

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนาระบบค่าล่วงเวลาด้วยหลักการเชิงวัตถุ

The Development of Overtime System Using Object-Oriented Paradigm



H002465



วัน เดือน ปี.....	22 ก.พ. 2550
เลขทะเบียน.....	02465
เลขเรียกหนังสือ.....	ดท. ๕๖๒๑๓ ๒๕๕๐
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการกรณีศึกษาพิเศษ
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2540
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาระบบค่าล่วงเวลาด้วยหลักการเชิงวัตถุ
นักศึกษา	น.ส.สุนีย์ อริยเจริญทรัพย์
อาจารย์ที่ปรึกษา	ดร. อมฤต เหล่ารักพงษ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	การจัดการเทคโนโลยีสารสนเทศ
พ.ศ.	2540

บทคัดย่อ

การพัฒนาระบบค่าล่วงเวลาด้วยหลักการเชิงวัตถุ(Object-Oriented) มีวัตถุประสงค์เพื่อศึกษาการพัฒนาระบบด้วยหลักการในเชิงวัตถุ ซึ่งมีปัจจัยสำคัญที่ต้องคำนึงถึงในการพัฒนาคือ Encapsulation, Inheritance, Polymorphism, Reuse เป็นต้น

ผลการศึกษาพบว่า การนำเอาหลักการเชิงวัตถุ มาใช้ในการพัฒนาระบบงาน สามารถแก้ปัญหาเรื่องการบำรุงรักษา(Maintenance) ซึ่งเป็นปัญหาที่รุนแรงและสิ้นเปลืองค่าใช้จ่าย เพราะสามารถปรับปรุงแก้ไขได้ง่าย และผู้อื่นสามารถทำความเข้าใจกับโปรแกรมของเดิมได้ ตลอดจนสามารถนำส่วนประกอบที่ได้พัฒนาไปแล้ว กลับมาใช้งานใหม่ในการพัฒนาระบบอื่น ๆ ได้อย่างมีประสิทธิภาพ

Title The Development of Overtime System Using Object-Oriented Paradigm
Student Miss Sunee Ariyacharoensup
Advisor Dr. Amarit Laorakpong
Level of Study Master of Science in Information Technology
Major Information Technology Management
Year 1997

ABSTRACT

Purpose of the development of overtime system using object-oriented paradigm is study using about the object-oriented paradigm technique. Main factor for development is Encapsulation, Inheritance, Polymorphism, Reuse etc..

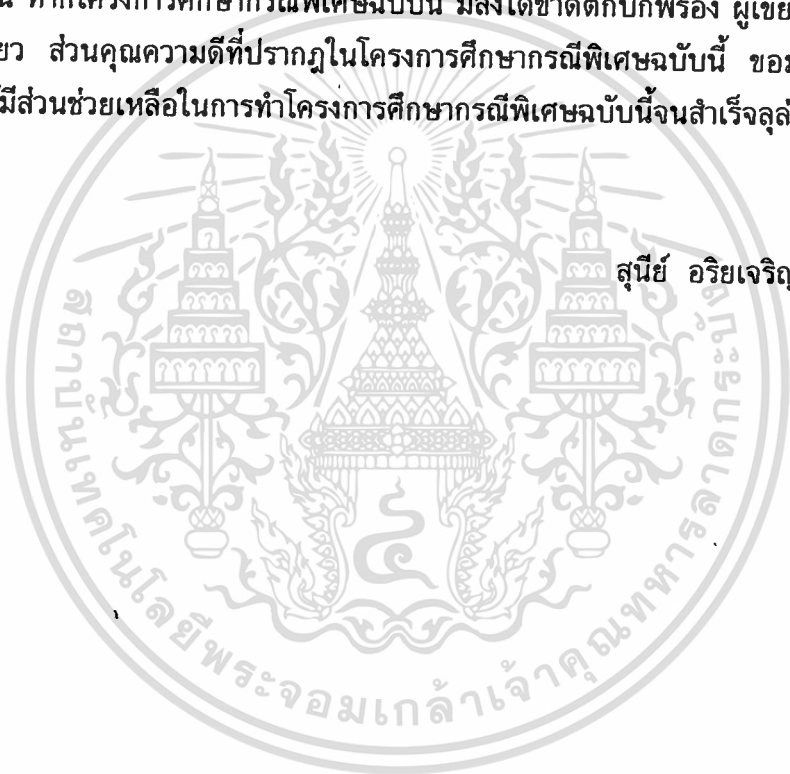
The result can be used to correct problem about maintenance, that is critical and many cost , because it can easily correction and everyone understand this program. And it can be used to work in developing other working system for more efficiency.

กิตติกรรมประกาศ

ความสำเร็จของโครงการศึกษาระดับปริญญาโท สาขาวิชาศึกษาศาสตร์ ผู้เขียนได้รับความกรุณาและความช่วยเหลือจากบุคคลหลายท่าน โดยเฉพาะอย่างยิ่ง ดร.อมฤต เหล่ารักพงษ์ อาจารย์ที่ปรึกษาโครงการศึกษาระดับปริญญาโท ซึ่งกรุณาสละเวลาอันมีค่ายิ่งในการให้แนวทางและคำปรึกษา ตลอดระยะเวลาในการทำโครงการศึกษาระดับปริญญาโท

ผู้เขียนขอขอบคุณ ผู้ปฏิบัติงานการไฟฟ้าฝ่ายผลิตแห่งประเทศไทย ทุกท่าน ที่ได้ให้คำปรึกษาเทคนิคทางด้านการออกแบบและเขียนโปรแกรม ตลอดจนให้กำลังใจด้วยดีมาตลอด ท้ายนี้ หากโครงการศึกษาระดับปริญญาโท มีสิ่งใดขาดตกบกพร่อง ผู้เขียนขอน้อมรับไว้แต่เพียงผู้เดียว ส่วนคุณความดีที่ปรากฏในโครงการศึกษาระดับปริญญาโท ขอมอบให้เป็นคุณความดีของผู้มีส่วนช่วยเหลือในการทำโครงการศึกษาระดับปริญญาโทจนสำเร็จลุล่วงได้ด้วยดี

