

ระปปร้ป-ส่ง E-mail ผ่าน web

Web Mail



นาย ไพสิฐ พัฒนอมร

เลขหมู่.....

เลขทะเบียน 61806

วัน,เดือน,ปี 21 ก.ค. 2549

b.....

i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรับ-ส่ง E-mail ผ่าน web

Web Mail



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตรปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

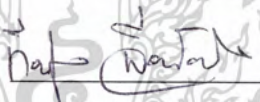
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรับ-ส่ง E-mail ผ่าน web

Web Mail

ผู้จัดทำ

1. นาย ไพสิฐ พัฒนอมร รหัสประจำตัว 44010354



อาจารย์ที่ปรึกษา

อ. เกียรติกุล เจียรนัยชนะกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรับ-ส่ง E-mail ผ่าน web

นาย ไพสิฐ พัฒนอมร 44010354

อ. เกียรติภูม เจียรนัยชนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

ปัจจุบันนี้คอมพิวเตอร์เข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก และ อินเทอร์เน็ตก็มีใช้กันอย่างแพร่หลาย หลายสิ่งหลายอย่างสามารถหาได้จากอินเทอร์เน็ต รวมทั้งการรับ-ส่งจดหมายอิเล็กทรอนิกส์ก็มีขึ้นเพื่ออำนวยความสะดวกให้กับผู้ใช้งาน ซึ่งระบบรับ-ส่ง เมล์นั้นสามารถเขียนได้หลายภาษา เช่น ASP , PHP เป็นต้น

วิทยานิพนธ์ฉบับนี้เป็นอีกแนวทางหนึ่งในการสร้างระบบรับ-ส่ง อีเมลล์ โดยใช้ภาษา JSP บนระบบปฏิบัติการลินุกซ์ โดยระบบสามารถ ตรวจจับไวรัส และ สแปมเมลล์ได้ ซึ่งเป็นปัญหาที่พบเจอมากทางอินเทอร์เน็ต ในปัจจุบัน และเนื่องจาก ภาษา JSP ดีกว่าภาษาอื่นตรงที่สามารถเรียนรู้ได้ง่าย และสามารถนำโปรแกรมไปใช้งานบนระบบปฏิบัติการของคอมพิวเตอร์แบบใดก็ได้โดยไม่ต้องทำการเปลี่ยนแปลงใดๆทั้งสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Mail

Mr. Paisit Pattana-amom

Mr. Kietikul Jearanaitanakij advisor

ABSTRACT

Recently , Computer has influenced for everyday life a lot and Internet mostly used worldwide. Everything can be found from Internet including receiving and sending E-mail for convenience of the user. Web Mail has been written in many language for example ASP, PHP

This thesis is one way for making Web Mail by using JSP on Linux .This system can detect viruses and spam mail, which is the most problem from internet nowadays. Because of JSP is better than other language due to easy to learn and can be used for every Operating System without any change.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้อาจไม่สำเร็จได้ด้วยดี ถ้าหากไม่ได้รับความช่วยเหลือ และ ความร่วมมือ จากหลายๆฝ่ายด้วยกัน บุคคลแรกที่ควรจะกล่าวถึง คือ อาจารย์ เกียรติภูมิจัย ภิรมย์ธนกิจ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้โอกาส ให้ความเอาใจใส่ แนะนำ และ ช่วยเหลือผม เสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ที่เลี้ยงดู และ เอาใจใส่ เสมอมา ในทุกๆ ด้าน ขอกราบขอบพระคุณมา ณ ที่นี้

ไพสิฐ พัฒนอมร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 เป้าหมายและขอบเขตของโครงการ	1
1.4 ขั้นตอนวิธีการดำเนินงาน	1
1.5 ประโยชน์ที่ได้รับจากโครงการนี้	2
บทที่ 2 การติดตั้งโปรแกรม สำหรับระบบปฏิบัติการลินุกซ์	3
2.1 โปรแกรมที่ต้องการของระบบ	3
2.2 การติดตั้ง DNS Server	3
2.3 การติดตั้ง Sendmail	7
2.4 การติดตั้ง JDK และ package ต่างๆ	10
2.5 การติดตั้ง Jakarta Tomcat	11
บทที่ 3 ทฤษฎี	13
3.1 DNS	13
3.2 ภาษาจาวา และการทำงานของ JSP	15
3.3 หลักการทำงานของ E-mail และ โพรโตคอลต่างๆ	17
3.3.1 POP3	17
3.3.2 IMAP	18
3.3.3 SMTP	20
3.3.4 MIME	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
บทที่ 4 การออกแบบโปรแกรม	22
4.1 รายละเอียดของโปรแกรม	22
4.2 เครื่องมือที่ใช้ในการพัฒนา	25
4.3 การทดสอบการทำงาน	26
4.4 โครงสร้างโปรแกรม (Diagrams ต่างๆ)	27
บทที่ 5 สรุปและวิจารณ์	38
5.1 ปัญหาและอุปสรรคในการทำงาน	38
5.2 แนวทางการแก้ไขปัญหาและอุปสรรค	38
5.3 แนวทางการพัฒนาต่อไปในอนาคต	38
5.4 ข้อสรุปและข้อเสนอแนะ	38
บรรณานุกรม	39
ภาคผนวก ก.	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้าที่

รูปที่ 2-1 ไฟล์ /etc/named.conf	4
รูปที่ 2-2 ไฟล์ localhost.zone	4
รูปที่ 2-3 ไฟล์ name.local	5
รูปที่ 2-4 ไฟล์ jonnetwork.zone	5
รูปที่ 2-5 ไฟล์ /etc/nsswitch.conf	6
รูปที่ 2-6 ไฟล์ /etc/resolv.conf	6
รูปที่ 2-7 ไฟล์ /etc/hosts	6
รูปที่ 2-8 การแสดงค่า macro ในไฟล์ sendmail.cf	7
รูปที่ 2-9 ไฟล์ local-host-names	8
รูปที่ 2-10 ไฟล์ access	8
รูปที่ 2-11 การแก้ไขไฟล์ ipop3	8
รูปที่ 2-12 การแก้ไขไฟล์ imap	9
รูปที่ 2-13 แสดงการเปิดพอร์ต 110และ143 (POP3/IMAP)	9
รูปที่ 2-14 ไฟล์ sendmail.mc	10
รูปที่ 3-1 การทำงานร่วมกับระหว่าง DNS กับ Web Server	13
รูปที่ 3-2 การใช้ nslookup ตรวจสอบค่า DNS ของ Mail Server	14
รูปที่ 3-3 ตัวอย่างการใช้โปรแกรม dig	14
รูปที่ 3-4 การแปลงและคอมไพล์ JSP ไฟล์	16
รูปที่ 4-1 ภาพรวมของระบบ	22
รูปที่ 4-2 แสดงหน้าการ เข้าสู่ระบบ แบบคร่าวๆ	23
รูปที่ 4-3 แสดงหน้า main page	23
รูปที่ 4-4 แสดงโค้ดการเรียกดู INBOX	24
รูปที่ 4-5 แสดงไฟล์ compose.jsp	24
รูปที่ 4-6 แสดงไฟล์ send.jsp	25
รูปที่ 4-7 แสดงการเรียกดู INBOX ผ่าน Mozilla Mail	26
รูปที่ 4-8 แสดงการเรียกดู INBOX ด้วย web browser	27
รูปที่ 4-9 Usecase Diagram	27
รูปที่ 4-10 Class Diagram	28
รูปที่ 4-11 Object Diagram	28
รูปที่ 4-12 Login sequence diagram	29
รูปที่ 4-13 Login collaboration diagram	29

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นๆ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

	หน้าที่
รูปที่ 4-14 Show INBOX sequence diagram	30
รูปที่ 4-15 Show INBOX collaboration diagram	30
รูปที่ 4-16 Show message sequence diagram	31
รูปที่ 4-17 Show message collaboration diagram	31
รูปที่ 4-18 Show Sent sequence diagram	32
รูปที่ 4-19 Show Sent collaboration diagram	32
รูปที่ 4-20 Show Folder sequence diagram	33
รูปที่ 4-21 Show Folder collaboration diagram	33
รูปที่ 4-22 Show Trash sequence diagram	34
รูปที่ 4-23 Show Trash collaboration diagram	34
รูปที่ 4-24 Compose sequence diagram	35
รูปที่ 4-25 Compose collaboration diagram	35
รูปที่ 4-26 Delete sequence diagram	36
รูปที่ 4-27 Delete collaboration diagram	36
รูปที่ 4-28 Logout sequence diagram	37
รูปที่ 4-29 Logout collaboration diagram	37
รูปที่ ก-1 แสดงการทำงานของ JSP	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้าที่
ตารางที่ 3-1 ลักษณะพิเศษของภาษาจาวา	15
ตาราง ที่ 3-2 แสดงแท็กต่างๆของ JSP	17
ตารางที่ ก-1 แสดงรูปแบบ page directive	41



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ(Introduction)

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันนี้คอมพิวเตอร์เข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมากและอินเทอร์เน็ตก็มีใช้กันอย่างแพร่หลาย หลายสิ่งหลายอย่างสามารถหาได้จากอินเทอร์เน็ตรวมทั้งการรับ-ส่งจดหมายอิเล็กทรอนิกส์ก็มีขึ้นเพื่ออำนวยความสะดวกให้กับผู้ใช้งาน

1.2 วัตถุประสงค์ของโครงการ

เพื่อสร้างระบบที่สามารถรับ-ส่งจดหมายอิเล็กทรอนิกส์ เพื่อให้บริการสำหรับผู้ที่ใช้บริการการสื่อสารผ่านทางเครือข่ายอินเทอร์เน็ต ได้อันเป็นแนวทางของเทคโนโลยีสารสนเทศอิเล็กทรอนิกส์

1.3 เป้าหมายและขอบเขตของโครงการ

- 1.สามารถทำการส่งข้อมูลจากบุคคลหนึ่งไปยังบุคคลอื่นๆ ได้ โดยระบุที่อยู่(e-mail address) ผู้รับ ผ่านทางระบบการรับ-ส่งข้อมูล โดยใช้เว็บเบราว์เซอร์ผ่านทางอินเทอร์เน็ต
- 2.ในระบบมีเมนูและหน้าจอต่างๆของระบบซึ่งออกแบบมาเพื่อการสื่อความหมายที่เข้าใจง่ายและง่ายต่อการใช้ โดยคำนึงถึงหลักการติดต่อสื่อสารกับผู้ใช้ (GUI)
- 3.ระบบมีการกำหนดพื้นที่เพื่อให้บริการรับ-ส่งจดหมายอิเล็กทรอนิกส์สำหรับสมาชิกแต่ละคน

1.4 ขั้นตอนวิธีการดำเนินงาน

1. ศึกษาระบบปฏิบัติการลินุกซ์ เพื่อสามารถใช้งานและลง โปรแกรมได้อย่างถูกต้องวิธี
2. ศึกษาเทคโนโลยีที่ใช้ ซึ่งก็คือ jsp และ servlet ว่าจะเอาไปใช้ในการทำงานอย่างไร
3. ศึกษา การสร้าง และติดตั้ง DNS Server
4. ศึกษา และติดตั้งระบบ Mail Server โดยเลือกใช้ Sendmail
5. ศึกษาและการติดตั้ง Web Server โดยเลือกใช้ Jakarta Tomcat
6. ทดสอบระบบ DNS Server ให้ใช้งานได้จริง
7. ทดสอบการเชื่อมต่อระหว่าง web server กับ JSP และ JAVA การเซตค่าต่างๆ ให้สามารถทำงานร่วมกันได้
8. ศึกษาการติดตั้งและทดสอบการใช้งานจริงของ mail server เพื่อให้ทำงานบน

web server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.การทดสอบการทำงานของระบบ โดยต้องทำการทดสอบผ่านทางระบบอีเมลโดยจะทำการทดสอบตลอดการจัดทำโปรแกรม

1.5 ประโยชน์ที่ได้รับจากโครงการนี้

1. สามารถรับ-ส่งจดหมายอิเล็กทรอนิกส์ระหว่างบุคคลหนึ่งไปยังอีกบุคคลหนึ่งได้
2. ป้องกันไวรัสและสแปมเมล ในระบบได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การติดตั้งโปรแกรม สำหรับระบบปฏิบัติการลินุกซ์

2.1 โปรแกรมที่ใช้งานในระบบ Mail

1. DNS Server (Bind)
2. Sendmail
3. jdk1.5
4. Jakarta Tomcat

2.2 DNS Server

บริการชื่อ โดเมนหรือระบบชื่อ โดเมน (Domain Name System : DNS) เป็นระบบการตั้งชื่อให้กับ “ทรัพยากรเครือข่าย” ซึ่งที่พบกันบ่อยก็คือการตั้งชื่อ โฮสต์เพื่อใช้แทนค่า IP Address ของโฮสต์ที่จำได้ยากกว่าชื่อทั่วไป ยกตัวอย่างเช่น การเรียกเว็บไซต์ไปที่ <http://www.cat.or.th> ถ้าไม่มีระบบการชื่อ โดเมนเราจะต้องเรียกใช้งานเว็บไซต์ผ่านบราวเซอร์เป็น <http://202.6.107.14> ซึ่งจะเห็นว่าถ้ามีจำนวน โฮสต์มาก ๆ อย่างในเครือข่ายอินเทอร์เน็ตก็จะยากมากต่อการจดจำ

ระบบชื่อ โดเมนจะทำหน้าที่แปลงชื่อ โดเมนหรือชื่อ โฮสต์เป็น IP Address ที่เป็นมาตรฐานในการสื่อสารของระบบเครือข่ายคอมพิวเตอร์ ซึ่งการทำงานแบบนี้จะเรียกว่า Forward Master Zone และในบางครั้งอาจมีความจำเป็นที่จะต้องแปลงค่า IP Address ให้เป็นชื่อ โฮสต์ ระบบชื่อ โดเมนก็มีความสามารถที่จะทำงานในลักษณะดังกล่าวได้ ซึ่งเรียกกันว่า Reverse Master Zone

ซอฟต์แวร์ที่ให้บริการดีเอ็นเอสที่นิยมแพร่หลายกันคือชุดซอฟต์แวร์ BIND (Berkeley Internet Name Domain) ซึ่งเมื่อเอาซอฟต์แวร์ประเภทนี้ไปติดตั้งบนเซิร์ฟเวอร์แล้วก็จะเรียกเซิร์ฟเวอร์ดังกล่าวว่าเป็น Name Server หรือ DNS Server ซึ่ง Name Server จะมีสองแบบคือ Master Name Server กับ Slave Name Server ซึ่งจะต้องอยู่ต่างเครื่องกัน โดย Master Name Server จะอ่านข้อมูลประจำโชนจากแฟ้มในดิสก์และถือว่ามีอำนาจหน้าที่ใน โชนนั้น ส่วน Slave Name Server จะสำเนาข้อมูลมาจาก Master Name Server โดยการถ่ายโอนผ่านเครือข่าย โดยทั่วไป โชนหนึ่ง ๆ จะมี Master Name Server ได้เพียงเครื่องเดียว แต่มี Slave Name Server ได้มากกว่าหนึ่งเครื่อง แต่ในทางปฏิบัติก็มีหลายหน่วยงานที่ใช้ Master Name Server มากกว่าหนึ่ง

วิธีการเซต Name Server สามารถทำได้ 2 วิธีการคือการเซตผ่าน Command Prompt ก็เป็นการเซตผ่านระบบกราฟฟิค Xwindow

```

## named.conf - configuration for bind
#
options {
    directory "/var/named";
};
controls {
    inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
    type hint;
    file "named.ca";
};

zone "localhost" IN {
    type master;
    file "localhost.zone";
    allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "named.local";
    allow-update { none; };
};

zone "jonnetwork.org" {
    type master;
    file "jonnetwork.zone";
};

include "/etc/rndc.key";

```

รูปที่ 2-1 ไฟล์ */etc/named.conf*

```

$TTL 86400
$ORIGIN localhost.
@ 1D IN SOA @ root (
    42 ; serial (d.
    adams)
    3H ; refresh
    15M ; retry
    1W ; expiry
    1D )
;
minimum
    1D IN NS @
    1D IN A 127.0.0.1

```

รูปที่ 2-2 ไฟล์ *localhost.zone*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$TTL 86400
@ IN SOA localhost. root.localhost. (
    1997022700 ; Serial
    28800 ; Refresh
    14400 ; Retry
    3600000 ; Expire
    86400 ) ; Minimum
IN NS localhost.

