

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ฐานข้อมูลเชิงวัตถุบนเว็บเซอร์วิส

OBJECT ORIENTED DATABASE WEB SERVICE



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมคอมพิวเตอร์


คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์ฐานข้อมูลเชิงวัตถุบนเว็บเซอร์วิส

OBJECT ORIENTED DATABASE WEB SERVICE

ผู้จัดทำ

1. นางสาวฐนิตา สีนทะเกิด รหัสประจำตัว 48015335
2. นายนที มีเพียร รหัสประจำตัว 48015341
3. นายภูวดล คำอั้น รหัสประจำตัว 48015357



อาจารย์ที่ปรึกษา

(อ.บัณฑิต พัสยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ฐานข้อมูลเชิงวัตถุบนเว็บเซอร์วิส

นางสาวฐนิดา	สินทะเกิด	48015335
นายนที	มีเพียร	48015341
นายภูวคณ	คำอั้น	48015357
อ.บัณฑิต	พัศยา	อาจารย์ที่ปรึกษา
ปีการศึกษา 2550		

บทคัดย่อ

ในปัจจุบัน การพัฒนาโปรแกรมส่วนใหญ่ ยึดหลักการของโปรแกรมเชิงวัตถุมากขึ้น จนหลักการดังกล่าวได้เริ่มเข้ามามีบทบาทสำคัญ ในการพัฒนาระบบฐานข้อมูลเชิงวัตถุ (Object – Oriented Database) โดยการนำแนวคิดเชิงวัตถุเข้ามาใช้กับระบบฐานข้อมูล ซึ่งระบบจัดการฐานข้อมูลเชิงวัตถุสามารถรองรับการใช้งานข้อมูลที่มีความซับซ้อน ได้มีประสิทธิภาพมากกว่า การใช้ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

โครงการนี้ได้ทำการศึกษากระบวนการระบบฐานข้อมูลเชิงวัตถุ โดยได้เลือกใช้ระบบจัดการฐานข้อมูลแคชเช่(Caché) และได้ทำการศึกษาวิธีการติดต่อใช้งานระบบจัดการฐานข้อมูลเชิงวัตถุโดยใช้ภาษาโปรแกรมจาวาในการติดต่อกับฐานข้อมูล นอกเหนือจากนั้นปัญญานิพนธ์เล่มนี้ยังอธิบายถึงรายละเอียดของเว็บเซอร์วิสทั้งหมด ซึ่งประกอบไปด้วย การทำงานของเว็บเซอร์วิส,สถาปัตยกรรมของเว็บเซอร์วิส, เทคโนโลยีที่ใช้ในเว็บเซอร์วิส ซึ่งได้แก่ XML, SOAP, WSDL และUDDI โดยดูว่าในแต่ละเทคโนโลยีที่นำมาใช้นั้นมีการนำมาใช้ในลักษณะใดและนำเอาหลักการต่างๆ ดังที่ได้กล่าวมาแล้วมาประยุกต์ใช้กับระบบการชำระค่าปรับทางด้านการจราจร โดยมี แอปพลิเคชันที่เกี่ยวข้องคือ สำนักงานตำรวจแห่งชาติ กรมการขนส่งทางบก กรมการปกครองและธนาคาร โดยจะมีการเซอร์วิสข้อมูลกันตามหลักการที่กล่าวไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OBJECT ORIENTED DATABASE WEB SERVICE

Ms.Thanita Sintakird 48015335

Mr.Natee Meepian 48015341

Mr.Puwadon Kum-Un 48015357

Mr.Bundit Pasaya Advisor

Academic Year 2007

ABSTRACT

Nowadays almost programming development adheres object oriented programming principle more until it begins to have impact on object database by using object oriented concept for database which supports complex data better than relational database.

In this project, we learn object - oriented database concept by using Caché database and using OODB base on Java programming Concept Moreover, this thesis also provides other deep knowledge about web-service. Topics are divided into three major parts that are web service process, web-service architecture, and web service technology. For web service technology, This thesis will explain in details about the application of four major technologies, which are XML, SOAP, WSDL and UDDI. Lastly, a case study of the system of Traffic Fine System, compose Application Royal Thai Police, Application Department of Land Transport, Application Department of Provincial Administration and Bank applied all mentioned principles into the system, will also be presented at the end of the thesis.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลงได้เป็นอย่างดี ด้วยคำแนะนำ คำปรึกษา และความช่วยเหลือจาก อาจารย์ บัณฑิต พัสยา ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการที่เอาใจใส่และเป็นห่วงเสมอมา ซึ่งขอกราบขอบพระคุณเป็นอย่างสูง และ ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาให้

ขอขอบคุณเพื่อนๆ พี่ๆน้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา และครอบครัวอันเป็นที่เคารพรักยิ่ง ที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้สามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลงด้วยดี คุณค่าและประโยชน์อันพึงมาจากปริญญาบัตรฉบับนี้ ขอมอบแด่ผู้มีพระคุณทุกท่านและผู้ที่เกี่ยวข้องศึกษา

ฐนิตา สินทะเกิด
นที มีเพียร
ภูวดล คำอัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและความเป็นมาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตของโครงการ.....	2
1.5 วิธีการดำเนินงาน.....	2
1.6 ส่วนประกอบของรายงาน.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับเว็บเซอร์วิส.....	4
2.1 Service-Oriented Architecture (SOA)	5
2.1.1 แนะนำ SOA.....	5
2.1.2 จุดเด่นของ SOA.....	5
2.1.3 ประโยชน์ของ SOA	5
2.1.4 รูปแบบการทำงานของ SOA.....	7
2.2 Extensible Markup Language (XML).....	8
2.2.1 จุดเริ่มต้นของภาษา XML.....	8
2.2.2 บทบาทของ XML ในปัจจุบัน.....	9
2.2.3 เป้าหมายของมาตรฐาน XML.....	9
2.2.4 ส่วนประกอบของเอกสาร XML.....	10
2.2.4.1 กฎพื้นฐานในการเขียน XML (Well-Formed)	10
2.2.4.2 องค์ประกอบของเอกสาร XML.....	10
2.2.4.3 XML Element, Attribute และ Entity.....	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3 Simple Object Access protocol (SOAP)	14
2.3.1 แนะนำ SOAP.....	14
2.3.1.1 Message Envelope.....	14
2.3.1.2 กฎของการ Encoding.....	15
2.3.1.3 ระเบียบแบบแผนของ Remote Procedure Call (RPC).....	15
2.3.1.4 การใช้ Transport Protocol ร่วมกับ SOAP.....	15
2.3.2 การส่งข้อมูลผ่าน SOAP	15
2.3.3 SOAP Model.....	16
2.3.3.1 แนะนำ SOAP Model	16
2.3.3.2 โครงสร้างของ SOAP Message.....	16
2.3.4 SOAP Encoding.....	17
2.3.5 รูปแบบการติดต่อสื่อสารของ SOAP.....	18
2.3.5.1 SOAP RPC.....	18
2.3.5.2 SOAP Messaging.....	18
2.3.6 SOAP กับ HTTP.....	19
2.4 Web Service Description Language (WSDL).....	20
2.4.1 แนะนำ WSDL	20
2.4.2 กระบวนการร้องขอบริการ.....	20
2.4.3 โครงสร้างของเอกสาร WSDL.....	21
2.4.4 WSDL Binding.....	22
2.4.4.1 อีลีเมนต์ soap:binding.....	22
2.4.4.2 อีลีเมนต์ soap:operation	23
2.4.4.3 อีลีเมนต์ soap:body.....	23
2.4.4.4 อีลีเมนต์ soap:address.....	23
2.5 Universal Description Discovery and Integration (UDDI).....	23
2.5.1 แนะนำ UDDI.....	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5.1.1 ข้อมูลทางธุรกิจและการจัดกลุ่มข้อมูล.....	24
2.5.1.2 ข้อกำหนดของ UDDI.....	24
2.5.1.3 การลงทะเบียน.....	25
2.5.2 คำศัพท์ที่เกี่ยวข้องกับ UDDI.....	25
2.5.3 การประกาศ Service (Service Publication).....	26
2.5.3.1 การสร้างและปรับปรุง Service.....	26
2.5.3.2 การลบ Service.....	27
2.5.4 การค้นหา Service (Service Discovery).....	27
2.5.4.1 Information Browsing.....	28
2.6 ข้อกำหนดเพิ่มเติมที่สำคัญของ Web Service.....	28
2.6.1 WS-Addressing.....	28
2.6.2 WS-Reliable Messaging.....	29
2.6.3 ความเกี่ยวเนื่องกัน (Correlation).....	30
2.6.4 WS-Policy.....	31
2.6.5 WS-Metadata Exchange.....	32
2.6.6 WS-Security.....	32
2.6.7 WS-Notification.....	33
บทที่ 3 ทฤษฎีที่เกี่ยวข้องกับฐานข้อมูลเชิงวัตถุ.....	34
3.1 องค์ประกอบของการพัฒนาโปรแกรมเชิงวัตถุ.....	34
3.1.1 อ้างอิงออบเจกต์ (OID).....	34
3.1.2 คุณสมบัติ (Property).....	34
3.1.3 เมธอด (Method).....	34
3.1.4 เมสเสจ (Message).....	34
3.1.5 คลาส (Class) และ อินสแตนซ์ (Instance).....	35
3.2 แนวคิดของการพัฒนาโปรแกรมเชิงวัตถุ.....	35
3.2.1 การห่อหุ้ม (Encapsulation).....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2.2 การสืบทอด (Inheritance).....	36
3.2.3 ภาวะหลายรูปแบบ (Polymorphism).....	36
3.3 รายละเอียดการทำงานของฐานข้อมูลเชิงวัตถุ.....	37
3.3.1 โครงสร้างข้อมูล.....	37
3.3.2 ภาษาที่ใช้ในการจัดการฐานข้อมูล.....	38
3.3.3 กลไกที่ใช้ในการเข้าถึงข้อมูล.....	38
3.3.4 ประโยชน์และข้อดีของฐานข้อมูลเชิงวัตถุ.....	38
3.4 OODBMS (Object-Oriented Database Management System).....	39
3.4.1 สถาปัตยกรรมของการใช้ OODBMS.....	39
3.4.2 Characteristics of Object-Oriented Databases in Depth.....	39
3.5 ODMG.....	40
3.5.1 โมเดลออบเจกต์ (Object Model).....	40
3.5.2 ภาษาในการคิวรีออบเจกต์ (Object Query Language).....	40
3.5.3 การใช้ร่วมกับภาษา C++ (C++ Language Binding).....	40
3.5.4 การใช้ร่วมกับภาษา Smalltalk (Smalltalk Language Binding).....	41
3.5.5 การใช้ร่วมกับภาษาจาวา (Java Language Binding).....	41
บทที่ 4 การวิเคราะห์และออกแบบระบบ.....	42
4.1 บทนำ.....	42
4.2 สิ่งที่ต้องการในระบบ.....	42
4.3 Use Case Diagram ของระบบ.....	43
4.4 Web service Model.....	44
4.5 Class Diagram ของระบบ.....	46
4.6 Sequence Diagram ของระบบ.....	50
บทที่ 5 การใช้งานระบบ.....	50
5.1 เว็บกรมการปกครอง.....	52
5.2 เว็บกรมการขนส่งทางบก.....	54
5.3 เว็บสำนักงานตำรวจแห่งชาติ.....	57
5.4 เว็บธนาคาร.....	62

เอกสารนี้เป็นเอกสารเรียนควรใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านธุรกิจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 6 บทวิจารณ์และสรุป.....	65
6.1 บทสรุป.....	65
6.2 วิจารณ์สิ่งที่ได้จากโครงการ.....	65
6.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข.....	65
บรรณานุกรม.....	66
ภาคผนวก ก. การติดตั้งและใช้งานฐานข้อมูล Caché.....	67
ขั้นตอนการติดตั้งระบบฐานข้อมูล Caché.....	67
วิธีการใช้งาน โปรแกรม Caché Studio.....	71
ภาคผนวก ข. การติดต่อใช้งานฐานข้อมูล Caché กับ JAVA.....	86
การติดต่อใช้งานฐานข้อมูล Caché กับ ภาษา JAVA.....	86
เพิ่มไลบรารีของ Caché ให้กับ NetBeans.....	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
2.1 ความสัมพันธ์ระหว่าง UDDI, WSDL, SOAP และ Web Service.....	4
2.2 การขอบริการระหว่างเว็บเซอร์วิส.....	4
2.3 SOA Model.....	7
2.4 องค์ประกอบของเอกสาร XML.....	11
2.5 ส่วนประกอบของ Document Element.....	12
2.6 การส่งคำร้องขอ (Request) ไปยังผู้ให้บริการ.....	15
2.7 การส่งคำตอบรับ (Response) ไปยังผู้ร้องขอบริการ.....	15
2.8 โครงสร้างของ SOAP Message.....	16
2.9 ตัวอย่าง SOAP Message.....	18
2.10 กระบวนการขอบริการ.....	20
2.11 Abstract Description และ Concrete Description ของเอกสาร WDSL.....	21
2.12 โครงสร้างของเอกสาร WSDL.....	21
2.12 รูปแบบของ SOAP Protocol.....	22
2.13 สถาปัตยกรรมของ UDDI Registry.....	25
2.14 WS-Addressing ที่ทำให้ Message กลายเป็นหน่วยอิสระในการติดต่อสื่อสาร.....	28
2.15 กระบวนการของ Reliable Messaging.....	29
2.16 การเก็บสถานภาพการแลกเปลี่ยนข้อมูลไว้ใน Message.....	30
2.17 รูปแบบของ Policy.....	31
2.18 Service A แจ้งให้ Service B ส่ง Notification.....	32
2.19 Service B ส่ง “การแจ้งเตือน (Notification)”.....	32
3.1 การติดต่อกับออบเจกต์.....	34
3.2 การรวมแนวคิดเชิงวัตถุเข้ากับระบบจัดการฐานข้อมูล.....	38
4.1 Use Case Diagram ของระบบ.....	43
4.2 การบริการข้อมูลซึ่งกันและกันของระบบ.....	44
4.3 Class Diagram ของระบบสำนักงานตำรวจแห่งชาติ.....	44
4.4 Class Diagram ของระบบกรมขนส่งทางบก.....	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

รูปที่	หน้า
4.5 Class Diagram ของระบบธนาคาร.....	48
4.6 Class Diagram ของระบบกรมการปกครอง.....	49
4.7 Sequence Diagram การตรวจสอบและชำระค่าปรับ.....	50
4.8 Sequence Diagram การบันทึกข้อมูลความผิด.....	51
5.1 หน้าล็อกอินเว็บกรมการปกครอง.....	52
5.2 หน้ากรอกข้อมูลทะเบียนราษฎร์.....	52
5.3 หน้าตรวจสอบข้อมูลทะเบียนราษฎร์.....	53
5.4 หน้าเพิ่มข้อมูลใบขับขี่.....	54
5.5 หน้าตรวจสอบข้อมูลใบขับขี่กรณีปกติ.....	54
5.6 หน้าตรวจสอบข้อมูลใบขับขี่กรณีถูกอายัด.....	55
5.7 หน้าเพิ่มข้อมูลยานพาหนะ.....	55
5.8 หน้าการเสียภาษี.....	56
5.9 หน้าตรวจสอบข้อมูลยานพาหนะ.....	56
5.10 หน้าล็อกอินเว็บสำนักงานตำรวจแห่งชาติ.....	57
5.11 หน้าตรวจสอบก่อนกรอกข้อมูล.....	58
5.12 หน้าตรวจสอบข้อมูลสถานะไม่ปกติ.....	58
5.13 หน้ากรอกข้อมูลความผิด.....	59
5.14 Popup คู่มือใบขับขี่กับข้อมูลยานพาหนะ.....	60
5.15 หน้าตรวจสอบข้อมูลความผิด.....	60
5.16 ใบสั่งความผิดทางการจราจร.....	61
5.17 หน้าเพิ่มข้อมล่าวหา.....	61
5.18 หน้าล็อกอินเว็บธนาคาร.....	62
5.19 หน้าตรวจสอบการชำระค่าปรับ.....	62
5.20 หน้าชำระค่าปรับ.....	63
5.21 หน้าชำระค่าปรับกรณีชำระไม่ได้.....	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่	หน้า
ก.1 ไฟล์ติดตั้งของ Caché.....	67
ก.2 ไดอะล็อกสำหรับเลือกภาษา.....	67
ก.3 หน้าจอเริ่มการติดตั้ง	68
ก.4 เลือกที่อยู่สำหรับติดตั้ง.....	68
ก.5 รายละเอียดข้อมูลต่างๆ ในการติดตั้ง.....	69
ก.6 ระบบกำลังทำการติดตั้ง.....	69
ก.7 เสร็จสิ้นกระบวนการติดตั้ง.....	70
ก.8 Caché กำลังทำงานอยู่.....	70
ก.9 เว็บเพจคู่มือการใช้งาน Caché.....	71
ก.10 หน้าตาโปรแกรม Caché Studio.....	71
ก.11 หน้าต่าง Log on เข้าสู่ระบบ	72
ก.12 ไดอะล็อกเลือก Namespace.....	72
ก.13 หน้าโปรแกรม Caché Studio.....	73
ก.14 การเลือกเมนูเพื่อเริ่มต้นใช้งาน.....	73
ก.15 ไดอะล็อกเลือกไฟล์ชนิดต่างๆที่ต้องการใช้งาน.....	74
ก.16 Wizard ในการสร้างคลาส.....	74
ก.17 การเลือกชนิดของคลาส.....	75
ก.18 โค้ดหลังจากที่เสร็จสิ้น Wizard การสร้างคลาส.....	75
ก.19 การเลือกเมนูสร้าง Property	76
ก.20 Wizard ในการสร้าง Property.....	76
ก.21 ไดอะล็อกเลือกชนิดของ Property.....	77
ก.22 โค้ดที่ Wizard สร้างให้.....	77
ก.23 การเลือกเมนูสร้างเมธอด.....	78
ก.24 Wizard ในการสร้างเมธอด.....	78
ก.25 การเลือกชนิดข้อมูลที่คืนค่าออกมา และ กำหนดค่าพารามิเตอร์.....	79
ก.26 การเลือกภาษาที่ใช้ในการเขียนเมธอด.....	79
ก.27 โค้ดที่ Wizard สร้างมาให้.....	80
ก.28 การเลือกเมนูสร้าง Query.....	80

เอกสารนี้เป็นเอกสารตัวอย่างเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ใดๆ
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่	หน้า
ก.29 Wizard ในการสร้าง Query.....	81
ก.30 ขั้นตอนการกำหนดพารามิเตอร์และชนิดข้อมูล.....	81
ก.31 การเลือกข้อมูลที่ต้องการให้แสดงออกมา.....	82
ก.32 การสร้างเงื่อนไข.....	82
ก.33 การจัดเรียงข้อมูล.....	83
ก.34 โค้ดที่ Wizard สร้างให้.....	83
ก.35 การเลือกเมนูสร้าง Projection.....	84
ก.36 Wizard ในการสร้าง Project.....	84
ก.37 การกำหนดที่อยู่ของ Source Code ภาษาจาวา.....	85
ก.38 โค้ดที่ Wizard สร้างมาให้.....	85
ข.1 เมนูเลือก Properties ของ Project.....	86
ข.2 ไดอะล็อก Properties ของ Project.....	87
ข.3 การเพิ่มไลบรารี Cache ให้กับ Project	87
ข.4 ไลบรารีของ Cache ที่เราได้เพิ่มไป.....	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและความเป็นมาของโครงการ

เนื่องจาก ในปัจจุบันเทคโนโลยีในด้านระบบสารสนเทศได้เข้ามามีบทบาทในการดำเนินกิจกรรมและ การทำงานด้านต่างๆ มากมาย หนึ่งในเทคโนโลยีนั้นก็คือ เว็บเซอร์วิส (Web Services) เป็นเทคโนโลยีด้านการ “บริการ” ที่เป็นระบบซอฟต์แวร์ ออกแบบมาเพื่อสนับสนุนการทำงาน ระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ผ่านระบบเครือข่าย โดยภาษาที่ใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์ คือภาษาเอ็กซ์เอ็มแอล (XML) โดยการติดต่อสื่อสารจะไม่ขึ้นกับแพลตฟอร์ม (Platform) และรูปแบบของภาษาด้วย การจัดการกับข้อมูลจึงมีความซับซ้อนมากขึ้น ดังนั้นจึงมีความจำเป็นในเรื่องของการจัดการกับข้อมูลเพิ่มมากขึ้นด้วย สาเหตุเหล่านี้ จึงเป็นที่มาของการพัฒนาระบบการจัดการฐานข้อมูลเชิงวัตถุ (Object-Oriented Database) ขึ้นมา เพื่อรองรับกับข้อมูลที่มีหลายรูปแบบขึ้น โดยสามารถจัดการกับข้อมูลที่มีความซับซ้อน ได้มากกว่า ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ในส่วนของโครงการนี้ได้ทำการศึกษาระบบการจัดการฐานข้อมูลเชิงวัตถุ (Object-Oriented Database) และนำไปประยุกต์ใช้กับเทคโนโลยีเว็บเซอร์วิส (Web Services) ระบบจัดการฐานข้อมูลที่เลือกใช้คือ “Cache” โดยใช้ภาษาในตัวของ Cache เองทำเว็บเซอร์วิสการติดต่อกับฐานข้อมูลจะใช้ภาษาจาวา (JAVA) และใช้ ภาษา Java Server Pages (JSP) ในการพัฒนาระบบเว็บไซต์ ในที่นี้จะจำลองระบบการชำระค่าปรับทางด้านการจราจรมาเป็นกรณีศึกษา คือ ผู้กระทำผิดกฎจราจรสามารถเสียค่าปรับผ่านทางธนาคารสาขาไหนก็ได้ โดยไม่ต้องเสียเวลาไปที่สถานีตำรวจ ส่วนทางด้านตำรวจสามารถตรวจสอบข้อมูลผู้กระทำผิดจากกรมการปกครอง และข้อมูลยานพาหนะจากกรมการขนส่งทางบกได้ว่ามีข้อมูลจริงหรือไม่ ก่อนที่จะบันทึกข้อมูลความผิด ส่วนการอายัดใบขับขี่สามารถทำได้เลยในกรณีคะแนนเกินที่กำหนด

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาการใช้งานฐานข้อมูลเชิงวัตถุ (Object-Oriented Database)
- 1.2.2 เพื่อศึกษาหลักการทำงาน, สถาปัตยกรรมและส่วนประกอบต่างๆของ เว็บเซอร์วิส
- 1.2.3 เพื่อศึกษาการสร้างเว็บเซอร์วิสรวมถึงวิธีการเรียกใช้เว็บเซอร์วิส
- 1.2.4 เพื่อนำหลักการของเว็บเซอร์วิสมาประยุกต์ใช้งานร่วมกับฐานข้อมูลเชิงวัตถุ
- 1.2.5 พัฒนาระบบการชำระค่าปรับทางด้านการจราจรเพื่อทดลองเรียกใช้เว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 ได้รับความรู้เกี่ยวกับแนวคิดของฐานข้อมูลเชิงวัตถุ
- 1.3.2 ได้รับความรู้ความเข้าใจเกี่ยวกับการใช้ภาษาจาวาในการติดต่อฐานข้อมูลเชิงวัตถุ
- 1.3.3 ได้รับความรู้เกี่ยวกับหลักการและการทำงานของเว็บเซอร์วิส
- 1.3.4 สามารถวิเคราะห์และออกแบบฐานข้อมูลเชิงวัตถุได้
- 1.3.5 สามารถพัฒนาระบบเว็บเซอร์วิสที่ประยุกต์ร่วมกับฐานข้อมูลเชิงวัตถุได้

1.4 ขอบเขตของโครงการ

โครงการนี้จะศึกษาเกี่ยวกับรายละเอียดของเว็บเซอร์วิสทั้งหมด ซึ่งประกอบไปด้วย การทำงานของเว็บเซอร์วิส, สถาปัตยกรรมของเว็บเซอร์วิส, เทคโนโลยีที่ใช้ในเว็บเซอร์วิส อันได้แก่ XML, SOAP, WSDL และ UDDI และศึกษาในส่วนของระบบฐานข้อมูลเชิงวัตถุโดยใช้ซอฟต์แวร์ Caché จากนั้นนำหลักการทั้งหมดไปพัฒนาระบบด้วยภาษาจาวา (JAVA) โดยระบบดังกล่าวเป็นระบบที่จำลองการชำระค่าปรับทางด้านการจราจร ส่วนเว็บแอปพลิเคชันที่เกี่ยวข้องกับระบบจะสร้างจาก ภาษา Java Server Pages (JSP)

1.5 วิธีการดำเนินการงาน

- 1.5.1 ศึกษาทฤษฎีและหลักการเกี่ยวกับเว็บเซอร์วิส
- 1.5.2 ศึกษาหลักการและแนวคิดของระบบฐานข้อมูลเชิงวัตถุ
- 1.5.3 ศึกษาการพัฒนาเว็บแอปพลิเคชันโดยใช้ภาษาเจเอสพี (JSP)
- 1.5.4 ศึกษาการพัฒนาเว็บเซอร์วิสด้วยโปรแกรม Caché Studio
- 1.5.5 ศึกษาการใช้งานฐานข้อมูล Caché
- 1.5.6 ศึกษาการติดต่อฐานข้อมูลเชิงวัตถุด้วยภาษาจาวา (JAVA)
- 1.5.7 ศึกษาการใช้งานเว็บเซอร์วิสติดต่อกับฐานข้อมูล
- 1.5.8 วิเคราะห์และออกแบบระบบ
- 1.5.9 ศึกษาการวิเคราะห์และออกแบบเชิงวัตถุตามหลักของ UML
- 1.5.10 พัฒนาเว็บแอปพลิเคชันให้ติดต่อกับฐานข้อมูลเชิงวัตถุ
- 1.5.11 พัฒนาแต่ละแอปพลิเคชันให้เซอร์วิสข้อมูลกันได้
- 1.5.12 ตรวจสอบผลการทำงานของระบบ
- 1.5.13 แก้ไขให้ระบบใช้งานได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ส่วนประกอบของรายงาน

รายงานฉบับนี้แบ่งเป็น 6 บท ได้แก่

บทที่ 1 กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ของโครงการ ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของโครงการ วิธีการดำเนินงาน และส่วนประกอบของรายงานฉบับนี้

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานเกี่ยวกับเว็บเซอร์วิส

บทที่ 3 กล่าวถึงทฤษฎีพื้นฐานเกี่ยวกับระบบฐานข้อมูลเชิงวัตถุ

บทที่ 4 กล่าวถึงรายละเอียดของการออกแบบระบบต่างๆที่เกี่ยวข้อง พร้อมทั้งอธิบายถึงโครงสร้างการทำงานของระบบ

บทที่ 5 กล่าวถึงการทดลองและผลการทดลองในการทำงานของระบบ

บทที่ 6 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้รับจากโครงการ ปัญหาอุปสรรคและแนวทางแก้ไข และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนา
คือ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับเว็บเซอร์วิส

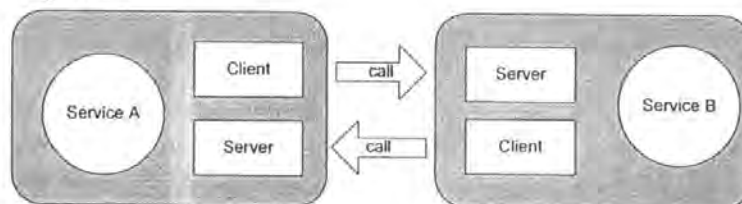
เว็บเซอร์วิส (Web Service)

เว็บเซอร์วิส คือ แอปพลิเคชันหรือซอฟต์แวร์ที่ทำงานอย่างใดอย่างหนึ่งในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากแอปพลิเคชันอื่นๆ ผ่านทางหน้าเว็บ ในรูปแบบ RPC (Remote Procedure Call) โดยที่ภาษาที่ใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์คือภาษาเอ็กซ์เอ็มแอล (XML) การอธิบายวิธีการใช้งานของเว็บเซอร์วิสนั้น จะอธิบายโดยใช้ภาษาวิสเคิล WSDL (Web Service Description Language) ซึ่งเป็นภาษา XML ประเภทหนึ่ง ระบบอื่นๆ จะสามารถติดต่อและทำงานกับเว็บเซอร์วิสได้ โดยใช้โปรโตคอลที่ชื่อว่า SOAP (Simple Object Protocol) ซึ่งใช้ภาษา XML เป็นมาตรฐานในการติดต่อระหว่างระบบผ่านทางโปรโตคอลอื่นที่ใช้ในการส่งข้อมูลบนเว็บ อาทิ เช่น โปรโตคอล HTTP



รูปที่ 2.1 ความสัมพันธ์ระหว่าง UDDI, WSDL, SOAP และ Web Service

เว็บเซอร์วิส ที่สร้างขึ้นสามารถเรียกใช้เว็บเซอร์วิสอื่นได้ กล่าวคือ แต่ละเว็บเซอร์วิสสามารถเป็นได้ทั้งไคลเอนต์และเซิร์ฟเวอร์ ลักษณะดังกล่าวคล้ายกับระบบ Peer-to-Peer มากกว่าระบบ Client-Server ดังรูปที่ 2.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ...
รูปที่ 2.2 การขอบริการระหว่างเว็บเซอร์วิส

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 Service-Oriented Architecture (SOA)

2.1.1 แนะนำ SOA

สถาปัตยกรรมเชิงบริการ(Service-Oriented Architecture: SOA) คือ สถาปัตยกรรมของซอฟต์แวร์ (Software Architecture) ซึ่งอยู่บนพื้นฐานของแนวคิดเชิงบริการ (Service-Oriented) โดยจะนิยามถึงวิธีการใช้ Service ที่มีความเป็นอิสระต่อกัน (Loosely Coupled) หรือ การเชื่อมโยง Service แบบหลวมๆ ทำให้การปรับเปลี่ยน หรือประกอบ Service ทำได้อย่างอิสระ และสามารถทำงานร่วมกันข้ามระบบได้ นอกจากนี้ยังช่วยสนับสนุนการนำ Service ที่พัฒนาไว้แล้วกลับมาใช้ใหม่ (Reuse) ทำให้สามารถรองรับความต้องการของผู้ใช้ซอฟต์แวร์ และกระบวนการทางธุรกิจ (Business Process) ซึ่งเกิดจากการทำงานร่วมกันของ Service ตามลำดับขั้นตอนที่กำหนดไว้ได้

2.1.2 จุดเด่นของ SOA

จุดเด่นของ SOA คือ ความสามารถในการทำงานร่วมกัน (Interoperability) โดยไม่สนใจระบบ โครงสร้างและเทคโนโลยีของผู้ที่ต้องการติดต่อ กล่าวคือ ความสามารถในการใช้ฟังก์ชันระหว่าง Platform ชนิดใดก็ได้โดยไม่ต้องคำนึงถึงภาษาที่ใช้พัฒนาโปรแกรม ระบบปฏิบัติการ หรือแม้แต่ประเภทของคอมพิวเตอร์ เช่น สร้างฟังก์ชัน “Check Inventory” สำหรับให้บริการแก่แอปพลิเคชันหนึ่งเพื่อควบคุมรายการสินค้า ในขณะที่เดียวกันก็สามารถนำฟังก์ชัน “Check Inventory” ไปใช้สนับสนุนเครื่องมือควบคุมรายการสินค้าแบบ Web Base ได้โดยไม่ต้องแก้ไขการนำฟังก์ชันดังกล่าวมาใช้ใหม่ถือเป็นข้อได้เปรียบ เนื่องจากช่วยลดต้นทุนการพัฒนางาน นอกจากนี้จุดมุ่งหมายระยะยาวของการนำฟังก์ชันมาใช้ซ้ำ คือ ลดฟังก์ชันที่เกินความจำเป็น ลดความซ้ำซ้อนของโครงสร้างพื้นฐานและลดค่าใช้จ่ายในการบำรุงรักษา ทำให้มีการเชื่อมโยงระบบที่รวดเร็วและยืดหยุ่นยิ่งขึ้น สามารถตรวจสอบเช็คระบบได้อย่างรวดเร็วเมื่อเปรียบเทียบกับการใช้งานระบบแบบเดิม

2.1.3 ประโยชน์ของ SOA

SOA มีประโยชน์ต่อองค์กรหลายรูปแบบขึ้นอยู่กับจุดประสงค์และวิธีการในการนำ SOA ไปใช้งาน และการสนับสนุนของผลิตภัณฑ์ รวมถึงเทคโนโลยีที่นำมาประยุกต์ใช้ด้วย โดยในที่นี้จะเสนอประโยชน์ทั่วไปที่จะได้รับจาก SOA ซึ่งเป็นเพียงส่วนหนึ่งเท่านั้น

