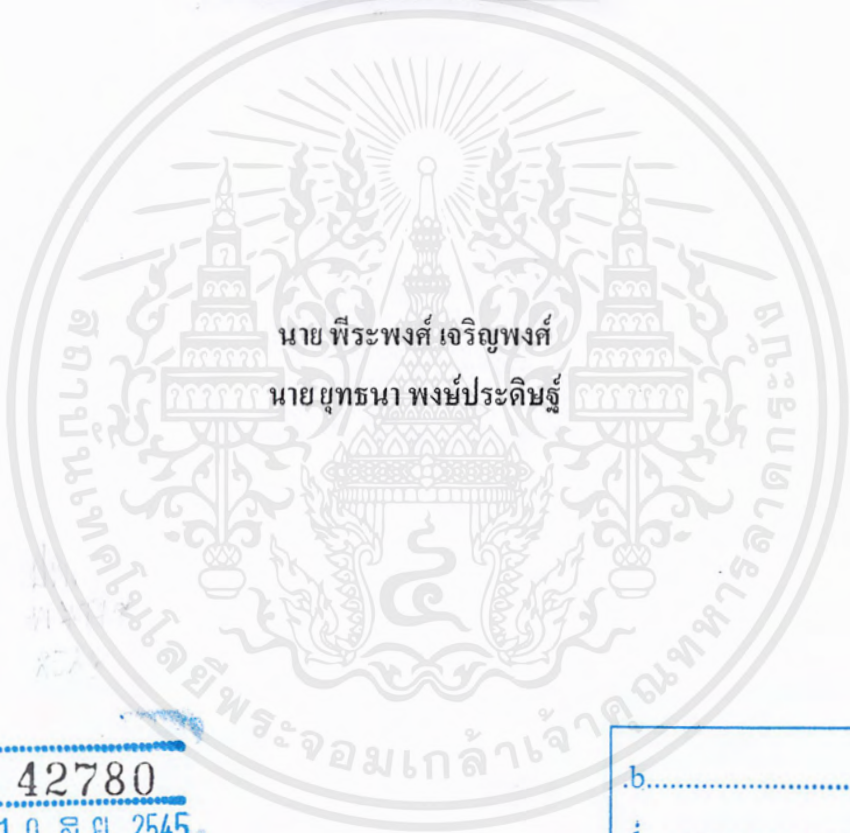


ระบบจดหมายอิเล็กทรอนิกส์บนเว็บ

Electronic mail on web



นาย พีระพงศ์ เจริญพงศ์
นาย ยุทธนา พงษ์ประดิษฐ์

เลขที่.....
เลขทะเบียน... 42780
วัน, เดือน, ปี 10 ส.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจดหมายอิเล็กทรอนิกส์บนเว็บ

Electronic mail on web



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

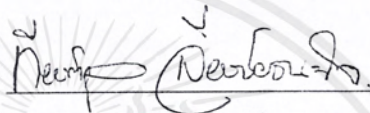
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจดหมายอิเล็กทรอนิกส์บนเว็บ

Electronic mail on web

ผู้จัดทำ

1. นาย พีระพงศ์ เจริญพงศ์ รหัสประจำตัว 40010526
2. นาย ยุทธนา พงษ์ประดิษฐ์ รหัสประจำตัว 40010619



อาจารย์ที่ปรึกษา

(อ.เกียรติกุล เจียรนัยระกิจ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจดหมายอิเล็กทรอนิกส์บนเว็บ

Electronic mail on web

ผู้จัดทำ

1. นาย พีระพงศ์ เจริญพงศ์ รหัสประจำตัว 40010526

2. นาย ชุทธนา พงษ์ประดิษฐ์ รหัสประจำตัว 40010619

อาจารย์ที่ปรึกษา

(อ.เกียรติกุล เขียรนัยชนะกิจ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจดหมายอิเล็กทรอนิกส์บนเว็บ

นาย พิระพงศ์ เจริญพงศ์ 40010526

นาย ยุทธนา พงษ์ประดิษฐ์ 40010619

อ.เกียรติคุณ เจียรนัยชนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ระบบจดหมายอิเล็กทรอนิกส์บนเว็บถูกออกแบบสำหรับให้ผู้ใช้สะดวกในการรับส่งจดหมายอิเล็กทรอนิกส์ของทางภาควิชาคอมพิวเตอร์ โดยทั่วไปผู้ใช้จะคุ้นเคยกับโปรแกรมพาย ซึ่งอาจมีการใช้งานยากสำหรับผู้ใช้ที่เริ่มใช้งาน จึงได้จัดทำระบบรับส่งจดหมายอิเล็กทรอนิกส์บนเว็บขึ้นเพื่อให้โปรแกรมรับส่งจดหมายอิเล็กทรอนิกส์เป็นมิตรกับผู้ใช้มากขึ้น ซึ่งได้มีการนำเทคโนโลยีของจาวามาใช้ในการออกแบบและเขียนโปรแกรม เนื่องจากภาษาจาวาเป็นภาษาเชิงวัตถุ สนับสนุนการนำมาใช้ใหม่และทำให้การพัฒนาเป็นไปอย่างง่ายขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electronic Mail on Web

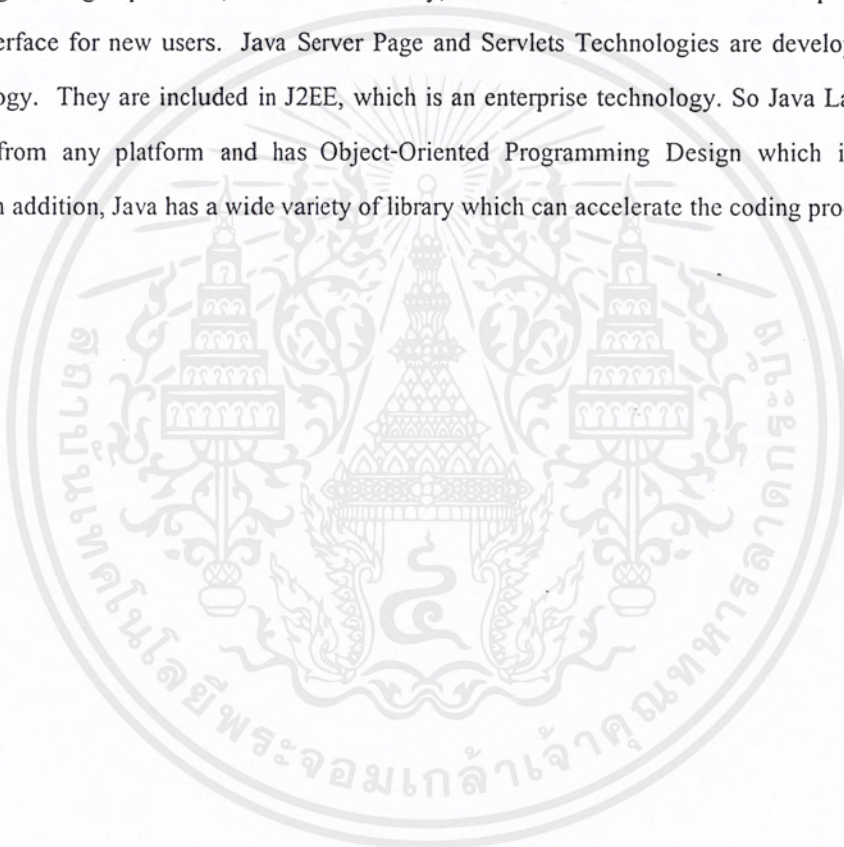
Peerapong Charoenpong

Yuttana Pongpradit

Kietkul Jeeranaitanakij Advisor

ABSTRACT

Web based Electronic mail is designed for users to conveniently access to their e-mail in the Computer Engineering department, KMUTL. Normally, users are familiar with the PINE program that has a bad interface for new users. Java Server Page and Servlets Technologies are developed from Java Technology. They are included in J2EE, which is an enterprise technology. So Java Language is independent from any platform and has Object-Oriented Programming Design which is easy to implement. In addition, Java has a wide variety of library which can accelerate the coding process.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

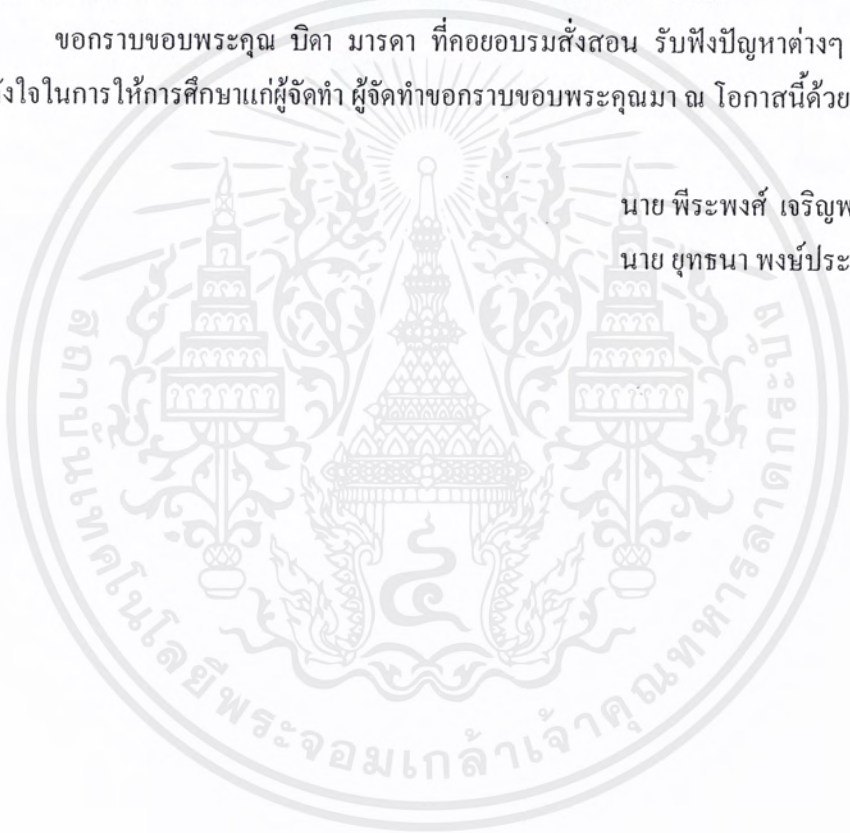
กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ สำเร็จลงได้ด้วยดี เพราะความกรุณาจากอาจารย์ต่างๆ ที่ให้การอบรมสั่งสอนวิชาความรู้ตั้งแต่ต้นจนถึงปัจจุบัน โดยเฉพาะอย่างยิ่ง อาจารย์ที่ปรึกษา อาจารย์ เกียรติคุณ เจียรนัยชนะกิจ ที่คอยให้คำปรึกษา พร้อมทั้งแนวคิด ข้อเสนอแนะ ตลอดจนการแก้ไขข้อบกพร่องต่างๆ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูงมา ณ โอกาสนี้

ขอกราบขอบพระคุณ ศูนย์วิจัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้สถานที่ในการทำงาน และเพื่อนๆ ทุกคนที่ให้ความสนับสนุนผู้จัดทำ โดยการให้คำแนะนำ ข้อคิดต่างๆ และคอยให้ความช่วยเหลือ และคอยให้กำลังใจในการทำงานจนสำเร็จลงด้วยดี

ขอกราบขอบพระคุณ บิดา มารดา ที่คอยอบรมสั่งสอน รับฟังปัญหาต่างๆ และทุ่มเทกำลังใจกำลังใจในการให้การศึกษแก่ผู้จัดทำ ผู้จัดทำขอกราบขอบพระคุณมา ณ โอกาสนี้ด้วย

นาย พีระพงศ์ เจริญพงศ์
นาย ยุทธนา พงษ์ประดิษฐ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีประกอบโครงงาน	4
2.1 การสื่อสารในระบบจดหมายอิเล็กทรอนิกส์	4
2.1.1 หน้าที่หลักของ MTA	4
2.1.2 โพรโตคอลประยุกต์ที่ใช้ในระบบจดหมายอิเล็กทรอนิกส์	5
2.1.2.1 POP3 (Post Office Protocol)	5
2.1.2.2 SMTP (Simple Mail Transfer Protocol)	7
2.1.2.3 ไอแมป (IMAP)	8
2.2 การพัฒนาโปรแกรมเชิงวัตถุ (Object Oriented Programming)	9
2.2.1 Methodologies	10
2.2.2 Information Engineering	11
2.2.3 Object-oriented methodologies	11
2.3 UML (Unified Modeling Language)	12
2.3.1 แผนภาพและสัญลักษณ์ของ UML	17
2.3.2 รูปแบบของการออกแบบ (Design Pattern)	18
2.4 สถาปัตยกรรมต่างๆที่เกี่ยวข้องกับระบบ	22
2.4.1 Java Servlet	22
2.4.2 JSP	24
2.4.3 Application Server	26
2.5 การประยุกต์ใช้แพทเทิร์น โมเดล-วิว-คอนโทรลเลอร์กับระบบจดหมายอิเล็กทรอนิกส์	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบและการสร้าง	30
3.1 การออกแบบ	30
3.2 การสร้าง	32
3.3 วิว(View)	33
3.4 โมเดล	36
บทที่ 4 การทดลองการทำงานและการแสดงผล	40
4.1 การติดตั้งระบบเว็บเมล์	40
4.2 การทดลองการทำงาน	40
4.3 ผลการทดลองการทำงาน	48
4.4 สรุปผลการทดลอง	49
บทที่ 5 สรุปและวิจารณ์	50
5.1 ปัญหาและอุปสรรคในการจัดทำโครงงาน	50
5.2 สรุปผลที่ได้จากการทำโครงงาน	50
5.3 ข้อเสนอแนะ	50
ภาคผนวก	52
ภาคผนวก ก คลาสทั้งหมดในระบบ	53
ภาคผนวก ข คลาสไดอะแกรมส่วนจัดการค่าคงที่ตัวแปรที่ใช้ในระบบ	56
ภาคผนวก ค อินเทอร์เน็ตทั้งหมดในระบบ	57
บรรณานุกรม	58

สารบัญรูปภาพ

หน้าที่

รูปที่ 2-1 แสดงระบบอิเล็กทรอนิกส์เมล์	4
รูปที่ 2-2 แสดงเทคนิคโครงสร้างข้อมูลในสมัยก่อน	10
รูปที่ 2-3 แสดงการเปรียบเทียบระหว่างข้อดีของเทคนิค Methodology	10
รูปที่ 2-4 แสดงเทคนิคแบบ Information Engineering	11
รูปที่ 2-5 แสดงการเปรียบเทียบระหว่างข้อดีของเทคนิค Information Engineering	11
รูปที่ 2-6 แสดงการเปรียบเทียบระหว่างข้อดีของเทคนิค Object Oriented Methodology	12
รูปที่ 2-7 แสดงการซ่อนรายละเอียดของข้อมูลในเทคนิค Object Oriented Methodology	12
รูปที่ 2-8 แสดงรูปผู้พัฒนา UML และสัญลักษณ์ที่ใช้แทน UML	13
รูปที่ 2-9 แสดงลักษณะของโมเดลที่มีการพัฒนาก่อนที่จะมาเป็น UML ของแต่ละคน	13
รูปที่ 2-10 แสดงส่วนต่างๆที่มีความเกี่ยวข้องกับภาษา UML	14
รูปที่ 2-11 แสดงตัวอย่างแผนภาพ Use Case	14
รูปที่ 2-12 แสดงสัญลักษณ์และความหมายต่างๆของ Class Diagram	15
รูปที่ 2-13 แสดงตัวอย่าง Class Diagram	16
รูปที่ 2-14 แสดงตัวอย่าง State Diagram	16
รูปที่ 2-15 แสดงตัวอย่าง Component Diagram	17
รูปที่ 2-16 แสดงตัวอย่าง Activity Diagram	17
รูปที่ 2-17 โครงสร้างของดีไซน์แพทเทิร์น Model-View-Controller	20
รูปที่ 2-18 โครงสร้างการทำงานของดีไซน์แพทเทิร์น Model-View-Controller	21
รูปที่ 2-19 แสดง Servlet Engine และการทำงาน	22
รูปที่ 2-20 รูปของระบบที่ประกอบไปด้วย Servlet Engine และระบบอื่นๆ	23
รูปที่ 2-21 แสดงการทำงานเริ่มต้นที่มีการอ้างอิงมาที่ JSP นี้	25
รูปที่ 2-22 การทำงานของ Application Server	26
รูปที่ 2-23 การติดต่อกันระหว่าง Tier ต่างๆ	27
รูปที่ 2-24 MVC ที่นำมาประยุกต์กับ JSP	28
รูปที่ 2-25 JSP Model 2 Architecture	29
รูปที่ 3-1 MVC ที่นำมาใช้ใน Web-base E-mail	30
รูปที่ 3-2 Sequential diagram ในการส่งเมลล์	31
รูปที่ 3-3 Sequential diagram การอ่านจดหมาย	32
รูปที่ 3-4 แสดง Class Diagram ของ class ใน model	36
รูปที่ 4-1 แสดงหน้าแรกของการเข้าสู่ระบบ	42
รูปที่ 4-2 แสดงหน้าจอเมื่อทำการเข้าสู่ระบบได้แล้ว	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-3 แสดงหน้าจอเข้าสู่ผู้จดหมาย	43
รูปที่ 4-4 แสดงหน้าจออ่านจดหมาย	43
รูปที่ 4-5 แสดงหน้าจอเข้าสู่โหมดPrintable	44
รูปที่ 4-6 แสดงหน้าจอการตอบจดหมาย	44
รูปที่ 4-7 แสดงหน้าจอการเขียนจดหมายใหม่	45
รูปที่ 4-8 แสดงหน้าจอการการสร้างโฟลเดอร์ใหม่	45
รูปที่ 4-9 แสดงหน้าจอผลลัพธ์จากการสร้างโฟลเดอร์ใหม่	46
รูปที่ 4-10 แสดงหน้าจอการเลือกโฟลเดอร์ที่จะทำการลบ	46
รูปที่ 4-11 แสดงหน้าจอผลลัพธ์ที่ได้จากการลบ	47
รูปที่ 4-12 แสดงหน้าจอทดลองใส่ข้อมูลที่ไม่มีอยู่จริง	47
รูปที่ 4-13 แสดงหน้าจอการแสดงความผิดพลาดบอกแก่ผู้ใช้	48



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในระยะเวลาหลายปีที่ผ่านมา ระบบจดหมายอิเล็กทรอนิกส์ (Electronic Mail System) ได้มีบทบาทมากขึ้นในองค์กรธุรกิจและสถาบันต่างๆ มากขึ้นเรื่อยๆ ไม่ว่าจะเป็นด้านการจัดการ การส่งข้อมูลระหว่างหน่วยงานภายในองค์กรและต่างองค์กร การติดต่อสื่อสารต่างๆ ล้วนแต่เป็นสิ่งที่ระบบจดหมายอิเล็กทรอนิกส์เข้าไปมีส่วนร่วมในการทำให้ข้อมูลที่ส่งถึงกันให้เป็นไปอย่างมีประสิทธิภาพและรวดเร็วขึ้น และได้มีการเปลี่ยนรูปแบบในการรับ-ส่งจดหมายให้อยู่ในภายในเว็บเพจขึ้น

หลายๆคนคงรู้จักโปรแกรมที่ทำหน้าที่รับ-ส่ง จดหมายอิเล็กทรอนิกส์ ในยูนิคซ์ที่มีชื่อว่าไพน์ (PINE) โดยไพน์นั้นทำงานผ่านทางเทอร์มินอล แบบเท็กซ์โหมดหรือ โหมดที่มีแต่ตัวอักษร และเนื่องจากเหตุนี้เองทำให้ความสามารถบางอย่างของไพน์ผู้ใช้ไม่สามารถทราบได้หมดได้นอกจากจะอ่านได้จากคู่มือ และในกรณีที่ผู้ใช้อยู่ในสภาวะแวดล้อมที่ถูกปิดกั้นโดยไฟร์วอลล์ โดยปกติแล้ว ไฟร์วอลล์จะทำหน้าที่ปิดกั้นช่องว่างที่ติดต่อกับภายนอกให้เหลือแต่เพียงพอร์ต 80 ที่ใช้สำหรับเปิดเว็บเท่านั้น ทำให้ผู้ใช้ไม่สามารถ login เข้าไปเพื่อเปิด ไพน์ เพื่ออ่านจดหมายอิเล็กทรอนิกส์ได้ โดยปกติแล้ว ไฟร์วอลล์จะทำหน้าที่ปิดกั้นช่องว่างที่ติดต่อกับภายนอกให้เหลือแต่เพียงพอร์ต 80 ที่ใช้สำหรับเปิดเว็บเท่านั้น ทำให้ผู้ใช้ไม่สามารถล็อกอินเข้าไปเพื่อเปิดไพน์เพื่ออ่านจดหมายอิเล็กทรอนิกส์ได้ ดังนั้นจึงได้มีการปรับปรุงการเก็บค่าที่อยู่ปลายทางในการผ่านจดหมายไป (forward mail)

เพราะความต้องการระบบจดหมายอิเล็กทรอนิกส์ผ่านเว็บเพิ่มขึ้นอย่างรวดเร็ว เนื่องจากทำให้เกิดความสะดวกแก่ผู้ใช้ อีกทั้งยังสามารถนำมาประยุกต์ใช้งานได้เหมาะสมแก่ความต้องการ เราจึงเกิดแนวความคิดที่จะทดลองสร้างและพัฒนาระบบจดหมายอิเล็กทรอนิกส์ผ่านเว็บขึ้น ซึ่งจะประยุกต์ใช้กับการรับ-ส่ง จดหมายอิเล็กทรอนิกส์ภายใน ภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

โดยเริ่มแรกจะทำการศึกษาถึงโพรโตคอลต่างๆที่ใช้ในการรับและส่งจดหมายอิเล็กทรอนิกส์รวมถึงการศึกษารูปแบบโปรแกรมโดยใช้วิธีพัฒนาแบบเชิงวัตถุ เนื่องจากการพัฒนาโปรแกรมเชิงวัตถุสนับสนุนการนำกลับไปใช้ใหม่

ส่วนในด้านความสำคัญนั้น ด้วยเหตุที่ว่าโปรแกรมนี้จะสามารถช่วยให้การรับและส่งจดหมายอิเล็กทรอนิกส์มีความสะดวกมากยิ่งขึ้น

สรุปจากที่มาและเหตุผลที่กล่าวมาข้างต้น จึงเกิดเป็นที่มาของโครงการ ระบบจดหมายอิเล็กทรอนิกส์ผ่านเว็บขึ้น

1.2 วัตถุประสงค์ของโครงการวิจัย

- 1.2.1 เพื่อศึกษาลักษณะการทำงานของระบบจดหมายอิเล็กทรอนิกส์ในปัจจุบัน
- 1.2.2 เพื่อศึกษาความสามารถและการประยุกต์ให้การทำงานได้ตรงกับโปรแกรม โฟล์
- 1.2.3 เพื่อศึกษาการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming)
- 1.2.4 เพื่อศึกษาเทคโนโลยีจาวา (Java Technology) เช่น เจเอสพี (JSP) , เซิร์ฟเล็ต (Servlets)
- 1.2.5 เพื่อแก้ปัญหาเมื่อผู้ใช้ไม่สามารถล็อกอินเข้ามาใช้โปรแกรมโฟล์ได้

