

โปรแกรมจัดการนัดหมายบนเครือข่ายอินเทอร์เน็ต
Internet Appointment Board



โดย
นางสาวเกศรา โพธิปรีพัตร
นายวรงค์ ไตสุชาติ

เลขหมู่.....
เลขทะเบียน 42769
วัน, เดือน, ปี 10 ส.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจัดการนัดหมายบนเครือข่ายอินเทอร์เน็ต
Internet Appointment Board

โดย

นางสาวเกศรา โพธิ์ปริพัตร
นายวันดี โตสุชาติ

อาจารย์ที่ปรึกษา
ดร.ชุตติเมษฎ์ ศรีนิลทา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมจัดการนัดหมายบนเครือข่ายอินเทอร์เน็ต

Internet Appointment Board

ผู้จัดทำ

1. นางสาวเกศรา โทธิปริพัตร รหัสประจำตัว 40010075

2. นายพนิต โทสุชาติ รหัสประจำตัว 40010667



สุวิมล ภิรมย์

อาจารย์ที่ปรึกษา

(ดร.สุติเมษณ์ ศรีนิลทา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจัดการนัดหมายผ่านบนเครือข่ายอินเทอร์เน็ต

นางสาวเกศรา โพธิ์ปริพัตร 40010075

นายวันดี โดสุชาติ 40010667

ดร.ชุตินเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา

ปีการศึกษา 2543

บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อใช้ศึกษาวิธีการสร้าง และการออกแบบโฮมเพจที่สามารถใช้เป็นตัวกลางในการสื่อสาร และนัดหมายผ่านทางอินเทอร์เน็ต นอกจากนี้ยังได้นำเทคโนโลยี WAP หรือเรียกว่า Wireless Application Protocol มาใช้ควบคู่กับสร้างโฮมเพจ เพื่อรองรับแอปพลิเคชันที่จะเกิดขึ้นในอนาคตต่อไป

Internet Appointment Board เป็นโฮมเพจที่ทำงานคล้ายกับเป็นเลขาส่วนตัว ผู้ใช้สามารถทำการบันทึกตารางเวลาการของกิจกรรมต่างๆ ในรอบวัน รอบสัปดาห์ รอบเดือน หรือรอบปี เพื่อช่วยในการเตือนความทรงจำ เมื่อถึงเวลาของการนัดหมาย หรือกิจกรรมต่างๆที่ผู้ใช้ได้ทำการบันทึกไว้ จะมีการเรียกเตือนผ่านทางอีเมล (E-Mail) , โทรศัพท์ไร้สาย (Mobile Phone) หรือเพจเจอร์ (Pager) นอกจากนี้ยังช่วยในการตรวจสอบตารางเวลาของสมาชิก เพื่อหาเวลาว่างในการทำการนัดหมายเป็นหมู่คณะได้

เพิ่มเป็นการเพิ่มความสามารถเว็บไซต์จึงได้นำเทคโนโลยีทางการสื่อสารที่เรียกว่า WAP มาประยุกต์ใช้ทำให้สามารถใช้โทรศัพท์ไร้สายเข้ามาในเว็บนี้ เพื่อทำการอ่าน , บันทึก และแก้ไขข้อมูลต่างๆ ได้อย่างสะดวก และรวดเร็วมากยิ่งขึ้น

Internet Appointment Board

Ketsara Potiparipat

Wanut Tosuchart

Dr. Chutimet Srinilta

Abstract

The objective of the project is to study how to build and design a web site that has capability of being a center of communication between friends and colleagues. Our appointment board is equipped with various functionalities to make it easy, flexible and convenient to use. In addition to the calendar and to-do list, our appointment board gives its members an assistance in making group appointment and a reminder that helps remind members of important activities through e-mail, mobile phone (Short Message Service) and pager.

We had also brought in the WAP, Wireless Application Protocol, technology to make the site accessible from literally everywhere.

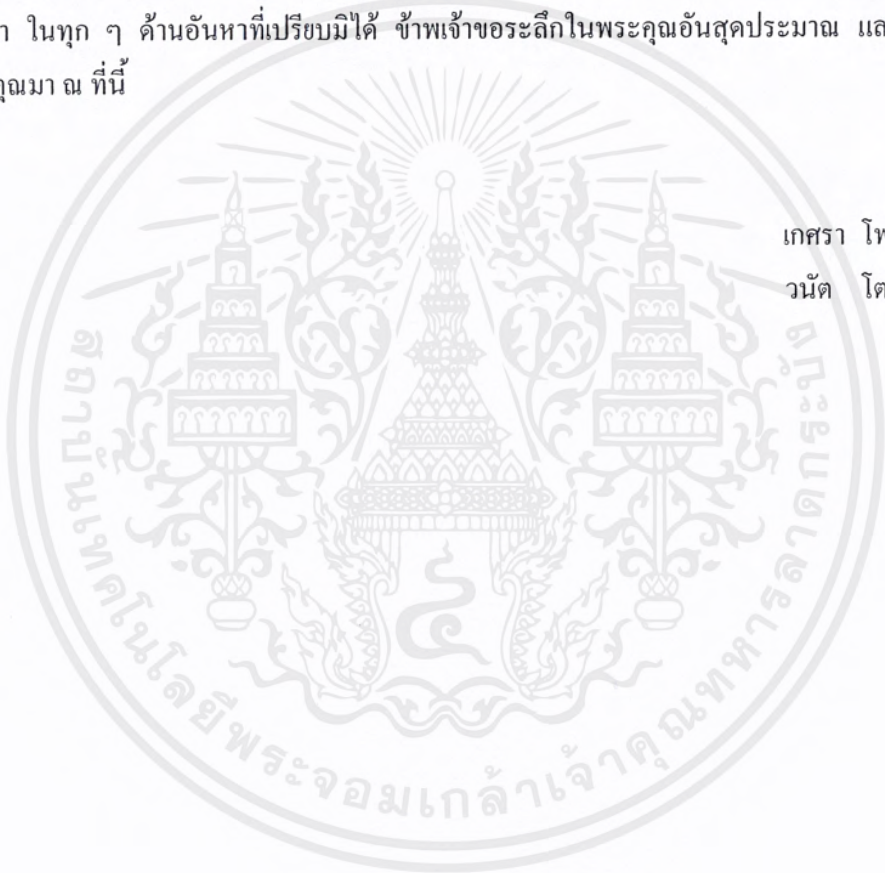
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ ชุตติเมษย์ ศรีนิลทา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ขอขอบคุณครูบาอาจารย์ทุกท่าน ที่ได้อบรมสั่งสอนทั้งด้านวิชาการ และจริยธรรม

และสุดท้ายนี้ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ที่ได้เลี้ยงดูมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

เกศรา โทธิปริพัตร
วันดี โตสุชาติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญ และความเป็นมาของ โครงการงาน	1
1.2 วัตถุประสงค์ของโครงการงาน	2
1.3 ขอบเขตของโครงการงาน	2
1.4 วิธีการดำเนินงาน	3
บทที่ 2 จาวาเซิร์ฟเว็ท (Java Servlet)	4
2.1 เว็บโปรแกรมมิ่ง (Web Programming)	4
2.1.1 ไคลเอนต์ไซด์สคริปต์ (Client Side Script)	4
2.1.2 เซิร์ฟเวอร์ไซด์สคริปต์ (Server Side Script)	5
2.2 จาวาเซิร์ฟเว็ท Java Servlet	6
2.2.1 Servlet Engine	6
2.2.2 วิธีการเขียน โปรแกรมเซิร์ฟเว็ท (Servlet)	7
2.2.3 การรันเซิร์ฟเว็ท (Servlet)	14
บทที่ 3 ฐานข้อมูล (Data Base)	15
3.1 คำจำกัดความของฐานข้อมูล	15
3.2 ออราเคิล (Oracle 8I)	16
3.2.1 สถาปัตยกรรมของระบบจัดการฐานข้อมูล(Architecture of DBMS)	16
3.2.2 ส่วนเชื่อมต่อสามส่วน(The Three Interface)	16
3.2.3 เทเบิลในระบบฐานข้อมูล	17
3.3 ภาษา SQL	19
3.3.1 การสืบค้น	19
3.3.2 คู่มือด้วย SQL	20
3.3.3 การติดต่อกับ Front End Tools	21
3.3.3.1 วิธีการติดต่อกับฐานข้อมูล	22
3.3.4 Tools ที่ได้มาพร้อมออราเคิล	22
บทที่ 4 WAP (Wireless Application Protocol)	25
4.1 บทนำ และประวัติความเป็นมา	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
4.2 ข้อแตกต่างระหว่างเว็บ(Web) และ เว็บบ (WAP)	27
4.3 องค์ประกอบของสถาปัตยกรรมของเว็บ(WAP)	29
4.3.1 Wireless Application Environment (WAE)	30
4.3.2 Wireless Session Protocol (WSP)	31
4.3.3 Wireless Transaction Protocol (WTP)	31
4.3.4 Wireless Transport Layer Security (WTLS)	32
4.3.5 Wireless Datagram Protocol (WDP)	33
4.3.6 Bearers	33
4.4 ประโยชน์จากการใช้ WAP	33
บทที่ 5 WML (Wireless Markup Language)	35
5.1 โครงสร้างของเอกสาร WML	35
5.2 WML Tag	36
5.2.1 แท็กแอตทริบิวต์	36
5.2.2 แท็ก p (<p>...</p>)	36
5.2.3 แอตทริบิวต์ align	36
5.2.4 แอตทริบิวต์ mode	36
5.2.5 แท็ก card (<card>...</card>)	37
5.2.6 แอตทริบิวต์ title	37
5.2.7 แอตทริบิวต์ id	37
5.3 การตกแต่งข้อความ	37
5.4 การนำเสนอข้อมูลในลักษณะตาราง	37
5.4.1 แอตทริบิวต์ที่ควรรู้ของแท็ก table	38
5.4.2 การนำเสนอข้อมูลในตาราง	39
5.5 การสร้างลิงค์	39
5.5.1 การสร้างลิงค์ด้วยแท็ก <do>	39
5.5.2 การสร้างลิงค์จากแท็ก <anchor> และ <a>	40
5.5.3 แท็กของเหตุการณ์ต่าง ๆ (Event) ที่ใช้สำหรับเปลี่ยน URI	41
5.6 แท็กคำสั่งของงาน (Task)	41
5.6.1 แท็ก <go>	41
5.6.2 แท็ก <prev>	42
5.7 การรับข้อมูลจากผู้ใช้	42
5.8 การส่งค่าตัวแปร	43
5.8.1 การส่งข้อมูลแบบ GET	43
5.8.2 การส่งข้อมูลแบบ POST	44
บทที่ 6 การเขียน โปรแกรมส่งข้อความเข้าเพจเจอร์ หรือมือถือ	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
6.1 การส่งโดยผ่านโปรแกรม Telnet	46
6.1 เขียนโปรแกรมส่งข้อความเข้าเพจเจอร์ และมือถือ	48
บทที่ 7 การคำนวณ การสร้าง และการออกแบบ	54
7.1 ขอบเขตการดำเนินงานของ โปรแกรมการนัดหมายผ่านเครือข่ายอินเทอร์เน็ต	54
7.2 ขั้นตอนการดำเนินงาน	54
7.3 ขั้นตอนการกำหนดอุปกรณ์ และ โปรแกรมที่ต้องใช้ใน โครงการงาน	55
7.4 ขั้นตอนการออกแบบ	56
7.4.1 Use-Case Diagram	56
7.4.2 Class Diagram	59
7.4.3 Object Diagram	60
7.4.3 Sequence Diagram	61
7.4.4 อีอาไดอะแกรม (ER Diagram)	63
7.4.5 การออกแบบส่วนที่ติดต่อกับผู้ใช้ และฟังก์ชันการทำงาน	65
บทที่ 8 บทวิจารณ์และสรุป	75
8.1 สรุปผลการดำเนินงาน และสรุป	75
8.2 แนวทางในการพัฒนาต่อ	75
ภาคผนวก	76
ก. การติดตั้ง Jakarta-Tomcat เว็บเซิร์ฟเวอร์	76
ข. ขั้นตอนการติดตั้ง โปรแกรมออร์กาเคิล 8I	80
บรรณานุกรม	83

สารบัญรูปภาพ

	หน้าที่
รูปที่ 2.1 แสดงการทำงานของเว็บเพจแบบสถิติก (Static Web Page)	4
รูปที่ 2.2 แสดงการทำงานของเว็บเพจแบบไดนามิก(Dynamic Web Page)	4
รูปที่ 2.3 แสดงถึง Servlet Engine and Servlets ภายในระบบ Servlet Engine / Servlet Container	7
รูปที่ 3.1 แสดงฐานข้อมูลและโปรแกรมการจัดการ	16
รูปที่ 3.2 แสดงองค์ประกอบทั้ง 4 ส่วน	16
รูปที่ 3.3 แสดงหน้าจอล็อกอินของ SQL Plus	20
รูปที่ 3.4 ภาพแสดงหน้าต่างไอคอน SQL Plus	23
รูปที่ 3.5 ภาพแสดงโปรแกรม SQL*Plus	23
รูปที่ 3.6 หน้าตาของ SQL*Plus หลังจาก Log on เสร็จเรียบร้อย	24
รูปที่ 4.1 ภาพแสดงสถาปัตยกรรมของอินเทอร์เน็ต	27
รูปที่ 4.2 แสดงสถาปัตยกรรมการสื่อสารเว็บ (WAP)	28
รูปที่ 4.3 แสดงรูปโปรโตคอลการสื่อสารของเว็บ (WAP)	28
รูปที่ 4.4 แสดงรูปโปรโตคอลการสื่อสารของเว็บ(WAP) แบบที่ 2	29
รูปที่ 4.4 แสดงเลขเอร์ที่ใช้กับ WAP	30
รูปที่ 6.1 แสดงการเชื่อมต่อของไคลเอนต์กับเซิร์ฟเวอร์	45
รูปที่ 6.2 แสดงวิธีการทำงานในการส่งข้อความเข้าเพจเจอร์ หรือมือถือ	47
รูปที่ 6.3 แสดงการส่งข้อความเข้าเพจเจอร์ 1144 โดยใช้เขียน โปรแกรม PERL	51
รูปที่ 6.4 แสดงการเขียนโปรแกรมจาวาเพื่อติดต่อส่งข้อความเพจเจอร์	52
รูปที่ 6.5 แสดงการคอมไพล์ และรัน โปรแกรมจาวาเพื่อส่งข้อความเข้าเพจเจอร์	53
รูปที่ 6.6 แสดงข้อมูลที่เซิร์ฟเวอร์ผู้ให้บริการส่งกลับ	53
รูปที่ 7.1 แสดงภาพ User-Case Diagram	56
รูปที่ 7.2 แสดงกราฟไดอะแกรม (Class Diagram)	59
รูปที่ 7.3 แสดงออบเจกต์ไดอะแกรม	60
รูปที่ 7.4 แสดง Sequence Diagram	61
รูปที่ 7.5 แสดง ER Diagram	63
รูปที่ 7.6 แสดงหน้าจอการสมัครเป็นสมาชิก	65
รูปที่ 7.7 แสดงหน้าจอ Login ของผู้ใช้	65
รูปที่ 7.8 แสดงหน้าจอหลักของโครงการนี้	66
รูปที่ 7.9 แสดงหน้าจอของ To Do ,Quick Add และ Find	66
รูปที่ 7.10 แสดงหน้าจอปฏิทินแบบวัน	67
รูปที่ 7.11 แสดงหน้าจอ Insert Event	67
รูปที่ 7.12 แสดงหน้าจอการ Repeat และ Remind	68
รูปที่ 7.13 แสดงหน้าจอที่แสดงเหตุการณ์ต่าง ๆ	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
รูปที่ 7.14 แสดงหน้าจอการ Update และ Delete เหตุการณ์	69
รูปที่ 7.15 แสดงรายละเอียดของกลุ่ม	69
รูปที่ 7.16 แสดงหน้าจอรายละเอียดของกลุ่ม	70
รูปที่ 7.17 แสดงหน้าจอเหตุการณ์ต่าง ๆ ของกลุ่ม	70
รูปที่ 7.18 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงการ Login และสมัครสมาชิก	71
รูปที่ 7.19 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงฟังก์ชันการทำงานและหน้าจอ View Calendar	71
รูปที่ 7.20 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงหน้าจอ Calendar Day	72
รูปที่ 7.21 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงหน้าจอ QuickAdd และ Show Event	72
รูปที่ 7.22 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดง View To Do และรายละเอียดของ To Do	73
รูปที่ 7.23 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงฟังก์ชัน Find และ Result Find	73
รูปที่ 7.24 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงรายละเอียดของกลุ่ม	74
รูปที่ 7.25 แสดงหน้าจอโทรศัพท์ที่แสดงหัวข้อและรายละเอียดของเหตุการณ์ที่มีการนัดหมาย	74
รูปที่ ก-1 แสดงการเช็คค่า System Environment สำหรับการติดตั้ง JDK	76
รูปที่ ก-2 แสดงการ Unzip ไฟล์ที่ได้ดาวน์โหลดมา	77
รูปที่ ก-3 แสดงเซตค่า System Environment TOMCAT_HOME และ JAVA_HOME	77
รูปที่ ก-4 แสดงการรันเว็บเซิร์ฟเวอร์ Tomcat	78
รูปที่ ก-5 แสดงการแก้ไขไฟล์ server.xml	79
รูปที่ ข-1 แสดงภาพการติดตั้งโปรแกรม Oracle Server ขั้นที่ 1	80
รูปที่ ข-2 แสดงภาพการติดตั้งโปรแกรม Oracle Server ขั้นที่ 2	80
รูปที่ ข-3 แสดงภาพการติดตั้งโปรแกรม Oracle Server ขั้นที่ 3	81
รูปที่ ข-4 แสดงภาพการติดตั้งโปรแกรม Oracle Server ขั้นที่ 4	81
รูปที่ ข-4 แสดงภาพการติดตั้งโปรแกรม Oracle Server ขั้นที่ 5	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้าที่
ตารางที่ 7-1 แสดงตารางฐานข้อมูล Calendar	64
ตารางที่ 7-2 แสดงตารางฐานข้อมูล Userdata	64
ตารางที่ 7-3 แสดงตารางฐานข้อมูล Repeat	64
ตารางที่ 7-4 แสดงตารางฐานข้อมูล Remind	64
ตารางที่ 7-5 แสดงตารางฐานข้อมูล To Do	64
ตารางที่ 7-6 แสดงตารางฐานข้อมูล Member	64
ตารางที่ 7-7 แสดงตารางฐานข้อมูล Groupcal	64



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

1.1 ความสำคัญ และความเป็นมาของโครงการ

ในปัจจุบันเทคโนโลยีทางด้านอินเทอร์เน็ตถือว่ามีความสำคัญมากในชีวิตประจำวัน เพราะสามารถติดต่อพูดคุยกันได้ไม่ว่าจะอยู่ที่ไหนของโลก และยังมีความสามารถในการรับส่งข้อมูลต่าง ๆ ด้วยความเร็วสูง และมีประสิทธิภาพ อีกทั้งยังประหยัดเวลาและค่าใช้จ่ายต่าง ๆ การนำเทคโนโลยีทางด้านนี้มาใช้ประโยชน์ต่าง ๆ อย่างมากมาย ทั้งด้านธุรกิจ การศึกษา และด้านบันเทิง เป็นต้น ทำให้อินเทอร์เน็ตทุกวันนี้เป็นที่นิยมใช้กันอย่างแพร่หลาย

สังคมในปัจจุบันทุกนี้มีการแข่งขันกันสูง ดังนั้นจึงต้องแข่งทั้งด้านเวลา ความสามารถ และอื่น ๆ ทำให้เกิดความเร่งรีบและสับสน ดังนั้นจึงมีผู้คิดค้นออร์แกไนเซอร์ (Organizer) ส่วนตัวขึ้นมา เพื่อช่วยเตือนความจำให้เราเปรียบเทียบเป็นเลขส่วนตัว ทำให้เกิดความสะดวกและรวดเร็วในการใช้งานมากยิ่งขึ้น บริษัทยักษ์ใหญ่หลายรายจึงได้เห็นความสำคัญและความจำเป็นด้านออร์แกไนเซอร์ส่วนตัวนี้ จึงมีการคิดค้นเครื่องมือที่ทำหน้าเป็นออร์แกไนเซอร์ส่วนตัวขึ้นมา โดยใช้เทคโนโลยีด้านอิเล็กทรอนิกส์แทนกระดาษในการจดบันทึกเหตุการณ์ต่างอย่างสมัยก่อน ๆ ซึ่งทั้งสะดวกและรวดเร็ว และยังมีหน่วยความจำสูง เครื่องมือชนิดนี้เรียกว่า ปาล์ม (Palm) จากประโยชน์ของเทคโนโลยีด้านอิเล็กทรอนิกส์และปาล์ม จึงเห็นความสำคัญว่าควรจะทำโครงการด้านการนัดหมายผ่านเครือข่ายอินเทอร์เน็ตขึ้น เพราะเครื่องมือปาล์มถึงแม้มีประโยชน์ต่าง ๆ มากมาย แต่ราคาค่อนข้างสูงมาก ทำให้ผู้ที่สามารถใช้ได้จึงมีจำนวนจำกัด เพราะฉะนั้นการทำกรนัดหมายผ่านเครือข่ายอินเทอร์เน็ต ทำให้สามารถใช้ได้อย่างแพร่หลาย อีกทั้งไม่จำกัดจำนวนผู้ใช้ และประหยัดค่าใช้จ่าย แต่ด้วยข้อจำกัดที่ผู้ใช้สามารถใช้บริการนี้ได้ต้องมีเครื่องคอมพิวเตอร์เท่านั้น ซึ่งไม่สะดวกถ้าเกิดความจำเป็นต้องใช้บริการนอกสถานที่ ดังนั้นจึงคิดว่าจะใช้เครื่องโทรศัพท์ที่ไร้สายแทนเครื่องคอมพิวเตอร์ ในการเข้าใช้บริการด้านการนัดหมายผ่านโทรศัพท์ไร้สายโดยใช้เทคโนโลยีเว็บ (WAP) ซึ่งทำให้ผู้ใช้สามารถเข้าใช้บริการ โปรแกรมการนัดหมายผ่านเครือข่ายอินเทอร์เน็ตได้ไม่ว่าจะอยู่ที่ไหนก็ตาม ทั้งสะดวก ประหยัด และมีประสิทธิภาพ นอกจากนี้ด้วยข้อดีของวิธีการออกแบบ และการเขียนโปรแกรมแบบออบเจกต์ - โอเรียนเต็ด (Object-Oriented) นั้น สามารถลดความซ้ำซ้อนของโครงสร้างการเขียนโปรแกรม สนับสนุนการนำกลับมาใช้ใหม่ ง่ายในการตรวจสอบความผิดพลาด และการแก้ไข จึงเป็นการดีที่จะนำมาใช้ในการออกแบบ และเขียนโปรแกรม

ด้วยเหตุนี้โครงการนี้จึงถูกพัฒนาขึ้นเพื่อสร้างออร์แกไนเซอร์ส่วนตัวขึ้นมาใช้งาน โดยผ่านเครือข่ายอินเทอร์เน็ต และผ่านทางโทรศัพท์ไร้สาย โดยใช้เทคโนโลยีเว็บ ขึ้นเพื่อความสะดวก รวดเร็ว ไม่ว่าจะอยู่ที่ไหนก็ตาม และยังสามารถจะให้มีการเรียกเตือน ผ่านทางเพจเจอร์ อีเมลล์ โทรศัพท์ไร้สาย และไอซีคิว (ICQ) ซึ่งเปรียบเทียบมีเลขส่วนตัว โดยใช้การออกแบบการเขียนโปรแกรมแบบออบเจกต์ - โอเรียนเต็ด (Object-Oriented) ด้วยภาษาจาวาเซิร์ฟเฟเร็ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงการ

- เพื่อศึกษาการออกแบบ และการเขียนโปรแกรมแบบออบเจกต์ – โอเรียนเต็ลใช้ภาษาจาวา
- เพื่อศึกษาวิธีการสร้าง และออกแบบเว็บไซต์ (Web Site)
- เพื่อศึกษาโครงสร้างและวิธีการเขียน โปรแกรมที่ใช้ในการสร้างเว็บไซต์ (Web Side)
- เพื่อศึกษาโครงสร้างและวิธีการเขียนโปรแกรมที่ใช้ในการสร้างเว็บไซต์ (WAP Side)
- เพื่อศึกษาการเขียน โปรแกรมภาษา Java Servlet ในการสร้างแอปพลิเคชันต่าง ๆ บนเว็บ และเว็บ
- เพื่อศึกษาโครงสร้างและวิธีการเขียนโปรแกรมภาษา WML เพื่อแสดงผลบนหน้าจอของโทรศัพท์มือถือ
- เพื่อพัฒนาและออกแบบการเขียนโปรแกรมด้วยภาษาจาวาเพื่อการจัดการฐานข้อมูลผ่านเครือข่ายอินเทอร์เน็ตและเทคโนโลยีเว็บ
- เพื่อศึกษาโปรแกรมการจัดการฐานข้อมูล โดยใช้ภาษา SQL มาควบคุมออราเคิล8ไอ
- เพื่อศึกษาหลักการพื้นฐานของ HTTP โพรโตคอล เพื่อใช้ในการพัฒนาการเขียนโปรแกรมเน็ตเวิร์กเรียกเตือนเหตุการณ์สำคัญต่าง ๆ ได้ โดยผ่านทางเพจเจอร์, อีเมล, โทรศัพท์มือถือ และไอซีคิว (ICQ)

1.3 ขอบเขตของโครงการ

โครงการนี้ออกแบบและเขียนโปรแกรมออบเจกต์ โอเรียนเต็ลโดยใช้ภาษาจาวาเซิร์ฟเรต และออกแบบฐานข้อมูลโดยใช้อีอาร์ไดอะแกรม(ER Diagram) เพื่อใช้เป็นฐานข้อมูลของผู้ที่เข้าใช้บริการนัดหมาย ในการติดต่อผู้ใช้จะเลือกใช้งานในรูปแบบของการเชื่อมต่อผ่านทางเว็บเพจ และเว็บเพจ ส่วนโปรแกรมการจัดการฐานข้อมูลจะใช้ออราเคิล8ไอ (Oracle8I) โดยใช้ภาษาจาวาเซิร์ฟเรตเป็นตัวติดต่อ และใช้ภาษาเอสคิวแอล (SQL) เป็นตัวควบคุม ในส่วนของการสร้างส่วนที่ติดต่อกับผู้ใช้ (User Interface) จะใช้จาวาเซิร์ฟเรตสร้างเป็นเอกสาร HTML และ WML เพื่อติดต่อกับผู้ใช้ผ่านเว็บ (Web) และเว็บ (WAP) ตามลำดับ นอกจากนี้จะใช้จาวาเซิร์ฟเรตทำส่วนที่ติดต่อกับผู้ใช้แล้ว ยังใช้เป็นส่วนที่ทำหน้าที่คล้าย CGI (Common Gateway Interface) เพื่อจัดการกับข้อมูลต่างๆ นอกจากนี้ยังได้ทำการศึกษาการเขียนโปรแกรมด้านทางเน็ตเวิร์ก (Network Programming) เพื่อทำการเขียนโปรแกรมส่งเพจเจอร์ และโทรศัพท์มือถือ ด้วยภาษาจาวา

แต่โครงการนี้ถือว่าเป็นการทดลองสร้าง ดังนั้นเครื่องเซิร์ฟเวอร์ที่ใช้จึงใช้เครื่องของนักศึกษาเอง ทำให้บางครั้งผู้ใช้อาจไม่สามารถเข้าใช้บริการได้ และอาจเกิดความไม่แน่นอนของข้อมูล แต่ถึงอย่างไรก็รองรับต่อความต้องการของผู้ใช้อย่างแน่นอน ส่วนของเว็บนั้น จะใช้เว็บเบราว์เซอร์แทนการทำงานของโทรศัพท์มือถือจริง เพราะถ้าทดลองใช้โทรศัพท์มือถือ ต้องเสียค่าใช้จ่ายเป็นจำนวนมาก แต่ถึงอย่างไรก็สามารถทำงานอย่างสมบูรณ์แน่นอนเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับโครงการ ซึ่งก็มีด้วยกันหลายเรื่องเช่น การเขียนโปรแกรมจาวาเซิร์ฟเวร์ตเพื่อจัดการค่าเบส การออกแบบ และการเขียน (ER Diagram) อีอาร์ไดอะแกรม การเขียนโปรแกรมภาษา WML (Wireless Markup Language) เพื่อใช้กับเทคโนโลยีเว็บ (WAP) HTTP โพรโตคอลพื้นฐาน เพื่อเขียนเน็ตเวิร์คโปรแกรมมิ่ง(Network Programming) ส่งข้อความเตือนผ่านโดยผ่านทางเพจเจอร์, อีเมล, โทรศัพท์ไร้สาย และไอซีคิว (ICQ) ซึ่งมีรายละเอียดดังในบทที่ 6 จากนั้นก็นำเอาความรู้ที่ได้ศึกษาทั้งหมดมาพัฒนา และออกแบบระบบซึ่งมีรายละเอียดในบทที่ 7 โดยเริ่มจากการติดตั้งโปรแกรม และอุปกรณ์ต่างๆ ที่ใช้ในโครงการซึ่งรายละเอียดมีอยู่ในภาคผนวก ต่อก็เขียนโปรแกรมเพื่อพัฒนาระบบขึ้นมา ในขั้นตอนนี้ยังมีการตรวจสอบความผิดพลาดต่างที่เกิดขึ้น เพื่อแก้ไขให้โปรแกรมมีความถูกต้อง และตรงความต้องการมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

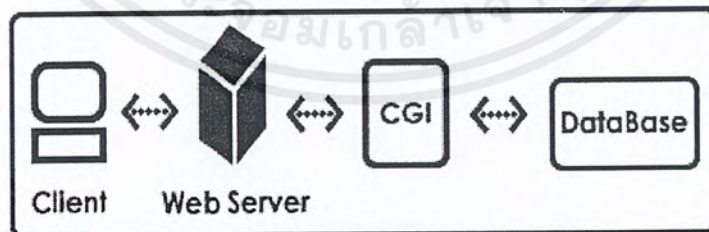
บทที่ 2 จาวาเซิร์ฟเว็ต (Java Servlet)

2.1 เว็บโปรแกรมมิง (Web Programming)

แม้ว่าโลกของอินเทอร์เน็ตจะเพิ่งเกิดขึ้นเพียงไม่กี่ปีก็ตาม เทคโนโลยีที่ใช้กับอินเทอร์เน็ต กลับมีการเปลี่ยนแปลงไปอย่างรวดเร็วมาก จำได้ว่าสมัยแรกๆ เพจต่างๆ ที่อยู่ในเว็บจะเป็นลักษณะของสแตติกเพจเว็บเพจ (Static Web Page) ที่ไม่มีการเปลี่ยนแปลงเนื้อหาไม่ว่าจะนานเท่าไรหรือนอกเสียจากว่าผู้ดูแลเพจนั้นจะทำการอัปเดตเว็บเพจดังกล่าว เว็บเพจลักษณะนี้เป็นเว็บเพจที่นิยมใช้กันทั่วไปในอินเทอร์เน็ต สมัยแรกเพราะอินเทอร์เน็ตยังนิยมกันอยู่ในวงแคบ โดยกลุ่มผู้ใช้จะเป็นกลุ่มบุคคลที่อยู่ในวงการศึกษานั้น ต่อมาจากนั้นไม่นานทางผู้ผลิตบราวเซอร์ได้ทำการเพิ่มความสามารถให้กับเว็บเพจโดยอนุญาตให้เว็บเพจสามารถแทรก สคริปต์เล็ก ๆ ลงไปรวมกับส่วนที่เป็น HTML ได้ซึ่งจุดนี้ก็คือจุดเริ่มต้นของ Client Side JavaScript นั่นเอง แม้ว่าเพจจะเริ่มมีความสามารถในการโต้ตอบกับผู้ใช้โดยอิงความสามารถจาก JavaScript แล้วก็ตาม ถ้าพูดถึงในแง่ของส่วนเนื้อหาของตัวเว็บเพจจริงๆ แล้ว ตัวเว็บเพจเองก็ยังคงเป็นสแตติกเว็บเพจ (Static Web Page) อยู่เช่นเดิม เมื่อกลุ่มผู้ใช้อินเทอร์เน็ตเริ่มมีมากขึ้นความต้องการที่จะให้เว็บเพจสามารถทำการรับส่งข้อมูลรวมไปถึงเปลี่ยนแปลงเนื้อหาได้โดยอัตโนมัติก็เกิดขึ้น เทคโนโลยีที่เกิดขึ้นเพื่อรองรับความต้องการเหล่านี้ก็คือ Server Side Application นั่นเอง



รูปที่ 2.1 แสดงการทำงานของเว็บเพจแบบสแตติก (Static Web Page)



รูปที่ 2.2 แสดงการทำงานของเว็บเพจแบบไดนามิก (Dynamic Web Page)

ข้อแตกต่าง และลักษณะของ ไคลเอนต์ไซด์สคริปต์ (Client Side Script) กับ เซิร์ฟเวอร์ไซด์สคริปต์ (Server Side Script)

2.1.1 ไคลเอนต์ไซด์สคริปต์ (Client Side Script)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือการเขียนโปรแกรมให้ทำงานที่ฝั่งไคลเอนต์ ซึ่งในที่นี้ก็หมายถึง เว็บเบราว์เซอร์ของเครื่องผู้ใช้นั้นเอง เช่น จาวาสคริปต์ (JavaScript) , วิวาลเบสิกสคริปต์ (VisualBasic Script) และ ไดนามิก HTML (Dynamic HTML)

ลักษณะของไคลเอนต์ไซด์สคริปต์ (Client Side Script)

สามารถเขียนสคริปต์แทรกลงในไฟล์ HTML ได้เลย โดยไม่ต้องติดตั้งโปรแกรม หรือเครื่องมือใดๆ เพิ่มเติม

- สคริปต์ที่เขียนขึ้นมา นั้น ผู้เข้าเยี่ยมชมสามารถมองเห็นซอสโค้ด(Source Code) ได้ โดยใช้คำสั่ง View Source ที่เว็บเบราว์เซอร์
- ไม่ขึ้นกับระบบปฏิบัติการ ไม่ว่าเครื่องของผู้ใช้จะใช้ระบบปฏิบัติการใดๆ เช่น วินโดวส์ ยูนิกซ์ หรือ แมคอินทอชก็ใช้ได้เหมือนกัน
- ขึ้นอยู่กับเบราว์เซอร์ของผู้ใช้ เบราว์เซอร์รุ่นเก่าอาจไม่รองรับการทำงานของสคริปต์ที่เราเขียน

2.1.2 เซิร์ฟเวอร์ไซด์สคริปต์ (Server Side Script)

คือ โปรแกรมประเภท CGI (Common Gateway Interface) ซึ่งเป็นกลไกในการที่จะทำให้เครื่องคอมพิวเตอร์ฝั่งไคลเอนต์ หรือเครื่องของผู้ใช้สามารถติดต่อกับเครื่องคอมพิวเตอร์ฝั่งเซิร์ฟเวอร์ได้โดยรับข้อมูลจากผู้ใช้งานที่เว็บเบราว์เซอร์ แล้วนำมาประมวลผล เช่น ค้นหาข้อมูลในฐานข้อมูลแล้วส่งผลลัพธ์ที่ได้จากการประมวลผลนั้นกลับไปยังผู้ใช้อีกครั้งหนึ่ง ลักษณะของ CGI ที่นิยมใช้กันได้แก่ การทำเว็บบอร์ด (Web Board) การพูดคุยผ่านทางเว็บเบราว์เซอร์(Chat) การส่งอีเมล(Email) เป็นต้น ภาษาคอมพิวเตอร์ที่นิยมใช้เป็น CGI ได้แก่ Practical Extraction And Reporting Language (PERL) , Active Server Page (ASP) , Personal Home Page (PHP) , Visual Basic , JSP , Servlet

ลักษณะของเซิร์ฟเวอร์ไซด์สคริปต์(Server Side Script)

- ต้องติดตั้งโปรแกรมที่เครื่องเซิร์ฟเวอร์ ไม่สามารถรันได้ด้วยตัวเอง
- เนื่องจากการประมวลผลที่เครื่องเซิร์ฟเวอร์ จึงทำให้เกิดปัญหาในส่วนของระบบปฏิบัติการ และ ชนิดของเว็บเบราว์เซอร์ที่ไคลเอนต์ไปได้
- มองไม่เห็นซอสโค้ดของภาษาสคริปต์ เมื่อใช้คำสั่ง View Code ที่เบราว์เซอร์ จะเห็นเป็นเหมือนไฟล์ HTML ธรรมดา
- มีนามสกุลตามภาษาแต่ละสคริปต์ เช่น PERL มีนามสกุลเป็น *.PL หรือใช้ *.CGI , ASP นามสกุลเป็น *.ASP , PHP มีนามสกุลเป็น *.PHP

เซิร์ฟเวอร์ไซด์แอปพลิเคชัน (Server Side Application) ในระยะแรก ๆ มักถูกเขียนขึ้นด้วยคอนเซ็ปต์ของ CGI (Common Gateway Interface) โดยหลักการทำงานง่าย ๆ ก็คือเว็บเบราว์เซอร์จะทำการส่งข้อมูลที่เกิดจากการกระทำของ User เช่น การคลิกลิงค์หรือการกรอกแบบสอบถามไปยังเว็บเซิร์ฟเวอร์ โดยแทนที่เว็บเซิร์ฟเวอร์จะทำการส่งเว็บเพจที่เป็นสแตติกเว็บเพจ (Static Web Page) กลับมา เว็บเซิร์ฟเวอร์จะทำการส่งข้อมูลดังกล่าวไปยังโปรแกรมซึ่งถูกจัดไว้ โปรแกรมดังกล่าวจะทำการประมวลผลข้อมูลที่ได้นี้แล้วจะส่งผลกลับไปยังเว็บเซิร์ฟเวอร์ ซึ่งทางเว็บเซิร์ฟเวอร์ก็จะส่งผลที่ได้นี้กลับไปยังเว็บเบราว์เซอร์อีกทีหนึ่ง หลาย ๆ คนคงเคยเห็นเซิร์ฟเวอร์ไซด์แอปพลิเคชัน (Server Side Application) ที่เป็นผลผลิตของ CGI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรุ่นแรก ๆ ที่ยังหลงเหลืออยู่ในปัจจุบันนี้ ยกตัวอย่างเช่น Counter, Image Map, Guestbook, SendMail เป็นต้น จริง ๆ แล้วตัว CGI เองสามารถเขียนด้วยภาษาอะไรก็ได้ แต่ที่นิยมมากที่สุดเห็นจะเป็น C และ Perl อาจจะเป็นเพราะว่า CGI เป็นสิ่งที่มีมากับอินเทอร์เน็ต ตั้งแต่ช่วงแรก ๆ ดังนั้นไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์ไหนก็ตาม Server Side Application พื้นฐานที่ทางเว็บเซิร์ฟเวอร์เหล่านั้นจะต้องสนับสนุนก็คือ CGI ซึ่งจุดนี้เองที่เป็นจุดเด่นทำให้ CGI เป็นที่นิยมใช้กันอย่างกว้างขวางจนกระทั่งปัจจุบันนี้

2.2 จาวาเซิร์ฟเว็ต (Java Servlet)

เป็นที่รู้จักกันว่า CGI หรือ Common Gateway Interface ถูกนำมาใช้ประโยชน์ในเว็บแอปพลิเคชันอย่างมากมาย ในการเขียนโปรแกรม CGI สามารถเขียนขึ้นด้วยหลายภาษาเช่น C , C++ , Perl , PHP หรือแม้แต่จาวาก็สามารถนำมาเขียนได้ แต่โปรแกรม CGI มีข้อจำกัดต่างเช่น

1. โปรแกรมจะสร้างโปรเซส 1 โปรเซส ต่อ 1 การร้องขอ (Request) ทำให้การทำงานแต่ละอย่างจะเปลืองทรัพยากรของระบบมาก ประสิทธิภาพในการทำงานจะลดลงอย่างมากเมื่อมีการร้องขอ (Request) จากไคลเอนต์เข้ามามากๆ

