

คอมโพเนนต์สำหรับการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร

COMPONENTS FOR TIME SCHEDULING AND INFORMATION
BROADCASTING



กิริติ อนุรักษ์ภราดร
จันทนา สอดระภูด
เฉลิมขวัญ พิพัฒน์เจริญวงศ์

เลขหมู่.....
เลขทะเบียน..... 47327
วัน, เดือน, ปี 30 ส.ย. 2545

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

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**COMPONENTS FOR TIME SCHEDULING AND INFORMATION
BROADCASTING**



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2002**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ

คอมพิวเตอร์สำหรับการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร

COMPONENTS FOR TIME SCHEDULING AND INFORMATION BROADCASTING

ชื่อนักศึกษา

นายกิติ อรุณย์ภราดร

42050369

นางสาวจันทนา สอตระกุล

42050372

นางสาวเฉลิมขวัญ พัฒน์เจริญวงศ์

42050375

ภาควิชา

คณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา

ดร.ศรัณย์ อินทโกสุม

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้แก้ปัญหานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ปีการศึกษา 2545

คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	ผศ.ดร.วีระ บุญจริง
กรรมการ	อ.รัชต์พร อาชีวะ
กรรมการและอาจารย์ที่ปรึกษา	ดร.ศรัณย์ อินทโกสุม

(ผู้ช่วยศาสตราจารย์ไพรบุลย์ พันธรัักษ์พงษ์)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ	คอมพิวเตอร์สำหรับการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร	
ชื่อนักศึกษา	นายกীরติ อนุรักษัภราดร	42050369
	นางสาวจันทนา สอดตระกูล	42050372
	นางสาวเฉลิมขวัญ พิพัฒน์เจริญวงศ์	42050375
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2545	
อาจารย์ที่ปรึกษา	ดร.ศรัณย์ อินทโกสุม	

บทคัดย่อ

เนื่องจากปัญหาที่พบส่วนมากในองค์กรขนาดใหญ่ คือ การสูญหายของข้อมูลข่าวสาร และการนัดหมายภายในองค์กร ดังนั้นการทำหัวข้อปัญหาพิเศษครั้งนี้จึงได้จัดทำขึ้นเพื่อแก้ไขปัญหานี้ โดยทำการพัฒนาโปรแกรมเพื่อกระจายข้อมูลข่าวสาร และจัดการตารางเวลานัดหมายภายในองค์กร ซึ่งใช้แนวคิดในการเขียนโปรแกรมแบบกระจาย และใช้เทคนิคการพัฒนาเป็นคอมพิวเตอร์เพื่อสร้างคอมพิวเตอร์ที่สามารถนำกลับมาใช้ใหม่ สำหรับงานดังกล่าว

Special Project Title	COMPONENTS FOR TIME SCHEDULING AND INFORMATION BROADCASTING	
Students	Mr. Keerati Anurakpradorn	42050369
	Miss Janthana Sotrakul	42050372
	Miss Chaleamkwan Pipatcharoenwong	42050375
Degree	Bachelor of Science	
Department	Mathematics and Computer Science, Faculty of Science	
Programme	Computer Science	
Academic Year	2002	
Special Project Advisor	Dr. Sarun Intakosum	

ABSTRACT

Two major problems in a large organization are losing some important information and scheduling an appointment. This special project proposes a solution to the problems by developing a program to distribute information and organize time schedule. The project applies distributed programming concept and component-based development technique to create reusable components for such tasks.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องคอมพิวเตอร์สำหรับการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ดร. ศรีณย์ อินทโกสุม อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งเพื่อน ๆ และน้อง ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ
มีนาคม 2546



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VIII

สารบัญรูป.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 สมมติฐานของการศึกษา.....1

1.4 ขอบเขตของการศึกษา.....2

1.5 ขั้นตอนการศึกษา.....2

1.6 ข้อตกลงเบื้องต้น.....4

1.7 ข้อยกเว้นของภาษา.....4

1.8 คำจำกัดความที่ใช้ในการศึกษา.....5

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....6

2.1 คอมพิวเตอร์.....6

2.1.1 คุณสมบัติของคอมพิวเตอร์.....7

2.1.2 ข้อดีของคอมพิวเตอร์.....7

2.2 กรู๊ปแวร์.....8

2.2.1 ประเภทของระบบ 2 ระบบที่ไม่นับเป็นกรู๊ปแวร์.....9

2.2.2 ความสามารถของเทคโนโลยี.....9

2.2.3 ส่วนประกอบของกรู๊ปแวร์.....10

2.3 ยูเอ็มแอล (UML, Unified Modeling Language)16

2.3.1 ประเภทไดอะแกรมของ UML.....17

2.4 เจดีบีซี (JDBC).....28

เอกสารนี้เป็นเอกสารลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4.2 รายละเอียดของ JDBC แต่ละประเภท.....	30
2.4.3 การเขียนโปรแกรมเพื่อติดต่อฐานข้อมูล JDBC ทั้ง 4 ประเภท.....	33
2.5 อาร์เอ็มไอ (RMI)	40
บทที่ 3 การออกแบบโปรแกรม.....	45
3.1 ขอบเขตของระบบการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร.....	45
3.1.1 ระบบการจัดการสมาชิก.....	45
3.1.2 ระบบการจัดการเอกสาร.....	45
3.1.3 ระบบจัดการตารางเวลา.....	45
3.2 กิจกรรมหลัก.....	46
3.2.1 กิจกรรมระบบการจัดการสมาชิก.....	46
3.2.2 กิจกรรมระบบการจัดการเอกสาร.....	46
3.2.3 กิจกรรมระบบการจัดการตารางเวลา.....	48
3.3 ยูสเคสไดอะแกรม.....	49
3.4 คลาสไดอะแกรม.....	53
3.4.1 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการสมาชิกฝั่งไคลเอนต์.....	53
3.4.2 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการสมาชิกฝั่งเซิร์ฟเวอร์.....	54
3.4.3 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการกระจายข้อมูลข่าวสารฝั่งไคลเอนต์.....	55
3.4.4 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการกระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์.....	56
3.4.5 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการตารางเวลาฝั่งไคลเอนต์.....	57
3.4.6 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการตารางเวลาฝั่งเซิร์ฟเวอร์.....	58
3.5 ซีควเอนซ์ไดอะแกรม.....	61
3.5.1 ระบบการจัดการสมาชิก.....	61
3.5.2 ระบบการจัดการเอกสาร.....	62
3.5.3 ระบบการจัดการตารางเวลา.....	64
3.6 แอ็กทิวิตีไดอะแกรม.....	66
3.6.1 ระบบการจัดการสมาชิก.....	66

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.6.2 ระบบการจัดการเอกสาร.....	67
3.6.3 ระบบการจัดการตารางเวลา.....	69
3.7 โมเดล-วิว-คอนโทรลเลอร์.....	71
3.7.1 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์การจัดการสมาชิกฝั่งไคลเอนต์.....	71
3.7.2 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์การจัดการสมาชิกฝั่งเซิร์ฟเวอร์.....	72
3.7.3 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์กระจายข้อมูลข่าวสารฝั่งไคลเอนต์.....	73
3.7.4 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์กระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์.....	74
3.7.5 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์จัดการตารางเวลาฝั่งไคลเอนต์.....	75
3.7.6 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์จัดการตารางเวลาฝั่งเซิร์ฟเวอร์.....	76
3.8 สิ่งที่คอมโพเนนต์ต้องการและจัดเตรียมไว้ให้.....	77
3.8.1 สิ่งที่คอมโพเนนต์สำหรับจัดการสมาชิกต้องการและจัดเตรียมไว้ให้.....	77
3.8.2 สิ่งที่คอมโพเนนต์สำหรับกระจายข้อมูลข่าวสารต้องการและจัดเตรียมไว้ให้.....	78
3.8.3 สิ่งที่คอมโพเนนต์สำหรับจัดการตารางเวลาต้องการและจัดเตรียมไว้ให้.....	79
บทที่ 4 ประเมินและอภิปรายผล.....	80
4.1 ลักษณะของระบบการจัดการตารางเวลานัดหมายและกระจายข้อมูลข่าวสาร.....	80
4.2 ข้อกำหนดของการประเมิน.....	81
4.3 การประเมิน.....	81
4.4 การอภิปรายผล.....	82
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....	83
5.1 สรุปผลการดำเนินงาน.....	83
5.2 ข้อเสนอแนะ.....	84
บรรณานุกรม.....	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

ภาคผนวก

ภาคผนวก ก คู่มือการใช้งาน.....	86
ภาคผนวก ข รายละเอียดของอินเทอร์เน็ต.....	103



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางการทำงานตามขั้นตอนของการศึกษา.....	3
2.1 Multiplicity.....	21
2.2 โครงสร้างข้อมูลภายในชุด ResultSet.....	38
2.3 การเปรียบเทียบระหว่าง distributed กับ nondistributed.....	40



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 Stand-alone Machine.....	6
2.2 ลักษณะการประกอบกันของคอมโพเนนต์.....	8
2.3 ตัวอย่างการทำงานของ Workflow.....	11
2.4 การไหลของข้อมูลแบบเรียงตามลำดับ.....	12
2.5 การไหลของข้อมูลแบบคู่ขนาน.....	12
2.6 การไหลของข้อมูลแบบมีการย้อนกลับของงาน.....	12
2.7 การไหลของข้อมูลแบบวงกลม.....	12
2.8 การไหลของข้อมูลแบบวงล้อ.....	12
2.9 การไหลของข้อมูลแบบการเชื่อมต่ออย่างสมบูรณ์.....	13
2.10 API สำหรับอีเมล.....	13
2.11 ยูสเคสไดอะแกรม.....	17
2.12 สัญลักษณ์ของระบบ.....	17
2.13 สัญลักษณ์ของยูสเคส.....	18
2.14 สัญลักษณ์ของแอ็กเตอร์.....	18
2.15 สัญลักษณ์ของคลาส.....	19
2.16 ความสัมพันธ์แบบ Dependency.....	20
2.17 ความสัมพันธ์แบบ Generalization.....	20
2.18 ความสัมพันธ์แบบ Binary Association.....	21
2.19 ความสัมพันธ์แบบ N-ary Association.....	21
2.20 สัญลักษณ์ของความสัมพันธ์แบบ Composition และ Aggregation.....	22
2.21 ซีเควนซ์ไดอะแกรม.....	23
2.22 ส่วนประกอบของออบเจกต์.....	23
2.23 คอลแลบอเรชันไดอะแกรม.....	24
2.24 สเตตชาร์ตไดอะแกรม.....	24
2.25 รายละเอียดภายในสเตต.....	25
2.26 แอ็กทิวิตีไดอะแกรม.....	26
2.27 สัญลักษณ์ของคอมโพเนนต์.....	27
2.28 องค์ประกอบหลักของคอมโพเนนต์ไดอะแกรม.....	27
2.29 สัญลักษณ์ของอาร์คแวร์ (โหนด).....	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.30 ดีพลอยเมนต์ไคอะแกรม.....	28
2.31 การเข้าถึงฐานข้อมูลโดยผ่าน JDBC ซึ่งใช้ไดรฟ์เวอร์พื้นฐาน.....	29
2.32 การติดต่อ JDBC โดยใช้ ODBC ไดรฟ์เวอร์.....	29
2.33 การติดต่อกับฐานข้อมูลโดยผ่าน ODBC.....	30
2.34 การติดต่อฐานข้อมูลประเภทที่ 2.....	31
2.35 การติดต่อฐานข้อมูลประเภทที่ 3.....	32
2.36 การติดต่อฐานข้อมูลประเภทที่ 4.....	33
2.37 การกำหนดพฤติกรรมโดยใช้อินเทอร์เน็ต.....	42
2.38 การสนับสนุนให้สามารถสร้าง executable code ของ RMI.....	42
2.39 สถาปัตยกรรมของ RMI.....	43
3.1 ยูสเคสไคอะแกรมของระบบการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร.....	49
3.2 ยูสเคสไคอะแกรมของระบบการจัดการสมาชิก.....	50
3.3 ยูสเคสไคอะแกรมของระบบการจัดการเอกสาร.....	50
3.4 ยูสเคสไคอะแกรมของระบบการส่งเอกสาร.....	51
3.5 ยูสเคสไคอะแกรมของระบบการรับเอกสาร.....	51
3.6 ยูสเคสไคอะแกรมของระบบจัดการตารางเวลาส่วนบุคคล.....	52
3.7 ยูสเคสไคอะแกรมของระบบจัดเวลานัดหมายภายในองค์กร.....	52
3.8 คลาสไคอะแกรมสำหรับคอมโพเนนต์ส่วนจัดการสมาชิกฝั่งไคลเอนต์.....	53
3.9 คลาสไคอะแกรมสำหรับคอมโพเนนต์ส่วนจัดการสมาชิกฝั่งเซิร์ฟเวอร์.....	54
3.10 คลาสไคอะแกรมสำหรับคอมโพเนนต์ส่วนกระจายข้อมูลข่าวสารฝั่งไคลเอนต์.....	55
3.11 คลาสไคอะแกรมสำหรับคอมโพเนนต์ส่วนกระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์.....	56
3.12 คลาสไคอะแกรมสำหรับคอมโพเนนต์ส่วนจัดการตารางเวลาฝั่งไคลเอนต์.....	57
3.13 คลาสไคอะแกรมสำหรับคอมโพเนนต์ส่วนจัดการตารางเวลาฝั่งเซิร์ฟเวอร์.....	58
3.14 คลาสไคอะแกรมของพนักงาน.....	59
3.15 คลาสไคอะแกรมของฐานข้อมูลตารางเวลา.....	59
3.16 คลาสไคอะแกรมของฐานข้อมูลเอกสาร.....	60
3.17 คลาสไคอะแกรมของฐานข้อมูลผู้ใช้.....	60
3.18 ซีควเอนซ์ไคอะแกรมของระบบการจัดการสมาชิก.....	61
3.19 ซีควเอนซ์ไคอะแกรมของระบบการส่งเอกสาร.....	62

เอกสารนี้เป็นเอกสารทูลงานไวสำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 3.20 ซีควเอนซ์ไคอะแกรมของระบบการรับเอกสาร..... 63
 ไม่ว่าการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.21 ซีเควนซ์ไคอะแกรมของระบบการจัดการตารางเวลาส่วนบุคคล.....	64
3.22 ซีเควนซ์ไคอะแกรมของระบบการจัดการตารางเวลานัดหมายภายในองค์กร.....	65
3.23 แอ็กทิวิตี้ไคอะแกรมของระบบการจัดการสมาชิก.....	66
3.24 แอ็กทิวิตี้ไคอะแกรมของระบบการส่งเอกสาร.....	67
3.25 แอ็กทิวิตี้ไคอะแกรมของระบบการรับเอกสาร.....	68
3.26 แอ็กทิวิตี้ไคอะแกรมของระบบการจัดการตารางเวลาส่วนบุคคล.....	69
3.27 แอ็กทิวิตี้ไคอะแกรมของระบบจัดการตารางเวลานัดหมายภายในองค์กร.....	70
3.28 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์การจัดการสมาชิกฝั่งไคลเอนต์.....	71
3.29 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์การจัดการสมาชิกฝั่งเซิร์ฟเวอร์.....	72
3.30 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์กระจายข้อมูลข่าวสารฝั่งไคลเอนต์.....	73
3.31 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์กระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์.....	74
3.32 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์กระจายข้อมูลข่าวสารฝั่งไคลเอนต์.....	75
3.33 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมโพเนนต์กระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์.....	76
3.34 สิ่งที่คอมโพเนนต์สำหรับจัดการสมาชิกต้องการและจัดเตรียมไว้ให้.....	77
3.35 สิ่งที่คอมโพเนนต์สำหรับกระจายข้อมูลข่าวสารต้องการและจัดเตรียมไว้ให้.....	78
3.36 สิ่งที่คอมโพเนนต์สำหรับจัดการตารางเวลาต้องการและจัดเตรียมไว้ให้.....	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันนี้องค์กรต่างๆ มีการขยายตัวจนมีขนาดใหญ่ ทำให้เกิดปัญหาในด้านการติดต่อสื่อสารขึ้นภายในองค์กร เช่น เกิดความซับซ้อน ความยุ่งยาก และการใช้ระยะเวลาที่มากในการส่งข้อมูลข่าวสารชุดหนึ่งไปยังบุคคลที่เกี่ยวข้องกับข้อมูล ซึ่งทำให้ข้อมูลข่าวสารที่ได้รับนั้นไม่ทันต่อเหตุการณ์ และข้อมูลข่าวสารที่ส่งนั้นอาจเกิดความผิดพลาดหรือสูญหายได้ นอกจากนี้ การที่บุคลากรภายในองค์กร อาจมีตารางเวลาการทำงานไม่ตรงกัน ซึ่งทำให้การนัดหมายทำได้โดยลำบาก และอาจก่อให้เกิดความผิดพลาดในการนัดหมาย เหตุการณ์นี้อาจนำมาซึ่งความเสียหายขององค์กร และด้วยเหตุนี้จึงมีแนวคิดที่จะนำระบบเครือข่ายมาช่วยในการติดต่อ สื่อสารและกระจายข้อมูลต่างๆภายในองค์กร โดยพัฒนาโปรแกรมเพื่อกระจายข้อมูลข่าวสารถึงบุคลากรทุกฝ่ายที่เกี่ยวข้องภายในองค์กร และจัดการตารางเวลานัดหมาย โดยโปรแกรมที่พัฒนาขึ้นจะใช้แนวคิดของการพัฒนาแบบคอมโพเนนต์ ซึ่งสามารถนำไปพัฒนาต่อทำให้โปรแกรมดูแลรักษาได้โดยสะดวกและคอมโพเนนต์ที่พัฒนาขึ้น อาจสามารถนำไปประยุกต์ใช้กับโปรแกรมอื่นที่เกี่ยวข้องได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

จัดทำต้นแบบโปรแกรมเพื่อใช้ในการจัดการตารางนัดหมายและจัดการข่าวสาร โดยใช้การออกแบบและพัฒนาโปรแกรมในรูปแบบคอมโพเนนต์

1.3 สมมติฐานของการศึกษา

สร้างโปรแกรมที่ใช้ภายในองค์กรโดยโปรแกรมมีความสามารถในการจัดการตารางนัดหมาย แจ้งเตือนกำหนดการนัดหมายต่างๆ และจัดการกระจายข่าวสารให้แก่บุคลากรที่เกี่ยวข้องภายในองค์กรนั้นๆ โปรแกรมนี้สามารถมีผู้ใช้งานตั้งแต่ 2 คนขึ้นไป ซึ่งโปรแกรมนั้นจะแบ่งออกเป็น 2 ฟังก์ชัน คือ ฟังก์ชันในเครื่องเซิร์ฟเวอร์ และฟังก์ชันในเครื่องไคลเอนต์ โดยทั้ง 2 ฟังก์ชันจะติดต่อกันโดยผ่านเครือข่ายอินเทอร์เน็ต ซึ่งเขียนด้วยภาษาจาวา โดยที่เครื่องในฝั่งเซิร์ฟเวอร์นั้นจะใช้ระบบปฏิบัติการวินโดวส์ 2000 เซิร์ฟเวอร์ และเครื่องในฝั่งไคลเอนต์จะใช้ระบบปฏิบัติการวินโดวส์ 98 ซึ่งจะติดต่อกันโดยใช้โปรโตคอล TCP/IP โดยโปรแกรมจะถูกออกแบบ และเขียนโปรแกรมโดยวิธีคอมโพเนนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของการศึกษา

1.4.1 ออกแบบและสร้างคอมโพเนนต์เพื่อช่วยในการกระจายข้อมูลข่าวสารภายในองค์กร

1.4.2 ออกแบบและสร้างคอมโพเนนต์ที่มีความสามารถในการจัดตารางเวลาและจัดการกำหนดการนัดหมายต่างๆ

1.4.3 ออกแบบและสร้างคอมโพเนนต์ที่สามารถแจ้งเตือนกำหนดการต่างๆที่มีการกำหนดไว้ในตารางนัดหมาย

1.4.4 นำคอมโพเนนต์ที่ได้ออกแบบไว้มาจัดทำโปรแกรม

1.5 ขั้นตอนของการศึกษา

1.5.1 ศึกษาปัญหาและรวบรวมข้อมูลเกี่ยวกับปัญหา

1.5.2 ศึกษาโปรแกรมที่เกี่ยวข้องกับการแก้ปัญหา

1.5.3 ออกแบบโปรแกรม

1.5.4 พัฒนาโปรแกรม

1.5.5 ทดสอบ และแก้ไขโปรแกรม

1.5.6 ประเมินผลงาน

1.5.7 ทำรายงาน

1.5.8 ส่งผลงานและนำเสนอผลงาน

ตารางเวลา

เดือนวันที่	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
มิถุนายน																															
กรกฎาคม																															
สิงหาคม																															
กันยายน																															
ตุลาคม																															
พฤศจิกายน																															
ธันวาคม																															
มกราคม																															
กุมภาพันธ์																															
มีนาคม																															

- = ศึกษาหาข้อมูลที่จะทำโปรเจกต์และยื่นเรื่องเสนอหัวข้อโปรเจกต์
- = ศึกษาโปรแกรมที่เกี่ยวข้องกับปัญหา
- = ออกแบบโปรแกรมให้อยู่บนพื้นฐาน component based
- = ทำการพัฒนาโปรแกรม
- = ทดสอบ และแก้ไขโปรแกรม
- = ประเมินผลงาน
- = ทำรายงาน
- = ส่งผลงานและนำเสนอผลงาน

ตารางที่ 1.1 แสดงตารางการทำงานตามขั้นตอนของการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ข้อตกลงเบื้องต้น

1.6.1 ใช้คอมไพเลอร์ภาษาจาวา (JDK 1.4 ขึ้นไป)

1.6.2 ใช้โปรโตคอล TCP/IP

1.6.3 องค์ประกอบจำเป็นสำหรับเซิร์ฟเวอร์

- ระบบปฏิบัติการวินโดวส์ 2000 เซิร์ฟเวอร์
- หน่วยประมวลผลกลาง(CPU) ความเร็วอย่างน้อย 700 เมกะเฮิร์ตซ์
- หน่วยความจำหลัก อย่างน้อย 128 เมกะไบต์
- หน่วยความจำสำรอง ซึ่งมีขนาดอย่างน้อย 10 กิกะไบต์
- ซีดีรอม ความเร็วไม่ต่ำกว่า 32x
- การ์ดแลน

1.6.4 องค์ประกอบจำเป็นสำหรับไคลเอนต์

- ระบบปฏิบัติการวินโดวส์ 98
- หน่วยประมวลผลกลาง (CPU) ความเร็วอย่างน้อย 350 เมกะเฮิร์ตซ์
- หน่วยความจำหลัก อย่างน้อย 128 เมกะไบต์
- หน่วยความจำสำรอง ซึ่งมีขนาดอย่างน้อย 5 กิกะไบต์
- ซีดีรอม ความเร็วไม่ต่ำกว่า 32x
- การ์ดแลน

1.7 ข้อจำกัดของการศึกษา

โปรแกรมที่ใช้ช่วยจัดการข้อมูลและนัดหมายต่างๆ ในองค์กรเป็นโปรแกรมที่มีขนาดใหญ่ ซึ่งประกอบด้วยฟังก์ชันในการทำงานเป็นจำนวนมาก เนื่องจากข้อจำกัดด้านระยะเวลาของการทำ ปัญหาพิเศษจึงเลือกเฉพาะส่วนที่มีความสำคัญที่สุด คือ การกระจายข้อมูลข่าวสาร และการจัดตารางเวลานัดหมาย

1.8 คำจำกัดความที่ใช้ในการศึกษา

Group คือ กลุ่มของบุคคลที่อยู่ในกลุ่มงานเดียวกัน หรือมีความสัมพันธ์กัน

Component คือ ซอฟต์แวร์เล็กๆที่เป็นอิสระจากกัน ซึ่งถูกพัฒนาและทดลองโดยแยกแต่ละคอมโพเนนต์ออกเป็นส่วนๆ ไม่ขึ้นกับโปรแกรมใดๆ และสามารถนำไปประกอบกับส่วนอื่นๆได้ง่าย

Groupware คือ ซอฟต์แวร์ที่สนับสนุนการทำงานเป็นกลุ่ม โดยมีพื้นฐานมาจากเทคโนโลยีหลัก ๆ จากการจัดการเอกสารมัลติมีเดีย(Multimedia Documents) การไหลของข้อมูล(Workflow) จดหมายอิเล็กทรอนิกส์(Electronics Mail) การจัดการประชุม(Conference) และการจัดการตารางเวลานัดหมาย(Schedule)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

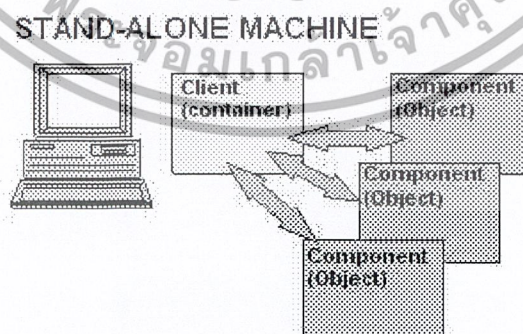
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 คอมโพเนนต์ (Component)

คอมโพเนนต์ (Components) หรือ ซอฟต์แวร์คอมโพเนนต์ (Software Components) คือ ซอฟต์แวร์เล็กๆที่เป็นอิสระจากกัน ซึ่งถูกพัฒนาและทดลองโดยแยกแต่ละคอมโพเนนต์ออกเป็น ส่วนๆ โดยแต่ละคอมโพเนนต์จะไม่ขึ้นกับโปรแกรมใดๆ และสามารถนำไปประกอบกับส่วนอื่นๆ ได้ง่าย หรืออาจมองได้ว่าคอมโพเนนต์เปรียบเสมือนออบเจกต์ (Object) ซึ่งถูกออกแบบเป็นอย่างดี เพื่อที่จะนำไปใช้กับส่วนอื่นๆ ได้ต่อ ซึ่งนี่คือข้อดีข้อหนึ่งของคอมโพเนนต์

คอมโพเนนต์สามารถทำงานได้ในระบบเครือข่าย และระบบปฏิบัติการที่แตกต่างกัน โดยที่ แต่ละคอมโพเนนต์จะเปรียบเสมือนส่วนประกอบต่างๆ ที่มาประกอบรวมกันจนเกิดโปรแกรม ขนาดใหญ่ ซึ่งแต่ละคอมโพเนนต์จะทำงานเหมือนกล่องดำ (Black-Box) หมายความว่าสามารถนำ คอมโพเนนต์นั้นๆ ไปใช้งานได้ โดยที่ไม่จำเป็นต้องเข้าใจถึงวิธีการทำงานของคอมโพเนนต์ รู้แต่ เพียงว่าใช้คอมโพเนนต์นั้นอย่างไร และคอมโพเนนต์นั้นให้ผลอะไรออกมาเท่านั้น

คอมโพเนนต์อาจจะมีขนาดใหญ่หรือเล็ก และอาจถูกพัฒนาโดยโปรแกรมเมอร์หลายๆคน ซึ่ง ใช้เครื่องมือในการพัฒนาหรือระบบปฏิบัติการที่แตกต่างกัน โดยคอมโพเนนต์สามารถทำงานได้ทั้ง แบบ Stand-alone Machine (ดังตัวอย่างรูปที่ 2.1) หรือสามารถนำคอมโพเนนต์ไปใช้ในระบบ เครือข่าย เช่น ระบบเครือข่ายแบบแลน (LAN) ระบบเครือข่ายแบบอินทราเน็ต (Intranet) หรือ ระบบเครือข่ายแบบอินเทอร์เน็ต (Internet) ก็ได้



รูปที่ 2.1 แสดง Stand-alone Machine

2.1.1 คุณสมบัติของคอมโพเนนต์

2.1.1.1 เป็นคอมโพเนนต์ที่ผู้ใช้งานต้องการ เช่น คอมโพเนนต์นั้นๆ มีความต้องการของผู้ใช้สูง หรือผู้ใช้เรียกใช้งานคอมโพเนนต์นั้นๆ มาโดยเฉพาะ

2.1.1.2 สร้างจากออบเจกต์ (Objects) หรือ Elements ต่างๆ และอาจจะเป็นโมดูล(Module) เนื่องจากแต่ละคอมโพเนนต์ควรจะเป็นอิสระจากกัน ทำให้สามารถปรับปรุง แก้ไข หรือพัฒนาคอมโพเนนต์นั้นๆ ได้โดยง่าย

2.1.1.3 สามารถนำคอมโพเนนต์ต่างๆ มาประกอบกันจนเกิดเป็นโปรแกรมขนาดใหญ่ซึ่งโปรแกรมขนาดใหญ่ที่เกิดขึ้นนั้นสามารถทำงานได้หลากหลาย

2.1.1.4 สามารถติดตั้งร่วมกับคอมโพเนนต์อื่นๆ ได้ โดยไม่เกิดผลกระทบกับส่วนอื่นๆ (Plug-and-Play)

2.1.1.5 ใช้งานง่าย มีลักษณะการใช้งานเหมือนหรือใกล้เคียงกับโปรแกรมต้นแบบทำให้สามารถศึกษาและทำความเข้าใจได้ง่าย

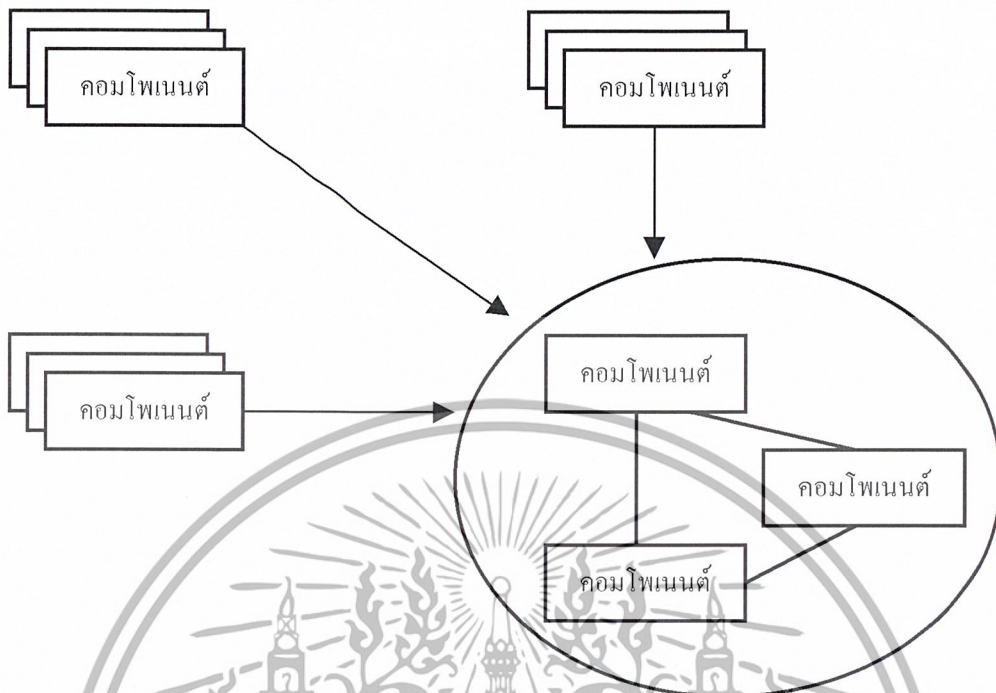
2.1.1.6 สามารถใช้งานร่วมกับออบเจกต์ (Objects) อื่นๆ ได้

2.1.1.7 มีความยืดหยุ่นสูง สามารถปรับปรุง แก้ไข หรือพัฒนาคอมโพเนนต์นั้นๆ ได้โดยง่าย

2.1.2 ข้อดีของคอมโพเนนต์

2.1.2.1 ผู้ใช้ (End-User) เนื่องจากคอมโพเนนต์จะมีลักษณะเป็นแบบ Dynamic ผู้ใช้จึงสามารถเปลี่ยนคอมโพเนนต์ที่ใช้ได้ ยกตัวอย่างเช่น การเลือกใช้โปรแกรม HTML Editor หรือโปรแกรมรับ-ส่งจดหมายทางอินเทอร์เน็ต(E-mail) ใน Microsoft Internet Explorer ว่าต้องการจะใช้โปรแกรมใดในการจัดการ

2.1.2.2 ผู้พัฒนา (Programmer) ผู้พัฒนาสามารถวางโครงสร้างของโปรแกรมให้เป็นอิสระจากกันได้ โดยสามารถที่จะเพิ่มคอมโพเนนต์ต่างๆ หรือเปลี่ยนคอมโพเนนต์ที่ใช้ได้ (ดังตัวอย่างรูปที่ 2.2) ในกรณีที่มีการพัฒนาคอมโพเนนต์นั้นๆ ขึ้นมาในภายหลัง ยกตัวอย่างเช่น การเปลี่ยนคอมโพเนนต์ที่ใช้ในการจัดเรียงข้อมูลจาก Bubble Sort มาเป็นคอมโพเนนต์ที่มีการจัดเรียงข้อมูลแบบ Quick Sort เป็นต้น



รูปที่ 2.2 แสดงลักษณะการประกอบของคอมพิวเตอร์

2.2 กรุปแวร์ (Groupware)

กรุปแวร์ (Groupware) คือ ระบบคอมพิวเตอร์ หรือ ซอฟต์แวร์ที่สนับสนุนการทำงานเป็นกลุ่ม เพื่ออำนวยความสะดวกในการแลกเปลี่ยนข้อมูล การใช้ข้อมูลร่วมกัน การจัดการตารางเวลานัดหมาย เป็นต้น โดยจัดให้มีอินเทอร์เน็ตในการใช้สภาพแวดล้อมร่วมกัน ซึ่งกรุปแวร์ได้รวบรวมเอาเทคโนโลยีที่สนับสนุนการทำงานไว้ดังนี้

1. การจัดการเอกสารมัลติมีเดีย (Multimedia Documents)
2. การไหลของข้อมูล (Workflow)
3. จดหมายอิเล็กทรอนิกส์ (E-mail)
4. การประชุม (Conferencing)
5. การจัดการเวลา (Scheduling)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ประเภทของระบบ 2 ระบบที่ไม่นับเป็นกรุปแวร์ คือ

2.2.1.1. Stand-alone personal computers ในกรณีนี้ปัญหาที่เกิดขึ้นกับผู้ใช้งานจะเป็นปัญหาที่เกิดขึ้นกับผู้ใช้นั้นเอง ดังนั้นขอบเขตในการแก้ไขปัญหาก็จะมีรูปแบบหนึ่งเท่านั้น จากการที่มีผู้ใช้งานจำนวนหนึ่งได้พยายามค้นหาระบบที่สามารถติดต่อสื่อสารกันได้กับบุคคลอื่น เพื่อให้สามารถทำงานร่วมกันได้ และยังใช้ข้อมูลร่วมกันได้ด้วย

2.2.1.2. Mainframe ในที่นี้คือคลังข้อมูลที่มีองค์กรเป็นเจ้าของนั่นเอง ซึ่งข้อมูลเหล่านั้นจะถูกใช้ให้เกิดประโยชน์จากกลุ่มคนในองค์กรนั้นๆ ดังนั้นควรจะมีการจัดการและการวางแผนที่ดี รวมถึงต้องจัดเก็บข้อมูล และป้องกันข้อมูลไม่ให้ผู้ที่ไม่ได้รับอนุญาตสามารถดูข้อมูลได้ ดังนั้นการเข้าใช้ข้อมูลของบุคคลที่ได้รับสิทธิ์

ในทางกลับกันอาจกล่าวได้ว่า personal computing ไม่ใช่กรุปแวร์ แต่มันจะวิวัฒนาการขึ้นไปเป็นรูปแบบที่หลากหลายของ interpersonal computing โดยสามารถให้ผู้ใช้งานติดต่อและทำงานร่วมกันกับผู้ใช้งานอื่น

2.2.2 ความสามารถของเทคโนโลยี

ข้อดีของเทคโนโลยีที่ทำให้มีการพัฒนารูปแบบไปลักษณะที่อยู่ในศูนย์กลาง มีอยู่ด้วยกัน 2 ข้อ คือ เน็ตเวิร์คที่มีความเร็วสูง และระบบการกระจายข้อมูล

การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคล หรือเรียกว่า PC LAN ได้พัฒนาขึ้นโดยสามารถให้ผู้ใช้งานเข้าใช้งาน และสนับสนุนการใช้งานในสภาพแวดล้อมร่วมกัน เช่น ให้ผู้ใช้งานสามารถใช้ที่เก็บข้อมูลร่วมกัน รับและส่งจดหมายอิเล็กทรอนิกส์ และแลกเปลี่ยนข้อมูลระหว่างกัน รวมทั้งการใช้ตารางเวลาร่วมกัน เป็นต้น

การประชุมผ่านคอมพิวเตอร์ และการประกาศข่าวผ่านกระดานข่าวอิเล็กทรอนิกส์นั้นเป็นยุคแรกของแอปพลิเคชัน ซึ่งนิยมใช้ในวงการการศึกษาและนักเทคนิคมากกว่า 10 ปีแล้ว ปัจจุบันนี้สามารถนำมาประยุกต์ใช้กับองค์กรธุรกิจได้ จึงเป็นไปได้ที่กลุ่มคนสามารถร่วมการสนทนา และประชุมที่ใดและเวลาใดก็ได้

ในรูปแบบของการเชื่อมต่อของคอมพิวเตอร์ส่วนบุคคลและ Workstation ในยุคแรกนั้นใช้เพื่อให้สามารถใช้ทรัพยากรที่มีราคาแพงร่วมกันได้โดยผ่านไฟล์ หรือ print server เป็นต้น ในขณะที่การควบคุมการทำงานอยู่บนอุปกรณ์ตั้งโต๊ะ

แต่ในปัจจุบันนี้ได้มีการตระหนักถึงเครื่องมือช่วยที่สำคัญ เช่น การรวมระบบการทำงานเอกสาร จดหมายอิเล็กทรอนิกส์ การประชุมผ่านคอมพิวเตอร์ และอุปกรณ์เชื่อมต่อเน็ตเวิร์คทั้งหมดนี้ได้ถูกออกแบบมาเพื่อให้ประสิทธิภาพในการทำงานของแต่ละคนเพิ่มมากขึ้น โดยจัดให้มีการเข้าใช้ข้อมูลแต่ละประเภทได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ส่วนประกอบของกรู๊ปแวร์

จากข้างต้นได้กล่าวมาแล้วว่า กรู๊ปแวร์ พัฒนาโดยมีพื้นฐานมาจากเทคโนโลยีหลักๆ 5 ประการ คือ การจัดการเอกสารมัลติมีเดีย การไหลของข้อมูล อีเมล การประชุม และการจัดการตารางเวลา ซึ่งกรู๊ปแวร์ได้รวมเอาเทคโนโลยีและสร้างการทำงานร่วมกันขึ้นมาใหม่ โดยที่เทคโนโลยีสำหรับการจัดการเอกสารมัลติมีเดีย และการไหลของข้อมูลได้มาจากระบบเอกสารรูปภาพที่เป็นอิเล็กทรอนิกส์ ส่วนอีเมลและการจัดการตารางเวลานั้นมาจากการทำงานที่เป็นอัตโนมัติ การประชุมนั้นสร้างขึ้นเพื่อกรู๊ปแวร์โดยเฉพาะ ซึ่งรายละเอียดของส่วนต่าง ๆ มีดังนี้

2.2.3.1 การจัดการเอกสารมัลติมีเดีย

เทคโนโลยีของการจัดการเอกสารมัลติมีเดียของกรู๊ปแวร์นั้น มีพื้นฐานมาจากรูปภาพอิเล็กทรอนิกส์ ซึ่งระบบของรูปภาพอิเล็กทรอนิกส์นี้ขึ้นอยู่กับ Database-oriented ซึ่งมีไคลเอนต์/เซิร์ฟเวอร์แอปพลิเคชัน โดยที่ไคลเอนต์จะทำการจับภาพ และจัดการกับภาพโดยทำหน้าที่ทางด้าน front-ends ที่เกี่ยวกับการเก็บข้อมูลในเซิร์ฟเวอร์เก็บรูปภาพ ซึ่งมีขั้นตอนการทำงาน ดังนี้

- 1) ใช้ scanner เป็นตัวจับภาพที่ต้องการเพื่อให้ได้ภาพที่เป็นดิจิทัล
- 2) หลังจากได้ภาพที่มีรูปแบบเป็นดิจิทัลแล้ว ก็ตรวจสอบคุณภาพของรูปนั้น และถ้ารูปนั้นยังไม่ดีพอก็ทำการ scan ภาพนั้นใหม่
- 3) ขณะที่ทำการแสดงภาพอยู่นั้น ข้อมูลที่ถูกดึงออกมาจะมีรูปแบบการจัดเก็บโดยอาจมีรูปแบบการจัดเก็บแบบง่าย ๆ โดยมีกำหนด index และสามารถอ้างอิงถึงข้อมูลนั้น ๆ หลังจากนั้นก็สามารถจัดเก็บข้อมูลได้
- 4) ซอฟต์แวร์หรือฮาร์ดแวร์ ที่ทำงานด้านการบีบอัดข้อมูลก็จะทำการบีบอัดข้อมูลรูปภาพ จากนั้นก็จะส่งข้อมูลไปยังเซิร์ฟเวอร์ที่ทำการจัดเก็บข้อมูล
- 5) ไคลเอนต์ยังคงสามารถเข้าถึงข้อมูลในเซิร์ฟเวอร์ โดยที่สามารถขอคุณภาพนั้นใหม่อีกครั้ง พิมพ์เอกสาร ส่ง fax ให้คำจำกัดความโดยการขีดเส้นใต้หรือมี electronic note ได้

ฝั่งเซิร์ฟเวอร์ที่ทำการจัดการเกี่ยวกับรูปภาพนั้นจะทำการให้มีการใช้ฐานข้อมูลรูปภาพร่วมกันโดยที่เซิร์ฟเวอร์ที่เก็บรูปภาพนั้นจะเก็บข้อมูลในฐานข้อมูล SQL ซึ่งตัวมันจะเป็นเอกสารที่เก็บในไฟล์เซิร์ฟเวอร์

Image server ยังจัดการเกี่ยวกับการไหลของข้อมูล การรักษาความปลอดภัย การจัดทำ image index (หรือ metadata) และมี pointer เป็นตัวอ้างอิงไปที่ file systems ที่ซึ่ง BLOBs จะจัดเก็บแบบ physical ซึ่งหน้าที่บางประการมีดังนี้

- 1) เซิร์ฟเวอร์จัดการเกี่ยวกับการจัดเก็บเอกสาร ซึ่งโพลเดอร์ที่เก็บเอกสารเหล่านั้น มี

รูปแบบเป็นชุดของหน้าเอกสารแบบมัลติมีเดีย

เอกสารนี้เป็นเอกสารที่ส่งวันไวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เมื่อมีการเคลื่อนย้ายเอกสาร จะมีการจัดการเกี่ยวกับการไหลข้อมูลแบบอัตโนมัติ โดยที่การเคลื่อนย้ายจากการทำงานด้านหนึ่งไปยังส่วนถัดไป จะเป็นไปตามที่กำหนดไว้ตามกฎหมายหรือทางเดินเอกสาร

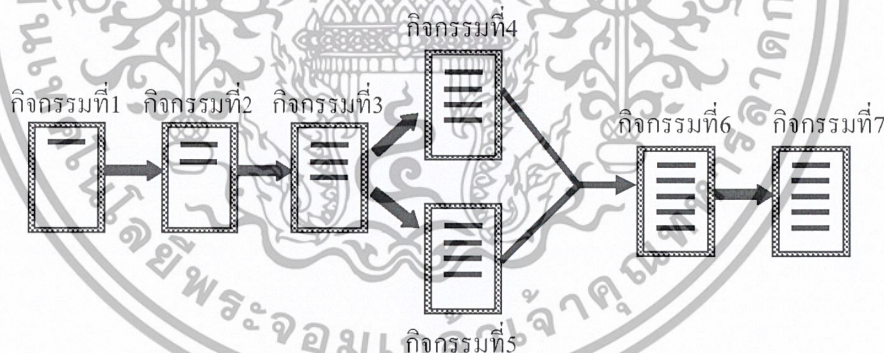
3) โครงสร้างของผู้ใช้และลำดับการใช้งานสามารถสร้างและบำรุงรักษาบนเซิร์ฟเวอร์สำหรับกลุ่มงานที่ได้รับงานนั้นโดยเฉพาะ

4) จัดให้มีการรายงานในส่วนของการคู่ค่าต่าง ๆ และประเภทของงานที่กำลังทำอยู่ในระบบ รวมไปถึงการบันทึกย้อนงานนั้น ๆ ได้

5) เอกสารจะถูกจัดเก็บในรูปแบบที่หลากหลาย

2.2.3.2 การไหลของข้อมูล

การไหลของข้อมูล คือ เทคโนโลยีคลเอนด์/เซิร์ฟเวอร์ ที่สามารถกำหนดเส้นทางอัตโนมัติจากกำหนดการหนึ่งไปยังกำหนดการถัดไป ในสภาพแวดล้อมของคลเอนด์/เซิร์ฟเวอร์ทั้งที่มีโครงสร้างแน่นอน และไม่มีโครงสร้าง ตัวอย่างของการไหลของข้อมูลแบบง่าย ๆ เช่น แม่น้ำก็จะมี การไหลของน้ำ เหมือนกับการไหลของงานจากส่วนหนึ่ง ไปยังอีกส่วนหนึ่งในระหว่างทางนั้นค่าที่ได้รับก็จะเพิ่มลงไปด้วย



รูปที่ 2.3 แสดงตัวอย่างการทำงานของ Workflow

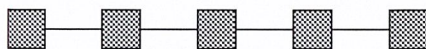
โครงสร้างของการไหลของข้อมูล ประกอบไปด้วยส่วนต่าง ๆ 3 ส่วนดังนี้

