

การพัฒนาโปรแกรมระบบงานเชิงวัตถุโดยใช้ด้วยภาษาจาวา 2
OBJECT ORIENTED SYSTEM DEVELOPMENT USING JAVA 2



โดย

นางสาวสุรธานี
นายสุรพงศ์

เลี้ยวไฟโรจน์ รหัสประจำตัว 40010882
กนกทิพย์สถาพร รหัสประจำตัว 40010908

อาจารย์ที่ปรึกษา

ดร. วรวัฒน์ ลิ่มโกศา

เลขหมู่.....
เลขทะเบียน.....42805
วัน, เดือน, ปี 10 ส.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาโปรแกรมระบบงานเชิงวัตถุโดยใช้ด้วยภาษาจาวา

OBJECT ORIENTED SYSTEM DEVELOPMENT USING JAVA

ผู้จัดทำ

1. นางสาวสุธาสินี

เลี้ยวไพโรจน์

รหัสประจำตัว 40010882

2. นายสุรพงศ์

กนกพิทยัสถาพร

รหัสประจำตัว 40010908



(ดร. วรวัฒน์ ลิ้มโกศา)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมระบบงานเชิงวัตถุโดยใช้ด้วยภาษาจาวา 2

นางสาวสุธาสินี เลี้ยวไพโรจน์ 40010882

นายสุรพงศ์ กนกทิพย์สถาพร 40010908

ดร. วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษา

ปีการศึกษา 2543

บทคัดย่อ

ภาษาจาวา (Java) เป็นภาษาหนึ่งที่มีลักษณะการโปรแกรมเป็นแบบเชิงวัตถุ (Object Oriented Programing, OOP) ซึ่งมีจุดเด่นมากที่สุดที่เราทำการเขียนโปรแกรมขึ้นมาเพียงครั้งเดียว แต่สามารถนำโปรแกรมนั้นไปใช้งานได้ในทุกระบบปฏิบัติการดังคำกล่าวที่ว่า “ Write once run anywhere ” และ จาวายังประกอบด้วยส่วนประกอบที่ใช้งานในระดับคอมพิวเตอร์สมัยใหม่ เช่น ส่วนของกราฟฟิกและระบบการติดต่อผู้ใช้ด้วยกราฟฟิกส์ (GUI) ส่วนประกอบของระบบเครือข่ายและ ระบบติดต่อกับฐานข้อมูลด้วยเหตุนี้เองในปัจจุบันจึงมีคณนิยมใช้ภาษาจาวามากขึ้นเรื่อยๆทั่วโลก

ในปัจจุบันภาษาจาวาได้มีการใช้งานเอ็นเตอร์ไพสจาวาบีน (Enterprise JavaBeans: EJB) ซึ่งมีสถาปัตยกรรมแบบคอมโพเนนต์ที่สามารถติดต่อดาต้าเบส (Database) ซ้ำมเซิร์ฟเวอร์ (Server) ได้ดีเหมาะสำหรับนำมาใช้กับงานมิดเดิลแวร์ (Middleware) ได้ดีมาก นอกจากนี้ยังมีมาตรฐานที่ใช้สนับสนุนการใช้งานด้านมัลติเทียร์ โปรแกรมมิง โมเดล (Multitiered programming model) ซึ่งก็คือเทคโนโลยีของ Java 2 Platform, Enterprise Edition (J2EE) ซึ่งได้ใช้ประโยชน์จาก Java 2 Platform, Standard Edition ซึ่งทำให้มีความเสถียร (stable) และความมั่นคงปลอดภัยมากยิ่งขึ้น

การใช้งานเจทูอีบีนั้นสามารถใช้ได้บนหลายแพลตฟอร์ม (platform) ทำให้แอปพลิเคชันระดับเอ็นเตอร์ไพรส์ (Enterprise-class Application) สามารถทำงาน ณ ที่ใดก็ได้ และทำให้โปรแกรมมิง โมเดล (Programming model) สามารถศึกษาเข้าใจได้ง่ายขึ้น นอกจากนี้เจทูอีบียังสนับสนุนเทคโนโลยีระดับเอ็นเตอร์ไพรส์ซึ่งมีอีเจบีประกอบอยู่ด้วย

Object Oriented System Development Using Java 2

Miss.Sutasinee Lieopiroj 40010882

Mr. Surapong Kanoktipsataporn 40010908

Dr. Worawat Limpoka Advisor

ABSTRACT

Java language is an Object Oriented Programming language. It's main advantage are portability and hardware independency as the saying " Write once run anywhere" . Java has many components that use in the new computer programming such as Graphic User Interface (GUI) component, Network component and Database component. Many programmers in the world use Java.

Today ,Java has the Enterprise JavaBeans (EJB) which is a component architecture that allows for connect to the database across the server so that it good to use in middleware. Beside this it also has the Java 2 Platform, Enterprise Edition (J2EE) defines the standard for developing multitier enterprise applications that takes advantage of many features of the Java 2 Platform, Standard Edition such as stability and security .

J2EE provides a platform that will allow enterprise-class application the ability to run anywhere and a single easy-to-learn blueprint programming model for J2EE. J2EE is a collection of enterprise technologies, of which EJB is an integral part. By understanding and using J2EE properly, you can build portable, object-oriented, enterprise-class applications in Java.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ วรวัฒน์ ลิ้มโกศา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

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

สุธาสินี
สุรพงศ์

เลี้ยวไพโรจน์
กนกทิพย์สถาพร



สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII

บทที่ 1 บทนำ

1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	3

บทที่ 2 ภาษาจาวา (Java)

2.1 การเขียนโปรแกรมแบบอ็อบเจกต์ (Object Oriented Programming)	4
2.2 คอมไพเลอร์	5
2.3 ภาษาจาวา	6
2.4 จาวาสคริปต์	6
2.5 จาวาแอปเพล็ต	7
2.6 จาวาเซิร์ฟเล็ต	8
2.7 จาวาบีน	9
2.8 เอนเตอร์ไพส จาวาบีน	9
2.9 จาวา แอปพลิเคชัน	9
2.10 ไคลเอนต์ – เซิร์ฟเวอร์	10

บทที่ 3 สถาปัตยกรรมบนฝั่งเซิร์ฟเวอร์

3.1 วิวัฒนาการของ Server Side Application	15
3.2 Muti-tier Architecture	16
3.3 Java 2 Platform, Enterprise Edition (J2EE)	20
3.4 Enterprise JavaBeans (EJB)	23
3.5 Remote Method Invocation (RMI) and RMI-IIOP	24
3.6 Java Naming and Directory Interface (JNDI)	25
3.7 Java Database Connectivity (JDBC)	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค	การเปรียบเทียบเทคโนโลยีระหว่าง Com Plus กับ EJB	101
ภาคผนวก ง	Getting Started	103
บรรณานุกรม		



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่

หน้าที่

ตารางที่ 2 -1 แสดงลักษณะของภาษาจาวา

6

ตารางที่ 2-2 ข้อเปรียบเทียบของการประมวลผลที่ไคลเอนต์และเซิร์ฟเวอร์

12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้าที่
รูปที่ 2.1 COM Object ซึ่งมี Interface 3 Interface	5
รูปที่ 2-2 การแปลคำสั่งของของโปรแกรมที่เป็นแบบ Java Applet	7
รูปที่ 2-3 Servlet Engine and its Servlets	8
รูปที่ 2-4 การแปลคำสั่งของของโปรแกรมที่เป็นแบบ Java Application	10
รูปที่ 2-5 การประมวลผลที่ไคลเอนท์	10
รูปที่ 2-6 การประมวลผลที่เซิร์ฟเวอร์	11
รูปที่ 2-7 Applet, servlet และ Enterprise JavaBeans	13
รูปที่ 3-1 เบสิค client/server architecture เริ่มแรกสุดที่ใช้กันบนเว็บ	15
รูปที่ 3-2 การรวมส่วน Business Logic Layer เข้ากับ Presentation Layer	17
รูปที่ 3-3 การรวมส่วน Business Logic Layer เข้ากับ Data Layer	18
รูปที่ 3-4 แสดง 3-Tier Web-Based Deployment	19
รูปที่ 3-5 แสดงโครงสร้างสถาปัตยกรรมของ J2EE ในระบบ 3-Tier Server	21
รูปที่ 3-6 แสดงโครงสร้างของระบบ 3-Tier Client/Server Application	22
รูปที่ 3-7 แสดง RMI distributed application	25
รูปที่ 3-8 แสดงโมดูลการเชื่อมต่อ JDBC อย่างง่าย	27
รูปที่ 4-1 EJB Container pooling of entity bean	33
รูปที่ 4-2 State save และ State load ในการทำ Passivation และ Activation ของ entity bean	34
รูปที่ 5-1 แสดงการแบ่งความรับผิดชอบในการสร้างระบบที่รองรับการใช้งาน ของ Enterprise Javabeans	37
รูปที่ 5-2 entity bean จะใช้ data access logic layer ใน multi-tier architectures	39
รูปที่ 5-3 แสดงการใช้งานร่วมกันของ session bean และ entity bean ใน Pricing Application	40
รูปที่ 6-1 แสดงการทำงานของ JSP	45
รูปที่ 6-2 แสดงการติดต่อกับฐานข้อมูลของ JSP	46
รูปที่ 6-3 การทำงานของ JSP ร่วมกับ Enterprise JavaBeans	46
รูปที่ 6-4 แสดงการร้องขอบริการจากไคลเอนต์ไปยังเซิร์ฟเล็ต (Servlet)	47
รูปที่ 7-1 แสดงการเชื่อมต่อแบบ Thin-client	48
รูปที่ 7-2 The Flat Transaction	50
รูปที่ 7-3 The Nested Transactions	52
รูปที่ 8-1 แสดงหลักการของ Application Server	58
รูปที่ 9-1 ยูสเคสไดอะแกรมโรงแรม	61
รูปที่ 9-2 คาด้าโฟลไดอะแกรมโรงแรมลิเวล 1	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 9-3 ค้าปลีกโพลีโอะแกรมโรงแรมลิเวิล 2 Search	63
รูปที่ 9-4 ค้าปลีกโพลีโอะแกรมโรงแรมลิเวิล2 Manage After Reservation	64
รูปที่ 9-5 อีโอะแกรมโรงแรม	65
รูปที่ 9-6 คลาสโอะแกรมโรงแรม	67
รูปที่ 9-7 ยูสเคสโอะแกรมภัตตาคาร	68
รูปที่ 9-8 ค้าปลีกโพลีโอะแกรมภัตตาคารลิเวิล 1	69
รูปที่ 9-9 ค้าปลีกโพลีโอะแกรมภัตตาคารลิเวิล 2 Manage After Reservation	70
รูปที่ 9-10 อีโอะแกรมภัตตาคาร	71
รูปที่ 9-11 คลาสโอะแกรมภัตตาคาร 1	72
รูปที่ 9-12 คลาสโอะแกรมภัตตาคาร 2	73
รูปที่ 9-13 ยูสเคสโอะแกรมโอล่าล่าทัวร์	74
รูปที่ 9-14 ค้าปลีกโพลีโอะแกรมโอล่าล่าทัวร์ลิเวิล 0	75
รูปที่ 9-15 ค้าปลีกโพลีโอะแกรมโอล่าล่าทัวร์ลิเวิล 1	78
รูปที่ 9-16 อีโอะแกรมโอล่าล่าทัวร์	81
รูปที่ 9-17 คลาสโอะแกรมโอล่าล่าทัวร์ Log System	82
รูปที่ 9-17 คลาสโอะแกรมโอล่าล่าทัวร์ OlalaAgency System	84
รูปที่ 9-17 คลาสโอะแกรมโอล่าล่าทัวร์ Package System	85
รูปที่ 9-17 คลาสโอะแกรมโอล่าล่าทัวร์ User System	86
รูปที่ 1 Olalatour	87
รูปที่ 2 AirlineSearch	88
รูปที่ 3 AirlineSearchList	89
รูปที่ 4 BusSearch	90
รูปที่ 5 BusSearchList	91
รูปที่ 6 HotelSearch	92
รูปที่ 7 HotelSearchList	92
รูปที่ 8 PackageSearch	93
รูปที่ 9 PackageSearchList	94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันพัฒนาการทางด้านสถาปัตยกรรมคอมพิวเตอร์บนฝั่งเซิร์ฟเวอร์ที่สามารถทำให้การสร้างแอปพลิเคชัน(application)ในระดับเอนเตอร์ไพสทำได้ง่ายขึ้นนอกจากนี้ยังทำให้แอปพลิเคชันมีความน่าเชื่อถือ (Reliable) มีระบบรักษาความปลอดภัย (Secure) มีความถูกต้องมากขึ้นโดยไม่ต้องเขียนโครงสร้างระบบที่มีความซับซ้อนทั้งหมดเอง ทำให้มีการพัฒนาผลิตภัณฑ์จากสถาปัตยกรรมบนฝั่งเซิร์ฟเวอร์ (Server-Side Component Architectuer Solutions) ออกโดยค่ายต่างๆดังนี้ เช่น

- Microsoft's Distributed internet Application Architecture (DNA)
- Sun Microsystem's Java 2 Platform,Enterprise Edition (J2EE)
- The Object Management Group's COBRA Standard

สำหรับในโครงการที่จัดทำขึ้นนี้จะศึกษาและอิมพลีเมนต์ (implement) เฉพาะผลิตภัณฑ์ของค่าย SUN ซึ่งก็คือ Sun Microsystem's Java 2 Platform,Enterprise Edition หรือ เจทูอีอี โดยเป็นผลิตภัณฑ์ที่เกิดจากการได้ตระหนักถึงความจำเป็นและความสามารถของสถาปัตยกรรมแบบคอมพิวเตอร์บนฝั่งเซิร์ฟเวอร์ (Server-Side Component Architecture) ซึ่งจะต้องมีการรองรับในการทำงานแบบ N เทียร์ที่มีจาวาเป็นฐานและมีลักษณะของการ โปรแกรมเชิงวัตถุ (Object Oriented Programming) ซึ่งนับว่าได้ก้าวหน้ามาก และนับวันแนวโน้มของการใช้งานจะมีเพิ่มมากขึ้นเรื่อย ๆ และจริง ๆ แล้วโดยตัวภาษาจาวาเองเหมาะสำหรับการใช้งานบนเซิร์ฟเวอร์จาวา ส่วนการใช้งานที่ฝั่งของไคลเอ็นต์จะมีปัญหาอยู่หลายเรื่อง เช่น แพลตฟอร์ม (platform) ของยูสเซอร์อินเทอร์เฟซ (User Interface) เข้ากันไม่ได้ หรือยูสเซอร์อินเทอร์เฟซซ้ำ รวมไปถึงเรื่องของ Java Virtual Machine (JVM) ที่รันบนไคลเอ็นท์แมชชีน (Client Machine) ผิดเวอร์ชันกัน แต่สิ่งเหล่านี้จะไม่เป็นปัญหากับฝั่งเซิร์ฟเวอร์เพราะการดีพลอยบนฝั่งเซิร์ฟเวอร์จะอยู่ภายใต้การควบคุมของเอ็นไวรอนเมนต์ (Environment)

ภาษาจาวาจะเป็นภาษาที่สะดวกในการเขียนคอมพิวเตอร์ไสไว้บนฝั่งเซิร์ฟเวอร์เพราะว่าผลิตภัณฑ์ของฝั่งเซิร์ฟเวอร์จะมียูนิกซ์ (UNIX) และเมนเฟรม (Mainframe) เป็นแกนหลักซึ่งหมายความว่าภาษาที่สามารถทำงานข้ามแพลตฟอร์มได้จะมีความสำคัญมากเพราะผู้พัฒนาจะต้องเขียนคอมพิวเตอร์เพียงครั้งเดียวแต่สามารถนำมาใช้งานได้บนหลายเอ็นไวรอนเมนต์

ในการตอบสนองของจาวาบนฝั่งเซิร์ฟเวอร์นั้นบริษัท ซัน-ไมโครซิสเต็มส์ (Sun Microsystem) จึงได้ออกแพลตฟอร์มที่ชื่อว่า “Java 2 Platform, Enterprise Edition (J2EE)” โดย เจทูอีอี เป็นแพลตฟอร์มที่มีความอิสระ, สามารถนำไปใช้ได้อย่างมีประสิทธิภาพ, สนับสนุนกับผู้ใช้ได้ทีละหลาย ๆ คน (Multi-user), มีระบบรักษาความปลอดภัย (security) โดยแกนหลักในเทคโนโลยีของ เจทูอีอี นี้คือ Enterprise Javabeans (EJB) ซึ่งเป็นมาตรฐานในการสร้างคอมพิวเตอร์บนฝั่งเซิร์ฟเวอร์โดยการใช้ภาษาจาวา

เทคโนโลยีของ เจทูอีอี (J2EE Technologies) ที่ได้ทำการศึกษาในโครงการนั้นประกอบไปด้วยส่วนต่างๆดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เอนเตอร์ไพร์สจาวันส์ (Enterprise JavaBeans : EJB)
- ข้อกำหนดทางด้านการสื่อสารระหว่างเครื่องคอมพิวเตอร์ในรูปแบบของ Java Remote Method Invocation (RMI)
- การเชื่อมต่อกับฐานข้อมูลด้วยภาษาจาวาหรือ Java Database Connectivity (JDBC)
- มาตรฐานที่ใช้ในการติดต่อกับออบเจกต์ที่ทำงานร่วมกันภายใต้ระบบเครือข่ายหรือ Java Naming and Directories Interface (JNDI)
- การจัดการกับทรานส์แอ็กชันหรือ Java Transaction APIs (JTA), Java Transaction Services (JTS)

โดยในส่วนของแอปพลิเคชัน (application) ในโครงการนั้นจะใช้ภาษาจาวา (Java) ในการอิมพลีเม้นท์ เนื่องจากภาษาจาวาเป็นหนึ่งในภาษาที่เป็นลักษณะเชิงวัตถุหรือออบเจกโอเรียนเต็ด โปรแกรมมิ่ง (OOP) ซึ่งคือ มุมมองในการออกแบบซอฟต์แวร์ (software) ที่มองเอนติตี้ (entity) ต่างๆในระบบเป็นออบเจกต์ซึ่งมีคุณสมบัติและความสามารถเฉพาะตัว ที่สามารถทำหน้าที่ของแต่ละออบเจกต์ นั้นได้ เพื่อที่จะทำงานร่วมกันให้บรรลุจุดประสงค์ของระบบที่ตั้งไว้

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อให้สามารถวิเคราะห์ ออกแบบ และพัฒนาระบบงาน ตามแนวคิดของออบเจกต์
- 1.2.2 ศึกษาการพัฒนาโดยใช้สถาปัตยกรรมแบบเซิร์ฟเวอร์ไซด์คอมโพเนนท์ (Server Side Component)
- 1.2.3 ศึกษาการสร้างแอปพลิเคชันด้วยการใช้สถาปัตยกรรมแบบเซิร์ฟเวอร์ไซด์คอมโพเนนท์ว่า จะต้องมีการออกแบบในส่วนใดบ้างและจะสามารถนำไปอิมพลีเม้นต์ได้อย่างไร ซึ่งอาจจะนำความรู้ที่ได้รับนี้ไปประยุกต์ใช้ในงานต่างๆ ได้
- 1.2.4 เพื่อเป็นการนำเสนอแนวคิดและผลการปฏิบัติงานในการสร้างระบบด้วยการใช้สถาปัตยกรรมแบบเซิร์ฟเวอร์ไซด์คอมโพเนนท์ซึ่งมีเอนเตอร์ไพร์สจาวันส์เป็นซอฟต์แวร์ คอมโพเนนต์ (software – component) บนฝั่งเซิร์ฟเวอร์ที่สามารถนำไปใช้งานในดิสทริบิวท์มัลติเทียร์เอ็นไวรอนเมนต์ (distributed multi-tier environment) ได้

1.3 ขอบเขตของโครงการ

งานวิจัยนี้จะสร้างระบบการค้าเชิงอิเล็กทรอนิกส์ (Electronic Commerce) ที่เกี่ยวกับการท่องเที่ยวขึ้นมา โดยการใช้เว็บลอจิกเซิร์ฟเวอร์ (WebLogic Server) และออราเคิลดาต้าเบส (Oracle Database) ซึ่งมีการออกแบบส่วนติดต่อกับผู้ใช้ด้วยเบ้าแวร์ซึ่งเบื้องหลังนั้นจะมีการทำงานของเอนเตอร์ไพร์สจาวันส์ต่างๆ ที่บรรจุอยู่ในอีเจบีคอนเทนเนอร์ (EJB Container) นอกจากนี้ยังมีส่วนการทำงานของจาวาเซิร์ฟเล็ต ซึ่งเป็นโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ที่เป็นเซิร์ฟเวอร์เป็นหลักโดยจาวาเซิร์ฟเล็ตนั้นเป็นโปรแกรมที่สร้างจากภาษาจาวา และใช้ประมวลผลหรือทำงานบนเครื่องเซิร์ฟเวอร์ โดยจะทำงานควบคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับเครื่องของผู้ใช้ และเมื่อผู้ใช้ต้องการประมวลผล ก็สั่งให้เซิร์ฟเล็ตที่อยู่บนเซิร์ฟเวอร์ทำงานได้จากเครื่องคอมพิวเตอร์ของผู้ใช้ แล้วส่งผลลัพธ์กลับมายังเครื่องผู้ใช้อีกทีซึ่งผลลัพธ์จากการทำงานนั้นจะอยู่ในรูปแบบของข้อมูลภาษา HTML โดยในงานวิจัยนี้จะต้องออกแบบสถาปัตยกรรมของระบบทั้งหมด

นอกจากนั้นในโครงงานนี้ยังถือว่าเป็นโครงการที่ทดลองสร้างแอปพลิเคชันเพื่อศึกษาความเป็นไปได้ในการใช้งานของระบบ ดังนั้นจึงต้องมีการจัดการด้านต่างๆ ของระบบ เช่น การจัดการด้าน ซิควิรตี้ (Security), รีไลบิลิตี้ (Reliability), เฟลโอเวอร์ (Fail-over), คลัสเตอร์ริง (Clustering) และโหลดบาลานซ์ (Load balancing) เพื่อให้มีการทำงานที่มีประสิทธิภาพและสามารถรองรับผู้ใช้ปริมาณมาก ๆ ได้

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลักๆ คือ ภาษาจาวา, สถาปัตยกรรมบนฝั่งเซิร์ฟเวอร์, เอนเตอร์ไพซ์จาวาบี, เอนเตอร์ไพซ์จาวาบีคอนเทนเนอร์และเว็บคอนเทนเนอร์ซึ่งมีรายละเอียดดังในบทที่ 2, 3, 4 และ 5 จากนั้นในบทที่ 6 จะเป็นการศึกษาเรื่องการติดต่อจากไคลเอนต์ และบทที่ 7 จะกล่าวถึงการจัดการด้านฐานข้อมูล เมื่อศึกษาถึงระบบทั้ง 3 เทียร์แล้ว จึงทำการศึกษาด้านความปลอดภัยในการใช้งานระบบจากไคลเอนต์เพื่อให้ระบบมีเสถียรภาพและมีความน่าเชื่อถือ รายละเอียดดังกล่าวจะถูกนำเสนอในบทที่ 8

จากความรู้ที่ได้ศึกษาทั้งหมดมาออกแบบสถาปัตยกรรมของระบบจากนั้นจึงนำมาสร้างเป็นแอปพลิเคชัน ซึ่งมีรายละเอียดในบทที่ 9 และ บทที่ 10 ซึ่งเป็นบทสุดท้ายจากจะทำการวิเคราะห์และสรุปผลการทำงาน ผลที่ได้รับจากงานวิจัยชิ้นนี้ และแนวทางในการพัฒนางานวิจัยนี้เพิ่มเติม และแนวทางในการนำไปประยุกต์ใช้

บทที่ 2

ภาษาจาวา

2.1 การเขียนโปรแกรมแบบอ็อบเจกต์ (Object Oriented Programming)

การเขียนโปรแกรมแบบอ็อบเจกต์ (OOP) คือ มุมมองในการออกแบบซอฟต์แวร์ที่มองเ็นคิตีต่างๆในระบบเป็นอ็อบเจกต์ซึ่งมีคุณสมบัติและความสามารถเฉพาะตัว ที่สามารถทำหน้าที่ของแต่ละอ็อบเจกต์นั้นได้ เพื่อที่จะทำงานร่วมกันให้บรรลุจุดประสงค์ของระบบที่ตั้งไว้ ส่วนอ็อบเจกต์นั้นคือการรวบรวมข้อมูลและฟังก์ชันที่ดำเนินการต่อข้อมูลนั้นเข้าไว้ด้วยกันเป็นหน่วยเดียวกัน โดยในการทำงานร่วมกันของอ็อบเจกต์แต่ละอ็อบเจกต์นั้นจะสื่อสารกันโดยส่งข้อมูลข่าวสารจากอ็อบเจกต์หนึ่งไปยังอีกอ็อบเจกต์หนึ่งได้ โดยใช้การส่งข้อความ (message)

การเขียนโปรแกรมแบบโอโอพีต่างจากการเขียนโปรแกรมแบบโครงสร้าง (Structure) พอสมควร เนื่องจากแบบ โอโอพี ผู้เขียนต้องมองโปรแกรมออกเป็นชิ้นส่วนย่อยๆที่มีที่เก็บข้อมูล มีการทำงานภายในชิ้นส่วนย่อยๆให้ได้ แล้วนำเอาชิ้นส่วนย่อยๆเหล่านั้นมาผสมปนเปกันในอัตราที่เหมาะสม จนเกิดเป็นโปรแกรมขึ้นใช้งาน แต่ถ้าเป็นแบบโครงสร้าง แล้ว ผู้เขียนต้องมองโปรแกรมออกเป็นลักษณะการทำงานแบบที่การทำงานต้องมีรูปแบบที่แน่นอน เช่น แบบเรียงลำดับ แบบมีเงื่อนไข หรือแบบซ้ำไปมาแต่เราสามารถนำเอาความคิดแบบ โครงสร้างไปใช้ร่วมกับแบบโอโอพีได้ในการทำชิ้นส่วนย่อยๆ (Object) ทำงานแบบมีโครงสร้างมากขึ้น

2.1.1 คุณสมบัติหลักของ Object-Oriented

1. เอ็นแคปซูลชัน (Encapsulation) คือ คุณสมบัติในการห่อหุ้มข้อมูลที่อยู่ภายในอ็อบเจกต์ด้วยเมทอดที่เตรียมไว้เพื่อให้จัดการกับข้อมูลนั้น เพื่อป้องกันไม่ให้ภายนอกสามารถเข้าถึงข้อมูลโดยตรง ทำให้มีความปลอดภัยและง่ายต่อการเมนเทนแนนซ์ (maintenance)

2. อินเฮริเทนซ์ (Inheritance) คือ คุณสมบัติในการสืบทอดคุณสมบัติและความสามารถของ อ็อบเจกต์หนึ่งไปยังอีกอ็อบเจกต์หนึ่ง เพื่อแทนที่จะประกาศคลาสหลายๆคลาสที่มีคุณสมบัติและความสามารถบางส่วนเหมือนกัน เราสามารถประกาศพารินท์คลาส (Parent class) แล้วประกาศคลาสอื่นๆมาสืบทอดคุณสมบัติและความสามารถของคลาสนี้ได้ ทำให้ซอฟต์แวร์มีความรียูสเอเบิล (Reusable)

3. โพลิมอร์ฟิซึม (Polymorphism) คือ คุณสมบัติในการที่มีเมทอดหลายๆเมทอดที่ชื่อเดียวกันแต่มีความสามารถต่างกัน ซึ่งการเลือกจะใช้เมทอดไหนก็ขึ้นอยู่กับพารามิเตอร์ที่รับเข้ามา คุณสมบัตินี้ทำให้ซอฟต์แวร์มีความยืดหยุ่น (Flexible)

2.1.2 คลาส (class)

คลาสคือ ต้นแบบ (Template) ของอ็อบเจกต์ โดยจะประกอบไปด้วยองค์ประกอบพื้นฐานที่สำคัญสองส่วนคือ ดาต้าเมมเบอร์ และเมมเบอร์ ฟังก์ชัน

ดาต้า เมมเบอร์ (Data member) หรือ คุณลักษณะ (Attribute) หมายถึง ดาต้าของอ็อบเจกต์นั้นๆ เป็นลักษณะเฉพาะตัวที่แสดงให้เห็นถึงความแตกต่างกันของวัตถุแต่ละชนิด เช่น สี,ขนาด,จำนวน ส่วน

เมมเบอร์ ฟังก์ชัน (Member function) หรือเมธอดจะหมายถึงโพสเซซซิ่งโค้ด (processing code) ของออบเจกต์ เป็นส่วนที่จะบอกว่า คลาสนั้นทำอะไรได้บ้าง สามารถทำงานร่วมกับคลาสอื่นๆ ได้อย่างไร และคลาสอื่นๆ สามารถส่งงานคลาสนั้นได้อย่างไร

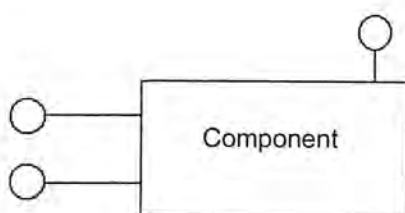
2.2 คอมโพเนนต์ (Component)

การพัฒนาระบบงานในปัจจุบันมีหลายแนวทาง เช่น การพัฒนาระบบแบบโครงสร้าง การพัฒนาระบบแบบออบเจกต์โอเรียนเท็ด (Object Oriented) แต่ละแนวทางต่างเสนอวิธีการต่างๆ เพื่อให้การพัฒนาระบบเป็นขั้นตอน สามารถจัดการระบบที่มีขนาดใหญ่และมีความซับซ้อน เพื่อให้ได้ระบบงานที่มีคุณภาพตรงกับความต้องการของผู้ใช้งาน

แต่อย่างไรก็ตามการพัฒนาระบบด้วยวิธีดังกล่าวก็ยังมีปัญหาต่างๆ มากมาย ในกรณีของการพัฒนาระบบแบบโครงสร้างจะประสบปัญหาด้านโมเดลของระบบที่ไม่เชื่อมต่อกันตลอดวัฏจักรของการพัฒนา ในส่วนของการพัฒนาระบบแบบออบเจกต์โอเรียนเท็ดมีปัญหาที่เห็นได้ชัดคือ การที่ออบเจกต์ผูกติดกันมากเกินไป (Tightly Coupling) หากมีการแก้ไขออบเจกต์ใดอาจทำให้เกิดผลกระทบต่อออบเจกต์ถูกหลานได้ หรืออาจเกิดปัญหาด้านเวอร์ชันของซอฟต์แวร์ กล่าวคือ การเปลี่ยนแปลงไลบรารีเพื่อเพิ่มเติมฟังก์ชันใหม่จะทำให้ ซอฟต์แวร์เดิมอาจมีปัญหาในการทำงาน เป็นต้น

คอมโพเนนต์ออบเจกต์โมเดล (Component Object Model) เป็นมาตรฐานในการทำงานร่วมกันระหว่างออบเจกต์ที่ไม่ขึ้นกับภาษาที่ใช้พัฒนา โดยมีการติดต่อกันผ่านอินเตอร์เฟสเท่านั้น อีกทั้งยังเพิ่มความสามารถในด้านการเรียกใช้แบบไม่ขึ้นกับรูปแบบ (Location Transparency) ทำให้ผู้พัฒนาสามารถเรียกใช้ออบเจกต์ได้ตั้งแต่ในรูปแบบพื้นฐาน คือ อยู่เครื่องเดียวกัน จนถึงในรูปแบบการเรียกใช้แบบข้ามเครื่อง จึงสามารถทำระบบแบบกระจาย (Distributed System) ซึ่งเป็นประโยชน์หลายประการ เช่น การทำโหนดบาลาสซึ่งเพื่อเป็นการลดภาระของเครื่องเซิร์ฟเวอร์ และประเด็นที่สำคัญในออบเจกต์ คือการนำกลับมาใช้ใหม่ (Reusability) ซึ่งจะช่วยให้ปัญหาในการพัฒนาซอฟต์แวร์ต่างๆ ที่กล่าวมาแล้วได้

คอมโพเนนต์เป็นมาตรฐานในการทำงานร่วมกันระหว่างคอมโพเนนต์ ซึ่งมาตรฐานนี้ไม่ขึ้นกับตัวแอปพลิเคชันและภาษาที่ใช้พัฒนา โดยแอปพลิเคชันจะติดต่อกับแอปพลิเคชันอื่นๆ และกับระบบ โดยผ่านกลุ่มของฟังก์ชันหรืออินเตอร์เฟส ซึ่งอินเตอร์เฟส ก็คือ ข้อตกลงระหว่างซอฟต์แวร์คอมโพเนนต์กับผู้ใช้ งาน ดังรูปที่ 2.1



รูปที่ 2.1 Component ซึ่งมี Interface 3 Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ภาษาจาวา

จาวาเป็นภาษาที่ใช้คอนเซ็ปของออบเจ็กต์โอเรียนเท็ด ในการเขียน หลายคนที่เกี่ยวข้องกับการเขียนโปรแกรมคงจะทราบดีว่าออบเจ็กต์โอเรียนเท็ด สามารถลดความซับซ้อนโครงสร้างของโปรแกรม รวมไปถึงอำนวยความสะดวกในการ rius (reuse) ส่วนประกอบต่าง ๆ ของโปรแกรมที่เขียนไว้แล้วเพียงใดนอกจากนี้จาวายังไม่ขึ้นกับแพลตฟอร์มซึ่งจะช่วยให้เราสามารถทำการพัฒนาระบบโดยใช้ เอนไวรอนเมนต์อะไรก็ได้ซึ่งโดยทั่วไปมักจะเป็นวินโดวส์เอ็นไวรอนเมนต์ (Widows environment)

ภาษาจาวาถูกพัฒนามาจากบริษัทซัน (Sun Microsystems) ซึ่งจัดให้เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่ใช้หลักการออกแบบตัวภาษาด้วยวิธีเชิงวัตถุ และตัวภาษาถูกใช้เป็นเครื่องมือสำหรับพัฒนาโปรแกรมด้วยแนวคิดเชิงวัตถุ โดยตัวภาษามีลักษณะพิเศษดังนี้

Portability	สามารถในการใช้งานในสภาวะแวดล้อมที่แตกต่าง โดยไม่ต้องมีการปรับแต่ง
Simple	ความง่ายในการเขียนโปรแกรม
Robust	ความคงสภาพในการทำงาน มีโอกาสเกิดข้อผิดพลาดที่ไม่พึงประสงค์ได้น้อย
Secure	การรองรับมาตรฐานความปลอดภัยในการใช้งานรูปแบบต่างๆ
Distributed	มีความสามารถในการประมวลผลแบบกระจาย
Object-Oriented	มีหลักการของแนวคิดเชิงวัตถุ ในการสร้างโปรแกรม

ตารางที่ 2-1 แสดงลักษณะของภาษาจาวา

ลักษณะภาษาจาวา

การสร้าง : ต้องเขียนคำสั่งตามรูปแบบของภาษาจาวาแล้วเก็บไว้ในไฟล์ที่เรียกว่าซอร์สโค้ด จากนั้นนำไปคอมไพล์ไป เป็นไฟล์คลาส

การใช้งาน : ถ้าภาษาจาวาที่ใช้สร้างไฟล์คลาสเขียนในรูปแบบจาวาแอปพลิเคชันให้ใช้งานบนเครื่องได้ทันทีผ่าน JVM เช่น ภาษาจาวาที่ใช้สร้างไฟล์คลาสเขียนในแบบจาวาแอปเล็ต ก็ต้องสร้างเว็บมาเรียกใช้งานจาวาแอปเล็ตอีกที (2 ขั้นตอน) และเวลาแอปเล็ตทำงานเจีเอ็มที่อยู่บนบราวเซอร์ก็จะรันแอปเล็ตให้ทำงาน เป็นต้น

2.4 จาวาสคริปต์ (JavaScripts)

จาวาสคริปต์เป็นภาษาสคริปต์ที่เกิดขึ้นมาจากความนิยมในอินเทอร์เน็ต และเป็นหนึ่งในภาษาสำหรับอินเทอร์เน็ตที่มีการยอมรับ และได้รับความนิยมอย่างสูง ด้วยลักษณะของภาษาสคริปต์ที่ง่ายในการศึกษา เรียนรู้ และทำความเข้าใจ แต่ให้ศักยภาพและความสามารถเทียบเท่ากับภาษาระดับสูง

เว็บเพจที่เพิ่มความสามารถด้วยจาวาสคริปต์ สามารถสร้างความแตกต่างจากเว็บเพจปกติทั่วไป โดยในจาวาสคริปต์นั้นจะประกอบไปด้วยออบเจ็กต์ต่างๆที่สร้างไว้ในเอกสารเอชทีเอ็มแอล เช่น วินโดว์ เฟรม ฟอรัม ปุ่ม ภาพ ลิสต์บ็อกซ์ ปลั๊กอิน และ แอปเพล็ตที่ฝังตัวอยู่ ฯลฯ ซึ่งเราสามารถควบคุมออบเจ็กต์เหล่านี้ได้ดีกว่าการเขียนด้วยจาวาแอปเพล็ตเพราะเราสามารถโต้ตอบกับออบเจ็กต์บนเพจได้โดยตรง

การเขียนจาวาสคริปต์นั้นมีข้อดีที่สามารถใช้บนระบบปฏิบัติการใดๆก็ได้และยังสามารถกระโดดไปมาระหว่างระบบปฏิบัติการเหล่านี้ได้อีกด้วยซึ่งจะเป็นประโยชน์ในกรณีที่เราต้องการที่จะให้มีการทำงานของสคริปต์ภายใต้เบราว์เซอร์ในระบบปฏิบัติการต่าง ๆ โดยจาวาสคริปต์ทำงานได้ทั้งฝั่งไคลเอนต์ (สคริปต์จะทำงานหลังจากถูกโหลดลงหน่วยความจำในเครื่องของผู้ใช้แต่ละคน) และฝั่งเซิร์ฟเวอร์ (สคริปต์จะทำงานบนเว็บเซิร์ฟเวอร์ภายใต้สภาพแวดล้อม LiveWire ของ Netscape)

ถ้าจะเปรียบเทียบการใช้งานระหว่างจาวาสคริปต์กับภาษาจาวาแล้วอาจจะพิจารณาได้จาก เช่น ถ้าต้องการความง่ายและรวดเร็วก็ใช้จาวาสคริปต์ แต่ถ้าต้องการสร้างโปรแกรมที่ไม่ซับซ้อน และมีประสิทธิภาพก็จะใช้ภาษาจาวาซึ่งแน่นอนกว่า แต่ถึงอย่างไรมันก็มีหลายปัจจัยอื่นประกอบ เช่น เวลาในการใช้งาน ต้องการเร็วหรือช้า

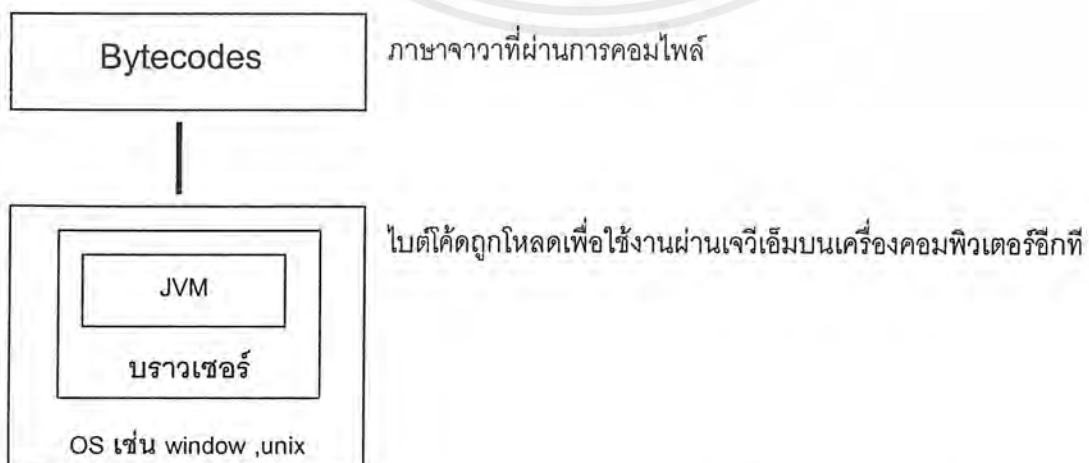
ลักษณะของจาวาสคริปต์

การสร้าง : ต้องเขียนร่วมกับภาษา เอชทีเอ็มแอล เพื่อใช้สร้างเว็บ

การใช้งาน : ต้องถูกโหลดใช้บนเบราว์เซอร์ และเบราว์เซอร์จะทำหน้าที่แปลคำสั่งจาวาสคริปต์ที่ละคำสั่งเพื่อทำงานตามคำสั่ง

2.5 จาวาแอปเพล็ต (java Applet)

จาวาแอปเพล็ตคือ โปรแกรมคอมพิวเตอร์ที่เขียนด้วยภาษาจาวา ที่ถูกแปลงอยู่ในรูปของ ไบต์โค้ด ด้วยวิธีการคอมไพล์ และการทำงานของโปรแกรมจะทำงานผ่าน จาวาเวอร์ชวลแมชีน (JVM) ที่อยู่บนเบราว์เซอร์ เช่น Netscape, Internet Explorer, Opera ที่สนับสนุนการใช้งานจาวาแอปเพล็ต



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2-2 การแปลคำสั่งของโปรแกรมที่เป็นแบบจาวาแอปเพล็ต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 จาวาบีนส์ (Javabeans)

จาวาบีนส์เป็นรูปแบบการใช้งานออปเจกต์ที่ถูกสร้างไว้แล้ว และนำมาประกอบกันเป็นซอฟต์แวร์ ถ้าเทียบไปแล้วจะคล้ายกับแอคทีฟคอนโทรล (Active Control) หรือ แอ็กทีฟเอ็กซ์ ที่ใช้ในวีบี (VB) หรือ วีซี (VC) แต่ จาวาบีนส์จะสร้างด้วยภาษาจาวา เพื่อให้ใช้งานได้ทุกแพลตฟอร์มทำให้บีนมีการสร้างเพื่อขายทางการจากผู้ผลิต เช่น Third Party ด้วย นอกจากนี้จาวาบีนส์ยังมีลักษณะอื่นๆ อีก คือ

- เป็นคลาสในภาษาจาวาเพื่อเรียกใช้ได้ในโปรแกรมพัฒนาจำพวกวิซวล เช่น บินเกี่ยวกับปุ่ม บินเกี่ยวกับรูปภาพ บินเกี่ยวกับเมนู เป็นต้น ถ้าเปรียบเทียบกับ visual basic ก็คือสิ่งที่เรียกว่าคอนโทรลนั่นเอง
- เป็นส่วนประกอบทางกราฟฟิคที่สามารถนำมาประกอบกันเป็นโปรแกรม ที่ติดต่อกับผู้ใช้แบบกราฟิกยูสเซอร์อินเตอร์เฟสได้ง่ายขึ้นสะดวกขึ้น เหมือนที่ใช้ใน visual basic, visual c++ หรือใน delphi เป็นต้น
- เป็นลักษณะคล้าย ๆ กล่องที่เราสามารถเปลี่ยนลักษณะการทำงานและคุณสมบัติต่าง ๆ ข้างในได้โดยเซ็ทที่ตัวพร็อเพอร์ตี้ (property) ของมัน (คล้าย ๆ พวก ActiveX ของ microsoft)

2.8 Enterprise JavaBeans (EJB)

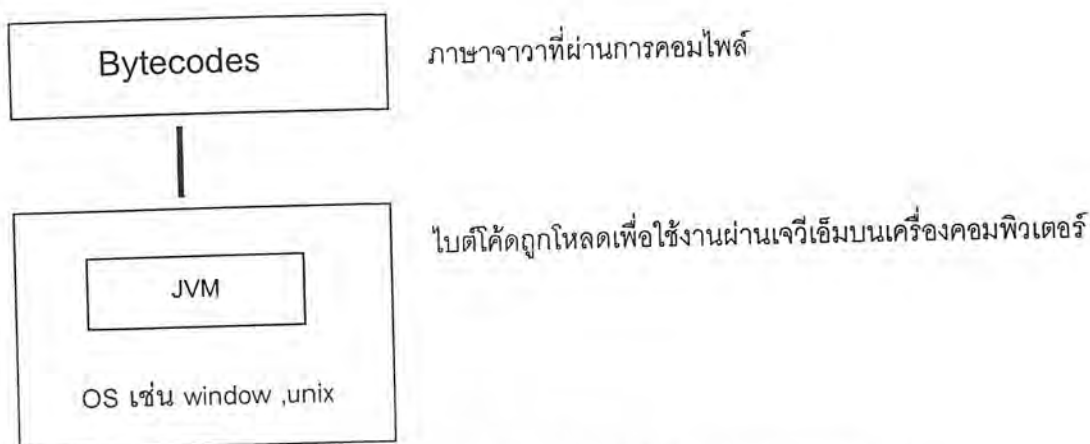
เป็นสถาปัตยกรรมคอมโพเนนต์บนฝั่งเซิร์ฟเวอร์ที่ทำให้การสร้างแอปพลิเคชันในระดับเอนเตอร์ไพสซ์ทำได้ง่ายขึ้นและมีความน่าเชื่อถือ (Reliable) เนื่องจากมีระบบรักษาความปลอดภัย (Secure) โดยไม่ต้องเขียนโครงสร้างระบบที่มีความซับซ้อนทั้งหมดเอง และอีเจบีจะถูกออกแบบให้สนับสนุนการนำแอปพลิเคชันเดิมมาใช้ใหม่ได้ (Reusability) อีเจบีเป็นสถาปัตยกรรมคอมโพเนนต์ที่สามารถดีพลอยได้โดยเราจะเรียกว่า "เอนเตอร์ไพสซ์บีน" (Enterprise bean)

อีเจบีเป็นแอปพลิเคชันคอมโพเนนต์ในระดับ coarser-grained สามารถดีพลอยเพียงลำพังหรือ ดีพลอยโดยรวมกับคอมโพเนนต์อื่นๆ (ประกอบกันขึ้นเป็นระบบแอปพลิเคชันที่มีขนาดใหญ่ขึ้น) โดยในการดีพลอยต้องมีคอนเทนเนอร์ให้บริการ โดยเอนเตอร์ไพสซ์บีนนั้นมีคอมโพเนนต์ของจาวาอยู่ 2 ชนิด ที่คล้ายกันมากคือ

1. แอปเพล็ต
2. เซิร์ฟเล็ต

2.9 จาวาแอปพลิเคชัน (Java Application)

จาวาแอปพลิเคชันคือโปรแกรมคอมพิวเตอร์ที่เขียนด้วยภาษาจาวา ที่ถูกแปลอยู่ในรูปของ ไบต์โค้ดด้วยวิธีการคอมไพล์ และการทำงานของโปรแกรมจะทำงานผ่าน จาวาเวอร์ชวลแมชีน (JVM) ที่ทำหน้าที่แปลงคำสั่งไบต์โค้ดให้อยู่ในรูปของคำสั่งที่สามารถประมวลผลได้ บนเครื่องคอมพิวเตอร์ ที่ติดตั้งโอเอสใด ๆ



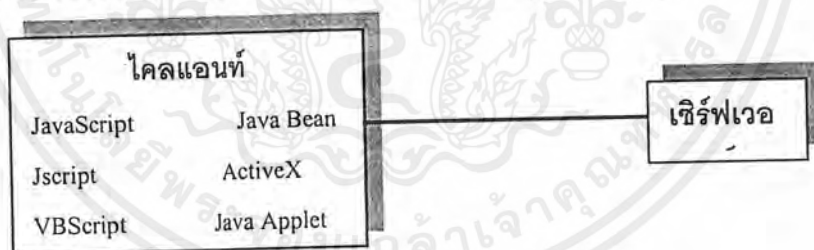
รูปที่ 2-4 การแปลคำสั่งของของโปรแกรมที่เป็นแบบจาวาแอปพลิเคชัน

การทำงานของจาวาแอปพลิเคชัน จึงเหมือนกับการทำงานของโปรแกรมทั่วไป ที่ถูกเขียนจากภาษาอื่นๆ เช่น ซี, วีบี, เคลไฟท์ เป็นต้น ต่างกัน เพียงแต่จาวาแอปพลิเคชันต้องทำงานผ่านเจี๊ยมและด้วยเหตุผลนี้เอง ทำให้จาวาคงเป็นลักษณะของ Write Once, Run Anywhere คือเขียนครั้งเดียวใช้ได้ทุกที่

2.10 Client-Server

การทำงานหรือการประมวลผลของเว็บ ถูกแยกออกเป็นสองแบบ โดยพิจารณาที่การประมวลผลเป็นหลัก คือ

2.10.1 การประมวลผลที่ไคลเอนท์ (Client-Side Processing)



รูปที่ 2-5 การประมวลผลที่ไคลเอนท์

รูปแบบการทำงานลักษณะนี้ ไคลเอนท์ซึ่งก็คือ โปรแกรมบราวเซอร์ที่จำเป็นต้องทำงานหนัก เนื่องจากต้องแปลงคำสั่งต่างๆที่บรรจุอยู่บนเว็บให้สามารถทำงานได้อย่างถูกต้อง ดังนั้นเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมไคลเอนท์ ต้องใช้ทรัพยากรเพียงพอกับการทำงาน เทคนิคเหล่านี้ เป็นเทคโนโลยีที่ทำงานบนฝั่งไคลเอนท์

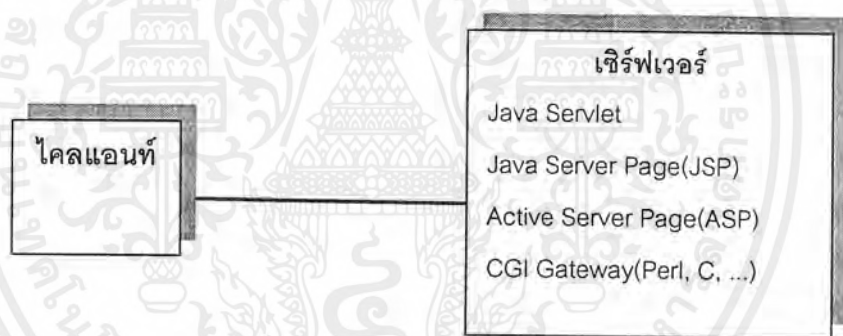
- JavaScript สคริปต์ภาษาจาวา ทำงานบนบราวเซอร์ทั่วไป โดยบราวเซอร์จะแปลงคำสั่งสคริปต์ก่อน แล้วจึงจะทำงานได้ตามคำสั่ง
- JScript สคริปต์ภาษาจาวาของไมโครซอฟต์ทำงานได้เฉพาะอินเทอร์เน็ตเอ็กพลอเรอร์ (Internet Explorer) ของไมโครซอฟต์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **VBScript** สคริปภาษาเบสิกของไมโครซอฟต์ทำงานได้เฉพาะอินเทอร์เน็ตเอกซ์พลอเรอร์ของไมโครซอฟต์เองเท่านั้นเช่นกัน
- **Java Applet** ไปได้ก็เขียนด้วยภาษาจาวาและถูกคอมไพล์แล้วทำงานบนบราวเซอร์คือ คำสั่งในไปได้ก็ถูกแปลง แล้วจึงทำงาน
- **Java Bean** เป็นภาษาจาวาเป็นคอมโพเนนต์ที่ทำงานเหมือนแอปเพล็ต แต่ถูกสร้างขึ้นมาจากลักษณะของจาวาเขียนด้วยจาวาจึงทำงานได้ทุกแพลตฟอร์ม
- **ActiveX** ก็คือคอมโพเนนต์คล้ายกับจาวาเป็นแต่ต่างกันที่ภาษาที่ใช้สร้าง ส่วนใหญ่สร้างจากเครื่องมือที่ใช้งานบนวินโดวส์ ดังนั้น ActiveX จึงทำงานได้บนอินเทอร์เน็ตเอกซ์พลอเรอร์

2.10.2 การประมวลผลที่เซิร์ฟเวอร์(Server-Side Processing)