สุนีย์ อริยเจริญทรัพย์



สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	1
บทคัดย่อภาษาอังกฤษ.....	2
กิตติกรรมประกาศ.....	3
สารบัญ.....	4
สารบัญภาพ.....	5
สารบัญตาราง.....	7
บทที่ 1. บทนำ	8
1.1 ความเป็นมา	8
1.2 วัตถุประสงค์	8
1.3 ขั้นตอนการศึกษา	9
1.4 ประโยชน์ที่คาดว่าจะได้รับ	9
2. วรรณกรรมและทฤษฎีที่เกี่ยวข้อง	10
2.1 แนวคิดพื้นฐานเกี่ยวกับ Object.....	10
2.2 แนวคิดและเทคนิคของ Object-Orientation.....	11
2.3 แนวคิดเกี่ยวกับเทคโนโลยี.....	14
3. วิธีดำเนินการศึกษา	19
3.1 การศึกษาและวิเคราะห์ข้อมูลค่าล่วงเวลา.....	19
3.2 การวิเคราะห์และออกแบบระบบ.....	21
3.3 การพัฒนาโปรแกรม.....	52
4. บทสรุปและข้อเสนอแนะ	74
บรรณานุกรม	77
ภาคผนวก	
ภาคผนวก ก Object Model ของ Coad Notation.....	78
ภาคผนวก ข Data Description.....	80
ภาคผนวก ค Object ที่นำมาใช้ในการพัฒนาระบบ.....	84
ภาคผนวก ง คู่มือการใช้งานระบบค่าล่วงเวลา.....	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 3.1 ส่วนประกอบของ Object Model และส่วนประกอบอื่น ๆ ที่เกี่ยวข้อง.....	23
รูปที่ 3.2 Class และ Object ของผู้กระทำ.....	24
รูปที่ 3.3 สรุปรูป PD และ NT ของผู้กระทำและผู้มีส่วนร่วม.....	24
รูปที่ 3.4 สรุปรูป Object และ PD ของสถานที่	25
รูปที่ 3.5 สรุปรูป Object และ PD ของสิ่งของ.....	25
รูปที่ 3.6 แสดงความสัมพันธ์ระหว่างผู้กระทำ กับ Transaction	26
รูปที่ 3.7 แสดงความสัมพันธ์ระหว่างสถานที่ กับ Transaction	26
รูปที่ 3.8 แสดงความสัมพันธ์ระหว่างTransactionกับ Transaction	26
รูปที่ 3.9 แสดงภาพรวม Object และความสัมพันธ์ระหว่างกัน	27
รูปที่ 3.10 แสดง Attribute และความสัมพันธ์ของ Object ผู้กระทำ	28
รูปที่ 3.11 แสดง Attribute และความสัมพันธ์ของ Object สถานที่.....	29
รูปที่ 3.12 แสดง Attribute และความสัมพันธ์ของ Object ชื่อ Holiday Schedule.....	29
รูปที่ 3.13 แสดง Attribute และความสัมพันธ์ของ Object ชื่อ Working Schedule.....	30
รูปที่ 3.14 แสดง Attribute และความสัมพันธ์ของ Object ชื่อ Rate Category.....	30
รูปที่ 3.15 แสดง Attribute และความสัมพันธ์ของ Object ชื่อ OT Transaction.....	31
รูปที่ 3.16 แสดง Attribute และความสัมพันธ์ของ Object ชื่อ Monthly OT.....	32
รูปที่ 3.17 แสดง Attribute และความสัมพันธ์ของ Object ทั้งหมด ในระบบ.....	33
รูปที่ 3.18 แสดงการวิเคราะห์การทำงานของ PD Object ในระบบ.....	34
รูปที่ 3.19 แสดงโครงสร้าง HI ของระบบค่าล่วงเวลา.....	38
รูปที่ 3.20 แสดง FormEmployee Object.....	39
รูปที่ 3.21 แสดง FormOfficer Object.....	39
รูปที่ 3.22 แสดง FormWorkGroup Object.....	40
รูปที่ 3.23 แสดง FormRate Object.....	40
รูปที่ 3.24 แสดง FormHolEmp Object.....	41
รูปที่ 3.25 แสดง FormHolOffi Object.....	41
รูปที่ 3.26 แสดง FormDept Object.....	42
รูปที่ 3.27 แสดง Logon Window Object.....	42
รูปที่ 3.28 แสดง FormClerk Object.....	43
รูปที่ 3.29 แสดง FormOtTrans Object.....	43
รูปที่ 3.30 แสดง FormNewMonth Object.....	44
รูปที่ 3.31 แสดง FormPrintTrans Object.....	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ

	หน้า
รูปที่ 3.32 แสดง FormRptDailyOt Object.....	45
รูปที่ 3.33 แสดง FormRptMonthlyOt Object.....	45
รูปที่ 3.34 แสดงการวิเคราะห์การเข้าระบบ.....	46
รูปที่ 3.35 แสดงการวิเคราะห์การเริ่มต้นบันทึกค่าล่วงเวลา.....	46
รูปที่ 3.36 แสดงการวิเคราะห์การสรุปค่าล่วงเวลา.....	47
รูปที่ 3.37 แสดงการวิเคราะห์การออกจากระบบ.....	47
รูปที่ 3.38 Data Management Object.....	49
รูปที่ 3.39 แสดงการเปลี่ยน DM Object เป็น Table.....	50
รูปที่ 3.40 แสดง Entity และความสัมพันธ์ของระบบค่าล่วงเวลา.....	52
รูปที่ 3.41 แสดงการสร้าง FormBasicJob.....	53
รูปที่ 3.42 แสดงการสร้าง FormWorker.....	55
รูปที่ 3.43 แสดงการสร้าง FormHol.....	56
รูปที่ 3.44 แสดงการสร้าง FormEmployee.....	57
รูปที่ 3.45 แสดงการสร้าง FormOfficer.....	58
รูปที่ 3.46 แสดงการสร้าง FormWorkgroup.....	59
รูปที่ 3.47 แสดงการสร้าง FormRate.....	60
รูปที่ 3.48 แสดงการสร้าง FormHolEmp.....	61
รูปที่ 3.49 แสดงการสร้าง FormHolOffi.....	62
รูปที่ 3.50 แสดงการสร้าง FormDept.....	63
รูปที่ 3.51 แสดงการสร้าง FormLogin.....	64
รูปที่ 3.52 แสดงการสร้าง FormClerk.....	65
รูปที่ 3.53 แสดงการสร้าง FormOtTrans.....	66
รูปที่ 3.54 แสดงการสร้าง FormNewMonth.....	67
รูปที่ 3.55 แสดงการสร้าง FormPrintDailyOt.....	68
รูปที่ 3.56 แสดงการสร้าง FormRptDailyOt.....	69
รูปที่ 3.57 แสดงการสร้าง FormRptMonthlyOt.....	71
รูปที่ 3.58 แสดงการสร้าง Window ที่ใช้เป็นเมนู.....	72

