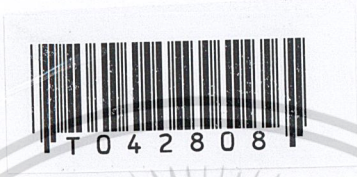
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบจัดเก็บสถิติสำหรับเว็บเพจ

WEB STAT



นายจิราเจตน์ อ่อนสา
นายเอกชัย ลาน้ำคำ

จ 5315
2543

เลขหมู่.....
เลขทะเบียน 42808
วัน, เดือน, ปี 10 มิ.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดเก็บสถิติสำหรับเว็บเพจ

(Web Stat)

โดย

นาย จิราเจตน์ อ่อนสา 41013524

นาย เอกชัย ลาน้ำคำ 41013562



อาจารย์ที่ปรึกษา

รศ. บรรจง ปียธำรง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจัดเก็บสถิติสำหรับเว็บเพจ

WEB STAT

ผู้จัดทำ

1. นาย จิราเจตน์ อ่อนสา
2. นาย เอกชัย ลาน้ำคำ



..... อาจารย์ที่ปรึกษา

(รศ. บรรจง ปิยธำรง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดเก็บสถิติสำหรับเว็บเพจ

นายจิราเจตน์ อ่อนสา 41013524
นายเอกชัย ลาน้ำคำ 41013562
รศ.บรรจง ปิยธำรง อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ปัจจุบันโลกของอินเทอร์เน็ตได้เติบโตขึ้นอย่างรวดเร็ว มีเว็บไซต์เกิดใหม่เปิดตัวสู่สายตาชาวโลกมากมายไม่เว้นชั่วโมง บางเว็บไซต์ได้รับความนิยมอย่างล้นหลาม ในขณะที่อีกหลายเว็บไซต์กลับไม่ได้รับความนิยมนักในขณะที่ทุ่มทุนสร้างอย่างมหาศาลนั่นหมายถึงตัวเงินอันมีค่า กำไรหรือขาดทุน ขึ้นอยู่กับความพึงพอใจของผู้เยี่ยมชมเว็บไซต์เป็นหลักหากเว็บไซต์ใดสามารถสร้างความพึงพอใจให้กับผู้เยี่ยมชมเว็บไซต์ได้แล้ว นั่นหมายถึงการเข้ามาเยี่ยมชมครั้งต่อๆไป และการบอกต่อคนสนิท ซึ่งจะทำให้เว็บไซต์เหล่านั้นได้รับความนิยมและประสบความสำเร็จได้

งานวิจัยนี้ได้พัฒนาระบบให้บริการจัดเก็บสถิติและแสดงผลสำหรับเว็บเพจแบบออนไลน์ เพื่อใช้ในการจัดเก็บข้อมูลของผู้เข้าเยี่ยมชมเว็บเพจของผู้รับบริการ โดยข้อมูลที่สามารถจัดเก็บมาทำสถิติได้นั้นจะช่วยให้ผู้บริหารเว็บไซต์สามารถนำมาใช้ในการช่วยตัดสินใจในการวางนโยบายสำหรับพัฒนาเว็บไซต์ในระดับเบื้องต้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Stat

Jirarjej Onsar

Ekachai Lanamkum

Assoc.Prof.Banjong Piyatamrong

ABSTRACT

In the world of the Present, Internet is quickly grow up. Many new websites are visible. Some websites are populars while many websites are extreme. Although these websites are acquired with high investment that mean to lose or profit. These are dependent satisfy of people.If website can make for visitor.That mean access again and talk to friend which make these website are popular and sucessful.

This research is develop serve system to keep statics and show result for online webpage. It's use to keep information of vistor who is access member's webpage. These information are used to support manager consider to planning for develop website basic level.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จสิ้นได้ด้วยดี ถ้าไม่ได้รับการสนับสนุนช่วยเหลือจากบุคคลหลายฝ่าย ทั้งนี้ต้องขอขอบคุณบุคคลที่ถือว่าเป็นส่วนสำคัญอย่างยิ่ง ที่ให้การช่วยเหลือตั้งแต่เริ่มต้นโครงงานวิจัย จวบจนเสร็จสิ้นโครงการวิจัย นั่นคือ อาจารย์บรรจง ปิยธำรง ซึ่งถือได้ว่า ถ้าไม่ได้รับความช่วยเหลือด้านคำแนะนำและแนวทางในการดำเนินงานจากอาจารย์แล้ว โครงงานวิจัยนี้ก็ไม่อาจจะสำเร็จลงได้ด้วยดี ขอขอบคุณ คุณพ่อ คุณแม่ ที่ให้ชีวิต ให้ความคิด ให้การอบรมบ่มนิสัยที่ดี แก่ผู้จัดทำ ให้การเลี้ยงดูที่ดีมาโดยตลอด ขอขอบคุณเพื่อนๆ ทุกคนที่คอยให้กำลังใจด้วยดีเสมอมา

จิราเจตน์ อ่อนสา
เอกชัย ลาน้ำคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ภาษาจาวา	4
2.1 ความสามารถของภาษาจาวา	4
2.2 การนำจาวาไปประยุกต์สร้างและพัฒนาในงานด้านต่าง ๆ	5
2.3 สภาพแวดล้อมเสมือน (Java Virtual Machine)	6
2.4 หลักการทำงานของภาษาจาวา	7
2.4.1 การคอมไพล์	7
2.4.2 การวางตำแหน่งในหน่วยความจำ	8
2.4.3 การรันโปรแกรม	8
2.4.4 Class Loader	9
2.4.5 ตรวจสอบไบต์โค้ด	9
2.4.6 การทำงานตามโค้ด	10
2.4.7 การสร้างและรันภาษาจาวา	10
2.5 เว็บบราวเซอร์ที่สนับสนุนจาวา	11
2.6 ความแตกต่างระหว่างจาวาแอพเพล็ต และจาวาแอพพลิเคชัน	12
บทที่ 3 Java Servlet	13
3.1 ความหมายของจาวาเซิร์ฟเล็ท ?	13
3.2 Servlet Engine	13
3.2.1 ความหมายของ Servlet Engine	13
3.2.1 การทำงานของ Servlet Engine ?	14
3.2.2 Interface javax.servlet.Servlet	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

3.3 Hello World 20
3.4 การรัน Servlet 22
บทที่ 4 Java Script 30
4.1 ความหมายของ JavaScript 30
4.2 Java กับ JavaScript 31
4.3 JavaScript กับ HTML 32
4.3 ความสามารถอื่น ๆ ของ JavaScript 34
4.4 หลักการเขียน JavaScript ใน HTML 34
4.4.1 การใช้คำสั่ง <SCRIPT>...</SCRIPT> 34
4.4.2 การระบุภาษาและรุ่นของสคริปต์ 35
4.4.3 JavaScript กับ Internet Explorer 36
4.4.4 ตำแหน่งของ SCRIPT 37
4.5 เว็บเบราเซอร์ที่ไม่สนับสนุน JavaScript 41
4.6 หมายเหตุในจาวาสคริปต์ 41
บทที่ 5 การสร้าง และทำงานกับไฟล์ฐานข้อมูล 42
5.1 การสร้างตารางและฟิลด์ 42
5.2 การเข้าสู่มุมมอง เทเบิล-ดีไซนด์ 42
5.3 ส่วนประกอบต่าง ๆ ในมุมมอง เทเบิล-ดีไซนด์ 43
5.4 การการสร้าง และกำหนดคุณสมบัติพื้นฐานของฟิลด์ 44
5. 5 ขั้นตอนการสร้างฟิลด์แบบพื้นฐาน 45
5.6 การกำหนดคีย์หลัก (Primary Key) ของตาราง 46
4.7 การใส่ข้อมูลในตาราง 47
5.8 การทำงานกับเรคอร์ดในมุมมอง Table Datasheet 47
5.8.1 ส่วนประกอบต่าง ๆ ในมุมมอง Table Datasheet 47
5.8.2 การเพิ่มเรคอร์ดลงในตาราง 48
5.8.3 การแก้ไข เลือกลบ และเพิ่มเรคอร์ดลงในตาราง 49
บทที่ 6 การออกแบบโปรแกรมระบบ Web Stat 52
6.1 Use Case Diagram 52
6.2 Sequence Diagram 52
6.2.1 Register Sequnce Diagram 52
6.2.2 Statics Sequence Diagram 53
6.2.3 Login Admin Sequence Diagram 53
6.2.4 Search Sequence Diagram 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

6.2.5 Insert Sequence Diagram 54
6.2.6 Delete Sequence Diagram 55
6.2.7 Update Sequence Diagram 55
6.2.8 Check Website Diagram 56
6.2.9 เก็บข้อมูล Sequence Diagram 56
6.3 Class Diagram 57
6.3.1 Class Diagram การเก็บข้อมูล 57
6.3.2 Class Diagram การลงทะเบียน 58
6.3.3 Class Diagram การดูสถิติ 59
6.3.4 Class Diagram การดูแลระบบ 60
6.4 Stat Diagram 61
6.4.1 Register State Diagram 61
6.4.2 Login State Diagram 61
6.4.3 SendMail State Diagram 62
6.4.4 Check Website State Diagram 62
6.4.5 Webpage State Diagram 62
6.4.6 ReceiveDate State Diagram 63
6.4.7 Www Admin_Tools State Diagram 63
6.4.8 Www Register State Diagram 63
6.4.9 WwwResultRegister State Diagram 64
6.4.10 WwwStatics State Diagram 64
6.4.11 WwwInsertData State Diagram 64
6.4.12 Wwwlogin State Diagram 65
6.4.13 WwwIndex State Diagram 65
6.4.14 WwwDeleteData State Diagram 65
6.4.15 DeleteData State Diagram 66
6.4.16 WwwResultDelete State Diagram 66
6.4.17 WwwUpdateData State Diagram 66
6.4.17 UpdateData State Diagram 67
6.4.18 WwwResultUpdate State Diagram 67
6.4.19 WwwSearchData State Diagram 67
6.4.20 SearchData State Diagram 68
6.4.21 WwwResultSearch State Diagram 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

6.4.22 PersonalDatabase State Diagram 68
6.4.23 DataDatabase State Diagram 69
6.4.24 FileUrl State Diagram 69
6.5 ออกแบบฐานข้อมูลโดยใช้ ER Diagram 70
บทที่ 7 สรุปและวิจารณ์ 72
7.1 ภาพรวม 72
7.2 แนวทางในการพัฒนาต่อ 73
ภาคผนวก 74
ก. Dowload Java Web Server and Documentation 74
ข. คู่มือการติดตั้ง Java Web Server บน WindowsNT 74
ค. Administration Tool: Network Setup 78
ง. Administration Tool: Configuring Servlets 79
บรรณานุกรม 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้าที่
4-1 เปรียบเทียบคุณสมบัติของ JavaScript กับ Java	32
5-1 แสดงรายละเอียดของชนิดข้อมูล (Data Type) ที่มีในไมโครซอฟท์แอคเซส 97	44
5-2 แสดงรายละเอียดคุณสมบัติ Field Size ของ Field ที่มีชนิดข้อมูลเป็น Number	45
7-1 เปรียบเทียบข้อดี-ข้อเสียของภาษาจาวาสคริปต์	72
7-2 เปรียบเทียบข้อดี-ข้อเสียของจาวาเซิร์ฟเล็ท	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้าที่
2-1 การประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน	8
2-2 การประมวลผลของโปรแกรมภาษาจาวา	8
3-1 Servlet Engine and its Servlets	14
3-2 การโหลด Servlet ต่าง ๆ ของ SimpleServletEngine	17
3-3 HelloWorld Servle	24
3-4 HelloWorld2 Servlet	25
3-5 HelloWorld3 Servlet	27
5-1แสดงการเข้าสู่ เทเบิล-ดีไซนด์	42
5-2 แสดงหน้าต่างเทเบิล-ดีไซนด์	43
5-3 ส่วนประกอบของมุมมองเทเบิล-ดีไซนด์	43
5-4 แสดงขั้นตอนการสร้างฟิลด์	46
5-5 แสดงขั้นตอนการสร้างคีย์หลัก	46
5-6 แสดงการใส่ข้อมูลในตาราง	47
5-7 แสดงการเข้าสู่มุมมอง Table Datasheet	48
5-8 แสดงการเพิ่มเรคอร์ดลงในตาราง	49
5-9 แสดงการแก้ไขข้อมูล	50
5-10 แสดงการลบระเบียน	50
5-11 แสดงการเพิ่มเรคอร์ด	51
6-1 Use Case Diagram	52
6-2-1 Register Sequnce Diagram	52
6-2-2 Statics Sequence Diagram	53
6-2-3 Login Admin Sequence Diagram	53
6-2-4 Search Sequence Diagram	54
6-2-5 Insert Sequence Diagram	54
6-2-6 Delete Sequence Diagram	55
6-2-7 Update Sequence Diagram	55
6-2-8 Check Website Diagram	56
6-2-9 เก็บข้อมูล Sequence Diagra	56
6-3-1 Class Diagram การเก็บข้อมูล	57
6-3-2 Class Diagram การลงทะเบียน	58
6-3-3 Class Diagram การดูสถิติ	59
6-3-4 Class Diagram การดูแลระบบ	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่	หน้าที่
6-4-1 Register State Diagram	61
6-4-2 Login State Diagram	61
6-4-3 SendMail State Diagram	62
6-4-4 Check Website State Diagram	62
6-4-5 Webpage State Diagram	62
6-4-6 ReceiveDate State Diagram	63
6-4-7 Www Admin_Tools State Diagram	63
6-4-8 Www Register State Diagram	63
6-4-9 WwwResultRegister State Diagram	64
6-4-10 WwwStatics State Diagram	64
6-4-11 WwwInsertData State Diagram	64
6-4-12 Wwwlogin State Diagram	65
6-4.13 WwwIndex State Diagram	65
6-4-14 WwwDeleteData State Diagram	65
6-4-15 DeleteData State Diagram	66
6-4-16 WwwResultDelete State Diagram	66
6-4-17 WwwUpdateData State Diagram	66
6-4-17 UpdateData State Diagram	67
6-4-18 WwwResultUpdate State Diagram	67
6-4-19 WwwSearchData State Diagram	67
6-4-20 SearchData State Diagram	68
6-4-21 WwwResultSearch State Diagram	68
6-4-22 PersonalDatabase State Diagram	68
6-4-23 DataDatabase State Diagram	69
6-4-24 FileUrl State Diagram	69
6-5 ออกแบบฐานข้อมูลโดยใช้ ER Diagram	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบันโลกของอินเทอร์เน็ตได้ติบโตขึ้นอย่างรวดเร็วมีเว็บไซต์เกิดใหม่เปิดตัวสู่สายตาชาวโลกมากมายไม่เว้นชั่วโมง บางเว็บไซต์ได้รับความนิยมอย่างล้นหลาม ในขณะที่อีกหลายเว็บไซต์กลับไม่ได้รับความนิยมนักในขณะที่ทุ่มทุนสร้างอย่างมหาศาลนั่นหมายถึงตัวเงินอันมีค่า กำไรหรือขาดทุน ขึ้นอยู่กับความพึงพอใจของผู้เยี่ยมชมเว็บไซต์เป็นหลักหากเว็บไซต์ใดสามารถสร้างความพึงพอใจให้กับผู้เยี่ยมชมเว็บไซต์ได้แล้ว นั่นหมายถึงการเข้ามาเยี่ยมชมครั้งต่อๆไป และการบอกต่อคนสนิท ซึ่งจะทำให้เว็บไซต์เหล่านั้นได้รับความนิยมและประสบความสำเร็จได้

มีเหตุผลมากมายที่จะนำมาอธิบายได้ว่า จะทำอย่างไรให้เว็บไซต์ได้รับความนิยมเพิ่มขึ้น ซึ่งพอจะสรุปได้ง่ายๆว่าต้องทำให้ผู้เข้าเยี่ยมชมได้รับความพึงพอใจนั่นเอง ความจริงก็คือเราไม่สามารถทำให้ผู้เยี่ยมชมทุกคนพึงพอใจได้ หากแต่เราสามารถทำให้ผู้เข้าเยี่ยมชมส่วนใหญ่พึงพอใจได้

การที่จะทำให้ผู้เข้าเยี่ยมชมส่วนมากพึงพอใจได้นั้นเราควรทราบข้อมูลว่า

- เขาเหล่านั้นคือใคร
- มาจากที่ไหนบ้าง
- ใช้ภาษาอะไร
- เพศอะไร
- อาชีพอะไร
- ชอบและไม่ชอบส่วนไหนในเว็บไซต์
- ฯลฯ

จะเป็นการดีมากหากผู้บริหารเว็บไซต์ได้ทราบข้อมูลต่างๆเหล่านี้ นั่นหมายถึงผู้บริหารเว็บไซต์สามารถที่จะรู้ได้ว่าผู้เยี่ยมชมส่วนใหญ่ต้องการอะไร เพื่อใช้ช่วยตัดสินใจในการวางนโยบายสำหรับพัฒนาเว็บไซต์ให้เป็นไปในทางที่ผู้เยี่ยมชมส่วนใหญ่น่าจะพึงพอใจเพิ่มขึ้นได้

เป็นเรื่องยากมากที่เราจะได้ข้อมูลทั้งหมดของผู้เยี่ยมชมเว็บไซต์ได้ นอกเสียจากผู้เยี่ยมชมทุกคนทุกคนจะยินดีตอบแบบสอบถามให้ ซึ่งในความเป็นจริงแล้วไม่มีทางเกิดขึ้นได้ ความจริงที่เป็นไปได้ก็คือเราสามารถเก็บข้อมูลที่เกิดจากการสื่อสารระหว่างเว็บเบราเซอร์กับเว็บเพจได้โดยไม่ต้องไปรบกวนผู้เข้าเยี่ยมชมเว็บไซต์แต่อย่างใด โดยที่ข้อมูลเหล่านี้ควรจะต้องถูกจัดเก็บและสรุปผลออกมาอย่างอัตโนมัติที่สุดเพื่อเป็นการแบ่งเบาภาระของผู้บริหารเว็บไซต์

งานวิจัยนี้ได้พัฒนาระบบให้บริการจัดเก็บสถิติและแสดงผลสำหรับเว็บเพจแบบออนไลน์ เพื่อใช้ในการจัดเก็บข้อมูลของผู้เข้าเยี่ยมชมเว็บเพจของผู้รับบริการ โดยข้อมูลที่สามารถจัดเก็บมาทำสถิติได้นั้นจะช่วยให้ผู้บริหารเว็บไซต์สามารถนำมาใช้ในการช่วยตัดสินใจในการวางนโยบายสำหรับพัฒนาเว็บไซต์ในระดับเบื้องต้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 ศึกษาการพัฒนาระบบให้บริการจัดเก็บสถิติและแสดงผลสำหรับเว็บเพจแบบออนไลน์

1.2.2 ศึกษาการพัฒนาโปรแกรมประยุกต์แบบไคล์เอ็น-เซิร์ฟเวอร์ โดยใช้โปรแกรมภาษาจาวาในการพัฒนา

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะสร้างระบบให้บริการจัดเก็บสถิติและแสดงผลสำหรับเว็บเพจแบบออนไลน์โดยระบบที่ทำการพัฒนาประกอบด้วย 2 ส่วนหลักๆคือ

1.เว็บเซิร์ฟเวอร์ของผู้ให้บริการจัดเก็บสถิติ พัฒนาโดยใช้โปรแกรมภาษาจาวา

2.เว็บเพจของผู้รับบริการที่นำโค้ด JavaScript (จาวาสคริปต์) ของผู้ให้บริการไปแปะไว้ สำหรับทำการส่งข้อมูลกลับมาที่เซิร์ฟเวอร์ของผู้ให้บริการ

ข้อมูลที่ระบบที่สามารถจัดเก็บได้จะขึ้นอยู่กับขอบเขตของเครื่องมือที่นำมาใช้ในการรับข้อมูลจากเว็บเบราเซอร์ของผู้เข้าเยี่ยมชมเว็บเพจ โดยงานวิจัยนี้ได้เลือกใช้ภาษาจาวาสคริปต์เป็นเครื่องมือ

ข้อมูลจากเว็บเบราเซอร์ของผู้เข้าเยี่ยมชมเว็บเพจ ที่สามารถจัดเก็บได้มีดังนี้

- วันเดือนปีและเวลาที่เข้าเยี่ยมชมเว็บเพจ
- ชื่อเบราเซอร์ที่ใช้ในการเข้าเยี่ยมชม
- เวอร์ชันเบราเซอร์
- ระบบปฏิบัติการที่ใช้และแพล็ตฟอร์มที่ใช้
- ค่าความละเอียดจอภาพขณะเข้าเยี่ยมชม

จากข้อมูลที่ได้ทั้งหมดโปรแกรมที่ผั่งเซิร์ฟเวอร์ที่พัฒนาขึ้นโดยใช้ภาษาจาวาเซิร์ฟเล็ทนั้นสามารถจะนำมาสรุปเป็นรายงานเหล่านี้ได้

- Total of Visitors แสดงจำนวนผู้เข้าเยี่ยมชมทั้งหมด
- Dates แสดงสถิติจำนวนผู้เยี่ยมชมในแต่ละวัน
- Times แสดงสถิติจำนวนผู้เยี่มชมในวันปัจจุบันในแต่ละชั่วโมง
- Days แสดงสถิติจำนวนผู้เยี่มชมในแต่ละวันทั้ง 7 (อาทิตย์ถึงจันทร์)
- Months แสดงสถิติจำนวนผู้เยี่ยมชมในแต่ละเดือน
- Years แสดงสถิติจำนวนผู้เยี่ยมชมในแต่ละปี
- Operating Systems แสดงสถิติระบบปฏิบัติการของผู้เข้าชม
- Referrers แสดงสถิติเว็บไซต์ที่นำพามาเยี่ยมชม (เว็บลิงค์)
- Browsers แสดงสถิติเบราเซอร์ของผู้เยี่ยมชม
- Versions แสดงสถิติเวอร์ชันของเบราเซอร์
- Platforms แสดงสถิตของแพลตฟอร์มของเครื่องผู้เข้าเยี่ยมชม
- Screens แสดงสถิติของค่าความละเอียดจอภาพขณะเข้าเยี่ยมชม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีต่างๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลักๆ อยู่ ดังนี้คือ การโปรแกรมภาษา Java, Java Servlet API, JavaScript, การดึงค่าจากเบราเซอร์ของผู้เยี่ยมชมเว็บไซต์ผ่านจาวาคริปต์, การส่งข้อมูลจากจาวาสคริปต์ไปให้เซิร์ฟเวอร์, การใช้งาน Java Web Server V.2 ร่วมกับ Java Servlet API และ การเก็บข้อมูลลงใน MS Access Database ผ่าน ODBC โดยใช้ SQL command ที่สร้างจาก Java Servlet API ทฤษฎีเบื้องต้นในบางส่วนอยู่ในบทที่ 2 ถึงบทที่ 5 นี้แล้ว

ภายหลังศึกษาทฤษฎีต่างๆ ก็คือการออกแบบระบบออกมาเพื่อให้เป็นไปตามความต้องการคือให้ได้ระบบให้บริการจัดเก็บสถิติและแสดงผลสำหรับเว็บเพจแบบออนไลน์ขึ้นมานั่นเอง โดยได้อาศัยโมเดล UML ในการพัฒนาทั้งนี้เพราะ UML เป็นภาษาสากลที่อธิบายด้วยรูปและแผนภาพต่างๆที่เข้าใจได้ง่าย ช่วยให้การออกแบบหาจุดบกพร่องได้และมองเห็นภาพรวมของระบบได้ง่าย ดังปรากฏอยู่ในบทที่ 6

ท้ายที่สุดคือการนำเอาโมเดลที่ได้จากการออกแบบทั้งหมดมาใช้สร้างพัฒนาระบบและทดสอบปรับปรุงแก้ไขหาจุดบกพร่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ภาษาจาวา

ภาษาจาวา หรือ Java Programming Language ถูกพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์ เริ่มแรกซันได้คิดค้นภาษาจาวาขึ้นมาเพื่อใช้กับระบบปฏิบัติการ Set-top box คือโครงการของระบบตลาดอิเล็กทรอนิกส์ ที่ต้องการให้ระบบสามารถทำการควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้โดยไม่ขึ้นอยู่กับแพล็ตฟอร์ม (Platform) ที่ใช้คือการพัฒนาระบบปฏิบัติการหรือซอฟต์แวร์ให้สามารถทำงานได้โดยไม่สนใจว่าไมโครโพรเซสเซอร์ที่ใช้อยู่จะเป็นรุ่นใด ยี่ห้อไหน ฮาร์ดแวร์เป็นของอะไรทั้งยังได้รวมความสามารถไปถึงการเชื่อมต่อการเปลี่ยนข้อมูลข่าวสารทั่ว ๆ ไป และทางด้านธุรกิจผ่านทางระบบเครือข่ายของคอมพิวเตอร์อีกด้วย

ทางทีมงานซึ่งพัฒนาโปรแกรมนี้ได้เล็งเห็นถึงการเจริญเติบโตและความสามารถในการแลกเปลี่ยนข่าวสารบนอินเตอร์เน็ตจึงได้วางแผนและพัฒนาระบบอินเทอร์เฟซแบบใหม่เพื่อใช้กับอินเตอร์เน็ตขึ้น ด้วยสาเหตุนี้เองภาษาจาวาจึงได้บังเกิดขึ้น รวมทั้งยังได้มีการเปิดตัวเว็บบราวเซอร์ชื่อว่า Hotjava เพื่อใช้สำหรับแสดงความสามารถของภาษาจาวา

2.1 ความสามารถของภาษาจาวา

ลักษณะของภาษาจาวา เป็นการเขียนโปรแกรมอ้างอิงเชิงวัตถุ OOP (Object Oriented Programming) สามารถสรุปความสามารถการทำงานได้ดังต่อไปนี้

1. การทำงานที่ไม่ขึ้นอยู่กับแพล็ตฟอร์ม โดยใช้วิธีการแปลงข้อมูลแบบไบต์โค้ด
2. การใช้เมธอดจัดการกับข้อมูลแทนการใช้ฟังก์ชัน
3. ความสามารถในการโต้ตอบแบบอินเตอร์แอ็กทีฟ กับงานด้านมัลติมีเดีย
4. ความสามารถในการดึงเอาคลาสไลบรารีจากที่ต่าง ๆ ผ่านทาง HTTP และ FTP
5. ความสามารถในการเล่นไฟล์เสียงไฟล์วิดีโอและ ไฟล์ภาพกราฟิกเคลื่อนไหวการ
6. โต้ตอบแบบเรียลไทม์
7. เป็นระบบมัลติเธรดคือสามารถทำงานได้หลายงานพร้อมกันในเวลาเดียว
8. ไม่มีตัวแปรประเภทพอยน์เตอร์จึงหมดปัญหาเกี่ยวกับการขอใช้หน่วยความจำโดยตรง
9. มีความปลอดภัยสูง ด้วยความสามารถที่ไม่ให้ผู้ใช้เข้าไปยุ่งเกี่ยวกับหน่วยความจำของระบบ จึงปลอดภัยจากไวรัส
10. การดาวน์โหลดซอฟต์แวร์ใหม่ ๆ และอัพเกรดเวอร์ชันใหม่ได้อย่างง่ายดายโดยผ่านทางเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การนำจาวาไปประยุกต์สร้างและพัฒนาในงานด้านต่าง ๆ

จากความสามารถดังที่กล่าวผ่านมาข้างต้นคงพอจะเป็นข้อสรุปได้แล้วว่าจาวาเป็นภาษาโปรแกรมที่มีความสามารถรอบด้าน อีกทั้งยังมีความยืดหยุ่นภายในตัวสูง การที่นำมันมาสร้างและพัฒนางานดี ๆ มีประสิทธิภาพคงจะทำได้ไม่ยากนัก ซึ่งเราสามารถใช้จาวาสร้างและพัฒนาแอพพลิเคชันต่าง ๆ โดยยกตัวอย่างพอสังเขปได้ดังนี้

ไคลเอ็นต์

จาวาสามารถเป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้านไคลเอ็นต์ได้เพียงพอโดยเฉพาะด้านกราฟิกและระบบติดต่อกับผู้ใช้ เพราะปัจจุบันโปรแกรมที่เขียนขึ้นจากภาษาจาวาจะทำงานที่ฝั่งไคลเอ็นต์เป็นส่วนมากจะเห็นได้จากจาวาแอพเพล็ตต่าง ๆ บนเวบบราวเซอร์ แต่จะให้ดีและสมบูรณ์ยิ่งขึ้นต่อความเป็นไคลเอ็นต์ซอฟต์แวร์ที่ดีก็คือ การติดต่อระบบฐานข้อมูลจากไคลเอ็นต์ไปเป็นเซิร์ฟเวอร์ แม้ว่าในปัจจุบันจาวาจะสามารถทำได้แล้วก็ตามแต่ก็ต้องใช้เครื่องมือชนิดอื่นเป็นตัวช่วย เช่น การเชื่อมต่อกับฐานข้อมูลแบบ ODBC (Object-Oriented Database Connectivity) ซึ่งการทำงานของทางฝั่งเซิร์ฟเวอร์ต้องใช้ภาษาคอมพิวเตอร์ชนิดอื่นช่วยเพื่อการติดต่อกับฐานข้อมูล

เซิร์ฟเวอร์

จาวาเมื่อใช้สร้างโปรแกรมประยุกต์บนเซิร์ฟเวอร์นั้นจาวาจำเป็นที่จะต้องได้รับการปรับปรุงการทำงานอยู่สามอย่างคือ การเชื่อมต่อกับฐานข้อมูลโดยตรง การทำงานให้สอดคล้องกับอินพุต เอาต์พุต และความเร็วของการรันต้องเพิ่มขึ้นเท่า ๆ กับโปรแกรมประยุกต์ทั่วไป

ระบบ JDBC (Java Database Connectivity) ได้เข้ามาแก้ปัญหาในส่วนของการติดต่อกับฐานข้อมูลโดยตรง โดยการร่วมมือกันของบริษัทซอฟต์แวร์ที่มีชื่อเสียงหลายบริษัท ได้สร้างมาตรฐานการเชื่อมต่อกับฐานข้อมูลของจาวาขึ้น แต่ในขณะนี้ก็ยังไม่เป็นที่รู้จักแพร่หลายนักสำหรับ JDBC คาดว่าในอนาคตอันใกล้นี้การเชื่อมต่อฐานข้อมูลจะทำได้ง่ายขึ้นจากไคลเอ็นต์สู่เซิร์ฟเวอร์โดยใช้จาวา

Security

จาวาได้สร้างสภาพแวดล้อมเสมือน "Java Virtual Machine" ซึ่งจะอนุญาตให้เฉพาะจาวาแอพเพล็ต หรือจาวาแอพพลิเคชันซึ่งเป็นไบต์โค้ด เท่านั้นที่สามารถรันได้โดยที่ไบต์โค้ดจะปราศจากไวรัสหรือส่วนที่จะทำอันตรายต่อระบบ อันเนื่องจากสภาพแวดล้อมเสมือนจะจำกัดสิทธิ์การเข้าใช้ทรัพยากรของ ไบต์โค้ดในอนาคตการรักษาความปลอดภัยอีกระบบหนึ่งก็กำลังถูกนำมาปรับใช้คือ "การเข้ารหัสข้อมูล" เพื่อเพิ่มความปลอดภัยของข้อมูล

การใช้จาวาประยุกต์สร้างและพัฒนางานยังสามารถแบ่งเป็นประเภทตามลักษณะของกลุ่มงานได้ดังนี้

- งานด้านการศึกษา โดยจัดทำเป็นลักษณะสั่งการสอนคล้ายกับ CAI มีความสามารถในเชิงโต้ตอบระหว่างผู้ใช้งานกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตอบสนองงานทางด้านเว็บเพจทำให้งานสร้างเว็บเพจดูมีชีวิตชีวาขึ้น โดยการนำเอาความสามารถทางด้านมัลติมีเดียมาใช้อย่างเต็มที่
- สร้างสรรค์เครื่องมือพัฒนา (Software Developer tool kits)
- สร้างแอพพลิเคชันทางธุรกิจ มีความสามารถในการประมวลผลข้อมูลด้วยตัวเอง
- สร้างตัวจัดการบนอินเตอร์เน็ต มีความสามารถในการเรียกหาข้อมูลผ่านทางระบบเครือข่าย
- พัฒนาเกม จะมีความเป็นมัลติ คือสามารถเล่นได้หลาย ๆ คนพร้อมกันผ่านทางระบบเครือข่ายหลากหลายพื้นที่ไม่ว่าจะเล่นด้วยกัน ณ ตำแหน่งใด ทั่วทุกมุมโลก

2.3 สภาพแวดล้อมเสมือน (Java Virtual Machine)

สภาพแวดล้อมเสมือนเป็นหลักการสร้างคอมพิวเตอร์จำลองของจาวา โดยการสมมติให้มีคอมพิวเตอร์อีกเครื่องหนึ่งขึ้นมาโดยเครื่องนี้จะใช้ในการคอมไพล์โปรแกรมภาษาจาวาทุกโปรแกรม เมื่อต้องการให้โปรแกรมภาษาจาวาไปทำงานบนคอมพิวเตอร์จริง ๆ เครื่องใดเราก็เพียงแต่สร้างตัวอินเทอร์พรีเตอร์ของคอมพิวเตอร์จำลองตัวนี้บนเครื่องนั้น ๆ ภาษาจาวาทุกโปรแกรมก็จะสามารถทำงานบนระบบคอมพิวเตอร์นั้น ๆ ได้ตามต้องการด้วยหลักการนี้ก็เป็นที่มาของคุณสมบัติของจาวาที่ไม่ขึ้นกับแพล็ตฟอร์มและฮาร์ดแวร์ใด ๆ หรือพอร์เทเบิล (Portable) เราอาจจะเรียก Java Virtual Machine สั้น ๆ ว่า JVM ก็ได้

สำหรับเหตุผลที่ต้องมี JVM นี้จริง ๆ แล้วเป็นการกำหนดคำขึ้นมา เพื่อเป็นคำจำกัดความเฉพาะในความคิดเท่านั้น เพื่อให้นักพัฒนาจะได้ไม่ถูกบังคับให้ต้องสร้างตัวอินเทอร์พรีเตอร์ตามแนวทางแบบใดแบบหนึ่งโดยเฉพาะแต่ตัวอินเทอร์พรีเตอร์ที่สร้างขึ้นตามข้อกำหนดดังกล่าวไม่ว่าจะอยู่บนแพล็ตฟอร์มใดจะสามารถรันโปรแกรมที่เขียนขึ้นในภาษาจาวาได้ โดยให้ผลลัพธ์ออกมาเหมือนกัน