1 IN PTR localhost.

```

รูปที่ 2-3 ไฟล์ *name.local*

```

$TTL 86400
@ SOA ns.jonnetwork.org. root.jonnetwork.org. (
    1997022704 ; Serial
    28800 ; Refresh
    14400 ; Retry
    3600000 ; Expire
    86400 ) ; Minimum
IN NS cresendo.jonnetwork.org.
IN A 161.246.5.155
IN MX 10 mail.jonnetwork.org.

localhost IN A 127.0.0.1
cresendo IN A 161.246.5.155
ns IN A 161.246.5.155
www IN A 161.246.5.155
mail IN A 161.246.5.155

```

รูปที่ 2-4 ไฟล์ *jonnetwork.zone*

ไฟล์ *nsswitch.conf* (name service switch) เป็นไฟล์ที่ใช้กำหนดลำดับในการค้นหาโดเมน พิจารณาที่ *hosts* โดยลำดับแรกจะอ่านที่ *hosts file* ก่อน แล้วจะไปหาใน *dns*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

...

passwd: files
shadow: files
group: files

hosts: files dns

networks: files
protocols: files
rpc: files
services: files

.....

```

รูปที่ 2-5 ไฟล์ `/etc/nsswitch.conf`

ไฟล์ `resolv.conf`: ในกรณีที่เรากำหนดให้ค้นหาชื่อ โดเมน โดยเริ่มต้นจาก `hosts file` ก่อนเมื่อค้นหาไม่เจอ ระบบจะมาอ่านไฟล์ `resolv.conf` ตรวจสอบดูว่าจะต้องเข้าไปถามเนมเซิร์ฟเวอร์อะไรได้บ้าง โดยจะไปถามเนมเซิร์ฟเวอร์ที่อยู่ในไฟล์จากบนลงล่าง

```

domain jonnetwork.org
nameserver 161.246.5.155
nameserver 161.246.4.3
nameserver 161.246.52.21

```

รูปที่ 2-6 ไฟล์ `/etc/resolv.conf`

```

jonnetwork.org
127.0.0.1 localhost.localdomain cresendo.jonnetwork.org
localhost cresendo

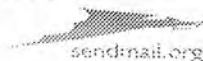
```

รูปที่ 2-7 ไฟล์ `/etc/hosts`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Master Name Server ไม่สามารถให้บริการได้ เครื่องที่เป็น Slave Name Server ก็ต้องทำงานทดแทน ดังที่ได้กล่าวมาแล้วว่า การทำงานของ Slave Name Server จะมีการส่งข้อมูลมาจาก Master Name Server ด้วยการถ่ายโอนผ่านเครือข่าย ฉะนั้นการเซต Slave Zone ก็จะมีขั้นตอนที่ไม่มากนัก แต่เรากำหนดค่าเริ่มต้นที่จำเป็นให้กับ โปรแกรม Domain Name Service แล้ว โปรแกรมก็จะไปส่งข้อมูลต่าง ๆ มาจาก Master Name Server เองทำให้ไม่ต้องเสียเวลาเพิ่มโฮสต์ต่าง ๆ

2.3 Sendmail



เป็นโปรแกรมที่ทำหน้าที่เป็น MTA (Mail Transfer Agent) บนลินุกซ์มีแพ็คเกจที่เกี่ยวข้องอยู่ 4 แพ็คเกจ คือ sendmail , sendmail-cf,m4,imap โดยปกติ โปรแกรม sendmail จะติดตั้งมากับลินุกซ์แต่แรกแล้ว หรือจะตรวจสอบด้วยคำสั่ง `rpm -qi sendmail` ถ้าติดตั้งแล้วจะแสดงรายละเอียดโปรแกรม เช่น version , release package เป็นต้น

การติดตั้งไฟล์ `sendmail-cf` ซึ่งเป็นไฟล์ configuration ของโปรแกรม sendmail สำหรับกำหนดหน้าที่การทำงานต่างๆ เช่น กำหนดให้เป็น Mail Server ของเครือข่าย การเซตอัพโปรโตคอลที่ใช้รับ-ส่งเมล การเก็บเมลเอาไว้ที่ไหน การสร้างกฎ เป็นต้น ติดตั้งด้วยคำสั่ง `rpm -ivh sendmail-cf-8.12.xx.xx.i386.rpm` หลังจากติดตั้งแล้ว ใช้คำสั่ง `vi /etc/mail/sendmail.cf` เพื่อเข้าไปตรวจรายละเอียดไฟล์

```
#####
# local info #
#####
# my LDAP cluster
# need to set this before any LDAP lookups are done (including classes)
#D{sendmailMTACluster}Sm

Cwlocalhost
# file containing names of hosts for which we receive email
Fw/etc/mail/local-host-names // ให้เข้าไปเรียกใช้ไฟล์ local-host-names

# my official domain name
# ... define this only if sendmail cannot automatically determine your domain
Djjonnetwork.org // เพื่อใช้แสดงชื่อ โดเมนเวลาส่งเมลออกไปภายนอก

# host/domain names ending with a token in class P are canonical
CP.

# "Smart" relay host (may be null)
n๑
```

รูปที่ 2-8 การแสดงค่า macro ในไฟล์ sendmail-cf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# local-host-names - include all aliases for your machine
here.
localhost.localdomain
localhost
jonnetwork.org
```

รูปที่ 2-9 ไฟล์ local-host-names

การแก้ไขไฟล์ `/etc/mail/access` เป็นไฟล์ที่ใช้กำหนดว่าจะให้ เครื่องใด หรือ IP Address ใดสามารถทำ Relay ผ่านเครื่อง Mail Server หรือการห้ามเครื่องใด โดเมนใด IP Address รวมทั้ง E-mail address ไม่ให้ผ่านไปได้

```
localhost.localdomain RELAY
localhost RELAY
127.0.0.1 RELAY
jonnetwork.org RELAY
```

รูปที่ 2-10 ไฟล์ access

หลังจากแก้ไขเพิ่มเคมชื่อเครื่อง โดเมน IP Address รวมทั้ง E-mail address เสร็จแล้วให้พิมพ์คำสั่ง `makemap hash /etc/mail/access < /etc/mail/access` เพื่ออัปเดตข้อมูล

การติดตั้ง แพคเกจ `imap` สามารถตรวจสอบการติดตั้ง โดยใช้คำสั่ง `rpm -q imap` ถ้าติดตั้งแล้วจะแสดงชื่อแพ็คเกจออกมา แต่ถ้ายังไม่ติดตั้ง ให้พิมพ์คำสั่ง `rpm -ivh imap-xx-xx.i386.rpm`

การแก้ไขไฟล์ `/etc/xinetd.d` การทำงานของโปรโตคอล POP3 และ IMAP มีไฟล์ที่เกี่ยวข้อง 2 ไฟล์คือ `/etc/xinetd.d/ipop3` และ `/etc/xinetd.d/imap` ซึ่งปกติจะถูกปิดการทำงานเอาไว้จะต้องเข้าไปเปิดให้ทำงานดังนี้

```
service pop3
{
    socket_type          = stream
    wait                = no
    user                 = root
    server               = /usr/sbin/ipop3d
    log_on_success      += HOST DURATION
    log_on_failure      += HOST
    disable              = no
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 2-11 การแก้ไขไฟล์ ipop3 อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

service imap
{
    socket_type      = stream
    wait             = no
    user             = root
    server           = /usr/sbin/imapd
    log_on_success   += HOST DURATION
    log_on_failure += HOST
    disable          = no
}

```

รูปที่ 2-12 การแก้ไขไฟล์ *imap*

ตรวจสอบพอร์ตบริการที่ใช้เชื่อมต่อกับ POP3(110) และ IMAP(143)

การสแตตเซอร์วิส(Daemon)ของ POP3/IMAP โดยใช้คำสั่ง `/etc/init.d/xinetd start`

```

root@crescendo:~$ /etc/init.d/xinetd restart
Stopping xinetd:
Starting xinetd:
root@crescendo:~$ netstat -na | grep tcp
tcp        0      0 0.0.0.0:32770          0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:110           0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:143           0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:111           0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:32786       0.0.0.0:*              LISTEN
tcp        0      0 161.246.5.155:53     0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:53         0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:631        0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:25           0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:953        0.0.0.0:*              LISTEN
tcp        0      0 161.246.5.155:32785  161.246.5.155:143     ESTABLISHED
tcp        0      0 161.246.5.155:143    161.246.5.155:32785   ESTABLISHED
tcp        1      0 127.0.0.1:32774      127.0.0.1:631        CLOSE_WAIT
tcp        0      0 :::ffff:127.0.0.1:8005  :::*                 LISTEN
tcp        0      0 :::8080              :::*                 LISTEN
tcp        0      0 :::80                 :::*                 LISTEN
tcp        0      0 :::22                 :::*                 LISTEN
tcp        0      0 :::443                :::*                 LISTEN
root@crescendo:~$

```

รูปที่ 2-13 แสดงการเปิดพอร์ต 110และ143 (POP3/IMAP)

การกำหนดขนาดในการรับส่งเมล `sendmail` จะมีไฟล์ `/etc/mail/sendmail.mc`

ที่ใช้สำหรับสร้างมาโครให้กลายเป็น config file ที่แท้จริงคือ `/etc/mail/sendmail.cf` การกำหนดขนาดของเมลโดยค่า Default จะเป็น 2000000 bytes ซึ่งค่านี้สามารถกำหนดได้ในไฟล์ `/etc/mail/sendmail.mc` ตรงค่าที่เขียนว่า

```
define('UUCP_MAILER_MAX', '2000000')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากค่าคอนฟิกของ Mail Server ในส่วนของ SMTP daemon options ได้กำหนดเอาไว้เป็น Port=smtp,Addr=127.0.0.1 คือไม่อนุญาตให้เครื่องอื่นเข้ามาทางพอร์ต SMTP ต้องแก้ไขเป็น Addr=0.0.0.0 โดยพิมพ์คำสั่ง vi /etc/mail/sendmail.mc ซึ่งของเดิมเป็น

```
DaemonPortOptions=Port=smtp,Addr=127.0.0.1,Name=MTA
```

ให้แก้ไขเป็น

```
DaemonPortOptions=Port=smtp,Addr=0.0.0.0,Name=MTA
```

เสร็จแล้วให้ทำการรันมาโครด้วยคำสั่ง

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

(เสร็จแล้วให้รีสตาร์ทเซอร์วิสของ Sendmail ด้วยคำสั่ง service sendmail start)

```
define('ALIAS_FILE', '/etc/aliases')dnl
define('STATUS_FILE', '/var/log/mail/statistics')dnl
define('UUCP_MAILER_MAX', '2000000')dnl
define('confUSERDB_SPEC', '/etc/mail/userdb.db')dnl
define('confPRIVACY_FLAGS', 'authwarnings,novrfy,noexpn,restrictqrun')dnl
define('confAUTH_OPTIONS', 'A')dnl
.....
DAEMON_OPTIONS('Port=smtp,Addr=0.0.0.0, Name=MTA')
```

รูปที่ 2-14 ไฟล์ sendmail.mc

2.4 JDK และ package ต่างๆ

เหตุผลที่ต้องติดตั้ง JDK เพราะว่า tomcat จำเป็นต้องใช้องค์ประกอบหลายอย่างของ JDK เวลาที่ทำงาน เมื่อติดตั้งเสร็จแล้วจะต้องกำหนดค่าแปร Path เพื่อให้คอมไพเลอร์จาวา (javac) สามารถเรียกจากที่ใดก็ได้ นอกจากไดเรกทอรีที่ติดตั้งไว้

สำหรับแพ็คเกจอื่น คือ mail.jar ซึ่งมีคลาส JavaMail API รวมทั้ง providers ด้วย และ activation.jar ซึ่งช่วยในการระบุประเภทข้อมูลของ MIME protocol เพื่อจะได้จัดการกับข้อมูลนั้นได้อย่างถูกต้อง โดยต้องกำหนดค่า CLASSPATH ขึ้นมาที่สองไฟล์นี้ หรือจะนำสองไฟล์นี้ไปไว้ที่ %JAVA_HOME%\jre\lib\ext ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสของ JavaMail API แบ่งได้เป็นสองประเภท

1. *Transport* - กลไกเกี่ยวกับการส่ง e-mail
2. *Store* - กลไกเกี่ยวกับการติดต่อเข้าไปยัง mail server เพื่อนำ e-mail ออกมา

2.5 Jakarta Tomcat

Jakarta Tomcat ทาง apache ยังคงทำการพัฒนา Tomcat ให้มีประสิทธิภาพมากเพื่อใช้เป็นมาตรฐานในอนาคต ดังนั้นถ้าเข้าไปที่เว็บไซต์ในส่วนของดาวโหลด เราจะเห็นตัวเลือกในการดาวโหลดได้คร่าว ๆ เป็นสองแบบ คือ แบบ source files ที่ต้องมาคอมไพล์เอง โดยใช้เครื่องมือช่วยที่เรียกว่า *Ant* ซึ่งเป็นส่วนที่นักพัฒนามอบมาควบคู่กัน กับแบบที่คอมไพล์แล้ว (binary) พร้อมทั้งจะรันซึ่งเหมาะสำหรับคนที่อยากนำ Tomcat มาใช้งานอย่างเดียว และเป็นอย่างที่เราจะใช้

โดยสามารถดาวโหลดได้จาก www.apache.org จากนั้นจึงทำการติดตั้ง

โดยก่อนที่เราจะเริ่มเขียน JSP สิ่งหนึ่งที่เราจะทำความเข้าใจคือ โครงสร้างของไคลเอนต์ต่าง ๆ สำหรับไฟล์ของเราใน Tomcat ในสมัยก่อนเราไม่ค่อยได้ให้ความสนใจกับการจัดเรียงไฟล์ต่าง ๆ ซักเท่าไร ข้อเสียที่เห็นได้ชัดคือการยากในการจัดเก็บ และการยากในการย้ายไฟล์จากเซิร์ฟเวอร์หนึ่ง ไปยังอีกเซิร์ฟเวอร์หนึ่ง เพราะโดยทั่วไปแต่ละ webserver จะมีลักษณะการจัดเก็บไฟล์ไม่เหมือนกัน แต่เมื่อ Java Servlet Specification เวอร์ชัน 2.2 ออกมา เซิร์ฟเวอร์ที่ใช้รัน Servlet v2.2 จะถูกบังคับให้มีต้องการสนับสนุนการจัดเก็บไฟล์แบบหนึ่ง ที่เรียกว่า Web Application คือกลุ่มของไคลเอนต์และไฟล์ที่อาจจะประกอบด้วย html, jsp, servlet, javabean และอื่น ๆ ซึ่งอยู่รวมกันในลักษณะของระบบไฟล์ (file system) หรือถูกอัดอยู่ในไฟล์เดียวกัน โดยจะเรียกว่า Web Archive (.war) ไฟล์ ซึ่งมีประโยชน์ในการโยกย้ายและติดตั้งจากเซิร์ฟเวอร์หนึ่ง ไปยังอีกเซิร์ฟเวอร์หนึ่ง

ตัวอย่างไคลเอนต์ต่าง ๆ ใน Tomcat

1) root directory หรือส่วนบนสุด : ไฟล์ *.html, *.jsp มักจะถูกเก็บไว้ส่วนบนสุดของ web application แต่อาจจะมีไคลเอนต์ย่อยลงมาจาก root เพื่อเก็บ ไฟล์ images ก็ได้

2) WEB-INF/web.xml : ไฟล์ web.xml เป็นส่วนที่ใช้ในการบรรยายส่วนประกอบต่าง ๆ (Web Application Deployment Descriptor) รวมไปถึงการป้อนค่าเริ่มต้นให้กับ โปรแกรมต่าง ๆ ที่บรรจุอยู่ใน web application นี้ ยกตัวอย่างเช่น เราสามารถป้อนค่าเริ่มต้นให้กับ Servlet ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใส่ค่าเข้าไปในXML Tagตรงส่วนที่เรียกว่า <init-param> เป็นคั่น (หาอ่านเพิ่มเติมได้จาก JavaServlet Specification v2.2, <http://java.sun.com/products/servlet/>)