1. ระบบทำงานร่วมกันได้ดีขึ้น

SOA ที่ใช้เทคโนโลยี Web Service นั้นไม่ยึดติดกับโครงสร้างของการสื่อสารแบบใดแบบหนึ่ง เมื่อนำคุณสมบัติดังกล่าวมาประยุกต์ใช้กับ Service Description และการรับส่งข้อมูลด้วย Message จะได้ผลลัพธ์ คือทำให้เกิดการทำงานร่วมกันอย่างแท้จริง โดยการนำแอปพลิเคชันที่ทำงานอย่างอิสระต่อกันมาทำงานร่วมกัน

2. การสืบทอดเพื่อนำกลับมาใช้ใหม่

Service-Oriented สนับสนุนการออกแบบ Service ด้วยวิธีการสืบทอด (Inherent) เพื่อนำกลับมาใช้ใหม่ ช่วยลดระยะเวลาในการพัฒนาได้เป็นอย่างมาก และช่วยลดค่าใช้จ่ายในการพัฒนาระบบด้วย Service-Oriented ได้เป็นอย่างดี

3. ปรับปรุง Architecture และ Solution ให้ดียิ่งขึ้น

ประโยชน์ของการปรับปรุงสถาปัตยกรรม คือ ลดขั้นตอนการดำเนินงาน และทำให้ความต้องการรวมถึงปัญหาต่างๆของ Solution น้อยลง

4. เพิ่มอำนาจการลงทุน

ในปัจจุบันเทคโนโลยีของเว็บเซอร์วิสได้รับการยอมรับอย่างกว้างขวาง ทำให้เกิดการปรับตัวของกลไกทางการตลาด โดยสามารถนำระบบเดิมที่มีอยู่มาทำงานร่วมกับระบบใหม่ที่พัฒนาด้วย SOA ช่วยให้ระบบงานที่ถูกพัฒนาด้วยเทคโนโลยีที่ต่างกันสามารถทำงานร่วมกันได้ โดยมีค่าใช้จ่ายต่ำ และทำให้ประสิทธิภาพการทำงานของระบบดีขึ้น แม้ว่าอาจมีปัญหาระยะเวลาในการจัดการกับระบบเดิม และปริมาณการใช้งานที่มากขึ้น รวมทั้งความพร้อมในการนำ Service-Oriented มาใช้สร้าง Solution ต่างๆก็ตาม

5. เพิ่มประสิทธิภาพในการสื่อสารข้อมูลด้วย XML

หลายหลักการของ SOA ถูกสร้างอยู่บนพื้นฐานของ XML โดยมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพในการสื่อสารข้อมูลของระบบ เนื่องจาก XML สามารถทำงานได้ทุก Platform รูปแบบมาตรฐานของข้อมูลที่ถูกกำหนดด้วย XML ช่วยลดความซับซ้อนต่างๆ ของระบบเช่น ทำให้สามารถขยาย และปรับปรุงการติดต่อสื่อสารผ่านระบบเครือข่ายได้ง่ายขึ้น

6. โครงสร้างของการติดต่อสื่อสารที่มีความยืดหยุ่นสูง

เว็บเซอร์วิส ถูกพัฒนาโดยใช้โครงสร้างการติดต่อสื่อสารที่เป็นกลางของ SOA ทำให้สามารถใช้เป็นส่วนประกอบมาตรฐานของโครงสร้างระบบ เพื่อใช้ติดต่อสื่อสารกันภายใน หรือระหว่างแอปพลิเคชันได้ ช่วยลดค่าใช้จ่าย และทำให้องค์กรต่างๆมีโครงสร้างของการติดต่อสื่อสารบนเทคโนโลยีเดียวกัน

7. เป็นวิธียาขระบบที่มีประสิทธิภาพ

ข้อจำกัดของการนำเทคโนโลยีในรูปแบบที่ต่างกันมาใช้งานร่วมกัน เพื่อตอบสนองต่อความต้องการของระบบอัตโนมัติในองค์กรธุรกิจถือเป็นปัญหาสำคัญ เนื่องจากมีค่าใช้จ่ายสูง และนำมาพัฒนาได้ยาก แม้ว่า SOA จะไม่สามารถแก้ปัญหาดังกล่าวได้ทั้งหมด แต่ก็เป็นที่ทางเลือกที่ช่วยให้การดำเนินงานของธุรกิจ และระบบสารสนเทศมีประสิทธิภาพมากขึ้น หัวใจสำคัญของ Service-Oriented คือ การสนับสนุนการขยายระบบอย่างมีประสิทธิภาพเนื่องจาก SOA ได้รับการพัฒนาอยู่บนโครงสร้างของการติดต่อสื่อสารที่เป็นกลาง โดยไม่ยึดติดกับวิธีการพัฒนา หรือ Platform ของ Middleware ใดๆ โดยใช้วิธีรวบรวมกลุ่มของระบบอัตโนมัติ และแสดงให้เห็นผ่าน Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่โดยไม่ระบุชื่อเป็นการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface เพื่อเป็นทางเลือกในการนำ Implement ให้กับ Service ทำให้เกิดความยืดหยุ่น และสามารถขยายระบบได้อย่างมีประสิทธิภาพ

8. เพิ่มความคล่องตัว (Agility) ให้กับองค์กร

ประโยชน์สำคัญที่ได้จากการนำ SOA มาใช้กับระบบองค์กร คือ การป้องกันผลกระทบจากวิวัฒนาการของเทคโนโลยีที่เปลี่ยนแปลงไปต่อระบบขององค์กร เช่น การนำกลับมาใช้ใหม่ (Reuse) และการทำงานร่วมกัน (Interoperability) เป็นต้น ซึ่งช่วยเพิ่มระดับความน่าเชื่อถือ และยกระดับขององค์กร แต่การนำรูปแบบดังกล่าวมาใช้งานจะประสบผลสำเร็จได้ก็ต่อเมื่อได้รับการออกแบบ และกำหนดมาตรฐานอย่างเหมาะสม

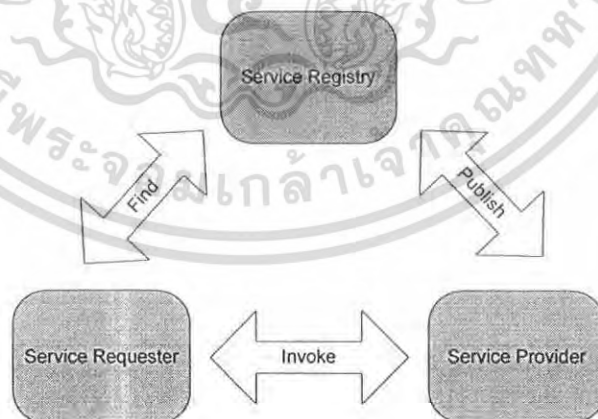
2.1.4 รูปแบบการทำงานของ SOA

SOA ถูกแบ่งออกเป็นประเภทต่างๆ โดยขึ้นอยู่กับเทคโนโลยีที่นำมา Implement เพื่อสร้าง Service โดยแบบจำลอง SOA (SOA Model) ในช่วงแรกๆเป็นจุดเริ่มต้นมาตรฐานที่สำคัญของเว็บเซอร์วิส ประกอบด้วย 3 ส่วนประกอบสำคัญ คือ ผู้ให้บริการ (Service Provider), ผู้ร้องขอบริการ (Service Requester) และหน่วยสืบค้นบริการ (Service registry) โดยมีรายละเอียดดังนี้

ผู้ให้บริการ (Service Provider) คือ แอปพลิเคชันที่จัดเตรียมบริการไว้ให้แอปพลิเคชันอื่นเรียกใช้

ผู้ร้องขอบริการ (Service Requester) คือ แอปพลิเคชันที่ขอบริการจากแอปพลิเคชันที่ได้ประกาศ (Publish) บริการไว้

หน่วยสืบค้นบริการ (Service registry) คือ ฐานข้อมูลกลางซึ่งรวบรวมบริการต่างๆ ที่ผู้ให้บริการประกาศไว้ ให้พิจารณา SOA Model ดังรูปที่ 2.3



รูปที่ 2.3 SOA Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปข้างต้นแสดง Model ของ SOA โดยผู้ให้บริการ (Service Provider) จะประกาศ (Publish) บริการ (Service) ของตนเองไปยังหน่วยสืบค้นบริการ (Service registry) เพื่อให้ผู้ร้องขอบริการ (Service Requester) ค้นหา (Find) บริการผ่าน Service Registry เมื่อพบบริการที่ต้องการแล้ว จึงเรียกใช้ (Invoke) บริการดังกล่าวผ่าน Service Provider ต่อไป เมื่อนำแบบจำลองดังกล่าวมาพิจารณา สามารถกำหนดมาตรฐานของเว็บเซอร์วิสในยุคแรกได้ดังนี้

WSDL (Web service Definition Language) คือ คำอธิบายรายละเอียดของบริการ

SOAP (Simple Object Access Protocol) คือ ผู้จัดเตรียมรูปแบบของ Message ที่ใช้ระหว่างผู้ให้บริการ และผู้ร้องขอบริการ

UDDI (Universal Description, Discovery and Integration) คือ ผู้กำหนดรูปแบบมาตรฐานสำหรับลงทะเบียน Service

2.2 Extensible Markup Language (XML)

2.2.1 จุดเริ่มต้นของภาษา XML

ภาษา Markup ถูกพัฒนาอย่างต่อเนื่องและยาวนาน โดยภาษา Markup ตัวแรกมีชื่อว่า “Generalized Markup Language (GML)” ถูกคิดค้นโดยบริษัท IBM (ประมาณปี ค.ศ. 1960) มีจุดประสงค์เพื่อใช้แสดงข้อมูลในรูปแบบของเท็กซ์ (Text) ทำให้มีความสามารถในการแลกเปลี่ยนข้อมูลบน Platform ที่ต่างกัน ต่อมาได้มีการพัฒนาภาษา Markup ที่มีความซับซ้อนมากขึ้น เรียกว่า “Standard Generalized Markup Language (SGML)” โดย SGMA ได้รวบรวมมาตรฐานต่างๆ ไว้มากมาย รวมทั้งมาตรฐานที่ไม่เกี่ยวข้องกับเว็บ ด้วยเหตุนี้แม้ว่าภาษา SGML เป็นภาษาที่มีประสิทธิภาพ แต่ยากต่อการเรียนรู้ จึงทำให้ไม่ได้รับความนิยมเท่าที่ควร

จุดสำคัญของภาษา Markup เกิดขึ้นในปลายปี 1980 ซึ่งเป็นจุดเริ่มต้นของภาษา “Hyper Text Markup Language (HTML)” โดยภาษานี้มีจุดมุ่งหมายเพื่อนำไปใช้แสดงผลข้อมูลบนอินเทอร์เน็ต ซึ่งถือเป็นเครื่องมือสำคัญที่ทำให้อินเทอร์เน็ตได้รับความนิยมอย่างสูงในปัจจุบัน ช่วงเวลานั้นเว็บไซต์ส่วนใหญ่ทั่วโลกจะใช้ภาษา HTML เนื่องจากรูปแบบของโครงสร้างภาษาที่เข้าใจง่าย ทำให้ภาษา HTML เป็นมาตรฐานของการสร้างเว็บมาอย่างยาวนานแต่การสร้างเว็บด้วย HTML ยังมีข้อจำกัด คือ ขาดความยืดหยุ่น มีลักษณะของโค้ดที่ซ้ำกัน แก้ไขยาก และไม่สามารถใช้เป็นแหล่งแลกเปลี่ยนข้อมูลที่อยู่ต่าง Platform ได้ แม้ว่าจะมีการพัฒนาเครื่องมือเพื่อช่วยในการสร้างเว็บไซต์ทำได้ง่ายขึ้น เช่น Macromedia Dreamweaver (ปัจจุบันพัฒนาถึงเวอร์ชัน 8) เป็นต้นก็ตาม แต่ก็ไม่ใช่การแก้ปัญหาที่ต้นเหตุ ทำให้มีการคิดค้นมาตรฐานใหม่ขึ้นมา นั่นคือ Extensible Markup Language (XML) ที่ดึงมาตรฐานต่างๆ ที่เกี่ยวข้องกับเว็บของภาษา SGML มาใช้งาน ทำให้เข้าใจง่ายกว่า โดย XML ถูกนำเสนอในปี ค.ศ. 1996 ภายใต้การดูแลของ W3C (World Wide Web Consortium) XML เป็น Meta Language ที่ใช้แท้้กในการสื่อความหมายของข้อมูล ซึ่งผู้พัฒนา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษา เมื่อผู้รู้เห็นหนังสือฉบับนี้แล้ว กรุณาอย่านำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถกำหนดแท็กขึ้นใช้งานได้เองตามความต้องการ สำหรับเอกสาร XML ที่ได้จะมีนามสกุล เป็น .xml

2.2.2 บทบาทของ XML ในปัจจุบัน

ภาษา XML ได้ถูกนำไปใช้พัฒนามาตรฐานใหม่ เช่น ภาษา WML ที่นำมาใช้สร้าง WAP ในอุปกรณ์มือถือ หรือภาษา MathML ที่ใช้ในวงการคณิตศาสตร์ เป็นต้น นอกจากนี้ยังมีการพัฒนา เทคโนโลยีต่างๆที่รองรับภาษา XML เพื่อเพิ่มประสิทธิภาพการทำงานของ XML ให้มากขึ้น เช่น Xlink ที่ใช้เชื่อมโยงเอกสาร, XSL สำหรับนำเอกสาร XML มาแสดงผลลัพท์บน Browser, XQL ใช้ ในการเข้าถึงฐานข้อมูลเหมือนกับภาษา SQL, DTD ใช้สำหรับนิยามเอกสาร XML และ XHTML ซึ่งนำภาษา HTML มาปรับปรุงให้เข้ากับมาตรฐานของ XML เป็นต้น

ในปัจจุบันธุรกิจด้านอุตสาหกรรมซอฟต์แวร์ได้พยายามปรับปรุงซอฟต์แวร์ของตนให้เข้า กับมาตรฐาน XML ไม่ว่าจะเป็นบริษัทยักษ์ใหญ่อย่าง Microsoft ที่นำเสนอโครงการ Microsoft Visual Studio.NET ซึ่งนำแนวคิดที่ว่าในอนาคตซอฟต์แวร์ต่างๆจะอยู่บนระบบเครือข่ายมากกว่า บนเครื่องคอมพิวเตอร์ของผู้ใช้ ดังนั้นไม่ว่าจะอยู่ในสถานที่ใดก็สามารถใช้งานแอปพลิเคชันได้ โดยผ่านระบบอินเทอร์เน็ต นอกจากนี้ Microsoft ยังได้ใช้เทคโนโลยีของ XML เป็นมาตรฐานใน การแลกเปลี่ยนข้อมูลบน .NET อาจกล่าวได้ว่าภาษา XML จะมีบทบาทต่อวงการ IT (Information Technology) อย่างมากในอนาคต โดยเฉพาะเทคโนโลยีของ Web Service ที่ถูกสร้างขึ้นมาเพื่อ รองรับการแลกเปลี่ยนข้อมูลบนระบบเครือข่าย ทั้งที่อยู่ต่าง Platform หรือใน Platform เดียวกัน

2.2.3 เป้าหมายของมาตรฐาน XML

แรกเริ่ม XML ถูกพัฒนาขึ้นมาเพื่อแก้ไขข้อบกพร่องของภาษา HTML แต่ต่อมาได้เพิ่มเติม และขยายขอบเขตความสามารถออกไป โดยสรุปได้ดังนี้

1. ต้องมีความเข้ากันได้ (Compatible) กับมาตรฐานของ SGML เนื่องจากถูกพัฒนามาจาก ภาษา SGML จึงควรที่จะเข้ากันได้ เช่น ถ้ามีองค์กรที่ใช้มาตรฐานของภาษา SGML อยู่แล้ว ภาษา SGML นี้จะต้องเข้ากันได้กับภาษา XML และในทางกลับกันถ้าองค์กรนั้นใช้มาตรฐานของภาษา XML จะต้องเข้ากันได้กับภาษา SGML เป็นต้น
2. สามารถแลกเปลี่ยนข้อมูลกันได้โดยไม่ขึ้นกับ Platform ใด Platform หนึ่ง ทำให้ XML ถูกดึงเข้ามามีบทบาทกับเทคโนโลยี Web Service ที่ต้องการให้เครื่องคอมพิวเตอร์ทั่วโลกสามารถ ติดต่อสื่อสารกันได้อย่างไร้ขีดจำกัด
3. ต้องการให้ภาษา XML เป็นภาษามาตรฐานบนเว็บที่ใช้กันทั่วโลก เพื่อให้ผู้พัฒนาเว็บมี การพัฒนาที่เป็นรูปแบบเดียวกัน ซึ่งจะช่วยให้การเรียนรู้ การควบคุม และการบำรุงรักษาทำได้ง่าย ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 ส่วนประกอบของเอกสาร XML

แท็กเป็นส่วนประกอบสำคัญของภาษา Markup การกำหนดแท็กเริ่มต้น (Start Tag) นั้น ชื่อแท็กจะอยู่ในเครื่องหมาย “<” และ “>” เช่น <INVENTORY> ส่วนการกำหนดแท็กสิ้นสุด (End Tag) จะกำหนดชื่อของแท็กที่อยู่ในเครื่องหมาย “</” และ “>” เช่น </INVENTORY> โดยจะมีเครื่องหมาย “/” แทรกอยู่หน้าชื่อ ส่วนข้อมูลต่างๆจะอยู่ระหว่าง Start Tag บางครั้งอาจอยู่ในรูปของแอททริบิวต์

2.2.4.1 กฎพื้นฐานในการเขียน XML (Well-Formed)

ไวยากรณ์ คือ กฎเกณฑ์สำคัญที่ใช้สร้างข้อมูลของ XML ซึ่งจะต้องเป็นไปตามข้อกำหนดเหล่านี้

1. ทุกอีลีเมนต์ของ XML จะต้องประกอบด้วยแท็กเริ่มต้น และ แท็กสิ้นสุด โดยทั้งสองแท็กจะต้องมีชื่อเหมือนกัน เช่น <INVENTORY>...</INVENTORY>
2. การกำหนดชื่อแท็กจะคำนึงถึง Case Sensitive คือ ตัวอักษรพิมพ์ใหญ่และพิมพ์เล็กมีความหมายแตกต่างกัน เช่น ถ้ากำหนดชื่อเป็น <INVENTORY>...</Inventory> โปรแกรมจะแสดงข้อความเตือนเนื่องจากผิดไวยากรณ์
3. ทุกเอกสาร XML จะต้องมี Root Element และมีได้เพียงหนึ่ง Root เท่านั้น โดยเป็นแท็กที่อยู่บนสุดตามหลังส่วนของการประกาศ XML
4. อีลีเมนต์ของ XML ต้องซ้อนกันอย่างเป็นลำดับ เช่น <INVENTORY> <BOOK> <TITLE>...</TITLE> </BOOK> </INVENTORY> เป็นต้น โดยไม่สามารถสลับตำแหน่งของแท็กปิดได้
5. ช่องว่าง (Space) ในเอกสาร XML ที่เกิดจากการกด Tab และ Spacebar จะเรียกว่า White Space ซึ่งมีความแตกต่างกับเอกสาร HTML คือ ในเอกสาร HTML ไม่ว่าจะเว้นช่องว่างขนาดใดก็จะมองเป็น 1 ช่องว่างเท่านั้น แต่ในเอกสาร XML สามารถรักษขนาดของช่องว่างไว้ได้
6. การตั้งชื่ออีลีเมนต์ของเอกสาร XML สามารถใช้อักขระ ตัวเลข และอักขระพิเศษได้ ยกเว้นเครื่องหมาย “&” และไม่สามารถใช้ตัวเลข หรือตัวอักษรนำหน้าชื่อของอีลีเมนต์ได้ นอกจากนี้ยังห้ามเว้นช่องว่างระหว่างชื่ออีลีเมนต์ด้วย

รูปแบบข้างต้นเป็นกฎเกณฑ์สำคัญ ที่ต้องปฏิบัติตาม มิฉะนั้นจะเกิดข้อผิดพลาดในการประมวลผลเอกสาร XML

2.2.4.2 องค์ประกอบของเอกสาร XML

เอกสาร XML มีส่วนประกอบหลัก 2 ส่วน คือ Prolog Element และ Document (หรือ Root Element) ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Prolog Element	<pre> <?xml version="1.0" encoding="ISO-8859-1"?> <!--filename = note.xml --> <note date = "12/09/2007"> <to>Tove</to> <address> <street>123 Main St.</street> <city>San Francisco</city> <state>CA</state> <zip>94117</zip> </address> <from>Jani</from> <heading>Reminder</heading> <body>Don't forget me this weekend!</body> </note> </pre>	}	Document Element
-------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---	---------------------

รูปที่ 2.4 องค์ประกอบของเอกสาร XML

Prolog Element

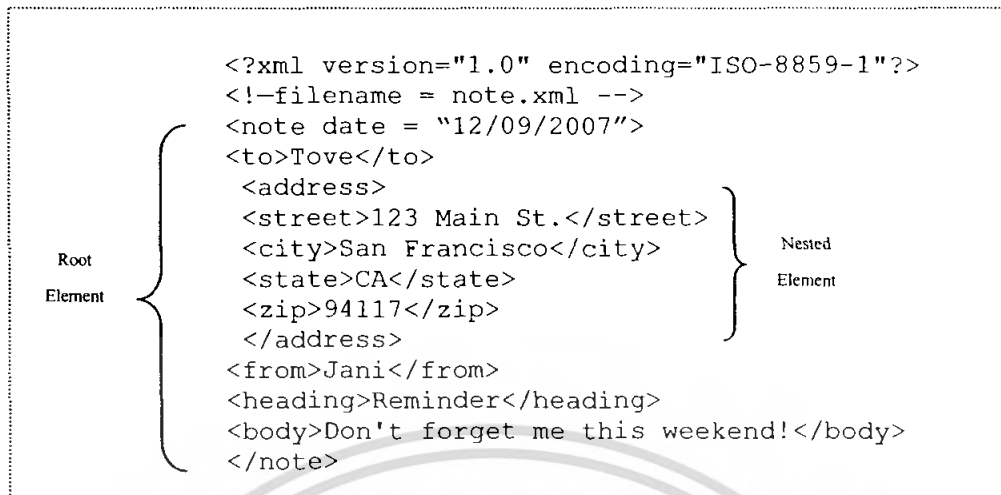
คือ ส่วนของการประกาศเอกสาร XML สามารถบรรจุส่วนต่างๆ ได้ดังนี้

- XML Declaration คือ ส่วนที่แจ้งให้ทราบว่าเอกสารนี้เป็นเอกสาร XML โดยระบุเวอร์ชัน และระบุการเข้ารหัส เช่น `<?xml version = "1.0" encoding = "ISO-8859-1" ?>`
- Comment คือ ข้อความอธิบายเอกสาร โดยตัวประมวลผลของ XML (XML Parser) จะไม่นำบรรทัดที่เป็นหมายเหตุมาประมวลผลด้วย โดยสามารถกำหนดข้อความไว้ระหว่างเครื่องหมาย `<!--` และ `-->` ได้ เช่น `<!-- filename = note.xml -->`
- DTD คือ เอกสาร XML รูปแบบหนึ่ง ที่ใช้กำหนดความถูกต้องให้กับข้อมูลในอีลีเมนต์ (จะกล่าวถึงรายละเอียดของการกำหนด DTD ภายหลัง)
- Procession Instruction คือ ส่วนประมวลผลข้อมูลที่ XML Parser จะส่งให้กับภาษาที่ใช้แสดงผลข้อมูล เช่น Style Sheet, HTML, XSL และภาษาอื่นๆ ที่สามารถทำงานร่วมกับ XML ได้ เป็นต้น

Document Element

Document Element คือ อีลีเมนต์ที่ประกอบด้วยอีลีเมนต์ย่อย (Nested Element) อื่นๆ ซ้อนกันอยู่เป็นลำดับ ทั้งนี้ จำนวนอีลีเมนต์ หรือระดับชั้นของอีลีเมนต์จะขึ้นอยู่กับลักษณะของข้อมูลและการออกแบบของผู้พัฒนา โดยที่ชื่อของอีลีเมนต์ ก็คือ คำอธิบายที่สามารถบอกได้ว่าข้อมูลในอีลีเมนต์นั้น คือข้อมูลอะไร ดังนั้นการกำหนดชื่อของอีลีเมนต์ จึงควรกำหนดให้สอดคล้องกับข้อมูล ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ส่วนประกอบของ Document Element

ในหนึ่งเอกสาร XML สามารถมี Root Element ได้เพียงหนึ่งอีลีเมนต์ แต่สามารถมี Nested Element ได้ไม่จำกัดจำนวน สำหรับรายละเอียดของอีลีเมนต์ จะกล่าวถึงในหัวข้อต่อไป

2.2.4.3 XML Element, Attribute และ Entity

หัวข้อนี้จะกล่าวถึงความหมาย รวมทั้งวิธีการกำหนดอีลีเมนต์ (Element) แอตทริบิวต์ (Attribute) และเอนทิตี (Entity) ของเอกสาร XML โดยมีรายละเอียด ดังนี้

แนะนำอีลีเมนต์ (Element)

อีลีเมนต์ คือ ส่วนที่ใช้แสดงโครงสร้าง และเนื้อหาต่างๆของเอกสาร XML โดยอีลีเมนต์บนสุด เรียกว่า “Root Element” ซึ่งสามารถมีได้เพียงอีลีเมนต์เดียวเท่านั้น ส่วนอีลีเมนต์อื่นๆจะเรียกว่า “Nested Element” ถ้าพิจารณาจากรูปที่ 2.5 จะเห็นได้ว่ามี Root Element ชื่อ “NOTE” ที่สามารถใช้สื่อถึงข้อมูลภายในได้ทันที นอกจากนี้ยังประกอบด้วย Nested Element โดยแต่ละอีลีเมนต์ก็สามารถสื่อถึงข้อมูลที่ถูกเก็บอยู่ภายในได้เช่นเดียวกัน อีกทั้งแต่ละ Nested Element ยังถูกแบ่งออกเป็น 2 กลุ่ม เพื่อใช้เก็บรายการหนังสือสองเล่ม ดังนั้น Nested Element สามารถมีชื่อที่ซ้ำกันได้ โดยการอ้างอิงข้อมูลขึ้นมาใช้งานจะต้องอ้างอิงถึงกลุ่มลำดับของข้อมูลให้ถูกต้องด้วย

ถ้าพิจารณาโครงสร้างของ XML อีกครั้งจะเห็นได้ว่า XML เป็นเอกสารที่มีโครงสร้างแบบต้นไม้ (Tree) คือ มีการกำหนดอีลีเมนต์เป็นลำดับชั้น ดังนั้นอาจกล่าวได้ในอีกลักษณะหนึ่งว่า Root Element คือ Parent ที่ประกอบด้วย Child และ Grandchild จากรูปที่ 2.5 สามารถกล่าวได้ว่า “NOTE” คือ Parent Element ที่ประกอบด้วยอีลีเมนต์ลูกชื่อ

ADDRESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีลีเมนต์สามารถแบ่งตามโครงสร้างได้ 3 ประเภท ดังนี้

1. Simple Element คือ อีลีเมนต์ที่ไม่มีอีลีเมนต์อื่นอยู่ภายใน โดยทั่วไปนิยมกำหนดให้กับข้อมูลที่ไม่มีองค์ประกอบย่อยอื่นๆ
2. Mixed Element คือ อีลีเมนต์ที่มีอีลีเมนต์อื่นอยู่ภายใน โดยทั่วไปนิยมกำหนดให้กับข้อมูลที่มีองค์ประกอบย่อย
3. Empty Element คือ อีลีเมนต์ที่ไม่ได้บรรจุข้อมูล หรือช่องว่างใดๆไว้ภายใน โดย Empty Element สามารถกำหนดได้ 2 รูปแบบคือ `emptyElement>` `</emptyElement>` หรือ `<emptyElement/>`

ความแตกต่างระหว่างอีลีเมนต์ และแท็ก

ผู้เริ่มศึกษาภาษา Markup ส่วนใหญ่จะสงสัยความแตกต่างระหว่างแท็ก (Tag) กับ อีลีเมนต์ (Element) โดยอีลีเมนต์ คือ Conceptual Object กล่าวคือ อีลีเมนต์ถูกมองเป็นอ็อบเจกต์ที่มีข้อมูล หรือมีอีลีเมนต์อื่นอยู่ภายใน ส่วนแท็ก คือ สิ่งที่นำมาใช้กำหนดตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดของอีลีเมนต์ นั่นเอง ตัวอย่างเช่น `<INVENTORY>...</INVENTORY>` หมายถึง แท็กชื่อ “INVENTORY” โดยองค์ประกอบภายในทั้งหมดจะถูกเรียกรวมกันว่าอีลีเมนต์ “INVENTORY” ดังนั้นแท็ก และอีลีเมนต์ไม่ได้เป็นสิ่งเดียวกัน แต่ในทางปฏิบัติแล้วสามารถเรียกแท็ก และอีลีเมนต์แทนกันได้ เพราะทั้งสองส่วนเป็นการกล่าวถึงข้อมูลชุดเดียวกัน แต่อย่างไรก็ตาม ควรคำนึงถึงความแตกต่างของคำศัพท์ทั้งสองนี้ด้วย

แนะนำ Entity

Entity คือ กลุ่มของอักขระที่ถูกกำหนดความหมายไว้แล้ว โดย XML Parser จะประมวลผลกลุ่มอักขระนั้นๆแล้วส่งค่าออกมาเป็นผลลัพธ์เช่น ในภาษา HTML มี Entity ที่สำคัญคือ “ ” โดยจะเห็นได้ว่าเว็บเบราว์เซอร์ไม่ได้แสดงข้อความ “ ” ออกมา แต่แสดงเป็นช่องว่างตามจำนวนกลุ่มอักขระที่กำหนดไว้สำหรับ Entity ของ XML มีอยู่ด้วยกัน 2 ประเภท ดังนี้

- Entity ที่กำหนดโดย W3C

เป็น Entity ที่ถูกกำหนดขึ้นแล้วโดยองค์กร W3C ทำให้ XML Parser สามารถประมวลผล Entity นั้นๆได้อย่างถูกต้อง โดยมี Entity ที่สำคัญ ดังนี้

- < มีค่าเท่ากับ <
- > มีค่าเท่ากับ >
- & มีค่าเท่ากับ &

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- " มีค่าเท่ากับ “
- ' มีค่าเท่ากับ ‘

จาก Entity ข้างต้น จะสังเกตได้ว่า ผลลัพธ์ของ Entity จะเป็นอักขระพิเศษ ที่มีผลต่อการทำงานของ XML Parser เช่น ถ้าใส่เครื่องหมาย < ลงในส่วนของคุณสมบัติ จะทำให้เกิดข้อผิดพลาดขึ้นได้ เนื่องจาก XML Parser จะมองเครื่องหมายดังกล่าวเป็นเครื่องหมายของแท็กที่ใช้กำหนดคอสีเมนต์ เป็นต้น

- Entity ที่ผู้พัฒนากำหนดเอง

เนื่องจาก XML เป็นภาษาที่ให้อิสระในการกำหนดข้อมูล ดังนั้นผู้พัฒนาจึงสามารถกำหนด Entity ขึ้นใช้เองได้ แต่ต้องมีการประกาศความหมายของ Entity ก่อนโดยประกาศไว้ในส่วนของ Document Type Definition

2.3 Simple Object Access protocol (SOAP)

2.3.1 แนะนำ SOAP

Web Service ใช้เอกสาร XML ในการแลกเปลี่ยนข้อมูล โดยการส่งเอกสาร XML ผ่านอินเทอร์เน็ตจะใช้ SOAP ซึ่งเป็นโปรโตคอลที่มีพื้นฐานอยู่บนมาตรฐาน XML และนิยมใช้ HTTP เป็นโปรโตคอลร่วมสำหรับส่งผ่านข้อมูลบนระบบอินเทอร์เน็ต SOAP ใช้สำหรับการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (Decentralized หรือ Distributed Environment) คุณสมบัติที่สำคัญของ SOAP คือสามารถเคลื่อนย้ายข้อมูล XML ไปยังปลายทางผ่านระบบอินเทอร์เน็ต โดยไม่ขึ้นกับ Transport Protocol ใด

เมื่อเปรียบเทียบ SOAP กับโปรโตคอลที่ใช้แลกเปลี่ยนข้อมูลแบบ Distributed System อื่นๆ เช่น Internet Inter-ORB Protocol (IIOP) ของ CORBA, Java Remote Method Protocol (JRMP) ของ RMI หรือ Object Remote Procedure Call (ORPC) ของ DCOM จะเห็นว่า SOAP ถูกออกแบบมาให้มีความยืดหยุ่น และไม่ขึ้นกับ Transport Protocol ใด กล่าวคือ SOAP สามารถใช้ Transport Protocol ทั่วไปได้ แต่เนื่องจาก HTTP เป็นโปรโตคอลที่ใช้ในการสื่อสารผ่านระบบอินเทอร์เน็ตอย่างแพร่หลาย ดังนั้นจึงนิยมใช้ SOAP ร่วมกับโปรโตคอล HTTP เพื่อให้สามารถแลกเปลี่ยนข้อมูลข่าวสารกันผ่านระบบอินเทอร์เน็ต SOAP มีหลักเกณฑ์ที่สำคัญ 4 ข้อ คือ Message Envelope, กฎของการ Encoding, ระเบียบแบบแผนของ RPC และการใช้ Transport Protocol ร่วมกับ SOAP