แต่ในข้อกำหนดของ JVM ก็มีการให้นิยามที่ชัดเจนมาก ๆ เกี่ยวกับการออกแบบอินเทอร์พรีเตอร์ในหลาย ๆ ส่วนโดยเฉพาะอย่างยิ่งส่วนที่เกี่ยวข้องกับการกระจายโค้ดของจาวา ไปใส่ไว้ในรูปแบบที่กำหนด ข้อกำหนดนี้ได้แก่ไวยากรณ์ของออบโค้ดและโอเปอร์แรนด์ พร้อมกับค่าประจำตัว การจัดโครงสร้างของโค้ด การจัดวางรูปแบบของออบเจ็กต์ของจาวา ข้อกำหนดต่าง ๆ เหล่านี้จะทำให้นักพัฒนาอินเทอร์พรีเตอร์ทั้งหลาย สามารถจะสร้างอินเทอร์พรีเตอร์ขึ้นมาใหม่บนแพล็ตฟอร์มใด ๆ ก็ได้ ดังนั้นการพัฒนาภาษาจาวาจึงไม่ได้ถูกปิดกั้นอยู่กับระบบของซันซึ่งเป็นผู้คิดค้นภาษาจาวานี้ขึ้นมาเท่านั้น

จากที่กล่าวมาข้างต้นถึงคุณสมบัติของจาวาที่ไม่ยึดติดอยู่กับแพล็ตฟอร์มใด ๆ ทำให้การทำงานกับระบบคอมพิวเตอร์ที่ทำงานแบบกระจาย (Distributed Computing) ได้รับการตอบสนองอย่างเหมาะสม นั่นคือ ทำให้เกิดความพอร์เทเบิลนั่นเอง

อย่างไรก็ตาม JVM ก็ยังมีข้อจำกัดอยู่ ข้อจำกัดของ JVM นั้นจะอยู่ที่ข้อจำกัดของการออกแบบตัวอินเทอร์พรีเตอร์แทน เช่น การจำกัดค่าของโอเปอร์แรนด์ และขนาดของสแต็ก เป็นต้น ข้อกำหนดเหล่านี้หมายถึงว่า JVM สามารถจะอ้างถึงหน่วยความจำได้เฉพาะในห้วงแอดเดรสเท่าที่มีอยู่เท่านั้น

ข้อจำกัดภายในของตัว JVM เองนั้นมีห้วงแอดเดรสให้ใช้อยู่ถึง 4 GB เพราะขนาดของความกว้างของการอ้างแอดเดรสเป็น 32 บิตเมธอดต่าง ๆ ของจาวามีขนาดได้เพียง 32 KB เพราะมีข้อจำกัดของการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งกระโดด เป็นแบบ 16 บิต โดยมีบิตแรกเป็นตัวบอกว่ากระโดดไปข้างหน้า หรือข้างหลัง และบิตต่อ ๆ ไป บอกระยะทางการกระโดดจากจุดเหล่านี้มีขนาดเพียง 8 บิต นอกจากนั้นจำนวนของค่าคงที่ที่อยู่ในส่วนกลาง ยังถูกจำกัดด้วยขนาดของดัชนี 16 บิต ทำให้มีจำนวนได้เพียง 32,000 ค่าในแต่ละเมธอด

การที่เราถือว่าการมีค่าเหล่านี้เป็นข้อจำกัด ก็อาจจะเป็นการมองการณ์ไกลไปสักหน่อย เนื่องจากปัจจุบันเครื่องคอมพิวเตอร์ส่วนใหญ่มีหน่วยความจำกัดเพียง 16 หรือ 32 MB กันเท่านั้น หน่วยความจำขนาด 4 GB จึงยังเป็นเรื่องที่ไม่ต้องคิดหนักในตอนนี้ ในขณะที่ขนาดของเมธอดจำกัดที่ 32 KB ก็เป็นเพียงเมธอดเดียวเท่านั้น

2.4 หลักการทำงานของภาษาจาวา

ในที่นี้ขออธิบายการทำงานของภาษาจาวาเริ่มตั้งแต่การคอมไพล์ตัวโปรแกรมจนกระทั่งเราเรียกใช้งานในเว็บเพจ ดังนี้

2.4.1 การคอมไพล์

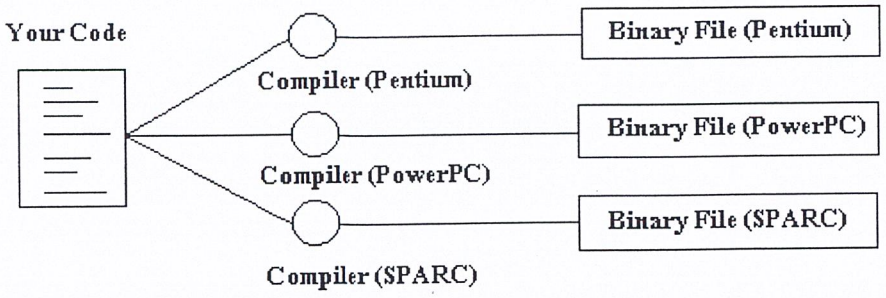
คอมไพเลอร์ของภาษาจาวาก็เช่นเดียวกับคอมไพเลอร์ในภาษาอื่น ๆ นั่นคือมันจะสร้างรหัสภาษาเครื่อง (Machine Code หรือ Assembler Code) จากภาษาในระดับที่สูงกว่าเพื่อให้ซีพียู (CPU : Central Processing Unit) สามารถนำไปใช้งานได้

แต่ข้อแตกต่างที่สำคัญระหว่างคอมไพเลอร์ของภาษาจาวากับภาษาอื่น ๆ คือซีพียูหรือโปรเซสเซอร์ที่จะทำหน้าที่ในการปฏิบัติตามคำสั่งที่ได้จากการคอมไพล์ภาษาจาวานั้นไม่มีอยู่จริง เป็นเพียงสิ่งที่สมมติขึ้นมาที่เรียกว่า Java Virtual Machine นอกจากนี้การอ้างถึงส่วนต่าง ๆ ของโปรแกรมที่คอมไพล์ด้วยคอมไพเลอร์ของภาษาจาวาก็จะมีวิธีการที่แตกต่างออกไป

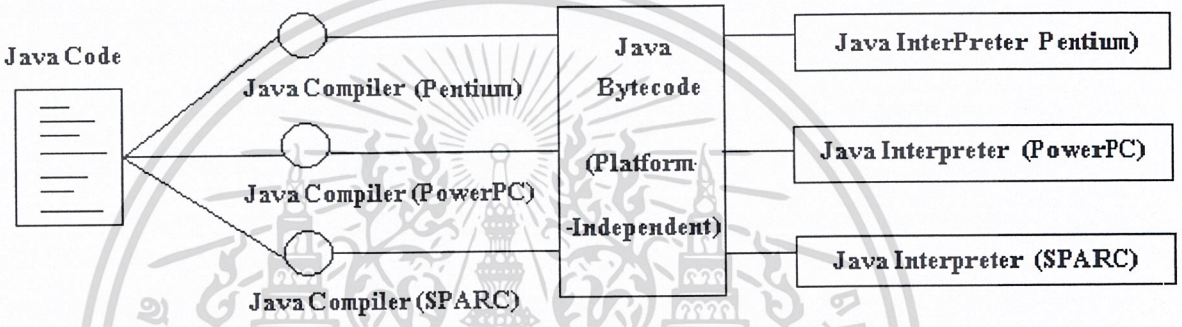
คอมไพเลอร์ของภาษาจาวาจะไม่เปลี่ยนการอ้างถึงส่วนของโปรแกรมจากการใช้ชื่อแบบในภาษาสูง ไปเป็นตัวเลขเหมือนคอมไพเลอร์ภาษาอื่น ๆ ทำกันและคอมไพเลอร์ภาษาจาวาก็จะไม่มีการสร้างแผนที่ของการจัดวางโปรแกรมบนหน่วยความจำขึ้นมาในระหว่างการคอมไพล์ด้วยเหตุผลที่สำคัญคือเพื่อเป็นการสร้างความพอร์เทเบิลให้กับตัวโปรแกรมเพราะการจัดวางตำแหน่งของโปรแกรมจะต้องขึ้นอยู่กับลักษณะการทำงานของโปรเซสเซอร์ตัวใดตัวหนึ่ง การยังไม่จัดวางตำแหน่งช่วยทำให้โปรแกรมที่ได้จากการคอมไพล์มีความเป็นกลาง สามารถนำไปใช้บนคอมพิวเตอร์แพล็ตฟอร์มอื่น ๆ ได้นอกจากนั้นยังทำให้เกิดความปลอดภัยอีกด้วย

สิ่งที่ได้จากการคอมไพล์ในภาษาจาวาเราเรียกว่า ไบต์โค้ด (Byte Code)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 การประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน



รูปที่ 2-2 การประมวลผลของโปรแกรมภาษาจาวา

2.4.2 การวางตำแหน่งในหน่วยความจำ

ในภาษาจาวาจะไม่มีการลดรูปแบบการอ้างถึงส่วนต่าง ๆ ของโปรแกรมจากการเรียกเป็นชื่อให้เหลือเพียงตัวเลขหรือแอดเดรสที่กำหนดขึ้นจากการจัดวางตำแหน่งของโปรแกรมลงในหน่วยความจำ คอมไพเลอร์ภาษาจาวาจะทิ้งชื่อของแต่ละส่วนของโปรแกรม (โดยเฉพาะเมธอด) เอาไว้ในตัวโปรแกรมที่สร้างขึ้นเมื่อโปรแกรมทำงาน จะเป็นหน้าที่ของตัว อินเทอร์พรีเตอร์ที่จะคอยเปิดตารางค้นหาที่อยู่ของเมธอดที่ต้องการเรียกใช้งาน โดยก่อนที่จะเริ่มทำงานจริง อินเทอร์พรีเตอร์จะต้องสร้างแผนที่ในการจัดวางสิ่งต่าง ๆ ลงในหน่วยความจำขึ้นมาก่อนแล้วจึงสร้างตารางขึ้นมาเพื่อช่วยหาตำแหน่งของเมธอดเมื่อมีการเรียกใช้งานโดยใช้ชื่อของเมธอด

2.4.3 การรันโปรแกรม

การรันโค้ดที่คอมไพล์เอาไว้สำหรับ Java Virtual Machine เป็นหน้าที่ของตัวอินเทอร์พรีเตอร์ การรันโปรแกรมจะแบ่งได้เป็น 3 ขั้นตอนหลัก ๆ คือ การอ่าน การตรวจสอบความถูกต้อง และการทำตามโค้ด หน้าที่ในการอ่านโค้ดเข้าสู่ระบบจะเป็นของ Class Loader หน้าที่การทำงานในส่วนนี้จะไม่ได้อ่านเข้ามาเฉพาะ ๆ ไฟล์จาวาที่กำลังจะเรียกใช้เท่านั้น แต่จะอ่านคลาสที่มีการอ้างถึงและคลาสที่มีการสืบทอดต่อมาโดยคลาสที่อ้างถึง เมื่อผ่านขั้นตอนนี้แล้ว โค้ดทั้งหมดก็จะถูกส่งผ่านตัวตรวจสอบไบต์โค้ดเพื่อให้แน่ใจว่าโค้ดที่ส่งมามีความถูกต้องตามมาตรฐานของจาวา และจะไม่รบกวนเสถียรภาพของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผ่านการตรวจสอบแล้วโค้ดก็จะถูกส่งต่อไปยังระบบรันใหม่ (Run-Time System) ซึ่งจะสั่งงานไปยังฮาร์ดแวร์อีกต่อหนึ่งซึ่งหลักการทำงานในแต่ละขั้นตอนที่กล่าวมาข้างต้นเป็นดังนี้

2.4.4 Class Loader

ตัวคลาสโหลดเดอร์ จะทำหน้าที่ดึงโค้ดทั้งหมดที่จำเป็นในการทำงานของแอพพลิเคชัน ไม่ว่าจะเป็นคลาสที่ถูกสืบทอดมาหรือคลาสอื่น ๆ ที่มีการเรียกใช้เมื่อโหลดเดอร์ดึงคลาสใดเข้ามาแล้วก็จะจัดคลาสนั้น ๆ ใส่เข้าไว้ใน namespace ของมันเองโดยการเก็บจะใช้ชื่อของคลาสเป็นสำคัญ ไม่ได้ใช้การอ้างถึงเป็นตัวเลขหลักการนี้ก็จะเหมือนกันกับการทำงานของ Virtual Machine ที่โอเอส (OS : Operating System) สร้างขึ้นให้แอพพลิเคชันแต่ละตัวทำงานถ้าไม่ได้มีการเจาะจงเรียกใช้คลาสที่อยู่นอก namespace นี้ การเรียกใช้ชื่อต่าง ๆ ในคลาสก็จะไม่มีการรบกวนกันระหว่างคลาสเลย

คลาสทั้งหมดที่อยู่บนเครื่องโลคอล จะได้รับห้วงแอดเดรส (Address Space) เป็นของตัวเอง ส่วนคลาสต่าง ๆ ที่ดึงมาจากภายนอกจะได้รับ namespace เป็นของตัวเองการทำงานลักษณะนี้จะช่วยให้คลาสที่อยู่บนโลคอลทำงานได้ประสิทธิภาพดีขึ้นเพราะใช้ namespace ร่วมกันได้ แต่ก็ยังมีการป้องกันความผิดพลาดที่อาจจะเกิดจากคลาสที่ดึงเข้ามาจากภายนอกและ ในทางกลับกันคลาสที่อิมพอร์ตเข้ามาก็ปลอดภัยจากความผิดพลาดที่อาจจะเกิดขึ้นจากคลาสโลคอลด้วย

เมื่อคลาสทั้งหมดที่เกี่ยวข้องกับการทำงานถูกอิมพอร์ตเข้ามาเรียบร้อยแล้ว การจัดวางหน่วยความจำสำหรับเริ่มต้นการทำงานก็จะเกิดขึ้นได้การเรียกชื่อต่าง ๆ ก็จะสามารถจับคู่กับแอดเดรสจริง ๆ ของหน่วยความจำได้ แล้วตัวโหลดเดอร์จะสร้างตารางสำหรับค้นหาที่อยู่เลี่ยงจากการทำงานผิดปกติของซูเปอร์คลาส (Super class) และการอ้างแอดเดรสที่ไม่ถูกต้องได้

2.4.5 ตรวจสอบไบต์โค้ด

เมื่อโค้ดเดินทางมาจนถึงขั้นตอนการสร้างตารางจับคู่ชื่อกับแอดเดรสแล้ว ก็ยังไม่สามารถแน่ใจได้ว่าโค้ดที่อ่านเข้ามาจะมีความปลอดภัยดังนั้นจึงต้องมีตัว Verifier หรือตัวตรวจสอบไบต์โค้ด ทำหน้าที่ตรวจสอบความถูกต้องทีละบรรทัดว่าเป็นไปตามข้อกำหนดของจาวาและ สอดคล้องกับการทำงานของตัวโปรแกรมเองหรือไม่ การตรวจสอบโค้ดในเชิงทฤษฎีจะสร้างและค้นหาปัญหาต่าง ๆ ได้หลายอย่าง เช่น จะไม่มีการสร้างพอยต์เตอร์ที่เกินกว่าหน่วยความจำจริง ไม่มีคำสั่งใดสามารถละเมิดสิทธิ์การทำงานของตัวโปรแกรมได้ ไม่มีการจับคู่ออบเจ็คผิด จะไม่มีการให้โอเปอร์แรนด์มากหรือน้อยเกินไป การกำหนดค่าต่าง ๆ สำหรับไบต์โค้ดจะต้องถูกต้องครบถ้วน และจะไม่มีการแปลงข้อมูลผิดรูปแบบ

การใช้ตัวตรวจสอบตอบสนองจุดประสงค์ 2 ประการที่สำคัญคือสิ่งต่าง ๆ ดังที่กล่าวมาแล้วจะถูกตรวจสอบก่อน ทำให้ตัวอินเทอร์พรีเตอร์มั่นใจได้ว่า ไบต์โค้ดที่ส่งเข้าไปทำงานจะไม่มีขั้นตอนการทำงานที่สร้างปัญหาให้กับตัวระบบ และจุดประสงค์ที่สองก็คือตัวอินเทอร์พรีเตอร์จะทำงานตามไบต์โค้ดได้รวดเร็วกว่า เพราะไม่ต้องคอยระวังว่าจะมีปัญหาเกิดขึ้น และไม่ต้องหยุดเป็นช่วง ๆ เมื่อพบปัญหาและต้องแก้ไข

ในการทำงานไบต์โค้ดจะถูกตรวจสอบเพียงครั้งเดียวเท่านั้น และจะทำงานไปได้ตลอดไม่ต้องมีการตรวจสอบซ้ำอีกเมื่อมีการเรียกกลับมาทำงานที่ส่วนเดิมของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6 การทำงานตามโค้ด

เมื่อตัวโหลดเดอร์ได้รวบรวมโค้ดเข้ามาสู่ระบบทำการจัดวางในหน่วยความจำ และตัวตรวจสอบได้ทำการตรวจสอบความถูกต้องแล้วโค้ดก็จะถูกส่งต่อไปยังตัวอินเทอร์พรีเตอร์เพื่อทำงานตามคำสั่ง การทำงานตามคำสั่งของโค้ดก็คือการเปลี่ยนโค้ดให้กลายเป็นคำสั่ง การทำงานจริงที่ตัวระบบไคลเอ็นต์ที่รันโค้ดนี้สามารถทำงานได้ ซึ่งวิธีการที่ทำได้ก็มีอยู่ 2 วิธีด้วยกันคือ ตัวอินเทอร์พรีเตอร์ทำการคอมไพล์โค้ดหล่านี้ให้กลายเป็นเนทีฟโค้ดที่ตัวเครื่องไคลเอ็นท์เข้าใจแล้วค่อยทำงาน เพื่อให้ได้ความเร็วสูงสุดในการทำงาน กับอีกวิธีหนึ่งก็คือตัวอินเทอร์พรีเตอร์อ่านโค้ดเข้ามาแล้วตีความทำงานไปทีละคำสั่ง และทำการตีความไปเรื่อย ๆ ตลอดเวลาที่มีการทำงาน

โดยปกติแล้วผู้สร้างตัวอินเทอร์พรีเตอร์มักจะเลือกใช้วิธีการหลัง รูปแบบของไบต์โค้ดในภาษาจาวามีความยืดหยุ่นเพียงพอที่จะสามารถเปลี่ยนไปทำงานบนเครื่องไคลเอ็นท์แบบต่าง ๆ ได้โดยไม่มีการก่อให้เกิดโอเวอร์เฮดมากมายเกินความจำเป็นอย่างไรก็ตาม ไคลเอ็นท์ของจาวาบางระบบจะมีความสามารถในการทำงานได้ทั้งสองวิธีคือโปรแกรมเมอร์สามารถจะเลือกใช้วิธีการคอมไพล์กับงานที่เน้นการคำนวนมาก ๆ เพื่อเป็นการเพิ่มสมรรถนะในการทำงานให้ได้เต็มที่ ซึ่งไคลเอ็นต์แบบนี้จะให้ได้ทั้งความพอร์เทเบิล และสมรรถนะที่ดี

การสร้างระบบรันไทม์ที่ดีจะต้องถ่วงดุลย์ความสำคัญ 3 ประการให้ได้พอเหมาะ นั่นคือความพอร์เทเบิล ความปลอดภัย และสมรรถนะเรื่องของความพอร์เทเบิลทำได้โดยการใช้รูปแบบของไบต์โค้ดที่มีความเป็นกลางเพียงพอ สามารถนำไปใช้รันบนเครื่องคอมพิวเตอร์แบบต่าง ๆ ได้โดยง่ายนอกจากนั้นการที่ตัวอินเทอร์พรีเตอร์ทำการกำหนดการจัดวางตำแหน่งในหน่วยความจำในช่วงรันไทม์ แทนที่จะเป็นระหว่างการคอมไพล์เหมือนภาษาอื่น ๆ ก็เป็นการเพิ่มความแน่นอนว่าคลาสต่าง ๆ ที่อิมพอร์ตเข้ามาจะยังคงใช้ได้อยู่ตลอดเวลา เรื่องของความปลอดภัยนั้น เป็นสิ่งที่มีการคำนึงถึงอยู่ตลอดการทำงานของระบบรันไทม์ โดยเฉพาะอย่างยิ่งในส่วนของตัวตรวจสอบไบต์โค้ด ที่ทำให้แน่ใจได้ว่าโปรแกรมจะทำงานได้ถูกต้องตามข้อกำหนดของจาวา ส่วนเรื่องของสมรรถนะนั้นก็สามารถจัดการได้ในสองระยะ คือ พยายามเอาโอเวอร์เฮดทั้งหลายไปใส่ไว้ที่ตอนเริ่มต้นโหลดโปรแกรมเข้ามาสู่ระบบ หรือไม่ก็กำหนดให้ทำงานเป็นแบบแบ็กกราวนด์ (Back-Ground Thread)

ด้วยสิ่งต่าง ๆ เหล่านี้ทำให้จาวาสามารถปล่อยสมรรถนะระดับที่น่าพอใจออกมาได้โดยที่ยังคงไว้ซึ่งความพอร์เทเบิลและสภาพแวดล้อมที่ปลอดภัยนอกจากนั้นยังสามารถจะดึงสมรรถนะระดับสูงสุดออกมาใช้ได้ทันทีเมื่อต้องการ

2.4.7 การสร้างและรันภาษาจาวา

การสร้างโปรแกรมจากจาวา ก็เหมือนกับภาษาโปรแกรมอื่น ๆ คือเขียนโค้ดโปรแกรมจากเอดิเตอร์ใด ๆ ก็ได้บันทึกอยู่ในนามสกุล .java จากนั้นจึงนำไปคอมไพล์ด้วยคอมไพเลอร์ของจาวาจะได้ไฟล์ใหม่ในนามสกุล .class ที่เก็บรหัสของการคอมไพล์ไว้จาวาเรียกรูปแบบข้อมูลที่อยู่ในไฟล์ใหม่นี้ว่าไบต์โค้ดไฟล์ที่ได้คือแอพเพล็ตนั่นเองแอพเพล็ตยังไม่สามารถรันได้ทันทีเหมือนไฟล์นามสกุล .exe หรือ .com ที่เราคุ้นเคยกัน เพราะข้อมูลแบบไบต์โค้ดจะมีรูปแบบข้อมูลที่อยู่กึ่งกลางระหว่างโค้ดโปรแกรม (Source Code) กับโค้ดที่คอมพิวเตอร์อ่านแล้วนำไปปฏิบัติงานได้ทันที (Machine Code) หากจะรันแอพเพล็ตบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบใด ๆ จะต้องใช้อินเทอร์พรีเตอร์จาวาของระบบนั้น ๆ เพื่อตรวจสอบความถูกต้องของไบต์โค้ดแล้วแปลให้เป็นรหัสภาษาเครื่องส่งให้ระบบปฏิบัติการนำไปรันต่อไปขั้นตอนการทำงาน

2.5 เว็บบราวเซอร์ที่สนับสนุนจาวา

HTML (Hypertext Markup Language) เป็นภาษาที่มีรูปแบบหนึ่งของภาษา SGML (Standard Generalized Markup Language) นิยมใช้กันทั่วไปบนอินเทอร์เน็ตเป็นภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งของคอมพิวเตอร์มีลักษณะของโปรแกรมเป็นไฟล์ตัวอักษรในมาตรฐานของรหัสแอสกีประกอบด้วย Reserve Markup Directive หรือคำสั่งต่าง ๆ ซึ่งมีลักษณะเป็นคำสั่งที่ง่ายต่อการทำความเข้าใจ เพื่อให้การสร้างเว็บเพจมีความสะดวกและง่ายมากยิ่งขึ้น ทั้งนี้เพื่อส่งผ่านโครงสร้างของข้อมูลระหว่างผู้ใช้

ในปัจจุบันภาษา HTML ได้ถูกกำหนดมาตรฐานขึ้นมาเป็นรุ่นที่ 4.0 HyperText ที่ถูกสร้างขึ้นมาจะอยู่ในรูปของแฟ้มเอกสาร HTML (Document File) ที่มีการกำหนดคุณสมบัติ (markup) ของเว็บเพจเข้าไป ยังไม่สามารถแสดงผลข้อมูลออกมาให้ใช้งานได้โดยตรง ถ้าต้องการดูผลลัพธ์ที่เกิดจากการสร้างต้องผ่านโปรแกรมที่ทำหน้าที่แปลคำสั่งนั้นก่อน เราเรียกโปรแกรมที่ทำหน้าที่นี้ว่า "โปรแกรมเว็บบราวเซอร์" (Web Browser Program) หน้าที่หลักของโปรแกรมนี้คือเป็นตัวแปลคำสั่งของ HyperText แล้วแสดงผลออกมาเป็นรูปภาพ เสียง ข่าวสารและ ข้อมูล คุณสมบัติอื่น ๆ ของโปรแกรมเว็บบราวเซอร์ ไม่ว่าจะเป็นการดาวน์โหลดไฟล์ การดึงรูปภาพมาใช้งาน การพิมพ์เอกสาร HTML ออกทางเครื่องพิมพ์ การส่งจดหมายอิเล็กทรอนิกส์และความสามารถด้านอื่น ๆ อีกมากมาย ทำให้เราสามารถใช้งาน อินเทอร์เน็ตหรืออินทราเน็ตได้อย่างสะดวกรวดเร็ว ง่าย และสนุกสนาน ซึ่งทีมพัฒนาภาษาจาวาได้วางแผนและพัฒนาระบบอินเทอร์เฟซโดยเล็งเห็นถึงการเจริญเติบโตและความสามารถของภาษาจาวาจึงได้เปิดตัวเว็บบราวเซอร์มีชื่อว่า Hotjava เพื่อใช้สำหรับแสดงศักยภาพของจาวาที่มีอยู่อย่างเต็มที่ราวปี 2538 เพราะว่าในขณะนั้นยังไม่มีบราวเซอร์ตัวใดที่รู้จักและสามารถทำงานร่วมกับภาษาจาวาได้เลย และแน่นอนว่าบราวเซอร์ที่ได้รับการพัฒนาต่อ ๆ มาย่อมจะมีการเพิ่มเติมความสามารถที่ดีกว่า HotJava อย่างแน่นอนคือ Netscape Navigator เป็นบราวเซอร์ที่บริษัท Netscape Communications ได้สร้างและเป็นบริษัทแรกที่ได้ลิขสิทธิ์ภาษาจาวาจากบริษัท Sun Web Browser โดย Netscape เวอร์ชันแรกที่สนับสนุนจาวาคือ Netscape 3.0 เพื่อใช้สำหรับการทำงานบน Windows3.X, Windows95, WindowsNT, Solaris และ Apple Macintosh และยังมีเว็บบราวเซอร์ของอีกบริษัทหนึ่งที่มีชื่อเสียงไม่แพ้ Netscape คือบริษัท Microsoft ซึ่งได้ออกเว็บบราวเซอร์ที่สนับสนุนจาวาในชื่อ Internet Explorer 3.0 เพื่อให้เว็บบราวเซอร์ที่บริษัทพัฒนาเป็นเวอร์ชันล่าสุดในขณะนั้น สามารถรองรับการออกแบบโฮมเพจที่ถูกออกแบบด้วยภาษาจาวา

ปัจจุบันนี้มีบริษัทที่สร้างเว็บบราวเซอร์อื่น ๆ อีกเช่น AOL (America online) , IBM ซึ่งแน่นอนว่าปัจจุบันนี้เว็บบราวเซอร์เหล่านั้นสามารถสนับสนุนกับโฮมเพจในเครือข่ายอินเตอร์เน็ตที่ใช้ภาษาจาวาแน่นอนเพราะปัจจุบันได้มีการนำจาวาไปประยุกต์ใช้งานในงานด้านต่าง ๆ ทางธุรกิจ การศึกษา บันเทิง ฯลฯ อย่างมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ความแตกต่างระหว่างจาวาแอพเพล็ต และจาวาแอพพลิเคชัน

จาวาแอพเพล็ตกับจาวาแอพพลิเคชัน เป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อใช้งานบนอินเทอร์เน็ตเหมือนกัน ทำงานอยู่บนโฮมเพจร่วมกับภาษา HTML ที่ใช้สร้างโฮมเพจเหมือนกันลักษณะที่แตกต่างกันก็คือ

- จาวาแอพเพล็ต ไม่สามารถทำงานเดี่ยว ๆ ด้วยตนเองได้จะทำงานร่วมกันกับไฟล์ .html โดยเราจะเรียกใช้ไฟล์ที่เขียนจากจาวานี้ผ่านทาง Tag คำสั่ง <APPLET>...</APPLET> ภายในไฟล์ HTML จากนั้นก็จะเรียกใช้ไฟล์โปรแกรม javac.exe มาทำการคอมไพล์เพื่อที่จะได้ไฟล์ที่มีนามสกุล .class ไว้สำหรับให้ไฟล์ HTML เรียกผ่านวิธีสั่งแสดงผลการทำงานของจาวาแอพเพล็ต เราจะใช้โปรแกรม appletviewer.exe หรือเว็บบราวเซอร์เช่น Netscape, Hotjava, Internet Explorer ฯลฯ โดยเลือกใช้ตัวใดตัวหนึ่งก็ได้
- จาวาแอพพลิเคชันตัวที่ 2 นี้มันจะสามารถปฏิบัติงานเดี่ยว ๆ ได้ด้วยตัวเองคล้ายกับการเขียนโค้ดในภาษาโปรแกรมทั่วไป สิ่งที่ควรจดจำไว้ให้ดีในการเขียนจาวาแอพพลิเคชันก็คือ เราจะต้องมีเมธอดของ Main(string args[]) อย่างน้อย ๆ 1 เมธอดเพื่อใช้เป็นจุดเริ่มต้นของโปรแกรม อันนี้ถ้าหากสังเกตให้ดีจะเหมือนกับการเขียนโปรแกรมภาษาซีทั่ว ๆ ไป วิธีการคอมไพล์ก็จะใช้โปรแกรม javac.exe เช่นเดียวกับจาวาแอพเพล็ต ส่วนการสั่งแสดงผลเราจะใช้โปรแกรม java.exe แทน ซึ่งจะแตกต่างจากจาวาแอพเพล็ต ตรงจุดนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Java Servlet

3.1 ความหมายของจาวาเซิร์ฟเล็ท ?

เซิร์ฟเล็ท (Servlet) เป็น Server Side Application แบบหนึ่งซึ่งอ้างอิงคอนเซ็ปมาจาก CGI ข้อดีของเซิร์ฟเล็ทที่อยู่เหนือ CGI อย่างแรกก็คือตัวภาษาที่ใช้เขียนซึ่งก็คือจาวานั่นเอง จาวาเป็นภาษาที่ใช้คอนเซ็ปของ Object Oriented ในการเขียน หลายคนที่เกี่ยวข้องกับการเขียนโปรแกรมสำหรับโปรเจคใหญ่ ๆ จะทราบดีว่า Object Oriented สามารถลดความซับซ้อนของโครงสร้างโปรแกรมรวมไปถึงการอำนวยความสะดวกในการนำกลับมาใช้ใหม่ (reuse) ส่วนของโปรแกรมที่เขียนไว้แล้วเพียงไร นอกจากนี้จาวายังเป็นภาษาที่เป็นลักษณะแบบ platform independent ซึ่งจะช่วยให้สามารถที่จะทำการพัฒนาระบบโดยใช้สภาวะแวดล้อมอะไรก็ได้ซึ่งโดยทั่วไปมักนิยมใช้สภาวะแวดล้อมบนระบบปฏิบัติการวินโดวส์ โดยจะนำโปรแกรมที่เขียนเสร็จแล้วมารันบนสภาวะแวดล้อมบนระบบปฏิบัติการยูนิกส์ (UNIX) เพื่อเพิ่มความเสถียรภาพของโปรแกรมแทน

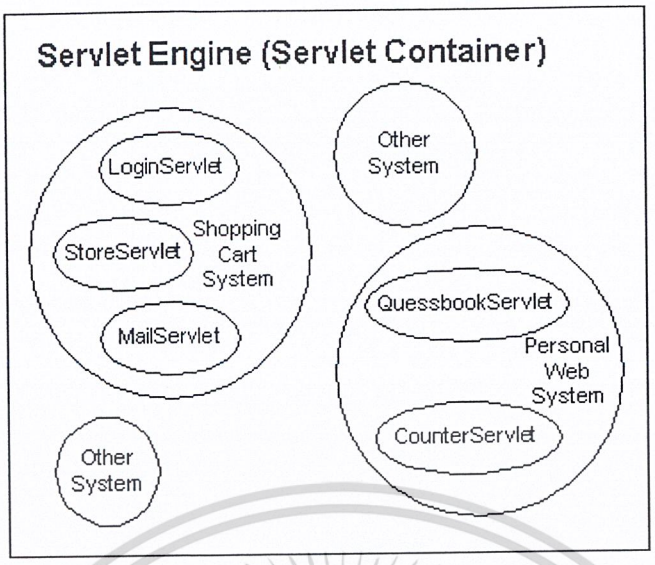
นอกจากนี้เซิร์ฟเล็ทยังมีความเร็วที่สูงกว่า CGI เพราะเซิร์ฟเล็ทใช้หลักการของเทรด (thread) โดยจะทำการสร้าง 1 เทรดต่อหนึ่ง รีเควสที่มาจากไคลเอ็นท์ซึ่งในทางกลับกัน CGI จะทำการสร้าง 1 โปรเซส (process) ต่อหนึ่งรีเควส ซึ่งจะทำให้เปลืองทรัพยากรมากกว่าและโปรเซส ในการรันก็จะช้ากว่าด้วย ท้ายที่สุดจุดเด่นที่สำคัญของเซิร์ฟเล็ทก็คือ API (Application Programming Interface) โดยระบบที่ทำการพัฒนาโดยใช้คอนเซ็ปของเซิร์ฟเล็ทจะสามารถเรียกใช้ API ที่ทางจาวามีมาให้ (javax.servlet.*, javax.servlet.http.*) ซึ่งจะช่วยทำให้การพัฒนาระบบดังกล่าวง่ายและเร็วยิ่งขึ้น

3.2 Servlet Engine

3.2.1 ความหมายของ Servlet Engine

ในการรันระบบที่เขียนขึ้นโดยใช้หลักการของเซิร์ฟเล็ทจะต้องนำระบบดังกล่าวมาบรรจุอยู่ในสิ่ง ๆ หนึ่งที่เรียกว่าเซิร์ฟเล็ทเอ็นจิ้น (Servlet Engine) ให้นึกว่าเซิร์ฟเล็ทเอ็นจิ้นคล้าย ๆ กับกล่อง ๆ หนึ่งที่ใส่ลูกปิงปองไว้หลายลูก โดยลูกปิงปองแต่ละลูกก็คือระบบ ๆ หนึ่งนั่นเอง หลายคนอาจสงสัยทำไมถึงใช้คำว่าระบบ โดยทั่วไป Server Side Application หนึ่ง ๆ ที่ถูกเขียนขึ้นโดยใช้ Servlet API จะถูกเรียกว่าเซิร์ฟเล็ทในหนึ่งระบบอาจประกอบด้วยเซิร์ฟเล็ทหลายอัน ยกตัวอย่างเช่น ระบบที่เกี่ยวกับ Shopping Cart อาจจะประกอบด้วยเซิร์ฟเล็ทที่ทำหน้าที่ในการเช็คล็อกอิน, เซิร์ฟเล็ทที่ทำหน้าที่ในการเก็บข้อมูลสินค้า, เซิร์ฟเล็ทที่ทำหน้าที่ในการส่งเมล์กลับไปยังลูกค้าเพื่อบอกว่าได้ทำการส่งของไปให้แล้ว เป็นต้น ดังนั้นถ้ามองโดยรวมแล้วเซิร์ฟเล็ทเอ็นจิ้นก็คือที่รวมของระบบตั้งแต่หนึ่งระบบถึงหลายระบบ โดยแต่ละระบบจะประกอบด้วยเซิร์ฟเล็ทหนึ่งอันหรือมากกว่า ดังรูปที่ 3-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-1 Servlet Engine and its Servlets

เซิร์ฟเล็ทเอ็นจิ้นเป็นเพียงกล่อง ๆ หนึ่งที่ใช้บรรจุและรันกลุ่มของเซิร์ฟเล็ทเท่านั้น ในการที่จะทำการติดต่อสื่อสารกับไคลเอ็นท์ตัวเซิร์ฟเล็ทเอ็นจิ้น นี้จะต้องทำงานร่วมกันเว็บเซิร์ฟเวอร์ซึ่งเปรียบเสมือนฉากหน้าที่ติดต่อกับไคลเอ็นท์อีกทีหนึ่ง เมื่อใดก็ตามที่มีรีเควสส่งมาจากไคลเอ็นท์ ถ้ารีเควส นั้นเจาะจงมาที่ตัวเซิร์ฟเล็ททางเว็บเซิร์เวอร์ก็จะทำการส่งต่อ (forward) ตัวรีเควสนั้นมาให้เซิร์ฟเล็ทเอ็นจิ้น ซึ่งทางเซิร์ฟเล็ทเอ็นจิ้นก็จะทำการเรียกเซิร์ฟเล็ทที่ไคลเอ็นท์ต้องการขึ้นมาทำการประมวลผลรีเควสนั้น โดยท้ายสุดเซิร์ฟเล็ทจะส่งผลกลับไปให้เซิร์ฟเล็ทเอ็นจิ้น, เซิร์ฟเล็ทเอ็นจิ้นก็จะส่งต่อผลที่ได้กลับไปให้เว็บเซิร์ฟเวอร์ ซึ่งเว็บเซิร์ฟเวอร์ก็จะส่งผลกลับไปให้ไคลเอ็นท์

เซิร์ฟเล็ทเอ็นจิ้นอาจจะเป็นส่วนที่ติดมากับเว็บเซิร์ฟเวอร์อยู่แล้วยกตัวอย่างเช่น เซิร์ฟเล็ทเอ็นจิ้นที่อยู่ใน Netscape Enterprise Server, IBM WebSphere หรืออาจจะเป็นส่วนที่เป็น Add-on ให้กับ เว็บเซิร์ฟเวอร์ก็ได้เช่น Apache Jserv, Tomcat, JRun หรือแม้กระทั่งเป็นส่วนหนึ่งที่อยู่ในเว็บแอพลิเคชัน เช่น BEA Weblogic เป็นต้น ทั้งนี้การเลือกใช้ เซิร์ฟเล็ทเอ็นจิ้นแต่ละชนิดก็มักขึ้นอยู่กับปัจจัยหลายอย่างเช่น ความสะดวกในการรวมระบบที่จะสร้างขึ้นมาใหม่กับระบบที่มีอยู่แล้ว, งบประมาณที่มีอยู่สำหรับโครงการหรืออาจจะรวมไปถึงทักษะและประสบการณ์ส่วนตัวของนักพัฒนาแต่ละคน

3.2.1 การทำงานของ Servlet Engine ?