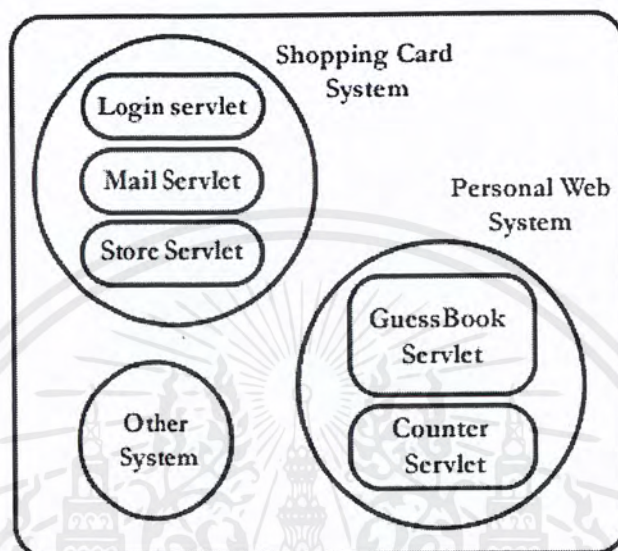
2. ไม่สนับสนุนการออกแบบ และการเขียนโปรแกรมแบบ Object Oriented เท่าที่ควร จึงไม่ได้ข้อดีของการเขียนโปรแกรมแบบ Object Oriented เช่น การนำกลับมาใช้ใหม่ ง่ายต่อการพัฒนา และการเปลี่ยนแปลงที่จะเกิดขึ้น เป็นต้น

Servlet เป็น Server Side Application แบบหนึ่งซึ่งอ้างอิงคอนเซ็ปต์มาจาก CGI ข้อดีของ Servlet ที่อยู่เหนือ CGI อย่างแรกก็คือตัวภาษาที่ใช้เขียนซึ่งก็คือจาวานั้นเอง จาวาเป็นภาษาที่ใช้คอนเซ็ปของ ออบเจกต์-โอเรียนเต็ด (Object Oriented) ซึ่งสามารถลดความซับซ้อนของโครงสร้างโปรแกรมรวมถึงการอำนวยความสะดวกในการ reuse ส่วนของโปรแกรมที่เขียนไว้แล้ว นอกจากนี้จาวายังเป็นภาษาที่เป็นลักษณะแบบ Platform Independent ซึ่งจะช่วยให้เราสามารถที่จะทำการพัฒนาระบบโดยใช้บนแพลตฟอร์มอะไรก็ได้ โดยจะนำโปรแกรมที่เขียนเสร็จแล้วมารันบน Unix Environment เพื่อเพิ่มประสิทธิภาพของโปรแกรมแทนก็ได้ นอกจากนี้ Servlet ยังมีความเร็วที่สูงกว่า CGI เพราะ Servlet ใช้หลักการของเทรด โดยจะทำการสร้าง 1 เทรดต่อหนึ่ง Request ที่มาจากไคลเอนต์ ซึ่งในทางกลับกัน CGI จะทำการสร้าง 1 โปรเซส ต่อหนึ่ง Request ซึ่งจะทำให้เปลืองทรัพยากรมากกว่า และโปรเซสในการรันก็จะช้ากว่าด้วย ท้ายที่สุดจุดเด่นที่สำคัญของ Servlet ก็คือ API (Application Programming Interface) โดยระบบที่ทำการพัฒนาโดยใช้คอนเซ็ปต์ของ Servlet จะสามารถเรียกใช้ API ที่ทางจาวามีมาให้ (javax.servlet.*, javax.servlet.http.*) ซึ่งจะช่วยให้การพัฒนาระบบดังกล่าวง่ายและเร็วยิ่งขึ้น

2.2.1 Servlet Engine

ในการรันระบบ*ที่เขียนขึ้นโดยใช้หลักการของ Servlet เราจะต้องนำระบบดังกล่าวมาบรรจุอยู่ในสิ่ง ๆ หนึ่งที่เรียกว่า Servlet Engine ใหนักถึง Servlet Engine คล้าย ๆ กับกล่อง ๆ หนึ่งที่ใส่ลูกปิงปองไว้หลายลูก โดยลูกปิงปองแต่ละลูกก็คือระบบ ๆ หนึ่งนั่นเอง หลายคนอาจสงสัยทำไมถึงใช้คำว่าระบบ โดยทั่วไปเซิร์ฟเวอร์ไซด์แอปพลิเคชัน(Server Side Application)หนึ่ง ๆ ที่ถูกเขียนขึ้นโดยใช้ Servlet API จะถูกเรียกว่า Servlet ในหนึ่งระบบอาจประกอบด้วย Servlet หลายอัน ยกตัวอย่างเช่น ระบบที่เกี่ยวข้องกับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shopping Cart อาจจะประกอบด้วย Servlet ที่ทำหน้าที่ในการเช็คสต็อกสินค้า, Servlet ที่ทำหน้าที่ในการเก็บข้อมูลสินค้า, Servlet ที่ทำหน้าที่ในการส่งอีเมลกลับไปยังลูกค้าเพื่อบอกว่าได้ทำการส่งของไปให้แล้ว เป็นต้น ดังนั้นถ้ามองโดยรวมแล้ว Servlet Engine ก็คือที่รวมของระบบตั้งแต่หนึ่งระบบถึงหลายระบบ โดยแต่ละระบบจะประกอบด้วย Servlet หนึ่งอันหรือมากกว่า ดังรูปที่ ระบบ ในที่นี้อาจจะหมายถึง Zone Apache JServ หรือ Web Application (Tomcat) ก็ได้



รูปที่ 2.3 แสดงถึง Servlet Engine and Servlets ภายในระบบ Servlet Engine / Servlet Container

Servlet Engine เป็นเพียงกล่อง ๆ หนึ่งที่ใส่บรรจุและรันกลุ่มของ Servlet เท่านั้น ในการที่จะทำการติดต่อสื่อสารกับ ไคลเอนต์ ตัว Servlet Engine นี้จะต้องทำงานร่วมกับเว็บเซิร์ฟเวอร์ซึ่งเปรียบเสมือนฉากหน้าที่ติดต่อกับไคลเอนต์ อีกทีหนึ่ง เมื่อใดก็ตามที่มี Request ส่งมาจากไคลเอนต์ ถ้า Request นั้นจะจูงมาที่ตัว Servlet ทางเว็บเซิร์ฟเวอร์ ก็จะทำการส่ง ตัว Request นั้นมาให้ Servlet Engine ซึ่งทาง Servlet Engine ก็จะทำการเรียก Servlet ที่ต้องการขึ้นมาทำการประมวลผล Request นั้น โดยท้ายสุด Servlet จะส่งผลกลับไปให้ Servlet Engine Servlet Engine ก็จะส่งผลที่ได้กลับไปให้เว็บเซิร์ฟเวอร์ ซึ่งเว็บเซิร์ฟเวอร์ก็จะส่งผลกลับไปให้ไคลเอนต์

Servlet Engine อาจจะเป็นส่วนที่ติดมากับเว็บเซิร์ฟเวอร์อยู่แล้วยกตัวอย่างเช่น Servlet Engine ที่อยู่ใน Netscape Enterprise Server , IBM WebSphere หรืออาจจะเป็นส่วนที่เป็น Add-on ให้กับเว็บเซิร์ฟเวอร์ก็ได้เช่น Apache JServ, Tomcat , JRun หรือแม้กระทั่งเป็นส่วนหนึ่งที่อยู่ในเว็บแอปพลิเคชัน เช่น BEA Weblogic เป็นต้น ทั้งนี้การเลือกใช้ Servlet Engine แต่ละชนิดก็มักขึ้นอยู่กับปัจจัยหลายอย่างเช่น ความสะดวกในการรวมระบบที่จะสร้างขึ้นใหม่กับระบบที่มีอยู่แล้ว, งบประมาณที่มีอยู่สำหรับโครงการ หรืออาจจะรวมไปถึงทักษะและประสบการณ์ส่วนตัวของนักพัฒนาแต่ละคน

2.2.2 วิธีการเขียนโปรแกรมเซิร์ฟเว็ท Servlet

หลักการการทำงานของ Servlet จริง ๆ อยู่ที่ interface ที่ชื่อ javax.servlet.Servlet* โดยทุก Servlet ที่ถูกเขียนขึ้นจะต้องทำการ implement ตัว interface นี้ไม่ว่าทางตรงก็ทางอ้อม ทางตรงก็คือการ implement ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

interface นี้เลย ส่วนทางอ้อมก็คือการให้ Servlet ทำการ subclass คลาสบางคลาสที่ได้ทำการ implement ตัว interface นี้ไว้แล้ว เหตุผลว่าทำไมต้อง implement ตัว interface นี้เพราะว่าเมื่อไรก็ตามที่มี request จาก ไคลเอนต์เข้ามาถึง Servlet Engine ตัว Servlet Engine จะทำการหา Servlet ที่ Request ดังกล่าวอ้างถึง หลังจากนั้น Servlet Engine จะทำการเรียกฟังก์ชันต่าง ๆ ที่อยู่ใน Servlet เพื่อทำการประมวลผล Request ของ ไคลเอนต์ โดยฟังก์ชันที่ Servlet Engine จะทำการเรียกก็คือฟังก์ชันที่ Servlet ได้ทำการ implement ซึ่งเป็นฟังก์ชันที่ถูก define อยู่ใน javax.servlet.Servlet interface นั่นเอง หลายคนอาจจะถามว่าทำไม Servlet Engine ถึงเรียกฟังก์ชันที่ถูก define อยู่ใน interface นี้ เหตุผลง่าย ๆ ก็คือ Servlet เป็นส่วนที่ถูกโหลดเข้าไปใน Servlet Engine ในช่วง Runtime ตัว Servlet Engine เองไม่สามารถทราบได้ว่า Servlet ต่าง ๆ มีฟังก์ชันอะไรประกอบอยู่บ้างนอกเสียจากว่า Servlet นั้นได้ทำการ implement ฟังก์ชันที่เป็นมาตรฐานที่ Servlet Engine รับรู้ซึ่งนี่ก็คือเหตุผลว่าทำไมทุก Servlet จะต้องทำการ implement ตัว javax.servlet.Servlet interface อย่างไรก็ตามเราสามารถให้ Servlet Engine เรียกฟังก์ชันอื่น ๆ ที่อยู่ใน Servlet ได้ซึ่งวิธีการก็คือการใส่ฟังก์ชันดังกล่าวเข้าไปในส่วน implementation ของฟังก์ชันต่าง ๆ ที่ถูกระบุอยู่ใน javax.servlet.Servlet interface นั่นเอง

ดังนั้นหลักการง่าย ๆ ในการสร้าง Servlet ก็คือการสร้างคลาสขึ้นมาคลาสหนึ่งโดยคลาสนั้นจะต้องทำการ implement ตัว interface ที่ชื่อ javax.servlet.Servlet เพียงเท่านี้ก็จะได้ Servlet เป็นของตัวเองแล้ว อย่างที่กล่าวมาข้างต้นว่าในการเขียน Servlet เราอาจจะทำการ implement ตัว javax.servlet.Servlet interface ไม่ทางตรงก็ทางอ้อม เพื่อที่จะอำนวยความสะดวกให้กับนักพัฒนา ทางจาวาจึงได้มีการสร้างคลาสพื้นฐานที่ได้ทำการ implement ตัว javax.servlet.Servlet interface ขึ้นมาสองคลาสคือคลาส javax.servlet.GenericServlet และคลาส javax.servlet.http.HttpServlet (ซึ่งเป็น subclass ของ javax.servlet.GenericServlet อีกทีหนึ่ง) ดังนั้นสิ่งที่ต้องทำก็คือการ subclass คลาสใดคลาสหนึ่งในสองคลาสนี้แล้ว override ฟังก์ชันที่ต้องการซึ่งโดยทั่วไปก็คือฟังก์ชันที่ชื่อ service() นั่นเอง ถ้าดูความสัมพันธ์ระหว่าง javax.servlet.Servlet (interface) javax.servlet.http.GenericServlet (abstract class) และ javax.servlet.http.HttpServlet (abstract class) เราจะเห็นว่าเริ่มแรก javax.servlet.Servlet จะถูก define ด้วย 5 ฟังก์ชันพื้นฐานที่ทุก Servlet จะต้องทำการ implement ดังลิสของ 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 จะทำการ implement ฟังก์ชันต่าง ๆ ที่อยู่ใน javax.servlet.Servlet ดังตัวหน้าที่ปรากฏอยู่ข้างล่าง ซึ่งนอกจากนี้คลาส GenericServlet ยังเพิ่มเติมฟังก์ชันเสริมอีกส่วนหนึ่งเพื่อช่วยให้นักพัฒนาทำการพัฒนา Servlet ได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น

```
public abstract class GenericServlet implements Servlet
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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();
จะสังเกตว่าคลาส GenericServlet เป็น abstract คลาสเพราะฟังก์ชัน public void service(ServletRequest
req,...) ของ javax.servlet.Servlet interface จะไม่ถูก implement โดยคลาสนี้ ท้ายที่สุดคลาส
GenericServlet จะถูก subclass อีกทีหนึ่งโดยคลาส HttpServlet ซึ่งเป็นคลาสที่ถูกสร้างขึ้นเพื่อรองรับ
Servlet ที่ถูกเขียนขึ้นสำหรับการติดต่อสื่อสารกับไคลเอนต์ที่ใช้ HTTP โพรโตคอล ดังนั้นในการ
เขียน Servlet ของ ก็เพียงทำการ subclass คลาส 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;
protected void doTrace(HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException;
protected void service(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public void service(ServletRequest req, ServletResponse res) throws ServletException,
IOException;
```

```
protected long getLastModified(HttpServletRequest req);
```

คลาส HttpServlet ถูกกำหนดให้เป็น abstract คลาสไม่ใช่เพราะว่ามีบางฟังก์ชันในตัวมันที่ไม่ได้มีการ implement แต่เพราะว่าผู้ที่เขียนคลาสนี้ขึ้นมาไม่ต้องการให้เราทำการเรียกใช้คลาสนี้โดยตรงแต่ต้องการให้เราสร้าง Servlet ขึ้นมาจากการ subclass คลาสนี้อีกหนึ่ง

ตัวอย่าง การเขียนโปรแกรม servlet Hello World

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

```

คลาสนี้เป็นคลาสที่ subclass คลาส GenericServlet โดยทำการ override ฟังก์ชัน public void service (ServletRequest request, ServletResponse response)... ซึ่งเป็นฟังก์ชันที่อยู่ใน Servlet interface ฟังก์ชันนี้เป็นฟังก์ชันที่สำคัญที่สุดสำหรับทุก Servlet เพราะเมื่อใดก็ตามที่มี Request เข้ามายัง Servlet ฟังก์ชัน Service ที่อยู่ใน Servlet ดังกล่าวจะถูกเรียกโดย Servlet Engine เพื่อทำการประมวลผล Request ที่เข้ามาเสมอ GenericServlet เป็น abstract คลาสที่ทำการ implement ตัว Servlet interface โดยทิ้งฟังก์ชัน service ให้เป็น abstract คือ ไม่มีส่วนที่เป็น implementation ซึ่งทางคลาสนี้ที่เป็น subclass ของ GenericServlet จะต้องมีหน้าที่ในการ implement ฟังก์ชันนี้ยกตัวอย่างเช่นคลาส HelloWorld ของเป็นต้น

Code ที่อยู่ใน HelloWorld ในสามบรรทัดแรกเราจะเห็นว่าจะมีสาม package ที่ถูก import เข้ามาซึ่งก็คือ java.io.*, javax.servlet.* และ javax.servlet.http.* ที่เรียก package java.io เข้ามาเพราะว่าต้องการใช้

คลาส IOException และ คลาส PrintWriter ส่วน package javax.servlet และ javax.servlet.http จะใช้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเรียกคลาสที่เกี่ยวข้องกับ Servlet ทั้งหมด ส่วน code ที่อยู่ในฟังก์ชัน service ตัวฟังก์ชัน service จะมีอาร์กิวเมนต์อยู่ด้วยกันสองตัวคือ ServletRequest และ ServletResponse โดยเมื่อใดก็ตามที่ฟังก์ชัน service ถูกเรียกโดย Servlet Engine ตัว Servlet Engine จะทำการส่งผ่านออบเจกต์มายังอาร์กิวเมนต์เหล่านี้ ซึ่งก็คือ ServletRequest และ ServletResponse ออบเจกต์นั่นเอง ServletResponse ออบเจกต์เป็นตัวที่เก็บรายละเอียดของ Request ที่ส่งมาโดยไคลเอนต์ทั้งหมดโดย Servlet สามารถใช้ออบเจกต์นี้เพื่ออ่านพารามิเตอร์ต่าง ๆ ที่อยู่ใน Request เข้ามาประมวลผลได้ หลังจากที่ Servlet ทำการประมวลผลเสร็จแล้ว Servlet ก็สามารถส่งผลที่ได้กลับออกไปให้กับไคลเอนต์ได้โดยใช้ ServletResponse ออบเจกต์ อย่างไรก็ตาม ฟังก์ชัน Service ที่อยู่ใน Hello World Servlet ของเราไม่ได้มีการประมวลผลใด ๆ เพียงแต่จะส่ง HTML Stream ที่เขียนคำว่า "Hello World" กลับไปให้ Client เท่านั้น

ในการส่งผลต่าง ๆ กลับไปให้ไคลเอนต์ ขั้นตอนแรกที่ Servlet ต้องทำการเซต Header โดยในตัวอย่างของเรา Header ที่ถูกส่งกลับไปที่คือ Content Type (MIME Type) Header ซึ่งเป็นตัวบอกไคลเอนต์ว่าข้อมูลที่กำลังจะได้รับเป็น ข้อมูลชนิดไหน ให้นึกถึง MIME Type เป็นลักษณะคล้าย ๆ กับไฟล์ extension ยกตัวอย่างเช่น text/html จะบอกว่าข้อมูลที่กำลังส่งออกไปเป็น HTML Stream หรือ image/gif จะบอกว่าข้อมูลที่ส่งออกไปเป็น Stream ของ gif image เป็นต้น ตัว Servlet สามารถทำการเซต Content Type ได้โดยใช้ฟังก์ชัน setContentType() ซึ่งเป็นฟังก์ชันที่อยู่ใน ServletResponse นั่นเอง ในการส่งข้อมูลต่าง ๆ ไปให้ Client ตัว Servlet สามารถส่งข้อมูลออกไปได้สองแบบคือ text Stream และ byte Stream ในกรณีของเรา ๆ จะส่งข้อมูลออกไปเป็นแบบ text Stream ดังนั้นคลาสที่เราจะใช้ในการเขียนลงไปที่ ServletResponse ก็คือคลาส PrintWriter นั่นเอง ในการเรียกใช้คลาส PrintWriter สำหรับเขียนลงไปที่ ServletResponse เราก็สามารถทำได้ง่าย ๆ โดยการเรียกฟังก์ชัน ServletResponse.getWriter() ซึ่งจะคืนค่าตัว PrintWriter อินสแตนซ์การเขียนเอาต์พุตต่าง ๆ ลงไปยังตัว ServletResponse ที่เราใช้สำหรับเรียก อินสแตนซ์ ดังกล่าวขึ้นมา ท้ายสุด HelloWorld Servlet จะใช้ PrintWriter ทำการพิมพ์ text ต่าง ๆ ที่ใช้ในการสร้าง HTML Output Stream ที่เขียนคำว่า "Hello World" ออกไปให้ไคลเอนต์

ตัวอย่างที่ 2 เป็นตัวอย่าง HelloWorld อีกแบบแต่แทนที่คลาส Servlet ของเราจะทำการ subclass คลาส GenericServlet เราจะทำการ subclass คลาส HttpServlet ซึ่งเป็น subclass ของคลาส GenericServlet อีกทีหนึ่งแทน คลาส GenericServlet และคลาส HttpServlet จะต่างกันตรงที่ว่าคลาส GenericServlet จะเป็นคลาสที่ไม่เจาะจงโพรโตคอลใดโพรโตคอลหนึ่ง สำหรับ Servlet ที่ต้องการ implement ตัว Servlet ที่ใช้สำหรับโพรโตคอลจำเพาะเจาะจงขึ้นมาเองยกตัวอย่างเช่น Servlet สำหรับ RMI หรือแม้กระทั่ง Servlet สำหรับ Http โพรโตคอลซึ่งก็คือ HttpServlet ตัวอย่างเช่น

```
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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<body>");
out.println("<head>");
out.println("<title>Hello World2</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Hello World2</h1>");
out.println("</body>");
out.println("</html>");
} }

```

จะสังเกตเห็นว่าฟังก์ชันที่ทำการ override ก็คือฟังก์ชัน service แต่ต่างกันตรงที่ว่าฟังก์ชันนี้ไม่ใช่ฟังก์ชัน service(ServletRequest req, ServletResponse res)... แต่กลับกลายเป็น service(HttpServletRequest req, HttpServletResponse res)... แทนซึ่งเป็นฟังก์ชันที่จำเพาะจงสำหรับ Http โพรโตคอลโดยเฉพาะ อาจมีคำถามว่าฟังก์ชัน service(HttpServletRequest req, HttpServletResponse res)... จะสามารถถูกเรียกโดย Servlet Engine ได้อย่างไรในเมื่อ Servlet Engine จะเรียกเฉพาะฟังก์ชัน service(ServletRequest req, ServletResponse res)... ซึ่งเป็นฟังก์ชันที่ถูก define อยู่ใน javax.servlet.Servlet interface เท่านั้น ถ้าเข้าไปดู code ที่อยู่ใน HttpServlet จะเห็นว่าเป็นลักษณะคล้าย ๆ อย่างนี้

```

public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException {
    HttpServletRequest request;
    HttpServletResponse response;
    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 ก็ยังคงทำการ implement ฟังก์ชัน service(ServletRequest req,...) ซึ่งเป็นฟังก์ชันที่อยู่ใน javax.servlet.Servlet interface อยู่เพียงแต่ว่าเมื่อใดก็ตามที่มีการเรียกฟังก์ชันนี้โดย Servlet Engine ทาง HttpServlet จะทำการ cast ตัว ServletRequest และ ServletResponse ให้กลายเป็น HttpServletRequest และ HttpServletResponse ตามลำดับ หลังจากนั้นจะส่งผ่านออบเจกต์ สองตัวนี้เข้าไปยัง service(request, response) ซึ่งก็คือฟังก์ชัน service(HttpServletRequest req, HttpServletResponse res) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

res)... อีกต่อหนึ่ง โดยโค้ดที่พิมพ์คำว่า HelloWorld ที่อยู่ในฟังก์ชันนี้ก็จะถูกเรียกโดย Servlet Engine ในท้ายสุด

ตัวอย่าง HelloWorld Servlet สุดท้ายที่กำลังจะกล่าวถึงนี้เป็นตัวอย่างของ HelloWorld Servlet ที่ถูกยกขึ้นมาใช้เป็นตัวอย่างของ HelloWorld Servlet ในแทบทุก Servlet Tutorial ที่มีอยู่ในโลกนี้ซึ่งที่มาที่ไปของ Servlet นี้ก็มาจากสอง 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>");
        out.println("<body>");
        out.println("<h1>Hello World3</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

จะเห็นว่า Servlet นี้ไม่ได้ทำการ override ฟังก์ชัน service แต่อย่างใดแต่กลับทำการ override ฟังก์ชัน doGet() แทน ฟังก์ชัน doGet() นี้เป็นฟังก์ชันที่ถูกกระบุอยู่ใน HttpServlet เพื่อรับ Request ที่มาจากไคลเอนต์ ในแบบ HTTP เมธอด GET ซึ่งก็คือ Request ที่มาจาก URL ธรรมดาทั่ว ๆ ไปนั่นเอง นอกจากฟังก์ชัน doGet() แล้วคลาส HttpServlet ยังมีฟังก์ชันอื่น ๆ ที่อนุญาตให้เราสามารถทำการ override เพื่อรองรับ request แบบอื่น ๆ ได้อีก ยกตัวอย่างเช่น

doPost() ใช้สำหรับรับ Request ที่มาจากไคลเอนต์ในแบบเมธอด POST

doDelete() ใช้สำหรับรับ Request ที่มาจากไคลเอนต์ ในแบบเมธอด DELETE เพื่ออนุญาตให้ไคลเอนต์ทำการลบ

Resource บางอย่างออกจากเซิร์ฟเวอร์ได้

ยังมีฟังก์ชันอีกสามอันคือ doPut(), doOptions() และ doTrace() ที่อยู่ในคลาส HttpServlet เพื่อรองรับ

Request แบบ PUT, OPTIONS และ TRACE แต่ไม่นิยมทำการ Override ฟังก์ชันเหล่านี้ด้วยเหตุที่ว่าส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดำเนินการ ของฟังก์ชันเหล่านี้ที่อยู่ในคลาส HttpServlet ได้ถูกจัดการอย่างดีเหมาะสำหรับการใช้งานทั่ว ๆ ไปแล้ว

2.2.3 การรัน Servlet

Servlet เป็นคอนเซ็ปต์หนึ่งใน Java ที่เราเขียนขึ้นเพื่อใช้งานสำหรับ Server Side Application ถ้าใครเคยเขียนแอปพลิเคชัน โดยใช้ Java มาก่อนจะเห็นว่าตัว Enter Point ของแอปพลิเคชันจะเป็น static ฟังก์ชันที่ชื่อ main() ยกตัวอย่างเช่น

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

ตัว Servlet เองจะมีลักษณะคล้าย ๆ Applet ตรงที่ว่าในการรัน ฟังก์ชันที่ใช้เป็น Enter Point จะไม่ใช่ฟังก์ชัน main() แต่จะกลายเป็นฟังก์ชันอื่นที่ถูกระบุไว้ใช้สำหรับตัวมันเองโดยเฉพาะยกตัวอย่างเช่นใน Applet ฟังก์ชันที่ถูกใช้เพื่อเป็น Enter Point ก็จะเป็นฟังก์ชันพวก init(), start() ซึ่งสำหรับ Servlet แล้วฟังก์ชันที่ถูกใช้เพื่อเป็น Enter Point ก็คือ service() ซึ่งก็คือฟังก์ชันที่ถูก define อยู่ใน javax.servlet.Servlet interface นั่นเอง ในการรัน Servlet สิ่งที่ต้องการมีดังต่อไปนี้คือ

1. JDK (Java Development Kit) สำหรับคนที่ใช้ Windows, Sun หรือ Linux ให้ไปดาวโหลดที่ <http://java.sun.com>

สำหรับคนที่รัก Linux โดยเฉพาะอาจจะไปดาวโหลดที่ <http://www.blackdown.org>

2. Servlet API Package and Servlet Engine Package นี้เป็นตัวที่เก็บ Servlet API ทั้งหมดไว้ซึ่งก็คือ javax.servlet.* และ javax.servlet.http.* package

ในการรัน Servlet เราสามารถดาวโหลดตัวที่เป็น Java Servlet Development Kit จาก Sun ซึ่งเป็นตัวที่เก็บ Servlet API และตัว Servlet Engine พื้นฐานเอาไว้โดยเราสามารถดาวโหลดได้จาก <http://java.sun.com/products/servlet/download.html> จากที่อย่างทีกล่าวมาข้างต้นว่า Servlet Engine อาจจะเป็นส่วนหนึ่งของเว็บเซิร์ฟเวอร์ หรืออาจจะเป็นตัว add-on ก็ได้ ดังนั้นถ้ามีเว็บเซิร์ฟเวอร์เปล่า ๆ ที่ยังไม่มี Servlet Engine ติดอยู่ก็อาจจะเลือก Servlet Engine ที่เป็นลักษณะ add-on โดยคิดเข้าไปเป็น Extension Module สำหรับเว็บเซิร์ฟเวอร์ก็ได้ นอกจากนี้ก็อาจจะเลือกดาวโหลดเว็บเซิร์ฟเวอร์ ที่มี Servlet Engine อยู่ในตัวแล้วก็ได้ สำหรับลิสของ Servlet Engine ที่มีอยู่สามารถหาจากที่นี่ได้ <http://www.javaskyline.com/serv.html>

บทที่ 3 ฐานข้อมูล (Data Base)

3.1 คำจำกัดความของฐานข้อมูล

เมื่อพูดถึงระบบฐานข้อมูล ทุก ๆ คนจะนึกถึงระบบที่ใช้จัดเก็บข้อมูลอย่างมีระบบระเบียบ สามารถนำมาใช้ประโยชน์ได้ทันทีด้วยวิธีการที่กำหนดไว้ เช่น ในโรงพยาบาลก็จะมีระบบฐานข้อมูลคนไข้จัดเก็บไว้ในตู้เอกสารรายละเอียดเกี่ยวกับคนไข้จัดเก็บไว้ในตู้เอกสารรายละเอียดเกี่ยวกับคนไข้แต่ละคนจะถูกจัดเก็บไว้ในนั้น และเมื่อคนไข้ผู้นั้นเจ็บป่วยมาหาหมอ คุณหมอก็จะบอกนางพยาบาลให้ไปหยิบแฟ้มของคนไข้คนนั้นมาศึกษาเพื่อจะได้ทราบประวัติสุขภาพของคนไข้ และช่วยในการตัดสินใจเกี่ยวกับอาการของคนไข้ขณะนั้น

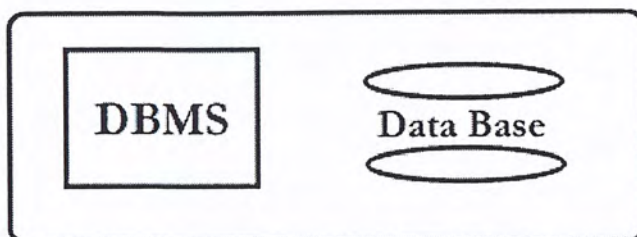
จากตัวอย่างที่ยกมาข้างต้น ประวัติคนไข้ เป็นระบบฐานข้อมูลอย่างหนึ่งทั้งสิ้น ดังนั้นเราอาจสรุปความหมายง่าย ๆ ของระบบฐานข้อมูลหรือดาต้าเบส ได้ว่าเป็นการรวบรวมข้อมูลที่มีประโยชน์อย่างมีระบบระเบียบในที่จัดเก็บเพื่อนำมาใช้ต่อไปในอนาคต ข้อมูลอะไรที่เก็บแล้วนำมาใช้ประโยชน์ไม่ได้ไม่ถือเป็นฐานข้อมูล

เมื่อก้าวถึงฐานข้อมูลแล้ว อีกสิ่งหนึ่งที่จะลืมไม่ได้และจะต้องพูดถึงควบคู่กันไปก็คือวิธีการจัดการดูแล และการเรียกใช้ จากตัวอย่าง เราจัดเก็บประวัติคนไข้ ด้วยการจดบันทึกไว้ในแฟ้มหรือแผ่นกระดาษแล้วนำไปเก็บไว้ในตู้เอกสาร จากนั้นก็คอยระวังไม่ให้ปลวกแทะ หากทิ้งไว้นานก็ต้องคอยปิดฝุ่น และเมื่อถึงเวลาจำเป็นต้องเรียกใช้ก็เดินไปค้นตู้รายการที่มีอักษรเรียงตามรายชื่อคนไข้เพื่อให้หยิบง่าย เหล่านี้เป็นตัวอย่างของวิธีการจัดเก็บ ดูแล และเรียกใช้ฐานข้อมูลอย่างง่ายตามรูปแบบของงานนั้น ๆ

เมื่อวิทยาการต่าง ๆ ก้าวหน้ามากยิ่งขึ้น การพัฒนารูปแบบของการจัดเก็บข้อมูลจึงเปลี่ยนไปเพื่อเพิ่มความสะดวกสบายให้กับผู้จัดเก็บและผู้ใช้ และเพื่อให้เพียงพอต่อปริมาณข้อมูลที่เพิ่มมากขึ้นทุกวัน จึงได้มีการนำระบบคอมพิวเตอร์มาใช้ในการจัดเก็บข้อมูล จากที่ใช้ตู้เอกสารและแฟ้มกระดาษก็เปลี่ยนไปเก็บในจานแม่เหล็ก (Magnetic Disk) โดยจานแม่เหล็กชุดหนึ่งสามารถเก็บข้อมูลได้เป็นพันล้านตัวอักษร เทียบได้กับประวัติคนไข้เป็นแสนเป็นล้านคน จากที่คุณหมอบอกนางพยาบาลให้ไปค้นประวัติจากตู้เอกสารจำนวนมากมาเป็นการป้อนคำสั่งผ่านแป้นพิมพ์ ข้อมูลต่าง ๆ ของคนไข้ก็ปรากฏบนจอแสดงภาพในพริบตา

สิ่งที่ใช้ในการรับคำสั่ง และค้นหาผลลัพธ์มาแสดงให้กับผู้ใช้ เราเรียกว่า โปรแกรมระบบงานคอมพิวเตอร์ (แอปพลิเคชัน โปรแกรม) และสิ่งที่ใช้ในการบันทึกข้อมูลลงไปในฐานข้อมูลเพื่อให้บริการข้อมูลแก่ผู้ใช้และดูแลข้อมูลไม่ให้สูญหาย เราเรียกว่า ระบบจัดการฐานข้อมูล (ดาต้าเบสแมนเนจเมนต์ซิสเต็ม หรือ DBMS) ซึ่งในที่นี้ก็คือ ออราเคิลนั่นเอง เราจะเห็นความสัมพันธ์ระหว่างฐานข้อมูล และ DBMS ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



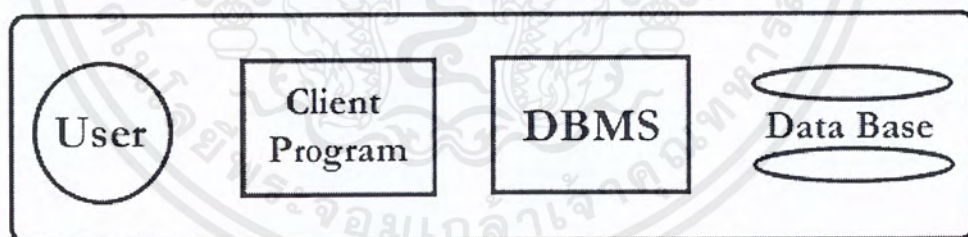
รูปที่ 3.1 แสดงฐานข้อมูลและโปรแกรมการจัดการ

3.2 ออราเคิล (Oracle 8I)

Oracle8i คือ ฐานข้อมูลเพื่ออินเทอร์เน็ต เกิดขึ้นมาเพื่อเปลี่ยนแปลงวิธีการจัดการและเข้าใช้ข้อมูลสารสนเทศ ในยุคอินเทอร์เน็ตเช่นทุกวันนี้ และได้เพิ่มความสามารถใหม่เพื่อการทำงานในแบบ OLTP และชุดแอปพลิเคชัน Data warehouse ความสามารถของ Oracle8i ทำให้องค์กรต่าง ๆ สามารถพัฒนาและใช้งานเพื่อประโยชน์ทางอินเทอร์เน็ตได้อย่างไม่จำกัด Oracle8i ได้นำ Java เข้ามาร่วมด้วย โดยผนวก Java VM ไว้กับตัว Oracle8i ด้วย นอกจากนี้ยังมี Oracle WebDb, iFS(Internet File System) และ interMedia ทั้งหมดนี้เป็นสิ่งที่ได้ผนวกเพิ่มเข้ามา

3.2.1 สถาปัตยกรรมของระบบจัดการฐานข้อมูล (Architecture of DBMS)

ออราเคิล8i ถูกออกแบบมาในลักษณะ ไคลเอนต์เซิร์ฟเวอร์สี่ชั้นนั้ล DBMS ประกอบด้วย 4 ส่วนหลัก (เป็นส่วนเชื่อมโยง 3 ส่วน และส่วนข้อมูล 1 ส่วน) ซึ่งแสดงอยู่ในรูป



รูปที่ 3.2 แสดงองค์ประกอบทั้ง 4 ส่วน

3.2.2 ส่วนเชื่อมโยงสามส่วน (The Three Interface)

ผู้ใช้คือใครก็ตามที่กำลังใช้โปรแกรมเพื่อทำงานเกี่ยวกับข้อมูลที่เราเรียกว่า โปรแกรมลูกข่าย (ไคลเอนต์โปรแกรม) ซึ่งกำลังติดต่อกับโปรแกรม DBMS ซึ่งอาจเรียกว่า เซิร์ฟเวอร์ ก็ได้ เซิร์ฟเวอร์จะทำหน้าที่ติดต่อกับตัวฐานข้อมูล เมื่อขยายจากรูปที่ เป็นรูปที่ จะพบว่าส่วนต่าง ๆ ทั้งสี่ อันได้แก่ ผู้ใช้ โปรแกรมลูกข่าย (ไคลเอนต์โปรแกรม) โปรแกรม DBMS และฐานข้อมูลเชื่อมโยงเข้าด้วยกันโดยตัวเชื่อมสามส่วนตามรูปที่ 2.2 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวเชื่อมที่ 1 ยูสเซอร์อินเทอร์เฟซ (User Interface) ก็คือส่วนที่แสดงบนจอแสดงผลเมื่อผู้ใช้ส่งคำสั่งเข้าไปผ่านแป้นพิมพ์ ส่วนเชื่อมที่ 1 นี้จะใช้ง่ายและให้ผลที่เข้าใจง่ายเพียงใดก็ขึ้นอยู่กับการออกแบบวิธีการรับคำสั่ง และแสดงผลของ ไคลเอ็นต์โปรแกรม ที่กำลังใช้อยู่

- ตัวเชื่อมที่ 2 ไคลเอ็นต์-เซิร์ฟเวอร์ อินเทอร์เฟซ (Client - Server Interface) เป็นตัวเชื่อมระหว่างโปรแกรมลูกข่าย (ไคลเอ็นต์โปรแกรม) กับโปรแกรมแม่ข่าย (เซิร์ฟเวอร์โปรแกรม) สำหรับออรากิล8i ตัวเชื่อมนี้ก็คือภาษา SQL

ในการทำงานจริง โปรแกรมลูกข่ายและโปรแกรมแม่ข่ายจะอยู่บนเครื่องคอมพิวเตอร์คนละเรื่อง ติดต่อกันด้วยระบบเครือข่ายแบบนี้คือ โปรแกรม SQL*Plus

- ตัวเชื่อมที่ 3 อินพุต เอาต์พุตอินเทอร์เฟซ (I/O Interface) เป็นตัวเชื่อมระหว่างโปรแกรมแม่ข่ายกับข้อมูล คือการอ่านหรือเขียนข้อมูลบนจานแม่เหล็กหรือดิสก์ ส่วนนี้เราสามารถปรับแต่งให้การอ่านเขียนข้อมูลเร็วขึ้นได้

ในการทำงานจริงแม้ว่าจะสามารถใช้งานดิสก์ที่อยู่บนเครื่องแม่ข่ายอื่นได้ แต่ขอแนะนำให้ติดตั้งออรากิล8i (โปรแกรมแม่ข่าย) และเพิ่มฐานข้อมูลอยู่บนเครื่องคอมพิวเตอร์เดียวกัน

3.2.3 เทเบิลในระบบฐานข้อมูล

คำว่าเทเบิล คือที่จัดเก็บข้อมูลที่สัมพันธ์กันไว้ด้วยกัน โดยแบ่งข้อมูลที่สัมพันธ์กันนั้นออกเป็น ส่วน ๆ ตามแนวตั้งที่เรียกว่า คอลัมน์ ดังนั้นเพื่อให้เกิดความรู้สึกที่ตรงกันในด้านความหมาย ผู้แปลจะทับศัพท์บางคำไว้ เช่น เทเบิล และคอลัมน์ เป็นต้น

เมื่อเราทำการติดตั้งโปรแกรม ออรากิล8i เสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปคือสร้างที่จัดเก็บข้อมูลซึ่งก็คือสิ่งที่เราเรียกว่า เทเบิล (ส่วนการสร้างฐานข้อมูลเพื่อรองรับการสร้างเทเบิลได้กระทำเรียบร้อยแล้วในขั้นตอนการติดตั้งออรากิล) การสร้างเทเบิลเราใช้คำสั่ง SQL*Plus ดังนี้

CREATE Table Table Name (Column Name Data Type);

คำอธิบายส่วนต่างมีดังนี้

- ชื่อเทเบิล (Table Name)

