



การพัฒนาแอปพลิเคชันแบบใช้ภาษาจาวาที่ใช้นอินเทอร์เน็ตโดยใช้ภาษาจาวา
GroupWare Application Development on WWW by JAVA language



โดย
นายกาญจน์ สุขวงศ์
นายบุญชัย รัตนวรานันท์

วัน เดือน ปี.....	15.ค.ค.2541
เลขทะเบียน.....	038969
เลขเรียกหนังสือ.....	T.20210 ก.221 ก.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง **038969**

การพัฒนาแอปพลิเคชันแบบใช้งานเป็นกลุ่มที่ใช้บนอินเทอร์เน็ตโดยใช้ภาษาจาวา
GroupWare Application Development on WWW by JAVA language



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2540

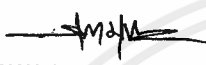
ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาแอปพลิเคชันแบบใช้งานเป็นกลุ่มที่ใช้บนอินเทอร์เน็ตโดยใช้ภาษาจาวา

ผู้จัดทำ

1. นายกาญจน์ สุขวงศ์
2. นายบุญชัย รัตนวรานันท์


.....อาจารย์ที่ปรึกษา
(อาจารย์ธนา หงษ์สุวรรณ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการพัฒนาแอปพลิเคชันแบบใช้งานเป็นกลุ่มที่ใช้บนอินเทอร์เน็ตโดยใช้ภาษาจาวา

นายกาญจน์	สุขวงศ์
นายบุญชัย	รัตนวานันท์
อาจารย์ธนา	หงษ์สุวรรณ
ปีการศึกษา	2540

บทคัดย่อ

ในปัจจุบันเป็นที่ยอมรับกันอย่างกว้างขวางในภาครัฐและเอกชนว่า ระบบคอมพิวเตอร์และระบบการสื่อสารข้อมูลนั้น มีความสำคัญและมีความจำเป็นต่อการพัฒนาเพื่อเพิ่มประสิทธิภาพในการทำงานทุกระบบงาน ไม่ว่าจะเป็นงานทำประจำหรืองานส่งเสริมการบริหารของหน่วยงานนั้นๆ และในการที่หน่วยงานจะประสบความสำเร็จในการดำเนินงานนั้นๆ จะต้องมีเครื่องมือที่ดีและทรัพยากรต่างๆ เพียงพอ ทรัพยากรมนุษย์จัดเป็นทรัพยากรอย่างหนึ่งที่สำคัญต่อหน่วยงานและเป็นทรัพยากรที่ขาดการควบคุม ไม่ว่าหน่วยงานนั้นจะมีทรัพยากรอื่นๆ เพียงพร้อมเพียงใดก็ตาม ถ้าการบริหารทรัพยากรมนุษย์เป็นไปอย่างไม่มีประสิทธิภาพแล้วจะส่งผลกระทบต่อผลการดำเนินงานโดยรวมได้ อีกทั้งในปัจจุบันนี้ระบบอินเทอร์เน็ตได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ทั้งด้านการติดต่อสื่อสารภายในองค์กรและระหว่างองค์กร

ดังนั้นโครงการพัฒนาแอปพลิเคชันแบบใช้งานเป็นกลุ่มที่ใช้บนอินเทอร์เน็ตโดยใช้ภาษาจาวาจึงได้ถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกในการติดต่อสื่อสารเกี่ยวกับเรื่องต่างๆ ดังนี้

ระบบทะเบียนรุ่น พัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกในด้านการแก้ไขปรับปรุงข้อมูล คือ ผู้ใช้สามารถทำการแก้ไขปรับปรุงข้อมูลของตนเองในทะเบียนรุ่นได้ เป็นต้น

ระบบการตอบคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต ช่วยอำนวยความสะดวกในด้านการแนะแนวและการส่งการบ้านผ่านระบบอินเทอร์เน็ตได้ เป็นต้น

ระบบการนัดหมายการประชุม มีประโยชน์สำหรับอาจารย์และนักศึกษา รวมถึงผู้ที่จบการศึกษาไปแล้วในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ คือ ช่วยอำนวยความสะดวกในการนัดหมายการประชุม เป็นต้น

โครงการพัฒนาแอปพลิเคชันแบบใช้งานเป็นกลุ่มที่ใช้บนอินเทอร์เน็ตโดยใช้ภาษาจาวานี้ใช้เครื่องมือในการออกแบบและการสร้าง คือ Visual Cafe' และจัดการเกี่ยวกับระบบฐานข้อมูลโดยใช้ MS. Access โดยออกแบบฐานข้อมูลโดยใช้ ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GroupWare Application Development On WWW by JAVA Language

Karn Sukwong
Boonchai Rattanawaranun

Thana Hongsuwan
1997

Abstract

It is widely accepted that Human resource is one of the most important resources in an organization and it is typically difficult to control. Also, because the cost concerning human resource in an organization is considerably high it is necessary to have a computer-based information system, the Load of Lecturer Management System, in order to manage the human resource efficiently. And now Internet System is coming to popular or be in life cycle more than in past almost in company and between companies.

Therefore GroupWare Application Development on WWW by JAVA Language was produced for increasing information facilities about these.

Batch Residence was developed for data facilities and updating facilities that the users can update their information in Batch Residence by themselves by using Internet and others.

Lecturer Discussion is comfortable for suggest and asking for homework by using Internet and others.

Meeting Scheduler is available for lecturers, student and graduated student of Computer Engineer in KMIT'L that is available for meeting schedule and others

กิตติกรรมประกาศ

โครงการนี้ได้รับการส่งเสริมและสนับสนุนจากหลายฝ่ายซึ่งถ้าไม่มีหน่วยงาน หรือบุคคลดังกล่าวนี้ ผลงานวิจัยนี้คงไม่ประสบผลสำเร็จได้ ดังนั้นจึงใคร่ขอขอบพระคุณ

1. บิดา มารดา เป็นผู้ที่มีพระคุณมาก ได้ให้กำเนิดและเลี้ยงดู ส่งเสริมให้ได้รับ และให้กระทำในสิ่งที่ดีมาโดยตลอด เป็นผู้ให้กำลังใจในด้านต่างๆมากมายซึ่งรวมถึงการศึกษาเล่าเรียนเพื่อให้ความรู้ และช่วยสนับสนุนในสิ่งที่ไม่เป็นในทุกๆด้าน
2. อาจารย์ธนา หงษ์สุวรรณ เป็นอาจารย์ที่ปรึกษาปริญญาโทที่กรุณาให้คำปรึกษา แนะนำในด้านการศึกษาปัญหา และออกแบบระบบงานและแนวทางในการแก้ปัญหาต่างๆที่เกิดขึ้นในขณะทำการพัฒนาระบบงาน ตลอดจนสนับสนุนอุปกรณ์ต่างๆที่ใช้ในการพัฒนาระบบงานนี้
3. เพื่อนๆทุกท่านที่ช่วยสนับสนุนการพัฒนาระบบงาน และช่วยเหลืองานด้านเอกสาร
4. คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้เครื่องคอมพิวเตอร์ดีๆได้ใช้ในการพัฒนาระบบ
5. ตัวผู้จัดทำเองที่มีพลังและความพยายามในการพัฒนาระบบจนจบจนกระทั่งสำเร็จดังที่หวังไว้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	IX
สารบัญภาพ	X
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของการพัฒนาระบบงาน	1
1.3 ทฤษฎีที่ใช้ในการพัฒนาระบบงาน	2
1.4 เป้าหมายของการพัฒนาระบบงาน	2
1.5 ขอบเขตของการพัฒนาระบบงาน	2
1.6 เครื่องมือที่ใช้ในการพัฒนาระบบงาน	3
บทที่ 2 ทฤษฎีที่ใช้ในการทำโครงงาน	4
2.1 ความรู้ทั่วไปเกี่ยวกับภาษาจาวา	4
2.1.1 คุณสมบัติของจาวา	4
2.1.2 การทำงานของจาวา	5
2.1.3 การเขียนโปรแกรมภาษาจาวา	6
2.1.3.1 ตัวแปรและชนิดของข้อมูล	6
2.1.3.2 ตัวดำเนินการ	8
2.1.3.3 สเตทเมนต์ที่ใช้ในการควบคุมโปรแกรม	11
2.1.3.4 อารเรย์และสตริง	14
2.1.3.5 ออบเจกต์และคลาส	15
2.1.3.6 Handling Errors Using Exception	24
2.1.3.7 Thread of Control	27
2.1.4 ข้อแตกต่างระหว่างแอปพลิเคชันกับแอปเพล็ต	32
2.1.5 การเขียนแอปเพล็ต	32
2.1.5.1 วงจรชีวิตของแอปเพล็ต	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.2	เมธอดที่ใช้สำหรับตรวจสอบ	34
2.1.5.3	เมธอดสำหรับ Drawing และ Event Handling	34
2.1.5.4	เมธอดที่ใช้ในการเพิ่ม UI Component	34
2.1.5.5	ข้อจำกัดและความสามารถของแอพเพล็ต	35
2.1.5.6	การเพิ่มแอพเพล็ตเข้าไปใน HTML	36
2.1.5.7	การสิ้นสุดการทำแอพเพล็ต	37
2.2	JDBC (Java Database Connectivity)	39
2.2.1	JDBC คืออะไร	39
2.2.2	เลขอร์ของ JDBC	39
2.2.3	การทำงานของ JDBC	40
2.2.3.1	การเชื่อมต่อไปยังฐานข้อมูล	40
2.2.3.2	SQL Statement	41
2.2.3.3	กระบวนการทำ Result Set	41
2.2.3.4	การ Update Transaction	43
2.2.3.5	การเรียกกระบวนการเก็บ	43
2.2.3.6	การเข้าถึง Metadata	45
2.2.4	เปรียบเทียบ CGI และ JDBC	45
2.2.5	การพิจารณาการออกแบบ Client-Server	47
2.2.6	ชนิดของ JDBC Driver	49
2.3	ความรู้และทฤษฎีพื้นฐาน โมเดลเชิงสัมพันธ์ (Relational Model)	54
บทที่ 3	ทะเบียนรุ่น	64
3.1	วัตถุประสงค์ของระบบทะเบียนรุ่น	64
3.2	ลักษณะของระบบทะเบียนรุ่น	64
3.3	รายละเอียดของระบบทะเบียนรุ่น	64
3.4	แผนผังลำดับขั้นตอนการดำเนินงานของระบบทะเบียนรุ่น	65
บทที่ 4	ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต	67
4.1	วัตถุประสงค์ของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต	67
4.2	ลักษณะของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต	67
4.3	รายละเอียดของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต	67
4.4	แผนผังลำดับขั้นตอนของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต	69
บทที่ 5	ระบบการจัดการนัดหมายการประชุม	71
5.1	วัตถุประสงค์ของระบบการจัดการนัดหมายการประชุม	71
5.2	ลักษณะของระบบการจัดการนัดหมายการประชุม	71
5.3	รายละเอียดของระบบการจัดการนัดหมายการประชุม	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ส่วนประกอบหลักของระบบการจัดการนัดหมายการประชุม	72
บทที่ 6 การพัฒนาโครงการ	73
6.1 ส่วนประกอบของโครงการ	73
6.2 รูปแบบการทำงาน	73
6.2.1 ระบบทะเบียนรุ่น	75
6.2.2 ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต	77
6.2.3 ระบบการจัดการนัดหมายการประชุม	86
6.2.4 การเปลี่ยนพาสเวิร์ดของผู้ใช้	99
6.3 รูปแบบฐานข้อมูล	101
บทที่ 7 ผลการทดลอง	111
7.1 การทดสอบการทำงานในส่วน of ระบบการจัดการนัดหมายการประชุม	111
7.2 การทดสอบการทำงานในส่วน of ทะเบียนรุ่น	111
7.3 การทดสอบการทำงานในส่วน of ระบบการถามคำถามของอาจารย์	112
บทที่ 8 สรุปผลการพัฒนาโครงการ	114
8.1 ผลการพัฒนาโครงการ	114
8.2 การติดตั้งใช้งาน	115
8.3 คุณสมบัติและประโยชน์ของโปรแกรม	115
8.4 ข้อเสนอแนะ	115
ภาคผนวก ก. คู่มือการโปรแกรม	116
ภาคผนวก ข. คู่มือการใช้งาน	141
บรรณานุกรม	169

สารบัญตาราง

ชื่อตาราง		หน้า
ตารางที่ 2-1	ชนิดของตัวแปรในจาวา	7
ตารางที่ 2-2	คีย์เวิร์ดในจาวา	7
ตารางที่ 2-3	binary arithmetic operation ของจาวา	9
ตารางที่ 2-4	เครื่องหมายที่ใช้บอกค่าของ operand	9
ตารางที่ 2-5	การใช้ prefix และ postfix ในการเพิ่มและลดค่า	9
ตารางที่ 2-6	relational operator ในจาวา	10
ตารางที่ 2-7	condition operator ในจาวา	10
ตารางที่ 2-8	Bitwise operator ในจาวา	10
ตารางที่ 2-9	assignment operator เพิ่มเติม	11
ตารางที่ 2-10	ประเภทของสเตทเมนต์ที่ใช้ควบคุมการทำงาน	11
ตารางที่ 2-11	ระดับการเข้าถึงของแต่ละ specified	19
ตารางที่ 2-12	ข้อแตกต่างระหว่างแอฟพลิเคชันกับแอฟเพล็ท	32

สารบัญรูปภาพ

รูปภาพที่	หน้า
รูปที่ 2-1 การทำงานของจาวา	6
รูปที่ 2-2 superclass และ subclass	22
รูปที่ 2-3 class hierarchy	22
รูปที่ 2-4 ตัวอย่างของ exception hierarchy	24
รูปที่ 2-5 class hierarchy ของ Throwable class	27
รูปที่ 2-6 สถานะของ Thread ในจาวา	28
รูปที่ 2-7 Thread Group class	31
รูปที่ 2-8 hierarchy ของ Applet class	33
รูปที่ 2-9 แสดงโมเดลชั้นการทำงานของเจดีบีซี	39
รูปที่ 2-10 แสดงรูปแบบที่ใช้ใน JDBC	40
รูปที่ 2-11 การติดต่อไปยังฐานข้อมูล โดยใช้ CGI	46
รูปที่ 2-12 การติดต่อไปยังฐานข้อมูลโดยผ่าน JDBC	47
รูปที่ 2-13 แสดง Two-Tier model	48
รูปที่ 2-14 แสดง Three-Tier mode	48
รูปที่ 2-15 แสดง JDBC-ODBC Bridge	50
รูปที่ 2-16 แสดง Native-API (Partly-Java) Drivers	51
รูปที่ 2-17 แสดง Net-Protocol (All-Java) Drivers	52
รูปที่ 2-18 แสดง Native-Protocol (All-Java) Drivers	53
รูปที่ 2-19 แสดงประเภทลักษณะสัญลักษณ์ของค้ำไฟล์ไดอะแกรม(DFD)	61
รูปที่ 3-1 การเข้าสู่ข้อมูลระบบทะเบียนรุ่น	65
รูปที่ 3-2 การปรับปรุงแก้ไขเพิ่มเติมข้อมูลในระบบทะเบียนรุ่น	66
รูปที่ 4-1 การตั้งหัวข้อคำถามและคำถามของอาจารย์	69
รูปที่ 4-2 การตอบคำถามในระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต	70
รูปที่ 6-1 หน้าจอหลักของระบบทะเบียนรุ่น (Com- Member Info)	74
รูปที่ 6-2 หน้าจอแสดงข้อความเตือน (Warning Message)	75
รูปที่ 6-3 หน้าจอแสดงข้อมูลรายละเอียดของบุคคล (Details Form)	75
รูปที่ 6-4 หน้าจอการตรวจสอบ Security ในการ Edit (Edit Security)	76
รูปที่ 6-5 หน้าจอการแก้ไขข้อมูลรายละเอียดของบุคคล (Edit Form)	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6-6	หน้าจอการ Login เข้าใช้ระบบ (User Security Form)	77
รูปที่ 6-7	หน้าจอหลักของระบบ Lecturer Discussion (Main Form)	78
รูปที่ 6-8	หน้าจอการปรับปรุงแก้ไขหัวข้อคำถาม (Forum Update Form)	79
รูปที่ 6-9	หน้าจอการปรับปรุงแก้ไขคำถาม (Question Update Form)	79
รูปที่ 6-10	หน้าจอแสดงข้อความเตือน (Warning Message)	80
รูปที่ 6-11	หน้าจอแสดงข้อความเตือน (Warning Message)	80
รูปที่ 6-12	หน้าจอแสดงข้อความเตือน (Warning Message)	80
รูปที่ 6-13	หน้าจอ Menu การ Insert (Insert Menu Form)	81
รูปที่ 6-14	หน้าจอการ Insert หัวข้อคำถาม (Forum Insertion Form)	81
รูปที่ 6-15	หน้าจอการเพิ่มคำถาม (Question Insertion Form)	82
รูปที่ 6-16	หน้าจอแสดงข้อความเตือน (Warning Message)	82
รูปที่ 6-17	หน้าจอถามความแน่ใจในการ Delete	83
รูปที่ 6-18	หน้าจอแสดงข้อความเตือน (Warning Message)	83
รูปที่ 6-19	หน้าจอแสดงรายละเอียดของคำถาม (Question Details Form)	84
รูปที่ 6-20	หน้าจอแสดงข้อความเตือน (Warning Message)	84
รูปที่ 6-21	หน้าจอการเพิ่มคำตอบ (Reply Form)	85
รูปที่ 6-22	หน้าจอการสอบถามสิทธิในการใช้งาน	86
รูปที่ 6-23	แสดงทูลบาร์หลักของโปรแกรมการนัดหมายการประชุม	86
รูปที่ 6-24	หน้าจอแรกที่เข้าสู่โปรแกรมการนัดหมายการประชุม	87
รูปที่ 6-25	หน้าจอแสดงรายละเอียดของการนัดหมายการประชุม (Detail of meeting)	89
รูปที่ 6-26	หน้าจอแสดงการนัดหมายการประชุมทั้งหมดที่ผู้ใช้ได้รับมา	90
รูปที่ 6-27	หน้าจอแสดงผลของการตัดสินใจของแต่ละบุคคล	91
รูปที่ 6-28	หน้าจอแสดงรายรายละเอียดในการนัดหมายการประชุม	92
รูปที่ 6-29	หน้าจอแสดงรูปแบบในการสร้างการนัดหมายการประชุมขึ้นมาใหม่	93
รูปที่ 6-30	หน้าจอแสดงโครงสร้างต้นไม้ที่เก็บรายชื่อ	94
รูปที่ 6-31	หน้าจอแสดงการเลือกสมาชิกกลุ่ม	95
รูปที่ 6-32	แสดงผลจากการคลิกที่ปุ่ม OK ในรูปที่ 6-31	95
รูปที่ 6-33	แสดงหน้าจอเตือนเมื่อมีการซ้ำซ้อนในเรื่องเวลา	96
รูปที่ 6-34	หน้าจอแสดงกลุ่มที่ผู้ใช้ได้ตั้งไว้แล้ว	97
รูปที่ 6-35	หน้าจอในการจัดตั้งกลุ่มใหม่	98
รูปที่ 6-36	หน้าจอในการเปลี่ยน Password	99
รูปที่ 6-37	หน้าจอยืนยัน Password ใหม่	99
รูปที่ 6-38	หน้าจอแสดงข้อความเตือน (Warning Message)	99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6-39 หน้าจอแสดงข้อความเตือน	100
รูปที่ 6-40 แสดง ER Diagram ของระบบทะเบียนรุ่น	102
รูปที่ 6-41 แสดง ER Diagram ของระบบการถามคำถามของอาจารย์ผ่านอินเทอร์เน็ต	105
รูปที่ 6-40 แสดง ER Diagram ของระบบการนัดหมายการประชุม	108



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในปัจจุบันนี้ แนวความคิดที่จะนำระบบคอมพิวเตอร์เข้ามาช่วยลดความยุ่งยากในการติดต่อสื่อสารภายในองค์กรหรือบริษัทมีมากขึ้น อีกทั้งระบบอินเทอร์เน็ตกำลังเป็นที่นิยมและมีความสะดวกในการติดต่อสื่อสารมาก ทางคณะผู้จัดทำจึงได้พัฒนาโครงการพัฒนากรู๊ปแวร์แอปพลิเคชันบนเว็ลด์ไวด์เว็บ โดยใช้ภาษาจาวา ซึ่งเป็นโปรแกรมที่มีความสามารถในการใช้งานได้แบบเป็นกลุ่ม เพื่อใช้ภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ และให้มีความสามารถในการทำงานบนระบบอินเทอร์เน็ตได้ด้วยโดยใช้ภาษาจาวาในการสร้าง โปรแกรมดังกล่าว ดังนั้น โปรแกรมที่ได้จึงสามารถเรียกใช้ผ่านระบบอินเทอร์เน็ตได้โดยตรงโดย

1.2 วัตถุประสงค์ของการพัฒนาระบบงาน

1.2.1 เป็นการส่งเสริมให้มีการนำเทคโนโลยีทางคอมพิวเตอร์และการสื่อสาร เข้ามาใช้ในองค์กรของรัฐ โดยมีกลุ่มเป้าหมายเป็นมหาวิทยาลัยของรัฐบาลเพื่อสนับสนุนการบริหารงานซึ่งสอดคล้องกับนโยบายรัฐบาล

1.2.2 เพื่อจัดการขั้นตอนการทำงานเกี่ยวกับระบบทะเบียนรุ่น การถามคำถามของอาจารย์โดยผ่านทางอินเทอร์เน็ต การนัดหมายการประชุม ให้สามารถทำงานบนระบบอินเทอร์เน็ตได้

1.2.3 เพื่อสนับสนุนและเพิ่มประสิทธิภาพในการติดต่อสื่อสาร ในภาควิชา

1.2.4 เพื่อช่วยเพิ่มประสิทธิภาพในการทำงานโดยรวมสูงขึ้น

1.3 ทฤษฎีที่ใช้ในการพัฒนาระบบงาน

โครงการพัฒนากรู๊ปแวร์แอปพลิเคชันบนเว็ลด์ไวด์เว็บโดยใช้ภาษาจาวานี้ ได้ถูกพัฒนาด้วย Visual Cafe' ภายใต้ระบบปฏิบัติการ Microsoft Windows NT Server โดยระบบงานนี้จะใช้วิธีแผนภาพแสดงข้อมูล (DFD : Data Flow Diagram) ในการพิจารณาความสัมพันธ์ในรายละเอียดการทำงานและใช้วิธี อีอาร์โมเดล (E-R diagram : Entity Relationship Model) เพื่อออกแบบฐานข้อมูล ซึ่งผลที่ได้จะวิเคราะห์จะได้ตารางที่ประกอบด้วยแอททริบิว (attribute) ต่างๆและตารางต่างๆที่ได้จากการออกแบบนี้จะมีคุณสมบัติครบตามบรรทัดฐานที่สาม (3NF : Third Normal Form) แล้วทำการนอร์มอลไลต์ (Normalization) ตามจนถึง บรรทัดฐานที่ห้า (5NF : Fifth Normal Form)

1.4 เป้าหมายของการพัฒนาระบบงาน

1.4.1 ระบบทะเบียนรุ่น

ต้องการให้มีการจัดเก็บข้อมูลและรายละเอียดต่างๆ เช่น ที่อยู่ที่สามารถติดต่อได้ ของอาจารย์และนักศึกษาภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ที่จบการศึกษาไปแล้ว เพื่อให้ผู้ที่สนใจในข้อมูลดังกล่าวสามารถขอข้อมูลผ่านระบบอินเทอร์เน็ตได้

1.4.2 ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

ต้องการให้อาจารย์สามารถส่งคำถามผ่านทางระบบอินเทอร์เน็ตได้ โดยที่ใครๆก็สามารถทำการตอบคำถามนั้นๆผ่านทางระบบอินเทอร์เน็ตได้เช่นเดียวกัน

1.4.3 ระบบการจัดการนัดหมายการประชุม

ต้องการให้อาจารย์และนักศึกษาภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สามารถทำการนัดหมายการประชุมหรือนัดพบปะสังสรรค์ผ่านทางระบบอินเทอร์เน็ตได้

1.5 ขอบเขตของการพัฒนาระบบ

โครงการพัฒนากรุปแวร์แอปพลิเคชันบนเว็ลด์ไวด์เว็บโดยใช้ภาษาจาวา นี้ จะมีการทำงานครอบคลุมส่วนต่างๆที่สำคัญดังนี้ คือ

1.5.1 ระบบทะเบียนรุ่น

1.5.1.1 การเข้าดูรายละเอียดของอาจารย์และนักศึกษาภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ที่จบการศึกษาไปแล้ว

1.5.1.2 การแก้ไขปรับปรุงเพิ่มเติมรายละเอียดของอาจารย์และนักศึกษาภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ที่จบการศึกษาไปแล้ว

1.5.2 ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

1.5.2.1 การเข้าดูรายละเอียดต่างๆของคำถามของอาจารย์

1.5.2.2 การตอบคำถามของอาจารย์

1.5.2.3 การเพิ่มหัวข้อคำถามและตัวคำถามของอาจารย์

1.5.3 ระบบการจัดการนัดหมายการประชุม

1.5.3.1 การสร้างและยกเลิกกลุ่มที่ใช้ในการนัดหมายการประชุม

1.5.3.2 การนัดหมายการประชุมและการยกเลิกการนัดหมาย

1.5.3.3 การแสดงข้อความเตือนวันเวลาที่นัดหมายการประชุม

1.6 เครื่องมือที่ใช้ในการพัฒนาระบบงาน

1.6.1 รายละเอียดทางด้านฮาร์ดแวร์

1. เนื้อที่ฮาร์ดดิสก์มากกว่า 100 Mb สำหรับ การติดตั้งโปรแกรมทั้งหมด
2. เครื่อง Pentium
3. แรม 16 Mb (ควรจะใช้ 32 Mb หรือมากกว่านั้น)

1.6.2 รายละเอียดทางด้านซอฟต์แวร์

1. MS Windows 95
2. MS Window NT Server
3. IDS Server
4. Visual Café PDE
5. Microsoft Access 7.0



บทที่ 2

ทฤษฎีที่ใช้ในการทำโครงการงาน

2.1 ความรู้ทั่วไปเกี่ยวกับภาษาจาวา

ในหัวข้อนี้จะกล่าวถึงภาษาจาวาโดยทั่วไป ทั้งคุณสมบัติ การทำงาน องค์กรประกอบที่จำเป็นเพื่อการเขียนโปรแกรม รวมถึงการพัฒนาภาษาจาวาในรูปแบบต่างๆ

2.1.1 คุณสมบัติของจาวา

จาวาเป็นภาษาที่ใช้ในการเขียนโปรแกรมแบบหนึ่ง ซึ่งถูกพัฒนาโดยบริษัท Sun Microsystems จาวาถูกพัฒนาเพื่อใช้ในการพัฒนาโปรแกรมสำเร็จรูปทางสภาพแวดล้อมที่ต่างกัน และในสภาพแวดล้อมที่มีการกระจายของโปรแกรมไปในเครือข่าย โดยมีลักษณะหรือพื้นฐานมาจากภาษา C++ คุณสมบัติของภาษาจาวาคือ

2.1.1.1 ง่าย

ลักษณะอย่างแรกของภาษาจาวาก็คือเป็นภาษาที่เข้าใจได้ง่ายเนื่องจากจาวาจะคล้ายกับภาษา C++ แต่ได้กำจัดส่วนที่เป็นปัญหาออก ทำให้จาวาเป็นภาษาที่ง่าย สามารถเรียนรู้ได้ง่าย

2.1.1.2 เป็น Object-Oriented

ทำให้คลาสของจาวาอยู่ในรูปของออบเจกต์ ทำให้สามารถเคลื่อนย้ายระหว่างคอมพิวเตอร์เครื่องที่ให้บริการ (host computer) และ ตัวแปลภาษาจาวา (java interpreter) ผ่านเครือข่ายได้สะดวก

2.1.1.3 มีสภาพแวดล้อมที่กระจัดกระจาย

โดยมีการให้ไลบรารีของคลาสสำหรับการเข้าถึงอินเทอร์เน็ต คุณสามารถใช้ออบเจกต์จากที่ไหนก็ได้ในเครือข่ายมาทำโปรแกรมสำเร็จรูปของคุณหรือสร้างออบเจกต์ซึ่งสามารถถูกเข้าถึงหรือถูกจำลองโดยผู้พัฒนาโปรแกรมคนอื่น ซึ่งจะช่วยให้คุณแน่ใจได้ว่าคุณได้ใช้คลาสที่จะลงมาที่ใหม่ที่สุด

2.1.1.4 ตัวแปลภาษา

เริ่มแรกคอมไพเลอร์ (compiler) จะอ่านซอร์ซโค้ดและแปลงเป็น ไบต์โค้ด (อยู่ระหว่าง ซอร์ซโค้ด กับ ไบต์โค้ด) ซึ่งจะถูกส่งไปดำเนินการต่อที่ตัวแปลภาษาในเครื่องใดๆที่ตัวดำเนินการและระบบ runtime ตั้งอยู่ ตัวแปลภาษาจะอ่าน ไบต์โค้ดและจะดำเนินการคำสั่งภาษาเครื่องที่เหมาะสม ซึ่งเป็นการช่วยแก้ปัญหาในสภาพแวดล้อมที่มีหลายแพลตฟอร์ม

2.1.1.5 มีประสิทธิภาพสูง

การที่ตัวแปลภาษาสามารถรันด้วยความเร็วเต็มที่ โดยไม่ต้องทำการตรวจสอบสภาพแวดล้อมในขณะเวลารัน เนื่องจากมี Java Virtual Machine อยู่ และการกำจัดข้อมูลที่ไม่ความหมายโดยอัตโนมัติ (automatic garbage collector) ซึ่งจะรันเสมือนเป็นเทรคอยู่เบื้องหลัง เพื่อทำให้มีหน่วยความจำใช้งานได้เมื่อต้องการ ทำให้ประสิทธิภาพโดยรวมในการทำงานสูงขึ้น

2.1.1.6 มีความแข็งแรงและน่าเชื่อถือ

การทำ dynamic linking มีผล 2 ประการคือ การคอมไพล์จะทำการตรวจสอบการอ้างอิงไปถึงคลาสอื่นทุกตัวเพื่อความถูกต้องของชนิด และโปรแกรมจะถูกเชื่อมกับคลาสที่ใหม่ที่สุดที่โปรแกรมใช้อยู่ คุณไม่จำเป็นต้องทำการคอมไพล์ซ้ำๆเมื่อคลาสมีการเปลี่ยนแปลง และในขณะที่การติดต่อ (interface) ของคลาสเหล่านั้นเหมือนเดิม คุณสามารถใช้คลาสเหล่านั้นได้เหมือนเดิมโดยไม่ต้องเปลี่ยนโค้ดของคุณ นอกจากนั้นจาวายังได้ยกเลิกการใช้พอยเตอร์ ซึ่งทำให้ง่ายต่อการจัดการหน่วยความจำทำให้มีความน่าเชื่อถือมากยิ่งขึ้น

2.1.1.7 มีความปลอดภัย

เพื่อความปลอดภัยในจาวาจะไม่มีเมธอดใดที่ใช้ทำให้เกิดอันตรายหรือเชื่อมไปยังโค้ดที่คอมไพล์แล้วเพื่อทำการที่ทำให้เกิดอันตรายได้ จาวาจะทำการทดสอบไฟล์สำหรับการอนุญาตให้เข้าถึงก่อนที่จะอนุญาตให้โปรแกรมบีตไฟล์นั้นๆ ตัวแปลภาษาจะทำการตรวจทานไบต์โค้ดเพื่อตรวจจับการโน้มน้าว เช่น คอมไพล์เลอร์ที่บางคนเขียนให้ใช้พอยเตอร์ได้ และจากการทำการแมปหน่วยความจำในขณะรัน ทำให้ป้องกันแฮ็กเกอร์เข้าถึงโครงสร้างของข้อมูลได้

2.1.1.8 มีสถาปัตยกรรมที่เป็นกลาง

เนื่องจากจาวาถูกออกแบบมาเพื่อสนับสนุนโปรแกรมสำเร็จรูป ซึ่งจะถูกระบายไปยังสภาพแวดล้อมของเครือข่ายที่แตกต่างกัน ในสภาพแวดล้อมเหล่านั้นโปรแกรมจะต้องสามารถดำเนินการบนสถาปัตยกรรมของฮาร์ดแวร์ที่แตกต่างกัน ระบบปฏิบัติการที่ต่างกัน ฯลฯ ได้ เพื่อให้สามารถทำงานในสภาพแวดล้อมที่มีการจัดการที่หลากหลาย คอมไพล์เลอร์ของจาวาจะสร้างไบต์โค้ด (เป็นรูปแบบของสถาปัตยกรรมที่เป็นกลาง) ออกแบบเพื่อให้สามารถส่งโค้ดได้อย่างมีประสิทธิภาพไปยังฮาร์ดแวร์และซอฟต์แวร์ที่มีรูปแบบหลากหลาย ซึ่งการแปล (interpret) จาวาจะช่วยแก้ปัญหา การกระจายไบนารี และปัญหาเรื่องเวอร์ชัน

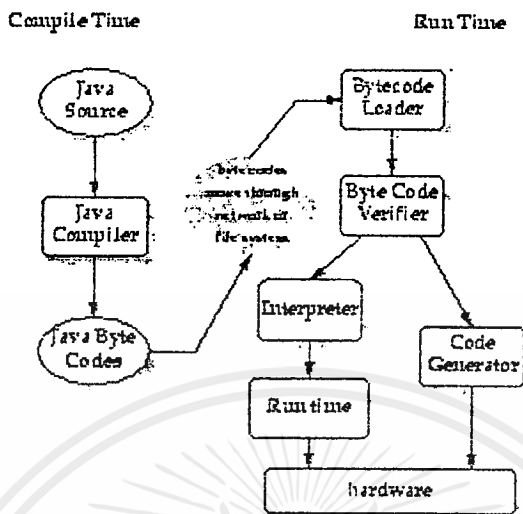
2.1.1.9 สามารถเคลื่อนย้ายได้

การสถาปัตยกรรมที่เป็นกลางก็เป็นส่วนหนึ่งของระบบที่สามารถเคลื่อนย้ายได้ จาวาใช้ความสามารถในการเคลื่อนย้ายได้ในการกำหนดขอบเขตของภาษาพื้นฐาน โดยกำหนดขนาดชนิดของข้อมูลพื้นฐานและพฤติกรรมของตัวดำเนินการทางคณิตศาสตร์ ทำให้โปรแกรมของคุณเหมือนกันในทุกๆแพลตฟอร์ม ซึ่งจะทำให้ชนิดของข้อมูลทุกตัวสามารถเข้ากับสถาปัตยกรรมของฮาร์ดแวร์และซอฟต์แวร์ที่ต่างกันได้

2.1.2 การทำงานของจาวา

เมื่อเราทำการสร้างไฟล์จาวาขึ้นมาได้แล้ว เราต้องทำให้ไฟล์นี้สามารถใช้งานได้ โดยการนำไฟล์จาวาที่สร้างขึ้นมาไปทำการคอมไพล์ หลังจากคอมไพล์แล้วจะไปได้ไฟล์ที่เรียกว่า คลาสไฟล์ ซึ่งเป็นไฟล์ที่เก็บไบต์โค้ดไว้ (ไบต์โค้ดจะมีลักษณะใกล้เคียงคำสั่งเครื่องมากที่สุด) โดยกระบวนการที่กล่าวมาทั้งหมดจะกระทำที่เครื่องรับบริการ ไบต์โค้ดที่ได้นี้จะถูกรันบน *Java Virtual Machine (JVM)* เป็นสิ่งที่สมมุติขึ้นซึ่งพฤติกรรมได้ถูกกำหนดมาก่อนแล้วโดยบริษัท Sun Microsystems ลักษณะของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JVM ได้มีการอธิบายถึงพฤติกรรมที่คาดหวังของเครื่องคอมพิวเตอร์ในการรันไบต์โค้ดที่ให้เป็น ไบต์โค้ดที่จะถูกรันบน JVM จะถูกส่งให้ตัวแปลภาษา ซึ่งจะทำการตรวจสอบไบต์โค้ดที่ได้ จากนั้นจะอ่านไบต์โค้ดและดำเนินการคำสั่งเครื่องที่เหมาะสม



รูปที่ 2-1 การทำงานของจาวา

2.1.3 การเขียนโปรแกรมภาษาจาวา

เนื้อหาในส่วนนี้จะรวมความรู้ทั้งหมดที่จำเป็นในเขียนโปรแกรม เช่น ชนิดของข้อมูล, ชนิดของสเตทเมนต์ ที่ใช้ในการควบคุมการทำงาน, การสร้างคลาส และเมธอด เป็นต้น

2.1.3.1 ตัวแปรและชนิดของข้อมูล

ในการประกาศใช้ตัวแปรจะต้องประกอบด้วย 2 ส่วนคือ ชื่อของตัวแปร และ ชนิดของตัวแปร และอาจรวมถึงที่ที่ตัวแปรนั้นถูกประกาศใช้ เพื่อจะได้พิจารณาถึงขอบเขตของตัวแปร ชนิดของตัวแปร

ตัวแปรทุกตัวในภาษาจาวาจะต้องมีชนิดของข้อมูล ชนิดของตัวแปรจะบอกถึงชนิดของข้อมูลที่จะเก็บและการทำงานที่อาจเกิดขึ้นกับตัวแปร

ในภาษาจาวามีชนิดของตัวแปร อยู่ 2 ชนิด คือ primary type และreference type primary type จะเก็บค่าได้ค่าเดียวและมีชนิดค่าๆดังนี้

Type	Size and Format	Description
<i>(whole numbers)</i>		
byte	8-bit two's complement	Byte-length integer
short	16-bit two's complement	Short integer
int	32-bit two's complement	Integer
long	64-bit two's complement	Long Integer
<i>(real numbers)</i>		
float	32-bit IEEE 754	Single-precision float point
double	64-bit IEEE 754	Double-precision float point

(other types)		
char	16-bit Unicode character	A single character
boolean	true or false	A boolean value (true or false)

- ตารางที่ 2-1 ชนิดของตัวแปร ในจาวา

ส่วน reference type ค่าของ reference type (เป็นในลักษณะของพอยเตอร์) จะอ้างไปยังค่า หรือกลุ่มของค่าของตัวแปร ไม่ได้เก็บค่าของตัวแปรนั้นไว้โดยตรง คลาสและอินเทอร์เฟซก็เป็น reference type อย่างหนึ่ง ฉะนั้นเมื่อคุณสร้างคลาสหรือ อินเทอร์เฟซใหม่ คุณจำเป็นต้องกำหนดชนิดของข้อมูลใหม่ **ชื่อของตัวแปร**

โปรแกรมจะอ้างถึงค่าของตัวแปร โดยผ่านทางชื่อของตัวแปรนั้นๆ ในที่ประชุม ได้มีการตกลงกันว่า ชื่อของตัวแปรจะขึ้นต้นด้วยตัวเล็ก และชื่อคลาสขึ้นต้นด้วยตัวใหญ่

ชื่อของตัวแปร ในภาษาจาวา

1. จะต้องประกอบขึ้นด้วยตัวอักษร ในยูนิโคท โดยยูนิโคทจะเป็นระบบ โค้ดของตัวอักษรที่ออกแบบมา เพื่อสนับสนุนข้อความ ในภาษามนุษย์ที่มีแตกต่างกันมากมายยูนิโคทสามารถจะมีตัวอักษรได้ถึง 65,536 ตัว (ปัจจุบันมีเพียง 34,168 ตัว) ทำให้ผู้เขียน โปรแกรมสามารถเขียน โค้ดที่สื่อความหมายใน ภาษาของตนเอง
2. จะต้องไม่ใช่ชื่อซ้ำกับคีย์เวิร์ดหรือ true หรือ false ของบูติน
3. ไม่ได้ใช้ในปัจจุบันแล้ว

Abstract	Boolean	Break	byte	case	Catch	char	class
Const *	Continue	Default	do	double	Else	extends	final
Finally	Float	For	goto *	if	Implement	import	instanceof
Int	Interface	Long	native	new	null	package	private
Protected	Public	Return	short	Static	Super	switch	Synchronized
this	Throw	Throws	Transient *	Try	void	volatile	while

ตารางที่ 2-2 คีย์เวิร์ดในจาวา

3. ชื่อจะต้องไม่ซ้ำกับตัวแปรอื่นที่ถูกประกาศภายในขอบเขตเดียวกัน

ในที่ประชุม ได้ตกลง ให้ชื่อของตัวแปรขึ้นต้นด้วยตัวเล็ก ถ้าประกอบด้วยคำมากกว่า 1 คำในชื่อ เช่น isValueOf ตัวอักษรที่ขึ้นต้นของแต่ละคำยกเว้นคำแรก ให้ขึ้นต้นด้วยตัวใหญ่

ขอบเขต

เป็นของเขตที่มี โค้ดอยู่ภายในซึ่งตัวแปรสามารถถูกเข้าถึงได้ขอบเขตของตัวแปรจะถูกพิจารณา เมื่อตัวแปรถูกสร้างและทำลาย ขอบเขตนี้จะทำให้ตัวแปรแบ่งออกเป็น 4 ประเภท

• member variable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- local variable
- method parameter
- exception handle parameters

member variable จะเป็นสมาชิกของคลาสหรือออบเจกต์ และถูกประกาศอยู่ในคลาส(แต่ไม่ได้อยู่ในเมธอดของคลาส)

local variable จะถูกประกาศไว้ภายในเมธอดหรือ กลุ่มของโค้ดภายในเมธอด โดยทั่วไป *local variable* จะถูกเข้าถึงได้ตั้งแต่ตัวแปรนั้นถูกประกาศจนจบขอบเขตของโค้ดที่ตัวแปร ถูกประกาศไว้ภายใน

method parameter จะถูกใช้ในการผ่านค่าไปให้เมธอดและคอนสตรัคเตอร์ ขอบเขตของ *method parameter* คือภายในเมธอดหรือ ภายในคอนสตรัคเตอร์ ที่มันเป็นพารามิเตอร์อยู่

exception handle parameters จะเหมือนกับ *method parameter* แต่จะถูกใช้ใน *exception handel* แทนที่จะเป็น ในเมธอดหรือคอนสตรัคเตอร์ การให้ค่าเริ่มต้นกับตัวแปร

local variable และ *method variable* สามารถกำหนดค่าเริ่มต้น ได้เมื่อมันถูกประกาศเป็นครั้งแรก โดยค่าที่กำหนดให้ต้องสอดคล้องกับชนิดของตัวแปรนั้นๆ

method parameter และ *exception handle parameter* ไม่สามารถกำหนดค่าเริ่มต้นให้ได้ ค่าของมันจะถูกกำหนดโดย *caller*

2.1.3.2 ตัวดำเนินการ

operator จะทำงานได้ทั้งบน *operand* (ตัวถูกดำเนินการ) ตัวเดียวและ 2 ตัว *operator* ที่ต้องการ *operand* 1 ตัวเรียกว่า *unary operators* โคนสามารถเป็นได้ทั้ง *prefix* (*operator* อยู่หน้า *operand*) และ *profix* (ตัว *operator* อยู่หลัง *operand*) ส่วน *operator* ที่ต้องการ *operand* 2 ตัวเรียกว่า *binary operators* ซึ่งเป็นแบบ *prefix* อย่างเดียว (*operator* อยู่ระหว่าง *operand*)

ในการทำงานของ *operator* จะมีการให้ค่ากลับมา ซึ่งค่าและชนิดของค่าที่ส่งกลับมานั้นขึ้นอยู่กับ *operator* และ ชนิดของ *operand*

operator แบ่งออกเป็น

- arithmetic
- relational and conditional
- bitwise logical
- assignment

Arithmetic operator

ในจาวามีการสนับสนุน *arithmetic operator* จำนวนมาก ซึ่งมี *operator* ต่างๆดังนี้

Operator	Use	Description
+	op1 + op2	Adds op1 and op2
-	op1 - op2	Subtracts op2 from op1
*	op1 * op2	Multiplies op1 by op2
/	op1 / op2	Divides op1 by op2
%	op1 % op2	Computes the remainder of dividing op1 by op2

ตารางที่ 2-3 binary arithmetic operation ของ C++

หมายเหตุ เครื่องหมาย + ยังใช้ในการเชื่อม String เข้าด้วยกัน

เครื่องหมาย + และ - ยังใช้โคคในการบอกถึงค่าของ operand ว่ามากกว่าหรือน้อยกว่า 0

Operator	Use	Description
+	+ op	Indicates a positive values
-	- op	Arithmetically negates op

ตารางที่ 2-4 เครื่องหมายที่ใช้บอกค่าของ operand

นอกจากนี้เรายังใช้เครื่องหมาย ++ และ -- แทนการเพิ่มค่าและลดค่าที่ละ 1 ของ operand ได้

Operator	Use	Description
++	Op++	Increments op by 1; evaluates to value before incrementing
++	++op	Increments op by 1; evaluates to value after incrementing
--	Op--	Decrements op by 1; evaluates to value before decrementing
--	--op	Decrements op by 1; evaluates to value after decrementing

ตารางที่ 2-5 การใช้ prefix และ postfix ในการเพิ่มและลดค่า

Relational and Conditional Operators

relational operator จะใช้ในการเปรียบเทียบค่าของ operand 2 ตัว และ พิจารณาความสัมพันธ์ระหว่าง operand

Operator	Use	Return true if
>	op1 > op2	op1 is greater than op2
>=	Op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less then op2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<=	Op1 <= op2	op1 is less then or equal to op2
=	Op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

ตารางที่ 2-6 relational operator ในจาวา

บ่อยครั้งที่ relational operator จะถูกใช้ร่วมกับ condition operator เพื่อใช้ในการพิจารณา หรือ คัดลึนใจที่ซับซ้อน

Operator	Use	Return true if
&&	Op1 && op2	op1 and op2 are both true
	op1 op2	Either op1 or op2 is true
!	! op	op is false

ตารางที่ 2-7 condition operator ในจาวา

เครื่องหมาย & สามารถใช้แทน && เช่นเดียวกับ | ซึ่งใช้แทน || ได้ ถ้า operand ทั้ง 2 ตัวเป็น boolean

Bitwise operators

bitwise operator อนุญาตให้คุณทำงานกับบิต ของข้อมูลได้ bitwise operator แสดงในตารางต่อไปนี้

Operator	Use	Operation
>>	op1 >> op2	shift bit of op1 right by distance op2
<<	op1 << op2	shift bit of op1 left by distance op2
>>>	op1 >>> op3	shift bit of op1 right by distance op2 (unsigned)
&	op1 & op2	bitwise and
	op1 op2	bitwise or
^	op1 ^ op2	bitwise xor
~	~ op	bitwise complement

ตารางที่ 2-8 Bitwise operator ในจาวา

Assignment operators

คุณสามารถใช้เครื่องหมาย "=" ในการกำหนดค่าให้กับตัวอื่นๆ นอกจากการกำหนดค่าโดยใช้ เครื่องหมาย = แล้ว จาวายังเพิ่ม assignment operator เพื่อให้คุณสามารถทำ arithmetic , logical หรือ bitwise operator ได้ 1 ครั้งก่อนที่จะให้ค่านั้นกับ operand

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operator	Use	Equivalent to
+=	Op1 += op2	op1 = op1 += op2
-=	Op1 -= op2	op1 = op1 -= op2
*=	Op1 *= op2	op1 = op1 *= op2
/=	op1 /= op2	op1 = op1 /= op2
%=	op1 %= op2	op1 = op1 %= op2
&=	op1 &= op2	op1 = op1 &= op2
=	op1 = op2	op1 = op1 = op2
^=	op1 ^= op2	op1 = op1 ^= op2
<<=	op1 <<= op2	op1 = op1 <<= op2
>>=	op1 >>= op2	op1 = op1 >>= op2
>>>=	op1 >>>= op2	op1 = op1 >>>= op2

ตารางที่ 2-9 assignment operator เพิ่มเติม

2.1.3.3 สเตตเมนต์ที่ใช้ในการควบคุมการทำงาน

ในจาวามี statement ที่ใช้ในการควบคุม หรือ ลำดับการทำงานอยู่หลายประเภท ดังตาราง

Statement	Keyword
making decision .	if - else , switch - case
loop	for , while do - while
exception	try - catch - finally , throw
miscellaneous	break , continue , label: , return

ตารางที่ 2-10 ประเภทของสเตตเมนต์ที่ใช้ควบคุมการทำงาน

if - else Statement

if - else statement ทำให้โปรแกรมของคุณสามารถเลือกทำงานตามเหตุการณ์ที่เข้ามา โดยรูปแบบที่ง่ายที่สุดอยู่ในรูป

```
if (expression)
```

```
statement
```

โปรแกรมจะทำ statement ถ้า expression เป็นจริง โดยจะไม่สนใจถ้า expression เป็นเท็จ แต่ถ้าต้องการพิจารณาในกรณีที่ expression เป็นเท็จด้วยเราจะเพิ่ม statement else เข้าไปเป็น

```
if (expression) {
```

```
// codes to perform when expression is true --
```

```
} else {
```

```
// codes to perform when expression is false -
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}
 else statement ยังมีอีกรูปแบบหนึ่งคือ else if ซึ่งจะทำการ statement โดยขึ้นอยู่กับ expression อื่น
 มีรูปแบบดังนี้

```

if (expression) {
    statement
} else if (expression1) {
    statement1
} else if (expression2) {
    ...
} else {statement n}
  
```

ในกรณีนี้ถ้า expression เป็นจริง โปรแกรมก็จะทำการ statement แต่ถ้า expression เป็นเท็จก็จะไป
 ตรวจสอบ expression1 ต่อว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงก็ทำการ statement1 ถ้าเป็นเท็จก็จะไปตรวจสอบ
 expression2 ต่อ แต่ถ้าไม่มี expression ใดที่เป็นจริงเลย โปรแกรมจะทำการ statement n if statement
 สามารถมี else if statement ได้หลายตัว แต่มี else statement ได้เพียงตัวเดียว

switch Statement

switch statement จะทำการพิจารณา expression ของมัน และจะเลือกทำการ case statement ที่เหมาะ
 สม ถ้าสังเกตเราจะเห็นว่า เราสามารถใช้ else if statement เขียนแทน switch statement ได้

switch statement ยังสามารถมี break statement หลัง case statement แต่ละตัว break statement
 จะทำให้การทำงานของ switch statement หยุดและไปทำการ statement แรกที่ตามหลัง switch statement
 break statement มีความจำเป็นเนื่องจาก case statement ทำงานแบบเรียงตามลำดับไปเรื่อยๆจนหมด
 case statement

```

int month;
int numDays;
...
switch (month) {
    case 1: System.out.println("January");
    case 3: System.out.println("March");
    case 5: System.out.println("May");
    case 7: System.out.println("July");
    case 8: System.out.println("August");
    case 10: System.out.println("October");
    case 12: System.out.println("December");
        numDays = 31; break;
    case 4: System.out.println("April");
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 6: System.out.println("June");
case 9: System.out.println("September");
case 11: System.out.println("November");
        numDays = 30; break;
case 2: System.out.println("February");
    if ( ((year % 4 == 0) && !(year % 100 == 0)) || (year % 400 == 0) )
        numDays = 29;
    else    numDays = 28; break;
default : System.out.println("Hay! That's is not a valid month.");
}

```

ท้ายสุดคุณสามารถใช้ default statement ที่ท้ายสุดของ switch เพื่อจะรองรับค่าที่ไม่ปรากฏใน case ตัวใดเลย ดังตัวอย่าง ถ้าค่าของเดือนที่รับมา ไม่ใช่ 1-12 ก็จะปรากฏข้อความ Hay! That's not a valid month.

loop Statement

สำหรับ loop statement จะเป็นการทำ statement หรือ กลุ่มของ statement ซ้ำเรื่อยๆ ในขณะที่ expression เป็นจริง แบ่งออกเป็น 3 แบบคือ

- while statement มีรูปแบบดังนี้

```

while (expression)
    statement

```

โดย ถ้า expression ของ while statement เป็นจริง ถึงจะทำ statement โคนที่ statement สามารถเป็น statement เดียว หรือเป็นกลุ่มของ statement ซึ่งตามกฎของจาวาจะต้องอยู่ภายในเครื่องหมายปีกกา (“{“ และ “}”)

- for loop จะใช้เมื่อคุณทราบถึงข้อบังคับที่แน่นอนของ loop (ค่าเริ่มต้น , ค่าสิ้นสุด , ค่าที่เพิ่มขึ้นในแต่ละครั้ง) for loop มักใช้กับการทำงานกับข้อมูลใน array หรือตัวอักษรใน string โดย for loop มีรูปแบบดังนี้

```

for (initialization; termination; increment;)
    statement

```

initialization เป็น expression ที่ใช้ในการเริ่มต้น loop ซึ่งจะถูก execute ครั้งเดียวเมื่อเริ่มต้นทำ loop termination เป็น expression ที่ใช้ในการพิจารณาเมื่อไหร่ถึงจะสิ้นสุด loop expression นี้จะถูกพิจารณาก่อนที่จะเริ่มทำงานในแต่ละ loop เมื่อพิจารณาว่าเป็นเท็จ loop ก็จะสิ้นสุด increment เป็น statement ที่จะทำในทุกๆ loop หลังจากที่ได้พิจารณา terminate statement แล้วว่าเป็นจริง

องค์ประกอบของ for loop ที่กล่าวมาแต่ละตัว หรือ ทั้งหมดสามารถละได้ โดยใส่เครื่องหมาย Semi-colon (;) แทน statement แต่ละตัวที่จะละ

- do-while loop จะคล้ายกับ while loop ต่างกันที่ do-while loop จะทำการพิจารณา

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง หากมีการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

expression หลังการทำงานจบแต่ละ loop โดยมีรูปแบบดังนี้

```
do {
    statements
} while (booleanExpression);
```

Exception handle Statements

เมื่อเกิดข้อผิดพลาดขึ้นใน method ของจาวา method สามารถที่จะส่ง exception ไปยังตัวที่เรียกใช้ method นั้น เพื่อบอกให้รู้ว่าเกิดข้อผิดพลาดขึ้น และ ชนิดของข้อผิดพลาดที่เกิด method ที่เป็นตัวเรียกสามารถใช้ try , catch และ finally statement ในการจับและแก้ไขข้อผิดพลาดที่เกิด

Branching Statements

- break statement ในตัวอย่างของ switch statement break statement ทำให้คุณสามารถกระโดดการทำงานจาก statement ปัจจุบัน ไปยัง statement ที่อยู่ถัดจาก statement ปัจจุบันทันที

ยังมี break statement รูปแบบอื่นที่จะกระโดดการทำงานไปยัง label statement โดยคุณสามารถกำหนด label ให้กับ statement ได้ โดยใส่ label ตามด้วยเครื่องหมาย “:” แล้วตามด้วย statement

```
breakToHere : someJavaStatement
```

การกระโดดทำงานจาก break statement ไปยัง label statement ทำได้โดยใช้ break statement ตามด้วยชื่อของ label statement ที่ต้องการจะกระโดดไปทำงาน ดังนี้

```
break breakToHere ;
```

- continue statement จะอยู่ใน loop เพื่อใช้กระโดดจาก statement ปัจจุบัน ไปยังต้นของ loop หรือไปยัง label statement ถ้าต้องการกระโดดไปยัง label statement จะต้องใช้ continue statement ที่ตามด้วย label ของ statement ที่จะทำ

- return statement ใช้เพื่อออกจาก method ปัจจุบัน กลับไปยัง statement ที่อยู่ใน method ที่เป็นตัวเรียก การ return มี 2 แบบคือ มีการส่งค่ากลับไปด้วย กับ ไม่มีการส่งค่ากลับ การส่งค่ากลับไปด้วยทำได้โดย เติมค่า หรือ expression ที่ใช้หาค่า หลัง return

2.1.3.4 Arrays and Strings

Array

เหมือนกับ variable อื่น ก่อนที่จะใช้ array ได้จะต้องมีการประกาศก่อน ซึ่งการประกาศต้องประกอบด้วย ชนิดของ array และ ชื่อของ array ชนิดของ array คือชนิดของข้อมูลที่จะเก็บใน array นั้นๆ โดยจะต้องกำหนดชนิดของ array เพื่อประกาศใช้ array การประกาศ array ไม่ได้มีการจองเนื้อที่เพื่อใช้เก็บข้อมูลของ array ถ้าโปรแกรมพยายามเข้าถึงหรือเก็บค่าลงใน array ก่อนมีการจองเนื้อที่ compiler จะบอกถึงความผิดพลาด ในการจองเนื้อที่ให้กับ array คุณต้องทำการ instance array ก่อน (ขั้นตอนเหมือนกับการ instance object จาก class : declaration , instantiation , initialization)

ปรกติในการสร้าง array เราใช้ operator new ตามด้วยชนิดของข้อมูลใน array และจำนวนของข้อมูลที่จะเก็บซึ่งอยู่ภายในเครื่องหมาย “[” และ “]”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elementType[] arrayName = new elementType [arraySize]
```

