

การนัดหมายดำเนินงานผ่านเว็บ  
ORGANIZING A TEAMWORK SCHEDULE ON WAP



นางสาวจินตนา เดียวแสงสุข  
นายชัยพร ศรีทอง

เลขที่.....  
เลขทะเบียน 42790  
วัน, เดือน, ปี 10 ส.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14/2/2547

การนัดหมายดำเนินงานผ่านเวิร์ฟ

ORGANIZING A TEAMWORK SCHEDULE ON WAP



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2543

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

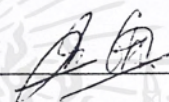
เรื่อง การนัดหมายดำเนินงานผ่านเว็บ

Organizing a Teamwork Schedule on WAP

ผู้จัดทำ

1. น.ส.จินตนา เตียวแสงสุข รหัสประจำตัว 40010128
2. นายชัยพร ศรีทอง รหัสประจำตัว 40010175

ปรึกษา



อาจารย์ที่

(ดร.วิศิษฐ์ หิรัญกิติ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การนัดหมายดำเนินงานผ่านเว็บ

นางสาวจินตนา เดียวแสงสุข 40010128  
นายชัยพร ศรีทอง 40010175  
ดร. วิศิษฎ์ หิรัญกิตติ อาจารย์ที่ปรึกษา  
ปีการศึกษา 2543

### บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อใช้ศึกษาวิธีการสร้าง และการออกแบบ การให้บริการนัดหมายการทำงานผ่าน Wireless Application Protocol ( WAP ) โดยใช้ Java Server Page ( JSP ) เป็นตัวให้บริการ การนัดหมายดำเนินงานผ่าน WAP ถูกออกแบบมาเพื่อให้เหมาะสมกับการให้บริการการนัดหมาย เพราะโทรศัพท์มือถือเป็นอุปกรณ์ที่มีขนาดเล็ก ผู้ใช้มีความสะดวกในการพกพา การนัดหมายดำเนินงานนี้มีความสามารถให้บริการบันทึกตารางเวลาของผู้ใช้เหมือนเป็นสมุดจดตารางเวลาส่วนตัว บริการจัดกลุ่มสมาชิก เพื่อให้ผู้ใช้บริการผู้อื่นมาอยู่ในกลุ่มการทำงานเดียวกัน นำไปสู่บริการนัดหมายกลุ่มสมาชิก เพื่อเลือกช่วงเวลาที่เหมาะสมกับการทำงานของทุกคน ถ้าไม่สามารถนัดหมายได้ทุกคนก็จะมีการพยายามร้องขอเลื่อนเวลาของสมาชิกเพื่อที่จะนัดหมายทำได้ในที่สุด เมื่อใกล้ถึงเวลานัดหมายก็จะมีการแจ้งเตือนผ่านบริการ Shot Message Service ( SMS ) เพื่อให้ผู้ใช้บริการได้ทราบล่วงหน้า

## Organizing a Teamwork Schedule on WAP

Jintana Teaouseangsuk

Chaiporn Sritong

Dr. Visit Hirankitti      Advisor

### ABSTRACT

This project is aimed at studying and developing an appointment service on Wireless Application Protocol ( WAP ) using Java Server Page ( JSP ) .

Organizing a teamwork schedule on WAP has to be specially designed because the mobile telephones are small and portable devices. This WAP appointment service is able to provide the users the ability to enter his/her personal schedule and also provide teamwork arrangement service for the users who work in the same group. The latter service performs a teamwork appointment in order to select the appropriate time for an appointment of all members. If the time does not fit for some members then it tries to find a new time for the appointment. When the appointment time is approaching, it will remind the users in advance via Shot Message Service ( SMS ) on their mobile telephones.

### กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ วิศิษฐ์ หิรัญกิตติ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

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

นางสาวจินตนา เตียวแสงสุข  
นายชัยพร ศรีทอง

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 Wireless Application Protocol	4
2.1 บทนำ	4
2.2 เหตุผลที่ใช้เทคโนโลยีที่ใกล้เคียงกับรูปแบบของอินเทอร์เน็ต	5
2.3 ความเป็นมาของเว็บ	5
2.4 รูปแบบอินเทอร์เน็ต ( The Internet Model )	5
2.5 รูปแบบเว็บ ( The WAP Model )	6
2.6 สถาปัตยกรรมของเว็บ ( WAP Architecture )	7
2.6.1 สิ่งแวดล้อมในการใช้งานเว็บ ( WAP Application Environment )	8
2.6.2 เว็บโพรโทคอล ( WAP Protocol )	10
บทที่ 3 Wireless Markup Language	13
3.1 บทนำ	13
3.2 พื้นฐาน WML	13
3.3 เด็ค ( Deck )	15
3.4 การ์ด ( Cards )	15
3.5 เนื้อหา ( Content )	16
3.6 การลิงก์ ( Link )	16
3.7 ตั้งเวลาของการ์ดด้วย ontimer	17
3.8 แสดงรูปภาพฟิก <image>	18
3.9 รับข้อมูลด้วย <input>	18
3.10 เลือกข้อมูลด้วย <select>	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11	สรุปโดยรวมของ WML	19
บทที่ 4	Servlet และ JavaServer Pages	21
4.1	บทนำ	21
4.2	อะไรคือ Servlets	22
4.3	ทำไมจึงต้องใช้ Servlet	22
4.4	ความเป็นมาของ JavaServer Pages	22
4.4.1	Active Server Pages	24
4.4.2	Servlets และ JavaServer Pages	24
4.5	JSPs แบบง่าย ( Simple JSPs )	25
4.5.1	คำอธิบาย ( Comments )	26
4.5.2	Templating	26
4.5.3	การรวมไฟล์	27
4.5.4	ข้อผิดพลาด ( Errors )	27
4.6	Request-Time Expressions	28
4.6.1	Expressions	28
4.6.2	The Implicit Objects	28
4.6.3	A First Application	30
4.7	การใช้ Beans	31
4.7.1	อะไรคือ Beans	31
4.7.2	การใช้ Beans	32
4.7.3	การรับคุณสมบัติ ( Getting a Property )	33
4.7.4	การตั้งคุณสมบัติ ( Setting Properties )	33
4.7.5	การเชื่อมต่อระหว่างฟอร์มกับคุณสมบัติ	34
4.8	Bean Scopes	35
4.8.1	ขอบเขตเพจ ( The Page Scope )	35
4.8.2	ขอบเขตการร้องขอ ( The Request Scope )	35
บทที่ 5	JSP Sessions	40
5.1	บทนำ	40
5.2	การทำงานของ HTTP	40
5.2.1	การเชื่อมต่อแบบฝังแน่น ( Persistent Connection )	41
5.3	เทคนิคสำหรับการรักษาข้อมูลของผู้ใช้ข้ามเพจ	41
5.3.1	ข้อมูล HTTP ( HTTP Information )	41
5.3.2	Hidden Fields	41
5.3.3	Cookies	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 การทำงานกับ cookies และ Java	42
5.5 Sessions	44
5.5.1 อะไรคือ Session	45
5.5.2 Servlet สนับสนุน session และ วงจรชีวิต session	45
5.6 วงจรชีวิต Session	45
5.6.1 Contexts	47
<b>บทที่ 6 การวิเคราะห์และการออกแบบ</b>	<b>50</b>
6.1 ความต้องการของระบบ	50
6.1.1 หลักการ	50
6.1.2 ขอบเขตการให้บริการ	50
6.1.3 กิจกรรมต่างๆ	50
6.1.4 การลงทะเบียนสมาชิกใหม่	50
6.1.5 การเข้าสู่ระบบ	50
6.1.6 การรับส่งข้อความ	51
6.1.7 การจัดตารางเวลา	51
6.1.8 การจัดกลุ่มสมาชิก	51
6.1.9 การนัดหมายกลุ่มสมาชิก	51
6.1.10 การแจ้งเตือน	54
6.1.11 การเปลี่ยนแปลงข้อมูลของตัวเอง	54
6.1.12 การออกจากระบบ	54
6.2 การวิเคราะห์และออกแบบ	54
6.2.1 ER diagram	54
6.2.2 Table	56
6.3 การเตรียมพร้อมข้อมูลที่ไม่ถูกต้อง	57
6.4 สถาปัตยกรรมของระบบ ( System Architecture )	57
6.4.1 เว็บเซิร์ฟเวอร์	57
6.4.2 Database Server	58
6.4.3 Web programming program	58
6.4.4 WML Browser	58
<b>บทที่ 7 การใช้งานโปรแกรม</b>	<b>60</b>
7.1 การทำงานฝั่ง Server	60
7.1.1 การทำงานของ MySQL	60
7.1.2 การทำงานของ Apache Tomcat Web Server	60
7.1.3 การทำงานของโปรแกรมฉากหลัง	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 การทำงานฝั่ง Client	61
7.2.1 การลงทะเบียนสมาชิกใหม่	61
7.2.2 การเข้าสู่ระบบ	61
7.2.3 การรับส่งข้อความ ( Message )	62
7.2.4 การจัดการตารางเวลา ( Schedule )	64
7.2.5 การจัดกลุ่มสมาชิก ( Team Working )	67
7.2.6 การจัดการการนัดหมาย ( Appointment )	70
7.2.7 การเก็บข้อมูลส่วนตัวของผู้ใช้ ( Account Information )	73
บทที่ 8 การตรวจสอบการทำงานของฟังก์ชัน	75
8.1 การตรวจสอบการทำงานของฟังก์ชัน ( Functional Testing )	75
8.2 จุดประสงค์การทดสอบ	75
8.3 วิธีการทดสอบ	75
8.4 ผลการทดสอบ	76
8.5 สรุปผลการทดสอบ	76
บทที่ 9 บทวิจารณ์และสรุป	77
9.1 สรุปผลเปรียบเทียบกับเป้าหมายและวัตถุประสงค์	77
9.2 แนวทางการพัฒนาต่อ	77
ภาคผนวก	78
ภาคผนวก ก การติดตั้งโปรแกรม	79
ภาคผนวก ข TOOL ที่ใช้พัฒนาในโครงการ	83
บรรณานุกรม	84

## สารบัญตาราง

	หน้า
ตารางที่ 3-1 แสดงค่าประมาณของหน่วยความจำในอุปกรณ์เว็บ	16
ตารางที่ 4-1 ตัวอย่างเมธอดในออบเจกต์ request	29
ตารางที่ 5-1 เมธอดของอินเทอร์เฟซ HTTPServletRequest	45
ตารางที่ 5-2 ตัวอย่างวิธีการเก็บข้อมูลและอ่านข้อมูลจาก session	47
ตารางที่ 8-1 ผลการทดสอบการทำงานของฟังก์ชัน	76



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญภาพ

	หน้า
รูปที่ 2-1 รูปแบบอินเตอร์เน็ต	6
รูปที่ 2-2 รูปแบบเว็บ	7
รูปที่ 2-3 ลำดับชั้นของเว็บ	8
รูปที่ 4-1 การไหลของข้อมูลของเพจที่มีการใช้ Servlet	25
รูปที่ 4-2 รูปแบบของ Java beans	32
รูปที่ 5-1 วงจรชีวิตของ session	46
รูปที่ 6-1 ER Diagram ของระบบการนัดหมายดำเนินงานผ่านเว็บ	55
รูปที่ 6-2 Nokia WAP Toolkit version 2.0	62
รูปที่ 7-1 แสดงหน้า Login Page	61
รูปที่ 7-2 แสดงขั้นตอนการลงทะเบียนเพื่อเข้าสู่ระบบ	61
รูปที่ 7-3 แสดงขั้นตอนการเข้าสู่ระบบ	62
รูปที่ 7-4 แสดงหน้าหลักของการใช้โปรแกรม	62
รูปที่ 7-5 แสดงหน้าจอ Message	63
รูปที่ 7-6 การดูข้อความในบริการ Message	63
รูปที่ 7-7 แสดงขั้นตอนการส่งข้อความในบริการ Message	63
รูปที่ 7-8 แสดงขั้นตอนการลบข้อความในบริการ Message	64
รูปที่ 7-9 แสดงหน้าจอ Schedule	64
รูปที่ 7-10 แสดงขั้นตอนการดูตารางเวลาในบริการ Schedule	65
รูปที่ 7-11 แสดงขั้นตอนการเพิ่มหมายกำหนดการในบริการ Schedule	66
รูปที่ 7-12 แสดงขั้นตอนการลบหมายกำหนดการ ในบริการ Schedule	66
รูปที่ 7-13 แสดงหน้าจอ teamwork	67
รูปที่ 7-14 แสดงการขอลูกกลุ่มในบริการ teamwork	68
รูปที่ 7-15 แสดงขั้นตอนการเพิ่มกลุ่มในบริการ teamwork	68
รูปที่ 7-16 แสดงขั้นตอนการลบกลุ่มในบริการ teamwork	69
รูปที่ 7-17 แสดงขั้นตอนการขอเป็นสมาชิกในกลุ่มของบริการ teamwork	69
รูปที่ 7-18 แสดงขั้นตอนการขอออกจากกลุ่มของบริการ teamwork	70
รูปที่ 7-19 แสดงการดูรายชื่อกลุ่มที่ผู้ใช้เป็นสมาชิก	70
รูปที่ 7-20 แสดงหน้าจอ Appointment	71
รูปที่ 7-21 แสดงขั้นตอนการสร้างการนัดหมาย	71
รูปที่ 7-22 แสดงการเลือกวิธีการนัดหมาย	72
รูปที่ 7-23 แสดงขั้นตอนการดูการนัดหมาย	72
รูปที่ 7-24 แสดงขั้นตอนการรับการนัดหมาย	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7-25 แสดงหน้าจอ Requested Appointment	73
รูปที่ 7-26 แสดงหน้าจอ Information ในบริการ Account Information	74
รูปที่ ข-1 Nokia WAP Toolkit	83



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

การติดต่อสื่อสารจำเป็นสำหรับการนัดหมายการทำงานเป็นกลุ่ม โดยเฉพาะอย่างยิ่งถ้าบุคคลในกลุ่มนั้นอยู่ในสถานที่ที่ห่างไกลกัน เพื่อเพิ่มความสะดวกรวดเร็วในการนัดหมายทำงานร่วมกัน จึงนัดหมายผ่านการใช้โทรศัพท์มือถือที่อยู่บนพื้นฐานเว็บ ( WAP : Wireless Application Protocol ) ไม่เพียงแต่ใช้บริการโทรศัพท์ได้เท่านั้นแต่รวมถึงบริการเสริมด้วย เช่น บริการบันทึกการนัดหมายส่วนตัวของผู้ใช้เหมือนเป็นสมุดจดตารางเวลาส่วนตัว บริการจัดกลุ่มสมาชิก และบริการนัดหมายกลุ่มสมาชิก เพื่อเลือกช่วงเวลาที่เหมาะสมกับการทำงานของทุกคน เป็นต้น

เหตุผลที่เลือกใช้โทรศัพท์มือถือเป็นอุปกรณ์ในการนัดหมายเพราะ โทรศัพท์มือถือเป็นเครื่องมือสื่อสารที่พกพาสะดวกและน้ำหนักเบา อีกทั้งยังได้รับความนิยมเป็นอย่างมาก โทรศัพท์มือถือจึงเป็นเป้าหมายทางการตลาดและทางการพัฒนาอีกด้วย ดังนั้นจึงมีความพยายามที่จะเพิ่มบริการเสริมต่างๆให้แก่โทรศัพท์มือถือ

และอินเทอร์เน็ต ( Internet ) ก็เป็นอีกบริการหนึ่งที่มีความสนใจ เนื่องจากอินเทอร์เน็ตได้เข้ามามีบทบาทต่อการดำรงชีวิตของผู้คนจำนวนมาก ทั้งการติดต่อสื่อสารส่วนตัวและด้านธุรกิจ แต่อินเทอร์เน็ตบนคอมพิวเตอร์นั้นขาดความคล่องตัวในการเชื่อมต่อเข้าไปใช้งานเพราะต้องหาอุปกรณ์ร่วมมาใช้งานด้วย ดังนั้นถ้าหากสามารถใช้บริการของอินเทอร์เน็ตผ่านโทรศัพท์มือถือได้ จะมีความคล่องตัวในการใช้บริการมากกว่า และไม่มีข้อจำกัดเรื่องสถานที่ สามารถเข้าถึงข้อมูลที่ต้องการได้ตลอดเวลา ดังนั้นจึงเกิดแว่พขึ้นมา เพื่อนำแนวคิดของอินเทอร์เน็ตมาเป็นรูปแบบของการให้บริการบนโทรศัพท์มือถือ

การบันทึกการนัดหมายส่วนตัวโดยทั่วไปในทุกวันนี้ ใช้วิธีจดในสมุดบันทึก, ในคอมพิวเตอร์มือถือ หรือแม้กระทั่งในเซิร์ฟเวอร์ ( Server ) ซึ่งใช้ผ่านเว็บ ( Web ) แต่วิธีการเหล่านี้ยังมีข้อด้อยบางจุด เช่น ถ้าใช้สมุดบันทึก การแก้ไขการนัดหมายอาจไม่สวยงาม ถ้าใช้คอมพิวเตอร์มือถือ การนัดหมายทำงานร่วมกับผู้อื่นอาจไม่สะดวก หรือถ้าใช้บันทึกในเซิร์ฟเวอร์ที่ต้องใช้คอมพิวเตอร์ในการเข้าเว็บ ก็จะไม่สะดวกในการพกพา ดังนั้นการนัดหมายดำเนินงานผ่านเว็บเป็นวิธีการในรูปแบบใหม่ในการนัดหมาย โดยใช้อินเทอร์เน็ตเป็นสื่อกลางในการนัดหมาย ผู้ใช้เพียงใช้อุปกรณ์ที่มีเว็บเบราว์เซอร์ ( WAP Browser ) ก็จะมาอยู่ในระบบ ที่สามารถใช้บริการนัดหมายดำเนินงานกับผู้อื่นได้โดยสะดวก ซึ่งการนัดหมายดำเนินงานผ่านเว็บมีข้อดีดังนี้

#### - การพกพา

เนื่องจากอุปกรณ์ที่ใช้เว็บเป็นอุปกรณ์ที่มีขนาดเล็ก การพกพาก็สะดวกสบายไปด้วย

#### - บริการ

การให้บริการสามารถเพิ่มความสามารถของการบริการได้ไม่มีที่สิ้นสุดเพราะเป็นบริการที่เซิร์ฟเวอร์ ดังนั้นการปรับปรุงบริการจึงทำที่เซิร์ฟเวอร์ทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## - การดูแลรักษาข้อมูล

เนื่องจากข้อมูลอยู่ที่เซิร์ฟเวอร์ ดังนั้นเราสามารถดูข้อมูลได้จากทุกเครื่องโดยไม่จำกัดว่าต้องเป็นเครื่องตัวเองเท่านั้นที่มีข้อมูลเช่น โทรศัพท์เคลื่อนที่ในปัจจุบัน

### 1.2 วัตถุประสงค์ของงานวิจัย

ศึกษาเทคโนโลยีเว็บ ให้สามารถเขียนโปรแกรมในการพัฒนาบริการบนโทรศัพท์มือถือ โดยบริการนั้นจะต้องเป็นไดนามิกเพจ (dynamic page) ที่มีโปรแกรมทางฝั่งเซิร์ฟเวอร์ และศึกษา JSP เพื่อนำมาใช้เป็นตัวให้บริการ และเขียนโปรแกรมทางฝั่งเซิร์ฟเวอร์ด้วยภาษาจาวา (Java)

ศึกษาการสร้างบริการที่มีความชาญฉลาด ตั้งแต่วิธีการคิดและการออกแบบ คิดหาอัลกอริทึม (algorithm) ที่จะทำให้บริการมีวิธีการที่คล้ายมนุษย์ได้

นำเสนอแนวคิด และศึกษาความเป็นไปได้ในการสร้างระบบการนัดหมายดำเนินงานเป็นกลุ่มบนโทรศัพท์มือถือ เพื่อเป็นแนวทางในการสร้างบริการใหม่ ๆ บนโทรศัพท์มือถือที่อยู่บนพื้นฐานเว็บ สามารถสร้างบริการของระบบการนัดหมายดำเนินงานผ่านเว็บ ได้แก่ การจัดการนัดหมายสมาชิกในกลุ่มอย่างชาญฉลาด การส่งข้อความแก่สมาชิก และการบันทึกการนัดหมายส่วนตัวของผู้ใช้ เป็นต้น

### 1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะสร้างระบบการนัดหมายดำเนินงานผ่านเว็บ โดยระบบนี้จะออกแบบให้อยู่บนเซิร์ฟเวอร์ที่ชื่อว่า Tomcat จะทำการสร้างบริการที่เกี่ยวข้องกับการรับส่งข้อมูลระหว่างบุคคลที่มาสมัครเป็นสมาชิก แต่ไม่คิดรวมบริการโทรศัพท์ (telephony services)

งานวิจัยนี้จะทำการสร้างระบบขึ้นมาใหม่ทั้งหมด ตั้งแต่การเขียนโปรแกรม การติดตั้งบนเซิร์ฟเวอร์ Tomcat การให้บริการ รวมไปถึงการสร้างฐานข้อมูลที่ใช้ในระบบนี้ด้วย

### 1.4 วิธีการดำเนินงาน

1. ศึกษา ค้นคว้า รวบรวมความรู้พื้นฐานเกี่ยวกับเว็บ
2. ศึกษาภาษา WML, เครื่องมือที่ใช้ในการเขียนและแสดงผลของภาษา WML, และฝึกการเขียนภาษา WML
3. ศึกษาการทำงานของเว็บเซิร์ฟเวอร์ (Web Server) ที่ชื่อว่า TOMCAT และทดลองการร้องขอเอกสาร WML จากอินเทอร์เน็ต
4. ศึกษาการทำเอกสาร WML แบบไดนามิกด้วย Servlet ทดลองเขียน Servlet บน TOMCAT
5. ศึกษาการเขียน JSP (Java Server Page)
6. ศึกษาอัลกอริทึมในการนัดหมายดำเนินงาน ซึ่งจะต้องทำให้บริการมีความชาญฉลาด สามารถมีวิธีการคิดที่คล้ายมนุษย์
7. ศึกษาการจัดการฐานข้อมูล MySQL เพื่อสร้างฐานข้อมูลให้แก่ระบบ
8. ออกแบบระบบการนัดหมายดำเนินงานผ่านเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เขียน โปรแกรม JSP เพื่อให้บริการการนัดหมายดำเนินงานผ่านเว็บ
10. ทดสอบการทำงานของระบบ และแก้ไขปัญหาที่เกิดขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### Wireless Application Protocol

#### 2.1 บทนำ

เครือข่ายไร้สายของทุกวันนี้ ไม่ได้เตรียมการสำหรับรองรับการให้บริการที่ค่อนข้างซับซ้อน ดังนั้นจึงเกิดเว็บขึ้นมาเพื่อนำแนวคิดของอินเทอร์เน็ตมาเป็นรูปแบบของการให้บริการ

ในปี ค.ศ. 1997 บริษัทอีริคสัน (Ericsson), โมโตโรลา (Motorola), โนเกีย (Nokia), และ อันไวร์พลาเน็ต (Unwired Planet) ได้ร่วมมือกันจัดตั้งเว็บฟอรัม (WAP Forum) ที่ทำหน้าที่นำความสะดวกสบายของอินเทอร์เน็ตไปสู่การให้บริการของเครือข่ายไร้สาย ในปัจจุบันมีมากกว่า 90 บริษัทชั้นนำของโลกในธุรกิจการติดต่อสื่อสารไร้สายที่เข้ามาเป็นสมาชิก

ลักษณะสำคัญที่เสนอโดยเว็บ คือ

- รูปแบบการโปรแกรมคล้ายอินเทอร์เน็ต  
การนำแนวคิดของอินเทอร์เน็ตมาใช้ใหม่ในเว็บ ทำให้สามารถแนะนำบริการพื้นฐานเว็บ ได้อย่างรวดเร็ว เพราะทั้งผู้พัฒนาบริการและผู้ผลิตคุ้นเคยกับแนวคิดเหล่านี้คืออยู่แล้ว
- WML (Wireless Markup Language)  
ภาษาที่ใช้อธิบายภาษาอื่น (Markup language) ใช้สำหรับบริการผู้เขียนเหมือนกับภาษา HTML (HyperText Markup Language) ที่ใช้บนเว็บ แต่ WML นี้ถูกออกแบบให้เหมาะกับอุปกรณ์พกพาขนาดเล็ก
- WMLScript  
WMLScript ใช้เพิ่มฟังก์ชันการทำงานของบริการ เหมือนกับจาวาสคริปต์ (Java Script) ที่ใช้ใน HTML ทำให้เพิ่มความสามารถในการทำงานลอจิกและฟังก์ชันการคำนวณให้แก่บริการบนเว็บ
- WTA (Wireless Telephony Application)  
WTA framework คือชุดของลักษณะสำคัญที่เตรียมวิธีทางสู่การสร้างบริการโทรศัพท์ ที่สำคัญต้องมีการแนะนำอินเตอร์เฟซ (interface) จากไคลเอนต์ (client) สู่เครือข่ายไร้สาย มีการใช้บริการในลักษณะเรียลไทม์ (real-time) และมีเครื่องกล (mechanism) ที่สนับสนุนเซิร์ฟเวอร์ที่เป็นจุดเริ่มให้บริการ
- สแต็คของโพรโตคอลที่ถูกทำให้กระชับขึ้น (Optimised protocol stack)  
โพรโตคอลที่ใช้ในเว็บ อยู่ในพื้นฐานโพรโตคอลอินเทอร์เน็ตที่รู้จักกันดี เช่น HTTP (HyperText Transport Protocol) และ TCP (Transmission Control Protocol) แต่ถูกลดขนาดเพื่อบังคับให้เหมาะสมกับสิ่งแวดล้อมของอุปกรณ์ไร้สาย เช่น ความกว้างแถบความถี่ (bandwidth) ต่ำและการชอนอยู่ภายใน (latency) สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 เหตุผลที่ใช้เทคโนโลยีที่ใกล้เคียงกับรูปแบบของอินเทอร์เน็ต

อินเทอร์เน็ตและเว็บถูกใช้อย่างกว้างขวาง ไม่เพียงแต่การบริการของมันที่ดึงดูดเรา แต่รวมถึงทางที่สะดวกสบายของการเข้าถึงมันผ่านทางเบราว์เซอร์ ( browser ) เราสามารถเข้าถึงบริการเดียวกันทั่วทั้งโลก ราวเท่าที่เราเข้าถึงคอมพิวเตอร์และเครือข่ายได้

การให้บริการอินเทอร์เน็ตในโทรศัพท์มือถือโดยใช้เทคโนโลยีเว็บ จะเป็นการลดข้อจำกัดที่ต้องใช้เครื่องคอมพิวเตอร์, โมเด็ม ( modem ) ทำให้เราสามารถเข้าอินเทอร์เน็ตได้โดยใช้แค่โทรศัพท์มือถือเพียงอย่างเดียว

นอกจากนี้ผู้จัดหาบริการยังได้ผลประโยชน์จากการใช้รูปแบบอินเทอร์เน็ต นั่นคือ บริการเป็นอิสระกับตำแหน่งของผู้ใช้ บริการถูกสร้างและเก็บบันทึกไว้บนเซิร์ฟเวอร์ หมายความว่ามันง่ายมากที่จะเปลี่ยนแปลงบริการเหล่านั้นให้ตรงตามความต้องการของลูกค้า บริการถูกสร้างขึ้นมาได้อย่างรวดเร็ว เพราะสะดวกสบายต่อการดำเนินโครงการ ทำให้ลดเวลาในการพัฒนาบริการ และลดราคาลงเมื่อเปรียบเทียบกับการพัฒนาบริการตามแบบธรรมดา

## 2.3 ความเป็นมาของเว็บ

เว็บเป็นผลของการพยายามร่วมกันโดยการรวมกลุ่มบริษัทที่เรียกว่าเว็บฟอร์มมีวัตถุประสงค์ คือ การสร้างมาตรฐานร่วมกันที่ไม่คิดค่าลิขสิทธิ์ มาตรฐานในการรับส่งข้อมูลและบริการ โทรศัพท์ในอุปกรณ์ไร้สาย โดยการเข้าถึงบริการเว็บ ใช้รูปแบบอินเทอร์เน็ต

ในเดือนมิถุนายน ปี 1997 อีริคสัน, โมโตโรลา, โนเกียและอันไรว์พลาเน็ต ได้เริ่มต้นสร้างมาตรฐานอย่างรวดเร็วสำหรับบริการประยุกต์ภายใต้โดเมนของอุปกรณ์ไร้สาย ในเดือนธันวาคม ปี 1997 เว็บฟอร์มถูกสร้างขึ้นอย่างเป็นทางการ ในเดือนเมษายน ปี 1998 ได้ปล่อยรายละเอียดของ เว็บเวอร์ชัน 1.0 ( WAP 1.0 Specifications ) หลังจากนั้น เว็บฟอร์มถูกเปิดออกสู่ทุกคน

วัตถุประสงค์หลักของเว็บฟอร์มคือ

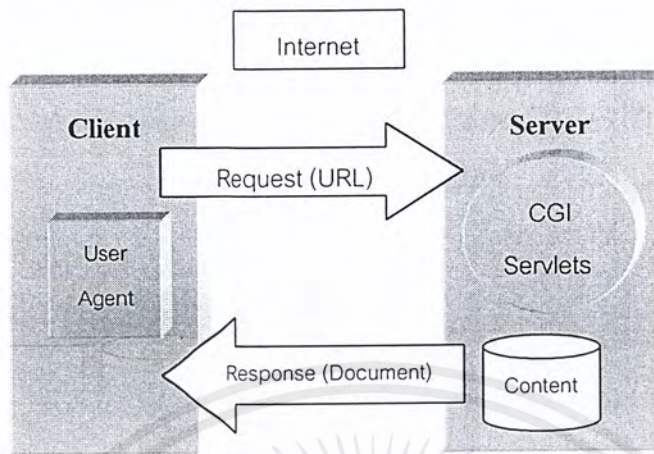
- มาตรฐานเครือข่ายไร้สาย ( wireless network standard ) มีความเป็นอิสระ ไม่ขึ้นอยู่กับใคร
- เปิดสู่ทุกคน
- จะเป็นข้อเสนอสู่ตัวมาตรฐานที่เป็นที่พอใจ
- แอปพลิเคชันมีการส่ง ( transport ) หลายทางเลือก เช่น GSM , IS-95 , IS-136 , PDC เป็นต้น
- แอปพลิเคชันใช้ข้ามชนิดอุปกรณ์ได้ เช่น โทรศัพท์เคลื่อนที่ , PDA เป็นต้น
- ขยายความสามารถได้ นอกเหนือจากเวลากำหนด เพื่อเครือข่ายใหม่ๆ และ การส่ง เช่น ระบบ 3 G

## 2.4 รูปแบบอินเทอร์เน็ต ( The Internet Model )

รูปแบบอินเทอร์เน็ตนั้นให้โคลเอนต์ใช้บริการที่มีอยู่มากมายบนเซิร์ฟเวอร์โดยมีแอดเดรส ( address ) ต่างกันที่ระบุโดย URL ( Uniform Resource Locator ) ที่ไม่ซ้ำกัน เนื้อหาที่เก็บไว้บนเซิร์ฟเวอร์เก็บไว้หลายรูปแบบ โดยที่ HTML จะมีมากที่สุด HTML จะอธิบายสิ่งที่จะปรากฏให้เห็นของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริการในโครงสร้างเอกสารเรียบ ๆ ถ้ามีลักษณะการใช้การคำนวณลอจิกจะจำเป็นต้องใช้ภาษาสคริปต์ เช่น จาวาสคริปต์หรือ วิวอลเบสึคสคริปต์



รูปที่ 2-1 รูปแบบอินเทอร์เน็ต

ในรูปที่ 2-1 แสดงถึงวิธีที่ไคลเอนต์ร้องขอทรัพยากรที่เก็บไว้บนเว็บเซิร์ฟเวอร์ ซึ่งถูกระบุโดย URL โดยในรูปทรัพยากรคือเอกสาร HTML

เนื้อเรื่อง ( content ) อาจเป็นแบบคงที่ ( static ) หรือแบบที่เปลี่ยนแปลงเสมอ ( dynamic ) เนื้อเรื่องแบบคงที่จะถูกสร้างขึ้นมาเพียงครั้งเดียวและไม่เปลี่ยนแปลงบ่อยนัก เช่น การแสดงตัวของบริษัท ส่วนเนื้อเรื่องแบบที่เปลี่ยนแปลงเสมอ จำเป็นใช้เมื่อข้อมูลมีการเปลี่ยนแปลงบ่อย เช่น ตารางเวลา, ข่าว, การแจ้งราคาหุ้น และข้อมูลบัญชี เทคโนโลยีบางตัว เช่น ASP ( Active Server Pages ), CGI ( Common Gateway Interface ) และ Servlets ทำให้เนื้อเรื่องถูกสร้างแบบเปลี่ยนแปลงเสมอได้

## 2.5 รูปแบบเว็บ ( The WAP Model )

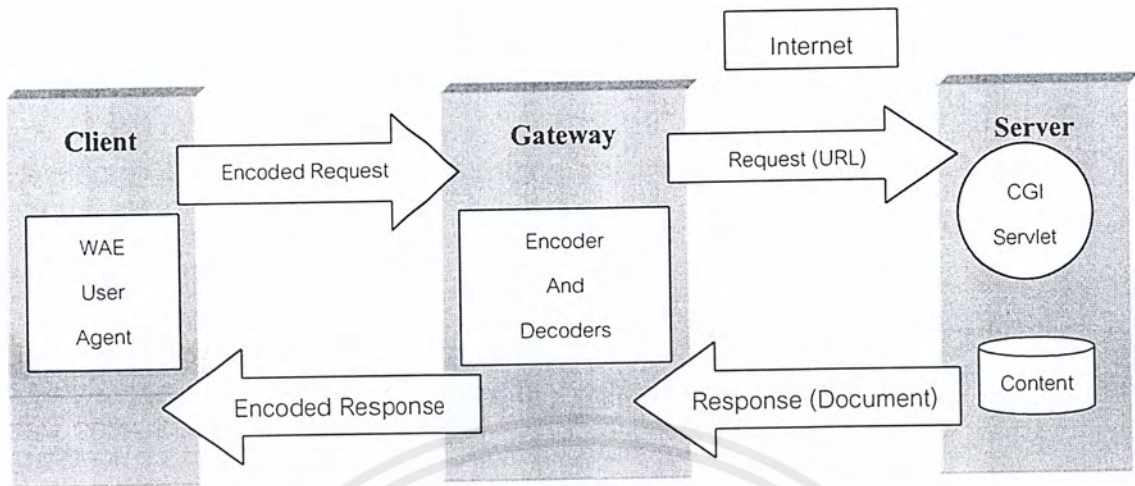
เว็บไซต์ประโยชน์ของรูปแบบอินเทอร์เน็ตเพื่อเตรียมพื้นฐานบริการที่ยืดหยุ่นได้ เพื่อให้เหมาะแก่อุปกรณ์ไร้สายในการเข้าถึงข้อมูลที่เสนอโดยเว็บ เว็บไซต์อยู่บนพื้นฐานเทคโนโลยีอินเทอร์เน็ตที่รู้จักกันดีที่ถูกลดขนาดลงมาให้เข้ากับข้อบังคับของสิ่งแวดล้อมไร้สาย