ชื่อเทเบิลในดาต้าเบสหนึ่ง ๆ จะต้องไม่ซ้ำกัน แต่ต้องไม่มีช่องว่างในชื่อ และต้องไม่ขึ้นต้นด้วยตัวเลข

- ชื่อคอลัมน์ (Column Name)

ชื่อคอลัมน์ในเทเบิลหนึ่ง ๆ จะต้องไม่ซ้ำกันและสามารถตั้งชื่อได้ยาวถึง 30 ตัวอักษร จะเป็นชื่อภาษาไทยหรืออังกฤษก็ได้ แต่ต้องไม่มีช่องว่างในชื่อ และต้องไม่ขึ้นต้นด้วยตัวเลข

- ชนิดของข้อมูล (Data Type)

ชนิดของข้อมูลจะต้องเป็นไปตามชนิดดังต่อไปนี้

1. CHAR ข้อมูลชนิด CHAR เป็นข้อมูลแบบตัวอักษรหรือตัวเลขที่ไม่ใช่เพื่อการคำนวณ เมื่อเราระบุขนาดความกว้างของคอลัมน์แล้ว ขนาดของที่บรรจุหรือที่จัดเก็บจะคงที่แม้ว่าเราจะใส่ข้อมูลไม่เต็มช่องก็ตาม

สำหรับช่องที่เหลือนั้น ออราเคิล8i จะเติมช่องว่างให้จนเต็ม แต่หากเราใส่ข้อมูลเกินขนาดของคอลัมน์ ออราเคิล8i ก็จะตัดส่วนที่เกินทิ้งให้เช่นกัน

การเปรียบเทียบค่าของคอลัมน์ชนิด CHAR หากคอลัมน์ใดสั้นกว่า ออราเคิล8i จะเติมช่องว่างให้ยาวเท่ากับคอลัมน์ที่ยาวกว่า จากนั้นจึงเปรียบเทียบค่าของทั้งสองคอลัมน์

2. DATE ชนิดของข้อมูลแบบวันที่ DATE นี้จะจัดเก็บข้อมูลทั้งวันและเวลา คือ ปี เดือน วัน ชั่วโมง นาที วินาที และเศษของวินาที รูปแบบของ DATE จะจัดเก็บข้อมูลโดยบีบอัดไว้ในที่เก็บขนาด 7 ไบต์ รูปแบบมาตรฐานที่ใช้อ้างอิงถึงข้อมูลแบบ DATE คือ 31-DEC-2001 หรือ “DD-MM-YYYY” , 31/12/2001 15:30 หรือ “dd/mm/yyyy hh:24.min” ข้อมูลแบบวันที่ที่เราสามารถจัดเก็บได้จะต้องอยู่ในช่วงเวลาระหว่างวันที่ 1 มกราคม 4712 ก่อนคริสตกาล ถึงวันที่ 31 ธันวาคม คริสตศักราช 4712 หรือ Jan 1,4712 B.C. – Dec 31,4712 AD

3. VARCHAR2 ชนิดข้อมูลแบบ VARCHAR2 จะคล้ายกับ CHAR แต่จะจัดเก็บตามขนาดที่เป็นจริง เช่นถ้าเรากำหนดชนิดของคอลัมน์เป็น VARCHAR 2 ขนาด 20 ตัวอักษร เราสามารถเก็บข้อมูลที่ตัวอักษรก็ได้ แต่จะต้องไม่เกิน 20 ตัว จำนวนไบต์ที่ใช้จัดเก็บก็จะเท่ากับจำนวนตัวอักษร ไม่มีการเติมช่องว่างให้เปลืองที่จัดเก็บ ซึ่งทำให้ประหยัดกว่าแบบ CHAR มาก เราสามารถกำหนดขนาดของ VARCHAR2 ได้สูงสุด 2000 ตัวอักษร

4. RAW ชนิดข้อมูลแบบ RAW จะเหมือนกับ VARCHAR2 แต่สามารถเก็บข้อมูลนอกเหนือจากตัวเลขและตัวอักษรได้ สามารถเก็บได้สูงสุด 255 ไบต์

5. LONG ข้อมูลชนิด LONG จะคล้ายกับ VARCHAR2 แต่จะขยายขนาดของข้อมูลได้ถึง 2 พันล้านตัวอักษร (กิกะไบต์)

6. LONG RAW มีลักษณะเหมือนกับ LONG รวมทั้งข้อจำกัดต่าง ๆ ด้วย เรามักจัดเก็บข้อมูลประเภทภาพ เสียง ภาพเคลื่อนไหว เพลงต่าง ๆ ไว้ในคอลัมน์ชนิดนี้

7. NUMBER ชนิดข้อมูล NUMBER ใช้เก็บตัวเลขที่ใช้ในการคำนวณ มีเพียงแบบเดียวใน ออราเคิล8i แต่สามารถเก็บตัวเลขได้ทุกชนิด ขนาดของตัวเลขที่สามารถเก็บได้คือ 38 ตำแหน่ง หรืออยู่ในช่วง $-9.99..X 10^{125}$ + $9.99..X 10^{125}$

เมื่อเรากำหนดให้คอลัมน์ใดคอลัมน์หนึ่งใช้ข้อมูลแบบ NUMBER เราสามารถระบุได้ทั้งจำนวนตำแหน่งทั้งหมดที่จัดเก็บ และจำนวนตำแหน่งสูงสุดหลังจุดทศนิยม

8. ROWID ข้อมูลชนิด ROWID เป็นแบบที่ใช้ภายใน ออราเคิลเอง โดยประกอบขึ้นจากข้อมูลสามส่วน คือ Block, Row และ File เพื่อใช้ชี้ตำแหน่งที่จัดเก็บข้อมูลที่แท้จริงของแต่ละรายการ

- ขนาดของข้อมูล

ขนาดข้อมูลที่กำหนดเพื่อใช้บรรจุข้อมูล และจะต้องไม่เกินค่าสูงสุดของข้อมูลแต่ละชนิด และต้องมากกว่าศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data Scale

หากคอลัมน์นี้ใช้เป็นแบบตัวเลข เราสามารถกำหนดจำนวนตำแหน่งหลังจุดทศนิยมได้จากฟิลด์นี้ ค่าในฟิลด์นี้จะต้องน้อยกว่าค่าที่กำหนดไว้ในฟิลด์ Data Size

- Default

ถ้าเพิ่มรายการเข้าไปในเทเบิลและไม่มีค่าให้กับบางคอลัมน์ และถ้าคอลัมน์นั้นมีการกำหนดค่าปริยาย (Default) ไว้ ออราเคิลจะนำค่านั้นใส่เข้าไปในคอลัมน์ แต่หากไม่มีการกำหนดจะให้ค่า NULL แทน

- Primary Key

ถ้าคอลัมน์ใดถูกกำหนดให้เป็น Primary Key ค่าที่เก็บในคอลัมน์นั้นจะต้องไม่ซ้ำกับรายการอื่นในคอลัมน์เดียวกัน และจะเป็นค่า NULL ไม่ได้

- Foreign Key

Foreign Key เป็นการกำหนดให้ค่าในคอลัมน์นั้นจะต้องตรงกับค่าใดเพียงค่าหนึ่งเท่านั้นใน Primary Key ของเทเบิลที่กำหนด

3.3 ภาษา SQL

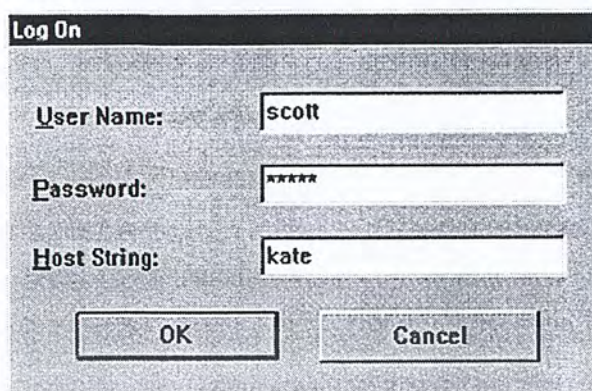
คำสั่งในภาษา SQL เป็นภาษาที่ใช้ติดต่อกับออราเคิล และระบบจัดการฐานข้อมูลแบบสัมพันธ์มาตรฐานทั่ว ๆ ไป ระบบฐานข้อมูลจะประกอบด้วยเทเบิลที่มีหน้าที่ในการจัดเก็บข้อมูล แต่ละเทเบิลก็จะเก็บข้อมูลประเภทเดียวกัน เป็นรายการหลาย ๆ รายการประกอบกันเป็นกลุ่มหรือเซต กลุ่มของรายการอาจมีเพียงหนึ่งรายการหรือเป็นร้อยเป็นพันรายการก็ได้ ในทางทฤษฎีกล่าวว่าระบบฐานข้อมูลจะจัดการกับข้อมูลเป็นกลุ่มหรือเซต เช่นการเพิ่มข้อมูล การปรับปรุง การลบหรือ การสืบค้นก็อาจจะทำกับข้อมูลทีละรายการหรือหลาย ๆ รายการก็ได้ โดยเราสามารถกล่าวได้ว่ามีการกระทำกับกลุ่มของข้อมูล

ข้อดีของการทำงานกับข้อมูลที่เป็นกลุ่มคือ เราสามารถส่งคำสั่งเพียงคำสั่งเดียวเพื่อไปจัดการกับข้อมูลจำนวนมากให้ทำงานซ้ำ ๆ กันได้ โดยไม่ต้องออกคำสั่งใหม่ทุกครั้งที่เราอ่านข้อมูลรายการใหม่

3.3.1 การสืบค้น

เมื่อไรก็ตามที่เราต้องการนำข้อมูลที่จัดเก็บในฐานข้อมูลมาใช้งาน เช่น ให้แสดงรายงานยอดขายในปีที่แล้ว แยกตามไตรมาส ความต้องการนี้ส่วนแต่ต้องใช้คำสั่ง SQL เข้าไปค้นหาจากในฐานข้อมูลทั้งสิ้น และคำสั่งในการค้นหาเหล่านี้เราอาจเรียกรวม ๆ ว่าเป็นคำสั่งสืบค้น

คำสั่งในการสืบค้นมักขึ้นต้นด้วยคำว่า Select และตามด้วยวลี ซึ่งจะได้อธิบายต่อไป โปรแกรมที่ใช้คำสั่งสืบค้นได้มีอยู่มากมาย แต่ในที่นี้จะใช้วลีที่ชื่อ SQL*Plus ให้ติดต่อกับฐานข้อมูลด้วยผู้ใช้ชื่อ Scott หรือชื่ออื่น ๆ ได้ตามต้องการ แล้วใส่รหัสผ่าน หากเป็น Scott ก็ให้ใช้ Tiger



รูปที่ 3.3 แสดงหน้าจอล็อกอินของ SQL Plus

3.3.2 ดูข้อมูลด้วย SQL

ถ้าเราต้องการจะทราบข้อมูลต่าง ๆ ในเทเบิล จะใช้คำสั่ง `SELECT * FROM TABLE NAME;` การใช้เครื่องหมาย (*) หมายความว่าให้แสดงข้อมูลทุกคอลัมน์ แต่ถ้าเราเปลี่ยนจากเครื่องหมายดอกจันเป็นชื่อคอลัมน์แทน เราก็จะได้เฉพาะข้อมูลของคอลัมน์นั้นเท่านั้น แต่ถ้าเราสนใจข้อมูลหลายคอลัมน์ ก็ให้พิมพ์ชื่อคอลัมน์เหล่านั้นตามคำสั่ง `select` โดยใช้เครื่องหมายจุดภาค (.) คั่นระหว่างคอลัมน์นั้น

โดยมีข้อแม้ว่าชื่อที่นำมาใช้นั้นจะต้องมีปรากฏอยู่ในเทเบิลที่เราเลือกด้วย ชื่อเทเบิลที่เราเลือกนี้จะต้องระบุไว้หลังคำว่า `FROM` เสมอ คำสั่งสืบลิ้นเริ่มต้นด้วยคำสั่ง `SELECT` และมีคำว่า `FROM` เป็นคำประกอบระบุแหล่งที่มาของข้อมูล ปกติถ้าเราออกคำสั่งสืบลิ้นโดยระบุคำประกอบเพียง `SELECT` และ `FROM` เราจะดูได้ทุกรายการ แต่ถ้าหากเราต้องการระบุขอบเขตหรือเงื่อนไขในการค้นหาข้อมูลบางรายการ ก็สามารถทำได้โดยการระบุคำประกอบ `WHERE` แล้วตามด้วยเงื่อนไข

จะสังเกตเห็นว่าคำสั่ง ชื่อเทเบิล หรือชื่อคอลัมน์ก็ตาม เราสามารถพิมพ์ตัวเล็กตัวใหญ่ก็ได้ โดยจะให้ผลลัพธ์ที่เหมือนกัน แต่หากเป็นการเปรียบเทียบแล้ว ค่าที่นำมาเปรียบเทียบจะต้องตรงกับค่าที่จัดเก็บอยู่ในคอลัมน์เท่านั้น ซึ่งหมายความว่าตัวพิมพ์เล็กจะให้ผลที่ต่างกับตัวพิมพ์ใหญ่ และถ้าคอลัมน์นั้นเป็นข้อมูลแบบตัวอักษรก็ต้องล้อมด้วยเครื่องหมายฟันหนู (" ") หรือฝนทอง (‘ ’) เสมอ

- `From` ส่วนถัดมาในคำสั่งสืบลิ้นคือ `From` เป็นส่วนที่ใช้แสดงรายชื่อเทเบิลที่ต้องการค้นหาข้อมูล โดยเทเบิลที่นำมาแสดงนี้จะมีจำนวนเท่าไรก็ได้ภายใต้เงื่อนไขที่สมเหตุสมผล เพราะถ้าเราไม่กำหนดเงื่อนไขที่ดีแล้ว ผลที่ได้จะไม่สื่อความหมายอันใด ส่วนมากแล้วที่จะมีการประกาศชื่อเทเบิลมากกว่าหนึ่งก็เพราะต้องการผลของการ `Join` ของเทเบิลเหล่านั้น การ `Join` คือการนำเงื่อนไขมาใช้ค้นหาข้อมูลตั้งแต่สองเทเบิลขึ้นไปเพื่อให้ได้ผลลัพธ์ที่ตรงตามเงื่อนไขของทั้งสองเทเบิล

- WHERE ใช้ในการกำหนดเงื่อนไขในการเรียกดูข้อมูลเฉพาะบางรายการ โดยที่ออราเคิลจะทำการตรวจสอบแต่ละรายการว่าตรงตามเงื่อนไขได้มากกว่าหนึ่งโดยใช้ And หรือ Or เป็นตัวเชื่อม หากมีเงื่อนไขมากอาจใช้วงเล็บช่วยบอกลำดับในการเปรียบเทียบได้

- ORDER BY ในภาษา SQL จะใช้คำประกอบ ORDER BY ในการเรียงลำดับผลที่ได้จากคำสั่งสืบค้นตามชื่อคอลัมน์ที่ระบุอยู่หลังคำ Order by ชื่อคอลัมน์เหล่านั้นจะต้องถูกเลือกขึ้นมาแสดงด้วย

โดยปกติการเรียงลำดับจะเรียงจากน้อยไปหามาก และเรียงคอลัมน์ที่ระบุก่อนหลังตามลำดับ แต่หากต้องการเรียงจากมากไปหาน้อย ต้องระบุ DESC คำ DESC เป็นการระบุให้เรียงจากมากไปหาน้อยตามคอลัมน์ที่ระบุอยู่หน้าคำ DESC นั้น บางกรณีที่ต้องการเรียงลำดับตามนิพจน์ที่ระบุหลัง Select จะต้องระบุเป็นหมายเลขประจำตำแหน่งในลำดับ อนุประโยค ORDER BY จะต้องระบุเป็นส่วนสุดท้ายในประโยคคำสั่งสืบค้นเสมอ

- GROUP BY ในคำสั่งสืบค้นที่เห็นทั่ว ๆ ที่เราเรียกใช้ฟังก์ชัน มักจะเป็นการใช้ฟังก์ชันที่ทำงานกับข้อมูลที่ละรายการ แต่ก็มีฟังก์ชันที่สามารถประมวลผลกับข้อมูลเป็นชุด ที่เรียกว่า กรุปฟังก์ชัน GROUP BY จะแบ่งแยกข้อมูลเป็นกลุ่ม แล้วทำตามฟังก์ชันตามกลุ่มเหล่านั้น จึงได้ผลลัพธ์ตามต้องการ

- HAVING ในการกำหนดเงื่อนไขในคำสั่งสืบค้น เราจะระบุในอนุประโยค WHERE เป็นเงื่อนไขง่าย ๆ ไม่มีการระบุฟังก์ชันกลุ่มในเงื่อนไขนั้น แต่ถ้าหากเราต้องการให้มีการเปรียบเทียบโดยใช้ฟังก์ชันกลุ่ม เราจะใช้ใน Where ไม่ได้ จะต้องเปลี่ยนไปใช้ใน HAVING แทน

- INSERT คำสั่ง INSERT เป็นคำสั่งที่ใช้เพิ่มเติมข้อมูลเข้าไปในฐานข้อมูลโดยจัดเก็บไว้ในเทเบิล เราอาจเพิ่มทีละรายการหรือทีละหลาย ๆ รายการก็ได้ การเพิ่มทีละรายการจะใช้คำสั่ง

```
SQL> INSERT INTO TABLE NAME VALUES();
```

เราจะต้องระบุคำสั่งโดยเริ่มด้วยคำว่า INSERT INTO แล้วตามด้วยชื่อเทเบิล ตามด้วยคำว่า VALUES และค่าที่เราจะเพิ่มเติมตามจำนวนคอลัมน์ ตามลำดับ

การระบุค่าต้องระบุค่าให้ตรงกับชนิดของข้อมูลของคอลัมน์ เราสามารถระบุการเพิ่มข้อมูลบางคอลัมน์ได้โดยต้องพึงระวังไว้ด้วยว่า คอลัมน์ที่เราไม่ระบุจะต้องยอมรับค่า Null เพราะออราเคิล จะให้ค่า Null กับคอลัมน์ที่ไม่ระบุ ส่วนการเพิ่มทีละหลายรายการ เราสามารถเพิ่มข้อมูลเข้าไปในเทเบิลทีละหลายรายการได้โดยใช้คำสั่ง Select มาช่วย

- UPDATE การออกคำสั่ง Update จะเป็นการแก้ไขข้อมูลรายการใดรายการหนึ่งหรือหลายรายการในเทเบิลได้โดยใช้คำสั่ง SQL> UPDATE TABLE NAME SET" COLUMN NAME = ค่าที่ต้องการ;

- DELETE เป็นการลบรายการออกจากเทเบิลโดยการระบุคำสั่ง SQL>DELETE FROM TABLE ;

- TRANSACTION ในการออกคำสั่ง Insert, Update, Delete ข้อมูลจากเทเบิล เราจะออกคำสั่งลดย ๆ ไม่ได้ ต้องมีการระบุยืนยันคำสั่งหรือยกเลิกคำสั่งเหล่านี้ปิดท้ายคือ

COMMIT ยืนยันคำสั่ง

ROLLBACK ยกเลิกคำสั่ง

3.3.3 การติดต่อกับ Front End Tools

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกครั้งที่เราจะติดต่อกับออราเคิล ไม่ว่าจะด้วยโปรแกรมใด ๆ ก็ตาม จะต้องบอกให้ออราเคิลทราบว่าเราจะติดต่อในนามของใคร นั่นคือต้องบอกชื่อผู้ใช้พร้อมทั้งรหัสผ่าน ก่อนเสมอ เพื่อเหตุผลทางด้านการรักษาความปลอดภัยของระบบฐานข้อมูล โดยออราเคิลจะตรวจสอบดูว่า บุคคลผู้นั้นมีสิทธิเข้ามาใช้งานฐานข้อมูลหรือไม่ และถ้าเข้ามาได้จะเข้ามาทำอะไรได้บ้าง ในที่นี้จะขอใช้ชื่อผู้ใช้ Scott ที่มีรหัสผ่านว่า Tiger เป็นตัวอย่าง

3.3.3.1 วิธีการติดต่อกับฐานข้อมูล

ทุกครั้งที่เราจะเริ่มต้นขอใช้งานกับออราเคิล เราจะพบหน้าจอถามถึงสิ่งสามสิ่ง คือ ชื่อที่จะติดต่อ รหัสผ่านของชื่อนั้น และเครื่องแม่ข่ายที่จะติดต่อ หรือที่ศัพท์ภาษาอังกฤษว่า โฮสสตริง โฮสสตริง นี้บางกรณีเราอาจจะระบุถึงโปรโตคอลที่ใช้ด้วย รูปแบบของการส่ง ยูสเซอร์เนม, พาสเวิร์ด และโฮสสตริง ไปให้กับออราเคิล ก็คือ

Client-program Username, Password และ Host String ไปให้กับออราเคิล ถ้าเรายกตัวอย่างให้

Client-program = Sqlplus

ชื่อผู้ใช้ = Scott

รหัสผ่าน = Tiger

Host String = Tom

เราจะได้คำสั่งดังนี้ `Sqlplus Scott/tiger@tom`

บางกรณีเราอาจซ่อน โฮสสตริง ไว้ที่อื่นเพื่อให้เกิดความสะดวกในการพิมพ์คำสั่ง คือ ไม่ต้องพิมพ์ยาว และไม่ต้องเสียเวลาจำ โดยเพียงแต่กำหนดค่าพารามิเตอร์ในไฟล์ออราเคิล หรือกำหนดผ่านโปรแกรมลงทะเบียน พารามิเตอร์นั้นคือ

LOCAL = Tom

ต่อไปเมื่อเราจะติดต่อกับออราเคิล เราก็บอกเพียงแค่ชื่อและรหัสผ่านดังนี้

Sqlplus Scott/tiger

3.3.4 Tools ที่ได้มาพร้อม ออราเคิล

ออราเคิลจะมีโปรแกรมดี ๆ ให้เราใช้เป็นจำนวนมาก เช่น SQL*Plus, Database Manager, Session Manager และอื่น ๆ อีกมากมาย หากดูได้จากรูปไอคอน ในกลุ่มของออราเคิล หลังจากติดตั้งเรียบร้อยแล้ว

SQL*Plus

เป็นโปรแกรมยอดนิยมของเหล่าโปรแกรมเมอร์ที่ต้องใช้งานหรือพัฒนาระบบงานบนฐานข้อมูลออราเคิลมาก เพราะเป็นโปรแกรมที่ถูกออกแบบมาให้ผู้ใช้พิมพ์คำสั่ง SQL เพื่อทำงานกับออราเคิล ในลักษณะคอมมานดไลน์ แต่ผู้ใช้บางกลุ่มก็บอกว่าไม่ชอบเพราะต้องจำคำสั่งเยอะ ใครจะจำได้หมด ก็ไม่เป็นไร ถ้าจำไม่หมด เปิดคู่มือก็ได้อีก เราสามารถใช้ SQL*Plus ทำสิ่งเหล่านี้ให้เราได้ คือ

- สร้าง แก้ไข ลบ เทเบิล

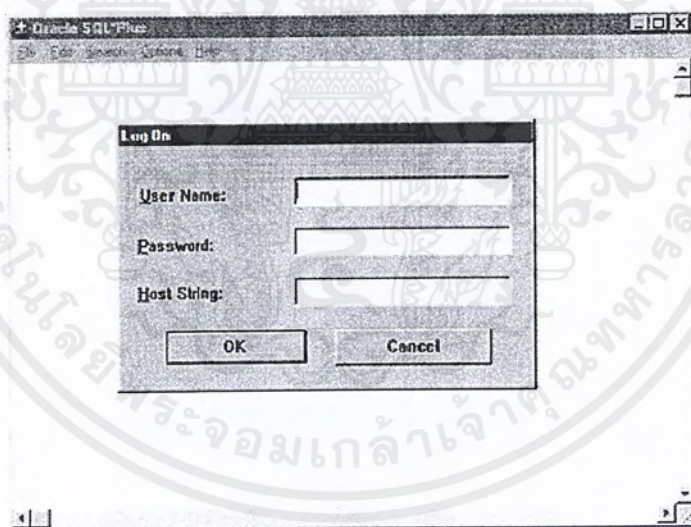
- เพิ่ม ลบ ปรับปรุงข้อมูลในเทเบิลหรือ วิว
- สร้าง ลบ แก้ไขชื่อผู้ใช้
- เพิ่มเนื้อที่ให้ฐานข้อมูล หรือแก้ไขพารามิเตอร์บางอย่างของการจองเนื้อที่ว่าง
- รันสคริปต์ที่มีคำสั่ง SQL หลาย ๆ คำสั่งเป็นชุด
- ทำคำสั่ง SQL ทั่ว ๆ ไป

เราจะทำสิ่งดังกล่าวได้ก็ต่อเมื่อเรามีสิทธิ์ในเรื่องนั้น ๆ เท่านั้น และงานบางงานที่กล่าวไปข้างต้นสามารถทำได้ด้วยโปรแกรมอื่นโดยไม่ต้องจำคำสั่งยาก ๆ เหล่านั้นเลย ในการใช้งาน SQL*Plus จะต้องหาไอคอนหน้าต่างต่อไปนี้ให้



รูปที่ 3.4 ภาพแสดงหน้าต่างไอคอน SQL Plus

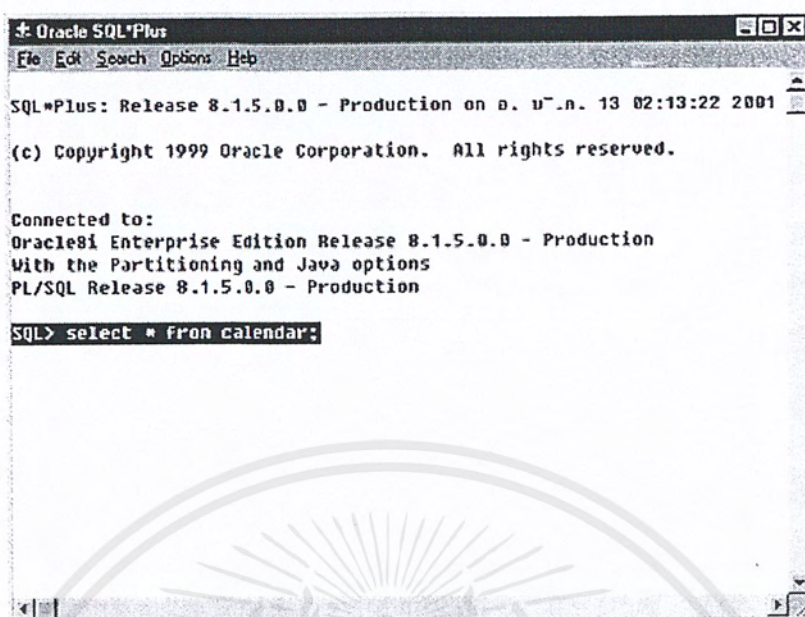
พบดังรูปที่ 3.4 โดยมากมักอยู่ในกลุ่มหรือโฟลเดอร์ของออรากเคิล จากนั้นก็เรียกใช้โดยคลิก 2 ครั้งที่ไอคอนนั้น เราจะได้หน้าจอข้งล่างรอรับผู้ใช้ดังได้กล่าวมาแล้ว



รูปที่ 3.5 ภาพแสดงโปรแกรม SQL*Plus

จากนั้นเราก็ใส่ข้อมูลเกี่ยวกับผู้ใช้ให้ครบถ้วน คือ `Scott/Tiger@Tom` แล้วกดปุ่ม OK เมื่อทุกอย่างถูกต้องเราก็จะได้หน้าต่างวินโดว์ดังในภาพที่ 3.6 ถึงตอนนี้เราก็สามารถติดต่อกับออรากเคิล ผ่าน SQL*Plus ได้แล้ว และถ้าต้องการยกเลิกการใช้งานก็เพียงแต่พิมพ์คำสั่ง `Exit` ที่ `SQL >` ออกจากโปรแกรมได้โดยง่ายหรือเรียกใช้เมนู File ตามด้วย Exit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

Oracle SQL*Plus
File Edit Search Options Help

SQL*Plus: Release 8.1.5.0.0 - Production on อ. น.ค. 13 02:13:22 2001
(c) Copyright 1999 Oracle Corporation. All rights reserved.

Connected to:
Oracle8i Enterprise Edition Release 8.1.5.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.5.0.0 - Production

