



ระบบ รับ-ส่ง จดหมายอิเล็กทรอนิกส์ผ่านเว็บเบราว์เซอร์

WEB BASED E-MAIL

นายดำรง	วงศ์เหล็กภัย	รหัส 42360503
นายอนุฤทธิ์	ศรีคมขำ	รหัส 42360727
นางสาวสลักจิตร์	มนอิน	รหัส 42360693

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ.....7/เม.ย. 2553/.....
เลขทะเบียน.....14942061 e-2
เลขเรียกหนังสือ.....๗/๕.....
มหาวิทยาลัยนเรศวร ๑๔๙3 5 25๕๕

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2545



ใบรับรองโครงการวิจัย

หัวข้อโครงการ	ระบบ รับ-ส่ง จดหมายอิเล็กทรอนิกส์ผ่านเว็บเบราว์เซอร์		
ผู้ดำเนินโครงการ	นายดำรงค์	วงศ์เหล็กภัย	รหัส 42360503
	นายอนุฤทธิ์	ศรีคมขำ	รหัส 42360727
	นางสาวสลักจิตร์	มณเฑียร	รหัส 42360693
อาจารย์ที่ปรึกษา	อาจารย์ภาณุพงศ์ สอนคม		
สาขา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
การศึกษาปี	2545		

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการ
ศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
คณะกรรมการตรวจสอบโครงการ

.....ประธานกรรมการ
(อาจารย์ตีติโชค เชาวกุล)

.....กรรมการ
(อาจารย์พรพิศุทธิ์ วรจิรันถัน)

.....กรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

.....กรรมการ
(อาจารย์รัฐภูมิ วรานุสาสน์)

หัวข้อโครงการ	ระบบ รับ-ส่ง จดหมายอิเล็กทรอนิกส์ผ่านเว็บเบราว์เซอร์		
ผู้ดำเนินโครงการ	นายดำรงค์	วงศ์เกตุกัญ	รหัส 42360503
	นายอนุฤทธิ์	ศรีคมขำ	รหัส 42360727
	นางสาวสลักจิตร	มนอิน	รหัส 42360693
อาจารย์ที่ปรึกษา	อาจารย์ภาณุพงศ์ สอนคม		
สาขา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
การศึกษานี้	2545		

บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อพัฒนาโปรแกรมรับส่งจดหมายอิเล็กทรอนิกส์ผ่านเว็บเบราว์เซอร์ เพื่อใช้ทดแทนโปรแกรมในการรับส่งอีเมลล์(เช่น โปรแกรม Outlook) เพราะมีการใช้งานที่ง่ายกว่า หลักการทำงานคือ เมื่อผู้ใช้กรอกยูอาร์แอล (URL) ของโปรแกรมรับส่งจดหมาย และทำการล็อกอิน เพื่อเข้าสู่ระบบรับ-ส่งจดหมายของแต่ละผู้ใช้ ผู้ใช้จะสามารถรับ-ส่งจดหมายอิเล็กทรอนิกส์ และยังสามารถบันทึก ชื่อ ข้อมูล รายละเอียด ในสมุดรายชื่อ รวมถึงสามารถปรับแต่งค่าต่าง ๆ ของโปรแกรมรับ-ส่งจดหมายได้ ซึ่งโครงการนี้ ใช้ภาษาพีเอชพี (PHP) ในการพัฒนาโปรแกรม และใช้มายเอสคิวแอล (MySQL) เป็นตัวจัดการฐานข้อมูล โดยทำงานบนเครื่องคอมพิวเตอร์แม่ข่าย (Server) ที่เป็นระบบปฏิบัติการลินุกซ์ (Linux)

ผลที่ได้จากการทำโครงการนี้ คือ ได้โปรแกรมรับ-ส่งจดหมายอิเล็กทรอนิกส์ที่ผู้ใช้สามารถทำงานได้จริงบนอินเทอร์เน็ต (Internet) ทำให้ใช้งานได้สะดวก โดยที่ไม่ต้องตั้งค่าใด ๆ

Project Title	Web based e-mail		
Name	Mr. Damrong	Wongleegpay	ID. 42360503
	Mr. Anurit	Sikomkam	ID. 42360727
	Miss Salukjit	Monin	ID. 42360693
Project Advisor	Mr.Panupong	Sonkom	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2002		

.....

ABSTRACT

This Project has been built to develop e-mail program on web browser for replacing e-mail program application (e.g. Outlook), because it has easier application. Principle of e-mail program ; user put his URL of e-mail program and then login to come into the e-mail system of each user. User can send and receive his e-mail and can record data of other one in address book including exchanging value of e-mail program. This project is used by PHP language to develop the program and used MySQL to manage database and runs on server which use Linux operating system.

The result of this project is e-mail program which really run on internet and help user more comfortable without setting any value.

กิตติกรรมประกาศ

ผู้จัดทำโครงการ ขอขอบพระคุณ อาจารย์กานทุหงส์ สอนคม ที่ให้คำแนะนำเกี่ยวกับปัญหาที่พบ และช่วยหาทางแก้ไขให้ ตลอดจนสละเวลาให้คำปรึกษา เอาใจใส่ให้ความช่วยเหลืออย่างดีตลอดเวลา ในการทำโครงการนี้

รวมทั้งขอขอบพระคุณอาจารย์ในภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ทุกท่าน ที่คอยดูแล ให้ความช่วยเหลือและให้ความสะดวกในการทำโครงการนี้

นายดำรงค์ วงศ์เหล็กภัย
นายอนุฤทธิ์ ศรีคมขำ
นางสาวสลักจิตร มนอิน



สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ

บทที่ 1 บทนำ.....	1
-------------------	---

1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบข่ายงาน.....	1
1.4 กิจกรรมการดำเนินงาน.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	2
1.6 งบประมาณ.....	3

บทที่ 2 ความรู้พื้นฐาน และทฤษฎีที่เกี่ยวข้อง.....	4
---	---

2.1 ความรู้พื้นฐานเกี่ยวกับอีเมล.....	4
2.1.1 ความเป็นมา.....	4
2.1.2 ความหมาย.....	4
2.1.3 โครงสร้างและรูปแบบของชื่ออีเมลในเบื้องต้น.....	5
2.1.4 รายละเอียดของอีเมล.....	5
2.1.5 ประเภทต่างๆ ของอีเมล.....	6
2.1.6 กลไกการรับ – ส่งอีเมล.....	7
2.2 ทฤษฎีของ Protocol ที่เกี่ยวข้องกับอีเมล.....	8
2.2.1 SMTP (Simple Mail Transfer Protocol).....	8
2.2.2 POP (Post Office Protocol).....	9

สารบัญ(ต่อ)

หน้า

2.2.3 IMAP (Internet Message Access Protocol).....	10
2.3 PHP กับ Web based E-mail.....	11
2.3.1 การ Login และ Logout.....	11
2.3.1.1 Cookie.....	11
2.3.1.2 Session.....	13
2.3.2 การส่งอีเมล.....	18
2.3.2.1 การส่งเมลโดยใช้ฟังก์ชัน mail() ของ PHP.....	18
2.3.2.2 การส่งเมลโดยติดต่อ Protocol SMTP.....	21
2.3.2.3 การส่งเมลโดยใช้ PHP ติดต่อในระดับ Socket.....	23
2.3.2.4 การตรวจสอบ อีเมลแอดเดรส ว่ามีอยู่จริงหรือไม่.....	27
2.3.3 MIME รูปแบบมาตรฐานในการส่งเมล.....	30
2.3.3.1 ความหมายของ MIME.....	30
2.3.3.2 โครงสร้างของ MIME.....	30
2.3.3.3 การส่งแบบ Text และ HTML.....	31
2.3.3.4 การเข้ารหัส Binary เป็น Text แบบ base 64.....	32
2.3.3.5 การแนบไฟล์ไปกับการส่งอีเมล (Attachment).....	33
2.3.4 การรับอีเมล.....	42
2.3.4.1 การรับเมลโดยติดต่อ Protocol POP3.....	42
2.3.4.2 การรับเมลโดยใช้ PHP ติดต่อในระดับ Socket.....	47
2.3.4.3 ทำความรู้จักกับ IMAP ที่เกี่ยวข้องกับ PHP.....	47
2.3.4.4 ทดสอบการใช้ฟังก์ชัน IMAP ใน PHP.....	48
2.3.4.5 การใช้ฟังก์ชันของ PHP ที่เกี่ยวข้องกั IMAP แทน POP3.....	50
บทที่ 3 การออกแบบโปรแกรม.....	58
3.1 ออกแบบการทำงานของ web based mail โดยใช้ Data Flow Diagram.....	59
3.2 ระบบฐานข้อมูลโดยใช้ ER-Model.....	66

สารบัญ(ต่อ)

หน้า

บทที่ 4 การพัฒนาโปรแกรมและผลการทดลอง.....	71
4.1 ระบบ Login	71
4.2 ระบบรับ e-mail	74
4.3 ระบบการส่ง e-mail.....	77
4.4 ระบบ Address book.....	80
4.5 ระบบ Create User.....	83
บทที่ 5 บทสรุป	
5.1 สรุปผล.....	85
5.2 ปัญหา และ แนวทางแก้ไข.....	85
5.3 แนวทางในการพัฒนาเพิ่มเติม.....	86
เอกสารอ้างอิง.....	87
ภาคผนวก.....	88
ภาคผนวก ก ศึกษาการเซตอัฟ Linux Redhat 7.3 ให้ทำงานเป็นเครื่อง Server.....	88
ภาคผนวก ข การตั้งค่าของ Program ต่างๆ เพื่อให้รองรับการทำงานของอีเมล.....	100
ประวัติผู้ทำโครงการ.....	103

สารบัญตาราง

ตารางที่	หน้า
1.1 กิจกรรมการดำเนินงาน.....	2
2.1 ตัวอย่างแสดง SMTP ของ ISP (บางแห่ง).....	18
2.2 ตัวอย่างแสดงรหัส Reply Code ของ SMTP.....	27
2.3 สรุปคำสั่ง POP 3.....	42
3.1 ฐานข้อมูล Sign Up.....	67
3.2 ฐานข้อมูล User.....	68
3.3 ฐานข้อมูล e-mail.....	68
3.4 ฐานข้อมูลการ Attach File.....	69
3.5 ฐานข้อมูล Address Book.....	70



สารบัญรูป

รูปที่	หน้า
3.1 ภาพรวมของระบบรับ-ส่ง e-mail.....	59
3.2 ระบบ Login.....	60
3.3 ระบบ Sign Up.....	62
3.4 ระบบช่วยเหลือเมื่อลืม password.....	63
3.5 ระบบรับ-ส่ง e-mail.....	64
3.6 ระบบฐานข้อมูลโดยใช้ E-R Model.....	66
4.1 ฟังก์ชันของระบบ Login.....	71
4.2 ลักษณะของโปรแกรมในส่วนของการ login เข้าสู่ระบบ.....	72
4.3 ลักษณะของโปรแกรมในส่วนของการ login error	72
4.4 ลักษณะของโปรแกรมในส่วนของการ login error โดยไม่มีการกรอกข้อมูล.....	73
4.5 ลักษณะโปรแกรมในส่วนของการ login error โดยมีการกรอกข้อมูล user ที่ไม่มีจริง.....	73
4.6 ฟังก์ชันของระบบการรับ e-mail	74
4.7 ลักษณะของโปรแกรมในส่วนของ Inbox	75
4.8 ลักษณะของโปรแกรม ที่ทำการเปิดอ่าน e-mail	75
4.9 ลักษณะของโปรแกรมเปิดอ่าน e-mail ที่มีไฟล์ attach file และจะทำการ download	76
4.10 ฟังก์ชันของระบบการส่ง e-mail	77
4.11 ลักษณะของโปรแกรมในส่วนของการส่ง e-mail	78
4.12 ลักษณะของโปรแกรมในส่วนของการส่ง e-mail แบบ attach file.....	79
4.13 ฟังก์ชันของระบบAddress Book	80
4.14 ลักษณะของโปรแกรมในส่วนของสมุดรายชื่อ	81
4.15 ลักษณะโปรแกรมของการเพิ่มรายชื่อในสมุดรายชื่อ	81
4.16 ลักษณะโปรแกรมของรายละเอียดแต่ละคนในสมุดรายชื่อ	82
4.17 ฟังก์ชันของระบบ Create User	83
4.18 ลักษณะของโปรแกรมในส่วนของการ Create User	84

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

การรับส่งจดหมายอิเล็กทรอนิกส์ มีหลายรูปแบบ เช่น การใช้โปรแกรมในการ รับ-ส่ง e-mail ยกตัวอย่างเช่น outlook express และอีกรูปแบบหนึ่งคือการใช้ web based e-mail ซึ่งจะสามารถ รับ-ส่ง e-mail ผ่านหน้า web browser ได้โดยตรง ไม่จำเป็นต้องใช้โปรแกรม รับ-ส่ง e-mail อื่น ซึ่งจะมีความสะดวกและรวดเร็วกว่า

ในโครงการนี้ เราสนใจที่จะจัดทำ web based e-mail โดยพัฒนาจากภาษา PHP และ ใช้ MySQL เป็นตัวจัดการฐานข้อมูล โดยใช้ระบบปฏิบัติการ Linux เป็น Server

1.2 วัตถุประสงค์

1. เพื่อพัฒนาระบบการ รับ-ส่ง จดหมายอิเล็กทรอนิกส์ผ่าน web browser โดยใช้ PHP และ MySQL ซึ่งเป็น Freeware
2. เพื่อเป็นแนวทางในการพัฒนา web based e-mail ให้สำหรับผู้ที่สนใจ
3. เพื่อให้เกิดความสะดวกในการรับส่ง e-mail โดยไม่ต้องตั้งค่าใดๆ เพียงแค่มี web browser ก็สามารถใช้งานได้
4. ลดความเสี่ยงจากการติดไวรัส ที่อาจจะแฝงมากับ e-mail
5. เพื่อสนับสนุนให้คนไทยหันมาใช้บริการรับส่งจดหมายอิเล็กทรอนิกส์มากยิ่งขึ้น

1.3 ขอบข่ายงาน

โครงการนี้จัดทำขึ้นเพื่อ ให้บริการรับส่งจดหมายอิเล็กทรอนิกส์ โดยมีความสามารถ คือ

1. มีการลงทะเบียนสมาชิกใหม่ (create user)
2. มีระบบรักษาความปลอดภัยของผู้ใช้ โดยมีการ log in
3. ในส่วนของการส่ง e-mail มีบริการดังนี้
 - สามารถส่ง e-mail เป็นแบบ text ธรรมดา หรือส่งเป็นเว็บเพจแบบ html โดยผู้ใช้สามารถออกแบบเป็นแบบ html ได้

- มีการตรวจเช็ค e-mail address ที่ส่งไปว่ามีอยู่จริงหรือไม่ รวมถึง รายงานการส่งว่า ส่งถึงหรือไม่ มีปัญหาในการส่งหรือไม่
 - สามารถแนบ file attachment ไปในการส่ง e-mail แต่ละครั้งได้
4. ในส่วนของการรับ e-mail มีบริการดังนี้
- มีการแสดงฐานข้อมูลจดหมายในกล่องต่าง ๆ เช่น กล่องนำเข้า กล่องจดหมายส่งแล้ว กล่องขยะ
 - มีการบริการ forward e-mail ไปยัง e-mail address อื่นที่ต้องการ
 - สามารถบดล็อก e-mail ที่ไม่ต้องการรับได้
5. มีบริการในส่วนของ address book

1.4 กิจกรรมการดำเนินงาน

ตารางที่ 1.1 กิจกรรมการดำเนินงาน

กิจกรรม	พ.ช.	ช.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.
1. เขียนโครงสร้างการทำงาน	↔									
2. รวบรวมข้อมูลและเอกสาร		↔			↔					
3. ออกแบบและเขียนโปรแกรม				↔				↔		
4. ทดสอบและแก้ไขโปรแกรม						↔		↔		
5. ดำเนินงานบน Sever linux									↔	
6. ทดสอบการทำงานและแก้ไข									↔	↔
7. ทำโครงงานฉบับสมบูรณ์								↔		↔
8. ตรวจสอบโครงงานฉบับสมบูรณ์										↔

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถให้บริการ web based e-mail บน Internet ได้จริง
2. เพื่อให้เป็นแนวทางในการพัฒนา web based e-mail โดยใช้ PHP และ MySQL เป็นเครื่องมือในการพัฒนา โดยเน้นที่เป็น freeware

1.6 งบประมาณ

ค่าจัดทำรูปเล่ม	1,200 บาท
ค่าหนังสือ	1,000 บาท
ค่า Software (CD-ROM)	800 บาท
รวมค่าใช้จ่าย	3,000 บาท (สามพันบาทถ้วน)



บทที่ 2

ความรู้พื้นฐาน และทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานเกี่ยวกับอีเมล [1]

2.1.1 ความเป็นมา

ในปัจจุบัน "จดหมายอิเล็กทรอนิกส์" หรือที่นิยมเรียกกันว่า "อี-เมล" จัดว่าเป็นเครื่องมือในการติดต่อสื่อสารที่ได้รับความนิยมอย่างมาก เนื่องจากอีเมลเป็นเครื่องมือที่ช่วยให้เราสามารถติดต่อกับคนที่อยู่ห่างไกลถึง คนละซีกโลกได้ภายในเวลาเพียงไม่กี่วินาที โดยไม่ต้องเสียค่าใช้จ่ายมาก จากความสะดวกรวดเร็ว และความประหยัด ของอีเมลนี้ ทำให้มีการประมาณการกันว่าในวันหนึ่งๆคนทั่วโลกส่งอีเมลกันเป็นจำนวนหลายพันล้านฉบับ

การส่งอีเมลครั้งแรกเกิดขึ้นในราวปี ค.ศ. 1971 โดยวิศวกรชาวอเมริกันชื่อ นายเรย์ ทอมลินสัน ได้เริ่มคิดค้นระบบที่สามารถช่วยให้สามารถส่งจดหมายไปยังเครื่องคอมพิวเตอร์อื่นๆที่อยู่ในระบบอินเทอร์เน็ตได้ โดยอาศัยเครื่องหมาย "@" เป็นตัวระบุชื่อเครื่องคอมพิวเตอร์ปลายทางที่ต้องการจะส่งจดหมายไป และผู้รับไม่จำเป็นต้องเปิดเครื่องคอมพิวเตอร์อยู่ ณ ขณะนั้นก็ได้ เมื่อผู้รับเปิดเครื่องคอมพิวเตอร์ก็จะได้รับจดหมายนั้นทันที โดยต่างจากการส่งจดหมายทางไปรษณีย์ คือการส่งทางไปรษณีย์นั้นใช้เวลานาน และจดหมายอาจสูญหายได้ ซึ่งจดหมายอิเล็กทรอนิกส์ในยุคแรกๆ นั้นจะประกอบไปด้วยข้อความตัวอักษรเพียงอย่างเดียวเท่านั้น

ต่อมาได้มีการพัฒนาการส่งให้มีความสามารถเพิ่มขึ้นคือ สามารถส่งได้ทั้งรูปภาพ ภาพเคลื่อนไหว รวมทั้งเสียงประกอบต่างๆ โดยให้มีการแนบไฟล์ไปพร้อมกับการส่งอีเมลได้ด้วย ซึ่งเราเรียกว่า Attach File ทำให้การส่งจดหมายอิเล็กทรอนิกส์มีความน่าสนใจมากยิ่งขึ้น อีเมลจึงได้รับความนิยมไปทั่วโลกดังเช่นทุกวันนี้

2.1.2 ความหมาย

คำว่า E-mail (Electronics Mail) หมายถึง จดหมายอิเล็กทรอนิกส์ที่ใช้ติดต่อสื่อสารกันบนอินเทอร์เน็ต ดังนั้น ความหมายง่าย ๆ ของอีเมล ก็คือ เป็นเครื่องมือสำหรับติดต่อสื่อสาร ระหว่างกันโดยที่จากเดิม เราอาจจะใช้วิธีการส่งข้อความ ไปหาผู้อื่นด้วยการเขียนเป็นจดหมาย และส่งผ่านทางไปรษณีย์ แต่ในโลกของอินเทอร์เน็ต จะมีบริการที่เรียกว่า อีเมล ซึ่งสามารถทำการส่งข้อความต่าง ๆ ไปยังผู้รับปลายทาง (ที่ใช้บริการอีเมล) ได้ และในปัจจุบันนี้ ยังสามารถทำการแนบ ไฟล์เอกสาร ของคอมพิวเตอร์ หรือรูปภาพต่าง ๆ ไปอับอีเมลได้ด้วย จึงทำให้เพิ่มความ สะดวกสบายได้มากขึ้น ซึ่งก่อนที่คุณจะมีอีเมลใช้นั้นคุณจะต้องมี E-mail Address (ที่อยู่อีเมล) ของตัวเองเสียก่อน ถ้าคุณใช้บริการอินเทอร์เน็ตโดยการสมัครเป็นสมาชิกโดยตรงกับ ISP (ผู้ให้บริการอินเทอร์เน็ต) คุณก็จะได้รับแจ้งที่อยู่

อีเมลของตัวเองตั้งแต่สมัครสมาชิกเลยแต่ถ้า ISP ไม่ได้ให้มาคุณก็สามารถมีที่อยู่อีเมลได้ โดยขอได้ฟรี จากผู้ให้บริการอีเมลในอินเทอร์เน็ตทั้งที่ฟรีและไม่ฟรี

2.1.3 โครงสร้างและรูปแบบของชื่ออีเมลเบื้องต้น

ถ้าใครได้เคยเห็นรูปแบบ และชื่อของอีเมลมาบ้างแล้ว ลองมาทำความเข้าใจกับ ระบบการตั้งชื่ออีเมลกันก่อน สมมติว่าใครคนหนึ่ง บอกอีเมลของเขาว่า somchai@hotmail.com (อ่านออกเสียงว่า สมชาย-แอด-สอทเมท คอทคอม) เครื่องหมาย @ จะออกเสียงว่า "แอด" ซึ่งจะเป็นสิ่งที่ใช้คั่นอยู่ระหว่างชื่อและ server ของอีเมลนั้น ๆ เสมอ ชื่อของ server ที่ลงท้ายนี้ อาจจะเปลี่ยนไปได้ตามชื่อของ server ที่เปิดให้บริการอีเมลนั้น ๆ ด้วยเช่น อาจจะลงท้ายด้วย @yahoo.com @thailand.com @mail.com หรืออะไรก็ได้ ที่มีเปิดให้บริการ

2.1.4 รายละเอียดของอีเมล

รูปแบบของที่อยู่ของอีเมล จะประกอบด้วย

ชื่ออีเมล ตามติดด้วยเครื่องหมายแอด @ และโดเมนเนม (Domain Name)

ชื่ออีเมล หมายถึงชื่อ หรือรหัสประจำตัวผู้ใช้ ซึ่งมักจะเรียกว่า LoginName หรือ Account Name

โดเมนเนม หมายถึงชื่อเครื่องที่ใช้นั้นมีทะเบียนอยู่

เช่น ผู้ที่ทำงานที่จุฬาลงกรณ์มหาวิทยาลัย มีที่อยู่ดังนี้ Sayan.S@Chula.ac.th

ในการส่งอีเมลนั้น จดหมายจะประกอบด้วยส่วนต่างๆ ดังนี้

- **From** ถ้าเป็นจดหมายที่เราพิมพ์ออกไป ก็จะหมายถึงชื่ออีเมลของเราซึ่งโปรแกรมส่งอีเมล มักพิมพ์อัตโนมัติให้ หรือถ้าเป็นจดหมายที่เรารับเข้ามาก็จะเป็นชื่ออีเมลของผู้ส่งจดหมายมาให้
- **To** หมายถึง อีเมลผู้รับปลายทาง ผู้ส่งต้องรู้ชื่ออีเมลที่สะกดถูกต้องของผู้รับด้วย ไม่งั้นนั้นอาจมีการส่งจดหมายผิด หรือจดหมายอาจตีกลับโดยอัตโนมัติ ถ้าไม่มีผู้รับตามกำหนด นอกจากนั้นในการส่งจดหมายนั้นอาจส่งถึงหลายๆ คนพร้อมกันได้ในครั้งเดียวก็ได้
- **Subject** หมายถึง หัวเรื่องที่จะพูดคุยเป็นหัวข้อสรุปเนื้อความ ในจดหมาย
- **CC (Carbon Copy)** หมายถึงการสำเนาจดหมายเดียวกันนี้ ถึงชื่ออีเมลผู้อื่น ให้รับทราบด้วย
- **BCC (Blind Carbon Copy)** สำเนาจดหมายเดียวกันนี้ถึงชื่ออีเมลผู้อื่นให้รับทราบ คล้าย CC แต่จะซ่อนชื่ออีเมลผู้รับ ไม่ให้ใครเห็น
- **Attachment** หมายถึงการแนบเอกสารอื่น ไปพร้อมกับอีเมล เช่น ไฟล์รูปภาพ, ไฟล์เสียง เป็นต้น โดยจะระบุถึงชื่อไฟล์ที่ต้องการแนบไปกับอีเมลนี้
- **Body** หมายถึงพื้นที่ สำหรับเนื้อความของจดหมาย ให้ผู้ใช้พิมพ์ข้อความจดหมายได้ทันที
- **Signature** หมายถึง ข้อความสำหรับลงท้ายจดหมาย ซึ่งเจ้าของอีเมลนั้นๆ สามารถกำหนดขึ้นเองได้ โดยพิมพ์ไว้เพียงครั้งเดียวในส่วนของจิกเนเจอร์ มักนิยมใช้กับ คำขอบคุณ ชื่อผู้ส่ง ที่อยู่ เบอร์โทรศัพท์

แล้วโปรแกรมจะแทรกซิกเนเจอร์ ต่อท้ายเนื้อความให้โดยอัตโนมัติ

- New Message หรือ Compose New Mail เริ่มเขียนจดหมายใหม่
- Reply ใช้ในกรณีที่ต้องการตอบจดหมายฉบับที่กำลังอ่าน
- Reply to all ใช้ในกรณีที่ต้องการตอบจดหมายฉบับที่กำลังอ่านอยู่ แต่จดหมายฉบับนั้น มีผู้รับหลายคน และถ้าผู้อ่านต้องการตอบจดหมายถึงผู้รับทุกคน รีพลาเยอออก จะคัดลอกชื่อผู้อื่นที่เกี่ยวข้องกับอีเมลล์ฉบับนี้ ให้ได้รับเมลล์ตอบพร้อมกันด้วย
- FWD (Forward) ส่งจดหมายฉบับที่กำลังอ่าน ต่อไปให้ผู้อื่น
- Request-Read Recipient หมายถึงการเพิ่มคุณสมบัติในการส่งอีเมลล์ คือเมื่อผู้รับเปิดอ่านอีเมลล์แล้ว ให้มีอีเมลล์กลับมาบอกผู้ส่งว่าผู้รับได้รับแล้ว ให้โดยอัตโนมัติ
- Send-Receive ใช้ในกรณีที่ต้องการให้โปรแกรม ส่ง (Send) หรือ รับ (Receive)จดหมาย
- Inbox หมายถึงกล่องหรือที่สำหรับเก็บอีเมลล์ ที่มีผู้ส่งเข้ามา
- Outbox หมายถึงกล่องหรือที่เก็บอีเมลล์ ที่กำลังจะส่งออกไปหาผู้อื่น
- Sent Items หมายถึงกล่องหรือที่เก็บอีเมลล์ ที่เราได้เคยส่งออกไปหาผู้อื่นแล้ว
- Delete Items หมายถึงกล่องหรือที่เก็บอีเมลล์ ที่ได้ทำการลบทิ้งจาก Inbox แต่ยังคงเก็บสำรองไว้อยู่
- Drafts หมายถึงกล่องหรือที่เก็บอีเมลล์ สำหรับใช้เก็บอีเมลล์ต่าง ๆ ชั่วคราว ซึ่งอาจจะมีหรือไม่มีก็ได้
- Compose หรือ New Mail จะเป็นการส่งอีเมลล์ใหม่ ไปหาผู้อื่น
- Address Book หมายถึงสมุดรายชื่อของอีเมลล์ต่าง ๆ ที่เราสามารถเก็บไว้ เพื่อนำมาใช้งานได้ง่ายขึ้น
- > (เครื่องหมายมากกว่า) มักจะพบในส่วนเนื้อความของจดหมาย คือถ้าเป็นจดหมายที่ตอบโดยใช้ Reply โปรแกรมจะสำเนาข้อความเดิมมาไว้ในจดหมายที่กำลังจะส่งด้วย โดยมีเครื่องหมาย ">" นำหน้าทุกบรรทัด เพื่อให้รู้ว่าไม่ใช่ข้อความใหม่

2.1.5 ประเภทต่าง ๆ ของอีเมลล์

เว็บไซต์ที่ให้บริการ ฟรีอีเมลล์ มีอยู่มากมาย แต่ถ้าหากจะแยกประเภทของการใช้งาน สามารถแยกออกได้เป็น 2 แบบดังนี้

Web Base Mail เช่น อีเมลล์ของ hotmail.com, chaiyo.com หรือ email.in.th ซึ่งหากต้องการใช้งานอีเมลล์เหล่านี้ จะต้องใช้งานโดยผ่านทางหน้าเว็บเพจเท่านั้น ข้อดีคือ สามารถไปใช้เครื่องคอมพิวเตอร์ที่ไหนเปิดอ่านอีเมลล์นั้นก็ ได้ โดยการเข้าไปที่หน้าเว็บไซต์นั้น ไม่ต้องทำการตั้งค่าต่าง ๆ ให้ยุ่งยาก แต่อาจจะช้าและเสียเวลานาน ในการอ่านหรือรับส่งอีเมลล์ ข้อดีของบริการอีเมลล์แบบ Web Base Mail คือ ไม่จำเป็นต้องใช้โปรแกรมต่าง ๆ ในการอ่านหรือรับส่งอีเมลล์ ทำให้ไม่เปลืองพื้นที่ของฮาร์ดดิสก์

POP Mail เช่น yahoo.com จะมีบริการการอ่านอีเมลล์แบบ POP Mail ได้ด้วย ซึ่งโดยส่วนมากก็มักจะใช้งานในลักษณะของ เว็บเมลล์ได้ด้วย หากเราเลือกใช้งานอีเมลล์ที่มีบริการ POP Mail ก็จะทำให้

สามารถตั้งโปรแกรมสำหรับ รับ-ส่งอีเมลล์ทั่ว ๆ ไปเช่น Outlook ให้ทำการอ่านอีเมลล์แบบนี้ได้ และจะเป็นการสะดวกมากกว่าการใช้งานแบบ Web Base Mail มากครับ แต่ก็ต้องทำการตั้งค่าต่าง ๆ ของโปรแกรมที่ใช้รับ-ส่งอีเมลล์ก่อน จึงจะใช้งานได้ โดยที่เมื่อทำการต่อหรือเปิดโปรแกรม สำหรับการอ่านอีเมลล์ โปรแกรมจะต้องทำการดาวน์โหลด อีเมลล์ทั้งหมดมาเก็บไว้ใน ฮาร์ดดิสก์ ของเครื่องคอมพิวเตอร์ก่อน ทำให้เปลืองพื้นที่ของฮาร์ดดิสก์ไปบางส่วน แต่ว่าการอ่าน จะสามารถทำได้รวดเร็วและสะดวกกว่าการใช้ Web Base Mail

2.1.6 กลไกการรับ - ส่งอีเมลล์

โดยทั่วไประบบของการจัดส่งอีเมลล์นั้น จะประกอบไปด้วยเซิร์ฟเวอร์ (Server) ซึ่งมีหน้าที่หลักเป็น ผู้จดหมายให้กับผู้ใช้ และเซิร์ฟเวอร์ยังทำหน้าที่สำคัญ คือเป็นผู้รับส่งจดหมาย ระหว่างเครื่องคอมพิวเตอร์ต่างๆ โดยการส่งอีเมลล์นั้นเครื่องที่ส่ง และเครื่องที่รับแต่ละเครื่องจะต้อง มีระเบียบวิธีในการรับและส่งอีเมลล์เหมือนกัน คือผู้ส่งและผู้รับจะต้องใช้ระเบียบวิธีเดียวกัน จึงจะสามารถส่งอีเมลล์ถึงกันได้ ระเบียบวิธีในการรับและส่งนี้เรียกว่า โพรโตคอล (Protocol) คือ ในระบบการจัดส่งอีเมลล์นี้จะใช้โปรโตคอลชื่อ SMTP (Simple Mail Transfer Protocol) การรับส่งระหว่างเมลล์เซิร์ฟเวอร์โดยใช้โปรโตคอล SMTP นั้นตัวเซิร์ฟเวอร์ยังทำหน้าที่หลักที่สำคัญคือ เป็น Mail exchange หรือ Mail relay สำหรับองค์กรที่มีขนาดใหญ่ โดยโปรโตคอลนี้จะมีระบบการจัดการเมลล์ หรือเมลล์เกตเวย์ (Mail Gateway) ที่ต้องอาศัยการส่งต่อเพื่อ ให้เข้าสู่จุดหมายปลายทางได้ ซึ่งผู้ใช้จะทำตัวเองเป็นเมลล์ไคลเอ็นต์ (Mail Client) สำหรับติดต่อเพื่อ เชื่อมโยงกับเมลล์เซิร์ฟเวอร์ (Mail Server) จึงได้มีการพัฒนาโปรโตคอลในส่วนที่ผู้ใช้ติดต่อกับเซิร์ฟเวอร์ให้สามารถใช้งานได้อย่างง่ายดาย

โปรโตคอลที่นิยมใช้กัน โดยทั่วไปคือ POP3-Post Office Protocol และ IMAP (Internet Message Access Protocol) โปรโตคอลทั้งสองทำให้วิธีการติดต่อได้รับการพัฒนาผ่านเข้าไปยังตัวเว็บเบราว์เซอร์ (Web Browser) จนเป็นที่รู้จักกันดีในนามของ เว็บเมลล์ (Web mail) ซึ่งจะเน้นรูปแบบยูสเซอร์อินเทอร์เฟซ (User Interface) คือให้ผู้ใช้สามารถใช้งานได้อย่างง่ายดาย