รูปแบบการทำงานลักษณะนี้ เซิร์ฟเวอร์คือเว็บเซิร์ฟเวอร์ จำเป็นต้องทำงานหนัก เนื่องจากเมื่อผู้ใช้เรียกใช้งาน ต้องทำการประมวลผลก่อน แล้วจึงค่อยส่งผลลัพธ์กลับไปให้ผู้ใช้นั้น ดังนั้นเครื่องผู้ใช้นั้นจำเป็นต้องมีทรัพยากรมากมาย (ผู้ใช้ชอบ ไม่ต้องเสียดังค์มาก) และยังสามารถควบคุมให้ทำงานได้ ในกรณีที่ผู้ใช้ใช้บราวเซอร์แตกต่างกัน แต่ในทางกลับกันเซิร์ฟเวอร์ต้องทำงานหนักขึ้น และอาศัยทรัพยากรเพิ่มขึ้น เทคนิคเหล่านี้ทำงานบนเซิร์ฟเวอร์



รูปที่ 2-6 การประมวลผลที่เซิร์ฟเวอร์

- **จาวาเซิร์ฟเล็ต** เป็นรูปแบบของโปรแกรมจาวาที่ทำงานบนเซิร์ฟเวอร์ โดยส่งผลลัพธ์ผ่านโปรโตคอลเอชทีทีพีกลับไปยังบราวเซอร์ของผู้ใช้ เนื่องจากเขียนด้วยจาวา จึงทำงานได้กับเซิร์ฟเวอร์ ที่มีแพลตฟอร์มแตกต่างกันได้
- **จาวาเซิร์ฟเวอร์เพจ (JSP)** เป็นผลพวงมาจากจาวาเซิร์ฟเล็ตสามารถสร้างเว็บด้วยภาษา เอชทีเอ็มแอล แล้วแทรกรูปแบบของภาษาจาวาสไคล์เจสพีลงไปในเว็บไซต์ โดยไฟล์เว็บนี้ต้องถูกประมวลผลในส่วนที่เป็นคำสั่งเจสพีก่อนส่งข้อมูลที่เป็นภาษาเอชทีเอ็มแอล และ ผลลัพธ์จากเจสพีกลับไป
- **แอ็กทีฟเซิร์ฟเวอร์เพจ (ASP)** เหมือนกับเจสพีแต่ภาษาที่อยู่รวมกับเอชทีเอ็มแอลเป็นสไคล์ของภาษาเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CGI Gateway คือรูปแบบการสร้างโปรแกรมที่ถูกเรียกใช้งานบนเว็บ แต่สามารถสื่อสารกับผู้ใช้ผ่านโปรโตคอลเอชทีทีพีในตัวแปรสภาพแวดล้อม (Environment Variables) ดังนั้น โปรแกรมที่ว่าจะเขียนด้วยภาษาอะไรก็ได้ เช่น เพิล, ซี, วิววลซีพลัสพลัส, วีบี, เดลไฟล์, ... (ยังมีอีกเยอะ) แต่โปรแกรมที่ว่าจะต้องเข้าใจ ตัวแปรสภาพแวดล้อมของเอชทีทีพี

2.10.3 ข้อเปรียบเทียบของการประมวลผลที่ไคลแอนท์และเซิร์ฟเวอร์

การทำงาน	ไคลแอนท์	เซิร์ฟเวอร์
อินเทอร์เน็ตทีฟ	ดีกว่า	จัดการยุ่งยาก และซับซ้อน
ความต้องการทรัพยากรที่บราวเซอร์	จำเป็นอย่างยิ่ง	-
ความต้องการทรัพยากรที่เซิร์ฟเวอร์	-	จำเป็นอย่างที่สุด เพราะต้องรองรับผู้ใช้จำนวนมากๆ
การติดตั้งและคอนฟิกที่บราวเซอร์	จำเป็น เมื่อต้องการทำงาน ในลักษณะเฉพาะ เช่นฐานข้อมูล	-
เวลาในการทำงาน	ขึ้นอยู่กับทรัพยากร ไคลแอนท์ และเทคนิค	ขึ้นอยู่กับเซิร์ฟเวอร์ และเทคนิคการใช้

ตารางที่ 2-2 ข้อเปรียบเทียบของการประมวลผลที่ไคลแอนท์และเซิร์ฟเวอร์

10.2.4 การดีพลอยแต่ละคอมโพเนนท์

- เอนเตอร์ไพซ์

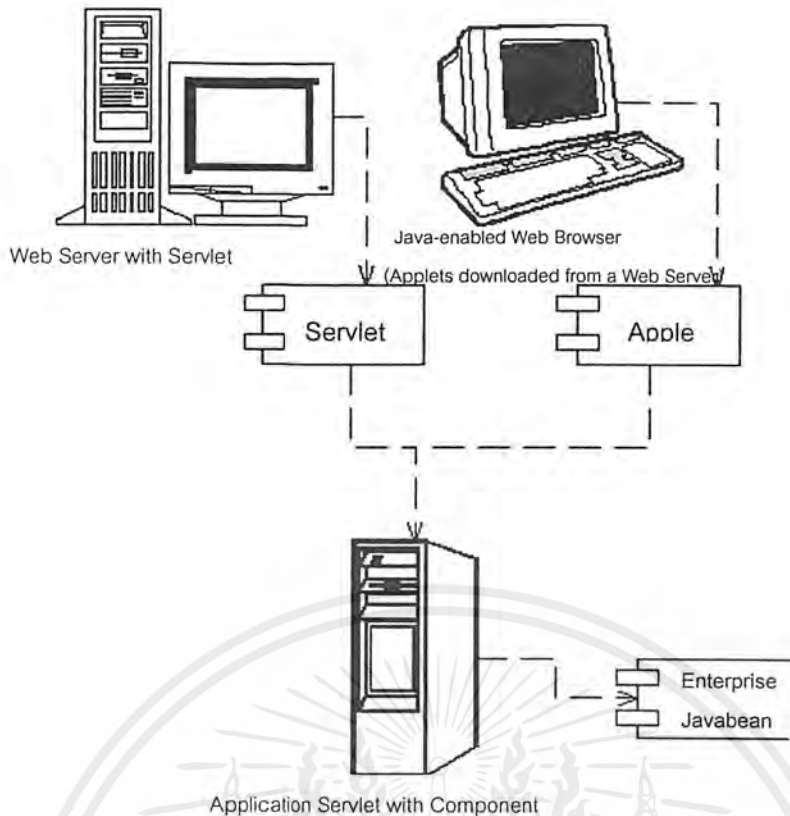
สามารถดีพลอยได้ในแอปพลิเคชันเซิร์ฟเวอร์โดยแอปพลิเคชันเซิร์ฟเวอร์จะจัดการคอนเทนเนอร์ในขณะเวลารัน (run time)

- แอปเพล็ต

สามารถดีพลอยได้บนหน้าเว็บ โดยจะมีแอปเพล็ตวิวเวอร์ (Appletviewer) ที่อยู่ในบราวเซอร์จัดการคอนเทนเนอร์ในเวลารัน

- เซิร์ฟเล็ต

สามารถดีพลอยได้บนเว็บเซิร์ฟเวอร์ โดยจะมีเซิร์ฟเล็ตเอ็นจินที่อยู่ในเว็บเซิร์ฟเวอร์จัดการคอนเทนเนอร์ในเวลารัน ดังรูป



รูปที่ 2-7 Applet, servlet และ Enterprise JavaBeans

จุดเด่นที่สุดของเอ็นเตอร์ไพซ์จาวาบีเอสคือมันสามารถติดต่อฐานข้อมูลข้ามเซิร์ฟเวอร์ได้ดี และมีลักษณะ float (คือลอย) อยู่ตลอดไม่เหมือนกับ ภาษาสคริปต์อื่น จึงทำให้เอ็นเตอร์ไพซ์จาวาบีเอสเหมาะสำหรับนำมาใช้กับงาน middleware ได้ดีมาก คาดว่าคงจะแทนคอร์บา (corba) ได้ไม่ยากเพราะความเด่นตรงที่เป็นจาวานั่นเอง คือคุยกันด้วยภาษาเดียวไม่เหมือนกับคอร์บา

2.10.5 ความแตกต่างระหว่างเอ็นเตอร์ไพซ์จาวาบีเอส, แอปเพล็ต และ เซิร์ฟเล็ต

- แอปเพล็ต

เป็นโปรแกรมจาวาที่สามารถดาวน์โหลดและเอ็กซีคิว (Execute) ได้ เช่น สามารถดาวน์โหลดจากเว็บเซิร์ฟเวอร์ลงในเว็บเบราว์เซอร์และจากเว็บเบราว์เซอร์แสดงผลผ่านยูสเซอร์อินเตอร์เฟซไปยังผู้ใช้ได้

- เซิร์ฟเล็ต

เป็นคอมโพเนนท์ที่สามารถใช้ในการขยายความสามารถของเว็บเซิร์ฟเวอร์ได้ สามารถร้องขอบริการ (Request) และตอบสนองบริการ (Response) ได้ เช่น เมื่อมีไคลเอ็นท์โฮสต์ (Client Host) เช่น เว็บเบราว์เซอร์มาร้องขอบริการก็จะตอบสนองบริการกลับไป ลักษณะการทำงานเช่นนี้ทำให้เซิร์ฟเล็ตเหมาะกับการทำงานเกี่ยวกับเว็บ เช่น เร็นเดอร์ (render) ไฟล์ เอชทีเอ็มแอล ติดต่อไปยังอี-คอมเมิร์ซ แคตตาล็อก (E-Commerce Catalog)

ทั้งแอปเพล็ตและเซิร์ฟเล็ตเหมาะกับการทำงานบนฝั่งไคลเอ็นท์ เช่น เร็นเดอร์จียูไอ (render GUIs: Render Graphic User Interfaces), ลอจิกที่เกี่ยวกับการแสดงผล (Presentation-relate Logic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **อินเทอร์เน็ตไชน**

เป็นคอมพิวเตอร์ที่ทำงานบนฝั่งเซิร์ฟเวอร์ เช่น เอ็กซ์คิวต์ลกริทีมที่ซับซ้อน หรือ ทำงานเกี่ยวกับทรานแซกชัน (Business Transaction) ที่มีปริมาณมากๆ โดยจะต้องมีทรัพยากรในการเอ็กซ์คิวต์ที่สูง, การทำงานที่เป็น ทรานแซกชัน, มีระบบรักษาความปลอดภัยรองรับสำหรับผู้หลายคน เซิร์ฟเวอร์ที่ใช้ เช่น แอปพลิเคชันเซิร์ฟเวอร์จะเป็นตัวจัดการคอมพิวเตอร์บนฝั่งเซิร์ฟเวอร์สำหรับอินเทอร์เน็ตไชน

สรุปแล้วก็คือแอปพลิเคชัน, เซิร์ฟเล็ตและอินเทอร์เน็ตไชนเป็นเทคโนโลยีที่มีความสำคัญทั้งหมด เช่น เราสามารถใช้จาวาบินในการสร้างประกอบกันขึ้นมาเป็นอินเทอร์เน็ตไชนที่สามารถดีพลอยได้และสามารถจัดการติดต่อผู้ใช้ไปยังอินเทอร์เน็ตไชนได้โดยใช้แอปพลิเคชันหรือเซิร์ฟเล็ต



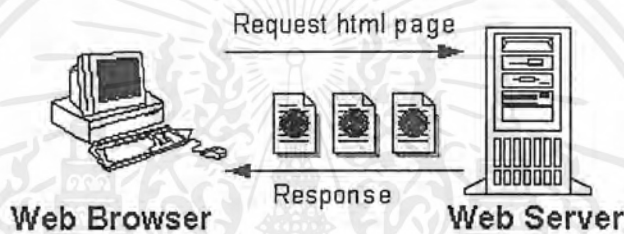
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

สถาปัตยกรรมบนฝั่งเซิร์ฟเวอร์

3.1 วิวัฒนาการของเซิร์ฟเวอร์ไซด์แอปพลิเคชัน (Server Side Application)

เซิร์ฟเวอร์ไซด์แอปพลิเคชันที่ถูกใช้บนเวปตั้งแต่สมัยแรก ๆ นั้นส่วนประกอบหลักของเวปไม่ค่อยจะมีอะไรซับซ้อนเท่าไร โดยใช้หลักการพื้นฐานของสถาปัตยกรรมไคลเอ็นท์/เซิร์ฟเวอร์ซึ่งส่วนที่เป็นไคลเอ็นท์ที่ในเว็บก็คือเว็บเบราว์เซอร์ยกตัวอย่างเช่น NCSA Mosaic, Lynx, Netscape และส่วนที่เป็นเซิร์ฟเวอร์ซึ่งก็คือเว็บเซิร์ฟเวอร์และโดยทั่วไปเว็บเซิร์ฟเวอร์หนึ่ง ๆ จะสามารถบริการส่งข้อมูลให้ตัวเว็บเบราว์เซอร์ได้หลายตัว ซึ่งจะเป็นลักษณะที่คนหลายคนสามารถที่จะเข้าไปดูข้อมูลต่าง ๆ จากเวปไซด์เดียวกัน (1 เซิร์ฟเวอร์) ได้ในเวลาเดียวกันนั่นเอง



รูปที่ 3-1 เบสิค client/server architecture เริ่มแรกสุดที่ใช้กันบนเวป

ต่อมาทั้งคนทำเว็บและคนดูเว็บก็คงจะเริ่มเบื่อบ้างจึงช่วยกันคิดค้นหาวิธีที่ทำให้เวปมีสีสันมากขึ้น เทคโนโลยีที่เกิดขึ้นตามมาก็คือ ซีจีไอ (Common Gateway Interface) หลักการของ ซีจีไอ ก็คือแทนที่ข้อมูลที่เราส่งไปให้เว็บเซิร์ฟเวอร์ (โดยผ่านทางเว็บเบราว์เซอร์) ก็จะถูกโปรเซสโดยตัวเว็บเซิร์ฟเวอร์อย่างเดียว ข้อมูลนี้ก็จะถูกส่งไปให้อีกโปรแกรมหนึ่งซึ่งตัวโปรแกรมนี้จะสามารถถูกเรียกได้โดยเว็บเซิร์ฟเวอร์เพื่อทำการประมวลผลข้อมูลที่มาจากเรา โดยทั้งนี้ข้อมูลของเราก็จะถูกโปรเซสที่ได้ก็ขึ้นอยู่กับเซตต่าง ๆ ที่ทางเว็บไซต์ที่เราเข้าไปดูกำหนดขึ้นมา ยกตัวอย่างเช่น ถ้าเราต้องการดูเพลงธรรมดา ๆ หลักจากที่เราคลิกไปที่ตัวลิงค์ตัวเว็บเซิร์ฟเวอร์ก็จะไปอ่านเพลง (html ไฟล์) ที่เราต้องการแล้วส่งกลับมาเป็นเท็กสตรีม (text stream) ให้เรา แต่ถ้าเราต้องการที่จะโพสต์ (post) ข้อความลงไปในเว็บไซต์ ทางเว็บเซิร์ฟเวอร์ก็จะส่งข้อความที่เราพิมพ์ไปให้โปรแกรมที่ทำหน้าที่ดูแลและเก็บข้อความที่เราส่งเข้าไป

หลักจากที่ ซีจีไอ กลายเป็นที่นิยมจนเป็นส่วนหนึ่งของ WWW Standard แล้วเซิร์ฟเวอร์ไซด์แอปพลิเคชันที่เป็นลูกหลานก็เกิดขึ้นกันมาอย่างมากมาย ยกตัวอย่างเช่น เอเอสพี, พีเอชพี, ColdFusion, เซิร์ฟเล็ต, เจเอสพีและอื่น ๆ (อย่างไรก็ตามในโครงการนี้จะกล่าวถึงเทคโนโลยีที่เกี่ยวข้องกับเซิร์ฟเวอร์ไซด์แอปพลิเคชันที่ใช้จาวาแพลตฟอร์มเป็นหลักเท่านั้น)

3.2 Multi-tier Architecture

เซิร์ฟเวอร์ที่ดีนั้นจะต้องสามารถทำงานได้อย่างมีประสิทธิภาพ, มีความน่าเชื่อถือ, มีประสิทธิภาพ และสามารถสนับสนุนการทำงานเมื่อมีผู้ใช้ทำงานพร้อมกันหลาย ๆ คนได้ รวมทั้งสนับสนุนการทำงานของไคลเอนต์ทั้งหมดได้เป็นอย่างดี เพราะการทำงานของไคลเอนต์นั้นจะขึ้นอยู่กับเซิร์ฟเวอร์กลางเป็นสำคัญ ถ้าเกิดเซิร์ฟเวอร์กลางเกิดดาวน์ขึ้นจะทำให้ยากที่จะกู้คืนใหม่ได้ และอาจมีส่วนที่เป็นอันตรายเข้าไปในระบบได้ ดังนั้นการที่จะทำการดีพลอยบนฝั่งเซิร์ฟเวอร์จะต้องทดสอบทีละชั้น

ในการทำดีพลอยเมนต์ที่ดีนั้นในทางลอจิกจะต้องแบ่งออกเป็นเลเยอร์ (Layer) โดยแต่ละเลเยอร์จะตอบสนองการทำงานที่ต่างกันและสามารถมีได้มากกว่า 1 คอมโพเนนต์ดังนี้

- **Presentation Layer**

เป็นคอมโพเนนต์ที่เกี่ยวกับการติดต่อกับผู้ใช้ เช่นพรีเซนต์เดชันเลเยอร์ (Presentation Layer) ของแอปพลิเคชันสามารถเขียนโดยใช้ภาษาวิวอลเบติก, พรีเซนต์เดชันเลเยอร์ในทางเว็บ-เบสดีพลอยเมนต์สามารถใช้จาวาเซิร์ฟเล็ต,จาวาเซิร์ฟเวอร์เพจ หรือจาวาแอปเพล็ต

- **Business Logic Layer**

เป็นคอมโพเนนต์ที่แก้ปัญหาเกี่ยวกับ Business problem ต้องมีเอนจินที่มีประสิทธิภาพสูง เช่น Catalog Engine, Pricing Engine โดยทั่วไปแล้วคอมโพเนนต์นี้ต้องเขียนด้วยภาษาที่มีระบบความปลอดภัยที่ดี เช่นจาวาหรือซีพลัสพลัส

- **Data Layer**

คอมโพเนนต์นี้จะถูกใช้โดยบีชีเนสลอจิกเลเยอร์ (Business Logic Layer) โดยศูนย์กลางที่ดาต้าเลเยอร์ (Data Layer) จะติดต่อไปถึงได้แก่ ฐานข้อมูลซึ่งสามารถติดต่อไปถึงได้หลายฐานข้อมูล

ข้อดีของการแบ่งเป็นแต่ละเลเยอร์คือ ทำให้สามารถเปลี่ยนแปลงเลเยอร์แต่ละเลเยอร์ได้โดยแทบไม่มีผลกระทบต่อเลเยอร์อื่น เพราะว่าแต่ละส่วนแยกเป็นอิสระจากกัน เช่น การเปลี่ยนแปลงฐานข้อมูลในส่วนดาต้าเลเยอร์

แต่ในการแบ่งการดีพลอยออกเป็นแต่ละเลเยอร์นั้นเป็นการแบ่งในทางลอจิกคอลซึ่งจะแตกต่างจากการแบ่งในเชิงฟิสิกคอลคือการแบ่งจะแบ่งเป็นหน่วยที่เรียกว่า "เทียร์ " (Tier) โดยจะมีสถาปัตยกรรมที่เป็น 2 เทียร์, 3 เทียร์ และแบบหลายเทียร์

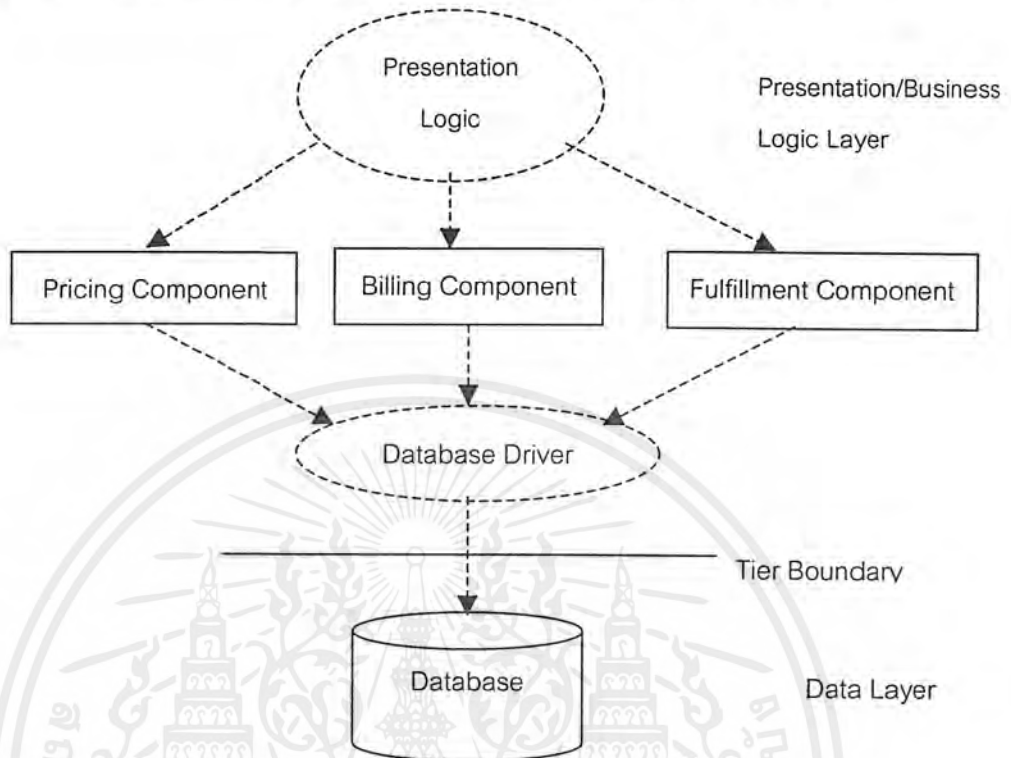
3.2.1 สถาปัตยกรรม 2 เทียร์ (2 - Tier Architecture)

ในสถาปัตยกรรมนี้จะเป็นการนำส่วนบีชีเนสลอจิกเลเยอร์ เข้าไปรวมกับเลเยอร์อื่นจะมี 2 แบบที่เป็นไปได้คือ การรวมบีชีเนสลอจิกเลเยอร์เข้ากับพรีเซนต์เดชันเลเยอร์ (Presentation Layer) และ รวม บีชีเนสลอจิกเลเยอร์เข้ากับดาต้าเลเยอร์

- **การรวมส่วนบีชีเนสลอจิกเลเยอร์เข้ากับพรีเซนต์เดชันเลเยอร์**

คือการรวมบีชีเนสลอจิกเลเยอร์เข้ากับพรีเซนต์เดชันเลเยอร์เอาไว้เป็นเทียร์แรก และเอาส่วนดาต้าเลเยอร์เอาไว้เป็นเทียร์ที่สอง ดังรูป 3-2 โดยเทียร์แรกนั้นจะเป็นไคลเอนต์และเทียร์ที่สองเป็นเซิร์ฟเวอร์ ทำให้สถาปัตยกรรมนี้มีลักษณะที่ไคลเอนต์นั้นจะอ้วน (Fat) ส่วนเซิร์ฟเวอร์นั้นจะผอม (Thin) ในสถาปัตย

กรรมนี้แอปพลิเคชันจากไคลเอ็นต์จะติดต่อกับส่วนดาต้าเลเยอร์โดยผ่านดาต้าบิชอปีไอ (Database bridge API) เช่น Open Database Connectivity (โอดีบีซี) หรือ Java Database Connectivity (JDBC)

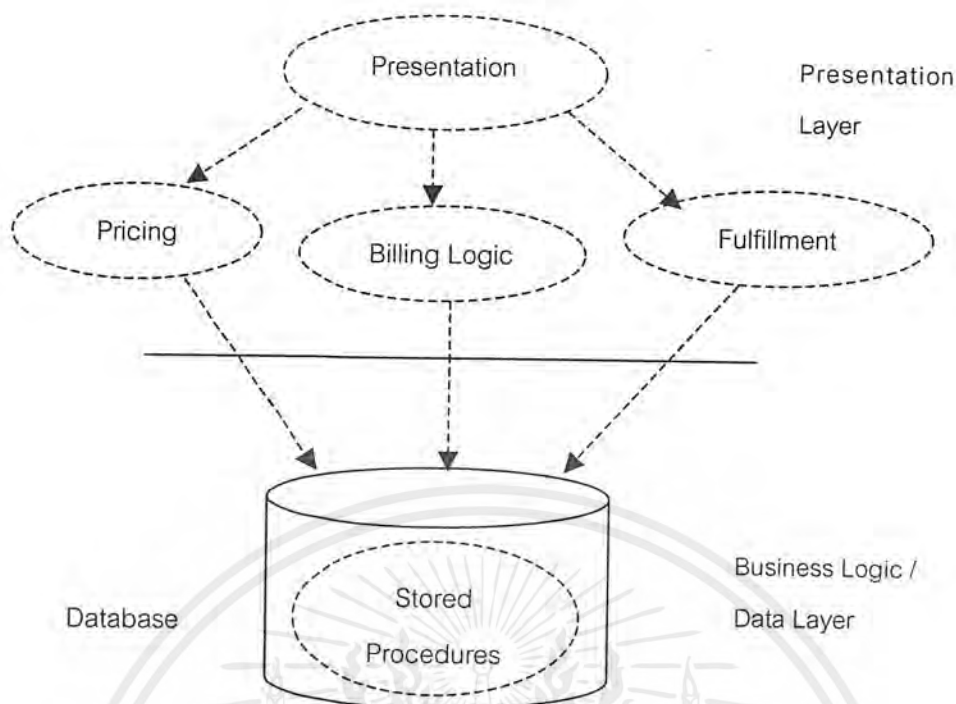


รูปที่ 3-2 การรวมส่วน Business Logic Layer เข้ากับ Presentation Layer

● การรวมส่วนบีซิเนสลอจิกเลเยอร์เข้ากับดาต้าเลเยอร์

คือ การรวมบีซิเนสลอจิกเลเยอร์เข้ากับดาต้าเลเยอร์เอาไว้เป็นเทียร์แรก และเอาส่วนพีริเซนต์เลเยอร์เอาไว้เป็นเทียร์ที่สอง ดังรูป 3.3 โดยคิดว่าเทียร์แรกเป็นเซิร์ฟเวอร์และเทียร์ที่สองเป็นไคลเอ็นต์ ทำให้สถาปัตยกรรมนี้มีลักษณะที่เซิร์ฟเวอร์นั้นจะอ้วน (Fat) ส่วนไคลเอ็นต์นั้นจะผอม (Thin) ในสถาปัตยกรรมนี้จะใส่บีซิเนสลอจิกเอาไว้ในฐานข้อมูล โดยฐานข้อมูลจะอนุญาตให้สามารถเอ็กซีคิวต์ (Execute) Context ที่อยู่ในฐานข้อมูลได้เมื่อสร้างโมดูลขึ้นมาเรียกใช้เรียกว่า "สตอร์โปรซีเจอร์ (Stored Procedures)" โดยการใส่ส่วนของลอจิกเข้าไปในสตอร์โปรซีเจอร์ซึ่งจะช่วยทำให้การทำงานมีประสิทธิภาพมากขึ้นเพราะว่าในการใส่ลอจิกเข้าไปในสตอร์โปรซีเจอร์ทำให้ระบบเครือข่าย (Network) ลดเวลาในการวิ่งกลับไปกลับมาจากลอจิกไปยังฐานข้อมูล แทนที่จะต้องคิวรี (Query) ข้อมูลจากฐานข้อมูล N ครั้ง เป็นการลดทราฟฟิก (Traffic) ภายในเครือข่าย ช่วยแก้ปัญหาที่มีในการรวมส่วนบีซิเนสลอจิกเลเยอร์เข้ากับพีริเซนต์เลเยอร์แต่ว่าสถาปัตยกรรมนี้ก็ไม่สามารถแก้ปัญหาได้ทั้งหมดดัง เช่น ภาษาของสตอร์โปรซีเจอร์จะผูกมัดทำให้ไคลเอ็นต์เป็นส่วนหนึ่งของฐานข้อมูล แต่จุดประสงค์ของการที่มีดาต้าบิชอปีไอต้องการทำให้สามารถเปลี่ยนแปลงแก้ไขฐานข้อมูลได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 การรวมส่วนบิซิเนสลอจิกเลเยอร์เข้ากับดาต้าเลเยอร์

3.2.2 ลักษณะของสถาปัตยกรรม 2 เทียร์

ลักษณะของสถาปัตยกรรมแบบ 2 เทียร์ คือ

1. ค่าใช้จ่ายสูง เพราะในไดเวอร์ของฐานข้อมูล (Database driver) จะต้องติดตั้งไคลเอ็นต์เทียร์แต่ละเทียร์ไว้ด้วย

2. ค่าใช้จ่ายเมื่อต้องการเปลี่ยนแปลงในการไดเวอร์ของฐานข้อมูลมีสูง โดยจะต้องติดตั้งไดเวอร์ของฐานข้อมูลภายในแต่ละไคลเอ็นต์ใหม่ทั้งหมดซึ่งเป็นค่าใช้จ่ายในการซ่อมบำรุง (Maintenance) ที่สูงมาก

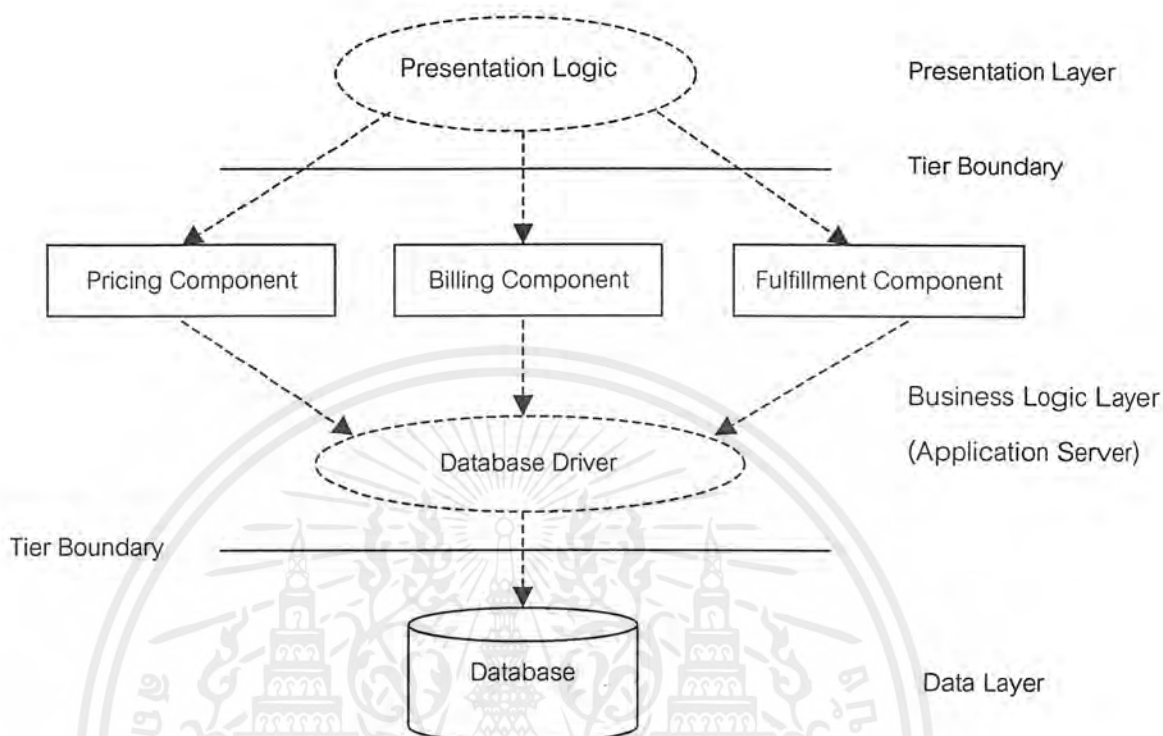
3. ค่าใช้จ่ายในการเปลี่ยนแปลงดาต้าเบสสกรีม่า (Database schema) มีสูง เนื่องจากเทียร์ที่เป็นไคลเอ็นต์นั้นจะต้องเข้าถึงฐานข้อมูลโดยตรงโดยผ่าน เจดบีซี หรือ โอดีบีซี หรือ เอสคิวแอล/เจซึ่งนั้นหมายความว่าไคลเอ็นต์จะต้องติดต่อกับดาต้าเบสสกรีม่าถ้าต้องการที่จะเปลี่ยนแปลงกระบวนการทางธุรกิจ (Business Process) ใหม่จะต้องดีพลอยแต่ละไคลเอ็นต์ใหม่ด้วย

4. ค่าใช้จ่ายเมื่อต้องการเปลี่ยนแปลงชนิดของฐานข้อมูลใหม่มีสูง เช่น เมื่อต้องการเปลี่ยนจากฐานข้อมูลที่เป็นรีเลชันนอลมาเป็นฐานข้อมูลที่เป็นอ็อบเจกต์ซึ่งไม่เพียงแต่จะต้องดีพลอย แต่ละไคลเอ็นต์ใหม่เท่านั้นแต่จะต้องแก้ไขโค้ดของไคลเอ็นต์ให้เหมาะสมกับชนิดของฐานข้อมูลด้วย

- 5. ในแต่ละครั้งที่บิซิเนสลอจิกเลเยอร์เข้ากับดาต้าเบสเลเยอร์จะต้องมีการวิ่งกลับไปกลับมาหลายรอบเพราะถูกแยกออกเป็นคนละส่วนกัน ทำให้เสียเวลาเป็นการจำกัดความสามารถของฐานข้อมูล และทำให้ระบบเครือข่ายให้ทำงานช้าลงและเป็นการลดแบนด์วิดท์ (bandwidth) การทำงานของผู้ใช้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 สถาปัตยกรรม N เทียร์



รูปที่ 3-4 แสดง 3-Tier Web-Based Deployment

- เป็นสถาปัตยกรรมที่เพิ่มเทียร์ให้เพิ่มขึ้นจากที่มีอยู่แล้วจากแบบ 2 เทียร์โดยจะแยกพีริเซนต์ชั้นเลเยอร์, บีซิเนสลอจิกเลเยอร์ และดาต้าเลเยอร์กันออกมาเป็นแต่ละเทียร์ในทางฟิสิกคอล ถ้าเป็นสถาปัตยกรรมที่มีมากกว่า 3 เทียร์จะแยกแต่ละเลเยอร์ออกจากกันอิสระ จากตัวอย่างดังรูป 3-4 เป็น 3-Tier เว็บเบสดีพลอยเมนต์ (Web-Based Deployment) โดยเทียร์จะแตกออกเป็นดังนี้

- พีริเซนต์ชั้นเทียร์ (Presentation Tier)

สามารถรันได้บนหลายเว็บเซิร์ฟเวอร์ภายในจะประกอบไปด้วยจาวาเซิร์ฟเล็ต, สคริปต์ (เช่น เอเอสพี, เจเอสพี) และลอจิกที่เชื่อมการทำงานของแต่ละส่วน

- บีซิเนสลอจิกเทียร์ (Business Logic Tier)

สามารถรันได้บนหลายแอปพลิเคชันเซิร์ฟเวอร์โดยแอปพลิเคชันเซิร์ฟเวอร์จะต้องมีคอนเทนเนอร์ที่ใช้ในการรันบีซิเนสลอจิกและมีการบริการ (Service) เพื่อรองรับการจัดการคอมโพเนนท์ที่มีประสิทธิภาพ เช่น แอปพลิเคชันเซิร์ฟเวอร์จะจัดการการเข้าถึงข้อมูลภายในฐานข้อมูลให้กับบีซิเนสคอมโพเนนท์

- ดาต้าเทียร์ (Data Tier)

ประกอบด้วยฐานข้อมูลซึ่งสามารถมีได้มากกว่า 1 ฐานข้อมูลและสตอร์โปรซิเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 ลักษณะของสถาปัตยกรรม N เทียร์

1. ค่าใช้จ่ายในการดีพลอยต่ำ เพราะว่าไคลเอนต์ฐานข้อมูลจะถูกติดตั้งบนฝั่งเซิร์ฟเวอร์มากกว่าที่จะติดตั้งบนฝั่งไคลเอนต์
2. ค่าใช้จ่ายเมื่อมีการเปลี่ยนแปลงฐานข้อมูลต่ำ เพราะว่าไคลเอนต์ไม่ได้เข้าถึงฐานข้อมูลโดยตรง แต่จะเข้าถึงโดยผ่านเทียร์ตรงกลาง ทำให้เมื่อมีการเปลี่ยนแปลงดาต้าเบสกรีม้าจะเปลี่ยนเพียงไคลเอนต์ของฐานข้อมูลไม่ต้องทำการดีพลอยไคลเอนต์ใหม่
3. ค่าใช้จ่ายเมื่อมีการเปลี่ยนแปลงบิซิเนสลอจิกต่ำ เพราะไม่ต้องคอมไพล์และดีพลอยเทียร์ของไคลเอนต์ใหม่
4. มีไฟร์วอลล์ (Firewalls) เป็นตัวรักษาความปลอดภัย คือ ป้องกันการเข้าถึงข้อมูลภายในแอปพลิเคชันเซิร์ฟเวอร์
5. มีการจัดการที่มีประสิทธิภาพเมื่อมีการเรียกใช้ทรัพยากรภายนอก (External Resources) เช่น พรินเตอร์ และเมื่อขณะที่ไคลเอนต์ทำงานไม่ต้องใช้ทรัพยากร (Resource) เช่น การ render การแสดงผลที่ต้องส่งไปยังผู้ใช้ ซึ่งขณะนั้นไม่จำเป็นต้องใช้ทรัพยากร ทรัพยากรจะถูกจัดสรรไปให้ไคลเอนต์อื่นที่ต้องการที่ต้องทำเช่นนี้เพราะว่า
 - ความเสี่ยงในการเข้าถึงฐานข้อมูลนั้นมีสูงจึงต้องจำกัดจำนวนของผู้ที่ต้องการเข้าถึงฐานข้อมูลในขณะใดขณะหนึ่ง
 - ไม่จำเป็นที่ต้องติดต่อไปยังทรัพยากรตลอดเวลา
 - เป็นการเพิ่มประสิทธิภาพของแอปพลิเคชัน
6. ในแต่ละเทียร์มีการทำงานอย่างอิสระ เช่น สามารถเพิ่มหรือแก้ไขฐานข้อมูลได้โดยแก้ไขโค้ดน้อยมากและต้องคอมไพล์ใหม่เฉพาะเทียร์ที่แก้ไขเท่านั้น
7. เมื่อมีเทียร์ใดเกิดโอเวอร์โหลด (Overload) เทียร์อื่นก็ยังคงมีคุณสมบัติครบเหมือนเดิม
8. ถ้ามีเทียร์ใดเกิดเออเรอร์ (Error) ขึ้นมาก็จะเป็นเฉพาะเทียร์นั้นไม่เกี่ยวกับเทียร์อื่น เช่น ถ้าแอปพลิเคชันเซิร์ฟเวอร์เกิดแครช (Crashed) เว็บเซิร์ฟเวอร์จะรายงานไปยังบราวเซอร์ของไคลเอนต์ว่า “Site Down”
9. ค่าใช้จ่ายในการซ่อมบำรุงสูง

3.3 Java 2 Platform, Enterprise Edition (J2EE)

ผลิตภัณฑ์จากสถาปัตยกรรมบนฝั่งเซิร์ฟเวอร์ (Server-Side Component Architectuer Solutions) นั้นได้มีออกมาโดยค่ายต่างๆดังนี้

1. Microsoft's Distributed interNet Application Architecture (DNA)
 2. Sun Microsystem's Java 2 Platform,Enterprise Edition (J2EE)
 3. The Object Management Group's COBRA Standard
- แต่ในที่นี้จะขออธิบายเฉพาะค่ายของ SUN ซึ่งคือ เจทูอีอี

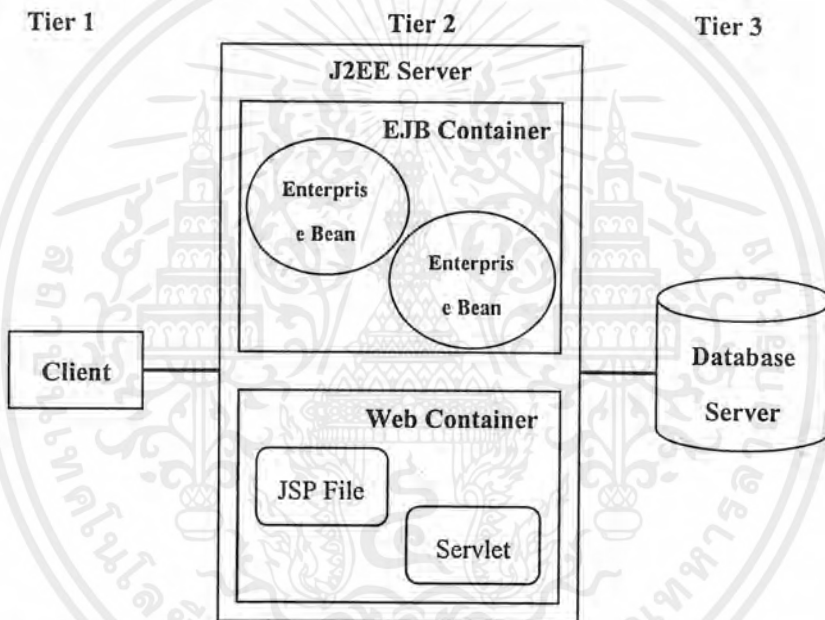
3.3.1 Sun Microsystem's Java 2 Platform,Enterprise Edition (J2EE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาจาวาจะเป็นภาษาที่สะดวกในการเขียนคอมโพเนนต์ใส่ไว้บนฝั่งเซิร์ฟเวอร์เพราะว่าตลาดของฝั่งเซิร์ฟเวอร์จะมี UNIX และเมนเฟรม (Mainframe) เป็นแกนหลักซึ่งหมายความว่าภาษาที่สามารถข้ามแพลตฟอร์มได้จะมีความสำคัญมากเพราะผู้พัฒนาจะต้องเขียนคอมโพเนนต์เพียงครั้งเดียวแต่สามารถดีพลอยได้บนหลายเอ็นไวรอนเมนต์

ในการตอบสนองของจาวาบนฝั่งเซิร์ฟเวอร์ชั้นจึงได้ออกแพลตฟอร์มที่ชื่อว่า “Java 2 Platform, Enterprise Edition (J2EE) ซึ่งเป็นแพลตฟอร์มที่มีความอิสระ (independent), สนับสนุนกับผู้ใช้ได้หลาย ๆ คน (Multi-user) และยังมีระบบรักษาความปลอดภัย โดยแกนหลักของ เจทูอีอี คือ EJB (Enterprise Javabeans) ซึ่งเป็นมาตรฐานในการสร้างคอมโพเนนต์บนฝั่งเซิร์ฟเวอร์โดยใช้ภาษาจาวา

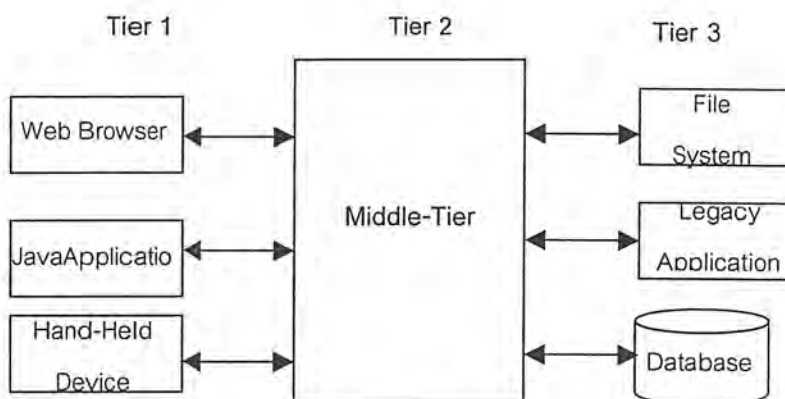
3.3.2 สถาปัตยกรรม เจทูอีอี



รูปที่ 3-5 แสดงโครงสร้างสถาปัตยกรรมของ เจทูอีอี ในระบบ 3-Tier Server

จากรูป 3-5 จะเห็นว่า เจทูอีอี นั้นเป็นมัลติเทียร์เซิร์ฟเวอร์ (Middle-Tier Servers) ซึ่งเป็นส่วนสำคัญหลักในระบบ 3-เทียร์เซิร์ฟเวอร์โดยมีหน้าที่บริการการร้องขอจากไคลเอ็นท์แบ่งการทำงานในการติดต่อกับระบบแบคออฟฟิศ (back office) และฐานข้อมูลออกเป็นส่วนๆ เพื่อลดความซับซ้อน นอกจากนี้ยังช่วยในการจัดการกับทรัพยากรต่าง ๆ ของระบบรวมทั้งการรักษาความปลอดภัย โดยมัลติเทียร์เซิร์ฟเวอร์นี้จะสนับสนุนการทำงานหลายอย่างของไคลเอ็นท์ เช่น เว็บเบราว์เซอร์, จาวาแอปพลิเคชัน และไคลเอ็นท์จะเป็นตัวคอยดูแลเกี่ยวกับการติดต่อกับผู้ใช้นั้น การที่ยกหน้าที่ต่างเหล่านี้ให้เป็นของมัลติเทียร์เลยทำให้ไคลเอ็นท์มีขนาดเล็ก, ไม่ซับซ้อนและพัฒนาไปได้ช้าทำให้เราสามารถรวมไคลเอ็นท์ใหม่เข้ากับแอปพลิเคชันและฐานข้อมูลที่มีอยู่ได้มัลติเทียร์เซิร์ฟเวอร์ทำให้คุณสามารถสร้างแอปพลิเคชันแต่ละ แอปพลิเคชัน ที่มีขนาดใหญ่ได้ และสถาปัตยกรรม เจทูอีอี จึงเป็นอีกทางเลือกในการพัฒนามัลติเทียร์เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-6 แสดงโครงสร้างของระบบ 3-Tier Client/Server Application

จากรูป 3-6 นั้น-tier จะประกอบด้วยหลาย ๆ ไคลเอ็นท์ที่ร้องขอการบริการไปที่ มิดเดิลเทียร์ (middle-tier server) ซึ่งอยู่ใน-tier 2 โดยใน-tier 2 จะเข้าถึงข้อมูลจากระบบใน-tier 3 โดยจะทำการ โปรเซส (Process Rule) ที่เข้ามา และส่งผลลัพธ์กลับไปยังไคลเอ็นท์ใน-tier 1

เจทูอี เป็นแพลตฟอร์ม ที่สร้างขึ้นโดยมีเป้าหมายเพื่อพัฒนา และจัดการแอปพลิเคชันแบบมัลติ-tier เซิร์ฟเวอร์ โดย เจทูอี เซิร์ฟเวอร์มีความสามารถด้านต่าง ๆ ดังนี้

- สามารถใช้หลักการของเนมมิงและไดเรกทอรีซึ่งจะอนุญาตให้โปรแกรมของการใช้งานและคอมโพเนนต์ ผ่านจาวาเนมมิงและไดเรกทอรีอินเตอร์เฟส (เจเอ็นดีไอ)
- มีความสามารถที่จะสร้างความปลอดภัยให้กับระบบโดยให้มีการออกเทนทิเคชัน (Authentication) ซึ่งจะต้องให้ผู้ใช้ล็อกอินเข้าสู่ระบบ
- สนับสนุนการใช้เซทที่พีซึ่งทำให้เว็บเบราว์เซอร์เข้าถึงเซิร์ฟเล็ตและไฟล์ เจเอสพี
- สนับสนุนการทำงานของอีเจบีโดยอนุญาตให้ไคลเอ็นท์เรียกเมทอดของเอนเตอร์ไพซ์บีนได้
- เป็นแพลตฟอร์มที่รวบรวมความสามารถในการสร้างและจัดการซอฟต์แวร์ในระดับเอนเตอร์ไพซ์
- อนุญาตให้แอปพลิเคชันในระดับเอนเตอร์ไพซ์สามารถรันได้ทุกที่ และรวบรวมความสามารถในการบริการระดับเอนเตอร์ไพซ์ไว้อย่างสมบูรณ์

3.3.3 เทคโนโลยีของ เจทูอี (J2EE Technologies)

เทคโนโลยีของ เจทูอี ประกอบด้วยส่วนต่างๆดังนี้

- เอนเตอร์ไพซ์ จาวาบีนส์ (Enterprise JavaBean :EJB)
- ข้อกำหนดทางด้านการสื่อสารระหว่างเครื่องคอมพิวเตอร์ในรูปแบบของ Java Remote Method Invocation (RMI)
- การเชื่อมต่อกับฐานข้อมูลด้วยภาษาจาวาหรือ Java Database Connectivity (JDBC)
- มาตรฐานที่ใช้ในการติดต่อกับออบเจกต์ที่ทำงานร่วมกันภายใต้ระบบเครือข่ายหรือ Java Naming and Directories Interface (เจเอ็นดีไอ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การจัดการกับทรานส์แอ็กชันหรือ Java Transaction APIs (JTA), Java Transaction Services (JTS)
- การให้บริการด้านการส่งแมสเสจโดยภาษาจาวาหรือ Java Messaging Services (JMS)
- จาวาเซิร์ฟเล็ทและ จาวาเซิร์ฟเวอร์เพจ
- Java IDL
- จาวาเมล (JavaMail)
- คอนเน็กเตอร์ (Connectors)
- ความสามารถในการแยกส่วนของข้อมูลเพื่อประโยชน์ในการแสดงผลโดยใช้ Extensible Markup Language (XML)

3.4 เอนเตอร์ไพซ์ จาวาบีนส์ (Enterprise JavaBean : EJB)

จุดเด่นที่สุดของเอนเตอร์ไพซ์ จาวาบีนส์คือบีนที่สามารถติดต่อฐานข้อมูลข้ามเซิร์ฟเวอร์ได้ดี และมีลักษณะ float หรือลอยอยู่ตลอดไม่เหมือนกับ ภาษาสคริปต์อื่น จึงทำให้อีเจบีเหมาะสำหรับนำมาใช้กับงานมิดเคิลแวร์ (middleware) ได้ดีมาก คาดว่าคงจะแทนคอร์บายได้ไม่ยากเพราะความเด่นตรงที่เป็น จาวานั้นเอง คือถูกกันด้วยภาษาเดียวไม่เหมือนกับคอร์บาย นอกจากนี้เอนเตอร์ไพซ์จาวาบีนส์ยังเป็นส่วนประกอบส่วนหนึ่งของเซิร์ฟเวอร์ที่เขียนด้วยภาษาจาวามี 2 ชนิด คือ

3.4.1 เซสชันบีน (Session Beans)

เซสชันบีนเป็นเสมือนไคลเอ็นต์ใน เจทูอีอี เซิร์ฟเวอร์ซึ่งไคลเอ็นต์จะทำการติดต่อกับ เจทูอีอี เวิร์เวอร์ โดยการเรียกเมทอดของเอนเตอร์ไพซ์บีน เช่น ออนไลน์ช้อปปิ้งไคลเอ็นต์จะเรียกเมทอด enterOrder ของเซสชันบีนให้สร้างลำดับไคลเอ็นต์สามารถเปลี่ยนแปลงเซสชันบีนได้ สามารถคิดได้ว่าเซสชันบีนเป็นส่วนขยายของไคลเอ็นต์ในแต่ละเซสชันบีน สามารถมีได้เพียงไคลเอ็นต์เดียวและอยู่ไม่ถาวรเมื่อไคลเอ็นต์ยกเลิกการติดต่อก็จะทำให้เซสชันบีนยกเลิกไปด้วย นอกจากนี้เซสชันบีนยังแบ่งออกเป็น 2 ประเภท คือ สเตตฟูลเซสชันบีน (stateful session beans) และสเตตเลสเซสชันบีน (stateless session beans) ซึ่งจะกล่าวถึงรายละเอียดในบทที่ 5