ขณะนี้ได้มีการจองเนื้อที่บางส่วนให้กับ array ของคุณแล้ว คุณสามารถเก็บค่าหรือเข้าถึงข้อมูลภายใน array ได้ ในจาวา array จะเริ่มที่ 0 และ สิ้นสุดที่ ขนาดของ array ลบด้วย 1 ข้อมูลใน array ของจาวานั้น สามารถเก็บ reference type ได้ เมื่อเราทำการสร้าง array ของ reference type เนื้อที่ที่จองไว้จะเก็บค่าที่ใช้ในการ reference ไปยังข้อมูลอีกที่ แต่ไม่ได้จองเนื้อที่สำหรับข้อมูลไว้ ฉะนั้นหากพยายามเข้าถึงข้อมูลโดยตรงจะเกิดข้อผิดพลาดขึ้น

String

ตัวอักษรที่นำมาเรียงต่อกันเราเรียกว่า string โดยถูกนำมาใช้ในจาวาในรูปแบบ string class (อยู่ใน package java.lang)

การเชื่อมต่อ string สำหรับจาวาอนุญาตให้ใช้เครื่องหมาย "+" โดยมากมักใช้ในการพิมพ์ output ดังตัวอย่างข้างล่าง

```
"Input has " + count + " chars."
```

เป็นการนำ string 2 ตัวมาเชื่อมต่อกันคือ "Input has " กับ " chars." (string จะอยู่ในเครื่องหมาย "") string ตัวที่ 3 (ตัวที่อยู่ตรงกลาง) ซึ่งเป็น variable อาจเป็นชนิดอื่นนอกจาก string ก็ได้ จะถูกแปลงเป็น string และถูกเชื่อมเข้ากับตัวอื่น (String ไม่สามารถแปลงไปเป็นชนิดอื่นได้หลังจากที่มันถูกสร้างขึ้นแล้ว)

2.1.3.5 ออบเจกต์และคลาส

วงจรชีวิตของออบเจกต์

เราสามารถแบ่งวงจรชีวิตของจาวาออกเป็น 3 ช่วงด้วยกันคือ

1. สร้างออบเจกต์ หรือเรียกอีกอย่างว่าสร้าง instance ของ class ซึ่งจะประกอบด้วยขบวนการ 3 อย่างคือ declaration , instantiation และ initialization ดังตัวอย่างข้างล่าง

```
Rectangle rect = new Rectangle (0 , 0 , 100 , 200);
```

statement ข้างบนจะสร้าง Rectangle object ขึ้นมาใหม่ (Rectangle เป็น Class หนึ่งใน package java.awt)

Declaration การ declare จะเหมือนกับการ declare variable ทั่วไป โดยประกอบด้วย ชนิดของ object และ ชื่อของ object

```
type name ( Rectangle rect )
```

การ declare ไม่ได้สร้าง object ขึ้นมา เป็นเพียงการบอกให้ compiler รู้ว่าเราจะให้ name ในการอ้างอิง variable ที่มีชนิดเป็น type

Instantiation new operator จะสร้าง instance ของ class โดยการจองเนื้อที่ในหน่วยความจำ new ต้องการ argument ตัวเดียวคือ ตัวเรียก constructor method constructor method เป็น method พิเศษที่มีให้ในแต่ละ class ของจาวา โดยจะตอบสนองในการ initialization object ใหม่ ดังนั้น new จะสร้าง object ใหม่ และ constructor method จะทำ initialization

Rectangle (0, 0, 100, 200) จะไปเรียก constructor สำหรับ Rectangle new operator จะส่งค่าที่อ้างไปยัง object ใหม่กลับมา ซึ่งจะถูกกำหนดให้กับ variable ที่มี type ที่เหมาะสม

Initialization ดังที่กล่าวไว้ class จะมี constructor method ซึ่งใช้ในการ initialize object ใหม่ class หนึ่งๆสามารถมี constructor ได้หลายตัวเพื่อให้สามารถทำการ initialization ได้หลายแบบ constructor method นี้จะมีชื่อเหมือนกับ class โดยจะเป็นแบบไม่ส่งค่ากลับ

Rectangle (0, 0, 100, 200)

constructor ที่ไม่ต้องการ argument (ในตัวอย่างข้างบนต้องการ argument 4 ตัว) เรียกว่า *default constructor* ซึ่ง class ส่วนมากจะมี 1 constructor และเป็น default constructor

2. ใช้ออบเจกต์ โดยมีวิธีการในการใช้ดังนี้

การอ้างถึง variable ของ object ในการอ้างถึง variable ของ object ทำได้โดย

objectReference . *variable*

เมื่อคุณเข้าถึง variable ของ object หมายความว่า คุณเข้าถึง variable ของ object เฉพาะแต่ละตัว เนื่องจากในการอ้างถึง variable ใน object ที่ได้กล่าวแล้ว ต้องเป็นสิ่งที่อ้างอิงไปถึง object ดังนั้นเราสามารถใช้นew operator ซึ่งจะให้ค่าที่อ้างถึง object กลับมาได้

height = new Rectangle (0, 0, 100, 200).height

การเรียกใช้ method ของ object การเรียกใช้ method ของ object เหมือนกับการเรียกใช้ variable ของ object โดยการเอาชื่อ method ที่ต้องการตามหลัง object reference โดยมี จุด (".") ชั้นกลาง และส่ง argument ไปยัง method ภายในเครื่องหมายวงเล็บ ถ้าต้องการ argument ก็ให้ใส่วงเล็บเปล่า

objectReference . *methodName* (*argumentList*); หรือ

objectReference . *methodName* ();

การเรียกใช้ method เราอาจเรียกอีกอย่างว่า *messages* ซึ่งมีองค์ประกอบ 3 อย่าง (ชื่อ object ผู้รับ, ชื่อ method, parameter ที่ method ต้องการ) ดังตัวอย่างข้างล่าง

rect . move (36, 56);

เป็น message ที่ส่งไปให้ object rect บอกให้ทำ method move โดยส่ง parameter 36 กับ 56 ให้กับ method

3. ทำลายออบเจกต์ที่ไม่ได้ใช้ ในจาวาอนุญาตให้เราสร้าง object ได้มากเท่าที่ต้องการ โดยที่ระบบยังสามารถรองรับได้ โดยจะไม่ทำลาย JRE (Java Runtime Environment) จะลบ object ออกถ้ามันเห็นว่าไม่ได้ใช้อีกต่อไปแล้ว ซึ่งขบวนการนี้เรียกว่า *garbage collection* object ที่ถูกเลือกเพื่อทำลายคือ object ที่ไม่มีตัวใดอ้างถึง object นั้นแล้ว ประสิทธิภาพการอ้างถึงใน variable มักจะถูกทิ้งไปเมื่อออกนอก scope หรือคุณสามารถทิ้ง object reference ได้โดยการให้ค่ากับ variable ที่มีชนิดของข้อมูลเป็น reference type นั้น

ก่อนที่ object จะถูกทำลาย garbage collector (JRE) จะให้โอกาส object ที่จะปล่อย resource ที่ใช้ หรือ drop reference เพื่อให้ object อื่นมีโอกาสได้ใช้ ซึ่งขบวนการนี้เรียกว่า *finalization* โดย finalize

method จะอยู่ใน object class class จะต้องทำการเขียน finalize method ใหม่เพื่อให้เหมาะกับ object ของ type นั้นๆ

การสร้างคลาส

การ implement class ประกอบด้วย 2 อย่างคือ class declaration และ class body

```
classDeclaration {
    classBody
}
```

Class declaration

ในการ declare class อย่างน้อยที่สุดต้องมี class ซึ่งเป็น keyword และ ชื่อ class

```
class NameOfClass {
    ...
}
```

ชื่อ class ต้องขึ้นต้นด้วยตัวใหญ่ อย่างไรก็ตาม class declaration สามารถบอกรายละเอียดเกี่ยวกับ class ได้อีก ได้แก่

- ประกาศว่า Class ใดเป็น superclass
- แสดง interface ที่ implement โดย class
- บอกว่า class เป็นแบบ public , abstract หรือ final

การประกาศว่าคลาสใดเป็น superclass

ในจาวาทุกๆ class จะต้องมี superclass ถ้าคุณไม่ได้กำหนด มันจะถือว่าเป็น object class (เป็น class ที่เป็น superclass ของทุก class ในจาวาทั้งทางตรงและทางอ้อม ประกาศอยู่ใน java.lang) ในการกำหนด superclass ให้ใช้ keyword extends ตามด้วยชื่อของ superclass โดยจะอยู่ระหว่าง ชื่อ class และ “{“

```
class NameOfClass extends SuperClassName {
    ...
}
```

แสดง interface ที่ implement โดยคลาส

interface จะประกาศกลุ่มของ method และ ข้อกำหนด ที่ไม่ได้มีการ implement ให้กับ method ใด เมื่อ class ต้องการจะ implement interface ก็จะมีหมายถึง class ต้องการที่จะ implement method ทุกตัวที่ประกาศใน interface

class สามารถที่จะ implement interface ได้มากกว่าหนึ่ง โดยใช้ keyword interfaces ตามด้วย ชื่อ interface (ถ้ามีมากกว่า 1 ให้ใช้เครื่องหมาย “,” ขึ้นระหว่างชื่อ) ถ้า implement ไม่ครบทุก method ที่มีใน interface compiler จะฟ้องถึงข้อผิดพลาด

public , abstract และ final class

คุณสามารถเลือก class modifier (public , abstract , final) ตัวใดตัวหนึ่งสำหรับ class declaration เพื่อประกาศว่า class ของคุณเป็นแบบ public , abstract หรือ final modifier จะมาก่อน keyword class

public modifier จะประกาศว่า class ของคุณสามารถถูกใช้โดย object หรือ class ที่อยู่นอก package ปัจจุบันได้ โดยปรกติแล้ว class จะถูกใช้โดย class อื่นที่อยู่ใน package เดียวกับที่มันถูกประกาศไว้เท่านั้น โดยมรการตกลงว่า ถ้าใช้ public keyword ใน class declaration จะพยายามให้มันอยู่เป็นตัวแรกในการ ประกาศ

abstract modifier จะประกาศว่า class ของคุณเป็น abstract class ซึ่ง abstract class จะมี abstract method (เป็น method ที่ยังไม่มีการ implement) อยู่ภายใน abstract class มีแนวโน้มที่จะเป็น subclass และ ไม่สามารถสร้าง instance ได้

final modifier จะประกาศว่า class ของคุณ นั่นคือ class ของคุณไม่สามารถเป็น subclass ได้ เนื่องจากเหตุผล 2 ประการคือ เพื่อความปลอดภัย และ เพื่อการออกแบบ

สรุปรูปแบบของ class declaration ได้ดังนี้

```
[modifiers] class ClassName [extends SuperClassName] [implements InterfaceName] {
...
}
```

Class Body

ตัว class จะแบ่งออกเป็น 2 ส่วนคือ variables declaration และ methods โดยมากเราจะทำการ ประกาศ variable ก่อนตามด้วย method declaration และ implement

```
ClassDeclaration {
    memberVariableDeclarations
    methodDeclarations
}
```

Declaration member variables

การประกาศ variable จะประกอบด้วย 2 ส่วนคือ ชนิดของ variable และ ชื่อของ variable

```
type variableName ;
```

การ declare นี้จะเหมือนกับ local variables และ method parameters ทุกอย่างแม้แต่การตั้งชื่อ โดยชื่อของ variable สามารถซ้ำกับชื่อของ method ใน class เดียวกันได้

นอกจากนี้เรายังสามารถกำหนดคุณสมบัติอื่นๆของ variable ได้ดังนี้

```
[accessSpecification] [static] [final] [transient] [volatile] type variableName ;
```

- *accessSpecification* กำหนดการเข้าถึง variable จาก class อื่น การควบคุมการเข้าถึง method ก็ใช้ specification เดียวกันดังตาราง

Specifier	class	subclass	package	world
Private	X			
Protected	X	X *	X	
Public	X	X	X	X

Package	X		X	
---------	---	--	---	--

ตารางที่ 2-11 ระดับการเข้าถึงของแต่ละ specified

- static ใช้เพื่อแสดงว่า variable เป็นสมาชิกของ class variable (ไม่ใช่ของ instance variable) เราสามารถใช้ static ในการประกาศ class method ได้
- final ใช้เพื่อแสดงว่า variable เป็นค่าคงที่
- transient ใช้เพื่อแสดงว่า variable ไม่ใช่ส่วนหนึ่งของสถานะที่เป็นอยู่ของ object
- volatile ใช้แสดงว่า variable สามารถเปลี่ยนแปลงแบบ asynchronous สำหรับการทำงานของ thread พร้อมกัน

Implementing methods

เหมือนกับการ implement class มีองค์ประกอบ 2 ส่วนคือ : การประกาศ method และ ตัว method

```
methodDeclaration {
    methodBody
}
```

Method declaration อย่างน้อยต้องประกอบด้วย ชื่อ method และ ชนิดของข้อมูลที่ method ส่งกลับ

```
returnType methodName () {
    ...
}
```

ในการส่งค่ากลับ ชนิดของข้อมูลที่ส่งกลับเป็นได้ทั้ง primitive type และ reference type (ถ้าไม่มีการส่งค่ากลับให้ใส่ keyword void) method จะใช้ return operator ในการส่งค่ากลับ method ใดที่ไม่ได้ประกาศ void ได้ ต้องมี return statement ชนิดของข้อมูลที่ส่งกลับมาต้องเหมือนกับที่ method ได้ประกาศไว้

ชื่อของ method มีวิธีการตั้งชื่อดังนี้

1. method สามารถใช้ชื่อซ้ำกันได้ แต่มีการส่งค่า parameter ต่างกัน (ชนิดของ parameter ต่างกัน)
 2. method ที่มีชื่อเดียวกับ class จะเป็น *constructor* ซึ่งจะทำหน้าที่ในการ initial object ที่สร้างขึ้นใหม่ของ class type นั้น
 3. class สามารถ override method ใน superclass ของมันได้ การ override method จะต้องมีการมีชื่อ , return type , parameter list เหมือนกับ method ที่จะ override
- ลักษณะเพิ่มเติมของ method declaration ซึ่งจะบอกถึง argument ที่ method ยอมรับ

[*accessSpecifier*] [static] [abstract] [final] [native] [synchronized]

returnType methodName ([paramList]) [throws exceptionsList]

Passing Information into a Method เมื่อคุณเขียน method คุณประกาศจำนวนและชนิดของ argument ที่ method นั้นต้องการ โดยจะต้องบอกชื่อและชนิดของ argument แต่ละตัวใน method signature ถ้ามี argument มากกว่า 1 ตัว แต่ละตัวให้ขึ้นด้วยเครื่องหมาย “,”

```
methodDeclaration ( argumentType argumentName ) {
    methodBody
}
```

ชนิดของ argument สามารถเป็นได้ทั้ง primitive type และ reference type ส่วนชื่อของ argument สามารถซ้ำกับชื่อของ class variable ได้ ซึ่งมักใช้ใน constructor การใช้ชื่อของ argument ที่ซ้ำกับ ชื่อของ class variable ทำให้การอ้างชื่อใน method หมายถึง argument ถ้าต้องการอ้างถึง class variable (member variable) ให้ใช้ keyword this ตามด้วยจุด (“.”) และชื่อของ class variable ที่ซ้ำกับ argument

Method Body ในตัวของ method คุณสามารถใช้ keyword this ในการอ้างถึงสมาชิกใน object ปัจจุบัน (object ที่ method ถูกเรียก) ใช้ keyword super อ้างถึงสมาชิกใน superclass ซึ่ง object ปัจจุบันถูกซ่อน หรือ ถูก override นอกจากนั้นภายใน method ยังสามารถทำการ declare variable ซึ่งเป็น local variable ได้ด้วย

Instance and Class member

เมื่อคุณประกาศ member variable เช่น

```
Class MyClass {
    float aFloat ;
}
```

หมายถึงคุณได้ประกาศ *instance variable* ทุกครั้งที่คุณสร้าง instance ของ class ระบบจะสร้างสำเนา instance variable ของ class ให้กับแต่ละ instance variable จะต่างกับ *class variable* (ซึ่งคุณประกาศโดยใช้ static modifier) โดยระบบจะทำการจองเนื้อที่ให้กับ class variable เพียงครั้งเดียวเมื่อพบ class นั้นครั้งแรก instance อื่นของ class นั้นจะให้ class variable นี้ร่วมกัน คุณสามารถเข้าถึง class variable โดยผ่าน instance หรือ class โดยตรง

method ก็เช่นเดียวกัน มีได้ทั้ง *instance method* และ *class method* (ทำได้โดยประกาศใน method declaration โดยใช้ static modifier เช่นเดียวกับ class variable) โดยที่ instance method จะทำงานกับ instance variable ของ object ปัจจุบัน แต่ก็สามารถเข้าถึง class variable ได้ ส่วน class method จะทำงานกับ class variable เท่านั้นไม่สามารถเข้าถึง instance variable ที่ประกาศอยู่ใน class เดียวกับมันได้ อย่างไรก็ตามสมาชิกที่ถูกระบุอยู่ใน class เราเรียกว่า instance member

instance ของ class ทุกตัวจะทำการ implement class method ร่วมกัน instance method จะอ้างถึง instance variable ของ object โดยใช้ชื่อของ instance variable นั้นได้เลย เนื่องจาก object จะซ่อน instance variable ไว้โดย parameter ของ method

ข้อแตกต่างอีกข้อของ instance member กับ class member คือ class member สามารถเข้าถึงจาก class ได้เลย ไม่ต้อง instantiate class เพื่อเข้าถึง class member

Life Cycle Methods

วงจรชีวิตของ method แบ่งออกเป็น 2 ส่วนด้วยกันคือ

Constructor เป็นช่วงเริ่มต้นของวงจรชีวิต ทุก class ในจาวาจะมี method พิเศษที่เรียกว่า constructor ซึ่งใช้ในการเริ่ม object ใหม่ของ type นั้น constructor จะมีชื่อเดียวกับ class และ class หนึ่งๆจะมี constructor ได้หลายตัว เราแยกแยะ constructor เหล่านี้ได้โดยจำนวน หรือ ชนิดของ argument

constructor จะใช้ argument ของมันในการ initial สถานะของ object ใหม่ เมื่อคุณสร้าง object คุณต้องเลือก constructor ที่มี argument ตรงตามที่คุณต้องการมากที่สุด

เมื่อคุณเขียน class ของคุณขึ้นมาเอง คุณไม่จำเป็นต้องเขียน constructor สำหรับมัน constructor ที่ไม่ต้องการ argument จะถูกจัดให้ระบบโดยอัตโนมัติ (สำหรับทุก class) การ declare และ implement จะเหมือนกับ method ทั่วไป แต่ใน constructor จะไม่มีการส่งค่ากลับ

ในบางครั้งเราต้องการ initial โดยใช้ constructor ของ superclass ซึ่งทำได้โดย

```
super ("ClassName ");
```

statement super นี้จะอยู่เป็น statement แรกในตัว constructor

ในการประกาศ constructor คุณสามารถใช้ specifier ในการกำหนดว่า object ใดบ้างที่สามารถสร้าง instance ของ class คุณได้

private ไม่มี class ใดสามารถ instantiate class ของคุณเป็น object ได้

protected เฉพาะ subclass ของ class คุณเท่านั้นที่สามารถสร้าง instance ของมันได้

public ใครก็ได้สามารถสร้าง instance ของ class คุณได้

package ไม่มี class ใดที่อยู่นอก package (ที่คุณอยู่) ที่จะสร้าง instance ของ class คุณได้

Writing a finalize Method ก่อนที่ method จะถูกทำลาย ระบบจะเรียก finalize method เพื่อให้มันปล่อย resource ที่มีอยู่ก่อนถูกทำลาย คุณสามารถสร้าง finalize method ได้โดยกำหนดและ implement method ของคุณให้มีชื่อ finalize การประกาศ finalize method ทำได้ดังนี้

```
protected void finalize() throws Throwable
```

finalize method ถูกประกาศไว้ใน class java.lang.Object ดังนั้นเมื่อคุณเขียน finalize ของ class คุณก็เป็นการ override ใน superclass ตัวหนึ่ง

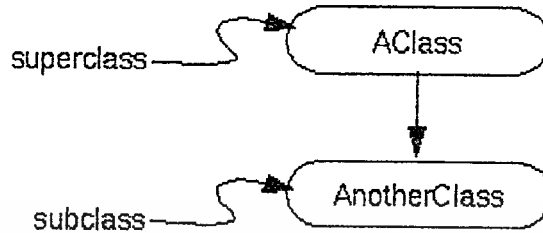
ถ้า superclass ของ class คุณมี finalize method finalize method ของ class คุณควรเรียก finalize ของ superclass หลังจากที่มันทำความสะอาดเสร็จแล้ว

```
protected void finalize() throws Throwable {
    ... // Clean up code for this class here
    super.finalize();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

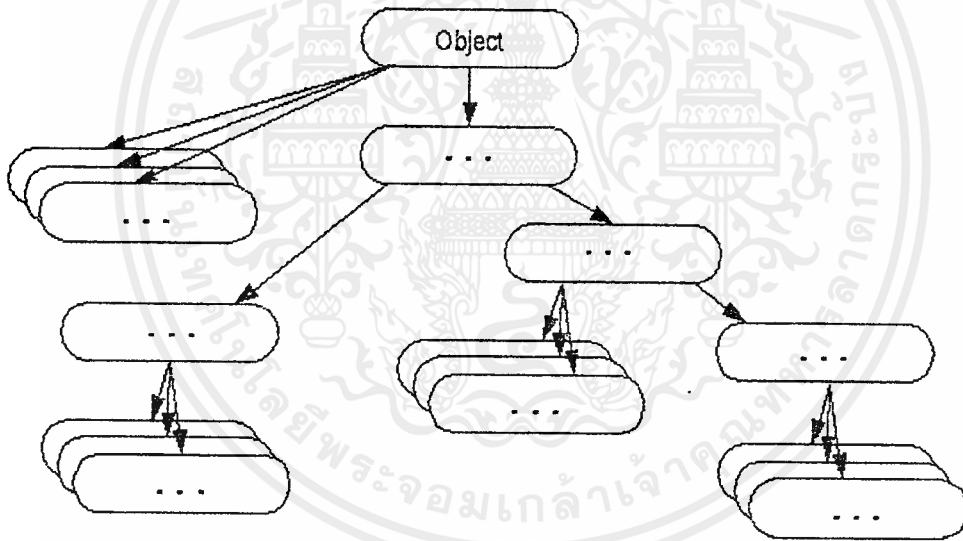
Subclass , Superclass and Interface

จากที่เราได้รามาแล้วว่า class อาจได้มาจาก class อื่น class ที่มีกำเนิดจาก class อื่นเรียกว่า *subclass* ส่วน class ที่ให้กำเนิด class อื่นเรียกว่า *superclass*



รูปที่ 2-2 superclass และ subclass

ความเป็นจริงแล้ว ในจาวาทุก class จะ ได้มาจาก class อื่นทั้งสิ้น โดย class ที่เป็นต้นแบบหรือให้กำเนิด class อื่นๆทั้งหมดในจาวาคือ class Object ซึ่งอยู่ใน java.lang



รูปที่ 2-3 class hierarchy

subclass จะรับช่วง state และ behavior ในรูปของ variable และ method จาก superclass ของมัน subclass สามารถใช้สิ่งที่ได้รับช่วงมาจาก superclass เหมือนกับเป็นของตนเอง และสามารถที่จะเปลี่ยนแปลงหรือ override มันได้ ทำให้ยังมีการสืบทอดหลายครั้งยิ่งทำให้ class ยังมีคุณสมบัติพิเศษมากขึ้น

การสร้าง subclass

คุณประกาศว่า class นี้เป็น subclass ของ class อื่นได้ในช่วง class declaration

```
class SubClassName extends SuperClassName {
```

```
...
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในจาวาแต่ละ class สามารถมี superclass ได้เพียงตัวเดียว (จาวาไม่รองรับ *multiple inheritance*)

What Member Variable does a Subclass inherit

กฎ : subclass จะรับช่วงต่อ member variable ทั้งหมดที่อยู่ใน superclass ที่สามารถเข้าถึงได้

จาก subclass (ยกเว้น member variable ที่ซ่อน โดย subclass)

ข้อตกลงในการรับช่วงต่อ member variable

- subclass จะรับช่วงต่อ member variable ที่ประกาศให้เป็น public หรือ protected
- subclass จะรับช่วงต่อ member variable ที่ประกาศโดยไม่มี access specifier ในขณะที่ subclass อยู่ใน package
- เดียวกับ superclass
- subclass จะไม่รับช่วงต่อ member variable ของ superclass ถ้า subclass ประกาศใช้ member variable ของมันที่มีชื่อซ้ำกับของ superclass (เนื่องจาก member variable ของ subclass จะซ่อน member variable ใน superclass ไว้)
- subclass จะไม่รับช่วงต่อ member variable ของ superclass ที่เป็น private

Hiding Member Variable

จากที่ได้กล่าวไปแล้วว่า member variable ใน subclass จะซ่อน member variable ที่มีชื่อเดียวกันใน superclass โดยเราสามารถเข้าถึง member variable ที่ซ่อนอยู่โดยผ่าน superclass ของมัน โดยใช้

`super . memberVariable`

super เป็น keyword ในจาวาที่ให้ method สามารถเข้าถึง member variable ที่ซ่อนอยู่ และให้ override method ใน superclass ได้

What Method does a Subclass inherit

กฎ : subclass จะรับช่วงต่อ member method ทั้งหมดที่อยู่ใน superclass ซึ่งสามารถเข้าถึงได้

จาก subclass (ยกเว้น method ที่ override โดย subclass)

ข้อตกลงในการรับช่วงต่อ member method

- subclass จะรับช่วงต่อ member method ที่ประกาศให้เป็น public หรือ protected
- subclass จะรับช่วงต่อ member method ที่ประกาศโดยไม่มี access specifier ในขณะที่ subclass อยู่ใน package
- subclass จะไม่รับช่วงต่อ member method ของ superclass ถ้า subclass ประกาศใช้ method ที่มีชื่อซ้ำกับของ superclass method ใน subclass จะถูกเรียกว่า override method ใน superclass
- subclass จะไม่รับช่วงต่อ member method ของ superclass ที่ประกาศเป็น private

subclass สามารถ override การ implement หรือเพิ่มการทำงานให้กับ method ที่รับช่วงมา

Method ที่ subclass ไม่สามารถจะ override ได้

- subclass ไม่สามารถ override method ที่ประกาศเป็นแบบ final ใน superclass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- subclass ไม่สามารถ override method ที่ประกาศเป็นแบบ static ใน superclass (subclass ไม่สามารถ override class method ได้)

Method ที่ subclass ต้อง override

subclass จะต้อง override method ที่ประกาศเป็นแบบ abstract ใน superclass หรือตัว subclass จะต้องเป็นแบบ abstract

2.1.3.6 Handling Errors Using Exceptions

นิยาม exception เป็นเหตุการณ์ที่เกิดขึ้นระหว่างการ execute โปรแกรม ซึ่งทำลายการไหลของชุดคำสั่ง

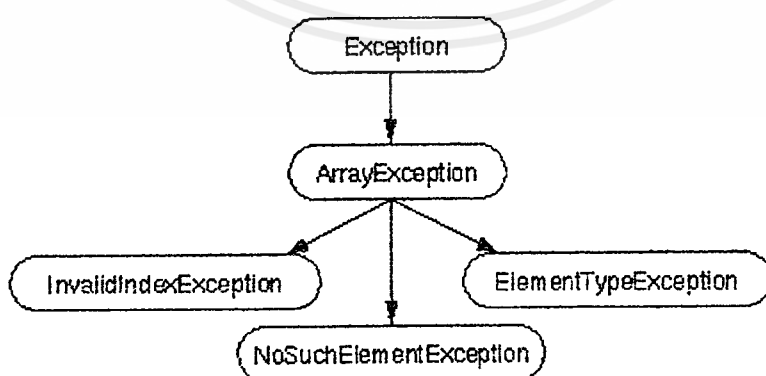
exception object จะเก็บข้อมูลเกี่ยวกับ exception รวมทั้งชนิดและสถานะของโปรแกรมเมื่อเกิดข้อผิดพลาดขึ้น ในจาวาการสร้าง exception object และส่งไปยัง runtime system เรียกว่า *throwing an exception*

หลังจาก method ได้ส่ง exception แล้ว runtime system จะทำการหาบางอย่างมารับ exception ที่เกิดขึ้น ซึ่งบางสิ่งที่ว่าคือกลุ่มของ method ใน call stack ของ method ที่เกิดข้อผิดพลาดขึ้น runtime system จะทำการค้นหาข้อย่อยหลังใน call stack เริ่มจาก method ที่เกิดข้อผิดพลาด จนกระทั่งพบ method ที่เก็บ *exception handler* ที่เหมาะสม exception handler จะถูกพิจารณาว่าเหมาะสมถ้าชนิดของ exception throw เป็นชนิดเดียวกับชนิดของ exception ที่ exception handler มี โดย exception ที่ถูกเลือกเรียกว่า *catch the exception*

ถ้า runtime system ทำการหาจนหมด call stack แล้วยังไม่เจอ exception handler ที่เหมาะสม runtime system และตามด้วย โปรแกรมจาวาก็จะสิ้นสุดลง

ด้วยการใช้ exception เพื่อจัดการกับข้อผิดพลาดที่เกิดขึ้น โปรแกรมจาวาจะได้ประโยชน์ดังนี้

- แยกโค้ดที่ใช้จัดการกับข้อผิดพลาดออกจากโค้ดทั่วไป
- สามารถตรวจสอบข้อผิดพลาดจาก call stack ได้
- รวบรวมชนิดของข้อผิดพลาดและแสดงความแตกต่างของข้อผิดพลาด ดังนี้



รูปที่ 2-4 ตัวอย่างของ exception hierarchy

Java's Catch or Specify Requirement

Java ต้องการ method catch หรือ specify ในการตรวจ exception ที่อาจถูกส่งมาใน scope ของ method ความต้องการนี้มื่อองค์ประกอบซึ่งจะอธิบายดังนี้ catch , specify , checked exception และ exception ซึ่งอาจถูกส่งเข้ามายัง scope ของ method

- **catch** method สามารถ “catch” exception ได้โดยจัดหา exception handler สำหรับ exception ชนิดนั้นๆ
- **specify** ถ้า method เลือกว่าจะไม่ “catch” exception method จะต้อง specify ว่ามันสามารถ throw exception นั้นได้ เนื่องจาก exception ที่สามารถ throw ได้จะเป็นส่วนหนึ่งของการเขียนโปรแกรมแบบ public interface ของ method นั่นคือตัวที่เรียกใช้ method จะต้องรู้เกี่ยวกับ exception ซึ่ง method สามารถ throw ได้ ดังนั้นใน method signature คุณ specify exception ซึ่ง method สามารถ throw ได้
- **checked exceptions** ใน Java จะมีชนิดของ exception ต่างกัน ที่เราจะพูดถึงในขณะนี้คือ runtime exception ซึ่งจะเกิดขึ้นในช่วง runtime system runtime exception สามารถเกิดขึ้นที่ใดก็ได้ในโปรแกรม สิ่งที่ต้องสูญเสียในการตรวจสอบสำหรับ runtime exception มักจะมากกว่า catching และ specifying มัน ดังนั้น compiler จึงไม่ต้องการ catch หรือ specify runtime exception *checked exceptions* เป็น exception ซึ่งไม่ใช่ runtime exception และถูกตรวจสอบโดย compiler compiler จะตรวจสอบว่า exception เหล่านี้เป็น catch หรือ specify
- **exception ที่สามารถส่งเข้ามายัง scope ของ method ได้** ซึ่งในที่นี้หมายความว่ารวมถึง exception ใดๆที่สามารถส่งในขณะที่การควบคุมยังคงดำเนินอยู่ภายใน method รวมถึง
 - exception ที่ถูกส่งโดยตรงจาก method ด้วย throw statement ของ Java
 - exception ที่ถูกส่งโดยทางอ้อมโดย method โดยการเรียกไปยัง method อื่น

Catching and Handling with Exceptions

ในหัวข้อต่อไปนี้จะพูดถึงองค์ประกอบหลัก 3 อย่างของ exception handler คือ try , catch , final block ซึ่งทั้งหมดจะแสดงถึงการเขียน exception handler

- **Try block** อันดับแรกเราควรรู้ถึง statement ที่อาจส่ง exception เข้ามายัง try block ได้ โดย try block มีโครงสร้างดังนี้

```
try {
    Java Statements
}
```

Java Statements ประกอบด้วย statement ของ Java ที่สามารถส่ง exception ได้ try statement ครอบคลุม statement ที่อยู่ในนั้นและกำหนดขอบเขตของ exception handler ใดๆที่เกี่ยวข้องกับมัน หรือพูดอีกอย่างว่าถ้าเกิด exception ขึ้นภายใน try statement exception นั้นจะถูกจัดการโดย exception handler ที่เหมาะสมที่เกี่ยวข้องกับ try statement นั้น try statement จะต้องมาพร้อมทั้ง catch block หรือ finally block อย่างน้อย 1 ตัว

เอกสารนี้เป็นเอกสารที่เผยแพร่ฟรีสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Catch block** คู่กันเกี่ยวกับ exception handler ด้วย try block โดยจัดหา catch block หนึ่งหรือมากกว่าให้โดยตรงหลัง try block (ไม่มีโค้ดใดระหว่างท้าย try block กับ ต้น catch block)

```
try {
    ...
} catch (...) {
    ...
} catch (...) {
    ...
} ...
```

รูปแบบทั่วไปของ catch statement มีลักษณะดังนี้

```
catch (SomeThrowableObject variableName) {
    Java Statements
}
```

จะเห็นว่า catch statement ต้องการ argument หนึ่งตัว ชนิดของ argument (*SomeThrowableObject*) ประกาศชนิดของ exception ที่ exception handler สามารถจัดการและเป็นชื่อของ class ที่รับช่วงต่อจาก Throwable class (อยู่ใน package java.lang) เมื่อโปรแกรมจาวา throw exception ก็คือการ throw object และ object ที่รับช่วงต่อจาก Throwable class เท่านั้นที่สามารถ throw ได้ *variableName* เป็นชื่อที่ exception handler สามารถใช้อ้างถึง exception ซึ่ง catch โดย handler การเข้าถึง instance variable และ method เหมือนกับการเข้าถึง instance variable และ method ของ object อื่น

catch block จะเก็บกลุ่มของ statement ไว้ ซึ่ง statement เหล่านี้จะถูก execute เมื่อ exception handler ทำงาน โดย runtime system ให้ exception handler ทำงานเมื่อ handler เป็นตัวแรกใน call stack ซึ่งมีชนิดตรงกับ exception ที่ส่งมา

- **Finally Block** สุดท้ายเป็นการจัดขั้นตอนในการจัดการล้างสถานะของ method ก่อนที่จะผ่านการควบคุมไปยังส่วนอื่นของโปรแกรม ซึ่งทำได้โดยการใส่โค้ดที่ใช้ทำความสะอาดนี้ลงไปใน finally block runtime system จะ execute โค้ดใน finally block โดยขึ้นอยู่กับเหตุการณ์ที่เกิดขึ้นใน try block

How to throw an Exception

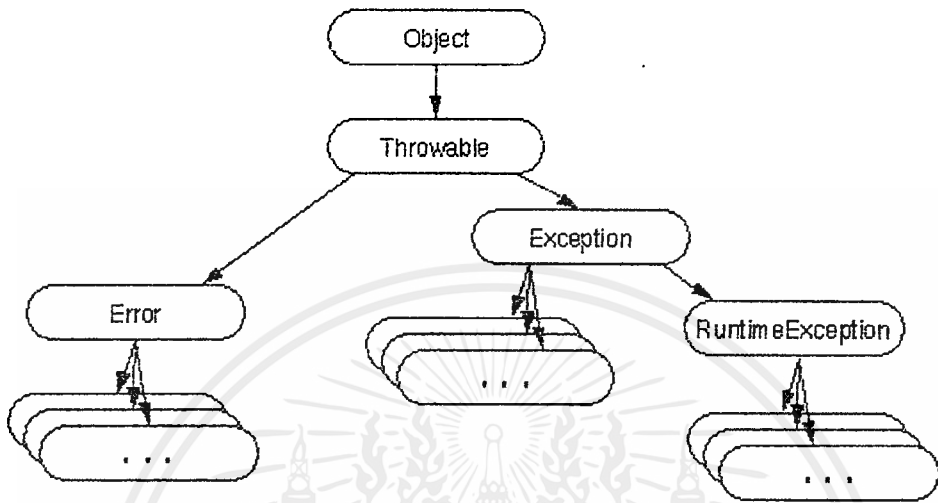
- **Throw statement** method ของจาวาทุกตัวจะใช้ throw ในการส่ง exception throw statement ต้องการ argument ตัวเดียวคือ : *throwable object* ในระบบของจาวา throwable object เป็น instance ของ subclass ใดๆของ Throwable class ตัวอย่างของ throwable statement แสดงดังต่อไปนี้

```
throw someThrowableObject ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

throw statement จะอยู่หลังชื่อของ method

- **Throwable class และ subclass ของมัน** รูปต่อไปนี้จะแสดงถึง class heirarchy ของ Throwable class และ subclass ที่สำคัญ



รูปที่ 2-5 class hierarchy ของ Throwable class

subclass ที่สำคัญของ throwable มี 3 ตัวด้วยกันคือ

1. **Errors** เมื่อเกิดข้อผิดพลาดขึ้นใน virtual machine virtual machine จะส่ง Error ซึ่งโปรแกรมโดยทั่วไปไม่สามารถ catch Error นี้ได้
2. **Exception** โปรแกรมส่วนมากจะ throw และ catch object ที่ได้มาจาก Exception class exception จะบอกว่าเกิดปัญหาขึ้นแต่ไม่ได้เป็นปัญหาของระบบที่สำคัญ subclass ของ Exception ที่สำคัญตัวหนึ่งคือ
3. **RuntimeException** เป็น class ซึ่งจะแสดงถึง exception ที่เกิดขึ้นใน Java virtual machine (ในช่วง runtime)

2.1.3.7 Threads of Control

Thread คืออะไร

thread มีลักษณะเหมือนกับโปรแกรมทั่วไป thread 1 ตัวจะมีจุดเริ่มต้น จุดสิ้นสุด โปรแกรม (เป็นเพียงตอนหรือช่วงหนึ่ง) และเวลาใดๆขณะที่ thread กำลังทำงานจะมีจุดประสงค์ของการ execute เพียงอย่างเดียว อย่างไรก็ตาม thread โดยตัวมันเองแล้วไม่ใช่โปรแกรม เพราะไม่สามารถทำงานด้วยตัวเองได้ แต่ต้องทำงานกับโปรแกรม

นิยาม thread เป็นการทำงานอย่างใดอย่างหนึ่งในโปรแกรม

การทำงานของ thread ที่จะพูดถึงไม่ใช่การทำงานของ thread ตัวเดียว แต่เป็นการทำงานแบบ multiple threads ภายในโปรแกรมเดียวกัน โดย thread จะทำงานพร้อมๆกันแต่จะทำหน้าที่คนละอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหนังสือบางเล่มจะใช้คำว่า *lightweight process* แทนคำว่า *thread* (เนื่องจาก *thread* มีลักษณะคล้าย *process* และที่เรียกว่า *lightweight* เนื่องจาก *thread* จะทำงานภายใต้สิ่งแวดล้อมเดียวกับโปรแกรม และใช้ *resource* ที่ถูกจองให้กับโปรแกรม) แต่อย่างไรก็ตาม *thread* จะต้องมี *resource* บางอย่างที่เป็นของตัวเอง เช่น *stack*, *program counter* เป็นต้น โค้ดที่อยู่ใน *thread* จะต้องทำงานภายใต้สิ่งแวดล้อมที่กล่าวมาเท่านั้น ฉะนั้นเราอาจเรียก *thread* อีกอย่างว่า *execution context*

Thread Attributes

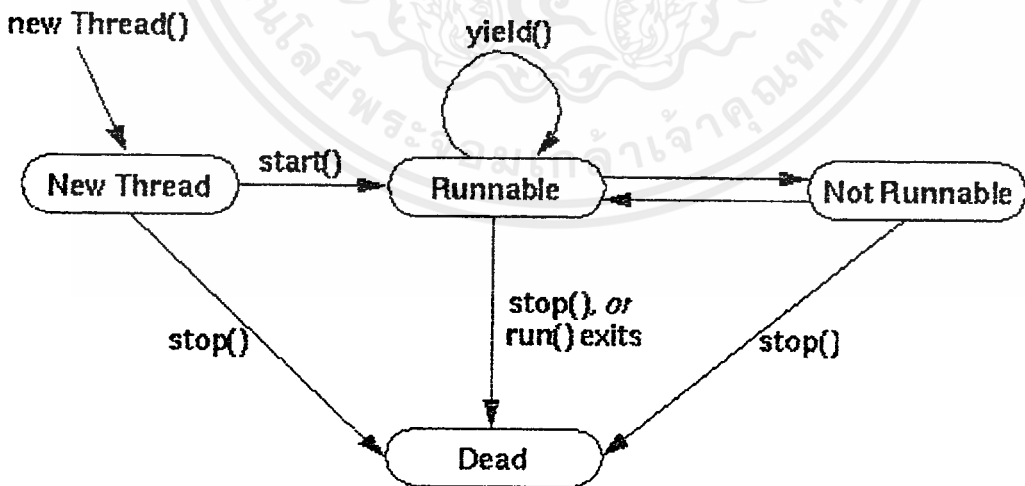
thread ของจาวาถูก implement โดย *Thread class* ซึ่งอยู่ใน package *java.lang* *thread class* จะ implement ข้อกำหนดที่เป็นอิสระจากระบบของ *thread*

Thread Body การทำงานทุกอย่างเกิดขึ้นใน *thread body* (ใน *run method*) หลังจากที่ *thread* ถูกสร้างและ *initial* แล้ว *runtime system* จะเรียก *run method* ของ *thread* นั้น โค้ดใน *run method* จะ implement การทำงานของ *thread* ที่สร้างขึ้น

บ่อยครั้งที่การทำงานของ *thread* จะเป็นการทำงานแบบ *loop* ทำให้บางครั้งการทำงานใช้เวลานาน คุณสามารถเลือกหนึ่งในสองวิธีต่อไปนี้ เพื่อลดขนาดของ *run method* สำหรับ *thread* ของจาวา

1. สร้าง subclass ของ *Thread class* ใน *java.lang* และ override *run method*
2. สร้างซึ่ง implement *Runnable interface* ซึ่งอยู่ใน package *java.lang* เช่นกัน ดังนั้นถ้าคุณสร้าง instance ของ *thread* (ทั้งทางตรงจาก *Thread class* และทางอ้อมจาก subclass ของ *Thread*) ให้สร้าง handle ของ instance ของ *Runnable class* ของคุณให้กับ *thread* ที่สร้างขึ้นใหม่ *runnable object* นี้จะเตรียม *run method* ให้กับ *thread*

Thread State รูปข้างล่างนี้แสดงสถานะของ *thread* ในจาวา



รูปที่ 2-6 สถานะของ *Thread* ในจาวา

- **New Thread**: statement ต่อไปนี้ใช้ในการสร้าง *thread* ใหม่ แต่ยังไม่มีการทำงานเกิดขึ้น

```
Thread myThread = new myThreadClass ();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเป็น thread ใหม่ มันจะเป็น thread object ที่ว่างเปล่า ยังไม่มีการจอง resource ของระบบให้ เมื่อ thread อยู่ในสถานะ new thread มันสามารถจะเริ่มหรือหยุด thread ได้เท่านั้น

- Runnable : พิจารณาได้คือไปนี้

```
Thread myThread = new myThreadClass ();
myThread . start ();
```

start method จะทำการจอง resource ที่จำเป็นในการทำงานของ thread ให้ จัดตารางการทำงานของ thread และเรียก run method ของ thread เมื่อถึงจุดนี้ thread จะอยู่ในสถานะ "Runnable" ที่ไม่เรียก running เนื่องจากยังไม่มีการทำงานเกิดขึ้น เนื่องจากคอมพิวเตอร์ส่วนใหญ่มี processor ตัวเดียว ทำให้ไม่สามารถ runnable thread ทุกตัวทำงานพร้อมกันได้ runtime system ของจาวาจะต้อง implement หลักการจัดตาราง ซึ่งจะแบ่งการใช้งาน processor ให้กับ runnable thread ทุกตัว

- Not Runnable : thread จะมาอยู่ที่สถานะ not runnable เมื่อเหตุการณ์ใดเหตุการณ์หนึ่งใน 4 เหตุการณ์นี้เกิดขึ้น

1. เมื่อมีการอ้าง sleep method ของมัน
2. เมื่อมีการอ้าง suspend method ของมัน
3. thread ใช้ wait method ของมันเอง เพื่อรอสถานะของ variable
4. thread ติดอยู่กับ I/O

การออกจากสถานะ not runnable (ไม่ว่าจะเข้ามาทางไหน) เกิดขึ้นเมื่อ

1. ถ้า thread เข้าสู่สถานะ sleep จะออกได้เมื่อเวลาที่กำหนดไว้หมดลง
2. ถ้า thread เข้าสู่สถานะ suspend จะออกได้เมื่อมีการเรียก resume method
3. ถ้า thread เข้าสู่สถานะ wait เพื่อรอสถานะ variable จะออกได้เมื่อ object ที่เป็นเจ้าของ variable ปลดอ้อมันโดยการเรียก notify หรือ notifyAll
4. ถ้า thread ติดอยู่กับ I/O จะออกได้เมื่อ I/O ทำงานเสร็จ

- Dead : thread สามารถสิ้นสุดได้ 2 วิธี คือ สิ้นสุดด้วยตัวเอง (ออกจาก run method ตามปกติ) หรือ ถูกทำให้สิ้นสุด(หยุด) ซึ่ง thread สามารถถูกทำให้หยุดได้ตลอดเวลาโดยการเรียก stop method stop method จะส่ง ThreadDeath object thread จะสิ้นสุดเมื่อมันได้รับ ThreadDeath exception

Thread Priority การจะให้ multithread ทำงานบน CPU ตัวเดียวนั้น จะต้องมีการทำ scheduling โดย runtime system ของจาวาจะมีขั้นตอนในการทำ scheduling เรียกว่า *fixed priority scheduling* ซึ่งขั้นตอนเหล่านี้จะจัด thread ตาม priority (ความสำคัญ) ซึ่งจะสัมพันธ์กับ thread อื่น

thread แต่ละตัวในจาวาจะถูกกำหนดค่าความสำคัญให้ซึ่งจะอยู่ระหว่าง MIN_PRIORITY และ MAX_PRIORITY (เป็นค่าคงที่ซึ่งกำหนดใน Thread class) ณ เวลาใดๆเมื่อ multithread พร้อมทั้งจะ

ทำงาน thread ที่มีความสำคัญมากที่สุดจะถูก execute ก่อน เมื่อ thread ที่ทำงานอยู่สิ้นสุดหรือ พักการทำงานเท่านั้น thread ที่มีความสำคัญน้อยกว่าถึงจะสามารถทำงานได้

scheduling ของ CPU จะมีสิทธิในการเลือก thread เดิมที ถ้า thread ที่มีความสำคัญมากกว่า thread ที่ทำงานอยู่ในขณะนั้น thread ที่มีความสำคัญมากกว่าจะถูกทำ schedule ในทันที

runtime system จะไม่ใช่สิทธิในการเรียกการทำงานคือเพื่อให้ thread อื่นที่มีความสำคัญเท่ากันทำงาน

ในขณะใดๆ thread อาจยกเลิกสิทธิในการทำงานโดยการเรียก yield method thread สามารถมอบ CPU ให้กับ thread อื่นที่มีความสำคัญเท่ากัน (ไม่สามารถให้กับ thread ที่มีความสำคัญต่ำกว่าได้)

เมื่อ runnable thread ในระบบทุกๆตัวมีความสำคัญเท่ากัน runtime system จะทำ scheduling โดยการเลือก thread ที่อยู่ถัดไป

thread ที่มี while loop อยู่ภายใน run method เมื่อทำงานมันจะไม่ปล่อยการควบคุม CPU ให้กับ thread อื่นจนกว่ามันจะสิ้นสุดด้วยตัวมันเอง (while loop สิ้นสุด) หรือจนกระทั่งถูกอ้างสิทธิ์จาก thread ที่มีความสำคัญมากกว่า thread เชนนี้เรียกว่า *selfish thread* ในบางระบบหาวิธีแก้ selfish thread ด้วยการทำ *time-slicing* time-slicing จะมีบทบาทเมื่อมี runnable thread หลายตัวที่มีความสำคัญเท่ากัน และ thread เหล่านั้นซึ่งเป็น thread ที่มีความสำคัญสูงสุดแข่งกันกันเพื่อแย่ง CPU โดยระบบ time-sliced จะแบ่ง CPU ออกเป็นเวลาออกเป็นช่วงๆและจะแบ่งให้กับ thread ทุกๆตัวที่มีความสำคัญเท่ากันและมีความสำคัญสูงสุดเป็นรอบๆ ในแต่ละรอบจะให้ช่วงเวลา thread ละ 1 ช่วงเพื่อให้ทำงานของมัน จะสังเกตได้ว่า time-sliced จะไม่รับประกันความถี่หรือลำดับการทำ schedule

Daemon Thread thread ในจาวาสามารถเป็น *daemon thread* ได้ daemon thread เป็น thread ที่ให้บริการ thread อื่นที่กำลังทำงานอยู่ใน process เดียวกัน run method ของ daemon thread ทำซ้ำไม่มีที่สิ้นสุดเพื่อรอการขอบริการจาก thread อื่น เมื่อ thread เดียวที่เหลือใน process เป็น daemon thread interpreter จะออกจากการทำงาน (เนื่องจากไม่มี thread อื่นที่ต้องบริการ)

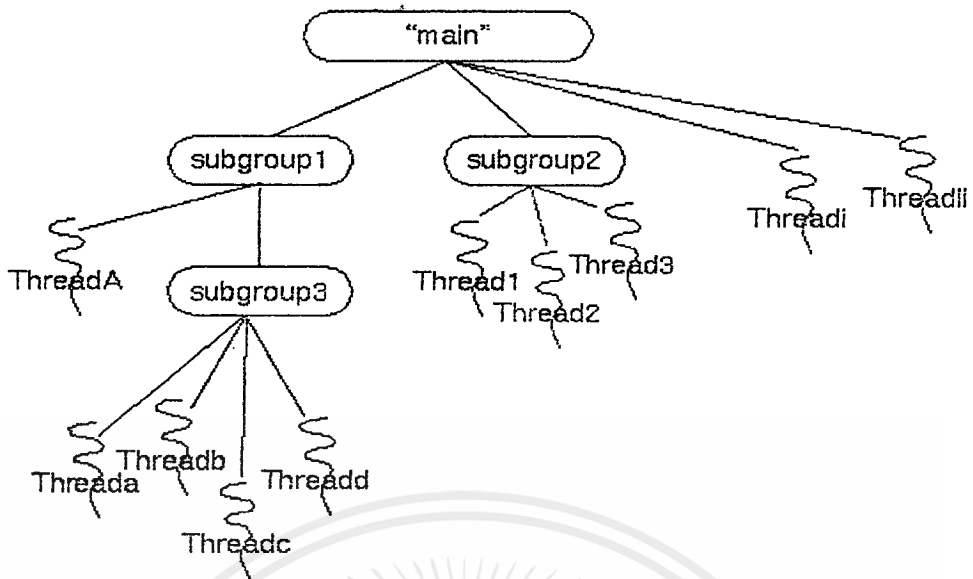
ในการกำหนดว่าให้ thread เป็น daemon thread ทำได้โดยการเรียก setDaemon method พร้อม argument "true" ส่วนการตรวจสอบว่า thread ใดเป็น daemon thread ทำได้โดยใช้ isDaemon method

Thread Group ทุก thread ในจาวาเป็นสมาชิกของ *thread group* thread group เป็นเครื่องมือในการรวม multithread เข้าด้วยกันเป็น object เดียวและใช้ thread เหล่านั้นทั้งหมดพร้อมกัน thread group ถูก implement ใน ThreadGroup class ใน package java.lang

runtime system จะทำรวม thread เข้า thread group ระหว่างที่ thread ถูกสร้าง เมื่อคุณสร้าง thread ใหม่ คุณสามารถให้ runtime system รวม thread ของคุณ เข้ากับ thread group ใดๆที่มีอยู่ หรือจะตั้ง thread group ใหม่ก็ได้ เมื่อ thread เป็นสมาชิกของ thread group ใดแล้วไม่สามารถจะย้าย thread group ได้

เมื่อโปรแกรมจาวาเริ่มทำงาน runtime system จะสร้าง thread group ชื่อว่า main ถ้าไม่ได้เจาะจงไว้ thread จะถูกรวมเข้าไว้ใน main thread group (แต่ถ้าคุณสร้าง thread ใน applet thread group ใหม่อาจใช้ชื่ออื่นที่ไม่ใช่ main)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-7 Thread Group class

Multithreaded Programs

Synchronizing Threads บ่อยครั้งที่ thread ต้องการใช้ข้อมูลร่วมกันและต้องพิจารณาถึงสถานะและกิจกรรมที่ thread อื่นทำอยู่ โดยกลุ่มของโปรแกรมที่อยู่ในสถานะการณ์นี้เรียกว่า producer / consumer โดย producer จะสร้างข้อมูล ซึ่งจะถูกรับไปใช้ ซึ่งทำให้การทำงานของทั้ง producer และ consumer เข้าใจหวั่นกัน (synchronize) ปัญหาของการทำงานไม่เข้าใจหวั่นกันเรียกว่า *race conditions* เกิดขึ้นเมื่อ multithread ที่ทำงานไม่เข้าใจหวั่นกันเข้าถึง object ตัวเดียวกันในเวลาเดียวกัน ทำให้ได้ผลลัพธ์ที่ผิดพลาด เพื่อป้องกัน *race conditions* ไม่ให้เกิดขึ้นสำหรับการทำงานแบบ producer / consumer เรามีวิธีแก้ 2 วิธีคือ monitors และ ใช้ notifyAll กับ wait method

1. **Monitors** : object ที่ถูกใช้ร่วมกันระหว่าง thread สองตัวซึ่งการเข้าถึงจะต้องเข้าใจหวั่นกัน เรียกว่า *condition variable* monitors จะเกี่ยวกับการกำหนดข้อมูล (condition variable) และการทำงานเช่น lock ในข้อมูลเหล่านั้น เมื่อ thread ทำ monitor บนข้อมูลตัวใด แล้วมี thread อื่นมาทำการ lock บนข้อมูลตัว จะทำให้ thread ที่ทำ monitor บนข้อมูลนั้น (thread ที่ไม่ได้เป็นผู้ lock) ไม่สามารถใช้ข้อมูลตัวนั้นได้ ส่วนของโค้ดที่อยู่ในโปรแกรมที่เข้าถึงข้อมูลตัวเดียวกันใน thread เรียกว่า *critical sections* ในจาวาคูสามารถกำหนด critical sections ในโปรแกรมของคุณได้โดยใช้ synchronized keyword
 2. **notifyAll และ wait methods** : method ทั้ง 2 ตัวนี้เป็นสมาชิกของ java.lang.Object class
- Note** : notifyAll และ wait method ถูกอ้างโดย thread ที่ถือ lock เท่านั้น

notifyAll method : ใช้ในการบอกให้กับ thread ตัวอื่นๆที่รอ บน monitor โดย thread ปัจจุบันให้คืนขึ้น หนึ่งใน thread ที่รออยู่ก็จะทำ monitor และทำงานต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wait method : จะทำให้ thread ปัจจุบันหยุดการทำงานจนกว่า thread อื่นจะบอกมันถึงสถานะที่เปลี่ยนไป

2.1.4 ข้อแตกต่างระหว่างแอปพลิเคชันกับแอปเพล็ต

ในการเขียนโปรแกรมด้วยภาษาจาวา สามารถเขียนได้ 2 แบบคือ เขียนเป็น Application หรือเขียนเป็น Applet ซึ่งทั้งคู่ก็เป็นโปรแกรมภาษาจาวาด้วยกันสามารถทำงานบนเครือข่ายได้เหมือนกัน แต่ข้อแตกต่างหลักก็คือ applet ต้องทำการโหลดมาจากเครื่อง Server และดำเนินการบนโปรแกรมบราวเซอร์ (Browser) ที่สนับสนุนการทำงานของจาวา ข้อแตกต่างอื่นๆ ได้แสดงไว้ดังตารางที่ 2-12