การสร้างบริการที่ใช้ HTML ไม่เหมาะสำหรับอุปกรณ์มือถือขนาดเล็ก เพราะ HTML ตั้งใจใช้บนคอมพิวเตอร์ที่มีหน้าจอขนาดใหญ่ อุปกรณ์ไร้สายที่มีความกว้างแถบความถี่ต่ำจึงไม่เหมาะสำหรับส่งข้อมูลที่มีขนาดใหญ่อย่าง HTML

WML เสนอรูปแบบเนวิเกชัน ( Navigation ) ออกแบบสำหรับอุปกรณ์ที่มีจอแสดงผลขนาดเล็ก และการรับอินพุตถูกจำกัด ( ไม่มีเมาส์ และคีย์บอร์ดมีจำกัด ) เพื่อให้เหมาะกับแถบความถี่ในเครือข่ายไร้สาย WML สามารถเข้ารหัสให้เป็นแบบไบนารีได้โดย WAP Gateway/Proxy เอนติตี ( entity ) ติดต่อกับโดเมนไร้สาย ( wireless domain ) ด้วยอินเทอร์เน็ต

เว็บเพียงเตรียมทางสำหรับสนับสนุนงานที่พัฒนามากขึ้น โดยใช้ WMLScript

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 รูปแบบเว็บ

ในรูปที่ 2-2 แสดงถึงรูปแบบของเว็บโปรแกรมมิ่ง ( WAP programming ) มีความคล้ายคลึงกันกับรูปแบบอินเทอร์เน็ตมาก มีข้อแตกต่างตรงที่มี WAP Gateway/Proxy

ถ้าสังเกตให้ดีจะรู้ว่าการร้องขอ ( request ) ที่ส่งจากไคลเอนต์ไป WAP Gate/Proxy ใช้โพรโตคอล WSP ( Wireless Session Protocol ) ที่สำคัญคือ WSP เป็นเวอร์ชันไปนารีของ HTTP

เว็บถูกออกแบบให้สามารถใช้งานเครือข่ายไร้สายได้ เช่น GSM, IS-95, IS-136 และ PDC ในท้ายที่สุด WAP protocol stack ถูกออกแบบในวิธีการแบ่งเป็นชั้นๆ ( layer ) หมายความว่านั้นสามารถขยายได้และพิสูจน์ได้ในภายหลัง

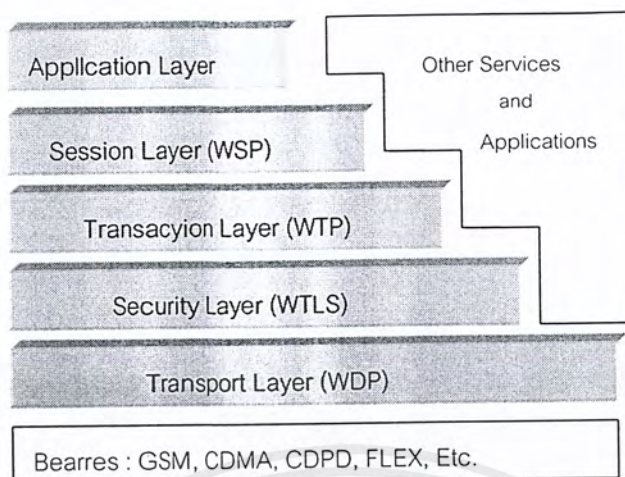
## 2.6 สถาปัตยกรรมของเว็บ ( WAP Architecture )

เว็บถูกออกแบบในวิธีการแบ่งเป็นชั้นๆ เพื่อความสามารถในการขยาย, ยืดหยุ่น, และลำดับชั้นได้ ด้วยรูปแบบ OSI model ( Open System Interconnection model ) สแต็กของเว็บ ( WAP-stack ) โดยพื้นฐานแล้ว แบ่งได้เป็น 5 ชั้น คือ

- ลำดับชั้นการประยุกต์ใช้ ( Application layer )
  - WAE ( Wireless Application Environment )
- ลำดับชั้นการแบ่งเป็นช่วง ( Session layer )
  - WSP ( Wireless Session Protocol )
- ลำดับชั้นการติดต่อ ( Transaction layer )
  - WTP ( Wireless Transaction Protocol )
- ลำดับชั้นการรักษาความปลอดภัย ( Security layer )
  - WTLS ( Wireless Transport Layer Security )
- ลำดับชั้นการส่ง ( Transport layer )

### WDP ( Wireless Datagram Protocol )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 ลำดับชั้นของเว็บ

แต่ละชั้นของสแต็กของเว็บจะไปถึงอินเทอร์เน็ตเฟสไปยังชั้นที่เหนือกว่า หมายความว่าชั้นที่ต่ำกว่าไม่สามารถมองไปยังชั้นที่เหนือกว่าได้ สถาปัตยกรรมแบบลำดับชั้นยอมให้ออปพลิเคชันและบริการอื่นๆ ใช้ลักษณะที่จัดเตรียมโดยสแต็กของเว็บได้ดี ทำให้เป็นไปได้ที่จะใช้สแต็กของเว็บสำหรับบริการและออปพลิเคชันที่ปัจจุบันนี้ไม่ได้ระบุโดยเว็บ

### 2.6.1 สิ่งแวดล้อมในการใช้งานเว็บ ( WAP Application Environment )

ชั้นบนสุดในสแต็กของเว็บ คือสิ่งแวดล้อมการประยุกต์ใช้แบบไร้สาย ( WAE ) ซึ่งจัดเตรียมสิ่งแวดล้อมที่จะกำหนดขอบเขตของออปพลิเคชันให้กว้างพอเหมาะสำหรับอุปกรณ์ไร้สาย ส่วนประกอบของ WAE คือ

- รูปแบบการแอดเดรส ( Addressing model )  
การสร้างข้อความ ( syntax ) เหมาะสำหรับการตั้งชื่อทรัพยากรที่เก็บบนเซิร์ฟเวอร์
- WML  
เป็นภาษาที่มีน้ำหนักเบา ออกแบบมาเพื่อเข้ากับข้อบังคับของสิ่งแวดล้อมไร้สายด้วยความกว้างแถบความถี่ต่ำ และเข้ากับอุปกรณ์มือถือขนาดเล็ก
- WMLScript  
ภาษาสคริปต์ที่มีน้ำหนักเบา
- WTA  
เฟรมเวิร์คและโปรแกรมมิ่งอินเทอร์เน็ตเฟสสำหรับบริการโทรศัพท์

User-agent เป็นตัวแปลเนื้อเรื่อง และปฏิกริยาของผู้ใช้ ( user-interaction ) โดยทั่วไปในการติดต่อใช้สายจะมี user-agent ตามปกติคือเบราว์เซอร์ แต่เพื่อจะใช้ลักษณะในส่วนข้างบน เว็บจะมี user-agent 2 แบบ คือ WML user-agent กับ WTA user-agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WML user-agent จะแปลเอกสาร WML รวมถึง WMLScript ด้วย ส่วน WTA user-agent จะให้บริการทางด้านโทรศัพท์ ( handling telephony service )

- **รูปแบบการแอดเดรส ( Addressing Model )**

เว็บไซต์ใช้รูปแบบการแอดเดรสเหมือนกับที่ใช้บนอินเทอร์เน็ต นั่นคือการใช้ URL นั่นเอง URL ใช้ระบุถึงทรัพยากรแบบไม่ซ้ำกัน โดยใช้โปรโตคอลที่เป็นที่รู้จักกันดี นอกจากนี้เว็บไซต์ยังใช้ URI ( Uniform Resource Identifiers ) อีกด้วย URI ใช้กำหนดที่อยู่ทรัพยากร โดยไม่จำเป็นต้องใช้โปรโตคอลที่เป็นที่รู้จักกันดี ตัวอย่างเช่น การเข้าถึงฟังก์ชันโทรศัพท์ในอุปกรณ์ไร้สาย

- **WML ( Wireless Markup Language )**

WML ของเว็บไซต์เป็นภาษาที่คล้ายคลึงกับ HTML ที่ใช้บนเว็บ WML อยู่บนพื้นฐานของ XML ( Extensible Markup Language ) WML สามารถเข้ารหัสให้เป็นแบบไบนารี โดย WAP Gateway/Proxy ทำให้เหมาะแก่การใช้กับอุปกรณ์มือถือ

- **WMLScript**

WMLScript อยู่บนพื้นฐาน ECMAScript ซึ่งเหมือนกับจาวาสคริปต์ WMLScript เพิ่มความฉลาดให้แก่บริการใน WML เช่น การทำงานลอจิก, การวนลูป, การใช้ฟังก์ชันเงื่อนไข และการทำงานการคำนวณ เป็นต้น WMLScript สามารถเข้ารหัสไบนารีโดย WAP Gateway/Proxy เพื่อลดจำนวนข้อมูลที่จะส่งผ่านในอากาศ

- **การใช้งานประยุกต์โทรศัพท์ไร้สาย ( WTA : Wireless Telephony Application )**

สิ่งแวดล้อม WTA ( WTA environment ) เตรียมเส้นทางสำหรับการสร้างบริการโทรศัพท์ในโปรโตคอลเว็บไซต์

WTA user-agent อยู่บนพื้นฐาน WML user-agent แต่ถูกขยายด้วยฟังก์ชันที่ต้องการเป็นพิเศษสำหรับบริการโทรศัพท์ ฟังก์ชันนี้ประกอบด้วย

- **WTAI ( Wireless Telephony Application Interface )**

อินเตอร์เฟซไปทางชุดของฟังก์ชันที่มีความสัมพันธ์ไปทางโทรศัพท์ในโทรศัพท์เคลื่อนที่ ที่สามารถถูกอ้างจาก WML และ/หรือ WMLScript ฟังก์ชันเหล่านี้เช่น การจัดการการเรียกสาย, การจัดการข้อความ และการควบคุมสมุดจดโทรศัพท์

WTAI ถูกแบ่งออกเป็น 3 อย่าง คือ network common function, network specific function และ public function โดย Common function จะมีอยู่ในทุกเครือข่าย ขณะที่ specific function จะระบุฟังก์ชันที่ไม่ซ้ำกันไปยังเครือข่ายที่แน่นอน ส่วน public function จะสามารถอ้างจาก WML User-Agent ได้เลย

ในปัจจุบัน library ของ public function จะมีเพียงฟังก์ชันสำหรับการตั้งการเรียกสายเท่านั้น

- **ที่เก็บรวม ( Repository )**

หลายๆบริการ WTA ต้องการการทำงานแบบ real-time แสดงว่ามันเป็นไปได้ถ้าการอ้างถึงมีการล่าช้าแล้วจะกู้เนื้อหาจาก server ดังนั้นจึงเกิดที่เก็บรวมเพื่อทำให้มันเป็นไปได้ที่จะเก็บบริการ WTA ฝังแน่นไว้ในอุปกรณ์ เพื่อสามารถเข้าถึงมันได้โดยไม่ต้องเข้าเครือข่าย

- **การถือเหตุการณ์ ( Event handling )**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของเหตุการณ์ในเครือข่ายมือถือคือ การเรียกสาย, การตัดสาย และการตอบรับสาย เพื่อที่จะสร้างบริการ โทรศัพท์มันต้องเป็นไปได้อันที่ทำงานในเหตุการณ์เหล่านี้ Event handling ใน WTA ทำให้บริการ WTA ถูกเก็บไว้ในที่เก็บรวม เพื่อเริ่มต้นในการตอบสนองต่อสักเหตุการณ์หนึ่ง เหตุการณ์สามารถกำหนดขอบเขตการกระทำที่แน่นอนใน WML ได้ เพื่อทำให้มันเป็นไปได้ที่จะถือเหตุการณ์ภายในบริการ

- ตัวชี้บริการ WTA ( WTA Service indication )

ชนิดของเนื้อหาที่ยอมให้ผู้ใช้แจ้งเกี่ยวกับเหตุการณ์ต่างๆ เช่น new voice mail และให้ความเป็นไปได้ที่จะเริ่มต้นบริการที่สมควรเพื่อจะถือเหตุการณ์ โดยพื้นฐานแล้วใช้ตัวชี้บริการ WTA ทำให้มันเป็นไปได้ที่จะส่ง URL และข้อความไปยังอุปกรณ์ไร้สายได้ ข้อความจะถูกแสดงต่อผู้ใช้ ส่วน URL จะชี้จุดเริ่มต้นของบริการ ถ้าต้องการเลื่อนตัวชี้บริการไปสำหรับใช้งานที่หลัง ตัวชี้บริการควรถูกส่งไปยังอุปกรณ์โดยการ push

WTA framework อาศัยอยู่บน WTA User-Agent ที่สามารถทำตามฟังก์ชันที่ไม่ได้เตรียมโดย WML User-Agent ทั่วไป มีเพียงผู้จัดเตรียมเนื้อหาที่เชื่อถือได้เท่านั้นที่สามารถสร้างเนื้อหาได้ สำหรับ WTA User-Agent ผู้จัดเตรียมเนื้อหาที่เชื่อถือได้คือ Operator ดังนั้นต้องเป็นไปได้อันที่จะแบ่งแยกความแตกต่างระหว่าง server ที่ยอมให้บริการที่มีฟังก์ชันเหล่านี้ไปยัง User-Agent กับ server ที่ไม่ยอมให้จ่ายบริการเหล่านั้น

WTA server อาจเป็นเว็บเซิร์ฟเวอร์ธรรมดาใช้สำหรับใส่เนื้อหา มันอาจสามารถติดต่อสื่อสารกับเอนิตี้อื่นๆได้ เช่น IN-node หรือระบบ voice mail เพื่อเตรียมการขยายฟังก์ชันที่สัมพันธ์กับการโทรศัพท์ การติดต่อสื่อสารกับแต่ละเอนิตีสามารถถูกควบคุมโดยแอปพลิเคชันบน WTA server ที่สามารถค้นหาได้ภายในบริการ WTA โดยใช้ URL

## 2.6.2 แวร์โพรโตคอล ( WAP Protocol )

ชุดแวร์โพรโตคอลมี 4 โพรโตคอล สำหรับการติดต่อสื่อสารระหว่างไคลเอนต์กับ WAP Gateway/Proxy โพรโตคอลเหล่านี้ถูกออกแบบมาให้ใช้บนอินเทอร์เน็ต และสามารถใช้งานได้แตกต่างกัน 4 แบบ คือ

- Connectionless Mode

ลักษณะนี้ใช้เพียง WSP บนข้างบนสุดของ WDP มันเสนอบริการเดต้าแกรม ( Datagram ) หมายความว่า ไม่ acknowledge ข้อความที่ส่งมา และไม่รับประกันการส่งที่ถูกเสนอมา ฟังก์ชันสามารถมองเป็นรูปแบบส่งแล้วลืมแบบง่าย ๆ

- Connectionless Mode with Security

เป็นสิ่งที่เพิ่มเติมมาจาก Connectionless Mode โดยเพิ่ม WTLS เพื่อเตรียมการตรวจสอบ และการเข้ารหัส

- Connection Mode

ใช้ WTP เป็นส่วนเพิ่มเติมจาก WSP กับ WDP โดย WTP เสนอการส่งผ่านที่เชื่อถือได้ หมายความว่าส่งข้อความแล้วต้องถูก acknowledge และอาจจะส่งอีกครั้งถ้าเกิดการสูญหาย มันใช้ mode ของ WSP ที่มี long lived session ด้วย

- Connection Mode with Security

เป็นสิ่งที่เพิ่มเติมมาจาก Connection Mode โดยเพิ่ม WTLS เพื่อเตรียมการตรวจสอบ และการเข้ารหัส

● WSP ( Wireless Session Protocol )

WSP เป็นอินเตอร์เฟซระหว่าง WAE กับส่วนที่เหลือของสแต็กของโพรโตคอล โดย WSP เป็นเวอร์ชันไบนารีของ http1.1 พร้อมด้วยสิ่งเพิ่มเติมเช่น

- ความสามารถในการตกลงกัน ( Capability Negotiation )
- การเก็บช้อนเฮดเดอร์ ( Header Catching )
- session ที่อยู่ได้นาน ( Long-lived session )
- การpush ( Push )

ฟังก์ชันหลักของ Connection Mode ของ WSP ก็คือการตั้ง session ระหว่างไคลเอนต์กับ WAP Gateway/Proxy โดย session นี้มีความสามารถในการตกลงกันที่การตั้ง session และการ interrupt การติดต่อสื่อสารด้วย เช่น การเปลี่ยนตัวกลางการสื่อสาร มันสันนิษฐานว่ามีอายุยืนยาว และสามารถถูกพักชั่วคราว และสามารถกระทำต่อไปได้ แทนที่จะเลิกการติดต่อ ถ้าในบางเวลาไม่ต้องการการติดต่อสื่อสาร หมายความว่า ไม่จำเป็นต้องมีการตกลงกันใหม่เมื่อ session ถูกกระทำต่อไป ทำให้ traffic load น้อยลง สนับสนุน asynchronous handling ของการร้องขอ หมายความว่าถ้าการร้องขอข้อมูลต่างๆถูกส่งไป คำตอบอาจจะถูกส่งมาในลำดับที่แตกต่างกัน

WSP ยังคงสนับสนุนการเก็บช้อนเฮดเดอร์เพื่อลดขนาดการใช้ตัวกลางสื่อสาร ใน http ที่ไม่สนับสนุนการเก็บช้อนเฮดเดอร์ประมาณ 90% ของการร้องขอบรรจุเฮดเดอร์ครั้งที่จำเป็นจะต้องถูกส่งหลายครั้ง

● WTP ( Wireless Transaction Protocol )

WTP ถูกตอบสนองได้สำหรับควบคุมการส่งและรับข้อความ มันเตรียมทางการติดต่อสื่อสารที่เชื่อถือได้ที่ข้อความถูกระบุแบบไม่ซ้ำ ดังนั้นจะไม่มีกรยอมรับซ้ำครั้งที่สอง และอาจจะส่งอีกครั้งไปยังตัวรับถ้ามีการสูญหายในการส่ง ไม่มี coupling ระหว่างการติดต่อสื่อสารเหมือนกับว่า แต่ละลำดับการติดต่อสื่อสารจะคงอยู่ระหว่างการแลกเปลี่ยน set ข้อความของแต่ละลำดับนั้นเท่านั้น WTP ทำงานกับคลาส ( class ) ข้อความแตกต่างกัน 3 คลาส

- การส่งที่เชื่อถือไม่ได้กับข้อความที่ไม่มีผล  
ไม่มีการส่งอีกครั้งถ้าข้อความที่ส่งสูญหายไป
- การส่งที่เชื่อถือได้กับข้อความที่ไม่มีผล  
ฝั่งรับจะ acknowledge ข้อความที่ถูกส่งมา หรือไม่ข้อความจะถูกส่งเข้ามาใหม่
- การส่งที่เชื่อถือได้กับข้อความที่มีผลที่เชื่อถือได้ ( การติดต่อสื่อสาร 3 ทาง )

การร้องขอข้อมูลถูกส่ง และผลถูกรับมาที่ในที่สุดถูก acknowledge โดยส่วนเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WTP ยังถูกปรับปรุงเพื่อบังคับตัวกลางสื่อสารไร้สายที่ลดขนาด protocol overhead โดยการแนะนำฟังก์ชันที่ใช้ลดจำนวนของการส่ง ( การส่งอีกครั้ง ) เช่น การต่อข้อความ และการ acknowledge ของการร้องขอข้อมูลที่ได้รับ WTP สามารถถูกขยายด้วยฟังก์ชันสำหรับแบ่งส่วนข้อความ และนำข้อความกลับมารวมใหม่ รวมถึงการเลือกการส่งใหม่ของส่วนที่สูญหาย

- **WTLS ( Wireless Transport Layer Security )**

จุดประสงค์ของ WTLS คือการเตรียมความปลอดภัยในลำดับชั้นการส่งระหว่าง WAP client กับ WAP Gateway/Proxy ซึ่ง WTLS อยู่บนพื้นฐานความปลอดภัยในชั้นการขนส่ง ( TLS : Transport Layer Security ) เวอร์ชัน 1.0 แต่ถูกทำให้กระชับขึ้นสำหรับช่องการติดต่อสื่อสารที่แคบ ลักษณะเด่นมีดังนี้

- ความครบถ้วนในการใช้ MAC ( Message Authentication Codes )
- การเข้ารหัสเป็นความลับ
- การตรวจสอบ และการไม่ละทิ้งของเซิร์ฟเวอร์และไคลเอนต์ โดยใช้ digital certificate

ลักษณะเหล่านี้ทำให้มันเป็นไปได้ที่จะรับประกันว่าการส่งข้อมูลจะไม่ถูกจัดทำโดยบุคคลที่ 3 รับรองความเป็นส่วนบุคคลที่ผู้แต่งข้อความสามารถระบุชี้ชัด และทั้งสองฝ่ายไม่สามารถละทิ้งการส่งข้อความได้ ความปลอดภัยในการติดต่อสื่อสารถูกตั้งในช่วงการติดตั้งที่มีการตกลงกัน เช่น การตั้งพารามิเตอร์ ( parameter ), การแลกเปลี่ยนกุญแจ และการตรวจสอบ ทั้งสองฝ่ายสามารถทำลายความปลอดภัยในการติดต่อสื่อสารระหว่างการติดตั้ง หรือเวลาใดก็ได้ในภายหลัง

WTLS เป็น option ที่สามารถถูกใช้กับทั้ง Connectionless Mode และ Connection Mode ของโครงสร้าง WAP stack ถ้าใช้ WTLS มันมักอยู่บนส่วนยอดของ WDP

- **WDP ( Wireless Datagram Protocol )**

พื้นฐานของ WAP Protocol stack คือชั้นเคต้าแกรมซึ่ง WDP นี้นำเสนออินเตอร์เฟซถาวรให้กับชั้นสูงขึ้นไปของสแต็ก

ถ้าเราใช้บนตัวกลางที่สนับสนุน UDP ( User Datagram Protocol ) แล้ว WDP ก็ไม่เป็นที่ต้องการ แต่บนตัวกลางอื่นๆ เช่น GSM, SMS นั้นฟังก์ชันเคต้าแกรมถูกจัดเตรียมโดย WDP ดังนั้นเราจึงใช้ UDP หรือไม่ก็ WDP ให้บริการเคต้าแกรมที่ซ่อนลักษณะของตัวกลางที่แตกต่างกัน และเตรียมฟังก์ชัน port number และถ้าหากจำเป็น WDP ก็สามารถขยายฟังก์ชันสำหรับการแยกส่วน และการรวมตัวกันใหม่ของเคต้าแกรมที่ใหญ่เกินสำหรับการอยู่ภายใต้ตัวกลาง

มีความเป็นไปได้ที่จะขยาย WDP ด้วย optional protocol สำหรับการรายงานความผิดพลาดที่เรียกว่า WCMP ( Wireless Control Message Protocol ) โพรโตคอลนี้สามารถถูกใช้เมื่อเราไม่ได้ใช้บนตัวกลาง IP ( IP มี control message protocol ของตัวเอง ) WCMP สามารถถูกใช้สำหรับจุดประสงค์ด้านข้อมูลและการวิจัยอีกด้วย

## บทที่ 3

### Wireless Markup Language

#### 3.1 บทนำ

WML ของเว็บเป็นภาษาที่คล้ายคลึงกับ HTML ที่ใช้บนเว็บ WML อยู่บนพื้นฐานของ XML ( Extensible Markup Language )

WML ใช้รูปแบบการเปรียบเทียบกับเต็ค/การ์ด ( deck/card metaphor ) เพื่อระบุถึงบริการ การ์ดเป็นหน่วยของการทำงานโต้ตอบกับผู้ใช้ นั่นคือทั้งการนำเสนอข้อมูล และการร้องขอข้อมูลจากผู้ใช้ ชุดรวมของการ์ดเรียกว่าเต็ค ซึ่งมักจะประกอบกันขึ้นเป็นบริการ

ลักษณะสำคัญของ WML คือ

- ตัวแปร ( Variables )
- ลักษณะรูปแบบข้อความ ( Text formatting features )
- สนับสนุนรูปภาพ ( Support for images )
- สนับสนุนการใช้ปุ่ม ( Support for soft-buttons )
- การควบคุมการค้นหา ( Navigation control )
- การควบคุมประวัติเบราว์เซอร์ ( Control of browser history )
- สนับสนุนการถือเหตุการณ์ ( Support for event handling ) เช่น บริการ โทรศัพท์
- ชนิดที่แตกต่างกันของปฏิริยาผู้ใช้ ( Different types of user interaction ) เช่น การเลือกรายการ การใส่  
อินพุต

WML สามารถเข้ารหัสเป็นรหัสไบนารีได้ โดย WAP Gate/Proxy เพื่อเหมาะสมแก่การใช้ในแถบความถี่ที่จำกัดของโคเมนไรราย

#### 3.2 พื้นฐาน WML

- อิลิเมนต์ ( Elements )

WML นิยามอิลิเมนต์โดยแต่ละอิลิเมนต์มีแท็ก ( tag ) ที่ชื่อแท็กเป็นตัวอักษรตัวเล็ก มี 2 รูปแบบ คือ รูปแบบที่ 1 แท็กที่มีสิ่งที่บรรจุไว้ข้างใน

```
<tag> tag contents </tag>
```

รูปแบบที่ 2 แท็กที่ไม่มีสิ่งที่บรรจุไว้ข้างใน

```
<tag/>
```

- แอตทริบิวต์ ( Attributes )

แอตทริบิวต์ใช้อธิบายหน้าตาของอิลิเมนต์ ประกาศโดยใช้ตัวอักษรตัวเล็ก และค่า ( values ) ของแอตทริบิวต์จะประกาศอยู่ในเครื่องหมายอัญประกาศ ( ‘ หรือ “ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอตทริบิวต์หนึ่งที่ใช้กับทุกอิลิเมนต์ที่สามารถแสดงข้อความได้ คือ xml:lang ที่ใช้นิยามว่ามีการใช้ภาษาธรรมชาติ ซึ่งคล้ายกับภาษาพูด หรือภาษาเขียนท้องถิ่น และยังมีอีก 2 แอตทริบิวต์ที่ใช้กับทุกอิลิเมนต์ได้ คือ id ที่ใช้ระบุถึงแต่ละอิลิเมนต์ และ class ที่แสดงถึงความผูกพันของอิลิเมนต์

- การนำเสนอการสร้างประโยค ( Syntax Presentation )

ขึ้นต้นด้วยแท็กเปิดของอิลิเมนต์ แล้วใส่แอตทริบิวต์ของอิลิเมนต์นั้นไว้ในแท็ก ตามด้วยสิ่งที่จะบรรจุอยู่ในนั้น แล้วปิดด้วยแท็กปิด อิลิเมนต์สามารถมีอิลิเมนต์ไว้ภายในได้

ค่าของแอตทริบิวต์สามารถมีได้ 1 ค่า แต่ตัวเลือกของค่าของแอตทริบิวต์สามารถมีได้หลายค่า ซึ่งแบ่งโดยเส้นตามแนวตั้ง ( vertical bar ) ค่า default จะแสดงเป็นตัวหนา

- URLs ( Uniform Resource Locators )

WML เป็นภาษาที่ใช้สร้างแอปพลิเคชันที่อยู่บนพื้นฐานเว็บ ดังนั้นจึงต้องการสิ่งที่จะอ้างอิงถึงเนื้อหาและโปรแกรมบนเว็บไซต์ ดังนั้น WML จึงใช้ URL ในการอ้างอิงถึงเนื้อหาที่จำเพาะเว็บไซต์หรือ IP address แต่ถ้าต้องการชี้ไปภายในเอกสารเดียวกัน ส่วนการติดตั้งเริ่มต้นด้วย URL ตามด้วยเครื่องหมาย # แล้วตามด้วยชื่อของส่วนนั้น สำหรับเอกสาร WML ชื่อส่วนจะระบุถึงการ์ดในเอกสารนั้น

- บทความ ( Context )

บทความจะรวมถึงการนิยามตัวแปรและค่าของมัน และสแต็กประวัติของ URL ที่ user agent ได้เข้าไป บทความ user agent ( user agent context ) จะฝังแน่นอยู่ในอุปกรณ์ ระหว่างการแปลเอกสาร, การคิดต่อเครือข่าย และเหตุการณ์อื่นๆ user agent ที่นี้อาจจะต้องรักษายบทความระหว่างปิดเครื่องเพื่อประหยัดค่าเครือข่าย

- ชุดตัวอักษร ( Character Set )

WML ใช้ชุดอักขระของ XML คือ ISO/IEC-10646 [ISO10646] และเปลี่ยนมาใช้ Unicode 2.0 [UNICODE] WML สนับสนุนการใช้อักขระตัวเลขแทนอักขระพิเศษบางตัว ใช้ทั้งเลขฐานสิบ ( ขึ้นต้นด้วย "&#" ) และเลขฐานสิบหก ( ขึ้นต้นด้วย "&#x" )

- อักขระพิเศษ ( Special Characters )

อักขระเครื่องหมายบางตัวที่ต้องการแสดงออกไปจะใช้อิลิเมนต์อักขระ ( character element ) แล้วตามด้วยเครื่องหมายอัฒภาค ( ; ) เช่น เครื่องหมาย & จะเขียนด้วย &amp;

- ข้อความ ( Text )

WML จะมี อิลิเมนต์, แอตทริบิวต์ และค่าของแอตทริบิวต์ที่ระบุไว้แล้ว จะต้องเขียนด้วยตัวอักษรตัวเล็ก

- คำอธิบาย ( Comments )

WML ใช้รูปแบบเดียวกันกับของ HTML คือ <!-- comment -->

- เครื่องหมายอัญประกาศ ( Quotation Marks )

WML ใช้เครื่องหมายอัญประกาศ ( ' หรือ " ) เมื่อใช้เป็นขอบเขตของค่าของแอตทริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 เด็ค ( Deck )

เด็คเป็นหน่วยที่เล็กที่สุดของ WML ที่ถูกส่งไปยังอุปกรณ์ ซึ่งเว็บถูกออกแบบมาสำหรับอุปกรณ์เคลื่อนที่ ดังนั้นเด็คควรจะมีความเล็กเท่าที่จะเป็นไปได้ ซึ่งน่าจะต่ำกว่า 1 กิโลไบต์ ถ้าแอปพลิเคชันมีความซับซ้อนมากก็ควรจะแบ่งออกเป็นเด็คย่อยๆ

รูปแบบโดยถูกต้องของเด็คของ WML ควรเริ่มต้นด้วยบทนำ ตามด้วยข้อมูลในส่วนต้นบางอย่าง แล้วจึงตามด้วยลำดับของการ์ด การ์ดเป็นหน่วยเดียวของการทำงานที่มีผลกระทบต่อผู้ใช้ เนื้อหาที่จะแสดงออกให้เห็นของการ์ด จะมีการแสดงออกมาพอดีกับหน้าจอเล็กๆของอุปกรณ์เคลื่อนที่ แต่ถ้าเนื้อหาที่มีขนาดใหญ่กว่าหน้าจอ จะต้องมีการ scrolling มีทั้งแนวตั้งและแนวขวาง ถ้ามีการรับข้อมูลที่ขนาดไม่พอเหมาะกับหน้าจอจะต้องแบ่งการรับข้อมูลออกเป็นแต่ละหน้าจอ

แอปพลิเคชันบนเว็บ เป็น โปรแกรม XML จะใช้บทนำดังนี้

```
<?xml version="1.0"?>
```

```
<!DOCTYPE <WML> PUBLIC "-//WAPFORUM//DTD <WML> 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

บทนำของเด็คตามด้วยเด็คที่แท้จริง กันเขตโดยแท็ก <wml> เด็คนั้นไม่มีแอตทริบิวต์

```
<wml>
```

```
<head>
```

```
<template>
```

```
<card>
```

```
</wml>
```

<head> บรรจุข้อมูลเกี่ยวกับเด็คทั้งหมดรวมทั้งการควบคุมการเข้าและข้อมูล meta อิลิเมนต์นี้ไม่มีแอตทริบิวต์

<template> บรรจุข้อมูลเกี่ยวกับเหตุการณ์ในระดับเด็ค

<card> นิยามอินเตอร์เฟซผู้ใช้ ( user interface ) กับการทำงานลอจิกของเด็ค เด็คต้องมีอย่างน้อย 1 การ์ดเพื่อให้โปรแกรมเว็บ ถูกต้อง

### 3.4 การ์ด ( Cards )

การ์ดบรรจุข้อมูลที่มีรูปแบบ ( formatting information ), เนื้อหาที่ใช้แสดงผล ( displayable content ) และคำสั่งที่ใช้ในการปฏิบัติการ ( processing instructions ) ข้อมูลในอิลิเมนต์นี้ถูกแบ่งเป็น 2 อย่าง คือ คำสั่ง ( instructions ) และเนื้อหา

การ์ดสามารถมีพารากราฟ ( อิลิเมนต์ <p> ) และพารากราฟสามารถบรรจุอิลิเมนต์ต่างๆไว้ภายใน เนื้อหาที่จะทำให้มองเห็นได้ในการ์ดจะต้องอยู่ภายในอิลิเมนต์พารากราฟ

การ์ดในเด็คต้องมีอิลิเมนต์ที่อย่างน้อยต้องมีลักษณะดังนี้ คือ เนื้อหา, คำสั่งถือเหตุการณ์ ( event handling instruction ) หรือการเข้าข้อมูล ( data entry )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 เนื้อหา ( Content )

เนื้อหา คือ ข้อความที่จะถูกแสดงบนหน้าจอของอุปกรณ์เว็บ โดยมี user agent เป็นตัวตัดสินใจในครั้งสุดท้ายว่าจะแสดงเนื้อหาออกมาอย่างไร

### 3.6 การลิงก์ ( Link )

การสร้างลิงก์ให้กระโดดไปทำงานยังการ์ดต่างๆ ภายในเด็ก จะใช้แท็ก <a> ตัวอย่างการเขียนโค้ด WML เพื่อลิงก์กันระหว่าง 2 การ์ด แสดงไว้ในสคริปต์ linkcard.wml

```
<wml>
  <card id="card1" title="Card no. 1">
    <p> This is card 1<br/> <br/>
    <a href="#card2">Go card 2</a></p>
  </card>
  <card id="card2" title="Card no. 2">
    <p> Now in card 2 <br/> <br/>
    <a href="#card1">Back to card 1</a></p>
  </card>
</wml>
```

สคริปต์ linkcard.wml

มีข้อควรทราบเกี่ยวกับเด็ก คือ หน่วยความจำสำหรับเก็บข้อมูลเด็กในอุปกรณ์เว็บจะมีขนาดเล็กมาก ดังค่าประมาณต่อไปนี้

เว็บเบราว์เซอร์	ขนาดของ เด็ก
Ericsson R320	3.0 KB
Ericsson R380	3.5 KB
Nokia 7110	1.5 KB
UP. Browser ( ในอุปกรณ์เว็บ )	1.4 KB

ตารางที่ 3-1 แสดงค่าประมาณของหน่วยความจำในอุปกรณ์เว็บ

ดังนั้นในการสร้างเว็บเพจหรือเอกสาร WML จึงต้องระวังขนาดของไฟล์ด้วย เพราะถ้าขนาดใหญ่เกินกว่าขนาดของเด็กที่เว็บเบราว์เซอร์จะรับได้ จะเกิดปัญหาในการแสดงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้การลิงก์ระหว่างการ์ดภายในเด็กเดียวกันด้วยแท็ก <a> แล้ว การลิงก์ระหว่างเอกสาร WML ก็สามารถใช้แท็ก <a> ได้

### 3.7 ตั้งเวลาของการ์ดด้วย ontimer

ในแต่ละการ์ดของ WML สามารถเขียนส่วนขยายเพื่อการตรวจสอบเหตุการณ์หรือตั้งค่าการทำงานบางอย่างได้ เช่น ส่วนขยาย ontimer และแท็ก <time> ที่ใช้ตั้งเวลาการทำงานของการ์ด โดยเมื่อถึงเวลาที่ตั้งไว้ก็จะไปทำงานที่การ์ดอื่นได้

ตัวอย่างการใช้งานได้แสดงไว้ในสคริปต์ textanimate.wml โดยจะมีการ์ดทั้งหมด 4 การ์ด แต่ละการ์ดจะแสดงข้อมูลออกมา 3 วินาที ( เวลาที่กำหนดไว้ในแท็ก <timer> มีหน่วยเป็นมิลลิวินาที ) หลังจากนั้นจะกระโดดไปทำงานยังการ์ดอื่นวนต่อกันไปเรื่อยๆ คือ card1 -> card2 -> card3 -> card4 ->card1...