3) WEB-INF/classes/* : เป็นส่วนที่ใช้ในการเก็บไฟล์ *.class ทั้งหมดที่ใช้ใน web application ซึ่งอาจจะเป็นทั้ง Servlet หรือ ไฟล์ class ธรรมดาที่ไม่ได้อยู่ในรูปของ .jar ก็ได้ ยกตัวอย่างเช่น ถ้าไฟล์ของเราอยู่ใน package ชื่อ com.mycompany.mypackage.MyServlet ไฟล์นี้ก็จะถูกเก็บอยู่ในไดเรกทอรี WEB-INF/classes/com/mycompany/mypackage/MyServlet.class (เกี่ยวกับ package สามารถอ่านเพิ่มเติมได้จาก JAVA PACKAGE)

4) WEB-INF/lib/*.jar : ไดเรกทอรีนี้ใช้เก็บไฟล์ .jar ที่เราต้องการใช้สำหรับ web application ซึ่งอาจจะเป็นส่วนที่ได้มาจาก third party หรือเป็นพวก Database driver ก็ได้ ประโยชน์อย่างหนึ่งที่เห็นได้ในการเก็บไลบรารีต่างๆ ไว้ที่ไดเรกทอรีนี้คือ การง่ายต่อการติดตั้ง เหตุผลคือ หลังจาก web application ของเราถูกใส่เข้าไปใน server แล้ว ตัว server จะโหลดไฟล์ที่อยู่ในไดเรกทอรีนี้โดยอัตโนมัติ เราจึงไม่ต้องทำการเซต CLASSPATH ให้ยุ่งยาก



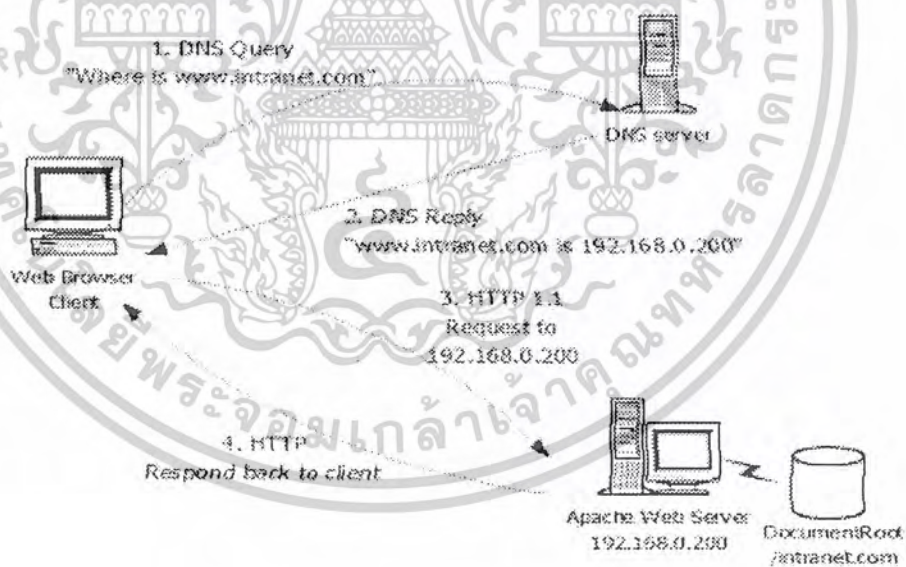
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ทฤษฎีและหลักการ

3.1 DNS

ในด้านการเรียกเข้าชมเว็บไซต์จากผู้ใช้ทั่วไป ระบบของเราจำเป็นต้องอาศัยระบบ Domain Name Service หรือ DNS เพื่อช่วยให้ผู้ชมสามารถเข้าถึงเว็บไซต์ของเราได้ด้วยการเรียกด้วยชื่อของเว็บไซต์ แทนที่จะเรียกเข้ามาด้วยหมายเลขไอพี หากเป็นการสร้างเว็บไซต์ขึ้นเพื่อใช้งานภายในขององค์กรก็จำเป็นต้องจัดตั้ง DNS Server ขึ้นเพื่อช่วยแปลงชื่อโฮสต์ (เช่น www.intranet.com) ให้เป็นหมายเลขไอพีของโฮสต์ที่เป็นเว็บไซต์ของเรา ซึ่งอาจจะใช้ DNS Server ที่มีอยู่แล้วในองค์กรก็ได้ แต่ถ้ายังไม่เคยมี DNS Server มาก่อนก็สามารถคอนฟิกโปรแกรม BIND ให้ทำหน้าที่เป็น DNS Server ก็ได้



รูปที่ 3-1 การทำงานร่วมกับระหว่าง DNS กับ Web Server

จากรูป แสดงการทำงานของ DNS Server โดย Client ได้ส่ง DNS Query ไปสอบถาม DNS Server ว่า intranet.com นั้นมี IP เป็นอะไร แล้ว DNS Server จะส่งค่า IP ของโดเมนนั้นกลับมายัง Client เพื่อจะส่ง HTTP Request ไปยัง Web Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจดทะเบียนโดเมนให้เครื่องServerของระบบในที่นี้ได้ลงทะเบียนโดเมนชื่อ jonnetwork.org และกำหนดชื่อ Mail Server เป็น mail.jonnetwork.org สามารถทดสอบได้ด้วยคำสั่ง nslookup -sil และพิมพ์คำสั่ง set query=any แล้วพิมพ์ชื่อโดเมนลงไปคือ jonnetwork.org เพื่อให้แสดงรายละเอียดของเซิร์ฟเวอร์และโดเมนอย่างละเอียด ดังรูป

```

root@crescendo root]# nslookup -sil
> set query=any
jonnetwork.org
Server:          161.246.5.155
Address:         161.246.5.155#53

jonnetwork.org
  origin = ns.jonnetwork.org
  mail addr = root.jonnetwork.org
  serial = 1997022704
  Refresh = 36800
  Retry = 14400
  expire = 3600000
  minimum = 86400
jonnetwork.org name server = crescendo.jonnetwork.org.
Name:   jonnetwork.org
Address: 161.246.5.155
jonnetwork.org mail exchanger = 10 mail.jonnetwork.org.

```

รูปที่ 3-2 การใช้ nslookup ตรวจสอบค่า DNS ของ Mail Server

```

root@crescendo root]# dig jonnetwork.org
;<<> DiG 9.2.3 <<> jonnetwork.org
;; global options: printed
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 43357
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
; jonnetwork.org. IN A
;; ANSWER SECTION:
jonnetwork.org. 86400 IN A 161.246.5.155
;; AUTHORITY SECTION:
jonnetwork.org. 86400 IN NS crescendo.jonnetwork.org.
;; ADDITIONAL SECTION:
crescendo.jonnetwork.org. 86400 IN A 161.246.5.155
;; Query time: 2 msec
;; SERVER: 161.246.5.155#53(161.246.5.155)
;; WHEN: Mon Mar 14 16:28:30 2005
;; MSG SIZE rcvd: 88

root@crescendo root]#

```

รูปที่ 3-3 ตัวอย่างการใช้โปรแกรม dig

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ภาษาจาวา และการทำงานของ JSP

ภาษาจาวา ถูกพัฒนามาจากบริษัทซัน(Sun Microsystems)ซึ่งจัดให้เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่ใช้หลักการออกแบบตัวภาษาด้วยวิธีเชิงวัตถุและตัวภาษาถูกใช้เป็นเครื่องมือสำหรับพัฒนาโปรแกรมด้วยแนวคิดเชิงวัตถุ โดยตัวภาษามีลักษณะพิเศษดังนี้

Portability	สามารถในการใช้งานในสภาพแวดล้อมที่แตกต่าง โดยไม่ต้องมีการปรับแต่ง
Simple	ความง่ายในการเขียน โปรแกรม
Robust	ความคงสภาพในการทำงานมีโอกาสเกิดข้อผิดพลาดที่ไม่พึงประสงค์ได้น้อย
Secure	การรองรับมาตรฐานความปลอดภัยในการใช้งานรูปแบบต่างๆ
Distributed	มีความสามารถในการประมวลผลแบบกระจาย
Object-Oriented	มีหลักการของแนวคิดเชิงวัตถุ ในการสร้างโปรแกรม

ตารางที่ 3-1 ลักษณะพิเศษของภาษาจาวา

ข้อดีของการเขียน โปรแกรมภาษาจาวา การเขียนโปรแกรมภาษาจาวาจะมีข้อดีในเรื่องต่อไปนี้

1 ภาษาจาวา สามารถเรียนรู้ง่ายสำหรับผู้กำลังเริ่มศึกษาภาษาจาวาคือ การที่รู้สึกว่าการเขียน โปรแกรมภาษาจาวาจะเป็นเรื่องยุ่งยากและซับซ้อนแต่ในความเป็นจริงแล้วภาษาจาวาเป็นโปรแกรมภาษาที่ออกแบบมาเพื่อให้เขียน ได้ง่ายกะทัดรัด แต่แฝงด้วยประสิทธิภาพ

2. ภาษาจาวา ใช้ได้กับคอมพิวเตอร์ทุกระบบ ข้อดีที่ทำให้ภาษาจาวาหรือภาษาอื่นคือ เมื่อพัฒนาโปรแกรมด้วยภาษาจาวาจะสามารถนำไปใช้งานบนระบบปฏิบัติการของคอมพิวเตอร์แบบใดๆก็ได้โดยไม่ต้องทำการเปลี่ยนแปลงใดๆทั้งสิ้นข้อดีนี้มีจุดประสงค์ในลักษณะที่เมื่อคุณต้องการ ให้โปรแกรมที่สร้างขึ้นมาสามารถใช้ได้กับทุกระบบปฏิบัติการ

3.ภาษาจาวาสามารถสร้างโปรแกรมบนอินเทอร์เน็ตได้ เนื่องจากโปรแกรมที่เขียนด้วยภาษาจาวาเป็น โปรแกรมที่ทำงานโดยไม่ขึ้นอยู่กับระบบปฏิบัติการ จึงเป็น โปรแกรมที่เหมาะสมอย่างยิ่งในการนำมาใช้งานบนอินเทอร์เน็ต เนื่องจากอินเทอร์เน็ตเป็นระบบเครือข่ายที่ไม่สามารถระบุเจาะจงได้อย่างแน่ชัดว่าผู้ใช้งานจะใช้อินเทอร์เน็ตนี้อยู่บนระบบปฏิบัติการใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ JSP

JSP เป็นเทคโนโลยีที่เป็นผลรวม ของการนำเอาหลักการสร้างเว็บแบบ static ด้วย HTML มารวมกับการสร้างเว็บแบบไดนามิก ก่อให้เกิดการแสดงผลเว็บ ที่มีการเปลี่ยนแปลงข้อมูลได้ตามการใช้งานของผู้ใช้ เห็นได้ว่าหากพูดถึงหลักการนำเสนอเว็บแล้ว เป็นหลักการที่ทำให้เว็บมีการเปลี่ยนแปลงข้อมูล เช่นเดียวกับ ที่ใช้โปรแกรมจําพวก CGI ทั่วไป แต่ในความหมายของ JSP แล้ว เป็นขั้นตอนและวิธีการสร้างที่ทําร่วมกับคำสั่ง HTML และเก็บบันทึกไว้ในไฟล์เดียวกัน แต่ในขณะที่ CGI เป็นโปรแกรมที่ผลิต HTML ให้เมื่อถูกสั่งให้ทํางาน

ถ้าดูผืน ๆ จากทาง browser การเรียกไฟล์ JSP คงคล้ายๆ กับการเรียกไฟล์ HTML ธรรมดาๆ แต่ความจริงแล้ว หลังจาก request จาก client ส่งมาถึงเซิร์ฟเวอร์, JSP Container จะทำการแปลงไฟล์ JSP ให้กลายเป็น Servlet source ไฟล์ (ในกรณีที่ JSP ไฟล์ดังกล่าวถูกเรียกเป็นครั้งแรก)ซึ่ง ไฟล์ Servlet source ที่ได้จะถูกคอมไพล์เป็น .class เพื่อใช้ในการประมวลผล request ของ client แล้วส่งกลับไปให้เซิร์ฟเวอร์ในรูปแบบของ outputStream ซึ่งถูกส่งไปที่ client ในท้ายสุด หลังจากนั้น ถ้า JSP ไฟล์ดังกล่าวถูกเรียกอีก การแปลงไฟล์หรือคอมไพล์จะไม่เกิดขึ้น เพราะ JSP Container จะใช้ไฟล์ .class ที่เก็บไว้แล้วประมวลผลแทน ข้อสังเกตคือ การเรียกไฟล์ JSP ครั้งแรกจะรู้สึกว่าช้า แต่ครั้งถัดไป ๆ จะเห็นได้ว่าเร็วขึ้นมาก เพราะได้มีการลดขั้นตอนต่าง ๆ ที่เสียเวลาไป ใน JSP Specification 1.1 ยังมีการแนะนำให้ JSP Engine รองรับการโหลดไฟล์ JSP ขึ้นมาคอมไพล์ใหม่แบบอัตโนมัติ (Reloading) ซึ่งมักจะเกิดขึ้น ในกรณีที่ JSP ไฟล์ ได้มีแก้ไข แล้วจะต้องมีการคอมไพล์ไฟล์ Servlet ที่กำลังใช้งานอยู่ใหม่



รูปที่ 3-4 การแปลงและคอมไพล์ JSP ไฟล์

จากข้อความที่อธิบายข้างต้น เราสามารถแบ่งระยะเวลาของไฟล์ JSP หนึ่ง ๆ ออกเป็นสองช่วงคือ

1) Translation Time คือช่วงเวลาที่ JSP ไฟล์ถูกแปลงให้กลายเป็น Servlet ไฟล์และถูกคอมไพล์ให้กลายเป็น .class ไฟล์ ซึ่งจะเกิดก่อนการรับ request จาก client ครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Client Request Time คือช่วงเวลาที่ .class ของ JSP ไฟล์ ทำการรับ request จากแต่ละ client แล้วทำการประมวลผล

คำสั่ง JSP	รูปแบบการเขียน	ความหมาย
Directive	<%@ %>	ใช้กำหนดลักษณะการทำงานของหน้าเว็บหน้านั้น
Declaration	<! %>	สำหรับกำหนดตัวแปร และเมธอดเพื่อใช้งาน
Expression	<%= %>	ใช้แสดงผลลัพธ์ให้ปรากฏบนหน้าเว็บ
Scriptlet	<% %>	ใช้แทรกคำสั่ง เพื่อให้เกิดการทำงาน ปกติคือภาษาจาวา
Comment	<%- %>	ใช้แทรกหมายเหตุ

ตารางที่ 3-2 แสดงเท็กต่างๆของ JSP

3.3 หลักการทำงานของ E-mail

จะมี โปรโตคอลที่ช่วยทำงานดังนี้

- POP3 (Post Office Protocol) ซึ่งในปัจจุบันเป็น protocol มาตรฐานที่ใช้สำหรับรับ-ส่ง จดหมายอิเล็กทรอนิกส์ ในปัจจุบันนี้
- IMAP (Internet Message Access Protocol)
- SMTP (Simple Mail Transfer Protocol)
- MIME (Multipurpose Internet Mail Extension)

วิธีการทำงานของ POP3

POP3 = Post Office Protocol version 3

POP ถูกออกแบบสำหรับการเข้าถึงแบบ offline คือ จดหมายอยู่ใน server และผู้ใช้สามารถใช้โปรแกรมที่สนับสนุน POP ในการเข้าถึงจดหมายจากระยะไกล การจัดการใดๆ กับจดหมายจะเป็นการจัดการ ในเครื่องของผู้ใช้เท่านั้น ถึงแม้ว่าข้อจำกัดของการเข้าถึงแบบ offline จะทำให้เกิดความคิดที่จะทำให้ POP สามารถใช้งานในแบบ online หรือ แบบ disconnected ได้ แต่ POP ขาดคุณสมบัติที่จำเป็นบางอย่าง ส่วนการเข้าถึงแบบเสมือน online (pseudo-online) จดหมายจะไม่ถูกลบออกจาก server แต่ก็ไม่ใช้การเข้าถึงแบบ online ที่แท้จริง เพราะขาดโปรโตคอลในการเข้าถึงระบบ ไฟล์ในเครื่อง server ในการที่จะเข้าถึงหรือเปลี่ยนแปลง folder หรือสถานะต่างๆ ของจดหมาย ตัวอย่างเช่น การตั้งค่าสถานะของจดหมาย เช่น ได้ถูกอ่านแล้ว จะถูกตั้งค่าเพียงแต่ในเครื่อง โคลเอนท์ที่ใช้เท่านั้น แต่สถานะของจดหมายที่แท้จริงบนเซิร์ฟเวอร์ยังคงไม่เปลี่ยนแปลง หรือการจัดเก็บจดหมายลง folder จะถูกจัดเก็บลงในเครื่อง โคลเอนท์เท่านั้น ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จริงควรจะต้องเก็บไว้บนเซิร์ฟเวอร์ เพราะหากว่ามีการใช้ไคลเอนต์เครื่องอื่น ก็ยังสามารถเห็น folder นั้นๆ ได้