2.2 ทฤษฎีของ Protocol ที่เกี่ยวข้องกับอีเมลล์ [5]

2.2.1 SMTP (Simple Mail Transfer Protocol)

เป็นโปรโตคอลแบบ TCP/IP ที่ใช้ในการรับส่ง email ซึ่งมีข้อจำกัดในเรื่องของความสามารถในการรับส่ง mail ว่ามันสามารถทำได้แบบเป็นคิวเท่านั้น จึงทำให้เกิดโปรโตคอลที่จะมาแก้ไขในเรื่องนี้ ซึ่งก็คือ POP กับ IMAP แต่อย่างไรก็ตาม แม้ว่า SMTP จะมีข้อจำกัดในการรับ mail แต่สำหรับการส่ง mail หลาย ๆ โปรแกรมก็ยังคงนิยมใช้ SMTP ในการส่ง mail อยู่เช่นเดิม

หรือพูดง่าย ๆ ก็คือ เครื่องคอมพิวเตอร์ที่มีหน้าที่คล้ายเจ้าหน้าที่ในที่ทำกรไปรษณีย์ ทำหน้าที่รับจดหมายจากเราเพื่อส่งต่อไปให้ผู้รับจดหมายที่เราเจ้าหน้าที่ของเรา

SMTP เป็นโปรโตคอลที่ใช้ส่ง e-mail จาก user agent ของผู้ส่งไปยัง MTP ของผู้ส่ง และส่งต่อไปยัง MTP เครื่องอื่น ๆ ที่เป็นจุดผ่านในการเชื่อมต่อไปยังผู้รับ โปรโตคอล SMTP จะทำงานร่วมกับโปรโตคอล TCP โดยใช้พอร์ต 25 ซึ่งคำสั่งต่าง ๆ ของ SMTP จะเป็นลักษณะเดียวกับ POP3 คือเป็น ASCII ลงท้ายด้วย Carriage Return และ Line Feed ส่วนข้อความที่ตอบกลับมานำหน้าด้วยเลข 3 หลัก เป็นสัญลักษณ์แสดงสถานะการทำงานของคำสั่งที่ได้รับ

เมื่อเริ่มต้นการติดต่อ SMTP จะกำหนดให้ user agent ของผู้ส่งต้องส่งคำสั่ง HELLO พร้อมกับรายละเอียดด้านผู้ส่งออกไป จากนั้นจะส่งคำสั่ง MAIL เพื่อแจ้งให้เซิร์ฟเวอร์เตรียมรับอีเมล ในส่วนของเซิร์ฟเวอร์เมื่อพร้อมที่จะรับอีเมลก็จะตอบรับกลับมาด้วยคำสั่ง OK จากนั้นที่ด้านส่งก็จะเริ่มส่งโดยใช้คำสั่ง RCPT เพื่อกำหนดอีเมลแต่ละฉบับที่ส่งไป ซึ่งการส่งข้อมูลของอีเมลจะถูกระบุด้วยคำสั่ง DATA

การส่งอีเมลของโปรโตคอล SMTP ได้จัดเตรียมคำสั่งอื่น ๆ ไว้เพื่ออำนวยความสะดวกและคล่องตัวในการทำงาน ซึ่งประกอบด้วยคำสั่ง VRFY เพื่อให้ด้านที่ส่งตรวจสอบรายชื่อว่าผู้ใช้นี้มีสิทธิ์ใช้งานอีเมลป็อกซ์นั้นหรือไม่ , คำสั่ง EXPN ใช้จัดการและตรวจสอบรายชื่อจากลิสต์รายชื่อ และคำสั่ง TURN ใช้สลับให้ไคลเอนต์ของผู้ส่งทำหน้าที่รับข้อมูลจากเซิร์ฟเวอร์แทน

เมื่อได้รับคำสั่งต่าง ๆ ของผู้ส่งแล้ว เซิร์ฟเวอร์จะมีหน้าที่ตรวจสอบความถูกต้องของคำสั่ง จากนั้นจึงทำงานตามคำสั่งและส่งผลตอบกลับมา ส่วนลักษณะของข้อมูลที่ตอบกลับ (reply message) นั้นจะเป็นข้อมูลที่อยู่ในรูปของ text ที่เป็น ASCII โดยจะประกอบด้วยตัวเลขนำหน้าข้อความ 3 หลัก ทำหน้าที่แสดงสถานะการทำงานของเซิร์ฟเวอร์ และเปลี่ยนสถานะการทำงานของโปรโตคอล SMTP ด้วย ถัดจากตัวเลขจะคั่นด้วยช่องว่างแล้วตามด้วยข้อความ ซึ่งปิดท้ายด้วยเครื่องหมาย Carriage Return และ Line Feed

ในการส่งอีเมลของโปรโตคอล SMTP นั้น จะใช้วิธีอ้างถึงเซิร์ฟเวอร์อื่น ๆ ตามแบบ DNS หรือ Domain Name System เช่นเดียวกับระบบอื่น ๆ ในอินเทอร์เน็ต และยังสามารถส่งอีเมลไปยังผู้รับคนเดียวหรือหลาย ๆ คนพร้อมกันได้

2.2.2 POP (Post Office Protocol)

เป็นโปรโตคอลที่ใช้รับ mail ซึ่งที่ได้รับความนิยมมากที่สุดเป็น POP version 3 หรือเรียกสั้น ๆ ว่า POP3 ซึ่งจะมีการทำงานแบบ Store-and-Forward ซึ่งไม่ควรนำไปสับสนกับ SMTP เพราะ POP จะใช้ในการรับ mail เท่านั้น ส่วน SMTP จะใช้ในการส่ง mail

หรือพูดง่าย ๆ ก็คือ เครื่องคอมพิวเตอร์ที่มีหน้าที่คล้ายบุรุษไปรษณีย์ ก็จะนำอีเมลที่มีคนส่งมาหาเรา ไปยัง คอมพิวเตอร์เครื่องอื่น หรือแม้แต่เครื่องคอมพิวเตอร์ของเราเอง

POP3 เป็นโปรโตคอลตัวแรกที่ถูกออกแบบมาเพื่อใช้รับอีเมล และเพื่อให้สนับสนุนการทำงานในแบบ Offline จึงมีการกำหนดรายละเอียดไว้ใน RFC 1939 ด้วย ซึ่งกลไกของ POP3 นี้จะทำงานในแบบ Offline โดยติดต่อเข้าไปยังเมลเซิร์ฟเวอร์แล้วดาวน์โหลดอีเมลทั้งหมดมาไว้ที่ user agent จากนั้น

จะลบอีเมลที่เซิร์ฟเวอร์นั้นทิ้งไป เพื่อป้องกันการดาวน์โหลดซ้ำ แต่ผู้ใช้จะทำงานแบบ Online กับเซิร์ฟเวอร์ไม่ได้ เนื่องจากการอ่านอีเมลจะดึงอีเมลที่เก็บไว้ใน user agent ขึ้นมาให้อ่านหลังจากที่ดาวน์โหลดมาเก็บไว้ ซึ่งในขณะนั้นอาจจะไม่ Online อยู่กับ Network ก็ได้

โปรโตคอลของ POP3 นี้จะทำงานในแบบไคลเอนต์เซิร์ฟเวอร์ คือมีโปรแกรม POP Server ในแม่ข่ายเซิร์ฟเวอร์ และ POP client ในเครื่องผู้รับ ซึ่งปกติจะฝังอยู่ในโปรแกรมที่เป็น user agent เลย โปรแกรมทั้งสองจะติดต่อกันโดยใช้ชุดคำสั่งที่เป็นรหัส ASCII คือเมื่อด้านที่รับทำคำสั่งก็จะทำงานตามคำสั่งนั้น แล้วตอบกลับมาโดยมีค่าเป็น +OK หมายถึงทำงานได้เรียบร้อย หรือ -ERR หมายถึงเกิดปัญหาขึ้นทำงานไม่ได้ ซึ่งในคำสั่งที่ต้องมีการตอบกลับและส่งข้อมูลกลับมา โดยประกอบด้วยข้อมูลหลาย ๆ บรรทัดนั้น POP3 จะให้บรรทัดสุดท้ายเป็นเครื่องหมายจุด (.) ตามด้วย Carriage Return และ Line Feed หมายถึงการสิ้นสุดชุดข้อมูล แต่ในกรณีที่ข้อมูลบรรทัดสุดท้ายมีข้อมูลที่เป็นจุดด้วย จะใช้เทคนิคที่เรียกว่า Character Stuffing เพื่อแก้ปัญหา โดยจะเติมจุดลงไปอีกหนึ่งตัว เพื่อเป็นตัวบ่งชี้ว่าข้อมูลนั้นเป็นจุด ซึ่งจะแตกต่างจากสัญลักษณ์แสดงการสิ้นสุดของข้อมูล

การทำงานของ POP3 จะทำงานร่วมกับโปรโตคอล TCP โดยทั่วไปจะใช้พอร์ต 110 ในการติดต่อ ขั้นตอนการทำงานของ POP3 ประกอบด้วย 3 สถานะคือ

1. สถานะขออนุมัติ(Authorization State) เมื่อเริ่มต้นติดต่อกับเซิร์ฟเวอร์จะเป็นการเข้าสู่สถานะการขออนุมัติ โดยไคลเอนต์จะต้องแจ้งชื่อผู้ใช้และรหัสผ่าน (password) เพื่อขออนุมัติจากเซิร์ฟเวอร์ก่อน โดยไคลเอนต์จะใช้คำสั่ง user เพื่อระบุชื่อผู้ใช้ หรือคำสั่ง pass เพื่อกำหนด Password แต่ในกรณีที่ชื่อและ password ถูกเข้ารหัสไว้ และไม่ได้เป็นค่า ASCII ทั่วไป ไคลเอนต์จะใช้คำสั่ง APOP ทำงานแทนคำสั่ง user และ pass
2. สถานะรับส่งรายการ(Transaction State) หลังจากที่ได้อนุมัติจากเซิร์ฟเวอร์แล้ว ก็จะเข้าสู่สถานะที่ใช้คำสั่งในการทำงานต่าง ๆ
3. สถานะปรับปรุงข้อมูล(Update State) เมื่อ user agent เลิกใช้งานด้วยคำสั่ง Quit ของ POP3 เซิร์ฟเวอร์จะเข้าสู่สถานะปรับปรุงข้อมูล เพื่อลบอีเมลที่ดาวน์โหลดเรียบร้อยแล้วออกไป จากนั้นก็จะเข้าสู่สถานะขออนุมัติใหม่โดยอัตโนมัติ เพื่อรอรับการดำเนินงานครั้งต่อไป

2.2.3 IMAP (Internet Message Access Protocol)

เป็นโปรโตคอลที่ใช้สำหรับรับ mail จาก server ซึ่งรุ่นที่ได้รับความนิยมมากที่สุด คือ IMAP version 4 หรือเรียกสั้น ๆ ว่า IMAP4 ซึ่งจะมีความสามารถในการเลือกเฉพาะ header กับ sender หรือสิ่งที่เราต้องการได้ นอกจากนี้ยังสามารถเลือก download เฉพาะ mail ที่เราต้องการได้ด้วย แต่ IMAP ก็ต้องอาศัยการติดต่อกับ server มากกว่า POP Imap มีความสามารถเหมือนกับ POP3 แต่มีความสามารถมากกว่าในด้านของการ Attach File

นอกจากนี้ IMAP ยังสามารถรองรับการทำงานได้ทั้งแบบ Offline, Online และ Disconnected อีกด้วย ดังนั้นประโยชน์ที่ได้รับจาก IMAP ก็คือ หากผู้ใช้มีอีเมลแอดเดรสเพียงชื่อเดียว แต่มีเครื่องที่ใช้งานอยู่หลายเครื่อง ก็จะสามารถเลือกดาวน์โหลดเฉพาะอีเมลที่ต้องการมาเก็บไว้ที่เครื่องใดก็ได้ แต่ถ้าเป็น POP3 การดาวน์โหลดจะต้องทำพร้อมกันหมดทุกอีเมล ดังนั้น IMAP จึงเป็นโปรโตคอลที่สามารถใช้งานกับสายสื่อสารที่มีความเร็วต่ำได้เป็นอย่างดี

การทำงานของ IMAP นี้จะเหมือนกับโปรโตคอลอื่น ๆ โดยทำงานร่วมกับ TCP ใช้พอร์ตหมายเลข 143 และจะแบ่งเป็นสถานะต่าง ๆ ออกเป็น 4 สถานะ โดยแต่ละสถานะจะมีวัตถุประสงค์และคำสั่งที่ใช้งานแตกต่างกัน โดยมีรายละเอียดดังนี้

สถานะก่อนอนุมัติ (non-authenticated state) เป็นสถานะที่กำลังรอให้ไคลเอนต์ติดต่อเข้ามาเพื่อขออนุมัติใช้ ดังนั้นในด้านไคลเอนต์จะต้องแจ้งชื่อ Login ของ Mail Server นั้นและ password ด้วยคำสั่ง Login หรือ Authenticate ก่อนจึงเริ่มใช้งานได้ จากนั้นจึงเปลี่ยนไปเป็นสถานะได้รับการอนุมัติ

สถานะได้รับการอนุมัติ(authenticated state) เป็นสถานะที่สามารถใช้คำสั่งต่าง ๆ ที่เกี่ยวกับการเลือกและใช้งานเมลบ็อกซ์ เช่น select เพื่อเลือกเมลบ็อกซ์ หรือคำสั่ง create เพื่อสร้างเมลบ็อกซ์ เป็นต้น ในการเลือกเมลบ็อกซ์ด้วยคำสั่ง select หรือ examine นี้จะเป็นการเปลี่ยนไปเป็นสถานะการเลือกเมลบ็อกซ์ สถานะเลือกเมลบ็อกซ์(selected state) เป็นสถานะที่จะเข้าใช้งานอีเมลในแต่ละเมลบ็อกซ์หลังจากที่เลือกเมลบ็อกซ์ไว้แล้วในสถานะก่อนหน้านี้

สถานะเลิกใช้งาน(Logout State) เมื่อต้องการเลิกใช้งาน หรือสิ้นสุดการทำงานของ IMAP จะเข้าสู่สถานะการเลิกใช้งาน โดยใช้คำสั่ง Logout

จากสถานะทั้ง 4 นี้ไม่จำเป็นต้องเรียงต่อกันเสมอไป บางครั้งอาจจะมีการทำงานข้ามจากสถานะหนึ่งไปยังอีกสถานะหนึ่งได้

คำสั่งต่าง ๆ ของ IMAP ที่ไคลเอนต์ส่งให้กับเซิร์ฟเวอร์นั้น จะต้องระบุรหัสหน้าคำสั่งนั้น ๆ ด้วย หรือที่เรียกว่า แท็ก (tag) และในกรณีที่เซิร์ฟเวอร์ตอบผลการทำงานของคำสั่งนั้น ๆ กลับมา ก็ต้องระบุแท็กหมายเลขเดียวกันด้วย ซึ่งในบางกรณีก็อาจไม่จำเป็นต้องมีแท็กนำหน้าคำสั่งก็ได้ ซึ่งเรียกว่า Untagged response หรือ Asterisk Tag (*) ส่วนคำตอบของเซิร์ฟเวอร์ที่เป็นผลการทำงานตามคำสั่งต่าง ๆ ของไคลเอนต์นั้น จะประกอบไปด้วยคำสั่งทั้งหมด 5 แบบ คือ

OK	แสดงว่าผลการทำงานของคำสั่งนั้น ๆ สำเร็จและเรียบร้อยแล้ว
NO	แสดงว่าการทำงานตามคำสั่งนั้นไม่สำเร็จ
BAD	แสดงว่าคำสั่งที่ส่งไปไม่ถูกต้อง หรือมีพารามิเตอร์ไม่ถูกต้อง
PREAUTH	แสดงว่าไม่จำเป็นต้องใช้คำสั่ง Login เพราะเคยมีการเข้าสู่สถานะที่ได้รับการอนุมัติ (authenticated state) แล้ว
BYE	แสดงว่าเซิร์ฟเวอร์ได้จบการทำงานไปแล้ว

2.3 PHP กับ Web based E-mail

PHP (PHP: Hypertext Preprocessor) [4] เป็นโปรแกรมประเภท server side scripting ประเภทหนึ่ง คล้ายๆ กับ SSI และ ASP ซึ่งจะช่วยให้เราสร้าง Website สามารถดึงข้อมูลจากฐานข้อมูลได้ และการทำงานของ PHP จะเหมือนกับการรวมเอาความสามารถของ ภาษา C,Java,Perl เข้าไว้ด้วยกันหรือกล่าวอีกในหนึ่ง PHP คือการนำเอาข้อดีของ ภาษา C,Java,Perl มารวมกันนั่นเอง

2.3.1 การ Login และ Logout [3]

2.3.1.1 คุกกี (Cookies)

คุกกี้ คือ TextFile ขนาดเล็ก ที่มีขนาดในแต่ละชุดได้ไม่เกิน 4 KB โดยถูกจัดเก็บอยู่บนเครื่องคอมพิวเตอร์ด้านฝั่ง Client (ผู้ใช้โปรแกรม Browser) โดยภายในคุกกี้ แต่ละตัวจะมีส่วนประกอบคือ ชื่อคุกกี้(Name) ค่าคุกกี้(Value) และวันหมดอายุ(Expiry Date) พร้อมกับข้อมูลเกี่ยวกับ Host และ Path

คุกกี้ถูกสร้างขึ้นเมื่อไร

โดยปกติแล้วคุกกี้จะถูกกำหนดมาพร้อมกับข้อมูลที่ Web Server ส่งมาให้กับ Client โดยจะอยู่ในส่วนของ HTTP Header ดังตัวอย่างรายละเอียดในส่วนของ HTTP Header ดังนี้

```
HTTP/1.1 200 OK
Date: Mon, 17 June 2002 22:35:11 GMT
Server: Apache/1.3.20(Win32) PHP/4.1.2
Set-Cookie: Counter=1; expires=Monday,17-June-02 22:35:11 GMT;path=/;domain=server.com
Connection: Close
Content-Type: text/html
```

เมื่อฝั่ง Client ได้รับ HTTP Header มา Client จะทำการตรวจสอบหาประโยชน์ที่ขึ้นต้นด้วยคำว่า Set-Cookie ใน HTTP Header นั้น ถ้ามีประโยชน์ดังกล่าว Client จะทำการอ่านข้อมูลต่างๆ ใน HTTP Header จากนั้นจึงนำข้อมูลที่ได้นำมาสร้างเป็นแฟ้มข้อมูลคุกกี้เก็บไว้ที่เครื่องของฝั่ง Client เอง เพื่อใช้ในการติดต่อร้องขอข้อมูลกับ เครื่อง Web Server ในครั้งใหม่ ต่อไป

การใช้โปรแกรม Internet Explorer อ่านเอกสาร HTML แฟ้มข้อมูลคุกกี้ต่างๆจะถูกเก็บไว้ที่ PATH = C:\WINDOWS\Cookies และเมื่อผู้ใช้ได้รับคุกกี้ คุกกี้ทั้งหมดจะถูกเก็บไว้ที่ C:\WINDOWS\Cookies ทั้งหมด จากนั้น เมื่อผู้ใช้เปิดอ่าน หน้าเว็บเพจนั้นอีกครั้ง Cookie Client จะทำการตรวจสอบว่า คุกกี้ที่หมดอายุหรือไม่(โดยเมื่อคุกกี้หมดอายุ คุกกี้จะถูกลบออกไปโดยอัตโนมัติ) จากนั้นจึงตรวจสอบอีกว่า Domain และ Path ของเว็บเพจนั้น ตรงกับค่าที่เก็บไว้ในคุกกี้หรือไม่ ถ้าตรงกัน Client จะส่งค่าในคุกกี้ไปให้กับWebServer อีกครั้ง

จะเห็นได้ว่าการทำงานของคุกกี้ เปรียบเสมือนว่าเครื่อง WebServer ได้สร้าง Connection ติดต่อกับเครื่อง Client ตลอดเวลานั่นเอง

PHP กับ Cookie

การสร้าง Cookie

การสร้างคุกกี้ ใน PHP เราจะเรียกใช้ฟังก์ชัน `setcookie()` โดยมีข้อกำหนดว่าจะต้องวาง function ไว้ก่อนแท็ก `<HTML>` หรือ `<HEAD>` เสมอ มิฉะนั้นจะเกิดข้อผิดพลาดขึ้น ดังนี้

Warning: Cannot add header information - headers already sent by (output started at c:\apache\htdocs\mail\cookie.php:19) in c:\apache\htdocs\mail\cookie.php on line 21

รูปแบบการใช้ฟังก์ชัน `setcookie`

```
setcookie(str name [,str value[,int expire[,str path[,str domain[,int secure]]]])
```

`str name` คือ ข้อมูลชนิด `string` สำหรับกำหนดชื่อตัวแปรคุกกี้

`str value` คือ ข้อมูลชนิด `string` สำหรับกำหนดค่าของตัวแปรคุกกี้

`int expire` คือ ข้อมูลชนิด `integer` สำหรับกำหนดวันหมดอายุของคุกกี้ โดยมีหน่วยเป็นวินาทีเช่น ถ้าต้องการสร้าง cookie ให้มีอายุ 1 วัน ให้ระบุดังนี้ `time()+3600*24` เพราะว่า 1 ชั่วโมงมี 3600 วินาที (60*60) และ 1 วันมี 24 ชั่วโมง จึงเอา 24 คูณเข้าไปทำให้เท่ากับ 1 วัน

`str path` คือ ข้อมูลชนิด `string` สำหรับกำหนดค่า Path และ URL เช่น `www.thaiphpdev.com/mail/` ถ้าไม่ระบุจะถือเอา Path ของเอกสารนั้นเป็นหลัก

`str domain` คือ ข้อมูลชนิด `string` สำหรับกำหนดชื่อ Domain เช่น `thaiphpdev.com` ใช้สำหรับให้ Client ตรวจสอบว่า URL ที่ร้องขอไปยัง WebServer ได้ถูกต้อง เช่น `thaiphpdev.com` จะตรงกับ `mail.thaiphpdev.com` เพราะอยู่ภายใต้ Domain เดียวกัน โดยปกติถ้าไม่กำหนดจะถือเอา Host ที่ส่งคุกกี้มาแทน (ก่อน Client จะทำการส่งคุกกี้ให้กับ Webserver cookie ของ Client จะทำการตรวจสอบ Domain และ Path ก่อนทุกครั้ง)

`int secure` คือ ค่าที่กำหนดเกี่ยวกับความปลอดภัยของข้อมูล โดยถ้ากำหนดเป็น 1 เครื่อง Client จะทำการติดต่อในรูปแบบ HTTPS Connection เพื่อส่งคุกกี้ไปยัง WebServer ทั้งต้องสนับสนุนมาตรฐาน SSL (Secure Sockets Layer) ด้วย แต่โดยปกติถ้าเราไม่ได้กำหนดจะมีค่าเป็น 0 คือ เครื่อง Client จะทำการติดต่อในรูปแบบของ HTTP Connection หรือ แบบปกตินั่นเอง

การอ้างถึงตัวแปรของ Cookie

ในกรณีที่ ต้องการใช้ประโยชน์จากตัวแปรคุกกีที่ได้ประกาศไปแล้ว เราสามารถทำได้ 4 วิธี ดังตัวอย่างดังต่อไปนี้

กำหนดให้ Setcookie("username","off",time()+3600*24)

```
echo"$HTTP_COOKIE<br>"; //พิมพ์ ชื่อตัวแปร = ค่าตัวแปร เช่น username=off
echo getenv("HTTP_COOKIE")."<br>"; //พิมพ์ ชื่อตัวแปร = ค่าตัวแปร เช่น username=off
echo"$username<br>"; //พิมพ์ ค่าตัวแปร username คือ off
echo"$HTTP_COOKIE_VARS[username]<br>"; //พิมพ์ ค่าตัวแปร username คือ off
```

การลบคุกกี้

หากต้องการลบตัวแปรคุกกีทั้งหมดที่เราจะต้องใช้ฟังก์ชัน Setcookie() แล้วระบุชื่อของคุกกีที่ต้องการจะลบ ดังตัวอย่างนี้
ถ้าเราต้องการลบค่าคุกกี ตัวแปรที่ชื่อ username

```
Setcookie("username");
```

2.3.1.2 Session

เมื่อเราเปิด Browser ขึ้น เพื่อร้องขอเพิ่มเอกสารจาก WebServer หากเพิ่มเอกสารดังกล่าวเป็นเพิ่มเอกสารธรรมดาทั่วไปแล้ว การทำงานก็ถือว่าเป็นธรรมดาปกติทั่วไป แต่หากว่า เพิ่มเอกสารดังกล่าวที่ได้ร้องขอไปยัง WebServer นั้นมีการระบุโค้ดในการตั้ง Start Session ขึ้น โปรแกรม Browser ของเราจะได้รับหมายเลขประจำตัวที่ส่งมาจาก WebServer ขึ้นมาทันที โดยค่าหมายเลขนี้เรียกว่า Session ID ซึ่งถูกเข้ารหัสอยู่ในรูปแบบของ MD5 ทำให้ Client ที่ร้องขอเอกสารนั้น จะได้รับค่า Session ID ที่ไม่ซ้ำกัน

เมื่อ Client แต่ละตัวได้รับ Session ID แล้วเราก็สามารถอ้างอิงค่าของ Session ID ไปได้ยังทุกๆ เว็บเพจ จนกว่า Client นั้นๆ จะทำการปิดหน้าต่างโปรแกรม Browser ลง เราก็ไม่สามารถอ้างอิงค่า Session ID ของ Client ดังกล่าวได้อีกต่อไป

นอกจาก Client จะทำการเปิดโปรแกรม Browser ขึ้นมาใหม่ ก็จะส่งผลให้หน้าต่างของโปรแกรม Browser ที่ถูกเปิดขึ้นมาใหม่นั้นได้รับค่า Session ID ค่าใหม่นั้นเอง

ด้วยความสามารถของ Session เราสามารถนำไปใช้ในการพัฒนาเว็บแอปพลิเคชันต่างๆ เช่น ร้านค้า ออนไลน์ หรือการตรวจสอบผู้ใช้ที่ออนไลน์อยู่ขณะนั้น เป็นต้น

ดังนั้น Session จะต่างกับ Cookie ตรงที่การหมดอายุของ Session จะขึ้นอยู่กับ การปิดโปรแกรม Browser ของ Client แต่ถ้าเป็น Cookie การหมดอายุจะขึ้นอยู่กับ การตั้งเวลาหมดอายุของ Cookie เป็นสำคัญ

PHP กับ SESSION

การกำหนดค่าไฟล์ Config (php.ini) เพื่อให้ PHP รองรับ Session

สำหรับ WebServer ที่เป็น Linux ปกติเมื่อติดตั้ง PHP เสร็จ เราสามารถใช้งาน Session ได้ทันที เนื่องจาก ไฟล์ php.ini มีการกำหนดค่าของ session.save_path=/tmp มาแล้ว แต่เมื่อได้ทำการติดตั้งบน Windows เราจะต้องทำการเปลี่ยนแปลงค่า session.save_path ให้ชี้ไปยัง Directory ที่ต้องการเก็บข้อมูลระบบ Session เช่น c:\winnt\temp หรือ c:\windows\temp

การเริ่ม Start Session ของ PHP

ในการทำงานกับ Session ทุกครั้ง เราต้องเรียกใช้ฟังก์ชัน session_start() ก่อนเสมอ เพื่อเป็นการเริ่มต้นการทำงานของ Session แต่ถ้าเราต้องการให้ Session ทำการกำหนดค่าเริ่มต้นโดยอัตโนมัติก็ให้เข้าไปแก้ไขเพิ่มข้อมูล php.ini ดังนี้

```
;Initialize session on request startup.  
session.auto_start=1
```

หมายเหตุ: ในการกำหนดค่า Session.auto_start=1 เราจะต้องแน่ใจว่าต้องการที่จะทำการ Start Session ในทุกๆ เเพจที่เป็นเอกสาร PHP

หลังจากทำการ Start session เรียบร้อยแล้ว เราสามารถตรวจสอบ Session ID ที่ได้รับด้วยการเรียกใช้งานฟังก์ชัน session_id() ดังตัวอย่าง

```
<?php  
    echo"คุณมี Session-ID เท่ากับ ".session_id();  
?>
```

ผลลัพธ์ที่ได้คือ

```
คุณมี Session-ID เท่ากับ c6fa24af21af92317d01994d092955d0
```

เมื่อ Script ถูกรันผ่านโปรแกรม Browser ในครั้งแรก ทางฝั่งของ WebServer จะส่งค่าผ่าน Header มาให้กับ Client เพื่อสร้างคุกกี้เก็บไว้ที่ Client ดังตัวอย่างเช่น

HTTP/1.1 200 OK

Date: Mon, 17 June 2002 22:35:11 GMT

Server: Apache/1.3.20(Win32) PHP/4.1.2

Set-cookie: PHPSESSID= c6fa24af21af92317d01994d092955d0; path=/

Connection: Close

Content-Type: text/html

ดังนั้น เราสามารถอ้างอิงค่า Session ID ผ่านทางตัวแปร \$PHPSESSID ได้ทันที ส่วนทางฝั่งของ WebServer จะสร้างแฟ้มข้อมูลที่ขึ้นต้นด้วย sess_ และตามด้วย Session ID ตัวอย่างเช่น

```
sess_c6fa24af21af92317d01994d092955d0
```

กำหนดตัวแปร Session ด้วยฟังก์ชัน session_register()

หากเราต้องการสร้างตัวแปรธรรมดาให้เป็นตัวแปรแบบ Session เพื่อสามารถอ้างอิงตัวแปร Session ได้ทุกเว็บเพจ เราจะต้องเรียกใช้ฟังก์ชัน session_register() ดังตัวอย่างเช่น

```
<?php
session_start();
session_register("user");
session_register("password");
session_register("mailserver");
$user="damrong";
$password="123456789";
$mailserver="mail.server.com";
echo session_encode();
?>
```

ผลลัพธ์ที่ได้คือ

```
user|s:7:"damrong";password|s:9:"123456789";mailserver|s:15:"mail.server.com";
```

การกำหนดตัวแปรแบบ Session ในตัวอย่างด้านบน จะเห็นได้ว่า การใช้ฟังก์ชัน session_register() จะขึ้นอยู่กับจำนวนของตัวแปรที่ต้องการกำหนดเป็นตัวแปรแบบ Session ซึ่งจะยุ่งยากและสิ้นเปลือง

ดังนั้น ขอแนะนำอีกวิธีหนึ่ง คือ การกำหนดตัวแปรแบบ Session ในรูปของอาร์เรย์แบบจับคู่ โดยเราจะสร้างตัวแปร Session ขึ้นมาเพียงตัวเดียว แล้วกำหนดค่าตัวแปรด้วยอาร์เรย์แบบจับคู่ ลงศึกษา ดังตัวอย่างต่อไปนี้

```
<?php
session_start();
session_register("SESSION");
$SESSION["user"]="damrong";
$SESSION["password"]="123456789";
$SESSION["mailserver"]="mail.server.com";
echo session_encode();
?>
```

ผลลัพธ์ที่ได้คือ

```
SESSION|a:3:{s:4:"user";s:7:"damrong";s:8:"password";s:9:"123456789";s:10:"mailserver";s:15:"mail.server.com";}
```

หลังจากที่เรากำหนดตัวแปรแบบ Session กันแล้ว ต่อไปเราจะนำตัวแปรแบบ Session ดังกล่าว ออกมาแสดง ซึ่งมีรูปแบบการอ้างอิงเหมือนกับตัวแปร โดยทั่วไป ดังตัวอย่างต่อไปนี้

1. การเข้าถึงตัวแปรแบบ Session(ธรรมดา)

```
<?php
session_start();
echo"ตัวแปร user =$user<br>\n";
echo"ตัวแปร password =$password<br>\n";
echo"ตัวแปร mailserver=$mailserver<br>\n";
?>
```

2. การเข้าถึงตัวแปรแบบ Session(อาร์เรย์แบบจับคู่)

```
<?php
session_start();
echo"ตัวแปร SESSION[user] =$SESSION[user]<br>\n";
echo"ตัวแปร SESSION[password]=$SESSION[password]<br>\n";
```

```
echo"ตัวแปร SESSION[mailserver]=$SESSION[mailserver]<br>\n";
?>
```

การยกเลิกตัวแปร Session และลบค่าตัวแปรทั้งหมด

หากเราต้องการยกเลิกหรือล้าง Session ทิ้ง ให้เรียกใช้ฟังก์ชัน session_destroy() โดยไม่ต้องกำหนดค่าอาทิเวเมนต์ใดๆ ทั้งสิ้น หากว่าเราต้องการลบค่าตัวแปร Session ทิ้งด้วย ให้เรียกใช้งานฟังก์ชัน session_unset() ดังตัวอย่างเช่น

```
<?php
session_start();
session_register("counter");
$count=1;
session_destroy();
echo"$count"; //ผลลัพธ์ที่ได้จะพิมพ์ 1
?>
```

สาเหตุที่ยังพิมพ์ค่า 1 เพราะว่า เราล้างเฉพาะ Session แต่ยังไม่ได้ลบตัวแปรทิ้ง แต่ถ้าเป็นตัวอย่างนี้

```
<?php
session_start();
session_register("counter");
$count=1;
session_unset();
session_destroy();
echo"$count"; //ผลลัพธ์ที่ได้จะพิมพ์ค่าว่าง โดยไม่แสดงอะไรเลย
?>
```

การตรวจสอบตัวแปร Session ว่าถูก Register ไว้หรือไม่

หากเราต้องการตรวจสอบ Session ว่าถูก Register ไว้หรือไม่ให้เรียกใช้งานฟังก์ชัน session_is_registered() ตามด้วยชื่อตัวแปร Session ที่ต้องการตรวจสอบ ตัวอย่างเช่น

```
if(session_is_registered("counter"))
{echo"ค่าตัวแปร Counter มีค่าเท่ากับ $count";}
```

2.3.2 การส่ง e-mail [1]