1.3 ขอบเขตของโครงการวิจัย

ปริญญานิพนธ์นี้ จะกล่าวถึง การสร้างระบบจดหมายอิเล็กทรอนิกส์ผ่านเว็บในรูปแบบของการใช้กับเซิร์ฟเวอร์แม่ข่ายของทางภาควิชา และของทางสถาบัน โดยผ่านทางเว็บเซิร์ฟเวอร์ที่ให้บริการไว้ ลักษณะการทำงานหรือข้อกำหนด มีดังนี้

- 1.3.1 การรับส่งจดหมายอิเล็กทรอนิกส์จากเซิร์ฟเวอร์สามารถทำงานได้ทั้ง เซิร์ฟเวอร์ที่ใช้ โพรโทคอลแบบป๊อป3 (POP3) และแบบไอแมป (IMAP)
- 1.3.2 สนับสนุนการส่งจดหมายอิเล็กทรอนิกส์ผ่านทาง โพรโทคอลเอสเอ็มทีพี (SMTP)
- 1.3.3 ผู้ใช้สามารถส่งตรวจสอบไฟล์ที่มีคนอื่นส่งมาให้ว่ามีไวรัสคอมพิวเตอร์ หรือ โทรจันหรือไม่ (ขึ้นอยู่กับโปรแกรมสแกนไวรัสที่ติดตั้งไว้)
- 1.3.4 ผู้ใช้สามารถทำการส่งไฟล์ได้ทีละหลายๆไฟล์ในการส่งครั้งเดียว
- 1.3.5 สามารถนำการตั้งค่าต่างๆบน โปรแกรมโฟล์นำมาใช้ร่วมกัน รวมทั้งเมื่อมีการเปลี่ยนแปลงค่าจากทางเว็บก็สามารถมีผลไปยังการตั้งค่าเมื่อเปิดผ่านทางโปรแกรมโฟล์ด้วย

เพราะมีความต้องการที่จะให้เกิดความสะดวกในการรับ-ส่ง จดหมายอิเล็กทรอนิกส์ ให้สามารถเหมาะสมกับผู้ใช้งานมากขึ้น โครงการวิจัยนี้ จึงมีข้อกำหนดที่เป็นแนวทางกว้างๆ ให้สามารถพัฒนาและประยุกต์ให้ใกล้เคียงกับลักษณะงานที่จะนำไปใช้และเป็นไปตามความต้องการของผู้ใช้ต่อไปได้

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลัก ๆ อยู่ 3 เรื่องด้วยกัน คือ การพัฒนาโปรแกรมเชิงวัตถุ การทำงานของระบบจดหมายอิเล็กทรอนิกส์ และเทคโนโลยีเซิร์ฟเล็ตและเจเอสพี ซึ่งมีรายละเอียดดังในบทที่ 2 จากนั้นก็จะนำเอาความรู้ที่ได้ศึกษาทั้งหมดมาออกแบบสถาปัตยกรรมของระบบและออกแบบระบบ ซึ่งเป็นรูปแบบข้อมูลที่จะใช้เก็บแบบจำลองข้อมูลระดับแนวคิด ซึ่งมีรายละเอียดในบทที่ 3

จากนั้นก็เริ่มเข้าสู่ขั้นตอนของการพัฒนาโปรแกรม โดยจะกล่าวถึงองค์ประกอบโดยรวมของระบบที่พัฒนาขึ้นมาทั้งหมด และยังอธิบายไปถึงรูปแบบการติดต่อกับผู้ใช้ และการประมวลผลเบื้องต้นก่อนที่จะส่งไปให้กับส่วนประกอบต่าง ๆ จากนั้นจะอธิบายแต่ละส่วน โดยเริ่มจากส่วนนิยามข้อมูลซึ่งจะเป็นการสร้างวิวกให้กับชนิดความจริง และส่วนนำเข้าข้อมูล

สำหรับบทที่ 4 ก็จะเป็นการทดสอบระบบรวมทั้งหมด และบทที่ 6 ซึ่งเป็นบทสุดท้ายก็จะเป็นการสรุปการทำงาน ผลที่ได้รับจากงานวิจัยชิ้นนี้ และแนวทางในการพัฒนางานวิจัยนี้เพิ่มเติม และแนวทางในการนำไปประยุกต์ใช้



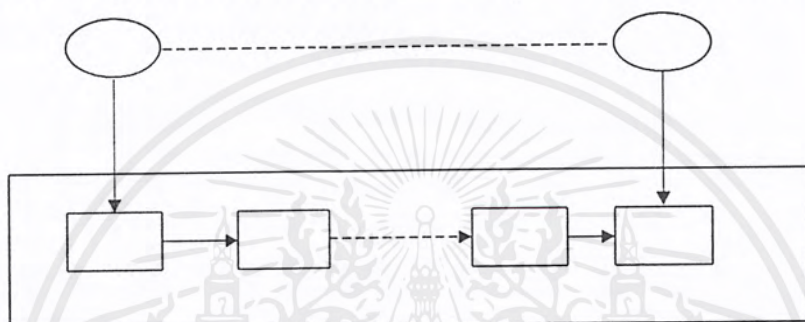
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีประกอบโครงงาน

2.1 การสื่อสารในระบบจดหมายอิเล็กทรอนิกส์

อิเล็กทรอนิกส์เมลล์เมสเสจจะมีการรับ-ส่งโดยระบบเมสเสจทรานส์เฟอร์ (Message Transfer System หรือ MTS) ซึ่งจะประกอบด้วยเมสเสจทรานส์เฟอร์เอเจนต์ (Message Transfer Agents หรือ MTA) เป็นระบบกระจายเมสเสจซึ่งอาจมี 1 เมสเสจทรานส์เฟอร์ หรือมากกว่า และที่ส่วนปลายของระบบจะเป็นยูเซอร์เอเจนต์ (user agent หรือ UA) ซึ่งจะทำหน้าที่ติดต่อกับผู้ใช้และเมสเสจทรานส์เฟอร์เอเจนต์



รูปที่ 2-1 แสดงระบบอิเล็กทรอนิกส์เมลล์

2.1.1 หน้าที่หลักของ MTA

- รับเมสเสจจากยูเซอร์เอเจนต์ทางฝั่งผู้ส่ง ไปยังยูเซอร์เอเจนต์ทางฝั่งผู้รับ
- รับเมสเสจจาก MTA อื่นเพื่อส่งไปยังยูเซอร์เอเจนต์ของผู้รับหรือส่งต่อไปยัง MTA อื่น
- ทำการวิเคราะห์รายชื่อผู้รับในเมสเสจและทำการตัดสินใจเลือกเส้นทางในการส่ง
- หากแอดเดรสในเมสเสจมีมายังยูเซอร์เอเจนต์ที่ติดต่อกับ MTA นั้น ก็จะทำการส่งเมสเสจไปยังยูเซอร์เอเจนต์และทำการสร้าง delivery notification(DN)
- หากแอดเดรสที่เมสเสจจะส่งไปจำเป็นต้องส่งไปยังเมสเสจทรานส์เฟอร์เอเจนต์อื่น ก็จะทำการรีเลย์สำหรับเมสเสจทรานส์เฟอร์เอเจนต์ที่จะส่งไป
- หากแอดเดรสไม่สามารถหาได้ (Resolve) ก็จะทำการสร้าง nondelivery notification (NDN) สำหรับแอดเดรสนั้น

หน้าที่หลักของยูเซอร์เอเจนต์

- เมื่อเมสเสจถูกสร้าง จะมีการแปลงรูปแบบ (encoded) เพื่อให้ยูเซอร์เอเจนต์ทางฝั่งผู้รับสามารถแปลความหมายได้
- ทำการส่งเมสเสจตลอดจนข้อมูลต่างๆ ที่จำเป็นในการรับ-ส่ง ไปยัง MTA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ยูเซอร์เอเจนต์จะทำการรับเมสเสจจากเมสเสจทรานส์เฟอร์เอเจนต์ที่ยูเซอร์เอเจนต์ติดต่อด้วย ตลอดจนข้อมูลที่เกี่ยวข้องกับการส่งที่มากับเมสเสจ
- ทำหน้าที่แปลงรูปแบบเมสเสจ (decoded message) ที่ได้รับจากเมสเสจทรานส์เฟอร์เอเจนต์ และทำการนำเสนอให้กับผู้รับในรูปแบบที่เป็นมิตร (User friendly)

เมื่ออิเล็กทรอนิกส์เมลล์เมสเสจทำการส่งจากผู้ใช้ หนึ่ง ไปยังผู้ใช้อื่น จะมีลำดับเหตุการณ์ดังนี้ ผู้ส่งซึ่งเฉพาะเจาะจงแอดเดรสของผู้รับไปยังยูเซอร์เอเจนต์, ยูเซอร์เอเจนต์ทำการแทนที่แอดเดรสปลายทางและแอดเดรสของผู้ส่งไปยังส่วนที่เรียกว่า envelope และทำการส่งเมสเสจไปยัง MTA ซึ่งจะมีการพิจารณาถึง โพรโตคอลและรูปแบบของอิเล็กทรอนิกส์เมลล์เมสเสจที่ถูกส่งมาด้วยว่าถูกต้องตามรูปแบบหรือไม่ เมื่อรูปแบบโพรโตคอลและเมสเสจถูกต้อง จะมีการทำการส่งเมสเสจหากรูปแบบไม่ถูกต้องก็จะแสดงข้อความผิดพลาดขึ้น

2.1.2 โพรโตคอลประยุกต์ที่ใช้ในระบบจดหมายอิเล็กทรอนิกส์

แบ่งออกได้เป็น 2 ประเภทคือ

1. โพรโตคอลที่ใช้ในการรับจดหมายอิเล็กทรอนิกส์ ได้แก่
 - ป๊อป 3 (POP3 - Post Office Protocol Version 3)
 - ไอแมป์ (IMAP)
2. โพรโตคอลที่ใช้ในการส่งจดหมายอิเล็กทรอนิกส์ ได้แก่
 - เอสเอ็มทีพี (SMTP)

เราลองมาทำความรู้จักกับโพรโตคอลชนิดต่างกันดีกว่า

2.1.2.1 POP3 (Post Office Protocol)

ป๊อป 3 เป็นโพรโตคอลที่ใช้สำหรับให้เวิร์กสเตชันสามารถทำการค้นคืนเมสเสจ(Retrieve Message) จากเซิร์ฟเวอร์ ได้ซึ่งจะมีการทำงานแบบออฟไลน์โดยใช้ TCP พอร์ต 110

โพรโตคอลป๊อป 3 จะทำงานในแบบไคลเอนต์เซิร์ฟเวอร์ คือมีโปรแกรมป๊อปเซิร์ฟเวอร์ในเมลล์เซิร์ฟเวอร์ และ ป๊อปไคลเอนต์ในเครื่องผู้รับ ซึ่งปกติจะฝังอยู่ในโปรแกรมที่เป็นยูเซอร์เอเจนต์เลย โปรแกรมทั้งสองจะติดต่อกันโดยใช้ชุดคำสั่งที่เป็นรหัสแอสกี (ASCII) คือเมื่อด้านที่รับทำคำสั่งก็จะทำงานตามคำสั่งนั้น แล้วตอบกลับมาโดยมีค่าเป็น +OK หมายถึงทำงานได้เรียบร้อย หรือ -ERR หมายถึงเกิดปัญหาขึ้นทำงานไม่ได้ ซึ่งใน คำสั่งที่ต้องมีการตอบกลับและส่งข้อมูลกลับมา โดยประกอบด้วยข้อมูลหลาย ๆ บรรทัดนั้น POP3 จะให้บรรทัดสุดท้ายเป็นเครื่องหมายจุด (.) ตามด้วย Carriage Return และ Line Feed หมายถึงการสิ้นสุดชุดข้อมูล แต่ในกรณีที่ข้อมูลบรรทัดสุดท้ายมีข้อมูลที่เป็นจุดด้วย จะใช้เทคนิคที่เรียกว่า คาแร็กเตอร์สตัฟฟ์ (Character Stuffing) เพื่อแก้ปัญหา โดยจะเติมจุดลงไปอีกหนึ่งตัว เพื่อเป็นตัวบ่งชี้ว่าข้อมูลนั้นเป็นจุด ซึ่งจะแตกต่างจากสัญลักษณ์แสดงการสิ้นสุดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของ ป็อบ 3 ประกอบด้วย 3 สถานะคือ สถานะขออนุมัติ สถานะรับส่งรายการ และสถานะปรับปรุงข้อมูล ซึ่งในแต่ละสถานะจะรับรู้คำสั่งต่าง ๆ ของโพรโตคอลแตกต่างกัน โดยมีรายละเอียดต่างกัันดังนี้

- สถานะขออนุมัติ (Authorization State) เมื่อเริ่มต้นติดต่อกับเซิร์ฟเวอร์จะเป็นการเข้าสู่สถานะการขออนุมัติ โดยไคลเอนต์จะต้องแจ้งชื่อผู้ใช้และรหัสผ่าน เพื่อขออนุมัติจากเซิร์ฟเวอร์ก่อน โดยไคลเอนต์จะใช้คำสั่ง USER เพื่อระบุชื่อผู้ใช้ หรือ คำสั่ง PASS เพื่อกำหนด Password แต่ในกรณีที่ชื่อและ Password ถูกเข้ารหัสไว้ และไม่ได้เป็นค่า ASCII ทั่วไป ไคลเอนต์จะใช้คำสั่ง APOP ทำงานแทนคำสั่ง USER และ PASS

- สถานะรับส่งรายการ (Transaction State) หลังจากที่ได้รับอนุมัติจากเซิร์ฟเวอร์แล้ว ก็จะเข้าสู่สถานะที่ใช้คำสั่งในการทำงานต่าง ๆ

- สถานะปรับปรุงข้อมูล (Update State) เมื่อ User Agent เลิกใช้งานด้วยคำสั่ง QUIT ของ POP3 เซิร์ฟเวอร์ก็จะเข้าสู่สถานะปรับปรุงข้อมูล เพื่อลบอีเมลที่ควาน์โหลเตรียมเรียบร้อยแล้วออกไป จากนั้นก็จะเข้าสู่สถานะขออนุมัติใหม่โดยอัตโนมัติเพื่อรอรับการทำงานครั้งต่อไป
คำสั่งใน ป็อบ 3

USER	แจ้งชื่อผู้ใช้และระบุมेलบ็อกซ์ที่จะใช้
PASS	เป็นคำสั่งที่ใช้ระบุรหัสผ่านโดยจะใช้ต่อจากคำสั่ง USER
APOP	ทำหน้าที่เหมือนคำสั่ง USER และ PASS รวมกัน แต่ข้อมูลจะถูกเข้ารหัสก่อนส่งไป
STAT	เป็นคำสั่งตรวจสอบสภาพเซิร์ฟเวอร์
UIDL	ใช้ตรวจสอบหมายเลขประจำของอีเมล
LIST	ใช้ตรวจสอบหมายเลขของอีเมลและขนาดอีเมล
RETR	เป็นคำสั่งที่ใช้ส่งข้อมูลของอีเมล
DELE	เป็นการระบุเครื่องหมายการลบลงในอีเมลที่จะลบและอีเมลเหล่านั้นจะถูกลบออกจากเมลบ็อกซ์เมื่อสิ้นสุดการทำงาน
RSET	คำสั่งนี้เป็นการยกเลิกเครื่องหมายการลบอีเมล ที่เคยกำหนดไว้ด้วยคำสั่ง DELE ออกไป
ทุก ๆ อีเมล	
TOP	เซิร์ฟเวอร์จะส่งข้อมูลย้อนกลับไปเท่ากับจำนวนบรรทัดที่ระบุไว้
NOOP	เป็นคำสั่ง No Operation
QUIT	เป็นคำสั่งใช้เมื่อจบการทำงานหามีอีเมลซึ่งทำเครื่องหมายว่าจะลบไว้ อีเมลเหล่านั้นจะถูกลบจากเมลบ็อกซ์ในขั้นตอนนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 SMTP (Simple Mail Transfer Protocol)

เป็นโพรโทคอลที่ใช้ส่งอีเมลล์จาก UA ของผู้ส่งไปยัง MTA ของผู้ส่ง และส่งต่อไปยัง MTA เครื่องอื่น ๆ ที่เป็นจุดผ่านในการเชื่อมต่อไปยังเครื่องของผู้รับ ซึ่งจะทำงานร่วมกับโพรโทคอล 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 HELO พร้อมรายละเอียดด้านผู้ส่งออกไป จากนั้นจะส่งคำสั่ง MAIL เพื่อแจ้งให้เซิร์ฟเวอร์เตรียมรับอีเมลล์ ในส่วนของเซิร์ฟเวอร์เมื่อพร้อมที่จะรับอีเมลล์ก็จะตอบรับกลับมาด้วยคำสั่ง OK จากนั้นที่ด้านส่งก็จะเริ่มส่งโดยใช้คำสั่ง RCPT เพื่อกำหนดอีเมลล์แต่ละฉบับที่จะส่งไป ซึ่งการส่งข้อมูลของอีเมลล์จะถูกระบุด้วยคำสั่ง DATA

การส่งอีเมลล์ของโพรโทคอล SMTP ได้จัดเตรียมคำสั่งอื่น ๆ ไว้เพื่ออำนวยความสะดวกและคล่องตัวในการทำงาน ซึ่งประกอบด้วยคำสั่ง VRFY เพื่อให้ด้านที่ส่งได้ตรวจสอบรายชื่อว่ามีสิทธิใช้งานเมลล์บ็อกซ์นั้นหรือไม่ คำสั่ง EXPN ใช้จัดการและตรวจสอบรายชื่อจากลิสต์รายชื่อ และคำสั่ง TURN ใช้สลับให้ไกลเอนด์ของผู้ส่งทำหน้าที่รับข้อมูลจากเซิร์ฟเวอร์แทน

เมื่อได้รับคำสั่งต่าง ๆ ของผู้ส่งแล้ว เซิร์ฟเวอร์จะมีหน้าที่ตรวจสอบความถูกต้องของคำสั่ง จากนั้นจึงทำงานตามคำสั่งและส่งผลตอบกลับมา ส่วนลักษณะของข้อมูลที่ตอบกลับ (Reply Message) นั้นจะเป็นข้อมูลที่อยู่ในรูปของ text ที่เป็น ASCII โดยจะประกอบด้วยตัวเลขนำหน้าข้อความ 3 หลัก ทำหน้าที่แสดงสถานะการทำงานของเซิร์ฟเวอร์ และเปลี่ยนสถานะการทำงานของโพรโทคอล SMTP ถัดจากตัวเลขจะคั่นด้วยช่องว่างแล้วตามด้วยข้อความ ซึ่งปิดท้ายด้วยเครื่องหมาย Carriage Return และ Line Feed ตัวอย่างเช่น 500 Syntax error, command unrecognized หมายถึงคำสั่งที่ส่งไปไม่ถูกต้อง หรือ 503 Bad sequence of commands หมายถึง ลำดับการส่งคำสั่งไม่ถูกต้อง เหล่านี้เป็นต้น

ในการส่งอีเมลล์ของโพรโทคอล SMTP นั้น จะใช้วิธีอ้างถึงเซิร์ฟเวอร์อื่น ๆ ตามแบบ DNS หรือ Domain Name System เช่นเดียวกับระบบอื่น ๆ ในอินเทอร์เน็ต และยังสามารถส่งอีเมลล์ไปยังผู้รับคนเดียวหรือหลาย ๆ คนพร้อมกันก็ได้

The SMTP procedure

ในการส่งเมลล์จะมีขั้นตอนหลักอยู่ 3 ขั้นตอนในการทำ Transaction คือ เริ่มต้นด้วย คำสั่ง MAIL เพื่อระบุถึงตัวผู้ส่ง ตามด้วยคำสั่ง RCPT ซึ่งจะระบุถึงข้อมูลของผู้รับ และสุดท้าย คำสั่ง DATA ใช้สำหรับข้อมูลที่ส่งไปทาง E-mail

ขั้นตอนที่ 1

MAIL<SP>FROM:<reverse-path><CRLF>

คำสั่งนี้จะบอก SMTP-receiver ว่ากำลังมี new mail transaction กำลังเริ่มต้นและให้ทำการ reset state tables และ buffer ทั้งหมดรวมทั้งผู้รับและ mail data , reverse-path ถูกส่งไปเพื่อใช้ในการ report error และถ้ามีการ accept ตัว receiver-SMTP จะส่งค่า 250-OK กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2

RCPT<SP>TO:<forward-path><CRLF>

เป็นคำสั่งที่ใช้ระบุผู้รับ ถ้า Receiver-SMTP accept จะมีการส่งค่า 250 OK ตอบกลับมาและทำการเก็บค่า forward-path ไว้ แต่หากผู้รับ unknown ,Receiver-SMTP จะส่งค่า 550 Failure กลับมา

ขั้นตอนที่ 3

DATA<CRLF>

ถ้า accept , Receiver-SMTP จะส่งค่า 354 Intermediate กลับมาและจะทำการพิจารณาว่า ข้อความที่ตามมาเป็น message text เมื่อสิ้นสุด text แล้ว จะทำการรับและเก็บไว้ หลังจากนั้น SMTP-receiver จะส่งค่า 250 OK กลับมา

ตัวอย่างการส่ง Mail

S : MAIL FROM:<s0010526@ce.kmitl.ac.th>
 R : 250 2.1.0 <s0010526@ce.kmitl.ac.th>... Sender ok
 S : RCPT TO:<slatun@hotmail.com>
 R : 250 2.1.5 <slatun@hotmail.com>... Recipient ok
 S : DATA
 R : 354 Enter mail, end with "." on a line by itself
 S : Hello Slatun.
 S : .
 R : 250 2.0.0 e9A41R509415 Message accepted for delivery

2.1.2.3 ไอแมป (IMAP)

เป็น โพรโตคอล ที่ สามารถส่งเมล มาให้เราทำงานในขณะออฟไลน์ ได้ โดยใน ออนไลน์ โหมด เมล จะถูกส่งไปที่ เซิร์ฟเวอร์ แต่ เมล โคลเอ็นต์ ขณะส่งไปที่ เซิร์ฟเวอร์ จะไม่ได้ทำการลบ เมล ที่อยู่บนเครื่อง เซิร์ฟเวอร์ เมล ที่รับมา ขณะที่ โคลเอ็นต์ สามารถที่จะถาม เซิร์ฟเวอร์ ตรงส่วน ส่วนค้น หรือ ส่วนเนื้อหา ใน ข้อความ ที่เฉพาะเจาะจงได้ หรือทำการ ค้นหา ข้อความ จาก ข้อความ ทั้งหมดที่อยู่ใน เมลล์บ็อกซ์ ที่ เมลล์ เซิร์ฟเวอร์ ก็ได้ ข้อความ ที่อยู่ใน เมลล์ เซิร์ฟเวอร์ จะถูกทำเครื่องหมายด้วยสถานะ ต่างๆ ยกตัวอย่างเช่น "delete" หรือ "answer" ซึ่งสถานะนี้จะอยู่ต่อไปจนกว่า ผู้ใช้ จะทำการสั่ง อธิบายย่อๆได้ว่า ไอแมป ถูกออกแบบให้จัดการกับ เมลล์บ็อกซ์ระยะไกล ซึ่ง ผู้ใช้ สามารถที่จะเก็บ ข้อความ ไปไว้ที่ เครื่อง โคลเอ็นต์ หรือ เก็บไปที่ เซิร์ฟเวอร์ หรือ ไว้ทั้ง 2 ที่เลยก็ได้