2.3.1.1 Message Envelope

Message Envelope ใช้กำหนดระเบียบแบบแผนสำหรับเนื้อหา และรวบรวม Message ต่างๆ เป็นแพ็คเกจ นอกจากนี้ยังใช้กำหนดเส้นทาง และการประมวลผล Message อีกด้วย

2.3.1.2 กฎของการ Encoding

กฎของการ Encoding ใช้สำหรับกำหนดระเบียบแบบแผนสำหรับการ Mapping ข้อมูลของแอปพลิเคชัน ไปเป็นข้อมูลในรูปแบบของ XML โดยอยู่บนพื้นฐานของ XML Schema กฎของการ Encoding เป็นข้อบังคับที่ใช้เป็นแนวทางในการสร้าง และกำหนด รูปแบบของ Message

2.3.1.3 ระเบียบแบบแผนของ Remote Procedure Call (RPC)

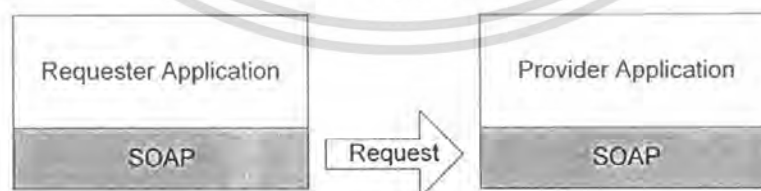
RPC มีการทำงานในรูปแบบของการส่งคำร้องขอ (Request) และการตอบรับ (Response) โดยข้อมูลที่เป็นคำร้องขอจะถูกจัดลำดับ และส่งให้กับแอปพลิเคชันปลายทาง เมื่อแอปพลิเคชันปลายทางได้รับข้อมูลก็จะส่งการตอบรับกลับมา โดย SOAP ใช้กฎเกณฑ์ ของ RPC

2.3.1.4 การใช้ Transport Protocol ร่วมกับ SOAP

SOAP เป็นโปรโตคอลที่ไม่ขึ้นกับ Transport ใด กล่าวคือ SOAP สามารถทำงาน ร่วมกับ Transport Protocol ใดก็ได้ อย่างไรก็ตาม นิยมใช้โปรโตคอล HTTP ทำงานร่วมกับ SOAP ตามที่กล่าวไว้ในตอนต้น

2.3.2 การส่งข้อมูลผ่าน SOAP

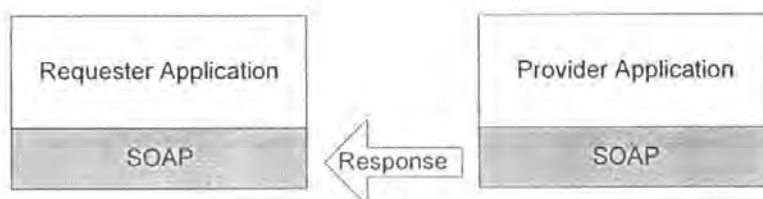
กระบวนการส่งข้อมูลระหว่างผู้ให้บริการ และผู้ร้องขอบริการจะเริ่มต้นจากแอปพลิเคชัน ของผู้ร้องขอบริการสร้าง SOAP Message เพื่อเรียกใช้ Web Service ที่ผู้ให้บริการประกาศไว้ ดังรูป ที่ 2.6



รูปที่ 2.6 การส่งคำร้องขอ (Request) ไปยังผู้ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ให้บริการได้รับ SOAP Message ที่ผู้ร้องขอบริการส่งมาให้แล้ว Web Service จะประมวลผลตามคำร้องขอนั้น และส่งผลลัพธ์ให้กับผู้ร้องขอบริการโดยสร้าง SOAP Message พร้อมกับแนบผลลัพธ์กลับไปยังผู้ร้องขอบริการ ดังรูปที่ 2.7



รูปที่ 2.7 การส่งค่าตอบรับ (Response) ไปยังผู้ร้องขอบริการ

2.3.3 SOAP Model

จุดประสงค์หลักของ SOAP คือ กำหนดรูปแบบมาตรฐานให้กับ Message โดย SOAP มีโครงสร้าง และรูปแบบที่เรียบง่าย ยืดหยุ่น และมีประสิทธิภาพ SOAP ถูกออกแบบมาให้มีโครงสร้างในรูปแบบของ XML โดยมีส่วนประกอบหลัก 3 ส่วน คือ SOAP Envelope, SOAP Header และ SOAP Body

2.3.3.1 แนะนำ SOAP Model

โครงสร้างพื้นฐานของ SOAP มีส่วนประกอบหลัก 3 ส่วน คือ SOAP Envelope, SOAP Header และ SOAP Body โดยแต่ละส่วนประกอบมีหน้าที่ดังนี้

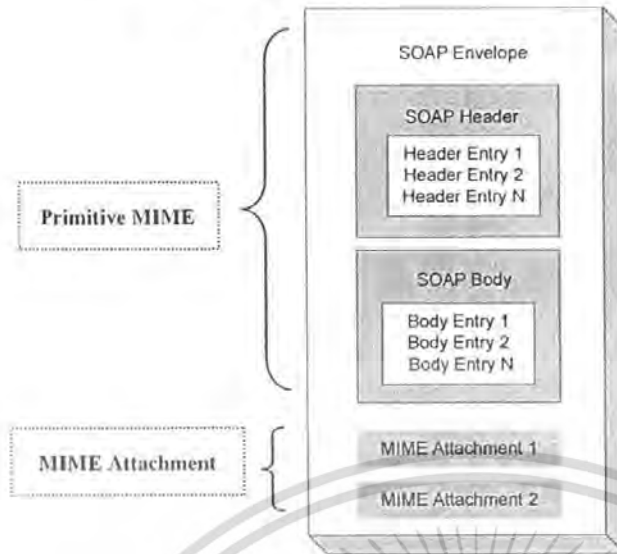
1. SOAP Envelope คือ เนื้อหา (Content) ทั้งหมดของเอกสาร
2. SOAP Header คือ ส่วนอธิบายเอกสาร SOAP โดยจะกำหนดหรือไม่กี่ได้
3. SOAP Body คือ ส่วนที่อธิบายวิธีเรียกใช้งานบริการ และผลลัพธ์ของบริการ

SOAP Model เปรียบเทียบได้กับการส่งจดหมาย ซึ่งซองจดหมายเปรียบเสมือน SOAP Envelope ที่ภายในบรรจุจดหมายไว้ โดยที่ส่วนหัวของจดหมายระบุที่อยู่ของผู้รับไว้ เปรียบเสมือนกับ SOAP Header ส่วนเนื้อหาของจดหมายเปรียบเสมือน SOAP Body นั่นเอง

2.3.3.2 โครงสร้างของ SOAP Message

โครงสร้างของ SOAP Message (หรือเอกสาร SOAP) ประกอบด้วย 3 อีลีเมนต์สำคัญ คือ อีลีเมนต์ Envelope, Header และ Body ให้พิจารณาโครงสร้างของ SOAP Message ดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 โครงสร้างของ SOAP Message

จากรูปที่ 2.8 สามารถแบ่งส่วนประกอบของ SOAP Message ได้เป็น 2 ส่วน โดยส่วนแรกเรียกว่า “Primary MIME” ซึ่งบรรจุ Text และ XML ไว้ ส่วนที่ 2 คือ “MIME Attachment”

ส่วนที่เป็น Text และ XML คือ ส่วนประกอบหลักของ SOAP Message โดยกำหนดอีลีเมนต์ Envelope ซึ่งเป็น Element หลักให้บรรจุ Element ย่อยไว้ภายใน 2 Element คือ อีลีเมนต์ Header และ Body โดยอีลีเมนต์ Envelope และ Body เป็น Element สำคัญที่ต้องมีในทุก SOAP Message ในขณะที่อีลีเมนต์ Header จะกำหนดหรือไม่ก็ได้ นอกจากนี้อีลีเมนต์ Envelope และ Body สามารถบรรจุคำสั่งได้มากกว่า 1 บล็อก โดยบล็อกคำสั่งในอีลีเมนต์ Header เรียกว่า “Header Entry” ส่วนบล็อกคำสั่งในอีลีเมนต์ Body เรียกว่า “Body Entry”

SOAP Body ถูกออกแบบมาเพื่อบรรจุเฉพาะข้อมูลที่เป็นตัวอักษร ต่อมา SOAP จึงได้ถูกพัฒนาให้สามารถแนบข้อมูลในรูปแบบของ Binary ไว้กับ SOAP Envelope ได้ โดยข้อมูลแบบ Binary ที่แนบไปจะถูกส่ง และเข้ารหัสในรูปแบบของ MIME

2.3.4 SOAP Encoding

SOAP Encoding คือ การเข้ารหัสของ SOAP เป็นการแทนข้อมูลในรูปแบบที่ต้องการ ข้อมูลอาจถูกเก็บอยู่ในส่วนของ Header Entry หรือ Body Entry แต่ SOAP อาจไม่มีการกำหนด Encoding ไว้ก็ได้ หากต้องการใช้ SOAP Encoding จะต้องกำหนด Encoding Schema ซึ่งไม่มีการกำหนดค่าเริ่มต้นไว้ โดยสามารถระบุ Encoding Schema ได้ โดยใช้แอททริบิวต์ encoding Style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Encoding Schema ของ SOAP ที่จัดเตรียมไว้ถูก Encoding ด้วย XML Schema เพื่อใช้อธิบายชนิดข้อมูลที่สามารถใช้ภายใน SOAP ได้ โดยแบ่งเป็น 2 ประเภท คือ Simple Type และ Compound Type

2.3.5 รูปแบบการติดต่อสื่อสารของ SOAP

แอปพลิเคชันที่ใช้ SOAP จะสามารถติดต่อสื่อสารกันได้อย่างอิสระไม่ขึ้นกับ Platform หรือภาษาโปรแกรมใดๆ โดยการติดต่อสื่อสารระหว่าง SOAP Node นั้นมี 2 รูปแบบ คือ SOAP RPC และ SOAP Messaging

2.3.5.1 SOAP RPC

SOAP RPC เป็นการติดต่อสื่อสารในรูปแบบของการร้องขอ (Request) และการตอบรับ (Response) โดยใช้กลไกการติดต่อสื่อสารแบบ Remote Procedural Call (RPC) โดย SOAP Message อาจกำหนดส่วนที่ใช้เชื่อมต่อ (Interface) ด้วยการบรรจุชื่อ พารามิเตอร์ หรือค่าที่ถูกส่งคืน (Return Value) ขึ้นอยู่กับรูปแบบของ SOAP Message ซึ่งมี 2 ชนิด คือ Request Message และ Response Message นั่นเอง

- Request Message ของ SOAP จะทำหน้าที่เป็น Method Call เพื่อเรียกใช้ Remote Object ที่เครื่อง Server จัดเตรียมไว้ให้บริการ โดย Method Call จะมีพารามิเตอร์หรือไม่ก็ได้
- Response Message ของ SOAP อาจจะมีการส่งคืนค่า (Return Value) เพื่อใช้เป็นผลลัพธ์

2.3.5.2 SOAP Messaging

SOAP Messaging (หรือการส่ง SOAP ในรูปแบบเอกสาร XML) เป็นการติดต่อสื่อสารที่มีความเป็นอิสระ (Loosely Coupled) ด้วยวิธีการใช้ Message ในรูปแบบของเอกสาร XML เพื่อแลกเปลี่ยนข้อมูล โดยภายใน Body ของ SOAP Message จะบรรจุเอกสาร XML หรือข้อมูลต่างๆที่ถูก Encode ไว้ตามข้อกำหนดของ XML Schema และใช้ SOAP Node ในการส่ง (Sending) และรับ (Receiving) ข้อมูล โดย SOAP Node ที่เป็นผู้ส่งจะส่ง Message ไปพร้อมกับบรรจุเอกสาร XML ไว้ภายใน Body ของ Message ส่วน SOAP Node ที่เป็นผู้รับ (Receiver) จะรับ Message เพื่อนำไปประมวลผล ให้พิจารณา SOAP Message ที่ใช้การติดต่อสื่อสารแบบ Messaging ดังรูปที่ 2.9

```

<env:Envelope xmlns:env="http://www.w3.org/2001/09/soap-envelope">
  <env:Header>
    <n:ProductListNotice xmlns:n="http://ktp.com/ProductList">
      <n:productID>BKTP9999</n:productID>
    </n:ProductListNotice>
  </env:Header>
  <env:Body>
    <m:product xmlns:m="http://ktp.com/product">
      <m:name>SOA and Web Service</m:name>
      <m:quantity>1000</m:quantity>
    </m:product>
  </env:Body>
</env:Envelope>

```

รูปที่ 2.9 ตัวอย่าง SOAP Message

โค้ดข้างต้นเป็น SOAP Message ที่อยู่ในรูปแบบของ Messaging โดยภายใน Message จะบรรจุ Header Block ชื่อ ProductListNotice และ Body ชื่อ product ไว้ โดย Header ชื่อ ProductListNotice ใช้สำหรับเก็บข้อมูลที่ผู้รับ (Receiver) ต้องการ ส่วน Body ชื่อ product ใช้สำหรับเก็บ Message ที่ใช้ส่งออกไป

2.3.6 SOAP กับ HTTP

การใช้ HTTP เป็น Transport Protocol คือ วิธีที่เหมาะสมที่สุดสำหรับการติดต่อสื่อสารของ SOAP ในรูปแบบ RPC โดยจะอยู่บนพื้นฐานของการร้องขอ (Request) และการตอบรับ (Response) ผ่าน HTTP ทำให้สามารถรับส่งข้อมูลผ่านระบบอินเทอร์เน็ต หรืออินทราเน็ตได้ด้วยการส่งพารามิเตอร์ต่างๆ ไปกับ HTTP Request และรับพารามิเตอร์ต่างๆ จาก HTTP Response การใช้ SOAP บนโปรโตคอล HTTP จะมีการส่ง SOAP Message ผ่านพอร์ตมาตรฐานของ HTTP (พอร์ต 80) โดยไม่ต้องร้องขอให้ Firewall เปิดพอร์ต เนื่องจาก พอร์ตดังกล่าวจะถูกเปิดไว้อยู่แล้ว

- SOAP Request บน HTTP

การติดต่อสื่อสารด้วย SOAP บน HTTP ฟังผู้ร้องขอบริการ (Service Requester) จะต้องติดต่อไปยังผู้ให้บริการ (Service Provider) เมื่อสร้างการเชื่อมต่อแล้วผู้ร้องขอบริการจะส่ง SOAP Request บน HTTP ไปยังผู้ให้บริการ

- SOAP Response บน HTTP

เมื่อผู้ให้บริการได้รับ SOAP Request จะทำการประมวลผล และส่ง SOAP Response บน HTTP กลับไปให้ผู้ร้องขอบริการ โดยภายในจะบรรจุ Status Code ที่บอกถึงสถานะของการร้องขอ และผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Web Service Description Language (WSDL)

2.4.1 แนะนำ WSDL

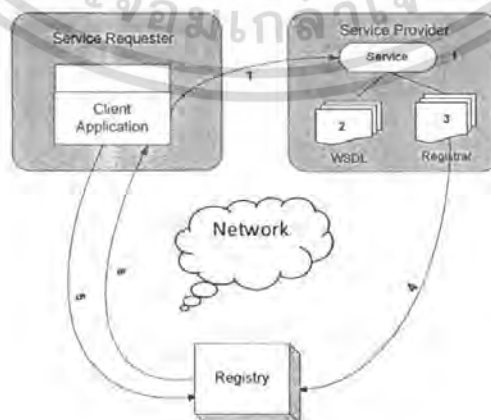
WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของ Web Service และวิธีการติดต่อเพื่อขอรับบริการจาก Web Service จึงถือได้ว่า WSDL ทำหน้าที่เป็น Service Description ภาษา WSDL ถูกคิดค้นโดยบริษัท IBM และ Microsoft อยู่ภายใต้การดูแลของ W3C ตัวอย่างของรายละเอียดที่เอกสาร WSDL จัดเตรียมไว้ ได้แก่ ชื่อของ Web Service ชื่อเมธอด (Method) ที่เปิดให้บริการ พารามิเตอร์ที่ใช้รับส่งไปยังเมธอด ชนิดข้อมูลของพารามิเตอร์และตำแหน่งของ Web Service เป็นต้น โดยเอกสาร WSDL ประกอบด้วย 6 อีลีเมนต์หลัก ดังนี้

types	ใช้กำหนดรูปแบบของข้อมูลต่างๆ
message	ใช้อธิบายข้อมูลที่ใช้แลกเปลี่ยนระหว่างผู้ให้บริการ และผู้ร้องขอบริการ
portType	ใช้อธิบายการทำงาน (Operation) ที่ Service เตรียมไว้
binding	ใช้อธิบายถึงโปรโตคอลที่ใช้งาน
port	ใช้ระบุตำแหน่งของการ binding และรวบรวม port ที่เกี่ยวข้องไว้ด้วยกัน
service	ใช้ระบุถึงบริการต่างๆ

เอกสาร WSDL จะนิยามข้อมูลต่างๆของ Web Service ผ่านชุดของอีลีเมนต์ข้างต้น โดย WSDL จะมอง Web Service เหมือนกับพอร์ต (Port) ที่ใช้สำหรับติดต่อสื่อสารของโปรโตคอลต่างๆเช่น พอร์ต TCP/IP

2.4.2 กระบวนการร้องขอบริการ

แอปพลิเคชันที่มีการร้องขอบริการระหว่างกัน จะต้องมีข้อตกลงต่างๆร่วมกัน โดยทั่วไปจะมีองค์ประกอบสำคัญ 3 ส่วน คือ Server Application (หรือ Service Provider), Client Application (หรือ Service Requester) และ Registry (หรือ Service Broker) มีขั้นตอนการทำงานเบื้องต้น ดังรูปที่ 2.10



รูปที่ 2.10 กระบวนการขอบริการ

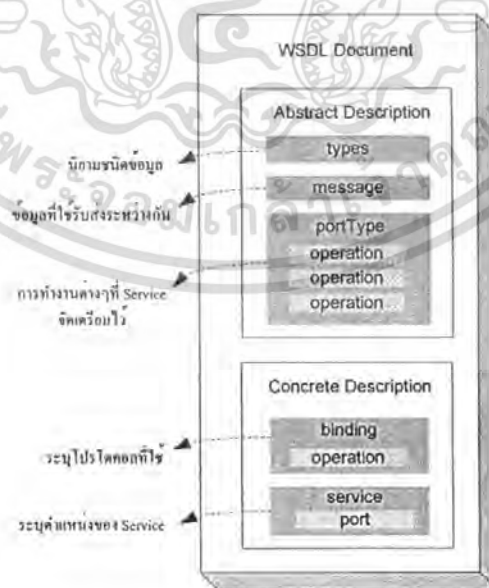
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาก็เท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปสามารถอธิบายขั้นตอนต่างๆ ได้ ดังนี้

- ขั้นตอนที่ 1 Service Provider สร้าง Service เพื่อให้บริการในรูปแบบของ Web Service
- ขั้นตอนที่ 2 Service Provider สร้างเอกสาร WSDL เพื่อบอกตำแหน่ง และรายละเอียดต่างๆ ของบริการที่ Service Provider จัดเตรียมไว้ให้กับ Service Requester โดยไฟล์ WSDL อาจถูกสร้างด้วยเครื่องมือต่างๆ
- ขั้นตอนที่ 3 สร้างแอปพลิเคชัน หรือใช้ซอฟต์แวร์ที่ทำหน้าที่เป็นนายทะเบียน (Registrar) เพื่อประกาศ (Publish) ข้อมูลเกี่ยวกับบริการไว้บน Registry โดยข้อมูลดังกล่าว คือ ไฟล์ WSDL นั่นเอง
- ขั้นตอนที่ 4 เมื่อสร้าง Registrar แล้วให้ประกาศเอกสาร WSDL ไว้บน Registry
- ขั้นตอนที่ 5 Service Requester จะค้นหาข้อมูลของบริการที่ต้องการใช้งานบน Registry
- ขั้นตอนที่ 6 Registry จะส่งเอกสาร WSDL ให้กับ Service Requester เพื่อบอกรายละเอียดต่างๆ ของ Service ให้ Service Requester ทราบ
- ขั้นตอนที่ 7 Service Requester จะร้องขอบริการด้วยวิธี SOAP Request และ SOAP Response ผ่าน โพรโตคอล HTTP

2.4.3 โครงสร้างของเอกสาร WSDL

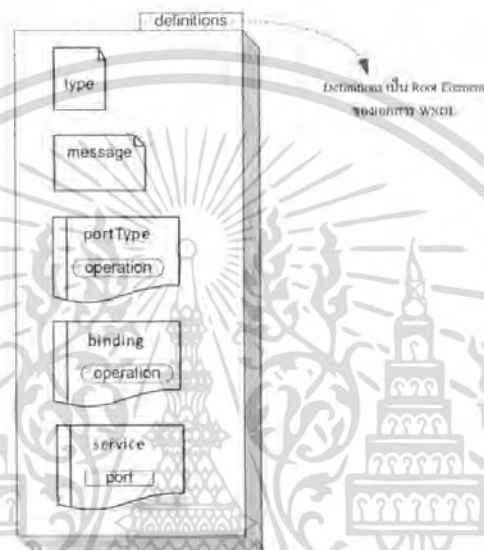
เอกสาร WSDL สามารถแยกได้เป็น 2 ส่วน คือ Abstract Description และ Concrete Description ให้พิจารณารูปที่ 2.11



รูปที่ 2.11 Abstract Description และ Concrete Description ของเอกสาร WSDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลใน Abstract Description ประกอบด้วยอิลีเมนต์ที่สำคัญ คือ types, message และ portType โดยจะไม่ขึ้นกับ Platform และเทคโนโลยีใดๆแต่ส่วนของโปรโตคอล เทคโนโลยีรูปแบบในการรับส่งข้อมูล และรายละเอียดในการ Implement จะถูกกำหนดไว้ใน Concrete Description แทน โดย Concrete Description ประกอบด้วย 3 อิลีเมนต์สำคัญ คือ binding, service และ port อย่างไรก็ตาม ทุกอิลีเมนต์ของเอกสาร WSDL จะต้องอยู่ภายใต้อิลีเมนต์ definitions ให้พิจารณาโครงสร้างของเอกสาร WSDL ดังรูปที่ 2.12



รูปที่ 2.12 โครงสร้างของเอกสาร WSDL

2.4.4 WSDL Binding

WSDL Binding จะเกี่ยวข้องกับการเข้ารหัสข้อมูล และการผูก (Binding) เอกสาร WSDL ไว้กับโปรโตคอลที่ใช้ในการติดต่อสื่อสาร โดยเอกสาร WSDL สามารถเพิ่มอิลีเมนต์ที่เป็นส่วนขยายที่เรียกว่า “Extensibility Element” ลงในอิลีเมนต์ต่างๆในรูปแบบของ Namespace ได้ (เช่น soap:binding) และอิลีเมนต์นี้ใช้เพื่อระบุเทคโนโลยีที่ใช้ในการ Binding หรือเรียกว่า “WSDL Binding Extensibility” โดย Extensibility Element ได้จัดเตรียมกลไกเพื่อให้ WSDL รองรับโปรโตคอลต่างๆที่ได้รับการปรับปรุงโดยไม่ต้องแก้ไขข้อกำหนดของ WSDL ไว้ โดยที่ WSDL Binding Extensibility สามารถแบ่งได้เป็น 3 แบบ คือ SOAP Binding, HTTP GET & POST Binding และ MIME Binding โดยในหัวข้อนี้จะกล่าวถึงเฉพาะ SOAP Binding เท่านั้น ในพิจารณาอิลีเมนต์ต่างๆที่สำคัญของ SOAP Binding ดังต่อไปนี้