3.4.2 เอ็นติตี้บีน (Entity Bean)

เอ็นติตี้บีนเป็นเสมือนอ็อบเจ็กต์ที่เรากำหนดขึ้นถูกเก็บไว้ในฐานข้อมูล เช่น เอ็นติตี้บีนอาจแทนลูกค้า ซึ่งต้องเก็บเป็นโรว์ในตารางลูกค้าของรีเลชันนอลดาต้าเบส ข้อมูลของเอ็นติตี้บีนจะไม่ถูกเก็บใน รีเลชันนอลดาต้าเบสสามารถเก็บไว้ในอ็อบเจ็กต์-ดาต้าเบส อาจเก็บเป็นไฟล์, แอปพลิเคชัน โดยชนิดการเก็บนั้นจะขึ้นอยู่กับเทคโนโลยีของ EJB

เอ็นติตี้บีนเป็นอีเจบีที่อยู่ถาวร สามารถถูกจัดการโดยเอ็นติตี้บีนด้วยกันหรืออีเจบีคอนเทนเนอร์ โดยเอ็นติตี้บีนต้องการให้เขียนโค้ดเข้าถึงข้อมูลภายในบีนเพราะตัวเอ็นติตี้บีนจริง ๆ จะอยู่ในดาต้าเบส เซิร์ฟเวอร์ เช่น ภายในเอ็นติตี้บีนลูกค้าจะมีค่าส่ง เอสคิวแอล ในการเข้าถึงรีเลชันนอลดาต้าเบสผ่าน เจดีบีซี โดย เอ็นติตี้บีนยังแบ่งย่อยได้เป็น 2 ประเภท คือ บีนเมนเนจเพอร์ซิสแทนท์ (bean-managed persistent)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ คอนเทนเนอร์แมนเนจเพอร์ซีสแทนท์ (container-managed persistent) ซึ่งจะกล่าวถึงในบทที่ 5 เช่นกัน

3.5 รีโมตเมทอดอินโวลเคชัน (Remote Method Invocation : RMI) and RMI-IIOP

อาร์เอ็มไอหรือรีโมตเมทอดอินโวลเคชันเป็นข้อกำหนดทางด้านการสื่อสารระหว่างเครื่องคอมพิวเตอร์ข้อหนึ่งที่สามารถนำไปใช้ประโยชน์ได้ในลักษณะเดียวกันกับ อาร์พีซี (Remote Procedure Call) ในขณะที่ อาร์พีซี ถูกนำไปใช้กับระบบซอฟต์แวร์ที่มีการโปรแกรมแบบสตรัคเจอร์โมเดลแต่ อาร์เอ็มไอจะใช้กับระบบซอฟต์แวร์ที่มีการโปรแกรมแบบอ็อบเจกต์อเรียนเตดโมเดล

คิสทริบิวชิสเต็ม (Distributed system) หรือ ระบบการประมวลผลแบบกระจาย มีความต้องการประสิทธิภาพที่สูงในการติดต่อสื่อสารระหว่างโพรเซสซึ่งอยู่ต่างเครื่องกัน ภาษาจาวาสนับสนุนการใช้ช็อกเก็ตซึ่งเป็นกลไกการติดต่อสื่อสารระหว่างโพรเซสแบบพื้นฐานที่อยู่ในระดับดี เนื่องจากมีความยืดหยุ่น และมีประสิทธิภาพเพียงพอสำหรับการติดต่อสื่อสารที่ใช้กันทั่วไป แต่ช็อกเก็ตมีความต้องการโพรโตคอลซึ่งมีการทำงานในลักษณะ โคลเอ็นท์/เซิร์ฟเวอร์ที่จะจัดการการติดต่อสื่อสารในระดับแอปพลิเคชันเพื่อที่จะเข้ารหัส และถอดรหัสแอสเซสที่จะใช้เพื่อการแลกเปลี่ยน ซึ่งการออกแบบโพรโตคอลนี้ย่อมเป็นภาระของผู้พัฒนาระบบซอฟต์แวร์ และโพรโตคอลที่ต้องพัฒนาขึ้นเองที่อาจเกิดข้อผิดพลาดขึ้น อันจะส่งผลถึงประสิทธิภาพของระบบที่นำโพรโตคอลนี้มาใช้ด้วย

อาร์พีซี เป็นข้อกำหนดตัวหนึ่งที่มีความสามารถดังความต้องการข้างต้น ซึ่งจะสร้างการติดต่อสื่อสารให้เกิดขึ้นได้ในลักษณะของการเรียกใช้โพรซีเจอร์ (Procedure call) แทนการส่งข้อมูลหรือแอสเซสผ่านช็อกเก็ตโดยตรงเมื่อเกิดความจำเป็นที่ต้องมีการติดต่อสื่อสารระหว่างอ็อบเจกต์ในระดับโปรแกรม ซึ่งอ็อบเจกต์เหล่านั้นมีอยู่ในเครื่องที่ต่างเครื่องกัน แต่เราไม่สามารถนำ อาร์พีซี มาดัดแปลงใช้กับคิสทริบิวชิสเต็มที่เป็นอ็อบเจกต์โมเดลได้ ฉะนั้นคิสทริบิวชิสเต็มที่เป็นอ็อบเจกต์โมเดลจึงต้องการข้อกำหนดตัวใหม่ออย่าง RMI

จาวาอาร์เอ็มไอถูกพัฒนาขึ้นโดยทีมงานพัฒนาซอฟต์แวร์ของซันไมโครซิสเต็มซึ่งออกแบบมาให้ทำงานได้ในสภาพแวดล้อมของจาวาท่านั้น ในขณะที่อาร์เอ็มไอตัวอื่นๆ สามารถนำมาใช้กับอ็อบเจกต์ของจาวาได้ แต่ไม่มีความสะดวก ไม่มีความราบรื่น ในการติดต่อสื่อสารระหว่างอ็อบเจกต์ของระบบเหล่านั้นกับ อ็อบเจกต์ของจาวาเท่าที่ควร (อย่างเช่น คอรับาที่สามารถจัดการกับสภาพแวดล้อมที่ต่างระบบกันได้ ซึ่งอ็อบเจกต์ของคอรับาจำเป็นต้องมีรูปแบบของภาษาที่เป็นกลาง) ในทางกลับกันอาร์เอ็มไอที่เป็นของจาวา (Java RMI) โดยเฉพาะ จะทำให้ความสามารถของอ็อบเจกต์ของจาวามาใช้ได้อย่างเต็มที่

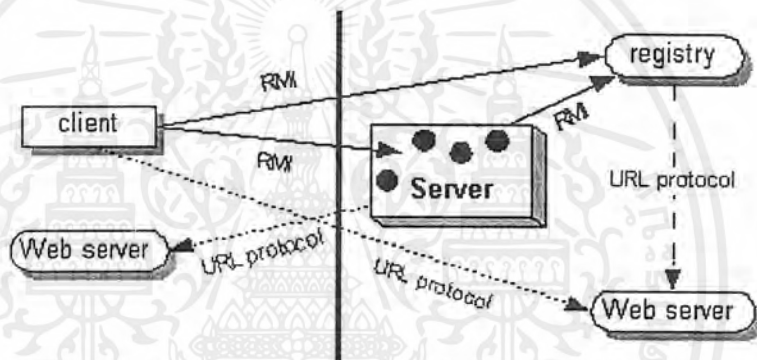
โปรแกรมที่ถูกพัฒนาขึ้นมาโดยใช้เทคโนโลยีของจาวาสามารถที่จะเรียกใช้อ็อบเจกต์ที่อยู่ต่างเครื่องได้โดยอาศัยค่าอ้างอิงของอ็อบเจกต์เหล่านั้นซึ่งมันได้รับมา โดยการได้มาของค่าอ็อบเจกต์นี้อาจได้มาจาก

- เข้าไปค้นหามาจากบูตสเตรพเนมมิงเซอร์วิส (Bootstrap naming service) ที่ระบบอาร์เอ็มไอได้จัดเตรียมไว้ให้
- ได้จากอาร์กิวเมนต์ (Argument) หรือรีเทอร์นเวลู (Return value) ที่ได้มาโดยการเรียกหาเมทอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องทุกเครื่องสามารถเรียกใช้อ็อบเจกต์จากต่างเครื่องได้ โดยไม่มีการกำหนดสถานะไว้ว่า เครื่องนี้ต้องเป็นเซิร์ฟเวอร์หรือไคลเอ็นต์แต่เพียงอย่างเดียวตลอดเวลา และอาร์เอ็มไอยังใช้อ็อบเจกต์เรียลไลเซเบิล (Object serialization) เพื่อทำการมาร์แชล (Marshal) และอันมาร์แชล (Unmarshal) ตัวพารามิเตอร์โดยไม่ตัดข้อมูลที่จะบอกชนิดของอ็อบเจกต์ทำให้สามารถจัดการกับอ็อบเจกต์ชนิดที่ผู้ใช้ (ผู้พัฒนาจาวาแอปพลิเคชัน) กำหนดขึ้นเองได้

จากรูป 3-7 นี้จะอธิบายถึง RMI ดิสทริบิวต์แอปพลิเคชันที่มีการใช้รีจิสทรีเพื่อที่จะสามารถเข้าไปอ้างถึงรีโมทอ็อบเจกต์ โดยเซิร์ฟเวอร์จะเป็นผู้เรียกตัวรีจิสทรีเพื่อที่จะกำหนดชื่อให้กับรีโมทอ็อบเจกต์ ส่วนตัวไคลเอ็นต์นั้นจะมองหารีโมทอ็อบเจกต์โดยใช้ชื่อของตัวรีโมทอ็อบเจกต์นั้นที่อยู่ในเซิร์ฟเวอร์รีจิสทรี (server's registry) จากนั้นก็จะทำการเรียกขอเมทอดที่มีอยู่ในรีโมทอ็อบเจกต์นั้น นอกจากนั้นในรูปยังแสดงถึง RMI ซิสเต็มที่มีการใช้เว็บเซิร์ฟเวอร์ในการโหลดไบต์โค้ดจากเซิร์ฟเวอร์ไปยังไคลเอ็นต์และจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์เมื่อมีการใช้อ็อบเจกต์นั้นอีกด้วย



รูปที่ 3-7 แสดง RMI distributed application

3.6 จาวาเนมมิง และ ไดเรกทอรีอินเตอร์เฟซ (Java Naming and Directory Interface : เจเอ็นดีไอ)

เจเอ็นดีไอ นั้นถือว่าเป็นคอมโพเนนท์ที่สำคัญของ Java 2 Platform, Enterprise Edition (J2EE) โดยจะมีอินเตอร์เฟซมาตรฐานที่ใช้ในการติดต่อกับผู้ใช้ , เมทชีน (machine) , เครือข่าย (network), อ็อบเจกต์ และบริการต่างๆ เช่น เราสามารถใช้ เจเอ็นดีไอ ในการเข้าติดต่อกับปรีนเตอร์ที่อยู่ในระบบบริหารเน็ตที่ทำงานร่วมกันอยู่ได้ นอกจากนี้ยังสามารถใช้ในการเข้าถึงจาวาอ็อบเจกต์หรือในการเชื่อมต่อกับฐานข้อมูลได้อีกด้วย นอกจากนี้ใน EJB, เจเอ็นดีไอ ยังถูกใช้อย่างมากมายในการเข้าถึงอ็อบเจกต์ข้ามระบบเครือข่าย

3.6.1 เนมมิง และ ไดเรกทอรี (Naming and Directory Services)

เนมมิงเซอร์วิสนั้นเหมือนกับโอเปอเรเตอร์ที่ทำงานกับโทรศัพท์ (telephone operator) ที่เมื่อเราต้องการเรียกไปยังอีกคนหนึ่งแต่เราไม่รู้เบอร์โทรของคนนั้น ดังนั้นเราจึงต้องโทรไปยังศูนย์บริการลูกค้า เพื่อที่จะขอข้อมูลเบอร์โทรศัพท์ของคนนั้นโดยการบอกชื่อของคนที่ต้องการติดต่อ จากนั้นโอเปอเรเตอร์จะทำการตรวจหาและทำการเชื่อมต่อไปยังบุคคลนั้นให้แก่เรา ดังนั้นเนมมิงเซอร์วิสจึงมีหน้าที่ดังนี้

- ทำการกำหนดชื่อให้กับออบเจกต์ซึ่งเรียกว่าการไบด์คิงเนม (binding name) ให้กับออบเจกต์
- ให้บริการในการหาออบเจกต์เมื่อทราบชื่อของออบเจกต์นั้น หรือเรียกว่าการลุกกิงอัพ (looking up) หรือเซิร์ชชิงอ็อบเจกต์ (searching object)

ตัวอย่างการเนมมิงอื่น ๆ เช่น การเชื่อมต่อไปยังแมชชีนอื่น ๆ บนเน็ตเวิร์กนั้น โดยโดเมนเนมซิสเต็ม (Domain Name System :DNS) จะถูกใช้ในการแปลงแมสชีนเนมไปเป็นไอพีแอสแดรส (IP address)

โดยปกติแล้วเนมมิงเซอร์วิสจะสามารถหาออบเจกต์ที่อยู่ทั่วไปได้ แต่จะไม่สามารถหาออบเจกต์บางชนิดได้ซึ่งก็คือไคลเอนต์หรือออบเจกต์หรือไคลเอนต์เ็นติตี้ซึ่งมีความแตกต่างจากออบเจกต์ทั่วไปคือการที่สามารถเก็บแอตทริบิวต์ต่าง ๆ ไว้กับไคลเอนต์หรือออบเจกต์ได้เพื่อใช้ในวัตถุประสงค์ต่าง ๆ เช่น สามารถใช้ไคลเอนต์หรือออบเจกต์ในการเก็บข้อมูลของผู้ใช้เช่นพาสเวิร์ดซึ่งถ้าเรามีแอปพลิเคชันที่จะต้องมีการ ออเท้นทิเคชัน (authentication) เราก็สามารถเก็บล็อกอินและพาสเวิร์ดของผู้ใช้ไว้ในส่วนของ แอตทริบิวต์ใน ไคลเอนต์หรือออบเจกต์ได้และเมื่อไคลเอนต์ทำการขอการเชื่อมต่อมายังแอปพลิเคชันนั้น ไคลเอนต์ก็ทำการป้อนล็อกอินและพาสเวิร์ดซึ่งเราสามารถนำมาเปรียบเทียบกับล็อกอินเนมและพาสเวิร์ดเก็บไว้เป็นแอตทริบิวต์ของไคลเอนต์หรือออบเจกต์ได้เพื่อดูว่าข้อมูลนั้นถูกต้องหรือไม่

ไคลเอนต์หรือเซอร์วิสเป็นเนมมิงเซอร์วิสที่ให้บริการในการหาไคลเอนต์หรือออบเจกต์ที่เก็บแอตทริบิวต์ต่าง ๆ เอาไว้ภายใน

3.6.2 ข้อดีของ เจเอ็นดีไอ

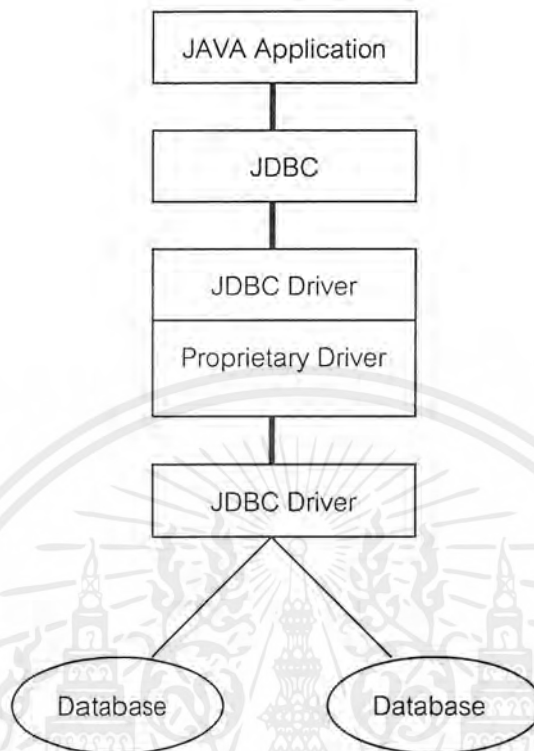
ข้อดีของ เจเอ็นดีไอ นั้นเป็นดังนี้

- เจเอ็นดีไอ นั้นเป็นระบบที่ช่วยในการเข้าถึงข้อมูลทุกชนิดที่อยู่ภายในไคลเอนต์หรือออบเจกต์ได้ เช่น ล็อกอิน ,พาสเวิร์ด, หมายเลขโทรศัพท์, อีเมลแอสเดส, ที่อยู่ตามทะเบียนบ้าน หรือ เน็ตเวิร์กแอสเดส
- เจเอ็นดีไอ เป็นซิงเกิลเอนทิตี้ที่ใช้ในการเข้าถึงไคลเอนต์ต่าง ๆ ด้วยโปรโตคอลต่าง ๆ กัน
- เจเอ็นดีไอ นั้นมีการทำงานที่ยืดหยุ่นโดยสามารถนำไปใช้ร่วมกับไคลเอนต์ต่าง ๆ ได้ โดยไม่มีผลกระทบต่อส่วนของโค้ดที่ฝั่งไคลเอนต์ (client code)

การใช้ เจเอ็นดีไอ นั้นทำให้เราสามารถทำการ อ่านหรือเขียนจาวาอ็อบเจกต์ทั้งหมดที่อยู่ในไคลเอนต์ต่าง ๆ ได้

3.7 จาวาด้าเบสคอนเน็คทวิตี (Java Database Connectivity : JDBC)

ในภาษาโปรแกรมสมัยใหม่จะมีช่องทางสำหรับเชื่อมต่อกับด้าเบสอยู่ ในภาษาจาวาก็เช่นเดียวกัน ซึ่งสามารถทำได้โดยการใช้ เจดีบีซี (Java Database Connectivity) ซึ่งมีโมดูลการเชื่อมต่ออย่างง่าย ดังรูป 3-12



รูปที่ 3-8 แสดงโมดูลการเชื่อมต่อ เจดีบีซี อย่างง่าย

3.7.1 ลักษณะของ เจดีบีซี

- เป็นมิดเคิลแวร์ (middleware) ที่สร้างมาจากภาษาจาวา
- ทำให้ เจดีบีซี ทำงานได้หลายแพลตฟอร์ม เนื่องจากข้อกำหนด Write once, run anywhere
- เจดีบีซี เป็นรูปแบบในการติดต่อกับฐานข้อมูลมีการแบ่งวิธีการในการทำงานออกเป็น 4 Type คือ Type1, Type2, Type3 และ Type4
- เจดีบีซี ไดรเวอร์ คือ ไดรเวอร์ส่วนที่จะนำมาประกอบกับรูปแบบเจดีบีซีให้เข้าใช้งานข้อมูล เช่น Oracle เจดีบีซี ไดรเวอร์ คือ ไดรเวอร์ที่เข้าใจการเก็บข้อมูลของออรากิล

3.7.1.1 JDBC Type1 (JDBC – ODBC Bridge)

- คือ เจดีบีซี ที่ทำงานบน โอดีบีซี อีกที
- ช่วยให้นำเอาฐานข้อมูลเดิมที่ทำงานบนพื้นฐานของวินโดวส์มาใช้งานจาวา
- เพื่อหลีกเลี่ยงปัญหาจากความจำเป็นต้องปรับเปลี่ยนระบบทั้งระบบเป็นการปรับเปลี่ยนเพียงบางส่วนเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่เหมาะกับการนำไปใช้งานบนระบบงานที่มีขนาดใหญ่ๆ เนื่องจากทำให้เกิดความช้าในการทำงานและประสิทธิภาพในการทำงานที่ไม่ดี
- มีข้อมูลในส่วนโอเวอร์เฮดสูงเนื่องจากต้องมีส่วนในการติดต่อระหว่างเจดีบีซี และ โอดีบีซีเพิ่มเติม
- ไม่สนับสนุนความสามารถทั้งหมดของมาตรฐาน เจดีบีซี เนื่องจากข้อจำกัดของ โอดีบีซี

3.7.1.2 เจดีบีซี Type2 (Partial Java Driver)

- มีประสิทธิภาพในการทำงานดีกว่าประเภทแรกเมื่อเปรียบเทียบกัน
- คำสั่งในการติดต่อกับเซิร์ฟเวอร์จะเป็นคำสั่งที่ดีที่สุดสำหรับเซิร์ฟเวอร์นั้น ๆ โดยเฉพาะทำให้การทำงานโดยรวมดีกว่า
- ผู้ใช้โปรแกรมในส่วนไคลเอ็นต์ยังต้องการใช้ไดรเวอร์สำหรับเซิร์ฟเวอร์โดยเฉพาะ
- เมื่อมีการเปลี่ยนแปลงเซิร์ฟเวอร์เป็นตัวอื่น โปรแกรมในส่วนไคลเอ็นต์ต้องมีการเปลี่ยนแปลงและ คอมไพล์ใหม่เสมอ

3.7.1.3 JDBC Type3 (Pure Java Driver)

- มีประสิทธิภาพในการทำงานดีกว่าประเภทที่ 1 และ 2 เมื่อเปรียบเทียบกัน
- เหมาะสำหรับองค์กรที่เซิร์ฟเวอร์ทางด้านฐานข้อมูลที่หลากหลายรูปแบบ
- การทำงานของไคลเอ็นต์ไม่จำเป็นต้องติดตั้ง เจดีบีซี ไว้ในทุกตัว
- การติดตั้งและการดูแลระบบไคลเอ็นต์สามารถทำได้ง่ายและสะดวก
- ยังต้องการไดรเวอร์สำหรับแต่ละผลิตภัณฑ์เพื่อติดตั้งไว้ที่แอปพลิเคชันเซิร์ฟเวอร์

3.7.1.4 JDBC Type4 (Direct-to-DB/Native to protocol)

- มีประสิทธิภาพในการทำงานดีที่สุด
 - มีความยุ่งยากในการพัฒนาเพราะผู้พัฒนาต้องมีการเรียนรู้การทำงานของเซิร์ฟเวอร์มาก่อน
- สำหรับในโครงการนี้เราเลือกใช้ เจดีบีซี Type2 (Partial Java Driver) สำหรับการติดต่อกับอราเคิลดาต้าเบส ในการเลือกใช้ เจดีบีซี ไดรเวอร์นั้นจะขึ้นอยู่กับตัวดาต้าเบสแต่ละตัวว่าสนับสนุน เจดีบีซี ไดรเวอร์ชนิดใด

บทที่ 4

อีเจบีคอนเทนเนอร์ (EJB Container)

4.1 การทำงานของอีเจบีคอนเทนเนอร์ (อีเจบีคอนเทนเนอร์)

อีเจบีคอนเทนเนอร์นี้จะเป็นตัวควบคุมเอ็นเตอร์ไพรส์ขึ้นเองด้วย ออบเจกต์ของเอ็นเตอร์ไพรส์บีนจะรันอยู่ในอีเจบีคอนเทนเนอร์ นอกจากจะควบคุมขณะที่รันแล้ว ยังทำหน้าที่เป็นตัวให้บริการกระบวนการทำงานต่าง ๆ ด้วย อีเจบีคอนเทนเนอร์มีหน้าที่ให้บริการกับเอ็นเตอร์ไพรส์บีนในด้านต่าง ๆ คือ

- การบริหารการใช้งานทรัพยากรและอายุในการใช้งานบีนที่อยู่ในระบบ (Resource Management and Bean Life Cycle Management)
- การจัดการกับทรานส์แอ็กชัน (Transaction Management)
- การสนับสนุนในการสร้างความปลอดภัยให้กับระบบ (Security)
- ความสามารถในการคงสถานะเดิมของบีน (Persistence)
- ความสามารถในการเรียกใช้งานบีนแบบรีโมตแอ็สเสสลิบิลิตี้

4.1.1 การบริหารการใช้งานทรัพยากร และอายุในการใช้งานบีนที่อยู่ใน

อีเจบีคอนเทนเนอร์นั้นจะรับผิดชอบในการให้บริการด้านการบริหารทรัพยากรของระบบ (resource management) โดยทรัพยากร (resource) นั้นอาจจะเป็น เทรด, การติดต่อแบบซ็อกเก็ต, การติดต่อกับฐานข้อมูลซึ่งอาจจะถูกร้องเรียกโดยแอ็พริเคชันเซิร์ฟเวอร์และนอกจากการจัดการด้านรีซอร์สเมเนจเม้นท์แล้วอีเจบีคอนเทนเนอร์ยังรับผิดชอบในด้านการควบคุมวงจรและอายุในการใช้งานของคอมโพเน้นต์ที่เป็นแบบเอ็นเตอร์ไพรส์บีนซึ่งถูกดีพลอยไว้โดยการอินสแตนทิเอท, เดิสทรอยหรือการรียูส เช่น เมื่อไคลเอนต์ร้องขอบีนซึ่งไม่ได้อยู่ในหน่วยความจำในขณะนั้น ดังนั้นอีเจบีคอนเทนเนอร์ก็จะทำการสร้างอินสแตนซ์ตัวใหม่ขึ้นมาในหน่วยความจำเพื่อให้ไคลเอนต์สามารถใช้งานได้หรืออีกนัยหนึ่งในกรณีที่บีนนั้นมีอยู่ในหน่วยความจำอยู่แล้วอีเจบีคอนเทนเนอร์ก็ไม่ต้องทำการสร้างบีนนั้นขึ้นมาใหม่หรือในกรณีที่ถ้าในระบบนั้นเหลือหน่วยความจำอยู่น้อยแล้ว อีเจบีคอนเทนเนอร์นั้นก็จะทำการรีแอ็สไซท์บีนจากไคลเอนต์ตัวหนึ่งให้ไคลเอนต์อีกตัวหนึ่ง หรืออาจจะทำลายบีนตัวอื่นที่ไม่ค่อยได้มีการใช้งานโดยกระบวนการนี้เรียกว่า อินสแตนซ์พูลลิ่ง

4.2 อินสแตนซ์ พูลลิ่ง

4.2.1 การพูลลิ่งและรียูส ของคอนเทนเนอร์ในสเต็ทเลสเสซ์ชันบีน

เนื่องจากเม็ทรูดของอีเจบีครีเอท (EJB create()) ในสเต็ทเลสเสซ์ชันบีนนั้นไม่ต้องผ่านพารามิเตอร์ ทำให้ไคลเอนต์จึงไม่ต้องส่งคลิตีออลอินฟอร์เมชันที่บีนอินสแตนซ์ต้องใช้ในการสตาร์ทอัพ ดังนั้นอีเจบีคอนเทนเนอร์จึงใช้สิ่งนี้ให้เกิดประโยชน์ในการที่จะทำพรีครีเอ็ทอินสแตนซ์ของสเต็ทเลสเสซ์ชันบีนดังนั้นอีเจบีคอนเทนเนอร์จึงต้องสามารถทำการพูลสเต็ทเลสเสซ์ชันบีนอินสแตนซ์ก่อนที่จะไคลเอนต์จะทำการติดต่อมา เมื่อไคลเอนต์ทำการเรียกใช้เม็ทรูดใด ๆ แล้วคอนเทนเนอร์ก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถได้อินสแตนตันนั้นจากการพูลลิ่งและให้บริการกับไคลเอนต์นั้นและหลังจากนั้นก็ส่งค่าอินสแตนตันนั้นกลับไปยังบีนพูลนั้น วิธีการนี้ทำให้คอนเทนเนอร์สามารถกำหนดอินสแตนต์ของบีนให้กับไคลเอนต์ต่าง ๆ ได้อย่างไดนามิก

ผลของการทำงานแบบนี้ก็คือทุก ๆ ครั้งที่มีการร้องขอเข้ามาจากไคลเอนต์จะมีอินสแตนต์ของสเตทเลสเช็สชันบีนหลายตัวที่สามารถให้บริการต่อการร้องขอนั้นได้เนื่องจากสเตทเลสเช็สชันบีนนั้นจะให้บริการเมื่อที่ถูกร้องขอมาเพียง 1 การร้องขอเท่านั้น และก็จะป็นอิสระจากไคลเอนต์สเปคิฟีก หลังจากการเรียกเม็ทโอดนั้นนั่นคือ สเตทเลสเช็สชันบีนทุกตัวจะมีสถานะเหมือนกัน หลังจากการเรียกเม็ทโอดแต่ละครั้งนอกจากนี้มันจะไม่รู้ว่ามีเกิดเรียกเม็ทโอดขึ้นก่อนหน้านั้นหรือไม่ ดังนั้นคอนเทนเนอร์จึงสามารถทำการรีเอ็สไซ์บีนต่าง ๆ ให้กับการร้องขอของไคลเอนต์ได้ในต่อเม็ทโอดเลเวลซึ่งก็คือการทำคอนเทนเนอร์สเปคิฟีก

ข้อดีของการทำบีนอินสแตนต์พูลลิ่งก็คือบีนต่าง ๆ ที่ใช้ในการให้บริการจะมีจำนวนน้อยกว่าจำนวนของการติดต่อจากไคลเอนต์ซึ่งเกิดจากการทำงานของไคลเอนต์ แบบ “think time” เช่น การเกิดเน็ตเวิร์คแล็กหรืออิวเมนต์ชิชัน ในฝั่งของไคลเอนต์โดยในขณะที่ไคลเอนต์เกิดการ “think time” นี้ คอนเทนเนอร์ก็สามารถใช้บีนอินสแตนต์เพื่อให้เซิร์ฟเวอร์กับไคลเอนต์ตัวอื่นได้เพื่อเป็นการประหยัดชิสเต็มริชอร์สได้อีกทางหนึ่ง

บีนที่ใช้ในกระบวนการพูลลิ่ง (bean pool) นั้นจะมีขนาดแตกต่างกันไปขึ้นอยู่กับคอนเทนเนอร์ว่าจะสามารถเปลี่ยนขนาดพูลไซส์ได้หรือไม่เมื่อโหนดมีการเปลี่ยนแปลงอยู่เสมอ ๆ เช่น ถ้ามีไคลเอนต์ติดต่ออยู่กับการดีพลอยเม้นท์ในเวลากลางวันมากกว่าเวลากลางคืนกับคอนเทนเนอร์ก็จะมีขนาดของการพูลในเวลากลางวันมากกว่าเวลากลางคืนเช่นกัน วิธีการนี้จะทำให้ฟรีชิสเต็มริชอร์สสามารถถูกนำไปใช้ได้ในงานอื่น ๆ ในขณะที่มีการใช้งานระบบสูงได้

4.2.2 การพูลลิ่งของคอนเทนเนอร์ในสเตทฟูลเช็สชันบีน

สเตทฟูลเช็สชันบีนจำนวนมากที่ถูกใช้โดยไคลเอนต์หลาย ๆ ตัวที่มาขอการติดต่อนั้น แต่ละตัวจะมีการเก็บคอนเวอร์เซชันสเตท ในการทำงานที่มันทำกับไคลเอนต์นั้น ๆ และเนื่องจากอ็เจ็คคอนเทนเนอร์/ เซิร์ฟเวอร์ มีจำนวนริชอร์สจำกัด เช่น หน่วยความจำ , คาด้าเบสคอนเน็คชัน และ ช็อกเก็ต คอนเน็คชัน ดังนั้นถ้าคอนเวอร์เซชันสเตทนั้นมีขนาดใหญ่ก็จะทำให้คอนเทนเนอร์ / เซิร์ฟเวอร์ นั้นมีริชอร์สในการทำงานไม่เพียงพอ แต่สิ่งนี้จะไม่เป็นปัญหากับสเตทเลสเช็สชันบีน เพราะว่าคอนเทนเนอร์นั้นสามารถพูลบีนได้จำนวนจำกัดในการให้บริการกับไคลเอนต์หลายๆตัว

การพูลลิ่งในสเตทฟูลเช็สชันบีนนั้นจะต่างกับในสเตทเลสเช็สชันบีน คือเมื่อไคลเอนต์ทำการเรียกเม็ทโอดในบีนมันก็จะเริ่มคอนเวอร์เซชันกับบีนนั้นพร้อมทั้งทำการเก็บคอนเวอร์เซชันสเตทในบีนนั้นด้วยซึ่งสเตทนี้ก็จะยังคงอยู่สำหรับไคลเอนต์ตัวนั้นที่มันติดต่อยู่เพื่อในเม็ทโอดร้องขอครั้งต่อไป ดังนั้นกับคอนเทนเนอร์จึงไม่สามารถทำการพูลลิ่งบีนและเอ็สไซ์บีนให้กับไคลเอนต์ตัวอื่นๆ ที่มาร้องขออย่างไดนามิกได้เพราะแต่ละบีนจะเก็บสเตทของไคลเอนต์ที่มันทำงานอยู่เท่านั้น ดังนั้นเพื่อให้การทำงานแบบนี้เป็นไปได้จึงต้องใช้ริชอร์สอย่างประหยัดและเพิ่มประสิทธิภาพในการทำงานของระบบให้สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกครั้งที่มีการทำงานของแอฟริเคชั่นบนเครื่องคอมพิวเตอร์จะมีฟิสคัลแอสเซมบลีที่จำเป็นในการทำงานนั้นเป็นจำนวนจำกัดโอเพอร์เรตติงซิสเต็มจึงต้องหาวิธีในการจัดการเมื่อมีแอฟริเคชั่นหลาย ๆ ตัวเข้ามาจนถึงแม้ว่าแอฟริเคชั่นเหล่านั้นจะต้องการหน่วยความจำรวมมากกว่าฟิสคัลแอสเซมบลีที่มีอยู่เพื่อที่จะแก้ไขปัญหานี้จึงมีการใช้ฮาร์ดดิสมาเป็นแอ็กแทนซ์ฟิสคัลแอสเซมบลีซึ่งทำให้เกิดการขยายส่วนของเวอร์ชวลแอสเซมบลีที่ระบบมีอยู่เมื่อแอฟริเคชั่นอยู่ในภาวะโอเคอวแอสเซมบลีส่วนของแอฟริเคชั่นนี้จะถูกสเวปเอาต์ออกจากฟิสคัลแอสเซมบลีไปยังฮาร์ดดิสและเมื่อแอฟริเคชั่นกลับเข้ามาแอ็กทีฟอีกครั้ง ข้อมูลที่ต้องการใช้ก็จะถูกสเวปอินจากฮาร์ดดิสไปยังฟิสคัลแอสเซมบลีโดยการสเวปนี้จะเกิดขึ้นบ่อยเมื่อมีการสวิตช์ระหว่างแอฟริเคชั่น (เรียกว่า context switch)

โอเจบีคอนเทนเนอร์ใช้ประโยชน์จากหลักการนี้เพื่อจัดการกับการใช้รีซอร์สในสเตทฟูลเชสชันบนในการที่จะจำกัดจำนวนของสเตทฟูลเชสชันบนอินสแตนซ์ในหน่วยความจำนั้น คอนเทนเนอร์ สามารถที่จะสเวปเอาต์สเตทฟูลเชสชันบนโดยการเซฟคอนเวอร์เซชันสเตทลงฮาร์ดดิสหรือแหล่งเก็บข้อมูลอื่น ๆ เรียกว่า พาสติเวชัน หลังจากการพาสติเวชันของสเตทฟูลเชสชันบนจะทำให้รีซอร์ส เช่น หน่วยความจำสามารถถูกนำไปใช้ให้บริการกับไคลเอนต์ตัวอื่นได้และเมื่อไคลเอนต์เดิมทำการเรียกเม็ทรูดเข้ามาอีกครั้งสเตทที่ถูกนำไปเก็บไว้ก็จะถูกสเวปอินเข้ามาในบีนนั้นอีกครั้ง เรียกว่า แอคทีเวชัน

โอเจบีนั้นสนับสนุนกับการพูลลิ่งของสเตทฟูลเชสชันบนนั้นก็จะมีอินสแตนซ์เพียงจำนวนหนึ่งเท่านั้นที่มีอยู่ในหน่วยความจำ เมื่อมีการติดต่อกับไคลเอนต์จำนวนมากแต่การพูลลิ่งที่นำเทคนิคพาสติเวชัน / แอคทีเวชัน มาใช้นี้อาจทำให้เกิดบอททลเน็คได้ซึ่งไม่เกิดในสเตทเลสเชสชันบน

การตัดสินใจว่าบีนไหนที่จะทำการแอ็กทีเวชันหรือพาสติเวชันนั้นจะขึ้นอยู่กับคอนเทนเนอร์ โดยคอนเทนเนอร์ส่วนมากนั้นจะใช้หลักการของเลสรีเซทริบูต (LRU) นั่นก็จะทำการพาสติเวทบีนที่มีการเรียกน้อยที่สุดวิธีนี้เหมาะในกรณีของรีโมตไคลเอนต์ที่มักเกิดการยกเลิกการติดต่อจากระบบเครือข่ายทำให้บีนนั้นตกค้างอยู่โดยไม่มีไคลเอนต์จึงควรทำการพาสติเวทออกไป โดยการพาสติเวชันนั้นสามารถเกิดได้ตลอดเวลาเมื่อบีนนั้นไม่เกี่ยวข้องกับเม็ทรูดคอลลแล้วโดยการตัดสินใจจะขึ้นอยู่กับคอนเทนเนอร์แต่มีข้อยกเว้นคือบีนที่เกี่ยวข้องกับทรานแซกชันทุกตัวจะ ไม่สามารถทำการพาสติเวทได้จนกระทั่งทรานแซกชันนั้นเสร็จสมบูรณ์

การแอ็กทีเวชันนั้นคอนเทนเนอร์ส่วนมากจะใช้วิธีการแบบจัสอินไทม (just – in – time) นั่นคือบีนต่าง ๆ จะถูกแอ็กทีเวทขึ้นมาตามความต้องการในขณะนั้น เช่นเมื่อมีการร้องขอจากไคลเอนต์เข้ามาซึ่งถ้าในขณะนั้นสเตทของบีนถูกพาสติเวชันอยู่คอนเทนเนอร์ก็จะทำการแอ็กทีเวทบีนนั้นเข้ามาสู่หน่วยความจำเพื่อใช้งานต่อไป

ปกติแล้วพาสติเวชันและแอ็กทีเวชันนั้นจะไม่เหมาะกับสเตทเลสเชสชันบนเพราะบีน ชนิดนี้ไม่มีสเตทดังนั้นสเตทเลสเชสชันบนจึงสามารถทำลายได้ง่ายโดยคอนเทนเนอร์นอกจากนี้การพาสติเวชัน/แอ็กทีเวชันยังเหมาะสำหรับเ็นติตี้บีนอีกด้วย

4.2.3 การพูลลิ่งของคอนเทนเนอร์เ็นติตี้บีนอินสแตนซ์

ในการสร้างโอเจบีคอนเทนเนอร์ / เซิร์ฟเวอร์ ที่ดินนั้นจะต้องสามารถทำการอินสแตนทิเอทเ็นติตี้ได้โดยเ็นติตี้บีนนั้นคือข้อมูลที่เก็บอยู่ในฐานข้อมูลที่เกิดขึ้นเมื่อไคลเอนต์มีการติดต่อและดิสคอนเน็คเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นคือจะมีการสร้างและทำลายบินเพื่อให้บริการแก่ไคลเอนต์นั้นๆ ซึ่งวิธีแบบนี้จะไม่เหมาะสมในการสร้างแอพลิเคชันเซิร์ฟเวอร์เพราะการสร้างและทำลายของอ็อบเจกต์จะเป็นการสิ้นเปลืองมากโดยเฉพาะถ้าไคลเอนต์นั้นมีการร้องขอบ่อยๆ

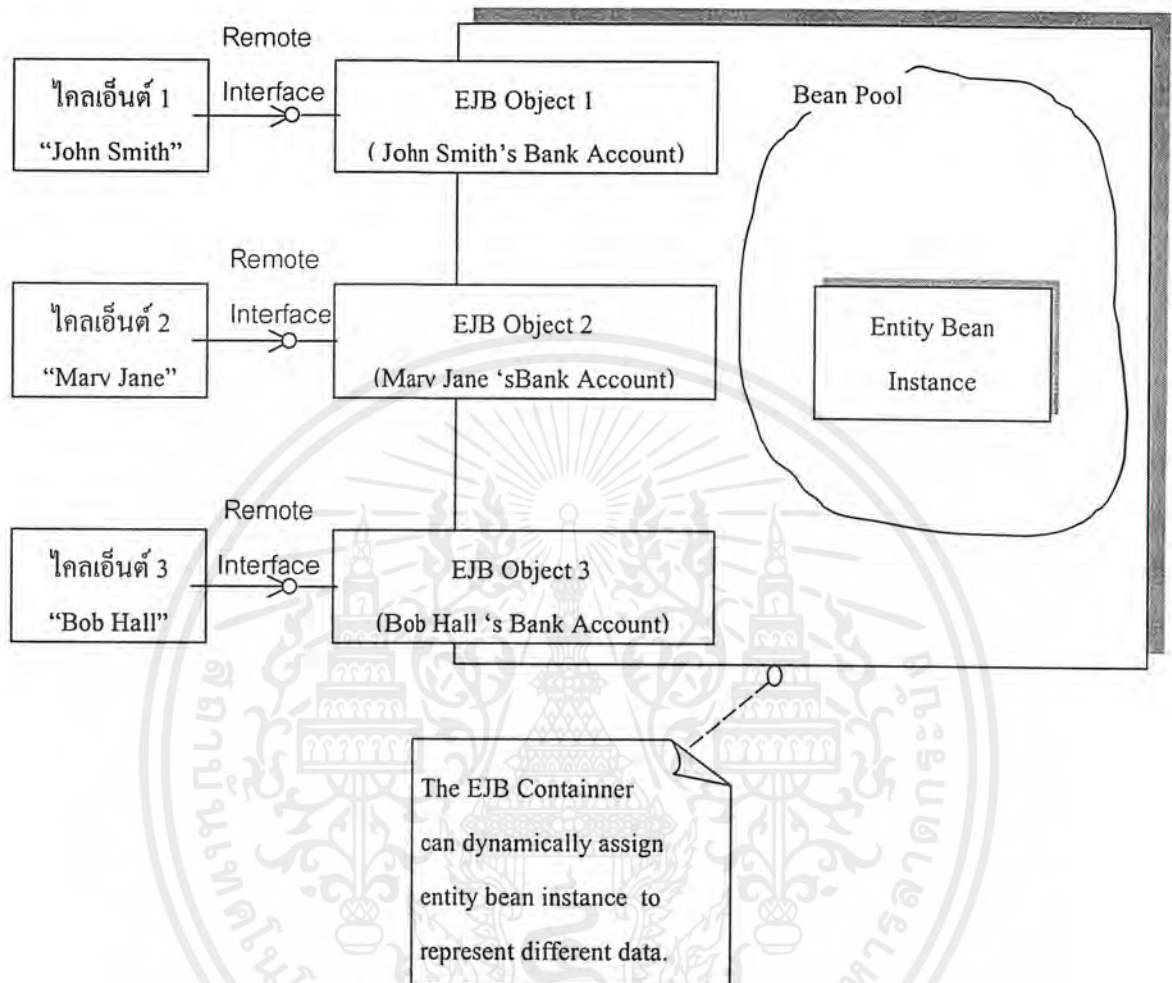
เอนิตีบีป็นคลาสนั้นจะอธิบายถึงฟิลด์และกฎเกณฑ์ต่างๆ สำหรับเอนิตีบีป็นนั้นแต่ไม่ได้ระบุข้อมูลที่แน่นอน เช่น เอนิตีบีป็นคลาสที่ระบุเบี่ยงค์แอ็คเคาท์แต่ละตัวก็จะมีฟิลด์ ต่าง ๆ คือ

- ชื่อของเจ้าของบัญชี
- แอ็คเคาท์ ID
- จำนวนเงินในบัญชีในขณะนั้น

ในบินคลาสนั้นจะแสดงถึงอินสแตนซ์ของข้อมูลในฐานะข้อมูลที่แตกต่างกันออกไป เช่นข้อมูลในเบี่ยงค์แอ็คเคาท์ใน 1 เร็คคอร์ด ก็จะมีฟิลด์ต่างๆ ที่มีข้อมูลต่างกันออกไปซึ่งในบินคลาสนั้นไม่ได้กำหนดข้อมูลที่ตายตัวไว้ภายใน ดังนั้นเพื่อที่จะลดเวลาในการอินสแตนซ์เทออ็อบเจกต์ใดๆนั้นอินสแตนซ์ของเอนิตีบีป็นจะสามารถนำกลับมาใช้ได้อีกครั้งโดยการพูลลิ่งซึ่งในการทำนั้นจะขึ้นอยู่กับคอนเทนเนอร์โพลีซี, คอนเทนเนอร์สามารถทำการพูลและรียูอินสแตนซ์เอนิตีบีป็นนั้น เพื่อที่จะแสดงถึงอินสแตนซ์ที่แตกต่างกันของข้อมูลชนิดเดียวกันที่อยู่ในที่เก็บข้อมูล เช่น คอนเทนเนอร์สามารถใช้เบี่ยงค์แอ็คเคาท์เอนิตีบีป็นอินสแตนซ์เพื่อที่จะแสดงถึงเบี่ยงค์แอ็คเคาท์เร็คคอร์ดต่างๆ ดังนั้นเมื่อมีการใช้เอนิตีบีป็นอินสแตนซ์, อินสแตนซ์นั้นอาจใช้ในการมอบหมายให้กับไคลเอนต์ต่างๆ ที่มีการร้องขอเข้ามาและอาจจะใช้แสดงถึงข้อมูลในรูปแบบต่างๆ โดยการทำงานของคอนเทนเนอร์นั้นจะเป็นแบบไดนามิค

วิธีนี้จะช่วยให้คอนเทนเนอร์ไม่จำเป็นต้องทำอินสแตนซ์เทอพีนอินสแตนซ์ทุกครั้งที่มีการร้องขอของไคลเอนต์ และยังช่วยลดการใช้รีซอร์สของระบบได้ ดังรูป 4-1

อีเจบี คอนเทนเนอร์ /



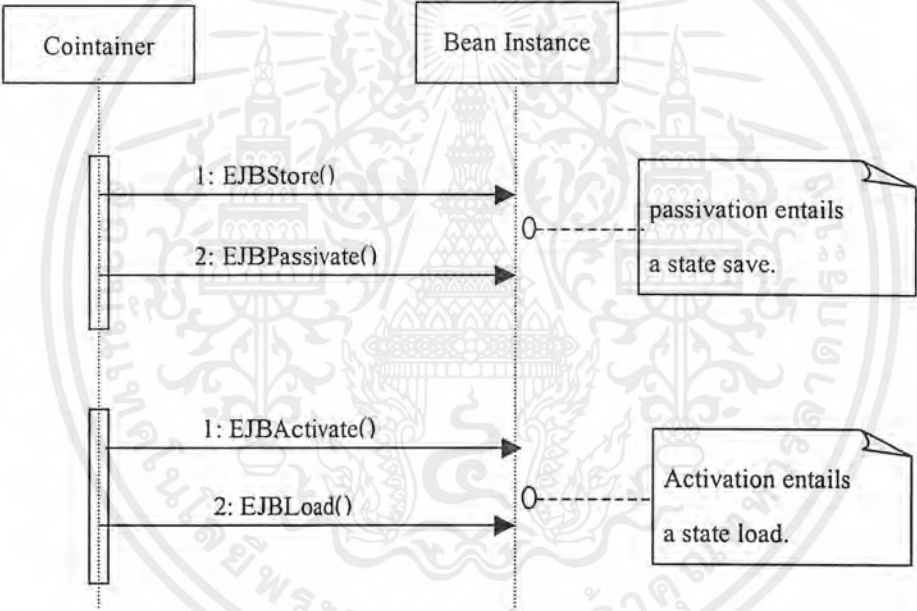
รูปที่ 4-1 EJB pooling of entity bean

การทำอินสแตนซ์พูลลิ่งนั้นคอนเทนเนอร์จะเป็นตัวจัดการซึ่งนอกจากจะใช้กับเอนตีตี้บีนแล้วยังสามารถใช้กับสเตตเลสเซสชันบีน แต่ถ้าใช้กับอีเจบีออบเจกต์เมื่อเอนตีตี้บีนถูกเอ็สไซส์ให้กับอีเจบีออบเจกต์ตัวหนึ่งแล้วมันอาจต้องใช้รีซอร์สบางอย่าง เช่น การติดต่อผ่านซ็อกเก็ตแต่เมื่อมันอยู่ในพูลมันก็จะไม่ต้องใช้ซ็อกเก็ต ดังนั้นเพื่อที่จะให้บีนสามารถเข้าครอบครองและปลดปล่อยรีซอร์สได้นั้นเอนตีตี้บีนคลาสจะต้องมีคอลแบ็คเม็ทโธด คือ

EJBactivate() : เป็นคอลแบ็คเม็ทโธดที่คอนเทนเนอร์จะทำการเรียกบีนอินสแตนซ์ เมื่อมีการเปลี่ยนบีนนั้นออกจากอินสแตนซ์พูล โดยกระบวนการนี้เรียกว่า แอคทิเวชัน และจะเป็นตัวบ่งชี้ว่าคอนเทนเนอร์ได้เข้าไปมีส่วนในการจัดการกับบีนด้วยสเปซิฟิกอีเจบีออบเจกต์และสเปซิฟิกไพรมารีตี้ โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EJBactivate() เมื่อโทรอด นั้นจะต้องใช้รีซอร์สต่างๆ เช่น ซ็อกเก็ตที่ต้องใช้เมื่อทำการมอบหมายไปยังพาดิกลาอีเจบีอ็อบเจ็ก EJBpassivate() เป็นคอลแบ็คเมื่อโทรอดที่คอนเทนเนอร์เรียกใช้เมื่อมีการเปลี่ยนบินเข้าไปยังอินสแตนซ์พูลเรียกกระบวนการนี้ว่า พาสสิเวชัน และจะเป็นตัวบ่งชี้ว่าคอนเทนเนอร์นั้นจะไม่มีส่วนเกี่ยวข้องกับบินจากสเปซิฟิกอีเจบีอ็อบเจ็กและสเปซิฟิไพรมารีตี้โดย EJBpassivate() เมื่อโทรอด นั้นจะปลดปล่อยรีซอร์ส เช่น ซ็อกเก็ตที่ได้มีการใช้ในภายหลังการทำงานของ EJBactivate()

เมื่อเอ็นติตี้บินอินสแตนซ์นั้นถูกพาสสิเวทแล้วนอกจากจะต้องปลดปล่อยรีซอร์สต่าง ๆ แล้ว ยังต้องเซฟเสตท์ของมันลงสตอร์เล็ทอีกด้วยทำให้สตอร์เล็ทนั้นถูกอัพเดทให้ถูกต้องตามเสตท์ในขณะนั้น และในการเซฟไฟว์ดของอินสแตนซ์นั้นลงฐานข้อมูล ,คอนเทนเนอร์จะทำการเรียก EJBStore() เมื่อโทรอด ก่อนที่จะทำการพาสสิเวชันในทำนองเดียวกันเมื่อเอ็นติตี้บินถูกแอคทิเวท นอกจากจะต้องการรีซอร์สแล้วยังจะต้องดึงข้อมูลที่ต้องใช้มาจากฐานข้อมูลด้วยโดยในการดึงข้อมูลไปยังบินอินสแตนซ์นั้น คอนเทนเนอร์จะเรียก EJB Load() เมื่อโทรอดหลังจากการทำแอ็คทิเวชัน ดังรูป 4-2



รูปที่ 4-2 เสตท์เซฟ และสเสตท์โหลดในการทำพาสสิเวชันและแอคทิเวชันของเอ็นติตี้บิน

จริง ๆ แล้วเอ็นติตี้บินจะค่อนข้างคล้ายกับเสตท์ฟูลเซสชันบิน โดยทั้ง 2 บิน นี้สามารถทำพาสสิเวท/แอคทิเวท แต่จะมีข้อแตกต่างอยู่คือเอ็นติตี้บินจะต้องเรียก EJBStore() คอลแบ็คในการเซฟเสตท์ในระหว่างการทำพาสสิเวท และ EJBLoad() สำหรับดึงข้อมูลในระหว่างการทำแอคทิเวชัน แต่ในเสตท์ฟูลเซสชันบินนั้นไม่ต้องใช้เพราะว่าคอนเทนเนอร์จะใช้อ็อบเจ็กซีเรียวไรเซเบิล (equivalent) ในการรักษาไฟว์ดต่าง ๆ ของเสตท์ฟูลเซสชันบินไว้ ซึ่งการทำแบบนี้จะมีความซับซ้อนน้อยกว่าแบบของเอ็นติตี้บิน.

