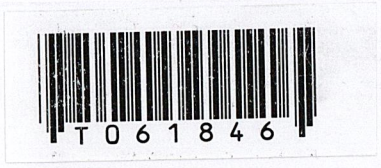
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบทะเบียนบัณฑิตวิทยาลัย

The Development of Student Information System



โดย

นายพีรพัฒน์ อโศกธรรมรังสี 43010312

อาจารย์ที่ปรึกษา

อ. ประสาร ตังติสานนท์

เลขหมู่.....
เลขทะเบียน 61846
วัน,เดือน,ปี 21 ก.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

ปริญญานิพนธ์ปีการศึกษา 2546

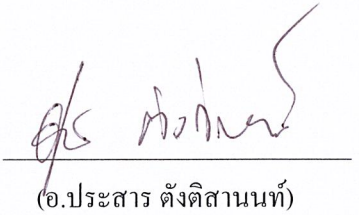
ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบทะเบียนบัณฑิตวิทยาลัย

The Development of Student Information System

ผู้จัดทำ 1. นายพีรพัฒน์ อโศกธรรมรังสี รหัสประจำตัว 43010312



อาจารย์ที่ปรึกษา

(อ.ประสาร ตังติสานนท์)

ระบบทะเบียนบัณฑิตวิทยาลัย

นายพีรพัฒน์ อโศกธรรมรังสี 43010312
อ. ประสาร ดังติสานนท์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

ในการวิเคราะห์และออกแบบระบบเชิงวัตถุโดยใช้ภาษาที่เรียกว่ายูเอ็มแอล (Unified Modeling - Language) ในการวิเคราะห์และออกแบบและจะทำการพัฒนาระบบที่ได้รับการออกแบบมา โดยที่ต้องการพัฒนาระบบเชิงวัตถุนั้นเป็นระบบที่มีความเป็นกลางซึ่งระบบนั้นจะสามารถที่จะปรับเปลี่ยนตามความต้องการของผู้ใช้ได้ โดยนำระบบสารสนเทศนักศึกษาในระดับบัณฑิตศึกษาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังมาเป็นกรณีศึกษา ในการศึกษาเริ่มต้นตั้งแต่ขั้นตอนวิเคราะห์รายการปัญหาแล้วนำมาสร้างเป็นระบบเชิงวัตถุ จากนั้นจึงทำการออกแบบระบบและรายละเอียดเพื่อทำการพัฒนาโปรแกรม

โปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นใช้สถาปัตยกรรมแบบเว็บแอพพลิเคชั่น พัฒนาโดยใช้ภาษาจาวากับเจเอสพี โดยมีทอมแค็ทเป็นเว็บเซิร์พเวอร์และจัดเก็บข้อมูลโดยใช้ระบบจัดเก็บข้อมูลเชิงสัมพันธ์ออราเคิล9ไอ

The Development of Student Information System

Peerapat Asoktummarungsri

Prasarn Tangtisanon Advisor

ABSTRACT

Analysis and Design object-oriented system using a language called "UML" (Unified Modeling Language) This project is to analyze, design and develop the system such that the developed object-oriented system must be in general compatibility. The system is able to adapt to meet User's needs. The Student Information System(SIS) of graduated school is used as a study case for this thesis. The process of this study starts from defining problem statement. This result is used for creating the object-oriented model. The more information about system design and detail design is studied for a program development.

This program is developed in Web-Based architecture.

กิตติกรรมประกาศ

ข้าพเจ้าขอเอ่ยนามบุคคลเหล่านี้ เพื่อขอขอบพระคุณที่มีส่วนสำคัญอย่างยิ่งในการทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้

อาจารย์ ประสาร ตังติสานนท์ อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์และอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้มีคุโณปการอย่างยิ่งในทุกๆ ด้านต่อข้าพเจ้าตั้งแต่ข้าพเจ้าได้เข้ามาศึกษาในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังจนถึงทุกวันนี้

พี่ปิยะวรรณ ทองเลื่อน หัวหน้าแผนกโปรแกรมที่ให้ความเอาใจใส่และให้คำแนะนำเสมอมา

พี่ธีรนัท เริ่มคิดการ ผู้ให้คำปรึกษาเกี่ยวกับระบบทะเบียนและคำแนะนำในการพัฒนา

ขอขอบพระคุณบุคลากรและคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้ประสิทธิประสาทความรู้ให้โดยตลอด รวมถึงทุกน้ำใจที่เกิดจากเพื่อนร่วมสถาบันโดยเฉพาะเพื่อนๆในภาควิชาวิศวกรรมคอมพิวเตอร์

สุดท้ายขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมาในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

หากปราศจากบุคคลเหล่านี้ก็คงไม่มีวิทยานิพนธ์เล่มนี้

นายพีรพัฒน์ อโศกธรรมรังสี

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญประกอบภาพ	X
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและความเป็นมา	‘’
1.2 วัตถุประสงค์ของโครงงาน	‘’
1.3 ปัญหาที่เกิดจากระบบในปัจจุบัน	2
1.4 แนวทางการแก้ไข	‘’
1.5 ขอบเขตของโครงงาน	‘’
1.6 ผลที่คาดว่าจะได้รับ	
บทที่ 2 การออกแบบโปรแกรมด้วยวิธีเชิงวัตถุ	3
2.1 หลักการของวิธีเชิงวัตถุ	‘’
2.2 คุณสมบัติของหลักการเชิงวัตถุ	4
2.2.1 เอ็นแคปซูเลชั่น	‘’
2.2.2 การถ่ายทอดคุณสมบัติ	‘’
2.2.3 โพลิมอร์ฟิซึม	‘’
2.3 ภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ	‘’
2.4 ภาษาจาวา	5
2.5 หลักการทำงานของภาษาจาวา	‘’
2.6 จาวาเซิร์ฟเล็ต	6
2.7 จาวาบีน	7
บทที่ 3 จาวาเซิร์ฟเวอร์เพจ	8
3.1 โครงสร้างและขั้นตอนการทำงานของ JSP	‘’
3.2 เปรียบเทียบ JSP กับเทคโนโลยีอื่นๆ	10
3.2.1 JSP กับ Servlet	‘’
3.2.2 JSP กับ ASP	‘’

3.2.3 JSP กับ JavaScript 10
3.3 แท็กพื้นฐานและคำสั่งควบคุม 11
3.4 แท็กหมายเหตุ 12
3.5 Directive กับ Action ‘’
3.5.1 Directive ‘’
3.5.2 Action ‘’
3.6 Implicit Object 13
บทที่ 4 การวิเคราะห์และการออกแบบระบบ 14
4.1 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ ‘’
4.1.1 การวิเคราะห์ความต้องการของระบบหรือผู้ใช้ ‘’
4.1.2 การวิเคราะห์ระบบ ‘’
4.1.3 การออกแบบระบบ ‘’
4.1.4 การสร้างโปรแกรมระบบ 15
4.1.5 การทดสอบระบบ ‘’
4.2 Unified Modeling Language ‘’
4.3 ไดอะแกรมที่เป็นโครงสร้างแบบสแตติก 16
4.3.1 ยูสเคสไดอะแกรม ‘’
4.3.2 คลาสไดอะแกรม 17
4.4 ความสัมพันธ์กันระหว่างคลาส 18
4.4.1 Dependency ‘’
4.4.2 Generalization ‘’
4.4.3 Association ‘’
4.4.3.1 Normal Association ‘’
4.4.3.2 Aggregation 19
4.4.3.3 Composition ‘’
4.5 ไดอะแกรมที่เป็นโครงสร้างแบบไดนามิก 20
4.5.1 สเตตชาร์ตไดอะแกรม ‘’
4.5.2 ซีเควนต์ไดอะแกรม ‘’
4.5.3 คอลแลบอเรชั่นไดอะแกรม 21
4.5.4 แอ็กทิวิตี้ไดอะแกรม 22
4.6 อิมพลีเมนเตชั่นไดอะแกรม 23
4.6.1 คอมโพเนนต์ไดอะแกรม ‘’

4.6.2 ดีพลอยเมนต์ไดอะแกรม 24
บทที่ 5 ระบบเครือข่ายแบบไคลเอ็นต์/เซิร์ฟเวอร์ 25
5.1 การติดต่อแบบไคลเอนต์/เซิร์ฟเวอร์ ‘’
5.2 รูปแบบของไคลเอ็นต์/เซิร์ฟเวอร์ 26
5.2.1 Stand alone Client/Server ‘’
5.2.2 Department Client/Server ‘’
5.2.3 Workgroups Client/Server ‘’
5.2.4 Enterprise Client/Server 27
5.3 การจัดแบ่งประเภทของไคลเอ็นต์/เซิร์ฟเวอร์ ‘’
5.3.1 One Tier ‘’
5.3.2 Two Tiers ‘’
5.3.3 Three Tiers 28
5.4 การจัดแบ่งประเภทของเซิร์ฟเวอร์ ‘’
5.4.1 File Server ‘’
5.4.2 Database Server ‘’
5.4.3 Application Server ‘’
5.5 เว็บเซิร์ฟเวอร์ และเว็บบราวเซอร์ 29
5.5.1 เว็บเซิร์ฟเวอร์ ‘’
5.5.2 เว็บบราวเซอร์ ‘’
5.6 การทำงานของเว็บเซิร์ฟเวอร์และเว็บบราวเซอร์ ‘’
5.7 HTTP ‘’
5.8 HTML 30
5.9 Client Side Script และ Server Side Script ‘’
5.9.1 Client Side Script 31
5.9.2 Server Side Script ‘’
บทที่ 6 ทฤษฎีและความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล 32
6.1 ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ ‘’
6.1.1 ความสามารถของระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ‘’
6.1.2 ความสามารถของระบบจัดการฐานข้อมูลแบบออบเจ็กต์ ‘’
6.2 ประเภทของข้อมูลในฐานข้อมูล 33
6.3 ชนิดข้อมูลแบบกำหนดเอง 34
6.3.1 ชนิดข้อมูลแบบออบเจ็กต์ ‘’

6.3.2 ชนิดข้อมูลแบบกลุ่ม 34
6.4 ฐานข้อมูล 35
6.5 Java Database Connectivity ‘’
6.5.1 รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC 36
6.5.1.1 Two Tiers model ‘’
6.5.1.2 Three Tiers Model ‘’
6.6 สถาปัตยกรรมของ Oracle 9i ‘’
6.7 ความสามารถของฐานข้อมูล Oracle 9i 38
6.7.1 เปิดฐานขึ้นมาใช้งานได้อย่างรวดเร็ว ‘’
6.7.2 การจัดการฐานข้อมูลแบบออนไลน์ ‘’
6.7.3 ปรับเปลี่ยนค่าหน่วยความจำที่จองไว้ใช้งานได้ทันที ‘’
6.7.4 เป็นฐานข้อมูลแบบ Application Cluster ที่แท้จริง 39
6.7.5 การเข้ารหัสเพื่อเพิ่มความปลอดภัยให้กับข้อมูล ‘’
6.7.6 รองรับการทำงานกับข้อมูลที่มีขนาดใหญ่ได้เป็นอย่างดี 40
6.7.7 การจัดการ Tablespace กำหนดเป็นแบบ Locally Management ‘’
6.7.8 การกำหนด Default Temporary Tablespace ให้กับผู้ใช้ ‘’
6.7.9 ลบไฟล์ข้อมูลที่เป็นส่วนประกอบใน Tablespace ให้โดยอัตโนมัติ ‘’
6.7.10 รูปแบบใหม่ในการจัดการไฟล์ผ่าน Oracle-Managed File 41
6.7.11 กำหนดเวลาในการกู้ฐานข้อมูลคืนได้ ‘’
6.7.12 ขนาดของ DB Block ไม่จำเป็นต้องเท่ากันทุก Tablespace ‘’
บทที่ 7 การออกแบบโปรแกรมประยุกต์การจัดการงานสอน 42
7.1 เครื่องมือที่เลือกใช้ในการพัฒนาโปรแกรม ‘’
7.2 รายละเอียดการออกแบบของโปรแกรมประยุกต์การจัดการงานสอน ‘’
7.3 ส่วนเข้าสู่ระบบ 43
7.3.1 ส่วนเข้าสู่ระบบของนักศึกษา 44
7.3.2 ส่วนเข้าสู่ระบบของเจ้าหน้าที่ ‘’
7.3.3 ส่วนการเปลี่ยนรหัสผ่านของทั้งนักศึกษาและเจ้าหน้าที่ ‘’
7.4 การจัดเก็บข้อมูล 45
7.4.1 ข้อมูลคณะ ‘’
7.4.2 ข้อมูลภาควิชา ‘’
7.4.3 ข้อมูลรายวิชา 46
7.4.4 ข้อมูลหลักสูตร 47

7.4.5 ข้อมูลอาจารย์ 48
7.4.6 ข้อมูลนักศึกษา ‘’
7.5 ส่วนลงทะเบียน 49
7.5.1 การลงทะเบียน ‘’
7.5.2 การเพิ่มวิชาเรียน 51
7.5.3 การเปลี่ยนวิชาเรียน 53
7.5.4 การถอนวิชาเรียน 55
7.6 ยูสเคสไดอะแกรมระบบ 57
7.7 คลาสไดอะแกรมของระบบ 58
บทที่ 8 สรุปผลการทดลอง 59
8.1 ผลการวิจัยและพัฒนา ‘’
8.1.1 การกำหนดรายการปัญหา ‘’
8.1.2 การวิเคราะห์ ‘’
8.1.3 การออกแบบระบบ 60
8.1.4 การออกแบบรายละเอียด ‘’
8.1.5 การพัฒนาระบบงาน ‘’
8.2 แนวทางการใช้งาน ‘’
8.3 ข้อเสนอแนะ 61

ภาคผนวก

ภาคผนวก ก ฐานข้อมูลที่ใช้ในระบบ 62

บรรณานุกรม 72

สารบัญรูปภาพ

	หน้าที่
บทที่ 3 จาวาเซิร์ฟเวอร์เพจ	
รูปที่ 3-1 รูปแสดงโครงสร้างและขั้นตอนการประมวลผลไฟล์ JSP	8
	9
บทที่ 4 การวิเคราะห์และการออกแบบระบบ	
รูปที่ 4-1 แสดง ยูสเคสไดอะแกรม	14
รูปที่ 4-2 แสดง คลาสไดอะแกรม	16
รูปที่ 4-3 แสดง ความสัมพันธ์แบบพึ่งพิง	17
รูปที่ 4-4 แสดง ความสัมพันธ์ระหว่างซูเปอร์คลาสและซับคลาส	18
รูปที่ 4-5 แสดง ความสัมพันธ์ระหว่างคลาสแบบ Normal Association	,,
รูปที่ 4-6 แสดง ความสัมพันธ์ระหว่างคลาสแบบ Aggregation Association	19
รูปที่ 4-7 แสดง ความสัมพันธ์ระหว่างคลาสแบบ Composition Association	,,
รูปที่ 4-8 แสดง สเตตชาร์ตไดอะแกรม	,,
รูปที่ 4-9 แสดงซีเควนต์ไดอะแกรม	20
รูปที่ 4-10 คอลแลบอเรชั่นไดอะแกรม	21
รูปที่ 4-11 แอ็กทิวิตี้ไดอะแกรม	22
	23
บทที่ 5 ระบบเครือข่ายแบบไคลเอ็นต์/เซิร์ฟเวอร์	
รูปที่ 5-1 การทำงานในลักษณะ Host-based System และ Client/Server	25
รูปที่ 5-2 การจัดแบ่งไคลเอ็นท์/เซิร์ฟเวอร์แบบ 1 tier	26
รูปที่ 5-3 การจัดแบ่งไคลเอ็นท์/เซิร์ฟเวอร์แบบ 2 tier	27
รูปที่ 5-4 การจัดแบ่งไคลเอ็นท์/เซิร์ฟเวอร์แบบ 3 tier	,,
	28
บทที่ 6 ทฤษฎีและความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล	
รูปที่ 6-1 แสดงสถาปัตยกรรมของ Oracle 9i ที่รองรับการทำงานแบบ 3-Tier	32
รูปที่ 6-2 รูปแสดง SGA ของฐานข้อมูล	37
รูปที่ 6-3 รูปแบบ Oracle9i Real Application Cluster	38
รูปที่ 6-4 รูปแสดงการเข้ารหัสของข้อมูลในฐานข้อมูลเพื่อความปลอดภัยให้กับฐานข้อมูล	39
	40
บทที่ 7 การออกแบบโปรแกรมประยุกต์การจัดการงานสอน	
รูปที่ 7-1 รูปแสดง user interface ของส่วนเข้าสู่	43

รูปที่ 7-2 รูปแสดง user interface ของส่วนเปลี่ยนรหัสผ่าน 43
รูปที่ 7-3 รูปแสดง user interface การจัดเก็บข้อมูลคณะ 45
รูปที่ 7-4 รูปแสดง user interface ของการจัดเก็บข้อมูลภาควิชา ‘’
รูปที่ 7-5 รูปแสดง user interface ของการจัดเก็บข้อมูลรายวิชา 46
รูปที่ 7-6 รูปแสดง user interface ของการเปิดเซ็กชั่น ‘’
รูปที่ 7-7 รูปแสดง user interface ของการจัดเก็บข้อมูลหลักสูตร 47
รูปที่ 7-8 รูปแสดง user interface ของการจัดเก็บข้อมูลอาจารย์ 48
รูปที่ 7-8 รูปแสดง user interface ของการจัดเก็บข้อมูลอาจารย์ ‘’
รูปที่ 7-10 รูปแสดง user interface ของการลงทะเบียน 49
รูปที่ 7-11 รูปแสดง user interface ของการยืนยันการลงทะเบียน ‘’
รูปที่ 7-12 รูปแสดง state diagram ของการลงทะเบียน 50
รูปที่ 7-13 รูปแสดง user interface ของการเพิ่มวิชาเรียน 51
รูปที่ 7-14 รูปแสดง state diagram ของการเพิ่มวิชาเรียน 52
รูปที่ 7-15 รูปแสดง user interface ของการเปลี่ยนวิชาเรียน 53
รูปที่ 7-16 รูปแสดง state diagram ของการมเปลี่ยนวิชาเรียน 54
รูปที่ 7-17 รูปแสดง user interface ของการถอนวิชาเรียน 55
รูปที่ 7-18 รูปแสดง state diagram ของการถอนวิชาเรียน 56
รูปที่ 7-19 รูปแสดงยูสเคสไดอะแกรมของระบบ 57
รูปที่ 7-20 รูปแสดงคลาสไดอะแกรมของระบบ 58

สารบัญตาราง

ตารางที่ 3-1 แสดง Implicit Object 13

บทที่ 1
บทนำ

1.1 ความสำคัญและความเป็นมา

แนวความคิดในปัจจุบันของเทคโนโลยีคอมพิวเตอร์เน้นหนักไปทางด้านแนวคิดเชิงวัตถุ เนื่องจาจากแนวคิดเชิงวัตถุเป็นวิธีที่พิจารณาขอบเขตของสิ่งที่สนใจให้เป็นเหมือนกับที่เก็บออปเจ็กหรือวัตถุที่มีปฏิกิริยาต่อกัน โดยแต่ละออปเจ็กประกอบกันระหว่างคุณลักษณะและพฤติกรรมของออปเจ็กนั้นแนวคิดนี้ได้ถูกนำไปประยุกต์ใช้ได้อย่างกว้างขวาง ทั้งในเรื่องของ ฮาร์ดแวร์ ภาษาโปรแกรม ดาต้าเบส ยูซเซอร์อินเทอร์เฟส และซอฟแวร์เอนจิเนียริ่ง และในการวิเคราะห์และออกแบบระบบงาน ก็ได้นำแนวคิดเชิงวัตถุเข้ามาใช้เนื่องจากพบว่าจะมีความสะดวกและยืดหยุ่นกว่าวิธีการแบบลำดับขั้น (Procedure) เพราะการวิเคราะห์ระบบด้วยวิธีการเชิงวัตถุจะทำให้ทราบโครงสร้างของข้อมูลและประโยชน์ใช้สอย (Function) ของโปรแกรมไปด้วยพร้อมกัน ทำให้ทั้งฐานข้อมูลและโปรแกรมสามารถถูกพัฒนาไปพร้อมกันได้ นอกจากนั้นยังเอื้ออำนวยต่อความสะดวกมนการบำรุงรักษา และการนำมาใช้ใหม่อย่างมีศักยภาพอีกด้วย

เนื่องด้วยบัณฑิตวิทยาลัยเป็นหน่วยงานที่ให้บริการแก่นักศึกษาในระดับปริญญาโทและปริญญาเอกซึ่งได้มีการนำเทคโนโลยีคอมพิวเตอร์เข้ามาช่วยในการจัดการงานด้านต่างๆ เพื่อให้การดำเนินงานเป็นไปอย่างรวดเร็วและมีประสิทธิภาพและเกิดประโยชน์สูงสุดต่อการบริการและการเรียนการสอนของนักศึกษา

1.2 วัตถุประสงค์ของโครงงาน

- นำแบบจำลองระบบสารสนเทศนักศึกษาในระดับบัณฑิตศึกษาที่ได้ทำการวิเคราะห์มาพัฒนาเป็นโปรแกรมต้นแบบ
- เพื่อศึกษาการเขียนโปรแกรมประยุกต์เว็บเพจ (Web Application) โดยใช้ภาษาประเภท Server Side Script
- เพื่อศึกษาการออกแบบและใช้งานระบบฐานข้อมูล
- เพื่อนำหลักการที่ได้เรียนมาประยุกต์ในการสร้างโปรแกรมประยุกต์จริง

1.3 ปัญหาที่เกิดจากระบบในปัจจุบัน

ทุกวันนี้ระบบทะเบียนบัณฑิตวิทยาลัยได้ใช้ระบบคอมพิวเตอร์เพื่อช่วยในการทำงานซึ่งตัวโปรแกรมที่ใช้อยู่นั้นเป็นลักษณะของไคลเอ็น/เซิร์พเวอร์ซึ่งได้นำส่วนของการประมวลผลมาไว้ที่ตัวไคลเอ็น ทำให้เมื่อมีการเปลี่ยนแปลงในส่วนของประมวลผล จะต้องตามไปแก้ไขที่เครื่องไคลเอ็นทุกเครื่อง ก่อให้เกิดความไม่สะดวกในการทำงานและการปรับปรุงแก้ไขระบบ

1.4 แนวทางการแก้ไข

ในปัจจุบันได้มีการพัฒนาเทคโนโลยีที่เรียกว่าเว็บแอพพลิเคชั่นซึ่งได้นำส่วนของการประมวลผลออกจากเครื่องไคลเอ็นมาเก็บไว้ที่ตัวเว็บเซิร์พเวอร์โดยเครื่องไคลเอ็น จะทำหน้าที่ในส่วนการแสดงผลเท่านั้นและเครื่องไคลเอ็นจะเข้ามาใช้งานในส่วนของการประมวลผลผ่านทางเว็บบราวเซอร์ซึ่งทำให้ถ้ามีการแก้ไขในส่วนของ ก็แก่ที่ตัวเว็บเซิรพ์เวอร์เท่านั้น

1.5 ขอบเขตของโครงงาน

ส่วนของนักศึกษา

- การสอบถามข้อมูล
- การลงทะเบียน

ส่วนของเจ้าหน้าที่

- บันทึกข้อมูลต่างๆ เช่น ภาควิชา, คณะ, หลักสูตร, ฯลฯ
- การลงทะเบียน, เพิ่ม, เปลี่ยน, ถอน สำหรับนักศึกษา
- การประมวลผลการศึกษา

1.6 ผลที่คาดว่าจะได้รับ

- ได้รับความรู้ในการพัฒนาระบบแบบเว็บแอพพลิเคชั่นและส่วนอื่นๆที่เกี่ยวข้อง เช่น ดาต้าเบส
- ได้ระบบทะเบียนตัวใหม่ที่สามรถนำไปใช้งานได้
- ได้ตัวต้นแบบของระบบทะเบียนที่สามารถนำไปในการศึกษาเพื่อพัฒนาต่อไป
- ได้แนวคิดในการพัฒนาระบบทะเบียนในลักษณะของเว็บแอพพลิเคชั่นเพื่อใช้ในการพัฒนาต่อไป

บทที่ 2
การออกแบบโปรแกรมด้วยวิธีเชิงวัตถุ

ในการออกแบบโปรแกรมด้วยวิธีเชิงวัตถุเป็นหลักการซึ่งมีองค์ประกอบต่าง ๆ ที่เรียกว่าโมเดลของวัตถุ (Object Model) ในโมเดลของวัตถุนั้นแบ่งออกเป็นการแยกแยะเอกลักษณ์ (Abstraction) การซ่อนข้อมูล (Encapsulation) การรวมกลุ่มความสัมพันธ์ (Modularity) ลำดับชั้น (Hierarchy) ชนิด (Typing) การทำงานพร้อมกัน (Concurrency) และการรักษาสถานะ (Persistence) สำหรับลักษณะการวิเคราะห์และออกแบบระบบด้วยวิธีเชิงวัตถุ จะแตกต่างจากการออกแบบระบบด้วยวิธีโครงสร้าง เนื่องจากรูปแบบในการคิดและการมองระบบจะต่างกัน และภาษาที่ใช้สำหรับการเขียนโปรแกรมเชิงวัตถุแต่ละภาษาก็มีรายละเอียดที่ต่างกันด้วย

2.1 หลักการของวิธีเชิงวัตถุ

การพัฒนาระบบแบบโครงสร้างนั้นจะพยายามให้นักพัฒนาระบบแตกปัญหาออกเป็นหลายๆ ฟังก์ชัน (Function) หรือโพรซีเดอร์ (Procedure) แล้วแก้ปัญหาในแต่ละส่วนด้วยอัลกอริทึม (Algorithm) และโครงสร้างข้อมูล (Data Structure) เฉพาะเพื่อช่วยให้การพัฒนาโปรแกรมเป็นไปได้ง่ายโดยเฉพาะโปรแกรมขนาดเล็ก

สำหรับการพัฒนาระบบด้วยวิธีเชิงวัตถุก็เช่นเดียวกัน แต่จะแก้ไขปัญหาโดยการแตกปัญหาที่กำลังพิจารณาออกเป็นส่วนย่อย ๆ ซึ่งมีความซับซ้อนน้อยลงและเรียกแต่ละส่วนเหล่านี้ว่า "วัตถุ"

วัตถุต่าง ๆ เหล่านี้จะถูกประกอบขึ้นมาเป็นระบบที่สมบูรณ์ที่สุด ทำให้สามารถใช้ได้กับโปรแกรมขนาดใหญ่และมีความซับซ้อนเพิ่มมากขึ้นได้ การพัฒนาระบบด้วยวิธีเชิงวัตถุไม่เพียงจะเป็นแนวความคิดใหม่ในการเขียนโปรแกรมเท่านั้นแต่ยังมีข้อดีจากโมเดลของวัตถุ ซึ่งเป็นแนวความคิดใหม่ในวงการคอมพิวเตอร์อีกด้วย สามารถนำหลักการในการพัฒนาระบบด้วยวิธีเชิงวัตถุไปประยุกต์ใช้กับการออกแบบ ส่วนติดต่อกับผู้ใช้ (User Interface) ระบบฐานข้อมูลหรือแม้แต่การออกแบบสถาปัตยกรรมฮาร์ดแวร์ของคอมพิวเตอร์ เนื่องจากหลักการของการพัฒนาระบบด้วยวิธีเชิงวัตถุนั้นสามารถนำไปประยุกต์ใช้ในงานต่างๆได้อย่างกว้างขวาง

2.2 คุณสมบัติของหลักการเชิงวัตถุ

2.2.1 เอ็นแคปซูเลชั่น

Encapsulation หมายถึง การจัดกลุ่มองค์ความคิดที่คล้ายคลึงกันเข้าเป็นหน่วยเดียวกันเพื่อทำการอ้างถึงด้วยชื่อเดียวกัน ในเชิงซอฟต์แวร์โปรแกรมเมอร์มักจะพบว่าชุดคำสั่งที่คล้ายคลึงกันปรากฏอยู่หลายครั้งในตัวโปรแกรม ดังนั้นจึงเกิดแนวความคิดในการจับรูปแบบการซ้ำดังกล่างเข้ามาอยู่ในสิ่งเดียวกันอันเป็นที่มาของเอ็นแคปซูเลชั่นนั่นเอง สิ่งที่ตามมานอกจากความง่ายในการทำความเข้าใจตัวโปรแกรมแล้วยังเป็นการช่วยประหยัดหน่วยความจำอีกด้วย