วิธีการทำงานของ IMAP

มีความสามารถในการเข้าถึงทั้งแบบ offline และแบบ online โดยในแบบ online จดหมายจะไม่ถูกดึงมา แต่จะเป็นแบบ โค้ดคอบกับ server นั่นคือผู้ใช้สามารถดึงเฉพาะหัวข้อจดหมาย บางส่วนของจดหมาย หรือค้นหาจดหมายที่ตรงความต้องการ โดยจดหมายที่ถูกเก็บไว้บน server และสามารถตั้งค่าสถานะของจดหมายต่างๆ เช่น ถูกลบไปแล้ว ตอบไปแล้ว และจะยังอยู่ใน server จนกว่าผู้ใช้จะสั่งลบ อธิบายง่ายๆ คือ IMAP ถูกออกแบบให้การเข้าจดหมายจากระยะไกล เหมือนกับการเข้าถึงจดหมายจากภายในเครื่องของผู้ใช้เอง ซึ่งขึ้นอยู่กับ โปรแกรมที่ใช้ว่าจะเก็บจดหมายไว้ในเครื่องของผู้ใช้เอง หรือเก็บไว้ใน server หรือให้ผู้ใช้เลือกเองอย่างใดอย่างหนึ่ง

สรุปความแตกต่างระหว่าง offline และ online

1. ความแตกต่างในแง่การใช้งาน

offline: เหมาะกับการเข้าถึง โดยใช้คอมพิวเตอร์เครื่องเดียว และไม่ต้องการความสามารถพิเศษ

อื่นๆ

online : เหมาะกับการเข้าถึงแบบคอน โค้ดกับ server และมีความสามารถในการเข้าถึง mailbox

2. ข้อดีของแบบ offline ใช้เวลาในการคิดค้นน้อย ใช้ทรัพยากรของระบบน้อย

3. ข้อดีของแบบ online

สามารถเข้าถึงจดหมาย จากคอมพิวเตอร์เครื่องใดก็ได้ ในเวลาใดๆ เหมาะสำหรับเครื่อง ที่มีเนื้อที่ดิสก์น้อย เช่น เครื่องคอมพิวเตอร์ในห้องปฏิบัติการ ไม่ขึ้นกับ platform ในการเข้าถึง mailbox หลายๆ อัน ผู้ใช้หลายคนสามารถทำการ ใช้ mailbox เดียวกันได้ในเวลาเดียวกัน จุดสำคัญก็คือ ในแบบ online สามารถเข้าถึงได้จากคอมพิวเตอร์หลายๆเครื่องในเวลาใดๆ โดยไม่ขึ้นกับ โปรโตคอลในการเข้าถึงระบบไฟล์ในเครื่อง server (ซึ่งอาจไม่มีในทุก platform และอาจมีปัญหาเกี่ยวกับประสิทธิภาพ และ file locking) ถึงแม้ว่าความสามารถนี้ไม่มีความจำเป็นสำหรับผู้ที่ใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียว ในการเข้าถึงจดหมาย แต่จะจำเป็นมากสำหรับผู้ที่ใช้เครื่องคอมพิวเตอร์หลายๆ เครื่อง

ความสามารถในส่วนที่มีทั้งใน POP และ IMAP

สนับสนุนการเข้าถึงแบบ offline จดหมายใหม่จะถูกส่งไปเก็บไว้ยังเครื่อง server รอผู้ใช้คิดต่อเข้าไปตรวจเช็ค สามารถเข้าถึงจดหมายได้จาก platform หลากหลายรูปแบบ สามารถเข้าถึงได้จากทุกๆ ที่ในเครือข่าย เป็น โปรโตคอลแบบเปิด ซึ่งเป็นไปตาม Internet RFCs มีเครื่องมือ (Implementations) ต่างๆ (รวมทั้งซอร์สโค้ด) แจกฟรี มีโปรแกรมสำหรับการเข้าถึงจดหมายทั้งใน PC, Mac และ UNIX มีเครื่องมือ (Implementations) ต่างๆ ขยาย

เอกสารนี้เป็นเอกสารที่สงวนสิทธิ์ ไม่ต้องการ SMTP mail gateways ถูกใช้ในการเข้าถึงจดหมายเท่านั้น ส่วนการส่งจะใช้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

protocol SMTP สนับสนุน persistent message IDs (สำหรับการเข้าถึงแบบ disconnected) ข้อดีของโปรโตคอล POP ไม่ซับซ้อน จึงง่ายในการ Implement มีโปรแกรมสนับสนุนจำนวนมาก

ข้อดีของโปรโตคอล IMAP สามารถจัดการกับสถานะของจดหมาย (status flags) เช่น อ่านแล้ว,ลบแล้ว,ตอบแล้ว หรือ สถานะใดๆที่ผู้กำหนดเอง สามารถเพิ่มจดหมายลงใน folder ที่ต้องการ สามารถเข้าถึง และจัดการกับ mailbox ได้หลายอัน รวมทั้งสามารถเปลี่ยนชื่อ,สร้าง,ลบ,แสดงรายชื่อ และเข้าถึง folder ต่างๆ นอกเหนือจาก folder จดหมายปกติ (Inbox) ซึ่ง folder เหล่านี้อาจจะไม่ได้อยู่บน server เดียวกันก็ได้ โดยเราสามารถเรียกดูได้พร้อมๆกัน หรือทำการเคลื่อนย้ายจาก folder หนึ่งไปยังอีก folder หนึ่ง ผู้ใช้หลายคนสามารถทำการเข้าถึง mailbox เดียวกันพร้อมกัน โดยเมื่อผู้ใดทำการแก้ไขอะไรไป ผู้ใช้คนอื่นๆ ก็จะเห็นการแก้ไขนั้นๆ ในทันที สามารถเข้าถึงข้อมูลที่ไม่จำกัดแค่ e-mail แต่สามารถเข้าถึงข้อมูลชนิดอื่นด้วย เช่น NetNews, Documents สนับสนุนการเข้าถึงแบบ offline ในกรณีที่เชื่อมต่อผ่านทาง dialup ที่มีราคาต่อนาทีสูง โดยไม่ต้องการความสามารถพิเศษอื่นๆ หรือในกรณีที่เครื่อง server มีความสามารถต่ำ แต่ไม่ใช่ทุกโปรแกรมที่สนับสนุน IMAP สามารถทำได้ แต่โดยโปรโตคอล IMAP แล้วสามารถทำได้

มีโปรโตคอลในการจัดเก็บข้อมูลส่วนตัวของผู้ใช้แต่ละคนลงในเซิร์ฟเวอร์ เช่น Internet Message Support Protocol (IMSP) ซึ่งทำให้ผู้ใช้จาก platform ใดๆ สามารถเข้าถึงข้อมูลส่วนบุคคลของตนเอง เช่น สมุดรายชื่อที่อยู่ (address book) ทางเลือกใหม่ที่ทรงประสิทธิภาพก็คือ Application Configuration Access Protocol (ACAP) โปรโตคอลที่ไม่เพียงแต่ช่วยให้สร้าง หรือเก็บออบเจกต์ และ address-book information ที่เลือกไว้แล้วเท่านั้น แต่ยังสามารถสร้าง โพรซีเจอร์เพื่อจัดการกับผู้ใช้ที่เกี่ยวข้องที่ถูกร่วมกัน เช่น spelling check เป็นต้น

ปรับปรุงประสิทธิภาพในการเข้าถึง โดยเมื่อทำการติดต่อผ่านทางสายสื่อสารความเร็วต่ำ โดย IMAP มีความสามารถในการเลือกดึงบางส่วนของจดหมายที่ต้องการ หรือให้ server ค้นหาคำที่ต้องการให้ โดยมันจะเป็นประโยชน์อย่างมาก หากการติดต่อผ่านสายสื่อสารที่ช้ามาก ในการเลือกที่จะดึงเฉพาะบางส่วนของจดหมาย เช่น หัวข้อจดหมาย (header) โดยเฉพาะหากจดหมายนั้นเป็นจดหมายชนิด MIME (Multipurpose Internet Mail Extensions ซึ่งเป็นมาตรฐานในการส่งแฟ้มข้อมูลติดไปกับจดหมายตาม RFC-822 โดยโปรโตคอล SMTP) ซึ่งหากมีการส่งแฟ้มขนาดใหญ่ติดมากับจดหมายด้วย โดยมีส่วนของจดหมายเพียงไม่กี่บรรทัดจะทำให้เสียเวลาในการดึงข้อมูลที่ยังไม่ต้องการ เป็นอย่างมาก โดยเฉพาะเมื่อทำการติดต่อผ่านทางสายสื่อสารความเร็วต่ำ หรือค่าใช้จ่ายในการติดต่อมีราคาแพง และนี่เป็นข้อดีที่แสดงให้เห็นว่า IMAP เหนือกว่า POP

ข้อดีของโปรโตคอล IMAP

1. IMAP สามารถให้บริการในรูปแบบ remote ได้ดีกว่า (คือการควบคุมการใช้แม่เหล็กจากเครื่องเราไปยัง Server) เช่น อ่านแม่เหล็กแบบออนไลน์ แยกแม่เหล็กกับส่วนประกอบเอกสาร (Attachment) ออกจากกันได้ เราสามารถเลือกดาวน์โหลดจดหมายมาเก็บไว้เครื่องเรา โดยที่ส่วนประกอบเอกสารไว้ที่ Server เพื่อดาวน์โหลดในภายหลังหรือยามว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. IMAP สนับสนุนโฟลเดอร์แบบลำดับชั้นและสามารถแบ่งโฟลเดอร์ให้ใช้งานร่วมกันได้ (folder hierarchies and folder sharing) ในขณะที่ POP ไม่สามารถทำได้

3. IMAP อนุญาตให้ทำการค้นหาจดหมายหรือบางส่วนของจดหมาย รวมทั้งเลือกจดหมายที่ต้องการจะนำมาเก็บ ไว้ที่เครื่องเราได้ (การค้นหานี้จะทำโดย server ไม่ใช่ Client)

ข้อเสียของโปรโตคอล IMAP

โปรโตคอลมีความซับซ้อน และยากในการ Implement มีซอฟต์แวร์ที่สนับสนุนน้อยกว่า POP IMAP เหนือกว่า POP ใน 3 ส่วนหลักๆ คือ

มีคำสั่งในการจัดการข้อมูลจำนวนมาก มีความสามารถในการจัดการ folder อื่นๆ นอกเหนือจาก inbox มีจุดเด่นในการเพิ่มประสิทธิภาพของแบบ online โดยเฉพาะกับจดหมายที่เป็น MIME และเพราะว่าขณะนี้มีการแจก development libraries ของ IMAP ฟรี ดังนั้นความซับซ้อนของมันคงไม่มีผลต่อความนิยมใช้ที่จะเพิ่มมากขึ้นในอนาคต โดยเน็ตสเคปวางแผนที่จะรวม IMAP เข้าไว้ในเมลเซิร์ฟเวอร์รุ่นต่อไปของตน ซึ่งน่าจะออกมาได้ในปีนี้ ยิ่งไปกว่านั้น SunSoft ก็มี IMAP Server และ Client ขณะที่ยังมี IMAP Client ที่ชื่อว่า Embla ของ ICL และ ICL/TeamWare ที่ให้ Internet Messaging Server ที่สนับสนุน POP และ IMAP ส่วนผลิตภัณฑ์อื่นๆ ที่ร่วมขบวนของ IMAP ก็ได้แก่ Control Data Mail Hub server, NetManage Z-Mail Pro และ messaging server ที่มาจาก Software.com

วิธีการทำงานของ SMTP

เป็นโปรโตคอลในการรับส่งอีเมลล์ ใช้ติดต่อ Mail Server โดยโปรโตคอล SMTP จะทำงานร่วมกับโปรโตคอล TCP/IP ทางพอร์ต 25 เมื่อทาง user agent ส่งเมลล์มาที่ MTA จะมีการเรียกใช้โปรโตคอล SMTP ในการส่งเมลล์ไปยัง MTA ตัวอื่นๆ จนถึงปลายทาง อีเมลล์ที่ส่งจากต้นทางอาจจะไม่ได้ส่งตรงไปยัง MTA ปลายทางเลย แต่มีการส่งผ่านเครื่อง Relay Host ซึ่งทำให้ต้องผ่าน MTAs หลายตัวเช่นกัน ทำให้เกิดกระบวนการเก็บแล้วส่งต่อ (store and forward) แต่เนื่องจากโปรโตคอล SMTP ถูกออกแบบมาใช้ส่งข่าวสารสั้นๆ จึงทำให้ส่งได้แค่ข้อความรหัส ASCII เท่านั้น ไม่สามารถที่จะส่ง ไฟล์รูปภาพ เพลง ได้ แต่มีการสร้างโปรโตคอล MIME เพื่อแก้ปัญหาเหล่านี้แล้ว

วิธีการทำงานของ MIME

เนื่องจาก E-Mail สมัยแรกๆ ที่เริ่มต้นในระบบอินเทอร์เน็ตจะมีแค่แค่เพียงเครื่องมือในการส่งข้อความสั้น โดยที่คุณไม่สามารถที่จะแนบเอกสารหรือรูปภาพที่คุณชอบส่งไปได้ จนกระทั่งได้มีการพัฒนา กำหนด Protocol ใหม่ที่ชื่อว่า MIME ซึ่งเป็นมาตรฐานในการเข้ารหัสแฟ้มข้อมูลหลายชนิดไปพร้อมกับ E-Mail ผ่านอินเทอร์เน็ตซึ่งในปัจจุบันนี้ไม่มีไฟล์ประเภทไหนที่ MIME

ไม่รู้จักรหัสจึงสามารถส่งไฟล์ทุกประเภทไปพร้อมกับ E-mail ได้ โดยมีวิธีการคือแปลงไฟล์รูป

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพ เสียง วิดีโอ ซึ่งอยู่ในรูปแบบ Binary ให้มาอยู่ในรูปแบบตัวอักษร MIME เป็นตัวมาตรฐานที่กำหนดขึ้นเพื่อรองรับจุดประสงค์ที่หลากหลายจากการใช้งาน internet Mail ทั้งนี้เพื่อขยายประโยชน์ใช้สอยของอีเมลได้มากขึ้น เพิ่มข้อมูลมาตรฐาน MIME สามารถใช้ร่วมกับการเก็บไฟล์ของส่งผ่าน ไปทางมาตรฐาน SMTP และ UUCP รวมถึง BitNet X.400 SNADS PROFS และยังมีความสามารถในการแลกเปลี่ยนข้อมูลบนระบบปฏิบัติการ ที่ต่างกันแต่ชนิดของซอฟต์แวร์ที่ใช้ต่างกันได้อย่างน่าอัศจรรย์

สรุป

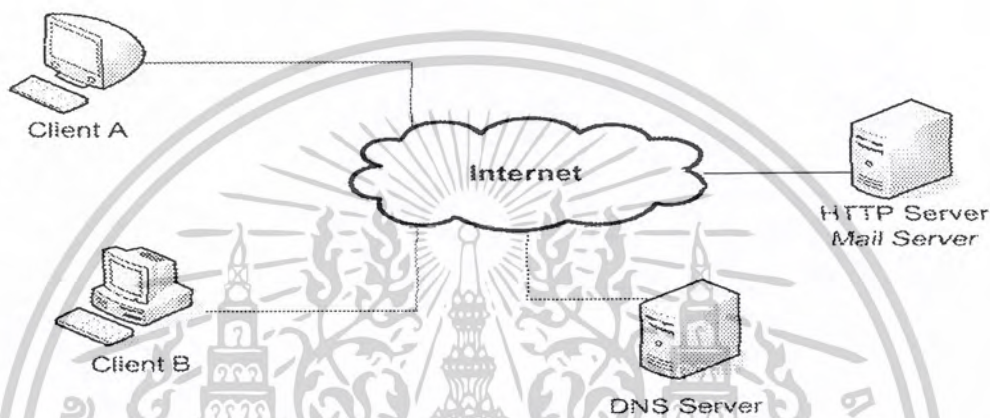
ถึงแม้จะมี Protocol มากมายที่ใช้สำหรับอินเทอร์เน็ตเมลซึ่งแต่ละอันก็มีทั้งข้อดีและข้อเสียของตัวเอง ซึ่งโดยทั่วไปก็จะใช้ POP3 ร่วมกับ SMTP โดยจะใช้ SMTP ในการส่งเมลออกไปยังปลายทางและใช้ POP ในการรับ เก็บจดหมาย E-mail เป็นมาตรฐานในการใช้ E-mail ในปัจจุบัน ซึ่งการใช้งานนี้ก็สามารถใช้ได้อย่างมีประสิทธิภาพดีแล้วในปัจจุบัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบโปรแกรม



รูปที่ 4-1 ภาพรวมของระบบ

4.1 รายละเอียดของโปรแกรม

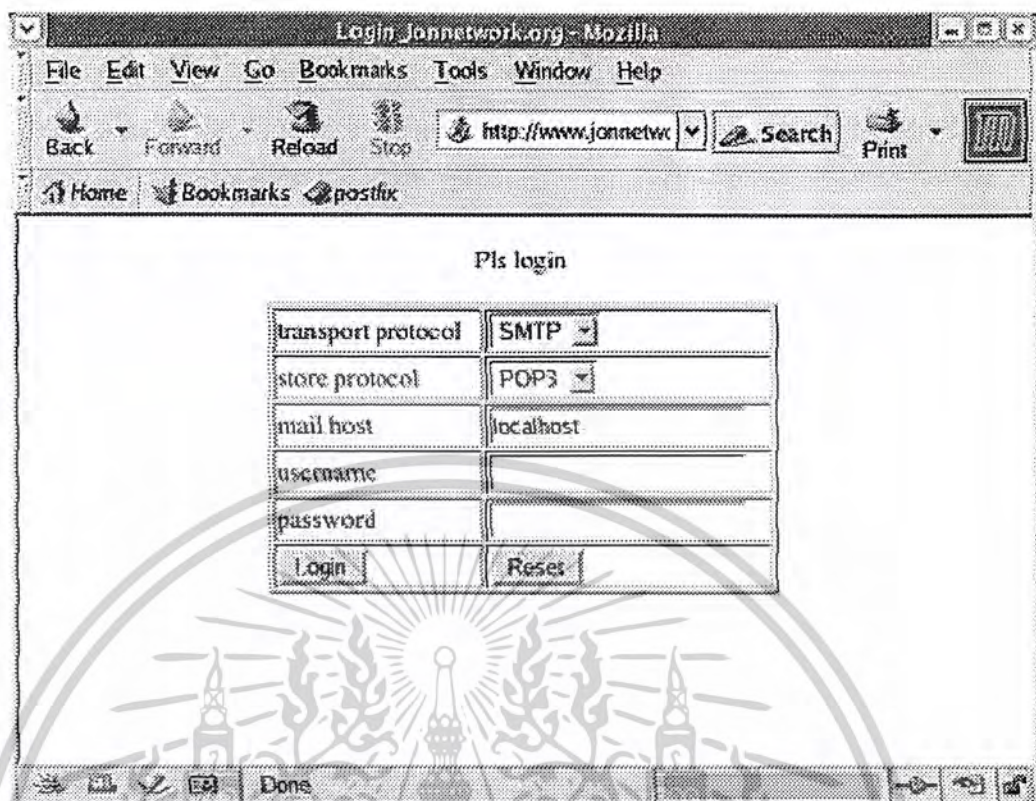
เป็นการออกแบบ User Interface ของระบบรับ-ส่งอีเมลล์ผ่าน Web Browser โดยมี

Sendmail	เป็น	Mail Server
Tomcat	เป็น	Web Server
Bind	เป็น	DNS Server

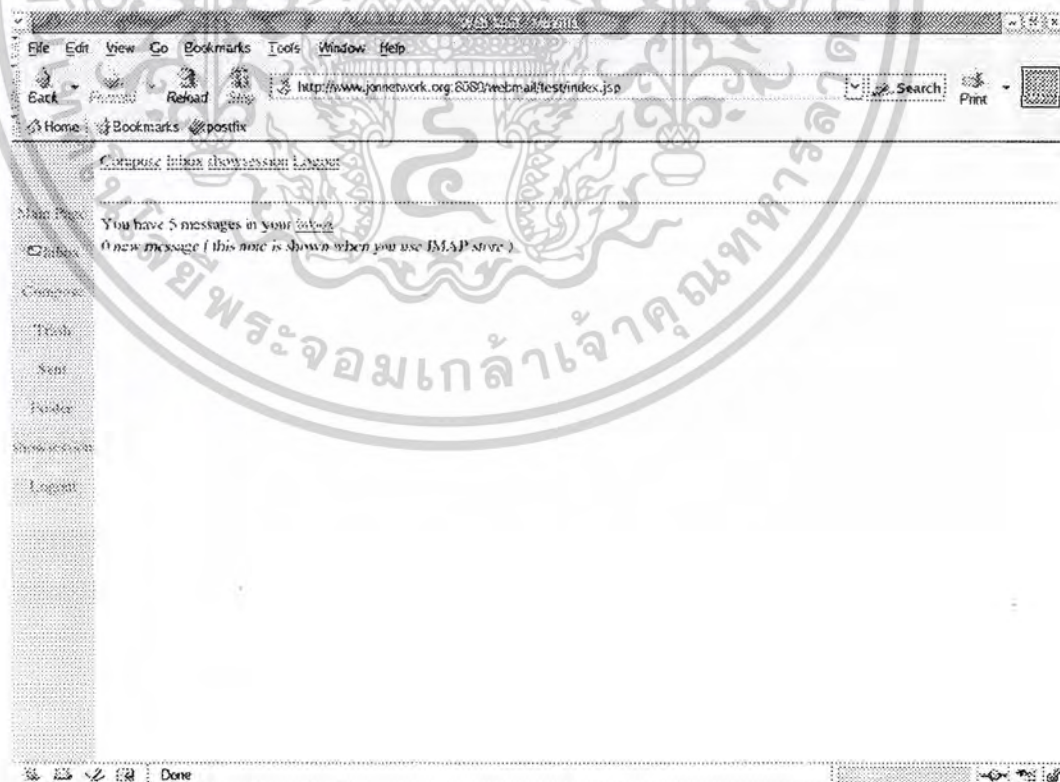
ลักษณะระบบจะคล้ายกับระบบรับ-ส่งอีเมลล์ทั่วไป คือ

1. มีการ Login เข้ามาในระบบ โดยการทำ Session
2. แสดงผลของ INBOX
3. แสดงรายละเอียดของเมลล์
4. ส่งอีเมลล์ออกไปหาผู้อื่นได้ รวมทั้งการ Attachment
5. ลบอีเมลล์ที่ไม่ต้องการทิ้ง
6. สร้าง Mailling List ได้
7. ป้องกัน Spam เมลล์ และ Spam ไวรัสได้

ในส่วนของ username และ password จะต้องกำหนดไว้ก่อนดังเช่นเมลล์สถาบัน



รูปที่ 4-2 แสดงหน้าการ เข้าสู่ระบบ แบบคร่าวๆ
ในการ login เข้าสู่ระบบ สามารถเลือกได้ว่า จะเลือกใช้โปรโตคอลการอ่าน e-mail แบบ POP3 หรือ IMAP



รูปที่ 4-3 แสดงหน้า main page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<%
    String host      = (String)session.getValue("host");
    String username  = (String)session.getValue("username");
    String password  = (String)session.getValue("password");
    String store     = (String)session.getValue("store");

    try{
        String fold = "INBOX";
        Properties pro = new Properties();
        pro.put("mail.store.host", host);
        Session ses = Session.getDefaultInstance(pro, null);
        Store st = ses.getStore(store);
        st.connect(host, username, password);
        Folder f = st.getFolder(fold);
        f.open(Folder.READ_WRITE);
        Message msg[] = f.getMessages();
        int j = msg.length - 1 ;
        for (int i=j; i>=1; i--){
            <tr><td><input type="checkbox" name="checkbox" value="<%=i%>"></td>
            <td><%=msg[i].getFrom()[0]%></td>
            <td><a href="showmail.jsp?index=<%=i%>&fold=<%=fold%>"> :
            <%=msg[i].getSubject()%></a></td>
            <td><%=msg[i].getSize()%></td>
            <td><%=msg[i].getSentDate()%></td></TD>
        }
        f.close(false); st.close();
    }
%>

```

รูปที่ 4-4 แสดงโค้ดการเรียกดู INBOX

```

<form action="send.jsp" method="post">
  <table align="center">
    <tr><td>To</td><td><input name="to" size="50" /></td></tr>
    <tr><td>CC</td><td><input name="cc" size="50" /></td></tr>
    <tr><td>BCC</td><td><input name="bcc" size="50" /></td></tr>
    <tr><td>Subject</td><td><input name="subject" size="50" /></td></tr>
    <tr><td>Attachment</td><td><input type="file" name="attach"></td></tr>
    <tr><td colspan="2"><textarea name="text" cols="50" rows="20"></td></tr>
    <tr><td colspan="2" align="center"><input type="submit" value="Send Email"/></td></tr>
  </table>
</form>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4-5 แสดงไฟล์ `compose.jsp` ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<%@ page import="java.util.*, javax.mail.*, javax.mail.internet.*, javax.activation.*, java.io.*" %>
<%
Properties props = new Properties();
props.put("mail.smtp.host", (String)session.getValue("host"));
Session s = Session.getInstance(props,null);
MimeMessage message = new MimeMessage(s);
InternetAddress from = new InternetAddress((String)session.getValue("username")+"@jonnetwork.org");
    message.setFrom(from);