SQL> select * from calendar;

```

รูปที่ 3.6 หน้าตาของ SQL*Plus หลังจาก Log on เสร็จเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 WAP (Wireless Application Protocol)

4.1 บทนำ และประวัติความเป็นมา

ทุกวันนี้ การสื่อสารไร้สายก้าวเข้ามาในชีวิต จนเป็นเรื่องธรรมดาไปแล้วซึ่งต่างจากไม่กี่ปีที่ผ่านมา ลิปลิป และนับจากนี้ไปใครจะรู้ว่า โลกการสื่อสารที่ว่า ไร้พรมแดนอย่างอินเทอร์เน็ต ก็กำลังถูกขย่งลงมาอยู่ในกำมือของคุณเช่นกัน เทคโนโลยีสำหรับมือถือ ถูกพัฒนาไปกว้างไกลยิ่งขึ้น ในระยะนี้แนวคิดที่กำลังได้รับความ สนใจมากที่สุดเรื่องหนึ่ง คือ การใช้อินเทอร์เน็ตผ่านมือถือ โดยใช้ WAP (Wireless Application Protocol) ซึ่งเป็นโพรโตคอลที่ถูกดัดแปลงมาจากโพรโตคอลของอินเทอร์เน็ต เพื่อให้เหมาะกับการทำงานของ โทรศัพท์เคลื่อนที่ มาตรฐานนี้ถือกำเนิดได้ประมาณ 2 ปีเท่านั้น จากกลุ่มผู้ผลิตรายใหญ่อย่าง อีริคสัน, โมโตโรล่า, โนเกีย และ Unwired Planet จนจัดตั้งเป็นสมาคม WAP Forum ขึ้น ก่อนหน้านี้นี้ต่างบริษัท ต่างก็มีโพรโตคอลของตนเองสำหรับพัฒนาแอปพลิเคชัน จึงจำกัดการพัฒนาตลาดใหม่ๆ ที่จะมาใช้กับการสื่อสารไร้สาย ดังนั้นโพรโตคอล WAP ได้กำหนดขึ้นมาเพื่อใช้ร่วมกัน เมื่อวันที่ 30 เมษายน 1998 ข้อกำหนดทางเทคนิคของ WAP(1.0) ได้ประกาศใช้ไป แม้ว่าจะมีบางส่วนที่ไม่สมบูรณ์ก็ตาม แนวโน้มผู้ใช้บริการโทรศัพท์ WAP PHONE คาดว่าจะเพิ่มขึ้นจาก 150 ล้านรายในปีนี้เป็น 800 ล้านรายภายในปี 2003 อย่างแน่นอน ทางผู้ผลิตอุปกรณ์สื่อสารไร้สายจึงหันมาสร้างสรรค์ผลิตภัณฑ์ใหม่ของตน เพื่อรองรับการใช้งานอินเทอร์เน็ตกันยกใหญ่ ประเดิมรายแรกด้วยโนเกีย ซึ่งผลิตภัณฑ์ที่ใช้ WAP มาเป็นครั้งแรกในประเทศไทย คือ โนเกีย 7110 ทางอีริคสัน และซีเมนส์ ก็เพิ่งเปิดตัวไปหมาดๆ กับผลิตภัณฑ์ใหม่ อีริคสัน R320s และซีเมนส์ ซีรีส์ 35 และแน่นอนว่า ต่างยังคงพัฒนารุ่นใหม่ต่อไปไม่หยุด และมีแนวโน้มชัดเจนว่าเครื่องรุ่นใหม่ๆ ทุกเครื่องต้องใช้บริการอินเทอร์เน็ตได้ด้วย

WAP (Wireless Application Protocol) เป็นโพรโตคอลตัวหนึ่ง ที่โทรศัพท์มือถือในยุคอนาคต จะรองรับหรือสนับสนุนโพรโตคอลนี้ เพื่อที่จะใช้งานโทรศัพท์มือถือเชื่อมต่อเข้ากับอินเทอร์เน็ตได้ สามารถรับส่งข้อมูล อีเมล เว็บไซ และ แอปพลิเคชัน อื่นๆ ขึ้นกับการประยุกต์ใช้ต่อไปในอนาคต WAP ขึ้นค้ำอยู่ระหว่าง 2 เทคโนโลยี นั่นคือ เทคโนโลยี ข้อมูลไร้สาย และ เทคโนโลยีอินเทอร์เน็ตซึ่งทั้งสองเทคโนโลยีนี้ ในยุคปัจจุบัน กำลังเจริญเติบโต อย่างรวดเร็วมาก จะเป็นแรงผลักดันให้เกิด บริการเทคโนโลยีข่าวสารใหม่ๆ เหมาะสมกับยุคมากขึ้น ถ้าใช้ 2 เทคโนโลยีนี้ ร่วมกันอย่าง เต็มประสิทธิภาพ

ข้อจำกัดของโทรศัพท์มือถือในส่วนของตัวเครื่องมีอะไรบ้างดังต่อไปนี้

- ชิพประมวลผล (Processor) ที่มีอยู่บนโทรศัพท์มือถือไม่มีกำลังความสามารถมากนักเมื่อเทียบเท่ากับเครื่องคอมพิวเตอร์ตั้งโต๊ะ หรือ เครื่องคอมพิวเตอร์ กระเป๋าหิ้ว
- หน่วยความจำ (Memory รวมถึง ROM และ RAM) มีน้อย
- มีข้อจำกัดเรื่องการบริโภคพลังงานจากแบตเตอรี่
- มีหน้าจอแสดงผล เล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีอุปกรณ์ที่ทำหน้าที่เป็น Key input หลากหลายแบบ ที่แตกต่างกัน
ข้อจำกัดของโทรศัพท์มือถือในแง่ของเครือข่ายข้อมูล มีดังนี้
- มีช่องทางเดินข้อมูลที่ไม่มากนัก (Less Bandwidth)
- มีความล่าช้าของการส่งข้อมูล
- มีความเสถียรในการเชื่อมต่อไม่สูงนัก ในขณะที่มีการ เคลื่อนที่เร็วๆ หรือ มีการใช้ โครข่ายข้อมูล อย่าง

หนาแน่น

- ไม่สามารถ คาดการณ์เหตุการณ์ว่าจะเกิดอะไรขึ้นกับ เครือข่ายได้ชัดเจนนัก

จะเห็นว่า ก็มีข้อจำกัด (Constrain) อยู่หลายข้อที่ เกี่ยวกับ เครือข่ายข้อมูล โทรศัพท์มือถือ ซึ่งผู้ที่ทำหน้าที่ ให้บริการนั้น จะต้องคิดหรือ ทำอะไรที่อยู่บนพื้นฐานของข้อจำกัดนี้ทั้งหมดโดยอย่างน้อยๆ ถ้าจะทำ ระบบเครือข่ายข้อมูลของโทรศัพท์มือถือเพื่อเชื่อมโยงเข้ากับระบบเครือข่ายข้อมูล Internet นั้นจะต้อง ทำ ความสามารถต่างๆ เหล่านี้ได้

Interoperable ความสามารถในการ สื่อสารติดต่อกันของ ผู้ผลิต โทรศัพท์มือถือ ในแต่ละค่าย บน โครข่ายข้อมูลแบบ ไร้สาย

Scaleable ความสามารถในการที่จะเพิ่มมูลค่า (value added) หรือ ขยายบริการ สำหรับลูกค้า

Efficient และ Effective รองรับคุณภาพของบริการ (Service) ที่ดีและเหมาะสมตามลักษณะของโครข่าย ที่มีอยู่ เพื่อให้มี Efficient และ Effective ที่ดีที่สุด

Secure ความสามารถในการรักษาความปลอดภัยของข้อมูลบน เครือข่ายไร้สาย

Reliable ความสามารถในการเพิ่มความเชื่อถือให้กับระบบ สามารถ คาดการณ์เหตุการณ์ ที่จะเกิดขึ้นกับ ระบบได้อย่างชัดเจนมากขึ้น

การใช้ WAP สำหรับโครข่ายข้อมูล ไร้สายในปัจจุบัน การออกแบบรายละเอียดของ WAP ตั้งใจที่จะให้ ใช้โครข่ายข้อมูลเดิมในปัจจุบันให้เป็นประโยชน์มากที่สุด จะไม่เปลี่ยนโครข่ายข้อมูลในปัจจุบันที่ใช้อยู่ โดยเราจะมาดูกันว่า Requirement ในการที่จะ ทำหรือ ออกแบบ WAP ในปัจจุบันนี้ เป็นอย่างไรบ้างดังต่อไปนี้

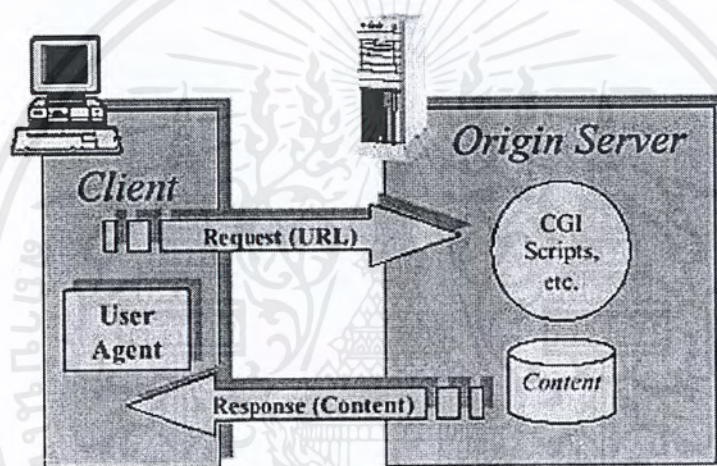
- ใช้มาตรฐานที่เกี่ยวกับโครข่ายข้อมูล ไร้สาย ให้คงไว้แบบเดิมมากที่สุด และ ยังทำงานได้ดี
- WAP จะถูกแบ่งเป็นชั้น(Layer) เพื่อความสามารถในการขยายในแต่ละชั้น อย่างอิสระซึ่งเป็นรูปแบบที่เหมาะสมใน ปัจจุบันนี้ (มอง WAP ในลักษณะ Abstraction)
- WAP จะต้องสนับสนุน โครข่ายข้อมูล ไร้สายในหลายๆแบบของโลกปัจจุบันให้มากที่สุด
- WAP ปรับแต่ง (Tune up or Optimization) การส่งข้อมูล บนช่องทางข้อมูลที่ ไม่ใหญ่นัก และ มีความล่าช้าสูง ให้ดีที่สุด
- WAP จะต้องใช้งานภายใต้ข้อจำกัดของ โทรศัพท์มือถือให้มีประสิทธิภาพมากที่สุด เช่น ใช้พลังงานต่ำ
- WAP จะต้องรองรับการขนส่งข้อมูลหรือใช้ข้อมูล ด้วยความปลอดภัย (Secure Channel)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- WAP จะต้องให้ผู้ใช้ ใช้งานได้ง่าย เหมาะสม และมีประสิทธิภาพมากที่สุด
- WAP จะต้องทำงานร่วมกับ ผู้ผลิต โทรศัพท์มือถือ แต่ละแบบ ในปัจจุบัน ให้มากที่สุด
- WAP จะต้องมีความสามารถ ในการให้ ผู้ใช้สามารถเขียน หรือ ปรับแต่ง โปรแกรม สำหรับ บริการ ที่ ผู้ใช้อยากจะเพิ่มเติม (Programming model for telephony service)

4.2 ข้อแตกต่างระหว่างเว็บ(Web) และ แว็บ (WAP)

อินเทอร์เน็ตในยุคปัจจุบันถ้ากล่าวถึงเรื่อง World-Wide Web แล้วปัจจุบันใครๆ หลายคน ก็คงจะรู้จักกัน ซึ่ง สถาปัตยกรรมของ WWW(World-Wide Web) นั้น เป็นโครงสร้างที่ มีความคล่องตัวสูง และสามารถที่จะรองรับการ โปรแกรมโดยสามารถดูได้จากรูปที่ 4.1



รูปที่ 4.1 ภาพแสดงสถาปัตยกรรมของอินเทอร์เน็ต

ตัวโปรแกรม(Application) และ ข้อมูล(Content) จะอยู่ในรูปของมาตรฐาน และสามารถเข้าถึงข้อมูลหรือโปรแกรมได้ด้วยเว็บเบราว์เซอร์ซึ่งทาง ไคลเอนต์จะเป็นผู้เรียกเข้ามา ซึ่งการทำงานของเว็บเบราว์เซอร์จะส่งคำร้อง (Request) ไปยังเซิร์ฟเวอร์ ผู้ให้บริการว่าจะเลือกเอาชื่อข้อมูล (Named Data Object) อะไรแล้วทาง Server จะตอบสนองกลับมาพร้อมกับส่งข้อมูลที่มีการเข้ารหัสตามมาตรฐานมาให้ (ภายใต้ HTTP Application Protocol)

ภายใต้โครงสร้างสถาปัตยกรรมแบบนี้เป็นโครงสร้างที่อนุญาตให้ โปรแกรมเมอร์สามารถเขียนโปรแกรม หรือพัฒนาโปรแกรม เพื่อใช้งานบนเว็บได้อย่างง่าย และมีประสิทธิภาพ (เราจะรู้จักกันในรูปของ CGI : Common Gateway Interface ซึ่งเป็นมาตรฐาน การเขียนโปรแกรม เพื่อใช้งานร่วมกับเว็บ

WWW โพรโตคอลได้จำแนกลักษณะของเซิร์ฟเวอร์ออกเป็น 3 ประเภทด้วยกัน นั่นคือ

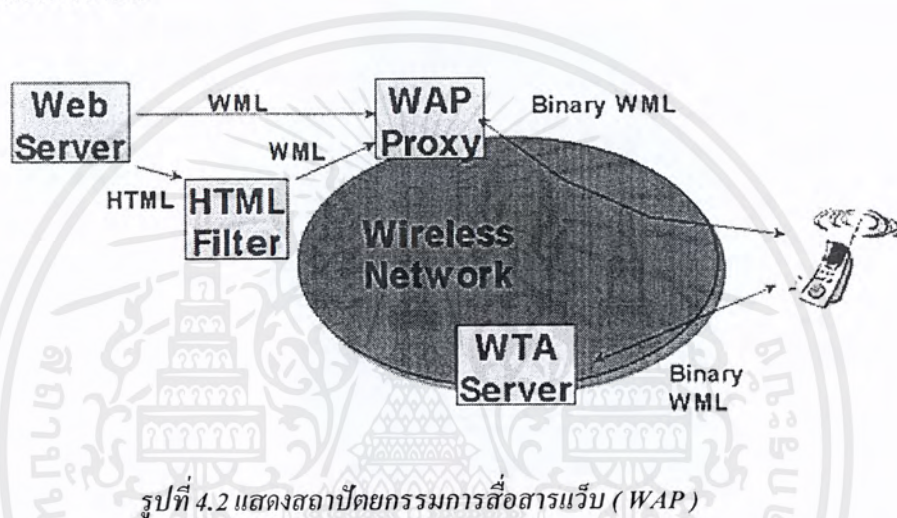
1. Origin Server ตัว เซิร์ฟเวอร์ผู้ให้บริการที่มีข้อมูล (Information) อยู่บนนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

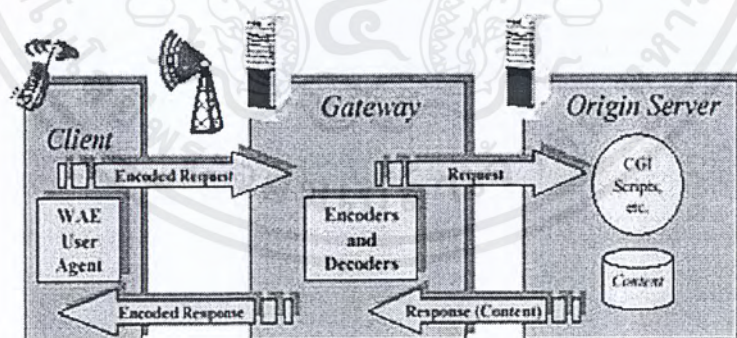
2. Proxy เป็นตัวกลางระหว่างเซิร์ฟเวอร์ และไคลเอนต์ที่มีหน้าที่เป็นตัวกลาง รับ คำร้องจากไคลเอนต์ แล้วก็ Request ต่อ ไปยังฝั่งเซิร์ฟเวอร์

3. Gateway เป็นเซิร์ฟเวอร์ที่เป็นตัวกลางเช่นเดียวกัน แต่จะไม่เหมือน Proxy ตรงที่ว่า Gateway จะรับ คำร้องจากไคลเอนต์ก็ต่อเมื่อ ตัวมันเองเป็น Origin Server มันคล้ายๆ กับว่า เป็นตัวผ่าน หรือทางผ่านของข้อมูลเพื่อเข้าถึง Origin Server ก็ว่าได้

โครงสร้างสถาปัตยกรรมของ WAP ในเชิงโปรแกรมมิ่ง (Programming) นั้นจะอ้างอิงเหมือนของ WWW ซึ่ง WWW ซึ่งมีข้อดีหลายอย่าง ตามที่กล่าวไว้ข้างต้น แต่ว่า ตัว WAP จะมีส่วนที่ จะเพิ่มเติมจาก WWW ดังรูปที่แสดงต่อไปนี้



รูปที่ 4.2 แสดงสถาปัตยกรรมการสื่อสารเว็บ (WAP)

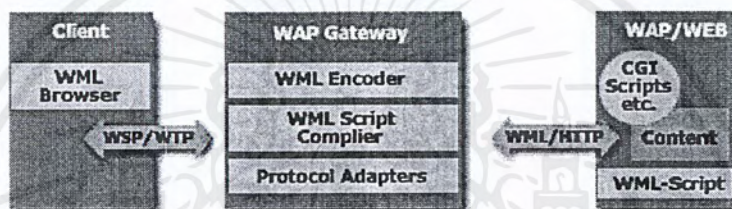
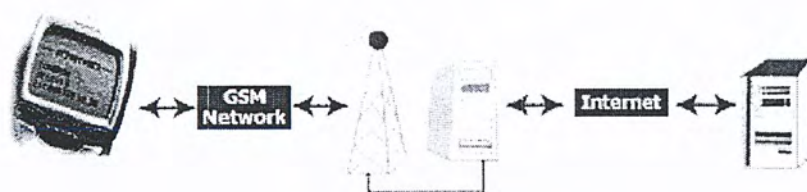


รูปที่ 4.3 แสดงรูปโปรโตคอลการสื่อสารของเว็บ (WAP)

ข้อมูลและโปรแกรมที่ใช้งานร่วมกับ WAP นั้นจะอยู่ในรูปแบบของมาตรฐานซึ่งคล้ายกับ WWW โดย ฝั่งไคลเอนต์นั้นจะใช้ไมโครเบราว์เซอร์(Micro Browser) ซึ่งปรับเอาไว้สำหรับเครื่องโทรศัพท์มือถือที่มีข้อ จำกัดในหลายๆ ด้าน แต่โดยรวมแล้ว จะเป็นเบราว์เซอร์คล้ายๆ กับของเว็บเบราว์เซอร์ แต่ตัดบางอย่างที่ไม่จำเป็นออก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAP จะใช้ Proxy เพื่อเป็นตัวเชื่อมระหว่าง เครือข่ายโทรศัพท์แบบไร้สาย เข้ากับ WWW (Internet) ซึ่ง Proxy นี้จะต้องทำหน้าที่ดังต่อไปนี้ได้



รูปที่ 4.4 แสดงรูปโปรโตคอลการสื่อสารของเว็บ (WAP) แบบที่ 2

1. โปรโตคอลเกตเวย์ (Protocol Gateway) ซึ่งจะทำหน้าที่แปลงคำร้องจาก WAP (Protocol Stack-WSP, WTP, WTLS, WDP) ซึ่งจะกล่าวถึงรายละเอียดในช่วงถัดไป) ไปยัง WWW (Protocol Stack ซึ่งอยู่บน HTTP และ TCP/IP)

2. Content Encode and Decoder จะทำหน้าที่เข้ารหัสข้อมูล WAP ให้อยู่ในรูปที่เหมาะสมกระชับ เพื่อลดขนาดของข้อมูลที่จะวิ่งบนเครือข่าย

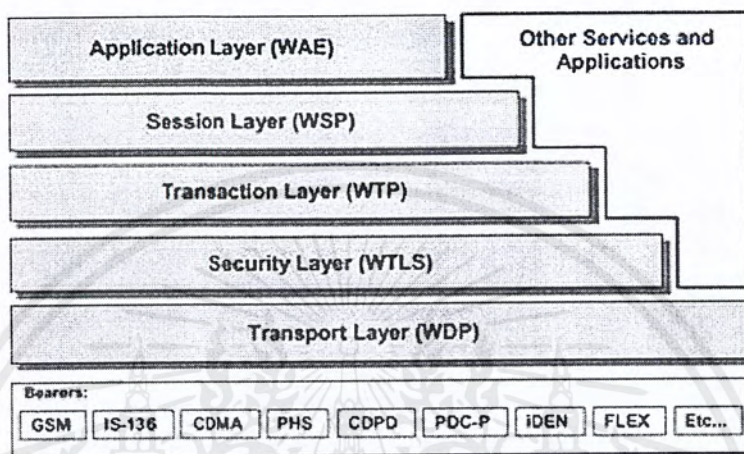
จากโครงสร้างสถาปัตยกรรมนี้จะเห็นว่าเครื่องโทรศัพท์มือถือที่จะรองรับ WAP นั้นจะสามารถเรียกใช้ข้อมูล ที่ผ่าน WAP ได้หลากหลายมาก ซึ่งนักพัฒนาโปรแกรมสามารถพัฒนาโปรแกรม เพื่อใช้งานสำหรับ WAP นั้น โดยทำงานบน WWW เซิร์ฟเวอร์ได้เช่นกัน โดยอยู่ในรูปของ CGI นั้นเอง โดย WAP จะเชื่อมต่อ การทำงาน โดยผ่าน Proxy ซึ่งจะ เป็น ตัวเชื่อมต่อ WWW เซิร์ฟเวอร์อีกที ซึ่งจะมองได้ว่า เขาสร้าง WAP ขึ้นมาเพื่อ ให้ใช้กับโทรศัพท์มือถือ ให้สามารถใช้งานอินเทอร์เน็ตได้นั่นเอง กล่าวง่าย ๆ ก็คือเอา WAP มาครอบ WWW อีกชั้นหนึ่ง นั่นเอง

4.3 องค์ประกอบของสถาปัตยกรรมของ (WAP)

WAP เป็นโปรโตคอลซึ่งแบ่งเป็นชั้น (Layer : Protocol Stack) ได้ดังต่อไปนี้ ดังรูปที่ 4.4 ซึ่งแต่ละเลเยอร์มีชื่อดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Application Layer (WAE)
2. Session Layer (WSP)
3. Transaction Layer (WTP)
4. Security Layer (WTLS)



รูปที่ 4.5 แสดงเลเยอร์ที่ใช้กับ WAP

5. Transport Layer (WDP)
6. Bearers Base.

ซึ่งจะเห็นว่าโพรโตคอลนี้จะมีกองชั้น (Stack) ซึ่งมันอนุญาตให้ผู้ที่พัฒนาแอปพลิเคชันบนชั้นที่สูงสามารถเข้าถึงชั้นล่าง ได้โดยสามารถเข้าถึงได้โดยตรงเลย หรือไม่ก็เข้าถึงโดยผ่านชั้นอื่นๆ ตามลำดับ WAP อนุญาตตรงจุดนี้ได้

4.3.1 Wireless Application Environment (WAE)

เป็นชั้นสูงสุดของ WAP ซึ่ง Application ทั่วไปจะทำงานในชั้นนี้ โดยมี พื้นฐาน ตามแบบของ World-Wide Web (WWW) และ เทคโนโลยี โทรศัพท์เคลื่อนที่ (Mobile Telephony) โดยจุดมุ่งหมาย ของการทำงาน ในชั้นนี้ก็คือ ให้สามารถทำงาน ร่วมกันได้ ในหลากหลายสภาวะ โดยที่ ทางผู้ให้บริการ สามารถสร้าง โปรแกรม หรือ Application และ บริการ ที่หลากหลาย เพื่อใช้งาน ได้ โดย WAE จะรวมถึง micro browser ซึ่ง จะทำงานดังต่อไปนี้ได้

- Wireless Markup Language (WML) ซึ่งเป็น Marking Up HTML ที่ตัดบางอย่างที่ ไม่จำเป็น ต่อโมโครบราวเซอร์ ออกนั่นเอง เพื่อความเหมาะสม ในการใช้งาน ร่วมกับ โทรศัพท์มือถือ
- WML Script ซึ่งจะคล้ายๆ กับ JavaScript แต่จะตัดบาง ส่วนออกด้วย เหตุผล เดียวกัน
- Wireless Telephony Application เป็น ลักษณะการเขียน โปรแกรม เกี่ยวกับ บริการทางการ โทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

→ Content Service จะเป็นกลุ่มรูปแบบของ การแสดงข้อมูลที่ เป็นมาตรฐาน ที่รู้จัก เช่น images, phone book records , calendar information

4.3.2 Wireless Session Protocol (WSP)

WSP คอยจัดแจงเรื่องการแลกเปลี่ยน ข้อมูลระหว่างกัน ในลักษณะของ รูปแบบ Client / Server Application โดย จัดการ เรื่องอะไรบ้างนั้นดังนี้

- ทำ เซสชัน (Session) ที่เกิดเมื่อไคลเอนต์ส่งคำร้อง ไปยังเซิร์ฟเวอร์ให้มีความน่าเชื่อถือ
- แลกเปลี่ยนข้อมูลระหว่าง ไคลเอนต์ และเซิร์ฟเวอร์โดยข้อมูลนั้นมีการเข้ารหัส (Encoding)
- แช่ และ เลิกแช่ (Suspend and Resume Session) ในระหว่างที่ มีเหตุการณ์ ถูกเดิน เช่น พลังงานต่ำ เป็นต้น

จะเห็นว่า การจัดการ WSP จัดการให้ นั้น เหมาะสมอย่างยิ่ง สำหรับ Application ประเภท browsing โดย WSP กำหนด โพรโตคอลขึ้นมาอีก 2 ตัวเพื่อใช้ทำงาน นั่นคือ

1. เป็นโพรโตคอลที่จัดการในเรื่องของ Connection session บน transaction service ซึ่งจะเน้นในเรื่องของการยืนยันข้อมูล
2. เป็นโพรโตคอลที่จะไม่ยืนยันข้อมูล เป็นลักษณะ Connectionless ซึ่งส่งไปแล้ว ส่งไปเลยทำงานอยู่บน Datagram transport service

ความสามารถที่ WSP จะต้องมี เป็นพื้นฐาน

WSP จะจัดแจงเรื่องชนิดของข้อมูล ให้สำหรับ Application Layer (WAE) โดยที่ HTTP/1.1 ในส่วนหัวของการส่งข้อมูล (Header) จะมีการกำหนด ชนิดของ ข้อมูลที่ส่งไป, ชนิดของการ เข้ารหัสอักษร ภาษา ซึ่งสิ่งต่างเหล่านี้ WSP จะเป็น ผู้จัดแจงให้กับ WAE เพื่อใช้งาน โดยตัว WSP เองจะไม่แปล หรือว่า ไปยุ่งเกี่ยวกับ Header ที่ส่งมาเหล่านั้น

อีกความสามารถ ที่น่าสนใจ นั่นคือ ช่วงเวลาชีวิตของ Session ที่เกิดขึ้นนั้น ไม่ได้ขึ้น อยู่กับ การสื่อสารข้อมูลในเลเยอร์ ชั้นล่างๆ ถัดจาก WSP , WSP สามารถใช้ คุณสมบัติในการ แช่ได้ (Suspend) เพื่อปลดปล่อย การเชื่อมต่อกับ เครือข่าย หรือ เพื่อ ประหยัดพลังงานของแบตเตอรี่ เนื่องจาก สามารถแช่ได้แล้ว ก็สามารถยกเลิก การแช่ได้เช่นเดียวกัน (Resume) โดย Resume นี้จะไม่ทำ ให้เกิด overhead ขึ้นในขณะที่ทำให้ Session กลับมา ติดต่อกันได้ เหมือนเดิม ก็เป็นอีกคุณสมบัติหนึ่ง ที่น่าสนใจ

4.3.3 Wireless Transaction Protocol (WTP)

WTP เป็นชั้นที่ทำงานอยู่บนชั้นของ Datagram service ซึ่ง WTP สามารถที่ จะทำงาน แบบปลอดภัย (Secure) กับแบบ ไม่ปลอดภัย (non-Secure) ได้โดยขึ้น อยู่กับ Application นั้นว่าจะเลือกใช้งานในชั้นของ WTLS ที่จะกล่าวต่อไปหรือไม่

WTP แบ่งการทำ Transaction Service ออกเป็น 3 ระดับด้วยกันดังนี้

- (Class 0) Unreliable one-way requests: Class นี้จะถูกใช้งาน จาก Application ที่ต้องการ "unreliable push" โดยลักษณะ การทำงานของ Class 0 จะเป็นดังนี้

เมื่อ Initiator ส่ง ข้อมูลไปยัง Responder แล้ว ในฝั่งของ Initiator ซึ่งเป็น ผู้ส่ง จะถือว่า ส่งเสร็จแล้ว เมื่อ ส่งข้อมูล นั้นไปแล้ว จะไม่สนใจ ว่า Responder จะได้รับ หรือ ไม่ได้รับ ในขณะที่ ฝั่งของ Responder จะถือว่า สื่อสารเสร็จก็ต่อเมื่อ ได้รับข้อมูลนั้นๆ จาก Initiator นี้คือ ลักษณะ การทำงานของ Class 0 ซึ่ง จะเห็นว่า มันไม่มี ภาวะควบคุม (stateless) ซึ่งจะไม่สามารถ ยกเลิกการส่งข้อมูลนั้น ได้

- (Class 1) Reliable one-way requests: Class นี้จะถูกใช้งานจาก Application ประเภทที่ ต้องการ "reliable push" โดยลักษณะการทำงานของ Class 1 นี้จะเป็นดังนี้ เมื่อ Initiator ส่งข้อมูลไปยัง Responder ข้อมูลนั้น จะ ถูกตอบรับโดย Responder โดยที่ Responder จะสามารถให้ Initiator ส่งข้อมูลนั้นมาใหม่ อีกครั้งได้ การ ทำงานของ Initiator จะสิ้นสุดลง ก็ต่อเมื่อได้รับ การตอบรับว่าได้ข้อมูล(Acknowledge) จาก Responder เป็นที่ เรียบร้อย การทำงานแบบ Class 1 นี้จะมี ภาวะควบคุมอยู่ เราจะสามารถ ยกเลิกการส่งข้อมูลนั้น ได้ตลอดเวลา ผิดจาก Class 0

- (Class 2) Reliable two-way request-reply transaction: Class นี้ จะมีลักษณะ การทำงานดังต่อไปนี้

Initiator ส่ง ข้อมูลไปหา Responder แล้ว Responder จะส่งผลลัพธ์กลับมา (มันเป็น การส่งแบบ ตอบรับ อย่างหนึ่ง ว่าได้ข้อมูลแล้ว โดยในที่นี้จะส่ง ผลกลับมาให้ด้วย) แต่ถ้า Responder ใช้เวลานาน ที่จะส่ง ผลลัพธ์ มาให้ Responder จะส่ง ข้อมูลตอบรับ กลับ ไปให้ทาง Initiator ก่อน จะเป็นลักษณะ ทางเทคนิคที่เรียกว่า "hold on" เพื่อไม่ให้ Initiator ส่งข้อมูลนั้น กลับมาซ้ำอีกรอบ เมื่อ Responder ประมวลผลจนได้ผลลัพธ์แล้ว ก็จะส่ง ผลกลับ ไปหา Initiator โดยที่ Initiator จะเป็นผู้ตอบรับไปทาง Responder ด้วย ซึ่งจุดนี้เอง Responder จะเป็นผู้ ตรวจสอบ การตอบกลับของ Initiator ว่าตอบรับว่า ได้ผลลัพธ์แล้วหรือยัง ถ้ายัง ก็จะทำการส่งใหม่อีกครั้ง แล้วก็จะ จบการทำงาน .. การทำงานแบบ Class 2 นี้ เราจะสามารถยกเลิกการส่งข้อมูลนั้น ได้ตลอดเวลา เช่นเดียวกับกับ Class 1

4.3.4 Wireless Transport Layer Security (WTLS)

WTLS นี้เป็นโพรโตคอล ที่มีความปลอดภัย โดย ออกแบบทำขึ้นตามมาตรฐานของ Transport Layer Security Protocol (TLS) ซึ่งเป็นที่รู้จักกันดี ในวงการ Internet นั่นคือ Sockets Secure Layer (SSL) โดย WTLS มีคุณสมบัติ อย่างไรบ้าง นั้นดังต่อไปนี้

- Data Integrity WTLS จะช่วยอำนวยความสะดวก และ สร้างความมั่นใจ ให้กับ ข้อมูลที่ ถูกส่ง ระหว่าง ตัวไคลเอนต์กับ Application Server ว่าจะไม่ ถูกเปลี่ยนแปลง หรือ เสียหาย

- Privacy WTLS จะคอยช่วยปกป้อง ข้อมูลที่จะส่งระหว่างไคลเอนต์ และ Application Server ว่าจะไม่ ถูกตัวกลางอื่นใด ที่ไม่ได้รับเชิญ สามารถแกะ ข้อมูลที่ เข้ารหัสเหล่านั้น และ ล้วงข้อมูลไปได้

- Authentication WTLS จะช่วยใน Authenticate ระหว่างไคลเอนต์และ Application Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Denial of service protection WTLS จะคอยตรวจสอบและสกัดกั้นข้อมูล ที่ไม่ได้ผ่าน การตรวจสอบ (Verified) อย่าง

สมบูรณ์ ออก เพื่อความ ปลอดภัย และ ปกป้องเลขเอร์ของโพร โทคอลที่สูงกว่า.

WTLS สามารถใช้ช่องทางที่ปลอดภัยนี้ ระหว่าง ตัวโทรศัพท์มือถือ (Terminal) ด้วยกันได้ เช่น อาจจะใช้ ทำการตรวจสอบ ข้อมูลประเภท electronic business card เป็นต้น WTLS นี้จะถูกแอฟพลิเคชันในเลเยอร์ที่สูงกว่า เลือกว่าจะใช้ WTLS หรือว่าจะไม่ใช้ มันขึ้นอยู่กับ ความต้องการ ในความปลอดภัย ว่ามีมากน้อยเพียงใด และ ก็ขึ้นอยู่กับ ลักษณะของ โครงข่ายข้อมูลในเลเยอร์ต่ำ ซึ่งถ้ามันรองรับเรื่องความปลอดภัยอยู่แล้ว การ ใช้ WTLS อาจจะถูกตัดทิ้งไปก็ได้ เป็นต้น

4.3.5 Wireless Datagram Protocol (WDP)

เป็นโพรโทคอลที่เกี่ยวกับการขนส่งข้อมูล ใน โครงสร้างสถาปัตยกรรม WAP นั้น จะยกให้กับ WDP โดย ที่ WDP จะทำงานอยู่บน บริการฐานรองรับ ของ โครงข่ายสื่อสาร ข้อมูลแบบ ไร้สาย(bearer) หลายๆ ประเภท เช่น GSM, CDMA, PHS, FLEX, etc WDP เป็นโพรโทคอลที่เป็น ตัวเชื่อม(Interface) ร่วมกันของโพรโทคอลเลเยอร์ที่สูงๆ ของ WAP เช่น Security, Session, Application ซึ่ง โพรโทคอลเหล่านี้ไม่ต้องห่วงเรื่องช่อง โครงข่ายสื่อสาร ข้อมูล แบบไร้สาย (bearer แบบต่าง ๆ นั้นเอง) เพราะมันเป็น อิสระต่อกัน ตามคุณสมบัติของการออกแบบให้เป็น Layer

4.3.6 Bearers

WAP เป็นโพรโทคอลที่ถูกออกแบบขึ้นมาเพื่อใช้งานบน Bearers หลากหลายแบบ อาทิเช่น short message, circuit-switched data, packet data เป็นต้น แต่ละชนิด ก็จะมีระดับคุณภาพ แตกต่างกันไป ก็จะมี เรื่อง อัตราเร็วในการส่งข้อมูล อัตราความ ผิดพลาด และ ในเรื่องของ ความล่าช้าแตกต่างกันไป ซึ่ง WAP จะ ต้องรองรับ การทำงาน ในแต่ละ Bearers ให้มีประสิทธิภาพ มากที่สุด ซึ่ง WDP ซึ่งเป็น Protocol Layer หนึ่งใน WAP นั้นจะต้องรองรับ การทำงานของ Bearers และ WAP Stack Protocol ส่วนอื่นๆ ซึ่ง WDP จะต้องรองรับการทำงาน ของ Bearers ในแต่ละประเภท ให้ครอบคลุมมากที่สุด เพราะว่า Bearers ก็จะมีการเปลี่ยนแปลง ตลอด อาจจะมี Bearers ชนิดใหม่เข้ามาเรื่อยๆ ก็ได้ ซึ่ง WDP จะต้องรองรับได้.

4.4 ประโยชน์จากการใช้ WAP

การใช้บริการอินเทอร์เน็ต ไม่ว่าจะ เป็น โทรศัพท์มือถือรุ่นใด จะมีลักษณะเหมือนกัน คือ ต้องทำการตั้งค่า การติดต่อเช่น ไอที แอดเดรส, มีชื่อยูสเซอร์และพาสเวิร์ด เป็นต้น ติดต่อผ่านศูนย์บริการหรือเกตเวย์ (Gateway)ซึ่งการทำงานจะเป็นไปในลักษณะเช่นเดียวกับการต่อ โมเด็มบนพีซีเพื่อเข้าอินเทอร์เน็ต ในรายละเอียดคงต้องศึกษาจากคู่มือของแต่ละผลิตภัณฑ์นั้น และเมื่อต่อเข้าได้แล้ว เราก็จะสามารถเรียกดูข้อมูลผ่าน WAP Server ที่ผู้ให้บริการอินเทอร์เน็ตจัดบริการไว้ การเปิดบริการ WAP ของ Nokia 7110 ต้องทำการยื่นเรื่อง ขอใช้บริการและติดตั้งเองส่วนทางอิริคสัน R320s ได้ติดตั้งไว้เรียบร้อยแล้วให้คุณใช้ได้ทันทีเลย เมื่อมีมือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ท่องอินเทอร์เน็ตได้ จึงเหมือนกับการพกพาข่าวสารไปด้วยทุกที่ เรียกดูข้อมูลได้อย่างต่อเนื่องและทุกช่วงเวลา ทั้งการพยากรณ์อากาศของทุกจังหวัดทั่วประเทศ การรายงานการจราจร (<http://wap.mweb.co.th>) ข่าวทางด้านธุรกิจ (<http://www.asiao.nthego.com/mainguide.wml>) การเมือง ข่าวบันเทิง หรือการเงิน เช่น เรียกดูอัตรา การแลกเปลี่ยนเงินตราของเงินสกุลหลัก (<http://wap.samarts.com>) รายงานข่าวราคาซื้อขายหุ้นทั่วโลก แบบเรียลไทม์ (www.wap.ast.co.th, <http://wap.globalthai.com>) เป็นต้น ผลการแข่งขันกีฬานัดสำคัญ หลากหลายประเภท (<http://wap.thaigo.lfer.com>) หรือ จะผ่อนคลายด้วยการอ่านดวงชะตา หาข้อมูลการท่องเที่ยวได้ เช่น ที่พัก, สายการบินในกรุงเทพฯ นึกอยากดูหนัง ก็ตรวจสอบเวลาของเครื่องวีซีดีทันที (<http://wap.wopwap.com>) แสดงอันดับเพลงฮิต เรียกอ่านอีเมลล์ได้โดยตรงอีกด้วย เช่น ของ www.thaicentral.com สามารถดูตารางเวลา และสำรองที่นั่งสายการบิน (<http://202.183.205.97/index.wml>) จองตั๋วภาพยนตร์ หาข้อมูลตำแหน่งงาน (<http://wap.jobpilot.co.th>) รวมไปถึงข้อมูลอาหารและสั่งอาหาร (<http://wap.a-roi.com>) ได้อย่างสะดวกรวดเร็ว ผ่อนคลายด้วยการแะดูเว็บไซต์ต่างๆ หรือจะส่ง ICQ message ส่งการ์ด ก็ยังได้ ทั้งนี้บริการดังกล่าวช่วงนี้ยังไม่เสียค่าบริการ เสียเพียงค่าโทรศัพท์เข้าศูนย์บริการเท่านั้น เพราะยังอยู่ในช่วงเวลาพิเศษเปิดตัวโครงการต่างๆ คงทยอยเปิดออกมาบริการอีกมากมายแน่นอน การเลือกใช้มือถือแบบใด จึงขึ้นอยู่กับความพอใจในระบบบริการอื่นในแต่ละเครื่องของผู้ใช้ของเราเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 WML (Wireless Markup Language)

5.1 โครงสร้างของเอกสาร WML

โครงสร้างของเอกสาร WML หรือ เว็บ ไซต์ ถูกออกแบบมาให้มีการทำงานเหมือนกับไฟ ที่เราคงเคยเล่นกันมาบ้างแล้วในชีวิตจริง โดยที่ไฟแต่ละใบหรือการ์ด เปรียบได้กับ เว็บเพจหนึ่งหน้า และเมื่อเรารวมการ์ด หรือเว็บเพจหลาย ๆ หน้าเข้าด้วยกันเป็นสำรับ หรือที่เรียกว่าเด็ค ก็จะเป็นเว็บไซต์ได้นั่นเอง ซึ่งการที่ WML ถูกออกแบบให้มีลักษณะเหมือนไฟหลาย ๆ ใบในสำรับหนึ่ง ก็เพราะอุปกรณ์เว็บ จะได้ไม่เสียเวลาในการส่งคำร้องขอไปยังเซิร์ฟเวอร์หลายครั้ง ซึ่งเป็นการช่วยลดเวลาและแบนด์วิดธลง

ดังนั้น โครงสร้างที่ใหญ่ที่สุดของเอกสาร WML คือ เด็ค ซึ่งก็คือเนื้อหาทั้งหมดที่อยู่ในเอกสาร WML ในแต่ละ เด็คจะประกอบด้วยกลุ่มเอกสารย่อย ๆ ซึ่งเรียกว่า การ์ด ในแต่ละการ์ดจะเป็นที่เก็บข้อมูลต่าง ๆ ที่เราต้องการให้แสดงที่หน้าจอ รวมไปถึงคำสั่งต่าง ๆ ที่เราต้องการให้เอกสาร WML ของเราดอบสนองต่อความต้องการของผู้ใช้ เช่น การทอ้งไปยังหน้า (การ์ด) ต่าง ๆ เป็นต้น

เอกสารต่อไปนี้แสดงคำว่า Sawadee WML ที่หน้าจอ เอกสาร WML นี้เป็นตัวอย่างแสดงให้เห็นว่าในเอกสาร WML นั้น อย่างน้อยจะต้องประกอบไปด้วยอะไรบ้าง

บรรทัดที่	เอกสาร WML
1	<?xml version = "1.0"?>
2	<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML1.1/EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
3	<wml>
4	<card>
5	<!-- Write Sawasdee WML to screen -->
6	<p> Sawasdee WML </p>
7	</card>
8	</wml>

ถ้าเรามีความคุ้นเคยกับเอกสาร HTML เราจะเห็นว่า WML นั้นมีลักษณะคล้ายกับ HTML จากตัวอย่างในแต่ละบรรทัดมีความหมายดังนี้

บรรทัดที่	ความหมาย
1-2	เป็นส่วนที่เรียกว่า เฮดเดอร์ ซึ่งเป็นส่วนที่บอกบราวเซอร์ว่าเอกสารนี้เป็นเอกสาร WML เวอร์ชัน 1.0 ในเอกสาร WML ทุกเอกสารจะต้องมีเฮดเดอร์นี้ มิฉะนั้นจะทำให้การแสดงผลข้อมูลไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3 และ 8 เป็นจุดเริ่มต้นและจุดสิ้นสุดของเด็ก (<wml>...</wml>) เราจะเห็นว่าลักษณะของแท็กจะ คล้ายกับ HTML คือ <html>...</html>
- 4และ 7 เป็นจุดเริ่มต้นและจุดสิ้นสุดของการ์ด (<card>...</card>)
- 5 เป็นส่วนหมายเหตุ (<!-- ... -->) ข้อความที่อยู่ในแท็กหมายเหตุนี้จะไม่ถูกแสดง โดยบราวเซอร์ ประโยชน์ของมันก็คือเป็นที่สำหรับเขียนบันทึกเตือนความจำ หรือคำอธิบายต่าง ๆ
- 6 เป็นส่วนแสดงข้อมูลสู่หน้าจอ (<p>...</p>) ข้อความต่าง ๆ ที่อยู่ภายในแท็กนี้จะถูกแสดงที่หน้าจอของบราวเซอร์ ซึ่งก็คือจอของเครื่องมือถือนั่นเอง

5.2 WML Tag

แท็ก <Tag> คือ สิ่งที่บอกให้บราวเซอร์ทราบว่ามันควรจะทำอะไรกับข้อมูลที่อยู่ภายในแท็กนั้น ๆ แท็กจะอยู่คู่ลักษณะคือ แท็กที่มีแท็กย่อย และแท็กเดี่ยว

แท็กที่มีแท็กย่อย จะมีแท็กเริ่มต้น และแท็กสิ้นสุด เช่น แท็กการ์ด จะมี <card> เป็นแท็กเริ่มต้น และ </card> เป็นแท็กสิ้นสุด แท็ก p เป็นส่วนหนึ่งในในแท็กย่อยของแท็กการ์ด และเราคงจะเดาได้ว่าแท็ก p ก็เป็นแท็กที่มีแท็กย่อยเช่นกัน เพราะแท็ก p มีแท็กเริ่มต้น <p> และแท็กสิ้นสุด </p>

แท็กเดี่ยว คือ แท็กที่ไม่มีแท็กย่อย แท็กประเภทนี้จะมีลักษณะดังนี้ <ชื่อแท็ก/> เช่น
, <prev/>, <noop/> เป็นต้น

5.2.1 แท็กแอตทริบิวต์

แท็กแอตทริบิวต์คือ คุณสมบัติ และค่าต่าง ๆ ที่เราสามารถกำหนดได้ สำหรับแท็กนั้น ๆ เช่น ชื่อ ความกว้าง ความยาว การจัดวางตัวอักษร เป็นต้น

5.2.2 แท็ก p (<p>...</p>)

แท็กนี้ใช้สำหรับแสดงข้อมูลสู่จอภาพ ดังนั้นจึงมีแอตทริบิวต์ที่ใช้กำหนดการจัดวางตัวอักษร และการแสดงตัวอักษรคือ แอตทริบิวต์ align และ mode

5.2.3 แอตทริบิวต์ align

แอตทริบิวต์ align ใช้กำหนดการจัดวางตัวอักษร ซึ่งเราสามารถกำหนดได้ 3 แบบ คือชิดซ้าย ชิดขวา และตรงกลาง (left,right,center) วิธีการกำหนดค่าทำได้โดยพิมพ์ align="ค่าที่ต้องการกำหนด" ไว้ในส่วนเริ่มต้นของแท็ก (<p>) เช่น <p align="center">...</p>

5.2.4 แอตทริบิวต์ mode

แอตทริบิวต์ mode ใช้สำหรับจัดการกับข้อความในแต่ละบรรทัดว่าจะให้ขึ้นบรรทัดใหม่หรือไม่ ถ้าข้อความนั้นมีความยาวเกินกว่าที่หน้าจอจะแสดงได้ในหนึ่งบรรทัด ค่าของแอตทริบิวต์นี้คือ wrap และ nowrap

- wrap หมายถึง กำหนดให้การแสดงตัวอักษร จะต้องขึ้นบรรทัดใหม่ ถ้าข้อความมีความยาวมากกว่าที่จะแสดงได้ในหนึ่งบรรทัด

- nowrap หมายถึง กำหนดให้การแสดงตัวอักษร ไม่ต้องขึ้นบรรทัดใหม่ถ้าข้อความมีความยาวมากกว่าที่จะแสดงได้ในหนึ่งบรรทัด วิธีการกำหนดค่าทำได้โดยพิมพ์ mode = “ค่าที่ต้องการ” ในส่วนเริ่มต้นของแท็ก (<p>) เช่น <p mode = “wrap”>

5.2.5 แท็ก card (<card>...</card>)

แท็ก card กำหนดจุดเริ่มต้นและจุดสิ้นสุดของการ์ด ในเด็กเล็กหนึ่งสามารถมีการ์ดได้หลาย ๆ การ์ด ซึ่งแต่ละการ์ดก็จะบรรจุข้อมูลและวิธีการตอบโต้กับผู้ใช้ที่แตกต่างกันไป

5.2.6 แอตทริบิวต์ title แอตทริบิวต์ title ใช้สำหรับใส่ข้อความที่เป็นชื่อ

5.2.7 แอตทริบิวต์ id แอตทริบิวต์ id ใช้สำหรับกำหนดชื่อให้กับการ์ด เพื่อสามารถอ้างอิงจากการ์ดอื่น ๆ

โครงสร้างของเอกสาร WML นั้นจะต้องประกอบด้วยส่วนหัวซึ่งเป็นส่วนที่ใช้บอกให้เบราว์เซอร์ทราบว่าเอกสารของเราเป็นเอกสาร WML จากนั้นเป็นส่วนเด็กซึ่งเป็นส่วนที่บรรจุข้อมูลต่าง ๆ ที่เราต้องการนำเสนอ ข้อมูลเหล่านี้จะถูกเก็บไว้ในหน้าย่อย ๆ ที่เรียกว่าการ์ด ในการ์ดก็จะมีแท็กต่าง ๆ ที่ทำหน้าที่ควบคุมการแสดงผล เราจะเห็นว่าการสร้างเอกสาร WML นั้นทำได้ไม่ยากและมีลักษณะคล้ายกับเอกสาร HTML

5.3 การตกแต่งข้อความ

การนำเสนอข้อมูลที่เป็นตัวอักษรนั้นจะต้องอยู่ในแท็ก p (<p>...</p>) และเราก็ทราบอีกว่าแท็ก p นี้เป็นแท็กที่มีแท็กย่อย เรามาดูกันว่าแท็กย่อยเหล่านี้มีอะไรที่สามารถใช้ตกแต่งข้อความได้

- ขึ้นบรรทัดใหม่ด้วยแท็ก br (
) ใช้สำหรับกำหนดให้ขึ้นบรรทัดใหม่สำหรับข้อความที่อยู่ต่อจากแท็กนี้ ซึ่งจะเหมือนกับแท็ก br ของ HTML
- ทำให้ตัวอักษรเด่นขึ้นด้วยแท็ก em (...)
- ทำให้ตัวอักษรเด่นขึ้นด้วยแท็ก strong (...)
- ทำให้ตัวอักษรเอียงด้วยแท็ก i (<i>...</i>)
- ทำให้ตัวอักษรหนาขึ้นด้วยแท็ก b (...)
- ชิดเส้นใต้ให้กับตัวอักษรด้วยแท็ก u (<u>...</u>)
- ทำให้ตัวอักษรใหญ่ขึ้นด้วยแท็ก big (<big>...</big>)
- ทำให้ตัวอักษรเล็กลงด้วยแท็ก small (<small>...</small>)

5.4 การนำเสนอข้อมูลในลักษณะตาราง

- แท็ก table (<table>...</table>) ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของตาราง
- แท็ก tr (<tr>...</tr>) ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของแถวหนึ่งแถวในตาราง
- แท็ก td (<td>...</td>) ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของคอลัมน์หนึ่งคอลัมน์ในตาราง

รูปแบบของแท็กทั้งสามจะต้องเป็นดังต่อไปนี้

รูปแบบที่ 1 ตารางที่มีหนึ่งแถวและหนึ่งคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<table>
  <tr>
    <td> ข้อมูลที่ต้องการนำเสนอ</td>
  </tr>
</table>

```

รูปแบบที่ 2 ตารางที่มีหนึ่งแถวแต่หลายคอลัมน์

```

<table>
  <tr>
    <td> ข้อมูลที่ต้องการนำเสนอ</td>
    <td> ข้อมูลที่ต้องการนำเสนอ</td>
  </tr>
</table>