สารบัญตาราง

หน้า

ตารางที่ 1.1 ขั้นตอนการศึกษา.....	9
ตารางที่ 2.1 Windows จากการออกแบบในขั้นตอน PD ยกเว้น Transaction Object.....	36
ตารางที่ 2.2 Windows ส่วนที่ต้องใช้งานเพิ่มเติม	36
ตารางที่ 2.3 Windows ที่ได้จากการวิเคราะห์ความต้องการ.....	37
ตารางที่ 2.4 Windows สำหรับเป็นเมนู.....	37
ตารางที่ 3.1 แสดงรายชื่อ Table ของแต่ละ Data Management.....	51



บทที่ 1

บทนำ

1.1 ความเป็นมา

การพัฒนาโปรแกรมในปัจจุบัน ประสบปัญหาที่สำคัญอันหนึ่งคือ ปัญหาการบำรุงรักษา (Maintenance) ซึ่งเป็นปัญหาที่รุนแรงเพิ่มมากขึ้นเรื่อย ๆ และสิ้นเปลืองค่าใช้จ่าย ตลอดจนเสียทรัพยากรโดยรวมในการพัฒนา Software มากกว่าครึ่งหนึ่งของการพัฒนาทั้งหมด จึงมีการนำวิธีการพัฒนาแบบ Object-Oriented เพื่อแก้ไขปัญหาลักษณะนี้ โดยเป็นวิธีการพัฒนาที่จะมุ่งเน้นไปที่ การนำส่วนประกอบของ Software กลับมาใช้ใหม่ (Reusability of S/W Component) เป็นโปรแกรมที่สามารถปรับปรุงแก้ไขได้ง่าย และผู้อื่นสามารถทำความเข้าใจกับโปรแกรมของเดิมได้

ข้อดีของการพัฒนาที่มุ่งเน้น Object มากกว่าจะเน้นที่บทบาทคือ

1. Object ในระบบงานนั้นไม่ค่อยเปลี่ยนแปลงเท่ากับบทบาทงาน โดยเฉพาะงานฐานข้อมูล ทำให้การเขียนโปรแกรมจัดการข้อมูลง่ายขึ้น
2. การประกอบ Software ส่วนย่อยเข้าด้วยกัน ทำได้ง่ายขึ้นเพราะไม่ต้องกังวลกับความแตกต่างเล็กน้อยในเรื่องโครงสร้างข้อมูลหรือความเปลี่ยนแปลงภายในคำสั่งปฏิบัติการของแต่ละชุดคำสั่งย่อยของแต่ละส่วนที่จะต้องมาต่อประกอบกัน

จากปัญหาและข้อดีในการพัฒนาแบบ Object-Oriented ดังกล่าวข้างต้น จึงเกิดโครงการศึกษากรณีพิเศษ เรื่องการพัฒนาระบบค่าล่วงเวลาด้วยหลักการเชิงวัตถุ เพื่อศึกษาการพัฒนาแบบ Object-Oriented ในด้านความสามารถในการนำไปใช้ในการพัฒนาระบบได้จริง และระบบที่พัฒนา ด้วยเทคนิค Object-Oriented จะสามารถแก้ปัญหาดังกล่าวข้างต้นได้

1.2 วัตถุประสงค์

- 1) เพื่อวิเคราะห์และออกแบบระบบค่าล่วงเวลา โดยใช้เทคนิคของ Object-Oriented analysis and design บนพื้นฐานของ Object Modeling Technique (OMT) ของ Peter Coad
- 2) เพื่อจัดทำโปรแกรมต้นแบบระบบค่าล่วงเวลาโดยใช้เทคนิคของ Object-Oriented Programming(OOP)

1.3 ขั้นตอนการศึกษา

ลำดับที่	รายละเอียด	ก.ค	ส.ค	ก.ย	ต.ค
1	วิเคราะห์ระบบค่าล่วงเวลา	■			
2	ศึกษาและคัดเลือก Object-Oriented Method		■		
3	ออกแบบระบบ			■	
4	ออกแบบฐานข้อมูล			■	
5	ศึกษาภาษา Delphi			■	
6	พัฒนาโปรแกรม				■
7	ทดสอบและปรับปรุงแก้ไข				■
8	สรุปผลการศึกษาและข้อเสนอแนะ				■
9	เขียนรายงานและการแก้ไข				■

ตารางที่ 1.1 ขั้นตอนการศึกษา

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถนำแนวคิดและทฤษฎีต่าง ๆ ที่ได้ศึกษาในการทำโครงการศึกษากรณีพิเศษ ไปประยุกต์ใช้กับการพัฒนาระบบงานจริงได้ ซึ่งได้แก่
 - การวิเคราะห์และออกแบบระบบ ด้วยหลักการ Object-Oriented
 - การพัฒนาโปรแกรมแบบ Object-Oriented Programming
- 2) ต้นแบบระบบค่าล่วงเวลาในการออกแบบและพัฒนาโปรแกรมโดยใช้หลักการของ Object-Oriented

บทที่ 2

วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

2.1 แนวคิดพื้นฐานเกี่ยวกับ Object

วิธีการพัฒนาระบบแบบ Object-Oriented จะต้องสนับสนุนและใช้แนวความคิดของสิ่งเหล่านี้

1) Objects

Objects เป็น entities ซึ่งมีการโต้ตอบกับ messages messages เป็นคำสั่งให้ Object กระทำการอย่างใดอย่างหนึ่ง การกระทำของ Object แต่ละตัวคือ กลุ่มของ Methods(functions) ที่จัดไว้และมอบให้กับ Object นั้น แต่ละ Object ได้จัดหาคคุณสมบัติของ encapsulation ซึ่งเป็นการ implementation การกระทำและโครงสร้างข้อมูลภายในของ Object ให้ถูกซ่อนไว้จากผู้ใช้งาน ซึ่งทำให้คุณสมบัติของ encapsulation มีประโยชน์ในเรื่องช่วยซ่อนรายละเอียด ที่ไม่จำเป็นที่ผู้ใช้งานต้องรู้ และช่วยให้การ implementation อีกครั้งของ Object จะไม่ส่งผลกระทบต่อ Object อื่น ๆ ในระบบ

2) Classes

Classes เป็นคำที่ใช้เรียกกลุ่มของ objects ที่มีพฤติกรรมเหมือนกัน Objects เหล่านี้ บางทีก็เรียกว่า instances หรืออาจกล่าวอีกนัยหนึ่งก็คือ Object หรือ instances สามารถสร้างขึ้นจาก class