2.3.2.1 การส่งเมลโดยใช้ฟังก์ชัน mail() ของ PHP

การส่ง e-mail ไปยังบุคคลอื่นๆ ให้เรียกใช้งาน function mail() โดยเฉพาะถ้าเป็นระบบปฏิบัติการ Unix จะมีตัวโปรแกรม Sendmail ไว้คอยบริการ รับ-ส่ง e-mail ทำให้ php เข้าไปถึงความสามารถตรงจุดนี้มาใช้งานได้โดยอัตโนมัติ ซึ่งจะแตกต่างจาก asp ที่ต้องไปหา mail component มา register ถึงจะทำงานได้ นี่ถือเป็นข้อดีของ php

แต่ถ้าเราใช้ระบบปฏิบัติการ Windows ให้เข้าไประบุค่า SMTP ในไฟล์ php.ini ให้ชี้ไปยัง SMTP ของ ISP ที่เรา online เข้า Internet อยู่ ซึ่งแต่ละ ISP จะมีชื่อ SMTP ที่แตกต่างกัน เช่น

[mail function]

; For Win32 only

SMTP = griffon.ccc.chula.ac.th

ตารางที่ 2.1 ตัวอย่างแสดง SMTP ของ ISP (บางแห่ง)

ISP	SMTP
A-Net	main.a-net.net.th
Asia Access	bkk1.asiaaccess.net.th
Asia Infonet	classic.asianet.co.th
CS Internet	mail.cscoms.com
Data Line Thai	mail.linethai.co.th
FarEast Internet	mail.fareast.net.th
Idea Net	mail.idn.co.th
Infonews	smtp.infonews.co.th
INET	mozart.innet.co.th
KSC	ksc.th.com
Loxinfo	bkk3.loxinfo.co.th
Roynet	mbox.roynet.co.th

รูปแบบทั่วไปในการส่ง e-mail โดยใช้ function ใน php

mail(อีเมลผู้รับ , หัวเรื่อง Subject , ข้อความจดหมาย [,header])

- หากต้องการส่ง e-mail ให้ผู้รับหลายๆ คน ให้ใช้เครื่องหมายคอมมา , คั่นระหว่างเบอร์ e-mail
- สำหรับผู้รับ , หัวเรื่อง , ข้อความ ก็ปกติเหมือนกับที่เราส่ง e-mail ทั่วๆ ไป แต่ header เป็นส่วนขยาย
- ในเนื้อความจดหมาย (Message) หากต้องการขึ้นบรรทัดใหม่ ให้ใช้ \n

ตัวอย่างการส่งอีเมลล์โดยใช้บรรทัดเดียว

```
<?php
mail("pop@hotmail.com","test send mail","test send e-mail by php function","From:
webmaster@popmail.com");
?>
```

สำหรับตัวอย่างนี้แสดงการส่งอีเมลล์แบบง่าย ๆ ใช้เพียงบรรทัดเดียวก็สามารถส่งได้ แต่ตามปกติแล้วการใช้งานจริงหรือถ้าเป็นข้อความยาว ๆ ควรเก็บค่าไว้ในตัวแปร แล้วค่อยเรียกใช้ฟังก์ชันส่งเมลล์

ตัวอย่างต่อไปนี้จะแสดงการส่งเมลล์โดยเก็บไว้ในตัวแปร

```
<?php
$email="you@hotmail.com";
$from="me@popmail.com";
$subj="test send mail";
$msg="ทดสอบการส่ง e-mail โดยการเก็บค่าไว้ในตัวแปร";
mail($email,$subj,$msg,"From: $from");
?>
```

การตรวจสอบว่าส่งอีเมลถึงหรือไม่

การส่งอีเมลล์ด้วยฟังก์ชัน mail() ใน php ค่าที่ได้ออกมาจะเป็น true ดังนั้น การที่เราจะตรวจสอบว่าอีเมลล์ที่ส่งถึงหรือไม่ ทำได้โดยใช้ if แล้วใช้ echo เพื่อรายงานผลการส่งเข้ามาช่วย ดังตัวอย่างต่อไปนี้

```
if(mail(ผู้รับ , subject , ข้อความ , "From: ผู้ส่ง")) {
    echo "ส่งอีเมลล์เรียบร้อยแล้ว"; }
else {
    echo "ส่งอีเมลล์ไม่ได้";
}
```

การกำหนด header ลักษณะต่าง ๆ

e-mail header เป็นข้อมูลส่วนขยายเพิ่มเติมสำหรับการส่ง e-mail เช่น From: สำเนาด้วย CC:, BCC: ตัวอย่างต่อไปนี้เป็นการแสดงการใช้ header ต่าง ๆ ได้แก่

- สำเนาอีเมล CC: และสำเนาลับ BCC:
- การกำหนดระดับความสำคัญของอีเมล (email priority)
- การกำหนด return email ในกรณีที่ไม่มีผู้รับ ให้ตีกลับเมลมาที่ไหน

```
<?php
```

```
$email="pp@popmail.com ,";
```

```
$email="aa@popmail.com";
```

```
$from="Sys Admin<admin@popmail.com>";
```

```
$subj="ทดสอบการส่งอีเมลแบบมี header";
```

```
$msg="การส่งอีเมลแบบมี header";
```

```
$headers="From: $from\n";
```

```
$headers.="X-Priority: 1\n";
```

```
$headers.="Return-Path: <master@popmail.com>\n";
```

```
$headers.="cc: bb@popmail.com\n";
```

```
$headers.="bcc: dd@popmail.com , ss@popmail.com\n";
```

```
if(mail($email, $subj, $msg, $headers)) {
```

```
    echo "ส่งเมลเรียบร้อยแล้ว"; }
```

```
else {
```

```
    echo "ส่งเมลไม่ได้";
```

```
}
```

```
?>
```

การส่งเมลแบบ HTML

ข้อได้เปรียบของการส่งเมลแบบ HTML

- แบบ html มีความน่าสนใจมากกว่าแบบ text
- แบบ html มีความสวยงามมากกว่าแบบ text
- มีลูกเล่นมากกว่า

หลักการ คือ การกำหนด MIME (Multipurpose Internet Mail Extensions) ใน header ให้ Content-Type เป็นแบบ text/html เพราะถ้าไม่กำหนดจะถือว่าเป็นแบบ text/plain หลังจากกำหนด Content-Type เป็น text/html แล้ว เนื้อความจดหมายที่เขียนต้องเขียนเป็นแบบ html ด้วย สรุปหลักเกณฑ์ดังนี้

- กำหนด Content-Type : text/html ไว้ใน header และถ้าส่งเมลภาษาไทย ควรจะกำหนด charset=windows-874 ไว้ด้วย
- หลัง header ทุกส่วนต้องลงท้ายด้วย \n
- ข้อความจดหมายให้ใช้รูปแบบ HTML
- การกำหนดลิงค์ให้คลิกได้ในเมล ต้องเขียนให้เต็มรูปแบบ HTML เช่น yourdomain.com สำหรับลิงค์คลิก email address ก็เช่นเดียวกัน ต้องเขียนให้ครบรูปแบบ เช่น

รูปแบบการส่งเมลแบบ HTML

```
<?php
```

```
กำหนดข้อมูลผู้รับ $email , ผู้ส่ง $from , หัวเรื่อง $subject
$msg="ข้อความจดหมายเขียนในรูปแบบ HTML <h2>..</h2><hr><br>";
# กำหนดข้อมูล header Content-Type และอื่น ๆ ตามต้องการ
$headers= "From: $from\n";
$headers.="Content-Type: text/html; charset=windows-874\n";
$headers.="cc: aa@mymail.com\n";
mail($email, $subject, $msg, $headers);
?>
```

2.3.2.2 การส่งเมลโดยติดต่อ Protocol SMTP

การส่ง e-mail โดยการติดต่อทาง socket

ข้อดีของการเชื่อมต่อระดับซ็อกเก็ต

- สามารถตรวจเช็คเบอร์ (email address) ว่ามีอยู่จริงหรือไม่
- ตรวจเช็คเฉพาะ โดเมน (domain) ของอีเมลว่ามีอยู่จริงหรือไม่
- ออฟโหลดไฟล์เข้าเซิร์ฟเวอร์โดยไม่ต้องใช้ฟอร์มเลือกไฟล์
- ดึงเว็บเพจจากเว็บไซต์อื่นมาแสดงร่วมกับเว็บเพจของเว็บไซต์เรา
- เช็คอีเมลและเปิดอ่านอีเมลแบบ pop3 จากเซิร์ฟเวอร์ใด ๆ ก็ได้

การส่งอีเมลโดยใช้ Telnet

Telnet เป็นโปรแกรมด้าน Client ที่สามารถเชื่อมต่อกับเซิร์ฟเวอร์หรือโฮสต์ในระดับ port ได้ ตัวอย่างต่อไปนี้จะเป็นการใช้ Telnet ส่งเมลให้กับ Mail Server (SMTP) ที่ติดตั้งอยู่ในเครื่อง web server จำลองของเราเอง และส่งให้ hotmail.com ใน Internet จริง หากต้องการส่งโดยไม่ online ให้ติดตั้งโปรแกรม Mail Server

ขั้นตอนในการติดต่อทาง Telnet ใน Windows 2000

- 1.คลิก [Start] > [Run..]
- 2.ใส่คำสั่ง telnet คลิก [OK]
- 3.เมื่อหน้าต่าง telnet เปิดออก ให้พิมพ์คำสั่ง set local_echo แล้วกด Enter (ทุกครั้งหรือทุกบรรทัดที่ป้อนคำสั่ง, ข้อมูลใด ๆ เสร็จให้กด Enter) เพื่อให้มีการแสดงคำสั่งที่เราป้อนเข้าไปด้วย เรียกว่าสะท้อนหรือ echo (มิฉะนั้นจะมองไม่เห็นที่เราพิมพ์อะไรเข้าไป)
- 4.เชื่อมต่อเข้ากับ Mail Server ที่ localhost port 25 (SMTP) โดยพิมพ์ open localhost 25 (ที่ใช้ localhost เพราะ Mail Server อยู่ที่เครื่องเราใช้เครื่องเราเป็นตัวส่ง)
- 5.ดูการตอบรับจาก Mail Server ว่ามีการตอบรับจากการที่เราเข้าไปเชื่อมต่อหรือไม่ เช่น 220 ชื่อ Mail Server ... แสดงว่าพร้อมแล้ว หากเงียบไม่มีการตอบรับ หรือมีข้อความ can't open host แสดงว่า SMTP ของ Mail Server ไม่ทำงานให้ลองตรวจสอบดู
- 6.เริ่มพิมพ์คำสั่ง ดังนี้

HELO	ชื่อ โดเมนหรือชื่อท่าน	
250 ...	เครื่องตอบรับกลับมา	
MAIL FROM: support@mymail.com		(อีเมลล์ผู้ส่ง)
250 ...	เครื่องตอบรับกลับมา	
RCPT TO: pop@mymail.com		(อีเมลล์ผู้รับ)
250 ...	เครื่องตอบรับกลับมา	
DATA		(เริ่มต้น data ของเมล)
354 ...	เครื่องตอบรับกลับมา	
to: POP		(ชื่อผู้รับที่ติดอยู่หัวเมล ใช้อะไรก็ได้)
from: support@mymail.com		(ชื่อผู้ส่งที่ติดอยู่หัวเมล)
subject: พิมพ์หัวข้อเรื่อง subject ที่ต้องการ		
กด [Enter] เว้น 1 บรรทัด		
พิมพ์เนื้อจดหมายบรรทัดที่ 1		
พิมพ์เนื้อจดหมายบรรทัดที่ 2		

พิมพ์เนื้อจดยกหมายบรรทัดต่อไป ...

(พิมพ์จุด . เดียวแล้วกด Enter เพื่อกำหนดว่าจบอีเมลล์เท่านั้น)

250 ... เครื่องตอบรับมา

QUIT (ยุติการติดต่อ)

ขั้นตอนในการติดต่อทาง Telnet ใน Windows 98/ME

- 1.คลิก [Start] > [Run]
- 2.พิมพ์คำสั่ง telnet พร้อมทั้งระบุโฮสต์หรือเซิร์ฟเวอร์และ port แล้วคลิก [OK]
- 3.หน้าต่าง telnet เปิดออก ดูว่ามีการตอบรับจาก Mail Server หรือไม่ หากเงียบ หรือมีข้อความ can't open host แสดงว่า SMTP ของ Mail Server ไม่ทำงาน ให้ลองตรวจสอบดู
- 4.เซท local echo เพื่อให้มีการแสดงคำสั่งที่เราป้อนเข้าไปด้วย มิฉะนั้นจะมองไม่เห็นที่เราพิมพ์อะไรเข้าไป โดยคลิก [Terminal] > [Preferences..]
- 5.คลิกให้มี / ที่ local echo
- 6.คลิก [OK] แล้วเริ่มพิมพ์คำสั่งอื่น ๆ เหมือนเดิม

ตัวอย่างการใช้ Telnet ส่งเมลล์ให้กับ hotmail.com

- 1.เปิดหน้าต่าง Telnet แต่ชื่อ Mail Server ที่ใช้ให้ใช้ของ hotmail คือ mc1.law13.hotmail.com (ตามปกติถ้าเป็น Mail Server ของเครื่องเราเองจะใช้ localhost) ถึง mc5.law13.hotmail.com ตัวใดตัวหนึ่ง ส่วนหมายเลข port ใช้ 25 เหมือนเดิม
- 2.เมื่อ Mail Server ของ hotmail ตอบรับก็พิมพ์คำสั่งตามหลักการเดิม

2.3.2.3 การส่งเมลล์โดยใช้ PHP ติดต่อในระดับ Socket

รูปแบบของการเปิด socket ด้วย fsockopen

```
fsockopen (hostname, port[$errno [, $errstr [, timeout]])
```

- hostname คือ ชื่อเซิร์ฟเวอร์หรือ host ที่ต้องการเชื่อมต่อ จะเป็นเลข IP address ก็ได้
- port คือ หมายเลข port ที่ต้องการเชื่อมต่อ เช่น SMTP คือ port 25 , pop3 คือ port 110 ฯลฯ
- \$errno , \$errstr เป็นค่าที่ได้ ในกรณีเกิด Error
- timeout เป็นการกำหนดระยะเวลาในการเชื่อมต่อ ว่าจะสิ้นสุดกี่วินาที

fsockopen() ใช้สำหรับเชื่อมต่อซอกเก็ตเข้ากับ โฮสต์หรือเซิร์ฟเวอร์ และหมายเลข port ที่ต้องการ เมื่อเชื่อมต่อได้แล้วจะได้รหัสอ้างอิงหรือ pointer กลับมา

ตัวอย่างเช่น

```
$sock=fsockopen ("localhost" , 25);
```

 หรือ

```
$sock=fsockopen ("mc1.law5.hotmail.com" , 25);
```

หลังจากที่เชื่อมต่อด้วย fsockopen() สำเร็จ จะได้ค่าเป็นรหัสอ้างอิงเรียกว่า file pointer เวลาใช้งานจริงๆ เราจะใช้ if ตรวจสอบว่าเชื่อมต่อได้หรือไม่ ดังตัวอย่างต่อไปนี้

```
if(!$sock) {
    echo "<br>ติดต่อ server ไม่ได้";
}
else {
    echo "ติดต่อ server ได้";
}
```

ปัญหาของการเชื่อมต่อเซิร์ฟเวอร์หรือ โฮสต์ไม่ได้

- เว็บเซิร์ฟเวอร์ที่เรารัน php อยู่ขณะนั้นไม่สนับสนุน fsockopen()
- ชื่อโฮสต์หรือเซิร์ฟเวอร์ที่ php เข้าไปขอเชื่อมต่อนั้นไม่มีตัวตน
- พอร์ตที่เข้าไปขอเชื่อมต่อนั้นปฏิเสธ หรือไม่อนุญาต หรือไม่พร้อม ฯลฯ

การส่งคำสั่ง ด้วยฟังก์ชัน fputs

หลังจากเชื่อมต่อได้แล้วต้องรีบส่งคำสั่ง เพราะอาจหมดเวลา (timeout) ฟังก์ชันที่ใช้คือ fputs ใช้สำหรับส่งคำสั่งหรือข้อมูลไปยังโฮสต์หรือเซิร์ฟเวอร์ที่เชื่อมต่ออยู่ ดังรูปแบบต่อไปนี้

fputs (รหัสอ้างอิง pointer , ข้อมูลหรือคำสั่งที่จะส่ง)

- รหัสอ้างอิง pointer คือ ค่าที่อ้างอิงที่เราได้จากการเชื่อมต่อชื่อที่เกิดด้วย fsockopen จากตัวอย่างนี้ค่าอ้างอิงเก็บไว้ที่ \$sock
- ข้อมูลหรือคำสั่งที่จะส่ง เช่น HELO clientname ให้ปิดท้ายด้วย \n หรือ \r\n ตัวอย่างเช่น

```
fputs ($sock, "HELO iampop\r\n");
```

 หรือ

```
fputs ($sock, "HELO $from\r\n");
```

การรับคำตอบรับด้วยฟังก์ชัน fgets

การสื่อสารที่ถูกต้องจะเป็นลักษณะได้ตอบ เมื่อเราส่งคำสั่ง หรือ ข้อมูล ไปแล้ว (ส่งด้วย fputs) เราก็ต้องรับฟังว่าเซิร์ฟเวอร์หรือ โฮสต์อีกฝั่งที่เราเชื่อมต่อ ตอบรับมาว่าอย่างไร พร้อมหรือว่า error โดยใช้ fgets ใช้สำหรับรับรหัสหรือข้อมูลจากโฮสต์หรือเซิร์ฟเวอร์ที่เชื่อมต่ออยู่ มีรูปแบบดังนี้

fgets (รหัสอ้างอิง pointer , ความยาว)

- รหัสอ้างอิง pointer คือ คำอ้างอิงเหมือนกับ fputs
- ความยาว หมายถึง จำนวน string ที่รับเข้ามา (ลบด้วย 1 ไบต์)

14942061 02

ปกติความยาวของการตอบรับที่ Mail Server ส่งออกมาจะไม่เกิน 1024 ไบต์ ดังนั้น เราสามารถกำหนดความยาวได้ดังนี้

```
fputs ($sock, "HELO iam pop\r\n");
```

```
$reply = fgets ($sock, 1024);
```

```
echo "$reply";
```

หากการสื่อสารสมบูรณ์ ที่บราวเซอร์จะแสดงรหัส Reply code และข้อมูลที่มีลักษณะคล้ายดังนี้

```
220 Mail Server Version ...
```

ผส.

04938

2545

ตัวอย่างการใช้ php เชื่อมต่อระดับซ็อกเก็ต กับ hotmail.com

```
<?php
define('CRLF', "\r\n");
$server='mc1.law13.hotmail.com';
$recipient="user@hotmail.com";
$from="note@man.com";
$subj="ส่งเมลล์ผ่าน fsockopen";
$msg="<h2>Halo, Kob</h2><hr>ทดสอบส่งเมลล์แบบ HTML ผ่าน fsockopen..";
echo "<b>เชื่อมต่อซ็อกเก็ต....</b>";
$sock = fsockopen ($server, 25);
if (!$sock) {
    echo "<br> ติดต่อ $server ไม่ได้ !";
}
else {
    echo "ติดต่อ SMTP $server Ok..<br>";
    fputs ($sock, "HELO $from\r\n");
    echo "ส่ง: HELO $from<br>";
    $reply = fgets ($sock, 1024);
    echo "$reply <hr>";

    fputs ($sock, "MAIL FROM: <$from>\r\n");
    echo htmlspecialchars("ส่ง: MAIL FROM: <$from>")."<br>";
```

```

$reply = fgets ($sock, 1024);
echo "$reply<hr>";

fputs ($sock, "RCPT TO: <$recipient>\r\n");
echo htmlspecialchars("ส่ง: RCPT TO: <$recipient>")."<br>";
$reply = fgets ($sock, 1024);
echo "$reply<hr>";
fputs ($sock, "DATA\r\n");
echo "ส่ง: DATA<br>";
$reply = fgets ($sock, 1024);
echo "$reply<hr>";
$body="To:$recipient".CRLF.
    "From: $from".CRLF.
    "Subject: $subj".CRLF.
    "Content-Type: text/html; charset=windows-874".CRLF.
    "X-Priority: 1".CRLF.
    CRLF.
    $msg.CRLF;
fputs ($sock, "$body\r\n.\r\n");
echo "ส่ง: ข้อมูล body และจบด้วย . จุด<br>";
$reply = fgets ($sock, 1024);
echo "$reply<hr>";
fputs ($sock, "QUIT\r\n");
echo "ส่ง: QUIT<br>";
$reply = fgets ($sock, 1024);
echo "$reply<br>";
}
?>

```


2.3.2.4 การตรวจสอบ อีเมลแอดเดรส ว่ามีอยู่จริงหรือไม่

หลักการตรวจเช็คว่ามีอีเมลอยู่จริงหรือไม่ ก็ใช้วิธีเช่นเดียวกับการส่งเมล ซึ่งถ้า server ไม่พบชื่อ user ในระบบ (เท่ากับว่าไม่มีเมลนั้นอยู่จริง) ก็จะส่งรหัส reply ข้อผิดพลาดกลับมาให้ php ทราบ

ตารางที่ 2.2 ตัวอย่างแสดงรหัส Reply Code ของ SMTP

รหัส	SMTP
211	แสดง System Status หรือ System help
220	SMTP พร้อมบริการ (Service ready)
221	ปิดการเชื่อมต่อ (Service Closing)
250	แสดงการตอบรับ OK, ถูกต้อง ...
251	เมลล์จะ forward ส่งต่อไปอีกทอดหนึ่ง
354	พร้อมรับ DATA ให้ส่งข้อมูลเมลล์มาได้
421	SMTP ไม่พร้อมบริการ (Service not available)
451,452	มีปัญหาเกี่ยวกับระบบ Mail box ชัดข้อง
500	รูปแบบคำสั่งหรือข้อมูล error (Syntax error)
501	รูปแบบอาร์กิวเมนต์ error (Syntax error)
502	คำสั่งใช้ไม่ได้, ไม่รู้จักคำสั่ง ฯลฯ
503	ลำดับคำสั่งผิดคิว (Bad sequence)
550 ขึ้นไป	มีปัญหา เช่น ไม่มี user, mailbox ชัดข้อง คำสั่งที่ส่งมาถูกปฏิเสธ

ตัวอย่างเช่น

```
<?php
$recipient="pop@hotmail.com";
$sock=fsockopen("mc1.law5.hotmail.com",25);
$fputs($sock,"REPT TO: <$recipient>\r\n");
$reply=fgets($sock,1024);
echo "$reply";
?>
```

ถ้าไม่มีเบอร์อีเมลอยู่ในเซิร์ฟเวอร์ รหัส reply ตอบกลับจะไม่ใช่ 220, 250, 354 แต่จะเป็น 500 ขึ้นไป

ปัญหาของการเช็คแบบข้าง คือ จะทราบชื่อเซิร์ฟเวอร์อย่างไร เพราะเซิร์ฟเวอร์ SMTP ตัวรับอาจไม่ได้ชื่อเดียวกันกับโดเมน เช่น hotmail.com จะมีชื่อเซิร์ฟเวอร์เป็น mc1.law5.hotmail.com

การหาชื่อเซิร์ฟเวอร์ด้วยฟังก์ชัน getmxrr
รูปแบบทั่วไป

```
getmxrr(hostname , ตัวแปร)
```

- hostname หรือโดเมนเป็นชื่อที่เราต้องการให้หา MX record
- ตัวแปร ทำหน้าที่เก็บรายชื่อ MX record ที่ได้

MX record (Mail Exchange record) เป็นรายชื่อเหมือนกับ DSN แต่รายชื่อนี้เป็นตัวบอกชื่อโฮสต์หรือเซิร์ฟเวอร์ไหนทำหน้าที่เกี่ยวกับเมล

การแยกชื่อ user กับโดเมนออกจากกัน

การใช้ getmxrr เราต้องเอาเฉพาะ โดเมนออกมาจากเบอร์อีเมลก่อน วิธีการ คือ ใช้ split แยกเอา string หลัง @ มาไว้ที่ตัวแปร \$domain ดังนี้

```
list($user , $domain) = split("@", "pop@hotmail.com");
```

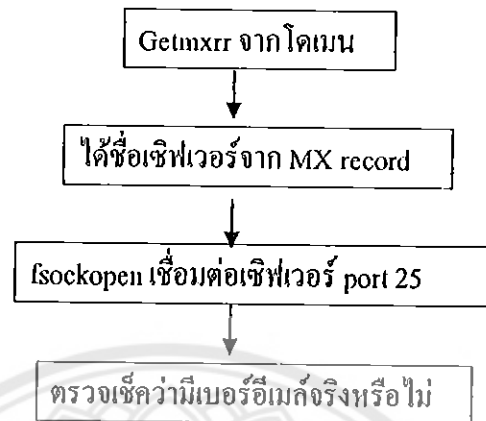
\$user	มีค่าเป็น	"pop"
\$domain	มีค่าเป็น	"hotmail.com"

ตัวอย่างการใช้งาน

```
<?php
$email="your@hotmail.com";
list($user,$domain) = split("@",$email);
echo "User: $user <br>Domain: $domain<br>";
getmxrr ($domain, $mxhosts);
if( !getmxrr($mxhosts)) {
echo "ไม่มี $domain"; }
else {
echo "มี $domain"; }
?>
```

การใช้ `getmxrr` ให้ทำงานร่วมกับ `fsockopen`

มีขั้นตอนการทำงาน ดังนี้



การใช้ฟังก์ชัน `checkdnsrr`

ทำหน้าที่เช็ค DNS โดยดูว่ามี MX record หรือไม่ ถ้าไม่มีจะได้ค่ากลับมาเป็น `false` แสดงว่าผู้ใช้ป้อนโดเมนผิด แต่ถ้ามีจะได้ค่ากลับมาเป็น `true`

ตัวอย่างการใช้งาน

```

<?php
$email="your@hotmail.com";
list($user,$domain) = split("@",$email);
echo "User: $user <br>Domain: $domain<br>";
if(!checkdnsrr($domain,"MX")) {
echo "ไม่มี $domain"; }
else {
echo "มี $domain"; }
?>
  
```

2.3.3 MIME รูปแบบมาตรฐานในการส่งเมล [1]

2.3.3.1 ความหมายของ MIME

MIME (Multipurpose Internet Mail Extensions) เป็น Protocol อย่างหนึ่ง ที่ใช้เป็นหลักมาตรฐานของรูปแบบในการส่งเมล อาทิเช่น ส่งเมลแบบ Text/HTML , การกำหนดความสำคัญ (Priority) และที่สำคัญคือการส่งแนบไปกับเมล (Attachment)

ข้อมูลในส่วนต่างๆ ของการส่งอีเมล

ในฟังก์ชันของการส่งอีเมลทั่วไป จะประกอบด้วยข้อมูลในส่วนต่างๆ ดังนี้

Mail(\$email,\$subject,\$msg,\$headers)

- \$email คือ อีเมลของผู้รับ
- \$subject คือ หัวเรื่อง
- \$msg คือ เนื้อความ (body message)
- \$headers คือ อีเมลของผู้ส่ง รูปแบบในการส่ง

ในการส่งเมลแบบธรรมดาทั่วไป ในส่วนของ \$headers จะประกอบด้วย อีเมลของผู้ส่ง เท่านั้น แต่เมื่อเราใช้มาตรฐานในการส่งเมลแบบ MIME ในส่วนของ \$headers ก็จะมีการกำหนดในรูปแบบของการส่งแนบ ไม่ว่าจะเป็นการกำหนดความสำคัญ(Priority) , การเลือกส่งแบบ Text หรือ HTML รวมถึงการส่งแนบไฟล์แนบไปการอีเมลด้วย (Attachment) และอื่นๆ

2.3.3.2 โครงสร้าง MIME

MIME จะอยู่ในส่วนของ \$headers ซึ่งจะมีโครงสร้างหรือส่วนประกอบด้วยนี้

* MIME-Version

เป็นตัวกำหนดเวอร์ชันของ MIME ซึ่งก็คือ 1.0

* Content-Type

เป็นตัวบ่งบอกว่าข้อมูลในส่วนนั้นเป็นชนิดใด เช่น Text/plain ,text/html. Image/gif, image/jpg . application/zip video หรือผสมกันหลายๆ ชนิด (multipart)

* Content-Transfer-Encoding

เป็นตัวบ่งบอกว่าข้อมูลส่วนนั้นเข้ารหัส (Encoding) มาด้วยวิธีแบบใด (บ่งบอกไว้เพื่อโปรแกรมของอีเมลต่างๆจะได้ถอดรหัสได้ถูกต้อง) ส่วนใหญ่ที่ใช้กันมีดังนี้

- 1) quoted-printable ใช้กับข้อความ Text ปกติ
- 2) base64 ใช้กับข้อมูลไบนารีทั่วไป เช่นไฟล์ที่แนบไปกับเมล(รูปภาพ เสียงเพลง ฯลฯ)

นอกจากนี้ยังมีการเข้ารหัสแบบ 7bit ,8bit binary ,custom แต่ไม่เป็นที่นิยมจึงไม่ขอพูดถึง รายละเอียดในที่นี้

*** Content-Disposition**

เป็นตัวบ่งบอกรายละเอียด attachment และ ชื่อไฟล์

*** Content-Description**

สำหรับเขียนคำอธิบายเพิ่มเติมอื่นๆ (ไม่มีก็ได้)

โครงสร้าง MIME สำหรับ Text ธรรมดา และ HTML

โครงสร้างของ MIME ต่อไปนี้จะกำหนด Content - Type หลักเป็นแบบ multipart/alternative ซึ่งแบ่งข้อมูลออกเป็น 2 ส่วนคือ ส่วนที่เป็น text/plain และส่วนที่เป็น text/html ดังนี้

Content-Type: multipart/alternative;

boundary="abc123" กำหนดรหัสแบ่งเขต เป็น abc123 (ตั้งเป็นอะไรก็ได้)

--รหัสแบ่งเขต

Content-Type: text/plain;

ข้อมูลสำหรับ Text ธรรมดา.....

--รหัสแบ่งเขต

Content-Type: text/html;

แท็กข้อมูลจดหมายเขียนอยู่ในรูปแบบ HTML

--รหัสแบ่งขอบเขต -- //เป็นการปิดขอบเขต

2.3.3.3 การส่งแบบ Text และ HTML

โครงสร้างของ MIME ต่อไปนี้จะกำหนด Content - Type หลักเป็นแบบ multipart/mixed แบ่งเป็นส่วนต่างๆ แยกแต่ประเภทที่ต้องการแนบไฟล์ และมีการกำหนด Content-Transfer-Encoding ตามประเภทที่ต้องการแปลงรหัส ดังตัวอย่างนี้

MIME-Version: 1.0;CRLF.