โดยทั่วไป เซิร์ฟเวอร์ ที่ใช้ ไอแมป จะใช้ พอร์ต 143 ในการติดต่อ

2.1.2.3.1 คำสั่งและค่าตอบรับในการร้องขอ

การติดต่อ ไอแมป ประกอบด้วยการร้องขอจาก โคลเอ็นต์ เมื่อ เซิร์ฟเวอร์ ตอบรับ โคลเอ็นต์ ก็จะส่ง command ในรูปของตัวอักษรตัวใหญ่ สั้นๆ หรืออาจอยู่ในรูปของ เลขฐาน 16 ก็ได้ ตามด้วย ตัวแปรที่จะส่งไปให้ ฟังก์ชัน เซิร์ฟเวอร์ จบด้วย CRLF เพื่อบอกว่าจบคำสั่งที่ส่งไป เมื่อ เซิร์ฟเวอร์ ได้รับก็จะตอบกลับด้วยคำสั่งเดียวกัน

2.1.2.3.2 โพรโทคอล ที่ใช้ส่งฝั่ง โคลเอ็นต์ และ โพรโทคอล ที่ใช้รับฝั่ง เซิร์ฟเวอร์

คำสั่ง โคลเอ็นต์ จะขึ้นต้นด้วยการปฏิบัติการ แต่ละคำสั่งของ โคลเอ็นต์ จะขึ้นต้นด้วย ตัวบ่งบอกแต่ละคำสั่ง (ยกตัวอย่างเช่น ตัวอักษรตัวใหญ่สั้นๆ) ซึ่งเรียกว่า แท็ก (tag) ที่แตกต่างกันก็จะใช้แทนคำสั่งที่แตกต่างกัน

โพรโทคอล ที่ เซิร์ฟเวอร์ ใช้รับ คำสั่งที่ โคลเอ็นต์ ส่งมาจะอ่าน ค่าที่ละบรรทัดที่ โคลเอ็นต์ ส่งมา และส่งข้อมูลและ คำสั่งของ เซิร์ฟเวอร์ ตอบกลับไป

2.1.2.3.3 โพรโทคอล ที่ เซิร์ฟเวอร์ ส่ง และ โคลเอ็นต์ รับ

ข้อมูลที่ เซิร์ฟเวอร์ ใช้ส่งให้ โคลเอ็นต์ จะเป็น การตอบสถานะ ผลลัพธ์ที่ เซิร์ฟเวอร์ ตอบกลับ ไปจะบอกว่า การร้องขอของ โคลเอ็นต์ นั้นประสบความสำเร็จ หรือ ล้มเหลว โดยใช้ แท็ก เดียวกับที่ โคลเอ็นต์ ส่งมาให้ เซิร์ฟเวอร์ ดังนั้นถ้าเกิด โคลเอ็นต์ ส่งคำสั่งมาที่เดี่ยวนอกกว่า 1 คำสั่ง แท็ก ที่ เซิร์ฟเวอร์ จะส่งตอบกลับไปที่ โคลเอ็นต์ ก็จะตอบเป็นคู่ไป กับที่ โคลเอ็นต์ ส่งมา โดย เซิร์ฟเวอร์สามารถตอบกลับมาได้ 3 กรณี OK คือ ประสบความสำเร็จ NO หมายถึง ล้มเหลว หรือ BAD บอณา โพรโทคอลผิดพลาด หรือ syntax ที่ส่งมาผิดพลาด

โคลเอ็นต์ ต้องเตรียมการตอบรับคำสั่งที่ เซิร์ฟเวอร์ ส่งกลับมา รวมทั้งข้อมูลที่เครื่อง เซิร์ฟเวอร์ ส่งมาอาจจะบันทึกเก็บข้อมูลไว้ที่เครื่อง โคลเอ็นต์ ด้วยก็ได้ ดังนั้น โคลเอ็นต์ จะทำการส่งขอข้อความที่เครื่อง เซิร์ฟเวอร์ ให้บริการ และรับข้อมูลที่เครื่อง เซิร์ฟเวอร์ ส่งและทำสำเนาไว้ที่เครื่อง โคลเอ็นต์

2.2 การพัฒนาโปรแกรมเชิงวัตถุ (Object Oriented Programming)

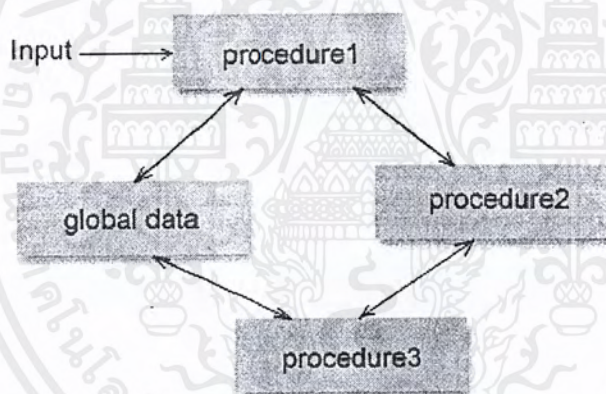
ในการพัฒนาซอฟต์แวร์นั้น แรกเริ่มเดิมที การพัฒนาโปรแกรมจะเริ่มจากการรับฟังปัญหา หรือ สิ่งที่ผู้ใช้ต้องการ (Requirements) จากนั้นนำมาวิเคราะห์ (Analysis) และ ออกแบบ (Design) จึงนำไปพัฒนาโปรแกรม (Development) และทดสอบโปรแกรม (Testing) ก่อนที่จะนำไปโปรแกรมนั้นไปให้ผู้ใช้ใช้งาน (Conversion & User Acceptance) และดูแลระบบต่อไป (Maintenance) ขั้นตอนดังกล่าวเราเรียกว่า Waterfall Model ซึ่งมีข้อดี คือมีการวางแผนโครงการ ต่างๆก่อนที่จะนำไปพัฒนาใช้งาน แต่ข้อเสียคือ ใช้เวลานานในการพัฒนาที่จะเห็นผลลัพธ์ และ ระบบจะขึ้น อยู่กับ critical path ถ้ามีการเปลี่ยนแปลง requirement จะต้องย้อนกลับไปยังสแต็ปเริ่มต้นใหม่ดังที่เรียกว่า Fountain Model และเมื่อมีการสร้างระบบใหม่ ก็จะเริ่มต้นตั้งแต่ขั้นตอนเริ่มต้นใหม่ ซึ่งจะทำให้เสียเวลาและทรัพยากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรืออีกวิธีการที่นิยมใช้กัน ได้แก่ Prototyping ซึ่งมีลักษณะพิเศษตรงที่ไม่จำเป็นต้องรู้ requirement ทั้งหมด ก็สามารถสร้างระบบจำลองขึ้นมาก่อนได้ ที่เรียกว่า prototype และเมื่อมีการปรับเปลี่ยน requirement ก็นำ prototype นั้น มาทำการแก้ไขและใช้งาน Prototype ที่ไม่ต้องนำมาแก้ไขเลย สามารถนำไปใช้งานได้ 'จริงเรียกว่า Evolutionary Prototyping แต่ข้อเสียจะเกิดขึ้นเมื่อ prototype นั้นไม่ตรงกับความต้องการ ไม่สามารถนำไปใช้งานได้ ทำให้เสียเวลาในการพัฒนามากขึ้น และไม่สามารถประเมินเวลาของโครงการที่แน่นอนได้

2.2.1 Methodologies

Structure Methodologies เป็นเทคนิคในเรื่องโครงสร้างข้อมูล ถูกคิดค้นตั้งแต่ปลายทศวรรษที่ 1960 แต่การออกแบบและวิเคราะห์ มาเริ่มในกลางและปลายทศวรรษที่ 1970 ซึ่งจะเน้นในเรื่องของการทำงานในรูปแบบของ Topdown และเน้นที่ functions ต่างๆในงานนั้นมากกว่าข้อมูล (data) ข้อดี คือ เหมาะสำหรับระบบที่ฟังก์ชันการทำงานสำคัญและซับซ้อนกว่าข้อมูล แต่ข้อเสีย คือ ในงานทั่วไป ฟังก์ชันมักจะเปลี่ยนแปลงเสมอ (unstable) และเมื่อมีการเปลี่ยนแปลง requirement ต้องทำงานหลายขั้นตอน เริ่มตั้งแต่การวิเคราะห์(system analysis) จากนั้นทำการ (design) แล้วจึงทำการเขียน โปรแกรม (programming)



รูปที่ 2-2 แสดงเทคนิคโครงสร้างข้อมูลในสมัยก่อน

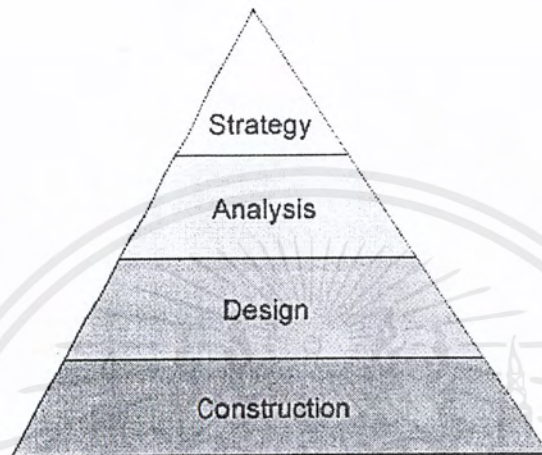
	potential for maintainability	potential for reusability
data	low	low
function	low	low

รูปที่ 2-3 แสดงการเปรียบเทียบข้อดีของเทคนิค Methodologies

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 Information Engineering

เป็นเทคนิคที่ริเริ่มใช้กันในปลายทศวรรษที่ 1980 โดยจะเป็น enterprise-wide data model โดยเน้นที่ข้อมูล (data) เป็นหลัก Application จะถูกสร้างตาม enterprise data ข้อดี คือ โครงสร้างระบบข้อมูลจะ stable แต่ข้อเสียคือ ปัญหาในการปรับเปลี่ยนหรือเพิ่มเติมโปรแกรมในภายหลังได้ยาก ระบบผูกติดกับ I-CASE tools



รูปที่ 2-4 แสดงเทคนิคแบบ Information Engineering

	potential for maintainability	potential for reusability
data	high	high
function	low	low

รูปที่ 2-5 แสดงการเปรียบเทียบระหว่างข้อดีของเทคนิค Information Engineering

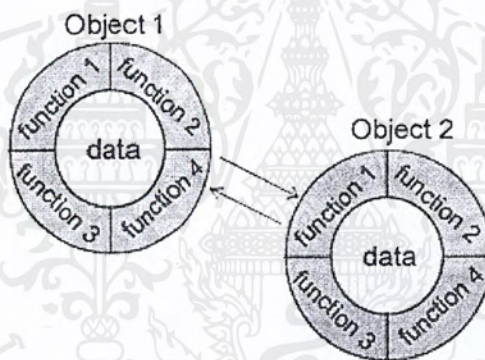
2.2.3 Object-oriented methodologies

โครงสร้างของระบบใช้ Objects ซึ่ง object จะทำการ encapsulate ทั้งข้อมูล (data) และ function และ object ทำการส่ง message ระหว่างกันได้ ข้อดี คือ ข้อมูลถูก encapsulate ซึ่งสามารถเข้าถึงข้อมูลได้โดยผ่าน ฟังก์ชัน (function) เท่านั้น และช่วยลดขั้นตอนการพัฒนา สามารถปรับเปลี่ยนหรือเพิ่มเติมโปรแกรมในภายหลังได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	potential for maintainability	potential for reusability
data	high	high
function	high	high

รูปที่ 2-6 แสดงการเปรียบเทียบระหว่างข้อดีของเทคนิค *Object Oriented Methodologies*

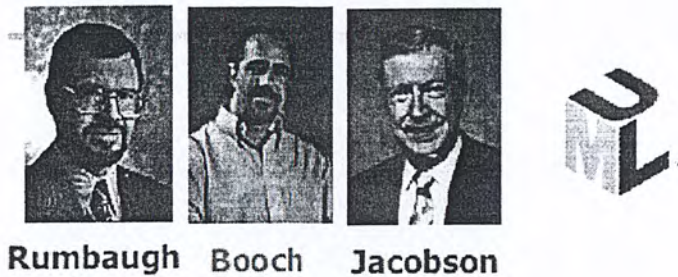


รูปที่ 2-7 แสดงการซ่อนรายละเอียดของข้อมูลในเทคนิค *Object Oriented Methodologies* โดยการพัฒนาโปรแกรมเชิงวัตถุได้มีภาษาที่ใช้อธิบายการออกแบบที่มีชื่อว่า UML

2.3 UML

UML หรือ Unified Modeling Language เป็นภาษาในการโมเดลมาตรฐาน UML เกิดจากการพัฒนาร่วมกันของผู้นำเทคโนโลยีทางด้านวัตถุ 3 ท่านคือ Grady Booch, Ivar Jacobson และ Jim Rumbaugh โดยก่อนที่จะมาเป็น UML นั้นผู้นำทั้ง 3 คนนี้ต่างก็มีโมเดลสำหรับการพัฒนาเชิงวัตถุเป็นของตัวเอง ต่อมาทางบริษัท Rational Software จึงได้ร่วมมือกับผู้นำทั้ง 3 คนนี้ในการพัฒนาโมเดลร่วมกัน เพื่อให้เกิดความเป็นหนึ่งเดียวสำหรับการพัฒนาระบบเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 แสดงรูปผู้พัฒนา UML และสัญลักษณ์ที่ใช้แทน UML

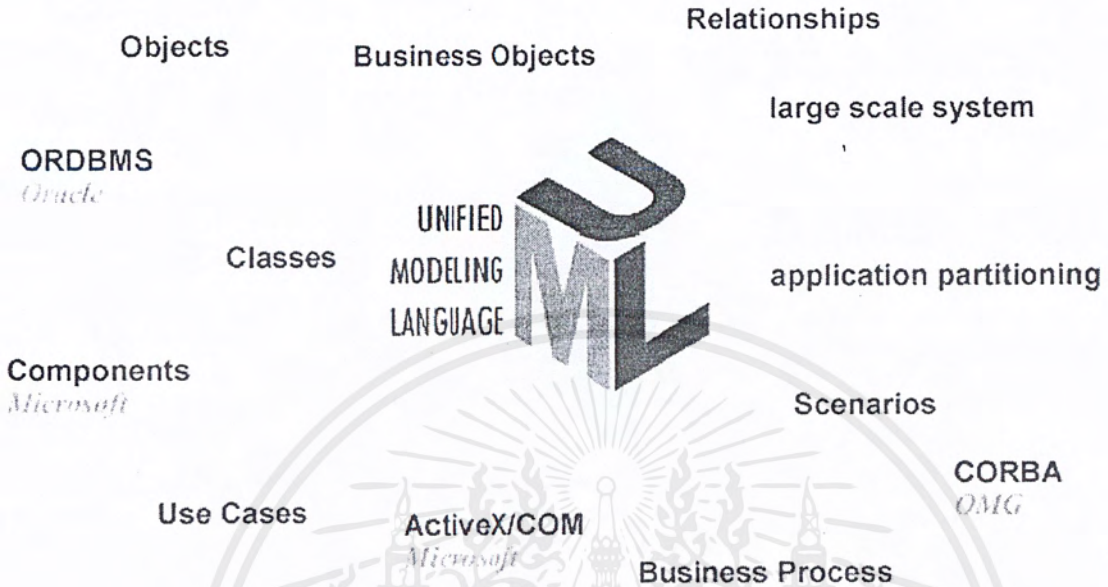
UML นั้นได้รวมแนวความคิดของวิธีการต่างๆ เข้าไว้ด้วยกัน จุดประสงค์ของ UML ก็คือ ต้องการสร้างโมเดลในการพัฒนาที่เข้าใจและสร้างได้ง่าย แต่สามารถนำไปใช้ได้กับทุกระบบ ตัว UML เองนั้นได้รับการสนับสนุนจากผู้นำในอุตสาหกรรมทางด้านคอมพิวเตอร์หลายๆ ราย เช่น HP, ICON Computing ,IBM , I-Logix, Intellicorp, MCI System house, Microsoft, ObjecTime, Oracle ,Platinum Technology Ptech, Reich Technologies, Softeam, Sterling Software, Taskon และ Unisys

	Analysis		Design	
	Technique	Deliverable	Technique	Deliverable
OMT (J.Rumbaugh) Back bone methodology	- Object Modeling - dynamic modeling	- data dictionary - object model - state transition diagrams	- System design - object design	- technical architecture - application architecture - refined object model - refined STDs
OOSE (I.Jacobson)	- use case analysis	- informal use cases - interaction diagrams		- detailed interaction diagrams
RDD (R.Wirfs-Brock)	- class / responsibilities / collaborators analysis	- CRC cards - collaborations graphs		- detailed CRC cards - detailed collab. graphs

รูปที่ 2-9 แสดงลักษณะของโมเดลที่มีการพัฒนาก่อนที่จะมาเป็น UML ของแต่ละคน

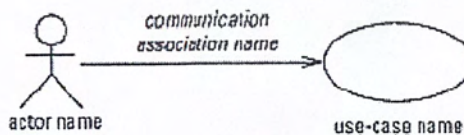
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน UML มีโมเดลที่สื่อสารด้วยภาพได้สำหรับระบบหลายๆ โมเดล โดยแต่ละโมเดลก็จะแสดงมุมมองต่อระบบที่ไม่เหมือนกัน ซึ่งประกอบไปด้วย



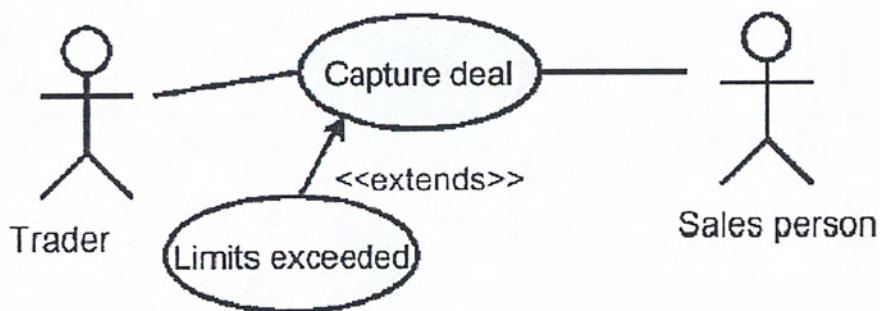
รูปที่ 2-10 แสดงส่วนต่างๆที่มีความเกี่ยวข้องกับภาษา UML

1. แผนภาพ Use-Case ที่ใช้แสดงการติดต่อระหว่างระบบกับผู้ใช้



รูปที่ 2-11 แสดงตัวอย่างแผนภาพ Use Case

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-11 แสดงสัญลักษณ์ที่มีใช้กับ Use Case Diagram

2. แผนภาพคลาส (Class Diagram) ที่ใช้แสดง โครงสร้างทางตรรกของระบบ

Visibility and properties

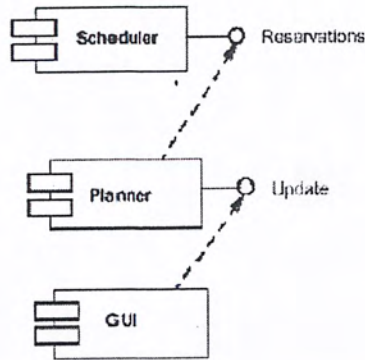
Class
- private attribute
protected attribute
/- private derived attribute
+\$class public attribute
+ public operation
protected operation
- private operation
+\$class public operation

Optional visibility icons

Attributes	Operations
public	public
protected	protected
private	private
implementation	implementation

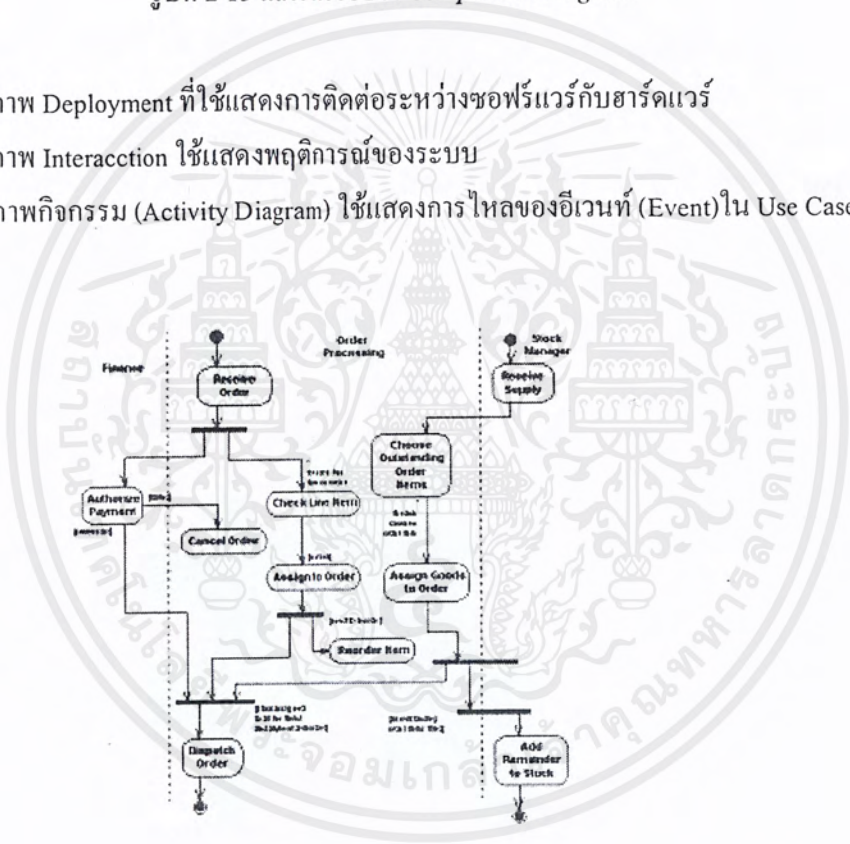
รูปที่ 2-12 แสดงสัญลักษณ์และความหมายต่างๆของ Class Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-15 แสดงตัวอย่าง Component Diagram

6. แผนภาพ Deployment ที่ใช้แสดงการติดต่อระหว่างซอฟต์แวร์กับฮาร์ดแวร์
7. แผนภาพ Interaction ใช้แสดงพฤติกรรมของระบบ
8. แผนภาพกิจกรรม (Activity Diagram) ใช้แสดงการไหลของอีเวนต์ (Event) ใน Use Case



รูปที่ 2-16 แสดงตัวอย่าง Component Diagram

2.3.1 แผนภาพและสัญลักษณ์ของ UML

UML นั้นประกอบไปด้วยสัญลักษณ์และเครื่องหมายจำนวนมาก โดยสัญลักษณ์และเครื่องหมายจำนวนมากของ UML เหล่านี้ ทำให้สามารถสร้างโมเดลแผนภาพได้อย่างยืดหยุ่น และครอบคลุมโดเมนของปัญหาทั้งหมด จะไม่ขอก้าวถึงสัญลักษณ์ทั้งหมดของ UML แต่จะกล่าวถึงสัญลักษณ์บางอย่างที่จำเป็นต้องใช้ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแต่งแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Text Note
- Constrain
- Stereotype