สมมุติว่ามีอินเทอร์เฟส (interface) อันหนึ่งชื่อเซิร์ฟเล็ทโดยอินเทอร์เฟส นี้ประกอบไปด้วยฟังก์ชันสองฟังก์ชันดัง code snippet ข้างล่างนี้

```
interface Servlet {
public void init();
public void service();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการสร้างคลาส ๆ หนึ่งชื่อ MyServlet โดยคลาสดังกล่าวอิมพลีเมนต์ *Servlet interface* คลาส MyServlet อาจจะเป็นอย่างข้างล่างนี้

```
class MyServlet implements Servlet {
public void init() {
System.out.println("Calling MyServlet init()...");
}
public void service() {
System.out.println("I'm MyServlet");
}
}
```

จะสังเกตเห็นว่าคลาส MyServlet ทำการอิมพลีเมนต์ (implement) ฟังก์ชั่น init() และ service() ซึ่งเป็นฟังก์ชันที่ถูกกำหนดอยู่ในเซิร์ฟเล็ทอินเทอเฟส (Servlet interface)

เซิร์ฟเล็ทเอ็นจิ้นใช้ประโยชน์จากคอนเซ็ปของอินเทอร์เฟสในการโหลดเซิร์ฟเล็ทต่าง ๆ ในช่วง Runtime โดยเซิร์ฟเล็ทต่าง ๆ จะถูกโหลดและ cast กลับไปเป็นอยู่ในรูปของอินเทอร์เฟสแทน ถ้าดูจากตัวอย่างหลังจาก MyServlet ถูก cast กลับไปเป็นอินสแตนท์ (instance) ของเซิร์ฟเล็ทอินเทอร์เฟสแล้วทางเซิร์ฟเล็ทเอ็นจิ้นก็จะสามารถเรียกฟังก์ชันอะไรก็ได้ที่ถูกกำหนดอยู่ในเซิร์ฟเล็ทอินเทอร์เฟส (ในที่นี้ก็คือ init() และ service())

ตัวอย่างโค้ด (code) ง่าย ๆ ที่เซิร์ฟเล็ทเอ็นจิ้นใช้ในการโหลดเซิร์ฟในช่วง Runtime อาจจะเป็นอย่างข้างล่างนี้

```
public class SimpleServletEngine {
public static void main(String args[]) throws Exception {
if (args.length <= 0) {
System.out.println("Usage: java SimpleServletEngine javaClassName");
System.exit(0);
}
System.err.println("loading class " + args[0] + "...");
Class cls = Class.forName(args[0]);
Servlet servlet = (Servlet) cls.newInstance(); // casting
servlet.init();
servlet.service();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเซิร์ฟเล็ทที่เซิร์ฟเล็ทเอ็นจิ้นโหลดจะถูกอ้างอิงมาจากอาร์กิวเมนต์ (argument) ที่ชื่อ javaClassName โดยอาร์กิวเมนต์นี้จะถูกส่งผ่านไปให้ Class.forName() เพื่อเรียก Class object ที่จะใช้สร้างเซิร์ฟเล็ทอินสแตนท์ขึ้นมาโดยใช้คำสั่ง Class.newInstance() ซึ่งหลังจากนั้น เซิร์ฟเล็ทอินสแตนท์ที่ได้จะถูก cast กลับให้กลายเป็นอินสแตนท์ของเซิร์ฟเล็ทอินเทอร์เฟสเพื่อเซิร์ฟเล็ทเอ็นจิ้นจะได้ใช้ในการเรียกฟังก์ชั่น init() และ service()

อาจพูดง่าย ๆ ว่าหลักการทำงานของเซิร์ฟเล็ทเอ็นจิ้นก็คือการโหลดคลาสใดคลาสหนึ่งขึ้นมาโดยคลาสนั้นจะต้องอิมพลีเมนต์ตัวอินเตอร์เฟสที่ชื่อเซิร์ฟเล็ทซึ่งทางเซิร์ฟเล็ทเอ็นจิ้นทราบอยู่แล้วว่ามีฟังก์ชันอะไรประกอบอยู่ในเซิร์ฟเล็ทอินเทอร์เฟสบ้างโดยถ้าเซิร์ฟเล็ทเอ็นจิ้นทำการ cast คลาสที่อิมพลีเมนต์เซิร์ฟเล็ทอินเทอร์เฟส (ในที่นี้ก็คือ MyServlet) กลับไปเป็นอินสแตนท์ของเซิร์ฟเล็ทอินเทอร์เฟส แล้วตัวเซิร์ฟเล็ทเอ็นจิ้นก็จะสามารถเรียกฟังก์ชันที่ชื่อ init() และ service() ที่กำหนดอยู่ในเซิร์ฟเล็ทอินเทอร์เฟสจากคลาสดังกล่าวได้

ในกรณีที่ไม่ต้องการอิมพลีเมนต์ฟังก์ชันทุกฟังก์ชันที่ถูกกำหนดอยู่ในเซิร์ฟเล็ทอินเทอร์เฟส ก็อาจจะสร้างคลาสที่เป็น abstract ขึ้นมาคลาสหนึ่งก่อนโดยคลาสนี้จะทำการอิมพลีเมนต์ ฟังก์ชันบางฟังก์ชันที่อยู่ในเซิร์ฟเล็ทอินเทอร์เฟสแล้วให้ซับคลาส (subclass) ของ abstract คลาสนี้ทำการอิมพลีเมนต์ฟังก์ชันที่เหลือ ยกตัวอย่างเช่น

```
abstract class HttpServlet implements Servlet {
public void init() {
System.out.println("Calling HttpServlet init()...");
}
// not implemented yet
public abstract void service();
}
```

ถ้าต้องการสร้างเซิร์ฟเล็ทขึ้นมาโดยจะอิมพลีเมนต์ แค่ฟังก์ชัน service() อย่างเดียว ก็เพียงซับคลาสคลาส HttpServlet เท่านั้นโดยคลาสเซิร์ฟเล็ทอาจจะเป็นดังตัวอย่างข้างล่างนี้

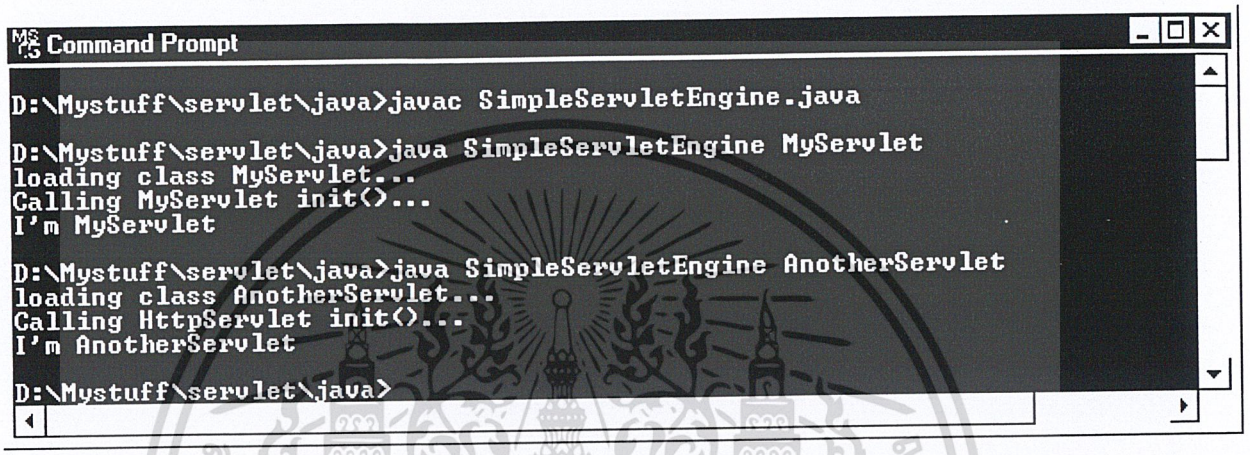
```
class AnotherServlet extends HttpServlet {
public void service() {
System.out.println("I'm AnotherServlet");
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

จะสังเกตเห็นว่าคลาส AnotherServlet จะไม่ต้องทำการอิมพลีเมนต์ฟังก์ชัน init() เพราะตัว parent คลาสของมันซึ่งก็คือ HttpServlet ได้ทำการอิมพลีเมนต์ไปแล้ว

ในการโหลดคลาส AnotherServlet เข้าไปยังเซิร์ฟเล็ทเอ็นจิ้นใช้หลักการเดียวกันกับคลาส MyServlet ข้างต้นคือโหลดคลาส AnotherServlet แล้ว cast กลับให้กลายเป็นอินสแตนท์ของเซิร์ฟอินเทอร์เฟสจากการรันโปรแกรม SimpleServletEngine โดยโหลดเซิร์ฟเล็ท (หรือคลาส) ที่ชื่อ MyServlet และ AnotherServlet ก็จะออกมาเป็นดังรูป



รูปที่ 3-2 การโหลด Servlet ต่าง ๆ ของ SimpleServletEngine

ในการใช้งานจริงเมื่อไรก็ตามที่ต้องการติดตั้งเซิร์ฟเล็ทเข้าไปยังเซิร์ฟเล็ทเอ็นจิ้นจะต้องทำการใส่รายละเอียดต่าง ๆ ของเซิร์ฟเล็ทเช่น ชื่อคลาส, parameters ต่าง ๆ ลงไปยัง properties file ของ เซิร์ฟเล็ทเอ็นจิ้น (อาจอยู่ในรูปของ xxx.properties ไฟล์หรือ xxx.xml ไฟล์) เพื่อที่จะบอกเซิร์ฟเล็ทเอ็นจิ้น ให้รับรู้และโหลดเซิร์ฟเล็ทได้อย่างถูกต้อง

3.2.2 Interface javax.servlet.Servlet

ถ้าใครอ่านหลักการทำงานของเซิร์ฟเล็ทเอ็นจิ้น จากหัวข้อที่แล้วก็อาจจะเริ่มเข้าใจแล้วว่าเซิร์ฟเล็ททำงานอย่างไร (สำหรับคนที่ข้ามมาอ่านหัวข้อนี้เลยก็ไม่ผิดกติกาแต่อย่างใด) หัวใจของเซิร์ฟเล็ทจริง ๆ อยู่ที่อินเทอร์เฟสที่ชื่อ javax.servlet.Servlet* โดยทุกเซิร์ฟเล็ทที่ถูกเขียนขึ้นจะต้องทำการอิมพลีเมนต์ตัวอินเทอร์เฟสนี้ไม่ทางตรงก็ทางอ้อม (ทางตรงก็คือการอิมพลีเมนต์ตัวอินเทอร์เฟสนี้เลย ส่วนทางอ้อมก็คือการให้เซิร์ฟเล็ททำการซับคลาสคลาสบางคลาสที่ได้ทำการอิมพลีเมนต์ตัวอินเทอร์เฟสนี้ไว้แล้ว) เหตุผลว่าทำไมต้องอิมพลีเมนต์ตัวอินเทอร์เฟสนี้เพราะว่าเมื่อไรก็ตามที่มีรีเควสจากไคลเอ็นท์เข้ามายังเซิร์ฟเล็ทเอ็นจิ้น ตัวเซิร์ฟเล็ทเอ็นจิ้น จะทำการหาเซิร์ฟเล็ทที่รีเควสดังกล่าวอ้างถึง หลังจากนั้นเซิร์ฟเล็ทเอ็นจิ้น จะทำการเรียกฟังก์ชันต่าง ๆ ที่อยู่ในเซิร์ฟเล็ทเพื่อทำการประมวลผลรีเควสของไคลเอ็นท์โดยฟังก์ชันที่เซิร์ฟเล็ทเอ็นจิ้น จะทำการเรียกก็คือฟังก์ชันที่เซิร์ฟเล็ทได้ทำการอิมพลีเมนต์ซึ่งเป็นฟังก์ชันที่ถูกกำหนดอยู่ใน javax.servlet.Servlet อินเทอร์เฟสนั่นเอง หลายคนอาจจะถามว่าทำไมเซิร์ฟเล็ทเอ็นจิ้น ถึงเรียกฟังก์ชันที่ถูกกำหนดอยู่ในอินเทอร์เฟสนี้ เหตุผลง่าย ๆ ก็คือเซิร์ฟเล็ทเป็นส่วนที่ถูกโหลดเข้าไปในเซิร์ฟเล็ทเอ็นจิ้น

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42808

ในช่วง Runtime ตัวเซิร์ฟเล็ทเอ็นจิ้น เองไม่สามารถทราบได้ว่าเซิร์ฟเล็ทต่าง ๆ มีฟังก์ชันอะไรประกอบอยู่บ้างนอกเสียจากว่าเซิร์ฟเล็ทนั้นได้ทำการอิมพลีเมนต์ฟังก์ชันที่เป็นมาตรฐานที่เซิร์ฟเล็ทเอ็นจิ้น รับรู้ซึ่งนี่ก็คือเหตุผลว่าทำไมทุกเซิร์ฟเล็ทจะต้องทำการอิมพลีเมนต์ตัว javax.servlet.Servlet อินเทอร์เฟส

อย่างไรก็ตามสามารถทำให้เซิร์ฟเล็ทเอ็นจิ้น เรียกฟังก์ชันอื่น ๆ ที่อยู่ในเซิร์ฟเล็ทได้ซึ่งวิธีการก็คือการใส่ฟังก์ชันดังกล่าวเข้าไปในส่วน implementation ของฟังก์ชันต่าง ๆ ที่ถูกกำหนดอยู่ใน javax.servletServlet อินเทอร์เฟสนั่นเอง

สำหรับคนที่ยังใหม่กับจาวาอาจจะสงสัยว่า javax คืออะไร? javax คือ standard extention package ของ Sun ที่อยู่นอกเหนือจาก core API ที่อยู่ใน JDK ตัวเซิร์ฟเล็ทที่พูดถึงอยู่นี้ก็เป็นหนึ่งใน standard extension ของ Sun เช่นเดียวกัน ดังนั้น API ทุกอย่างที่เกี่ยวกับเซิร์ฟเล็ทก็จะเป็น javax.servlet.XXX.YYY.ZZZ

ดังนั้นหลักการง่าย ๆ ในการสร้างเซิร์ฟเล็ทก็คือการสร้างคลาสขึ้นมาคลาสหนึ่งโดยคลาสนั้นจะต้องทำการอิมพลีเมนต์ตัวอินเทอร์เฟสที่ชื่อ javax.servlet.Servlet เพียงเท่านี้ก็ได้เซิร์ฟเล็ทเป็นของตัวเองแล้ว อย่างที่กล่าวมาข้างต้นว่าในการเขียนเซิร์ฟเล็ทอาจจะทำการอิมพลีเมนต์ตัว javax.servlet.Servlet อินเทอร์เฟสไม่ทางตรงก็ทางอ้อม เพื่อที่จะอำนวยความสะดวกให้กับนักพัฒนา ทางจาวาจึงได้มีการสร้างคลาสพื้นฐานที่ได้ทำการอิมพลีเมนต์ตัว javax.servlet.Servlet อินเทอร์เฟสขึ้นมาสองคลาสคือคลาส javax.servlet.GenericServlet และคลาส javax.servlet.http.HttpServlet (ซึ่งเป็นซับคลาสของ javax.servlet.GenericServlet อีกทีหนึ่ง) ดังนั้นสิ่งที่นักพัฒนาจะต้องทำก็คือการซับคลาสคลาสใดคลาสหนึ่งในสองคลาสนี้แล้ว override ฟังก์ชั่นที่ต้องการซึ่งโดยทั่วไปก็คือฟังก์ชั่นที่ชื่อ service() นั่นเอง

ถ้าดูความสัมพันธ์ระหว่าง javax.servlet.Servlet **(interface)**, javax.servlet.http.GenericServlet **(abstract class)**และ javax.servlet.http.HttpServlet **(abstract class)** จะเห็นว่าเริ่มแรก javax.servlet.Servlet จะถูกกำหนดด้วย 5 ฟังก์ชันพื้นฐานที่ทุกเซิร์ฟเล็ทจะต้องทำการอิมพลีเมนต์ดังลิสของ API ที่ปรากฏอยู่ข้างล่างนี้

```
public interface Servlet
public void init(ServletConfig config) throws ServletException;
public ServletConfig getServletConfig();
public void service(ServletRequest req, ServletResponse res) throws IOException,
ServletException;
public String getServletInfo();
public void destroy();
```

หลังจากนั้นคลาส javax.servlet.http.GenericServlet จะทำการอิมพลีเมนต์ฟังก์ชั่นต่าง ๆ ที่อยู่ใน javax.servlet.Servlet ดังตัวหนาที่ปรากฏอยู่ข้างล่าง ซึ่งนอกจากนี้คลาส GenericServlet ยังเพิ่มเติมฟังก์ชันเสริมอีกส่วนหนึ่งเพื่อช่วยให้นักพัฒนาทำการพัฒนาเซิร์ฟเล็ทได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public abstract class GenericServlet implements Servlet
public GenericServlet();
public String getInitParameter();
public Enumeration getInitParameterNames();
public ServletConfig getServletConfig();
public ServletContext getServletContext();
public String getServletInfo();
public void init();
public void init(ServletConfig config) throws ServletException;
public void log(String message);
public void log(String message, Throwable cause);
public abstract void service(ServletRequest req, ServletResponse res) throws ServletException,
IOException;
public void destroy();
```

สังเกตว่าคลาส GenericServlet เป็น abstract คลาสเพราะฟังก์ชัน public void service(ServletRequest req,...) ของ javax.servlet.Servlet อินเทอร์เฟสจะไม่ถูกอิมพลีเมนต์โดยคลาสนี้ ท้ายที่สุดคลาส GenericServlet จะถูกซับคลาสอีกทีหนึ่งโดยคลาส HttpServlet ซึ่งเป็นคลาสที่ถูกสร้างขึ้นเพื่อรองรับเซิร์ฟเล็ทที่ถูกเขียนขึ้นสำหรับใช้ในการติดต่อสื่อสารกับไคลเอ็นท์ที่ใช้ Http โปรโตคอล ดังนั้นในการเขียนเซิร์ฟเล็ท ก็เพียงทำการซับคลาสคลาส HttpServlet แล้วทำการ override ฟังก์ชันที่ต้องการเท่านั้น

```
public abstract class HttpServlet extends GenericServlet implements Serializable
public HttpServlet();
protected void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException;
protected void doPost(HttpServletRequest req,HttpServletResponse res) throws
ServletException, IOException;
protected void doPut(HttpServletRequest req,HttpServletResponse res) throws
ServletException, IOException;
protected void doDelete(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException;
protected void doOptions(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
protected void doTrace(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException;
protected void service(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException;
public void service(ServletRequest req, ServletResponse res)throws ServletException,
IOException;
protected long getLastModifed(HttpServletRequest req);
```

คลาส HttpServlet ถูกกำหนดให้เป็น abstract คลาสไม่ใช่เพราะว่ามีบางฟังก์ชันในตัวมันที่ไม่ได้มีการอิมพลีเมนต์แต่เพราะว่าผู้ที่เขียนคลาสนี้ขึ้นมาไม่ต้องการให้ทำการเรียกใช้คลาสนี้โดยตรง แต่ต้องการให้สร้างเซิร์ฟเล็ทขึ้นมาจากการซับคลาสคลาสนี้อีกทีหนึ่ง

3.3 Hello World

ในการเรียนรู้ภาษาอะไรก็ตาม โปรแกรมแรกที่มักจะเขียนก็คือโปรแกรมที่พิมพ์คำว่า Hello World ออกมา ดังนั้นในการเขียนเซิร์ฟเล็ทก็จะทำเช่นเดียวกัน โปรแกรม Hello World ที่เขียนแบบเซิร์ฟเล็ทอาจจะเป็นอย่างข้างล่างนี้

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends GenericServlet {
public void service(ServletRequest request, ServletResponse response)
throws IOException, ServletException
{
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<body>");
out.println("<head>");
out.println("<title>Hello World!</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Hello World!</h1>");
out.println("</body>");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out.println("</html>");
}
}
```

คลาส HelloWorld เป็นตัวอย่างพื้นฐานที่อธิบายหลักการทำงานของเซิร์ฟเล็ทได้เป็นอย่างดี คลาสนี้เป็นคลาสที่ซับคลาสคลาส GenericServlet โดยทำการ override ฟังก์ชัน public void service (ServletRequest request, ServletResponse response)... ซึ่งเป็นฟังก์ชันที่อยู่ในเซิร์ฟเล็ทอินเทอเฟสฟังก์ชันนี้เป็นฟังก์ชันที่สำคัญที่สุดสำหรับทุกเซิร์ฟเล็ทเพราะว่าเมื่อใดก็ตามที่มี request เข้ามายังเซิร์ฟเล็ทฟังก์ชัน service ที่อยู่ในเซิร์ฟเล็ทดังกล่าวจะถูกเรียกโดยเซิร์ฟเล็ทเอ็นจิ้น เพื่อทำการประมวลผลรีเควสที่เข้ามาเสมอ

GenericServlet เป็น abstract คลาสที่ทำการอิมพลีเมนต์ตัวเซิร์ฟเล็ทอินเทอเฟสโดยทิ้งฟังก์ชัน service ให้เป็น abstract (ไม่มีส่วนที่เป็น implementation) ซึ่งทางคลาสที่เป็นซับคลาสของ GenericServlet จะต้องมีหน้าที่ในการอิมพลีเมนต์ฟังก์ชันนี้ยกตัวอย่างเช่นคลาส HelloWorld เป็นต้น หันกลับมาดู code ที่อยู่ใน HelloWorld กันบ้าง ในสามบรรทัดแรกจะเห็นว่าจะมีสาม package ที่ถูก import เข้ามาซึ่งก็คือ java.io.*, javax.servlet.* และ javax.servlet.http.* เรียก package java.io เข้ามาเพราะว่าต้องการใช้คลาส IOException และ คลาส PrintWriter ส่วน package javax.servlet และ javax.servlet.http จะใช้สำหรับเรียกคลาสที่เกี่ยวข้องกับเซิร์ฟเล็ททั้งหมด

โค้ดที่อยู่ในฟังก์ชัน service ตัวฟังก์ชัน service จะมีอาร์กิวเมนต์อยู่ด้วยกันสองตัวคือ ServletRequest และ ServletResponse โดยเมื่อใดก็ตามที่ฟังก์ชัน service ถูกเรียกโดยเซิร์ฟเล็ทเอ็นจิ้น ตัวเซิร์ฟเล็ทเอ็นจิ้น จะทำการส่งผ่านออฟเจคมายังอาร์กิวเมนต์เหล่านี้ซึ่งก็คือ ServletRequest และ ServletResponse ออฟเจคนั่นเอง ServletResponse ออฟเจคเป็นตัวที่เก็บรายละเอียดของ Request ที่ส่งมาโดยไคลเอ็นท์ทั้งหมดโดยเซิร์ฟเล็ทสามารถใช้ออฟเจคนี้เพื่ออ่าน parameters ต่าง ๆ ที่อยู่ในรีเควส เข้ามาประมวลผลได้ หลังจากที่เซิร์ฟเล็ททำการประมวลผลเสร็จแล้วเซิร์ฟเล็ทก็สามารถส่งผลที่ได้กลับออกไปให้ไคลเอ็นท์ได้โดยใช้ ServletResponse อ็อบเจ็กต์ อย่างไรก็ตามฟังก์ชัน service ที่อยู่ใน Hello World Servlet ของไม่ได้มีการประมวลผลใด ๆ เพียงแต่จะส่ง HTML Stream ที่เขียนคำว่า "Hello World" กลับไปให้ไคลเอ็นท์เท่านั้น

ในการส่งผลต่าง ๆ กลับไปให้ไคลเอ็นท์ขั้นตอนแรกที่เซิร์ฟเล็ทต้องทำคือการเซ็ต Header โดยในตัวอย่าง Header ที่ถูกส่งกลับไปก็คือ Content Type (MIME Type) Header ซึ่งเป็นตัวบอกไคลเอ็นท์ว่า Stream ที่กำลังจะได้รับเป็น Stream ชนิดไหน (ให้นึกถึง MIME Type เป็นลักษณะคล้าย ๆ กับไฟล์ extension) ยกตัวอย่างเช่น text/html จะบอกว่า Stream ที่กำลังส่งออกไปเป็น HTML Stream หรือ image/gif จะบอกว่า Stream ที่ส่งออกไปเป็น Stream ของ gif image เป็นต้น ตัวเซิร์ฟเล็ทสามารถทำการเซ็ต Content Type ได้โดยใช้ฟังก์ชัน setContentType() ซึ่งเป็นฟังก์ชันที่อยู่ใน ServletResponse นั่นเอง ในการส่ง Stream ต่าง ๆ ไปให้ไคลเอ็นท์ตัวเซิร์ฟเล็ทสามารถส่ง Stream ออกไปได้สองแบบคือ text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stream และ byte Stream ในกรณีนี้จะส่ง Stream ออกไปเป็นแบบ text Stream ดังนั้นคลาสที่จะใช้ในการเขียนลงไปที่ ServletResponse ก็คือคลาส PrintWriter นั่นเอง

ในการเรียกใช้คลาส PrintWriter สำหรับเขียนลงไปที่ ServletResponse ก็สามารถทำได้ง่าย ๆ โดยการเรียกฟังก์ชัน ServletResponse.getWriter() ซึ่งจะ return ตัว PrintWriter instance ที่จะใช้ในการเขียน output ต่าง ๆ ลงไปยังตัว ServletResponse ที่ใช้สำหรับเรียกอินสแตนท์ ดังกล่าวขึ้นมา ท้ายสุด HelloWorld Servlet จะใช้ PrintWriter ทำการพิมพ์ text ต่าง ๆ ที่ใช้ในการสร้าง HTML Output Stream ที่เขียนคำว่า "Hello World" ออกไปให้ไคลเอ็นท์

3.4 การรัน Servlet

เซิร์ฟเล็ทเป็นคอนเซ็ปหนึ่งในจาวาที่เขียนขึ้นเพื่อใช้งานสำหรับ Server Side Application ถ้าใครเคยเขียนแอพลิเคชันโดยใช้จาวามาก่อนจะเห็นว่าตัว entry point ของ application จะเป็น static ฟังก์ชันที่ชื่อ main() ยกตัวอย่างเช่น

```
public class HelloWorld {
public static void main(String args[]) {
System.out.println("Hello World");
}
}
```

ตัวเซิร์ฟเล็ทเองจะมีลักษณะคล้าย ๆ Applet ตรงที่ว่าในการรัน ฟังก์ชันที่ใช้เป็น entry point จะไม่ใช่ฟังก์ชัน main() แต่จะกลายเป็นฟังก์ชันอื่นที่ถูกกำหนดไว้ใช้สำหรับตัวมันเองโดยเฉพาะยกตัวอย่างเช่นใน Applet ฟังก์ชันที่ถูกใช้เป็น entry point ก็จะเป็นฟังก์ชันพวก init(), start() ซึ่งสำหรับเซิร์ฟเล็ทแล้วฟังก์ชันที่ถูกใช้เป็น entry point ก็คือ service() ซึ่งก็คือฟังก์ชันที่ถูกกำหนดอยู่ใน javax.servlet.Servlet อินเทอร์เฟสนั่นเอง

ในการรันเซิร์ฟเล็ทสิ่งที่ต้องการมีดังต่อไปนี้คือ

1) JDK

สำหรับคนที่ใช้ Windows, Sun หรือ Linux ให้ไปดาวโหลดที่ http://java.sun.com

สำหรับคนที่รัก Linux โดยเฉพาะอาจจะไปดาวโหลดที่นี่ก็ได้ http://www.blackdown.org

2) Servlet API Package and Servlet Engine

Package นี้เป็นตัวที่เก็บ Servlet API ทั้งหมดไว้ซึ่งก็คือ javax.servlet.* และ javax.servlet.http.* package ในการรันเซิร์ฟเล็ทสามารถดาวน์โหลดตัวที่เป็น Java Servlet Development Kit จาก Sun ซึ่งเป็นตัวที่เก็บ Servlet API และตัวเซิร์ฟเล็ทเอ็นจิ้น พื้นฐานเอาไว้โดยสามารถดาวโหลดได้จาก http://java.sun.com/products/servlet/download.html (เลือกในส่วนที่เรียกว่า Java Servlet Development

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kit) อย่างไรก็ตามสำหรับผู้ที่ต้องการเรียนรู้เซิร์ฟเล็ทอย่างจริงจังและต้องการที่จะนำไปใช้งานจริงนั้นควรที่จะใช้เซิร์ฟเล็ทเอ็นจิ้น ที่เป็น standard ซึ่งโดยทั่วไปจะรวม Servlet API เข้าไปอยู่ในตัวเรียบร้อยแล้ว อย่างที่กล่าวมาข้างต้นว่าเซิร์ฟเล็ทเอ็นจิ้น อาจจะเป็นส่วนหนึ่งของเว็บเซิร์ฟเวอร์หรืออาจจะเป็นตัว add-on ก็ได้ ดังนั้นสำหรับคนที่มีเว็บเซิร์ฟเวอร์เปล่า ๆ ที่ยังไม่มีเซิร์ฟเล็ทเอ็นจิ้น ติดอยู่ก็อาจจะเลือกเซิร์ฟเล็ทเอ็นจิ้น ที่เป็นลักษณะ add-on โดยติดเข้าไปเป็น extension module สำหรับเว็บเซิร์ฟเวอร์ของตัวเองก็ได้ แต่สำหรับคนที่ยังไม่มีทั้งเว็บเซิร์ฟเวอร์และเซิร์ฟเล็ทเอ็นจิ้น ก็อาจจะเลือกดาวโหลดเว็บเซิร์ฟเวอร์ที่มีเซิร์ฟเล็ทเอ็นจิ้น อยู่ในตัวแล้วก็ได สำหรับลิสของเซิร์ฟเล็ทเอ็นจิ้น ที่มีอยู่สามารถหาดูจากที่นี่ได้ http://www.javaskyline.com/serv.html

เซิร์ฟเล็ทเอ็นจิ้นท์ที่จะใช้ในการสาธิตการติดตั้งและรันเซิร์ฟเล็ทจะเป็นเซิร์ฟเล็ทเอ็นจิ้น ที่มาจาก Apache Group ซึ่งถือว่าเป็นเซิร์ฟเล็ทเอ็นจิ้น และ JSP Reference Implementation ของ Sun โดยมี code name ที่ชื่อว่า Tomcat

สำหรับผู้ที่สนใจการติดตั้ง Tomcat เป็นภาษาไทยสามารถหาอ่านได้จาก

http://www.jarticles.com/jsp/index.html

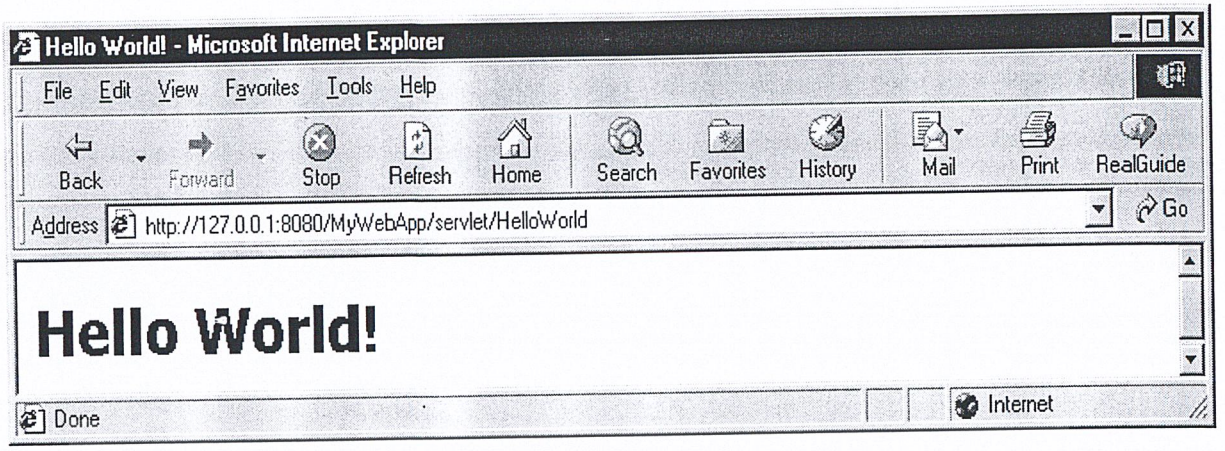
หากต้องการรัน HelloWorld Servlet ที่ Context path ที่ชื่อ MyWebApp โดยชี้ไปที่ไดเรคทรอรี่ที่ชื่อ D:\MyWebApplication\ ก็สามารถทำการแอท Context path นี้เข้าไปใน Tomcat ได้โดยการเปิดไฟล์ที่ชื่อ server.xml ซึ่งจะอยู่ในไดเรคทรอรี่ conf ของ Tomcat Home directory เสร็จแล้วให้ทำการเพิ่มบรรทัดข้างล่างนี้เข้าไปในส่วนของ <Context path> tag