Content-Type: multipart/mixed;

boundary="abcd12345"

'X-Priority: 3'

--abcd12345

Content-Type: image/gif;

// แยกแต่ประเภทที่ต้องการแนบไฟล์

name="pointer.gif"

Content-Transfer-Encoding: base64

// ตามประเภทที่ต้องการแปลงรหัส

Content-Disposition: attachment;

filename="pointer.gif"

<CRLF>

ส่วนที่เป็นข้อมูลไฟล์ที่เข้ารหัส base64 แล้ว ซึ่งจะเป็นการแปลงจากไบนารีมาเป็นรูปแบบ Text
ธรรมดา

```
R0IGODlhFwAYAPcAAAAAP///wAAAAAAAAACH5BAAAAAALAAAAAXABGA  
AAhiAAMIHEiwoMGDAgEAQMhw4MIADxsefBhRYkGKFicmzGiQokKODjdC/JgRI8SEFR  
GaxJgS5cKVII6VCmSYMSWFzWebIjzZM+QOIX+/Ak0KE2eSJMeVZrTYk+iDhVKnQqSYO  
AAOw=='.CRLF.
```

--abcd12345-- // สิ้นสุดขอบเขต (มีสองขีดนำหน้า และตามหลังรหัส)

2.3.3.4 การเข้ารหัสไบนารีเป็น Text แบบ base64

ฟังก์ชันหลักที่ใช้สำหรับเข้ารหัส (encoding) แบบนี้คือ

base64_encode(ข้อมูลไบนารี)

ซึ่งข้อมูลไบนารีก็ได้มาจากการเปิดอ่านไฟล์โดยตรง ดังนั้นการใช้ฟังก์ชัน base64_encode จึง
ต้องอาศัยฟังก์ชันเกี่ยวกับไฟล์ด้วยดังนี้

- fopen (ชื่อไฟล์, โหมด) สำหรับเปิดหรือติดต่อไฟล์
- filesize(ชื่อไฟล์) สำหรับหาขนาดของไฟล์
- fread(ไฟล์พอยน์เตอร์, ขนาด)อ่านข้อมูลจากไฟล์ที่ต้องการ
- fclose(ไฟล์พอยน์เตอร์) ปิดหรือยกเลิกการติดต่อกับไฟล์

เมื่อเอมารวมกันจะได้ตัวอย่างดังนี้

```
<?php  
$file="pointer.gif"; // ชื่อไฟล์ที่เปิด  
$fp=fopen($file, "rb"); // เปิดไฟล์เพื่ออ่าน  
$data = fread($fp, filesize($file)); // อ่านข้อมูลไบนารีจากไฟล์เข้ามา  
fclose ($fp); // อ่านจบแล้วให้ปิดไฟล์  
$data = base64_encode($data); // เข้ารหัส base64  
$data = chunk_split($data); // ตัดเปลี่ยนสตริงบางส่วน  
echo $data; // แสดงข้อมูลที่เข้ารหัสแล้วออกบราวเซอร์  

```

หลักการทํางาน การเข้ารหัสไบนารีเป็น Text แบบ base64

- เริ่มจากการเก็บชื่อไฟล์ไว้ในตัวแปร \$file ดังนี้

```

$file="pointer.gif";           // กรณีไฟล์ที่เปิด อยู่ในไดเรกทอรีเดียวกัน
หรือ $file="/www/server/ pointer.gif"; // กรณีไฟล์ที่เปิด อยู่ต่าง ไดเรกทอรีกัน
หรือ $file="folder/index.zip"; // กรณีไฟล์ที่เปิด อยู่ในไดเรกทอรีย่อยชื่อ folder

```

- เสร็จแล้วก็เปิดไฟล์เพื่ออ่านด้วย \$fp=fopen(\$file, "rb");
 - จากนั้นก็เก็บข้อมูลไว้ที่ \$data โดยขนาดก็ได้จากฟังก์ชัน filesize(\$file) เมื่ออ่านเสร็จแล้ว

ก็ยกเลิกการติดต่อ

```

$data = fread($fp, filesize($file));

```

```

fclose ($fp);

```

- ขั้นตอนต่อไปนำข้อมูล \$data ที่อ่านได้ (ยังเป็นไบนารีอยู่) มาเข้ารหัสแปลงจากไบนารีเป็น Text ด้วย base64-encode แล้วตัดเปลี่ยนสตริงบางส่วนเพื่อให้ได้ขนาดข้อมูลที่เล็กลง

```

$data = base64_encode($data);

```

```

$data = chunk_split($data);

```

- แสดงข้อมูลที่เข้ารหัสแล้วออกบราวเซอร์

```

echo $data;

```

2.3.3.5 การแนบไฟล์ไปกับการส่งอีเมล (Attachment)

สภาพแวดล้อมและไดเรกทอรี

เนื่องจากสคริปต์นี้มีการอัปโหลดไฟล์จากเครื่องผู้ใช้งานมาที่เก็บที่เซิร์ฟเวอร์เพราะไฟล์ที่จะแนบ Attach ไปกับอีเมลนั้นจะต้องอัปโหลดเข้ามาพักที่เซิร์ฟเวอร์นี้ก่อน เมื่อมีการส่งเมลจริง เครื่องเซิร์ฟเวอร์ก็จะเปิดไฟล์ที่อัปโหลด แล้วเอาไปเข้าดั่งนั้นสิ่งที่ควนคำนึงมีดังต่อไปนี้

ไดเรกทอรีที่เก็บไฟล์

กรณีที่อัปโหลดไฟล์ Attachment ไว้ไดเรกทอรีเดียวกับสคริป ให้กำหนดค่าไว้ใน

\$upload_dir -ดังนี้ (วิธีใช้กับ Unix หรือ Windows ก็ได้)

<pre>\$upload_dir="."; //ไม่จำเป็นต้องกำหนดค่าใน</pre>
--

อย่างไรก็ตามเพื่อไม่ให้ไฟล์ที่อัปโหลดเข้ามานั้นปะปนกับไฟล์ระบบของเราดั่งนั้นควรแยกไดเรกทอรีไว้ต่างหาก แบ่งเป็น 2 กรณีดังนี้

- เก็บไว้ไดเรกทอรีย่อยลงไปจากไดเรกทอรีที่เก็บสคริป เช่น (ใช้กับ Unix หรือ Windows ก็ได้)

```
// ไดรเรททอรีย่อยของสคริปชื่อ temp ($basedir ไม่ต้องกำหนด)
$upload_dir="temp/";
```

- เก็บไว้ไดรเรททอรีต่างหากที่ไม่ได้อยู่ในเส้นทางเดียวกับสคริป (คนละที่) ให้อ้างอิงพาทหรือไดรเรททอรีแบบเต็ม ๆ ซึ่งกรณีนี้เราจะต้องทราบพาทเต็ม ๆ ของไดรเรททอรีรากเสียก่อน ดูหัวข้อต่อไป

การหาพาทเต็ม ๆ ของไดรเรททอรีราก (Root)

ทำได้หลายวิธีดังนี้

- สอบถามจากฝ่าย support ของโฮสที่เราเช่าอยู่
- ใช้ตัวแปรระบบ \$DOCUMENT_ROOT (สำหรับ Unix, Linux) เช่น

```
$basedir = '4DOCUMENT_ROOT';
```

- สำหรับ Windows ไม่มี 4DOCUMENT_ROOT ให้อ่านจากตัวแปร 4PATH_TRANSLATED แทนเช่น

```
<?echo 4PATH_TRANSLATED";M?>
```

จะได้ข้อมูลพาทและไฟล์ที่รันสคริปเช่น (ชื่อไฟล์ที่กำลังรันคือ te4st.php และเอาไว้ xyz)

"d:myweb/root/xyz/test.php" หรือ "d:\myweb\root\xyz\test.php"

จากนั้นตัดเอาชื่อพาทและไฟล์ของสคริปออกแล้วเก็บไว้ที่ \$basedir ใช้เครื่องหมาย / แทน \ ดังนี้

```
$basedir = "dL/myweb/root";
```

ได้ไดรเรททอรีราก(ROOT) แล้วก็มารวมกับชื่อไดรเรททอรีเก็บไฟล์

เมื่อไดรเรททอรีราก (Root) แล้วก็มารวมกับชื่อไดรเรททอรีที่เราเก็บไฟล์ Attachment ไว้ สมมุติว่าเก็บไว้ที่ upload ดังนั้นกำหนดดังนี้

```
$basedir = "/www/server/html" ; // ไดรเรททอรีรากของเว็บไซต์
$upload_dir = "$basedir/upload/"; // ไดรเรททอรีเก็บไฟล์ Attachment
```


หรือสำหรับ Windows

```
$basedir = "d:/myweb/root";
$upload_dir = "$basedir/upload/"; // โดเรททอรีเก็บไฟล์ Attachment
```

เซทโหมดหรือ Permission ของโดเรททอรีให้ Write ได้

เพื่อให้ user ใด ๆ สามารถอัปโหลดไฟล์เข้าไปเก็บได้ จะต้องเซทให้โดเรททอรีดังกล่าวสามารถ Write หรือเขียนไฟล์เข้าไปได้ ซึ่งต้องเซทอ็อปดังนี้

- Unix, Linux ต้องเซทโหมด chmod โดเรททอรีที่เก็บไฟล์ให้เป็น 777 แต่โฮสต์บางแห่งเซทให้เป็น 755 ก็สามารถอัปโหลดไฟล์เข้าไปเก็บได้แล้ว เช่น I2s ฯลฯ
- ถ้าเป็น Windows NT,2000 (IIS) ใช้ระบบไฟล์ NTFS จะใช้ chmod เหมือน Unix ไม่ได้ ซึ่งถ้าเป็นโฮสต์เช่าจะต้องให้ผู้ดูแลหรือ admin เป็นผู้เซทให้ หรือบางเว็บมี Webbase Control Panel ก็เซทให้เป็น Write เสียเลย (Win 9x, ME ไม่จำเป็นต้องเซทอะไร สามารถอัปโหลดแล้ว Write ได้ทันที)

การเลือกไฟล์ Attachment

```
Add_attachment ($userfile, $upload_dir);
```

จะเป็นการเรียกฟังก์ชัน add_attachment ซึ่งค่าที่ส่งเข้าฟังก์ชันคือ \$userfile (ไฟล์ที่อัปโหลดเข้ามา) และ \$upload_dir (โดเรททอรีที่เก็บไฟล์)

เนื่องจากการอัปโหลด ดังนั้นตัวแปรที่เกิดขึ้นอัตโนมัติคือ \$userfile_name (ชื่อไฟล์), \$userfile_type (ชนิดของไฟล์), \$userfile_size (ขนาดไฟล์) ข้อมูลทั้งหมดนี้เรานำเอามาต่อกันแล้วเก็บเข้าตัวแปร \$att_str ให้เป็นรูปแบบดังนี้

```
ชื่อไฟล์-[ขนาดไฟล์(bytes)]@Content
$att_str = "$userfile_name~[$userfile_size]@$userfile_type"; เช่น
abcd.gif-[2045]@image/gif
```

หลักการ Remove ยกเลิกเอา Attachment ออก

กรณีที่ยกเลิกไม่เอา Attachment แล้วและคลิกปุ่ม Remove Attachment ตรวจสอบแล้วจะตรงกับ “Remove Attachment” ดังนั้นจึงเข้ามาทำงานในส่วนนี้ ซึ่งก็คือ เรียกฟังก์ชัน unlink_file เพื่อลบไฟล์ Attachment ที่อัปโหลดเข้ามา ออกไปจากเซิร์ฟเวอร์และเคลียร์ค่าตัวแปร \$att_str

```
unlink_file ($att_str);           // ลบไฟล์ออกจากเซิร์ฟเวอร์
$att_str = "";
```

การทำงานฟังก์ชัน unlink_file ในส่วนที่ลบไฟล์ก็จะแยกชื่อไฟล์ออกมาก่อน (ยังคงจำได้ว่า \$att_str จะเก็บค่าเช่น filename.ext~[ขนาด]@content) ด้วย split โดยใช้ ~ เป็นตัวคั่น จากนั้นจึงลบไฟล์ออกไป ดังนี้

```
list ($fname, $size) = split ("~" $att_str);           // แยกเอาเฉพาะชื่อไฟล์และนามสกุล
if (file_exists ("$upload_dir$fname")) {             // เช็คว่ามีไฟล์อยู่หรือไม่
    unlink ("$upload_dir$fname");                     // ลบไฟล์
```

เมื่อเคลียร์ค่าในตัวแปร \$att_str แล้วก็เท่ากับว่าชื่อไฟล์ในกรอบแสดง Attachment ที่ฟอร์มก็จะหายไปด้วย

ป้องกันชื่อไฟล์ซ้ำด้วย id

กรณีนี้ ในช่วงเวลาเดียวกัน มีคนหลายคนส่งไฟล์ Attachment ชื่อเดียวกันมาให้เรา อาจทำให้การ Upload ไฟล์ขึ้นไปบน Server ซ้อนทับกันได้ และจะส่งไปได้ไม่ครบถ้วน เราจึงแก้ปัญหาดังนี้ โดยการกำหนดรหัสให้แต่ละ file Attachment จึงสร้างรหัสตัวเลข 4 หลักขึ้นมาไว้หน้าหน้าชื่อไฟล์ ซึ่งเท่ากับว่าหมดปัญหาการซ้ำหรือทับไฟล์

มีจุดที่น่าสังเกตคือว่า เนื่องจากสคริปมีการ โหลดหรือ Reload ทุกครั้งที่คลิกปุ่มเช่น Add, Remove, send ดังนั้นจึงต้องใช้ isset เข้ามาตรวจสอบเช็คก่อนว่ามี เลข id อยู่แล้วหรือไม่ ถ้ายังถึงจะต้องเลขสุ่มขึ้นมา ดังนี้

```
if (!isset ($id)) {
    srand ((double) microtime() * 1000000);
    $id = rand (1000,9999); // เลขสุ่มหลัก 4 หลัก 1000 ถึง 9999
}
```

หากไม่ใช้ isset ดังกล่าว ตัวเลข id จะเปลี่ยนไปเรื่อยๆ จะมีผลให้ตอนส่งเมลนั้น เครื่องหาไฟล์ Attachment ไม่เจอ

เมื่อผู้ใช้คลิก Add Attachment เข้ามาแล้ว ที่ฟังก์ชัน add_attachment จะมีการ copy ไฟล์นั้นเข้าสู่ไดเรกทอรีที่เตรียมไว้ (ในที่นี้คือ upload) โดยเติม id เลขคู่ 4 หลักเข้าไปหน้าชื่อไฟล์จริง เช่น pic1.zip ก็จะถูกกลายเป็น 2547pic1.zip จากนั้นก็เก็บรายชื่อเข้า \$att_str ดังนี้

```
$ userfile_name = $id.$userfile_name; // นำหน้าชื่อไฟล์ด้วย id
if (copy ($userfile, "$upload_dir$userfile_name")) { // ก๊อปปี้ไฟล์เข้าไดเรกทอรี
    $ att_str = "$userfile_name~[$userfile_size]@$userfile_type;";
```

ตัวแปรที่เก็บรายชื่อ Attachment

จากในหัวข้อการเลือกไฟล์ Attachment ที่มี เพียงไฟล์เดียว การเก็บชื่อไฟล์ Attachment จึงไม่ยุ่งยากนัก คือเก็บชื่อและรายละเอียดเข้าไว้ใน \$att_str ตรงๆ

แต่ในที่นี้มี Attachment หลายไฟล์ ดังนั้นจึงต้องมีหลักในการจัดเก็บเพราะจะทำให้จัดการได้ง่าย ซึ่งในที่นี้แบ่งเป็นฟิลด์โดยใช้เครื่องหมาย ; คั่น สมมุติว่าไฟล์ Attachment มี 3 ไฟล์คือ file1.zip, file2.gif, file3.jpg ดังนั้นรูปแบบการเก็บจะเป็นดังนี้

```
file1.zip~[ขนาด]@type; file1. Zip~[ขนาด] @type; file1. Zip~[ขนาด] @type;
```

โดยการกำหนดค่าเข้าไว้ในตัวแปร \$att_str (ในฟังก์ชัน add_attachment) หลังจากที่อัปโหลดและ copy ไฟล์เข้าไดเรกทอรีเรียบร้อยแล้ว ดังนี้

```
$ att_str = "$userfile_name~[$userfile_size]@$userfile_type;";
```

แสดงรายชื่อ Attachments ในกรอบ listbox

จริงๆ แล้วการแสดงรายชื่อ Attachments นั้นทำได้ง่ายมาก คือใช้ echo แสดงค่าในตัวแปร \$att_str ออกมาโดยตรงก็ได้ เพราะเป็น string ที่มีรายชื่อไฟล์ครบอยู่แล้ว

แต่ในที่นี้เราจะให้แสดงรายชื่อในกรอบ listbox แทน จุดประสงค์หลักคือให้ผู้ใช้สามารถคลิกเลือกเอาไฟล์ Attachment ใดออกไปก็ได้และอีกประการหนึ่ง

ปกติแล้วอินพุทที่เป็น select (เรียกว่า combobox, dropdown list ก็ได้) เรามักใช้กันในลักษณะที่เป็นบรรทัดเดียว เช่น

```
<select name=xxx>
  <option value= ข้อมูล1 > ชื่อรายการ 1 ที่แสดงในกรอบ
  <option value= ข้อมูล2 > ชื่อรายการ 2 ที่แสดงในกรอบ
  .....option ต่อๆไป...
</select>
```

แต่ในที่นี่เพื่อให้ผู้ใช้เห็นรายการ Attachment ทั้งหมดโดยไม่ต้องคลิกดู ดังนั้นจึงเลือกใช้แบบ listbox ซึ่งวิธีการใช้ก็คล้ายกับ combobox แต่เรากำหนด size เข้าไปเลยว่าต้องการขนาดกี่บรรทัด เช่น ต้องการแสดง 4 บรรทัด ดังนี้

```
<select name=xxx size =4>
  <option value= ข้อมูล1 > ชื่อรายการ 1 ที่แสดงในกรอบ
  <option value= ข้อมูล2 > ชื่อรายการ 2 ที่แสดงในกรอบ
  .....option ต่อๆไป...
</select>
```

ฟังก์ชันแสดงรายชื่อ Attachment

ฟังก์ชันแสดงรายชื่อ Attachment ในที่นี้คือ dsp_att_list อยู่ตรงบริเวณแบบฟอร์ม
ค่าที่ส่งเข้าฟังก์ชันคือ \$att_srl ซึ่งมีรูปแบบเป็นฟิลด์คั่นด้วย: ดังนี้

file.Zip~(ขนาด) @type; file1.zip~(ขนาด) @type ;

เมื่อส่งเข้าฟังก์ชันแล้ว ก็จะถูกแบ่งออกเป็นส่วนๆ โดยใช้ split ดังนี้ \$att_list = split (“”

\$att_srl)

จากนั้นใช้ select และ option แสดงข้อมูลโดยวนรอบด้วย foreach ดังนี้

```
echo"<select name= attach size=4>"; // ใช้ input แบบ select ให้ size =4
echo "<option>-Attachments List";
foreach ($att_list as $value) { // วนรอบแสดงรายชื่อ Attachment
  $option=regi_replace("([-[:alnum:.]])*@([-[:alnum:]]V)*$","\\ 1" $value);
  $ option= subsrt_replace ($option, ",",0,4); // ตัด id เลข 4 ตัวหน้าออก
  echo "<option value = \" $value \">$option";
} # foreach
echo "</select>";
```

โปรดสังเกตว่าตรง value ของ option กับข้อมูลรายการที่แสดงในกรอบ listbox จริงๆ นั้นแตกต่างกัน เนื่องจากมีการตัดเอาเฉพาะชื่อไฟล์กับขนาดออกมาเท่านั้น ด้วยสคริปดังนี้

```
$option=eregi_replace("[[:alnum:]]*"@"[[:alnum:]]V*$","\\1" $value);
```

ซึ่งก็คือตัด content หรือข้อความหลังเครื่องหมาย @ ทิ้งไป เช่นจาก

```
4587 file . gif~ [1204]@image/gif // 4 ตัวแรกคือ id ที่เราเติมเข้าไป
```

เหลือแค่ดังนี้ (แบ่งเป็น 2 ไฟล์โดยใช้ ๑ คั่น แล้วเอาไฟล์แรกมา)

```
4587 file .ext~ [1204]
```

และตัดเอาเลข id 4 หลักออกไปด้วย substr_replace ดังนี้

```
$option= substr_replace ($option, "",0,4); // ตัด id เลข 4 ตัวหน้าออก
```

หาก Add Attachment แล้ว view source ดู html โค้ดตรงส่วน select จะได้ข้อมูลลักษณะดังต่อไปนี้

```
<select name= attach size = 4>
<option> Attachment List-
<option value="4587 temp2.gif~{4172}@image/gif"> temp2.gif~ {4172}
<option value="4587 pointer.gif~{901}@image/gif"> pointer.gif~ {901}
</select>
```

ส่วนที่เป็น value ใน select

เหตุที่ต้องตัดเอา id ออก เพราะว่าผู้ใช้จะให้เห็นแค่ชื่อและนามสกุลไฟล์ที่เลือกเข้ามาเท่านั้น ไม่ต้องให้เห็นเลข id จะได้ไม่เกิดการสงสัย

ในกรณีที่ผู้ใช้เลือก

Remove Attachment ออก ก็จะเป็นการส่งค่าใน value นี้เข้าไปประมวลผล ดังนั้นใน value จึงต้องคงรูปแบบชื่อไฟล์เต็มๆ ที่มีเลข id นำหน้าเพราะต้องการเอา value นี้ไปหักออกจากรายชื่อนั้นเอง (ดูการ Remove Attachment)

ข้อควรรู้

หากต้องการให้ Value แสดงข้อมูลแบบเต็มๆ ไม่มีการตัดอะไรออก ให้ใช้ echo "<option value = \"'\$value'\">\$value";

กรณี Remove Attachment มากกว่า 1 ไฟล์

ในกรณีที่คลิก Attachment ทั้งหมดนั้นง่ายๆ คือ ลบไฟล์ทั้งหมดแล้วเคลียร์ค่าในตัวแปร \$att_srt ให้เหลือเพียง "" เท่านั้น

แต่ถ้าคลิกบางรายการจะยุ่งยากกว่าเพราะต้องลบไฟล์ที่เลือกนั้น และ ต้องลบเฉพาะรายชื่อ นั้นๆ ออกไปจากตัวแปร \$att_srt

การทำงานเริ่มจากการตรวจสอบปุ่มกดเป็น Remove แล้ว สคริปจะเข้ามาตรวจสอบว่าเลือกรายการ option ไฟล์ใดจากอินพุทแบบ select หรือยัง ถ้ายังไม่ได้เลือกไฟล์เราจะได้ค่า \$attach เป็น "" หรือ "_Attachment List-" ก็ให้บราวเซอร์แจ้งทำนองว่า "ยังไม่ได้เลือกไฟล์" ดังนี้

```
case "Remove" // กรณีคลิก Attachment บางรายการ
if (($attach == "") || ($attach == "_Attachments List-")) {
echo "ยังไม่ได้เลือกไฟล์ที่จะเอาออกจากรายการ <br>";
```

จุดหลักของการลบชื่อไฟล์ (ที่เลือกจากอินพุท select name = attach) ให้ออกไปจากตัวแปร \$att_srt คือใช้ eregi_replace แทนที่ \$attach (ชื่อไฟล์ที่เลือก Remove นั้นเอง) ด้วยสตริงว่างๆ เข้าไป ดังนี้

```
$att_srt = eregi_replace("$attach; ", "", $att_srt);
```

แต่เนื่องจากสตริงใน \$attach และ \$att_srt มีอักขระ [และ] อยู่ด้วยดังนั้นการแทนที่ตรงๆ เลยทันที ยังไม่ได้ ให้เปลี่ยนเป็นอักขระอื่นก่อน เช่นในที่นี้เปลี่ยน [เป็น zzxxx และ] เป็น xxxzz เสียก่อน ดังนี้

```
$att_srt = eregi_replace("[", "zzxxx", $att_srt);
$att_srt = eregi_replace("]", "xxxzz", $att_srt);
$att_srt = eregi_replace("[", "zzxxx", $att_srt);
$att_srt = eregi_replace("]", "xxxzz", $att_srt);
```

จากนั้นจึงเคลียร์ชื่อไฟล์ attachment ออกไปจากตัวแปร \$att_srt ดังนี้

```
$att_srt = eregi_replace("$attach; ", "", $att_srt); // เคลียร์เฉพาะที่เลือกให้เป็น ""
```

เมื่อเคลียร์ชื่อไฟล์ออกไปแล้วก็เปลี่ยนเครื่องหมาย [และ] ให้กลับเป็นเหมือนเดิมดังนี้

```
$att_srt = eregi_replace("[", "zzxxx", $att_srt);
$att_srt = eregi_replace("]", "xxxzz", $att_srt);
```

ลำดับสุดท้ายคือลบไฟล์ออกจากเซิร์ฟเวอร์ ดังนี้

```
list ($fname, $size) = split ("~", $attach); // แยกเอาชื่อไฟล์ออกมา
if (file_exists ($upload_dir.$fname)) { // ตรวจสอบว่ามีไฟล์อยู่หรือไม่
    unlink ($upload_dir.$fname); // ลบไฟล์
```

ฟังก์ชันนับจำนวนไฟล์ Attachment

เนื่องจากสคริปต์อนุญาตให้แนบไฟล์ Attachment มากกว่า 1 ไฟล์ แต่มีการจำกัดว่าไม่ให้แนบเกินกี่ไฟล์ เพื่อให้สะดวกต่อการใช้งาน ดังนั้นจึงสร้างฟังก์ชันนับจำนวนไฟล์ Attachment ขึ้น หลักการคือใช้ฟังก์ชัน split แยกข้อมูลโดยใช้เครื่องหมาย ; เป็นตัวคั่น จากนั้นก็ใช้ ฟังก์ชัน count นับจำนวนแล้วคืนค่าด้วย return ดังนี้

```
function count_att ($att_str) {
    $att_list = split (";", $att_str); // แยกข้อมูลเป็นฟิลด์ๆ (คั่นด้วย;)
    return count ($att_list); // นับจำนวน
```

กรณี Send ส่งเมล

การทำงานจะมีหลายขั้นตอนดังนี้

- ในกรณีที่ไม่มีไฟล์แนบ ก็จะอ่านข้อมูลในไฟล์ Attachment (ด้วยฟังก์ชัน build_mime_att) แล้วนำมาเข้ารหัส base 64 จากนั้นจัดให้เข้ารูปแบบ MIME โดยวนทำลักษณะนี้จนครบทุกไฟล์ Attachment
- เมื่อได้ข้อมูลเข้ารหัส base64 แล้วก็จะนำไปต่อกับข้อมูล MIME ของ ข้อความจดหมาย (ที่ฟังก์ชัน mimemail) เพื่อสร้างรูปแบบเมล MIME มีเนื้อความจดหมายและไฟล์แนบที่ผ่านการเข้ารหัสเรียบร้อยแล้ว (อีเมลสำเร็จพร้อมส่ง)
- ส่งเมลและรายงานสถานะส่งเมลสำเร็จหรือไม่
- ลบไฟล์ Attachment ออกจากเซิร์ฟเวอร์ (ฟังก์ชัน unlink_all_file)

2.3.4 การรับอีเมล [2]

2.3.4.1 การรับเมลโดยติดต่อ Protocol POP3

คำสั่งสำหรับ POP3 Server

คำสั่งที่ใช้ติดต่อกับเซิร์ฟเวอร์มีหลายคำสั่งด้วยกัน บางคำสั่งต้องส่งข้อมูลเข้าไปด้วย ซึ่งข้อมูลที่ส่งไปพร้อมกับคำสั่งเขาเรียกว่า Arguments เช่น RETR 2 Argument คือ 2 รายละเอียดคำสั่งต่าง ๆ มีดังนี้

ตารางที่ 2.3 สรุปคำสั่ง POP 3

คำสั่ง	หน้าที่
USER username	ส่งชื่อ username แสดงตนเมื่อเริ่มติดต่อ
PASS password	ส่งรหัสผ่าน Password
STAT	เช็คจำนวนอีเมลและขนาดของ Inbox บางแห่งแสดงขนาดเป็นเลขฐาน 8 (octet)
LIST (msgno)	ลิสจำนวนและขนาดของอีเมลแต่ละฉบับ
RETR msgno	เปิดอีเมลฉบับที่ต้องการ
DELE msgno	ทำการมาร์คเพื่อเตรียมลบอีเมลฉบับที่ต้องการยุติการเชื่อมต่อ พร้อมกับ Update หรือลบอีเมลที่มาร์คไว้ด้วย DELE
Optional Command	
TOP msgno n	เปิดอีเมลฉบับที่ msgno และลิสเนื้อความ n บรรทัด
UIDL (msgno)	ลิสอีเมลและชื่อไฟล์ (Unique ID Listing)

USER username

หน้าที่คือส่งชื่อ Username หรือชื่อสมาชิกที่เป็นเจ้าของตู้อีเมล Arguments: ชื่อหรือ username บางแห่งต้องใส่ @domain.com ด้วย เช่น

USER kob

+OK password required for kob

USER abc

-ERR no mailbox abc

PASS password

ใช้ต่อจากคำสั่ง USER เพื่อแจ้งรหัสผ่านควบคู่กับ username หากรหัสผ่านผิด ต้องวนไปเริ่มที่คำสั่ง USER ใหม่ จะส่งรหัสผ่านซ้ำแก้ตัวไม่ได้

Arguments : รหัสผ่าน (string)

เช่น

USWE kob

+OK password required for kob

PASS php

-ERR invalid password

USER kob

+OK password required for kob

PASS demo

+OK maildrop has 3 messages (784 octets)

STAT

ใช้ขณะอยู่ในสภาวะ Transaction state หลังจากที่เราตรวจสอบรหัสผ่านถูกต้องแล้ว หน้าทีคือเช็คจำนวนอีเมลล์และขนาดของตู้อีเมลล์ (Inbox หรือเรียกอีกอย่างว่า maildrop)

เมื่อถูกต้องเซิร์ฟเวอร์จะตอบรับด้วย +OK เว้น 1 space จำนวนอีเมลล์ที่อยู่ในตู้เว้นวรรค 1 Space และขนาดของตู้ (ส่วนใหญ่บอกเป็นเลขฐานแปด Octets)

Arguments: ไม่มี

เช่น

STAT

+OK 2 520

LIST (msgno)

ใช้ขณะอยู่ในสภาวะ Transaction state เพื่อลิสหมายเลขและขนาดของอีเมลล์ในตู้ Inbox (เรียกว่า scan listing) ไม่ได้แสดงรายละเอียดข้อความในอีเมลล์

หากไม่ระบุหมายเลขอีเมลล์ ก็จะเป็นการลิสอีเมลล์ทั้งหมด แต่ถ้าระบุหมายเลข จะเป็นการลิสดูขนาดเฉพาะอีเมลล์ฉบับนั้น ๆ

หากถูกต้องเซิร์ฟเวอร์จะตอบรับ +OK พร้อมกับลิสหมายเลขและขนาดของอีเมลล์ฉบับนั้นๆ (ส่วนใหญ่บอกเป็นเลขฐานแปด Octets) โดยข้อมูลที่ส่งมาจะมีหลายบรรทัดขึ้นอยู่กับจำนวนอีเมลล์ หากสิ้นสุดจะจบด้วยจุด

Arguments: หมายเลขอีเมลล์ หรือไม่ระบุก็ได้

เช่น

LIST

+OK 2 messages (562 octets)

1 200

2 362

LIST 2

+OK 2 362

LIST 4

-ERR no such message.

RETR msgno

สำหรับเปิดดูหรือแสดงรายละเอียดอีเมลล์ฉบับที่ต้องการ (ดูรูป 3 ประกอบ) หากมีอีเมลล์
หมายเลขที่ต้องการอยู่ เซิร์ฟเวอร์จะส่งรายละเอียดทั้งเฮดเดอร์และเนื้อจดหมายออกมา (body message)
โดยสิ้นสุดจดหมายจะมีเครื่องหมายจุด

Arguments: หมายเลขอีเมลล์

เช่น

RETR 1

+OK 250 octets

รายละเอียดเฮดเดอร์จดหมาย

รายละเอียดเนื้อความในจดหมาย

DELE msgno

สำหรับลบอีเมลล์ฉบับที่ต้องการ โดยเซิร์ฟเวอร์จะทำการมาร์คไว้ก่อนไม่ได้ลบทันที เมื่อสั่ง
QUIT แล้วถึงจะลบให้

Arguments: หมายเลขอีเมลล์ที่ต้องการลบ

เช่น

DELE 2

+OK message 2 deleted

DELE 3

+OK message 3 deleted

DELE 4

-ERR no such messages (ไม่มีอีเมลที่อยู่ในเนกซ์บ็อก)

DELE 2

-ERR message 2 already deleted (อีเมลนี้ถูกมาร์คไว้แล้ว)

NOOP

คำสั่งส่งให้เซิร์ฟเวอร์บอกว่าไม่มีการทำอะไร

Arguments: ไม่มี

RSET

สำหรับรีเซท ซึ่งจะมีผลคือ หากมีอีเมลใดที่มาร์คด้วย DELE แล้ว คำสั่งนี้จะยกเลิกการมาร์ค (unmarked)

Arguments: ไม่มี

เช่น

RSET

+OK maildrop has 2 messages (462 octets)

QUIT

ใช้สำหรับบอกยุติการติดต่อ หากมีการใช้ DELE มาร์คอีเมลที่จะลบไว้ก่อนหน้านี้ เซิร์ฟเวอร์ก็จะทำการลบอีเมลนั้นออกไปด้วย

อย่างไรก็ตามเมื่อ QUIT ขณะอยู่ใน Transaction state ก็จะเป็นการเข้าสู่ Update state หากส่งคำสั่ง QUIT ในช่วง Authorization state (ยังไม่ผ่าน password ก็ไม่ถือว่าเข้าสู่ Update state แต่อย่างไรก็ตาม)

Arguments: ไม่มี

เช่น

QUIT

+OK server signing off (3 messages left)

TOP msgno line

สำหรับเปิดอ่านหรือแสดงรายละเอียดอีเมลคล้าย RETR แต่คำสั่ง TOP นี้เหมาะสำหรับอีเมลที่ยาว ๆ เพราะหากใช้ RETR การลิสก็จะยาวล้นจอ ไม่เห็นส่วนต้น ๆ เนื่องจากหมายถึงการใช้คำสั่ง TOP จะสามารถระบุจำนวนบรรทัดเนื่องจากหมายถึงต้องการได้

สำหรับคำสั่งนี้เป็นประเภท Optional Command อาจจะมีในเซิร์ฟเวอร์บางค่ายเท่านั้น

Arguments: หมายเลขอีเมลล์ จำนวนบรรทัดเนื้อความ
 เช่น เปิดอีเมลล์ฉบับที่ 5 และให้แสดงเพียง 10 บรรทัด
 LIST 5 10
 +OK
 แสดงรายละเอียดเซดเดอร์
 บรรทัดว่าง ๆ
 แสดงรายละเอียดเนื้อความ 10 บรรทัดแรก

UIDL (msgno)

สำหรับแสดงชื่ออีเมลล์ (Uniaue ID Listing) หรืออีกนัยหนึ่ง คือชื่อไฟล์ที่เก็บอีเมลล์นั่นเองคล้ายกับคำสั่ง LIST แต่กรณี UIDL นี้จะลิสหมายเลข ชื่อ ส่วนคำสั่ง LIST นั้นจะลิสหมายเลขและขนาด (UIDL เป็น Optional Command เช่นกัน)

Arguments: หมายเลขอีเมลล์ หรือ ไม่ระบุก็ได้ (หาก ไม่ระบุถือว่าลิสทั้งหมด)
 เช่น
 UIDL
 +OK
 1 0xz7uicvkm0021
 274yuopkegh12w5
 UIDL 2
 +OK 2 74yuopkegh12w5

ตัวอย่างรวม

ตัวอย่างนี้แสดงให้เห็นการสื่อสารกับ POP 3 Server อย่างรวม ๆ โดยตัวหนาคือคำสั่งที่เราส่งไป (ตัวเข้มคือที่เราพิมพ์เข้าไป ส่วนตัวเอนคือที่เมลท์เซิร์ฟเวอร์ตอบรับ Reply กลับมา)

Telnet หรือ ใช้ PHP เชื่อมต่อกับ POP 3 Server
 USER kob
 +OK password required for kob
 PASS mailphp
 +OK maildrop has 3 messages (784 octets)
 STAT
 +OK 2 562
 LIST

+OK 2 messages (562 octets)

1 200

2 362

RETR 1

+OK 200 octets

รายละเอียดเซคเตอร์จดหมาย

รายละเอียดเนื้อความในจดหมาย

(จบด้วยจุด.)