4.3 ดาต้าเบสคอนเน็คชันพูลลิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อกับฐานข้อมูลนั้นจะมีความจำกัดในด้านเวลาและจำนวนการติดต่อ เพื่อแก้ไขปัญหานี้ จึงให้ คอนเทนเนอร์เป็นตัวจัดการพลของการติดต่อฐานข้อมูลโดยเอ็นเตอร์ไพรซ์บีนจะได้รับการติดต่อจากพล โดยเมื่อบีนหนึ่งยกเลิกการติดต่อแล้วบีนอื่นจะสามารถติดต่อแทนได้ทำให้ใช้งานฐานข้อมูลอย่างมีประสิทธิภาพ

4.4 สเกลาเบิลิตี้

อีเจบีเซิร์ฟเวอร์นั้นอาจจะสามารถวัดจำนวนรีซอร์สที่ใช้ไปในขณะใด ๆ ได้ โดยถ้าเราเพิ่มกลไกพิเศษที่มีความสามารถด้านต่าง ๆ ที่เท่ากันลงไป เช่น หน่วยความจำ , ความสามารถของตัวประมวลผล, เนื้อที่บนดิสหรือแบนวิทของเครือข่ายกลไกที่เพิ่มเข้าไปนี้จะทำให้จำนวนของผู้ใช้งานที่ฝั่งเซิร์ฟเวอร์สามารถรองรับได้และจำนวนของทรานแซคชันที่ซิสเต็มสามารถจัดการได้ต่อวินาทีนั้นเพิ่มขึ้นอย่างเป็นเชิงเส้น

4.5 ความปลอดภัยให้กับระบบ

ในระบบของอีเจบีนั้นจะมีระบบรักษาความปลอดภัยที่ไคลเอนต์จะต้องผ่านก่อนที่จะขอใช้บริการจากอีเจบีได้ คือ

- ไคลเอนต์จะต้องถูกออกเห็นทิเคทโดยการออกเห็นทิเคทนั้นจะเป็นการเป็นการพิสูจน์ว่าไคลเอนต์นั้นเป็นคนที่ถูกระบุไว้ว่าตนเองนั้นเป็นบุคคลที่ถูกต้อง เช่น ไคลเอนต์อาจจะใส่ยูสเซอร์เนมและพาสเวิร์ดในบราวเซอร์เพื่อตรวจสอบกับโปรไฟล์ของไคลเอนต์ที่เก็บในดาต้าเบสหรือแอลดีเอพีเซิร์ฟเวอร์และเมื่อใดที่ไคลเอนต์ถูกออกเห็นทิเคทแล้วหลังจากนั้นมันจะมีซีเคียวริตี้ไอดีเอ็นตีตี้ (security idEntity) ไปตลอดเซสชันของมัน
- ไคลเอนต์จะต้องถูกออกโทไรซ์หลังจากที่ไคลเอนต์นั้นถูกออกเห็นทิเคทแล้วนอกจากนี้ยังจะต้องได้รับอนุญาตในการทำโอเปอเรชันต่างๆ เช่นในการสั่งซื้อสินค้าใครก็สามารถสั่งซื้อสินค้าได้แต่เฉพาะเจ้าของเท่านั้นที่จะสามารถเข้าไปตรวจสอบรายการสั่งซื้อสินค้าได้

บทที่ 5

เอ็นเตอร์ไพซ์จาваบีน (Enterprise Java Bean)

เอ็นเตอร์ไพซ์จาваบีนใช้หลักการ divide-and-conquer ในการแบ่งความรับผิดชอบในการสร้างระบบทั้งหมดออกเป็นส่วน ๆ โดยแบ่งเป็น 6 ส่วนโดยทั้ง 6 ส่วนนี้รับผิดชอบโดยผู้ที่มีความเชี่ยวชาญในด้านของตนเองซึ่งการทำงานร่วมกันของฝ่ายต่าง ๆ ทั้ง 6 ฝ่ายซึ่งจะเป็นกุญแจสำคัญที่จะนำไปสู่ความสำเร็จของ เอ็นเตอร์ไพซ์จาваบีน ดีพลอยเมนต์ เพราะความเป็นมืออาชีพของทุก ๆ ฝ่าย เวลาที่ใช้ในการสร้าง เอ็นเตอร์ไพซ์-class ดีพลอยเมนต์ จึงต่ำมากเมื่อเทียบกับโดยทั่ว ๆ ไป โดยสามารถดูความสัมพันธ์ในการทำงานร่วมกันของส่วนต่าง ๆ ทั้ง 6 ส่วนได้ดังรูป 5-1

1. บีนโพรไวเดอร์เป็นผู้สร้างเรียสเอบิลิซึเนสคอมโพเนนท์ที่สามารถหาซื้อและนำไปใช้ในธุรกิจของตนได้ ซึ่งในความเป็นจริงแล้วบีนไม่ใช่แอปพลิเคชันโดยสมบูรณ์ในตัวเอง แต่เป็นดีพลอยเอเบิลคอมโพเนนท์ที่สามารถนำไปประกอบกันเป็นแอปพลิเคชันที่สมบูรณ์พร้อมใช้ในธุรกิจต่าง ๆ ตัวอย่างของบีนโพรไวเดอร์ในปัจจุบัน เช่น บริษัทเอ็นเตอร์ไพซ์ซอฟต์แวร์ซึ่งประกอบด้วยเอสเอพีและ Trilogy จะออกผลิตภัณฑ์ของตัวเองเป็นเอ็นเตอร์ไพซ์บีนหรือไม่ก็ออกคอนเน็กเตอร์ที่จะมาเชื่อมต่อกับซอฟต์แวร์ของตนเอง

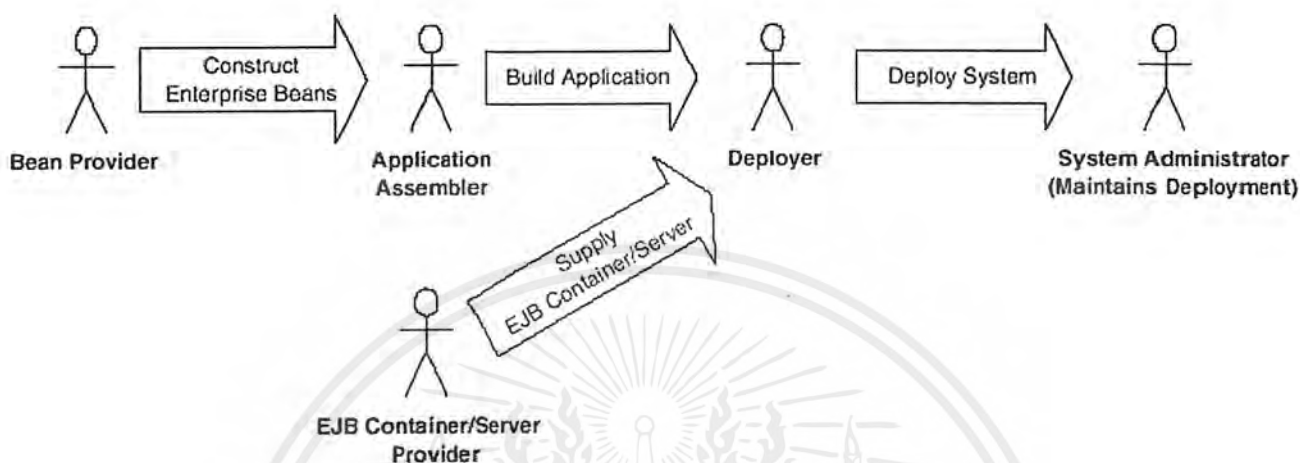
2. คอนเทนเนอร์โพรไวเดอร์เป็นผู้รับผิดชอบจัดเตรียมสภาวะแวดล้อมระดับต่ำที่เอเจบีแอปพลิเคชันต้องการในขณะทำงาน

3. เซิร์ฟเวอร์โพรไวเดอร์เป็นผู้รับผิดชอบแอปพลิเคชันเซิร์ฟเวอร์ที่จะเป็นที่บรรจุบริหารจัดการคอมโพเนนต์ต่าง ๆ โดยในขณะนี้ยังไม่มีข้อแตกต่างระหว่างเอเจบีคอนเทนเนอร์โพรไวเดอร์และเอเจบีเซิร์ฟเวอร์โพรไวเดอร์ ตัวอย่างของเอเจบีคอนเทนเนอร์/เซิร์ฟเวอร์ เช่น เว็บลอจิก จาก BEA Inc, NetDynamics จากซันไมโครซิสเต็ม, ออราเคิล 8i จากออราเคิล และอินโพรเซสแอปพลิเคชันเซิร์ฟเวอร์ จาก Inprise

4. แอปพลิเคชันแอสเซมเบลอร์เป็นผู้มีความรับผิดชอบที่จะนำคอมโพเนนต์ต่าง ๆ มาประกอบกัน และเขียนแอปพลิเคชันที่ใช้คอมโพเนนท์เหล่านั้นโดยที่อาจจะต้องเขียนบีนบางบีนขึ้นมาเอง ตัวอย่างของแอปพลิเคชันแอสเซมเบลอร์ก็เช่น ซิสเต็มอินทิเกรเตอร์หรืออาจจะเป็นโปรแกรมเมอร์ก็ได้

5. ผู้ดีพลอยเป็นผู้นำคอมโพเนนต์ที่ถูกเลือกจากแอปพลิเคชันแอสเซมเบลอร์มาใช้งานประกอบกับแอปพลิเคชันเซิร์ฟเวอร์แต่ละตัวที่มีอยู่เพื่อช่วยในการติดตั้งคอมโพเนนต์ให้สามารถทำงานในสภาวะแวดล้อมจริงได้

6. ซิสเต็มแอดมินิสเตรเตอร์เป็นผู้ดูแลให้ระบบงานทั้งหมดทำงานร่วมกันไปได้อย่างราบรื่น



รูปที่ 5-1 ความรับผิดชอบในการสร้างระบบที่รองรับการใช้งานของ เอนเตอร์ไพร์สจาวาบีน

5.1 ชนิดของบีนส์ (Type of Beans)

ใน อีเจบี 1.0 และ 1.1 นั้นได้กำหนดเอนเตอร์ไพร์สบีนไว้ 2 ชนิดซึ่งประกอบด้วยเซสชันบีนและเอนติตี้บีน

5.1.1 เซสชันบีน

เซสชันบีนแสดงงานที่ถูกกระทำจากไคลเอนต์ที่เรียกใช้มันเซสชันบีนคือออบเจกต์ที่ใช้ในการประมวลผลโดยการทำบิซิเนสลอจิก, บิซิเนสตรูต, และเวิร์คโฟล ตัวอย่างเช่น การแจ้งราคาสินค้า, สั่งสินค้า, การฝากถอนเงิน, และอีกมากมาย ดังนั้นมันจึงเป็นคอมโพเนนต์ที่บรรจุการทำงานต่าง ๆ ไว้ นอกจากนี้เซสชันบีนยังสามารถแบ่งออกเป็น 2 ประเภทซึ่งประกอบด้วยเซสชันบีนแบบสเตทฟูลและเซสชันบีนแบบสเตทเลสโดยจะกล่าวถึงในภายหลัง

เซสชันบีนถูกเรียกว่าเซสชันบีนเพราะว่ามันมีอายุเท่ากับเซสชันของไคลเอนต์ที่เรียกใช้มัน ยกตัวอย่างเช่น ถ้าส่วนไคลเอนต์ติดต่อกับเซสชันบีนเพื่อสั่งสินค้าซึ่งจะมีแอปพลิเคชันเซิร์ฟเวอร์หรือ (อีเจบี คอนเทนเนอร์/เซิร์ฟเวอร์) ในการรับผิดชอบการอินสแตนทิเอทเซสชันบีนคอมโพเนนต์นั้นขึ้นมาและเมื่อไคลเอนต์ยกเลิกการติดต่อตัวแอปพลิเคชันเซิร์ฟเวอร์ก็จะรับหน้าที่ทำลายอินสแตนทีนั้น

ณ เวลานั้นเซสชันบีนสามารถถูกเรียกใช้จากไคลเอนต์เพียงหนึ่งตัวเท่านั้น นั่นคือจะไม่มีการใช้เซสชันบีนร่วมกันระหว่างไคลเอนต์ต่าง ๆ ที่เข้ามาขอบริการ

อีเจบีเซิร์ฟเวอร์เป็นผู้รับผิดชอบอายุการใช้งานของบีนทุกบีนนั่นคือไคลเอนต์ไม่ได้ อินสแตนทิเอทบีนโดยตรงแต่อีเจบีคอนเทนเนอร์จะเป็นผู้ทำให้โดยอัตโนมัติและอีเจบีคอนเทนเนอร์จะเป็นผู้ทำลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชตขันธ์ขึ้นด้วยเมื่อถึงเวลาที่เหมาะสมจะทำให้เป็นถูกจัดสรรและนำกลับมาใช้ใหม่กับหลาย ๆ โกลเอ็นต์ที่เข้ามาได้

5.1.1.1 สเตทฟูลเชตขันธ์ขึ้น

บิซิเนสโพรเซสบางอย่างสามารถทำได้ในเมทรีอดเดียว เช่น การคำนวณราคาสินค้า, การตรวจสอบยอดบัตรเครดิต แต่ในงานบางอย่างต้องใช้หลายเมทรีอดร่วมกันและบางที่อาจเกี่ยวกับทรานแซกชันด้วย

ร้านค้าบนเว็บเป็นตัวอย่างหนึ่งของบิซิเนสโพรเซสที่ไม่สามารถทำงานให้เสร็จภายใน 1 เมทรีอดได้ลูกค้าเป็นผู้เลือกสินค้าใส่ในรถเข็นออนไลน์ซึ่งในการเพิ่มหรือลดสินค้าในรถเข็นนั้นจะต้องขึ้นอยู่กับสถานะปัจจุบันของรถเข็นของลูกค้าแต่ละคนด้วย

อีกตัวอย่างหนึ่งก็คืองานธนาคาร โดยในงานธนาคารนั้นเราอาจจะมีโค้ดที่ทำหน้าที่เป็นพนักงานธนาคารที่ติดต่อกับลูกค้าแต่ละคนในระยะเวลาหนึ่ง ๆ โดยที่โค้ดนั้นอาจจะทำทรานแซกชันให้กับลูกค้าแต่ละคน เช่น เช็คยอดเงินฝาก, โอนเงิน, ถอนเงิน

สเตทฟูลเชตขันธ์ขึ้นเป็นบิซิเนสที่ถูกรื้อออกแบบมาสำหรับให้บริการบิซิเนสโพรเซสที่ต้องทำหลาย ๆ เมทรีอดหรือหลาย ๆ ทรานแซกชันโดยที่สเตทฟูลเชตขันธ์ขึ้นจะคงสถานะของลูกค้าแต่ละคนไว้ตลอดเวลาที่ถูกใช้งาน

5.1.1.2 สเตทเลสเชตขันธ์ขึ้น

บิซิเนสโพรเซสบางบิซิเนสโพรเซสก็มีรูปแบบเป็นการขอบริการจากลูกค้าไปยังผู้ให้บริการเพียงครั้งเดียวและเนื่องจากมีการขอบริการเพียงครั้งเดียวจึงไม่จำเป็นจะต้องคงสถานะไว้เพื่อไว้ใช้ในการขอบริการครั้งต่อไป ดังนั้นสเตทเลสเชตขันธ์ขึ้นจึงเป็นคอมโพเนนต์ที่เหมาะสมกับการใช้งานในบิซิเนสโพรเซสประเภทนี้ โดยสเตทเลสเชตขันธ์ขึ้นนั้นจะให้บริการกับผู้ที่ร้องขอเข้ามาโดยไม่สนใจสถานะเดิมใด ๆ ทั้งสิ้น

ตัวอย่างของสเตทเลสเชตขันธ์ขึ้นอาจจะเป็นบิซิเนสที่ไว้ใช้ในการแก้ไขปัญหาทางคณิตศาสตร์ที่มีความซับซ้อนจากอินพุตที่รับเข้ามาเช่น การบีบอัดข้อมูลภาพและเสียงบิซิเนสจะรับข้อมูลดิบ และอัลกอริทึมบีบอัดเข้ามาทำการบีบอัดแล้วส่งผลลัพธ์ที่ได้กลับไปให้ผู้เรียกใช้ แล้วก็ไปให้บริการคนอื่นต่อบิซิเนสโพรเซสทั้งหมดอยู่ภายในเมทรีอดเดียวบิซิเนสไม่ต้องเก็บสถานะใด ๆ จากการเรียกใช้ครั้งที่แล้วไว้

ตัวอย่างอื่นก็เช่นสเตทเลสเชตขันธ์ขึ้นที่เป็นคอมโพเนนต์ที่ทำหน้าที่ตรวจสอบบัตรเครดิต โดยบิซิเนสจะรับเอาหมายเลขบัตรเครดิต, วันหมดอายุ, ชื่อผู้ใช้ และยอดเงิน เมื่อบิซิเนสตรวจสอบข้อมูลทั้งหมดแล้วก็จะส่งผลลัพธ์กลับไปให้ผู้เรียกใช้ แล้วก็ไปให้บริการแก่คนอื่นต่อโดยที่ไม่ต้องเก็บข้อมูลของลูกค้าคนที่แล้วไว้ในบิซิเนสเลย

5.1.2 เอ็นติตีบิซิเนส

ส่วนหลัก ๆ อีกส่วนหนึ่งที่ระบบงานส่วนใหญ่มีกันก็คือ ส่วนของข้อมูลทางธุรกิจที่จะถูกนำไปใช้ใน บิซิเนส โพรเซส ดังตัวอย่างเช่น

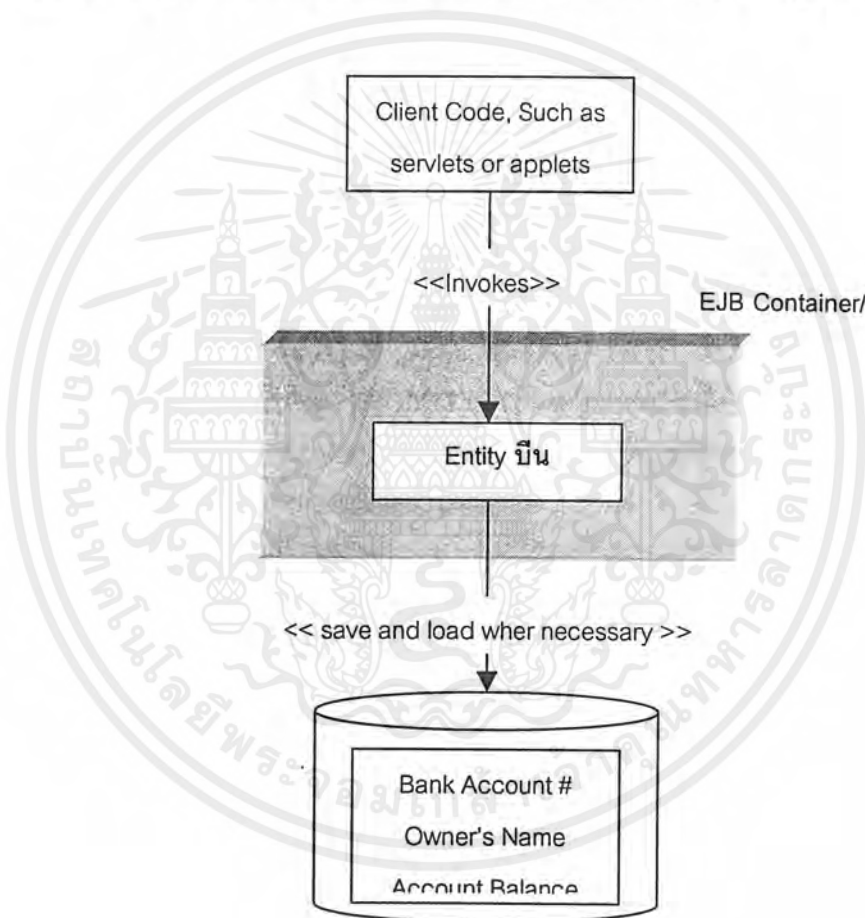
- คอมโพเนนต์ที่ทำหน้าที่พนักงานธนาคาร ทำการฝากเงิน, ถอนเงิน, โอนเงิน แต่ข้อมูลทั้งหมดเป็นข้อมูลที่มาจากบัญชีธนาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-คอมพิวเตอร์ที่ทำหน้าที่รับรายการสั่งซื้อสินค้าทำหน้าที่จับมีรายการสินค้าที่ถูกคำสั่งเช่น การสั่งซื้อเครื่องคอมพิวเตอร์ ข้อมูลที่ถูกสร้างขึ้นมาจากการรับสั่งสินค้าจะต้องประกอบด้วยจำนวนชิ้นส่วนต่างๆที่จะนำมาประกอบเป็นเครื่องคอมพิวเตอร์เพื่อที่จะส่งต่อไปให้คอมพิวเตอร์อื่นต่อไป

ในทุก ๆ สถานการณ์ที่กล่าวมานี้จะมีขั้นตอนที่หน้าที่เปลี่ยนแปลงข้อมูลที่อยู่ในดาต้าสตอเรจ เช่น ในฐานข้อมูลเอ็นติตี้บีน ก็คือคอมพิวเตอร์ที่เป็นตัวแทนของข้อมูลเหล่านั้น เช่น เอ็นติตี้บีนบัญชี, เอ็นติตี้บีน การสั่งซื้อสินค้าเอ็นติตี้บีนจะเป็นตัวแทนของ ดาต้าออบเจกต์ที่อยู่ในโลกของความต่าง

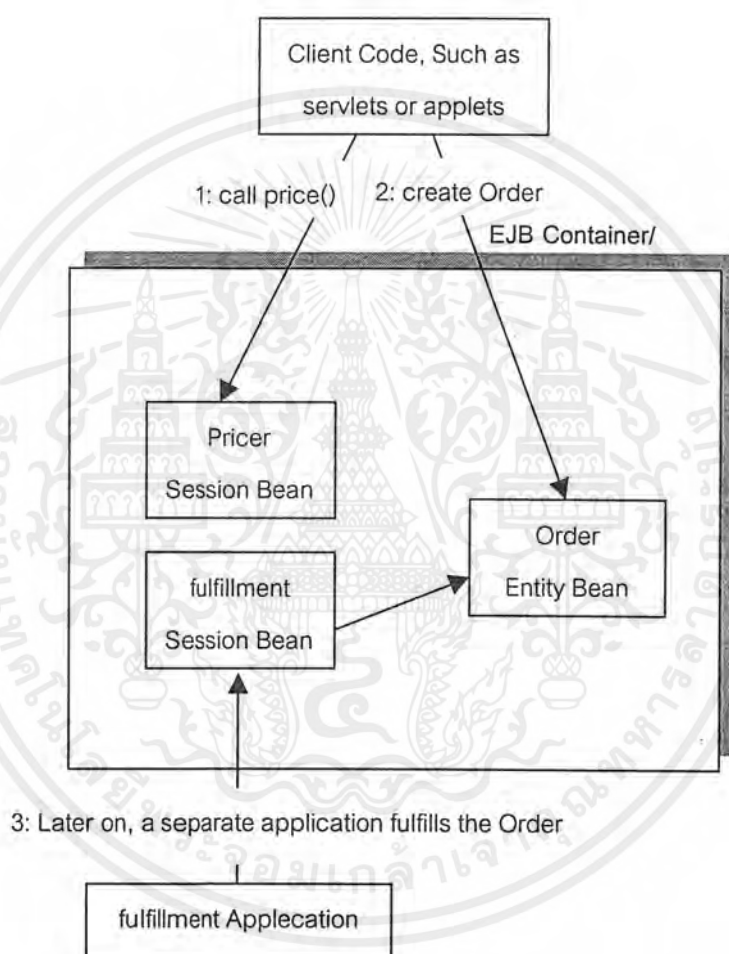
ในการสร้าง เอ็นติตี้บีนนั้นเราจะไม่ได้เขียนสโพรเซสไว้ในเอ็นติตี้บีน เนื่องจากมันเหมาะสมที่จะเอาไว้ใส่ข้อมูลมากกว่าส่วนตัวเซชันบีนนั้นจะเป็นตัวบรรจุสโพรเซสไว้ แล้วอาจจะมีการใช้เอ็นติตี้บีนเป็นตัวแทนข้อมูลที่มันติดต่อกับ เหมือนกับพนักงานธนาคารใช้ข้อมูลจากบัญชีธนาคาร



รูปที่ 5-2 เอ็นติตี้บีน จะใช้ดาต้าแอคเซสสโพรเซสในสถาปัตยกรรมมัลติ-tier

ประโยชน์ของเอ็นติตี้บีนก็คือ ทำให้เราสามารถมองข้อมูลที่อยู่ในดาต้าสตอร์เป็นข้อมูลแบบอ็อบเจกต์-ออร์เร็นเตด (ในเวลาที่มีถูกโหลดขึ้นหน่วยความจำ) ซึ่งถ้าเป็นเมื่อก่อนขณะที่แอปพลิเคชันจะทำการเรียกใช้ข้อมูลก็ต้องลงไปยุ่งกับตารางในฐานข้อมูลในการสั่งให้อ่านข้อมูล, เขียนข้อมูลแต่เอ็นติตี้บีนทำให้เราเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถมองข้อมูลทั้งหมดเป็นออบเจกต์เราสามารถทำงานกับข้อมูลที่อยู่ในฐานข้อมูลได้เหมือนเป็นออบเจกต์ตัวอย่างเช่น เราสามารถอ่านข้อมูลบัญชีที่ต้องการออกมาจากฐานข้อมูลแล้วนำไปใส่ไว้ในเอนิตีบีบีเราสามารถเรียกใช้เมทอด `withdraw` (ถอนเงิน) เพื่อที่จะลดค่าของตัวแปรบาลาสซึ่งเป็นชนิด `private` ที่อยู่ภายในเอนิตีบีบีได้โดยที่เอนิตีบีบีจะมีคุณสมบัติเพอร์ซิสเตนท์ ก็รับประกันว่าข้อมูลในฐานข้อมูลจะถูกเปลี่ยนแปลงแล้ว แน่ ๆ และจะอยู่ตลอดไปดังนั้นจึงเป็นการสะดวกต่อนักพัฒนาอย่างมากที่จะได้คุณสมบัติ เพอร์ซิสเตนท์ และเอนแคปซูเลชันในขณะเดียวกันเอนิตีบีบีจะใช้ดาต้าแอกเซสลอจิก เลเยอร์ในสถาปัตยกรรมมัลติเทียร์



รูปที่ 5-3 แสดงการใช้งานร่วมกันของเซสชันบีบี และ เอนิตีบีบี ใน Pricing แอปพลิเคชัน

ความแตกต่างอย่างชัดเจนของเอนิตีบีบีกับเซสชันบีบีอีกอย่างก็คือ เอนิตีบีบีสามารถให้บริการไคลเอนต์หลาย ๆ ไคลเอนต์ได้พร้อมกันดังเช่นการที่ไคลเอนต์หลาย ๆ ไคลเอนต์เข้าถึงข้อมูลในฐานข้อมูลพร้อมกันโดยที่เราสามารถจะทำให้ไคลเอนต์แต่ละไคลเอนต์ทำงานได้โดยเหมือนกับทำงานอยู่เพียงคนเดียวโดยใช้ทรานแซกชัน

ในความเป็นจริงแล้วเราอาจจะมีข้อมูลในฐานข้อมูลอยู่ก่อนแล้วก็ได้ซึ่งจะเป็นประโยชน์มาก เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับอีเจบี เพราะทำให้เราสามารถนำข้อมูลในฐานข้อมูลไปใช้เป็นออบเจกต์ในภาษาจาวา (Java) ได้ โดยที่การแปลงข้อมูลนี้อาจเกิดขึ้นในบิ้นหรือ อาจให้ อีเจบี คอนเทนเนอร์ ทำให้โดยอัตโนมัติก็ได้

เอ็นติตีบิ้นแบ่งย่อยได้เป็น 2 ประเภท คือ บิ้น-เมเนจ เพอร์ซิสแตนท์ และ คอนเทนเนอร์-เมเนจ เพอร์ซิสแตนท์ โดยมีรายละเอียดดังนี้

5.1.2.1 บิ้น-เมเนจ เพอร์ซิสแตนท์ เอ็นติตี บิ้น

เอ็นติตีบิ้น เป็นคอมโพเนนต์ที่มีคุณสมบัติเพอร์ซิสแตนท์ เพราะว่าสถานะของบิ้นจะถูกบันทึกลงใน secondary storage เช่น ฐานข้อมูลแบบรีเลชันนอลตัวอย่างเช่น เมื่อทำการอ็อบเจกต์-รีเลชันนอล แมปปิง (ping) แล้วเราจะสามารถแมปออบเจกต์ที่อยู่ในหน่วยความจำ (memory) ไปเก็บเป็นเรคคอร์ดในฐานข้อมูลได้ และในทางกลับกันเราสามารถนำเรคคอร์ดเหล่านั้นมาสร้างออบเจกต์นั้น ๆ ขึ้นมาใหม่ เราจะสามารถใช้ออบเจกต์-ดาต้าเบส (อ็อบเจกต์ดาต้าเบส) ได้ โดยจะทำให้เราสามารถเก็บออบเจกต์ลงในฐานข้อมูลจริง ๆ ได้แทนที่จะเป็นเรคคอร์ด

เอ็นติตีบิ้น แบบ บิ้น-เมเนจ เพอร์ซิสแตนท์ เป็น เอ็นติตีบิ้น ที่ผู้พัฒนาต้องทำการเก็บข้อมูลด้วยตนเอง หมายความว่าใน โค้ด ของบิ้นของเราจะต้องมีส่วนที่ทำการ แมปฟิลด์ต่าง ๆ ในออบเจกต์ไปเป็น ฟิลด์ต่าง ๆ ในเรคคอร์ดในฐานข้อมูลซึ่งจะต้องมีการเตรียมโค้ดสำหรับการอ่าน, การเขียน หรือการค้นหาข้อมูล โดยผ่านทางเพอร์ซิสแตนท์เอพีไอ เช่น เจดีบีซี หรือ เอสคิวแอล/เจ

5.1.2.2 คอนเทนเนอร์-เมเนจ เพอร์ซิสแตนท์ เอ็นติตีบิ้น

อีเจบีทำให้นักพัฒนาความสะดวกมากขึ้นโดยไม่ต้องพะวงกับการเขียนโค้ดที่เกี่ยวกับการ เพอร์ซิสของข้อมูลอีเจบี 1.1 จะทำให้เอ็นติตีบิ้น ของเราสามารถเพอร์ซิสได้โดยอัตโนมัติเมื่อเราต้องการ

โดยที่ คอนเทนเนอร์/เซิร์ฟเวอร์ จะทำการเขียน, การอ่าน หรือการค้นหาข้อมูลของคอมโพเนนต์ให้ทุกอย่าง ทำให้เราไม่ต้องเขียนโค้ดที่มีเอพีไอขอฐานข้อมูลแต่ละยี่ห้อที่จะทำให้ประหยัดเวลาในการเขียนบิ้นและทำให้บิ้นนั้น ไม่ยึดติดกับฐานข้อมูล เป็นการเพิ่มความริยูสเอเบิล เราเพียงแค่กำหนดว่าเราต้องการให้ฟิลด์ไหนเพอร์ซิสต่อ คอนเทนเนอร์เท่านั้นหลังจากนั้นคอนเทนเนอร์ก็จะจัดการให้โดยอัตโนมัติ

ในคลาตปัจจุบัน คอนเทนเนอร์/เซิร์ฟเวอร์ มีการ แมป หลากหลายรูปแบบและทุกที่ใช้ในการแมปก็มีหลายตัวบางตัวก็เป็นการแมปแบบง่าย ๆ บางตัวก็มีฟังก์ชันการแมปที่ซับซ้อน เช่น เว็บล็อกิเซิร์ฟเวอร์ จากบีอีเอซึ่งสามารถทำการแมป อ็อบเจกต์/รีเลชันนอล แบบง่าย ๆ ได้

5.2 คลัสเตอร์อีเจบี

อีเจบีทุกตัวนั้นสามารถรันได้กับทุกเซิร์ฟเวอร์ที่อยู่ภายในคลัสเตอร์ โดยถึงแม้ว่าอีเจบีถูกดีพลอยบนเซิร์ฟเวอร์หลาย ๆ ตัวแต่ละ อีเจบี นั้นก็สามารถอยู่บนเซิร์ฟเวอร์ได้ทั้งหมด

5.2.1 อีเจบี โฮม

โฮมของบิ้นทุก ๆ บิ้นก็สามารถรันได้ในทุกเซิร์ฟเวอร์เช่นกันโดยโฮมที่อยู่ในแต่ละเซิร์ฟเวอร์ ซึ่งอยู่ในคลัสเตอร์นั้นจะใช้ชื่อเดียวกันเมื่อไคลเอ็นต์ทำการลอคอัพโฮม , สตัป จะทำการอ้างอิงไปยังโฮมตัวใดตัวหนึ่งบนเซิร์ฟเวอร์ที่ได้ทำการดีพลอยบิ้นนั้นไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 สเตทเลส อีเจบี

เมื่อโฮมทำการสร้าง (ครีเอท) สเตทเลส บีน ขึ้นมาแล้ว โฮมจะรีเทิร์นสตັปที่สามารถใช้ได้กับทุก ๆ เซิร์ฟเวอร์ที่มีบีนนั้นคิปล้อยอยู่ได้เนื่องจากสเตทเลสบีนจะไม่เก็บสเตทของไคลเอนต์ไว้

สตັปของสเตทเลส อีเจบีไม่เป็นไอดีมโปเท้นท์ (idempotent) นอกจากเราจะต้องกำหนด ไอดีมโปเท้นท์เมทอด เอาไว้ใน คิปล้อยเมนต์เดสคริปเตอร์ไว้เอง มันจึงจะจัดการให้

5.2.3 สเตทฟูล อีเจบี

เมื่อโฮมทำการครีเอท สเตทฟูล บีน ขึ้นมันก็จะทำการรีเทิร์นสตັปกลับไปยังไคลเอนต์ที่เป็นคนเรียกนั้นเนื่องจากสเตทฟูล อีเจบี จะเก็บสเตทของไคลเอนต์ไว้ดังนั้นในทุก ๆ การคอล (call) ที่เข้ามาจะต้องทำการหาเส้นทางไปยัง อินสแตนท์ ที่อยู่บนเซิร์ฟเวอร์ตัวเดียวกันทุกครั้ง

ในกรณีของสเตทฟูลบีนการเกิดโหลด บาลาสซิ่งและการเฟล (fail) สามารถเกิดขึ้นได้เพียงขณะทำการคอลไปยังโฮม ซึ่งสำหรับในการร้องขอของบีนแบบสเตทฟูลนั้นจะต้องร้องขอไปยังเซิร์ฟเวอร์ตัวเดียวกันเท่านั้นถ้ามีการเฟลเกิดขึ้นไคลเอนต์จะต้อง catch remote exception เอาไว้ เมื่อต้องการติดต่อใหม่ต้องทำการครีเอทอินสแตนท์ตัวใหม่โดยใช้ โฮมที่มีอยู่เดิม

5.2.4 เอ็นติตี้ อีเจบี

เอ็นติตี้ ที่นิยมใช้มีอยู่ 2 ชนิดคือ read-write เอ็นติตี้ และ read-only เอ็นติตี้ เมื่อโฮมทำการค้นหาหรือครีเอท เอ็นติตี้ read-write ก็จะมี อินสแตนท์เกิดขึ้นและจะรีเทิร์น สตັปกลับไป การโหลด บาลาสซิ่ง และการเฟลจะเกิดขึ้นเพียงในขณะติดต่อกับโฮมเท่านั้นเนื่องจากว่าภายในคลัสเตอร์สามารถมีเอ็นติตี้ได้หลายตัวและแต่ละอินสแตนท์จะต้องทำการข้อมูลจากฐานข้อมูลก่อนที่จะมีการคอมมิท ทรานแซกชั่น เพราะจะทำให้ข้อมูลมีการเปลี่ยนแปลง

และเมื่อโฮมทำการค้นหาหรือ ครีเอท read-only เอ็นติตี้ บีน จะรีเทิร์นสตັปกลับไปสตັปนี้จะทำหน้าที่กระจายงานให้เซิร์ฟเวอร์แต่ละเท่า ๆ กัน และเมื่อมีการเฟลเกิดขึ้น read-only บีน จะทำการ cache เซิร์ฟเวอร์ทุกตัวไว้เพื่อป้องกันไม่ให้เกิดการอ่านข้อมูลจากฐานข้อมูล

บทที่ 6

จาวาเซิร์ฟเวอร์เพจ คอมโพเนนท์ (Java Server Pages Component)

6.1 จาวาเซิร์ฟเวอร์เพจ (Java Server Page : เจเอสพี)

เป็นเทคโนโลยีในการสร้างหน้าเว็บ (เว็บเพจ) โดยถูกออกแบบมาเพื่อให้ง่ายและเร็วต่อการสร้างแอปพลิเคชันที่เป็น เว็บ-เบส ที่สามารถทำงานได้กับหลากหลายเว็บเซิร์ฟเวอร์ (เว็บเซิร์ฟเวอร์), แอปพลิเคชันเซิร์ฟเวอร์ (แอปพลิเคชันเซิร์ฟเวอร์), บราวเซอร์ (Browser) หรือเครื่องมือในการพัฒนาอื่นๆ (Development Tools)

6.2 ประวัติการพัฒนาแอปพลิเคชันที่เป็น เว็บ-เบส

วิวัฒนาการของเว็บไซต์เว็บมีวิวัฒนาการมาจากการแสดงข้อมูลเพื่อการค้าหรือก็คือ "E-Commerce" แอปพลิเคชันที่สามารถใช้เบราว์เซอร์-เบสไคลเอ็นท์ที่มีข้อดีหลายอย่าง เช่น ไคลเอ็นท์/เซิร์ฟเวอร์-เบส แอปพลิเคชัน จะไม่จำกัดการเข้าถึงของไคลเอ็นต์, การจัดการและการเบสที่สามารถทำได้ง่ายขึ้นในการอัปเดต (update) แอปพลิเคชันผู้พัฒนาเพียงแค่แก้ไขโปรแกรมบนเซิร์ฟเวอร์โดยไม่จำเป็นต้องติดตั้งแอปพลิเคชันบนไคลเอ็นต์ใหม่ ซึ่งทำให้อุตสาหกรรมซอฟต์แวร์สามารถพัฒนาไปได้อย่างรวดเร็วในการสร้างแอปพลิเคชันที่เป็นหลาย-tier โดยใช้ เบิร์ดเซอร์-เบส ไคลเอ็นท์

เมื่อมีแอปพลิเคชันที่เป็นเว็บ-เบสเพิ่มมากขึ้นก็ต้องการการเปลี่ยนแปลงเทคโนโลยีในการพัฒนาเอชทีเอ็มแอล จึงเป็นทางเลือกในการแสดงรายละเอียดของเอกสาร จึงได้มีความพยายามในการพัฒนาให้หน้าของเว็บนั้นขึ้นอยู่กับความต้องการของผู้ใช้หรือสถานะของระบบ

ต่อมาจึงได้เกิดซีจีไอขึ้น โดยผู้พัฒนาจะเป็นผู้เขียนโปรแกรมขึ้นและเว็บ-เบสแอปพลิเคชันจะเรียกโปรแกรมผ่านเว็บเซิร์ฟเวอร์ โดยการทำเช่นนี้จะทำให้เกิดความยืดหยุ่นมากขึ้นคือในการเรียก ซีจีไอแต่ละตัวจะทำให้เกิดโปรเซสใหม่บนเซิร์ฟเวอร์ ถ้าผู้ใช้หลายคนเข้าถึงโปรแกรมพร้อมกันจะมีการแย่งทรัพยากร (Resource) กันทำให้การทำงานช้าลง

ในผู้ขายเว็บเซิร์ฟเวอร์แต่ละรายก็ได้มีความพยายามที่จะพัฒนาแอปพลิเคชันบนเว็บให้ทำได้ง่ายขึ้นโดยการสร้าง "Plug-ins" และเอพีไอขึ้นบนเซิร์ฟเวอร์แต่ปัญหาก็คือไม่สามารถใช้ผลิตภัณฑ์ข้ามระหว่างแต่ละผู้ขายได้ เช่น เทคโนโลยีแอ็กทีฟเซิร์ฟเวอร์เพจของไมโครซอฟท์ทำให้การสร้างหน้าเว็บทำได้ง่ายขึ้นแต่ต้องทำงานร่วมกับเพอร์ซันนอลเว็บ เซิร์ฟเวอร์ หรือ ไมโครซอฟท์ ลิส เท่านั้น

เทคโนโลยีที่เกิดขึ้นอีกก็คือจาวาเซิร์ฟเล็ท ทำให้การเขียนโค้ดบนฝั่งเซิร์ฟเวอร์สามารถทำได้ง่ายขึ้นสามารถติดต่อเซิร์ฟเล็ทได้โดยการสร้างแอปพลิเคชันด้วยภาษาจาวาติดต่อโดยจะเป็นโปรแกรมที่รันบนฝั่งเซิร์ฟเวอร์ตรงข้ามกับแอปพลิเคชันที่รันบนบราวเซอร์ ผู้พัฒนาสามารถเขียนเซิร์ฟเล็ทเพื่อรองรับเอชทีทีพีที่ส่งมาจากเว็บบราวเซอร์ได้ (อาจต้องคิวรีจากฐานข้อมูล) และเซิร์ฟเล็ทจะตอบสนองกลับไปโดยการส่งในรูปของเอกสารเอชทีเอ็มแอล หรือว่าเอ็กซเอ็มแอลกลับไปยังบราวเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อลำดับการทำงานเป็นดังที่กล่าวมาแล้ว หน้าของเว็บเพจทุกหน้าจะถูกประกอบขึ้นสมบูรณ์ภายใน เซิร์ฟเล็ต ถ้าต้องการที่แก้ไขโครงสร้างของหน้าเว็บเพจจะต้องไปอีดิท (edit) และคอมไพล์เซิร์ฟเล็ตใหม่ ดังนั้นคุณสมบัติที่สำคัญของ เซิร์ฟเล็ตในการสร้างหน้าเว็บเพจมีดังนี้

- เซิร์ฟเล็ตสามารถทำงานได้กับเว็บเซิร์ฟเวอร์และแอปพลิเคชันเซิร์ฟเวอร์หลายยี่ห้อ
- จะต้องแยกหน้าเว็บในส่วนแอปพลิเคชันลอจิกออกจากหน้าเว็บเพจที่เห็นได้
- โพรเซสสามารถติดต่อด้วยได้ง่าย เช่น เว็บ-เบสแอปพลิเคชัน

6.3 เทคโนโลยีของจาวาเซิร์ฟเวอร์ เพจที่เป็นการพัฒนาแอปพลิเคชันบนเว็บ

บริษัทชั้นร่วมกับผู้นำทางด้านเว็บเซิร์ฟเวอร์ (เว็บเซิร์ฟเวอร์), แอปพลิเคชันเซิร์ฟเวอร์ (แอปพลิเคชัน เซิร์ฟเวอร์) ได้ร่วมมือกันพัฒนาจาวาเว็บเซิร์ฟเวอร์ขึ้น โดยเทคโนโลยีของจาวาเซิร์ฟเวอร์เพจจะเป็นพัฒนาหน้าเว็บเพจในลักษณะที่เป็นไดนามิก คือมีลักษณะดังนี้คือ

1. แยกโปรแกรม (Content) ออกจากส่วนแสดงผล (Presentation)

ในเทคโนโลยีของเจเอสพีผู้พัฒนาจะใช้เทคเอชทีเอ็มแอล หรือ เอ็กชเอ็มแอล ในการออกแบบหน้าเว็บ (ส่วนฟรีเซชันเตชัน) และใช้เทคเจเอสพีหรือสคริปต์เล็กในการสร้างส่วนของโปรแกรม (Dynamic Content) คือ การแยกส่วนแสดงผลกับส่วนของโปรแกรมแยกออกจากกัน จะทำให้เราสามารถแสดงผลข้อมูลที่ต้องการได้โดยไม่ต้องสร้างหน้าเว็บใหม่ทั้งหมดเพราะจะเขียนโปรแกรมเป็นตัวประมวลผลและส่งไปแสดงผลในส่วนฟรีเซชันเตชันแทน โดยลอจิกต่างๆของโปรแกรมที่เราเขียนนั้นจะถูกซ่อน (Encapsulate) ไว้ในเทคและทั้งส่วนฟรีเซชันเตชันและส่วนโปรแกรมทั้งหมดนี้จะถูกรวมกันเรียกว่า “สคริปต์” (สคริปต์) สคริปต์ทั้งหมดนี้จะทำการประมวลผลที่ฝั่งเซิร์ฟเวอร์ และเว็บมาสเตอร์(เว็บmaster) จะสามารถทำการแก้ไขหน้าเว็บ (ส่วนฟรีเซชันเตชัน) ได้โดยจะไม่กระทบต่อส่วนของโปรแกรม

บนฝั่งเซิร์ฟเวอร์เจเอสพี เอนจิน (เจเอสพีเอนจิน) จะตีความเทคเจเอสพีและสคริปต์ เมื่อทำการประมวลผลตามลอจิกในโปรแกรมแล้วก็จะส่งเอาพุตที่กลับไปยังบราวเซอร์ในรูปของ เอชทีเอ็มแอล หรือ เอ็กชเอ็มแอล ในการทำงานเช่นนี้จะเป็นการป้องกันการเข้าถึงโค้ดและยังสามารถติดต่อไปยังเว็บบราวเซอร์ได้หลายตัวอีกด้วย

2. สามารถนำคอมโพเนนท์แต่ละส่วนกลับมาใช้ใหม่ได้

ความสามารถนี้เป็นความสามารถที่สำคัญของเจเอสพีและในการประมวลที่ซับซ้อนสามารถข้ามแพลตฟอร์มกันได้ ทำให้ในระบบที่มีผู้ใช้จำนวนมาก ผู้พัฒนาสามารถใช้คอมโพเนนท์ร่วมกันหรือว่าแลกเปลี่ยนคอมโพเนนท์กันได้

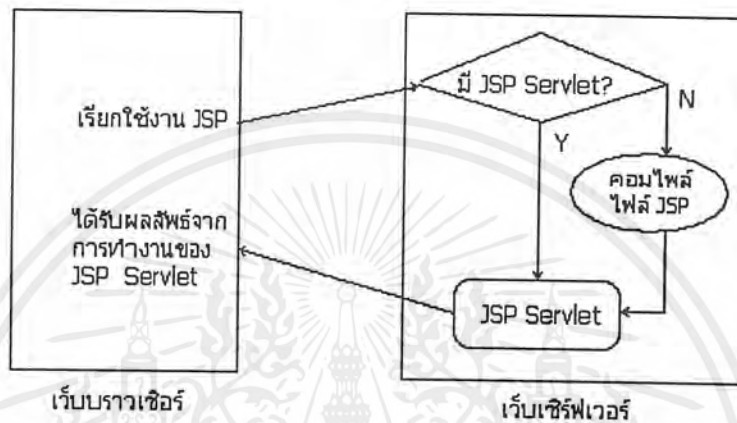
3. ทำให้การพัฒนาเว็บสามารถทำได้ง่ายขึ้น

เนื่องจากผู้พัฒนาเว็บส่วนใหญ่นั้นจะไม่ใช้โปรแกรมเมอร์ที่คุ้นเคยกับการใช้ภาษาสคริปต์มากนัก เจเอสพี จะสร้างฟังก์ชันที่จำเป็นในการใช้งานไว้พร้อมให้ใช้ ทำให้สามารถใช้งานได้ง่าย นอกจากนี้ เจเอสพี ยังอนุญาตให้ผู้ใช้สามารถสร้าง เทคไลบรารี (library) ขึ้นได้เอง และยังสนับสนุนการทำงานของแอปพลิเคชันในระดับเอนเทอร์ไพซ์ด้วย

เจเอสพี ได้รวมความสามารถต่างๆของเทคโนโลยีจาวาเอาไว้ได้แก่ มีการจัดการหน่วยความจำที่ดี, มีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยที่ดี เพราะภาษาสคริปต์จะมีพื้นฐานมาจากภาษาจาวาอยู่แล้วและ เจเอสพี นั้นเวลาคอมไพล์จะคอมไพล์ใน เซิร์ฟเล็ต และ เจเอสพี ยังมีลักษณะที่เรียกว่า Write Once Run Anywhere คือสร้างขึ้นเพียงครั้งเดียวแต่สามารถรันได้ในทุกที่ซึ่งเป็นคุณสมบัติของภาษาจาวาอีกด้วยโดยลักษณะของ เจเอสพี นั้นจะคล้ายกับ เอชทีเอ็มแอล หรือ เอ็ชเอ็มแอล

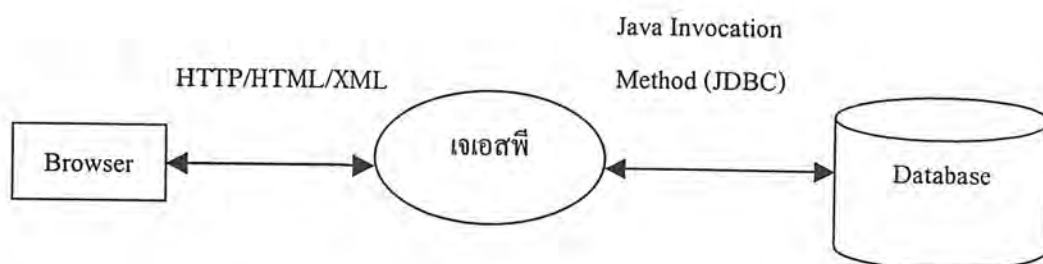
6.4 การทำงานของ เจเอสพี



รูปที่ 6-1 แสดงการทำงานของ เจเอสพี

เมื่อเว็บเบราว์เซอร์มีการร้องขอโดยผู้ใช้เพื่อเรียกข้อมูลเจเอสพีที่อยู่บนเซิร์ฟเวอร์ ซึ่งก็หมายถึงไฟล์ที่มีนามสกุลเจเอสพี เว็บเซิร์ฟเวอร์จะทำการตรวจสอบว่าไฟล์นั้นมีการคอมไพล์ไปเป็นเจเอสพีเซิร์ฟเล็ต แล้วหรือยังถ้ายังไฟล์เจเอสพีจะถูกคอมไพล์โดยตัวเว็บเซิร์ฟเวอร์ซึ่งมีเจเอสพีเอนจินไปเป็นเจเอสพีเซิร์ฟเล็ต แต่ถ้าไฟล์ถูกคอมไพล์แล้ว (ซึ่งเกิดขึ้นจากการใช้งานในครั้งแรก) เว็บเซิร์ฟเวอร์จะนำเจเอสพีเซิร์ฟเล็ตมาใช้งาน โดยการเรียกเพื่อประมวลผลได้ผลลัพธ์อย่างไรก็จะเป็นข้อมูล ซึ่งเว็บเซิร์ฟเวอร์ จะส่งกลับไปให้ผู้เรียกใช้งานและการประมวลผลของเจเอสพีเซิร์ฟเล็ตนี้เองที่ก่อให้เกิดข้อมูลเว็บในแบบไดนามิก

เบราว์เซอร์ (เบราว์เซอร์) จะเรียกใช้เจเอสพีซึ่งจะสร้างหน้าเว็บที่เบราว์เซอร์ต้องการ โดยเจเอสพีอาจจะต้องการร้องขอข้อมูลไปยังฐานข้อมูลโดยตรงเจเอสพีสามารถเรียกไปยังฐานข้อมูลโดยผ่าน เจดีบีซี หรือ จาวา Blend คอมโพเนนท์เพื่อนำผลลัพธ์มาจัดการแสดงผลเป็นเอชทีเอ็มแอล ส่งกลับไปให้เบราว์เซอร์



