อมูลกันไว้ด้วย หน้าจอในการเรียกดูนี้สามารถสร้างได้โดยการใช้วิซาร์ดของโปรแกรมฐานข้อมูลสร้างได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

จากการออกแบบและทดลองใช้โปรแกรมที่ใช้ในการแปลงเอกสาร HTML พบว่าสามารถทำการจัดเก็บเอกสาร HTML ลงไปในฐานข้อมูลได้จริงตามวัตถุประสงค์

5.1.1 ซอฟต์แวร์ (Software) ที่เลือกใช้ในโครงงาน

1. โปรแกรมที่ใช้ในการแปลงเอกสาร HTML ใช้โปรแกรม Visual Basic
2. โปรแกรมที่ใช้เป็นฐานข้อมูลในการจัดเก็บเอกสาร HTML ใช้โปรแกรม Cache

5.1.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

1. การศึกษาและรวบรวมข้อมูลของเอกสาร HTML ทำได้ลำบาก เนื่องจากแหล่งข้อมูลอ้างอิงนั้นมีอยู่เป็นจำนวนมาก แต่ไม่สามารถครอบคลุมรายละเอียดทั้งหมดของเอกสาร HTML ได้ จึงทำให้จำเป็นต้องทำการเก็บรวบรวมและศึกษารายละเอียดและโครงสร้างของเอกสาร HTML เป็นเวลานาน
2. มีความเข้าใจในระบบซอฟต์แวร์และขอบเขตที่จะทำการพัฒนาน้อย จึงทำให้การทำการวิเคราะห์และออกแบบระบบทำได้ยาก
3. การใช้เครื่องมือในการสร้างแบบจำลองของระบบมีความสับสน เนื่องจากมีการใช้เอกสารเพื่ออ้างอิงหลายเล่มแต่มีเนื้อหาที่ขัดแย้งกันในบางส่วน เมื่อทำการออกแบบมาแล้วจึงต้องมีการเปลี่ยนแปลงแก้ไขเพื่อให้ถูกต้อง
4. ในขั้นตอนการเขียนโปรแกรมเพื่อทำการติดต่อกับฐานข้อมูล สามารถทำได้ยาก เนื่องจากต้องใช้คำสั่งพิเศษของโปรแกรมฐานข้อมูลเพื่อทำการติดต่อกับฐานข้อมูล ซึ่งคำสั่งเหล่านั้นต้องใช้เวลาในการทำความเข้าใจ และยากแก่การค้นคว้าหาข้อมูลเพิ่มเติม หรือค้นหาตัวอย่างของโปรแกรมใช้คำสั่งพิเศษเหล่านั้น

5.2 วิจารณ์ผลการทดลอง

5.2.1 การจัดเก็บเอกสาร HTML ลงฐานข้อมูลมีความเร็วอยู่ในระดับที่ช้ากว่าที่ต้องการ หากทำการจัดเก็บเอกสารที่มีปริมาณของข้อมูลเป็นจำนวนมาก เนื่องจากในการทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแยกข้อมูลออกเป็นข้อมูลเชิงวัตถุนั้นจะต้องทำการตรวจสอบข้อมูลอย่างละเอียด ซึ่งส่งผลต่อความเร็วในการจัดเก็บเอกสาร เมื่อเทียบกับการจัดเก็บลงในฐานข้อมูลเชิงสัมพันธ์จะมีความเร็วในการประมวลผลช้ากว่า

5.2.2 การลบเอกสาร HTML สามารถทำงานได้เร็วกว่าในส่วนของการทำงานอื่นๆ เนื่องจากเป็นการประมวลผลและจัดการกับข้อมูลที่อยู่ภายในฐานข้อมูลเท่านั้น ความเร็วในการประมวลผลถือว่าใกล้เคียงกับการทำงานในฐานข้อมูลเชิงสัมพันธ์

5.2.3 การทำการแก้ไขเอกสาร HTML มีการทำงานที่ช้าที่สุด เนื่องจากจะต้องทำการตรวจสอบเอกสาร HTML ทั้งที่อยู่ภายในและภายนอกฐานข้อมูล ยิ่งเอกสาร HTML มีปริมาณข้อมูลมากก็ยิ่งใช้เวลาในการประมวลผลมาก ซึ่งเมื่อเทียบกับการจัดเก็บลงในฐานข้อมูลแบบเชิงสัมพันธ์แล้ว ในส่วนของการทำงานตรงส่วนนี้เป็นข้อเสียเปรียบในการทำงานมากที่สุด

5.2.4 การเรียกดูเอกสาร HTML จะทำงานได้ช้ากว่าเมื่อเทียบการจัดเก็บเอกสารในฐานข้อมูลเชิงสัมพันธ์ เนื่องจากจะต้องมีการทำการเรียบเรียงข้อมูลเชิงวัตถุที่ดึงออกมาจากฐานข้อมูลเสียก่อน แต่ก็มีความเร็วในระดับที่น่าพอใจ

5.2.5 ในส่วนของการทำงานของตัวโปรแกรมฐานข้อมูลมีความเร็วในการทำงานที่ยังอยู่ในระดับที่ยังไม่เพียงพอหากต้องการใช้งานในระดับที่มีข้อมูลจำนวนมาก และควรที่จะมีการพัฒนาให้ผู้ใช้สามารถเข้าใจการทำงานและเริ่มใช้งานได้ง่ายยิ่งขึ้นในรุ่นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. W3C HTML 4.01 Specification Website
   http://www.w3.org/TR/html401
2. WDG Web Design Group HTML 4.0 Reference
   http://www.htmlhelp.com/reference/html40
3. InterSystems Caché Website
   http://www.e-dbms.com/index.html
4. Introduction to Caché Version 3.1
   http://www.e-dbms.com/downloads/documentation/cache31/docs/intro/intro-cacheTOC.html
5. Object Database Website
   http://www.odbmsfacts.com/articles/object_database_definition.html
6. หนังสือ Visual Basic 6 ฉบับโปรแกรมเมอร์
   กิตติ ภักดีวัฒนะกุล – จำลอง ครูอุตสาหะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

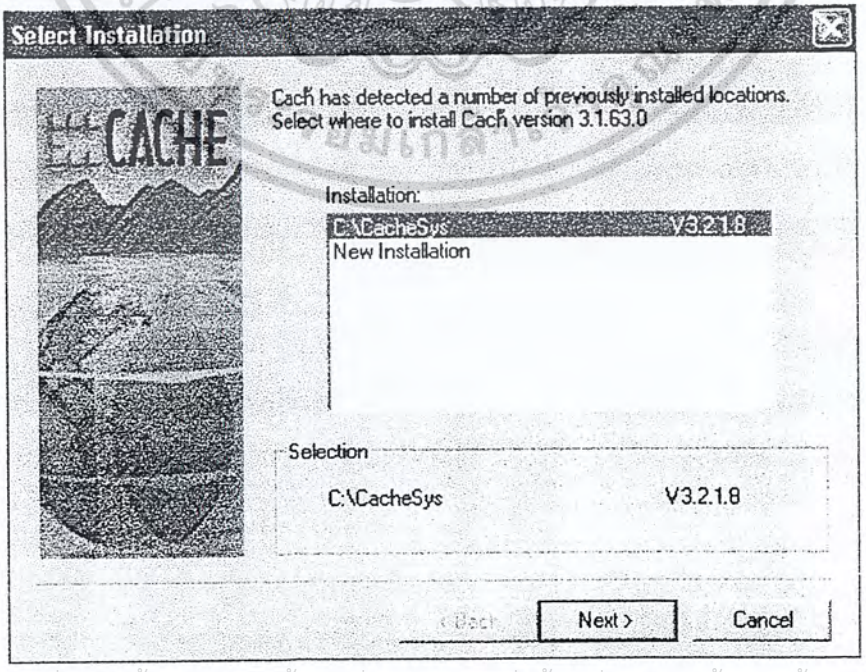
ภาคผนวก ก

ระบบจัดการฐานข้อมูล (DBMS) ที่นำมาใช้งานคือ Caché (Version 3.2) ซึ่งเป็นระบบจัดการฐานข้อมูล (DBMS)ที่ถูกพัฒนาขึ้นโดยบริษัท InterSystem Corp. โดย Caché ถูกจัดอยู่ในพวก DBMS แบบ Post-Relational Database ซึ่งหัวใจหลักของ Caché คือกระบวนการ Transactional Multidimensional Data Model (TMDM) ข้อมูลจะถูกเก็บผ่านทาง multidimensional array ที่มีหลายมิติ เช่น 2 หรือ 3 จนถึง n มิติ ทำให้มีสามารถในการรองรับผู้ใช้จำนวนมากเพื่อทำการเข้าถึงข้อมูลได้