2.3.1.1 Text Note

เป็นส่วนหนึ่งของแผนภาพที่ไม่มีความหมายต่อโมเดลแต่อย่างไร โดยทั่วไปจะใช้ Text Note เพื่ออธิบายสิ่งที่อยู่ในแผนภาพบางสิ่งเพื่อให้ผู้ที่อ่านหรือใช้โคดอะแกรมเข้าใจความหมายได้ดีขึ้น โดยปกติ Text Note จะเป็นตัวอักษรที่อยู่บนขอบขององค์ประกอบ ในแผนภาพ

2.3.1.2 Constrain

เป็นเงื่อนไขที่เพิ่มเข้ามาสำหรับองค์ประกอบในแผนภาพ ตัวอย่างเช่น ในเงื่อนไขของอินพุทที่อยู่ในแผนภาพสถานะ โดยเงื่อนไขอินพุทนี้ จะเป็นข้อกำหนดที่ใช้แสดงว่าระบบจะต้องตอบสนองต่ออินพุทที่มีเข้ามาจากภายนอก โดยปกติ Constrain จะมีอยู่ในแยกหรือ Curly Braces และอาจจะปรากฏอยู่ใน Text Area ก็ได้

2.3.1.3 Stereotype

เป็น Metamodel ของ UML หรือจะกล่าวง่ายๆว่า Stereotype เป็น Entity ซึ่งเป็นส่วนขยายสำหรับ UML เพราะถึงแม้ว่าใน UML นั้นจะมีสัญลักษณ์และเครื่องหมายจำนวนมาก แต่ก็ยังไม่เพียงพอที่จะใช้แทนทุกๆสิ่งที่มีในโลกความเป็นจริง แต่สามารถใช้ Metamodel นี้ในการแทนความหมายของสิ่งเหล่านั้นตามที่ต้องการได้

อย่างไรก็ดี สัญลักษณ์และเครื่องหมายของ UML ทั้งหมดสามารถเขียนอยู่ในรูปของ Stereotype ได้เช่นกัน ตัวอย่างเช่น ในรูป สามารถเขียนแมสเชงที่มี ลักษณะแบบ Episodic-Waiting ใน UML ได้ตามรูป A และสามารถเขียนเป็น Stereotype ได้ในรูป B นอกจากนั้นยังสามารถสร้าง Metaclass โดยขยายข้อกำหนดหรือ เพิ่มความเฉพาะเจาะจงให้กับสัญลักษณ์หรือเครื่องหมายของ UML ที่มีอยู่ได้ ซึ่งคุณสมบัตินี้ก็เหมือนกันการสืบทอดคุณสมบัตินั่นเอง

สำหรับสัญลักษณ์ของ Stereotype นั้นจะเขียนชื่อ Entity ที่จะ เป็น Stereotype ภายในวงเล็บ <<>> เช่น "<<type>> tree " เป็นการกำหนด Stereotype ของชนิดข้อมูลเพื่อใช้ในการเขียนโปรแกรมเมื่อผู้โคดอะแกรมเห็นสัญลักษณ์ของ Stereotype นี้จะเข้าใจทันทีว่าข้อมูลนั้นมีชนิดเป็น tree ตัว UML ก็มีกำหนดสัญลักษณ์ของ Stereotype พื้นฐานไว้สำหรับแอปพลิเคชันเฉพาะด้านเช่นกัน

Entity ในโมเดลของ UML ทุกตัวสามารถเขียนเป็น Stereotype ได้ ตัวอย่างเช่น แมสเชง บางตัวอย่างของ Stereotype

2.3.2 รูปแบบของการออกแบบ (Design Pattern)

ดีไซน์แพทเทิร์น นั้นเป็นการนำรูปแบบหรือแนวความคิดในการออกแบบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับมาใช้ใหม่ นอกจากนั้นดีไซน์แพทเทิร์นยังช่วยปรับปรุงการทำเอกสารประกอบการออกแบบรวมทั้งช่วยในการดูแลระบบที่มีอยู่

สรุปง่าย ๆ ก็คือการใช้ดีไซน์แพทเทิร์นนั้นทำให้ผู้ออกแบบระบบสามารถออกแบบระบบที่ถูกต้องได้โดยใช้เวลาน้อยลง

โดยทั่วไปแพทเทิร์นจะประกอบด้วยส่วนหลัก 4 ส่วน คือ

□ ชื่อของแพทเทิร์น

ชื่อของแพทเทิร์นจะบ่งบอกถึงปัญหาของการออกแบบ ทางออกสำหรับปัญหาและอาจจะ เป็นคำที่บ่งบอกถึงประเด็นหลักของแพทเทิร์นนั้น

□ ปัญหา

ปัญหาเป็นส่วนที่อธิบายว่าปัญหาและสิ่งที่เกี่ยวข้องกับระบบ ในส่วนของปัญหาอาจจะ อธิบายปัญหาในการออกแบบเฉพาะตัวอย่างเช่นอัลกอริทึมที่ใช้ในวัตถุ หรืออาจจะอธิบายโครงสร้างของคลาสและวัตถุ

ในบางครั้งปัญหาอาจจะระบุเงื่อนไขของระบบที่จะนำแพทเทิร์นนั้นๆ ไปประยุกต์ใช้ด้วย

□ โขลุ่ยชัน

โขลุ่ยชันจะอธิบายส่วนประกอบที่ต้องมีในการออกแบบ ความสัมพันธ์ความรับผิดชอบและ การทำงานร่วมกัน ในส่วน โขลุ่ยชันจะไม่ได้อธิบายรูปแบบการออกแบบหรือการพัฒนา

โดยเฉพาะเจาะจงเนื่องจากว่าดีไซน์แพทเทิร์นนั้นสามารถนำไปประยุกต์ใช้กับระบบทั่วไป แต่ตัวดีไซน์แพทเทิร์นจะอธิบายถึงปัญหาในการออกแบบและวิธีการวางโครงสร้างการทำงาน ของคลาสและวัตถุเพื่อแก้ปัญหาเหล่านั้นๆ

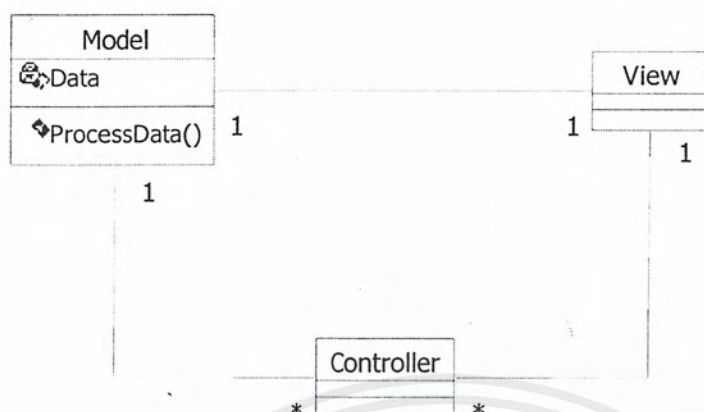
□ บทสรุป

บทสรุปจะอธิบายถึงข้อดีข้อเสียในการประยุกต์ใช้ดีไซน์แพทเทิร์น บางครั้งผู้ออกแบบ ระบบควรจะศึกษาบทสรุปของดีไซน์แพทเทิร์นก่อนที่จะนำดีไซน์แพทเทิร์นนั้นไปประยุกต์ใช้

ทั้งนี้เนื่องจากปัจจัยหลักในการออกแบบระบบด้วยวิธีเชิงวัตถุคือการนำกลับไปใช้ใหม่ ดังนั้นบทสรุปของดีไซน์แพทเทิร์นจะอธิบายความยืดหยุ่น ความสามารถในการขยายหรือความสามารถในการนำไปใช้กับระบบใหม่หลังจากที่ได้มีการประยุกต์ใช้ดีไซน์แพทเทิร์นเข้าไปแล้ว

2.3.2.1 ดีไซน์แพทเทิร์นที่มีการนำมาใช้ในระบบ

แพทเทิร์น Model-View-Controller



รูปที่ 2-17 โครงสร้างของดีไซน์แพทเทิร์น Model-View-Controller

จุดประสงค์

แพทเทิร์น Model-View-Controller เป็นแพทเทิร์นที่ช่วยในการแยกส่วนในการตอบสนองกับผู้ใช้ ส่วนแสดงผลและส่วนประมวลผลข้อมูลออกจากกัน ทั้งนี้เพื่อให้ผู้ออกแบบระบบสามารถนำส่วนประกอบในแพทเทิร์นกลับไปใช้ใหม่และยังทำให้ระบบมีความยืดหยุ่นต่อการเปลี่ยนแปลงในส่วนใดส่วนหนึ่งด้วย

การประยุกต์ใช้

ถ้าในระบบได้รับการกำหนดให้วัตถุหนึ่งทำหน้าที่ที่ทั้งติดต่อกับผู้ใช้ ประมวลผลข้อมูลและแสดงผลแล้วผู้ออกแบบระบบสามารถแยกวัตถุนั้นออกเป็น 3 วัตถุได้ที่ประกอบไปด้วย

- Model ซึ่งเป็นส่วนประมวลผลข้อมูล
- View ทำหน้าที่แสดงผลข้อมูล
- Controller ทำหน้าที่รับและประมวลผลอินพุตที่มีเข้ามาจากวัตถุ View

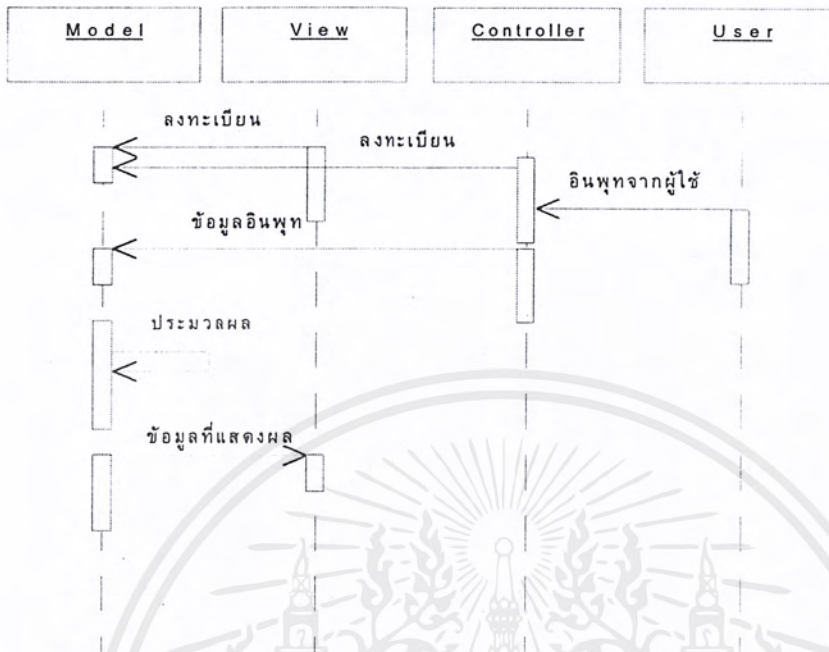
ผู้ออกแบบระบบสามารถประยุกต์ใช้ดีไซน์แพทเทิร์น Model-View-Controller นี้กับแพทเทิร์นอื่นได้ด้วย

บทบาทของแต่ละส่วนประกอบในดีไซน์แพทเทิร์น

- Model เป็นวัตถุที่เก็บและประมวลผลข้อมูลที่สำคัญในระบบ วัตถุ Model จะรับข้อมูลจากวัตถุ Controller และแสดงผลข้อมูลออกที่วัตถุ View โดยทั่วไปวัตถุ Model จะเป็นวัตถุหลักของระบบ
- View เป็นวัตถุที่รับข้อมูลจากวัตถุ Model เพื่อแสดงผล
- Controller เป็นวัตถุที่ทำหน้าที่ติดต่อหรือรับอินพุตจากผู้ใช้และส่งไปให้กับวัตถุ Model เพื่อประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างการทำงาน



รูปที่ 2-18 โครงสร้างการทำงานของดีไซน์แพทเทิร์น Model-View-Controller

1. วัตถุ View และ Controller ลงทะเบียนกับวัตถุ Model เพื่อให้วัตถุ Model ทราบว่ามีวัตถุใดที่ทำหน้าที่รับอินพุตและแสดงผลต่อผู้ใช้
2. เมื่อมีอินพุตมาจากผู้ใช้วัตถุ Controller จะทำหน้าที่ส่งข้อมูลนั้น ไปให้กับวัตถุ Model
3. วัตถุ Model เมื่อได้รับข้อมูลอินพุตแล้วก็จะประมวลผลซึ่งในกรณีนี้อาจจะใช้ข้อมูลประกอบการประมวลผลอื่นๆ ด้วย
4. วัตถุ Model ส่งข้อมูลที่ต้องการแสดงผลไปให้กับวัตถุ View เพื่อจัดการแสดงผลต่อผู้ใช้

ผลลัพธ์ที่ได้

ทำให้มีการแยกแยะส่วนประมวลผล ส่วนแสดงผลและส่วนรับอินพุตออกจากกันทำให้ระบบมีความยืดหยุ่น ในการปรับเปลี่ยนในส่วนใดส่วนหนึ่งเพราะจะไม่กระทบกับส่วนอื่นนอกจากนั้นยังสนับสนุนการนำกลับไปใช้ใหม่ของวัตถุในแพทเทิร์นด้วยตัวอย่างเช่นผู้ออกแบบสามารถนำวัตถุ View ไปใช้กับระบบอื่นได้ถ้าระบบใหม่นั้นมีส่วนแสดงผลเหมือนกัน

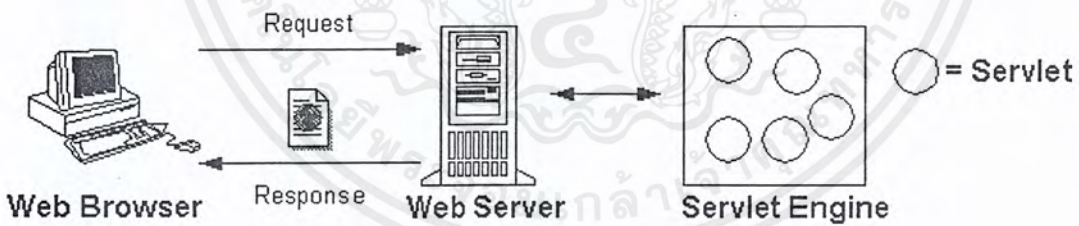
2.4 สถาปัตยกรรมต่างๆที่เกี่ยวข้องกับระบบ

2.4.1 Java Servlet

จาวามีสโตนแกนอันหนึ่งที่กล่าวไว้ว่า "สิ่งใดก็ตามที่เป็นที่นิยมใช้กันอย่างกว้างขวาง ท้ายที่สุดสิ่งนั้นจะกลายเป็นจาวา library" โดยผลพวงหนึ่งที่มาจากสโตนแกนนี้ก็คือเซิร์ฟเลตนั่นเอง

Servlet อ้างอิงหลักการของ CGI โดยข้อดีของ Servlet ที่อยู่เหนือ CGI อย่างแรกก็คือตัวภาษาที่ใช้เขียนซึ่งก็คือจาวานั้น จาวาเป็นภาษาที่ใช้คอนเซ็ปของ Object Oriented ในการเขียน หลายคนที่เกี่ยวข้องกับการเขียนโปรแกรมคงจะทราบดีว่า Object Oriented สามารถลดความซับซ้อนโครงสร้างของโปรแกรม รวมถึงอำนวยความสะดวกในการ reuse ส่วนประกอบต่าง ๆ ของโปรแกรมที่เขียนไว้แล้วเพียงใด นอกจากนี้จาวายังเป็นภาษาที่เป็น platform independent ซึ่งจะช่วยให้เราสามารถทำการพัฒนาระบบโดยใช้ environment อะไรก็ได้ซึ่งโดยทั่วไปมักจะเป็น Windows environment โดยจะนำโปรแกรมที่เขียนเสร็จแล้วมารันบน Unix environment เพื่อเพิ่มความเสถียรภาพของโปรแกรมอีกทีหนึ่ง นอกจากนี้ Servlet ยังมีความเร็วที่เหนือกว่า CGI เพราะ Servlet ใช้หลักการของ thread โดยจะทำการสร้าง 1 thread ต่อหนึ่ง request ที่มาจาก client ซึ่งในทางกลับกัน CGI จะทำการสร้าง 1 process ต่อหนึ่ง request ซึ่งจะทำให้เปลืองทรัพยากรมากกว่าและ process ในการรันก็จะช้ากว่าด้วย

ถึงแม้ว่า Servlet จะอ้างอิงหลักการของ CGI อย่างไรก็ตามในการที่จะทำการรัน Servlet แล้ว ตัว web server จะไม่สามารถส่งข้อมูลไปให้ Servlet ได้โดยตรงเหมือนกับหลักการของ CGI แต่ตัว web server จะต้องเพิ่มอีกส่วนหนึ่งซึ่งเป็นส่วนที่ใช้เป็นเสมือนตัวห่อหุ้ม Servlet ต่าง ๆ ไว้โดยส่วนที่เพิ่มขึ้นมานี้เราเรียกว่า Servlet Engine หรือ Servlet Container



รูปที่ 2-19 แสดง Servlet Engine และการทำงาน

โดยทั่วไป Servlet Engine จะเป็นส่วนที่มี Java Virtual Machine (JVM) อยู่ในตัวเองโดย Servlet Engine นี้จะมีหน้าที่รับ request จาก web server (ซึ่งมาจาก เว็บเบราว์เซอร์) แล้วทำการเลือกตัว Servlet ขึ้นมาทำการประมวลผล request นั้นภายใต้ JVM ของมัน โดยผลที่ได้จากการประมวลผลของ Servlet ที่ถูกเลือกจะถูกส่งกลับไปยัง web server โดย web server นี้จะส่งผลกลับไปยัง เว็บเบราว์เซอร์

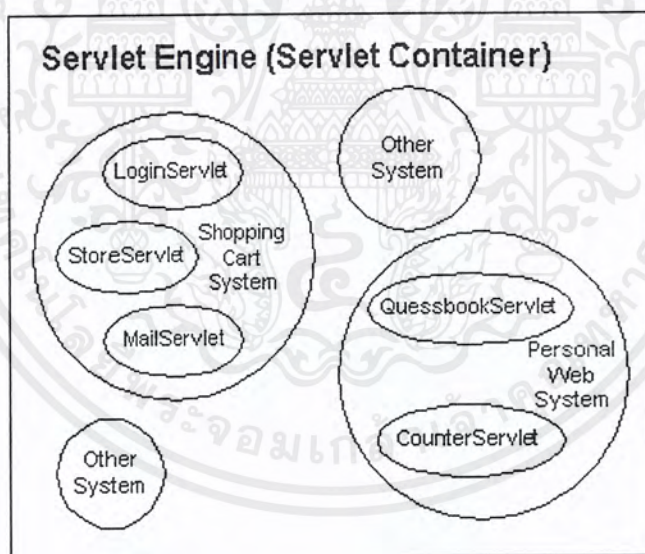
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในท้ายที่สุด ตัวอย่างของ Servlet Engine ที่ใช้กันอยู่ในปัจจุบันก็อาจจะเป็น Apache JServ, Apache Tomcat, Allaire JRun, IBM Websphere, BEA Weblogic, Servlet Exec เป็นต้น

Servlet Engine

ในการรันระบบ*ที่เขียนขึ้นโดยใช้หลักการของ Servlet เราจะต้องนำระบบดังกล่าวมาบรรจุอยู่ในสิ่ง ๆ หนึ่งที่เรียกว่า Servlet Engine ใหนักถึง Servlet Engine คล้าย ๆ กับกล่อง ๆ หนึ่งที่ใส่ลูกปิงปองไว้หลายลูก โดยลูกปิงปองแต่ละลูกก็คือระบบ ๆ หนึ่งนั่นเอง หลายคนอาจสงสัยทำไมถึงใช้คำว่าระบบ โดยทั่วไป Server Side Application หนึ่ง ๆ ที่ถูกเขียนขึ้นโดยใช้ Servlet API จะถูกเรียกว่า Servlet ในหนึ่งระบบอาจประกอบด้วย Servlet หลายอัน ยกตัวอย่างเช่น ระบบที่เกี่ยวข้องกับ Shopping Cart อาจประกอบด้วย Servlet ที่ทำหน้าที่ในการเช็คสต็อกอิน, Servlet ที่ทำหน้าที่ในการเก็บข้อมูลสินค้า, Servlet ที่ทำหน้าที่ในการส่งเมลกลับไปยังลูกค้าเพื่อบอกว่าได้ทำการส่งของไปให้แล้ว เป็นต้น ดังนั้นถ้ามองโดยรวมแล้ว Servlet Engine ก็คือที่รวมของระบบตั้งแต่หนึ่งระบบถึงหลายระบบ โดยแต่ละระบบจะประกอบด้วย Servlet หนึ่งอันหรือมากกว่า

* ระบบ ในที่นี้อาจจะหมายถึง Zone (Apache JServ) หรือ Web Application (Tomcat) ก็ได้



รูปที่ 2-20 รูปของระบบที่ประกอบไปด้วย Servlet Engine และระบบอื่นๆ

Servlet Engine เป็นเพียงกล่อง ๆ หนึ่งที่ใช้บรรจุและรันกลุ่มของ Servlet เท่านั้น ในการที่จะทำการติดต่อสื่อสารกับ Client ตัว Servlet Engine นี้จะต้องทำงานร่วมกับ Web Server ซึ่งเปรียบเสมือนฉากหน้าที่ติดต่อกับ Client อีกทีหนึ่ง เมื่อใดก็ตามที่มี request ส่งมาจาก Client ถ้า request นั้นจะเจงมาที่ตัว Servlet ทาง Web Server ก็จะทำการ forward ตัว request นั้นมาให้ Servlet Engine ซึ่งทาง Servlet Engine ก็จะทำการเรียก Servlet ที่ Client ต้องการขึ้นมาทำการประมวลผล request นั้น โดยท้ายสุด Servlet จะส่งเอกสารนี้เป็นเอกสารที่ส่งวนไวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลกลับไปให้ Servlet Engine, Servlet Engine ก็จะ forward ผลที่ได้กลับไปให้ Web Server ซึ่ง Web Server ก็จะส่งผลกลับไปให้ Client

Servlet Engine อาจจะเป็นส่วนที่ติดมากับ Web Server อยู่แล้วยกตัวอย่างเช่น Servlet Engine ที่อยู่ใน Netscape Enterprise Server, IBM WebSphere หรืออาจจะเป็นส่วนที่เป็น Add-on ให้กับ Web Server ก็ได้เช่น Apache Jserv, Tomcat, JRun หรือแม้กระทั่งเป็นส่วนหนึ่งที่อยู่ใน Web Application เช่น BEA Weblogic เป็นต้น ทั้งนี้การเลือกใช้ Servlet Engine แต่ละชนิดก็มักขึ้นอยู่กับปัจจัยหลายอย่างเช่น ความสะดวกในการรวมระบบที่จะสร้างขึ้นใหม่กับระบบที่มีอยู่แล้ว, งบประมาณที่มีอยู่สำหรับโครงการหรืออาจจะรวมไปถึงทักษะและประสบการณ์ส่วนตัวของนักพัฒนาแต่ละคน