1. การกำหนดเส้นทาง จะเป็นการกำหนดเส้นทางการเดินทางที่วัตถุจะต้องเคลื่อนไป
2. กฎ จะกำหนดว่าข้อมูลอะไรที่ถูกกำหนดเส้นทางไว้ และใครเป็นเจ้าของข้อมูลนั้น กฎต่าง ๆ จะถูกกำหนดโดยเงื่อนไขของการไหลของข้อมูลและจะต้องพบกับสถานะถัดไป รวมทั้งการวิธีการจัดการ
3. บทบาท จะเป็นการกำหนดหน้าที่ของงานไว้ว่า บุคคลไหนที่จะมาทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

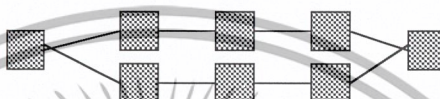
การกำหนดเส้นทางของการไหลของข้อมูล มีรูปแบบต่าง ๆ ดังนี้

1. แบบเรียงตามลำดับ การเคลื่อนย้ายงานจากส่วนหนึ่ง ไปอีกส่วนหนึ่งตามลำดับการทำงาน



รูปที่ 2.4 แสดงการไหลของข้อมูลแบบเรียงตามลำดับ

2. แบบคู่ขนาน แบบนี้จะมีทางให้เลือกทำงานได้



รูปที่ 2.5 แสดงการไหลของข้อมูลแบบคู่ขนาน

3. แบบมีการย้อนกลับของงาน ยกตัวอย่างเช่น การกลับมาทำงานอันเก่าที่ทำมาแล้ว

ใหม่



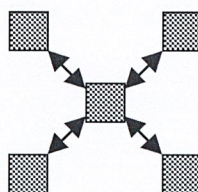
รูปที่ 2.6 แสดงการไหลของข้อมูลแบบมีการย้อนกลับของงาน

4. แบบวงกลม



รูปที่ 2.7 แสดงการไหลของข้อมูลแบบวงกลม

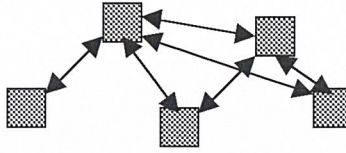
5. แบบวงล้อ



รูปที่ 2.8 แสดงการไหลของข้อมูลแบบวงล้อ

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. แบบการเชื่อมต่ออย่างสมบูรณ์



รูปที่ 2.9 แสดงการไหลของข้อมูลแบบการเชื่อมต่ออย่างสมบูรณ์

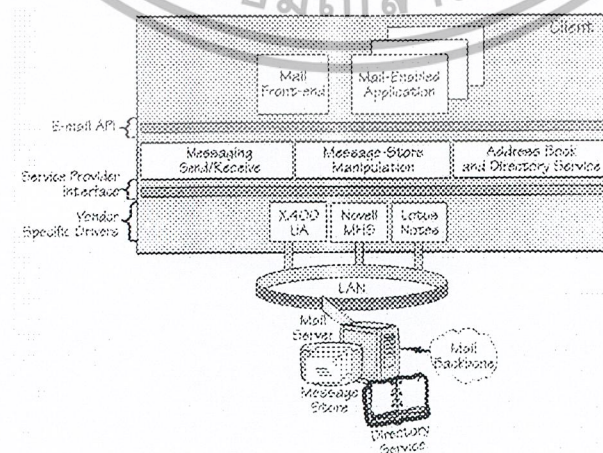
2.2.3.3 จดหมายอิเล็กทรอนิกส์

กรุ๊ปแวร์ได้เสนอให้มีจดหมายอิเล็กทรอนิกส์ขึ้น เพื่อช่วยสนับสนุนการทำงานกันเป็นกลุ่ม เหตุผลที่จดหมายอิเล็กทรอนิกส์มีความสำคัญคือการทำงานเป็นกลุ่ม เพราะว่าจดหมายอิเล็กทรอนิกส์เป็นอีกวิธีหนึ่งที่สามารถทำให้ผู้ทำงานร่วมกันสามารถติดต่อสื่อสารกันได้ โดยที่ไม่ต้องทำการติดต่อกับอีกคนหนึ่งแบบ Real time ส่วนผู้รับจดหมายอิเล็กทรอนิกส์นั้นก็ไม่ต้องตอบสนองผู้ส่งจนกว่าจะได้ทำการอ่านจดหมายอิเล็กทรอนิกส์นั้นแล้ว ดังนั้นจึงนับว่าเป็นวิธีที่ง่ายที่สุดในการติดต่อสื่อสารแบบอิเล็กทรอนิกส์

ก่อนจดหมายอิเล็กทรอนิกส์จะเชื่อมต่อกับ mail backbone และ gateway การทำงานของจดหมายอิเล็กทรอนิกส์ในส่วน front-end นั้นไม่ใช่แค่วิธีการส่งข้อความเท่านั้น Mail APIs เช่น VIM และ MAPI ถูกออกแบบมาเพื่อให้ประยุกต์ใช้ในการเป็นพื้นฐานโครงสร้างของการส่งจดหมาย

อีเมล APIs : VIM, MAPI และ CMC

อีเมล APIs ให้เราเขียนโปรแกรมเพื่อใช้ในการเข้าถึงส่วนที่เป็นการบริการการส่งเมล mail directory และการเก็บข้อความ



รูปที่ 2.10 แสดง APIs สำหรับอีเมล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมล API พื้นฐาน ได้แก่

1. Simple messaging service เป็นหน้าที่การทำงานสำหรับการกำหนดที่อยู่ การส่งจดหมาย และการรับจดหมาย รวมทั้งไฟล์ที่แนบมาด้วย โดยจะเสนอทางเลือกให้ผู้ใช้งานโดยจัดให้มีการ log on กำหนดที่อยู่ของจดหมาย และส่วนของข้อความจดหมาย

2. Message store manipulation ในส่วนนี้จะรวมการเปิดและอ่านข้อความที่ถูกส่งมายังส่วนที่เก็บข้อมูล การบันทึกข้อความ การลบข้อความ การนำข้อความไปเก็บในส่วนที่เก็บข้อความ การค้นหาข้อความ การย้ายข้อความ

3. Address book and directory services จะเป็นการเก็บรวบรวมรายชื่อผู้รับข้อความเฉพาะคน หรือผู้รับที่เป็นกลุ่ม และเราสามารถเพิ่มหรือลบกลุ่มคนหรือสมาชิกจาก Address book ได้และสามารถค้นหารายชื่อได้

4. Mail Object Manipulation API ขอมให้เราเข้าไปในส่วนของการคอมโพเนนต์ย่อยๆ ของจดหมายนั้น ยกตัวอย่างเช่น ข้อความจะประกอบไปด้วยส่วนที่เป็นส่วนหัวของจดหมาย และส่วนที่เป็นข้อมูลของจดหมาย

5. Authentication and Security Services จะเป็นการขอมให้มีการ log on และทำการตรวจสอบความถูกต้องของผู้ใช้ที่ได้รับสิทธิ์

6. Service provider interface ในส่วนนี้จะจัดให้มีการเสริมการทำงานจาก Back-end ไป Front-end ในส่วนนี้จัดให้มีอินเทอร์เฟซที่สร้างขึ้นมา เพื่อเปิดสภาพแวดล้อมสำหรับการจัดทำ mail service

นอกจากนี้ยังมีการเพิ่ม VIM MAPI และ CMC ขึ้นมาอีก ดังมีรายละเอียด ดังนี้

1. Vendor Independent Messaging (VIM) เป็นอินเทอร์เฟซที่ทำให้เชื่อมส่วนพื้นหลังเข้าด้วยกัน ได้แก่ Lotus, Apple, IBM, Borland, MCI, Oracle, WordPerfect, และ Novell ส่วน VIM นั้นได้รับการออกแบบมาจากกลุ่มของผู้ที่ทำงานข้ามแพลตฟอร์มกัน โดยที่ VIM ประกอบไปด้วย 55 API ซึ่งมี 10 ทางเลือกที่สนับสนุนการส่งจดหมาย การเก็บข้อมูล และส่วนของ Address book services นอกจากนี้ VIM ยังจัดให้มี Simple Mail Interface(SMI) ซึ่งประกอบไปด้วย SMISendMail และ SMISendDocuments

2. Messaging API (MAPI) เป็นส่วนที่ Microsoft's WOSA เสนอให้ใช้สำหรับจดหมายอิเล็กทรอนิกส์ MAPI พื้นฐานจะประกอบไปด้วย 12 API ที่จัดการเกี่ยวกับการส่งเมล การเก็บข้อความ และส่วนของ Address book services

3. Common Mail Call (CMC) เป็นอินเทอร์เฟซของ X.400 โดยที่จะมี 10 API ซึ่งเป็นส่วนหนึ่งของ VIM และ MAPI โดยที่ CMC จะเป็นการจัดการเฉพาะส่วนกับเมลเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.4 การจัดการตารางเวลาและปฏิทิน

การจัดการตารางเวลานี้เป็นเทคโนโลยีของกรู๊ปแวร์ โดยที่ได้ออกแบบการจัดการตารางเวลาเพื่อรวมกลุ่มของหน้าที่หลัก ๆ ของไคลเอนต์/เซิร์ฟเวอร์ เช่น ถ้าเรามีการจัดการตารางเวลาและปฏิทินการทำงานแล้วนำมาใช้กับการจัดการการไหลของข้อมูล จดหมายอิเล็กทรอนิกส์ และการจัดการเอกสารแบบมัลติมีเดีย เราก็จะสามารถเพิ่มการประชุมในปฏิทินงานในส่วนของผู้รับเอกสารนั้น ๆ สามารถจัดตารางห้องประชุม และมีทำการแจ้งเตือนไปให้ผู้รับโดยอัตโนมัติได้ ยกตัวอย่างเช่น Groupware blade ซึ่งการทำงานของแอปพลิเคชันที่ทำงานบน Browser โดยมีการทำงาน ดังนี้

เมื่อทำการเปิดหน้าแดรกร์ชีตารางเวลานัดหมายแล้ว หน้าจอจะทำการแสดงปฏิทินที่แสดงผลเป็นรายเดือน โดยจะเน้นวันที่แสดงผลและวันที่มีการนัดหมายหรือมีเหตุการณ์ที่กำหนดไว้ จากนั้นทำการคลิกที่วันที่ต้องการดูตารางการทำงานสำหรับวันนั้น เราสามารถเลือกกลุ่มที่ต้องการดูกำหนดการนั้นได้โดยเลือกจากรายชื่อกลุ่มที่แสดงเป็นแบบ Drop down

การกำหนดรูปแบบการดูตารางเวลา ทำได้โดยที่

- 1) การแสดงผลประจำวัน ทำการแสดงตารางเวลาการทำงานทุกชั่วโมงสำหรับวันนั้น ๆ
- 2) การแสดงผลประจำสัปดาห์ ทำการแสดงเหตุการณ์ที่เกิดขึ้นในสัปดาห์นั้น ๆ
- 3) การแสดงผลประจำเดือน ทำการแสดงเหตุการณ์ที่เกิดขึ้นประจำเดือน
- 4) การแสดงผลแบบปฏิทิน ทำการแสดงปฏิทินที่เป็นรายเดือน
- 5) การแสดงผลแบบปี ทำการเน้นตรงวันที่ที่มีเหตุการณ์หรือตารางเวลาอยู่
- 6) การแสดงเป็นรายการ ทำการแสดงรายการที่เกิดขึ้นตามวัน

การเพิ่มกำหนดการทำได้ดังนี้

- 1) กำหนดหัวข้อที่ต้องการ
- 2) มีรูปแบบให้เลือกไม่ว่าคุณต้องการที่จะลงทะเบียนเวลาเริ่มหรือเวลาสิ้นสุดงานหรือไม่ว่างานนี้จะต้องกลับมาทำอีก
- 3) เพิ่มเหตุการณ์ที่ต้องทำ และวันเวลาที่เริ่มทำงานและสิ้นสุดการทำงาน
- 4) เลือกกลุ่มที่ต้องการทำงาน
- 5) จากนั้นกดปุ่ม Add เพื่อทำการเพิ่มกำหนดการทำงานลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไขและลบกำหนดการ ทำได้ดังนี้

- 1) การแก้ไขทำได้โดยการเลื่อนงานจากหน้าจรรงสีที่แสดงกำหนดการ จากนั้นกดปุ่ม Edit แล้วหน้าจอจะปรากฏข้อมูล แล้วทำการแก้ไขและกดปุ่ม Update
- 2) การกดปุ่ม Delete ทำเพื่อการลบงานที่เลือกออกจากรายการหมายกำหนดการ

การบำรุงรักษา

- 1) การบำรุงรักษานี้อนุญาตให้เรลบหมายกำหนดการที่ได้ผ่านมาแล้วได้ ในส่วนนี้จะกำหนดวันที่ที่ต้องการลบทุก ๆ ช่วงเหตุการณ์
- 2) กดปุ่ม Preview เพื่อทำการดูเหตุการณ์ในวันที่ได้เลือกก่อนที่จะทำการลบจริง

2.2.3.5 การประชุม

การประชุมหรือเรียกอีกอย่างว่า Electronic meeting นั้นมี 2 ประเภทด้วยกัน คือแบบ Realtime และแบบ Anytime โดยมีรายละเอียดของแต่ละประเภท ดังนี้

- 1) การประชุมแบบ Realtime การประชุมแบบนี้จะยอมให้กลุ่มผู้ทำงานร่วมกันทำงานโดยใช้เอกสารอิเล็กทรอนิกส์ ไวท์บอร์ด การใช้สีที่แตกต่างกันของ cursors ของผู้ใช้แต่ละคน และมีการกำหนดผู้เข้าร่วมประชุมเพื่อควบคุมการใช้เอกสารร่วมกัน ผู้เข้าประชุมสามารถพูดผ่านไมโครโฟนของแต่ละคน และสามารถมองเห็นผู้เข้าร่วมประชุมคนอื่นผ่านหน้าจอที่แสดงเป็น Video แต่ถ้าเรามีอัตราการส่งข้อมูลน้อยเราสามารถจัดการประชุมโดยใช้ movie clip แทน
- 2) การประชุมแบบ Anytime การประชุมแบบนี้จะอนุญาตให้ผู้มีส่วนร่วมประชุมสามารถแสดงความคิดเห็นเวลาใดก็ได้ที่ต้องการ และสามารถออกไปเมื่อไรก็ได้

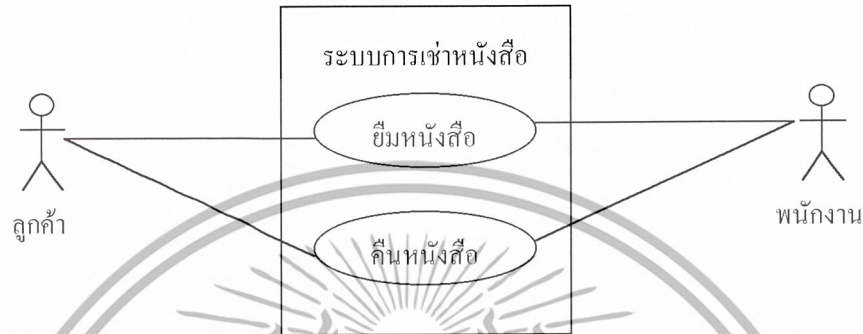
2.3 ยูเอ็มแอล (UML, Unified Modeling Language)

ยูเอ็มแอล เป็นภาษาสัญลักษณ์รูปภาพ สำหรับใช้ในการสร้างโมเดลเชิงวัตถุ ยูเอ็มแอลมีลักษณะของเมต้าโมเดล (metamodel) คือเป็นโมเดลที่เอาไว้เพื่ออธิบายโมเดลอื่นๆ อีกที สำหรับระบบงานของการพัฒนาซอฟต์แวร์นั้น จำเป็นที่ต้องมีภาษาที่สามารถอธิบายสถาปัตยกรรมของระบบในมุมมองต่างๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 ประเภทไดอะแกรมของ UML

2.3.1.1 ยูสเคสไดอะแกรม (Use Case Diagram) ยูสเคสไดอะแกรมใช้แสดงความสัมพันธ์ระหว่าง แอ็กเตอร์และยูสเคส ข้อดีของยูสเคสไดอะแกรมคือเราจะเห็นได้อย่างชัดเจนว่าขอบเขตของระบบที่เรากำลังสนใจมีอยู่แค่ไหน



รูปที่ 2.11 แสดงยูสเคสไดอะแกรม

1) สัญลักษณ์ที่ใช้ในยูสเคสไดอะแกรม

- ระบบ (Systems) สิ่งที่เราพัฒนาเรียกว่า ระบบ งานที่เป็นระบบนั้นไม่จำเป็นต้องเป็นงานที่เกี่ยวกับคอมพิวเตอร์เสมอไป ระบบในยูสเคสไดอะแกรมจะถูกแทนด้วยรูปกล่องสี่เหลี่ยม ซึ่งมีชื่อของระบบเขียนอยู่ข้างในกล่องสี่เหลี่ยม

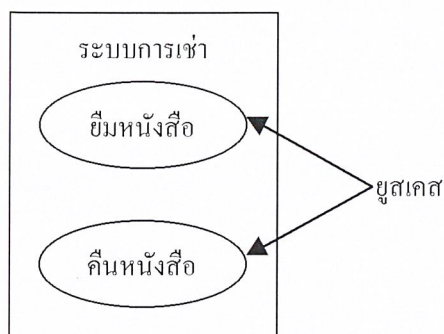


รูปที่ 2.12 แสดงสัญลักษณ์ของระบบ

- ยูสเคส (Use Case) คือตัวระบบที่เรากำลังสนใจอยู่ซึ่งจะบอกว่าระบบจะทำอะไร ยูสเคสแทนด้วยรูปวงรี มีชื่อยูสเคสอยู่ข้างใน และทุกยูสเคสจะอยู่ในกรอบสี่เหลี่ยม(ระบบ)

คุณสมบัติของยูสเคส มีดังนี้

1. ต้องถูกกระทำโดยแอ็กเตอร์
2. รับข้อมูลจากแอ็กเตอร์ และส่งข้อมูลให้แอ็กเตอร์



รูปที่ 2.13 แสดงสัญลักษณ์ของยูสเคส

- แอ็กเตอร์ (Actor) คือผู้ที่กระทำกับระบบ โดยจะเป็นคนหรือไม่ก็ได้ นั่นคือแอ็กเตอร์เป็นผู้ที่ทำให้เกิดเหตุการณ์ แอ็กเตอร์จะเป็นส่วนที่อยู่นอกระบบ (ระบบควบคุมไม่ได้) และสามารถทำการส่งข้อมูล หรือรับข้อมูล หรือแลกเปลี่ยนข้อมูลข่าวสารกับระบบ นอกจากนี้ระบบยังสามารถทำตัวเป็นแอ็กเตอร์ได้อีกด้วย เช่นกรณีที่เชื่อมต่อกับอีกระบบหนึ่งภายนอก แอ็กเตอร์แทนด้วยรูปคน ทุกแอ็กเตอร์จะต้องกระทำกับยูสเคส อย่างน้อยหนึ่งยูสเคสเสมอ

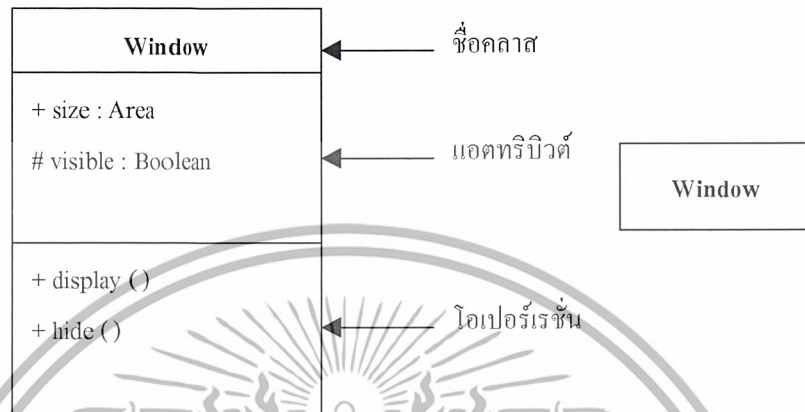
รูปที่ 2.14 แสดงสัญลักษณ์ของแอ็กเตอร์

- เส้นแสดงความสัมพันธ์ (Relationship) มีความสัมพันธ์มากมายระหว่างแต่ละยูสเคส หรือระหว่างแอ็กเตอร์และยูสเคส ดังนี้

1. **Association** จะมีการกำหนดถึงบทบาทของแต่ละแอ็กเตอร์ในยูสเคสที่มีความสัมพันธ์ร่วมกัน
2. **Extends** การที่นำเอายูสเคสเดิมที่มีอยู่แล้วมาเพิ่มการทำงานบางอย่าง
3. **Generalization** เป็นการถ่ายทอดคุณสมบัติ หรือพฤติกรรมบางอย่างจากยูสเคสหนึ่งไปยังอีกยูสเคสหนึ่ง หรือจากแอ็กเตอร์หนึ่งไปยังอีกแอ็กเตอร์หนึ่ง
4. **Include/Uses** ยูสเคสที่ถูกยูสเคสอื่นๆ เรียกใช้งานมากกว่าหนึ่งยูสเคสขึ้นไป ซึ่งยูสเคสหนึ่งๆ อาจจำเป็นต้องอาศัยการทำงานของยูสเคสอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2 คลาสไดอะแกรม (Class Diagram) เป็นไดอะแกรมที่ใช้อธิบายความสัมพันธ์ระหว่างคลาส คลาสเป็นการนำเอากลุ่มของออบเจ็กต์มาอธิบายความหมาย การกำหนดคลาสจะแทนด้วยสัญลักษณ์รูปสี่เหลี่ยมผืนผ้า โดยแบ่งเป็น 3 ส่วน



รูปที่ 2.15 แสดงสัญลักษณ์ของคลาส

- 1) ชื่อคลาส จะขึ้นต้นด้วยตัวอักษรตัวใหญ่แบบหนา และจะเอียงในกรณีที่เป็น *Abstract Class*
- 2) แอตทริบิวต์ (Attribute) เป็นการบอกถึงคุณสมบัติของคลาส ประกอบด้วยองค์ประกอบย่อย ดังนี้
 - ชนิดของการเข้าถึง
 1. Public(+) วัตถุทั้งหมดภายในระบบสามารถเข้าถึงได้
 2. Private(-) วัตถุที่สามารถเข้าถึงต้องเป็นวัตถุที่อยู่ในคลาส หรือสับคลาสเดียวกัน
 3. Protect(#) วัตถุที่สามารถเข้าถึงจะต้องเป็นวัตถุที่อยู่ในคลาสเดียวกันเท่านั้น
 - ชื่อของแอตทริบิวต์
 - ประเภทของแอตทริบิวต์ ซึ่งจะอยู่ต่อจากเครื่องหมายโคลอน (:)
 - ค่าเริ่มต้นของแอตทริบิวต์ ซึ่งอาจไม่มีก็ได้ แต่ถ้ามีจะอยู่ต่อจากเครื่องหมายเท่ากับ(=)
- 3) โอเปอเรชัน (Operation) คือพฤติกรรมที่เราสามารถกระทำกับออบเจ็กต์ได้ ประกอบด้วยองค์ประกอบย่อย ดังนี้
 - ชนิดของการเข้าถึง
 1. Public(+) วัตถุทั้งหมดภายในระบบสามารถเข้าถึงได้
 2. Private(-) วัตถุที่สามารถเข้าถึงต้องเป็นวัตถุที่อยู่ในคลาส หรือ สับคลาสเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Protect(#) วัตถุที่สามารถเข้าถึงจะต้องเป็นวัตถุที่อยู่ในคลาสเดียวกันเท่านั้น
- ชื่อของโอเปอเรชัน
 - พารามิเตอร์ (Parameter)
 - ประเภทของค่าที่ส่งคืน (Return Type)

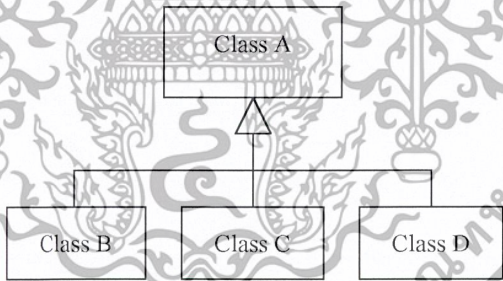
ความสัมพันธ์ระหว่างคลาส (Relationships)

1. Dependency ความสัมพันธ์แบบนี้เกิดขึ้นเมื่อคลาสหนึ่งมีการเปลี่ยนแปลงไปแล้วส่งผลกระทบต่ออีกคลาสหนึ่ง สัญลักษณ์ คือลูกศรที่มีลักษณะเป็นเส้นประ โดยที่คลาสที่อยู่ตรงส่วนหางของลูกศรจะขึ้นอยู่กับคลาสที่อยู่ตรงส่วนหัวของลูกศร



รูปที่ 2.16 แสดงความสัมพันธ์แบบ Dependency

2. Generalization เป็นการถ่ายทอดคุณสมบัติหรือพฤติกรรมบางอย่างจากคลาสหนึ่งไปไปยังอีกคลาสหนึ่ง สัญลักษณ์ คือเส้นตรงทึบที่มีปลายข้างหนึ่งเป็นลูกศรหัวโปร่ง



รูปที่ 2.17 แสดงความสัมพันธ์แบบ Generalization

3. Association เป็นความสัมพันธ์ที่อธิบายถึงการเชื่อมโยงระหว่างคลาส การแสดงความสัมพันธ์จะประกอบด้วย

- 1) Association Name ชื่อของความสัมพันธ์
- 2) Multiplicity การกำหนดปริมาณของคลาส หรือออบเจ็กต์ที่สัมพันธ์กันอยู่

1	มี 1 คลาส / ออบเจกต์
N	มี n คลาส / ออบเจกต์
0..1	ไม่มี หรือมีได้เพียง 1 คลาส / ออบเจกต์
1..*	มีได้ 1 คลาส / ออบเจกต์หรือมากกว่า
*	ไม่มี หรือมีได้ไม่จำกัด

ตารางที่ 2.1 แสดง Multiplicity

รูปแบบของความสัมพันธ์

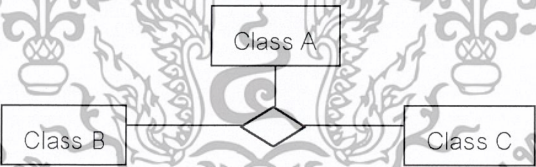
1. Normal Association รูปแบบของ Normal Association มีดังนี้

- Binary Association ความสัมพันธ์ระหว่างคลาส 2 คลาส



รูปที่ 2.18 แสดงความสัมพันธ์แบบ Binary Association

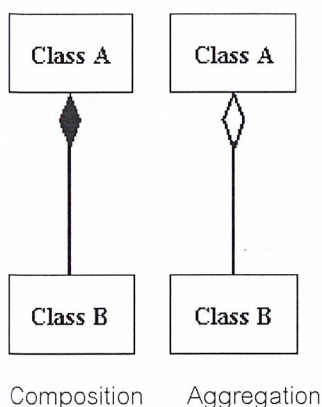
- N-ary Association ความสัมพันธ์ระหว่างคลาสจำนวน 3 คลาสหรือมากกว่า 3 คลาสขึ้นไป โดยความสัมพันธ์จะใช้สัญลักษณ์สี่เหลี่ยมขนมเปียกปูน



รูปที่ 2.19 แสดงความสัมพันธ์แบบ N-ary Association

2. Aggregation เป็นความสัมพันธ์ในแง่ของการรวมกันหรือประกอบกัน ซึ่งองค์ประกอบที่นำมาประกอบนั้นจะมีหรือไม่มีก็ได้ สามารถแสดงด้วยเส้นตรงทึบ โยงระหว่างคลาส โดยมีสัญลักษณ์หัวทศมาสติดโป่ง ติดอยู่ระหว่างปลายเส้นความสัมพันธ์กับคลาส

3. Composition เป็นความสัมพันธ์ในแง่ของการรวมกันหรือประกอบกัน ซึ่งองค์ประกอบที่นำมาประกอบนั้นเป็นปัจจัยสำคัญไม่มีไม่ได้ สามารถแสดงด้วยเส้นตรงทึบ โยงระหว่างคลาส โดยมีสัญลักษณ์หัวทศมาสติดทึบ ติดอยู่ระหว่างปลายเส้นความสัมพันธ์กับคลาส



รูปที่ 2.20 แสดงสัญลักษณ์ของความสัมพันธ์แบบ Composition และ Aggregation

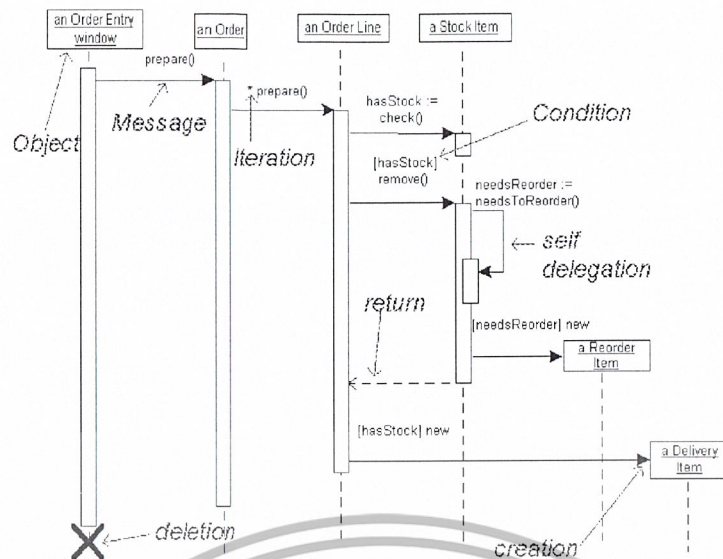
2.3.1.3 บีเฮฟเวอร์ไดอะแกรม (Behavior Diagram) ไดอะแกรมที่บอกถึงพฤติกรรมของตัวระบบเส้นที่ใช้ในการส่งข้อมูล (Message) มีรูปแบบ ดังนี้

1. เส้นทั่วไป (—→) ใช้ส่งข้อมูลแบบทั่วไปไม่เฉพาะเจาะจง
2. เส้นซิงโครนัส (—▶) เป็นเส้นที่ส่งข้อมูลไปแล้วจำเป็นต้องรอการตอบกลับ
3. เส้นอะซิงโครนัส (—) เป็นเส้นที่ส่งข้อมูลไปแล้วไม่จำเป็นต้องรอการตอบกลับ
4. เส้นส่งกลับจากการเรียกฟังก์ชัน (·····▶) มักใช้คู่กับเส้นทั่วไป เมื่อมีค่าบางอย่างต้องการส่งค่ากลับมา

บีเฮฟเวอร์ไดอะแกรม ประกอบด้วยไดอะแกรม ดังนี้

1. อินเทอร์แอคชันไดอะแกรม (Interaction Diagram) เป็นไดอะแกรมที่แสดงพฤติกรรมของการแลกเปลี่ยนข้อมูล (Message) ของออบเจกต์ต่างๆ ซึ่งประกอบด้วย

- 1) ซีควเन्ซ์ไดอะแกรม (Sequence Diagram) ไดอะแกรมนี้แสดงให้เห็นว่าวัตถุแต่ละตัวติดต่อสื่อสารกัน และมีขั้นตอนการทำงานอย่างไร โดยเน้นไปที่แกนเวลา ซีควเन्ซ์ไดอะแกรมประกอบด้วย 2 แกน คือแกนนอนเป็นแกนที่แสดงขั้นตอนการทำงานหรือการส่งเมสเสจระหว่างวัตถุ และแกนตั้งเป็นแกนเวลา

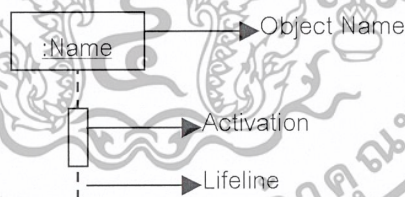


รูปที่ 2.21 แสดงซีควเอนซ์ไดอะแกรม

ซีควเอนซ์ไดอะแกรม มีองค์ประกอบอยู่ 3 ส่วน คือ

1. ออบเจกต์ (Object) ประกอบด้วย

- Object Name ประกอบด้วย ชื่อออบเจกต์:ชื่อคลาส (ชื่อออบเจกต์อาจจะไว้ได้) และขีดเส้นใต้
- Lifeline แสดงถึงชีวิตของวัตถุ
- Activation ช่วงเวลาที่วัตถุกำลังปฏิบัติงาน



รูปที่ 2.22 แสดงส่วนประกอบของออบเจกต์

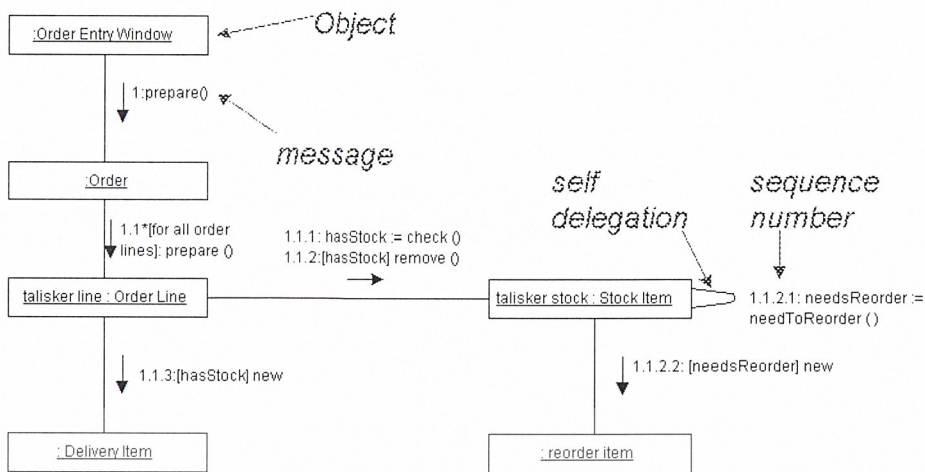
2. เมสเสจ (Message) เป็นการติดต่อระหว่างออบเจกต์

3. ช่วงเวลา (Time) การแสดงเวลาของซีควเอนซ์ไดอะแกรมเป็นลักษณะจากบนลงล่าง เมสเสจที่อยู่ด้านบนจะเป็นส่วนที่เกิดขึ้นก่อนเมสเสจที่อยู่ด้านล่าง

2) คอลแลบอเรชันไดอะแกรม (Collaboration Diagram) จะแสดงขั้นตอนการทำงาน

ของยูสเคสเช่นเดียวกับซีควเอนซ์ไดอะแกรม แต่วิธีการเขียนจะต่างกัน คือ คอลแลบอเรชันไดอะแกรมจะไม่แสดงถึงแกนเวลาอย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



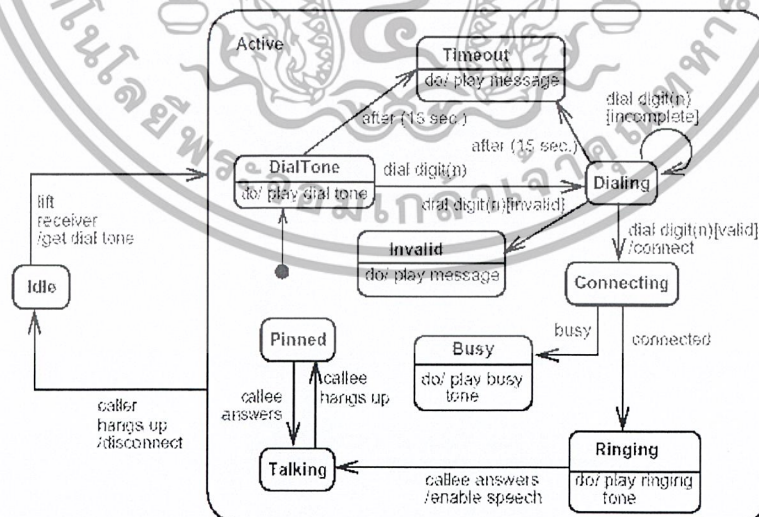
รูปที่ 2.23 แสดงคอลแลบอเรนซ์ไคอะแกรม

คอลแลบอเรนซ์ไคอะแกรมจะประกอบไปด้วย

1. ออบเจกต์ ประกอบด้วย ชื่อของออบเจกต์/บทบาท:ชื่อคลาส และ ชิดเส้นได้
2. ลิงก์ เส้นเชื่อมกันระหว่างวัตถุ ซึ่งขึ้นแสดงขั้นตอนการทำงานตามทิศของ

ลูกศร โดยมีตัวเลขกำกับไว้เพื่อบอกขั้นตอนการทำงาน

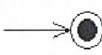
2.3.1.4 สเตตชาร์ตไคอะแกรม (Statechart Diagram) บอกถึงสถานะต่าง ๆ ของคลาสในระบบว่ามีสถานะอะไรบ้าง จะเปลี่ยนสถานะเมื่อเกิดเหตุการณ์อะไร



รูปที่ 2.24 แสดงสเตตชาร์ตไคอะแกรม

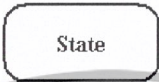
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์ในสแตตชาร์ตไดอะแกรม

- 1. จุดเริ่มต้น (Initial) 
- 2. จุดสิ้นสุด (Final) 

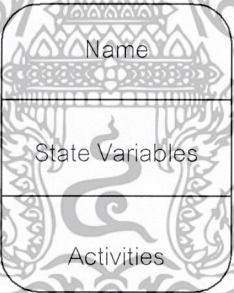
3. Transition Line เส้นที่เชื่อมโยงสแตต เราสามารถใส่รายละเอียดบางอย่างเข้าไปได้ เช่นอาจมีการใส่เหตุการณ์ที่ทำให้เกิด Transition ได้เรียกเหตุการณ์นั้นว่า Trigger event



- 4. สแตต (State) 

รายละเอียดภายในสแตต

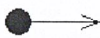
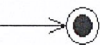
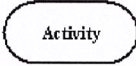
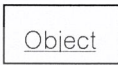
- ชื่อของสถานะการทำงาน (Name)
- ตัวแปรของสถานะการทำงาน (State Variables) ค่าที่ใช้ในสแตต เช่น ต้องการทราบว่าสแตตนี้ทำงานมาแล้วกี่ครั้งจะใช้ State Variable ช่วยในการนับ
- กิจกรรม (Activities) อธิบายถึงพฤติกรรมการทำงานของสแตต

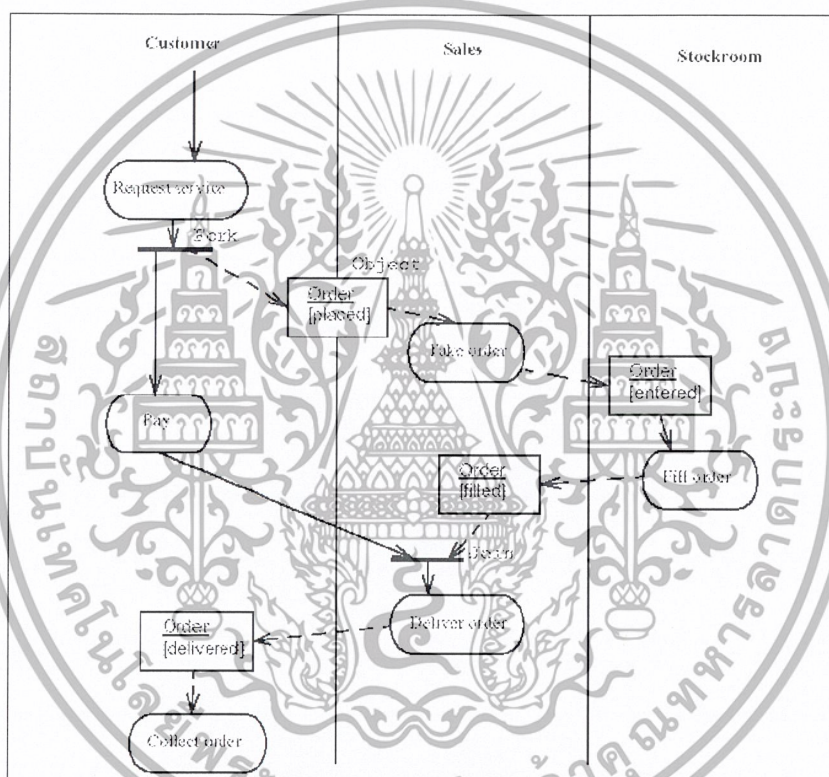


รูปที่ 2.25 แสดงรายละเอียดภายในสแตต

2.3.1.5 แอ็กทิวิตีไดอะแกรม (Activity Diagram) เป็นการแสดงขั้นตอนการทำงานเช่นเดียวกับซีเควนซ์ไดอะแกรม และคอลเลบอเรชันไดอะแกรม แต่จะเน้นไปที่งานย่อยของวัตถุ และสามารถแสดงรายละเอียดของกิจกรรมระหว่างออบเจ็กต์ต่างๆ ได้ แอ็กทิวิตีไดอะแกรมจะเปลี่ยนสถานะเองได้โดยไม่ต้องมีเหตุการณ์ที่กำหนดไว้ในไดอะแกรม แต่จะเปลี่ยนสถานะเองตามกระบวนการทำงานแอ็กทิวิตีไดอะแกรมจะมีลักษณะเดียวกับ Flowchart นอกจากนั้นไดอะแกรมยังแบ่งเป็นสวิตช์ซึ่งแบ่งกลุ่มแอ็กทิวิตีออกเป็นหลายๆ ตามออบเจ็กต์

สัญลักษณ์ที่ใช้ในแอ็กทिवิตีไดอะแกรม

1. จุดเริ่มต้น (Initial) 
2. จุดสิ้นสุด (Final) 
3. Fork มีงานหลายงานที่มีการทำงานไปพร้อมกัน
4. Join แสดงการทำงานที่ต้องทำกิจกรรมเดิมให้เสร็จสิ้นก่อนจึงเริ่มกิจกรรมต่อไป
5. แอ็กทिवิตี (Activity) 
6. วัตถุเจ็ท (Object) 

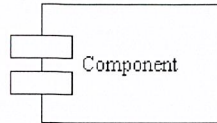


รูปที่ 2.26 แสดงแอ็กทिवิตีไดอะแกรม

2.3.1.6 อิมพลีเมนต์เตชันไดอะแกรม (Implementation Diagram) เป็นสถาปัตยกรรมที่เกิดจากการมองระบบในความเป็นจริง ประกอบไปด้วย 2 ไดอะแกรม คือ

1) คอมโพเนนต์ไดอะแกรม (Component Diagram) เป็นการอธิบายถึงซอฟต์แวร์ต่างๆ ที่เป็นคอมโพเนนต์ของระบบว่ามีความสัมพันธ์และเชื่อมต่อกันอย่างไร สัญลักษณ์ถูกแทนด้วยสี่เหลี่ยมที่ประกอบด้วยสี่เหลี่ยมเล็กๆ อีก 2 รูปติดอยู่ที่ยอดด้านซ้าย

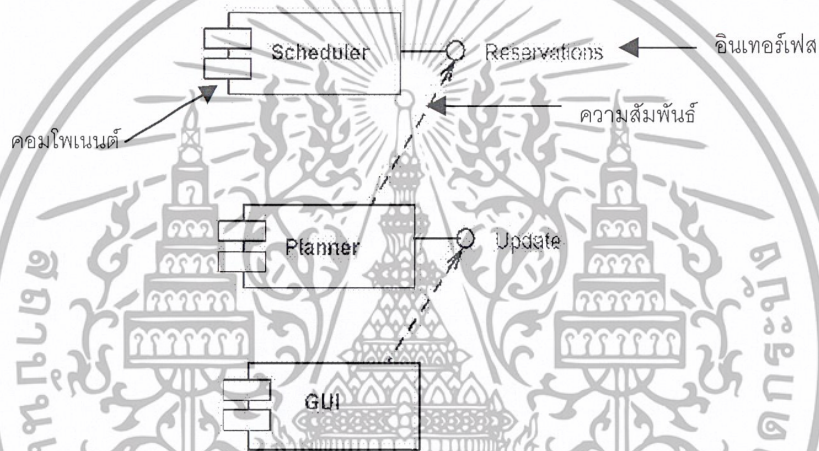
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 แสดงสัญลักษณ์ของคอมโพเนนต์

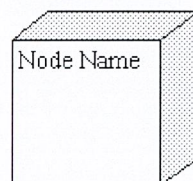
องค์ประกอบหลักของคอมโพเนนต์ไดอะแกรม

1. คอมโพเนนต์ (Component)
2. อินเทอร์เฟซ (Interface)
3. ความสัมพันธ์ (Relationships)



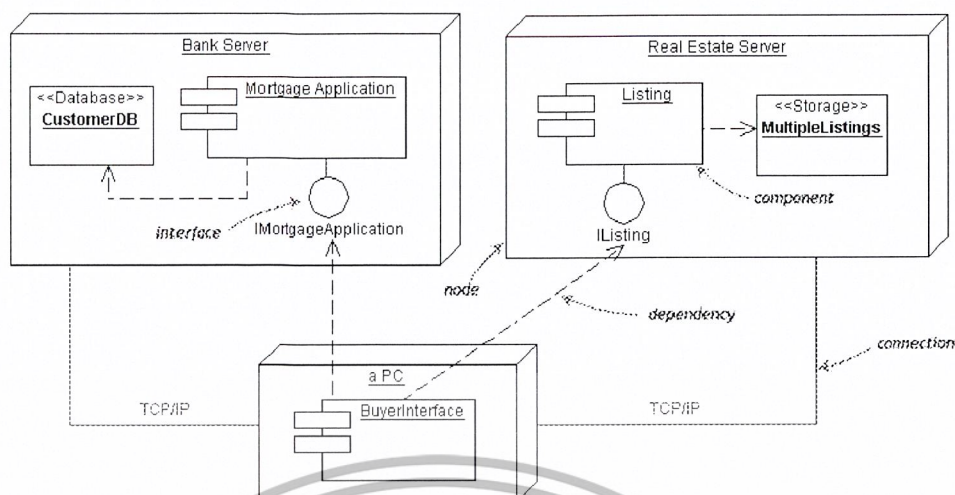
รูปที่ 2.28 แสดงองค์ประกอบหลักของคอมโพเนนต์ไดอะแกรม

2) ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram) เป็นไดอะแกรมแสดงการเชื่อมต่ออุปกรณ์ฮาร์ดแวร์ในระบบ และมักใช้ร่วมกับคอมโพเนนต์ไดอะแกรม โดยภายในฮาร์ดแวร์ (โหนด) จะประกอบไปด้วยซอฟต์แวร์คอมโพเนนต์ การอธิบายความสัมพันธ์จะเหมือนกับคอมโพเนนต์ไดอะแกรม สัญลักษณ์ในการวาดดีพลอยเมนต์ไดอะแกรมจะแทนด้วยรูปลูกบาศก์ 3 มิติ ภายในบรรจุชื่อที่แสดงถึงประเภทของโหนด ส่วนสัญลักษณ์ลูกศรเส้นประ (--->) แสดงให้เห็นว่าคอมโพเนนต์ต้องการขอบริการจากโหนดใด



รูปที่ 2.29 แสดงสัญลักษณ์ของฮาร์ดแวร์ (โหนด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 แสดงดีพลอยเมนต์ไอโอะแกรม

2.4 เจดีบีซี JDBC (Java Database Connectivity)

JDBC คือ Java API (Java Application Programming Interface) มาตรฐานเพื่อใช้งานเฉพาะกับโปรแกรมที่สร้างขึ้นด้วยภาษาจาวาในการติดต่อฐานข้อมูล ใช้ข้อความคำสั่ง SQL โดย JDBC ประกอบด้วยชุดของคลาสและอินเทอร์เฟซที่เขียนด้วยโปรแกรมภาษา Java โดย JDBC สามารถใช้งานได้ทั้งกับระบบฐานข้อมูลในปัจจุบันที่มีอยู่เช่น MS SQL, Oracle, DB2, Informix, mySQL และอื่น ๆ

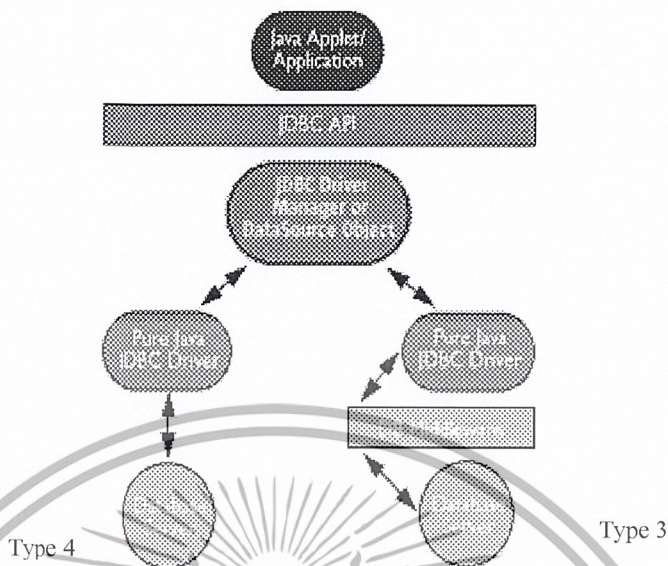
2.4.1 สถาปัตยกรรมของ JDBC

ประกอบไปด้วย 2 ส่วนสำคัญคือ ส่วนของ JDBC Application และส่วนของไดรฟ์เวอร์ ซึ่งไดรฟ์เวอร์สามารถเข้าถึงฐานข้อมูลโดยผ่านทาง JDBC Application โดยที่ JDBC เป็นมิดเดิลแวร์ตัวหนึ่งที่มีรูปแบบ และชื่อเรียกที่แบ่งออกเป็น 4 ประเภทด้วยกัน คือ ประเภทที่ 1 ประเภทที่ 2 ประเภทที่ 3 และประเภทที่ 4 โดยที่

- ประเภทที่ 1 ต้องติดต่อผ่าน ODBC โดยจะมีการแปลงคำสั่งจาก JDBC ไปเป็นคำสั่ง ODBC แล้วส่งคำสั่งนั้นไปที่ ODBC ไดรฟ์เวอร์ เพื่อติดต่อกับฐานข้อมูลที่ต้องการติดต่อด้วย
- ประเภทที่ 2 ไคลเอนต์จะมีไดรฟ์เวอร์สำหรับติดต่อกับฐานข้อมูลแต่ละประเภท
- ประเภทที่ 3 มีการนำ JDBC ไดรฟ์เวอร์ไปไว้ที่ตัวแอปพลิเคชันเซิร์ฟเวอร์ เพื่อเป็นตัวติดต่อกับฐานข้อมูลอีกที
- ประเภทที่ 4 เป็นการทำงานโดยติดต่อกับระบบฐานข้อมูลโดยตรงจากที่เซิร์ฟเวอร์ มีรูปแบบรองรับการใช้งาน JDBC อยู่แล้ว

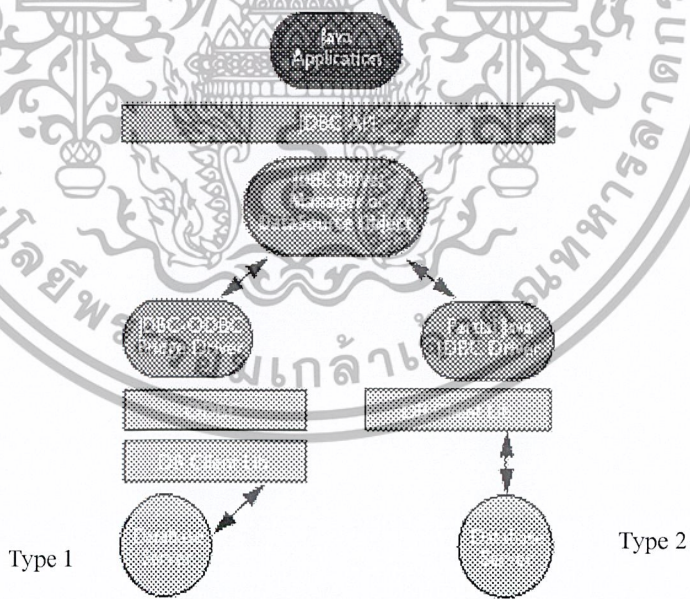
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าถึงฐานข้อมูลโดยผ่านทาง JDBC ซึ่งใช้ไดรฟ์เวอร์พื้นฐานแสดงได้ ดังรูป



รูปที่ 2.31 แสดงการเข้าถึงฐานข้อมูลโดยผ่านทาง JDBC ซึ่งใช้ไดรฟ์เวอร์พื้นฐาน

ส่วนการติดต่อ JDBC โดยใช้ ODBC ไดรฟ์เวอร์แสดงได้ดังรูป



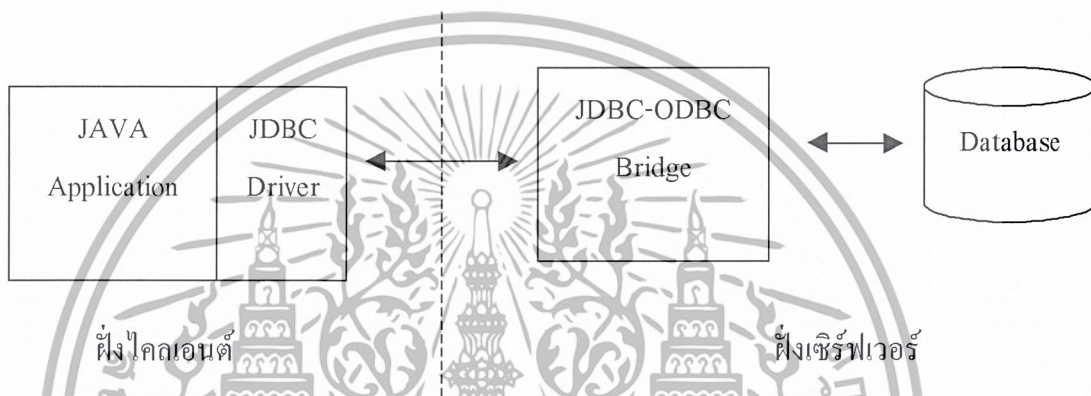
รูปที่ 2.32 แสดงการติดต่อ JDBC โดยใช้ ODBC ไดรฟ์เวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 รายละเอียดของ JDBC ของแต่ละประเภทมี ดังนี้

2.4.2.1 ประเภทที่ 1 (Type 1: JDBC-ODBC Bridge)

หมายถึง JDBC ที่ทำงานอยู่บนชั้นการสื่อสารมิดเดิลแวร์ที่ชื่อ ODBC ซึ่ง JDBC/ODBC bridge ได้ถูกพัฒนาขึ้นโดยการร่วมมือกันระหว่าง JavaSoft และ Intersolv เพื่อที่จะเพิ่มความสามารถให้กับฐานข้อมูลจำนวนมากที่ใช้ ODBC โดย bridge จะทำการแปลงคำสั่งจาก JDBC ไปเป็นคำสั่งของ ODBC แล้วส่งคำสั่งนั้นไปยังตัว ODBC driver เพื่อจัดการกับฐานข้อมูล ดังรูป



รูปที่ 2.33 แสดงการติดต่อกับฐานข้อมูลโดยผ่าน ODBC

ข้อดีของการนำไปใช้งาน

- โปรแกรมที่เขียนขึ้นจะเข้าถึงข้อมูลจากระบบฐานข้อมูลที่มีอยู่ในระบบปัจจุบันได้

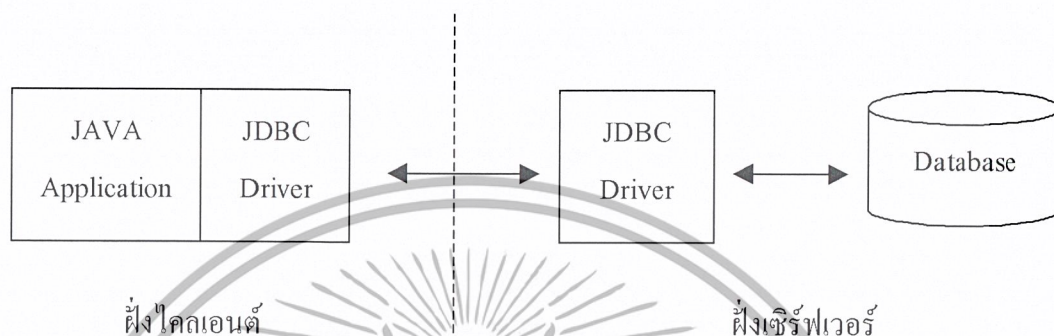
โดยง่าย โดยการเลือก ODBC driver ที่เหมาะสม

ข้อเสียของการนำไปใช้งาน

- ไม่เหมาะกับการนำไปใช้งานบนระบบที่มีขนาดใหญ่ เนื่องจากทำให้เกิดความล่าช้าในการทำงานและมีประสิทธิภาพในการทำงานที่ไม่ดี
- มีข้อมูลในส่วนโอเวอร์เฮด (Overhead) สูง เนื่องจากต้องมีส่วนในการติดต่อระหว่าง JDBC และ ODBC เพิ่มเติม
- ไม่สนับสนุนความสามารถทั้งหมดของมาตรฐาน JDBC เนื่องจากข้อจำกัดของ ODBC ที่มีรูปแบบการทำงานน้อยกว่า JDBC

2.4.2.2 ประเภทที่ 2 (Type 2: Native API หรือ Partly API)

มิดเดิลแวร์ JDBC ประเภทนี้จะถูกโหลดไว้ที่ไคลเอนต์หรือเซิร์ฟเวอร์ที่ติดต่อกับผู้ใช้ ดังนั้น รูปแบบในการสร้างโปรแกรมไคลเอนต์ จะต้องการไคร์ฟเวอร์สำหรับติดต่อกับฐานข้อมูล แต่ละประเภทที่ต้องการติดต่อด้วย เช่น ถ้ามีระบบฐานข้อมูล Oracle โปรแกรมในส่วนไคลเอนต์ ต้องสร้างคำสั่งในการโหลดไคร์ฟเวอร์ Oracle ขึ้นใช้งาน



รูปที่ 2.34 แสดงการติดต่อกับฐานข้อมูลประเภทที่ 2

ข้อดีของการนำไปใช้งาน

- ลดขั้นตอนการติดต่อ ODBC ลง และทั้งฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ต่างก็เป็น JDBC ด้วยกันทั้งคู่ จึงสามารถทำงานร่วมกันได้ดี

ข้อเสียของการนำไปใช้งาน

- ผู้ใช้โปรแกรมในส่วนไคลเอนต์ยังต้องการไคร์ฟเวอร์ สำหรับผลิตภัณฑ์เซิร์ฟเวอร์ในแต่ละเซิร์ฟเวอร์

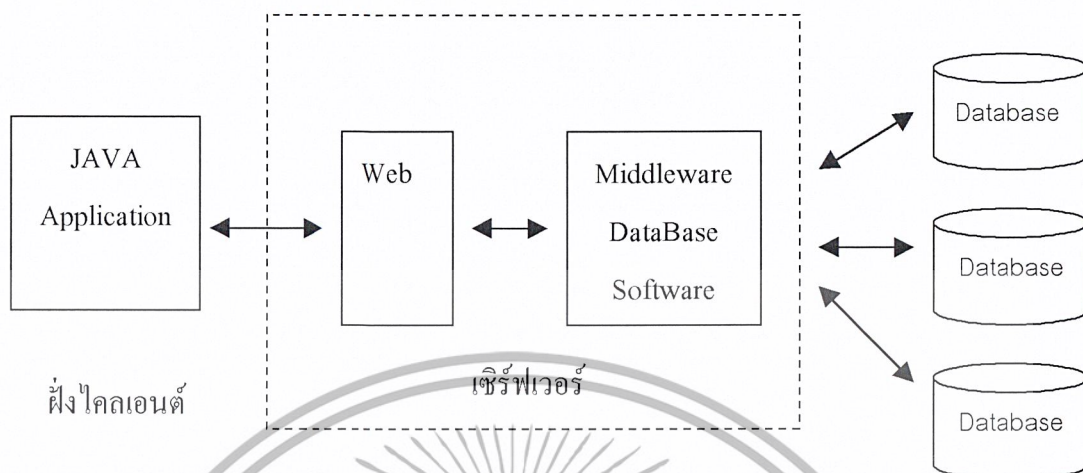
- เมื่อมีการเปลี่ยนแปลงระบบจัดการฐานข้อมูลเป็นผลิตภัณฑ์ตัวอื่น โปรแกรมในส่วนไคลเอนต์ต้องมีการเปลี่ยนแปลงและคอมไพล์ใหม่เสมอ

2.4.2.3 ประเภทที่ 3 (Type 3: JDBC-Net)

คำสั่งจากการเรียกใช้งาน JDBC จะถูกส่งผ่านโปรโตคอลการสื่อสารมาตรฐานส่วนกลางใด ๆ ไปที่ระบบฐานข้อมูลบนเซิร์ฟเวอร์หากมองง่าย ๆ จะมีลักษณะเหมือนกับสถาปัตยกรรมหลายเทียร์ซึ่งจำเป็นต้องมีแอปพลิเคชันเซิร์ฟเวอร์ทำงานร่วมอยู่ในระบบ แอปพลิเคชันเซิร์ฟเวอร์จะมาลดข้อยุ่งยากในการติดตั้ง หรือดูแลระบบในส่วนเทียร์ไคลเอนต์ ดังนั้นลักษณะของ JDBC ประเภทที่ 3 คือการนำเอา JDBC ไคร์ฟเวอร์ไปวางไว้ที่ตัวแอปพลิเคชันเซิร์ฟเวอร์เพื่อเป็นตัวเชื่อมโยงการสื่อสารและจะเหมาะสม อย่างยิ่งกับระบบงานที่ทำงานผ่านเครือข่ายอินเทอร์เน็ต หรือ

อินทราเน็ตในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 แสดงการติดต่อฐานข้อมูลประเภทที่ 3

ข้อดีของการนำไปใช้งาน

- เป็นการลดการทำงานในส่วนของไคลเอนต์ลง โดยมีมิดเดิลแวร์คอยติดต่อกับฐานข้อมูลในแต่ละผลิตภัณฑ์

ข้อเสียของการนำไปใช้งาน

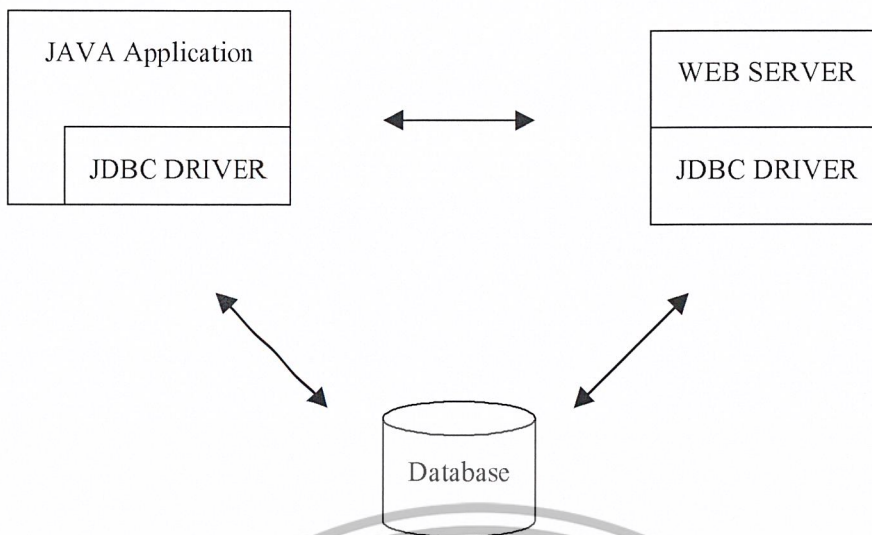
ข้อเสียของการนำไปใช้งาน

- ต้องติดตั้งแอปพลิเคชันเซิร์ฟเวอร์ขึ้นเพื่อทำงานระหว่างโปรแกรม(ไคลเอนต์) กับฐานข้อมูล (เซิร์ฟเวอร์)
- ต้องการ ไดรฟ์เวอร์สำหรับแต่ละผลิตภัณฑ์ เพื่อติดตั้งไว้ที่แอปพลิเคชันเซิร์ฟเวอร์

2.4.2.4 ประเภทที่ 4 (Type 4: Native Protocol)

การทำงานของ JDBC เป็นการทำงานโดยติดต่อกับระบบฐานข้อมูลโดยตรง จากที่เซิร์ฟเวอร์มีรูปแบบรองรับการใช้งาน JDBC อยู่แล้ว ดังนั้นการทำงานจึงไม่ต้องผ่านไดรฟ์เวอร์หรือแอปพลิเคชันเซิร์ฟเวอร์ใดๆ เหมือนกับประเภทที่ 1 ถึง ประเภทที่ 3 ดังนั้นเซิร์ฟเวอร์ต้องรองรับการติดต่อที่เป็นไปตามข้อกำหนดของ JDBC ทำให้ JDBC ประเภทนี้มีรูปแบบการทำงานที่รวดเร็ว แต่จำเป็นต้องสร้างไคลเอนต์สำหรับฐานข้อมูลแต่ละระบบ ซึ่งในบางครั้งไคลเอนต์สำหรับฐานข้อมูลตัวหนึ่งจึงไม่สามารถนำไปใช้กับระบบฐานข้อมูลตัวอื่น ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 แสดงการติดต่อฐานข้อมูลประเภทที่ 4

2.4.3 การเขียนโปรแกรมเพื่อติดต่อฐานข้อมูลของ JDBC ทั้ง 4 ประเภท มีขั้นตอน ดังนี้

- 2.4.3.1 ขั้นตอนเริ่มติดต่อกับฐานข้อมูล (Establish Database)
- 2.4.3.2 ขั้นตอนการส่งคำสั่ง SQL ให้กับฐานข้อมูล (Send SQL)
- 2.4.3.3 ขั้นตอนการแสดงผลลัพธ์ผู้ใช้ (Display Result)
- 2.4.3.4 ขั้นตอนการยกเลิกการติดต่อกับฐานข้อมูล (Close Connection)

2.4.3.1 ขั้นตอนการเริ่มติดต่อกับฐานข้อมูล (Establish Database)

การเขียนคำสั่งเพื่อเริ่มติดต่อกับระบบฐานข้อมูลนั้น แบ่งออกเป็น 2 ขั้นตอนดังนี้ คือ

1) ขั้นตอนการโหลดไดรฟ์เวอร์เพื่อใช้งาน

เป็นขั้นตอนการโหลดไดรฟ์เวอร์ JDBC ขึ้นมาใช้งานในโปรแกรม เพื่อเป็นการระบุถึงรูปแบบการติดต่อว่าเป็นประเภทใด รวมถึงชื่อฐานข้อมูล

รูปแบบ

```
Class.forName("Driver_Name")
```

คลาสดักจับข้อผิดพลาด

```
ClassNotFoundException
```

โดยที่

Driver_Name คือ ชื่อไดรฟ์เวอร์สำหรับใช้งานระบบฐานข้อมูลที่ต้องการโหลดขึ้นมาใช้งาน ซึ่งจะต้องระบุชื่อให้ตรงกับระบบฐานข้อมูลที่ต้องการติดต่อ

ในการโหลดไดรฟ์เวอร์ไม่จำเป็นต้องสร้างอินสแตนซ์เพื่อใช้งาน เนื่องจากหลังจากโหลดเป็นผลสำเร็จแล้ว การใช้งานในคำสั่งต่อๆ ไปโปรแกรมจะทราบโดยอัตโนมัติว่าเป็นการใช้ไดรฟ์เวอร์ที่ถูกโหลดมาก่อนหน้านั้น

ตัวอย่าง

1. `Class.forName("sun.jdbc.odbc.JdbcOdbcDriver")`

เป็นการเขียนคำสั่งเพื่อโหลดไดรฟ์เวอร์ประเภท JDBC-ODBC ขึ้นมาใช้งาน

2. `Class.forName("jdbc.DatabaseDriver")`

เป็นการเรียกไดรฟ์เวอร์จากผู้ผลิตอื่น ๆ

2) ขั้นตอนการติดต่อไปยังระบบฐานข้อมูลปลายทาง

ขั้นตอนนี้ถูกทำหลังจากไดรฟ์เวอร์ถูกโหลดเป็นผลสำเร็จเท่านั้น เปรียบเสมือนเป็นการเปิดช่องทางการติดต่อไปยังฐานข้อมูลปลายทาง การสร้างคำสั่งสำหรับขั้นตอนนี้ต้องอาศัยข้อมูลประกอบกัน 3 ตัวในการทำงาน คือ

1. ข้อมูล URL หมายถึงข้อมูลที่ระบบเส้นทางหรือแอคเคสในระดับเครือข่าย ใช้กำหนดที่อยู่ของระบบฐานข้อมูล โดยส่วนใหญ่แล้วเป็นฐานข้อมูลที่มีคุณสมบัติเป็นเซิร์ฟเวอร์หรือฐานข้อมูลภายในระบบผ่าน ODBC

รูปแบบ

`jdbc:<subprotocol>:<subname>`

โดยที่

jdbc คือ ส่วนที่ใช้ระบุการติดต่อสื่อสารว่าอยู่ในรูปแบบข้อกำหนดของ JDBC
 <subprotocol> คือ โปรโตคอลการสื่อสารข้อมูล เพื่อเจาะจงไปยังเครื่องปลายทาง ซึ่งมีข้อมูลแตกต่างกันไปตามรูปแบบการสื่อสาร เช่น

- `odbc` สำหรับการสื่อสารผ่านระบบ ODBC
- `dbnet` สำหรับการสื่อสารผ่านโปรโตคอล TCP/IP
- `<subname>` คือ ชื่อฐานข้อมูลภายในระบบฐานข้อมูลที่อ้างผ่าน subprotocol

หรือชื่อแหล่งข้อมูล (Datasource) หาก subprotocol คือ `odbc` แล้ว subname ก็คือชื่อ `dsn` นั่นเอง

ตัวอย่าง

1. `jdbc:odbc:dbStudent`

แสดงการติดต่อโดยการใช้งาน `odbc` โดยใช้ไปยังชื่อ `dbStudent` ซึ่งเป็นชื่อแหล่งข้อมูล (ต้องกำหนดผ่าน `odbc administrator`) ซึ่งการใช้งานผ่าน `odbc` เป็นการใช้งานฐานข้อมูลภายในระบบตัวเองเป็นส่วนใหญ่

2. jdbc:dbnet://127.0.0.1:888/human

เป็นการเข้าถึงเครื่องคอมพิวเตอร์ที่มีหมายเลขไอพี 127.0.0.1 ผ่านพอร์ตสื่อสาร หมายเลข 888 และเข้าถึงฐานข้อมูลภายในที่ชื่อ human ดังนั้นในเครื่องคอมพิวเตอร์ดังกล่าวจำเป็นต้องมีโปรแกรมฐานข้อมูลระดับเซิร์ฟเวอร์ที่รองรับการติดต่ออยู่ที่พอร์ตหมายเลข 888 ด้วย

2. ข้อมูลผู้ใช้ (Login) หมายถึงข้อมูลที่ต้องระบุเพื่อแสดงตัวผู้ใช้งานว่าคือใคร ให้กับระบบฐานข้อมูลปลายทางรับทราบว่ามีสิทธิ์

บริหารระบบฐานข้อมูล (Database Administrator) ที่ควบคุมระบบฐานข้อมูลนั้น ๆ เพื่อให้เพิ่มรายชื่อเข้าไปในบัญชีรายชื่อสำหรับฐานข้อมูล รวมถึงกำหนดสิทธิ์การใช้งานข้อมูลในส่วนที่ต้องการ

3. ข้อมูลรหัสผ่าน (Password) หมายถึงข้อมูลรหัสผ่าน ซึ่งเป็นข้อมูลอีกส่วนที่มักอยู่ควบคู่กับข้อมูลผู้ใช้ ใช้เพื่อยืนยันว่าผู้ติดต่อเข้าระบบเป็นเจ้าของข้อมูลผู้ใช้งานจริง ไม่ได้เป็นผู้อื่นที่นำข้อมูลผู้ใช้ไปใช้งาน ผู้ใช้จึงควรจดบันทึกข้อมูลทั้งสองตัวโดยไม่ให้ผู้อื่นนำไปใช้งาน เพราะนั่นคือการรักษาความปลอดภัยของข้อมูลวิธีหนึ่ง

หลังจากที่ได้ข้อมูลครบทั้งสามส่วนแล้ว ขั้นตอนต่อไปคือ การนำมาใช้งานกับเมธอด getConnection ซึ่งเป็นเมธอดในการติดต่อไปยังปลายทางโดยนำข้อมูล url ข้อมูลผู้ใช้ ข้อมูลรหัสผ่าน ในการติดต่อและยืนยันสิทธิ์ในการใช้งาน ดังนี้

รูปแบบการติดต่อกับฐานข้อมูล

Connection Instance_Name = DriverManager.getConnection (URL, Login, Password)

รูปแบบเมธอดในอินเตอร์เฟส Connection

void setAutoCommit(boolean autoCommit) ให้มีการยืนยันการเปลี่ยนแปลงข้อมูลตามข้อมูล

void commit() ยืนยันการเปลี่ยนแปลงข้อมูล

void rollback() ยกเลิกการเปลี่ยนแปลงข้อมูลที่ผ่านมา

รูปแบบเมธอดในคลาส DriverManager

static Connection getConnection(String url)

การติดต่อโดยอาศัยข้อมูล url

static Connection getConnection(String url, String user, String password)

การติดต่อโดยอาศัยข้อมูล url บอกเส้นทาง user บอกชื่อผู้ใช้ และ password บอกรหัสผ่าน

ผ่าน

static void setLoginTimeout(int seconds)

ตั้งช่วงเวลารอคอยการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสดักจับข้อผิดพลาด

SQLException

ตัวอย่าง

```
String URL = "jdbc:odbc:myData";

Connection con = DriverManager.getConnection (URL, "myLogin",
"myPassword");
```

2.4.3.2 ขั้นตอนการส่งคำสั่ง SQL ให้กับฐานข้อมูล (Send SQL)

ขั้นตอนนี้จะเป็นการทำงานกับระบบฐานข้อมูลโดยการใช้คำสั่ง SQL เพื่อส่งให้ระบบฐานข้อมูลประมวลผลคำสั่งตามความต้องการ

เมรอตที่ถูกกำหนดไว้สำหรับส่งคำสั่ง SQL ไปให้กับฐานข้อมูล ถูกแบ่งตามรูปแบบคำสั่ง SQL ได้ 2 แบบ โดยผู้ใช้ต้องเลือกเมรอตที่เหมาะสมกับคำสั่ง SQL ที่ใช้งาน คือ

หากคำสั่ง SQL เป็นประเภทดำเนินการ ซึ่งสร้างด้วยคีย์เวิร์ดจำพวก CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, ... การทำงานของคำสั่งลักษณะนี้เป็นการส่งคำสั่งไปยังฐานข้อมูล เพื่อให้เกิดการเปลี่ยนแปลงใด ๆ ในระบบฐานข้อมูล ดังนั้นผู้ใช้คำสั่งเพียงต้องการรับทราบแล้วว่าทำงานสำเร็จหรือไม่เท่านั้นการเรียกใช้คำสั่งเหล่านี้ให้ทำงานจึงต้องใช้งานผ่านเมรอต ExecuteUpdate

หากคำสั่ง SQL เป็นประเภทสืบค้นข้อมูล ซึ่งสร้างด้วยคีย์เวิร์ด SELECT เนื่องจากคำสั่งประเภทสืบค้นข้อมูลมีลักษณะพิเศษ คือเมื่อระบบฐานข้อมูลได้รับคำสั่ง SQL แล้วจะต้องส่งข้อมูลผลลัพธ์ที่ต้องการกลับมาในรูปแบบของกลุ่มข้อมูลเรคคอร์ด (Record Set) การเรียกใช้งานคำสั่ง SQL ประเภทนี้จึงต้องเรียกใช้งานผ่านเมรอต ExecuteQuery และต้องมีคลาสมารับผลลัพธ์ที่ได้รับกลับมาในรูปแบบของกลุ่มเรคคอร์ด คือ คลาส ResultSet

เมรอตดังกล่าวสามารถเรียกใช้งานได้จากคลาส Statement และคลาส PreparedStatement อย่างใดอย่างหนึ่ง แต่ทั้งสองคลาสดังกล่าวใช้สำหรับการสร้างคำสั่ง SQL ที่แตกต่างกัน คือ

คลาส

Statement ใช้สำหรับสร้างคำสั่ง SQL ที่เป็นข้อมูลคงที่ (ข้อมูลในคำสั่ง SQL ไม่เปลี่ยนแปลง)

PreparedStatement ใช้สำหรับสร้างคำสั่ง SQL ที่สามารถเปลี่ยนแปลงข้อมูลบางส่วนภายในตามพารามิเตอร์ที่ต้องการใช้งาน

รูปแบบคลาส Statement

int ExecuteUpdate() สำหรับสั่งงานคำสั่ง SQL ประเภทจัดการข้อมูล

ResultSet executeQuery() สำหรับสั่งงานคำสั่ง SQL ประเภทสืบค้นข้อมูล

รูปแบบคลาส PreparedStatement

void setBoolean(int index, boolean x) กำหนดข้อมูล boolean ให้กับพารามิเตอร์
ตำแหน่ง index

void setByte(int index, byte x) กำหนดข้อมูล byte ให้กับพารามิเตอร์ตำแหน่ง index

void setShort(int index, short x) กำหนดข้อมูล short ให้กับพารามิเตอร์ตำแหน่ง

index

void setInt(int index, int x) กำหนดข้อมูล int ให้กับพารามิเตอร์ตำแหน่ง index

void setLong(int index, long x) กำหนดข้อมูล long ให้กับพารามิเตอร์ตำแหน่ง index

void setFloat(int index, float x) กำหนดข้อมูล float ให้กับพารามิเตอร์ตำแหน่ง index

void setDouble(int index, double x) กำหนดข้อมูล double ให้กับพารามิเตอร์

ตำแหน่ง index

void setString(int index, String x) กำหนดข้อมูล String ให้กับพารามิเตอร์ตำแหน่ง

index

void setDate(int index, Date x) กำหนดข้อมูล Date ให้กับพารามิเตอร์ตำแหน่ง

index

void setTime(int index, Time x) กำหนดข้อมูล Time ให้กับพารามิเตอร์ตำแหน่ง

index

void clearParameter() ลบข้อมูลออกจากพารามิเตอร์ทุกตัว

int executeUpdate() สำหรับสั่งคำสั่ง SQL ประเภทจัดการข้อมูล

ResultSet executeQuery() สำหรับสั่งคำสั่ง SQL ประเภทสืบค้นข้อมูล

void close() ปิดช่องทางการสื่อสาร

2.4.3.3 ขั้นตอนการแสดงผลลัพธ์ผู้ใช้ (Display Result)

สำหรับในขั้นตอนนี้จะถูกใช้งาน ก็ต่อเมื่อผู้ใช้งานส่งคำสั่งประเภทสืบค้นข้อมูลไปยังระบบฐานข้อมูลด้วยเมธอด executeQuery เนื่องจากเมธอดดังกล่าวจะให้ผลลัพธ์เป็นชุดข้อมูลกลับมา ผู้เขียนโปรแกรมต้องดึงข้อมูลในชุดดังกล่าวมาแสดง โดยอาจจะแสดงผ่านทางหน้าจอคอนโซล (Text Mode) หรือผ่านทางสวิงซ์คอมโพเนนต์ ตามความต้องการ

ผลลัพธ์ในชุด ResultSet มีโครงสร้างคล้ายกับข้อมูลในเทเบิล คือออกแบบเป็นแนวแถวเรียกว่า Row (หรือเรคคอร์ด) และแนวคอลัมน์เรียกว่า Column (หรือฟิลด์) ดังนั้นหลักการดึงข้อมูล คือต้องระบุ Row ให้ชัดเจนก่อนแล้วจึงค่อยระบุ Column

Column ที่ 1	Column ที่ 2	Column ที่ 3	Column ที่ 4	ตำแหน่งตัวชี้
				Before First
xxxx	xx	xxxxxxx	Xxxxxxxxxxxx	First(1)
xxxx	xx	xxxxxxx	Xxxxxxxxxxxx	
xxxx	xx	xxxxxxx	Xxxxxxxxxxxx	
xxxx	xx	xxxxxxx	Xxxxxxxxxxxx	Last(n)
				After Last

ตารางที่ 2.2 แสดงโครงสร้างข้อมูลภายในชุด ResultSet

ตัวอย่างแสดงการเรียกใช้ ResultSet

```
...
try {
    // Use some connection we've already created
    Statement Stmt = Conn.createStatement();
    ResultSet RS = Stmt.executeQuery("SELECT field1 from Some_Table");
    while (RS.next()) {
        System.out.println(RS.getString(1));
    }

    // Clean up after ourselves
    RS.close();
    Stmt.close();
    Conn.close();
}

catch (SQLException E) {
    System.out.println("SQLException: " + E.getMessage());
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        System.out.println("SQLState: " + E.getSQLState());

        System.out.println("VendorError: " + E.getErrorCode());

    }

    ...

```

จากตัวอย่างด้านบนนี้เป็นการเขียนโปรแกรมภาษา Java ติดต่อกับฐานข้อมูล ซึ่งผ่านการสร้าง connection มาแล้ว เมื่อเราส่งคำสั่ง Select field1 from Some_Table ไปเมธอดexecuteQuery จะคืนค่ากลับมาเป็นออบเจกต์ของ class ResultSet ซึ่งจะอยู่ในรูปของ Array เราสามารถดึงค่าขึ้นมาใช้ได้ โดยใช้ Methods getXxx(index); เช่นถ้าเราจะดึงค่าที่เป็น String ออกมาเราก็ต้องใช้ methods getString

2.4.3.4 ขั้นตอนการยกเลิกการติดต่อกับฐานข้อมูล (Close Connection)

สำหรับขั้นตอนยกเลิกการติดต่อกับฐานข้อมูลนี้เป็นขั้นตอนที่ง่าย และไม่ยุ่งยากในการทำงานหากเมื่อพิจารณาจากการดำเนินการติดต่อกับข้อมูล มีอยู่ 2 ส่วนด้วยกัน ส่วนแรก คือช่องทางการติดต่อที่สร้างมาจากเมธอด getConnection และส่วนที่สอง คือ ชุดคำสั่ง ResultSet ที่ได้จากเมธอด executeQuery หลังจากการใช้งานควรหยุดการสื่อสารทั้งสองตัวแปรด้วยเมธอด close ตามตัวอย่าง

```

try{
    Connection con = DriverManager.getConnection(url, "myLogin", "myPassword");
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT * FROM Products")
    ...
    ...
    ...
    rs.close();
    con.close();
} } เป็นการหยุดการสื่อสาร

} catch (SQLException ex){
    System.out.println("exception: " + ex.getMessage());
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 อาร์เอ็มไอ RMI (Remote Method Invocation)

วัตถุประสงค์ของ RMI คือเพื่ออำนวยความสะดวกในการพัฒนา distributed programs โดยใช้ syntax และ semantics แบบเดียวกันกับที่คุ้นเคยอยู่แล้วบน non-distributed programs โดย RMI จะเป็นตัวกำหนดว่าออบเจกต์ต้องมีพฤติกรรมอย่างไร จะจัดการให้เกิด exception เมื่อไร และอย่างไร จะจัดการหน่วยความจำอย่างไร และจะส่งพารามิเตอร์กับ remote method อย่างไร

กฎการส่งค่า (Parameters) และคืนค่า (return) ของ Method สำหรับ Remote Method Interface

- 1. จะต้องเป็น simple data type เช่น boolean, byte, char, int ...
- 2. ถ้าเป็น Object type จะต้อง implement จาก java.io.Serializable

ปัญหาอยู่ที่ parameters และ return ที่เป็น Object type ใน Java Language/Platform ต้องเป็น ออบเจกต์ที่ active เท่านั้นทำให้ หากไม่มี Active Object ก็จะต้อง download ผ่าน HTTP หรือ FTP ตาม URL ที่กำหนดไว้

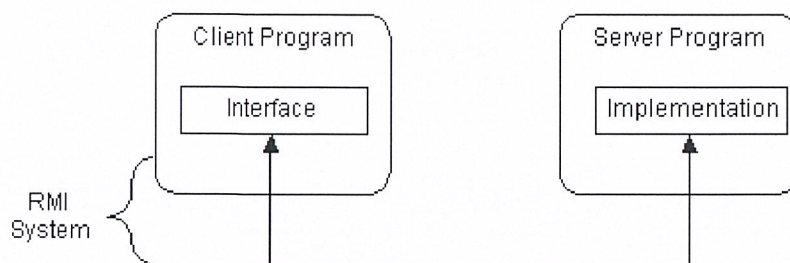
เพื่อให้เห็นการเปรียบเทียบระหว่าง distributed กับ nondistributed ชัดเจน ให้ดูตารางนี้ประกอบ

	Local Object	Remote Object
Object Definition	กำหนดโดย a Java class	exported behavior ถูกกำหนดโดยอินเทอร์เฟซอื่นหนึ่งซึ่งเป็น อินเทอร์เฟซที่จะต้อง extend มาจาก Remote interface.
Object Implementation	implement โดย Java class.	Java class ที่ implements รีโมตอินเทอร์เฟซจะทำการ execute behavior ของ remote object นั้น
Object Creation	การสร้าง instance ใหม่ของ local object จะใช้ new operator	การสร้าง instance ใหม่ของรีโมตออบเจกต์ จะสร้างที่ host computer โดยใช้ new operator. แต่ไคลเอนต์ไม่สามารถสร้าง รีโมตออบเจกต์ใหม่ได้โดยตรง นอกจากจะใช้ Java 2 Remote Object Activation
Object Access	สามารถ access โดยตรง โดยผ่านทาง object reference variable	Access โดยผ่านทาง object reference variable ซึ่งชี้ไปที่ proxy stub implementation ของฝั่ง remote interface.
References	ในระบบ JVM เดียวสามารถชี้ไปโดยตรงที่ object ใน heap	ใช้ remote reference เป็นตัวชี้ไปที่ proxy object (คือ stub นั้นเอง) ที่อยู่ใน local heap. และสแตบนี้จะมีข้อมูลที่จะเชื่อมต่อไปยัง remote object
active	ในระบบ single JVM ถ้า	ในระบบ distributed อาจเป็นไปได้ที่ remote JVMs

ferences	มีอย่างน้อยหนึ่ง reference ไปถึง object ใดๆ จะถือว่า object นี้ "alive"	จะเสียหาย หรือต่อเน็ตเวิร์คไม่ได้ ดังนั้นจะต้องมีการ access ไปที่ remote object นั้นๆภายในช่วงเวลาหนึ่ง (เรียกว่า lease period) จึงจะถือว่า รีโมตออบเจกต์ นั้นมี active remote reference ไปถึง ถ้าไม่เช่นนั้น รีโมตออบเจกต์ นี้ก็ต้องถูกจัดการโดย distributed garbage collection.
Finalization	ถ้าออบเจกต์มีการ implement เมธอด finalize() มันจะถูกเรียกใช้งานก่อนที่จะถูกจัดการโดย garbage collector	ถ้ารีโมตออบเจกต์มีการ implements อินเทอร์เฟส Unreferenced เมธอดที่สร้างจากอินเทอร์เฟสนี้จะถูกเรียกเมื่อมีการยกเลิก remote reference ทั้งหมด
Garbage Collection	จะเกิดเมื่อ local references ของออบเจกต์ ถูก drop ไป	distributed garbage collector ทำงานร่วมกับ local garbage collector. ถ้าไม่มี remote references และ local references ที่ชี้ไปยัง remote object ทั้งหมด ถูก drop ก็จะเริ่มกระบวนการ garbage collection ตามปกติ
Exceptions	ไม่ว่าจะเป็น Runtime exceptions หรือจะเป็น Exceptions อื่นใด, ตัว Java compiler จะบังคับให้โปรแกรมจัดการกับ Exceptions.ทั้งหมด	RMI จะเป็นตัวบังคับให้โปรแกรมจัดการกับออบเจกต์ RemoteException ใดๆ ที่อาจเกิดขึ้น

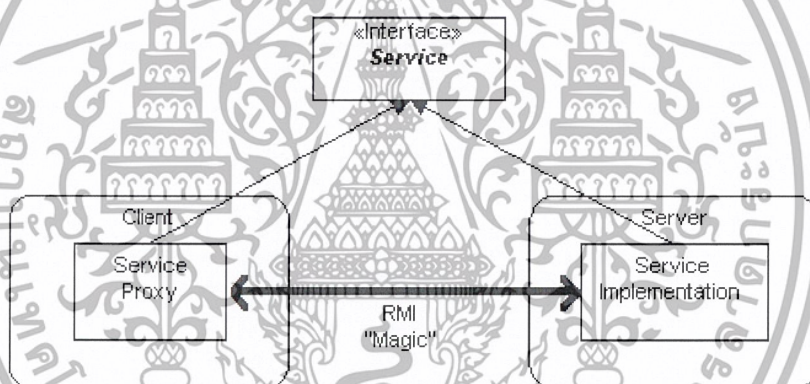
ตารางที่ 2.3 แสดงการเปรียบเทียบระหว่าง distributed กับ nondistributed

คุณสมบัติอีกประการหนึ่งของ RMI ก็คือ การแบ่งแยกกันอย่างชัดเจนระหว่าง การกำหนด พฤติกรรมซึ่งจะใช้อินเทอร์เฟส และการ implement behavior นั้นจะทำให้ class ดังรูป



รูปที่ 2.37 แสดงการกำหนดพฤติกรรมโดยใช้อินเทอร์เฟส

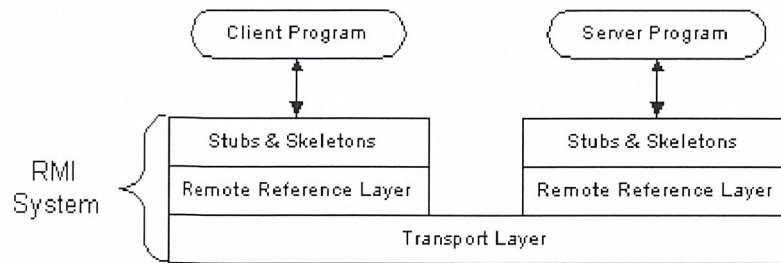
ในส่วนของอินเทอร์เฟสนั้น ไม่สามารถสร้าง executable code ได้ RMI จึงสนับสนุนให้มีสองคลาสที่ implement interface เดียวกันได้ โดยคลาสแรกเป็นการ implement behavior และทำงานบนเซิร์ฟเวอร์ ส่วนคลาสที่สองทำหน้าที่เป็นตัวแทนสำหรับ remote service ทำงานบนไคลเอนต์ ดังรูป



รูปที่ 2.38 แสดงการสนับสนุนให้สามารถสร้าง executable code ของ RMI

สถาปัตยกรรมของ RMI แบ่งเป็นสามระดับคือ ชั้นของ stub และ skeleton ทำหน้าที่ดักจับการเรียกใช้เมธอดจากไคลเอนต์ ซึ่งเรียกไปที่ interface reference variable โดยจะเปลี่ยนการเรียกเหล่านั้นไปที่ remote RMI service แทน ในระดับชั้นถัดมาคือ Remote Reference Layer จะแปลและจัดการ การอ้างอิงที่ไคลเอนต์สร้างไปหา remote service objects ส่วนในระดับล่างสุดจากมุมมองของ RMI ก็คือ TCP/IP นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 แสดงสถาปัตยกรรมของ RMI

เมื่อต้องการใช้งาน ต้องมีการตั้งชื่อให้กับ remote object ก่อน ซึ่ง RMI เองก็สามารถใช้ directory service ได้หลายระบบ และจะต้องมี Java Naming and Directory Interface (JNDI) ด้วย ถ้าจะใช้งานแบบง่าย ๆ RMI ก็มี service ของตัวเองคือ RMI Registry ให้บริการบนพอร์ต 1099 โดยมีขั้นตอนการทำงาน ดังนี้ เซิร์ฟเวอร์จะทำการเรียก registry เพื่อทำการเชื่อมต่อ remote object โดยการใช้ชื่อในการเรียกส่วน ไคลเอนต์จะทำการค้นหา remote object โดยใช้ชื่อที่มี server register และร้องขอ method มาใช้

RMI applications ส่วนใหญ่ประกอบไปด้วยโปรแกรมสองส่วน คือ ส่วนที่เป็นเซิร์ฟเวอร์ และ ไคลเอนต์ โดยส่วนที่เป็น Server application จะทำหน้าที่สร้าง remote objects ทำการอ้างอิงการเข้าถึงของ remote object เหล่านั้น และรอให้ฝั่งไคลเอนต์เรียกใช้เมธอด ส่วนทางด้านฝั่ง client application จะได้รับค่าที่ใช้อ้างอิงถึง remote objects ที่อยู่บนฝั่งเซิร์ฟเวอร์แล้วก็จะทำการเรียกใช้เมธอดทางด้าน RMI จัดให้มีกลไกที่ทำให้ฝั่งเซิร์ฟเวอร์และไคลเอนต์สามารถทำการติดต่อกัน และส่งค่า information ระหว่างกันได้ ในบางครั้งโปรแกรมประเภทนี้เรียกว่า Distributed object application โดยที่

Java Class ที่ฝั่งเซิร์ฟเวอร์ ประกอบด้วย

1. RemoteImplement.class
2. RemoteInterface.class
3. policy.txt

Java Class ที่ฝั่งไคลเอนต์ ประกอบด้วย

1. RemoteClient.class
2. RemoteInterface.class
3. policy.txt

RMI Server มีหน้าที่ดังนี้

1. ทำการ compile เพิ่ม Interface Class (`javac RemoteInterface.java`)
2. ทำการ compile เพิ่ม Server Class (`javac RemoteImplement.java`)

เอกสารนี้เป็นเอกสารทงสวนวสสำหรับกรใชงานเพอการศกษาเท่านั้น ไมออนุญาตให้นำไปใชประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการ generate เพิ่ม Stub Class (rmic -v1.2 RemoteImplement)

RMI Client มีหน้าที่ดังนี้

1. ทำการคัดสำเนา RemoteInterface.class และ RemoteImplement_Stub.class จากฝั่งเซิร์ฟเวอร์มารวมไว้ฝั่งไคลเอนต์

2. ทำการคอมไพล์เพิ่ม Client Class (javac RemoteClient.java)

ขั้นตอน Start RMI

1. ฝั่งเซิร์ฟเวอร์ให้ run โปรแกรม rmiregistry

2. ฝั่งเซิร์ฟเวอร์ให้ run โปรแกรม