```
<wml>
  <card id="card1"ontimer="#card2">
    <timer value="30"/>
    <p align="center"><br/>
    W A P ?</p>
  </card>

  <card id="card2"ontimer="#card3">
    <timer value="30"/>
    <p align="center"><br/>
    W = Wireless</p>
  </card>

  <card id="card3"ontimer="#card4">
    <timer value="30"/>
    <p align="center"><br/>
    A = Applicaton</p>
  </card>

  <card id="card4"ontimer="#card1">
    <timer value="30"/>
    <p align="center"><br/>
    P = Protocol</p>
  </card>
</wml>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สคริปต์ textanimate.wml

### 3.8 แสดงรูปภาพ <image>

การเรียกรูปภาพมาแสดงในเว็บเพจของ WML จะใช้แท็ก <image> เหมือนกับ HTML แต่มีข้อกำหนดว่า ต้องเขียนส่วนขยายของ alt ด้วย ( ถ้าเป็น HTML จะไม่บังคับให้เขียน ) รูปภาพมาตรฐานของเว็บ คือแบบ WBMP ( Wireless BitMap )

ตัวอย่างการเขียนแท็ก <image> เพื่อเรียกรูปภาพ cute.wbmp มาแสดงในเว็บเพจ แสดงไว้ในสคริปต์ image.wml

```
<wml>
  <card id="card1">
    <p></p>
  </card>
</wml>
สคริปต์ image.wml
```

### 3.9 รับข้อมูลด้วย <input>

การรับข้อมูลจากผู้ใช้ถือว่าเป็นเรื่องที่สำคัญอีกอย่างหนึ่ง ใน HTML นั้นจะมีฟอร์มและอินพุตต่างๆให้เลือกใช้ ขึ้นอยู่กับว่าเราต้องการให้ผู้ใช้ป้อนข้อมูลเข้ามาแบบไหน สำหรับ WML ก็มีแท็ก <input> เพื่อรับข้อมูลจากผู้ใช้เหมือนกัน ลักษณะการเขียนจะคล้ายกับ HTML แต่จะไม่มีแท็ก <form> คุ้มครองที่เป็นอินพุตแต่อย่างใด ในสคริปต์ basicinput.wml จะเป็นตัวอย่างแสดงให้เห็นถึงการใส่แท็ก <input> เพื่อรับข้อมูล User Name และรหัสผ่านจากผู้ใช้

```
<wml>
  <card title="Input Demo"><p>
    User : <input type="text" name="usr">
    Password :
    <input type="password"/></p>
  </card>
</wml>
สคริปต์ basicinput.wml
```

### 3.10 เลือกข้อมูลด้วย <select>

นอกจากแท็ก <input> ที่ใช้รับข้อมูลแล้ว ยังมีแท็กสำหรับสร้างตัวเลือกข้อมูลที่ต้องการได้อย่างง่าย ๆ ด้วย คือแท็ก <select> ซึ่งถ้าเป็น HTML แล้วแท็กนี้จะสร้างตัวเลือกที่เรียกว่า Drop Down List Box เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่สำหรับ WML แล้ว แท้ก็สร้างได้ทั้ง Radio และ Checkbox ขึ้นอยู่กับการระบุในส่วนขยายในแท็ก <select>

ถ้าไม่มีการกำหนดหรือระบุค่าใดๆ ก็ทำงานเป็น Radio แต่ถ้าระบุส่วนขยาย multiple="true" เอาไว้ ก็ทำงานเป็น Checkbox ตัวอย่างการใช้งานแท็ก <select> แสดงไว้ในสคริปต์ selectinput.wml

```
<wml>
  <card title="WAP Tour">
    <p>
      <b>Province :</b>
      <select name="province" title="&gt;&gt;">
        <option value="bk">Bangkok</option>
        <option value="cm">Chiangmai</option>
        <option value="cr">Chiangrai</option>
        <option value="pk">Phuket</option>
        <option value="sk">Songkhla</option>
      </select> <br/>
      <b>Information :</b>
      <select name="info" title="&gt;&gt;" multiple="true">
        <option value="ht">Hotel</option>
        <option value="rt">Restaurant</option>
        <option value="tv">Travel</option>
        <option valie ="wt">Weather</option>
      </select>
    </p>
  </card>
</wml>
สคริปต์ selectinput.wml
```

### 3.11 สรุปโดยรวมของ WML

ภาษา WML เป็นภาษาที่ได้รับการถ่ายทอดมาจากภาษา XML ซึ่งเป็นมาตรฐานในการทำ Internet markup Language

ตามข้อกำหนดของ WAP 1.1 ต้องเริ่มต้นด้วย

```
<?xml version="1.0"?>
```

```
<!DOCTYPE WML PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"http://www.wapforum.org/wml_1.1.xml>
```

ในส่วนประกอบพื้นฐานในภาษา WML นั้นเริ่มต้นด้วย <wml> จะมีรูปแบบดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<wml>
  <card>
    ...
  </card>
  <card>
    ...
  </card>

```

```

</wml>

```

ในแต่ละ <card> ... </card> จะเป็นหน้าจอหนึ่งหน้าจอเท่านั้น  
ตัวอย่างเช่น

```

<wml>
  <card>
    <p>
      Hello Word
    </p>
  </card>
</wml>

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### Servlet และ JavaServer Pages

#### 4.1 บทนำ

แม้ว่าโลกของอินเทอร์เน็ตจะเพิ่งเกิดขึ้นเพียงไม่กี่ปีก็ตาม เทคโนโลยีที่ใช้กับอินเทอร์เน็ตกลับมีการเปลี่ยนแปลงไปอย่างรวดเร็วมาก จำได้ว่าสมัยแรก ๆ เพจต่าง ๆ ที่อยู่ในเว็บจะเป็นลักษณะของ static page หรือเพจที่ไม่มีการเปลี่ยนแปลงเนื้อหาที่ไม่ว่าจะนานเท่าไรนอกเสียจากว่าผู้ดูแลเพจนั้นจะทำการอัปเดตเพจดังกล่าว เพจลักษณะนี้เป็นเพจที่นิยมใช้กันทั่วไปในอินเทอร์เน็ตสมัยแรก ต่อมาจากนั้นไม่นานทางผู้ผลิตเบราว์เซอร์ได้ทำการเพิ่มความสามารถให้กับเพจโดยอนุญาตให้เพจสามารถแทรกสคริปต์เล็ก ๆ ลงไปพร้อมกับส่วนที่เป็น HTML ได้ซึ่งจุดนี้ก็คือจุดเริ่มต้นของ JavaScript ทางฝั่งไคลเอนต์นั่นเอง หรือแม้แต่ว่า Java applet เองที่สามารถไปทำงานในฝั่งไคลเอนต์ได้ แม้ว่าเพจจะเริ่มมีความสามารถในการโต้ตอบกับผู้ใช้โดยอิงความสามารถจาก JavaScript แล้วก็ตาม ถ้าพูดถึงในแง่ของส่วนเนื้อหาของตัวเพจจริง ๆ แล้วตัวเพจเองก็ยังคงเป็น static page อยู่เช่นเดิม เมื่อก่อนผู้ใช้อินเทอร์เน็ตเริ่มมีมากขึ้น ความต้องการที่จะให้เพจสามารถทำการรับส่งข้อมูลรวมไปถึงเปลี่ยนแปลงเนื้อหาได้โดยอัตโนมัติก็มีมากขึ้น แต่ด้วยข้อจำกัดที่ว่าต้องเป็นเบราว์เซอร์รุ่นใหม่เท่านั้นที่สามารถรองรับเทคโนโลยีเหล่านี้ และยังมีอีกหลายส่วนที่ไม่สามารถใช้เบราว์เซอร์ที่มีความสามารถสูงได้ เช่น Browser WML ในโทรศัพท์มือถือ เทคโนโลยีที่เกิดขึ้นเพื่อรองรับความต้องการเหล่านี้ก็คือแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์นั่นเอง

แอปพลิเคชันทางฝั่งเซิร์ฟเวอร์ในระยะแรก ๆ มักถูกเขียนขึ้นด้วยแนวคิดของ CGI โดยหลักการทำงานง่าย ๆ ก็คือเว็บเบราว์เซอร์จะทำการส่งข้อมูลที่เกิดจากการกระทำ ( Action ) ของผู้ใช้ เช่น การคลิกลิงก์ หรือการกรอกแบบสอบถามไปยังเว็บเซิร์ฟเวอร์ โดยแทนที่เว็บเซิร์ฟเวอร์จะทำการส่งเพจที่เป็น static page กลับมา เว็บเซิร์ฟเวอร์จะทำการ forward ข้อมูลดังกล่าวไปยังโปรแกรมซึ่งถูกจัดไว้ โปรแกรมดังกล่าวจะทำการประมวลผลข้อมูลที่ได้แล้วจะส่งผลกลับไปยังเว็บเซิร์ฟเวอร์ ซึ่งทางเว็บเซิร์ฟเวอร์ก็จะส่งผลที่ได้นี้กลับไปยังเว็บเบราว์เซอร์อีกทีหนึ่ง

JSP ( JavaServer Pages ) เป็นภาษาที่ใช้ในการเขียน Servlet ซึ่ง JSP เป็น web-scripting เทคโนโลยีคล้ายกับ Netscape server-side JavaScript ( SSJS ) หรือ Microsoft Active Server Pages ( ASP ) แต่ผิคนตรงที่หัวใจของ JSP คือจาวา ( Java ) ซึ่งเป็นภาษาที่แนวคิดหลักอยู่ที่ออบเจกต์ ( object-oriented style ) ซึ่งช่วยทำให้ง่ายต่อการพัฒนาในโปรเจกต์ใหญ่ ๆ ตลอดจนสามารถนำส่วนประกอบต่าง ๆ กลับมาใช้ได้อีก ( software reusable ) จุดเด่นที่สำคัญของ JSP คือสามารถทำงานได้โดยไม่ขึ้นอยู่กับผู้ผลิตซอฟต์แวร์รายใดรายหนึ่งโดยเฉพาะ ซึ่งโดยทั่วไปเทคโนโลยีต่าง ๆ มักจะออกมาในลักษณะของผลิตภัณฑ์จากบริษัทผู้ผลิตแห่งใดแห่งหนึ่ง แต่ JSP ใช้ลักษณะของ specification ซึ่งกำหนดโดย Sun Microsystems ดังนั้นผู้ผลิตซอฟต์แวร์จึงสามารถอ้างอิง specification ที่กำหนดขึ้น ผลิต JSP Container ( ตัวที่ใช้ในการรัน JSP ) ขึ้นมาใช้กับแพลตฟอร์มใดก็ได้ JSP Container ที่เราจะใช้กันคือ Jasper ซึ่งเป็น JSP Container ที่อยู่ใน Tomcat Servlet Engine จากค่าย apache

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 อะไรคือ Servlets

Servlet ก็คือ applet ที่อยู่บนเซิร์ฟเวอร์ โดย Servlet จะถูกใช้งานโดยเว็บเซิร์ฟเวอร์ในขณะที่ applet จะถูกใช้งานที่เว็บเบราว์เซอร์

Servlet ถูกใช้งานที่ด้านเซิร์ฟเวอร์ ในด้านความปลอดภัยนั้นไม่จำเป็นต้องได้รับการดูแลเหมือน applet เพราะเว็บเบราว์เซอร์ไม่ได้ติดต่อกับ Servlet โดยตรง ซึ่ง Servlet จะถูกใช้งานที่ด้านเซิร์ฟเวอร์โดยเว็บเซิร์ฟเวอร์นั้น หมายถึงถ้าเว็บเซิร์ฟเวอร์อยู่หลัง firewall ซึ่งมีความปลอดภัยแล้ว Servlet ก็จะมีความปลอดภัยด้วย

## 4.3 ทำไมจึงต้องใช้ Servlet

Servlet สามารถแทนที่ CGI ที่โดยส่วนใหญ่เขียนด้วย Perl หรือ C ที่ผูกมัดในแต่ละเซิร์ฟเวอร์แพลตฟอร์ม ( platform ) แต่ Servlet ถูกเขียนด้วยจาวาซึ่งไม่ขึ้นกับแพลตฟอร์มและมีข้อดีที่เหนือกว่า CGI scripts ดังนี้

- Servlet อยู่คงทนถาวร คือถูกโหลด ( load ) เพียงครั้งเดียวโดยเว็บเซิร์ฟเวอร์และรองรับบริการได้ต่อเนื่อง ในขณะที่ CGI scripts จะอยู่ชั่วคราวแต่ละครั้งที่ให้บริการก็จะถูกโหลดและทำงานโดยเว็บเซิร์ฟเวอร์ เมื่อให้บริการเสร็จสิ้นก็ออกจากหน่วยความจำเป็นอย่างนี้ทุกครั้งที่ถูกใช้
- Servlet มีความเร็วสูงเพราะมีการโหลดเพียงครั้งเดียวซึ่งมีประสิทธิภาพมากกว่า CGI scripts
- Servlet ไม่ขึ้นกับแพลตฟอร์ม
- Servlet มีความสามารถในการขยาย เพราะถูกเขียนด้วยจาวาซึ่งจาวาเป็นภาษาแบบ object-oriented มีความง่ายในการขยายอยู่แล้ว
- Servlet มีความปลอดภัยสูงเพราะมีเพียงทางเดียวที่จะติดต่อกับ Servlet คือผ่านเว็บเซิร์ฟเวอร์โดยเฉพาะถ้าเว็บเซิร์ฟเวอร์อยู่หลัง firewall

## 4.4 ความเป็นมาของ JavaServer Pages

ก่อนที่จะใช้เว็บในการรับข้อมูลจากคอมพิวเตอร์เครื่องหนึ่งไปยังคอมพิวเตอร์อีกเครื่องหนึ่ง ในตอนแรกใช้วิธี UNIX-to-UNIX copy ( UUCP ) และวิธี File Transfer Protocol ( FTP ) แต่ทั้ง 2 วิธีนั้นยึดติดอยู่กับการรับส่งไฟล์มากทำให้ขาดการจัดการและการนำเสนอข้อมูล จึงเริ่มเข้าสู่การใช้ HTTP กับ HTML ในรูปแบบเว็บ ซึ่ง HTTP จะทำให้สามารถดึงเนื้อหาจากที่ต่างๆที่ระบุโดย URLs ได้ และ HTML จะระบุถึงการแสดงข้อมูลว่าจะออกมาในลักษณะใดรวมทั้งการแสดงรูปภาพ, ตาราง และเทคนิคอื่นๆ การใช้วิธีนี้มักจะเชื่อมเอกสารหนึ่งไปยังเอกสารอื่นๆ และจะเชื่อมต่อกันไปเรื่อยๆ ทำให้เว็บมีการขยายตัวอย่างรวดเร็ว และแม้ว่าจะสามารถร้องขอเอกสาร HTML ผ่าน FTP ได้ แต่ก็ยุ่งยากเกินไป จึงมีการสร้างโปรแกรมใหม่ขึ้นมา 2 โปรแกรม คือ เว็บเบราว์เซอร์ที่เป็นตัวร้องขอและอ่านเอกสาร HTML กับเว็บเซิร์ฟเวอร์ที่เป็นตัวเก็บข้อมูลและค้นหาไฟล์ให้เบราว์เซอร์ ซึ่งคุยกันผ่านทาง HTTP

เว็บเบราว์เซอร์นั้นอย่างที่เราคุ้นๆว่าเป็นโปรแกรมใหญ่และมีความซับซ้อนเพื่อการแสดงผลกับการเชื่อมต่อเอกสาร HTML ดังเช่นการแสดงรูปภาพหรือข้อความที่สามารถถูกคลิก ( click ) เพื่อไปยังเอกสาร เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อื่น และยังมีการแข่งขันกันพัฒนาเว็บเบราว์เซอร์จนทำให้ในปัจจุบันสามารถมีการประมวลผลที่ฝั่งเบราว์เซอร์ได้ เช่น สามารถควบคุมภาพเคลื่อนไหว, Java applets, JavaScript, การแสดงเสียงกับภาพเคลื่อนไหว เป็นต้น อย่างไรก็ตามก็ยังมีการพัฒนาในฝั่งเซิร์ฟเวอร์ด้วย ทำให้เว็บเซิร์ฟเวอร์สามารถประมวลผลและส่งผลกลับไปให้แก่ผู้ใช้ได้ การเปลี่ยนจากการส่งไฟล์มาเป็นการโปรแกรมทำให้เกิดความแตกต่างอย่างใหญ่หลวงในสิ่งที่เว็บสามารถทำได้ ก่อให้เกิดประโยชน์ เช่น

- เว็บไซต์ ( web site ) ที่ตีพิมพ์ข้อมูลแจ้งราคาหุ้น จะมีผู้ใช้จำนวนมาก และมีการปรับเปลี่ยนข้อมูลจำนวนมากอยู่บ่อยๆ การจะเก็บข้อมูลแบ่งเป็นไฟล์ต่างๆ นั้นจะต้องเก็บเป็นจำนวนมาก และการจะตามอัปเดต ( update ) ข้อมูลบ่อยๆ ก็เป็นไปได้ ดังนั้นจึงควรเปลี่ยนมาใช้โปรแกรมดีกว่าการเก็บไฟล์
- เว็บการค้าที่อนุญาตให้ผู้ใช้เลือกซื้อสินค้าได้ จำเป็นจะต้องมีโปรแกรมสำหรับการส่งสินค้า
- ปัจจุบันเว็บมีขนาดใหญ่มาก การค้นหาข้อมูลจำเป็นต้องใช้ search engine ที่ต้องการโปรแกรมเพื่อการสืบค้นหาไฟล์ที่มีค่าตรงกับที่ผู้ใช้ต้องการหาได้
- ผู้ใช้ส่วนมากชอบเว็บไซต์ที่สามารถเข้าไปลงคะแนน ( vote ) ได้ หรือเล่นเกมแข่งกันกับผู้ใช้คนอื่นได้

ทั้งหมดนี้เป็นตัวอย่างของเนื้อหาแบบไดนามิก ( dynamic content ) ที่มีผลทันทีตามความต้องการของผู้ใช้ ในระยะแรกการใช้โปรแกรมทางฝั่งเซิร์ฟเวอร์จะต้องใช้ผ่าน CGI โปรแกรมที่สร้างเนื้อหาแบบไดนามิกนั้นจะเรียกว่า CGIs ( CGI programs ) โดยโปรแกรมนี้ส่วนมากใช้ภาษา C หรือ Perl ในการเขียน แต่ก็สามารถใช้ภาษาอื่นได้ รูปแบบ CGI นี้มีปัญหาในเรื่องความเร็วเพราะต้องเสียเวลาในตอนเริ่มต้นโปรแกรมทุกครั้งที่มีการร้องขอ และปัญหาอีกประการหนึ่งที่สำคัญคือ การรวมกันของการนำเสนอกับบล็อก เป็นการยากที่จะหาผู้ที่เชี่ยวชาญการเขียน HTML กับการเขียนโปรแกรม และถ้าหากมีการออกแบบเพจใหม่ ก็ไม่สามารถเปลี่ยน HTML ได้โดยตรง จะต้องคำนึงถึงโปรแกรมด้วย

เพื่อการแก้ไขปัญหาของ CGI ทำให้เกิดการพัฒนารูปแบบอื่นๆ จากรูปแบบ CGI ไปเป็นรูปแบบ Fast CGI, Web Server Extensions, Application Server ซึ่งแต่ละรูปแบบก็ยังคงมีปัญหาที่แตกต่างกันไป จนในที่สุดสามารถแยก HTML ออกจาก CGI และนำไปวางในไฟล์ที่แยกออกต่างหาก ซึ่ง CGI จะเปิดอ่าน และแสดง HTML นั้นได้ ทำให้ผู้เขียน HTML สามารถแก้ไขไฟล์ได้โดยไม่ต้องไปแตะต้องส่วนของโปรแกรม ผู้เขียน HTML สามารถแสดงการใช้โปรแกรมได้โดยใช้แท็กพิเศษ ตัวอย่างเช่น แท็กพิเศษ <USER\_NAME\_HERE> และถ้า CGI เขียนในภาษา Perl แท็กนี้ก็จะสามารถเป็นตัวแปรภาษา Perl อย่าง \$USERNAME ได้ และในขณะที่เมื่อโปรแกรมอ่าน HTML ก่อนที่จะส่งผลไปให้ผู้ใช้ มันก็จะหาแท็กพิเศษนี้แล้วกระทำการใดๆ เพื่อให้ได้ USERNAME มา จากนั้นแทนที่แท็กพิเศษนั้นด้วยชื่อที่หามาได้ในทุกตำแหน่งที่แท็กปรากฏ แล้วค่อยส่งเพจไปให้แก่ผู้ใช้ วิธีการนี้ CGI ทำหน้าที่แปลงแท็กให้ไปเป็นการกระทำ เช่นระบบที่เรียกว่า templating แต่รูปแบบนี้ก็ไม่มีความสามารถมากพอที่จะทำทุกอย่างที่เพจแบบไดนามิกต้องการ หรือถ้ามีความสามารถมากพอ แท็กก็จะมีจำนวนมาก ด้วยเหตุผลนี้ ระบบส่วนมากที่สร้างตามแนวคิดนี้จึงมีการแนะนำเครื่องกลไกที่ยอมให้แบบแผน ( scheme ) พื้นฐานขยายออกไปได้แบบไดนามิก การสร้างแท็ก, ฟังก์ชัน ( function ) และออบเจกต์ ( object ) ขึ้นมาใหม่สามารถขยายวิธี template เพื่อการรับข้อมูล และจะไม่ยอมให้มีการควบคุมการแสดงของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมา ระบบ templating ส่วนมากยอมให้สำหรับบางรูปแบบของ scripting ซึ่ง scripting นี้ยอมให้อิลิเมนต์ที่เป็นภาษาโปรแกรมมีงั้นเต็มๆ ถูกรวมอยู่ในเพจ วิธีนี้เป็นการรวมโค้ดกับ HTML อีกครั้งหนึ่ง แต่ตั้งแต่งานหนักส่วนมากถูกทำให้สำเร็จในออบเจกต์หรือฟังก์ชัน ผู้เขียน HTML ก็ต้องการรู้เพียงโครงสร้างการควบคุมความสัมพันธ์อย่างง่าย ๆ เช่น การเปรียบเทียบค่า 2 ค่า หรือการวนลูป (loop) ตามค่าที่ตั้งไว้ พร้อมด้วยการแสดงบางอย่างในแต่ละครั้ง

#### 4.4.1 Active Server Pages

ระบบ templating ที่เป็นที่รู้จักกันดีและใช้กันมากที่สุดคือ Active Server Pages ( ASPs ) ของไมโครซอฟต์ ( Microsoft ) มันสนับสนุนสคริปต์ทั้ง Visual Basic และ JScript มันยอมให้โปรแกรมเมอร์ขยายระบบโดยสร้างออบเจกต์ Active X ( Active X objects ) ที่ ASPs สามารถเข้าไปได้ นอกจากนั้นไมโครซอฟต์ ยังใช้ความพยายามอย่างมากที่จะทำให้ ASPs มีประสิทธิภาพมากขึ้น โดยการรวมมันเข้ากับ IIS ที่เป็นเว็บเซิร์ฟเวอร์ของพวกเขา

ASPs ที่รวมเข้ากันกับ IIS และ Active X เป็นวิธีแก้ปัญหาของไมโครซอฟต์ ที่จะกำหนดขอบเขตหนทางที่จะวัดขนาดของไซต์ได้ถ้ามันกลายเป็นที่นิยมขึ้นมา ในการร้องขอไฟล์ .asp ยังคงต้องอ่านและแปลโดย scripting engine ถ้าไซต์ถูกสร้างขึ้นโดยใช้ IIS กับ ASP มันจะไม่สามารถรัน ( run ) บนเซิร์ฟเวอร์อื่นได้ แต่ถ้าเป็นการสร้างเพจและโค้ดสำหรับไซต์จะสามารถเขียนเพียงหนึ่งครั้งแล้วนำไปรันได้ทุกที่

#### 4.4.2 Servlets และ JavaServer Pages

ย้อนกลับไปที่ปัญหาของ CGIs คือการเริ่มใหม่ ( restart ) ของโปรแกรมทุกครั้งที่มีการร้องขอทำให้เสียเวลา สาเหตุมาจากโปรแกรมไม่ฝังแน่นระหว่างการเชื่อมต่อ ( connections ) มันจึงยากที่จะรักษาสถานะเอาไว้ มีความเป็นไปได้ที่จะแก้ปัญหาด้วย Fast CGIs หรือ Server extension แต่ถ้าทำอย่างนี้จะทำให้การเพิ่มฟังก์ชันใหม่ค่อนข้างยาก วิธีการของ Sun คือการใช้ Servlets ถ้า applet เป็นแอปพลิเคชันขนาดเล็กที่ขยายการทำงานของเว็บเบราว์เซอร์ Servlet ก็เป็นชิ้นส่วนเล็กๆของโค้ดที่ขยายการทำงานของเซิร์ฟเวอร์

Servlet เขียนให้เป็นออบเจกต์ในภาษาจาวา ที่ติดตั้งขึ้นเพื่อรับการร้องขอ, สร้างและส่งผลตอบกลับ การใช้ภาษาจาวา ในการเขียน Servlet ทำให้ได้รับข้อดีของภาษามาด้วยรวมถึงความเร็วที่ถึงแม้ว่าจะช้ากว่าภาษา C แต่ก็ยังเร็วกว่าภาษาอื่นๆ และที่สำคัญอีกอย่างคือจาวาสามารถพัฒนาได้ภายใต้หลาย ๆ แพลตฟอร์ม เช่น Linux, NT หรือ Solaris โดยไม่จำเป็นต้องเปลี่ยนแปลงหรือคอมไพล์ใหม่ ( recompile )

สิ่งที่น่าสนใจสำหรับผู้พัฒนาเว็บคือ มันสามารถกำจัดปัญหาในการขยายตัวของเว็บเซิร์ฟเวอร์ ถ้าเซิร์ฟเวอร์สนับสนุนการใช้งานจาวา มันก็สามารถโหลด Servlet ใหม่ได้แค่เพียงบอกให้ทำโดยไม่ต้องแก้ไขเซิร์ฟเวอร์

สถาปัตยกรรมของ Servlet ถูกออกแบบมาให้กำจัดความต้องการการโหลดใหม่ทุกอย่างทุกครั้งที่มีการร้องขอ มันจะโหลดในครั้งแรกเพียงครั้งเดียวแล้วมันจะยังคงอยู่ทำงานได้ ทำให้ตอบกลับได้เร็วเท่าที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

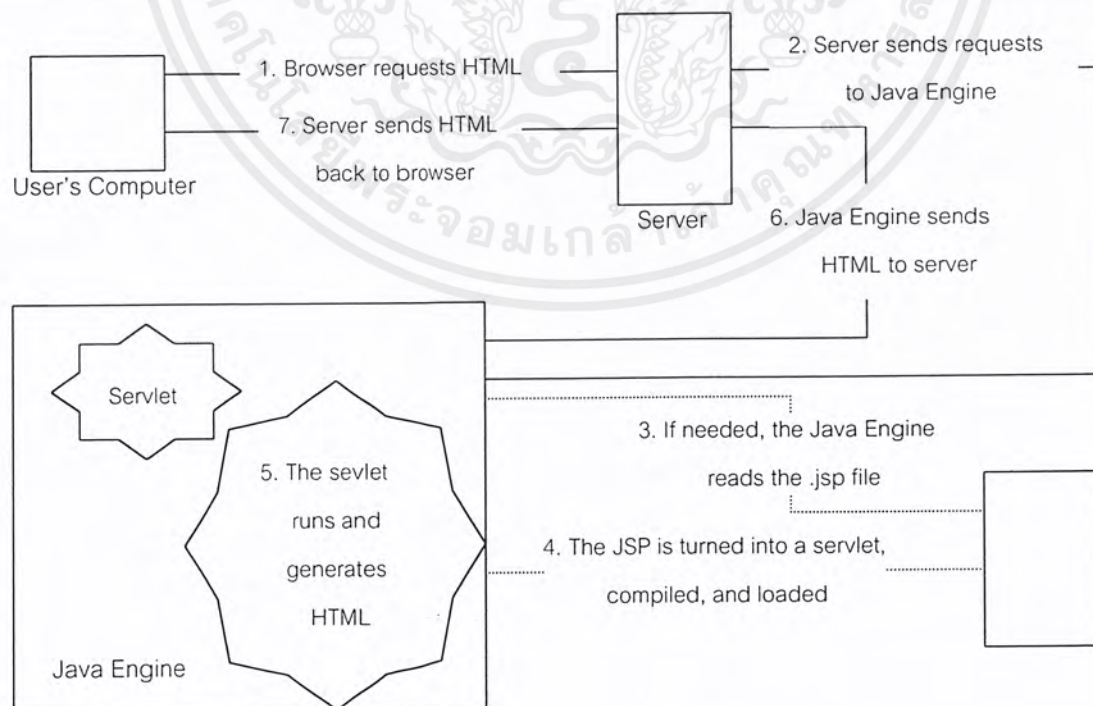
เซิร์ฟเวอร์สามารถทำได้ Servlets จึงสามารถแทนที่ CGI's ได้สมบูรณ์แบบโดยไม่มี downside และ ประโยชน์อีกอย่างของ Servlets คืออยู่ในระบบที่ปลอดภัยเพราะใช้ภาษาที่ออกแบบมาเพื่อความปลอดภัย

ขั้นตอนแบบลोजิกจะสร้างระบบ templating บนบนสุดของ Servlets เหมือนกับที่ ASPs ถูกสร้าง บนบนสุดของ CGI's อย่างไรก็ตาม บริษัท Boston กลุ่ม ATG ( Art Technology Group ) ได้มีแนวคิดใหม่ คือ จะทำการแปลเท็กพิเศษไปเป็นจาวาโดยตรง จากนั้นทำการคอมไพล์และรันโค้ดจาวานั้น วิธีนี้ในครั้งแรกที่ทำการร้องขอเพจมา จะทำการแปลซึ่งใช้เวลาพอสมควร แต่หลังจากนั้นเมื่อมีการร้องขอเข้ามาอีก จะสามารถตอบสนองได้เร็วเท่ากับ Servlet บริษัท Sun ซื้อเทคโนโลยีนี้มาเพื่อทำ Java Web Server แต่ก็ยังมีปัญหาคือ เท็กพิเศษมีทั้งแบบยุ่งยาก แบบที่ถูกจำกัด และแบบที่ไม่เหมือนเท็กเลย ทำให้ Sun ต้องแก้ไข เท็กเหล่านี้ใหม่โดยการสร้าง JSPs ( JavaServer Pages )

JSPs ได้รวมลักษณะที่ดีเอาไว้หลายอย่าง มันใช้ภาษาจาวา ดังนั้นมันจึงสามารถใช้ข้ามแพลตฟอร์มได้ และยังสืบทอดความสามารถอื่น ๆ ของจาวาอีกด้วย มันถูกสร้างบนบนสุดของ Servlets ทำให้เปลี่ยนแปลงมันได้ง่ายและรวดเร็ว มีความสามารถในการขยายได้ และโปรแกรมเมอร์สามารถสร้าง ออบเจกต์ใหม่กับฟังก์ชันการทำงานที่ใช้ Java beans ได้ง่าย ในขณะที่ผู้เขียนเพจก็สามารถใช้งานได้ง่ายเช่นกัน

#### 4.5 JSPs แบบง่าย ( Simple JSPs )

ลักษณะง่าย ๆ ของ JSPs เริ่มต้นจากที่บริษัทได้ติดตั้ง Tomcat เพื่อความพร้อมในการสร้างไซต์ของพวกเค้าที่โฮมเพจ ( home page ) ใหม่ ไฟล์ JSPs เบื้องต้นมีหน้าตาคล้ายกับเอกสาร HTML แต่จะตั้งชื่อลงท้ายด้วย .jsp แสดงว่าการเปลี่ยนเอกสาร HTML ไปเป็น JSPs นั้นทำได้ง่าย