DELE 1

+OK message 1 deleted

E

DELE 2

+OK message 2 deleted

Quit

+OK Server signing off (maildrop empty)

2.3.4.2 การรับเมลโดยใช้ PHP ติดต่อในระดับ Socket

ลักษณะการใช้ PHP ในการติดต่อระดับ Socket ของการรับ (POP3) และการส่ง (SMTP) จะใช้ฟังก์ชัน ในการติดต่อตัวเดียวกัน ต่างกันตรงที่ Port ที่ใช้ในการเชื่อมต่อ ของการส่งจะเป็น Port 25 แต่ในส่วนของการรับจะใช้ Port 110 ตัวอย่างการใช้ฟังก์ชัน เช่น

```
$sock = fsockopen ("localhost", 110);
```

```
fputs($sock, "USER $user");
```

```
fgets($sock, "USER $user");
```

ซึ่งรายละเอียดของในแต่ละฟังก์ชัน สามารถดูได้ในหัวข้อ การส่งอีเมลในระดับ Socket

2.3.4.3 ทำความรู้จักกับ IMAP ที่เกี่ยวกับ PHP

IMAP มาจากคำว่า Internet Message Access Protocol เริ่มพัฒนาโดยมหาวิทยาลัย Stanford University ตั้งแต่ 1986 จากนั้นก็มีมหาวิทยาลัยและองค์กรอื่นๆ ร่วมกันพัฒนาต่อกันมาเรื่อยๆ

องค์ประกอบโดยรวมของ IMAP นั้นมีส่วนที่ทำงานร่วมกัน 2 ส่วนดังนี้
IMAP Server สำหรับเป็นเครื่องบริการหรือ Server ซึ่งถ้าต้องการให้เครื่องใดทำงานเป็น IMAP Server ก็จะต้องหาโปรแกรมประเภท IMAP Server มาติดตั้ง เช่น ถ้าในระบบปฏิบัติการ Linux ที่ใช้ โปรแกรม

Qmail เป็น Mail Server เราสามารถหาโปรแกรม IMAP Server ที่ชื่อว่า Courier-IMAP มาลงเสริม เพื่อให้เครื่อง Mail Server นั้นสามารถรองรับการทำงานของ IMAP ได้ ซึ่งคุณสามารถกำหนดค่า Inbox เป็นลักษณะของ Maildir ได้ ท่านสามารถหาอ่านข้อมูลเพิ่มเติมได้มา <http://www.qmail.org>

IMAP Client สำหรับเป็นตัวถูกเข้าไปเชื่อมต่อแล้วรับข้อมูลจาก Server ซึ่งการใช้ฟังก์ชัน IMAP ใน PHP นั้นจะเป็นติดต่อกับส่วนนี้ ทำให้เมื่อเราใช้ ฟังก์ชัน IMAP ใน PHP เราจำเป็นต้องติดตั้งโปรแกรม IMAP Client ด้วย โดยท่านที่ใช้ Linux เป็น Server สามารถหาโปรแกรม IMAP Client ที่ชื่อว่า imap-2001a.tar.Z ได้จาก <ftp://ftp.cac.washington.edu/imap>

และเนื่องจาก IMAP นั้นได้มาที่หลัง POP3 ดังนั้น IMAP Client ส่วนใหญ่จะติดต่อกับ POP3 ได้ รวมทั้งฟังก์ชัน IMAP ใน PHP ที่เรากำลังจะศึกษากันนี้ก็ได้เชื่อมต่อกับ POP3 Server ได้ด้วย

ข้อดีของ IMAP นั้นมีหลายประการ เช่น เครื่อง Client สามารถจัดการกับ Message ได้โดยไม่ต้อง Download Message มาเก็บไว้ที่เครื่อง Client ทั้งหมด หรืออีกประการคือมีการแยกแยะได้ว่า Message อีเมลฉบับใดเปิดอ่านแล้ว และฉบับใดยังไม่ได้เปิดอ่าน โดยบางท่านอาจสงสัยว่า web based mail ฟรี ส่วนใหญ่เช่น Hotmail Yahoo หรืออื่น ๆ ที่ใช้ POP3 ก็สามารถทราบได้ว่าฉบับใดเปิดอ่านแล้วฉบับใดยังไม่ได้เปิดอ่าน ซึ่งลักษณะดังกล่าวนี้จะอาศัยการดึงอีเมลมาเก็บไว้ที่ Database จากนั้นจึงใช้วิธีมาร์คหรือเก็บข้อมูลใน Database นั้นว่ามีมีการคลิกเข้ามาดูหรือยัง

ถึงแม้ IMAP Server จะมีจุดเด่นเหนือกว่า POP3 ก็ตาม แต่ในปัจจุบันนี้ยังมีเครื่องที่ใช้ POP3 Server มากกว่า อย่างไรก็ตาม ในอนาคตคงมีการใช้งาน IMAP Server กันมากขึ้น ท่านสามารถอ่านรายละเอียดเพิ่มเติมได้ที่ <http://www.imap.org>

2.3.4.4 ทดสอบการใช้ฟังก์ชัน IMAP ใน PHP

เมื่อเราสนใจที่จะเขียน Script PHP ให้เป็น Client เข้าไปติดต่อกับ Mail Server และเนื่องจากเครื่อง Server ที่รัน PHP นั้นอาจจะไม่สามารถรันฟังก์ชัน IMAP ใน PHP ได้ เพราะบางเครื่องก็ไม่ได้ถูกติดตั้งโปรแกรม IMAP Client ไว้ ดังนั้นเราจะต้องทำการตรวจสอบก่อนว่า เครื่องสามารถใช้ฟังก์ชัน IMAP ใน PHP ได้หรือไม่ โดยสามารถตรวจสอบได้โดยการใช้วิธีต่างๆ ดังนี้

1. ใช้ฟังก์ชัน `phpinfo()`
2. ใช้ฟังก์ชัน `function_exists`
3. ใช้ฟังก์ชัน `imap_open()`

ใช้ฟังก์ชัน `phpinfo()`

เขียน Script ง่ายๆ ดังต่อไปนี้ แล้วทดลองใช้ Browser เรียก จากนั้นสังเกต ตรงส่วน IMAP จะต้องมีลักษณะดังรูป

imap

IMAP Client Version	4.1
---------------------	-----

ใช้ฟังก์ชัน `function_exists`

เป็นวิธีที่ใช้ `function_exists` ตรวจสอบเช็คว่ามี `imap_open` อยู่หรือไม่ โดยมีรายละเอียด Script ดังนี้

```
<?php
if(function_exists('imap_open')){
    echo"สามารถใช้ ฟังก์ชัน imap_open ได้";
}else{
    echo"ไม่สามารถใช้ ฟังก์ชัน imap_open ได้";
}
?>
```

ใช้ฟังก์ชัน `imap_open`

วิธีนี้เป็นอีกวิธีหนึ่ง คือเขียน Script `imap_open` เพื่อลองเชื่อมต่อ POP3 Server โดยตรง ดังนี้(ยังไม่ต้องสนใจ `username,password`)

```
<?php
$inbox=imap_open('{localhost:110/pop3}INBOX','username','password');
?>
```

หาก PHP ไม่สามารถใช้ฟังก์ชัน `imap_openw` ได้ จะมีข้อความ error ดังต่อไปนี้

```
Fatal error: Call to undefined function: imap_open() in
C:\Apache\htdocs\mail\imap_opentest.php on line 2
```

ถ้าฟังก์ชัน IMAP ใช้ได้ แต่ว่ารหัสผ่านไม่ถูกหรือติดต่อ POP3 Server ไม่ได้ จะมีข้อความ error ดังต่อไปนี้

Warning: Couldn't open stream {localhost:110/pop3}INBOX in
C:\Apache\htdocs\mail\imap_opentest.php on line 2

แสดงว่าฟังก์ชันที่เกี่ยวกับ IMAP ทำงานได้ เพียงแต่ error เนื่องจาก user หรือ password ผิด หรือ POP3 Server ยังไม่ได้ทำงานเท่านั้น

หมายเหตุ: สำหรับการติดตั้ง IMAP บน Linux และ Windows นั้นจะมีเนื้อหาอยู่ใน บทที่ 3

2.3.4.5 การใช้ฟังก์ชันของ PHP ที่เกี่ยวข้องกับ IMAP แทน POP3

ฟังก์ชันเกี่ยวกับ IMAP ส่วนใหญ่เริ่มมีมาตั้งแต่ PHP3 เป็นต้นมา บางฟังก์ชันรองรับ POP3 Server บางฟังก์ชันรองรับเฉพาะ IMAP Server ซึ่งในที่นี่จะขอยกส่วนที่ใช้กับ POP3 มาอธิบายดังนี้

imap_open	เชื่อมต่อกับ Mail Server (เปิด Mail Box)
imap_last_error	แสดง Error message ถ้าสุด
imap_close	ยุติการเชื่อมต่อ(ปิด Mail Box)
imap_mailboxmsginfo	อ่านข้อมูลของ mailbox เช่น จำนวนอีเมล และขนาดของ Inbox
imap_num_msg	นับจำนวนอีเมลในตู้ mailbox
imap_headerinfo	อ่านข้อมูล header ของอีเมล
imap_headers	อ่าน header ของอีเมลทั้งหมดใน mailbox
imap_fetchheader	อ่านข้อมูล header ของอีเมล
imap_mime_header_decode	ถอดรหัส MIME header เช่น เมล์ที่ส่งจาก Outlook ฯลฯ
imap_fetchstructure	อ่านโครงสร้างของอีเมล
imap_body	อ่านเนื้อหาของจดหมาย อีเมล(message body)
imap_fetchbody	อ่านข้อมูล message body ส่วนที่ต้องการ เช่น เนื้อความ ฯลฯ
imap_qprint	ถอดรหัสเปลี่ยนข้อความจดหมายให้อ่านได้เช่น เมล์จาก Outlook
imap_delete	มาร์คอีเมลที่ต้องการลบ
imap_expunge	ลบอีเมลที่ได้มาร์คไว้

เริ่มต้นกับฟังก์ชัน `imap_open` เพื่อเชื่อมต่อสู่ mailbox

ทำหน้าที่เริ่มต้นการเชื่อมต่อเข้ากับ MailServer ทุกครั้งต้องการทำอะไรกับ MailServer นั้น จะต้องเริ่มต้นด้วยฟังก์ชัน `imap_open` ก่อนเสมอ เช่น การเช็คเมลล์ อ่านเมลล์ และ ลบเมลล์ ฯลฯ

เมื่อต้องการติดต่อกับ POP3Server นั้น จะใช้พอร์ต 110 โดยใช้รูปแบบดังตัวอย่างต่อไปนี้

```
$mbox=imap_open("{yourpop3.server.com:110/pop3}INBOX","user_id","password");
```

เมื่อต้องการติดต่อกับ IMAPServer นั้น จะใช้พอร์ต 143 โดยใช้รูปแบบดังตัวอย่างต่อไปนี้

```
$mbox=imap_open("{yourimap.server.com:143}INBOX","user_id","password");
```

หาก user,password ถูกต้อง จะได้ค่าอ้างอิงของ mailbox มา โดยเก็บค่าไว้ที่ตัวแปร \$mbox ซึ่งค่านี้เป็นค่าอ้างอิงที่จำเป็นต้องใช้ในการอ่านข้อมูลต่างๆ เช่น การเปิดอ่านอีเมลล์ ลบอีเมลล์ ฯลฯ (ค่านี้เปรียบเสมือน file pointer เมื่อเราเปิดไฟล์อ่านหรือเขียนนั่นเอง)

ในทางปฏิบัติ เรามักเขียน Script ให้ตรวจสอบเช็คก่อนว่าเปิด mailbox ได้หรือไม่ โดยเช็คจากค่า \$mbox ถ้าได้ถึงจะทำงานต่อ แต่ถ้าไม่ได้ก็จะให้แจ้ง Error ให้ทราบ หรือสั่งให้หยุด โปรแกรม

แสดง Error ด้วย ฟังก์ชัน `imap_last_error`

สำหรับแสดงข้อผิดพลาด(Error message) ล่าสุด ปกติมักใช้บอกเพื่อให้ทราบว่า มี Error อะไรเกิดขึ้น รูปแบบการใช้งานดังตัวอย่างต่อไปนี้

```
$mbox= @imap_open("{yourpop3.server.com:110/pop3}INBOX","user_id","password") or die("เชื่อมต่อ MailServer ไม่ได้ กรุณาตรวจสอบค่าตัวแปร user,password:".imap_last_error());
```

จากตัวอย่างดังกล่าว หากเชื่อมต่อกับ MailServer ไม่ได้ ฟังก์ชัน `imap_last_error()` จะมีความแจ้งให้ทราบ

ยุติการเชื่อมต่อ ปิด Mailbox โดย ฟังก์ชัน `imap_close`

สำหรับยุติการเชื่อมต่อกับ mailbox เช่น เมื่อคุณใช้ ฟังก์ชัน `imap_open` ไปแล้ว โดยให้ค่ากับตัวแปร \$mbox ไว้ หากคุณต้องการยุติการเชื่อมต่อกับ MailServer ก็ให้เรียกใช้ฟังก์ชัน ดังต่อไปนี้

```
imap_close($mbox);
```

ในกรณีที่ปิด mailbox พร้อมกับลบอีเมลล์ที่มาร์คไว้ด้วย `imap_delete` ให้กำหนด `CL_EXPUNGE` ด้วยดังนี้

```
imap_delete($mbox,$msgno);
imap_close($mbox,CL_EXPUNGE);
```

อ่านข้อมูล mailbox ด้วยฟังก์ชัน `imap_mailboxmsginfo`

อ่านข้อมูล mailbox เช่น จำนวนอีเมล, mailbox size ดังตัวอย่างเช่น

```
<?php
$mbox=imap_open('{your.imap.host}INBOX','username','password') or die("can't
connect:".imap_last_error());
$check = imap_mailboxmsginfo($mbox);
if($check){
    echo"Message : ".$check->Nmsgs."<br>\n"; //จำนวนอีเมล
    echo"Size : ".$check->Size."<br>\n"; //inbox size
    echo"Date : ".$check->Date."<br>\n"; //Date time
}else{
    echo"imap_check failed:".imap_last_error()."<br>\n";
}
?>
```

เช็คจำนวนอีเมลในตู้ด้วย ฟังก์ชัน `imap_num_msg`

สำหรับนับจำนวนอีเมลใน mailbox เช่น

```
imap_num_msg($mbox);
```

อ่านข้อมูล header ที่ต้องการด้วย ฟังก์ชัน `imap_headerinfo`

สำหรับอ่านข้อมูล header ของอีเมลที่ต้องการ ได้แก่ From, To, Subject, Date ฯลฯ ตัวอย่างการใช้งาน เช่น

```
<?php
$mbox=imap_open('{your.imap.host}INBOX','username','password') or die("can't
connect:".imap_last_error());
$msgno=1; //หมายเลข อีเมล
```

```

$headinfo = imap_headerinfo($mbox,$msgno);
$from=$headinfo->fromaddress;      //From
$subject=$headinfo->subject;        //Subject
$date=date("D,d M y -G:i",$headinfo->udate); //Date
$to=$headinfo->toaddress;            //To:
$cc=$headinfo->ccaddress;            //cc:
$bcc=$headinfo->bccaddress;         //bcc:
$replyto=$headinfo->reply_toaddress; //Replyto
imap_close($mbox);
echo"From : $from <br>\n";
echo"Subject : $subject <br>\n";
echo"Date : $date<br>\n";
?>

```

เรียก header ในตู้ mailbox ด้วย ฟังก์ชัน `imap_headers`

อ่านข้อมูล header ทุกอีเมล ใน mailbox ซึ่งสามารถประยุกต์ใช้กับอีเมลได้ โดยในที่นี้จะขอ
ยกตัวอย่างในบทต่อไป

อ่าน header ของอีเมลด้วย ฟังก์ชัน `imap_fetchheader`

ใช้สำหรับอ่าน header ของอีเมลฉบับที่ต้องการ (header รวมแบบเต็มรูปแบบ ต่างกับที่ใช้
`imap_headerinfo` ที่แยกออกเป็นส่วนๆ) เช่น

```

<?php
$mbox=imap_open('{your.imap.host}INBOX','username','password') or die("can't
connect:".imap_last_error());
$msgno=1;
$hd=imap_fetchheader($mbox,$msgno);
echo"<br>header ".nl2br(htmlspecialchars($hd));
imap_close($mbox);
?>

```

ถอดรหัส header ด้วย ฟังก์ชัน `imap_mime_header_decode`

ใช้สำหรับถอดรหัส header อีเมลที่ส่งมาโดยเข้ารหัส MIME ไว้ เช่น ส่งจากโปรแกรม Outlook ฯลฯ เพราะถ้าไม่ถอดรหัสแล้วจะไม่สามารถอ่านได้ โดยมีการใช้ฟังก์ชัน ดังนี้

```
$msgno=1;
$headinfo=imap_headerinfo($mbox,$msgno);
echo"mail number : $msgno<br>\n";
echo"From : ".htmlspecialchars($headinfo->fromaddress)."<br>\n";
echo"Date : ".date("r",$headinfo->update)."<br>\n";
echo"To : ".htmlspecialchars($headinfo->toaddress)."<br>\n";
$subj_arr=imap_mime_header_decode($headinfo->subject); //ถอดรหัส
for($i=0;$j<count($subj_arr);$j++){
    $subject .= $subj_arr[$j]->text;
    echo"<br>subj หลัง decode : ".htmlspecialchars($headinfo->subject);
}
?>
```

อ่านข้อมูลโครงสร้างอีเมลด้วย ฟังก์ชัน `imap_fetchstructure`

สำหรับอ่านข้อมูลโครงสร้างอีเมล เช่น มีขนาด(SIZE) กี่ไบต์ และมีข้อมูลอยู่ที่ส่วน(parts) ปกติแล้วมีอีเมลที่ส่งมาเป็นข้อความธรรมดาจะมี mailbody อยู่ส่วนเดียวคือ ที่เป็นเนื้อความจริงๆ แต่ถ้าเป็นอีเมลที่มี Attachment แนบมาก็ถือว่า mailbody มีมากกว่า 1 ส่วน

ส่วนการใช้งาน เราจะนำข้อมูลเข้าฟังก์ชัน `imap_fetchstructure` ก่อน จากนั้นจึงใช้ฟังก์ชัน `imap_fetchbody` อ่านข้อมูล mailbody ในส่วนที่ต้องการ เช่น อ่านเนื้อจดหมาย ,เปิด Attachment แรก ,เปิด Attachment อันถัดต่อไป ฯลฯ

ตัวอย่างการใช้งาน เช่น

```
$msgno=1;
$struct=imap_fetchstructure($mbox,$msgno); //อ่าน โครงสร้าง
echo"<br>byte : ".htmlspecialchars($struct->bytes);
$part=$headinfo->parts;
echo"<br>จำนวน parts : ".headinfo->parts; //จำนวน part
$partno=1;
$body1=imap_fetchbody($mbox,$msgno,$partno); //อ่าน part ที่ต้องการ
echo $body1;
```

นอกจากนี้ยังมีหน้าที่สำคัญคือสามารถเช็คค่า Transfer encoding เข้ารหัสมาใน body อีเมลส่วนนั้น ๆ เป็นแบบใด (\$struct->parts[\$i]->encoding;)ซึ่งค่าที่ได้คือ 0 ถึง 5 มีความหมายดังต่อไปนี้

- 0 7BIT
- 1 8BIT
- 2 BINARY
- 3 BASE64
- 4 QUOTED-PRINTABLE
- 5 OTHER

สำหรับตัวอย่างในการแยก parts ในแต่ละส่วน เพื่อแยก Download ไฟล์ Attachment ได้นั้น สามารถดูได้จากบทต่อไป

อ่านข้อมูล mailbody ทั้งหมดได้จาก imap_body

สำหรับเปิดข้อมูลส่วนที่เป็น mailbody อย่างเดียว(header ละท้าย) ซึ่ง mailbody นี้รวมไปถึง เนื้อความจดหมาย และ Attachment ด้วย ยกตัวอย่าง เช่น

```
<?php
    if(empty($msgno)) $msgno=1;
    $mbox=imap_open('{your.imap.host}INBOX','username','password');
    echo"<hr>";
    $body=imap_body($mbox,$msgno); //อ่านข้อมูลส่วนที่เป็น mail body
    echo nl2br(nl2br);
    imap_close($mbox);
?>
```

ปกติจะใช้ imap_fetchbody มากกว่าเพราะ imap_body นี้จะได้ข้อมูลมาทั้งหมด ส่วนimap_fetchbody สามารถเลือกเอา body ส่วนใดมาก็ได้ เช่นเนื้อความจดหมาย,Attachment1, Attachment2 ฯลฯ เลือกข้อมูล mailbody ส่วนที่ต้องการด้วย imap_fetchbody

ในกรณี mailbody มีหลายส่วน(multipart) เช่นอีเมลที่มี Attachment ดิคมมา เราจะใช้ฟังก์ชันนี้ อ่านแล้วนำข้อมูลใน mailbody ส่วนที่ต้องการได้ เช่น

```

$inbox=imap_open('{your.imap.host}INBOX','username','password');
$msgno=1;
$structure=@imap_fetchstructure($inbox,$msgno);
$totalpart=sizeof($structure->parts);
//part 0=header ถ้ามี 1 part ต้องแสดง $totalpart+1 เพราะ object=0
if($totalpart == 0){
    echo"Single part<br>";
    $part=1;
    $body=imap_fetchbody($inbox,$msgno,$part);
    echo"body : ".nl2br($body)."<hr>";
} else { //totalpart > 0 ถ้าอีเมลมี โครงสร้างเป็นแบบหลาย part(Multipart)
    echo"multipart :";
    for($i=1;$i<=$totalpart;$i++){
        echo"i : $i ";
        $body=imap_fetchbody($inbox,$msgno,$i); //อ่าน body ส่วนที่ $i
        $disposition=$structure->parts[$i-1]->disposition;
        $encoding=$structure->parts[$i-1]->encoding;
        $type=$structure->part[$i-1]->type;
        echo"BODY $body";
    }
}

```

ถอดรหัสข้อความด้วย `imap_qprint`

สำหรับถอดรหัสเปลี่ยนข้อความจดหมายให้เป็นแบบ 8 bit string เช่น อีเมลที่มาจาก Outlook ต้องใช้ `imap_qprint` เพื่อเปลี่ยนเป็นข้อความให้สามารถอ่านได้

โดยปกติแล้วจะใช้ฟังก์ชัน `imap_fetchstructure` ตรวจสอบโครงสร้าง encoding ก่อนถ้าได้รหัส 4 ก็ใช้ `imap_qprint` เปลี่ยนข้อความเสีย ดังนี้

```

$structure=@imap_fetchstructure($inbox,$msgno);
$body=imap_fetchbody($inbox,$msgno,$partno);
$disposition=$structure->part[$i-1]->disposition;

```

```

$encoding=$structure->parts[$i-1]->encoding;
$type=$structure->part[$i-1]->type;
if($encoding==4){           //ถ้ารหัสแบบ QUOTED-PRINT(QP)
    $body=imap_qprint($body);
}
echo $body;

```

ลบอีเมล `imap_delete`, `imap_expunge`

อีเมลที่ต้องการลบออกจากเมล MailServer ให้ใช้ `imap_delete` ทำการมาร์ค โดยระบุหมายเลขอีเมลที่ต้องการไว้ก่อน จากนั้นใช้ ฟังก์ชัน `imap_expunge` ลบทิ้ง ดังตัวอย่างเช่น

```

$mbx=imap_open('{your.imap.host}INBOX','username','password');
$msgno = 3;           //หมายเลขอีเมลที่ต้องการลบ
imap_delete($mbx,$msgno); //มาร์คอีเมลที่ต้องการลบ
imap_expunge($mbx); //ลบอีเมลที่มาร์คไว้

```

นอกจากนี้ยังใช้ `imap_close($mbx,CL_EXPUNGE)`; ทำหน้าที่ลบแทนก็ได้เช่นกัน

```

$mbx=imap_open('{your.imap.host}INBOX','username','password');
$msgno = 3;           //หมายเลขอีเมลที่ต้องการลบ
imap_delete($mbx,$msgno); //มาร์คอีเมลที่ต้องการลบ
imap_close($mbx,CL_EXPUNGE); //ยุติการเชื่อมต่อและลบอีเมลที่มาร์คไว้

```

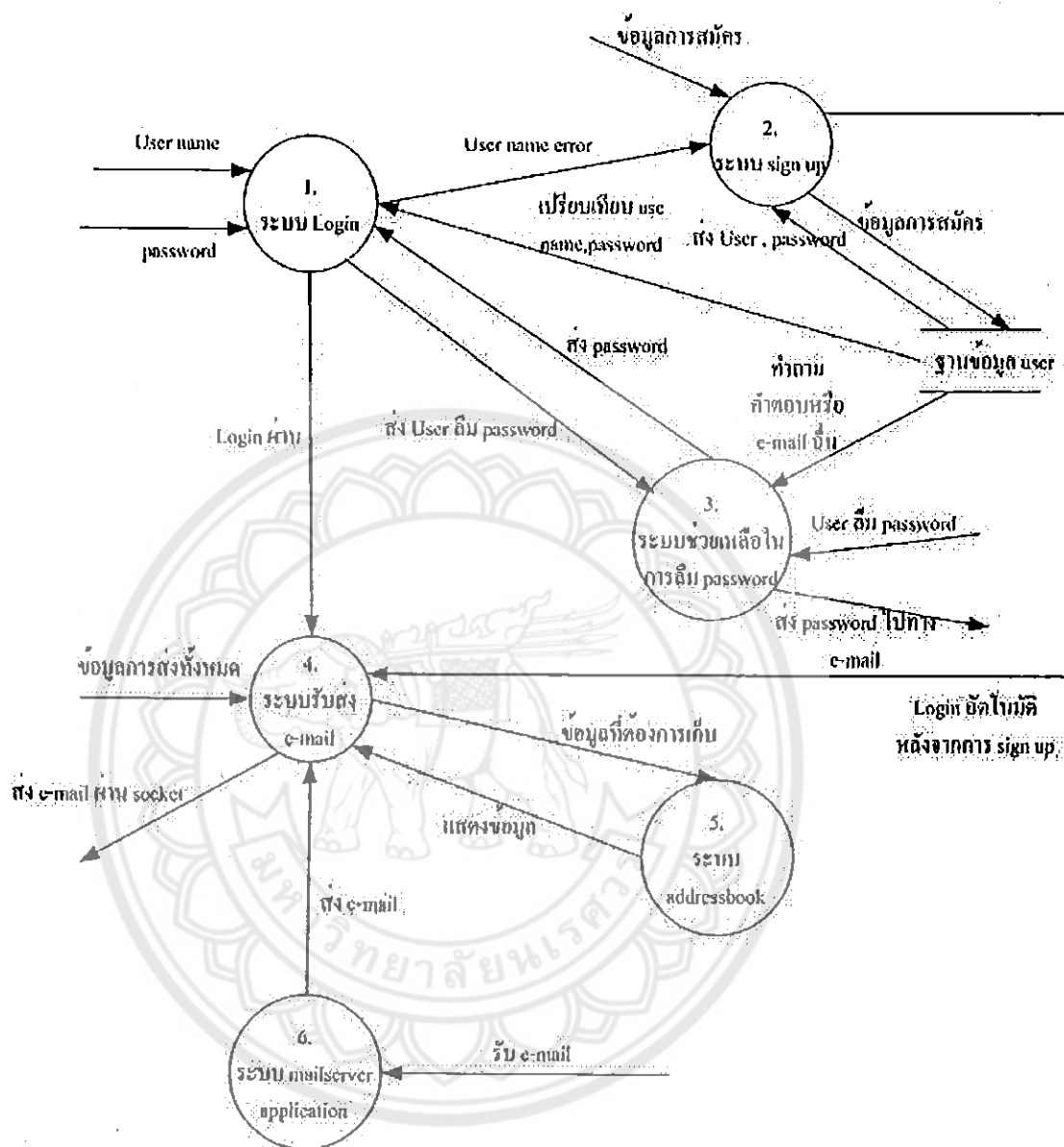
บทที่ 3

การออกแบบโปรแกรม

ในการทำโครงงานในครั้งนี้ คณะผู้จัดทำได้ดำเนินงานต่างๆ เป็นขั้นตอน ดังต่อไปนี้

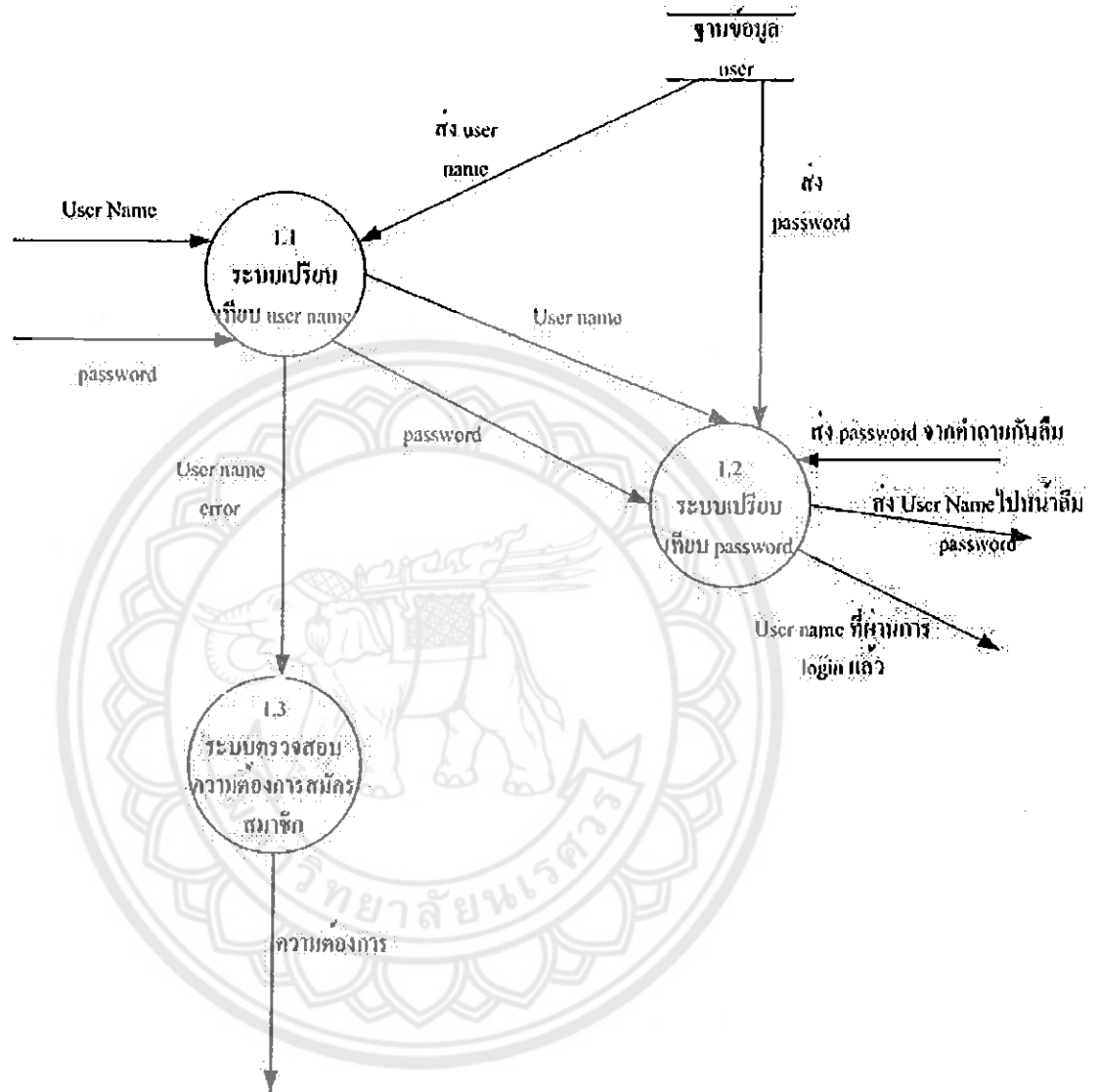
- ศึกษา ระบบการทำงานของ Protocol ที่เกี่ยวข้องกับอีเมลล์ เช่น SMTP ,POP3 ,IMAP, ฯลฯ และ การใช้ภาษา PHP ในการเชื่อมต่อกับ Protocol ดังกล่าว
- ศึกษา การเซตอัพ Linux Redhat 7.3 ให้ทำงานเป็นเครื่อง Server ต่างๆ ที่จำเป็นสำหรับการทำ web based mail เช่น DNS Server ,Web Server ,Ftp Server ,Mail Server (Support POP3 Server ,IMAP Server,IMAP Client) ,SSH ,Database Server และได้ทำการติดตั้ง Server ต่างๆ อย่างสมบูรณ์
- ออกแบบการทำงานของ web based mail โดยใช้ Data Flow Diagram และระบบฐานข้อมูล โดยใช้ ER-Model
- ออกแบบหน้าเว็บเพจ และเขียนโปรแกรม โดยแบ่งเป็น แต่ละ module คือ Register Add New User ,Login Logout Security ,Sendmail ,Inbox ,Addressbook
- ทำการทดลอง โปรแกรมบนเครื่อง Server เพื่อหาข้อผิดพลาดของโปรแกรม แล้วทำการแก้ไข โปรแกรมส่วนที่ผิดพลาดและยังไม่สมบูรณ์

3.1 ออกแบบการทำงานของ web based mail โดยใช้ Data Flow Diagram



รูปที่ 3.1 ภาพรวมของระบบรับ-ส่ง e-mail

1. ระบบ Login



รูปที่ 3.2 ระบบ Login

การทำงานในส่วนนี้ จะทำการตรวจสอบ Username และ Password มีขั้นตอนการทำงานดังนี้

1.1 ระบบเปรียบเทียบ Username

ในส่วนนี้เราจะรับค่า username and password มาแต่จะทำการตรวจสอบ Username ก่อน โดยจะดึงข้อมูล username จากฐานข้อมูลของ user ที่สมัครไว้แล้ว ซึ่งมี 2 กรณี

- ถ้ามี username อยู่ในฐานข้อมูล จะส่ง username and password ไปยัง (1.2)
- ถ้าไม่มี username อยู่ในฐานข้อมูล จะแจ้ง error และแนะนำให้สมัครก่อน ส่งต่อไปยัง (1.3)