ในแนวคิดเชิงวัตถุ เอ็นแคปซูเลชั่น หมายถึงการรวบรวมโอเปอเรชั่น (operation) และแอตทริบิวต์ (attribute) เข้าเป็นหน่วยเดียวกันเพื่อที่ว่าแอตทริบิวต์สามารถถูกเปลี่ยนแปลงได้อย่างเหมาะสมโดยผ่านโอเปอเรชั่น และเราเรียกผลที่เกิดจากการใช้งานเอ็นแคปซูเลชั่นว่า การซ่อนข้อมูล (information hiding)

2.2.2 การถ่ายทอดคุณสมบัติ

Inheritance คือ วิธีในการสร้างคลาสใหม่จากคลาสเดิมที่มีอยู่ ทั้งนี้คลาสที่สร้างขึ้นใหม่จะมีวัตถุประสงค์ในการทำงานที่เฉพาะเจาะจงมากยิ่งขึ้น

การถ่ายทอดคุณสมบัติและความสามารถของคลาสหนึ่งไปยังอีกคลาสหนึ่ง ลักษณะความสัมพันธ์ของคลาสที่ใช้ข้อมูลและพฤติกรรมร่วมกันโดยมีความสัมพันธ์ระหว่างคลาสในลักษณะความสัมพันธ์แบบ "is-a" ในการถ่ายทอดคุณสมบัติของคลาสอันประกอบด้วยแอตทริบิวต์ (Attribute) และเมธอด (Method) ในที่นี้คลาสที่ถูกถ่ายทอดคุณสมบัติจะถูกเรียกว่าซูเปอร์คลาส (Superclass , Parent class) และเรียกคลาสที่ได้รับการถ่ายทอดคุณสมบัติว่าซับคลาส (Subclass , Child class , Derived class) ซึ่งทำให้ซอฟต์แวร์มีความสามารถนำกลับมาใช้ได้ (Reusable)

2.2.3 โพลิมอร์ฟิซึม

Polymorphism หมายถึง การเปลี่ยนรูปร่างของออบเจกต์หนึ่งๆ ซึ่งในเชิงโปรแกรมมิ่งจะเป็นการที่ตัวแปรออบเจกต์ของคลาสใดคลาสหนึ่งสามารถเปลี่ยนรูปแบบไปจากคลาสเดิมได้ เช่น

การสร้างออบเจกต์ของซับคลาสและกำหนดตัวแปรออบเจกต์ดังกล่าวให้เป็นประเภทซูเปอร์คลาสแทนนอกจากนี้ยังรวมถึงการที่โอเปอเรชั่นเดียวกันที่มีพฤติกรรมที่แตกต่างกันเมื่อถูกออบเจกต์ที่เกิดจากคนละคลาสกัน

การส่งแมสเซจเดียวกันให้วัตถุที่ต่างกันทำให้วัตถุสามารถแสดงการทำงานที่ต่างกันออกมาได้ คุณสมบัตินี้ทำให้ซอฟต์แวร์มีความยืดหยุ่น (Flexible)

2.3 ภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ

การเขียนโปรแกรมด้วยภาษาเชิงวัตถุ เป็นวิธีการในการสร้างโปรแกรมให้มีโครงสร้างในการทำงานร่วมกันของวัตถุ ซึ่งวัตถุนั้นจะเป็นตัวตนหรือเป็นอินสแทนซ์ (Instance) ของคลาส และคลาสก็เป็นส่วนหนึ่งของลำดับชั้น (Hierarchy) ของคลาสที่มีความสัมพันธ์ในการสืบทอดคุณสมบัติ (Inheritance) โดยคุณสมบัติเด่นของภาษาในการเขียนโปรแกรมเชิงวัตถุได้นั้นจะต้องมีคุณสมบัติ 3 ประการเรียกว่า PIE ซึ่งย่อมาจาก Polymorphism , Inheritance , และ Encapsulation

ถ้าภาษาใดขาดคุณสมบัติหนึ่งในสามข้อนี้ไปก็จะไม่เรียกว่าเป็นภาษาในการเขียนโปรแกรมเชิงวัตถุ ตัวอย่างเช่น ถ้าภาษาใดสนับสนุนการสร้างคลาสและวัตถุ แต่ไม่สนับสนุนการสืบทอดคุณสมบัติของคลาส

แล้วเราจะเรียกภาษานั้นว่า ภาษาที่ทำงานบน Abstract Data Type หรือเรียกว่าภาษาในการเขียนโปรแกรมบนพื้นฐานของวัตถุ (Object Based) มิใช่ภาษาในการเขียนโปรแกรมเชิงวัตถุ (Object Oriented)

ตัวอย่างของภาษาในการเขียนโปรแกรมเชิงวัตถุ เช่น Smalltalk , Object Pascal , C++ , Eiffel , CLOS และ Java ส่วนภาษาในการเขียนโปรแกรมบนพื้นฐานของวัตถุก็ เช่น Ada เป็นต้น ซึ่งในการที่จะเขียนโปรแกรมด้วยภาษาเชิงวัตถุให้ได้อย่างมีประสิทธิภาพ ผู้เขียนโปรแกรมควรจะต้องมีการออกแบบด้วยหลักการพัฒนาระบบด้วยวิธีเชิงวัตถุเช่นกัน

2.4 ภาษาจาวา

จาวาเป็นภาษาโปรแกรมขั้นสูงที่พัฒนาขึ้นมาโดยบริษัท Sun Microsystems โดยมีวัตถุประสงค์เพื่อให้เป็นภาษาระดับสูงที่ใช้พัฒนาซอฟต์แวร์สำหรับอุปกรณ์ที่ควบคุมโดยไมโครโพรเซสเซอร์ (Embedded System) ที่สามารถนำไปรันได้โดยไม่ขึ้นกับแพลตฟอร์ม (Platform)

ภาษาจาวาได้รับการยอมรับว่าเป็นภาษาที่ดีที่สุดในการพัฒนาซอฟต์แวร์ที่ต้องการความปลอดภัยสูงแบบกระจายผ่านระบบเครือข่าย โดยสามารถใช้พัฒนาได้ตั้งแต่โปรแกรมของอุปกรณ์เน็ตเวิร์กที่ควบคุมโดยไมโครโพรเซสเซอร์ โปรแกรมในเว็บบราวเซอร์ จนถึงแอพพลิเคชันทั่วไป ตัวภาษาจาวามีลักษณะเด่นดังนี้

- เป็นภาษาพัฒนาโปรแกรมเชิงวัตถุ
- สามารถนำไปใช้งานได้โดยไม่ขึ้นกับแพลตฟอร์ม
- สามารถประมวลผลแบบกระจาย โดยดึงคลาสไลบรารีจากที่ต่างๆ ผ่านทาง HTTP และ FTP
- สามารถทำงานได้หลายงานพร้อมกันในเวลาเดียวกัน
- ไม่มีตัวแปรแบบพอยน์เตอร์ (Pointer) เพื่อขจัดปัญหาในการจัดการหน่วยความจำ
- สามารถตรวจสอบและจัดการกับความผิดพลาด (Exception) ที่เกิดขึ้น
- มีการรองรับมาตรฐานความปลอดภัยในการใช้งานรูปแบบต่างๆ

2.5 หลักการทำงานของภาษาจาวา

จาวาเป็นภาษาระดับสูงเชิงวัตถุซึ่งมีกลไกการแปลภาษาและการประมวลผลในลักษณะผสมผสานระหว่างแบบคอมไพเลอร์ (Compiler) และแบบอินเทอร์พรีเตอร์ (Interpreter) เพื่อจุดประสงค์ที่จะรับข้อดีของทั้งสองแบบ

แบบคอมไพเลอร์นั้น ตัวคอมไพเลอร์จะทำหน้าที่วิเคราะห์โค้ดภาษาระดับสูงและแปลเป็นภาษาเครื่อง (Machine code) ซึ่งจะมีข้อดีคือ จะได้โปรแกรมภาษาเครื่องที่มีประสิทธิภาพ เนื่องจากตัวคอมไพเลอร์ได้ทำการวิเคราะห์และปรับแต่งตัวโค้ดให้ได้ภาษาเครื่องที่มีประสิทธิภาพที่สุด แต่ข้อเสียของแบบนี้คือภาษาเครื่องที่ได้จะไม่สามารถนำไปใช้กับแพลตฟอร์มอื่นได้

ส่วนแบบอินเทอร์พรีเตอร์นั้น ตัวอินเทอร์พรีเตอร์จะทำการอ่านโค้ดภาษาระดับสูงทีละบรรทัด

แล้วแปลเป็นภาษาเครื่องและทำงานทันที แล้วก็จะกลับไปอ่านโค้ดบรรทัดถัดไป ทำอย่างนี้ไปจนจบโปรแกรม วิธีนี้จะต้องทำการอ่านโค้ดและแปลเป็นภาษาเครื่องสลับกันไป ทำให้ทำงานได้ช้ากว่าแบบคอมไพเลอร์ แต่ก็มีข้อดีคือตัวอินเทอร์พรีเตอร์สามารถสร้างได้ง่ายกว่าตัวคอมไพเลอร์

ภาษาจาวาได้นำข้อดีของทั้งสองแบบมารวมกัน โดยเมื่อทำการแปลโค้ด ตัวจาวาคอมไพเลอร์ (Java Compiler) จะทำการแปลโค้ดภาษาจาวาเป็นไบต์โค้ด (Byte Code) ที่ไม่ขึ้นกับแพลตฟอร์ม และเมื่อจะทำการประมวลผล ตัวจาวาเวอร์ชวลแมชชีน (Java Virtual Machine) จะทำการแปลไบต์โค้ดทีละคำสั่งเป็นภาษาเครื่องของแพลตฟอร์มที่กำลังทำงานแล้วส่งให้ตัวประมวลผลทำงานตามคำสั่ง

วิธีการนี้ทำให้โปรแกรมที่เขียนในภาษาจาวาสามารถทำงานได้ในลักษณะ " เขียนครั้งเดียวใช้งานได้ทุกที่ " คือไม่ว่าจะทำการคอมไพล์โค้ดในแพลตฟอร์มใด ก็สามารถนำไบต์โค้ดที่ได้ไปใช้กับแพลตฟอร์มอื่นที่มีตัวจาวาเวอร์ชวลแมชชีนสำหรับใช้ประมวลผล

2.6 จาวาเซิร์ฟเล็ต

จาวาเซิร์ฟเล็ตเป็นโปรแกรมที่เขียนขึ้นมาเป็นคลาสของภาษาจาวา เพื่อทำงานบนฝั่งเซิร์ฟเวอร์ โดยจาวาเซิร์ฟเล็ตนั้นจะทำการประมวลผลบนเครื่องเซิร์ฟเวอร์โดยจะทำงานควบคู่กับเครื่องของผู้ใช้ และเมื่อผู้ใช้ต้องการประมวลผลก็ส่งคำร้องให้จาวาเซิร์ฟเล็ตที่อยู่บนเซิร์ฟเวอร์ทำงานแล้วส่งผลลัพธ์กลับมาให้เครื่องของผู้ใช้อีกที ทำให้เครื่องของผู้ใช้ไม่จำเป็นต้องเป็นเครื่องที่มีประสิทธิภาพ

จาวาเซิร์ฟเล็ตนั้นอ้างอิงหลักการของ CGI ใช้สำหรับการเขียนโปรแกรมในลักษณะเดียวกับ CGI เพื่อทำหน้าที่อ่านข้อมูลที่รับมาจากผู้ชมเว็บไซต์ แล้วเอาข้อมูลนั้นมาประมวลผลจากนั้นจึงส่งผลลัพธ์กลับไปให้แก่ผู้ชมเว็บไซต์นั้น โดยข้อดีของจาวาเซิร์ฟเล็ตที่อยู่เหนือ CGI อย่างแรกก็คือตัวภาษาที่ใช้เขียนซึ่งก็คือภาษาจาวานั่นเอง นอกจากนี้จาวาเซิร์ฟเล็ตยังมีความเร็วที่เหนือกว่า CGI เพราะเซิร์ฟเล็ตใช้หลักการของเธรดโดยจะทำการสร้าง 1 เทรดต่อหนึ่งคำร้องที่มาจากไคลเอ็นต์ แต่ตัว CGI จะทำการสร้าง 1 โพรเซสต่อหนึ่งคำร้อง ซึ่งจะสิ้นเปลืองทรัพยากรของระบบมากกว่า

ถึงแม้ว่าเซิร์ฟเล็ตจะใช้หลักการของ CGI แต่ตัวเว็บเซิร์ฟเวอร์จะไม่สามารถส่งข้อมูลไปให้เซิร์ฟเล็ตโดยตรงได้เหมือนกับ CGI แต่ต้องทำการเพิ่มส่วนที่ใช้เป็นเสมือนตัวห่อหุ้มเซิร์ฟเล็ตต่างๆ ไว้ โดยส่วนที่เพิ่มขึ้นมานี้เราเรียกว่าเซิร์ฟเล็ตเอนจิ้น (Servlet Engine) หรือเซิร์ฟเล็ตคอนเทนเนอร์ (Servlet Container)

เนื่องจากเซิร์ฟเล็ตมีจุดเด่นที่สำคัญมากมาย เช่น มีประสิทธิภาพและความเร็วสูงในการทำงาน สามารถปรับปรุงแก้ไขและพัฒนาได้ง่ายเพราะใช้ภาษาจาวา ซึ่งเป็นภาษาเชิงวัตถุในการพัฒนา เป็นต้น จึงเป็นทางเลือกหนึ่งในการพัฒนา CGI และได้รับความนิยมนำมาพัฒนาเว็บแอพลิเคชั่นต่างๆ (Web Application)

2.7 จาวาบีน

จาวาบีนเป็นคอมโพเนนต์ของซอฟต์แวร์ (software component) ที่เขียนขึ้นด้วยภาษาจาวา หรือถ้าพูดอีกอย่างจาวาบีนก็เป็นคลาสของภาษาจาวานั่นเอง เราอาจจะเรียกจาวาบีนอย่างสั้นๆว่า บีน (Bean) ซึ่งบีนที่สร้างขึ้นมานี้จะต้องเป็นไปตามข้อกำหนดของ JavaBean API ที่บริษัท Sun ได้ร่วมมือกับบริษัทอื่นๆ ตั้งขึ้นมาเป็นมาตรฐานในการพัฒนาคอมโพเนนต์เพื่อใช้งาน

รูปแบบการใช้งานของคอมโพเนนท์ที่ถูกสร้างไว้แล้ว และนำมาประกอบกันเป็นซอฟต์แวร์ ถ้าเทียบไปแล้วจะคล้ายกับตัวแอคทีฟคอนโทรล (Active Control) ที่ใช้ในโปรแกรมภาษาวิชวลเบสิก (Visual Basic) แต่จาวาบีนจะสร้างด้วยภาษาจาวาเพื่อให้ใช้งานได้ทุกแพลตฟอร์ม ทำให้จาวาบีนมีการสร้างเพื่อขายเป็นการค้าจากผู้ผลิตรายอื่นๆ (Third Party) ด้วย จาวาบีนจะมีลักษณะดังนี้คือ

• เป็นคลาสในภาษาจาวาเพื่อเรียกใช้ได้ในโปรแกรมพัฒนาจำพวกวิชวล เช่น บีนเกี่ยวกับปุ่มบีนเกี่ยวกับรูปภาพ บีนเกี่ยวกับเมนู เป็นต้น ถ้าเปรียบเทียบกับวิชวลเบสิกก็คือสิ่งที่เรียกว่าคอนโทรลนั่นเอง

• เป็นส่วนประกอบทางกราฟฟิกที่สามารถนำมาประกอบกันเป็นโปรแกรมที่ติดต่อกับผู้ใช้แบบ GUI ได้ง่ายขึ้นสะดวกขึ้น เหมือนที่ใช้ในโปรแกรมภาษาวิชวลต่าง ๆ

• เป็นลักษณะคล้าย ๆ กล่องที่เราสามารถเปลี่ยนลักษณะการทำงานและคุณสมบัติต่าง ๆ ข้างในได้โดยเซตที่ตัวพร็อพเพอตี้ของมัน

บทที่ 3
จาวาเซิร์ฟเวอร์เพจ

จาวาเซิร์ฟเวอร์เพจ (JavaServer Pages : JSP) เป็นเทคโนโลยีที่ใช้ " สคริปต์ " ในการพัฒนาโปรแกรมประยุกต์เว็บเพจ (Web Application) เพื่อทำงานทางฝั่งเซิร์ฟเวอร์ (Sever-side-script) และส่งผลกลับมายังเว็บเบราเซอร์เป็นภาษา HTML เหมือนกับเทคโนโลยีอื่นๆ เช่น ASP , PHP เป็นต้น

การเขียนสคริปต์ JSP จะใช้ภาษาจาวาเพื่อเพิ่มประสิทธิภาพให้หน้าเว็บเพจมีความยืดหยุ่นมากขึ้น โครงสร้างของ JSP จะเป็นลักษณะของแท็ก (tag) ชนิดพิเศษที่แทรกเข้าไปในเอกสาร HTML และเปลี่ยนนามสกุลเป็น .jsp แทนที่จะเป็น .html โดยแท็กเหล่านี้เว็บบราวเซอร์ (web browser) จะไม่สามารถตีความได้ จะต้องนำไปประมวลผลก่อนที่เว็บเซิร์ฟเวอร์ก่อน แล้วนำผลลัพธ์ทั้งหมดส่งกลับมายังเว็บบราวเซอร์ในลักษณะของเอกสาร HTML ซึ่งเว็บบราวเซอร์สามารถตีความและนำมาแสดงผลได้

หัวใจของ JSP คือภาษาจาวา ซึ่งเป็นภาษาที่มีหัวใจหลักอยู่ที่ออบเจ็กต์ ซึ่งช่วยให้ง่ายต่อการพัฒนาในโปรเจ็กต์ใหญ่ๆ ตลอดจนสามารถนำเอาส่วนประกอบหรือคอมโพเนนต์ต่างๆ (component) กลับมาใช้ได้อีก ซึ่งเป็นประโยชน์อย่างยิ่งโดยเฉพาะในการพัฒนาโปรแกรมขนาดใหญ่

3.1 โครงสร้างและขั้นตอนการทำงานของ JSP

สิ่งที่มีบทบาทสำคัญในการทำงานของ JSP ได้แก่ JSP Container (หรือที่เรียกอีกอย่างหนึ่งว่า JSP Engine) ซึ่งเป็นส่วนประกอบสำคัญที่อยู่ในเว็บเซิร์ฟเวอร์ เพราะทำหน้าที่ควบคุมและประมวลผลไฟล์ JSP ที่มีการร้องขอ (request) เข้ามา และตอบสนอง (response) คำขอร้องนั้นกลับไปยังไคลเอนต์

ขั้นตอนการประมวลผลไฟล์ JSP ทั้งหมด แบ่งเป็น 8 ขั้นตอน ดังนี้

1 . ฝั่งไคลเอนต์ส่งคำร้องขอเอกสาร JSP ไปที่เว็บเซิร์ฟเวอร์

2 . เว็บเซิร์ฟเวอร์ตรวจสอบคำร้องขอ พบว่าเป็นไฟล์ JSP จึงส่งต่อไปให้แก่ JSP Container

3 . JSP Container ตรวจสอบว่าไฟล์ JSP ที่ร้องขอมา เคยแปลงเป็น Servlet หรือเปล่า และคอมไพล์เป็นไฟล์ .class อยู่หรือเปล่า ถ้ายังไม่มี ก็จะกระโดดข้ามไปทำงานตามขั้นตอนข้อ 4 ต่อ แต่ถ้ามีอยู่แล้ว ก็จะตรวจสอบต่ออีกว่า หลังจากที่แปลงไฟล์ JSP เป็น Servlet และ คอมไพล์เป็นไฟล์ .class ครั้งล่าสุดแล้ว ไฟล์ JSP นั้นจะมีการเปลี่ยนแปลงแก้ไขหรือเปล่า ถ้ามีการแก้ไข ก็จะกระโดดไปทำงานขั้นตอนข้อ 4 ต่อเช่นกัน แต่ถ้าไม่มีการแก้ไข แสดงว่า ไฟล์ JSP นั้นยังคงเดิมไม่เปลี่ยนแปลงจึงข้ามไปทำงานขั้นตอนข้อ 6 ได้เลย

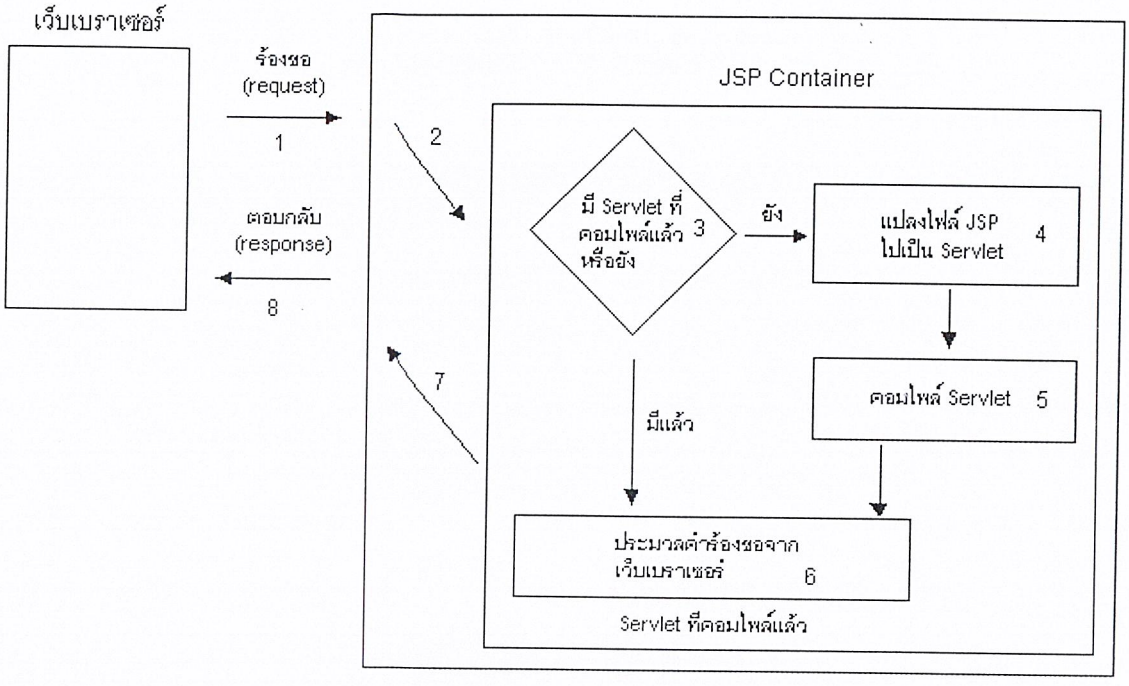
4 . JSP Container แปลงไฟล์ JSP เป็น Servlet

5 . JSP Container คอมไพล์ไฟล์ Java Servlet เป็นไฟล์ .class

6 . JSP Container ประมวลผลตามคำร้องขอนั้น

7 . JSP Container ส่งผลลัพธ์ที่ได้จากการประมวลผล ให้แก่เว็บเซิร์ฟเวอร์

8 . เว็บเซิร์ฟเวอร์ส่งผลลัพธ์นั้นไปยังไคลเอนต์หรือเว็บเบราเซอร์อีกทอดหนึ่ง



รูปที่ 3-1รูปแสดงโครงสร้างและขั้นตอนการประมวลผลไฟล์ JSP

จากขั้นตอนการประมวลผลไฟล์ JSP ข้างต้น สามารถแบ่งออกได้เป็น 2 ช่วงหลักๆ คือ

1 . ช่วง translation ได้แก่ขั้นตอนข้อ 4 และข้อ 5 ซึ่งเป็นการแปลงเอกสาร JSP (ไฟล์ .jsp) ให้เป็น Servlet (ไฟล์ .java) จากนั้นก็จะคอมไพล์ไฟล์ Servlet ให้เป็นไฟล์ .class

2 . ช่วง execution ได้แก่ขั้นตอนข้อ 6 ซึ่งเป็นการนำเอาไฟล์ .class ที่ได้จากการคอมไพล์ มาประมวลผลหรือทำงานตามคำร้องขอจากไคลเอนต์นั่นเอง

3.2 เปรียบเทียบ JSP กับเทคโนโลยีอื่นๆ

3.2.1 JSP กับ Servlet

Servlet เป็นเทคโนโลยีที่เริ่มมาก่อน JSP และใช้ภาษาจาวามาเป็นพื้นฐาน การทำงานของทั้งสองก็คล้ายกัน แต่ JSP จะมีขั้นตอนที่เพิ่มขึ้นมา คือ จะต้องแปลงเอกสาร JSP ให้เป็น Servlet ก่อน ผลลัพธ์สุดท้ายที่ได้ก็เป็น Servlet เหมือนกัน

Servlet จะยุ่งยากกว่าตรงที่ไม่สามารถแทรกแท็ก HTML เข้าไปได้ ต้องพิมพ์เข้าไปเอง โดยคำสั่ง out.print() แต่ JSP สามารถนำแท็ก HTML รวมกับแท็ก JSP ได้เลย ซึ่งจะสะดวกในการใช้งานมากกว่า Servlet

3.2.2 JSP กับ ASP

ความสามารถโดยรวมของ JSP จะคล้ายกับ ASP มาก เพราะสามารถแทรกแท็กเข้าไปในเอกสาร HTML ได้เหมือนกัน ต่างกันที่ JSP พัฒนามาบนพื้นฐานของภาษาจาวา มีข้อดีเหนือภาษา Visual Basic คือ สามารถจัดการข้อผิดพลาดและการทำงานในลักษณะ MultiThreading และ JSP สามารถทำงานได้ทุกระบบปฏิบัติการ

3.2.3 JSP กับ JavaScript

JavaScript เป็นการทำเนื้อหาไดนามิกให้กับเว็บเหมือนกัน แต่ JavaScript จะเกิดการประมวลผลที่ไคลเอ็นต์ ถ้าเว็บบราวเซอร์ไม่สนับสนุนจาวาสคริปต์ก็จะทำให้ไม่สามารถทำงานได้ แต่ JSP จะเกิดการประมวลผลที่เว็บเซิร์ฟเวอร์โดยตรงทำให้ไม่มีปัญหากับเว็บบราวเซอร์

3.3 แท็กพื้นฐานและคำสั่งควบคุม

ในการเขียนสคริปต์ก็คือ แทรกสคริปต์ลงไปในเอกสาร HTML โดยเรียกแท็กของ JSP ว่า scripting element หรือแท็กที่ใช้ในการเขียนสคริปต์ JSP แบ่งออกเป็น 3 ประเภท ได้แก่ scriptlet , expression และ declaration โดยที่และประเภทมีรูปแบบการใช้งานที่แตกต่างกัน

1 . scriptlet เป็นแท็กที่ใช้สำหรับเขียนคำสั่งภาษา Java ทั่วๆไป ซึ่งมีรูปแบบการเขียนดังนี้

<% statement; %>

สเตตเมนต์ (statement) คือประโยคคำสั่งที่เขียนขึ้นมาเพื่อทำงานอย่างใดอย่างหนึ่ง เช่นการคำนวณ , การใช้คำสั่งเงื่อนไข , การเรียกใช้เมธอดต่างๆ เป็นต้น โดยจะต้องมีเครื่องหมาย ; ปิดท้ายสเตตเมนต์แต่ละประโยคเสมอ

2 . expression เป็นแท็กที่ใช้สำหรับแสดงค่าของตัวแปรหรือเมธอดในรูปแบบย่อ

<%= expression %>

เอ็กซ์เพรสชัน (expression) อาจจะเป็นตัวแปรโดดๆหรือการคำนวณตัวแปรหลายตัว หรืออาจจะเป็นเมธอดก็ได้ สังเกตว่าการระบุเอ็กซ์เพรสชัน ไม่จำเป็นต้องปิดท้ายด้วยเครื่องหมาย ;

3 . declaration เป็นแท็กที่ใช้ในการประกาศตัวแปรหรือเมธอดต่างๆ ที่จะนำมาใช้ในงานสคริปต์จะต้องมีการประกาศขึ้นมาก่อน