2.4.4.1 อิลีเมนต์ soap:binding

อิลีเมนต์ soap:binding ใช้อธิบายถึงรูปแบบของ SOAP Protocol ที่ใช้ ให้พิจารณา

โค้ดดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<binding name = "myDataBinding" type="tns:myData">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
</binding>

```

รูปที่ 2.12 รูปแบบของ SOAP Protocol

แอททริบิวต์ style ใช้ระบุถึงรูปแบบการรับส่งข้อมูล ในที่นี้ใช้แบบ RPC (หรือ Request-Response นั่นเอง) ดังนั้นภายในอีลีเมนต์ operation จะต้องกำหนดทั้งอีลีเมนต์ input และ output ไว้ภายใน ส่วนแอททริบิวต์ transport ใช้กำหนด Transport Protocol ที่จะนำมาใช้กับ SOAP เช่น http://schemas.xmlsoap.org/soap/http หมายถึง โพรโทคอล HTTP หรือ http://schemas.xmlsoap.org/soap/smtp หมายถึง โพรโทคอล SMTP เป็นต้น โดยในที่นี้ใช้โพรโทคอล HTTP

2.4.4.2 อีลีเมนต์ soap:operation

อีลีเมนต์ soap:operation ใช้กำหนดข้อมูลที่เกี่ยวข้องกับรูปแบบการติดต่อสื่อสาร โดยมีแอททริบิวต์ style สำหรับระบุรูปแบบการรับส่งข้อมูล (เหมือนกับอีลีเมนต์ soap:binding) และแอททริบิวต์ soapAction ใช้กำหนดค่าให้กับ SOAPAction Header ในรูปแบบของ URI โดยจะต้องทำงานกับ SOAP บนโพรโทคอล HTTP ในกรณีที่ไม่ได้ระบุ URI ไว้ในแอททริบิวต์ soapAction หมายถึง การส่งคำร้องขอของ SOAP ไปบนโพรโทคอล HTTP

2.4.4.3 อีลีเมนต์ soap:body

อีลีเมนต์ soap:body ใช้ระบุถึงส่วนประกอบของ Message ที่อยู่ในอีลีเมนต์ SOAP Body โดย soap:body จะจับคู่ Input Message และ Output Message กับ SOAP Protocol

2.4.4.4 อีลีเมนต์ soap:address

อีลีเมนต์ soap:address ใช้สำหรับระบุ port ซึ่งเป็นที่อยู่ (Address) ของ Service ไว้ในแอททริบิวต์ Location โดยทั่วไปจะอยู่ในรูปแบบของ URI

2.5 Universal Description Discovery and Integration (UDDI)

2.5.1 แนะนำ UDDI

Universal Description Discovery and Integration (UDDI) เป็นข้อกำหนดที่จัดตั้งขึ้นเพื่อรองรับการทำงานระหว่างบริษัทหรือองค์กรต่างๆ เพื่อค้นหาบริการของ Web Service โดยถูกออกแบบให้สามารถจัดการกับข้อมูลของ Web Service ต่างๆระหว่างคู่ค้าธุรกิจได้ เปรียบเสมือนฐานข้อมูลขนาดใหญ่ที่รวบรวมข้อมูลของ Web Service ที่ให้บริการในการจัดเก็บ ปรับปรุง การค้นหาและบริการอื่นๆ นอกจากนี้ยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มเติมข้อมูลต่างๆที่เกี่ยวข้องกับบริการหรือผลิตภัณฑ์ช่วยทำให้ทราบถึงรายละเอียดต่างๆที่เกี่ยวข้องกับผลิตภัณฑ์และบริการของบริษัทหรือคู่ค้าทางธุรกิจได้

2.5.1.1 ข้อมูลทางธุรกิจและการจัดกลุ่มข้อมูล

ในปัจจุบันการเติบโตและความอยู่รอดของธุรกิจต่างๆมักขึ้นอยู่กับชนิดและขอบเขตของข้อมูลว่าจะสามารถเข้าถึงลูกค้าได้มากน้อยเพียงใด ลูกค้าสามารถค้นหาและค้นพบข้อมูลเกี่ยวกับบริการหรือผลิตภัณฑ์ของตนได้อย่างไร เพื่อเป็นเครื่องชี้นำให้ลูกค้าสามารถเข้าถึงตัวบริการหรือผลิตภัณฑ์ของผู้ให้บริการได้ ลูกค้าก็จะสามารถค้นหาบริการหรือผลิตภัณฑ์ผ่านสื่อต่างๆได้สะดวกขึ้น เช่น สมุดโทรศัพท์หน้าขาว และสมุดโทรศัพท์หน้าเหลือง เป็นต้น โดยที่สมุดโทรศัพท์หน้าขาวจะแสดงเฉพาะหมายเลขโทรศัพท์และที่ตั้งของบริษัทหรือองค์กรที่ให้บริการเท่านั้น ซึ่งจัดหมวดหมู่โดยการเรียงลำดับตามตัวอักษรของชื่อบริษัท สำหรับสมุดโทรศัพท์หน้าเหลืองนั้นจะแสดงถึงบริการหรือผลิตภัณฑ์ของบริษัทต่างๆ โดยจัดหมวดหมู่ตามประเภทของธุรกิจและบริการ ความแตกต่างกันระหว่างสมุดโทรศัพท์หน้าขาวและสมุดโทรศัพท์หน้าเหลือง ก็คือรายละเอียดของข้อมูลทางธุรกิจและการจัดกลุ่ม กล่าวคือ สมุดโทรศัพท์หน้าเหลืองจะจัดกลุ่มตามประเภทของธุรกิจและบริการ ช่วยให้ลูกค้าสามารถค้นหาและเลือกบริการที่ตนต้องการจากบริการในกลุ่มเดียวกันได้และยังทราบรายละเอียดของบริการบางส่วนได้อีกด้วย แต่สมุดโทรศัพท์หน้าขาวจะให้เพียงข้อมูลเบอร์โทรศัพท์และที่ตั้งเท่านั้น ทำให้ลูกค้าไม่ทราบรายละเอียดของบริการที่ตนต้องการได้ ส่งผลให้การเข้าถึงบริการเป็นไปได้ยาก จากที่กล่าวมาข้อมูลทางธุรกิจและการจัดกลุ่มข้อมูลเป็นสิ่งที่มีความสำคัญต่อการติดต่อทางธุรกิจ ช่วยให้ลูกค้าสามารถเข้าถึงบริการหรือผลิตภัณฑ์ได้เป็นอย่างดี ซึ่ง UDDI ก็มีหน้าที่เสมือนสมุดโทรศัพท์หน้าขาวและหน้าเหลืองดังกล่าวนั่นเอง

2.5.1.2 ข้อกำหนดของ UDDI

การให้บริการต่างๆของ UDDI นั้นถูกออกแบบอยู่บนข้อกำหนดของ UDDI เพื่อจัดเก็บ แบ่งปัน และนำข้อมูลต่างๆที่เกี่ยวข้องกับบริการไปใช้งาน ซึ่งข้อกำหนดของ UDDI จะช่วยในการสร้างแอปพลิเคชันที่ใช้จัดเก็บ ปรับปรุง แบ่งปัน ค้นหา และใช้ข้อมูลทางธุรกิจร่วมกัน

UDDI Browser Registry (UBR) มีหน้าที่ในการจัดเก็บข้อมูลทางธุรกิจใหม่ๆรวมทั้งปรับปรุงข้อมูลที่มีอยู่และการใช้คีย์เวิร์ด (Keyword) ในการค้นหาข้อมูลทางธุรกิจ โดยจะนำเสนอบริการที่ใช้ในการประกาศข้อมูลต่างๆขององค์กร สำหรับข้อกำหนดของ UDDI นั้นมีการกำหนดมาตรฐานโดยผู้ผลิตซอฟต์แวร์ต่างๆไว้มากมาย เช่น IBM, Microsoft, Arriba, Fujitsu เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.3 การลงทะเบียน

การลงทะเบียน Service นั้นมีอยู่ 2 ประเภท คือ

1. Public Registry เป็นการจัดตั้งข้อกำหนดของ UDDI โดยผู้ผลิตซอฟต์แวร์ชั้นนำต่างๆเพื่อเป็นศูนย์กลางในการค้นหา Service เช่น IBM, Microsoft เป็นต้น
2. Private Registry จะเป็นการจัดตั้งข้อกำหนดของ UDDI เพื่อใช้เฉพาะภายในกลุ่มลูกค้าทางธุรกิจเดียวกันเท่านั้น

2.5.2 คำศัพท์ที่เกี่ยวข้องกับ UDDI

ก่อนที่จะกล่าวถึงรายละเอียดของ UDDI จำเป็นต้องทำความเข้าใจกับคำศัพท์เหล่านี้ก่อน

- Node API Set

เป็นกลุ่มของ Node API ซึ่งมี API ที่ใช้ในการติดต่อกันภายใน UDDI โดยมีวัตถุประสงค์เพื่อจัดการข้อมูลที่อยู่ใน UDDI ที่มีหน้าที่และการใช้งานที่แตกต่างกัน

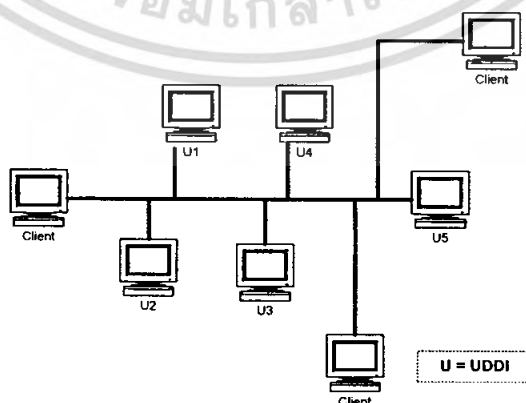
- UDDI Node

ในระบบจะต้องประกอบด้วยกลุ่มของ Node API อย่างน้อยหนึ่งตัวเรียกว่า UDDI Node ซึ่งจะต้องมีคุณสมบัติดังนี้

1. สามารถตอบสนองกับข้อมูล UDDI ผ่านกลุ่ม API ได้
2. จะต้องผ่านการ Registry เพียงครั้งเดียวเท่านั้น
3. มีความสามารถในการเข้าถึงและจัดการข้อมูล UDDI ได้

- UDDI Registry

เป็นการรวมกันของ UDDI Node ตั้งแต่หนึ่งตัวขึ้นไป ซึ่ง UDDI Node เหล่านี้จะออกแบบให้ทำหน้าที่ในการเก็บ และจัดการกับข้อมูลทางธุรกิจที่ผ่านการลงทะเบียนแล้ว ดังรูปที่ 2.13



รูปที่ 2.13 สถาปัตยกรรมของ UDDI Registry

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้เห็นหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data Structure

คือ โครงสร้างข้อมูล เป็นส่วนประกอบสำคัญในข้อกำหนดของ UDDI โครงสร้างข้อมูลของ UDDI จะอยู่ในรูปแบบของภาษา XML โดยบรรจุข้อมูลทางธุรกิจไว้ใน UDDI Node สำหรับโครงสร้างข้อมูลที่สำคัญของ UDDI นั้นมี 6 อีลีเมนต์ คือ

1. businessEntity
2. businessService
3. bindingTemplate
4. tModel
5. publisherAssertion
6. subscription

2.5.3 การประกาศ Service (Service Publication)

การประกาศ Service จะเกี่ยวข้องกับการสร้าง การเพิ่มเติมข้อมูลให้ทันสมัย และการย้ายข้อมูลที่สัมพันธ์กับธุรกิจหรือ Service ขององค์กรต่างๆบน UDDI Registry หน้าที่การประกาศ Service นั้นจะเป็นของผู้ดูแลระบบโดยลูกค้าและผู้ใช้ทั่วไปจะไม่มีส่วนเกี่ยวข้อง การประกาศ Service จะทำงานผ่านแอปพลิเคชันที่มีไว้รองรับ โดยเฉพาะ แอปพลิเคชันนี้จะทำการติดต่อกับ Registry ด้วย SOAP โดยที่แอปพลิเคชันนี้จะเลือกใช้ API ที่เหมาะสมมาช่วยในการประกาศ Service

2.5.3.1 การสร้างและปรับปรุง Service

การประกาศข้อมูลต่างๆ มีความสำคัญต่อการนำมาช่วยตัดสินใจในการวางกลยุทธ์ตามข้อมูลจากการลงทะเบียนในส่วนที่บริษัทสนใจ แต่ข้อมูลทั้งหมดก็ไม่ได้มีความสำคัญและจำเป็นต่อ Web Service เสมอ ปริมาณข้อมูลが多เกินไปก็อาจส่งผลให้เกิดความสับสนและยากที่จะเข้าถึงข้อมูลที่น่าเชื่อถือได้ ส่วนปริมาณข้อมูลที่มีอยู่อย่างจำกัดหรือน้อยเกินไป ก็อาจไม่ได้รับความสนใจจากผู้ใช้ที่ต้องการข้อมูลของบริการอย่างละเอียด ดังนั้นจึงควรมีปริมาณของข้อมูลที่เหมาะสม การสร้าง การเพิ่มเติมข้อมูลให้ทันสมัย และการลบข้อมูลของ UDDI นั้นสามารถทำได้ด้วย SOAP Message ซึ่งมี 4 ประเภท คือ Save Business, Save Service, Save Binding และ Save tModel อธิบายได้ดังนี้

- Save Business

นำเสนอด้วยอีลีเมนต์ `save_business` ซึ่งจะทำหน้าที่จัดเก็บและเพิ่มเติมข้อมูลที่บรรจุอยู่ในอีลีเมนต์ `businessEntity` ให้ทันสมัย การจัดเก็บข้อมูลควรพิจารณาถึงข้อมูลที่มีอยู่เดิมก่อน เพื่อไม่ให้เกิดการจัดเก็บซ้ำซ้อนซึ่งอาจส่งผลกระทบต่อข้อมูลของธุรกิจทั้งหมดได้ ยกเว้นการจัดเก็บข้อมูลในครั้งแรก

- Save Service

เป็นการแทรกหรือปรับปรุงข้อมูลที่อยู่ในอีลีเมนต์ businessService และเมื่อมีการเปลี่ยนแปลงจะส่งผลกระทบต่ออีลีเมนต์ bindingTemplate และ tModel ด้วย

- Save Binding

เป็นการแทรกหรือปรับปรุงข้อมูลที่อยู่ในอีลีเมนต์ bindingTemplate และจะทำหน้าที่คอยดูแลอีลีเมนต์ tModel ด้วย

- Save tModel

เป็นการแทรกหรือเพิ่มเติมข้อมูลที่อยู่ในอีลีเมนต์ tModel

2.5.3.2 การลบ Service

มีลักษณะคล้ายกับการสร้างและปรับปรุง Service แต่การลบข้อมูลจำเป็นต้องใช้ UUID (Universal Unique Identifier) ซึ่ง UUID นั้นจะกำหนดขึ้น เมื่อมีการสร้างและแทรกโครงสร้างข้อมูลลงใน UDDI Registry โดยใช้อัลกอริทึมที่ได้มาตรฐานจาก ISO เป็นตัวกำหนด การลบบริการมี Message ที่ใช้ร่วมกับอีลีเมนต์ต่างๆทั้งหมด 4 ประเภท คือ Delete Business, Delete Service, Delete Binding และ Delete tModel อธิบายได้ดังนี้

- Delete Business

อยู่ในรูปแบบของอีลีเมนต์ delete_business ซึ่งจะให้ข้อมูลต่างๆที่เกี่ยวข้องกับการลบข้อมูลธุรกิจ โดยจะต้องใช้ UUID เพื่อยืนยันการลบข้อมูล

- Delete Service

อีลีเมนต์ delete_service จะใช้สำหรับการเข้ารหัสข้อมูลของบริการที่ต้องการลบ ซึ่งต้องการ UUID ที่ถูกต้องเพื่อให้การลบข้อมูลของบริการเป็นไปอย่างสมบูรณ์ การลบนี้จะมีผลต่ออีลีเมนต์ที่เกี่ยวข้องคือ Binding Template และ tModel สำหรับ Delete Binding และ Delete tModel จะจัดการข้อมูลในส่วนของ Binding และ tModel ตามลำดับ

2.5.4 การค้นหา Service (Service Discovery)

การค้นหา Service นี้จะเป็นส่วนที่ถูกค้าหรือผู้ใช้ทั่วไปสามารถเข้ามาใช้งานได้ โดยมีหน้าที่ในการค้นหาและแสดงรายละเอียดของ Service จาก UDDI Registry ซึ่งผู้ใช้งานจะต้องใช้งานผ่านแอปพลิเคชันที่สร้างขึ้นเพื่อค้นหาข้อมูลของ Service ต่างๆโดยจะใช้ SOAP เป็นตัวกลางในการติดต่อระหว่างแอปพลิเคชันและ Registry แอปพลิเคชันที่ใช้ในการค้นหา Service จะเลือก API ที่เหมาะสมมาช่วยสร้าง SOAP เพื่อใช้ค้นหา Service ซึ่งแบ่งได้เป็น 2 ระดับดังนี้

2.5.4.1 Information Browsing

เป็นการค้นหา Service จาก UDDI Registry ในระดับที่ไม่ลึกมาก ข้อมูลที่ได้จะเป็นรายละเอียดของ Service โดยไม่ต้องระบุตัวคนมี 4 ประเภท คือ Find Business, Find Service และ Find tModel อธิบายได้ดังนี้

- Find Business

อีลีเมนต์ `find_business` มีหน้าที่ในการระบุตำแหน่งของข้อมูลธุรกิจ ซึ่งอาจจะมากกว่าหนึ่งตำแหน่งก็ได้

- Find Binding

อีลีเมนต์ `find_binding` มีหน้าที่ในการระบุตำแหน่งการผูกรวมกันเอาไว้ของ `businessService` ที่ลงทะเบียนตั้งแต่หนึ่งตัวขึ้นไป

- Find Service

อีลีเมนต์ `find_service` มีหน้าที่ในการระบุตำแหน่ง Service ของ Business Entity ที่ลงทะเบียน

- Find tModel

อีลีเมนต์ `find_tModel` มีหน้าที่ในการระบุตำแหน่งของอีลีเมนต์ tModel ตั้งแต่หนึ่งตัวขึ้นไป

2.5.4.2 Information Drilldown

เป็นการค้นหา Service ในระดับรายละเอียดของ Service นั้นซึ่งมี 4 ประเภท ดังนี้

- Get Business Detail

อีลีเมนต์ `get_businessDetail` ทำหน้าที่ค้นหารายละเอียดของข้อมูลที่เกี่ยวข้องกับธุรกิจ ซึ่งจำเป็นต้องใช้คีย์เพื่อระบุตัวตนในการค้นหาด้วย

- Get Service Detail

อีลีเมนต์ `get_serviceDetail` ทำหน้าที่ค้นหารายละเอียดข้อมูลที่เกี่ยวข้องกับบริการ ซึ่งจำเป็นต้องใช้คีย์เพื่อระบุตัวตนในการค้นหาด้วย สำหรับ Get Binding และ Get tModel Detail จะใช้อีลีเมนต์ `get_bindingDetail` และ `get_tModelDetail` ตามลำดับซึ่งก็จำเป็นต้องใช้คีย์ในการระบุตัวตนเช่นกัน

2.6 ข้อกำหนดเพิ่มเติมที่สำคัญของ Web Service

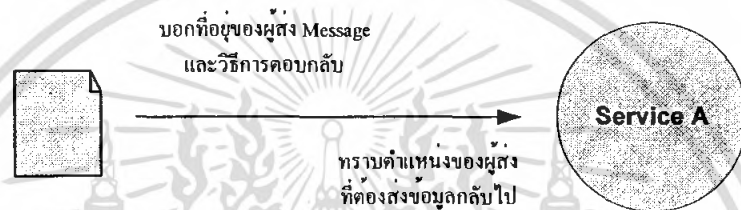
2.6.1 WS-Addressing

WS-Addressing ได้จัดเตรียมวิธีการขนส่งที่อยู่ (Address) ของ Web Service และ Message ที่เป็นกลางไว้หรือเรียกวิธีนี้ว่า “Addressing” ซึ่งข้อกำหนดนี้ถูกกำหนดไว้ใน SOAP Message โดยระบุถึง Web Service ที่อยู่ปลายทาง (Endpoint) และการแสดงตน (Identification) ของผู้รับและผู้ส่ง โดยการทำงานของ Addressing คล้ายกับการจัดทำ “ใบรายการสินค้าขนส่ง (Waybill)” เพื่อจัดส่งเอกสารใบเสนอราคาหรือใบแจ้งหนี้ที่ส่งไปยังลูกค้า เมื่อผู้ซื้อได้ชำระเงินเรียบร้อยแล้ว ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งของ (Shipping) กล่าวคือ ไม่ว่าจะจัดส่งสินค้าผ่านช่องทางใด หรือสินค้าจะมาจากคลังสินค้าใดก็ตาม จะต้องมีการ Waybill กำกับสินค้านั้นด้วย โดยจะบอกรายละเอียดต่างๆของสินค้า ดังนี้

1. สถานที่ที่สินค้าถูกส่งมา
2. สถานที่ที่สินค้าจะถูกส่งไป
3. บุคคลที่จะได้รับสินค้า
4. วิธีสำรองในการขนส่ง หากเกิดข้อผิดพลาดจากการขนส่งตามวิธีแรกที่ได้วางแผนไว้

ข้อกำหนดของ WS-Addressing มีลักษณะคล้ายกับการทำรายการสินค้าขนส่ง โดยการกำหนดรายละเอียดต่างๆจะอยู่ในส่วนของ SOAP Header



รูปที่ 2.14 WS-Addressing ที่ทำให้ Message กลายเป็นหน่วยอิสระในการติดต่อสื่อสาร

Addressing เป็นข้อกำหนดที่ประกอบด้วย 2 แนวคิดสำคัญ คือ Endpoint Reference และ Message Information Header

2.6.2 WS-Reliable Messaging

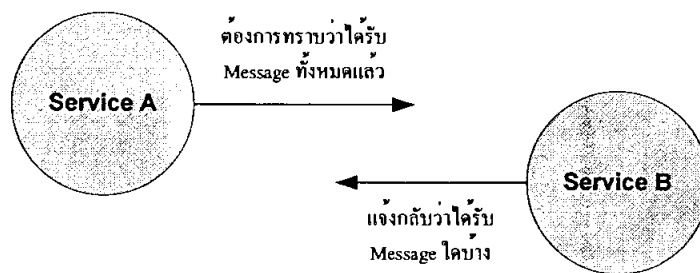
แม้ว่าการส่งข้อมูลในรูปแบบของ Messaging จะทำให้เกิดความเป็นอิสระต่อกัน (Loosely Coupled) ในการรับส่งข้อมูล แต่เมื่อส่ง Message ออกไปแล้วจะไม่สามารถควบคุมกระบวนการติดต่อสื่อสารของ Message ได้ ตัวอย่างของกระบวนการที่ไม่สามารถควบคุมได้ มีดังนี้

1. Message ได้ถูกส่งไปยังจุดมุ่งหมายปลายทางที่ถูกกำหนดไว้ครบถ้วนสมบูรณ์หรือไม่
2. เกิดความผิดพลาดในการส่ง Message และต้องการการส่งซ้ำหรือไม่
3. ลำดับการส่ง Message ได้เป็นไปตามลำดับที่ถูกกำหนดไว้หรือไม่

Reliable Message (RM) คือ มาตรการที่ถูกกำหนดขึ้นเพื่อเพิ่มความน่าเชื่อถือให้กับกระบวนการรับส่งข้อมูล โดยเป็นการรับรองว่า Message ที่ส่งไปถึงมือผู้รับอย่างถูกต้อง โดย WS-Reliable Messaging ทำให้เกิดกระบวนการที่สำคัญ คือ

1. Service Provider จะต้องแจ้งผลลัพธ์ของการส่ง Message กลับไป ไม่ว่าจะการส่ง Message นั้นจะเสร็จสมบูรณ์หรือไม่ก็ตาม
2. ผู้รับได้รับ Message ตามลำดับที่ระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 กระบวนการของ Reliable Messaging

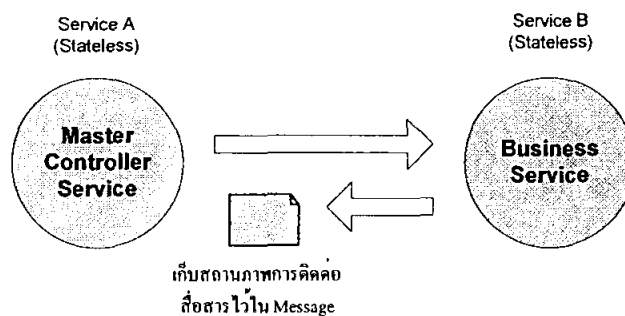
จุดประสงค์สำคัญของ WS-Reliable Message คือ การสร้างกลไกย่อยๆ เพื่อเพิ่มความน่าเชื่อถือให้กับการส่ง Message โดยระบุ Messaging Protocol และร่องรอยเส้นทางเดิน (Track) ของ Message รวมถึงควบคุมกิจกรรมต่างๆ ของ Message ที่ใช้รับส่งกันระหว่างต้นทางและปลายทาง เพื่อเพิ่มความน่าเชื่อถือให้กับ Message ยิ่งขึ้น

แม้ว่า WS-Reliable Messaging จะสามารถควบคุมกิจกรรม (Activity) ของ Service ได้ แต่ข้อกำหนดของ WS-Reliable Message ก็แตกต่างจากข้อกำหนดของการจัดการ Activity (Activity Management) กล่าวคือ Reliable Messaging ไม่ได้สร้าง Coordinator Service ขึ้นมาเพื่อควบคุมสถานะต่างๆ ที่เกิดขึ้นของ Activity แต่ใช้ SOAP Header ที่อยู่ภายใน Message มาควบคุมพฤติกรรมของ Message แทน

2.6.3 ความเกี่ยวเนื่องกัน (Correlation)