2.4.2 JSP

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

JSP Containers

JSP Pages (ไฟล์ที่เขียนขึ้นโดยใช้ JSP script และลงท้ายด้วย .jsp) จะถูกรันโดย JSP Container ซึ่งมักจะเป็นส่วนประกอบที่อยู่ใน Webserver หรือ เป็นตัวแอดออนใน Application Server. โดยทั่วไป JSP Container จะเป็นตัวรับ request จาก client ส่งผ่านไปยัง JSP Page และส่งค่าที่ได้จากการประมวลผลโดย JSP Page กลับไปยัง client. JSP Container ที่ใช้กันอยู่มีมาจากหลายค่าย ยกตัวอย่างเช่น GNU JSP, Expresso, Tomcat Jakarta, Resin, Weblogic เป็นต้น

JSP Container ที่เราจะใช้กันคือ Jasper ซึ่งเป็น JSP Container ที่อยู่ใน Tomcat Servlet Engine จากค่าย apache (www.apache.org) ซึ่งทาง Sun ใช้เป็นตัวอ้างอิงในวงการ JSP โดยในอนาคต Tomcat จะเป็นตัวที่ใช้ในการรัน Servlet แทน Jserv ซึ่งใช้กันอยู่ในปัจจุบัน

การทำงานของ JSP

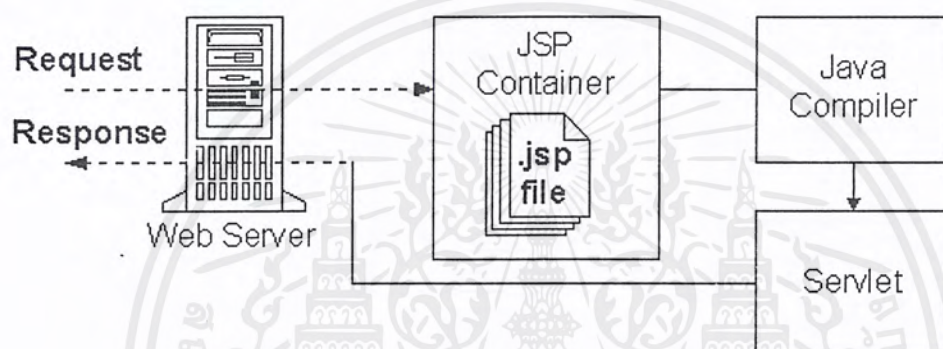
ถ้าดูเผิน ๆ จากทาง browser การเรียกไฟล์ JSP คงคล้าย ๆ กับการเรียกไฟล์ HTML ธรรมดา ๆ แต่ความจริงแล้ว หลังจาก request จาก client ส่งมาถึงเซิร์ฟเวอร์, JSP Container จะทำการแปลงไฟล์ JSP ให้กลายเป็น Servlet source ไฟล์ (ในกรณีที่ JSP ไฟล์ดังกล่าวถูกเรียกเป็นครั้งแรก) ซึ่งไฟล์ Servlet source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ได้จะถูกคอมไพล์เป็น .class เพื่อใช้ในการประมวลผล request ของ client แล้วส่งกลับไปให้เซิร์ฟเวอร์ในรูปของ outputStream ซึ่งถูกส่งไปที่ client ในท้ายสุด

หลังจากนั้น ถ้า JSP ไฟล์ดังกล่าวถูกเรียกอีก การแปลงไฟล์หรือคอมไพล์จะไม่เกิดขึ้น เพราะ JSP Container จะใช้ไฟล์ .class ที่เก็บไว้แล้วประมวลผลแทน ข้อสังเกตคือ การเรียกไฟล์ JSP ครั้งแรกจะรู้สึกช้า แต่ครั้งถัดไป ๆ จะเห็นได้ว่าเร็วขึ้นมากเพราะได้มีการลัดขั้นตอนต่าง ๆ ที่เสียเวลาไป

ใน JSP Specification 1.1 ยังมีการแนะนำให้ JSP Engine รองรับการโหลดไฟล์ JSP ขึ้นมาคอมไพล์ใหม่แบบอัตโนมัติ (Reloading) ซึ่งมักจะเกิดขึ้นในกรณีที่ JSP ไฟล์ได้มีการแก้ไข แล้วจะต้องมีการคอมไพล์ไฟล์ Servlet ที่กำลังใช้งานอยู่ใหม่



รูปที่ 2-21 แสดงการทำงานเริ่มต้นที่มีการอ้างอิงมาที่ JSP นี้

จากข้อความที่อธิบายข้างต้น เราสามารถแบ่งระยะเวลาของไฟล์ JSP หนึ่ง ๆ ออกเป็นสองช่วงคือ

- 1) Translation Time คือช่วงเวลาที่ JSP ไฟล์ถูกแปลงให้กลายเป็น Servlet ไฟล์และถูกคอมไพล์ให้กลายเป็น .class ไฟล์ ซึ่งจะเกิดก่อนการรับ request จาก client ครั้งแรก
- 2) Client Request Time คือช่วงเวลาที่ .class ของ JSP ไฟล์ ทำการรับ request จากแต่ละ client แล้วทำการประมวลผล

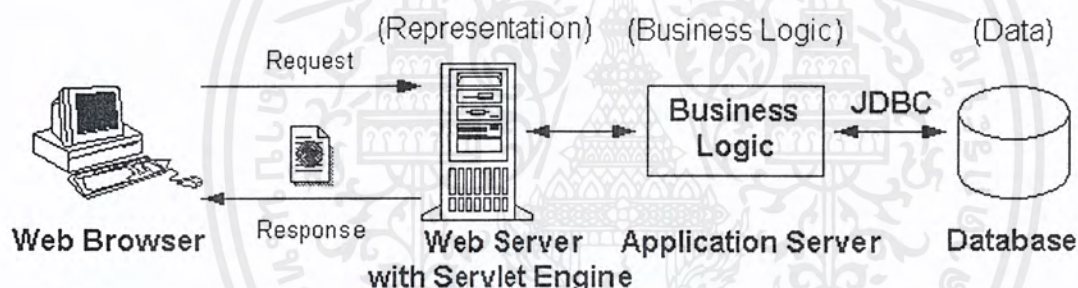
2.4.3 Application Server

สำหรับเว็บไซต์เล็ก ๆ แล้วโครงสร้างของ Server Side Application ที่กล่าวถึงข้างต้นก็มักจะเพียงพอต่อความต้องการสำหรับประโยชน์ใช้สอย อย่างไรก็ตามสำหรับเว็บไซต์ใหญ่ ๆ ที่มีผู้ใช้งานมหาศาล ส่วนประกอบที่สำคัญอีกอย่างหนึ่งที่มีจะถูกเพิ่มเข้าไปมักจะเป็นส่วนที่เรียกว่า Application Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Server เป็นส่วนประกอบอีกส่วนหนึ่งที่ถูกสร้างขึ้นมาเพื่อสนองความต้องการที่ว่า ในการสร้างระบบที่ซับซ้อนหนึ่งขึ้นมา เราควรจะแยกส่วนที่ทำหน้าที่เกี่ยวกับการบริการต่าง ๆ รวมไปถึงการจัดสรรข้อมูลออกมาเป็นอีกส่วนต่างหาก(Business Logic) ทั้งนี้เพื่อเพิ่มความคล่องตัวในการเปลี่ยนแปลงของระบบ รวมไปถึงการจัดระบบให้แบ่งออกเป็นส่วน ๆ อย่างชัดเจนมากยิ่งขึ้น โดยถ้าดูจากบทบาทของ Application Server ที่ผ่านตั้งแต่อดีตจนถึงปัจจุบันเราอาจจะพูดได้ว่า Application Server ก็คือส่วนที่ช่วยอำนวยความสะดวกให้กับ Server Side Application โดยมักจะเป็นตัวช่วยในการคำนวณข้อมูล (Services), ช่วยเป็นตัวกลางในการจัดส่งข้อมูล(Messaging Services), ช่วยในการควบคุมการจัดเก็บข้อมูลลงไปเ็นเคต้าเบสอย่างถูกต้อง(Transaction)

ตลอดจนช่วยในการเพิ่มความเร็วให้กับระบบโดยการเก็บออฟเจคต่าง ๆ ที่ใช้หรือไม่ใช้แล้วไว้เพื่อนำมาใช้อีก(Caching) เป็นต้น (ในแง่ของการออกแบบระบบ Application Server มักจะเป็นส่วนที่เราใช้ในส่วนที่เรียกว่า Services Layer ซึ่งในจำว่าแล้วส่วนนี้ก็มักจะเกี่ยวข้องกับ Enterprise Java ต่าง ๆ เช่น EJB, JTS, JMS, JNDI นั่นเอง)

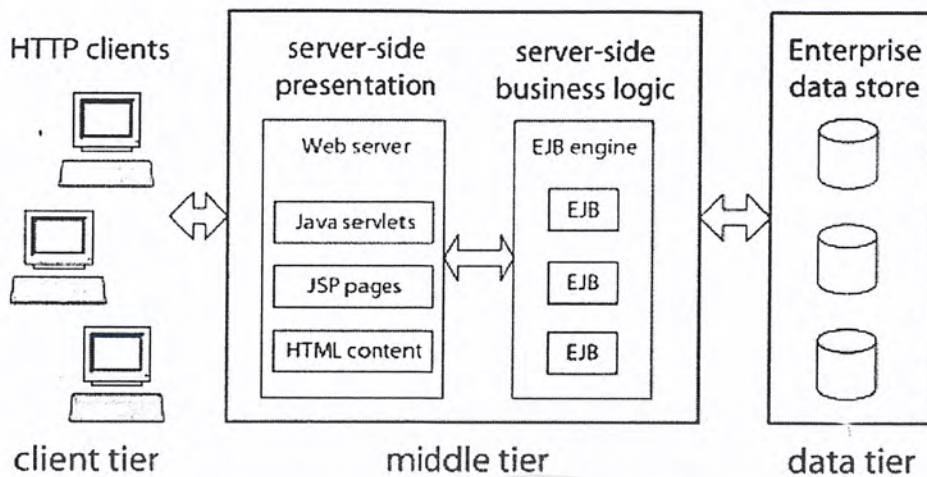


รูปที่ 2-22 การทำงานของ Application Server

ประโยชน์อย่างหนึ่งของ Application Server ที่เรามักพบเห็นโดยทั่วไปก็คือการนำมาใช้สำหรับรัน EJB (Enterprise Java Bean)

ถ้าพูดกันในเชิงนิยามแล้ว EJB ไม่ได้ถูกรันที่ Application Server แต่จะถูกรันกับสิ่งหนึ่งที่เรียกว่า EJB Server อย่างไรก็ตาม Application Server ส่วนมากมักจะสนับสนุน EJB ซึ่งเป็นส่วนหนึ่งของ J2EE ดังนั้น Application Server ส่วนมากจึงมีส่วนหนึ่งที่ทำหน้าที่เป็น EJB Server เพื่อรัน EJB ต่าง ๆ ได้. EJB แบ่งออกอย่างคร่าว ๆ เป็นสามแบบคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-23 การติดต่อกันระหว่าง Tier ต่างๆ

1. Session Bean

เป็น EJB ที่มักจะถูกใช้เป็นตัวเก็บรายละเอียดของผู้ใช้ที่กำลังทำกิจกรรมบางอย่างกับระบบหรืออาจจะถูกใช้เป็น Object Pool สำหรับ object ต่าง ๆ ที่ JSP หรือ Servlet สามารถเรียกใช้เพื่ออำนวยความสะดวกในการประมวลผลตลอดจนใช้ในการเปลี่ยนแปลงสถานะต่าง ๆ ของระบบ โดยเรามักเรียกส่วนนี้ว่า Services Layer นั้นเอง

2. Entity Bean

เป็น EJB ที่ใช้สำหรับเก็บเค้ต่าง ๆ ที่เราทำการแมปมาจากเค้ที่อยู่ในเค้เบส, ไฟล์, LDAP objects หรืออื่น ๆ ซึ่งจุดหลักของการนำเค้ต่าง ๆ มาใส่ไว้ใน Entity Bean ก็คือการเพิ่มความง่ายในการเปลี่ยนแปลงเค้โดยผ่านทาง EJB Object, การง่ายต่อการควบคุมการอ่านและเขียนเค้ต่าง ๆ ได้อย่างถูกต้องในกรณีที่มีระบบมีมากกว่า 1 thread เข้ามาเกี่ยวข้องในการอ่านเขียนหรือเปลี่ยนแปลงเค้ตัวเดียวกันในเวลาเดียวกัน รวมไปถึงความเร็วที่เพิ่มขึ้นในการเก็บเค้ต่าง ๆ ไว้ในหน่วยความจำ (Bean Caching) แทนที่จะต้องทำการอ่านเค้ใหม่จากเค้เบสทุกครั้งที่เราต้องการทำกิจกรรมบางอย่างกับเค้เหล่านั้น โดยข้อดีต่าง ๆ ที่กล่าวมาข้างต้นเหล่านี้จะเกิดขึ้นได้จากการทำงานร่วมกันระหว่าง Entity Bean, EJB Container และ EJB Server

อย่างไรก็ตามนักพัฒนาระบบก็ยังคงถกเถียงกันอยู่ถึง performance ที่เกิดขึ้นจากการใช้ Entity Bean เพื่อเก็บเค้ต่าง ๆ ไว้ โดยผลกระทบนี้มาจากการที่แต่ละ thread จะต้องแย่งกันเข้าไปใช้ Entity Bean ตัวเดียวกันเพื่อทำการเขียนอ่านหรือแปลงแปลงเค้ (Bean Contention) ซึ่งผลที่เกิดขึ้นอาจจะทำให้ระบบช้าลงกว่าการใช้ JDBC หรือ Object Database ธรรมดาทั่ว ๆ ไปก็ได้

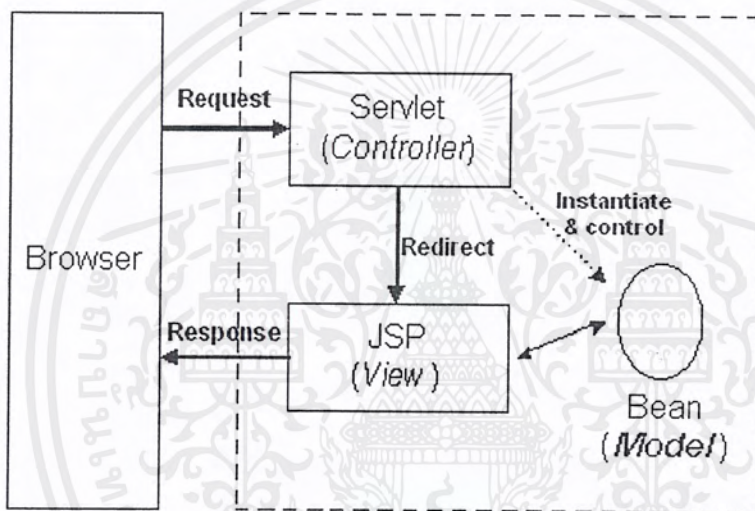
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Message-driven Bean (2.0 Proposed Final Draft)

เป็น EJB ชนิดใหม่ที่ถูกเพิ่มเข้ามาเพื่อทำการสนับสนุน JMS (Java Message Service) โดย EJB นี้จะถูกควบคุมได้ใช้ event ต่าง ๆ ที่เกิดขึ้นจาก JMS Message (สำหรับรายละเอียดเกี่ยวกับ EJB ผู้อ่านสามารถอ่านเพิ่มเติมได้จาก <http://java.sun.com/products/ejb/>)

2.5 การประยุกต์ใช้แพทเทิร์น Model-View-Controller กับระบบจดหมายอิเล็กทรอนิกส์

เนื่องจาก Model-View-Controller (MVC) ได้ถูกออกแบบให้มีแยกการทำงานของส่วนที่ประมวลผลทางโลจิก (logic) กับส่วนที่ใช้แสดงข้อมูลออกจากกัน พื้นฐานการทำงานจะเป็นดังรูป

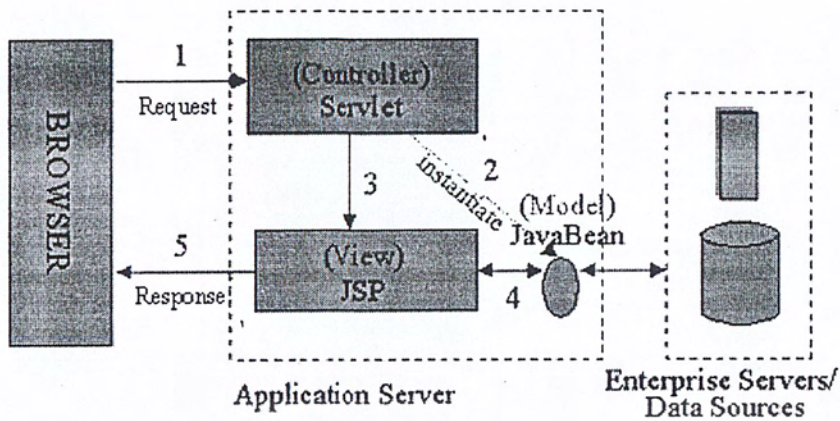


รูปที่ 2-24 MVC ที่นำมาประยุกต์กับ JSP

บราวเซอร์จะทำการส่งการร้องขอไปที่แก่ส่วน Controller ซึ่งในการทำงานจริงจะเป็นส่วนของ Servlet แล้วจะไปทำการเรียก Bean ที่เหมาะสมกับชนิดของการเรียกเพื่อคืนค่า View ที่ถูกต้องกลับไปยังบราวเซอร์ ซึ่ง Model ข้างต้นได้นำมาจาก สถาปัตยกรรม JSP Model 2 มาปรับปรุง ให้มีการทำงานตรงตามหน้าที่ของแต่ละงานจากกรณีใช้แต่ละกรณีไป

โดยเมื่อเริ่มต้นโหนดใช้งาน Servlet Engine จะทำการกำหนดค่าเริ่มต้นต่างๆ ที่ใช้ภายในโปรแกรม เข้าไปไว้ในหน่วยความจำ เพื่อนำมาใช้ภายหลังในส่วนของโปรแกรม เช่น พวกโปรโตคอลที่ใช้ภายในระบบ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-25 JSP Model 2 Architecture



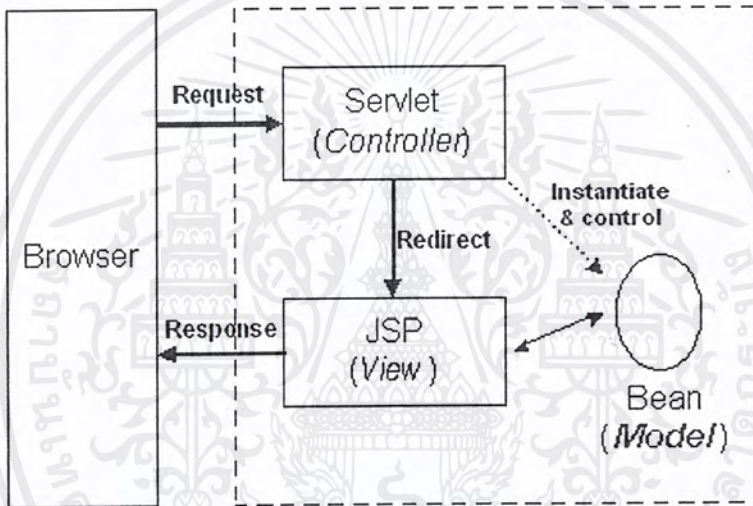
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้าง

3.1 การออกแบบ

ใน web-base e-mail จะเป็นการทำงานในลักษณะของโมเดล-วิว-คอนโทรลเลอร์ (Model-View-Controller) ซึ่งจะมีคอนโทรลเลอร์เป็นแกนกลางคอยจัดการทุกสิ่ง เช่น การรับข้อมูลที่มีการโพสต์ (post) จากผู้ใช้โดยจะไม่ได้เป็นผู้กำหนดเองแต่จะมีการกำหนดขึ้นมาก่อนอยู่แล้ว วิว(view) จะเป็นส่วนที่ใช้สำหรับติดต่อระหว่างผู้ใช้งาน ซึ่งก็คืออินเตอร์เฟซ (interface) และ โมเดล(mode) จะเป็นข้อมูลของผู้ใช้แต่ละคนซึ่งจะมีการสร้างใหม่ทุกครั้ง โดยจะเป็นการใช้จาวาบี๋น (Java Bean)



รูปที่ 3-1 MVC ที่นำมาใช้ใน Web-base E-mail

โดยในส่วนแรกของการออกแบบเรานำการทำงานของระบบนี้เมื่อมองจากผู้ใช้ภายนอกเข้ามาใช้ระบบ สามารถอธิบายได้โดยใช้ภาพจาก Use Case Diagram ซึ่งเราแยกการทำงานหลักของผู้ใช้ออกเป็น

การรับจดหมายอิเล็กทรอนิกส์ ซึ่งอาจจะเป็นการรับจดหมายที่มีการส่งเข้ามาใหม่ในระหว่างใช้โปรแกรมอยู่หรืออ่านจดหมายเก่าที่มีการจัดเก็บไว้ใน mailbox หรือ folder ต่างๆแล้วก็ได้

การส่งจดหมายอิเล็กทรอนิกส์ เพื่อส่งออกไปยังข้างนอกสถาบันหรือภายในสถาบันด้วยตนเอง

การ forward mail ที่มีอยู่แล้วหรือ mail ที่กำลังอ่านอยู่ไปยังคนอื่นต่ออีกที

การรับไฟล์ที่มีการส่งมากับจดหมายอิเล็กทรอนิกส์ แต่เนื่องจากว่าไฟล์เหล่านั้นอาจมีการคิดไวรัสคอมพิวเตอร์มาด้วยก็ได้ ดังนั้นผู้ใช้จะต้องทำการตรวจสอบได้จากภายในระบบโดยที่ไม่ต้องเสี่ยงที่จะตรวจสอบจากเครื่องของผู้ใช้เอง

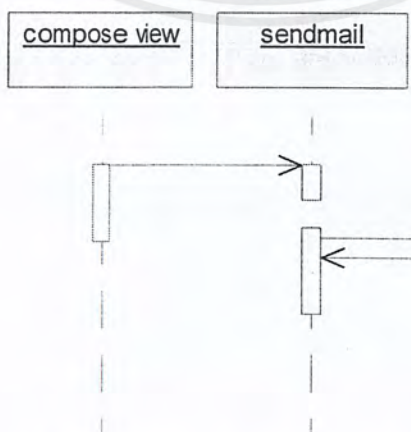
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างสมุดคณันท์ที่อยู่ของผู้รับปลายทางหรือเพื่อนของผู้ส่งเพื่อเป็นการช่วยให้ผู้ใช้จดจำ email ที่ติดต่อกับคนสนิทได้ง่าย และสามารถนำเอาสมุดคณันท์ที่มีการสร้างเอาไว้แล้วจากโปรแกรม PINE ในระบบ UNIX นำมาใช้ได้

โดยในถัดไปเราจะกล่าวถึงลำดับการทำงานของหน้าที่หลักระบบนี้ ใช้การอธิบายด้วย แผนภาพ ลำดับขั้นการทำงานดังนี้