```
<Context path="/MyWebApp" docBase="D:\MyWebApplication\" debug="0" reloadable="true"/>
```

จากนั้นให้ทำการสร้าง subdirectory ที่ชื่อ WEB-INF ขึ้นมาใต้ D:\MyWebApplication\ไดเรคทรอรี่ เสร็จแล้วให้สร้างอีก subdirectory ที่ชื่อ classes ขึ้นมาใต้ WEB-INF อีกทีหนึ่งโดย subdirectory classes นี้จะเป็นที่ ๆ ใช้เก็บคลาสเซิร์ฟเล็ททั้งหมด ซึ่งท้ายสุดไฟล์ HelloWorld.java ก็จะอยู่ใน directory stucture ดังนี้ D:\MyWebApplication\WEB-INF\classes\HelloWorld.java

จากนั้นให้ทำการ compile ไฟล์ HelloWorld เหมือนปกติ โดยถ้าใครคอมไฟล์นี้ไม่ได้ ให้เช็คดูว่า Servlet API ถูกรวมอยู่ใน CLASSPATH แล้วหรือยัง เสร็จแล้วให้ทำการ start ตัว Tomcat แล้วพิมพ์ http://127.0.0.1:8080/MyWebApp/servlet/HelloWorld ก็จะได้ผลดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 HelloWorld Servlet

Tomcat จะทำการแมป /servlet ของ Context path เข้ากับเซิร์ฟเล็ทต่าง ๆ ที่อยู่ใน subdirectory classes โดยอัตโนมัติ

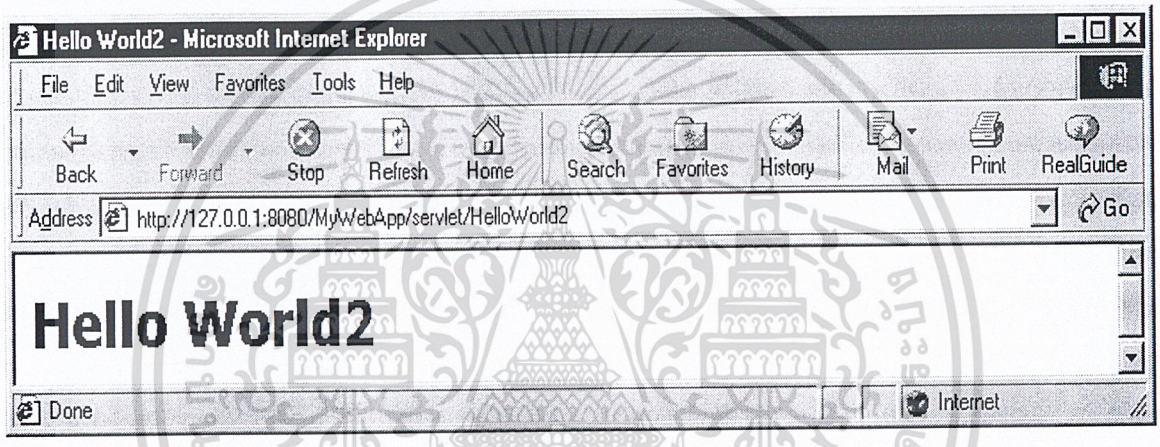
มาดูตัวอย่างที่สองกันบ้าง ตัวอย่างนี้เป็น HelloWorld อีกแบบแต่แทนที่คลาสเซิร์ฟเล็ทจะทำการซับคลาสคลาส GenericServlet ก็จะทำการซับคลาสคลาส HttpServlet ซึ่งเป็นซับคลาสของคลาส GenericServlet อีกทีหนึ่งแทน คลาส GenericServlet และคลาส HttpServlet จะต่างกันตรงที่ว่าคลาส GenericServlet จะเป็นคลาสที่ไม่เจาะจงโปรโตคอลใดโปรโตคอลหนึ่งซึ่งเหมาะสำหรับเป็น base class สำหรับเซิร์ฟเล็ทที่ต้องการอิมพลีเมนต์ตัวเซิร์ฟเล็ทที่ใช้สำหรับโปรโตคอลจำเพาะเจาะจงขึ้นมาเองยกตัวอย่างเช่นเซิร์ฟเล็ทสำหรับ RMI, Servlet สำหรับ IIOP (CORBA) หรือแม้กระทั่งเซิร์ฟเล็ทสำหรับ Http โปรโตคอลซึ่งก็คือ HttpServlet ที่กำลังพูดถึงอยู่นั่นเอง คลาส HelloWorld แบบที่สองก็จะเป็นอย่างข้างล่างนี้

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld2 extends HttpServlet {
public void service(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException
{
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<body>");
out.println("<head>");
out.println("<title>Hello World2</title>");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out.println("</head>");
out.println("<body>");
out.println("<h1>Hello World2</h1>");
out.println("</body>");
out.println("</html>");
}
}
```

ถ้าทำการรันก็จะได้ผลดังนี้



รูปที่ 3-4 HelloWorld2 Servlet

จะสังเกตเห็นว่าฟังก์ชันที่ทำการ override ก็คือฟังก์ชัน service แต่ต่างกันตรงที่ว่าฟังก์ชันนี้ไม่ใช่ฟังก์ชัน service(ServletRequest req, ServletResponse res)... แต่กลับกลายเป็น service(HttpServletRequest req, HttpServletResponse res)... แทนซึ่งเป็นฟังก์ชันที่จำเจาะจงสำหรับ Http โปรโตคอลโดยเฉพาะ หลายคนอาจสงสัยว่าฟังก์ชัน service(HttpServletRequest req, HttpServletResponse res)... จะสามารถถูกเรียกโดยเซิร์ฟเล็ทเอ็นจิ้น ได้อย่างไรในเมื่อเซิร์ฟเล็ทเอ็นจิ้น จะเรียกเฉพาะฟังก์ชัน service(ServletRequest req, ServletResponse res)... ซึ่งเป็นฟังก์ชันที่ถูกกำหนดอยู่ใน javax.servlet.Servlet อินเทอร์เฟสเท่านั้น ถ้าใครเคยเข้าไปดู code ที่อยู่ใน HttpServlet จะเห็นว่าเป็นลักษณะคล้าย ๆ อย่างนี้

```
public void service(ServletRequest req, ServletResponse res) throws ServletException,
IOException {
    HttpServletRequest request;
    HttpServletResponse response;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
try {
request = (HttpServletRequest) req;
response = (HttpServletResponse) res;
} catch (ClassCastException e) {
throw new ServletException("non-HTTP request or response");
}
service(request, response); // service(HttpServletRequest res, HttpServletResponse res) ...
}
```

จะเห็นว่าคลาส HttpServlet ก็ยังคงทำการอิมพลีเมนต์ฟังก์ชัน service(ServletRequest req,...) ซึ่งเป็นฟังก์ชันที่อยู่ใน javax.servlet.Servlet อินเทอร์เฟสอยู่เพียงแต่ว่าเมื่อใดก็ตามที่มีการเรียกฟังก์ชันนี้โดยเซิร์ฟเล็ทเอ็นจิ้น ทาง HttpServlet จะทำการ cast ตัว ServletRequest และ ServletResponse ให้กลายเป็น HttpServletRequest และ HttpServletReponse ตามลำดับ หลังจากนั้นจะส่งผ่าน object สองตัวนี้เข้าไปยัง service(request, response) ซึ่งก็คือฟังก์ชัน service(HttpServletRequest req, HttpServletResponse res)... อีกต่อหนึ่งโดยโค๊ดที่พิมพ์คำว่า HelloWorld ที่อยู่ในฟังก์ชันนี้ก็จะถูกเรียกโดยเซิร์ฟเล็ทเอ็นจิ้น ในท้ายสุด

ตัวอย่าง HelloWorld Servlet สุดท้ายที่กำลังจะกล่าวถึงนี้เป็นตัวอย่างของ HelloWorld Servlet ที่ถูกยกขึ้นมาใช้เป็นตัวอย่างของ HelloWorld Servlet ในแทบทุก Servlet Tutorial ที่มีอยู่ในโลกนี้ซึ่งที่มาที่ไปของเซิร์ฟเล็ทนี้ก็มาจากสอง HelloWorld Servlet ที่พูดถึงข้างต้นนั่นเอง

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld3 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException
{
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<body>");
out.println("<head>");
out.println("<title>Hello World3</title>");
out.println("</head>");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out.println("<body>");
out.println("<h1>Hello World3</h1>");
out.println("</body>");
out.println("</html>");
}
}
```

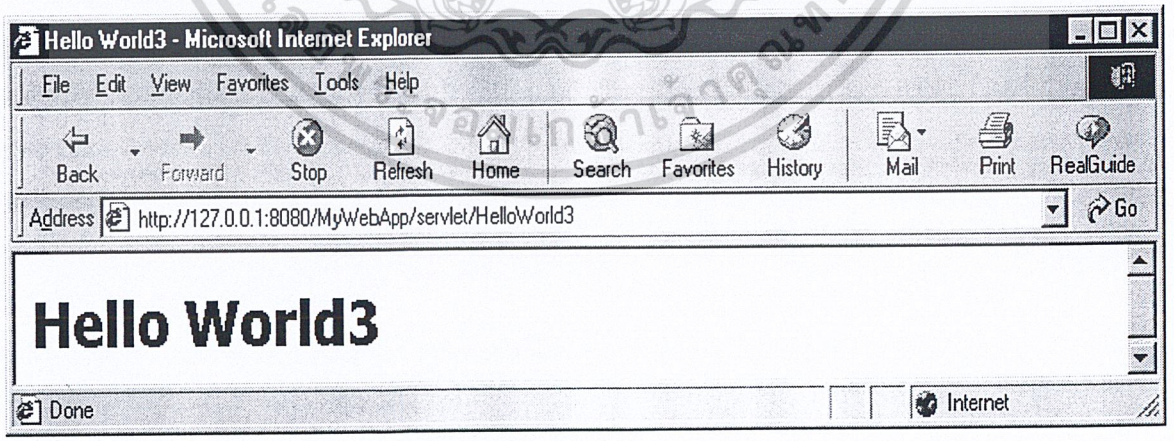
จะเห็นว่าเซิร์ฟเล็ทนี้ไม่ได้ทำการ override ฟังก์ชัน service แต่อย่างใดแต่กลับทำการ override ฟังก์ชัน **doGet()** แทน

ฟังก์ชัน doGet() นี้เป็นฟังก์ชันที่ถูกกำหนดอยู่ใน HttpServlet เพื่อรับ request ที่มาจากไคลเอ็นท์ในแบบ Http GET Method ซึ่งก็คือ request ที่มาจาก URL ธรรมดาทั่ว ๆ ไปนั่นเอง นอกจากฟังก์ชัน doGet() แล้วคลาส HttpServlet ยังมีฟังก์ชันอื่น ๆ ที่อนุญาติให้สามารถทำการ override เพื่อรองรับ request แบบอื่น ๆ ได้อีก ยกตัวอย่างเช่น

doPost() ใช้สำหรับรับ request ที่มาจากไคลเอ็นท์ในแบบ POST Method (Form request)

doDelete() ใช้สำหรับรับ request ที่มาจากไคลเอ็นท์ในแบบ DELETE Method เพื่ออนุญาติให้ไคลเอ็นท์ทำการลบ resource บางอย่างออกจากเซิร์ฟเวอร์ได้

ยังมีฟังก์ชันอีกสามอันคือ **doPut(), doOptions()** และ**doTrace()** ที่อยู่ในคลาส HttpServlet เพื่อรองรับ request แบบ PUT,OPTIONS และ TRACE แต่ไม่นิยมทำการ override ฟังก์ชันเหล่านี้ด้วยเหตุที่ว่าส่วน implementation ของฟังก์ชันเหล่านี้ที่อยู่ในคลาส HttpServlet ได้ถูกอิมพลีเมนต์มาอย่างดีเหมาะสำหรับการใช้งานทั่ว ๆ ไปแล้ว



รูปที่ 3-5 HelloWorld3 Servlet

หลายคนอาจจะสงสัยอีกครั้งว่าทำไมฟังก์ชัน doGet(), doPost() และอื่น ๆ สามารถถูกเรียกได้ในเมื่อเซิร์ฟเล็ทเอ็นจิ้น จะเรียกฟังก์ชัน service(ServletRequest req,...) ซึ่งจะส่งผ่านไปยัง service

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(HttpServletRequest req,...) เท่านั้น เหตุผลง่าย ๆ ก็คือในกรณีทั่ว ๆ ไปมักไม่นิยม override ฟังก์ชัน service(HttpServletRequest req,...) ที่อยู่ใน HttpServlet อย่างที่ทำใน HelloWorld2 เพราะว่าในฟังก์ชัน service(HttpServletRequest req,...) ได้มีส่วนที่เป็น implementation บรรจุอยู่แล้วโดย code ส่วนนี้จะเป็นส่วนที่ทำการส่งผ่าน call ของเซิร์ฟเล็ทเอ็นจิ้น ไปยังฟังก์ชัน **doXXX()** ต่าง ๆ ดัง code snippet ข้างล่าง

```
protected void service(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
String method = req.getMethod();
if (method.equals(METHOD_GET)) {
...
doGet(req, resp);
...
} else if (method.equals(METHOD_HEAD)) {
...
doHead(req, resp); // will be forwarded to doGet(req, resp)
} else if (method.equals(METHOD_POST)) {
doPost(req, resp);
} else if (method.equals(METHOD_PUT)) {
doPut(req, resp);
} else if (method.equals(METHOD_DELETE)) {
doDelete(req, resp);
} else if (method.equals(METHOD_OPTIONS)) {
doOptions(req,resp);
} else if (method.equals(METHOD_TRACE)) {
doTrace(req,resp);
} else {
...
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าถ้าไม่ทำการ override ฟังก์ชัน service นี้แล้ว ฟังก์ชันนี้ก็จะทำการส่ง request ไปให้ฟังก์ชันย่อยต่าง ๆ ที่รับผิดชอบ Http Method ต่าง ๆ ซึ่งในกรณีนี้ก็คือฟังก์ชัน doGet() นั่นเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

JavaScript

4.1 ความหมายของ JavaScript

จาวาสคริปต์ (JavaScript) เป็นภาษาสำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต เราสามารถเขียนโปรแกรมจาวาสคริปต์เพิ่มเข้าไปในเว็บเพจเพื่อใช้ประโยชน์สำหรับงานด้านต่าง ๆ ทั้งทางคำนวณ การแสดงผล การรับ-ส่งข้อมูล และที่สำคัญสามารถที่จะโต้ตอบกับผู้ใช้ได้อย่างทันทีทันใด นอกจากนี้ยังมีความสามารถด้านอื่น ๆ อีกหลายประการที่ช่วยสร้างความน่าสนใจให้กับเว็บเพจของเราได้อย่างมาก

จาวาสคริปต์ถือกำเนิดมาจากบริษัท เน็ตสเคป คอมมิวนิเคชันส์ (Netscape Communications Corporation) ถูกเปิดตัวครั้งแรกในชื่อ LiveScript พร้อมกับโปรแกรม Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจที่สามารถแลกเปลี่ยนข้อมูลกับเซิร์ฟเวอร์แบบ LiveWire ได้ หลังจากที่บริษัท ซัน ไมโครซิสเต็มส์ได้นำภาษาจาวาออกสู่ท้องตลาด ซันกับเน็ตสเคปจึงได้ร่วมมือกันปรับปรุงให้เบราเซอร์ของเน็ตสเคปสามารถใช้งานภาษาจาวาได้ และนำภาษา LiveScript มาแก้ไขปรับปรุงใหม่จากนั้นจึงเปลี่ยนชื่อเป็นจาวาสคริปต์ เมื่อปีพ.ศ. 2538

จาวาสคริปต์ เป็น “ภาษาสคริปต์เชิงวัตถุ” ที่ช่วยให้เราสามารถควบคุมเว็บเพจได้อย่างง่ายดาย สามารถทำงานข้ามแพลตฟอร์มได้ ทำหน้าที่เป็นตัวประสานระหว่างเว็บเพจ HTML, Java applet และเว็บเบราเซอร์ทั้งทางฝั่งไคล์เอนท์และเซิร์ฟเวอร์ ช่วยให้เว็บเพจที่บรรจุจาวาสคริปต์มีความน่าสนใจและสมบูรณ์มากกว่าเว็บเพจทั่วไป

จาวาสคริปต์มิได้มีความสามารถเพียงแค่ช่วยตกแต่งเว็บเพจเท่านั้น แต่มีความสามารถเหนือกว่านั้นอีกคือเหมาะสำหรับการพัฒนาโปรแกรมใช้งานในระบบอินเทอร์เน็ตทั้งทางฝั่งไคล์เอ็นต์และฝั่งเซิร์ฟเวอร์ โดยโปรแกรม Netscape Navicator ตั้งแต่เวอร์ชัน 2.0 ขึ้นไป ได้ผนวกความสามารถในการแปลชุดคำสั่งของจาวาสคริปต์ที่มากับเว็บเพจ HTML และ LiveWire ได้ด้วยในตัว

ถ้าจำแนกจาวาสคริปต์ออกตามลักษณะการทำงานทางฝั่งไคล์เอนท์และทางฝั่งเซิร์ฟเวอร์แล้ว สามารถจำแนกออกได้เป็นสองแบบ คือ

1. Navigator JavaScript เป็น Client-Side JavaScript ซึ่งหมายถึงจาวาสคริปต์ที่ถูกแปลทางฝั่งไคลเอนท์ (หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ใช้ ไม่ว่าจะเป็นเครื่องพีซี เครื่องแมคอินทอช หรืออื่น ๆ) สามารถใช้ได้เฉพาะกับ LiveWire ของเน็ตสเคปโดยตรง

2. LiveWire JavaScript เป็น Server-Side JavaScript ซึ่งหมายถึงจาวาสคริปต์ที่ถูกแปลทางฝั่งเซิร์ฟเวอร์ (หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ให้บริการเว็บ) สามารถใช้ได้เฉพาะกับ LiveWire ของเน็ตสเคปโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 Java กับ JavaScript

เมื่อครั้งที่เผยโฉมออกมาในเวอร์ชันแรกจาวาสคริปต์มิได้มีส่วนเกี่ยวข้องกับภาษาจาวาเลย เพราะเป็นการนำภาษา LiveScript ของเน็ตสเคปเองมาเปลี่ยนชื่อเท่านั้น ในเวอร์ชันต่อมาจึงมีการเพิ่มส่วนของอ๊อบเจ็กต์ในภาษาจาวามากขึ้น มีความสามารถโต้ตอบกับผู้ใช้ แต่อย่างไรก็ตามจาวาสคริปต์เองก็ยังไม่มีคลาสที่สามารถขยายเป็นลำดับชั้น ไม่มีกลไกสำหรับเก็บอ๊อบเจ็กภายในแต่กลับมีความสามารถในการจัดการกับ HTML และแอปเพล็ตของจาวาเพิ่มมากขึ้น ทำให้สามารถรวบรวมสอ๊อบเจ็กต์แต่ละส่วนมาเป็นแอปพลิเคชันเดียวกันได้ตามความต้องการ

โครงสร้างของภาษาจาวาสคริปต์มีความคล้ายคลึงกับภาษาจาวามาก โดยจาวาสคริปต์เป็นคอมพลีเมนต์ของจาวา สามารถติดต่อกับส่วนต่าง ๆ ของจาวาแอปเพล็ตโดยสคริปต์ที่เขียนขึ้นมาได้ คำสั่งของจาวาสคริปต์สามารถนำมาใช้แสดง, กำหนดคุณสมบัติ, สอบถามสถานะ หรือควบคุมการกระทำของแอปเพล็ตและปลั๊กอิน นอกจากนี้จาวาสคริปต์ยังสนับสนุนรูปแบบนิพจน์และการควบคุมพื้นฐานของภาษาจาวาอีกด้วย จาวาสคริปต์ได้ถูกออกแบบมาเพื่อใช้เป็นส่วนขยายในภาษา HTML โดยเฉพาะช่วยให้สามารถควบคุมเว็บเพจได้อย่างง่ายดาย เหมาะกับการทำงานอย่างรวดเร็ว และเน้นที่ความถูกต้องเป็นสำคัญ

ภาษาจาวาประกอบไปด้วยเอ็กซ์คลูซีฟของคลาสและเมธอดต้องมีการกำหนดคลาสและเขียนเมธอด และเน้นเรื่องความถูกต้อง โปรแกรมที่เขียนในภาษาจาวาจะมีความสมบูรณ์กว่าการเขียนด้วยจาวาสคริปต์

จาวาสคริปต์เป็นภาษาแบบอินเตอร์พรีเตอร์ (interpreter) ฉะนั้นเพียงเขียนคำสั่งในภาษาจาวาสคริปต์เก็บไว้เป็นไฟล์ข้อความร่วมกับเว็บเพจ HTML ก็ทำงานได้แล้ว ไฟล์คำสั่งในภาษาจาวาสคริปต์ อาจมีส่วนขยายเป็น htm หรือ html เหมือนไฟล์กับไฟล์เว็บเพจทั่วไป หรือมีส่วนขยายเป็น js ก็ได้

แต่ภาษาจาวาเป็นภาษาแบบคอมไพเลอร์คำสั่งในภาษาจาวาจึงต้องเขียนเก็บไว้เป็นไฟล์ข้อความที่มีส่วนขยายเป็น java หลังจากนั้นต้องนำไฟล์ดังกล่าวไปผ่านการคอมไพล์ให้เป็นไบต์โค้ด (มีส่วนขยายเป็น class) เสียก่อน จากนั้นจึงนำไฟล์มาสร้างเป็นอ๊อบเจ็กต์และแอปเพล็ตเพื่อใช้งานต่อไป

ส่วนที่เหมือนกันของภาษาทั้งสองคือ รูปแบบของโครงสร้างภาษาทั้งสองคือ รูปแบบของโครงสร้างภาษา และการเขียนโปรแกรมเชิงอ๊อบเจ็กต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JavaScript	Java
ทำงานในลักษณะอินเตอร์พรีเตอร์ไม่ต้องคอมไพล์ให้เป็น executable file	ต้องผ่านการคอมไพล์ให้เป็นไบต์โค้ดเพื่อทำงานบนเครื่องผู้ใช้
เขียนอ็อบเจ็กต์อยู่ในเว็บเพจเลย	มี object-oriented applet ที่ประกอบไปด้วยคลาสอ็อบเจ็กต์ที่สืบทอดจากคลาสตัวแม่ที่บรรจุอยู่ในเว็บเพจ
กำหนดคำสั่งจาวาสคริปต์โดยฝังอยู่กับ HTML เลย	ใช้แอปเพลตช่วยในการแสดงผลและทำงานร่วมกับเว็บเพจ HTML
ไม่จำเป็นต้องกำหนดค่าตัวแปร	จำเป็นต้องกำหนดค่าตัวแปร
มีการตรวจสอบอ๊อปเจ็กต์อ้างอิงที่รันไทม์ (dynamic binding)	มีการเรียกใช้งานอ็อปเจ็กต์อ้างอิงขณะคอมไพล์ (static binding)
ไม่สามารถบันทึกข้อมูลลงในอาร์ดดิสก์ได้โดยตรง	ไม่สามารถบันทึกข้อมูลลงในอาร์ดดิสก์ได้โดยตรง

ตาราง 4-1 เปรียบเทียบคุณสมบัติของ JavaScript กับ Java

4.3 JavaScript กับ HTML

จาวาสคริปต์เป็นคุณสมบัติเฉพาะของโปรแกรม Netscape Navigator (หมายความว่าทำงานได้ดีที่สุดกับโปรแกรม Netscape Navigator) จึงจำเป็นอย่างยิ่งที่เราควรจะทำความเข้าใจกับรูปแบบกลไกการทำงานภายใน หรือที่เรียกโดยรวมว่าเลย์เอาต์ ของ Netscape Navigator ให้ดี เพราะเลย์เอาต์มีส่วนเกี่ยวข้องกับการกำหนดรูปแบบของ HTML เพื่อทำงานร่วมกับจาวาสคริปต์เป็นอันมาก

การเขียนจาวาสคริปต์ เราอาจเขียนรวมอยู่ในไฟล์เดียวกันกับ HTML ได้ ซึ่งแตกต่างจากการเขียนโปรแกรมภาษาจาวาที่ต้องแยกออกเป็นไฟล์ต่างหาก ไม่สามารถเขียนรวมอยู่ในไฟล์ HTML ได้

วิธีการเขียนจาวาสคริปต์เพื่อสั่งให้เว็บเพจทำงาน มีอยู่ด้วยกัน 2 วิธี ดังนี้

- เขียนด้วยชุดคำสั่งและฟังก์ชันของจาวาสคริปต์เอง หรือ
- เขียนตามเหตุการณ์ที่เกิดขึ้นตามการใช้งานจากชุดคำสั่งของ HTML

เมื่อเริ่มใช้งานโปรแกรม Navigator จะอ่านข้อมูลจากส่วนบนของเพจ HTMLและทำงานไปตามลำดับจากบนลงล่าง (top-down) โดยเริ่มที่ส่วน <HEAD>...</HEAD ก่อน จากนั้นจึงทำงานในส่วน <BODY>..</BODY> เป็นลำดับต่อมา

นอกเหนือจากช่วยลดภาระการทำงานของเซิร์ฟเวอร์แล้ว จาวาสคริปต์ยังมีความสามารถที่เหนือกว่าภาษาสำหรับการตรวจสอบความถูกต้องในฟอร์มทั่วไป จาวาสคริปต์เป็นเหมือนกาวที่ทางฝั่งไคล์เอนท์ใช้ในการเก็บส่วนประกอบของเว็บเพจเข้าด้วยกันโดยการใช้ลอจิกขั้นพื้นฐานที่ตัดสินใจได้ว่าจะใช้เบราเซอร์หรือปลั๊กอินตัวใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาวาสคริปต์กลายเป็นภาษากลางสำหรับควบคุมและติดต่อสื่อสารระหว่างอ็อบเจ็กต์อย่างปลั๊กอิน, เลเยอร์, เฟรม, ฟอร์ม, สไตล์ชีต และแอปเพล็ต ซึ่งได้รับการยอมรับมากขึ้นยิ่งกว่าภาษา HTML เสียอีก

การทำงานของจาวาสคริปต์ดูไม่แตกต่างไปจาก HTML เท่าใดนัก แต่ HTML จะวางเลย์เอาต์โครงสร้างของอ็อบเจ็กภายในและส่วนเชื่อมโยงกับเว็บเพจเท่านั้น ในขณะที่จาวาสคริปต์สามารถเพิ่มเติมส่วนของการเขียนโปรแกรมและลอจิกเข้าไป

```
<FORM NAME="statform">
        <INPUT type="text" name="username" size = 20>
        <INPUT type="text" name="userage" size = 3>
</FORM>
```

สมาชิก (ในที่นี้ input 2 ชุด) ในแบบฟอร์ม statform ทำหน้าที่สะท้อนไปยังอ็อบเจ็กต์ document.statform.username และ document.statform.userage จากการอ้างอิงโดยจาวาสคริปต์ทำให้สามารถนำอ็อบเจ็กต์มาใช้งานได้ทันทีที่แบบฟอร์มถูกกำหนดขึ้นมา อย่างไรก็ตามเราไม่สามารถใช้อ็อบเจ็กต์นี้ก่อนที่แบบฟอร์ม statform จะถูกกำหนดขึ้นมาได้ ตัวอย่างต่อไปนี้จะแสดงถึงค่าต่าง ๆ ของอ็อบเจ็กต์ในสคริปต์ที่อ้างถึงแบบฟอร์มตัวอย่าง

```
<SCRIPT>
        document.write(document.statform.username.value)
        document.write(document.statform.userage.value)
</SCRIPT>
```

ถ้าเราเขียนสคริปต์ไว้ก่อนคำสั่งกำหนดแบบฟอร์ม เราจะพบข้อผิดผลาดจากการเรียกใช้อ็อบเจ็กต์ที่ไม่ได้มีอยู่จริงในโปรแกรม Navigator

เช่นเดียวกัน การกำหนดค่าพร็อพเพอร์ตีบางชนิดจะไม่สามารถเปลี่ยนแปลงค่าได้เลยตัวอย่างเช่น เรามีเว็บเพจที่กำหนดส่วนไตเติลเป็น <TITLE>My JavaScript Page</TITLE> ข้อความ My JavaScript Page จะอ้างถึงอ็อบเจ็กต์ document.title สำหรับจาวาสคริปต์ โปรแกรม Navigator แสดงส่วนนี้ในเลย์เอาต์ (ในที่นี้คือการแสดงข้อความในส่วนไตเติลบาร์ของวินโดวส์) โดยไม่สามารถเปลี่ยนแปลงได้โดยจาวาสคริปต์ (read only)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ความสามารถอื่น ๆ ของ JavaScript

แม้ว่าจาวาสคริปต์จะถูกสร้างขึ้นมาสำหรับงานตกแต่งเว็บเพจโดยเฉพาก็ตาม แต่ตัวจาวาสคริปต์เองมีขีดความสามารถเหนือกว่าที่จะใช้เป็นเครื่องมือสำหรับตกแต่งเว็บเพจเท่านั้น กล่าวคือก่อนหน้าที่จาวาสคริปต์ถือกำเนิดขึ้นมา การติดต่อสื่อสารระหว่างผู้ใช้กับเว็บเพจนั้น ต้องติดต่อโดยผู้ใช้ส่งข้อมูลผ่านทางแบบฟอร์มที่มีการเขียน CGI รองรับไว้ในเครื่องคอมพิวเตอร์เซิร์ฟเวอร์เมื่อมีการส่งข้อมูลจากแบบฟอร์มมายังเซิร์ฟเวอร์ โปรแกรม CGI จะประมวลผลข้อมูลทุกครั้ง แล้วนำผลลัพท์ที่ได้ส่งกลับคืนไปยังเครื่องคอมพิวเตอร์ของผู้ใช้ในรูปของเว็บเพจ HTML ใหม่ หากต้องมีการประมวลผลในลักษณะอย่างนี้ซ้ำ ๆ กันหลาย ๆ ครั้ง (อาจเป็นเพราะผู้ใช้หลายคนส่งข้อมูล หรืออาจมีผู้ใช้คนเดียวกันที่สั่งประมวลผลหลายครั้งก็ได้) ซีพียูที่เครื่องเซิร์ฟเวอร์จะทำงานซ้ำ ๆ กันมากเกินความจำเป็นด้วยเหตุนี้จึงมีการเอาจาวาสคริปต์เข้ามาแก้ไขจุดบกพร่องดังกล่าวโดยให้เครื่องคอมพิวเตอร์ของผู้ใช้ประมวลผลเอง เพื่อช่วยลดภาระการทำงานของซีพียูที่เซิร์ฟเวอร์

นอกจากนี้เรายังสามารถเขียนจาวาสคริปต์เป็นกลไกพื้นฐานเพื่อให้โปรแกรมตัดสินใจว่าเว็บเบราเซอร์ที่ผู้ใช้กำลังใช้งานอยู่เป็นของบริษัทใด ควรเรียกใช้โปรแกรมเสริมจำพวกปลั๊กอิน โปรแกรมใดมาเพิ่มเติมหรือไม่ และสุดท้ายเราสามารถใช้จาวาสคริปต์ สร้าง Dynamic HTML ได้ด้วย

จาวาสคริปต์นั้นนอกจากจะทำงานในเว็บเพจได้อย่างมีประสิทธิภาพแล้วยังสามารถนำไปใช้ร่วมกับเทคโนโลยีอื่น ๆ เช่น ActiveX, CGI,Plug-In, Java ฯลฯ เพื่อนำมาเพิ่มประสิทธิภาพในการทำงานให้มากขึ้นได้ด้วย

4.4 หลักการเขียน JavaScript ใน HTML

จาวาสคริปต์เป็นภาษาซึ่งทำหน้าที่เพิ่มขยายขีดความสามารถและคุณสมบัติต่าง ๆ ที่ไม่มีใน HTML ดังนั้นลักษณะคำสั่งของจาวาสคริปต์จึงแตกต่างออกไปจากภาษาคอมพิวเตอร์อื่น ๆ เราสามารถเพิ่มคำสั่งจาวาสคริปต์เข้าไปในเว็บเพจได้ด้วยวิธีการดังต่อไปนี้

- เขียนคำสั่งหรือกำหนดฟังก์ชันไว้ภายในส่วนหัวของเว็บเพจ
- เรียกใช้ฟังก์ชันจากสคริปต์อื่น ๆ ที่เขียนต่อมา
- เรียกใช้ฟังก์ชันโดยค่าแอตริบิวต์ (attribute) ของคำสั่ง HTML