พื้นฐานของการแลกเปลี่ยนข้อมูลผ่าน Web Service คือ ความสามารถในการส่งรายละเอียด และสภาพแวดล้อมของ Message ต่างๆ ซึ่ง Service จะต้องอยู่บนพื้นฐานของความเป็นอิสระต่อกันอย่างแท้จริง นั่นคือ Response Message สามารถได้ตอบกับ Request Message ของ Service อื่นได้แบบอัตโนมัติ แต่การแลกเปลี่ยน Message ด้วยวิธี Primitive MEP (เช่น Request-Response) ไม่สามารถทำงานในลักษณะดังกล่าวได้ จึงเกิดแนวคิดของ Correlation ขึ้น

แนวคิดของ Correlation คือ Service ที่กระทำ Activity ร่วมกันไม่จำเป็นต้องรู้ถึงสถานะภาพในการแลกเปลี่ยนข้อมูลของตนเอง (หรือ Stateless) แต่บทบาทหน้าที่ และความสัมพันธ์ระหว่างกันของ Service ที่ใช้ในการแลกเปลี่ยน Message นั้น จะถูกเก็บอยู่ภายใน Message นั้น โดยตรง ทำให้ Service มีความเป็นอิสระต่อกันอย่างแท้จริง



รูปที่ 2.16 การเก็บสถานะจากการแลกเปลี่ยนข้อมูลไว้ใน Message

เมื่อ Service ส่ง Message ไปยัง Service อื่นแล้ว Service นั้นจะไม่สามารถควบคุม Message ที่ส่งออกไปได้ ดังนั้น จึงได้เสนอแนวคิดของ Correlation เพื่อให้ Service ทราบสถานะต่างๆของ Message ที่ถูกส่งออกไปได้ ด้วยการกำหนดข้อมูลเพิ่มเติมลงใน Message และส่งข้อมูลนั้นให้กับ Service ของ Message อื่นๆที่มีความเกี่ยวข้องกันได้รับทราบ เมื่อ Service ประมวลผล Message และทราบตำแหน่งของ Message แล้วจะกำหนด Context (หรือคำอธิบาย) เพื่อใช้เชื่อมโยงกับ Message อื่นๆ

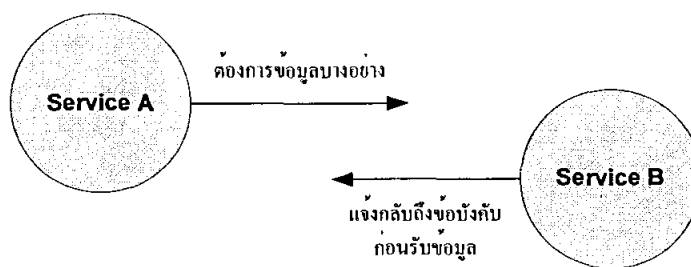
Correlation คือ กลไกสำคัญที่สนับสนุนหลักการ Autonomy และ Statelessness ของ Service กล่าวคือ Correlationทำให้ Message ของ Service สามารถเชื่อมต่อกับ Message อื่นๆได้ด้วยตนเอง โดยไม่ต้องร้องขอบริการใดๆจาก Service ทำให้ Service อยู่ในสถานะ Stateless เพื่อรองรับ Message อื่นๆเข้ามาประมวลผลได้

2.6.4 WS-Policy

Policy หรือ นโยบาย เป็นข้อกำหนดเพิ่มเติมที่ช่วยให้การดำเนินงานต่างๆเป็นไปตามกฎเกณฑ์ และข้อบังคับที่กำหนดไว้ Policy ถูกนำไปใช้เพื่อควบคุมการทำงานของ Service ต่างๆให้ เป็นไปตามแผนงานที่วางไว้โดยข้อกำหนดต่างๆที่กำหนดขึ้นจะพิจารณาจาก

1. ความต้องการทางธุรกิจของงานนั้นๆ
2. ปัจจัยพื้นฐานของการแลกเปลี่ยนข้อมูล
3. มาตรฐานความปลอดภัยขององค์กร

โดยทุก Message ของทุก Service จะมีคุณสมบัติพิเศษเฉพาะตัวที่ Service อื่นจำเป็นต้องทราบเพื่อปฏิบัติตาม เช่น คุณสมบัติที่เกี่ยวกับพฤติกรรมของ Service, สิทธิพิเศษต่างๆ, ข้อจำกัดทางเทคนิค และคุณสมบัติด้านคุณภาพของบริการ (Quality of Service: QoS) เป็นต้น ซึ่งคุณสมบัติต่างๆเหล่านี้ จะถูกนำมาใช้ในการแลกเปลี่ยนข้อมูลระหว่างกัน และถูกเก็บไว้ในส่วนของ Policy



รูปที่ 2.17 รูปแบบของ Policy

ประโยชน์ของ Policy คือ การทำให้ Service สามารถมีคุณสมบัติและสิทธิพิเศษต่างๆเพื่อควบคุมพฤติกรรมให้เป็นไปตามข้อกำหนดและกฎเกณฑ์ตามที่ต้องการได้ โดย Policy จะเป็นส่วนที่ไม่สามารถ Implement ได้ และจะถูกจัดเก็บแยกต่างหาก เพื่อให้สามารถจัดการกับคุณสมบัติต่างๆได้อย่างอิสระ

2.6.5 WS-Metadata Exchange

เมื่อ Service Requester ต้องการร้องขอบริการจาก Service Provider (ซึ่งทำหน้าที่เป็น Ultimate Receiver) จะต้องมีการ Service Description หรือเอกสาร WSDL ของ Service Provider ซึ่งจัดเตรียมข้อมูลต่างๆที่จำเป็นสำหรับการส่ง SOAP Message ไปยัง Service Provider โดย Web Service จะใช้ Metadata เพื่ออธิบายถึงรายละเอียด และข้อกำหนดในการติดต่อสื่อสารที่ Service Requester จำเป็นต้องทราบ โดยใช้ Policy แสดงความสามารถ (Capability) ความต้องการ (Requirement) และคุณสมบัติต่างๆไปของ Web Service นอกจากนี้ยังใช้เอกสาร WSDL เพื่ออธิบายรูปแบบของ Message โปรโตคอลที่ใช้และที่อยู่ของ Web Service โดยการแลกเปลี่ยนข้อมูลเหล่านี้จะต้องอาศัยการติดต่อกันระหว่าง Web Service โดยเรียกระบวนการนี้ว่าการแลกเปลี่ยนข้อมูลระหว่างกัน (Metadata Exchange)

2.6.6 WS-Security

ปัจจุบันความปลอดภัยของระบบถือเป็นหัวใจสำคัญของยุคข้อมูลข่าวสาร รวมถึงแอปพลิเคชันที่พัฒนาด้วย SOA จะต้องมามีเครื่องมือที่คอยควบคุม และป้องกันไม่ให้เกิดความเสียหายกับข้อมูล ตลอดจนต้องมีการตรวจสอบสิทธิ์ในการเข้าถึงทรัพยากรต่างๆของระบบอย่างละเอียดด้วย เพื่อให้มั่นใจได้ว่าข้อมูลที่สำคัญต่างๆจะไม่รั่วไหลไปสู่บุคคลภายนอก โดยข้อกำหนดเพิ่มเติมด้านความปลอดภัย (Security Specification) ที่นำมาใช้กับ SOA มีอยู่เป็นจำนวนมาก เช่น WS-Security, WS-SecurityPolicy, WS-Trust, WS-SecureConversation, WS-Federation และ WS-I Basic Security Profile เป็นต้น แต่มี 3 ข้อกำหนดที่สำคัญ คือ WS-Security, XML-Signature และ XML-Encryption

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจัยที่สนับสนุนแนวคิดของ Security มีอยู่ 5 อย่างได้แก่ การแสดงตน (Identification), การพิสูจน์ตัวตน (Authentication), การกำหนดสิทธิ์การใช้งาน (Authorization), การรักษาความลับ (Confidentiality) และความสมบูรณ์ (Integrity) ของข้อมูล

2.6.7 WS-Notification

การติดต่อสื่อสารระหว่าง Service Provider กับ Service Requester จะมีการสร้างความสัมพันธ์เฉพาะระหว่าง Service ทั้งสอง โดยมีการแลกเปลี่ยนข้อมูลที่เรียกว่า “การแจ้งเตือน (Notification)” เช่น Service A ต้องการให้ Service B ส่งข้อมูลการแจ้งเตือน (Notification) กลับมาเมื่อมีเหตุการณ์บางอย่างเกิดขึ้นกับ Service B ดังรูป



รูปที่ 2.18 Service A แจ้งให้ Service B ส่ง Notification

เมื่อ Service B เกิดเหตุการณ์ที่ Service A สนใจ Service B จะส่งการแจ้งเตือน (Notification) กลับไปยัง Service A ดังรูป



รูปที่ 2.19 Service B ส่ง “การแจ้งเตือน (Notification)”

บทที่ 3

ทฤษฎีที่เกี่ยวข้องกับฐานข้อมูลเชิงวัตถุ

ฐานข้อมูลเชิงวัตถุ OODB (Object-Oriented Database)

ในปัจจุบันการพัฒนาซอฟต์แวร์เชิงวัตถุ (OOP Object-Oriented Programming) ได้เข้ามามีบทบาทมากขึ้น ซึ่งถือได้ว่าเป็นการเปลี่ยนแปลงแนวคิดของการพัฒนาโปรแกรมแบบเก่าอย่างสิ้นเชิง โดยที่การเขียนโปรแกรมเชิงวัตถุนั้นจะมีรายละเอียดดังต่อไปนี้

3.1 องค์ประกอบของการพัฒนาโปรแกรมเชิงวัตถุ

3.1.1 อังอิงออบเจกต์ (OID)

Object Identity หรือ Object Identifier (OID) เป็นสิ่งที่ใช้ในการระบุหรือใช้ในการอ้างอิงถึงออบเจกต์ ซึ่งในแต่ละออบเจกต์จะมีค่า OID ไม่ซ้ำกัน เมื่อออบเจกต์ถูกสร้างขึ้น ระบบจะทำการกำหนดค่า OID ให้ และไม่สามารถเปลี่ยนแปลงค่า OID ภายหลังได้ และออบเจกต์นั้นถูกลบออกไปจากระบบ OID ของออบเจกต์นั้นจะถูกลบไปด้วย และจะไม่มีการนำค่า OID ที่ถูกลบไปแล้วกลับมาใช้ใหม่

3.1.2 คุณสมบัติ (Property)

Property คือข้อมูลของออบเจกต์ ซึ่งเป็นได้ทั้งข้อมูลชนิดพื้นฐานโดยทั่วไป ประกอบไปด้วย Integer, Float, String เป็นต้น ส่วนข้อมูลที่ซับซ้อนจะประกอบไปด้วยการนำเอาข้อมูลพื้นฐานมารวมเป็นโครงสร้างได้แก่ Array หรืออาจจะเป็นข้อมูลออบเจกต์ก็ได้

3.1.3 เมธอด (Method)

การตอบสนองของออบเจกต์ที่เก็บข้อมูลต่างๆ หรืออาจเรียกได้ว่าการตอบสนองของออบเจกต์นั้น จะถูกเรียกใช้งานผ่านทาง เมธอดเท่านั้น ตามคุณสมบัติการเอนแคปซูเลชัน (Encapsulation) โดยการเรียกใช้งานเมธอดก็เพื่อที่จะดูข้อมูลหรือการเปลี่ยนแปลงของคุณสมบัติของออบเจกต์นั้นๆ โดยที่เมธอดจะประกอบไปด้วย ชื่อเมธอด, รูปแบบการเรียกใช้, การทำงานของเมธอด

3.1.4 เมสเซจ (Message)

วัตถุสามารถที่จะแจ้งสถานะหรือข้อมูล ตลอดจนเปลี่ยนแปลงสถานะเมื่อมีการร้องขอจากภายนอก นอกจากนี้วัตถุยังสามารถตอบสนองได้ มีทั้งการเปลี่ยนสถานะของตน การตอบกลับ และการส่งข้อความไปยังวัตถุอื่นต่อ โดยเมสเซจ ที่ถูกส่งออกไปนี้จะเป็นตัวสั่งให้เมธอดในออบเจกต์อื่นๆทำงาน

3.1.5 คลาส (Class) และ อินสแตนซ์ (Instance)

กลุ่มของออบเจกต์ที่มีโครงสร้างภายในเหมือนกัน ตอบสนองต่อเมสเสจเดียวกัน โดยที่มีรายละเอียดของคุณสมบัติและรายละเอียดของเมธอดที่เหมือนกันชนิดตัวแปรเดียวกัน ในบางกรณีอาจพบว่าเราสามารถให้ออบเจกต์ตัวหนึ่งๆ เป็นตัวแทนของคลาสมากกว่าหนึ่งคลาสได้ โดยการสืบทอดคุณสมบัติ และในบางครั้งออบเจกต์เราอาจจะเรียกออบเจกต์ที่อยู่ต่างเวลากันว่า อินสแตนซ์ (Instance)

จากที่กล่าวมาทั้งหมดข้างต้นนั้น ก็องค์ประกอบของการพัฒนาโปรแกรมเชิงวัตถุ นอกจากนี้แล้วการพัฒนาโปรแกรมเชิงวัตถุยังมีแนวคิดที่แตกต่างออกไปจากเดิม ตามที่ได้กล่าวไปแล้วในข้างต้น ต่อไปนี้จะเป็นรายละเอียดของแนวคิดของการพัฒนาโปรแกรมเชิงวัตถุ ซึ่งมีทั้งหมดดังนี้

3.2 แนวคิดของการพัฒนาโปรแกรมเชิงวัตถุ

3.2.1 การห่อหุ้ม (Encapsulation)

เอนแคปซูเลชันคือรากฐานอย่างหนึ่งของแนวความคิดในเชิง Object-Oriented ซึ่งข้อดีของเอนแคปซูเลชัน คือการป้องกัน แอททริบิวต์ของ ออบเจกต์จากความเสียหาย เพราะถ้าส่วนของโปรแกรมทั้งหมด อนุญาตให้มีการเข้าถึง แอททริบิวต์ ได้ตามที่ต้องการแล้วนั้น จะส่งผลให้ แอททริบิวต์ นั้นง่ายต่อการถูกใช้อย่างผิดๆ ทำให้ค่า แอททริบิวต์ เปลี่ยนแปลงไป ซึ่งก่อให้เกิดความเสียหายตามมา กล่าวง่ายๆ คือ เอนแคปซูเลชัน คือการห่อหุ้ม แอททริบิวต์ และ เมธอด เข้าไว้ด้วยกัน โดยที่ เอนแคปซูเลชัน จะทำหน้าที่ป้องกันมิให้ ออบเจกต์อื่นที่อยู่ภายนอก เข้าถึงออบเจกต์หนึ่งๆ ได้อย่างอิสระ มีเฉพาะ เมธอด ที่อยู่ในออบเจกต์เท่านั้นจะสามารถติดต่อกับ แอททริบิวต์ ที่อยู่ในออบเจกต์เดียวกันได้ เรียกได้ว่าการ เอนแคปซูเลชัน มีคุณสมบัติของ Information hiding (การซ่อนข้อมูลและเมธอดส่วนตัว)



รูปที่ 3.1 การติดต่อกับออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.1 จะแสดงถึงการติดต่อกับออบเจกต์ที่จะต้องผ่านเมสเซจและ Return โดยจะไม่มี การเข้าถึงข้อมูลของออบเจกต์โดยตรง อย่างไรก็ตามแม้ว่าการเข้าถึงข้อมูลของออบเจกต์โดยตรงอาจจะ ผิดมารยาทของการเขียนโปรแกรมแบบ Object-Oriented แต่ตัวภาษาก็อาจจะยอมให้ชุดคำสั่งอื่นๆที่อยู่ภายนอกออบเจกต์เข้าถึงข้อมูลในออบเจกต์โดยตรงเลยก็ได้ คือการเข้าถึงข้อมูลโดยผ่าน เมสเซจ

3.2.2 การสืบทอด (Inheritance)

Inheritance หรือ การสืบทอดคุณสมบัติ กล่าวคือ คุณสมบัติที่คลาสๆ หนึ่ง สามารถสืบทอดลักษณะของ แอททริบิวต์ และ เมธอดของ อีกคลาสหนึ่งได้การทำเช่นนี้ทำให้คุณสามารถ สร้างคลาสใหม่ขึ้นโดยนำสาระสำคัญของ เหมือนกันของ แอททริบิวต์และ behavior(method) จากคลาสอื่นมาใช้ได้ จากการเขียนโปรแกรมแบบ Procedural จะทำการจัดหาส่วนของ โปรแกรมที่มีการใช้ซ้ำบ่อยๆ เพื่อนำกลับมาใช้ใหม่ (reuse) ในรูปของโปรแกรมย่อย (Procedure หรือ Function) ซึ่งเราจะสามารถเรียกใช้ซ้ำได้หลายครั้งตามต้องการ อย่างไรก็ตาม ใน Object-Oriented Programming มีคุณสมบัติข้อนี้เช่นกัน แต่ยังมีกรอบที่พิเศษมากขึ้นกว่าการ reuse คือ ความสามารถในการจัดการกับ คลาสและกลุ่มของคลาสต่างๆ โดยคุณสมบัติที่เรียกว่า “Inheritance” ที่ประกอบด้วยหลักการของ Super class/Subclass, Abstraction และ is-a relationship ซึ่งหลักการต่างๆ เหล่านี้จะช่วยให้การออกแบบ โครงสร้างโปรแกรมแบบ Object-Oriented ทำได้ง่ายขึ้น

3.2.3 ภาวะหลายรูปแบบ (Polymorphism)

Polymorphism หมายถึง การบอกแบบเดียว แต่ได้รับการตอบสนองหลายรูปแบบ (Poly=“many”, morph = “form”) ซึ่งเป็นไปตามหลักการของเทคโนโลยีเชิงวัตถุที่กล่าวมาแล้ว คือจะเน้นที่ตัวปฏิบัติการมากกว่าการปฏิบัติการ ดังนั้นรายละเอียดหรือหน้าที่ต่างๆ จะขึ้นอยู่กับเราว่าจะไปมีความสัมพันธ์กับตัวปฏิบัติการเหล่านั้นอย่างไร โดยที่ Polymorphism ใช้กับการสืบทอดคลาสลูกกับคลาสแม่อาจมีเมธอดชื่อเดียวกัน แต่ทำงานไม่เหมือนกัน

จากทั้งหมดที่กล่าวมานั้นจะทำให้เกิดข้อดีของการพัฒนาโปรแกรมเชิงวัตถุคือ

1. สามารถพัฒนาโปรแกรมได้ง่าย และยังสามารถแก้ไขได้ง่าย เช่นกัน
2. สามารถนำคลาสที่เราเขียนขึ้นมา นำกลับมาใช้ได้อีก (Reuse)

จากข้อดีของการพัฒนาโปรแกรมเชิงวัตถุบางข้อที่กล่าวมาข้างต้น ทำให้เกิด ฐานข้อมูลเชิงวัตถุ (OODB) ซึ่งข้อดีของฐานข้อมูลเชิงวัตถุนั้นก็คือ สามารถที่จะจัดการกับข้อมูลที่มีความซ้ำซ้อน รวมไปถึงฐานข้อมูลที่มีขนาดใหญ่ มากๆ ได้ ซึ่งในปัจจุบันนี้ได้มีผู้พัฒนาเทคโนโลยีการจัดการฐานข้อมูลแบบใหม่ โดยนำเอาข้อดีของฐานข้อมูลเชิงวัตถุ และข้อดีของฐานข้อมูลเชิงสัมพันธ์ ซึ่งมีใช้อยู่กันอย่างแพร่หลายทั้งในอดีตและปัจจุบัน เข้ามาประยุกต์รวมกันเป็นระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลเชิงวัตถุสัมพันธ์ (ORDBMS; Object-Relational Database Management System) โดยระบบฐานข้อมูลแบบนี้กำลังได้รับความนิยมอย่างสูง ทำให้บริษัทที่พัฒนาซอฟต์แวร์ระบบการจัดการฐานข้อมูลหลายแห่ง ได้เพิ่มเติมความสามารถในการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์นี้ให้กับผลิตภัณฑ์ของตนเอง โดยอ้างอิงตามมาตรฐาน SQL3 และยังสามารถพัฒนาระบบจัดการฐานข้อมูลเชิงวัตถุ (OODBMS, Object-Oriented Database Management System)

โดยทั่วไปการเก็บข้อมูลภายในระบบฐานข้อมูลเชิงวัตถุนั้นจะเก็บเป็นลำดับชั้น(Hierarchy) ตามความสัมพันธ์ที่ได้กำหนดไว้ ซึ่งการเข้าถึงข้อมูลภายในระบบฐานข้อมูลเชิงวัตถุนั้นสามารถใช้การเขียนโปรแกรมเพื่อเข้าไปใช้โครงสร้างของอ็อบเจกต์ในฐานข้อมูลได้ทันที ส่งผลให้ไม่จำเป็นต้องมีการแปลงข้อมูลเหมือนกับฐานข้อมูลชนิดอื่นๆ

อย่างไรก็ตามถึงแม้ว่าการใช้งานฐานข้อมูลเชิงวัตถุ จะเป็นทางเลือกที่เหมาะสมสำหรับซอฟต์แวร์ที่พัฒนาขึ้นด้วยเชิงวัตถุ แต่ฐานข้อมูลเชิงวัตถุยังมีข้อด้อยอยู่เหมือนกัน เช่น เทคโนโลยีของ ODBMS ยังเป็นเรื่องใหม่ทำให้มีแหล่งข้อมูลในการศึกษาไม่มากนัก ทำให้เวลาในการพัฒนาโปรแกรมกินเวลานานมากกว่าปกติได้

จากที่กล่าวมาทั้งหมดข้างต้นนั้นเป็นแนวคิดเกี่ยวกับฐานข้อมูลเชิงวัตถุและแนวคิดของการพัฒนาโปรแกรมเชิงวัตถุ ต่อไปจะเป็นรายละเอียดการทำงานของฐานข้อมูลเชิงวัตถุซึ่งมีดังนี้

3.3 รายละเอียดการทำงานของฐานข้อมูลเชิงวัตถุ

3.3.1 โครงสร้างข้อมูล

การใช้งานข้อมูลในฐานข้อมูลเชิงวัตถุจะอยู่ในรูปแบบของ คลาสซึ่งเป็นแนวความคิดพื้นฐานของทฤษฎีทาง Object-Oriented โดยที่ในคลาสจะประกอบด้วยแอททริบิวต์ และ เมธอด รวมไปถึง Integrity Constrain ของคลาส นั้นๆ ซึ่งจะมีการกำหนด Object Identifier (OID) เอาไว้เพื่อแบ่งแยกความแตกต่างระหว่าง คลาสนอกจากนี้ยังสนับสนุนการทำงานแบบเอนแคปซูเลชัน และ อินเฮอริแตนซ์ ด้วยสำหรับชนิดของข้อมูลนั้นสามารถรองรับชนิดข้อมูลแบบ Abstract Data Type ได้ จากมาตรฐานของ ODMG ได้มีการจำแนกประเภทของวัตถุเป็น 2 ชนิดคือ Object และ Literal รวมไปถึงการแบ่งแยกประเภทคุณลักษณะของวัตถุเอาไว้เป็น 2 ชนิดคือ Operation และ Property

เนื่องจากฐานข้อมูลเชิงวัตถุนี้เกิดขึ้นจากการรวมเอาพื้นฐานแนวความคิดเชิงวัตถุ กับหลักการด้านภาษาสำหรับการ โปรแกรมเชิงวัตถุ และความสามารถด้านฐานข้อมูลเข้าด้วยกัน ผลลัพธ์ที่ได้คือทำให้เกิดความเข้ากันได้เป็นอย่างดีระหว่างโครงสร้าง ข้อมูลภายในฐานข้อมูล กับโครงสร้างข้อมูลที่ใช้ในการพัฒนาโปรแกรม ซึ่งอำนวยความสะดวกในการใช้ภาษาสำหรับการ โปรแกรมเชิงวัตถุต่างๆ เพื่อพัฒนาโปรแกรมที่ใช้งานร่วมกับฐานข้อมูลเชิงวัตถุ ทำให้สามารถลดจำนวนโค้ดที่ใช้ในการเขียน โปรแกรมต่างๆลงได้

3.3.2 ภาษาที่ใช้ในการจัดการฐานข้อมูล

ในฐานข้อมูลเชิงวัตถุจะมีภาษาสำหรับโปรแกรมเชิงวัตถุ ทั้งในการจัดการฐานข้อมูลและการพัฒนาโปรแกรม เนื่องจาก Application Object กับ Stored Object ในระบบฐานข้อมูลชนิดนี้มีความสัมพันธ์กันโดยตรง ดังนั้นการทำ Data Definition, Data Manipulation และ Query จึงสามารถทำได้ด้วยภาษาสำหรับการโปรแกรมเชิงวัตถุดังกล่าว นอกจากนั้นแล้ว ตามมาตรฐาน ODMG-93 ได้มีการกำหนด Declarative Language ขึ้นมาอีกภาษาหนึ่งคือ OQL (Object Query Language) ซึ่งจะมีความแตกต่างกันออกไปจาก SQL ตามมาตรฐาน SQL2 เพราะในมาตรฐาน SQL2 ไม่มีความสามารถในการเข้าถึงข้อมูลแบบวัตถุได้ อย่างไรก็ตามในมาตรฐานใหม่ของ SQL3 จะได้มีบางส่วนที่จะได้เพิ่มความสามารถเหล่านี้ลงไป ซึ่งผลจากการพัฒนา OQL จะสามารถสร้างความหลากหลายของผลลัพธ์ที่เกิดขึ้นจากการทำ Query ได้ อันได้แก่ ผลลัพธ์ที่อยู่ในรูปของ Atom, Structure, Literal, Object หรือแม้แต่ Set ของ Object