รูปที่ 4-1 การไหลของข้อมูลของเพจที่มีการใช้ Servlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเป็นไฟล์ HTML เว็บเซิร์ฟเวอร์จะทำการอ่านเนื้อหาอย่างง่าย ๆ และส่งมันไปยังเบราว์เซอร์ แต่ ถ้าเป็นไฟล์ JSPs เว็บเซิร์ฟเวอร์จะถามไปยัง JSP Engine แล้ว JSP Engine จะแปลงไฟล์ให้เป็นโปรแกรม จาวาที่เรียกว่า servlet จากนั้นจะส่งการร้องขอไปยัง servlet และ servlet ก็จะสร้างผลตอบสนอง (response) ออกมาแล้วจึงส่งผลกลับไปให้เบราว์เซอร์ ดังแสดงในรูปที่ 4-1

แต่การทำงานจริงจะซับซ้อนเกินกว่าที่ไฟล์จะออกมาในรูปของเอกสาร HTML ธรรมดา จะมี โปรแกรมมากมายอยู่ในนั้น ดังนั้นจึงเกิด JSP Engine ขึ้นมา เพื่อช่วยรับการทำงานที่ซับซ้อนของ โปรแกรม มันแปลงแท็กธรรมดาให้เป็นโค้ดที่มีความซับซ้อนมาก ๆ แล้วก็สร้าง servlet และสิ่งแวดล้อม สำหรับรันโค้ดนั้น ๆ

#### 4.5.1 คำอธิบาย ( Comments )

เอกสาร HTML โดยทั่วไปแล้วจะมีคำอธิบายที่ผู้ใช้จะสามารถเห็นได้จากการ view source มันมี ประโยชน์ต่อผู้สร้างแต่จะไม่มีประโยชน์เลยสำหรับผู้อ่านธรรมดา อีกทั้งคำอธิบายอาจแสดงรายละเอียดที่ เป็นความลับหรือแสดงความรู้สึกส่วนตัว ซึ่งมันเป็นเรื่องน่าอายที่ใครจะอ่าน การแก้ปัญหานี้ของ JSP คือ แทนที่คำอธิบายของ HTML ด้วยคำอธิบายของ JSP ที่มีรูปแบบดังตัวอย่าง

รูปแบบคำอธิบายของ HTML ( HTML Comment )

```
<!-- Start Header -->
```

รูปแบบคำอธิบายของ JSP ( JSP Comment )

```
<%-- Start Header --%>
```

หลังจากที่แทนที่คำอธิบายแล้ว เมื่อ JSP Engine เห็นแท็กนี้มันจะรู้ว่าเป็นคำอธิบายแล้วก็จะไม่ ส่งไปที่ servlet ที่สร้างขึ้น ดังนั้นคำอธิบายก็จะไม่ถูกส่งไปให้ผู้ใช้และจะไม่แสดงให้เห็นใน view source ด้วย

#### 4.5.2 Templating

หน้าดัชนี ( index page ) มีองค์ประกอบหลัก 3 อย่าง คือ เฮดเดอร์ ( header ), เนวิกชัน และ ส่วน ของเนื้อหา เพจต่างกันเนื้อหาที่ต่างกัน แต่ก็ที่เหตุผลที่จะคาดหวังว่าเฮดเดอร์ และเนวิกชันซ้ำกันตลอดทั้ง 'ไซต์' ซึ่งมันเป็นปัญหาใหญ่อันหนึ่ง เพราะผู้เขียนเพจใหม่ต้องจำว่า ต้องใส่ชิ้นส่วนเหล่านี้ลงไป และต้อง คอยกังวลว่าทุกอย่างถูกต้อง และในกรณีที่แย่ที่สุดคือ เมื่อมีการแนะนำชิ้นส่วนใหม่เข้ามาทุกคนจะต้อง กลับไปสร้างเพจขึ้นมาใหม่ วิธีการแก้ไขคือ templating ในที่นี้ template เป็นเพียง HTML page ที่มีเนื้อที่ สำหรับถ้อยความและตัวชี้ว่าถ้อยความควรถูกพบที่ใด มันทำให้สามารถเก็บเฮดเดอร์ไว้ในไฟล์หนึ่ง และ แต่ละเพจจะเอาไฟล์นี้ไปใส่ไว้ในเนื้อที่ของมัน ดังนั้นจึงมีการแยกเก็บเฮดเดอร์และเนวิกชันไว้คนละ 'ไฟล์' จะได้ไฟล์ header.jsp กับ navigation.jsp

JSP เตรียมการสำหรับ templating เรียกว่า include directive ที่ยอมให้ไฟล์ JSP นำไปรวมกับไฟล์ JSP หรือ HTML อื่นๆ เหมือนกับว่าสามารถเรียกเพจได้โดยตรง โดยที่ไฟล์ HTML มาจากที่ไหนก็ได้บน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ ไฟล์ JSP ต้องมาจากไดเรกทอรีที่ชื่อว่า JSP ( JSP directory ) และอิลิเมนต์ที่เราต้องการนำมา  
รวมเป็นสิ่งที่อยู่ทั่วโลก ( global ) จึงควรสร้างไดเรกทอรีชื่อ global ภายใต้ไดเรกทอรี JSP

จากการที่แยกไฟล์เฮดเดอร์และเนวิเกชันเราจะสามารถเรียกกลับมาใช้ในโฮมเพจได้ด้วยแท็ก

```
<%@include file="global/header.jsp"%>
```

```
<%@include file="global/navigation.jsp"%>
```

การแยกไฟล์นี้มีข้อดีคือ เมื่อมีการเปลี่ยนแปลงโครงสร้างของอิลิเมนต์จะมีการเปลี่ยนแปลงเพียง  
เล็กน้อย แต่ JSP ก็ยังเหมือนเทคนิค templating ตัวอื่นๆเรียกว่าการรวมทางฝั่งเซิร์ฟเวอร์ ( server-side  
include ) ทำให้เซิร์ฟเวอร์รับภาระหนักมาก และอาจทำให้การทำงานช้าลงอย่างสังเกตเห็นได้ชัด

#### 4.5.3 การรวมไฟล์

แต่ละ JSP จะผ่านขั้นตอนที่ต่างกัน 2 ขั้น ขั้นตอนแรก เมื่อ JSP Engine เปลี่ยนไฟล์เป็น servlet  
เรียกว่าช่วงการแปล ( translation time ) ขั้นตอนที่ 2 คือ เมื่อผลของ servlet ถูกรันทำให้เกิดเพจเรียกว่าช่วง  
การร้องขอ ( request time ) ทั้ง 2 ขั้นตอนนี้มีความแตกต่างกัน การแบ่งแยกให้ถูกจึงมีความสำคัญ

การรวมไฟล์โดยตรง หรือเรียกว่า include directive ถูกดำเนินการโดย JSP Engine ในช่วงการ  
แปล มันถูกเรียกว่า directive เพราะมันควบคุม JSP Engine ให้ทำบางอย่างโดยเฉพาะการเปิดไฟล์ที่จะนำ  
มารวมและจัดวางเนื้อหาใน servlet ที่สร้างขึ้น โดยตรง ด้านดีคือ servlet จะไม่ต้องเปิดและรวมไฟล์ทุกครั้ง  
ที่มันรัน ดังนั้นมันจะรันได้เร็วขึ้น แต่ด้านเสียคือ ถ้าไฟล์ที่นำมารวมเกิดการเปลี่ยนแปลง servlet จะไม่มี  
ทางรู้เลยว่ามันจะต้องเปลี่ยนแปลงตัวเองจริงๆแล้วมันไม่ใช่ปัญหาร้ายแรง เพราะ JSP Engine จะทำการ  
สร้าง servlet ใหม่เมื่อมีการรวมไฟล์อื่นเข้ามา ภายใต้การทำงานของ Tomcat มันจำเป็นที่จะต้อง Shut  
down และ Restart เซิร์ฟเวอร์

แม้ว่า JSP จะสามารถตัดคำอธิบายออกจากผล HTML ในการทำไฟล์หลัก แต่ก็ไม่สามารถรับ  
ประกันได้ว่าจะทำได้ผลเดียวกันในไฟล์ที่นำมารวม เพราะไฟล์ที่นำมารวมสามารถมีไฟล์อื่นเข้ามา  
ประกอบด้วยและมีต่อไปได้เรื่อยๆไม่สิ้นสุด

#### 4.5.4 ข้อผิดพลาด ( Errors )

เมื่อมีการเขียนเพจ JSP ไม่ถูกต้องชัดเจน JSP Engine ก็ไม่สามารถทำงานต่อไปได้ และจะแจ้ง  
ความผิดพลาดไปยังหน้านั้น ข้อผิดพลาดที่พบบ่อยคือการลืมปิดแท็ก โดยเฉพาะแท็กคำอธิบายที่อาจปิด  
แท็กแบบ HTML แทนที่จะเป็นแบบ JSP เช่นการลืม % ปิดท้ายดังนี้ <%- - Text -> นอกจากนี้ก็ยังมีข้อ  
ผิดพลาดที่เกิดจากการรวมไฟล์ เช่น การพิมพ์ชื่อไฟล์ที่จะนำมารวมไม่ตรงกับชื่อไฟล์ที่มี, การพยายาม  
เรียกไฟล์ตัวเอง หรือไฟล์ที่นำมารวมมีการเรียกไฟล์หลัก เป็นต้น แต่ JSP Engine รุ่นใหม่จะสามารถแจ้ง  
ข้อมูลความผิดพลาดได้ ภายใต้การทำงานของ Tomcat มันจะแจ้งว่าอะไรคือปัญหาที่บรรทัดแรก แต่ก็ยัง  
ไม่สามารถระบุไฟล์และบรรทัดที่ผิดพลาดได้

#### 4.6 Request-Time Expressions

การแปลจะเกิดขึ้นเพียงครั้งเดียวในครั้งแรกที่มีการร้องขอมาจากผู้ใช้ แต่เนื่องจะทำให้มีเนื้อหาแบบไดนามิก มันจึงต้องมีการสร้างใหม่ทุกครั้งที่มีการร้องขอ ซึ่งจะอยู่ในกระบวนการที่อยู่ในช่วงการร้องขอ เป็นเวลาที่ servlet ถูกรันและให้ผลลัพธ์ออกมา

##### 4.6.1 Expressions

องค์ประกอบไดนามิกอย่างแรก คือ expression ซึ่งเป็นส่วนเล็ก ๆ ของโค้ดจาวา เมื่อมันถูกพบใน servlet มันจะเปลี่ยนไปเป็นอะไรก็ตามที่สามารถใส่ลงไปในเพจ expression ถูกแสดงโดยใส่มันไว้ในแท็กพิเศษ `<%= %>` เช่น `<%= 1+1 %>` จะให้ผลลัพธ์เท่ากับ 2 วิธีการก็คือเริ่มต้นจากไฟล์ JSP เดิม ผ่านไปยัง JSP Engine ไปเป็น servlet โดยทุก servlet จะส่งกลุ่ม HTML ที่บรรจุทุกอย่างที่ขึ้นกับ expression นั่นคือมันจะส่งกลุ่ม HTML ที่อยู่ก่อนหน้าการคำนวณไปก่อน จากนั้นก็ส่งผลจากการคำนวณค่าออกไป ตามด้วยส่วนที่อยู่ตามหลังการคำนวณ กล่าวอีกอย่างหนึ่งก็คือ ทุกอย่างที่อยู่ในแท็ก `<%= %>` จะต้องถูกคำนวณหรือหาค่า ดังนั้นจะเป็นสิ่งที่แสดงออกโดยตรงไม่ได้ JSP Engine ทำกับสิ่งที่อยู่นอกแท็กเป็นเหมือนข้อความธรรมดาที่จะถูกส่งกลับไปให้ผู้ใช้ แต่จะทำกับสิ่งอยู่ในแท็กในฐานะตัวแทนของบางสิ่ง และมันสร้าง servlet มาเพื่อหาว่าบางสิ่งนั้นคืออะไร ข้อดีของวิธีนี้คือ ในบางครั้ง expression อันเดียวกัน สามารถหมายถึงสิ่งที่ต่างกันในเวลาต่างกัน

สิ่งสำคัญในที่นี้คือแท็ก `<%= %>` เป็นส่วนประกอบอย่างแรกของเนื้อหาแบบไดนามิกเพื่อสร้างเพจแบบไดนามิก ชั้นแรกต้องตัดสินใจว่าอติเมนต์ใดจะเปลี่ยนไปสำหรับแต่ละการร้องขอ หรือแต่ละผู้ใช้ และสร้าง expression ที่เป็นตัวแทนข้อมูลที่จะเติมเต็มในอติเมนต์เหล่านั้น จากนั้นในช่วงการร้องขอ servlet จะใช้ expression ในการสร้างหรือหาเวอร์ชันที่เหมาะสมสำหรับเพจที่ผู้ใช้ถามหา แต่การสร้าง expression มันยากที่จะหาตัวแทนที่เหมาะสมต่อความต้องการ, ความเปลี่ยนแปลงได้ และเป็นข้อมูลที่มีประโยชน์ อย่างไรก็ตาม JSPs จะหาข้อมูลที่เป็นประโยชน์ที่สุดที่ผู้เขียนสามารถใช้ได้อย่างถูกต้อง มันถูกรวบรวมไว้และเรียกว่า implicit object ที่ผู้เขียนไม่จำเป็นต้องประกาศอย่างชัดเจนว่าจะใช้ออบเจกต์เหล่านี้ เพราะมันสามารถใช้ได้โดยอัตโนมัติ

##### 4.6.2 The Implicit Objects

Implicit Object มีจำนวนมาก ตัวแรกที่เราจะพิจารณาเรียกว่า request มันเป็นออบเจกต์ที่มีข้อมูลมากมายเกี่ยวกับการร้องขอที่เบราว์เซอร์ของผู้ใช้ส่งมา รวมถึงมีข้อมูลของเบราว์เซอร์และคอมพิวเตอร์ที่กำลังใช้ เพจที่กำลังถูกร้องขอและข้อมูลที่เบราว์เซอร์จะยอมรับเป็นผลตอบกลับ ตัวอย่างเช่น การขอชื่อคอมพิวเตอร์ของผู้ใช้ที่กำลังดูเพจนี้อยู่ จะใช้ expression ดังนี้ `<%= request.getRemoteHost( ) %>` สิ่งสำคัญคือ expression นี้จะรับออบเจกต์ request และจากนั้นจะใช้เมธอด ( method ) `getRemoteHost( )` เพื่อหาชื่อของคอมพิวเตอร์

Method	วัตถุประสงค์
GetCharacterEncoding( )	กลุ่มตัวอักษรที่เพจนีใช้เข้ารหัส ( เช่น ISO )
GetProtocol ( )	โพรโตคอลที่ใช้สำหรับการร้องขอ เช่น HTTP/1.0
GetScheme( )	ส่วน scheme ของ URL เช่น HTTP
GetServerName( )	ชื่อของคอมพิวเตอร์บนเซิร์ฟเวอร์ที่กำลังรัน
GetServerPort( )	พอร์ต ( port ) ของคอมพิวเตอร์ทางฝั่งเซิร์ฟเวอร์ที่กำลังใช้งาน
GetRemoteAddr( )	แอดเดรสของคอมพิวเตอร์ที่ส่งการร้องขอมา
GetRemoteHost( )	ชื่อของคอมพิวเตอร์ที่ส่งการร้องขอมา
GetAuthType( )	ชนิดของการอนุญาตที่ใช้สำหรับการร้องขอนี้ ( ถ้าเพจนีไม่มีการป้องกัน มันจะมีค่าเป็น null )
GetHeader( )	header ใดๆที่ถูกส่งมาจากเบราว์เซอร์
GetMethod( )	วิธีการที่ใช้สำหรับการร้องขอนี้ ( มักจะเป็น GET หรือ POST แต่อาจเป็น HEAD, PUT, DELETE เป็นต้น )
GetPathInfo( )	ข้อมูลเส้นทาง ( ถ้า JSP อยู่ที่ /stories.jsp และเบราว์เซอร์ขอ /stories.jsp/story1 ข้อมูลเส้นทางจะเป็น /story1 )
GetPathTranslated( )	เส้นทางที่ไฟล์ที่ถูกร้องขออยู่บนระบบไฟล์ของเซิร์ฟเวอร์
getQueryString( )	ส่วนคำถามของ URL ( เช่น อะไรก็ตามที่อยู่หลังเครื่องหมายคำถาม )
GetRemoteUser( )	ชื่อของผู้ใช้ที่เข้ามา ( ถ้าเพจนีไม่มีการป้องกัน มันจะมีค่าเป็น null )
GetRequseURI( )	การร้องขอ URL เต็มรูปแบบของ JSP

ตารางที่ 4-1 ตัวอย่างเมธอดในออบเจกต์ request

JSPs จะสามารถตอบสนองต่ออินพุตของผู้ใช้มันต้องสามารถรับตัวแปรที่มีรูปแบบได้ อย่างไรก็ตามมีสามปัญหาเกี่ยวกับการรับข้อมูลผ่านทาง `getQueryString( )`

1. ค่าทุกค่าจะรวมอยู่ด้วยกันทำให้ยากที่จะดึงเอาการตอบสนองต่อคำถามเฉพาะออกมาได้
2. ทุกเนื้อที่และตัวอักษรที่ไม่เป็นตัวเลขและตัวพยัญชนะจะต้องถูกเปลี่ยนไปเป็นตัวอักษรอื่น มันจำเป็นต้องทำ เพราะเซิร์ฟเวอร์และเบราว์เซอร์อาจจะสับสนกับมัน เช่น เครื่องหมายทับ ( / ) อาจจะปรากฏอยู่ในกลุ่มตัวอักษร แต่เซิร์ฟเวอร์และเบราว์เซอร์จะไม่มีทางรู้ว่าเครื่องหมายทับนี้ส่งมาเป็นรูปแบบหนึ่งหรือเป็นส่วนหนึ่งของชื่อใดเรกทอรี
3. ยังมีอีกวิธีหนึ่งสำหรับส่งรูปแบบของข้อมูลโดยไม่ใช่ query string เลย ถ้าวิธีการ GET ในเพจ HTML ถูกเปลี่ยนไปเป็นวิธีการ POST ข้อมูลก็จะไม่แสดงใน URL และจะถูกนำกลับมาโดย `getQueryString( )`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธี POST มักจะดีกว่าวิธี GET ทั้งสองวิธีสามารถจัดการกับข้อมูลจำนวนมากๆได้ เพราะมันไม่แสดงใน URL ถ้าข้อมูลที่เป็นส่วนตัวหรือเป็นความลับถูกส่งไป ผู้ใช้อาจจะไม่ต้องการให้ใครอ่านผลตอบกลับให้ เพียงดูได้ที่ URL บนเบราว์เซอร์ของเขา

โชคดีที่การร้องขอออบเจกต์มีหลายวิธีที่จะทำให้มันง่ายขึ้นในการรับตัวแปรที่มีรูปแบบ อย่างแรกคือ `getParameter()` และต้องทำการตั้งชื่อให้กับพารามิเตอร์ที่จะรับด้วย มีตัวอย่างดังนี้

```
<%= request.getParameter("conferences") %>
```

#### 4.6.3 A First Application

เราจะเริ่มต้นที่การคำนวณง่ายๆ ที่นำเอาสองจำนวนมาบวกกัน ขั้นแรกจะต้องมีทางให้ผู้ใช้ใส่ค่าเข้ามา ดังนี้

```
<HTML>
<HEAD><TITLE>A simple calculator</TITLE></HEAD>
<BODY>
<P>
Enter two numbers and click the 'calculate' button.
</P>
<FORM ACTION="calculate.jsp" METHOD="GET">
<INPUT TYPE="TEXT" NAME="value1"><BR>
<INPUT TYPE="TEXT" NAME="value1"><BR>
<INPUT TYPE="SUBMIT" NAME="Calculate" VALUE="Calculate">
</FORM>
</BODY>
</HTML>
```

ในที่นี้โครงสร้างของเพจผลลัพธ์ควรจะดูง่ายเพียงใช้ออบเจกต์ `request` ในการรับค่า และบวกค่าเข้าด้วยกัน แต่ผลลัพธ์ที่ได้จะผิด ถ้าเขียนดังตัวอย่าง

```
<HTML>
<HEAD><TITLE>A simple calculator: results</TITLE></HEAD>
<BODY>
<P>
The sum of your two numbers is
<%= request.getParameter("value1") +
request.getParameter("value2") %>
</P>
</BODY>
</HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าใส่ค่าเข้าไปเป็น 8 กับ 53 ในตัวอย่างข้างต้น จะได้ผลลัพธ์ออกมาเป็น 853 ซึ่งเป็นผลลัพธ์ที่ไม่ถูกต้อง ปัญหานี้เกิดจากเมธอด `getParameter()` ส่งค่ากลับมาเป็น `string` ไม่ใช่ `integer` ทำให้กลายเป็นการบวก `string` ที่นำอักขระมาต่อกัน สามารถแก้ไขได้โดยแปลง `string` ให้เป็น `integer` ก่อนที่จะทำการคำนวณ ซึ่งในภาษาจาวาจะมีเมธอด `parseInt()` สำหรับแปลง `string` ให้เป็น `integer` ดังนั้นจะเขียนให้ถูกต้องได้ดังนี้

```
<HTML>
<HEAD><TITLE>A simple calculator: results</TITLE></HEAD>
<BODY>
<P>
The sum of your two numbers is
<%= Integer.parseInt(request.getParameter("value1")) +
    Integer.parseInt(request.getParameter("value2")) %>
</P>
</BODY>
</HTML>
```

กลวิธีของมันคือ ต้องมีความรู้เกี่ยวกับคลาส `Integer` กับเมธอดของ `Integer` ถึงแม้ `JSPs` จะช่วยรับภาระด้านโปรแกรมมาจากผู้แต่งเพจ แต่ก็ควรจะต้องรู้พื้นฐานของจาวา คลาสเหล่านี้จะเป็นเครื่องมือที่ช่วยให้ทำงานได้ง่ายขึ้น

#### 4.7 การใช้ Beans

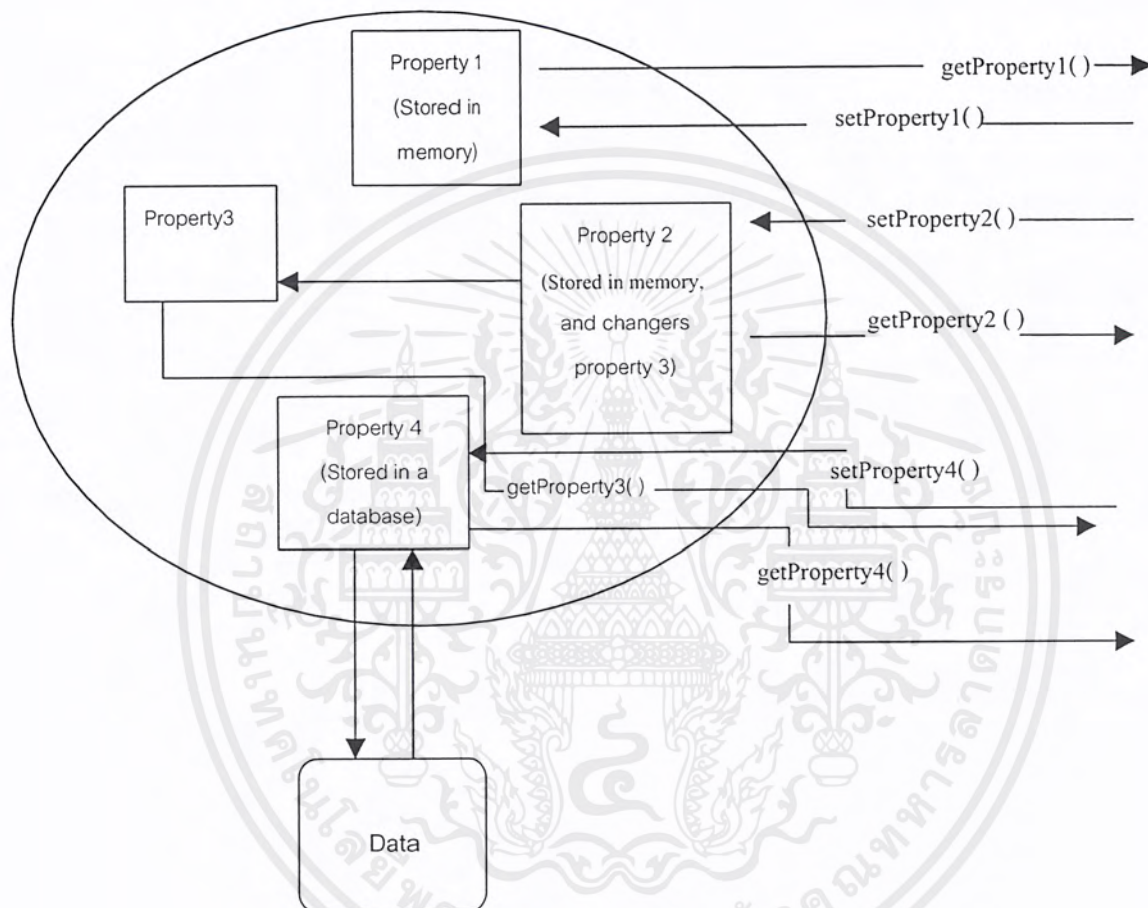
ถึงแม้ว่าจะสามารถทำเพจไดนามิกแบบง่าย ๆ ได้ โดยทำเพจให้เปลี่ยนไปตามอินพุตของผู้ใช้ หรือข้อมูลที่ถูกส่งมาจากเบราว์เซอร์ แต่ก็ยังไม่สามารถทำให้เกิดการกระทำที่แตกต่างกันสำหรับแต่ละอินพุต การสนับสนุนอย่างหนึ่งของ `JSPs` คือ การดึงเอาโค้ดออกนอกเพจและวางไว้ใน `bean` ซึ่งสามารถใช้ข้อมูลร่วมกันกับเพจอื่นๆ

##### 4.7.1 อะไรคือ Beans

`Java bean` คือตัวที่จะตั้งค่าคุณสมบัติที่สามารถอ่านหรือเปลี่ยนแปลงได้ มันเป็นไปได้ที่จะค้นหาคุณสมบัติที่ `bean` มีอยู่ ซึ่งเหมาะสำหรับรูปแบบทุกประเภทของโลกแห่งความจริงที่สามารถอธิบายออกมาเป็นชุดของคุณสมบัติ ตัวอย่างเช่น `CD` สามารถทำให้เป็นรูปแบบของ `bean` ด้วยคุณสมบัติที่แสดงถึงปีที่มันออกจำหน่าย, บริษัทที่ผลิต, และอื่นๆ

สองสิ่งที่ทำให้ `bean` มีประโยชน์เป็นพิเศษ อย่างแรกคือทั้งโปรแกรมเมอร์และผู้แต่ง `JSP` ไม่จำเป็นต้องรู้ทุกอย่างเกี่ยวกับ `bean` เพื่อใช้มัน `CD bean` สามารถจัดซื้อจากบริษัทที่ผลิต `bean` และมันต้องการเพียงแค่ติดตั้งไว้บนระบบสำหรับผู้เขียน `JSP` จะสามารถใช้มันได้ทันที ประโยชน์อย่างที่สองเกี่ยวกับ `bean` คือ มันไม่จำเป็นต้องทำการกู้คืนหรือเปลี่ยนแปลงคุณสมบัติ บางทีการร้องขอเพื่อจะรับคุณสมบัติต้องการ `bean` เพื่อหาข้อมูลในฐานข้อมูล และในบางครั้งเมื่อคุณสมบัติของ `bean` ถูกเปลี่ยนแปลงไป มันเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งจดหมายไปยังผู้บริหาร ในความเป็นจริง ทั้งการเข้าถึงและการเปลี่ยนแปลงคุณสมบัติสามารถ กระตุ้นความต้องการ ความซับซ้อนของการกระทำ แต่ผู้เขียน JSP ไม่ต้องกังวลกับเรื่องพวกนี้ด้วยเหตุผล ที่ว่า bean ทำตัวเหมือนเป็นกล่องคำลึกลับที่เปิดปิด และอ่านข้อมูลออกมาได้ ผู้เขียนเพจสามารถหมุนปุ่ม เพื่อเปลี่ยนคุณสมบัติและอ่านคุณสมบัติโดยไม่ต้องรู้ว่าข้างในกล่องทำอะไร รูปแบบของ Java beans แสดงดังรูป



รูปที่ 4-2 รูปแบบของ Java beans

นี่เป็นสิ่งที่ทำให้ bean สามารถแบ่งโปรแกรมออกจากการเขียนเพจได้ โปรแกรมเมอร์ใส่ลอจิก ทั้งหมดเข้าไปใน bean และกำหนดคุณสมบัติที่กระตุ้นลอจิก ผู้เขียนเพจใช้คุณสมบัติในการหาค่าที่จะอยู่ บนเพจ การเข้าถึงนี้ต้องผ่านแท็กใหม่ 3 แท็ก

#### 4.7.2 การใช้ Beans

JSPs เตรียม 3 แท็กสำหรับทำงานกับ bean ตัวแรกสำหรับหาและใช้ bean ตัวที่สองเพื่อรับคุณสมบัติ และตัวที่สามใช้สร้างคุณสมบัติหนึ่งหรือมากกว่าหนึ่งตัว มีการใช้ bean ได้หลายทาง ดังนั้นการใช้ แท็กจึงมีหลายวิธี รูปแบบที่ใช้มากที่สุดที่จะทำให้ใช้ bean ได้ใน JSP คือการใช้แท็กดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<jsp:useBean id="bean name" class="bean class" />
```

ในที่นี้ bean name เป็นชื่อที่อ้างถึง bean มีข้อจำกัด 2 ข้อในการตั้งชื่อ คือ มันจะต้องใช้ตัวอักษรที่จำวามืออยู่ กับต้องมีความโดดเด่นแตกต่างกับตัวอื่น ๆ ในที่ที่มันถูกใช้ หมายความว่าสำหรับ bean ที่ต่างกันจะต้องมีชื่อไม่เหมือนกัน ส่วน bean class เป็นชื่อของคลาสจาวาที่กำหนดให้เป็น bean และคลาส bean ต้องมีอยู่ใน JSP Engine ดังนั้นมันต้องอยู่ที่ใดที่หนึ่งใน CLASSPATH รายละเอียดในการทำ CLASSPATH จะเปลี่ยนไประหว่างการติดตั้ง แต่ส่วนมากจะเตรียมไคเรกทอรีมาตรฐานไว้ให้ bean ใหม่สามารถติดตั้งได้ เครื่องหมายทับที่ท้ายแท็กก็มีความสำคัญ มันเป็นสัญญาณบอกว่า ไม่มีแท็ก <jsp:useBean> เป็นแท็กปิด และจะมีการตรวจสอบความผิดพลาดที่ช่วงการแปล นอกจากนี้ยังมีการใช้แท็ก useBean อีกวิธีหนึ่งที่รับพารามิเตอร์ชนิดที่เป็นเหมือนคลาส มีรูปแบบดังตัวอย่างนี้

```
<jsp:useBean id="name" class="bean class" type="another class">
```

#### 4.7.3 การรับคุณสมบัติ ( Getting a Property )

bean ที่บรรจุด้วย jsp:useBean จะรับคุณสมบัติได้ง่ายโดยใช้แท็ก getProperty ดังตัวอย่าง

```
<jsp:getProperty name="bean name" property="property name" />
```

bean name เป็นชื่อเดียวกับที่ใช้ใน id และ property name คือ ชื่อของคุณสมบัติที่จะรับมา สิ่งแรก que JSP ทำคือรับ bean เข้ามา ที่นี้ ID จะมีความสัมพันธ์บางอย่างกับคลาส name มันไม่จำเป็นที่แท็ก useBean จะต้องปรากฏบนส่วนบนของเพจ ถึงแม้ว่ามันจะต้องปรากฏก่อนแท็ก getProperty

ครั้งหนึ่งที่ bean ถูกโหลด จะได้รับคุณสมบัติหลายอย่างจากมัน สิ่งนี้ทำการซ่อนรายละเอียดของโปรแกรมไว้จากผู้เขียนเพจ แท็ก getProperty อาจคล้ายแท็ก expression อย่างน้อยที่สุดทั้งคู่ก็เป็นตัวใส่ค่าเข้าไปในเพจ ในความเป็นจริงแท็ก getProperty สามารถแปลงไปเป็นแท็ก expression ที่เรียกว่าวิธีการ GET โดยตรงของ bean ตัวอย่างเช่น

```
<jsp:getProperty name="bean1" property="seventhPrimeNumber" />
```

สามารถเขียนเป็น

```
<%= bean1.getSeventhPrimeNumber() %>
```

#### 4.7.4 การตั้งคุณสมบัติ ( Setting Properties )

มีหลายทางที่จะตั้งคุณสมบัติของ bean แต่ที่ดูง่ายที่สุดน่าจะคล้ายกับการรับคุณสมบัติ

```
<jsp:setProperty name="bean name" property="property name" value="property value" />
```

name ในที่นี้ก็เหมือนกับก่อนหน้านี้คือเป็น ID และ property name ก็คือชื่อของคุณสมบัติที่ดูตั้งขึ้นมา value เป็นค่าใหม่ที่จะกำหนดให้แก่คุณสมบัติ ลักษณะที่ธรรมดาที่สุดของ value คือ string ที่ปิดด้วยอัญประกาศ เช่น "red" หรือ "3"

ค่าที่กำหนดตายตัวดีสำหรับหลายจุดประสงค์ แต่สำหรับ bean ที่มีส่วนเกี่ยวข้องกับเพจแบบไดนามิกนั้นต้องสามารถทำงานร่วมกับอิลิเมนต์ไดนามิกตัวอื่นๆ ทางหนึ่งที่จะทำให้สำเร็จคือใช้ expression

ในการตั้งคุณสมบัติแท็ก setProperty สามารถยอมรับค่าใดๆที่ expression สามารถใส่เข้าไปในแท็ก <%= %>