```
java -Djava.rmi.server.codebase=http://www.jinisoft.com/jininetwork/ -
```

```
Djava.security.policy=policy.txt RemoteImplement
```

3. ฝั่งไคลเอนต์ให้ run โปรแกรม

```
java -Djava.rmi.server.codebase=http://www.jinisoft.com/jininetwork/ -
```

```
Djava.security.policy=policy.txt RemoteClient
```


บทที่ 3

การออกแบบโปรแกรม

ในการออกแบบนั้นเราจะใช้ยูเอ็มแอล (UML : Unified Modeling Language) มาเป็นเครื่องมือช่วยในการออกแบบ

3.1 ขอบเขตของระบบการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร

โปรแกรมจะแบ่งออกเป็น 2 ส่วน คือส่วนที่อยู่บนเครื่องไคลเอนต์ และส่วนที่อยู่บนเครื่องเซิร์ฟเวอร์ เมื่อผู้ใช้งานต้องการใช้โปรแกรม ผู้ใช้จะสามารถเข้าใช้ได้ทั้งที่เครื่องไคลเอนต์เครื่องใดก็ได้ เมื่อโปรแกรมถูกเปิดขึ้น ผู้ใช้จะต้องทำการใส่ชื่อผู้ใช้และรหัสผ่าน โดยโปรแกรมจะนำชื่อผู้ใช้และรหัสผ่านนี้ไปทำการตรวจสอบในฐานข้อมูลผู้ใช้ เมื่อชื่อผู้ใช้และรหัสผ่านถูกต้อง โปรแกรมจะทำการแจ้งว่าวันนี้มีกำหนดการที่จะต้องทำก็กำหนดการ และมีเอกสารเข้าใหม่กี่ฉบับ โดยถ้าผู้ใช้งานต้องการทราบรายละเอียดก็ให้ไปเลือกเข้าเมนูการทำงานที่ต้องการ

การทำงานหลักแบ่งได้ดังนี้

3.1.1 ระบบการจัดการสมาชิก เป็นการกำหนดสิทธิ์ในการทำงานของพนักงาน ซึ่งงานในแต่ละงานพนักงานอาจมีสิทธิ์ต่างกัน ซึ่งขึ้นอยู่กับกลุ่มงาน และตำแหน่งของพนักงานในงานนั้นๆ

3.1.2 ระบบการจัดการเอกสาร แบ่งการทำงานออกเป็น 2 ส่วน คือ

3.1.2.1 การส่งเอกสาร ซึ่งภายในเอกสารที่ทำการส่งนั้นจะทำการเก็บชื่อผู้ส่ง เพื่อความสะดวกแก่การรับรู้ว่าเอกสารชุดนี้มาจากบุคคลใด

3.1.2.2 การรับเอกสาร

3.1.3 การจัดการตารางเวลา แบ่งการทำงานออกเป็น 2 ส่วน คือ

3.1.3.1 ระบบการจัดการตารางเวลาส่วนบุคคล เป็นตารางเวลาที่เก็บกำหนดการทำงานของผู้ใช้ ซึ่งการสร้างกำหนดการแบ่งเป็น 2 ส่วน คือ ส่วนแรกผู้ใช้เป็นผู้กำหนดกำหนดการในตารางเวลาขึ้นมาเองนั้น ซึ่งผู้ใช้สามารถกำหนดให้มีการเตือนหมายกำหนดการได้ และอีกส่วนคือกำหนดการจะถูกส่งมายังตารางเวลา โดยผู้ใช้ไม่ใช่ผู้กำหนดเอง โดยจะมีการกำหนดให้มีการเตือนกำหนดการนัดหมาย

3.1.3.2 ระบบการจัดการตารางนัดหมายภายในองค์กร เป็นการนำตารางเวลาส่วนบุคคลของผู้ใช้ที่อยู่ในกลุ่มที่ต้องการนัดมาประยุกต์ใช้ ซึ่งจะเกิดขึ้นเมื่อมีผู้ร้องขอในการจัดตารางเวลา ซึ่งผู้ร้องขอนั้นจะต้องเป็นผู้ที่มีสิทธิ์ในการจัดตารางเวลานัดหมาย หรือบุคคลที่มีตำแหน่งเป็นผู้บริหารภายในองค์กรเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 กิจกรรมหลัก

3.2.1 กิจกรรมระบบการจัดการสมาชิก

3.2.1.1 เมื่อผู้ใช้ทำการเปิดระบบการจัดการสมาชิก ระบบจะแสดงรายการกิจกรรม (เพิ่มกลุ่มงานใหม่ แก้ไขกลุ่มงาน ลบกลุ่มงาน) แล้วให้ผู้ใช้เลือกกิจกรรมที่ต้องการ

3.2.1.2 ในกรณีที่ผู้ใช้เลือก เพิ่มกลุ่มงานใหม่

- 1) ผู้ใช้ตั้งชื่อกิจกรรม
- 2) ระบบทำการร้องขอข้อมูลรายชื่อพนักงานจากฐานข้อมูลผู้ใช้ และแสดงรายชื่อพนักงาน
- 3) ผู้ใช้เลือกพนักงานที่ต้องการจัดเข้ากลุ่ม
- 4) ผู้ใช้กำหนดสิทธิ์ของพนักงานแต่ละคนภายในกลุ่มงาน

3.2.1.3 ในกรณีที่ผู้ใช้เลือก แก้ไขกลุ่มงาน

- 1) ระบบทำการร้องขอข้อมูลรายชื่อกลุ่มงานที่มีในระบบจากฐานข้อมูลผู้ใช้ และแสดงรายชื่อกลุ่มงาน
- 2) ผู้ใช้เลือกกลุ่มงานที่ต้องการแก้ไข
- 3) ระบบทำการร้องขอข้อมูลรายชื่อพนักงานจากฐานข้อมูลผู้ใช้ และแสดงรายชื่อพนักงาน
- 4) ผู้ใช้เลือกพนักงานที่ต้องการจัดเข้ากลุ่ม หรือลบออกจากกลุ่ม
- 5) ผู้ใช้กำหนดสิทธิ์ของพนักงานแต่ละคนภายในกลุ่มงาน

3.2.1.4 ในกรณีที่ผู้ใช้เลือก ลบกลุ่มงาน

- 1) ระบบทำการร้องขอข้อมูลรายชื่อกลุ่มงานที่มีในระบบจากฐานข้อมูลผู้ใช้ และแสดงรายชื่อกลุ่มงาน
- 2) ผู้ใช้เลือกกลุ่มงานที่ต้องการลบ
- 3) ระบบทำการลบกลุ่มงาน รายชื่อพนักงานภายในกลุ่ม และสิทธิ์ของพนักงานที่มีภายในกลุ่มที่ผู้ใช้เลือกออกจากฐานข้อมูลผู้ใช้

3.2.2 กิจกรรมระบบการจัดการเอกสาร

3.2.2.1 กิจกรรมการส่งเอกสาร ก่อนจะทำกิจกรรมการส่งเอกสารนั้นผู้ใช้จะต้องเปิดโปรแกรมและ login ก่อน โดยจะต้องใส่ชื่อผู้ใช้ (username) และรหัสผ่าน (password) ให้ถูกต้องเสียก่อน จึงจะทำการส่งเอกสารได้ โดยมีกิจกรรมหลัก ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ทำการสร้างเอกสารที่ต้องการส่งก่อน โดยที่การสร้างเอกสารมี ซึ่งรูปแบบของการสร้างเอกสารอยู่ 2 แบบด้วยกัน คือ รูปแบบแรกจะเป็นการสร้างเอกสารใหม่ เพื่อให้ผู้ส่งเอกสารสามารถกำหนดรายละเอียดต่าง ๆ ของเอกสารนั้นด้วยตนเอง ส่วนรูปแบบที่สองจะเป็นการนำข้อความจากเอกสารเก่าที่ผู้ส่งทำการเปิดอ่านมาก่อนมาแสดง โดยแบ่งออกตามความต้องการของผู้ใช้ว่าต้องการที่จะตอบกลับ (Reply) หรือต้องการที่จะส่งต่อไปยังกลุ่มผู้รับกลุ่มอื่นๆ (Forward) ถ้าเป็นการตอบกลับนั้นจะทำการกำหนดที่อยู่ของผู้รับ โดยนำค่าที่อยู่ของผู้ที่ส่งเอกสารนั้นมาให้มา กำหนดค่า พร้อมกับแสดงข้อความที่ผู้ส่งส่งมาให้ด้วย ส่วนการส่งต่อนั้นจะทำการแสดงข้อความก่อนหน้าเช่นเดียว และผู้ใช้สามารถทำการพิมพ์เอกสารหรือข้อความที่ต้องการส่งต่อไปเพิ่มเติมได้

2) จากนั้นถ้าผู้ส่งเอกสารต้องการแนบไฟล์เอกสารก็สามารถเลือกไฟล์เอกสารมาแนบ โดยให้ส่งไปพร้อมกับเอกสารที่ต้องการส่งด้วยก็ได้

3) โปรแกรมจะทำการแสดงรายละเอียดของเอกสารให้ผู้ส่งตรวจสอบความถูกต้อง ก่อนจะทำการส่งเอกสารนั้น ๆ

4) กำหนดกลุ่มผู้รับข้อความหรือเอกสารนั้น ๆ

5) ทำการสร้างเอกสาร โดยการบันทึกเอกสารนั้นลงในฐานข้อมูลเอกสารของผู้รับแต่ละคน

3.2.2.2 การรับเอกสาร

1) หลังจากที่ใช้ได้ทำการเปิดโปรแกรมแล้ว จะได้รับแจ้งว่ามีเอกสารใหม่ที่ส่งมาให้ผู้ใช้นั้นจำนวนเท่าใด

2) เมื่อผู้ใช้ต้องการอ่านเอกสารที่ได้รับ ก็จะทำการเข้าไปสู่กล่องรับข้อความ ซึ่งกล่องข้อความจะแสดงรายการเอกสารที่ผู้รับได้รับทั้งหมด

3) ทำการเลือกเอกสารที่ต้องการอ่าน

4) หลังจากที่ได้รับทำการอ่านเอกสารแล้วจะทำการตัดสินใจว่า จะจัดการกับเอกสารอย่างไร โดยเลือกจากการลบเอกสาร การตอบกลับ การส่งเอกสารต่อไปยังกลุ่มผู้รับอื่น ๆ หรือยังเก็บเอกสารนั้นไว้ในฐานข้อมูลของผู้รับต่อไป

5) ถ้าผู้รับต้องการลบเอกสาร จะทำการลบเอกสารนั้นจากฐานข้อมูลของผู้รับเอกสาร แต่ถ้าต้องการตอบกลับ หรือ ส่งเอกสารไปยังกลุ่มผู้รับอื่น ๆ แล้ว โปรแกรมจะเข้าสู่การทำงานส่วนของการส่งเอกสารต่อไป

6) จากนั้นทำการตัดสินใจว่าต้องการอ่านเอกสารต่อหรือไม่ ถ้าต้องการอ่านเอกสารต่อโปรแกรมจะแสดงรายการเอกสารอีกครั้ง แต่ถ้าไม่ต้องการอ่านเอกสารแล้วก็จะเสร็จสิ้นการรับเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 กิจกรรมระบบการจัดการตารางเวลา

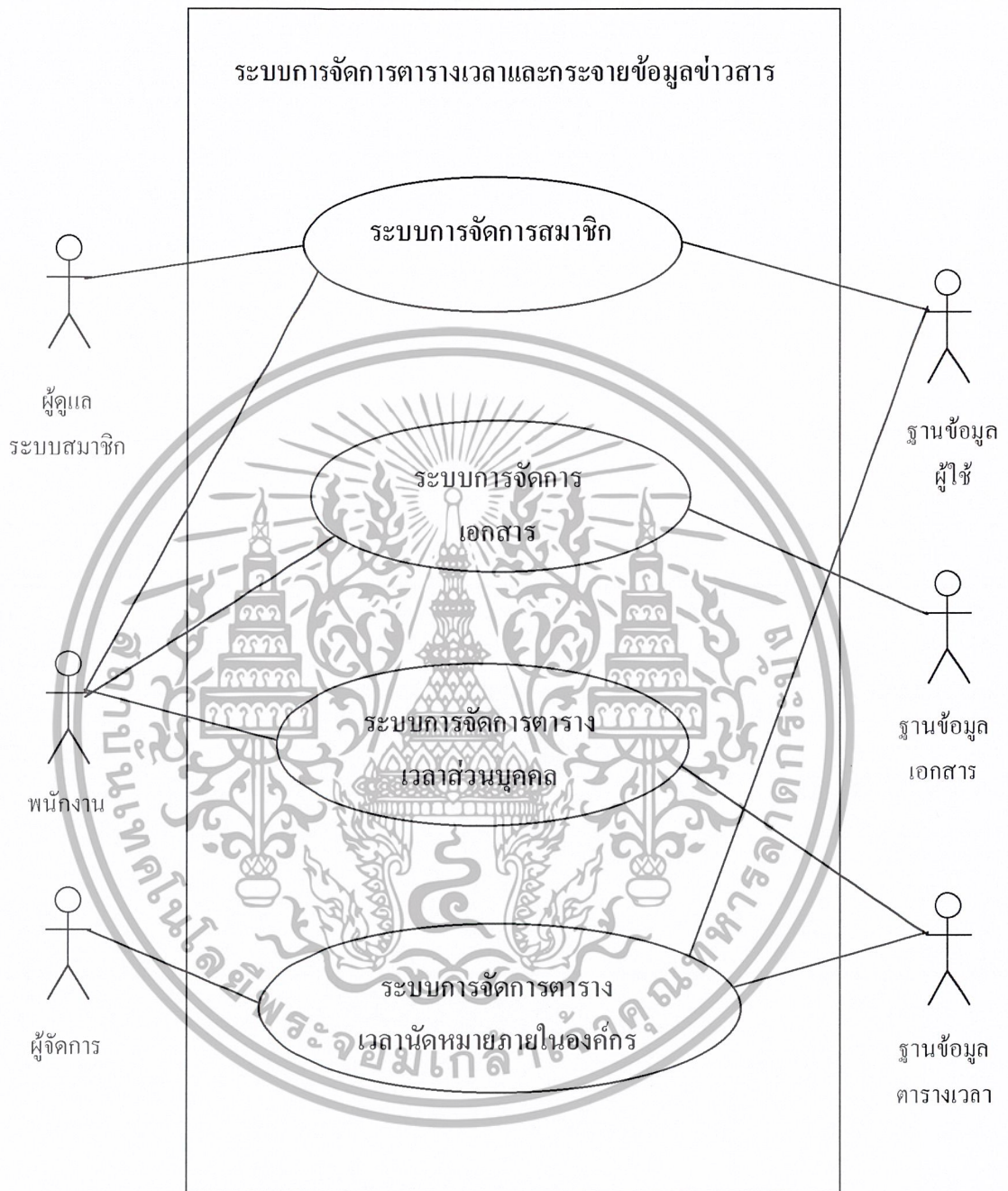
3.2.3.1 กิจกรรมระบบการจัดการตารางเวลาส่วนบุคคล

- 1) เมื่อผู้ใช้ทำการเปิดระบบการจัดการตารางเวลาส่วนบุคคล ระบบจะทำการร้องขอข้อมูลจากฐานข้อมูลตารางเวลา
- 2) แสดงตารางเวลาการทำงานส่วนบุคคล
- 3) ผู้ใช้กำหนดตารางเวลาส่วนบุคคล (เพิ่ม, แก้ไข, ลบ กำหนดการ)
- 4) ผู้ใช้กำหนดการแจ้งเตือนล่วงหน้า
- 5) ส่งข้อมูลที่เปลี่ยนแปลงไปบันทึกยังฐานข้อมูลตารางเวลา

3.2.3.2 ระบบการจัดการตารางเวลานัดหมายภายในองค์กร

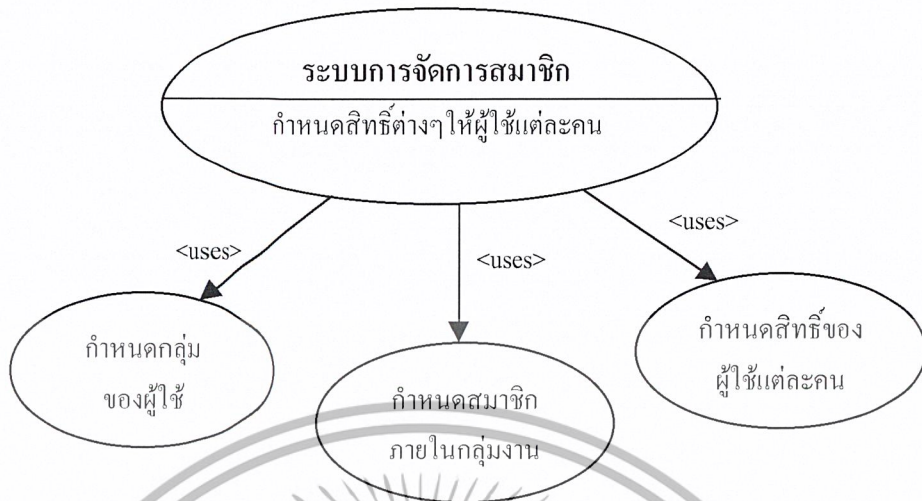
- 1) กำหนดกลุ่มหรือบุคคลที่ต้องการทำการนัดหมาย โดยจะทำการเลือกกลุ่มบุคคลจากฐานข้อมูลผู้ใช้
- 2) กำหนดช่วงเวลาที่ต้องการทำการนัดหมาย
- 3) นำข้อมูลตารางเวลาส่วนบุคคลของกลุ่มบุคคลที่ต้องการทำการนัดหมาย ไปทำการจัดการเวลานัดหมาย โดยเวลาที่จะทำการจัดนั้นต้องอยู่ในช่วงเวลาที่ได้อำนาจกำหนดไว้
- 4) แสดงเวลาที่สามารถนัดพบได้
- 5) เลือกเวลาที่จะแสดงมาเป็นเวลาที่ต้องการทำการนัดพบ
- 6) กำหนดรายละเอียดในการนัดพบ
- 7) ทำการแจ้งให้สมาชิกในกลุ่มที่ต้องการนัดพบทราบกำหนดการ โดยจะทำการกำหนดการนัดหมายนี้ไปยังฐานข้อมูลตารางเวลาของบุคคลนั้นๆ

3.3 ยูสเคสไดอะแกรม (Use Case Diagram)



รูปที่ 3.1 แสดงยูสเคสไดอะแกรมของระบบการจัดการตารางเวลาและกระจายข้อมูลข่าวสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

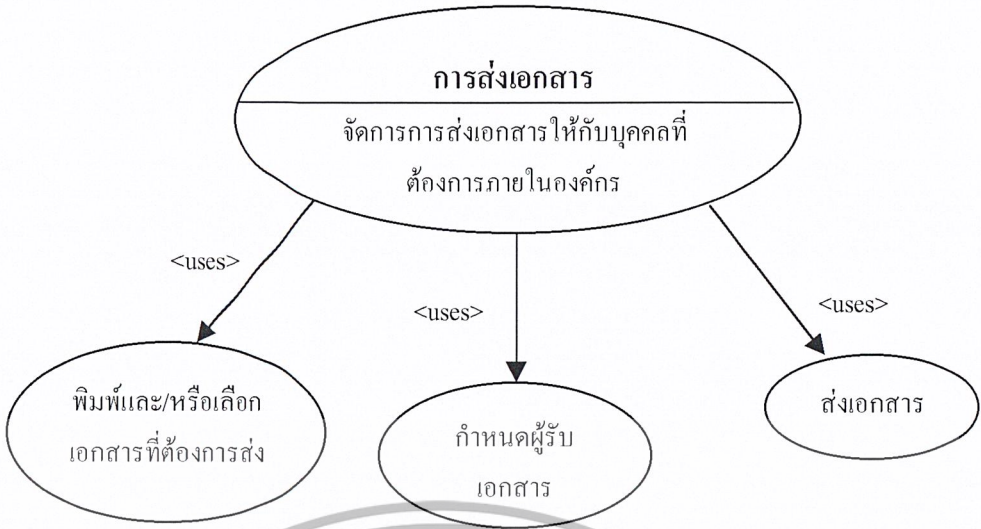


รูปที่ 3.2 แสดงยูสเคสของระบบการจัดการสมาชิก

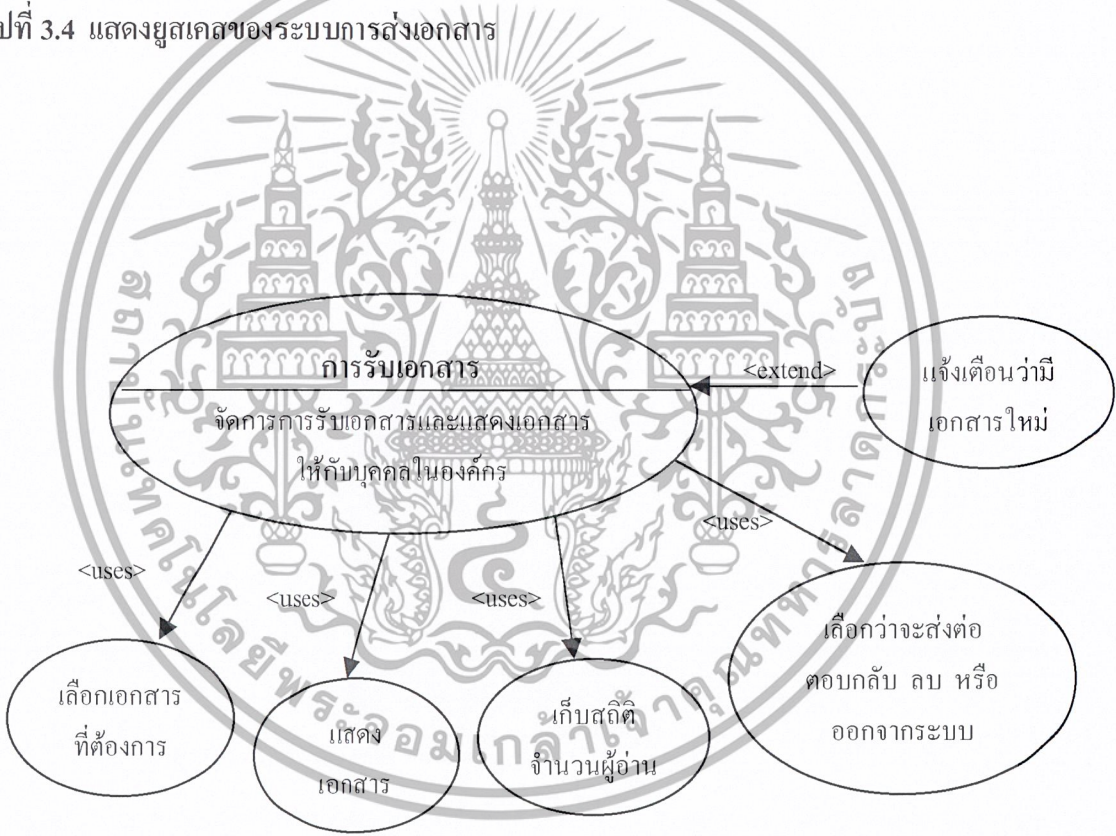


รูปที่ 3.3 แสดงยูสเคสของระบบการจัดการเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

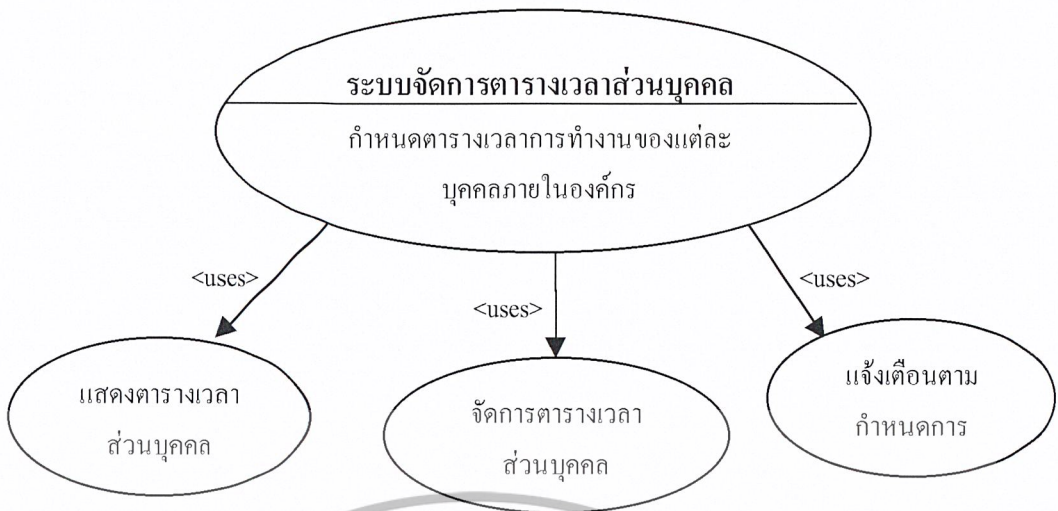


รูปที่ 3.4 แสดงยูสเคสของระบบการส่งเอกสาร

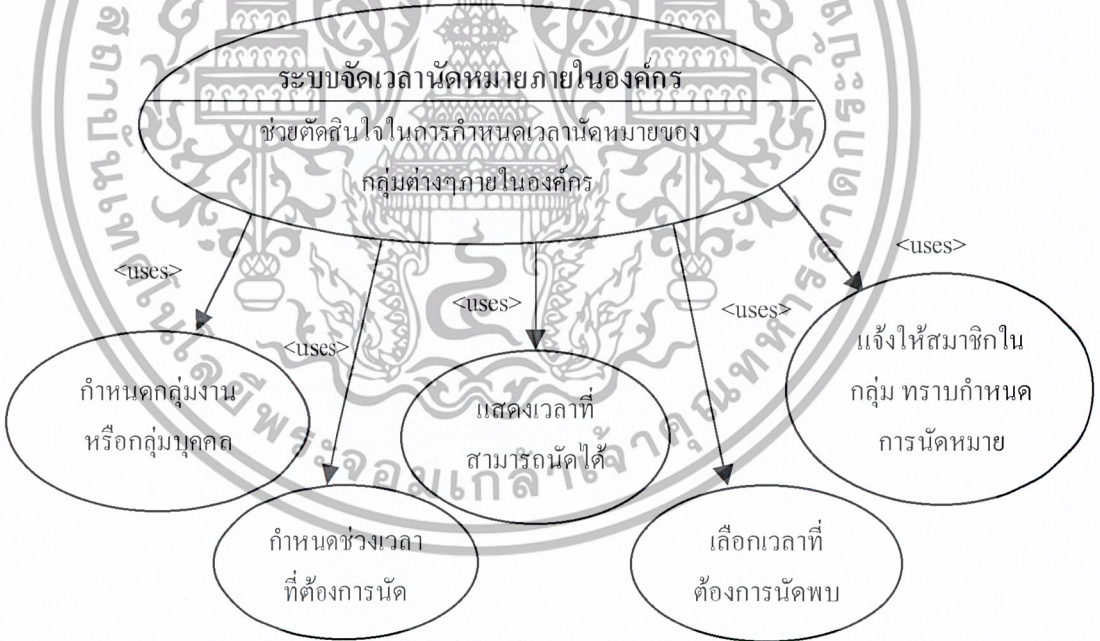


รูปที่ 3.5 แสดงยูสเคสของระบบการรับเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงยูสเคสของระบบจัดการตารางเวลาส่วนบุคคล

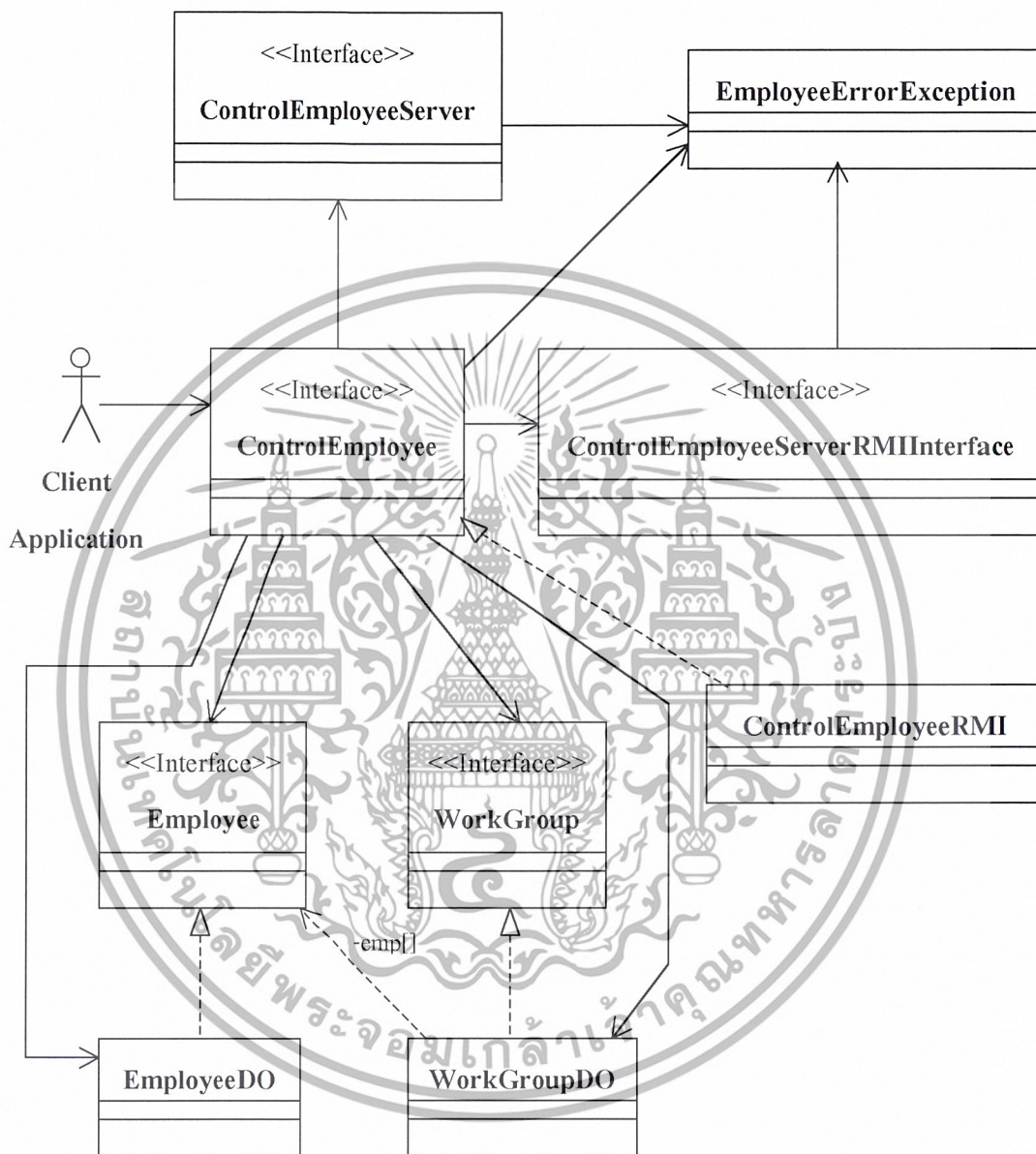


รูปที่ 3.7 แสดงยูสเคสของระบบจัดเวลานัดหมายภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 คลาสไดอะแกรม (Class Diagram)

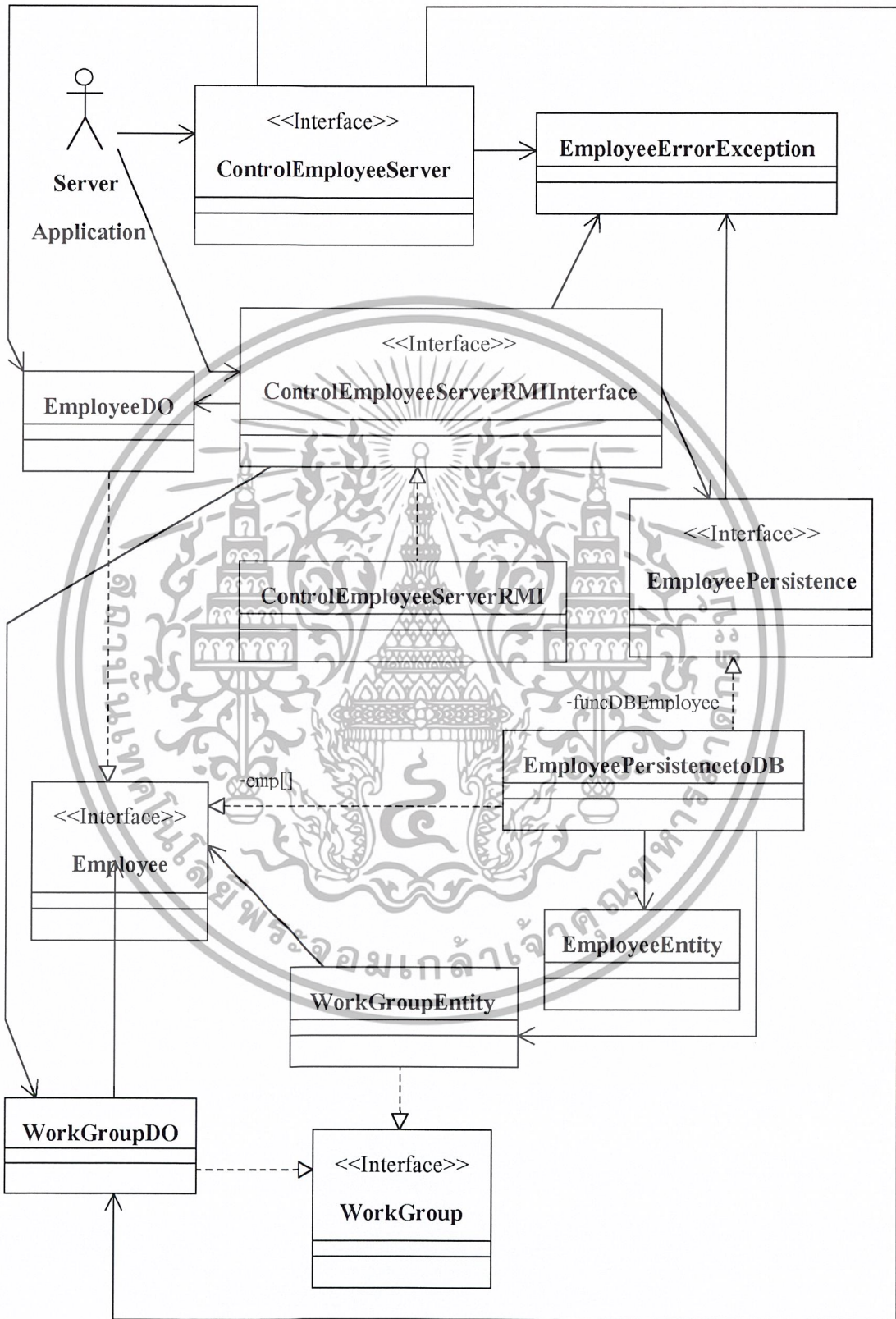
3.4.1 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนการจัดการสมาชิกฟ่งไคลเอนต์



รูปที่ 3.8 แสดงคลาสไดอะแกรมของคอมพิวเตอร์ส่วนการจัดการสมาชิกฟ่งไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

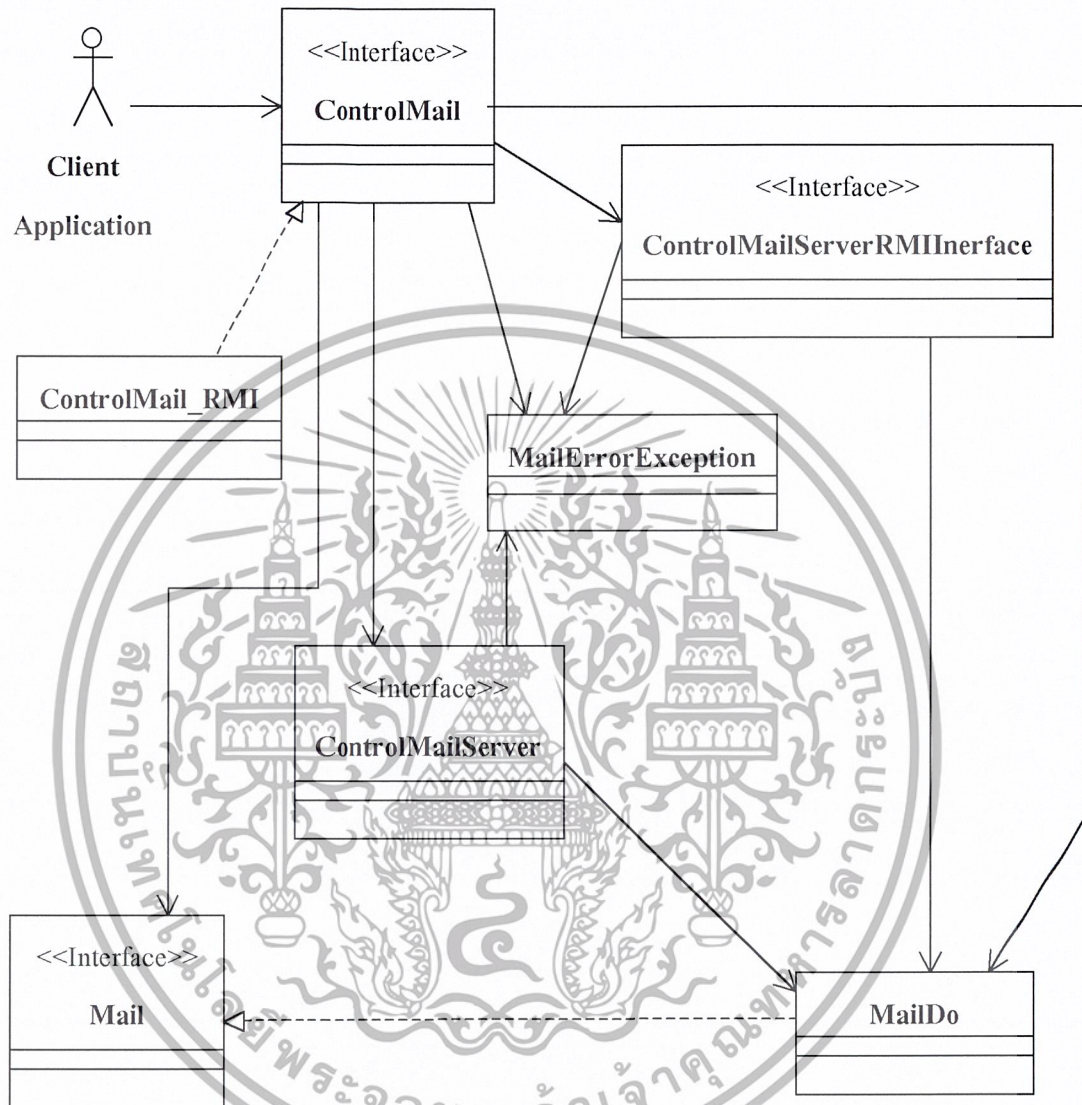
3.4.2 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนการจัดการสมาชิกฝั่งเซิร์ฟเวอร์



รูปที่ 3.9 แสดงคลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนการจัดการสมาชิกฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

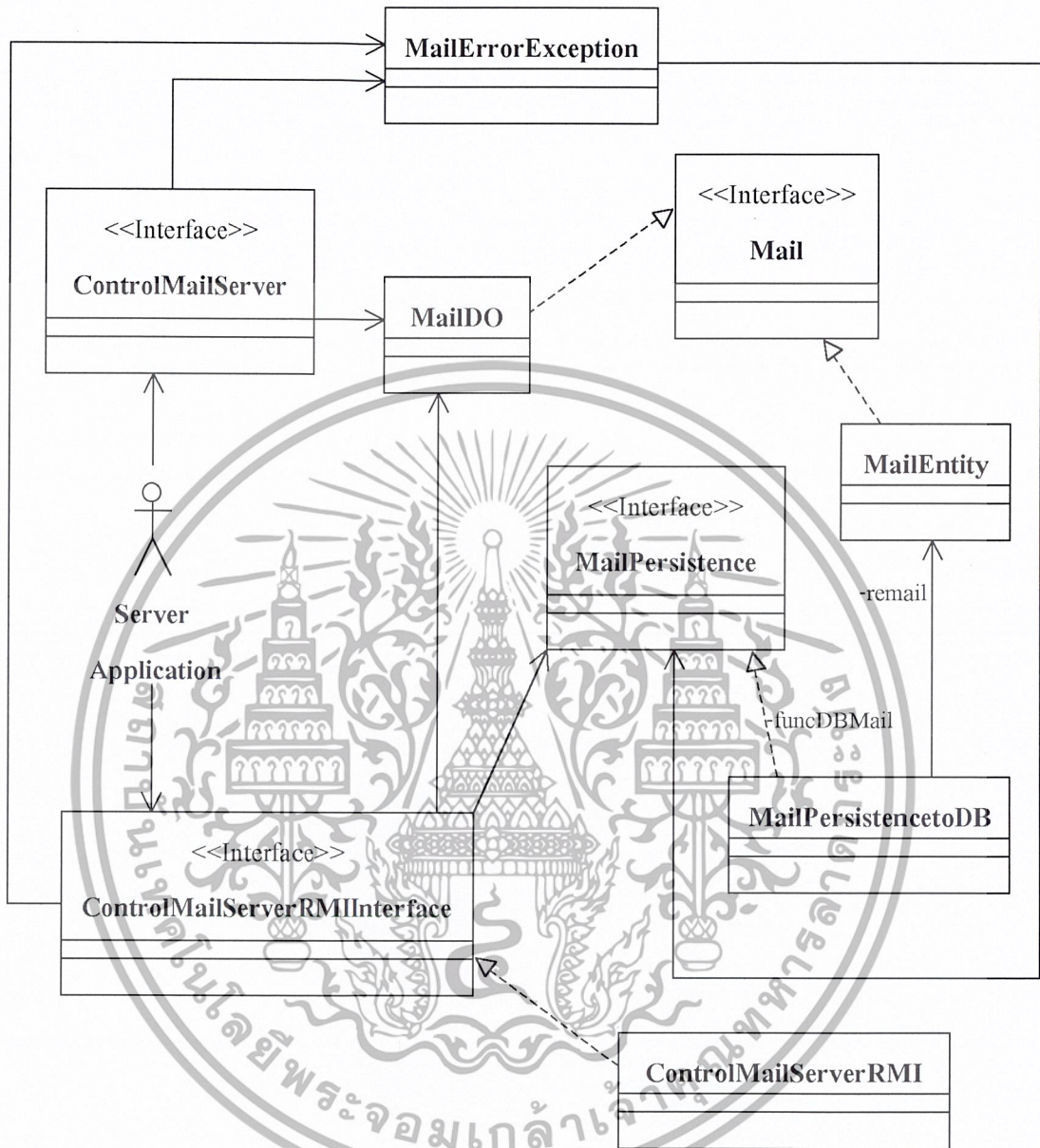
3.4.3 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนกระจายข้อมูลข่าวสารฝั่งไคลเอนต์



รูปที่ 3.10 แสดงคลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนกระจายข้อมูลข่าวสารฝั่งไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

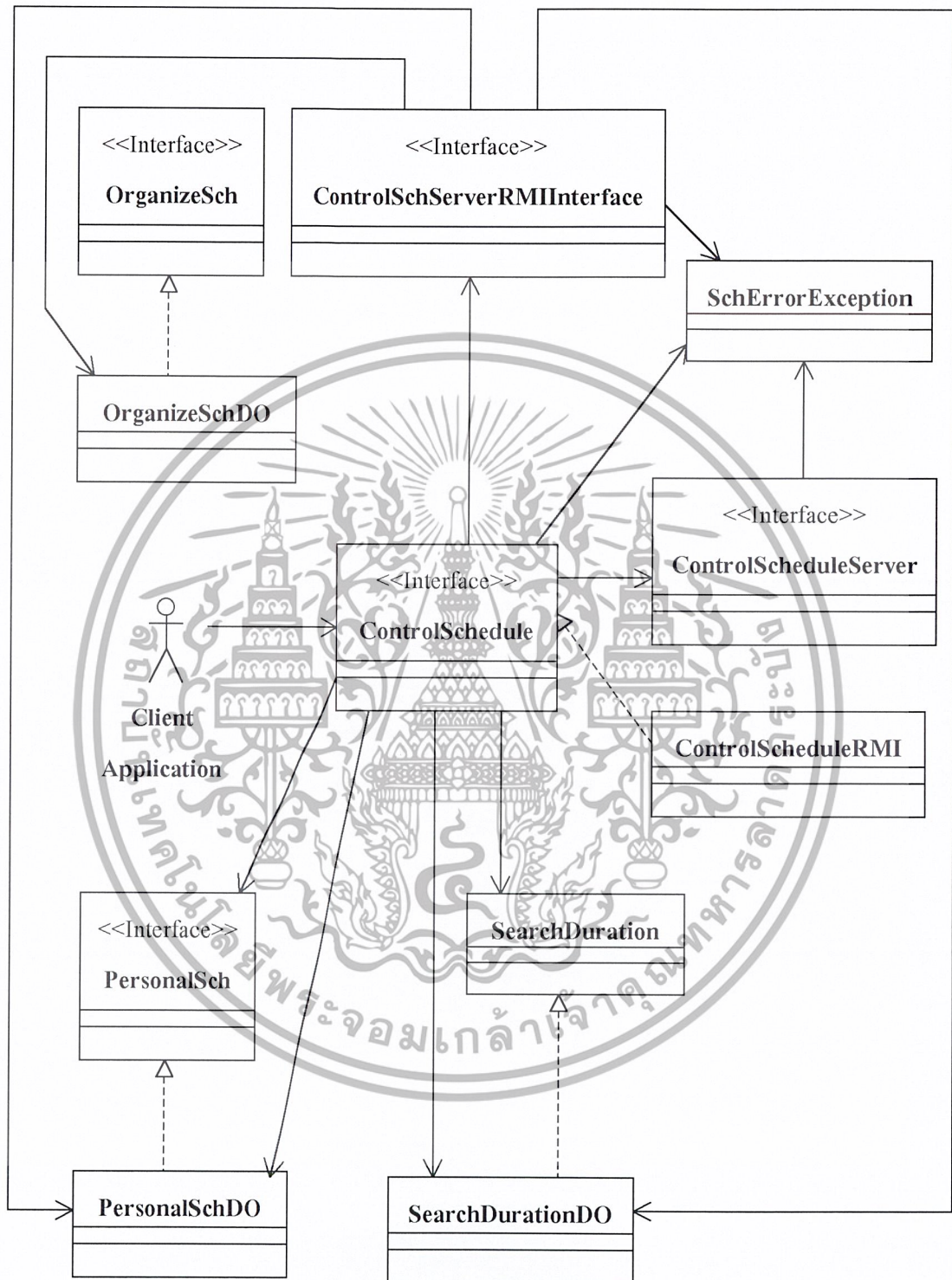
3.4.4 คลาสไดอะแกรมสำหรับคอมโพเนนต์ส่วนกระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์



รูปที่ 3.11 แสดงคลาสไดอะแกรมสำหรับคอมโพเนนต์ส่วนกระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

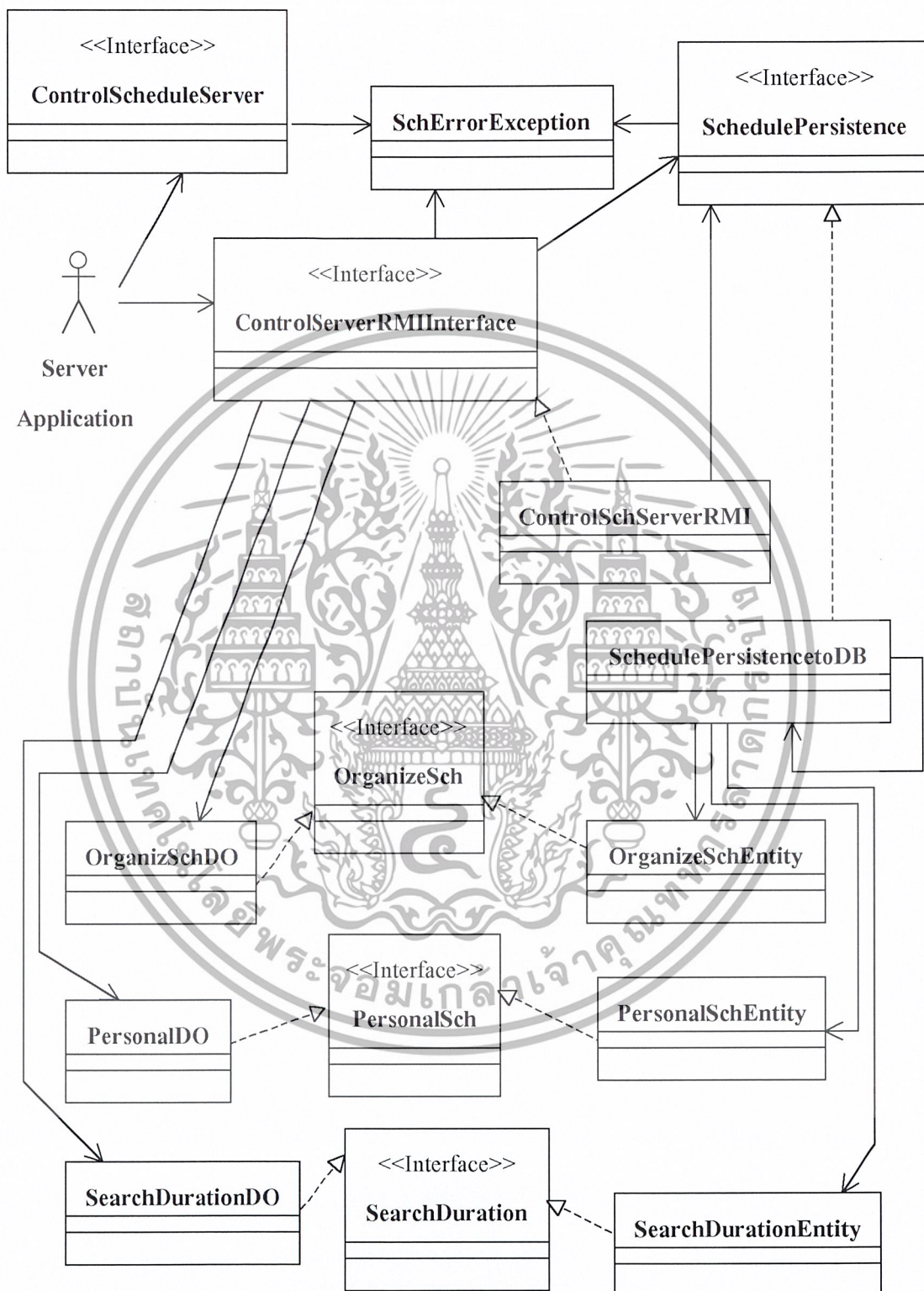
3.4.5 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการตารางเวลาฝั่งไคลเอนต์



รูปที่ 3.12 แสดงคลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการตารางเวลาฝั่งไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.6 คลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการตารางเวลาฝั่งเซิร์ฟเวอร์



รูปที่ 3.13 แสดงคลาสไดอะแกรมสำหรับคอมพิวเตอร์ส่วนจัดการตารางเวลาฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พนักงาน
+ รหัสพนักงาน - ชื่อ - นามสกุล - ชื่อผู้ใช้ - รหัสผ่าน
+ ส่งเอกสาร + รับเอกสาร - สร้างเอกสาร - ขอเข้าใช้ระบบ - แก้ไขข้อมูลส่วนตัว - จัดตารางเวลาด่วนบุคคล - จัดตารางเวลาดำเนินการภายในองค์กร

รูปที่ 3.14 แสดงคลาสไดอะแกรมของพนักงาน

ฐานข้อมูลตารางเวลา
+ รหัสพนักงาน - วันที่นัดหมาย : Date - เวลานั้นนัดหมาย - การเตือนกำหนดการ : Boolean - รายละเอียดกำหนดการ - เวลาในการเตือนล่วงหน้า
- แก้ไขรายละเอียดกำหนดการ - เตือนกำหนดการล่วงหน้า

รูปที่ 3.15 แสดงคลาสไดอะแกรมของฐานข้อมูลตารางเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลเอกสาร
+ รหัสเอกสาร + รหัสกลุ่มผู้รับผิดชอบ - ผู้ส่งเอกสาร - ชื่อเอกสาร - ข้อความเอกสาร - เอกสารที่แนบมา
- แก้ไขเอกสาร - กำหนดสิทธิ์ในการดูเอกสาร - บันทึกข้อมูลในการส่งเอกสาร - เปิดเอกสาร

รูปที่ 3.16 แสดงคลาสไดอะแกรมของฐานข้อมูลเอกสาร

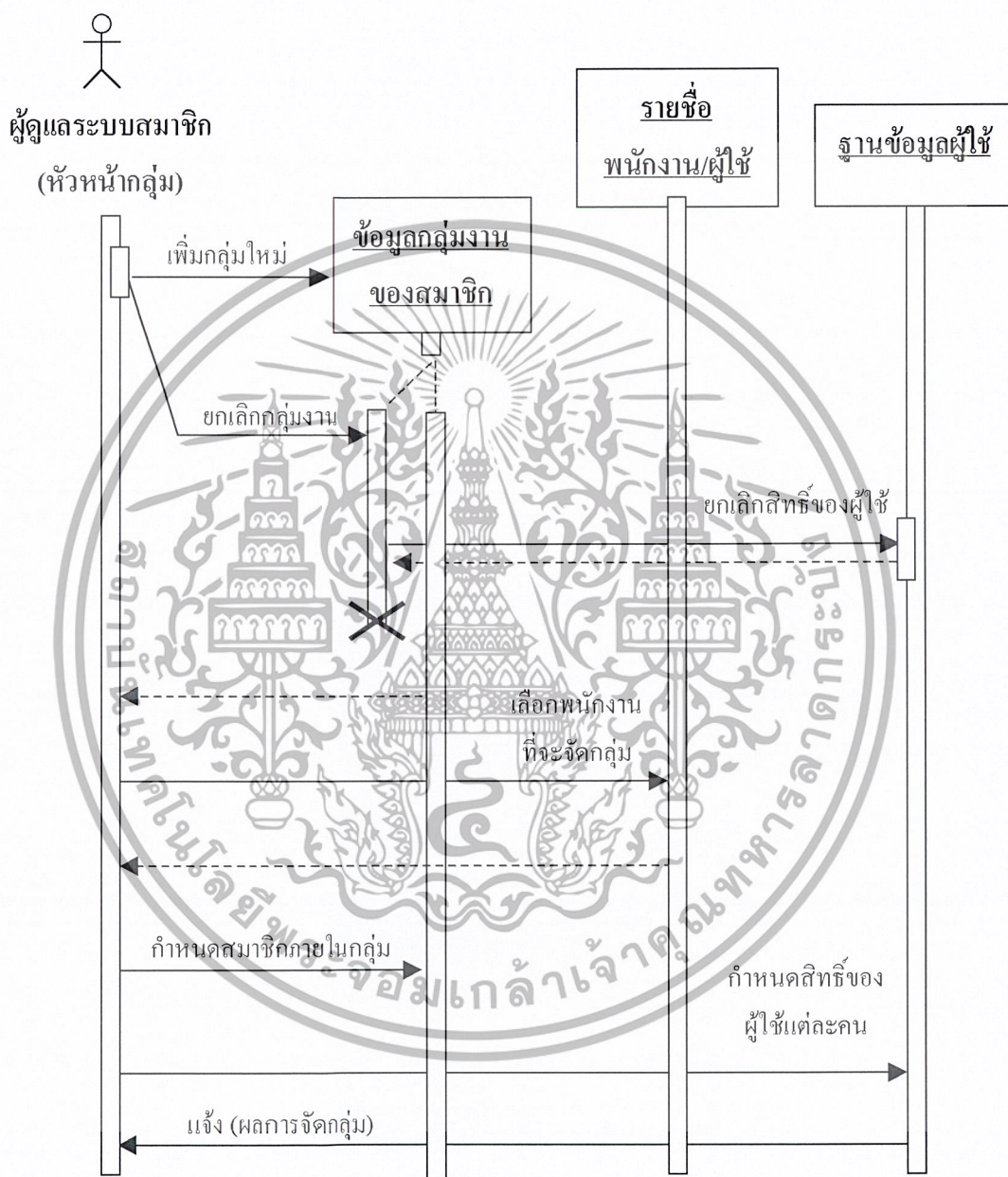
ฐานข้อมูลผู้ใช้
+ รหัสงาน + รหัสพนักงาน + ตำแหน่งของพนักงานในงานนี้ - สิทธิ์ในการเข้าใช้ข้อมูลในงานนี้
- แก้ไขงาน - แก้ไขพนักงาน - กำหนดสิทธิ์ในการเข้าใช้ของพนักงาน - ตรวจสอบชื่อผู้ใช้ และรหัสผ่าน - กำหนดสิทธิ์ในการเข้าใช้แก่พนักงาน

รูปที่ 3.17 แสดงคลาสไดอะแกรมของฐานข้อมูลผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ซีเควนซ์ไดอะแกรม (Sequence Diagram)

3.5.1 ระบบการจัดการสมาชิก

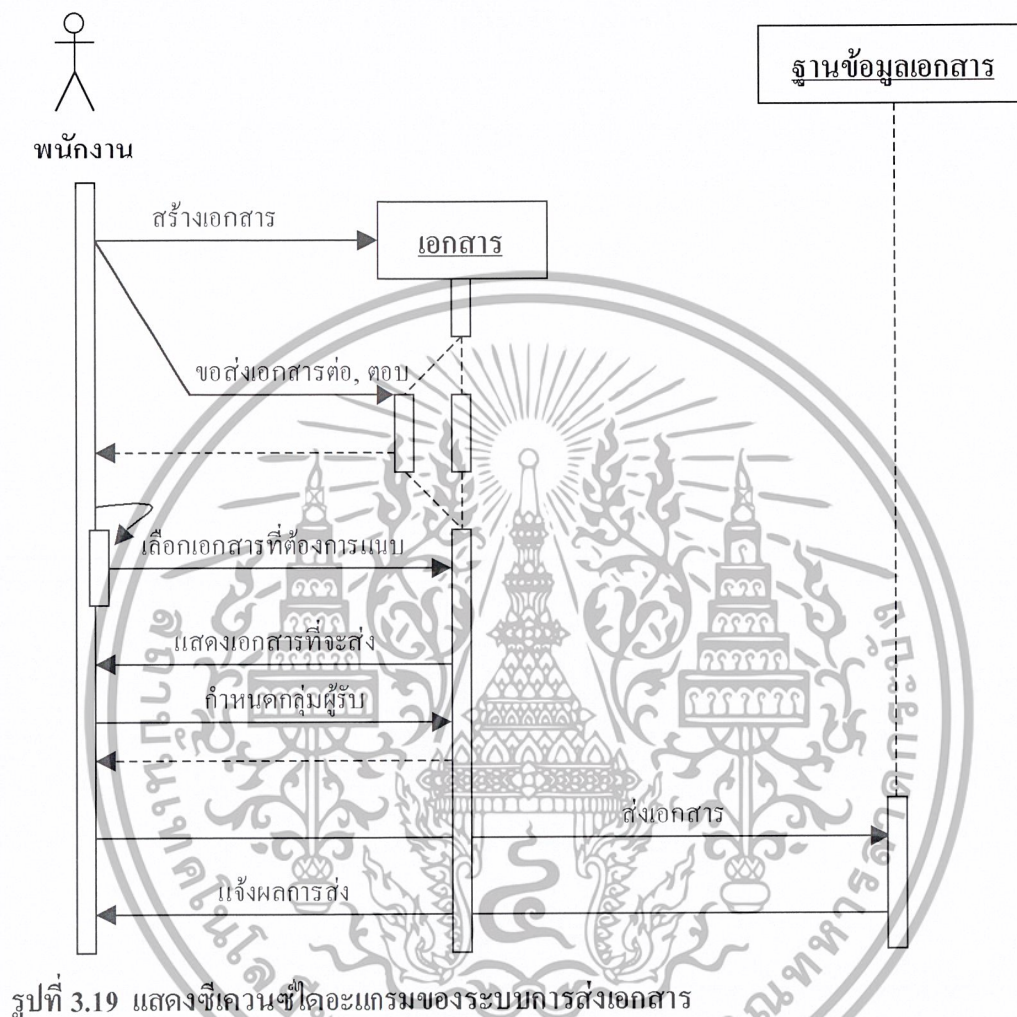


รูปที่ 3.18 แสดงซีเควนซ์ไดอะแกรมของระบบการจัดการสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 ระบบการจัดการเอกสาร

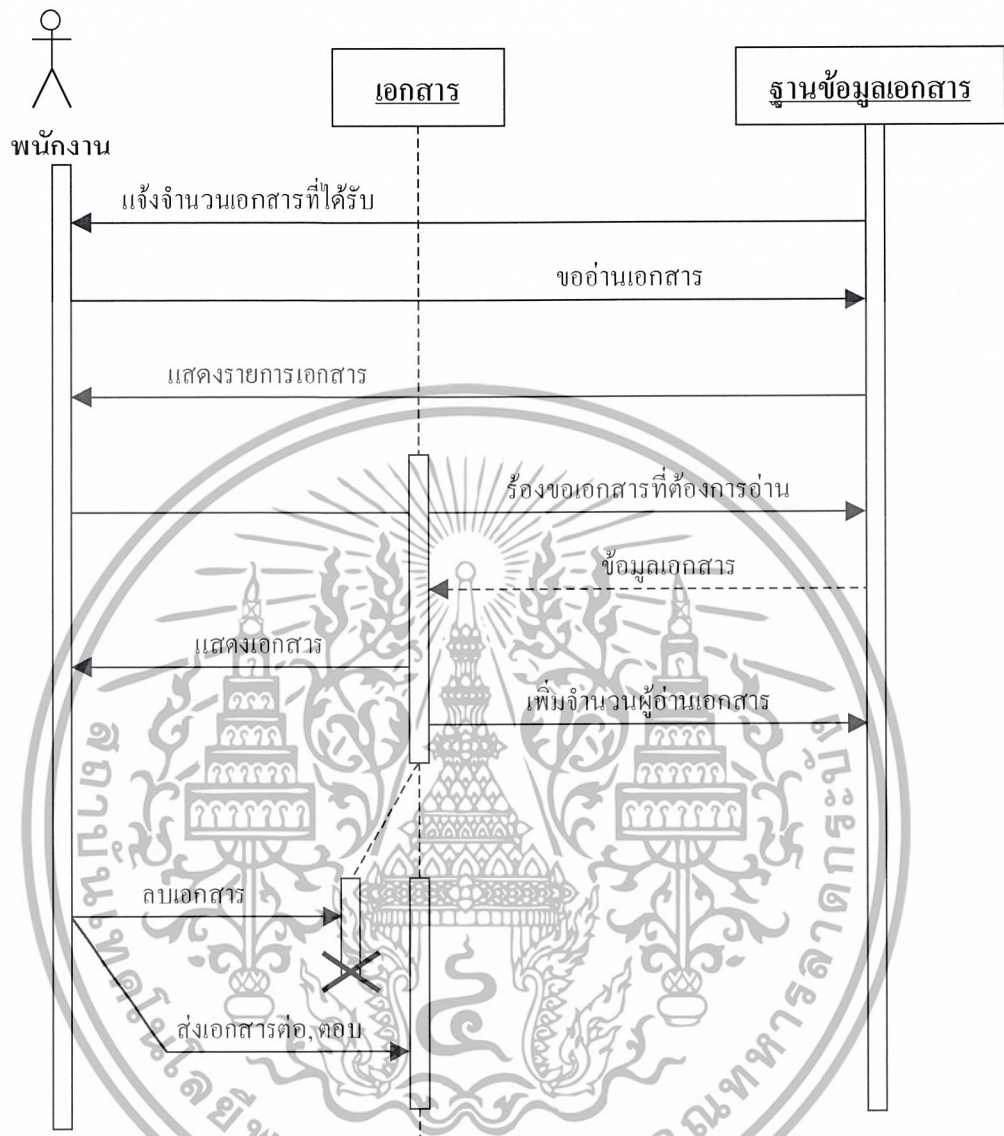
3.5.2.1 การส่งเอกสาร



รูปที่ 3.19 แสดงขั้นตอนชีวิตอะแกรมของระบบการส่งเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.2 การรับเอกสาร

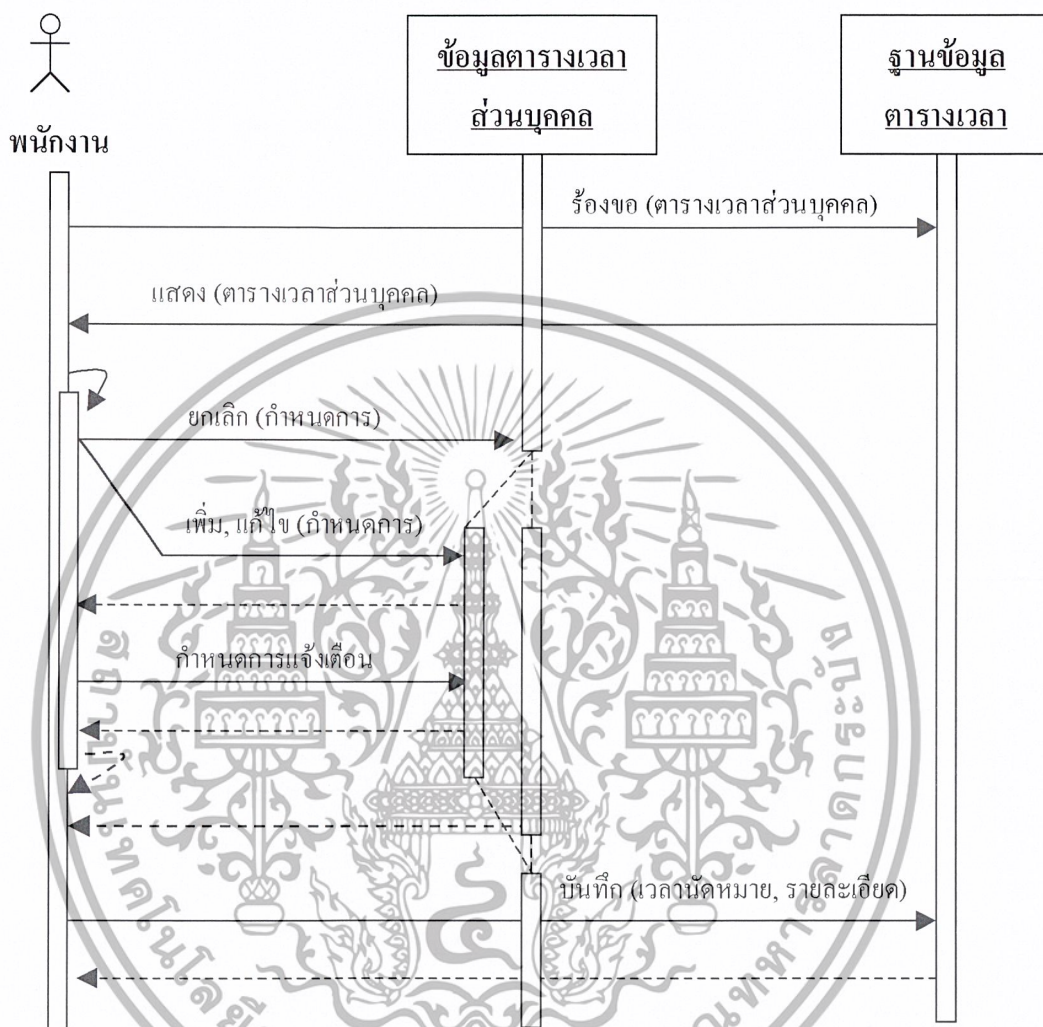


รูปที่ 3.20 แสดงซีเควนซ์ไดอะแกรมของระบบการรับเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

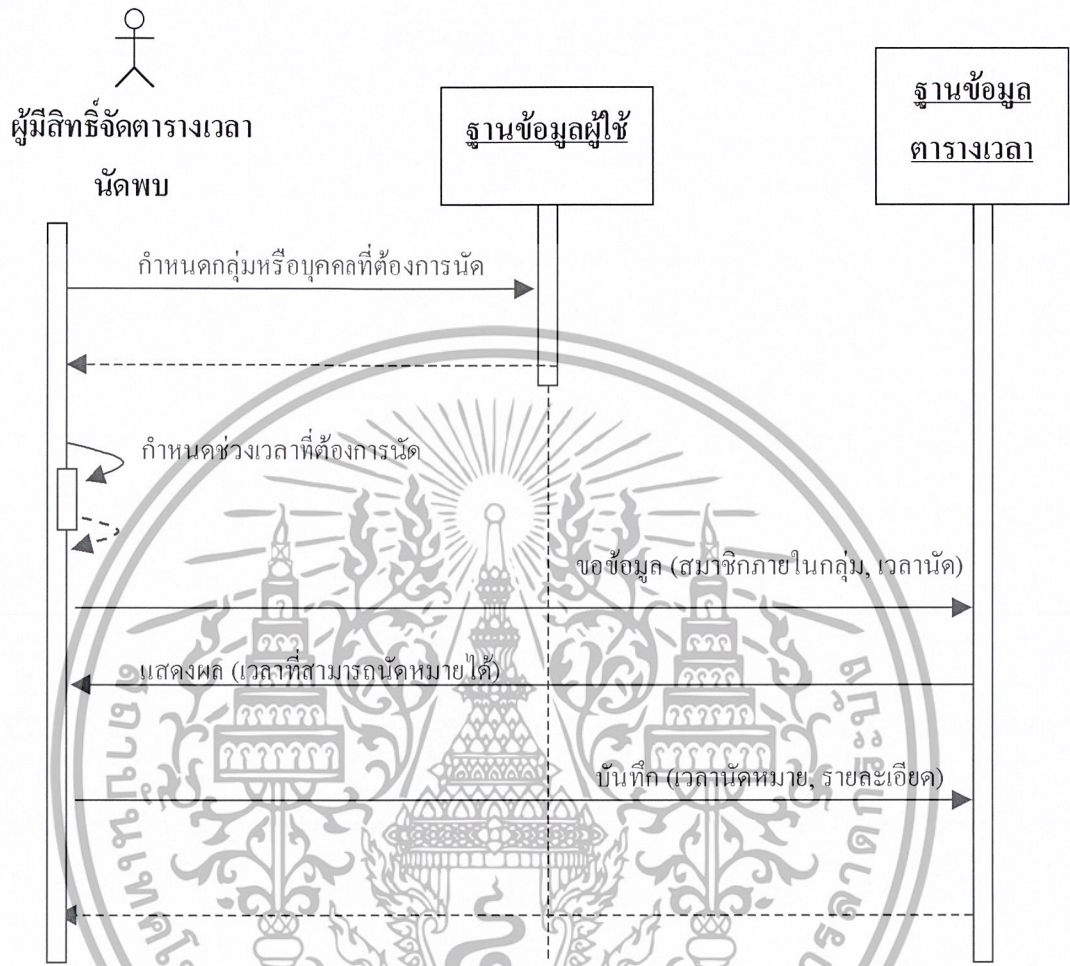
3.5.3 ระบบการจัดการตารางเวลา

3.5.3.1 ระบบการจัดการตารางเวลาส่วนบุคคล



รูปที่ 3.21 แสดงซีเควนซ์ไดอะแกรมของระบบการจัดการตารางเวลาส่วนบุคคล

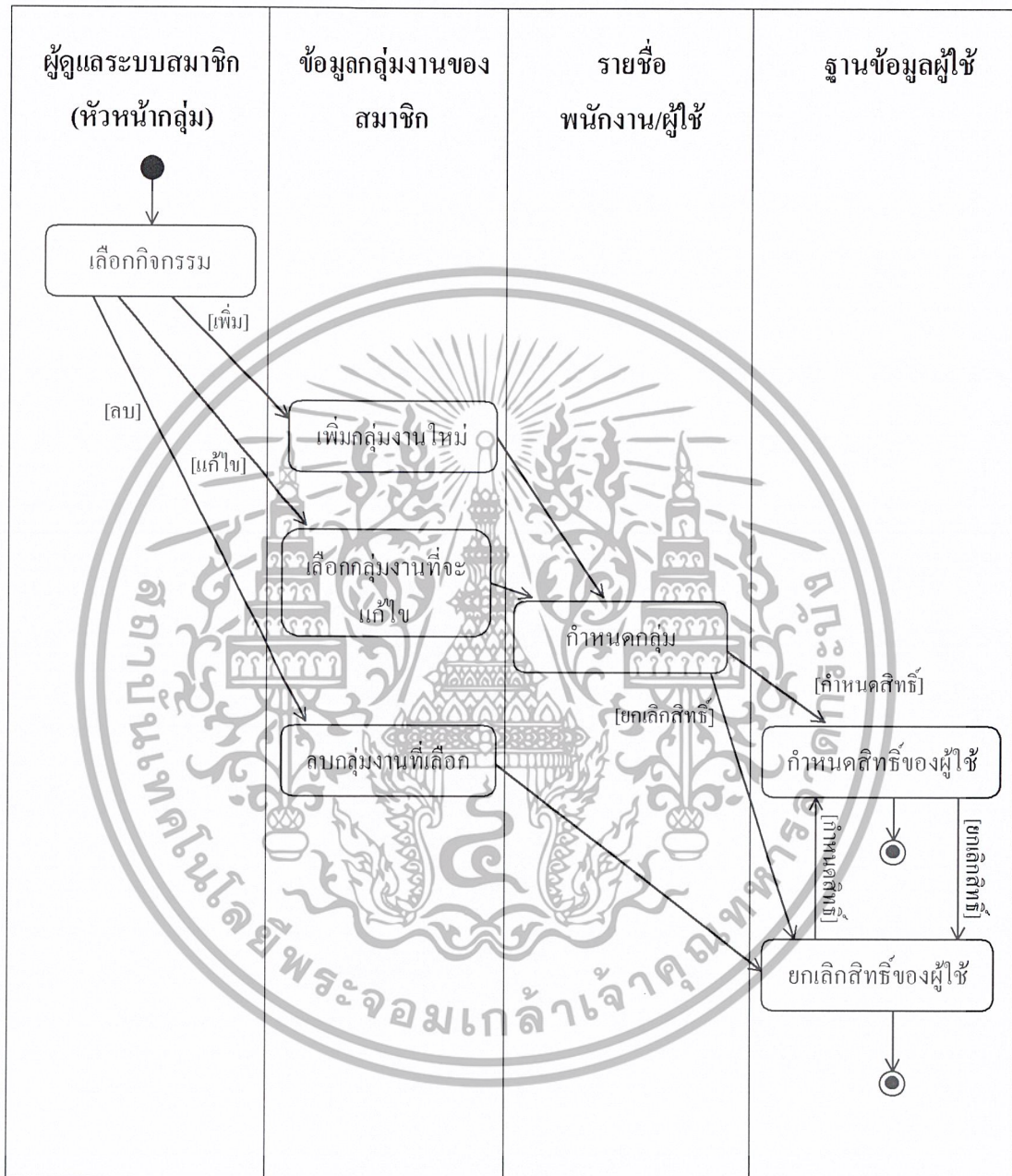
3.5.3.2 ระบบการจัดการตารางเวลานัดหมายภายในองค์กร



รูปที่ 3.22 แสดงซีเควนซ์ไดอะแกรมของระบบการจัดการตารางเวลานัดหมายภายในองค์กร

3.6 แอ็กทิวิตีไดอะแกรม (Activity Diagram)

3.6.1 ระบบการจัดการสมาชิก

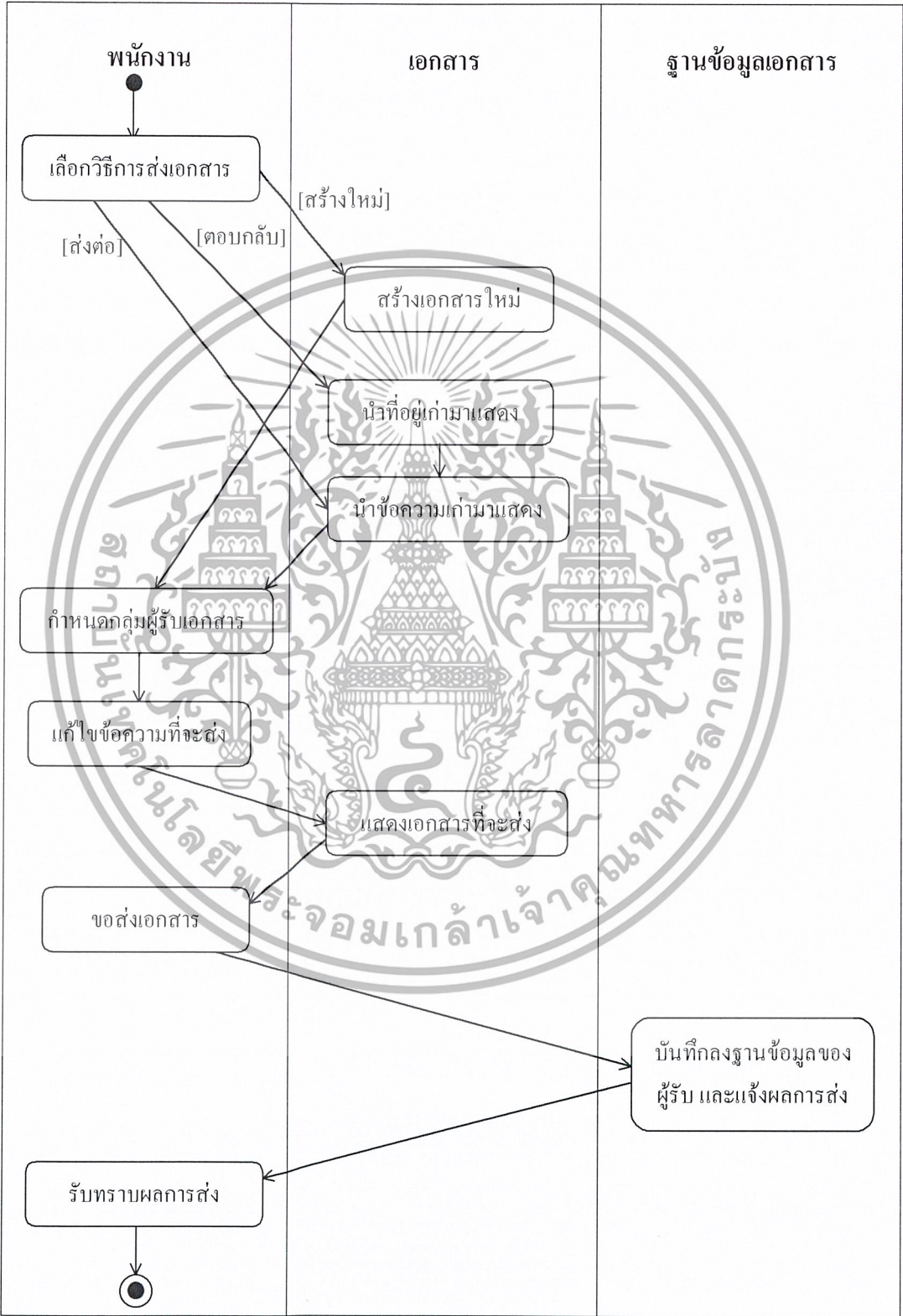