การรับจดหมาย ในขั้นแรกเมื่อผู้ใช้กรอกรหัสผ่านและรหัสประจำตัวที่ต้องการเพื่อใช้ในการ พิสูจน์ตัวตนกับ mail server ผ่านทาง เว็บเบราว์เซอร์ จากนั้น เว็บเบราว์เซอร์ จะทำการส่งตัวแปรที่ผู้ใช้ กรอกมายังส่วนที่ทำงานฝั่ง server ส่วนที่ทำงานจะทำการ ส่งไปยังส่วนควบคุม โดยหน้าที่หลักของส่วน ควบคุมนี้จะทำหน้าที่ตรวจสอบตัวแปรและกระจายการทำงานจากชนิดของการร้องขอต่อไปยังส่วนที่ทำ การทำงานจริงโดยในที่นี้ ส่วนควบคุมจะทำการสร้าง session ขึ้นมาก่อน เพื่อที่จะเก็บตัวแปรต่างๆที่ ระบบใช้แสดงสถานะต่างๆของผู้ใช้ และตรวจสอบความถูกต้องของผู้ใช้ที่เคยทำการ login มาแล้วด้วย เมื่อทำการสร้างที่เก็บตัวแปรเสร็จแล้วส่วนควบคุมจะทำการ แสดงหน้าจอแสดง folder และ mailbox ออก มาให้ผู้ใช้เลือก และจะรอรับการส่งเหตุการณ์ต่างๆ จาก เว็บเบราว์เซอร์ ต่อไป แต่หากการสร้าง session ไม่ ถูกต้องอาจเนื่องมาจากว่า รหัสผ่านของผู้ใช้ไม่ถูกต้อง หรือ server ที่เก็บ mail ไม่สามารถใช้งานได้ขึ้นมา โปรแกรมจะทำการแสดงหน้าจอบอกผู้ใช้ถึงข้อผิดพลาดที่เกิดขึ้นจากการไม่สามารถเข้าไปรับmail ได้ เพื่อที่ผู้ใช้จะได้ทำการส่งรหัสผ่านออกมาใหม่หรือยกเลิกการตรวจสอบ

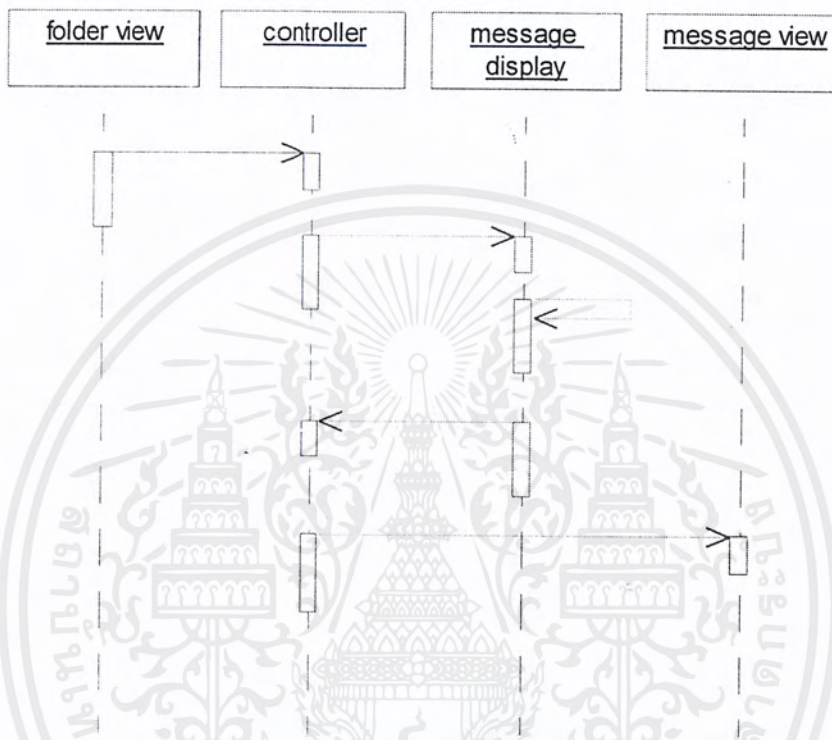
การส่งจดหมาย เมื่อผู้ใช้กรอกรายละเอียดของแต่ละอินพุทในฟอร์มที่จำเป็นต่อการส่งเรียบร้อยแล้ว เว็บเบราว์เซอร์ จะทำการส่งตัวแปรเหล่านั้นไปยัง ส่วนที่ทำการส่งจดหมาย โดยส่วนนี้จะทำการตรวจ สอบตัวแปรที่ส่งมาอีกทีว่าประกอบด้วยอะไรบ้างเพื่อสร้างรูปแบบของจดหมาย โดยข้อมูลที่จำเป็นขาด ไม่ได้จะประกอบไปด้วย ที่อยู่ผู้รับ , จำนวนของ และ เนื้อหา โดยอาจจะมีส่วนอื่นคิดมาด้วยเช่นการแนบ เอกสารต่างๆ ต่อท้ายจดหมาย โปรแกรมจะทำการรวมส่วนต่างๆไว้ด้วยกัน แล้วทำการเชื่อมต่อ SMTP server ที่ทำการส่ง mail เมื่อทำการส่งเรียบร้อยแล้วโปรแกรม จะแสดงหน้าจอเข้าไปใน folder mailbox อีกทีหนึ่ง



รูปที่ 3-2 Sequential diagram ในการส่งเมลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการอ่านจดหมายนั้นผู้ใช้จะต้องทำการเลือกจดหมาย ที่ถูกเก็บอยู่ใน folder หรือไม่ก็ใน mailbox ก่อน แสดงว่าผู้ใช้ต้องทำการเลือก mailbox ที่ต้องการก่อนแล้วจึงเข้าไปเลือกจดหมายที่ถูกเก็บไว้ใน mailbox นั้นๆ หลังจากเลือกจดหมายที่แสดงรายการอยู่ใน mailbox แล้ว เว็บเบราว์เซอร์ จะทำการส่งลำดับที่ของจดหมายและคำสั่งที่ปฏิบัติการกับจดหมายนั้น ไปยังส่วนควบคุม เพื่อให้ส่วนควบคุมกระจายการทำงานต่อไปยังส่วนที่รับผิดชอบในการเลือกจดหมายที่อยู่ใน mailbox และส่งไปยังส่วนที่ทำหน้าที่แสดงผล



รูปที่ 3-3 Sequential diagram การอ่านจดหมาย

การติดต่อกับสมุดบันทึกของ PINE โปรแกรมจะนำข้อมูลค่าต่างๆของ PINE มาเก็บไว้ตั้งแต่ตอนที่เข้าใช้ระบบตอนแรกที่มีการส่งรหัสผ่านเลย แล้วจะทำการเก็บลง Session ที่ถูกเก็บไว้ที่ฝั่ง server ซึ่งการแก้ไขจะมีกลับไปยังค่าของ PINE ก็ต่อเมื่อ ผู้ใช้ได้ทำการออกจากระบบด้วยวิธีที่ถูกต้องเท่านั้น เนื่องจากโปรแกรมจะคอยรับเหตุการณ์เฉพาะ ที่ผู้ใช้ทำการออกจากระบบและเชื่อมต่อกับ mail server เพื่อทำการบันทึกค่าคิดตั้งของ PINE โดยผ่านทาง FTP protocol ดังนั้นหากผู้ใช้ไม่ยอมให้ค่าที่ทำการเพิ่มหรือ ลบมีผลโดยตรงในการใช้งานครั้งต่อไป ก็ต้องทำการกดปุ่ม logout เท่านั้น

3.2 การสร้าง

โดยการออกแบบในระดับรายละเอียดของการทำงานต่างๆในระดับ Class นั้นจะขอล่าแบ่งออกเป็น 3 ส่วนได้แก่ ส่วนที่ทำหน้าที่เป็น Model , Controller และ View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วิว(View)

ในการออกแบบวิวจะเน้นการใช้งานเพื่อความสะดวกกับผู้ใช้จะมีวิวดังนี้

Addressbook	จะเป็นวิวที่มีหน้าที่แสดง,แก้ไข และเพิ่มเติม ข้อมูลสมุดที่อยู่ของผู้ใช้ซึ่งอาจมีการสร้างไว้ก่อนอยู่แล้วในระบบที่จะมีการนำเอาโปรแกรมนี้ไปใช้ คำศัพท์ที่ใช้จะมี fullname ชื่อ-สกุล nickname ชื่อเล่นที่ผู้ใช้สามารถจดจำได้ email ที่อยู่
Addressbook_entry	จะเป็นวิวที่ทำหน้าที่สำหรับให้ผู้ใช้ทำการแก้ไขข้อมูลภายในสมุดที่อยู่ของแต่ละคนเท่านั้น คำศัพท์ที่ใช้มี fullname ชื่อ-สกุล nickname ชื่อเล่น comment คอมเมนต์ fcc โฟลเดอร์คอลเลกชัน email ที่อยู่
Compose	เป็นวิวสำหรับการเขียนจดหมายอิเล็กทรอนิกส์เพื่อส่งให้ผู้รับตัวแปรที่ใช้คือ to ที่อยู่ของผู้รับ cc ที่อยู่ของผู้รับอื่น ๆ ที่ต้องการส่งไปเป็นฉบับสำเนา bcc ที่อยู่ของผู้รับอื่น ๆ ที่ต้องการส่งไปเป็นฉบับสำเนา แต่ไม่ต้องการให้เห็นว่าส่งไปให้ผู้ใดบ้าง attachment ไฟล์เอกสารที่ทำการแนบเพื่อส่งไปด้วย subject หัวข้อจดหมาย body ข้อความของจดหมาย
error	เป็นวิวสำหรับแสดงข้อความผิดพลาด คำศัพท์ที่ใช้คือ error เก็บตัวข้อความผิดพลาด
Folder	เป็นวิวที่แสดงโฟลเดอร์(folder) และตู้จดหมาย(mailbox)ของผู้ใช้ คำศัพท์ที่ใช้ folders เก็บข้อมูลของโฟลเดอร์ทั้งหมด mailboxes เก็บข้อมูลของตู้จดหมายทั้งหมด
Loggedout	เป็นวิวที่แสดงข้อความตอนผู้ใช้ออกจากระบบ
Login	เป็นวิวที่ใช้สำหรับให้ทำการเข้าสู่ระบบ คำศัพท์ที่ใช้ username รหัสผู้ใช้ password รหัสผ่านเข้าสู่ระบบ hostname ชื่อของโฮสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mailbox	เป็นวิวที่แสดงเมสเสจของตู้จดหมายว่าในแต่ละตู้จดหมายตัวแปรที่ใช้ messageno เลขที่ของเมสเสจ
Message	เป็นวิวที่แสดงข้อความของเมสเสจ
Preferences	เป็นวิวสำหรับแสดง และแก้ไขข้อมูลส่วนตัวของผู้ใช้
Printmessage	เป็นวิวเพื่อแสดงเมสเสจในอีกรูปแบบหนึ่งเพื่อให้ผู้ใช้สามารถทำการพิมพ์ข้อความออก ทางเครื่องพิมพ์เอกสารได้

การกระทำที่สามารถเกิดขึ้นได้นั้นจะมีตัวแปร 2 ตัวเพื่อบ่งบอกการกระทำ คือ

action บอกถึงว่าจะมีการกระทำเกี่ยวกับอะไร

todo บอกถึงว่าจะทำอะไร

การกระทำของ action จะแบ่งเป็น 5 กระทำดังนี้

Acton	คำอธิบาย
Session	เป็นการกระทำเกี่ยวกับ session
Folder	เป็นการกระทำที่เกี่ยวกับโฟลเดอร์(folder)
Mailbox	เป็นการกระทำที่เกี่ยวกับตู้จดหมาย(mailbox)
Message	เป็นการกระทำที่เกี่ยวกับข้อความ(message)
Preference	เป็นการกระทำที่เกี่ยวกับข้อมูลส่วนตัว
Addressbook	เป็นการกระทำที่เกี่ยวกับสมุดที่อยู่

ในแต่ละการกระทำของ action จะมีดังนี้

session

Todo	คำอธิบาย	ค่าพารามิเตอร์	ไปยัง
Login	ทำการเข้าสู่ระบบและ สร้าง session ของผู้ใช้	Username Password Hostname	folder(imap) mailbox(pop3)
Logout	ทำการออกจากระบบ		loggedout
Redirect	ทำการไปยังวิวต่าง ๆ	view(folder,mailbox,me ssage)	ขึ้นอยู่กับค่าที่ส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

folder

todo	คำอธิบาย	ค่าพารามิเตอร์	ไปยัง
Display	เตรียมข้อมูลของโฟลเดอร์เพื่อแสดง	Path	folder(imap)
Move	ย้ายโฟลเดอร์ตามที่ระบุ	Path Destenation	folder(imap)
Create	สร้างโฟลเดอร์ตามที่ระบุ	Aname	folder(imap)
Delete	ลบโฟลเดอร์ตามที่ระบุ	Path	folder(imap)

mailbox

Todo	คำอธิบาย	ค่าพารามิเตอร์	ไปยัง
display	เตรียมข้อมูลของข้อความจากผู้จดหมายเพื่อแสดง	Number	message
displaypart	แสดงส่วนของข้อความ(part)	Number	-
move	ย้ายข้อความไปยังผู้จดหมายที่ระบุ	Numbers Destination	mailbox
delete	ลบข้อความ(message)	Numbers	mailbox
compose	ทำการเตรียมโครงสร้างของข้อความเพื่อทำการส่ง	To Reply	compose

preferences

Todo	คำอธิบาย	ค่าพารามิเตอร์	ไปยัง
Update	ทำการอัปเดตข้อมูลของผู้ใช้	all data of user	preferences

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

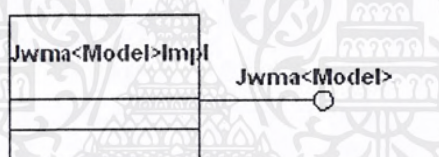
addressbook

Todo	คำอธิบาย	ค่าพารามิเตอร์	ไปยัง
Add	ทำการเพิ่มข้อมูลของสมุดที่อยู่	all data of addressbook	addressbook
Update	ทำการอัปเดตข้อมูลของสมุดที่อยู่	- nickname -all data of that nickname	addressbook
Remove	ทำการย้ายที่อยู่ตามที่ระบุ	Nickname	addressbook

3.4 โมเดล (Model)

อธิบายรายละเอียดภายใน package model นี้

model จะเป็นการเตรียมการทำงานต่างๆที่ส่วน controller ส่งมาทำงานแล้วทำการตอนผลลัพธ์ที่กลับไปยัง view ให้ถูกต้อง



รูปที่ 3-4 แสดง Class Diagram ของ class ใน model

โดย view จะทำหน้าที่เพียงแค่เรียกการทำงานผ่านทาง interface เหตุที่ต้องทำเช่นนี้เพื่อเป็นการลดความซับซ้อนทางด้านการทำงานใน view โดยให้หน้าที่หลักๆของ view ทำแค่การแสดงผลทางด้าน Graphic เท่านั้น

ในส่วนของ Model มี class และ interface ต่างๆแยกได้ดังต่อไปนี้

รายละเอียดของ interface ที่อยู่ภายใน Model

Address

เป็น interface ที่กำหนด method ต่างๆเกี่ยวกับการจัดการเกี่ยวกับที่อยู่

AddressBook

เป็น interface ที่กำหนด method ในการจัดการกับสมุดบันทึก

Error

เป็น interface ที่ใช้กำหนด attribute และ error ที่กำหนดไว้ในระบบนี้

Folder

เป็น interface กำหนด method ต่างๆที่ใช้จัดการกับโฟลเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HtmlHelper

เป็น interface ที่กำหนดส่วนปลีกย่อยในการติดต่อกับส่วนที่เป็น HTML

InboxInfo

เป็น interface ที่กำหนด method ต่างๆที่ใช้ในการเรียกดูว่าคุณสมบัติของผู้รับจดหมาย

Mailbox

เป็น interface ที่กำหนด method ต่างๆในการจัดการเกี่ยวกับการทำงานใน inbox

Message

เป็น interface ที่กำหนด method ต่างๆในการจัดการเกี่ยวกับ message ใน inbox

MessageInfo

เป็น interface ที่กำหนด method ต่างๆในการอ่านค่าคุณสมบัติต่างๆของ message (ในที่นี้ก็คือจดหมาย นั่นเอง)

MessagePart

เป็น interface ที่กำหนด method ต่างๆในการจัดการเกี่ยวกับส่วน message ใน inbox โดยจริงๆแล้ว message จะถูกแบ่งออกเป็นหลายๆโดยส่วนแรกอาจจะเป็นส่วนข้อความและส่วนหลังๆที่ตามมาอาจจะเป็นไฟล์ที่ผู้ส่งส่งมาให้ก็ได้

Preferences

เป็น interface ที่กำหนด method ต่างๆในการจัดการเกี่ยวกับค่าต่างๆที่ระบบใช้เช่น การเรียกหาค่า SMTP host มาเพื่อใช้ connect

StoreInfo

เป็น interface ที่กำหนด method ต่างๆในการจัดการเกี่ยวกับการเก็บข้อมูลของโฟลเดอร์ทั้งหมด

TrashInfo

เป็น interface ที่กำหนด method ต่างๆในการจัดการเกี่ยวกับถังขยะที่ใช้เก็บ mail ที่กำลังจะลบ

รายละเอียด class ทั้งหมดที่เกี่ยวกับส่วน model

ส่วนมากจะเป็นการ implement กับส่วน interface ข้างต้น

AddressBookImpl

เป็น class ที่ implement มาจาก AddressBook Interface

AddressImpl

เป็น class ที่ implement มาจาก Address Interface

ComposeMessage

เป็น class ที่ใช้ในการส่ง mail ออกไป

DisplayMessage

เป็น class ที่ใช้แสดงผล message ที่อยู่ใน folder

FolderImpl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น class ที่ implement มาจาก Folder Interface

HtmlHelperImpl

เป็น class ที่ implement มาจาก HtmlHelper Interface

InboxInfoImpl

เป็น class ที่ implement มาจาก InboxInfo Interface

MailboxImpl

เป็น class ที่ implement มาจาก Mailbox Interface

MessageInfoImpl

เป็น class ที่ implement มาจาก MessageInfo Interface

MessagePartImpl

เป็น class ที่ implement มาจาก MessagePart Interface

PreferencesImpl

เป็น class ที่ implement มาจาก Preferences Interface

StoreInfoImpl

เป็น class ที่ implement มาจาก StoreInfo Interface

TrashInfoImpl

เป็น class ที่ implement มาจาก TrashInfo Interface

Class ที่เกี่ยวกับการดักจับข้อผิดพลาดภายใน Model

Exception

เป็น class ที่ implement มาจาก Error Interface

package ต่างๆที่จำเป็นต่อการทำงานกับ Model ได้แก่

JavaMail 1.2 และ the Java Activation Framework(JAF) 1.0

ถัดมาจะพูดถึงส่วนอีก package หนึ่งที่สำคัญกับทุกส่วนก็คือ util

อธิบายการทำงานของ package util

ประกอบไปด้วย class ที่ทำหน้าที่อ่านค่าคุณสมบัติ และ class ที่ทำหน้าที่ช่วยในการ debug โดยการสร้าง log file รวมทั้ง class ที่ช่วยในการตรวจสอบข้อมูลที่ upload มาด้วย ใน multipart/form-data โดยรวมแล้วก็เป็น class ที่ช่วยให้การทำงานเล็กๆต่างๆภายในทั้ง Model และ View

รายละเอียดของ class ต่างๆภายใน util

FormdataMultipart

เป็น class ที่ extends มาจาก class Multipart เพื่อช่วยในการทำงานกับ content แบบ multipart/form-data

Log

เป็น class ที่ทำหน้าที่เก็บขั้นตอนการทำงานภายในไว้ในไฟล์เพื่อช่วยในการ debug

MimeBodyPartDataSource

เป็น Class ซึ่ง implements มาจาก MimeBodyPartDataSource.

MultipartInputStream

เป็น class ที่ห่อหุ้ม class InputStream ไว้อีกทีเพื่อใช้ในการทำงานกับ message

MultipartRequest

ใช้ทำงานกับส่วนที่เป็นการทำงานกับ multipart

PropertiesLoader

เป็น class ช่วยเหลือที่ทำหน้าที่อ่านค่าไฟล์ที่มีการตั้งค่าส่วนตัวไว้เข้ามาเพื่อใช้ในระบบ

package ต่างๆที่จำเป็นต่อ package นี้ได้แก่

JavaMail 1.2 และ the Java Activation Framework(JAF) 1.0

Class หลักๆของระบบ

Controller

ทำหน้าที่เหมือนนุรุษไปรษณีย์โดยเมื่อผู้ใช้ทำการส่งการร้องขอมาให้ผ่านทาง View ก็จะมีการตรวจสอบว่าตรงกับปลายทางในการทำงานกับ model อันไหนแล้วก็จะทำการส่งตัวแปรต่างๆ ไปยัง model เพื่อทำการประมวลค่าหรือแสดงผลอีกที

Kernel

เป็นตัวกลางที่ใช้จัดการกับ utility ภายในระบบเช่น การสร้าง log file , การเก็บค่าลง log file การอ่านค่าคอนฟิกมา เป็นต้น

Sendmail

ทำหน้าที่อ่านค่าจากตัวแปรที่ส่งมาจากคอนส่งจดหมายแล้วทำการแยกค่าตัวแปรแล้วทำการส่งเมล มีเพียงการส่งเมลเพียงอย่างเดียวเท่านั้นที่การทำงานไม่จำเป็นต้องให้ Controller เป็นตัวส่งไปยัง model ต่างๆทำงาน มันจะทำงานโดยเรียกตรงๆผ่านทาง class นี้เลย

Session เป็น class ที่ทำงานจัดการกับการพิสูจน์ตัวตน การตรวจสอบ username และ passwd กับทาง mail server แล้วทำการรับเมลมาแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองการทำงานและแสดงผล

4.1 การติดตั้งระบบเว็บเมล์

- อุปกรณ์

1. เครื่องคอมพิวเตอร์รุ่น Pentium III 733 , Ram 128 2 เครื่อง
2. เครื่องคอมพิวเตอร์รุ่น K6 III 500, Ram 64 1 เครื่อง
3. เครื่องคอมพิวเตอร์รุ่น Sun Ultra Sparc Ram 640 1 เครื่อง
4. ซอฟต์แวร์ผลิตภัณฑ์ Alliare Jrun 3.0
5. ระบบเครือข่ายที่เชื่อมต่อเครื่องคอมพิวเตอร์ทั้ง 3 เครื่องเข้าด้วยกัน