1.2 ระบบเปรียบเทียบ password

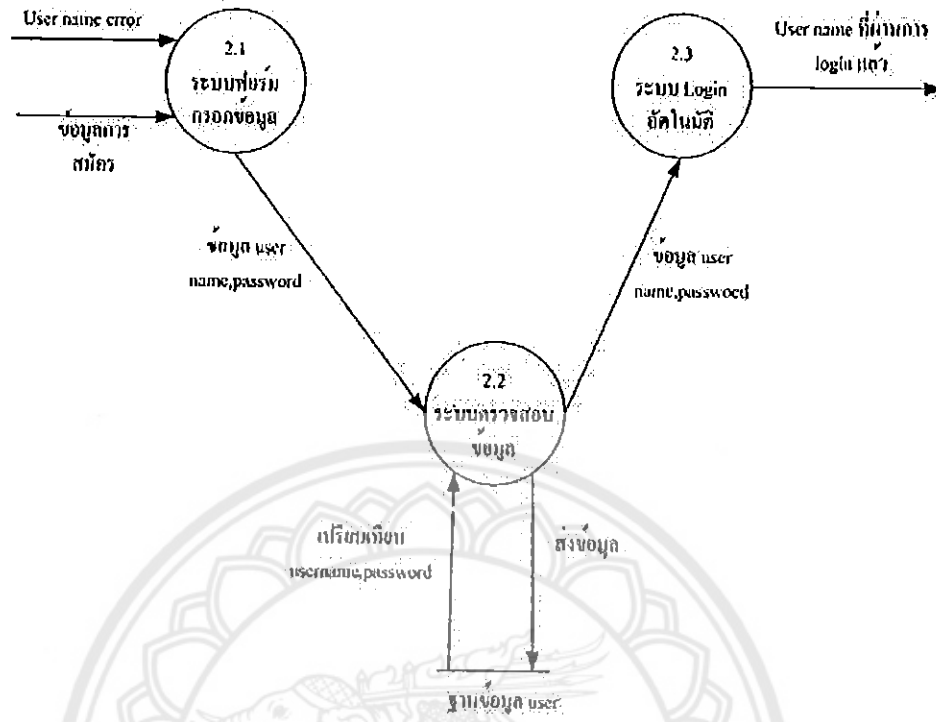
ในส่วนนี้จะทำการตรวจสอบ password ว่าตรงกับ username ที่รับค่ามาในตอนแรกหรือไม่ โดยจะดึงข้อมูล ของ password จากฐานข้อมูลของ user ที่สมัครไว้แล้ว จะได้ผล 2 อย่างคือ

- กรณีที่ username and password ถูกต้อง ก็จะไปในส่วนของการรับส่งอีเมลล์ (4)
- กรณีที่ลืม password จะต้องไปในส่วนที่ (3) ส่วนคำถามกันลืม เพื่อตอบคำถามเพื่อจะได้ password ที่ต้องการ

1.3 ระบบตรวจสอบความต้องการสมัครสมาชิก

จะถามความสมัครใจในการสมัคร ถ้าต้องการจะส่งไปยังส่วน การ sign up เพื่อทำการสมัคร ถ้าไม่ต้องการสมัครก็ไปยังหน้าหลัก

2. ระบบ Sign Up



รูปที่ 3.3 ระบบ Sign Up

จะเป็นส่วนสำคัญในการสมัครสมาชิกใหม่ ซึ่งจะมีขั้นตอนดังนี้

2.1 ระบบฟอร์มกรอกข้อมูล

ในกรณีที่ผู้ที่ยังไม่ได้เป็น สมาชิก จะต้องเข้ามาในส่วนนี้จะเป็นการกรอกข้อมูล เพื่อทำการสมัครเป็นสมาชิก ตัวอย่างเช่น username , password, name , address ฯลฯ และในส่วนนี้จะส่งข้อมูลในส่วนของ username and password ไปยัง (2.2) เพื่อตรวจสอบ

2.2 ระบบตรวจสอบข้อมูล

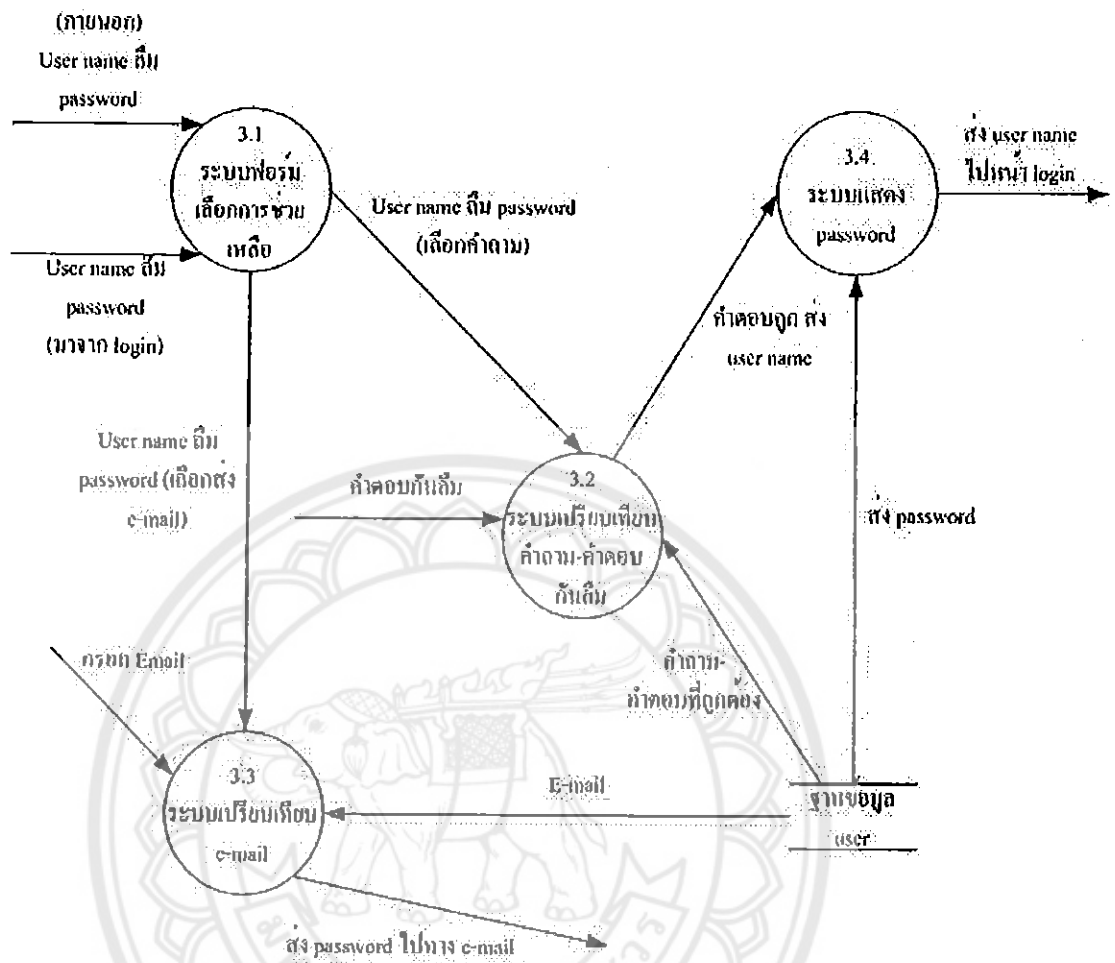
ในส่วนนี้จะรับข้อมูล username and password จาก (2.1) เพื่อนำค่าเหล่านั้นมาเปรียบเทียบกับมี username อยู่ในฐานข้อมูลของ user จะได้ 2 กรณีคือ

- ถ้ามีซ้ำกับในฐานข้อมูลก็จะส่ง กลับไป (2.1) เพื่อทำการตั้งค่าใหม่
- กรณีที่ไม่มี username นี้ในฐานข้อมูล ก็จะทำการสร้าง ฐานข้อมูลขึ้นมาใหม่ และนำค่า username and password ส่งต่อไปให้ระบบ (2.3) เพื่อทำการ login อัตโนมัติ

2.3 ระบบ login อัตโนมัติ

ในส่วนนี้ก็จะทำการรับค่า username and password จาก (2.2) และทำการ login อัตโนมัติ เพื่อผ่านเข้าไปในส่วนของการรับส่งจดหมาย (4)

3 ระบบช่วยเหลือเมื่อลืม password



รูปที่ 3.4 ระบบช่วยเหลือเมื่อลืม password

ในส่วนนี้จะทำหน้าที่ช่วยเหลือ สำหรับ ผู้ที่ลืม password ให้สามารถ login เข้าไปใช้งานได้ต่อไป จะมีขั้นตอน และหลักการดังนี้

3.1 ระบบฟอรัมเลือกการช่วยเหลือ

ในกรณีที่ user ลืม password จะเข้ามาในส่วนนี้และทำการเลือกการช่วยเหลือ โดยมี 2 กรณีคือ

- กรณีที่ต้องการตอบคำถามกันลืม ก็จะส่ง username ไปยังส่วนของ (3.2)
- กรณีที่ต้องการให้ส่ง password ไปยังอีเมลอื่น ก็จะส่ง username ไปยังส่วนของ (3.3)

3.2 ระบบเปรียบเทียบคำถาม-คำตอบกันลืม

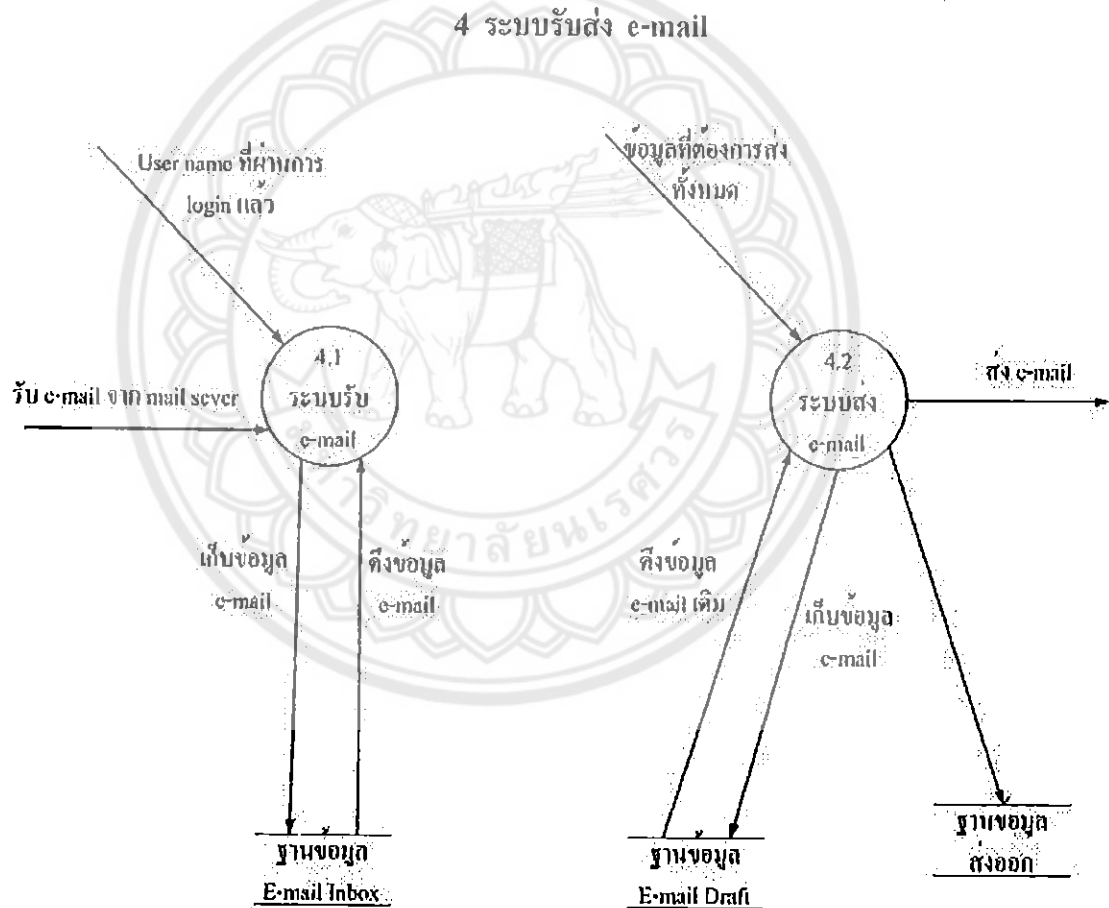
ในส่วนนี้จะรับค่า username และให้ user ตอบคำถาม และค่า user and answer นั้นไปตรวจสอบกับฐานข้อมูลของ user ว่าถูกต้องหรือไม่ ตอบจนว่าจะได้ และถ้าผ่านแล้วก็จะส่งต่อไปยังส่วนที่แสดง password (3.4)

3.3 ระบบเปรียบเทียบ e-mail

ในส่วนนี้จะรับค่า username และให้ user กรอก อีเมลอื่น ที่เคยกรอกไว้ตอนสมัคร และนำค่า user and other_mail มาเปรียบเทียบกับค่ากับ ฐานข้อมูลของ user ถ้าถูกต้องก็จะส่ง password ไปยัง e-mail ดังกล่าว ถ้าไม่ตรงตามฐานข้อมูลก็จะให้กรอกใหม่ว่าจะทำได้

3.4 ระบบแสดง password

ในกรณี user ตอบคำถามกันลืมผ่าน ระบบจะไปดึง password จากฐานข้อมูล user มาแสดง และส่งต่อไปยังหน้า login



รูปที่ 3.5 ระบบรับ-ส่ง e-mail

ในส่วนนี้จะเป็นส่วนของการรับส่งจดหมายทั้งหมด ไม่ว่าจะเป็นการแสดงจดหมาย การลบ การย้ายจดหมาย การเขียนจดหมายเพื่อส่ง ซึ่งแบ่งเป็น 2 ระบบย่อยคือ

4.1 ระบบรับ e-mail

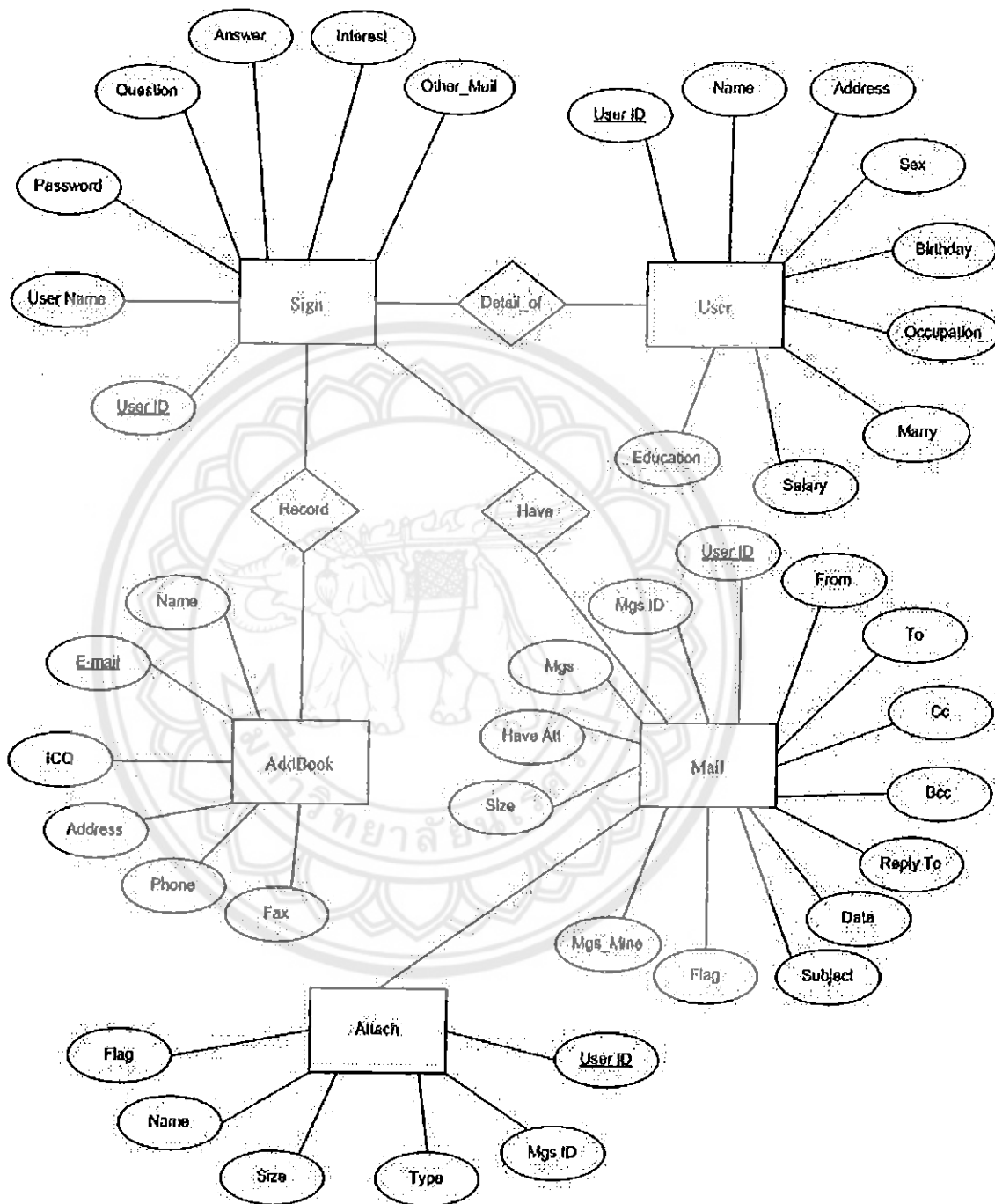
ในส่วนนี้จะทำการดึงอีเมล์จาก mail server เพื่อนำมาเก็บไว้ในฐานข้อมูลจดหมาย และเมื่อมีการ login ก็จะมีการเปรียบเทียบว่าเป็น username ใด และในส่วนนี้จะทำการจัดการเกี่ยวกับจดหมายทั้งหมดคือ ไม่ว่าจะเป็นการ แสดงจดหมายในส่วนต่างๆ การลบ การย้ายจดหมาย

4.2 ระบบส่ง e-mail

ในส่วนนี้จะเป็นการส่งจดหมาย ไปยัง e-mail ที่ user ต้องการ ในส่วนนี้สามารถเก็บจดหมายที่ยังเขียนไม่เสร็จ และดึงจดหมายเหล่านั้น (ซึ่งจะบันทึกไว้ในฐานข้อมูลจดหมาย) ขึ้นมาเขียนต่อได้ รวมถึงเมื่อมีการส่งไปแล้วก็จะมีการบันทึกไว้เช่นกัน ว่าเราส่งอะไรไปให้ใครบ้าง



3.2 ระบบฐานข้อมูลโดยใช้ ER-Model



รูปที่ 3.6 ระบบฐานข้อมูลโดยใช้ E-R Model

ในฐานข้อมูลอีเมลของเราจะประกอบด้วยฐานข้อมูล 5 Table คือ

1. **Sign** จะเป็นฐานข้อมูลที่เกี่ยวข้องกับข้อมูลการสมัคร อาทิเช่น User_id , username password เป็นต้น
2. **User** จะเป็นฐานข้อมูลที่เกี่ยวข้องกับข้อมูลส่วนตัวของผู้สมัคร อาทิเช่น ชื่อ นามสกุล ที่อยู่ อาชีพ เป็นต้น
3. **Mail** จะเป็นฐานข้อมูลที่เกี่ยวข้องกับรายละเอียดของจดหมาย อาทิเช่น เนื้อความจดหมาย วันที่ส่ง ส่งมาจากใคร เป็นต้น
4. **Attach** จะเป็นฐานข้อมูลที่เกี่ยวข้องกับไฟล์ที่แนบมากับจดหมาย อาทิเช่น ชื่อไฟล์ที่แนบ ขนาดประเภทของไฟล์ เป็นต้น
5. **AddBook** จะเป็นฐานข้อมูลที่เกี่ยวข้องกับสมุดรายชื่อ อาทิเช่น ชื่อ อีเมล เป็นต้น

ตารางที่ 3.1 ฐานข้อมูล Sign Up

User_id	เก็บหมายเลข user เพื่อนำไปเป็น id อ้างอิงใช้ร่วมกับ User, Mail, Attach
Username	เก็บชื่อ username ซึ่งชื่อนี้จะเป็นชื่อสำหรับ Login และจะต้องตรงกับ user ที่มีอยู่ในเมลเซิร์ฟเวอร์ด้วย
Password	เก็บรหัสผ่านสำหรับ Login และจะต้องตรงกับ user ที่มีอยู่ในเมลเซิร์ฟเวอร์ด้วย
Signature	เก็บข้อความลงท้ายจดหมาย
Question	เก็บคำถามกันลืม ในกรณีที่ลืม password
Answer	เก็บคำตอบกันลืม
Interest	เก็บความสนใจของผู้สมัคร
Other_mail	เก็บอีเมลอื่น ใช้สำหรับในการติดต่อ เพื่อส่งข่าวสาร

ตารางที่ 3.2 ฐานข้อมูล User

User_id	เก็บหมายเลข user เพื่อนำไปเป็น id อ้างอิงใช้ร่วมกับ Sign, Mail, Attach
Name	เก็บชื่อและนามสกุลของผู้สมัคร
Address	เก็บที่อยู่ที่สามารถติดต่อได้
Sex	เก็บเพศของผู้สมัคร
Birthday	เก็บวันเดือนปีเกิด เพื่อจะได้ทราบอายุปัจจุบัน
Occupation	เก็บอาชีพของผู้สมัคร
Marry	เก็บสถานภาพการแต่งงาน
Salary	เก็บเงินเดือน
Education	เก็บการศึกษาขั้นสูงสุด

ตารางที่ 3.3 ฐานข้อมูล e-mail

ID_msg	เก็บหมายเลข id สำหรับอีเมลแต่ละฉบับ เพื่อใช้อ้างอิงในการจัดการต่างๆ เช่น ลบ, Reply ฯลฯ
ID_user	เก็บหมายเลข user เพื่อเป็นการบอกว่าอีเมลนี้เป็นของ user คนไหน
From	เก็บข้อมูล header ว่าอีเมลฉบับนี้ใครส่งมา (From:)
To	เก็บข้อมูล header ว่าอีเมลฉบับนี้ส่งถึงใครบ้าง (To:)
CC	เก็บ ว่าอีเมลฉบับนี้สำเนา ถึงใครบ้าง
BCC	เก็บ ว่าอีเมลฉบับนี้สำเนาซ่อน ถึงใครบ้าง
Cc	เก็บข้อมูล header ว่าอีเมลฉบับนี้สำเนา CC ถึงใคร
Reply_to	เก็บข้อมูล header ว่าอีเมลฉบับนี้ตอบกลับให้ใคร
Date	เก็บข้อมูลวันเวลาที่รับส่งอีเมล
Subject	เก็บหัวข้อเรื่อง
Size	เก็บขนาดของอีเมล (KB)
Message	เก็บข้อความจดหมาย (message body เท่านั้น ไม่เกี่ยวกับ

ตารางที่ 3.3 ฐานข้อมูล e-mail (ต่อ)

	ไฟล์ attachment)
Msg_mine	เก็บชนิดของ message body เช่น เป็นข้อความธรรมดา หรือเป็นแบบ HTML
Have_att	เก็บสถานะว่าอีเมลนี้ Attachment หรือไม่ เพื่อตอนลิส Inbox จะได้แสดงสัญลักษณ์ เพื่อแสดงให้เราทราบว่าอีเมลีได้มีการ Attachment คิดมาด้วย
Flag	เก็บสถานะของอีเมลฉบับนั้นๆ ว่า <ul style="list-style-type: none"> 0 มาใหม่ซึ่งไม่ได้เปิดอ่าน (อยู่ใน Inbox) 1 เปิดอ่านแล้ว (ยังอยู่ใน Inbox) 2 ถูกลบไปแล้วและอยู่ในถังขยะ(Trash can) 3 เมล์ส่งออกไปแล้วและถูกเก็บไว้ในตู้ Sent Box

ตารางที่ 3.4 ฐานข้อมูลการ Attach File

Id_msg	เก็บหมายเลข id สำหรับอีเมลแต่ละฉบับ เพื่อใช้อ้างอิงในการจัดการต่างๆ เช่น ลบ, Reply ฯลฯ
Id_user	เก็บหมายเลข user เพื่อเป็นการบอกว่าอีเมลนี้เป็นของ user คนไหน
Type	เก็บชนิดข้อมูลของไฟล์ Attachment ว่าเป็นอะไร เช่น image/gif, application ฯลฯ
Size	เก็บขนาดของ Attachment
Name	ชื่อไฟล์ Attachment จำเป็นต้องใช้ตอนที่ผู้เปิดอีเมล download ชื่อนี้จะปรากฏที่ dialog box Save as filename
Flag	เก็บสถานะของ Attachment (คล้ายกับ tbl_mail)

ตารางที่ 3.5 ฐานข้อมูล Address Book

Name	เก็บชื่อที่ต้องการบันทึก
E-mail	เก็บ e-mail address
ICQ	เก็บ ICQ
Address	เก็บที่อยู่
Phone	เก็บเบอร์โทรศัพท์
FAX	เก็บเบอร์ FAX

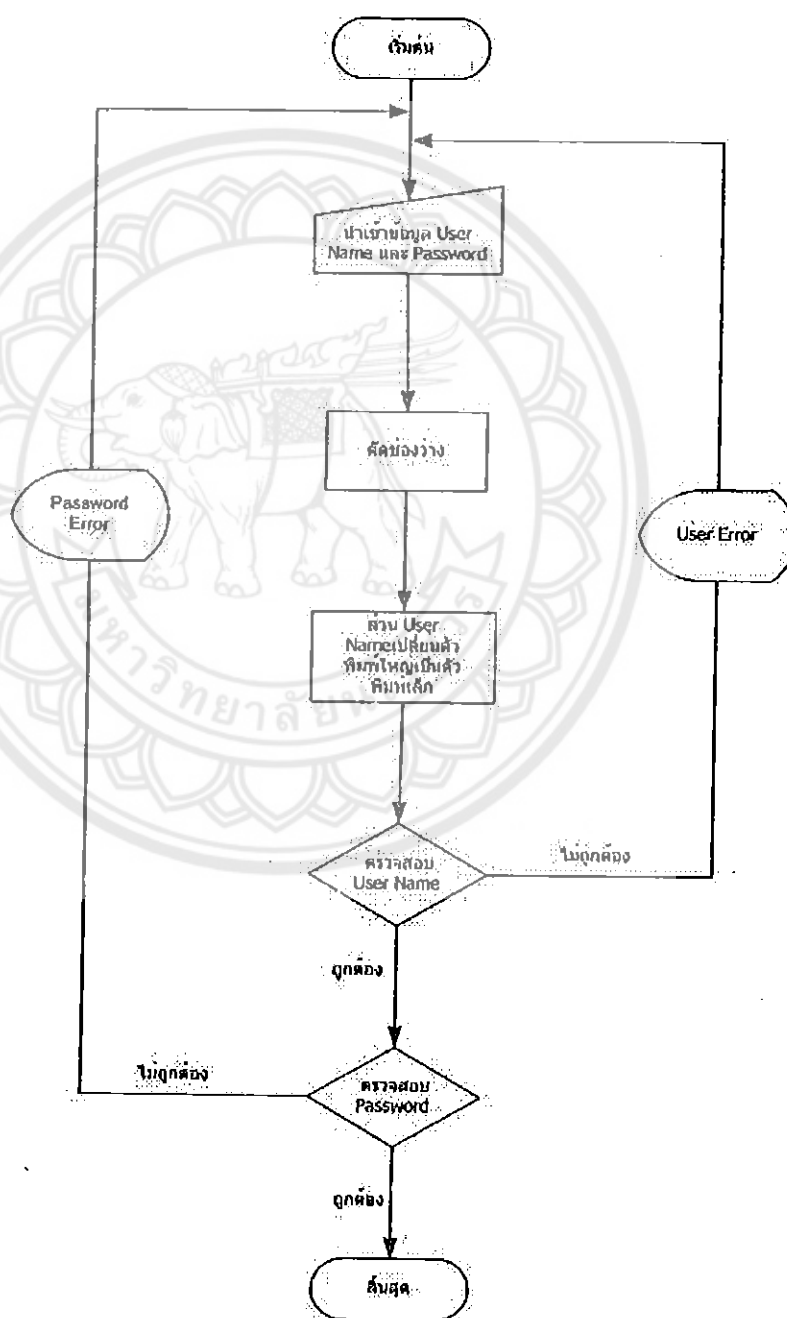


บทที่ 4

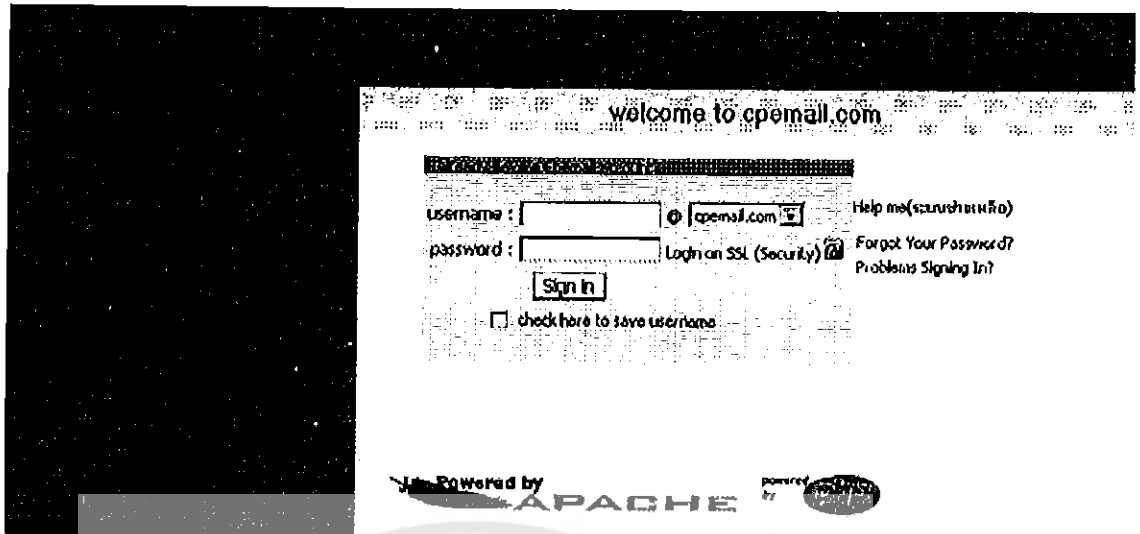
การพัฒนาโปรแกรมและผลการทดลอง

โปรแกรมรับ-ส่งจดหมายอิเล็กทรอนิกส์ ใช้ภาษา PHP ในการพัฒนาโปรแกรม และใช้ MySQL เป็นตัวจัดการฐานข้อมูล โดยทำงานบน Server ที่เป็นระบบปฏิบัติการ Linux ซึ่งมีแนวคิดดังนี้

4.1 ระบบ Login

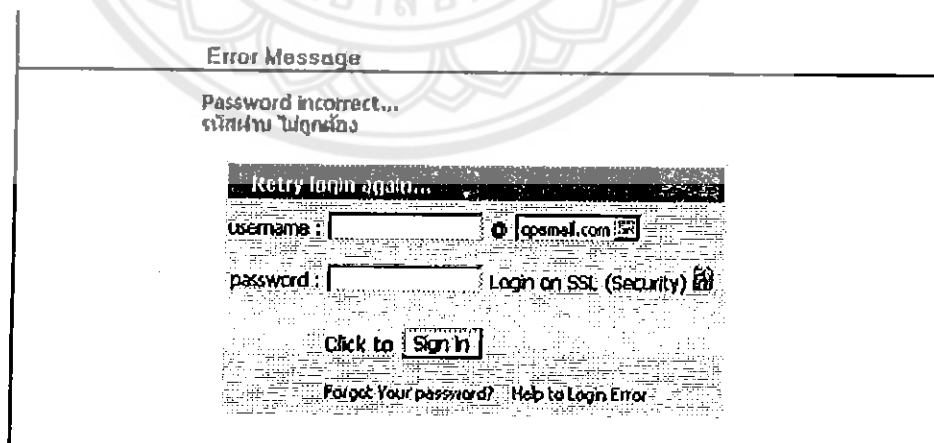


รูปที่ 4.1 ฟังงานของระบบ Login



รูปที่ 4.2 ลักษณะของโปรแกรมในส่วนของการ login เข้าสู่ระบบ

เมื่อผู้ใช้ต้องการเข้าสู่ระบบรับ-ส่งจดหมายของตนเองนั้น จำเป็นต้องทำการ login ก่อนทุกครั้งเพื่อความปลอดภัยของจดหมายและข้อมูลอื่นๆ โดยให้ผู้ใช้ทำการกรอกค่า username และ password ลงในช่องที่กำหนดให้และทำการคลิกที่ปุ่ม Sign in จากนั้นโปรแกรมจะทำการส่งค่า username และ password ไปตรวจสอบในระบบ ถ้าถูกต้อง โปรแกรมจะทำการเข้าสู่ระบบ รับ-ส่งจดหมายของแต่ละผู้ใช้งานที่ แต่ถ้าเกิดข้อผิดพลาด โปรแกรมจะกลับไปยังหน้า login error และให้ผู้ใช้ทำการกรอกค่าต่างๆ ใหม่อีกครั้ง ดังรูปที่ 4.3 รูปที่ 4.4 รูปที่ 4.5



รูปที่ 4.3 ลักษณะของโปรแกรมในส่วนของการ login error

Error Message

ไม่มีการกรอกค่า username...

Retry login again...