<%! declaration %>

ดีแคลเรชัน (declaration) อาจเป็นประโยคของการประกาศตัวแปรหรือประกาศเมธอดก็ได้ ถ้าเป็นการประกาศตัวแปร จะต้องปิดท้ายด้วยเครื่องหมาย ; เสมอ ส่วนการประกาศเมธอดไม่ต้องปิดท้ายด้วยเครื่องหมาย ;

แท็กทั้ง 3 แบบที่ได้อธิบายไปข้างต้น สามารถประกาศไว้ที่ตำแหน่งใดก็ได้ภายในเอกสาร HTML อาจจะประกาศไว้ก่อนแท็ก <html> หรือประกาศภายในแท็ก <html> ก็ได้และในหมู่แท็ก scriptlet , expression , declaration ด้วยกันเอง เราจะระบุแท็กแบบไหนก่อน-หลังก็ได้ ในเอกสาร HTML

3.4 แท็กหมายเหตุ

ในการเขียนข้อความหมายเหตุหรือคอมเมนต์ (comment) ลงในไฟล์เอกสาร JSP สามารถเขียนได้ 3 รูปแบบ

1 . เขียนคอมเมนต์ในรูปแท็กที่ JSP กำหนดมาให้แต่ห้ามเขียนแทรกไว้ในแท็ก scriptlet , expression และ declaration ต้องเขียนไว้ลอย ๆ ดังนี้

< % - comment - % >

2 . เขียนคอมเมนต์ตามรูปแบบของภาษา Java ซึ่งเขียนได้ทั้งใน scriptlet , expression และ declaration เช่นถ้าเขียนคอมเมนต์ในแท็ก scriptlet จะมีลักษณะดังนี้

```
<%
    //      comment 1 line
    /*      comment many line
            comment many line      */
%>
```

3 . เขียนคอมเมนต์ในรูปแบบของเอกสาร HTML หรือ XML แต่ห้ามเขียนแทรกไว้ใน scriptlet , expression และ declaration

<!- - comment in html tag - >

3.5 Directive กับ Action

เป็นแท็กที่ใช้แทรกเข้าไปในเอกสาร HTML ทำให้กลายเป็นเอกสาร JSP เหมือนกัน scripting element เพียงแต่จุดประสงค์ในการใช้งานจะแตกต่างกันออกไป คือ เป็นแท็กที่ใช้กำหนดคุณสมบัติของไฟล์ JSP ที่สร้างขึ้นมา รวมทั้งใช้ในการติดต่อเชื่อมโยงกับออบเจกต์ต่างๆภายนอกด้วย เช่น การเรียกใช้ไฟล์ภายนอก , การสร้างแท็กใช้เอง , การ forward จากสคริปต์เดิมไปยังสคริปต์อื่นๆ

3.5.1 Directive

Directive มาตรฐาน มีอยู่ 3 ประเภท คือ page directive , include directive และ taglib directive ซึ่งมีการเขียนเหมือนกัน คือ ขึ้นต้นด้วย <%@ และปิดท้ายด้วย %>

1 . page directive ใช้สำหรับกำหนดคุณสมบัติ (attribute) ต่าง ๆ ของเอกสาร JSP แอตทริบิวต์ที่กำหนดได้มี 12 รูปแบบ อาทิ info , language , import , contentType เป็นต้น ไฟล์ของเอกสาร JSP แต่ละไฟล์สามารถกำหนดแอตทริบิวต์ได้เพียงครั้งเดียวเท่านั้น ยกเว้น import สามารถกำหนดได้หลายๆครั้ง ตัวอย่างเช่น

<%@ language = ”javascript” %>

<%@ import = “java.util.Random” %>

2 . include directive ไฟล์ JSP ที่มีการเรียกใช้ include directive หมายความว่า ให้นำเอาเนื้อความจากไฟล์อื่นๆที่ระบุไว้ใน include directive มาแทรกไว้ในไฟล์ JSP นั้น ณ ตำแหน่งที่วาง include directive โดยไฟล์ที่ถูก include เข้ามา อาจจะเป็นไฟล์ HTML หรือไฟล์ JSP เหมือนกันก็ได้ ตัวอย่างเช่น

<%@ include file = “header.jsp” %>

หมายความว่า header.jsp เป็นไฟล์ที่ถูก include เข้ามา เราสามารถวาง include directive ไว้ตำแหน่งใดในไฟล์ JSP ก็ได้

3 . taglib directive เป็นการบอก JSP Container ให้รู้ว่า ภายในสคริปต์ JSP จะมีการใช้แท็กที่สร้างขึ้นมาใหม่

3.5.2 Action

Action เป็นแท็กทีใช้สำหรับควบคุมการส่งข้อมูลระหว่างเพจแต่ละเพจ รวมถึงการติดต่อกับออบเจกต์อื่นๆด้วย เช่น Java Applet หรือ คอมโพเนนท์ JavaBean เป็นต้น แท็ก action แบ่งออกเป็น 4 ประเภท คือ forward action , include action , plugin action และ JavaBean action ตัวอย่างเช่น

1 . forward action ใช้สำหรับส่งคำร้องขอจากไฟล์ JSP ไปยังไฟล์อื่นๆ ตามที่ระบุไว้

<jsp : forward page = “destination file” />

2 . include actionใช้แก้ปัญหาของ include directive ที่ไม่สามารถล่วงรู้การเปลี่ยนแปลงแก้ไขไฟล์ที่ include เข้ามา

<jsp : include page = “header.jsp” flush = “false” >

3 . plugin action ใช้สำหรับแทรกคอมโพเน็นท์ (component) อื่นๆเข้ามาในไฟล์ JSP

<jsp : plugin type = “applet” code = “myapplet.class” width = “475” height = “350”>

</jsp : plugin>

4 . JavaBean action ใช้เมื่อต้องการนำเอา JavaBean มาใช้ร่วมกับไฟล์ JSP โดย action มาตรฐานอยู่ 3 แบบ คือ <jsp : useBean> , <jsp : setProperty> และ <jsp : getProperty>

3.6 Implicit Object

เป็นอ็อบเจกต์ที่ใช้งานบ่อยๆ ดังนั้น JSP Container จึงช่วยอำนวยความสะดวกโดยการสร้างอ็อบเจกต์เหล่านั้นขึ้นมาโดยอัตโนมัติ ทำให้เราสามารถเรียกใช้งานได้ทันทีโดยไม่ต้องประกาศก่อนใช้งาน มีอยู่ทั้งหมด 9 อ็อบเจกต์

อ็อบเจกต์	สร้างขึ้นมาจากคลาส...
request	javax.servlet.http.HttpServletRequest
response	javax.servlet.http.HttpServletResponse
session	javax.servlet.http.HttpSession
application	javax.servlet.ServletContext
out	javax.servlet.jsp.JspWriter
exception	java.lang.Throwable
pageContext	javax.servlet.jsp.PageContext
config	javax.servlet.ServletConfig
page	java.lang.Object

ตารางที่ 3-1 แสดง Implicit Object

บทที่ 4
การวิเคราะห์และการออกแบบระบบ

นักพัฒนาโปรแกรมระดับอาชีพจะเริ่มต้นพัฒนาระบบอย่างเป็นขั้นตอนและมีหลักการ คือ ขั้นตอนการวิเคราะห์และออกแบบ ทั้งนี้การศึกษาและการใช้กระบวนการพัฒนาเชิงวัตถุต่างๆ จำเป็นต้องมีความเข้าใจแนวคิดเชิงวัตถุอย่างแม่นยำ ในระบบซอฟต์แวร์ก็เช่นเดียวกัน ผู้ที่จะพัฒนาระบบซอฟต์แวร์ที่ต้องการใช้ภาษาในการเขียนโปรแกรมเชิงวัตถุ (OOP) ก็ควรจะต้องมีการวิเคราะห์และออกแบบระบบซอฟต์แวร์ด้วยกระบวนการพัฒนาเชิงวัตถุเช่นกัน ปัจจุบันมีวิธีหรือกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุมากมายหลายวิธี โดยเฉพาะกระบวนการที่มีความเกี่ยวข้องกับภาษายูแอมแอล

4.1 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ

วัตถุประสงค์ของทุกกระบวนการพัฒนาซอฟต์แวร์ คือ การแปลงความต้องการของผู้ใช้ให้เป็นระบบที่มีคุณภาพและสามารถใช้งานได้จริง อีกทั้งยังช่วยลดระยะเวลาที่ต้องทำการเขียนโปรแกรมและแก้ไขข้อผิดพลาดให้น้อยลงด้วย กระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุประกอบไปด้วย 5 ขั้นตอน

4.1.1 การวิเคราะห์ความต้องการของระบบหรือผู้ใช้ (Requirement Analysis)

การวิเคราะห์ความต้องการของผู้ใช้เป็นการค้นหาขอบข่ายของระบบ และเป็นการเตรียมข้อมูลด้านความสามารถของระบบ (System Function) จากมุมมองภายนอกที่จะต้องถูกทำการพัฒนาโดยไม่คำนึงถึงรายละเอียดหรือกรรมวิธีทางเทคนิคต่างๆ

ในการเริ่มต้นเฟสนี้เป็นการกำหนดข้อตกลงระหว่างผู้ใช้งานกับผู้พัฒนา ซึ่งฝ่ายพัฒนาจะต้องเก็บความต้องการของผู้ใช้อย่างละเอียด เรียกขั้นตอนนี้ว่า User Requirement Elicitation

4.1.2 การวิเคราะห์ระบบ (Domain Analysis)

การวิเคราะห์ระบบ (Domain Analysis) หรือ OOA (OO Analysis) เป็นการวิเคราห์โครงสร้าง (Structure) และพฤติกรรม (Behavior) ของระบบที่จะทำการพัฒนาซึ่งจะถูกนำไปกำหนดรายละเอียดเชิงเทคนิคในเฟสการออกแบบและถูกสร้างจริงเป็นลำดับต่อไปในการอิมพลีเมนต์ระบบ

4.1.3 การออกแบบระบบ (Design)

การออกแบบระบบ (Design) หรือ OOD(OO Design) เป็นการคิดค้นวิธีแก้ปัญหาหรือพิจารณารายละเอียดเชิงเทคนิคเพื่อเตรียมที่จะอิมพลีเมนต์ระบบขึ้นจริง ซึ่งเป็นการนำผลการวิเคราะห์จากเฟสสองมาทำการแก้ไขเพิ่มเติมรายละเอียดเชิงเทคนิคเพื่อสามารถที่จะถูกนำไปสร้างขึ้นเป็นระบบซอฟต์แวร์จริงได้อย่างสมบูรณ์ ดังนั้นในเฟสนี้จะต้องอาศัยความรู้ทางเทคนิคและเทคโนโลยีต่างๆ ที่มีอยู่เพื่อสามารถเลือกได้อย่างเหมาะสม

4.1.4 การสร้างโปรแกรมระบบ (Construction , Coding , Implementation)

โดยส่วนใหญ่กิจกรรมในเฟสนี้จะเป็นการสร้างโปรแกรมหรือการอิมพลีเมนต์ระบบอันเป็นขั้นตอนของ OOP(OO Programming) ซึ่งต้องอาศัยความรู้ในตัวภาษาโปรแกรมที่ใช้ในการสร้างโค้ดในขั้นตอนนี้จะถูกดำเนินการภายหลังจากที่ได้รับข้อมูลการออกแบบเพียงพอ

4.1.5 การทดสอบระบบ (Testing)

เป็นการทดสอบความถูกต้องของระบบที่พัฒนาเพื่อค้นหาข้อผิดพลาดเชิงเทคนิค และการตรวจสอบความสอดคล้องกับความต้องการของผู้ใช้งาน ทั้งนี้การค้นพบข้อผิดพลาดถืออว่าเป็นความสำเร็จของการทำงานเฟสนี้ นอกจากนี้ยังเป็นการประเมินความสมบูรณ์ของระบบว่าจำเป็นต้องทำการวิเคราะห์ออกแบบเพิ่มเติมอีกหรือไม่ โดยจะมีการจัดเตรียมข้อมูลที่ใช้สำหรับการทดสอบและประเมินผลลัพธ์เรียกว่า Test Cases ซึ่งใช้ในการทดสอบส่วนต่างๆของระบบในทุกแง่ทุกมุมของการทำงานทั้งหมดที่เป็นไปได้ ผลของการทดอบจะถูกบันทึกลงรายงานการทดสอบซึ่งรวมถึงการบรรยายรายละเอียดข้อผิดพลาดที่ปรากฏเพื่อทำการแก้ไขต่อไป

4.2 Unified Modeling Language

Unified Modeling Language หรือ UML เป็นภาษาในการโมเดลมาตรฐาน วัตถุประสงค์เบื้องต้นเป็นการพัฒนากระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุที่เป็นหนึ่งเดียวกัน (Unified Method) ต้องการสร้างโมเดลในการพัฒนาที่เข้าใจได้และสร้างได้ง่าย และสามารถนำไปใช้ได้กับทุกๆระบบ ใน UML มีโมเดลที่สื่อสารด้วยภาพได้สำหรับระบบหลายๆโมเดล โดยแต่ละโมเดลก็จะแสดงมุมมองต่อระบบที่ไม่เหมือนกัน ประกอบไปด้วยไดอะแกรมต่างๆ 8 ไดอะแกรมให้เลือกใช้ได้ตามความเหมาะสม

- ยูสเคสไดอะแกรม (Use Case Diagram) ใช้ในการโมเดลฟังก์ชันการทำงานของระบบ
- คลาสไดอะแกรม (Class Diagram) ใช้ในการโมเดลคลาสต่างๆที่จำเป็นในระบบ
- แอ็กทิวิตี้ไดอะแกรม (Activity Diagram) มีหลักการเดียวกันกับโฟลว์ชาร์ต (Flowchart)
- สเตตชาร์ตไดอะแกรม (Statechart Diagram) ใช้สำหรับแสดงถึงสถานะของออบเจกต์ในระหว่างการทำงาน
- คอลแลบอเรชั่นไดอะแกรม (Collaboration Diagram) ใช้ในการแสดงการทำงานร่วมกันของออบเจ็กต์ในระบบ
- ซีเควนต์ไดอะแกรม (Sequence Diagram) ใช้ในการโมเดลกิจกรรมต่างๆที่เกิดขึ้นกับออบเจ็กต์ในระบบ
- คอมโพเนนต์ไดอะแกรม (Component Diagram) ใช้สำหรับสร้างโมเดลของคอมโพเนนต์ในระบบ
- ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram) ใช้แสดงการติดตั้งใช้งานส่วนประกอบต่างๆของระบบ

4.3 ไดอะแกรมที่เป็นโครงสร้างแบบสแตติก (Static Diagram)

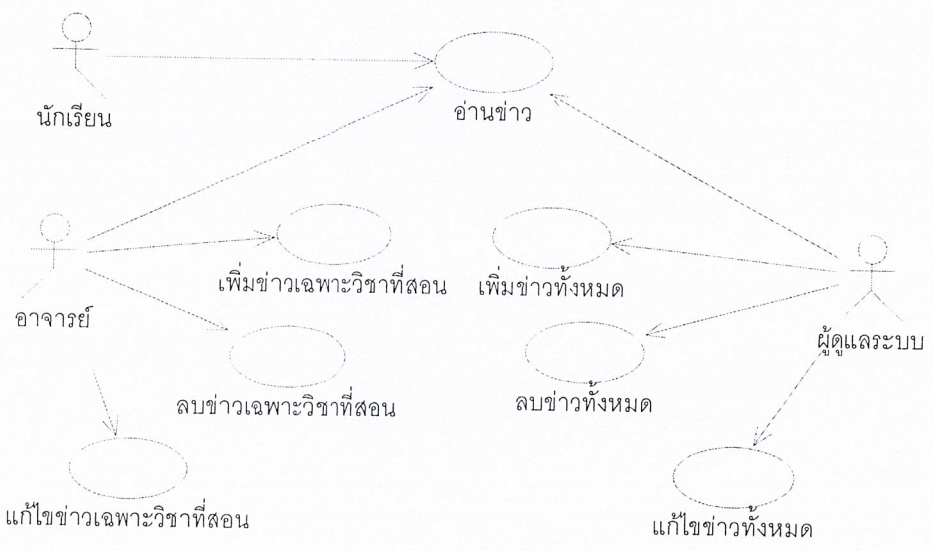
คือเป็นโครงสร้างการทำงานที่หยุดนิ่ง ไม่มีการเปลี่ยนแปลงตามเวลา ซึ่งได้แก่ ยูสเคสไดอะแกรม (Use Case Diagram) และคลาสไดอะแกรม (Class Diagram)

4.3.1 ยูสเคสไดอะแกรม (Use Case Diagram)

ในการสร้างยูสเคสไดอะแกรมสิ่งสำคัญ คือ การค้นหาว่าระบบทำอะไรได้บ้าง โดยไม่สนใจว่าข้างในสิ่งที่ระบบต้องทำได้เหล่านั้นมีกลไกการทำงานอย่างไร หรือใช้เทคนิคการสร้างอย่างไร เปรียบเสมือนเป็น" กล่องดำ "(Black Box)

ส่วนประกอบของยูสเคสไดอะแกรมจะมีอยู่ 3 ส่วนคือ

- ยูสเคส (Use Case) คือความสามารถหรือฟังก์ชันที่ระบบซอฟต์แวร์จะต้องทำได้ เช่น ค้นหาข้อมูลหนังสือ ยืมหนังสือ คืนหนังสือ บันทึกรายการหนังสือ ลบรายการหนังสือ ฯลฯ
- แอ็กเตอร์ (Actor) คือ ผู้ที่กระทำกับยูสเคสหรือใช้งานยูสเคสนั้นๆ เช่น นักศึกษา(สามารถยืม คืน ค้นหนังสือ) บรรณารักษ์(สามารถเพิ่ม ลบรายการหนังสือ) เป็นต้น
- เส้นแสดงความสัมพันธ์ (Relationship) เป็นการเชื่อมโยงระหว่างยูสเคสกับแอ็กเตอร์



รูปที่ 4-1 แสดง ยูสเคสไดอะแกรม

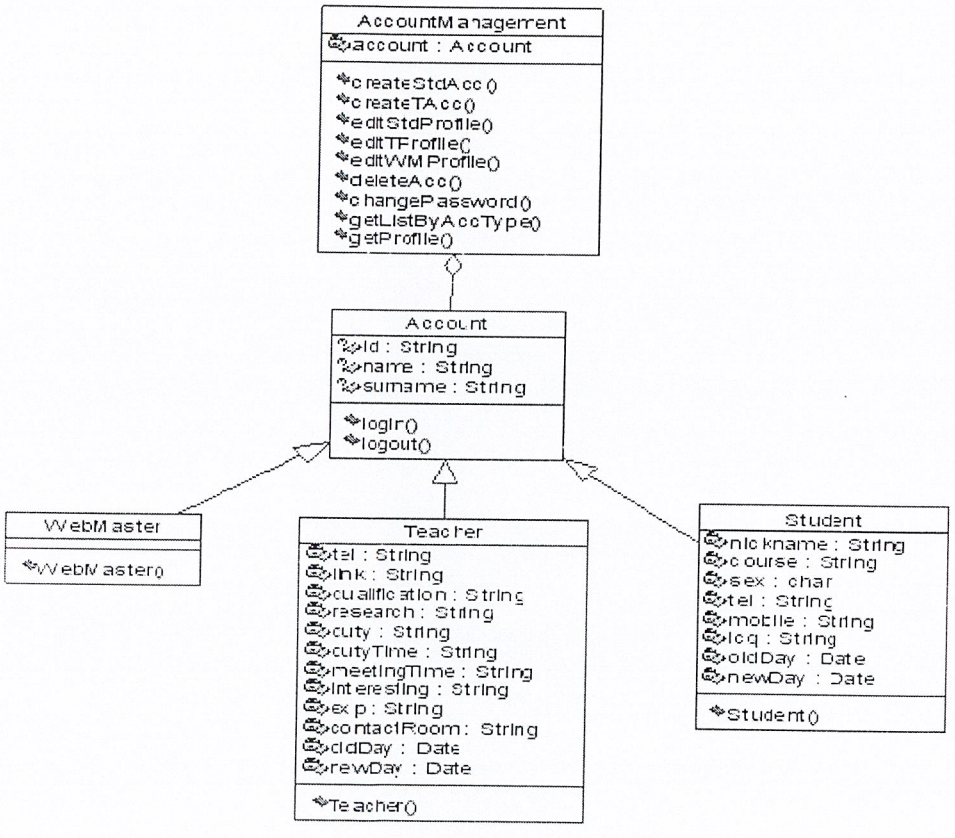
จากรูปจะพบว่ามีนักเรียน อาจารย์ และผู้ดูแลระบบ เป็นแอ็กเตอร์ โดยแอ็กเตอร์เหล่านี้จะใช้งานหรือกระทำกับยูสเคสที่แอ็กเตอร์ชี้เข้าไป จะมีตัวหนังสือที่แอ็กเตอร์และยูสเคสเพื่อบ่งบอกชื่อของแต่ละองค์ประกอบ

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

4.3.2 คลาสไดอะแกรม (Class Diagram)

ในการพัฒนาซอฟต์แวร์เชิงวัตถุ จะมีการใช้งานคลาส ออบเจ็กต์ และมีการสร้างความสัมพันธ์ระหว่างคลาสหรือออบเจ็กต์เหล่านั้น ดังนั้นการโมเดลระบบเชิงวัตถุจึงมีความจำเป็นอย่างยิ่งที่จะต้องสร้างคลาสไดอะแกรมที่แสดงถึงองค์ประกอบดังกล่าวทั้งหมดอย่างชัดเจน เราเรียกไดอะแกรมนี้ว่า คลาสไดอะแกรม

วัตถุประสงค์ของการสร้างคลาสไดอะแกรมก็เพื่อแสดงถึงโครงสร้างของระบบอันประกอบไปด้วยคลาสต่างๆ และความสัมพันธ์ระหว่างคลาสเหล่านั้น ซึ่งจะถูกใช้เป็นไดอะแกรมหลักในการสร้างไดอะแกรมอื่นๆ อีกหล่ายประเภท เมื่อนำไปเขียนโค้ดในการแปลงคลาสไดอะแกรมไปเป็นโค้ดนั้นจะค่อนข้างง่ายและตรงไปตรงมาทั้งนี้เนื่องจากภาษาโปรแกรมเชิงวัตถุจะมีวากยสัมพันธ์ (Syntax) ที่ใช้ในการอิมพลีเมนต์คลาสโดยตรง



รูปที่ 4-2 แสดง คลาสไดอะแกรม

จากรูปจะเป็นสัญลักษณ์ของคลาส ตัวอย่างเช่น Account จะมี

1. ชื่อคลาส คือ Account
2. ส่วนแอตทริบิวต์ จะมี id , name และ surname
3. ส่วนโอเปอร์เรชั่น จะมี login() , logout()

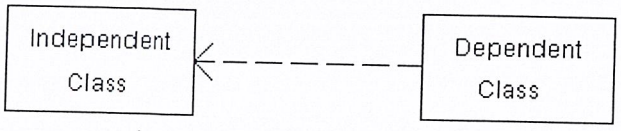
61846

และมีเส้นแสดงความสัมพันธ์กันระหว่างคลาส (Relationships) และรวมถึงความสัมพันธ์ระหว่างออบเจ็กต์ของคลาสด้วย ซึ่งความสัมพันธ์นี้สามารถเป็นไปได้ 3 รูปแบบ

4.4 ความสัมพันธ์กันระหว่างคลาส

4.4.1 Dependency

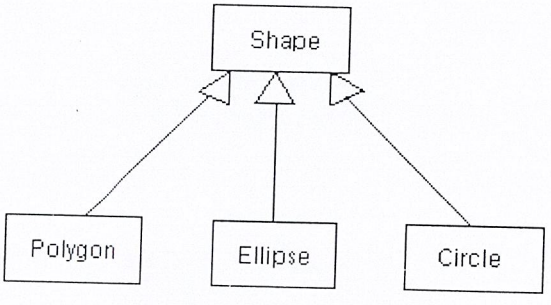
Dependency หรือความสัมพันธ์แบบพึ่งพิง ความสัมพันธ์แบบนี้จะเกิดขึ้นเมื่อการเปลี่ยนแปลงที่เกิดขึ้นกับคลาสที่ถูกพึ่งพิง (Independent Class) จะส่งผลต่อคลาสที่พึ่งพิง (Dependent Class)



รูปที่ 4-3 แสดง ความสัมพันธ์แบบพึ่งพิง

4.4.2 Generalization

Generalization คือความสัมพันธ์ระหว่างซูเปอร์คลาสและซับคลาส



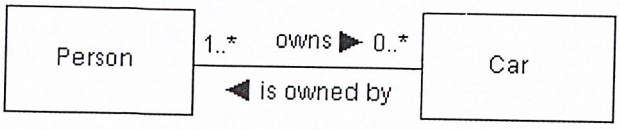
รูปที่ 4-4 แสดง ความสัมพันธ์ระหว่างซูเปอร์คลาสและซับคลาส

4.4.3 Association

Association เป็นความสัมพันธ์อีกชนิดหนึ่งระหว่างคลาสซึ่งแบ่งได้ดังนี้คือ

4.4.3.1 Normal Association

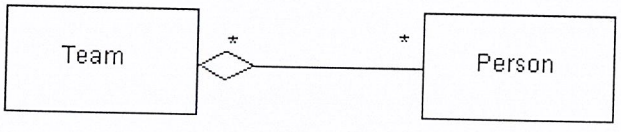
มักใช้ในการโมเดลระบบที่ซับซ้อนโดยเฉพาะระบบสารสนเทศ นอกจากนี้ยังมีการกำหนดปริมาณของคลาสหรือออบเจ็กต์ที่สัมพันธ์กันอยู่ เรียกว่า Multiplicity ซึ่งสามารถกำหนดได้หลายรูปแบบ เป็นตัวเลขใส่ไว้ที่ปลายด้านใดด้านหนึ่งของเส้นความสัมพันธ์ เช่นตัวอย่าง สามารถอ่านได้ว่า “A person owns many cars” และ “A car can be owned by many persons”



รูปที่ 4-5 แสดง ความสัมพันธ์ระหว่างคลาสแบบ Normal Association

4.4.3.2 Aggregation

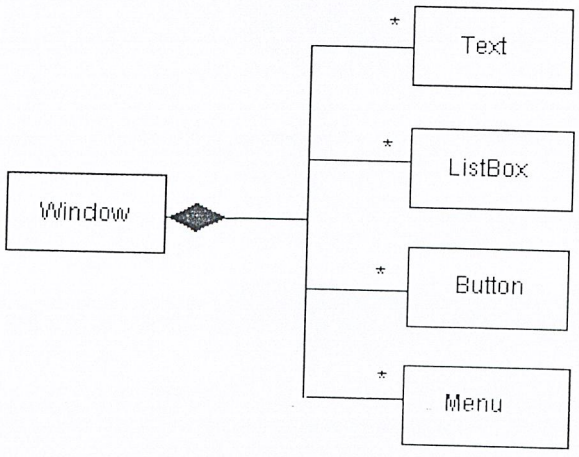
เป็นความสัมพันธ์ระหว่างคลาสหรือออบเจ็กต์ในแง่ของการรวมกันหรือการประกอบกัน สามารถอธิบายได้ตามรูป คือ แต่ละทีมประกอบด้วยสมาชิกหลายคน ในทางกลับกันแต่ละคนอาจสังกัดอยู่มากกว่าหนึ่งทีม



รูปที่ 4-6 แสดง ความสัมพันธ์ระหว่างคลาสแบบ Aggregation Association

4.4.3.3 Composition

คล้ายคลึงกับความสัมพันธ์แบบ Aggregation หากแต่คลาสที่เป็นองค์ประกอบจะเป็นส่วนหนึ่งของคลาสใหญ่กว่า และเมื่อคลาสที่ใหญ่กว่าถูกทำลาย คลาสที่เป็นองค์ประกอบก็จะถูกทำงายไปด้วยพร้อมๆกัน เช่นดังรูป คลาสวินโดวส์ จะประกอบไปด้วย คลาสปุ่ม คลาสเมนู เป็นต้น ซึ่งคลาสเหล่านั้นจะอาศัยอยู่ในวินโดวส์ทั้งหมด และเมื่อวินโดวส์ถูกปิดลง ปุ่ม เมนู ก็จะหายไปพร้อมๆกันด้วย นอกจากนี้ในความสัมพันธชนิดนี้ ค่า Multiplicity ของฝั่งที่ใหญ่กว่าจะต้องเป็น 1 เท่านั้น