3.3.3 กลไกที่ใช้ในการเข้าถึงข้อมูล

การสร้างและแก้ไขข้อมูลในระบบฐานข้อมูลเชิงวัตถุ จะใช้การเข้าถึงโดยตรงจากภาษาสำหรับการโปรแกรมเชิงวัตถุในลักษณะของ Native Language สำหรับออบเจกต์ที่ถูกสร้างขึ้นในระบบฐานข้อมูลเชิงวัตถุนี้จะถูกกำหนด OID ให้โดยอัตโนมัติ และ OID นี้จะไม่ซ้ำกันไปตลอดอายุการใช้งานออบเจกต์นั้นๆ นอกจากนี้ในตัวออบเจกต์แต่ละออบเจกต์ยังสามารถจัดเก็บ OID ของออบเจกต์ตัวอื่น เพื่อสร้างเป็น Logical Reference ซึ่ง Reference นี้เอง จะเป็นประโยชน์ในการสร้างความสัมพันธ์ระหว่างออบเจกต์ที่เกิดขึ้นตามโลกแห่งความเป็นจริงที่สามารถแบ่งตามกฎเกณฑ์ได้ เช่น Bidirectional Relationship, Versioning, Composite Object และ Distribution ในระบบฐานข้อมูลเชิงวัตถุบางระบบจะใช้ OID ในลักษณะทาง Physical ด้วย กล่าวคือ จะทำการแปลงรูปจาก Logical Identifier ให้มาเป็น Pointer เพื่ออ้างอิงในการเข้าถึงข้อมูล เมื่อข้อมูลถูกนำมาเก็บไว้ใน Memory (cached) เพื่อการใช้งานในโปรแกรมเชิงวัตถุที่ถูกสร้างขึ้น โดยที่การแปลง Reference ดังกล่าวมาเป็น Memory Pointer เรียกว่า Pointer Swizzling ซึ่งจะทำให้การใช้เวลาในการเข้าถึงข้อมูลน้อยลง เพราะเป็นการเข้าถึงข้อมูลใน Cache จึงมีความเร็วในการเข้าถึงข้อมูลในระดับของเมมสเชจ แทนการใช้การส่งเมสเชจผ่านไปยังเซอร์เวอร์ในระบบเก่า

3.3.4 ประโยชน์และข้อดีของฐานข้อมูลเชิงวัตถุได้แก่

- การคิดแบบธรรมชาติ
- การกระทำที่ไม่ขึ้นตรงกับระบบฐานข้อมูลซึ่งสามารถเคลื่อนย้ายได้
- ชนิดข้อมูลของฐานข้อมูลเชิงวัตถุ สามารถนำไปใช้กับข้อมูลที่ซับซ้อนได้คืออย่างเช่น ข้อมูลรูปภาพ, Audio/Video และข้อมูล Multimedia อื่นๆ

3.4 OODBMS (Object-Oriented Database Management System)

เพื่อกำจัดขอบเขตของ RDBMS ในปี 1980 โปรแกรมเมอร์ได้พัฒนาระบบฐานข้อมูลเชิงวัตถุขึ้นมา ซึ่งจุดประสงค์หลักของการคิดค้นระบบฐานข้อมูลเชิงวัตถุขึ้นมานั้นก็เพื่อ ต้องการข้อมูลที่เป็นอิสระ, ความปลอดภัย, ควบคุมและให้บริการจัดการแก่ฐานข้อมูลเชิงวัตถุ อีกทั้งยังจัดการกับข้อมูลที่มีขนาดใหญ่และมีความซับซ้อนสูงได้ดีอีกด้วย

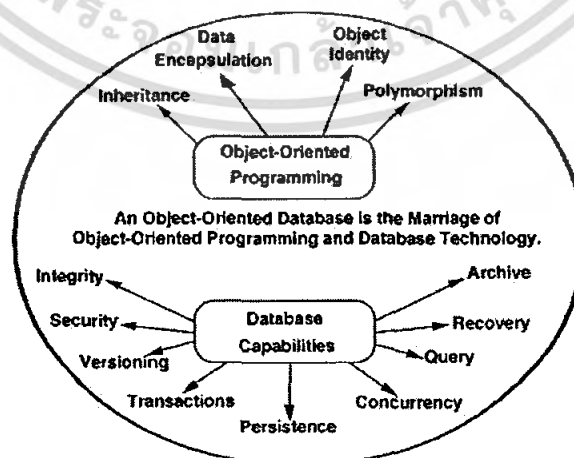
ลักษณะสำคัญของฐานข้อมูลเชิงวัตถุนั้น คือการนำเอาการพัฒนาโปรแกรมเชิงวัตถุและเทคโนโลยีทางด้านฐานข้อมูลเอาไว้ด้วยกัน ซึ่งการพัฒนาโปรแกรมเชิงวัตถุนั้นจะได้ผลลัพธ์ออกมาด้วยกัน 4 ลักษณะคือ การสืบทอด (Inheritance), การห่อหุ้มข้อมูล (Encapsulation), เอกลักษณ์ของวัตถุ (Object Identity) และ ภาวะหลายรูปแบบ (Polymorphism) เหมือนที่เคยกล่าวไว้ข้างต้นแล้ว

3.4.1 สถาปัตยกรรมของการใช้ OODBMS

ในการใช้ OODBMS นั้นเราสามารถแบ่งสถาปัตยกรรมของซอฟต์แวร์ออกได้เป็น 2 แบบ คือ สถาปัตยกรรมสำหรับแอปพลิเคชันที่ทำงานเดี่ยว (Stand-Alone Application) และสถาปัตยกรรมซึ่งใช้ OODB ร่วมกับระบบข้อมูลที่มีอยู่เพื่อช่วยเพิ่มประสิทธิภาพ แนวคิดของสถาปัตยกรรมแบบนี้เราเรียกอีกอย่างหนึ่งว่า Persistent Cache สำหรับแอปพลิเคชันที่ได้รับการพัฒนาขึ้นโดยใช้ภาษา C หรือ Java ทั้งนี้ก็เนื่องจากระบบฐานข้อมูลเชิงวัตถุ จะใช้ในการเก็บข้อมูลซึ่งมักจะถูกเรียกใช้งานบ่อย

3.4.2 Characteristics of Object-Oriented Databases in Depth

ดังที่กล่าวมาแล้วข้างต้นที่ว่าฐานข้อมูลเชิงวัตถุ ได้รวมเอาแนวคิดเชิงวัตถุผสมผสานกับเทคโนโลยีฐานข้อมูล จากรูปที่ 3.2 เป็นการแสดงแนวคิดของฐานข้อมูลและการพัฒนา โปรแกรมที่เรียกว่า ฐานข้อมูลเชิงวัตถุ



รูปที่ 3.2 การรวมแนวคิดเชิงวัตถุเข้ากับระบบจัดการฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประโยชน์ในการศึกษาเท่านั้น เมื่อผู้ใดนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต หรือใช้เพื่อวัตถุประสงค์อื่นที่ไม่ใช่ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางที่อาจเรียกได้ว่า คุณลักษณะที่สำคัญของเทคโนโลยีฐานข้อมูลเชิงวัตถุก็คือการรวมกันระหว่างการพัฒนาโปรแกรมเชิงวัตถุและเทคโนโลยีฐานข้อมูล

3.5 ODMG

ในปัจจุบันมาตรฐาน ODMG สำหรับระบบฐานข้อมูลเชิงวัตถุได้รับการกำหนดและปรับปรุงจนถึงเวอร์ชัน 3.0 แล้ว มาตรฐาน ODMG ได้รับการกำหนดขึ้นโดยกลุ่มของผู้พัฒนาผลิตภัณฑ์และผู้สนใจที่เข้ามาทำงานร่วมกัน ในการสร้างข้อกำหนดสำหรับผลิตภัณฑ์ระบบฐานข้อมูลเชิงวัตถุ และฐานข้อมูลเชิงวัตถุสัมพันธ์

ODMG 3.0 เป็นข้อกำหนดกลางที่ได้รับการออกแบบมาสำหรับแอปพลิเคชันที่ทำงานกับผลิตภัณฑ์ระบบจัดการฐานข้อมูลเชิงวัตถุ ให้สามารถทำงานร่วมกับผลิตภัณฑ์อื่นๆ ได้ข้อกำหนด ODMG 3.0 สนับสนุนภาษา Java, C++, Smalltalk ในการติดต่อกับระบบฐานข้อมูล โดยส่วนประกอบที่มีความสำคัญในข้อกำหนด ODMG 3.0 นั้นประกอบไปด้วย

3.5.1 โมเดลออบเจกต์ (Object Model)

สำหรับโมเดลข้อมูลพื้นฐานที่ได้รับการสนับสนุนโดยข้อกำหนด ODMG นั้นมีพื้นฐานอยู่บน OMG Object Model สำหรับโมเดลของ OMG เองนั้นได้รับการออกแบบมาสำหรับ Object Request Broker, ระบบฐานข้อมูลเชิงวัตถุ, ภาษาในการเขียนโปรแกรมรวมถึงแอปพลิเคชันอื่นๆ ด้วยการใช้องค์ประกอบโมเดลข้อมูลของ OMG ส่งผลให้นักพัฒนาที่คุ้นเคยกับแอปพลิเคชันของ OMG สามารถใช้งานผลิตภัณฑ์ที่เข้ากันได้กับมาตรฐาน ODMG ได้ง่าย

ภาษาข้อกำหนดของออบเจกต์ (Object Specification Language) ข้อกำหนด ODMG มีภาษาข้อกำหนดอยู่ 2 ภาษาคือ Object Definition Language (ODL) และ Object Interchange Format (OIF) สำหรับภาษา ODL นั้นเป็นภาษาข้อกำหนดที่มีพื้นฐานบนภาษา OMG IDL ส่งผลให้สามารถใช้ภาษา ODL ในการกำหนดชนิดของออบเจกต์ที่เข้ากันได้กับโมเดล ODMG Object Model ได้ ส่วนภาษา OIF เป็นภาษาข้อกำหนดที่ใช้ในการเก็บและโหลดออบเจกต์เข้ามาจากไฟล์

3.5.2 ภาษาในการคิวรีออบเจกต์ (Object Query Language)

เป็นภาษาแบบ Declarative (Nonprocedural) หรือที่เรียกว่าภาษาแบบ 4GL ใช้สำหรับการค้นหาและแก้ไขเปลี่ยนแปลงออบเจกต์ สำหรับภาษา OQL นี้มีพื้นฐานมาจากภาษา SQL-92

3.5.3 การใช้ง่ายร่วมกับภาษา C++ (C++ Language Binding)

เป็นรูปแบบการเชื่อมต่อระหว่างแอปพลิเคชันที่เขียนขึ้นด้วยภาษา C++ กับผลิตภัณฑ์ ODBMS ที่เข้ากันได้กับข้อกำหนด ODMG โดยเราเรียกรูปแบบทั้งหมดว่า C++ OML (Object Manipulation Language) ซึ่งประกอบด้วยภาษา ODL ในเวอร์ชันที่ใช้ไวยากรณ์ของ C++, กลไกในการเรียก OQL และกระบวนการสำหรับการประมวลผลและทำการทรานแซคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 การใช้งานร่วมกับภาษา Smalltalk (Smalltalk Language Binding)

เช่นเดียวกับภาษา C++ คือทำหน้าที่กำหนดรูปแบบในการเรียกใช้งานระหว่าง ODL กับ ภาษา Smalltalk

3.5.5 การใช้ร่วมกับภาษาจาวา (Java Language Binding)

เป็นการกำหนดการทำงานระหว่างโมเดล ODMG Object Model (ODL และ OML) กับ ภาษาจาวาตามข้อกำหนด Java 2 Platform และเช่นเดียวกัน ในรูปแบบจะประกอบไปด้วยกลไกการ เรียก OQL และกระบวนการสำหรับการประมวลผลและทรานแซกชัน

ภายใต้ข้อกำหนด ODMG ผู้พัฒนาสามารถใช้แอปพลิเคชันที่ได้รับการเขียนขึ้นด้วยภาษา C++, Smalltalk หรือ ภาษาจาวาในการเรียกใช้งานข้อมูลเดียวกันได้ ซึ่งชนิดของข้อมูลออบเจกต์ที่ปรากฏในแต่ละภาษาก็จะเหมือนกันด้วย อย่างไรก็ตามข้อกำหนดของ ODMG ยังมีข้อแตกต่างจากภาษา SQL ในระบบฐานข้อมูลเชิงสัมพันธ์ ทั้งนี้เนื่องจากภาษาในการประมวลผลข้อมูลของ ODMG นั้นจะได้รับการกำหนดให้ขึ้นอยู่กับภาษาที่ใช้ในการเขียน โปรแกรมของแอปพลิเคชัน ทั้งนี้เพื่อให้ นักพัฒนามีสภาวะแวดล้อมในการพัฒนาที่ครบถ้วนและเป็นหนึ่งเดียว กล่าวง่ายๆ ก็คือ ถ้าเป็นภาษา SQL นักพัฒนาสามารถใช้ภาษา SQL เดียวกันในการค้นหาและแก้ไขเปลี่ยนแปลงข้อมูล แต่ถ้าเป็น ODMG แล้ว ภาษาที่นักพัฒนาใช้ติดต่อกับระบบฐานข้อมูลจะขึ้นอยู่กับภาษาที่ใช้เขียน โปรแกรม ซึ่งแอปพลิเคชันที่ใช้ภาษา C++ ก็จะแตกต่างจากแอปพลิเคชันที่ใช้ภาษาจาวาเป็นต้น

บทที่ 4

การวิเคราะห์และออกแบบระบบ

4.1 บทนำ

การทำงานของระบบการชำระค่าปรับทางด้านการจราจร เป็นการทำงานในลักษณะการเรียกใช้บริการผ่านกระบวนการเว็บเซอร์วิส(Web Service) การทำงานของระบบชำระค่าปรับทางด้านการจราจรนี้จะเริ่มจากเจ้าพนักงานจราจรป้อนข้อมูลความผิดลงไปในระบบ ระบบก็จะตรวจเช็คข้อมูลผู้กระทำผิดจากแอปพลิเคชันกรมการปกครอง และ ข้อมูลยานพาหนะจากกรมการขนส่งทางบกทันที ถ้าหากไม่มีปัญหาระบบก็จะนำข้อมูลความผิดไปประมวลผลและคำนวณค่าปรับรอการชำระ ถ้าหากมีการหักแต้มหรืออายุใบขับขี่ก็ส่งไปให้แอปพลิเคชันกรมการขนส่งทางบกเป็นผู้รับผิดชอบ ส่วนเรื่องการชำระค่าปรับผู้ใช้งานระบบสามารถตรวจสอบข้อมูลความผิดและจำนวนค่าปรับได้ผ่านทางแอปพลิเคชันของสำนักงานตำรวจแห่งชาติ และชำระค่าปรับกับทางธนาคารได้เลยโดยไม่ต้องไปที่สถานีตำรวจ เมื่อชำระเสร็จแล้วทางธนาคารจะส่งข้อมูลการชำระเงินไปให้กับทางสำนักงานตำรวจแห่งชาติอัตโนมัติ

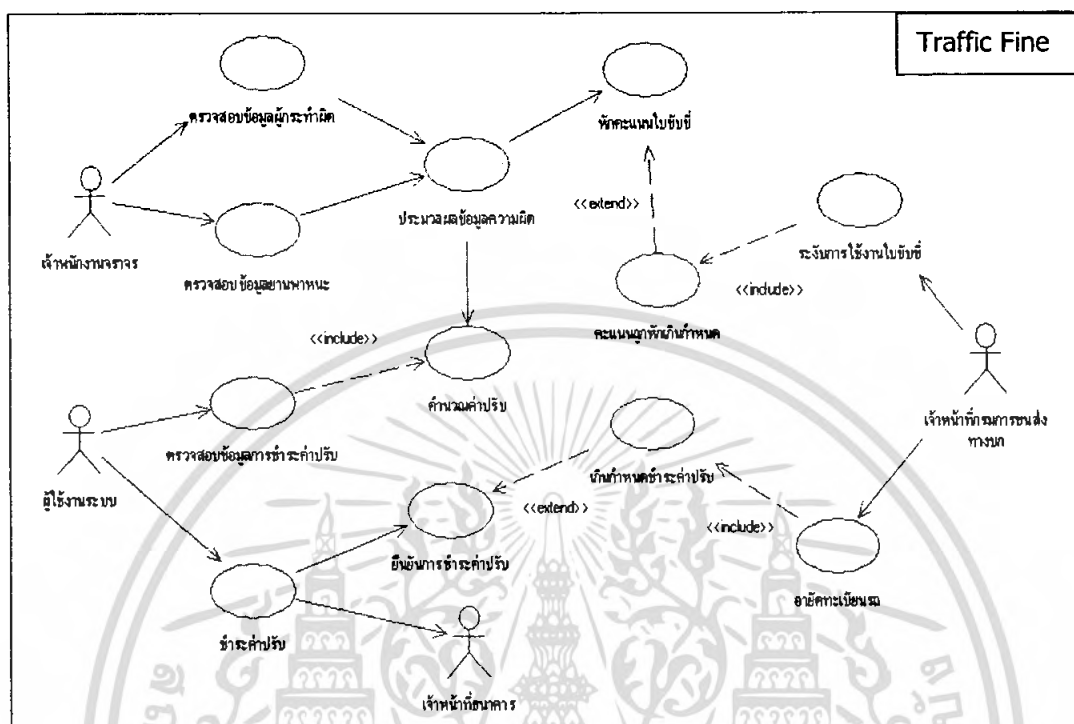
จากการวิเคราะห์ปัญหาที่เกิดขึ้นจากการชำระค่าปรับแบบเดิมคือ ในส่วนของผู้กระทำผิดจะเสียเวลาในการไปเสียค่าปรับและถ้าหากถูกยึดใบขับขี่ไปตัดคะแนนกว่าจะได้คืนต้องรออีกหลายวัน ส่วนเจ้าหน้าที่ตำรวจจะต้องมาวุ่นวายกับเรื่องเงินค่าปรับและการส่งใบขับขี่ไปตัดคะแนน เป็นต้น โดยระบบการชำระค่าปรับทางด้านการจราจรนี้จะช่วยในการลดปัญหาต่างๆดังที่กล่าวนี้คือจะเป็นระบบที่ชำระค่าปรับและหักคะแนนใบขับขี่แบบออนไลน์ ทำให้ไม่เสียเวลาอีกต่อไป

4.2 สิ่งที่ต้องการในระบบ

1. สามารถตรวจสอบข้อมูลผู้กระทำผิดได้
2. สามารถตรวจสอบข้อมูลยานพาหนะได้
3. คำนวณค่าปรับและคำนวณคะแนนที่หักได้
4. ผู้ใช้งานสามารถตรวจสอบข้อมูลความผิดของตนเองได้
5. ชำระค่าปรับได้
6. ตรวจสอบการชำระค่าปรับ
7. ระวังการใช้งานใบขับขี่
8. อายุคทะเบียนรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 Use Case Diagram ของระบบ



รูปที่ 4.1 Use Case Diagram ของระบบ

จากรูปที่ 4.1 เป็น Use Case Diagram ของระบบทั้งหมดมีรายละเอียดต่างดังนี้
Actor ของระบบได้แก่

- ผู้ใช้งานระบบ คือ บุคคลที่เข้ามาใช้งานระบบสามารถทำกิจกรรมได้ 2 อย่างคือ ตรวจสอบข้อมูลการชำระค่าปรับและชำระค่าปรับ
- เจ้าหน้าที่ธนาคาร ทำหน้าที่จัดการเกี่ยวกับการชำระเงินของผู้ใช้งานระบบ
- เจ้าหน้าที่กรมการขนส่งทางบก ทำหน้าที่ระงับ ใบขับขี่และอายัดทะเบียนรถ
- เจ้าหน้าที่งานจราจร ทำหน้าที่ตรวจสอบข้อมูลผู้กระทำผิดและข้อมูลยานพาหนะก่อนจะส่งให้กับระบบการประมวลผลความผิด

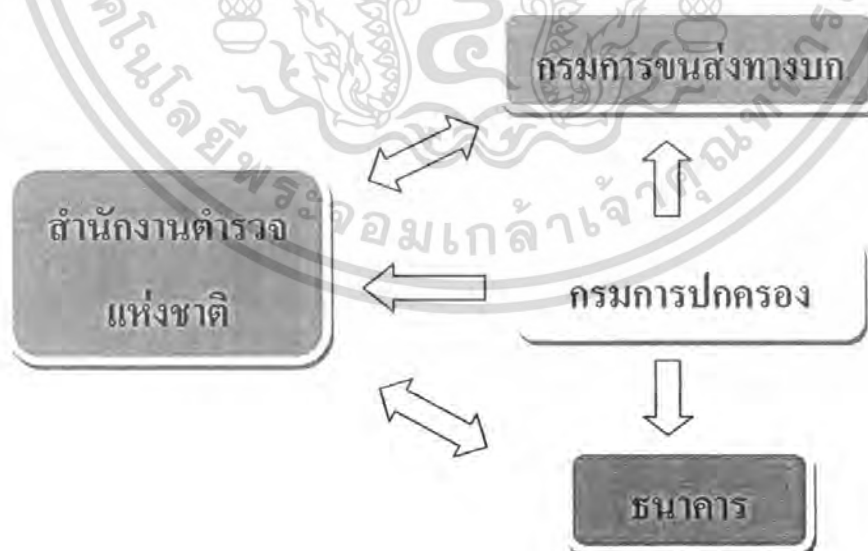
Use Case ของระบบได้แก่

- ตรวจสอบข้อมูลผู้กระทำผิด ทำหน้าที่ตรวจสอบข้อมูลส่วนตัวจากกรมการปกครอง
- ตรวจสอบข้อมูลยานพาหนะ ทำหน้าที่ตรวจสอบข้อมูลของยานพาหนะที่ถูกจับจากกรมการขนส่งทางบก
- ประมวลผลข้อมูลความผิด ทำหน้าที่ เป็นตัวจัดการเกี่ยวกับความผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กำหนดค่าปรับ ทำหน้าที่ กำหนดเงินที่ผู้กระทำความผิดต้องชำระ โดยดูจากความผิด
- หักคะแนนใบขับขี่ ทำหน้าที่ หักคะแนนในใบขับขี่ของผู้ใช้งานระบบ
- คะแนนถูกหักเกินกำหนด ทำหน้าที่ ตรวจสอบคะแนนใบขับขี่ว่าเกินกว่า 60 คะแนนหรือไม่ ถ้าเกินจะสั่งให้ระงับการใช้งานใบขับขี่
- ระงับการใช้งานใบขับขี่ ทำหน้าที่ ระงับใบขับขี่ชั่วคราวถ้าหากคะแนนถูกหักเกิน 60 คะแนน
- ตรวจสอบข้อมูลการชำระค่าปรับ ทำหน้าที่ ในการไปตรวจสอบข้อมูลการชำระค่าปรับของผู้ใช้งานระบบคนนั้น
- ชำระค่าปรับ ทำหน้าที่ ให้บริการกับผู้ใช้ระบบในการเสียค่าปรับ
- ยืนยันการชำระค่าปรับ คือจะทำการเช็คที่ฐานข้อมูลของสำนักงานตำรวจแห่งชาติว่าจ่ายแล้ว
- เกินกำหนดชำระค่าปรับ ทำหน้าที่ ตรวจสอบว่าเกินกำหนดที่จะมาชำระค่าปรับหรือยัง
- อายัดทะเบียน ทำหน้าที่ อายัดทะเบียนรถถ้าหากไม่ชำระค่าปรับภายในกำหนด

4.4 Web service Model



รูปที่ 4.2 การบริการข้อมูลซึ่งกันและกันของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.2 เป็นการบริการข้อมูลซึ่งกันและกันของระบบทั้งหมด คือ ระหว่างสำนักงานตำรวจแห่งชาติกับกรมการปกครอง, สำนักงานตำรวจแห่งชาติกับกรมการขนส่งทางบก, สำนักงานตำรวจแห่งชาติกับธนาคาร, และกรมการขนส่งทางบกกับกรมการปกครอง

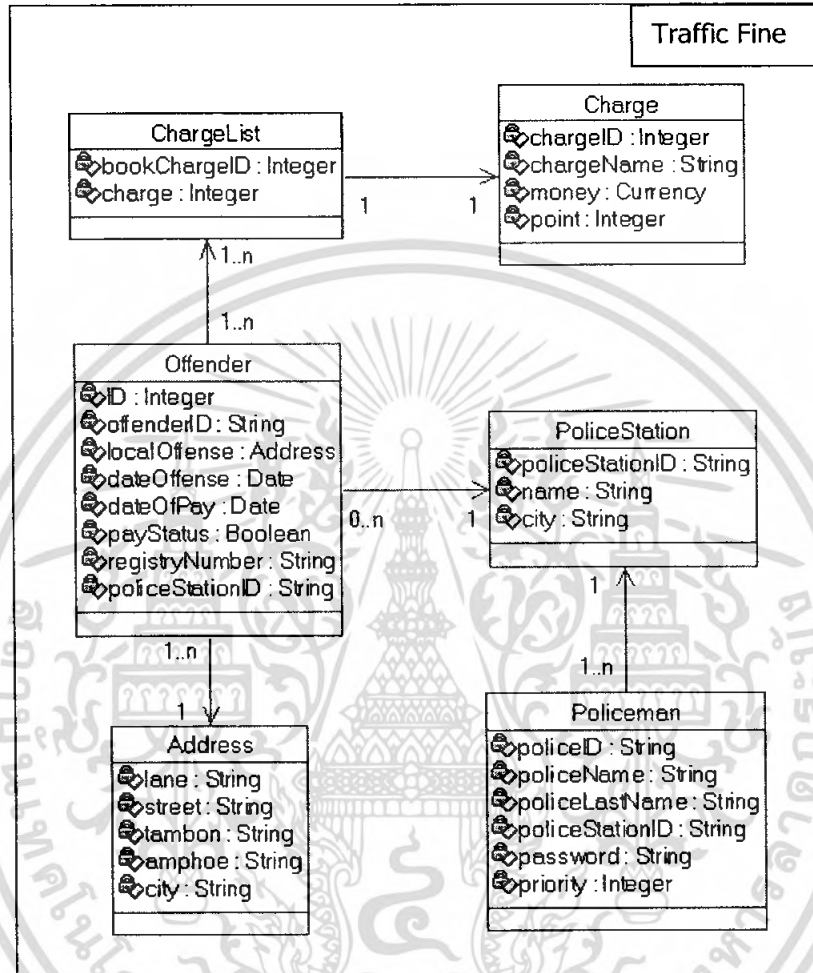
โดยเริ่มจากส่วนของกรมการปกครองที่ให้บริการข้อมูลให้สำนักงานตำรวจแห่งชาติมาตรวจสอบรายละเอียดของผู้กระทำความผิด และให้บริการกับกรมการขนส่งทางบกคือให้มาตรวจสอบข้อมูลเจ้าของรถได้

ส่วนการบริการข้อมูลระหว่างสำนักงานตำรวจแห่งชาติและกรมการขนส่งทางบกนั้น จะเป็นการบริการข้อมูลให้สำนักงานตำรวจแห่งชาติไปตรวจสอบข้อมูลยานพาหนะได้ โดยส่งแค่หมายเลขทะเบียนรถไปตรวจสอบมี 2 ส่วนด้วยกันคือ บริการข้อมูลของใบอนุญาตขับขี่ซึ่งระบุว่าถูกกระทำการใช้งานอยู่หรือไม่ และส่วนที่ 2 คือ บริการข้อมูลของรถว่าถูกอายัดอยู่หรือไม่ และสิ่งที่บริการให้กับกรมการขนส่งทางบก ซึ่งเป็นข้อมูลที่เกี่ยวข้องกับข้อมูลของผู้กระทำความผิดที่กรมการขนส่งทางบกจำเป็นต้องใช้ในการหักแต้มของใบอนุญาตขับขี่ ถ้าหากมีแต้มที่ถูกหักเกินกำหนดกรมการขนส่งทางบกจะทำการระงับการใช้งานบัตรทันที และอีกกรณีคือถ้าทางสำนักงานตำรวจแห่งชาติไม่ได้รับข้อมูลการชำระเงินค่าปรับจากรถในระยะเวลา 7 วัน ระบบจะบริการข้อมูลในส่วนนี้ให้กับกรมการขนส่งทางบกเพื่อทำการอายัดทะเบียนรถ

ส่วนธนาคารจะบริการข้อมูลการชำระค่าปรับ ในกรณีที่ผู้กระทำความผิดมาชำระค่าปรับเรียบร้อยแล้วประกอบไปด้วย สาขาของธนาคารที่ชำระเงิน วันและเวลาที่ทำการชำระเงิน ส่วนในทางสำนักงานตำรวจแห่งชาติจะบริการข้อมูลของผู้กระทำความผิดว่ามีข้อหาอะไรและต้องชำระค่าปรับเป็นจำนวนเท่าไร

4.5 Class Diagram ของระบบ

4.5.1 Class Diagram ของระบบสำนักงานตำรวจแห่งชาติ



รูปที่ 4.3 Class Diagram ของระบบสำนักงานตำรวจแห่งชาติ

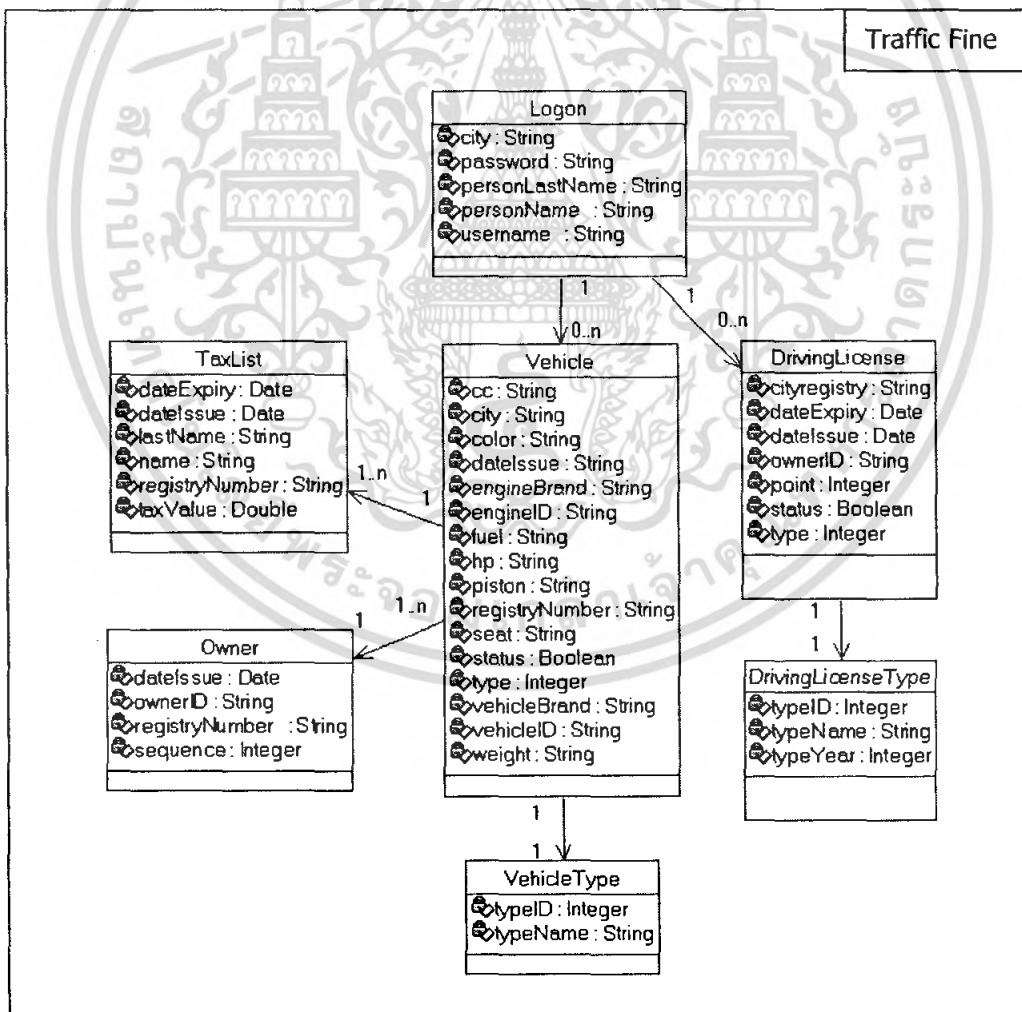
จากรูปที่ 4.3 เป็นคลาสไดอะแกรมของระบบสำนักงานตำรวจแห่งชาติ ซึ่งประกอบด้วย คลาส ต่างๆดังนี้

- PoliceStation คือ คลาสที่เก็บข้อมูลสถานีตำรวจประกอบด้วย หมายเลขสถานีตำรวจ ชื่อ สถานีตำรวจ และชื่อจังหวัดที่สถานีนั้นตั้งอยู่
- Policeman คือ คลาสที่เก็บข้อมูลของเจ้าหน้าที่ตำรวจ ประกอบด้วยหมายเลขของ เจ้าหน้าที่ตำรวจ ชื่อนามสกุล หมายเลขของสถานีตำรวจที่ประจำการอยู่ รหัสผ่านในการ ล็อกอินเข้าระบบ และความสำคัญของเจ้าหน้าที่แต่ละคน โดยมี 0 กับ 1 คือ “1” จะสามารถ จัดการกับระบบได้มากกว่าความสำคัญ “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Offender คือคลาสที่เก็บข้อมูลของผู้กระทำความผิด ประกอบด้วย รหัสของผู้กระทำความผิด วันที่เกิดเหตุ สถานที่เกิดเหตุ วันสุดท้ายที่สามารถชำระเงินได้ หมายเลขทะเบียนรถ รหัสสถานีตำรวจ และสถานะของการชำระเงิน
- Address คือคลาสที่เก็บรายละเอียดของที่เกิดเหตุโดยประกอบด้วย ซอย ถนน ตำบล อำเภอ จังหวัด
- ChargeList คือคลาสที่เก็บข้อมูลข้อมูลของใบสั่ง ประกอบด้วย หมายเลขใบสั่ง และหมายเลขของข้อกล่าวหา
- Charge คือคลาสที่เก็บรายละเอียดของข้อกล่าวหาต่างๆ ประกอบด้วย หมายเลขข้อกล่าวหา ชื่อข้อกล่าวหา จำนวนเงินที่ต้องชำระ และคะแนนที่จะต้องหัก ของแต่ละข้อกล่าวหา

4.5.2 Class Diagram ของระบบการขนส่งทางบก



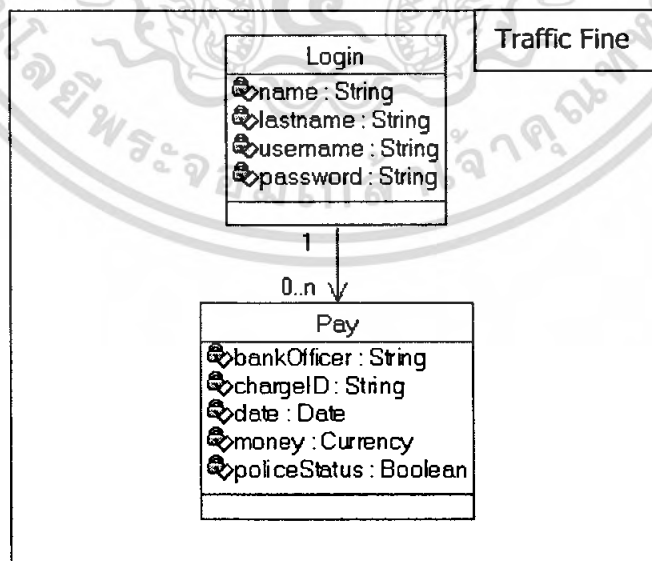
รูปที่ 4.4 Class Diagram ของระบบการขนส่งทางบก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่อผู้ใดเห็นจำเป็นต้องใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.4 เป็นคลาสไดอะแกรมของระบบการขนส่งทางบก ซึ่งประกอบด้วยคลาสต่างๆดังนี้

- Login คือ คลาสที่เก็บข้อมูลสำหรับการล็อกอินเข้าระบบประกอบด้วย ชื่อ นามสกุล ยูสเซอร์ พาสเวิร์ด และจังหวัดที่เจ้าหน้าที่คนนั้นประจำอยู่