Application	Applet
สามารถรันบนเครื่อง client, server หรือ ทั้งคู่ได้	สามารถรันบนเครื่อง client ได้อย่างเดียว
ขึ้นอยู่กับ Java runtime environment	ต้องอาศัย Java-enable Web browser
สามารถ access ไฟล์บนเครื่อง client ได้	ไม่สามารถ access ไฟล์บนเครื่อง client ได้ถ้าถูก download มาจาก Web server
Software ต้องถูก install บนเครื่อง client	software ถูก download ขณะ runtime มายัง Web browser ซึ่งวิ่งอยู่บนเครื่อง client

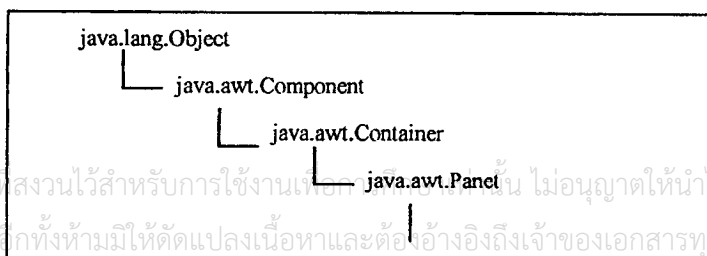
ตารางที่ 2-12 ข้อแตกต่างระหว่างแอปพลิเคชันกับแอปเพล็ต

2.1.5 การเขียนแอปเพล็ต

applet เป็นโปรแกรมที่ได้รับอนุญาตให้ทำงานได้ในโปรแกรมบราวเซอร์ ที่รองรับจาวา โดยมีขั้นตอนในการสร้าง applet อย่างคร่าวๆดังนี้

1. สร้างโค้ดของจาวา applet
 - 1.1 นำเอา class และ package เข้ามา
 - 1.2 กำหนด Applet subclass
 - 1.3 ทำการสร้าง methods ให้กับ applet
2. ทำการ compile โค้ดจาวาที่เขียนเสร็จแล้ว
3. สร้างไฟล์ HTML ที่รวมเอา applet ที่สร้างขึ้นไว้ข้างใน
4. นำ applet ไปใช้งาน โดยการเรียกใช้ไฟล์ HTML

applet ทุกตัวจะถูก implement โดยการสร้าง subclass ของ Applet class รูปต่อไปนี้จะแสดงถึง heirarchy ของ Applet class



___ java.applet.Applet

รูปที่ 2-8 hierarchy ของ Applet class

2.1.5.1 วงจรชีวิตของแอพเพล็ต

1. **Loading and applet** เมื่อเราทำการ load applet จะเกิดเหตุการณ์ต่อไปนี้

- subclass ของ Applet class จะถูกสร้างขึ้น
- applet จะ initial ตัวเอง
- applet จะเริ่มทำงาน

2. **เข้าและออกจากหน้าของ applet** เมื่อเราออกจากหน้าที่ applet ทำงานอยู่ไปยังหน้าอื่น applet จะหยุดทำงาน และเมื่อเรากลับมายังหน้าของ applet อีกครั้ง applet ก็จะเริ่มทำงานใหม่

3. **ทำการ load applet ใหม่** โปรแกรมบราวเซอร์บางตัวให้ผู้ใช้สามารถ reload applet ได้ โดยจะ unload applet แล้วค่อยทำการ load ใหม่ ก่อนที่ applet จะ unload มันจะหยุดตัวเองก่อน แล้วทำการ *final cleanup* ซึ่ง applet จะทำการปล่อย resource ที่มันถืออยู่ทั้งหมด จากนั้นจึงทำการ load applet อีกครั้ง

4. **ออกจากโปรแกรมบราวเซอร์** เมื่อผู้ใช้ออกจากโปรแกรมบราวเซอร์ applet จะหยุดการทำงานและทำการ final cleanup ก่อนที่จะออกจากโปรแกรมบราวเซอร์

2.1.5.2 Method ที่ใช้สำหรับตรวจสอบ

```
public class Simple extends Applet {
    . . .
    public void init () { . . . }
    public void start () { . . . }
    public void stop () { . . . }
    public void destroy () { . . . }
    . . .
}
```

ใน applet ได้มีการ override method อยู่ 4 ตัวด้วย ซึ่งมันจะตอบสนองต่อเหตุการณ์ต่างกัน

init : ใช้ในการ initial applet ในแต่ละครั้งที่มันถูก load (หรือ reload)

start : จะถูกใช้เมื่อ applet ถูก load แล้ว หรือเมื่อผู้ใช้กลับเข้ามายังหน้าที่มี applet อยู่อีกครั้ง

stop : จะถูกใช้เมื่อผู้ใช้ออกจากหน้าที่มี applet อยู่ หรือออกจากโปรแกรมบราวเซอร์

destroy : ใช้เพื่อทำการ final cleanup สำหรับ unload

ในการเขียน applet ไม่จำเป็นต้อง override method ที่กล่าวมาก็ได้ init method จะมีประโยชน์สำหรับการ initial ครั้งแรกที่ไม่ต้องใช้เวลาานาน โดยทั่วไป init method จะเก็บโค้ดที่คุณจะใส่เข้าไปใน constructor เพราะ applet ที่ไม่มี constructor ใน applet จะไม่รับประกันว่าจะมีสภาพแวดล้อมที่พร้อมจนกว่า init method จะถูกเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

applet ทุกตัวที่ทำงานบางอย่างหลังจาก initial แล้ว (ยกเว้นรอรับการตอบสนองจากผู้ใช้) จะต้อง override start method start method จะทำงานของ applet หรือเริ่มการทำงานของ thread

applet ที่ override start method มักจะ override stop method ด้วย stop method จะรวมอยู่ในการทำงานของ applet ด้วย มันจะไม่ใช่ resource ของ applet ถ้าผู้ใช้ไม่ได้ดูหน้าที่มี applet อยู่

ส่วน destrop method ไม่จำเป็นจะต้อง override ถ้า stop method ทำสิ่งที่จำเป็นในการหยุดการทำงาน of applet แล้ว อย่างไรก็ตาม destroy method ก็ยังจำเป็นสำหรับ applet ที่ต้องการปล่อย resource เพิ่มเติม

2.1.5.3 เมธอดสำหรับ Drawing และ Event Handling

Simple applet ในตัวอย่างข้างล่างนี้เป็นการ override paint method :

```
class Simple extends Applet {
    . . .
    public void Paint (Graphics g) { . . . }
    . . .
}
```

paint method เป็น หนึ่งในสองวิธีที่ใช้ในการแสดงผล ซึ่ง applet สามารถ override ได้

paint : เป็น method ที่ใช้แสดงผลพื้นฐาน applet หลายตัว implement paint method เพื่อใช้เพื่อใช้แสดงผลของ applet ในหน้าของโปรแกรมบราวเซอร์

update : เป็น method ที่คุณสามารถใช้พร้อมกับ paint method เพื่อปรับปรุงการแสดงผล

applet จะรับช่วงต่อ paint และ update method มาจาก Applet class ซึ่งรับช่วงต่อจาก Abstract Windows Toolkit (AWT) component class อื่นๆ

applet จะรับช่วงต่อกลุ่มของ event-handling methods มาจาก Component class ซึ่งจะมี method ที่ใช้รองรับเหตุการณ์ต่างๆที่เกิดขึ้นเช่น mouseDown , mouseUp , action เป็นต้น หรือ handleEvent method ซึ่งจะใช้ในการรองรับเหตุการณ์ที่เกิดขึ้นทั้งหมดด้วย method เดียว ในการตอบสนองต่อเหตุการณ์ที่เกิดขึ้น applet จะต้อง override method ที่ใช้รองรับเหตุการณ์ที่เหมาะสม หรือ handleEvent method

2.1.5.4 เมธอดที่ใช้ในการเพิ่ม UI Component

เป็นการช่วยเพิ่มส่วนแสดงผลที่มีประสิทธิภาพมากขึ้นให้กับ applet

UI Components ที่มีมาให้

AWT จะจัดหา UI Components ต่อไปนี้ไว้ให้ (class ซึ่ง implement components แต่ละตัวจะแสดงไว้ในวงเล็บ)

- Buttons (java.awt.Button)
- Checkboxes (java.awt.Checkbox)
- Single-line text fields (java.awt.TextField)
- Larger text display and editing areas (java.awt.TextArea)

- Labels (java.awt.Label)
- Lists (java.awt.List)
- Pop-up list of choices (java.awt.Choice)
- Sliders and Scrollbars (java.awt.Scrollbar)
- Drawing areas (java.awt.Canvas)
- Menus (java.awt.Menu , java.awt.MenuItem , java.awt.CheckboxMenuItem)
- Containers (java.awt.Panel , java.awt.Window และ subclass ของมัน)

เมธอดสำหรับใช้ UI Components ในแอปเพล็ต

เนื่องจาก Applet class รับช่วงต่อจาก AWT Container class ทำให้ง่ายในการเพิ่ม component ลงไปใน applet และใช้ layout manager ในการจัดการตำแหน่งของ component บนจอภาพ ต่อไปนี้เป็น method ของ Container ที่ applet สามารถใช้ได้

add : ใช้เพิ่ม component ที่กำหนดไว้
 remove : ใช้ย้าย component ที่กำหนดไว้
 setLayout : ใช้กำหนด layout manager

2.1.5.5 ข้อจำกัดและความสามารถของ แอปเพล็ต

ข้อจำกัดทางด้านความปลอดภัย

โปรแกรมบราวเซอร์ทุกตัวจะมีการ implement ถึงความปลอดภัย เพื่อให้ความปลอดภัยกับ applet อย่างไรก็ตามศักยภาพความปลอดภัยของโปรแกรมบราวเซอร์แต่ละตัวจะแตกต่างกันไป

โปรแกรมบราวเซอร์ในปัจจุบันได้มีการกำหนดข้อจำกัดสำหรับ applet ใดๆที่ load ผ่านเครือข่ายมา

- applet ไม่สามารถ load library หรือ กำหนด native method ได้
- applet ไม่สามารถอ่านหรือเขียนไฟล์บน host ที่กำลัง execute มันได้
- applet ไม่สามารถสร้างการเชื่อมต่อผ่านเครือข่ายยกเว้น ไปยัง host ที่มันมา
- applet ไม่สามารถเรียกใช้งาน โปรแกรมอื่นๆบน host ที่กำลัง execute มันได้
- applet ไม่สามารถอ่านคุณสมบัติของระบบได้
- window ที่ applet สร้างจะต่างกับ window ที่โปรแกรมสร้างขึ้น

โปรแกรมบราวเซอร์แต่ละตัวจะมี SecurityManager object ซึ่ง implement อยู่ในศักยภาพความปลอดภัย เมื่อ SecurityManager object จับได้ว่ามีการละเมิดข้อบังคับ มันจะส่งต่อไปยัง SecurityException applet ของคุณสามารถตรวจจับ SecurityException นี้ได้และจะตอบสนองไปอย่างเหมาะสม

ความสามารถของ applet

java.applet package จัดหา API ซึ่งจะเพิ่มความสามารถของ applet ที่โปรแกรมสำเร็จรูปทั่วไป ไม่มี เช่น applet สามารถส่งเสียงได้ โปรแกรมบราวเซอร์และ applet viewer อื่นๆ ในปัจจุบันจะอนุญาตให้ applet ทำสิ่งต่างๆต่อไปนี้ได้

- applet สามารถสร้างการติดต่อข้างเครือข่ายไปยัง host ที่มันมาได้
- applet สามารถเข้าถึง public method ของ applet อื่นๆ ในหน้าเดียวกันได้
- applet ที่กำลังทำงานอยู่ในโปรแกรมบราวเซอร์ สามารถแสดงผลได้ง่ายโดย HTML ไฟล์
- applet ที่ load จาก local file system (จากไดเรกทอรีใน CLASSPATH ของผู้ใช้) จะไม่มีข้อจำกัดเหมือนกับ applet ที่ถูก load ข้ามเครือข่ายมา

2.1.5.6 การเพิ่ม แอปเพล็ตเข้าไปในหน้าของ HTML

การทำให้ applet ทำงานได้ในโปรแกรมบราวเซอร์หรือใน JDL applet viewer applet จำเป็นจะต้องรวมเข้ากับหน้าของ HTML โดยใช้ tag <APPLET> จากนั้นก็กำหนด URL ของหน้าของ HTML ให้กับโปรแกรมบราวเซอร์ของคุณหรือ applet viewer

<APPLET> Tag

ข้างล่างนี้จะแสดงถึงการกำหนดลักษณะที่ applet จะถูกแสดงออกทางหน้าของ HTML โดยใช้ <APPLET> tag ตัวหนาเป็นคำที่คุณจะต้องพิมพ์ลงไป (ไม่จำเป็นต้องเป็นตัวพิมพ์ใหญ่) ส่วนตัวเอียงเป็นสิ่งที่你必须เอาค่ามาใส่เอง เครื่องหมาย “[]” บอกว่าคุณจะพิมพ์ลงไปหรือไม่ก็ได้ ส่วนที่ไม่มีเครื่องหมาย “[]” หมายความว่าจำเป็นต้องพิมพ์ลงไปด้วย

<APPLET

[CODEBASE = *codebaseURL*]

CODE = *appletFile*

[ALT = *alternateText*]

[NAME = *appletInstanceName*]

WIDTH = *pixels*

HEIGHT = *pixels*

[ALIGN = *alignment*]

[VSPACE = *pixels*]

[HSPACE = *pixels*]

>

[< PARAM NAME = *appletParameter1* VALUE = *value* >]

[< PARAM NAME = *appletParameter2* VALUE = *value* >]

...

[*alternateHTML*]

</APPLET>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODEBASE = *codebaseURL*

กำหนด URL พื้นฐานของ applet (เป็นไคลเรกทอรีที่เก็บโค้ดของ applet ไว้ ถ้าไม่กำหนดไว้จะใช้ไคลเรกทอรีที่ HTML ไฟล์อยู่

CODE = *appletFile*

เป็นคุณสมบัติที่จำเป็นซึ่งจะเก็บชื่อไฟล์ของ applet ที่ compile แล้ว ไฟล์นี้จะต้องสัมพันธ์กับ URL พื้นฐานของ applet

ALT = *alternateText*

จะกำหนดข้อความที่จะถูกแสดงเมื่อโปรแกรมบราวเซอร์เข้าใจ <APPLET> tag แต่ไม่สามารถให้ applet ทำงานได้

NAME = *appletInstanceName*

จะกำหนดชื่อของ applet instance เพื่อให้ applet อื่น ในหน้าเดียวกันสามารถหาหรือติดต่อกันได้

WIDTH = *pixels*

HEIGHT = *pixels*

เป็นคุณสมบัติที่จำเป็นต้องกำหนดเพื่อกำหนดขนาดที่จะใช้ในการแสดง applet

ALIGN = *alignment*

ใช้ในการกำหนดตำแหน่งของ applet โดยค่าที่เป็นไปได้คือ left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom

VSPACE = *pixels*

HSPACE = *pixels*

กำหนดจำนวนจุด (pixel) ทั้งบนและล่าง (VSPACE) ของ applet และทั้งซ้ายและขวา (HSPACE) ของ applet

<PARAM NAME = *appletParameter1* VALUE = *value* >

<PARAM> tag เป็นวิธีเดียวในการกำหนด parameter ของ applet applet จะอ่านค่า parameter ที่ผู้ใช้กำหนดโดยใช้ `getParameter()` method

alternateHTML

ถ้าหน้า HTML มี <APPLET> tag ซึ่งแสดงโดยโปรแกรมบราวเซอร์ที่ไม่เข้าใจ <APPLET> tag นี้ โปรแกรมบราวเซอร์จะไม่สนใจ <APPLET> และ <PARAM> tag และจะแสดงข้อความที่อยู่ระหว่าง <APPLET> และ </APPLET> tag แทน

2.1.5.7 การสิ้นสุดการทำแอปเพล็ต

ก่อนนำ applet ออกใช้งานจริง

ก่อนที่จะนำ applet ที่คุณสร้างขึ้นไปใช้งานจริง คุณความแน่ใจก่อนว่าคุณสามารถทำคำถามเหล่านี้ให้เป็นจริงได้

1. คุณได้นำเอาจุดบกพร่องเกี่ยวกับการแสดงผลออกไปหมดหรือยัง ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อผิดพลาดเกี่ยวกับการแสดงผลเช่น การใช้ System.out.println ซึ่งสร้างสะดวกให้กับคุณแต่อาจสร้างความสับสนหรือรบกวนกับผู้ใช้ ถ้าคุณต้องการแสดงข้อความให้กับผู้ใช้ทราบ พยายามแสดงอยู่ภายในส่วนแสดงผลของ applet หรือในแถบแสดงสถานะบริเวณด้านล่างของหน้าต่างโดยใช้ showStatus method

2. applet ของคุณหยุดการทำงานเมื่อออกนอกจอภาพหรือไม่ ?

applet ไม่ควรใช้ resource เมื่อโปรแกรมบราวเซอร์แสดงหน้าต่างที่ไม่มี applet นั้นบรรจุอยู่

ถ้าโค้ด applet ของคุณมี thread อยู่ คุณควรจะมี implement stop method (ซึ่งจะหยุดการทำงานของ thread โดยการกำหนดให้เป็น null)

3. ถ้า applet ของคุณทำสิ่งที่อาจเป็นการรบกวนผู้ใช้ คุณให้ผู้ใช้สามารถหยุดสิ่งรบกวนเหล่านั้นได้หรือไม่ ?

คุณควรให้ทางเลือกกับผู้ใช้ในการหยุดการทำงานสิ่งที่อาจรบกวนผู้ใช้ได้ โดยผู้ใช้ไม่จำเป็นต้องออกจากหน้านั้น

การจบ applet ที่สมบูรณ์

สิ่งที่จะพูดถึงต่อไปนี้เป็นข้อแนะนำในการทำให้ applet ของคุณน่าสนใจมากยิ่งขึ้น

1. พยายามให้ applet ของคุณมีความยืดหยุ่นมากที่สุด

โดยการกำหนด parameter เพื่อให้ applet ของคุณสามารถใช้ได้ทุกสถานการณ์ โดยไม่ต้องทำการเขียนใหม่

2. ทำการ implement getParameterInfo method

การ implement method นี้จะทำให้ applet ของคุณสามารถพัฒนาได้ง่ายในอนาคต ในปัจจุบันนี้ไม่มีโปรแกรมบราวเซอร์ตัวใดใช้ method นี้ ถึงแม้ว่าอย่างไรก็ตามในอนาคตอาจมีการใช้เพิ่มขึ้น เพื่อช่วยสร้าง GUI ซึ่งอนุญาตให้ผู้ใช้สามารถติดต่อกับ parameter เพื่อกำหนดค่าให้

3. ทำการ implement getMethodInfo method

method นี้จะส่งข้อมูลสั้นๆเกี่ยวกับ applet มาให้ ซึ่งปัจจุบันนี้ก็ยังไม่มีการบราวเซอร์ตัวใดใช้ method นี้

2.2 JDBC (Java Database Connectivity)

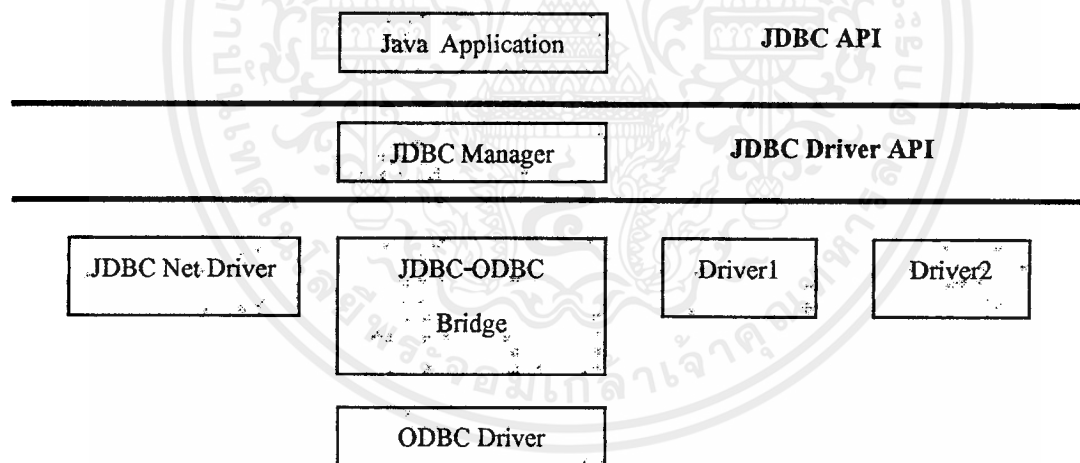
2.2.1 JDBC คืออะไร

JDBC หรือ Java Database Connection เป็นมาตรฐานมราช่วยในการติดต่อ ระหว่างตัวจาวากับระบบจัดการฐานข้อมูล (DBMS : Database Management System) โดยตัวที่ใช้ทำงานจริงๆระหว่างจาวากับระบบจัดการฐานข้อมูลก็คือ (JDBC API : Java Database Connectivity Application Programming Interface) สำหรับใช้ในการ execute คำสั่งเอสคิวแอล ซึ่งประกอบไปด้วยคลาสและอินเตอร์เฟสที่เขียนด้วยภาษาจาวา เจดีบีซีจะจัดหา API มาตรฐานสำหรับผู้ทำการพัฒนา tool หรือ ฐานข้อมูล และทำให้เป็นไปได้ในการเขียนแอปพลิเคชันเกี่ยวกับฐานข้อมูลโดยใช้ภาษาจาวา

2.2.2 เลเยอร์ของ JDBC

การทำงานจะอยู่ที่ 2 เลเยอร์หลักได้แก่

1. เจดีบีซีเอพีไอ ซึ่งจะทำการติดต่อระหว่างแอปพลิเคชันหรือแอปพลิเคชันใดๆ ที่เขียนขึ้นด้วยภาษาจาวากับเจดีบีซีเมนเนเจอร์
2. เจดีบีซีไดรฟ์เวอร์เอพีไอ จะทำการติดต่อระหว่างเจดีบีซีเมนเนเจอร์กับไดรฟ์เวอร์ของระบบฐานข้อมูลที่ติดต่อด้วยผังรูป



รูปที่ 2-9 แสดงโมเดลชั้นการทำงานของเจดีบีซี

เราจะเห็นได้ว่า 2 เลเยอร์หลัก จะมีองค์ประกอบที่จำเป็นของการทำงานของเจดีบีซี คือชั้นเจดีบีซีเอพีไอ จะมีแอปพลิเคชันหรือแอปพลิเคชัน และชั้นเจดีบีซีไดรฟ์เวอร์เอพีไอก็จะมีเจดีบีซีเมนเนเจอร์ ซึ่งทำหน้าที่ในการเป็นเจดีบีซีไดรฟ์เวอร์เอพีไอ เพื่อให้แอปพลิเคชันหรือแอปพลิเคชันในชั้นบนสามารถติดต่อกับไดรฟ์เวอร์ของระบบฐานข้อมูลได้

ในชั้นล่างสุดเราจะเห็นไดรฟ์เวอร์ต่างๆที่เจดีบีซี จะอาศัยเจดีบีซีเมนเนเจอร์ในการติดต่อเข้ากับแอปพลิเคชันหรือแอปพลิเคชัน และยังมีเจดีบีซี-โอดีบีซีบริดจ์ (JDBC - ODBC Bridge) ซึ่งเป็นตัวเชื่อมเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตเห็นไปใช้ประโยชน์ในการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

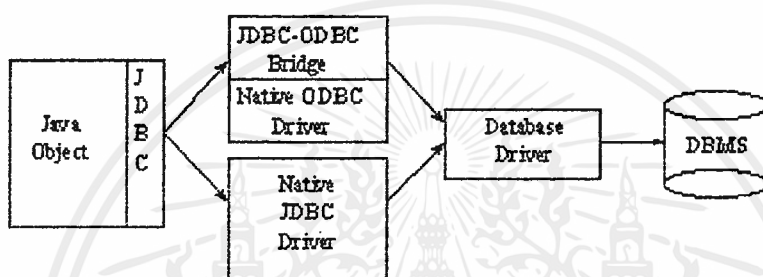
ระหว่างแอปพลิเคชันหรือแอปพลิเคชันใดๆ ที่ต้องการติดต่อกับฐานข้อมูลที่สนับสนุนมาตรฐานของไครีฟเวอร์โอดีบีซี โดยการทำงานก็เช่นเดียวกับการติดต่อกับไครีฟเวอร์อื่นๆ ไป

2.2.3 การทำงานของเจดีบีซี

เจดีบีซีจะทำหน้าที่หลัก 3 อย่างด้วยกันคือ

1. สร้างจุดเชื่อมต่อกับฐานข้อมูลที่ต้องการติดต่อกับ
2. ส่งคำสั่งเอสคิวแอล (SQL Statement) ไปยังฐานข้อมูลนั้น
3. รับผลจากคำสั่งที่ส่งไปในข้อ 2.

รูปแบบที่ใช้ใน JDBC จะเป็นดังรูปข้างล่างนี้



รูปที่ 2-10 แสดงรูปแบบที่ใช้ใน JDBC

ในรูปแบบนี้จะสังเกตเห็นได้ว่า Java object จะร้องขอไปยัง JDBC classes ซึ่งจะเป็นตัวเรียกไปยังตัว JDBC driver หรือตัว JDBC-ODBC bridge ที่กำหนด จากนั้นตัว driver จะเรียกตัว driver ของฐานข้อมูลจริง ๆ เพื่อที่จะทำการเชื่อมต่อไปยังฐานข้อมูลให้สมบูรณ์

ขั้นตอนในการเข้าถึง relational database โดยผ่าน JDBC มีดังนี้

1. ทำการติดตั้ง JDBC package และตัว driver ของฐานข้อมูลที่ต้องการ
2. ทำการเปิดการเชื่อมต่อกับฐานข้อมูล
3. ทำการสร้าง SQL statement หรือกระบวนการเรียกใช้
4. ทำการปฏิบัติตาม SQL statement หรือกระบวนการเรียกใช้
5. ทำการประมวลผลที่ได้จาก SQL statement หรือกระบวนการเรียกใช้
6. ทำการ commit หรือ rollback งาน transaction (ถ้าจำเป็น)

2.2.3.1 การเชื่อมต่อไปยังฐานข้อมูล

การเชื่อมต่อไปยังฐานข้อมูลจะทำผ่าน DriverManager class และการเชื่อมต่อ interface โดยทำตามขั้นตอนดังต่อไปนี้

การร้องขอ JDBC package:

```
import java.sql.*;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการ load ตัว driver ของฐานข้อมูลที่เหมาะสม:

```
// Use the JDBC-ODBC bridge as an example
Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
(Throws a ClassNotFoundException error.)
```

การสร้างฐานข้อมูล URL:

```
// The database URL takes the form jdbc:subprotocol:subname
String url = "jdbc:odbc:dsn"
```

การกำหนด USER ID และ Password:

```
String user = "guest";
String password = "guest";
```

การเปิดการเชื่อมต่อ:

```
// Call the Driver Manager to open a database connection
Connection con = DriverManager.getConnection (url, user, password);
(Throws a SQLException error.)
```

การ Settings อื่น ๆ:

```
// Transactions are automatically committed when executed
con.setAutoCommit (true);
// Put connection in read-only mode
con.setReadOnly (true).
(Throws a SQLException error.)
```

การปิดการเชื่อมต่อ:

```
con.close ();
(Throws a SQLException error.)
```

2.2.3.2 SQL Statement

SQL queries ถูกแสดงผ่านการ interface Statement และ ResultSet

การสร้าง Statement Object:

```
Statement stmt = con.createStatement ();
(Throws a SQLException error.)
```

การ Set Query String:

```
String query = "SELECT * FROM TABLE_NAME";
```

การปฏิบัติตาม Query และการสร้าง Result Set Object:

```
ResultSet rs = stmt.executeQuery (query);
(Throws a SQLException error.)
```

2.2.3.3 กระบวนการทำ Result Set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลของ SQL query ถูกนำออกผ่าน ResultSet object โดยใน object จะมีวิธีในการจัดการกับ rows และ columns จาก query

การอ่าน row ถัดไป:

```
boolean found = rs.next () ;           // returns true if row is read, false if end
of result set

(Throws a SQLException error.)
```

การ extract column values:

```
/** Issue a getXxxx() call for each column, base on its data type.

* Common data types are:

* INTEGER                getInt () or getBigDecimal (,0)
* NUMBER                 getBigDecimal ()
* VARCHAR                getString ()
* LONGVARCHAR            getAsciiStream ()
* TIMESTAMP              getDate (), getTime (), or getTimestamp ()

* Columns may be accessed through a relative position within the row (e.g.
column 1, 2, 3, ...) or through its actual column name.

* SQL NULL values will result in the Java null value being assigned to the
Java variable.

* /

int    personKey = rs.getInt (1);
String lastName  = rs.getString ("LAST_NAME");
String firstName = rs.getString ("FIRST_NAME");
Date   birthDate = rs.getDate (4);

(Throws a SQLException error.)
```

การปิด Result Set:

```
// Close the result set object after row processing is complete
rs.close ();

(Throws a SQLException error.)
```

การปิด Statement:

```
// Close the statement object when row processing is complete
stmt.close ();

(Throws a SQLException error.)
```

2.2.3.4 การ Update Transaction

SQL INSERT, UPDATE, หรือ DELETE คล้ายคลึงกับ QUERY ทั่ว ๆ ไป แตกต่างก็เพียงแค่ชื่อวิธีเรียก และชนิดของผลที่จะได้รับ

การสร้าง Statement Object:

```
Statement stmt = con.createStatement ();
(Throws a SQLException error.)
```

การ Set คำสั่งที่ Update:

```
String query = "INSERT INTO TABLE_NAME VALUES
(1, 'SMITH', 'JOHN', NULL)";
```

การ Execute Update:

```
int rowsUpdated = stmt.executeUpdate (query);
// returns number of rows updated
(Throws a SQLException error.)
```

การ Commit Transaction:

```
con.commit ();
(Throws a SQLException error.)
```

การ Rollback Transaction:

```
con.rollback ();
(Throws a SQLException error.)
```

การปิด Statement Object:

```
// Close the statement object after update is complete
stmt.close ();
(Throws a SQLException error.)
```

2.2.3.5 การเรียกกระบวนการเก็บ

การเรียกกระบวนการเก็บและ functions ต่าง ๆ จะเป็น bit มากกว่า SQL queries และการ updates โดยใช้ CallableStatement interface ซึ่งจะต่อจาก PreparedStatement interface โดยตัวแปรที่ใช้จะเป็น placeholder symbols เช่น INOUT ใช้ data types จาก Types class

การสร้างคำสั่งด้วย placeholders:

```
// Stored procedure call
String command = "{call PROCEDURENAME (?, ?, ?, ?)}";
// 4 placeholders
// Stored function call
String command = "{? = call FUNCTIONNAME (?, ?, ?, ?)}";
// 4 placeholders + 1 return value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง Callable Statement Object:

```
CallableStatement cstmt = con.prepareCall (command);\n\n(Throws a SQLException error.)
```

Register the IN and OUT Placeholders:

```
// Assign In parameters (use positional placement)\ncstmt.setInt (1, 50);           // first placeholder has an integer value of 50\n\nncstmt.setString (2, "Smith");\n\n    // second placeholder has a string value of "Smith"\nncstmt.registerOutParameter (3, Types.NUMERIC);\n\n    // NUMERIC is preferred over INTEGER with some databases\nncstmt.registerOutParameter (4, Types.VARCHAR);\n\n    // fourth placeholder is an OUT field of type VARCHAR.\n\n(Throws a SQLException error.)
```

หมายเหตุ: INOUT fields may be registered using the setXXXX() and registerOutParameter() methods for the same placeholder position. Array parameters (i.e. PL/SQL Tables) used in Oracle stored procedure and function calls are not supported by the JDBC.

การ Execute กระบวนการและ function การเรียก:

```
cstmt.execute () ;\n\n(Throws a SQLException error.)
```

กระบวนการ OUT Placeholders:

```
/** Use proper getXXXX() method based on parameter data types.\n\n * Placeholder 3 will receive a NUMERIC value as a BigDecimal object with\n * 0 decimal places.\n\n * Placeholder 4 will receive a VARCHAR value as a String object.\n\n */\n\nBigDecimal num = cstmt.getBigDecimal (3,0);\nString str = cstmt.getString(4);\n\n(Throws a SQLException error.)
```

การปิด CallableStatement Object:

```
// Close the call object after update is complete\nncstmt.close ();\n\n(Throws a SQLException error.)
```

2.2.3.6 การเข้าถึง Metadata

Metadata เป็นข้อมูลที่อธิบายเกี่ยวกับตัว โครงสร้างข้อมูล และการเชื่อมต่อภายในฐานข้อมูล ตัวอย่างของ metadata จะรวมถึง:

- ใครเป็นผู้ติดต่อ ไปยังฐานข้อมูล
- ฐานข้อมูลนี้เป็น read-only หรือ updatable
- data types ที่เหมาะสมสำหรับฐานข้อมูลนี้คืออะไร
- data types ของ column นี้คืออะไร
- vendor ของฐานข้อมูลเกี่ยวข้องกับกระบวนการเก็บอย่างไร
- database handle cursors ทำ rollbacks และ commits อย่างไร

เพิ่มเติมคือ metadata จะจัดหาโดยผลจาก query ตัวอย่างของ meta ที่สัมพันธ์กับ query จะรวมถึง

- มี columns กี่ columns ที่ถูกนำออกในแต่ละ row ของผลที่ได้
- ชื่อของ column และชนิดของ SQL คืออะไร
- ชื่อของ table ของ column และ table schema คืออะไร
- ใน 1 column มีตัวเลขของฐานลึบอยู่เท่าไร

ในการเข้าถึง metadata จะทำผ่าน interface 2 objects คือ DatabaseMetaData และ

ResultSetMetaData

```
DatabaseMetaData dmeta = con.getMetaData ();
```

```
ResultSetMetaData rsmeta = rs.getMetaData ();
```

ตัวอย่าง

ตัวอย่าง โปรแกรม Sample.java จะสาธิตการใช้ JDBC โดยใช้ Java application เพียงอย่างเดียว

ในการ compile โปรแกรมให้ทำดังนี้:

```
javac Sample.java
```

ทำการ run โปรแกรม และสังเกตผลที่ได้:

```
java Sample
```

2.2.4 เปรียบเทียบ CGI และ JDBC

ความพิเศษของการเลือกใช้ภาษา Java

การใช้ ภาษา Java สามารถทำการเข้าถึงข้อมูลได้ เนื่องจากภาษา Java เป็นภาษาที่มีคุณลักษณะ ต่างๆดังนี้

- มี performance
- เป็นแบบ database and platform independence
- มี scalability
- มีความ simplicity
- มีความ compatibility

ในการเข้าถึงฐานข้อมูลบน Web สามารถทำได้ 2 ทางคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

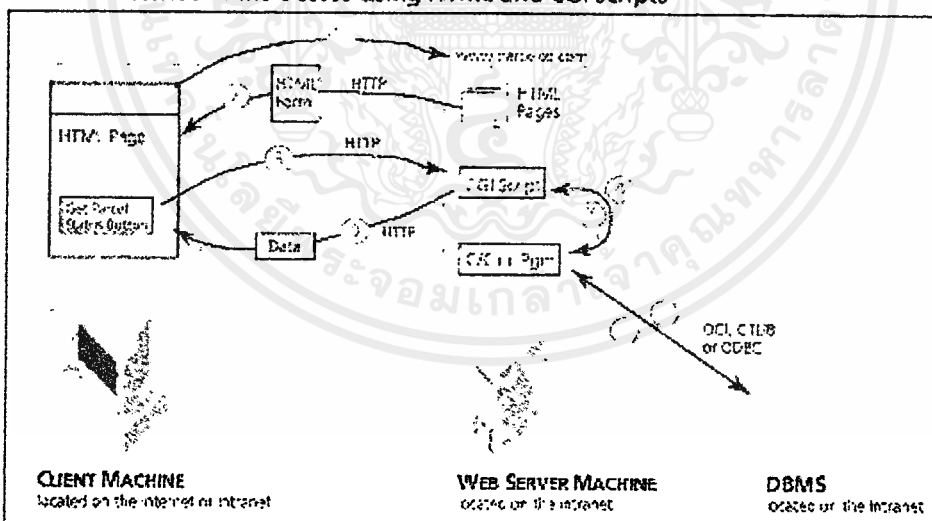
1. HTML pages โดยใช้ Common Gateway Interface (CGI) scripts บน server

ในการใช้ CGI script จะต้องใช้ ODBC driver ในการเข้าถึงข้อมูลของระบบฐานข้อมูล โดย Web server จะส่งรูปแบบของ HTML ไปยังผู้ใช้ Web browser นั้น ๆ อยู่ จากนั้นรูปแบบ HTML จะทำการเข้าถึงข้อมูลที่อยู่ใน DBMS ซึ่งมันจะร้องขอ CGI script บน Web server โดย CGI script ก็จะไปร้องขอไปยัง C/C++ program เพื่อให้ส่งค่า result กลับมาอีกที

ข้อเสีย :

- Low Performance เนื่องจากในการร้องขอไปยังฐานข้อมูลทุกครั้งจะต้องทำการ connection ไปยัง server machine ทุกครั้ง เกิด overhead เพิ่มขึ้น
- Complex Coding ในการใช้ CGI script บ่อยครั้งที่ต้องเขียนด้วย PERL ซึ่งเป็นภาษาที่ทำความเข้าใจได้ยาก ทำการสร้าง หรือ maintain ได้ยากเช่นกัน
- Platform-specific Deployments ในการใช้ CGI script จะต้องมีการเขียน code ทำการสร้าง และทดสอบเฉพาะของแต่ละ platform ไปเลย
- Poor scalability ใน HTTP จะจำกัดความสามารถในการใช้ application แบบ multi-threading เพราะการใช้งานของแต่ละคนกับฐานข้อมูลมักแยกคนละ process กัน
- Reduced Interactivity เพราะมันจะทำได้ที่เครื่อง server เท่านั้น
- Unsophisticated Data Handling

Internet DBMS access using HTML and CGI scripts



รูปที่ 2-11 การติดต่อไปยังฐานข้อมูลโดยใช้ CGI

1. Java programs โดยใช้ JDBC

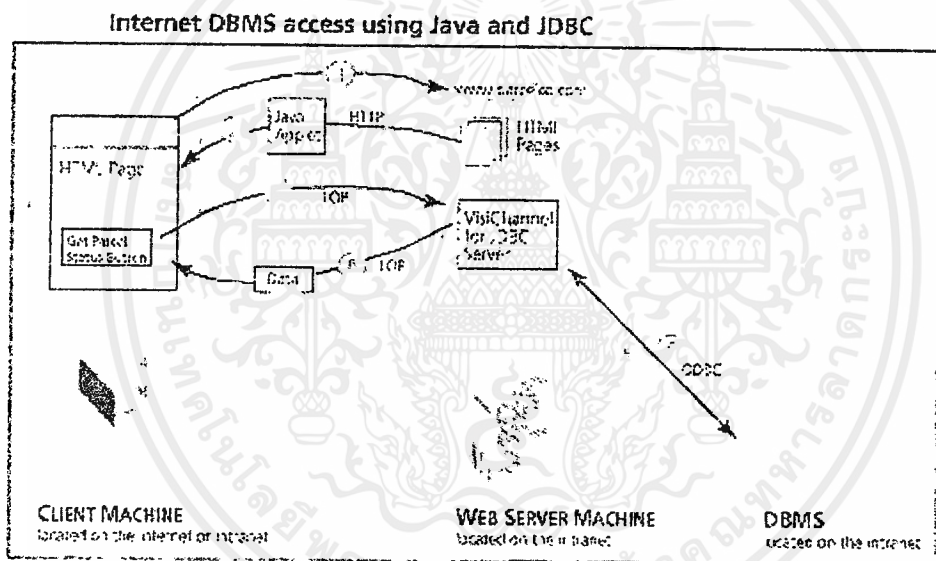
ในการใช้แบบที่ 2 นี้ ใน Web browser จะให้ user ใต้ URL และ Web server จะรับ request จาก user และใช้ HTTP return Web page ไปให้ Web browser ของ client ซึ่งมี Java applet อยู่ด้วย จากนั้นจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับ JDBC request จาก JDBC server และส่ง result ไปยัง JDBC client จากนั้น applet จะได้รับ result และแสดงผลออกมาบน Web browser ของ user

การใช้ตามแบบที่ 2 นี้จะช่วยแก้ปัญหาของข้อแรกได้:

- More Efficient Resource Usage
- Improved Reliability and Flexibility ภาษา Java เป็นภาษาที่มีความน่าเชื่อถือดี (reliable code)
- Complete Interoperability สามารถ run บนเครื่องใดก็ได้
- Increase Scalability มีความสามารถในการทำ multithreading และประโยชน์อื่น ๆ อีกคือ
- อนุญาตให้ใช้ software components และสามารถนำกลับมาใช้ใหม่ได้ เนื่องจากเป็น object-oriented architecture
- มีการจัดหา web-based services ให้



รูปที่ 2-12 การติดต่อไปยังฐานข้อมูลโดยผ่าน JDBC

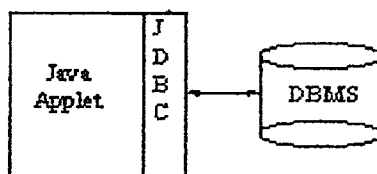
2.2.5 การพิจารณาการออกแบบ Client-Server

รูปแบบของ client-server แบบดั้งเดิมจะถูกสร้างโดยใช้ two หรือ three tiers ดังนี้

Two-Tier Model

รูปแบบนี้ถูกพบเมื่อ client ทำการเรียกแบบตรง ๆ ไปยังฐานข้อมูล ในโลกของ Java จะถูกแสดงโดย Java applets แสดงการเรียก JDBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-13 แสดง Two-Tier model

Java Applet → Database:

Java applet ถูก load เข้าไปใน browser ของ client และเรียกฐานข้อมูลผ่าน JDBC

ข้อดี:

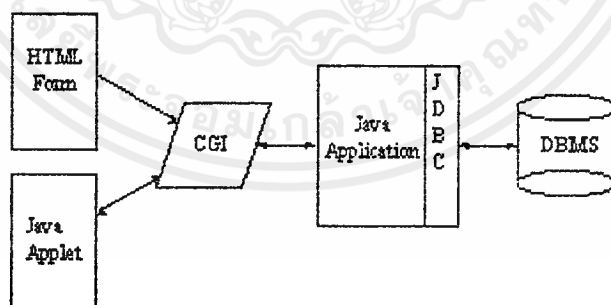
- ช่วยบำรุงการเชื่อมต่อของฐานข้อมูล
- ช่วยลดความต้องการ resource บน Web server โดยทำการประมวลผลบน client แบบ locally
- ทำให้ performance gains มีค่าสูงในเครือข่ายเพราะมีการทำโดยตรงภายในฐานข้อมูล

ข้อเสีย:

- JDBC drivers จะไม่ถูก support จาก commercial browsers เป็น database driver ทั่วไป เช่น ODBC หรือ vendor-provided load module (เช่น DLL) ทำให้ไม่ปลอดภัยเสี่ยงต่อไวรัส
- ทำให้ administrative requirements เพิ่มขึ้น

Three-Tier Model

เป็นการแนะนำ middle layer ที่แสดงการทำงานภายในฐานข้อมูล พบใน Common Gateway Interface ในรูปแบบนี้เราจะสร้าง client ที่ใช้รูปแบบ HTML หรือ Java applet



รูปที่ 2-14 แสดง Three-Tier mode

HTML Form → Java Application → Database:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ HTML นี้จะถูก load ไปใน browser ของ client และปล่อย Java application ไปบน Web server ผ่าน Common Gateway Interface (CGI) Java application จะติดต่อไปยังฐานข้อมูลทำการขอ ร้องและส่งผลกลับมายัง browser เพื่อทำการ display

Applet → Java Application → Database:

Java applet นี้จะถูก load ไปใน browser ของ client และปล่อย Java application ไปบน Web server ผ่าน Common Gateway Interface (CGI) Java application จะติดต่อไปยังฐานข้อมูล ทำการขอ ร้อง และส่งผลกลับมายัง applet เพื่อทำการประมวลผล

ข้อดี:

- มีความปลอดภัยต่อ client และถูก support โดย commercial browsers
- clients ไม่ต้องการการบำรุงเพราะการเข้าถึงฐานข้อมูลทั้งหมดถูกทำบน Web server
- จัดหา central control และ administration ของ logic ของฐานข้อมูลบน Web server

ข้อเสีย:

- มีบาง performance หายไป
- ไม่สามารถบำรุงการติดต่อฐานข้อมูล
- ความต้องการ resource บน Web server เพิ่มขึ้น

2.2.6 ชนิดของ JDBC ไดรฟ์เวอร์ (Driver)

JDBC ไดรฟ์เวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่างๆของ JDBC ถูกจัดแบ่ง ออกเป็นทั้งหมด 4 ชนิดคือ

1. JDBC-ODBC Bridge
2. Native-API (PARTLY-JAVA) Drivers
3. Net-Protocol (All-Java) Drivers
4. Native-Protocol (ALL-Java) Drivers

โดยมีรายละเอียดการทำงานของไดรฟ์เวอร์แต่ละชนิดอธิบายดังนี้

1. JDBC-ODBC Bridge

ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลโดยแปลงฟังก์ชันของ JDBC ให้อยู่ในรูปแบบฟังก์ชันของ ODBC ทำให้ JDBC สามารถติดต่อกับฐานข้อมูลได้โดยผ่านการทำงานของ ODBC รูปแบบการเชื่อมต่อกับฐานข้อมูลของ JDBC-ODBC Bridge แสดงดังรูป

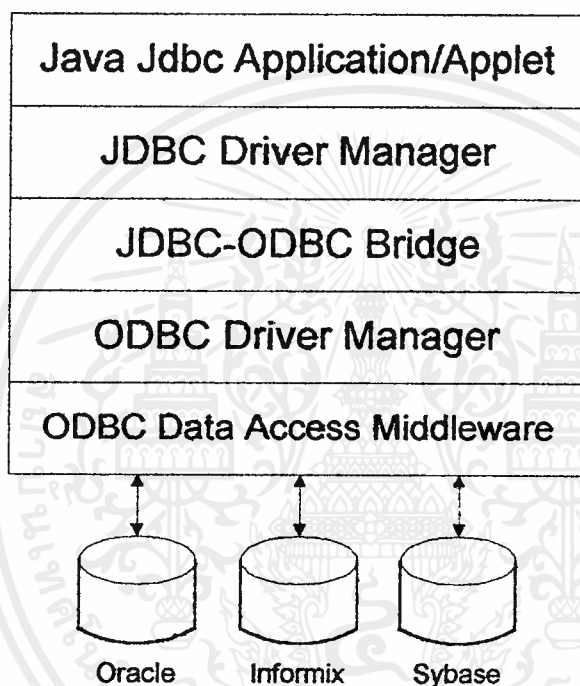
ข้อดี:

-ไม่ต้องติดตั้งและปรับเปลี่ยนไดรฟ์เวอร์ใหม่ทั้งหมด เพราะ ODBC ได้ถูกติดตั้งอยู่บนเครื่องคอมพิวเตอร์ส่วนมากที่ใช้งานในปัจจุบันอยู่แล้ว (Windows 95/NT, Macintosh, และ UNIX บางระบบ) ทำเพียงแค่นำ JDBC-ODBC Bridge มาต่อเชื่อมเพิ่มประสิทธิภาพให้กับ ODBC เท่านั้น

ข้อเสีย:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-JDBC-ODBC Bridge ไคร์ฟเวอร์ถูกเขียนขึ้นจากภาษา C/C++ ในบางส่วน จึงทำให้ไม่สามารถดาวน์โหลด (Download) ขึ้นเครื่องข่ายอินเทอร์เน็ตและแปล (Interpret) บนเว็บเบราว์เซอร์ (Webbrowser) ได้ เพราะภาษา C/C++ ต้องรันเฉพาะแต่ละระบบเท่านั้น ดังนั้น ODBC ไคร์ฟเวอร์ในรูปแบบของไลบรารีไฟล์หรือ DLL (Dynamic Link Library) และ JDBC-ODBC Bridge ไคร์ฟเวอร์ต้องได้รับการติดตั้งบนโคลนเอ็นทีแต่ละเครื่องเพื่อให้สามารถติดต่อกับฐานข้อมูลที่รู้จักได้สุดท้าย JDBC-ODBC Bridge จึงเหมาะสำหรับการใช้งานกับระบบ Three-Tier โดยไคร์ฟเวอร์ทั้งสองจะได้รับการติดตั้งอยู่ที่ Middle-Tier หรือคอมพิวเตอร์ตัวกลาง เช่น Web-Server เป็นต้น



รูปที่ 2-15 แสดง JDBC-ODBC Bridge

2. Native-API (Partly-Java) Drivers

ทำงาน โดยแปลงฟังก์ชันของ JDBC ให้เป็นฟังก์ชัน API เฉพาะของแต่ละฐานข้อมูลเป้าหมาย Native-API Drivers มีหลักการทำงานคล้ายกับ ODBC ที่ต้องมีไคร์ฟเวอร์ (DLL) ติดตั้งอยู่บนโคลนเอ็นทีแต่ละเครื่อง ยกตัวอย่างเช่น การเชื่อมต่อกับฐานข้อมูล Oracle ดังรูป ก็ต้องใช้ JDBC ไคร์ฟเวอร์และ DLL เฉพาะสำหรับติดต่อกับ โปรโตคอล (Protocol) SQL-NET ของ Oracle เท่านั้น ฉะนั้น Native-API Drivers จึงเหมาะกับการทำงานแบบ Two-Tier คือ โคลนเอ็นทีติดต่อกับฐานข้อมูลบนเซิร์ฟเวอร์โดยตรง

ข้อดี:

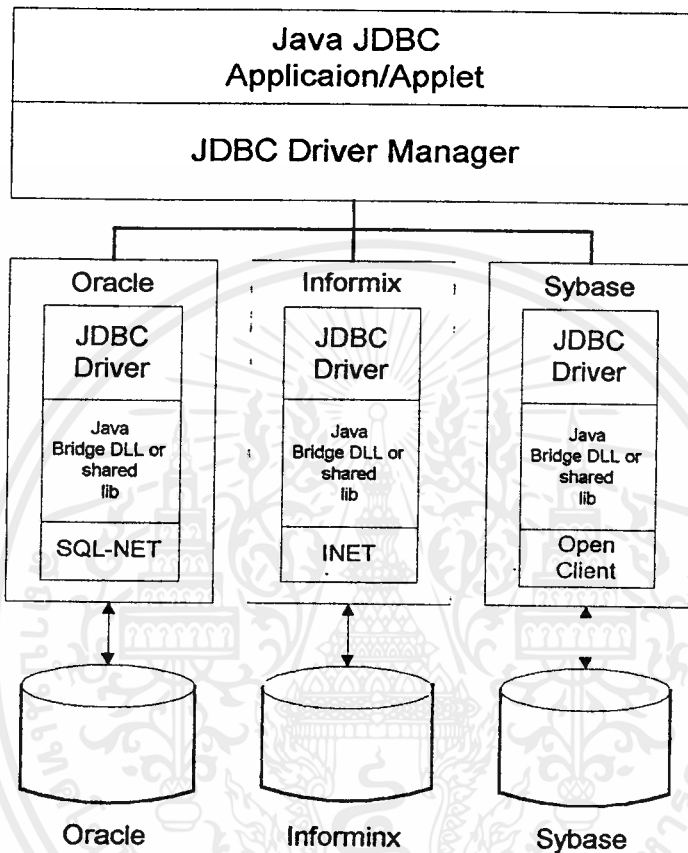
-สามารถใช้ Native-API Drivers ติดต่อกับฐานข้อมูลนั้นๆ โดยตรงโดยผ่าน

โปรโตคอลเดิมที่ใช้อยู่ก่อนแล้ว ทำให้เหมาะสมกับการเชื่อมต่อกับฐานข้อมูลแบบ Two-Tier Client-Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสีย:

- คล้ายคลึงกับ JDBC-ODBC Bridge คือต้องติดตั้งไครฟ์เวอร์ (DLL) และ โปรโตคอลไครฟ์เวอร์ (เช่น SQL-NET, INET, Open Client, ฯลฯ) บนเครื่องคอมพิวเตอร์นั้นๆ ที่สำคัญส่วนหนึ่งของ Native-API Drivers ถูกเขียนขึ้นจากภาษา C++ (Partly-Java) จึงไม่สามารถดาวน์โหลดผ่านเครือข่ายอินเทอร์เน็ตได้ และไม่สามารถเชื่อมต่อกับฐานข้อมูลข้ามชนิดกันได้



รูปที่ 2-16 แสดง Native-API (Partly-Java) Drivers

3. Net-Protocol (All-Java) Drivers

ทำหน้าที่แปลงฟังก์ชันของ JDBC ให้อยู่ในรูปแบบของเน็ตเวิร์ก โปรโตคอลร่วม (DBMS-Independent network Protocol) ซึ่งหลังจากนั้นจะถูกแปลงให้อยู่ในรูปแบบเฉพาะของแต่ละฐานข้อมูลบนเซิร์ฟเวอร์นั้นๆ รูปแบบการเชื่อมต่อข้อมูลแสดงดังรูป สำหรับ JDBC ไครฟ์เวอร์ชนิดนี้เป็นไครฟ์เวอร์ที่ใช้งานได้กว้างขวาง เพราะเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งทำให้สามารถเชื่อมต่อกับฐานข้อมูลหลายๆ ตัวที่แตกต่างกัน

ข้อดี:

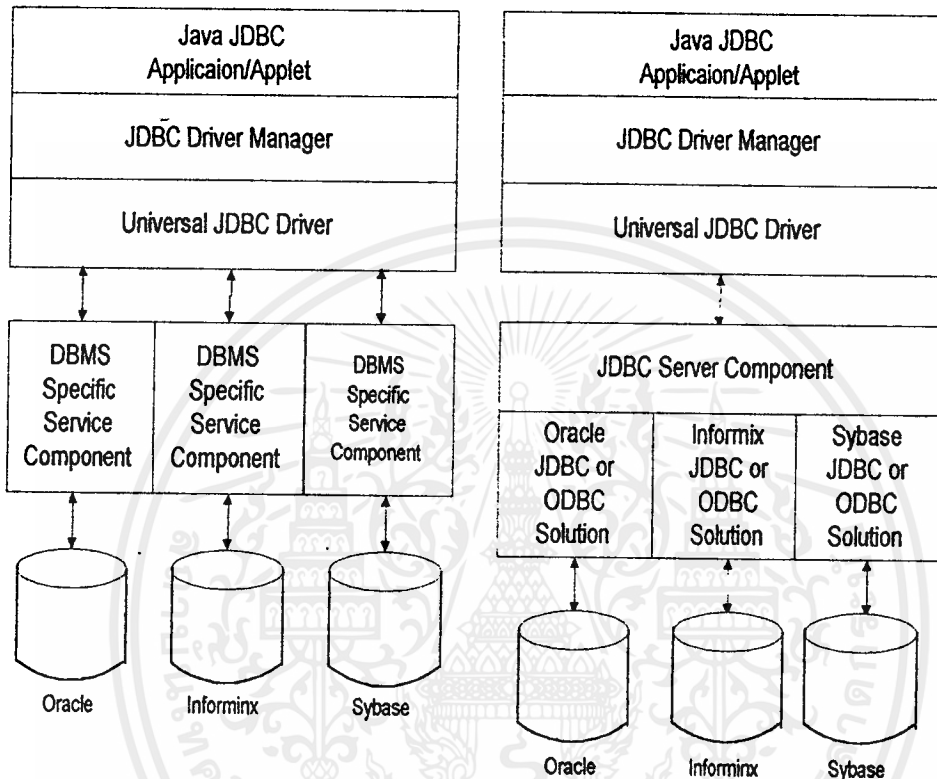
- เป็น ไครฟ์เวอร์ที่ยืดหยุ่นคล่องตัวที่สุด เพราะว่าเขียนขึ้นจากภาษาจาวาทั้งหมด ทำให้ไม่จำเป็นต้องมีไครฟ์เวอร์ร่วมที่เขียนที่เขียนขึ้นจากภาษาอื่นซึ่งต้องติดตั้งบนไคลน์เอ็นท์นั้น และสามารถรันบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบใดก็ได้ที่สนับสนุนสภาพแวดล้อมเสมือนของจาวาหรือ JVM (Java Virtual Machine) การปฏิบัติงานทุกรูปแบบ ทั้งความถี่โหลดข้ามเครือข่ายหรือรันบนเครื่องเดี่ยวๆสามารถกระทำได้หมด