```

รูปแบบที่ 3 ตารางที่มีหลายแถว แถวละหนึ่งคอลัมน์

```

<table>
  <tr>
    <td>ข้อมูลที่ต้องการนำเสนอ</td>
  </tr>
  <tr>
    <td>ข้อมูลที่ต้องการนำเสนอ</td>
  </tr>
</table>

```

5.4.1 แอตทริบิวต์ที่ควรรู้ของแท็ก table

แอตทริบิวต์ columns (<table columns="จำนวน">) ใช้สำหรับระบุจำนวนคอลัมน์ที่มีในตาราง ซึ่งเราจะต้องกำหนดค่าให้แอตทริบิวต์ทุกครั้ง ถ้าหากจำนวนแท็ก td มีน้อยกว่าค่าที่เรากำหนดให้กับแอตทริบิวต์นี้ บรรทัดจะทำการเพิ่มคอลัมน์ว่างให้โดยอัตโนมัติ แต่ถ้าหากจำนวนแท็ก td มีมากกว่าค่าที่เรากำหนด บรรทัดจะใส่คอลัมน์ที่เกินมาลงทางซ้ายของตาราง

แอตทริบิวต์ align (<table align = "ค่าที่ต้องการ">) ใช้สำหรับจัดวางตัวอักษรหรือรูปภาพที่แสดงในแต่ละคอลัมน์ ค่าที่เราสามารถกำหนดให้แอตทริบิวต์ได้ ได้แก่ C,L และ R

- C หมายถึง จัดให้อยู่กึ่งกลางคอลัมน์
- L หมายถึง จัดให้อยู่ชิดซ้ายของคอลัมน์
- R หมายถึง จัดให้อยู่ชิดขวาของคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดการจัดวางนี้สามารถระบุให้แต่ละคอลัมน์ได้ด้วยการระบุค่าเรียงกันไป เช่น align = "CRR" ซึ่งจะหมายถึง คอลัมน์ที่ 1 จัดให้อยู่ตรงกลาง คอลัมน์ที่ 2 และ 3 จัดให้อยู่ชิดขวา

5.4.2 การนำเสนอข้อมูลในตาราง

ข้อมูลต่าง ๆ ที่ต้องการนำเสนอจะต้องอยู่ในแท็ก td ดังนั้นภายในแท็ก td จึงมีแท็กย่อยอื่น ๆ อีกเพื่อใช้ควบคุมการแสดงผลภาพ หรือการตอบโต้กับผู้ใช้แท็กเหล่านั้น ได้แก่

- แท็ก a (<a>), แท็ก anchor (<anchor>) สองแท็กนี้ใช้เหมือนกันคือ ใช้สำหรับเชื่อมโยงไปยังข้อมูลส่วนอื่น หรือการ์ดอื่น

- แท็ก br (
) ใช้ขึ้นบรรทัดใหม่

- แท็กที่ใช้ในการตกแต่งตัวอักษรต่าง ๆ ได้แก่ em, strong, b, i, u, big และ small

- แท็ก img () ใช้สำหรับแสดงรูปภาพ

5.5 การสร้างลิงค์

การใช้ WAP เพจ เพื่อนำเสนอข้อมูลนั้น มีลักษณะคล้ายการทำหนังสือพิมพ์ แต่ต่างกันตรงที่สามารถเปิดไปยังหน้า หรือ Card ของข้อมูลที่ต้องการนั้นได้เลย โดยไม่ต้องพลิกเปลี่ยนทีละหน้าเช่นเดียวกับโครงสร้างของเอกสาร WML จะประกอบไปด้วย Deck หนึ่งสำหรับ และใน Deck หนึ่งสำหรับนั้น ปกติก็จะประกอบไปด้วย Card หลาย ๆ ใบ ซึ่งเราสามารถเปรียบเทียบได้กับเอกสาร HTML ที่เว็บไซต์หนึ่งจะประกอบไปด้วยเว็บเพจหลาย ๆ หน้านั่นเอง การที่เราจะเปลี่ยน Card จากใบหนึ่งไปสู่อีกใบหนึ่ง หรือระหว่าง Deck นั้นสามารถกระทำได้จาก

@ เกิดจากการกระทำอย่างใดอย่างหนึ่ง (Manual) เช่น จากการทำเรคคอร์ดชอฟิลด์ หรือเลือกลิงค์เชื่อมโยงไปสู่ Card หรือ Deck อื่น

@ เกิดขึ้นโดยอัตโนมัติ จากการตั้งเวลา เพื่อให้เอกสาร WML เปลี่ยนการ์ดหรือเด็คเอง

5.5.1 การสร้างลิงค์ด้วยแท็ก <do>

แท็ก <do> เป็นเครื่องมือที่ใช้ในการช่วยให้ผู้ใช้สามารถเข้าถึง และเชื่อมโยงจากการ์ดที่แสดงอยู่ในขณะนั้นไปสู่การ์ดหรือ URI อื่น ๆ ได้ ซึ่งแท็ก <do> จะทำหน้าที่เชื่อมต่อระหว่างสิ่งที่แสดงออกมาบนหน้าจอ ภาพ (อินเตอร์เฟซต่าง ๆ เช่น เมนู และปุ่มต่าง ๆ ที่แสดงบนหน้าจอ ฟังก์ชันที่กำกับซอฟต์แวร์ หรือเครื่องรับคำสั่งจากเสียง) กับงานที่เราต้องการให้หน้า WAP ทำ เมื่อแท็ก <do> นี้ถูกเรียกใช้ ซึ่งเราสามารถเขียนคำสั่งงานต่าง ๆ (Task) กับแท็ก do ได้ดังนี้ <do>< คำสั่งงาน...</do>

เราสามารถใส่แท็ก <do> ได้ทั้งในระดับการ์ด และเด็ค ซึ่งในระดับเด็ค นี้จะเป็นการกำหนดในธีมเพลตอีกที เรามาทำความรู้จักกับค่าแอตทริบิวต์ที่น่าสนใจของแท็ก <do> กัน ซึ่งมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

@ แอตทริบิวต์ type เป็นตัวแสดงแก่เครื่องโทรศัพท์มือถือ wap บราวเซอร์ ให้ทราบถึงเจตนาที่โปรแกรมเมอร์ต้องการเขียนแท็กนั้นขึ้นมาเพื่องานใด และแอตทริบิวต์นี้ยังช่วยให้บราวเซอร์สร้างอินเตอร์เฟซที่เหมาะสมกับงานนั้น ๆ ด้วยค่าของแอตทริบิวต์ที่มีได้คือ

- accept เป็นค่าที่ใช้กันมาก มีไว้สำหรับแสดงการยอมรับให้ทำงานตามที่กำหนด
- prev เป็นตัวแสดงถึงการย้อนกลับไปยังหน้าก่อน
- help เป็นตัวแสดงความต้องการความช่วยเหลือ
- options เป็นตัวแสดงถึงตัวเลือก หรือการทำงานอื่น ๆ ที่สามารถเพิ่มเติมได้
- reset เป็นตัวแสดงถึงการลบ หรือเริ่มต้นการทำงานใหม่ ๆ
- delete เป็นตัวแสดงถึงการลบวัตถุ หรือตัวเลือกต่าง ๆ
- unknown เป็นชนิดที่เราไม่ได้กำหนดลงไป เช่น type="" หรือเกิดจากการพิมพ์ค่าของแอตทริบิวต์ที่ไม่มีในนิยาม หรือพิมพ์ผิด

@ label เป็นแอตทริบิวต์ที่เราสามารถกำหนดชื่อของลิงค์ ที่จะแสดงออกมาบนหน้าจอในเมนูได้

@ menu เป็นแอตทริบิวต์ที่ใช้กำหนดชื่อของแท็ก <do> เพื่อใช้ในการอ้างอิง

@ option เป็นแอตทริบิวต์ที่อนุญาตให้ WAP บราวเซอร์ คิดว่าแท็กนี้เป็นตัวเลือก ซึ่งจะนำมาใช้หรือไม่ได้โดยเราสามารถกำหนดค่าเป็น "true" หรือ "false" ถ้ากำหนดเป็น "true" เครื่องอาจจะไม่นำแท็กนี้มาใช้ก็ได้

5.5.2 การสร้างลิงค์จากแท็ก <anchor> และ <a>

แท็ก <anchor> มีไว้สำหรับสร้างลิงค์ เพื่อเชื่อมโยงระหว่างการ์ด หรือหน้าเว็บเพจ ซึ่งมีลักษณะของการทำงานเช่นเดียวกับแท็ก <a> ในภาษา HTML แท็ก <anchor> สามารถทำงานได้เช่นเดียวกับ <do> แต่จะต่างกันตรงที่ข้อความในแท็ก <anchor> จะไม่ปรากฏในเมนู Options ของซอฟต์แวร์ แต่จะปรากฏหน้าจอโดยมีลักษณะเป็นลิงค์ที่เราสามารถเลือกได้เลย และเนื่องจากแท็กนี้ จะต้องปรากฏหน้าจอ เราจึงต้องใส่แท็กนี้ภายในระหว่างแท็ก <p> ด้วย รูปแบบการใช้แท็ก <anchor> มีดังนี้

```
<anchor> ชื่อของลิงค์
<go href="ชื่อ URI หรือการ์ด ที่ต้องการเชื่อมโยงไปถึง">
</anchor>
```

จะสังเกตได้ว่าในแท็ก <anchor> นี้จะต้องมีคำสั่งของงาน (Task) ที่ทำหน้าที่เปลี่ยนการ์ด หรือหน้าของเว็บเพจ เมื่อลิงค์ของแท็กนี้ถูกเลือก ซึ่งแท็กของงานที่สามารถบรรจุอยู่ในแท็ก <anchor> นี้ได้คือ go, prev และ refresh ส่วนแท็ก <a> นั้นเปรียบเสมือนเป็นตัวย่อ หรือชอร์ตคัตของแท็ก <anchor> กับแท็กคำสั่งของงาน <go> นั่นเอง ซึ่งเราสามารถเขียนลิงค์ที่ใช้แท็ก <anchor> จาก

```
<anchor> go to my link
<go href="my URI">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

</anchor>

โดยใช้แท็ก <a> แทนได้ดังนี้ go go my link จากนั้นไปเราจะมาทดลองสร้างลิงค์โดยใช้แท็ก <anchor> และ <a> เพื่อเชื่อมต่อระหว่างหน้า WAP เพจ หรือการ์ดกัน

5.5.3 แท็กของเหตุการณ์ต่าง ๆ (Event) ที่ใช้สำหรับเปลี่ยน URI

เราสามารถควบคุมเหตุการณ์ต่าง ๆ ที่จะเกิดขึ้นกับเครื่อง PDA ได้ ทั้งที่เกิดขึ้นจากตัวเครื่องเอง หรือจากผู้ใช้ เช่น กรณีที่ผู้ใช้กดที่คีย์หรือปุ่มต่าง ๆ เพื่ออ่านหน้า WAP เพจ หรือการที่เราทิ้งเครื่องไว้เฉย ๆ โดยไม่มีการทำงานใด ๆ เกิดขึ้นเลย

โดย WML ได้สร้างแท็กของเหตุการณ์ต่าง ๆ ขึ้นมาเพื่อรองรับการทำงานที่จะเกิดขึ้นเหล่านั้น แท็กของเหตุการณ์จะอนุญาตให้เราสามารถเขียนโปรแกรมขึ้นมาควบคุมเหตุการณ์ต่าง ๆ และทำให้โปรแกรมทำงานตามที่เรากำลังได้ ตัวอย่างเช่น เราสามารถเขียนโปรแกรมไว้สำหรับเหตุการณ์ เมื่อผู้ใช้คลิกเลือกที่เมนู และจะทำให้หน้าของเว็บเพจเปลี่ยนไปตามที่เราต้องการ เราจะมาพิจารณารายละเอียดของแท็กที่มีกันดังนี้

- ontimer เหตุการณ์ ontimer เราจะใช้เป็นแอตทริบิวต์ในแท็กการ์ด และเพิ่มเฟลต โดยส่วนใหญ่เราจะใช้เหตุการณ์ ontimer นี้ เมื่อเราต้องการกำหนดว่าหน้า WAP เพจของเราจะเปลี่ยนไปสู่การ์ดใหม่ หรือที่อยู่ใหม่ชื่ออะไร ในระยะเวลาที่เรากำหนด ซึ่งเราจะใช้คู่กับแท็ก <timer> ซึ่งเป็นแท็กที่ช่วยในการกำหนดระยะเวลาที่เราต้องการนั่นเอง การนับเวลาของ timer จะเป็นการนับถอยหลังตั้งแต่เราเริ่มเปิดการ์ดที่มีแท็กนี้ขึ้นมา และจะหยุดเวลาเมื่อเราออกจากการ์ดใบนั้น timer นี้จะนับเวลาถอยหลังไปเรื่อย ๆ จนกระทั่งหมดเวลา หรือได้เวลาเท่ากับ 0 จากนั้นจึงจะทำการบังคับให้เหตุการณ์ที่เรากำหนดใน timer ทำงาน

5.6 แท็กคำสั่งของงาน (Task)

จากในหัวข้อที่ผ่านมา เราจะพบว่าการสร้างลิงค์ด้วยแท็ก <do> หรือการเขียนโปรแกรมควบคุมเหตุการณ์ต่าง ๆ ที่จะเกิดขึ้นจากแท็กของเหตุการณ์ (Event) ต่างต้องการคำสั่งของงานในระหว่างแท็กนั้น ๆ ด้วย เช่น <go href="URI"> เพื่อทำหน้าที่สั่งให้หน้าของเว็บเพจเปลี่ยนไปตามที่เราต้องการ ในหัวข้อต่อไปนี้จะมาศึกษารายละเอียดของคำสั่งต่าง ๆ ที่เกี่ยวกับงานที่มีและใช้ใน WML

5.6.1 แท็ก <go> เป็นแท็กที่ใช้ในการกำหนดให้เว็บเบราว์เซอร์เปลี่ยนหน้าของเว็บเพจไปตามตำแหน่งที่กำหนดขึ้นโดยแอตทริบิวต์ href อาจจะเป็น URI หรือชื่อของการ์ดหรือเด็กในไซด์นั้นก็ได้ ดังนี้ <go href=URI />

และแท็ก <go> มีแอตทริบิวต์ที่น่าสนใจดังนี้

@ href เป็นแอตทริบิวต์ที่กำหนด URI หรือชื่อของการ์ดที่ต้องการเชื่อมต่อไปถึง โดยถ้าเราเขียนชื่อของไฟล์ WML ที่ต้องการเปิดเป็น <go href="thedeck.wml"/> นั่นเอง

@ method เป็นแอตทริบิวต์ที่กำหนดวิธีการของการส่งค่าผ่านโปรโตคอล HTTP ซึ่งมีค่าได้ 2 อย่างคือ POST หรือ GET

5.6.2 แท็ก <prev> แท็กนี้มีไว้ใช้สำหรับสร้างลิงค์กลับไปหน้าเก่า เพื่อให้เราสามารถย้อนกลับไปยังหน้าที่ผ่านมาแล้ว เมื่อเราคลิกที่ตัวเลือกเพื่อใช้แท็กนี้ ค่าของ URI ที่อยู่ในบันทึก (stack) ของ history จะถูกนำออกมาอ่านและเว็บเพจจะเปลี่ยนเป็นหน้าที่นำออกมาจาก history นั้นเอง

5.7 การรับข้อมูลจากผู้ใช้

เช่นเดียวกับเอกสาร HTML เอกสาร WML สามารถรับข้อมูลจากผู้ใช้ และส่งข้อมูลเหล่านั้นกลับไปยังเซิร์ฟเวอร์ได้ แท็กที่ใช้ในการรับข้อมูล ได้แก่ แท็กอินพุต, แท็ก select และแท็ก option ส่วนการส่งข้อมูลกลับไปยังเซิร์ฟเวอร์ให้ใช้แท็ก do

- แท็ก input (<input/>) ใช้รับข้อมูลที่เป็นตัวอักษร ซึ่งพิมพ์โดยผู้ใช้แอดทริบิวต์ที่น่าสนใจมีดังต่อไปนี้

- แอดทริบิวต์ name เป็นตัวกำหนดชื่อตัวแปรที่ใช้สำหรับเก็บข้อมูลที่ผู้ใช้ใส่เข้ามา ดังนั้นเราจะต้องกำหนดค่าให้กับแอดทริบิวต์นี้ทุกครั้ง เช่น <input name="age" type="text" maxlength="32"/> จากตัวอย่าง ข้อมูลที่รับมาจะถูกเก็บไว้ในตัวแปรชื่อ age

- แอดทริบิวต์ value ใช้สำหรับกำหนดค่าเริ่มต้นหรือค่าอัตโนมัติให้กับตัวแปรที่เราระบุในแอดทริบิวต์ name และเป็นค่าที่จะแสดงหน้าจอครั้งแรก เมื่อผู้ใช้ได้เข้ามาในหน้านี้

- แอดทริบิวต์ type ใช้กำหนดการแสดงผลตัวอักษรในช่องรับข้อมูล เราสามารถกำหนดค่าได้สองค่า คือ

• text หรือ password

text หมายถึง กำหนดให้การแสดงผลตัวอักษรให้เป็นไปตามปกติ

password หมายถึง กำหนดให้การแสดงผลตัวอักษรไม่ตรงตามที่ผู้ใช้พิมพ์เข้ามา โดยทั่วไปแล้วตัวอักษรที่บราวเซอร์แสดงคือ เครื่องหมายดอกจัน

- แอดทริบิวต์ format ใช้กำหนดรูปแบบในการรับข้อมูล

- แอดทริบิวต์ emptyok เอาไว้สำหรับอนุญาตให้ผู้ใช้ใส่ค่าหรือไม่ใส่ค่าก็ได้ ค่าที่เราสามารถกำหนดให้แอดทริบิวต์นี้คือ true หรือ false

true หมายความว่า ผู้ใช้ไม่ต้องใส่ข้อมูลก็ได้

false หมายความว่า ผู้ใช้จะต้องใส่ข้อมูล

- แท็ก select และ option เป็นแท็กที่ต้องใช้ร่วมกัน ทำหน้าที่แสดงตัวเลือกในลักษณะเมนู การจัดวางของแท็กทั้งสองจะเป็นดังนี้

```
<select>
```

```
<option > ตัวเลือกที่ 1 </option>
```

```
<option > ตัวเลือกที่ 2 </option>
```

```
<option > ตัวเลือกที่ 3 </option>
```

```
...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

</select>

ต่อไปเราจะมาดูกันว่าแต่ละแท็กมีแอตทริบิวต์ที่น่าสนใจอะไรบ้าง โดยจะเริ่มจากแท็ก select และตามด้วยแท็ก option

แอตทริบิวต์ของแท็ก <select>

- แอตทริบิวต์ multiple เอาไว้สำหรับกำหนดจำนวนตัวเลือกที่สามารถเลือกได้ โดยค่าที่เราสามารถกำหนดได้คือ true หรือ false

true หมายความว่า ผู้ใช้สามารถเลือกได้หลายตัวเลือก

false หมายความว่า ผู้ใช้สามารถเลือกได้เพียงตัวเลือกเดียว

- แอตทริบิวต์ name ใช้สำหรับตั้งชื่อตัวแปรที่จะรับค่าตัวเลือกที่ผู้ใช้เลือก เช่น ถ้าเรากำหนด name = "say" แล้วผู้ใช้เลือกที่มีค่าเป็น "hello" เราจะได้ตัวแปร say มีค่าเป็น "hello"

- แอตทริบิวต์ value ใช้สำหรับกำหนดค่าเริ่มต้นให้กับตัวแปรที่เราตั้งไว้ในแอตทริบิวต์ name ในกรณีเรากำหนดการเลือกให้สามารถเลือกได้หลายตัวเลือก ค่าแต่ละค่าต้องขึ้นต้นด้วยเครื่องหมายเซมิโคลอน (;) เช่น value = "a;b;c"

แอตทริบิวต์ของแท็ก option

- แอตทริบิวต์ value ใช้กำหนดค่าของตัวเลือก ซึ่งค่านี้จะเป็นค่าที่ตัวแปรในแอตทริบิวต์ name ของแท็ก select รับไปเมื่อผู้ใช้เลือกตัวเลือกนี้ เช่น ถ้าในแท็ก select เรากำหนด name = "year" และในแท็ก option เรากำหนด value = "2543" ถ้าหากผู้ใช้เลือกตัวเลือกนี้ ตัวแปร year จะมีค่าเป็น 2543

- แอตทริบิวต์ title ใช้ตั้งชื่อให้กับแท็ก

5.8 การส่งค่าตัวแปร

จากที่ผ่านมามาเราสามารถรับข้อมูลจากผู้ใช้ได้แล้ว ต่อไปเป็นการนำข้อมูลเหล่านั้นไปใช้ ซึ่งโดยส่วนใหญ่แล้วข้อมูลเหล่านี้จะถูกส่งไปยังไฟล์ที่ทำหน้าที่เป็นโปรแกรมจัดการข้อมูล ซึ่งอาจจะเป็น ASP CGI PHP ฯลฯ ในการส่งข้อมูลสามารถทำได้ทั้งแบบ GET และ POST

5.8.1 การส่งข้อมูลแบบ GET

การส่งข้อมูลแบบ GET สามารถทำได้สองวิธี คือ ส่งไปในแอตทริบิวต์ href ของแท็ก go หรือส่งโดยใช้แท็ก postfield การส่งข้อมูลไปในแอตทริบิวต์ href มีรูปแบบดังนี้

<go href="ชื่อไฟล์จัดการข้อมูล?ชื่อตัวแปร1=ค่าของตัวแปร1&ชื่อตัวแปร2=ค่าของตัวแปร2&..."> เช่น

<go href="do Login.asp?name=Jane&password=ffgrjo"> เราจะเห็นว่าตัวแปรแรกจะต้องต่อจากเครื่องหมายคำถาม (?) และตัวแปรต่อ ๆ มาจะต้องขึ้นต้นด้วยเครื่องหมายและ (&)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการระบุค่าตัวแปรในเอกสาร WML มีรูปแบบดังนี้ S (ชื่อตัวแปร) และเครื่องหมายและ (&) ก็เป็นอักษรพิเศษซึ่งจะต้องใช้กลุ่มอักษรแทน คือ & ดังนั้นรูปแบบการส่งข้อมูลจึงเป็นดังนี้

```
<go href="ชื่อไฟล์จัดการข้อมูล?ชื่อตัวแปร1=S(ชื่อตัวแปร1)&amp; ชื่อตัวแปร2=S(ชื่อตัวแปร2)&amp;...">
```

เช่น

```
<go href="doLogin.asp?name=S(name)&amp;password=ffgrio"/>
```

การส่งข้อมูลโดยใช้แท็ก postfield การใช้แท็ก postfield ส่งข้อมูลแบบ GET มีรูปแบบดังนี้

```
<go href="ชื่อไฟล์จัดการข้อมูล" method="get">
```

```
<postfield name="ชื่อตัวแปร" value="ค่าตัวแปร"/>
```

```
</go>
```

เราจะเห็นว่าการระบุวิธีการส่งนั้นทำในแท็ก go ด้วยแอตทริบิวต์ method ส่วนแท็ก postfield นั้นเป็นที่ระบุชื่อตัวแปรและค่าของมัน

5.8.2 การส่งข้อมูลแบบ POST

การส่งข้อมูลแบบ POST ทำได้โดยแท็ก postfield และ go เช่นเดียวกับการส่งแบบ GET แต่เปลี่ยนค่าแอตทริบิวต์ method ของแท็ก go เป็น post ดังนี้

```
<go href="ชื่อไฟล์จัดการข้อมูล" method="post">
```

```
<postfield.../>
```

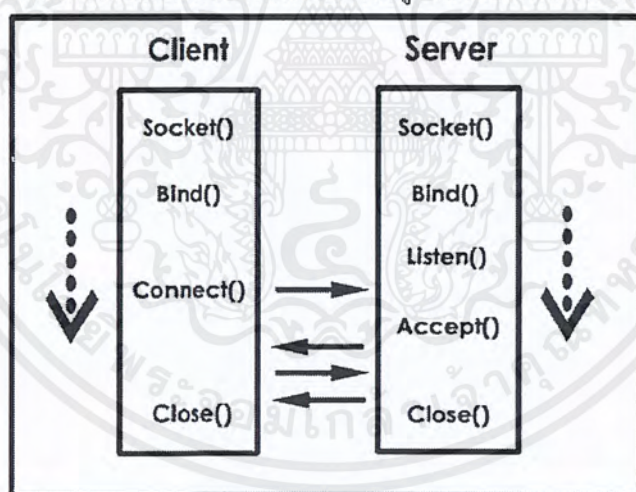
```
...
```

```
</go>
```

บทที่ 6 การเขียนโปรแกรมส่งข้อความเข้าเพจเจอร์ หรือมือถือ

จากที่กล่าวมาในเรื่องของ HTTP โพรโทคอล ซึ่งเป็นโพรโทคอลที่ใช้สำหรับติดต่อสื่อสาร และส่งข้อมูลระหว่างฝั่งไคลเอนต์ เช่นเว็บเบราว์เซอร์ (Web Browser) กับฝั่งเซิร์ฟเวอร์ เช่น เว็บเซิร์ฟเวอร์ (Web Server) โดยข้อมูลต่าง ๆ เหล่านี้อาจจะเป็นไฟล์ เช่น ไฟล์ HTML, ไฟล์รูปภาพ ฯ

ในการเขียนโปรแกรมเครือข่าย (Network Programming) อาจเป็นเรื่องที่ซับซ้อน และยาก แต่มีหลักการพื้นฐานที่ว่า การรับส่งข้อมูลผ่านเครือข่ายจะใช้สิ่งที่เรียกว่า ซ็อกเก็ต (Socket) ซึ่งทั้งผู้รับ และผู้ส่งจะต้องสร้างซ็อกเก็ตขึ้นมา ฝ่ายใดที่ต้องการขอข้อมูลจะต้องสั่งให้ซ็อกเก็ตของตนไปเชื่อมต่อกับซ็อกเก็ตของผู้รับ เมื่อซ็อกเก็ตของผู้รับยอมให้มีการเชื่อมต่อได้ ทั้ง 2 ฝ่ายจึงเริ่มรับส่งข้อมูลระหว่างกันได้คดขยทำงานผ่านซ็อกเก็ตตลอด ถ้าหากมองในด้าน โปรแกรมมิ่ง หลังจากที่โปรแกรมสร้างซ็อกเก็ตขึ้นมาได้แล้ว มี 2 กรณีที่เราสามารถเลือกสั่งให้ซ็อกเก็ตทำ กรณีแรกคือ ถ้าเราเขียนโปรแกรมให้เป็นเซิร์ฟเวอร์ เราจะต้องสั่งให้ซ็อกเก็ตรอการร้องขอติดต่อจากซ็อกเก็ตอื่น อีกกรณีคือ ถ้าเราเขียนโปรแกรมให้เป็นไคลเอนต์ เราจะต้องสั่งให้ซ็อกเก็ตที่เราสร้างขึ้น ทำการเชื่อมต่อกับเซิร์ฟเวอร์เป้าหมายที่ต้องการ เมื่อซ็อกเก็ตที่ปลายทางยอมรับการเชื่อมต่อแล้วเราจึงจะสามารถสั่งให้ซ็อกเก็ตของเราขอข้อมูลจากเซิร์ฟเวอร์ได้ เพื่อให้ง่ายต่อความเข้าใจเรื่องซ็อกเก็ตไม่ว่าซ็อกเก็ตนั้นจะทำหน้าที่เป็นไคลเอนต์ หรือเซิร์ฟเวอร์ก็ตาม จะขอเปรียบเทียบการทำงานของซ็อกเก็ตกับโทรศัพท์ที่เราใช้อยู่ในชีวิตประจำวันดังนี้



รูปที่ 6.1 แสดงการเชื่อมต่อของไคลเอนต์กับเซิร์ฟเวอร์

1. สร้างซ็อกเก็ต เปรียบเสมือนตัวเครื่องโทรศัพท์ในการติดต่อกับปลายทาง
2. ทำการ Bind() เป็นการกำหนดแอดเดรสให้กับซ็อกเก็ต เปรียบเสมือนการเสียบสายโทรศัพท์เข้ากับตัวเครื่อง
3. ทำการกำหนดแอดเดรสของซ็อกเก็ตปลายทางที่จะเชื่อมต่อ แล้วจึงเชื่อมซ็อกเก็ตในเครื่อง ไปยังเครื่องเซิร์ฟเวอร์ปลายทาง ด้วยฟังก์ชัน Connect() หรือเปรียบได้กับ การกำหนดเบอร์โทรศัพท์ปลายทางที่เราจะติดต่อด้วย แล้วทำการติดต่อด้วยโทรศัพท์ที่เราเตรียมไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หลังจากที่ทำการศึกษาติดต่อกับเครื่องเซิร์ฟเวอร์ได้แล้ว จึงทำการส่งข้อมูลให้กับเครื่องเซิร์ฟเวอร์โดยผ่านซ็อกเก็ตได้เรื่อยๆ จนกว่าฝ่ายใดฝ่ายหนึ่งจะบอกตัดการเชื่อมต่อ เหมือนกับการพูดคุยโทรศัพท์ ซึ่งเราสามารถคุยกันได้นานจนกระทั่งฝ่ายใดฝ่ายหนึ่งวางหูโทรศัพท์ การเชื่อมต่อระหว่าง 2 ฝ่ายจึงจบลง

ในการเขียนโปรแกรมสร้างซ็อกเก็ตเราจะต้องระบุได้ว่าซ็อกเก็ตมาจากเครื่องไหน และทำงานกับพอร์ตไหน ซ็อกเก็ตแต่ละตัวจะต้องมีแอดเดรสประจำตัวเพื่อแจ้งกับซ็อกเก็ตที่กำลังติดต่อกับ ข้อมูลที่รับส่งจึงจะไปถึงได้อย่างถูกต้อง การหาแอดเดรสของซ็อกเก็ตจะหาได้จาก แอดเดรสของเครื่องที่เกิดจากการนำหมายเลข IP Address มาแปลงรูป หมายเลขพอร์ตที่ซ็อกเก็ตจะใช้ และข้อมูลอื่นๆอีกบางส่วนมาผสมกันให้ได้แอดเดรสของซ็อกเก็ต สมมติว่าเราเขียนโปรแกรมเป็นโคลเอนต์รันอยู่ที่เครื่องที่มี IP แอดเดรสเป็น 161.246.5.231 เพื่อให้ติดต่อไปยังเซิร์ฟเวอร์ที่มี IP แอดเดรสเป็น 161.246.5.233 โปรแกรมของเราต้องทำดังนี้

1. เอา IP แอดเดรสเครื่องของเราคือ 161.246.5.231 ไปหาแอดเดรสของเครื่อง
2. นำแอดเดรสของเครื่องไปหาแอดเดรสของซ็อกเก็ต
3. สร้างซ็อกเก็ตขึ้นมา
4. นำแอดเดรสของซ็อกเก็ตที่หาได้ไปกำหนดให้ซ็อกเก็ต

หลังจากนี้ซ็อกเก็ตที่สร้างขึ้นจะมีแอดเดรสของตัวเองที่ใช้อ้างอิงเวลาติดต่อกับซ็อกเก็ตอื่นแล้ว ขั้นตอนต่อไปซ็อกเก็ตที่สร้างขึ้นต้องเชื่อมต่อไปยังซ็อกเก็ตที่เซิร์ฟเวอร์ซึ่งมี IP แอดเดรส 161.246.5.233 นั้น หมายความว่าโปรแกรมต้องการหาแอดเดรสของซ็อกเก็ตในเครื่องเซิร์ฟเวอร์ให้ได้เสียก่อนจึงจะติดต่อไปได้ เมื่อซ็อกเก็ตในเครื่องของเรามีแอดเดรสที่แน่นอน และแอดเดรสของซ็อกเก็ตที่รออยู่ในเซิร์ฟเวอร์แล้ว โปรแกรมก็จะสั่งให้ซ็อกเก็ตในเครื่องร้องขอเชื่อมต่อกับซ็อกเก็ตในเซิร์ฟเวอร์ได้

หลังจากเตรียมความพร้อม และความเข้าใจเรื่องโพรโทคอลมาแล้ว มาดูก่อนว่าบริการส่งข้อความเข้าเพจเจอร์ผ่านอินเทอร์เน็ตของผู้ให้บริการแต่ละรายในปัจจุบันมีขั้นตอนอย่างไรบ้าง

ผู้ให้บริการเพจเจอร์จะสร้างเว็บเพจ ที่มีฟอร์มสำหรับให้ผู้ใช้มากรอกข้อความที่ต้องการส่งเข้าเพจเจอร์ เมื่อผู้ใช้คลิกปุ่มส่งข้อมูล ข้อมูลจากฟอร์มในเว็บเพจจะส่งไปให้ Server Side Script เช่น CGI, ASP เป็นต้น ในเครื่องเซิร์ฟเวอร์ของผู้ให้บริการเพื่อจะส่งข้อความนั้นไปให้กับอุปกรณ์ที่เป็นฮาร์ดแวร์สำหรับส่งข้อความเข้าเครื่องเพจเจอร์ หรือมือถือของผู้รับอีกต่อหนึ่ง เนื่องจากทางเซิร์ฟเวอร์ผู้ให้บริการได้ใช้ CGI เป็นตัวรับข้อมูลจากผู้ที่ใช้ส่งข้อความผ่านอินเทอร์เน็ต ดังนั้นจึงมีหลายวิธีที่เราจะส่งข้อความเข้าเพจเจอร์ได้ โดยไม่ต้องไปส่งข้อความที่เว็บเพจของผู้ให้บริการ วิธีง่ายๆ คือแกะอินพุตจากฟอร์มในเว็บเพจของผู้ให้บริการ แล้วนำมาสร้างเป็นโปรแกรมสำหรับส่งข้อความเข้าเพจเจอร์ หรือมือถือ แต่ไม่ว่าจะส่งข้อความเข้าเพจเจอร์ด้วยวิธีใดก็ตาม ข้อความนั้นก็ต้องผ่าน CGI ในเซิร์ฟเวอร์ผู้ให้บริการอยู่ดี

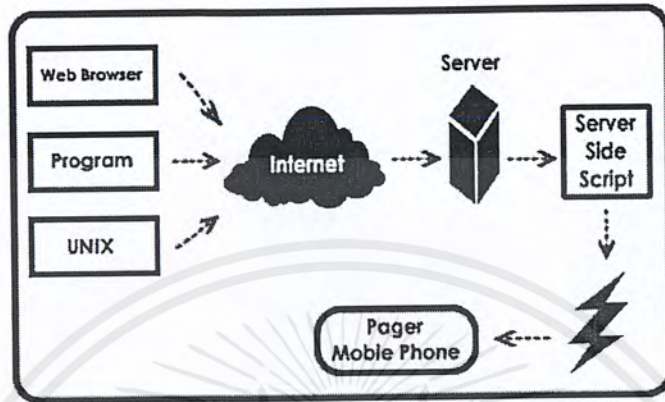
วิธีการส่งข้อความเข้าเพจเจอร์ หรือมือถือ

วิธีในการส่งข้อความเข้าเพจเจอร์ และมือถือโดยปราศจากการส่งผ่านเว็บเบราว์เซอร์ได้หลัก ๆ 2 วิธี

6.1 การส่งโดยผ่านโปรแกรม Telnet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Telnet สามารถทำมาประยุกต์ใช้กับงานอินเทอร์เน็ตได้หลายอย่าง ขึ้นอยู่กับว่าผู้ที่มีความเข้าใจเรื่องของโพรโทคอลมากน้อยเพียงไร การใช้โปรแกรม Telnet เพื่อส่งข้อความเข้าเพจเจอร์ หรือ มือถือสามารถทำได้โดยป้อนคำสั่งทาง Comand Line เพื่อเชื่อมต่อไปยังพอร์ต 80 ของเซิร์ฟเวอร์ผู้ให้บริการเพจเจอร์



รูปที่ 6.2 แสดงวิธีการทำงานในการส่งข้อความเข้าเพจเจอร์ หรือมือถือ

จากนั้นก็ส่งข้อความ โดยเลียนแบบข้อความร้องขอทุกอย่างที่ผู้ให้บริการเพจเจอร์ และมือถือกำหนดไว้ เซิร์ฟเวอร์เป้าหมายที่จะทดลองคือ <http://www.ais900.com> อันเป็นเซิร์ฟเวอร์ที่ส่งข้อความเข้าโทรศัพท์มือถือระบบ GSM ซึ่งการทำงานก็คล้ายๆ กับเพจเจอร์ เพียงแต่เปลี่ยนตัวรับจากเพจเจอร์เป็นมือถือเท่านั้น

ขั้นแรกเราต้องทราบว่าจะส่งให้ CGI ในเซิร์ฟเวอร์มีอะไรบ้าง โดยเปิดดูจากซอร์ซโค้ด HTML ของเว็บเพจที่มีฟอร์มสำหรับส่งข้อความซึ่งอยู่ที่ <http://www.ais900/gsmweb> ดังรูป

ข้อมูลต่อไปนี้ ได้มาจากการแกะอินพุตในฟอร์มของเว็บเพจ

- ชื่อของเซิร์ฟเวอร์คือ www.ais900.com
- URI ของ CGI ในเซิร์ฟเวอร์คือ `/cgi-gsmweb/web2gsm.cgi` รับข้อมูลแบบ POST
- ชื่อของตัวแปรในฟอร์ม
- `szGsmno` คือ หมายเลขโทรศัพท์มือถือ
- `szGsmPwd` คือ รหัสผ่านในการส่งข้อความ
- `szMesg` คือ ข้อความที่จะส่งเข้าโทรศัพท์มือถือ