รูปที่ 6-2 แสดงการติดต่อกับฐานข้อมูลของ เจเอสพี

จากรูป 6-2 นี้เป็นการแทนที่แนวคิด (concept) ของ ซีจีไอ-บินด้วย เจเอสพี ซึ่งมีข้อได้เปรียบเหนือกว่าคือ ง่ายในการเขียนและใช้เวลาพัฒนาน้อย โดยโมเดลนี้ทำงานได้ดีกับหลาย ๆ แอปพลิเคชัน (แอปพลิเคชัน) แต่ก็ยังมีข้อจำกัดสำหรับงานที่มีไคลเอนต์เรียกใช้ทรัพยากรพร้อมกันเป็นจำนวนมากๆ เช่น ใช้ข้อมูลจากฐานข้อมูล จะทำให้ต้องสร้างแต่ละการเชื่อมต่อสำหรับแต่ละไคลเอนต์ทำให้เป็นการลดประสิทธิภาพการทำงานของฐานข้อมูลอย่างมาก

6.4.21 การทำงานของเจเอสพีร่วมกับเอนเตอร์ไพซ์จาวาบิน

เจเอสพีสามารถเป็นมิดเดิลเทียร์ (middle tier) ให้กับเอนเตอร์ไพซ์จาวาบินได้โดยเจเอสพีจะเป็นตัวแทนติดต่อ back-end resources ผ่านคอมโพเนนต์ของ เอนเตอร์ไพซ์จาวาบินได้ดังรูป 6-3



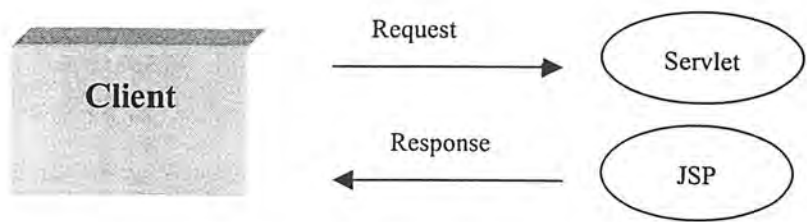
รูปที่ 6-3 การทำงานของ เจเอสพีร่วมกับ เอนเตอร์ไพซ์จาวาบิน

6.4.2 การทำงานของเจเอสพีร่วมกับจาวาเซิร์ฟเล็ต

เจเอสพี (จาวา เซิร์ฟเวอร์ เพจ) กับเซิร์ฟเล็ตนั้นจะมีความแตกต่างกันตรงวิธีการเขียน และการนำไปใช้เท่านั้นเอง ส่วนประสิทธิภาพของทั้งคู่จะมีความใกล้เคียงกันมาก สำหรับเจเอสพีเมื่อเราเขียนเสร็จมันจะเรียกเจเอสพีคอนเทนเนอร์ ซึ่งจะมีการนำเจเอสพีเอนจินมาใช้ในการคอมไพล์ให้เป็นแบบเซิร์ฟเล็ต (ซึ่งสุดท้ายเจเอสพีก็ต้องกลายเป็นเซิร์ฟเล็ตอยู่ดี) แต่สำหรับเซิร์ฟเล็ตเราต้องจัดการคอมไพล์เองโดยต้องมีเซิร์ฟเล็ตเอนจินมาใช้ในการคอมไพล์โดยโปรแกรมจะไม่ต้องถูกคอมไพล์ซ้ำเหมือน เจเอสพี

รายละเอียดในการทำงานร่วมกันนั้น ไคลเอนต์อาจจะขอบริการโดยตรงไปยังจาวาเซิร์ฟเล็ต ซึ่งจะเป็นตัวสร้างไดนามิก-คอนเท้นต์ (dynamic content) และให้ผลลัพธ์เป็นบีน (บีน) แล้วเรียกไปยังเจเอสพีซึ่งเจเอสพีจะนำไดนามิก-คอนเท้นต์ จากบีนนั้นมาจัดให้เหมาะสมแก่การแสดงผล (เป็น เอชทีเอ็มแอล) แล้วส่งไปยังเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-4 แสดงการร้องขอบริการจากไคลเอนต์ไปยังเซิร์ฟเล็ต (Servlet)

วิธีนี้เป็นการสร้างคอมโพเนนต์ (คอมโพเนนท์) ที่สามารถนำกลับมาใช้ได้หลาย ๆ แอปพลิเคชัน(แอปพลิเคชัน) ซึ่งอาจนำไปใช้เป็นส่วนหนึ่งของแอปพลิเคชันขนาดใหญ่ได้ และยังเป็นการทำให้สามารถจัดการกับทรัพยากรได้ดีขึ้นด้วย



บทที่ 7

การเชื่อมต่อกับฐานข้อมูล

7.1 การเชื่อมต่อกับฐานข้อมูล

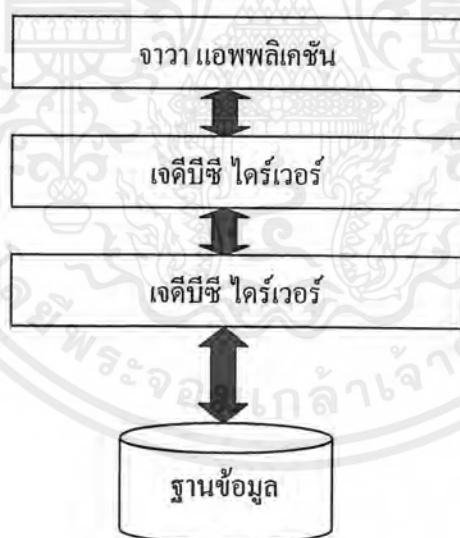
การเชื่อมต่อกับฐานข้อมูลด้วย เจดีบีซี จะมีรูปแบบหลักการทำงานดังนี้

Jdbc:<subprotocol>:<subname>

ในการเชื่อมต่อนั้นจะมีโปรโตคอลหลักในการทำงานเป็นเจดีบีซี ซึ่งมีการทำงานแบบเดียวในการเชื่อมต่อกับตัวจาวาแต่ละแตกต่างกันที่ชั้นโปรโตคอลซึ่งเป็นตัวบอกชนิดของการเชื่อมต่อกับตัวแม่ข่ายของฐานข้อมูลและชั้นเนมจะเป็นตัวบอกข้อมูลของการเข้าถึงฐานข้อมูลนั้น เช่น หมายเลขพอร์ต หรือชื่อของเครื่องแม่ข่าย สำหรับชั้นโปรโตคอลมีได้หลายแบบตามชนิดของการเชื่อมต่อ ซึ่งในโครงงานนี้จะใช้ ทินไดร์เวอร์ (thin driver) ในการเชื่อมต่อ

ทินไดร์เวอร์เป็นรูปแบบดั้งเดิมของการเชื่อมต่อด้วยเจดีบีซี เป็นการเชื่อมต่อที่ทั้งด้านเครื่องลูกข่ายและเครื่องแม่ข่ายทำการติดต่อกันผ่านทางโปรแกรมที่พัฒนาโดยจาวาทั้งหมด (Pure Java) โดยฐานข้อมูลส่วนใหญ่ที่สนับสนุน จาวาก็จะสนับสนุนการเชื่อมต่อแบบนี้ ตัวอย่างการขอเชื่อมต่อโดยใช้ทินไดร์เวอร์ เช่น

```
DriverManager.getConnection("jdbc:oracle:thin:@myhost:1521:orcl","scott","tiger");
```



รูปที่ 7-1 แสดงการเชื่อมต่อแบบ Thin-client

7.2 การจัดการกับทรานแซกชัน (Java Transaction)

ในคิสทรีบิวต์อ็อบเจ็กต์โอเปอเรชัน (Distributed object operations) นั้น การทำทรานแซกชัน (transaction) เป็นวิธีที่ปลอดภัยสำหรับการทำโอเปอเรชัน (operation) ที่เป็นแบบมัลติเพิลคอมโพเนนท์ สำหรับทรานแซกชันนั้นก็คือลำดับของคำสั่งซึ่งปรากฏในกระบวนการเอ็กซิกิวต์ ในการทำทรานแซกชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นจะยินยอมให้ผู้ให้หลายคนสามารถใช้ข้อมูลร่วมกันได้ และยังสามารถยืนยันได้ว่าข้อมูลเหล่านั้นจะสมบูรณ์ตามที่ได้แก้ไขไว้ทั้งหมดหรือไม่ โดยข้อมูลดังกล่าวนั้นจะไม่ถูกสอดคล้องด้วยข้อมูลของผู้ใช้คนอื่น ๆ เลย

การทำทรานแซกชันจะยืนยันได้ว่า การทำโอเปอเรชัน 2 โอเปอเรชันจะเป็นอิสระต่อกัน และจะสามารถป้องกันการเสียหายที่เกิดจากค่าเบสเสียหาย (Crash) ได้ ซึ่งหากไม่มีการทำทรานแซกชันแล้ว การจัดการฐานข้อมูลอาจจะไม่มีประสิทธิภาพได้

7.3 รูปแบบของทรานแซกชัน (Transaction Model)

วิธีการต่าง ๆ ในการทำทรานแซกชันแต่ละโมเดลนั้นจะมีความซับซ้อนและมีลักษณะของแต่ละโมเดล โดยในการทำทรานแซกชันสำหรับโมเดลที่นิยมจะมี 2 รูปแบบก็คือ

- แฟลตทรานแซกชัน (Flat Transaction)
- เนสทรานแซกชัน (Nested Transaction)

ในการใช้โมเดลดังกล่าวนี้ ทรานแซกชันเซอร์วิสต้องรองรับกับโมเดลแบบนั้นด้วย แต่ในปัจจุบันนี้ไม่ใช่ผู้ขายทั้งหมดที่จะกำหนดอ็อบเจกต์สเปกซิฟิเคชัน (EJB Specification) ให้สามารถใช้เนสทรานแซกชัน ได้ดังนั้นเอ็นเตอร์ไพร์สจาวาบีทีประกาศเป็นแฟลตทรานแซกชัน ก็สามารถไม่รองรับการทำเนสทรานแซกชัน ได้แต่อาจมีการเปลี่ยนแปลงในอนาคตตามความต้องการที่เพิ่มมากขึ้น

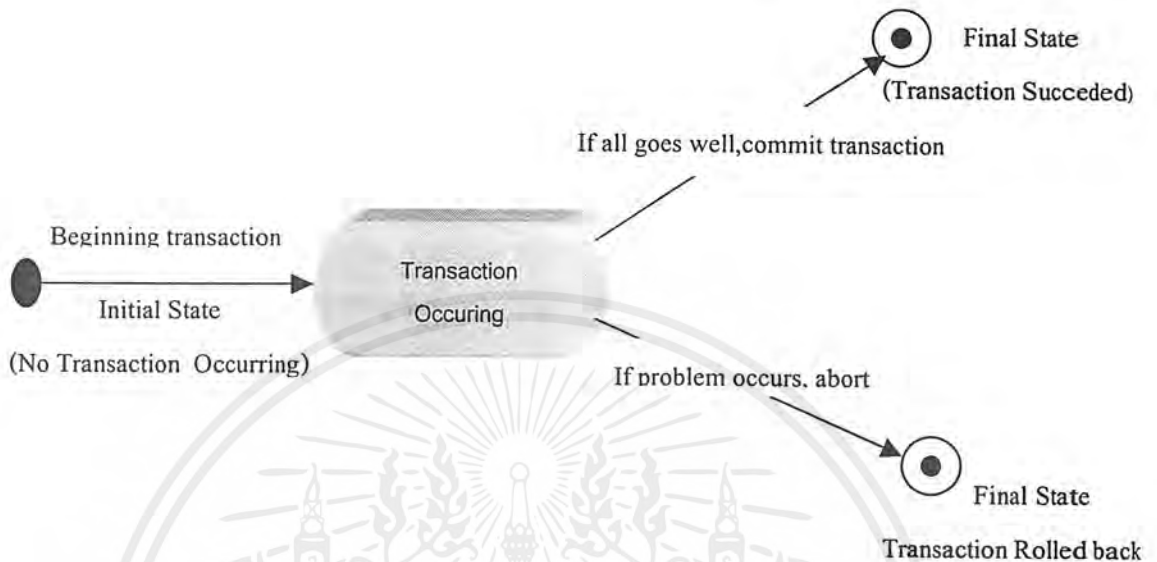
7.3.1 แฟลตทรานแซกชัน

แฟลตทรานแซกชันเป็นโมเดลของทรานแซกชันที่ง่ายต่อการทำความเข้าใจ โดยแฟลตทรานแซกชันก็คือ ลำดับของโอเปอเรชันที่ถูกจัดการอย่างอัตโนมัติแบบซิงเกิลยูนิตออฟเวิร์ก (Single unit of work) หลังจากเริ่มต้นการทำ

แฟลตทรานแซกชันแล้วแอปพลิเคชันก็จะเริ่มทำโอเปอเรชันนั้นๆ โดยบางโอเปอเรชันจะเป็นเพอร์ซิสแทนท์โอเปอเรชัน (persistent operations) และบางโอเปอเรชันก็ไม่ใช่ จนกระทั่งเมื่อจบการทำทรานแซกชันก็จะมีผลลัพธ์ออกมาคือ อาจจะสำเร็จหรือผิดพลาด โดยทรานแซกชันที่สำเร็จก็คือทรานแซกชันที่คอมมิต (committed) แล้วขณะที่ทรานแซกชัน ที่ผิดพลาดก็คือทรานแซกชันที่ Aborted

เมื่อทรานแซกชันนั้นคอมมิตแล้วการอัปเดตทั้งหมดที่อัปเดตไปยังรีซอร์ส (resource) ต่าง ๆ เช่น ค่าเบสก็จะถูกเก็บอย่างถาวรเพียงแค่ทรานแซกชันนั้นจบด้วยการคอมมิต แต่ถ้าทรานแซกชันนั้นออบอร์ท (aborted) ก็จะไม่กระทำการใด ๆ กับรีซอร์สที่ได้ทำการอัปเดตไป และก็จะทำการโรลแบค (rolled back) ในส่วนของเพอร์ซิสแทนท์โอเปอเรชันทั้งหมดที่แอปพลิเคชันจะต้องปฏิบัติ ก็จะไม่มีการทำอะไรทั้งสิ้น ซึ่งจะขึ้นอยู่กับระบบด้วย แอปพลิเคชันนั้นสามารถแจ้งกรณีของการเกิดออบอร์ท ได้ ดังนั้นแอปพลิเคชันจะไม่ทำการเปลี่ยนแปลงสิ่งใด ๆ ในหน่วยความจำขณะที่ทำทรานแซกชัน เรียกวิธีนี้ว่า all-or-nothing

“All-or-nothing” หากดูตามตัวอย่างของการฝากถอนเงินจากธนาคาร ก็จะหมายความว่าเราสามารถถอนเงินจากธนาคารหนึ่งได้และฝากในธนาคารอื่นได้โดยโอเปอเรชันเหล่านี้จะไม่เป็นทั้ง คอมมิตและโรว์แบค ซึ่งสามารถอธิบายกระบวนการของ แพลตฟอรมแซคชันได้ดังรูป 3-14



รูปที่ 7-2 แพลตฟอรมแซคชัน

7.3.1.1 การโรว์แบคของการทำทรานแซคชัน

ในกรณีที่เป็นการทำทรานแซคชันโมเดลซึ่งจะมีการรวบรวมโอเปอเรชันต่าง ๆ ไว้บน ฟิสิคัลคอลเพอร์มีเนนทรีซอร์ส (physical permanent resource) เช่น ดาต้าเบส โดยหลังจากเริ่มทำทรานแซคชันแล้วบีซิเนสคอมโพเนนท์ก็จะเริ่มต้องการการติดต่อไปยังดาต้าเบส ดังนั้นดาต้าเบสคอนเนคชัน (database connection) ก็จะถูกร้องขออย่างอัตโนมัติในทรานแซคชันที่คอมโพเนนต์นั้นเกี่ยวข้อง จากนั้นคอมโพเนนต์จะกระทำในส่วนของการเพอร์ซีสแทนท์โอเปอเรชัน (persistent operation) เช่น การอัปเดตข้อมูลในดาต้าเบสแต่ถ้าการจัดการรีซอร์สนั้นไม่สามารถทำได้ อาจจะต้องรอจนกระทั่งมีการคอมมิตก่อน ซึ่งจะมีการคอมมิตได้เมื่อบีซิเนสคอมโพเนนต์นั้น ๆ ได้มีการกระทำโอเปอเรชัน ต่าง ๆ ที่อยู่ภายใต้ทรานแซคชันเสร็จเรียบร้อยแล้วเท่านั้น

ในอีกประเด็นหนึ่งก็คือการที่บีซิเนสคอมโพเนนต์นั้น ตามธรรมดาแล้วจะไม่ทำการโรว์แบคเพอร์ซีสแทนท์สเตรท กล่าวคือเมื่อมีบอร์คเกิดขึ้นรีซอร์สนั้นก็จะไม่ถูกอัปเดตดังนั้นก็จะไม่มีการ อันดู (undo) ข้อมูลที่เป็นเพอร์มาเนนท์ดาต้า (permanent data) ในคอมโพเนนต์เหล่านั้น ซึ่งทั้งนี้ก็ขึ้นอยู่กับระบบด้วย โดยคอมโพเนนต์จะควบคุมการทำทรานแซคชันและสั่งให้ทรานแซคชันนั้นบอร์คแต่การทำการโรว์แบคกับเพอร์ซีสแทนท์สเตรทนั้นจะเกิดขึ้นโดยอัตโนมัติ ดังนั้นเมื่อบีซิเนสคอมโพเนนต์ทำโอเปอเรชัน

เรชันภายใต้ทรานแซกชัน แต่ละคอมโพเนนต์จะจัดการเพอร์ซิสแทนท์โอเปอเรชันทั้งหมด โดยถือว่าทรานแซกชันทั้งหมดจะเสร็จสิ้นอย่างเรียบร้อย

7.3.2 เนสทรานแซกชัน (Nested Transaction)

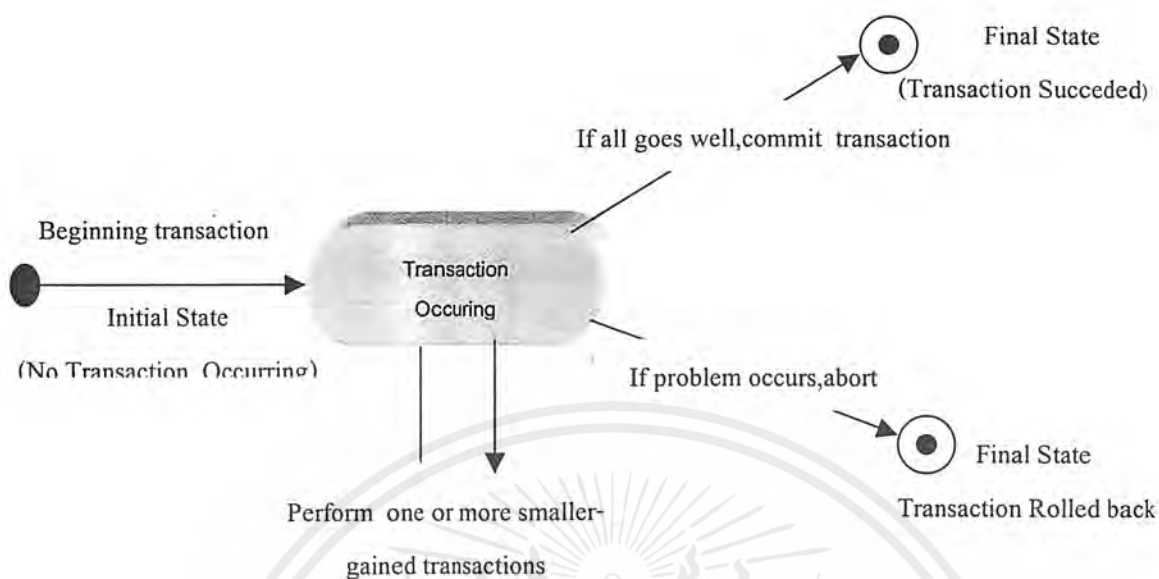
หากเราต้องการที่จะเขียนแอปพลิเคชันที่สามารถออกแบบเส้นทางท่องเที่ยวสำหรับใช้กับองค์กรหรือบริษัทที่เกี่ยวกับการท่องเที่ยว แอปพลิเคชันที่ต้องการนั้นต้องเขียนโค้ดที่ออกแบบเส้นทางรอบโลก และเป็นแอปพลิเคชันที่สามารถบริการการซื้อตั๋วที่ใช้ในการเดินทางได้ ซึ่งจะทำให้แอปพลิเคชันจะต้องมีโอเปอเรชันต่าง ๆ ดังนี้

1. แอปพลิเคชันนี้ต้องสามารถให้จองตั๋ว จาก Boston, USA ไป New York, USA
2. แอปพลิเคชันต้องสามารถจองตั๋วเครื่องบินจาก New York, USA ไป London England
3. แอปพลิเคชันต้องสามารถจองตั๋วบอลลูนจาก London, England ไปยัง Paris ประเทศฝรั่งเศส
4. แอปพลิเคชันต้องสามารถหาว่าเที่ยวบินไหนที่ไม่ได้ออกจากประเทศฝรั่งเศส

ในการจองตัวนั้นหากใช้แฟลตทรานแซกชันแล้วแอปพลิเคชันนั้นจะต้องการโอเปอเรชันเพื่อที่จะทำการโร้วแบคทรานแซกชัน เพราะว่าไม่มีเที่ยวบินใดที่ออกจากปารีสดังนั้นแอปพลิเคชันก็จะไม่มีข้อมูลการจองเที่ยวบินที่ไปปารีสแต่อาจจะมีการแทนการเดินทางเป็นอย่างอื่น เช่น รถไฟหรือเครื่องบิน ดังนั้นแฟลตทรานแซกชันจึงไม่เพียงพอ ดังนั้นเนสทรานแซกชัน สามารถที่จะแก้ปัญหาเหล่านี้ได้ โดยเนสทรานแซกชัน ขอมให้ทำการฝังอะตอมมิกยูนิตออฟเวิร์ค (atomic units of work) ลงในยูนิตออฟเวิร์คอื่น ๆ กล่าวคือยูนิตออฟเวิร์คจะถูกซ่อนอยู่ในยูนิตออฟเวิร์คอื่น ๆ ที่สามารถทำการโร้วแบคโดยไม่มีการบังคับทรานแซกชัน ทั้งหมดให้โร้วแบค

ดังนั้นยูนิตที่มีขนาดใหญ่กว่าจะพยายามทำส่วนของเอ็มเบ็ดยูนิตออฟเวิร์ค (Embedded unit of work) ซ้ำ ๆ ถ้าเอ็มเบ็ดยูนิตออฟเวิร์คนั้นเสร็จสมบูรณ์แล้วยูนิตที่มีขนาดใหญ่กว่าก็จะเสร็จด้วย แต่ถ้ายูนิตออฟเวิร์คไม่สามารถทำงานได้ มันก็จะบังคับให้ยูนิตนั้น ๆ เฟล (fail)

เราสามารถเปรียบเทียบเนสทรานแซกชันว่าเหมือนกับเป็นทรี (Tree) ของทรานแซกชันซึ่งจะมีการแตกย่อยลงมาจากรูท (root) หรือที่ออปัสเวลทรานแซกชันโดยรูททรานแซกชัน (root transaction) จะเป็นทรานแซกชันหลัก จากตัวอย่างของทริปแพลนนิ่งเอ็กแซมเพิล (trip-planning example) นั้นรูททรานแซกชัน เป็นโพเรสเซ็ทที่รวมเกี่ยวกับการจองตั๋วทั้งหมดทุก ๆ ทรานแซกชันในทรีทรานแซกชัน จะเรียกว่า ซับทรานแซกชัน ตามรูป 7-3



รูปที่ 7-3 The Nested Transactions

จุดที่พิเศษของเนสทรานแซกชันก็คือการที่ซัพทรานแซกชัน สามารถจะทำการโร้วแบคได้อย่างอิสระ โดยไม่มีผลกระทบต่อทรานแซกชันอื่น ๆ ที่อยู่ในระดับที่สูงกว่า ซึ่งนอกจากจะเป็นแนวความคิดที่ดีแล้วยังสามารถช่วยแก้ปัญหาที่กล่าวมาในข้างต้นได้กล่าวคือ ถ้าการจองตัวนั้นเป็น เนสทรานแซกชัน ก็จะสามารถทำการโร้วแบคการจองใด ๆ ก็ได้โดยไม่เกี่ยวข้องกับเงื่อนไขอื่น ๆ

บทที่ 8

การรักษาความปลอดภัยของระบบ

8.1 การรักษาความปลอดภัย

ลักษณะของการจัดการด้านการรักษาความปลอดภัยนั้นเพื่อสร้างการเชื่อมต่อการรักษาความปลอดภัยจากเว็บเบราว์เซอร์, จาวา ไคลเอ็นท์และอ็อบเจกต์เซิร์ฟเวอร์โดยจะมีลักษณะดังนี้

1. ขอบเขตในการรักษาความปลอดภัยนั้นจะมีการกำหนดเป็นกลุ่มต่างๆของผู้ใช้โดยมีจุดประสงค์เพื่อป้องกันทรัพยากรของเซิร์ฟเวอร์เราอาจไม่ได้ทำตามของเขตเหล่านี้ หรืออาจจะมีทางเลือกอื่น ๆ ก็เป็นไปได้ เช่น การใช้วินโดวส์เอ็นที (Window NT) , ยูนิก (UNIX), แอลดีเอพี ซีคิวริตี้สโตร์ (LDAP security store)
2. การอนุญาตการร้องขอเพื่อเข้าใช้ทรัพยากรของ เซิร์ฟเวอร์ ซึ่งการให้อนุญาตนี้สามารถทำได้โดยการใช้ยูสเซอร์เนมกับเพื่อให้อนุญาตกับไคลเอ็นท์
3. การใช้จาวาออเท้นทิเคชันและออเทอไรเซชันเซอร์วิส (JAAS) แอปพลิเคชัน โปรแกรมอินเตอร์เฟซ(API) สำหรับออเท้นทิเคชัน
4. ค่าความสมบูรณ์ (Data Integrity) และคอนฟิเดนทอลิตี้ (confidentially) ผ่านเอสเอสแอล โพรโทคอล ไคลเอ็นท์สามารถที่จะสร้างเอสเอสแอลเซสชัน ด้วยเว็บล็อกจิกเซิร์ฟเวอร์ โดยการใช้ไฮเปอร์เท็กซ์ทรานเฟอร์โพรโทคอล (HTTP) หรือ รีโมทเมทอดอินโวลูชัน (RMI) บนอินเตอร์โออาบี (Inter-ORB IIOP) .
5. การตรวจสอบอีเวนต์เช่น การพยายามเข้าถึงคอนหลังจากที่ล็อกอินไม่สำเร็จ (Failed login attempt), การร้องขอให้มีการรับรอง (authentication request)
6. การตรวจสอบการติดต่อเข้ามาของไคลเอ็นท์เพื่อการตอบรับหรือการปฏิเสธการร้องขอจากไคลเอ็นท์จากจุดเริ่มต้นซึ่งอาจจะเป็นโฮสต์เนมหรือเน็ตเวิร์กแอสเดส (network address) หรือ โพรโทคอลของไคลเอ็นท์

จากสถาปัตยกรรมของการรักษาความปลอดภัยนั้นส่วนที่อยู่ในขั้นแรกก็คือออเท้นทิเคชันซึ่งออเท้นทิเคชันจะป้องกันบุคคลที่จะเข้ามาติดต่อเซิร์ฟเวอร์เอ็นไวร์รอนเมนต์

ในการอเท้นทิเคชันจะเป็นการตรวจสอบว่าไคลเอ็นต์คือคนที่เขาอ้างว่าเขาเป็นหรือเปล่าและออเทอไรเซชัน ซึ่งคือการตรวจสอบผู้ที่ถูกอเท้นทิเคชันแล้วว่ามีความสิทธิ์ที่จะทำงานนั้น ๆ หรือไม่ซึ่งการอเท้นทิเคชันจะต้องถูกกระทำในเวลาใดเวลาหนึ่งก่อนที่จะเรียกใช้เมทอดของอ็อบเจกต์ ส่วนการออเทอไรเซชันจะถูกกระทำระหว่างการเรียกใช้เมทอดของอ็อบเจกต์โดยมีรายละเอียดดังนี้

ขั้นตอนที่ 1: การอเท้นทิเคชัน

อ็อบเจกต์ไม่ได้บังคับวิธีการอเท้นทิเคชันไว้ดังนั้นการที่จะทำการอเท้นทิเคชันจึงเป็นหน้าที่ของแอปพลิเคชันของผู้ใช้กับคอนเทนเนอร์ซึ่งหมายความว่าแต่ละคอนเทนเนอร์จะจัดการกับเรื่องนี้แตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันเช่นเว็บล็อกของบี (BEA's Weblogic) นั้นเราสามารถระบุชื่อเซิร์ฟเวอร์/พาสเวิร์ดในโค้ดของไคลเอ็นต์เมื่อมันใช้เจเอ็นดีโอเพื่อมองหาโฮมอ็อบเจ็กต์

```
Properties props = System.getProperties();

props.put(Context.SECURITY_PRINCIPAL, "EmployeeA");
props.put(context.SECURITY_CREDENTIALS, "myPassword1");

Context ctx = new InitialContext(props);
```

เนื่องจากอีเจบีสเปกซิฟิเคชันไม่ได้ระบุไว้ถึงวิธีการออกเท็นทิเคชันทำให้โค้ดเหล่านี้ไม่สามารถนำไปใช้บนแอปพลิเคชันเซิร์ฟเวอร์อื่นๆได้

เมื่อผู้ใช้งานโค้ดนี้แอปพลิเคชันเซิร์ฟเวอร์ จะแมปยูสเซอร์เนม/พาสเวิร์ดของผู้ใช้เข้ากับชื่อบุคคลไอดีเอ็นดีซี ขั้นตอนนี้ก็ขึ้นอยู่กับแต่ละแอปพลิเคชันเซิร์ฟเวอร์ เช่นกันบางแอปพลิเคชันเซิร์ฟเวอร์อนุญาตให้ผู้ใส่ยูสเซอร์เนม/พาสเวิร์ด ไว้ในพรอเพอร์ตี้ไฟล์ที่แอปพลิเคชันเซิร์ฟเวอร์จะอ่านตอนรันไทม์ บางเซิร์ฟเวอร์จะสนับสนุนระบบรักษาความปลอดภัยที่ขณะนี้ใช้กันอยู่แพร่หลายทั่วไป เช่น ยูสเซอร์เนม/พาสเวิร์ด ที่อยู่ในแอตตีเอพซีเซิร์ฟเวอร์

ขั้นตอนที่ 2 : การ ออกเทอร์ไรเซชัน

หลังจากไคลเอ็นต์ถูกออกเท็นทิเคทแล้ว มันจะต้องผ่านการออกเทอร์ไรเซชันเพื่อตรวจสอบสิทธิ์ในการเรียกใช้เมทอดของอีเจบี โดยจะมีแนวทางอยู่ 2 แนวทางในการออกเทอร์ไรเซชันคือดีเคเลทีฟลี (declaratively) และ โปรแกรมเมติกอลลี (programmatically)

ถ้าผู้ใช้เลือกการทำงานแบบดีเคเลทีฟลีออกเทอร์ไรเซชัน, คอนเทนเนอร์ จะจัดการตรวจสอบออกเทอร์ไรเซชันทั้งหมดให้เป็นวิธีที่สะดวกและง่ายดายช่วยให้ผู้ใช้สามารถเขียนบีนได้โดยไม่ต้องไปพะวงกับเรื่องความปลอดภัย

แต่ถ้าผู้ใช้เลือกการทำงานแบบโปรแกรมเมติกออกเทอร์ไรเซชัน, ผู้ใช้จะต้องเขียนการตรวจสอบความปลอดภัยไว้ในบีนของผู้ใช้เองทำให้บีนนั้นมีบิจินสล็อกกับส่วนการดูแลความปลอดภัยทำให้บีนมีขนาดใหญ่โตโดยไม่จำเป็น

8.2 ชีคิวรียูลส์ (Security Roles)

ออกเทอร์ไรเซชันในอีเจบีใช้คอนเซปต์ของชีคิวรียูลส์ซึ่งคือกลุ่มของไคลเอ็นท์ไอดีเอ็นดีซีในการที่ไคลเอ็นต์ทำโอเปอเรชันต่าง ๆ ไคลเอ็นต์จะถูกตรวจสอบว่าชีคิวรียูลส์ไอดีเอ็นดีซีของมันอยู่ในชีคิวรียูลส์ที่ถูกต้องรียูลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของการใช้สิทธิ์ที่รู้สึคือผู้ใช้ไม่ต้องระบุไอดีลงไปในตัวของผู้ใช้ นี่เป็นสิ่งที่จำเป็นในการพัฒนาขึ้นเพื่อที่จะไปตีผลอยู่ในหลาย ๆ ที่ซึ่งมีระบบความปลอดภัยที่ต่างกัน ทำให้เราสามารถเปลี่ยนแปลงสิทธิในการใช้เมทอดได้โดยไม่ต้องคอมไพล์บีนใหม่

8.3 โปรแกรมเมตริกคออเทอไรเซชัน (Programmatic Authorization)

ในการตรวจสอบออเทอไรเซชันด้วยบีนของผู้ใช้เองนั้นควรจะต้องกิวรีอ็อบเจกต์คอนเทกซ์เพื่อเรียกข้อมูลเกี่ยวกับไคลเอนต์ที่ติดค่ออยู่ในขณะนั้นซึ่งจะมีเมทอดที่เกี่ยวข้องอยู่ 2 เมทอด คือ `isCallerInRole()` และ `getCallerIdentity()`

● `isCallerInRole()`

`isCallerInRole(Identity role)` จะทำการตรวจสอบว่าผู้ที่เรียกใช้บีนอยู่ในโรลที่กำหนดหรือไม่ เมื่อผู้ใช้เรียกเมทอดนี้ผู้ใช้จะต้องส่งโรลที่ต้องการเปรียบเทียบกับ, ผู้ที่เรียกใช้บีนจะเป็นถึงความแตกต่างระหว่าง `isCallerInRole(Identity role)` และ declarative security คือ `isCallerInRole(Identity role)` จะไม่ตรวจสอบโรลที่ถูกระบุไว้ใน ดีพลอยเมนต์เดสคริปเตอร์แต่จะตรวจสอบกับโรลที่ระบุไว้ในโค้ดของผู้ใช้เท่านั้น

```
import java.security.Identity;
...
public class MyBean implements SessionBean {
    private SessionContext ctx;
    ....
    public void foo() {
        Identity id = new MyIdentity("administrators");
        If (ctx.isCallerInRole(id)) {
            System.out.println("An admin called me");
            return;
        }
        System.out.println("A non-admin called me");
    }
}
```

โค้ดข้างบนเป็นตัวอย่างในการทำงานต่างกันถ้าโรลของผู้เรียกใช้ต่างกัน เราจะต้องสร้างคลาส `MyIdentity` ซึ่งสืบทอดมาจาก `java.security.Identity` ซึ่งเป็นแอสเทรคคลาส (abstract class) ซึ่งตัวอย่างของคลาสสามารถแสดงได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import java.security.Identity;

public class MyIdentity extends Identity {

    public MyIdentity (String id) {
        super(id);
    }
}
```

● `getCallerIdentity()`

`getCallerIdentity()` จะส่งค่าชื่่วิธีไอดี้นิติของผู้ที่เรียกใช้ป็นในขณะนั้นกลับมา โดยสามารถที่จะ นำไอดี้นิตินั้นไปใช้ ทำงานต่าง ๆ ต่อได้ดังเช่นตัวอย่างโค้ดข้างล่าง