ขั้นตอนของการติดตั้งโปรแกรม Caché

1. ความต้องการขั้นต่ำก่อนที่จะทำการลง Caché คือ หน่วยประมวลผลที่มีความเร็วอย่างน้อย 50 MHz และเนื้อที่ว่าง 10-100 Mb ของ Disk Storage สามารถใช้งานได้กับระบบปฏิบัติการแบบ Window NT 4.0 , Window 95 , Window 98

2. ทำการใส่แผ่น CD โปรแกรมจากนั้นเลือก Run จาก Menu เพื่อทำการเปิด Text Box จากนั้นพิมพ์ข้อความ Drive(ในที่นี้หมายถึง CD-rom Drive):\nt\Setup.exe



รูปที่ ก-1 แสดงหน้าจอในการติดตั้งโปรแกรม Caché

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากหน้าจอ Caché Setup ให้เลือกที่ Install จะเป็นการเริ่มติดตั้งโปรแกรมลงใน Disk Storage

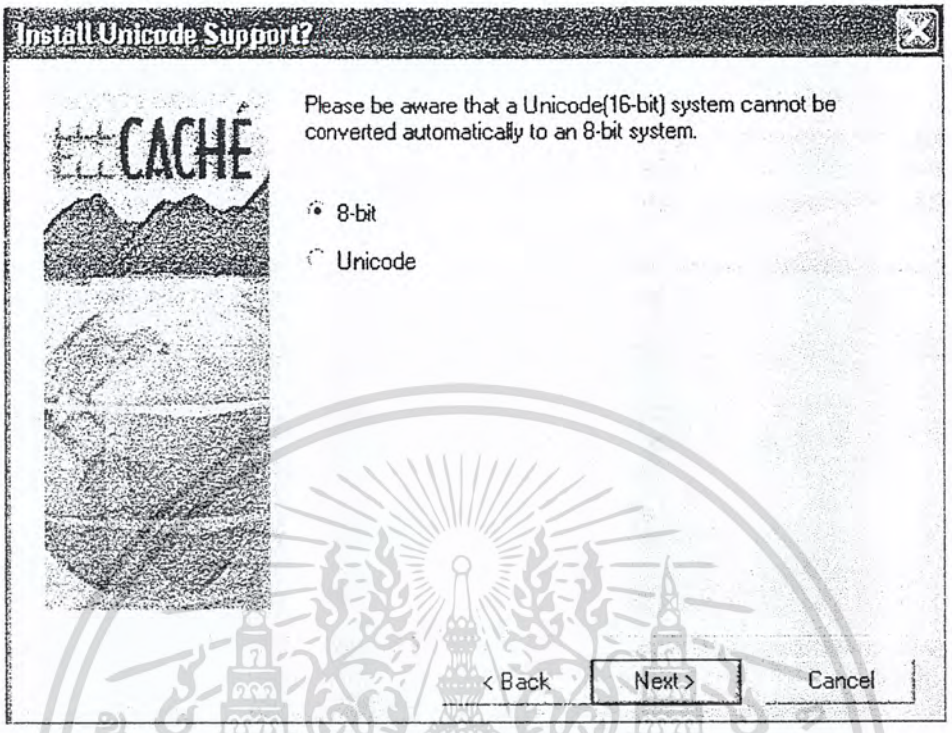
4. ถ้าเป็นการติดตั้งใหม่ให้ เลือก Yes เพื่อเป็นการยืนยันว่ายอมรับในข้อตกลงการใช้งาน Caché

5. จากนั้นจะปรากฏหน้าจอให้ใส่ชื่อ โดยชื่อเริ่มต้นที่กำหนดมาให้คือ CACHÉ

6. ที่หน้าจอ Setup Type จะเป็นการเลือกว่าจะทำการติดตั้งในฐานข้อมูลในรูปแบบใด
   - Standard จะทำการติดตั้ง Server,Client และ ODBC Client Driver component ต่างๆ รวมไปถึง Caché Objects ซึ่งสามารถใช้สร้างโปรแกรมประยุกต์ที่อยู่ในรูปของฐานข้อมูลเชิงวัตถุได้ ให้ทำการเลือกหัวข้อนี้ ถ้าต้องการใช้งานเป็น Caché Database Server
   - Client จะทำการติดตั้งเฉพาะ Caché Client component ให้ทำการเลือกหัวข้อนี้ ถ้าต้องการใช้งานระบบของ Caché โดยที่มีเครื่องอื่นเป็น Caché Database Server
   - Custom จะทำการเลือกติดตั้งเฉพาะส่วนที่ต้องการ
     ภายในหน้าจอนี้จะอนุญาตให้เราทำการเลือก Directory ที่จะทำการติดตั้งโปรแกรมได้ (โดย Directory ที่กำหนดมาให้คือ C:\CacheSys)

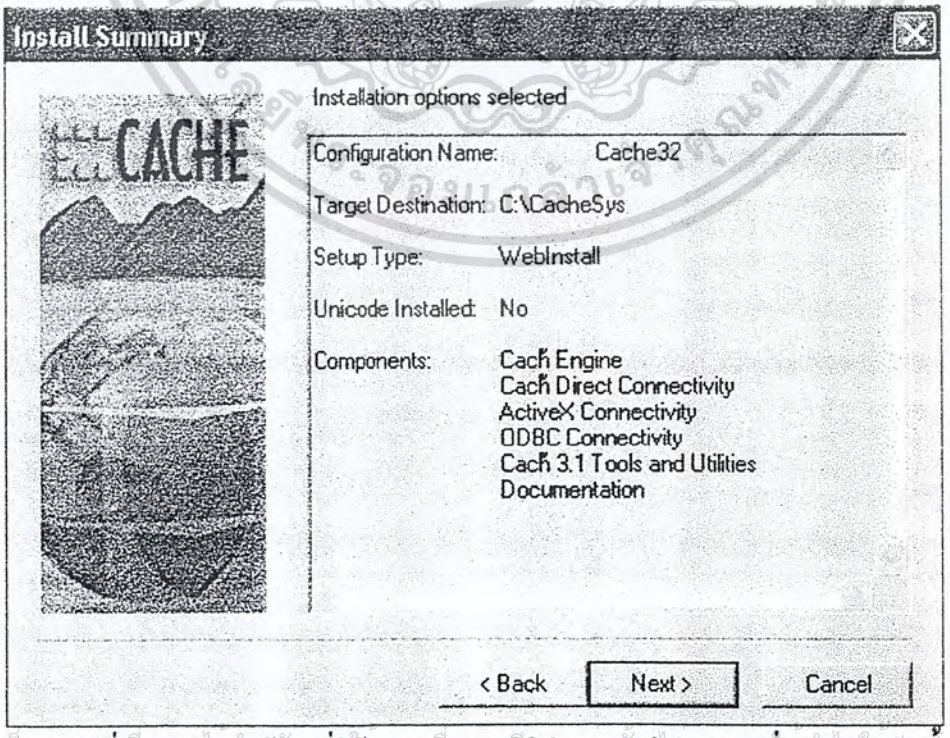
7. ทำการเลือกคุณสมบัติของโปรแกรมว่าเป็นแบบใด ระหว่าง 8 bit กับ Unicode โดย
   - 8 bit โปรแกรมจะจัดการให้ข้อมูลอยู่ในรูปแบบของข้อมูล 8 bit
   - Unicode โปรแกรมจะจัดการให้ข้อมูลอยู่ในรูปแบบของข้อมูล 16 bit ให้ทำการเลือกหัวข้อนี้ถ้าต้องการให้โปรแกรมจัดการกับภาษาที่อยู่ในรูปแบบของ 16 Bit-language เช่น ภาษาญี่ปุ่น
   - คำเตือนถ้าทำการเลือกติดตั้งแบบ Unicode จะไม่สามารถคืนข้อมูลกลับไปให้อยู่ในรูป 8 bit ได้โดยปราศจากการสูญเสียของข้อมูล เพราะว่าข้อมูลแบบ 16 bit ไม่สามารถที่จะเก็บลงในฐานข้อมูลที่มีการจัดการข้อมูลแบบ 8 bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-2 แสดงหน้าจอในการเลือกติดตั้งคุณสมบัติของโปรแกรมว่าเป็นแบบใด