4.4.1 การใช้คำสั่ง <SCRIPT>...</SCRIPT>

คำสั่ง <SCRIPT>...</SCRIPT> (เขียนเป็นตัวอักษรใหญ่หรือเล็ก หรือผสมกันได้) เป็นคำสั่งเพิ่มเติมในภาษา HTML ใช้กำหนดขอบเขตที่แน่นอนของคำสั่งภาษาสคริปต์ เพื่อให้เบราเซอร์สามารถจำแนกแยกแยะได้ว่าส่วนไหนเป็นชุดคำสั่งของภาษาสคริปต์ ส่วนไหนเป็นชุดคำสั่งของ HTML เพราะการทำงานของเบราเซอร์ในชุดคำสั่งทั้งสองแตกต่างกัน การเขียนจาวาสคริปต์มีรูปแบบดังนี้

```
<SCRIPT>
        JavaScript statements...
</SCRIPT>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง <SCRIPT> เป็นจุดเริ่มต้นที่เบราเซอร์จะทำงานตามชุดคำสั่งของ JavaScript ไปเรื่อย ๆ จนกระทั่งพบคำาง </SCRIPT> อันเป็นจุดสิ้นสุดการทำงานในส่วนของสคริปต์ หลังจากนั้นเว็บเบราเซอร์ก็จะอ่านชุดคำสั่งของ HTML ที่อยู่นอกสคริปต์ต่อไป ซึ่งภายในเว็บหนึ่ง ๆ จะใช้คำสั่ง <SCRIPT>... </SCRIPT> เท่าใดก็ได้ไม่จำกัดจำนวน

จาวาสคริปต์ตีความชุดคำสั่งต่าง ๆ ตามการเว้นวรรค และใช้เครื่องหมายอัฒภาค (;) คั่นระหว่างข้อความ และไม่ถือว่าการขึ้นบรรทัดใหม่เป็นเรื่องสำคัญ นอกจากนี้การเขียนรูปประโยคคำสั่งของจาวาสคริปต์นั้นเขียนได้หลายรูปแบบไม่จำเป็นต้องเหมือนกับตัวอย่างที่ผู้เขียนนำมาเสนอเป็นตัวอย่าง ทั้งนี้โดยยึดรูปแบบของประโยคคำสั่งเป็นหลัก เช่น การเขียนฟังก์ชัน สามารถเขียนได้หลายรูปแบบดังนี้

```
function FuctionName{
        Statement;
}
```

หรือ

```
function FuctionName{    Statement;  }
```

หรือ

```
function FuctionName
    {  Statement;  }
```

แม้ว่าการเขียนรูปประดยคของฟังค์ชันจะแตกต่างกัน แต่ก็สามารถทำงานได้เป็นปกติเหมือนกัน

4.4.2 การระบุภาษาและรุ่นของสคริปต์

เราสามารถกำหนดรูปแบบคำสั่งของจาวาสคริปต์ได้ด้วยการเพิ่มแอตทริบิวต์ Language เข้าไปในคำสั่ง <SCRIPT> เพื่อกำหนดชนิดของภาษาสคริปต์ที่ใช้งานว่าต้องการให้เป็นภาษาสคริปต์ภาษาใด ซึ่งในปัจจุบันมีที่นิยมใช้อยู่ด้วยกันสองภาษา คือ JavaScript และ VBScript โดยเขียนตามรูปแบบโครงสร้างดังนี้

นับตั้งแต่จาวาสคริปต์ออกสู่ท้องตลาด ได้ผ่านการพัฒนาปรับปรุงมาแล้ว 3 รุ่น คือ 1.0, 1.1 และ 1.2 เราสามารถระบุเวอร์ชันของจาวาสคริปต์ได้ด้วยการกำหนดหมายเลขเวอร์ชันลงไปในชุดคำสั่งดังนี้

```
<SCRIPT Language="JavaScrip">... </SCRIPT>
การระบุหมายเลขรุ่นจาวาสคริปต์ สำหรับ Navigator 2.0
<SCRIPT Language="JavaScrip1.1">... </SCRIPT>
การระบุหมายเลขรุ่น JavaScript สำหรับ Navigator 3.0
<SCRIPT Language="JavaScrip1.2">... </SCRIPT>
การระบุหมายเลขรุ่นจาวาสคริปต์สำหรับ Communicator 4.0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเลขเวอร์ชันต้องเขียนติดกับคำว่าจาวาสคริปต์เลยโดยไม่มีการเว้นวรรคให้เกิดช่องว่างใด ๆ ทั้งสิ้น เช่น JavaScript1.1 หรือ JavaScript1.2

โปรแกรม Netscape Navicator 2.x สามารถอ่านชุดคำสั่งที่กำหนดโดย <SCRIPT Language="JavaScrip"> ได้ แต่จะไม่ทำงานตามสคริปต์ในชุดคำสั่งแบบ <SCRIPT Language="JavaScrip1.1"> หรือ <SCRIPT Language="JavaScrip1.2">

ขณะที่โปรแกรม Netscape Navicator 3.x สามารถทำงานกับชุดคำสั่งได้ทั้งแบบ <SCRIPT Language="JavaScrip"> และ <SCRIPT Language="JavaScrip1.1"> แต่จะไม่ทำงานกับสคริปต์แบบ <SCRIPT Language="JavaScrip1.2">

ส่วนโปรแกรม Netscape Navicator 4.x สามารถมองเห็นจาวาสคริปต์ได้ทั้งเล่น 1.0, 1.1, 1.2

ตัวอย่าง การกำหนดฟังก์ชันที่แสดงได้ทั้งจาวาสคริปต์ 1.0 และ จาวาสคริปต์ 1.1

```
<SCRIPT Language="JavaScript">
        // Define 1.0-compatible functions such as doClick() here
</SCRIPT>

<SCRIPT Language="JavaScript1.1">
        // Redefine those functions using 1.1 features
        // Also define 1.1-only functions
</SCRIPT>
```

หากสังเกตุดูเว็บเพจที่มีการใช้งานจาวาสคริปต์ มักจะพบแต่คำสั่ง <SCRIPT Language="JavaScript"> เท่านั้น โดยไม่ระบุรุ่น 1.0, 1.1, 1.2 เลยกรณีเช่นนี้แสดงว่าเป็นการเขียนด้วยจาวาสคริปต์รุ่น 1.0 แต่ไม่ว่าจะเขียนหรือไม่เขียนก็ตาม สำหรับ Netscape Navicator หรือ Netscape Communicator ตั้งแต่รุ่น 4.0 เป็นต้นไป จะไม่มีผลต่อการทำงานของโปรแกรมอย่างใด และชุดคำสั่งแบบนี้สามารถใช้ได้กับเว็บเบราเซอร์ของเน็ตสเคปทุกรุ่น แต่การเขียนโดยไม่ระบุหมายเลขรุ่นเช่นนี้อาจทำให้เกิดปัญหาเมื่อมรการเรียกดูเว็บเพจในเบราเซอร์รุ่นที่ต่ำกว่าได้ เช่น เขียนจาวาสคริปต์ด้วยข้อกำหนดของรุ่น 1.2 จะทำงานได้ไม่ถูกต้องใน Navicator 3.x เป็นต้น

4.4.3 JavaScript กับ Internet Explorer

ภาษาคอมพิวเตอร์ทั่วไปนั้น เมื่อได้รับความนิยมและมีการนำไปใช้งานกันแพร่หลายด้วยความนิยมนั้นก็มักจะทำให้รูปแบบภาษาเกิดความผิดเพี้ยนไป เพราะบริษัทผู้ผลิตซอฟท์แวร์มักจะดัดแปลงภาษาให้สามารถทำงานได้กับเอ็น engine ที่บริษัทนั้นผลิตขึ้นมา ภาษาที่เคยเป็นหนึ่งเดียวใช้งานได้อย่างมาตรฐาน จึงเริ่มไม่เป็นมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหานี้เกิดขึ้นกับที่นิยมใช้กันอยู่ไม่ว่าจะเป็น C, C++, Pascal, Basic ดังนั้นเพื่อเป็นการปกป้องภาษาจาวาให้เป็นหนึ่ง บริษัทซันจึงได้กำหนดมาตรฐานห้ามดัดแปลงภาษาจาวาโดยเด็ดขาด ทำให้เรามั่นใจได้เลยว่าการเขียนโปรแกรมด้วยภาษาจาวาจะยังคงเป็นหนึ่งเดียวต่อไป

และหลังจากที่ไมโครซอฟท์วางตลาด Internet Explorer 3.0 (IE) การแข่งขัน การแข่งขันก็เริ่มรุนแรงขึ้น ทั้งนี้เพราะ IE เป็นโปรแกรมเว็บเบราเซอร์ที่มีประสิทธิภาพมาก มีคุณสมบัติพิเศษต่าง ๆ เพิ่มขึ้น อาทิเช่น การสนับสนุนภาษา จาวา, ActiveX, สนับสนุนการสร้างเฟรมแบบไม่แสดงเส้นขอบและลูกเล่นอื่น ๆ อีกมากมาย รวมทั้งสิ่งหนึ่งที่ขาดไม่ได้คือการสนับสนุนชุดคำสั่งของจาวาสคริปต์ แต่ก็เป็นการสนับสนุนแบบไม่สมบูรณ์เท่าใดนัก เนื่องจาก IE มีขีดความสามารถในการรองรับชุดคำสั่งของรุ่น 1.1 หรือสูงกว่ามาเรียกดูผ่าน IE โปรแกรมจะแสดงความผิดผลาดขึ้น

จาวาสคริปต์ใช้ค่าตัวแปร userAgent จากอ็อบเจ็กต์ navigator (โดยเขียนได้เป็น navigator.user.Agent) สำหรับตรวจสอบชนิดของเว็บเบราเซอร์ที่ใช้งานอยู่ ถ้าพบคำว่า “MSIE” แสดงว่าเป็น Internet Exploror ของบริษัทไมโครซอฟท์ แต่ถ้าพบ “Moszilla” แสดงว่าเป็น Navigator ของบริษัทเน็ตสเคปทำให้เราสามารถใช้การทดสอบเงื่อนไขนี้เป็นทางเลือกภายในสคริปต์คำสั่งเพื่อกำหนดให้โปรแกรมทำงานหรือไม่ทำงานกับเว็บเบราเซอร์ IE ดังตัวอย่างต่อไปนี้

```
<SCRIPT Language=”JavaScript”>
        var uagent=navigator.userAgent;
        if (uagent.indexOF(“MSIE”) == 25) { alert (“You are using Microsoft Internet
Exploror”) }
        else { alert (“You are not using Microsoft Internet Exploror”) }
</SCRIPT>
```

นอกเหนือจากจาวาสคริปต์แล้ว IE ยังสนับสนุนที่ไมโครซอฟท์พัฒนาขึ้นมาเองคือ VBScript ด้วย ทำให้การสนับสนุนจาวาสคริปต์ของ IE ยิ่งลดระดับความสำคัญลงไปอีกดังนั้นการนำจาวาสคริปต์มาใช้กับ IE จึงมีขีดจำกัดมากยิ่งขึ้น

ดังนั้นถ้าต้องการเขียนจาวาสคริปต์เพื่อป้องกันมิให้ถูกใช้งานโดย IE ซึ่งอาจทำงานไม่ถูกต้อง ให้ระบุรุ่นของจาวาสคริปต์ลงไปด้วย เช่น <SCRIPT Language=”JavaScript1.1”> เพราะ IE จะไม่สนใจคำสั่งจาวาสคริปต์ในขอบเขตที่ระบุเป็น JavaScript1.1 เลย แต่สามารถทำงานได้เป็นอย่างดีกับชุดคำสั่งที่อยู่ระหว่างคำสั่ง <SCRIPT Language=”VBScript”> และในทำนองเดียวกัน โปรแกรม Netscape Navigator ไม่รู้จักกับชุดคำสั่งของ VBScript เลย และไม่สนใจกับชุดคำสั่งดังกล่าวด้วย

4.4.4 ตำแหน่งของ SCRIPT

คำสั่ง <SCRIPT>...</SCRIPT> เป็นการกำหนดบริเวณพื้นที่และขอบเขตของการสั่งงานด้วยชุดคำสั่งของจาวาสคริปต์ซึ่งเราไม่สามารถเขียนชุดคำสั่งของจาวาสคริปต์นอกเหนือขอบเขตดังกล่าวได้ แต่ถ้าเผลอนำคำสั่งจาวาสคริปต์ไปเขียนไว้นอกขอบเขตที่กำหนด HTML ไปเขียนอยู่ภายในขอบเขตของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง <SCRIPT>...</SCRIPT> โปรแกรมจะแสดงข้อความขึ้นมา ยกเว้นกรณีที่นำเอาชุดคำสั่ง HTML นั้น ๆ ไปใช้รวมกับชุดคำสั่งตามแบบของจาวาสคริปต์

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

ตัวอย่างการเขียนชุดคำสั่งของจาวาสคริปต์ภายในส่วน <HEAD>...</HEAD>

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

หรือ

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

ตัวอย่างการเขียนสคริปต์ทั้งสองชุดให้ผลลัพท์ออกมาเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตัดสินใจเขียนจาวาสคริปต์ไว้ภายในส่วนใดนั้นขึ้นอยู่กับความพึงพอใจของผู้เขียนเป็นหลัก แต่ถ้าประเมินความนิยมแล้วพบว่านิยมเขียนไว้ในส่วน <HEAD>...</HEAD> มากกว่าเพราะมีข้อดีคือ ชุดคำสั่งและฟังก์ชันต่างที่เขียนนั้น จะถูกโหลดไว้ในหน่วยความจำก่อนการเกิด HTML ส่วนอื่น ๆ เมื่อเกิดเหตุการ (even) ต่าง ๆ ตามมาโดยการกำหนด " อีเวนต์แฮนด์เลอร์" (even handler) ไว้เช่น onLoad ในส่วน <BODY> (หมายถึงการกระตุ้นให้สคริปต์ทำงานเมื่อเว็บเพจถูกโหลด) หากเราเขียนสคริปต์ไว้ในส่วน <BODY>...</BODY> จะไม่สามารถใช้อีเวนต์แฮนด์เลอร์นี้ได้

นอกจากนี้ยังนิยมเขียน จาวาสคริปต์ภายในส่วน <BODY>...</BODY> สำหรับใช้งานร่วมกับแบบฟอร์ม <FORM>...</FORM> โดยกำหนดตำแหน่งอีเวนต์แฮนด์เลอร์ไว้กับส่วนประกอบต่าง ๆ ของฟอร์ม

ตัวอย่าง การกำหนดและการเรียกใช้ฟังก์ชันในจาวาสคริปต์

```
<HTML>
<HEAD>
        <SCRIPT LANGUAGE="JavaScript">
<!-- hide SCRIPT from old browsers
                function bar() {
                        document.write("<HR ALIGN='left' WIDTH=25%>")
                }
                function output(head, level, string) {
                        document.write("<H" + level + ">" + head + "</H" + level + ">
<P>" + string)
                }
                // end hiding from old browsers -->
        </SCRIPT>
</HEAD>
<BODY>
<SCRIPT>
<!-- hide SCRIPT from old browsers
                document.write(bar(),output("Make Me Big",3,"Make me ordinary."))
                // end hiding from old browsers -->
</SCRIPT>
<P>Thanks.

</BODY>
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</HTML>
```

ซึ่งจะได้ผลลัพท์ในหน้าต่างของเบราเซอร์

```
Make Me Big
Make me ordinary.undefinedundefined
Thanks.
```

คำสั่ง Client-Side JavaScript ที่ถูกบรรจุลงในไฟล์เว็บเพจ HTML สามารถสนองความต้องการของผู้ใช้ได้ทันที โดยเฉพาะเมื่อใช้ร่วมกับแบบฟอร์มป้อนข้อมูลลงในแบบฟอร์ม ไม่ว่าจะเป็น ชื่อ, ที่อยู่, หมายเลขโทรศัพท์, หมายเลขแฟกซ์ หรือ e-mail เป็นต้น

ชุดคำสั่งจาวาสคริปต์จะถูกเบราเซอร์แปลและทำงานทีละบรรทัดตามลักษณะการทำงานของภาษาแบบอินเตอร์พรีเตอร์ คำสั่งใดเขียนก่อนก็จะถูกแปลไปใช้งานก่อน แต่สำหรับบางคำสั่งจะทำงานหรือได้ก็ต่อเมื่อมี “เหตุการณ์” เกิดขึ้นก่อน เช่น ต้องคลิ๊กที่ปุ่มก่อน จึงจะทำงานต่อไปได้ เป็นต้น

ตัวอย่าง การทำงานของสคริปต์ตามลำดับก่อน-หลัง

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
        document.write("Hello net.")
</SCRIPT>
</HEAD>

<BODY>
That's all,Folks.
</BODY>
</HTML>
```

ซึ่งจะได้ผลลัพท์ในหน้าต่างของเบราเซอร์

```
Hello net. That's all,Folks.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 เว็บเบราเซอร์ที่ไม่สนับสนุน JavaScript

เนื่องจากจาวาสคริปต์ทำงานได้ดีเฉพาะเบราเวอร์ที่สนับสนุนเท่านั้น ฉะนั้นเ ระมัดระวังกัน ความผิดผลาดอันเกิดจากการ ไม่สนับสนุนโครงสร้างภาษาของจาวาสคริปต์เราจึงควรเริ่ม ด้วย เครื่องหมาย <!-- และปิดท้ายด้วยเครื่องหมาย -->

เบราเซอร์ที่ไม่สนับสนุนจาวาสคริปต์ จะมองไม่เห็นชุดคำสั่งที่อยู่ระหว่างเครื่องหมาย <!-- และ --> เพราะจะตีความว่าข้อความเหล่านั้นเป็น หมายเหตุ (comment) ตามรูปแบบคำสั่งกำหนดหมายเหตุของ HTML (<!--…-->)

4.6 หมายเหตุในจาวาสคริปต์

การเขียนหมายเหตุในจาวาสคริปต์มี 2 ลักษณะดังนี้

1.เขียนหมายเหตุจบในบรรทัดเดียว

โดยใช้เครื่องหมาย // กำกับไว้ที่หน้าข้อความนั้น โปรแกรมจะตีความว่าข้อความที่ตาหลังเครื่องหมาย // เป็นหมายเหตุทั้งหมดจนสุดบรรทัด หมายความว่าไม่ต้องสนใจว่าข้อความที่ตามหลังเครื่อหมาย // จะเป็นอะไร ให้ข้ามไปทำงานในบรรทัดถัดไปได้เลย

2.เขียนหมายเหตุโดยไม่จำกัดบรรทัดหรือความยาว

ให้ใช้เครื่องหมาย /* นำหน้าและเครื่อหมาย */ ปิดท้ายข้อความที่เป็นหมายเหตุ ส่วนใหญ่มักใช้ในระหว่างการทดสอบโปรแกรมที่ต้องมีการละเว้นการแปลข้อความบางบรรทัด หรือบางส่วนของโปรแกรม และมิให้เบราเซอร์แสดงข้อความนั้นบนจอภาพ