รูปที่ 3.23 แสดงแอ็กทิวิตีไดอะแกรมของระบบการจัดการสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 ระบบการจัดการเอกสาร

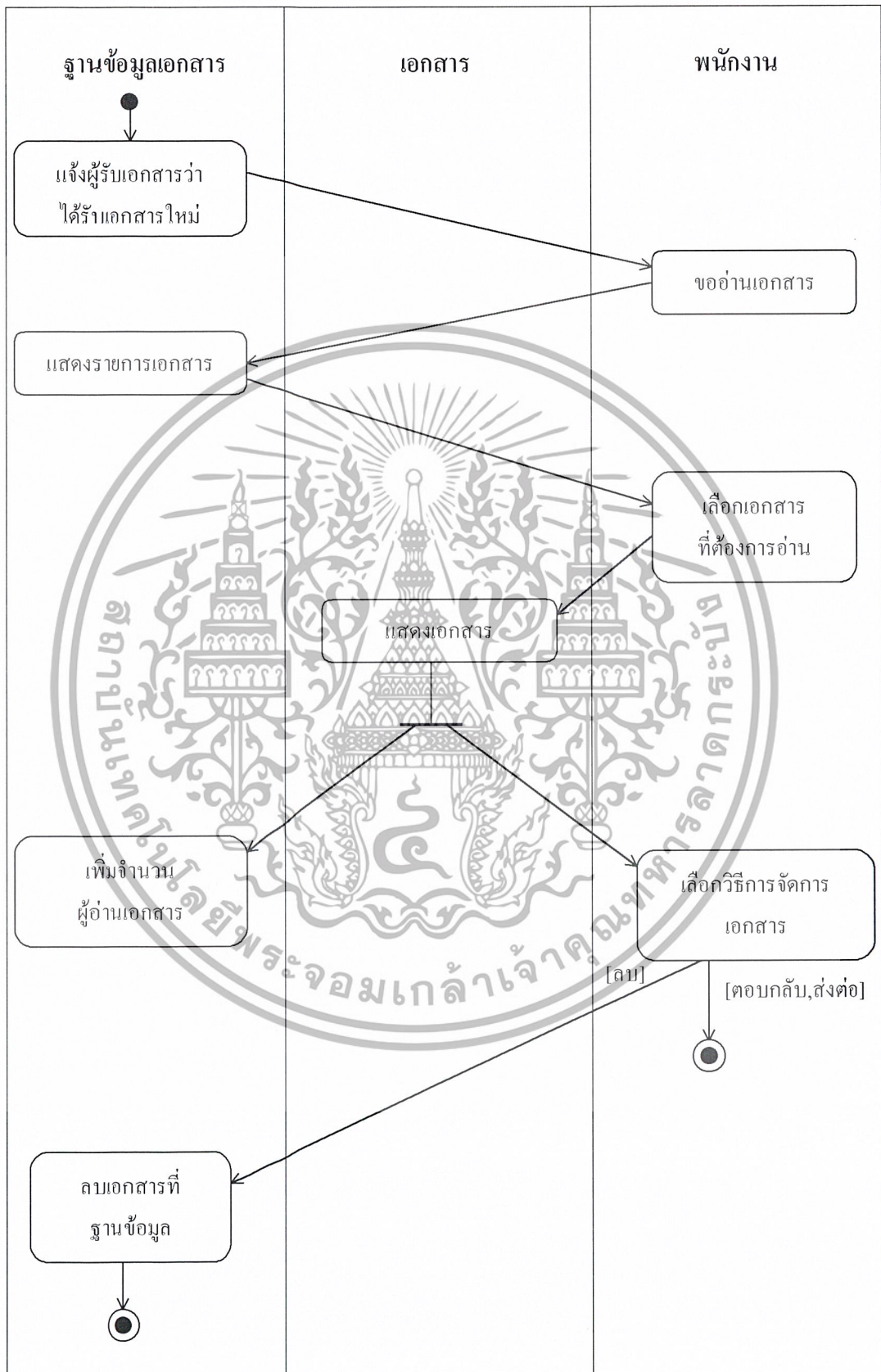
3.6.2.1 การส่งเอกสาร



รูปที่ 3.24 แสดงแอ็กทिवิตีไดอะแกรมของระบบการส่งเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2.2 การรับเอกสาร

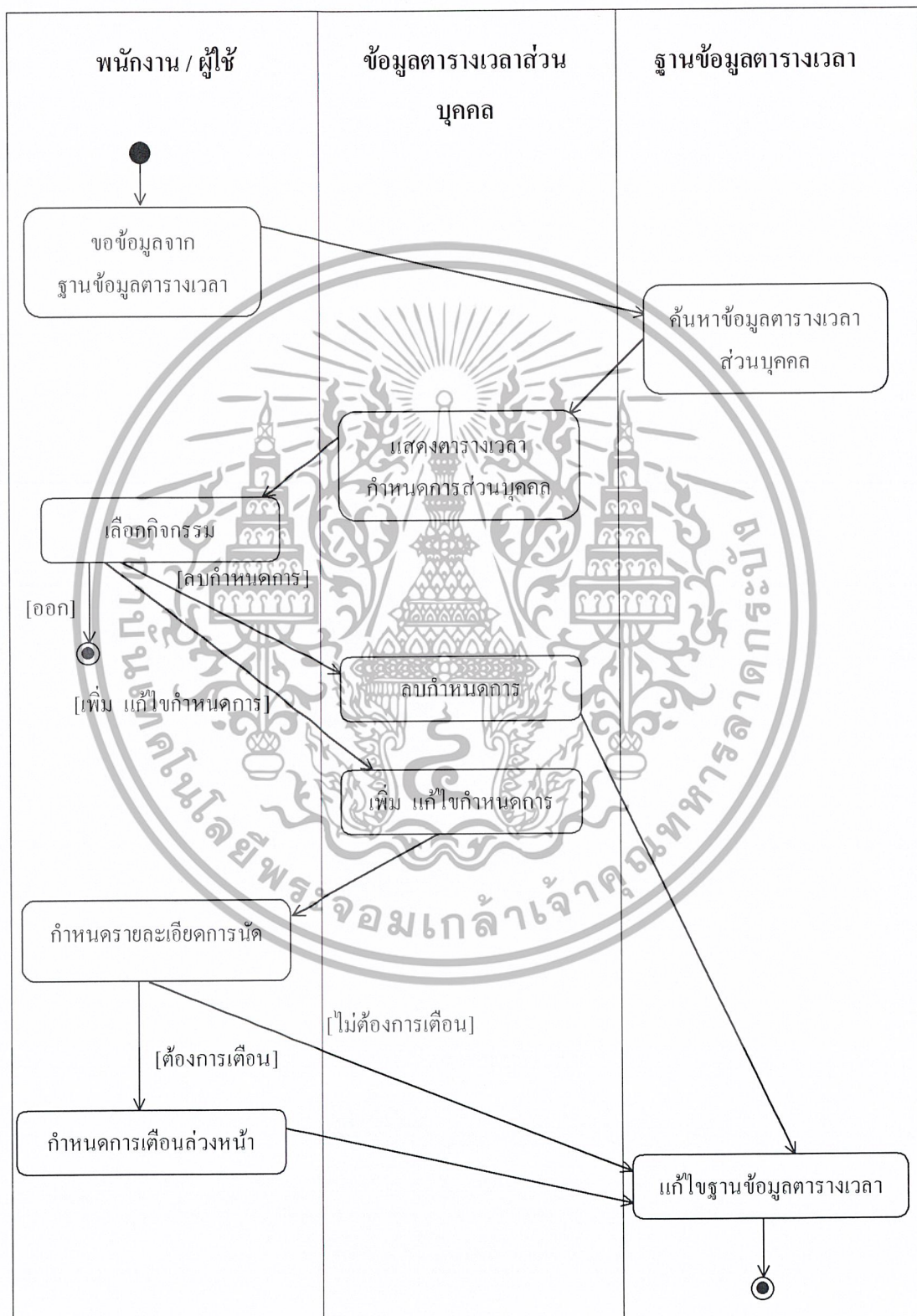


รูปที่ 3.25 แสดงแอ็กทिवิตีไดอะแกรมของระบบการรับเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3 ระบบการจัดการตารางเวลา

3.6.3.1 ระบบการจัดการตารางเวลาส่วนบุคคล



รูปที่ 3.26 แสดงแอ็กทिवิตีไดอะแกรมของระบบการจัดการตารางเวลาส่วนบุคคล

เอกสารนี้เป็นเอกสารทงวนไวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3.2 ระบบการจัดการตารางเวลานัดหมายภายในองค์กร

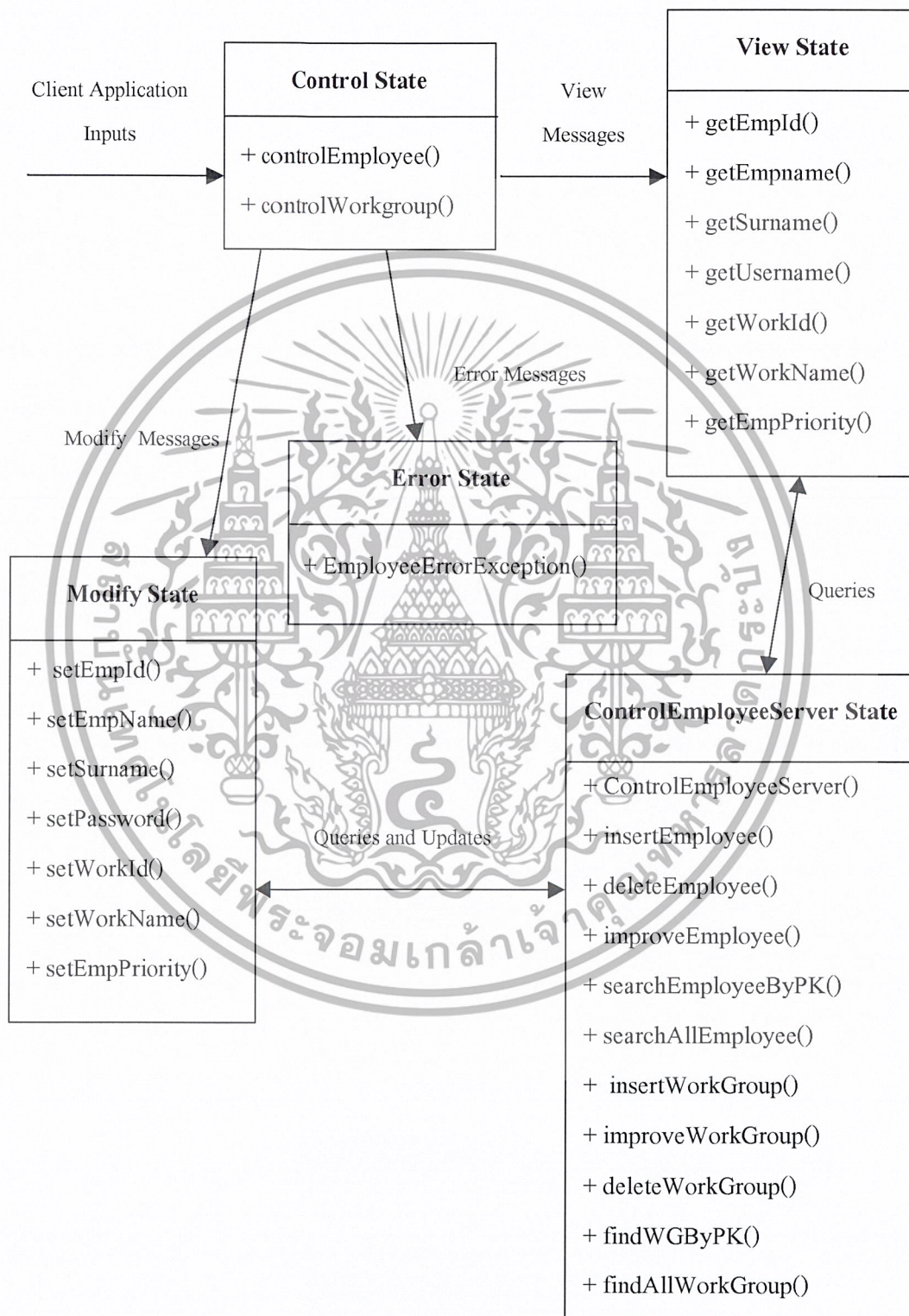


รูปที่ 3.27 แสดงแอ็กทिवิตีไดอะแกรมของระบบจัดการตารางเวลานัดหมายภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 โมเดล-วิว-คอนโทรลเลอร์ (Model-View-Controller)

3.7.1 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์เน็ตการจัการสมาชิกฟ่งไคเอนต์

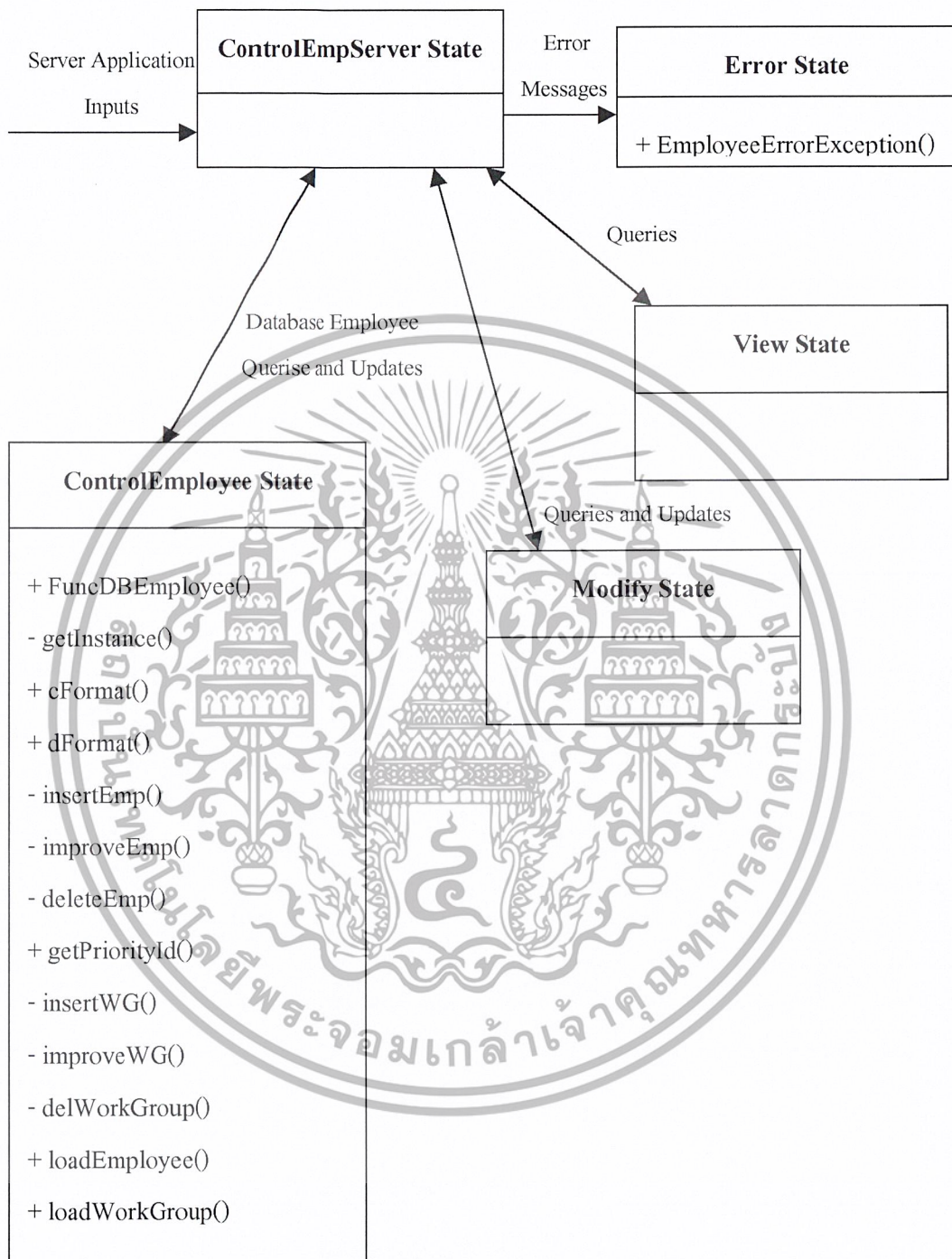


รูปที่ 3.28 แสดงโมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์เน็ตการจัการสมาชิกฟ่งไคเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

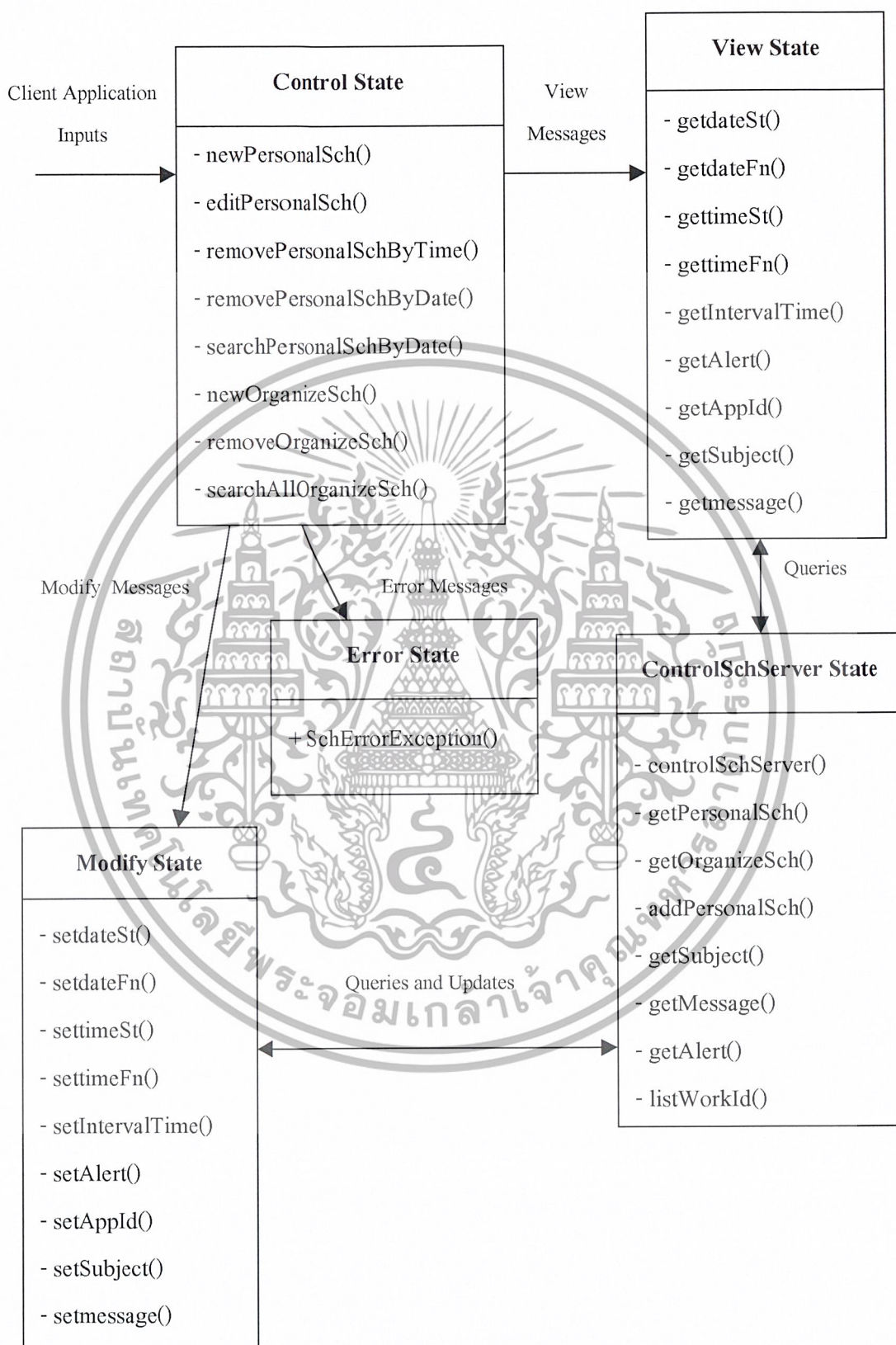
3.7.2 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์การจัดการสมาชิกฝั่งเซิร์ฟเวอร์



รูปที่ 3.29 แสดงโมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์การจัดการสมาชิกฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

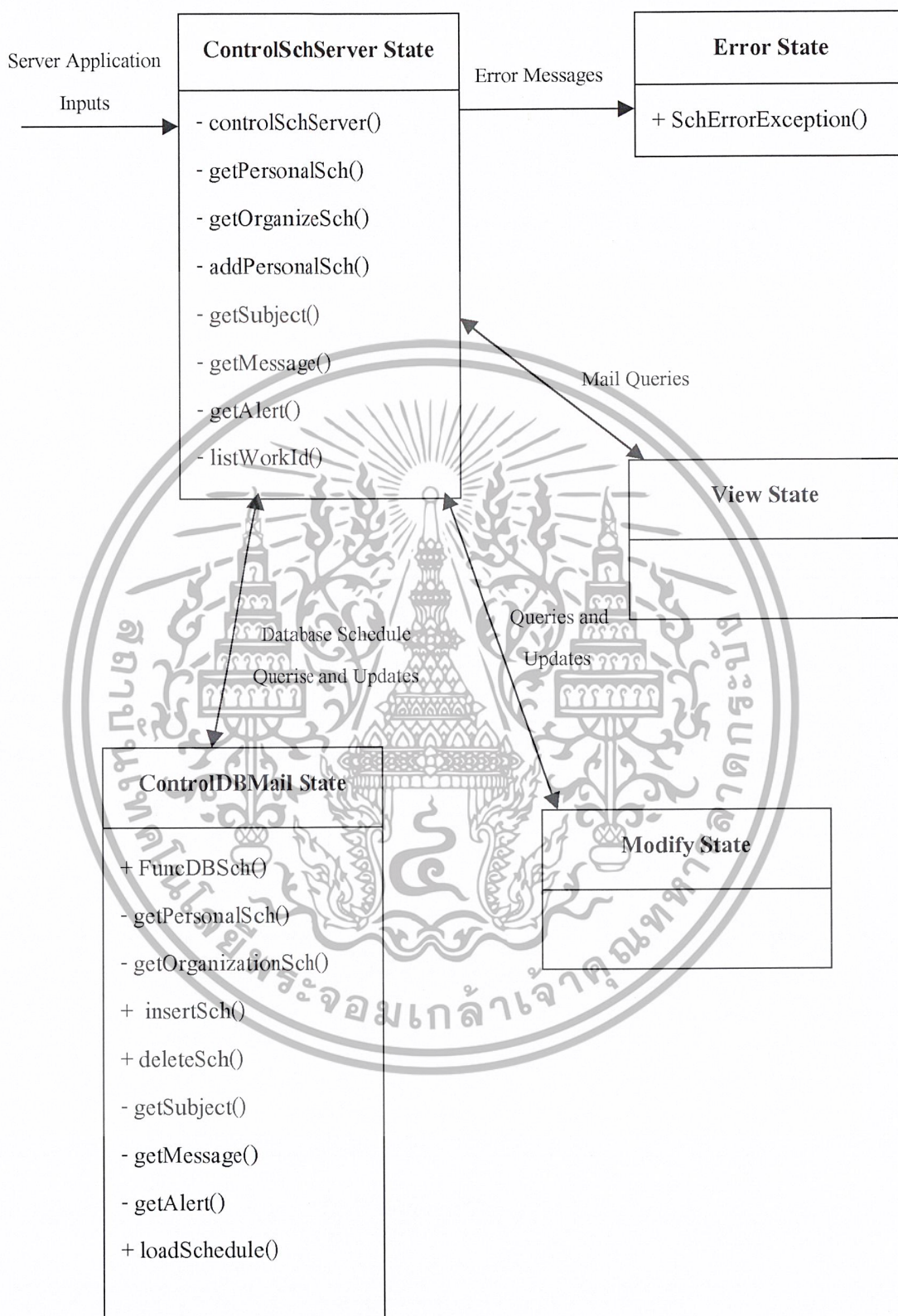
3.7.3 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์จัดการตารางเวลาฝั่งไคลเอนต์



รูปที่ 3.30 แสดงโมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์จัดการตารางเวลาฝั่งไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

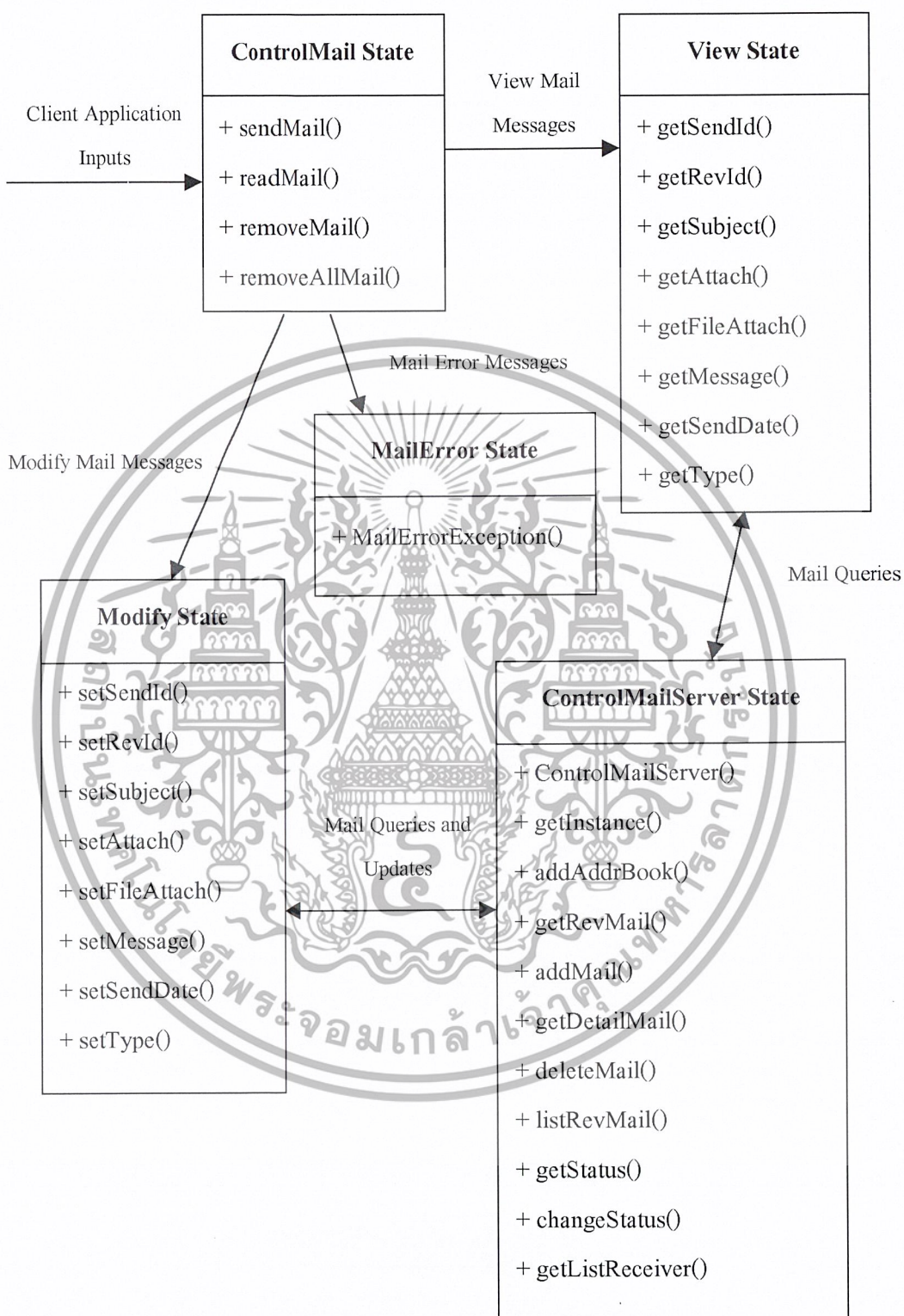
3.7.4 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์จัดการตารางเวลาฝั่งเซิร์ฟเวอร์



รูปที่ 3.31 แสดงโมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์จัดการตารางเวลาฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

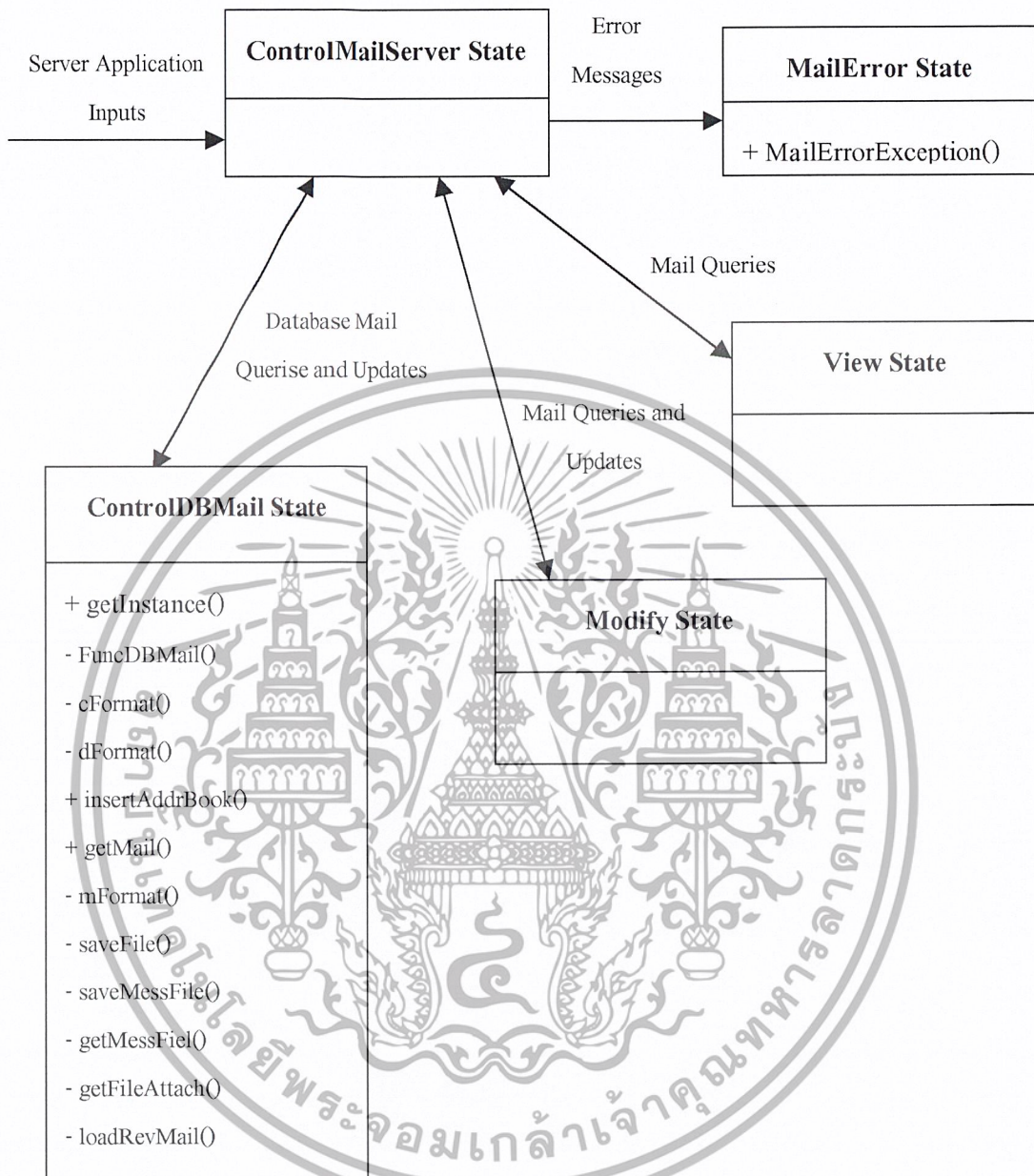
3.7.5 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์กระจายข้อมูลข่าวสารฟ่งไคลเอนต์



รูปที่ 3.32 แสดงโมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์กระจายข้อมูลข่าวสารฟ่งไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.6 โมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์กระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์

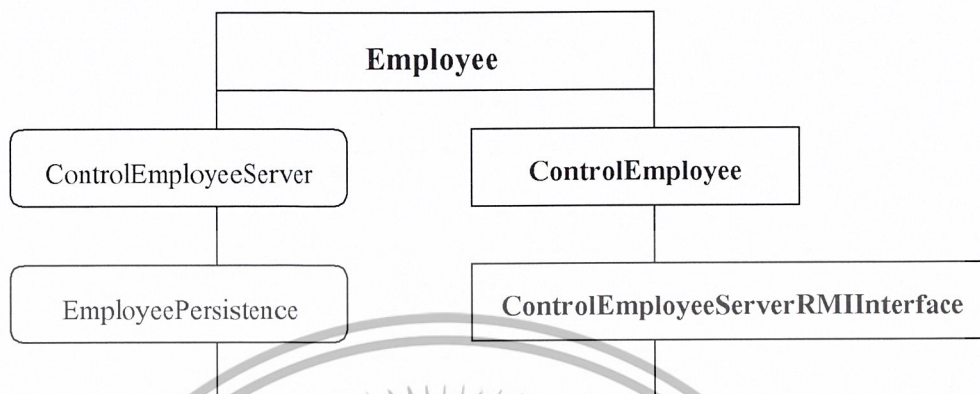


รูปที่ 3.33 แสดงโมเดล-วิว-คอนโทรลเลอร์สำหรับคอมพิวเตอร์กระจายข้อมูลข่าวสารฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 สิ่งทีคอมโพเนนต์ต้องการและจัดเตรียมไว้ให้ (Require and Provide of Component)

3.8.1 สิ่งทีคอมโพเนนต์สำหรับจัดการสมาชิกต้องการและจัดเตรียมไว้ให้



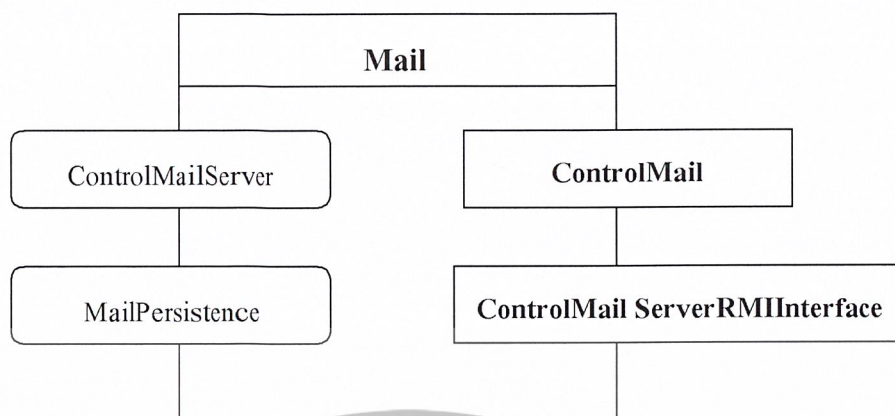
รูปที่ 3.34 แสดงสิ่งทีคอมโพเนนต์สำหรับจัดการสมาชิกต้องการและจัดเตรียมไว้ให้

อินเทอร์เฟซ **ControlEmployeeServer** เป็นส่วนที่จำเป็นต้อง Implement เพื่อให้คอมโพเนนต์นี้สามารถทำงานได้ เป็นอินเทอร์เฟซที่อยู่ฝั่งเซิร์ฟเวอร์ ที่ทำหน้าที่รับการร้องขอความต้องการจากไคลเอนต์ และตอบสนองความต้องการ ไปยังฝั่งไคลเอนต์

อินเทอร์เฟซ **EmployeePersistence** เป็นอินเทอร์เฟซของการทำงานของระบบการจัดการสมาชิกที่อยู่บนฝั่งเซิร์ฟเวอร์

อินเทอร์เฟซ **ControlEmployeeServerRMIIInterface** เป็นอินเทอร์เฟซที่ทำให้ไคลเอนต์ติดต่อกับเซิร์ฟเวอร์ได้ ทำหน้าที่รับการร้องขอความต้องการจากไคลเอนต์ และตอบสนองความต้องการ โดยมีรูปแบบการติดต่อเป็น RMI Protocol

3.8.2 สิ่งทีคอมโพเนนต์สำหรับกระจายข้อมูลข่าวสารต้องการและจัดเตรียมไว้ให้



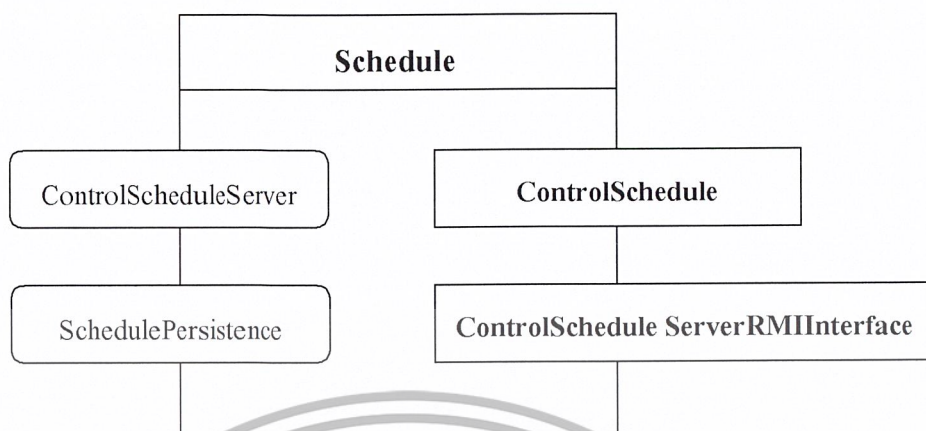
รูปที่ 3.35 แสดงสิ่งทีคอมโพเนนต์สำหรับกระจายข้อมูลข่าวสารต้องการและจัดเตรียมไว้ให้

อินเทอร์เฟส ControlMailServer เป็นส่วนที่จำเป็นต้อง Implement เพื่อให้คอมโพเนนต์นี้สามารถทำงานได้ อินเทอร์เฟสส่วนนี้เป็นส่วนที่รับการร้องขอบริการจากไคลเอนต์ และตอบสนองการทำงาน โดนทำงานในฝั่งเซิร์ฟเวอร์ซึ่งขณะนี้คอมโพเนนต์ที่กำลังพัฒนานี้จะทำงานโดยติดต่อกับฐานข้อมูล ดังนั้นส่วนที่คอมโพเนนต์นี้ต้องการในขณะนี้ คือ ControlEmployee ถ้าในอนาคตต้องการเปลี่ยนแปลงไปใช้ Mail Server ผู้พัฒนาสามารถทำได้โดยเปลี่ยนแปลงส่วนนี้ และสามารถยกเลิกการใช้ ControlEmployee

อินเทอร์เฟส MailPersistence เป็นส่วนที่จำเป็นต้อง Implement เพื่อให้คอมโพเนนต์นี้สามารถทำงานได้ อินเทอร์เฟสส่วนนี้เป็นส่วนที่ทำการประมวลผลในฝั่งเซิร์ฟเวอร์

อินเทอร์เฟส ControlMailServerRMIInterface เป็นอินเทอร์เฟสที่ทำหน้าที่ให้ไคลเอนต์ติดต่อกับเซิร์ฟเวอร์ได้ โดยใช้ติดต่อแบบ RMI Protocol

3.8.3 สิ่งทีคอมโพเนนต์สำหรับจัดการตารางเวลาต้องการและจัดเตรียมไว้ให้



รูปที่ 3.36 สิ่งทีคอมโพเนนต์สำหรับจัดการตารางเวลาต้องการและจัดเตรียมไว้ให้

อินเทอร์เฟซ **ControlScheduleServer** เป็นส่วนที่จำเป็นต้อง Implement เพื่อให้ไคลเอนต์กับเซิร์ฟเวอร์สามารถติดต่อกันได้ อินเทอร์เฟซส่วนนี้เป็นส่วนที่รับการร้องขอจากไคลเอนต์ และตอบสนองการทำงานไปยังเครื่องไคลเอนต์

อินเทอร์เฟซ **SchedulePersistence** เป็นอินเทอร์เฟซของการทำงานของระบบการจัดการตารางเวลา โดยอยู่ที่ฝั่งเซิร์ฟเวอร์

ControlEmployee เป็นส่วนที่จำเป็นต้องมี เนื่องจากคอมโพเนนต์สำหรับจัดการตารางเวลานั้นมีการทำงานแบ่งออกเป็น 2 ส่วนคือ ส่วนตารางเวลาส่วนบุคคล และส่วนตารางเวลาภายในองค์กร ซึ่งทั้งสองส่วนเป็นการทำงานที่จำเป็นต้องติดต่อกับส่วนของพนักงาน

อินเทอร์เฟซ **ControlScheduleServerRMIInterface** อินเทอร์เฟซที่รับการร้องขอจากไคลเอนต์ และตอบสนองไปยังเครื่องไคลเอนต์ โดยอินเทอร์เฟซนี้จะใช้รูปแบบการติดต่อเป็น RMI Protocol

ประเมินและอภิปรายผล

4.1 ลักษณะของระบบการจัดการตารางเวลานัดหมายและกระจายข้อมูลข่าวสาร

การพัฒนาคอมพิวเตอร์สำหรับการจัดการตารางเวลานัดหมายและกระจายข้อมูลข่าวสาร ได้แบ่งลักษณะการทำงานของโปรแกรมออกเป็น 2 ส่วนคือ โปรแกรมส่วนไคลเอนต์ และ โปรแกรมส่วนเซิร์ฟเวอร์ โดยที่

4.1.1 โปรแกรมส่วนไคลเอนต์ เป็นส่วนโปรแกรมที่เกี่ยวข้องกับผู้ใช้งาน เพื่อช่วยสนับสนุนการทำงานเป็นกลุ่ม ประกอบด้วยส่วนการทำงานเกี่ยวกับการจัดการสมาชิก การจัดการตารางเวลา และการจัดการเอกสาร โดยมีการพัฒนาโปรแกรมให้มีรูปแบบเป็นคอมพิวเตอร์ ด้วยภาษา Java โดยมีลักษณะดังนี้

- ระบบมีการร้องขอให้ผู้ใช้ทำการระบุ username และ password เพื่อทำการ login ก่อนใช้งานโปรแกรมทุกครั้ง
- ระบบจะมีการแจ้งแก่ผู้ใช้ให้ทราบว่าได้มีการนัดหมาย และมีข่าวสารที่ส่งมาให้ผู้ใช้อย่างไรหรือไม่
- รูปแบบอินเทอร์เฟซ คือ จะให้ผู้ใช้ทำการเลือกฟังก์ชันการทำงานโดยเลือกจากเมนูที่กำหนดให้ โดยจะมีการตรวจสอบสิทธิ์ในการใช้งานฟังก์ชันจาก username ที่ทำการ login
- ระบบสามารถทำการแสดงเวลานัดหมายส่วนบุคคลได้ โดยผู้ใช้สามารถทำการเพิ่มแก้ไข และยกเลิกการนัดหมายได้ และระบบยังสามารถแจ้งเตือนถึงกำหนดการนัดหมายนั้น ๆ ได้โดยจะมีการแสดงข้อความเตือนในรูปแบบของ Pop Up
- ระบบสามารถแสดงเวลาการนัดหมายกลุ่มของพนักงานได้ และผู้ใช้สามารถทำการจัดการนัดหมาย และยกเลิกการนัดหมายกลุ่มพนักงานได้ โดยผู้ที่ทำการนัดหมายจะต้องเป็นบุคคลที่มีสิทธิ์ที่จะทำการนัดหมายได้เท่านั้น
- ระบบสามารถกระจายข้อมูลข่าวสารได้ โดยจะมีลักษณะการทำงานเหมือนจดหมายอิเล็กทรอนิกส์ (e-mail)

4.1.2 โปรแกรมส่วนเซิร์ฟเวอร์ เป็นส่วนโปรแกรมที่เกี่ยวข้องกับ การตรวจสอบผู้ใช้งานระบบ (Authentication) และการจัดการและการให้บริการทรัพยากร เช่น ตารางการจัดเวลานัดหมายภายในองค์กร ฐานข้อมูลเอกสารนอกจากนี้ในส่วนของเซิร์ฟเวอร์ยังทำการจัดเก็บฐานข้อมูลไว้ด้วย โดยได้พัฒนาโปรแกรมส่วนเซิร์ฟเวอร์ ด้วยภาษา java โดยมีลักษณะ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบบมีการตรวจสอบผู้ใช้งาน (Authentication) โดยการตรวจสอบจาก username และ password ที่ผู้ใช้ทำการระบุไว้ขณะที่ทำการ login
- ระบบมีการตรวจสอบกับตารางเวลานัดหมายและตารางเอกสารของผู้ใช้ และทำการแจ้งสถานะให้ผู้ใช้ทราบ
- ระบบให้บริการการใช้งานฐานข้อมูลต่าง ๆ เช่น ฐานข้อมูลสมาชิก ฐานข้อมูลเวลานัดหมาย ฐานข้อมูลเอกสาร เป็นต้น เพื่อให้บริการแก่ผู้ใช้งานในฝั่งไคลเอนต์
- ระบบมีหน้าจอแสดงสถานะของเซิร์ฟเวอร์

4.2 ข้อกำหนดของการประเมิน

ติดตั้งโปรแกรมแต่ละส่วนไว้ในเครื่องคอมพิวเตอร์แต่ละเครื่อง โดยมีการเชื่อมต่อโดยผ่านระบบ LAN บนเครือข่ายอินเทอร์เน็ต โดยมีการจัดการฐานข้อมูลโดยอาศัย ODBC

4.3 การประเมิน

การทดสอบการใช้งานของระบบ ทำการทดสอบแบบเต็มระบบครั้งสุดท้าย เมื่อวันที่ 6 มีนาคม พ.ศ. 2546 โดยมีขั้นตอนการประเมินงาน ดังนี้

4.3.1 เริ่มการใช้งานโปรแกรมส่วนเซิร์ฟเวอร์ต้องทำการติดต่อกับฐานข้อมูลก่อน โดยมีการกำหนดค่า ODBC ก่อน

4.3.2 ทำการเข้าสู่ระบบจากเครื่องฝั่งไคลเอนต์ จากนั้นเครื่องฝั่งเซิร์ฟเวอร์จะทำการตรวจสอบ username และ password และให้บริการเครื่องฝั่งเซิร์ฟเวอร์ตามสิทธิของผู้ใช้

4.3.3 โปรแกรมทำการแจ้งจำนวนเอกสารที่ยังไม่ได้ทำการอ่าน และกำหนดการนัดหมายเมื่อผู้ใช้ได้เข้าสู่ระบบแล้ว

4.3.4 ส่วนจัดการตารางเวลา จะประกอบไปด้วยการทำงาน 2 ส่วนคือ การจัดการเวลานัดหมายส่วนบุคคล และการจัดการตารางนัดหมายภายในองค์กร ซึ่งมีรายละเอียดของการทำงานในแต่ละส่วน ดังนี้

1. ตารางเวลานัดหมายส่วนบุคคล

- โปรแกรมจะแสดงเวลานัดหมายของผู้ใช้ในวันที่ทำการเข้าสู่ระบบโดยแสดงเวลาตามเครื่องไคลเอนต์ โดยที่ผู้ใช้สามารถขอดูเวลานัดหมายย้อนหลัง-ล่วงหน้าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการเพิ่ม ขกเลิก หรือแก้ไข กำหนดการนัดหมายของตนเอง โดยที่โปรแกรมจะบันทึกกำหนดการไว้ที่ฐานข้อมูลที่เก็บไว้ในฝั่งเซิร์ฟเวอร์

2. ตารางเวลานัดหมายภายในองค์กร

- เมื่อต้องการทำการนัดหมายกลุ่มคนภายในองค์กรนั้น โปรแกรมจะทำการติดต่อฐานข้อมูลของพนักงานในกลุ่มนั้น ๆ เพื่อทำการหาวันและเวลาที่สามารถทำการนัดหมายได้
- เมื่อทำการเลือกวันและเวลาที่ทำการนัดหมายได้แล้ว โปรแกรมจะทำการแจ้งให้พนักงานแต่ละคนในกลุ่มนั้น ๆ ทราบว่าได้มีการนัดหมายเกิดขึ้น เมื่อพนักงานคนนั้นได้ทำการ login เพื่อเข้าสู่ระบบ

4.3.5 การทำงานส่วนการจัดการเอกสารนั้น ผู้ใช้สามารถทำการส่งเอกสารให้กับพนักงานแต่ละบุคคล หรือจะส่งเอกสารให้กับกลุ่มพนักงานก็ได้ และเมื่อทำการส่งเอกสาร ไปให้ผู้รับแล้ว โปรแกรมจะแจ้งให้ผู้รับเอกสารทราบถึงจำนวนเอกสาร ที่ผู้รับยังไม่ได้อ่านเมื่อผู้รับเอกสารได้ทำการเข้าสู่ระบบแล้ว

4.3.6 ในส่วนการทำงานเกี่ยวกับการจัดการสมาชิกนั้น จะกำหนดไว้ให้เป็นหน้าที่ของ ผู้ดูแลระบบเท่านั้น ซึ่งจะสามารถทำการจัดกลุ่มให้กับพนักงานตามงานที่พนักงานได้รับมอบหมายได้ด้วย

4.4 การอภิปรายผล

หลังจากทำการทดสอบการใช้งานของระบบ พบว่า ระบบสามารถเชื่อมโยงการทำงานระหว่าง 2 ส่วน ได้อย่างเรียบร้อย โดยที่เมื่อต้องการติดต่อกับเครื่องเซิร์ฟเวอร์ก็สามารถทำงานได้อย่างรวดเร็ว เช่นการขอใช้ข้อมูลจากฐานข้อมูลที่เก็บไว้ในเครื่องฝั่งเซิร์ฟเวอร์ก็สามารถทำงานตามที่ร้องขอได้เป็นอย่างดี แต่โปรแกรมนี้อย่างมีข้อจำกัดในการใช้งานอยู่ตรงที่การกระจายข้อมูลข่าวสารจะต้องกระทำภายในเครือข่ายขององค์กรนั้น ๆ เท่านั้น ผู้ใช้ไม่สามารถอ่านเอกสารได้จากเครือข่ายภายนอกองค์กร เหมือนกับจดหมายอิเล็กทรอนิกส์

การทดสอบในครั้งนี้จะเป็นแนวทางในการนำระบบการจัดการตารางเวลานัดหมายและกระจายข้อมูลข่าวสารไปใช้สนับสนุนการทำงานเป็นกลุ่ม เพื่อให้การประสานงานระหว่างบุคคลภายในองค์กรสมบูรณ์มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

การพัฒนาคอมโพเนนต์สำหรับการจัดการตารางเวลานัดหมายและกระจายข้อมูลข่าวสารภายในองค์กร ได้ดำเนินการตามวัตถุประสงค์และประโยชน์ที่วางไว้ คือ

5.1.1 คอมโพเนนต์ส่วนการจัดการสมาชิก โดยสามารถทำการจัดกลุ่มสมาชิกตามกลุ่มงานได้ และสามารถทำการแก้ไขและยกเลิกสมาชิก หรือกลุ่มงานนั้น ๆ ได้

5.1.2 คอมโพเนนต์ส่วนการจัดการตารางเวลานัดหมาย โดยแบ่งการทำงานเป็น 2 ส่วน คือ ส่วนจัดการตารางเวลานัดหมายส่วนบุคคล และส่วนจัดการตารางเวลานัดหมายภายในองค์กร โดย

- ส่วนการจัดการตารางเวลานัดหมายส่วนบุคคล ส่วนนี้สามารถจัดการเพิ่ม แก้ไข และยกเลิกกำหนดการส่วนบุคคลได้ และเมื่อมีการกำหนดเวลานัดหมายแล้ว โปรแกรมสามารถทำการแจ้งเตือนให้ทราบถึงกำหนดการที่ต้องการได้
- ส่วนการจัดการตารางนัดหมายภายในองค์กรนั้น สามารถจัดการนัดหมายกลุ่มสมาชิกภายในองค์กรได้ โดยโปรแกรมจะทำการหาวันและเวลาที่เหมาะสมที่สามารถทำการนัดหมายสมาชิกภายในกลุ่มได้ และเมื่อทำการเลือกวันและเวลาเพื่อทำการนัดหมายแล้ว โปรแกรมจะทำการแจ้งให้สมาชิกในกลุ่มงานนั้น ๆ รับทราบถึงกำหนดการนัดหมายได้ เมื่อต้องการยกเลิก หรือแก้ไขกำหนดการนั้นแล้ว ก็สามารถทำได้เช่นเดียวกัน