8. เมื่อทำการเลือกรูปแบบในการติดตั้งเสร็จสิ้นแล้วให้ กด Next เพื่อทำการติดตั้งโปรแกรมลงบน Directory



รูปที่ ก-3 แสดงหน้าจอที่ใช้แสดงคุณสมบัติต่างๆของโปรแกรมที่จะทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ที่หน้าจอ Setup Complete ให้เลือกที่ Finish โปรแกรม Caché จะถูกเรียกใช้งานโดยอัตโนมัติ จากนั้นให้เลือกที่ Exit เพื่อเป็นการสิ้นสุดการติดตั้งโปรแกรม Caché

ภายหลังจากทำการติดตั้งเสร็จสิ้นจะปรากฏ Icon Caché Cube อยู่บน Window Tool Bar ใน System Tray Area กดคลิกขวาเพื่อทำการเรียกดู Menu ของ Caché ส่วนที่เพิ่มอีกส่วนหนึ่งคือจะปรากฏ Caché icon อยู่ที่ Program Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

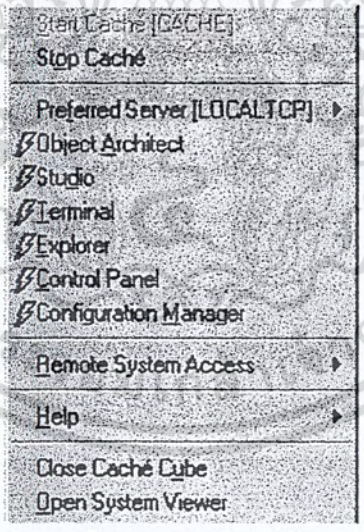
ส่วนประกอบต่างๆภายในโปรแกรม Caché

1. Caché Interface

หน้าจอหลักของ Caché คือ Caché Cube จาก Caché Cube สามารถที่จะเข้าถึงการใช้งานของ Caché Configuration และ Management Tools ได้

2. Caché Cube Icon

เมื่อทำการเริ่มโปรแกรม Caché Client บนระบบปฏิบัติการ Window 95 , 98 หรือ NT จะปรากฏรูป icon Caché Cube ที่ taskbar บน start menu bar เมื่อทำการ Double Click ที่ Caché Cube จะปรากฏ Caché System Viewer เพื่อให้สามารถดูแลกิจกรรมต่างๆของระบบได้ และถ้ากดคลิกขวาที่ Caché Cube จะปรากฏหน้าจอ Menu ของโปรแกรม Caché



รูปที่ ข-1 แสดงหน้าจอ Menu ของโปรแกรม Caché

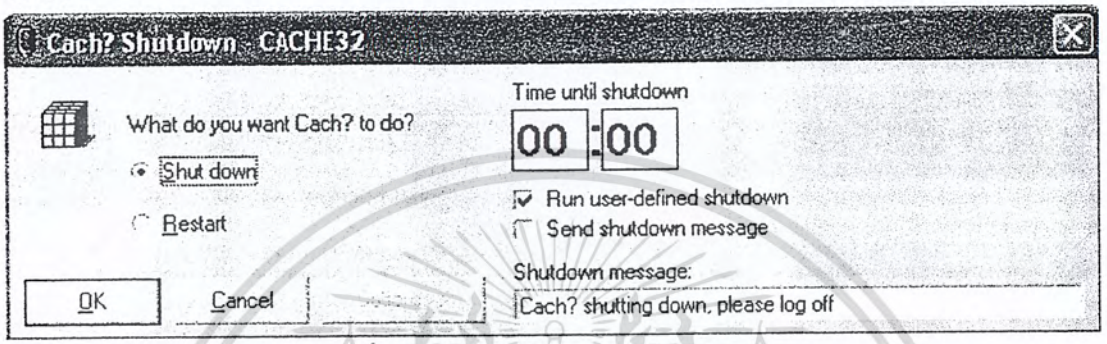
3. Start Caché (Configuration name : ในที่นี้คือ Cache)

เลือก Start Caché เพื่อทำการเริ่มโปรแกรม Caché แต่ถ้า Caché Server กำลังทำงานอยู่ ตัวเลือกนี้จะเป็นสีเทา เพื่อเป็นการบอกว่า ไม่สามารถใช้งานตัวเลือกนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Stop Caché

เลือก Stop Caché เพื่อทำการเลิกการใช้โปรแกรม Caché โดยภายในคำสั่ง Stop จะสามารถกำหนดระยะเวลาที่จะทำการเลิกการใช้โปรแกรม รวมไปถึงข้อความที่จะส่งไปถึง User ที่อยู่ในระบบ



รูปที่ ข-2 แสดงหน้าจอ Caché Shut Down

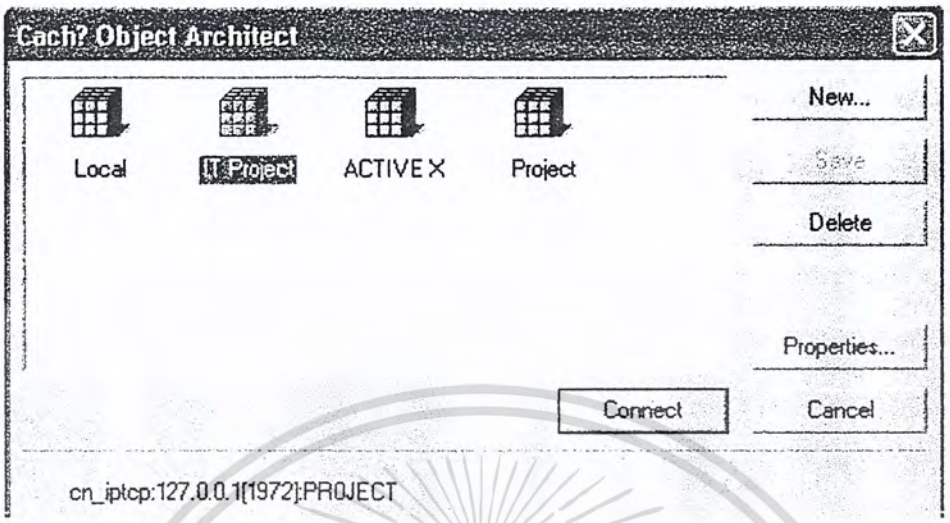
5. Preferred Server (Server Name : ในที่นี้คือ LOCALTCP)

ตัวเลือกนี้จะเป็นการอนุญาตให้เราสามารถทำการกำหนดชื่อเซิร์ฟเวอร์ (Server) ที่ใช้งานและสามารถทำการแก้ไขรายละเอียดต่างๆที่อยู่ภายในเซิร์ฟเวอร์ (Server) นั้นๆ โดย Caché จะไม่ทำการเชื่อมต่อแบบอัตโนมัติกับ Caché อื่นที่อยู่ภายในเครือข่ายเดียวกัน ถ้าต้องการทำการเชื่อมต่อกับเซิร์ฟเวอร์ (Server) อื่นๆ จะต้องการเชื่อมต่อโดยผ่านทาง Configuration Remote System Access