3) Inheritance

Inheritance เป็นการสืบทอดทายาท เช่น เมื่อมีการกำหนด class ไว้แล้ว โดยมีความสามารถและลักษณะท่าทางต่าง ๆ จะสามารถถ่ายทอดให้กันได้ โดยตัวต้นแบบจะเรียกว่า พ่อ(parent) ส่วนตัวที่รับมรดกหรือสืบทอดจากพ่อเรียกว่าลูก (Child) เมื่อเริ่มต้นลูกจะมีหน้าตาเหมือนพ่อทุกอย่าง จากนั้นลูกยังสามารถกำหนดความสามารถใหม่เพิ่มเติมเข้ามา หรือเปลี่ยนแปลงลักษณะบางอย่างจากตัวพ่อได้เมื่อมีการเปลี่ยนตัวพ่อตัวลูกก็จะถูกแก้ไขไปโดยอัตโนมัติ โดยที่ตัวภาษาจะรู้เอง

4) Polymorphism

Polymorphism คือความสามารถต่าง ๆ จะแปรเปลี่ยนไปตาม Object นั้น ๆ โดยที่มาจากจุดกำเนิดร่วมกัน เช่น เมื่อใส่ความสามารถในการวาด (Method Draw) ให้กับ Class แบบ เส้นตรง ซึ่งเราได้ทำการ inheritance class แบบเส้นตรง นี้มาให้กับ Class ใหม่ 2 แบบ ซึ่งได้แก่ วงกลมและ สี่เหลี่ยม โดยทั้งวงกลมและสี่เหลี่ยม จะใช้ความสามารถของเส้นตรง ที่มีอยู่ในการวาดด้วยเช่นกัน ดังนั้น เมื่อ Object ที่มี Class เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบวงกลมถูกเรียกด้วยคำสั่ง Draw มันจะวาดรูปวงกลม แต่ถ้าเป็น Object ที่มี Class เป็นสี่เหลี่ยม ถูกเรียกด้วยคำสั่งเดียวกัน จะได้ผลลัพธ์เป็นสี่เหลี่ยม

2.2 แนวคิดและเทคนิคของ Object-Oriented

วิธีการแบบ Object-Oriented ในการพัฒนา Software เริ่มเป็นที่นิยม มากขึ้นอย่างรวดเร็ว ซึ่งเชื่อกันว่าสามารถช่วยในการพัฒนา Software ให้มีความยืดหยุ่นสูง (Extensible) สามารถนำกลับมาใช้ใหม่(Reusable) และสามารถรองรับงานได้ดี(Robust)

2.2.1 การวิเคราะห์และออกแบบระบบแบบ Object-Oriented

จะเน้นไปที่ตัว Object ที่ประกอบกันเป็นระบบงาน ไม่ใช่เน้นไปยังบทบาทงาน (Action หรือ Function) ที่มีอยู่ในระบบงานอย่างที่เคยเป็นวิธีการแบบ Structured Analysis and Design ดังนั้นคำถามในการออกแบบเชิงวัตถุจะเป็นว่า ‘มีวัตถุอะไรบ้างในระบบงานนี้’ แทนที่จะเป็นว่า ‘มีงานอะไรบ้างที่ต้องกระทำในระบบงานนี้’ ซึ่งมีข้อดีคือ

- 1) วัตถุในระบบงานนั้นจะไม่ค่อยเปลี่ยนแปลงบ่อย เท่ากับบทบาทงานโดยเฉพาะกับงานฐานข้อมูล จึงทำให้การเขียนโปรแกรมจัดการข้อมูลเป็นไปได้ง่ายขึ้น
- 2) การประกอบ Software ส่วนย่อยเข้าด้วยกัน จะทำได้ง่ายขึ้นเพราะไม่ต้องกังวลกับความแตกต่างเล็กน้อยในเรื่องโครงสร้างข้อมูลหรือความเปลี่ยนแปลงภายในคำสั่งปฏิบัติการของแต่ละชุดคำสั่งย่อยของแต่ละส่วนที่จะมาประกอบกัน

การออกแบบ โดยวิธี Object-Oriented สร้างขึ้น เพื่อตอบสนองกับแนวคิดของ Object-Oriented Languages แต่ไม่ขึ้นอยู่กับรายละเอียดของภาษาที่ใช้ในการ implementation

2.2.2 ตัวอย่างแนวคิดแบบ Object-Oriented

เป้าหมายของการพัฒนา Software ด้วยวิธี Object-Oriented คือการทำ Object-Oriented decomposition จากความต้องการของผู้ใช้ เข้าสู่ executable language constructs ถึงแม้ว่าแต่ละแนวความคิดของ Object-Oriented ซึ่งมีอยู่มากมาย และมีความแตกต่างในการทำ Object-Oriented decomposition แต่ทุกแนวคิดอยู่บนพื้นฐานของ Object เหมือนกัน ตามที่ได้อธิบายความหมายไว้ในข้อ 2.1 ดังมีตัวอย่างแนวคิดดังต่อไปนี้

- 1) Booch's Object-Oriented Design Methods ได้เสนอการใช้เครื่องหมายแทน class utilities, class categories, class, state transition, objects, modules, subsystems, processors, device processes แต่ละองค์ประกอบจะแทนด้วยรูปภาพเรียกว่า Icons และแบบที่กำหนดไว้ก่อนเรียกว่า templates ซึ่งไม่ต้องอ่านมากแต่จะให้รายละเอียดได้มากกว่า และแนวคิดนี้เสนอขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กำหนด classes และ objects
- กำหนดความเปลี่ยนแปลงให้กับ classes และ objects
- กำหนดความสัมพันธ์ระหว่าง classes และ objects
- Implementing classes และ objects

2) The Responsibility-Driven approach ได้ กำหนดการทำงานเป็น 5 ขั้นตอน

- จำแนกประเภทของ classes และ class hierarchies
- อธิบายการทำงานในแต่ละ class หรือที่เรียกว่า responsibility
- object interactions
- จับกลุ่มของ class ให้เป็นส่วนหนึ่งของ subsystem สร้างโดยใช้พื้นฐานของ object interactions
- ดูความแตกต่างของ object interface เพื่อสร้าง protocols

3) Object-Oriented Software Engineering (OOSE) เป็นแนวคิดที่ใช้การอธิบายระบบงานด้วย Use case เพื่อแสดงงานที่ดำเนินการในระบบและผู้ใช้ระบบติดต่อกับระบบอย่างไร โดยมีขั้นตอนดังนี้