String toAddresses = request.getParameter("to");
    message.addRecipients(Message.RecipientType.TO, toAddresses);
String ccAddresses = request.getParameter("cc");
    message.setRecipients(Message.RecipientType.CC, ccAddresses);
String bccAddresses = request.getParameter("bcc");
    message.setRecipients(Message.RecipientType.BCC, bccAddresses);
String subject = request.getParameter("subject");
String subject2 = new String(subject.getBytes("ISO8859_1"),"TIS-620");
    message.setSubject(subject2);
String text = request.getParameter("text");
String text2 = new String(text.getBytes("ISO8859_1"),"TIS-620");
    message.setText(text2);
Transport.send(message);
%>

```

รูปที่ 4-6 แสดงไฟล์ send.jsp

ในการสร้าง e-mail จะใช้ ไฟล์ compose.jsp เป็นตัวสร้าง form ของ e-mail แล้วทำการส่งค่าต่างๆ ไปยัง send.jsp เพื่อส่ง e-mail ต่อไป

4.2 เครื่องมือที่ใช้ในการพัฒนา

Software :

- Linux Fedura Core 2
- JSP
- Jakarta Tomcat 5.5.4
- JDK 1.5.0.01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคคลในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Sendmail-cf-8.12.11-4.6
- M4-1.4.1-15
- Imap-2002d-3
- Bind

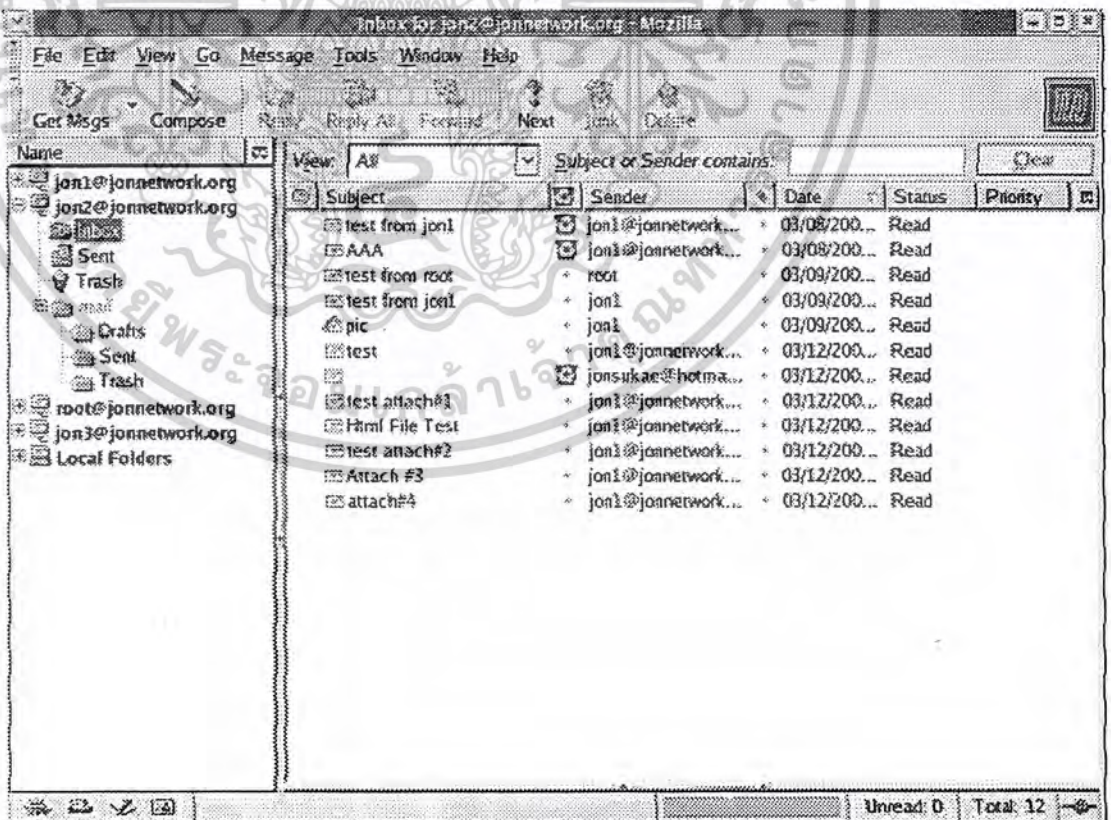
Hardware :

- Fujitsu C 2210
- CPU 1.8 GHz
- RAM 512 MB
- เครื่อง server IP = 161.246.5.155

4.3 การทดสอบการทำงาน

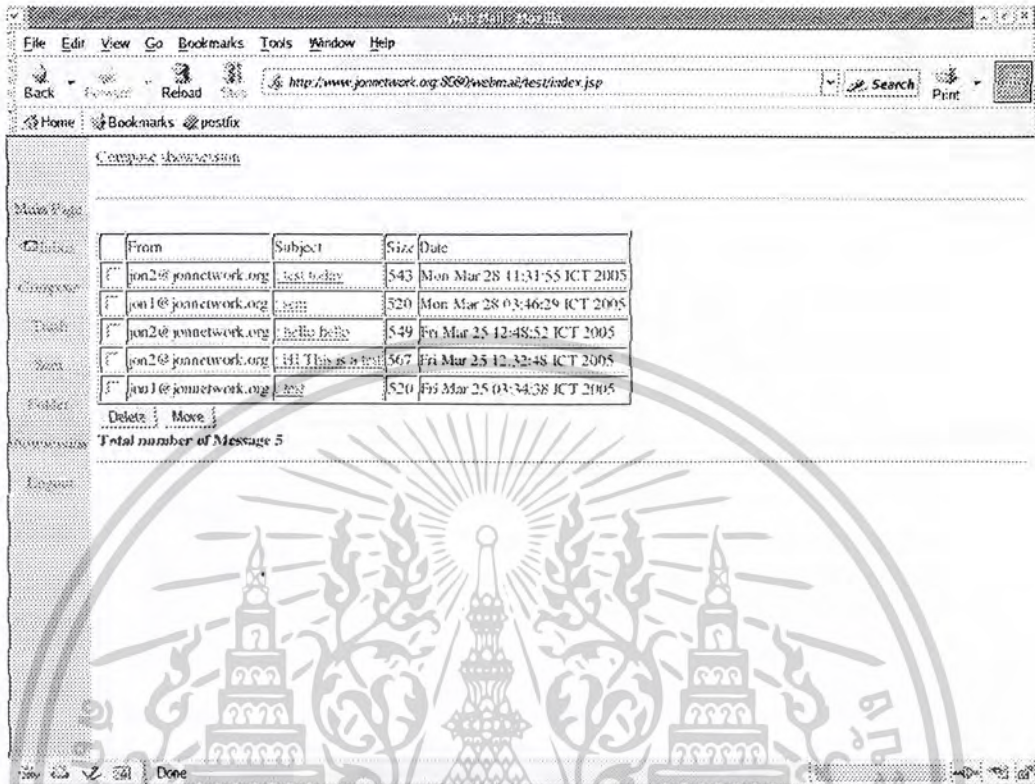
เพื่อให้แน่ใจว่าระบบจะสามารถทำงานได้ ควรจะตรวจสอบดังนี้

1. เปิด service ของ bind หรือยัง
2. start Tomcat หรือยัง
3. start service ของ POP3 / IMAP หรือยัง
4. เปิด พอร์ต SMTP หรือยัง



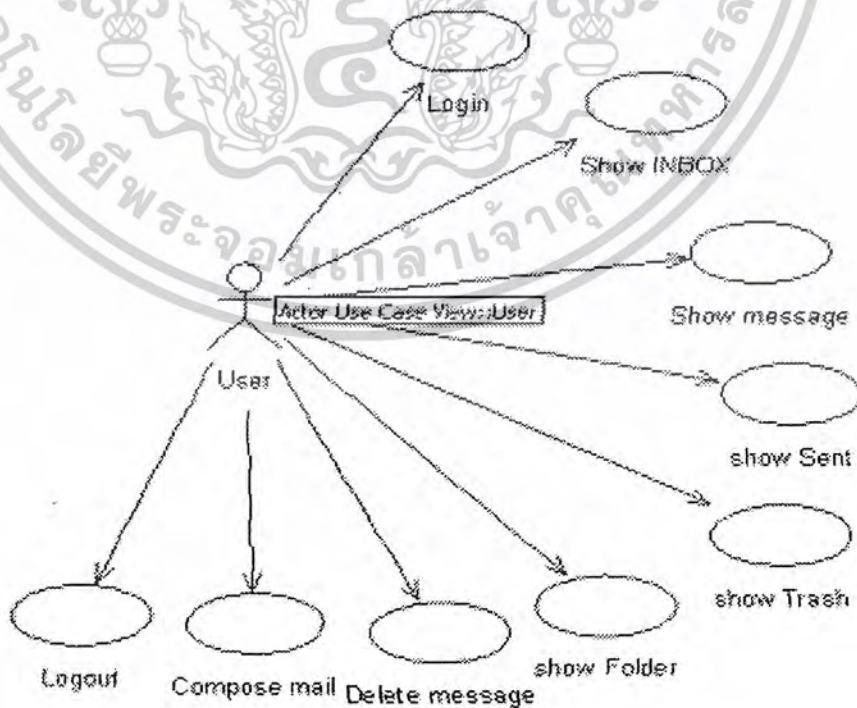
รูป 4-7 แสดงการเรียกดู INBOX ผ่าน Mozilla Mail

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



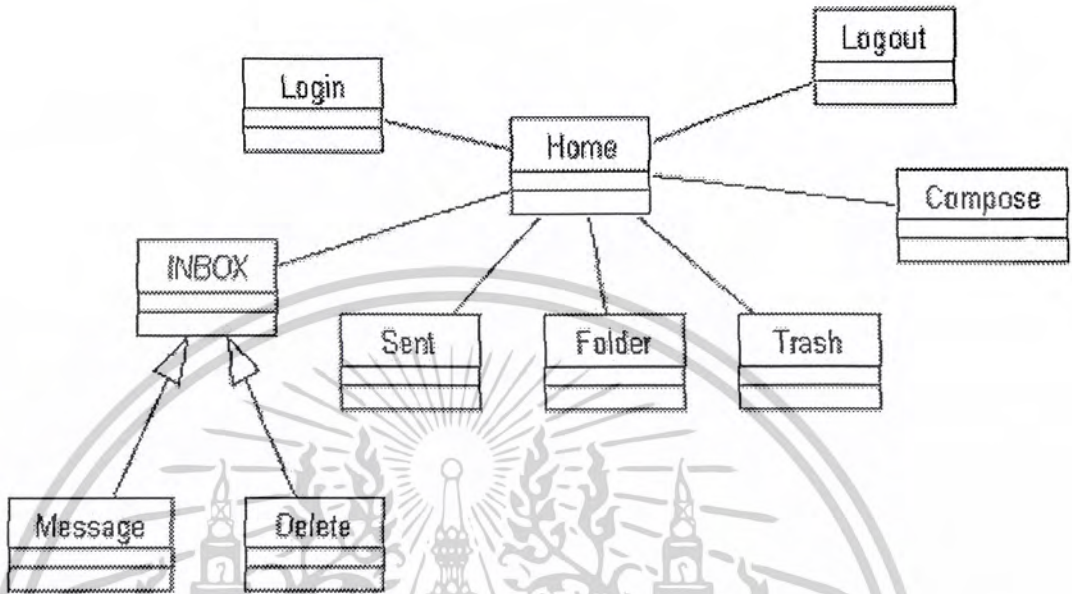
รูปที่ 4-8 แสดงการเรียกดู INBOX ด้วย web browser

4.4 โครงสร้างโปรแกรม (Diagrams ต่างๆ)



รูปที่ 4-9 Usecase Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในองค์กรซึ่งจะเผยแพร่โดยไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

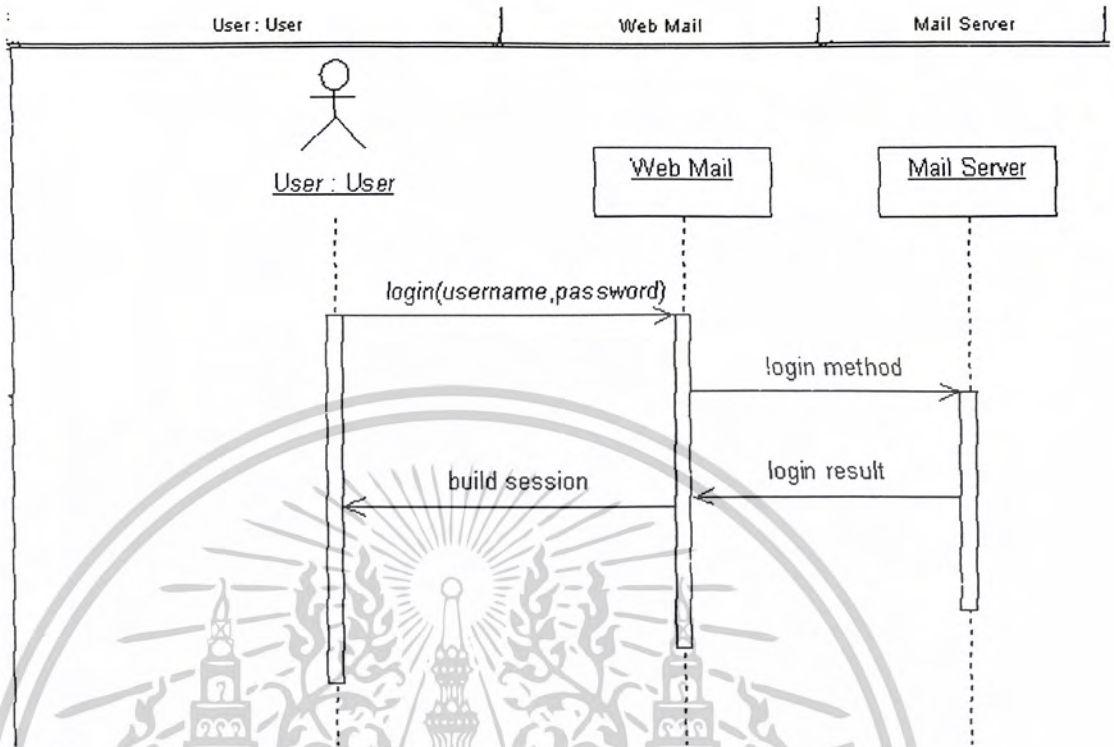


รูปที่ 4-10 Class Diagram

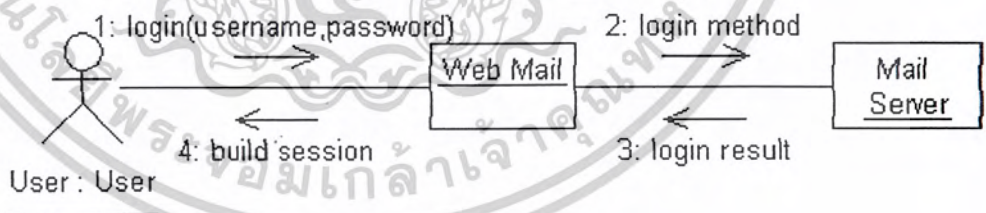


รูปที่ 4-11 Object Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

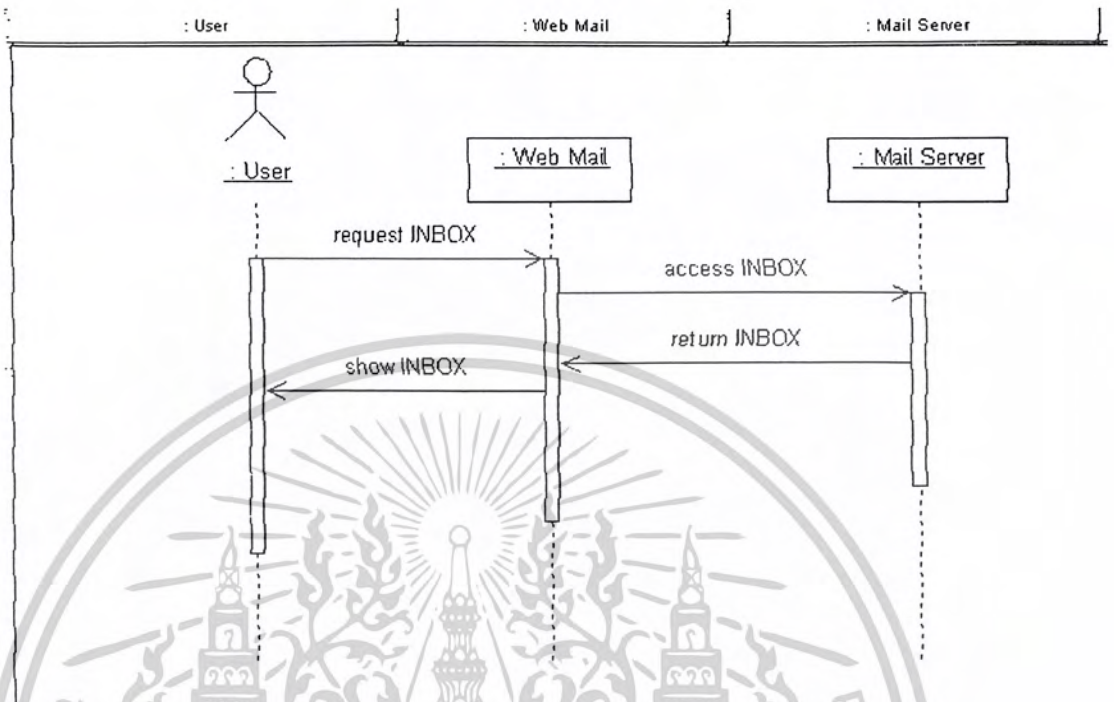


รูปที่ 4-12 Login sequence diagram

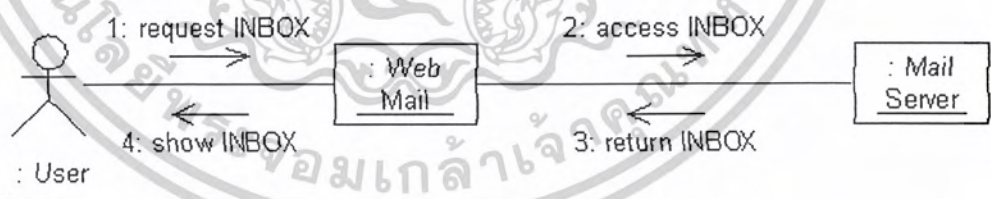


รูปที่ 4-13 Login collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

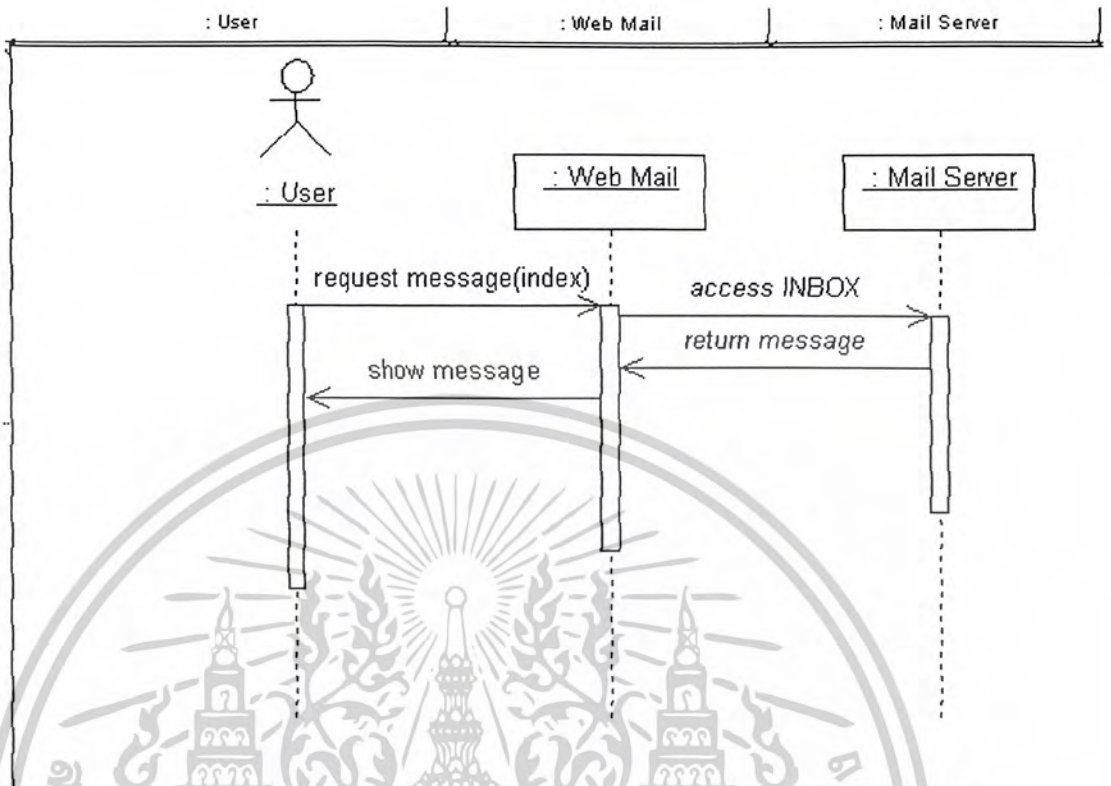


รูปที่ 4-14 Show INBOX sequence diagram

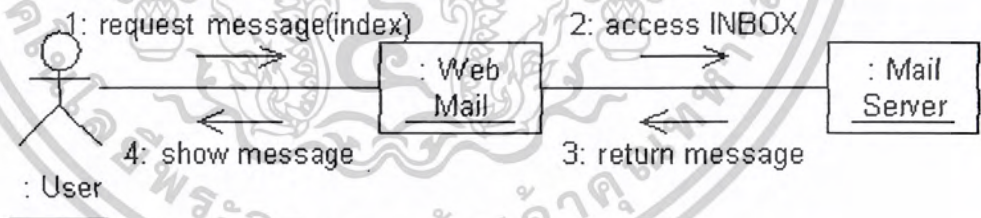


รูปที่ 4-15 Show INBOX collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

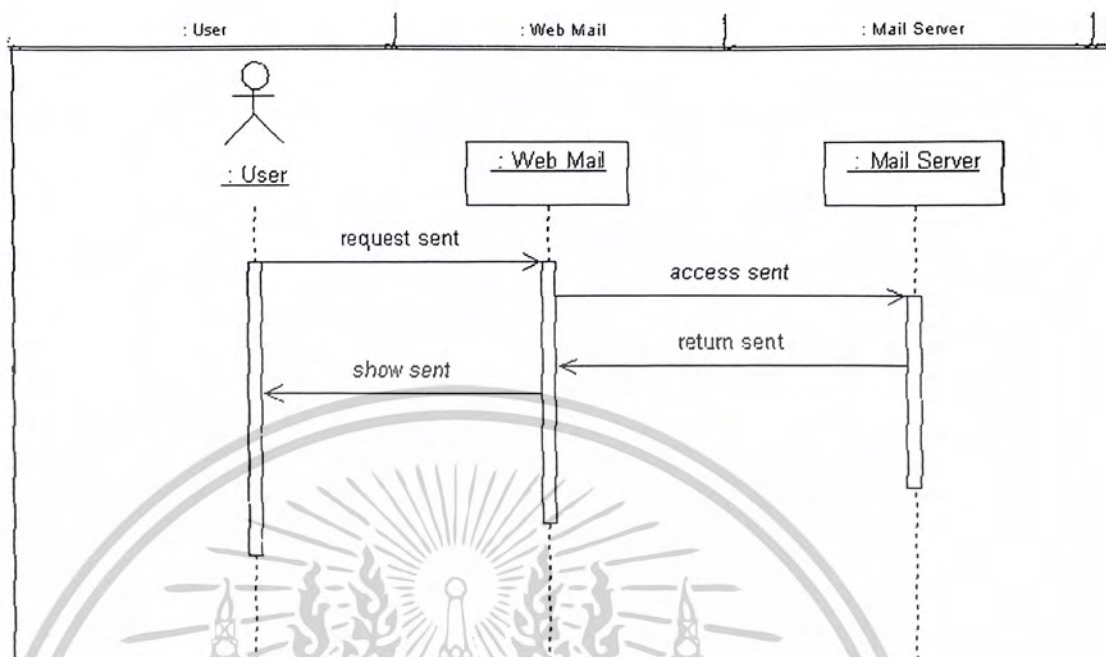


รูปที่ 4-16 Show message sequence diagram

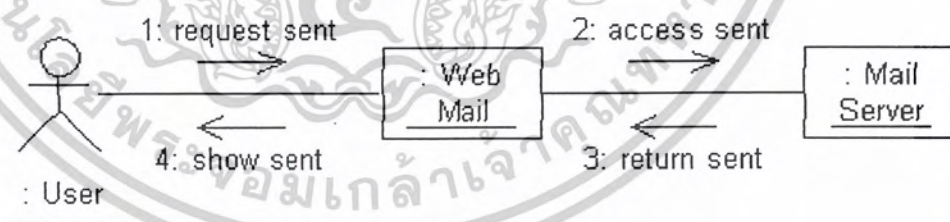


รูปที่ 4-17 Show message collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

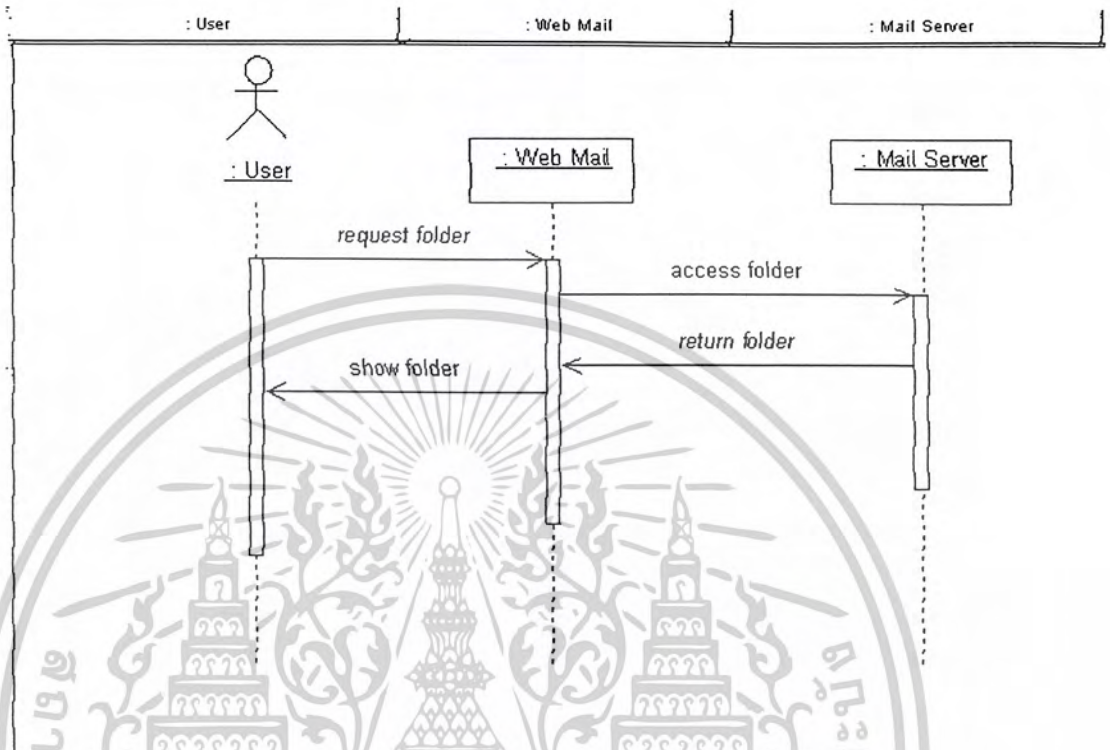


รูปที่ 4-18 Show Sent sequence diagram

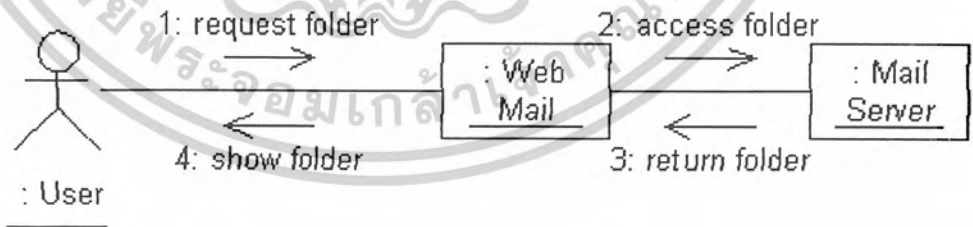


รูปที่ 4-19 Show Sent collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

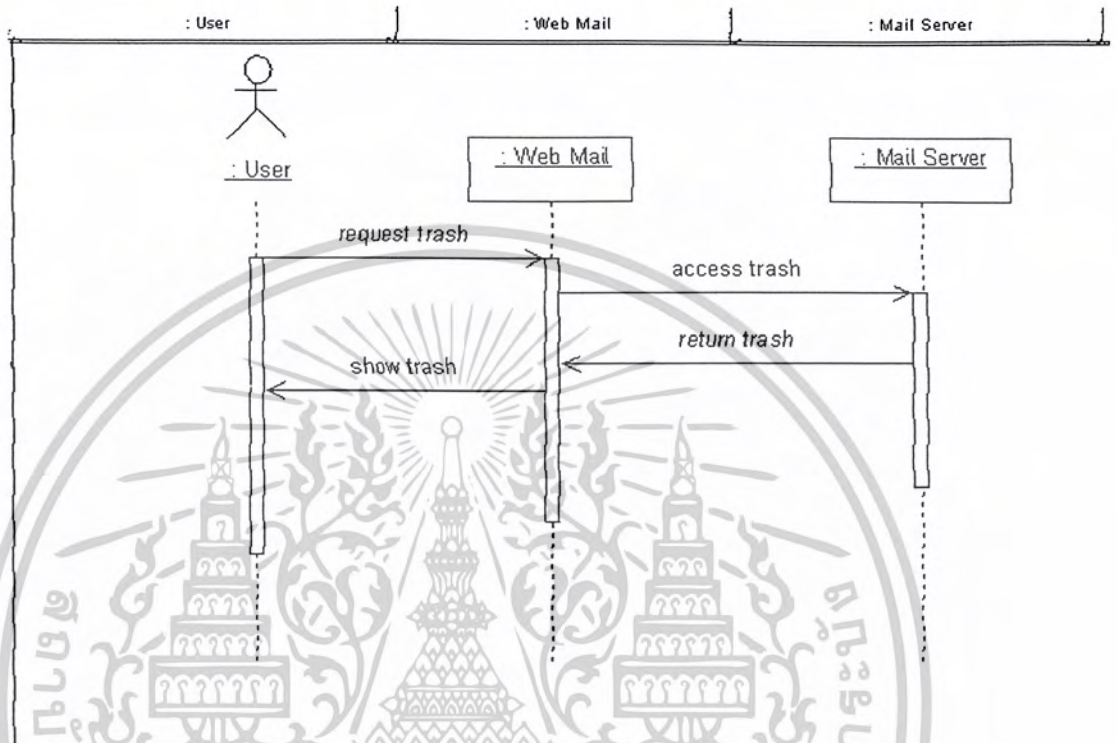


รูปที่ 4-20 Show Folder sequence diagram

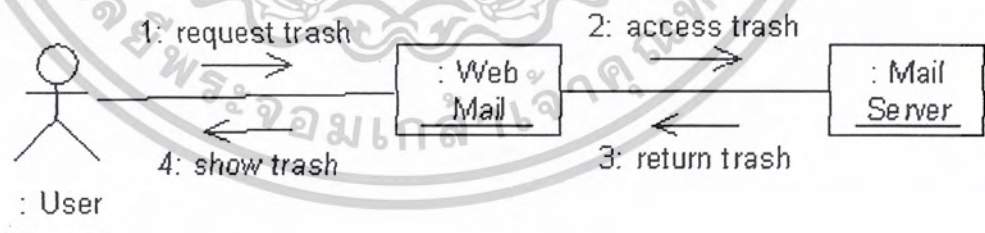


รูปที่ 4-21 Show Folder collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

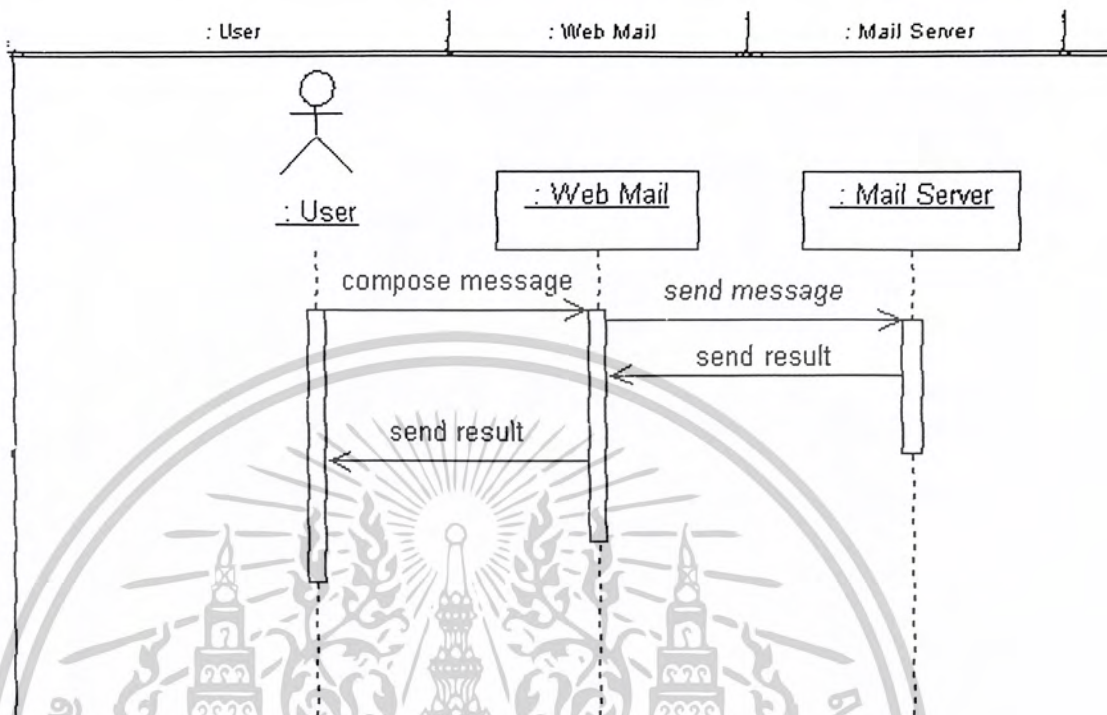


รูปที่ 4-22 Show Trash sequence diagram

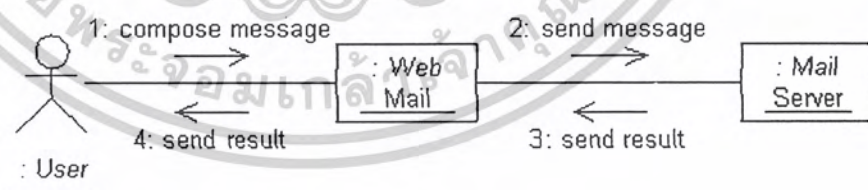


รูปที่ 4-23 Show Trash collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

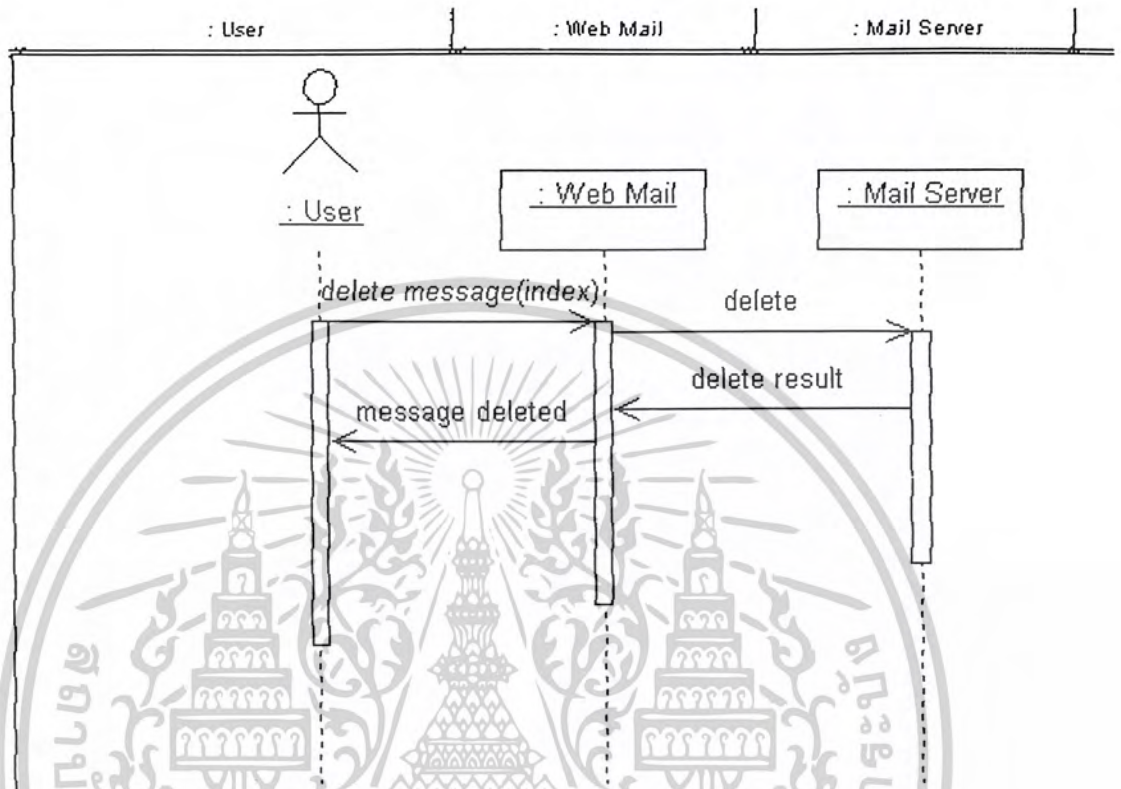


รูปที่ 4-24 Compose sequence diagram

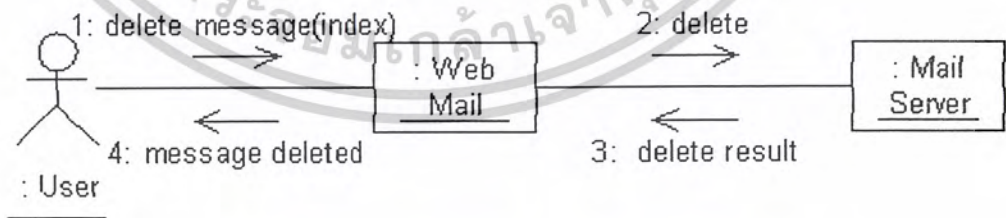


รูปที่ 4-25 Compose collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

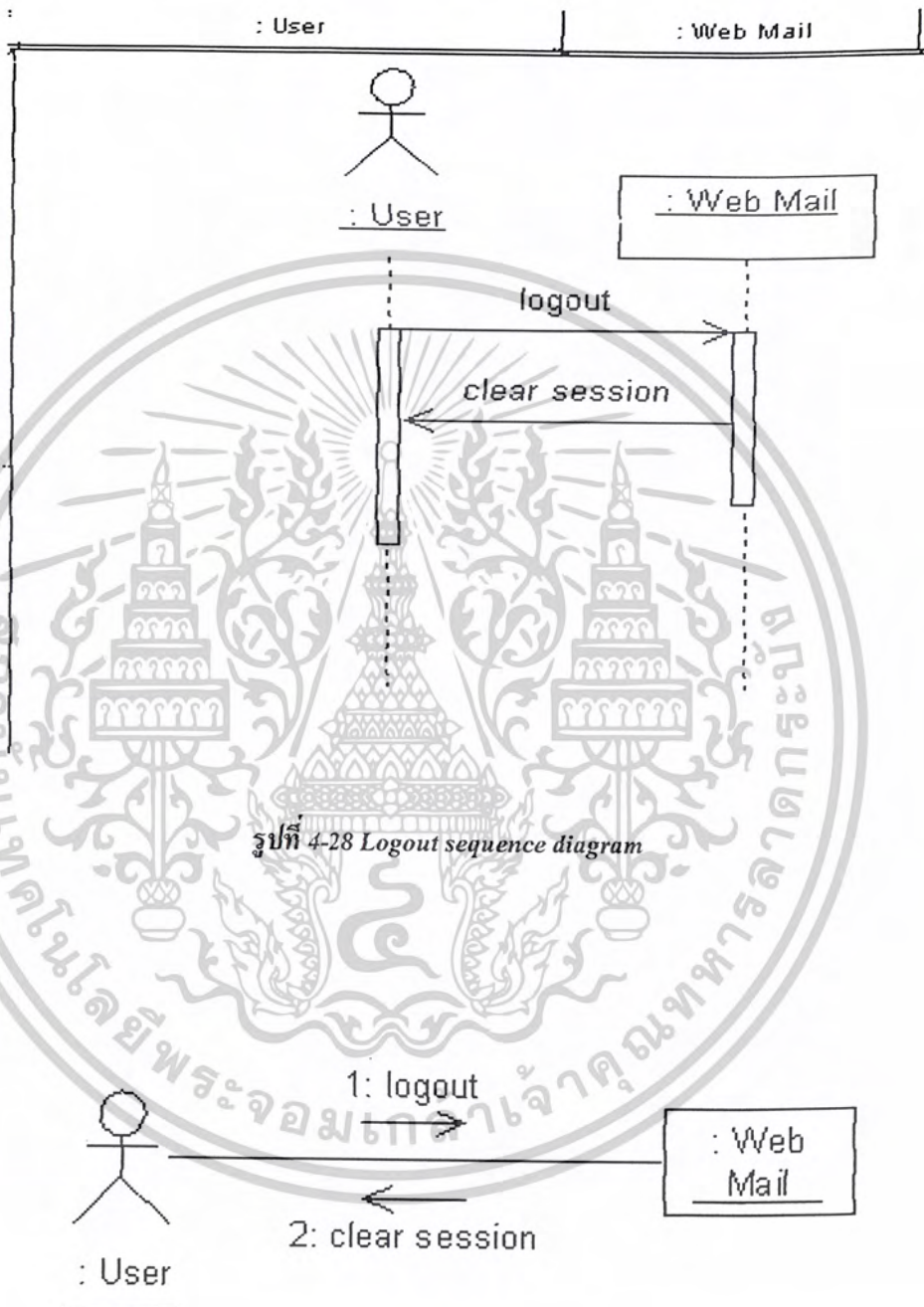


รูปที่ 4-26 Delete sequence diagram



รูปที่ 4-27 Delete collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-28 Logout sequence diagram

รูปที่ 4-29 Logout collaboration diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

5.1 ปัญหาและอุปสรรคในการทำงาน

- เนื่องจากผู้จัดทำ ไม่มีความรู้เรื่อง Linux มาก่อนจึงต้องเสียเวลาในการศึกษาเกี่ยวกับระบบปฏิบัติการ Linux ก่อน
- การติดตั้งโปรแกรมใน Linux ทำได้ยาก กว่าระบบ Windows

แนวทางการแก้ไขปัญหาและอุปสรรค

- ศึกษาข้อมูลต่างๆเกี่ยวกับ Linux จาก internet เช่นการลง โปรแกรม และคำสั่งต่างๆ
- หาหนังสือต่างๆที่เกี่ยวข้องมาอ่าน

5.2 แนวทางการพัฒนาต่อไปในอนาคต

สามารถนำระบบนี้ไปศึกษาต่อในระบบ WAP เพื่อที่จะสามารถ รับ-ส่ง อีเมลล์ผ่านมือถือได้ทาง ระบบ WAP (WAP-Mail)

5.3 ข้อสรุปและข้อเสนอแนะ

- ทางผู้จัดทำต้องขอยกยในหลายๆความผิดพลาด เพราะ ผู้จัดทำคิดว่า ระบบรับ-ส่ง เมลล์นี้ยังไม่ดีพอ เนื่องด้วยปัญหาบางประการ
- ข้อบกพร่องของโปรแกรม คือ ยังไม่สามารถแนบไฟล์ไปในอีเมลล์ได้ , ไม่มีการสร้างและตั้งชื่อไฟล์เคอร์เองได้ , ไม่มีการสร้างรายชื่อผู้รับ