สมมติว่าเบอร์โทรศัพท์มือถือเป้าหมายคือ 8381353 รหัสผ่านในการส่งข้อความไม่มี และข้อความที่ต้องการส่งคือ “ส่งจาก Telnet” เมื่อนำข้อมูลมาเขียนต่อกันตามรูปแบบส่งข้อมูลด้วยเมธอด POST จะได้เป็น “`szGsmno=8381383&szGsmPwd=&szMseg=ส่งจาก%20Telnet`” ความยาวทั้งหมดของข้อความที่จะส่งคือ 51 ตัวอักษร ขั้นตอนต่อไปคือการรันโปรแกรม ป้อนคำสั่งทางคอมมานด์ไลน์เป็น Telnet www.ais900.com 80 แล้วป้อนข้อความต่อไปนี้ลงในวินโดว์ Telnet

```
POST /cgi-gsmweb/web2gsm.cgi HTTP/1.0
```

```
Accept: */*
```

```
Content - type: application / x - www - form - urlencoded
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Content – length : 51

<Enter>

szGsmno=8381383&szGsmPwd=&szMseg=ส่งจาก%20Telnet

หลังจากพิมพ์บรรทัดที่เขียนว่า Content – Type : 51 แล้วต้องกดคีย์ Enter เพื่อเว้นบรรทัดว่าง 1 บรรทัด อันเป็นสัญญาณบอกว่าหมดรายการเซคเตอร์แล้ว ตามโครงสร้างโปรโตคอล HTTP จากนั้นจึงพิมพ์ข้อความที่จะส่ง เมื่อพิมพ์ข้อความบรรทัดสุดท้ายแล้วต้องกดคีย์ Enter อีกครั้ง ก็เป็นอันว่าเสร็จสิ้นการสร้างข้อความร้องขอไปยังเซิร์ฟเวอร์ หลังจากรอสักครู่เซิร์ฟเวอร์จะส่งผลตอบกลับมาเอง

6.2 เขียนโปรแกรมส่งข้อความเข้าเพจเจอร์ และมีถือ

โดยใช้หลักการในการเขียนโปรแกรมเพื่อติดต่อสื่อสารผ่านอินเทอร์เน็ต เราจะเขียนโปรแกรมให้เป็นไคลเอนต์ ทำการเชื่อมต่อไปยังซ็อกเก็ตของเครื่องเซิร์ฟเวอร์เป้าหมายที่ต้องการเพื่อทำการติดต่อสื่อสาร การเขียนโปรแกรมเพื่อติดต่อสื่อสารประกอบไปด้วยขั้นตอนดังนี้

- สร้างซ็อกเก็ต (Socket)
- ทำการ Bind() เป็นการกำหนดแอดเดรสให้กับซ็อกเก็ต
- ทำการกำหนดแอดเดรสของซ็อกเก็ต (Socket) ปลายทางที่จะเชื่อมต่อ แล้วจึงเชื่อมต่อซ็อกเก็ตในเครื่อง

ไปยังเครื่องเซิร์ฟเวอร์ปลายทาง

- หลังจากทำการติดต่อกับเครื่องเซิร์ฟเวอร์ได้แล้ว จึงทำการส่งข้อมูลให้กับเครื่องเซิร์ฟเวอร์โดยผ่าน

ซ็อกเก็ตได้เรื่อยๆ จนกว่าฝ่ายใดฝ่ายหนึ่งจะบอกตัดการเชื่อมต่อ

ในการเขียนโปรแกรมนั้นสามารถเลือกภาษาใดมาเขียนก็ได้เช่น PERL , C , VB , PHP , JAVA เป็นต้น โดยหลักการเขียนจะเป็นดังที่ได้อธิบายข้างต้น จะขออธิบายวิธีเขียนส่งเพจเจอร์โปรแกรมภาษา PERL และภาษาจาวา (Java)

การเขียนโปรแกรมส่งเพจเจอร์ด้วยภาษา PERL

ภาษา PERL (Practical Extraction And Report Language) ซึ่งถูกพัฒนาให้สามารถใช้ในระบบปฏิบัติการ UNIX เป็นครั้งแรก และต่อมาก็มีการพัฒนาให้สามารถใช้ในระบบปฏิบัติการอื่นๆ ได้เช่น Window NT เป็นต้น ภาษา PERL เป็นภาษาที่มีความคล้ายคลึงกับภาษาซี เพราะบางคำสั่งก็มีหน้าที่เหมือนกับในภาษาซีนั่นเอง ในการเขียนโปรแกรมภาษา PERL ให้รันบนยูนิกซ์ (UNIX) นั้นเราจะต้องตรวจสอบว่าในระบบยูนิกซ์ (UNIX) ที่เราใช้อยู่มีการติดตั้ง PERL หรือไม่ และติดตั้งไว้ที่ไหนบนยูนิกซ์ (UNIX) โดยมาก PERL จะถูกเก็บในไดเรกทอรี /user/bin/perl เราสามารถใช้คำสั่ง WHICH เพื่อค้นหา PATH ไดเรกทอรีของ PERL ในระบบยูนิกซ์ (UNIX) ได้

นอกจากนั้นไฟล์บนระบบยูนิกซ์ (UNIX) จะแตกต่างจาก DOS หรือ Window ตรงที่มันจะไม่นิยามสกุลในการแบ่งไฟล์ แต่จะใช้โหมด (MODE) หรือ (Attribute) แบ่งประเภทของไฟล์ ไฟล์บนระบบยูนิกซ์ (UNIX) จะสามารถทำงานได้ มันจะต้องอยู่ในโหมด (MODE) Execute หรือ 755 ซึ่งเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราเปลี่ยนรูปแบบของไฟล์ให้เป็น UNIX Format แล้ว เราจะต้องเปลี่ยนโหมดของไฟล์ให้เป็น 755 ด้วยก็ เพื่อที่จะให้โปรแกรมทำงานได้

ในการเขียนโค้ด (Code) เพื่อทำงานเป็นไคลเอนต์ติดต่อไปยังเซิร์ฟเวอร์ผู้ให้บริการเพจเจอร์ และมีมือถือทั้งหลายนั้นจะเริ่มจาก

1. #!/user/local/bin/perl

เป็นการบอกที่อยู่ของ PERL ในระบบยูนิกซ์

2. use Socket;

เป็นการใช้คำสั่ง use Socket หมายถึงเรียกเอาโมดูล Socket ของ PERL มาใช้งานในโปรแกรมด้วย

3. \$protocol = (getprotocalbyname('tcp'))[2];

เป็นการหาหมายเลขโปรโตคอล TCP จากฟังก์ชัน getprotocalbyname() เพื่อใช้ในการสร้างซ็อกเก็ต

4. \$target_host="www.paclink.co.th";

\$port="80";

\$uri = "/phtml/send.phtml";

\$number = "230184";

\$referer = "http://www.paclink.co.th/phtml/send.phtml";

\$host="www.paclink.co.th";

\$message = "HI I THIS IS MY PROJECT";

\$day="-1";

\$hour="0";

\$min="00";

\$count=length(\$message);

\$msgSend = pager_number=\$number&message=\$message&counter=\$count&wday=\$day&hour
=\$hour&minute=\$min&flag=send+flag+ja&x=49&y=12 ";

\$msgSend =~ s/ /%20/g;

\$lenMsg = length(\$msgSend);

เป็นการกำหนดตัวแปรต่างที่ใช้ในการเขียนโปรแกรม และที่ใช้ในการส่งข้อมูลไปให้กับเซิร์ฟเวอร์
ปลายทาง

5. \$My_Socket_Address = pack("S n a4 x8", AF_INET, 0, "\0\0\0\0");

\$Target_Com_Address = (gethostbyname(\$target_host))[4];

\$Target_Sock_Address = pack("S n a4 x8", AF_INET,\$port, \$target_com_addr);

เป็นการหาแอดเดรสให้กับซ็อกเก็ตของเรา และซ็อกเก็ตของเครื่องซ็อกเก็ตปลายทาง

6. socket (MY_SOCKET, AF_INET, SOCKET_STREAM , \$protocol)

เป็นฟังก์ชันในการการซ็อกเก็ต ซึ่งมีรูปแบบดังนี้ SOCKET(name,family,type,protocol)

name ชื่อของซ็อกเก็ต (ชื่อที่เราตั้งเอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

family ประเภทของซ็อกเก็ต AF_NET หรือ AF_UNIX
 type รูปแบบการรับส่งข้อมูล SOCKET_STREAM หรือ SOCK_DGRAM
 protocol หมายเลขโปรโตคอลที่จะใช้กับซ็อกเก็ต

7. bind(MY_SOCKET, \$MY_Socket_Address)

เป็นการนำแอดเดรสของซ็อกเก็ตที่กำหนดให้ไปกำหนดให้กับซ็อกเก็ตจริงๆ การกำหนดแอดเดรสให้ซ็อกเก็ตจะใช้ฟังก์ชัน bind() สิ่งที่จะต้องส่งเป็นพารามิเตอร์ให้ฟังก์ชัน bind() คือ ชื่อของซ็อกเก็ต และแอดเดรสของแอดเดรสของซ็อกเก็ต

8. connect(MY_SOCKET, Target_Socket_Address)

เป็นการเชื่อมโยงซ็อกเก็ตในเครื่องไปยังซ็อกเก็ตอื่น โดยใช้ฟังก์ชัน connect() สิ่งที่จะต้องบอกเป็นพารามิเตอร์คือชื่อของซ็อกเก็ตในเครื่องของเรา และแอดเดรสของซ็อกเก็ตเป้าหมายที่ต้องการ

9. print MY_SOCKET "POST Suri HTTP/1.0\n";

```
print MY_SOCKET "Referer: Sreferer\n";
print MY_SOCKET "Accept: */*\n";
print MY_SOCKET "User-Agent: Mozilla/4.6 [en] (Win95; I)\n";
print MY_SOCKET "Content-type: application/x-www-form-urlencoded\n";
print MY_SOCKET "Content-length: $lenMsg\n";
print MY_SOCKET "\n";
print MY_SOCKET "SmsgSend";
```

เป็นการส่งข้อความให้กับซ็อกเก็ตด้วยคำสั่ง print บรรทัดแรกเป็นการบอกส่งข้อความร้องขอด้วยเมธอด POST ตามด้วยเฮดเดอร์อีก 2-3 รายการ รายการที่ขาดไม่ได้คือ Content-Type และ Content-length

10. close \$MY_SOCKET;

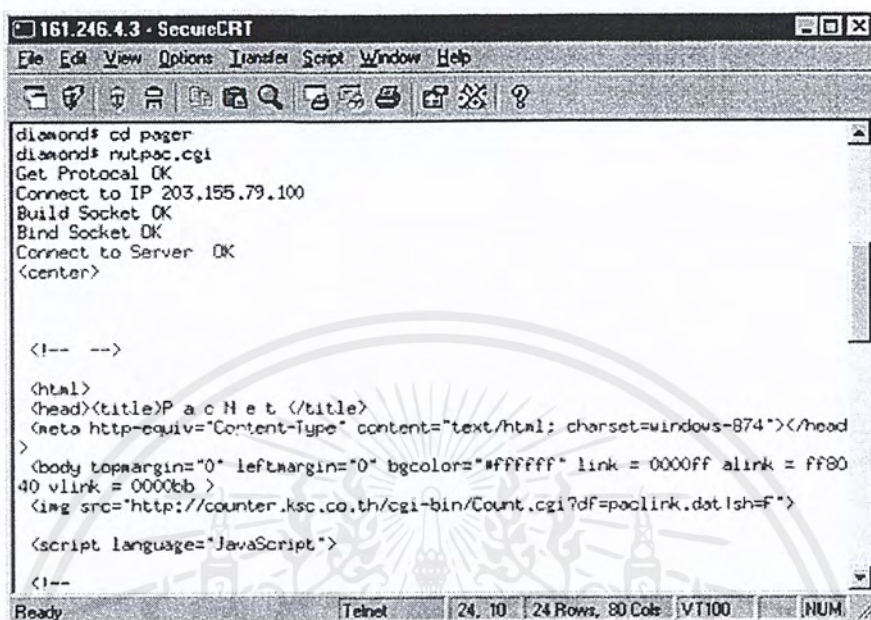
เป็นการปิดการทำงานของซ็อกเก็ตที่สร้างขึ้นด้วยฟังก์ชัน close()

ตัวอย่างการเขียนโปรแกรมภาษา PERL ส่งข้อความเข้าเพจเจอร์ หรือมือถือสามารถดูได้ดังรูปที่ 6.2 การเขียนโปรแกรมส่งข้อความเข้าเพจเจอร์ หรือมือถือด้วยภาษาจาวา

ในการเขียนโปรแกรมเพื่อติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตด้วยภาษาจาวานั้น เราสามารถใช้คราสต่างที่มีอยู่ในแพ็คเกจ java.net เพื่อใช้ในการติดต่อระหว่างไคลเอนต์ กับเซิร์ฟเวอร์ได้โดยตัวจาวานั้นจะใช้ โปรโตคอล TCP หรือ UDP ในการติดต่อสื่อสาร ซึ่ง TCP นั้นย่อมาจาก Transmission Control Protocol เป็น Connection-Based Protocol ที่ гаранตีว่าข้อมูลที่ส่งจากผู้ส่ง นั้นจะถึงผู้รับอย่างแน่นอน หลากๆกับการโทรศัพท์ ส่วน UDP ย่อมาจาก User Datagram Protocol เป็นโปรโตคอลที่ไม่สามารถ гаранตีได้ว่าข้อมูลที่ส่งจะสามารถถึงผู้รับหรือไม่ คล้ายกับการส่งจดหมายกับไปรษณีย์ ตัวอย่างของคราสต่างๆที่อยู่ในแพ็คเกจ java.net ที่ใช้โปรโตคอล TCP ในการสื่อสารนั้นได้แก่ URL , URLConnection Socket และ ServerSocket ส่วนคราสที่ใช้ โปรโตคอล UDP ในการสื่อสารนั้นได้แก่ DatagramPacket , DatagramSocket , MulticastSocket

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมจาส่งข้อความเข้าเพจเจอร์ หรือมือถือนั้นสามารถทำได้หลายวิธี แต่ขออธิบายแค่ 2 วิธีหลักคือ ใช้คลาส URLConnection กับเขียนโปรแกรมสร้างสร้างช็อกเก็ต



```

161.246.4.3 - SecureCRT
File Edit View Options Transfer Script Window Help
diamond# cd pager
diamond# nntpoc.cgi
Get Protocol OK
Connect to IP 203.155.79.100
Build Socket OK
Bind Socket OK
Connect to Server OK
<center>

<!-- -->

<html>
<head><title>P a c H e t </title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-874"></head>
<body topmargin="0" leftmargin="0" bgcolor="#ffffff" link = 0000ff alink = ff80
40 vlink = 0000bb >

<script language="JavaScript">
<!--
Ready Telnet 24. 10 24 Rows, 80 Cols VT100 NUM

```

รูปที่ 6.3 แสดงการส่งข้อความเข้าเพจเจอร์ 1144 โดยใช้เขียนโปรแกรม PERL

URLConnection คราสนี้มีเมธอดหลายเมธอดให้เลือกใช้ในการสื่อสารด้วย URL การเขียนโปรแกรมเพื่อส่งข้อมูลให้กับโปรแกรมที่อยู่ในเซิร์ฟเวอร์ของผู้ให้บริการเพจเจอร์ หรือมือถือนั้นจะประกอบไปด้วยขั้นตอนคร่าวๆดังนี้

1. ทำการสร้างออบเจกต์ URL โดยใช้คำสั่ง URL

```
url = new URL ( ชื่อของเซิร์ฟเวอร์ผู้ให้บริการเพจเจอร์ หรือมือถือ เช่น http://www.ais900.com );
```

2. ทำการเปิดติดต่อกับเซิร์ฟเวอร์โดยใช้คำสั่ง

```
URLConnection urlConn = url.openConnection();
```

3. เซ็ตค่าบูลีนสำหรับเอาต์พุตของ UrlConnection โดยใช้คำสั่ง

```
urlConn.setDoOutput (true);
```

4. สร้างเอาต์พุตสตรีม (Output Stream) และ PrintWriter สำหรับการติดต่อกับเซิร์ฟเวอร์โดยคำสั่ง

```
PrintWriter printout = new PrintWriter(urlConn.getOutputStream ());
```

5. ส่งข้อมูลที่ต้องการส่งโดยผ่านออบเจกต์ PrintWriter ที่ได้สร้างไว้โดยใช้คำสั่ง

```
printout.println(content);
```

6. ปิดเอาต์พุตสตรีม (Output Stream) ที่สร้างสร้างขึ้นมาโดยใช้คำสั่ง

```
printout.close ();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียน โปรแกรมจาวาสคริปต์เพื่อติดต่อกับเซิร์ฟเวอร์ปลายทางโดยใช้รหัสซ็อกเก็ต (Socket) จากนั้นก็ใช้ซ็อกเก็ตที่สร้างขึ้นรับส่งข้อมูลกับเซิร์ฟเวอร์ ขั้นตอนในการเขียน โปรแกรมประกอบ ด้วย

1. สร้างออบเจกต์ซ็อกเก็ตโดยใช้คำสั่ง

Socket ชื่อของซ็อกเก็ต = new Socket(ชื่อ โฮสของเซิร์ฟเวอร์, หมายเลขพอร์ต);

2. สร้างเอาต์พุตสตรีม ให้กับ PrintWriter เพื่อใช้ติดต่อกับเซิร์ฟเวอร์โดยใช้คำสั่ง

PrintWriter out = new PrintWriter(ชื่อของซ็อกเก็ต.getOutputStream(), true);

3. ทำการส่งข้อมูลให้กับเซิร์ฟเวอร์ผ่านซ็อกเก็ตที่สร้างขึ้น โดยใช้คำสั่ง

```
out.println("POST /cgi-gsmweb/web2gsm.cgi HTTP/1.0");
```

```
out.println("Connection: Keep-Alive");
```

```
out.println("User-Agent: Mozilla/4.6 [en] (Win95; I)");
```

```
out.println("Host: "+host);
```

```
out.println("Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, image/png, */*");
```

```
out.println("Referer:"+Reference);
```

```
out.println("Content-type: application/x-www-form-urlencoded");
```

```
out.println("Content-length:"+MsgLength);
```

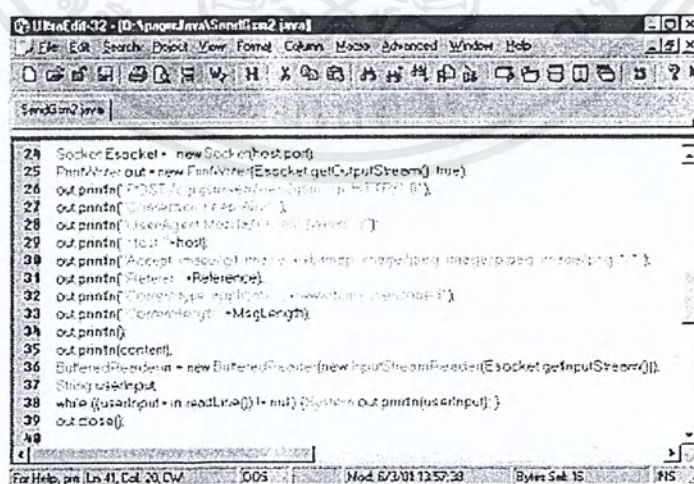
```
out.println();
```

```
out.println(ข้อมูลที่ส่งให้กับเซิร์ฟเวอร์);
```

จากตัวอย่างเป็นการส่งข้อมูลที่เป็นค่าแฮคเกอร์ และข้อมูลต่าง ที่จะใช้ส่งให้กับเซิร์ฟเวอร์

4. ปิดเอาต์พุตสตรีม และปิดซ็อกเก็ตด้วยคำสั่ง

```
out.close(); Esocket.close();
```



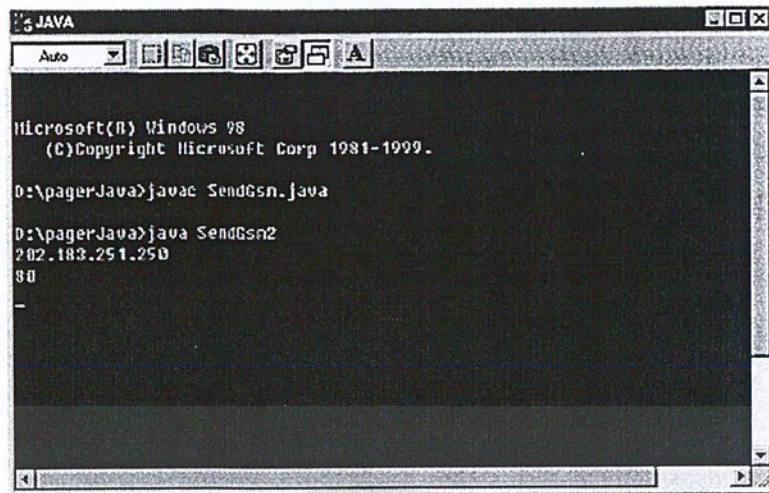
```

1 Socket Esocket = new Socket(host,port);
25 PrintWriter out = new PrintWriter(Esocket.getOutputStream(), true);
26 out.println("POST /cgi-gsmweb/web2gsm.cgi HTTP/1.0");
27 out.println("Connection: Keep-Alive");
28 out.println("User-Agent: Mozilla/4.6 [en] (Win95; I)");
29 out.println("Host: "+host);
30 out.println("Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, image/png, */*");
31 out.println("Referer: "+Reference);
32 out.println("Content-type: application/x-www-form-urlencoded");
33 out.println("Content-length: "+MsgLength);
34 out.println();
35 out.println(content);
36 BufferedReader in = new BufferedReader(new InputStreamReader(Esocket.getInputStream()));
37 String readInput;
38 while ((readInput = in.readLine()) != null) {System.out.println(readInput); }
39 out.close();
40

```

รูปที่ 6.4 แสดงการเขียนโปรแกรมจาวาสคริปต์เพื่อติดต่อกับเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



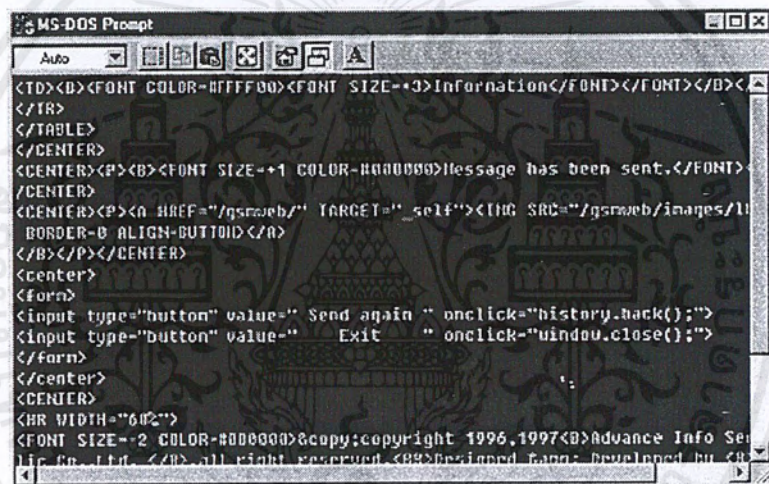
```

D:\pagerJava>javac SendGsm.java

D:\pagerJava>java SendGsm2
202.183.251.250
80

```

รูปที่ 6.5 แสดงการคอมไพล์ และรันโปรแกรมจาวาเพื่อส่งข้อความเข้าเพจเจอร์



```

<TD><D><FONT COLOR=#FFFF00><FONT SIZE=+3>Information</FONT></FONT></D><
</TR>
</TABLE>
</CENTER>
<CENTER><P><B><FONT SIZE=+1 COLOR=#000000>Message has been sent.</FONT>
</CENTER>
<CENTER><P><A HREF="/gsmweb/" TARGET="_self"><IMG SRC="/gsmweb/images/11
BORDER=0 ALIGN=BUTTON</A>
</B></P></CENTER>
<center>
<form>
<input type="button" value=" Send again " onclick="history.back();">
<input type="button" value=" Exit " onclick="window.close();">
</form>
</center>
<CENTER>
<HR WIDTH="60%">
<FONT SIZE=-2 COLOR=#000000>&copy;copyright 1996,1997<B>Advance Info Ser
lic Co., Ltd. </B> all right reserved <B>Assigned Page: Developed by </B>

```

รูปที่ 6.6 แสดงข้อมูลที่เซิร์ฟเวอร์ผู้ให้บริการส่งกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7 การคำนวณ การสร้าง และการออกแบบ

7.1 ขอบเขตการดำเนินงานของโปรแกรมการนัดหมายผ่านเครือข่ายอินเทอร์เน็ต

การทำงานของโปรแกรมการนัดหมายผ่านเครือข่ายอินเทอร์เน็ตคือ เป็นเว็บไซต์ (Web Site) ให้ผู้ใช้เข้ามาสมัครเป็นสมาชิก เพื่อเข้ามาทำการบันทึกเหตุการณ์ตามวันเวลาที่ต้องการ ช่วยในการจดบันทึกเหตุการณ์ที่เกิดขึ้น และช่วยเตือนความจำ โดยมีฟังก์ชันการทำงานหลักๆ คือ ช่วยจดบันทึกรายการสิ่งที่จะต้องทำ (To Do List) , ทำการบันทึกข้อมูลของเหตุการณ์ต่างๆ ตามวัน และเวลาตามที่ผู้ใช้ต้องการ (Add Event) , แสดงเตือนเหตุการณ์ต่างๆที่เกิดขึ้นตามรอบเวลา (Repeat) เช่น วันเกิด วันครบรอบแต่งงาน และ การประชุมนัดหมายทุกสัปดาห์ เป็นต้น , สามารถส่งข้อความเข้าเพจเจอร์ หรือโทรศัพท์มือถือ (Mobile Phone) เพื่อช่วยเรียกเตือนตามที่ผู้ใช้ต้องการ (Reminder) , สามารถสร้างกลุ่มของสมาชิก (Groups) เพื่อความสะดวกการนัดหมายสมาชิกต่างๆ ที่อยู่ภายในกลุ่มที่สร้างขึ้น

นอกจากความสามารถดังกล่าวแล้ว เว็บไซต์ (Web Site) ที่สร้างขึ้นนี้ยังสามารถเรียกดูผ่านโทรศัพท์มือถือ (Mobile Phone) ได้อีกด้วยเพื่อความสะดวกในการใช้งานมากยิ่งขึ้น

7.2 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงานในโครงการประกอบด้วย

- ขั้นแรกทำการศึกษาถึงทฤษฎีต่างๆที่จำเป็นในการทำโครงการ และศึกษาข้อดีและข้อเสียในส่วนต่าง เพื่อที่จะได้เลือกสิ่งที่เหมาะสมกับการพัฒนาโครงการ
- ทำการออกแบบและเขียน UML
- ทำการออกแบบฐานข้อมูลโดยใช้ ER ไดอะแกรม
- ทำการหา คัดตั้งอุปกรณ์ เซต โปรแกรมและอุปกรณ์ต่างๆที่จำเป็นต้องใช้ในการทำโครงการ
 - คัดตั้งโปรแกรมออร์เกิล (Oracle 8I)
 - คัดตั้งโปรแกรมเว็บเซิร์ฟเวอร์ Apache Tomcat
 - คัดตั้งโปรแกรม Nokia Toolkit
 - ทำการ โปรแกรมปลั๊กย่อยที่ใช้ในการทำโครงการ เช่น JDK1.3 เป็นต้น
- ทำการสร้างเว็บเพจ ออกแบบโดยใช้โปรแกรม Dreamweaver 4.0 ก่อนจากนั้นจึงค่อยใช้เขียนโปรแกรมจาวาเซิร์ฟเร็ดสร้างเว็บเพจตามทีออกแบบมาอีกที
- เขียนโปรแกรมจาวาเซิร์ฟเร็ดรับข้อมูลจากสมาชิกโดยผ่านเว็บเพจ และดึงข้อมูลจากดาต้าเบสเพื่อมาแสดงบนเว็บเพจ
- ทำการสร้างเว็บเพจ (WAP Page) โดยใช้ภาษา WML โดยหลักการจะคล้ายกับการสร้างเว็บเพจ
- ทำการเขียนโปรแกรมเพื่อส่งข้อความเข้าเพจเจอร์ ICQ E-Mail และ โทรศัพท์มือถือ
- ทำการทดสอบระบบทั้งหมดเพื่อหาข้อบกพร่อง และทำการปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 ขั้นตอนการกำหนดอุปกรณ์ และโปรแกรมที่ต้องใช้ในโครงการ

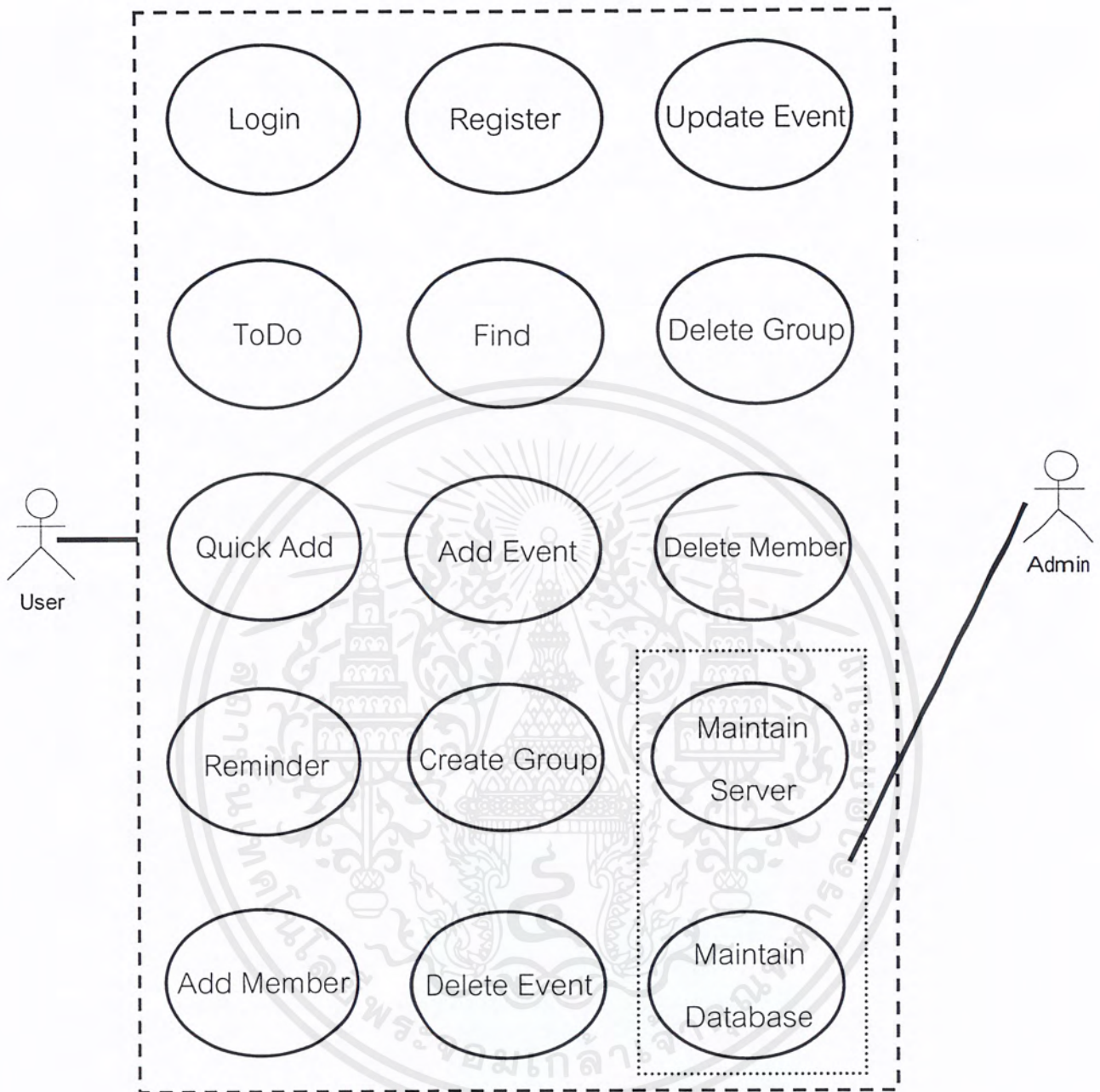
โครงการนี้ใช้ภาษาจาวาเซิร์ฟเฟิร์ต (Java Servlet) เป็นตัวจัดการการทำงานของเว็บไซต์ ซึ่งมีหลักการทำงานคล้ายกับ CGI (Common Gateway Interface) นอกจากนั้นยังใช้จาวาเซิร์ฟเฟิร์ต (Java Servlet) เป็นตัวติดต่อ และจัดการกับฐานข้อมูล ผ่านภาษา SQL เพื่อดึงข้อมูลที่อยู่ฐานข้อมูลมาใช้ และแสดงบนเว็บเพจ (Web Page) ซึ่งโครงการนี้ใช้โปรแกรมอราเคิล (Oracle 8i) เป็นโปรแกรมจัดการฐานข้อมูล และใช้โปรแกรม Apache Tomcat เป็นเว็บเซิร์ฟเวอร์ (Web Server) โดยโปรแกรมที่ใช้งานทั้งหมดนี้ทำงานอยู่บนระบบปฏิบัติการ (Operating system) วินโดวส์ 2000 (Window 2000)

นอกจากนี้โครงการยังต้องทำการสร้าง เว็บ (WAP : Wireless Application Protocol) เพจ เพื่อรองรับการใช้งานผ่านโทรศัพท์มือถือ จึงต้องมีการใช้โปรแกรมที่เรียกว่า Nokia Toolkit ในการสร้างเว็บเพจ โดยใช้ภาษา WML (Wireless Markup Language) ซึ่งมีโครงสร้างหลักคล้ายๆกับ ภาษา HTML (Hyper Text Markup Language) ส่วนตัวเซิร์ฟเวอร์ และโปรแกรมฐานข้อมูลที่ใช้กับเว็บ (WAP) นั้นยังคงใช้โปรแกรม Apache Tomcat เป็นเซิร์ฟเวอร์ และใช้โปรแกรมอราเคิล (Oracle 8i) เป็นฐานข้อมูลเช่นเดียวกับที่ใช้กับเว็บไซต์

7.4 ขั้นตอนการออกแบบ

ในขั้นตอนการออกแบบ ขั้นแรกจะทำการกำหนดฟังก์ชันการทำงานหลักๆ ของโปรแกรมก่อนว่าต้องมีฟังก์ชันอะไรบ้าง ซึ่งจะได้ออกมาเป็น Use Case ไดอะแกรม ดังหัวข้อที่ 7.4.1 จากนั้นก็ทำการวาดคราสไดอะแกรม และออบเจกต์ไดอะแกรม ดังหัวข้อที่ 7.4.2 และ 7.4.3 ตามลำดับ ต่อมาจะทำการวาด Sequence ไดอะแกรม กำหนดความสัมพันธ์ระหว่างคราสต่าง เพื่อตอบสนองต่อฟังก์ชันในการทำงานของระบบ ดังหัวข้อที่ 7.4.4 พอเสร็จขั้นตอนข้างต้น ก็ทำการออกแบบฐานข้อมูลที่จำเป็นในการทำงาน โดยใช้ ER ไดอะแกรม ในการออกแบบ หลังจากนั้นก็ทำการเขียนโปรแกรมตามรูปแบบที่ได้ทำการออกแบบไว้

7.4.1 Use-Case Diagram



รูปที่ 7.1 แสดงภาพ User-Case Diagram

จาก Use Case Diagram จะพบว่ามี Use Case หลัก ๆ อันจะเป็นบริการที่ระบบควรจะมีได้แก่

- Use Case : Login
 Actor : ผู้ใช้
 Description : ใช้สำหรับให้ผู้ใช้ใส่ ID และ Password เพื่อตรวจสอบผู้ที่จะเข้ามาใช้บริการเว็บไซต์และเว็บไซต์เราได้ จะต้องเป็นสมาชิกเท่านั้น
- Use Case : Register
 Actor : ผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Description : ใช้สำหรับให้ผู้ใช้ที่ไม่ได้เป็นสมาชิกเข้ามาสมัครสมาชิกก่อน จึงจะสามารถเข้าใช้บริการได้

Use Case : ToDo

Actor : ผู้ใช้

Description : ใช้สำหรับแสดงสิ่งที่จะต้องทำที่ผู้ใช้ได้บันทึกไว้ ซึ่งสิ่งที่จะต้องทำแต่ละอย่างล้วนมีความสำคัญแต่ต่างกันออกไป ซึ่งจะแสดงเป็นสีและข้อความให้เห็น และให้ผู้ใช้เข้ามาบันทึก แก้ไข และลบสิ่งที่จะต้องทำได้ และถ้าสิ่งที่จะต้องทำนั้นผ่านมาแล้ว จะทำการลบทิ้งให้เลยโดยอัตโนมัติ

Use Case : Update Event

Actor : ผู้ใช้

Description : ใช้สำหรับแก้ไขรายละเอียดของเหตุการณ์ที่ต้องการ โดยสามารถแก้ไขการเรียกเตือน และการบันทึกอีกครั้งเมื่อครบรอบวันเวลาต่าง ๆ ที่ต้องการ

Use Case : Delete Event

Actor : ผู้ใช้

Description : ใช้สำหรับลบเหตุการณ์ที่ต้องการนั้นทิ้งไป

Use Case : Find

Actor : ผู้ใช้

Description : ใช้สำหรับให้ผู้ใช้เข้ามาค้นหาข้อมูลต่าง ๆ ของเหตุการณ์ได้ โดยต้องทำการใส่ชื่อหัวข้อ และชนิดของเหตุการณ์ที่ต้องการค้นหา

Use Case : Quick Add

Actor : ผู้ใช้

Description : ใช้สำหรับให้ผู้ใช้เข้ามาเพิ่มเหตุการณ์ต่าง ๆ ได้ทันที โดยไม่ต้องเข้าไปที่วันเวลาของปฏิทินก่อน

Use Case : Add Event

Actor : ผู้ใช้

Description : ใช้สำหรับให้ผู้ใช้เข้ามาเพิ่มเหตุการณ์ต่าง ๆ ได้ แต่ต้องเข้ามาที่วัน เวลาที่ต้องการบันทึกก่อน โดยผ่านทางปฏิทินแบบวัน

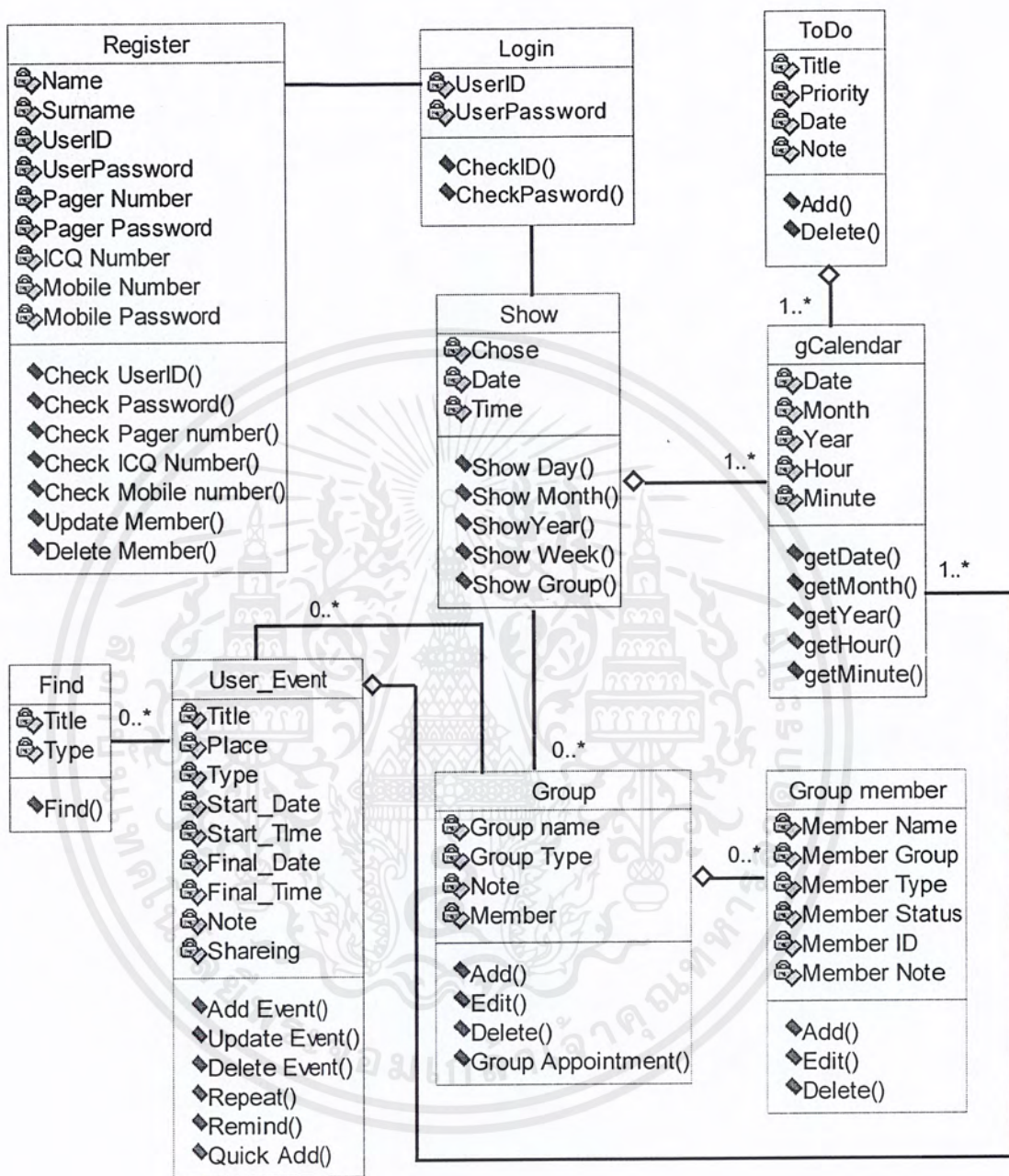
Use Case : Reminder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Actor : ผู้ใช้
- Description : ใช้สำหรับให้ผู้ใช้สามารถเลือกความต้องการให้มีการเตือนว่าจะเกิดเหตุการณ์นั้นก่อนเป็นเวลาเท่าใด และเลือกความต้องการเตือนผ่านทางโทรศัพท์มือถือ , อีเมล , เพจเจอร์ หรือ ไอซีคิว
- Use Case : Create Group
- Actor : ผู้ใช้
- Description : ให้ผู้ใช้สามารถเข้ามาสร้างกลุ่มที่ต้องการได้ แต่ต้องไม่ซ้ำกับกลุ่มและชนิดของกลุ่มที่มีอยู่
- Use Case : Create Member
- Actor : ผู้ใช้
- Description : ให้ผู้ใช้เข้ามาเพิ่มสมาชิกในกลุ่มที่ต้องการ แต่ต้องมี ID ของสมาชิกไม่ซ้ำกัน และต้องมีกลุ่มนั้นจริง
- Use Case : Delete Group
- Actor : ผู้ใช้
- Description : ให้ผู้ใช้เข้ามาลบกลุ่มที่มีอยู่ตามต้องการ ซึ่งจะเป็นการลบสมาชิกของกลุ่มออกไปทั้งหมดด้วย
- Use Case : Delete Member
- Actor : ผู้ใช้
- Description : ให้ผู้ใช้เข้ามาลบสมาชิกของกลุ่มตามต้องการ
- Use Case : Maintain Server
- Actor : Admin
- Description : ใช้สำหรับให้แอดมินทำการดูแลเซิร์ฟเวอร์ไม่ให้ down และดูแลระบบด้วย
- Use Case : Maintain Database
- Actor : Admin
- Description : ใช้สำหรับให้แอดมินดูแลฐานข้อมูลของระบบ ไม่ให้เกิดปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