#### 4.7.5 การเชื่อมต่อระหว่างฟอร์มกับคุณสมบัติ

ตั้งแต่ลอจิกโปรแกรมส่วนใหญ่อยู่ทางฝั่ง bean มันคือเป็นธรรมชาติที่ JSPs รับค่าอินพุตมาจาก ฟอร์ม ผ่านค่าเหล่านี้เข้าไปใน bean ผ่านแท็ก setProperty และจากนั้นแสดงคุณสมบัติอื่นที่เป็นตัวแทนผล ของการประมวลผล ค่าฟอร์มสามารถส่งผ่านไปยัง bean โดยการรวมเทคนิคที่เพิ่งมีการพิจารณาไปแล้ว เช่น

```
<jsp:setProperty name="id" property="someproperty" value="<%= request.getParameter("formparam") %>">
```

ได้มีการเตรียมชื่อย่อ ( shortcut ) สำหรับผู้เขียน JSP ถ้าฟอร์มเตรียมค่าที่เรียกว่า formparam และถ้า bean มีคุณสมบัติที่เรียกว่า formparam จะได้ว่าโค้ดข้างบนสามารถถูกแทนที่ได้โดยแท็ก

```
<jsp:setProperty name="id" property="formparam" />
```

ในบางครั้งชื่อของฟอร์มพารามิเตอร์ ( form parameter ) กับชื่อของคุณสมบัติ จะไม่ตรงกัน พวก มันสามารถเชื่อมต่อกันได้จากการใช้แท็ก setProperty ได้ ดังต่อไปนี้

```
<jsp:setProperty name="id" property="property name" param="paramname" />
```

JSP ใช้ฟอร์มพารามิเตอร์ที่เรียกว่า paramname ในการตั้งคุณสมบัติที่เรียกว่า propertyname นอกจากนี้ยังมีเวอร์ชันสุดท้ายของแท็ก setProperty ซึ่งเต็มไปด้วยพลัง เวอร์ชันนี้มองว่า พารามิเตอร์ถูกเตรียม โดยฟอร์ม และเมรอดถูกเตรียมโดย bean และเชื่อมพวกมันเข้าด้วยกันโดยอัตโนมัติ ได้ออกมาเป็น

```
<jsp:setProperty name="id" property="*">
```

ถ้าฟอร์มเตรียมค่าหลายค่าและ bean เตรียมเมรอดหลายเมรอด ทุกอย่างจะจับคู่กันได้ ถ้าฟอร์ม เตรียมบางตัวแปรที่ไม่มีเมรอด ( ค่าของตัวแปรจะจับคู่กับเมรอดเสมอถึงจะนำไปใช้งานได้ ) ตัวแปรเหล่านี้ จะถูกละทิ้งไปและไม่เกิดการแจ้งความผิดพลาด แต่ JSP สามารถทำบางอย่างกับตัวแปรเหล่านั้น รวมถึง ส่งผ่านไปบน bean ได้ ในทางกลับกัน ถ้า bean เตรียมเมรอดที่ไม่มีค่ามาสนับสนุน เมรอดนี้จะไม่ถูกเรียก แต่ JSP สามารถเรียกมันได้ถ้าจำเป็น

โค้ดต่อไปนี้แสดงถึงการคำนวณการบวกเลข 2 จำนวน โดยใช้ bean

```
<jsp:useBean id="calc" class="com.awl.jspbook.ch04.CalcBean"/>
```

```
<jsp:setProperty name="calc" property="*" />
```

```
<HTML>
```

```
<HEAD> <TITLE>A simple calculator: results</TITLE></HEAD>
```

```
<BODY>
```

```
<P>
```

```
The sum of your two numbers is
```

```
<jsp: getPorperty name="calc" porperty="sum"/>
```

```
</P>
```

```
</BODY>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<HTML>

เนื่องจากใช้ bean ทำให้มันง่ายที่จะคำนวณการคูณได้ในฟอร์มเดียวกัน แต่ถ้าผู้ใช้ใส่ค่าที่ไม่ใช่ตัวเลข ก็จะทำให้เกิดข้อผิดพลาดได้

#### 4.8 Bean Scopes

bean มีประโยชน์มาก แต่ก็ยังมีข้อจำกัดอยู่ ดังนั้นในหัวข้อนี้จะแสดงถึงการขยายขอบเขตข้อจำกัดของ bean หรือตัวแปร เพื่อยอมให้พฤติกรรมแบบไดนามิกใหม่ ๆ หลากหลายขึ้น

##### 4.8.1 ขอบเขตเพจ ( The Page Scope )

ขอบเขตเพจเป็นขอบเขตของ JSP ที่ธรรมดาที่สุด ออบเจ็กต์ที่มาพร้อมกับ page scope อาจจะถูกปรับปรุงบ่อยเท่าที่กับต้องการในเพจ แต่การเปลี่ยนแปลงเหล่านี้จะหายไปเมื่อออกจากเพจ ดังนั้นถ้าโหนด bean เป็นลำดับเข้าไปในขอบเขตเพจ แล้วคุณสมบัติหลาย ๆ อย่างของมันก็จะถูกเปลี่ยนแปลง แต่คนถัดไปที่เข้ามาที่เพจจะยังคงเห็นค่าดั้งเดิมอยู่

มันดูเหมือนกับว่าเพจจะยังคงเหมือนเดิมอยู่เสมอ ไม่ว่าจะเข้าไปกี่ครั้งก็ตาม และทุกสิ่งในขอบเขตเพจควรจะคงอยู่และนำมาใช้ได้เสมอสำหรับทุกการร้องขอ พุคอีกอย่างหนึ่งคือ มันอาจดูเหมือนว่าบางอย่างที่เรียกว่า ขอบเขตเพจ ควรจะมีพฤติกรรมเหมือน instance scope ตามแนวคิดนี้ URL ที่เดียวกันจะต้องให้เพจเหมือนกัน แต่ด้วย JSPs เมื่อใดก็ตามที่เข้ามาใน URL เพจใหม่ทั้งหมดก็จะถูกสร้างขึ้นในช่วงเวลาการร้องขอ และส่งไปให้ผู้ใช้ แนวคิดของเพจนี้เหมือนกับบางสิ่งถูกสร้างขึ้นในแต่ละครั้ง ให้ชื่อ ว่า ขอบเขตของเพจ

ตามปกติ ทุก bean มีขอบเขตของเพจ แต่ถ้าผู้เขียนเพจต้องการ แสดงออกมาอย่างชัดเจนว่า Scope="page" ก็สามารเพิ่มเข้าไปในแท็ก jsp:useBean ได้ดังนี้

```
<jsp:useBean id="id" class="beanclass" scope="page"/>
```

##### 4.8.2 ขอบเขตการร้องขอ ( The Request Scope )

ชนิดที่ 2 ของขอบเขต มีความสัมพันธ์กับออบเจ็กต์ request ออบเจ็กต์ใดๆที่สร้างในขอบเขตการร้องขอ จะมีใช้ได้นานเท่ากับที่ออบเจ็กต์ request คงอยู่ อาจจะดูไม่มีความแตกต่างจากขอบเขตเพจ เพราะสามารถสันนิษฐานได้ว่า ออบเจ็กต์ request มีอยู่นานเท่ากับเพจที่มันร้องขอ อย่างไรก็ตามก็ยังมียสิ่งที่ทำให้ request แบ่งแยกออกจากเพจได้

พิจารณา JSP เป็นแนวประกอบเครื่องในการทำของมาก ๆ ในโรงงานอุตสาหกรรม ( assembly line ) ที่เริ่มต้นด้วย request ที่ปลายข้างหนึ่ง และผลิต HTML ที่ปลายอีกข้างหนึ่ง สมมติให้เว็บเซิร์ฟเวอร์ หรือ Tomcat เป็นโรงงานใหญ่ที่มีแนวประกอบหลายสาย ตามปกติเว็บเซิร์ฟเวอร์จะวาง request ไว้บนสายเส้นหนึ่งเพื่อสร้างผลตอบสนองให้แก่ผู้ใช้ อย่างไรก็ตามไม่มีเหตุผลที่คนงานที่แนวประกอบหนึ่งจะไม่สามารถส่งผ่าน request ของเขาไปให้อีกคนบนอีกแนวประกอบหนึ่ง ที่ท้ายสายที่ 2 จะให้ผล HTML ส่งไปให้ผู้ใช้ และ request เดิมจะส่งผ่านกลับไปให้คนงานเดิมที่จะทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการ JSP สลับส่นรูปแบบ 2 อย่าง อย่างแรกคือ ที่ JSP เดิมไม่รับ request กลับ เรียกว่า JSP forward มีหลายเหตุผลว่าทำไม JSP หนึ่งจึงต้องมีการส่ง request ของมันไปให้อีกตัวหนึ่ง เช่นเหตุผลด้านความปลอดภัย ถ้าผู้ใช้พยายามเข้าถึงเพจที่ไม่ได้รับอนุญาต เพจก็จะส่งไปให้อีกเพจหนึ่ง ที่อธิบายว่าทำไมจึงปฏิเสธไม่ให้เข้า และบางทีอาจจะให้ทางเลือกอื่นในการใช้บริการ มันอาจเป็นส่วนของโค้ดต่อไปนี้

```
<% if user.isNotAuthorized( ) { %>
    <jsp:forward page="register.jsp">
<% } else { %>
    ... show the user the page ...
<% } %>
```

สิ่งนี้ไม่จำเป็นตั้งแต่เพจลงทะเบียน ( register page ) ใช้ if กับ else แต่วิธีการ forward request เป็นรูปแบบที่ดีสำหรับที่จะแสดงว่าอะไรที่เกิดขึ้น

รูปแบบที่ 2 การใช้ redirect คือการที่เว็บเซิร์ฟเวอร์บอกเบราว์เซอร์ว่าข้อมูลที่มันต้องการจริงๆอยู่ที่ URL อื่น ซึ่งเบราว์เซอร์จะถามเซิร์ฟเวอร์ว่า redirect มี 2 ปัญหา ปัญหาแรกคือ มีความล่าช้า เพราะมีการติดต่อสื่อสารมากระหว่างเบราว์เซอร์กับเซิร์ฟเวอร์ ปัญหาข้อที่สองคือ redirect บังคับชนิดของข้อมูลที่เพจแรกสามารถส่งให้เพจที่ 2 เพจที่ 2 จะไม่รู้ว่ามีผู้กำลังมาที่นี้ เพราะการใช้ redirect

การใช้ forward นั้น ชนิดของข้อมูลที่ส่งผ่านจากเพจแรกไปเพจที่ 2 เป็นชนิดใดก็ได้ ตั้งแต่ทั้ง 2 เพจมีออบเจกต์การร้องขอทั้งคู่ มันจึงสมเหตุสมผลที่จะส่งข้อมูลไปพร้อมกับการร้องขอ และนี่เป็นเหตุผลให้มีการใช้ขอบเขตการร้องขอ เพจแรกสามารถส่งข้อมูลใดๆในขอบเขตการร้องขอ และเพจที่ 2 สามารถมองหาข้อมูลและกระทำกับข้อมูลนั้นได้

คลาส request มี 2 เมธอดที่สนับสนุนการใส่ข้อมูลลงในขอบเขตการร้องขอ ตัวแรกเรียกว่า setAttribute( ) มันยอมให้ JSP เพิ่มแอตทริบิวต์ใหม่ ที่แสดงถึงข้อมูลไปสู่ request ตัวที่ 2 เรียกว่า getAttribute( ) มันยอมให้ JSP ู้คินค่าจากขอบเขตการร้องขอได้

ตัวอย่างต่อไปนี้แสดงรูปแบบง่ายๆที่ยอมให้ผู้ใช้เลือกเพจจุดหมายปลายทางได้ รูปแบบนี้จะเรียก JSP ที่จะใช้แท็ก forward เพื่อส่งการร้องขอไปยังเพจที่ถูกเลือก

```
<HTML>
<BODY>
<P>Which page would you like?</P>
<FORM ACTION="dispatcher.jsp">
<INPUT TYPE="radio" NAME="which" VALUE="red" CHECKED>red<BR>
<INPUT TYPE="radio" NAME="which" VALUE="green">green<BR>
<INPUT TYPE="radio" NAME="which" VALUE="blue">blue<BR>
What is your favorite number?
<INPUT TYPE="TEXT" NAME="favorite"><BR>
<INPUT TYPE="SUBMIT" NAME="Go" VALUE="Go">
</FORM>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</BODY>
```

```
</HTML>
```

ตัวอย่างต่อไปแสดง JSP ที่เก็บตัวเลขของผู้ใช้ในขอบเขตเพจ ใช้ scriptlet ในการเก็บค่าในออบเจ็กต์ request จะทำการหาเพจที่ร้องขอ และ forward การร้องขอไปให้มัน

```
<%- - Put the favorite number in the request scope,
    by placing it in the 'favorite' attribute
    of the request - -%>
<% request.setAttribute("favorite",
    request.getParameter("favorite")) ; %>
<%- - now dispatch to the user's request page - -%>
<% String which = request.getParameter("which") ; %>
<% if (which.equals("red")) { %>
    <jsp:forward page="/ch06/red.jsp"/>
<% } else if (which.equals("green")) { %>
    <jsp:forward page="/ch06/green.jsp"/>
<% } else if (which.equals("blue")) { %>
    <jsp:forward page="/ch06/blue.jsp"/>
<% } %>
<HTML>
<BODY>
You asked for a page which dose not exist!
</BODY>
</HTML>
```

จากตัวอย่างข้างต้น ถ้าเพจที่ร้องขอตรงกับเพจใดๆที่ JSP นี้รู้จัก ผู้ใช้จะเห็นผลเป็นเพจนั้น เหมือนกับไปที่เพจนั้นโดยตรง ถ้าการร้องขอเป็นเพจที่ไม่รู้จัก ผู้ใช้จะเห็น HTML ที่รวมอยู่ใน JSP นี้ นั่นคือถ้าใช้แท็ก forward ไป จะไม่กลับมาเพจนี้อีก และผู้ใช้จะไม่เห็นการแจ้งความผิดพลาด

มันเป็นไปได้ที่จะวางตัวแปรในออบเจ็กต์ request โดยตรง เหมือนตัวอย่างข้างต้น ถึงแม้ว่ามันต้องทำโปรแกรมพิเศษบางอย่าง แต่ถ้าใช้ bean มันจะทำให้ JSPs ใช้ขอบเขตการร้องขอได้ง่ายขึ้น สิ่งที่เป็นในการวาง bean ลงในขอบเขตการร้องขอคือ การเปลี่ยนของ Scope="page" ในแท็ก useBean ไปเป็น Scope="request" ตัวอย่างต่อไปนี้แสดงอีกเวอร์ชันหนึ่งที่ใช้ bean

```
<jsp:useBean id="which"
    class="com.awl.jspbook.ch07.WhichPageBean"
    scope="request"/>
<jsp:setProperty name="which" proper="*" />
<% if (which.getWhich( ).equals("red")) { %>
    <jsp:forward page="/ch06/red.jsp"/>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<% } else if (which.getWhich( ).equals ("green")) { %>
    <jsp:forward page="/ch06/green.jsp"/>
<% } else if (which.getWhich( ).equals ("blue")) { %>
    <jsp:forward page="/ch06/blue.jsp"/>
<% } %>
<HTML>
<BODY>
You asked for a page which dose not exist!
</BODY>
</HTML>

```

เวอร์ชันที่ใช้ bean สามารถรับตัวแปรฟอร์มได้อัตโนมัติ และ JSP ที่รับ forwarded request จะสามารถเข้าถึง bean เดียวกันเพื่อรับคุณสมบัติที่ปรับปรุงแล้วได้ ดังตัวอย่าง

```

<jsp:useBean id="which"
    class="com.awl.jspbook.ch07.whichPageBean"
    scope="request"/>
<HTML>
<BODY BGCOLOR="#00FF00">
<H1>Welcome to the green page</H1>
Your favorite number is
<jsp:getProperty name="which" property="favorite"/>
</BODY>
</HTML>

```

#### ● Includes Revisited

เมื่อคุณเปรียบเทียบความคล้ายกันของ JSP แนวประกอบในโรงงานอุตสาหกรรม ที่เราอ้างถึงมี 2 สิ่งที่สามารถเกิดขึ้นต่อการร้องขอหนึ่งๆ หลังจากที่แนวหนึ่งส่งการร้องขอไปสู่แนวอื่น สิ่งแรกคือ แนวที่ 2 สามารถส่งผล HTML ไปยังผู้ใช้ และเป็นการจบการร้องขอ ซึ่งเป็นสิ่งที่แท้ก forward ทำ อีกสิ่งหนึ่งคือ แนวที่ 2 สามารถส่งการร้องขอกลับไปยังแนวที่ 1 สิ่งนี้คือสิ่งที่แท้ก jsp:include ทำ

ใช้เวลามากในการอธิบายถึง include directive ที่บอก JSP Engine เพื่อรวบรวม text ที่เพิ่มเติมเหมือนมันสร้าง servlet เราจดจำไว้ว่า เนื่องจากสิ่งนี้เกิดขึ้นในช่วงเวลาการแปลผลของ servlet ไม่มีทางที่จะตัดสินใจได้ว่าไฟล์ที่รวบรวมเปลี่ยนแปลง

แท้ก include ไม่เหมือน include directive คือ รวมช่วงเวลาการร้องขอเข้าไปด้วย การรวมช่วงเวลาการร้องขอในหลายๆทางแล้วดีกว่าการรวมช่วงเวลาการแปล ข้อดีข้อแรกของ include tag คือ อนุญาตให้ผู้เขียน JSP ปฏิบัติกับทุกไฟล์ เหมือนเป็นเพลงที่อยู่เดี่ยวๆได้โดยปราศจากความกังวลเกี่ยวกับการที่เพลงหนึ่งถูกรวมกับเพลงอื่น การรวมของช่วงเวลาการแปลเท่ากับการคัดลอกไฟล์ที่จะถูกรวบรวมทั้งหมดไปยังไฟล์หลัก ที่เป็นสาเหตุให้ JSP Engine มองเห็นแท้ก useBean 2 แท้กเป็น bean เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีนี้จะไม่เกิดขึ้นเมื่อใช้แท็ก `include` เพราะจะทำการรวมเอาที่พูดจากเพจหนึ่งลงไปอีกเพจหนึ่งแทนที่การรวมไฟล์ต้นฉบับ

ข้อดีข้ออื่นของ `include` ของช่วงเวลาการร้องขอ คือ พวกมันแน่ใจว่าเมื่อมีการเปลี่ยนแปลงบนไฟล์ที่นำมารวม มันสามารถหาไฟล์ที่เปลี่ยนแปลงแล้วมาใช้ได้ทันที การใช้แท็ก `include` เมื่อไฟล์เปลี่ยนไป `servlet` จะถูกสร้างใหม่และทุกการร้องขอที่ถูกรวมจะเริ่มการใช้เวอร์ชันใหม่ในทันที ทำให้การดูแลรักษาไซต์ทำได้ง่ายมาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### JSP Sessions

#### 5.1 บทนำ

HTTP เป็นโพรโทคอลที่ไร้สถานะ ( stateless ) HTTP เป็นโพรโทคอลบนเครือข่ายที่ใช้ส่งทรัพยากร ( resource ) บนเว็บไปให้ไคลเอนต์ ทรัพยากรได้แก่ไฟล์ HTML, ไฟล์รูปภาพ, ผลแบบไดนามิก, เอาท์พุทจาก servlet หรือ JSP, หรือ CGI script ใดๆ เป็นต้น HTTP เป็นข้อตกลงในการติดต่อสื่อสารที่ยอมให้เซิร์ฟเวอร์และไคลเอนต์เข้าใจกัน ภายใต้การเชื่อมต่อ TCP/IP ก็เหมือนกับคน 2 คนคุยโทรศัพท์กัน คำสนทนาเหมือน HTTP และการต่อโทรศัพท์เหมือน TCP/IP HTTP ก็คล้ายกับโพรโทคอลอื่นๆที่ใช้รูปแบบไคลเอนต์-เซิร์ฟเวอร์ นั่นคือไคลเอนต์จะเชื่อมต่อกับเซิร์ฟเวอร์ผ่าน TCP/IP และส่งข้อความการร้องขอไปยังเซิร์ฟเวอร์ เซิร์ฟเวอร์จะส่งข้อความตอบสนองกลับ ที่มักจะบรรจุทรัพยากรที่ร้องขอมาไปด้วย หลังจากส่งผลตอบสนองแล้ว เซิร์ฟเวอร์สามารถปิดการเชื่อมต่อได้เหมือนกับการทำงานของเซิร์ฟเวอร์ได้เสร็จสิ้นไปแล้ว ตั้งแต่คำถามได้รับคำตอบการสนทนาจบลง มันทำให้การร้องขอถัดไปเป็นอิสระจากการร้องขอก่อนหน้านี้ HTTP จึงเป็นโพรโทคอลแบบไร้สถานะ

#### 5.2 การทำงานของ HTTP

เริ่มแรกไคลเอนต์จะเปิด TCP/IP socket จากนั้นมันจะติดต่อกับเซิร์ฟเวอร์ผ่าน HTTP โดยส่งบางอย่างออกไปบน socket ดังเช่นตัวอย่าง

```
GET /path/HelloWord.jsp HTTP/1.0
```

```
User - Agent : Mozilla/4.0
```

```
[some other HTTP headers here ]
```

```
[a blank line here]
```

เซิร์ฟเวอร์จะตอบสนองด้วยเฮดเดอร์ และทรัพยากรที่ร้องขอ ดังตัวอย่างต่อไปนี้ กลับไปบน socket เดิม

```
HTTP/1.0 200 OK
```

```
Date: Fri, 31 Dec 1999 23 : 59 : 59 GMT
```

```
Content - Type: text/html
```

```
Content - Length: 1354
```

```
<html>
```

```
<body>
```

```
<hl> Hello Word </hl>
```

```
</body>
```

```
</html>
```

หลังจากส่งผลกลับไปแล้ว เซิร์ฟเวอร์จะปิด socket

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.2.1 การเชื่อมต่อแบบฝังแน่น ( Persistent Connection )

ใน HTTP 1.0 ที่เป็นเวอร์ชันเก่า การเชื่อมต่อ TCP จะปิดลงหลังจากแต่ละการร้องขอและการตอบสนองเกิดขึ้น ดังนั้นแต่ละทรัพยากรที่จะถูกนำกลับมาใช้ใหม่ ต้องการการติดต่อของมันเอง การเปิดและปิดการเชื่อมต่อ TCP ต้องใช้เวลาของ CPU, แลบบความถี่, และหน่วยความจำ เพื่อประหยัดเวลาและทรัพยากร จึงยอมให้หลายๆการร้องขอและการตอบสนองถูกส่งผ่านการเชื่อมต่อแบบฝังแน่นอันเดียวกัน แนวคิดนี้นำมาปฏิบัติใน HTTP 1.1

อีกการส่งเสริมหนึ่งจากเบราว์เซอร์คือการทำการเก็บสำเนา ( copy ) ของทรัพยากรที่เคยถูกใช้ลงใน cache เบราว์เซอร์จะใช้สำเนาใน cache มากกว่าจะไปดาวน์โหลด ( download ) ไฟล์เดียวกันอีกครั้งจากเซิร์ฟเวอร์

แต่ HTTP ก็ยังเป็นโพรโตคอลแบบไร้สถานะ เพราะการเชื่อมต่อแบบฝังแน่นเป็นการช่วยการทำงานของ HTTP เท่านั้น เซิร์ฟเวอร์ยังคงสามารถปิดการเชื่อมต่อเวลาใดก็ได้

ในการสร้างแอปพลิเคชันบนเว็บให้มีประสิทธิภาพ ขึ้นอยู่กับความสามารถในการระบุลำดับของการร้องขอจากไคลเอนต์เดียวกัน

### 5.3 เทคนิคสำหรับการรักษาข้อมูลของผู้ใช้ข้ามเพจ

แนวคิดของการรักษาสถานะระหว่างการร้องขอที่ต่างกัน เรียกว่า session tracking มีเทคนิคที่ใช้ในเว็บแอปพลิเคชัน

#### 5.3.1 ข้อมูล HTTP ( HTTP Information )

สิ่งนี้จะรวมส่วนประกอบลงในเฮดเดอร์ เช่น เฮดเดอร์ของอีเมล ( email ) และข้อมูล IP ( IP information ) เทคนิคนี้ล้ำสมัย ไปอย่างรวดเร็ว ด้วยการแทนที่ของพร็อกซีเซิร์ฟเวอร์ ( proxy server )

#### 5.3.2 Hidden Fields

วิธีการนี้ยังคงเป็นที่นิยม ใช้แท็ก HIDDEN ของ HTML เพื่อฝังข้อมูลลงในแต่ละเพจ เช่น

```
< INPUT TYPE=HIDDEN NAME="MYKEY" VALUE="MYVALUE" >
```

เทคนิคนี้มักเกิดปัญหาด้านความปลอดภัย เพราะข้อมูลทุกข้อมูลถูกผู้เข้าชมเห็น ได้ด้วยวิธี View Source ในเบราว์เซอร์

#### 5.3.3 Cookies

Cookie เป็นชิ้นส่วนเล็กๆของข้อมูลที่ถูกนำมารวมกันอยู่บนไคลเอนต์โดยเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ส่งทรัพยากร HTTP กลับไปให้ไคลเอนต์ อาจจะส่ง cookie ที่บรรจุข้อมูลสถานะ ( state information ) ไปด้วย ซึ่งไคลเอนต์จะทำการเก็บข้อมูลนั้นเอาไว้ สิ่งที่มีรวมอยู่ในข้อมูลสถานะคือ คำอธิบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของขอบเขตของ URLs ที่สถานะถูกใช้แล้ว และในอนาคตเมื่อไคลเอนต์ทำการร้องขอ HTTP ที่อยู่ในขอบเขต URLs นั้น จะทำการส่งค่าในปัจจุบันของ cookie จากไคลเอนต์กลับไปยังเซิร์ฟเวอร์

Cookie ถูกแนะนำไปยังไคลเอนต์โดยการรวม Set-Cookies HTTP header เป็นส่วนหนึ่งของการตอบสนอง HTTP ดังรูปแบบต่อไปนี้

Set - Cookies : NAME=VALUE; expires=DATE; path=PATH; domain=DOMAIN\_NAME; secure  
ที่ NAME = VALUE เป็นการจับคู่ชื่อค่าที่บรรจุข้อมูล และ expires=DATE ระบุถึงวันที่ บอกล่วงเวลาที่ใช้ได้ของ cookie นั้น พิจารณาจากตัวอย่างข้างบน เมื่อไคลเอนต์ร้องขอทรัพยากรไปบนเซิร์ฟเวอร์ และรับการตอบสนองมาดังตัวอย่างนี้

Set-Cookies: CUSTOMER=JAMES\_M;

Path=/;

Expires=Wednesday, 09-Nov-99 23 : 12 : 40 GMT

สิ่งนี้จะถูกเก็บไว้ในเบราว์เซอร์ใน 1 ของ 2 ทาง ขึ้นอยู่กับอายุของมัน

- ถ้า cookies มีการระบุอายุที่ดี มันจะถูกเขียนในหน่วยความทรงจำแบบฝังแน่น
- ถ้า cookies ไม่ระบุอายุ มันจะเก็บไว้ในหน่วยความจำในขณะที่โปรแกรมเบราว์เซอร์กำลังรันเท่านั้น และเมื่อเบราว์เซอร์ถูกปิด cookies จะหายไปด้วย

ทั้ง 2 กรณี เมื่อไคลเอนต์ร้องขอทรัพยากรใดบนเซิร์ฟเวอร์ บางส่วนของการร้องขอจะรวม HTTP header ซึ่งบรรจุรายละเอียดของ cookies ซึ่งสามารถกระจายออกมาได้โดยเซิร์ฟเวอร์ เช่น

cookies : CUSTOMER=JAMES\_M

servlet API มีคลาส javax.servlet.http.Cookies ที่ยอมให้แอปพลิเคชันบนเซิร์ฟเวอร์ทำการตั้ง cookies จาก JSP ตามลำดับเพื่อรักษาสถานะ เช่น

```
Cookies uid = new Cookie ("uid","112kkjk222kjj989c");
```

มีอุปสรรคสำหรับ cookies อยู่ 2 ข้อ

- ผู้ใช้สามารถเลือกไม่ยอมรับมันได้ สิ่งนี้เกี่ยวข้องกับชุดที่เป็นความลับ ด้วยการเพิ่มขึ้นของการใช้ cookies เพื่อเก็บและส่งผ่านข้อมูลระหว่างกลุ่มและผู้ขาย ผู้คนจึงอ่อนไหวกับ cookies และปฏิเสธในการยอมรับมัน

- เกือบทุกเว็บเบราว์เซอร์ในปัจจุบัน ที่อาจจะไม่ทำงานกับ cookies

ตัวอย่างที่ดีของ cookies ในทางปฏิบัติคือ ใช้ cookies เพื่อเก็บกุญแจเข้ารหัสฐานข้อมูล ( encrypted database key ) บนไคลเอนต์ เวลาต่อมาเบราว์เซอร์ส่งการร้องขอไปยังเว็บไซต์เดิม มันจะรวม cookies ไปด้วย เซิร์ฟเวอร์สามารถใช้ค่าใน cookies เพื่อช่วยในการดึงข้อมูลที่เหมาะสมจากตารางฐานข้อมูล

#### 5.4 การทำงานกับ cookies และ Java

การพัฒนาแอปพลิเคชันที่ใช้ cookies นั้นง่ายใน Java เพราะใช้พื้นฐานออบเจกต์ ที่คลาส javax.servlet.http.Cookies เตรียมไว้ เมธอดในคลาสนี้สามารถถูกใช้โดย servlet หรือ JSPs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างข้างล่างแสดงถึงการใช้ cookies ใน JSP เพื่อพิมพ์เวลาใช้ครั้งสุดท้ายที่ผู้ใช้เคยมาที่เพจ

```
<%@ page language="java" import="java.text.DateFormat,java.util.Date" %>
```

```
<%
```

```
boolean found = false;
```

```
Cookie info = null;
```

```
String msg = "This is the first time you've visited this page.";
```

```
// Get all the cookie that came with the request
```

```
Cookie [ ] cookie = request.getCookies();
```

```
For(int i = 0; i < cookie.length; i++) {
```

```
    info = cookie [ i ];
```

```
    if (info.getName().equals("MyCookie")) {
```

```
        found = true;
```

```
        break;
```

```
    }
```

```
}
```

```
String newValue = "" + System.currentTimeMillis();
```

```
if (!found) {
```

```
    // Create a new Cookie and set its age.
```

```
    info = new Cookie("MyCookie", newValue);
```

```
    info.setMaxAge(60*1);
```

```
    info.setPath("/");
```

```
    response.addCookie(info);
```

```
} else {
```

```
    long conv = new Long(info.getValue()).longValue();
```

```
    msg = "You last visited this site on " + new Date(conv);
```

```
// Set the new value of the cookie, and add it to the response
```

```
info.setValue(newValue);
```

```
info.setMaxAge(10*24*60*60); // keep the cookie for 1 month
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Set the path so that cookie is available everywhere on the server
info.setPath ("/");
response.addCookie (info);
}
%>