5.1.3 คอมโพเนนต์ส่วนการกระจายข้อมูลข่าวสาร ในส่วนนี้สนับสนุนการกระจายข้อมูลข่าวสารให้กับสมาชิกในองค์กร โดยสามารถทำการแจ้งให้สมาชิกเฉพาะบุคคล หรือกลุ่มสมาชิกได้ และทำการแจ้งให้ผู้รับข้อความทราบสถานะ และจำนวนของเอกสารที่ได้รับได้ ในขณะที่อ่านเอกสารแล้วต้องการส่งเอกสารให้กับพนักงานคนอื่น ๆ ก็สามารถส่งต่อได้เช่นกัน นอกจากนี้ยังสามารถให้ผู้รับเอกสารที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

การพัฒนาคอมพิวเตอร์การจัดการตารางเวลานัดหมายและการกระจายข้อมูลข่าวสารภายในองค์กรนั้น เป็นคอมพิวเตอร์ส่วนหนึ่งเพื่อสนับสนุนการทำงานเป็นกลุ่ม ดังนั้นจึงมีข้อเสนอแนะซึ่งจะเป็นแนวทางหนึ่งที่ใช้ในการพัฒนาต่อไป ดังนี้

5.2.1 การพัฒนาโปรแกรมนี้จัดทำเป็นคอมพิวเตอร์ ดังนั้นการพัฒนาโปรแกรมต่อไปควรมีการศึกษาการทำงานของโปรแกรมต้นแบบให้ละเอียด และศึกษารูปแบบการติดต่อสื่อสารกันระหว่างแต่ละคอมพิวเตอร์

5.2.2 เมื่อต้องการทำการปรับปรุง หรือพัฒนาคอมพิวเตอร์ในส่วนการจัดการตารางเวลานัดหมายและการกระจายข้อมูลข่าวสารนั้น ควรศึกษาฟังก์ชันการทำงานในแต่ละคอมพิวเตอร์ให้ละเอียด และเนื่องจากคอมพิวเตอร์นี้มีการติดต่อกับคอมพิวเตอร์ส่วนการจัดการสมาชิก ดังนั้นต้องศึกษาถึงความสัมพันธ์และการทำงานที่เชื่อมโยงกันระหว่างคอมพิวเตอร์ด้วย

5.2.3 ส่วนการจัดการข้อมูลสมาชิกจะต้องมีการเก็บฐานข้อมูลของผู้ใช้ ซึ่งประกอบไปด้วย username และ password ดังนั้นควรมีการทำงานที่ใช้ในการเข้ารหัสข้อมูลส่วนนี้เก็บไว้เพื่อเป็นการเพิ่มการรักษาความปลอดภัยให้กับข้อมูลในส่วนนี้ด้วย

5.2.4 ส่วนการกระจายข้อมูลข่าวสาร ควรจะมีการพัฒนารูปแบบให้สามารถทำงานร่วมกับโปรแกรมการจัดการจดหมายอิเล็กทรอนิกส์ ที่ทำงานบนเครือข่ายอินเทอร์เน็ต โดยใช้โปรโตคอล TCP/IP ซึ่งต้องมีการเพิ่มส่วนที่เป็นแม่ชีฟเวอร์ขึ้น

5.2.5 เมื่อทำการพัฒนาคอมพิวเตอร์อื่นๆ ที่สนับสนุนการทำงานเป็นกลุ่มเพื่อให้เป็นโปรแกรมที่สมบูรณ์ยิ่งขึ้น ผู้พัฒนาควรศึกษาถึงวิธีการเชื่อมโยงคอมพิวเตอร์ที่ได้พัฒนาขึ้นมาให้สามารถทำงานร่วมกับคอมพิวเตอร์ส่วนอื่นๆ ได้

บรรณานุกรม

รุ่งโรจน์ โพนคำ และ ปราณี มณีรัตน์. 2537. **Java Programming** ฉบับสร้างแอปพลิเคชันด้วยจาวา

สวিংซ์. กรุงเทพมหานคร. บริษัท ชัคเชส มีเดีย จำกัด.

วีระศักดิ์ ชิงฉาวร. 2543. **Java Programming Volume I**. กรุงเทพมหานคร. ซีเอ็ดดูเคชั่น.

วีระศักดิ์ ชิงฉาวร. 2545. **Java Programming Volume II**. กรุงเทพมหานคร: ซีเอ็ดดูเคชั่น.

David Marc and Geoffrey Bock. 2535. Groupware : Software for Computer-Supported Cooperative Work. Los Almitos : The Institute of Electrical and Electronics Engineers Computer Society Press.

Justin Couch, 2542. **Java 2 Networking**. New York : McGraw-Hill companies, Inc.

Robert Orfali and Dan Harkey. 2537. **Client/Server Survival Guide with OS/2**. New York : Van Nostrand Reinhold.





ภาคผนวก ก

คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

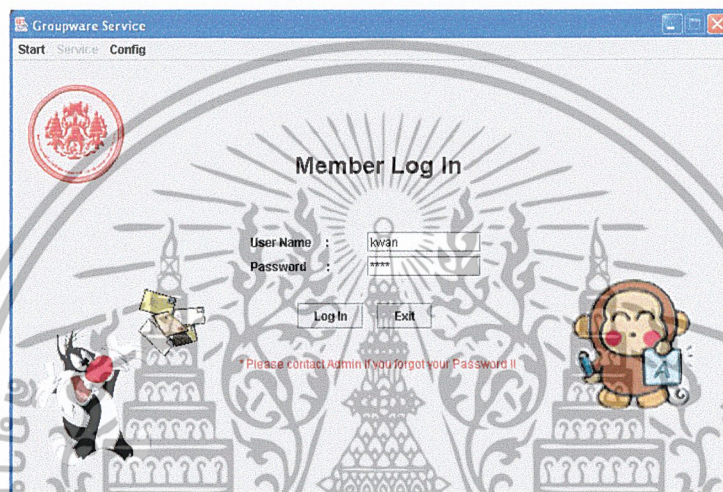
ภาคผนวก ก

คู่มือการใช้งาน

1. ขั้นตอนการใช้งานโปรแกรม

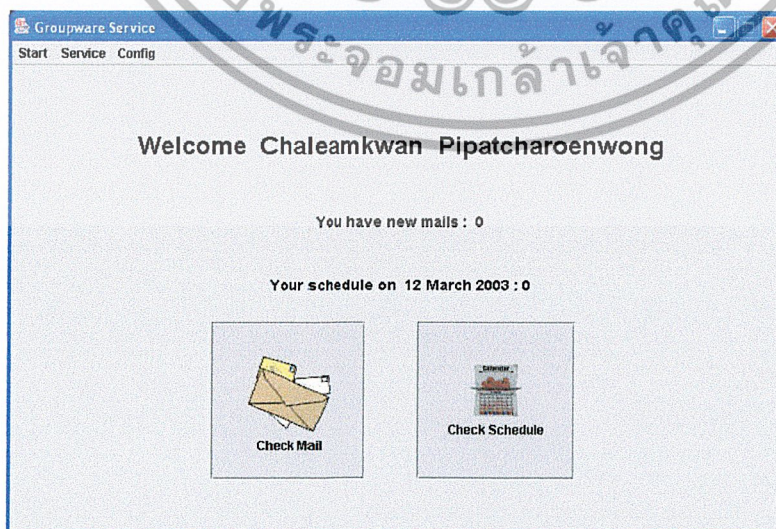
การใช้งานโปรแกรมมีขั้นตอน ดังนี้

เมื่อเปิดโปรแกรมแล้วโปรแกรมจะแสดงหน้าจอ เพื่อให้ผู้ใช้ทำการใส่ username และ password เพื่อทำการ login เข้าสู่ระบบ โดยโปรแกรมจะแสดงหน้าจอ ดังรูป



รูปที่ ก.1 แสดงหน้าจอ Login

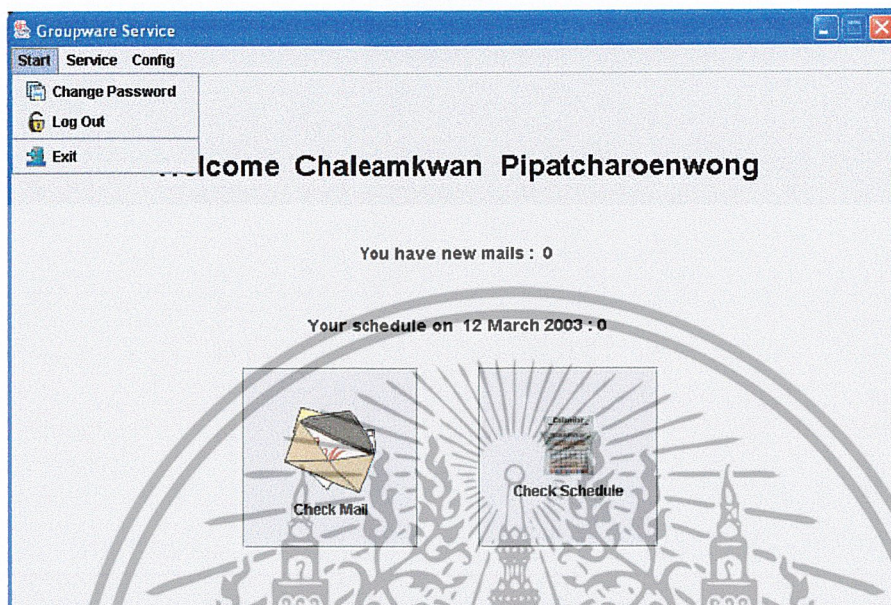
เมื่อทำการ login แล้ว โปรแกรมจะแสดงหน้าจอเพื่อบอกให้ผู้ใช้ทราบว่า มีจำนวนเอกสารใหม่ที่ยังไม่ได้อ่าน และจำนวนกำหนดการนัดหมายประจำวัน ดังรูปที่ ก.2



รูปที่ ก.2 แสดงหน้าจอแจ้งจำนวนเอกสารและจำนวนกำหนดการนัดหมายประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการ login เข้าสู่ระบบได้แล้วผู้ใช้งานสามารถทำการเลือกฟังก์ชันการทำงานจากเมนูที่แสดง โดยเมนูส่วน Start จะประกอบไปด้วยฟังก์ชันการทำงาน ดังนี้คือ การเปลี่ยน password การ logout และการออกจากโปรแกรม โดยแสดงดังรูปที่ ก.3



รูปที่ ก.3 แสดงหน้าจอเมนูในส่วน Start

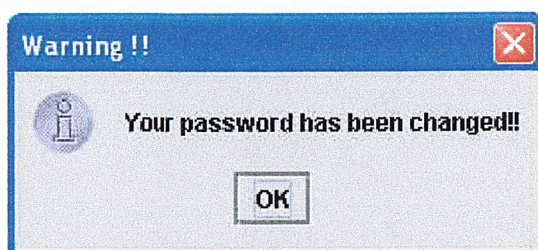
เมื่อต้องการเปลี่ยน password จะปรากฏหน้าจอดังนี้



รูปที่ ก.4 แสดงหน้าจอเปลี่ยน password

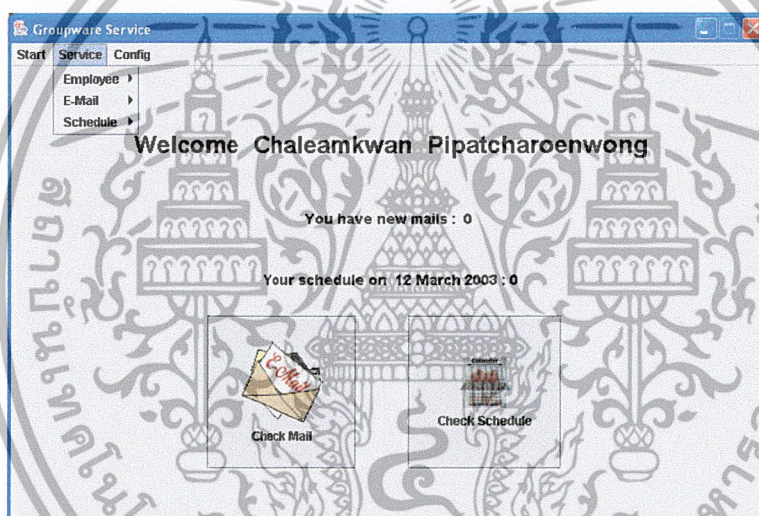
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมจะแจ้งให้ทราบว่าระบบได้ทำการเปลี่ยน password ให้แล้ว ดังรูป



รูปที่ ก.5 แสดงหน้าจอแจ้งให้ทราบว่าได้ทำการเปลี่ยน password เรียบร้อยแล้ว

เมนูส่วน Service จะเป็นส่วนที่เกี่ยวกับการจัดการสมาชิก การกระจายข้อมูลข่าวสาร และส่วนที่จัดการตารางเวลาดังหมาย ดังนี้



รูปที่ ก.6 แสดงหน้าจอเมนูในส่วน Service

เมื่อต้องการสร้างกลุ่มในการทำงานใหม่จะแสดงหน้าจอ ดังรูป โดยโปรแกรมจะทำการแสดงรหัสของกลุ่มงาน โดยรหัสนี้ไม่สามารถแก้ไขได้ แต่ในส่วนชื่อของกลุ่มนั้นเราสามารถกำหนดเองได้ นอกจากนั้นยังแสดงรายชื่อพนักงาน และให้ผู้ใช้ทำการเลือกรายชื่อพนักงานนั้นๆ ได้ โดยคลิกที่รายชื่อพนักงาน และกดปุ่ม << Add

Groupware Service

Start Service Config

Create Work Group

Work Group Id : G20030312001

Work Group Name :

Employee Id	Name - Surname	Position	Priority
E001	Chaleamkwan Pipa	Member	Edit Appointment & ...

Employee

Keerati Anurak...
Kid Zodiac
Janthana Sotr...
Chometavee S...
Eakkamon Loa...

<< Add
Remove >>
Save ... Clear

รูปที่ ก.7 แสดงหน้าจอสร้างกลุ่มการทำงานใหม่

เมื่อทำการเลือกพนักงานได้แล้ว จะมีรายชื่อพนักงานเพิ่มเข้ามาในส่วนของ Group member นอกจากนั้นในกรณีที่เรทำการเลือกพนักงานผิด เราสามารถเลือกรายชื่อพนักงานคนนั้นจาก Group member แล้วกดปุ่ม Remove >> โปรแกรมจะทำการลบชื่อพนักงานคนนั้นออกจาก group member ให้

Groupware Service

Start Service Config

Create Work Group

Work Group Id : G20030312001

Work Group Name : WorkGroup

Employee Id	Name - Surname	Position	Priority
E001	Chaleamkwan Pipa	Member	Edit Appointment & ...
E003	Janthana Sotrakul	Member	Read Only
E002	Kid Zodiac	Member	Read Only

Employee

Keerati Anurak...
Chometavee S...
Eakkamon Loa...

<< Add
Remove >>
Save ... Clear

รูปที่ ก.8 แสดงหน้าจอเมื่อทำการเพิ่มพนักงานในกลุ่มแล้ว

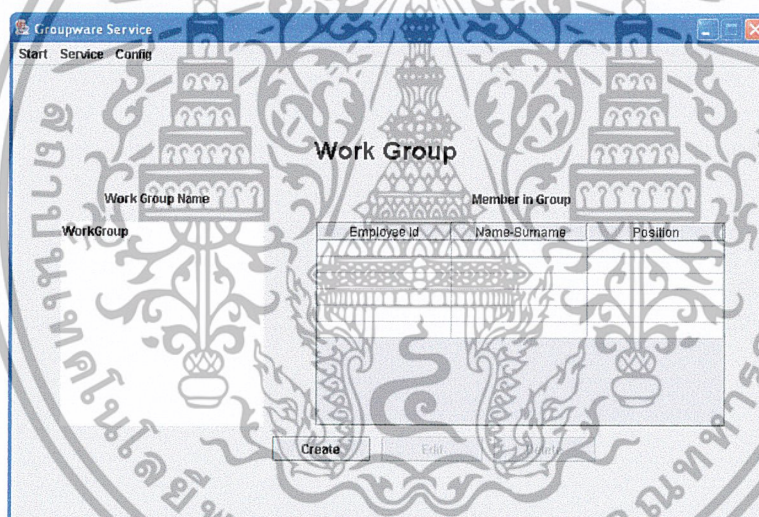
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดสมาชิกในกลุ่มงานเรียบร้อยแล้ว ให้คลิกปุ่ม Save.. จากนั้น โปรแกรมจะแจ้งให้ทราบว่าได้ทำการสร้างกลุ่มงานให้เรียบร้อยแล้ว ดังรูป



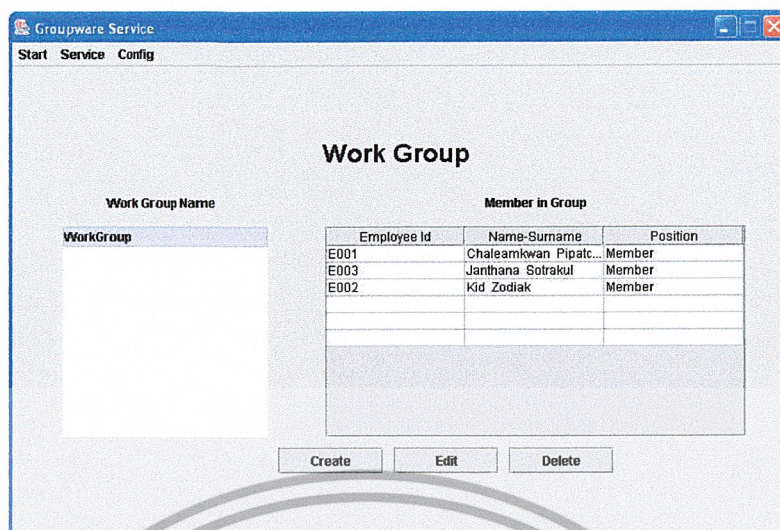
รูปที่ ก.9 แสดงหน้าจอแจ้งให้ทราบว่าได้ทำการสร้างกลุ่มงานเรียบร้อยแล้ว

จากนั้นเมื่อต้องการดูกลุ่มที่มีผู้ใช้คนนั้นอยู่ในกลุ่มนั้นด้วย หน้าจอจะแสดงรายชื่อของกลุ่มที่เรากำหนดไว้ ดังรูป



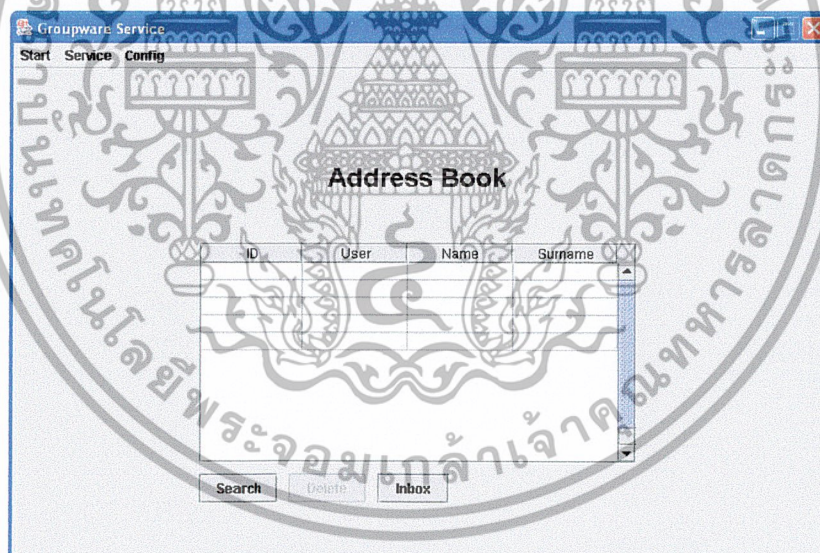
รูปที่ ก.10 แสดงหน้าจอแสดงกลุ่มงานที่มีผู้ร่วมทำงานด้วย

จากนั้นเมื่อเราต้องการทราบว่าในกลุ่มงานนั้นมีพนักงานคนใดบ้างอยู่ในกลุ่มก็สามารถคลิกที่ชื่อของกลุ่มงานนั้น ๆ จากนั้น โปรแกรมจะแสดงรายชื่อของพนักงานภายในกลุ่มให้ทราบ โดยจะบอกถึงตำแหน่งประจำกลุ่มงานของพนักงานแต่ละคนด้วย



รูปที่ ก.11 แสดงหน้าจอแสดงรายชื่อพนักงานในกลุ่มงาน

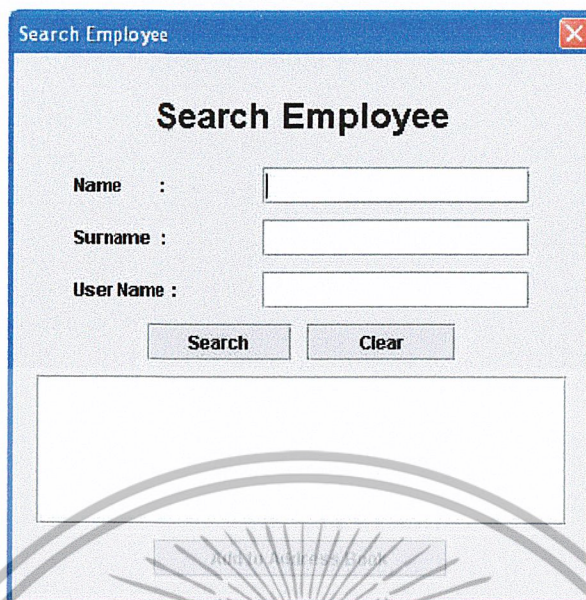
ในส่วน Address Book นั้นจะเป็นการเก็บรายชื่อพนักงานในกลุ่มไว้



รูปที่ ก.12 แสดงหน้าจอหน้า Address Book

เมื่อเราคลิกปุ่ม Search หน้าจอจะแสดง ดังรูป 4.13 โดยที่เราสามารถให้โปรแกรมทำการค้นหาจากชื่อ หรือนามสกุล หรือ username ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Search Employee

Search Employee

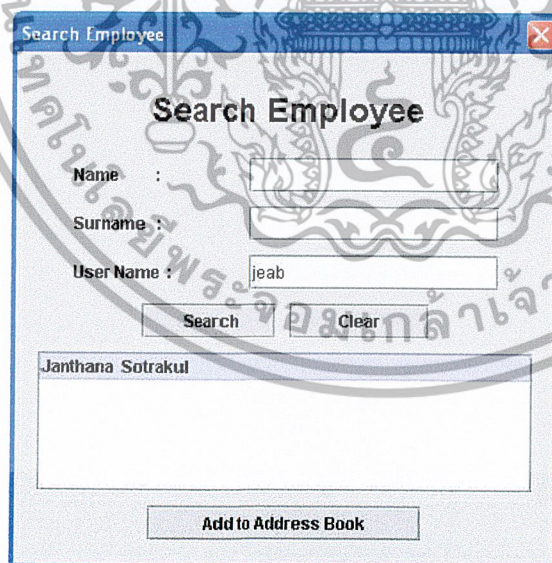
Name :

Surname :

User Name :

รูปที่ ก.13 แสดงหน้าจอ Search Employee

ตัวอย่างเช่น เราต้องการค้นหาจาก Username สามารถทำได้โดยการใส่ username ของพนักงานที่เราทราบ แล้วกดปุ่ม Search จากนั้น โปรแกรมจะแสดงชื่อพนักงานที่ใช้ username ที่เรากำหนด ดังรูป



Search Employee

Search Employee

Name :

Surname :

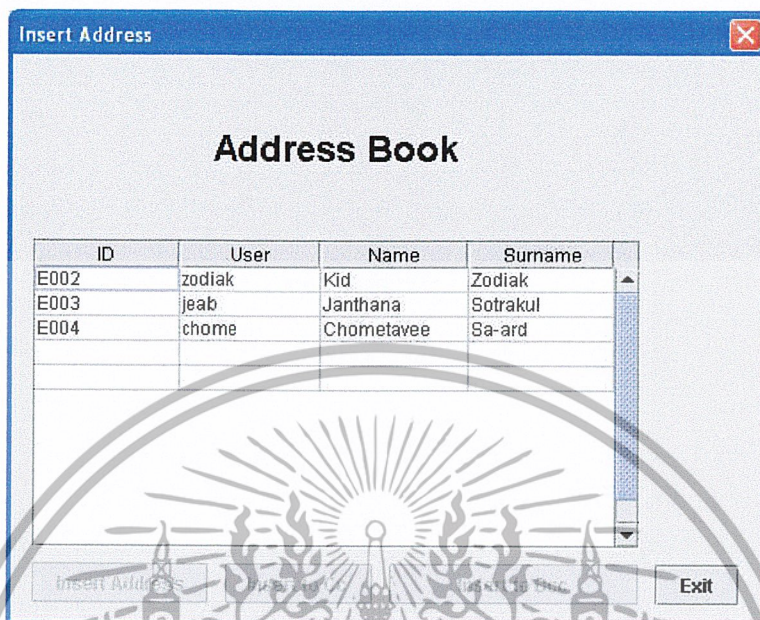
User Name :

Janthana Sotrakul

รูปที่ ก.14 แสดงหน้าจอ Search Employee เมื่อทำการค้นหาจาก Username

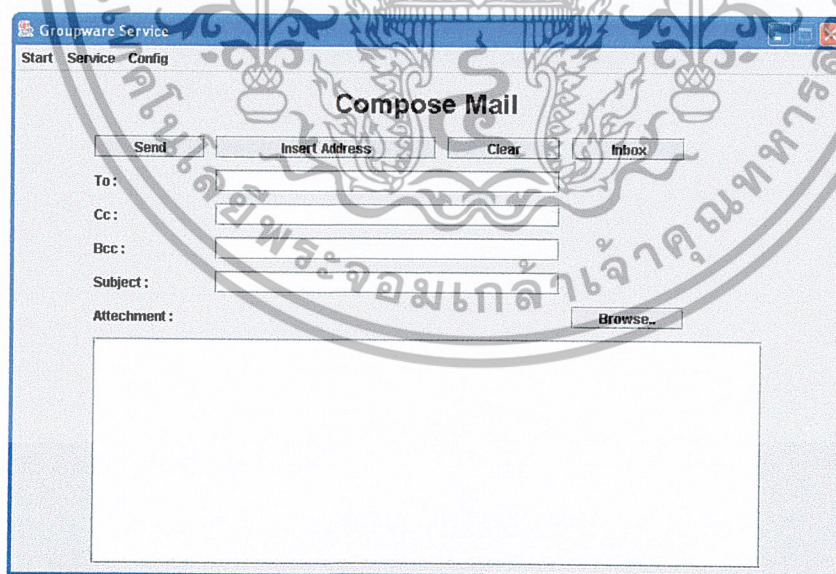
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเมื่อเราทำการเพิ่มพนักงานใน Address book แล้วรายชื่อพนักงานจะปรากฏอยู่ที่หน้า Address book ดังนี้



รูปที่ ก.15 แสดงหน้า Address Book เมื่อทำการเพิ่มรายชื่อพนักงานแล้ว

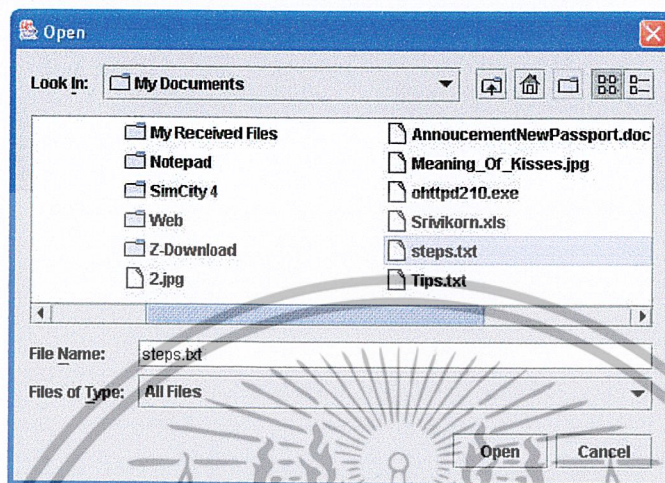
เมื่อต้องการส่งข้อความจะปรากฏหน้าจอ ดังนี้



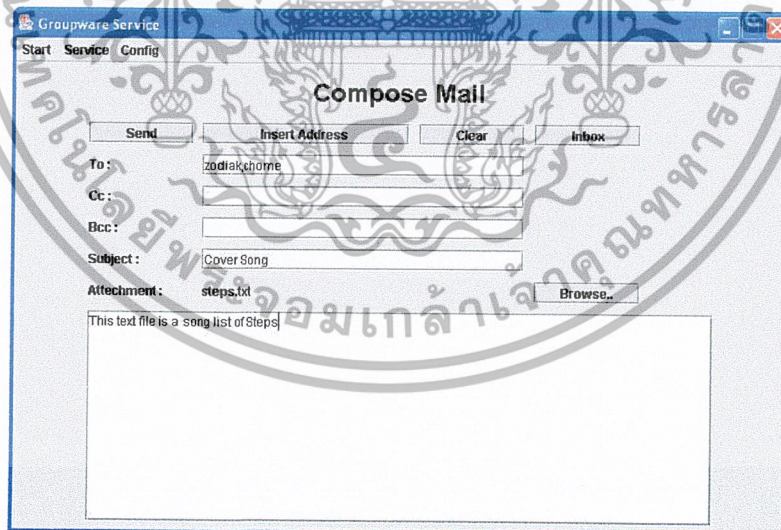
รูปที่ ก.16 แสดงหน้าจอ Compose Mail

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถเลือกรายชื่อพนักงานที่เราต้องการจะส่งข้อความไปให้จากหน้า Address book ได้ และเราสามารถแนบไฟล์ที่ต้องการส่งไปได้อีกด้วย โดยกดที่ปุ่ม Browse และเลือกไฟล์ที่ต้องการแนบไปด้วย ดังรูป



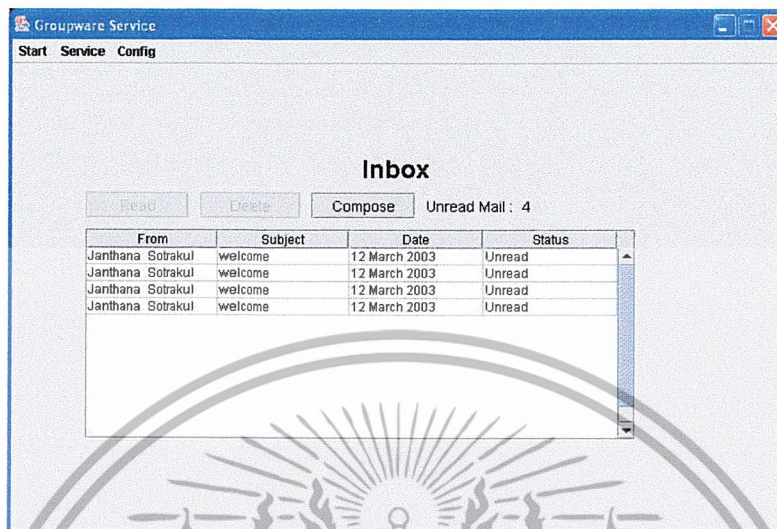
รูปที่ ก.17 แสดงหน้าจอเพื่อทำการเลือกไฟล์ที่ต้องการแนบไปพร้อมกับข้อความที่จะส่ง
เมื่อทำการเลือกไฟล์ที่จะส่งได้แล้วก็ส่งข้อความได้ โดยกดปุ่ม Send



รูปที่ ก.18 แสดงหน้าจอเมื่อแนบไฟล์และพิมพ์ข้อความที่ต้องการส่งแล้ว

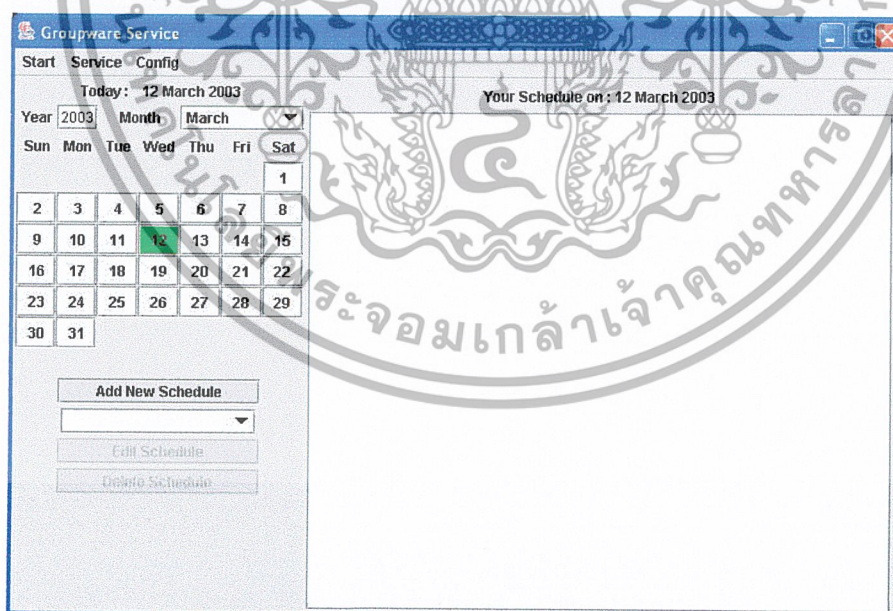
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้รับเอกสารเข้ามาใช้งานจะได้รับการแจ้งถึงเอกสารที่ได้รับและสามารถเข้าไปอ่านเอกสารนั้น ๆ โดยเข้าไปที่หน้า Inbox แล้วเลือกเอกสารที่ต้องการอ่าน



รูปที่ ก.19 แสดงหน้าจอ Inbox

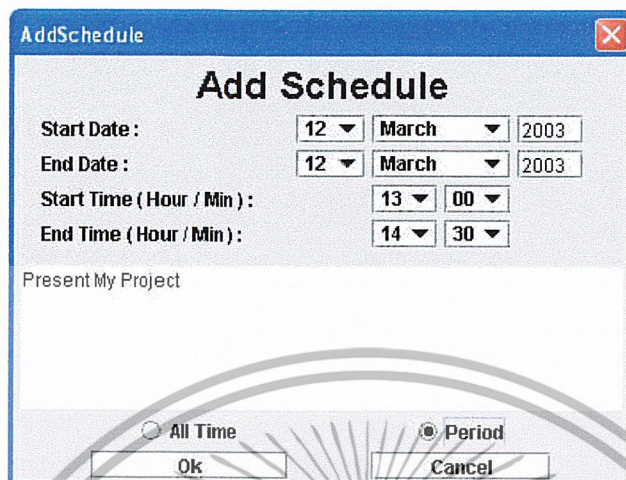
เมื่อต้องการกำหนดตารางเวลานัดหมายสามารถทำได้โดยหน้าจอจะแสดงกำหนดการนัดหมายส่วนบุคคล ดังนี้



รูปที่ ก.20 แสดงหน้าจอแสดงกำหนดการนัดหมายส่วนบุคคล

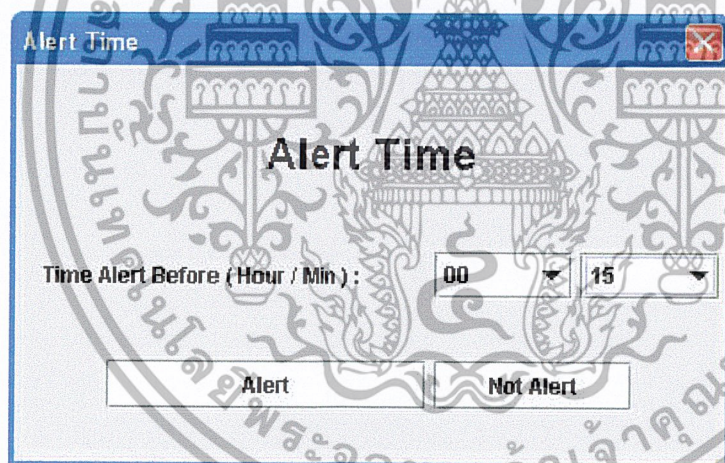
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการเพิ่มกำหนดการนัดหมายสามารถทำได้โดยกดปุ่ม Add New Schedule จะปรากฏหน้าจอ จากรูปที่ ก.21 ได้กำหนดการนัดหมายเป็นเพียงช่วงเวลา 13.00 – 14.30 เท่านั้น



รูปที่ ก.21 แสดงหน้าจอเพิ่มกำหนดการนัดหมายส่วนบุคคล

จากนั้นสามารถกำหนดเวลาเตือนล่วงหน้าได้



รูปที่ ก.22 แสดงหน้าจอกำหนดการแจ้งเตือนล่วงหน้า

ส่วนการนัดหมายภายในองค์กรนั้นทำได้โดยจะมีหน้าจอ ดังรูปที่ ก.23 จากนั้นเราสามารถเลือกกลุ่มงานที่ต้องการนัด โดยคลิกที่ชื่อกลุ่มงาน แล้วหน้าจอจะแสดงรายชื่อสมาชิกในกลุ่มงานนั้น ๆ จากนั้นกดปุ่ม Add เพื่อทำการเพิ่มกำหนดการนัดหมายภายในกลุ่ม

รูปที่ ก.23 แสดงหน้าจอ Organization Schedule

จากนั้นกำหนดรายละเอียดในการนัดหมาย โดยที่ต้องกำหนดช่วงวัน และเวลาที่ต้องการนัดหมาย แล้วโปรแกรมจะแสดงช่วงเวลาว่างของพนักงานในกลุ่มที่สามารถทำการนัดได้ จากนั้นทำการเลือกช่วงเวลาที่ต้องการ และกำหนดหัวข้อ และรายละเอียดในการนัดหมาย ดังรูป 4.24

รูปที่ ก.24 แสดงหน้าจอ Add Organize Schedule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นที่หน้า Organization Schedule จะแสดงกำหนดการนัดหมายในส่วน Organize Schedule ดังนี้

My Group		Member		
WorkGroup		Employee Id	Name-Surname	Position
My Group		E001	Chaleamkwan Pipatc...	Member
		E003	Janthana Sotrakul	Member
		E002	Kid Zodiak	Member

Date	Time	Subject
12 March 2003	14:30 - 16:00	Meeting
12 March 2003	16:00 - 18:30	Project Conference

รูปที่ ก.25 แสดงหน้าจอ Organization Schedule ที่ทำการนัดหมายแล้ว

โปรแกรมจะทำการเพิ่มกำหนดการนัดหมายภายในองค์กรนั้น ๆ ไปในส่วนกำหนดการส่วนบุคคลของพนักงานภายในกลุ่มด้วย ดังรูปที่ ก.26

Today: 12 March 2003 Your Schedule on: 12 March 2003

Year 2003 Month March

Sun Mon Tue Wed Thu Fri Sat

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

12 March 2003 : 13:00 - 14:30 Alert on 12 March 2003 - 12:45
- Present My Project

12 March 2003 : 14:30 - 16:00 Alert on 12 March 2003 - 14:15
- Meeting

12 March 2003 : 16:00 - 18:30 Alert on 12 March 2003 - 16:45
- Project Conference

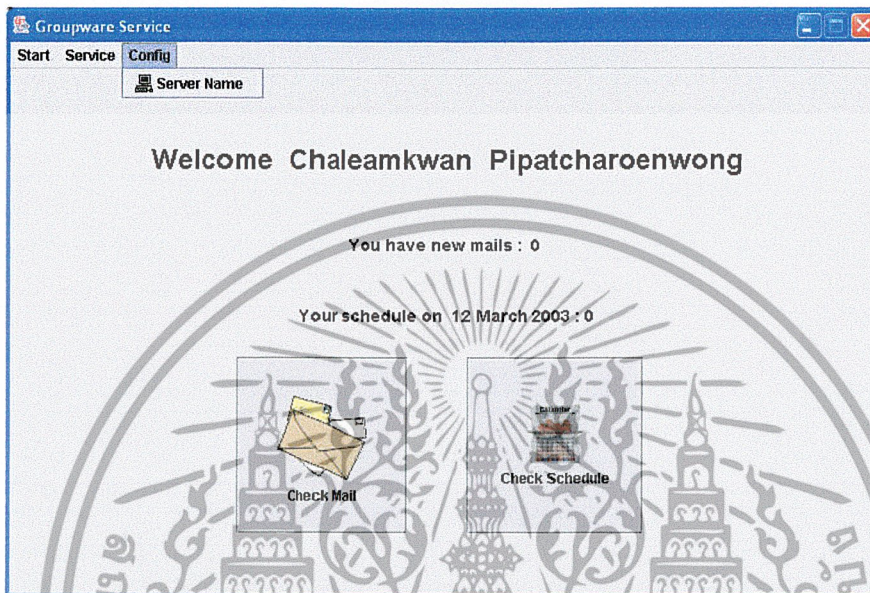
Add New Schedule 13:00 - 14:30 Edit Schedule Delete Schedule

รูปที่ ก.26 แสดงหน้าจอกำหนดการนัดหมายส่วนบุคคลหลังจากมีการนัดหมายภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

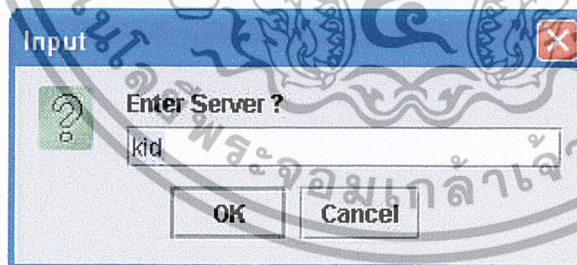
หลังจากนั้นเราสามารถทำการแก้ไขกำหนดการนัดหมาย และลบกำหนดการนั้น ๆ ได้โดยการกดปุ่ม Edit Schedule เมื่อต้องการแก้ไข และปุ่ม Delete Schedule เมื่อต้องการลบกำหนดการ

เมนูส่วน config จะเป็นเมนูในส่วนที่จัดการเกี่ยวกับ server name โดยแสดงดังรูป 4.27



รูปที่ ก.27 แสดงหน้าจอเมนูในส่วน config

หลังจากเลือกเมนู Server Name แล้วจะปรากฏหน้าจอ ดังรูปที่ ก.28



รูปที่ ก.28 แสดงหน้าจอเพื่อให้ผู้ใช้ระบุ Server name

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขั้นตอนการนำคอมโพเนนต์มาใช้

ขั้นตอนการนำคอมโพเนนต์มาใช้นั้นขั้นแรกต้องทำการตั้งค่าใน C:\autoexec.bat ดังนี้