- รูปแบบใหม่ๆที่ควรที่จะเพิ่มเข้าไป เช่น สร้างรายชื่อผู้รับ(mailing list) เก็บใน database , สามารถสร้างและตั้งชื่อ ไฟล์เคอร์ได้ , สามารถแนบไฟล์ไปในอีเมลล์ได้ , สร้างความปลอดภัยให้กับเว็บเมลล์ เช่น ใช้ ssl และป้องกัน session hijack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ดร.วีระศักดิ์ ชิงถาวร, “Fundamental of Java Programming volume2”,กรุงเทพฯ:บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน),512 หน้า,2547.
2. ก่อกิจ วีระอาชากุล, “ติดตั้งและปรับแต่งเซิร์ฟเวอร์ Linux สำหรับ Admin. Linux โดยเฉพาะ”,นนทบุรี : อินโฟเพรส , 432 หน้า ,2545.
3. อ. บัณฑิต จามาศูติ , “คัมภีร์ Red Hat Enterprise Linux เล่ม 2”,Bandhit Press , 468 หน้า ,2547
4. Marty Hall, Larry Brown, “ Core Servlets and Java Server Pages Volume 1 : Core Technologies 2nd Edition,Sun Microsystems Press
5. <http://java.sun.com/docs>
6. <http://www.sendmail.org>
7. <http://www.apache.org>
8. <http://www.linux.org>

ภาคผนวก(Appendix)

ภาคผนวก ก

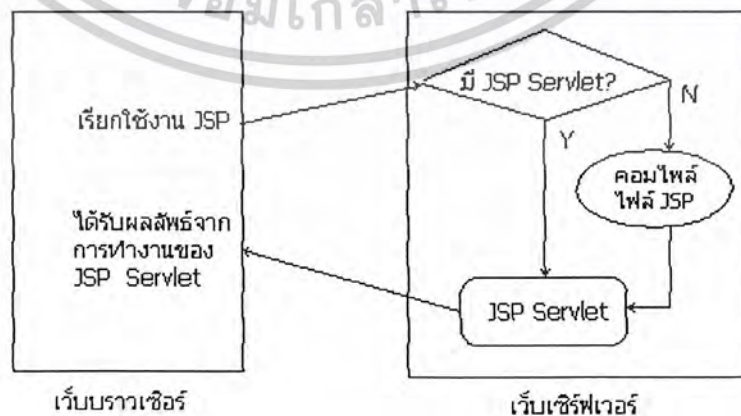
JavaServer Pages (JSP)

เป็น web-scripting เทคโนโลยีคล้ายกับ Netscape server-side JavaScript (SSJS) หรือ Microsoft Active Server Pages (ASP) แต่ผลิตภัณฑ์ที่หัวใจของ JSP คือ Java ซึ่งเป็นภาษาที่ค่อนข้างปลอดภัยที่ออบเจกต์ (object-oriented style) ซึ่งช่วยทำให้ง่ายต่อการพัฒนาในโปรเจกต์ใหญ่ ๆ ตลอดจนสามารถนำส่วนประกอบต่าง ๆ กลับมาใช้ได้อีก (software reusable) จุดเด่นที่สำคัญของ JSP คือสามารถทำงานได้โดยไม่ต้องขึ้นอยู่กับผู้ผลิตซอฟต์แวร์รายใดรายหนึ่งโดยเฉพาะ ซึ่งโดยทั่วไปเทคโนโลยีต่าง ๆ มักจะออกมาในลักษณะของผลิตภัณฑ์จากบริษัทผู้ผลิตแห่งใดแห่งหนึ่ง แต่ JSP ใช้ลักษณะของ specification ซึ่งกำหนดโดย Sun Microsystems ดังนั้นผู้ผลิตซอฟต์แวร์จึงสามารถอ้างอิง specification ที่กำหนดขึ้น ผลิต JSP Container (ตัวที่ใช้ในการรัน JSP) ขึ้นมาใช้กับแพลตฟอร์มใดก็ได้

JSP

JSP Pages (ไฟล์ที่เขียนขึ้นโดยใช้ JSP script และลงท้ายด้วย .jsp) จะถูกรันโดย JSP Container ซึ่งมักจะเป็นส่วนประกอบที่อยู่ใน Webserver หรือ เป็นตัวแอดออนใน Application Server. โดยทั่วไป JSP Container จะเป็นตัวรับ request จาก client ส่งผ่านไปยัง JSP Page และส่งค่าที่ได้จากการประมวลผลโดย JSP Page กลับไปยัง client. JSP Container ที่เราจะใช้กันคือ Jasper ซึ่งเป็น JSP Container ที่อยู่ใน Tomcat Servlet Engine จากค่าย apache (www.apache.org) ซึ่งทาง Sun ใช้เป็นตัวอ้างอิงในวงการ JSP โดยในอนาคต Tomcat จะเป็นตัวที่ใช้ในการรัน Servlet แทน Jserv ซึ่งใช้กันอยู่ในปัจจุบัน

การทำงานของ JSP



รูปที่ ก-1 แสดงการทำงานของ JSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเว็บเบราว์เซอร์ มีการร้องขอโดยผู้ใช้เพื่อเรียกข้อมูล JSP ที่อยู่บนเซิร์ฟเวอร์ ซึ่งก็หมายถึงไฟล์ที่มีนามสกุล JSP เว็บเซิร์ฟเวอร์จะทำการตรวจสอบว่า ไฟล์นั้นมีการคอมไพล์ไปเป็น JSP Servlet แล้วหรือยัง ถ้ายังไฟล์ JSP จะถูกคอมไพล์ โดยตัวเว็บเซิร์ฟเวอร์ ไปเป็น JSP Servlet แต่ถ้าไฟล์ถูกคอมไพล์แล้ว (ซึ่งเกิดขึ้นจากการใช้งานในครั้งแรก) เว็บเซิร์ฟเวอร์จะนำ JSP Servlet มาใช้งาน โดยการเรียกเพื่อประมวลผล ได้ผลลัพธ์อย่างไรก็จะเกิดเป็นข้อมูล ซึ่งเว็บเซิร์ฟเวอร์ จะส่งกลับไปให้ผู้เรียกใช้งาน และการประมวลผลของ JSP Servlet นี้เอง ที่ก่อให้เกิดข้อมูลเว็บในแบบไดนามิก

Directive

ไคเรกทีฟ (Directive) คือส่วนคำสั่งที่ถูกสร้างไว้ใน HTML ของไฟล์นามสกุล JSP เพื่อบอกให้เว็บเซิร์ฟเวอร์ทำการ ตามคำสั่งที่อยู่ในไคเรกทีฟ เช่น กระทำตามคำสั่งของภาษาจาวา หรือทำการกำหนดรูปแบบการใช้งาน เป็นต้น ไคเรกทีฟมีอยู่หลายรูปแบบ ดังต่อไปนี้

Page Directive เป็นไคเรกทีฟที่ใช้บอกรูปแบบการใช้งาน JSP ของผู้ใช้ มีรูปแบบในการกำหนดดังนี้

Attribute	ความหมาย	ค่าเริ่มต้น	ตัวอย่าง
info	ข้อมูลอธิบาย การทำงาน	-	<%@ page info="แบบฟอร์มลงทะเบียน" %>
language	ภาษาสคริปต์ที่ใช้งาน	"java"	<%@ page language="java" %>
contentType	ชนิดข้อมูล ในรูปแบบของ MIME	"text/html"	<%@ page contentType="text/html; charset=windows-874" %>
extends	ชื่อ Superclass ที่ถูกใช้งาน (ใช้งานน้อยครั้ง เพราะการทำงาน ถูกกำหนด โดยเซิร์ฟเวอร์)	-	<%@ page extends="com.taglib.wdjsp.MyJspPage" %>
import	สำหรับเรียกใช้งานคลาสใน แพ็กเกจภายนอก	-	<%@ page import="java.net.URL" %> <%@ page import="java.util.*, java.text.*" %>
session	กำหนดเซสชันในการทำงาน (กลุ่มกิจกรรมหนึ่งทำงานได้ หลายเว็บเพจ)	"true"	<%@ page session="true" %>
buffer	หน่วยความจำสำหรับทำบัฟเฟอร์ ให้กับข้อมูลผลลัพธ์	"8kb"	<%@ page buffer="12kb"%> <%@ page buffer="false" %>
autoFlush	ควบคุมการล้างข้อมูลบัฟเฟอร์	"true"	<%@ page autoFlush="false" %>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

isThreadSafe	กำหนดลักษณะการทำงานแบบ thread (หลายโปรเซสย่อยทำงานพร้อมกัน)	"true"	<%@ page isThreadSafe="true" %>
errorPage	กำหนดเว็บเพจ หาก JSP เกิดข้อผิดพลาดในการทำงาน ให้กับผู้ ใช้	-	<%@ page errorPage="resules/failed.jsp" %>
isErrorPage	กำหนดว่ามีเพจแสดงข้อผิดพลาดหรือไม่	"false"	<%@ page isErrorPage="false" %>

ตารางที่ ก-1 แสดงรูปแบบ page directive

- Page Directive สามารถกำหนด Attribute ได้มากกว่าหนึ่งส่วนได้ โดยแยกแต่ละส่วนด้วยช่องว่างเช่น