ข้อเสีย:

-ในเรื่องความปลอดภัยเพราะทางเลือกในการใช้งานต่างๆจะขึ้นอยู่กับผู้ผลิตไคร์ฟเวอร์ว่าจะอนุญาตให้ใช้งานส่วนใดได้บ้าง



รูปที่ 2-17 แสดง Net-Protocol (All-Java) Drivers

4. Native-Protocol (All-Java) Drivers

ทำหน้าที่แปลงคำสั่งของ JDBC ให้อยู่ในรูปแบบของเน็ตเวิร์คโปรโตคอลเฉพาะของฐานข้อมูลนั้นโดยตรง การร้องขอข้อมูลจากโคลนเอ็นท์จึงกระทำโดยตรงต่อเซิร์ฟเวอร์ดังแสดงในรูป ไคร์ฟเวอร์ชนิดนี้ถูกเขียนขึ้นจากภาษาจาวาใหม่ทั้งหมดจากผู้ผลิตระบบฐานข้อมูลนั้นๆ และไคร์ฟเวอร์ใหม่นี้จะทำงานที่แทนเน็ตเวิร์คโปรโตคอลเดิม (Oracle SQL-NET, Informix I-Connect, Sybase Open Client, ฯลฯ)

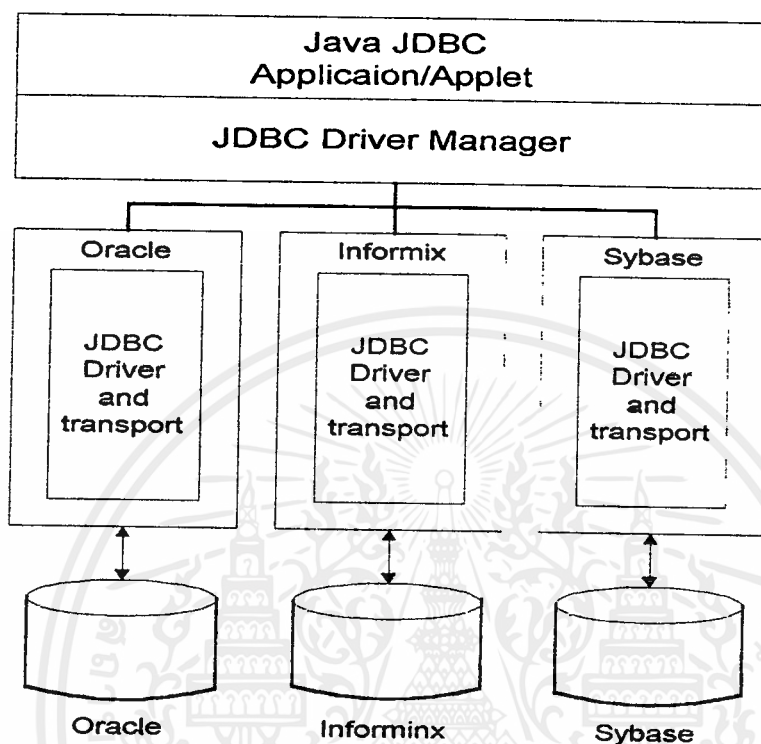
ข้อดี:

-ไคร์ฟเวอร์ชนิดนี้คือ ไม่ต้องมีการปรับเปลี่ยนระบบฐานข้อมูลเดิมที่ใช้งานอยู่แล้วในแต่ละองค์กร และไม่มีควมจำเป็นต้องติดตั้งไคร์ฟเวอร์ตัวกลาง (เช่น ODBC) บน โคลนเอ็นท์แต่ละเครื่อง จึงเหมาะสำหรับการพัฒนาการเชื่อมต่อฐานข้อมูลแบบ Two-Tier สำหรับเครือข่ายอิทรานส์ภายในองค์กร

ข้อเสีย:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไลร์ฟเวอร์ชนิดนี้ไม่เหมาะที่จะนำไปใช้บนเครือข่ายอินเทอร์เน็ต เพราะปัญหาเรื่องความปลอดภัย การเชื่อมต่อข้อมูล ไม่สามารถกระทำข้ามระบบที่แตกต่างกันได้ ต้องใช้ไลร์ฟเวอร์เฉพาะสำหรับฐานข้อมูลตัวนั้นๆ



รูปที่ 2-18 แสดง Native-Protocol (All-Java) Drivers

โดยทั่วไปแล้วแบบที่ 1 จะช้าที่สุด ส่วนแบบที่ 2 จะเร็วที่สุด แบบที่ 3 จะมีการโหลดแต่จะใช้เวลาน้อยส่วนการ execute จะไม่เร็วเท่ากับแบบที่ 2 แบบที่ 4 จะมีประสิทธิภาพที่สุด

JDBC driver ชนิดที่ 3 และ 4 คือ driver ที่คาดว่าจะจะเป็นสิ่งที่ต้องการสำหรับการเชื่อมต่อกับฐานข้อมูลในอนาคตเพราะถูกเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งมีความปลอดภัยและความคล่องตัวมากกว่า (เช่น มีการติดตั้ง driver โดยอัตโนมัติเมื่อมีการ download Java Applet) สำหรับ driver ชนิดที่ 1 และ 2 คือ driver ขัดศาทที่ซึ่งถูกพัฒนาขึ้นมาเพื่อใช้งานร่วมกับ driver ดั้งเดิม (ODBC หรือ Network protocol เฉพาะฐานข้อมูล) ที่มีใช้อยู่ในปัจจุบันไปพลาง ๆ ก่อน ซึ่งแน่นอนว่า driver ทั้ง 2 ชนิดนี้มีข้อจำกัดในการใช้งานมากกว่า 2 ชนิดแรก โดยเฉพาะเรื่องความปลอดภัยและความไม่เป็นอิสระจากระบบคอมพิวเตอร์

2.3 ความรู้และทฤษฎีพื้นฐานโมเดลเชิงสัมพันธ์ (Relation Model)

โมเดลเชิงสัมพันธ์เป็นโมเดลที่ใช้ในการอธิบายความสัมพันธ์ของข้อมูลที่ถูกเก็บด้วยระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ซึ่งเป็นผลงานของ ดร. คอดด์ (Codd) เสนอผลงานวิจัยให้ชาวโลกรู้จักในปี พ.ศ. 2513 โดยมีแนวความคิดในการพัฒนามาจากทฤษฎีทางคณิตศาสตร์ นั่นคือคิดว่าการแก้ปัญหาถ้าใช้ทฤษฎีทางคณิตศาสตร์ที่ถูกต้องแล้ว ไม่ว่าจะหาคำตอบมาจากวิธีการใดย่อมจะต้องได้ผลลัพธ์ที่เหมือนกัน ในปัจจุบันนี้มีการนำโมเดลนี้มาใช้กับเครื่องระดับเมนเฟรมลงไปถึงเครื่องระดับไมโครคอมพิวเตอร์ด้วย และก็เป็นที่ยอมรับกันแล้วว่า บรรดาผู้ใช้ระบบฐานข้อมูลจะมีความคุ้นเคยกับโมเดลเชิงสัมพันธ์นี้มากกว่าอีก 2 โมเดล คือ โมเดลเชิงแตกสาขา (Hierarchy Model) และ โมเดลเชิงโครงข่าย(Network Model) ที่มีมาก่อนหน้านี้

การเปรียบเทียบระหว่างโมเดลเชิงสัมพันธ์กับโมเดลอื่นๆ

นอกเหนือจากความแพร่หลายของโมเดลเชิงสัมพันธ์นี้แล้ว ข้อดีของโมเดลเชิงสัมพันธ์ที่มีมากกว่าอีก 2 โมเดล มีดังนี้

1. โมเดลเชิงสัมพันธ์เป็นโมเดลที่สร้างความเข้าใจได้ง่ายกว่า เพราะภาพลักษณ์ของข้อมูลที่เก็บโดยโมเดลเชิงสัมพันธ์จะมาจากมุมมองของผู้ใช้ ซึ่งจะมีความซับซ้อนน้อยกว่าภาพลักษณ์ของข้อมูลที่เก็บโดยอีก 2 โมเดล
2. ระบบส่วนใหญ่ที่ใช้โมเดลเชิงสัมพันธ์นี้มักจะมีเครื่องมือที่ช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลที่จัดเก็บด้วยโมเดลแบบอื่น
3. โมเดลเชิงสัมพันธ์นี้มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถค้นพบปัญหาที่เกิดขึ้นในการออกแบบระบบฐานข้อมูลได้โดยง่ายและยังง่ายในการแก้ไขการออกแบบที่ผิดพลาดนั้นด้วย
4. โมเดลเชิงสัมพันธ์เป็นโมเดลที่มีความสอดคล้องกับหลักการฐานข้อมูล ผู้ใช้ไม่ต้องพะวงกับรายละเอียดของการจัดเก็บข้อมูลเหมือนกับการจัดข้อมูลของโมเดลอื่น
5. ภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บระบบการจัดการฐานข้อมูลเชิงสัมพันธ์(ภาษา SQL ; Structure Query Language)เป็นภาษาแบบเซตโอเรียนต์(set oriented) ซึ่งจะต่างกับภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลของโมเดลอื่นที่เป็นภาษาแบบ record-at-a-time

แม้ว่าโมเดลเชิงสัมพันธ์จะมีข้อดีหลายประการดังได้กล่าวไปแล้ว แต่ในปัจจุบันก็ยังมีจุดอ่อนที่มีการอ้างอิงถึงเสมอ คือ ระบบจัดการฐานข้อมูลแบบโมเดลเชิงสัมพันธ์นี้มักจะมีประสิทธิภาพในการใช้งานสู้อีก 2 โมเดลไม่ได้ โดยเฉพาะในการประยุกต์ใช้งานขององค์กรขนาดใหญ่ จุดอ่อนนี้ก็ได้มีการแย้งกลับมาในแง่ที่ว่า โมเดลเชิงสัมพันธ์ก็ยังมีจำนวนระดับขั้นที่ได้พัฒนาไปแล้วยังน้อยกว่าอีก 2 โมเดล ดังนั้นหากต้องการเปรียบเทียบที่ระดับจำนวนขั้นการพัฒนาที่เท่ากันจึงจะสมเหตุสมผล

ศัพท์เฉพาะของโมเดลเชิงสัมพันธ์

ในหัวข้อนี้จะกล่าวถึงโมเดลเชิงสัมพันธ์โดยกำหนดนิยาม และกล่าวถึงคำศัพท์ต่างๆที่เกี่ยวข้องกับโมเดลนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการเก็บข้อมูลที่เก็บด้วยโมเดลเชิงสัมพันธ์จะถูกเก็บไว้ในตารางที่ถูกเรียกว่า “รีเลชัน” โดยที่รีเลชันทุกรีเลชันอาจแสดงในรูปของตาราง แต่ตารางบางตารางอาจไม่เป็นรีเลชันก็ได้ ตารางที่มีลักษณะเป็นรีเลชันต้องมีคุณสมบัติดังนี้

1. แต่ละช่องของตารางจะบรรจุข้อมูลได้เพียงค่าเดียว
2. ชื่อหัวชื่อในแต่ละคอลัมน์มีความแตกต่างกัน อันเป็นชื่อของแอตทริบิวของเอนทิตี
3. ค่าของข้อมูลในแต่ละคอลัมน์ คือ ค่าของแอตทริบิวของเอนทิตี
4. การเรียงลำดับคอลัมน์ไม่ถือว่ามีความสำคัญ
5. ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
6. การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

ตารางที่มีคุณสมบัติดังกล่าวจะเรียกว่ารีเลชัน ดังนั้นเราจะได้นิยามของฐานข้อมูลเชิงสัมพันธ์ คือ ฐานข้อมูลที่เกิดจากการรวมรีเลชันต่างๆที่มีความสัมพันธ์(Relationship)ระหว่างกัน

เราจะเรียกข้อมูลแต่ละแถวในแวนอนของรีเลชันว่า ทัพเพิล(Tuple) และเรียกข้อมูลในแต่ละแถวในแนวตั้ง หรือแนวคอลัมน์ว่า แอททริบิว (Attribute) โดยที่คำว่า คีย์ (key) จะหมายถึงข้อมูลที่เกิดจากแอตทริบิว 1 ตัว หรือ หลายตัวก็ได้

แต่ละรีเลชันจะต้องมีสิ่งที่เรียกว่า คีย์หลัก(Primary Key) คือ คีย์คู่แข่ง(Candidate Key) ตัวที่เลือกขึ้นมา ซึ่งมักจะเลือกตัวที่สั้นกว่า กระทั่งที่สุด เพื่อประโยชน์ในการอ้างอิงและโอกาสป้อนข้อมูลผิดพลาดน้อยลง

คีย์คู่แข่ง(Candidate Key) หรือคีย์สำรอง(Alternate Key) คือกลุ่มของแอตทริบิวที่มีคุณสมบัติดังนี้

1. Uniqueness คือมีค่า(value)ไม่ซ้ำ 2 แถว(tuple)ใดๆ
2. Minimally คือกลุ่มของแอตทริบิวที่มีจำนวนน้อยที่สุด ที่จะทำให้ได้คุณสมบัติ

Uniqueness

หรือถ้าจะกล่าวให้ง่ายก็คือ แอททริบิว 1 ตัวหรือมากกว่า 1 ตัวก็ได้ ที่สามรถใช้เป็นตัวบอกได้ว่าเมื่อเราอ้างอิงแอตทริบิวชุดนี้แล้วเราจะได้ทัพเพิลที่ไม่ซ้ำกัน และแอตทริบิวอื่นๆที่เหลือที่มีได้เป็นคีย์หลักและไม่ได้เป็นส่วนหนึ่งของคีย์หลัก ก็จะถูกเรียกว่าเป็น Nonkey Attribute

ค่าว่าง(Null Value) คือสิ่งที่มีลักษณะดังนี้

1. เป็นแอตทริบิวที่เกี่ยวข้องแต่ไม่ทราบค่า
2. เป็นแอตทริบิวที่ไม่เกี่ยวข้องกับแอตทริบิวตัวอื่นในทัพเพิลนั้นเลย

ส่วนคีย์ที่เป็นแอตทริบิวของรีเลชันอื่นที่ซ้ำกับแอตทริบิวที่เป็นคีย์หลักของรีเลชันหนึ่งจะเรียกว่า คีย์นอก(Foreign Key)

คำว่า โดเมน(Domain) จะหมายถึงกรอบของค่าต่างๆที่เป็นไปได้ เช่น โดเมนของแอตทริบิว วันที่ ก็จะหมายถึงค่าของวันที่ที่เป็นไปได้ คือ มีค่าเท่ากับ 1 ถึง 31

แต่ในการเก็บค่าข้อมูลลงในรีเลชันนั้น บางกรณีที่เรามีการกำหนดโดเมนให้แอตทริ-

บิวแล้ว แต่ข้อมูลที่จะถูกเก็บเข้าไปอาจถูกบรรจุเข้าไปในภายหลัง ลักษณะนี้จะทำให้เกิด ค่าว่าง(null value) ขึ้นชั่วขณะก่อนที่จะมีการบรรจุค่าข้อมูลที่อยู่ใน โดเมนที่กำหนดไว้เข้าไป

เมื่อมีการจัดเก็บ ในมูลฐานข้อมูลใดๆแล้วข้อมูลจะถูกแยกออกเป็นกลุ่มของข้อมูลเป็นชุดที่ประกอบด้วยแอททริบิวต์ต่างๆที่มีความสัมพันธ์กัน เช่นการเก็บข้อมูลของบุคคลากรในโรงเรียนก็อาจแยกเก็บเป็นกลุ่มของข้อมูลนักเรียน, กลุ่มข้อมูลของครูอาจารย์ และกลุ่มของนักการภารโรง เป็นต้น กลุ่มข้อมูลแต่ละกลุ่มนี้จะเรียกว่า เอนติตี้(Entity) ซึ่งแต่ละเอนติตี้จะประกอบไปด้วยแอททริบิวต์ต่างๆที่มีความสัมพันธ์ เช่น เอนติตี้ของนักเรียนก็จะประกอบไปด้วย ชื่อ, ที่อยู่, ชั้นเรียน เป็นต้น

จากการแยกจัดเก็บข้อมูลออกเป็นเอนติตี้ แต่ละเอนติตี้ก็มีความสัมพันธ์กัน และสามารถเปรียบเทียบคำศัพท์ที่ใช้ในเชิงวิชาการกับคำศัพท์ที่ใช้ทั่วไปได้ดังนี้

ศัพท์เชิงวิชาการ	ศัพท์เชิงธุรกิจ
relation	table ที่มี row ไม่ซ้ำกัน
tuple	row หรือ record
cardinality	จำนวนของ row ในแต่ละ table
attribute	column หรือ field
degree	จำนวน column ในแต่ละ table
primary key	column ที่อาจจะเป็น 1 column หรือมากกว่า 1column ที่เมื่อระบุค่าข้อมูลใน column ที่เป็น primary key 1 ค่า จะได้ข้อมูลใน table นั้น ออกมาเพียง 1 row
domain	ค่าของข้อมูลที่เป็นไปได้ทั้งหมดในแต่ละ column

ความสัมพันธ์ระหว่างเอนติตี้แบ่งออกเป็น 3 ชนิด คือ

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง(One to One)
- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม(One to Many)
- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม(Many to Many)

ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างเอนติตี้ หมายถึง เมื่อเอนติตี้หนึ่งมีข้อมูลของ คีย์หลักค่าหนึ่ง ค่าของข้อมูลดังกล่าวก็มีความสัมพันธ์กับค่าข้อมูลคีย์หลักของอีกเอนติตี้หนึ่ง เพียงค่า เดียวเท่านั้น เช่นหากเรากำหนดให้ความสัมพันธ์ระหว่างเอนติตี้กับนักเรียนกับเอนติตี้ผู้ปกครองสามารถ อ้างอิงถึงผู้ปกครองได้เพียงคนเดียวเท่านั้น และในทางตรงข้ามก็ต้องเป็นจริงด้วย คือเมื่อเราอ้างอิงถึงผู้ ปกครองคนใดคนหนึ่งแล้วสามารถอ้างอิงถึงนักเรียนคนเดียวเท่านั้น

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	b
C	c

ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนตีสี่ หมายถึง เมื่อเอนตีหนึ่งมีข้อมูลของคีย์หลักค่าหนึ่ง ค่าข้อมูลดังกล่าวก็มีความสัมพันธ์กับข้อมูลของคีย์หลักของอีกเอนตีหนึ่งได้หลายค่า เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่างเอนตีกับผู้ปกครอง เป็นแบบหนึ่งต่อกลุ่มแล้ว จะหมายความว่า การที่เราอ้างอิงถึงนักเรียนคนใดคนหนึ่ง ก็จะสามารถอ้างอิงถึงผู้ปกครองคนใดคนหนึ่งเพียงคนเดียวเท่านั้น และในทางตรงข้ามก็มีความหมายว่า เมื่อเราอ้างอิงผู้ปกครองคนใดคนหนึ่งแล้ว สามารถอ้างอิงถึงนักเรียนได้เพียงคนเดียวเท่านั้น แต่ผู้ปกครองที่เราอ้างอิงเป็นคนละคนกันก็จะอ้างอิงถึงนักเรียนคนเดียวกันได้

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	b
C	c

ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนตีสี่ หมายความว่า ค่าข้อมูลของคีย์หลักของเอนตีหนึ่งที่ต่างกันอาจอ้างอิงถึงค่าข้อมูลหลักของอีกเอนตีหนึ่ง ได้ค่าเดียวกันหรือหลายค่าก็ได้ เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่างเอนตีนักเรียนกับเอนตีผู้ปกครองเป็นแบบกลุ่มต่อกลุ่มแล้วก็จะหมายความว่า การที่เราอ้างอิงถึงนักเรียนคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงผู้ปกครองคนเดียวกันได้ และในทางกลับกัน การที่เราอ้างอิงถึงผู้ปกครองคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงนักเรียนคนเดียวกันได้

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	b
C	c
D	d

นอกจากนี้ เอนทิตียังมีเอนทิตีบางประเภทที่การอ้างถึงตัวมันได้อย่างสมบูรณ์นั้นจะต้องอ้างถึงเอนทิตีอื่นเสมอ เช่น หากระบบฐานข้อมูลของบุคคลากรในโรงเรียนมีเอนทิตีของคุณักการภารโรงด้วยแล้ว การอ้างถึงเอนทิตีนี้ได้จะต้องอ้างถึงเอนทิตีนักการภารโรงด้วยแล้ว การอ้างถึงเอนทิตีนี้ได้จะต้องอ้างถึงเอนทิตีนักการภารโรงด้วยเสมอ เอนทิตีประเภทนี้เราเรียกว่า เอนทิตีชนิดอ่อน(weak entity)

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relation Database Management System)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์เป็นซอฟต์แวร์ สำหรับจัดการข้อมูลและความสัมพันธ์ของข้อมูลที่เป็นแบบเชิงสัมพันธ์ รวมทั้งมีภาษาสำหรับกำหนดโครงสร้างของข้อมูลในระบบต่างๆตามสถาปัตยกรรมมาตรฐานของระบบฐานข้อมูล มีภาษาสำหรับการค้นหาและแก้ไขข้อมูลรวมทั้งการจัดการความปลอดภัยของข้อมูล การเข้าถึงข้อมูลและการติดต่อกับผู้ใช้และโปรแกรมของผู้ใช้ที่เขียนด้วยโปรแกรมอื่น ดังมีรายละเอียดของประโยชน์ของการใช้ระบบการจัดการข้อมูลเชิงสัมพันธ์ดังนี้

- ภาษาเอสคิวแอล(SQL : Structure Query Language)

ช่วยจัดการการทำงานเกี่ยวกับข้อมูล สามารถแบ่งตามลักษณะการทำงานได้ 2 ส่วนคือ

1. การจัดการเกี่ยวกับการกำหนดโครงสร้างข้อมูล (Data Definition language : DDL)

มีหน้าที่ในการกำหนดโครงสร้างข้อมูลที่ใช้แต่ละคนมองเห็น โครงสร้างข้อมูลนี้นอกจากเป็นแบบฐานข้อมูลมองเห็นและโครงสร้างข้อมูลที่จัดเก็บในอุปกรณ์เก็บข้อมูลซึ่งผลของการแปล DDL จะเก็บในไฟล์พิเศษที่เรียกว่าพจนานุกรมของข้อมูล (Data Dictionary)

2. การจัดการเกี่ยวกับข้อมูล (Data Manipulation Language : DML)มีหน้าที่ในด้านจัดการการเข้าถึงข้อมูล ได้แก่ การสอบถามหรือการเรียกค้นข้อมูลที่อยู่ในฐานข้อมูลการเพิ่มแก้ไขข้อมูลใหม่เข้าไปในฐานข้อมูล การลบข้อมูลออกจากฐานข้อมูล การเปลี่ยนแปลงแก้ไขข้อมูลที่อยู่ในฐานข้อมูล

- ประสิทธิภาพสูงและสะดวกรวดเร็วในการพัฒนาระบบงาน(Productivity)

เนื่องจากเมื่อต้องการข้อมูลจะอ่านข้อมูลจะอ่านขึ้นมาเป็นเซตของข้อมูล ทำให้สามารถทำงานได้สะดวกรวดเร็วและสามารถพัฒนาระบบงานได้รวดเร็วขึ้น นอกจากนี้โครงสร้างการทำงานเป็นลักษณะที่รองรับให้ผู้ใช้ได้หลายคน ดังนั้นจึงไม่ต้องเขียนโปรแกรมที่ควบคุมการให้ผู้ใช้สามารถทำงานได้พร้อมๆกันหลายคน ควบคุมด้านความปลอดภัยของข้อมูล ควบคุมด้านอำนาจสิทธิของผู้ใช้แต่ละคน นอกจากนี้ยังสามารถกู้ข้อมูลได้โดยอัตโนมัติเมื่อระบบงานเกิดความเสียหาย

- การจัดการด้านความเป็นอิสระของข้อมูล (Data Independence)

เมื่อมีการเปลี่ยนแปลงแก้ไขโครงสร้างข้อมูล ในนิยามข้อมูลระดับแนวความคิดหรือในนิยามข้อมูลระดับภายในนั้นจะไม่มีผลกระทบต่อโปรแกรมที่ให้ผู้ใช้งานในนิยามข้อมูลระดับภายนอก เนื่องจากระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ จะช่วยจัดการให้ผู้ใช้มองเห็นโครงสร้างข้อมูลในนิยามข้อมูลระดับภายนอกเหมือนเดิมและสามารถใช้งานได้ตามปกติ ซึ่งการจัดการด้านความเป็นอิสระของข้อมูลแบ่งออกเป็น 2 ลักษณะคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ความเป็นอิสระของข้อมูลในระดับตรรกะ (Logic Data Independence)

เมื่อมีการแก้ไขโครงสร้างข้อมูลในนิยามข้อมูลระดับแนวคิด จะไม่มีผลกระทบต่อโครงสร้างข้อมูลในนิยามข้อมูลระดับภายนอกที่ผู้ใช้ใช้งานกันอยู่ เช่น การเปลี่ยนแปลงขนาดคอลัมน์ หรือเพิ่มคอลัมน์เข้าไป

2. ความเป็นอิสระของข้อมูลในระดับกายภาพ (Physical Data

Independence)เมื่อมีการแก้ไขโครงสร้างข้อมูลในนิยามข้อมูลระดับภายใน จะไม่มีผลกระทบต่อโครงสร้างข้อมูลในนิยามข้อมูลระดับแนวคิด เช่น มีการเปลี่ยนแปลงวิธีการเข้าถึงข้อมูลในนิยามข้อมูลระดับภายใน โดยการเปลี่ยนแปลงลักษณะการจัดเก็บจาก ซีซ ไอแซม(C-ISAM) เป็นบีทรี(Btree) หรือแม้กระทั่งเปลี่ยนแปลงอุปกรณ์ที่ใช้จัดเก็บข้อมูลอย่างเช่น ฮาร์ดดิสต์(Harddisk) หรือการอัปเกรด(Upgrade)ระบบปฏิบัติการ

● ความปลอดภัยของข้อมูล(Data Security)

ระบบการจัดการฐานข้อมูลได้มี ระบบการสอบถามรายชื่อพร้อมทั้งรหัสลับของผู้เข้ามาใช้งานในระบบงานเพื่อให้เข้าทำงานในส่วนที่ผู้ใช้เกี่ยวข้องเท่านั้น นอกจากนี้ข้อมูลที่เก็บอยู่ในฐานข้อมูลเชิงสัมพันธ์นั้น จะถูกทำการเข้ารหัส (Encryption) เพื่อป้องกันการเข้าถึงข้อมูลด้วยภาษาอื่นๆ โดยไม่ผ่านระบบการจัดการฐานข้อมูลเชิงสัมพันธ์

● การจัดอำนาจในการใช้ข้อมูล(Data Authority)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ สามารถทำการกำหนดสิทธิอำนาจการใช้งานข้อมูลให้แก่ผู้ใช้ต่างๆ ได้โดยการใช้คำสั่ง GRANT หรือเพิกถอนสิทธิการใช้โดยใช้คำสั่ง REVOKE นอกจากนี้ยังใช้ VIEW เพื่อช่วยกำหนดสิทธิการใช้ข้อมูลของผู้ใช้แต่ละคนในนิยามข้อมูลระดับภายนอกได้อีกด้วย

● ระบบการกู้ข้อมูล โดยอัตโนมัติเมื่อระบบเกิดความเสียหาย(Recovery Control)

ระบบการจัดการข้อมูลฐานข้อมูลเชิงสัมพันธ์ ได้เตรียมการรองรับการทำการประมวลผลกลุ่มงาน (Transaction Processing) โดยกลุ่มงาน (Transaction)นี้มีความหมายถึงกลุ่มของงานในทางตรรกะ(Logic Unit of Work) ซึ่งยอมให้มีการละเมิดกฎสำหรับควบคุมความถูกต้องของข้อมูลได้ภายในกลุ่มของงาน และการทำงานทั้งหมดในกลุ่มงานนี้ ถ้าทำงานเสร็จตามความต้องการจะต้องเสร็จด้วยกันทั้งหมด ถ้าเกิดความบกพร่องขึ้นภายในกลุ่มงานจะต้องยกเลิกทั้งหมดโดยที่ Commit คือคำสั่งยืนยันการทำงานสำเร็จ ส่วน Rollback คือคำสั่งยกเลิกการทำงานของกลุ่มงานนั้น ดังนั้นถ้า Commit แล้วระบบเกิดความเสียหายขึ้นระบบการจัดการฐานข้อมูลเชิงสัมพันธ์จะรับรองว่าข้อมูลที่ Commit แล้วจะไม่สูญหายไปทางกลับกันกรณีของ Rollback และการทำงานที่ยังไม่ Commit นั้นระบบการจัดการฐานข้อมูลเชิงสัมพันธ์จะรับรองข้อมูลเดิมก่อนการทำงานยังไม่สูญหายไปไหน

● การดูแลผู้ใช้หลายคนให้สามารถทำงานได้พร้อมๆกัน (Concurrency Control)

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ จะมีการรองรับการทำงานพร้อมกันของผู้ใช้หลายคนได้โดยอาจมีปัญหาก่อเกิดขึ้นได้ 3 ประเภทใหญ่ๆ คือ

1. ปัญหาการสูญเสียการเปลี่ยนแปลงข้อมูล(The Lost Update Problem)

เกิดจากการอ่านข้อมูลตัวเดียวกันไปใช้งาน โดยที่มีการเปลี่ยนแปลงข้อมูลตัวนั้นในเวลาที่ไม่พร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ปัญหาที่เกิดจากการยกเลิกการเปลี่ยนแปลงข้อมูล (The Uncommitted Dependency Problem) เกิดจากมีการนำเอาข้อมูล ไปใช้ในขณะที่การประมวลผลกลุ่มงาน (Transaction Processing) ยังไม่เสร็จสิ้นและมีการยกเลิกการประมวลผลกลุ่มงานนั้น

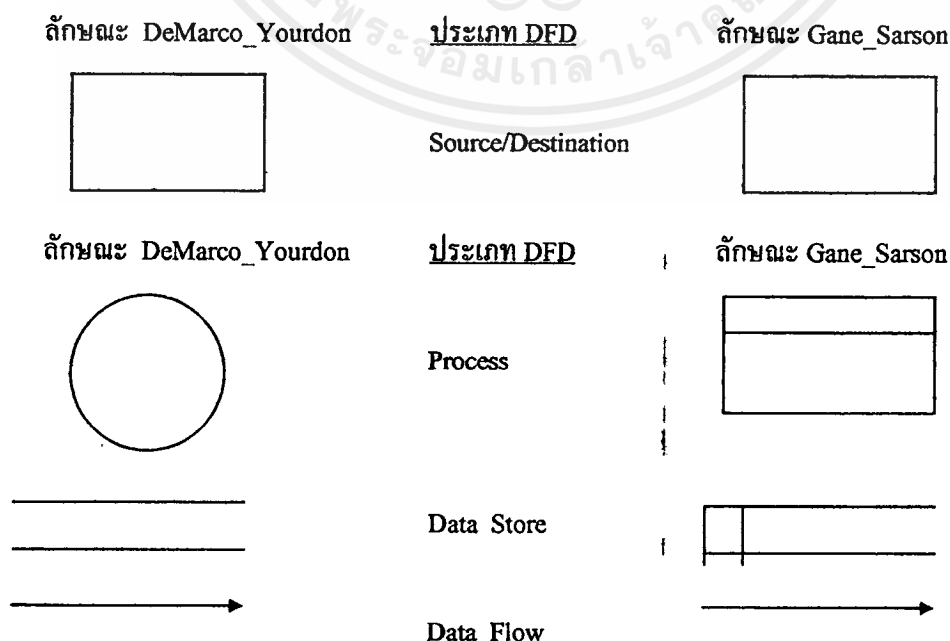
3. ปัญหาที่เกิดจากการนำเอาข้อมูล ไปประมวลผล และมีการยอมรับความเปลี่ยนแปลงนั้น โดยที่การเปลี่ยนแปลงที่เกิดขึ้นมีผลต่ออีกการทำงานหนึ่ง ซึ่งปัญหาต่างๆเหล่านี้สามารถป้องกันได้โดยวิธีการจองข้อมูล(Lock) ก่อนการใช้งาน ช่วงการจองข้อมูลนี้มีอยู่ 2 แนวทางคือ

-จองข้อมูลแบบแชร์(Shared-mode Lock) หมายถึง ถ้ากลุ่มงาน(Transaction) Ti จองแบบแชร์(Shared-mode Lock แทนด้วย S) บนข้อมูล Q ดังนั้น Ti สามารถอ่านข้อมูลจาก Q ได้โดยไม่สามารถเขียนลง Q ได้ซึ่งกลุ่มงาน (Transaction) อื่นที่เข้ามาใช้งาน Q หลังจากนั้นก็จะสามารถอ่านข้อมูลจาก Q ได้อย่างเดียวเท่านั้น

-จองข้อมูลแบบพิเศษ(Exclusive-mode Lock) หมายถึง ถ้ากลุ่มงาน (Transaction) Ti จองแบบพิเศษ(Exclusive-mode Lock แทนด้วย X) บนข้อมูล Q ดังนั้น Ti สามารถอ่านข้อมูลจาก Q ได้โดยไม่สามารถเขียนลง Q ได้ซึ่งกลุ่มงาน (Transaction) อื่นที่เข้ามาใช้งาน Q หลังจากนั้นก็จะต้องรอให้เลิกการจอง(Unlock) ก่อนจึงจะสามารถใช้งานข้อมูล Q ได้

ทฤษฎีการออกแบบระบบงาน

แนวทางการออกแบบระบบงานวิธีหนึ่งที่นิยมกันอย่างแพร่หลาย และสามารถแสดงความสัมพันธ์ในรายละเอียดได้ชัดเจนคือ วิธีการออกแบบค่าไฟลด์โคอะแกรม (DFD : Data Flow Diagram) ซึ่งวิธีนี้จะใช้สัญลักษณ์ที่แสดงความสัมพันธ์เพียง 4 สัญลักษณ์เท่านั้น และจะเริ่มค้นพิจารณาความสัมพันธ์ของระบบอย่างกว้างๆเป็นอันดับแรกก่อน หลังจากนั้นจะพิจารณาในรายละเอียดในแต่ละส่วนงานเพิ่มขึ้นเป็นอันดับต่อไป โดยที่ลักษณะสัญลักษณ์ของการออกแบบวิธีนี้แบ่งเป็น 2 รูปแบบใหญ่ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-19 แสดงประเภทลักษณะสัญลักษณ์ของค่าโพล์โคอะแกรม(DFD)

ในระบบงานนี้จะใช้สัญลักษณ์ของ Gane_Sarson ในการแสดงขั้นตอนการทำงานของระบบงาน และนอกจากนี้ยังมีข้อควรระวังซึ่งก่อให้เกิดปัญหาในการออกแบบระบบงาน 3 ประการดังมีรายละเอียดคือ

1. แบลคโฮล (Black Hole) คือ ขั้นตอนการทำงาน(Process) ที่มีแต่ข้อมูลเข้า(input) แต่ไม่มีผลลัพธ์(output)ของการทำงานออกมา
2. มิวราเคิล(Miracle) คือ ขั้นตอนการทำงาน(Process) ที่ไม่มีข้อมูลเข้า(input) แต่กลับมีผลลัพธ์(output)ของการทำงานออกมา
3. เกรโฮล(Gray Hole) คือขั้นตอนการทำงาน(Process) ที่ข้อมูลเข้า(input)ทั้งหมดไม่เพียงพอที่จะทำให้เกิดผลลัพธ์(output)ของการทำงานออกมา

ทฤษฎีการออกแบบฐานข้อมูลเชิงสัมพันธ์

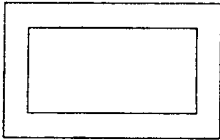
แนวทางการออกแบบฐานข้อมูล สามารถแบ่งแนวทางเป็น 2 แนวทางคือ

1. แบบวิเคราะห์(Analysis หรือ Decomposition) มีแนวทางการออกแบบ คือ พิจารณาแอททริบิวต์และความสัมพันธ์ในระบบงานทั้งหมด จากนั้นจะทำการแบ่งแยกออกเป็นกลุ่มของแอททริบิวต์ที่มีความสัมพันธ์ในกลุ่มที่เล็กลงมา ตัวอย่างของวิธีการออกแบบฐานข้อมูลแนวทางนี้คือ วิธีนอร์มอลไลเซชัน(Normalization)
2. แบบสังเคราะห์(Synthesis หรือ Reconstruction) มีแนวทางการออกแบบคือ พิจารณาแต่ละแอททริบิวต์ว่ามีความสัมพันธ์กับแอททริบิวต์อื่นอย่างไร แล้วเขียนแผนภาพแสดงความสัมพันธ์ เมื่อสามารถกำหนดความสัมพันธ์ให้กับแต่ละแอททริบิวต์ในระบบงานได้ทั้งหมดแล้วก็จะทำการสร้าง(rebuilt) กลุ่มของความสัมพันธ์โดยใช้คีย์หลักร่วมกัน(common Primary Key)ตัวเดียวกัน ตัวอย่างของวิธีการออกแบบฐานข้อมูลแนวทางนี้คือวิธีอีอาร์โคอะแกรม (ER : Entity Relationship Model)และวิธีไนแอม(NIAM : Nijssen's Information Analysis Methodology)

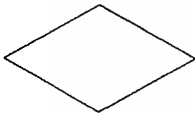
ในระบบงานนี้จะใช้วิธีอีอาร์โมเดล (ER : Entity Relationship Model) เป็นเครื่องมือในการออกแบบฐานข้อมูลเชิงสัมพันธ์ในระดับตรรกะ เนื่องจากแนวความคิดที่ให้ Conceptual Schema มีพื้นฐานมาจากโครงสร้างธรรมชาติ ใช้รูปประโยคที่มีประธาน กริยา กรรม วิธีแสดงรูปแบบความสัมพันธ์ระหว่างข้อมูล เป็นแบบจำลองที่มีความหมาย และมีเครื่องหมายแสดงความสัมพันธ์ของข้อมูลและข้อจำกัดของข้อมูลได้อย่างชัดเจนนอกจากนั้น ยังสามารถแปลงConceptual Schema เป็น Relational Database Schema ซึ่งจะอยู่ในรูป Third Normal Form และทำการนอร์มอลไลเซชัน(Normalization)ต่อเนื่องถึง Fifth Normal Form ซึ่งวิธีการแบบอีอาร์โคอะแกรมนี้จะใช้สัญลักษณ์แสดงความสัมพันธ์ของข้อมูลในระบบงานแสดงข้อกำหนดต่างๆเพื่อช่วยแสดงความสัมพันธ์ให้ชัดเจนยิ่งขึ้น ซึ่งมีสัญลักษณ์ต่างๆได้ดังนี้

สัญลักษณ์**ความหมาย**

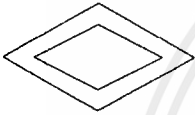
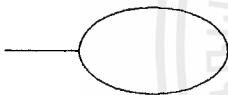
ชนิดของเอนติตี้(Entity Type)



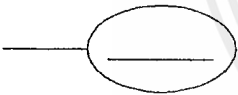
เอนติตี้ชนิดอ่อน(Weak Entity Type)



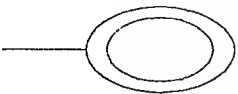
ความสัมพันธ์(Relationship Type)

ความสัมพันธ์ระหว่างเอนติตี้กับเอนติตี้ชนิดอ่อน หรือ
ความสัมพันธ์ระหว่างเอนติตี้ชนิดอ่อนกับเอนติตี้ชนิดอ่อน(Identifying
Relationship Type)

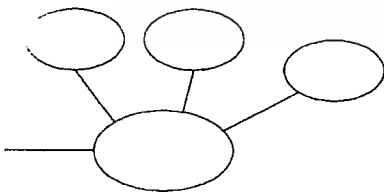
แอททริบิวต์(Attribute)



คีย์แอททริบิวต์(Key Attribute)

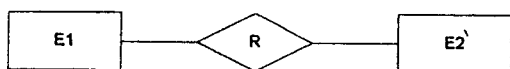


แอททริบิวต์ที่มีหลายค่า(Multivalued Attribute)

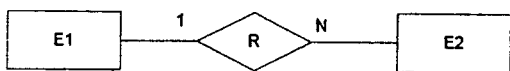


แอททริบิวต์ที่ประกอบด้วยหลายแอททริบิวต์ย่อย(Composite Attribute)

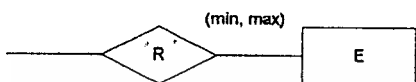
แอททริบิวต์ที่เกิดจากความสัมพันธ์ระหว่างแอททริบิวต์ด้วยกันเอง(Derived Attribute)

สัญลักษณ์ความหมาย

การขึ้นต่อกันอย่างสมบูรณ์ของ E2 ใน R



อัตราส่วนของความสัมพันธ์ 1:N สำหรับ E1 : E2 ใน R



ค่าคงที่ของโครงสร้าง (min, max) บนการขึ้นต่อกันของ E ใน R



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระบบทะเบียนรุ่น

3.1 วัตถุประสงค์ของระบบทะเบียนรุ่น

เพื่อเก็บข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ โดยสามารถแบ่งการทำงานได้ 2 หัวข้อคือ

- 3.1.1 การเข้าสู่ข้อมูลรายละเอียดของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
- 3.1.2 การปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

3.2 ลักษณะของระบบทะเบียนรุ่น

เป็นระบบที่จัดเก็บข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ โดยผู้ที่สนใจข้อมูลดังกล่าวสามารถเข้าสู่โดยผ่านทางระบบอินเทอร์เน็ตได้โดยตรง

3.3 รายละเอียดของระบบทะเบียนรุ่น

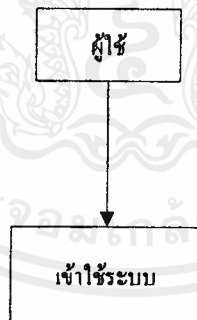
- 3.3.1 การเข้าสู่ข้อมูลรายละเอียดของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ทุกๆคนที่สามารถทำการติดต่อสื่อสารบนเว็ลด์ไวด์เว็บ ได้จะสามารถเข้าสู่ข้อมูลรายละเอียดของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ได้ ไม่มีการกำหนด Permission ในการเข้าสู่ข้อมูลรายละเอียดของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ นั่นคือใครๆก็สามารถเข้าสู่ข้อมูลรายละเอียดของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ได้
- 3.3.2 การปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้วของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
 - 3.3.2.1 ผู้ที่สามารถทำการปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ในระบบ ได้คือ อาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชา

วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์เท่านั้นผู้อื่นไม่สามารถทำการปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้วของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ได้ ดังนั้นจึงต้องมีการตรวจสอบ Security ในการการปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ด้วย

- 3.3.2.2 การปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์จะไม่สามารถทำการปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้วผู้อื่นได้ หมายความว่า อาจารย์และนักศึกษาที่จบการศึกษาไปแล้วสามารถที่จะทำการตั้งคำถามและหัวข้อคำถามได้เฉพาะในส่วนของตนเองเท่านั้น ไม่สามารถตั้งคำถามหรือหัวข้อคำถามให้อาจารย์หรือนักศึกษาที่จบการศึกษาไปแล้วท่านอื่นๆได้

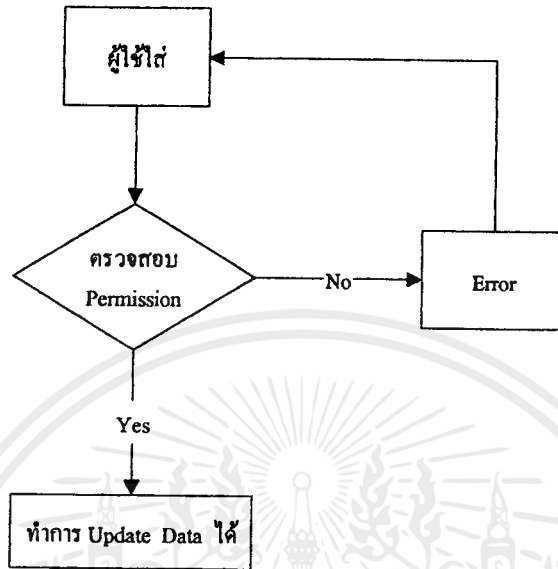
3.4 แผนผังลำดับขั้นตอนการดำเนินงานของระบบทะเบียนรุ่น

- 3.4.1 การเข้าสู่ข้อมูลรายละเอียดของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์



รูปที่ 3-1 การเข้าสู่ข้อมูลระบบทะเบียนรุ่น

3.4.2 การปรับปรุงแก้ไขเพิ่มเติมข้อมูลรายละเอียดต่างๆของอาจารย์และนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์



รูปที่ 3-2 การปรับปรุงแก้ไขเพิ่มเติมข้อมูลในระบบทะเบียนรุ่น

บทที่ 4

ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

4.1 วัตถุประสงค์ของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

เพื่อให้อาจารย์ภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สามารถถามคำถามต่างๆหรือสั่งการบ้านผ่านระบบอินเทอร์เน็ตได้ โดยสามารถแบ่งการทำงานได้ 3 หัวข้อคือ

- 4.1.1 การตั้งหัวข้อคำถามและคำถามของอาจารย์
- 4.1.2 การยกเลิกหัวข้อคำถามและคำถามของอาจารย์
- 4.1.3 การเข้าสู่รายละเอียดของคำถามของอาจารย์
- 4.1.4 การตอบคำถามของอาจารย์ผ่านทางระบบอินเทอร์เน็ต

4.2 ลักษณะของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

เป็นระบบที่อาจารย์ภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จะเป็นผู้ตั้งคำถามที่น่าสนใจในเรื่องต่างๆโดยที่ใครๆก็สามารถเข้ามาตอบคำถามโดยผ่านทางระบบอินเทอร์เน็ตได้ ซึ่งผู้ตอบคำถามไม่จำเป็นต้องเป็นคนภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ก็ได้ หมายความว่าไม่ว่าใครๆก็สามารถตอบคำถามได้

4.3 รายละเอียดของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

4.3.1 การตั้งหัวข้อคำถามและคำถามของอาจารย์

- 4.3.1.1 ผู้ที่สามารถทำการตั้งหัวข้อคำถามและคำถามในระบบได้คือ อาจารย์ภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์เท่านั้นผู้อื่นไม่สามารถทำการตั้งหัวข้อคำถามและคำถามของอาจารย์ได้ ดังนั้นจึงต้องมีการตรวจสอบสิทธิในการตั้งคำถามหรือหัวข้อคำถามด้วย
- 4.3.1.2 การตั้งคำถามและหัวข้อคำถามของอาจารย์จะไม่สามารถทำการตั้งคำถามหรือหัวข้อคำถามให้กับอาจารย์ผู้อื่นได้ หมายความว่า อาจารย์สามารถที่จะทำการตั้งคำถามและหัวข้อคำถามได้เฉพาะในส่วนของตนเองเท่านั้น ไม่สามารถตั้งคำถามหรือหัวข้อคำถามให้อาจารย์ท่านอื่นๆได้
- 4.3.1.3 การตั้งคำถามสามารถตั้งไว้ที่ใดก็ได้ในส่วนของตนเอง หมายความว่า คำถามที่ตั้งจะมีหัวข้อคำถามหรือไม่มีหัวข้อคำถามรองรับก็ได้

4.3.2 การยกเลิกหัวข้อคำถามและคำถามของอาจารย์

4.3.2.1 เช่นเดียวกับการตั้งหัวข้อคำถามและคำถามของอาจารย์คือ ผู้ที่สามารถทำการยกเลิกหัวข้อคำถามและคำถามในระบบได้คือ อาจารย์ภายในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์เท่านั้นผู้อื่น ไม่สามารถทำการยกเลิกหัวข้อคำถามและคำถามของอาจารย์ได้ ดังนั้นจึงต้องมีการตรวจสอบ Seสิทธิในการยกเลิกคำถามหรือหัวข้อคำถามด้วย

4.3.2.2 การยกเลิกคำถามและหัวข้อคำถามของอาจารย์จะไม่สามารถทำการยกเลิกคำถามหรือหัวข้อคำถามให้กับอาจารย์ผู้อื่นได้ หมายความว่า อาจารย์สามารถที่จะทำการยกเลิกคำถามและหัวข้อคำถามได้เฉพาะในส่วนของตนเองเท่านั้น ไม่สามารถยกเลิกคำถามหรือหัวข้อคำถามให้อาจารย์ท่านอื่นๆ ได้

4.3.2.3 การยกเลิกหัวข้อคำถามสามารถยกเลิกที่ใดก็ได้ในส่วนของตนเอง หมายความว่า หัวข้อคำถามที่ยกเลิกจะมีหัวข้อคำถามย่อยและคำถามหรือไม่มีหัวข้อคำถามย่อยและคำถามอยู่ภายในก็ได้ โดยถ้ามีหัวข้อคำถามย่อยหรือคำถามอยู่ภายในเมื่อทำการยกเลิกหัวข้อคำถามนั้นๆ หัวข้อคำถามย่อยและคำถามที่อยู่ภายในหัวข้อคำถามนั้นๆจะถูกยกเลิกไปด้วยโดยปริยาย

4.3.3 การเข้าดูรายละเอียดของคำถามของอาจารย์

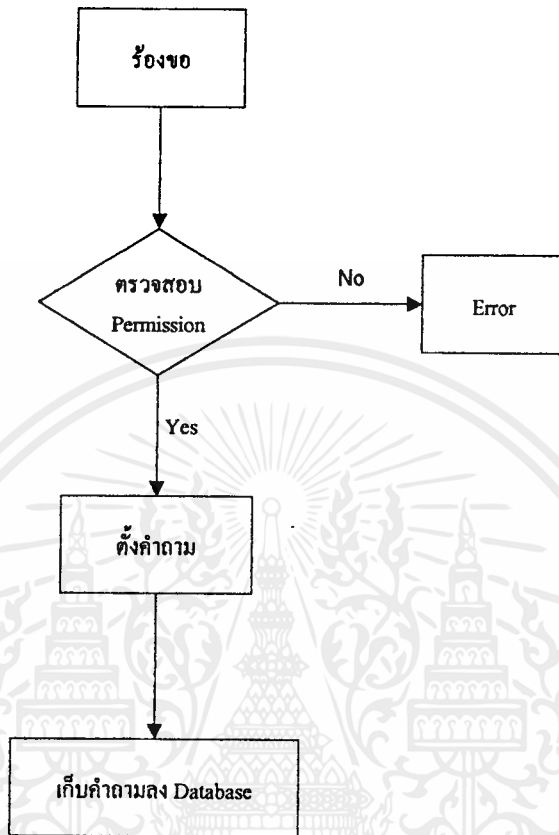
บุคคลทั่วไปจะสามารถเข้าดูรายละเอียดของคำถามของอาจารย์ได้ ไม่มีการกำหนดสิทธิในการเข้าดูรายละเอียดของคำถามของอาจารย์ นั่นคือใครๆก็สามารถเข้าดูได้

4.3.4 การตอบคำถามของอาจารย์ผ่านทางระบบอินเทอร์เน็ต

เช่นเดียวกับการเข้าดูรายละเอียดของคำถามของอาจารย์ นั่นคือ ใครๆก็สามารถทำการตอบคำถามของอาจารย์ผ่านทางระบบอินเทอร์เน็ตได้

4.4 แผนผังลำดับขั้นตอนการดำเนินงานของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

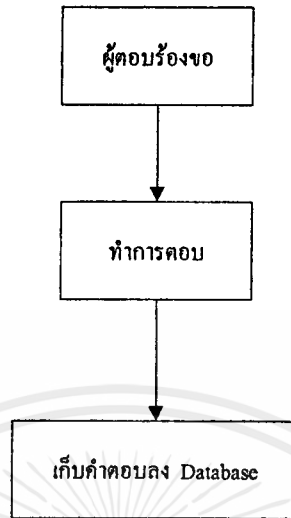
4.4.1 การตั้งหัวข้อคำถามและคำถามของอาจารย์



รูปที่ 4-1 การตั้งหัวข้อคำถามและคำถามของอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การตอบคำถามของอาจารย์ผ่านทางระบบอินเทอร์เน็ต



รูปที่ 4-2 การตอบคำถามในระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ระบบการจัดการนัดหมายการประชุม

5.1 วัตถุประสงค์ของระบบการจัดการนัดหมายการประชุม

เพื่อให้อาจารย์และนักศึกษาภายในภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ รวมถึงนักศึกษาที่จบการศึกษาแล้วของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สามารถที่ทำการจะนัดหมายการประชุมหรือทำการนัดพบกัน โดยผ่านทางระบบอินเทอร์เน็ตได้ โดยสามารถแบ่งการทำงานได้ 6 หัวข้อคือ

- 5.1.1 การกำหนดหรือสร้างการนัดหมายการประชุม
- 5.1.2 การยกเลิกการนัดหมายการประชุม
- 5.1.3 การจัดการเกี่ยวกับการนัดหมายการประชุม
- 5.1.4 การจัดตั้งกลุ่มสำหรับนัดหมายการประชุม
- 5.1.5 การยกเลิกกลุ่มที่ใช้ในการนัดหมายการประชุม
- 5.1.6 การแสดงผลการนัดหมายการประชุม

5.2 ลักษณะของระบบการจัดการนัดหมายการประชุม

เป็นระบบที่ช่วยในการจัดการนัดหมายการประชุมของอาจารย์และนักศึกษารุ่นปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ โดยผู้ใช้สามารถที่จะทำการกำหนดกลุ่มสำหรับการนัดหมายได้ด้วยตนเอง

5.3 รายละเอียดของระบบการจัดการนัดหมายการประชุม

5.3.1 การกำหนดหรือสร้างการนัดหมายการประชุม

อาจารย์และนักศึกษารุ่นปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ทุกคนสามารถที่จะกำหนดการนัดหมายการประชุมที่ตนต้องการขึ้นมาได้ โดยการนัดหมายประชุมดังกล่าวจะถูกส่งไปยังผู้เข้าร่วมประชุมทุกคนที่ถูกระบุไว้

5.3.2 การยกเลิกการนัดหมายการประชุม

การนัดหมายประชุมที่ถูกส่งไปแล้วนั้นสามารถที่จะยกเลิกได้โดยผู้ที่ทำการกำหนดการนัดหมายประชุมเท่านั้น ผู้อื่นไม่สามารถทำการยกเลิกได้ และการยกเลิกดังกล่าวจะถูกส่งไปยังผู้เข้าร่วมประชุมทุกคนให้ทราบด้วย

5.3.3 การจัดการเกี่ยวกับการนัดหมายการประชุม

เมื่อมีการกำหนดการนัดหมายการประชุมขึ้นมาแล้ว จะมีการตรวจสอบในเรื่องเวลาของผู้ที่เข้าร่วมประชุมทุกคนว่าว่างในช่วงเวลาที่นัดหมายกันไว้หรือไม่ ถ้าไม่ว่างก็จะมีการแจ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ผู้นัดหมายการประชุมทราบว่าผู้ที่ไม่ว่างคือใคร และไม่ว่างช่วงไหน ซึ่งช่วยให้ผู้นัดประชุมสามารถที่จะทำการตัดสินใจได้ว่าจะทำอย่างไรต่อไป

5.3.4 การจัดตั้งกลุ่มสำหรับนัดหมายการประชุม

อาจารย์และนักศึกษาปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สามารถที่จะทำการจัดตั้งกลุ่มสำหรับการนัดหมายการประชุมได้ โดยที่สมาชิกภายในกลุ่มจะต้องเป็นอาจารย์และนักศึกษาปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์เท่านั้น

5.3.5 การยกเลิกกลุ่มที่ใช้ในการนัดหมายการประชุม

อาจารย์และนักศึกษาปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ที่เป็นผู้จัดตั้งกลุ่มจะสามารถทำการยกเลิกกลุ่มนั้นๆได้ โดยที่ผู้อื่นที่ไม่ใช่ผู้จัดตั้งกลุ่มนั้นๆขึ้นมาจะไม่สามารถทำการยกเลิกกลุ่มนั้นๆได้ ดังนั้นจึงต้องมีการตรวจสอบสิทธิ์ในการยกเลิกกลุ่มด้วย

5.3.6 การแสดงผลการนัดหมายการประชุม

อาจารย์และนักศึกษาปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ทุกคนจะมีกล่องเก็บข้อมูลขาเข้าและขาออกเป็นของตนเอง ซึ่งในกล่องเก็บข้อมูลขาเข้าจะเก็บการนัดหมายการประชุมที่ถูกส่งมาและนำมาแสดงผลให้อาจารย์และนักศึกษาปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์คนนั้นๆทราบ ส่วนกล่องเก็บข้อมูลขาออกจะทำหน้าที่เก็บการนัดหมายการประชุมที่อาจารย์และนักศึกษาปัจจุบัน ตลอดจนนักศึกษาที่จบการศึกษาไปแล้ว ของภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์คนนั้นๆเป็นผู้ที่ส่งไปนัดผู้อื่น

5.4 ส่วนประกอบหลักของระบบการจัดการนัดหมายการประชุม

5.4.1 การกำหนดหรือสร้างการนัดหมายการประชุม

5.4.2 การยกเลิกการนัดหมายการประชุม

5.4.3 การจัดการเกี่ยวกับการนัดหมายการประชุม

5.4.4 การจัดตั้งกลุ่มสำหรับนัดหมายการประชุม

5.4.5 การยกเลิกกลุ่มที่ใช้ในการนัดหมายการประชุม

5.4.6 การแสดงผลการนัดหมายการประชุม

บทที่ 6

การพัฒนาโครงการ

6.1 ส่วนประกอบของโครงการ

ส่วนประกอบหลักของโครงการประกอบไปด้วย 3 ส่วนคือ

1. ระบบทะเบียนรุ่น
2. ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต
3. ระบบการนัดหมายการประชุม
4. การเปลี่ยนรหัสเวิร์คของผู้ใช้

โดยลักษณะสำคัญของแต่ละระบบคือ

6.2 รูปแบบการทำงาน

รูปแบบการทำงานของโปรแกรมแต่ละส่วน เป็นดังนี้



6.2.1 ระบบทะเบียนรุ่น

จะมีหน้าจอหลักในการทำงานคือ

Applet Viewer: form1.class

Applet:

Com. Member Info

Class: Student

Only Year: 2537 Show

Member List:

ID#	Name
37014001	Kanokporn Nabangplasol
37014011	Korkit Veeraarchakun
37014014	Karn Sukwong
37014215	Boonchai Rattanawaranun
37014216	Boonchai lammetta
37014050	Jakkapong Siriboon
37014040	Kananart Nitithananut
37014102	Thapanee Srichoung
34014402	Viyada Anantamongkhon

ID#: 37014215 Details

Applet started.