- Requirement model
- Analysis model
- Design model

4) Object-Oriented Analysis&Design(OOA) โดย Coad & Yourdon เป็นวิธีการที่ใช้หลัก step-by-step มีการอธิบายฟังก์ชันต่าง ๆ ที่เกี่ยวข้องกับระบบการติดต่อกับผู้ใช้, DBMS และเป็นสถาปัตยกรรมแบบ Matrix เพราะแยกการวิเคราะห์ออกเป็น 2 ระดับคือ

- ระดับตามแนวตั้ง ซึ่งมี 4 ขั้นตอน ได้แก่
 1. หา class และ object
 2. Identifying structure ดูโครงสร้างของ Class และ inheritance
 3. Define attribute เป็นการกำหนดคุณลักษณะของข้อมูลและอธิบายความสัมพันธ์ของข้อมูล
 4. Define services เป็นการนิยาม operation ของ class โดยชี้ถึงสถานะภาพและการบริหาร เช่น การค้นหาข้อมูล การเพิ่ม แก้ไข ลบข้อมูล เป็นต้น

- ระดับตามแนวนอน ซึ่งมี 4 ระดับคือ

1. Problem domain ตีมาจาก Analysis model
2. Human interaction ใช้ออกแบบการติดต่อกับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. System interaction กำหนด object interaction
4. Data management เพื่อสร้างความมั่นคงให้กับ objects

5) Object Modeling Technique (OMT) โดย Peter Coad ,David North และ Mark MayField โดยมุ่งเน้นที่ Object Model ซึ่งมีลักษณะคล้ายคลึงกับแนวคิดแบบอื่น ๆ โดยเฉพาะคล้ายคลึงกับแนวคิด แบบ Object-Oriented Analysis & Design โดย Coad&Yourdon แต่แนวคิดแบบนี้จะเน้นความสัมพันธ์ระหว่าง Objects มากขึ้น

จากแนวคิดแบบต่าง ๆ ดังกล่าวข้างต้น ซึ่งมีความแตกต่างกันในวิธีการและสัญลักษณ์ที่ใช้ในการออกแบบ สำหรับโครงการศึกษาระดับปริญญาโท ในการพัฒนาระบบ คำล่ำวเวลานี้ขอเลือกใช้แนวคิดแบบ OMT เนื่องจากมีหลักการแบบ step-by-step และใช้หลักการที่เข้าใจง่ายตลอดจนใช้สัญลักษณ์ ที่ง่ายต่อความเข้าใจของผู้อ่านแผนภาพ

2.2.3 ลักษณะของ Object-Oriented Programming (OOP)

- 1) OOP คือโปรแกรมที่สนับสนุน คุณสมบัติต่าง ๆ ที่กล่าวในข้อ 2.1
- 2) วิธีการของ OOP คือการนำชิ้นส่วนของโปรแกรมที่ทำไว้เสร็จแล้วเป็นส่วน ๆ เรียกแต่ละชิ้นเป็น Object มาต่อประกอบกัน วิธีการแบบนี้เป็นการสร้างแบบล่างขึ้นบน (Bottom-up) ซึ่งจะต่างจากวิธีแบบเดิมคือ Structured Programming ที่ใช้หลักแบบบนลงล่าง (Top-down)
- 3) OOP สามารถสร้าง Class ซึ่งข้อมูลชนิดนี้สามารถสร้างขึ้นจาก Inheritance และสามารถจะมีลักษณะของ Polymorphism
- 4) Class เป็นการสร้าง Object และเมื่อมีการส่งข้อความ (Message) ไปยัง Object Object ก็จรรู้ว่าต้องทำอะไรเมื่อได้รับข้อความ
- 5) สามารถลดการขึ้นต่อกันของแต่ละโมดูล เพราะเราสามารถดัดแปลงแก้ไขโปรแกรมโดยไม่มีผลกระทบต่อส่วนอื่น
- 6) Inheritance จะช่วยให้สามารถนำโปรแกรมเก่ามาใช้ใหม่ได้ และเมื่อใช้กับ Polimorphism โดย Inheritance จะช่วยให้เราใช้โปรแกรมเก่า ๆ และสร้างส่วนที่เพิ่มได้

2.2.4 วัตถุประสงค์ของ OOP

การใช้หลักการของ OOP มีวัตถุประสงค์คือ

- 1) เพื่อให้เกิดการทำงานแบบเป็นโมดูล(Modularization) คือสามารถแยกการทำงานเป็นหน่วยจำเพาะย่อย ๆ ซึ่งแต่ละหน่วยสามารถทำงานเป็นอิสระ แล้วค่อยทำการรวมแต่ละหน่วยเข้าด้วยกันเป็นโปรแกรมใหญ่
- 2) ความเข้ากันได้(Compatibility) เป็นประเด็นสำคัญ
- 3) การนำกลับมาใช้ใหม่ (Reusability) สำหรับโปรแกรมขนาดใหญ่ ซึ่งมีโครงสร้างข้อมูลขนาดใหญ่ ทำให้เกิดการกระจายของตัวแปร และข้อมูลต่าง ๆ การนำตัวแปรหรือข้อมูลกลับมาใช้ใหม่จึงเกิดความสับสน OOP สามารถที่จะแก้ปัญหานี้ได้โดยที่ผู้ใช้ไม่จำเป็นต้องรู้โครงสร้างทั้งหมดของโปรแกรม
- 4) ความต่อเนื่อง ซึ่งสามารถใช้คุณสมบัติ OOP ในแง่ของ การสรุป (Abstraction) ได้

2.3 แนวคิดเกี่ยวกับเทคโนโลยี

2.3.1 แนวคิดการใช้งานระบบ Client/Server

การพัฒนาระบบงานประยุกต์ทางด้านคอมพิวเตอร์มีมานานพร้อม ๆ กับการก่อกำเนิด ของระบบคอมพิวเตอร์ ซึ่งเมื่อก่อนระบบคอมพิวเตอร์จะมีขนาดใหญ่โต ต้องการ การดูแลบำรุงรักษาอย่างดี ส่วนใหญ่จะเป็นระบบคอมพิวเตอร์แบบ Mainframe หรือ Mini-computer ในช่วงระยะเวลาต่อมา มีการคิดค้นและผลิตอุปกรณ์คอมพิวเตอร์ให้มี ขนาดเล็กลง และมีประสิทธิภาพสูงขึ้น