รูปที่ 4-7 แสดง ความสัมพันธ์ระหว่างคลาสแบบ Composition Association

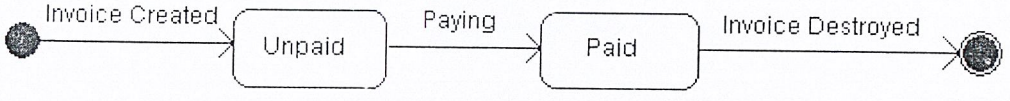
4.5 ไดอะแกรมที่เป็นโครงสร้างแบบไดนามิก (Dynamic Diagram)

จะใช้สำหรับการบรรยายพฤติกรรมของระบบที่มีการเปลี่ยนแปลงตามเวลาในขณะที่ระบบกำลังทำงาน ได้แก่ สเตตชาร์ตไดอะแกรม (Statechart Diagram) , คอลแลบอเรชั่นไดอะแกรม (Collaboration Diagram) , ซีเควนต์ไดอะแกรม (Sequence Diagram) และแอ็กทิวิตี้ไดอะแกรม (Activity Diagram) ทั้งหมดนี้จะเรียกโดยรวมว่า บีเฮฟเยอร์ไดอะแกรม (Behavior Diagram) ซึ่งก็คือ ไดอะแกรมที่บ่งบอกพฤติกรรมของระบบ

4.5.1 สเตตชาร์ตไดอะแกรม (Statechart Diagram)

จะบ่งบอกถึงสถานะต่างๆของคลาสในระบบว่ามีสถานะอะไรบ้าง ซึ่งเมื่อเวลาผ่านไป สถานะของคลาสจะสามารถเปลี่ยนไปได้ พฤติกรรมหรือสถสนะต่างๆเหล่านี้ย่อมเปลี่ยนไปได้เสมอ เช่นในระบบลิฟต์ ณ เวลาหนึ่งลิฟต์กำลังขึ้นนั่นคือ อยู่ในสถานะขึ้น (State Up) ถ้าลิฟต์กำลังลง ก็คืออยู่ในสถานะลง (State Down) เป็นต้น

สเตตชาร์ตไดอะแกรมในยูเอ็มแอลจะมีจุดเริ่มต้นสถานะและจุดสิ้นสุดสถานะ เส้นลูกศรที่ชี้จากสถานะหนึ่งไปยังอีกสถานะหนึ่ง สามารถเขียนคำอธิบายเหตุการณ์ที่ทำให้เปลี่ยนแปลงสถานะตรงลูกศรได้ จากรูปจะเป็นตัวอย่างแสดง สเตตชาร์ตไดอะแกรมของคลาสใบแจ้งราคาสินค้า (Invoice) ซึ่งมีสถานะคือยังไม่ได้จ่าย (Unpaid) และจ่ายแล้ว (Paid)

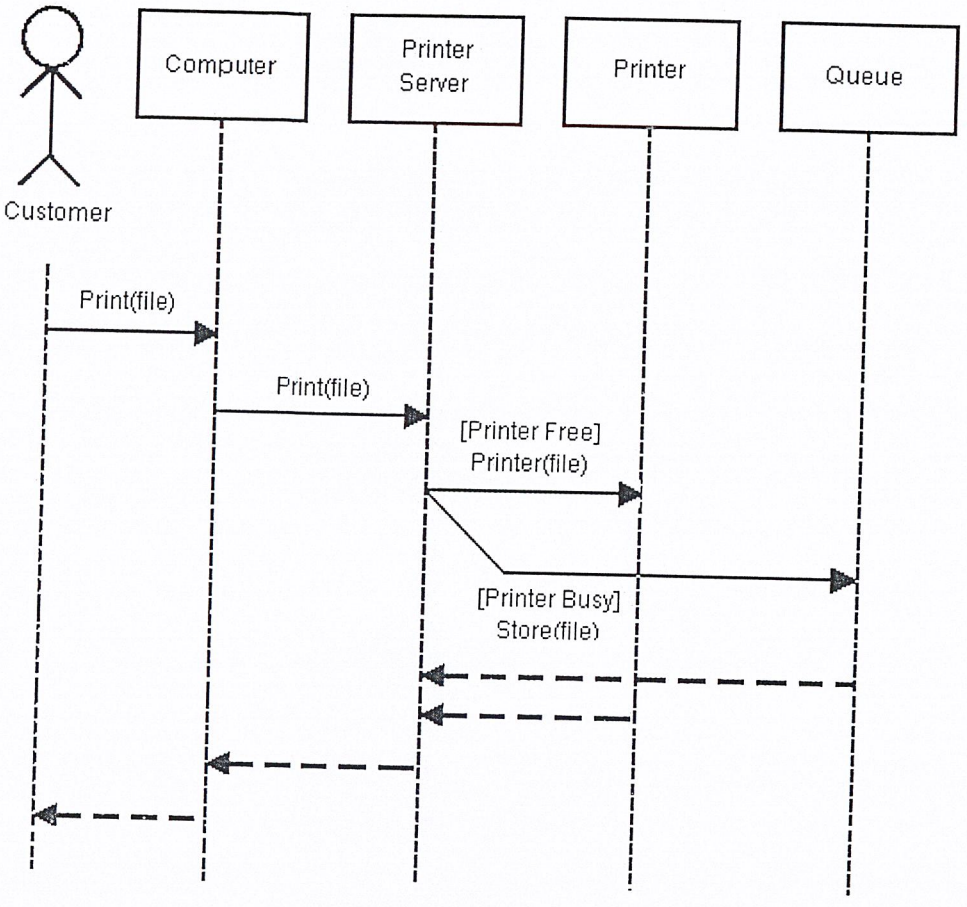


รูปที่ 4-8 แสดง สเตตชาร์ตไดอะแกรม

4.5.2 ซีเควนต์ไดอะแกรม (Sequence Diagram)

ซีเควนต์ไดอะแกรมจะบอกว่าใน Use Case นั้น วัตถุแต่ละตัวจะติดต่อสื่อสารกันอย่างไร มีขั้นตอนการทำงานอย่างไร โดยจะเน้นไปที่แกนเวลาเป็นสำคัญ ถ้าเวลาเปลี่ยน ขั้นตอนการทำงานจะเปลี่ยน โดยมีแอ็กเตอร์ เป็นผู้กระทำเริ่มต้น

ซีเควนต์ไดอะแกรมจะมีแกนสมมติ 2 แกนคือแกนนอน และแกนตั้ง แกนนอนจะแสดงขั้นตอนการทำงานหรือการส่งเมสเสจ (message) ระหว่างวัตถุ โดยแต่ละวัตถุจะส่งข้อมูลถึงกันว่าต้องทำอะไรเมื่อใด ส่วนแกนตั้งเป็นแกนเวลา แกนนอนและแกนตั้งต้องสัมพันธ์กัน

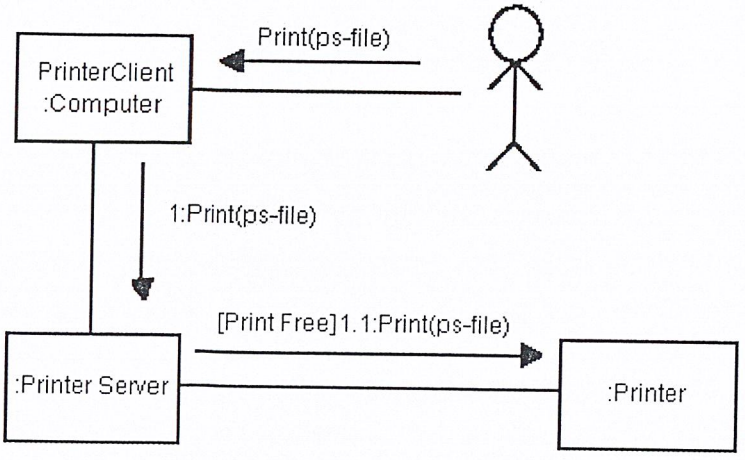


รูปที่ 4-9 แสดงซีเควนต์ไดอะแกรม

จากรูปเป็นการแสดงขั้นตอนการทำงานของการพิมพ์เอกสาร โดยเครื่องพิมพ์ เริ่มต้นคอมพิวเตอร์สั่งพิมพ์ไฟล์ ก็ส่งข้อมูลไปยังเซิร์ฟเวอร์ ต่อจากนั้นเซิร์ฟเวอร์จะดูว่าเครื่องพิมพ์ว่างหรือไม่ ถ้าว่างก็จัดการพิมพ์ ถ้าไม่ว่างก็รอคิวไว้ก่อน เมื่อเครื่องพิมพ์ทำการพิมพ์แล้วก็ส่งกลับไปจนถึงจุดเริ่มต้นคือคอมพิวเตอร์

4.5.3 คอลแลบอเรชั่นไดอะแกรม (Collaboration Diagram)

คอลแลบอเรชั่นไดอะแกรมจะมีหน้าที่เดียวกันกับซีเควนต์ไดอะแกรม แต่จะไม่แสดงถึงแกนเวลาอย่างชัดเจน ยกเว้นการตอบโต้กันระหว่างออบเจ็กต์



รูปที่ 4-10 คอลแลบอเรชั่นไดอะแกรม

จะแสดงถึงกลุ่มของออบเจ็กต์ที่ทำงานร่วมกันสอดคล้องกับความหมายของชื่อไดอะแกรม ลูกศรจะชี้ไปทางเดียวกัน ไม่มีการชี้ย้อนกลับในเส้นเดียวกัน

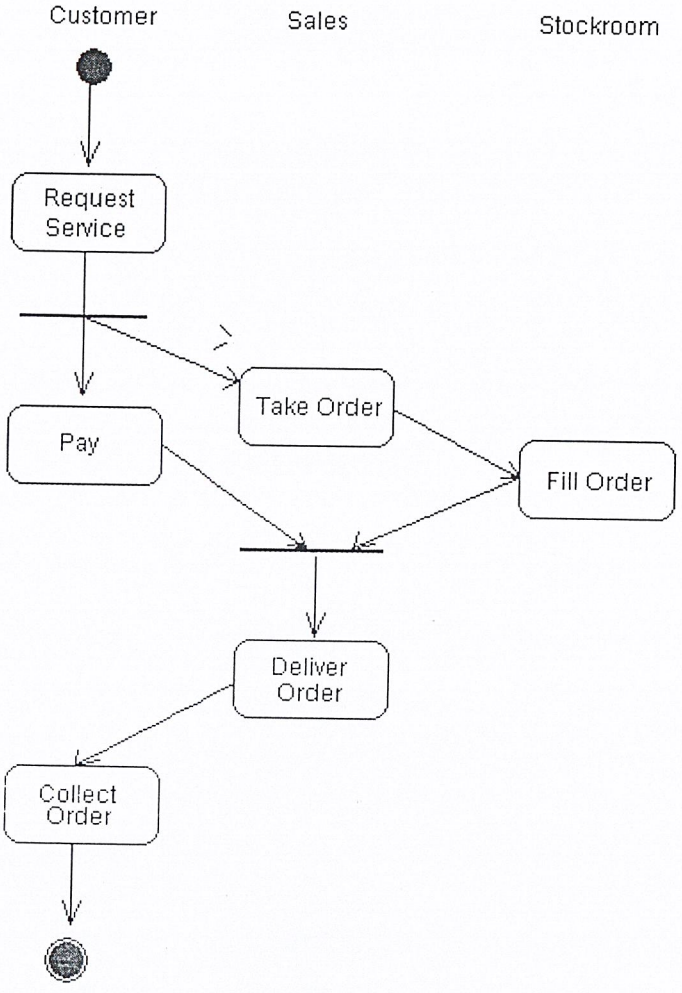
4.5.4 แอ็กทิวิตี้ไดอะแกรม (Activity Diagram)

แอ็กทิวิตี้ไดอะแกรมจะแสดงขั้นตอนการทำงานของ Use Case เช่นเดียวกับซีเควนต์ไดอะแกรมและคอลแลบอเรชั่นไดอะแกรม แต่จะเน้นไปที่งานย่อยของวัตถุ ซึ่งการเจาะจงไปที่งานๆหนึ่งของวัตถุเหมือนกับสเตตชาร์ตไดอะแกรมที่แสดงสถานะของวัตถุ แต่จริงๆแล้วแอ็กทิวิตี้ไดอะแกรมต่างกับสเตตชาร์ตไดอะแกรมตรงที่ แอ็กทิวิตี้ไดอะแกรมจะเปลี่ยนสถานะได้โดยไม่ต้องมีเหตุการณ์ที่กำหนดไว้ในไดอะแกรมมากระทำก่อน แต่มันจะเปลี่ยนสถานะเองตามกระบวนการทำงานคล้ายกับ Flowchart นั่นเอง

ตัวอย่างรูปด้านล่างเป็นการกำหนดว่างานในแต่ละเลนนั้นเกิดขึ้นกับออบเจ็กต์อะไร หรือกล่าวอีกนัยหนึ่งว่าเป็นการแสดงถึงกิจกรรมที่เกิดขึ้นกับออบเจกต์ที่เป็นเจ้าของเลนนั้น ๆ

ข้อดีของแอ็กทิวิตี้ไดอะแกรมคือ สามารถแสดงถึงการทำงานในวัตถุนั้นๆอย่างละเอียดคล้าย Flowchart และมีการแบ่งแยกหมวดหมู่งานตามออบเจ็กต์

แอ็กทิวิตี้ไดอะแกรมยังเหมาะกับการเขียนโมเดลในเชิงธุรกิจเพื่อให้ทราบกระแสการทำงาน (Workflow) ได้ สมารถช่วยแยกแยะผู้รับผิดชอบแต่ละงานได้ว่าใครควรจะเป็นคนทำงานในหมวดหมู่ใด



รูปที่ 4-11 แอ็กทิวิตี้ไดอะแกรม (Activity Diagram)

4.6 อิมพลีเมนเตชั่นไดอะแกรม (Implementation Diagram)

จะประกอบไปด้วยสัญลักษณ์ที่ใช้แสดงถึงโครงสร้างของซอร์สโค้ดหรือไฟล์ (ซอฟต์แวร์) และโครงสร้างของส่วนประกอบที่เชื่อมต่อกันในระบบ (ฮาร์ดแวร์)

กลุ่มอิมพลีเมนเตชั่นไดอะแกรม ประกอบไปด้วย 2 ไดอะแกรมคือ คอมโพเนนต์ไดอะแกรม (Component Diagram) และดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)

4.6.1 คอมโพเนนต์ไดอะแกรม (Component Diagram)

คอมโพเนนต์ไดอะแกรมจะแสดงความสัมพันธ์ที่เชื่อมต่อกันระหว่างซอฟต์แวร์คอมโพเนนต์ในระบบว่าประกอบไปด้วยไฟล์อะไรบ้าง ซึ่งอาจจะเป็น ไฟล์ซอร์ดโค้ด (Source Code) ไฟล์ไบนารีโค้ด (Binary Code) และไฟล์เอ็กซิคิวต์ (Executable Code)

การตั้งชื่อของคอมโพเนนต์ในคอมโพเนนต์ไดอะแกรมจะใช้ชื่อของคลาสจากคลาสไดอะแกรม ไม่ใช่ชื่อของอินสแทนซ์ (Instance) แต่ดีพลอยเมนต์ไดอะแกรมจะแสดงด้วยชื่อของอินสแทนซ์

4.6.2 ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)

ดีพลอยเมนต์ไดอะแกรมแสดงการเชื่อมต่อของอุปกาณ์ฮาร์ดแวร์ในระบบและมักใช้ร่วมกับคอมโพเนนต์ไดอะแกรม โดยข้างในฮาร์ดแวร์อาจประกอบไปด้วยซอฟต์แวร์คอมโพเนนต์ ดีพลอยเมนต์ไดอะแกรมจะแสดงอยู่ในรูปของอินสแทนซ์ และแสดงในช่วงเวลาของการรัน (Run -Time) หรือระหว่างการเอ็กซิคิวต์ ดังนั้นคอมโพเนนต์ของระบบที่ไม่ได้ใช้สำหรับรัน (เพราะถูกคอมไพล์ไปแล้ว เช่นไปฟล์ซอร์ดโค้ด) จะไม่ปรากฏในไดอะแกรมนี้ แต่จะมีในคอมโพเนนต์ของไฟล์ที่ใช้ทำงานจริง ๆ เท่านั้น

บทที่ 5
ระบบเครือข่ายแบบไคลเอ็นต์/เซิร์ฟเวอร์

ในการติดต่อสื่อสารบนเครือข่ายจะมีการติดต่อสื่อสารอยู่ 2 แบบ คือ แบบส่งข้อมูลและรับข้อมูล โดยเครื่องที่ทำการส่งข้อมูลเราเรียกว่าเครื่องให้บริการหรือเซิร์ฟเวอร์ (Server) และเครื่องที่รับข้อมูลเราเรียกว่าเครื่องรับบริการหรือไคลเอนต์ (Client) ซึ่งการติดต่อในลักษณะนี้เราเรียกว่าการติดต่อแบบไคลเอนต์-เซิร์ฟเวอร์(Client – Server)

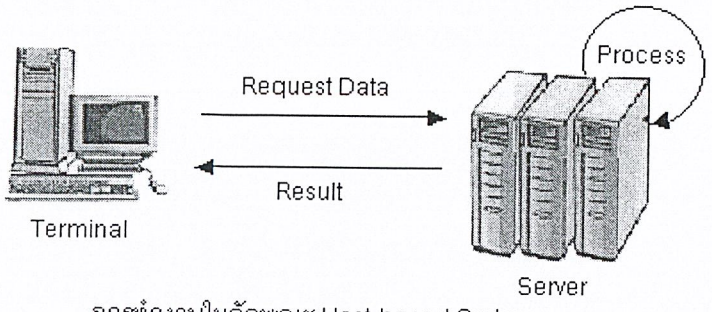
5.1 การติดต่อแบบไคลเอนต์/เซิร์ฟเวอร์ (Client/Server)

ในการใช้งานคอมพิวเตอร์ร่วมกันจะต้องมีการพึ่งพาอาศัยกันระหว่างเครื่องคอมพิวเตอร์หลายๆ เครื่องอยู่เสมอ เครื่องใดที่ให้บริการงานต่าง ๆ แก่เครื่องอื่นจะเรียกว่า เซิร์ฟเวอร์ (Server) ซึ่งในการให้บริการก็จะมีหลายแบบ เช่น ให้บริการรับส่งอีเมล์ จะเรียกว่า Mail Server ,ให้บริการเครื่องพิมพ์งาน จะเรียกว่า Print Server , ให้บริการเก็บไฟล์ข้อมูลต่าง ๆ จะเรียกว่า File Server เป็นต้น และสำหรับเครื่องที่ทำหน้าที่ให้บริการเว็บเพจ จะเรียกว่า Web Server ส่วนเครื่อง ที่เข้ามาขอใช้บริการ จากเซิร์ฟเวอร์จะเรียกว่าไคลเอนต์ (Client)

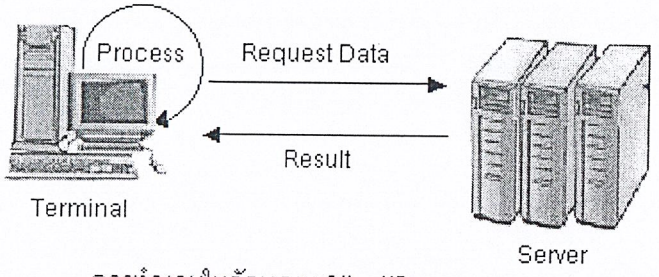
เครื่องคอมพิวเตอร์เครื่องหนึ่งไม่จำเป็นที่จะต้องเป็นเซิร์ฟเวอร์หรือไคลเอนต์อย่างใดอย่างหนึ่งเสมอไป แต่อาจสามารถเป็นอย่างใดอย่างหนึ่งก็ได้ เช่น เครื่องคอมพิวเตอร์เครื่องหนึ่งอาจจะใช้เป็น Print Server เพื่องานพิมพ์ แต่กลับเป็นไคลเอนต์เข้าใช้ไฟล์ต่าง ๆ จากเครื่อง File Server ก็ได้ หรือไม่เครื่องคอมพิวเตอร์เครื่องหนึ่ง สามารถที่จะเป็นเซิร์ฟเวอร์สำหรับหลายงานได้ และสามารถเป็นไคลเอนต์ของหลาย ๆ งานได้เช่นกัน เช่น เครื่องคอมพิวเตอร์เครื่องหนึ่งอาจเป็นทั้ง Print Server , File Server รวมทั้งยังเป็น Web Server ได้อีก นอกจากนี้เครื่องคอมพิวเตอร์ยังอาจเป็นได้ทั้งเซิร์ฟเวอร์และไคลเอนต์ของตัวมันเองในได้อีกด้วย คำว่า เซิร์ฟเวอร์ และไคลเอนต์ นั้นแท้จริงแล้วเป็นเพียงสถานภาพของเครื่องคอมพิวเตอร์เครื่องหนึ่งในเวลาหนึ่ง ๆ เท่านั้น

ในอดีต การพัฒนาระบบจะเป็นแบบ Host-based System คือ ในหน่วยงานจะมีเครื่องคอมพิวเตอร์ขนาดใหญ่ ซึ่งใช้เป็นทั้ง Application Server และ Database (File) Server และมีเครื่องคอมพิวเตอร์ขนาดเล็กเชื่อมต่อเข้ากับเครื่องเซิร์ฟเวอร์ทำหน้าที่เป็นเทอร์มินอล เครื่องเทอร์มินอลจะส่งข้อมูลไปประมวลผลที่เครื่องเซิร์ฟเวอร์ และส่งผลลัพธ์มาแสดงที่เครื่องเทอร์มินอล

ปัญหาที่เกิดขึ้น คือ เมื่อระบบมีขนาดใหญ่มากขึ้น มีปริมาณเทอร์มินอลที่เชื่อมต่อมากขึ้น เครื่องเซิร์ฟเวอร์จะต้องทำงานหนัก จนอาจมีผลต่อเวลาที่ใช้ในการตอบสนองผู้ใช้งานได้ เพื่อแก้ปัญหาดังกล่าวจึงได้เกิดแนวความคิดแบบไคลเอ็นต์/เซิร์ฟเวอร์ โดยเครื่องเซิร์ฟเวอร์จะทำหน้าที่เป็นเพียง Database (File) Server การประมวลผลทั้งหมดจะอยู่ที่เครื่องเทอร์มินอล (Client)



การทำงานในลักษณะ Host-based System



การทำงานในลักษณะ Client/Server

รูปที่ 5-1 การทำงานในลักษณะ Host-based System และ Client/Server

5.2 รูปแบบของไคลเอ็นต์/เซิร์ฟเวอร์ที่ใช้งานมีอยู่ 4 ชนิด

5.2.1 Stand alone Client/Server

การทำงานแบบนี้ผู้ให้บริการหรือเซิร์ฟเวอร์จะอยู่บนเครื่องเดียวกับผู้ขอใช้บริการหรือไคลเอ็นต์ ทำให้มีความเร็วในการติดต่อสื่อสารระหว่างผู้ให้บริการและผู้ขอใช้บริการสูงมาก แต่ประสิทธิภาพในการประมวลผลระบบฐานข้อมูลจะลดลงบ้าง ระบบนี้เรียกอีกอย่างว่า Tiny Client/Server

5.2.2 Department Client/Server หรือ LAN based single server

การทำงานแบบนี้จะมีผู้ให้บริการเกี่ยวกับฐานข้อมูล แอพพลิเคชัน ฯลฯ อยู่บนเครื่องเซิร์ฟเวอร์ และผู้ขอใช้บริการทั้งหลายจะอยู่บนเครื่องไคลเอ็นต์ โดยจะเชื่อมต่อกันด้วยระบบเครือข่ายท้องถิ่น (LAN) และมิดเดิลแวร์ (Middleware) เป็นตัวกลางที่ทำงานอยู่ระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ การติดต่อระหว่างผู้ให้บริการและผู้ขอใช้บริการจะช้ากว่าแบบ Stand alone เพราะจะต้องติดต่อผ่านระบบเครือข่าย ยิ่งถ้ามีผู้ขอใช้บริการเข้ามาดึงข้อมูลกันครั้งละมากๆ หลายเครื่อง ประสิทธิภาพจะลดลงอย่างเห็นได้ชัด ส่วนวิธีเพิ่มประสิทธิภาพ คือ การเพิ่มเครื่องเซิร์ฟเวอร์ขึ้นในระบบ

5.2.3 Workgroups Client/Server

การทำงานแบบเวิร์กกรุ๊ปนี้จะเป็นกลุ่มของเซิร์ฟเวอร์ที่หลากหลายแพลตฟอร์ม หลายผู้ผลิต มีความแตกต่างกันของเซิร์ฟเวอร์ แต่ทั้งหมดนี้จะเชื่อมต่อกันทางระบบเครือข่าย LAN หรือ WAN และใช้มิดเดิลแวร์มาตรฐานในการทำงาน

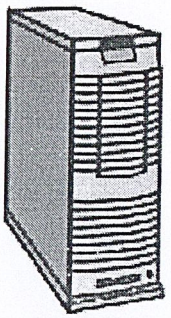
5.2.4 Enterprise Client/Server

การทำงานแบบเอ็นเตอร์ไพรท์จะทำให้มีการเชื่อมโยงเครื่องเซิร์ฟเวอร์หรือโฮสต์ที่มีแพลตฟอร์มต่างกันเข้าด้วยกัน ทำให้มีการใช้ทรัพยากรบนระบบได้อย่างมีประสิทธิภาพสูงสุด โดยที่ไคลเอ็นต์จะเลือกใช้ทรัพยากรฐานข้อมูลจากเซิร์ฟเวอร์เครื่องใดก็ได้ผ่านทางมิดเดิลแวร์

5.3 การจัดแบ่งประเภทของไคลเอ็นต์/เซิร์ฟเวอร์จะแบ่งตามลำดับชั้น (Tier)

5.3.1 One Tier

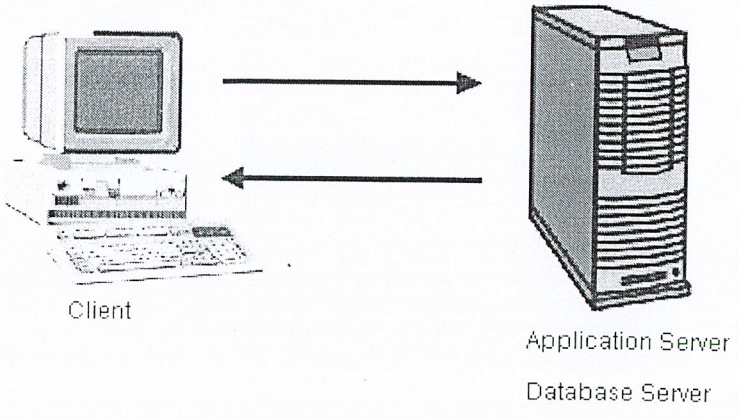
การรวมการทำงานทุกอย่างเบ็ดเสร็จภายในเครื่องเดียว เป็นทั้งไคลเอ็นต์และเซิร์ฟเวอร์ มีลักษณะการทำงานคล้ายกับคอมพิวเตอร์แบบ Stand Alone