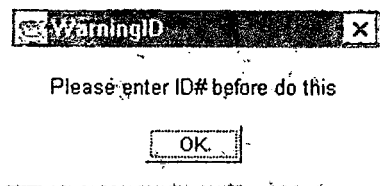
รูปที่ 6-1 หน้าจอหลักของระบบทะเบียนรุ่น (Com-Member Info)

ซึ่งผู้ใช้สามารถเลือกกลุ่มเป้าหมายที่ต้องการเข้าสู่ข้อมูลได้โดยเลือกที่คอมโบบ็อกซ์“Class :” ซึ่งประกอบไปด้วย 2 กลุ่มคือ อาจารย์ และนักศึกษาซึ่งถ้าผู้ใช้เลือก Class Student ก็จะมีช่องสำหรับให้กรอกเฉพาะปีที่ต้องการดูรายชื่อได้ โดยที่ลักษณะการเข้าสู่รายละเอียดของข้อมูลสามารถทำได้ 2 ทางคือ

1. โดยการดับเบิ้ลคลิกที่ชื่อของบุคคลที่ต้องการดูรายละเอียดของข้อมูลจากลิสต์หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โดยการคลิกที่ปุ่ม Details ซึ่งการคลิกที่ปุ่ม Details นี้จะต้องทำการใส่รหัสของบุคคลที่ต้องการดูรายละเอียดของข้อมูลด้วย หรือไม่ก็จะต้องทำการเลือกชื่อของบุคคลที่ต้องการดูรายละเอียดของข้อมูลจากลิสท์ก่อน ถ้าผู้ใช้ไม่ได้ทำอย่างใดอย่างหนึ่งก่อนคลิกที่ปุ่ม Details จะมีข้อความเตือนดังนี้ปรากฏขึ้น



รูปที่ 6-2 หน้าจอแสดงข้อความเตือน (Warning Message)

หลังจากเข้าดูรายละเอียดของข้อมูลของบุคคลนั้นๆ ได้แล้ว ก็จะปรากฏหน้าจอแสดงรายละเอียดดังนี้

รูปที่ 6-3 หน้าจอแสดงข้อมูลรายละเอียดของบุคคล (Details Form)

โดยผู้ใช้สามารถทำการแก้ไขข้อมูลของตนเองได้ โดยการคลิกที่ปุ่ม Edit เมื่อต้องการแก้ไข จากนั้นระบบจะสอบถาม Security ในการแก้ไขดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6-4 หน้าจอการตรวจสอบ Security ในการแก้ไข(Edit Security)

ถ้ารหัสและพาสเวิร์ดถูกต้องและมี Permission ในการ Edit ระบบก็จะอนุญาตให้ทำการ Edit ได้ โดยหน้าจอในการ Edit จะเป็นดังนี้

Personal Information

Name: Academic Year:

Address: Link Address:

Tel.(Home): Others Link:

Office Place now: Tel.(Office):

FAX:

Email:

รูปที่ 6-5 หน้าจอการแก้ไขข้อมูลรายละเอียดของบุคคล (Edit Form)

เมื่อผู้ใช้งานทำการแก้ไขข้อมูลของตนเสร็จเรียบร้อยแล้ว และผู้แก้ไขทำการคลิกปุ่ม OK แล้ว ตัวระบบก็จะทำการแก้ไขข้อมูลให้โดยอัตโนมัติ แล้วจึงกลับไปยังหน้าจอหลักให้

6.2.2 ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

เริ่มต้นก่อนที่จะเข้าใช้งาน ระบบจะสอบถามสิทธิในการเข้าใช้งานระบบดังนี้

รูปที่ 6-6 หน้าจอการล็อกอินเข้าใช้ระบบ (User Security Form)

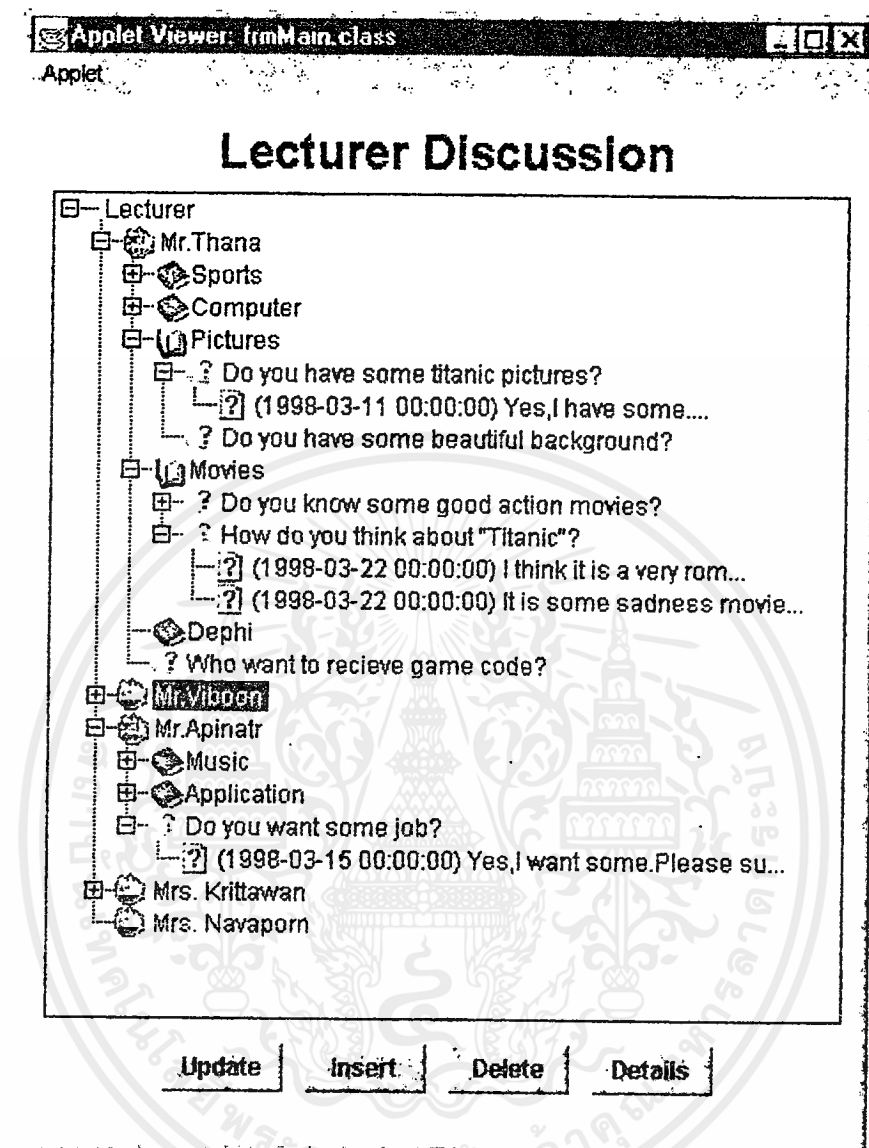
โดยสิทธิในการเข้าใช้งานระบบจะแบ่งเป็น 2 แบบคือ

1. เข้าใช้งานระบบแบบผู้ที่มาเยือน
2. เข้าใช้งานระบบแบบอาจารย์

ถ้าเข้าใช้ระบบแบบผู้ที่มาเยือนผู้เข้าใช้ระบบจะสามารถทำได้แค่เข้าไปดูรายละเอียดต่างๆ และ Add NewReply ได้เท่านั้น แต่ไม่สามารถทำการ แก้ไข, เพิ่มเติม, ลบหัวข้อคำถามหรือคำถามของอาจารย์ได้

ถ้าเข้าใช้ระบบแบบอาจารย์ผู้เข้าใช้นอกจากจะสามารถเข้าไปดูรายละเอียดต่างๆ และ Add New Reply ได้แล้ว ยังสามารถทำการเพิ่ม, แก้ไข, ลบหัวข้อคำถามหรือคำถามในส่วนของตนเองได้อีกด้วย

หลังจากระบบทำการตรวจสอบสิทธิในการเข้าใช้งานให้แล้ว จะปรากฏหน้าจอหลักดังนี้



รูปที่ 6-7 หน้าจอหลักของระบบ Lecturer Discussion (Main Form)

โดยเป็นการแสดงหัวข้อคำถาม , คำถาม , คำตอบ และชื่อของอาจารย์ที่เป็นเจ้าของออกมาในรูปแบบต้นไม้ โดยที่

- โหนดที่มีรูปหน้าคนอยู่ด้านหน้า เป็นโหนดที่แสดงชื่อของอาจารย์
- โหนดที่มีรูปหนังสืออยู่ด้านหน้า เป็นโหนดที่แสดงชื่อของหัวข้อคำถาม
- โหนดที่มีรูปเครื่องหมายคำถามอยู่ด้านหน้า เป็นโหนดที่แสดงคำถาม
- โหนดที่มีรูปกระดาษที่มีเครื่องหมายคำถามอยู่ด้านหน้า เป็นโหนดที่แสดงคำตอบของคำถาม

(Reply) และมีปุ่มต่างๆที่ใช้ทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ปุ่ม Update

ใช้ในการปรับปรุงหรือแก้ไขเพิ่มเติมหรือเปลี่ยนชื่อหัวข้อคำถามหรือคำถามของอาจารย์ท่านนั้นๆ

2. ปุ่ม Insert

ใช้ในการเพิ่มหัวข้อคำถามหรือคำถามใหม่เข้าไปในฐานข้อมูลและค้น ไม่โดยสามารถทำการเพิ่มได้เฉพาะของอาจารย์ท่านนั้นๆเท่านั้น

3. ปุ่ม Delete

ใช้ในการลบหัวข้อคำถามหรือคำถามของอาจารย์ท่านนั้นๆทิ้งไป

4. ปุ่ม Details

ใช้ในการเข้าสู่รายละเอียดของคำถามและคำตอบ

โดยที่หน้าจอหลักนี้จะมีการทำงานที่สำคัญคือ

1. การแก้ไขปรับปรุง (Update) หัวข้อคำถามหรือคำถาม

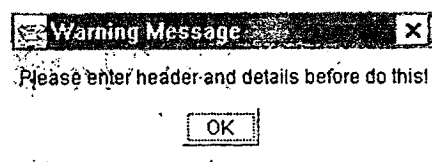
อาจารย์สามารถทำการอัปเดตหรือ เปลี่ยนชื่อหัวข้อคำถามหรือคำถามของตนเองได้โดยการคลิกที่ปุ่ม Update จากนั้นจะปรากฏหน้าจอแสดงรายละเอียดของหัวข้อคำถามหรือคำถามนั้นๆขึ้นมาให้ทำการแก้ไขได้ ดังนี้

รูปที่ 6-8 หน้าจอการปรับปรุงแก้ไขหัวข้อคำถาม (Forum Update Form)

รูปที่ 6-9 หน้าจอการปรับปรุงแก้ไขคำถาม (Question Update Form)

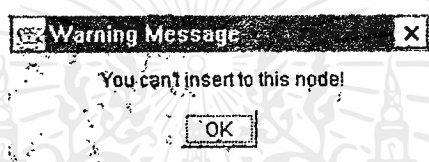
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแก้ไขหัวข้อคำถามหรือคำถามเสร็จแล้ว และผู้ใช้คลิก ปุ่ม OK ระบบจะทำการแก้ไขข้อมูลของหัวข้อคำถามหรือคำถามนั้นๆลงฐานข้อมูลให้โดยอัตโนมัติ แต่ถ้าตอนแก้ไขผู้ใช้ใส่ฟิลด์สำคัญต่างๆไม่ครบถ้วน ระบบก็จะแจ้งข้อความเตือนขึ้นมาเตือนดังนี้



รูปที่ 6-10 หน้าจอแสดงข้อความเตือน (Warning Message)

ถ้าผู้ใช้พยายามที่จะแก้ไขหัวข้อคำถามหรือคำถามของผู้อื่น ระบบก็จะแจ้งข้อความเตือนขึ้นมาเตือนดังนี้



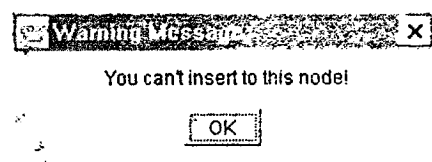
รูปที่ 6-11 หน้าจอแสดงข้อความเตือน (Warning Message)

2. การเพิ่ม (Insert) หัวข้อคำถามหรือคำถาม

อาจารย์สามารถทำการเพิ่มหัวข้อคำถามหรือคำถามใหม่ๆเข้าไปได้โดยการคลิกที่ปุ่ม Insert จากนั้นจะปรากฏเมนูในการเพิ่มมาให้เลือกรูปแบบการเพิ่มซึ่งสอดคล้องกับตำแหน่งที่จะทำการเพิ่มหัวข้อคำถามหรือคำถามดังนี้

- 2.1 ถ้าตำแหน่งที่จะทำการเพิ่มเป็นชื่อของอาจารย์ ผู้ใช้จะสามารถทำการเพิ่ม ได้ทั้งหัวข้อคำถามและคำถาม
- 2.2 ถ้าตำแหน่งที่จะทำการเพิ่มเป็นหัวข้อคำถาม ผู้ใช้จะสามารถทำการเพิ่ม ได้ทั้งหัวข้อคำถามและคำถาม
- 2.3 ถ้าตำแหน่งที่จะทำการเพิ่มเป็นคำถาม ผู้ใช้จะสามารถทำการเพิ่ม ได้แค่เฉพาะคำตอบสำหรับคำถามนั้นๆเท่านั้น

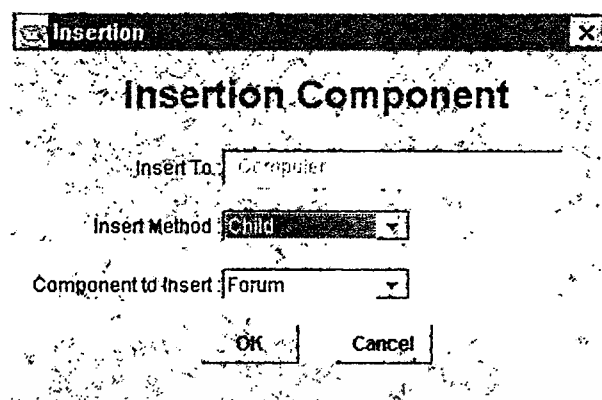
โดยถ้าผู้ใช้พยายามที่จะทำการเพิ่มหัวข้อคำถามหรือคำถามใส่ในของผู้อื่นซึ่งไม่ใช่ของตนเอง ระบบจะแจ้งข้อความเตือนขึ้นมาดังนี้



รูปที่ 6-12 หน้าจอแสดงข้อความเตือน (Warning Message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอของการเพิ่มที่กล่าวข้างต้นมีลักษณะดังนี้

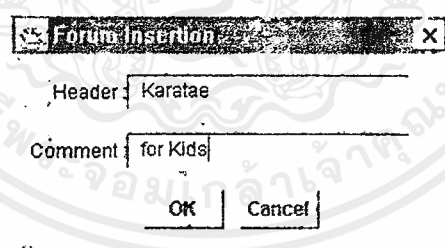


รูปที่ 6-13 หน้าจอเมนูการเพิ่ม (Insert Menu Form)

โดย Insert Method หมายถึง วิธีการในการเพิ่มซึ่งมีอยู่ 2 วิธีคือ

1. Child หมายความว่าคอมโพเนนต์ที่จะทำการเพิ่มจะเป็นลูกของโหนดที่ถูกเลือกอยู่
2. Sibling หมายความว่าคอมโพเนนต์ที่จะทำการเพิ่มจะเป็นโหนดถัดไป (โหนดข้างๆ) ของ no โหนดที่ถูกเลือกอยู่ และ Insert Component หมายถึง สิ่งที่ต้องการเพิ่มเข้าไปเช่น หัวข้อคำถาม (Forum) หรือ คำถาม Question) เป็นต้น

ถ้าเป็นการเพิ่มหัวข้อคำถาม (Forum) หลังจากเลือกจากเมนูและ OK แล้วจะปรากฏแบบฟอร์มสำหรับสร้างหัวข้อคำถามใหม่ดังนี้



รูปที่ 6-14 หน้าจอการเพิ่มหัวข้อคำถาม (Forum Insertion Form)

ถ้าเป็นการเพิ่มคำถามหลังจากเลือกที่เมนูแล้ว OK จะปรากฏแบบฟอร์มสำหรับสร้างคำถามใหม่ดังนี้

รูปที่ 6-15 หน้าจอการเพิ่มคำถาม (Question Insertion Form)

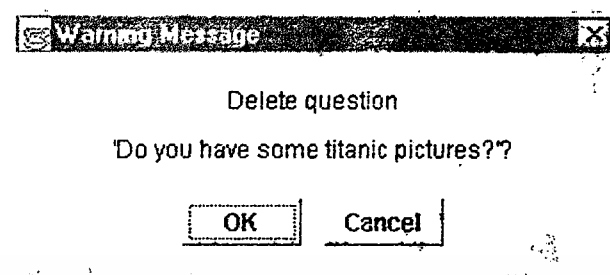
โดยผู้ใช้จะต้องใส่ฟิลด์ที่สำคัญๆ ในแบบ Form ให้ครบทุกฟิลด์ก่อนทำการคลิกปุ่ม OK เพื่อทำการเพิ่มเสมอ มิฉะนั้นระบบจะแสดงข้อความเตือนดังนี้

รูปที่ 6-16 หน้าจอแสดงข้อความเตือน (Warning Message)

3. การลบ (Delete) หัวข้อคำถาม , คำถาม และคำตอบของคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจารย์สามารถทำการลบหัวข้อคำถาม , คำถาม และคำตอบในคำถามของตนเองได้โดยการคลิกที่ปุ่ม Delete เมื่อต้องการลบหัวข้อคำถาม , คำถาม และคำตอบของคำถามนั้นๆทิ้งไป จากนั้นระบบจะเตือนให้ผู้ใช้งานอีกครั้งว่าต้องการลบหัวข้อคำถามหรือ คำถามนั้นๆทิ้งจริงหรือไม่ ดังนี้



รูปที่ 6-17 หน้าจอถามความแน่ใจในการลบ

ถ้าผู้ใช้คลิก OK จะเป็นการยืนยันว่าอนุญาตให้ระบบทำการลบหัวข้อคำถาม , คำถาม หรือคำตอบของคำถามนั้นๆทิ้งได้ ระบบก็จะทำการลบหัวข้อคำถาม , คำถาม หรือคำตอบของคำถามนั้นๆทิ้งทันที

เช่นเดียวกับการเพิ่มและการแก้ไขคือถ้าผู้ใช้พยายามที่จะทำการลบหัวข้อคำถาม , คำถาม หรือคำตอบของคำถามของคนอื่นซึ่งไม่ใช่ของตนเอง ระบบจะแสดงข้อความเตือนดังนี้



รูปที่ 6-18 หน้าจอแสดงข้อความเตือน (Warning Message)

4. การขอรายละเอียด (Details) ของหัวข้อคำถามหรือคำถาม

การขอรายละเอียดของคำถามจะต่างจากการเพิ่ม, การลบ, การแก้ไขตรงที่ ไม่ว่าใครก็สามารถที่จะทำการ ขอรายละเอียดของหัวข้อคำถามหรือคำถามได้ทั้งนั้น ไม่มีการกำหนดสิทธิในการเข้าดู

การขอรายละเอียดของคำถามสามารถทำได้ 2 วิธีคือ

4.1 ขอรายละเอียดของคำถาม โดยการดับเบิลคลิกที่คำถามนั้นๆ

4.2 ขอรายละเอียดของคำถาม โดยการคลิกที่ปุ่ม Details

ทั้งสองวิธีนี้จะให้ผลที่เหมือนกันคือ ระบบจะทำการเปิดแบบฟอร์มแสดงรายละเอียดของคำถามขึ้นมาดังนี้

Question Details

Question Date: 1997-02-17 00

Question By: Mr.Thana

Header: Do you have some titanic pictures?

Details :

If you have some, please give me for the picture of "Rose"

Reply: 1 Messages

Date: 1998-03-11 00:00:00

Replier: Chal

Replier Email: s7014215@diamond.ce.kmitl.ac.th

Details:

Yes, I have some.

New Ans Cancel

รูปที่ 6-19 หน้าจอแสดงรายละเอียดของคำถาม (Question Details Form)

โดยจะเป็นรายละเอียดทั้งหมดของคำถาม (Question) ที่ถูกเลือกนั้นๆ รวมไปถึงการตอบกลับของคำถามนั้นๆ และจำนวนของการตอบกลับมาด้วย

จาก แบบฟอร์มแสดงรายละเอียดนี้ผู้ใช้สามารถทำการเพิ่มคำตอบของคำถามนั้นๆ ได้โดยการคลิกที่ปุ่ม NewAns ก็จะปรากฏแบบฟอร์มสำหรับกรอกคำตอบใหม่นี้ โดยที่ผู้ใช้จะต้องกรอกรายละเอียดให้ครบทุกฟิลด์มิฉะนั้นระบบจะแจ้งข้อความเตือนดังนี้

Warning Message

Please enter 'Replier Name' before do this!

OK

รูปที่ 6-20 หน้าจอแสดงข้อความเตือน (Warning Message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reply Question

New Replyment

Replier Name : Chai

Replier Email : s7014215@diamond.ce.kmitl.ac.th

Answer Details :

I don't like pop music.

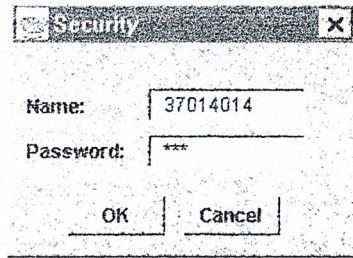
OK Cancel

รูปที่ 6-21 หน้าจอการเพิ่มคำตอบ (Reply Form)

และถ้าผู้ใช้กรอกข้อมูลครบทุกฟิลด์แล้ว เมื่อผู้ใช้ทำการคลิกปุ่ม OK ระบบจะทำการเพิ่มคำตอบนั้นๆลงฐานข้อมูลทันที

6.2.3 ระบบการจัดการนัดหมายการประชุม

เริ่มต้นก่อนที่จะเข้าใช้งาน ระบบจะสอบถามถึงสิทธิในการเข้ามาใช้งานของผู้ที่เรียกใช้โปรแกรม



รูปที่ 6-22 แสดงหน้าจอในการสอบถามสิทธิในการใช้งาน

ถ้าผู้ใช้สามารถใส่ข้อมูลที่ต้องการ(ชื่อและรหัสผ่าน)ได้ถูกต้อง ผู้ใช้ก็สามารถจะใช้งานโปรแกรมในขั้นต่อไปได้

การควบคุมการรับส่งการนัดหมายการประชุมโดยใช้ทูลบาร์



รูปที่ 6-23 แสดงทูลบาร์หลักของโปรแกรม

การสร้างการนัดหมายการประชุม (ทูลบาร์ที่ 1)

ใช้ในการสร้างการนัดหมายการประชุมขึ้นมาใหม่ จะได้น้ำจอในรูปที่ 8

การดูรายละเอียดการนัดหมายการประชุม (ทูลบาร์ที่ 2)

ใช้ในการดูรายละเอียดเกี่ยวกับการนัดหมายการประชุมที่เลือกเอาไว้ สามารถทำงานได้ทั้งส่วนที่ทำการรับการนัดหมายการประชุมและส่วนที่ทำการส่งการนัดหมายการประชุมออกไป โดยจะได้น้ำจอ ดังรูปที่ 6-25 และรูปที่ 6-28 ขึ้นอยู่กับว่าใช้งานอยู่ในส่วนใด

การเลือกการนัดหมายการประชุมที่ยังไม่ได้อ่าน (ทูลบาร์ที่ 3)

ใช้ในการเลื่อนแถบสีไปยังการนัดหมายการประชุมที่ผู้ใช้ยังไม่ได้เข้าไปอ่านรายละเอียด

การลบการนัดหมายการประชุม (ทูลบาร์ที่ 4)

ใช้ในการลบการนัดหมายการประชุมที่เลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

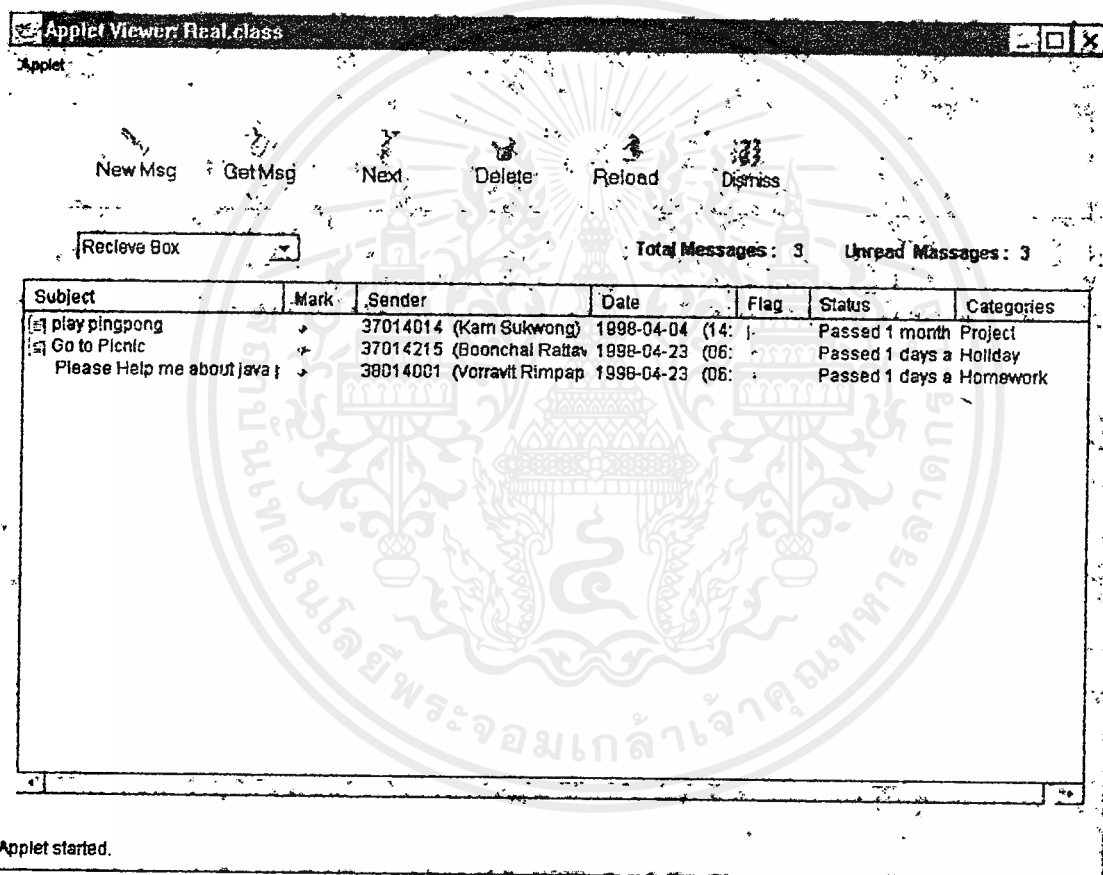
การเรียกข้อมูลจากฐานข้อมูลมาแสดงผล (ทูลบาร์ที่ 5)

ใช้ในการดึงการนัดหมายการประชุมที่ถูกเก็บไว้ในฐานข้อมูลขึ้นมาแสดงผลบนหน้าจอ
การยกเลิกการนัดหมายการประชุม (ทูลบาร์ที่ 6)

ใช้ในการยกเลิกการนัดหมายการประชุม ซึ่งจะใช้ได้เฉพาะเมื่อทำงานอยู่ในส่วนที่ทำการส่งการ
นัดหมายการประชุมออกไป โดยการยกเลิกการนัดหมายประชุมนี้จะถูกส่งไปยังผู้ที่เข้าร่วมการประชุมทุก
คนให้ทราบไว้

ส่วนที่ทำหน้าที่แสดงผลการนัดหมายการประชุมที่ได้รับมา

ส่วนนี้จะเป็นส่วนแรกสุดที่ผู้ใช้เห็นหลังจากที่ผ่านการตรวจสอบสิทธิในการใช้งานแล้ว



รูปที่ 6-24 แสดงหน้าจอแรกที่เข้าสู่โปรแกรม

รูปแบบการแสดงถึงการนัดหมายการประชุมที่ผู้ใช้ได้รับมาจะประกอบด้วยคอลัมน์ต่างๆดังนี้

1. Subject หมายถึง ชื่อของการประชุมที่ผู้ใช้ได้รับมา ส่วนรูปที่อยู่ด้านหน้าของชื่อการประชุม
จะแสดงถึงสถานะของการประชุม โดยมี 2 ชนิดคือ ถ้าเป็นรูปกระดานนั้นหมายถึงการประชุมนั้นอยู่ใน
สถานะปกติ แต่ถ้าเป็นรูปกากบาทสีแดงแสดงถึงการนัดหมายการประชุมนั้นได้ถูกยกเลิกไปแล้ว โดยผู้
ที่ทำการนัดประชุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Mark หมายถึง สถานะที่บอกว่าผู้ใช้ได้ทำการอ่านการนัดหมายการประชุมนั้นๆ ไปแล้วหรือยัง ถ้าเป็นรูปปุ่มสีเขียวแสดงถึงว่าผู้ใช้ยังไม่ได้อ่านการนัดหมายการประชุมนั้นๆ แต่ถ้าเป็นรูปจุดแสดงถึงว่าผู้ใช้ได้ทำการอ่านการนัดหมายการประชุมนั้นๆ ไปแล้ว (การอ่านการนัดหมายการประชุมทำได้โดยดับเบิลคลิกการนัดหมายการประชุมซึ่งอยู่ในช่วงคอลัมน์นี้ หรือโดยการคลิกเมาส์ที่ปุ่ม Get Msg)

3. Sender หมายถึง ผู้ที่ทำการส่งการนัดหมายการประชุมมาถึงผู้ใช้แสดงในรูปของรหัสประจำตัว / ชื่อ-นามสกุล

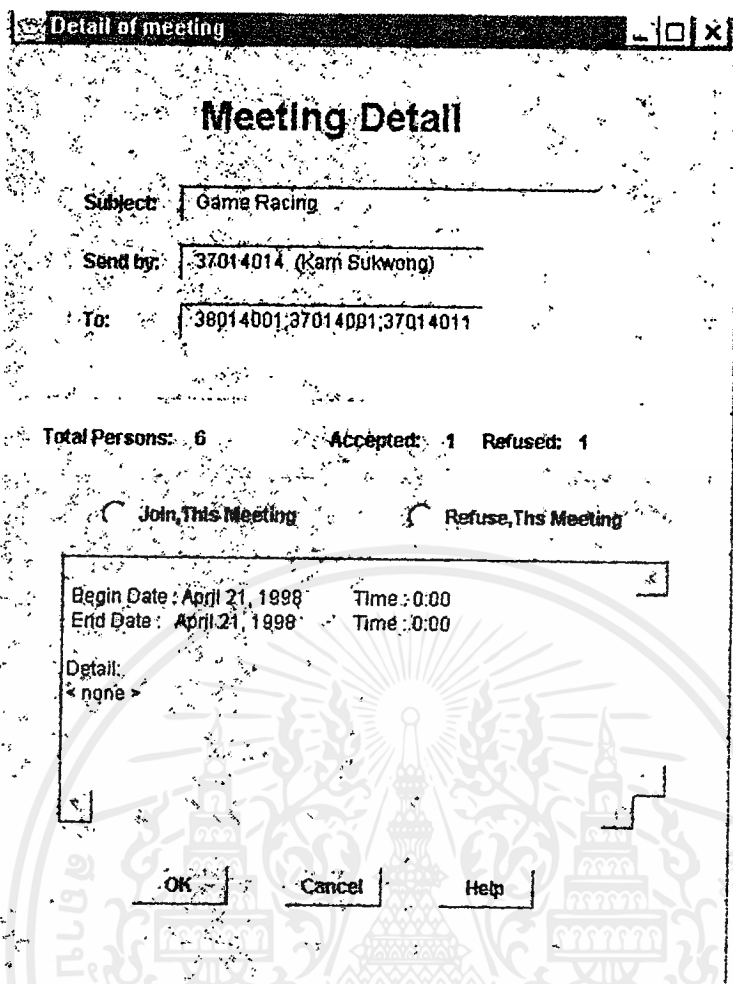
4. Date หมายถึง วัน-เวลาที่ได้รับการนัดหมายการประชุม

5. Flag หมายถึง ความสำคัญของการนัดหมายการประชุมที่ผู้ใช้เป็นผู้กำหนดขึ้น โดยที่ถ้าเป็นรูปธงหมายถึงการนัดหมายการประชุมนั้นสำคัญ แต่ถ้าเป็นรูปจุดหมายถึงผู้ใช้ไม่ได้ให้ความสำคัญ (การกำหนดความสำคัญทำได้โดยดับเบิลคลิกหัวข้อการนัดหมายการประชุมซึ่งอยู่ในช่วงคอลัมน์นี้)

6. Status หมายถึง การนัดหมายการประชุมนั้นจะมีขึ้นเมื่อไร เป็นไปได้ทั้งอดีตและอนาคต

7. Categories หมายถึง ประเภทของการนัดหมายการประชุม เช่น Job, Homework, Holiday ฯลฯ

ถ้าผู้ใช้ต้องการเข้าไปดูรายละเอียดของการนัดหมายการประชุมที่ได้รับ หรือต้องการตัดสินใจที่จะเข้าร่วมหรือปฏิเสธการประชุมก็สามารถทำได้โดยดับเบิลคลิกที่หัวข้อการนัดหมายการประชุมนั้นหรือโดยทำการคลิกที่ปุ่ม Get Msg ที่อยู่ด้านบนก็ได้ จากนั้นโปรแกรมก็จะแสดงหน้าจอใหม่ดังรูปด้านล่าง



รูปที่ 6-25 แสดงรายละเอียดของการนัดหมายการประชุม

จากรูปด้านบนจะเห็นได้ว่าผู้ใช้สามารถตัดสินใจเลือกที่จะเข้าร่วมหรือปฏิเสธก็ได้เป็นอย่างดีหนึ่งเท่านั้น หลังจากที่ตัดสินใจไปแล้วผลการตัดสินใจจะถูกส่งไปอัปเดตยังฐานข้อมูล ซึ่งผู้ที่ทำการนัดหมายก็จะสามารถเห็นผลการตัดสินใจของผู้ใช้ได้

ส่วนที่ทำหน้าที่ในการแสดงการนัดหมายการประชุมที่ผู้ใช้ได้ทำการส่งออกไป

ผู้ใช้จะสามารถเข้ามาดูการนัดหมายการประชุมที่ตนได้ส่งออกไปได้โดยการเปลี่ยนค่าที่อยู่ใน Check Box โดยทำการเชคค่าเป็น Send Box ซึ่งก็จะแสดงผลดังรูปด้านล่างนี้

Applet Viewer: Real.class

Applet

New Msg Get Msg Next Delete Reload Dismiss

Send To

Subject	Date	Total Receivers	Accept Receivers	Refuse Receivers	Status
Play tennis	1998-04-04 (14:50:03)	0	0	0	Passed 1 months ago
play pingpong	1998-04-04 (14:56:16)	0	0	0	Passed 1 months ago
test	1998-04-04 (16:05:21)	0	0	0	Passed 1 months ago
Go to Central	1998-04-23 (06:11:07)	0	0	0	Passed 1 days ago

Applet started.

รูปที่ 6-26 แสดงการนัดหมายการประชุมทั้งหมดที่ผู้ใช้ได้รับมา

รูปแบบการแสดงถึงการนัดหมายการประชุมที่ผู้ใช้ได้ส่งออกไปจะประกอบด้วยคอลัมน์ต่างๆดังนี้

1. Subject หมายถึง ชื่อของการประชุมที่ผู้ใช้ได้ส่งไป ส่วนรูปที่อยู่ด้านหน้าของชื่อการประชุมจะแสดงถึงสถานะของการประชุม โดยมี 2 ชนิดคือ ถ้าเป็นรูปกระดานนั้นหมายถึงการประชุมนั้นอยู่ในสถานะปกติ แต่ถ้าเป็นรูปกากบาทสีแดงแสดงถึงว่าผู้ใช้ได้ทำการยกเลิกการนัดหมายการประชุมนั้นไปแล้ว

2. Date หมายถึง วัน-เวลาที่ได้ส่งการนัดหมายการประชุมออกไป
3. Total Receivers หมายถึง จำนวนผู้ที่ได้รับการนัดหมายการประชุมที่ส่งไปทั้งหมด
4. Accept Receivers หมายถึง จำนวนผู้ที่ได้ตอบตกลงการนัดหมายการประชุมที่ส่งไปทั้งหมด
5. Refuse Receivers หมายถึง จำนวนผู้ที่ได้ตอบปฏิเสธการนัดหมายการประชุมที่ส่งไปทั้งหมด
6. Status หมายถึง การประชุมนั้นมีขึ้นเมื่อไรเมื่อเทียบกับเวลาปัจจุบัน

ถ้าผู้ใช้ต้องการดูผลการตัดสินใจของผู้ที่จะเข้าร่วมการประชุมก็สามารถทำได้โดยการดับเบิลคลิกที่หัวข้อการนัดหมายการประชุม ซึ่งก็จะปรากฏหน้าจอใหม่ดังรูปด้านล่าง

Receiver	Read	Accept	Refuse	Last login
38014001 (Vorravit Rimpapun)	<input type="checkbox"/>		No reply	
37014001 (Kanokpoom Seibea)	<input checked="" type="checkbox"/>	No reply	No reply	
37014011 (Korkit Verachakul)	<input checked="" type="checkbox"/>	No reply	No reply	
37014014 (Kam Sukwong)	<input checked="" type="checkbox"/>	No reply	No reply	
37014215 (Boonchai Rattawaranut)	<input type="checkbox"/>	No reply		
37014216 (Boonchai Eissaragkul)	<input checked="" type="checkbox"/>	No reply	No reply	

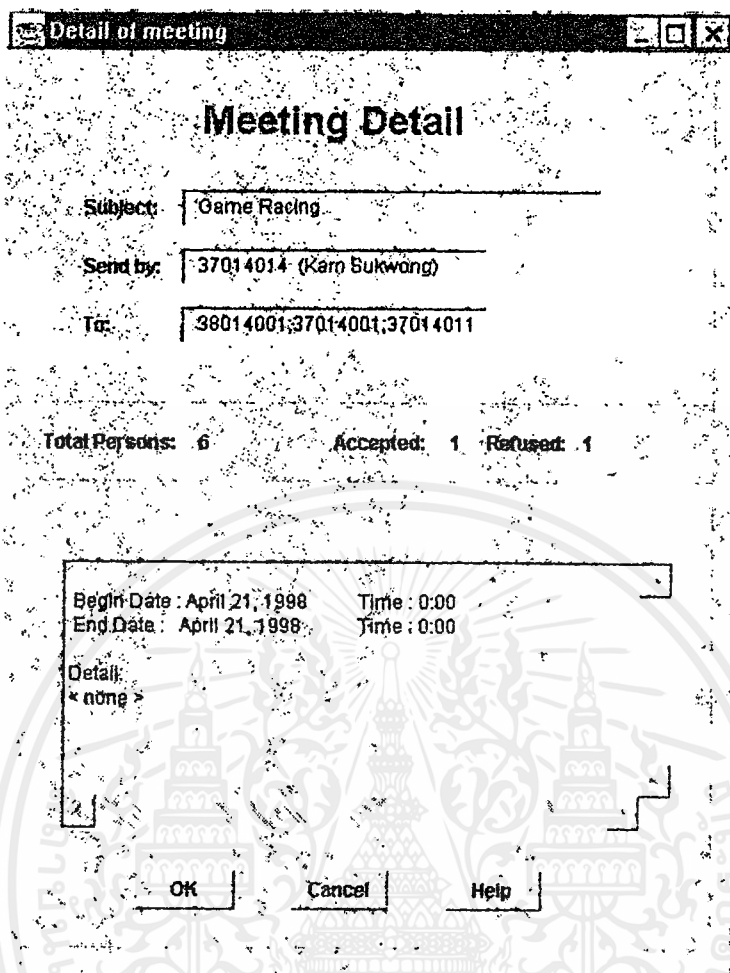
OK Help

รูปที่ 6-27 แสดงผลของการตัดสินใจของแต่ละบุคคล

รูปแบบการแสดงผลการตัดสินใจของแต่ละบุคคลที่ได้รับการนัดหมายการประชุมที่ผู้ใช้ได้ส่งออกไปจะประกอบด้วยคอลัมน์ต่างๆดังนี้

1. Receiver หมายถึง รหัสประจำตัว และชื่อนามสกุลของผู้ที่ได้รับการนัดหมายการประชุม
2. Read หมายถึง ผู้ที่ได้รับการนัดหมายการประชุมได้เข้ามาอ่านรายละเอียดของการนัดหมายการประชุมนั้นแล้วหรือยัง
3. Accept หมายถึง ผู้ที่ได้รับการนัดหมายการประชุมนั้นตอบตกลงว่าจะเข้าร่วมการประชุมหรือไม่ โดยที่ถ้าเป็นเครื่องหมายถูกสีเขียวก็แสดงว่าจะเข้าร่วม แต่ถ้าเป็น No Reply ก็แสดงว่าไม่ได้ตอบ
4. Refuse หมายถึง ผู้ที่ได้รับการนัดหมายการประชุมนั้นตอบว่าจะปฏิเสธการประชุมหรือไม่ โดยที่ถ้าเป็นเครื่องหมายถูกสีแดงก็แสดงว่าปฏิเสธ แต่ถ้าเป็น No Reply ก็แสดงว่าไม่ได้ตอบ
5. Last Login หมายถึง เวลาครั้งล่าสุดที่ผู้ที่ได้รับการนัดหมายการประชุมได้เข้ามาในระบบ ในส่วนนี้จะ เป็นประโยชน์แก่ผู้ใช้อย่างมากเพราะสามารถคาดการณ์ได้ว่า เป็นไปได้มากน้อยแค่ไหนที่ผู้ได้รับการนัดหมายการประชุมจะตัดสินใจกับการนัดหมายการประชุมที่ได้ส่งไปให้

ถ้าผู้ใช้ต้องการดูรายละเอียดการนัดหมายการประชุมก็สามารถทำได้โดยการคลิกที่ปุ่ม Get Msg ที่อยู่ด้านบนก็จะปรากฏหน้าจอใหม่ดังรูปด้านล่าง



รูปที่ 6-28 แสดงรายละเอียดการนัดหมายการประชุมที่ส่งออกไป

ส่วนที่ทำหน้าที่สร้างการนัดหมายการประชุม

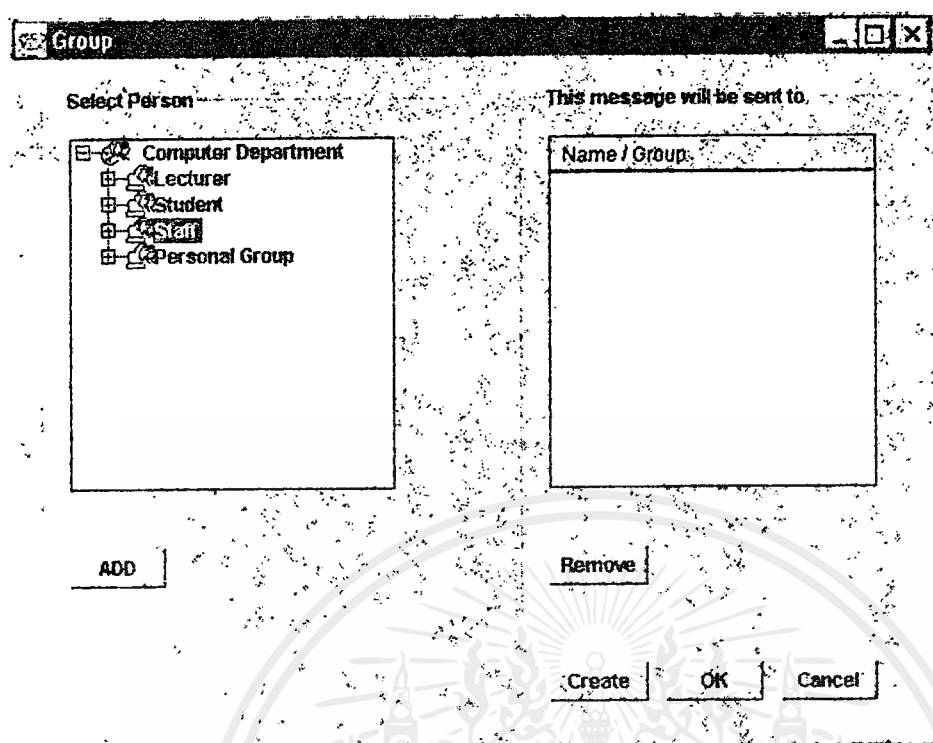
เมื่อผู้ใช้ต้องการจะทำการสร้างการนัดหมายประชุมขึ้นมา ก็สามารถทำได้ โดยการคลิกที่ปุ่ม New Msg ที่อยู่ด้านบนซึ่งก็จะทำการแสดงหน้าจอใหม่ดังรูปด้านล่าง

รูปที่ 6-29 แสดงรูปแบบในการสร้างการนัดหมายการประชุมขึ้นมาใหม่

จากรูปด้านบนจะแบ่งการทำงานออกเป็น 2 ส่วนก็คือ

- Meeting
- Summary

1. Meeting จะใช้สำหรับการส่งการนัดหมายการประชุมใหม่โดยจะประกอบด้วยส่วนที่ทำหน้าที่ในการรับข้อมูลจากผู้ซึ่งจะเป็นข้อมูลที่เกี่ยวข้องกับการนัดหมายการประชุมใหม่ที่จะถูกส่งออกไป ผู้ใช้จะต้องทำการกรอกรายละเอียดต่างๆ ให้ครบการนัดหมายการประชุมจึงจะสามารถถูกส่งออกไปได้
2. Summary จะใช้สำหรับการส่งสรุปผลการประชุม, สรุปการแบ่งงาน ไปยังผู้ที่เข้าร่วมประชุมที่ผู้ใช้ต้องการในการระบุที่อยู่ของผู้รับผู้ใช้จะทำการป้อนเข้าไปเอง หรือจะทำการเลือกจากโครงสร้างแบบต้นไม้ที่มีอยู่แล้วก็ได้ ถ้าผู้ใช้ต้องการจะเลือกจากโครงสร้างต้นไม้ก็สามารถทำได้โดยการคลิกที่ปุ่ม To ซึ่งก็จะปรากฏหน้าจอใหม่ดังรูปด้านล่าง



รูปที่ 6-30 แสดง โครงสร้างต้นไม้ที่เก็บรายชื่อ

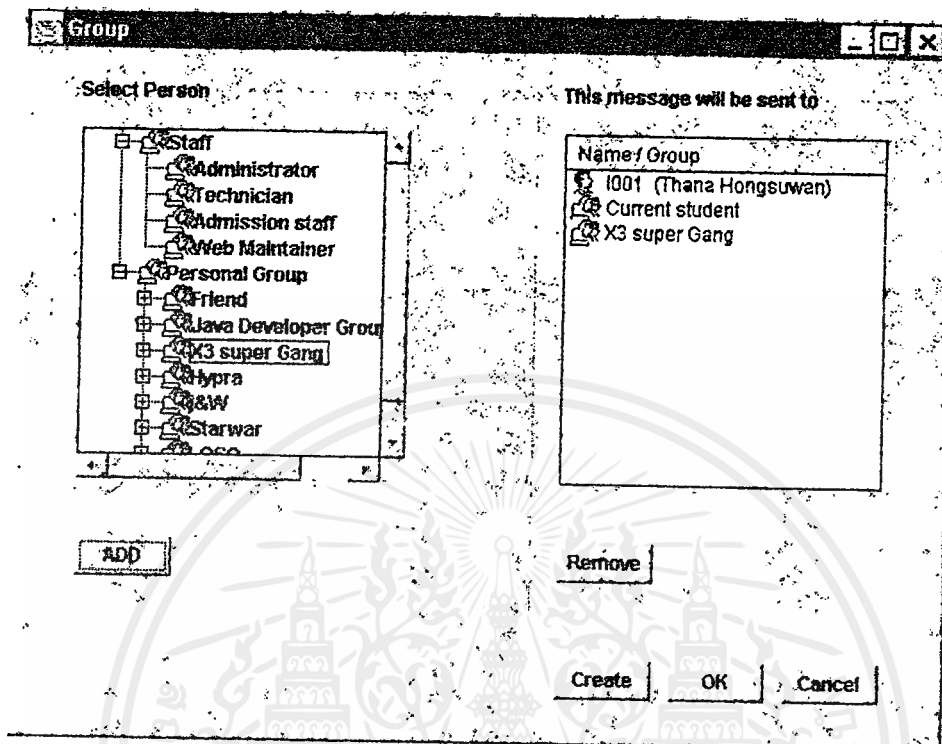
จากรูปด้านบนจะเห็น โครงสร้างต้นไม้ที่ใช้ในการเก็บชื่อ- รหัสประจำตัวของผู้ที่มีส่วนเกี่ยวข้องกับภาค วิชาคอมพิวเตอร์ซึ่งสามารถแบ่งประเภทออกได้ดังนี้

1. Lecturer คือ อาจารย์ภายในภาควิชาคอมพิวเตอร์
2. Student
 - 2.1 Current Student คือ นักศึกษาปัจจุบันของภาควิชาคอมพิวเตอร์
 - 2.2 Graduated Student คือ ศิษย์เก่าของภาควิชาคอมพิวเตอร์
3. Staff
 - 3.1 Administrator คือ เจ้าหน้าที่หรือนักศึกษาที่รับผิดชอบ-ดูแลระบบคอมพิวเตอร์ภายใน ภาควิชาคอมพิวเตอร์
 - 3.2 Technician คือ เจ้าหน้าที่หรือนักศึกษาที่รับผิดชอบ-ดูแลเกี่ยวกับอุปกรณ์ต่างๆภายใน ภาควิชาคอมพิวเตอร์
 - 3.3 Admission Staff คือ เจ้าหน้าที่ธุรการของภาควิชาคอมพิวเตอร์
 - 3.4 Web Maintainer คือ เจ้าหน้าที่หรือนักศึกษาที่รับผิดชอบ-ดูแลโครงการนี้

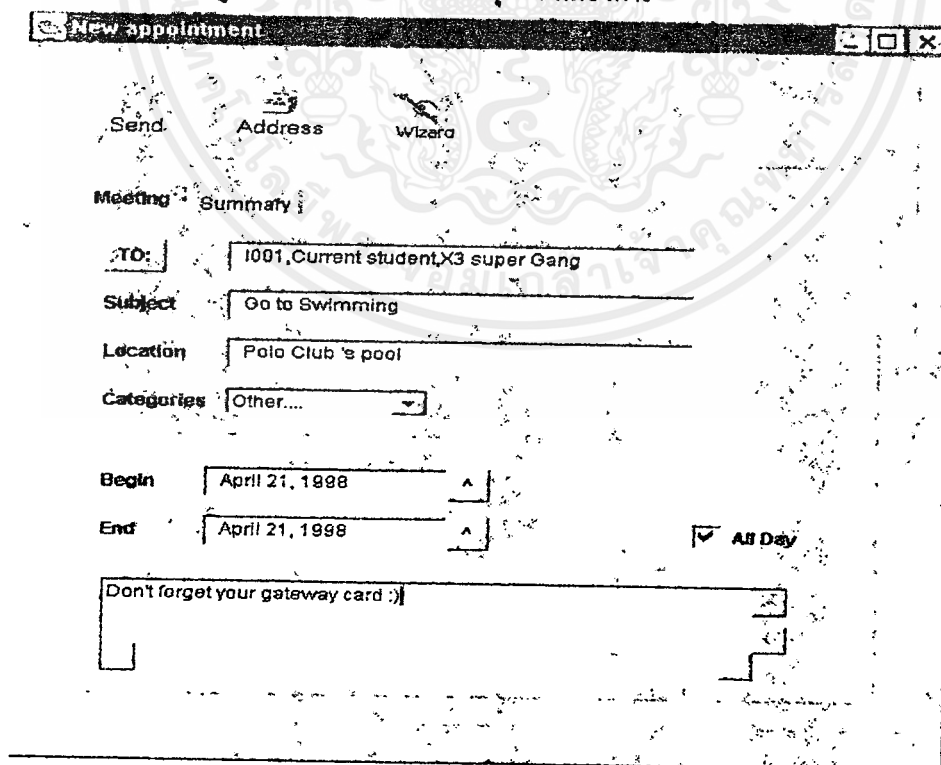
4. Personal Group คือ กลุ่มของสมาชิกที่ผู้ใช้เป็นคนกำหนดขึ้นมาเอง กลุ่มที่อยู่ภายใน Personal Group นี้ผู้ที่เป็นคนกำหนดขึ้นมาเท่านั้นที่จะดำเนินการต่างได้ เพราะบุคคลอื่นจะ ไม่สามารถเห็น กลุ่มที่สร้างขึ้นมาได้

เมื่อผู้ใช้งานต้องการเลือกสมาชิกจาก โครงสร้างต้นไม้ก็สามารถทำได้ โดยการคลิกที่ชื่อของบุคคลที่ เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการเรียนการสอนเท่านั้น เมื่ออนุญาตให้เผยแพร่เอกสารด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการจากนั้น ให้ทำการคลิกที่ปุ่ม Add ชื่อของบุคคลดังกล่าวก็จะไปปรากฏอยู่ทางกรอบด้านขวามือ เมื่อผู้ใช้เลือกบุคคลที่ต้องการครบแล้วก็ให้คลิกที่ปุ่ม OK ดังจะแสดงรูปด้านล่าง