ตัวอย่างการใช้หมายเหตุทั้งสองแบบข้อความที่ถูกระบายพื้นหลังไว้คือหมายเหตุไม่ถูกนำไปแปลความแต่อย่างใด

```
<HTML><HEAD>   /* This is simple page
to show comment format of JavaSccipt
message in highlight is comment*/
<SCRIPT LANGUAGE="JavaScript">
        document.write("Hello net.")  // print Hello net
</SCRIPT>
</HEAD>
<BODY>
That's all,Folks.
</BODY></HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การสร้าง และทำงานกับไฟล์ฐานข้อมูล

5.1 การสร้างตารางและฟิลด์

เมื่อสร้างไฟล์ฐานข้อมูลขึ้นมาได้แล้ว ขั้นต่อไปก็จะมาเริ่มต้นสร้างตารางซึ่งได้ออกแบบไว้ ขึ้นมาในไมโครซอฟท์แอคเซส 97 เนื่องจากตารางเป็นส่วนประกอบพื้นฐานที่ใช้เก็บข้อมูลที่จะนำมาใช้ในแอพลิเคชั่นฐานข้อมูล

การสร้างตารางใน ไมโครซอฟท์แอคเซส 97 นี้ ต้องสร้างตารางจากมุมมองเทเบิล-ดีไซนด์ (Table Design) ซึ่งจะมีเครื่องมือที่ใช้ในการกำหนดโครงสร้างต่าง ๆ ของตาราง เช่น การกำหนดคุณสมบัติต่าง ๆ ของฟิลด์ในตาราง กำหนดคีย์หลัก เป็นต้น

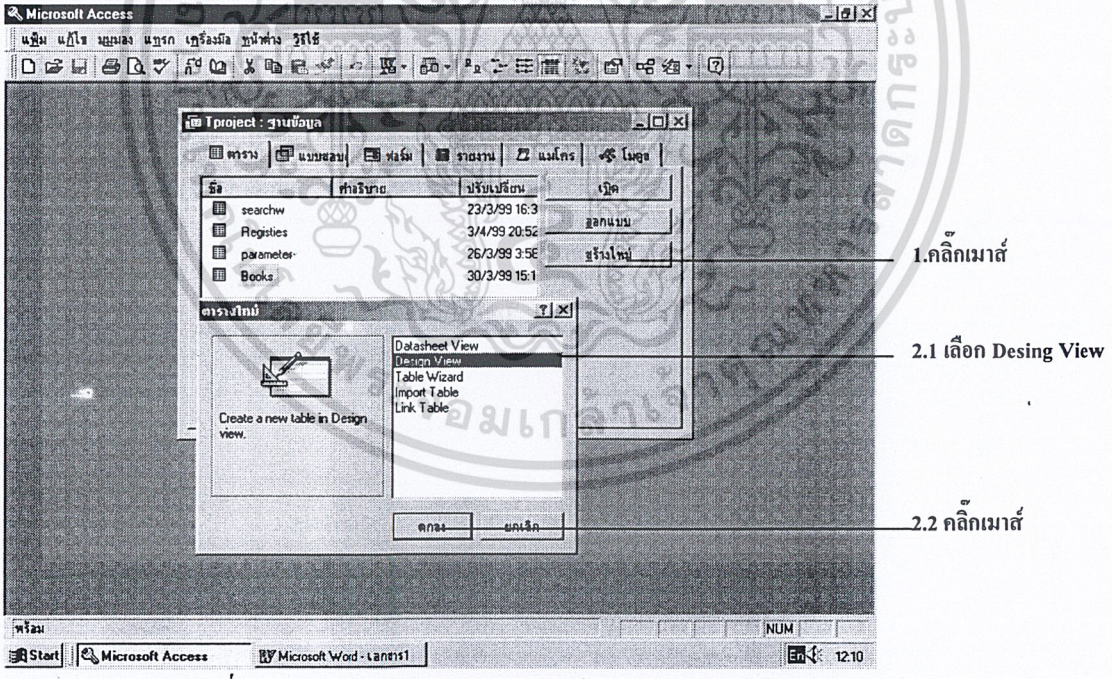
5.2 การเข้าสู่มุมมอง เทเบิล-ดีไซนด์

การเข้าสู่มุมมอง เทเบิล-ดีไซนด์ให้ทำตามขั้นตอนดังต่อไปนี้

1.จากหน้าต่าง ดาต้าเบส ให้ คลิ๊กเมาส์ ปุ่ม New

2.จะปรากฏไดอะลอกซ์ New Table ให้เราเลือก Design View เสร็จแล้ว คลิ๊กเมาส์ ปุ่ม OK

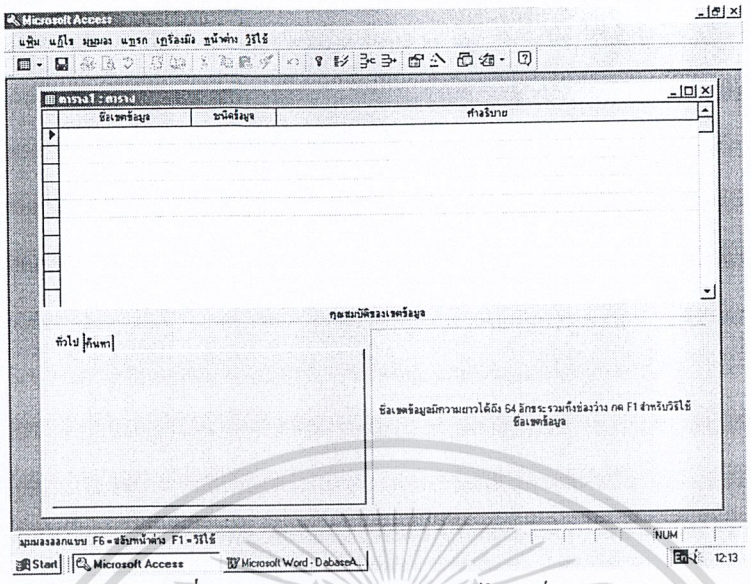
3.จะปรากฏมุมมอง เทเบิล-ดีไซนด์ขึ้นมา.....



รูปที่ 5-1แสดงการเข้าสู่ เทเบิล-ดีไซนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

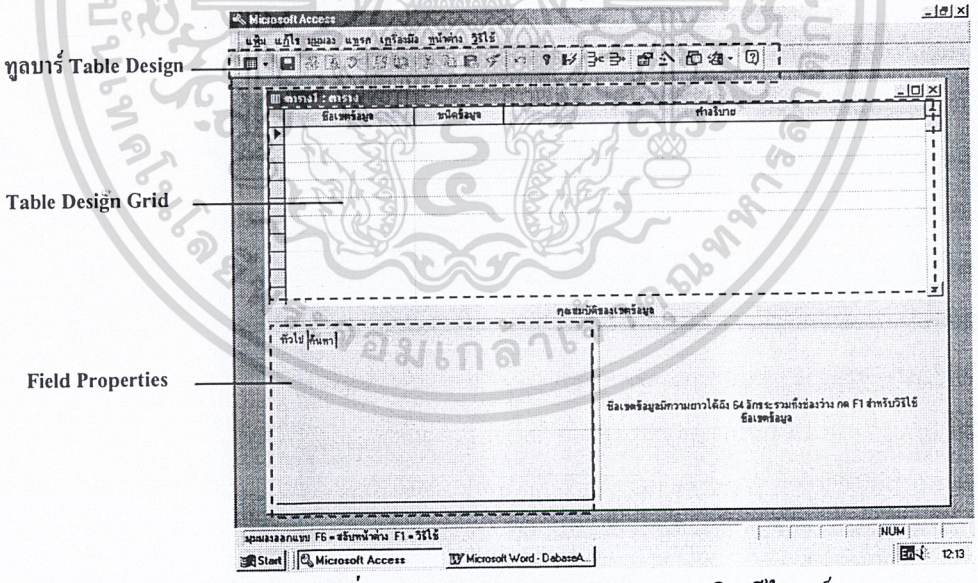
3. ปรากฏมุมมองTable Design



รูปที่ 5-2 แสดงหน้าต่างเทเบิล-ดีไซนด์

5.3 ส่วนประกอบต่างๆในมุมมอง เทเบิล-ดีไซนด์

พอเข้ามาในมุมมอง เทเบิล-ดีไซนด์แล้ว จะปรากฏ หน้าต่างที่มีส่วนประกอบต่างๆดังรูปที่5-3



รูปที่ 5-3 ส่วนประกอบของมุมมองเทเบิล-ดีไซนด์

รายละเอียดในส่วนประกอบต่างๆของหน้าต่างข้างต้นมีดังต่อไปนี้

Table Design Grid มีคอลัมน์อยู่ 3 คอลัมน์ที่ใช้กำหนดชื่อ (Field Name) ชนิดข้อมูลของฟิลด์ (Data Type) และอธิบายฟิลด์เพิ่มเติม (Field Description)

Field Properties ใช้ในการกำหนดคุณสมบัติต่างๆเพิ่มเติมให้กับฟิลด์ เช่น รูปแบบในการแสดงผล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table Design Toolbar จะเป็นทูลบาร์ที่ประกอบด้วยปุ่มคำสั่งที่ใช้ในการทำงานในมุมมองนี้

5.4 การสร้าง และกำหนดคุณสมบัติพื้นฐานของฟิลด์

ในหัวข้อนี้จะเริ่มต้นนำการสร้างตาราง ด้วยการสร้างฟิลด์แต่ละฟิลด์ในตารางขึ้นมาในมุมมอง Table Design โดยจะใส่ชื่อฟิลด์ และกำหนดคุณสมบัติพื้นฐานของฟิลด์ คือ ชนิดของข้อมูล (Data Type) และขนาดของฟิลด์ (Field Size) การกำหนดคุณสมบัติต่างๆเหล่านี้ เพื่อจุดประสงค์ต่างๆต่อไปนี้

- เพื่อให้มีประสิทธิภาพในการทำงานมากที่สุด ทั้งด้านเนื้อที่ที่ใช้ในการเก็บข้อมูล และความเร็วในการทำงานกับข้อมูลเช่น ข้อมูลชนิดข้อความโดยทั่วไปจะเสียเนื้อที่ในการเก็บมากกว่า ข้อมูลชนิดตัวเลข และเลขจำนวนเต็มจะใช้เวลาในการคำนวณน้อยกว่าเลขทศนิยมเป็นต้น
- เพื่อให้ตรงกับจุดประสงค์ในการใช้ข้อมูล ฟิลด์ราคาสินค้าจะต้องเป็นข้อมูลชนิดตัวเลขที่ใช้ในการคำนวณได้ ไม่ใช่ใช้ข้อมูลชนิดข้อความมาเก็บ ซึ่งนอกจากคำนวณไม่ได้แล้ว ยังสิ้นเปลืองเนื้อที่ในการเก็บข้อมูลอีกด้วย

ชนิดของข้อมูล	คำอธิบาย
Text	เป็นข้อความที่เก็บตัวอักขระได้ไม่เกิน 255 ตัว จำนวนอักขระที่สามารถเก็บได้สูงสุด จะต้องกำหนดในคุณสมบัติ Field Size ชนิดข้อมูลนี้ใช้เก็บค่าของฟิลด์ชื่อสินค้าได้
Memo	เป็นข้อความที่เก็บตัวอักขระได้สูงสุดถึง 64,000 ตัว ส่วนใหญ่ใช้เก็บข้อความที่มีความยาวมากๆ
Number	เป็นข้อมูลชนิดที่ใช้เก็บค่าตัวเลขที่สามารถกำหนดให้เป็นเลขจำนวนเต็มหรือทศนิยมก็ได้ โดยจะกำหนดในคุณสมบัติ Field Size
AutoNumber	เป็นข้อมูลชนิดตัวเลขที่จะกำหนดค่าให้อัตโนมัติ สำหรับแต่ละเรคอร์ดที่เพิ่มเข้าไปในตารางอาจจะโดยการเพิ่มค่า หรือสุ่มเอาแล้วแต่คุณสมบัติ New Values ชนิดข้อมูลนี้เหมาะสำหรับสร้างฟิลด์ที่เป็นคีย์หลัก
Yes/No	เป็นข้อมูลชนิดบูลีน ที่มีเพียงค่าใดค่าหนึ่งเท่านั้นในค่า 2ค่า เราสามารถกำหนดรูปแบบในการแสดงผลเป็น True/Flase, Yes/No หรือ On/Off ก็ได้
Currency	เป็นชนิดข้อมูลทศนิยมที่มีตำแหน่งหลังจุดทศนิยม 4 ตำแหน่งเสมอ จึงเหมาะที่จะเก็บค่าเงินที่ต้องการความถูกต้องแม่นยำ
Date/Time	เป็นชนิดข้อมูลที่เก็บเวลา และวันที่
OLE Object	เป็นชนิดข้อมูลที่ใช้เก็บออบเจ็กต์ เช่น รูปภาพ เสียง หรือ ออบเจ็กต์ที่สร้างจากโปรแกรมที่สนับสนุน OLE ก็ได้ เป็นต้น
Hyperlink	เป็นชนิดของข้อมูลที่เก็บที่อยู่ของไฟล์ในระบบเครือข่ายอินเตอร์เน็ต หรืออินทราเน็ต

ตารางที่ 5-1 แสดงรายละเอียดของชนิดข้อมูล(Data Type) ที่มีในไมโครซอฟท์แอคเซส97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากการกำหนดชนิดของข้อมูลที่เหมาะสมให้กับฟิลด์แล้วเรายังจะต้องกำหนดคุณสมบัติเพิ่มเติมบางอย่างให้กับฟิลด์ ซึ่งมีคุณสมบัติที่เราจะกล่าวถึงคือ คุณสมบัติ Field Size ที่ใช้กำหนดขนาดของฟิลด์ คุณสมบัตินี้จะอยู่ในส่วนที่ 2 (ที่เราเรียกว่า Field Properties ซึ่งอยู่ในหัวข้อส่วนประกอบต่างๆในมุมมอง Table Design) คุณสมบัตินี้จะมีกับข้อมูลชนิด Text และ Number เท่านั้น สำหรับข้อมูลชนิด Text จะเป็นการกำหนดตัวอักขระที่สามารถเก็บได้สูงสุด ส่วนข้อมูลชนิด Number จะให้เราเลือกชนิดข้อมูลย่อยซึ่งมีรายละเอียดดังตารางต่อไปนี้

ชนิดข้อมูลย่อย	รายละเอียด
Byte	เป็นเลขจำนวนเต็มที่มีค่าตั้งแต่ 0-255
Integer	เป็นเลขจำนวนเต็มที่มีค่าตั้งแต่ –32,768 ถึง +32,767
Long Integer	เป็นเลขจำนวนเต็มที่มีค่าตั้งแต่ –2,147,483,648 ถึง +2,147,483,647
Single	เป็นเลขทศนิยมที่มีค่าตั้งแต่ $-3.4 * 10^{38}$ ถึง $+3.4 * 10^{38}$
Double	เป็นเลขทศนิยมที่มีค่าตั้งแต่ $-1.797 * 10^{308}$ ถึง $+1.797 * 10^{308}$
Replication ID	เป็นข้อมูลที่ใช้ในเรื่อง Replication

ตารางที่ 5-2 แสดงรายละเอียดคุณสมบัติ Field Size ของ Field ที่มีชนิดข้อมูลเป็น Number

การเลือกชนิดข้อมูล และขนาดเป็นสิ่งจำเป็น ของให้เราเลือกให้เหมาะสมกับข้อมูลจริงที่เราจะใช้เก็บ โดยใช้ตารางรายละเอียดของข้อมูลชนิดต่างๆที่ผ่านมาแล้วช่วยในการตัดสินใจ

5. 5 ขั้นตอนการสร้างฟิลด์แบบพื้นฐาน

เป็นการกำหนดคุณสมบัติต่างๆของฟิลด์ ซึ่งมีขั้นตอนดังต่อไปนี้

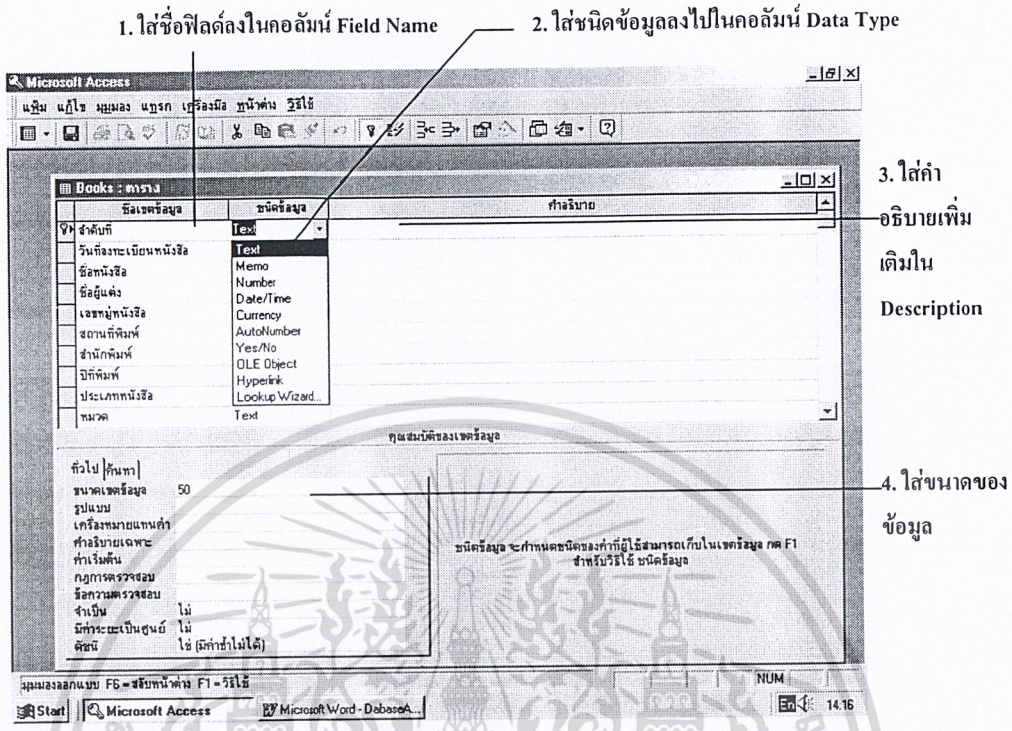
1.จากหน้าต่าง เทเบิล-ดีไซน์ ให้ใส่ชื่อฟิลด์ในคอลัมน์ Field Name ใน Table Design Grid

2.เลือกชนิดข้อมูลที่เหมาะสม จากคอลัมน์ Data Type ใน Table Design Grid

3.ใส่ข้อความอธิบายเพิ่มเติมได้ในคอลัมน์ Description ใน Table Design Grid

4.กำหนดคุณสมบัติ Filed Size ให้กำหนดฟิลด์ที่มีชนิดข้อมูลเป็น Text หรือ Number ในส่วน Field Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

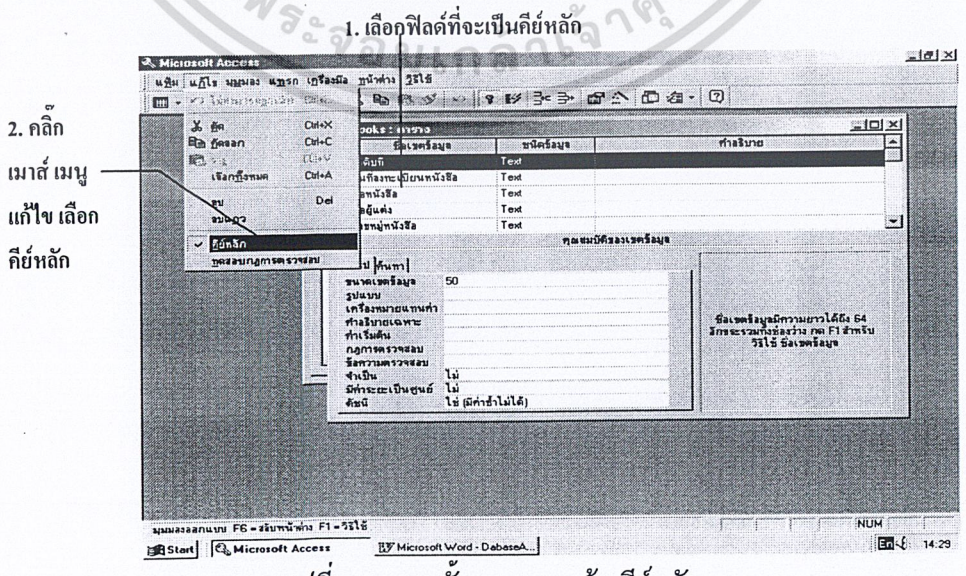


รูปที่ 5-4 แสดงขั้นตอนการสร้างฟิลด์

5.6 การกำหนดคีย์หลัก (Primary Key) ของตาราง

คีย์หลักเป็นส่วนประกอบของตารางที่สำคัญมาก เมื่อเราสร้างฟิลด์ต่างๆเสร็จแล้ว ต่อไปเราจะมากำหนดฟิลด์ที่จะเป็นคีย์หลักของตาราง

การกำหนดคีย์หลักของตาราง ให้เราเลือกฟิลด์ที่จะกำหนดเป็นคีย์หลัก เสร็จแล้วให้เรา Click Mouse เมนู Edit>Primary Key



รูปที่ 5-5 แสดงขั้นตอนการสร้างคีย์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การใส่ข้อมูลในตาราง

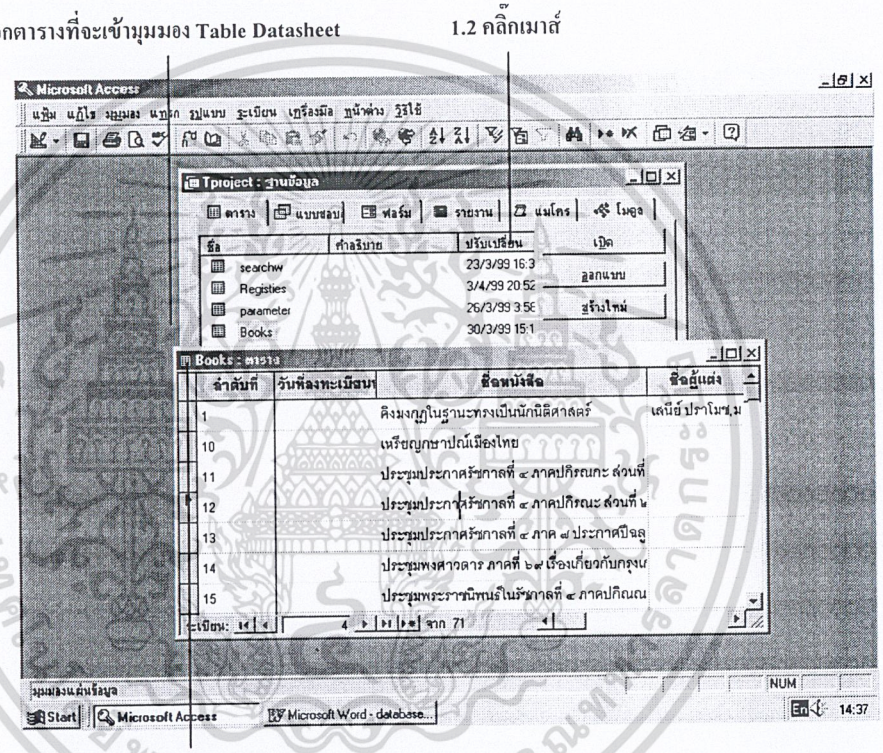
ในการใส่ข้อมูลลงในตารางที่เราสร้าง เราจะทำมุมมอง Table Datasheet ที่เป็นมุมมองที่เราจะใช้ในการทำงานกับข้อมูลในตารางได้หลายอย่าง เช่น การเพิ่ม แก้ไข และลบข้อมูลในตาราง เป็นต้น

การเข้าสู่มุมมอง Table Datasheet

การเข้าสู่มุมมอง Table Datasheet ทำตามขั้นตอนดังต่อไปนี้

1.จากหน้าต่าง Datasheet ให้เราเลือกตารางที่จะเข้ามุมมอง Table Datasheet เสร็จแล้ว Click mouse ปุ่ม Open

2.จะปรากฏมุมมอง Table Datasheet ขึ้นมา



รูปที่ 5-6 แสดงการใส่ข้อมูลในตาราง

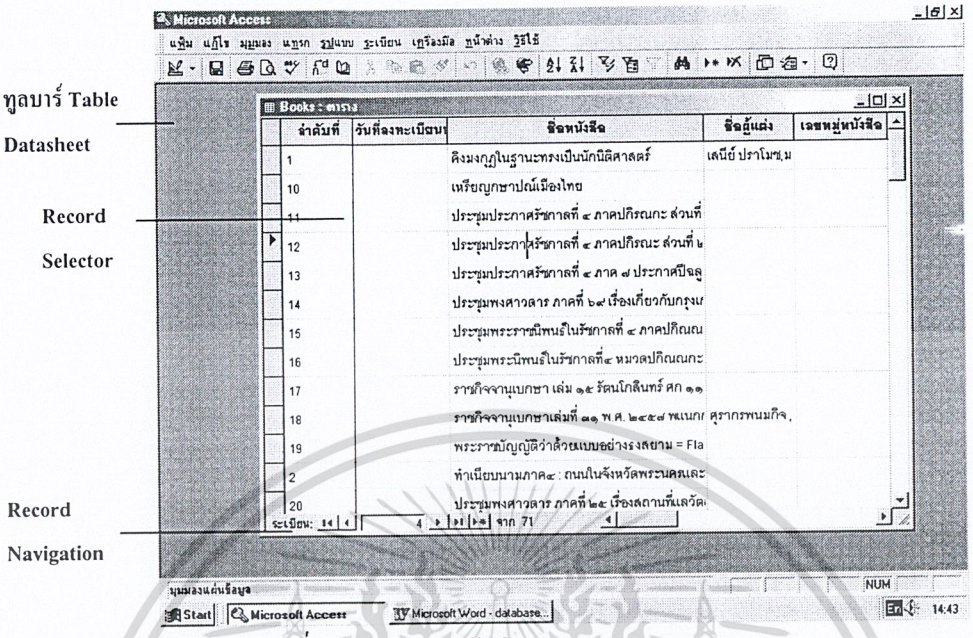
5.8 การทำงานกับเรคอร์ดในมุมมอง Table Datasheet

5.8.1 ส่วนประกอบต่างๆในมุมมอง Table Datasheet

มุมมอง Table Datasheet จะแสดงข้อมูลในตารางในลักษณะแถว และคอลัมน์โดยแถวจะเป็นเรคอร์ด แต่ละเรคอร์ด คอลัมน์จะเป็นฟิลด์แต่ละฟิลด์ โดยเรียงลำดับฟิลด์เหมือนกับที่เรากำหนดในมุมมอง Table Design และเรียกช่องแต่ละช่องในตารางว่าเซลล์ (Cells)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบต่างๆในมุมมอง Table Datasheet จะเป็นดังรูปที่ 5-7



รูปที่ 5-7 แสดงการเข้าสู่มุมมอง Table Datasheet

1.Record Selector จะใช้ในการเลือกเรคอร์ดแต่ละเรคอร์ดในตาราง เมื่อ click mouse ที่ปุ่มนี้จะเป็นแถบสีดำเลือกเรคอร์ดนั้นอยู่

2.Record Navigator จะใช้ในการไปยัง Record ที่ต้องการมีปุ่มต่างๆที่มีรายละเอียดดังต่อไปนี้

- ปุ่ม > ใช้ในการเลื่อนไปยังเรคอร์ดก่อนหน้าเรคอร์ดปัจจุบัน 1 เรคอร์ด
- ปุ่ม < ใช้ในการเลื่อนไปยังเรคอร์ดหลังจากเรคอร์ดปัจจุบัน 1 เรคอร์ด
- ปุ่ม |< ใช้ในการเลื่อนไปยังเรคอร์ดแรกสุดของตาราง
- ปุ่ม >| ใช้ในการเลื่อนไปยังเรคอร์ดสุดท้านของตาราง
- ปุ่ม >* ใช้ในการเพิ่มเรคอร์ดใหม่ลงไปในตาราง

นอกจากนี้เรายังสามารถไปยังเรคอร์ดที่ต่องการ ได้ด้วยการคีย์หมายเลขของเรคร์ดที่ต้องการ ลงใน Record Navigator ได้ และกดปุ่ม <Enter> เพื่อไปยังเรคอร์ดที่มีหมายเลขตรงกับที่เราใส่ไป

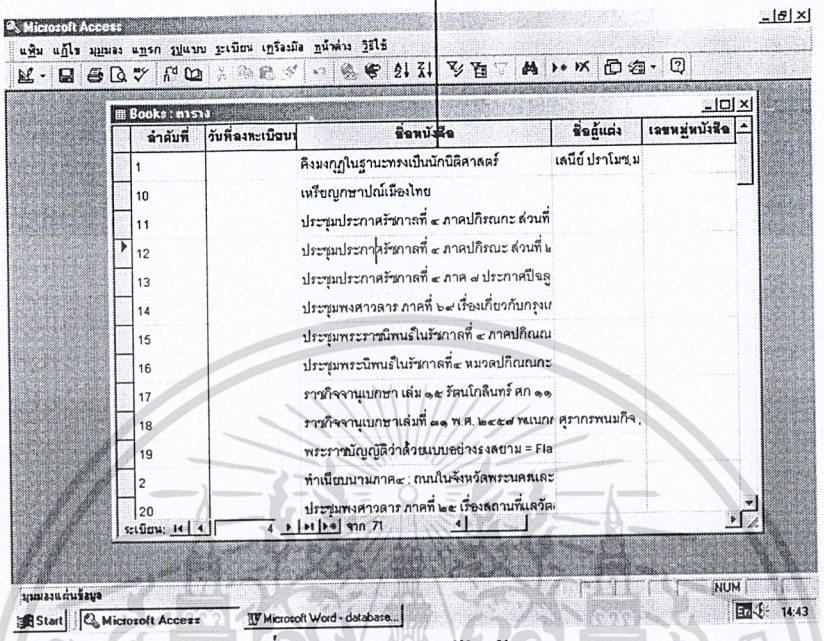
3.Table Datasheet Toolbar จะเป็นทูลบาร์ที่ใช้เก็บคำสั่งต่างๆที่ใช้งานในมุมมอง Table Datasheet

5.8.2 การเพิ่มเรคอร์ดลงในตาราง

การเพิ่มเรคอร์ดลงในตาราง ให้กดปุ่มลูกศรซ้ายขวาเลื่อนไปมาระหว่างฟิลด์ต่างๆในเรคอร์ดปัจจุบัน และพิมพ์ข้อมูลลงไป ถ้าต้องการไปยังเรคอร์ดอื่น ให้เลื่อนปุ่มลูกศรขึ้นลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

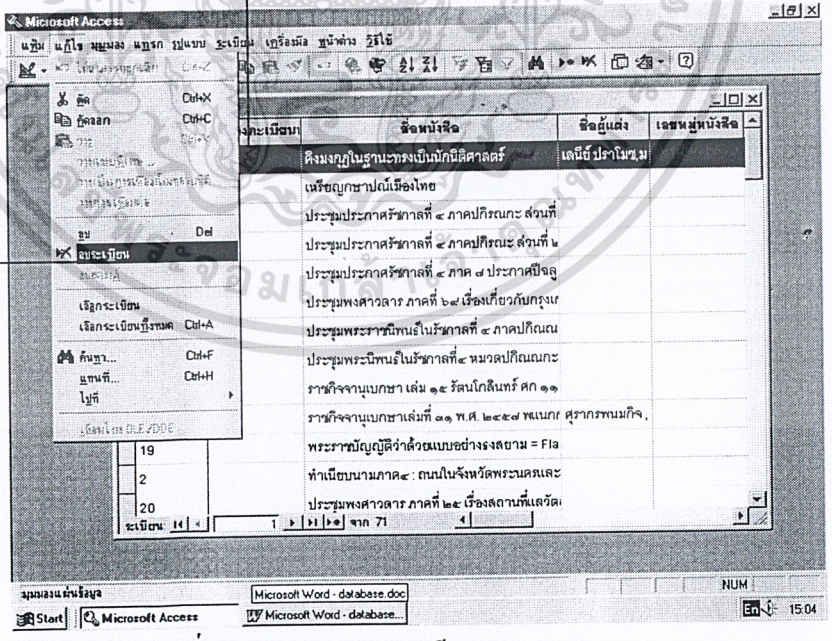
1. กดปุ่มทิศทางเพื่อไปฟิลด์ และเรคอร์ดที่ต้องการแก้ไข แล้วพิมพ์ข้อมูลลงไปใหม่



รูปที่ 5-9 แสดงการแก้ไขข้อมูล

2. คลิ๊กเมาส์ที่ Record Selector ของเรคอร์ดที่จะเลือก

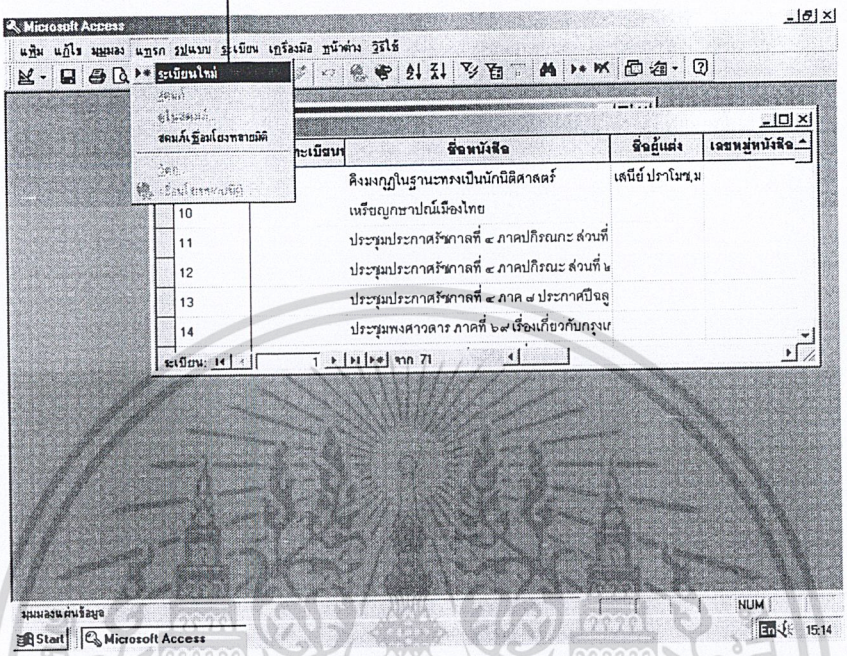
3. คลิ๊กเมาส์เมนู แก้ไข เลือกลบระเบียนลบเรคอร์ดที่เลือกไว้



รูปที่ 5-10 แสดงการลบระเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การเพิ่มเรคอร์ดให้ คลิ๊กเมาส์ เมนู แทรก เลือก ระเบียนใหม่



รูปที่ 5-11 แสดงการเพิ่มเรคอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การออกแบบโปรแกรมระบบ Web Stat

อธิบายออกแบบ Webstat โดยใช้ UML

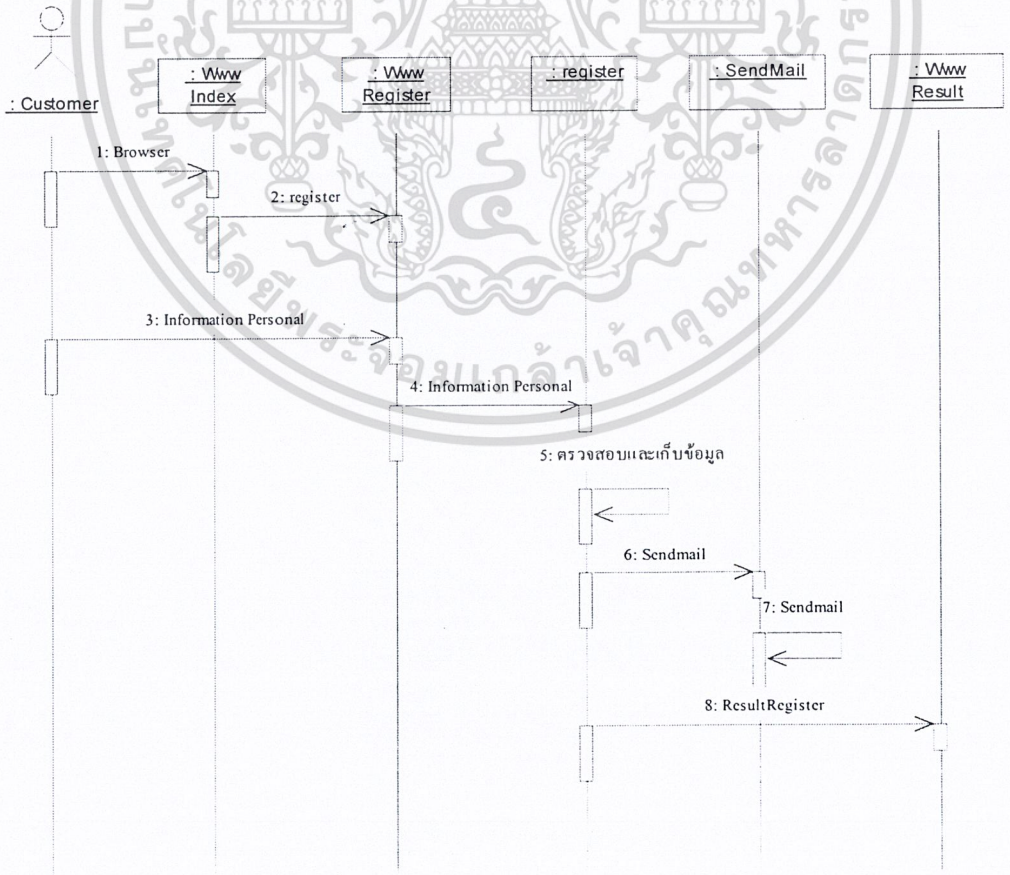
6.1 Use Case Diagram



รูปที่ 6-1 Use Case Diagram

6.2 Sequence Diagram

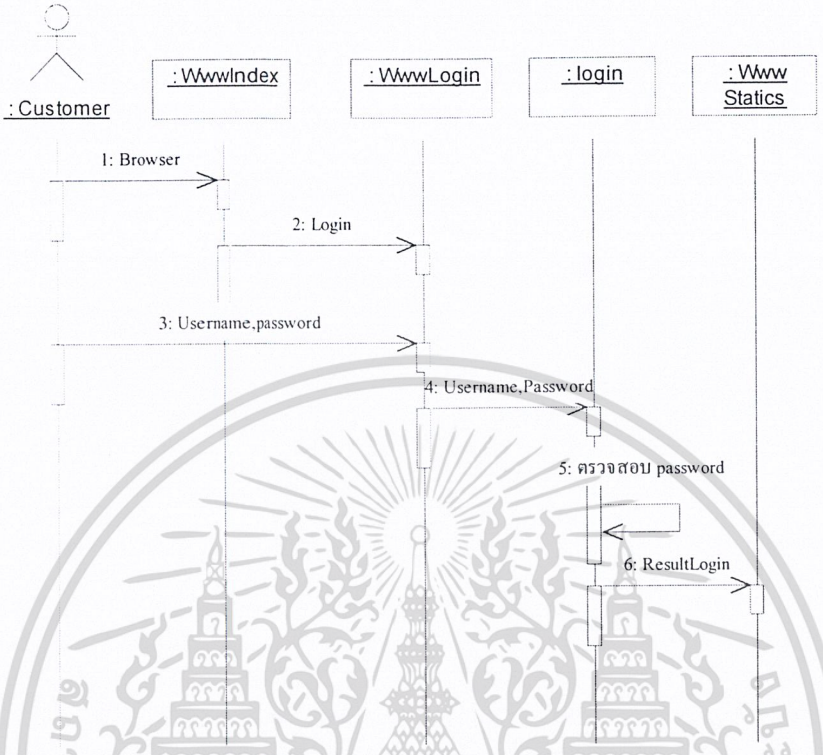
6.2.1 Register Sequnce Diagram



รูปที่ 6-2-1 Register Sequnce Diagram