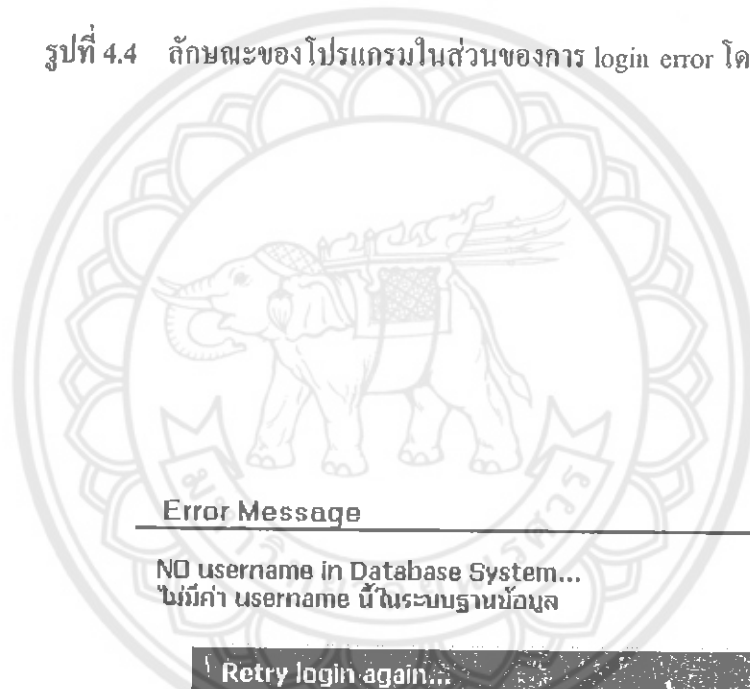
username : @ cpemail.com

password : Login on SSL (Security)

Click to

[Forgot Your password?](#) [Help to Login Error](#)

รูปที่ 4.4 ลักษณะของโปรแกรมในส่วนของการ login error โดยไม่มีการกรอกข้อมูล



Error Message

NO username in Database System...
ไม่มีค่า username นี้ในระบบฐานข้อมูล

Retry login again...

username : @ cpemail.com

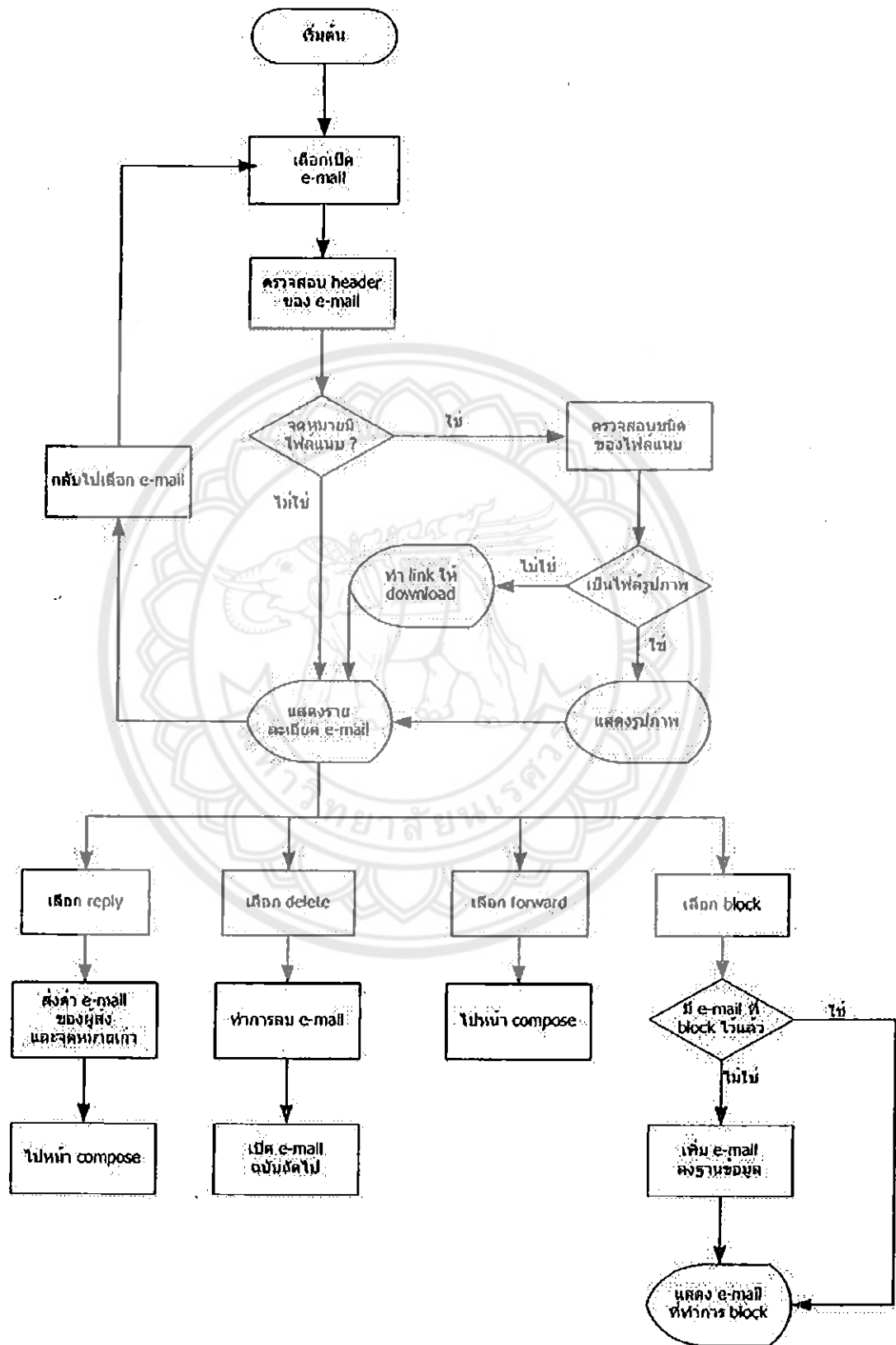
password : Login on SSL (Security)

Click to

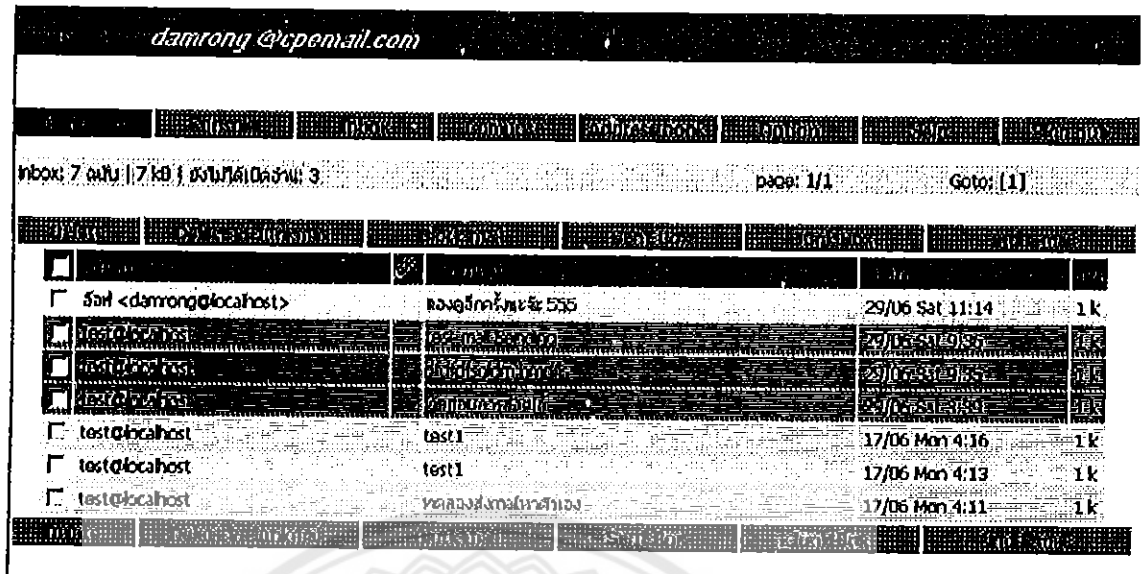
[Forgot Your password?](#) [Help to Login Error](#)

รูปที่ 4.5 ลักษณะของโปรแกรมในส่วนของการ login error โดยมีการกรอกข้อมูล user ที่ไม่มีจริง

4.2 ระบบรับ e-mail

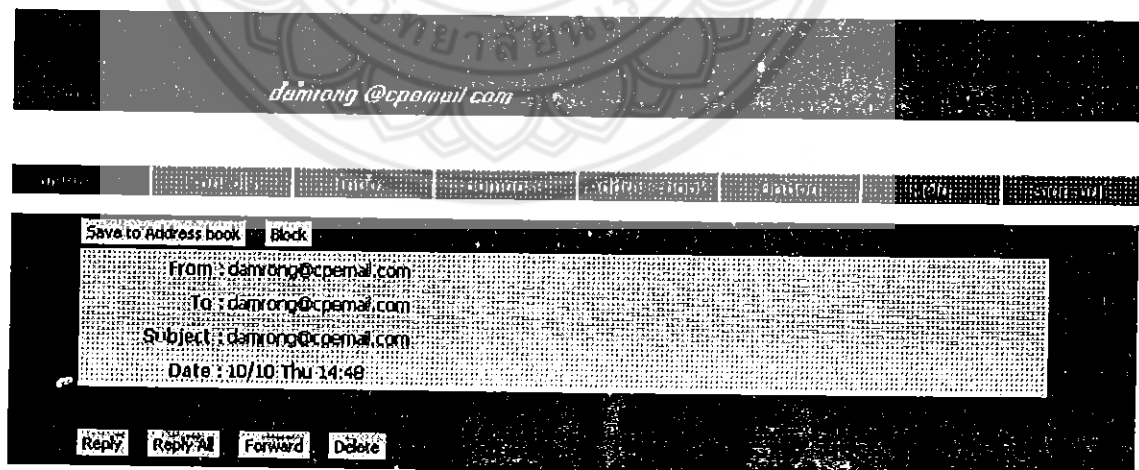


รูปที่ 4.6 ฟังก์ชันของระบบการรับ e-mail



รูปที่ 4.7 ลักษณะของโปรแกรมในส่วนของ Inbox

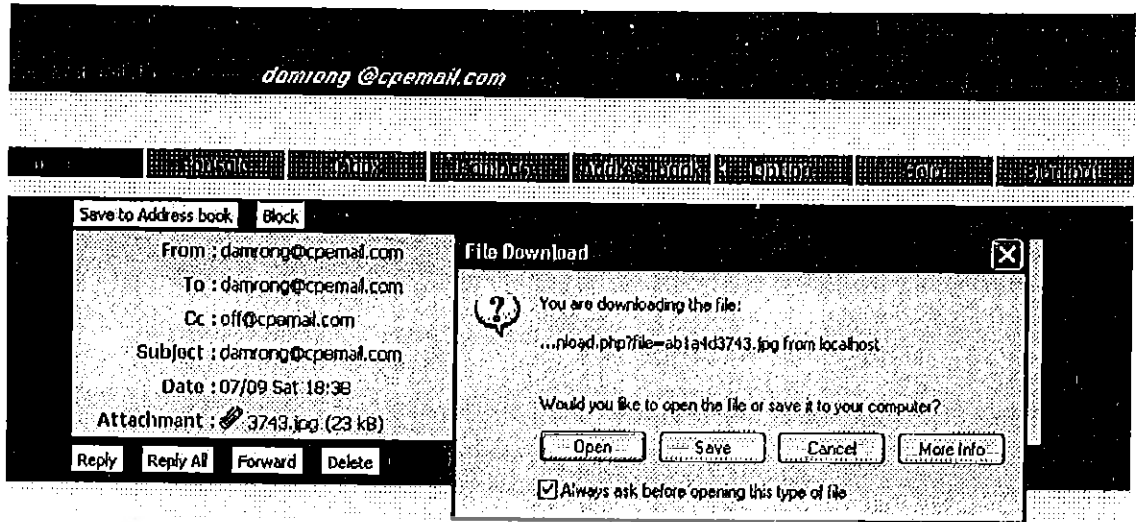
ในกล่องจดหมาย ระบบจะทำการแสดงข้อมูลของจดหมายบางส่วน โดยให้ผู้ใช้ทำการเลือกอ่านจดหมายโดยให้ทำการคลิกที่ Subject ของจดหมายฉบับนั้น และเมื่อต้องการจัดการกับจดหมายเช่น delete ,block, ฯลฯ ก็สามารถทำได้ โดยทำการคลิกที่ ช่อง Select ของจดหมายนั้น จากนั้นจึงทำการเลือกการกระทำต่างๆ โดยคลิกที่ link ต่างๆ



ทดสอบส่งจดหมาย

การทดสอบที่เชิงใหญ่

รูปที่ 4.8 ลักษณะของโปรแกรมที่ทำการเปิดอ่าน e-mail



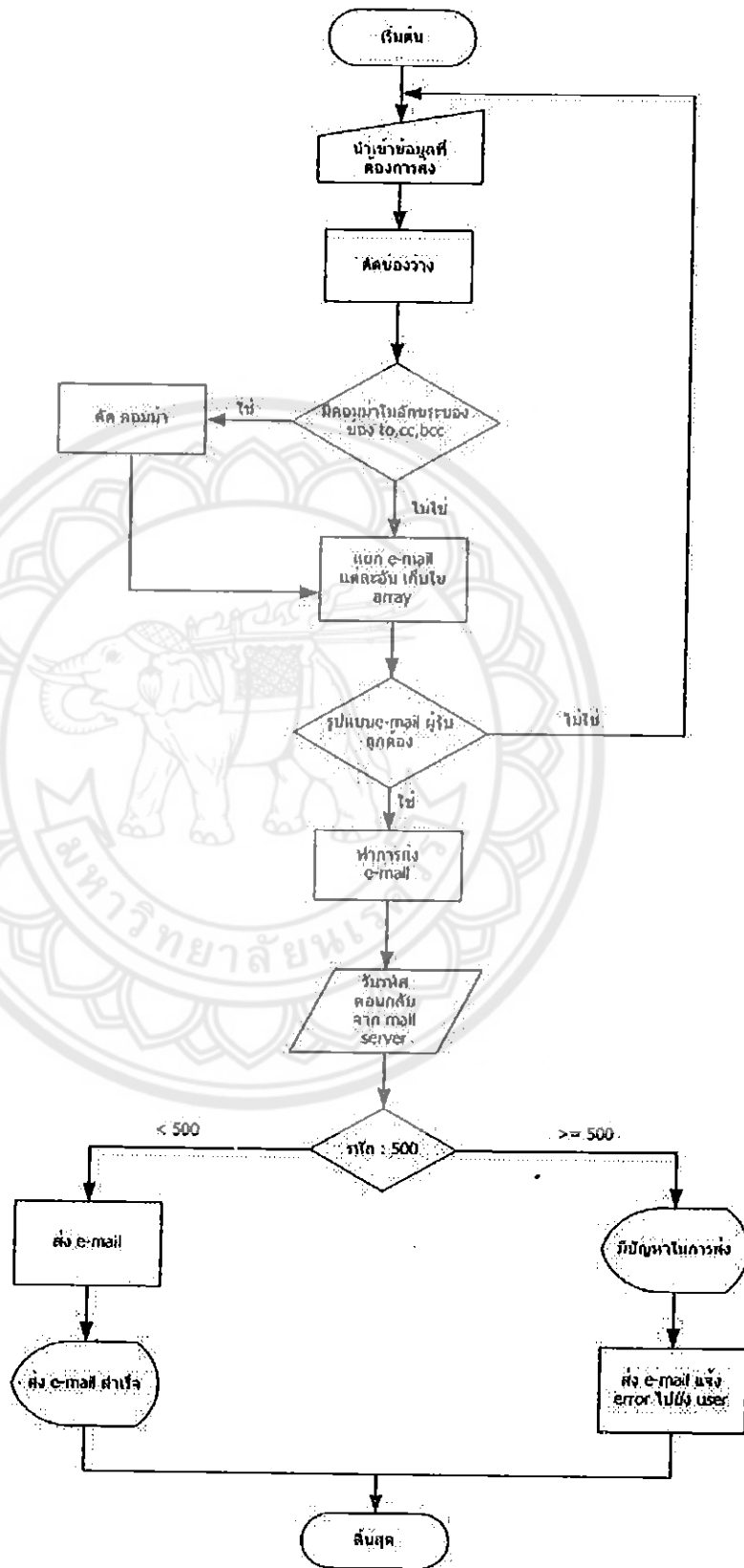
IIIIII

XXXXXXXXXX

รูปที่ 4.9 ลักษณะของ โปรแกรมเปิดอ่าน e-mail ที่มี ไฟล์ attach file และจะทำการ download



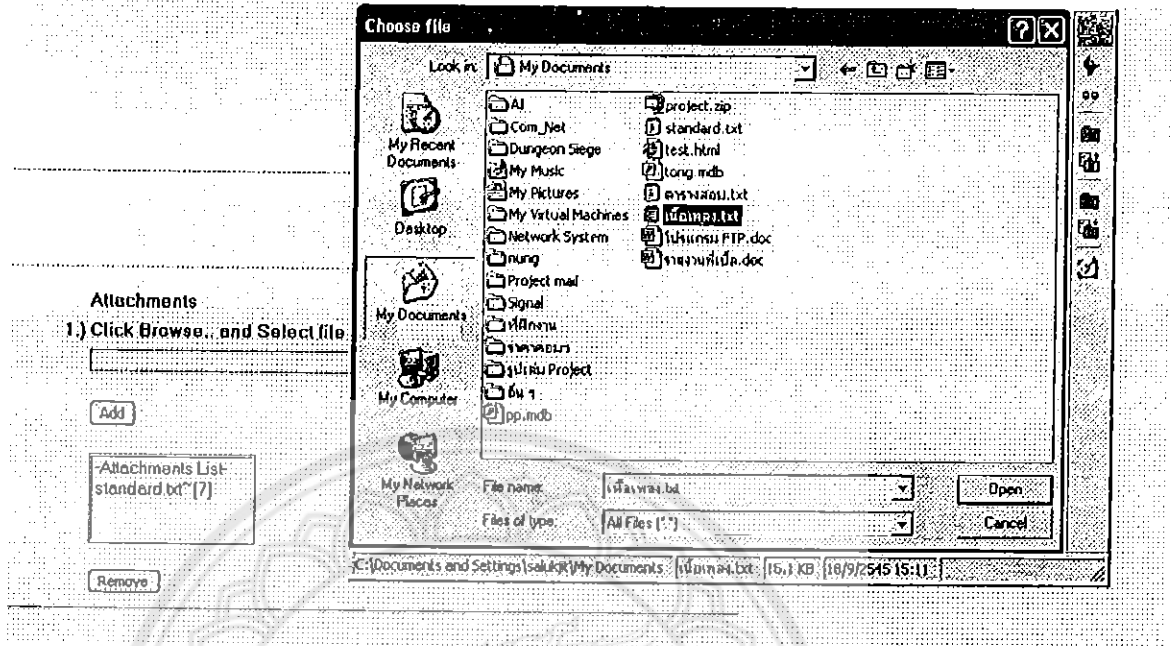
4.3 ระบบการส่ง e-mail



รูปที่ 4.10 ฟังก์ชันของระบบการส่ง e-mail

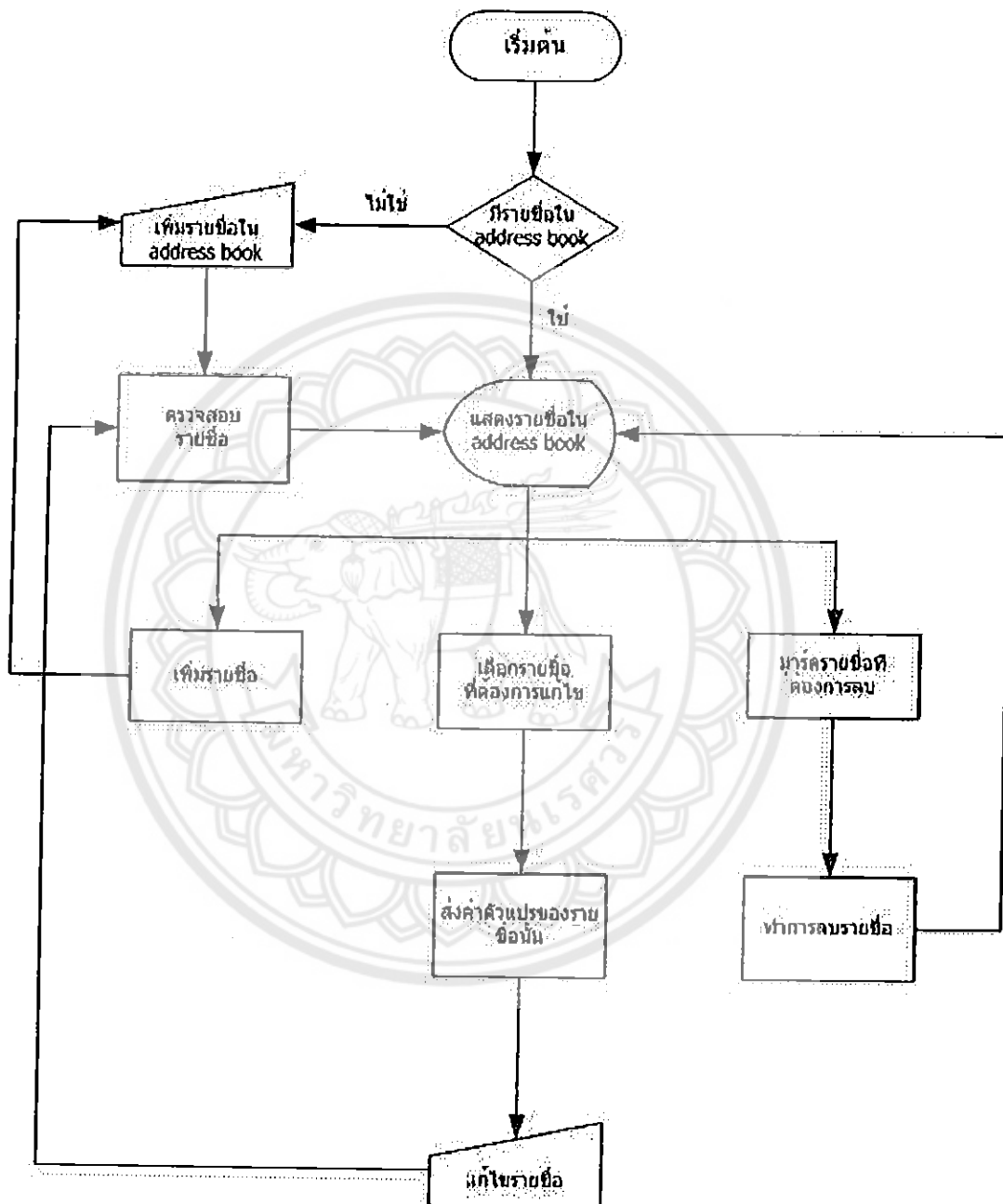
รูปที่ 4.11 ลักษณะของโปรแกรมในส่วนของการส่ง e-mail

Form การส่ง e-mail ในโปรแกรม จะให้ผู้ส่งทำการกรอกข้อมูลในการส่ง เช่น เบอร์ e-mail ของผู้รับ ในช่อง To: Cc: หรือ Bcc: (แล้วแต่จุดประสงค์ของผู้ใช้งาน) , Subject และ ข้อมูลจดหมาย โดยผู้สามารถ เลือกที่จะส่งแบบ text หรือ html ก็ได้ โดยเลือกที่ Type หรือถ้ายังเขียนไม่เสร็จสามารถ Save Draft ไว้ก่อนได้อีกด้วย และในส่วนของการแนบไฟล์ มีขั้นตอนคือให้ผู้ใช้ทำการ Browse ทำการเลือกไฟล์ แล้วทำการกดปุ่ม Add file จากนั้นระบบจะทำการตรวจสอบขนาดของไฟล์ ถ้าถูกต้อง ชื่อของไฟล์จะถูกเพิ่มใน ฟیلด์ Select หรือถ้าต้องการ ลบไฟล์ที่ Add เข้ามาแล้ว ก็ให้ทำการคลิกเลือกที่ ชื่อไฟล์ แล้วจึงทำการคลิกที่ปุ่ม Remove file

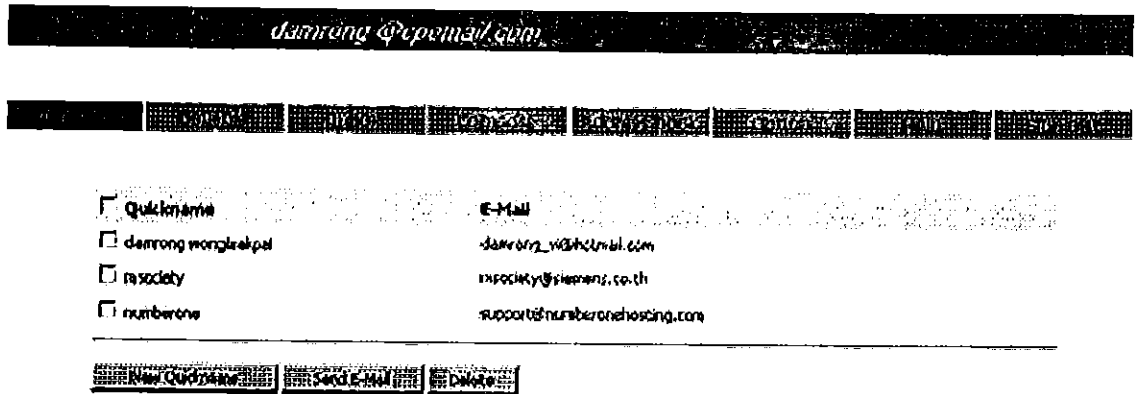


รูปที่ 4.12 ลักษณะของโปรแกรมในส่วนของการส่ง e-mail แบบ attach file

4.4 ระบบ Address book



รูปที่ 4.13 ฝั่งงานของระบบAddress Book



รูปที่ 4.14 ลักษณะของโปรแกรมในส่วนของสมุดรายชื่อ

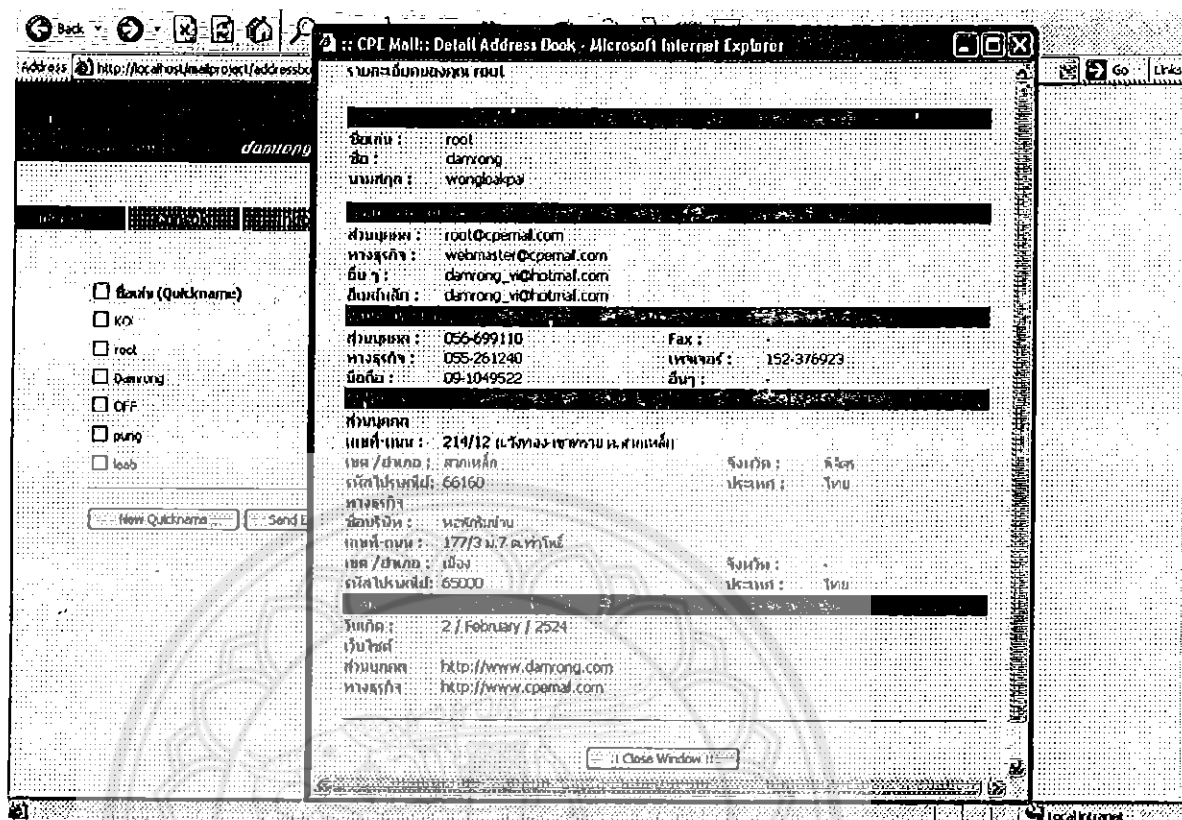
ถ้าผู้ใช้ต้องการบันทึกข้อมูลของผู้รับจดหมาย เช่น ชื่อผู้รับ เบอร์ e-mail ที่อยู่ ฯลฯ ก็สามารถทำได้ โดยถ้าผู้ใช้ต้องการเพิ่มรายชื่อผู้รับ ก็ให้ทำการ คลิกที่ ปุ่ม New Quickname หรือถ้าต้องการลบรายชื่อใด ๆ ก็ให้คลิก ที่ ช่อง Select ของรายชื่อ นั้น แล้วจึงทำการ คลิกที่ปุ่ม Delete

The screenshot shows the 'Create New Individual' form in an email client. The address bar displays 'damrong@cpemail.com'. The menu bar includes: 'File', 'Edit', 'View', 'Tools', 'Options', 'Help'. The form has the following fields and options:

- ชื่อเล่น (กรอกทศบุ):** [Text input field]
- ชื่อ:** [Text input field]
- นามสกุล:** [Text input field]
- อีเมล ส่วนบุคคล:** [Text input field]
- อีเมล ทางธุรกิจ:** [Text input field]
- อีเมล อื่น ๆ:** [Text input field]
- เลือก อีเมลส่วนตัว:** อีเมล ส่วนบุคคล

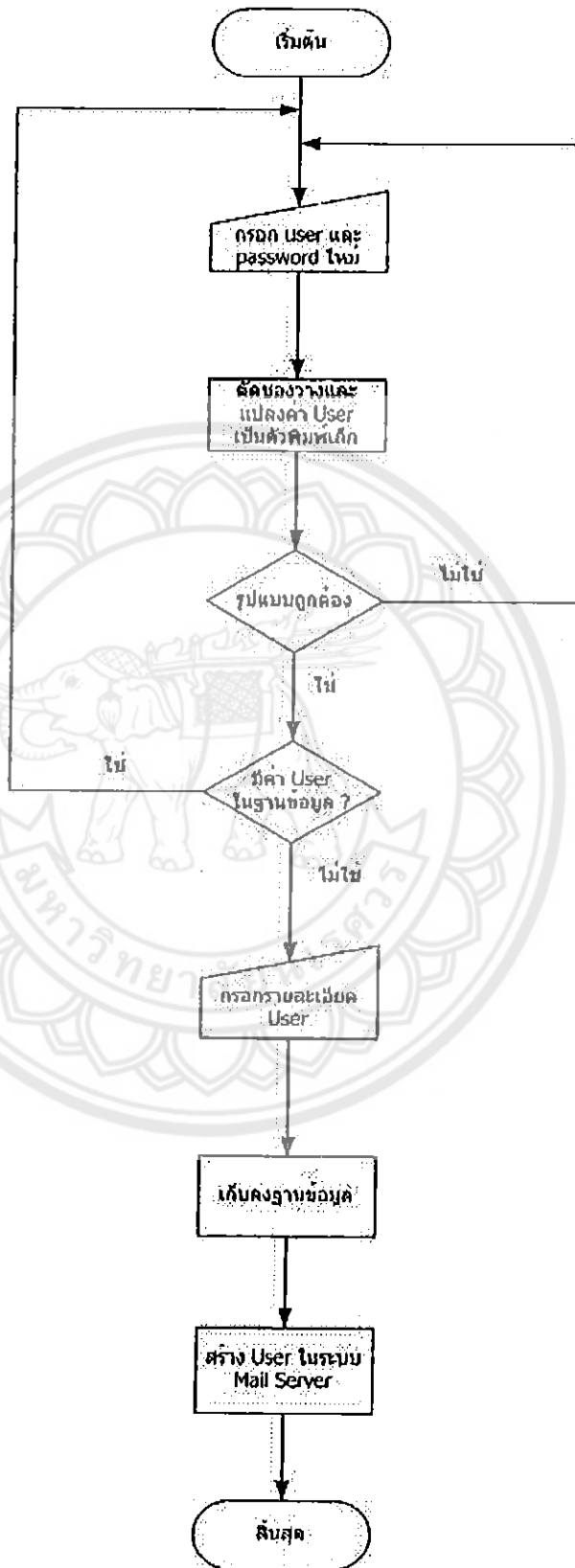
On the right side of the form, there is a note: 'ค่า "ชื่อเล่น" จะใช้ในการอ้างอิงค่าต่างๆ ในระบบ webbase mail นี้ทั้งหมด' and 'อีเมลส่วนตัว เป็น อีเมลที่เลือกให้ "ชื่อเล่น" ใช้ในการอ้างอิง'.