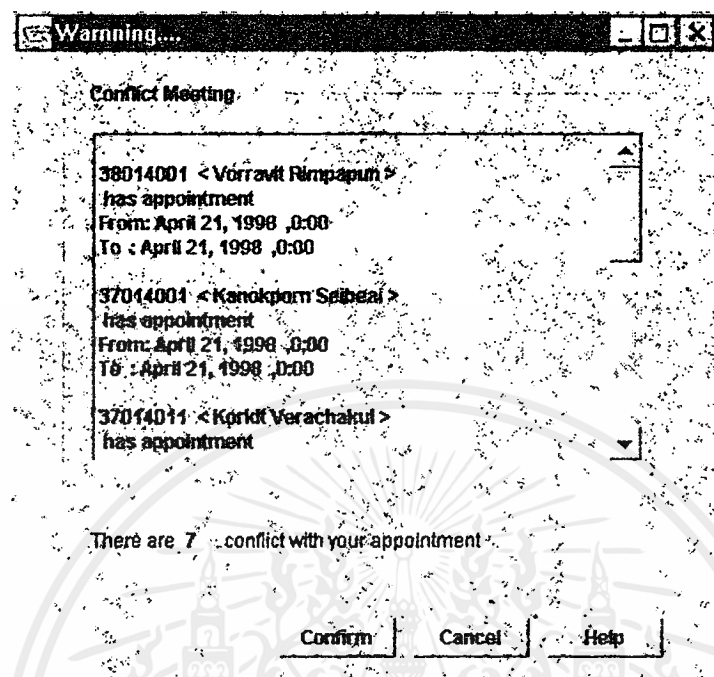
รูปที่ 6-31 แสดงการเลือกบุคคลที่ต้องการ



รูปที่ 6-32 แสดงผลจากการคลิกที่ปุ่ม OK ในรูป 6.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้กรอกข้อมูลทุกอย่างเรียบร้อยแล้วก็สามารถส่งการนัดหมายต่างๆ ได้โดยการคลิกที่ปุ่ม Send แต่ถ้าเกิดการซ้ำซ้อนในเรื่องของเวลาของผู้ที่จะเข้าร่วมการประชุมก็จะมีแจ้งเตือนดังรูปด้านล่าง

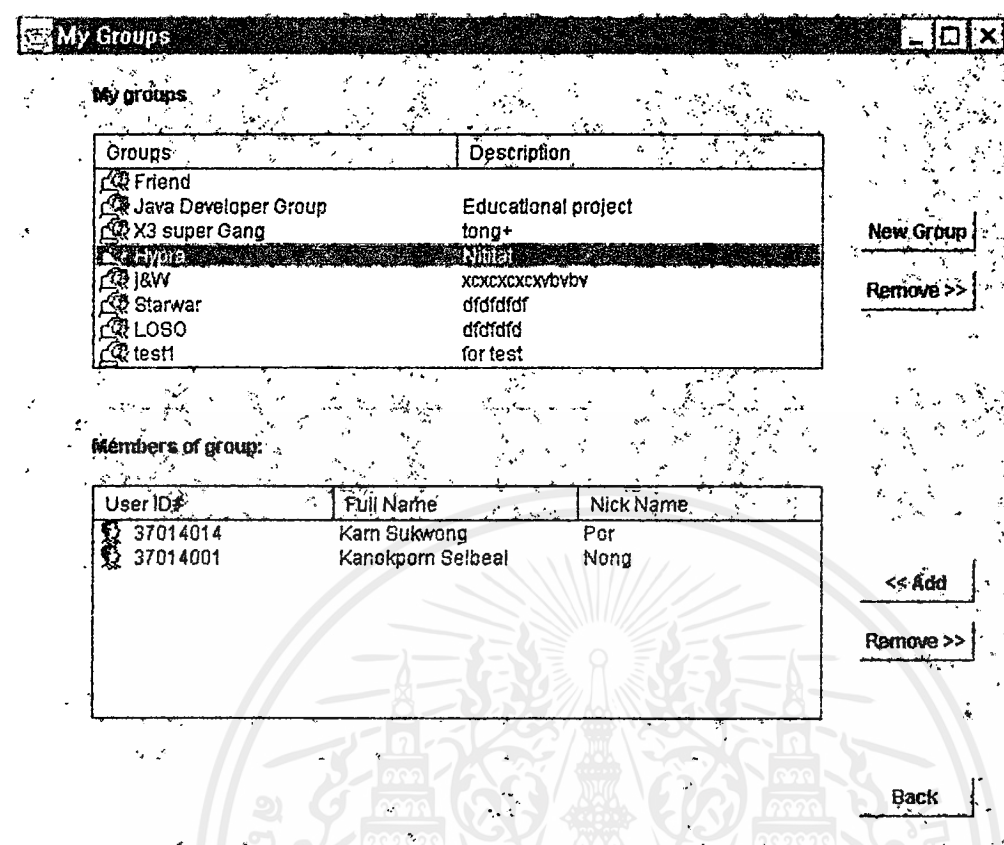


รูปที่ 6-33 แสดงหน้าจอเตือนเมื่อมีการซ้ำซ้อนในเรื่องของเวลา

จากรูปด้านบนผู้ใช้จะมีทางเลือกในการตัดสินใจอยู่ 2 ทางคือ ยืนยันการนัดหมายการประชุม กับ ยกเลิกการส่งการนัดหมายการประชุมออกไป ถ้าผู้ใช้ตัดสินใจยกเลิกการส่งการนัดหมายการประชุมก็จะกลับไปยังรูปที่ 6-31

ส่วนที่ทำหน้าที่ในการสร้างกลุ่มของผู้ใช้ขึ้นมา

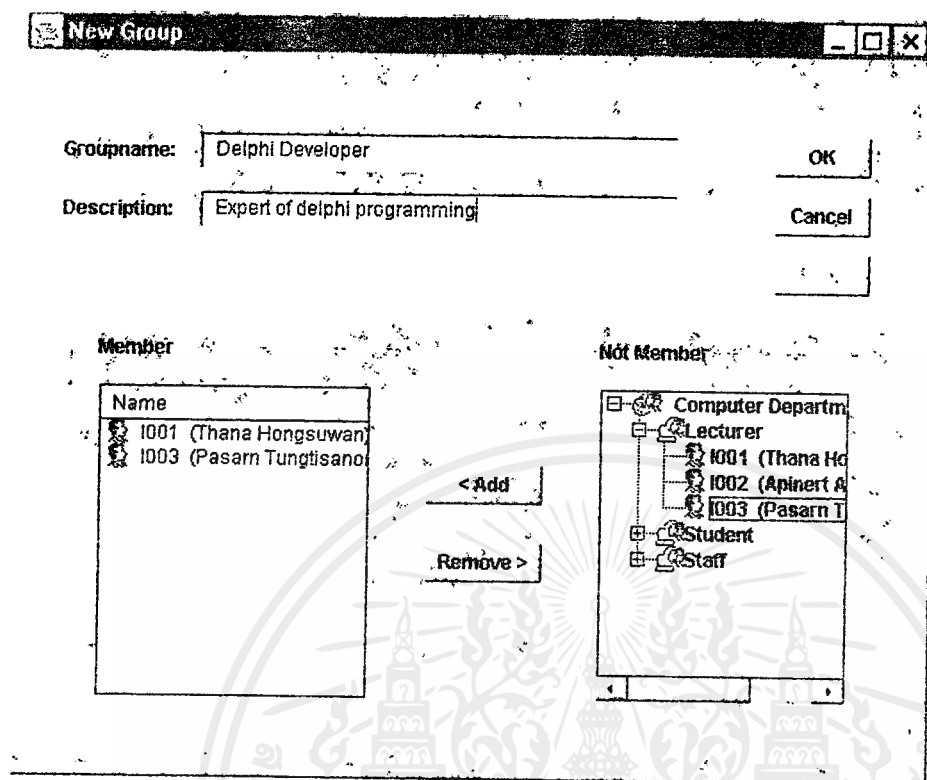
ผู้ใช้สามารถที่จะจัดตั้งกลุ่มของตนเองขึ้นมาใหม่ได้ โดยการกำหนดสมาชิกให้กับกลุ่มนั้นๆ โดยกลุ่มที่จัดตั้งขึ้นจะมองเห็นได้เฉพาะจากผู้ใช้ที่เป็นคนตั้งกลุ่มนั้นขึ้นมาเท่านั้น ผู้ใช้สามารถจัดตั้งกลุ่มใหม่ขึ้นมาได้โดยการคลิกที่ปุ่ม Create จากหน้าจอในรูปที่ 6-31 ก็จะปรากฏหน้าจอใหม่ดังรูปด้านล่าง



รูปที่ 6-34 แสดงหน้าจอกลุ่มที่ผู้ใช้ได้ตั้งไว้แล้ว

จากรูปด้านบนจะประกอบด้วย 2 ส่วน คือ ส่วนแรก (ด้านบน) จะเป็นรายชื่อของกลุ่มที่ผู้ใช้เป็นคนจัดตั้งขึ้นมา ส่วนที่สอง (ด้านล่าง) จะเป็นรายชื่อของสมาชิกในกลุ่มที่ผู้ใช้ทำการเลือก (โดยการคลิกที่ชื่อกลุ่มในส่วนแรก) ถ้าสังเกตให้ดีจะพบว่าจะมีปุ่ม Remove อยู่ 2 ปุ่ม ปุ่ม Remove ด้านบนจะทำการลบชื่อกลุ่มที่ผู้ใช้เลือกไว้และสมาชิกทั้งหมดในกลุ่มนั้นออกไป ส่วนปุ่ม Remove ที่อยู่ด้านล่างจะทำการลบสมาชิกที่ผู้ใช้เลือกออกไป (ผู้ใช้สามารถลบสมาชิกได้พร้อมกันหลายคน โดยการคลิกที่รายชื่อสมาชิกได้มากกว่า 1 คน)

เมื่อผู้ใช้พร้อมที่สร้างกลุ่มใหม่แล้ว ก็สามารถทำได้โดยการคลิกที่ปุ่ม New Group ก็จะปรากฏหน้าจอในด้านล่าง



รูปที่ 6-35 แสดงหน้าจอการจัดตั้งกลุ่มขึ้นมาใหม่

จากรูปด้านบนจะเป็นรูปแบบฟอร์มในการตั้งกลุ่มขึ้นมาใหม่ ผู้ใช้จะต้องทำการตั้งชื่อกลุ่มที่จัดตั้งขึ้นมาใหม่ โดยชื่อที่ตั้งขึ้นมาแล้วจะต้องไม่เหมือนกับชื่อกลุ่มที่ผู้ใช้เคยตั้งไว้แล้ว มิฉะนั้นแล้ว โปรแกรมจะไม่นอนุญาติให้ตั้งกลุ่มนั้นได้ หลังจากที่ผู้ใช้ทำการตั้งชื่อกลุ่มแล้วผู้ใช้ต้องทำการเลือกสมาชิกในกลุ่มโดยเลือกจาก โครงสร้างต้นไม้ที่อยู่ด้านขวามือ เมื่อใส่ข้อมูลที่ต้องการจนหมดแล้วผู้ใช้ก็จะทำการคลิกปุ่ม OK เพื่อจัดตั้งกลุ่มที่ต้องการ

6.2.4 การเปลี่ยนรหัสเวิร์ดของผู้ใช้

การเปลี่ยนรหัสเวิร์ดมีหน้าจอหลักดังนี้คือ

รูปที่ 6-36 หน้าจอในการเปลี่ยนรหัสเวิร์ด

เมื่อผู้ใช้ใส่ฟิลด์ต่างๆครบแล้วและคลิกปุ่ม OK ก็จะปรากฏหน้าจอเพื่อให้ผู้ใช้ยืนยัน Password ใหม่ดังนี้

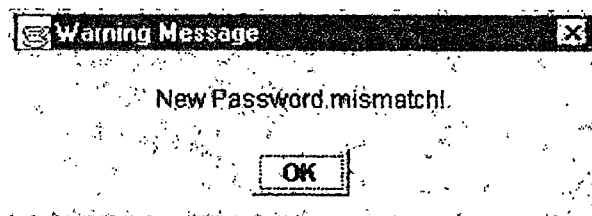
รูปที่ 6-37 หน้าจอยืนยันรหัสเวิร์ดใหม่

แต่ถ้าผู้ใช้ใส่ฟิลด์ต่างๆไม่ครบถ้วนแล้วคลิกปุ่ม OK ระบบก็จะแจ้งข้อความเตือนดังนี้

รูปที่ 6-38 หน้าจอแสดงข้อความเตือน (Warning Message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้นั้น พาสเวิร์ดใหม่ถูกต้องแล้วระบบก็จะทำการอัปเดตพาสเวิร์ดในฐานข้อมูล (Database) ให้ แต่ถ้าพาสเวิร์ดใหม่ที่ใช้นั้นไม่ถูกต้องระบบก็จะแสดงข้อความเตือนขึ้นมาดังนี้



รูปที่ 6-39 หน้าจอแสดงข้อความเตือน (Warning Message)



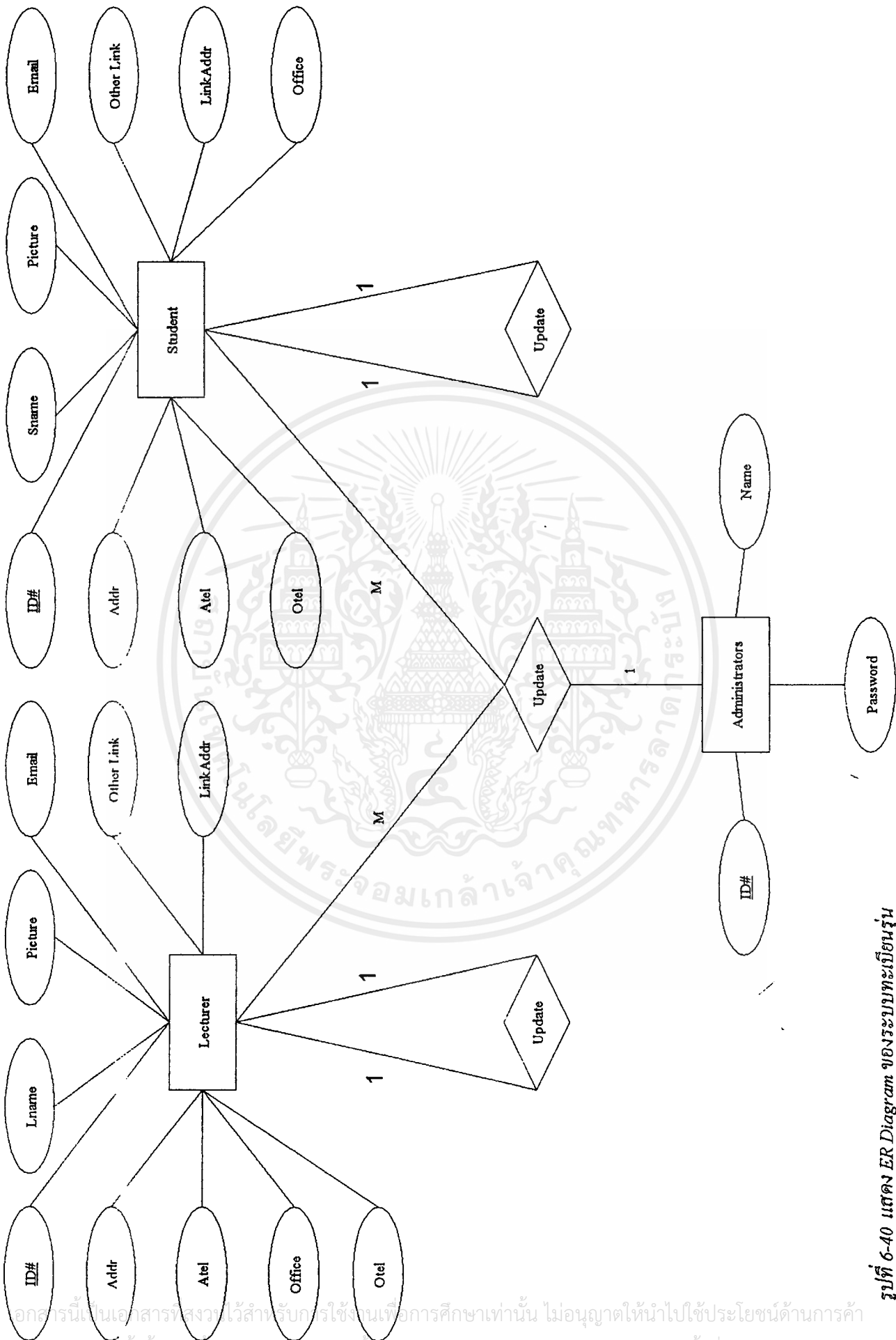
6.3 รูปแบบฐานข้อมูล (Database)

- ระบบทะเบียนรุ่น

แสดงเป็นอีอาร์ไดอะแกรมได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-40 แสดง ER Diagram ของระบบทะเบียนรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถแปลเป็นตารางที่ใช้ในการจัดเก็บข้อมูลได้ดังนี้

1.1 tblLecturer

เก็บข้อมูลต่างๆของอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ โดยมีฟิลด์ต่างๆดังนี้



Id#	Lname	Addr	Atel	RealAddr	Rtel	Office	Otel	Other_link	Fax	Email
-----	-------	------	------	----------	------	--------	------	------------	-----	-------

Picture	Password
---------	----------

Id#	เก็บรหัสของอาจารย์โดยเป็นค่าที่ไม่ซ้ำกัน (เป็น Primary Key)
Lname	เก็บชื่อและนามสกุลของอาจารย์
Addr	เก็บที่อยู่ปัจจุบันของอาจารย์
Atel	เก็บเบอร์โทรศัพท์ของที่อยู่ปัจจุบัน
RcalAddr	เก็บที่อยู่ตามภูมิตำแน่งทางทหาร
Rtel	เก็บเบอร์โทรศัพท์ของที่อยู่ตามภูมิตำแน่งทางทหาร
Office	เก็บที่ทำงานของอาจารย์
Otel	เก็บเบอร์โทรศัพท์ที่ทำงาน
Other_link	เก็บเบอร์อุปกรณ์ที่ใช้ในการติดต่ออื่นๆอีก เช่น เพจเจอร์ มือถือ เป็นต้น
Fax	เก็บเบอร์แฟกซ์ของที่ทำงาน
Email	เก็บอีเมลแอดเดรสของอาจารย์
Picture	เก็บชื่อไฟล์รูปภาพของอาจารย์
Password	เก็บพาสเวิร์ดของอาจารย์

1.2 tblStd

เก็บข้อมูลต่างๆของนักศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ โดยมีฟิลด์ต่างๆดังนี้



Id#	Sname	Addr	Atel	RealAddr	Rtel	Office	Otel	Other_link	Fax	Email
-----	-------	------	------	----------	------	--------	------	------------	-----	-------

Picture	Year	Password
---------	------	----------

Id#	เก็บรหัสของนักศึกษาโดยเป็นค่าที่ไม่ซ้ำกัน (Primary Key)
Sname	เก็บชื่อและนามสกุลของนักศึกษา
Addr	เก็บที่อยู่ปัจจุบันของนักศึกษา
Atel	เก็บเบอร์โทรศัพท์ของที่อยู่ปัจจุบัน
RealAddr	เก็บที่อยู่ตามภูมิตำแน่งทางทหาร
Rtel	เก็บเบอร์โทรศัพท์ของที่อยู่ตามภูมิตำแน่งทางทหาร

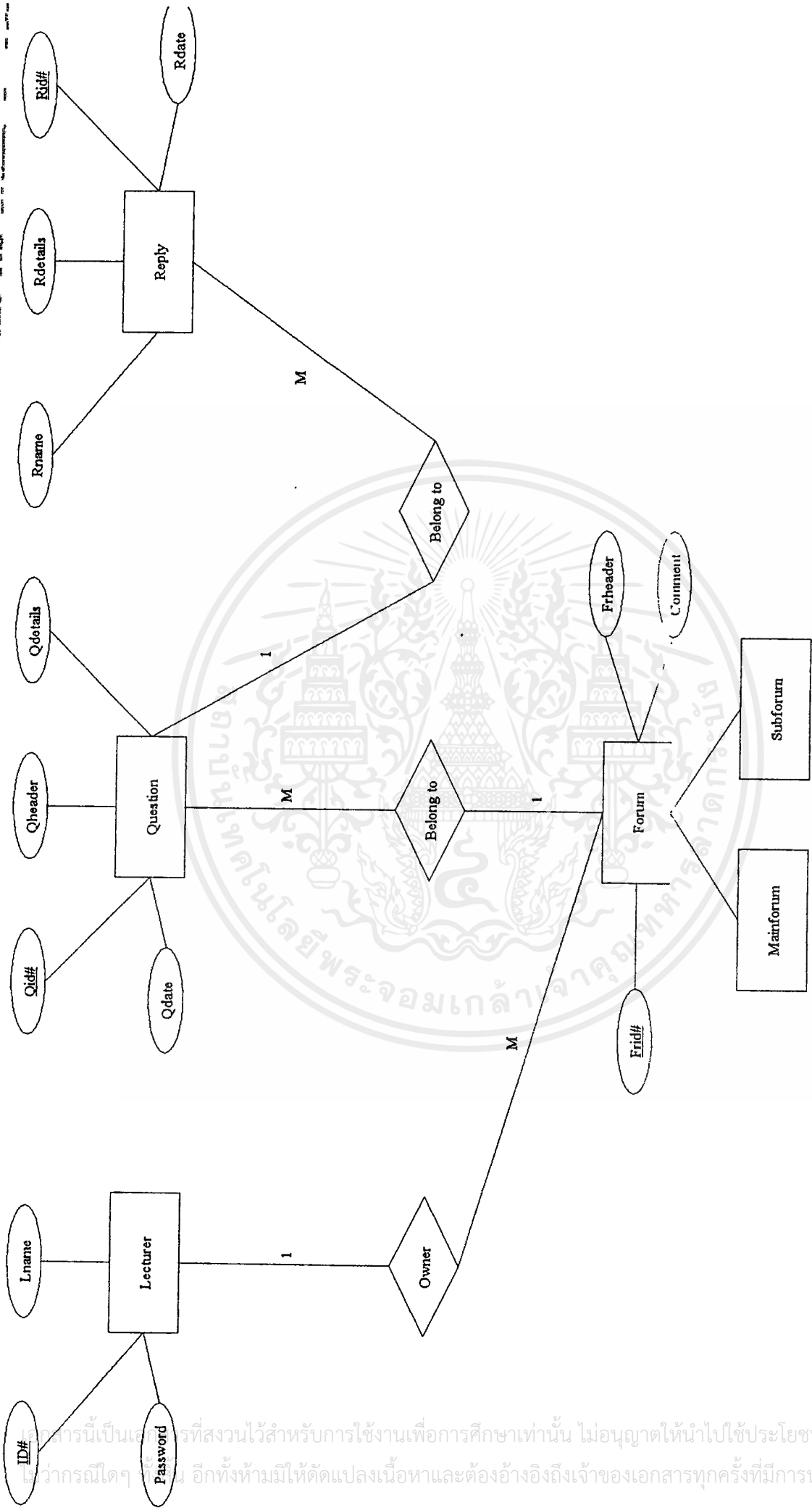
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Office	เก็บที่ทำงานของนักศึกษา
Otel	เก็บเบอร์โทรศัพท์ที่ทำงาน
Other_link	เก็บเบอร์อุปกรณ์ที่ใช้ในการติดต่ออื่นๆอีก เช่น เพจเจอร์ มือถือ เป็นต้น
Fax	เก็บเบอร์แฟกซ์ของที่ทำงาน
Email	เก็บอีเมลแอดเดรสของนักศึกษา
Picture	เก็บชื่อไฟล์รูปภาพของนักศึกษา
Year	เก็บปีที่เข้าศึกษา
Password	เก็บพาสเวิร์ดของนักศึกษา

- ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต
แสดงเป็นอีอาร์ไออะแกรมดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-41 แสดง ER Diagram ของระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถแมปเป็นตารางที่ใช้ในการจัดเก็บข้อมูลได้ดังนี้

2.1 tblOwner

เป็นตารางที่เกี่ยวกับความเป็นเจ้าของ Forum เริ่มต้น โดยมีฟิลด์ต่างๆดังนี้

←→	
Id#	Frid#

Id# เก็บรหัสของอาจารย์ผู้เป็นเจ้าของหัวข้อคำถามหลัก(Primary Key)

Frid# เก็บรหัสของหัวข้อคำถามหลัก (Primary Key)

2.2 tblForum

เก็บข้อมูลต่างๆของหัวข้อคำถาม โดยมีฟิลด์ต่างๆดังนี้

←→			
Frid#	Frheader	Comment	Id#

Frid# เก็บรหัสของหัวข้อคำถาม (Primary Key)

Frheader เก็บหัวข้อคำถาม

Comment เก็บรายละเอียดเพิ่มเติม

Id# เก็บรหัสของอาจารย์ผู้เป็นเจ้าของหัวข้อคำถาม

2.3 tblSubForum

เป็นตารางแสดงความสัมพันธ์ระหว่างหัวข้อคำถาม กับหัวข้อคำถามย่อย

โดยมีฟิลด์ต่างๆดังนี้

←→	
Frid#	Subid#

Frid# เก็บรหัสของหัวข้อคำถามที่เป็นเจ้าของหัวข้อคำถามย่อย(Primary Key)

Subid# เก็บรหัสของหัวข้อคำถามย่อย (Primary Key)

2.4 tblQuestion

เก็บข้อมูลต่างๆเกี่ยวกับคำถาม โดยมีฟิลด์ต่างๆดังนี้

←→					
Qid#	Header	Qdetails	Qdate	Frid#	Id#

Qid# เก็บรหัสของคำถาม (Primary Key)

Header เก็บหัวเรื่องของคำถาม

Qdetails เก็บรายละเอียดของคำถาม

Qdate เก็บวันที่ตั้งคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Frid# เก็บรหัสของหัวข้อความ

Id# เก็บรหัสของอาจารย์ผู้เป็นเจ้าของคำถาม

2.5 tblReply

เก็บข้อมูลการตอบคำถาม โดยมีฟิลด์ต่างๆดังนี้

Qid#	Rid#	Rname	Rdetails	Rdate	Raddr
------	------	-------	----------	-------	-------

Qid# เก็บรหัสของคำถาม (Primary Key)

Rid# เก็บรหัสของคำตอบ (Primary Key)

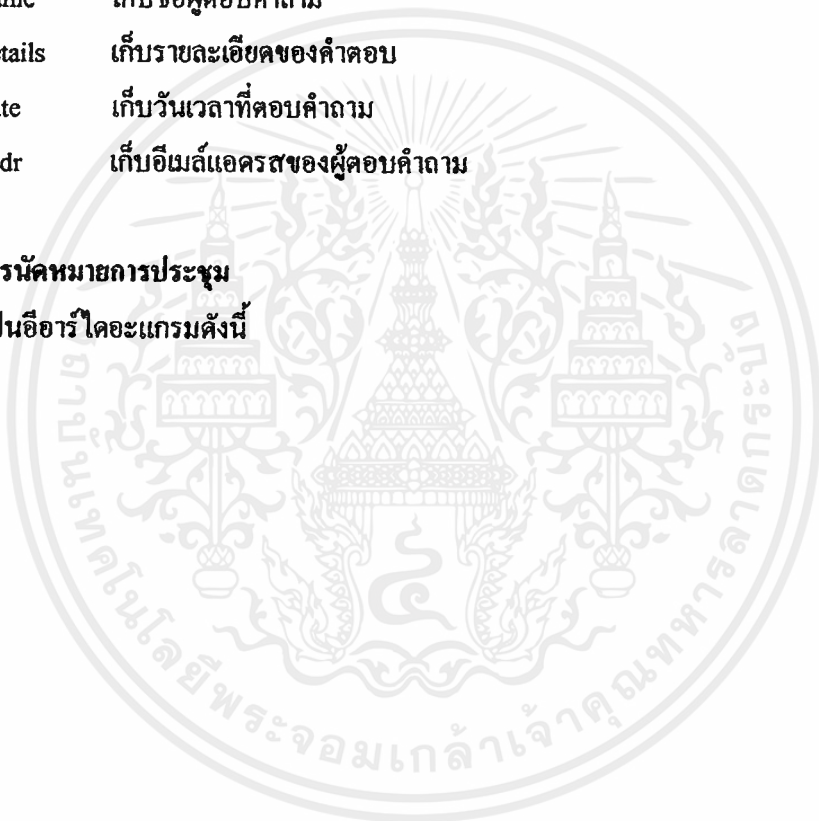
Rname เก็บชื่อผู้ตอบคำถาม

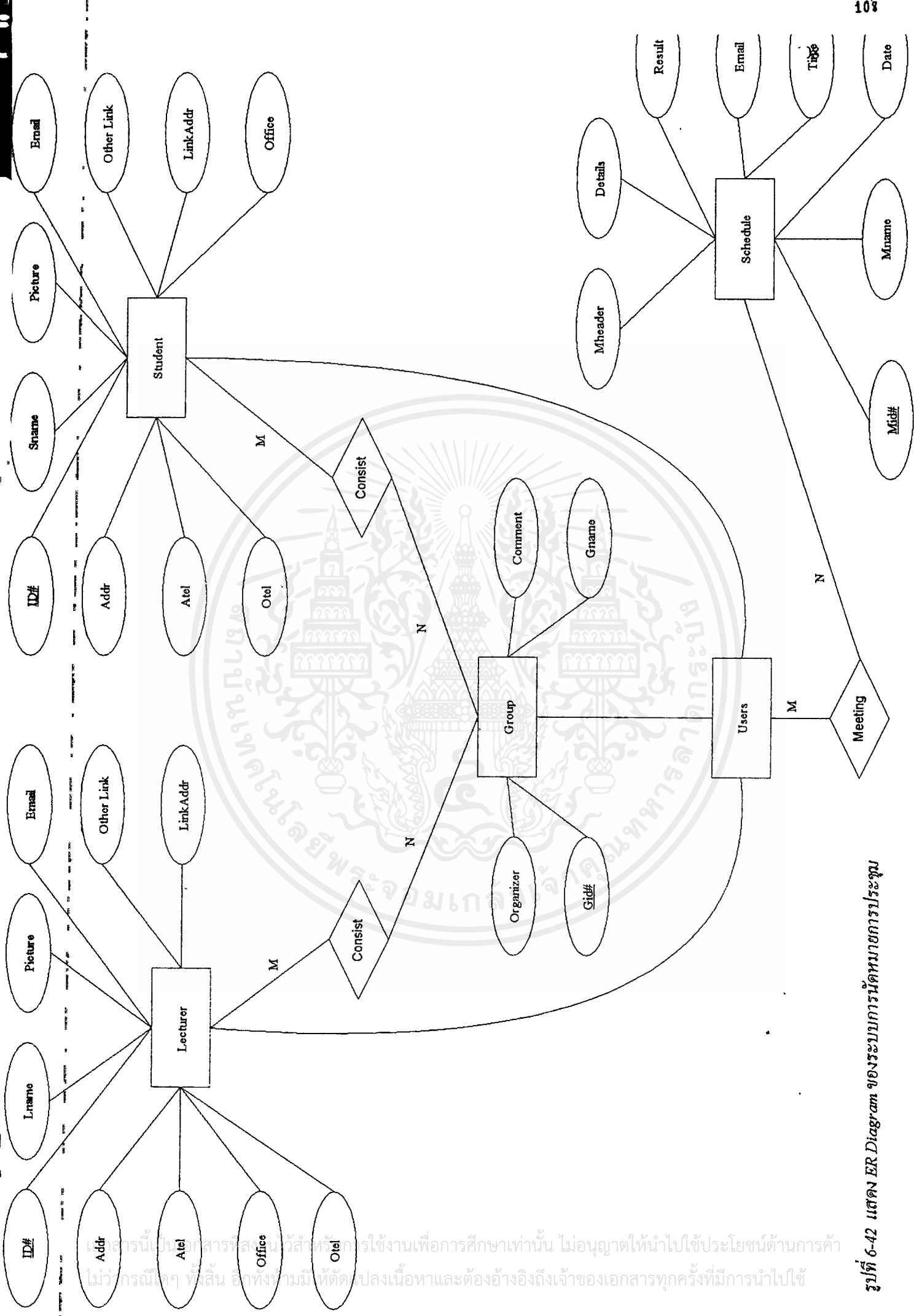
Rdetails เก็บรายละเอียดของคำตอบ

Rdate เก็บวันที่เวลาที่ตอบคำถาม

Raddr เก็บอีเมลแอดเรสของผู้ตอบคำถาม

- ระบบการจัดการนัดหมายการประชุม
แสดงเป็นอีอาร์ไดอะแกรมดังนี้





รูปที่ 6-42 แสดง ER Diagram ของระบบการนัดหมายการประชุม

โดยสามารถแมปเป็นตารางที่ใช้ในการจัดเก็บข้อมูลได้ดังนี้

3.1 tblSchedule

เป็นตารางที่เก็บข้อมูลเกี่ยวกับรายละเอียดของการนัดประชุม โดยมีฟิลด์ต่างๆดังนี้

Msubject	Mname	date	time	location	begind	enddd	begint	enddt	allday	detail
result	total									

Msubject	เก็บชื่อหัวข้อการประชุม
Mname	เก็บรหัสของผู้ที่ทำการนัดประชุม
date	เก็บวันที่ทำการกำหนดการประชุม
time	เก็บเวลาที่ทำการกำหนดการประชุม
location	เก็บสถานที่ที่จะทำการประชุม
begind	เก็บวันเริ่มต้นที่จะทำการประชุม
enddd	เก็บวันสุดท้ายที่จะทำการประชุม
begint	เก็บเวลาเริ่มต้นที่จะทำการประชุม
enddt	เก็บเวลาสุดท้ายที่จะทำการประชุม
allday	เก็บสถานะที่เป็นตัวบอกว่าเป็นการประชุมทั้งวัน
detail	เก็บรายละเอียดของหัวเรื่องที่จะทำการประชุม
result	เก็บสถานะที่บอกว่าจะยืนยันการประชุมหรือไม่
total	เก็บจำนวนผู้ที่จะเข้าประชุมทั้งหมด

3.2 tblMeeting

เป็นตารางที่เก็บข้อมูลเกี่ยวกับรายชื่อของสมาชิกผู้เข้าร่วมการประชุม โดยมีฟิลด์ต่างๆดังนี้

Mname	Date	Time	Dname	Dresult	Dmark	Dflag
-------	------	------	-------	---------	-------	-------

Mname	เก็บรหัสของผู้ที่ทำการนัดประชุม
Date	เก็บวันที่ทำการกำหนดการประชุม
Time	เก็บเวลาที่ทำการกำหนดการประชุม
Dname	เก็บรหัสของสมาชิกผู้เข้าร่วมการประชุม
Dresult	เก็บสถานะว่าจะยืนยันการประชุมหรือไม่
Dmark	เก็บสถานะว่าทำการมาร์คไว้หรือไม่
Dflag	เก็บสถานะว่าทำการเซตค่าแฟล็กไว้หรือไม่

3.3 tblGroup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตารางที่เก็บข้อมูลเกี่ยวกับรายชื่อกลุ่มทั้งหมด โดยมีฟิลด์ต่างๆดังนี้



Gid	Gname	idowner	comment
-----	-------	---------	---------

Gid เก็บรหัสของกลุ่ม

Gname เก็บชื่อของกลุ่ม

idowner เก็บรหัสของเจ้าของกลุ่ม

comment เก็บรายละเอียดเพิ่มเติมของกลุ่ม

3.4 tblConsist

เป็นตารางที่เก็บข้อมูลเกี่ยวกับรายชื่อสมาชิกในกลุ่ม โดยมีฟิลด์ต่างๆดังนี้



Gid	id
-----	----

Gid เก็บรหัสของกลุ่ม

Id เก็บรหัสของสมาชิกในกลุ่ม

บทที่ 7

ผลการทดลอง

7.1 การทดสอบการทำงานในส่วนของระบบการจัดการการนัดหมายการประชุม

1. เริ่มจากการทดสอบว่าผู้ใช้สามารถมองเห็นการนัดหมายประชุมที่อยู่ในกล่องเก็บข้อมูลขาเข้าและขาออกหรือไม่ซึ่งการแสดงผลจะปรากฏในมัลติลิสต์ซึ่งเป็นคอมโพเนนต์ที่กำหนดไว้ โดยข้อมูลนั้นเป็นข้อมูลที่กำหนดขึ้นชั่วคราวในฐานข้อมูล
2. ทดสอบว่าผู้ใช้สามารถที่จะดูรายชื่อผู้อื่นที่อยู่ในฐานข้อมูลหรือไม่ซึ่งแสดงอยู่ในรูปแบบโครงสร้างต้นไม้ ในส่วนนี้เป็นส่วนที่พบปัญหาเพราะได้มีการเขียนรีเคอซีฟ อัลกอริทึมซึ่งใช้เวลาในการทดสอบนานพอสมควรกว่าการทำงานจะถูกต้อง
3. ทดสอบว่าผู้ใช้สามารถส่งข้อมูลที่ตนต้องการซึ่งก็คือ หัวข้อการประชุม และรายละเอียดของการประชุม ว่าสามารถส่งไปยังผู้ใช้ที่ตนเลือกมาจากโครงสร้างต้นไม้ทั้งหมดได้หรือไม่ ในส่วนนี้เป็นอีกส่วนที่พบปัญหาเพราะได้มีการเขียนรีเคอซีฟ อัลกอริทึมซึ่งใช้เวลาในการทดสอบนานพอสมควรกว่าการทำงานจะถูกต้อง
4. ทดสอบถึงการชนกันของเวลาที่ผู้ที่จะได้รับการประชุมมีการนัดการประชุมกับผู้อื่นอยู่แล้วกับเวลาที่ผู้นัดประชุมต้องการ โดยทดสอบว่าถ้าเกิดการชนกันแล้วจะแจ้งให้ผู้ใช้(ผู้นัดหมาย)ทราบหรือไม่
5. ทดสอบการจัดตั้งกลุ่มของสมาชิกที่ผู้ใช้เป็นคนกำหนดขึ้นมาเองว่าสามารถจัดตั้งได้หรือไม่
6. ทดสอบว่าถ้าเปิดโดยใช้เว็บเบราว์เซอร์ว่าสามารถทำงานได้เหมือนกับเวลาที่ทดสอบใน Visual Café หรือไม่

7.2 การทดสอบการทำงานในส่วนของระบบทะเบียนรุ่น

1. เริ่มต้นทำการทดสอบคอมโพเนนต์ต่างๆของเครื่องมือ Visual Café โดยส่วนใหญ่จะเน้นไปทางด้านคอมโพเนนต์ที่จะต้องใช้ในการระบบ
2. ทำการทดสอบระบบกับการติดต่อผ่านฐานข้อมูลที่ได้ออกแบบไว้ว่าสามารถทำการติดต่อได้หรือไม่ อย่างไร
3. ทำการพัฒนาและทดสอบ โดยการสมมุติตัวเองเป็นผู้ใช้ในรูปแบบต่างๆแล้วทำการหาข้อผิดพลาดของโปรแกรม ซึ่งประกอบไปด้วยการทดสอบดังต่อไปนี้

3.1 ทดสอบการเลือกขอบเขตที่สนใจ ว่าข้อมูลในลิสต์เปลี่ยนแปลงตามถูกต้องหรือไม่

- 3.2 ทดสอบการเลือกให้แสดงชื่อเฉพาะรุ่นว่าข้อมูลที่ได้ตรงตามรุ่นนั้นๆจริงหรือไม่
- 3.3 ทดสอบการเข้าดูรายละเอียดโดยการดับเบิลคลิกที่ชื่อ ในลิสต์
- 3.4 ทดสอบการเข้าดูรายละเอียดโดยการคลิกที่ปุ่ม Details
- 3.5 ทดสอบการร้องขอการแก้ไขข้อมูลแล้วดูว่าการตรวจสอบเรื่องความปลอดภัยถูกต้องหรือไม่ อย่างไร
- 3.6 ทดสอบการแก้ไขปรับปรุงข้อมูลลงฐานข้อมูลว่าเป็นไปตามที่ความต้องการหรือไม่
- 3.7 ทดสอบระบบโดยรวมอีกครั้ง โดยการทดลองใช้งานจริง

7.3 การทดสอบการทำงานในส่วนจากระบบการถามคำถามของอาจารย์ผ่านระบบอินเทอร์เน็ต

1. เนื่องจากระบบการถามคำถามของอาจารย์ผ่านระบบอินเทอร์เน็ตนี้ทางผู้จัดทำได้ทำการออกแบบและพัฒนาต่อจากระบบทะเบียนรุ่นดังนั้นการทดสอบเกี่ยวกับคอมพิวเตอร์ บางอย่างของ Visual Café จึงไม่จำเป็นต่อทำ เพราะได้ทำการทดสอบมาแล้ว ดังนั้นจึงเริ่มต้นด้วยการทดสอบระบบกับการติดต่อผ่านฐานข้อมูลที่ได้ออกแบบไว้ว่าสามารถทำการติดต่อได้หรือไม่ อย่างไร
2. ทำการพัฒนาระบบและทดสอบโดยการสมมุติตัวเองเป็นผู้ใช้ในรูปแบบต่างๆแล้วทำการหาข้อผิดพลาดของโปรแกรม ซึ่งประกอบไปด้วยการทดสอบดังต่อไปนี้
 - 2.1 ทดสอบการสร้างต้นไม้ ของหน้าจอหลักว่าได้ตามที่ความต้องการหรือไม่ อย่างไร ซึ่งในการพัฒนาช่วงนี้ทำได้ค่อนข้างยากมากเพราะลักษณะการสร้างต้นไม้จะมีลักษณะเป็นรีเคอร์ซีฟด้วย
 - 2.2 ทดสอบการดูรายละเอียดในกรณีต่างๆดังนี้
 - 2.2.1 ทดสอบการดูรายละเอียดของคำถาม
 - 2.2.2 ทดสอบการดูรายละเอียดของหัวข้อคำถาม
 - 2.3 ทดสอบการเพิ่มคำตอบให้กับคำถาม โดยผ่านทางกรดูรายละเอียดของคำถาม
 - 2.4 ทดสอบการเพิ่มในกรณีต่างๆดังนี้
 - 2.4.1 ทดสอบการเพิ่มหัวข้อคำถาม
 - 2.4.2 ทดสอบการเพิ่มคำถาม
 - 2.4.3 ทดสอบการเพิ่มคำตอบ
 - 2.5 ทดสอบการลบหรือยกเลิกในกรณีต่างๆดังนี้
 - 2.5.1 ทดสอบการลบหัวข้อคำถาม
 - 2.5.2 ทดสอบการลบคำถาม
 - 2.5.3 ทดสอบการลบคำตอบ
 - 2.6 ทดสอบการแก้ไขในกรณีต่างๆดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 ทดสอบการแก้ไขหัวข้อคำถาม

2.6.2 ทดสอบการแก้ไขคำถาม

2.7 ทดสอบการทำงานโดยรวมของระบบอีกครั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

สรุปผลการพัฒนาโครงการ

8.1 ผลการพัฒนาโครงการ

ในการทำการพัฒนาโครงการด้วยภาษาจาวาสามารถสรุปขั้นตอนการพัฒนาได้ดังนี้

8.1.1 การศึกษารวบรวมข้อมูล

การศึกษาเก็บรวบรวมข้อมูลเพื่อใช้เป็นแนวทางในการออกแบบระบบฐานข้อมูล และการพัฒนาโปรแกรม โดยใช้วิธีค้นคว้าจากหนังสือต่างๆและค้นคว้าทางระบบอินเทอร์เน็ต โดยมีเป้าหมายเพื่อเพิ่มทักษะในการออกแบบฐานข้อมูล ประมวลผลข้อมูล และเพื่อค้นหาวิธีที่จะใช้ในการออกแบบและสร้างระบบ ซึ่งครอบคลุม ระบบทะเบียนรุ่น ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต และระบบการจัดการนัดหมายการประชุม

8.1.2 การวิเคราะห์และออกแบบระบบงาน

การวิเคราะห์และออกแบบระบบตลอดจนลำดับขั้นตอนในการปฏิบัติงานเพื่อเป็นต้นแบบในการพัฒนาโปรแกรมโดยใช้วิธีการที่เรียกว่า คาค้าโพลว์ไคอะแกรม โดยในการออกแบบระบบงานจะทำการแยกออกแบบเป็นระบบๆไป

8.1.3 การวิเคราะห์หาความสัมพันธ์ของข้อมูลและออกแบบฐานข้อมูล

การวิเคราะห์ถึงความสัมพันธ์ของข้อมูล ได้มาจากการเริ่มสอบถามถึงการทำงานระหว่างหน่วยงานต่างๆภายในภาควิชา ก่อน ,สอบถามถึงระบบการทำงานที่มีอยู่แล้ว,สอบถามถึงความคิดเห็นและความต้องการจากนักศึกษาและอาจารย์ ตลอดจนผู้ที่มีส่วนเกี่ยวข้องกับภาควิชาคอมพิวเตอร์ จากนั้นจึงนำเอาข้อมูลทั้งหมดมาออกแบบความสัมพันธ์ระหว่างกันโดยใช้วิธี ER Case tool ซึ่งจะ ได้รูปแบบของตารางออกมา จากนั้นจึงเริ่มคิดตั้งฐานข้อมูลโดยใช้ MS ACCESS

8.1.4 การพัฒนาโปรแกรมคอมพิวเตอร์

การพัฒนาโปรแกรมจะแบ่งการพัฒนาออกเป็นส่วนๆประกอบด้วย ระบบทะเบียนรุ่น, ระบบการตั้งคำถามของอาจารย์ และ ระบบการนัดหมายการประชุมล่วงหน้า เมื่อพัฒนาแต่ละส่วนเสร็จจะนำทุกส่วนมารวมกัน

8.2 การติดตั้งใช้งาน

เริ่มจากการติดตั้งให้ Window NT สามารถทำงานเป็นเว็บเซิร์ฟเวอร์ได้จากนั้นจึงออกแบบเว็บเพจหลักที่สามารถเชื่อมโยงไปยังโปรแกรมที่พัฒนาไว้ ดังนั้นผู้ใช้จะสามารถเข้ามาใช้โปรแกรมที่พัฒนาขึ้นมาได้ โดยการใช้เว็บเบราว์เซอร์ เช่น Netscape , Internet Explorer ต่อเข้ามายังเว็บเซิร์ฟเวอร์ที่ตั้งไว้

8.3 คุณสมบัติและประโยชน์ของโปรแกรม

8.3.1 โปรแกรมที่ออกแบบมาใช้ง่าย สะดวก เพื่อให้ผู้ใช้ที่ไม่มีความรู้ด้านคอมพิวเตอร์สามารถเรียกใช้ได้ง่าย

8.3.2 การปรับเปลี่ยนโปรแกรมเบื้องต้นสามารถทำได้ง่าย สามารถทำได้โดยการไปปรับเปลี่ยนข้อมูลในฐานข้อมูลเท่านั้นผู้ที่เข้ามาบำรุงรักษาไม่จำเป็นต้องมีความรู้ในการเขียน โปรแกรม

8.3.3 สามารถช่วยอำนวยความสะดวกในการติดต่อสื่อสารภายในภาควิชาคอมพิวเตอร์ ตลอดจนศิษย์เก่าที่จบไปแล้วสามารถที่จะติดต่อกัน ได้ง่ายยิ่งขึ้น โดยผ่านระบบการนัดหมายการประชุมล่วงหน้า

8.3.4 สามารถช่วยอำนวยความสะดวกในการค้นหาข้อมูลของนักศึกษาปัจจุบัน, อาจารย์, นักศึกษาที่จบการศึกษาไปแล้ว โดยที่การแก้ไขข้อมูลส่วนตัวก็สามารถทำได้ง่ายเช่นกัน โดยผ่านระบบทำเนียบรุ่น

8.3.5 เป็นแหล่งในการเผยแพร่ความรู้ซึ่งอยู่ในส่วนของระบบการตั้งคำถามของอาจารย์ ซึ่งจะมีการแลกเปลี่ยนความรู้และความคิดเห็นในเรื่องต่างๆ

8.4 ข้อเสนอแนะ

8.4.1 เนื่องจาก โปรแกรมที่พัฒนาขึ้นเป็นภาษาชั้นสูง และทำงานบนฐานข้อมูล ระบบนี้จึงควรนำไปใช้งานบนเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงด้วย

8.4.2 สามารถที่จะเผยแพร่ไปใช้งานในหน่วยงานอื่นได้

8.4.3 การประยุกต์ใช้งานจะมีประโยชน์อย่างยิ่งกับทุกๆคณะ ถ้าสามารถติดตั้งเข้ากับระบบคอมพิวเตอร์ที่เชื่อมต่อกับระบบเครือข่ายหลักของสถาบันฯ ผ่านโปร โทคอล TCP/IP แล้วจะทำให้สามารถเรียกใช้จากจุดใดๆก็ได้

ภาคผนวก ก.

คู่มือการ โปรแกรม (Program Manual)

การ โปรแกรมจะแบ่งส่วนหลักๆออกเป็น 3 ส่วนคือ

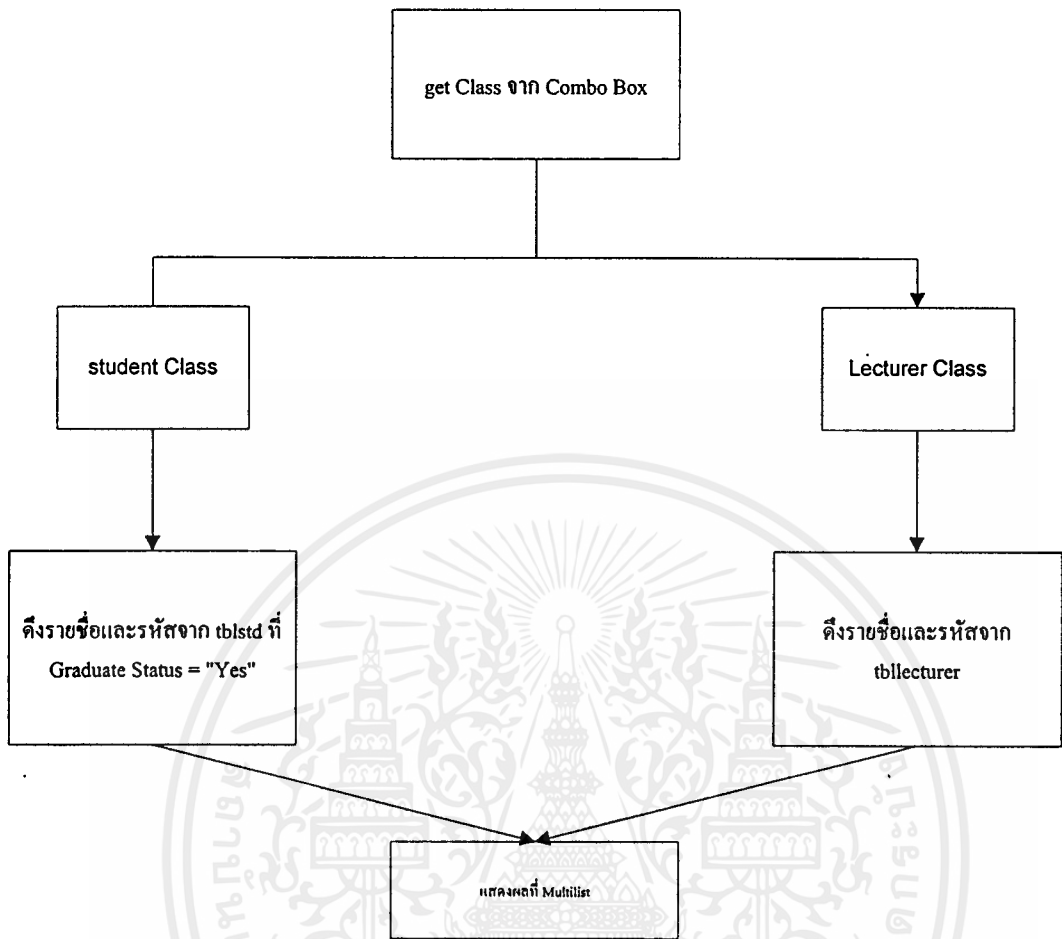
1. โปรแกรมระบบทะเบียนรุ่น (Name.vpj)
2. โปรแกรมระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต (Dc3.vpj)
3. โปรแกรมระบบการนัดหมายการประชุม (real.vpj)
4. โปรแกรมการเปลี่ยนพาสเวิร์ดของระบบ (setPass.vpj)

1. โปรแกรมระบบทะเบียนรุ่น (Computer Member Information)

เก็บอยู่ในไฟล์โปรเจกของ Visual Café 2 ชื่อ Name.vpj และมีโมดูลต่างๆดังนี้

- 1.1 โมดูลสำหรับการแสดงรายชื่อตอนเริ่มต้น
มีลักษณะการทำงานดังนี้





รูปที่ ก-1 Flowchart Diagram การแสดงรายชื่อ

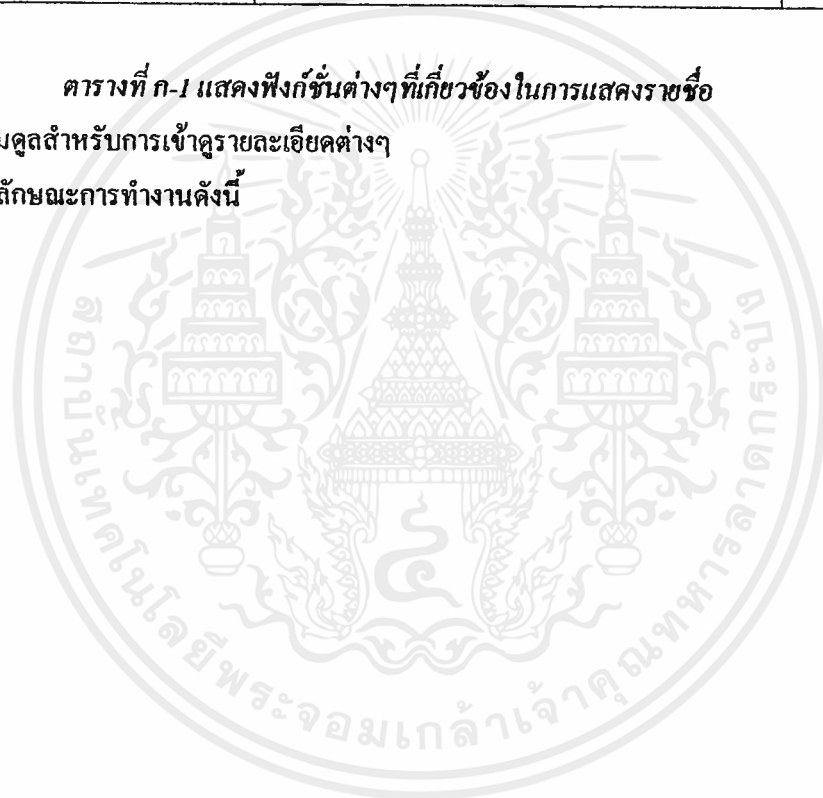
มีฟังก์ชันต่างๆที่เกี่ยวข้องใน โปรแกรมดังนี้

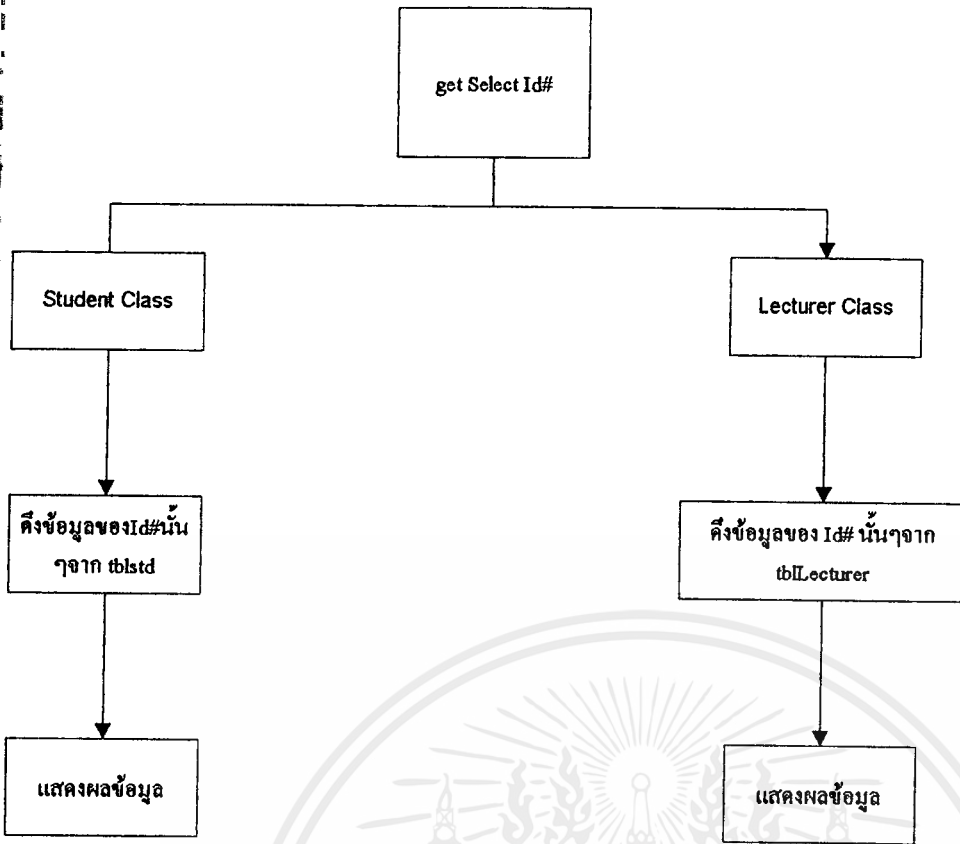
ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
ShowOnly(String year)	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงรายชื่อนักศึกษาที่เข้าเรียนเฉพาะในปีนั้นๆที่ระบุใน Show Only ลงใน Multilist	Form1.java
ConnectDbLecturer()	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงรายชื่อของอาจารย์ลงใน Multilist	Form1.java
ConnectDbStd()	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงรายชื่อของนักศึกษาที่จบไปแล้วทั้งหมดลงใน Multilist	Form1.java