<html>
<body>
<h2><%= msg %></h2>
<h2> Current system data is <%= new Data().toString() %> </h2>
</body>
</html>
```

ในโค้ด JSP เราใช้เมธอด `request.getCookies()` เพื่อดึงเอา cookies ที่ถูกส่งมาเป็นส่วนหนึ่งของการร้องขอ แต่ไม่มีเมธอดสำหรับหาชื่อ cookies ดังนั้นเราจำเป็นต้องหา MyCookies ซ้ำไปเรื่อยๆ ในอาร์เรย์ (array) ถ้าไม่มี cookies (เป็นครั้งแรกที่เราเข้ามาเยี่ยมชมเพจนี้ หรือมันมากกว่า 1 เดือน ที่เราเคยมาเป็นครั้งสุดท้าย) เราจะสร้าง cookies ขึ้นมาใหม่ แต่ถ้ามี cookies อยู่แล้ว เราจะอัปเดต cookies และแสดงเวลาครั้งสุดท้ายที่เราเข้ามา กับเวลาในขณะนั้น

ในทางปฏิบัติแอปพลิเคชันโค้ดของ cookies จะถูกรวมอยู่ใน bean ดังนั้น JSPs อื่นๆ ก็สามารถใช้โค้ดนี้ได้

สรุปแล้ว แอปพลิเคชันธรรมดา มักจะใช้ hidden field กับ cookies ร่วมกัน ซึ่งมีปัญหาหลัก 2 ข้อคือ

- ความปลอดภัยของข้อมูลใดๆ ที่คุณเก็บไว้ใน plain text
- โคลเอนต์ไม่ยอมรับ cookies โดยเหตุผลส่วนตัว

## 5.5 Sessions

ความจริงที่สังเกตได้ ทุกเครื่องกลไกของ session – tracking จะวางอยู่บน plain text ซึ่งมันไม่เหมาะกับรูปแบบที่เป็นออบเจกต์แบบซับซ้อน หรือแอปพลิเคชันของธุรกิจ และที่เห็นได้ชัดอีกอย่างก็ว่ามันเป็นไปได้ที่จะเก็บออบเจกต์ `java.util.Vector` ที่บรรจุกลุ่มของ Item ใน cookies ทำให้มีความต้องการเครื่องกลไกที่มีพื้นฐานออบเจกต์, ความปลอดภัย และอื่นๆ ซึ่งจะช่วยให้อินเทอร์เน็ตเฟซกับโค้ดเว็บแอปพลิเคชันที่ช่วยสร้างเพจแบบไดนามิก

### 5.5.1 อะไรคือ Session

session สามารถนิยามถึงลำดับความสัมพันธ์ระหว่างไคลเอนต์หนึ่งกับเซิร์ฟเวอร์ ซึ่งได้รับที่ตั้งเกินช่วงเวลา ในความเกี่ยวพันของออบเจกต์ session สามารถคิดว่าเป็นออบเจกต์หนึ่งได้ซึ่งจะอยู่ประจำและเปิดสู่แอปพลิเคชันเซิร์ฟเวอร์ ( servlet และ JSPs ) ที่สามารถถูกใช้สำหรับเก็บและกู้คืนออบเจกต์

ออบเจกต์ session ถูกสร้างบนเซิร์ฟเวอร์ ตัวระบุชื่อที่ไม่ซ้ำ เรียกว่า session ID ซึ่งมีความเกี่ยวเนื่องกับมัน และ ID นี้เป็นข้อมูลเดียวที่จะให้ไคลเอนต์

ทุกครั้งที่ไคลเอนต์เข้าถึงทรัพยากรบนเซิร์ฟเวอร์ ไคลเอนต์จะเตรียม session ID ที่มันถูกกำหนดการใช้ session ID นั้น servlet สามารถบรรจุออบเจกต์ session ที่เกี่ยวข้องได้ และทำการกู้คืนข้อมูลใดๆที่ถูกเก็บไว้ได้ มี 2 สิ่งที่จะต้องจดจำไว้คือ

- session ID ที่ถูกสร้างโดยเซิร์ฟเวอร์นั้น คูทางกายภาพแล้ว แตกต่างกันไปตามเซิร์ฟเวอร์
- มันสำคัญที่ต้องรู้ว่า session มีความสัมพันธ์แบบ 1 ต่อ 1 ระหว่างไคลเอนต์กับเซิร์ฟเวอร์ และถ้าผู้ใช้เข้าไปในไซต์ที่มีการใช้ servlets หรือ JSPs กับ JavaBean มาก จะทำให้เป็น session ID อันหนึ่ง และ session หนึ่งนั้นจะถูกใช้ร่วมกันท่ามกลางทุกทรัพยากรเหล่านี้

### 5.5.2 Servlet สนับสนุน session และ วงจรชีวิต session

Servlet API เตรียมอินเตอร์เฟซ javax.servlet.http.HttpSession และมันออกไปสู่ Servlet container เพื่อเตรียมการติดตั้งที่แสดงถึงมุมมองของ session ของเซิร์ฟเวอร์ JSPs และ Servlets เก็บ session ID บนไคลเอนต์ โดยการส่ง cookies หรือ โดยการเขียน URL ใหม่ ( URL-rewriting ) การเขียน URL ใหม่ จะพาดพิงถึงการรวม session ID เข้าไปต่อท้ายในแต่ละลิงค์บนเพจ

### 5.6 วงจรชีวิต Session

อินเตอร์เฟซ HTTPServlet Request บรรจุเมธอดที่มีประโยชน์มาก เมื่อมีการใช้ session ดังตารางที่ 5-1

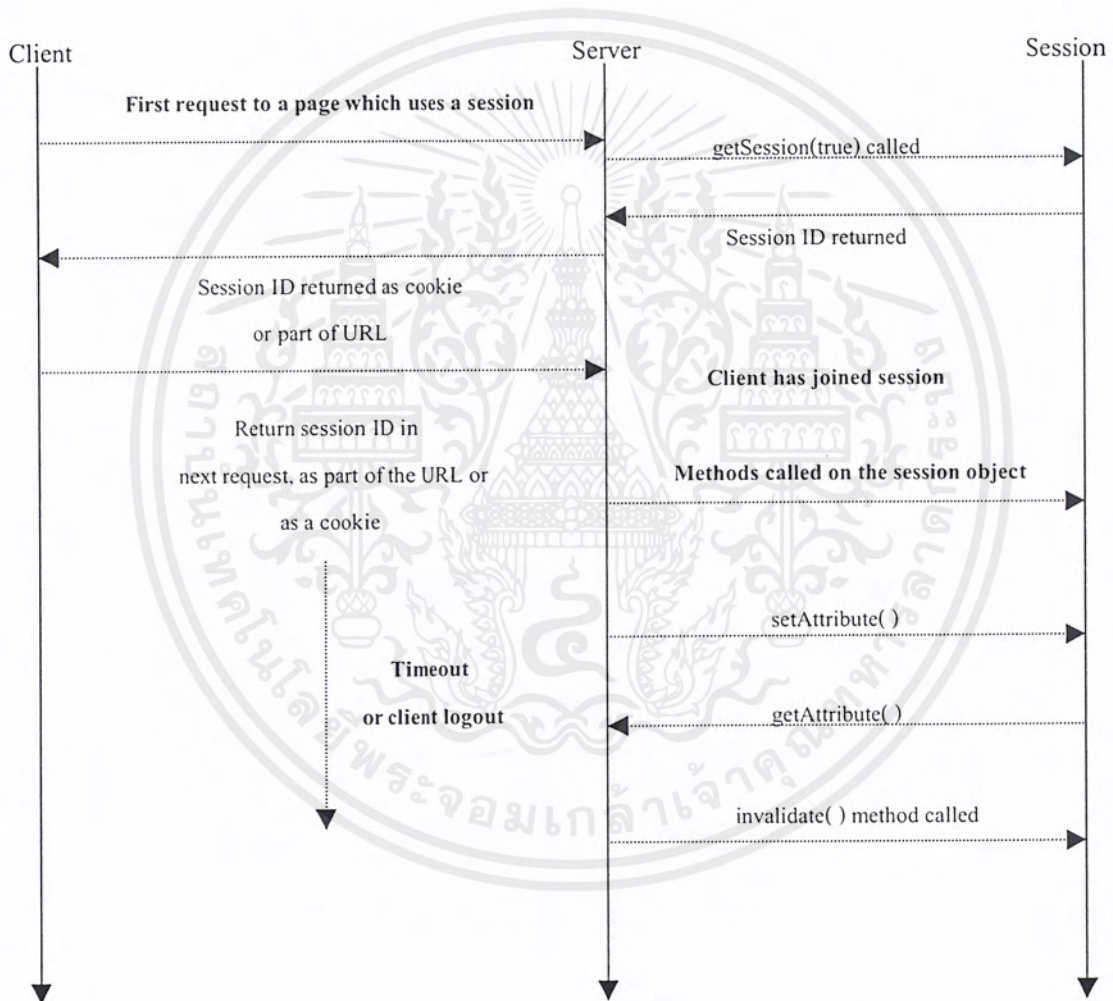
เมธอด	คำอธิบาย
HttpSession getSession ( )	ส่งกลับ session ในขณะนั้นที่สัมพันธ์กับการร้องขอนี้ หรือถ้าการขอไม่มี session ให้ทำการสร้าง session ใหม่
HttpSession getSession (boolean create)	ส่งกลับ Http Session ที่สัมพันธ์กับการร้องขอนี้ หรือถ้า create เป็นจริงจะทำการสร้าง session ใหม่สำหรับการร้องขอ
Boolean isRequestedSessionIdFromCookie ( )	ตรวจเช็ค session ID ว่าเป็นอันที่การร้องขอเสนอให้เป็น cookie หรือไม่ ถ้าใช่ มันจะส่งค่ากลับเป็น true

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Boolean isRequestedSessionIdFromUrl ( )	ตรวจสอบ session ID ว่าเป็นอันที่การร้องขอเสนอให้เป็นส่วนของ URL ของการร้องขอหรือไม่ ถ้าใช่ มันจะส่งค่ากลับเป็น true
Boolean isRequestedSessionIdValid ( )	ตรวจสอบว่าการร้องขอนี้มี session ที่สัมพันธ์กับมันหรือไม่ ถ้าใช่ มันจะส่งค่ากลับเป็น true

ตารางที่ 5-1 เมธอดของอินเทอร์เฟซ *HttpServletRequest*

รูปต่อไปนี้แสดงถึงวงจรชีวิตของ session



รูปที่ 5-1 วงจรชีวิตของ session

วงจรชีวิต session นั้นง่ายที่เดียว session ถูกสร้างบนเซิร์ฟเวอร์เป็นผลจากการร้องขอจากไคลเอนต์ และกำหนด session ID ให้เป็นพิเศษ และ ID นี้ จะถูกส่งผ่านไปให้ไคลเอนต์ ( เป็น cookie หรือตัวแปรที่ซ่อนอยู่ ) โดยตัวของ session เองจะพิจารณาว่าเป็น new จนกระทั่งไคลเอนต์มีการส่ง session ID กลับไปให้เซิร์ฟเวอร์ ( ผ่าน cookie หรือเป็นส่วนของ URL ที่ร้องขอ ) ที่ชี้ให้เห็นว่ามี session ที่เคยถูกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้าง ถ้าโคลนเอ็นดีไม่สนับสนุน cookie และไม่สามารถทำการเขียน URL ใหม่ได้บนเซิร์ฟเวอร์ ก็จะไม่มีการ session ID ส่งกลับไปให้เซิร์ฟเวอร์ และ session จะยังคงสถานะ new ต่อไป session จะคงอยู่บนเซิร์ฟเวอร์จนกระทั่งมันใช้ไม่ได้ หรือเซิร์ฟเวอร์หยุดการทำงาน

การฝังแน่น session ไม่ได้ใช้สำหรับเก็บข้อมูลในระยะยาวระหว่าง session นั่นคือ คุณไม่ควรจะเชื่อถือ session ที่ถูกเก็บไว้ในโคลนเอ็นดีเป็นช่วงระยะเวลาที่นานกว่าที่จะส่งกลับไปยังไชต์ ตัวอย่างเช่น คุณไม่ควรพยายามที่จะใช้ session เก่าของผู้ใช้ในลักษณะการล็อกอิน ( login ) อัปเดตโนมิตี แต่น่าจะใช้ cookie ซึ่งไม่หมดอายุสำหรับการขยายช่วงเวลาออกไป เพื่อดึงข้อมูลผู้ใช้จากฐานข้อมูล จากนั้นคุณสามารถสร้าง session ของผู้ใช้ขึ้นมาใหม่ได้

ตารางข้างล่างแสดงถึงตัวอย่างวิธีการเก็บข้อมูล และ อ่านข้อมูลจาก session

Storing Data in Servlet 1	Retrieving Data in Servlet 2
<pre>HttpSession session =     request.getSession( ); MyObject m = new MyObject( ); session.setAttribute("james", m);</pre>	<pre>HttpSession session =     request.getSession( ); MyObject n =     (MyObject) session.getAttribute("james");</pre>

ตารางที่ 5-2 ตัวอย่างวิธีการเก็บข้อมูลและอ่านข้อมูลจาก session

### 5.6.1 Contexts

JSP ถูกออกแบบมาให้สามารถเคลื่อนย้ายระหว่างเซิร์ฟเวอร์ที่ต่างกันได้ มันต้องการข้อมูลที่สรุปที่จะถูกถือโดยเว็บเซิร์ฟเวอร์ จาก JSPs กับออบเจกต์ Java โดยพาดพิงถึงการสร้างผลตอบสนอง รายละเอียด JSP อาศัยอยู่บน Context เพื่อเตรียมบทสรุป และข้ามให้พ้นช่องว่างระหว่างการติดตั้งรายละเอียดเซิร์ฟเวอร์ กับโค้ด JSP ที่สามารถเคลื่อนย้ายได้

Context เตรียมสถานะที่มองไม่เห็นสำหรับทรัพยากร และเตรียมอินเตอร์เฟซสำหรับทรัพยากรด้วย เพื่อการติดต่อสื่อสารกับสิ่งแวดล้อมของพวกมัน ตัวอย่างเช่น การปฏิบัติการของ Servlet ใน context (javax.servlet.ServletContext) ทุกอย่างที่ servlet ต้องการจะรู้เกี่ยวกับเซิร์ฟเวอร์ของมัน สามารถถูกกระจายมาจาก context และทุกอย่างที่เซิร์ฟเวอร์ต้องการเพื่อการติดต่อสื่อสารกับ servlet จะถูกส่งผ่าน context

กฎตายตัวคือ ทุกสิ่งใน JSP ต้องมีความสัมพันธ์กับ context และทุก context จะต้องมีขอบเขต (scope)

มีอินเตอร์เฟซ context ตัวอื่น สำหรับ JSPs คือ javax.servlet.jsp.PageContext ซึ่งออบเจกต์สามารถถูกกั้นด้วยขอบเขตต่างๆ

พิจารณาแท็กต่อไปนี้

```
<jsp:useBean id="catalog"
```

```
scope="page|request|session|application"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class="projsp.Catalog" />
```

เมื่อแท็กนี้ถูกคอมไพล์ด้วยค่าของขอบเขตที่ต่างกัน มันจะเกิดเหตุการณ์ที่ต่างกัน

#### 5.6.1.1 ขอบเขตเพจ ( Page Scope )

ขอบเขตเพจถูกแปลเป็น `PageContext.PAGE_SCOPE` และออบเจกต์เกี่ยวข้องกับ `javax.servlet.jsp.PageContext`

การอ้างอิงออบเจกต์ถูกละทิ้งไปที่ `Servlet.service()` ทำสำเร็จ ขณะสร้าง servlet นั้น servlet engine จะสร้างออบเจกต์ในเมธอด `service()` ซึ่งตามการพาขอบเขตออบเจกต์โดยปกติในจาวา ออบเจกต์จะคงอยู่สำหรับทุก ๆ โคลเอนต์ในการร้องขอทรัพยากร

#### 5.6.1.2 ขอบเขตการร้องขอ ( Request Scope )

ขอบเขตการร้องขอถูกแปลเป็น `PageContext.REQUEST_SCOPE` และออบเจกต์เกี่ยวข้องกับ `javax.servlet.jsp.PageContext`

การอ้างอิงออบเจกต์จะคงอยู่ได้นานเท่ากับที่ออบเจกต์ `HttpRequest` คงอยู่ ถึงแม้ว่าการร้องขอส่งผ่านหรือผูกมัดกับเพจต่าง ๆ การสร้าง servlet ผูกมัดกับออบเจกต์ `HttpServletRequest` และใช้เมธอด `setAttribute(String key, Object value)` ออบเจกต์จะแตกต่างกันสำหรับทุกการร้องขอของโคลเอนต์

#### 5.6.1.3 ขอบเขต Session ( Session Scope )

ขอบเขต session ถูกแปลเป็น `PageContext.SESSION_SCOPE` และออบเจกต์เกี่ยวข้องกับ `javax.servlet.jsp.PageContext`

ในการติดตั้งใช้งานก่อนหน้าของ servlet API นั้น session จะรันภายใต้ session context ต่างกัน อยู่เสมอ แต่ในทุกวันนี้ มีเพียง session ID เท่านั้นที่คุณสามารถรับได้ มันเป็นส่วนหนึ่งของการร้องขอจากโคลเอนต์ผ่านไปสู่ `PageContext` และทำให้มันใช้ได้กับ JSP

การสร้าง servlet ผูกมัดกับออบเจกต์ `HttpSession` และใช้เมธอด `setAttribute(String key, Object value)` ออบเจกต์จะแตกต่างกันสำหรับทุกโคลเอนต์ และจะคงอยู่ใช้ได้ยาวนานเท่ากับ session ของโคลเอนต์ที่ใช้ได้

#### 5.6.1.4 ขอบเขตแอปพลิเคชัน ( Application Scope )

ขอบเขตแอปพลิเคชันถูกแปลเป็น `PageContext.APPLICATION_SCOPE` และออบเจกต์เกี่ยวข้องกับ `javax.servlet.jsp.ServletContext`

สิ่งนี้เป็นขอบเขตที่ฝังแน่นมากที่สุด การสร้าง servlet ผูกมัดกับออบเจกต์ `ServletContext` และใช้เมธอด `setAttribute(String key, Object value)` สิ่งนี้ไม่มีการระบุเป็นพิเศษต่อแต่ละโคลเอนต์ และทุกโคลเอนต์เข้าถึงออบเจกต์เดียวกันทั้งหมด

โดยการกำหนดขอบเขตที่ใช้ได้เป็น "page|request|session|application" รูปแบบ JSP เป็นสิ่งที่ย่างสำหรับนักพัฒนาที่สามารถรวมลอจิกและการติดตั้งใช้งานของ bean หรือออบเจกต์ได้ ต้องแน่ใจได้ว่าด้วยนิยามขอบเขตที่ถูกต้อง ออบเจกต์จะสามารถถูกเข้าถึงได้เมื่อต้องการ

สิ่งหนึ่งที่เป็นภาระหนักในความคิดคือ การใช้ bean หมายความว่า จะเกิดการสร้างขึ้นใหม่ถ้ามันไม่คงมีอยู่ ไม่ว่าจะใช้ขอบเขตใดก็ตาม การใช้ขอบเขตให้ถูกต้องมีความสำคัญ ถ้าใช้อย่างไม่ระมัดระวัง มันสามารถเป็นเหตุให้เกิดผลที่ไม่คาดหวังไว้ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### การวิเคราะห์และการออกแบบ

#### 6.1 ความต้องการของระบบ

##### 6.1.1 หลักการ

เนื่องจากเว็บไซต์ใช้งานได้กับอุปกรณ์ไร้สาย ( Wireless Device ) ซึ่งมีความสามารถจำกัดดังนั้น บริการจึงอยู่ที่เซิร์ฟเวอร์เป็นหลัก เว็บไซต์นั้นจริงๆ แล้วก็ได้พัฒนาขึ้นมาจากมาตรฐานอินเทอร์เน็ตนี้เอง ซึ่งก็ใช้พื้นฐานเดียวกับอินเทอร์เน็ต เพียงแต่มีการปรับปรุงบางส่วนให้เหมาะสม ดังนั้นการจัดกลุ่มสมาชิกผ่านเว็บ ก็เหมือนการให้บริการผ่านเว็บทั่วไป คือ มีเว็บเซิร์ฟเวอร์คอยให้บริการ

##### 6.1.2 ขอบเขตการให้บริการ

สำหรับการนัดหมายดำเนินงานผ่านเว็บ จะเป็นบริการแก่คนทั่วไป โดยแต่ละบุคคลจักทำการสมัครสมาชิกกับระบบก่อนเพื่อขอใช้บริการและทำการจัดกลุ่มกับสมาชิกด้วยกันเท่านั้น

##### 6.1.3 กิจกรรมต่างๆ

- การลงทะเบียนสมาชิกใหม่
- การเข้าสู่ระบบ
- การรับส่งข้อความ
- การจัดตารางเวลา
- การจัดกลุ่มสมาชิก
- การนัดหมายกลุ่มสมาชิก
- การแจ้งเตือน
- การเปลี่ยนแปลงข้อมูลของตัวเอง
- การออกจากระบบ

##### 6.1.4 การลงทะเบียนสมาชิกใหม่

ผู้ที่มีความประสงค์ในการใช้ระบบ ต้องทำการลงทะเบียนเป็นสมาชิกกับระบบเสียก่อน ซึ่งในการสมัครสมาชิกนั้นผู้สมัครจะต้องกรอกข้อมูลเพื่อใช้ในการสืบค้นต่อไป

##### 6.1.5 การเข้าสู่ระบบ

ผู้ที่ทำการลงทะเบียนเป็นสมาชิกแล้ว เมื่อต้องการเข้าสู่ระบบก็ใส่ username และ password ที่สามารถเข้าสู่ระบบได้ และเซิร์ฟเวอร์ต้องเริ่มรักษา Session ให้แก่ผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.1.6 การรับส่งข้อความ

ผู้ที่มีความประสงค์ในการใช้บริการรับข้อความสามารถข้อความที่ส่งมาจากผู้ใช้ระบบด้วยกัน การใช้บริการการส่งข้อความ ผู้ส่งต้องใส่ชื่อผู้ที่รับข้อความ หัวเรื่องและข้อความ เมื่อรับข้อความแล้ว ต้องการลบก็สามารถทำได้โดยทำการเลือกข้อความที่ต้องการลบ

### 6.1.7 การจัดตารางเวลา

การใช้บริการจัดตารางเวลานั้นผู้ใช้ต้องใส่ วัน เดือน ปี เวลา หัวเรื่องและรายละเอียด เมื่อทำการบันทึกแล้วระบบจะทำการตรวจสอบว่าเวลาซ้อนทับกับที่มีอยู่ในตารางเวลาหรือไม่ ถ้าซ้อนทับก็จะไม่ยอมให้บันทึกลงในฐานข้อมูล เมื่อผู้ใช้ต้องการกลับมาดูว่าได้บันทึกตารางเวลา ก็จะทำได้โดยใส่ วัน เดือน ปี แล้วก็แสดงตารางเวลาของวันนั้นออกมาทั้งหมด ในกรณีที่ไม่ต้องการทำกิจกรรมก็สามารถทำการยกเลิกตารางเวลาที่ต้องการยกเลิกได้

### 6.1.8 การจัดกลุ่มสมาชิก

การใช้บริการจัดกลุ่มสมาชิกนี้แบ่งได้เป็นสองส่วนคือ ผู้สร้างกลุ่มการทำงาน และ ผู้เป็นสมาชิก

- ผู้สร้างกลุ่มการทำงาน

ผู้สร้างกลุ่มจะสร้างกลุ่มโดยการใส่ชื่อกลุ่มเข้าไป โดยผู้สร้างกลุ่มการทำงานหนึ่งคนสามารถสร้างกลุ่มได้ไม่จำกัด แต่ชื่อกลุ่มต้องไม่ซ้ำกัน การลบกลุ่มการทำงาน ก็สามารถทำได้โดยเลือกกลุ่มการทำงานที่ต้องการลบ หรือจะลบผู้เป็นสมาชิกเป็นรายบุคคลก็ได้ จะเห็นได้ว่าการลบกลุ่มการทำงาน และการลบสมาชิกในกลุ่ม เป็นหน้าที่ของผู้สร้างกลุ่มแต่เพียงผู้เดียว เพื่อที่จะป้องกันผู้ไม่ประสงค์ดีมารบกวนกลุ่ม ดังเช่นในห้อง chat ที่จะมีผู้คอยควบคุมห้อง

- ผู้เป็นสมาชิกกลุ่ม

ผู้ที่จะเป็นสมาชิกต้องใส่ชื่อผู้สร้างกลุ่มก่อน จากนั้นก็ทำการเลือกว่าจะเข้ากลุ่มไหน จึงจะเป็นสมาชิกของกลุ่มโดยสมบูรณ์ การที่เราให้เพียงผู้เป็นสมาชิกกลุ่มมาสมัครเพียงทางเดียวเท่านั้นเป็นเพราะป้องกันผู้ไม่ประสงค์ดี เช่น ถ้าให้ผู้ใดก็ได้มาเลือกเราเข้ากลุ่ม ก็หมายความว่าสามารถเข้าถึงข้อมูลไปได้ส่วนหนึ่ง ประโยชน์อีกประการหนึ่งที่ทำให้ผู้เป็นสมาชิกมาสมัครเองก็คือ ถ้าเกิดมีสมาชิกในกลุ่มเป็นจำนวนมาก ผู้สร้างกลุ่มก็ไม่ต้องมาคอยเพิ่มชื่อผู้เป็นสมาชิกที่มีเป็นจำนวนมากได้ แต่ปล่อยเป็นหน้าที่ของผู้ที่จะเป็นสมาชิกของกลุ่มมาสมัครเอง โดยความเป็นจริงแล้ว ผู้เป็นสมาชิกก็ไม่แน่ว่าจะสมัครอยู่กลุ่มการทำงานเป็นจำนวนมาก

### 6.1.9 การนัดหมายกลุ่มสมาชิก

การนัดหมายกลุ่มสมาชิกนี้ ผู้สร้างกลุ่มเท่านั้นที่จะทำการนัดหมายได้ เนื่องจากข้อที่แล้วที่ผู้เป็นสมาชิกกลุ่มมาสมัครด้วยความสมัครใจ ซึ่งต้องแปลว่า ผู้เป็นสมาชิกกลุ่มต้องยอมให้ผู้สร้างกลุ่มเข้าถึงตารางการนัดหมายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้สร้างกลุ่มเมื่อจะเริ่มการนัดหมาย ก็ใ้สัปดาห์ เดือน ปี เวลาที่จะเริ่มค้นหา เวลาที่สิ้นสุดการค้นหา จำนวนชั่วโมงการนัดประชุม หัวเรื่องและรายละเอียด เมื่อใส่ข้อมูลครบก็จะทำการหาเวลาที่เหมาะสม โดยการหา นั้นใช้วิธีการดังต่อไปนี้

เริ่มแรก เนื่องจากการเก็บตารางการนัดหมายขอทุกคนเก็บการนัดหมายไว้คือเวลาที่ไม่ว่าง กำหนด

ช่วงเวลา 24 ชั่วโมง:  $U = \{ P(0:00-24:00) \}$

ช่วงเวลาที่จะค้นหา:  $S = \{ P(8:00-18:00) \}$

จำนวนเวลาที่ต้องการ 1 ชั่วโมง

ตารางเวลาของ John:  $J = \{ P(8:30-10:00), P(12:00-13:00), P(15:00-16:00) \}$

ตารางเวลาของ Marry:  $M = \{ P(8:00-9:00), P(11:30-12:30), P(15:30-17:00) \}$

ตารางเวลาของ Buffe :  $B = \{ P(8:00-10:30), P(12:00-13:30), P(15:00-17:00) \}$

ตารางเวลาของ Adam :  $A = \{ P(9:00-10:00), P(12:00-13:00), P(16:00-17:00) \}$

จากนั้นก็นำเวลา 24 ชั่วโมงมา ลบกับตารางของแต่ละบุคคล  
จะได้

เซตของช่วงเวลาว่างของ John =  $U - J$

เซตของช่วงเวลาว่างของ John ในช่วงเวลาค้นหา

$$J' = S \cap (U - J)$$

$J' = \{ P(8:00-8:30), P(10:00-12:00), P(13:00-15:00), P(16:00-18:00) \}$

เซตของช่วงเวลาว่างของ Marry =  $U - M$

เซตของช่วงเวลาว่างของ Marry ในช่วงเวลาค้นหา

$$M' = S \cap (U - M)$$

$M' = \{ P(9:00-11:30), P(12:30-15:30), P(17:00-18:00) \}$

เซตของช่วงเวลาว่างของ Buffe =  $U - S$

เซตของช่วงเวลาว่างของ Buffe ในช่วงเวลาค้นหา

$$B' = S \cap (U - B)$$

$B' = \{ P(10:30-12:00), P(13:30-15:30), P(16:30-18:00) \}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซตของเวลาว่างของ Adam =  $U - A$

เซตของเวลาว่างของ Adam ในช่วงเวลาค้นหา

$$A' = S \cap (U - A)$$

$$A' = \{ P(8:00-9:00), P(10:00-12:00), P(13:00-16:00), P(17:00-18:00) \}$$

เมื่อได้เวลาว่างของทุกคนแล้วนำเวลาของทุกคนมา intersect กันก็จะได้เซตของเวลาว่างของทุกคน ที่เหมือนกัน

$$Time = J' \cap M' \cap B' \cap A'$$

$$Time = \{ P(10:30-11:30), P(13:30-15:00), P(17:00-18:00) \}$$

เมื่อได้เซตของเวลาว่างที่ตรงกันของทุกคนมาแล้ว ก็จะนำมาตรวจสอบระยะเวลาของแต่ละช่วง ว่ามากกว่าจำนวนชั่วโมงที่ต้องการนัดหรือไม่ ถ้ามากกว่าก็ให้ผู้สร้างกลุ่มเลือกเวลาที่ต้องการแล้วส่งข้อความไปบอกสมาชิกในกลุ่มว่ามีการนัดหมายแล้ว เมื่อสมาชิกได้รับข้อความก็จะบันทึกหรือไม่บันทึกตารางการนัดหมายของตัวเองก็ได้ ถึงแม้ว่าตารางการนัดหมายของตัวเองจะไม่มีการนัดหมายก็ตาม

แต่ถ้าช่วงเวลาใดมากกว่าเมื่อนำเวลาว่างของทุกคนมา intersect แล้วปรากฏว่าได้เซตว่างหรือไม่มีเวลาว่างไม่มากพอ ระบบเริ่มกระบวนการต่อไปคือการถามขอเลื่อนเวลาของผู้เป็นสมาชิก โดยมีด้วยกันอยู่ 2 แบบคือ การร้องขอทีละคน กับ การบังคับเลือก

การร้องขอทีละคน

เมื่อผู้สร้างกลุ่มเลือกวิธีการนี้ ในขั้นแรก ผู้สร้างกลุ่มจะส่งข้อความ ไปถามผู้เป็นสมาชิก ที่คนอื่นว่าง ยกเว้นสมาชิกคนที่ถูกร้องขอ ที่มีเวลาไม่ว่างจึงถูกร้องขอเลื่อนเวลาเช่น

$$\text{ช่วงเวลาที่ค้นหา: } S = \{ P(8:00-12:00) \}$$

จำนวนชั่วโมงที่ต้องการ 1 ชั่วโมง 30 นาที

$$Time = J' \cap M' \cap B' \cap A'$$

$$Time = \{ P(10:30-11:30) \}$$

ซึ่งไม่เพียงพอต่อความต้องการ

เมื่อจะร้องขอเวลาแก่ Marry เซตของช่วงเวลาที่ส่งให้ Marry คือ

$$M'' = J' \cap B' \cap A'$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$M'' = \{ P(10:30-12:00) \}$$

เมื่อนำมาเปรียบเทียบกับจำนวนชั่วโมงแล้วเพียงพอต่อความต้องการก็น่า ช่วงเวลา 10:30 ถึง 12:00 มาร้องขอแก่ Marry เมื่อ Marry ได้รับการร้องขอ Marry ก็อาจจะตอบรับการร้องขอการนัดหมายก็จะถูกบันทึกหรือ ไม่ตอบรับการร้องขอก็ได้ เมื่อไม่ตอบรับการร้องขอผู้สร้างกลุ่มจะส่งร้องขอผู้ต่อไป

#### การบังคับเลือก

เมื่อผู้สร้างกลุ่มเลือกวิธีการนี้ การเลือกเวลาจะไม่สนใจเลยว่าผู้เป็นสมาชิกของกลุ่มจะมีช่วงเวลาว่างหรือไม่ แต่จะถูกผู้สร้างกลุ่มกำหนดเวลาเพื่อเลือกช่วงเวลาเอง

#### 6.1.10 การแจ้งเตือน

ก่อนเวลาที่กิจกรรมในตารางเวลาจะเกิดขึ้น ก็จะมีการเตือนผู้ใช้ล่วงหน้าให้ทราบล่วงหน้า โดยบริการนี้จะเริ่มทำงานพร้อมขึ้นมาเซิร์ฟเวอร์ โดยสร้างเป็น Process หนึ่งค้างไว้เมื่อถึงเวลาที่กำหนดไว้ ก็จะเริ่มตรวจสอบกับฐานข้อมูลว่า กิจกรรมใดควรแจ้งเตือนได้แล้ว โดยการแจ้งเตือนจะผ่านบริการ SMS ( Shot Message Service )

#### 6.1.11 การเปลี่ยนแปลงข้อมูลของตัวเอง

ในขั้นตอนการสมัครเข้าสู่ระบบ จะมีการให้ใส่ข้อมูล เมื่อเวลาผ่านไปข้อมูลนั้นอาจเกิดการเปลี่ยนแปลง ดังนั้นบริการนี้ก็ให้เปลี่ยนแปลงข้อมูลที่ใส่มาตอนลงทะเบียน

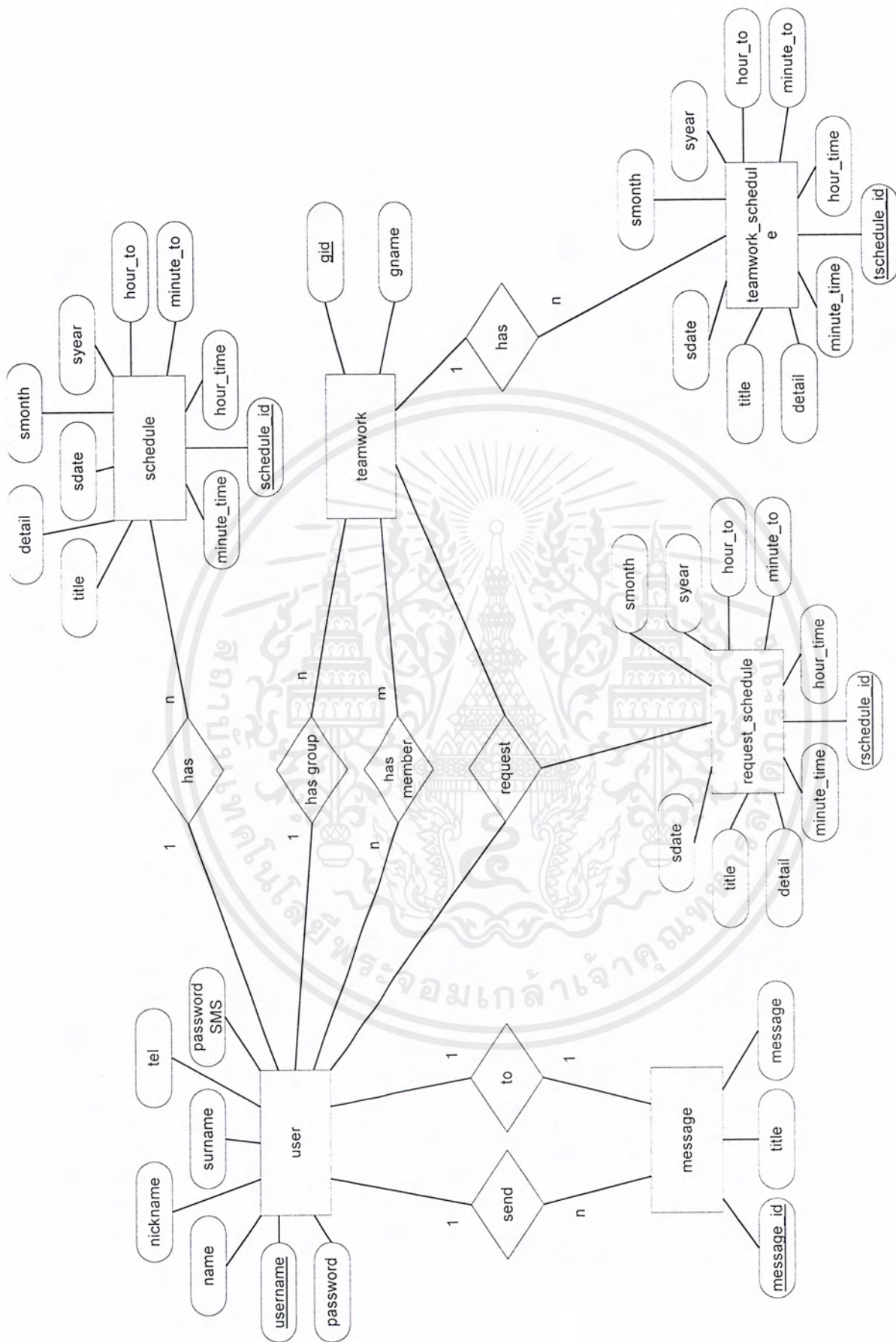
#### 6.1.12 การออกจากระบบ

เมื่อผู้ใช้ออกจากระบบเพื่อเลิกใช้บริการ เซิร์ฟเวอร์ต้องทำการเลิกการรักษา Session เพื่อความปลอดภัยของระบบ

### 6.2 การวิเคราะห์และออกแบบ

#### 6.2.1 ER diagram

มีการออกแบบฐานข้อมูลโดยใช้ ER diagram ซึ่งจะแสดงดังรูปที่ 6-1



รูปที่ 6-1 ER Diagram ของระบบการนัดหมายดำเนินงานผ่านเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักเรียนไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.2.2 Table

ฐานข้อมูลมีการสร้าง Table ดังนี้

User



username	name	surname	nickname	tel	password	password SMS
----------	------	---------	----------	-----	----------	--------------

Schedule



schedule_id	sdate	smonth	syear	hour_time
minute_time	hour_to	minute_to	title	detail
username				

Teamwork



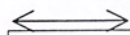
gid	gname	username
-----	-------	----------

Teamwork\_schedule



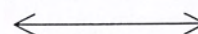
tschedule_id	sdate	smonth	syear	hour_time
minute_time	hour_to	minute_to	title	detail
username				

gmember



gid	username
-----	----------

Request\_schedule



rschedule_id	sdate	smonth	syear	hour_time	minute_time
hour_to	minute_to	title	detail	gid	username

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 การเตรียมพร้อมกับข้อมูลที่ไม่ถูกต้อง

ในการทำงานต่างๆ ของการบริการนัดหมายดำเนินงานผ่านเว็บ จะรับข้อมูลมาจากผู้ใช้ อินเทอร์เน็ตทั่วไป ซึ่งมีทั้งเด็ก ผู้ใหญ่ บุคคลต่างๆ ทั่วทุกมุมโลก ซึ่งแน่นอนในการป้อนข้อมูลเข้ามานั้น อาจจะมีข้อมูลที่ผิดพลาดบ้าง ทั้งที่ตั้งใจเพื่อจะทดลองหรือทดสอบการทำงานของโปรแกรม หรืออาจจะด้วยความไม่ตั้งใจ ซึ่งโปรแกรมที่ดีก็จะต้องมีการเตรียมรับมือกับข้อมูลที่ไม่คาดหวัง ทั้งนี้อย่างน้อยก็เพื่อให้การทำงานของโปรแกรมเป็นไปอย่างถูกต้อง และยังจะเป็นการสร้างเชื่อมั่นให้กับผู้ใช้บริการได้อีกด้วย

#### วิธีการป้องกันข้อมูลที่ไม่ถูกต้อง

##### 1. ป้องกันด้วยการใช้กำหนดข้อมูลให้เลือก

วิธีนี้สามารถรับประกันได้ว่า ข้อมูลที่ได้จะถูกต้องเสมอ เพราะเราเป็นคนกำหนดข้อมูลที่จะป้อนเข้ามา แล้วให้ผู้ใช้บริการเป็นคนเลือก เช่น การใช้ตัวเลือกวิทยุ ( Radio ), การตรวจสอบบ็อก ( Check Box ) หรือ ตัวเลือก ( Select ) ของเครื่องมือในฟอร์ม ( Form ) ของ HTML หรือกรณีกำหนดจำนวนตัวอักษรสูงสุดที่ให้ป้อน ก็อาจจะกำหนดจำนวนอักษรสูงสุดใน Text box ให้สามารถป้อนได้ไม่เกินที่กำหนด

##### 2. ใช้โปรแกรมเป็นตัวตรวจสอบ

วิธีนี้ถือว่าเป็นวิธีทั่วไปที่ใช้กัน เพราะเป็นการเขียนโปรแกรมขึ้นมาเพื่อทำการตรวจสอบข้อมูลที่ป้อนเข้ามาโดยเฉพาะ

##### 3. ใช้ฐานข้อมูลเป็นตัวตรวจสอบ

วิธีนี้ใช้กับการเก็บและแก้ไขข้อมูลในฐานข้อมูล โดยสามารถทำได้ 2 วิธีคือ การกำหนดชนิดของคอลัมน์ และ การใช้ Check ( เป็น Constrain ชนิดหนึ่ง ) โดยทั้งสองวิธีนี้สามารถรับรองได้ว่า ข้อมูลที่จะเก็บลงฐานข้อมูลได้นั้น จะต้องถูกต้องและเป็นไปตามข้อกำหนดเหล่านี้ เพราะถ้าหากข้อมูลไม่ถูกต้อง ฐานข้อมูลก็จะฟ้อง Error และหยุดการทำงานเลย เช่น กำหนดให้ Item\_ID เป็นคอลัมน์ที่เก็บข้อมูลชนิดตัวเลข เราก็จะไม่สามารถป้อนข้อมูลชนิดอื่นที่ไม่ใช่ตัวเลขลงในคอลัมน์นี้ได้

### 6.4 สถาปัตยกรรมของระบบ ( System Architecture )

ในการทำโครงงานนี้ส่วนประกอบที่สำคัญคือ

#### 6.4.1 เว็บเซิร์ฟเวอร์

##### 1. ฮาร์ดแวร์ ( Hardward )

Intel Pentium 166 MHz (MMX) H/D 3.2 Gbyte RAM 64 Mbyte Ethernet LAN card 10 Mbps

##### 2. ระบบปฏิบัติการ

ใช้ Windows NT Server version 4.0 SP 6a ของ Microsoft เป็นระบบปฏิบัติการ

##### 3. เว็บเซิร์ฟเวอร์โปรแกรม ( Web server program )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ Apache Tomcat version 3.2.1 for Windows เพื่อให้บริการนัดหมายดำเนินงานผ่านเว็บแก่ผู้ใช้ผ่าน โพรโทคอล HTTP

#### 6.4.2 Database Server

##### 1. ฮาร์ดแวร์

Intel Pentium 166 MHz (MMX) H/D 3.2 Gbyte RAM 64 Mbyte Ethernet LAN card 10 Mbps

##### 2. ระบบปฏิบัติการ

ใช้ Windows NT Server version 4.0 SP 6a ของ Microsoft เป็นระบบปฏิบัติการ

##### 3. Database Management System (DBMS )

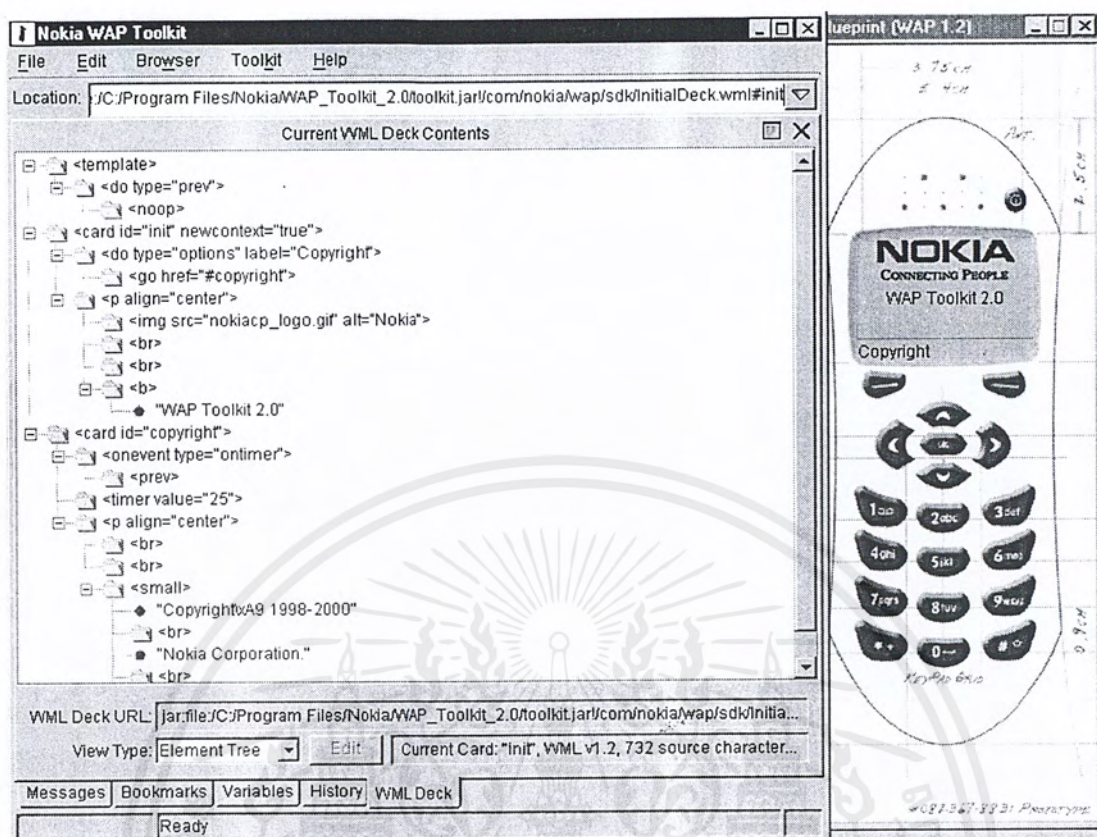
ใช้ MySQL version 3.23 for Windows เพื่อให้บริการเก็บฐานข้อมูลจากการให้บริการนัดหมายดำเนินงานแก่เว็บเซิร์ฟเวอร์

#### 6.4.3 Web programming program

ใช้ JSP 2.1

#### 6.4.4 WML Browser

ใช้ Nokia WAP Toolkit version 2.0 ในการตรวจสอบ ดังแสดงในรูป



รูปที่ 6-2 Nokia WAP Toolkit version 2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### การใช้งานโปรแกรม

#### 7.1 การทำงานฝั่ง Server

##### 7.1.1 การทำงานของ MySQL

- MySQL ใน Windows 95/98

การเริ่มต้นการทำงาน mysql server โดยการเปิด MS-DOS windows แล้วตามด้วยคำสั่ง