6. Object Architect

ตัวเลือกนี้เป็นการเสนอการใช้งาน Caché Architect ของเซิร์ฟเวอร์ที่ทำการอ้างถึง โดยจะมีเครื่องมือที่ใช้ในการสร้าง (Creating) , แก้ไข (Editing) , ลบ (Deleting) และ เรียบเรียง (Compiling) คลาส (Class) ต่างๆ

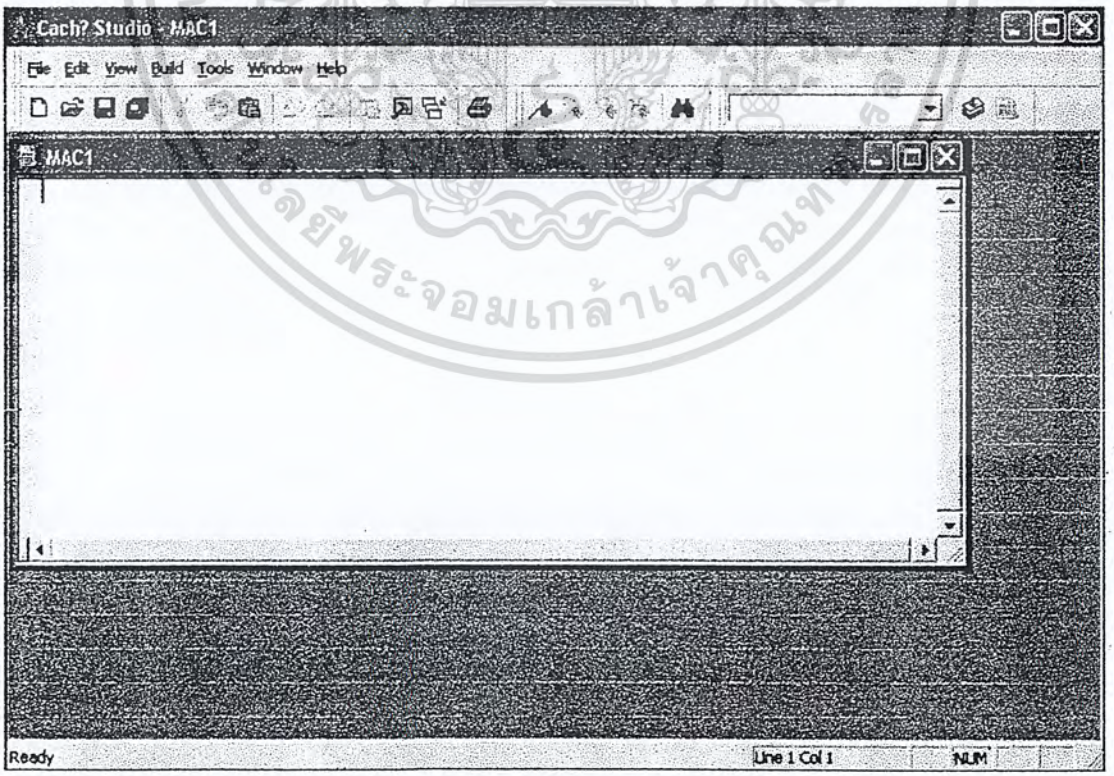
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-3 แสดงหน้าจอ Caché Object Architect

7. Studio

ตัวเลือกนี้เป็นการเสนอการใช้งาน Caché Studio ของเซิร์ฟเวอร์ที่ทำการอ้างอิง โดยจะมีเครื่องมือที่ใช้ในการสร้าง แก้ไข ลบ และเรียบเรียงCaché ObjectScript Routines

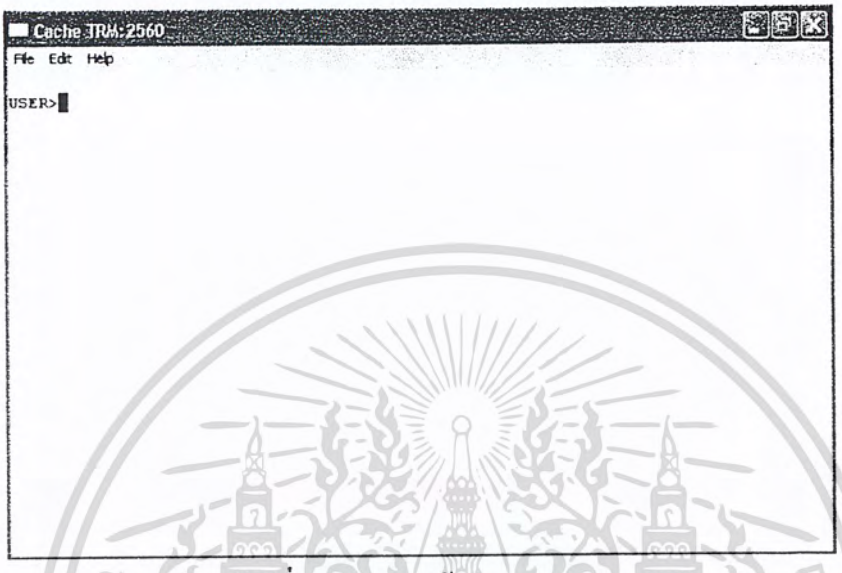


รูปที่ ข-4 แสดงหน้าจอ Caché Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Terminal

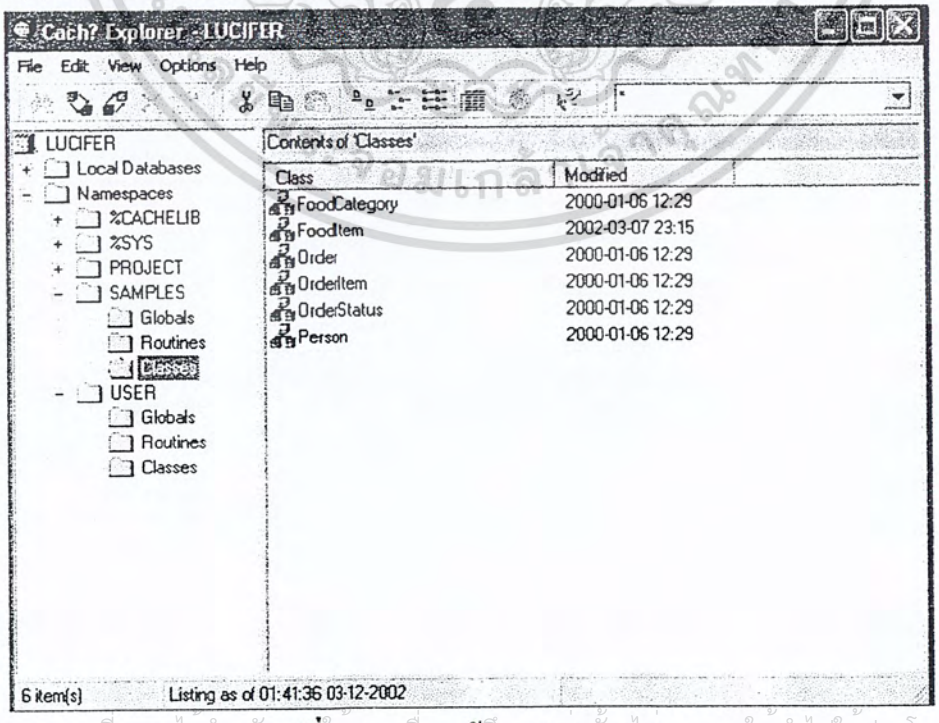
ตัวเลือกนี้เป็นการเสนอการใช้งาน Caché ObjectScript programmer ผ่านทางหน้าจอ window



รูปที่ ข-5 แสดงหน้าจอ Caché Terminal

9. Explorer

ตัวเลือกนี้เป็นการเสนอการใช้งาน Caché Explorer ของเซิร์ฟเวอร์ที่ทำการอ้างอิง โดยจะมีเครื่องมือที่ทำการเข้าถึง คลาส (Class) , โกบอล (Global) และการจัดการเกี่ยวกับฟังก์ชั่นของรูทีนต่างๆ (Routine management function)