รูปที่ 5-2 การจัดแบ่งไคลเอ็นท์/เซิร์ฟเวอร์แบบ 1 tier

5.3.2 Two Tiers

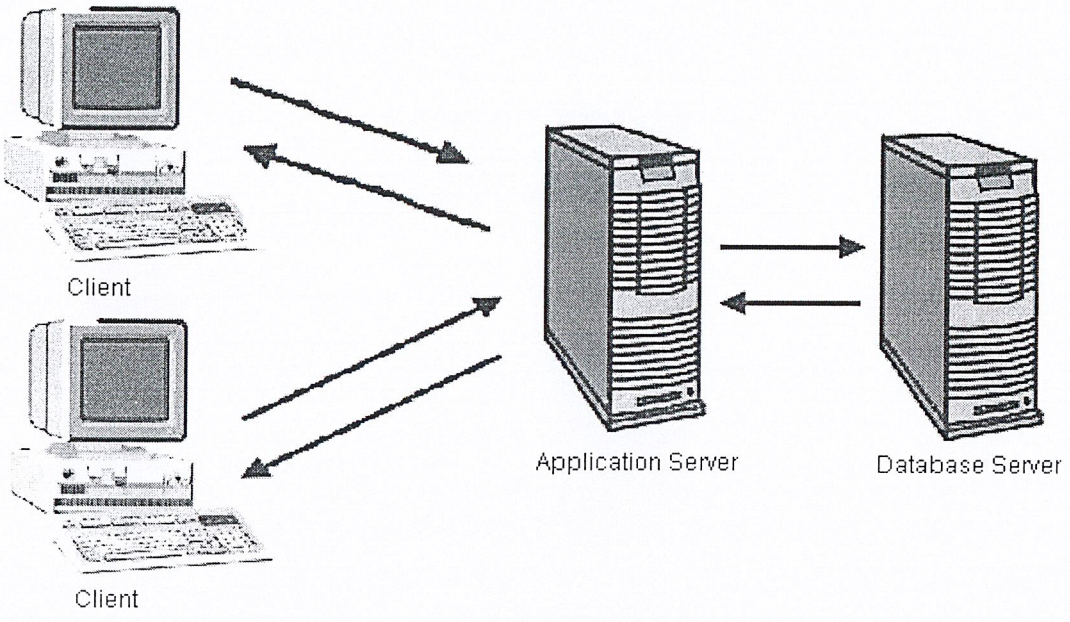
การแบ่งการทำงานเป็นไคลเอ็นต์และเซิร์ฟเวอร์โดยที่ฝั่งไคลเอ็นต์จะมีโปรแกรมเกี่ยวกับยูสเซอร์อินเทอร์เฟซที่สามารถประมวลผลเบื้องต้นได้ ก่อนที่จะสั่งคำสั่งไปขอข้อมูลจากฝั่งเซิร์ฟเวอร์ ซึ่งจะเป็นการลดภาระงานสำหรับเซิร์ฟเวอร์



รูปที่ 5-3 การจัดแบ่งไคลเอ็นท์/เซิร์ฟเวอร์แบบ 2 tier

5.3.3 Three Tiers

การแบ่งการทำงานเป็นไคลเอ็นต์และเซิร์ฟเวอร์ คล้ายกับแบบ 2 Tier แต่ว่าจะกระจายการทำงานของเซิร์ฟเวอร์ออกเป็นอีก 2 Tier โดยที่ Tier หนึ่งทำงานเป็นดาต้าเบสเซิร์ฟเวอร์ และอีก Tier หนึ่งทำงานเป็นแอพพลิเคชันเซิร์ฟเวอร์



รูปที่ 5-4 การจัดแบ่งไคลเอ็นท์/เซิร์ฟเวอร์แบบ 3 tier

5.4 การจัดแบ่งประเภทของเซิร์ฟเวอร์จะแบ่งตามประเภทการใช้งาน

5.4.1 File Server

เซิร์ฟเวอร์ทำหน้าที่จัดเก็บไฟล์ที่สร้างขึ้น เพื่อให้ไคลเอ็นต์อื่นๆ สามารถเข้ามาใช้ไฟล์ร่วมกันได้ โดยอาจจะกำหนดสิทธิ์ในการใช้ตามกลุ่มของผู้ใช้ เช่น พนักงานในแผนกการเงินสามารถเรียกใช้ข้อมูลได้เฉพาะภายในแผนกเท่านั้น ไม่สามารถเรียกข้อมูลที่สร้างโดยแผนกบุคคลได้ ในขณะที่ผู้บริหารสามารถเรียกใช้ข้อมูลของทุกแผนกได้

5.4.2 Database Server

เซิร์ฟเวอร์ที่ทำหน้าที่เป็น DBMS คอยจัดการกับข้อมูลที่จัดเก็บอยู่ในเซิร์ฟเวอร์เดียวกันนี้ ทำให้ผู้ใช้ที่เป็นไคลเอ็นต์สามารถเข้ามาแชร์ใช้ข้อมูลร่วมกัน ทำให้ข้อมูลที่ได้มีความทันสมัย ไม่ซ้ำซ้อน

5.4.3 Application Server

เซิร์ฟเวอร์ที่ทำหน้าที่จัดการกับโปรแกรมแอพพลิเคชันในการเชื่อมต่อระหว่าง Database Server กับผู้ใช้ (User) เซิร์ฟเวอร์กับยูสเซอร์ที่เป็นไคลเอ็นต์เพื่อให้ทั้งสองส่วนนี้สามารถเชื่อมต่อกันได้อย่างสมบูรณ์

5.5 เว็บเซิร์ฟเวอร์ (Web Server) และเว็บบราวเซอร์ (Web Browser)

การทำงานในระบบไคลเอนต์/เซิร์ฟเวอร์ จะมีโปรแกรมที่ใช้ในการทำงานร่วมกันนั่นคือ โปรแกรมที่ทำงานฝั่งเซิร์ฟเวอร์ และโปรแกรมที่ทำงานในฝั่งไคลเอนต์ ซึ่งในการใช้งานอินเตอร์เน็ต WWW (World Wide Web) นั้น โปรแกรมที่ทำงานฝั่งเซิร์ฟเวอร์ก็คือ ซอฟต์แวร์ประเภทเว็บเซิร์ฟเวอร์ (Web server) ส่วนโปรแกรมที่ทำงานฝั่งไคลเอนต์ของ WWW ก็คือโปรแกรมประเภทเว็บบราวเซอร์ (Web browser) นั่นเอง

5.5.1 เว็บเซิร์ฟเวอร์ (Web server)

เป็นโปรแกรมที่ทำงานฝั่งเซิร์ฟเวอร์ ทำหน้าที่หลักคือ แปลเอกสาร JSP หรือสคริปต์ที่ต้องทำการแปลผลฝั่งเซิร์ฟเวอร์ประเภทอื่น ๆ เช่น PHP เป็นต้น ให้เป็นเอกสารในรูปแบบ HTML เพื่อส่งไปแสดงผลในโปรแกรมเว็บบราวเซอร์ที่ฝั่งไคลเอนต์ โปรแกรมประเภทเว็บเซิร์ฟเวอร์ มีอยู่หลายค่าย เช่น Apache Web Server , Microsoft Internet Information Services (IIS) , Microsoft Personal Web Server , Netscape Enterprise Server เป็นต้น

5.5.2 เว็บบราวเซอร์ (Web Browser)

เป็นโปรแกรมที่ทำหน้าที่หลัก ๆ คือ นำเอกสารที่อยู่ในรูปแบบ HTML มาแสดงผลเป็นเว็บเพจให้ผู้ใช้ดู ซึ่งโปรแกรมประเภทเว็บบราวเซอร์มีหลายค่าย เช่น Netscape Navigator / Communicator , Microsoft Internet Explorer , Opera เป็นต้น

5.6 การทำงานของเว็บเซิร์ฟเวอร์และเว็บบราวเซอร์

การติดต่อระหว่างเซิร์ฟเวอร์และเว็บบราวเซอร์ จะเริ่มจากเว็บบราวเซอร์ส่งการเชื่อมต่อ และร้องขอข้อมูล (เว็บเพจ) ไปยังเว็บเซิร์ฟเวอร์ หากข้อมูลที่ร้องขอเป็นเท็กซ์ ,รูปภาพ หรือเสียง ธรรมดา เว็บเซิร์ฟเวอร์ก็จะส่งข้อมูลที่เว็บบราวเซอร์เรียกร้องไปให้โดยตรง เมื่อส่งข้อมูลไปให้เรียบร้อยแล้วก็จะตัดขาดการติดต่อจากกัน แต่ถ้ากรณีที่การร้องขอเป็นพวกเอกสาร JSP แล้ว เว็บเซิร์ฟเวอร์จะทำการแปลคำสั่งในเอกสาร JSP ก่อน จากนั้นจึงค่อยส่งผลรับที่ได้ไปให้เว็บบราวเซอร์แปบผลแสดงเป็นเว็บเพจให้ผู้ใช้ดูอีกที จากนั้นก็จะตัดขาดการติดต่อ

5.7 HTTP (Hypertext Transfer Protocol)

เป็นโปรโตคอลที่ใช้ในการควบคุมการรับส่งเอกสารใน WWW (World Wide Web) เปรียบเทียบได้ว่าเป็นภาษาหรือมาตรฐานในการสื่อสารระหว่างเว็บบราวเซอร์และเว็บเซิร์ฟเวอร์ ซึ่งการทำงานของโปรโตคอล HTTP นี้มีลักษณะเป็น Stateless คือเมื่อเสร็จสิ้นการติดต่อกันระหว่าง เว็บเซิร์ฟเวอร์และเว็บบราวเซอร์แล้ว จะไม่มีข้อมูลใดคงเหลือทิ้งไว้ที่เครื่องเซิร์ฟเวอร์หลังจากการติดต่อเสร็จแล้ว

5.8 HTML (Hypertext Markup Language)

HTML เป็นรูปแบบเอกสารที่ใช้งานกับโปรโตคอล HTTP ซึ่งเว็บบราวเซอร์จะรับข้อมูลที่อยู่ในรูปแบบเอกสาร HTML นี้ แล้วนำมาแสดงผลออกมาเป็นเว็บเพจ สำหรับไฟล์ที่เป็นเอกสาร JSP นั้น ความจริงแล้วยังไม่ได้อยู่ในรูปแบบของเอกสาร HTML จึงไม่สามารถแสดงผลผ่านทางโปรแกรมเว็บบราวเซอร์ได้โดยตรง

ดังนั้นก่อนที่จะแสดงผลเอกสาร JSP ผ่านโปรแกรมเว็บบราวเซอร์ได้นั้น เอกสาร JSP จะต้องถูกทำการแปลงให้อยู่ในรูปแบบ HTML เสียก่อน (ถึงแม้เวลาเรียกดูในเว็บบราวเซอร์จะแสดงไฟล์ .jsp อยู่ก็ตาม) โดยโปรแกรมเว็บเซิร์ฟเวอร์ จากนั้นจึงส่งข้อมูลในรูปแบบเอกสาร HTML ที่ได้ ไปให้โปรแกรมเว็บบราวเซอร์แสดงผลอีกทีภายในเอกสาร HTML จะเป็นข้อความที่ใช้รหัส ASCII ซึ่งรหัส ASCII นี้จะเป็นตัวอักษรสากลที่เครื่องคอมพิวเตอร์ทุกระบบปฏิบัติการสามารถอ่านได้ ข้อมูลที่ปรากฏภายในเอกสาร HTML สามารถแบ่งออกเป็น 2 ส่วนตามหน้าที่ได้ดังนี้

- ส่วนที่เป็นเนื้อหา ที่ผู้เขียนเว็บเพจต้องการให้ปรากฏบนหน้าจอ
- ข้อความที่ใช้กำหนดลักษณะการแสดงผลของส่วนที่เป็นเนื้อหา

ซึ่งเรียกว่าแท็ก (TAG) ส่วนนี้จะอยู่ในเครื่องหมาย < > โดยส่วนที่เรียกว่าแท็กในเอกสาร HTML นี้ จะไม่ปรากฏให้ผู้ชมเว็บเพจเห็น

5.9 Client Side Script และ Server Side Script

การทำงานของแอลพลิเคชั่นบนอินเตอร์เน็ตนั้น มีลักษณะการทำงานแบบไคลเอนต์และเซิร์ฟเวอร์ เพื่อให้เป็นการแบ่งเบาภาระซึ่งกันและกัน เราสามารถที่จะเขียนสคริปต์และกำหนดให้ทำงานที่ฝั่งไคลเอนต์หรือที่ฝั่งเซิร์ฟเวอร์ได้ โดยสคริปต์ที่ทำงานที่ฝั่งไคลเอนต์เราจะเรียกว่า Server Side Script ส่วนสคริปต์ที่ทำงานที่ฝั่งเซิร์ฟเวอร์เราจะเรียกว่า Server Side Script

การออกแบบแอพพลิเคชั่นบนอินเตอร์เน็ตนั้น สิ่งที่ต้องคำนึงถึง ซึ่งแตกต่างจากการเขียนแอพพลิเคชั่นที่ทำงานบนเครื่องคอมพิวเตอร์เพียงเครื่องเดียว ก็คือ ความล่าช้าในการติดต่อส่งข้อมูลระหว่างไคลเอนต์แลเซิร์ฟเวอร์ ดังนั้นการสร้างแอพพลิเคชั่นต้องมีการแบ่งงานกันให้เหมาะสมระหว่างงานที่ต้องทำที่ฝั่งไคลเอนต์และงานที่ต้องทำฝั่งเซิร์ฟเวอร์ เพื่อลดการสูญเสียเวลารอในช่วงเวลาที่ส่งข้อมูลถึงกัน และลดความหนาแน่นของการจราจรในระบบเครือข่าย เช่นหากมีการให้กรอกแบบฟอร์มเพื่อส่งข้อมูลมาประมวลผลที่เครื่องเซิร์ฟเวอร์ การที่ผู้ใช้กรอกข้อมูลที่ผิดมา ข้อมูลที่ได้ย่อมไม่สามารถนำไปประมวลผลได้ เราสามารถแบ่งเบาภาระให้กับเซิร์ฟเวอร์ได้โดยเขียนสคริปต์ที่ทำงานฝั่งไคลเอนต์เพื่อตรวจสอบข้อมูลก่อนที่จะส่งมาให้เซิร์ฟเวอร์ เป็นต้น

5.9.1 Client Side Script

Client Side Script คือสคริปต์ที่ทำงานที่ฝั่งไคลเอนต์ ถูกประมวลผลด้วยโปรแกรมเว็บบราวเซอร์ของผู้ใช้ และแสดงผลเป็นเว็บเพจออกมาให้ผู้ใช้ดู มีจุดเด่นตรงที่สามารถโต้ตอบกับผู้ใช้ได้อย่างรวดเร็ว การใช้ Client Side Script อย่างเหมาะสม จะเป็นการแบ่งเบาภาระงานของเซิร์ฟเวอร์ และช่วยลดความหนาแน่นการเจรจาข้อมูลในระบบเครือข่าย และเนื่องจากสคริปต์ชนิดนี้ทำงานที่ฝั่งผู้ใช้เอง

ดังนั้นจึงไม่มีข้อจำกัดในการเลือกใช้เซิร์ฟเวอร์ โดยเซิร์ฟเวอร์จะเป็นระบบปฏิบัติการใดก็ได้ แต่มีข้อเสียที่ต้องคำนึงว่าทางบราวเซอร์ของผู้ใช้สนับสนุนและสามารถใช้งานสคริปต์ที่เราเขียนได้หรือไม่

5.9.2 Server Side Script

Server Side Script คือ สคริปต์ที่ทำงานที่ฝั่งเซิร์ฟเวอร์ ถูกประมวลผลโดยโปรแกรมเว็บเซิร์ฟเวอร์ เพื่อเปลี่ยนแปลงเป็นเอกสารในรูปแบบ HTML แล้วส่งผลลัพธ์ที่ได้ไปให้เว็บบราวเซอร์ที่ฝั่งไคลเอนต์ สคริปต์ที่ทำงานฝั่งเซิร์ฟเวอร์นี้ มีจุดเด่นตรงที่ไม่ต้องคำนึงว่าทางฝั่งผู้ใช้งานบราวเซอร์ชนิดใด เพราะการประมวลผลเกิดขึ้นที่เซิร์ฟเวอร์เอง

แต่มีข้อเสียก็คือ หากมีการส่งข้อมูลมาให้เซิร์ฟเวอร์ประมวลผลมาก ๆ จะเป็นภาระให้เซิร์ฟเวอร์ในการประมวลผล และเป็นการสร้างความหนาแน่นให้เส้นทางจราจรในเครือข่ายอีกด้วย การเขียนสคริปต์ที่ทำงานฝั่งเซิร์ฟเวอร์นี้ต้องคำนึงถึงโปรแกรมเว็บเซิร์ฟเวอร์ด้วยว่า สามารถรองรับการทำงานของเทคโนโลยีการจัดการแอพพลิเคชั่นบนอินเตอร์เน็ตแบบใด และสามารถเขียนใช้งานอ็อบเจ็กต์ต่าง ๆ ในเทคโนโลยีนั้นด้วยภาษาอะไร

บทที่ 6
ทฤษฎีและความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล

6.1 ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Object-Relational Database Management System : ORDBMS) เป็นระบบจัดการฐานข้อมูลที่มีความสามารถของระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS)

6.1.1 ความสามารถของระบบจัดการฐานข้อมูลเชิงสัมพันธ์

ความสามารถในการเก็บและจัดการข้อมูลที่อยู่ในลักษณะความสัมพันธ์แบบรีเลชันซึ่งตัวอย่างหนึ่งของความสัมพันธ์แบบรีเลชันนี้ คือ ตาราง โดยมีความสัมพันธ์กันระหว่างแถว (Row) และสดมภ์ (Column) ตัวอย่างของระบบจัดการฐานข้อมูลแบบนี้คือ ระบบจัดการฐานข้อมูลที่ใช้ภาษา SQL เป็นหลัก (SQL – Base Database Management System) ซึ่งมีคุณสมบัติดังนี้

1. โครงสร้างข้อมูล (Data Structure หรือ Logical data structure) เป็นในรูปแบบของความสัมพันธ์โดยแสดงออกมาในรูปแบบของตารางซึ่งข้อมูลในตารางจะต้องมีคุณสมบัติดังนี้
   - ข้อมูลในแต่ละแถวไม่ซ้ำกัน
   - ไม่จำเป็นต้องมีการเรียงลำดับของแถวต่างๆ
   - ข้อมูลในแต่ละสดมภ์จะไม่สามารถแบ่งแยกอีกได้
2. กฎบังคับความถูกต้องของข้อมูล (Data Integrity) เป็นกฎบังคับว่าข้อมูลที่ทำการใส่ลงฐานข้อมูลนั้นต้องมีความถูกต้องตามกฎข้อบังคับ (Constraint) ที่ตั้งไว้
3. ภาษาที่ใช้ในการจัดการข้อมูล (Data Manipulation)

6.1.2 ความสามารถของระบบจัดการฐานข้อมูลแบบออบเจ็กต์

มีการเก็บข้อมูล (Data หรือ Property) และการใช้ข้อมูล (Procedure หรือ Method) เข้าด้วยกันอยู่ในลักษณะของออบเจ็กต์ ซึ่งมีคุณสมบัติทางออบเจ็กต์ดังนี้

1. ความสามารถในการซ่อนรายละเอียด (Encapsulation) เป็นการซ่อนการทำงานภายในของออบเจ็กต์ไว้ ซึ่งออบเจ็กต์อื่นจะรู้แค่อินเทอร์เฟซ (Interface) เท่านั้น
2. คุณสมบัติการโพลีมอร์ฟีซึม (Polymorphism)
3. ความสัมพันธ์แบบออบเจ็กต์ คือ
   - ความสัมพันธ์แบบแอกกริเกชั่น (Aggregation)
   - ความสัมพันธ์แบบสืบทอด (Generalization)

ในระบบจัดการฐานข้อมูลเชิงวัตถุเชิงสัมพันธ์ที่ทำการศึกษาและนำมาใช้งานคือ ระบบจัดการฐานข้อมูลออราเคิล ซึ่งมีความสามารถดังนี้

1. สามารถใช้ชนิดข้อมูลมาตรฐาน (Built – In Datatypes)
2. สามารถใช้ชนิดข้อมูลแบบกำหนดเอง (Users-Defined Datatypes) ซึ่งประกอบด้วย
    - ชนิดข้อมูลแบบออบเจ็กต์ (Object Types)
    - ชนิดข้อมูลแบบกลุ่ม (Collection Types)

6.2 ประเภทของข้อมูลในฐานข้อมูล

ระบบฐานข้อมูลออราเคิลสามารถเก็บข้อมูลลงในตารางข้อมูลได้หลากหลายประเภทโดยจัดเก็บไว้ในคอลัมภ์ของตารางข้อมูล ซึ่งเราจะต้องกำหนดประเภทของข้อมูลในแต่ละคอลัมภ์ตามลักษณะของข้อมูลที่ต้องการจัดเก็บ

ประเภทข้อมูลที่สามารถจัดเก็บในฐานข้อมูลออราเคิลได้ประกอบไปด้วย

- CHAR(size) เป็นข้อมูลประเภทตัวอักขระที่มีความยาวของข้อมูลคลที่ตามค่า size ที่กำหนด ค่าสูงสุดของข้อมูลที่สามารถจัดเก็บได้คือ 2000 ไบต์ และเมื่อทำการเพิ่มหรือเปลี่ยนแปลงข้อมูลที่มีความยาวน้อยกว่า size ที่กำหนด ข้อมูลชุดนั้นจะถูกแทนที่ด้วยช่องว่างโดยอัตโนมัติ
- VARCHAR2(size) เป็นข้อมูลประเภทอักขระที่มีความยาวผันเปลี่ยนไปตามความยาวของข้อมูล โดยค่าสูงสุดของข้อมูลที่สามารถจัดเก็บได้คือ 4000 ไบต์ ในกรณีที่ความยาวไม่ถึง size ที่กำหนด ข้อมูลจะถูกจัดเก็บตามความยาวจริงของข้อมูลเท่านั้น
- NCHAR(size) เป็นข้อมูลประเภทอักขระเช่นเดียวกับ CHAR แต่จะเก็บข้อมูลตามค่าอักขระที่กำหนดใน National Character Set
- NVARCHAR2(size) เป็นข้อมูลประเภทเดียวกับ VARCHAR2 แต่จะเก็บข้อมูลตามค่าอักขระที่กำหนดใน National Character Set
- LONG เป็นข้อมูลประเภทอักขระที่สามารถเก็บค่าข้อมูลได้สูงสุดถึง 2 GB จึงเหมาะสำหรับข้อมูลที่มีขนาดใหญ่
- NUMBER(n,m) เป็นข้อมูลประเภทตัวเลขที่กำหนดจำนวนหลักของตัวเลขที่จัดเก็บ ดังนี้ จำนวนหลักหน้าจุดทศนิยมเก็บได้ไม่เกิน n-m หลัก และจำนวนทศนิยมสามารถเก็บได้สูงสุด m หลัก
- Date เป็นข้อมูลประเภทวันและเวลา ซึ่งมีขนาดคงที่ 7 ไบต์
- RAW(size) เป็นข้อมูลประเภทไบนารี่ เช่น ข้อมูลรูปภาพ สามารถจัดเก็บได้โดยมีขนาดไม่เกิน size ที่กำหนด จำนวนสูงสุดที่จัดเก็บได้คือ 2000 ไบต์
- LONG RAW(size) เป็นข้อมูลประเภทไบนารี่เช่นเดียวกับ RAW แต่ต่างกันที่สามารถเก็บข้อมูลได้สูงสุดถึง 2 GB

- BLOB เป็นข้อมูลประเภทไบนารี่ที่มีขนาดใหญ่สามารถเก็บข้อมูลได้สูงสุด 4 GB
- CLOB เป็นข้อมูลประเภทอักขระที่มีขนาดใหญ่สามารถเก็บข้อมูลได้สูงสุด 4 GB เช่นกัน
- NCLOB เป็นข้อมูลประเภทอักขระเช่นเดียวกับ CLOB แต่เป็นการจัดเก็บข้อมูลตามค่าอักขระที่กำหนดใน National Character Set โดยสามารถเก็บข้อมูลได้สูงสุด 4 GB
- BFILE เป็นข้อมูลประเภทไบนารี่ไฟล์ โดยเป็นไฟล์ที่อยู่นอกฐานข้อมูลและสามารถอ้างอิงได้จากพอยน์เตอร์ในฐานข้อมูล ข้อมูลประเภทนี้สามารถเก็บข้อมูลได้สูงสุด 4 GB

6.3 ชนิดข้อมูลแบบกำหนดเอง

เกิดจากการสร้างชนิดข้อมูลขึ้นมาจากชนิดข้อมูลมาตรฐาน และชนิดข้อมูลกำหนดเองอื่นๆ ซึ่งได้ถูกแบ่งออกมาสองลักษณะดังนี้

6.3.1 ชนิดข้อมูลแบบออบเจ็กต์

เป็นชนิดของข้อมูลที่ผู้ใช้สามารถทำการสร้างขึ้นมาเอง โดยทำการกำหนดลักษณะของข้อมูลให้มีความใกล้เคียงกับความเป็นจริงมากที่สุด ซึ่งเป็นการใช้ลักษณะเด่นของข้อมูลนั้นเป็นตัวระบุ ซึ่งชนิดข้อมูลแบบออบเจ็กต์นี้ประกอบด้วยส่วนประกอบ 3 ส่วนดังนี้

1. ชื่อ (Name) ใช้ในการระบุถึงชนิดข้อมูลแบบออบเจ็กต์ ซึ่งต้องไม่ซ้ำกันในโครงสร้างที่เก็บข้อมูลแบบออบเจ็กต์เดียวกัน (Schema)
2. ลักษณะต่างๆ (Attributes) คือข้อมูลที่ออบเจ็กต์ควรมีเพื่อทำให้ออบเจ็กต์นั้นมีความใกล้เคียงกับความเป็นจริงให้มากที่สุด ซึ่งจะเป็นชนิดข้อมูลแบบมาตรฐาน หรือชนิดข้อมูลแบบกำหนดเองก็ได้
3. การกระทำ (Method) เป็นฟังก์ชันหรือโพรซีเจอร์ ที่สร้างจากภาษา PL/SQL เพื่อสร้างการกระทำให้กับออบเจ็กต์

6.3.2 ชนิดข้อมูลแบบกลุ่ม

เป็นลักษณะของข้อมูลที่มีชนิดของข้อมูลเหมือนกันและมีจำนวนที่ไม่สามารถระบุได้ว่ามีเท่าไร ซึ่งจะแบ่งเป็นชนิดข้อมูลแบบอาร์เรย์ (Array Types) หรือชนิดข้อมูลแบบ วีอาร์เรย์ (VArray) และชนิดข้อมูลแบบ ตาราง (Table Types) หรือชนิดข้อมูลแบบตารางซ้อนตาราง (Nested Table)

- ชนิดข้อมูลแบบวีอาร์เรย์

เป็นการเก็บข้อมูลที่มีชนิดของข้อมูลเหมือนกัน และข้อมูลที่อยู่ในแต่ละช่องของอาร์เรย์จะถูกระบุด้วยดัชนี (Index) ซึ่งเป็นลักษณะของเซตของข้อมูลแบบมีลำดับ

6.4 ฐานข้อมูล (Database) พื้นที่ตาราง (Tablespaces) และไฟล์ข้อมูล (Datafiles)

ข้อมูลของออราเคิลที่ทำการเก็บลงในฐานข้อมูลนั้นสามารถพิจารณาใน 2 มุมมอง คือ ในทางด้านความคิดหรือทางลอจิกนั้นจะเก็บข้อมูลลงในพื้นที่ตาราง (Tablespaces) และในมุมมองทางด้านกายภาพนั้นจะทำการเก็บข้อมูลลงในไฟล์ข้อมูลที่กำหนด

แม้ว่าฐานข้อมูล พื้นที่ตารางและไฟล์ข้อมูลจะมีความสัมพันธ์กันแต่ก็มีความแตกต่างกันอยู่บ้าง คือ ในฐานข้อมูลของออราเคิลจะมีหน่วยทางตรรกที่จัดเก็บข้อมูลทั้งหมดคือพื้นที่ตารางและในพื้นที่ตารางนั้นจะมีไฟล์ที่ทำการเก็บข้อมูลในทางกายภาพเรียกว่าไฟล์ข้อมูล

เมื่อเราสามารถสร้างพื้นที่ตารางแล้วเราสามารถสร้างตารางขึ้นมาใช้งานได้ โดยไม่เกี่ยวข้องกับพื้นที่ตารางระบบของทางออราเคิลที่ได้สร้างขึ้นมาเมื่อเราได้ติดตั้งระบบ

6.5 Java Database Connectivity (JDBC)

JDBC หรือที่เราเรียกกันอีกอย่างหนึ่งว่า Java Database Connectivity เป็น API ที่ช่วยอำนวยความสะดวกในการติดต่อฐานข้อมูล หรือจะกล่าวง่ายๆว่า JDBC เป็นกลุ่มของคลาสที่ใช้ในการติดต่อฐานข้อมูลนั่นเอง ดังนั้นเมื่อต้องการเขียนสคริปต์ JSP ติดต่อกับฐานข้อมูล ก็จะต้องเรียกไดรเวอร์ของ JDBC(JDBC Driver) ในการติดต่อ ซึ่ง JDBC Driver จะอยู่ในรูปของคลาส