- DrivingLicense คือ คลาสที่เก็บข้อมูลของใบขับขี่ ประกอบด้วย จังหวัดที่ออกใบขับขี่ วันออกบัตร วันหมดอายุ หมายเลขเจ้าของบัตร ประเภทของใบขับขี่ คะแนนของใบขับขี่ และสถานะที่บ่งบอกถึงว่าบัตรถูกอายัดหรือยัง
- DrivingLicenseType คือคลาสที่เก็บประเภทของใบขับขี่ประกอบด้วย หมายเลขประเภท ชื่อประเภท และ ประเภทที่ปึงของใบขับขี่
- Vehicle คือคลาสที่เก็บข้อมูลยานพาหนะ โดยจะเก็บข้อมูลรายละเอียดของรถทั้งหมดตามคู่มือรถ
- VehicleType คือคลาสที่เก็บข้อมูลประเภทของยานพาหนะ ประกอบด้วยหมายเลข และชื่อของประเภท เช่น รถบรรทุก 10 ล้อ รถจักรยานยนต์ เป็นต้น
- Owner คือ คลาสที่เก็บรายละเอียดของเจ้าของรถ ประกอบด้วย วันที่จดทะเบียน หมายเลขบัตรประชาชน หมายเลขทะเบียนยานพาหนะ และลำดับของเจ้าของเพราะยานพาหนะ 1 คันสามารถมีเจ้าของรถได้หลายคน
- TaxList คือคลาสที่เก็บข้อมูลของการต่อภาษีของรถ

4.5.3 Class Diagram ของระบบธนาคาร



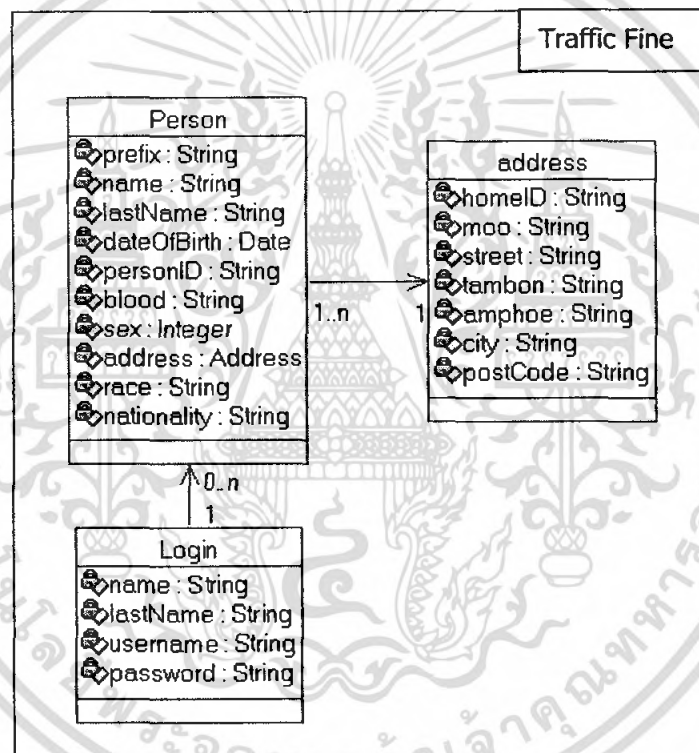
รูปที่ 4.5 Class Diagram ของระบบธนาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.5 เป็นคลาสไดอะแกรมของระบบธนาคาร ซึ่งประกอบด้วย 2 คลาสดังนี้

- Login คือคลาสที่เก็บข้อมูลของการล็อกอิน เข้าใช้งานระบบของเจ้าหน้าที่ ซึ่งประกอบด้วย ชื่อ นามสกุล ยูสเซอร์ และพาสเวิร์ด
- Pay คือ คลาสที่เก็บข้อมูลของการชำระค่าปรับ ประกอบด้วย รหัสของธนาคารที่ไปชำระค่าปรับ หมายเลขใบสั่ง วันที่ชำระค่าปรับ จำนวนเงินที่ชำระ สถานที่ส่งไปบอกการชำระค่าปรับของผู้กระทำผิดให้กับ สำนักงานกรมตำรวจแห่งชาติ ว่าสำเร็จหรือยัง

4.5.4 Class Diagram ของระบบกรมการปกครอง



รูปที่ 4.6 Class Diagram ของระบบกรมการปกครอง

จากรูปที่ 4.6 เป็นคลาสไดอะแกรมของระบบธนาคาร ซึ่งประกอบด้วย 3 คลาสดังนี้

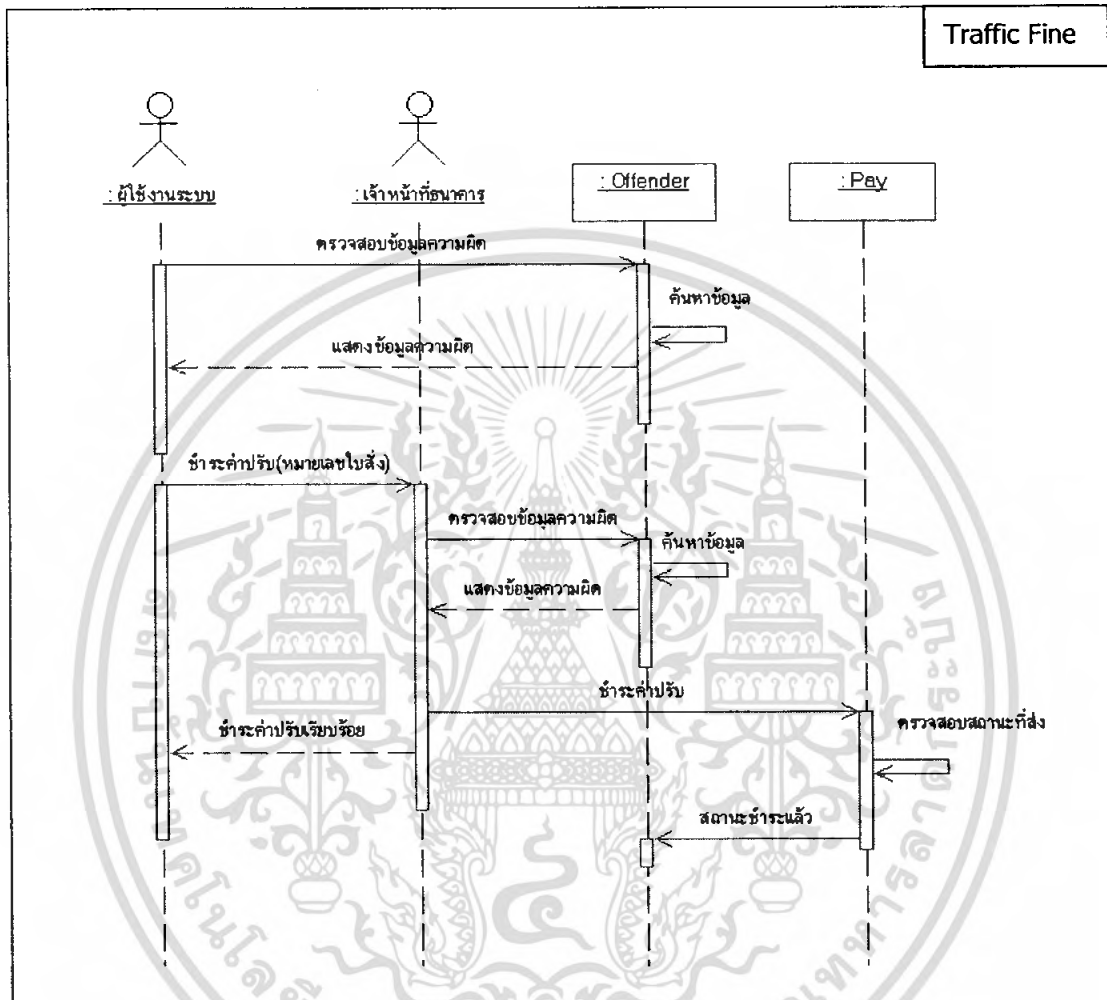
- Login เป็นคลาสที่เก็บข้อมูลการเข้าสู่ระบบของเจ้าหน้าที่กรมการปกครอง ซึ่งประกอบด้วย ชื่อ นามสกุล ยูสเซอร์ และ พาสเวิร์ด
- Person เป็นคลาสที่เก็บข้อมูลส่วนตัวทั้งหมด โดยมีรายละเอียดทุกอย่างตามบัตรประชาชน

- Address เป็นคลาสที่เก็บรายละเอียดของที่อยู่ โดยมีความสัมพันธ์กับ Person

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในทางอื่นใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 Sequence Diagram ของระบบ

4.6.1 Sequence Diagram การตรวจสอบและชำระค่าปรับ

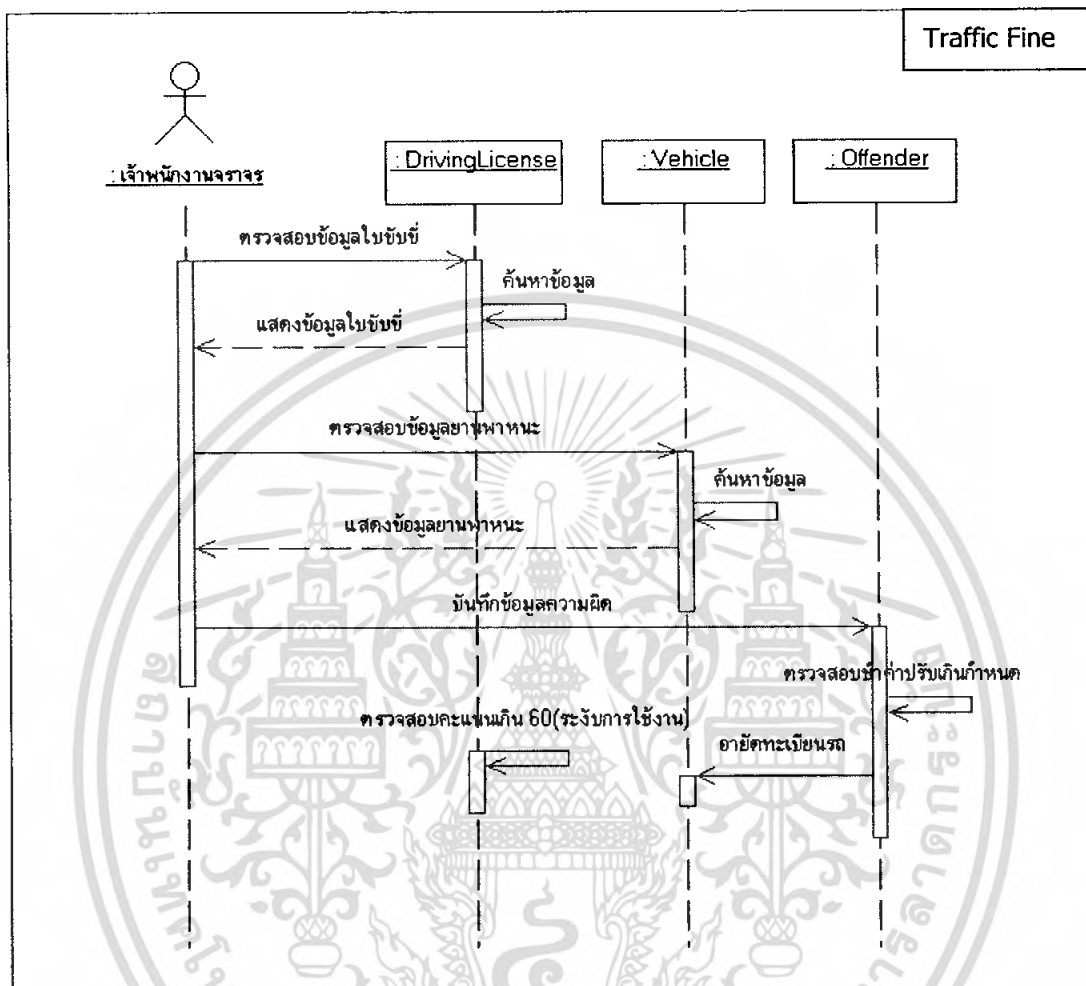


รูปที่ 4.7 Sequence Diagram การตรวจสอบและชำระค่าปรับ

จากรูปที่ 4.7 เป็นการแสดงลำดับการทำงานของ การตรวจสอบและชำระค่าปรับ โดยเริ่มจากผู้ใช้งานระบบเข้าไปตรวจสอบข้อมูลความผิดพร้อมทั้งจำนวนเงินที่ต้องชำระทั้งหมดจากคลาสข้อมูลผู้กระทำความผิด (Offender) จากนั้นก็จะแสดงข้อมูลให้ผู้ใช้งานระบบดู และส่วนการชำระค่าปรับสามารถชำระได้กับทางเจ้าหน้าที่ธนาคาร โดยนำหมายเลขใบสั่งให้กับเจ้าหน้าที่ธนาคาร จากนั้นเจ้าหน้าที่ธนาคารจะไปตรวจสอบความผิดจากคลาสข้อมูลผู้กระทำความผิด (Offender) ว่ามีข้อมูลความผิดของใบสั่งนี้จริงหรือไม่ เพื่อที่จะทำการชำระค่าปรับ ถ้าหากชำระเสร็จแล้วจะส่งข้อมูลสถานะไปบันทึกที่คลาสข้อมูลผู้กระทำความผิด (Offender) ว่าชำระค่าปรับเรียบร้อยแล้ว แต่ถ้าเกิดว่า

ติดต่อกับฐานข้อมูล Offender ไม่ได้ก็จะมีตัวตรวจสอบอยู่เรื่อยๆจนกว่าจะบันทึกข้อมูลได้ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 Sequence Diagram การบันทึกข้อมูลความผิด



รูปที่ 4.8 Sequence Diagram การบันทึกข้อมูลความผิด

จากรูปที่ 4.8 เป็นการแสดงลำดับการทำงานในส่วนของผู้ใช้งานจราจร โดยขั้นแรกเจ้าพนักงานจราจรจะกรอกข้อมูลหมายเลขบัตรประชาชนและหมายเลขทะเบียนรถ จากนั้นก็นำข้อมูลไปตรวจสอบโดยหมายเลขบัตรประชาชน ตรวจสอบกับคลาส DrivingLicense เพื่อดูว่าใบขับขี่นี้ถูกระงับการใช้งานหรือไม่ ส่วนหมายเลขทะเบียนรถจะตรวจสอบกับคลาส Vehicle เพื่อดูว่ายานพาหนะนี้ถูกอายัดหรือไม่ ถ้าหากว่าข้อมูลที่ตรวจสอบปกติทั้ง 2 อย่าง ก็จะบันทึกข้อมูลความผิด ลงคลาส Offender และ คลาส Offender นี้จะตรวจสอบการชำระค่าปรับว่าเกินกำหนดการชำระค่าปรับแล้วหรือยัง ถ้าเกินจะอายัดทะเบียนรถทันที ส่วนการระงับการใช้งานใบขับขี่คลาส DrivingLicense จะตรวจสอบว่าจะเกินถูกหักเกิน 60 คะแนนหรือไม่ ถ้าเกินจะทำการระงับทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การใช้งานระบบ

5.1 เว็บบกรมการปกครอง

กรมการปกครอง
Department of provincial Administration

Menu
Home
Add
Search

username:
password:

รูปที่ 5.1 หน้าล็อกอินเว็บกรมการปกครอง

กรมการปกครอง
Department of provincial Administration

เว็บไซต์โดย กรรวิจ สุริยา อภิชากรชนน

Menu
Home
Add
Search

ทำหนังสือ	นาย
ชื่อ	นามสกุล
นามสกุล	คำอื่น
หมายเลขประจำตัวประชาชน	1550500014454
เพศ	ชาย
วัน-เดือน-ปีเกิด	16 - ธันวาคม - 2527
หมู่เลือด	AB
เชื้อชาติ	ไทย
สัญชาติ	ไทย
บ้านเลขที่	124
หมู่ที่	8
ถนน	รวมการ
ตำบล	บัว
อำเภอ	บัว
จังหวัด	น่าน
รหัสไปรษณีย์	55120

รูปที่ 5.2 หน้ากรอกข้อมูลทะเบียนราษฎร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น เมื่อผู้ใช้ได้ให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรมการปกครอง
Department of provincial Administration

เข้าสู่ระบบโดย **กรรวิศ สุจิตา อรรถวาทะระบบ**

Menu
Home
Add
Search

Search by **หมายเลขบัตรประชาชน** Keyword **1550500014454**

Result 1 found

1. นายภูวดล คำธำนิ รหัสบัตรประชาชน:1550500014454 วันเกิด:16 ธันวาคม พ.ศ.2527
เพศ:ชาย หมู่โลหิต:AB เชื้อชาติ:ไทย สัญชาติ:ไทย
บ้านเลขที่:124 หมู่:8 ถนน:วรมนตร์ ตำบล:บึง
อำเภอ:บึง จังหวัด:พิจิตร รหัสไปรษณีย์ 55120

รูปที่ 5.3 หน้าตรวจสอบข้อมูลทะเบียนราษฎร

เว็บกรมการปกครองสามารถใช้งานได้เฉพาะเจ้าหน้าที่กรมการปกครองเท่านั้น ดังนั้นระบบจึงมีส่วนของล็อกอินดังรูปที่ 5.1 เพื่อตรวจสอบผู้ใช้งานระบบว่าเป็นเจ้าหน้าที่จริงหรือไม่ ส่วนรูปที่ 5.2 เป็นหน้ากรอกข้อมูลทะเบียนราษฎรสามารถทำได้โดยเจ้าหน้าที่เท่านั้น และ รูปที่ 5.3 เป็นตรวจสอบข้อมูลทะเบียนราษฎรโดยสามารถตรวจสอบได้ 3 อย่างคือ ตรวจสอบจากหมายเลขบัตรประชาชน ชื่อ และนามสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 เว็บกรมการขนส่งทางบก

The screenshot shows the website interface for the Department of Land Transport (DLT). The header includes the logo and name in Thai and English, along with navigation links: Home, Product, Service, Contact Us. The main content area is titled 'เพิ่มข้อมูลใบอนุญาตขับขี่' (Add Information License). It features a sidebar with navigation options: เข้าสู่ระบบ (Login), ตรวจสอบ (Check), and เพิ่มประเภท (Add Category). The main form area includes a 'กรุณกรอกข้อมูล' (Please enter information) section with fields for license number (1550500014454), license type (ใบอนุญาตขับรถส่วนบุคคล - 5 ปี), and province (กรุงเทพมหานคร). A 'ตกลง' (OK) button is present. Below the form, there is a section for 'ตรวจสอบข้อมูลใบอนุญาตขับขี่' (Check License Information) with a 'ตรวจสอบ' (Check) button. The bottom part of the page displays the license holder's details: นายประจักษ์ จงสวัสดิ์, born 1550500014454, residing at 124 M. 8, S. 3, S. 3, Bangkok, 16th floor, S. 2527, license issued 10/10/2551, valid until 10/10/2556, and license type 60 (Motor Vehicle).

รูปที่ 5.4 หน้าเพิ่มข้อมูลใบขับขี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่ให้ผู้อื่นเห็น เมื่อผู้ยื่นคำขอไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กรมการขนส่งทางบก
Department of Land Transport
www.dlt.go.th

Home Product Service Contact Us

เข้าสู่ระบบ

เข้าสู่ระบบโดย เดลิน อู่ท่าไม้
เจ้าหน้าที่จังหวัด น่าน
[Logout](#)

ใบอนุญาตขับขี่

เพิ่มข้อมูล
 ตรวจสอบ
 เพิ่มประเภท

ยานพาหนะ

เพิ่มข้อมูล
 ตรวจสอบ
 เพิ่มประเภท
 เพิ่มเจ้าของ
 เสียภาษี

ตรวจสอบข้อมูลใบอนุญาตขับขี่

หมายเลขบัตรประชาชน

เลขบัตรประชาชน 1550500014454
 50 ภูทอก นนทบุรี ตำบล
 ที่อยู่ 124 ม.8 ต.ปิว อ.ปิว จ.น่าน
 ปีได้รับที่ 16 ธันวาคม พ.ศ.2527
 ชนิด ใบอนุญาตขับรถจักรยานยนต์ส่วนบุคคล
 ครอบงวน 12 กุมภาพันธ์ พ.ศ.2551
 วันสิ้นอายุ 12 กุมภาพันธ์ พ.ศ. 2556
 นายประสิทธิ์จังหวัด น่าน
 แห่งกระทรวง - 10 แห่ง
 หมายเหตุ **ถูกต้อง**

รูปที่ 5.6 หน้าตรวจสอบข้อมูลใบขับขี่กรณีถูกต้อง



กรมการขนส่งทางบก
Department of Land Transport
www.dlt.go.th

Home Product Service Contact Us

เข้าสู่ระบบ

เข้าสู่ระบบโดย เดลิน อู่ท่าไม้
เจ้าหน้าที่จังหวัด น่าน
[Logout](#)

ใบอนุญาตขับขี่

เพิ่มข้อมูล
 ตรวจสอบ
 เพิ่มประเภท

ยานพาหนะ

เพิ่มข้อมูล
 ตรวจสอบ
 เพิ่มประเภท
 เพิ่มเจ้าของ
 เสียภาษี

กรอกข้อมูล

เลขทะเบียน	<input type="text" value="822"/>
ประเภท	<input type="text" value="รถจักรยานยนต์ส่วนบุคคล"/>
ผู้ครอบครอง	<input type="text" value="HONDA"/>
สี	<input type="text" value="แดง"/>
เลขตัวรถ	<input type="text" value="CT00P-0138557"/>
ผู้ถือครองแบบ	<input type="text" value="HONDA"/>
เลขเครื่องแบบ	<input type="text" value="C100FE-0138557"/>
เมื่อเสร็จ	<input type="text" value="เบนซิน"/>
จำนวนสูบ	<input type="text" value="1"/>
จำนวนซีวี	<input type="text" value="97"/>
จำนวนแรงม้า	<input type="text" value="-"/>
บ้านพัก	<input type="text" value="-"/>
ที่ตั้ง	<input type="text" value="2"/>

รูปที่ 5.7 หน้าเพิ่มข้อมูลยานพาหนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กรมการขนส่งทางบก
Department of Land Transport
www.dlt.go.th

Home Product Service Contact Us

เข้าสู่ระบบ

เข้าสู่ระบบโดย ประสิทธิ์ บุณศิริจะ
เจ้าหน้าที่จังหวัด กรุงเทพมหานคร
[Logout](#)

ชำระภาษีรถหมายเลขทะเบียน 2ฉ1967

ค่าภาษี 2500.0

หมายเลขทะเบียน ประสิทธิ์ - บุณศิริจะ

ใบอนุญาตขับขี่

- [เพิ่มข้อมูล](#)
- [ตรวจสอบ](#)
- [เพิ่มประเภท](#)

ยานพาหนะ

- [เพิ่มข้อมูล](#)
- [ตรวจสอบ](#)
- [เพิ่มประเภท](#)
- [เพิ่มเจ้าของ](#)
- [เสียภาษี](#)

รูปที่ 5.8 หน้าการเสียภาษี



กรมการขนส่งทางบก
Department of Land Transport
www.dlt.go.th

Home Product Service Contact Us

เข้าสู่ระบบ

เข้าสู่ระบบโดย ประสิทธิ์ บุณศิริจะ
เจ้าหน้าที่จังหวัด กรุงเทพมหานคร
[Logout](#)

ตรวจสอบข้อมูลรถ

หมายเลขทะเบียน 2ฉ - 1967

ใบอนุญาตขับขี่

- [เพิ่มข้อมูล](#)
- [ตรวจสอบ](#)
- [เพิ่มประเภท](#)

รายการทะเบียน
วันจดทะเบียน 10 มีนาคม พ.ศ.2551
เลขทะเบียน 2ฉ1967
จังหวัด กรุงเทพมหานคร
ประเภท รถยนต์นั่งส่วนบุคคลเกิน 7 คน
ยี่ห้อ TOYOTA
สี ขาว
เลขตัวรถ LN85-0021741
ยี่ห้อเครื่องยนต์ TOYOTA
เลขเครื่องยนต์ 2L-2186898
เชื้อเพลิง คาร์บ
จำนวนสูบ 4
จำนวนซีซี 2446
จำนวนแรงม้า 89
น้ำหนักสด 1500
ที่นั่ง 11
สถานะ ไม่ถูกอาชิต

เจ้าของรถ
ลำดับที่ 1
วันที่จดทะเบียนรถ 10 มีนาคม พ.ศ.2551
ผู้ครอบครอง ภูวดล คำอิน
เลขที่บัตร 1550500014454
วันที่เกิด 16 ธันวาคม พ.ศ.2527
ที่อยู่ 124 หมู่ 8 ตำบล/แขวง ปิ้ว อําเภอ/เขต ปิ้ว จังหวัด น่าน

รายการเสียภาษี
วันเสียภาษี 10 มีนาคม พ.ศ.2551
วันครบกำหนดเสียภาษี 10 มีนาคม พ.ศ.2552
ค่าภาษี 2500.0

รูปที่ 5.9 หน้าตรวจสอบข้อมูลยานพาหนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บของกรมการขนส่งทางบก สามารถใช้งานได้เฉพาะเจ้าหน้าที่กรมการขนส่งเท่านั้น โดยการใช้งานเริ่มจากเจ้าหน้าที่ต้องล็อกอินด้วย ยูสเซอร์ และ พาสเวิร์ด ก่อน จากรูปที่ 5.4 เป็นการเพิ่มข้อมูลใบขับขี่ โดยจะต้องใส่หมายเลขบัตรประชาชน และ ประเภทของใบขับขี่ จากนั้นรูปที่ 5.5 จะเป็นหน้าตรวจสอบข้อมูลใบขับขี่ โดยจะมีข้อมูลส่วนตัวไปเรียกเซอร์วิสจากเว็บกรมการปกครอง เพื่อเอาข้อมูลมาแสดง และจะบอกถึงส่วนของวันออกบัตร วันหมดอายุของใบขับขี่ด้วย ส่วนรูปที่ 5.6 เป็นการตรวจสอบกรณีทีใบขับขี่ถูกลบ

รูปที่ 5.7 เป็นหน้าเพิ่มข้อมูลยานพาหนะ โดยจะต้องป้อนข้อมูลของยานพาหนะนั้นๆ เมื่อเพิ่มข้อมูลรถไปแล้วต้องมีการเพิ่มเจ้าของรถโดยเจ้าของรถสามารถเพิ่มได้หลายคน โดยจะเรียงลำดับไปเรื่อยๆ รูปที่ 5.8 จะเป็นหน้าของการชำระภาษีของยานพาหนะ เมื่อเพิ่มข้อมูลยานพาหนะ ข้อมูลเจ้าของรถ และ การเสียไปแล้ว รูปที่ 5.9 จะเป็นการตรวจสอบข้อมูลทั้งหมด แสดงให้เห็นถึงสถานะปัจจุบันของ ยานพาหนะนั้นๆ

5.3 เว็บสำนักงานกรมตำรวจแห่งชาติ

รูปที่ 5.10 หน้าล็อกอินเว็บสำนักงานตำรวจแห่งชาติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักงานตำรวจแห่งชาติ
Royal Thai Police
http://www.royalthaipolice.go.th

เข้าสู่ระบบ
กรอกข้อมูลความผิด
ตรวจสอบความผิด
พิมพ์อัตรารวบรวม

กรอกกรอกหมายเลขบัตรประจำตัวประชาชนของผู้กระทำผิด และกรอกทะเบียนรถ

1550500014454

2๗ - 1967

เข้าสู่ระบบข้อมูล

รูปที่ 5.11 หน้าตรวจสอบก่อนกรอกข้อมูล

สำนักงานตำรวจแห่งชาติ
Royal Thai Police
http://www.royalthaipolice.go.th

เข้าสู่ระบบ
กรอกข้อมูลความผิด
ตรวจสอบความผิด
พิมพ์อัตรารวบรวม

ERROR! ยานเพราะระบุค่าผิด, ใบขึ้นชื่อกล่าวผิด

กรอกกรอกหมายเลขบัตรประจำตัวประชาชนของผู้กระทำผิด และกรอกทะเบียนรถ

1550500014454

2๗ - 1967

เข้าสู่ระบบข้อมูล

รูปที่ 5.12 หน้าตรวจสอบข้อมูลสถานะไม่ปกติ

จากรูปที่ 5.10 เป็นหน้าล็อกอินเว็บสำนักงานตำรวจแห่งชาติ โดยระบบนี้จะมีผู้ใช้งานอยู่ 3 ระดับด้วยกัน คือ ผู้ใช้งานทั่วไป สามารถใช้ได้เฉพาะตรวจสอบข้อมูลความผิด พนักงานสามารถบันทึกข้อมูลความผิดได้ และก็ตรวจสอบข้อมูลความผิดได้ ส่วน อีกระดับหนึ่ง สิ่งที่สามารถทำได้เพิ่มจากเจ้าหน้าที่ธรรมดา คือ สามารถเพิ่มข้อมูลค้นหาใหม่ได้ ส่วนรูปที่ 5.11 เป็นหน้าเอกสารที่เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบก่อนกรอกข้อมูลความผิด โดยระบบของสำนักงานกรมตำรวจแห่งชาตินี้จะส่งข้อมูลที่กรอกในหน้านี้ 2 อย่าง คือหมายเลขบัตรประชาชน และ หมายเลขทะเบียนรถ ไปตรวจสอบว่าสถานะใบขับขี่ปกติหรือไม่คือไม่ถูกระงับการใช้งาน และดูว่าข้อหาที่ค้างชำระมีกี่ข้อหา และเอาหมายเลขทะเบียนรถไปตรวจสอบว่าสถานะรถถูกอายัดหรือไม่ ถ้าข้อมูลไปปกติคือ ขานพาหนะถูกอายัด ใบขับขี่ถูกระงับการใช้งานจะแสดงดังรูปที่ 5.12

สำนักงานตำรวจแห่งชาติ
Royal Thai Police
http://www.royalthaipolice.go.th

กรอกข้อมูลความผิด

จะเขียน
1550500014454
ใบอนุญาตขับรถชนิดส่วนบุคคล ตรวจสอบ
2ม 1967 ตรวจสอบ

ขับรถฝ่าฝืนสัญญาณไฟจราจรสีแดง
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจรอื่น
 ขับรถฝ่าฝืนเครื่องหมายจราจรจราจร
 ขับรถไม่ปฏิบัติตามป้ายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร
 ขับรถฝ่าฝืนเครื่องหมายจราจร

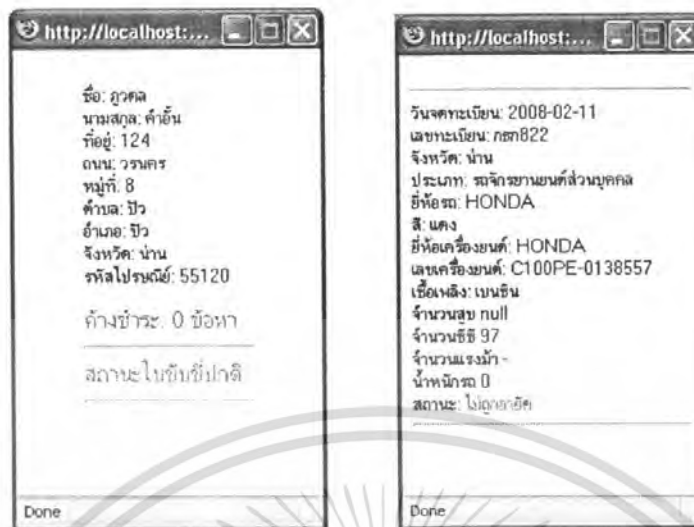
บุคคลิก
ตรงไปตรง
ถนน
ตำบลแขวง
อำเภอเขต
จังหวัด

เงินค่า 52
เดือนบวช-ลาพักะบั้ง
สหภาพบั้ง
สหภาพบั้ง
กรุงเทพมหานคร

บันทึกข้อหา

รูปที่ 5.13 หน้ากรอกข้อมูลความผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.14 Popup คู่มือใส่ข้อมูลใบขับขี่กับข้อมูลยานพาหนะ

เมื่อทำการตรวจสอบข้อมูลถ้าสถานะปกติทั้ง 2 อย่างก็จะเข้าสู่หน้ากรอกข้อมูลความผิดในรูปที่ 5.13 โดยอัตโนมัติ เมื่อเข้ามาสู่หน้านี้แล้วก็ยังสามารถดูรายละเอียดข้อมูลใบขับขี่และข้อมูลยานพาหนะได้ดังรูปที่ 5.14 จากนั้นพอกรอกข้อมูลเสร็จเรียบร้อยก็ทำการบันทึกข้อมูล พอบันทึกเสร็จก็จะมีใบสั่งดังรูปที่ 5.16 แสดงออกมาพร้อม ป้อนให้อัดโนมัติหรือเข้าที่ตรวจสอบข้อมูลความผิดดังรูปที่ 5.15 ก็จะแสดงรายการให้คุณสามารถเข้าไปดูรายละเอียดได้จะแสดงหน้าใบสั่งให้คุณ



รูปที่ 5.15 หน้าตรวจสอบข้อมูลความผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ใบสั่งของเจ้าพนักงานจราจร
 เลขที่ 38
 ค่ายคู่ซิมซี
 รถยนต์นั่งส่วนบุคคลเกิน 7 คน