ตารางที่ ก-1 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการแสดงรายชื่อ

1.2 โมดูลสำหรับการเข้าดูรายละเอียดต่างๆ

มีลักษณะการทำงานดังนี้





รูปที่ ก-2 Flowchart Diagram แสดงการเข้าดูรายละเอียด

มีฟังก์ชันต่างๆที่เกี่ยวข้องในโปรแกรมดังนี้

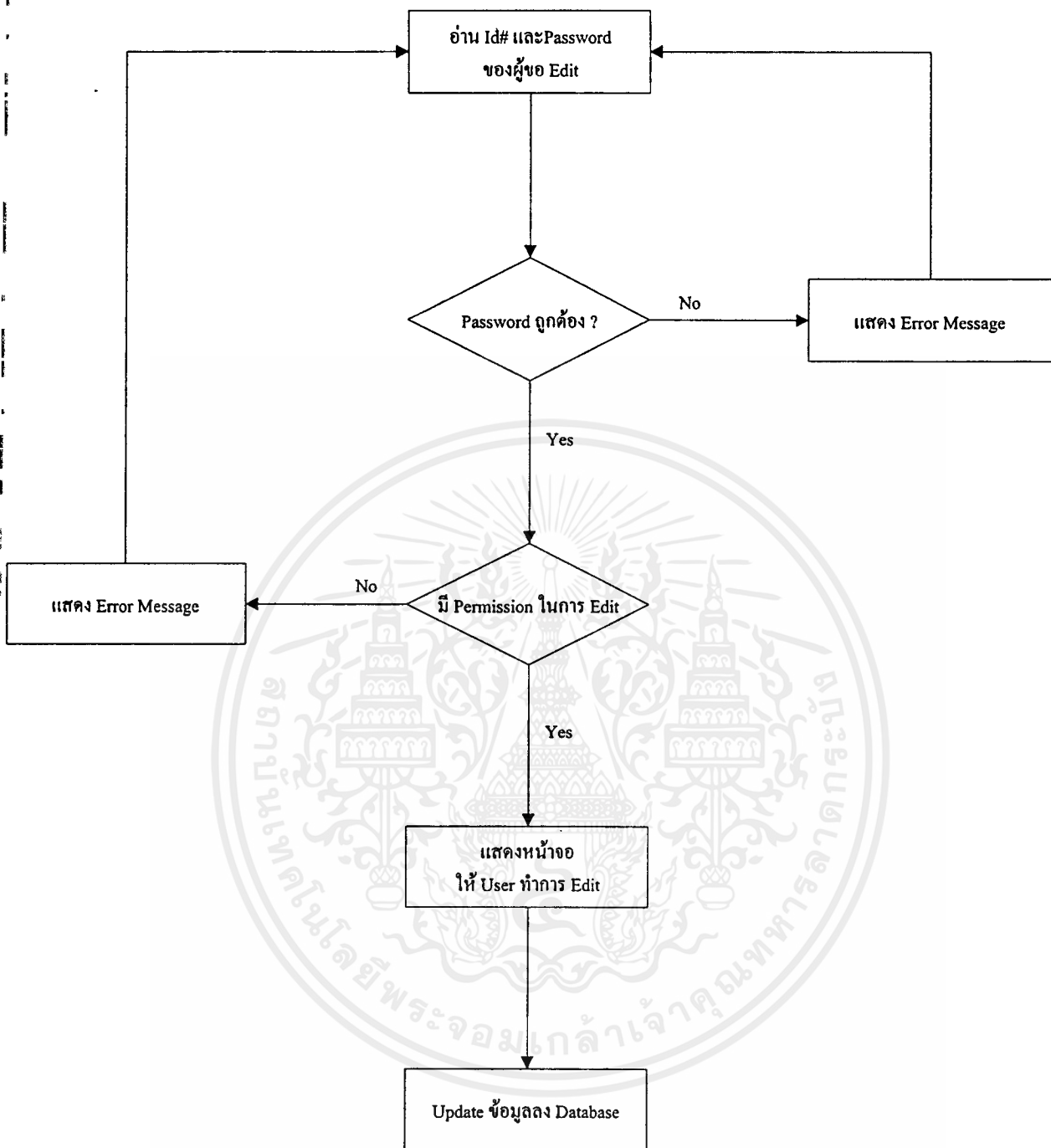
ชื่อฟังก์ชัน	หน้าที่, การทำงาน	อยู่ในไฟล์
LecturerDetails()	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงรายละเอียดของอาจารย์ท่านนั้นๆที่ถูกเลือกลงในหน้าจอแสดงรายละเอียด	Dialog1.java
StdDetails()	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงรายละเอียดของนักศึกษาท่านนั้นๆที่ถูกเลือกลงในหน้าจอแสดงรายละเอียด	Dialog1.java

ตารางที่ ก-2 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการเข้าดูรายละเอียดต่างๆ

1.3 โมดูลสำหรับการ Edit และการตรวจสอบ Security

มีลักษณะการทำงานดังนี้





รูปที่ ก-3 Flowchart Diagram แสดงการตรวจสอบ Security ในการ Edit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีฟังก์ชันต่างๆที่เกี่ยวข้องใน โปรแกรมดังนี้

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
UpdateLecturer()	ใช้ใน Update ฐานข้อมูลรายละเอียดของอาจารย์ ทำนองๆจากหน้าจอแสดงรายละเอียดในการ Edit	Dialog1.java
UpdateStd()	ใช้ใน Update ฐานข้อมูลรายละเอียดของนักศึกษา ทำนองๆจากหน้าจอแสดงรายละเอียดในการ Edit	Dialog1.java
ChkSecurity()	ใช้ในการติดต่อฐานข้อมูลเพื่อตรวจสอบ Security ในการขอทำการ Edit	Security.java

ตารางที่ ก-3 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการ Edit และตรวจสอบ Security

1.4 ฟังก์ชันอื่นๆที่ใช้ในโปรแกรม

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
ConnectToDatabase()	ใช้ในการทำการติดต่อระบบฐานข้อมูลครั้งแรก	Form1.java

ตารางที่ ก-4 แสดงฟังก์ชันอื่นๆที่ใช้ในโปรแกรมทะเบียนรุ่น

ตัวแปรต่างๆที่ใช้ในโปรแกรมมีดังนี้

ชื่อตัวแปร	ชนิดของข้อมูลที่เก็บ	อยู่ในไฟล์
Tablename	ชื่อของตารางที่ใช้ในการติดต่อฐานข้อมูล	Form1.java
SelectedId	รหัส (ID#) ของอาจารย์หรือนักศึกษาที่ถูกเลือก	Form1.java
WarningText	ข้อความที่จะให้ทำการแสดงในหน้าต่างของข้อความเตือน	Form1.java
Editmode	สถานะในการใช้งานหน้าจอรายละเอียดว่าเป็นสถานะปกติหรือให้ทำการ Edit	Form1.java
url	URL ที่ใช้ในการติดต่อระบบฐานข้อมูล (URL ของ Database Server ที่เก็บฐานข้อมูล)	Form1.java
Con	Connection ที่ใช้ในการติดต่อระบบฐานข้อมูล	Form1.java
PicURL	URL ที่เก็บรูปภาพของฐานข้อมูล	Dialog1.java
PicName	ชื่อรูปภาพที่จะเก็บลงในฐานข้อมูล	Dialog1.java
EditPassword	Password ที่ใช้ในการตรวจสอบ Security ในการ Edit	Security.java

ตารางที่ ก-5 แสดงตัวแปรต่างๆที่ใช้ใน โปรแกรมทะเบียนรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญได้หน้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต (Lecturer Discussion)

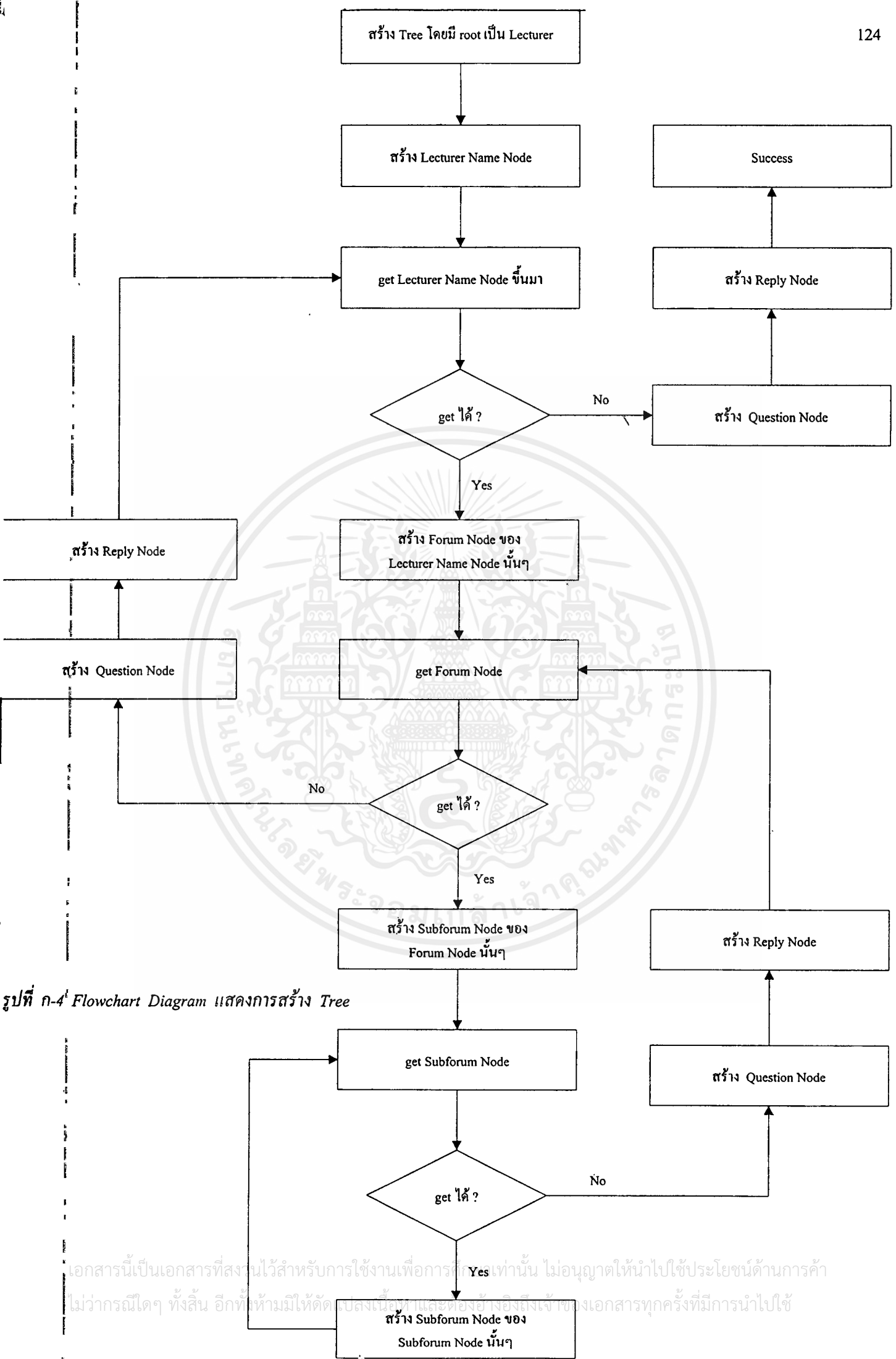
เก็บอยู่ในไฟล์โปรเจคของ Visual Café 2 ชื่อ Dc3.vpj และมีโมดูลต่างๆดังนี้

2.1 โมดูลสำหรับการสร้าง Tree

มีลักษณะการทำงานดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-4 Flowchart Diagram แสดงการสร้าง Tree

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

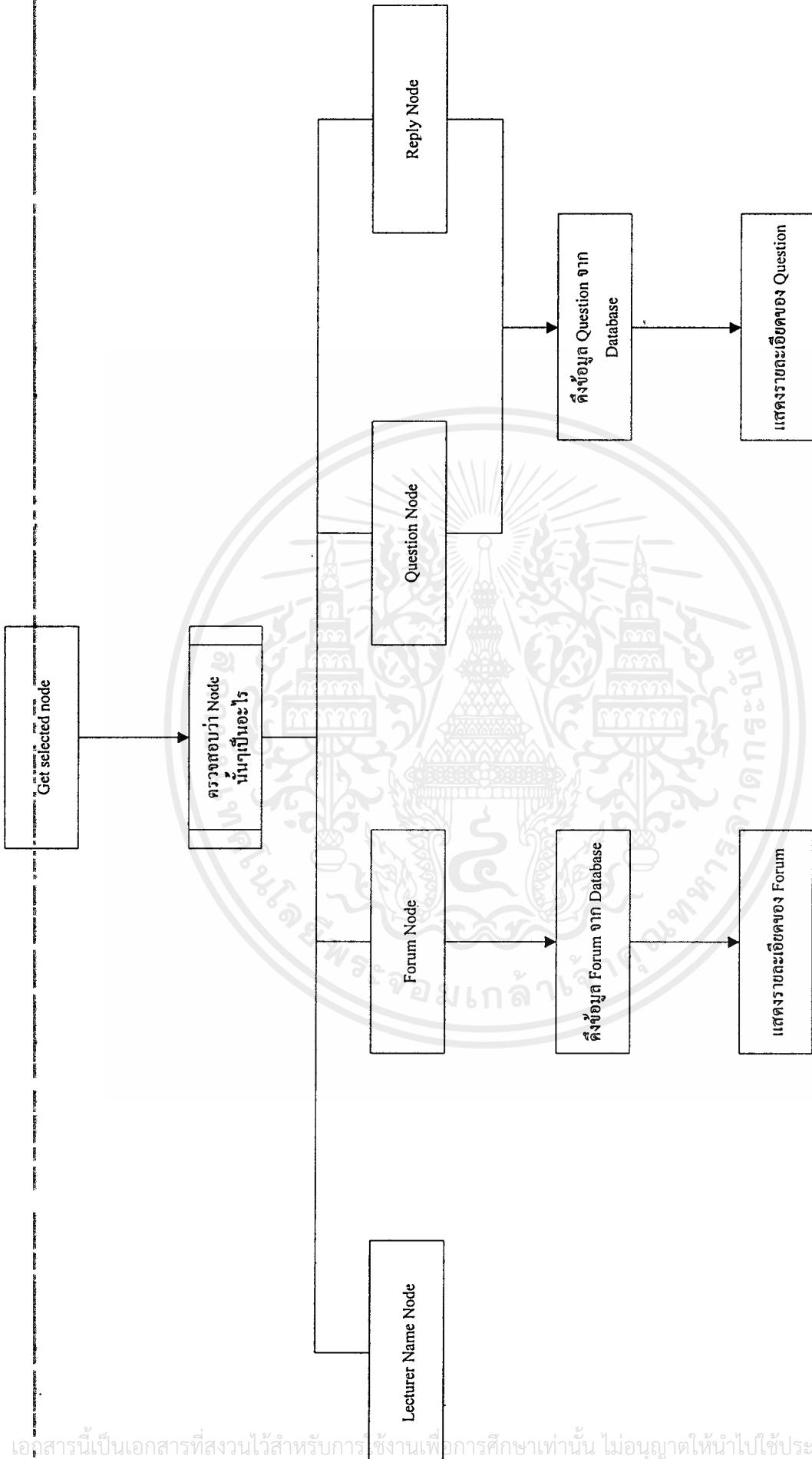
มีฟังก์ชันต่างๆที่เกี่ยวข้องในโปรแกรมดังนี้

ชื่อฟังก์ชัน	หน้าที่, การทำงาน	อยู่ในไฟล์
MakeNameNode()	ใช้ในการสร้าง Lecturer Name Node	FrmMain.java
MakeForumNode()	ใช้ในการสร้าง Forum Node ของแต่ละ Lecturer Name Node	FrmMain.java
MakeQuestionNode()	ใช้ในการสร้าง Question Node	FrmMain.java
MakeReplyNode(String QuestionId)	ใช้ในการสร้าง Reply Node	FrmMain.java
DoSubforum()	ใช้ในการสร้าง Subforum Node	FrmMain.java
MakeSubForumNode()	ใช้ในการสร้าง Subforum Node	FrmMain.java
MakeSubOfSubFrNode()	ใช้ในการสร้าง Subforum Node	FrmMain.java
MakeSubOfSubFrNode1()	ใช้ในการสร้าง Subforum Node	FrmMain.java
MakeSubOfSubFrNode2()	ใช้ในการสร้าง Subforum Node	FrmMain.java
MakeSubOfSubFrNode3()	ใช้ในการสร้าง Subforum Node	FrmMain.java
MakeSubOfSubFrNode4()	ใช้ในการสร้าง Subforum Node	FrmMain.java
MakeNoForumQuestion()	ใช้ในการสร้าง Question Node ที่ไม่มี Forum Node รองรับ	FrmMain.java

ตารางที่ ก-6 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการสร้าง Tree

2.2 โมดูลสำหรับการดูรายละเอียด (Details)

มีลักษณะการทำงานดังนี้



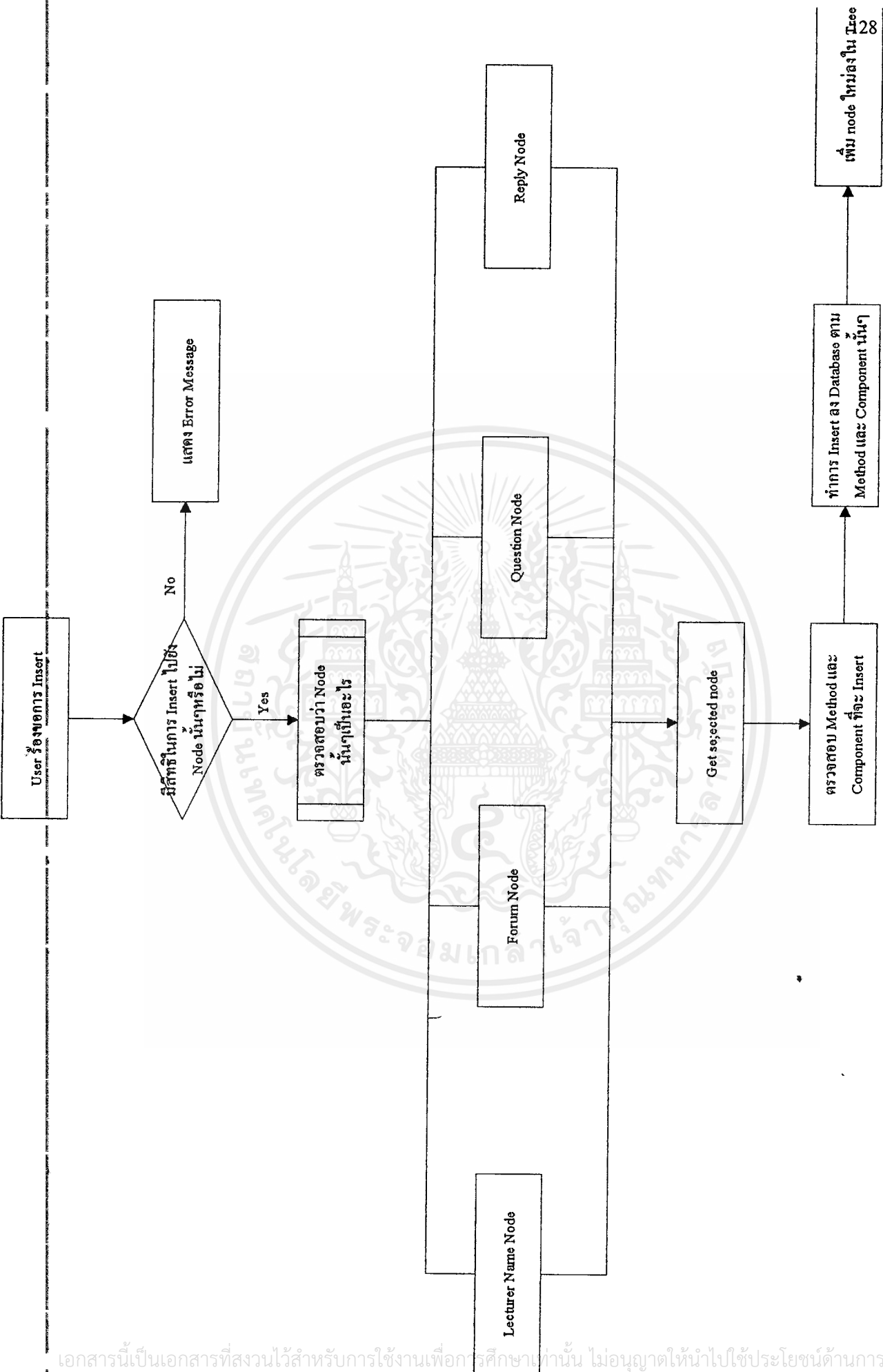
รูปที่ ก-5 Flowchart Diagram แสดงการดูรายละเอียด

มีฟังก์ชันต่างๆที่เกี่ยวข้องใน โปรแกรมดังนี้

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
ShowOldData()	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงข้อมูลเดิมของ Forum นั้นๆขึ้นมายังหน้าจอแสดงรายละเอียดของ Forum	Frdetails.java
ConnectDbQuestion()	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงข้อมูลรายละเอียดของ Question นั้นๆขึ้นมายังหน้าจอแสดงรายละเอียดของ Question	Qdetails.java
byLecturer(String name)	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงชื่อผู้ตั้งคำถามนั้นๆขึ้นมายังหน้าจอแสดงรายละเอียดของ Question	Qdetails.java
ConnectDbReply()	ใช้ในการติดต่อฐานข้อมูลเพื่อแสดงข้อมูลรายละเอียดของคำตอบของคำถามนั้นๆขึ้นมายังหน้าจอแสดงรายละเอียดของ Question	Qdetails.java

ตารางที่ ก-7 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการดูรายละเอียด

2.3 โมดูลสำหรับการเพิ่ม node หัวข้อคำถาม , คำถาม , คำตอบของคำถาม
มีลักษณะการทำงานดังนี้



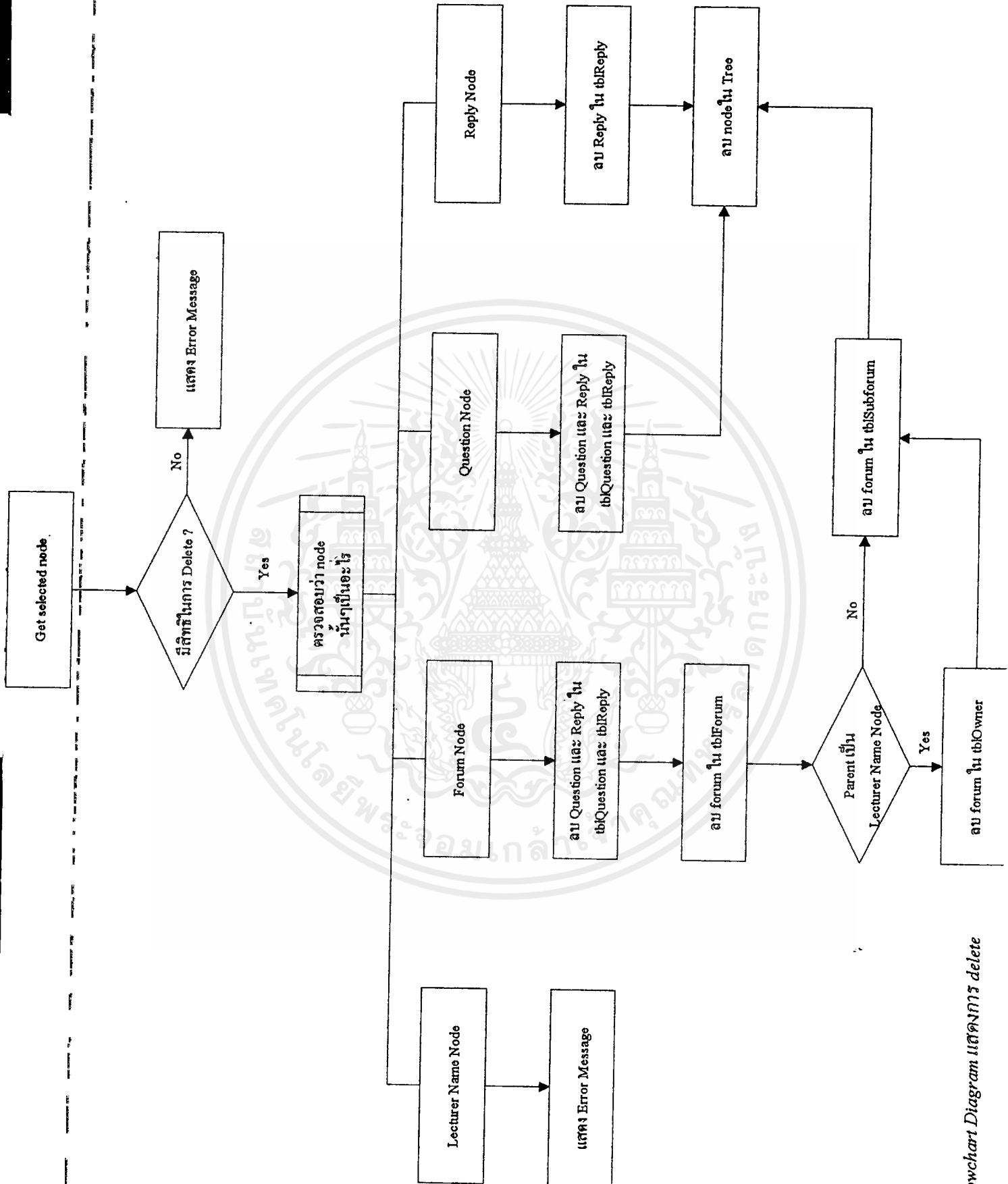
รูปที่ ๓ -6 Flowchart Diagram แสดงการ insert

มีฟังก์ชันต่างๆที่เกี่ยวข้องใน โปรแกรมดังนี้

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
InstblForum()	ใช้สำหรับการเพิ่มหัวข้อคำถามใหม่ลงในฐานข้อมูลที่ตาราง tblForum	InsertForum.java
GetFrid()	ใช้ในการ Get Frid# ของหัวข้อคำถามใหม่ที่ใส่ลงในฐานข้อมูลแล้วถูก Generate Frid# ขึ้นมาโดยอัตโนมัติเพื่อใช้ในการ Insert ลงตารางอื่นๆที่เกี่ยวข้องในฐานข้อมูลด้วย	InsertForum.java
InstblOwner()	ใช้สำหรับการเพิ่มหัวข้อคำถามใหม่ลงในฐานข้อมูลที่ตาราง tblOwner	InsertForum.java
InstblSubforum (String ParentId)	ใช้สำหรับการเพิ่มหัวข้อคำถามใหม่ลงในฐานข้อมูลที่ตาราง tblSubforum	InsertForum.java
InstblQuestion(String ForumId)	ใช้สำหรับการเพิ่มคำถามใหม่ลงในฐานข้อมูลที่ตาราง tblQuestion	InsertQuestion.java
InsertReply()	ใช้สำหรับการคำตอบใหม่ลงในฐานข้อมูลที่ตาราง tblReply	ReplyForm.java
FindRdate()	ใช้สำหรับการหาวันที่เพิ่มคำตอบใหม่นั้นๆเข้าไปในฐานข้อมูล	ReplyForm.java

ตารางที่ ก-8 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการเพิ่ม node ต่างๆใน Tree

2.4 โมดูลสำหรับการลบ node หัวข้อคำถาม , คำถาม , คำตอบของคำถาม
มีลักษณะการทำงานดังนี้



รูปที่ ก -7 Flowchart Diagram แสดงการ delete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

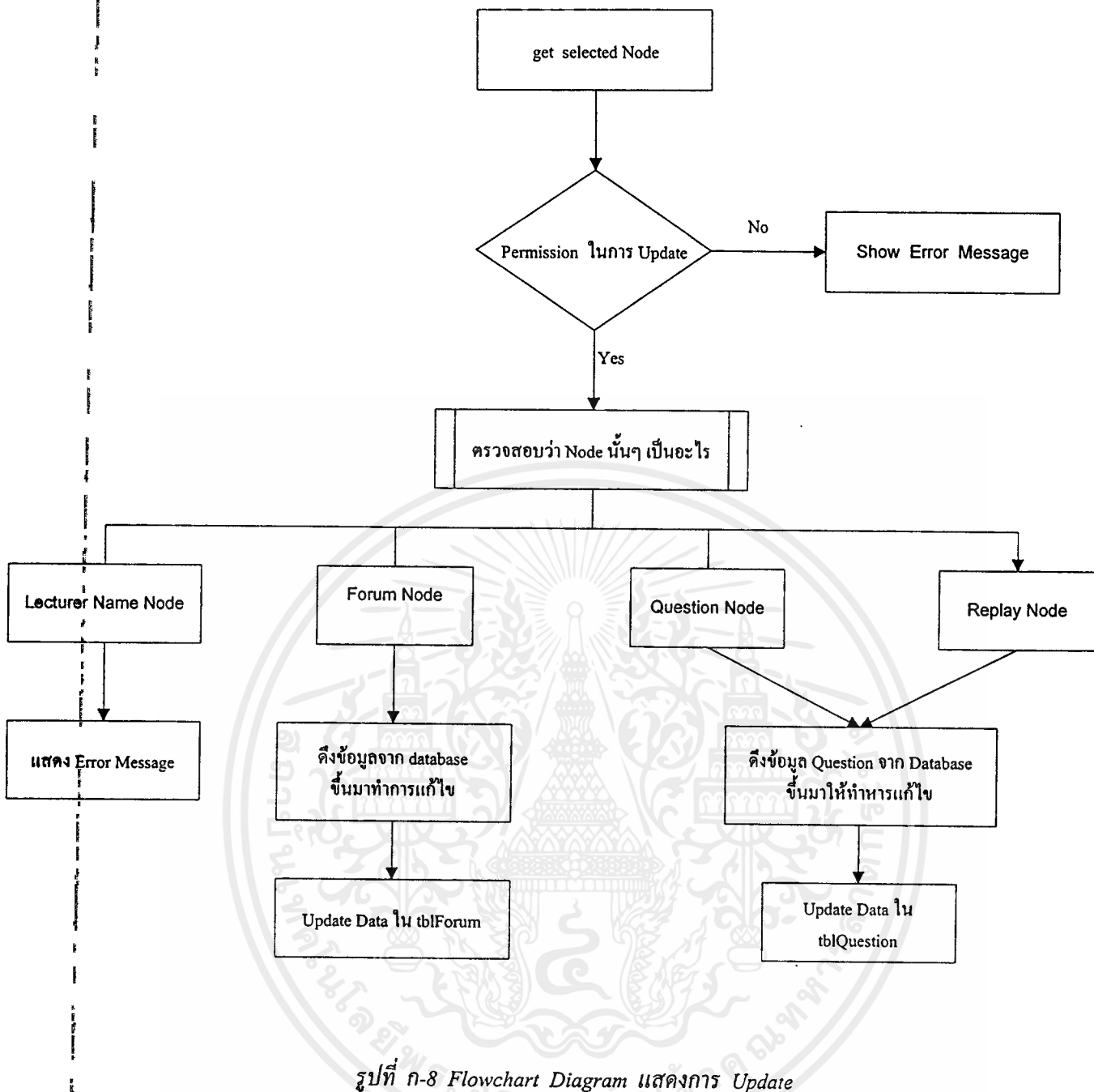
มีฟังก์ชันต่างๆที่เกี่ยวข้องในโปรแกรมดังนี้

ชื่อฟังก์ชัน	หน้าที่, การทำงาน	อยู่ในไฟล์
DeleteRow(String table,String Condition)	ใช้ในการลบแถวในตารางที่เก็บใน parameter table โดยมี Condition ตาม parameter Condition	DelWarning.java
DeleteReply()	ใช้ในการลบคำตอบออกจากฐานข้อมูล	DelWarning.java

ตารางที่ ก-9 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการลบ node ต่างๆใน Tree

2.5 โมดูลสำหรับการแก้ไขปรับปรุงหัวข้อคำถาม , คำถาม
มีลักษณะการทำงานดังนี้





รูปที่ ก-8 Flowchart Diagram แสดงการ Update

มีฟังก์ชันต่างๆที่เกี่ยวข้องใน โปรแกรมดังนี้

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
ShowOldData()	ใช้ในการแสดงข้อมูลเดิมของหัวข้อคำถามนั้นๆ	UpdateForum.java
UpdateForumData (String header,String comment)	ใช้ในการ Update ข้อมูลของหัวข้อคำถามลงฐานข้อมูล	UpdateForum.java
ShowOldData()	ใช้ในการแสดงข้อมูลเดิมคำถามนั้นๆ	UpdateQuestion.java
UpdateQuestionData (String head,String detail)	ใช้ในการ Update ข้อมูลของคำถามลงฐานข้อมูล	UpdateQuestion.java

ตารางที่ ก-10 แสดงฟังก์ชัน ต่างๆที่เกี่ยวข้องในการแก้ไขปรับปรุงหัวข้อคำถามและคำถาม

2.6 ฟังก์ชันอื่นๆที่ใช้ในโปรแกรม

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
ChangeNameToId(String name)	ใช้ในการเปลี่ยนจากชื่อ Lecturer Name Node นั้นๆไปเป็น ID#	FrmMain.java
ChangeForumToId()	ใช้ในการเปลี่ยนจากชื่อหัวข้อคำถามไปเป็น Frid#	FrmMain.java
ChangeNowToQid(String name)	ใช้ในการตรวจสอบว่า node ที่กำลังถูกเลือกอยู่ขณะนั้นเป็น Question Node หรือไม่ ถ้าใช่ก็จะเปลี่ยนจากชื่อคำถามไปเป็น Qid#	FrmMain.java
ChangeNowToRid(String name)	ใช้ในการตรวจสอบว่า node ที่กำลังถูกเลือกอยู่ขณะนั้นเป็น Reply Node หรือไม่ ถ้าใช่ก็จะเปลี่ยนจากชื่อของคำตอบไปเป็น Rid#	FrmMain.java
ChangeNowToFrid(String name)	ใช้ในการตรวจสอบว่า node ที่กำลังถูกเลือกอยู่ขณะนั้นเป็น Forum Node หรือไม่ ถ้าใช่ก็จะเปลี่ยนจากชื่อหัวข้อคำถามไปเป็น	FrmMain.java

	Frid#	
ConnectToDatabase()	ใช้ในการติดต่อระบบฐานข้อมูลในตอนเริ่มต้น	Security.java
ChkSecurity(String UserName)	ใช้ในการตรวจสอบ Security ในการเข้าใช้งานระบบ	Security.java

ตารางที่ ก-11 แสดงฟังก์ชันอื่นๆที่เกี่ยวข้องใน โปรแกรม *Lecturer Discussion*
ตัวแปรต่างๆที่ใช้ในโปรแกรมมีดังนี้

ชื่อตัวแปร	ชนิดของข้อมูลที่เก็บ	อยู่ในไฟล์
Root	เก็บ root node ของ Tree	FrmMain.java
Father	เป็น temporary node	FrmMain.java
Child	เป็น temporary node	FrmMain.java
Now	เก็บ node ปัจจุบันที่ถูกเลือก	FrmMain.java
ForumNode	เป็น temporary node	FrmMain.java
SubfrNode	เป็น temporary node	FrmMain.java
subfrNode1	เป็น temporary node	FrmMain.java
subfrNode2	เป็น temporary node	FrmMain.java
subfrNode3	เป็น temporary node	FrmMain.java
subfrNode4	เป็น temporary node	FrmMain.java
subfrNode5	เป็น temporary node	FrmMain.java
Fatherfr	เก็บ forum node ที่เป็น node พ่อของ node ที่ถูกเลือก	FrmMain.java
Qnode	เป็น temporary node	FrmMain.java
Mode	เก็บสถานะในการเข้าใช้ระบบ	FrmMain.java
User	เก็บชนิดของผู้เข้าใช้ระบบ (Guest , Lecturer)	FrmMain.java
Frid	เก็บ Frid#	FrmMain.java
ParentFrid	เก็บ Frid# ของ forum node ที่เป็น node พ่อ	FrmMain.java
Rid	เก็บ Rid#	FrmMain.java
Qid	เก็บ Qid#	FrmMain.java
WarningText	เก็บข้อความเตือนที่จะแสดงที่หน้าจอข้อความเตือน	FrmMain.java
About	เก็บข้อความเตือนเพิ่มเติมที่จะแสดงที่หน้าจอข้อ	FrmMain.java

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ความเดือนในการ Delete ด้วย	
SelectedNodeBe	เก็บสถานะของ node ที่ถูกเลือกว่าเป็นอะไร (Iname , forum , question , reply)	FrmMain.java
SelectedNodeLink	เก็บสถานะของ node ตัวพ่อของตัวที่ถูกเลือกว่าเป็นอะไร	FrmMain.java
SelectedNodeText	เก็บข้อความของ node ที่ถูกเลือก	FrmMain.java
InsertCondition	เก็บค่าที่ใช้ระบุกรณีในการ Insert	FrmMain.java
DelCondition	เก็บค่าที่ใช้ระบุกรณีในการ Delete	FrmMain.java
HaveChild	เก็บสถานะว่า node ที่ถูกเลือกมีลูกหรือไม่	FrmMain.java
Con	เก็บ Connection ในการติดต่อระบบฐานข้อมูล	FrmMain.java
Fathername	เก็บชื่อของ node ตัวพ่อของตัวที่ถูกเลือก	FrmMain.java
Id	เก็บค่า Id# ของ Lecturer Name	FrmMain.java
Subid	เก็บค่า Subid#	FrmMain.java
Frheader	เก็บ Frheader	FrmMain.java
Frlevel	เก็บระดับของหัวข้อคำถาม	FrmMain.java
Qicon	เก็บรูปภาพ Icon ของ Question Node ปกติ	FrmMain.java
Oqicon	เก็บรูปภาพ Icon ของ Question Node เมื่อเปิด	FrmMain.java
Bicon	เก็บรูปภาพ Icon ของ Forum Node ปกติ	FrmMain.java
Obicon	เก็บรูปภาพ Icon ของ Forum Node เมื่อเปิด	FrmMain.java
face1	เก็บรูปภาพ Icon ของ Lecturer Name Node ปกติ	FrmMain.java
face2	เก็บรูปภาพ Icon ของ Lecturer Name Node เมื่อเปิด	FrmMain.java
Ricon	เก็บรูปภาพ Icon ของ Reply Node	FrmMain.java
SelectedImage	เก็บรูปภาพ Icon ของ Node ที่ถูกเลือก	FrmMain.java
FatherImage	เก็บรูปภาพ Icon ของตัวพ่อของ Node ที่ถูกเลือก	FrmMain.java
TruePath	เก็บสถานะว่าทำการ Insert , Delete , Update ถูกเส้นทางหรือไม่	FrmMain.java
InsertCase	เก็บค่าระบุกรณีในการ Insert	InsertForm.java
NewFrid	เก็บ Frid# ของหัวข้อคำถามที่ถูกเพิ่มเข้าไปใหม่	InsertForum.java
ReplierName	เก็บชื่อผู้ทำการตอบคำถาม	ReplyForm.java
ReplierAddr	เก็บ Email Address ของผู้ทำการตอบคำถาม	ReplyForm.java

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ReplyDetails	เก็บรายละเอียดของคำตอบ	ReplyForm.java
Rdate	เก็บวันเวลาที่ทำการตอบคำถาม	ReplyForm.java
url	เก็บ URL ของตัว Server ที่เก็บฐานข้อมูล	Security.java
Pwd	เก็บ Password ของผู้เข้าใช้ระบบ	Security.java
UserName	เก็บชื่อของผู้เข้าใช้ระบบ	Security.java
RightUser	เก็บสถานะของการเข้าใช้ระบบว่าอนุญาตให้ทำการ Insert , Delete , Update ได้หรือไม่	Security.java

ตารางที่ ก-12 แสดงตัวแปรต่างๆที่ใช้ใน โปรแกรม Lecturer Discussion

3. โปรแกรมระบบการนัดหมายการประชุม (Meeting Scheduler)

เก็บอยู่ในไฟล์โปรเจกของ Visual Café 2 ชื่อ real.vpj และมีโมดูลต่างๆดังนี้

3.1 โมดูลที่ใช้ในการตรวจสอบพาสเวิร์ด

ชื่อฟังก์ชัน	หน้าที่, การทำงาน	อยู่ในไฟล์
Connectdb	ทำการเชื่อมต่อกับฐานข้อมูลรวมทั้งตรวจสอบสิทธิ์ของผู้ที่เข้ามาเรียกใช้โปรแกรม	PasswordDialog.java

ตารางที่ ก-13 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการตรวจสอบพาสเวิร์ด

3.2 โมดูลที่ใช้ในการสร้างหน้าจอหลัก

ชื่อฟังก์ชัน	หน้าที่, การทำงาน	อยู่ในไฟล์
Getdatexxx	สร้าง date object จากสตริงที่ส่งเข้าไป	Real.java
Getsysdate	ทำการดึงค่าวัน เวลา ปัจจุบันให้กับ date object ที่กำหนด	Real.java
Comparedate	ทำการเปรียบเทียบระยะห่างของเวลาของ date object ที่มีอยู่กับวัน เวลาปัจจุบัน	Real.java
Setsendbox	ทำการดึงค่ารายละเอียดของการนัดหมายการประชุมที่ได้รับมาเก็บไว้ในmultilist	Real.java
Updatestate	ใช้ในการปรับเปลี่ยนค่าในคอลัมน์ flag และคอลัมน์ mark ใน multilist ที่เกิดจากการกระทำบางอย่างของผู้ใช้เช่น การดับเบิลคลิกในเซลล์ที่ตรงกับคอลัมน์นั้นๆ	Real.java
Showimage	กำหนดชื่อของรูปภาพให้กับ image object	Real.java

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Showitem	ทำการแสดงการนัดหมายการประชุมให้อยู่ใน multilist	Real.java
Set_total_msg	แสดงจำนวนของการนัดหมายการประชุมที่มีอยู่ทั้งหมด	Real.java
Set_unread_msg	แสดงจำนวนของการนัดหมายการประชุมที่ยังไม่ได้เข้าไปดูรายละเอียด	Real.java

ตารางที่ ก-14 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการจัดการหน้าจอหลัก

3.3 โมดูลที่ใช้ในการแสดงรายละเอียดการนัดหมายการประชุม

ชื่อฟังก์ชัน	หน้าที่, การทำงาน	อยู่ในไฟล์
Set_status	ทำการเช็คค่าในRadio Button ตามการตัดสินใจของผู้ใช้ที่ถูกเก็บอยู่ในฐานข้อมูล	Frame1.java
Input_textarea	ทำการแสดงรายละเอียดของการนัดหมายประชุมใน text area	Frame1.java
Accept_refuse	แสดงจำนวนการตัดสินใจในทั้ง 2 ชนิดของผู้ที่ได้รับการนัดหมายการประชุมทุกคนว่ามีจำนวนเท่าใด	Frame1.java
Set_all	ทำการแสดงรายละเอียดในทุกๆ textfield	Frame1.java

ตารางที่ ก-15 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการแสดงรายละเอียดการนัดหมายการประชุม

3.4 โมดูลที่ใช้ในการจัดการส่งการนัดหมายการประชุม

Allvalueofc()	ทำการใส่ค่าเวลาทั้งหมดที่อนุญาตให้ผู้ใช้เลือกได้เข้าไปในอาร์เรย์	Frame3.java
Recover()	ทำการเช็คค่าของเวลาทั้งหมดใน choice ใหม่	Frame3.java
Setchoice2(String)	ทำการเช็คค่าของเวลาทั้งหมดใน choice ใหม่	Frame3.java
Getdate (String,String)	ทำการสร้าง date object จากสตริงทั้งหมดที่ถูกส่งเข้าไปในฟังก์ชันนี้	Frame3.java
Intrav(treenode)	ทำการหาค่าสตริงที่เหมือนกับชื่อของ treenode ใน treeview	Frame3.java
Test_exit(String)	ทำการตรวจสอบว่าสตริงที่รับเข้ามามีอยู่ใน treeview หรือไม่	Frame3.java

Prepare_test()	ทำการวนลูปดึงค่าแอสเครสของผู้รับที่อยู่ใน textfield ทั้งหมดมาตรวจสอบว่ามีอยู่จริงในฐานข้อมูลหรือไม่	Frame3.java
Checkvalidate()	ทำการเช็คว่ามีผู้ป้อนข้อมูลครบหรือไม่	Frame3.java
Setsystem_dt()	ทำการตั้งค่าวันเวลาปัจจุบันให้กับ date object ที่กำหนด	Frame3.java
Test_forid(String)	ตรวจสอบว่าสตริงที่รับเข้ามานั้นเป็นรหัสประจำตัวหรือชื่อกลุ่ม	Frame3.java
Insert_intodb(String)	ทำการใส่ข้อมูลของการนัดหมายการประชุมให้กับผู้ที่เข้าร่วมการประชุมทุกคนใน tblMeeting ซึ่งอยู่ในฐานข้อมูล	Frame3.java
Prepare_insert()	เตรียมลำดับขั้นตอนในการใส่ข้อมูลการนัดหมายการประชุมลงไปในฐานข้อมูล	Frame3.java
Insertforsender()	ทำการใส่ข้อมูลของการนัดหมายการประชุมเข้าไปใน tblSchedule ที่อยู่ในฐานข้อมูล	Frame3.java
Prepare_send()	ทำการวนลูปดึงค่าแอสเครสของผู้รับที่อยู่ในทุกๆ textfield มาตรวจสอบ	Frame3.java
Search_node(String)	ทำการหาโหนดที่มีชื่อเหมือนกับสตริงที่รับเข้ามา	Frame3.java
Updateid(String)	ทำการตรวจสอบว่ามีกรนัดหมายการประชุมที่มีเวลาซ้ำซ้อนกันสำหรับผู้ที่จะเข้าร่วมการประชุมหรือไม่	Frame3.java
Showconflict (String...)	ทำการแสดงเวลาที่เกิดการซ้ำซ้อนกันให้ผู้ผู้ได้ทราบ	Frame3.java
Intrav2(treenode)	ทำการรีเคอร์ซีฟหาโหนดที่ต้องการ	Frame3.java
Match2(treenode)	ทำการเปรียบเทียบโหนดที่ต้องการ	Frame3.java
Create_conf_frame()	ทำการสร้างเฟรม Conflict	Frame3.java

ตารางที่ ก-16 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการจัดการส่งการนัดหมายการประชุม

3.5 โมดูลการเลือกกลุ่มที่ต้องการ

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
Newupdate_node (treenode,String)	ทำการ insert node ชื่อสมาชิกหรือชื่อกลุ่มที่ผู้ใช้เป็นคนสร้างขึ้นมา	Frame8.java

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Update_personal()	เรียกฟังก์ชัน newupdate_node เพื่อทำการ insert node ที่ต้องการ	Frame8.java
Update_std (String,treenode)	ใช้ในการ insert ชื่อ-รหัส ประจำตัวของนักศึกษาเข้าไปใน treeview	Frame8.java
Inserttree()	กำหนดโครงสร้างของ treeview และทำการ insert ชื่อ-รหัสประจำตัวของอาจารย์เข้าไปใน treeview ด้วย	Frame8.java

ตารางที่ ก-17 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการจัดการส่งการนัดหมายการประชุม

3.6 โมดูลแสดงหน้าจอการจัดตั้งกลุ่มของผู้ใช้ขึ้นมาใหม่

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
Update_std (String,treenode)	ใช้ในการ insert ชื่อ-รหัส ประจำตัวของนักศึกษาเข้าไปใน treeview	Frame11.java
Inserttree()	กำหนดโครงสร้างของ treeview และทำการ insert ชื่อ-รหัสประจำตัวของอาจารย์เข้าไปใน treeview ด้วย	Frame11.java

ตารางที่ ก-18 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการจัดตั้งกลุ่มของผู้ใช้ขึ้นมาใหม่

4. โปรแกรมการเปลี่ยน Password ของระบบ

เก็บอยู่ในไฟล์โปรเจกของ Visual Café 2 ชื่อ setPass.vpj และมีโมดูลต่างๆดังนี้

4.1 โมดูลที่ใช้ในการเปลี่ยน Password

ชื่อฟังก์ชัน	หน้าที่ , การทำงาน	อยู่ในไฟล์
ChkUser	ใช้ในการตรวจสอบว่าเป็น User ในระบบหรือไม่	ChangePass.java
UpdatePassword (String table, String Id)	ใช้ในการ Update Password ลงฐานข้อมูล	Confirm.java

ตารางที่ ก-19 แสดงฟังก์ชันต่างๆที่เกี่ยวข้องในการเปลี่ยน Password

4.2 ฟังก์ชันอื่นๆที่ใช้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟังก์ชัน	หน้าที่, การทำงาน	อยู่ในไฟล์
ConnectToDatabase()	ใช้ในการติดต่อระบบฐานข้อมูลในตอนเริ่มต้น	ChangePass.java

ตารางที่ ก-20 แสดงฟังก์ชันอื่นๆที่เกี่ยวข้องในโปรแกรม *Change Password*
ตัวแปรต่างๆที่ใช้ในโปรแกรมนี้นี้

ชื่อตัวแปร	ชนิดของข้อมูลที่เก็บ	อยู่ในไฟล์
url	เก็บ URL ของ Server ที่เก็บฐานข้อมูล	ChangePass.java
Con	เก็บ Connection ในการติดต่อฐานข้อมูล	ChangePass.java
WarningText	ข้อความที่จะให้ทำการแสดงในหน้าต่างของข้อความเตือน	ChangePass.java
NewPass	เก็บค่า Password ใหม่ที่ผู้ใช้เปลี่ยน	ChangePass.java
Id	เก็บรหัส (ID#) ของนักศึกษาหรืออาจารย์	ChangePass.java
User	เก็บชื่อ-นามสกุลของที่เข้ามาใช้	ChangePass.java

ตารางที่ ก-21 แสดงตัวแปรต่างๆที่ใช้ในโปรแกรม *Change Password*

ภาคผนวก ข.

คู่มือการใช้งาน

ส่วนประกอบของโครงการ

ส่วนประกอบหลักของโครงการประกอบไปด้วย 3 ส่วนคือ

1. ระบบทะเบียนรุ่น
2. ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต
3. ระบบการนัดหมายการประชุม
4. การเปลี่ยนพาสเวิร์ดของผู้ใช้

โดยลักษณะสำคัญของแต่ละระบบคือ

รูปแบบการทำงาน

1. ระบบทะเบียนรุ่น

เป็นระบบที่จัดเก็บข้อมูลของอาจารย์และนักศึกษาในภาควิชาคอมพิวเตอร์ที่จบการศึกษาแล้ว เพื่อให้ผู้ที่สนใจหรือต้องการทราบข้อมูลเหล่านั้น สามารถเข้ามาดูได้ด้วยตนเองผ่านทางระบบอินเทอร์เน็ตได้

2. ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต

เป็นระบบที่ให้อาจารย์สามารถตั้งคำถามผ่านทางอินเทอร์เน็ตเพื่อให้ผู้ที่สนใจเข้ามาตอบได้ หรืออาจารย์สามารถส่งการบ้านผ่านทางอินเทอร์เน็ตได้ โดยที่ใครๆก็สามารถตอบผ่านทางระบบอินเทอร์เน็ตได้ด้วย

3. ระบบการนัดหมายการประชุม

เป็นระบบที่อาจารย์และนักศึกษาภายในภาควิชาคอมพิวเตอร์สามารถนัดหมายการประชุมหรือนัดพบกันได้ผ่านทางระบบอินเทอร์เน็ต โดยที่อาจารย์และนักศึกษาสามารถกำหนดหรือตั้งกลุ่มในการนัดหมายได้ด้วยตนเองด้วย

ระบบทะเบียนรุ่น

จะมีหน้าจอหลักในการทำงานคือ

Com. Member Info

Class : Student

Only Year : 2537 Show

Member List :

ID#	Name
37014001	Kanokporn Nabangplasoi
37014011	Korkit Veeraarchakun
37014014	Karn Sukwong
37014215	Boonchai Rattanawaranun
37014216	Boonchai lammetta
37014050	Jakkapong Siriboon
37014040	Kananart Nitithananut
37014102	Thapanee Srichoung
34014402	Viyada Anantamongkhon

ID# : 37014215 Details

Applet started.