ในทศวรรษที่ 1990s Microcomputers ได้รับการพัฒนา ทั้งด้าน Hardware, Software และอุปกรณ์ต่อพ่วงให้มีประสิทธิภาพสูงขึ้น มีความน่าเชื่อถือมากขึ้น ทำให้นำ ไปสู่การพัฒนาที่มีความสะดวก สบาย ง่ายต่อการใช้งานมากขึ้น และมีขนาดเล็กลงมาก ส่วนทางด้าน Software ก็ได้มีการพัฒนาให้ง่ายต่อการใช้งานโดยอาศัยเทคนิคของ Graphic แทนที่จะเป็น Text ซึ่งเรียกว่า Graphic User Interface (GUI) เช่น Window 3.11, Window 95, Window NT ซึ่งเป็นการใช้แนวคิดแบบ Object-oriented นอกจากนี้ยังมี เทคโนโลยีการพัฒนาระบบงานประยุกต์แบบ Client/Server Technology และ ระบบงานเครือข่ายแบบไร้สาย(Wireless Networks)

2.3.2 ความหมายของระบบ Client/Server

หมายถึง ระบบงานที่มีการจัดแบ่งหน้าที่ในการทำงาน การประมวลผลของแต่ละงาน ให้เครื่องคอมพิวเตอร์ เพื่อให้เกิดประสิทธิภาพและความเหมาะสมมากที่สุด โดยแบ่งเครื่องคอมพิวเตอร์ในระบบออกเป็น เครื่องที่เป็นผู้ขอใช้บริการ(Client) และเครื่องที่เป็นผู้ให้บริการ(Server)

2.3.3 องค์ประกอบของระบบ Client/Server ประกอบด้วย 3 องค์ประกอบคือ

- 1) Client หมายถึง เครื่องคอมพิวเตอร์ที่ผู้ใช้งานโดยทั่วไป สามารถใช้งานในระบบงานได้โดยตรง สามารถเรียกโปรแกรม ของระบบงานมาใช้ โดยทำงาน ในส่วนของ Front-End Processing และจะขอใช้ข้อมูล และResourceต่าง ๆ จากเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server โดยผ่านระบบเครือข่าย
- 2) Server หมายถึง เครื่องคอมพิวเตอร์ที่ทำหน้าที่ ในส่วนของการให้บริการในการจัดการด้านฐานข้อมูล และให้บริการต่าง ๆ กับเครื่องที่อยู่ในเครือข่าย
- 3) Network หมายถึง เครือข่ายคอมพิวเตอร์ ทั้งในส่วนของเครือข่ายสื่อสารระยะไกล(Wide Area Network) และ เครือข่ายสื่อสารระยะใกล้(Local Area Network) ที่สามารถเชื่อมเครือข่ายคอมพิวเตอร์ที่อยู่ในสถานที่ต่างกัน และอยู่ต่างระบบเข้ามาทำงานด้วยกันได้

2.3.4 วัตถุประสงค์ของการใช้งานระบบ Client/Server

- สามารถเชื่อมโยงให้ทุก Server ทุก Client ที่เชื่อมโยงถึงกันได้ และสามารถแลกเปลี่ยนข้อมูลข่าวสารกันได้
- เพื่อให้ Software ทุกชนิด ทุกประเภท ทั้งหมดในเครือข่ายคอมพิวเตอร์สามารถทำงานสอดคล้องกัน ประสานกันได้อย่างมีประสิทธิภาพ
- เพื่อประหยัด งบประมาณ ค่าใช้จ่าย ในระบบคอมพิวเตอร์และ สารสนเทศขององค์กร โดยสามารถวางแผนติดตั้งอุปกรณ์ ได้เป็นขั้นตอนตามความจำเป็น
- เพื่อให้เหมาะสมกับระบบงานที่ใช้อยู่
- เพื่อให้เกิดความยืดหยุ่นในการทำงาน และสามารถขยายระบบเพิ่มขึ้น เมื่อระบบงานมีปริมาณการใช้งานเพิ่มขึ้น
- เพื่อให้สามารถใช้อุปกรณ์ในเครือข่ายได้อย่างมีประสิทธิภาพ

2.3.5 ประโยชน์ของการพัฒนาระบบงานในแนวคิดแบบ Client/Server

- ช่วยประหยัดงบประมาณในการลงทุน
- ทำให้เกิดประสิทธิผลของระบบงานมากขึ้น
- มีความยืดหยุ่นและสามารถขยายตัวได้
- ทำให้เกิดการใช้ทรัพยากรที่มีอยู่ให้ได้ประโยชน์สูงสุด
- สามารถบริหารการควบคุมได้จากส่วนกลาง
- มีคุณสมบัติที่เป็นระบบแบบเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 เครื่องมือในการพัฒนาระบบงานแบบ Client/Server

ในการพัฒนาระบบงานประยุกต์ ทางด้านคอมพิวเตอร์ในแนวคิดแบบ Client/Server เราสามารถพิจารณาใช้ ภาษาโปรแกรม ระบบฐานข้อมูล และเครื่องมือในการพัฒนาระบบงานได้หลากหลายรูปแบบ ขึ้นอยู่กับผู้พัฒนาระบบจะเลือกใช้ โดยเครื่องมือที่ใช้มักจะแบ่งออกเป็น 2 ส่วนคือ

1) Front-End Development Tools

เป็นชุดของโปรแกรม ที่ใช้พัฒนาระบบงาน ในส่วนที่จะใช้ติดต่อกับผู้ใช้ระบบบน Client เช่น รับคำสั่ง ,แสดงผล หรือโต้ตอบกับผู้ใช้งาน ที่เรียกว่า User Interface รวมทั้งการประมวลผลและการตรวจสอบความถูกต้องเบื้องต้นของการทำงานของผู้ใช้งาน เครื่องมือที่นิยมใช้ เช่น Delphi, Visual Basic, Visual C++, Visual Foxpro, Power Builder, Microsoft Access, Oracle Developer 2000 เป็นต้น

2) Back-End Development Tools

เป็นโปรแกรมจัดการฐานข้อมูล ที่ทำงานบน Server โดยจะทำหน้าที่ในส่วนของการจัดการฐานข้อมูล เช่นการค้นหาการปรับปรุงข้อมูลบน Database รวมทั้งการประมวลผล หรือตรวจสอบข้อมูลเล็กน้อย และส่งข้อมูลกลับไปให้ Client เครื่องมือในส่วนนี้ ส่วนใหญ่จะเป็นโปรแกรมจัดการฐานข้อมูล เช่น DB2, Oracle, Informix, Ingres, Progress, SQLServer, InterBase เป็นต้น