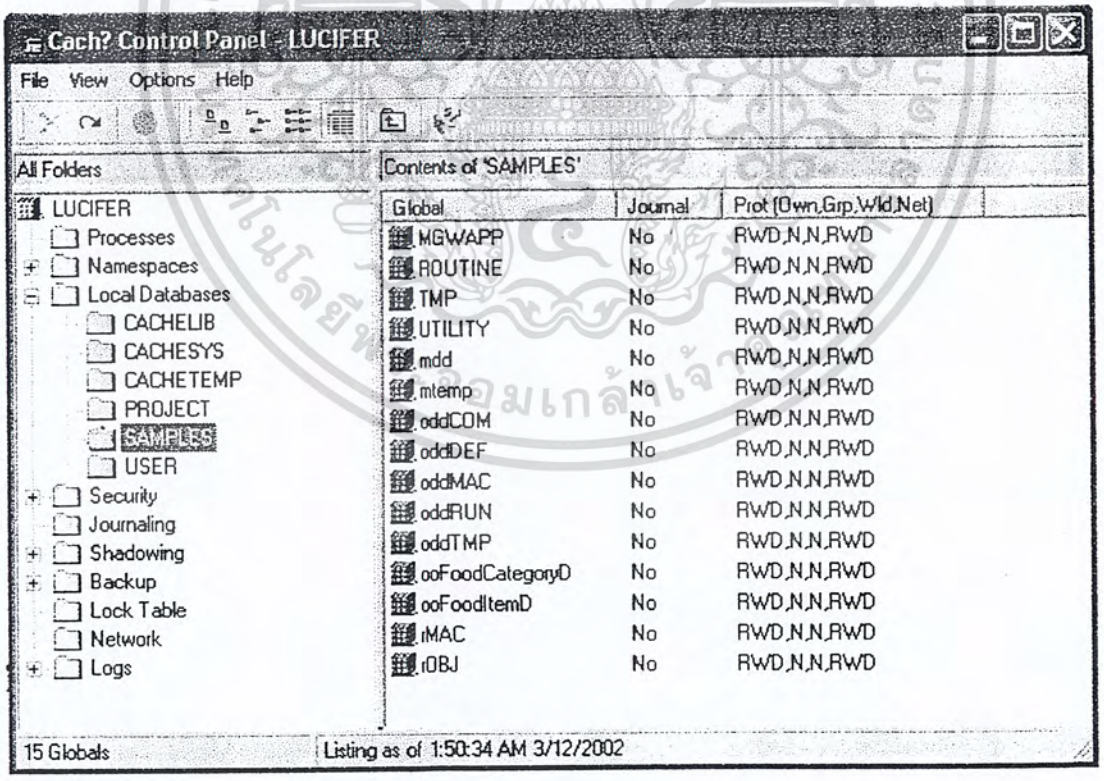
รูปที่ ข-6 แสดงหน้าจอ Cache Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Control Panel

ตัวเลือกนี้เป็นการเสนอการใช้งาน Caché Control Panel ของเซิร์ฟเวอร์ที่ทำการอ้างอิง โดย Caché Control Panel เป็นส่วนที่ใช้ในการเข้าถึงการใช้งานของส่วนต่างๆของระบบ เช่น

- Process management
- Database management
- User account and trusted application configuration
- Journal management
- Shadow system management
- Backup management
- Lock table manipulation
- Network management
- Log file viewing and manipulation

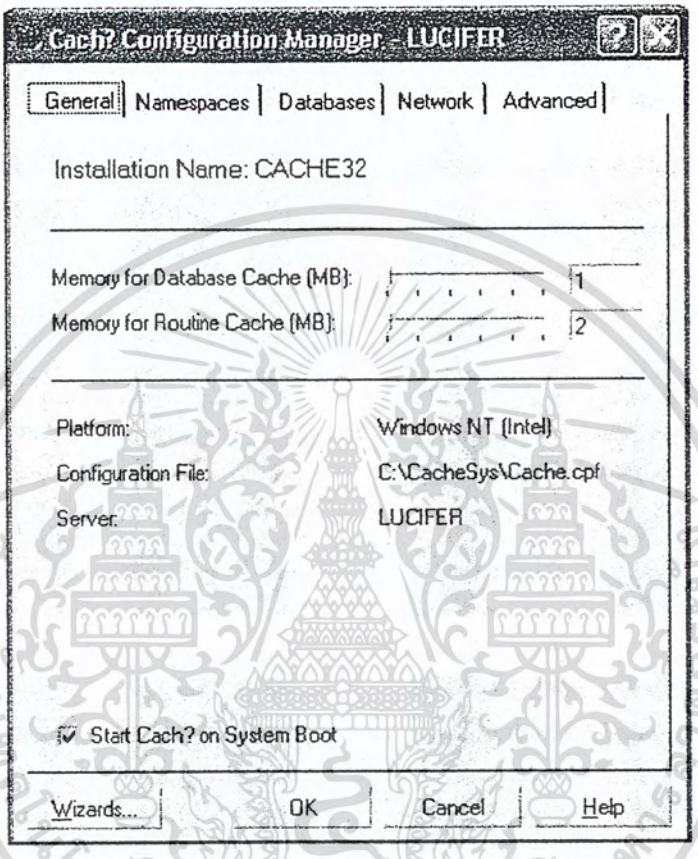


รูปที่ ข-7 แสดงหน้าจอ Caché Control Panel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. Configuration Manager

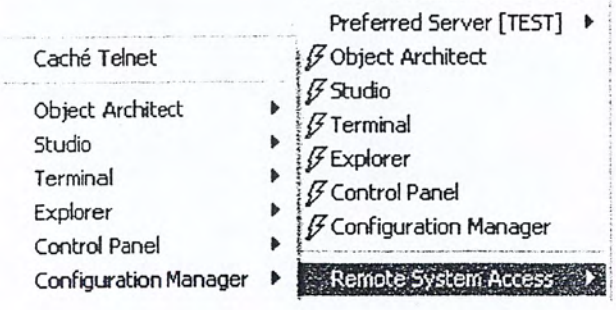
ตัวเลือกนี้เป็นการเสนอการใช้งาน Configuration Manager ซึ่งอนุญาตให้ทำการจัดการเกี่ยวกับคุณสมบัติต่างๆของ ระบบ (system) , namespace , ฐานข้อมูล (database) และเครือข่าย (network)



รูปที่ ข-8 แสดงหน้าจอ Caché Configuration Manager

12. Remote System Access

ตัวเลือกนี้เป็นการเสนอการใช้งาน Caché Tool ซึ่งเป็นอีกรูปแบบหนึ่งในการใช้งานส่วนอื่นๆของโปรแกรม Caché



รูปที่ ข-9 แสดงหน้าจอ Remote System Access

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. Help

ตัวเลือกนี้เป็นการเรียกใช้งาน Caché Help Menu ซึ่งจะมี Help Menu หลายรูปแบบ เช่น

- Caché Help จะมีลักษณะเป็น HTML Help และ Context-sensitive help
- Caché Documentation จะมีลักษณะเป็นรูปแบบเอกสาร HTML และ PDF
- Caché Tutorials จะมีลักษณะเป็นตัวอย่างของการสร้างโปรแกรมที่พัฒนาโดยโปรแกรม Caché โดยรูปแบบของโปรแกรมจะอยู่ในรูปแบบเอกสาร HTML
- About Caché แสดงข้อมูลของเวอร์ชันและลำดับในการพัฒนาของโปรแกรม Caché

14. Close Caché Cube

ตัวเลือกนี้เป็นการปิดการใช้งาน Caché Cube และย้าย Caché Cube ออกจาก System Tray แต่ไม่ได้ทำการ Shut down โปรแกรม Caché ถ้าต้องการจะเปิดโปรแกรม Caché ขึ้นมาใหม่ให้เปิดผ่านทาง Start Menu → Program → Caché

15. Open System Viewer

จะปรากฏเมื่อ Caché System Viewer ถูกย่อยให้มีขนาดเล็ก เลือกตัวเลือกนี้เพื่อใช้ในการแสดง Caché System Viewer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