```
SET CLASSPATH = .;C:\Groupware
```

ซึ่งตัวโปรแกรมจะถูกติดตั้งอยู่ที่ C:\Groupware

การแก้ไขส่วน main program ตัวอย่างเช่น ถ้าเราต้องการนำคอมโพเนนต์ในส่วนจัดการสมาชิก และกระจายข้อมูลข่าวสารมาใช้ในองค์กรนั้นเราต้องทำการ import คอมโพเนนต์มาก่อน ดังนี้

การเรียกใช้งานคอมโพเนนต์ฝั่งไคลเอนต์

```
import java.awt.*;
```

```
import groupware.view.*;
```

```
import groupware.control.*;
```

```
import groupware.mailcontrol.*;
```

```
import groupware.employeecontrol.*;
```

คอมโพเนนต์ส่วนการกระจายข้อมูลข่าวสาร

คอมโพเนนต์ส่วนจัดการสมาชิก

```
public class Groupware {
```

```
    private ControlMailRMI controlMail;
```

```
    private ControlEmployeeRMI controlEmp;
```

```
    private String host = new ServerName().getServer();
```

```
    public Groupware() {
```

```
        setComponent();
```

```
        MenuService mainMenu = new MenuService(controlMail, controlEmp);
```

```
        mainMenu.show();
```

```
    }
```

ส่วนที่มีไว้สำหรับจัดการควบคุมการ

ทำงานหลักของคอมโพเนนต์

```
    private void setComponents() {
```

```
        controlMail = new ControlMailRMI(host);
```

→ เป็นการสร้าง controlMail

```
        controlEmp = new ControlEmployeeRMI(host);
```

→ เป็นการสร้าง controlEmp