 หมายเลขทะเบียนรถ
 2ฉ1967 กรุงเทพมหานคร
 กระทำผิดฐาน

1 หยุดรถใบเซลดห้ามหยุด

เหตุเกิดที่

ตรอก/ซอย	จินดา 52	ถนน	อ่อนนุช-ลาดกระบัง
ตำบล/แขวง	ลาดกระบัง	อำเภอ/เขต	ลาดกระบัง
จังหวัด	กรุงเทพมหานคร		
เมื่อ	วันจันทร์ 11 มีนาคม 2008		

อาศัยความมาตรา ๑๕๑ และมาตรา ๑๕๑ แห่ง พ.ร.บ.จราจรทางบก พ.ศ.๒๕๒๒ ซึ่งแก้ไขเพิ่มเติมโดย พ.ร.บ.จราจรทางบก (ฉบับที่ ๕) พ.ศ. ๒๕๓๕ มาตรา ๑๗ และ มาตรา ๑๘ ให้ผู้ขับขี่หรือเจ้าของรถไปชำระค่าปรับภายในวันที่กำหนด

ออกใบสั่งโดย ส.ท. จระเข้ก้อย
 กรณียาชำระเงินภายใน วันจันทร์ 18 มีนาคม 2008

รูปที่ 5.16 ใบสั่งความผิดทางการจราจร

สำนักงานตำรวจแห่งชาติ
Royal Thai Police
<http://www.royalthaipolice.go.th>

เข้าสู่ระบบ

กรอกข้อมูลความผิด

ตราอาชญากรรม

เพิ่มข้อมูลสำนวน

เพิ่มข้อมูลสำนวน

ชื่อข้อมูลสำนวน	ชำระแล้ว/ไม่ชำระ
ราคาค่าปรับ	500
เพิ่มค่าปรับ	20

รูปที่ 5.17 หน้าเพิ่มข้อมูลสำนวน

จากรูปที่ 5.17 เป็นหน้าเพิ่มข้อมูลสำนวน โดยจะต้องกรอกชื่อข้อมูลสำนวน ค่าปรับที่ต้องชำระ กระแสเงินสดที่หัก โดยผู้ที่เพิ่มข้อมูลสำนวนได้จะต้องเป็นผู้ชำระระดับสูงสุด ประโยชน์ด้านการค้าไม่ว่าการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 เว็บธนาคาร



รูปที่ 5.19 หน้าตรวจสอบการชำระค่าปรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.21 หน้าชำระค่าปรับกรณีชำระไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเว็บธนาคารมีหน้าที่ชำระค่าปรับ โดยผู้ที่ใช้งานระบบได้จะต้องล็อกอินระบบ เป็นเจ้าหน้าที่ของธนาคารเท่านั้น ดังรูปที่ 5.18 กระบวนการชำระค่าปรับทำได้โดยผู้ที่ต้องการชำระ ค่าปรับจะนำหมายเลขของใบสั่งให้กับเจ้าหน้าที่ธนาคาร โดยเจ้าหน้าที่ธนาคารจะทำการตรวจสอบ ว่าใบสั่งหมายเลขนี้มีรายละเอียดอย่างไรบ้าง ดังรูปที่ 5.19 ว่าจะสามารถที่จะชำระได้หรือไม่ถ้าชำระ ได้จะสามารถกดปุ่มชำระเงินได้ และมีข้อความบอกว่ายังอยู่ในช่วงชำระเงิน ดังรูปที่ 5.20 ส่วนใน กรณีที่ชำระค่าปรับไม่ได้ คือ เกินกำหนดช่วงที่ให้ชำระ ใบสั่งนี้ชำระไปแล้ว และ ไม่มีข้อมูลของ ใบสั่งนี้ ดังรูปที่ 5.21

จากนั้นพอกกดชำระเงิน ระบบจะทำการส่งว่าชำระเงินแล้วไปให้กับระบบสำนักงาน ตำรวจแห่งชาติโดยที่จะมีตัวตรวจสอบว่าส่งไปถึงหรือยัง ถ้ายังก็จะพยายามส่งจนกว่าจะได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิจารณ์และสรุป

6.1 บทสรุป

ในปัจจุบันเทคโนโลยีเว็บเซอร์วิส (Web Services) เริ่มมีการคิดค้นและพัฒนาระบบขึ้นมาใช้งานเพิ่มมากขึ้น เปรียบเสมือนเป็นเทคโนโลยีด้านการ “บริการ” รูปแบบหนึ่ง ที่สร้างความสะดวกสบายให้กับแอปพลิเคชันต่างๆ และทางด้านระบบฐานข้อมูล โดยเริ่มมีการเปลี่ยนแปลงเข้าสู่ระบบฐานข้อมูลเชิงวัตถุ (Object-Oriented Database) เพราะข้อมูลข่าวสารในปัจจุบันมีความซับซ้อนและหลายรูปแบบกว่าในอดีต

โครงการนี้ได้มีการใช้งานฐานข้อมูลเชิงวัตถุบนเว็บเซอร์วิส ซึ่งมีการนำเอาเทคโนโลยีต่างๆข้างต้นมาประยุกต์ใช้ โดยได้จำลองระบบเพื่ออำนวยความสะดวกในการชำระค่าปรับทางการจราจร โดยมีการติดต่อสื่อสารผ่านทางเว็บเซอร์วิส

6.2 วิจารณ์สิ่งที่ได้จากโครงการ

จากการทำโครงการนี้ทำให้ทราบว่า การที่จะพัฒนาระบบใดๆขึ้นมาสักหนึ่งอย่างการวางแผนงานถือเป็นสิ่งที่มีความสำคัญอย่างมาก เพื่อให้ทำงานได้อย่างมีประสิทธิภาพและทำให้งานเสร็จตรงเวลา ส่วนการศึกษาทฤษฎีที่เกี่ยวข้องหลายๆทฤษฎีก่อนไปประยุกต์ใช้งานจริงก็ถือเป็นความได้เปรียบในการเลือกสิ่งที่เหมาะสมและดีที่สุดสำหรับโครงการได้

6.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

1. ปัญหาในการเลือกใช้งานฐานข้อมูลเชิงวัตถุ แก้ไขโดยสอบถามอาจารย์ที่ปรึกษา
2. ปัญหาการนำฐานข้อมูลเชิงวัตถุ (Cache) มาประยุกต์ใช้งาน โดยมีตัวอย่างการประยุกต์ใช้งานน้อย แก้ไขโดยศึกษาคู่มือการใช้งานของฐานข้อมูล
3. ปัญหาในการวิเคราะห์ออกแบบระบบ โครงการนี้ได้ทำระบบเกี่ยวกับการชำระค่าปรับทางการจราจร จึงไม่ค่อยมีข้อมูลเรื่องกระบวนการทำงานของระบบ แก้ไขโดยไปสอบถามเจ้าหน้าที่ตำรวจ
4. ปัญหาในการเลือกใช้ ทุล (Tool) มาใช้ในพัฒนาเว็บเซอร์วิส เนื่องจากมีให้เลือกใช้หลากหลายจึงจำเป็นต้องเลือก Tool ที่สามารถประยุกต์ใช้งานตามมาตรฐานของเว็บเซอร์วิสที่เราต้องการได้ การแก้ไขคือ ทดลองใช้ Tool ต่างๆ และเลือกอันที่ดีที่สุดมาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม. 2548. **คัมภีร์ การวิเคราะห์และออกแบบระบบ**

เชิงวัตถุด้วย UML. กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์.

อภิเนตร อุนากุล. 2546. **กระบวนการและวิธีการพัฒนาเว็บแอปพลิเคชันโดยใช้ UML.**

กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ศีพจน์ นามวัฒน์. 2549. **การศึกษาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์.** [Online].

Available : <http://www.rtafa.ac.th/article/ORDB.htm>.

สุรี พงศาสกุลชัย. 2550. **การพัฒนาเว็บด้วยสถาปัตยกรรมเชิงบริการบนเทคโนโลยีของ**

Web service. กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์.

Jagadish Chaterjee. 2005. **Introduction to RDBMS, OODBMS and ORDBMS.** [Online].

Available : <http://www.aspfree.com/c/a/Database/Introduction-to-RDBMS-OODBMS-andORDBMS/>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

การติดตั้งและใช้งานฐานข้อมูลเชิงวัตถุ Caché

ขั้นตอนการติดตั้ง

ขั้นตอนการติดตั้งระบบฐานข้อมูล Caché ขั้นตอนแรกทำการดับเบิลคลิกที่ไฟล์ติดตั้ง ชื่อ “CachePCKit” ดังรูปที่ ก.1



รูปที่ ก.1 ไฟล์ติดตั้งของ Caché

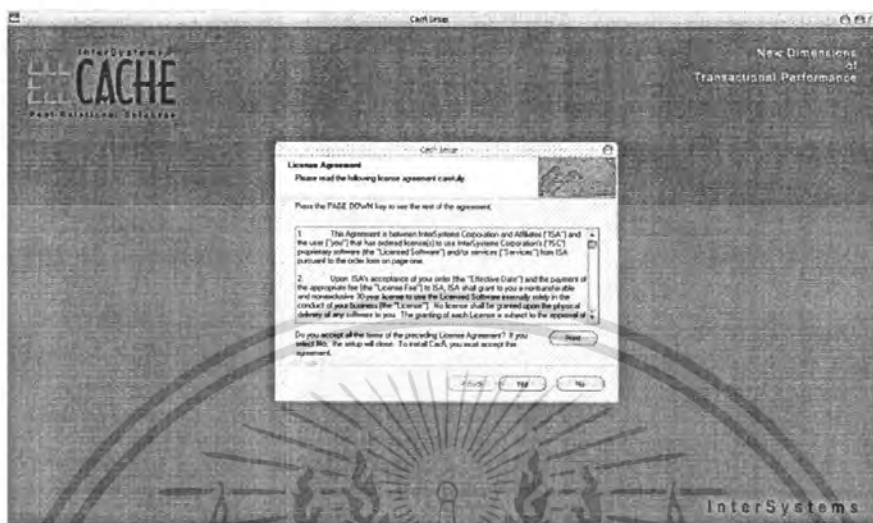
เมื่อดับเบิลคลิกไฟล์ดังกล่าวแล้วจะมีไดอะล็อก สำหรับเลือกภาษาในการติดตั้งขึ้น มาดังรูปที่ ก.2 ให้เลือก English



รูปที่ ก.2 ไดอะล็อกสำหรับเลือกภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเลือกภาษาไปแล้วจะเข้าสู่ขั้นตอนการติดตั้ง โดยจะมีลักษณะเหมือนการติดตั้ง โปรแกรมต่างๆ โดยทั่วไป จะทำตามขั้นตอนดังนี้



รูปที่ ก.3 หน้าจอเริ่มการติดตั้ง

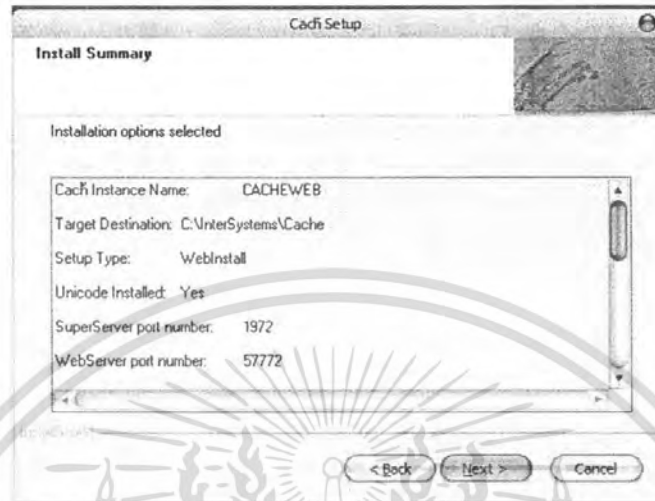
จากรูปที่ ก.3 ให้ทำการเลือก Yes เพื่อที่จะทำการยอมรับข้อตกลงและเข้าสู่ขั้นตอนถัดไป โดยขั้นตอนถัดไปนั้นจะเป็นการเลือกที่อยู่สำหรับติดตั้ง ดังรูปที่ ก.4



รูปที่ ก.4 เลือกที่อยู่สำหรับติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ทำการเลือกที่อยู่สำหรับติดตั้ง โปรแกรมผ่านไปแล้ว จะมีหน้าต่างใหม่ออกมาเพื่อ
บอกรายละเอียดต่างๆที่จำเป็นสำหรับการใช้งาน Caché ดังรูปที่ ก.5



รูปที่ ก.5 รายละเอียดข้อมูลต่างๆ ในการติดตั้ง

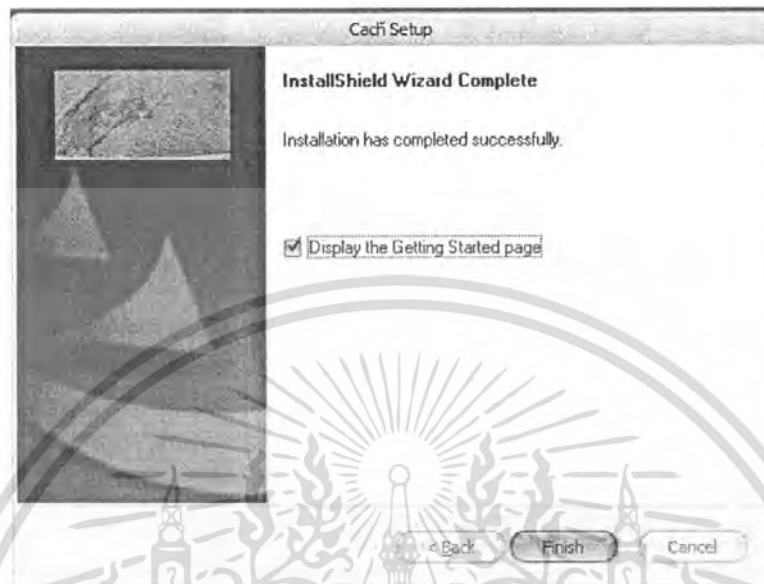
จากนั้นจะเป็นขั้นตอนการติดตั้งไฟล์ต่างๆ โดยขั้นตอนนี้จะใช้เวลาในการทำงาน ประมาณ
1-2 นาที รอจนกว่าระบบจะทำการติดตั้ง Caché เรียบร้อย



รูปที่ ก.6 ระบบกำลังทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเสร็จสิ้นกระบวนการติดตั้งทั้งหมดแล้ว จะมีหน้าต่างขึ้นมาบอกว่าได้ทำการติดตั้งเสร็จเรียบร้อยแล้ว ให้กด Finish



รูปที่ ก.7 เสร็จสิ้นกระบวนการติดตั้ง

หลังจากที่กด Finish ไปแล้ว ตัว Caché จะเริ่มการทำงานดังรูปที่ ก.8 พร้อมกับแสดงเว็บเพจที่แสดงคู่มือการใช้งานเบื้องต้น ดังรูปที่ ก.9



รูปที่ ก.8 Caché กำลังทำงานอยู่

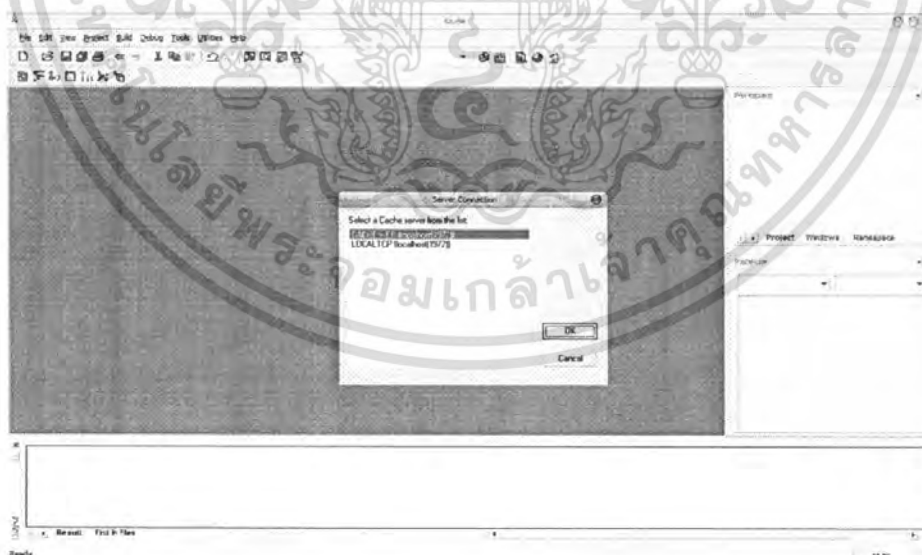
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ๑.9 เว็บเพจคู่มือการใช้งาน Caché

เริ่มต้นการใช้งานโปรแกรม Caché Studio

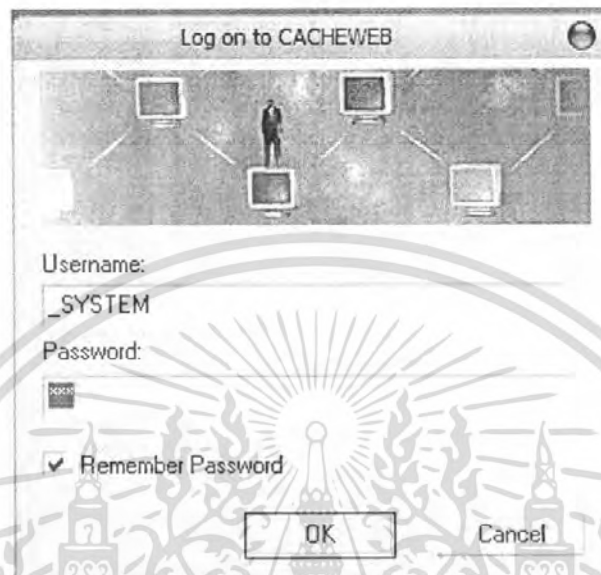
ขั้นตอนแรกทำการเปิดโปรแกรม Caché Studio โดย คลิกที่ Start > All Program > Caché > CACHEWEB > Studio จากนั้นจะเข้าสู่โปรแกรมดังรูปที่ ๑.10 และจะมีไดอะล็อกให้คลิก Server ในการติดต่อ ให้ทำการเลือก CACHEWEB (localhost(1972)) แล้วคลิก OK



รูปที่ ๑.10 หน้าตาโปรแกรม Caché Studio

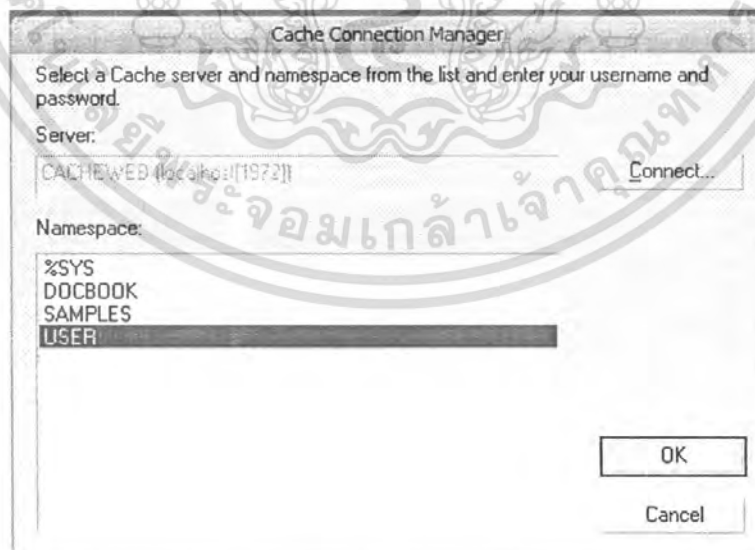
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเลือก ตามขั้นตอนที่ผ่านมาแล้วจะมีไดอะล็อกใหม่ขึ้นมาให้กรอก Username และ Password เพื่อทำการ Log on เข้าสู่ระบบ โดยค่าดีฟอลต์คือ Username = _SYSTEM และ Password = SYS



รูปที่ ก.11 หน้าต่าง Log on เข้าสู่ระบบ

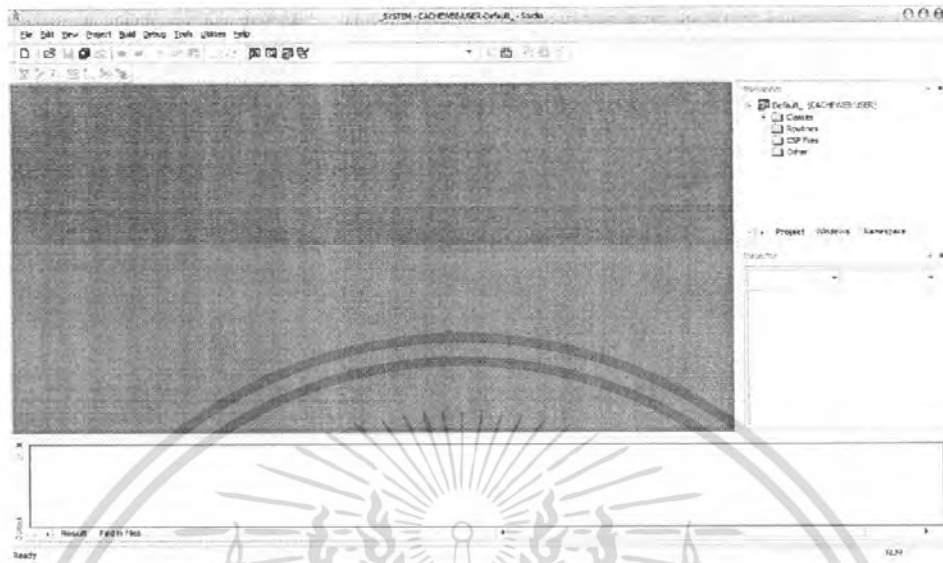
หลังจากที่กด OK ไปแล้ว จะเป็นขั้นตอนการเลือก User ที่ใช้เข้าสู่ Namespace โดยจะเลือก USER ดังรูปที่ ก.12



รูปที่ ก.12 ไดอะล็อกเลือก Namespace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกด OK ไปแล้วจะเข้าสู่ตัวโปรแกรม Caché Studio เพื่อเริ่มต้นการใช้งานดังรูปที่ ก.13



รูปที่ ก.13 หน้าโปรแกรม Caché Studio

การสร้างคลาส

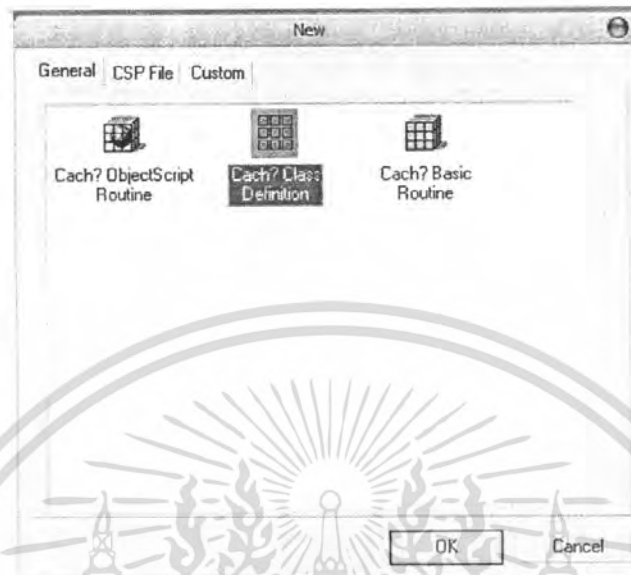
เริ่มต้นการใช้งานโดยการเลือก File > New ดังรูปที่ ก.14



รูปที่ ก -14 การเลือกเมนูเพื่อเริ่มต้นใช้งาน

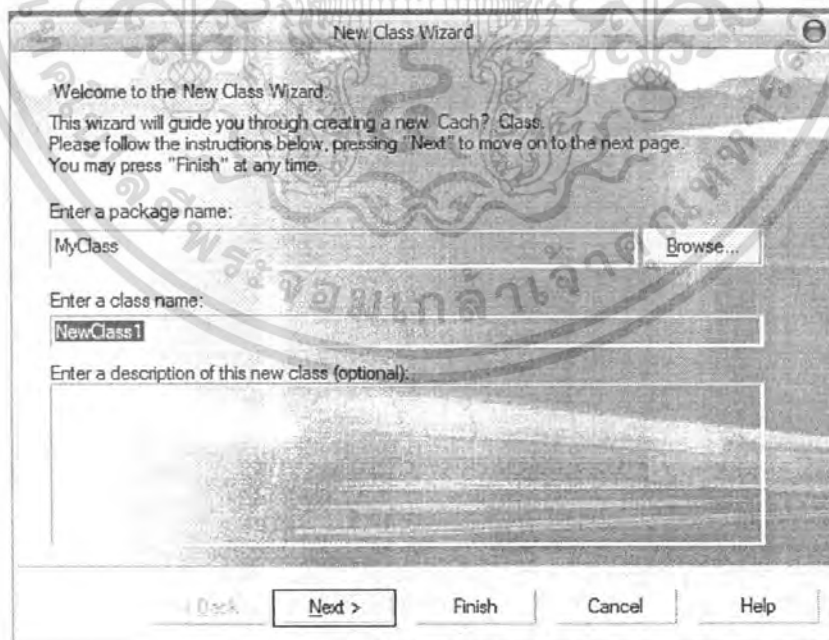
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเลือก New ไปแล้วจะมีไดอะล็อกขึ้นมาให้เลือกชนิดของข้อมูลที่ต้องการจะสร้าง โดยให้ทำการเลือก Cach? Class Definition ดังรูปที่ ก.15



รูปที่ ก.15 ไดอะล็อกเลือกไฟล์ชนิดต่างๆที่ต้องการใช้งาน

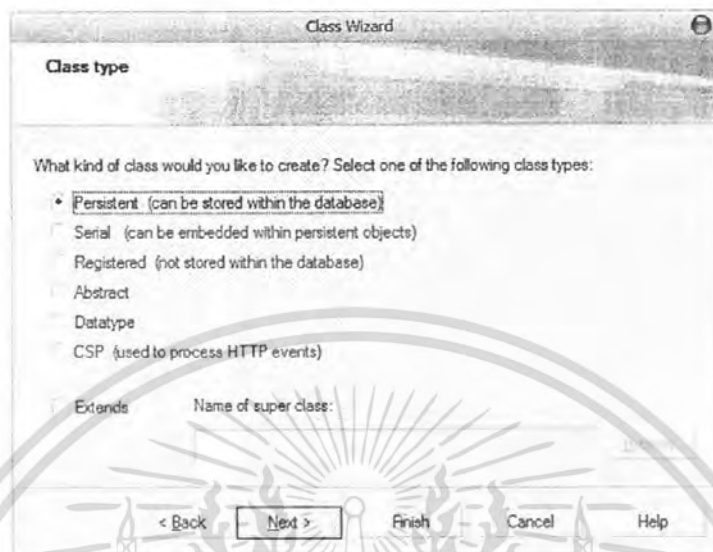
จากนั้นจะมี Wizard ช่วยเหลือในการสร้างคลาสขึ้นมาโดยให้ใส่ชื่อ Package และชื่อ Class ดังรูปที่ ก.16



รูปที่ ก.16 Wizard ในการสร้างคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากใส่ชื่อลงไปแล้วให้เลือก Next เพื่อเข้าสู่ขั้นตอน ในการเลือกชนิดของคลาส ซึ่งให้
เลือกแบบ Persistent เนื่องจากว่าเป็นคลาสของฐานข้อมูล จากนั้นคลิก Finish



รูปที่ ก.17 การเลือกชนิดของคลาส

หลังจากนั้น โปรแกรมจะสร้าง โค้ดออกมาให้ ดังรูปที่ ก.18



รูปที่ ก.18 โค้ดหลังจากที่เสร็จสิ้น Wizard การสร้างคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

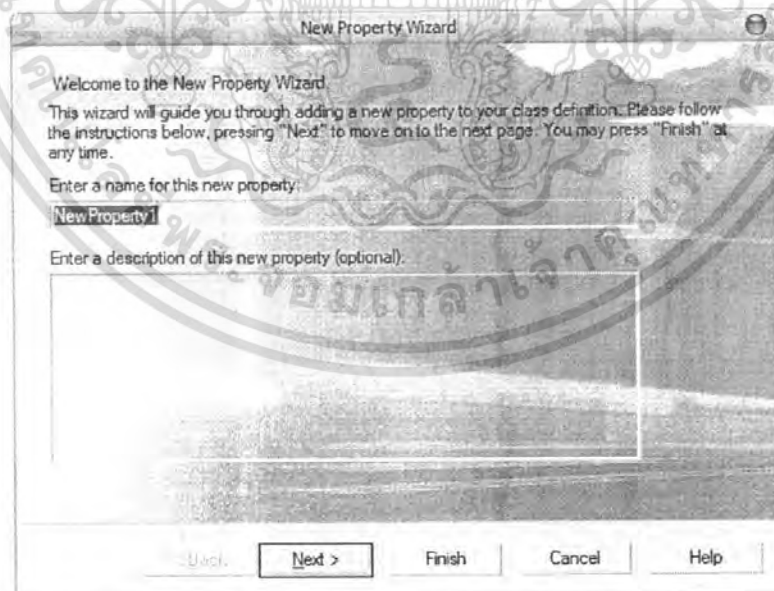
การสร้าง Property

หลังจากที่เราได้ทำการสร้างคลาสไปแล้วนั้น ขั้นตอนนี้จะเป็นการสร้าง Property โดยเลือกที่ Class > Add > New Property ดังรูปที่ ก.19



รูปที่ ก.19 การเลือกเมนูสร้าง Property

เมื่อทำการเลือกตามขั้นตอนที่ผ่านมาแล้วจะมี Wizard ขึ้นมาเหมือนกับการสร้างคลาส โดยจะให้ใส่ชื่อของ Property ลงไปดังรูปที่ ก.20



รูปที่ ก.20 Wizard ในการสร้าง Property

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นคลิก Next เพื่อเข้าสู่ขั้นตอนถัดไป ซึ่งจะเป็นขั้นตอนการกำหนดชนิดของ Property โดยเลือกที่ Browse... ดังรูปที่ ก.21



รูปที่ ก.21 โค้ดเลือกเลือกชนิดของ Property

หลังจากนั้น Wizard จะสร้าง โค้ดให้ดังรูปที่ ก.22



รูปที่ ก.22 โค้ดที่ Wizard สร้างให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างเมธอด

สามารถสร้างเมธอด โดยการเลือก Class > Add > New Method ดังรูปที่ ก.23



รูปที่ ก.24 Wizard ในการสร้างเมธอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

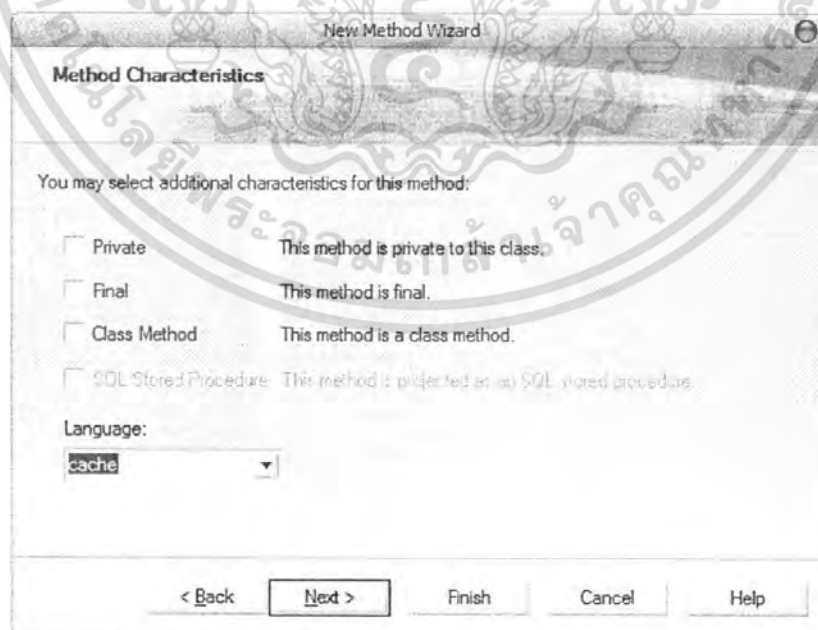
หลังจากที่กรอกชื่อของเมธอดลงไปแล้วให้คลิก Next เพื่อเข้าสู่ขั้นตอนถัดไป ซึ่งจะเป็นการกำหนดชนิดของการคืนค่าของเมธอด สามารถทำได้โดยการคลิก Browse แล้วทำการเลือก ชนิดตามต้องการ และสามารถกำหนดค่าพารามิเตอร์ได้ด้วย ดังรูปที่ ก.25



รูปที่ ก.25 การเลือกชนิดข้อมูลที่คืนค่าออกมา และ กำหนดค่าพารามิเตอร์

สามารถเลือกภาษาที่ใช้ในการเขียนเมธอด ซึ่งมีให้เลือก 3 ภาษาคือ Basic, Cache, Java ดัง

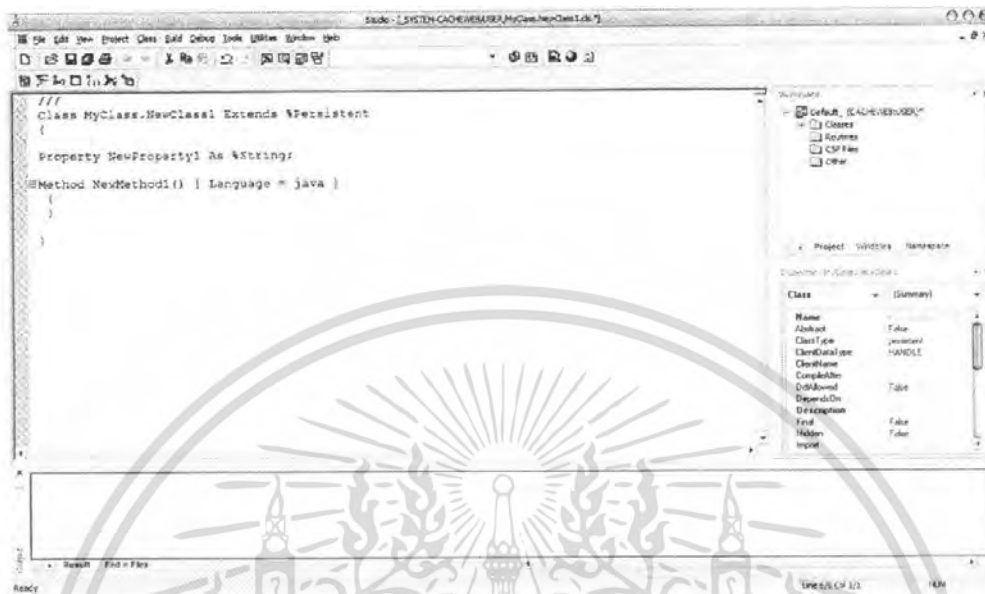
รูปที่ ก.26



รูปที่ ก.26 การเลือกภาษาที่ใช้ในการเขียนเมธอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการกด Finish เพื่อสิ้นสุดการสร้างเมธอด หลังจากนั้นจะได้โค้ดที่ Wizard สร้างมาให้ดังรูปที่ ก.27



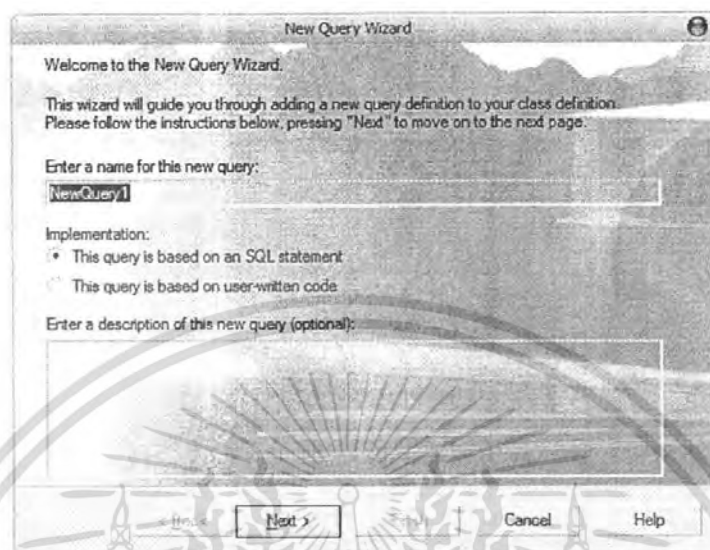
รูปที่ ก.27 โค้ดที่ Wizard สร้างมาให้
การสร้าง Query
สามารถสร้าง Query โดยการเลือก Class > Add > New Query ดังรูปที่ ก.28



รูปที่ ก.28 การเลือกเมนูสร้าง Query

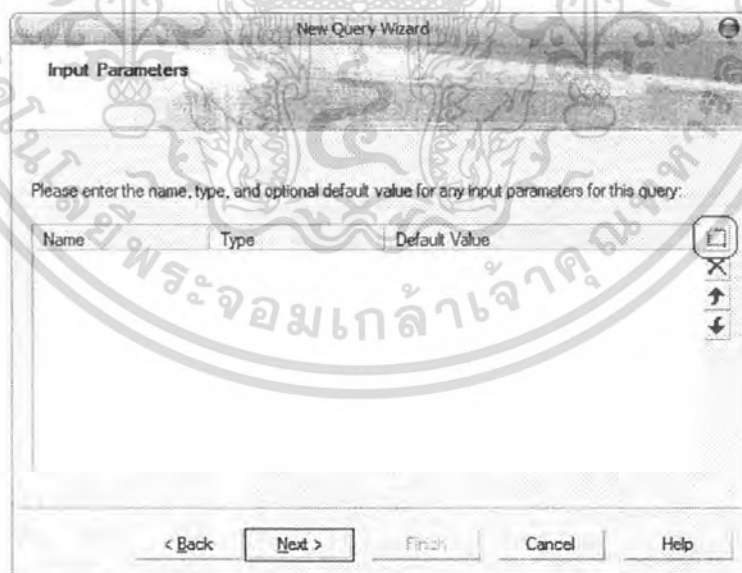
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่เลือกตามขั้นตอนข้างต้นแล้วจะมี Wizard ขึ้นมา ให้กรอกชื่อของ Query จากนั้นคลิก Next ดังรูปที่ ก.29



รูปที่ ก.29 Wizard ในการสร้าง Query

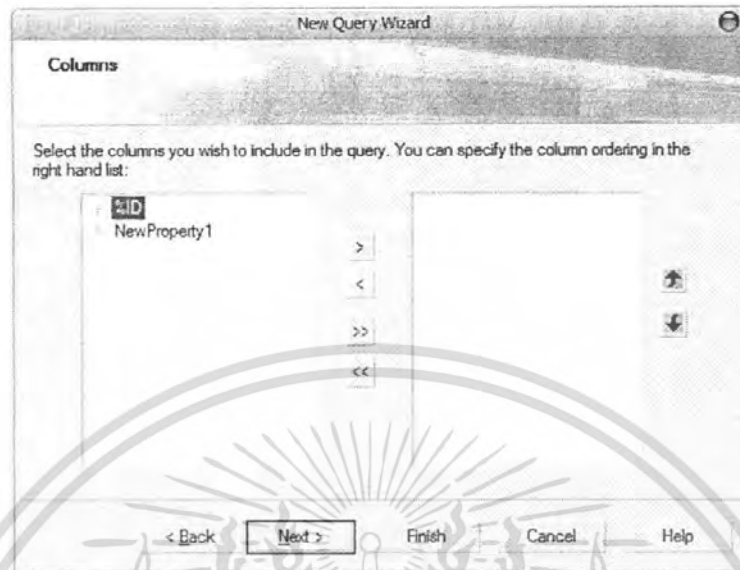
ขั้นตอนถัดไปเป็นการกำหนดพารามิเตอร์และชนิดของข้อมูลที่ต้องการ โดยคลิกที่วงกลมสีแดงดังรูปที่ ก.30



รูปที่ ก.30 ขั้นตอนการกำหนดพารามิเตอร์และชนิดข้อมูล

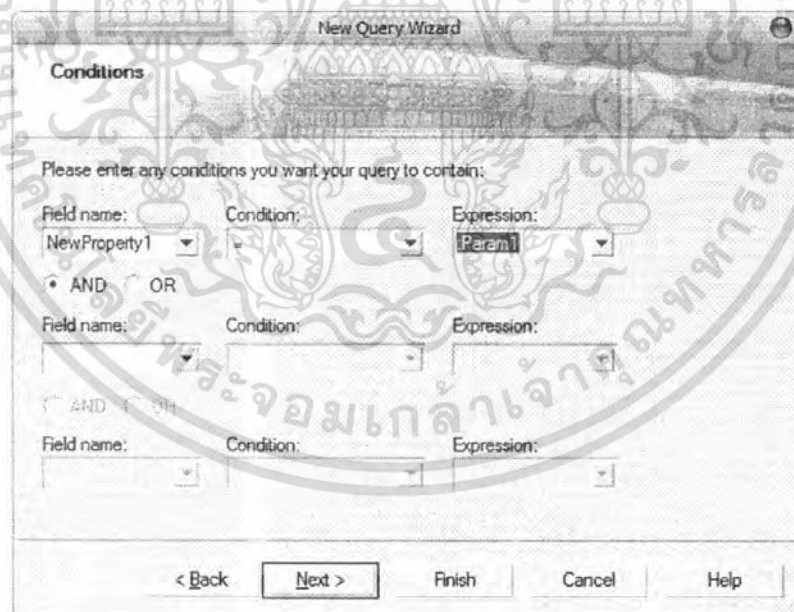
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นเป็นการเลือกข้อมูลที่ต้องการให้แสดงออกมา ดังรูปที่ ก.31



รูปที่ ก.31 การเลือกข้อมูลที่ต้องการให้แสดงออกมา

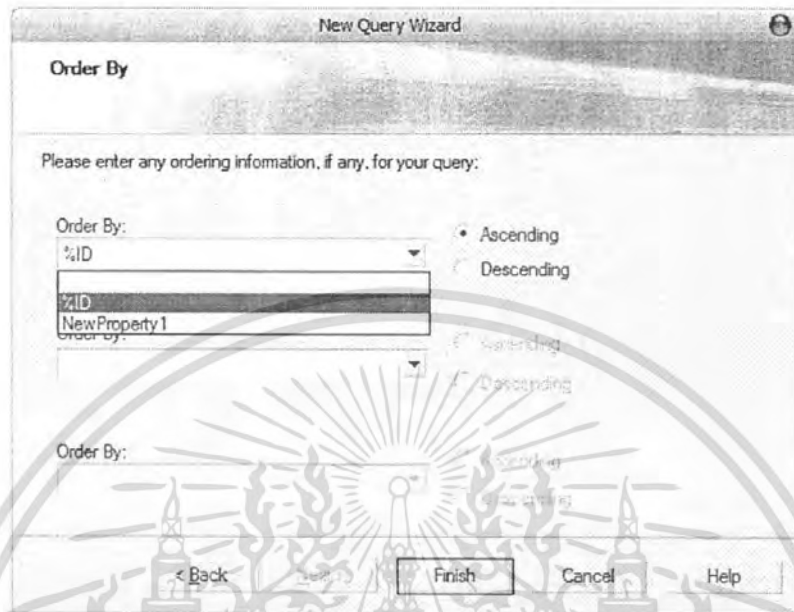
จากนั้นสร้างเงื่อนไขของ Query ดังรูปที่ ก.32



รูปที่ ก.32 การสร้างเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นสามารถที่จะจัดเรียงข้อมูลได้ ว่าจะเรียงตามคอลัมน์ไหน จากน้อยไปมาก หรือ มากไปน้อย ดังรูปที่ ก.33



รูปที่ ก.33 การจัดเรียงข้อมูล

เมื่อกด Finish ตัว Wizard จะสร้างโค้ดออกมาให้ดังรูปที่ ก.34

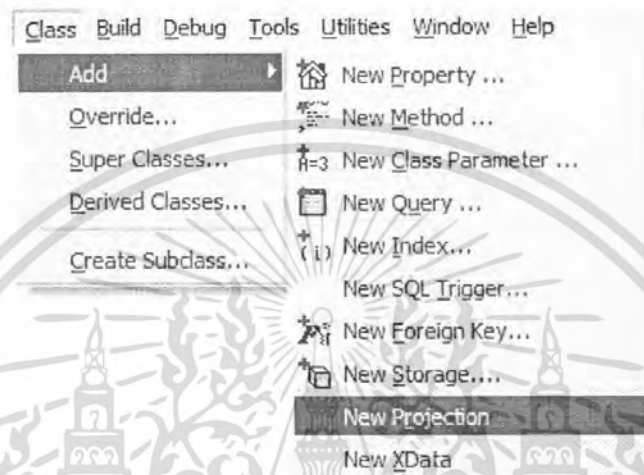


รูปที่ ก.34 โค้ดที่ Wizard สร้างให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

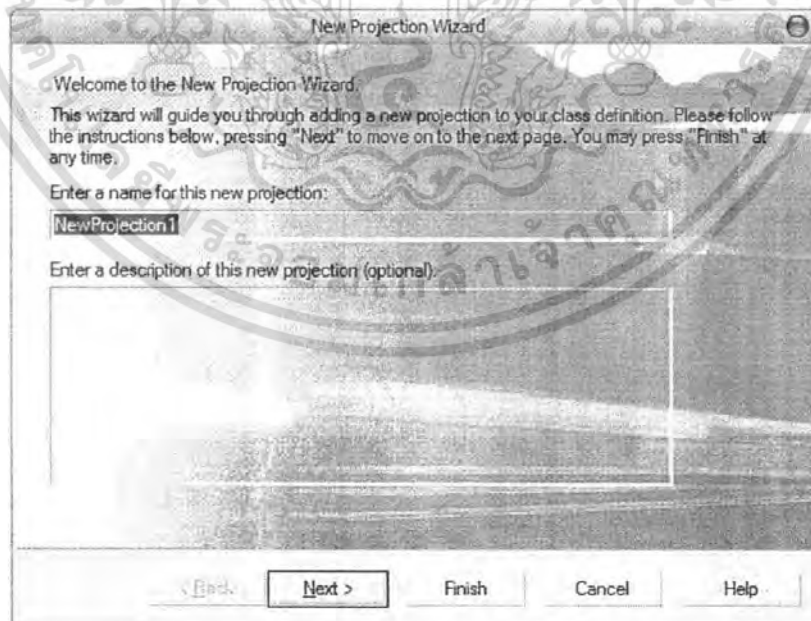
การสร้าง Projection

ตัวอย่างต่อไปนี้จะเป็นการสร้าง Projection เพื่อ Generate code ให้เป็นภาษาจาวา โดยจะแปลงคลาสฐานข้อมูลใน Caché ให้อยู่ในภาษาจาวา ซึ่งสามารถสร้างได้โดยเลือก Class > Add > New Projection ดังรูปที่ ก.35



รูปที่ ก.35 การเลือกเมนูสร้าง Projection

หลังจากที่เลือก New Project แล้ว จะมี Wizard ขึ้นมาดังรูปที่ ก.36



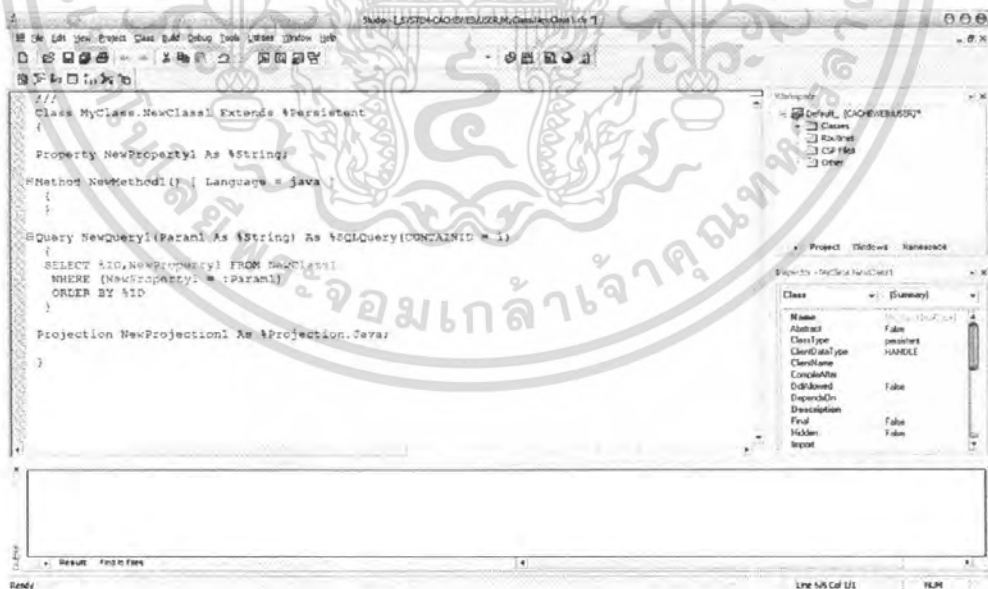
รูปที่ ก.36 แสดง Wizard ในการสร้าง Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นให้คลิก Next จะมีหน้าต่างใหม่ขึ้นมา ดังรูปที่ ก.37 ให้ทำการใส่ที่อยู่ของ Source Code ภาษาจาวาที่เราพัฒนาลงไปช่อง ROOTDIR



รูปที่ ก.37 การกำหนดที่อยู่ของ Source Code ภาษาจาวา
เมื่อคลิก Finish แล้ว ตัว Wizard จะสร้างโค้ดออกมาให้ดังรูปที่ ก.38



รูปที่ ก.38 โค้ดที่ Wizard สร้างมาให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

การติดต่อใช้งานฐานข้อมูล Caché กับ JAVA

การติดต่อฐานข้อมูล Caché กับภาษา Java

หลังจากที่ได้ทำการสร้างคลาสใน Caché Studio เรียบร้อยแล้ว ขั้นตอนต่อไปนี้จะเป็นการนำไปใช้งานร่วมกับ ภาษาจาวา ซึ่งในโครงงานนี้ได้ใช้ NetBeans ในการพัฒนา ซึ่งการใช้งานมีขั้นตอนดังต่อไปนี้

เพิ่มไลบรารีของ Caché ให้กับ NetBeans

หลังจากที่ได้ทำการสร้างคลาสใน Caché Studio และได้สร้างเป็น โค้ดภาษาจาวาเรียบร้อยแล้ว จากนั้นจะต้องนำโค้ดนั้นมาเขียนต่อใน NetBeans โดยต้องนำไลบรารีของ Caché ไปไว้ใน Project ของเราเสียก่อน โดยคลิกขวาที่ Project ที่ต้องการแล้วเลือก Properties ดังรูปที่ ข.1



รูปที่ ข.1 เมนูเลือก Properties ของ Project

เมื่อคลิก Properties ไปแล้วจะมีไอคอนเลือก Properties ขึ้นมาดังรูปที่ ข.2 จากนั้นเลือกเมนู Libraries

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 โค้ดบล็อก Properties ของ Project

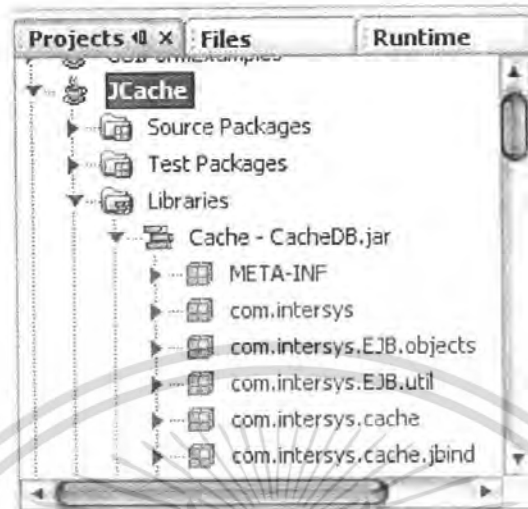
หลังจากนั้นคลิกที่ Add Library เพื่อเพิ่มไลบรารีของ Cache ให้กับ Project ของเรา จากนั้นเลือก Cache และกด Add Library ดังรูปที่ ข.3



รูปที่ ข.3 การเพิ่มไลบรารี Cache ให้กับ Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเสร็จเรียบร้อยแล้วเมื่อดูที่ Project ของเราจะเห็นว่ามีไลบรารีของ Cache อยู่ดังรูปที่ ข.4



รูปที่ ข.4 ไลบรารีของ Cache ที่เราได้เพิ่มไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้