โปรแกรมระบบฐานข้อมูลแต่ละโปรแกรม ต่างมี JDBC Driver เฉพาะของตน ถ้าเราต้องการเขียนสคริปต์ JSP เรียกใช้ JDBC Driver ของโปรแกรมฐานข้อมูลใด ก็ต้องติดตั้ง JDBC Driver ของฐานข้อมูลระบบนั้นเอาไว้ก่อน

การติดต่อกับฐานข้อมูลโดยใช้ JDBC สามารถแบ่งออกได้เป็น 4 ขั้นตอน ดังนี้

1 โหลดคลาสไดรเวอร์ของ JDBC
2 เปิดเชื่อมต่อไปยังฐานข้อมูล
3 ติดต่อกับฐานข้อมูลโดยใช้คำสั่ง SQL
4 จัดการกับผลลัพธ์ที่ได้จากคำสั่ง SQL

6.5.1 รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC

JDBC API สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบ Two-Tiers model และ Three-Tiers model

6.5.1.1 Two Tiers model

จาวาแอพเพล็ต (Java applet) หรือจาวาแอพพลิเคชัน (Application) จะติดต่อกับฐานข้อมูลโดยตรง จึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC ไดรเวอร์พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ คำสั่งในการค้นคืนข้อมูลในรูปของภาษา SQL (Structured Query Language) จะถูกส่งจากผู้ใช้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบจัดการฐานข้อมูล ก็จะถูกส่งกลับมาสู่ผู้ใช้ฐานข้อมูลดังกล่าวนี้ส่วนมากจะติดตั้งอยู่ต่างเครื่องคอมพิวเตอร์ โดยผู้ใช้สามารถเชื่อมต่อผ่านระบบเน็ตเวิร์ก (Network) รูปแบบ Two – Tiers นี้ใช้หลักการทำงานเช่นเดียวกันกับรูปแบบไคลเอ็นต์/เซิร์ฟเวอร์ที่เรารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้คือไคลเอ็นต์ และเครื่องคอมพิวเตอร์ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครือข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ Two – Tiers มักเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร

6.5.1.2 Three Tiers Model

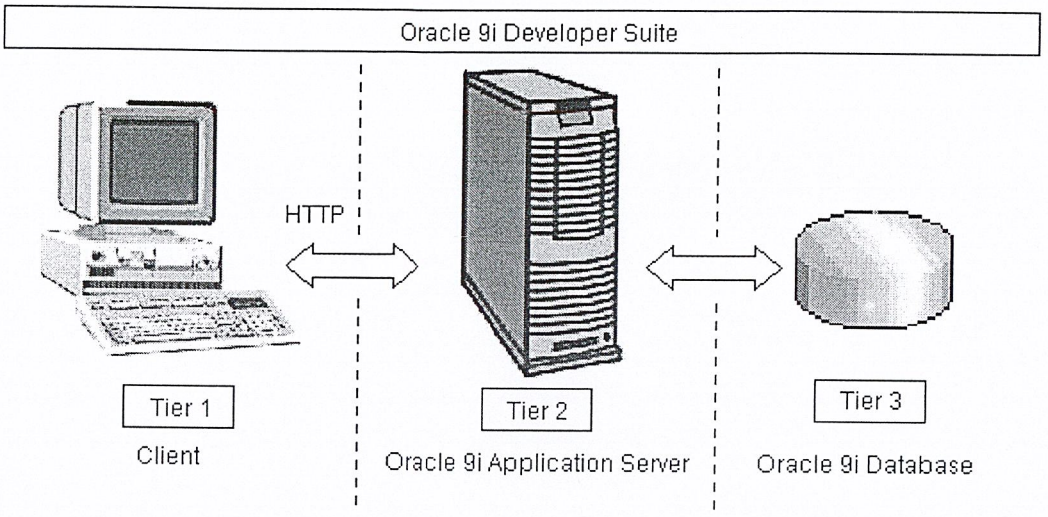
คำสั่งค้นคืนต่างๆจากผู้ใช้จะถูกส่งไปให้กับ Middle Tier หรือส่วนกลางของการบริการเสียก่อน หลังจากนั้น Middle Tier จะแปลงคำสั่งเหล่านี้ให้เป็นภาษา SQL เพื่อส่งไปให้กับระบบจัดการฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้ก็จะถูกส่งกลับคืนไปให้กับ Middle Tier และส่งต่อไปให้ผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบด้วยเครื่องคอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ ซึ่งทำหน้าที่เป็น Middle Tier ก็จะเป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งไคลเอ็นต์และเซิร์ฟเวอร์ฐานข้อมูลสามารถพูดคุยกันได้ การปรับเปลี่ยนระบบคอมพิวเตอร์ของไคลเอ็นต์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนฐานข้อมูลตัวใหม่จะไม่มีผลกระทบซึ่งกันและกันเกิดขึ้น

6.6 สถาปัตยกรรมของ Oracle 9i

ฟังก์ชันการทำงานของ Oracle เวอร์ชัน 9i ได้พัฒนาให้เกิดความง่ายต่อการใช้งานรวมทั้งรองรับการขยายตัวของข้อมูลในอนาคตได้เป็นอย่างดี

สถาปัตยกรรมของ Oracle 9i ถูกออกแบบมาเพื่อรองรับการทำงานแบบ 3-Tier ซึ่งการทำงานแบบนี้จะต่างจากการทำงานแบบ Client/Server ตรงที่การทำงานจากผู้ใช้ที่เป็นไคลเอ็นท์ จะไม่ติดต่อเข้ามายังเซิร์ฟเวอร์โดยตรงเหมือนแบบ Client/Server แต่จะติดต่อเข้ามายังแอพพลิเคชันเซิร์ฟเวอร์ตรงกลางก่อน

จากนั้นแอพพลิเคชันเซิร์ฟเวอร์จะส่งการทำงานนั้นไปยังฐานข้อมูลที่เครื่องเซิร์ฟเวอร์ให้ และเมื่อได้ผลลัพธ์จึงส่งให้กับไคลเอ็นท์ต่อไป ทั้งนี้เพื่อช่วยลดการติดต่อเข้าไปยังเครื่องเซิร์ฟเวอร์หากให้ไคลเอ็นท์ส่งงานเข้าไปทำโดยตรง ดังแสดงในรูป



รูปที่ 6-1 แสดงสถาปัตยกรรมของ Oracle 9i ที่รองรับการทำงานแบบ 3-Tier

สถาปัตยกรรมของ Oracle 9i ประกอบไปด้วย 3 องค์ประกอบดังนี้

- ระบบฐานข้อมูล (Oracle 9i Database) เป็นระบบฐานข้อมูล Oracle เวอร์ชัน 8 โดยเพิ่มฟังก์ชันการทำงานต่างๆ ในการจัดการฐานข้อมูลเพื่อให้ DBA ทำงานได้ง่ายขึ้น และยังมีการจัดการความปลอดภัยของข้อมูลภายในฐานข้อมูลได้ดีมากยิ่งขึ้นเพื่อรองรับการทำงานในระบบอินเตอร์เนตในปัจจุบัน ฐานข้อมูลเวอร์ชัน 9i นี้ถูกออกแบบมาเพื่อให้สามารถทำงานร่วมกับ Oracle 9i Application Server หรือทำงานแบบ Stand Alone แบบเดิมก็ได้เช่นกัน
- ระบบแอพพลิเคชันเซิร์ฟเวอร์ (Oracle 9i Application Server) เป็นแอพพลิเคชันเซิร์ฟเวอร์ที่ถูกออกแบบมาให้รองรับการทำงานกับระบบอินเตอร์เนตที่ต้องการเรียกดูข้อมูลผ่านทางฐานข้อมูล Oracle ซึ่งเป็นการให้บริการแบบ 3-Tier ที่กำลังได้รับความนิยม โดย Oracle 9i Application Server จะทำงานเป็น Middle Tier เพื่อรองรับการทำงานจากไคลเอ็นท์และส่งการทำงานนี้ต่อไปยังฐานข้อมูลที่เครื่องเซิร์ฟเวอร์อีกต่อหนึ่ง
- เครื่องมือในการพัฒนาโปรแกรม (Oracle 9i Developer Suite) ประกอบด้วยเครื่องมือที่ช่วยในการพัฒนาโปรแกรมสำหรับฐานข้อมูล Oracle 9i หลายๆรูปแบบซึ่งช่วยให้โปรแกรมเมอร์สามารถพัฒนาโปรแกรมที่มีประสิทธิภาพสูงและยืดหยุ่นตามความต้องการของผู้ใช้งานมากยิ่งขึ้น เครื่องมือที่ให้มา ได้แก่ Oracle Form Developer , Oracle Designer , Oracle Jdeveloper and Business Components for Java , Oracle Report Developer และ Oracle Discover เป็นต้น เราสามารถแบ่งผลิตภัณฑ์ฐานข้อมูลออกได้เป็น 3 แบบคือ Oracle 9i Enterprise Edition , Standard Edition และ Personal Edition

6.7 ความสามารถของฐานข้อมูล Oracle 9i

ฐานข้อมูล Oracle 9i มีความสามารถที่เป็นจุดเด่น ดังนี้

6.7.1 เปิดฐานขึ้นมาใช้งานได้อย่างรวดเร็ว

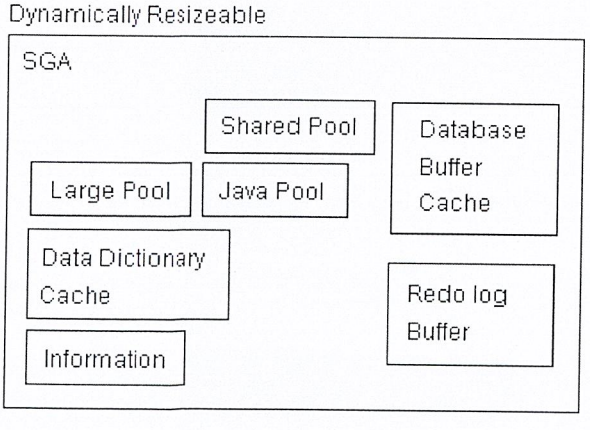
หลังจากที่ฐานข้อมูลหยุดการทำงานไปแบบไม่สมบูรณ์ เราจะสามารถเปิดฐานข้อมูลขึ้นมาใช้งานได้อย่างรวดเร็ว เมื่อเปิดการใช้งานฐานข้อมูลขึ้นมาใหม่ การกู้ Transaction ที่กำลังทำงานอยู่ ณ ขณะที่ฐานข้อมูลหยุดไปนั้นจะใช้ความสามารถที่เรียกว่า Fast Start Fault Recovery โดยจะเปิดให้ผู้ใช้ ใช้งานได้โดยไม่ต้องรอกการ Recovery เสร็จก่อน

6.7.2 การจัดการฐานข้อมูลแบบออนไลน์

คือการทำ Reorganize ตารางและอินเด็กซ์โดยที่ไม่ต้องมีการปิดการใช้งานตารางข้อมูลนั้นๆ โดยแบบเก่าจะต้องใช้การ Export/Import ช่วยในการทำงาน แต่แบบ Oracle 9i สามารถทำงานได้โดยไม่ต้องมีการปิดใช้งานตารางข้อมูลที่ต้องการ จึงช่วยเพิ่มความสะดวกในการทำงานให้กับ DBA และทำให้ประสิทธิภาพในการใช้งานตารางข้อมูลและอินเด็กซ์ดีขึ้นด้วย

6.7.3 ปรับเปลี่ยนค่าหน่วยความจำที่จองไว้ใช้งานได้ทันที

หน่วยความจำที่กล่าวถึงนี้ ได้แก่หน่วยความจำที่เรียกว่า SGA(Syatem Global Area) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับการทำงานฐานข้อมูล แบบเก่าเมื่อต้องการปรับเปลี่ยนค่าของ SGA จะต้องทำการปิดและเปิดฐานข้อมูลใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ใน Oracle 9i สามารถเปลี่ยนแปลงค่าของ SGA สามารถทำได้และจะมีผลลัพธ์ทันทีที่สั่งให้เกิดการเปลี่ยนแปลงโดยที่ไม่ต้องปิดและเปิดฐานข้อมูล

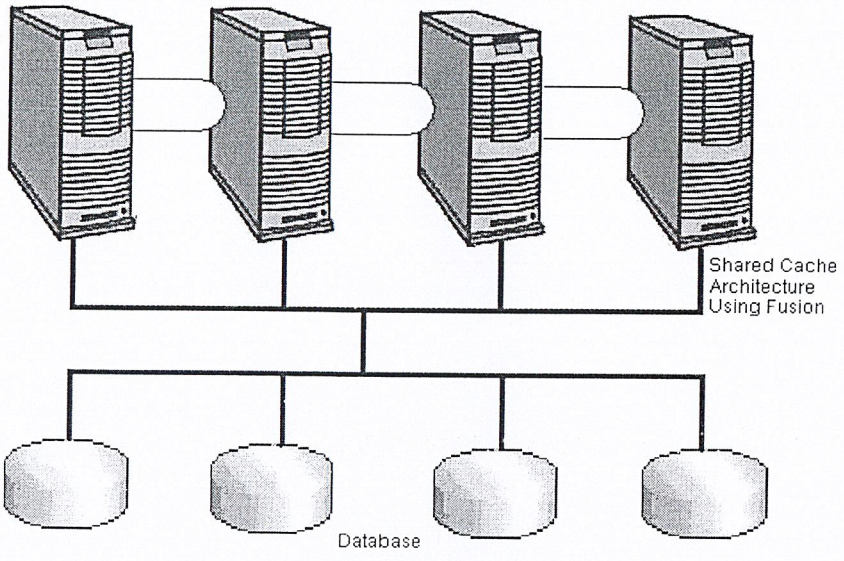


รูปที่ 6-2 รูปแสดง SGA ของฐานข้อมูล

6.7.4 เป็นฐานข้อมูลแบบ Application Cluster ที่แท้จริง

ฐานข้อมูล Oracle 9i เป็น Application Cluster ที่สมบูรณ์แบบ แอพพลิเคชันที่ทำงานอยู่บน Oracle 9i Application Cluster สามารถทำงานได้เหมือนกับทำงานอยู่บนเครื่องเซิร์ฟเวอร์เดียวกัน

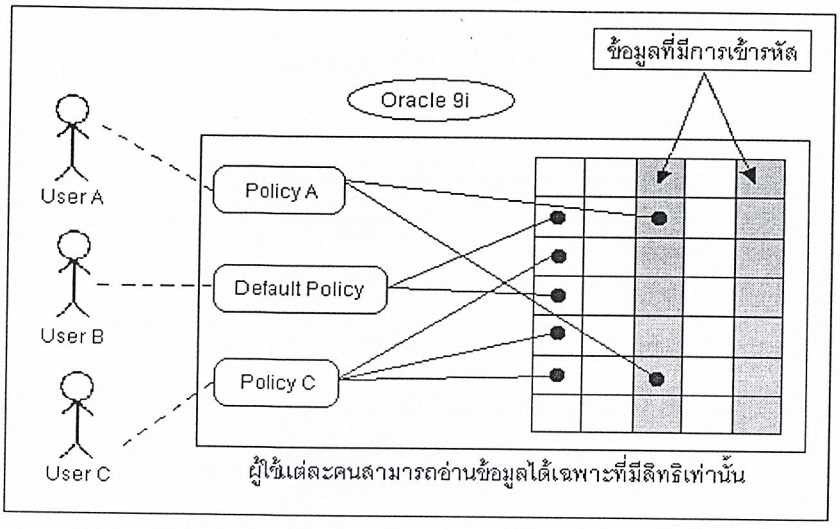
ความสามารถของ Oracle 9i Application Cluster ที่เด่นชัดคือ การทำงานแบบปรับเปลี่ยนโหลดการทำงานของแต่ละเครื่องภายใน Cluster เดียวกันได้โดยอัตโนมัติกรณีที่มีเครื่องใดเครื่องหนึ่งไม่สามารถทำงานได้ ทำให้ผู้ใช้สามารถใช้ฐานข้อมูลได้ตลอดเวลา



รูปที่ 6-3 รูปแบบ Oracle9i Real Application Cluster

6.7.5 การเข้ารหัสเพื่อเพิ่มความปลอดภัยให้กับข้อมูล

ฐานข้อมูล Oracle 9i สนันสนุนและรองรับการเข้ารหัสข้อมูลในระดับคอลัมน์ของตารางได้ เพื่อป้องกันไม่ให้ข้อมูลที่สำคัญรั่วไหลไปยังบุคคลอื่น การเข้ารหัสจะช่วยป้องกันให้ผู้ที่เข้ามาทำงานในฐานข้อมูลได้ ก็ยังไม่สามารถที่จะอ่านข้อมูลที่ถูกเข้ารหัสนี้ได้ ซึ่งเป็นการเพิ่มความปลอดภัยสูงสุดให้กับข้อมูลในฐานข้อมูล



รูปที่ 6-4 รูปแสดงการเข้ารหัสของข้อมูลในฐานข้อมูลเพื่อความปลอดภัยให้กับฐานข้อมูล

6.7.6 รองรับการทำงานกับข้อมูลที่มีขนาดใหญ่ได้เป็นอย่างดี

ฐานข้อมูล Oracle 9i ถูกออกแบบมาเพื่อรองรับการทำงานกับข้อมูลที่มีขนาดใหญ่ และให้ประสิทธิภาพในการทำงานที่สูง การออกแบบตารางข้อมูลและอินเด็กซ์ให้เป็นแบบ Partition เพื่อรองรับการเก็บข้อมูลที่มีขนาดใหญ่

6.7.7 การจัดการ Tablespace กำหนดเป็นแบบ Locally Management

สำหรับ Oracle 9i การสร้าง Tablespace นั้นจะกำหนดค่าดีฟอลต์ของการจัดการพื้นที่ภายในเป็นแบบ Locally Management ทั้งหมด ทำให้ประสิทธิภาพในการทำงานเกี่ยวกับการจองพื้นที่และการคืนพื้นที่ใน Tablespace ทำได้ดีมากขึ้น

6.7.8 การกำหนด Default Temporary Tablespace ให้กับผู้ใช้

โดยปกติคำสั่งในการสร้างผู้ใช้หใม่เข้ามาในฐานข้อมูลหากไม่กำหนด Temporary Tablespace ของผู้ใช้เท่ากับ SYSTEM Tablespace ซึ่งไม่ควรทำเป็นอย่างยิ่ง ดังนั้นใน Oracle 9i จึงมีการกำหนด Default Temporary Tablespace ให้กับผู้ใช้ที่สร้างขึ้นมาใหม่

6.7.9 ลบไฟล์ข้อมูลที่เป็นส่วนประกอบใน Tablespace ให้โดยอัตโนมัติ

ปกติการลบไฟล์ Tablespace ออกจากฐานข้อมูลสำหรับ Oracle อื่นๆที่ไม่ใช่ Oracle 9i นั้นจะไม่ลบไฟล์ข้อมูลที่เป็น Physical File ออกไปจากเครื่องให้ DBA ต้องลบไฟล์เหล่านั้นเองหลังจากที่ได้ลบ Tablespace ออกจากฐานข้อมูลแล้ว ซึ่งอาจจะมีความเสี่ยงของการทำงานที่ผิดพลาดของฐานข้อมูลเกิดขึ้นได้หาก DBA ลบไฟล์ข้อมูลผิดไป และหาก DBA ลืมลบไฟล์ทำให้เกิดการสิ้นเปลืองพื้นที่การใช้งานของดิสก์ และเกิดความสับสนในการทำงานเกิดขึ้นได้

ใน Oracle 9i ฟังก์ชันการทำงานที่น่าสนใจอีกแบบหนึ่งคือ เมื่อลบ Tablespace ออกไปจากฐานข้อมูลแล้วจะลบไฟล์ข้อมูลที่เป็น Physical File ให้ด้วยอัตโนมัติ จึงลดความผิดพลาดที่อาจจะเกิดขึ้นในกรณีที่ DBA ลบไฟล์ข้อมูลผิดไป และทำให้การทำงานของ DBA สะดวกมากยิ่งขึ้น

6.7.10 รูปแบบใหม่ในการจัดการไฟล์ผ่าน Oracle-Managed File

Oracle 9i ได้ลดงานของ DBA ในการจัดการ Physical File โดยนำเสนอการจัดการไฟล์ในรูปแบบใหม่ผ่าน Oracle-Managed File เพื่อช่วยในการสร้างไฟล์ต่างๆ ที่ประกอบเป็น Tablespace ในฐานข้อมูล

Oracle-Managed File หรือเรียกย่อๆได้ว่า OMF จะจัดการสร้างไฟล์ข้อมูลขึ้นมาตามมาตรฐานที่กำหนดเอาไว้เมื่อสร้าง Tablespace ใหม่เพิ่มเข้าไปในฐานข้อมูล เช่นเดียวกันเมื่อมีการลบ Tablespace ออกจากฐานข้อมูล OMF ก็จะลบออกให้โดยอัตโนมัติเช่นกัน

6.7.11 กำหนดเวลาในการกู้ฐานข้อมูลคืนได้

หากฐานข้อมูลเกิดหยุดการทำงานไปแบบไม่สมบูรณ์ เมื่อเปิดการใช้งานขึ้นมาใช้งานอีกครั้งย่อมจะต้องใช้เวลาในการ Recovery Transaction ที่หยุดทำงานไปแบบไม่สมบูรณ์ให้ข้อมูลในฐานข้อมูลกลับคืนมาสู่สภาวะปกติที่ผู้ใช้สามารถทำงานได้ ซึ่งเวลาที่เสียไปในช่วงนี้เราจะเรียกว่า Mean Time To Recover (MTTR)

ใน Oracle 9i มีความสามารถในการทำงานที่ให้ DBA กำหนดช่วงเวลา MTTR นี้ได้เช่นกำหนด MTTR เท่ากับ 10 นาที หมายความว่าหากฐานข้อมูลหยุดการทำงานไปแบบไม่สมบูรณ์ เมื่อเปิดการใช้งานขึ้นมาอีกครั้งจะใช้เวลาทั้งหมดไม่เกิน 10 นาทีตามที่กำหนด เป็นต้น

6.7.12 ขนาดของ DB Block ไม่จำเป็นต้องเท่ากันทุก Tablespace

ขนาดของ DB Block ซึ่งเป็นส่วนที่เล็กที่สุดใน Logical Structure ของฐานข้อมูลนั้นในฐานข้อมูลเวอร์ชันก่อนๆ แต่ละ Tablespace จะต้องมีขนาดของ DB Block เท่ากันและเหมือนกันทั้งหมดเท่านั้น แต่ใน Oracle 9i ได้เพิ่มความสามารถในการที่จะกำหนดให้แต่ละ Tablespace มีขนาดของ DB Block ต่างๆกันได้

บทที่ 7
การออกแบบระบบทะเบียนบัณฑิตวิทยาลัย

โปรแกรมประยุกต์ระบบทะเบียนเป็นส่วนที่สร้างขึ้นมาเพื่อช่วยจัดการงานทะเบียนในด้านการบันทึกข้อมูล การเปิดหลักสูตร การเปิดวิชาเรียน การลงทะเบียน การเพิ่มวิชาเรียน การเปลี่ยนวิชาเรียน การถอนวิชาเรียน เพื่อทำให้เกิดประสิทธิภาพในการให้บริการแก่นักศึกษาและเพื่อรองรับการขยายตัวในอนาคต

7.1 เครื่องมือที่เลือกใช้ในการพัฒนาโปรแกรม

- Server Side Script – JSP (Java Server Page)
- Web Server – Jakarta-Tomcat 4.1.27
- Database Server – Oracle 9i
- Tools ที่ใช้ในการเขียน UML – Rational Rose Enterprise Edition
- Tools ที่ใช้ในการเขียน ER-Diagram – Microsoft Office Visio 2003
- Tools ที่ใช้ในการออกแบบ User Interface – Macromedia Dreamweaver MX 2004

7.2 รายละเอียดการออกแบบของโปรแกรมประยุกต์การจัดการงานสอน

ในการเรียกใช้โปรแกรมประยุกต์การจัดการงานสอนนั้น จะต้องเรียกใช้ผ่านเว็บบราวเซอร์ สำหรับผู้ใช้จะต้องได้รับสิทธิเท่านั้นถึงจะเข้าไปภายในระบบได้

โปรแกรมประยุกต์การจัดการงานสอนได้แบ่งออกเป็นส่วนย่อย ๆ ดังต่อไปนี้

1). ส่วนที่เข้าสู่ระบบ (Login) จะส่วนที่ติดต่อกับผู้ใช้เป็นส่วนแรกเพื่อเข้าสู่ระบบ

2) ส่วนจัดเก็บข้อมูล

3). ส่วนลงทะเบียน

4) ส่วนออกรายงานสรุปต่างๆ

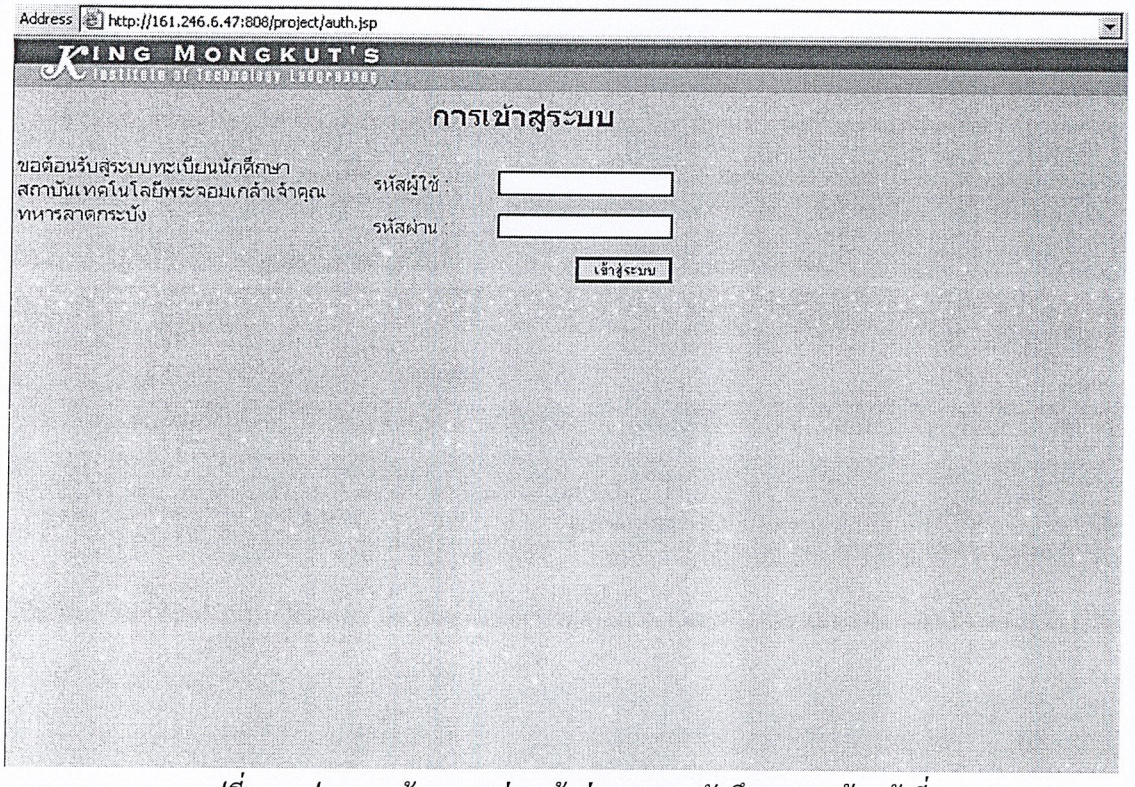
7.3 ส่วนเข้าสู่ระบบ

7.3.1 ส่วนเข้าสู่ระบบของนักศึกษา

- จะต้องลงทะเบียนก่อนแล้วจะได้รับรหัสผู้ใช้ กับรหัสผ่านมาเพื่อใช้ในการเข้าสู่ระบบสามารถแก้ไขรหัสผ่านภายหลังได้โดยจะได้รับสิทธิในการทำงานเพียงแค่ลงทะเบียนและดูเกรดเท่านั้น