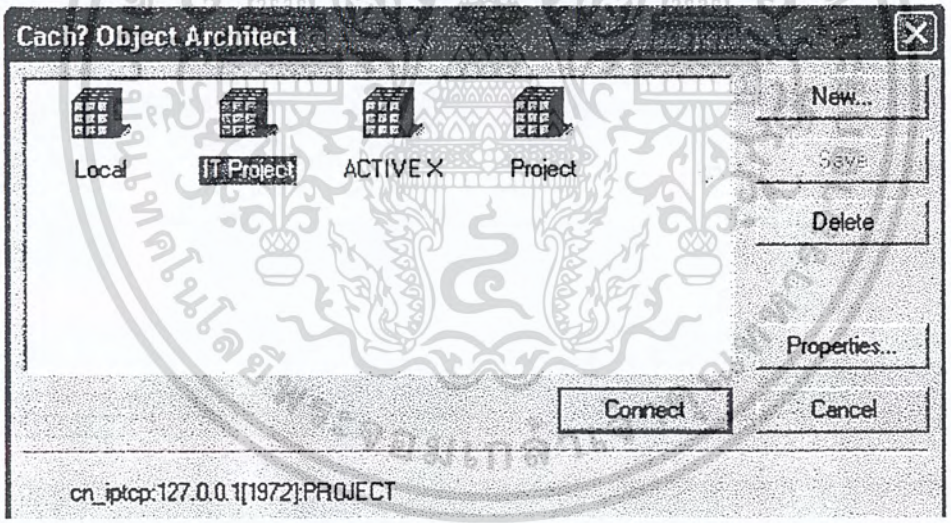
ภาคผนวก ค

ขั้นตอนของการสร้างคลาส (Class) ที่ใช้งานภายในโปรแกรม Caché

1. เปิดโปรแกรม Cache จะปรากฏ Cache Cube ที่ System Tray

รูปที่ ค-1 แสดงถึงรูป Cache Cube ที่ปรากฏอยู่ที่ System Tray ของ Window

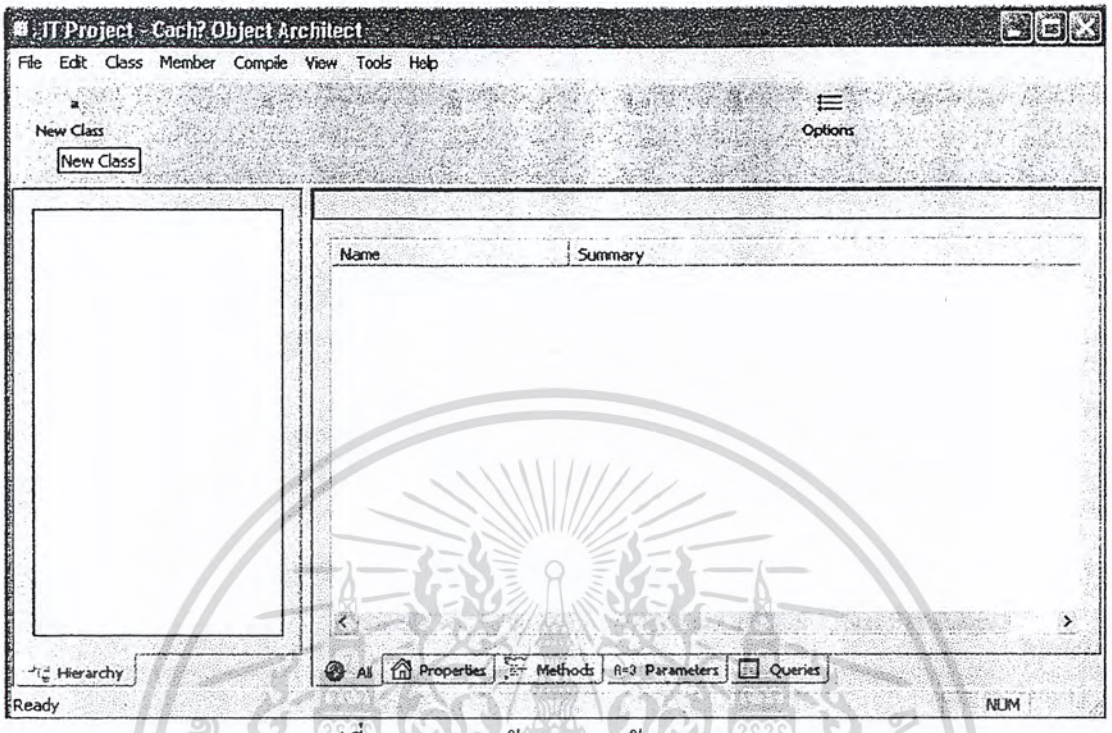
2. คลิกขวาที่ Cache Cube จากนั้นเลือกที่ Object Architect จะปรากฏหน้าจอ Cache Object Architect ให้ทำการเลือกเซิร์ฟเวอร์ที่จะใช้งาน จากนั้น เลือก Connect



รูปที่ ค-2 แสดงถึงหน้าจอ Cache Object Architect ที่ใช้เลือกเซิร์ฟเวอร์ที่ใช้งาน

3. เมื่อทำการเลือกที่ Connect แล้วจะปรากฏหน้าจอที่ใช้ในการสร้างคลาส (Class) ดังรูป ให้ทำการสร้างคลาส (Class) โดยทำการเลือกที่ New Class ที่ Tool Bar หรือเลือกที่ Class→ New Clrt + N

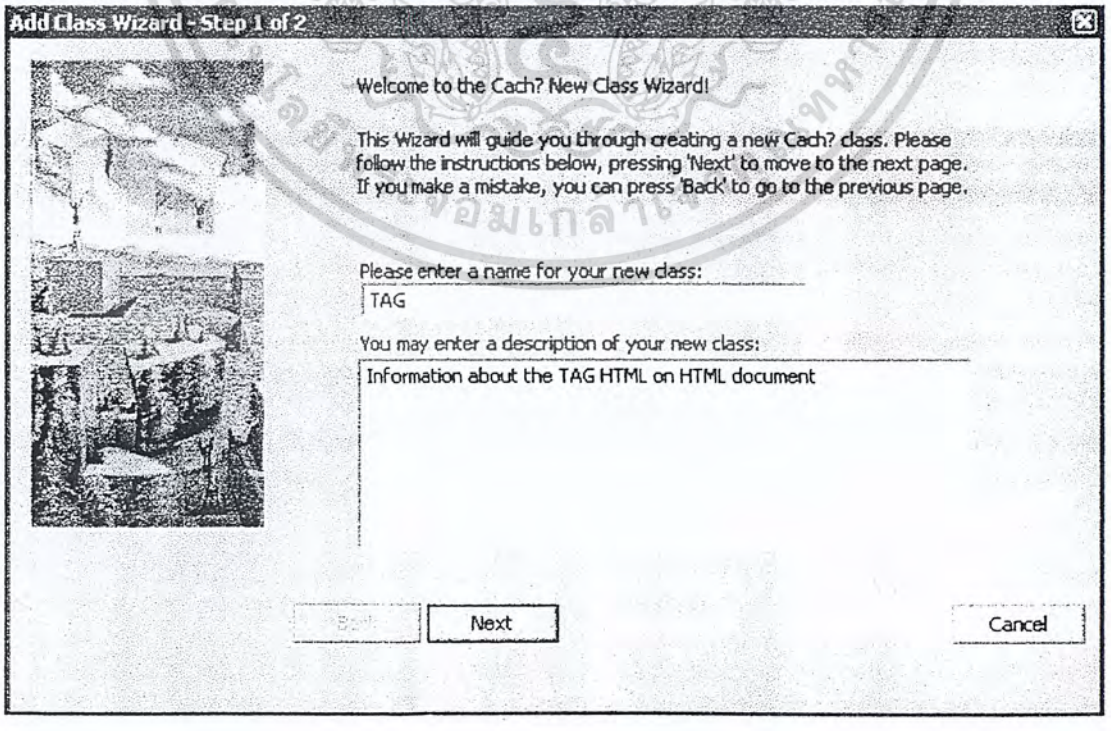
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-3 แสดงหน้าจอการสร้าง New Class