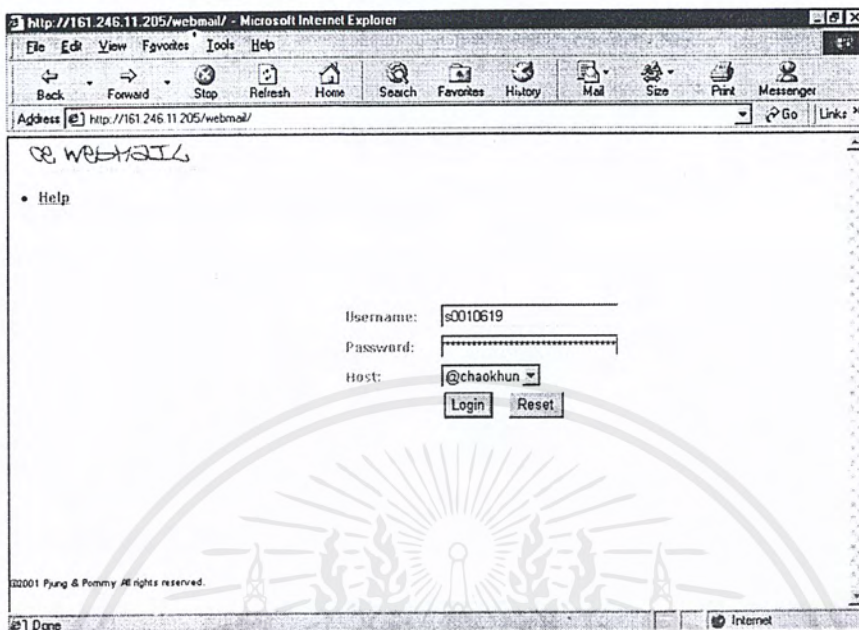
- การติดตั้ง Alliare Jrun

1. ลงโปรแกรมติดตั้งServlet Engine ของ Alliare Jrun ลงบนเครื่อง Linux ที่มีการติดตั้ง Web Server อยู่แล้ว เช่น Apache เป็นต้น
2. เพิ่มรายชื่อกลุ่ม Administrator เพื่อทำการ config ค่าเริ่มต้นของระบบ
3. ลงโปรแกรมเว็บเมล์ลงในไคลเอนต์เดียวกับที่ตั้งค่าเก็บ application ของ Jrun
4. ทำงานแก้ไขไฟล์คุณสมบัติต่างๆภายในระบบ โดยไฟล์นี้จะถูกเก็บอยู่ในไคลเอนต์ WEB-INF ชื่อไฟล์ว่า web.xml
5. ทำงานแก้ไขไฟล์คุณสมบัติต่างๆของโปรแกรมเว็บเมล์หลัก ซึ่งไฟล์นี้จะถูกเก็บอยู่ในไคลเอนต์ที่อยู่ใน WEB-INF ชื่อว่า etc

4.2 การทดลองการทำงาน

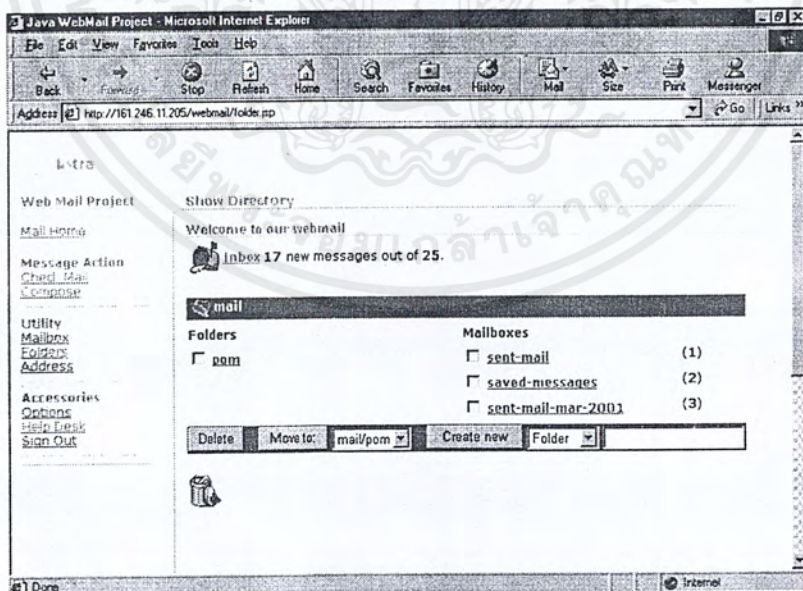
การทดลองนี้ เป็นการทดลองการทำงานของผู้ใช้ที่มีบัญชีรหัสผ่านอยู่บนเครื่องเจ้าคุณและไดมอนต์ อยู่แล้ว มีขั้นตอนดังนี้

- ทำการlogin เข้าสู่ระบบโดยการกรอก URL ของเว็บเซิร์ฟเวอร์ที่รันระบบของเราไว้ผ่านทางบราวเซอร์ ก็จะพบกับหน้าจอ login อันสุดสวายนำใช้นำดูชมอันนี้เข้า เพื่อนๆต้องทำการกรอกรหัสผ่านและรหัสบัญชีผู้ใช้เข้าไปให้ถูกต้องนะครับ และให้ตรงกับโฮสที่ต้องการรับเมลมาอ่านด้วยนะจะบอกให้



รูปที่ 4-1 แสดงหน้าแรกของการเข้าสู่ระบบ

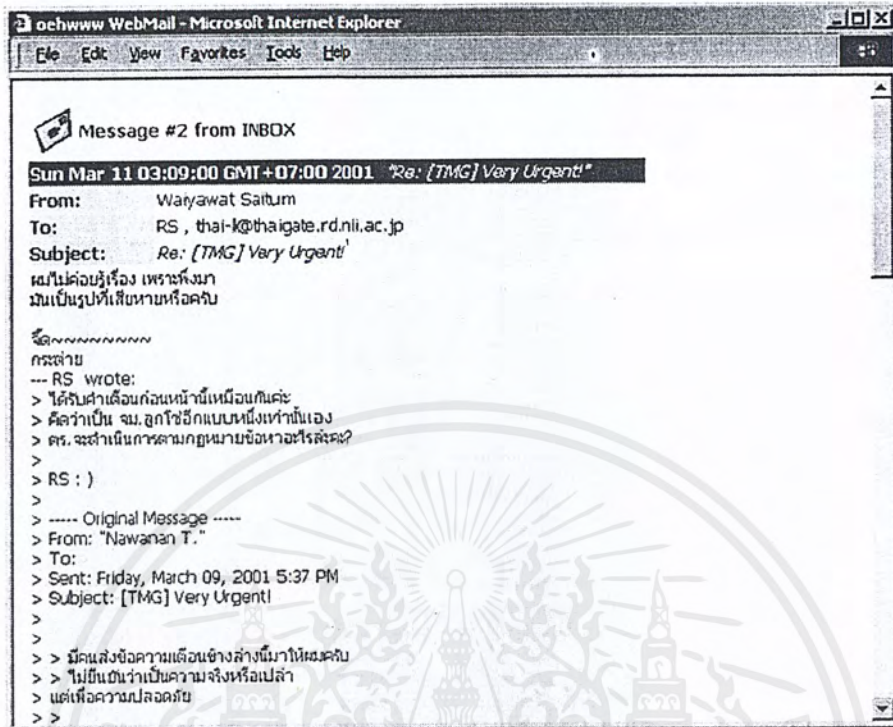
- จากนั้นทำการ login หากรหัสผ่านและชื่อประจำตัวมีอยู่จริงบนเมลเซิร์ฟเวอร์ (Mail Server) ที่เราใช้รับเมลมา (ในการทดลองใช้เครื่องเจ้าคุณเป็นเมลเซิร์ฟเวอร์) มีอยู่จริงก็จะเข้าสู่หน้าต่างเลือกโฟลเดอร์



รูปที่ 4-2 แสดงหน้าจอเมื่อทำการเข้าสู่ระบบได้แล้ว

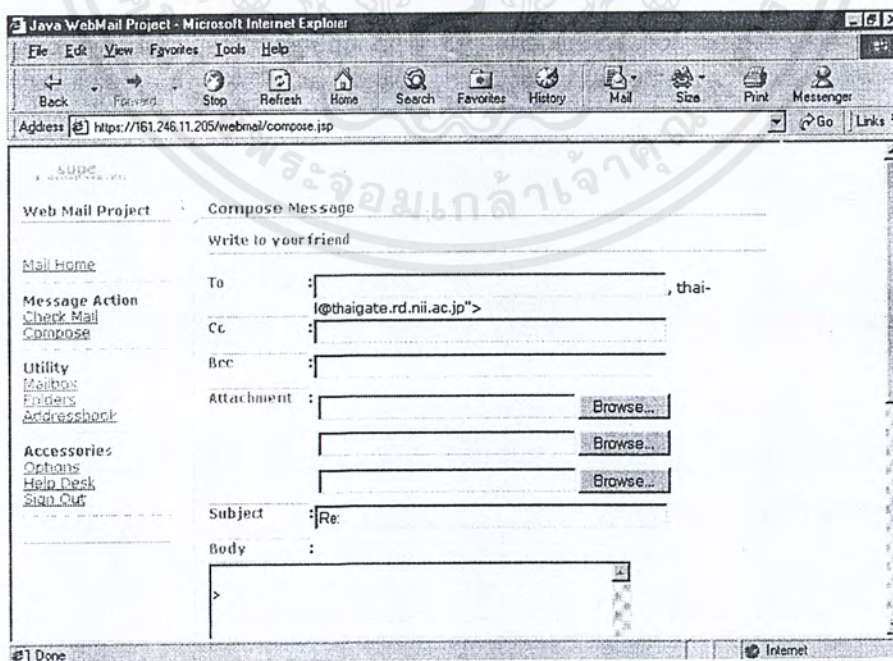
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทดลองเลือกเข้าไปดูตรงมุมมองที่สบายตาตรงเมนู อันนี้เป็นการลดภาพกราฟฟิกต่างๆเพื่อสะดวกต่อการนำไปพิมพ์ออกมาดู



รูปที่ 4-5 แสดงหน้าจอเข้าสู่โหมดPrintable

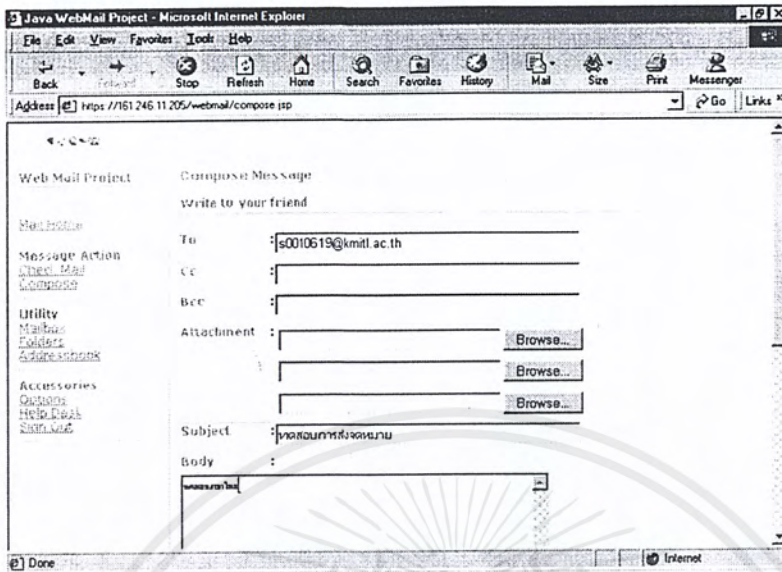
- ทดสอบการตอบจดหมายจากเมนูด้านซ้าย



รูปที่ 4-6 แสดงหน้าจอการตอบจดหมาย

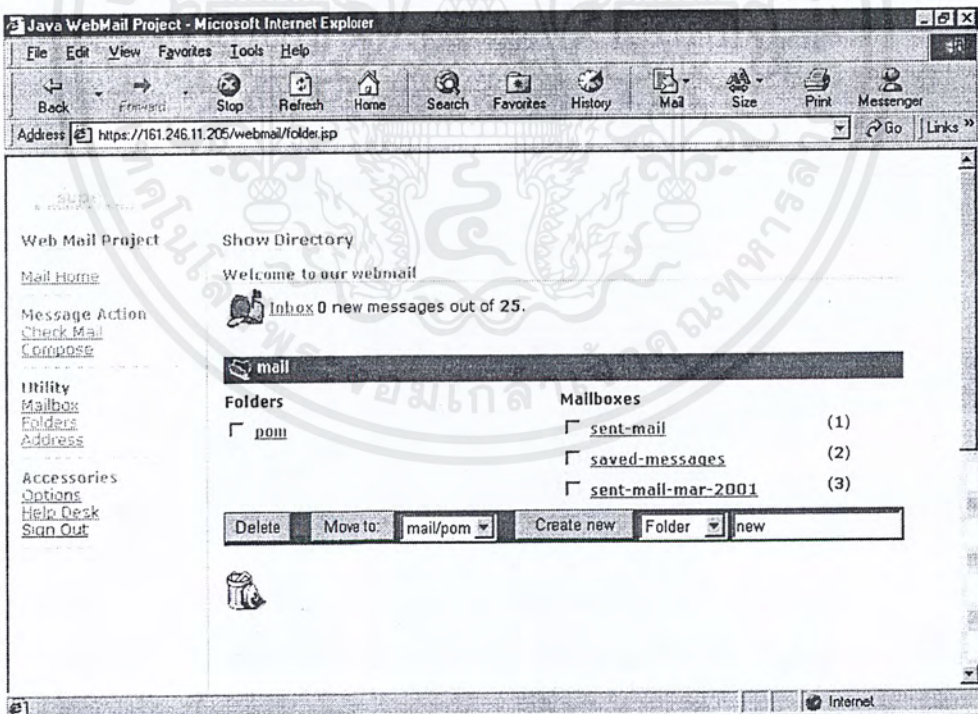
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

□ ทดสอบการเขียนจดหมายใหม่และทำการส่งจดหมาย



รูปที่ 4-7 แสดงหน้าจอการเขียนจดหมายใหม่

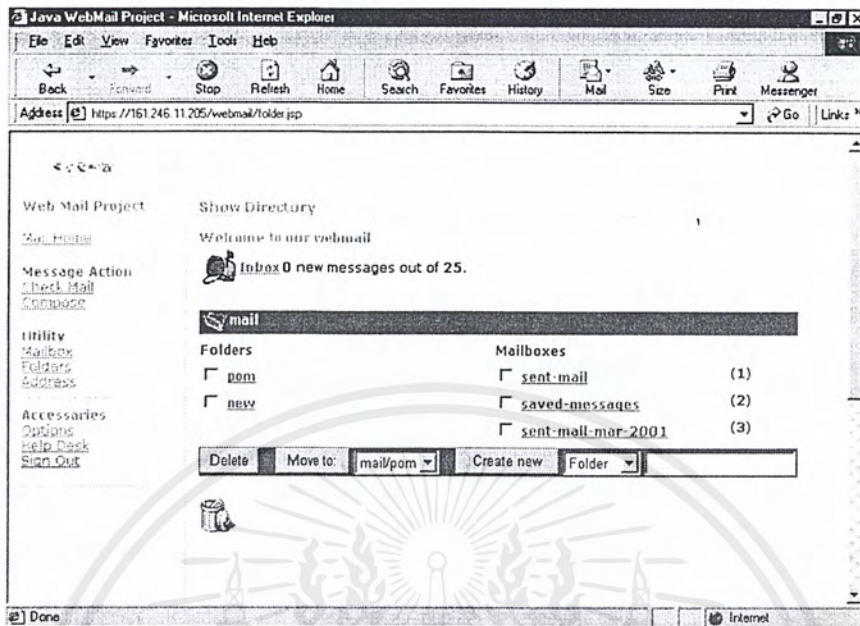
□ ทดสอบการสร้างโฟลเดอร์ใหม่



รูปที่ 4-8 แสดงหน้าจอการการสร้างโฟลเดอร์ใหม่

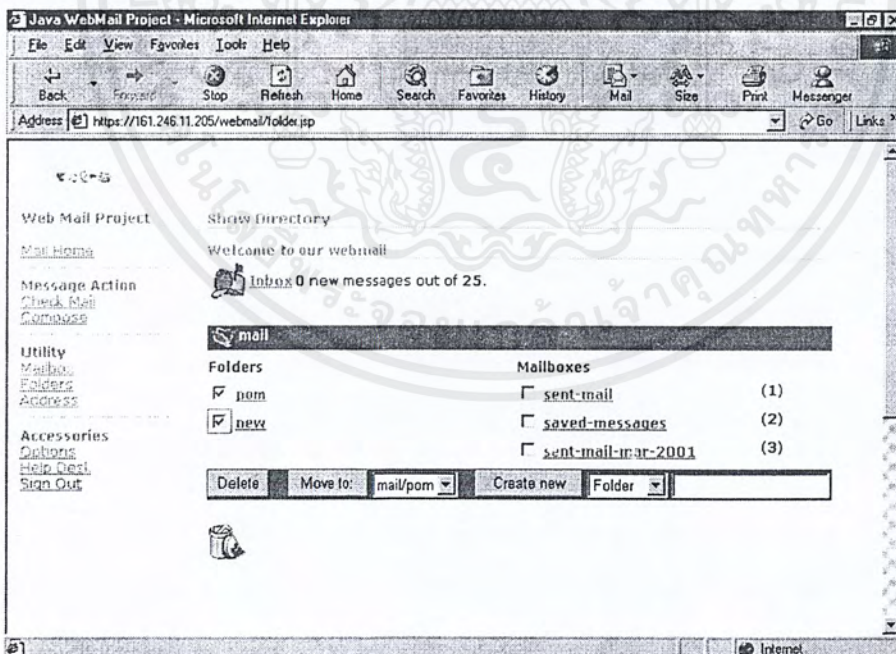
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หากการสร้างสำเร็จจะมีรายชื่อตรงโฟลเดอร์เพิ่มขึ้นมา



รูปที่ 4-9 แสดงหน้าแสดงผลที่ทำการสร้างโฟลเดอร์ใหม่

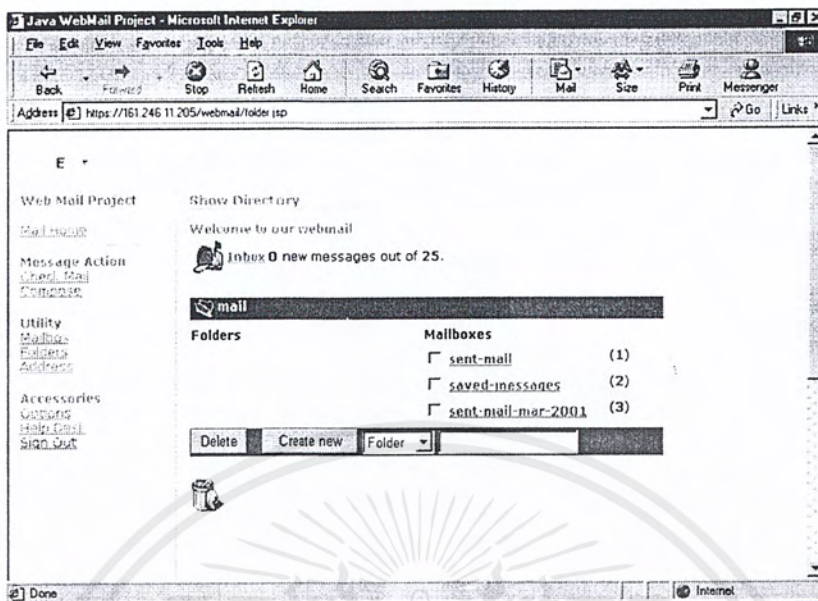
- ทดสอบการลบโฟลเดอร์



รูปที่ 4-10 แสดงหน้าจอการเลือกโฟลเดอร์ที่จะทำการลบ

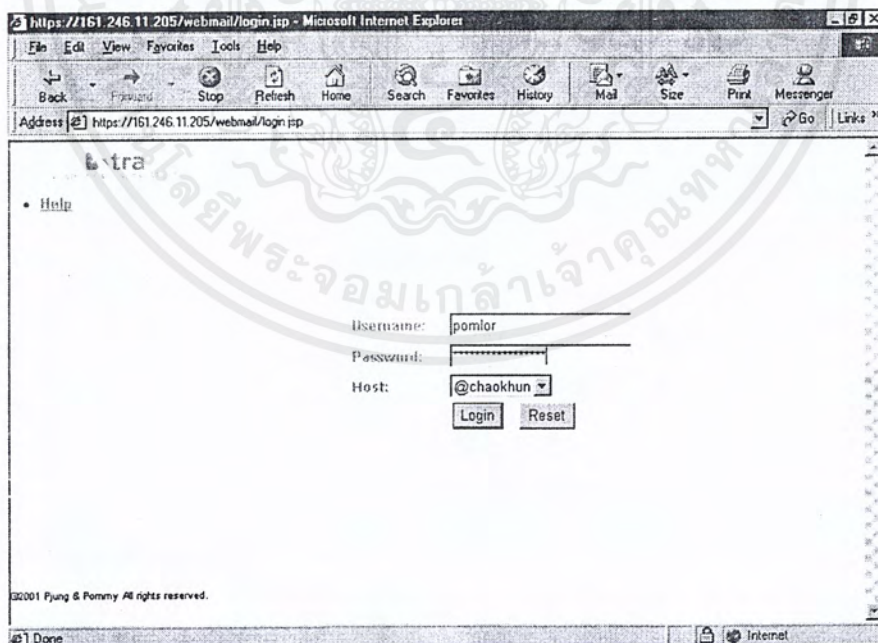
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หากการลบสำเร็จรายชื่อโฟลเดอร์ที่เลือกจะหายไป



รูปที่ 4-11 แสดงหน้าจอผลลัพธ์ที่ได้จากการลบ

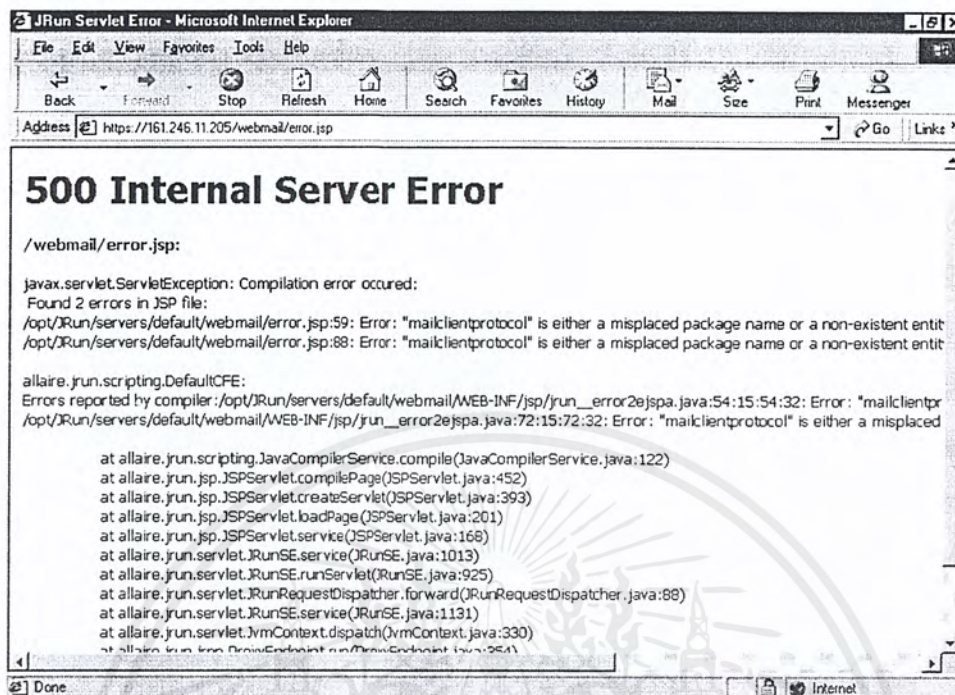
- ทดลองใส่ชื่อและรหัสผ่านที่ไม่มีอยู่จริงเพื่อทดสอบ