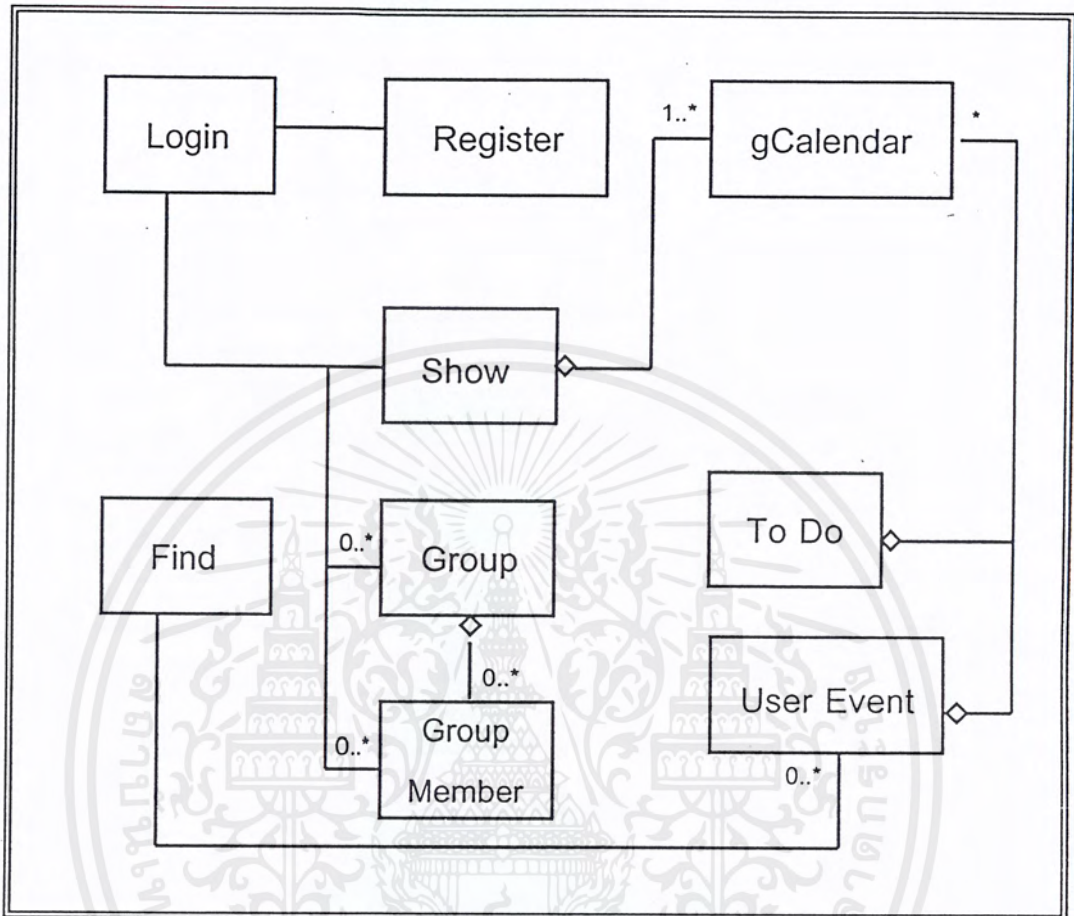
7.4.2 Class Diagram



รูปที่ 7.2 แสดงคราสไดอะแกรม (Class Diagram)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4.3 Object Diagram



รูปที่ 7.3 แสดงออบเจกต์ไดอะแกรม

ออบเจกต์ไดอะแกรม (Object Diagram)

ออบเจกต์ Login ทำหน้าที่ตรวจสอบ ID และ รหัสผ่าน

ออบเจกต์ Register ทำหน้าที่ตรวจสอบ และบันทึกข้อมูลในการสมัครสมาชิก

ออบเจกต์ Show ทำหน้าที่แสดงปฏิทินเป็น วัน สัปดาห์ และปีตามที่ผู้ใช้งานต้องการ

ออบเจกต์ To Do ทำหน้าที่บันทึก และแสดงเหตุการณ์ที่เป็น To Do ตามที่ผู้ใช้งานกำหนด

ออบเจกต์ User_Event ทำหน้าที่บันทึกเหตุการณ์ต่างๆ ตามเวลาที่ผู้ใช้งานต้องการ

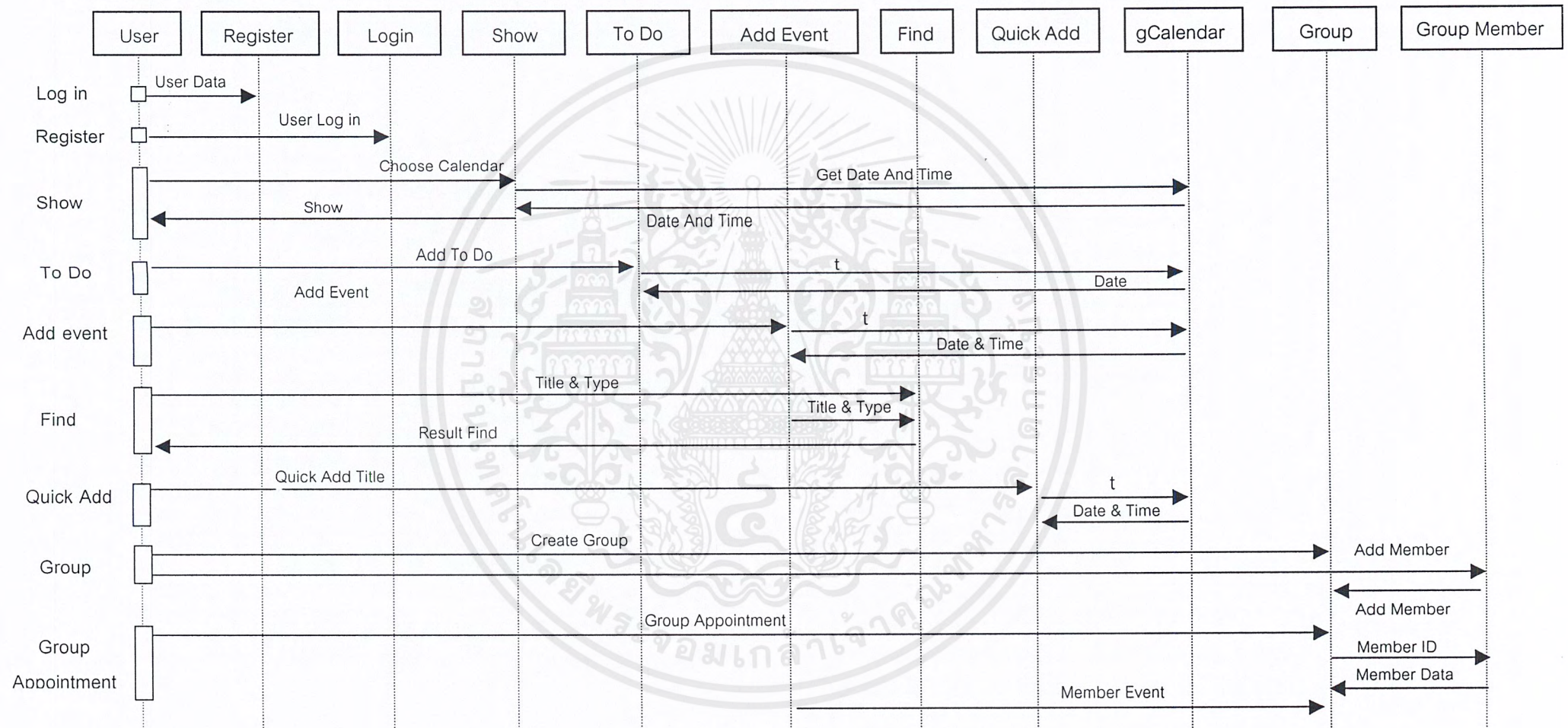
ออบเจกต์ Find ทำหน้าที่ค้นหาเหตุการณ์ต่างๆ ที่ผู้ใช้งานสามารถบันทึกไว้

ออบเจกต์ Group ทำหน้าที่สร้างกลุ่มสมาชิก เพื่อทำการนัดหมายเป็นหมู่คณะ หรือคนในกลุ่ม

ออบเจกต์ Group Member ทำหน้าที่เพิ่ม และเก็บข้อมูลของสมาชิกภายในกลุ่ม

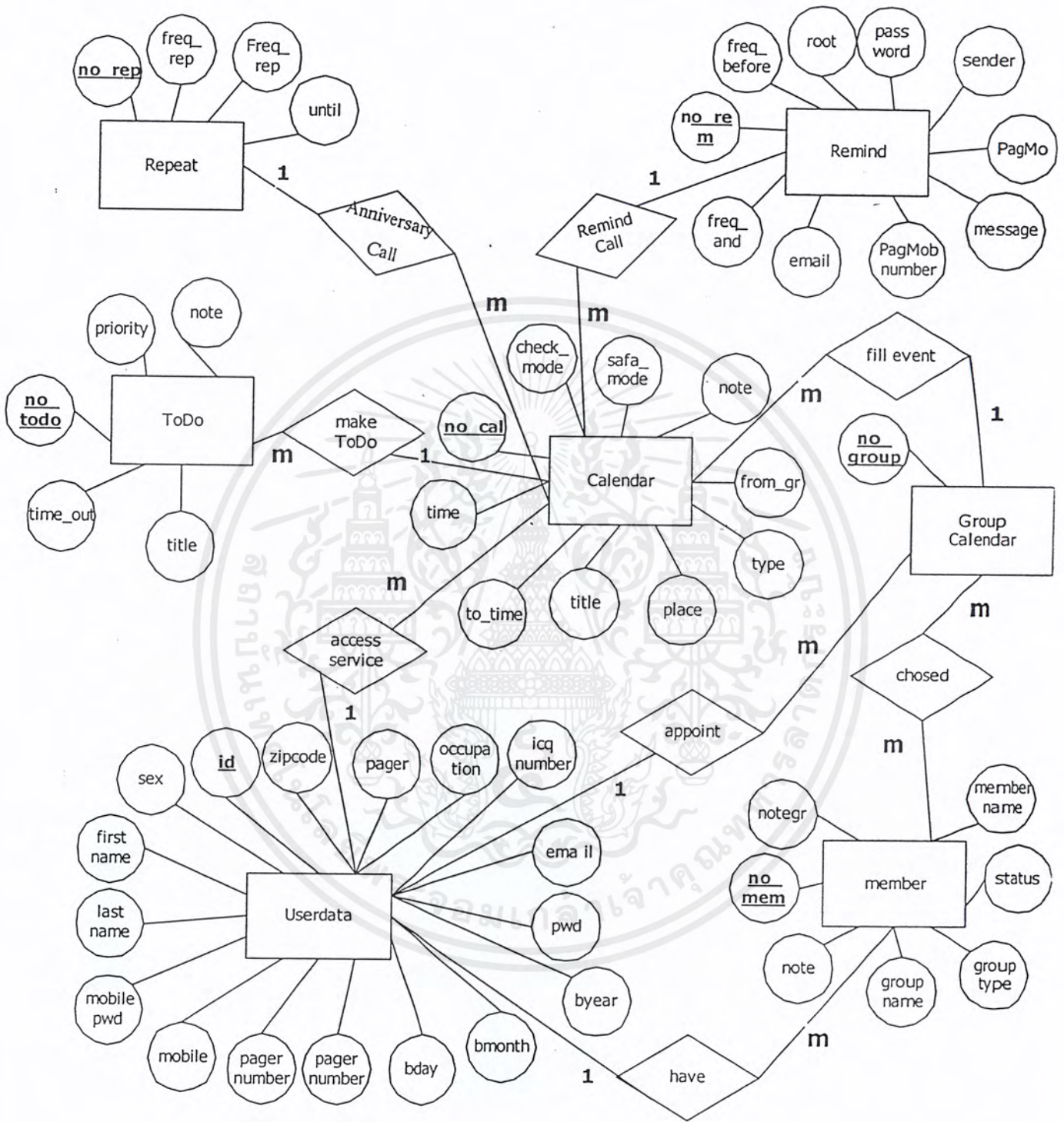
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4.3 Sequence Diagram



รูปที่ 7.4 แสดง Sequence Diagram

7.4.4 อีไอโอะแกรม (ER Diagram)



รูปที่ 7.5 แสดง ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง Calendar

P.K

No_cal	Time	To_time	Title	Type	Safe_mode
Check_mode	From_gr	Note	Place	No_rep	No_rem
No_group	id	Freq_rep			

ตารางที่ 7-1 แสดงตารางฐานข้อมูล Calendar

ตาราง Userdata

Id	Pwd	Firstname	Lastname	Country	Zipcode
Sex	Pager	Pagernumber	Pagerpwd	Mobile	MobileNum
Bday	Bmonth	Byear	icqnumber	Email	Time_Zone

ตารางที่ 7-2 แสดงตารางฐานข้อมูล Userdata

ตาราง Repeat

No_rep	Until	Freq_rep	Until_limit
--------	-------	----------	-------------

ตารางที่ 7-3 แสดงตารางฐานข้อมูล Repeat

ตาราง Remind

No_rem	Freq_before	Freq_and	Message	Pagmob	PagMobnum
sender	Email	password	Root		

ตารางที่ 7-4 แสดงตารางฐานข้อมูล Remind

ตาราง ToDo

No_todo	Priority	note	Title	Time_out	No_cal
---------	----------	------	-------	----------	--------

ตารางที่ 7-5 แสดงตารางฐานข้อมูล To Do

ตาราง Member

No_mem	Grouptype	Groupname	Note	Notegr	Status
Membername	idmem	password	id		

ตารางที่ 7-6 แสดงตารางฐานข้อมูล Member

ตาราง Groupcal

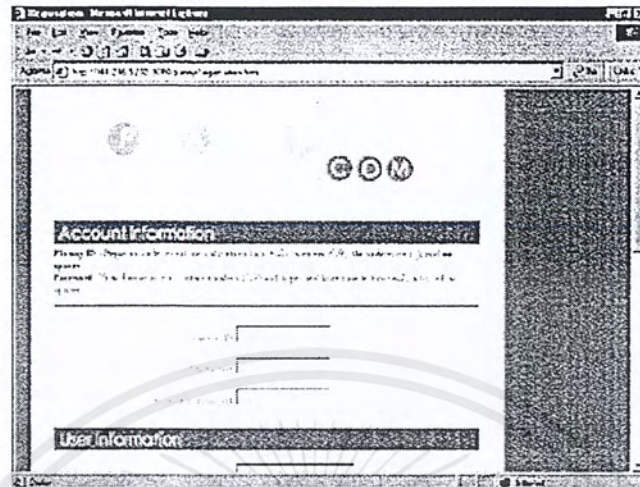
No_group	No_mem	id
----------	--------	----

ตารางที่ 7-7 แสดงตารางฐานข้อมูล Groupcal

หมายเหตุ Primary Key = ตัวหนา ขีดเส้นใต้ Foreign Key = ตัวหนา ตัวเอียง

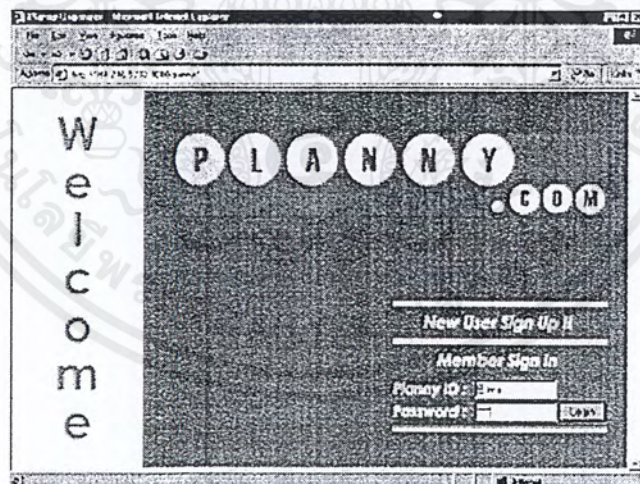
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4.5 การออกแบบส่วนที่ติดต่อกับผู้ใช้ และฟังก์ชันการทำงาน



รูปที่ 7.6 แสดงหน้าจอการสมัครเป็นสมาชิก

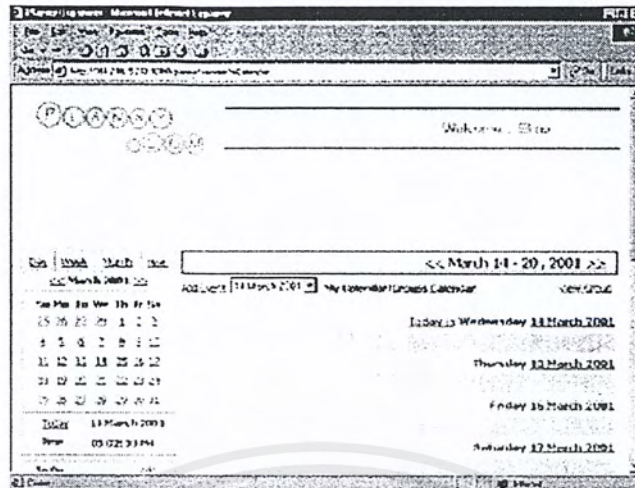
เมื่อผู้ใช้ต้องการเข้าใช้บริการ ต้องสมัครเป็นสมาชิกก่อนเป็นอันดับแรก โดยต้องกรอกรายละเอียดของผู้ใช้ให้ครบถ้วนถูกต้อง และต้องใส่ ID และ Password ที่คิดขึ้นมา โดย ID ต้องไม่ซ้ำกับผู้อื่น



รูปที่ 7.7 แสดงหน้าจอ Login ของผู้ใช้

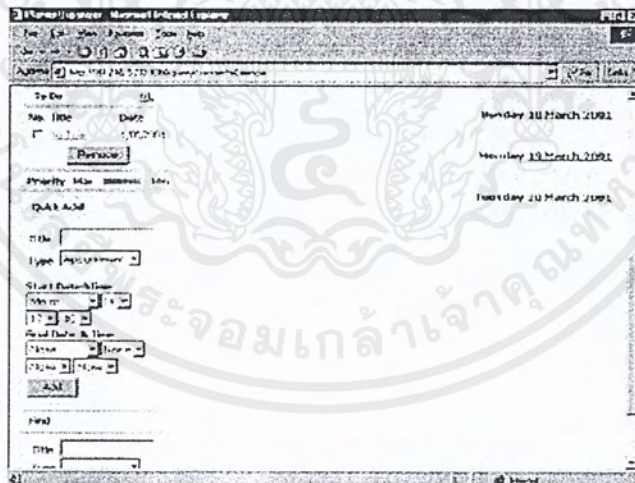
เมื่อผู้ใช้ได้สมัครเป็นสมาชิกแล้ว ก็สามารถจะเข้ามาใช้บริการต่าง ๆ ได้ โดยต้องผ่านเข้าหน้าจอ Login ก่อนเป็นอันดับแรก โดยต้องใส่ ID และ Password ของตัวเองให้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.8 แสดงหน้าจอหลักของโครงการนี้

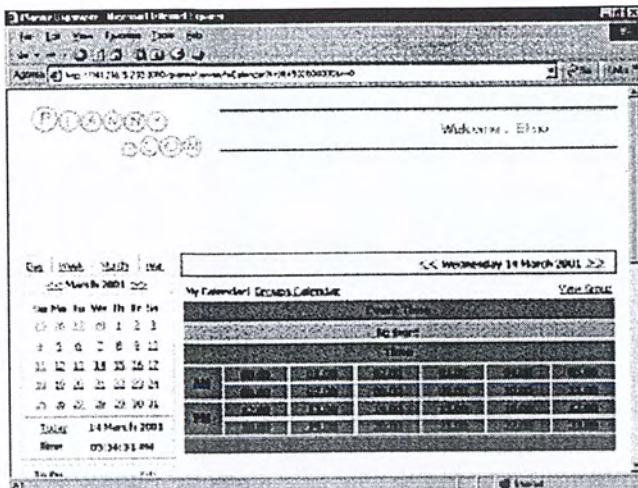
เมื่อผู้ใช้ Login เข้ามาหน้าแล้ว ถ้า ID และ Password ถูกต้อง ผู้ใช้ก็สามารถเข้ามาดูหน้านี้ โดยหน้าจะแสดงฟังก์ชันการทำงานทั้งหมดของโครงการนี้ โดยหน้าจอที่แสดงจะโชว์ตารางปฏิทิน โดยสามารถจะเลือกดูแบบเป็นวัน ,อาทิตย์,เดือน หรือปีได้ และยังแสดงวัน เวลาปัจจุบัน และถ้ามีเหตุการณ์สำคัญ ๆ ที่ได้บันทึกไว้แล้ว ก็จะแสดงเหตุการณ์ต่าง ๆ นั้นด้วย



รูปที่ 7.9 แสดงหน้าจอของ To Do ,Quick Add และ Find

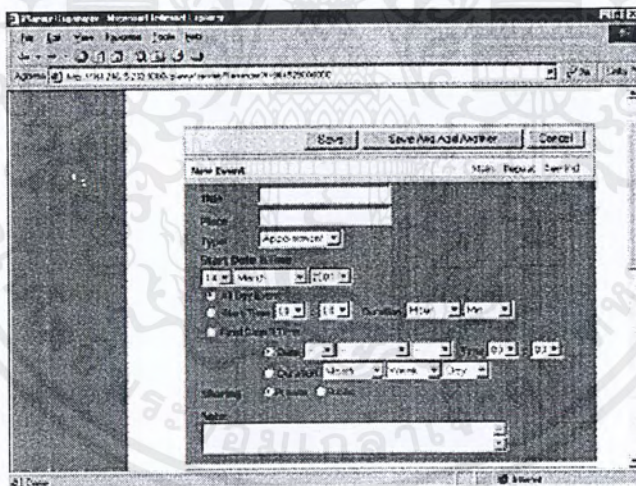
หน้านี้แสดงฟังก์ชัน To Do ซึ่งจะทำหน้าที่ให้ผู้ใช้สามารถเข้ามาบันทึกสิ่งที่จะต้องทำ โดยมีลำดับความสำคัญต่าง ๆ แสดงให้เห็นในแต่หัวข้อ , ฟังก์ชัน Quick Add ทำหน้าที่ให้ผู้ใช้เข้ามาบันทึกเหตุการณ์สำคัญต่าง ๆ ได้ทันที และฟังก์ชัน Find ทำหน้าที่ให้ผู้ใช้เข้ามาหาหัวข้อและชนิดของเหตุการณ์ต่าง ๆ ได้ โดยจะแสดงรายละเอียดของเหตุการณ์ที่ผู้ใช้ต้องการหาขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.10 แสดงหน้าจอปฏิทินแบบวัน

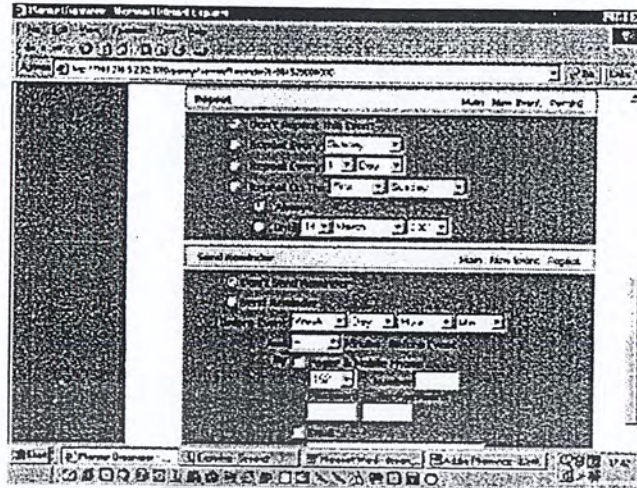
เมื่อผู้ใช้งานต้องการจะบันทึกเหตุการณ์สำคัญต่าง ๆ ต้องเลือกวันที่จะบันทึก โดยเมื่อเลือกวันแล้ว จะเข้ามาสู่หน้านี้ โดยจะแสดงเวลาต่าง ๆ ที่มีอยู่ในวันนั้น เพื่อให้ผู้ใช้เลือกเวลาที่บันทึกเหตุการณ์



รูปที่ 7.11 แสดงหน้าจอ Insert Event

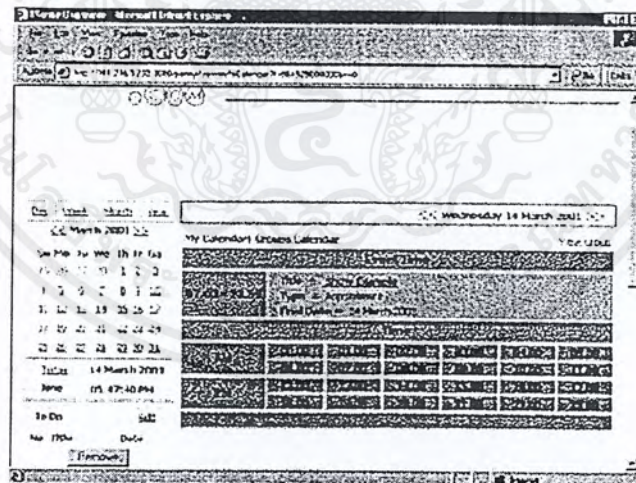
เมื่อผู้ใช้เลือกวัน เวลาที่จะบันทึกเหตุการณ์แล้ว จะเข้าสู่หน้าจอ Insert Event ซึ่งจะให้ผู้ใช้งานกรอกรายละเอียดของเหตุการณ์ที่จะบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.12 แสดงหน้าจอการ Repeat และ Remind

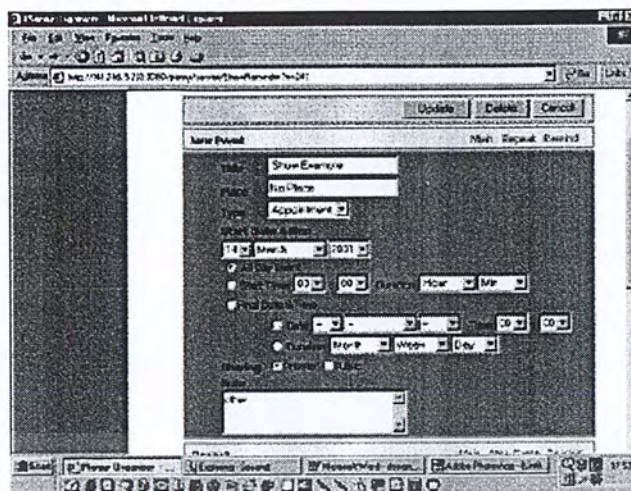
เมื่อผู้ใช้งานบันทึกเหตุการณ์แล้ว จะให้ผู้ใช้เลือกว่าจะให้มีการบันทึกเหตุการณ์นี้อีกไหม เมื่อครบรอบวัน เวลาต่าง ๆ ซึ่งจะให้ผู้เลือกได้ ซึ่งฟังก์ชันเรียกว่า Repeat และ จะให้ผู้ใช้เลือกว่าต้องการให้มีการเตือนผู้ใช้งานว่าจะเกิดเหตุการณ์นี้ก่อนเป็นเวลาเท่าไร ซึ่งฟังก์ชันเรียกว่า Repeat



รูปที่ 7.13 แสดงหน้าจอที่แสดงเหตุการณ์ต่าง ๆ

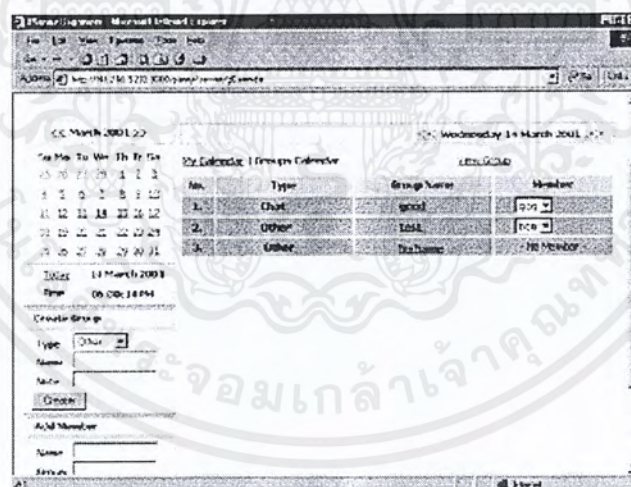
หน้าจอที่แสดงเหตุการณ์ต่าง ๆ ที่ได้มีการบันทึกไว้แล้ว โดยจะแสดงในช่องวันและเวลาที่เกิดเหตุการณ์นี้ เมื่อผู้ใช้ต้องการที่จะดูหรือแก้ไขเหตุการณ์ใด ก็ทำการคลิกที่หัวข้อของเหตุการณ์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.14 แสดงหน้าจอการ Update และ Delete เหตุการณ์

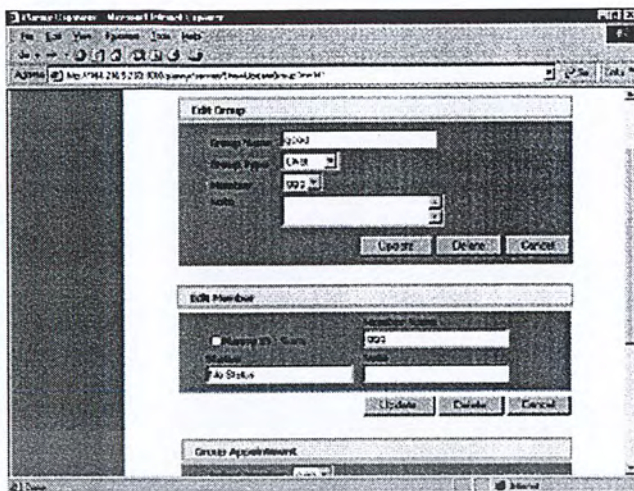
เมื่อผู้ใช้คลิกที่หัวข้อที่ต้องการดู หรือแก้ไข ก็จะเข้าสู่หน้านี้ ซึ่งจะแสดงรายละเอียดของเหตุการณ์นั้น ๆ ถ้าผู้ใช้ต้องการแก้ไข ก็สามารถลบข้อความเก่า และใส่ข้อความใหม่แทนที่ ซึ่งจะสามารถเปลี่ยนแปลงรายละเอียดของเหตุการณ์, Repeat และ Remind ซึ่งผู้ใช้จะต้องกดปุ่ม Update เหตุการณ์นั้นก็จะเปลี่ยนแปลงไป และถ้าผู้ใช้ต้องการลบเหตุการณ์นั้นทิ้ง ให้ผู้ใช้กดปุ่ม Delete เหตุการณ์นั้นก็จะหายไป



รูปที่ 7.15 แสดงรายละเอียดของกลุ่ม

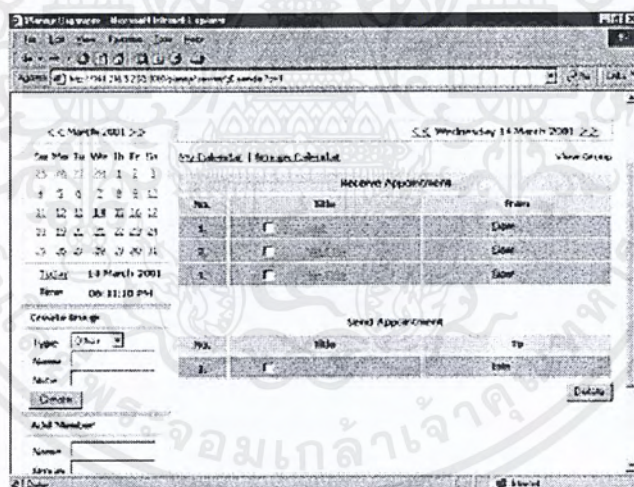
เมื่อผู้ใช้เข้ามาที่หน้าจอนี้ จะแสดงรายละเอียดของกลุ่มที่ผู้ใช้สร้างขึ้น โดยจะแสดงชื่อกลุ่ม และชนิดของกลุ่ม เช่น เพื่อน หรือ กีฬา เป็นต้น และจะแสดงรายชื่อสมาชิกในแต่ละกลุ่มด้วย ซึ่งหน้านี้ผู้ใช้สามารถสร้างกลุ่มขึ้นมาใหม่ได้ และสร้างสมาชิกขึ้นมาได้ โดยทำการกรอกรายละเอียดด้านซ้ายมือของหน้าจอให้ครบถ้วนถูกต้อง ซึ่งถ้ากรอกผิด จะแสดงหน้าจอเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.16 แสดงหน้าจอรายละเอียดของกลุ่ม

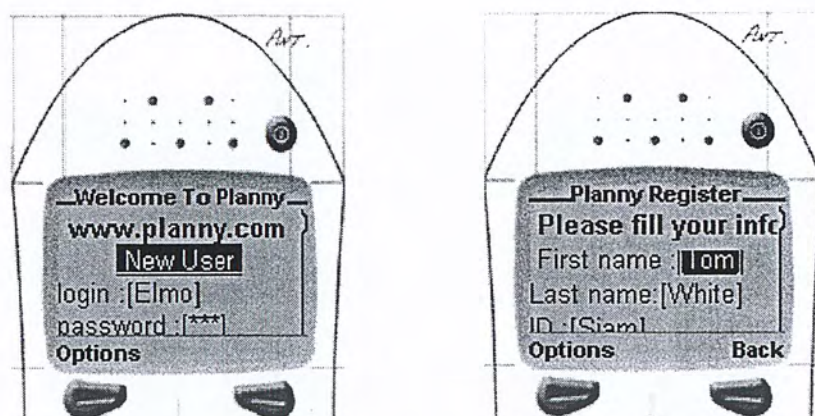
เมื่อผู้ใช้คลิกที่ชื่อกลุ่มใด ๆ จะเข้าสู่หน้านี้ โดยจะแสดงรายละเอียดของกลุ่ม และรายละเอียดของสมาชิก ส่วนด้านล่างของหน้าจอให้ผู้ใช้สามารถพิมพ์ข้อความและเหตุการณ์ที่จะทำกรนัดหมายสมาชิกในกลุ่มได้



รูปที่ 7.17 แสดงหน้าจอเหตุการณ์ต่าง ๆ ของกลุ่ม

เมื่อผู้ใช้เข้ามาที่ View Group จะเข้าสู่หน้าจอนี้ ซึ่งแสดงหัวข้อต่าง ๆ ที่มีคนนัดหมายกับเรา และเรานัดหมายผู้อื่น ซึ่งหัวข้อต่าง ๆ ล้วนมีสถานะแตกต่างกันไป คือ สมาชิกตอบรับการนัดหมาย, สมาชิกปฏิเสธการนัดหมาย, มีการนัดหมายจากผู้อื่น, การนัดหมายผู้อื่น, ใ้คำตอบจากสมาชิก และยังไม่ให้คำตอบ ซึ่งสถานะต่าง ๆ เหล่านี้จะแสดงเป็นสีต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.18 แสดงหน้าจอโทรศัพท์ที่ไร้สายที่แสดงการ Login และสมัครสมาชิก

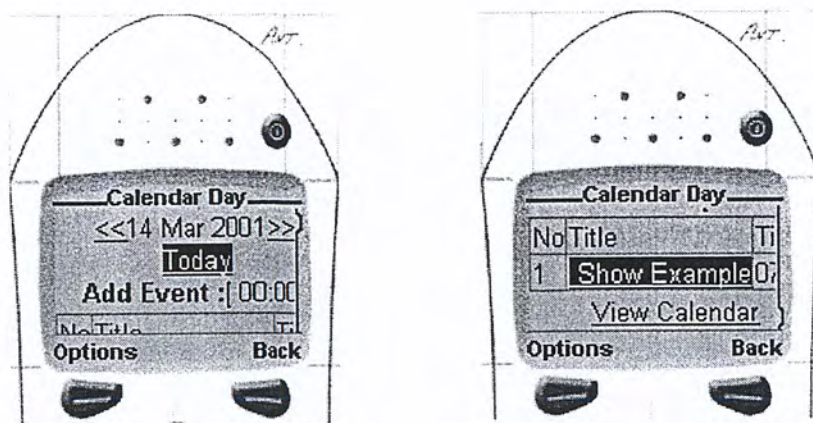
เมื่อผู้ใช้เข้าแอดเดรสของเราแล้วจะเข้าสู่หน้าแรกคือ Welcome To Planny และเมื่อผู้ใช้ไม่ได้เป็นสมาชิกต้องคลิกที่ New User ซึ่งจะไปหน้า Register ซึ่งเป็นหน้าที่ให้ผู้ใช้กรอกรายละเอียดของตัวเอง รวมทั้ง ID และ Password ด้วย เมื่อเรียบร้อยแล้ว ผู้ใช้ก็สามารถเข้าหน้าต่อไปได้ ส่วนผู้ที่已经是สมาชิกแล้ว ก็ให้ใส่ ID และ Password จากนั้นเลือกที่ Option แล้วเลือกคำสั่ง Login ก็จะเข้าหน้าจอต่อไป



รูปที่ 7.19 แสดงหน้าจอโทรศัพท์ที่ไร้สายที่แสดงฟังก์ชันการทำงานและหน้าจอ View Calendar

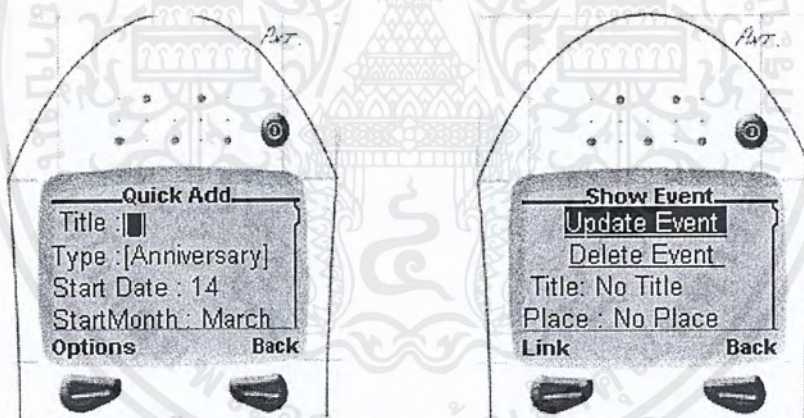
เมื่อผู้ใช้ Login เรียบร้อยแล้วจะเข้าสู่หน้าจอแสดงฟังก์ชันการทำงาน ซึ่งจะให้ผู้เลือกใช้ว่าการเข้าใช้บริการฟังก์ชันไหน ซึ่งในที่นี้เลือก My Calendar ก็จะเข้าสู่หน้าจอ View Calendar ซึ่งให้ผู้เลือกใช้ว่าการดูปฏิทินแบบวัน , สัปดาห์ และเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.20 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงหน้าจอ Calendar Day