รูปที่ 4.15 ลักษณะโปรแกรมของการเพิ่มรายชื่อในสมุดรายชื่อ



รูปที่ 4.16 ลักษณะโปรแกรมของรายชื่อแต่ละคนในสมุดรายชื่อ

4.5 ระบบ Create User



รูปที่ 4.17 ฟังก์ชันของระบบ Create User

Untitled Document - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Home Refresh

Address http://localhost/project/TMP6yynn3siz.php

ชื่อ

นามสกุล

ที่อยู่

จังหวัด

รหัสไปรษณีย์

เพศ ชาย หญิง ไม่ระบุ

วันเกิด

อาชีพ

ข้อมูลการสมัครสมาชิก

Sign-In Name @domain.com

Password

Re-enter Password

ฉันขอมติ

คำทอ

Done Local intranet

รูปที่ 4.18 ลักษณะของโปรแกรมในส่วนของการ Create User

เมื่อผู้ใ้ต้องการสร้างเบอร์อีเมลขึ้นนั้น ผู้ใ้ต้องทำการสมัครสมาชิกในระบบโปรแกรมก่อน โดยใ้ทำการกรอกรายละเอียดต่างๆ ใน ฟอรั่ม จากนั้นทำการกรอกค่า username ใหม่ขึ้น โดยระบบจะทำการตรวจสอบค่า username ว่ามีอยู่ในฐานข้อมูลหรือยัง ถ้ามีอยู่แล้ว ระบบจะใ้ทำการกรอกค่า username ใหม่อีกครั้ง เมื่อค่า username ยังไม่มีในฐานข้อมูล ระบบก็จะทำการสร้าง user ขึ้น แล้วจึงสามารถใช้งานในระบบได้

บทที่ 5

บทสรุป

5.1 สรุปผล

1. การ login และ การ logout สามารถทำงานได้ดีในระดับหนึ่ง โดยใช้มาตรฐานความปลอดภัย SSL (Security Socket Layer)
2. การสมัครสมาชิก (Create User) สามารถกรอกข้อมูล และทำการบันทึกลงฐานข้อมูลได้อย่างสมบูรณ์ แต่ยังไม่สามารถ สร้าง User ในเครื่อง Server ได้ ดังนั้นต้องไปศึกษาเพิ่มเติมในส่วนของการติดต่อ shell ของ Linux โดยใช้ภาษา PHP
3. การส่งจดหมายและการแนบไฟล์ไปกับจดหมาย สามารถส่งอีเมลล์ในแบบของ Text หรือ HTML รวมถึงการแนบไฟล์ได้หลายรูปแบบ (ไฟล์ข้อความ, ไฟล์รูปภาพ, ไฟล์บีบอัด) ไฟล์ที่แนบมีขนาดได้ถึง 1 MB สามารถเช็ครูปแบบของ E-mail Address ว่าถูกต้องและมีจริงหรือไม่ รวมถึงรายงานสถานะการส่ง
4. การรับอีเมลล์ สามารถถอดรหัสของอีเมลล์ที่ส่งมาได้หลายรูปแบบเช่น การรับอีเมลล์จากโปรแกรม Outlook และการรับอีเมลล์จาก Web base e-mail รวมถึงการตรวจสอบไฟล์ที่แนบมากับอีเมลล์ ถ้าเป็นไฟล์รูปภาพจะแสดงผล ถ้าเป็นไฟล์อื่นจะมี link ให้ทำการ Download
5. สมุดรายชื่อ(Address Book) สามารถใช้บันทึกข้อมูล เช่น อีเมลล์ ชื่อ ที่อยู่ ฯลฯ สามารถใช้ร่วมกับการส่งอีเมลล์

5.2 ปัญหา และ แนวทางแก้ไข

1. การจัดการการทำงานของโปรแกรมในส่วนของการ Login แบบ SSL ทางเราไม่สามารถการ จัดทำที่ถูกต้อง ควรศึกษาต้องให้ดี ก่อนนำไปใช้จริง
2. การ create USER บนระบบปฏิบัติการ Linux ผ่านทาง web page ยังไม่สามารถทำได้ ควรศึกษาเพิ่มเติมภาษา PHP ในการติดต่อกับ Shell บน Linux
3. การส่ง e-mail แบบ HTML ยังไม่มี เครื่องมือ(Tool)ในการช่วยเขียนแท็กต่างๆ ของภาษา HTML ซึ่งผู้ใช้ต้องทำการเขียน แท็กต่างๆ เอง
4. มีข้อจำกัดในขนาดของไฟล์ที่แนบไปกับ e-mail โดยสามารถแนบไฟล์ได้ขนาดที่จำกัด โดยขึ้นอยู่กับ ระบบ Mail Server

5.3 แนวทางการพัฒนาเพิ่มเติม

1. ส่วนของ Create User ต้องศึกษาเพิ่มเติมในส่วนของการ Create User ให้กับโปรแกรม Mail Server โดยสามารถจัดการผ่านทาง หน้าเว็บเพจได้ทันที ผู้เขียนแนะนำให้ศึกษาการเพิ่ม User ให้กับโปรแกรม Mail Server ก่อนที่จะเพิ่มได้อย่างไร เช่น การเพิ่ม User โดยใช้คำสั่ง `useradd` ผ่านทาง Shell จากนั้นจึงเริ่มเขียน โปรแกรม ติดต่อในส่วนของ Shell โดยใช้ฟังก์ชัน `exec` ฯลฯ ในภาษา PHP
2. ส่วนของ การรับ Mail ต้องพัฒนาเพิ่มเติมในส่วนของ ก่อตั้งจดหมาย เช่น Sand box ,Junk Box และในส่วนของ Block Mail ที่เราไม่ต้องการ โดย ในส่วนต่างๆ ของรับ Mail นี้ จะเกี่ยวข้องกับ การใช้ภาษา PHP ติดต่อในส่วนของฐานข้อมูล MySQL
3. ส่วนของ การส่ง Mail ต้องเพิ่มเติมในส่วนของการใช้ Tools ช่วยในการเขียน แท็ก HTML ผู้ที่สนใจอาจจะพัฒนา Tools ขึ้นมาใช้เอง โดยใช้ภาษา Java หรือลองค้นหา Tools ใน Internet



เอกสารอ้างอิง

- [1] กอบเกียรติ สระอุบล. สร้างเว็บเพจด้วย PHP ฉบับประยุกต์ เล่ม 1.พิมพ์ครั้งที่ 1 กรุงเทพฯ :
บี อี แอนด์ ซี.
- [2] กอบเกียรติ สระอุบล. สร้างเว็บเพจด้วย PHP ฉบับประยุกต์ เล่ม 2.พิมพ์ครั้งที่ 1 กรุงเทพฯ :
บี อี แอนด์ ซี.2545
- [3] กิตติศักดิ์ เจริญโกกานนท์. คัมภีร์การสร้าง Application PHP4. กรุงเทพฯ.
ซัคเซส มีเดีย จำกัด,บริษัท.2543
- [4] นิรุช อำนวยศิลป์. การสร้างเว็บเพจอย่างไรจึงดีจำกัด PHP เพื่อการประยุกต์ใช้งาน. กรุงเทพฯ :
ซัคเซส มีเดีย จำกัด,บริษัท.
- [5] สุวัฒน์ ปุณณชัยยะ,ต้น ต้นจักษ์สุทธิวงศ์ และ สุพจน์ ปุณณชัยยะ . เปิดโลกของ TCP/IP และ
โปรโตคอลของอินเทอร์เน็ต.พิมพ์ครั้งที่ 1 .กรุงเทพฯ. โปรวิชั่น จำกัด,บริษัท.2543



ภาคผนวก ก

ศึกษาการเซตอัฟ Linux Redhat 7.3 ให้ทำงานเป็นเครื่อง Server

1. DNS Server

Name Server เป็น Server สำหรับแม่พ็ช็อบน Domain Name System ให้เป็น IP Address กระทบการแม่พ็ช็อดังกล่าวอาจจะเกิดขึ้นในช่วงใดก็ได้ที่มีโปรเซสงานสำหรับการแม่พ็ช็อดังกล่าวรันอยู่ เช่น บน Server ภายใน LocalNetwork ของคุณ หรือบน Server ของผู้ให้บริการอินเทอร์เน็ต (ISP)

การติดตั้ง DNS Server

ทำการตรวจสอบดูว่าได้ทำการ install packet ดังต่อไปนี้หรือไม่

- bind-9.x.x.x.i386.rpm
- caching-nameserver-7.x.x.noarch.rpm

โดยใช้คำสั่ง # rpm -q bind-* ในการตรวจสอบการ install packet ถ้าได้ทำการ install แล้วจะมีชื่อของ packet นั้นโชว์ขึ้นมา แต่ถ้ายังไม่ได้ทำการ install จะโชว์คำว่า packet is not install ดังนั้นจึงต้องทำการลง packet นั้นก่อน โดย packet ต่างๆ นั้นจะอยู่แผ่น CD-ROM Redhat ใน /RedHat/RPMS ทำการ install โดยใช้คำสั่ง

```
# rpm -i bind-9.x.x.x.i386.rpm
# rpm -i caching-nameserver-7.x.x.noarch.rpm
```

จากนั้นทำการแก้ไข ไฟล์ config ต่างๆ ดังนี้

1. ตรวจสอบไฟล์ config ชื่อ resolv.conf โดยไฟล์นี้จะอยู่ใน /etc/

/etc/resolv.conf

```
Domain server.com
```

```
Nameserver
```

```
192.168.10.5
```

```
192.168.10.7
```

ไฟล์ resolv.conf นี้ทำการบันทึกไคเรคทีฟ domain โดยจะกำหนดชื่อโดเมน และไคเรคทีฟ nameserver กำหนดค่า IP Address ของ NameServer โดย resolver จะขอบริการจาก Server : 192.168.10.5 เป็นอันดับแรก หากไม่สามารถให้บริการได้ resolver จะพยายามขอบริการจาก Server : 192.168.10.7 เป็นอันดับถัดไป

ค้ำน NameServer จะมี แดมอน named ทำหน้าที่เป็น Server รอรับการร้องขอจาก resolver เมื่อ named เริ่มทำงานจะอ่านข้อมูลจากแฟ้ม /etc/named.boot หรือ /etc/named.conf

2.ทำการแก้ไขไฟล์ named.conf โดยทำการเพิ่มค่า Config ของ domain name ที่ต้องการเพิ่ม โดยค่า Config ของ 1 โดเมนนั้นแบ่งออกเป็น 2 ส่วน คือส่วน ของการแปลง ชื่อ domain name ให้เป็น IP Address และอีกส่วนหนึ่งคือ การ IP Address เป็นชื่อ domain name โดยทำการแก้ไข ดังนี้

/etc/named.conf

```
// generated by named-bootconf.pl
options {
    directory "/var/named"; //ทำการกำหนดค่า directoy ที่ใช้เก็บ ไฟล์ฐานข้อมูลHost ที่กำหนด
    ขึ้นในการแปลง
    /*
        //IP address เป็น ชื่อ domain name และ ชื่อ domain name เป็น IP
    * If there is a firewall between you and nameservers you want
    * to talk to, you might need to uncomment the query-source
    * directive below. Previous versions of BIND always asked
    * questions using port 53, but BIND 8.1 uses an unprivileged
    * port by default.
    */
    // query-source address * port 53;
};
// a caching only nameserver config
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 "server.com" IN {                                //ทำการเพิ่มชื่อ domain name ชื่อว่า server.com
    type master;
    file "server.name";                               //โดยเป็นการแปลงชื่อ domain name ให้เป็น IP Address ใน
    allow-update { none; };                           // ไฟล์ที่ชื่อ server.name
};

zone "10.168.192.in-addr.arpa" IN {                  //ทำการเพิ่มค่า IP Address ตามหลักการแปลงค่า IP โดยจะ
ใส่ค่า กลับกัน                                     // จากค่า 192.168.10.x ให้เขียนเป็น 10.168.192.in-addr.arpa
    type master;                                     // โดยไฟล์ที่ชื่อ server.ip นี้จะทำการแปลงค่า IP Address เป็น
ชื่อ domain                                         //
    file "server.ip";                               //
    allow-update { none; };
};

zone "thaiphpdev.com" IN {                          //เพิ่มชื่อ domain name คำอื่น ๆ
    type master;
    file "thaiphpdev.com.name";
    allow-update { none; };
};

include "/etc/rndc.key";

```


3. เพิ่มเพิ่มฐานข้อมูล Host ใน /var/named/

ทำการสร้างไฟล์ server.name โดยทำการ copy จากไฟล์ชื่อ named.local โดยใช้คำสั่งดังนี้

```
# cp named.local server.ip
```

แล้วทำการแก้ไฟล์ ดังตัวอย่างไฟล์ ข้างล่างนี้

```
/var/named/server.ip
```

```
$TTL 86400
@ IN SOA ns.server.com. root.ns.server.com. ( //กำหนดชื่อ DNS Server และ E-mail
ของ root
1997022700 ; Serial //ค่า config ต่าง ๆ ไม่ต้องเปลี่ยน
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS ns.server.com. //กำหนด ชื่อ DNS Server อันดับแรก
# IN NS ns1.server.com. //กำหนดชื่อ DNS Server อันดับสอง
**ถ้ามี**
5 IN PTR server.com. //แปลง ค่า IP Address เป็น domain
name ค่าต่าง ๆ
5 IN PTR www.server.com.
5 IN PTR mail.server.com.
5 IN PTR redhat.server.com.
```

และ ทำการสร้างไฟล์ server.name โดยทำการ copy จากไฟล์ server.ip

```
/var/named/server.name
```

```
$TTL 86400
@ IN SOA ns.server.com. root.ns.server.com. ( //กำหนดชื่อ DNS Server และ E-mail
ของ root
1997022700 ; Serial //ค่า config ต่าง ๆ ไม่ต้องเปลี่ยน
28800 ; Refresh
14400 ; Retry
3600000 ; Expire
86400 ) ; Minimum
IN NS ns.server.com. //กำหนด ชื่อ DNS Server อันดับแรก
```

IN	MX	10	mail.server.com.	//กำหนด ชื่อ Mail Exchanger อันดับ
www	IN	A	192.168.10.5	//แปลงค่า domain name เป็น ค่า IP
mail	IN	A	192.168.10.5	
redhat	IN	A	192.168.10.5	
server.com.	IN	A	192.168.10.5	

สังเกตการเพิ่มค่า domain ลงในไฟล์ทั้ง 2 อย่างลืมเพิ่ม จุด (.) ต่อท้ายชื่อ domain นั้นด้วย
จากนั้นทำการ restart แดมอน named โดยใช้คำสั่งดังนี้

```
# service named restart
```

ทำการทดสอบ DNS Server

เมื่อทำการ แก้ไขค่าในไฟล์ Config เสร็จแล้ว จากนั้นจึงทำการ restart แดมอน named เมื่อต้องการที่จะทดสอบค่า domain name ที่เราได้ทำการ Config เพิ่มขึ้น เราสามารถใช้คำสั่ง ping แล้วตามด้วยชื่อ domain name ที่ต้องการทดสอบ เช่น

```
# ping www.server.com //ตรวจสอบ domain www.server.com
```

```
# ping mail.server.com //ตรวจสอบ domain mail.server.com
```

หรือ จะใช้โปรแกรม nslookup ช่วยในการทดสอบอีกทางหนึ่งก็ได้ โดยเรียกใช้โปรแกรมดังตัวอย่าง

```
# nslookup //เริ่มต้นเข้าโปรแกรม
```

```
>www.server.com //ถามค่า IP Address จาก domain name
```

```
>192.168.10.5 //ถามค่า domain name จาก IP Address
```

```
>exit //ออกจากโปรแกรม
```

สรุปการติดตั้ง DNS Server

- install packet คือ bind-9.x.x.i386.rpm ,caching-nameserver-7.x.x.noarch.rpm
- แก้ไขไฟล์ Config resolv.conf โดยกำหนดค่า IP Address ของ เครื่อง DNS Server
- เพิ่มค่า Config ของ domain name ที่ต้องการในไฟล์ named.conf
- สร้างไฟล์ฐานข้อมูล Host ใน /var/named/ โดยไฟล์ต้นแบบคือ named.local
- Restart แดมอน named : # service named restart
- ทดสอบด้วย โปรแกรม ping หรือ nslookup

2. Database Server

Database บน เครือข่าย Internet ที่เราเลือกใช้คือ MySQL โดยท่านสามารถหา Software MySQL ได้จาก <http://www.mysql.com> โดยให้เลือก Download แบบ binary file

การติดตั้ง MySQL

ทำการ Download ไฟล์ mysql-3.23.49.tar.gz จาก <http://www.mysql.com> แล้ว copy ที่ต้องการ install ไว้ที่ /tmp จากนั้นทำตามขั้นตอนด้านล่าง

```
# tar xzvf mysql-3.23.49.tar.gz           //แตกไฟล์ mysql
# cd mysql-3.23.49
# useradd mysql                          //create user mysql
# ./configure --prefix=/usr/local/mysql  //ทำการ configure Ccheck path ที่ต้องการinstall
# make                                    //Compile
# make install                            //Install Program
# scripts/mysql_install_db               //สร้างฐานข้อมูล
# chown -R mysql:mysql /usr/local/mysql //เปลี่ยนค่ากลุ่มผู้ใช้งานในFolder mysql ทั้งหมด
# cp support-files/my-medium.cnf /etc/my.cnf //copy File config my.cnf
# cp support-file/mysql.server /etc/rc.d/init.d/mysql //File Start mysql
# cd /etc/rc.d/init.d
# chmod +x mysql                         //เลือกให้ไฟล์สามารถ execute ได้
# /sbin/chkconfig --del mysql
# /sbin/chkconfig --add mysql            //ทำการสร้างไฟล์ลิงค์ใน /rc3.d
# cd /etc/rc.d/rc3.d
# ls S90mysql                            //ตรวจสอบไฟล์ S90mysql
```

ทำการเพิ่มค่า PATH : "/usr/local/mysql/bin" ลงในไฟล์ profile ใน /etc
/etc/profile

```
# /etc/profile
# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc
PATH=$PATH:/usr/local/mysql/bin           //ทำการ เพิ่ม PATH
```

หมายเหตุ: หลังจากนั้นควรทำการ เพิ่ม user ลงในโปรแกรม MySQL เพื่อความปลอดภัย
การ Service MySQL start, stop, restart ใช้คำสั่ง ดังนี้

```
# cd /etc/rc.d/rc3.d
```

```
# ./S90mysql start | stop
```

หรือ

```
# service mysql start | stop | restart
```

ตรวจสอบการ Start MySQL ได้ จาก ไฟล์ /tmp/mysql.sock (MySQL is running) ที่เกิดจากหลังจากได้ทำการ Start MySQL แล้ว หรือใช้คำสั่ง

```
ps aux | grep mysql
```

3. IMAP Client

ใช้สำหรับเชื่อมต่อกับ IMAP Server หรือ POP3 Server โดยใช้ ฟังก์ชัน PHP IMAP ในการเชื่อมต่อกับServer ท่านสามารถSoftware imapclientได้ที่[ftp://ftp.cac.washington.edu/imap](http://ftp.cac.washington.edu/imap) และเลือก Download ไฟล์ชื่อว่า imap-2001a.tar.Z

การติดตั้ง IMAP Client

ทำการ copy ไฟล์ imap-2001a.tar.Z ไว้ที่ Folder /tmp ก่อน หลังจากนั้นทำตามขั้นตอนด้านล่าง

```
# cd /usr/
```

```
# tar xzf ../tmp/imap-2001a.tar.Z -C
```

```
# cd imap-2001a
```

```
# make slx
```

จากนั้นทำการ copy ไฟล์ c-client.a (ใน c-client) ไปไว้ที่ link path และ rfc822.h, mail.h ไปไว้ที่

Directory include file ดังตัวอย่าง เช่น

```
# cd c-client
```

```
# cp c-client.a /usr/local/lib
```

```
# cp rfc822.h mail.h linkage.h /usr/local/include
```

และเมื่อทำการติดตั้ง php โดยการ Compile ให้เพิ่ม `--with-imap="Directory imap install"`

4. Web Server

Web Server คือ Software ที่ให้บริการระบบ เวิร์ด ไรด์ เว็บ(World Wide Web)

WebServer ให้บริการถ่ายโอนข้อมูลเอกสาร HTML โดยรับการร้องขอจากBrowser โดยจะทำการติดต่อกันผ่าน Protocol HTTP(Hypertext Tranfer Protocal) การติดตั้ง บริการ WebServer จึงเป็นการติดตั้งเพื่อให้บริการจาก Protocol HTTP

การติดตั้ง Web Server

ในที่นี้ จะทำการติดตั้ง WebServer ให้สามารถรองรับการทำงานของ ภาษา PHP , Module IMAP Client และ Module MySQL โดยเลือกใช้โปรแกรม Apache เป็น WebServer จากนั้นทำการ Download โปรแกรม apache_1.3.24.tar.gz จาก <http://www.apache.org> และ Download php-4.1.2.tar.gz จาก <http://www.php.net> และ copy ไฟล์ต่างๆ ที่ Download มาไว้ที่ /tmp จากนั้นทำการ install ตามขั้นตอนต่อไปนี้

```
# tar xzvf apache_1.3.24.tar.gz
# tar xzvf php-4.2.1.tar.gz
# cd apache_1.3.24
# ./configure --prefix=/usr/local/apache
# cd ../php-4.2.1
# ./configure --with-apache=../apache_1.3.24 --with-ldap=/usr/local/imap-2001a --with-mysql
# make
# make install
# cd ../apache_1.3.24
# ./configure --prefix=/usr/local/apache --activate-module=src/module/php4/libphp4.a
# make
# make install
# cd /usr/local/apache/bin
# cp apachectl /etc/rc.d/init.d/httpd
# cd ../conf
```

จากนั้นใช้ Editor แก้ไขไฟล์ httpd.conf ที่ AddType เพิ่มนามสกุล .php .php3 .htm .html ดังตัวอย่าง

```
/usr/local/apache/conf/httpd.conf
```

```
Modify AddType for php this below
```

```
-----
AddType application/x-httpd-php .php
```

```
AddType application/x-httpd-php .php3
```

```
AddType application/x-httpd-php .htm
```

```
AddType application/x-httpd-php .html
```

```
AddType application/x-httpd-php-source .php
```

การ Service Apache start ,stop ,restart โดยใช้คำสั่ง ดังนี้

```
# service httpd start | stop | restart
# /etc/rc.d/init.d/httpd start | stop | restart
```

การติดตั้ง Apache + Apache SSL

ในการใช้ Apache SSL นั้น เราจะต้องทำการ สร้าง public key และ private key โดยใช้โปรแกรม

OpenSSL ก่อน โดยทำการ install packet openssl-0.9.6.rpm แล้ว ทำการ สร้าง public key และ private key ตามขั้นตอนดังต่อไปนี้

```
# cd /usr/share/ssl/private
# openssl genrsa -rand file1:...:file5 1024 > www.server.com.key
# cd ../certs
# openssl req -new -key ../private/www.server.com.key > www.server.com.csr
# openssl req -x509 -key ../private/www.server.com.key -in www.server.com.csr >
www.server.com.crt
```

เมื่อทำตามขั้นตอนเสร็จแล้ว จะได้ไฟล์ 3 ไฟล์ คือ

1. ไฟล์ www.server.com.key ใน /usr/share/ssl/private/
2. ไฟล์ www.server.com.crt ใน /usr/share/ssl/certs/
3. ไฟล์ www.server.com.csr ใน /usr/share/ssl/certs/

จากนั้นทำการ install Apache + Apache SSL

ทำการ tar Apache และ Apache SSL ใน Folder เดียวกัน จากนั้นจึงทำการปรับค่า Path ให้เป็น Path ของ Apache SSL โดยทำตามขั้นตอนดังนี้

```
# tar xzvf apache_1.3.24.tar.gz
# cd apache_1.3.24
# tar xzvf ../apache_1.3.24+ssl_1.48.tar.gz
```

ทำการปรับค่า Path

```
# patch -pl < SSLpatch
```

ตรวจสอบค่า Path ในไฟล์ src/Configuration.tmpl ใช้ Editor แก้ไขค่า Path ที่ไม่ตรง

```
# vi src/Configuration.tmpl
```

```
#KEYNOTE_BASE=/home/
SSL_BASE=/usr/share/ssl
SSL_INCLUDE= -I/usr/include/openssl -I$(SSL_BASE/include)
SSL_CFLAGS= -DAPACHE
SSL_LIB_DIR=/usr/lib
SSL_LIBS= -L$(SSL_LIB_DIR) -lssl -lcrt
SSL_APP_DIR=/usr/bin
SSL_APP=/usr/bin/openssl
```

Save ไฟล์แล้ว เริ่มทำการ install

```
# ./configure --prefix=/usr/local/apache
# make
# make install
ทำการ copy ไฟล์ Service ของ Apache SSL ไปที่ /etc/rc.d/init.d/ และเปลี่ยนชื่อเป็น httpd
# cp /usr/local/apache/bin/httpsdctl /etc/rc.d/init.d/httpd
# cp /usr/local/apache/conf/httpsd.conf /usr/local/apache/conf/httpd.conf
และทำการแก้ไขไฟล์ Config ดังนี้ httpd.conf
/usr/local/apache/conf/httpd.conf
```

```
# Section 2: 'Main' server configuration
..
# Port: The port to which the standalone server listens. For
# ports < 1023, you will need httpd to be run as root initially.
Include /usr/local/apache/conf/httpd-ssl.conf
..
<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.php
</IfModule>
..
#Section 3: Virtual Hosts
# Use name-based virtual hosting.
#
NameVirtualHost *:80
```

```
NameVirtualHost *:443
Include /usr/local/apache/conf/virtual_80.conf
Include /usr/local/apache/conf/virtual_443.conf
```

ทำการสร้างไฟล์ httpd-ssl.conf

/usr/local/apache/conf/httpd-ssl.conf

```
#----- SSL -----#
Port 443
Listen 443
Listen 80
SSLEnable
SSLNoCAList
SSLRandomFile /dev/urandom 1024
SSLRandomFilePerConnection /dev/urandom 1024
TransferLog logs/transfer_log
SSLCacheServerPath /usr/local/apache/bin/gcache
SSLCacheServerPort logs/gcache_port
SSLCacheServerRunDir /tmp
SSLSessionCacheTimeout 15
SSLCACertificatePath /usr/share/ssl/certs
SSLCertificateFile /usr/share/ssl/certs/www.server.com.crt
SSLCertificateKeyFile /usr/share/ssl/private/www.server.com.key
SSLVerifyClient 0
SSLVerifyDepth 10
SSLFakeBasicAuth
SSLBanCipher NULL-MD5:NULL-SHA
CustomLog logs/ssl_log "%t%{version}c %c %c %c"
#----- SSL -----#
```


สร้างไฟล์ virtual_443.conf และ virtual_80.conf

/usr/local/apache/conf/virtual_443.conf

```
<VirtualHost *:443>
SSLEnable
SSLCACertificatePath /usr/share/ssl/certs
SSLCertificateFile /usr/share/ssl/certs/www.server.com.crt
SSLCertificateKeyFile /usr/share/ssl/private/www.server.com.key
DocumentRoot /usr/local/apache/htdocs/
ServerName www.server.com
</VirtualHost>
```

/usr/local/apache/conf/virtual_80.conf

```
<VirtualHost *:80>
SSLDisable
DocumentRoot /usr/local/apache/htdocs/www/
ServerName server.com
</VirtualHost>

<VirtualHost *:80>
SSLDisable
DocumentRoot /usr/local/apache/htdocs/www/
ServerName www.server.com
</VirtualHost>

<VirtualHost *:80>
SSLDisable
DocumentRoot /usr/local/apache/htdocs/mail/
ServerName mail.server.com
</VirtualHost>
```

ภาคผนวก ข

การตั้งค่าของ Program ต่างๆ เพื่อให้รองรับการทำงานของอีเมลล์

1. Dreamweaver Ultradev

การที่เลือกใช้ Dreamweaver Ultradev เพราะเป็นโปรแกรมที่พัฒนาต่อมาจากโปรแกรม Dreamweaver ธรรมดา ดังนั้น Dreamweaver Ultradev จึงเป็นโปรแกรมที่มีความสามารถของ Dreamweaver อยู่ครบถ้วนไม่ถูกตัดทอนส่วนใดส่วนหนึ่งออกไป ตรงกันข้าม Dreamweaver Ultradev ยังได้รับการเพิ่มความสามารถพิเศษในการเขียนสคริปต์เข้าไปด้วย ทำให้ง่ายในการเขียนสคริปต์

ขั้นตอนการเซต Font ภาษาไทยให้ใช้งานได้บน Dreamweaver Ultradev

- คลิกที่ Edit แล้วเลือก Preferences...
- เมื่อหน้าต่าง Preferences เปิดออก ในช่อง Category ให้คลิกที่ Fonts / Encoding แล้วกด

Default Encoding	>	Other
Font Setting	>	Other
Proportional Font	>	font ภาษาไทย

- แล้วคลิก OK ถ้าเกิดยังใช้ภาษาไทยไม่ได้อีก เวลาเราจะใช้งานให้เซตเพิ่มเติม ดังนี้
- คลิก Modify แล้วเลือก Page Properties...
- ในหัวข้อ Document Encoding ให้กำหนดค่าเป็น Other แล้วคลิก OK

ขั้นตอนการ Setup โปรแกรม Dreamweaver Ultradev ให้สามารถใช้งาน php ได้

- ลงโปรแกรม Dreamweaver Ultradev ให้เรียบร้อย
- ดาวน์โหลด Extension ของ php มาจากเว็บของ Macromedia
- ลง Extension ของ php
- เวลาเปิดใช้งาน ในหน้า Data Bindings ให้คลิก Choose Server Model
- หน้าจอ Site Definitions for UD4 Tutorial จะเปิดออก แล้วคลิก Application Server
- ในช่อง Server Model ให้เลือกเป็น php แล้วคลิก OK
- ตอนนี้เราก็สามารถเขียนสคริปต์ของ php ได้

2. EditPlus

เป็นโปรแกรม Editor ที่มีความหลากหลายในการทำงาน ช่วยให้เราสามารถเลือกใช้แท็กของแต่ละภาษาในการเขียนโปรแกรมได้อย่างรวดเร็ว โดยที่เราไม่ต้องมาเขียนแท็กของภาษาต่าง ๆ เอง ซึ่ง EditPlus สามารถช่วยในการสร้างแท็กภาษา ดังนี้ HTML, CSS, PHP, ASP, Perl, C/C++, Java, JavaScript และ VBScript เราสามารถดาวน์โหลด EditPlus ได้จากเว็บไซต์ <http://www.editplus.com/download.html>

ขั้นตอนการเซตให้โปรแกรม EditPlus ใช้ Font ภาษาไทยได้

- เปิดโปรแกรมมาแล้วให้คลิกที่ Tools
- เลือก Preference... จะขึ้นหน้าต่าง Preference ให้
- ในช่องของ Categories ให้คลิกที่ Fonts
- ในส่วนของ Fonts ให้เลือก Font ที่ใช้ภาษาไทยได้
- แล้วคลิก OK

ขั้นตอนการเซตให้โปรแกรม EditPlus สามารถใช้ php ได้

- เปิดโปรแกรม EditPlus ขึ้นมา
- คลิกตรง File แล้วเอาเมาส์มาวางที่ New
- ในส่วนของ New จะมีภาษาต่าง ๆ ให้เลือก ถ้าเกิดเราจะใช้ภาษาใดก็สามารถเลือกได้
- ถ้าจะใช้ php ให้คลิกที่ Other จะปรากฏ php ขึ้น
- คลิกที่ php จะมีแท็ก php มาช่วยในการเขียน

3. Photoshop

เป็นโปรแกรมที่ช่วยในการตกแต่งรูปภาพ โดยสามารถสร้างรูปภาพขึ้นมาใหม่ หรือนำรูปภาพเดิมที่มีอยู่มาตกแต่งด้วย Photoshop ให้เหมาะสมกับการใช้งาน ก่อนการใช้งาน Photoshop ถ้าต้องการให้ Photoshop สามารถใช้ภาษาไทยได้ เราต้องทำการ Add Font ภาษาไทยเข้าไปก่อน

ขั้นตอนการติดตั้ง Font ภาษาไทยให้กับ Photoshop

- ใส่แผ่น CD-ROM ที่ Font ภาษาไทยลงไป
- เลือกเมนู Start -> Settings -> Control Panel เพื่อเปิดหน้าต่างวินโดว์
- ในวินโดว์ Control Panel ให้ดับเบิลคลิกไอคอน Fonts

- ในวินโดว์ Fonts ให้เลือกเมนู File -> Install New Font เพื่อติดตั้ง Font ใหม่
- ในวินโดว์ Add Fonts ให้เลือกใคร่ไปยังใคร่ที่มี Fonts ภาษาไทยเก็บอยู่ และเลือกโฟลเดอร์ที่มี Fonts ภาษาไทยเก็บอยู่ จะปรากฏรายชื่อ Fonts ให้คลิกเช็คถูกในช่อง Copy fonts to Fonts folder จากนั้น คลิกปุ่ม Select All เพื่อเลือก Fonts ทั้งหมด และคลิก OK
- ตอนนี้ Fonts ภาษาไทยของเราก็ติดตั้งลงในโฟลเดอร์ Fonts ของระบบวินโดว์ เราสามารถใช้งานได้แล้ว



ประวัติผู้ทำโครงการ

ชื่อ นายดำรงค์ วงศ์เล็กภัย
 วัน เดือน ปีเกิด 2 กุมภาพันธ์ 2524
 สถานที่เกิด จังหวัด พิชณุโลก
 ประวัติการศึกษา มัธยมศึกษาตอนต้น โรงเรียนพิจิตรพิทยาคม
 จังหวัด พิจิตร พ.ศ.2537
 มัธยมศึกษาตอนปลาย โรงเรียนจุฬากรณ์ราชวิทยาลัย
 จังหวัด พิชณุโลก พ.ศ.2540

ชื่อ นายอนุฤทธิ์ ศรีคมขำ
 วัน เดือน ปีเกิด 26 พฤษภาคม 2524
 สถานที่เกิด จังหวัด พิชณุโลก
 ประวัติการศึกษา มัธยมศึกษาตอนต้น โรงเรียนพุทธชินราชวิทยา
 จังหวัด พิชณุโลก พ.ศ.2537
 มัธยมศึกษาตอนปลาย โรงเรียนพุทธชินราชวิทยา
 จังหวัด พิชณุโลก พ.ศ.2540

ชื่อ นางสาวสลักจิตร์ มนอิน
 วัน เดือน ปีเกิด 13 มกราคม 2523
 สถานที่เกิด จังหวัด สกลนคร
 ประวัติการศึกษา มัธยมศึกษาตอนต้น โรงเรียนร่วมเกล้าสกลนคร
 จังหวัด สกลนคร พ.ศ.2537
 มัธยมศึกษาตอนปลาย โรงเรียนร่วมเกล้าสกลนคร
 จังหวัด สกลนคร พ.ศ.2540