7.3.2 ส่วนเข้าสู่ระบบเจ้าหน้าที่

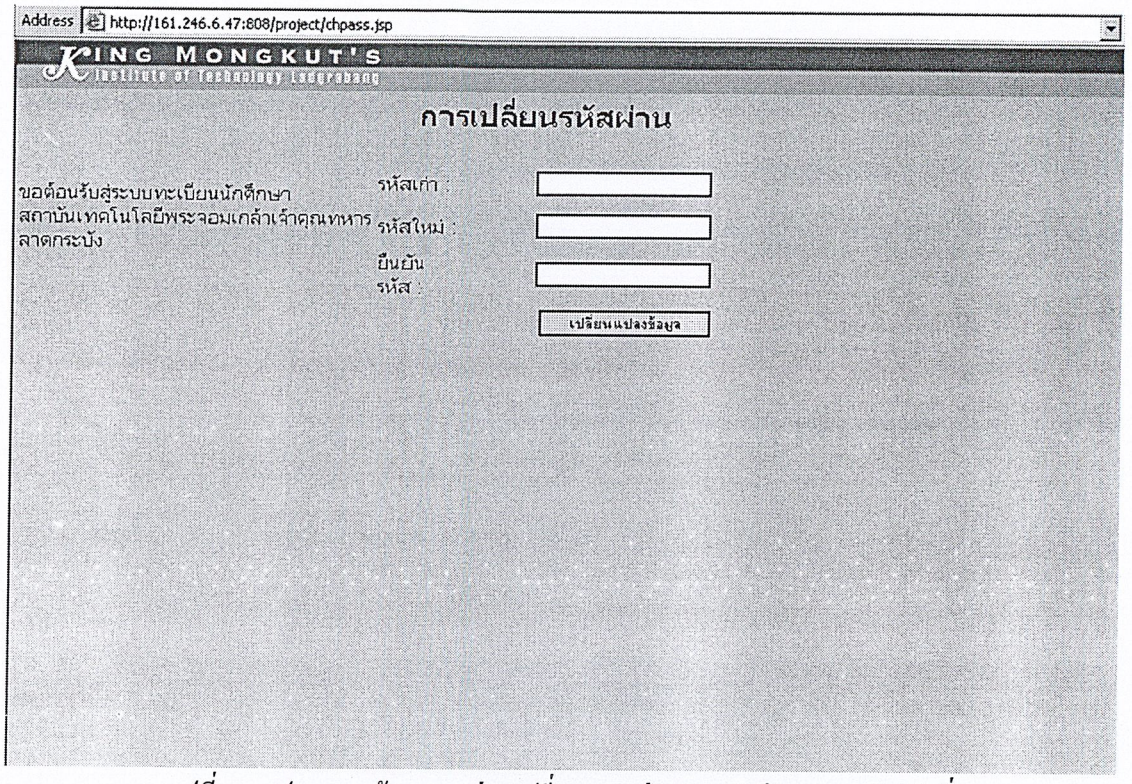
- จะได้รับรหัสผู้ใช้ กับรหัสผ่านจากผู้ดูแลระบบมาเพื่อใช้ในการเข้าสู่ระบบสามารถแก้ไขรหัสผ่านภายหลังได้โดยจะได้รับสิทธิในการทำงานเพียงแค่ลงทะเบียนและดูเกรดเท่านั้น



รูปที่ 7-1 รูปแสดงหน้าจอของส่วนเข้าสู่ระบบของนักศึกษาและเจ้าหน้าที่

7.3.3 ส่วนการเปลี่ยนรหัสผ่านของทั้งนักศึกษาและเจ้าหน้าที่

- ทำการป้อนรหัสเก่าและรหัสใหม่กับการยืนยันรหัสใหม่เพื่อทำการเปลี่ยนรหัสผ่าน

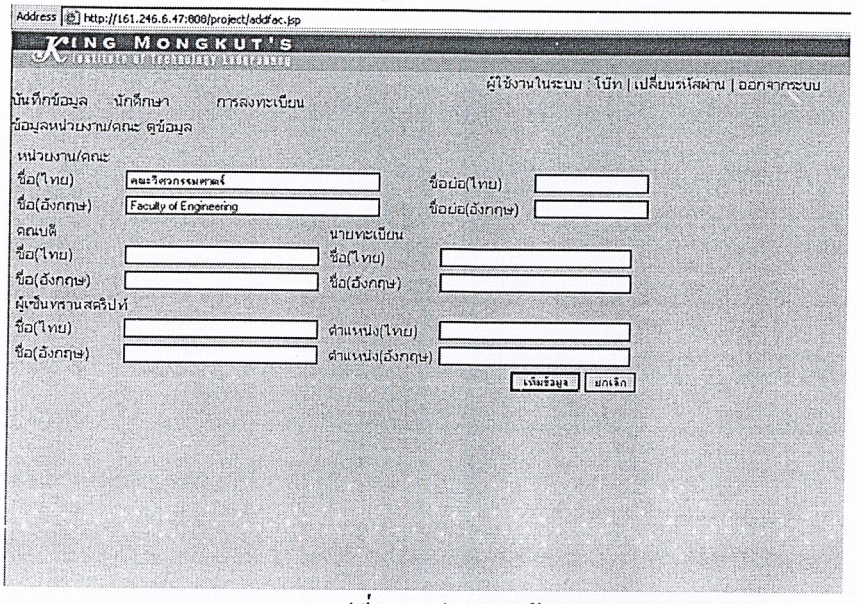


รูปที่ 7-2 รูปแสดงหน้าจอของส่วนเปลี่ยนรหัสผ่านของนักศึกษาและเจ้าหน้าที่

7.4 จัดเก็บข้อมูล

7.4.1 ข้อมูลคณะ

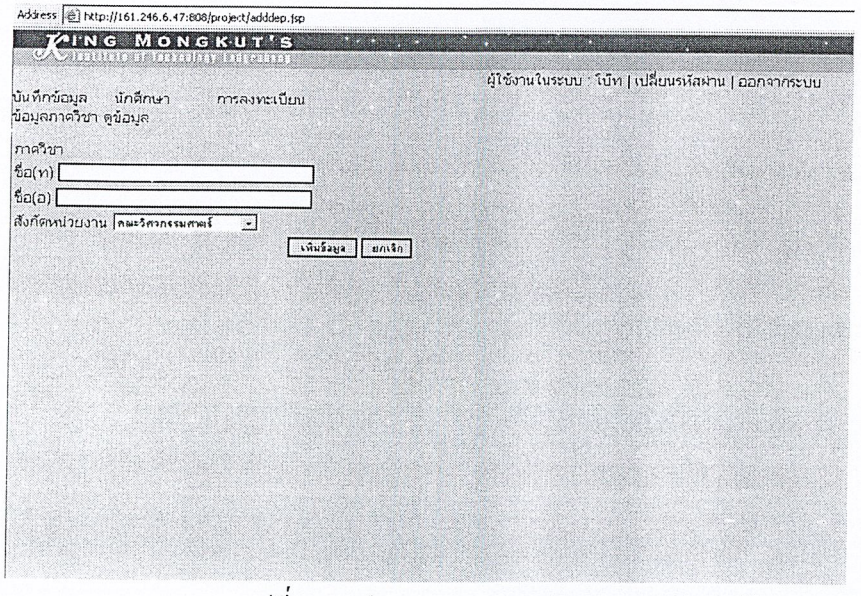
- ทำการจัดเก็บข้อมูล ชื่อคณะ คณบดี และ นายทะเบียนทั้งภาษาไทยและ ภาษาอังกฤษ ตามช่องที่สร้างไว้แล้วกดปุ่มเพิ่มข้อมูล



รูปที่ 7-3 รูปแสดงหน้าจอการจัดเก็บข้อมูลคณะ

7.4.2 ข้อมูลภาควิชา

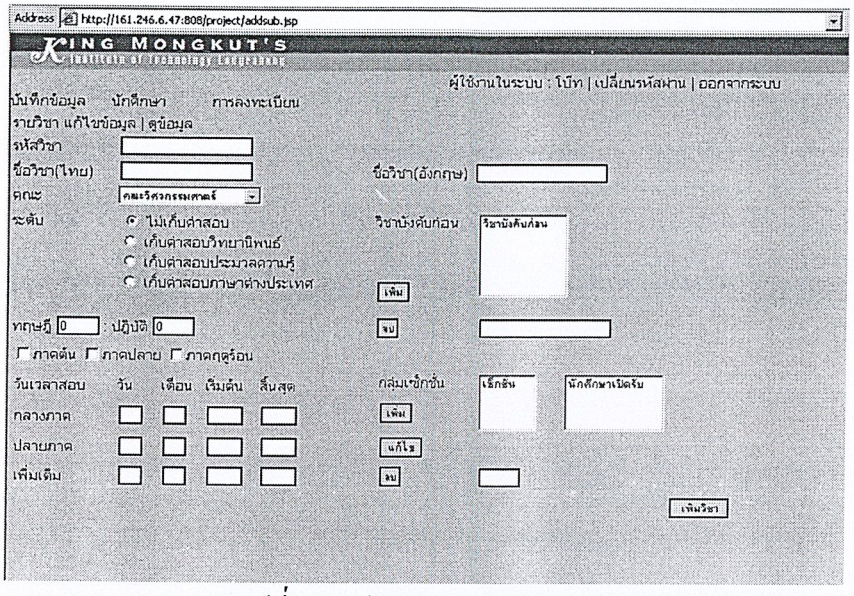
- ทำการจัดเก็บข้อมูลภาควิชาและเป็นภาควิชาในสังกัดคณะอะไรตามช่องที่สร้างไว้แล้วกดปุ่มเพิ่มข้อมูล



รูปที่ 7-4 รูปแสดง user interface ของการจัดเก็บข้อมูลภาควิชา

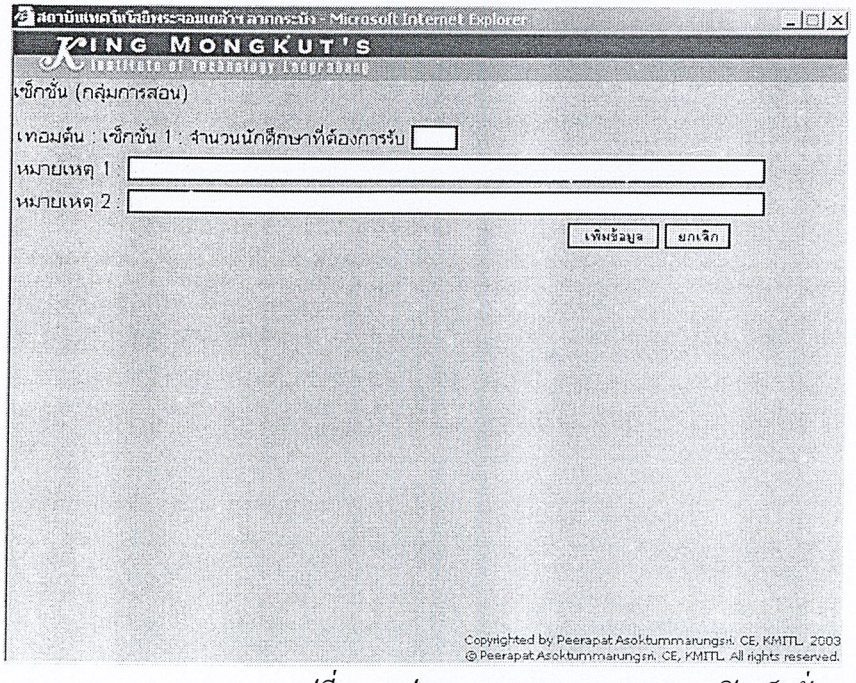
7.4.3 ข้อมูลรายวิชา

- ทำการจัดเก็บข้อมูลรายวิชา วิชาบังคับก่อนเรียน เทอมที่จะเปิดสอนและวันเวลาสอบ



รูปที่ 7-5 รูปแสดง user interface ของการจัดเก็บข้อมูลรายวิชา

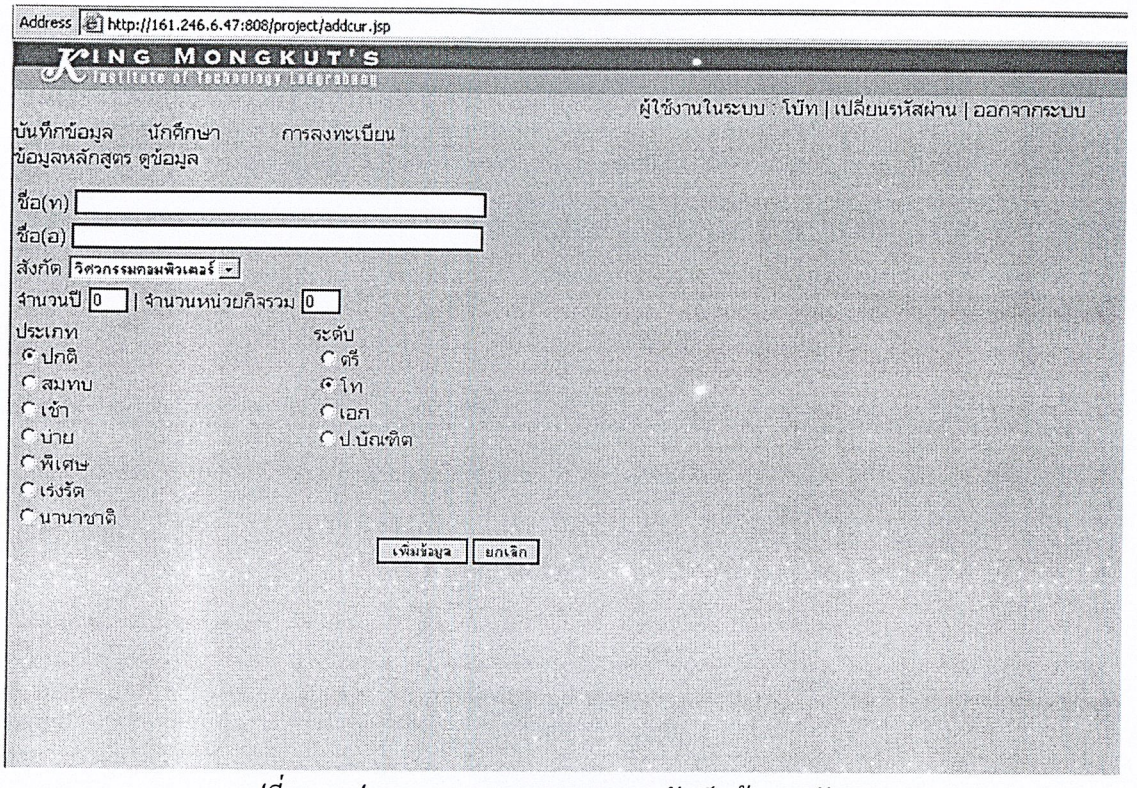
- ทำการเปิด เซ็กชัน จำนวนนักศึกษาที่ต้องการรับ



รูปที่ 7-6 รูปแสดง user interface ของการเปิดเซ็กชั่น

7.4.4 ข้อมูลหลักสูตร

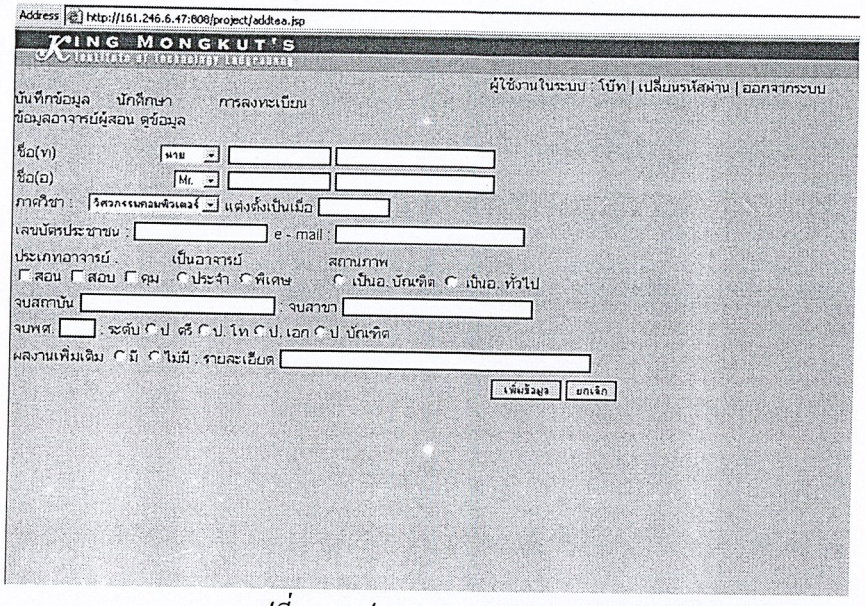
- ทำการจัดเก็บข้อมูลโครงสร้างหลักสูตร ชื่อหลักสูตร จำนวนปีที่เรียน จำนวนหน่วยกิจรวมของหลักสูตรและเป็นหลักสูตรในสังกัดคณะใด



รูปที่ 7-7 รูปแสดง user interface ของการจัดเก็บข้อมูลหลักสูตร

7.4.5 ข้อมูลอาจารย์

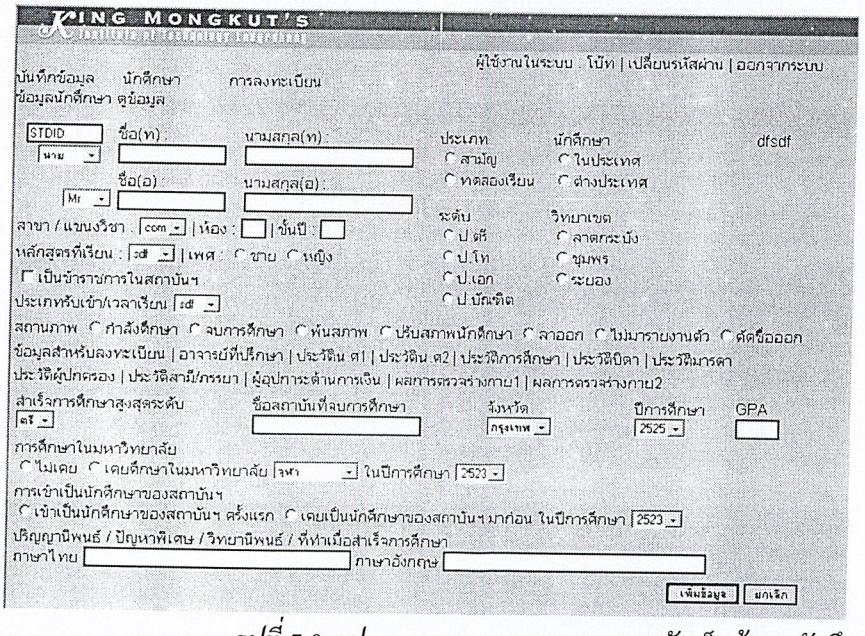
- ทำการจัดเก็บข้อมูลอาจารย์



รูปที่ 7-8 รูปแสดง user interface ของการจัดเก็บข้อมูลอาจารย์

7.4.6 ข้อมูลนักศึกษา

- ทำการจัดเก็บข้อมูลนักศึกษา

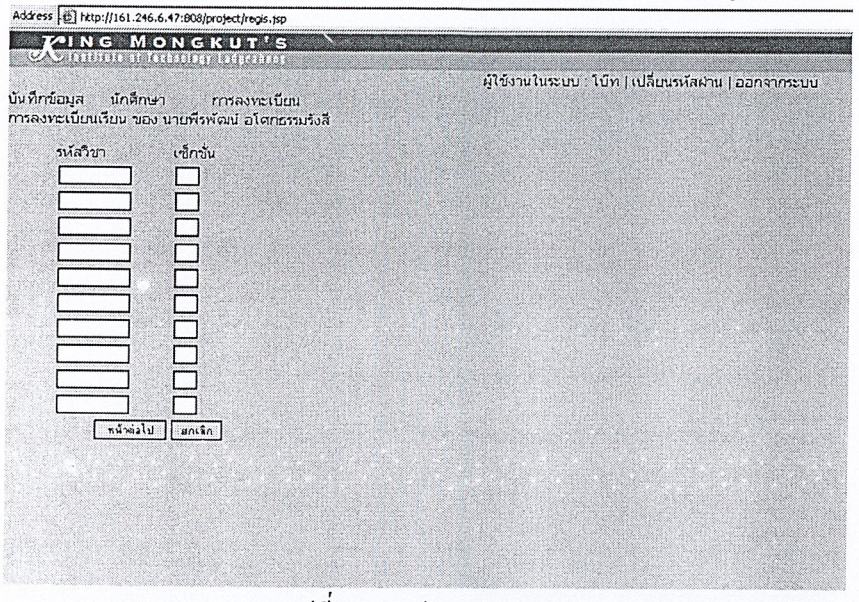


รูปที่ 7-9 รูปแสดง user interface ของการจัดเก็บข้อมูลนักศึกษา

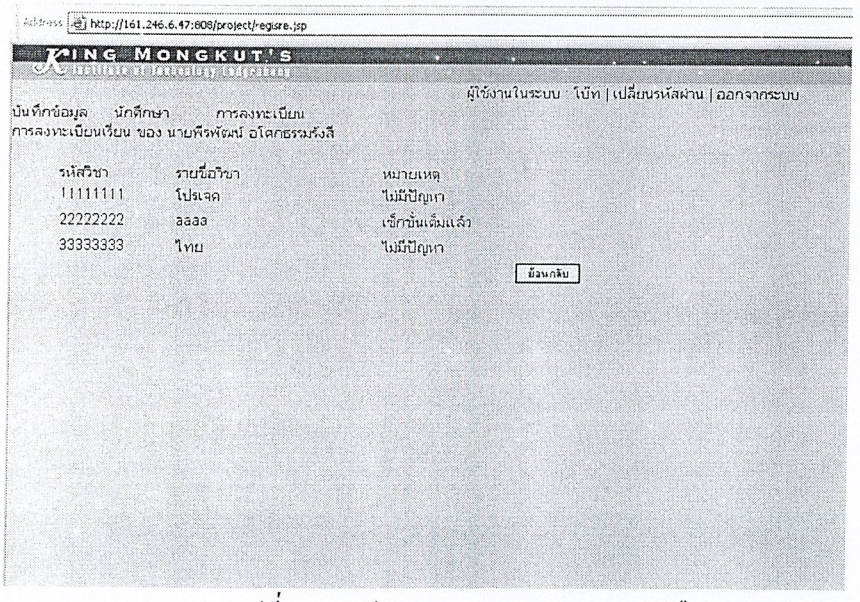
7.5 การลงทะเบียน

7.5.1 การลงทะเบียน

- ป้อนรายวิชาที่จะลงทะเบียน และ เช็กชันที่ต้องการจากนั้นกดปุ่มเพื่อไปหน้าต่อไป
- จากนั้นระบบจะทำการโชว์ว่าจะสามารถลงทะเบียนได้หรือไม่มีปัญหาอะไรเกิดขึ้นไหม ถ้าไม่มีก็สามารถที่จะลงทะเบียนได้ ถ้ามีก็ต้องทำการแก้ไขให้ถูกต้องก่อนทำการลงทะเบียน

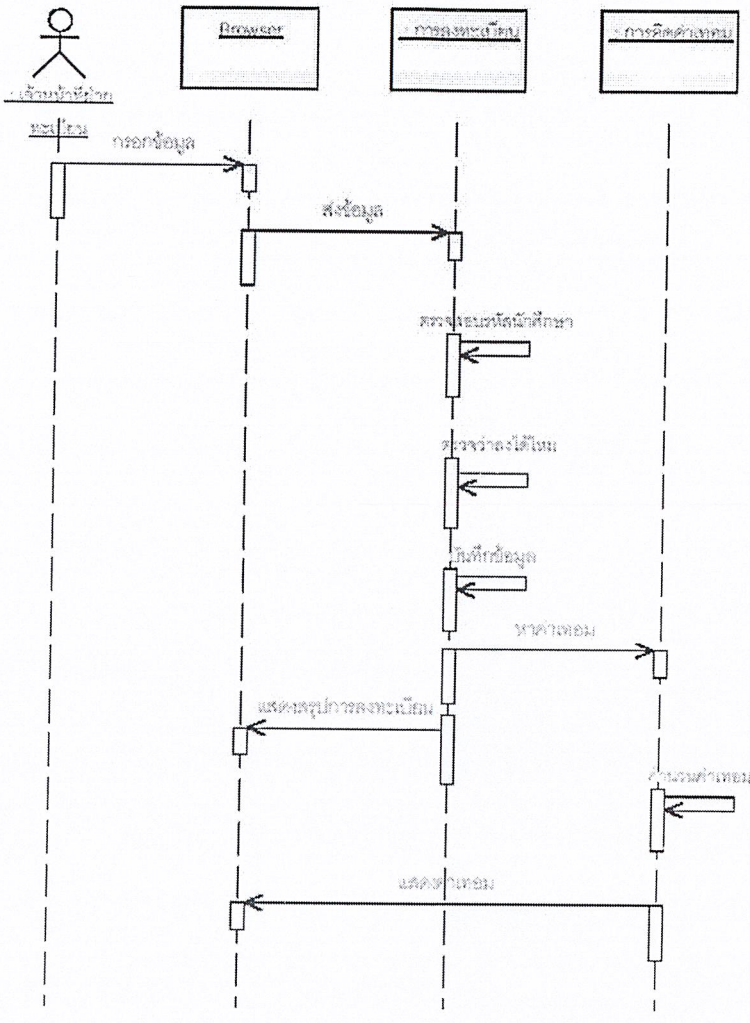


รูปที่ 7-10 รูปแสดง user interface ของการลงทะเบียน



รูปที่ 7-11 รูปแสดง user interface ของการยืนยันการลงทะเบียน

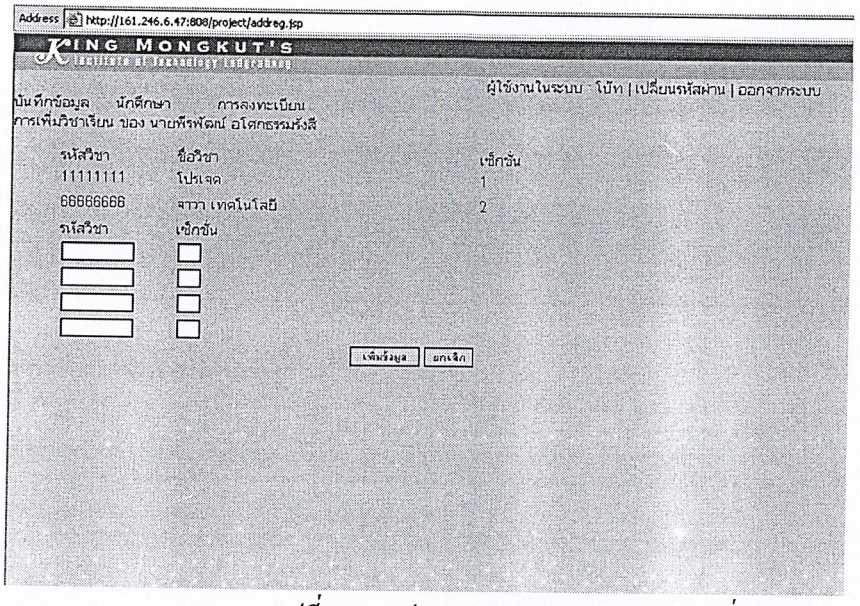
- สเตทไดอะแกรมแสดงขั้นตอนการลงทะเบียน



รูปที่ 7-12 รูปแสดง state diagram ของการลงทะเบียน

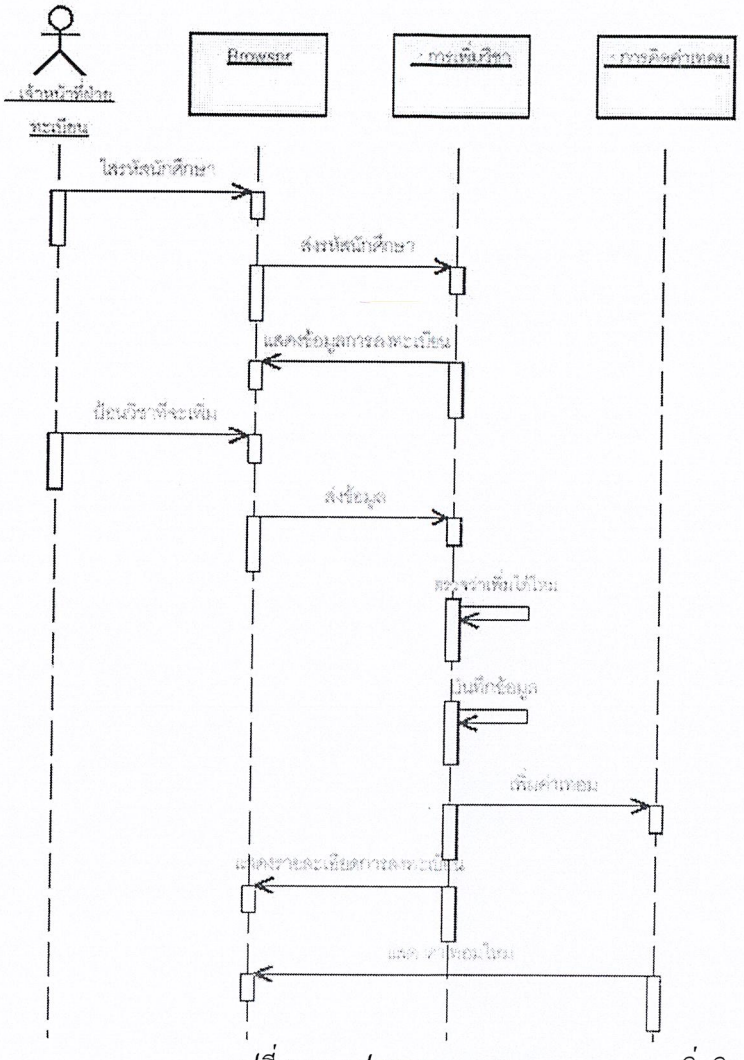
7.5.2 การเพิ่มวิชาเรียน

- ระบบจะแสดงวิชาที่ได้ลงทะเบียนไปแล้วให้เราทำการเพิ่มในรายวิชาที่ต้องการจากนั้นกดปุ่มเพิ่มรายวิชาเรียน