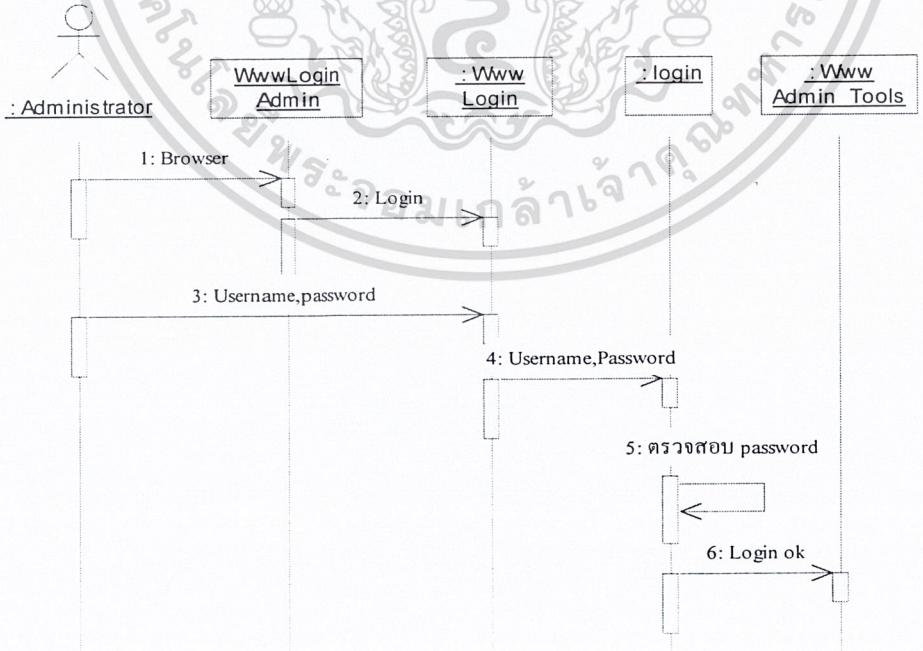
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 Statics Sequence Diagram



รูปที่ 6-2-2 Statics Sequence Diagram

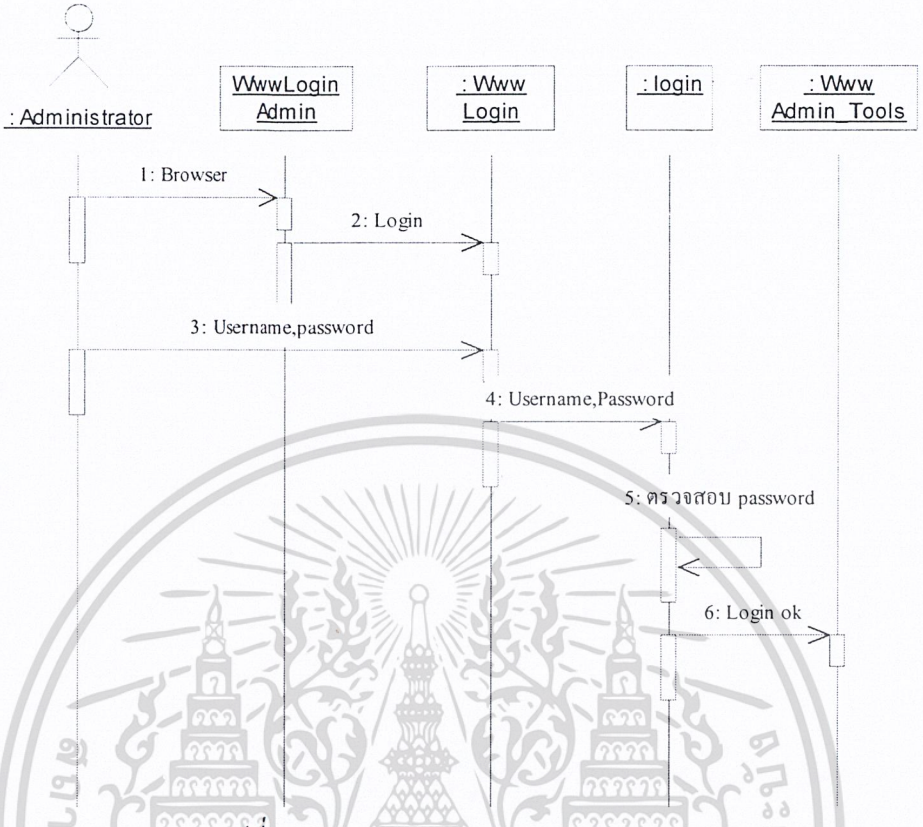
6.2.3 Login Admin Sequence Diagram



รูปที่ 6-2-3 Login Admin Sequence Diagram

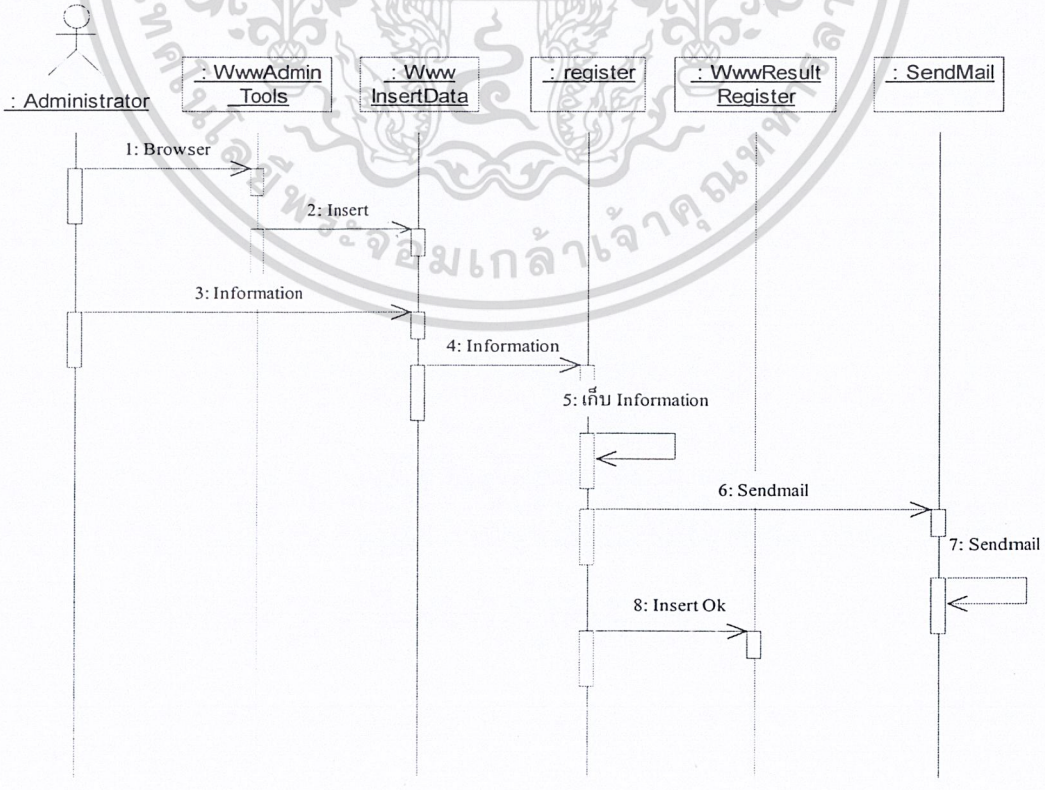
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 Search Sequence Diagram



รูปที่ 6-2-4 Search Sequence Diagram

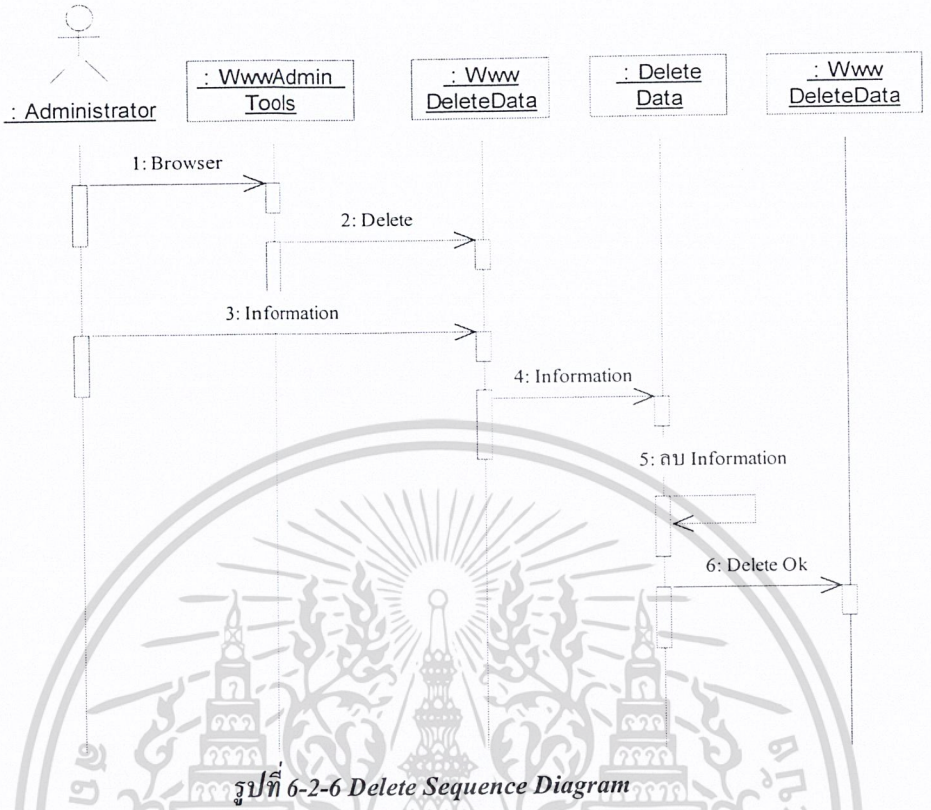
6.2.5 Insert Sequence Diagram



รูปที่ 6-2-5 Insert Sequence Diagram

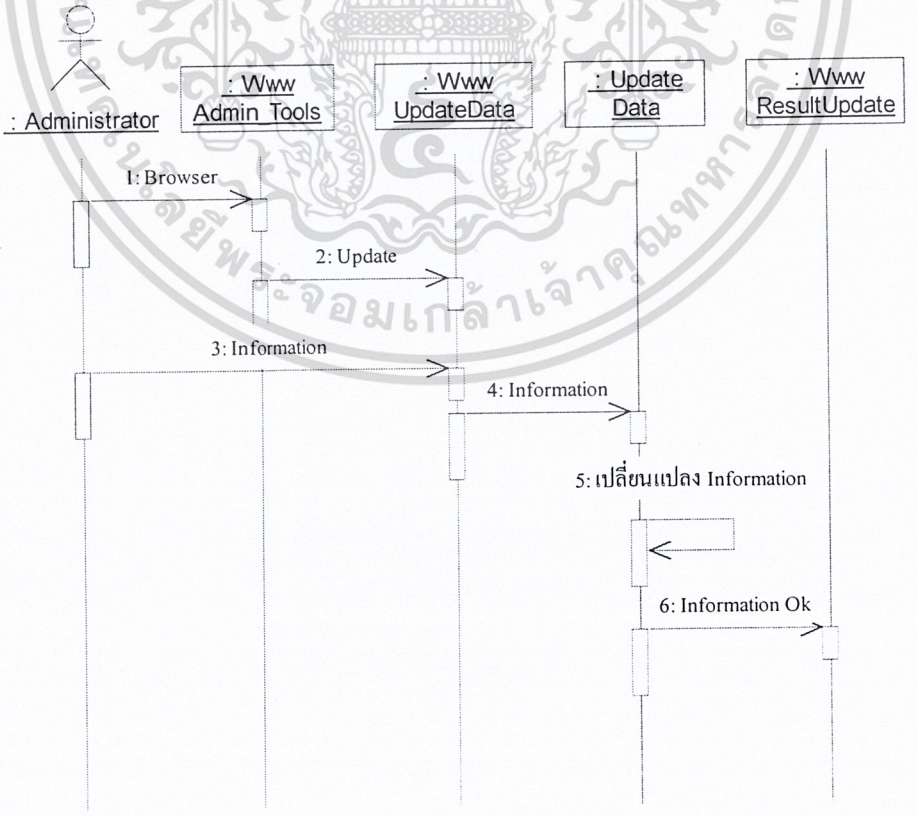
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.6 Delete Sequence Diagram



รูปที่ 6-2-6 Delete Sequence Diagram

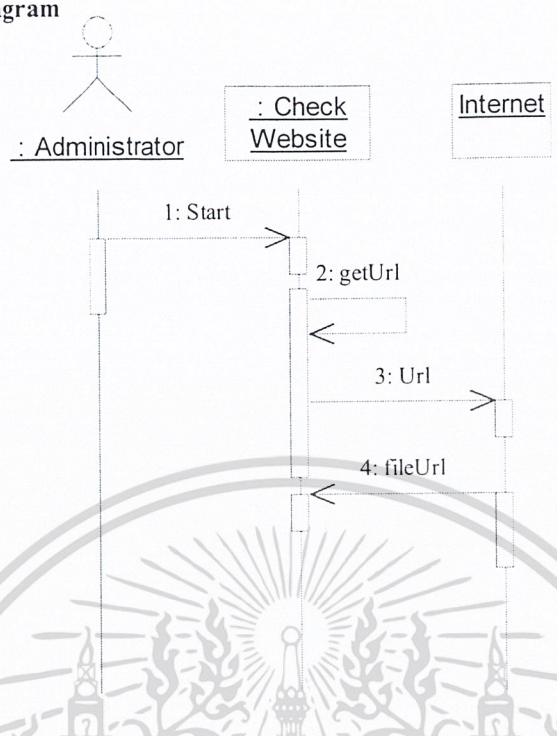
6.2.7 Update Sequence Diagram



รูปที่ 6-2-7 Update Sequence Diagram

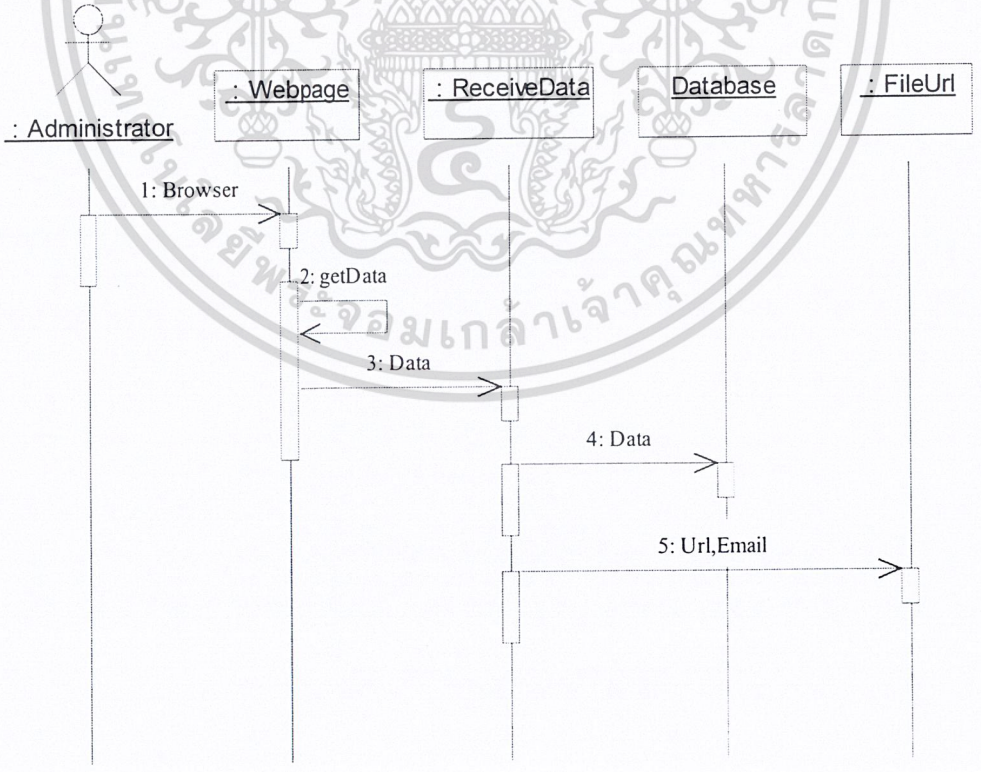
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.8 Check Website Diagram



รูปที่ 6-2-8 Check Website Diagram

6.2.9 เก็บข้อมูล Sequence Diagram

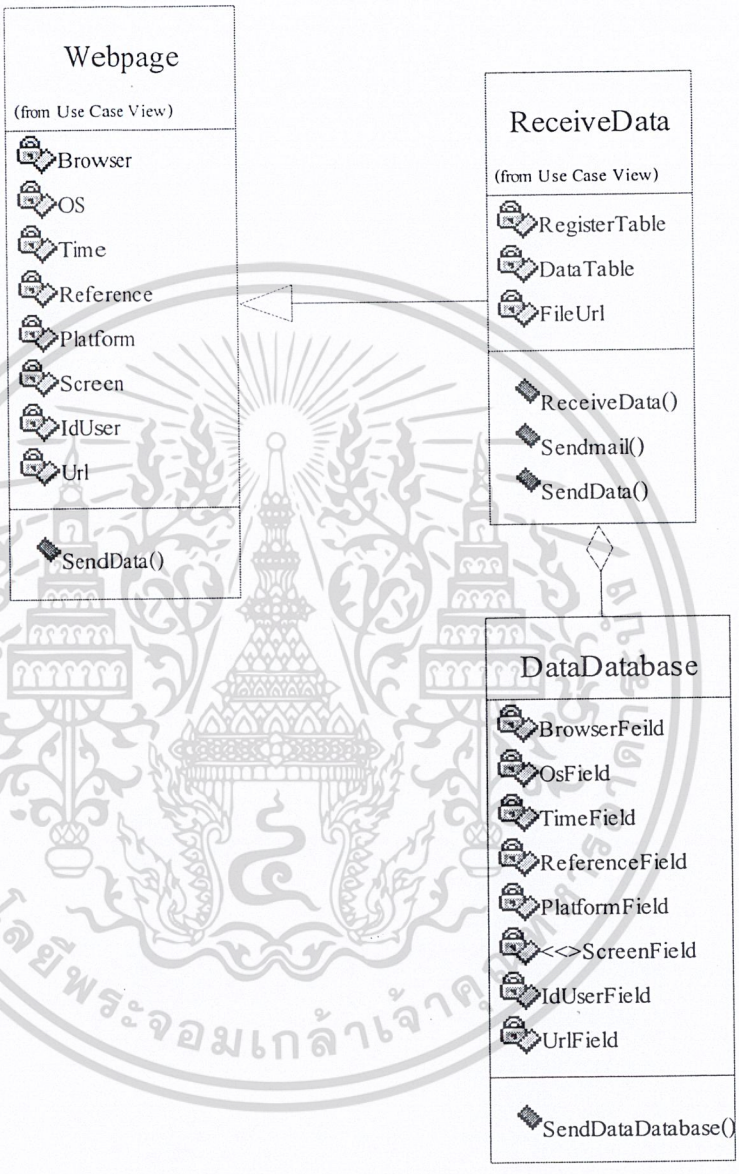


รูปที่ 6-2-9 เก็บข้อมูล Sequence Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 Class Diagram

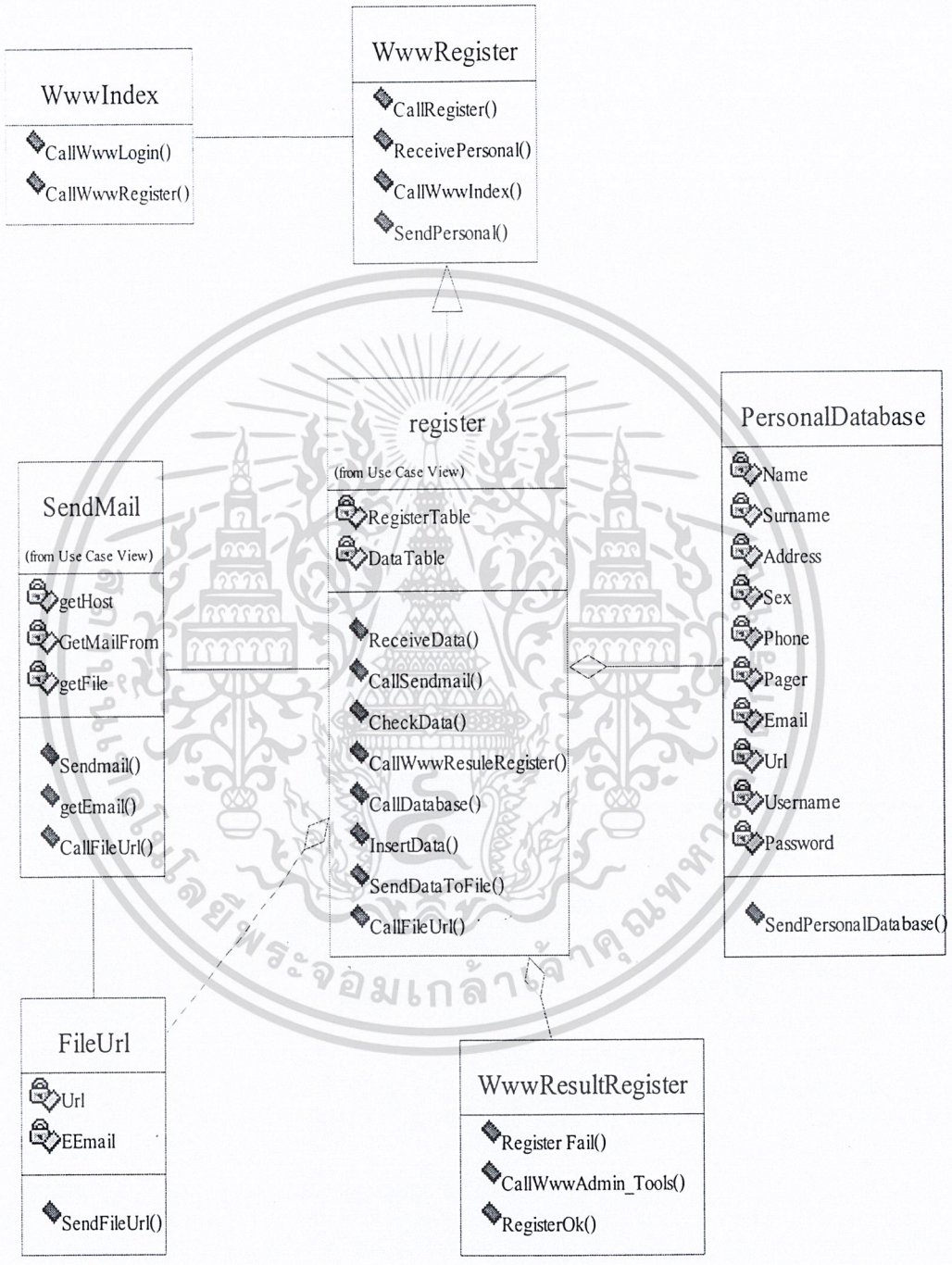
6.3.1 Class Diagram การเก็บข้อมูล



รูปที่ 6-3-1 Class Diagram การเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

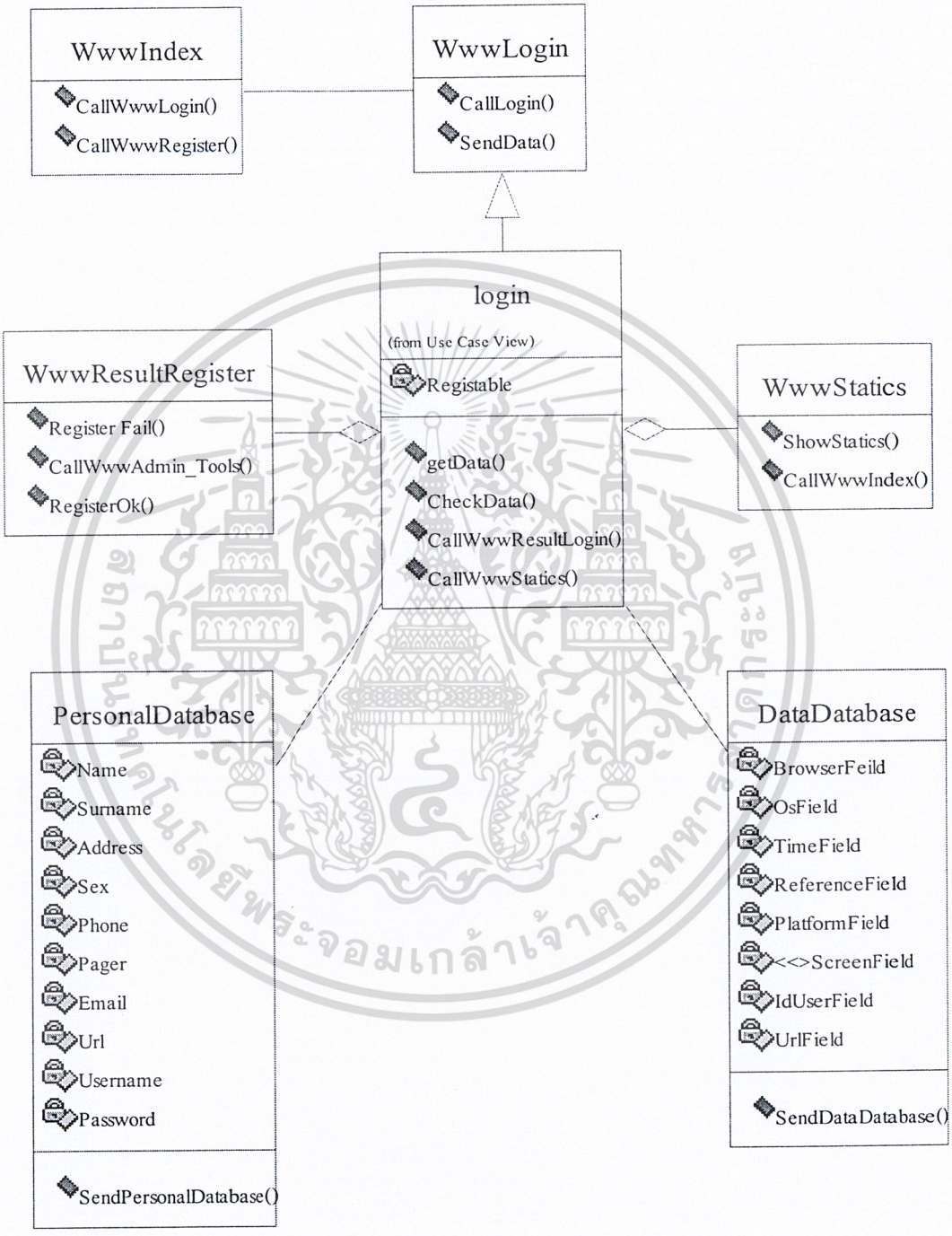
6.3.2 Class Diagram การลงทะเบียน



รูปที่ 6-3-2 Class Diagram การลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

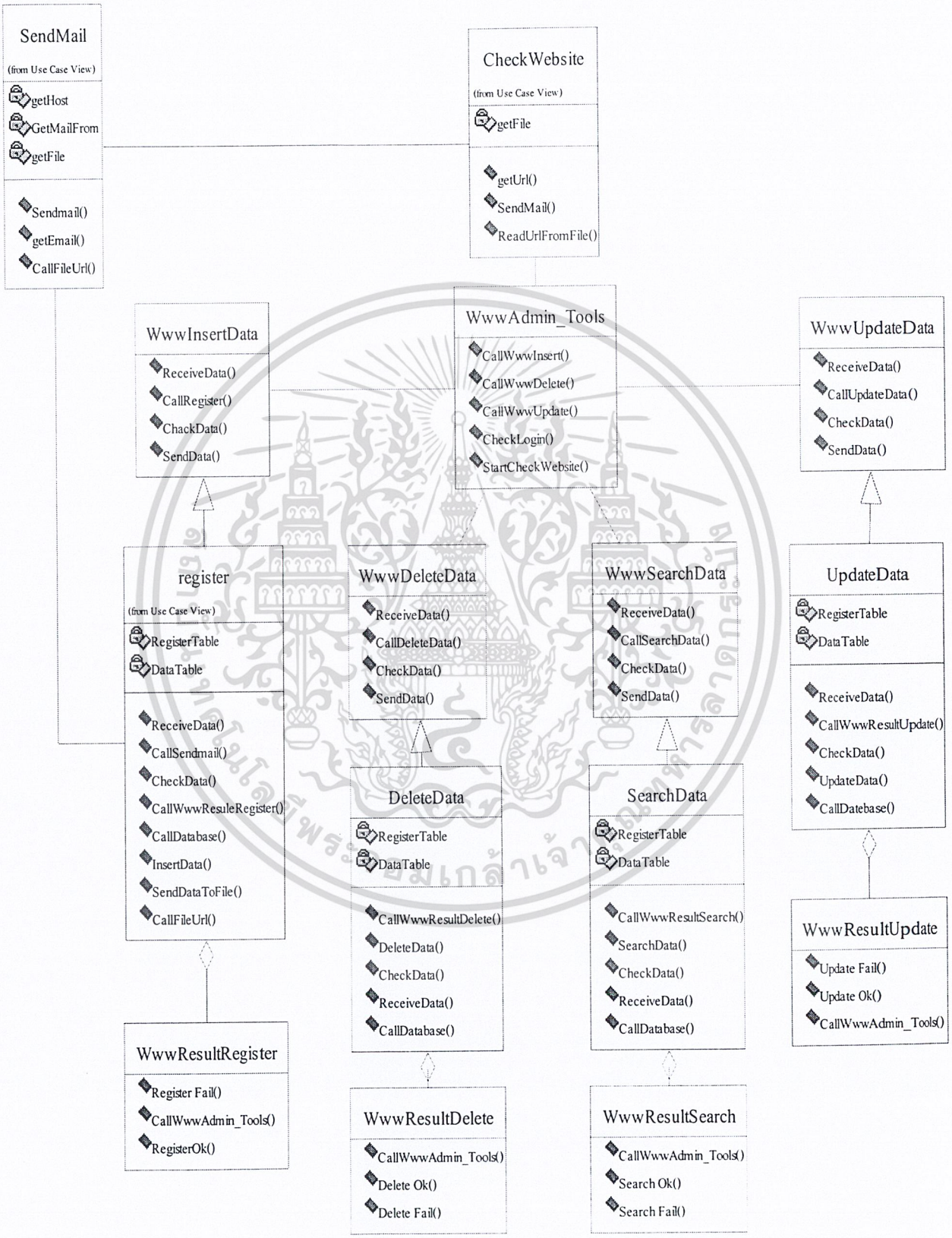
6.3.3 Class Diagram การดูสถิติ



รูปที่ 6-3-3 Class Diagram การดูสถิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.4 Class Diagram การดูแลระบบ

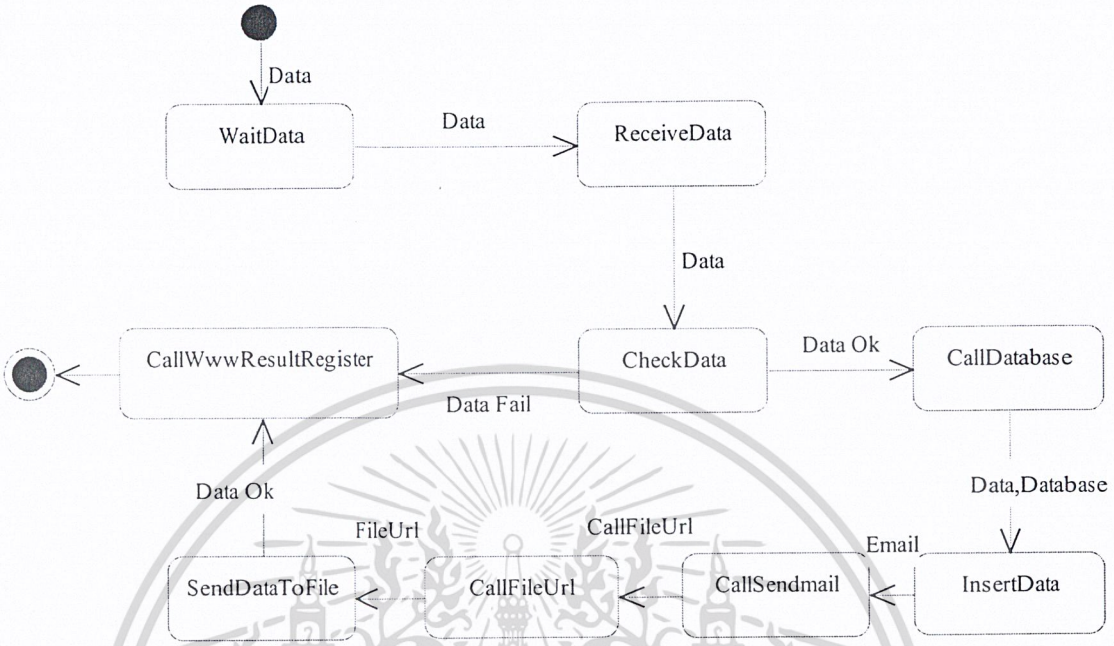


รูปที่ 6-3-4 Class Diagram การดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

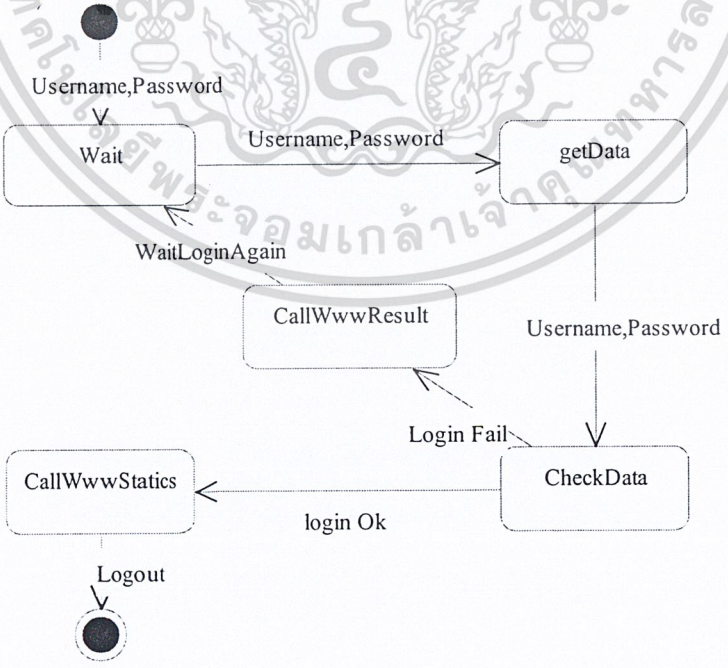
6.4 Stat Diagram

6.4.1 Register State Diagram



รูปที่ 6-4-1 Register State Diagram

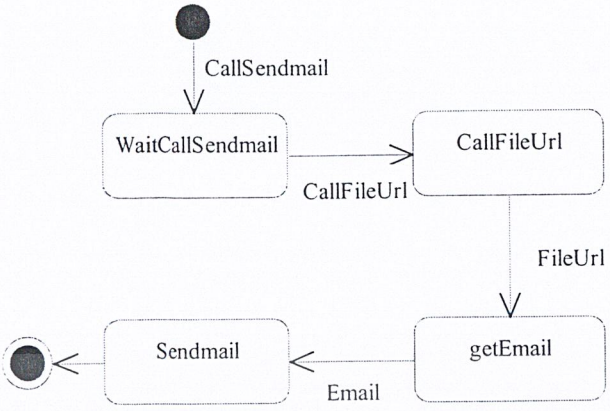
6.4.2 Login State Diagram



รูปที่ 6-4-2 Login State Diagram

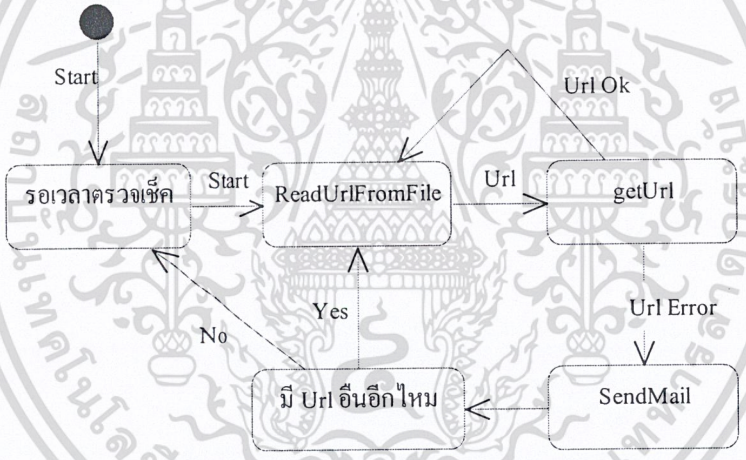
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.3 SendMail State Diagram



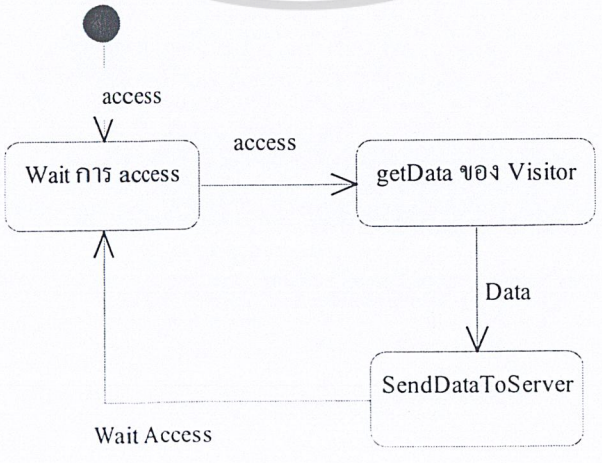
รูปที่ 6-4-3 SendMail State Diagram

6.4.4 Check Website State Diagram



รูปที่ 6-4-4 Check Website State Diagram

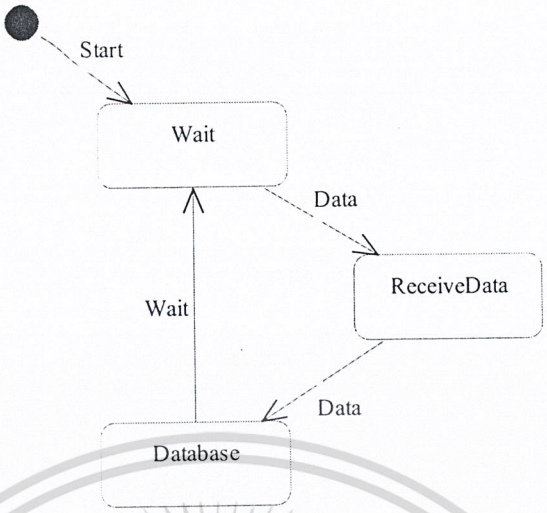
6.4.5 Webpage State Diagram



รูปที่ 6-4-5 Webpage State Diagram

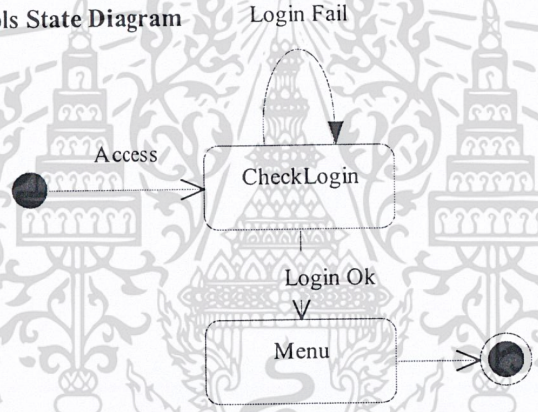
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.6 ReceiveDate State Diagram



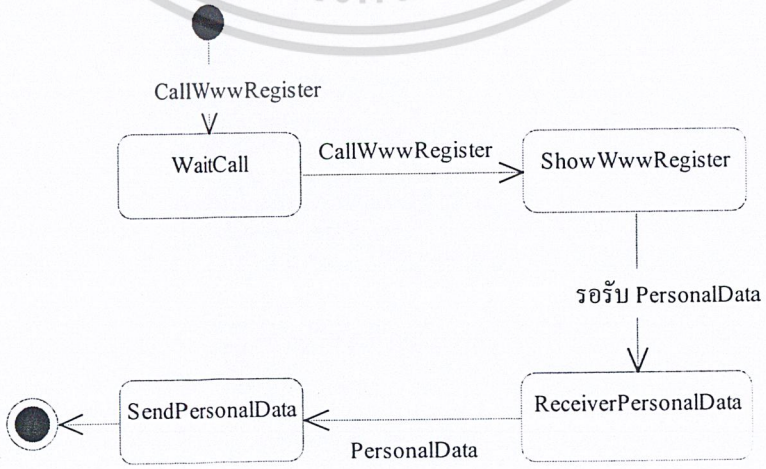
รูปที่ 6-4-6 ReceiveDate State Diagram

6.4.7 Www Admin_Tools State Diagram



รูปที่ 6-4-7 Www Admin_Tools State Diagram

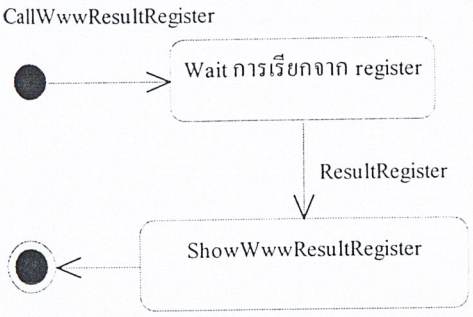
6.4.8 Www Register State Diagram



รูปที่ 6-4-8 Www Register State Diagram

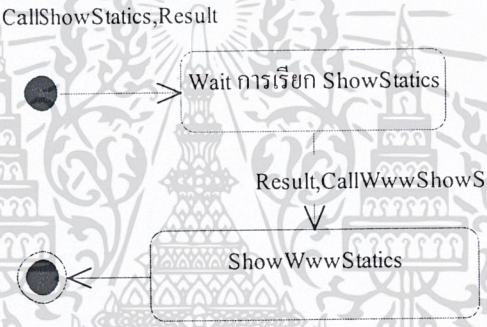
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.9 WwwResultRegister State Diagram



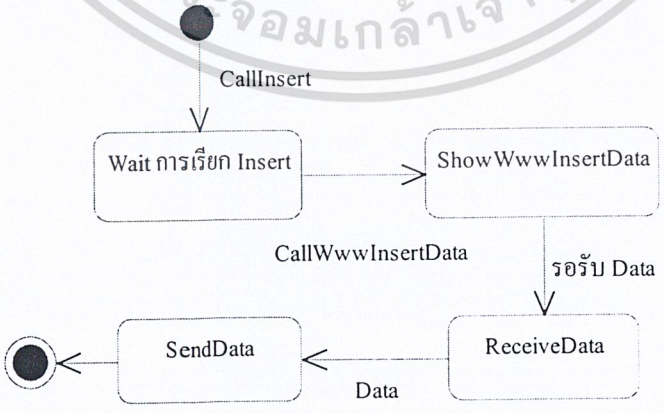
รูปที่ 6-4-9 WwwResultRegister State Diagram

6.4.10 WwwStatics State Diagram



รูปที่ 6-4-10 WwwStatics State Diagram

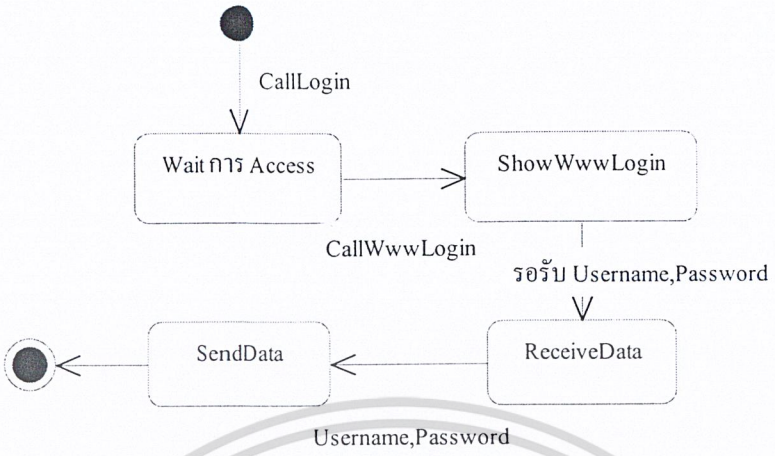
6.4.11 WwwInsertData State Diagram



รูปที่ 6-4-11 WwwInsertData State Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.12 Wwwlogin State Diagram



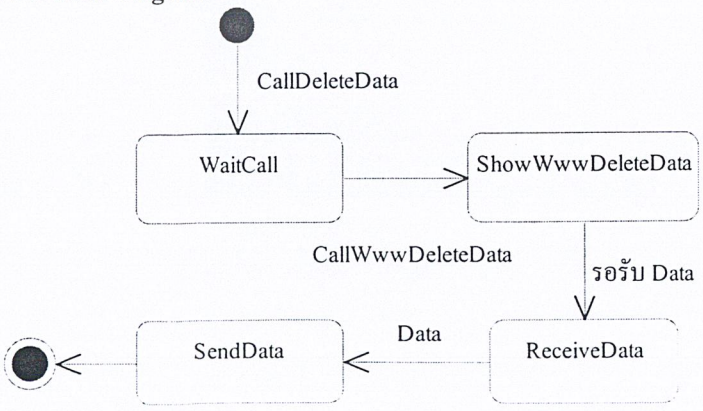
รูปที่ 6-4-12 Wwwlogin State Diagram

6.4.13 WwwIndex State Diagram



รูปที่ 6-4.13 WwwIndex State Diagram

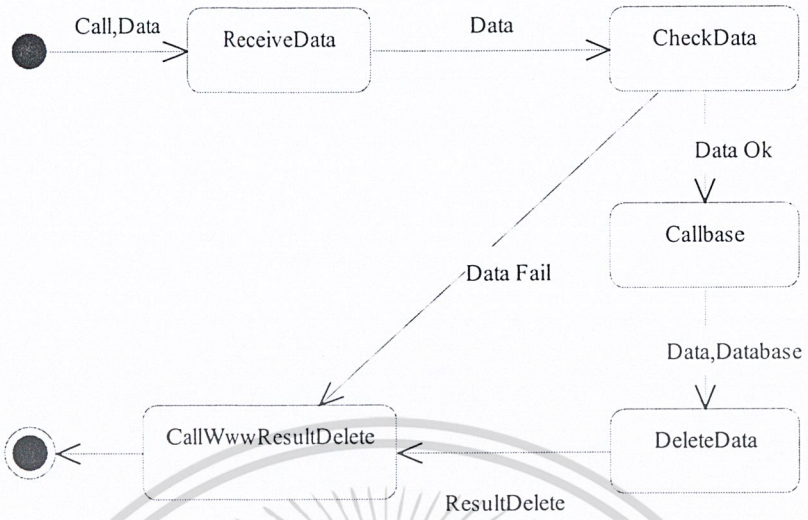
6.4.14 WwwDeleteData State Diagram



รูปที่ 6-4-14 WwwDeleteData State Diagram

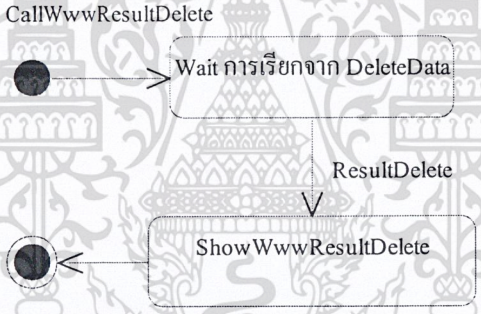
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.15 DeleteData State Diagram



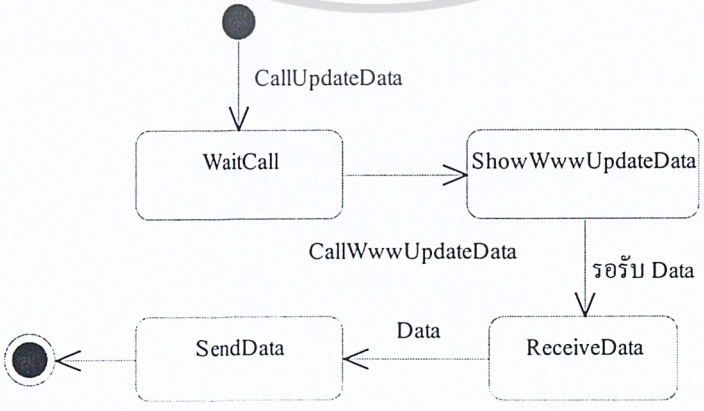
รูปที่ 6-4-15 DeleteData State Diagram

6.4.16 WwwResultDelete State Diagram



รูปที่ 6-4-16 WwwResultDelete State Diagram

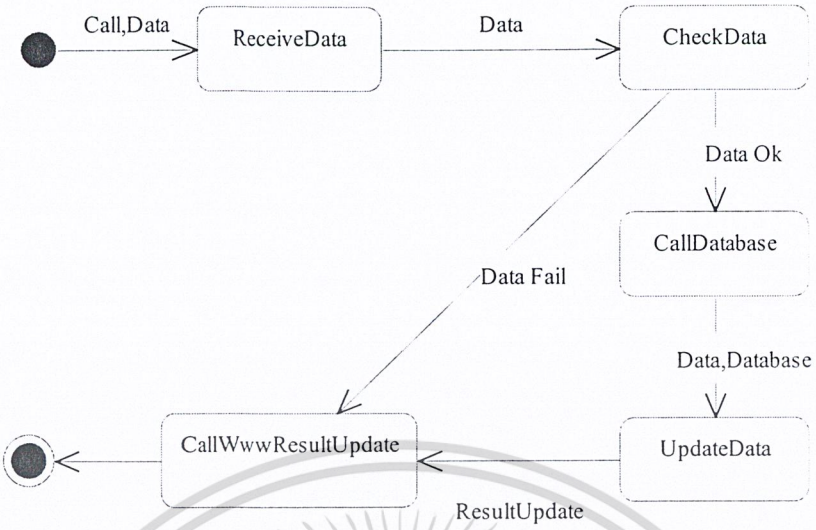
6.4.17 WwwUpdateData State Diagram



รูปที่ 6-4-17 WwwUpdateData State Diagram

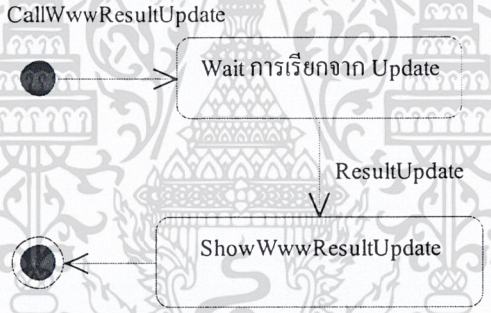
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.17 UpdateData State Diagram



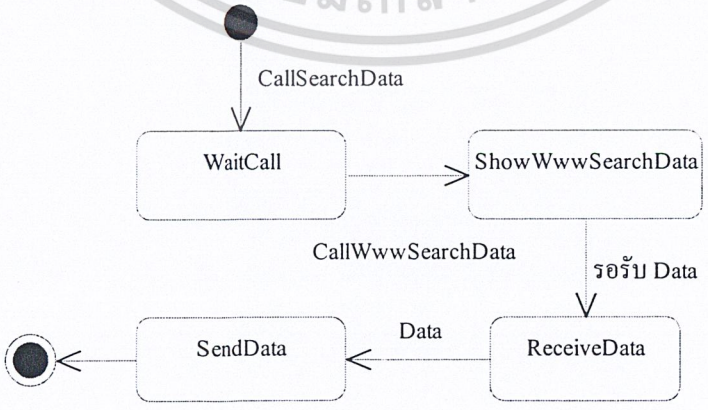
รูปที่ 6-4-17 UpdateData State Diagram

6.4.18 WwwResultUpdate State Diagram



รูปที่ 6-4-18 WwwResultUpdate State Diagram

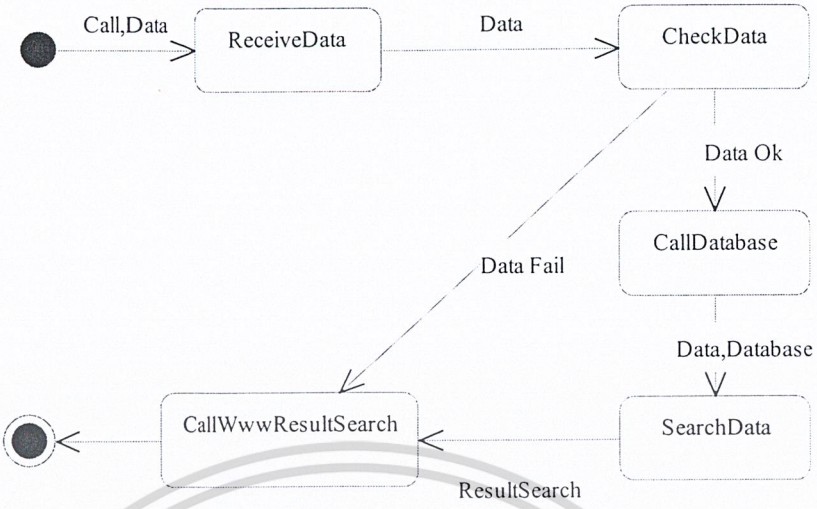
6.4.19 WwwSearchData State Diagram



รูปที่ 6-4-19 WwwSearchData State Diagram

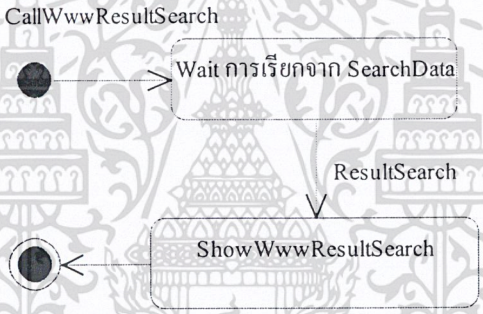
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.20 SearchData State Diagram



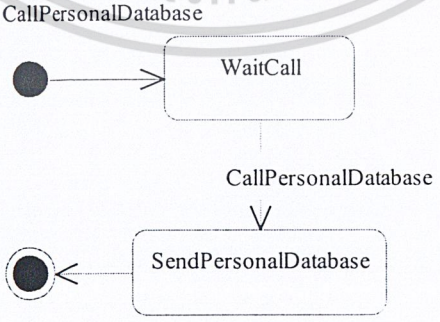
รูปที่ 6-4-20 SearchData State Diagram

6.4.21 WwwResultSearch State Diagram



รูปที่ 6-4-21 WwwResultSearch State Diagram

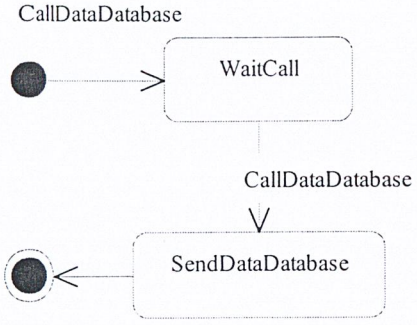
6.4.22 PersonalDatabase State Diagram



รูปที่ 6-4-22 PersonalDatabase State Diagram

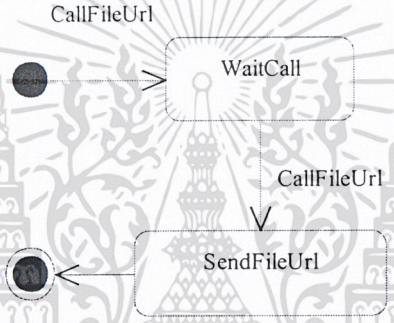
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.23 DataDatabase State Diagram



รูปที่ 6-4-23 DataDatabase State Diagram

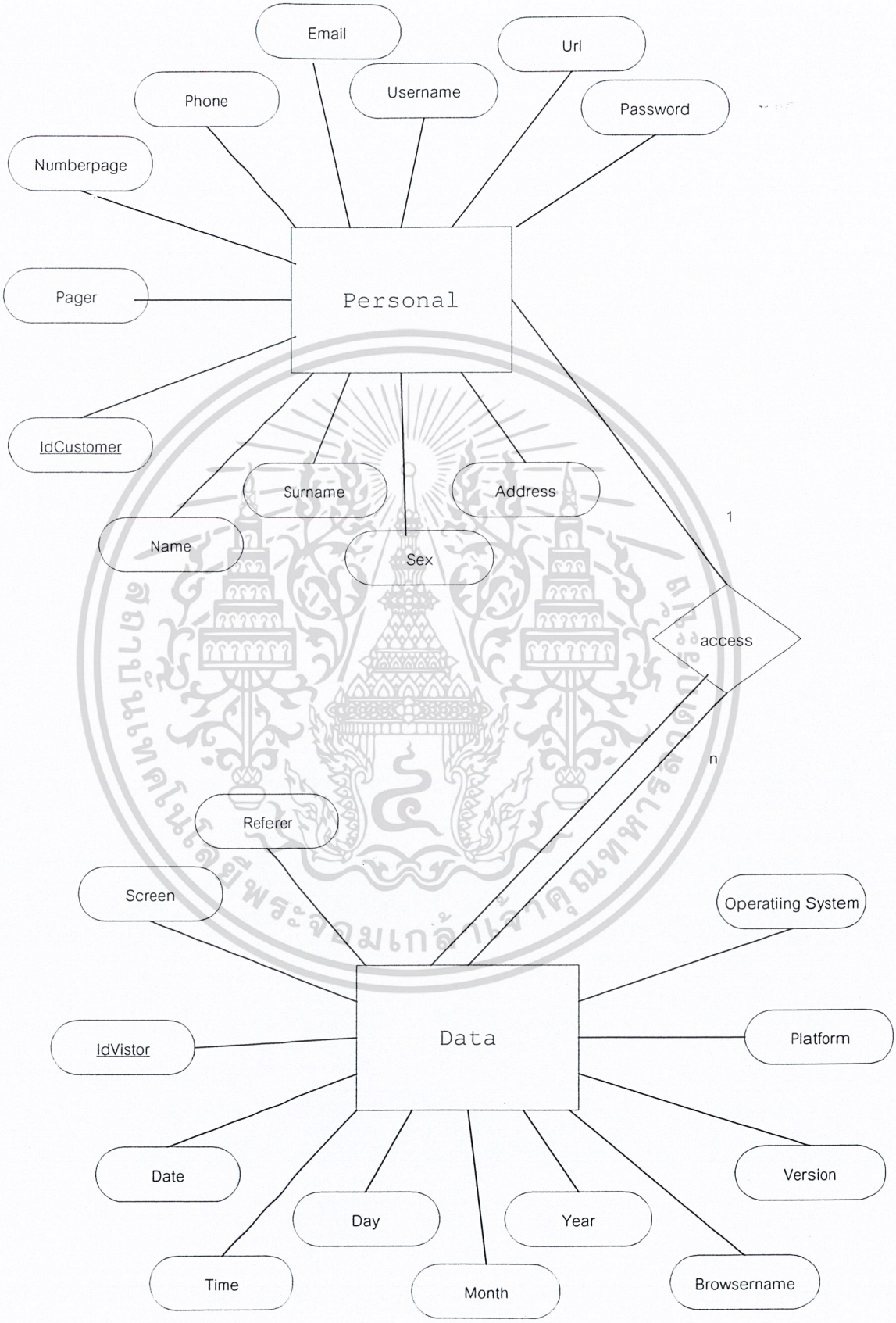
6.4.24 FileUrl State Diagram



รูปที่ 6-4-24 FileUrl State Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 ออกแบบฐานข้อมูลโดยใช้ ER Diagram

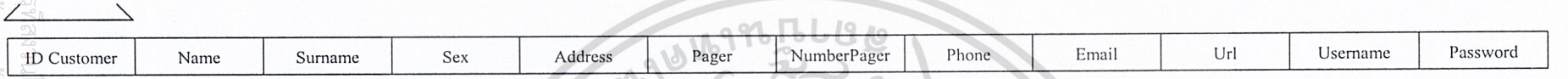


รูปที่ 6-5 ออกแบบฐานข้อมูลโดยใช้ ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

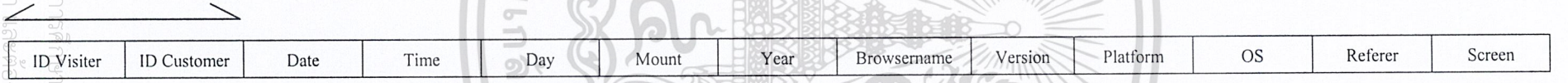
จาก ER Diagram จะได้ 2 Table ดังนี้

1.Personal Table



ID Customer	Name	Surname	Sex	Address	Pager	NumberPager	Phone	Email	Url	Username	Password

2.Data Table



ID Visiter	ID Customer	Date	Time	Day	Mount	Year	Browsername	Version	Platform	OS	Referer	Screen

บทที่ 7
สรุปและวิจารณ์

7.1 ภาพรวม

งานวิจัยนี้ได้พัฒนาระบบให้บริการจัดเก็บสถิติและแสดงผลสำหรับเว็บเพจแบบออนไลน์ เพื่อใช้ในการจัดเก็บข้อมูลของผู้เข้าเยี่ยมชมเว็บเพจของผู้รับบริการ โดยข้อมูลที่สามารถจัดเก็บมาทำสถิติได้นั้นจะช่วยให้ผู้บริหารเว็บไซต์สามารถนำมาใช้ในการช่วยตัดสินใจในการวางนโยบายสำหรับพัฒนาเว็บไซต์ในระดับเบื้องต้นได้

ในการวัดประสิทธิภาพของระบบจริง ๆ นั้นวัดระดับยากพอสมควรเพราะหากจะวัดกันจริงๆต้องทำการติดตั้งใช้งานจริงโดยวัดในขณะที่มีสมาชิกจำนวนมากๆเข้ามาใช้บริการและนั่นจะหมายถึงในแง่ของประสิทธิภาพของเซิร์ฟเวอร์ในระบบว่าจะสามารถรองรับการทำงานได้มากน้อยแค่ไหนด้วย ดังนั้นในที่นี้เราจะพูดประสิทธิภาพของสถิติที่ได้ซึ่งเป็นเนื้อหาจริงๆและในแง่ของซอฟท์แวร์ที่ใช้ในการพัฒนาวิจัยนี้มากกว่า

ซึ่งเครื่องมือที่ใช้ในงานวิจัยนี้ประกอบด้วย

- จาวาสคริปต์

  ใช้ในการดึงข้อมูลของผู้เข้าเยี่ยมชมเว็บไซต์และส่งมาให้เซิร์ฟเวอร์

ข้อดี	ข้อเสีย
สามารถดึงข้อมูลได้มากพอสมควร	ยังไม่สามารถดึงข้อมูลโฮสไอพี (real ip)ได้ ทำให้ไม่สามารถป้องกันการปั่นสถิติได้ (จริงๆทำได้เฉพาะกับเบราเซอร์ที่เป็นของ Netscape Compatible เท่านั้น)
ทำงานที่ฝั่งไคล์เอ็นท์ทำให้ไม่ส่งผลให้เซิร์ฟเวอร์ของเราทำงานหนัก	ต้องแปะสคริปต์ไว้กับเว็บเพจของผู้เข้ารับบริการ ซึ่งหากสคริปต์นี้ยาวเกินไปแล้วจะทำให้ไม่ได้รับการยอมรับจากผู้เข้ารับบริการ
เบราเซอร์ส่วนใหญ่สนับสนุนการใช้งานภาษาจาวาสคริปต์	เบราเซอร์ยอดนิยมของค่ายคู่แข่ง MS Internet Exploror นั้นไม่สนับสนุนจาวาสคริปต์ 100 เปอร์เซ็นต์ และโดยเฉพาะการไม่สนับสนุนฟังก์ชันที่ใช้ในการดึงค่าโฮสไอพี ทั้งนี้เพราะ Internet Exploror นั้นมีภาษา VBScript เป็นของตัวเองอยู่แล้ว

ตาราง 7-1 เปรียบเทียบข้อดี-ข้อเสียของภาษาจาวาสคริปต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จาวาเซิร์ฟเล็ท

  ใช้ติดต่อกับฟอร์ม HTML ของเว็บไซต์ผู้ให้บริการ และ รับข้อมูลจากจาวาสคริปต์มาประมวลผลจัดละเก็บลงดาตาเบส จัดได้ว่าเป็นหัวใจของงานวิจัยนี้เลยทีเดียว

ข้อดี	ข้อเสีย
Object Oriented สามารถลดความซับซ้อนของโครงสร้างโปรแกรม	ไม่เหมาะกับงานเล็กๆ
อำนวยความสะดวกในการนำกลับมาใช้ใหม่	
Platform Independent	
มีความเร็วที่สูงกว่า CGI	

ตารางที่ 7-2 เปรียบเทียบข้อดี-ข้อเสียของจาวาเซิร์ฟเล็ท

- จาวาเว็บเซิร์ฟเวอร์เวอร์ชัน 2

  ใช้งานได้ดีมากสนับสนุนจาวาเต็มที่ ติดตั้งได้ง่ายและปรับแต่งค่าสำหรับจาวาเซิร์ฟเล็ทได้ง่ายเช่นกัน เห็นจะมีข้อเสียอยู่ข้อเดียว คือ เป็นเวอร์ชันทดลองใช้ 30 วัน

- ไมโครซอฟท์แอคเซส

  ข้อดี คือ ใช้งานง่าย ข้อเสียคือ อาจจะรองรับและตอบสนองต่อรีเควสมาก ๆ ในงานจริงๆไม่ไหว แต่ต้องเสียเงินซื้อลิขสิทธิ์

7.2 แนวทางในการพัฒนาต่อ

หาวิธีหรือแนวทางในการดึงค่าโฮสไอพีให้ได้ในทุก ๆ เบราเซอร์ เพื่อเพิ่มความน่าเชื่อถือ

เพิ่มเติมบริการในเว็บไซต์ของผู้ให้บริการเอง เช่น จัดทำ TOP 10 จากสถิติของสมาชิกในระบบทั้งหมด, บริการตรวจสอบ bad link, pager alert เมื่อเซิร์ฟเวอร์ของผู้รับบริการขาดการติดต่อ

หันไปใช้ DBMS ยี่ห้ออื่นที่น่าจะดีกว่านี้ คือน่าจะรองรับต่อการร้องขอมากๆได้ และไม่ต้องเสียเงินซื้อเช่น MySQL เป็นต้น

ใช้ WebServer ยี่ห้ออื่นที่ไม่ต้องเสียเงินและเซ็ตอัพการใช้งานร่วมกันกับจาวาเซิร์ฟเล็ทได้เป็นอย่างดี อย่างเช่น Apache-Jakata-Tomcat เป็นต้น

พัฒนาในแง่ทางด้านความปลอดภัยของเว็บเพจผู้ให้บริการเอง โดยเฉพาะ Tools Admin Web Online ที่สามารถใช้ในการปรับปรุงแก้ไขดาตาเบสได้โดยตรงผ่านทางเบราเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ก. Dowload Java Web Server and Documentation

สามารถดาวน์โหลดเวอร์ชันทดลองใช้ 30 วัน ได้จากเว็บไซต์ของ Sun Microsystems ได้ฟรีที่ http://www.sun.com/software/jwebserver/techinfo/index.html;$sessionid$MGMNCOAAAFVU5AMTA1FU5YQ

และคู่ดาวน์โหลดคู่มือการติดตั้งและใช้งานได้ที่ http://www.sun.com/software/jwebserver/techinfo/doc/jws20_doc.zip;$sessionid$MGMNCOAAAFVU5AMTA1FU5YQ

ข. คู่มือการติดตั้ง Java Web Server บน WindowsNT