สำหรับการพัฒนาระบบค่าล่วงเวลานี้ จะใช้ Delphi Client/Server Suite เป็น Front-End Development tool และใช้ InterBase เป็น Back-End Development Tool

2.3.7 Delphi เครื่องมือพัฒนาในส่วน Front-End

Delphi เป็นโปรแกรม ที่จะช่วยพัฒนาโปรแกรมบน Windows ของบริษัท Borland ซึ่งเป็นบริษัทที่พัฒนาตัว Compile ภาษาคอมพิวเตอร์ต่าง ๆ เช่น Pascal, C ที่เป็นที่ยอมรับกันอยู่ในปัจจุบัน

Delphi เป็นภาษาคอมพิวเตอร์ใหม่ ซึ่งเป็นการเขียนโปรแกรม แบบ Visual โดยมีลักษณะของการพัฒนาโปรแกรมที่ง่ายต่อการเรียนรู้และนำมาใช้งาน รูปแบบของภาษาจะมีลักษณะเดียวกับ Pascal แต่โครงสร้างของการทำงานจะไม่มีลำดับการทำงาน โดยโปรแกรมจะทำงานตามเหตุการณ์ที่เกิดขึ้น

นอกจากนี้ Delphi ยังมีลักษณะของโปรแกรมเชิงวัตถุ (Object-Oriented Programming) และสามารถสร้าง Class หรือ Object ขึ้นเองได้ ซึ่งเป็นปัจจัยสำคัญใน

การเลือก Delphi มาพัฒนาระบบค่าล่วงเวลา โดยทุก Class และทุก Object ใน Delphi จะถือกำเนิดจาก Class ที่ชื่อว่า "TObject" ทั้งสิ้น

องค์ประกอบของ Delphi ประกอบด้วย

- Main Menu Bar แบ่งออกเป็น 4 ส่วน ได้แก่
 - Title Bar
 - Menu Bar
 - Speed Bar
 - Component Palette มีไว้ให้เลือก Object ต่าง ๆ ที่ สร้างขึ้นมา จาก Class ของ Delphi หรือ Object ที่ผู้ใช้งาน สร้างขึ้นมาใหม่ โดยสืบ ทอดจาก Class ที่ Delphi มีอยู่ (ใน Delphi จะเรียก Object ว่า Component) ถ้าต้องการรู้ว่า Object ที่ใช้งาน เกิดจากการ inherit มา จาก Class ไตบ้าง ให้ไปที่ Menu Bar จากนั้นเลือก View /Browser
- Object Inspector แบ่งเป็น 2 ส่วน ได้แก่
 - Properties สำหรับให้ค่า Properties ตามต้องการ
 - Events สำหรับกำหนดกิจกรรม Events
- Form Designer สำหรับวางรูปแบบของ Form
- Program Editor สำหรับป้อนโปรแกรม

2.3.8 InterBase เครื่องมือพัฒนาในส่วน Back-End

เป็นชุดของโปรแกรมจัดการฐานข้อมูลแบบ Relational (RDBMS) ของบริษัท BORLAND ซึ่งพัฒนาให้ใช้งานควบคู่กับชุดของ โปรแกรม Delphi ปัจจุบันพัฒนามาถึง Version 4.2 ใช้กับ Delphi 3.0 ซึ่งสามารถทำงานบนระบบที่เป็น Windows 95, Windows NT, Netware และ UNIX

InterBase Server มี 2 ชุด คือ

- 1) ชุด Client/Server เป็นชุดที่ใช้ติดตั้งบนเครื่องที่เป็น Server สามารถใช้งาน ได้กับผู้ใช้งานหลายคน
- 2) ชุด Local InterBase Server เป็นชุดที่ใช้งานได้กับผู้ใช้คนเดียว ซึ่งเหมาะ กับการใช้งานในระหว่างการพัฒนาและทดสอบระบบงานก่อนการใช้งานจริง โดยจะรวมมากับ ชุดของโปรแกรม Delphi

นอกจากเป็นโปรแกรมจัดการฐานข้อมูลแล้ว InterBase ยังมีเครื่องมือ ช่วยอำนวยความสะดวกต่อผู้ใช้งานดังนี้

- 1) Windows Tools :
 - Windows ISQL
 - Server Manager

- Communications Diagnostic Tool
- Registry Configuration Tool

2) Command-Line Utilities :

- Database Definition Utility
- Database Repair Utility
- User Security Utility
- Database Statistics Analyzer
- Database Backup/Restore Utility
- InterBase Registry Installation Utility
- Lock Manager Print Utility
- Interactive SQL Utility
- Server Shutdown and Restart
- Server Configuration Utility

3) ODBC Driver



บทที่ 3

วิธีดำเนินการศึกษา

3.1 การศึกษาและวิเคราะห์ข้อมูลค่าล่วงเวลา

3.1.1 วิธีปฏิบัติเกี่ยวกับการทำงานล่วงเวลาที่ใช้ในปัจจุบัน

การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย มีผู้ปฏิบัติงาน(พนักงานและลูกจ้าง) ประมาณ 38,000 คน ซึ่งแบ่งตามสายวิชาชีพ เช่น วิศวกร, แพทย์, เภสัชกร, วิทยาศาสตร์การแพทย์, นักบัญชี, พนักงานธุรการ,นักธรณีวิทยา,ช่าง เป็นต้น และมีเวลาทำงานอยู่ 3 กลุ่มใหญ่ ๆ คือ

- 1) ทำงานตามปกติ สำหรับพนักงานหรือลูกจ้างที่ทำงานในสำนักงาน
 - ผู้ปฏิบัติงานในสำนักงานใหญ่ (จังหวัด นนทบุรี) สามารถเลือกเวลาทำงานได้ 3 รอบ แล้วแต่ความสะดวกในการเดินทางคือ 7.30-15.30 , 8.30-16.30 และ 9.30- 17.30 น.
 - ผู้ปฏิบัติงานในส่วนภูมิภาค เวลาทำงานคือ 8.30-16.30 น.
- 2) ทำงานเป็นกะ สำหรับพนักงานหรือลูกจ้างที่ทำงานในโรงไฟฟ้า ที่ต้องทำงาน 24 ชม.
- 3) ทำงานแบบเวลาพิเศษ เช่น เข้าทำงาน 7.00 น. เลิกงาน 17.00 น. สำหรับพนักงานหรือลูกจ้างที่ทำงานในโครงการต่าง ๆ เช่น โครงการโรงไฟฟ้าแม่เมาะ เป็นต้น