```
    }
```

```
    public static void main(String[] args) {
```

```
        Groupware groupware = new Groupware();
```

```
    }
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียกใช้งานฝั่งเซิร์ฟเวอร์

```
import java.rmi.*;
import groupware.control.*;
import groupware.mailcontrol.*;
import groupware.mailcontrol.server.*;
import groupware.employeecontrol.*;
import groupware.employeecontrol.server.*;

public class GroupwareServer {
    public static void main(String[] args) {
        new GroupwareServer();
    }
    public GroupwareServer() {
        String host = new ServerName().getServer();
        String name = "/" + host;
        try {
            ControlEmployeeServerRMIInterface controlEmpServer = new
            ControlEmployeeServerRMI();
            controlEmpServer.setEmployeePersistence(EmployeePersistenceetoDB.getInstance());
            controlEmpServer.setDriverDB(driver_DB);
            controlEmpServer.setDriverDB(database_Name);
            Naming.rebind(name + "/ControlEmployeeServerRMIInterface", controlEmpServer);
            ControlMailServerRMIInterface controlMailServer = new ControlMailServerRMI();
            controlMailServer.setMailPresistence(MailPersistenceetoDB.getInstance());
            controlMailServer.setDriverDB(driver_DB);
            controlMailServer.setDBName(database_Name);
            Naming.rebind(name + "/ControlMailServerRMIInterface", controlMailServer);
            System.out.println("Start Groupware Server");
        } catch (Exception e) {    System.err.println("Exception : " + e.getMessage());    }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รายละเอียดของอินเทอร์เฟซ

1. อินเทอร์เฟซในคอมพิวเตอร์สำหรับการจัดการสมาชิก

Interface ControlEmployee

อินเทอร์เฟซ ControlEmployee เป็นอินเทอร์เฟซที่อยู่บนฝั่งไคลเอนต์ โดยจะทำหน้าที่ติดต่อการทำงานระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยจะมีเมธอด ดังนี้

Method Summary	
Employee	userLogin(String user, String pass) ร้องขอการ Login จากเซิร์ฟเวอร์ โดยส่งค่า Username และ password ไป และจะ return ค่าเป็นออบเจกต์ของ Employee
void	UserLogout(String id) ร้องขอการ logout เพื่อออกจากระบบ โดยส่งค่ารหัสพนักงาน
void	SaveEmployee(EmployeeDO emp) ร้องขอการบันทึกข้อมูลพนักงานใหม่ โดยส่งรายละเอียดของพนักงานที่ต้องการบันทึกไปยังเซิร์ฟเวอร์
void	EditEmployee(EmployeeDO emp) ร้องขอการแก้ไขข้อมูลพนักงานใหม่ โดยส่งรายละเอียดของพนักงานที่ต้องการแก้ไขไปยังเซิร์ฟเวอร์
void	removeEmployee(EmployeeDO emp) ร้องขอการลบพนักงาน โดยส่งรายละเอียดของพนักงานที่ต้องการลบไปยังเซิร์ฟเวอร์
Employee	SearchEmpByID(String empId) ร้องขอให้เซิร์ฟเวอร์ค้นหารายละเอียดของพนักงานโดยส่งค่ารหัสพนักงานไปให้ และจะ return ค่าออบเจกต์ของ Employee
Employee	SearchEmpByUserName(String userName) ร้องขอให้เซิร์ฟเวอร์ค้นหารายละเอียดของพนักงานโดยส่งชื่อผู้ใช้งานไปให้ และจะ return ค่าเป็นออบเจกต์ของ EmployeeDO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Employee[]	SearchEmpByWorkGroupName(String name) ร้องขอให้ค้นหารายละเอียดค้นหารายละเอียดพนักงานโดยส่งค่าชื่อกลุ่มงานไปให้ และ return ค่าเป็น array ของ EmployeeDO
Employee[]	SearchEmpByCase(String name, String surname, String user) ร้องขอให้ค้นหารายละเอียดพนักงานจากชื่อ นามสกุล หรือชื่อผู้ใช้ของพนักงาน โดยจะ return ค่าเป็น array ของ EmployeeDO
Employee[]	SearchAllEmp() ร้องขอให้ค้นหารายละเอียดของพนักงานทั้งหมดของพนักงาน โดยจะ return ค่าเป็น array ของ Employee
void	SaveWorkGroup(WorkGroupDO group) ร้องขอการบันทึกกลุ่มงานใหม่ โดยส่งรายละเอียดของกลุ่มงานที่ต้องการบันทึกไปยังเซิร์ฟเวอร์
void	EditWorkGroup(WorkGroupDO group) ร้องขอการแก้ไขรายละเอียดของกลุ่มงานโดยส่งรายละเอียดของกลุ่มงานไปยังเซิร์ฟเวอร์
void	RemoveWorkGroup(WorkGroupDO group) ร้องขอการลบกลุ่มงาน โดยส่งรายละเอียดของกลุ่มงานไปยังเซิร์ฟเวอร์
WorkGroup	SearchWorkGroupByPK(String workId) ร้องขอการค้นหารายละเอียดของกลุ่มงาน โดยจะส่งค่ารหัสของกลุ่มงานไปยังเซิร์ฟเวอร์ ซึ่งจะ return ค่าเป็นออบเจกต์ของ WorkGroup
WorkGroup[]	SearchAllWorkGroup() ร้องขอการค้นหารายละเอียดของกลุ่มงานทั้งหมด โดยจะ return ค่าเป็น array ของ WorkGroup
String	SearchPriorityByID(String prior) ร้องขอการค้นหารายละเอียดของสิทธิในการเข้าใช้ข้อมูลจากรหัสสิทธิ
String	SearchPriorityByDetail(String detail) ร้องขอการค้นหารายละเอียดของสิทธิในการเข้าใช้ข้อมูลจากรายละเอียดสิทธิ
String[][]	SearchAllPriority() ร้องขอค้นหารายละเอียดของสิทธิในการเข้าใช้ข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Employee[]	ShowAddress(String id) ร้องขอการค้นหารายละเอียดของสมุดที่อยู่ของผู้ใช้แต่ละคน
void	InsertAddress(String id, EmployeeDO emp) ร้องขอบันทึกรายละเอียดของสมุดที่อยู่ของผู้ใช้แต่ละคน
void	RemoveAddress(String userId, String listId) ร้องขอการลบรายละเอียดของสมุดที่อยู่ของผู้ใช้แต่ละคน

ตารางที่ ข.1 แสดงเมธอดใน Interface ControlEmployee

Interface ControlEmployeeServer

อินเทอร์เฟซ ControlEmployeeServer เป็นอินเทอร์เฟซที่อยู่ฝั่งเซิร์ฟเวอร์ ทำหน้าที่รับ
การร้องขอความต้องการจากไคลเอนต์ และตอบสนองความต้องการไปยังฝั่งไคลเอนต์ โดยมีเมธอด
ดังนี้

Method Summary	
void	setEmployeePersistence(EmployeePersistence employeePersistence) เป็นการกำหนดรูปแบบการทำงานให้กับ Employee Server
Employee	doLogin(String user, String pass) รับการร้องขอการ login โดยรับค่า Username และ password และ จะ return ค่าเป็นออบเจกต์ของ Employee
void	DoLogout(String id) รับการร้องขอการ Logout ออกจากระบบ โดยรับค่ารหัสพนักงาน มาจากไคลเอนต์
void	insertEmployee(EmployeeDO emp) รับการร้องขอการเพิ่มพนักงานใหม่เข้าสู่ระบบ โดยรับค่าเป็น ออบเจกต์ของ EmployeeDO
void	UpdateEmployee(EmployeeDO emp) รับการร้องขอการแก้ไขรายละเอียดของพนักงาน โดยรับค่าเป็น ออบเจกต์ของ EmployeeDO
void	deleteEmployee(EmployeeDO emp) รับการร้องขอการลบรายละเอียดพนักงาน โดยรับค่าเป็นออบเจกต์ ของ EmployeeDO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Employee	searchEmployeeByID(String empId) รับการร้องขอการค้นหานักงานโดยรับค่ารหัสพนักงาน
Employee	SearchEmployeeByUserName(String userName) รับการร้องขอการค้นหานักงานโดยรับค่า username และจะ return ค่าเป็นออบเจ็กต์ของ EmployeeDO
Employee[]	SearchEmployeeByWorkGroupName(String name) รับการร้องขอการค้นหานักงานโดยรับค่าชื่อกลุ่มงาน และจะ return ค่าเป็น array ของ EmployeeDO
Employee[]	SearchEmployeeByCase(String name, String surname, String user) รับการร้องขอการค้นหานักงาน โดยรับค่าชื่อ นามสกุล หรือชื่อผู้ ใช้ และ return ค่าเป็น array ของ EmployeeDO
Employee[]	searchAllEmployee() รับการร้องขอการค้นหานักงานทั้งหมด โดยจะ return ค่าเป็น array ของ Employee
void	insertWorkGroup(WorkGroupDO group) รับการร้องขอการเพิ่มกลุ่มงานใหม่ โดยรับค่าเป็นออบเจ็กต์ของ WorkGroupDO
void	updateWorkGroup(WorkGroupDO group) รับการร้องขอการแก้ไขรายละเอียดของกลุ่มงาน โดยรับค่าเป็น ออบเจ็กต์ของ WorkGroupDO
void	deleteWorkGroup(WorkGroupDO group) รับการร้องขอการลบกลุ่มงานออกจากระบบ โดยรับค่าเป็น ออบเจ็กต์ของ WorkGroupDO
WorkGroup	findWGByPK(String workId) รับการร้องขอการค้นหากกลุ่มงานจากคีย์หลัก ซึ่งก็คือรหัสกลุ่มงาน นั่นเอง แล้ว return ค่าเป็นออบเจ็กต์ของ WorkGroup
WorkGroup[]	findAllWorkGroup() รับการร้องขอการค้นหากกลุ่มงานทั้งหมด โดย return ค่าเป็น array ของ WorkGroup
String	FindPriorityByID(String prior) รับการร้องขอการค้นหารายละเอียดของสิทธิ์ในการเข้าใช้ข้อมูล จากรหัสสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String	FindPriorityByDetail(String detail) รับการร้องขอการค้นหารายละเอียดของสิทธิในการเข้าใช้ข้อมูล จากรายละเอียดของสิทธิ
String[][]	findAllPriority() รับการร้องขอการค้นหาสิทธิการในการเข้าใช้ข้อมูลทั้งหมด
Employee[]	SearchEmpInAddressBook(String id) รับการร้องขอการค้นหารายละเอียดของสมุดที่อยู่ของผู้ใช้แต่ละคน โดยรับค่าเป็นรหัสพนักงาน และ return ค่าเป็น array ของ Employee
void	insertAddressBook(String id, EmployeeDO emp) รับการร้องขอการเพิ่มรายละเอียดของสมุดที่อยู่ของผู้ใช้แต่ละคน โดยจะรับค่ารหัสพนักงาน และออบเจกต์ของ EmployeeDo
void	deleteAddress(String userId, String listId) รับการร้องขอการลบรายละเอียดของสมุดที่อยู่ของผู้ใช้แต่ละคน โดยรับค่ารหัสของพนักงาน และรหัสของสมุดที่อยู่ที่ต้องการลบ

ตารางที่ ข.2 แสดงเมธอดในอินเทอร์เฟซ ControlEmployeeServer

Interface ControlEmployeeServerRMInterface

อินเทอร์เฟซ ControlEmployeeServerRMInterface เป็นอินเทอร์เฟซที่ทำให้ไคลเอนต์และเซิร์ฟเวอร์สามารถติดต่อกันได้ โดยที่มีการร้องขอในรูปแบบของ rmi โดยมีเมธอด ดังนี้

Method Summary	
void	setEmployeePersistence(EmployeePersistence employeePersistence) เป็นการกำหนดรูปแบบการทำงานของระบบการจัดการสมาชิก
void	setDriverDB(String driverDB) เป็นการกำหนดไดรเวอร์ของฐานข้อมูลที่ทำกรติดต่อด้วย โดยรับ ค่าเป็นไดรฟ์เวอร์ของฐานข้อมูล
void	setDBName(String databaseName) เป็นการกำหนดชื่อของฐานข้อมูลที่ทำกรติดต่อด้วย โดยรับค่าเป็น ชื่อของฐานข้อมูล

Employee	doLogin(String user, String pass) รับการร้องขอการ login เพื่อเข้าสู่ระบบโดยรับค่าเป็น Username และ Password แล้วทำการ return ค่าเป็นออบเจกต์ของ Employee
void	doLogout(String id) รับการร้องขอการ logout เพื่อออกจากระบบ โดยรับค่าเป็นรหัสของพนักงานที่ต้องการออกจากระบบ
void	insertEmployee(EmployeeDO emp) รับการร้องขอการเพิ่มพนักงานใหม่เข้าสู่ระบบ โดยรับค่าเป็นออบเจกต์ของ EmployeeDO
void	updateEmployee(EmployeeDO emp) รับการร้องขอการแก้ไขข้อมูลของพนักงานโดยรับค่าเป็นออบเจกต์ของ EmployeeDO
void	deleteEmployee(EmployeeDO emp) รับการร้องขอการลบพนักงานออกจากระบบ โดยรับค่าเป็นออบเจกต์ของ EmployeeDO
Employee	searchEmployeeByID(String empId) รับการร้องขอการค้นหพนักงานจากรหัสพนักงาน โดยรับค่าเป็นรหัสพนักงาน และ return ค่าเป็นออบเจกต์ของ Employee
Employee	searchEmployeeByUserName(String userName) รับการร้องขอการค้นหพนักงานจากชื่อผู้ใ้ โดยรับค่าเป็นชื่อผู้ใ้ และ return ค่าเป็นออบเจกต์ของ EmployeeDO
Employee[]	searchEmployeeByWorkGroupName(String name) รับการร้องขอการค้นหาข้อมูลพนักงานจากชื่อกลุ่มงาน โดยรับค่าเป็นชื่อของกลุ่มงาน และ return ค่าเป็น array เป็น EmployeeDO
Employee[]	searchEmployeeByCase(String name, String surname, String user) รับการร้องขอรายละเอียดพนักงานจากชื่อ นามสกุล และชื่อผู้ใ้ โดย return ค่าเป็น array ของ EmployeeDO
Employee[]	searchAllEmployee() รับการร้องขอการค้นหพนักงานทั้งหมด โดยจะ return ค่าเป็น array ของ Employee
void	insertWorkGroup(WorkGroupDO group)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รับการร้องขอการเพิ่มกลุ่มงานใหม่เข้าสู่ระบบ โดยรับค่าเป็น ออบเจกต์ของ WorkGroupDO
void	updateWorkGroup(WorkGroupDO group) รับการร้องขอการแก้ไขรายละเอียดของกลุ่มงาน โดยรับค่าเป็น ออบเจกต์ของ WorkGroupDO
void	deleteWorkGroup(WorkGroupDO group) รับการร้องขอการลบกลุ่มงานออกจากระบบ โดยรับค่าเป็น ออบเจกต์ของ WorkGroupDO
WorkGroup	findWGBByPK(String workId) รับการร้องขอการค้นหากลุ่มงานจากคีย์หลัก โดยรับค่าเป็นรหัส ของกลุ่มงาน และ return ค่าเป็นออบเจกต์ของ WorkGroup
WorkGroup[]	findAllWorkGroup() รับการร้องขอการค้นหากลุ่มงานทั้งหมด โดย return ค่าเป็น array ของ WorkGroup
String	findPriorityByID(String prior) รับการร้องขอการค้นหาสิทธิในการเข้าใช้ข้อมูลจากรหัสสิทธิ
String	findPriorityByDetail(String detail) รับการร้องขอการค้นหาสิทธิในการเข้าใช้ข้อมูลจากรายละเอียด ของสิทธิ
String[][]	findAllPriority() รับการร้องขอการค้นหาสิทธิในการเข้าใช้ข้อมูลทั้งหมด
Employee[]	searchEmpInAddressBook(String id) รับการร้องขอการค้นหานักงานจากสมุดที่อยู่ โดยส่งค่ารหัส พนักงาน และ return ค่าเป็น array ของ Employee
void	insertAddressBook(String id, EmployeeDO emp) รับการร้องขอการเพิ่มรายละเอียดของสมุดที่อยู่ โดยรับค่ารหัส พนักงาน และออบเจกต์ของ EmployeeDO
void	deleteAddress(String userId, String listId) รับการร้องขอการลบสมุดที่อยู่ โดยรับค่าเป็นรหัสพนักงาน และ รหัสสมุดที่อยู่

ตารางที่ข.3 แสดงเมธอดในอินเทอร์เฟซ ControlEmployeeServerRMIInterface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface EmployeePersistence

อินเทอร์เฟซ EmployeePersistence เป็นอินเทอร์เฟซของการทำงานของระบบการจัดการสมาชิกที่ทำงานอยู่บนฝั่งเซิร์ฟเวอร์ โดยมีเมธอดดังนี้

Method Summary	
void	setDriver(String driver) เป็นการกำหนดไดรฟ์เวอร์ของฐานข้อมูล โดยรับค่าไดรฟ์เวอร์ของฐานข้อมูลมา
void	setDatabase(String db) เป็นการกำหนดชื่อให้กับฐานข้อมูลที่ต้องการติดต่อด้วย โดยรับค่าชื่อฐานข้อมูลมา
Employee	userLogin(String user, String pass) เป็นการ login เพื่อเข้าสู่ระบบ โดยรับค่า Username และ Password แล้วทำการ return ค่าเป็นออบเจกต์ของEmployee
void	userLogout(String id) เป็นการ logout เพื่อออกจากระบบ โดยรับค่ารหัสพนักงานมา
void	insertEmp(EmployeeEntity emp) เป็นการเพิ่มพนักงานใหม่เข้าสู่ระบบ โดยรับค่าเป็นออบเจกต์ของEmployeeEntity
void	updateEmp(EmployeeEntity emp) เป็นการแก้ไขข้อมูลของพนักงานในระบบ โดยรับค่าเป็นออบเจกต์ของ EmployeeEntity
void	deleteEmp(EmployeeEntity emp) เป็นการลบพนักงานออกจากระบบ โดยรับค่าเป็นออบเจกต์ของ EmployeeEntity
Employee	findEmpByID(String id) เป็นการค้นหาพนักงานจากรหัสพนักงาน โดยรับค่าเป็นรหัสพนักงาน และ return ค่าเป็นออบเจกต์ของ Employee
Employee	findEmpByUserName(String userName) เป็นการค้นหาพนักงานจากชื่อผู้ใช้ โดยรับค่าชื่อผู้เข้ามา และ return

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ค่าเป็นออบเจกต์ของ EmployeeEntity
Employee[]	searchEmpByWorkGroupName(String name) เป็นการค้นหาพนักงานจากชื่อของกลุ่มงาน โดยรับค่าเป็นชื่อของกลุ่มงาน และ return ค่าเป็น array ของ EmployeeEntity
Employee[]	searchEmpByCase(String name, String surname, String user) เป็นการค้นหาพนักงานจาก ชื่อ นามสกุล หรือ ชื่อผู้ใช้ โดยจะ return ค่าเป็น array ของ EmployeeEntity
Employee[]	findAllEmp() เป็นการค้นหาพนักงานทั้งหมด โดยจะ return ค่าเป็น array ของ Employee
void	insertWG(WorkGroupEntity group) เป็นการเพิ่มกลุ่มงานใหม่เข้าสู่ระบบ โดยรับค่าเป็นออบเจกต์ของ WorkGroupEntity
void	updateWG(WorkGroupEntity group) เป็นการแก้ไขรายละเอียดของกลุ่มงานที่มีอยู่ในระบบ โดยรับค่าเป็นออบเจกต์ของ WorkGroupEntity
void	delWorkGroup(WorkGroupEntity group) เป็นการลบกลุ่มงานที่มีอยู่ โดยรับค่าเป็นออบเจกต์ของ WorkGroupEntity
WorkGroup	findGroupByPK(String workId) เป็นการค้นหากกลุ่มงานจากคีย์หลัก โดยรับค่าเป็นรหัสของกลุ่มงาน และทำการ return ค่าเป็นออบเจกต์ของ WorkGroup
WorkGroup	searchAllWG() เป็นการค้นหากกลุ่มงานทั้งหมดที่มีในระบบ โดยจะ return ค่าเป็นออบเจกต์ของ WorkGroup
String	getPriorityId(String detail) เป็นการขอรหัสสิทธิ์ในการเข้าใช้ข้อมูล โดยรับค่าเป็นรายละเอียดของสิทธิ์ และ return ค่าเป็น String ของรหัสสิทธิ์
String	getPriorityDetail(String prior) เป็นการขอรายละเอียดของสิทธิ์ในการเข้าใช้ข้อมูล โดยรับค่าเป็นสิทธิ์ของการใช้งาน และ return ค่าเป็นรายละเอียดของสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String[][]	searchAllPrior() เป็นการค้นหาสิทธิ์ทั้งหมดโดยจะ return ค่าเป็น array ของ String ที่บอกถึงรายละเอียดทั้งหมดของสิทธิ์
Employee[]	showAddressBook(String id) เป็นการแสดงสมุดที่อยู่ของพนักงานแต่ละคน โดยรับค่าเป็นรหัสพนักงาน และ return ค่าเป็น array ของ Employee
void	insertAddress(String id, EmployeeEntity emp) เป็นการเพิ่มรายละเอียดของสมุดที่อยู่โดยรับค่าเป็นรหัสผู้ใช้สมุดที่อยู่ และออบเจกต์ของ EmployeeEntity
void	deleteAddressBook(String userId, String listId) เป็นการลบสมุดที่อยู่ของพนักงานแต่ละคน โดยรับค่าเป็นรหัสพนักงานที่ต้องการลบสมุดที่อยู่ และรหัสของสมุดที่อยู่

ตารางที่ ข.4 แสดงเมธอดในอินเทอร์เฟซ EmployeePersistence

Interface Employee

อินเทอร์เฟซ Employee เป็นอินเทอร์เฟซที่ใช้สำหรับรับ-ส่งรายละเอียดของพนักงาน โดยมีเมธอด ดังนี้

Method Summary	
String	getEmpId() ร้องขอรหัสพนักงาน
String	getName() ร้องขอชื่อพนักงาน
String	getSurname() ร้องขออนามสกุลของพนักงาน
String	getUsername() ร้องขอชื่อผู้ใช้ของพนักงาน
String	getPassword() ร้องขอรหัสผ่านของพนักงาน

ตารางที่ ข.5 แสดงเมธอดของอินเทอร์เฟซ Employee

Interface WorkGroup

อินเทอร์เฟซ WorkGroup เป็นอินเทอร์เฟซที่ใช้สำหรับรับ-ส่งรายละเอียดของกลุ่มงาน โดยมีเมธอด ดังนี้

Method Summary	
String	getWorkId() ร้องขอรหัสกลุ่มงาน
String	getWorkName() ร้องขอชื่อกลุ่มงาน
Employee[]	getEmployee() ร้องขอรายละเอียดพนักงานภายในกลุ่มงาน โดยจะ return ค่าเป็น array ของ Employee
String[]	getEmpPriority() ร้องขอสิทธิ์การเข้าใช้ข้อมูลของพนักงานในกลุ่มงาน
String[]	getEmpPosition() ร้องขอตำแหน่งของพนักงานในกลุ่มงาน
Boolean[]	getEmpStatus() ร้องขอสถานะของพนักงานในกลุ่มงานว่าพนักงานแต่ละคนยอมรับการเข้าร่วมกลุ่มหรือไม่

ตารางที่ ข.6 แสดงเมธอดในอินเทอร์เฟซ WorkGroup

2. อินเทอร์เฟซในคอมพิวเตอร์สำหรับกระจายข้อมูลข่าวสาร

Interface ControlMail

อินเทอร์เฟซ ControlMail เป็นอินเทอร์เฟซที่อยู่บนฝั่งไคลเอนต์ โดยจะทำหน้าที่ติดต่อกับเซิร์ฟเวอร์โดยจะมีเมธอด ดังนี้

Method Summary	
void	sendMail(MailDO m) ร้องขอการส่งเอกสารไปยังบุคคลตามที่กำหนดโดยจะส่งค่า ออบเจกต์ของ MailDO ไปยังเซิร์ฟเวอร์
void	sendGroupMail(MailDO[] m) ร้องขอการส่งเอกสารไปยังบุคคลในกลุ่มงานที่กำหนด โดยจะส่งค่า เป็น array ของ MailDO ไปยังเซิร์ฟเวอร์
Mail[]	readAllMail(String userId) ร้องขอการอ่านเอกสารทั้งหมดของผู้ใช้แต่ละคน โดยจะส่งค่ารหัส พนักงานไป และ return ค่าเป็น array ของ Mail
Mail	readMailByPK(String docId,String userId) ร้องขอการอ่านเอกสารจากคีย์หลัก โดยส่งค่ารหัสเอกสาร และรหัส พนักงาน และ return ค่าเป็นออบเจกต์ของ Mail
void	removeMail(String docId,String userId) ร้องขอการลบเอกสารตามทีระบุโดยส่งค่ารหัสเอกสาร และรหัส พนักงาน ไปยังเซิร์ฟเวอร์
void	removeAllMail(String userId) ร้องขอการลบเอกสารทั้งหมดของผู้ใช้แต่ละคน โดยจะส่งค่ารหัส พนักงานไปยังเซิร์ฟเวอร์

ตารางที่ ข.7 แสดงเมธอดในอินเทอร์เฟซ ControlMail

Interface ControlMailServer

อินเทอร์เฟซ ControlMailServer เป็นอินเทอร์เฟซที่อยู่ฝั่งเซิร์ฟเวอร์ ที่ทำหน้าที่รับการร้องขอความต้องการจากไคลเอนต์ และตอบสนองความต้องการไปยังฝั่งไคลเอนต์ โดยมีเมธอด ดังนี้

Method Summary	
void	setMailPresistence(MailPersistence mailPersistence) เป็นการกำหนดรูปแบบการทำงานของระบบรับ-ส่งเอกสาร
Mail[]	getMailAll(String revId) รับการร้องขอการอ่านเอกสารทั้งหมด โดยจะรับค่ารหัสของผู้รับเอกสาร และ return ค่าเป็น array ของ Mail
Mail	getMailByPK(String docId,String userId) รับการร้องขอการอ่านเอกสารโดยคีย์หลัก โดยรับค่ารหัสเอกสารและรหัสพนักงาน แล้ว return ค่าเป็นออบเจกต์ของ Mail
void	saveMail(MailDO group) รับการร้องขอการส่งเอกสาร โดยรับค่าเป็นออบเจกต์ของ MailDO
void	saveGroupMail(MailDO[] group) รับการร้องขอการส่งเอกสารเป็นกลุ่ม โดยรับค่าเป็น array ของ MailDO
void	deleteMail(String docId,String userId) รับการร้องขอการลบเอกสาร โดยรับค่ารหัสเอกสาร และรหัสพนักงานที่ต้องการลบเอกสารมา
void	deleteAllMail(String userId) รับการร้องขอการลบเอกสารทั้งหมด โดยรับค่ารหัสพนักงานที่ต้องการลบเอกสาร

ตารางที่ ข.8 แสดงเมธอดในอินเทอร์เฟซ ControlMailServer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface ControlMailServerRMIIInterface

อินเทอร์เฟซ ControlMailServerRMIIInterface เป็นอินเทอร์เฟซที่ให้ไคลเอนต์ติดต่อกับเซิร์ฟเวอร์ โดยใช้รูปแบบการติดต่อแบบ rmi โดยเฉพาะ เพราะว่ารูปแบบการติดต่อแบบ rmi จะต้องมีส่วน RemoteException แต่การติดต่อแบบอื่นๆ จะไม่จำเป็นต้องมี RemoteException

Method Summary	
void	setMailPresistence(MailPersistence mailPersistence) เป็นการกำหนดรูปแบบการทำงานของระบบการรับ-ส่งเอกสาร
void	setDriverDB(String driverDB) เป็นการกำหนดไดรฟ์เวอร์ของฐานข้อมูลที่ใช้ในการติดต่อ โดยส่งค่าเป็นไดรฟ์เวอร์ของฐานข้อมูลไป
void	setDBName(String databaseName) เป็นการกำหนดชื่อของฐานข้อมูลที่ใช้ โดยส่งชื่อของฐานข้อมูลไปยังเซิร์ฟเวอร์
Mail[]	getMailAll(String revId) ร้องขอการรับเอกสารทั้งหมด โดยจะส่งรหัสผู้รับเอกสารไปยังเซิร์ฟเวอร์และ return ค่าเป็น array ของ Mail
Mail	getMailByPK(String docId,String userId) ร้องขอการรับเอกสารโดยใช้คีย์หลัก โดยจะส่งค่ารหัสเอกสาร และรหัสผู้รับเอกสาร ไปยังเซิร์ฟเวอร์ และ return ค่าเป็นออบเจกต์ของ Mail
void	saveMail(MailDO group) ร้องขอการส่งเอกสารโดยส่งค่าเป็นออบเจกต์ของ MailDO
void	saveGroupMail(MailDO[] group) ร้องขอการส่งเอกสารเป็นกลุ่ม โดยส่งค่าเป็น array ของMailDO
void	deleteMail(String docId,String userId) ร้องขอการลบเอกสาร โดยส่งค่ารหัสเอกสาร และรหัสพนักงานไปยังเซิร์ฟเวอร์
void	deleteAllMail(String userId) ร้องขอการลบเอกสารทั้งหมด โดยส่งค่ารหัสพนักงานที่ต้องการลบเอกสารไปยังเซิร์ฟเวอร์

ตารางที่ ข.9 แสดงเมธอดในอินเทอร์เฟซ ControlMailServerRMIIInterface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface MailPersistence

อินเทอร์เฟซ MailPersistence เป็นอินเทอร์เฟซของการทำงานของระบบการจัดการเอกสาร โดยจะทำงานในฝั่งเซิร์ฟเวอร์ โดยมีเมธอด ดังนี้

Method Summary	
void	setDriver(String driver) เป็นการกำหนดไดรฟ์เวอร์ของฐานข้อมูลที่ใช้ โดยรับค่าไดรฟ์เวอร์
void	setDatabase(String db) เป็นการกำหนดชื่อฐานข้อมูล โดยรับค่าชื่อของฐานข้อมูลมา
Mail[]	searchMailAll(String id) เป็นการค้นหาเอกสารทั้งหมด โดยรับค่ารหัสพนักงานมาแล้ว return ค่าเป็น array ของ Mail[]
Mail	findMailByPK(String docId,String userId) เป็นการค้นหาจดหมายโดยใช้คีย์หลักซึ่งก็คือ รหัสเอกสาร และ รหัสผู้รับเอกสาร แล้ว return ค่ากลับเป็นออบเจกต์ของ MailEntity
void	saveSendMail(MailEntity mail) เป็นการบันทึกเอกสารโดยรับค่าออบเจกต์ของMailEntity
void	saveSendGroupMail(MailEntity[] mail) เป็นการบันทึกเอกสารเป็นกลุ่ม โดยรับค่า array ของ MailEntity
void	delMail(String docId,String userId) เป็นการลบเอกสาร โดยรับค่ารหัสเอกสาร และรหัสพนักงาน
void	delAllMail(String userId) เป็นการลบเอกสารทั้งหมด โดยรับค่ารหัสพนักงาน
void	deleteAddr(String id, String listId) เป็นการลบสมุดที่อยู่โดยรับค่ารหัสพนักงาน และรหัสสมุดที่อยู่

ตารางที่ ข.10 แสดงเมธอดในอินเทอร์เฟซ MailPersistence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface Mail

อินเทอร์เฟซ Mail เป็นอินเทอร์เฟซที่ใช้สำหรับรับ-ส่งรายละเอียดของเอกสาร โดยมีเมธอด ดังนี้

Method Summary	
String	getDocId() ร้องขอค่ารหัสเอกสาร
String	getSendId() ร้องขอรหัสผู้ส่งเอกสาร
String []	getRevId() ร้องขอรหัสผู้รับเอกสาร
String	getSubject() ร้องขอหัวข้อเอกสาร
String	getAttach() ร้องขอชื่อไฟล์ที่ทำการแนบมากับเอกสาร
byte[]	getFileAttach() ร้องขอไฟล์ข้อมูลที่ทำการแนบมากับเอกสาร
String	getMessage() ร้องขอข้อความของเอกสาร
String	getSendDate() ร้องขอวันที่ทำการส่งเอกสาร
String []	getType() ร้องขอประเภทของการส่งเอกสารว่าเป็นการส่งเอกสารถึงผู้รับแบบ ธรรมดา(To) การส่งเอกสารโดยส่งสำเนาไปยังผู้รับ (Cc) หรือการส่ง เอกสารโดยการส่งสำเนาโดยที่ไม่แสดงชื่อผู้รับ (Bcc)
boolean	getStatus() ร้องขอสถานะของเอกสารของผู้รับแต่ละคน ว่าถูกอ่านหรือยัง

ตารางที่ ข.11 แสดงเมธอดของอินเทอร์เฟซ Mail

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. อินเทอร์เฟซในคอมโพเนนต์สำหรับการจัดการตารางเวลา

Interface ControlSchedule

อินเทอร์เฟซ ControlSchedule เป็นอินเทอร์เฟซที่อยู่บนฝั่งไคลเอนต์ โดยจะทำหน้าที่ติดต่อกับการทำงานระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยจะมีเมธอด ดังนี้

Method Summary	
void	savePersonalSch(PersonalSchDO personSch) ร้องขอการบันทึกตารางเวลาส่วนบุคคลโดยจะส่งค่าเป็นออบเจกต์ของ PersonalSchDO ไปยังเซิร์ฟเวอร์
void	editPersonalSch(PersonalSchDO personSch, String olddate, String oldtimeSt, String oldtimeFn) ร้องขอการแก้ไขรายละเอียดของตารางเวลาส่วนบุคคล โดยส่งค่าออบเจกต์ของ PersonalSchDO วันที่ เวลาเริ่มต้นการนัดหมาย และเวลาสิ้นสุดการนัดหมายที่ต้องการแก้ไข
void	removePersonalSchByTime(PersonalSchDO personSch) ร้องขอการลบตารางเวลาส่วนบุคคลโดยใช้เวลา ซึ่งจะส่งค่าเป็นออบเจกต์ของ PersonalSchDO
PersonalSch[]	searchPersonalSchByDate(PersonalSchDO personSch) ร้องขอการค้นหารายละเอียดของตารางเวลาส่วนบุคคลจากวันที่ โดยส่งค่าออบเจกต์ของ PersonalSchDO และ return ค่าเป็น array ของ PersonalSch
void	saveAppointment(OrganizeSchDO orgSch) ร้องขอการบันทึกกำหนดการนัดหมายภายในองค์กร โดยส่งค่าออบเจกต์ของ OrganizeSchDO
void	removeAppointment(OrganizeSchDO orgSch) ร้องขอการลบกำหนดการนัดหมายภายในองค์กร โดยส่งค่าเป็นออบเจกต์ของ OrganizeSchDO ไปยังเซิร์ฟเวอร์

OrganizeSch[]	searchAppointmentByGroup(String workId) ร้องขอการค้นหารายละเอียดของการนัดหมายภายในองค์กรจากรหัสกลุ่มงาน โดยส่งค่ารหัสของกลุ่มงานไป และ return ค่าเป็น array ของ OrganizeSch
String[][]	searchTimeToAppoint(SearchDurationDO search) ร้องขอการค้นหาวเวลาที่สามารถทำการนัดหมายได้ โดยส่งค่าเป็นออบเจ็กต์ของ SearchDurationDO และ return ค่าเป็นช่วงเวลาที่สามารถนัดหมายได้กลับไปให้ในรูปแบบของ array ของ String
PersonalSch[]	searchAlert(String id) ร้องขอการค้นหาการเตือนล่วงหน้า โดยส่งค่ารหัสพนักงานไป และ returnค่าเป็น array ของ Personalsch
void	setAlert(PersonalSchDO alert) ร้องขอการกำหนดการแจ้งเตือนล่วงหน้า โดยส่งค่าเป็นออบเจ็กต์ของ PersonalSchDO
boolean	getAlert(PersonalSchDO alert) ร้องขอการรับการแจ้งเตือนล่วงหน้า โดยส่งค่าเป็นออบเจ็กต์ของ PersonalSchDO แล้ว return ค่าเป็นแบบ boolean

ตารางที่ ข.12 แสดงเมธอดในอินเทอร์เฟซ ControlSchedule

Interface ControlScheduleServer

อินเทอร์เฟซ ControlScheduleServer เป็นอินเทอร์เฟซที่อยู่ฝั่งเซิร์ฟเวอร์ ที่ทำหน้าที่รับการร้องขอความต้องการจากไคลเอนต์ และตอบสนองความต้องการไปยังฝั่งไคลเอนต์ โดยมีเมธอดดังนี้

Method Summary	
void	setSchedulePersistence(SchedulePersistence schedulePersistence) เป็นการกำหนดรูปแบบการทำงานของระบบการจัดการตารางเวลานัดหมาย
void	savePersonalSch(PersonalSchDO person) รับการร้องขอการบันทึกกำหนดการนัดหมายส่วนบุคคล โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchDO

void	updatePersonalSch(PersonalSchDO person, String olddate, String oldtimeSt, String oldtimeFn) รับการร้องขอการแก้ไขกำหนดการนัดหมายส่วนบุคคล โดยจะรับค่าเป็นออบเจ็กต์ของ PersonalSchDO วันที่ เวลาเริ่มต้น และเวลาสิ้นสุดที่ต้องการแก้ไข
void	removePersonalSchByTime(PersonalSchDO person) รับการร้องขอการลบกำหนดการนัดหมายส่วนบุคคลจากเวลา โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchDO
PersonalSch[]	searchPsByDate(PersonalSchDO person) รับการร้องขอการค้นหาข้อมูลการนัดหมายส่วนบุคคลจากวันที่ โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchDO และ return ค่าเป็น array ของ PersonalSch
void	saveAppoint(OrganizeSchDO orgSch) รับการร้องขอการบันทึกการนัดหมายภายในองค์กร โดยรับค่าเป็นออบเจ็กต์ของ OrganizeSchDO
void	removeAppoint(OrganizeSchDO orgSch) รับการร้องขอการลบกำหนดการนัดหมายภายในองค์กร โดยรับค่าเป็นออบเจ็กต์ของ OrganizeSchDO
OrganizeSch[]	searchAppointByGroup(String workId) รับการร้องขอการค้นหากำหนดการนัดหมายภายในองค์กรจากกลุ่มงาน โดยรับค่าเป็นรหัสของกลุ่มงาน และ return ค่าเป็น array ของ OrganizeSch
String[][]	searchTime(SearchDurationDO search) รับการร้องขอการค้นหาวงเวลาที่ทำการนัดหมาย โดยจะรับค่าเป็นออบเจ็กต์ของ SearchDurationDO และ return ค่าเป็น array ของ String เพื่อบอกช่วงเวลาที่สามารถนัดหมายได้
PersonalSch[]	searchAlertInDay(String id) รับการร้องขอการค้นหาการแจ้งเตือนล่วงหน้าในวันนั้น โดยรับค่าเป็นรหัสพนักงาน และ return ค่าเป็น array ของ PersonalSchDO
void	setAlertType(PersonalSchDO alert) รับการร้องขอการกำหนดการแจ้งเตือนล่วงหน้า โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchDO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

boolean	getAlertType(PersonalSchDO alert) รับการร้องขอการดูประเภทของการแจ้งเตือน โดยจะรับค่าเป็น ออบเจ็กต์ของ PersonalSchDO และ return ค่าเป็น boolean เพื่อบอกว่า ได้กำหนดให้มีการแจ้งเตือนล่วงหน้าหรือไม่
---------	--

ตารางที่ ข.13 แสดงเมธอดในอินเทอร์เฟซ ControlScheduleServer

Interface ControlScheduleServerRMIIInterface

อินเทอร์เฟซ ControlScheduleServerRMIIInterface เป็นอินเทอร์เฟซที่ให้ไคลเอนต์ติดต่อกับเซิร์ฟเวอร์ โดยใช้รูปแบบการติดต่อแบบ rmi โดยเฉพาะ เพราะว่าการติดต่อแบบ rmi จะต้องมีส่วน RemoteException แต่การติดต่อแบบอื่นๆ จะไม่จำเป็นต้องมี RemoteException

Method Summary	
void	setSchedulePersistence(SchedulePersistence schedulePersistence) เป็นการกำหนดรูปแบบการทำงานของระบบการจัดการตารางเวลานัดหมาย
void	setDriverDB(String driverDB) เป็นการกำหนดไดรฟ์เวอร์ของฐานข้อมูลที่ใช้การติดต่อด้วย โดยรับค่าเป็นไดรฟ์เวอร์ของฐานข้อมูลที่ใช้การติดต่อด้วย
void	setDBName(String databaseName) ทำการกำหนดชื่อฐานข้อมูลที่ใช้การติดต่อด้วย โดยรับค่าเป็นชื่อของฐานข้อมูล
void	savePersonalSch(PersonalSchDO person) รับการร้องขอการบันทึกกำหนดการนัดหมายส่วนบุคคลโดยรับค่าเป็นออบเจ็กต์ของ PersonalSchDO
void	updatePersonalSch(PersonalSchDO person, String olddate, String oldtimeSt, String oldtimeFn) รับการร้องขอการแก้ไขรายละเอียดการนัดหมายส่วนบุคคล โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchDO วันที่ เวลาเริ่มต้นและเวลาสิ้นสุดการนัดหมายที่ต้องการแก้ไข
void	removePersonalSchByTime(PersonalSchDO person) รับการร้องขอการลบกำหนดการนัดหมายส่วนบุคคล โดยรับค่าเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ออบเจกต์ของ PersonalSchDO
PersonalSch[]	searchPsByDate(PersonalSchDO person) รับการร้องขอการค้นหาคำหนดการนัดหมายส่วนบุคคลจากเวลา โดยรับค่าเป็นออบเจกต์ของ PersonalSchDO และ return ค่าเป็น array ของ PersonalSch
void	saveAppoint(OrganizeSchDO orgSch) รับการร้องขอการบันทึกกำหนดการนัดหมายภายในองค์กร โดยรับค่าเป็นออบเจกต์ของ OrganizeSchDO
void	removeAppoint(OrganizeSchDO orgSch) รับการร้องขอการลบกำหนดการนัดหมายภายในองค์กร โดยรับค่าเป็นออบเจกต์ของ OrganizeSchDO
OrganizeSch[]	searchAppointByGroup(String workId) รับการร้องขอการค้นหาคำหนดการนัดหมายจากกลุ่มงาน โดยรับค่ารหัสกลุ่มงาน และ return ค่าเป็น array ของ OrganizeSch
String[][]	searchTime(SearchDurationDO search) รับการร้องขอการค้นหาช่วงเวลาที่ จะทำการนัดหมาย โดยรับค่าเป็นออบเจกต์ของ SearchDurationDO และ return ค่าเป็น array ของ String โดยจะบอกถึงช่วงเวลาที่ทำการนัดหมายได้
PersonalSch[]	searchAlertInDay(String id) รับการร้องขอการค้นหาการแจ้งเตือนล่วงหน้าในวันนั้น โดยรับค่ารหัสพนักงานและ return ค่าเป็น array ของ PersonalSchDO
void	setAlertType(PersonalSchDO alert) รับการร้องขอการกำหนดประเภทการแจ้งเตือน โดยรับค่าเป็นออบเจกต์ของ PersonalSchDO
boolean	getAlertType(PersonalSchDO alert) รับการร้องขอการดูประเภทการแจ้งเตือน โดยรับค่าเป็นออบเจกต์ของ PersonalSchDO และ return ค่าเป็น boolean เพื่อแจ้งว่าได้มีการกำหนดให้มีการแจ้งเตือนล่วงหน้าหรือไม่

ตารางที่ ข.14 แสดงเมธอดในอินเทอร์เฟซ ControlScheduleServerRMInterface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface SchedulePersistence

อินเทอร์เฟซ SchedulePersistence เป็นอินเทอร์เฟซของการทำงานของระบบการจัดการตารางเวลานัดหมายที่อยู่ในฝั่งเซิร์ฟเวอร์ โดยจะมีเมธอด ดังนี้

Method Summary	
void	setDriver(String driver) เป็นการกำหนดไดรฟ์เวอร์ของฐานข้อมูลที่ทำกรติดต่อด้วย โดยรับค่าไดรฟ์เวอร์ของฐานข้อมูล
void	setDatabase(String db) เป็นการกำหนดชื่อฐานข้อมูลที่ทำกรติดต่อด้วย โดยรับค่าเป็นชื่อของฐานข้อมูลที่ทำกรติดต่อด้วย
void	insertPsSch(PersonalSchEntity calen) เป็นการเพิ่มกำหนดการนัดหมายส่วนบุคคล โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchEntity
void	updatePsSch(PersonalSchEntity calen,String olddate,String oldtimeSt,String oldtimeFn) เป็นการแก้ไขกำหนดการนัดหมายส่วนบุคคลโดยรับค่าเป็นออบเจ็กต์ของ PersonalSchEntity วันที่เริ่มต้น เวลาเริ่มต้น และเวลาสิ้นสุดของกำหนดการที่ต้องการแก้ไข
void	deletePsSchByTime(PersonalSchEntity calen) เป็นการลบกำหนดการนัดหมายส่วนบุคคลจากเวลา โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchEntity
PersonalSch[]	searchPsSchByDate(PersonalSchEntity person) เป็นการค้นหากำหนดการนัดหมายส่วนบุคคลจากวันที่ โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchEntity และ return ค่าเป็น array ของ PersonalSch
void	addAppoint(OrganizeSchEntity appoint) เป็นการเพิ่มกำหนดการนัดหมายภายในองค์กร โดยรับค่าเป็นออบเจ็กต์ของ OrganizeSchEntity
void	delAppoint(OrganizeSchEntity appoint) เป็นการลบกำหนดการนัดหมายภายในองค์กร โดยรับค่าเป็นออบเจ็กต์ของ OrganizeSchEntity

OrganizeSch[]	searchAppByGroup(String workId) เป็นการค้นหากำหนดการนัดหมายภายในองค์กรจากกลุ่มงานนั้นๆ โดยรับค่าเป็นรหัสของกลุ่มงาน และ return ค่าเป็น array ของ OrganizeSch
String[][]	searchAppoint(SearchDurationEntity dura) เป็นการค้นหาระยะเวลานัดหมายโดยส่งค่าเป็นออบเจ็กต์ของ SearchDurationEntity และ return ค่าเป็น array ของ String ที่บอกถึง ช่วงเวลาที่สามารถทำการนัดหมายได้
PersonalSch[]	searchAlert(String id) เป็นการค้นหาการแจ้งเตือนล่วงหน้าโดยรับค่าเป็นรหัสพนักงานที่ต้องการค้นหา และ return ค่าเป็น array ของ PersonalSchEntity
void	setAlert(PersonalSchEntity alert) เป็นการกำหนดการแจ้งเตือนล่วงหน้า โดยรับค่าเป็น PersonalSchEntity
boolean	getAlert(PersonalSchEntity alert) เป็นการดูการแจ้งเตือนล่วงหน้าว่ามีการกำหนดให้แจ้งเตือนหรือไม่ โดยรับค่าเป็นออบเจ็กต์ของ PersonalSchEntity และ return ค่าเป็น boolean เพื่อบอกว่ากำหนดการนั้นมีการกำหนดให้แจ้งเตือนหรือไม่

ตารางที่ ข.15 แสดงเมธอดในอินเทอร์เฟซ SchedulePersistence

Interface PersonalSch

อินเทอร์เฟซ PersonalSch เป็นอินเทอร์เฟซที่รับ-ส่งรายละเอียดตารางเวลาส่วนบุคคล โดยมีเมธอดดังนี้

Method Summary	
Sting	getEmpId() รื่องขอรหัสพนักงาน
String	getdateSt() รื่องขอวันที่เริ่มต้นการนัดหมาย
String	getdateFn() รื่องขอวันที่สิ้นสุดการนัดหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String	getTimeSt() ร้องขอเวลาเริ่มต้นทำการนัดหมาย
String	getTimeFn() ร้องขอเวลาสิ้นสุดทำการนัดหมาย
String	getMessage() ร้องขอรายละเอียดของการนัดหมาย
String	getIntervalTime() ร้องขอรูปแบบของการนัดหมายว่าเป็นช่วงเวลา หรือทั้งวัน
boolean	getAlert() ร้องขอว่ามีการแจ้งเตือนล่วงหน้าหรือไม่
String	getDateAlert() ร้องขอวันที่ทำการแจ้งเตือนล่วงหน้า
String	getTimeAlert() ร้องขอเวลาที่ทำการแจ้งเตือนล่วงหน้า

ตารางที่ ข.16 แสดงเมธอดของอินเทอร์เฟซ PersonalSch

Interface OrganizeSch

อินเทอร์เฟซ OrganizeSch เป็นอินเทอร์เฟซที่รับ-ส่งรายละเอียดตารางเวลานัดหมายภายในองค์กร โดยมีเมธอดดังนี้

Method Summary	
String	getAppId() ร้องขอรหัสของการนัดหมายภายในองค์กร
String	getSubject() ร้องขอหัวข้อการนัดหมายภายในองค์กร
String	getDate() ร้องขอวันที่ทำการนัดหมายภายในองค์กร
String	getTimeSt() ร้องขอเวลาเริ่มต้นที่ทำการนัดหมายภายในองค์กร
String	getTimeFn() ร้องขอเวลาสิ้นสุดที่ทำการนัดหมายภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String	getMessage() ร้องขอรายละเอียดที่ทำการนัดหมายในองค์กร
String	getWorkId() ร้องขอรหัสของกลุ่มงานที่ทำการนัดหมาย
String[]	getEmpIdList() ร้องขอรหัสพนักงานในองค์กรที่ทำการนัดหมาย

ตารางที่ ข.17 แสดงเมธอดของอินเทอร์เฟซ OrganizeSch

Interface SearchDuration

อินเทอร์เฟซ SearchDuration เป็นอินเทอร์เฟซที่ใช้ในการค้นหาเวลาในการนัดหมาย โดยมีเมธอดดังนี้

Method Summary	
String	getWorkId() ร้องขอรหัสของกลุ่มงานที่ทำการนัดหมาย
String	getDuration() ร้องขอช่วงเวลาทำการนัดหมาย
String	getdateSt() ร้องขอวันที่เริ่มต้นที่ต้องการนัดหมาย
String	getdateFn() ร้องขอวันที่สิ้นสุดที่ต้องการนัดหมาย

ตารางที่ ข.18 แสดงเมธอดของอินเทอร์เฟซ SearchDuration