```
import java.security.Identity;
...
public class MyBean implements SessionBean {
    private SessionContext ctx;
    ...
    public void bar() {
        Identity id = ctx.getCallerIdentity();
        String name = id.getName();
        System.out.println("The caller's name is " + name);
    }
}
```

8.4 เอสเอสแอล โพรโทคอล (SSL Protocol)

การนำเอสเอสแอลโพรโทคอลมาใช้ในการสร้างชื่่วิธีที่คอนเน็คชัน (security connection) นั้นเมื่อเอสเอสแอลโพรโทคอลบนเซิร์ฟเวอร์ถูกกำหนดโดยการใช้การอเท้นทิเคชันร่วมกันแล้ว การติดต่อของไคลเอ็นท์ก็จะต้องอาศัยดิจิทัลเซอร์ติฟิเคท (digital certificate) ที่ถูกระบุไว้อย่างถูกต้อง

เอสเอสแอลแฮนด์เชก (Secure Sockets Layer Handshake) โพรโทคอล ได้รับการพัฒนาโดย เนตเคปคอมมูนิเคชันคอร์ปอเรชัน (Netscape Communications Corporation) เพื่อจัดการด้านการรักษาความปลอดภัยและการใช้งานอย่างอิสระบนระบบอินเทอร์เน็ต เอสเอสแอลโพรโทคอลนี้สนับสนุนการทำอเท้นทิเคชันของสถาปัตยกรรมแบบเซิร์ฟเวอร์ - ไคลเอ็นท์ เอสเอสแอลโพรโทคอลนั้นเป็นแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็ชัณอินดิเพันเดนท์ (independent) ที่ให้อินนุญตกับโปรโตคอลต่ง ๆ เช่น เอชทีทีพี (HyperText Transfer Protocol : HTTP) เอฟทีพี (File Transfer Protocol: FTP) รวมไปถึงเทลเน็ต (Telnet) ด้วย

เอสเอสแอลโปรโตคอลสามารถติดต่อดว้ยเ็นคริปชัน (encryption) เหมือนกับการขออเท็นทิเคท (authenticate) กับเซิร์ฟเวอร์ก่อนที่จะทำการแลกเปลี่ยข้อมูลในแอปพลิเคชันระดับสูง (higher-level Application) เอสเอสแอลโปรโตคอลจะดูแลเรื่องของการรักษาความปลอดภัยและอินทิกริตี (Integrity) ของทรานสมิสชันเชแนล (Transmission channel) ด้วยวิธีการเ็นคริปชัน , ออเท็นทิเคชันและเมสเชสจออเท็นทิเคชันไค้ด.

เอสเอสแอลแซนเชกโปรโตคอล จะประกอบไปด้วย 2 เฟส คือ

- เซิร์ฟเวอร์ออเท็นทิเคชัน (Server Authentication)
- ออฟชันนอลไคลเ็นท์ออเท็นทิเคชัน (Optional Client Authentication)

สำหรับในเซิร์ฟเวอร์ออเท็นทิเคชันซึ่งเป็นเฟสแรกนั้นเซิร์ฟเวอร์จะตอบสนองการร้องขอจากไคลเ็นท์ด้วยการส่งเซอร์ติฟิเคท (certificate) และรหัส (ciphers) ไปให้ จากนั้นไคลเ็นท์จะสร้างมาสเตอร์คีย์ (master key) ซึ่งจะเกิดจากการเข้ารหัสกับพับบลิคคีย์ (public key) ของเซิร์ฟเวอร์และทำการส่งเ็นคริปมาสเตอร์คีย์ที่ผ่านการเข้ารหัสแล้วไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ได้รับมาสเตอร์คีย์แล้วจะทำการแชรรหัสออกมา จากนั้นก็จะทำการออเท็นทิเคทไปยังไคลเ็นท์ด้วยการส่งเมสเชสจออเท็นทิเคทไปกับมาสเตอร์คีย์

ในเฟสที่สอง ซึ่งก็คือออฟชันนอลเซคันเฟส (Optional second phase) นั้นเซิร์ฟเวอร์จะส่งการสอบถามไปยังไคลเ็นท์เพื่อให้ไคลเ็นท์ทำการออเท็นทิเคทตัวเองมายังเซิร์ฟเวอร์ก่อนด้วยการส่งไคลเ็นท์ดิจิตอลซิกเนเจอร์ (Client digital signature) กลับมาให้หรือที่เราเรียกกันว่าพับบลิคคีย์เซอร์ติฟิเคท (public-key certificate) นั้นเอง

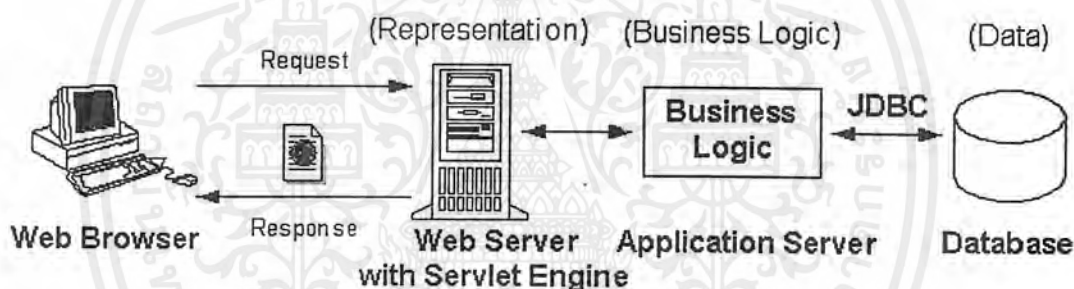
วิธีการในการเข้ารหัสมีมากมายหลายวิธีที่เอสเอสแอลสามารถใช้ได้ในช่วงการทำแซนเชกกิง (handshaking) นั้นจะใช้อาร์เอสเอพับบลิคคีย์คิฟโตซิสเต็ม (RSA public-key cryptosystem) หลังจากนั้นเมื่อมีการแลกเปลี่ยคีย์กันแล้วเลขรหัส (ciphers) ก็จะถูกนำมาใช้งาน ซึ่งจะประกอบด้วย อาร์ซี2 (RC2) , อาร์ซี4 (RC4), ไอดีอีเอ (IDEA), ดีอีเอส (DES) และทริเปิลดีอีเอส (triple-DES) สำหรับพับบลิคคีย์เซอร์ติฟิเคท (Public-key certificates) จะมีรูปแบบตามเอ็กซ์.509

8.5 แอปพลิเคชันเซิร์ฟเวอร์

สำหรับเว็บไซต์เล็ก ๆ แล้วโครงสร้างของเซิร์ฟเวอร์ไซต์แอปพลิเคชันก็เพียงพอต่อความต้องการสำหรับประโชชนใช้สอย แต่สำหรับเว็บไซต์ใหญ่ ๆ ที่มีผู้ใช้งานมหาศาลส่วนประกอบที่สำคัญอีกอย่างหนึ่งที่มีจะถูกเพิ่มเข้าไปมักจะเป็นส่วนที่เรียกว่า แอปพลิเคชันเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชันเซิร์ฟเวอร์เป็นส่วนประกอบอีกส่วนหนึ่งที่ถูกสร้างขึ้นมาเพื่อสนองความต้องการที่ว่าการสร้างระบบที่ระบบหนึ่งขึ้นมา เราควรจะแยกส่วนที่ทำหน้าที่เกี่ยวกับการบริการต่าง ๆ รวมไปถึงการจัดสรรข้อมูลออกมาเป็นอีกส่วนต่างหาก (Business Logic) ทั้งนี้เพื่อเพิ่มความคล่องตัวในการเปลี่ยนแปลงของระบบ รวมไปถึงการกระจายระบบให้แบ่งออกเป็นส่วน ๆ อย่างชัดเจนมากยิ่งขึ้น โดยถ้าดูจากบทบาทของแอปพลิเคชันเซิร์ฟเวอร์ที่ผ่านตั้งแต่อดีตจนถึงปัจจุบันเราอาจจะพูดได้ว่าแอปพลิเคชันเซิร์ฟเวอร์ ก็คือส่วนที่ช่วยอำนวยความสะดวกให้กับเซิร์ฟเวอร์ไซด์แอปพลิเคชันโดยมักจะเป็นตัวช่วยในการคำนวณข้อมูล (Services), ช่วยเป็นตัวกลางในการจัดส่งข้อมูล (Messaging Services) , ช่วยในการควบคุมการจัดเก็บข้อมูลลงไปในเคตแบบสอย่างถูกต้อง (Transaction) ตลอดจนช่วยในการเพิ่มความเร็วให้กับระบบโดยการเก็บอ็อปเจกต์ต่าง ๆ ที่ใช้หรือไม่ใช้แล้วไว้เพื่อนำมาใช้ซ้ำ (Caching) เป็นต้น ในแง่ของการออกแบบระบบ แอปพลิเคชัน เซิร์ฟเวอร์ มักจะเป็นส่วนที่เราใช้ในส่วนของที่เรียกว่า เซอร์วิสเลเยอร์ (Services Layer) ซึ่งในใจว่าแล้วส่วนนี้ก็มักจะเกี่ยวข้องกับเอ็นเตอร์ไพร์สจาว่าต่าง ๆ เช่น อีเจบี, เจทีเอส, เจเอ็มเอส, เจเอ็นดีโอ นั่นเอง) โดยประโยชน์อย่างหนึ่งของแอปพลิเคชันเซิร์ฟเวอร์ที่เราพบเห็นโดยทั่วไปก็คือการนำมาใช้สำหรับรันอีเจบี



รูปที่ 8-1 แสดงหลักการของ แอปพลิเคชัน เซิร์ฟเวอร์

8.6 เอ็นเตอร์ไพร์สจาว่า

ถ้าพูดกันในเชิงนิยามแล้ว อีเจบี ไม่ได้ถูกรันที่ แอปพลิเคชัน เซิร์ฟเวอร์ แต่จะถูกรันกับส่งหนึ่งที่เราเรียกว่า อีเจบี เซิร์ฟเวอร์ อย่างไรก็ตาม แอปพลิเคชัน เซิร์ฟเวอร์ ส่วนมากมักจะสนับสนุน อีเจบี ซึ่งเป็นส่วนหนึ่งของเจทูอีอีคั่นนั้น แอปพลิเคชัน เซิร์ฟเวอร์ ส่วนมากจึงมีส่วนหนึ่งที่ทำหน้าที่เป็น อีเจบี เซิร์ฟเวอร์ เพื่อรันอีเจบี ต่าง ๆ ได้โดยอีเจบีสามารถแบ่งออกอย่างคร่าว ๆ เป็นสามแบบคือ

8.6.1 เซลล์ชั้น

เป็นอีเจบีที่มักจะถูกใช้เป็นตัวเก็บรายละเอียดของผู้ใช้ที่กำลังทำกิจกรรมบางอย่างกับระบบหรืออาจจะถูกใช้เป็นตัวเก็บข้อมูลสำหรับ อีเจบี ต่าง ๆ ที่เจเอสพีหรือเซิร์ฟเลต สามารถเรียกใช้เพื่ออำนวยความสะดวกในการประมวลผลตลอดจนใช้ในการเปลี่ยนแปลงสถานะต่าง ๆ ของระบบโดยเรามักเรียกส่วนนี้ว่า Services Layer นั่นเอง

8.6.2 เอ็นดีบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น อีเจบี ที่ใช้สำหรับเก็บค่าต่าง ๆ ที่เราทำการแมปมาจากเคต้าที่อยู่ในเคต้าเบส, ไฟล์, แอลดี เอพีอีอบเจ็กต์ หรืออื่น ๆ ซึ่งจุดหลักของการนำเคต้าต่าง ๆ มาใส่ไว้ในเอ็นติตี้บีนก็คือการเพิ่มความง่ายในการเปลี่ยนแปลงเคต้าโดยผ่านทาง อีเจบี อีอบเจ็กต์, การง่ายต่อการควบคุมการอ่านและเขียนเคต้าต่าง ๆ ได้ อย่างถูกต้องในกรณีที่มีระบบมีมากกว่า 1 เทรดเข้ามาเกี่ยวข้องในการอ่านเขียนหรือเปลี่ยนแปลงเคต้าตัวเดียวกันในเวลาเดียวกัน รวมไปถึงความเร็วที่เพิ่มขึ้นในการเก็บเคต้าต่าง ๆ ไว้ในหน่วยความจำ (Bean Caching) แทนที่จะต้องทำการอ่านเคต้าใหม่จากเคต้าเบสทุกครั้งที่เราต้องการทำกิจกรรมบางอย่างกับเคต้าเหล่านั้น โดยข้อดีต่าง ๆ ที่กล่าวมาข้างต้นเหล่านี้จะเกิดขึ้นได้จากการทำงานร่วมกันระหว่าง เอ็นติตี้ บีน, อีเจบี คอนเทนเนอร์ และ อีเจบีเซิร์ฟเวอร์ อย่างไรก็ตามนักพัฒนาระบบก็ยังคงหลีกเลี่ยงกันอยู่ถึงประสิทธิภาพที่เกิดขึ้นจากการใช้เอ็นติตี้ บีน เพื่อเก็บเคต้าต่าง ๆ ไว้ โดยผลกระทบนี้มาจากการที่แต่ละเทรดจะต้องแย่งกันเข้าไปใช้เอ็นติตี้บีนตัวเดียวกันเพื่อทำการเขียนอ่านหรือเปลี่ยนแปลงเคต้า (Bean Contention) ซึ่งผลที่เกิดขึ้นอาจจะทำให้ระบบช้าลงกว่าการใช้เจดีบีซีหรือ อีอบเจ็กต์เคต้าเบสธรรมดาทั่ว ๆ ไปก็ได้



บทที่ 9

การออกแบบระบบการค้าเชิงอิเล็กทรอนิกส์

9.1.แนวคิดในการออกแบบ

9.1.1 คุณสมบัติส่วนของระบบงานให้บริการ

ระบบโอล่าล่าทัวร์ในโปรเจกต์นี้มีลักษณะแตกต่างจากระบบการค้าอื่นๆทั่วไปที่เราพบบนอินเทอร์เน็ต คือโดยทั่วไปแต่ละระบบการค้าจะให้บริการเพียงส่วนของตนเอง หรือบางแห่งก็รวบรวมเอาบริการหลายๆ ชนิดจากหลายๆ ที่มาเก็บไว้ที่ฐานข้อมูลของตน เมื่อระบบเหล่านี้ต้องการจะขายสินค้าหรือบริการก็นำเอารายการการให้บริการของตนมาฝากไว้กับระบบงานนั้น แต่ระบบงานให้บริการในโปรเจกต์ของเราจะมีคุณสมบัติคือ ตัวระบบงานส่วนกลางสามารถทำการติดต่อเชื่อมโยงกับระบบการให้บริการอื่นๆ ที่ทำการค้าร่วมกันอยู่ได้ เช่นการที่ลูกค้าสามารถเข้าไปค้นหาข้อมูลต่างๆ ของระบบการให้บริการหนึ่งๆ ได้โดยตรง เช่น ระบบโรงแรม ระบบงานบริการของรถบัส และสายการบิน โดยระบบส่วนกลาง หรือ โอล่าล่าทัวร์ นั้นจะเป็นผู้รับผิดชอบในเรื่อง ต่าง ๆ เช่น การเก็บประวัติและสถานะต่าง ๆ ของลูกค้า การดูแลในเรื่องความปลอดภัยในการเข้าใช้งานระบบของลูกค้าแต่ละคน รวมถึงความปลอดภัยในการเข้าถึงข้อมูลในส่วนต่างๆ อีกด้วย

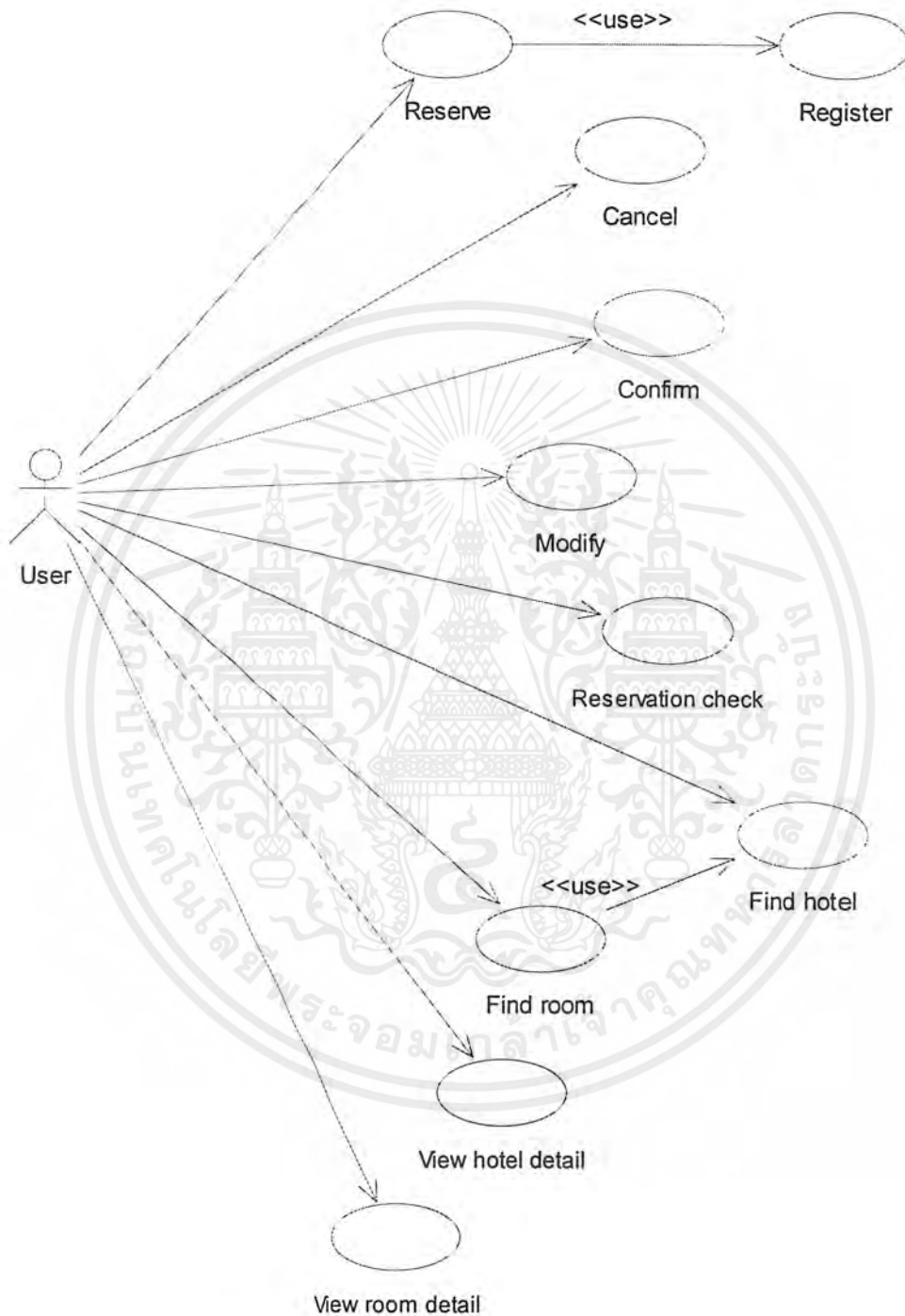
โดยภาพรวมแล้วนั้น ระบบโอล่าล่าทัวร์ เปรียบเสมือนบริษัทที่ทำหน้าที่นำเที่ยว จองโรงแรม และตั๋วเครื่องบิน ให้บริการผ่านอินเทอร์เน็ต ผู้ใช้ซึ่งใช้งานผ่านทางอินเทอร์เน็ตสามารถ ใช้งานระบบซึ่งแบ่งเป็นส่วนย่อย ได้ 6 ส่วนคือ

1. ส่วนการจองตั๋วเครื่องบิน (Airline)
2. ส่วนการจองตัวรถบัส (Bus)
3. ส่วนการจองโรงแรมและที่พัก (Hotel)
4. ส่วนธนาคาร (Bank)
5. ส่วนภัตตาคาร (Restaurant)
6. ส่วนโอล่าล่าทัวร์ (Olala Tour)

ในที่นี้จะอธิบายเพียงส่วนที่ 3, 4, 5 เท่านั้น

3. ระบบการจองโรงแรม (Hotel)

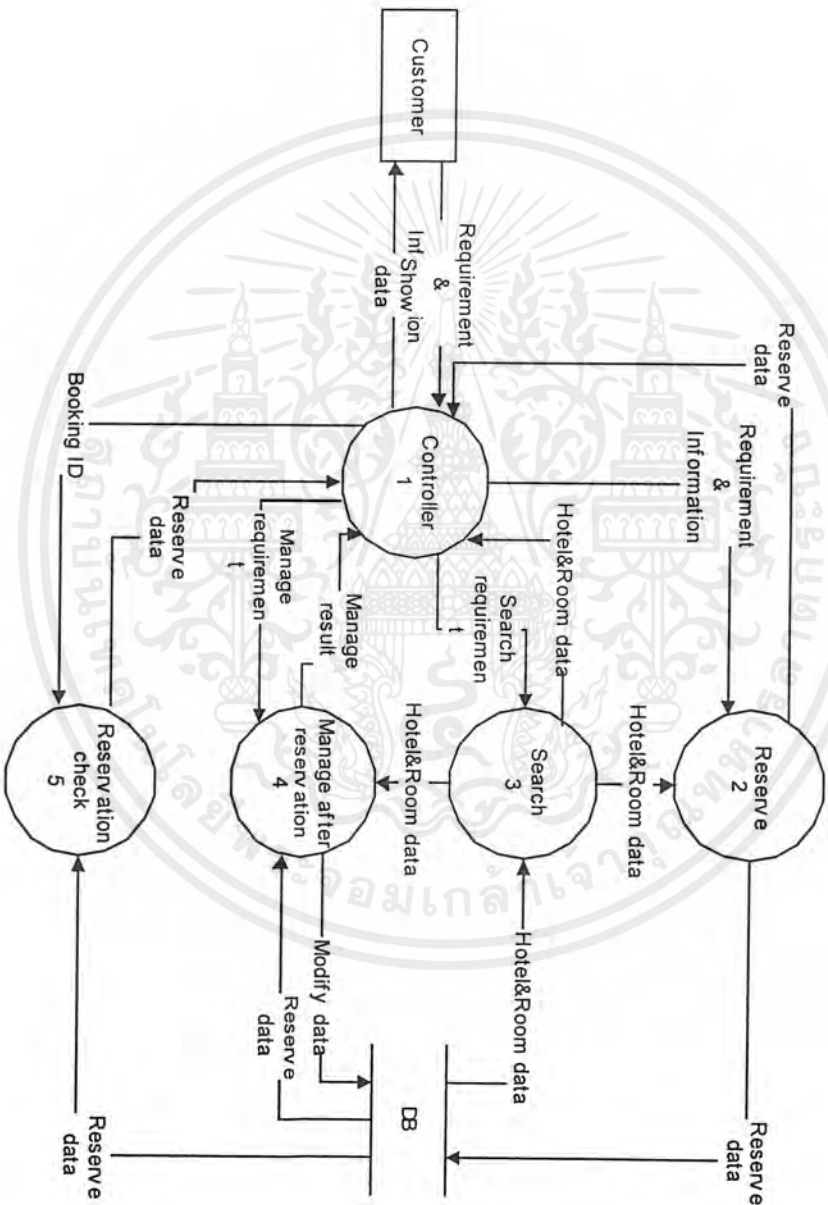
3.1 ยูสเคสไดอะแกรม (Use Case Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

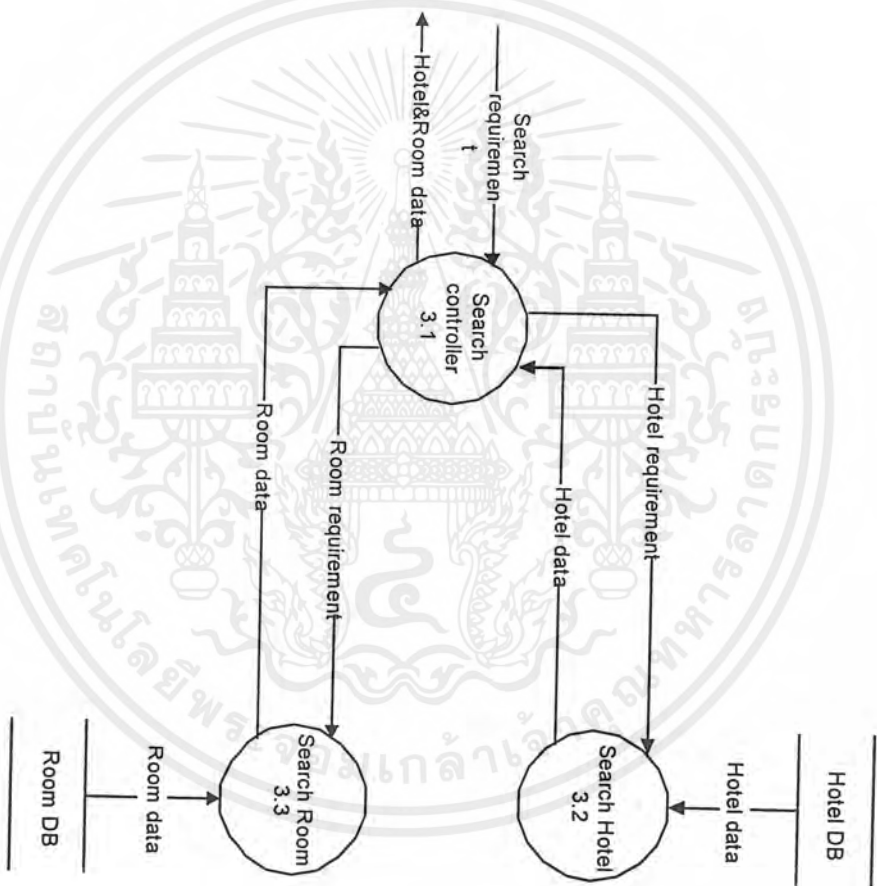
3.2 คาทำไฟล์ไลอะแกรม (Dataflow Diagram)

Level 1



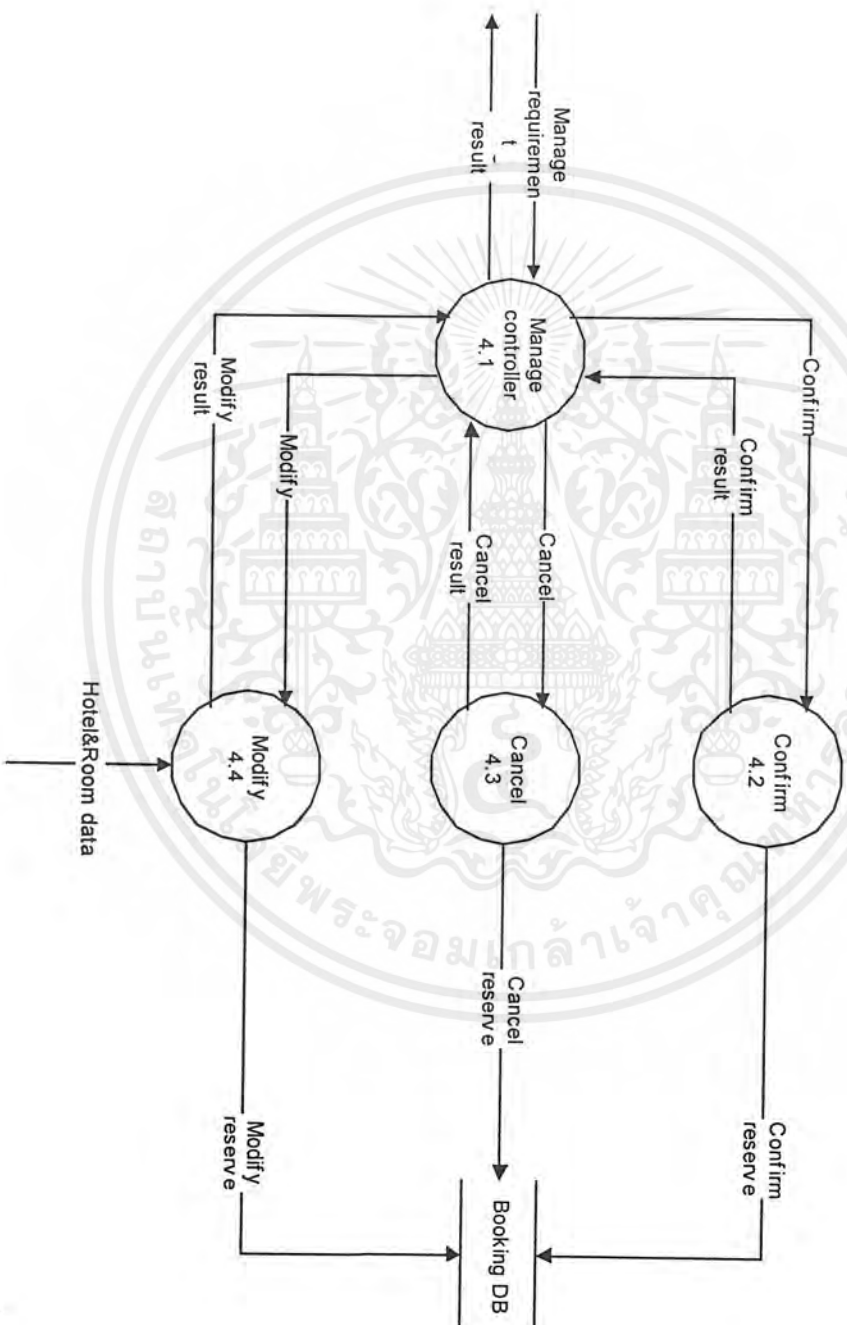
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Level 2 Search



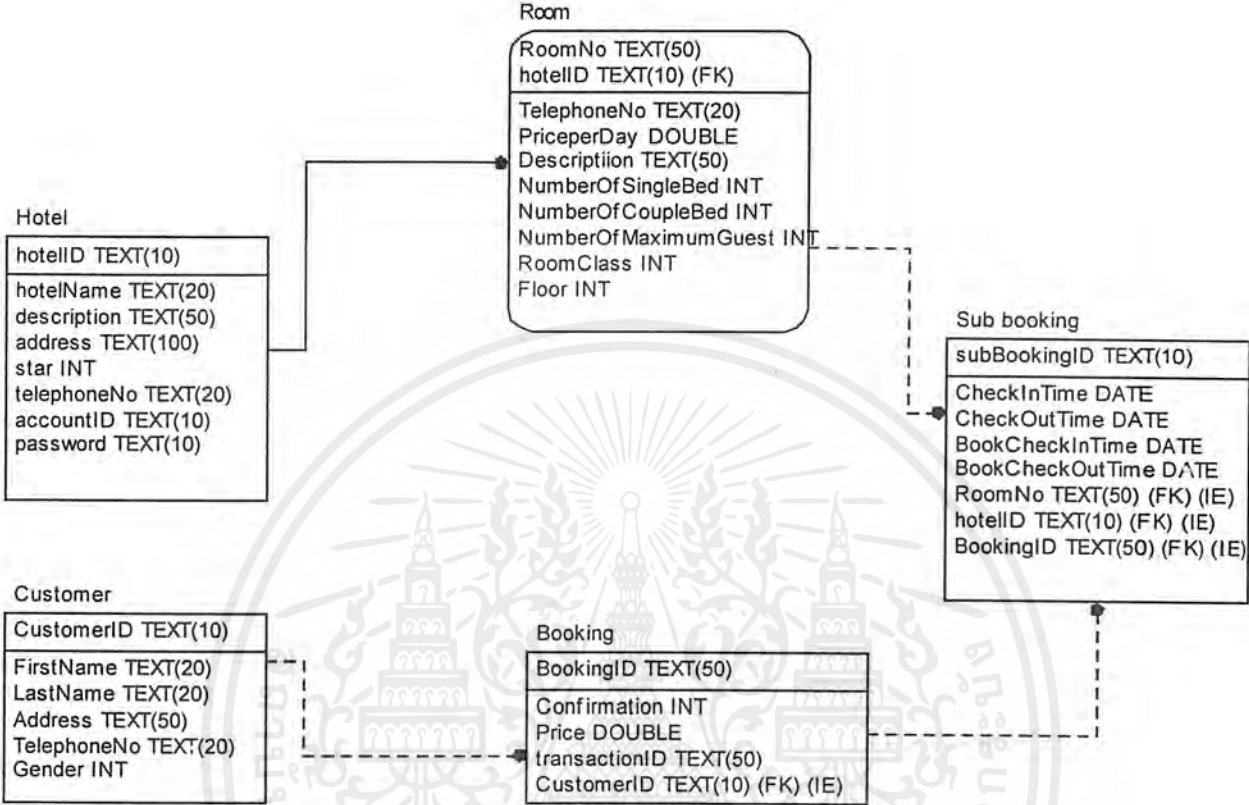
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Level 2 Manage After Reservation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

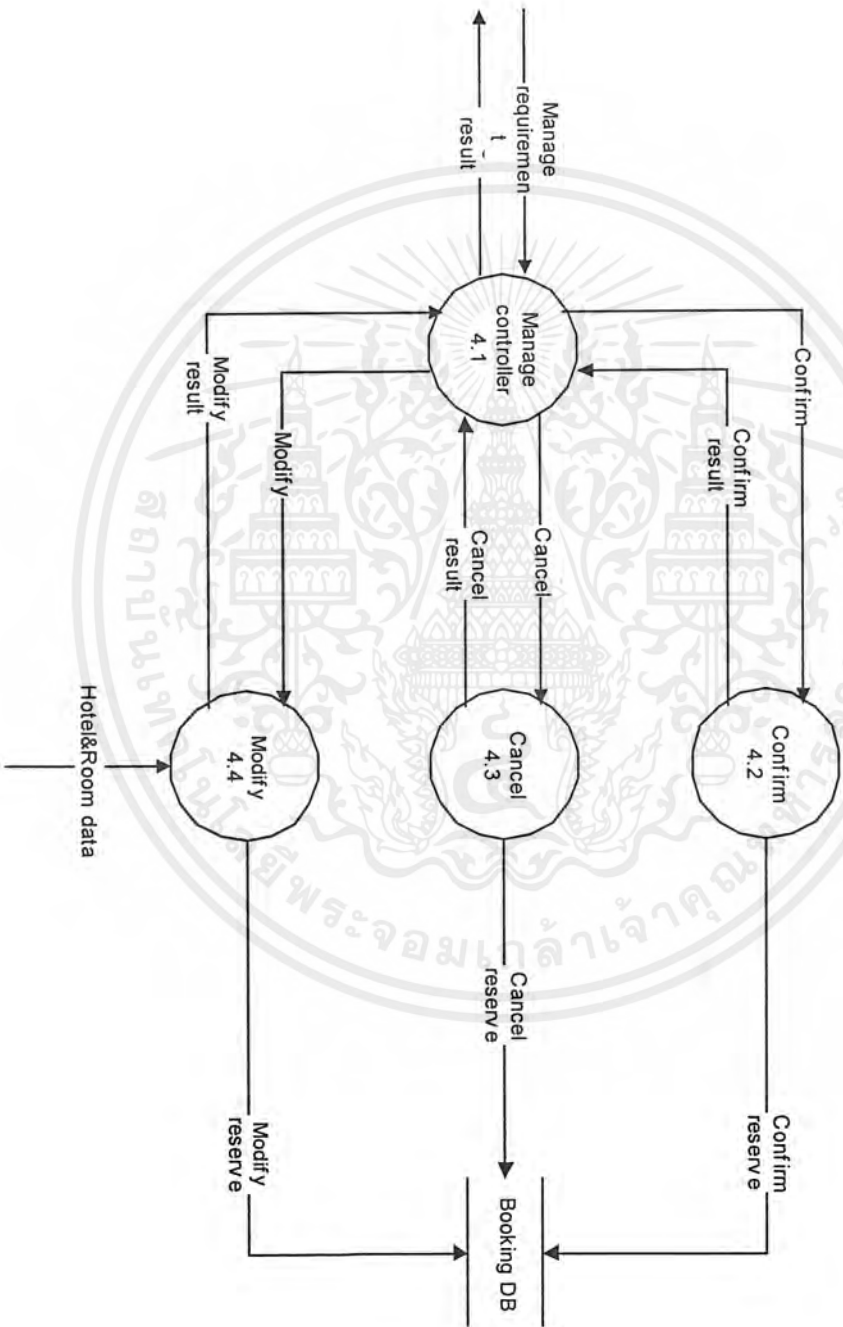
3.3 อีอีเอไออะแกรม (ER Diagram)



อีอีเอไออะแกรมแสดงความสัมพันธ์ระหว่าง ข้อมูล โรงแรม, ลูกค้าที่มาพัก และ การจองห้องพัก

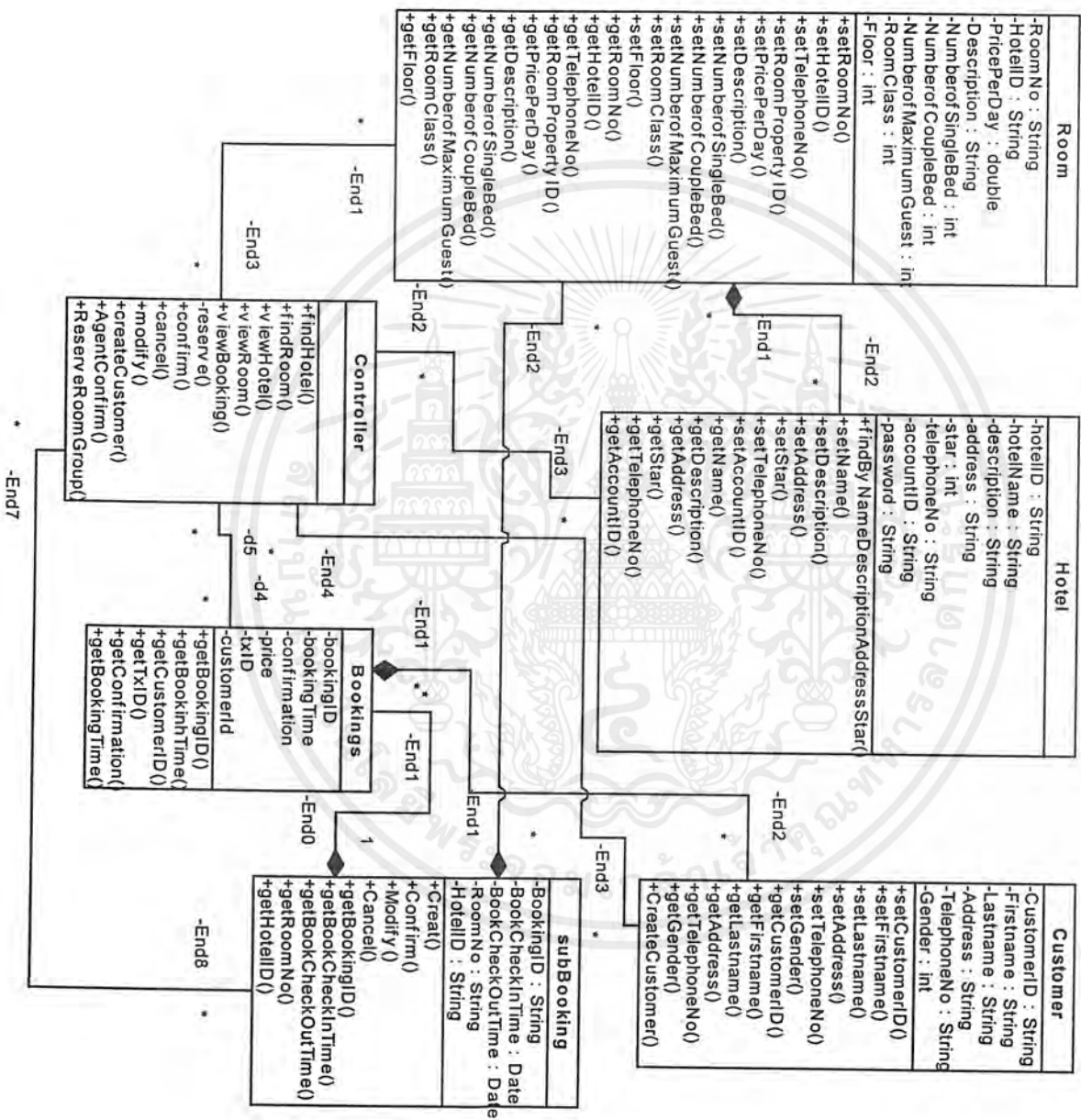
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Level 2 Manage After Reservation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

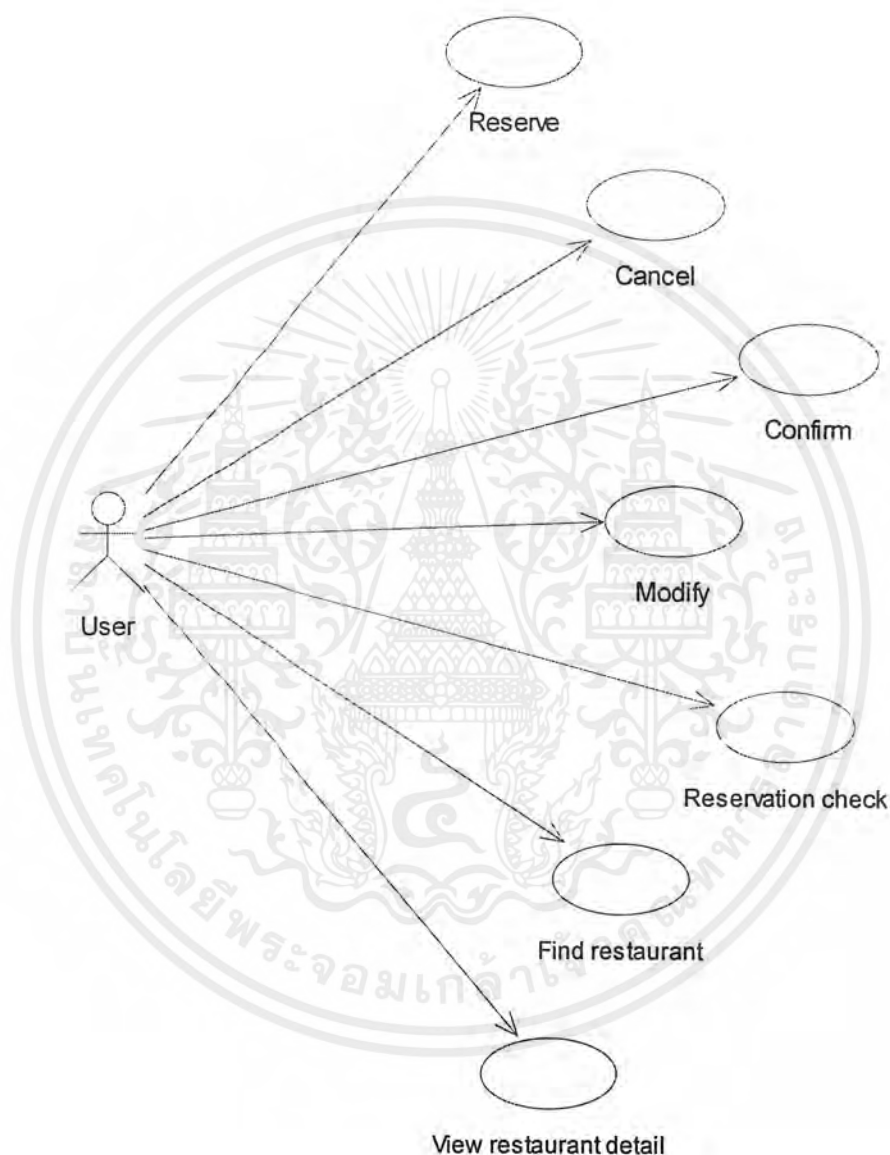
3.4 คลาสไดอะแกรม 2 (Class Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

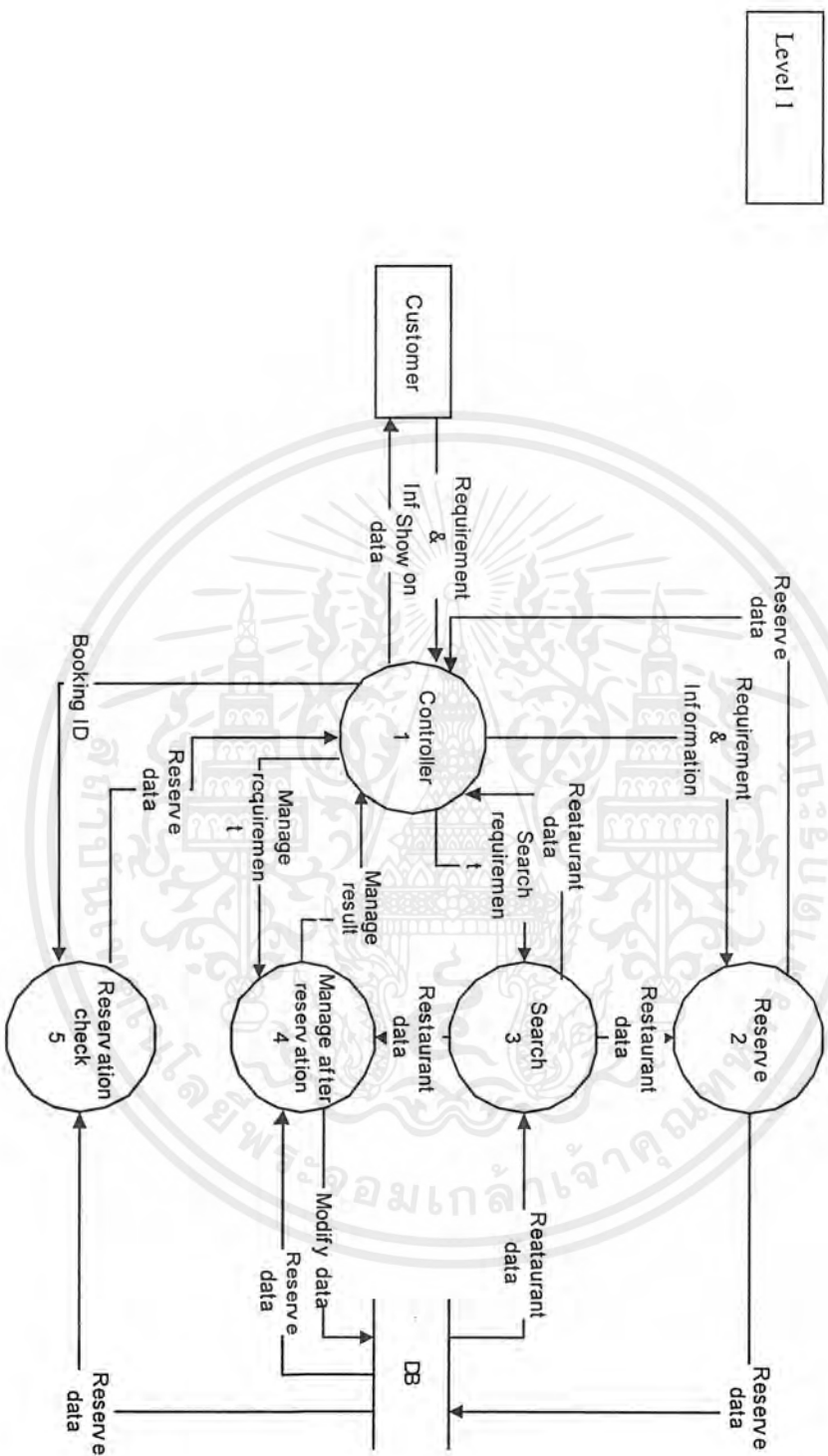
5. ระบบภัตตาคาร (Restaurant)

5.1 ยูสเคสไดอะแกรม (Use Case Diagram)



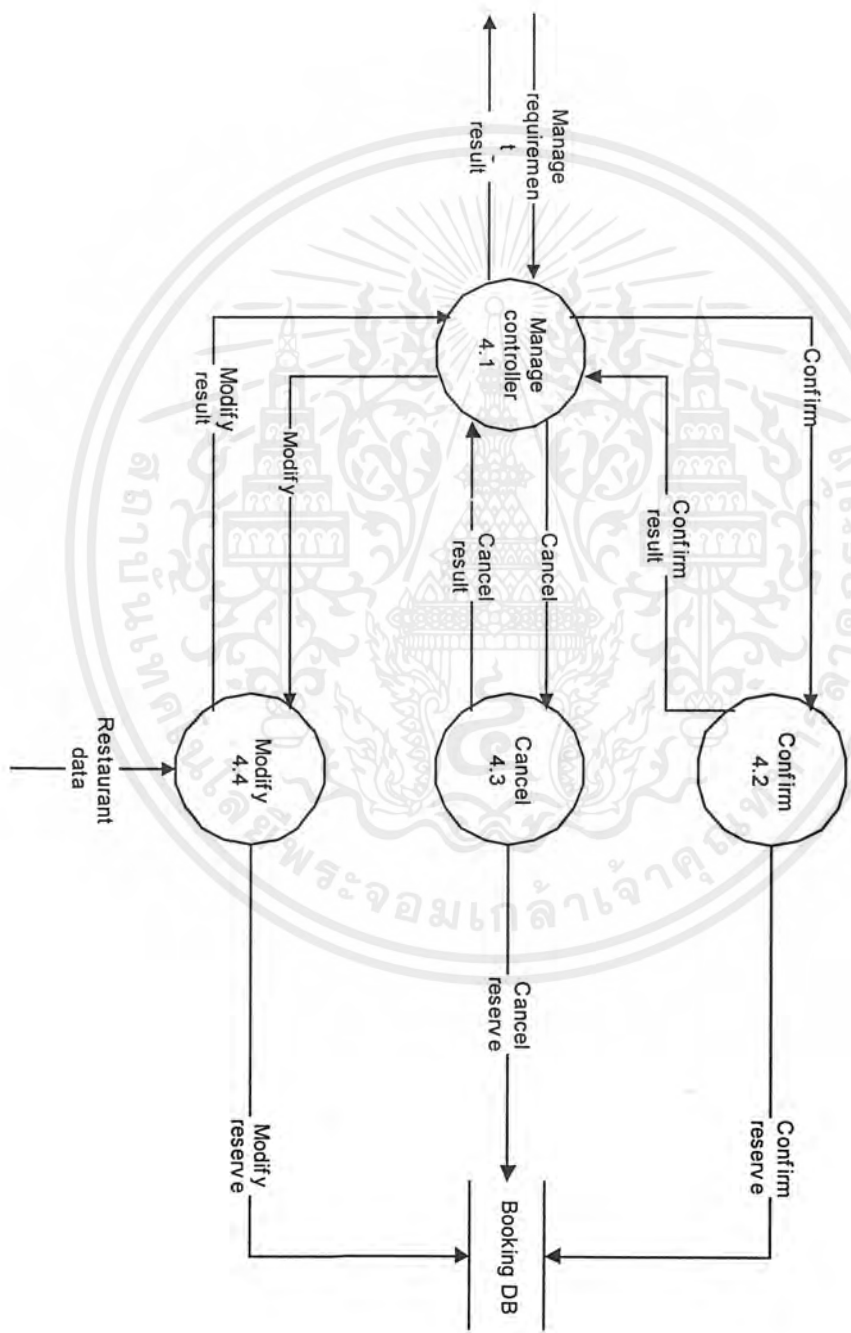
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 คาต้าไฟล์ โปรแกรม (Dataflow Diagram)



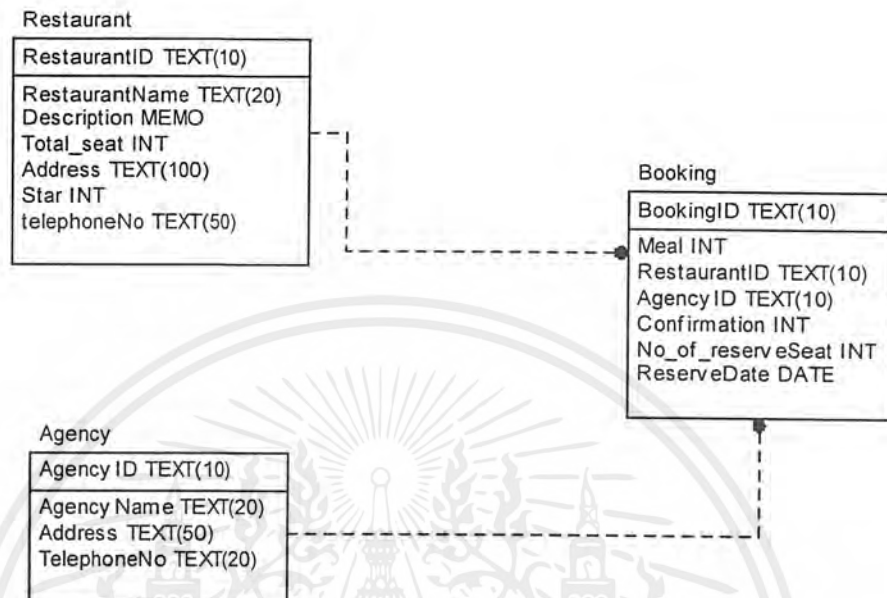
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Level 2 Manage After Reservation



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

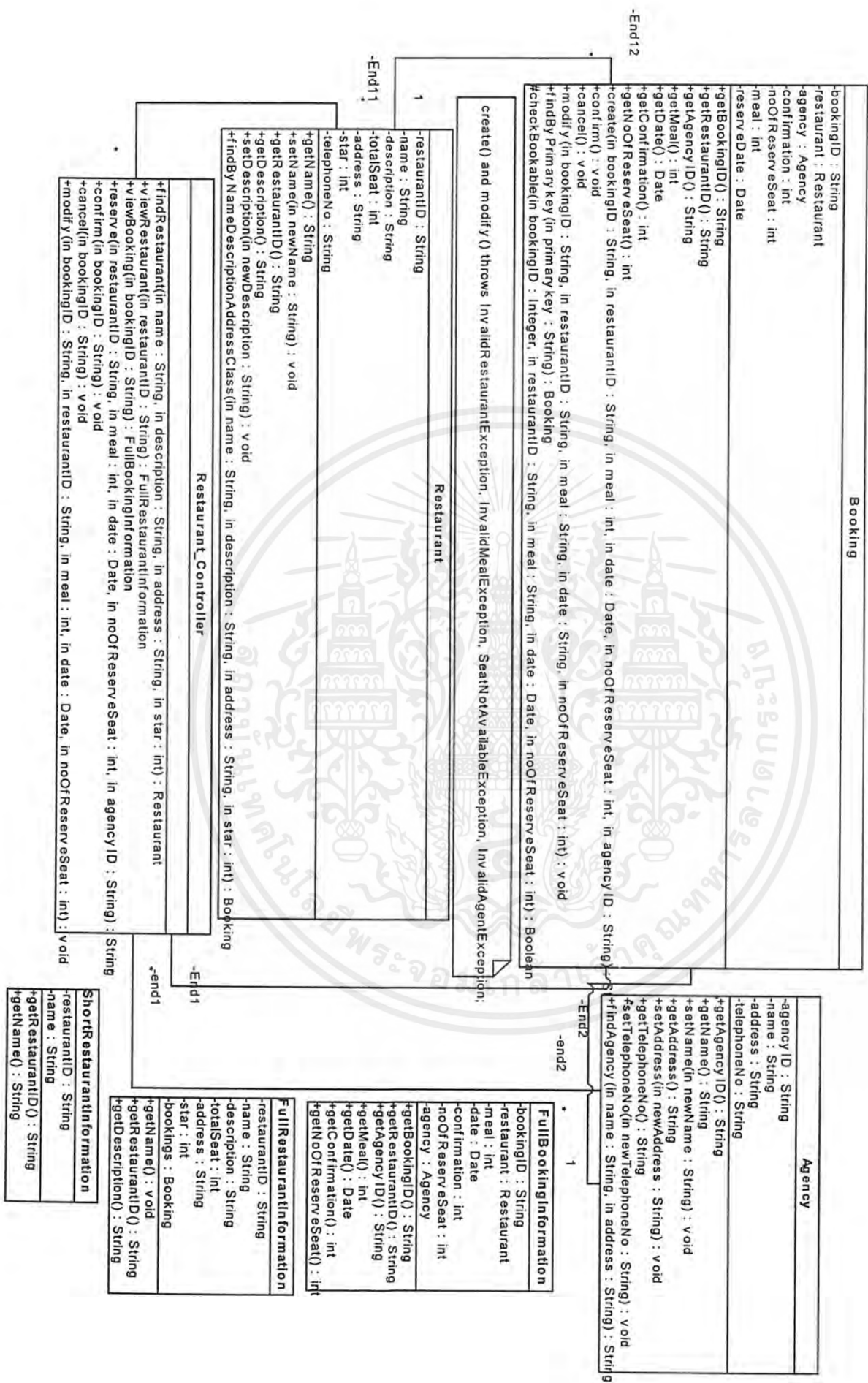
5.3 อีอาร์ไดอะแกรม (ER Diagram)



อีอาร์ไดอะแกรมแสดงความสัมพันธ์ระหว่างข้อมูล ภัตตาคาร, คนจองโต๊ะ และการจอง

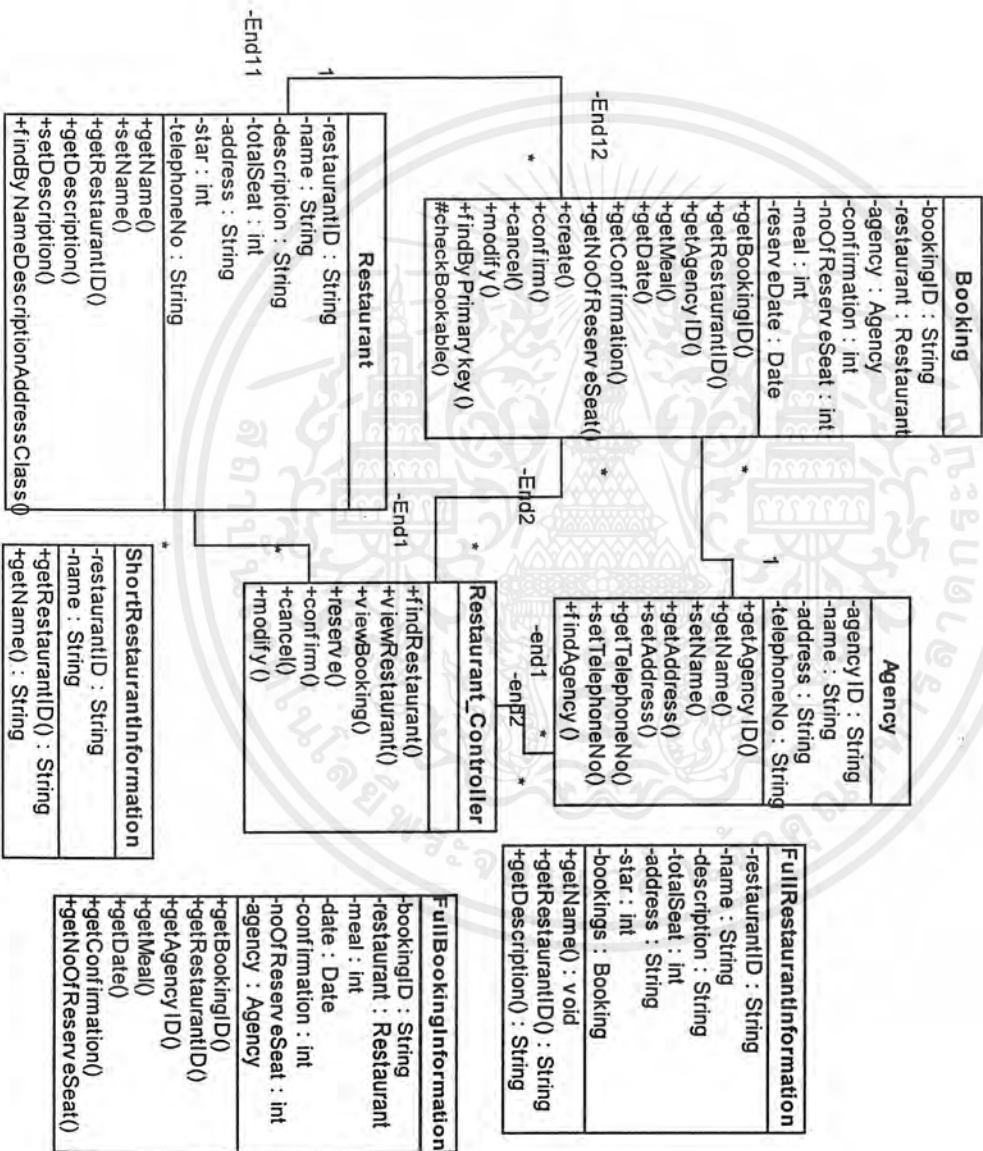
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 คลาสไดอะแกรม1 (Class Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

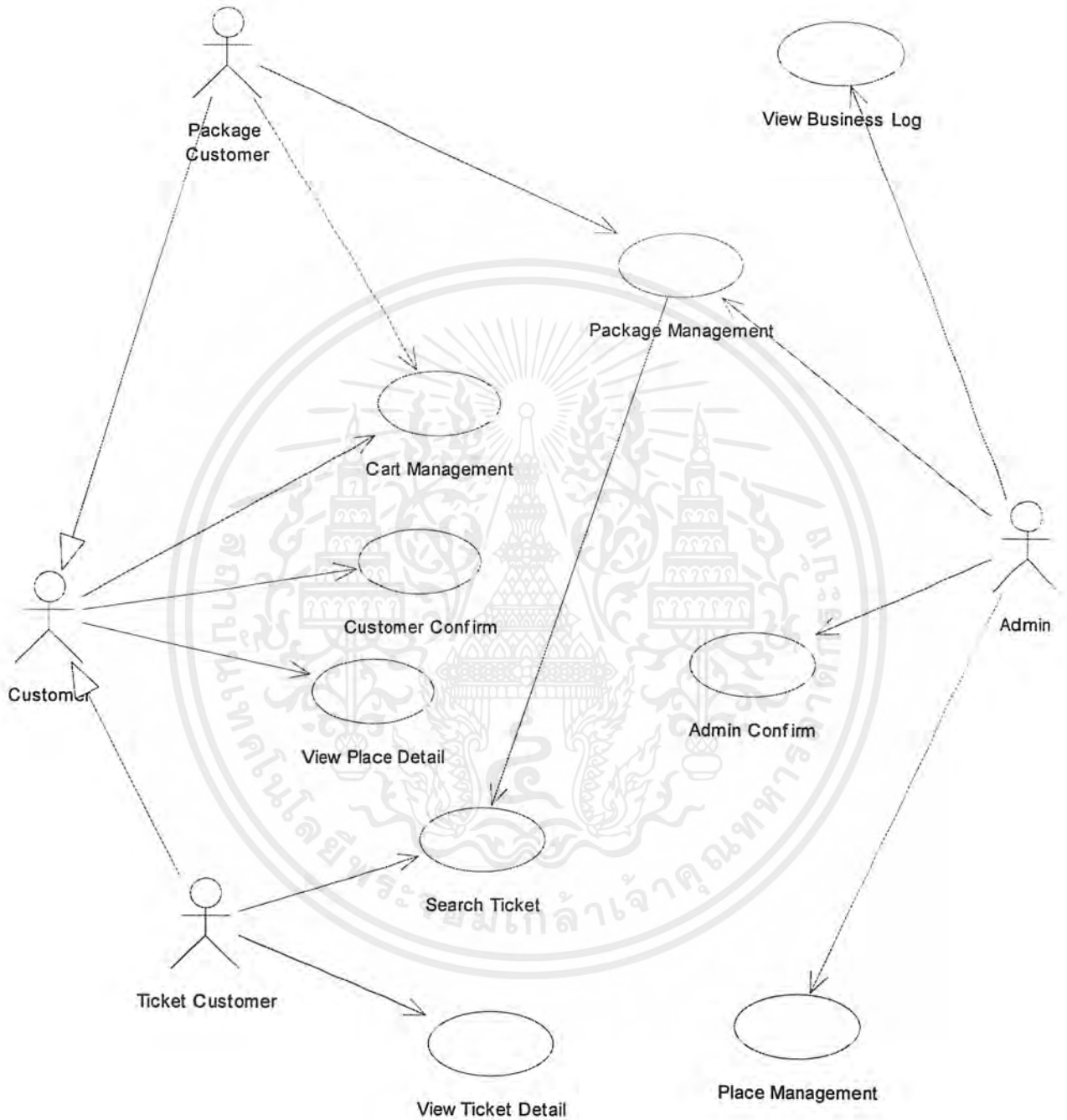
5.4 คลาสไดอะแกรม2 (Class Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

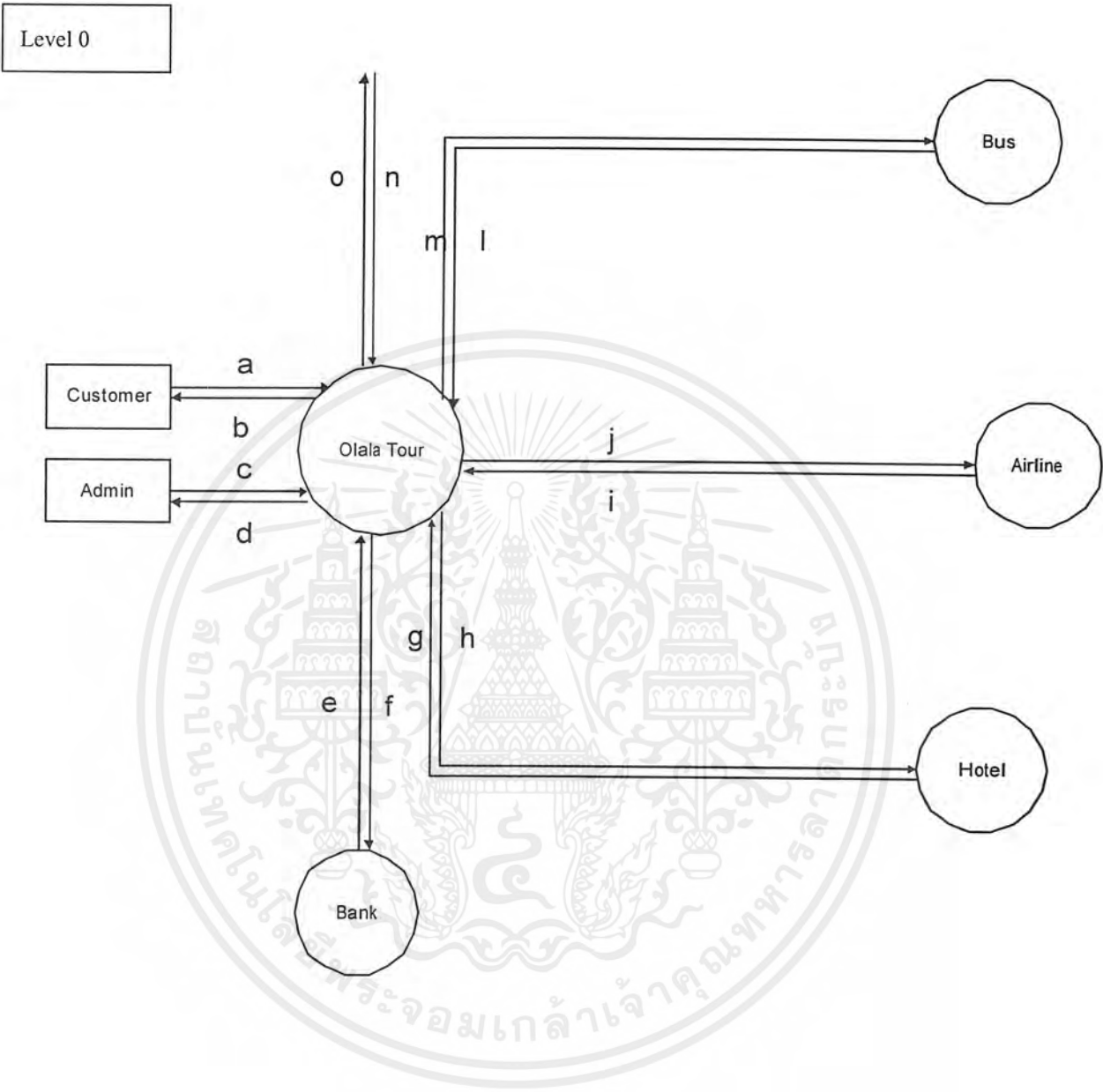
6. ระบบโอล่าทัวร์ (Olalatur)

6.1 ยูสเคสไดอะแกรม (Use Case Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ค้าไฟล์ไดอะแกรม (Dataflow Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายดาต้าโฟลไดอะแกรมโอล่าต่ำสุดเลเวล 0 มีดังนี้

A

- a.1) ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวที่ต้องการ, สถานที่ท่องเที่ยวที่ต้องการ, เวลาเดินทาง และ กิจกรรมของสถานที่ท่องเที่ยวที่ต้องการ (การค้นหาสถานที่ท่องเที่ยว)
- a.2) ข้อมูลเกี่ยวกับการเดินทางที่ต้องการ, เส้นทางเดินทางที่ต้องการ, เวลาเดินทาง, ข้อมูลเกี่ยวกับพาหนะที่ต้องการ, ต้นทาง และ ปลายทาง (การค้นหาคำเดินทาง)
- a.3) ข้อมูลเกี่ยวกับสถานที่พักที่ต้องการ, ที่พักที่ต้องการ, เวลาเข้า และออกที่พัก, รายละเอียดของห้องพัก และ รายละเอียดของโรงแรม (การค้นหาโรงแรม)
- a.4) ข้อมูลของลูกค้า
- a.5) เส้นทางเดินทางและ ที่พัก (การ Add, Remove และ Modify Cart)
- a.6) BookingID และ รหัสผ่าน (การ ดกตง, ยกเลิก และแก้ไขการจอง)
- a.7) a.1 + a.2 + a.3 (การดูรายละเอียดของแพคเกจ)

B

- b.1) ชื่อสถานที่, รายละเอียดสถานที่, ประเภท, ที่ตั้ง และกิจกรรมของสถานที่ (สถานที่)
- b.2) ชื่อเส้นทางเดินทาง, รายละเอียด, ราคา และ ตารางเวลา (การเดินทาง)
- b.3) ชื่อโรงแรม, ค่าที่พัก, รายละเอียดของห้องพัก, รายละเอียดของโรงแรม (ที่พัก)
- b.4) Item ใน Cart (ที่พัก, ตัวบ่งชี้เส้นทางเดินทาง (เช่น RouteID), สถานะของที่พัก และการเดินทาง
- b.5) ผลลัพธ์ในการ Submit Cart
- b.6) พาสเวิร์ดของการจอง และ BookingID (สำหรับพาสเวิร์ดจะส่งให้ทางอีเมล)
- b.7) b.1 + b.2 + b.3 + ตารางการเดินทาง (สำหรับดูรายละเอียดของแพคเกจ) และผลการจองแพคเกจ

C

- c.1) a.1 (การค้นหาสถานที่ท่องเที่ยว) + b.1 (การเพิ่ม และแก้ไขสถานที่ท่องเที่ยว)
- c.2) ชื่อเส้นทางเดินทาง, จำนวนที่นั่ง, ข้อมูลเพิ่มเติม และ วัน เวลาในการเดินทาง (ในการจองตั๋วเดินทาง)
- c.3) ชื่อที่พัก (โรงแรม), ประเภทห้องพัก, ข้อมูลเพิ่มเติม,จำนวนห้องพัก, วันเข้าพัก และวันออกจากที่พัก (ในการจองที่พัก)
- c.4) a.2 (การค้นหาเส้นทาง)
- c.5) a,3 (การค้นหาที่พัก)
- c.6) b.1 + c.1 + c.3 + c.4 + ชื่อแพคเกจ, ตารางการเดินทาง, ราคาแพคเกจ, GuideID (การสร้าง แพคเกจ)
- c.7) BookingID, รายละเอียดของลูกค้า (การยืนยันการจองตั๋วเดินทาง และที่พัก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

c.8) ข้อมูลของไกด์ (การค้นหาไกด์)

D

d.1) ผลลัพธ์การเพิ่มสถานที่ (การเพิ่มสถานที่)

d.2) ผลลัพธ์การจอง, เส้นทางใกล้เคียง

d.3) b.2 (การค้นหาเส้นทาง)

d.4) b.3 (การค้นหาที่พัก)

d.5) ชื่อไกด์, รายละเอียดของไกด์ (การค้นหาไกด์)

d.6) ผลลัพธ์การสร้างแพ็คเกจ (การสร้างแพ็คเกจ)

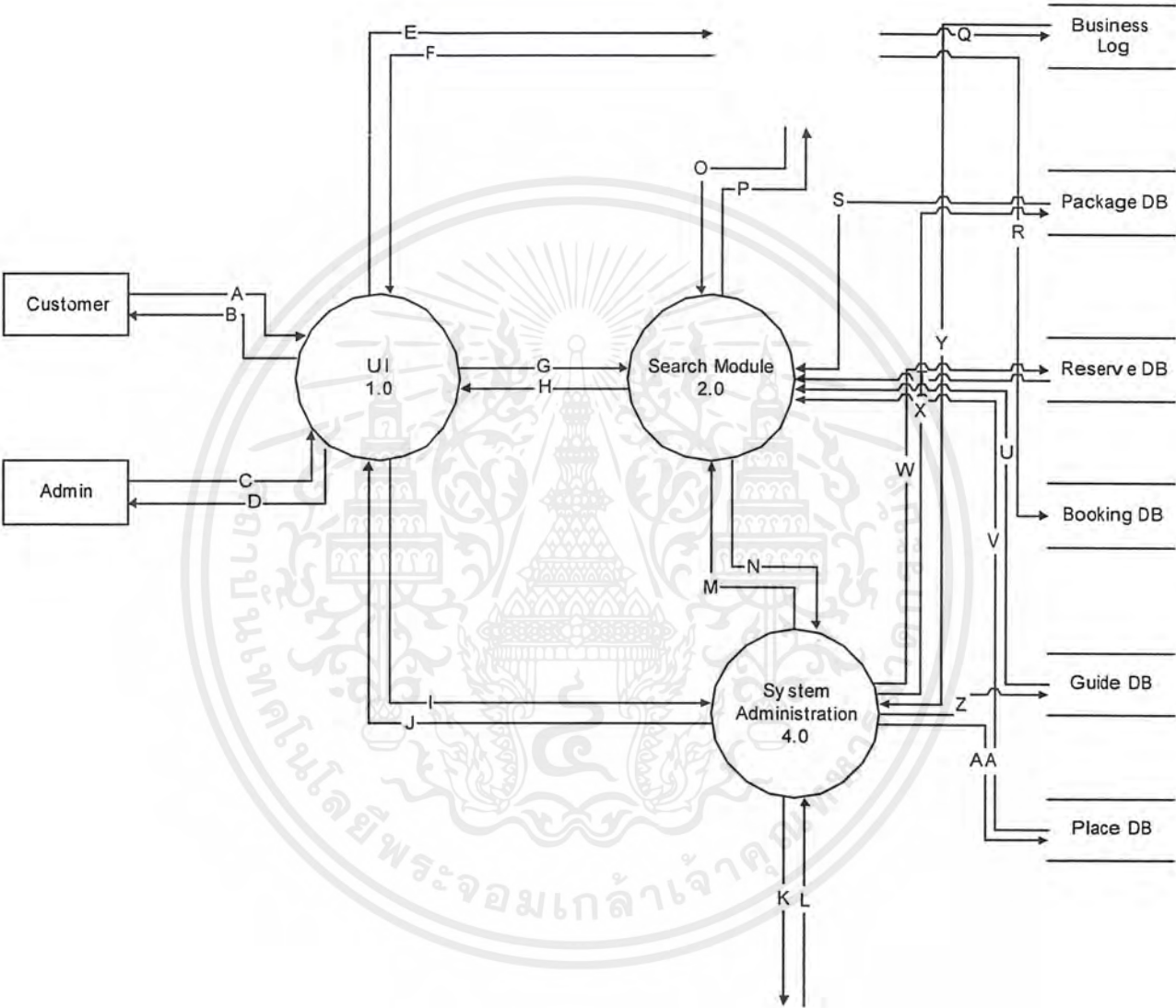
d.7) ผลลัพธ์การยืนยัน (การยืนยัน)

d.8) รายละเอียดการ Booking ของลูกค้าทุกคน, รายละเอียดการยืนยันของลูกค้าทุกคน รายละเอียดการเพิ่มเติม/แก้ไขสถานที่ของ Admin และการ สร้าง/แก้ไขแพ็คเกจของ Admin (ในการ View log)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Level 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายดาต้าไฟล์ไดอะแกรมโวลล่าทัวร์เลเวล 1 มีดังนี้

A

- a.1) ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวที่ต้องการ, สถานที่ท่องเที่ยวที่ต้องการ, เวลาเดินทาง และ Activity ของสถานที่ท่องเที่ยวที่ต้องการ (การ Search สถานที่ท่องเที่ยว)
- a.2) ข้อมูลเกี่ยวกับการเดินทางที่ต้องการ, เส้นทางการเดินทางที่ต้องการ, เวลาเดินทาง, ข้อมูลเกี่ยวกับพาหนะที่ต้องการ, ต้นทาง และ ปลายทาง (การค้นหาการเดินทาง)
- a.3) ข้อมูลเกี่ยวกับสถานที่พักที่ต้องการ, ที่พักที่ต้องการ, เวลาเข้า และออกที่พัก, รายละเอียดของห้องพัก และ รายละเอียดของโรงแรม (การค้นหาโรงแรม)
- a.4) ข้อมูลของลูกค้า
- a.5) เส้นทางการเดินทาง และ ที่พัก (การ Add, Remove และ Modify Cart)
- a.6) BookingID และ รหัสผ่าน (การ ตกลง, ยกเลิก และแก้ไขการจอง)
- a.7) $a.1 + a.2 + a.3$ (การดูรายละเอียดของแพคเกจ)

B

- b.1) ชื่อสถานที่, รายละเอียดสถานที่, ประเภท, ที่ตั้ง และกิจกรรมของสถานที่ (สถานที่)
- b.2) ชื่อเส้นทางการเดินทาง, รายละเอียด, ราคา และ ตารางเวลา (การเดินทาง)
- b.3) ชื่อโรงแรม, ค่าที่พัก, รายละเอียดของห้องพัก, รายละเอียดของโรงแรม (ที่พัก)
- b.4) Item ใน Cart (ที่พัก, ตัวบ่งชี้เส้นทางเดินทาง (เช่น RouteID), สถานะของที่พัก และการเดินทาง)
- b.5) ผลลัพธ์ในการ Submit Cart
- b.6) พาสเวิร์ดของการจอง และ BookingID (สำหรับพาสเวิร์ดจะส่งให้ทางอีเมล)
- b.7) $b.1 + b.2 + b.3 +$ ตารางการเดินทาง (สำหรับดูรายละเอียดของแพคเกจ) และผลการจองแพคเกจ

C

- c.1) $a.1$ (การค้นหาสถานที่ท่องเที่ยว) + $b.1$ (การเพิ่ม และแก้ไขสถานที่ท่องเที่ยว)
- c.2) ชื่อเส้นทางการเดินทาง, จำนวนที่นั่ง, ข้อมูลเพิ่มเติม และ วัน เวลาในการเดินทาง (ในการจองตั๋วเดินทาง)
- c.3) ชื่อที่พัก (โรงแรม), ประเภทห้องพัก, ข้อมูลเพิ่มเติม,จำนวนห้องพัก, วันเข้าพัก และวันออกจากที่พัก (ในการจองที่พัก)
- c.4) $a.2$ (การค้นหาเส้นทาง)
- c.5) $a.3$ (การค้นหาที่พัก)
- c.6) $b.1 + c.1 + c.3 + c.4 +$ ชื่อแพคเกจ, ตารางการเดินทาง, ราคาแพคเกจ, GuideID (การสร้างแพคเกจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

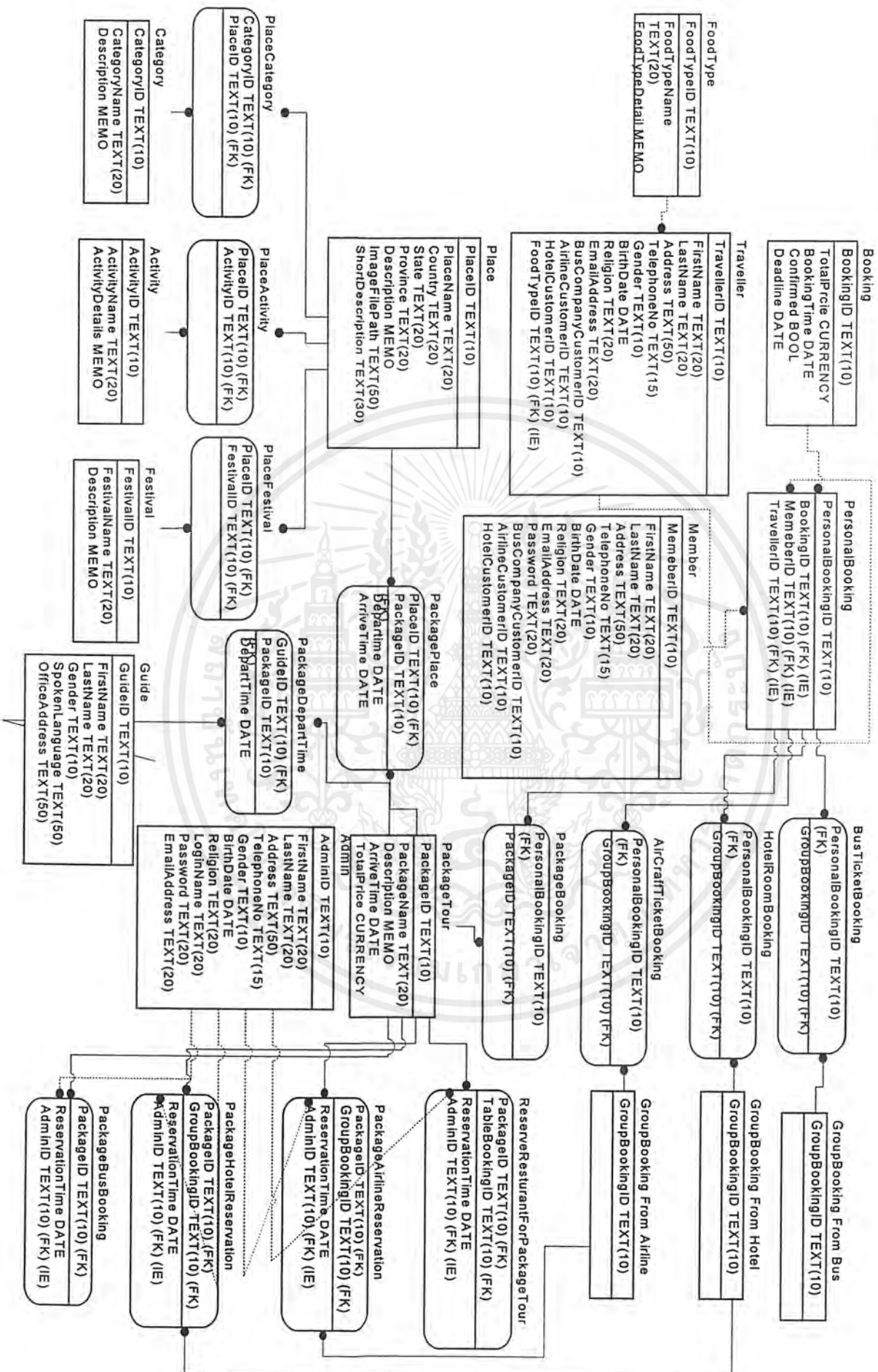
- c.7) BookingID, รายละเอียดของลูกค้า (การขึ้นชั้นการจองตัวเดินทาง และที่พัก)
- c.8) ข้อมูลของไกด์ (การค้นหาไกด์)

D

- d.1) ผลลัพธ์การเพิ่มสถานที่ (การเพิ่มสถานที่)
- d.2) ผลลัพธ์การจอง, เส้นทางใกล้เคียง
- d.3) b.2 (การค้นหาเส้นทาง)
- d.4) b.3 (การค้นหาที่พัก)
- d.5) ชื่อไกด์, รายละเอียดของไกด์ (การค้นหาไกด์)
- d.6) ผลลัพธ์การสร้างแพ็คเกจ (การสร้างแพ็คเกจ)
- d.7) ผลลัพธ์การขึ้นชั้น (การขึ้นชั้น)
- d.8) รายละเอียดการ Booking ของลูกค้าทุกคน, รายละเอียดการขึ้นชั้นของลูกค้าทุกคน รายละเอียดการเพิ่มเติม/แก้ไขสถานที่ของ Admin และการ สร้าง/แก้ไขแพ็คเกจของ Admin (ในการ View log)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

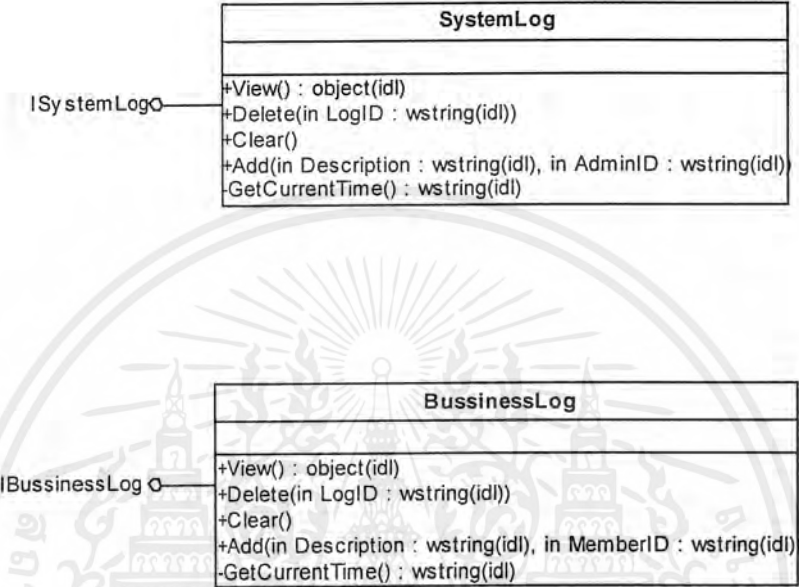
6.3 อีอาร์ไดอะแกรม (ER Diagram)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 คลาสไดอะแกรม (Class Diagram) ประกอบด้วยระบบแต่ละส่วนดังนี้

Log System

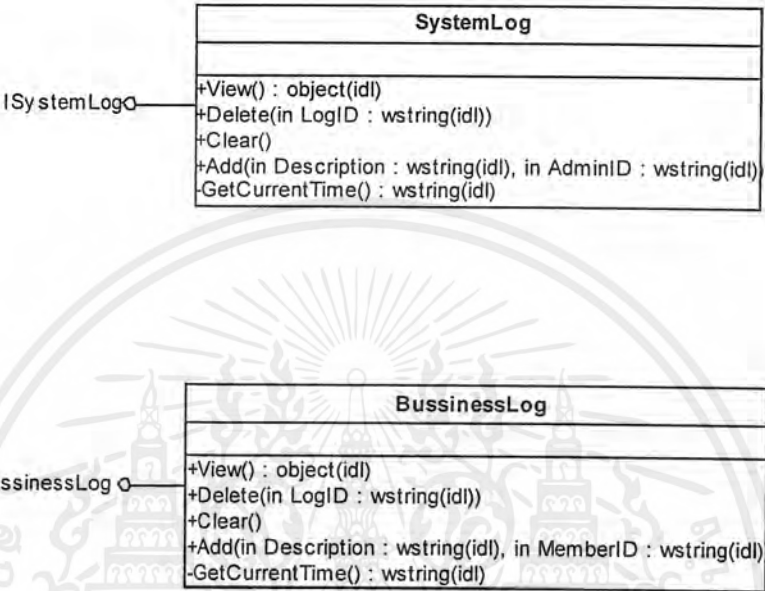


คลาสไดอะแกรม แสดง คลาสในระบบ เก็บสถิติการทำงาน ของ ระบบ OlalaTour