```
C:\<path to MySQL>\bin\mysqld
```

เมื่อต้องการจบการทำงานของ mysql server ใช้คำสั่ง

```
C:\<path to MySQL>\bin\mysqladmin -u root shutdown
```

- MySQL ใน Windows NT

ด้วยคำสั่ง

```
C:\<path to MySQL>\bin\mysqld-nt -install
```

จะทำการติดตั้งเป็นบริการบน Windows NT โดยคำสั่งเริ่มการทำงานและหยุดการทำงานคือ

```
NET START mysql
```

```
NET STOP mysql
```

##### 7.1.2 การทำงานของ Apache Tomcat Web Server

เริ่มต้นการทำงานด้วยคำสั่ง

```
C:\<path to Tomcat>\bin\startup.bat
```

จบการทำงานด้วยคำสั่ง

```
C:\<path to Tomcat>\bin\shutdown.bat
```

##### 7.1.3 การทำงานของโปรแกรมฉากหลัง

โปรแกรม ServerAlert ด้วยคำสั่ง

```
JAVA ServerAlert
```

โปรแกรม ServerSchedule ด้วยคำสั่ง

```
JAVA ServerSchedule
```

หมายเหตุ

ตัวโปรแกรมอยู่ในไดเรกทอรี Server

## 7.2 การทำงานฝั่ง Client

### 7.2.1 การลงทะเบียนสมาชิกใหม่

เป็นการสมัครขอใช้โปรแกรมที่มีอยู่ในระบบการจัดการสมาชิกผ่านเว็บ โดยผู้ใช้สามารถลงทะเบียนได้จากหน้า Login Page ( โดยกำหนด URL ไปที่ <http://<server>/jsp/login.jsp> ) ดังรูปที่ 7-1

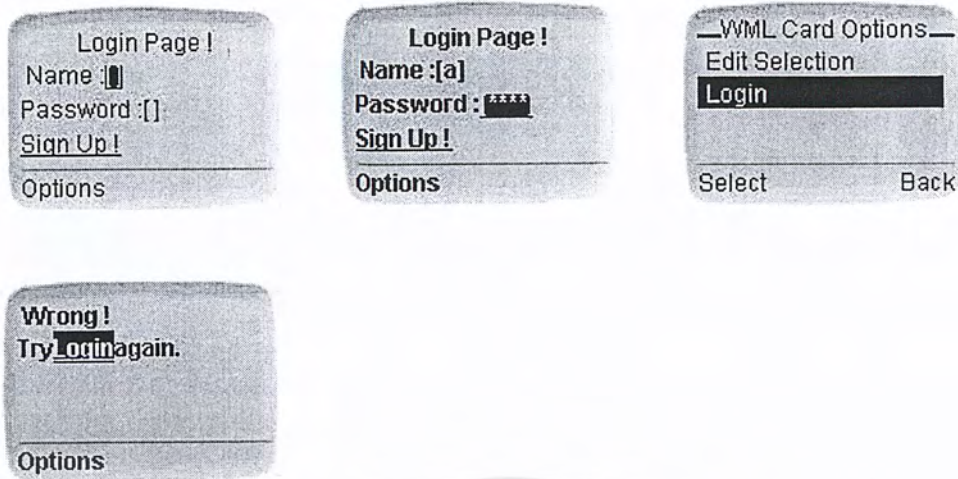
รูปที่ 7-1 แสดงหน้า Login Page

ผู้ใช้ที่ต้องการลงทะเบียน จะต้องเลือก Sign Up ในหน้า Login Page หลังจากนั้นจะปรากฏหน้าจอสำหรับใส่ชื่อและรหัสผ่าน ( Password ) พิมพ์รหัสผ่านซ้ำอีกครั้งให้ถูกต้อง เมื่อใส่ข้อมูลครบหมดแล้ว ให้กด Options เลือก Submit ก็ถือว่าการลงทะเบียนเสร็จสิ้นเรียบร้อยแล้ว ผู้ใช้สามารถเข้าระบบเพื่อใช้โปรแกรมได้

รูปที่ 7-2 แสดงขั้นตอนการลงทะเบียนเพื่อเข้าสู่ระบบ

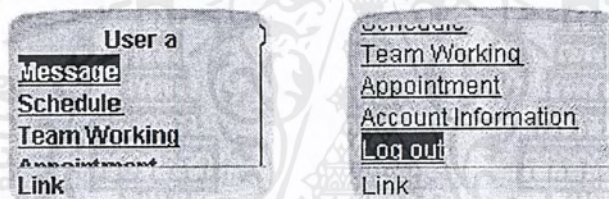
### 7.2.2 การเข้าสู่ระบบ

ผู้ใช้จะใช้โปรแกรมในการจัดการกลุ่มสมาชิกได้ ผู้ใช้จะต้องเข้าสู่ระบบก่อน หลังจากผู้ใช้ลงทะเบียนขอใช้บริการ ผู้ใช้จะมี User Name กับ Password ผู้ใช้เข้าสู่ระบบโดยการใส่ User Name กับ Password ลงในหน้า Login Page จากนั้นกด Options เลือก Login ถ้าใส่ User Name กับ Password ถูกต้อง จะสามารถเข้าไปยัง Main Page ของการใช้โปรแกรมได้ แต่ถ้าใส่ไม่ถูกต้อง จะต้องทำการ Login ใหม่ ดังแสดงในรูปที่ 7-3



รูปที่ 7-3 แสดงขั้นตอนการเข้าสู่ระบบ

หลังจากเข้าสู่ระบบแล้ว จะเข้าสู่หน้าหลักของการใช้โปรแกรม ดังรูปที่ 7-4



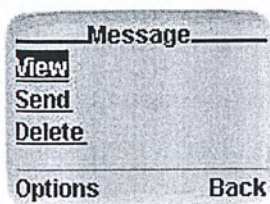
รูปที่ 7-4 แสดงหน้าหลักของการใช้โปรแกรม

หน้าหลักของการใช้โปรแกรมจะแสดงถึงรายการบริการที่สามารถใช้ได้ ซึ่งมีรายการดังนี้

- Message บริการรับส่งข้อความระหว่างสมาชิก
- Schedule บริการจัดการตารางเวลาของผู้ใช้บริการ
- Team Working บริการการจัดการกลุ่มการทำงาน
- Appointment บริการนัดหมายระหว่างสมาชิกในกลุ่ม
- Account Information การเก็บข้อมูลส่วนตัวของผู้ใช้
- Log out การออกจากระบบ

### 7.2.3 การรับส่งข้อความ ( Message )

ผู้ใช้สามารถรับและส่งข้อความระหว่างสมาชิกในกลุ่มได้ ซึ่งเหมือนกับการรับส่งข้อความระหว่าง 2 คนตามปกติ สามารถจัดการการเก็บรวบรวมข้อความที่ผู้อื่นส่งมาได้ จากหน้าหลักเมื่อผู้ใช้เลือก Message จะได้น้ำจอที่แสดงดังรูปที่ 7-5



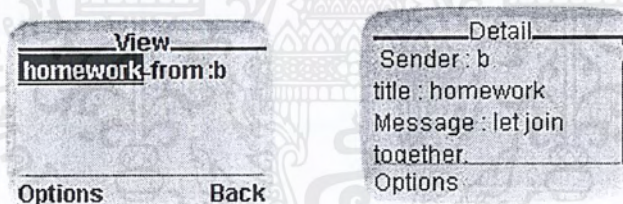
รูปที่ 7-5 แสดงหน้าจอ Message

ในหน้าจอนี้ สามารถเลือกการทำงานได้ 3 อย่าง คือ

- View ใช้สำหรับดูข้อความที่ผู้อื่นส่งมาให้
- Send ทำการส่งข้อความให้แก่ผู้อื่น
- Delete ใช้สำหรับลบข้อความที่ผู้อื่นส่งมา เป็นการจัดการการเก็บรวบรวมข้อความ

#### 7.2.3.1 การดูข้อความ ( View Message )

เป็นการดูข้อความที่ผู้อื่นส่งมา หลังจากเลือก View จากหน้าจอ Message จะแสดงรายชื่อ (title) ของข้อความที่ได้รับ และสามารถดูรายละเอียดของข้อความที่ส่งมาได้โดยการเลือกข้อความนั้น ดังแสดงในรูปที่ 7-6

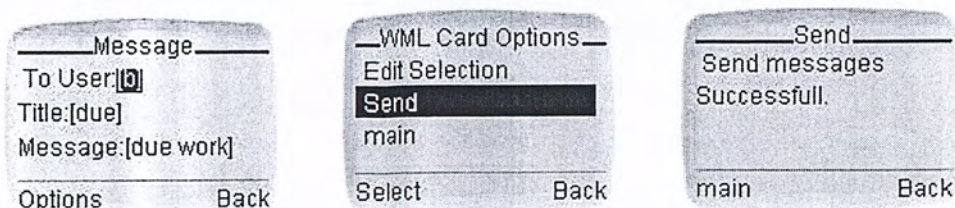


รูปที่ 7-6 การดูข้อความในบริการ Message

รายละเอียดของข้อความจะบอกถึง ผู้ส่ง, ชื่อหัวข้อข้อความ และข้อความที่ต้องการสื่อสาร

#### 7.2.3.2 การส่งข้อความ ( Send Message )

เป็นการส่งข้อความให้แก่ผู้อื่นที่เป็นสมาชิกในกลุ่มเดียวกัน ซึ่งมีขั้นตอนดังรูปที่ 7-7



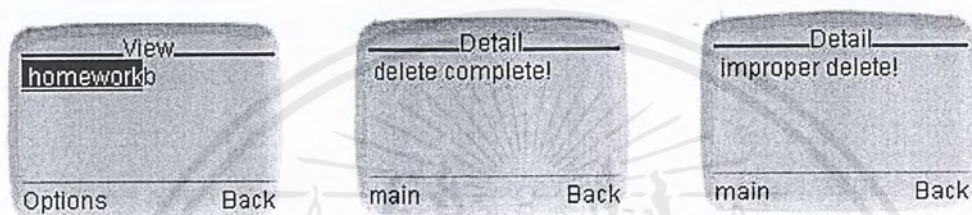
รูปที่ 7-7 แสดงขั้นตอนการส่งข้อความในบริการ Message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักผู้จัดทำนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่เลือก Send ในหน้าจอ Message จะขึ้นหน้าจอที่ให้ใส่ชื่อของผู้ที่ผู้ใช้จะส่งข้อความให้ หลังจากนั้นใส่ชื่อของข้อความ และข้อความที่ต้องการจะส่ง การส่งข้อความในโทรศัพท์มือถือควรจะเป็นข้อความสั้น ๆ ไม่ควรยาวเกินไป เพราะมีหน้าจอที่เล็กไม่สะดวกต่อการอ่านข้อความยาว ๆ เมื่อใส่ข้อความเรียบร้อยแล้ว ให้กด Options แล้วเลือก Send ระบบจะทำการส่งข้อความ ถ้าการส่งส่งได้เรียบร้อยแล้ว จะแสดงข้อความว่าส่งข้อความเรียบร้อยแล้วดังรูปสุดท้ายของรูป

### 7.2.3.3 การลบข้อความ ( Delete Message )

เป็นการลบข้อความที่ผู้อื่นส่งมาให้ ซึ่งมีขั้นตอนดังรูปที่ 7-8

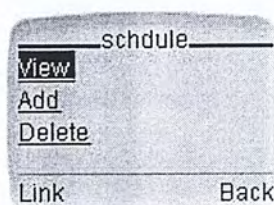


รูปที่ 7-8 แสดงขั้นตอนการลบข้อความในบริการ Message

หลังจากที่เลือก Delete ในหน้าจอ Message มันจะแสดงรายชื่อข้อความที่ผู้ใช้มีอยู่ ให้ผู้ใช้เลือกข้อความที่ต้องการลบ เมื่อเลือกแล้ว ระบบจะทำการลบข้อความนั้น ถ้าลบได้เรียบร้อยแล้ว จะแสดงข้อความว่าลบได้เรียบร้อยแล้วดังรูปที่สองในรูป แต่ถ้ามีปัญหาในการลบ เช่น การที่ผู้ใช้ใช้ Back กลับไปในหน้าจอเดิมหลังจากที่ทำการลบไปแล้ว และเลือกข้อความที่ลบไปแล้วมาทำการลบอีก จะทำให้เป็นการลบที่ไม่ถูกต้อง มันจะแสดงข้อความว่าเป็นการลบที่ไม่ถูกต้องดังรูปสุดท้ายในรูป

### 7.2.4 การจัดการเวลา ( Schedule )

เป็นการจัดการเวลาของผู้ใช้ โดยการเลือกบริการ Schedule ในหน้าหลักของการใช้โปรแกรม เป็นการจัดการเวลาส่วนตัว ว่าเวลาใดจะทำอะไรของตัวเอง ไม่ได้ทำการนัดหมายไปถึงผู้อื่น ซึ่งจะอยู่ในบริการ Appointment หลังจากเลือกบริการ Schedule แล้ว จะได้น้ำจอดังรูปที่ 7-9



รูปที่ 7-9 แสดงหน้าจอ Schedule

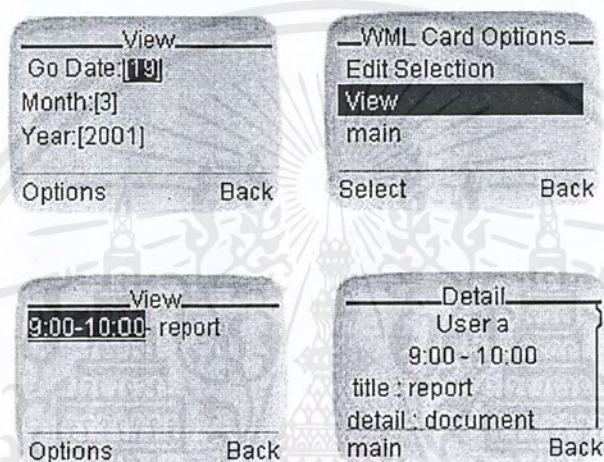
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้าจอ Schedule จะแสดงรายการในการใช้บริการดังนี้

- View ใช้ดูตารางเวลาที่ได้สร้างเอาไว้ เพื่อเป็นการเตือนให้ทำตามหมายกำหนดการ
- Add ใช้เพิ่มหมายกำหนดการเข้าไปในตารางเวลา
- Delete ใช้ลบหมายกำหนดการที่มีอยู่ในตารางเวลา

#### 7.2.4.1 การดูตารางเวลา ( View Schedule )

เป็นการขอดูหมายกำหนดการที่สร้างเอาไว้ โดยระบุวันที่ที่จะขอดู แล้วระบบจะแสดงหมายกำหนดการในวันทีระบุ ดังแสดงในรูปที่ 7-10

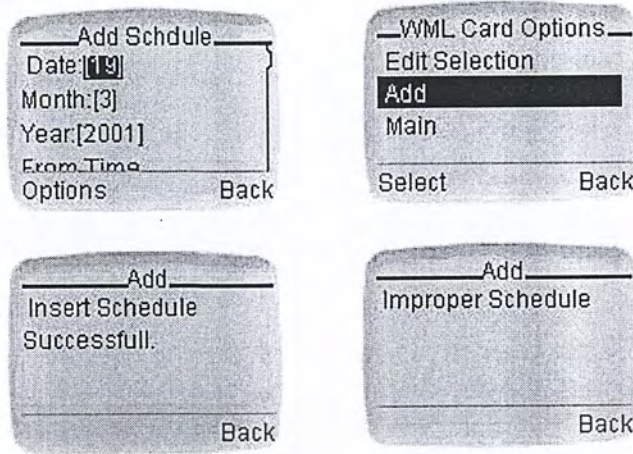


รูปที่ 7-10 แสดงขั้นตอนการดูตารางเวลาในบริการ Schedule

หลังจากเลือก View ในบริการ Schedule จะปรากฏหน้าจอสำหรับใส่วันที่ที่ต้องการดูหมายกำหนดการ เมื่อใส่วันที่เรียบร้อยแล้ว ให้กด Options เลือก View จากนั้นระบบจะแสดงรายการเวลาที่มีการกำหนดไว้พร้อมทั้งชื่อของหมายกำหนดการ ผู้ใช้สามารถรายละเอียดของหมายกำหนดการได้ โดยเลือกรายการที่ต้องการแล้วมันจะแสดงรายละเอียดออกมามีดังรูปสุดท้ายของรูป

#### 7.2.4.2 การเพิ่มหมายกำหนดการในตารางเวลา ( Add Schedule )

เป็นการเพิ่มหมายกำหนดการลงในตารางเวลาของผู้ใช้ ซึ่งมีขั้นตอนดังรูปที่ 7-11

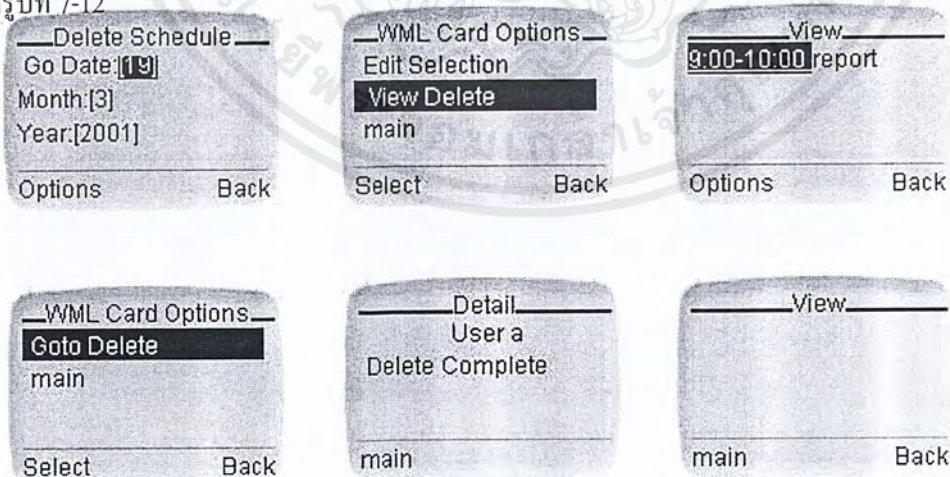


รูปที่ 7-11 แสดงขั้นตอนการเพิ่มหมายกำหนดการในบริการ Schedule

หลังจากที่เลือก Add ในหน้าจอ Schedule จะปรากฏหน้าจอสำหรับใส่วันที่, เวลาเริ่มต้นและเวลาที่สิ้นสุดของการทำงานตามหมายกำหนดการ, ชื่อของหมายกำหนดการ และรายละเอียดของหมายกำหนดการ เมื่อใส่ข้อมูลทั้งหมดเรียบร้อยแล้ว ให้กด Options เลือก Add ระบบจะทำการเพิ่มหมายกำหนดการเข้าไปในตารางเวลาของผู้ใช้ซึ่งอยู่ในฐานข้อมูล ถ้าการเพิ่มนี้ถูกต้อง จะแสดงข้อความว่าการเพิ่มนั้นสำเร็จเรียบร้อย แต่ถ้าการเพิ่มนี้มีปัญหา เช่น การเพิ่มหมายกำหนดการในวันที่เลขผ่านไปแล้ว มันจะแสดงข้อความว่าการเพิ่มหมายกำหนดการนั้น ไม่ถูกต้องดังรูปสุดท้ายในรูป

#### 7.2.4.3 การลบหมายกำหนดการในตารางเวลา ( Delete Schedule )

ผู้ใช้สามารถลบหมายกำหนดการที่ผู้ใช้ทำการเพิ่มเข้าไปในตารางเวลาได้ ซึ่งมีขั้นตอนดังแสดงในรูปที่ 7-12



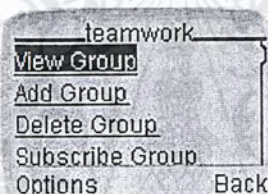
รูปที่ 7-12 แสดงขั้นตอนการลบหมายกำหนดการในบริการ Schedule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเลือก Delete ในหน้าจอ Schedule จะปรากฏหน้าจอสำหรับใส่วันที่ที่ต้องการลบรายการ  
 หมายกำหนดการ เมื่อใส่วันที่เรียบร้อยแล้ว ให้กด Options เลือก View Delete แล้วมันจะแสดงรายการ  
 หมายกำหนดการที่มีในวันที่ระบุ เลือกรายการที่ผู้ใช้ต้องการลบแล้วกด Options เลือก Goto Delete ระบบ  
 จะทำการลบรายการหมายกำหนดการนั้น หลังจากนั้น ถ้าผู้ใช้เข้าไปดูหมายกำหนดการ โดยการ View ผู้ใช้  
 จะไม่เห็นรายการที่ลบไปแล้วดังที่แสดงในรูปสุดท้ายของรูป

### 7.2.5 การจัดกลุ่มสมาชิก ( Team Working )

เป็นการจัดการกลุ่มสมาชิก เนื่องจากระบบนี้มีได้หลายกลุ่ม และมีผู้ใช้ได้หลายคน นั่นคือ กลุ่ม  
 หนึ่งมีได้หลายคน และคนหนึ่งอยู่ได้หลายกลุ่ม แต่ละกลุ่มจะมีหัวหน้ากลุ่มอยู่คนหนึ่ง ซึ่งเป็นคนที่เพิ่มชื่อ  
 กลุ่มเข้าไปในระบบ และคนอื่น ๆ สามารถขอเข้าเป็นสมาชิกได้ การจัดการนี้จะอยู่ในบริการ Team  
 Working หลังจากเลือกบริการ Team Working ในหน้าหลักของการใช้โปรแกรม จะได้หน้าจอแสดงใน  
 รูปที่ 7-13



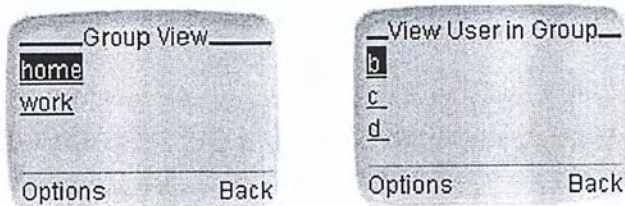
รูปที่ 7-13 แสดงหน้าจอ teamwork

ในหน้าจอ teamwork มีรายการการให้บริการ ดังนี้

- View Group เป็นการขอดูรายชื่อกลุ่มที่ผู้ใช้เป็นหัวหน้า และสามารถดูได้ว่ามีใครเป็นสมาชิกบ้าง
- Add Group เป็นการสร้างกลุ่มการทำงานเข้าไปในระบบ โดยที่ผู้สร้างจะเป็นหัวหน้ากลุ่ม
- Delete Group เป็นการเลือกลบกลุ่มการทำงานที่ผู้ใช้เป็นหัวหน้าออกจากระบบ
- Subscribe Group เป็นการขอเข้าเป็นสมาชิก ในกลุ่มการทำงานของผู้อื่น
- Unsubscribe Group เป็นการขอออกจากกลุ่มการทำงานที่ไปสมัครมาก่อนหน้านี้
- View Subscribe Group เป็นการขอดูว่าผู้ใช้เป็นสมาชิกในกลุ่มการทำงานใดบ้าง

#### 7.2.5.1 การขอดูกลุ่ม ( View Group )

เป็นการขอดูรายชื่อกลุ่มที่ผู้ใช้บริการเป็นหัวหน้า และสามารถดูได้ว่ามีใครเป็นสมาชิกบ้าง หลังจากเลือก View Group ในหน้าจอ teamwork จะได้หน้าจอที่แสดงรายชื่อกลุ่ม ดังแสดงในรูปที่ 7-14

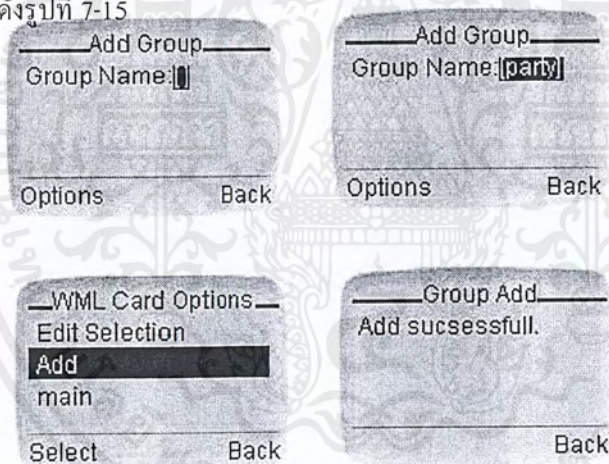


รูปที่ 7-14 แสดงการขอดูกลุ่มในบริการ teamwork

หลังจากที่แสดงรายชื่อกลุ่มแล้ว ผู้ใช้สามารถเลือกกลุ่มเพื่อเข้าไปดูว่ามีใครเป็นสมาชิกในกลุ่มบ้าง ดังแสดงในรูปสุดท้ายของรูป

#### 7.2.5.2 การเพิ่มกลุ่ม ( Add Group )

เป็นการเพิ่มกลุ่มการทำงานเข้าไปในระบบโดยผู้ให้บริการที่จะเป็นหัวหน้ากลุ่ม เป็นผู้ทำการสร้าง ซึ่งมีขั้นตอนดังรูปที่ 7-15



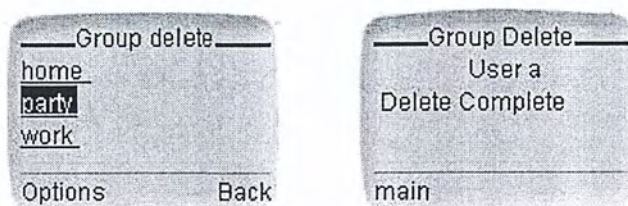
รูปที่ 7-15 แสดงขั้นตอนการเพิ่มกลุ่มในบริการ teamwork

หลังจากเลือก Add Group ในหน้าจอ teamwork จะปรากฏหน้าจอสำหรับใส่ชื่อกลุ่มที่ต้องการจะเพิ่มเข้าไปในระบบ เมื่อใส่ชื่อกลุ่มเสร็จแล้ว กด Options เลือก Add ระบบจะทำการเพิ่มกลุ่มเข้าไปในระบบ และแสดงข้อความว่าการเพิ่มสำเร็จเรียบร้อยแล้ว

#### 7.2.5.3 การลบกลุ่ม ( Delete Group )

เป็นการลบกลุ่มที่ผู้ใช้เป็นหัวหน้ากลุ่มได้สร้างไว้ในการ Add Group ออกจากระบบ โดยสามารถเลือกกลุ่มที่ได้ทำ หลังจากเลือก Delete Group ในหน้าจอ teamwork จะมีขั้นตอนในการลบกลุ่มดังแสดงในรูปที่ 7-16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

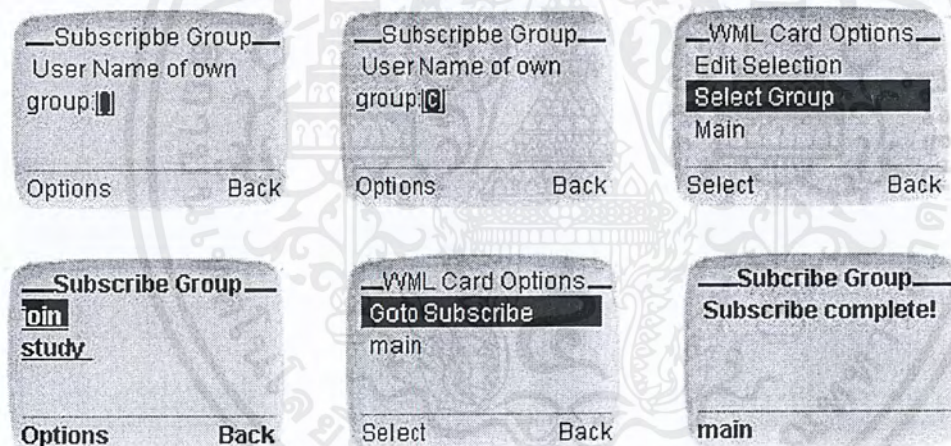


รูปที่ 7-16 แสดงขั้นตอนการลบกลุ่มในบริการ teamwork

หลังจากเลือก Delete Group จะปรากฏหน้าจอสำหรับเลือกกลุ่มที่ต้องการจะลบ เมื่อเลือกกลุ่มที่ต้องการจะลบแล้ว ระบบจะทำการลบกลุ่มนั้นออกจากระบบ และแสดงข้อความว่าการลบกลุ่มเสร็จสิ้นสมบูรณ์แล้ว

#### 7.2.5.4 การขอเข้าเป็นสมาชิกในกลุ่ม (Subscribe Group)

เป็นการขอเข้าเป็นสมาชิกในกลุ่มที่มีผู้อื่นเป็นหัวหน้า โดยมีขั้นตอนดังแสดงในรูปที่ 7-17



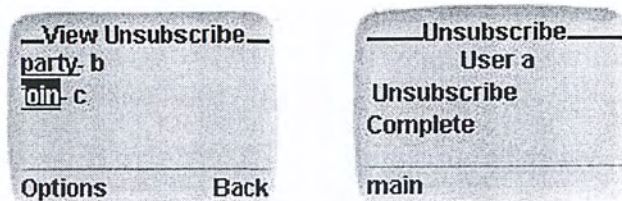
รูปที่ 7-17 แสดงขั้นตอนการขอเป็นสมาชิกในกลุ่มของบริการ teamwork

หลังจากเลือก Subscribe Group จะปรากฏหน้าจอสำหรับใส่ชื่อหัวหน้ากลุ่มคนอื่นที่ไม่ใช่ผู้ใช้ เมื่อใส่ชื่อแล้ว กด Options เลือก Select Group จะปรากฏหน้าจอที่แสดงรายชื่อกลุ่มที่บุคคลที่ระบุชื่อนั้นเป็นหัวหน้ากลุ่ม เลือกกลุ่มที่ต้องการเป็นสมาชิก แล้วกด Options เลือก Goto Subscribe ระบบจะทำให้ผู้ใช้เป็นสมาชิกในกลุ่มนั้น

#### 7.2.5.5 การขอลออกจากกลุ่ม (Unsubscribe Group)

เป็นการขอลออกจากกลุ่มที่มีผู้อื่นเป็นหัวหน้า ซึ่งมีขั้นตอนดังแสดงในรูปที่ 7-18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

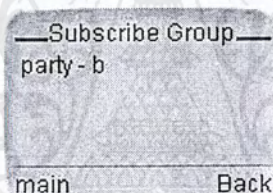


รูปที่ 7-18 แสดงขั้นตอนการขอยกออกจากกลุ่มของบริการ teamwork

หลังจากเลือก Unsubscribe Group แล้ว จะปรากฏหน้าจอที่แสดงรายชื่อกลุ่มที่ผู้ใช้เป็นสมาชิก พร้อมด้วยชื่อหัวหน้ากลุ่ม เลือกกลุ่มที่ต้องการจะออกจากการเป็นสมาชิก ระบบจะทำให้ผู้ใช้ออกจากกลุ่มนั้น และแสดงข้อความการออกจากกลุ่ม

#### 7.2.5.6 การขอดูรายชื่อกลุ่มที่ผู้ใช้เป็นสมาชิก ( View Subscribe Group )

เป็นการขอดูรายชื่อกลุ่มที่ผู้ใช้เป็นสมาชิกอยู่ โดยที่ผู้ใช้ไม่ได้เป็นหัวหน้า ซึ่งแสดงดังรูปที่ 7-19



รูปที่ 7-19 แสดงการดูรายชื่อกลุ่มที่ผู้ใช้เป็นสมาชิก

หลังจากเลือก View Subscribe Group ในหน้าจอ teamwork จะปรากฏหน้าจอที่แสดงรายชื่อกลุ่มที่ผู้ใช้เป็นสมาชิก พร้อมทั้งชื่อของหัวหน้ากลุ่มนั้น ๆ ด้วย

#### 7.2.6 การจัดการนัดหมาย ( Appointment )

การนัดหมายของหัวหน้ากลุ่ม สามารถส่งการนัดหมายไปยังสมาชิกคนอื่น ๆ ได้โดยอัตโนมัติ การนัดหมายต้องการรู้ขอบเขตเวลาโดยกว้าง และจำนวนชั่วโมงที่ต้องการใช้ จากนั้นระบบจะตรวจเช็คกับฐานข้อมูลที่เกี่ยวข้องกับตารางเวลาของแต่ละคนที่เป็นสมาชิก เพื่อดูว่าช่วงเวลาใดที่ว่างตรงกัน และพอเพียงกับชั่วโมงที่ต้องการใช้ เมื่อหาช่วงเวลาได้แล้ว จะแสดงช่วงเวลาที่สามารถทำได้ เพื่อให้ผู้ใช้ที่เป็นหัวหน้ากลุ่มทำการเลือกช่วงเวลาที่ต้องการ จากนั้นจะส่งการนัดหมายไปให้สมาชิกในกลุ่มทุกคน ผู้ใช้สามารถดูการนัดหมายทั้งจากกลุ่มที่ผู้ใช้เองเป็นหัวหน้า และจากกลุ่มที่มีผู้อื่นเป็นหัวหน้าและผู้ใช้เป็นสมาชิก