```
This is the README.install1NT file for
Java Web Server(tm) 2.0
Full-product version


last updated: 20July1999
============================

NOTE: On Windows platforms, you may need to use WordPad
(rather than NotePad) to read this file.

Windows Platforms

      Supported Platforms

      Prerequisites

      Defaults
        Installation Directory
        Java Runtime Environment
        Ports

      Installation

      Where to go from here


SUPPORTED PLATFORMS
-------------------

The Java Web Server(tm) is officially supported only under
the Solaris(tm) SPARC(tm) and Windows NT environments.
Windows 95 is supported for development but not
for deployment.

While Java Web Server 2.0 can run on any platform that
supports the JDK 1.1.7, platforms other than Solaris and
Windows are untested and therefore unsupported.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

To use the Java Web Server, you need a server environment
as well as a client environment.

For the server environment:

You require a Java Development Kit (JDK) 1.1.7 or
a JDK 1.1.7-compliant Java Runtime Environment (JRE).

Note:
Java Web Server 2.0 bundles a JDK 1.1.7-compliant JRE
which it will use by default. This is the recommended
way to run the Java Web Server.

For the client environment:

You require a JDK 1.1-compatible browser to run the
Java Web Server administrative tools.

Note:
Most recent browsers are JDK 1.1-compatible. Current
versions of Netscape Navigator (4.5) and Microsoft Internet
Explorer (4.0) are known to be compatible.

PREREQUISITES

Hardware and Operating System Prerequisites

Minimum
Intel 486-based PC
32 MB RAM
Windows 95 (with Winsock 2*) OR Windows NT 4.0

* Windows Sockets 2 (WinSock 2) may be downloaded from
Microsoft's
website at:

http://www.microsoft.com/windows95/downloads/contents/
wuadmintools/s_wunetworkingtools/w95sockets2/default.asp

Recommended
Intel Pentium 166 (or higher)-based PC
64 MB RAM
Windows NT 4.0 with Service Pack 4

The compressed file is approximately 8.5 MB.
To install it, you require roughly 13 MB of hard drive space.

Special Note for Windows 95 Users:

Windows 95 is supported as a development and
demonstration platform only. It is not supported as a
server deployment platform.

Microsoft does not recommend the use of Windows 95 as
a server platform.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEFAULTS

Installation Directory

By default, the product installs into

C:\JavaWebServer2.0\ (Windows example)

The directory path containing the installed Java Web Server 2.0 product is referred to as the . This term is used throughout the product documentation.

Note:
The default installation path means an entry is not made in the Start -> Programs listing.

Installing into a directory path that contains spaces is not recommended. For example, installing into C:\Program Files\ may cause problems for environment variables used by the Java Web Server.

Java Runtime Environment (JRE)

Java Web Server 2.0 supports JDK 1.1.7 or JDK 1.1.7-compliant JREs. By default, JRE 1.1.7 is automatically installed in the \jre directory. Where is the the directory containing the installed Java Web Server product.

For example

C:\JavaWebServer2.0\jre\ (Windows example)

Ports

The Java Web Server 2.0 installs its services on the following ports:

Web Service	8080
Secure Service	7070

These ports can be changed after installation using the Java Web Server Administration Tool, the product documentation tells you how. The Administration Tool is accessible on port 9090.

INSTALLATION on WINDOWS
----

1. Place the self-extracting file on your machine

Download the Java Web Server self-extracting file (.exe) from the Java Web Server web site (or copy the self-extracting file from the CD) onto your machine.

2. Extract installation files and launch setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Double-click on the self-extracting file to extract the
Java Web Server temporary installation files into the directory
and

launch the Setup program.

3. Follow the setup program's instructions

Follow the instructions in the Setup program to install
Java Web Server in a directory location on your hard disk.

Installation into a directory path containing spaces,
such as C:\Program Files\ is not recommended as it may
cause problems with environment variables used by
the Java Web Server.

By default, Setup installs Java Web Server in

C:\JavaWebServer2.0

The JavaWebServer2.0 directory is also called the .
This term is used throughout the Java Web Server documentation.

On Windows NT only:

4. Choose whether to install as an NT service

If you are installing on Windows NT, near the end of the
installation program you will be asked if you want to have
the Java Web Server start automatically on system reboot.
(That is, whether you want to install the Java Web Server
as an NT Service).

Note: You must have NT Administrator privileges to choose
this option.

If you click Yes:
An entry will be added to the Control Panels -> Services
and the JavaWebServer NT Service will be started up
automatically every time you restart your system.

If you click No:
No entry will be made in the Control Panel's Services panel.

If you change your mind later, the product documentation
provides instructions for how to setup the web server
to start automatically. For instructions, see the
file:

\doc\en\administration\server_start_Win.html

where is the directory containing the existing
installed Java Web Server.

5. At the end of the installation, click "View the README".

Where to go from here
=====================

If you have not done so already, be sure to the read
the Java Web Server 2.0 Release Notes and README file.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
The README contains Quick Start information such as:

      How to start the Java Web Server
      How to access the Administration Tool
      How to access the Documentation
      How to shut down the Java Web Server
      Where to find product information and news
      Additional Product Resources

The Release Notes contain important last-minute
  information that clarifies, corrects, or extends
  that provided in the product documentation.


Contacting the Java Web Server product team
===========================================
    E-mail can be sent to the following address:

      java-server-feedback@java.sun.com



END OF README.installNT

Copyright © 1999 Sun Microsystems, Inc.
```

ก. Administration Tool: Network Setup

Purpose

The Network Setup page enables you to define and change the basic network settings for the services running in your Java Web Server(™).

Location of panel in Administration Tool:

Log In --> *Service name* --> **Manage** --> Setup
|_ Network

Fields

Port
Defines the port that your HTTP or HTTPS service listens on to field requests.

A port number can be any number between 1 and 65535. (On UNIX(®), port numbers less than 1024 are considered protected ports and must be run as `root`. Therefore, to start Java Web Server on port 80, you would have to start it as `root`.)

The Java Web Server default ports are:

- **Web Service** (HTTP) - 8080

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Secure Web Service** (HTTPS) - 7070

Note: If you change the port number, you must restart the service by clicking Restart to have your change take effect.

Provide Service on
This field enables you to accept connections on more than one network address if your machine is set up for multihoming (for example, if you have installed two interface cards).

- **All Network Addresses**
  Sets your default port to listen to requests that come in on both (or all) network interface cards installed in your machine.

- **Network Address**
  Sets your default port to listen to requests coming in on only the card specified. Enter the host name assigned to the card, or the IP Address. If you have more than one interface card installed, requests coming in on that card are ignored if this field is selected.

Buttons

Restart
Temporarily stops and then restarts the service so your changes can be saved and activated. You only have to restart the service if you change the port.

Stop
If you choose, you can Stop a service rather than restart it. If you do this, the Restart button changes to read Start. To restart a stopped service, click Start.

Save
Writes your changes to Java Web Server and changes the settings.

Revert
Returns to the last, saved settings for all fields in the screen. Use this button to undo changes you have not yet saved.

Defaults
Displays the settings that were active when you entered the Java Web Server Network Setup page.

---

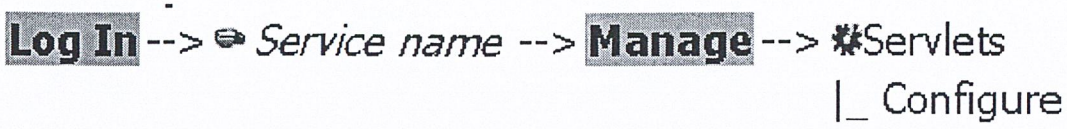
3. Administration Tool: Configuring Servlets

Purpose

The Configuring Servlets page enables you to enter, edit, or delete servlet configuration information for local or remote servlets. It also enables you to load and unload the servlet from the server. For more information on loading and invoking servlets, see Loading and Invoking Servlets.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Location of panel in Administration Tool:



Tabbed Panels

Configuration

Name
The name of the servlet (must be unique to the service).

Description
A text string describing the servlet. This field can be blank.

Servlet Class
The name of the associated class file for the servlet. If the servlet is part of a package, the class name should include the package name, if any. For example, if a servlet of class `OrderServlet` was in package `com.myCompany.sales`, then the class name provided should be `com.myCompany.sales.OrderServlet`

Load at Startup
Whether Java Web Server™ should load the servlet when the server starts.

Loaded Now
Whether the servlet is currently loaded.

Load Remotely
Whether Java Web Server loads the servlet from a remote location.

Class File URL
The URL that points to the class file for the remote servlet.

Properties

Name
The name of a property.

Value
The value of the property.

Buttons

Load/Unload
Loads or unloads the servlet.

Remove
Removes the servlet from the servlet list.

Save
Writes your changes to Java Web Server and changes the settings.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Revert
Returns you to the last, saved settings for all fields in the screen. Use this button to undo changes you have not yet saved.

Procedures

To Provide Configuration Information for a Servlet

To Delete Information for a Servlet

To Edit a Servlet Entry

To Add a Property

To Change a Property

To Delete a Property

To Provide Configuration Information for a Servlet:

1. In the Administration Tool, go to the *Service*->Manage->Servlets->Configure panel.
2. Click the Configuration tab. The Name and Servlet Class fields should have the same values as those you entered in the Adding Servlets screen.
3. Type a description of the servlet (optional) .
4. To have Java Web Server load the servlet when the server starts, click Yes in the Load at Startup field. (By default, servlets are loaded when you invoke them for the first time.)
5. If the servlet is a remote servlet, click Yes in the Load Remotely field, and, in the Class File URL field, type the full URL, (ending with a slash) of the servlet's class file on the remote server. For example, if `HelloWorld.class` exists on a remote Java Web Server named `nightsky.com` in the top level `public_html` directory, then the URL is `http://nightsky.com/`.
6. Click the Properties tab and add any initialization arguments.
7. Click Save.

To Delete Information for a Servlet:

1. In the Administration Tool, go to the *Service*->Manage->Servlets->Configure panel.
2. Click the Configuration tab.
3. Select the entry for the servlet that you want to delete.
4. Click Remove.

To Edit a Servlet Entry:

1. In the Administration Tool, go to the *Service*->Manage->Servlets->Configure panel.
2. Click the Configuration tab.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Select the entry.
4. Delete the entry.
5. Re-type the entry.
6. Click Save.

To Add a Property:

1. In the Administration Tool, go to the *Service*->Manage->Servlets->Configure panel.
2. Click the Properties tab.
3. Click the Add button. This will open a new row in the table.
4. Click in the Name column and enter the name of the property.
5. Click in the Value column add the property value.
6. Click Save. (If you are entering multiple properties, it is not necessary to click save after each entry.)

To Change a Property:

1. In the Administration Tool, go to the *Service*->Manage->Servlets->Configure panel.
2. Click the Properties tab.
3. Click the field (Name or Value) you want to modify.
4. Type the new text.
5. Click Save.

To Delete a Property:

1. In the Administration Tool, go to the *Service*->Manage->Servlets->Configure panel.
2. Click the Properties tab.
3. Select the row of the table that holds the property.
4. Click the Remove button to the right of the Properties table. (Do not click the Remove button at the bottom of the page, which removes the entire servlet from the service.)
5. Click Save.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ธาริน สิทธิธรรมชารี : "*โฟกัส Microsoft Access 97*",บริษัท ซัคเซส มีเดีย จำกัด กรุงเทพฯ

[2] กิตติ ภักดีวัฒนะกุล : "*Java ฉบับโปรแกรมเมอร์*", บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด 1999.

[3] ดร.วีระศักดิ์ ซึงถาวร : "*Fundamental of JAVA Programming Volume 1*", Sum Publishing Department, Sum System Company Limited. 1998.

[4] ดร.วีระศักดิ์ ซึงถาวร : "*Fundamental of JAVA Programming Volume 2*", Se-Education Public Company Limited. 2000.

[5] จิตเกษม พัฒนาศิริ : "**เสริมแต่งโฮมเพจให้มีชีวิตชีวาด้วย Java Script**",วิตตี้กรุ๊ป จำกัด,2537

[6] Karl Moss : "*Java Servlet*",McGraw-Hill 1999

[7] http://www.jarticles.com/tutorials/servlet/intro_servlet.html

[8] http://java.sun.com

[9] http://www.a1javascripts.com/tutorials

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้