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

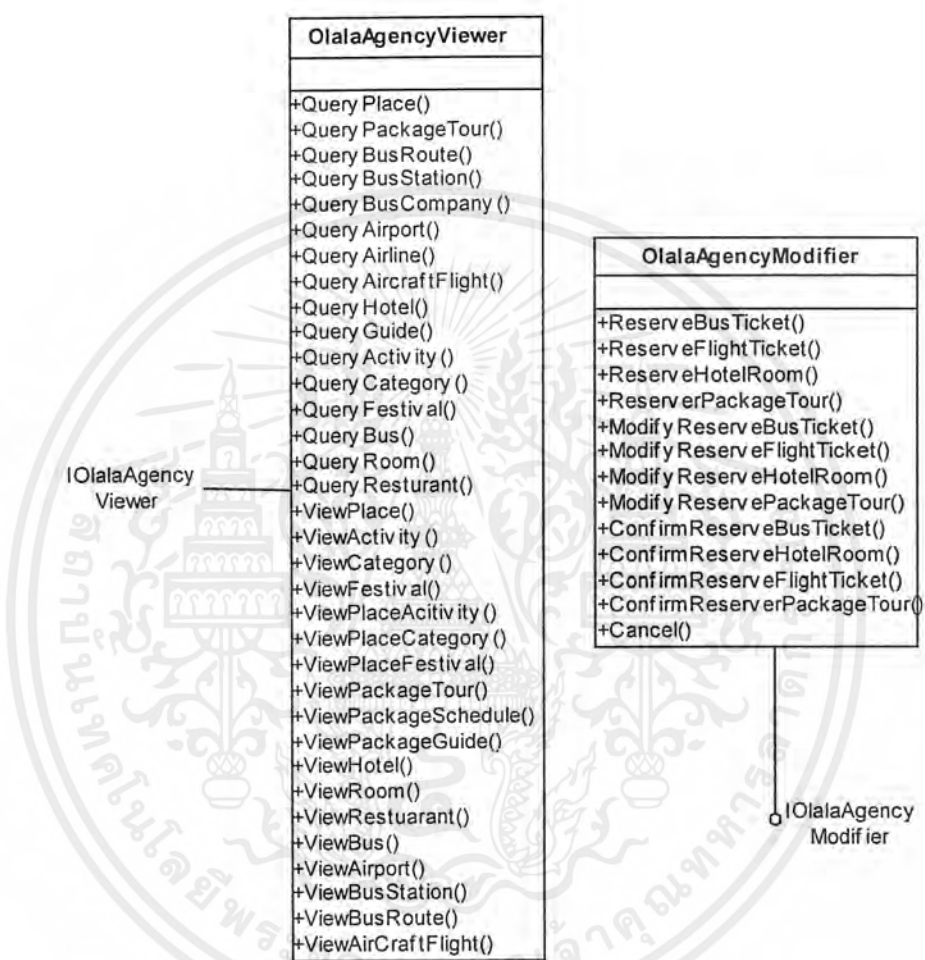
6.4 คลาสไดอะแกรม (Class Diagram) ประกอบด้วยระบบแต่ละส่วนดังนี้

Log System



คลาสดิอะแกรม แสดง คลาสในระบบ เก็บสถิติการทำงาน ของ ระบบ OlalaTour

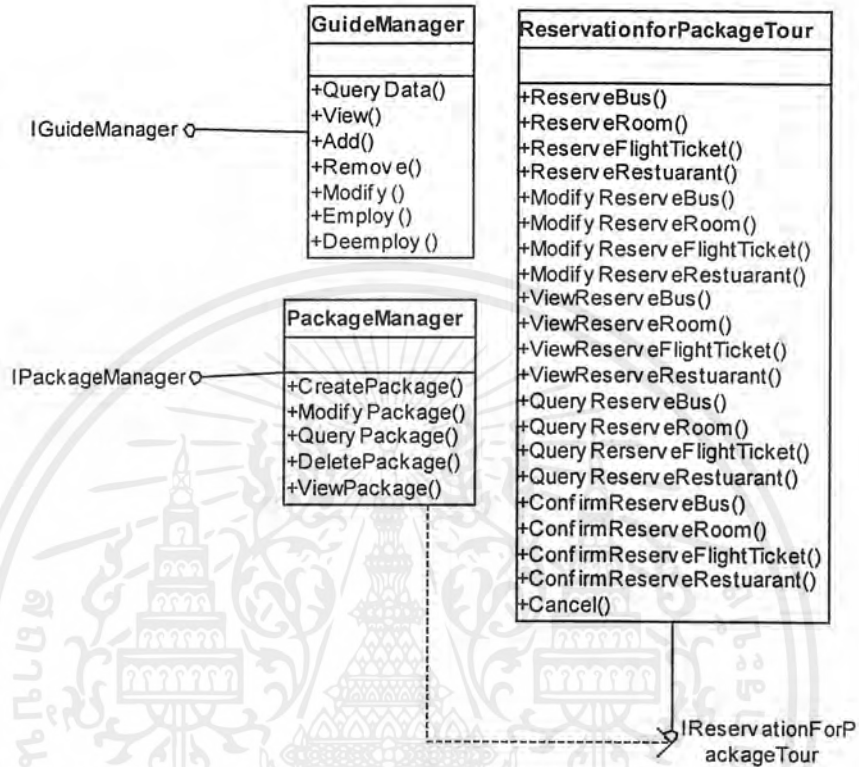
OlalaAgency System



คลาสไลอะแกรมแสดงคลาสที่ติดต่อกับผู้ใช้งานนอกทำหน้าที่คอยให้บริการงานต่าง ๆ ของระบบ OlalaTour

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

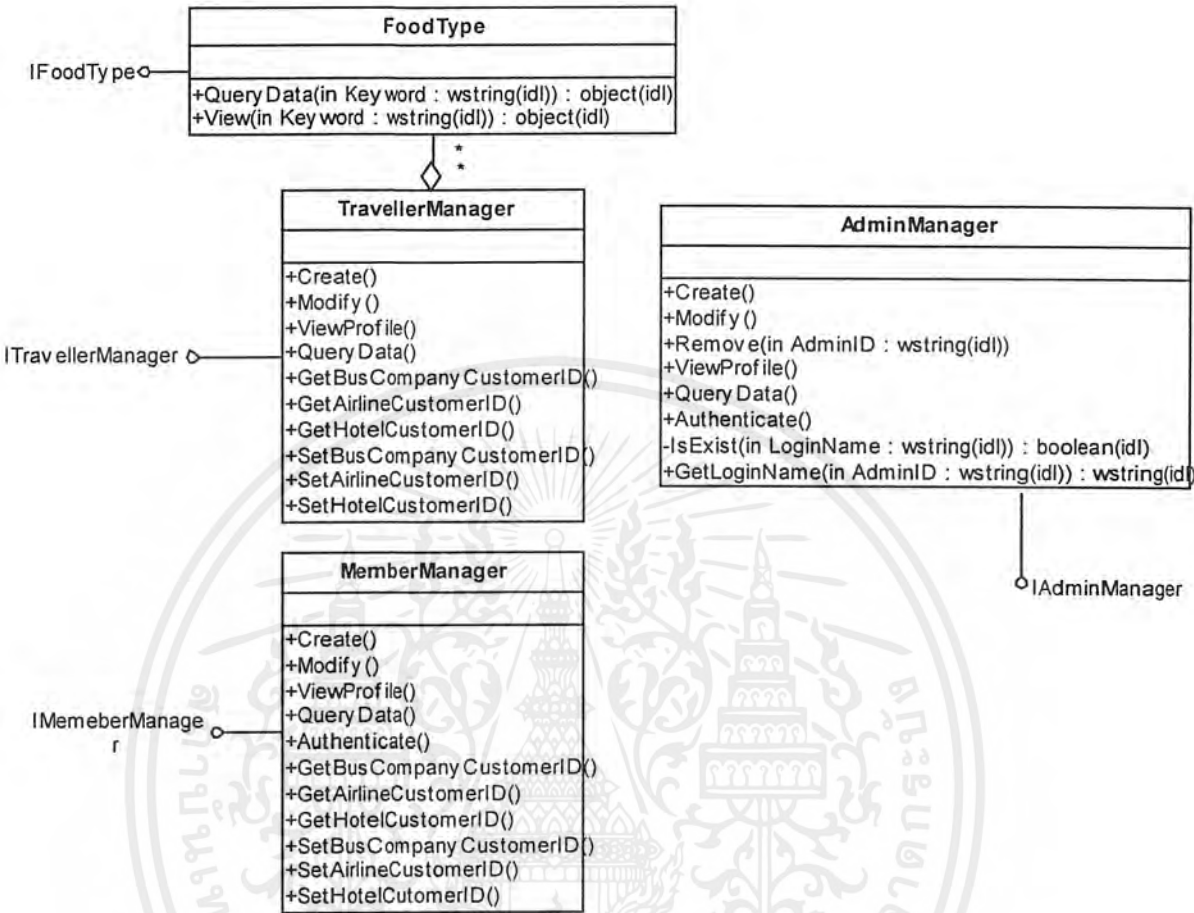
PackageSystem



คลาสในส่วนแพ็คเกจทำงานในส่วนจัดการ แพ็คเกจทัวร์ สร้าง, เปลี่ยนแปลง แพ็คเกจทัวร์
ทำการจอง พาหนะ ที่พัก ที่ใช้ในแพ็คเกจทัวร์นั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User System



คลาสในส่วนของผู้ใช้จะจัดการเกี่ยวกับข้อมูลลูกค้าและผู้เดินทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 หน้าเวบการทำงานของโอลาล่าทัวร์มีดังนี้

The screenshot shows the Olala Tour website interface. At the top, the text "OLALA TOUR" is prominently displayed. Below it, there are four numbered boxes (1, 2, 3, 4) each pointing to a different booking option: "Airline Booking", "Bus Booking", "Hotel Booking", and "Package Booking". To the right of these options is a "View Cart" button. Below the booking options, there is a section titled "HotPackage!" featuring two tour packages. The first package is "CLASSIC GOLDEN TRIANGLE TOUR" with a description of a 10-day trip in North Thailand and a price range starting at Thai Baht 11,950. The second package is "15 DAYS AUSTRALIA AND NEW ZEALAND ISLAND" with a description of a two-week adventure and a price of \$1,699.00. The background of the website features a large, faint watermark of a Thai royal seal.

รูปที่1 Olalatour

หน้าเว็บจะประกอบไปด้วย 4 ส่วน คือ การจองตั๋วเครื่องบิน,การจองตัวรถบัส,การจองโรงแรม และการจองแพ็คเกจทัวร์ โดยถ้าคลิกที่แต่ละหมายเลขจะปรากฏหน้าเว็บต่างๆดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข1 จะ แสดงหน้าเว็บ AirlineSearch

OLALA TOUR




[Airline Booking](#)


[Bus Booking](#)


[Hotel Booking](#)


[Package Booking](#)

Fill Data For Airline Search

How many of passengers? *

Total number of travellers?

Where would you like to go?*

Leaving From
 Going To

You can fly on any day that lowest fare? If to stop here
 By clicking "Show me Airlines", you indicate that you agree to the following

When would you like to travel?

Departing

Passenger and Pricing Information

Airline Class

☐ Show me non-stop flight first
☐ can smoking

What airlines do you like prefer?

Airlines Name

What seat position do you want?

Seat Position

รูปที่ 2 AirlineSearch

เมื่อทำการระบุข้อมูลในการค้นหาแล้วจึงคลิกที่ปุ่ม ShowMeAirSeat ก็จะปรากฏหน้าเว็บรายการที่
 ค้นหาดังนี้

OLALA TOUR


 Airline Booking


 Bus Booking


 Hotel Booking


 Package Booking

DEPART FLIGHT

 add to cart	Flight: flight1 AirlineName : aircraftname1 AircraftName : aircraftname1
	From: id1 3900-06-05 00 00 00 0
	To: id1ToPlace 3900-06-05 00 00 00 0
	ClassName : classname1
	TotalDistance : 500
	Description : null
	Price/Seat : 5000 0
 add to cart	Flight: flight2 AirlineName : aircraftname2 AircraftName : aircraftname2
	From: id1 3900-06-05 00 00 00 0
	To: id1ToPlace 3900-06-05 00 00 00 0
	ClassName : classname2
	TotalDistance : 500
	Description : null
	Price/Seat : 5000 0
 add to cart	Flight: flight3 AirlineName : aircraftname3 AircraftName : aircraftname3
	From: id1 3900-06-05 00 00 00 0
	To: id1ToPlace 3900-06-05 00 00 00 0
	ClassName : classname3
	TotalDistance : 500
	Description : null

รูปที่ 3 AirlineSearchList

หลังจากนั้นทำการเลือก Flight ที่ต้องการและทำการ add to cart รายการที่ต้องการก็จะไปอยู่ใน cart ดังจะแสดงรูป cart ต่อไปในส่วนของการจอง

หมายเลข 2 จะแสดงหน้าเว็บ BusSearch

OLALA TOUR

[View Cart](#)

Airline Bookings

Bus Bookings

Hotel Bookings

Passport Bookings

Fill Data For Bus Search

How many of passengers? *

Total number of travelDays?

Where would you like to go?(Internal Country) *

From

To

You can go on any day that lowest fare? If so, stop here
By clicking "Show the Buses", you indicate that you agree to the following

When would you like to go?

Departing

What bus company do you like prefer?

company

Optional

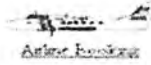
Bus Type

Seat Position

รูปที่ 4 BusSearch

เมื่อทำการระบุข้อมูลในการค้นหาแล้วจึงคลิกที่ปุ่ม ShowMeAirBus ก็จะปรากฏหน้าเว็บรายการที่ค้นหา ดังนี้

OLALA TOUR

 View Cart


result of bus search



Route: 1 company: company1
 From: id1 departure: 3901-06-10 00:00:00.0
 To: id1 arrivaltime: 3901-06-10 05:00:00.0
 class: class1
 Price/Seat: 300.0



Route: 2 company: company2
 From: id1 departure: 3901-06-10 00:00:00.0
 To: id1 arrivaltime: 3901-06-10 05:00:00.0
 class: class2
 Price/Seat: 320.0



Route: 3 company: company3
 From: id1 departure: 3901-06-10 00:00:00.0
 To: id1 arrivaltime: 3901-06-10 05:00:00.0
 class: class3
 Price/Seat: 340.0



Route: 4 company: company4
 From: id1 departure: 3901-06-10 00:00:00.0
 To: id1 arrivaltime: 3901-06-10 05:00:00.0
 class: class4
 Price/Seat: 356.0

รูปที่ 5 BusSearchList

หลังจากนั้นทำการเลือก Route ที่ต้องการและทำการ add to cart รายการที่ต้องการก็จะไปอยู่ใน cart ดังจะแสดงรูป cart ต่อไปในส่วนของการจอง

หมายเลข 3 จะแสดงหน้าเว็บ HotelSearch



รูปที่ 6 HotelSearch

เมื่อทำการระบุข้อมูลในการค้นหาแล้วจึงคลิกที่ปุ่ม ShowHotel ก็จะปรากฏหน้าเว็บรายการที่ค้นหา มาที่ละโรงแรมดังนี้



OLALA TOUR		ViewCart
Airline Booking	Bus Booking	Hotel Booking
Package Booking		
logo1 name1 10 4		
 ReserveRoom	roomClass = 14 numberOfSingleBed = 11, numberOfCoupleBed = 12, numberOfMaximumGuest = 13 this.description = des1 this.pricePerDay = 500.0	
 ReserveRoom	roomClass = 24 numberOfSingleBed = 21, numberOfCoupleBed = 22, numberOfMaximumGuest = 23 this.description = des2 this.pricePerDay = 550.0	
 ReserveRoom	roomClass = 34 numberOfSingleBed = 31, numberOfCoupleBed = 32, numberOfMaximumGuest = 33 this.description = des3 this.pricePerDay = 600.0	
 ReserveRoom	roomClass = 44 numberOfSingleBed = 41, numberOfCoupleBed = 42, numberOfMaximumGuest = 43 this.description = des4 this.pricePerDay = 700.0	

รูปที่ 7 HotelSearchList

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข 4 จะแสดงหน้าเว็บ PackageSearch

OLALA TOUR



ViewCart



Airline Booking



Bus Booking



Hotel Booking



Package Booking

Fill Data For Package Search

Where would you like to go?

Place Name :

Country :

State :

Province :

When would you like to travel?

Duration

1 ▾ Days

Departing :

9 ▾ May ▾ 2001 ▾

Passenger and Pricing Information

Price Range :

ShowMePackages

รูปที่ 8 PackageSearch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการระบุข้อมูลในการค้นหาแล้วจึงคลิกที่ปุ่ม ShowMePackage ก็จะปรากฏหน้าเว็บรายการที่ค้นหาดังนี้



รูปที่ 9 PackageSearchList

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10

บทวิจารณ์และสรุปผล

10.1 สรุปผลการดำเนินงาน

จากที่ได้ทำการออกแบบและวางโครงสร้างให้กับโปรแกรมที่ใช้งานในระบบ ซึ่งประกอบด้วยการทำงานที่เห็นได้ชัด คือ ส่วนของการจัดการและเรียกใช้เนื้อหา การทำงานของเซิร์ฟเวอร์พบว่าการพัฒนาจัดสร้างโปรแกรมได้ทำไปทุกการทำงานที่ได้วางไว้ แม้ว่าโครงสร้างปลีกย่อยจะได้ถูกเปลี่ยนแปลงเพื่อให้เหมาะสมกับการทำงานจริงในขณะทำการพัฒนาโปรเจกต์ไปบ้าง โดยสำหรับผลการทำงานของโปรแกรมหลังจากได้นำไปทดลองใช้จริงนั้น พบว่าในระยะแรกของการใช้งานจะมีข้อผิดพลาดของโปรแกรมบางส่วนถูกแสดงออกมา การพัฒนาในระยะหลังจึงเกี่ยวข้องกับการทำให้ข้อผิดพลาดของโปรแกรมเหล่านี้นหมดไป เนื่องจากจุดประสงค์ของการสร้างโปรแกรม คือ การทำให้การทำงานที่ได้ออกแบบไว้สามารถทำงานที่ได้ออกแบบไว้ออกมาถูกต้อง เพราะฉะนั้นการนำโปรแกรมออกทดลองใช้งานจึงเป็นส่วนสำคัญที่ทำให้พบข้อผิดพลาดที่อาจจะไม่สามารถพบได้ขณะเขียนโปรแกรม โดยหลังจากทำการแก้ไขข้อผิดพลาดที่พบเจอแล้ว โปรแกรมที่สร้างขึ้นในโครงงานก็สามารถทำงานตามที่ได้ออกแบบไว้ได้อย่างถูกต้อง

สำหรับปัญหาที่พบในขณะทำการพัฒนาโปรแกรมนั้น นอกเหนือจากข้อผิดพลาดของการทำงานซึ่งเป็นเรื่องปกติที่ต้องพบเจอในขณะทำการเขียนโปรแกรมแล้ว มีปัญหาหลักอีกส่วนที่ผู้คิดจะจัดสร้างโปรแกรมเพื่อทำธุรกรรมอิเล็กทรอนิกส์ควรคำนึงถึง นั่นก็คือ การออกแบบโครงสร้างของระบบงานให้ชัดเจนก่อนลงมือทำ โดยควรจบรวมรวมข้อมูลของฟังก์ชันการทำงานต่าง ๆ ที่จำเป็นทั้งหมด และออกแบบโครงสร้างของการทำงานเหล่านี้ให้ชัดเจน แต่ก็มีคามยืดหยุ่นพอที่หลังจากทำการเขียนโปรแกรมได้ง่ายโดยไม่ส่งผลกระทบต่อโปรแกรมส่วนอื่นที่ไม่เกี่ยวข้อง ซึ่งถ้าโครงสร้างของโปรแกรมมีความชัดเจนมากเท่าไร ก็จะส่งผลให้การพัฒนาโปรแกรมมีความชัดเจนมากขึ้น และมีข้อผิดพลาดลดลง

การนำโปรแกรมตามที่ได้จัดสร้างขึ้นไปใช้งานนั้น ผู้นำไปใช้สามารถใช้งานได้เลย ตามฟังก์ชันการทำงานที่ได้อธิบายไว้ โดยไม่จำเป็นต้องเปลี่ยนแปลงตัวโปรแกรมแต่อย่างใด เนื่องจากการออกแบบโปรแกรมให้สามารถติดตั้งได้สะดวก โดยข้อดีของโปรแกรมนี้นี้ก็คือแนวคิดที่แตกต่างจากการทำธุรกรรมทางอิเล็กทรอนิกส์รูปแบบอื่นออกไป นั่นคือ การนำเทคโนโลยีของเจทูอีเข้ามาใช้งาน ซึ่งเป็นเทคโนโลยีที่ยังไม่ค่อยมีคนนำไปใช้ในการพัฒนาระบบ ส่วนนี้จึงนับว่าเป็นจุดที่น่าสนใจที่จะสามารถนำมาทดสอบการทำงานและประสิทธิภาพของระบบที่พัฒนาขึ้นด้วยเทคโนโลยีแบบนี้

11.2 แนวทางการพัฒนาต่อ

จากโปรแกรมการทำงานที่ได้พัฒนาขึ้น สามารถถูกนำไปพัฒนาต่อได้อีกหลายทาง สุดแล้วแต่แนวความคิดของผู้พัฒนาต่อไป เนื่องจากตัวโปรแกรมนั้นเปรียบเสมือนส่วนหลักที่มีการทำงานที่จำเป็นไว้ครบถ้วนแล้ว การทำให้โปรแกรมมีความสมบูรณ์และสามารถก้าวทันกับเทคโนโลยีของ เจทูอี จึงสามารถทำได้โดยง่าย ผู้พัฒนาจำเป็นต้องศึกษาโครงสร้างของโปรแกรมที่มีอยู่โดยละเอียด แล้วจึงเพิ่มการทำงานส่วน

อื่นลงไปให้เหมาะสม ซึ่งการทำงานเหล่านี้มีตัวอย่าง เช่น การเพิ่มการทำงานในส่วนของ JavaMail หรือในส่วนของ Extensive Markup Language (XML) ซึ่งเป็นเทคโนโลยีที่สามารถรองรับการทำงานได้ในเจทูอี และมิประโยชน์ในด้านทำให้ผู้ใช้สามารถสร้างและดูแลเอกสารที่มีโครงสร้าง (structured documents) ที่บรรจุตัวอักษร (plain text) โดยทำให้สามารถ rendered หรือปรับเปลี่ยนการแสดงผลในรูปแบบที่หลากหลาย จุดประสงค์หลักของ XML คือการแยกส่วน ข้อมูลเพื่อประโยชน์ในการแสดงผลได้นอกเหนือจากนี้ยังสามารถนำเอาเทคโนโลยีทางด้านของแอปพลิเคชันเซิร์ฟเวอร์หรือเว็บเซิร์ฟเวอร์ที่มีการพัฒนาออกมาใหม่ ๆ เพื่อนำมาทดลองใช้งานกับโปรแกรมของระบบที่ได้สร้างขึ้นได้เนื่องจากโปรแกรมในโครงการนั้นใช้หลักการเชิงคอมโพเนนต์ทำให้ง่ายต่อการนำไปใช้เป็นอย่างมาก ซึ่งหลังจากการทดลองนั้นแล้วเราอาจจะเปรียบเทียบผลการทำงานกับระบบเดิมที่เคยใช้ได้ เพื่อเป็นแนวทางในการที่ผู้พัฒนาต่อสามารถนำไปใช้ได้ ทำให้จุดประสงค์ในการทำงานนี้มีความถูกต้องและสมบูรณ์มากยิ่งขึ้นไปอีก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก ความต้องการด้านทรัพยากรของระบบ

เซิร์ฟเวอร์

ระบบปฏิบัติการวินโดวส์ 2000 แอดวานซ์เซิร์ฟเวอร์ (Windows 2000 Advance Server)

- ซีพียู (CPU) ในขั้นต่ำคือรุ่นเพนเทียม 166 หรือสูงกว่านั้น ซึ่งที่แนะนำควรเป็นเพนเทียม II ขึ้นไปจะดีกว่า
- แรม (RAM) ในขั้นต่ำอยู่ที่ 128 MB หรือสูงกว่า เพื่อให้ทำงานในระบบเครือข่ายได้อย่างมีประสิทธิภาพ
- เนื้อที่ฮาร์ดดิสก์ อย่างน้อย 2 GB โดยประมาณและมีเนื้อที่ว่างอยู่อย่างน้อย 900 MB – 1 GB หรือมากกว่า

ไคลเอ็นท์

ระบบปฏิบัติการวินโดวส์ 2000 โพรเฟสชันนัล (Windows 2000 professional)

- ซีพียู (CPU) ในขั้นต่ำคือ รุ่นเพนเทียมของอินเทลหรือซีพียูของเอเอ็มดี เช่น เอเอ็มดี K6
- แรม (RAM) ในขั้นต่ำอยู่ที่ 64 MB
- เนื้อที่ฮาร์ดดิสก์ อย่างน้อย 1 GB โดยประมาณและมีเนื้อที่ว่างอยู่อย่างน้อย 700-800 MB

ระบบปฏิบัติการ วินโดวส์ 98 (Windows 98)

- ซีพียู (CPU) ในขั้นต่ำคือ เพนเทียม 166 ขึ้นไปและถ้าเป็นรุ่นเอ็มเอ็มเอ็กซ์ก็จะได้
- แรม (RAM) ในขั้นต่ำอยู่ที่ 32 MB
- เนื้อที่ฮาร์ดดิสก์ตั้งแต่ 1 GB ขึ้นไปและมีเนื้อที่ว่าง 120 MB ขึ้นไป

Software Requirement

1. ระบบปฏิบัติการ วินโดวส์ 2000 (Windows 2000)
 - ระบบปฏิบัติการ วินโดวส์ 2000 แอดวานซ์เซิร์ฟเวอร์ (Windows 2000 Advance server)
 - ระบบปฏิบัติการ วินโดวส์ 2000 โพรเฟสชันนัล (Windows 2000 professional)
 - ระบบปฏิบัติการ วินโดวส์ 98 (Windows 98)
2. Java 2 SDK Enterprise Edition
3. Java 2 SDK 1.3 Standard Edition
4. Jbuilder 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข การติดตั้งและการกำหนดค่าเริ่มต้นของซอฟต์แวร์เพื่อการสร้างและพัฒนาโปรแกรม

1. ระบบปฏิบัติการวินโดวส์

1.1 ระบบปฏิบัติการวินโดวส์ 2000 (Windows 2000)

วินโดวส์ 2000 เป็นระบบปฏิบัติการที่พัฒนามาจาก วินโดวส์ เอ็นที 4.0 (Windows NT 4.0) โดยหัวใจหลักของระบบปฏิบัตินั้นเป็นส่วนที่ได้รับอิทธิพลมาจาก วินโดวส์ เอ็นที 4.0 โดยที่มีส่วนที่ใช้ติดต่อกับผู้ใช้หรือ ยูสเซอร์ อินเตอร์เฟส (User Interface) ของวินโดวส์ 2000 นั้นจะใกล้เคียงกับ ยูสเซอร์ อินเตอร์เฟส ของ วินโดวส์ 98 (Windows 98) มากกว่า การออกแบบวินโดวส์ 2000 นั้นจะยึดตามแนวทางดังนี้

- พัฒนาต่อจากรากฐานของควมมีเสถียรภาพของและศักยภาพต่าง ๆ ของ วินโดวส์ เอ็นที 4.0
- ผสมผสานลักษณะที่ดีของ วินโดวส์ 98
- ใช้งานได้ง่ายและสะดวกมากกว่าวินโดวส์เวอร์ชันก่อน
- บริหารระบบง่ายและค่าใช้จ่ายในการดูแลระบบต่ำ
- สามารถรองรับเทคโนโลยีใหม่ ๆ ได้เป็นอย่างดีทั้งในด้านระบบเครือข่ายอินเทอร์เน็ต ฐานข้อมูล และการพัฒนาแอปพลิเคชัน

การตรวจสอบก่อนการติดตั้ง

1. การตรวจสอบฮาร์ดแวร์รายชื่อ HCL(Hardware Compatibility List) ซึ่งเป็นรายการชื่ออุปกรณ์ฮาร์ดแวร์ต่าง ๆ ที่ วินโดวส์ 2000 รู้จักและสามารถทำงานอยู่ภายใต้ระบบปฏิบัติการวินโดวส์ 2000 ได้
2. การกำหนดดิสก์พาร์ติชันและระบบไฟล์
 - FAT 32
เป็นระบบไฟล์ที่ใช้สำหรับวินโดวส์ 95 โอเอสอาร์ทู (Windows 95 OSR2) วินโดวส์ 98 และ วินโดวส์ 2000
 - NTFS เวอร์ชัน 5
ระบบไฟล์ NTFS ของวินโดวส์ 2000 นั้นไม่เหมือนกับ NTFS ของ วินโดวส์ เอ็นที 4.0 โดย NTFS ของวินโดวส์ 2000 จะได้รับการพัฒนาและปรับปรุงประสิทธิภาพให้ดีกว่า อย่างเช่น Dynamic Volume Management , File and Folder Security , Disk Quota รวมถึง Encryption File System
3. การเป็นสมาชิกของเวิร์กกรุ๊ป หรือของโดเมน

3.1 วินโดวส์ 2000 เซิร์ฟเวอร์

วินโดวส์ 2000 เซิร์ฟเวอร์นั้น มันสามารถทำงานได้หลายหน้าที่ในระบบเครือข่ายหนึ่ง ๆ ทั้งนี้จะขึ้นอยู่กับความต้องการของผู้ติดตั้งหรือผู้ดูแลระบบ ซึ่งหน้าที่ดังกล่าวก็ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วินโดวส์ 2000 เซิร์ฟเวอร์ ที่ทำหน้าที่เป็นโดเมนคอนโทรลเลอร์
 - วินโดวส์ 2000 เซิร์ฟเวอร์ ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ธรรมดาโดยเข้าเป็นสมาชิกของโดเมน เรียกว่า เมมเบอร์ เซิร์ฟเวอร์ (Member server)
 - วินโดวส์ 2000 เซิร์ฟเวอร์ ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ธรรมดาภายใต้เวิร์กกรุ๊ปเล็ก ๆ โดยไม่ยุ่งเกี่ยวกับสมาชิกของโดเมนใด เรียกว่า สแตนด์ ออลน เซิร์ฟเวอร์ (Stand Alone Server)
- สำหรับในโครงการนี้ เราใช้ วินโดวส์ 2000 เซิร์ฟเวอร์ ทำหน้าที่เป็นสแตนด์ ออลน เซิร์ฟเวอร์ เพื่อให้มันทำงานเฉพาะทางได้อย่างเต็มที่

3.2 วินโดวส์ 2000 โปรเฟสชันนัล

วินโดวส์ 2000 โปรเฟสชันนัล นั้น ได้ถูกจัดวางตำแหน่งให้ทำหน้าที่เป็นเครื่องไคลเอนต์ในระบบเครือข่ายอยู่แล้ว ซึ่งเราเพียงแค่นำเซิร์ฟเวอร์กรุปของเครื่องขณะติดตั้งเท่านั้น ในกรณีที่ติดตั้ง วินโดวส์ 2000 โปรเฟสชันนัล บนเครื่องคอมพิวเตอร์ธรรมดาที่บ้านหรือเครื่องใช้ส่วนตัวที่ไม่มีเน็ตเวิร์กการ์ดเสียบอยู่ โปรแกรมติดตั้งก็จะแสดงไดอะล็อกบ็อกพร้อมทั้งข้อความแจ้งให้ทราบว่ามันกำลังติดตั้งองค์ประกอบต่าง ๆ ของเน็ตเวิร์กอยู่ เพื่อเอาไว้สำหรับให้ผู้ใช้เชื่อมต่อกับเครือข่ายทางเน็ตเวิร์กในภายหลัง

1.2 ระบบปฏิบัติการวินโดวส์ 98 (Windows 98)

วินโดวส์ 98 เป็นระบบปฏิบัติการกึ่ง 32 บิต มีเสถียรภาพในการทำงานสูงกว่าวินโดวส์ 95 การจัดระบบไฟล์ในวินโดวส์ 98 แบบ FAT 32 จะทำให้กลัสดรฟ์ มีขนาดเล็กกว่าการจัดระบบไฟล์แบบ FAT16 ที่ใช้ในวินโดวส์ 95 นอกจากนี้วินโดวส์ 98 ยังสนับสนุนการใช้งานพอร์ตยูเอสบี (USB) ซึ่งเป็นพอร์ตที่สนับสนุนในด้านการเชื่อมต่ออุปกรณ์ต่าง ๆ เข้ากับพีซี เมื่อใช้งานความสามารถด้านมัลติมีเดียบนเครื่องที่ใช้ซีพียู MMX ก็จะได้ภาพที่สวยงามและสมจริงมากขึ้น แต่ในส่วนของการรักษาความปลอดภัยนั้นว่ายังด้อยอยู่มากเมื่อเทียบกับ วินโดวส์ 2000 โปรเฟสชันนัล ซึ่งถูกใช้เพื่อเป็นไคลเอนต์เหมือนกัน เนื่องจาก วินโดวส์ 2000 โปรเฟสชันนัล มีการพัฒนามาจาก วินโดวส์ เอ็นที แต่ วินโดวส์ 98 นั้นพัฒนามาจากวินโดวส์ 95 หรือกล่าวได้ว่าเป็น วินโดวส์ 95 ที่แก้บั๊กแล้วนั่นเอง

2. Java 2 SDK Enterprise Edition

ทำการดาวน์โหลดโปรแกรม Java 2 SDK Enterprise Edition ได้จากเว็บไซต์

<http://java.sun.com/j2ee/j2sdkee/>

ทำการติดตั้งโปรแกรมจากไฟล์ที่ดาวน์โหลดเรียบร้อยแล้วให้ทำการเซต PATH และ CLASSPATH ดังนี้