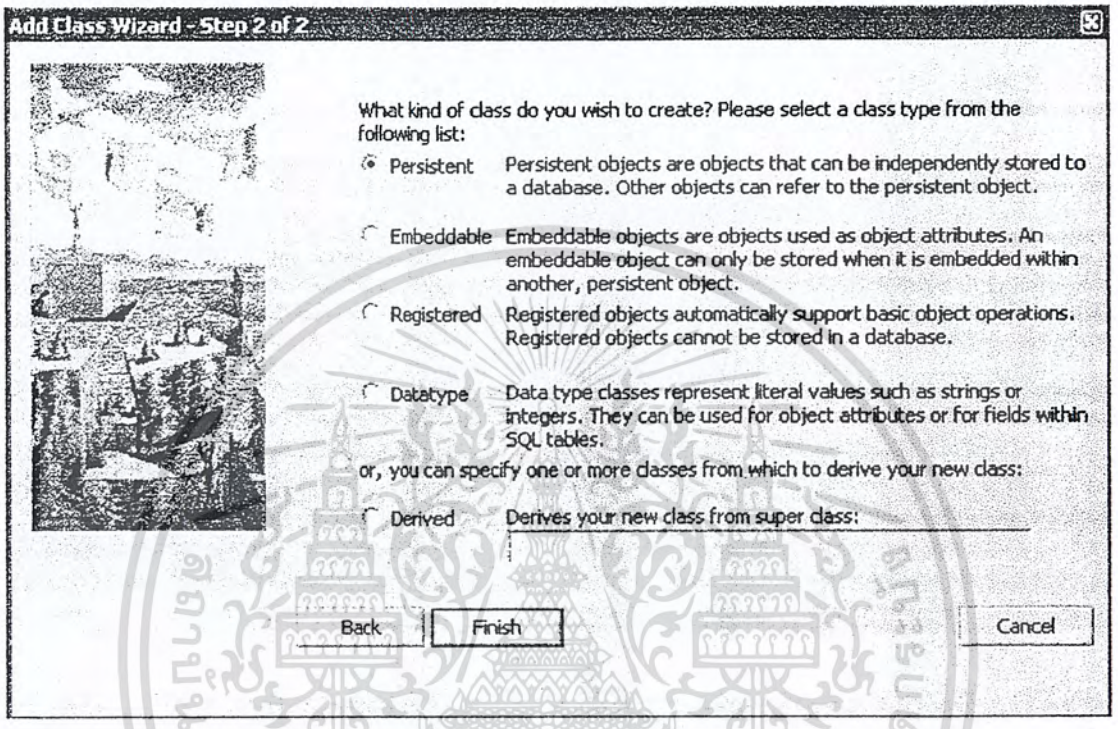
4. เมื่อทำการเลือกที่ New Class จะปรากฏหน้าจอ Wizard ที่ใช้ในการสร้างคลาส (Class) ขึ้นมาซึ่งจะให้ใส่ชื่อของคลาส (Class) และคำอธิบายเกี่ยวกับคลาส (Class) นั้น จากนั้นเลือก Next

รูปที่ ค-4 แสดงหน้าจอที่ใช้ในการตั้งชื่อและเขียนคำอธิบายเพิ่มเติม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

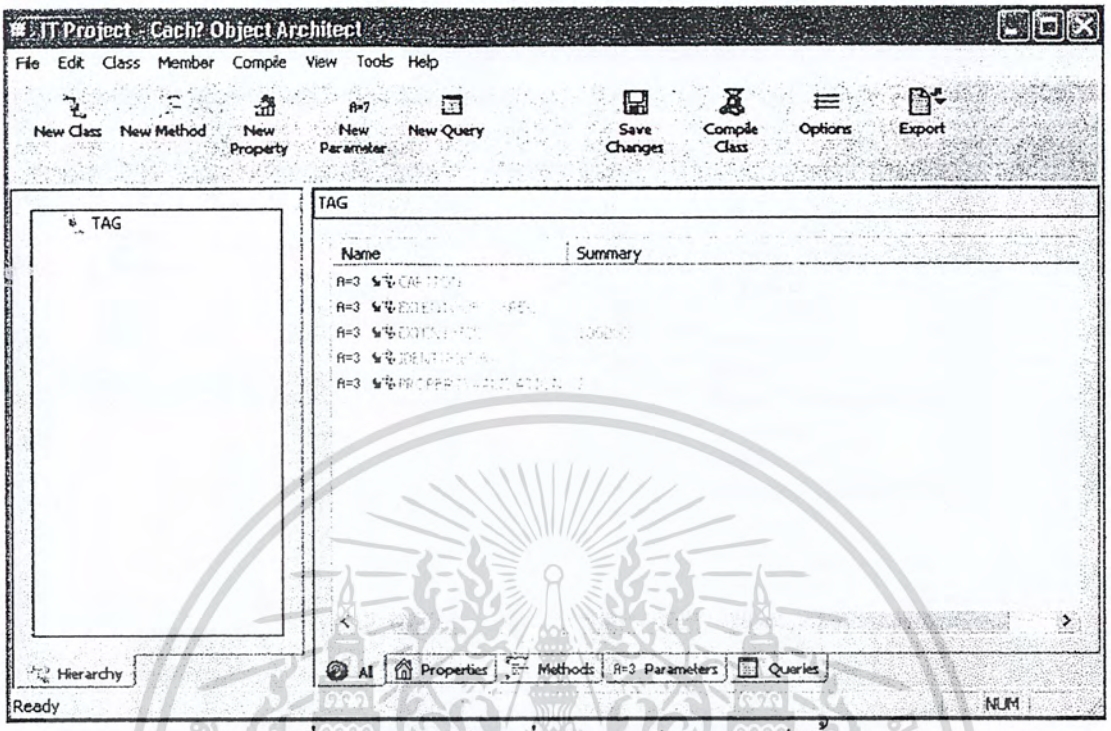
5. เมื่อทำการเลือกที่ Next จะปรากฏหน้าจอที่ให้เลือกประเภทของคลาส (Class) ที่ทำการสร้างขึ้นใหม่ ว่าเป็นคลาส (Class) ประเภทใด จากนั้นเลือก Next เป็นการสิ้นสุดการสร้างคลาส (Class) หนึ่งคลาส (Class)



รูปที่ ค-5 แสดงหน้าจอที่ใช้การเลือกประเภทของคลาส

6. เมื่อทำการสร้างคลาสเสร็จสิ้นแล้วจะปรากฏรูปคลาสที่หน้าจอ Hierarchy ทางซ้ายของจอภาพ และจะปรากฏคุณสมบัติเริ่มต้นทางขวาของจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