```
<%@ page language="java" import="java.net.*" %>
```

หมายถึง JSP ที่ใช้มีคำสั่งภาษาจาวา เป็นภาษาสคริปต์ และเรียกใช้งานคลาสจากแพ็คเกจ java.net มาใช้งาน

```
<%@ page info="This is a invalid page" session="false" buffer="16kb" autoFlush="false" session="false" %>
```

หมายถึงกำหนดข้อความให้กับ JSP (หน้าเว็บ) เป็นข้อมูล This is a invalid page และไม่ให้เพจนี้มีการทำงานแบบเซสชัน มีบัฟเฟอร์ในการใช้งาน 16 กิโลไบต์ ไม่ให้มีการล้างบัฟเฟอร์เมื่อเต็ม
- Include Directive คือ ไคเรกทีฟสำหรับ การผนวกเอาไฟล์อื่นมาแทรกลงในตำแหน่งที่มีไคเรกทีฟนี้อยู่ โดยมีรูปแบบการใช้งานคือ <%@ include file="ชื่อไฟล์ หรือ URL ที่นำข้อมูลมาผนวก" %>
- Tag Library Directive JSP มีข้อดีที่สำคัญคือ ผู้สร้าง สามารถกำหนดแท็ก JSP ขึ้นมาใช้งานได้เอง เพื่อทำให้เว็บ หรือคำสั่ง JSP ของตัวเองมีลักษณะเฉพาะตัว (เหมือนกับที่ XML มีแท็กคำสั่งที่สร้างขึ้นมาเองได้) แท็ก JSP ที่สามารถกำหนดขึ้นมาได้นี้ ช่วยให้สามารถขยายความสามารถของ JSP เพิ่มเติมขึ้นมาได้

นอกนั้นเป็นคำสั่ง HTML ปกติโดยทั่วไป และ JSP ต้องบันทึกไว้ในไฟล์นามสกุล JSP และถูกเก็บไว้ในเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงาน JSP ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้