จากที่กล่าวมาข้างต้น การนัดหมายจึงแบ่งบริการออกเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- New Appointment คือบริการที่ให้ผู้ใช้นัดหมายสมาชิกในกลุ่มของตนได้ ผู้ใช้ทำการสร้างการนัดหมายใหม่ แล้วส่งออกไปให้แก่สมาชิกในกลุ่ม
  - View Appointment คือบริการที่ให้ผู้ใช้งานสามารถดูการนัดหมายที่ผู้ใช้เป็นคนสร้างเอง
  - Delete Appointment คือบริการที่ให้ผู้เป็นหัวหน้ากลุ่มยกเลิกการนัดหมาย
  - Get Appointment คือบริการที่ให้ผู้เป็นสมาชิกไว้ดูว่ามีการนัดหมายจากกลุ่มการทำงาน
  - Requested Appointment คือบริการที่ให้ผู้ใช้ได้ถูกร้องขอจากหัวหน้ากลุ่มเพื่อขอเวลาในการนัดหมาย
- หน้าจอที่ปรากฏหลังจากที่เลือก Appointment ในหน้าหลักของการใช้โปรแกรม จะแสดงดังรูปที่

7-20

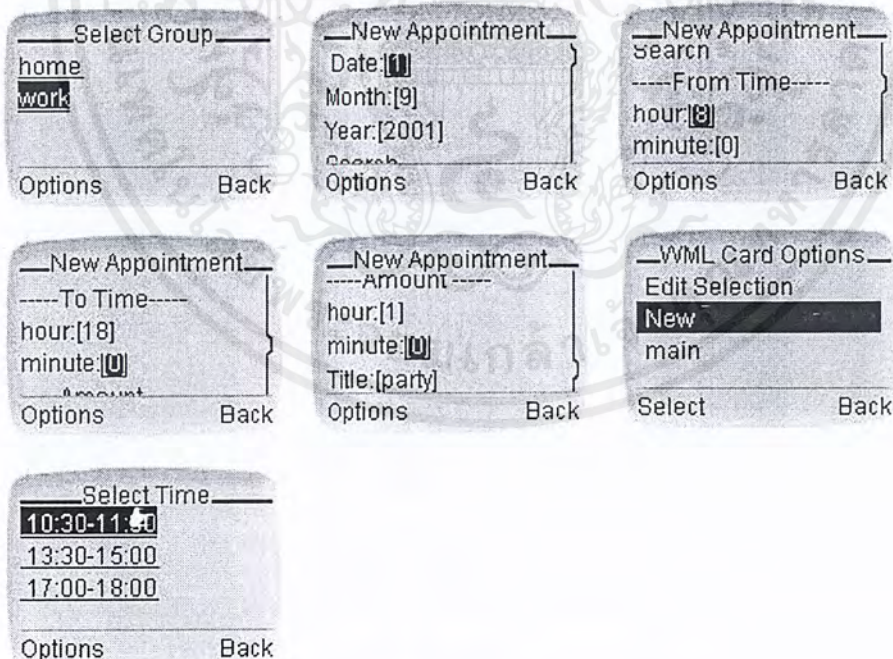


รูปที่ 7-20 แสดงหน้าจอ Appointment

#### 7.2.6.1 การสร้างการนัดหมาย (New Appointment)

เลือก New Appointment เพื่อทำการสร้างการนัดหมายที่ผู้ใช้เป็นหัวหน้ากลุ่ม ซึ่งมีขั้นตอนดังรูป

รูปที่ 7-21

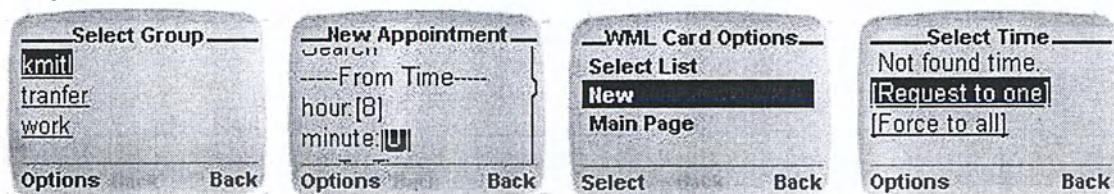


รูปที่ 7-21 แสดงขั้นตอนการสร้างการนัดหมาย

หลังจากเลือก New Appointment แล้ว จะปรากฏหน้าจอที่แสดงรายชื่อกลุ่มที่ผู้ใช้เป็นหัวหน้ากลุ่ม ให้เลือกกลุ่มที่ต้องการทำการนัดหมาย จากนั้นจะปรากฏหน้าจอที่ให้ใส่วันที่, ขอบเขตเวลาเริ่มต้น, ขอบเขตเวลาสิ้นสุด, จำนวนชั่วโมง, ชื่อหัวข้อการนัดหมาย, และรายละเอียดเกี่ยวกับการนัดหมาย เมื่อใส่เอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลครบแล้วให้กด Options เลือก New ระบบจะทำการหาช่วงเวลา และแสดงรายการออกมาให้ผู้ใช้เลือก เมื่อผู้ใช้เลือกช่วงแล้ว ในกรณีที่หาช่วงเวลาได้ ผู้เป็นหัวหน้ากลุ่มจะทำการเลือกช่วงเวลาและระบบจะทำการส่งการนัดหมายนี้ไปให้สมาชิกในกลุ่มคนอื่น ๆ ต่อไป

ในกรณีที่หาช่วงเวลาไม่พบระบบก็จะให้ผู้ใช้เป็นหัวหน้าทำการเลือกว่าจะใช้วิธีใดในการหาเวลาดังรูปที่ 7-22

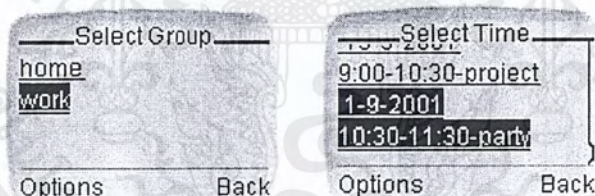


รูปที่ 7-22 แสดงการเลือกวิธีการนัดหมาย

โดยระบบจะให้เลือกว่าจะทำการขอเวลาไปยังบุคคลใดบุคคลหนึ่ง หรือกำหนดช่วงเวลาโดยไม่สามารถใจว่าสมาชิกในกลุ่มจะว่างหรือไม่

#### 7.2.6.2 การดูการนัดหมาย ( View Appointment )

เป็นการดูการนัดหมายที่ผู้ใช้เป็นคนสร้างขึ้น มา มีขั้นตอนดังแสดงในรูปที่ 7-23



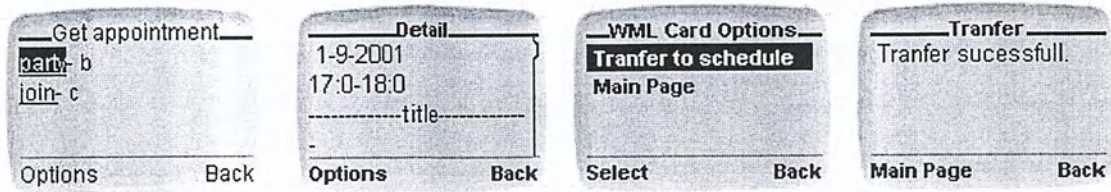
รูปที่ 7-23 แสดงขั้นตอนการดูการนัดหมาย

หลังจากเลือก View Appointment จะปรากฏหน้าจอที่ให้เลือกรุ่นที่ต้องการดูการนัดหมาย จากนั้นจะแสดงการนัดหมายที่มีในกลุ่มนั้น แสดงวัน, เวลา และชื่อการนัดหมาย ดังแสดงในรูป

#### 7.2.6.3 การรับการนัดหมาย ( Get Appointment )

เป็นการดูการนัดหมายจากกลุ่มการทำงาน ซึ่งแสดงขั้นตอนในรูปที่ 7-24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

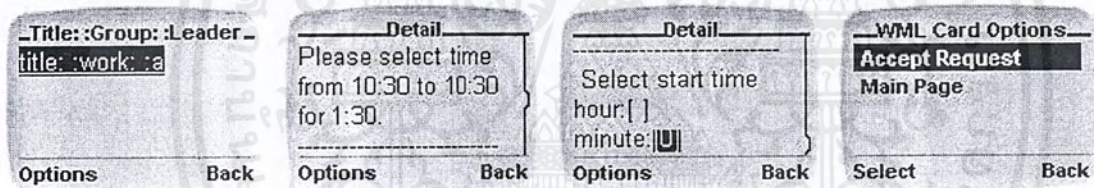


รูปที่ 7-24 แสดงขั้นตอนการรับการนัดหมาย

หลังจากเลือก Get Appointment จะปรากฏหน้าจอที่ให้เลือกรายชื่อกลุ่มที่มีผู้อื่นเป็นหัวหน้า รายชื่อกลุ่มจะแสดงชื่อหัวหน้ากลุ่มไว้ด้วย เมื่อเลือกรายชื่อที่ต้องการจะดูแล้ว มันจะแสดงวัน, เวลา, ชื่อการนัดหมาย และรายละเอียดการนัดหมายให้แก่ผู้ใช้ เมื่อผู้ใช้ต้องการให้เวลาการนัดหมายอยู่ในตารางเวลาของตัวเองก็สามารถถ่ายโอนมายังตารางเวลาของตัวเองได้โดยการเลือก Transfer to schedule

#### 7.2.6.4 การขอเวลานัดหมาย ( Requested Appointment )

เป็นการคว่ามีการขอเวลาจากกลุ่มการทำงานเพื่อที่จะนัดหมาย แต่เนื่องจากเวลาของผู้ใช้ไม่ว่าง จึงมีการขอเวลามาด้งในรูปที่ 7-25

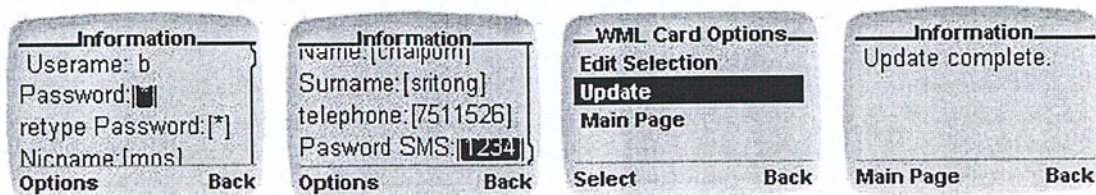


รูปที่ 7-25 แสดงหน้าจอ Requested Appointment

เมื่อเลือกที่จะพิจารณาการทำงานไหนแล้วก็เข้าไปยังหน้าจอที่บอกรายละเอียดว่าสามารถเลือกเวลาใดได้บ้าง เมื่อผู้ถูกขอสามารถให้เวลาได้ก็จะทำการเลือกเวลาที่อยู่ในช่วงที่กำหนดไว้ส่งไปยังหัวหน้ากลุ่มต่อไป

#### 7.2.7 การเก็บข้อมูลส่วนตัวของผู้ใช้ ( Account Information )

เป็นการเก็บข้อมูลส่วนตัวของผู้ใช้ เช่น ชื่อ, นามสกุล, ชื่อเล่น, เบอร์โทรศัพท์ รหัสที่ใช้ในการเข้าใช้บริการ ซึ่งผู้ใช้สามารถแก้ไขข้อมูลของตัวเองได้ แสดงข้อมูลด้งในรูปที่ 7-26



รูปที่ 7-26 แสดงหน้าจอ Information ในบริการ Account Information



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

### การตรวจสอบการทำงานของฟังก์ชัน

#### 8.1 การตรวจสอบการทำงานของฟังก์ชัน ( Functional Testing )

เมื่อโปรแกรมได้มีการทำถึงขั้นตอนของการเขียนโค้ดแล้ว ขั้นตอนต่อไปในการสร้างซอฟต์แวร์ ( Software ) ตามหลักการของวิศวกรรมซอฟต์แวร์ ( Software Engineering ) ก็คือการทดสอบการทำงานของโปรแกรมเพื่อหาข้อผิดพลาด ( Error ) และทำการแก้ไข เพื่อให้ได้มาซึ่งโปรแกรมที่ไม่มีข้อผิดพลาด

แต่เนื่องด้วยข้อจำกัดทางเศรษฐศาสตร์ เช่น เวลา ค่าใช้จ่าย และแรงงาน จึงทำให้การทดสอบโปรแกรมทำได้เพียงบางส่วนเท่านั้น ดังนั้นในการทดสอบโปรแกรมการนัดหมายดำเนินการผ่านเว็บ จึงสามารถทดสอบได้เพียงบางส่วน ดังตัวอย่างที่แสดงในผลการทดสอบ

#### 8.2 จุดประสงค์การทดสอบ

เป้าหมายในการทดสอบครั้งนี้ก็เพื่อหาข้อผิดพลาดที่อาจเกิดขึ้นของโปรแกรม โดยจะสร้างสถานการณ์จำลองขึ้นมา แล้วดูว่าการทำงานของฟังก์ชันถูกต้องหรือไม่

#### 8.3 วิธีการทดสอบ

1. เรียกโปรแกรมการนัดหมายดำเนินการ
2. เลือกไปที่ Sign Up เพื่อการลงทะเบียนสมาชิกใหม่
3. ป้อนข้อมูลตามขั้นตอน
4. ทำการตรวจสอบค่าในฐานข้อมูลว่าตรงกับที่ได้ป้อนลงไปหรือไม่
5. เริ่มเข้าสู่ระบบโดยการ login
6. ตรวจสอบว่าเมื่อมีการ login โดยผู้ใช้อื่นในช่วงเวลาเดียวกัน เกิดการผิดพลาดระหว่างกันหรือไม่
7. ทดลองใช้บริการส่งข้อความส่งถึงผู้อื่น
8. ตรวจสอบจากคนรับว่าถูกต้องหรือไม่
9. ทดลองใช้บริการจัดการตารางเวลา
10. ตรวจสอบว่าเมื่อบันทึกแล้วเกิดการซ้อนทับกับตารางเวลาเดิมหรือไม่
11. ทดลองใช้บริการจัดกลุ่มสมาชิก โดยการสร้างกลุ่มสมาชิกและสมัครเข้ากลุ่มสมาชิก
12. ตรวจสอบว่ามีกลุ่มที่สร้างอยู่หรือไม่ และสามารถไปอยู่กลุ่มสมาชิกของผู้อื่นได้หรือไม่
13. ทดลองบริการนัดหมายสมาชิกโดยใส่ข้อมูลจำลองเข้าไป
14. ตรวจสอบว่าเวลาที่ได้ถูกต้องหรือไม่
15. ทดลองใส่ตารางเวลาที่ต้องการ
16. ตรวจสอบว่ามีการแจ้งเตือนหรือไม่
17. ทดลองเปลี่ยนแปลงข้อมูลของตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18. ตรวจสอบกับฐานข้อมูลว่าเปลี่ยนตามด้วยหรือไม่
19. ทดสอบออกจากระบบ
20. ตรวจสอบว่าไปหน้าที่เราเคยไปมาแล้วในระบบได้หรือไม่

#### 8.4 ผลการทดสอบ

บริการ	ผลการทดลอง
การสมัครสมาชิก	✓
การเข้าสู่ระบบ	✓
การส่งข้อความ	✓
การใส่เวลานัดหมาย	✓
การจัดกลุ่มสมาชิก	✓
การนัดหมาย	✓
การแจ้งเตือน	✓ และ X
การออกจากระบบ	✓

ตารางที่ 8-1 ผลการทดสอบการทำงานของฟังก์ชัน

#### 8.5 สรุปผลการทดสอบ

จากผลที่ได้ในตาราง สามารถพิจารณาได้ว่าฟังก์ชันย่อยต่าง ๆ ในแต่ละฟังก์ชันหลักของโปรแกรมได้มีการเรียกใช้ให้ทำงานครบทุกฟังก์ชัน และให้ผลออกมาเป็นที่น่าพอใจ คือโปรแกรมสามารถรองรับการทำงานได้ทุกกรณี แต่ยังไม่สมบูรณ์ก็เพราะการเรียกใช้ CGI ของผู้ให้บริการโทรศัพท์เคลื่อนที่ซึ่งไม่ดีพอบางที่ก็ส่งได้ บางที่ก็ส่งไม่ได้ และในบางส่วนยังไม่สามารถสรุปหรือรับรองโปรแกรมได้ว่าจะทำงานได้ในทุกกรณีของ Data ที่ป้อนให้ เนื่องจากการทดสอบนี้เป็นเพียงการยกตัวอย่างข้อมูลส่งให้โปรแกรมทำงานเท่านั้น

## บทที่ 9

### บทวิจารณ์และสรุป

#### 9.1 สรุปผลเปรียบเทียบกับเป้าหมายและวัตถุประสงค์

การทำงานของระบบสามารถจัดการการนัดหมายระหว่างสมาชิกในกลุ่มได้ ซึ่งเป็นไปตามเป้าหมายหลักของงานวิจัย นอกจากนี้ก็มีการให้บริการอื่น ๆ อีก การให้บริการของระบบได้แก่

- การลงทะเบียนเพื่อเข้าสู่ระบบ
- การเข้าสู่ระบบ
- การรับส่งข้อความ
- การจัดตารางเวลา
- การจัดกลุ่มสมาชิก
- การจัดการการนัดหมาย
- การเก็บข้อมูลส่วนตัวของผู้ใช้
- การออกจากระบบ

จากบริการข้างต้นแม้จะไม่ครอบคลุมถึงความต้องการของผู้ใช้ทั้งหมด แต่ก็สามารถใช้งานได้ ถ้าคิดตามวัตถุประสงค์เพื่อการศึกษาแนวทางการใช้งานเว็บ ก็สามารถเข้าใจถึงหลักการทำงานและแนวทางการนำไปใช้ประโยชน์ได้จริง

#### 9.2 แนวทางการพัฒนาต่อ

- เพิ่มบริการให้มากขึ้นเพื่อครอบคลุมความต้องการของผู้ใช้
- เน้นอินเตอร์เฟซให้หลากหลาย ให้ผู้ใช้ใช้งานได้สะดวกมากขึ้น
- เสริมความชาญฉลาดในการนัดหมายให้เพิ่มมากขึ้น
- การเข้ารหัสให้กับข้อมูลเพราะในขณะนี้ผู้ดูแลระบบสามารถดูข้อมูลของทุกคนได้
- การทำบริการให้ใช้ได้บนเว็บ จะทำให้การใส่ข้อมูลเป็นไปด้วยความสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### การติดตั้งโปรแกรม

#### 1. การติดตั้ง Java™ 2 SDK, Standard Edition

1.1 ใช้โปรแกรม j2se version 1.3 สามารถดาวน์โหลดได้ที่

<http://java.sun.com/j2se/1.3>

เลือกดาวน์โหลดอันเดียวกับระบบปฏิบัติการแล้วทำการติดตั้งไว้ในที่ต้องการ

เช่น c:\java\jdk1.3

1.2 เปลี่ยนแปลง System Environment Variable

โดยเพิ่มตัวแปร JAVA\_HOME, PATH และ CLASSPATH

ตั้งค่า JAVA\_HOME ไว้ที่ติดตั้งไว้ในข้อ 1.1 เช่น

JAVA\_HOME = c:\java\jdk1.3

PATH = %PATH%;%JAVA\_HOME%\bin

CLASSPATH = %JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar

#### 2. การติดตั้ง Apache Tomcat Web Server

2.1 ใช้โปรแกรม Apache Tomcat version 3.2 สามารถดาวน์โหลดได้ที่

<http://jakarta.apache.org/site/binindex.html>

2.2 ติดตั้งโดยการขยายไฟล์ที่ดาวน์โหลดมาไว้ใน directory ที่ต้องการเช่น C:\TOMCAT

2.3 เปลี่ยนแปลง System Environment Variable

ตั้งค่า TOMCAT\_HOME เช่น

TOMCAT\_HOME = "C:\TOMCAT"

2.4 ในไฟล์ <path to tomcat>\conf\server.xml เปลี่ยนบรรทัด

<Parameter name="port" value="8080"/> เป็น

<Parameter name="port" value="80"/>

#### 3. การติดตั้ง MySQL

3.1 ใช้โปรแกรม MySQL version 3.23 สามารถดาวน์โหลดได้ที่

<http://www.mysql.com/downloads/index.html>

3.2 ทำการติดตั้งโปรแกรมตามขั้นตอนที่ปรากฏบนหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 ทำการติดตั้ง JDBC Driver type 4 version 2.0 ของ MySQL โดยดาวน์โหลดที่

<http://mymysql.sourceforge.net>

เมื่อได้มาแล้ว ทำการเปลี่ยนแปลง CLASSPATH ให้ไปที่ที่ไฟล์นั้นเช่น

```
CLASSPATH = %CLASSPATH%;c:\java\jdk1.3\lib\mm.mysql-2.0.2-bin.jar
```

### 3.4 ทำการสร้างฐานข้อมูลด้วยคำสั่ง

#### 3.4.1 สร้าง database ด้วยคำสั่ง

```
create database project_teamworking;
```

#### 3.4.2 สร้างตารางด้วยคำสั่ง

```
create table gmember_table (
gid int not null,
member varchar(15) not null,
status varchar(15),
primary key (gid,member)
);

create table message_table (
massid int not null default '0' auto_increment,
sender varchar(15),
receiver varchar(15),
title varchar(15),
message varchar(100),
primary key (massid)
);
```

```
create table reschedule_table (
reschedule_id int not null default '0' auto_increment,
syear int ,
smmonth int ,
sdate int ,
hour_time int ,
minute_time int,
minute_to int ,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hour_to int ,
ahour int,
aminute int,
title varchar(15),
detail varchar(100),
gid varchar(15),
user_name varchar(15),
primary key (reschedule_id)
);

```

```

create table schedule_table (
schedule_id int not null default '0' auto_increment,
syear int not null,
smonth int not null,
sdate int not null,
hour_time int not null,
minute_time int not null,
minute_to int not null,
hour_to int not null,
title varchar(15),
detail varchar(100),
user_name varchar(15) not null,
primary key (schedule_id)
);

```

```

create table teamwork_table (
gid int not null default '0' auto_increment,
gname varchar(15),
oname varchar(15),
primary key (gid)
);

```

```

create table tschedule_table (
tschedule_id int not null default '0' auto_increment,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

syear int not null,
smonth int not null,
sdate int not null,
hour_time int not null,
minute_time int not null,
hour_to int not null,
minute_to int not null,
title varchar(15),
detail varchar(100),
gid varchar(15) not null,
primary key (tschedule_id)
);

create table user_table (
user_name varchar(15) not null,
user_pass varchar(15),
nickname varchar(15),
name varchar(15),
surname varchar(15),
telephone varchar(15),
sms varchar(15),
primary key(user_name)
);

```

#### 4. การติดตั้งโปรแกรมนัดหมายดำเนินงานผ่านเว็บ

นำไดรกทอรี jsp ไว้ที่ <path to Tomcat>\webapps\root\  
นำไดรกทอรี classes ไว้ที่ <path to Tomcat>\webapps\root\web-inf\  
นำไดรกทอรี server ไว้ที่ <path to Tomcat>\  
นำไดรกทอรี project\_teamworking ไว้ที่ <path to MySQL>\data\

#### หมายเหตุ

ใน Windows NT ตั้งค่าใน Control Panel -> System -> Environment

ใน Windows 95-98 ตั้งค่าในไฟล์ AUTOEXEC.BAT โดยมีคำว่า "set" นำหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

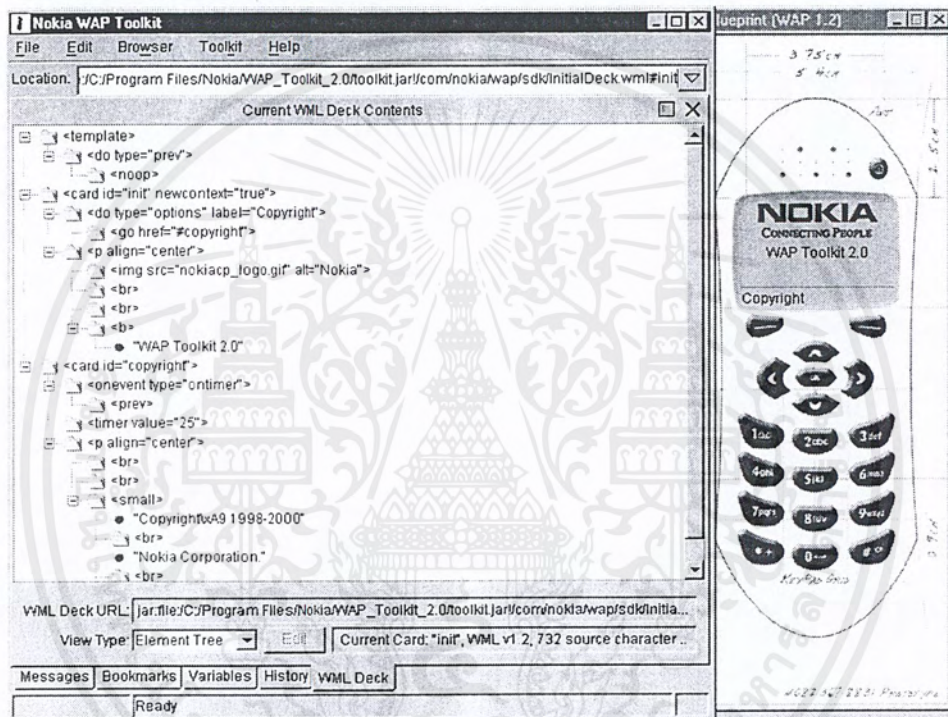
## ภาคผนวก ข

# TOOL ที่ใช้พัฒนาในโครงการ

### 1. Nokia WAP Toolkit

เป็นโปรแกรมที่ใช้จำลองการทำงานของ WML Browser ของโทรศัพท์เคลื่อนที่บน computer มีความสามารถในการแก้ไขและแสดงผลภาษา WML สามารถดาวน์โหลดได้ที่

<http://www.forum.nokia.com/main.html>



รูปที่ ข-1 Nokia WAP Toolkit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] Hunter, Jason et al: “ *Java Servlet Programming* “ , O’Reilly & Associate, Inc.,2000.
- [2] Cooper James W.: “ *Using JDBC to Create Database Objects : Principles of Object – Oriented Programming in Java* “ , IBM T.J. Watson Research Center , [www.ibm.com/java/](http://www.ibm.com/java/),2000.
- [3] Karl Moss: “ *Java Servlets* “ , McGraw.Hill,1999.
- [4] Steve Mann: “ *Programming Applications with the Wireless Application Protocol* “ , John Wiley & Sons,Inc.,1999.
- [5] “*Java Server Page* “ , <http://java.sun.com/products/jsp>,2000.
- [6] “*Apache Tomcat* “ , <http://java.sun.com/products/jsp>,2000.
- [7] “*MySQL* “ , <http://www.mysql.com>,2000.
- [8] “*Nokia WAP Toolkit* “ , <http://www.forum.nokia.com>,2000.
- [9] Larnie Pekowsky: “ *JavaServer Pages* “ , Addison Wesley Longman, Inc.,2000.