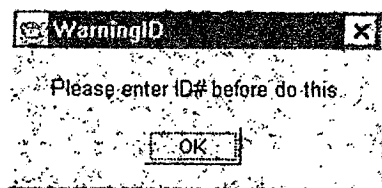
รูปที่ ข-1 หน้าจอหลักของระบบทะเบียนรุ่น

ซึ่งผู้ใช้สามารถเลือกกลุ่มเป้าหมายที่ต้องการเข้าดูข้อมูลได้โดยเลือกที่คอมโบบ็อกซ์ "Class :" ซึ่งประกอบไปด้วย 2 กลุ่มคือ อาจารย์ และนักศึกษาซึ่งถ้าผู้ใช้เลือก Class Student ก็จะมีช่องสำหรับให้กรอกเฉพาะปีที่ต้องการดูรายชื่อได้ โดยที่ลักษณะการเข้าดูรายละเอียดของข้อมูลสามารถทำได้ 2 ทางคือ

1. โดยการดับเบิลคลิกที่ชื่อของบุคคลที่ต้องการดูรายละเอียดของข้อมูลจากลิสต์หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โดยการคลิกที่ปุ่ม Details ซึ่งการคลิกที่ปุ่ม Details นี้จะต้องทำการใส่รหัสของบุคคลที่ต้องการดูรายละเอียดของข้อมูลด้วย หรือไม่ก็ต้องการเลือกชื่อของบุคคลที่ต้องการดูรายละเอียดของข้อมูลจากลิสท์ก่อน ถ้าผู้ใช้ไม่ได้ทำอย่างใดอย่างหนึ่งก่อนคลิกที่ปุ่ม Details จะมีข้อความเตือนดังนี้ปรากฏขึ้น



รูปที่ ข-2 หน้าจอแสดงข้อความเตือน (Warning Message)

หลังจากเข้าขอรายละเอียดของข้อมูลของบุคคลนั้นๆ ได้แล้ว ก็จะปรากฏหน้าจอแสดงรายละเอียดดังนี้

รูปที่ ข-3 หน้าจอแสดงข้อมูลรายละเอียดของบุคคล (Details Form)

โดยผู้ใช้งานสามารถทำการแก้ไขข้อมูลของตนเองได้โดยการคลิกที่ปุ่ม Edit เมื่อต้องการแก้ไข จากนั้นระบบจะสอบถาม Security ในการแก้ไขดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

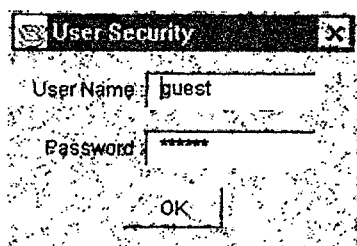
รูปที่ ข-4 หน้าจอการตรวจสอบ Security ในการแก้ไข(Edit Security)

ถ้ารหัสและพาสเวิร์ดถูกต้องและมี Permission ในการ Edit ระบบก็จะอนุญาตให้ทำการ Edit ได้ โดยหน้าจอในการ Edit จะเป็นดังนี้

รูปที่ ข-5 หน้าจอการแก้ไขข้อมูลรายละเอียดของบุคคล (Edit Form)

เมื่อผู้ใช้ทำการแก้ไขข้อมูลของตนเสร็จเรียบร้อยแล้ว และผู้แก้ไขทำการคลิกปุ่ม OK แล้ว ตัวระบบก็จะทำการแก้ไขข้อมูลให้โดยอัตโนมัติ แล้วจึงกลับไปยังหน้าจอหลักให้

ระบบการถามคำถามของอาจารย์ผ่านทางอินเทอร์เน็ต
เริ่มต้นก่อนที่จะเข้าใช้งาน ระบบจะสอบถามสิทธิในการเข้าใช้งานระบบดังนี้



รูปที่ ข-6 หน้าจอการล็อกอินเข้าใช้ระบบ (User Security Form)

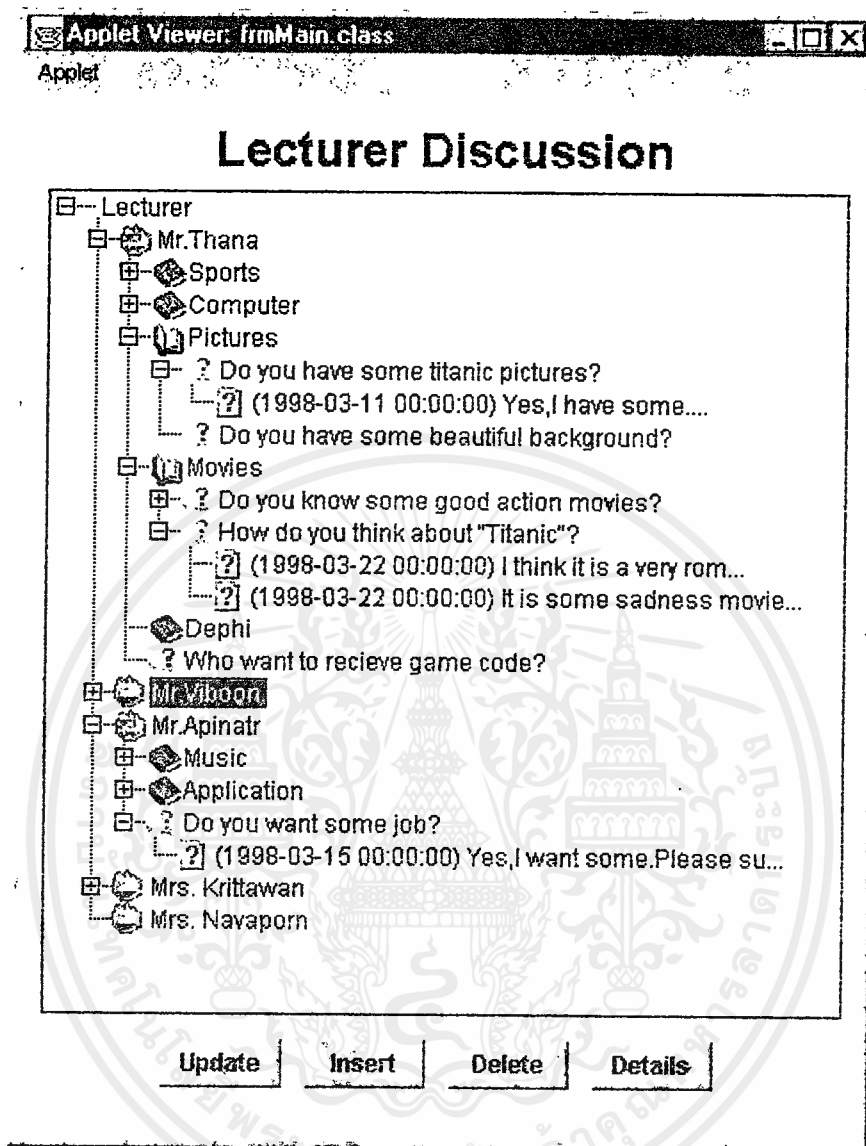
โดยสิทธิในการเข้าใช้งานระบบจะแบ่งเป็น 2 แบบคือ

1. เข้าใช้งานระบบแบบผู้ที่มาเยือน
2. เข้าใช้งานระบบแบบอาจารย์

ถ้าเข้าใช้ระบบแบบผู้ที่มาเยือนผู้เข้าใช้ระบบจะสามารถทำได้แค่เข้าไปดูรายละเอียดต่างๆ และ Add NewReply ได้เท่านั้น แต่ไม่สามารถทำการ แก้ไข, เพิ่มเติม , ลบหัวข้อคำถามหรือคำถามของอาจารย์ได้

ถ้าเข้าใช้ระบบแบบอาจารย์ผู้เข้าใช้ นอกจากจะสามารถเข้าไปดูรายละเอียดต่างๆ และ Add New Reply ได้แล้ว ยังสามารถทำการเพิ่ม, แก้ไข ,ลบหัวข้อคำถามหรือคำถาม ในส่วนของตนเอง ได้อีกด้วย

หลังจากระบบทำการตรวจสอบสิทธิในการเข้าใช้งานให้แล้ว จะปรากฏหน้าจอหลักดังนี้



รูปที่ ข-7 หน้าจอหลักของระบบ Lecturer Discussion (Main Form)

โดยเป็นการแสดงหัวข้อคำถาม , คำถาม , คำตอบ และชื่อของอาจารย์ที่เป็นเจ้าของออกมาในรูปแบบต้นไม้ โดยที่

- โหนดที่มีรูปหน้าคนอยู่ด้านหน้า เป็นโหนดที่แสดงชื่อของอาจารย์
- โหนดที่มีรูปหนังสืออยู่ด้านหน้า เป็นโหนดที่แสดงชื่อของหัวข้อคำถาม
- โหนดที่มีรูปเครื่องหมายคำถามอยู่ด้านหน้า เป็นโหนดที่แสดงคำถาม
- โหนดที่มีรูปกระดาษที่มีเครื่องหมายคำถามอยู่ด้านหน้า เป็น โหนดที่แสดงคำตอบของคำถาม

(Reply) และมีปุ่มต่างๆที่ใช้ทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ปุ่ม Update

ใช้ในการปรับปรุงหรือแก้ไขเพิ่มเติมหรือเปลี่ยนชื่อหัวข้อคำถามหรือคำถามของอาจารย์ท่านนั้นๆ

2. ปุ่ม Insert

ใช้ในการเพิ่มหัวข้อคำถามหรือคำถามใหม่เข้าไปในฐานข้อมูลและค้นไม่ว่าจะสามารถทำการเพิ่มได้เฉพาะของอาจารย์ท่านนั้นๆเท่านั้น

3. ปุ่ม Delete

ใช้ในการลบหัวข้อคำถามหรือคำถามของอาจารย์ท่านนั้นๆทิ้งไป

4. ปุ่ม Details

ใช้ในการเข้าดูรายละเอียดของคำถามและคำตอบ โดยที่หน้าจอหลักนี้จะมีการทำงานที่สำคัญคือ

1. การแก้ไขปรับปรุง (Update) หัวข้อคำถามหรือคำถาม

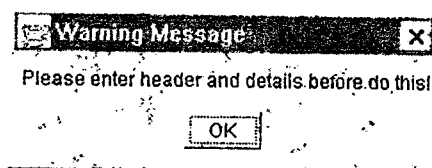
อาจารย์สามารถทำการอัปเดตหรือ เปลี่ยนชื่อหัวข้อคำถามหรือคำถามของตนเองได้โดยการคลิกที่ปุ่ม Update จากนั้นจะปรากฏหน้าจอแสดงรายละเอียดของหัวข้อคำถามหรือคำถามนั้นๆขึ้นมาให้ทำการแก้ไขได้ดังนี้

รูปที่ ข-8 หน้าจอการปรับปรุงแก้ไขหัวข้อคำถาม (Forum Update Form)

รูปที่ ข-9 หน้าจอการปรับปรุงแก้ไขคำถาม (Question Update Form)

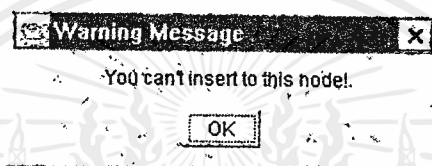
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแก้ไขหัวข้อคำถามหรือคำถามเสร็จแล้ว และผู้ใช้คลิก ปุ่ม OK ระบบจะทำการแก้ไขข้อมูลของหัวข้อคำถามหรือคำถามนั้นๆลงฐานข้อมูลให้โดยอัตโนมัติ แต่ถ้าตอนแก้ไขผู้ใช้ใส่ฟิลด์สำคัญต่างๆไม่ครบถ้วน ระบบก็จะแจ้งข้อความเตือนขึ้นมาเตือนดังนี้



รูปที่ ข-10 หน้าจอแสดงข้อความเตือน (Warning Message)

ถ้าผู้ใช้พยายามที่จะแก้ไขหัวข้อคำถามหรือคำถามของผู้อื่น ระบบก็จะแจ้งข้อความเตือนขึ้นมาเตือนดังนี้



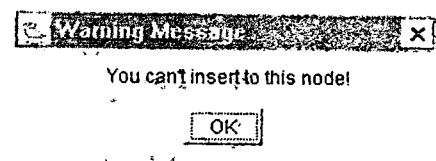
รูปที่ ข-11 หน้าจอแสดงข้อความเตือน (Warning Message)

2. การเพิ่ม (Insert) หัวข้อคำถามหรือคำถาม

อาจารย์สามารถทำการเพิ่มหัวข้อคำถามหรือคำถามใหม่ๆเข้าไปได้โดยการคลิกที่ปุ่ม Insert จากนั้นจะปรากฏเมนูในการเพิ่มมาให้เลือกรูปแบบการเพิ่มซึ่งสอดคล้องกับตำแหน่งที่จะทำการเพิ่มหัวข้อคำถามหรือคำถามดังนี้

- 2.1 ถ้าตำแหน่งที่จะทำการเพิ่มเป็นชื่อของอาจารย์ ผู้ใช้จะสามารถทำการเพิ่มได้ทั้งหัวข้อคำถามและคำถาม
- 2.2 ถ้าตำแหน่งที่จะทำการเพิ่มเป็นหัวข้อคำถาม ผู้ใช้จะสามารถทำการเพิ่มได้ทั้งหัวข้อคำถามและคำถาม
- 2.3 ถ้าตำแหน่งที่จะทำการเพิ่มเป็นคำถาม ผู้ใช้จะสามารถทำการเพิ่มได้แต่เฉพาะคำตอบสำหรับคำถามนั้นๆเท่านั้น

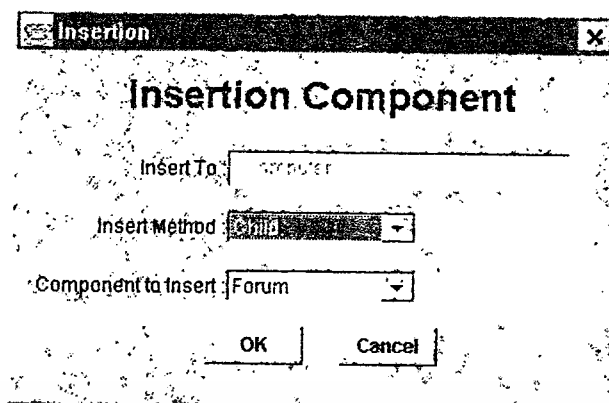
โดยถ้าผู้ใช้พยายามที่จะทำการเพิ่มหัวข้อคำถามหรือคำถามใส่ในของผู้อื่นซึ่งไม่ใช่ของตนเอง ระบบจะแจ้งข้อความเตือนขึ้นมาดังนี้



รูปที่ ข-12 หน้าจอแสดงข้อความเตือน (Warning Message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอของการเพิ่มที่กล่าวข้างต้นมีลักษณะดังนี้

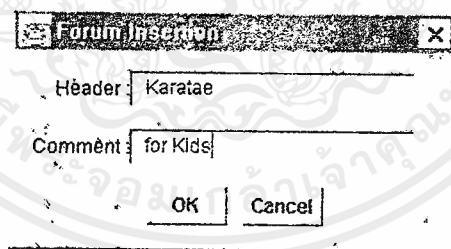


รูปที่ ข-13 หน้าจอเมนูการเพิ่ม (Insert Menu Form)

โดย Insert Method หมายถึง วิธีการในการเพิ่มซึ่งมีอยู่ 2 วิธีคือ

1. Child หมายความว่าคอมโพเนนต์ที่จะทำการเพิ่มจะเป็นลูกของโหนดที่ถูกเลือกอยู่
2. Sibling หมายความว่าคอมโพเนนต์ที่จะทำการเพิ่มจะเป็นโหนดถัดไป (โหนดข้างๆ) ของ no โหนดที่ถูกเลือกอยู่ และ Insert Component หมายถึง สิ่งที่ต้องการเพิ่มเข้าไปเช่น หัวข้อคำถาม (Forum) หรือ คำถาม Question) เป็นต้น

ถ้าเป็นการเพิ่มหัวข้อคำถาม (Forum) หลังจากเลือกจากเมนูและ OK แล้วจะปรากฏแบบฟอร์มสำหรับสร้างหัวข้อคำถามใหม่ดังนี้



รูปที่ ข-14 หน้าจอการเพิ่มหัวข้อคำถาม (Forum Insertion Form)

ถ้าเป็นการเพิ่มคำถามหลังจากเลือกที่เมนูแล้ว OK จะปรากฏแบบฟอร์มสำหรับสร้างคำถามใหม่ดังนี้

รูปที่ ข-15 หน้าจอการเพิ่มคำถาม (Question Insertion Form)

โดยผู้ใช้งานจะต้องใส่ฟิลด์ที่สำคัญๆ ในแบบ Form ให้ครบทุกฟิลด์ก่อนทำการคลิกปุ่ม OK เพื่อทำการเพิ่มเสมอ มิฉะนั้นระบบจะแสดงข้อความเตือนดังนี้

รูปที่ ข-16 หน้าจอแสดงข้อความเตือน (Warning Message)

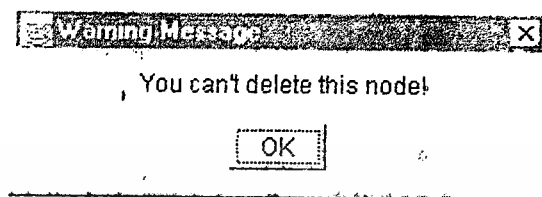
3. การลบ (Delete) หัวข้อคำถาม , คำถาม และคำตอบของคำถาม

อาจารย์สามารถทำการลบหัวข้อคำถาม , คำถาม และคำตอบในคำถามของตนเองได้โดยการคลิกที่ปุ่ม Delete เมื่อต้องการลบหัวข้อคำถาม , คำถาม และคำตอบของคำถามนั้นๆ ึ่งไป จากนั้นระบบจะเตือนให้ผู้ใช้ทราบอีกครั้งว่าต้องการลบหัวข้อคำถามหรือ คำถามนั้นๆ ึ่งจริงหรือไม่ ดังนี้

รูปที่ ข-17 หน้าจอถามความแน่ใจในการลบ

ถ้าผู้ใช้คลิก OK จะเป็นการยืนยันว่าอนุญาตให้ระบบทำการลบหัวข้อคำถาม , คำถาม หรือคำตอบของคำถามนั้นๆทิ้งได้ ระบบก็จะทำการลบหัวข้อคำถาม , คำถาม หรือคำตอบของคำถามนั้นๆทิ้งทันที

เช่นเดียวกับการเพิ่มและการแก้ไขคือถ้าผู้ใช้พยายามที่จะทำการลบหัวข้อคำถาม , คำถาม หรือคำตอบของคำถามของคนอื่นซึ่งไม่ใช่ของตนเอง ระบบจะแสดงข้อความเตือนดังนี้



รูปที่ ข-18 หน้าจอแสดงข้อความเตือน (Warning Message)

4. การขอรายละเอียด (Details) ของหัวข้อคำถามหรือคำถาม

การขอรายละเอียดของคำถามจะต่างจากการเพิ่ม, การลบ, การแก้ไขตรงที่ ไม่ว่าใครก็สามารถที่จะทำการ ขอรายละเอียดของหัวข้อคำถามหรือคำถาม ได้ทั้งนั้น ไม่มีการกำหนดสิทธิ์ในการเข้าดู

การขอรายละเอียดของคำถามสามารถทำได้ 2 วิธีคือ

4.1 ขอรายละเอียดของคำถาม โดยการดับเบิลคลิกที่คำถามนั้นๆ

4.2 ขอรายละเอียดของคำถาม โดยการคลิกที่ปุ่ม Details

ทั้งสองวิธีนี้จะให้ผลที่เหมือนกันคือ ระบบจะทำการเปิดแบบฟอร์มแสดงรายละเอียดของคำถาม ขึ้นมาดังนี้

Question Details

Question Date: 1997-02-17 00:00:00

Question By: Mr.Thana

Header: Do you have some titanic pictures?

Details:

If you have some, please give me for the picture of "Rose"

Reply: 1 Messages

Date: 1998-03-11 00:00:00

Replier: Chai

Replier Email: s7014215@diamond.ce.kmitl.ac.th

Details: Yes, I have some.

New Ans Cancel

รูปที่ ข-19 หน้าจอแสดงรายละเอียดของคำถาม (Question Details Form)

โดยจะเป็นรายละเอียดทั้งหมดของคำถาม (Question) ที่ถูกเลือกนั้นๆ รวมไปถึงการตอบกลับของคำถามนั้นๆ และจำนวนของการตอบกลับมาด้วย

จาก แบบฟอร์มแสดงรายละเอียดนี้ผู้ใช้สามารถทำการเพิ่มคำตอบของคำถามนั้นๆ ได้โดยการคลิกที่ปุ่ม NewAns ก็จะปรากฏแบบฟอร์มสำหรับกรอกคำตอบใหม่นี้ โดยที่ผู้ใช้จะต้องกรอกรายละเอียดให้ครบทุกฟิลด์มีฉะนั้นระบบจะแจ้งข้อความเตือนดังนี้

Warning Message

Please enter 'Replier Name' before do this!

OK

รูปที่ ข-20 หน้าจอแสดงข้อความเตือน (Warning Message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reply Question

New Replyment

Replier Name : Chai

Replier Email : s7014215@diamond.ce.kmitl.ac.th

Answer Details :

I dont like pcp music.

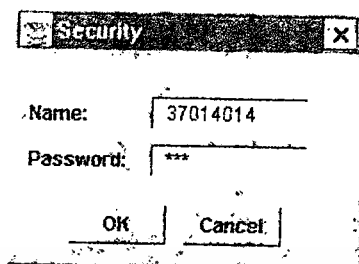
OK Cancel

รูปที่ ข-21 หน้าจอการเพิ่มคำตอบ (Reply Form)

และถ้าผู้ใช้กรอกข้อมูลครบทุกฟิลด์แล้ว เมื่อผู้ใช้ทำการคลิกปุ่ม OK ระบบจะทำการเพิ่มคำตอบนั้นๆ ลงฐานข้อมูลทันที

ระบบการจัดการนัดหมายการประชุม

เริ่มต้นก่อนที่จะเข้าใช้งาน ระบบจะสอบถามถึงสิทธิในการเข้ามาใช้งานของผู้ที่เรียกใช้โปรแกรม



รูปที่ ข-22 แสดงหน้าจอในการสอบถามสิทธิในการใช้งาน

ถ้าผู้ใช้สามารถใส่ข้อมูลที่ต้องการ(ชื่อและรหัสเวิร์ค)ได้ถูกต้อง ผู้ใช้ก็สามารถจะใช้งานโปรแกรมในขั้นต่อไปได้

การควบคุมการรับส่งการนัดหมายการประชุม โดยใช้ทูลบาร์



รูปที่ ข-23 แสดงทูลบาร์หลักของ โปรแกรม

การสร้างการนัดหมายการประชุม (ทูลบาร์ที่ 1)

ใช้ในการสร้างการนัดหมายการประชุมขึ้นมาใหม่ จะได้น้ำจอในรูปที่ 8

การดูรายละเอียดการนัดหมายการประชุม (ทูลบาร์ที่ 2)

ใช้ในการดูรายละเอียดเกี่ยวกับการนัดหมายการประชุมที่เลือกเอาไว้ สามารถทำงานได้ทั้งส่วนที่ทำการรับการนัดหมายการประชุมและส่วนที่ทำการส่งการนัดหมายการประชุมออกไป โดยจะได้หน้าจอ ดังรูปที่ 6-25 และรูปที่ 6-28 ขึ้นอยู่กับว่าใช้งานอยู่ในส่วนใด

การเลือกการนัดหมายการประชุมที่ยังไม่ได้อ่าน (ทูลบาร์ที่ 3)

ใช้ในการเลื่อนแถบสีไปยังการนัดหมายการประชุมที่ผู้ใช้ยังไม่ได้เข้าไปอ่านรายละเอียด

การลบการนัดหมายการประชุม (ทูลบาร์ที่ 4)

ใช้ในการลบการนัดหมายการประชุมที่เลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Mark หมายถึง สถานะที่บอกว่าผู้ใช้ได้ทำการอ่านการนัดหมายการประชุมนั้นๆไปแล้วหรือยัง ถ้าเป็นรูปปุมสีเขียวแสดงถึงว่าผู้ใช้ยังไม่ได้อ่านการนัดหมายการประชุมนั้นๆ แต่ถ้าเป็นรูปจุดแสดงถึงว่าผู้ใช้ได้ทำการอ่านการนัดหมายการประชุมนั้นๆไปแล้ว (การอ่านการนัดหมายการประชุมทำได้โดยดับเบิลคลิกการนัดหมายการประชุมซึ่งอยู่ในช่วงคอลัมน์นี้ หรือโดยการคลิกเมาส์ที่ปุ่ม Get Msg)

3. Sender หมายถึง ผู้ที่ทำการส่งการนัดหมายการประชุมมาถึงผู้ใช้แสดงในรูปของรหัสประจำตัว / ชื่อ-นามสกุล

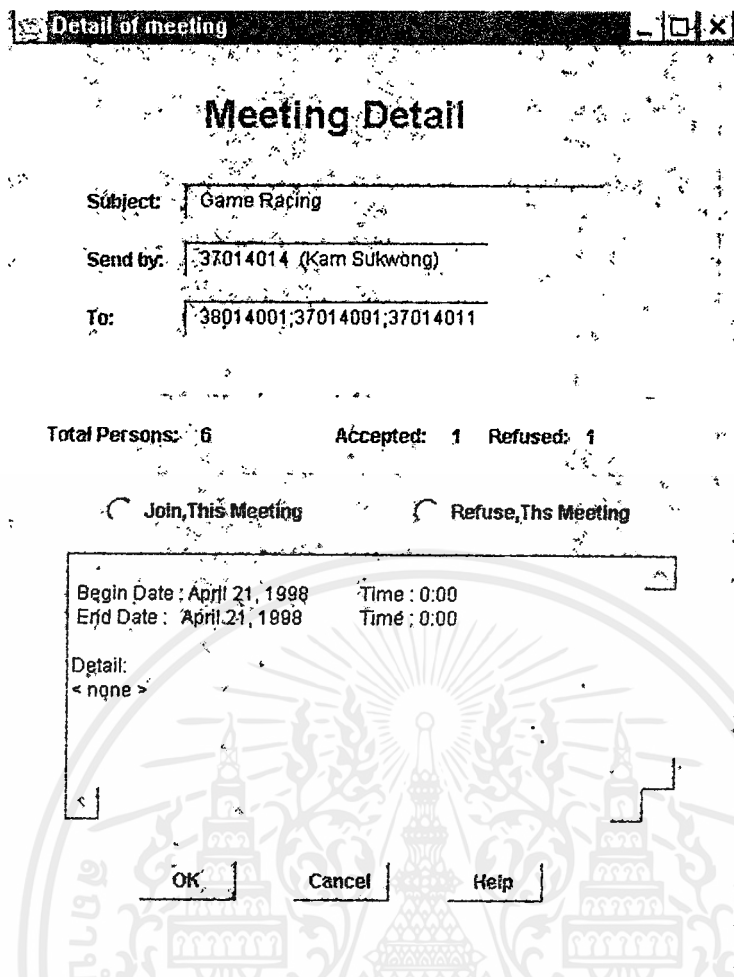
4. Date หมายถึง วัน-เวลาที่ได้รับการนัดหมายการประชุม

5. Flag หมายถึง ความสำคัญของการนัดหมายการประชุมที่ผู้ใช้เป็นผู้กำหนดขึ้น โดยที่ถ้าเป็นรูปธงหมายถึงการนัดหมายการประชุมนั้นสำคัญ แต่ถ้าเป็นรูปจุดหมายถึงผู้ใช้ไม่ได้ให้ความสำคัญ (การกำหนดความสำคัญทำได้โดยดับเบิลคลิกหัวข้อการนัดหมายการประชุมซึ่งอยู่ในช่วงคอลัมน์นี้)

6. Status หมายถึง การนัดหมายการประชุมนั้นจะมีขึ้นเมื่อไร เป็นไปได้ทั้งอดีตและอนาคต

7. Categories หมายถึง ประเภทของการนัดหมายการประชุม เช่น Job, Homework, Holiday ฯลฯ

ถ้าผู้ใช้ต้องการเข้าไปดูรายละเอียดของการนัดหมายการประชุมที่ได้รับ หรือต้องการตัดสินใจที่จะเข้าร่วมหรือปฏิเสธการประชุมก็สามารถทำได้โดยดับเบิลคลิกที่หัวข้อการนัดหมายการประชุมนั้นหรือโดยทำการคลิกที่ปุ่ม Get Msg ที่อยู่ด้านบนก็ได้ จากนั้น โปรแกรมก็จะแสดงหน้าจอใหม่ดังรูปด้านล่าง



รูปที่ ข-25 แสดงรายละเอียดของการนัดหมายการประชุม

จากรูปด้านบนจะเห็นได้ว่าผู้ใช้สามารถตัดสินใจเลือกที่จะเข้าร่วมหรือปฏิเสธก็ได้เป็นอย่างดีหนึ่งเท่านั้น หลังจากที่ตัดสินใจไปแล้วผลการตัดสินใจจะถูกส่งไปอ็อปเซตยังฐานข้อมูล ซึ่งผู้ที่ทำการนัดหมายก็จะสามารถเห็นผลการตัดสินใจของผู้ใช้ได้

ส่วนที่ทำหน้าที่ในการแสดงการนัดหมายการประชุมที่ผู้ใช้ได้ทำการส่งออกไป

ผู้ใช้จะสามารถเข้ามาดูการนัดหมายการประชุมที่ตนได้ส่งออกไปได้โดยการเปลี่ยนค่าที่อยู่ใน Check Box โดยทำการเซตค่าเป็น Send Box ซึ่งก็จะแสดงผลดังรูปด้านล่างนี้

Applet Viewer: Real.class

Applet

New Msg Get Msg Next Delete Reload Dismiss

Send Box

Subject	Date	Total Receivers	Accept Receivers	Refuse Receivers	Status
Play tennis	1998-04-04 (14:50:04)	3	0	0	Passed 1 months ago
play pingpong	1998-04-04 (14:56:14)	6	0	0	Passed 1 months ago
test	1998-04-04 (16:05:21)	1	0	0	Passed 1 months ago
Go to Central	1998-04-23 (06:11:07)	1	0	0	Passed 1 days ago

Applet started.

รูปที่ ข-26 แสดงการนัดหมายการประชุมทั้งหมดที่ผู้ใช้ได้รับมา

รูปแบบการแสดงถึงการนัดหมายการประชุมที่ผู้ใช้ได้ส่งออกไปจะประกอบด้วยคอลัมน์ต่างๆดังนี้

1. Subject หมายถึง ชื่อของการประชุมที่ผู้ใช้ได้ส่งไป ส่วนรูปที่อยู่ด้านหน้าของชื่อการประชุมจะแสดงถึงสถานะของการประชุม โดยมี 2 ชนิดคือ ถ้าเป็นรูปกระดานนั้นหมายถึงการประชุมนั้นอยู่ในสถานะปกติ แต่ถ้าเป็นรูปกากบาทสีแดงแสดงถึงว่าผู้ใช้ได้ทำการยกเลิกการนัดหมายการประชุมนั้นไปแล้ว

2. Date หมายถึง วัน-เวลาที่ได้ส่งการนัดหมายการประชุมออกไป
3. Total Receivers หมายถึง จำนวนผู้ที่ได้รับการนัดหมายการประชุมที่ส่งไปทั้งหมด
4. Accept Receivers หมายถึง จำนวนผู้ที่ได้ตอบตกลงการนัดหมายการประชุมที่ส่งไปทั้งหมด
5. Refuse Receivers หมายถึง จำนวนผู้ที่ได้ตอบปฏิเสธการนัดหมายการประชุมที่ส่งไปทั้งหมด
6. Status หมายถึง การประชุมนั้นมีขึ้นเมื่อไรเมื่อเทียบกับเวลาปัจจุบัน

ถ้าผู้ใช้ต้องการดูผลการตัดสินใจของผู้ที่จะเข้าร่วมการประชุมก็สามารถทำได้โดยการดับเบิลคลิกที่หัวข้อการนัดหมายการประชุม ซึ่งก็จะปรากฏหน้าจอใหม่ดังรูปด้านล่าง

Receiver	Read	Accept	Refuse	Last login
38014001 (Vorravit Rimpapun)	<input type="checkbox"/>		No reply	
37014001 (Kanokporn Selbeal)	<input checked="" type="checkbox"/>	No reply	No reply	
37014011 (Korkit Verachakul)	<input checked="" type="checkbox"/>	No reply	No reply	
37014014 (Kam Sukwong)	<input checked="" type="checkbox"/>	No reply	No reply	
37014215 (Boonchal Rattawaranur)	<input type="checkbox"/>	No reply		
37014216 (Boonchal Eissaragkul)	<input checked="" type="checkbox"/>	No reply	No reply	

OK Help

รูปที่ ข-27 แสดงผลของการตัดสินใจของแต่ละบุคคล

รูปแบบการแสดงผลการตัดสินใจของแต่ละบุคคลที่ได้รับการนัดหมายการประชุมที่ผู้ใช้ได้ส่งออกไปจะประกอบด้วยคอลัมน์ต่างๆดังนี้

1. Receiver หมายถึง รหัสประจำตัว และชื่อนามสกุลของผู้ที่ได้รับการนัดหมายการประชุม
2. Read หมายถึง ผู้ที่ได้รับการนัดหมายการประชุมได้เข้ามาอ่านรายละเอียดของการนัดหมายการประชุมนั้นแล้วหรือยัง
3. Accept หมายถึง ผู้ที่ได้รับการนัดหมายการประชุมนั้นตอบตกลงว่าจะเข้าร่วมการประชุมหรือไม่ โดยที่ถ้าเป็นเครื่องหมายถูกสีเขียวก็แสดงว่าจะเข้าร่วม แต่ถ้าเป็น No Reply ก็แสดงว่าไม่ได้ตอบ
4. Refuse หมายถึง ผู้ที่ได้รับการนัดหมายการประชุมนั้นตอบว่าจะปฏิเสธการประชุมหรือไม่ โดยที่ถ้าเป็นเครื่องหมายถูกสีแดงก็แสดงว่าปฏิเสธ แต่ถ้าเป็น No Reply ก็แสดงว่าไม่ได้ตอบ
5. Last Login หมายถึง เวลาครั้งล่าสุดที่ผู้ที่ได้รับการนัดหมายการประชุมได้เข้ามาในระบบ ในส่วนนี้จะประโยชน์แก่ผู้ใช้อย่างมากเพราะสามารถคาดการณ์ได้ว่าเป็นไปได้มากน้อยแค่ไหนที่ผู้ได้รับการนัดหมายการประชุมจะตัดสินใจกับการนัดหมายการประชุมที่ได้ส่งไปให้

ถ้าผู้ใช้ต้องการดูรายละเอียดการนัดหมายการประชุมก็สามารถทำได้โดยการคลิกที่ปุ่ม Get Msg ที่อยู่ด้านบนก็จะปรากฏหน้าจอใหม่ดังรูปด้านล่าง

Detail of meeting

Meeting Detail

Subject: Game Racing

Send by: 37014014, (Kam Sukwong)

To: 38014004;37014001;37014011

Total Persons: 6 Accepted: 1 Refused: 1

Begin Date: April 21, 1998 Time: 0:00
 End Date: April 21, 1998 Time: 0:00

Detail: < none >

OK Cancel Help

รูปที่ ข-28 แสดงรายละเอียดการนัดหมายการประชุมที่ส่งออกไป

ส่วนที่ทำหน้าที่สร้างการนัดหมายการประชุม

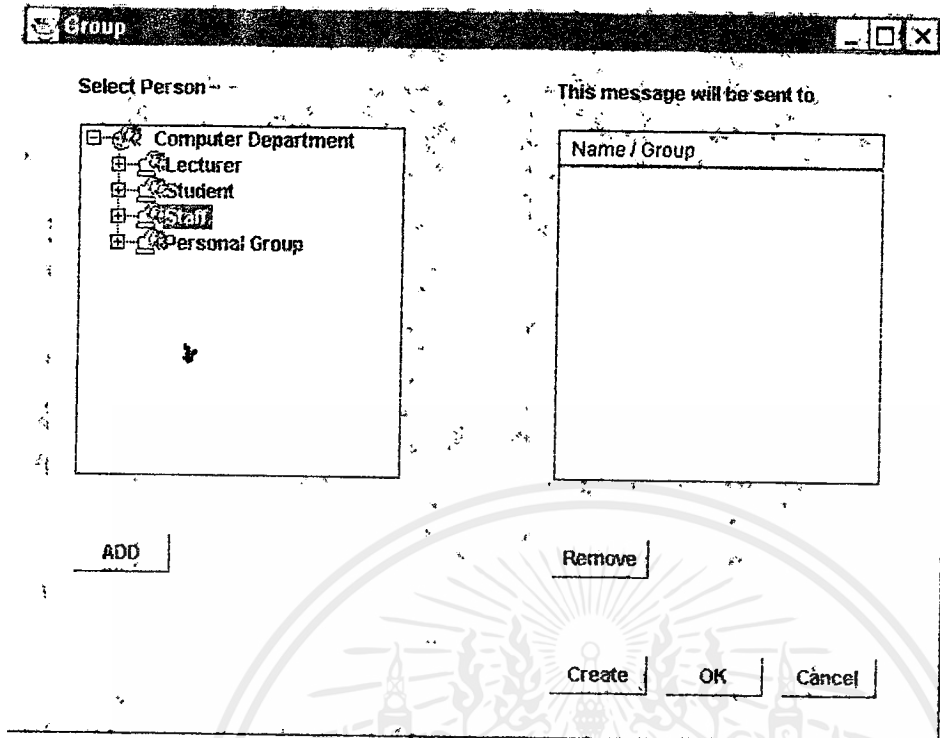
เมื่อผู้ใช้ต้องการจะทำการสร้างการนัดหมายการประชุมขึ้นมา ก็สามารถทำได้โดยการคลิกที่ปุ่ม New Msg ที่อยู่ด้านบนซึ่งก็จะทำการแสดงหน้าจอใหม่ดังรูปด้านล่าง

รูปที่ ข-29 แสดงรูปแบบในการสร้างการนัดหมายการประชุมขึ้นมาใหม่

จากรูปด้านบนจะแบ่งการทำงานออกเป็น 2 ส่วนก็คือ

- Meeting
- Summary

1. Meeting จะใช้สำหรับการส่งการนัดหมายการประชุมใหม่โดยจะประกอบด้วยส่วนที่ทำหน้าที่ในการรับข้อมูลจากผู้ใช้ซึ่งจะเป็นข้อมูลที่เกี่ยวข้องกับการนัดหมายการประชุมใหม่ที่จะถูกส่งออกไป ผู้ใช้จะต้องทำการกรอกรายละเอียดต่างๆ ให้ครบการนัดหมายการประชุมจึงจะสามารถถูกส่งออกไปได้
2. Summary จะใช้สำหรับการส่งสรุปผลการประชุม, สรุปการแบ่งงาน ไปยังผู้ที่เข้าร่วมประชุมที่ผู้ใช้ต้องการ ในการระบุที่อยู่ของผู้รับผู้ใช้จะทำการป้อนเข้าไปเอง หรือจะทำการเลือกจากโครงสร้างแบบต้นไม้ที่มีอยู่แล้วก็ได้ ถ้าผู้ใช้ต้องการจะเลือกจาก โครงสร้างต้นไม้ ก็ไม่สามารถทำได้โดยการคลิกที่ปุ่ม To ซึ่งก็จะปรากฏหน้าต่างใหม่ดังรูปด้านล่าง



รูปที่ ข-30 แสดง โครงสร้างต้นไม้ที่เก็บรายชื่อ

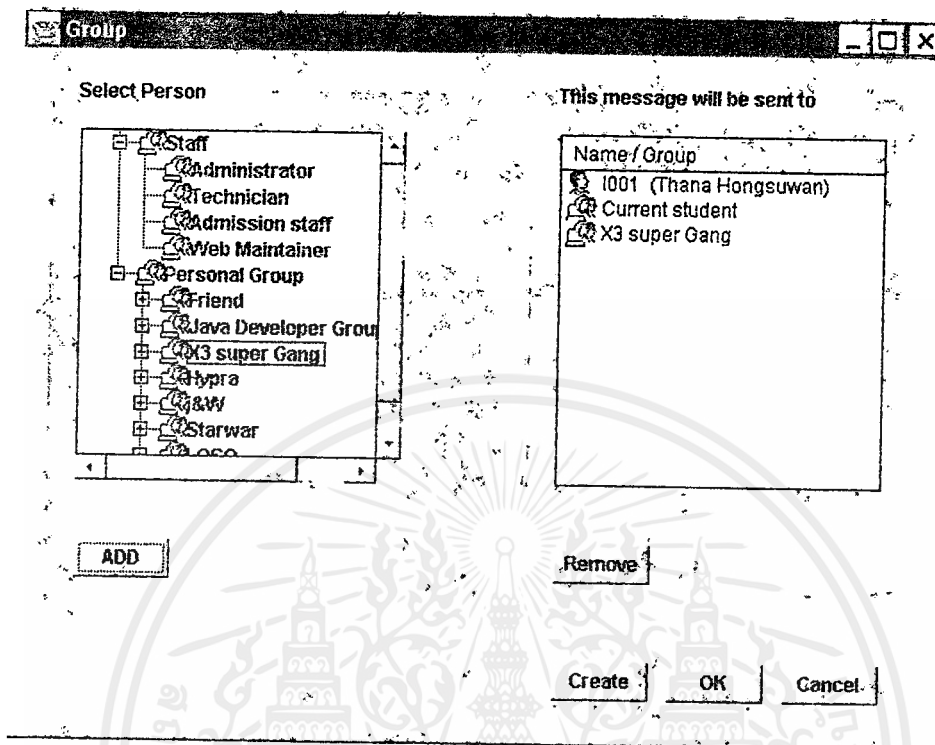
จากรูปด้านบนจะเห็น โครงสร้างต้นไม้ที่ใช้ในการเก็บชื่อ-รหัสประจำตัวของผู้ที่มีส่วนเกี่ยวข้องกับภาค วิชาคอมพิวเตอร์ซึ่งสามารถแบ่งประเภทออกได้ดังนี้

1. Lecturer คือ อาจารย์ภายในภาควิชาคอมพิวเตอร์
2. Student
 - 2.1 Current Student คือ นักศึกษาปัจจุบันของภาควิชาคอมพิวเตอร์
 - 2.2 Graduated Student คือ ศิษย์เก่าของภาควิชาคอมพิวเตอร์
3. Staff
 - 3.1 Administrator คือ เจ้าหน้าที่หรือนักศึกษาที่รับผิดชอบ-ดูแลระบบคอมพิวเตอร์ภายในภาควิชาคอมพิวเตอร์
 - 3.2 Technician คือ เจ้าหน้าที่หรือนักศึกษาที่รับผิดชอบ-ดูแลเกี่ยวกับอุปกรณ์ต่างๆภายในภาควิชาคอมพิวเตอร์
 - 3.3 Admission Staff คือ เจ้าหน้าที่ธุรการของภาควิชาคอมพิวเตอร์
 - 3.4 Web Maintainer คือ เจ้าหน้าที่หรือนักศึกษาที่รับผิดชอบ-ดูแลโครงการนี้
4. Personal Group คือ กลุ่มของสมาชิกที่ผู้ใช้เป็นคนกำหนดขึ้นมาเอง กลุ่มที่อยู่ภายใน

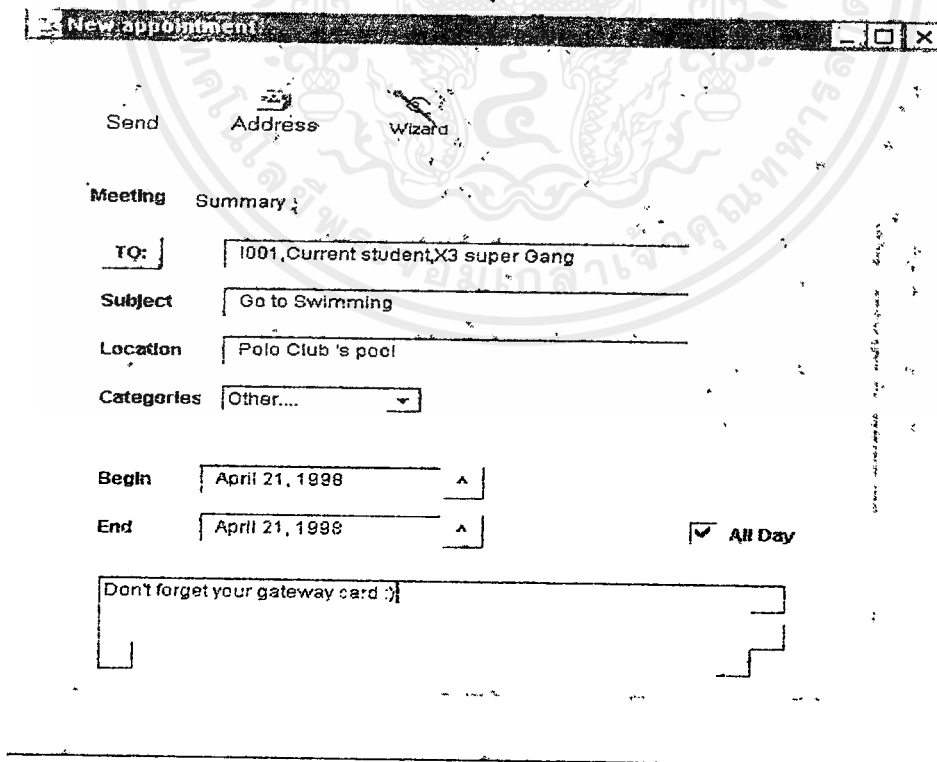
Personal Group นี้ผู้ที่เป็นคนกำหนดขึ้นมาเท่านั้นที่จะดำเนินการต่างได้ เพราะบุคคลอื่นจะไม่สามารถเห็นกลุ่มที่สร้างขึ้นมาได้

เมื่อผู้ใช้ต้องการเลือกสมาชิกจากโครงสร้างต้นไม้ก็สามารถทำได้โดยการคลิกที่ชื่อของบุคคลที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อผู้ดูแลเห็นรายชื่อเอกสารด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการจากนั้น ให้ทำการคลิกที่ปุ่ม Add ชื่อของบุคคลดังกล่าวก็จะไปปรากฏอยู่ทางกรอบด้านขวามือ เมื่อผู้ใช้เลือกบุคคลที่ต้องการครบแล้วก็ให้คลิกที่ปุ่ม OK ดังจะแสดงรูปด้านล่าง



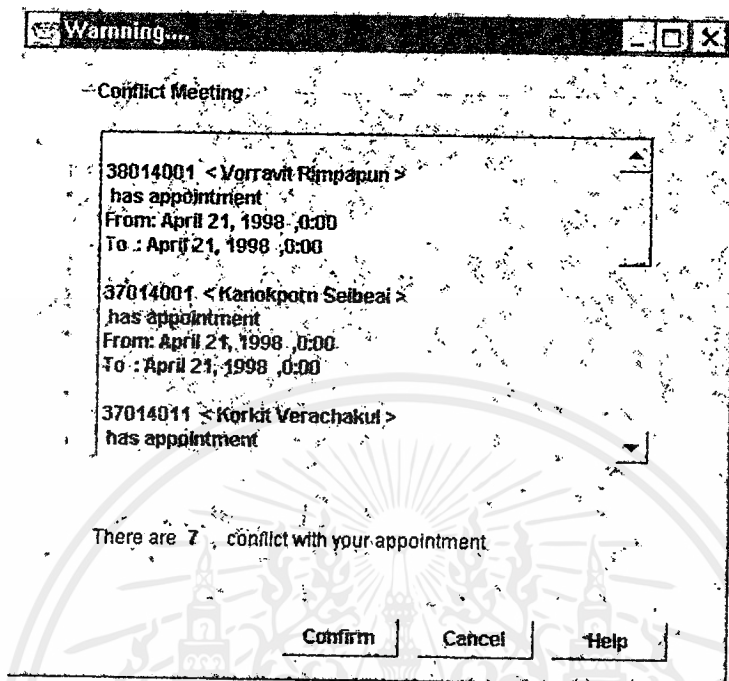
รูปที่ ข-31 แสดงการเลือกบุคคลที่ต้องการ



รูปที่ ข-32 แสดงผลจากการคลิกที่ปุ่ม OK ในรูป ข-30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้กรอกข้อมูลทุกอย่างเรียบร้อยแล้วก็สามารถส่งการนัดหมายต่างๆ ได้โดยการคลิกที่ปุ่ม Send แต่ถ้าเกิดการซ้ำซ้อนในเรื่องของเวลาของผู้ที่จะเข้าร่วมการประชุมก็จะมีแจ้งเตือนดังรูปด้านล่าง

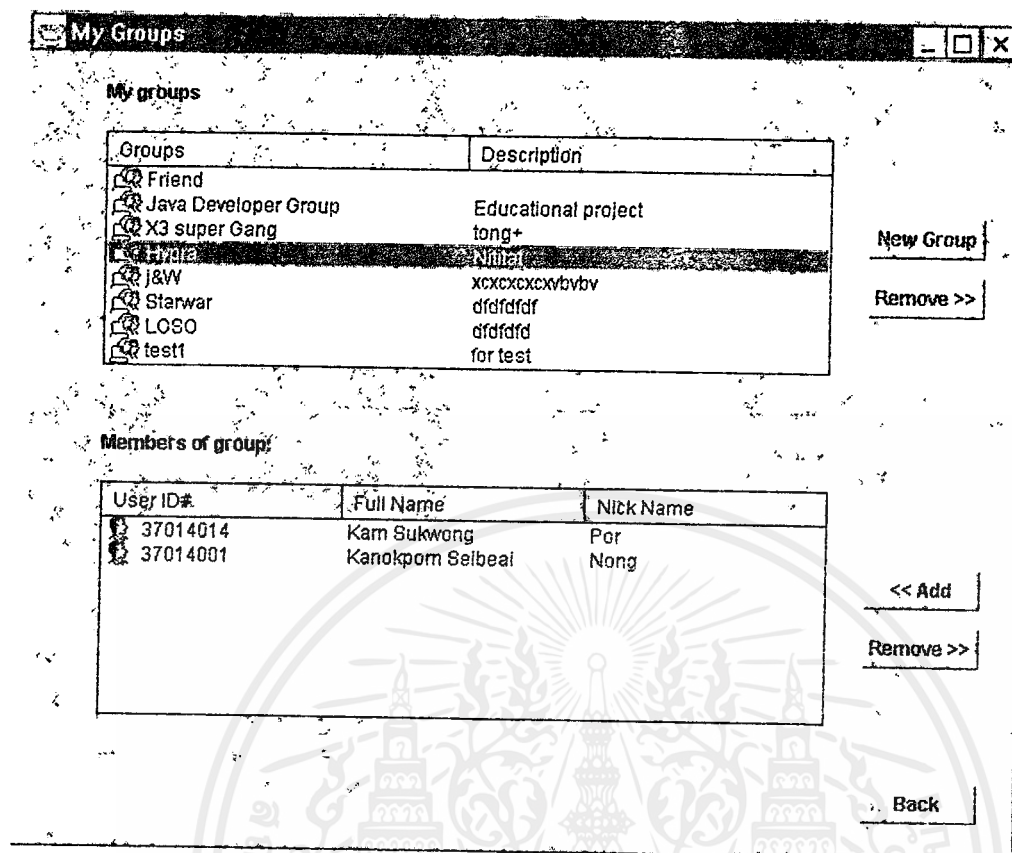


รูปที่ ข-33 แสดงหน้าจอเตือนเมื่อมีการซ้ำซ้อนในเรื่องของเวลา

จากรูปด้านบนผู้ใช้จะมีทางเลือกในการตัดสินใจอยู่ 2 ทางคือ ยืนยันการนัดหมายการประชุม กับ ยกเลิกการส่งการนัดหมายการประชุมออกไป ถ้าผู้ใช้ตัดสินใจยกเลิกการส่งการนัดหมายการประชุมก็จะกลับไปยังรูปที่ ข-31

ส่วนที่ทำหน้าที่ในการสร้างกลุ่มของผู้ใช้ขึ้นมา

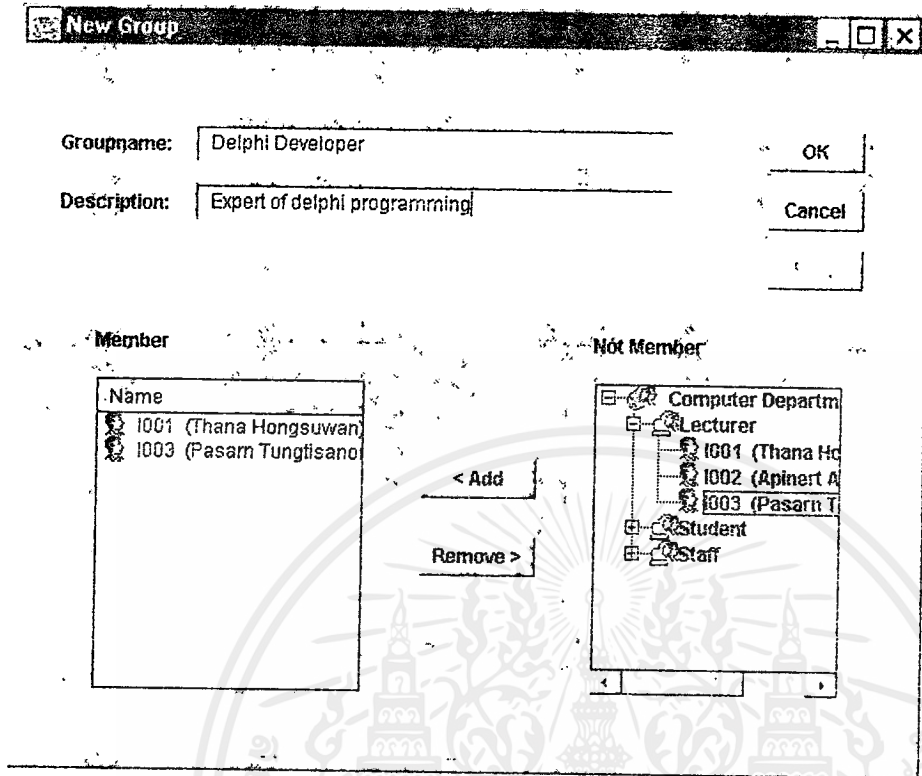
ผู้ใช้สามารถที่จะจัดตั้งกลุ่มของตนเองขึ้นมาใหม่ได้ โดยการกำหนดสมาชิกให้กับกลุ่มนั้นๆ โดยกลุ่มที่จัดตั้งขึ้นจะมองเห็นได้เฉพาะจากผู้ใช้ที่เป็นคนตั้งกลุ่มนั้นขึ้นมาเท่านั้น ผู้ใช้สามารถจัดตั้งกลุ่มใหม่ขึ้นมาได้โดยการคลิกที่ปุ่ม Create จากหน้าจอในรูปที่ ข-31 ก็จะปรากฏหน้าจอใหม่ดังรูปด้านล่าง



รูปที่ ข-34 แสดงหน้าจอกลุ่มที่ผู้ใช้ได้ตั้งไว้แล้ว

จากรูปด้านบนจะประกอบด้วย 2 ส่วน คือ ส่วนแรก (ด้านบน) จะเป็นรายชื่อของกลุ่มที่ผู้ใช้เป็นคนจัดตั้งขึ้นมา ส่วนที่สอง (ด้านล่าง) จะเป็นรายชื่อของสมาชิกในกลุ่มที่ผู้ใช้ทำการเลือก (โดยการคลิกที่ชื่อกลุ่มในส่วนแรก) ถ้าสังเกตให้ดีจะพบว่าจะมีปุ่ม Remove อยู่ 2 ปุ่ม ปุ่ม Remove ด้านบนจะทำการลบชื่อกลุ่มที่ผู้ใช้เลือกไว้และสมาชิกทั้งหมดในกลุ่มนั้นออกไป ส่วนปุ่ม Remove ที่อยู่ด้านล่างจะทำการลบสมาชิกที่ผู้ใช้เลือกออกไป (ผู้ใช้สามารถลบสมาชิกได้พร้อมกันหลายคน โดยการคลิกที่รายชื่อสมาชิกได้มากกว่า 1 คน)

เมื่อผู้ใช้พร้อมที่สร้างกลุ่มใหม่แล้ว ก็สามารถทำได้โดยการคลิกที่ปุ่ม New Group ก็จะปรากฏหน้าจอในด้านล่าง



รูปที่ ข-35 แสดงหน้าจอการจัดตั้งกลุ่มขึ้นมาใหม่

จากรูปด้านบนจะเป็นรูปแบบฟอร์มในการตั้งกลุ่มขึ้นมาใหม่ ผู้ใช้จะต้องทำการตั้งชื่อกลุ่มที่จัดตั้งขึ้นมาใหม่ โดยชื่อที่ตั้งขึ้นมาแล้วจะต้องไม่เหมือนกับชื่อกลุ่มที่ผู้ใช้เคยตั้งไว้แล้ว มิฉะนั้นแล้ว โปรแกรมจะไม่นอนุญาติให้ตั้งกลุ่มนั้นได้ หลังจากที่ผู้ใช้ทำการตั้งชื่อกลุ่มแล้วผู้ใช้ต้องทำการเลือกสมาชิกในกลุ่ม โดยเลือกจากโครงสร้างต้นไม้ที่อยู่ด้านขวามือ เมื่อใส่ข้อมูลที่ต้องการจนหมดแล้วผู้ใช้ก็จะทำการคลิกปุ่ม OK เพื่อจัดตั้งกลุ่มที่ต้องการ

6.2.4 การเปลี่ยนพาสเวิร์ดของผู้ใช้

การเปลี่ยนพาสเวิร์ดมีหน้าจอหลักดังนี้คือ

รูปที่ ข-36 หน้าจอในการเปลี่ยนพาสเวิร์ด

เมื่อผู้ใช้ใส่ฟิลด์ต่างๆครบแล้วและคลิกปุ่ม OK ก็จะปรากฏหน้าจอเพื่อให้ผู้ใช้ยืนยัน Password ใหม่ดังนี้

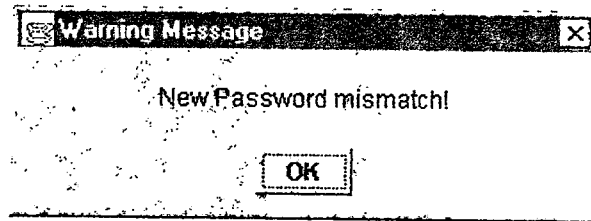
รูปที่ ข-37 หน้าจอยืนยันพาสเวิร์ดใหม่

แต่ถ้าผู้ใช้ใส่ฟิลด์ต่างๆไม่ครบถ้วนแล้วคลิกปุ่ม OK ระบบก็จะแจ้งข้อความเตือนดังนี้

รูปที่ ข-38 หน้าจอแสดงข้อความเตือน (Warning Message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้นั้น พาสเวิร์ดใหม่ถูกต้องแล้วระบบก็จะทำการอัปเดตพาสเวิร์ดในฐานข้อมูล (Database) ให้ แต่ถ้าพาสเวิร์ดใหม่ที่ใช้นั้น ไม่ถูกต้องระบบก็จะแสดงข้อความเตือนขึ้นมาดังนี้



รูปที่ ข-39 หน้าจอแสดงข้อความเตือน (Warning Message)





- [1] David M.Geary(1997): “graphic JAVA Mastering the AWT”,The Sunsoft Press Java Series,Sun Microsystem, California,(1997)
- [2] “Java Unleashed”,Sams.net Publishing,Indianapolis,(1996)
- [3] Jepson Brian.(1997) : “JAVA Database Programming”, John Wiley & Sons, Canada,(1997)
- [4] Taylor, Art. (1997) : “ JDBC developer ‘s resource: database programming on the Internet”, Informix Press, CA,1997.
- [5] โชคชัย เศรษฐพรรุ่ง(2541): “ เรียนรู้ Java ด้วย Visual J++ “ , ซีเอ็ดยูเคชั่น จำกัด (มหาชน),กรุงเทพฯ, (2541)
- [6] ณัฐวัฒน์ ลิ้มล้อมวงศ์(2540): “ สร้างสรรค์เว็บเพจสวยๆด้วยภาษา JAVA “ , ซีเอ็ดยูเคชั่น จำกัด (มหาชน),กรุงเทพฯ,(2540)