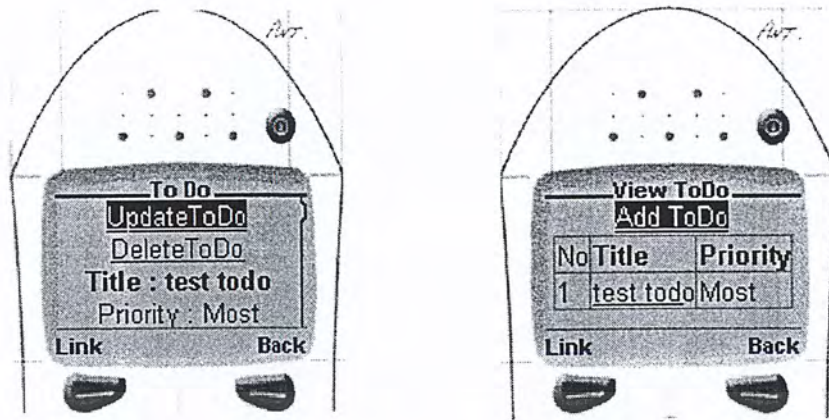
เมื่อผู้ใช้เลือกเข้าหน้า Calendar Day แล้ว จะเห็นหน้าจอที่แสดงวันที่ปัจจุบัน และสามารถที่จะเลื่อนไปวันอื่น ๆ ได้ตามต้องการ ซึ่งจะกลับมาที่หน้าเดิมได้ โดยคลิกที่ Today และยังสามารถที่จะบันทึกเหตุการณ์สำคัญต่าง ๆ ได้ โดยคลิกที่ Option แล้วตามด้วย Insert โดยสามารถเลือกเวลาที่ต้องการบันทึกเหตุการณ์นั้นได้ด้วย เมื่อคลิกลงมา จะเห็นหน้าจอที่แสดงเหตุการณ์สำคัญต่าง ๆ ที่จะเกิดขึ้นในวันนี้ ซึ่งเราสามารถเข้าไปดูรายละเอียดได้ โดยคลิกที่หัวข้อเหตุการณ์นั้น และเมื่อต้องการที่จะกลับไปเลือกดูปฏิทินอื่น ๆ ให้คลิกที่ View Calendar



รูปที่ 7.21 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงหน้าจอ QuickAdd และ Show Event

เมื่อผู้ใช้เลือกเวลาที่ต้องการบันทึกเหตุการณ์ได้แล้ว จะเข้ามาที่หน้า Quick Add ซึ่งจะให้ผู้กรอกรายละเอียดของเหตุการณ์ และเมื่อกรอกรียบร้อยแล้ว ให้กดปุ่ม Option แล้วเลือก Save ข้อมูลต่าง ๆ ของเหตุการณ์ก็จะถูกบันทึก ซึ่งหน้านี้ถ้าเลือกฟังก์ชัน Quick Add ที่หน้าฟังก์ชันหลัก ก็จะเข้ามาที่หน้านี้เช่นกัน ส่วนอีกหน้าแสดงรายละเอียดของเหตุการณ์ที่ได้บันทึกแล้ว โดยที่คลิกที่หัวข้อนั้น ๆ ซึ่งหน้านี้มีฟังก์ชัน Update Event ซึ่งจะไปหน้าที่ให้แก้ไขข้อมูลต่าง ๆ ของเหตุการณ์ และฟังก์ชัน Delete Event ซึ่งเป็นการลบเหตุการณ์นี้ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.22 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดง View To Do และรายละเอียดของ To Do

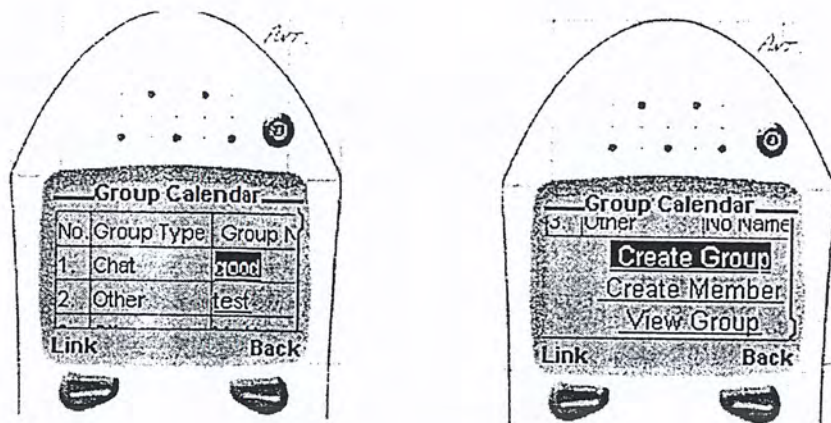
เมื่อผู้ใช้เลือกฟังก์ชัน To Do ในหน้าฟังก์ชันหลัก จะเข้าสู่หน้า View To Do ซึ่งจะแสดงสิ่งที่จะต้องทำต่าง ๆ โดยแต่ละสิ่งจะมีค่าความสำคัญแตกต่างกัน และเมื่อเลือก Add To Do จะเป็นการบันทึกสิ่งที่จะต้องทำสิ่งใหม่ ซึ่งเมื่อกรอกรายละเอียดเสร็จ ก็จะทำการบันทึกเก็บไว้ สำหรับหน้าที่สองคือหน้าแสดงรายละเอียดของสิ่งที่จะต้องทำ ซึ่งหน้านี้จะเข้ามาได้โดยคลิกที่หัวข้อของสิ่งที่จะต้องทำในหน้า View To Do และหน้านี้ยังมีฟังก์ชัน Update To Do ซึ่งทำหน้าที่เปลี่ยนแปลงแก้ไข รายละเอียดของสิ่งที่จะต้องทำ และฟังก์ชัน Delete To Do เป็นการลบสิ่งที่จะต้องทำออก



รูปที่ 7.23 แสดงหน้าจอโทรศัพท์ไร้สายที่แสดงฟังก์ชัน Find และ Result Find

เมื่อผู้ใช้เลือกฟังก์ชัน Find ในหน้าจอฟังก์ชันหลัก จะเข้ามาสู่หน้าจอ Find ซึ่งจะให้ผู้ใช้อกรอกชื่อหัวข้อที่ต้องการค้นหา หรือชนิดของเหตุการณ์ที่ต้องการค้นหา หรือสองอย่างก็ได้ จากนั้นให้คลิกที่ปุ่ม Option แล้วเลือก Find จะเข้าสู่หน้าจอถัดไปคือ หน้าจอ Result Find ซึ่งเป็นหน้าที่แสดงหัวข้อและชนิดของเหตุการณ์ที่ต้องการค้นหา และผู้ใช้สามารถเข้าไปดูรายละเอียดของเหตุการณ์ได้โดยการคลิกที่หัวข้อของเหตุการณ์ที่สนใจนั้น ๆ ก็จะไปหน้าที่แสดงเหตุการณ์นั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.24 แสดงหน้าจอโทรศัพท์ที่ไร้สายที่แสดงรายละเอียดของกลุ่ม

เมื่อผู้ใช้เลือก Group Calendar ที่หน้าจอฟังก์ชันหลัก จะเข้าสู่หน้าจอ Group Calendar ซึ่งแสดงรายละเอียดของกลุ่มที่มี โดยจะมีชื่อของกลุ่ม และชนิดของกลุ่ม โดยด้านล่างของหน้าจอจะเป็นฟังก์ชัน Create Group ซึ่งเป็นการเพิ่มกลุ่ม โดยจะให้กรอกรายละเอียดของกลุ่ม แล้วทำการบันทึก แต่มีเงื่อนไขว่าห้ามใส่ชื่อกลุ่ม และชนิดของกลุ่มซ้ำกับกลุ่มที่มีอยู่, ฟังก์ชันที่สองคือ Create Member เป็นการเพิ่มสมาชิกให้กับกลุ่ม โดยมีเงื่อนไขว่าต้องเพิ่มในกลุ่มที่มีอยู่ และฟังก์ชันสุดท้ายคือ View Group เป็นการเข้าไปดูรายละเอียดของเหตุการณ์ที่ได้มีการนัดหมาย



รูปที่ 7.25 แสดงหน้าจอโทรศัพท์ที่แสดงหัวข้อและรายละเอียดของเหตุการณ์ที่มีการนัดหมาย

เมื่อผู้ใช้เลือกฟังก์ชัน View Group แล้วจะเข้าสู่หน้า View Calendar โดยจะแสดงหัวข้อและสถานะต่าง ๆ ของเหตุการณ์ที่มีการนัดหมาย ซึ่งสถานะดังกล่าวมีดังนี้ คือ มีการนัดหมายจากผู้อื่น จะแสดงข้อความคือ "Got Appointment" , นัดหมายผู้อื่น จะแสดงข้อความ คือ "Waiting" , มีการตอบรับจากสมาชิก จะแสดงข้อความ คือ "OK" , มีการปฏิเสธจากสมาชิก จะแสดงข้อความ คือ "No" , ถ้ายังไม่เลือกคำตอบ จะแสดงข้อความ คือ "Not Answer" , ถ้าได้อ่านคำตอบจากสมาชิกแล้ว จะแสดงข้อความ คือ "Viewed" หน้าต่อมาคือ Show View Calendar ซึ่งจะแสดงรายละเอียดของเหตุการณ์นั้น ๆ ซึ่งสามารถที่จะเลือก ฟังก์ชัน Delete , Update หรือ Cancel ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8 บทวิจารณ์และสรุป

8.1 สรุปผลการดำเนินงาน

โครงการโปรแกรมการนัดหมายผ่านเครือข่ายอินเทอร์เน็ต สามารถอำนวยความสะดวกแก่ผู้ใช้ได้อย่างมาก แทนที่แต่เดิมจะใช้กระดาษเพื่อจดบันทึกสิ่งสำคัญต่าง ๆ ที่ต้องการไว้ ก็หันมาจดบันทึกบนเว็บไซต์แทนได้ หรือการใช้เครื่องปาล์ม (Palm) เพื่อช่วยในการจดบันทึกเหตุการณ์นั้นก็มีความสะดวกในการใช้งานมากแต่ค่าใช้จ่ายค่อนข้างแพง ซึ่งถ้ามาเปรียบเทียบกับการใช้โปรแกรมที่ได้พัฒนาขึ้นนั้น ก็มีความสะดวกเช่นกัน เพราะ สามารถใช้โทรศัพท์มือถือเข้ามาใช้บริการได้ หรือใช้งานผ่านเว็บเบราว์เซอร์ก็ได้ นอกจากนี้ค่าใช้จ่ายในการใช้งานก็น้อยกว่าการใช้เครื่องปาล์มอย่างมาก

โครงการนี้ก็ได้ใช้โปรแกรมจาวาเซิร์ฟเริ่ตมาทำงานแทนที่โปรแกรม CGI ซึ่งทำให้เกิดข้อได้เปรียบหลายประการ เช่น การเขียนโปรแกรมแบบออบเจ็ค โอเรียนเต็ล ,การทำงานของโปรแกรมเซิร์ฟเริ่ตจะรวดเร็วและใช้ทรัพยากรน้อยกว่าโปรแกรม CGI เพราะจาวาเซิร์ฟเริ่ตจะสร้างหนึ่งเทรคต่อหนึ่งการร้องขอของไคลเอนต์ ในขณะที่การเขียนโปรแกรม CGI จะสร้างหนึ่งโพรเซสต่อหนึ่งการร้องขอ นอกจากนี้ยังได้ใช้โปรแกรมออรากิล (Oracle 8i) มาจัดการฐานข้อมูล ซึ่งโปรแกรมออรากิลมีประสิทธิภาพและความปลอดภัยในการจัดการฐานข้อมูลสูง นอกจากนี้การเขียนโปรแกรมติดต่อด้วยภาษาด้วยภาษาจาวายังสามารถทำได้อย่างสะดวกและรวดเร็วอีกด้วย

เนื่องจากการทำงานของเว็บอยู่บนระบบปฏิบัติการวินโดว์ ทำให้มีประสิทธิภาพและเสถียรภาพน้อยกว่าการทำงานบนระบบปฏิบัติการบนยูนิกซ์ (UNIX) การเขียนโปรแกรมด้วยภาษาจาวาเซิร์ฟเริ่ตก็ยังมีข้อจำกัดในเรื่องของการสร้างเว็บเพจ คือ ไม่สามารถนำโปรแกรมที่ได้เขียนนั้นไปแทรกในเอกสาร HTML ที่ใช้แสดงผลบนเว็บเบราว์เซอร์ได้ ทำให้ต้องเขียนต้องโปรแกรมจาวาเซิร์ฟเริ่ตสร้างเอกสาร HTML ทั้งหมดเอง ทำให้เว็บเพจที่สร้างขึ้นมานั้นอาจไม่สวยงามเท่าที่ควร และยากในการอัปเดตเว็บเพจที่สร้างขึ้น โปรแกรมออรากิลก็ใช้ทรัพยากรบนเครื่องเซิร์ฟเวอร์ค่อนข้างมาก ทำให้ต้องใช้เครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง สำหรับเทคโนโลยีเว็บนั้น ยังมีข้อจำกัดอีกหลายอย่าง เช่น หน้าจอที่แสดงผลนั้นมีจำกัด และแสดงผลได้เฉพาะสีขาวดำ ทำให้ไม่สะดวกในการใช้งาน ส่วน User Interface ที่ได้ก็ไม่สวยงาม นอกจากนี้โครงการนี้พัฒนาขึ้นมาโดยใช้เว็บเบราว์เซอร์ ซึ่งได้จากการดาวโหลด โปรแกรม Nokia Toolkit เนื่องจากราคาของโทรศัพท์มือถือ และค่าบริการในการใช้เว็บบนโทรศัพท์มือถือยังคงมีราคาสูงอยู่มาก จึงไม่สามารถได้ทดลอง และใช้งานได้จริงกับโทรศัพท์มือถือที่สามารถเข้าดูเว็บเพจได้

8.2 แนวทางในการพัฒนาต่อ

ระบบที่เราพัฒนาขึ้นมาสามารถนำไปใช้งานได้จริง และตอบสนองต่อความต้องการขั้นพื้นฐาน และอำนวยความสะดวกให้กับผู้ใช้ ในการจัดการนัดหมาย เพื่อช่วยเรียกเตือนความจำสำหรับเหตุการณ์ต่าง ๆ ที่ได้บันทึกไว้ แต่ถ้าพิจารณากันดี ๆ แล้ว โครงการนี้ยังมีจุดบกพร่องหลายอย่าง คือ ไม่สามารถคำนวณวันข้างขึ้นข้างแรม เพื่อที่จะคำนวณหาวันหยุดและวันสำคัญต่าง ๆ ที่ขึ้นอยู่กับข้างขึ้นข้างแรมได้ , เซิร์ฟเวอร์ที่ใช้ทำงานอยู่บนระบบปฏิบัติการวินโดว์ที่อาจจะไม่เสถียร เท่ากับการทำงานบนยูนิกซ์หรือดี เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อสาธารณะไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นุกส์ได้ , และข้อมูลการนัดหมายที่ได้ทำการบันทึกไว้ ยังไม่สามารถใช้ร่วมกับเครื่องปาล์มได้ ซึ่งถ้ามีผู้ที่สนใจในโครงการชิ้นนี้ หวังว่าคงจะช่วยพัฒนาให้มีการทำงานร่วมกันระหว่างอินเทอร์เน็ต , โทรศัพท์มือถือ และเครื่องปาล์มได้ ซึ่งจะช่วยให้สร้างความสะดวก และครอบคลุมการทำงานได้อย่างมีประสิทธิภาพมาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ก. การติดตั้ง Jakarta-Tomcat เว็บเซิร์ฟเวอร์

Tomcat เป็น Servlet Engine จากค่าย Apache โดยในอนาคต Tomcat จะเป็นตัวที่ใช้ในการรัน Servlet แทน Jserv ซึ่งใช้กันอยู่ในปัจจุบัน Tomcat สามารถใช้ได้เป็นทั้ง stand-alone container คล้ายกับจาวาเว็บเซิร์ฟเวอร์ (Java Web Server) หรือ จะเป็นตัว add-on กับเว็บเซิร์ฟเวอร์ต่างเช่น Apache Web Server , IIS , Netscape Server และเว็บเซิร์ฟเวอร์อื่นที่สนับสนุน

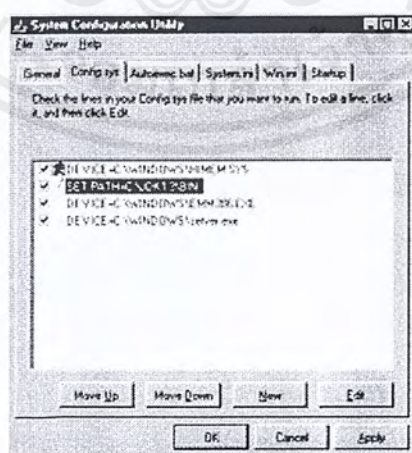
การติดตั้งเว็บเซิร์ฟเวอร์ Jakarta Tomcat จะต้องทำการติดตั้ง Java Development Kit (JDK) เพราะโปรแกรม Tomcat ถูกเขียนขึ้นโดยใช้จาวา ซึ่งจะรันกับ Java Development Kit (JDK) นอกจากนั้นทาง apache ยังคงทำการพัฒนา Tomcat ให้มีประสิทธิภาพมากเพื่อใช้เป็นมาตรฐานในอนาคต ดังนั้นถ้าเข้าไปที่เว็บไซต์ในส่วนของดาวน์โหลดคือ <http://jakarta.apache.org> จะเห็นตัวเลือกในการดาวน์โหลดได้คร่าว ๆ เป็นสองแบบ คือ แบบ Source File ที่ต้องมาคอมไพล์เองโดยใช้เครื่องมือช่วยที่เรียกว่า ANT ซึ่งเป็นส่วนที่นักพัฒนานิยมดาวน์โหลดกัน กับแบบที่คอมไพล์แล้ว binary พร้อมทั้งจะรันซึ่งเหมาะสำหรับคนที่อยากนำ Tomcat มาใช้งานอย่างเดียว

ขั้นตอนการติดตั้ง Web Server Jakarta -Tomcat บนไมโครซอฟต์วินโดวส์ (MicroSoft Window)

1. ทำการดาวน์โหลด Java Development Kit (JDK) สำหรับวินโดวส์และยูนิกซ์ให้ไปดาวน์โหลดที่ <http://java.sun.com/jdk> สำหรับ ลิงก์ให้ไปที่ <http://www.blackdown.org/>.

2. ทำการดาวน์โหลดเว็บเซิร์ฟเวอร์ Tomcat ได้จาก <http://jakarta.apache.org>

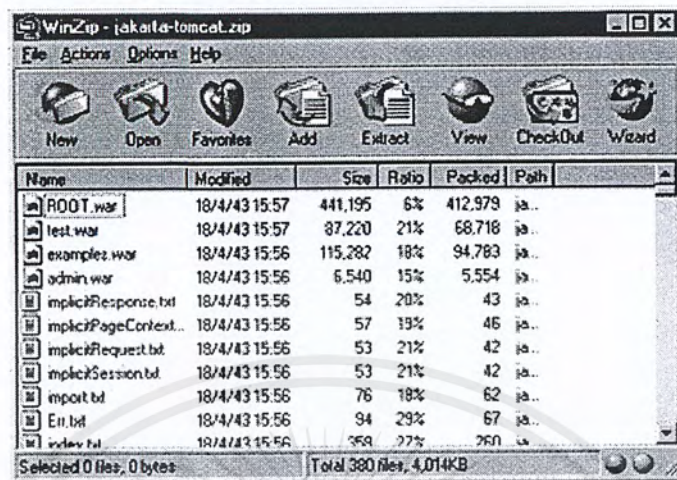
3. ทำการติดตั้ง JDK สมมติ JDK ที่ใช้เป็น JDK 1.3 ให้เซตค่า System Environment คือค่า PATH เป็น C:\jdk1.3\bin โดยใช้คำสั่งในคอส คือ Set PATH = C:\jdk1.3\bin ; หรือ คลิกขวาที่ My Computer แล้วเลือกที่ Environment แท็บ ที่ต้องเซต PATH เพราะ Tomcat จะรันโดยใช้คำสั่ง java.exe ซึ่งอยู่ใน bin ไดเรกทอรีของ JDK ดังรูป



รูปที่ ก-1 แสดงการเซตค่า System Environment สำหรับการติดตั้ง JDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

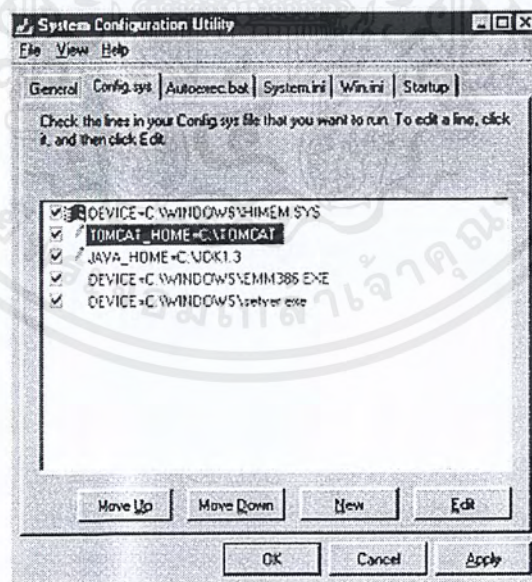
4. ทำการ Unzip ไฟล์ Jakarta-Tomcat.zip ที่ดาวน์โหลดมา หลังจากที่ได้ Unzip เสร็จให้ทำการเปลี่ยนชื่อไดเรกทอรีเป็น C:\tomcat



รูปที่ ก-2 แสดงการ Unzip ไฟล์ที่ได้ดาวน์โหลดมา

5. ทำการเซตค่า Environment Variable คือ TOMCAT_HOME เพื่อชี้ไปยังไดเรกทอรีของ tomcat ซึ่งในที่นี้คือ C:\tomcat

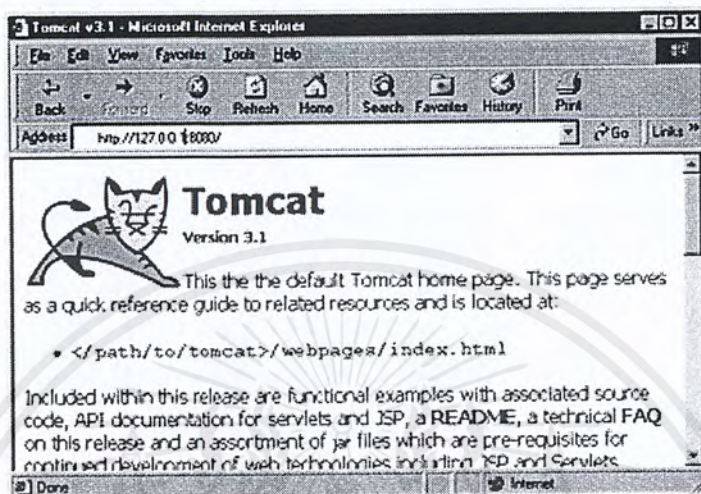
6. ทำการเซตค่า Environment Variable คือ JAVA_HOME เพื่อชี้ไปยังไดเรกทอรีของ JDK ซึ่งในที่นี้คือ C:\jdk1.3



รูปที่ ก-3 แสดงเซตค่า System Environment TOMCAT_HOME และ JAVA_HOME

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทำการทดสอบโดยเข้าไปในไดเรกทอรี C:\tomcat\bin จากนั้นให้รันคำสั่ง startup.exe หลังจากนั้นให้เปิด browser เพื่อเช็คว่า Tomcat ทำงานถูกต้องหรือเปล่า โดยไปที่ <http://127.0.0.1:8080> โดยให้เว็บเบราว์เซอร์ส่ง Request ไปที่เครื่องตัวเอง Local Host ที่พอร์ต 8080 ซึ่งเป็น Tomcat Default Port ถ้าจะปิด Tomcat ก็รันคำสั่ง shutdown.exe



รูปที่ ก-4 แสดงการรันเว็บเซิร์ฟเวอร์ Tomcat

8. ทำการแก้ไขไฟล์ server.xml ที่ \$TOMCAT_HOME/conf/server.xml ในส่วนของ <Context> ให้ชี้ไปที่ไดเรกทอรีที่เก็บเว็บแอปพลิเคชัน (Web Application) เช่น ถ้าเราเก็บเว็บแอปพลิเคชัน (Web Application) ไว้ที่ D:\Homepage ก็เพิ่มข้อความ <Context path="/Planny" docBase="D:/Homepage/" debug="0" reloadable="true"/> ในไฟล์ server.xml

```

59 <Parameter name="port" value="8007"/>
60 </Connector>
61
62 <!-- example - how to override AutoSetup actions -->
63 <Context path="/examples" docBase="webapps/examples" debug="0" reloadable="true" > </Context>
64 <!-- example - how to override AutoSetup actions -->
65 <Context path="" docBase="webapps/ROOT" debug="0" reloadable="true" ></Context>
66 <Context path="/test" docBase="webapps/test" debug="0" reloadable="true" ></Context>
67 <Context path="/planny" docBase="D:/Homepage" debug="0" reloadable="true" ></Context>
68
69 </ContextManager>
70 </Server>
71

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก-5 แสดงการแก้ไขไฟล์ *server.xml*

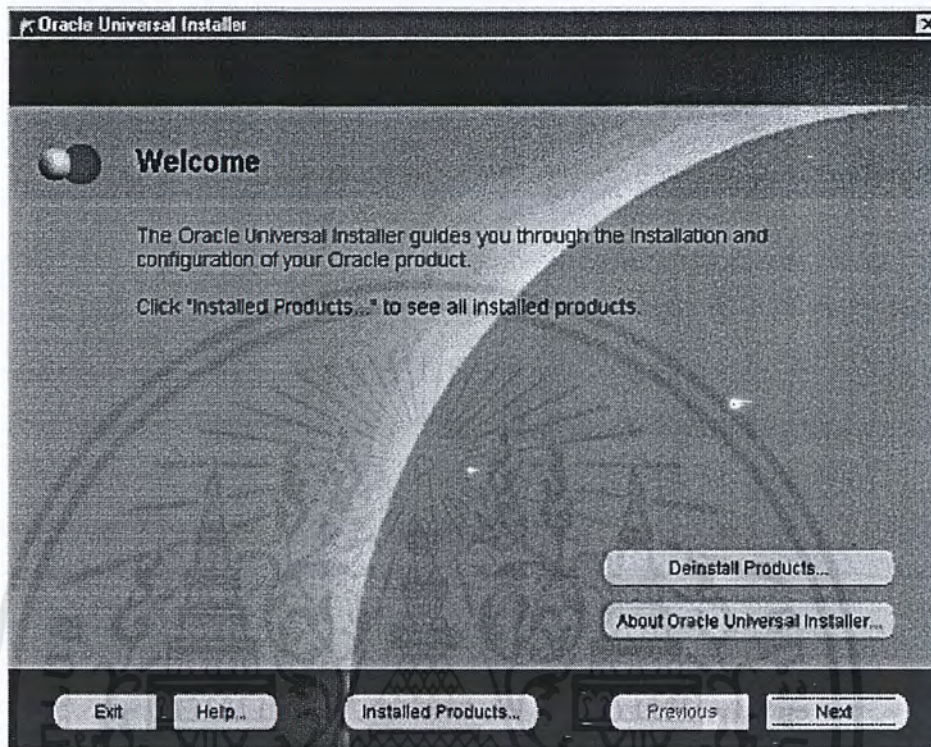
9. เพื่อให้เว็บเซิร์ฟเวอร์สามารถรองรับการทำงาน ของเอกสาร WML ที่อยู่บนเว็บเซิร์ฟเวอร์ได้ นั้น เราต้องมีการกำหนด MIME Type ของเอกสารบนเว็บเซิร์ฟเวอร์ซะก่อน เพราะในการสื่อสารระหว่าง ไคลเอนต์ กับเว็บเซิร์ฟเวอร์นั้น จะสื่อสารผ่าน HTTP โพรโทคอล ตัวเว็บเซิร์ฟเวอร์จะต้องบอกได้ว่าข้อมูลที่จะส่งไปให้กับไคลเอนต์นั้นเป็นข้อมูลชนิดใด โดยเราสามารถทำได้จากการกำหนด MIME Type นั้นเอง ในการกำหนด MIME Type ให้กับ Apache Tomcat เว็บเซิร์ฟเวอร์นั้น จะต้องไปทำการแก้ไขไฟล์ชื่อ *web.xml* อยู่ในไดเรกทอรี *TOMCAT_HOME/conf/* โดยเพิ่มข้อความดังต่อไปนี้ลงไป

```
<mime-mapping> <extension>wml</extension>
<mime-type>text/vnd.wap.wml</mime-type> </mime-mapping>
<mime-mapping> <extension>wbmp</extension>
<mime-type>image/vnd.wap.wbmp</mime-type> </mime-mapping>
<mime-mapping> <extension>wmlc</extension>
<mime-type>application/vnd.wap.wmlc</mime-type> </mime-mapping>
<mime-mapping> <extension>wmls</extension>
<mime-type>text/vnd.wap.wmlscript</mime-type> </mime-mapping>
<mime-mapping> <extension>wmlsc</extension>
<mime-type>application/vnd.wap.wmlsc</mime-type> </mime-mapping>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

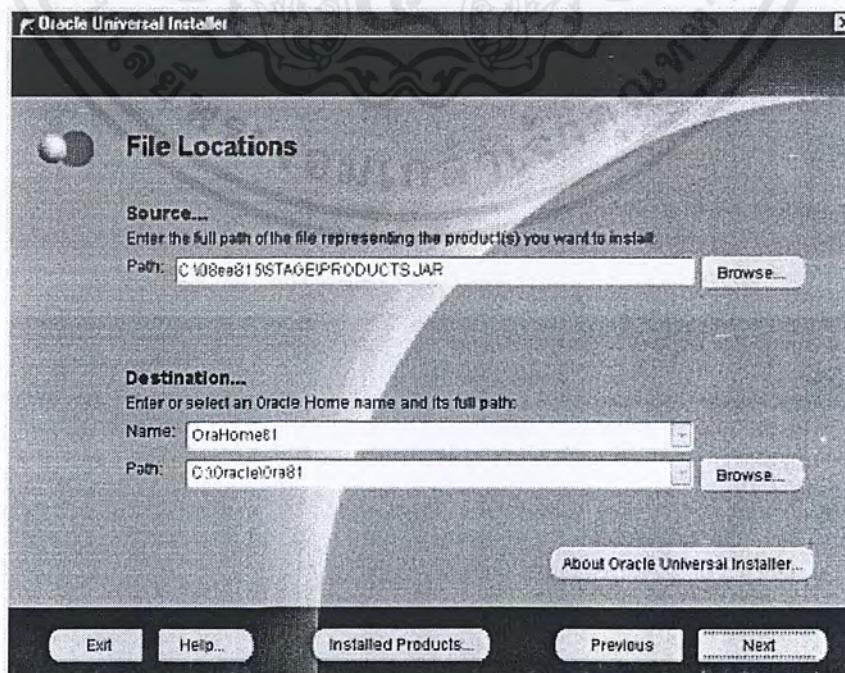
ข. ขั้นตอนการติดตั้งโปรแกรมออราเกิล 8I

1. นำแผ่น Oracle 8I ใส่ลงไปใน CD-ROM ไดรฟ์
2. เลือกเมนู Install/Deinstall Product จากนั้นจะปรากฏดังภาพ



รูปที่ ข-1 แสดงภาพการติดตั้งโปรแกรม Oracle Server ขั้นที่ 1

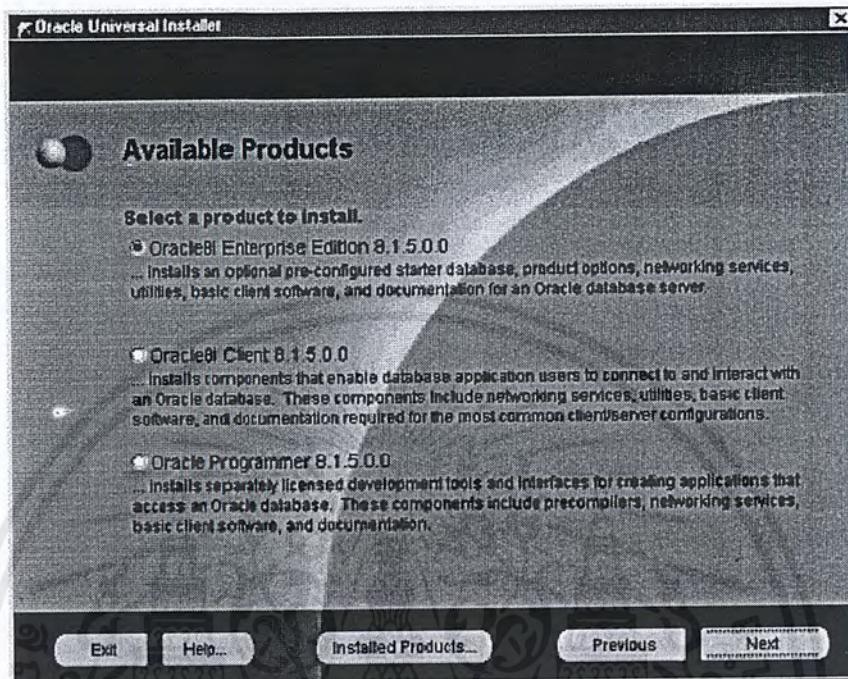
3. เลือกที่ปุ่ม Next จะปรากฏดังภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

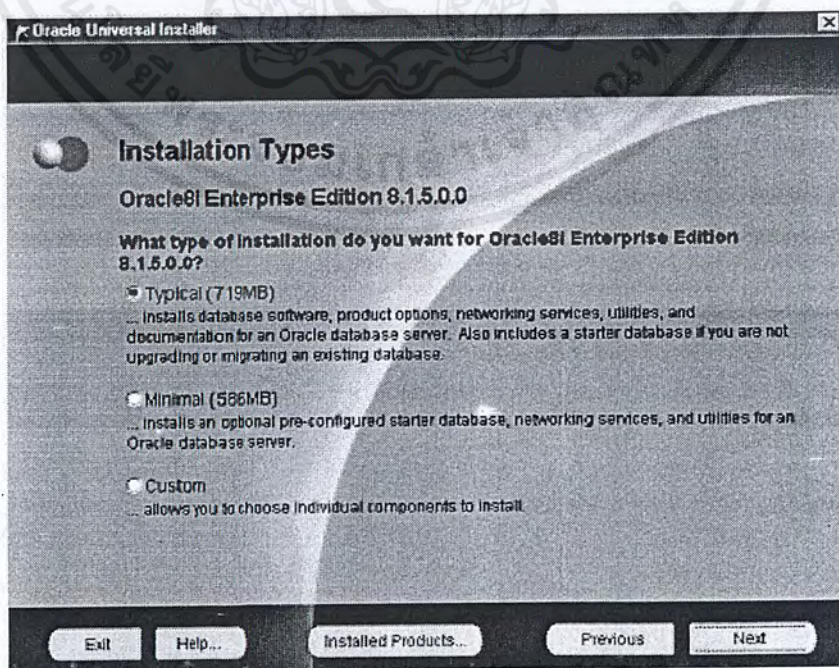
รูปที่ ข-2 แสดงภาพการติดตั้งโปรแกรม Oracle Server ขั้นที่ 2

4. ทำการเลือกชนิดของการติดตั้ง ในที่นี้เราจะติดตั้งให้เป็น Server ดังนั้นจึงเลือก Oracle8I Enterprise Edition 8.1.5.0.0 ดังรูป



รูปที่ ข-3 แสดงภาพการเลือกประเภทการติดตั้งโปรแกรม Oracle Server ขั้นที่ 3

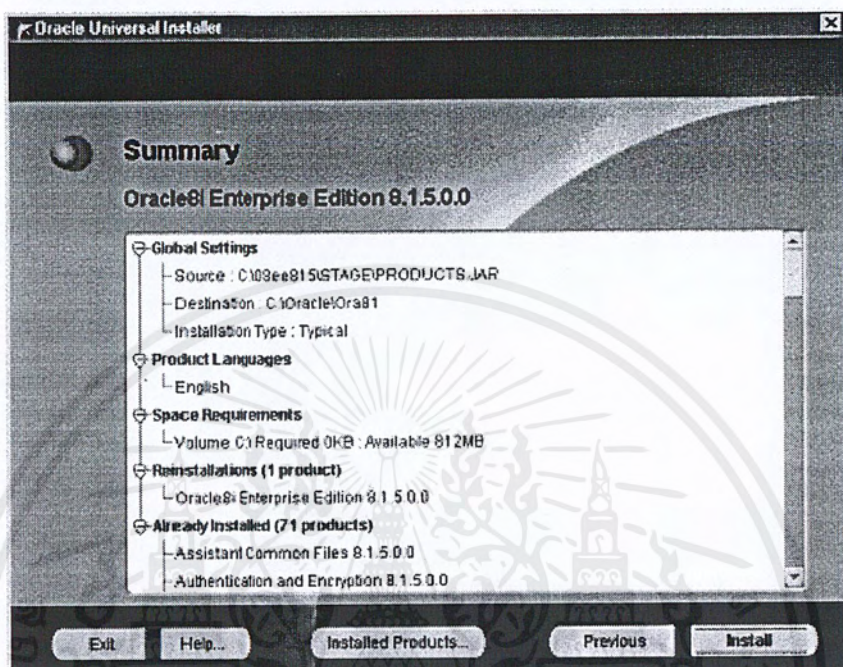
5. ทำการเลือกการประเภทของการติดตั้งว่าเป็นประเภท Typical, Minimal หรือ Custom ในที่นี้เราเลือก Typical ซึ่งจะให้เนื้อที่ประมาณ 719MB



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข-4 แสดงภาพติดตั้งโปรแกรม Oracle Server ขั้นที่ 4

6. เมื่อ กด Next จะติดตั้งโปรแกรม Oracle8i Enterprise Edition 8.1.5.0.0 จากนั้นแล้วทำการกด Next ไปเรื่อยๆ จนกว่าจะติดตั้งเสร็จเรียบร้อย



รูปที่ ข-5 ภาพแสดงการติดตั้ง Oracle Server ขั้นที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ทรงเกียรติ ภาวดี : “แกะรอย CGI เพื่อเขียนสคริปต์เรียกเพจเจอร์/มือถือผ่านเว็บ”, บริษัท วิตตี้ กรุ๊ป จำกัด , 1994
- [2] กิตติภูมิ วรฉัตร : “PHP เปลี่ยนวิธีสู่การสร้างโฮมเพจอย่างมือโปร”, บริษัท วิตตี้ กรุ๊ป จำกัด 1994
- [3] ดร.วีระศักดิ์ ชิงदारว :”Fundamental of JAVA Programming Volume 1” , SUM Publishing Department, SUM System Company Limited ,1998
- [4] ดร.วีระศักดิ์ ชิงदारว :”Fundamental of JAVA Programming Volume 2” , บริษัท ซีเอ็ดยูเคชั่น จำกัด ,2000
- [5] กิตติ ภัคดีวัฒนะกุล : “Java ฉบับโปรแกรมเมอร์” , หจก. ไทยเจริญการพิมพ์ ,1978
- [6] Steve Mann : “Programming applications with the wireless application protocol”, John Wiley & Sons, Inc , 2000
- [7] กฤษณะ สถิต : “Macromedia Dreamweaver2”, สำนักพิมพ์ อินโฟเพรส , 1999
- [8] James A. Senn :”ANALYSIS and DESIGN of INFORMATION SYSTEMS”, McGraw-Hill PUBLISHING COMPANY ,1989
- [9] กรภัทร์ สุทธิธรา :”Photoshop 5 Visual Guide Step by Step” , บริษัท ด้านสุทธการพิมพ์ จำกัด , 2521
- [10] นิรุช อำนวยศิลป์ :”สร้างเว็บเพจอย่างไรขีดจำกัด CGI&Perl เพื่อการประยุกต์ใช้งาน” , บริษัท ชักเชส มีเดีย จำกัด , 2521
- [11] พิชัย จันทร์จรัสทอง :”คู่มือ ORACLE 8” , แมกกรอ-ฮิล อินเทอร์เน็ต เนชั่นแนล เอ็นเตอร์ไพรส์ ,อิงค์ , 1999
- [12] อนุโชติ วุฒิพรพงษ์, พันธุ์เทพ แก้วมงคล :”สร้าง WP ด้วย WML Script” , สำนักพิมพ์ อินโฟเพรส 2543
- [13] Jason Hunter with Silliam Crawford :”Java Servlet Programming” , O’Reilly & Associates,Inc , 1998
- [14] Karl Moss :”Java Servlets”, The McGraw-Hill, Companies,Inc , 1999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้