```
PATH= %j2ee_Home%\bin
```

```
CLASSPATH= %j2ee_Home%\lib\j2ee.jar
```

3. Java 2 SDK 1.3 Standard Edition

ทำการดาวน์โหลดโปรแกรม Java 2 SDK 1.3 Standard Edition ได้จากเว็บไซต์

<http://java.sun.com/j2se/1.3/download-windows.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการติดตั้งโปรแกรมจากไฟล์ที่ดาวน์โหลดเรียบร้อยแล้วให้ทำการเซต PATH และ CLASSPATH ดังนี้

PATH= %j2sdk_Home%\bin

CLASSPATH= %j2sdk_Home%\lib\dt.jar; %j2sdk_Home%\lib\tools.jar

ในการเซต PATH และ CLASSPATH นั้นสามารถเซตจากการแก้ไขที่ไฟล์ Autoexec.BAT หรือเซตค่าที่ Environment Variables ซึ่งอยู่ใน แถบ Advanced ใน System Properties ก็ได้เช่นกันโดยสามารถเซตจาก Autoexec.BAT หรือเซตที่ Environment Variables ซึ่งอยู่ใน แถบ Advanced ใน System Properties ก็ได้เช่นกัน

4. Jbuilder เวอร์ชัน 4.0

Jbuilder เป็นโปรแกรมที่ช่วยในการพัฒนาแอปพลิเคชันที่เขียนด้วยภาษาจาวา ไม่ว่าจะเป็น Applet JSP/Servlets ,JavaBean และ Enterprise Java Bean รวมไปถึง J2EE Applications ที่เขียนสำหรับ The Java 2 Platform ด้วย

เมื่อติดตั้ง Jbuilder 4.0 จากแผ่นโปรแกรมเรียบร้อยแล้วให้ทำการเซต PATH ดังนี้

PATH= C:\JBuilder4\bin

จากตัวอย่างนั้น โปรแกรม Jbuilder 4.0 นั้นได้ติดตั้งที่ Drive C ที่ใดเรทอริ Jbuilder4 หากผู้ติดตั้งหรือผู้ดูแลระบบได้ติดตั้งที่ Drive อื่นก็เซตให้ถูกต้อง

ภาคผนวก ค การเปรียบเทียบเทคโนโลยีระหว่างคอมพัส (COM Plus) กับ อีเจบี (EJB)

1. Component Type Supported

- อีเจบีจะมีคอมโพเนนท์ 2 ชนิดในอีเจบี 1.1 สเปกซิฟิเคชัน (Specification) คือเอ็นดีบีบีและเซสชันบีบี โดยเอ็นดีบีบีจะเป็นคอมโพเนนท์ที่คอยจัดการข้อมูลในตารางในฐานข้อมูลและเซสชันบีบีเป็นคอมโพเนนท์ที่ทำบิซิเนสลอจิกให้ไคลเอนท์คอยเรียกใช้บริการและติดต่อเอ็นดีบีบีเพื่อทำการใดๆกับข้อมูล
- คอมพัส จะมีคอมโพเนนท์เพียงแบบเดียวและจะแนะนำให้เขียนเป็นสเตตเลสเพื่อไม่ต้องจัดการเกี่ยวกับสเตตและเพื่อสเกลลIBILITY

2. State Management

- อีเจบี เซสชันบีบี จะแบ่งเป็นสเตตฟูล และสเตตเลส ซึ่ง สเตตเลส เซสชันบีบี จะมีความสเกลลIBILITY ที่ดีกว่าและสามารถถูกเรียกใช้ได้จากหลายไคลเอนท์พร้อมกัน และ สเตตฟูล เซสชันบีบี คอนเทนเนอร์จะเป็นคนจัดการสเตตให้โดยจะให้ไคลเอนท์ ติดต่อกับคอมโพเนนท์ที่มี สเตต และไคลเอนท์ตัวนั้นเก็บอยู่เป็นคนให้บริการ ไคลเอนท์คนนั้นต่อ แต่เนื่องจากอาจจะเกิดการพาสซีว (passivate) สเตต ของไคลเอนท์ไปเก็บใน temporary storage จึงต้องมีการนำสเตตของไคลเอนท์ขึ้นมาจาก storage ไปใส่ในคอมโพเนนท์ก่อนจะให้บริการแก่ไคลเอนท์
- คอมพัส จะมีการจัดการสเตตของคอมโพเนนท์ผ่าน share property

3. Declarative Persistence

- อีเจบี จะมี เอ็นดีบีบี ประเภทหนึ่งเรียกว่า Container-Managed Persistence เอ็นดีบีบี ซึ่งเราสามารถกำหนดว่าแอททริบิวต์ไหนบ้างของคอมโพเนนท์นั้นๆที่จะกำหนดให้เพอร์ซิส (persist) ลงฐานข้อมูล โดยไม่ต้องเขียนโค้ดเอสคิวแอลหรือเขียนเองก็ได้
- คอมพัส ต้องเขียนเอสคิวแอลติดต่อกับตารางในฐานข้อมูลเองอย่างเดียว

4. Speed

- อีเจบีจากการทดลองพบว่าการทำงานค่อนข้างช้าเมื่อเทียบกับคอมพัส ในแอปพลิเคชันเดียวกัน เนื่องจากเป็นภาษาจาวาซึ่งช้าทั้งการคอมไพล์และการรัน
- คอมพัส เป็น Native Codeb ล้วนจึงสามารถทำงานได้เร็ว

5.Thread

- อีเจบี คอนเทนเนอร์จะเป็นผู้จัดการเทรดของทุกๆคอมโพเนนท์เพียงผู้เดียว ในการเขียนโค้ด

ห้ามเขียนโค้ดที่ยุ่งเกี่ยวกับเทรด โดยเด็ดขาดตามสเปกซิฟิเคชันเพราะจะทำให้คอนเทนเนอร์ไม่สามารถจัดการเทรด ได้อย่างอิสระ

- คอมพิวเตอร์จะมีหลายเทรดโมเดลให้ใช้โดยโปรแกรมเมอร์จะต้องเป็นผู้จัดการในเรื่องของการทำงานเทรดเอง ซึ่งมีความซับซ้อนมากกว่า แต่ก็ทำให้คิดพลาคลได้ง่ายกว่า

6.Open

- อีเจบี เป็นเพียงสเปกซิฟิเคชันเท่านั้น ใครก็สามารถทำโปรดัคซ์ออกมาขายตามสเปกซิฟิเคชันนี้ได้ และคอมโพเนนท์ที่เขียนขึ้นมาสามารถนำไป ดิฟฟิอุส บนทุกๆคอนเทนเนอร์ได้เหมือนกัน ทำให้เราสามารถเลือกคอนเทนเนอร์ได้อย่างอิสระไม่ติดกับ โปรดัคซ์หนึ่งและบางโปรดัคซ์เป็นแบบ open source และ แจกฟรีทำให้นักพัฒนาสามารถรู้กลไกการทำงานทั้งหมดได้

- คอมพิวเตอร์ เป็นของไมโครซอฟท์เพียงผู้เดียวตัวคอนเทนเนอร์มีมาให้พร้อม วินโดวส์2000 ทำให้เมื่อจะใช้ คอมพิวเตอร์ ต้องใช้โอเอสเป็นวินโดวส์2000 เท่านั้น และมีคอนเทนเนอร์ของไมโครซอฟท์ เพียงบริษัทเดียว แต่ภาษาที่ใช้ในการเขียนคอมโพเนนท์ที่มีได้หลายภาษาเช่น วีบี, เคลไพท์,วีซี แต่ อีเจบี ใช้จาวาได้เพียงอย่างเดียว

ภาคผนวก ง Getting Started

ในส่วนนี้จะเป็นการอธิบายในขั้นตอนการพัฒนา ,การดีพลอย และการรันแอปพลิเคชันที่เป็น ไคล์เอ็นท์-เซิร์ฟเวอร์ ที่ใช้เอ็นเตอร์ไพร์จาวาเป็น โดยเริ่มต้นจะกำหนดให้ไคล์เอ็นท์ชื่อว่า ConverterClient และให้เอ็นเตอร์ไพร์จาวาเป็นชื่อว่า ConverterEJB โดยมีขั้นตอนการทำงานตามลำดับขั้นดังนี้

1. ทำการเขียนโค้ดเอ็นเตอร์ไพร์จาวาเป็น
2. สร้างเจทูอีแอปพลิเคชัน
3. แฝกเกิดเอ็นเตอร์ไพร์จาวาเป็น
4. ดีพลอยเจทูอีแอปพลิเคชัน
5. สร้างไคล์เอ็นท์
6. รันไคล์เอ็นท์

1. ทำการเขียนโค้ดเอ็นเตอร์ไพร์จาวาเป็น

เอ็นเตอร์ไพร์จาวาเป็นทุกๆชิ้นต้องประกอบไปด้วย

- รีโมทอินเตอร์เฟส (Remote Interface)
- โฮมอินเตอร์เฟส (Home Interface)
- เอ็นเตอร์ไพร์บีนคลาส (Enterprise bean class)

1.1 การเขียนโค้ดรีโมทอินเตอร์เฟส (Coding the Remote Interface)

ภายในรีโมทอินเตอร์เฟสจะเป็นการประกาศเมทอดที่อนุญาตให้ไคล์เอ็นท์เรียกใช้ได้ โดยเอ็นเตอร์ไพร์จาวาเป็นจะต้องทำการอิมพลีเมนต์เมทอดที่ประกาศในรีโมทอินเตอร์เฟสนี้ โค้ดภายในรีโมทอินเตอร์เฟสมีดังนี้

```
import javax.ejb.EJBObject;
import java.rmi.RemoteException;

public interface Converter extends EJBObject {

    public double dollarToYen(double dollars) throws RemoteException;
    public double yenToEuro(double yen) throws RemoteException;
}
```

1.2 การเขียนโค้ดโฮมอินเทอร์เฟซ (Home the Home Interface)

โฮมอินเทอร์เฟซจะทำการกำหนดเมทอดเพื่อให้ไคลเอนต์ทำการ create, find หรือ remove เอ็นเตอร์ไพรส์จาวาบี๋น โดยภายในจะมีเพียงเมทอด create เพียงเมทอดเดียวโดยจะรีเทอร์นเป็นอ็อบเจ็คที่เป็น type เดียวกับรีโมตอินเทอร์เฟซ โค้ดภายในโฮมอินเทอร์เฟซมีดังนี้

```
import java.io.Serializable;
import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;

public interface ConverterHome extends EJBHome {

    Converter create() throws RemoteException, CreateException;
}
```

1.3 การเขียนโค้ดเอ็นเตอร์ไพรส์จาวาบี๋นคลาส

ภายในเอ็นเตอร์ไพรส์จาวาบี๋นตัวอย่างที่ยกมานี้จะเป็น stateless session bean ที่ชื่อว่า ConverterEJB โดยภายในบี๋นนี้จะมีบิซิเนสเมทอดอยู่ 2 เมทอดที่ประกาศไว้ใน Converter รีโมตอินเทอร์เฟซ คือ dollarToTYen และ YenToEuro โดยโค้ดภายใน ConverterEJB มีดังนี้

```
import java.rmi.RemoteException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;

public class ConverterEJB implements SessionBean {

    public double dollarToYen(double dollars) {

        return dollars * 121.6000;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public double yenToEuro(double yen) {
```

```
    return yen * 0.0077;
```

```
}
```

```
public ConverterEJB() {}
```

```
public void ejbCreate() {}
```

```
public void ejbRemove() {}
```

```
public void ejbActivate() {}
```

```
public void ejbPassivate() {}
```

```
public void setSessionContext(SessionContext sc) {}
```

```
}
```

1.4 ทำการคอมไพล์ซอร์สโค้ดของเอ็นเตอร์ไพรส์จาваเป็น

ตอนนี้เมื่อสร้างทั้ง 3 ไฟล์ได้แล้วก็พร้อมที่จะทำการคอมไพล์ คือ ไฟล์โมดอินเตอร์เฟส (Converter.java) , ไฟล์โฮมอินเตอร์เฟส (ConverterHome.java) และ ไฟล์เอ็นเตอร์ไพล์บีนาคลาส (ConverterEJB.java) โดยทำการวิธีการคอมไพล์ทำดังนี้

1. สร้างไฟล์ compilerEJB.bat ขึ้นมาให้ไปอยู่ในไดเรกทอรีที่ติดตั้งเจทูอีเอสดีเค โดยภายในเขียนดังนี้

```
set J2EE_HOME=<installation-location>
```

```
set CPATH=.;%J2EE_HOME%\lib\j2ee.jar
```

```
javac -classpath %CPATH% ConverterEJB.java ConverterHome.java Converter.java
```

2 ทำการรันไฟล์ compilerEJB.bat

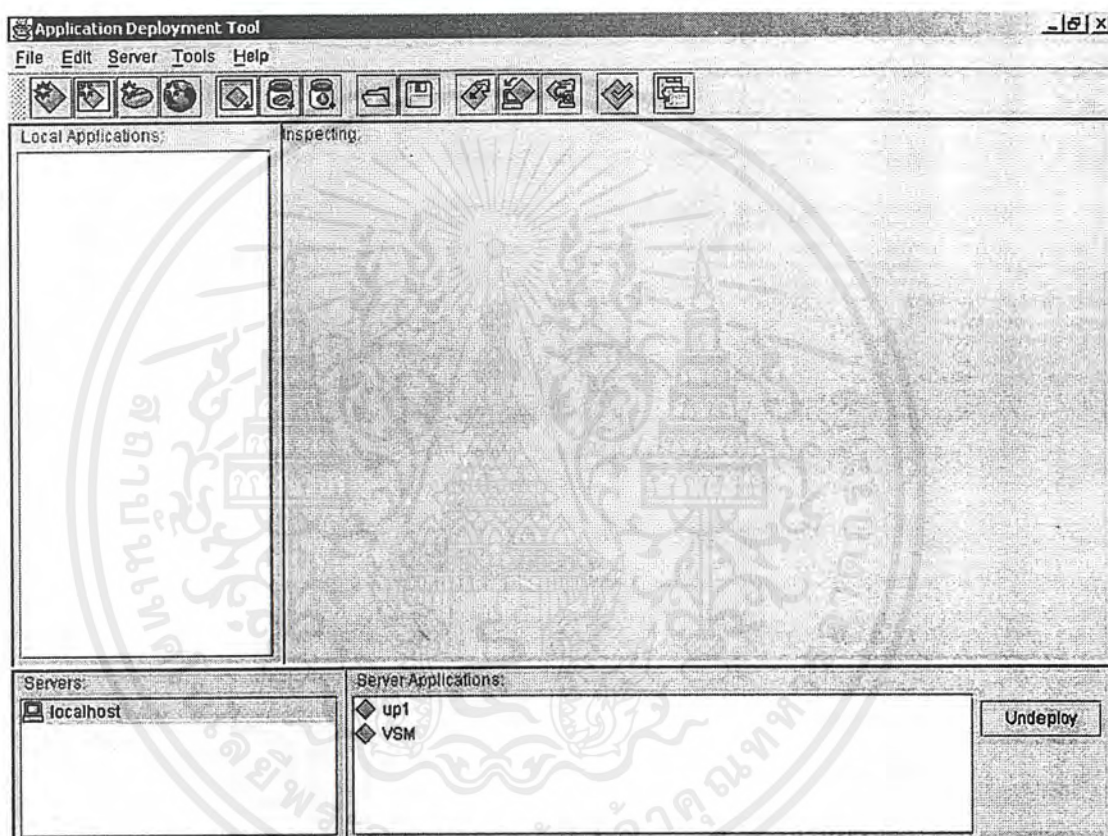
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สร้างเจทูอีแอปพลิเคชัน

คุณจะสามารถดีพลอยเอ็นเตอร์ไฟล์ขึ้นได้โดยตรงในเจทูอีเซิร์ฟเวอร์ต้องทำการ add บินเข้าไปไว้ในเจทูอี แอปพลิเคชัน จึงจะสามารถดีพลอยได้ โดยเราจะทำการสร้างแอปพลิเคชันใหม่ที่ชื่อว่า ConverterApp และเก็บไฟล์นี้ไว้ใน ConverterApp.ear

2.1 ทำการเปิดหน้าต่าง command prompt และสตาร์ทเจทูอี เซิร์ฟเวอร์ โดยพิมพ์ `j2ee -verbose`

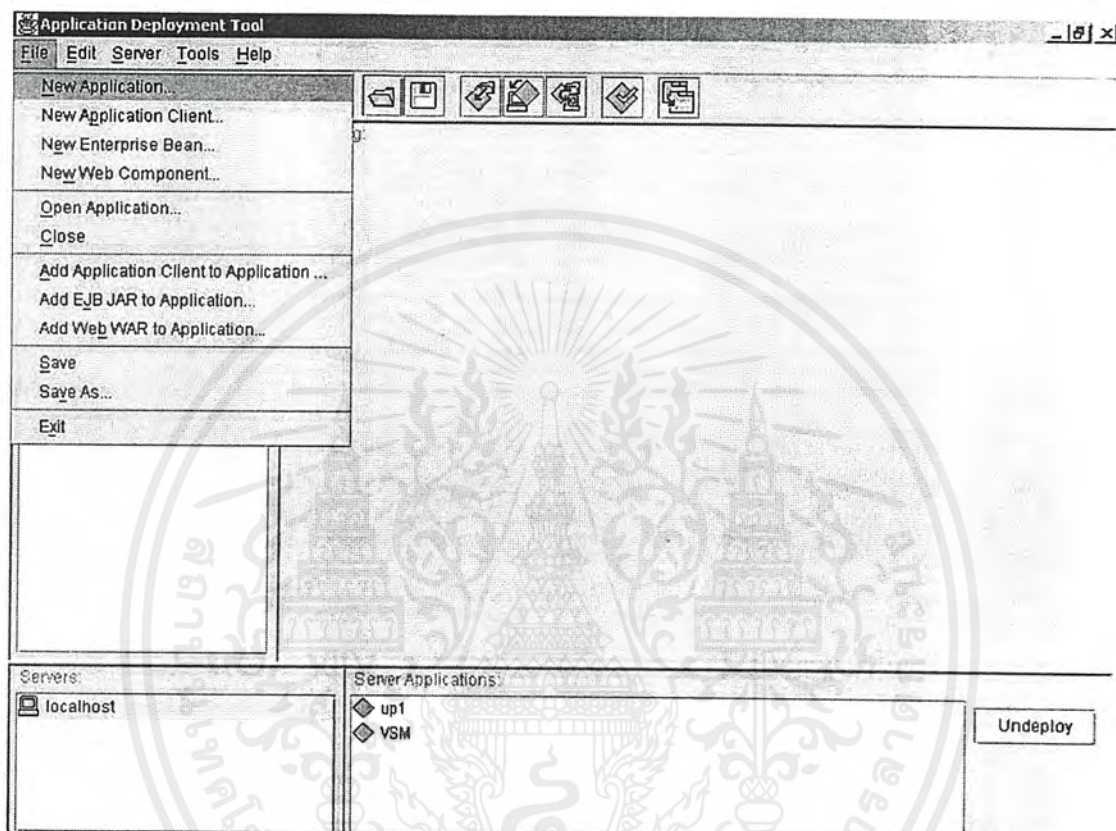
2.2 เปิด command prompt ขึ้นมาอีกหน้าต่างหนึ่งและพิมพ์ `deploytool` จะปรากฏหน้าจอดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

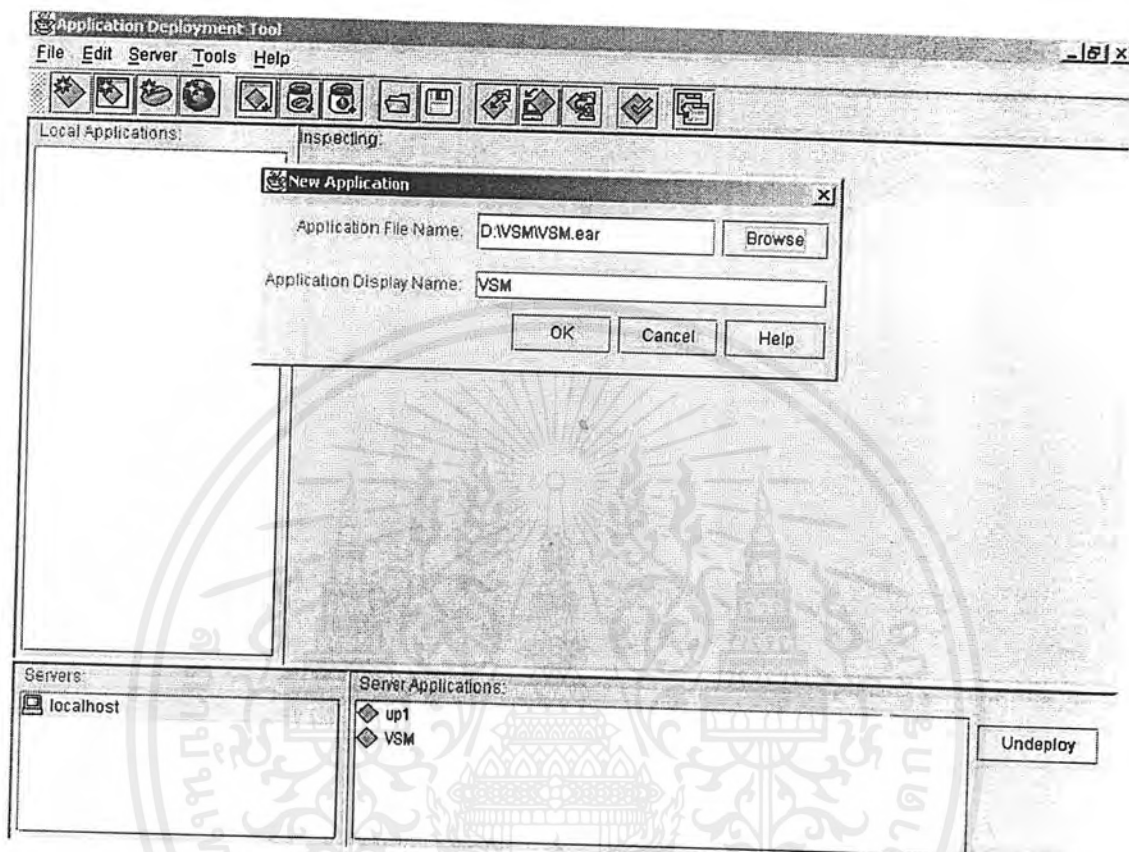
2.3 การเริ่มสร้างแอปพลิเคชัน

1. ไปที่ File>New Application



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

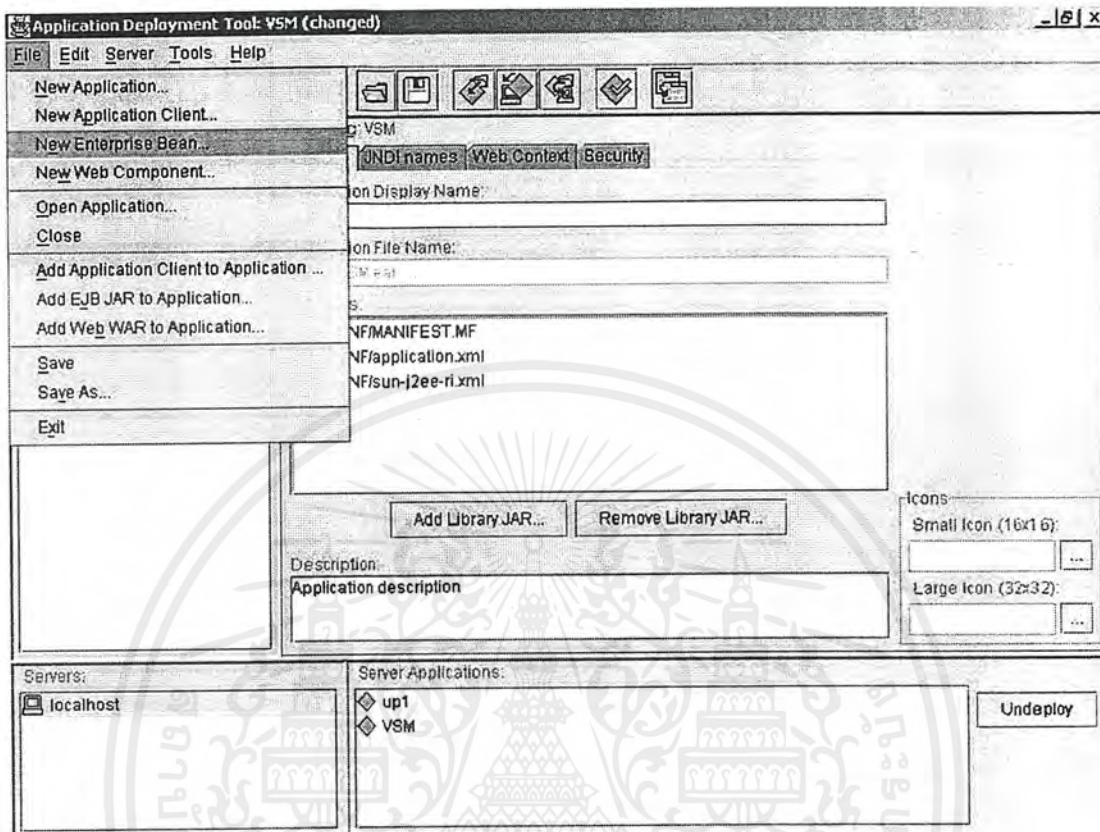
2. กำหนดที่อยู่และตั้งชื่อแอปพลิเคชัน ที่ต้องการเอาลงเซิร์ฟเวอร์ (ในที่นี้ให้ใส่ ConverterApp.ear)



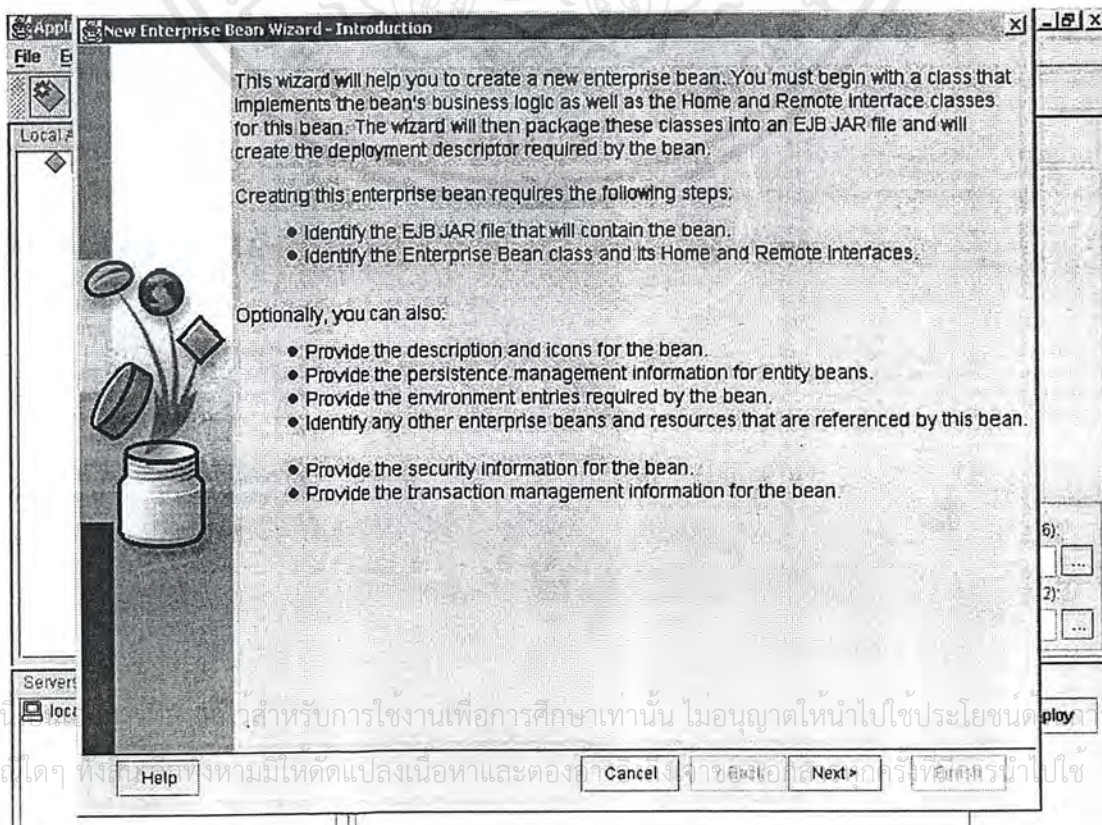
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แพ็กเก็ตเอ็นเตอร์ไพรส์จาว่าบีน

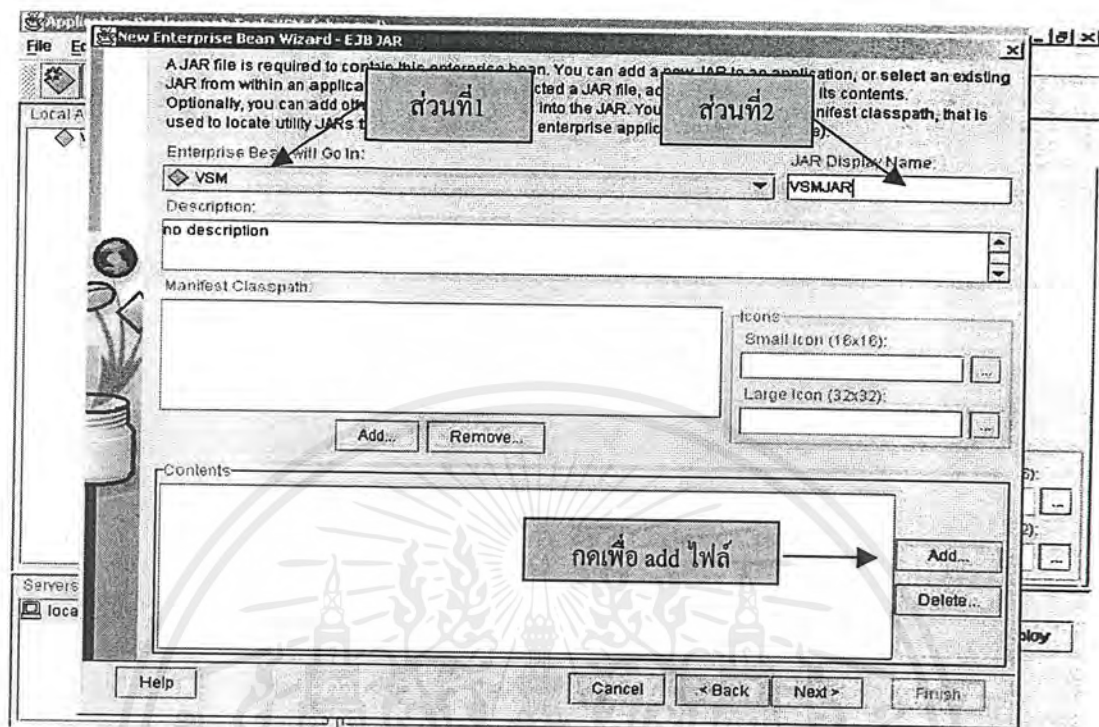
3.1 ไปที่ File>New Enterprise bean



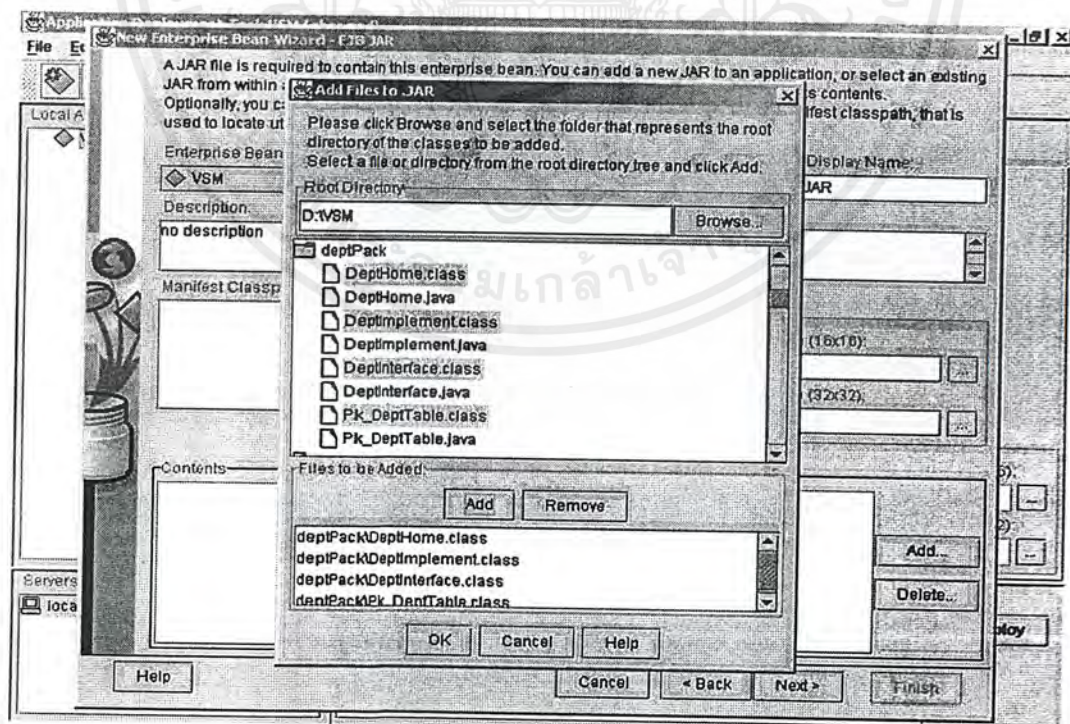
3.2 หน้าแรกจะแสดงรายละเอียดในการสร้างคอมโพเนนต์ เมื่ออ่านเสร็จกดปุ่ม OK



3.3 กำหนดแอปพลิเคชันที่จะให้คอมพิวเตอร์เข้าไปอยู่ (ส่วนที่1) กำหนดชื่อจาร์ไฟล์ (ส่วนที่2) เมื่อกำหนดชื่อเรียบร้อยแล้ว ต่อไปต้องทำการ add ด้านล่าง

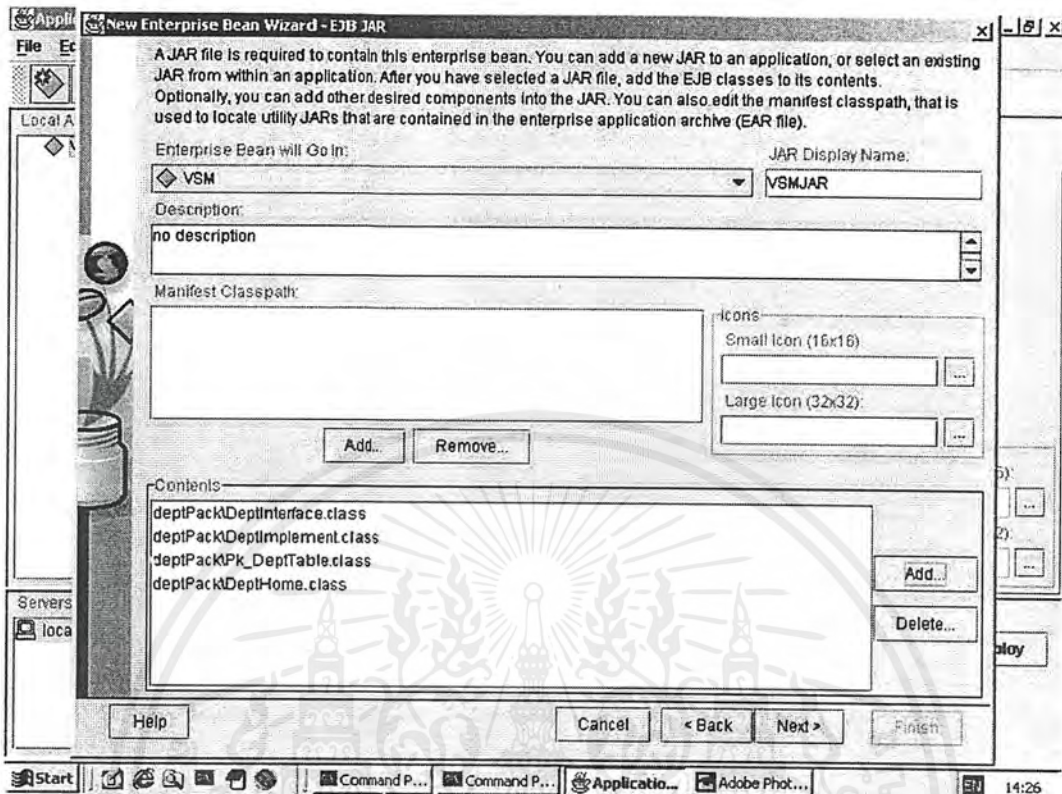


3.4 เลือกไฟล์ที่จะ add โดยถ้าเป็นแพ็คเกจอยู่ด้วย (ในที่นี้ให้เลือกไฟล์ Converter.class, ConverterEJB.class, ConverterHome.class) เลือกเสร็จกด add และ ok



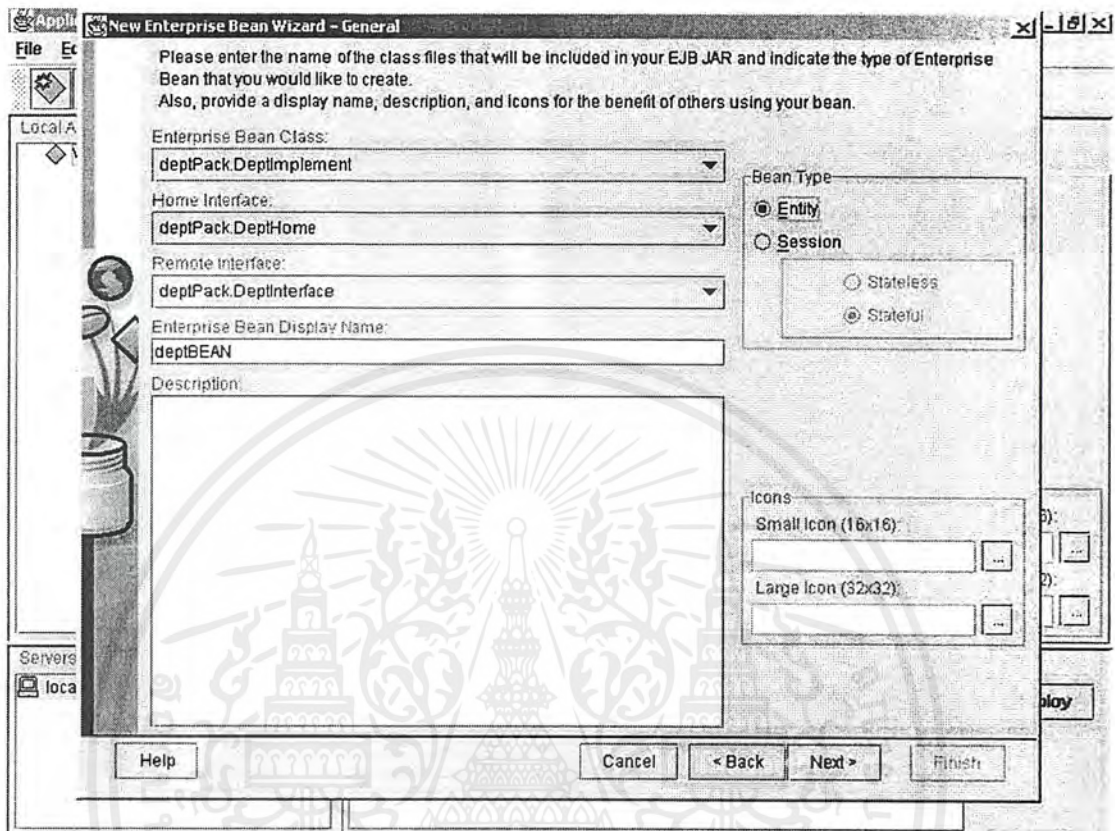
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 เมื่อ add เสร็จแล้วจะขึ้นรายชื่อไฟล์ในจาร์ ในส่วนคอนเทนท์กด Next



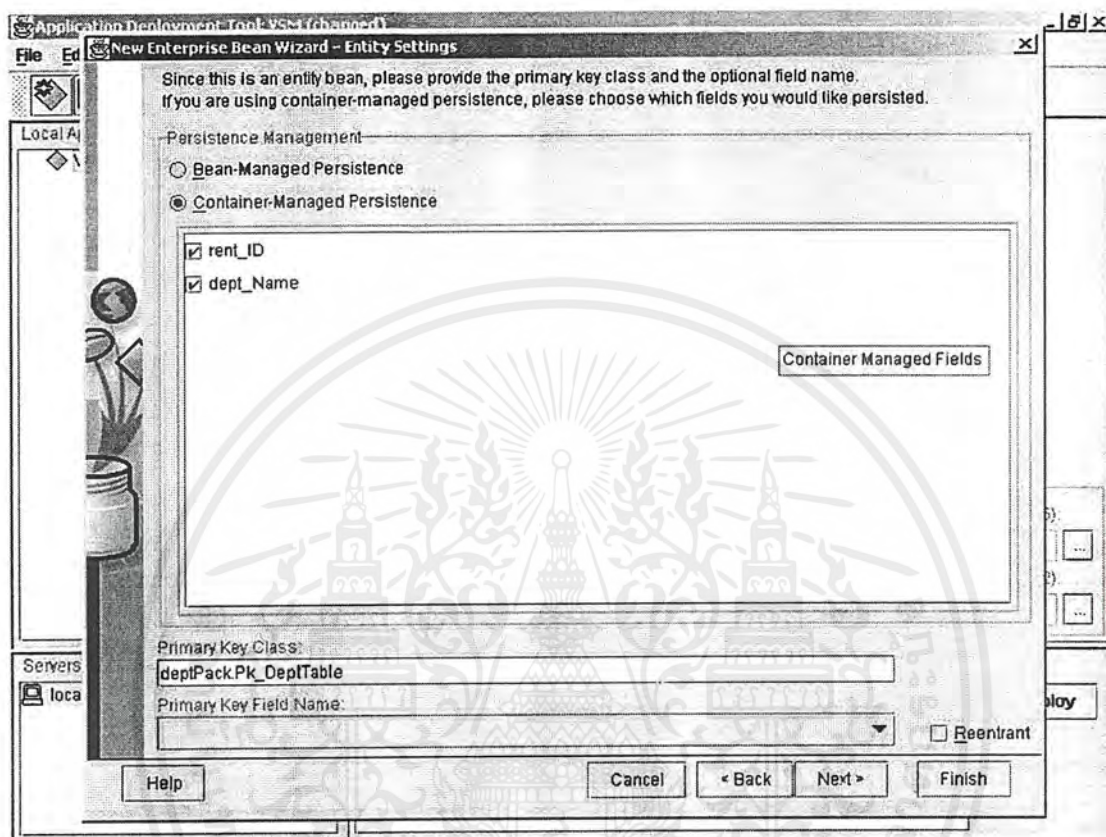
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 เลือกไฟล์ที่เป็นเอ็นเตอร์ไพรส์บีน โฮมอินเตอร์เฟซ รีโมทอินเตอร์เฟซกำหนดชื่อเอ็นเตอร์ไพรส์บีน และกำหนดชนิดของเอ็นเตอร์ไพรส์บีน เป็น เอ็นติตี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 กำหนดชนิดของเอนิตีดีเอ็นว่าเป็นแบบ bean management หรือ container management ใน
 โครงการนี้จะใช้แบบ container management จึงเลือกแบบที่สอง เลือกฟิวส์ที่เกี่ยวข้องกับฐานข้อมูล และ
 กำหนดคลาสที่เป็นไพรมารีคีย์



3.8 คลิกปุ่ม finish

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ดิพลอยเจทูอียีแอปพลิเคชัน

ตอนนี้เจทูอียีแอปพลิเคชันนี้พร้อมที่จะดีพลอยแล้ว โดยการดีพลอยมีขั้นตอนดังนี้
การใส่ JNDI Name ของเอ็นเตอร์ไพส์บี

4.1 ในหน้าต่าง Application Deploy Tool ทำการเลือก ConverterApp ใน tree view

4.2 เลือก JNDI Name tab

4.3 ในช่อง JNDI Name ใส่ My Converter และกดปุ่ม Return

การดีพลอยแอปพลิเคชัน

4.4 ใน เมนู Tools ให้เลือก Deploy Application

4.5 ทำการตรวจสอบ Target เซิร์ฟเวอร์ โดยการใส่ชื่อของโฮสต์ที่จะรันเจทูอียีเซิร์ฟเวอร์ หรือ ใส่
เป็น localhost

4.6 เลือก checkbox labelled “Return Client Jar”

4.7 คลิก Next

4.8 ใน dialog box ทำการตรวจสอบ JNDI Name ของ ConverterBean ว่าคือ MyConverter

4.9 คลิก Next

4.10คลิก Finish

4.11จะปรากฏหน้าต่าง Deployment Progress dialog box ให้คลิก OK เมื่อดีพลอยเสร็จ

5. สร้างไคลเอ็นท์

ทำการเขียนโค้ดไคลเอ็นท์ประกอบด้วย 2 ส่วนคือ

5.1 การเขียนโค้ดไคลเอ็นท์

โค้ดของไคลเอ็นท์ในที่นี้จะให้ชื่อว่า ConverterClient.java ประกอบด้วยพื้นฐาน 3 ส่วนคือ

5.1.1 Locate Home Interface คือ การสร้างการติดต่อ โดยเริ่มจากการอินสแตนทีเอทอ็อบเจกต์
ของโฮมอินเตอร์เฟส

5.1.2 Create an Enterprise Bean Instance คือ การสร้างอินสแตนทีของเอ็นเตอร์ไพส์บี

5.1.3 Invoke a Business Method คือ การเรียกบิซิเนสเมทอดที่ประกาศในรีโมทอินเตอร์เฟส
โดยแต่ละขั้นตอนมีรายละเอียดดังนี้

5.1.1 Locate Home Interface คือการสร้างการติดต่อ โดยเริ่มจากการอินสแตนทีเอทอ็อบเจกต์
ของโฮมอินเตอร์เฟส

ก่อนที่ ConverterClient จะทำการเรียกเมทอด create() ที่อยู่ใน

ConverterHome จะต้องทำการอินสแตนทีเอทอ็อบเจกต์ที่มี type เป็น ConverterHome ก่อน โดยขั้นตอนการ
ได้อ็อบเจกต์ของ ConverterHome มี 3 ขั้นตอนดังนี้

สร้าง JNDI naming context

Context initial = new InitialContext();

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อได้ naming context แล้วนำ JNDI naming context ไปทำการ lookup ไปยัง JNDI name

```
java.lang.Object objref = initial.lookup("MyConverter");
```

1. Narrow เพื่ออ้างอิงไปยังอ็อบเจกต์ของ ConverterHome

```
ConverterHome home = (ConverterHome) PortableRemoteObject.narrow  
(objref, ConverterHome.class);
```

5.1.2 Create an Enterprise Bean Instance คือ การสร้างอินสแตนซ์ของเอนเตอร์ไพสส์บีน การสร้างอินสแตนซ์ของเอนเตอร์ไพสส์บีนทำได้โดยการที่ไคลเอ็นต์ทำการเรียกเมทอด create() บนอ็อบเจกต์ ConverterHome โดยเมทอด create จะทำการรีเทิร์นอ็อบเจกต์ที่มี type เป็น Converter ภายในอินเทอร์เฟซ Converter จะประกาศชื่อของบิซิเนสเมทอดที่ต้องการให้ไคลเอ็นต์เรียกโดยมีเมทอดจริงกำหนดไว้ใน ConverterEJB เมื่อไคลเอ็นต์ทำการเรียกเมทอด create() คอนเทนเนอร์จะอินสแตนทิเอท ConverterEJB ขึ้นมาและทำการเรียกเมทอด ConverterEJB.ejbCreate()

```
Converter currencyConverter = home.create();
```

5.1.3 Invoke a Business Method คือ การเรียกบิซิเนสเมทอดที่ประกาศในรีโมทอินเทอร์เฟซ การเรียกบิซิเนสเมทอดทำได้โดยเรียกจากอ็อบเจกต์ Converter โดยคอนเทนเนอร์จะทำการเรียกเมทอดใน ConverterEJB ซึ่งรันอยู่บนเซิร์ฟเวอร์ เช่น ไคลเอ็นต์ทำการเรียกเมทอด dollarToYen() ดังนี้

```
double amount = currencyConverter.dollarToYen(100.00);
```

5.2 การคอมไพล์ไคลเอ็นต์

การคอมไพล์ไคลเอ็นต์ทำได้โดย

1. การสร้างไฟล์ compileClient.bat ดังนี้

```
set J2EE_HOME = <installation-location>  
set CPATH = .;%J2EE_HOME%\lib\j2ee.jar  
javac -classpath %CPATH% ConverterClient.java
```

หมายเหตุ <installation-location> คือ ไดรเวอร์ที่ติดตั้ง เจทูอี SDK

2. การรันไฟล์ compileClient.bat

6. รันไคลเอ็นต์

การรันไคลเอ็นต์ต้องมีไฟล์ ConverterAppClient.jar โดยไฟล์จาร์นี้ภายในจะใส่สแตปคลาส

ที่อนุญาตให้ไคลเอ็นต์ทำการติดต่อกับอินสแตนซ์ของเอนเตอร์ไพสส์บีนที่รันอยู่ในคอนเทนเนอร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การสร้างไฟล์ testClient.bat ดังนี้

```
set J2EE_HOME = <installation-location>
```

```
set CPATH = .;%J2EE_HOME%\lib\j2ee.jar;ConverterAppClient.jar
```

```
javac -classpath %CPATH% ConverterClient
```

หมายเหตุ <installation-location> คือ ไดรเวอร์ที่ติดตั้งเจทูอีเอสดีเค

2. การรันไฟล์ testClient.bat



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ชื่อผู้แต่ง ชื่อหนังสือ สำนักพิมพ์ ปีที่พิมพ์ หน้าที่

- [1] Tsichritzis D.C. and Klug A. (1978) : "*The ANSI/X3/SPARC DBMS Frame Work : Report of the Study Group on Data Base Management Systems*", "Information System" ,3 1978.
- [2] Ed Roman (1999) : "*Mastering Enterprise JavaBeans and the Java 2 Platform Enterprise Edition*" :Wiley Computer Publishing
- [3] Sun Microsystems,INC (2000) : "*The Java 2 Enterprise Edition Developing Guide*"
- [4] David Harel (1998) : "*Real-Time UML Developing Efficient Objects for Embedded System*" : ADDISON-WESLEY
- [5] Joseph O'Neil Edited by Herb Schildt (1998) : "*JavaBeansTM Programming from the Ground Up*" : McGraw-Hill
- [6] <http://www.java.sun.com>
- [7] <http://www.javacentrix.com>
- [8] <http://www.oracle.com>
- [9] <http://www2.theserverside.com>
- [10] www.bea.com