รูปที่ 7-13 รูปแสดง user interface ของการเพิ่มวิชาเรียน

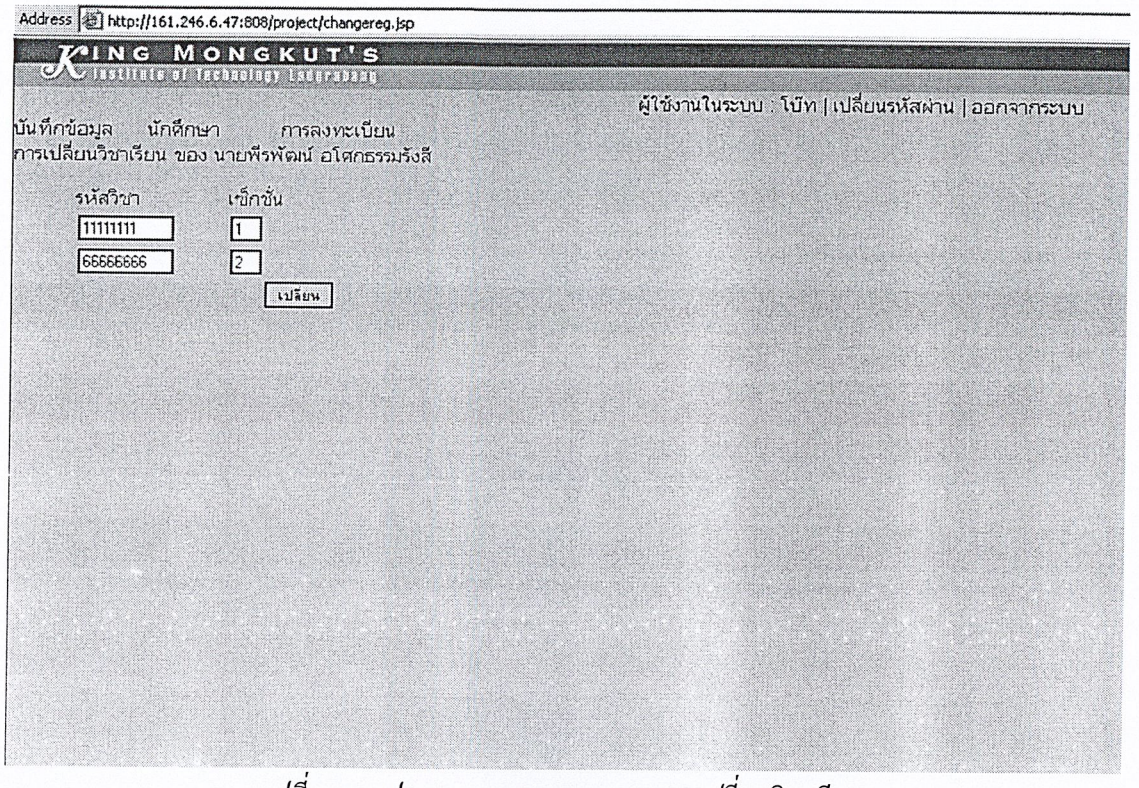
- สเตทไดอะแกรมแสดงขั้นตอนการเพิ่มวิชาเรียน



รูปที่ 7-14 รูปแสดง state diagram ของการเพิ่มวิชาเรียน

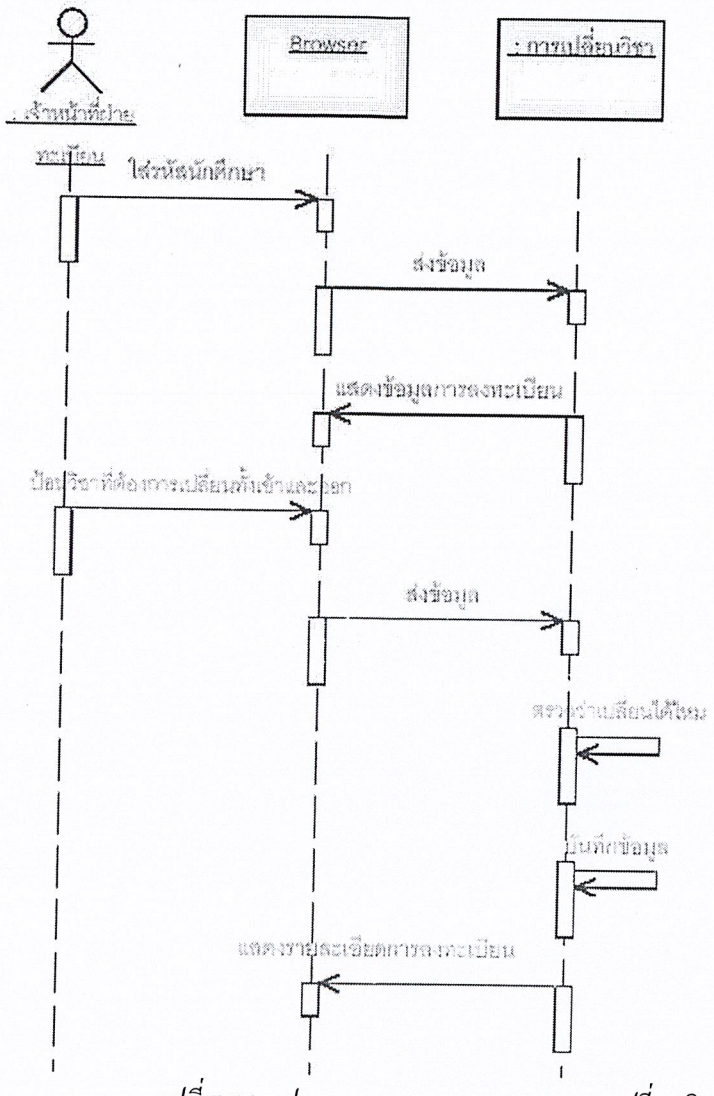
7.5.3 การเปลี่ยนวิชาเรียน

- ทำการเปลี่ยนวิชาเรียนในรายวิชาที่ต้องการ จากนั้นจึงทำการตกลง



รูปที่ 7-15 รูปแสดง user interface ของการเปลี่ยนวิชาเรียน

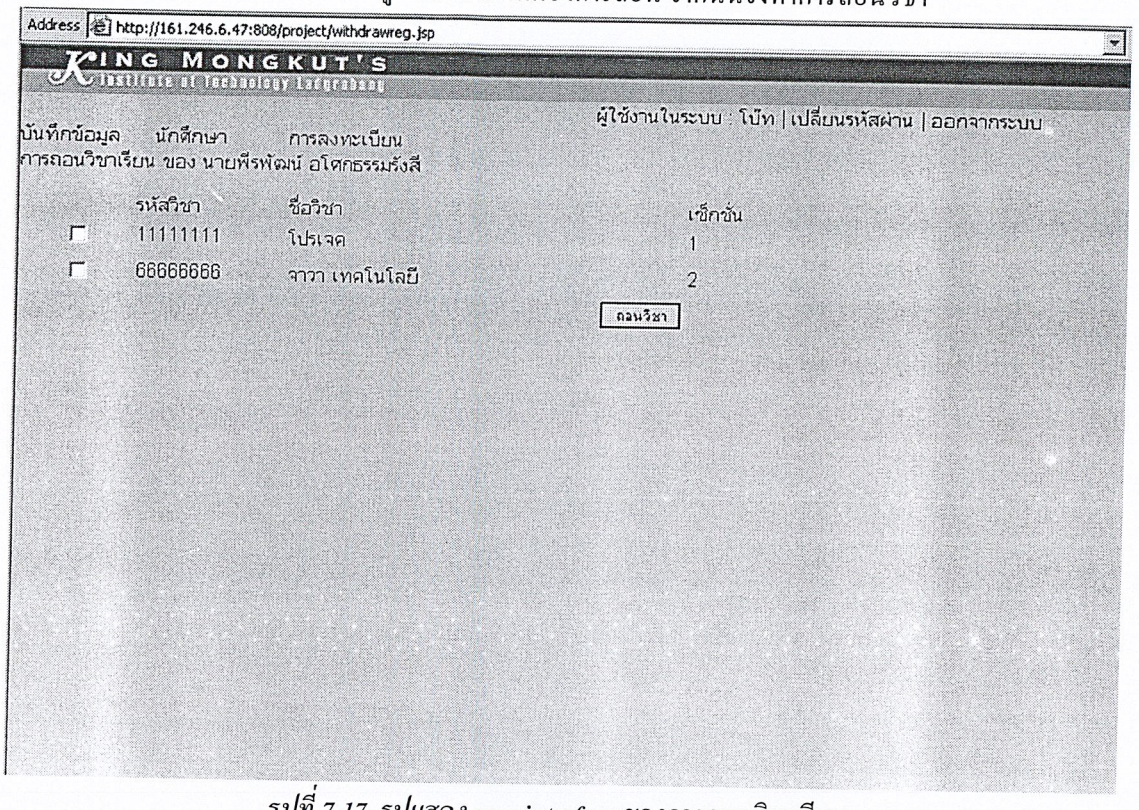
- สเตทไดอะแกรมแสดงขั้นตอนการเปลี่ยนวิชาเรียน



รูปที่ 7-16 รูปแสดง state diagram ของการมเปลี่ยนวิชาเรียน

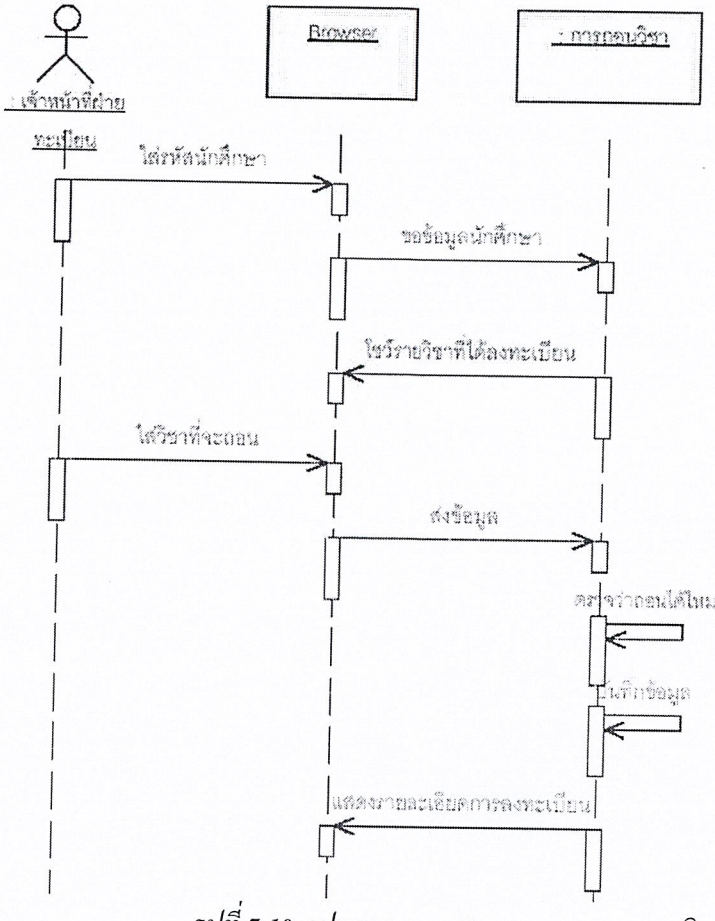
7.5.4 การถอนวิชาเรียน

- เลือกโดยใส่เครื่องหมายถูกในรายวิชาที่ต้องการถอน จากนั้นจึงทำการถอนวิชา



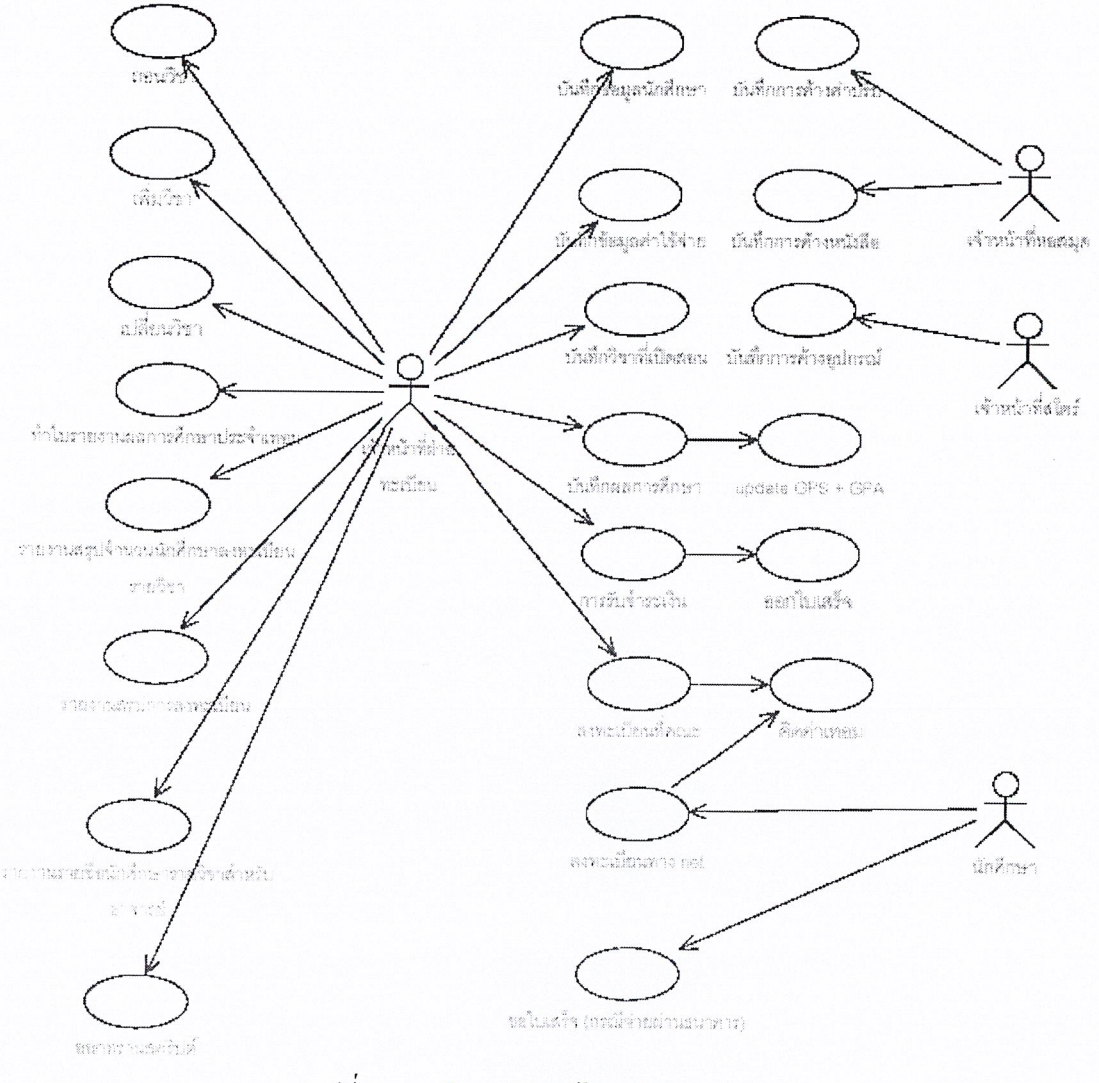
รูปที่ 7-17 รูปแสดง user interface ของการถอนวิชาเรียน

- สเตทไดอะแกรมแสดงขั้นตอนการถอนวิชาเรียน



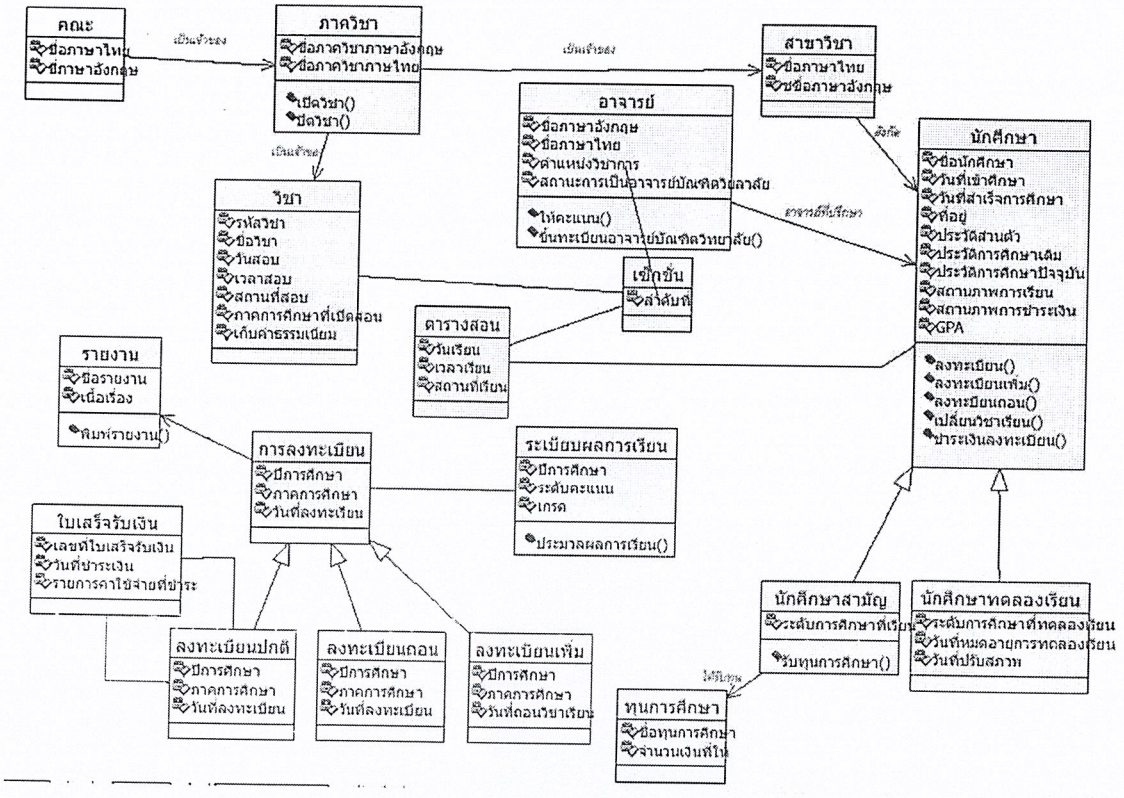
รูปที่ 7-18 รูปแสดง state diagram ของการถอนวิชาเรียน

7.6 ยูสเคสไดอะแกรมของระบบ



รูปที่ 7-19 รูปแสดงยูสเคสไดอะแกรมของระบบ

7.7 คลาสไดอะแกรมของระบบ



รูปที่ 7-20 รูปแสดงคลาสไดอะแกรมของระบบ

บทที่ 8
สรุปผลการทดลองและข้อเสนอแนะ

8.1 ผลการวิจัยและพัฒนา

การดำเนินโครงงานพัฒนาโปรแกรมระบบจัดการเอกสารได้ดำเนินการลุล่วงไปเป็นอย่างดี โดยผลความสำเร็จของโครงงานจะได้นำเสนอต่อไป

จากที่ได้กล่าวมาแล้วในบทนำว่าโปรแกรมระบบทะเบียนบัณฑิตวิทยาลัยนั้นได้ถูกออกแบบและพัฒนามาเพื่อที่จะนำเสนอวิธีการจัดการระบบทะเบียนบัณฑิตวิทยาลัย ด้วยการวิเคราะห์และออกแบบระบบอย่างรอบคอบ ในช่วงของการออกแบบได้มีการนำเสนอให้แก่อาจารย์ที่ปรึกษาเพื่อแนะนำและให้คำปรึกษา โดยการออกแบบได้มีการกลั่นกรองเป็นอย่างดี เนื่องจากวัตถุประสงค์ในการสร้างระบบจัดการอุปกรณ์นี้ขึ้นมีจุดประสงค์ที่จะให้โปรแกรมระบบเอกสารสามารถรองรับการพัฒนาต่อให้ระบบมีความสามารถเพิ่มเติมเนื่องจากการพัฒนาเป็นแนวทางแบบ **API** ได้ต่อไปในอนาคต จากนั้นทางคณะผู้จัดทำได้ทำการพัฒนาโปรแกรมด้วยความเอาใจใส่และพิถีพิถันในทุกรายละเอียด โดยในระหว่างการพัฒนาจะมีการตรวจสอบการทำงานของแต่ละส่วนเสมอ เพื่อหาข้อผิดพลาดที่อาจจะเกิดขึ้นได้ เมื่อตรวจพบก็จะแก้ไขในทันที

หลังจากการที่ได้พัฒนาระบบจัดการเอกสารในทุกฟังก์ชันการทำงานตามที่ได้ออกแบบแล้ว ก็จะมีการตรวจสอบการทำงานโดยการจำลองสถานการณ์ขึ้นมา เพื่อจะแสดงให้เห็นถึงความสามารถในการทำงานของระบบ ในขั้นตอนการทดสอบ เมื่อเกิดปัญหาขึ้นทางคณะผู้จัดทำได้ทำการแก้ไขในส่วนที่ผิดพลาด และได้มีการคิดหาวิธีทางที่เหมาะสมขึ้นอยู่อย่างสม่ำเสมอ เพื่อให้แน่ใจว่าโปรแกรมระบบจัดการเอกสารจะเป็นโปรแกรมที่สามารถทำงานได้อย่างถูกต้อง และที่สำคัญ สามารถตอบสนองงานในการจัดการระบบทะเบียนบัณฑิตวิทยาลัยได้เป็นอย่างดี

8.1.1 การกำหนดรายการปัญหา

งานวิจัยนี้ได้เริ่มต้นตรงส่วนของการกำหนดรายการปัญหาเป็นส่วนแรก โดยการกำหนดรายการปัญหานี่ผู้วิจัยได้ทำโดยการศึกษาระบบงานเดิมเพื่อทำการรวบรวมข้อมูล โดยทำการศึกษา ทั้งจากการติดตามการทำงานและสอบถามเจ้าหน้าที่บัณฑิตวิทยาลัย ศึกษาคู่มือหลักสูตรระดับบัณฑิตศึกษาศึกษา ศึกษาระเบียบบัณฑิตวิทยาลัยสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังและศึกษาเอกสารรายงานต่างๆที่ได้ใช้งานอยู่เดิม และนำมากำหนดเป็นรายการปัญหาเพื่อเป็นแนวทางในการพัฒนาระบบงาน

8.1.2 การวิเคราะห์

ในขั้นตอนการวิเคราะห์ ผู้วิจัยได้นำรายการปัญหามาทำการวิเคราะห์ ด้วยการสร้างแบบจำลอง โดยสร้างแบบจำลองออปเจ็กเป็นลำดับแรก เพื่อให้ได้เห็นโครงสร้างและความสัมพันธ์ระหว่างออปเจ็กในระบบงานจากนั้นจึงได้พิจารณาเพิ่มรายละเอียดของแอทริบิวส์ และโอเปอเรชั่นให้กับออปเจ็ก

จากนั้นได้ทำการสร้างแบบจำลองไดนามิก โดยใช้แผนภาพลำดับเหตุการณ์เพื่อแสดงเหตุการณ์ที่เกิดขึ้นในระบบ และได้สร้างเป็นแผนภาพทางเดินของเหตุการณ์และสเตทไดอะแกรมเพื่อแสดงถึงการตอบสนองในการทำงานของออปเจ็กกับเหตุการณ์ที่เกิดขึ้น

และในส่วนของการสร้างแบบจำลองฟังก์ชั่น ได้ใช้แผนภาพดาต้าโฟไดอะแกรม เพื่ออธิบายการเปลี่ยนแปลงของข้อมูลในระบบที่เกิดขึ้นจากการทำกระบวนการต่างๆ โดยกระบวนการต่างๆได้มาจากเหตุการณ์ในขั้นตอนการสร้างไดนามิกโมเดลนั่นเอง

8.1.3 การออกแบบระบบ

ในการออกแบบระบบสรุปได้ว่า

1. ระบบจะแบ่งออกเป็นระบบงานย่อยคือ
    - งานบันทึกข้อมูลหลัก ได้แก่ข้อมูลคณะ, ภาควิชา, รายวิชา, หลักสูตร, ค่าใช้จ่าย , อาจารย์, ฯลฯ
    - งานบันทึกข้อมูลนักศึกษา
    - งานบันทึกการลงทะเบียนเรียน ประกอบด้วย การลงทะเบียน การเพิ่ม การเปลี่ยน การถอน
    - งานการรับชำระค่าธรรมเนียมการศึกษาและพิมพ์ใบเสร็จรับเงิน
    - งานบันทึกและประมวลผลการศึกษาและพิมพ์รายงานผลการศึกษา
    - งานสอบถามข้อมูลใช้การสร้างรายงานเพื่อแสดงข้อมูลที่มักถูกสอบถามอยู่เป็นประจำ
2. โครงสร้างทางสถาปัตยกรรมของระบบสารสนเทศนักศึกษาแบบเว็บแอพพลิเคชั่นบนระบบเครือข่ายของสถาบัน
3. ในการจัดการฐานข้อมูลเลือกใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ **oracle9i**

8.1.4 การออกแบบรายละเอียด

ได้ทำการโดเมนและสร้างคีย์ของแอทริบิวส์ให้กับคลาสที่ได้จากการสร้างของออปเจ็กโมเดล และได้แปลงคลาสไปเป็นตาราง

8.1.5 การพัฒนาระบบงาน

ได้ทำบรรทัดฐานข้อมูลให้กับตารางที่สร้างไว้และได้กำหนดชนิดและขนาดของข้อมูลในตารางเพื่อนำไปสร้างเป็นตารางสำหรับฐานข้อมูลเชิงสัมพันธ์

8.2 แนวทางการใช้งาน

- ระบบสารสนเทศนักศึกษาจะถูกนำไปใช้ในหน่วยงานต่างๆ ที่เกี่ยวข้อง
- การเชื่อมต่อระหว่างอินเทอร์เฟสกับฐานข้อมูลและส่วนประมวลผลจะทำผ่านเครือข่ายของสถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง

8.3 ข้อเสนอแนะ

ระบบสารสนเทนักศึกษาที่พัฒนาขึ้นโดยใช้วิธีการเชิงวัตถุ เป็นเครื่องมือในการพัฒนานั้น ได้ใช้ฐานข้อมูลแบบรีเลชั่นแนลดาต้าเบสแต่หากในอนาคตจะมีการนำฐานข้อมูลเชิงวัตถุมาใช้งาน ก็ไม่จำเป็นต้องออกแบบระบบงานใหม่ สามารถนำระบบที่ออกแบบไว้ไปอิมพลิเมนต์ได้ทันที