รูปที่ 4-12 แสดงหน้าจอทดลองใส่ข้อมูลที่ไม่มีอยู่จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หากเกิดข้อผิดพลาด โปรแกรมจะทำการย้ายไปสู่หน้าที่แสดงข้อความผิดพลาดเพื่อให้ผู้ใช้รู้



รูปที่ 4-12 แสดงหน้าจอการแสดงความผิดพลาดบอกแก่ผู้ใช้

4.3 ผลการทดลองการทำงาน

- ฟอรัมที่สร้างขึ้นสามารถใช้งานได้ มีข้อผิดพลาดบ้างแต่เป็นข้อผิดพลาดที่สามารถแก้ไขได้ เช่น การกำหนดขนาดของตารางยาวเกินไป
- ผลลัพธ์ที่ได้จากการทดลองถูกต้อง
- ในการทดลองส่งจดหมายออกไปหาตัวเอง แล้วลอง refresh ผู้จดหมายพบว่าสามารถได้รับจดหมายใหม่โดยทันทีโดยไม่ต้องทำการตรวจสอบรหัสผ่านใหม่
- การ Scan Virus จากไฟล์ที่มีการส่งมาสามารถทำได้ถูกต้องแต่ความเร็วในการทำงานยังไม่ถึงขนาดทันทีทันใด และประสิทธิภาพของผลลัพธ์ที่ได้จากการตรวจสอบจะขึ้นอยู่กับตัวโปรแกรมที่ใช้ Scan Virus และการอัปเดตรายการไวรัสต่างๆด้วย
- การส่งไฟล์หลายๆไฟล์ไปยังผู้รับปลายทางสามารถทำงานได้ แต่หากรวมขนาดของไฟล์มีขนาดใหญ่กว่าที่กำหนดไว้ ไฟล์ที่ทำการตั้งจะมีการส่งข้อความมาบอกผู้ใช่ว่าขนาดของไฟล์ที่ส่งไปมีขนาดเกินจากที่กำหนดไว้ ทำให้ไม่สามารถส่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การแก้ไขข้อมูลในสมุดที่อยู่ยังไม่เป็นแบบ Real time จนกว่าจะทำการออกจากระบบข้อมูลที่แก้ไขในสมุดบันทึกจึงจะทำการอัปเดต
- ในกรณีที่ทำการใช้งานระบบจดหมายอิเล็กทรอนิกส์ผ่านเว็บอยู่ จะไม่สามารถเข้าไปปรับจดหมายจากการใช้โปรแกรม PINE หรือ โปรแกรมรับเมลล์ห้ออื่นพร้อมกันได้ เนื่องจากโปรแกรมทำการล็อกไคลเอนคอรันนั้นไว้

4.4 สรุปผลการทดลอง

1. การประยุกต์ใช้งานเว็บเมลล์ที่สร้างขึ้น สามารถทำงานได้ตามต้องการ แต่ในด้านการแสดงผลยังไม่ดีเท่าที่ควร
2. การใช้งานสามารถใช้งานจากที่ใดก็ได้ โดยให้เชื่อมต่อเครื่องคอมพิวเตอร์เข้ากับเครือข่าย อาจใช้เครื่องคอมพิวเตอร์ หรือ NoteBook ต่อกับ โมเด็ม การติดต่อกับเว็บเซิร์ฟเวอร์ จะใช้โพรโทคอล HTTP กับ HTTPS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

5.1 ปัญหาและอุปสรรคในการทำงาน

1. เทคโนโลยี JSP เป็นเทคโนโลยีที่ออกสู่ตลาดในระยะเวลาไม่ถึง 1 ปี และมีหนังสือออกมาได้ไม่นาน ดังนั้นแหล่งข้อมูลที่จะนำมาประกอบการศึกษาค้นคว้า ไม่ว่าจะเป็นหนังสือวิชาการต่างๆ หรือข้อมูลจากเว็บไซต์จึงน้อยมาก และส่วนใหญ่เป็นภาษาอังกฤษ ทำให้การศึกษาค้นคว้าต้องใช้เวลาอันยาวนานยิ่งขึ้น
2. เนื่องจากการทดสอบการทำงานของโครงการ ต้องทดสอบบนเครื่องข่ายคอมพิวเตอร์ ดังนั้นจึงต้องการเครื่องคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไป จึงต้องใช้เครื่องคอมพิวเตอร์ของกลุ่มอื่นๆ ทำให้เกิดความล่าช้าในการดำเนินงานเพราะต้องแบ่งเวลาการใช้เครื่อง
3. ความเชี่ยวชาญในการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) มีเพียงระดับหนึ่ง ทำให้ต้องใช้เวลาในการศึกษาเพิ่มเติม
4. ขาดความรู้ในการติดตั้งโปรแกรมประยุกต์ที่ทำหน้าที่เป็น Servlet Engine ทำให้การพัฒนาโปรแกรมทำได้ช้า

5.2 สรุปผลที่ได้จากการทำโครงการ

1. ได้รับความรู้และความเชี่ยวชาญในการเขียนโปรแกรมเชิงวัตถุมากขึ้น
2. ได้เรียนรู้เทคโนโลยีทางด้านจาวา และเรียนรู้การเขียนโปรแกรมโดยใช้ JavaMail API
3. ได้เรียนรู้ระบบจดหมายอิเล็กทรอนิกส์
4. ได้เรียนรู้การทำงานเป็นกลุ่ม
5. ได้เรียนรู้การออกแบบโปรแกรมโดยใช้ภาษา UML

5.3 ข้อเสนอแนะ

การประยุกต์ใช้งานระบบจดหมายอิเล็กทรอนิกส์ผ่านเว็บ ที่จัดทำขึ้นนี้ ควรที่จะนำไปใช้ติดต่อบนเครื่องที่ค่อนข้างจะมีประสิทธิภาพในการประมวลผล และมีหน่วยความจำหลักที่ค่อนข้างสูง เนื่องจากโดยทั่วไปแล้ว เจเอสพี (JSP) ต้องมีการแปลงโค้ดของ สคริปต์โปรแกรมประยุกต์ ให้อยู่ในรูปแบบเป็นลักษณะของภาษาจาวาก่อน และส่วนตัว ไบต์โค้ด (byte code) ของตัวภาษาจาวานั้นมักจะไม่ได้ทำงานบนหน่วยประมวลผลจริงหรือ ที่เรียกว่า CPU แต่จะทำการประมวลผลบนหน่วยประมวลผลเสมือน (Virtual Machine) ทำให้การเอ็กเซ็กคิวต์คำสั่งจะช้ากว่าการทำงานบนหน่วยประมวลผลจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสรุปแล้วจากการทำงานที่ผ่านมา นั้น ได้ความรู้ในเรื่องภาษาจาวเพิ่มขึ้นรวมถึงความรู้ในการเขียนโปรแกรมด้านอื่นๆ และวิธีการประยุกต์ใช้การแสดงผลข้อมูลรวมถึงการออกแบบส่วนติดต่อกับผู้ใช้ให้ผู้ใช้มีความสะดวกมากยิ่งขึ้น



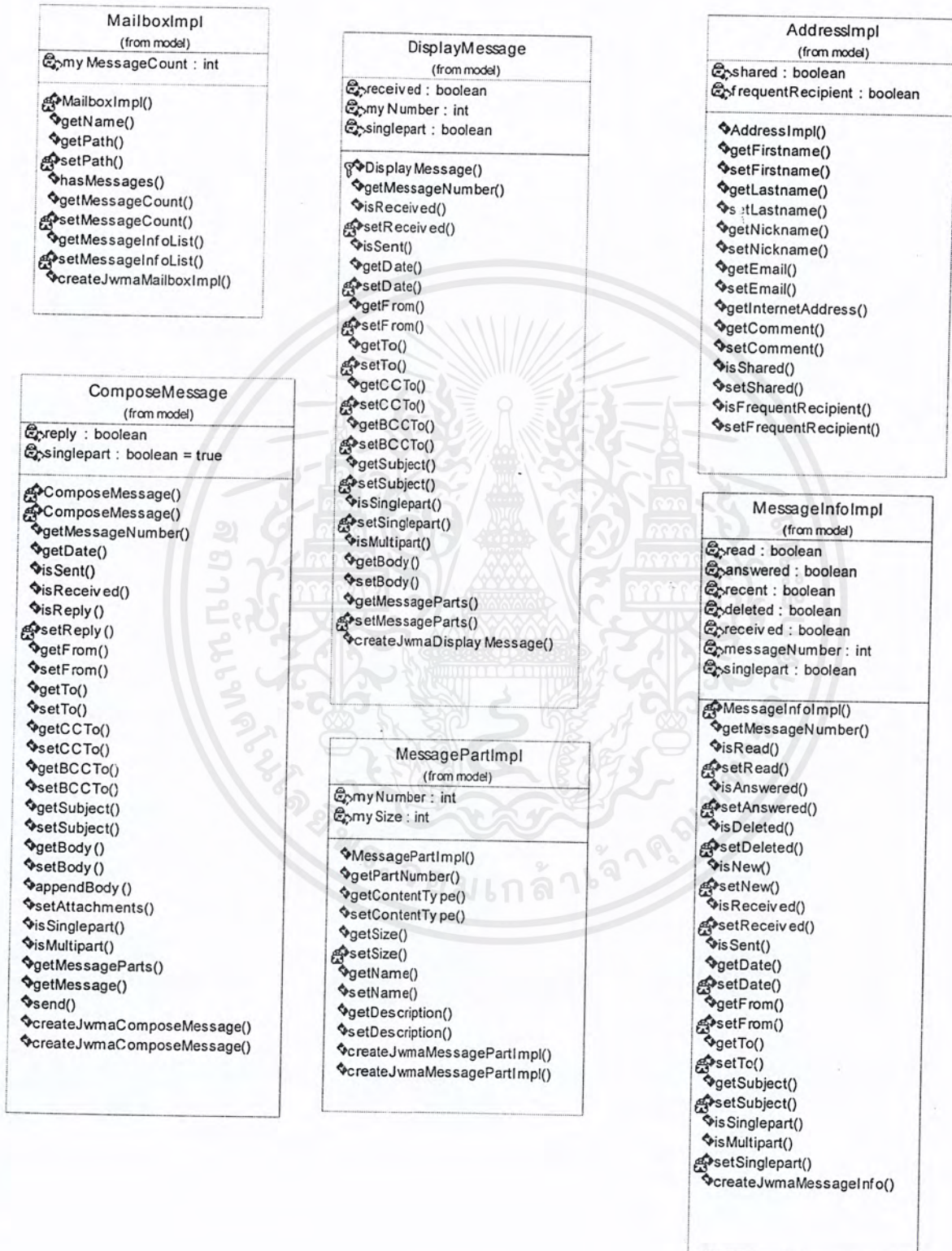
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



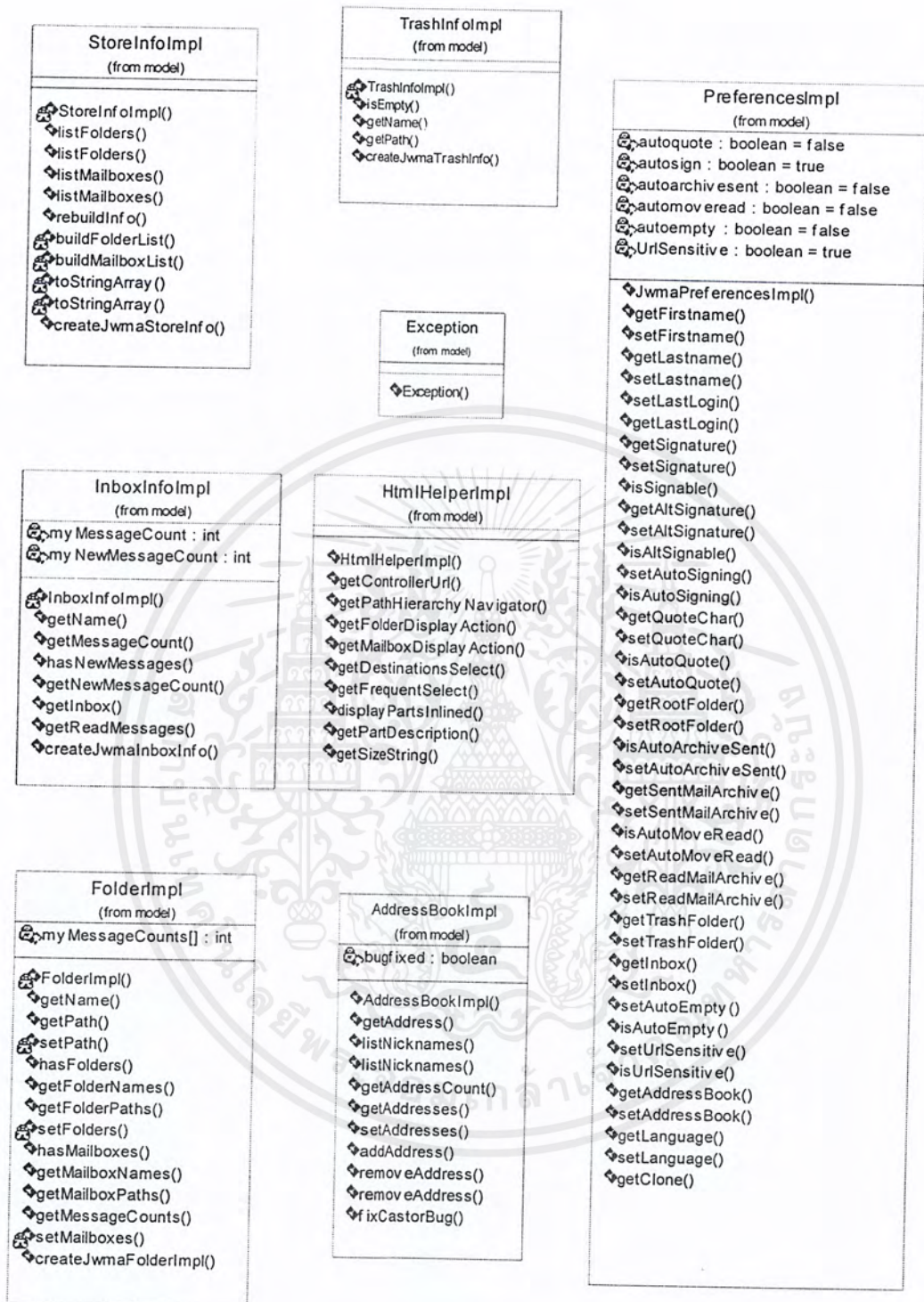
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คลาสทั้งหมดในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Controller (from webmail)
⚡init() ⚡getJwmaSession() ⚡service() ⚡doDispatchSessionActions() ⚡doDispatchFolderActions() ⚡doDispatchMailboxActions() ⚡doDispatchMessageActions() ⚡doDispatchPreferencesActions() ⚡doDispatchAddressbookActions() ⚡doLogin() ⚡doLogout() ⚡doRedirect() ⚡doDisplayFolder() ⚡doDeleteFolders() ⚡doCreateFolder() ⚡doMoveFolders() ⚡doDisplayMailbox() ⚡doCreateMailbox() ⚡doDisplayMessage() ⚡doDeleteMessages() ⚡doMoveMessages() ⚡doComposeMessage() ⚡doComposeMessage() ⚡doDisplayMessagePart() ⚡doUpdatePreferences() ⚡doAddAddress() ⚡doRemoveAddresses() ⚡valid() ⚡getFolders() ⚡getAllMessageNumbers() ⚡moveMessages() ⚡deleteMessages() ⚡processPlainText() ⚡getServletInfo()

MultipartInputStream (from util)
⚡myLimit : int ⚡bytesRead : int = 0
⚡MultipartInputStream() ⚡getName() ⚡getContentType() ⚡getInputStream() ⚡getOutputStream() ⚡read() ⚡read() ⚡checkLimit()

MultipartRequest (from util)
⚡myLimit : int = FORMDATA_LIMIT ⚡\$FORMDATA_LIMIT : int = 1024 * 1024
⚡MultipartRequest() ⚡getParameterNames() ⚡getParameter() ⚡getParameterValues() ⚡hasAttachments() ⚡getAttachments() ⚡processRequest()

Session (from webmail)
⚡myMessage : int ⚡died : boolean = false ⚡firsttime : boolean = false
⚡JwmaSession() ⚡setRequest() ⚡getRequest() ⚡setResponse() ⚡getResponse() ⚡getWebSession() ⚡setWebSession() ⚡redirect() ⚡redirectToLast() ⚡storeBean() ⚡retrieveBean() ⚡removeBeans() ⚡storeLogin() ⚡initMailSession() ⚡endMailSession() ⚡getMailSession() ⚡getStore() ⚡getActualFolder() ⚡setActualFolder() ⚡setActualMailbox() ⚡getActualMailbox() ⚡getTrashFolder() ⚡setTrashFolder() ⚡getMessage() ⚡setMessage() ⚡isFirstTime() ⚡getPreferences() ⚡savePreferences() ⚡loadPreferences() ⚡getFileName() ⚡getSenderIdentity() ⚡getSenderAddress() ⚡end() ⚡finalize() ⚡createJwmaSession()

MimeBodyPartDataSource (from util)
⚡myData[] : byte
⚡MimeBodyPartDataSource() ⚡getContentType() ⚡getName() ⚡getInputStream() ⚡getOutputStream()

PropertiesLoader (from util)
⚡loadProperties() ⚡loadProperties()

SendMail (from webmail)
⚡init() ⚡getJwmaSession() ⚡service() ⚡doSendMessage() ⚡getServletInfo()

Kemel (from webmail)
⚡poolsize : int
⚡Kernel() ⚡setup() ⚡prepareLog() ⚡sysLog() ⚡debugLog() ⚡getSiteUrl() ⚡getControllerUrl() ⚡getMailClientProtocol() ⚡getSmtphost() ⚡getViewUrl() ⚡getUrlMatchPattern() ⚡getPreferencesTemplate() ⚡loadPreferences() ⚡savePreferences() ⚡initMatcherPool() ⚡flushMatcherPool() ⚡getMatcher() ⚡releaseMatcher() ⚡getReference()

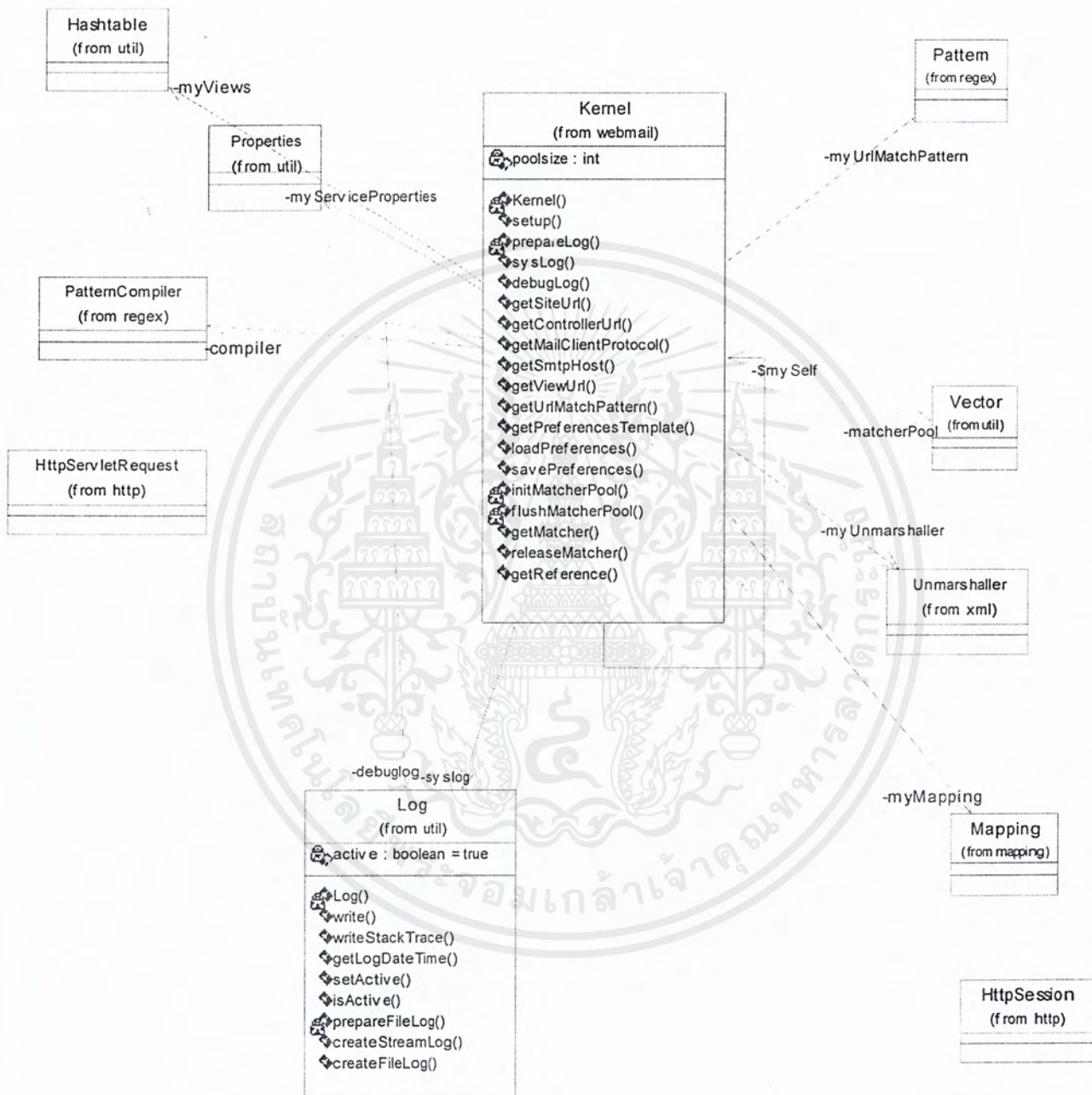
Log (from util)
⚡active : boolean = true
⚡Log() ⚡write() ⚡writeStackTrace() ⚡getLogDateTime() ⚡setActive() ⚡isActive() ⚡prepareFileLog() ⚡createStreamLog() ⚡createFileLog()

FomdataMultipart (from util)
⚡removed : boolean = false
⚡FomdataMultipart() ⚡FomdataMultipart() ⚡getParameters() ⚡processBodyParts() ⚡processBodyPart() ⚡extractName() ⚡extractFileName() ⚡extractValue() ⚡addParameter()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คลาสไดอะแกรมส่วนจัดการค่าคงที่ตัวแปรที่ใช้ในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

อินเทอร์เน็ตทั้งหมดในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] อาจารย์ อภินทร อุณาภูล : "Object-Oriented Analysis and Design", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2543
- [2] Marty Hall : "Core Servlets and JavaServer Pages (JSP) ",Prentice Hall PTR/Sun Microsystems Press 2000
- [3] Mark Wutka : "Special Edition Using Java Server Pages and Servlets ",Que 2000
- [4] Damon Hougland, Aaron Tavistock : "Core JSP", Prentice Hall Computer Books 2000
- [5] Jason Hunter, William Crawford : "Java Servlet Programming",O'Reilly & Associations 1998
- [6] ดร. วีระศักดิ์ ชิงฉาวร : "Fundamental of JAVA Programming Volume 1 ",School & University Media 1998
- [7] <http://java.sun.com/products/jsp/>
- [8] <http://java.sun.com/products/servlet/>
- [9] Govind Seshadri : "Understanding JavaServer Pages Model 2 architecture", http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc_p.html