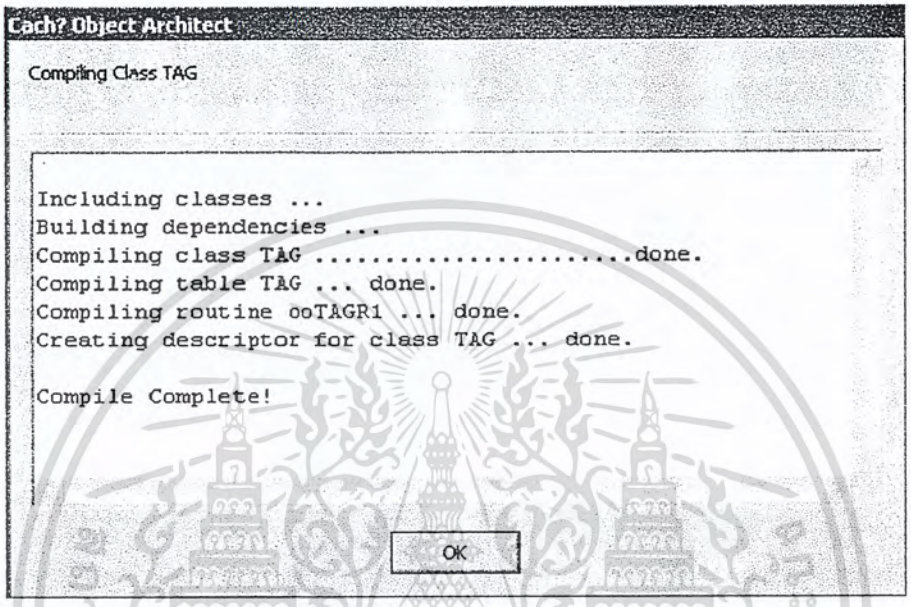


รูปที่ ค-6 แสดงหน้าจอเมื่อทำการสร้างคลาสเสร็จสิ้น

7. เมื่อทำการสร้างคลาส (Class) ขึ้นมาแล้ว ขั้นตอนต่อไปคือทำการสร้างส่วนประกอบต่างๆ ที่อยู่ภายในคลาส (Class) เช่นส่วนที่เป็น Properties , Methods , Parameters , Queries โดยใช้ Tool Bar ที่อยู่ทางด้านบนของหน้าจอ โดยจะยกตัวอย่างการสร้าง New Properties ให้ทำการเลือก Icon New Properties ที่ Tool Bar จะปรากฏหน้าจอ Wizard ที่ใช้ในการสร้าง New Properties ให้ทำการใส่ชื่อของ Property , ประเภท (Type) และคุณสมบัติอื่นๆเช่น Collection , Initial Value , Characteristics จากนั้นเลือก Ok

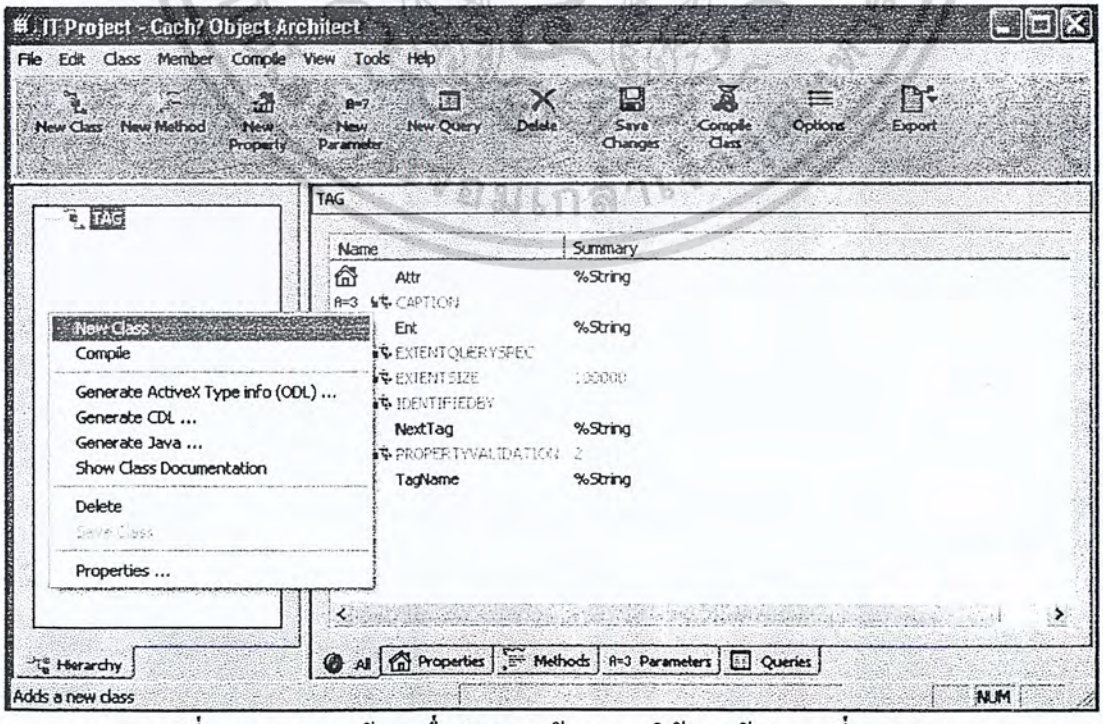
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เมื่อทำการเลือก Compile Class จะปรากฏหน้าจอ ถ้าคลาสที่สร้างขึ้นมีความถูกต้องตามที่โปรแกรมกำหนดจะปรากฏข้อความ Compile Complete แต่ถ้ามีความผิดพลาดจะปรากฏ Error ขึ้นเตือนผู้ใช้ว่าผิดพลาดในรูปแบบใด



รูปที่ ค-9 แสดงหน้าจอเมื่อทำการ Compile คลาสโดยไม่เกิด Error

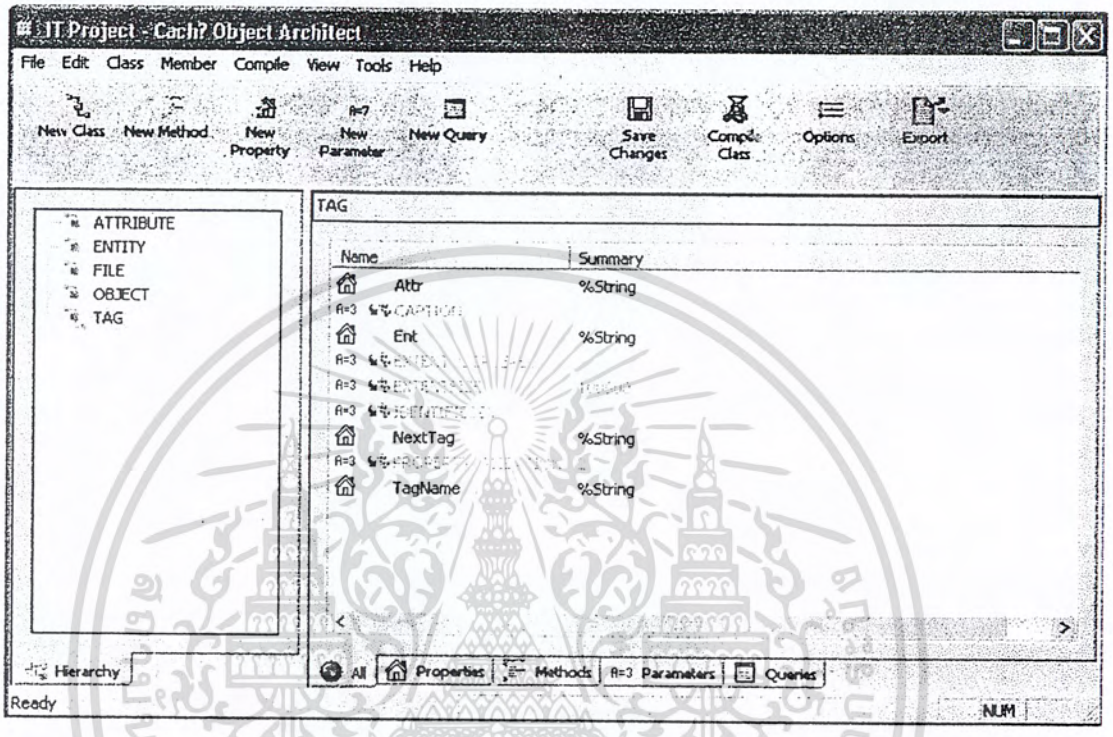
10. จากนั้นให้ทำซ้ำตั้งแต่ขั้นตอนที่ 3 เพื่อทำการสร้างคลาสขึ้นใหม่ให้ครบถ้วนตามที่ได้ออกแบบเพื่อให้ได้ฐานข้อมูลตามที่ผู้ใช้งานต้องการ



รูปที่ ค-10 แสดงหน้าจอเมื่อทำการสร้างคลาสให้ครบถ้วนตามที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. แสดงถึงฐานข้อมูลทั้งหมดที่ได้ทำการสร้างเสร็จสิ้นพร้อมนั้นมาใช้งานร่วมกับโปรแกรมประยุกต์ที่เราสร้างขึ้น



รูปที่ ค-11 แสดงหน้าจอเมื่อทำการสร้างคลาสทั้งหมดที่ได้ออกแบบที่ใช้กับโปรแกรมประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้