วิธีปฏิบัติเกี่ยวกับการทำงานล่วงเวลาที่ใช้ในปัจจุบัน คือ เมื่อผู้ปฏิบัติงาน มีการทำงานล่วงเวลา จะมีการลงบันทึกในสมุดประจำวันว่า ทำงานล่วงเวลาเริ่มเมื่อไร สิ้นสุดเมื่อไร และทำงานอะไร ต่อมาเมื่อสิ้นเดือนพนักงานธุรการของแต่ละหน่วยงาน จะออกรายงานสรุปข้อมูลจากรายวัน เป็นรายเดือน โดยมีรายละเอียดเกี่ยวกับ จำนวนชั่วโมงทำงานล่วงเวลาทั้งเดือนของแต่ละอัตราล่วงเวลาเช่น อัตรา 1 เท่าที่ชั่วโมง อัตรา 1.5 เท่าที่ชั่วโมง โดยการคำนวณมีหลักการคิดชั่วโมงทำงานล่วงเวลาดังนี้

- การหักชั่วโมงเนื่องจากรับประทานอาหาร
 - วันทำการ - ถ้าทำงานต่อเนื่องจากเวลาเลิกงาน และทำงานเกิน 19.00 น. จะถูกหัก 30 นาที
 - วันหยุด - ถ้าทำงานในช่วงเวลา 12.00-13.00 น. จะถูกหักเวลาช่วงนี้ออก
- การหาอัตรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วันทำการ - ต้องเป็นเวลาก่อนเวลาทำงานหรือหลังเลิกเวลาทำงานและ ได้อัตรา 1.5 เท่า
- วันหยุด - พนักงาน ถ้าทำงานไม่เกิน 7 ชั่วโมง ได้อัตรา 1 เท่า แต่ถ้าเกิน 7 ชม. ส่วนที่เกิน จะได้อัตรา 3 เท่า
- ลูกจ้าง ถ้าทำงานไม่เกิน 7 ชั่วโมงได้อัตรา 1 เท่า แต่ถ้าเกิน 7 ชม. ส่วนที่เกิน จะได้อัตรา 2 เท่า

จากนั้นนำข้อมูลกรอกลงแบบฟอร์ม ลว.1 สรุปแยกเป็นแต่ละหน่วยงาน และนำส่งให้หัวหน้าของแต่ละหน่วยงานที่ผู้ปฏิบัติงานสังกัด อนุมัติเพื่อนำส่งแผนเงินเดือนต่อไป และข้อมูลในส่วนนี้จะถูกนำไปบันทึกที่แผนกบันทึกข้อมูล และจะส่งไปประมวลผล ที่ Mainframe

ในขั้นตอนการประมวลผลบน Mainframe ข้อมูลค่าล่วงเวลา จะถูกตรวจสอบอีกครั้ง ก่อนนำไปคำนวณเป็นตัวเลข เพื่อรวมกับเงินเดือนและหักรายการต่าง ๆ ของผู้ปฏิบัติงาน ซึ่งรายละเอียดการตรวจสอบเกี่ยวกับค่าล่วงเวลามีดังนี้

- ตรวจสอบข้อมูล Duplicate
- เปรียบเทียบกับจำนวนชั่วโมงที่มีสิทธิ์ทำงานล่วงเวลา ถ้าเกินสิทธิ์จะได้จำนวนชั่วโมงทำงานล่วงเวลาเท่ากับสิทธิ์ของตนเท่านั้น

ถ้าการตรวจสอบข้างต้นมีผลความผิดพลาด จะออกรายงานแสดงความผิดพลาด และส่งไปให้แผนกเงินเดือน เพื่อตรวจสอบและแก้ไขข้อมูล จากนั้นส่งข้อมูลที่ต้องการแก้ไขเข้าประมวลผลที่ Mainframe อีกครั้งหนึ่ง

3.1.2 ปัญหาที่เกิดขึ้นกับระบบค่าล่วงเวลา

- 1) พนักงานธุรการ ใช้เวลามากในการคำนวณจำนวนชม. ล่วงเวลาทั้งเดือน ของผู้ปฏิบัติงานแต่ละคน ตลอดจนมีความซับซ้อนในการหาอัตราค่าล่วงเวลา และ คำนวณจำนวนชม.
- 2) แผนกบันทึกข้อมูล ต้องบันทึกข้อมูลที่มีปริมาณมากเพราะบันทึกข้อมูลของพนักงานทุกหน่วยงาน
- 3) การตรวจสอบว่า ข้อมูล Duplicate หรือ จำนวนชั่วโมงการทำงานล่วงเวลา เกินกว่าสิทธิ์ที่ทำได้หรือไม่ มีความล่าช้า เนื่องจากไปตรวจสอบที่เครื่อง Mainframe
- 4) หน่วยงานแต่ละแห่งไม่มีรายงานการทำงานล่วงเวลา ซึ่งถ้าต้องการ ต้องมาเอารายงานที่แผนกเงินเดือน ซึ่งล่าช้า และไม่ทันกับความต้องการ
- 5) หน่วยงานที่เป็น Business Unit ต้องการข้อมูลในส่วนที่เป็น จำนวนเงินของค่าล่วงเวลาโดยเร็วที่สุด เพื่อจัดการกับค่าใช้จ่ายในส่วนนี้ให้มีประสิทธิภาพ แต่ใน

- ปัจจุบันไม่มี และถึงแม้จะมีรายงานที่แผนกเงินเดือน ก็เป็นเพียงจำนวนชั่วโมง กับ อัตราที่ทำล่วงเวลาเท่านั้น
- 6) หัวหน้าของหน่วยงานแต่ละแห่ง อยากรายงานที่บอกได้ว่า ผู้ปฏิบัติงานอยู่ทำงานล่วงเวลา ให้กับงานอะไรบ้าง
 - 7) พนักงานที่ทำงานล่วงเวลา อยากรู้ว่าเดือนนี้ ทำงานล่วงเวลาวันไหนบ้าง ทำอะไร และจำนวนช.ม. ที่ทำงานล่วงเวลากี่ช.ม.

3.2 การวิเคราะห์และออกแบบระบบ

ในการวิเคราะห์และออกแบบระบบ เพื่อแก้ปัญหาของระบบค่าล่วงเวลาข้างต้น โดยจะใช้เทคนิค Object Modelling Technique(OMT) ของ Peter Coad ซึ่งมีกิจกรรมหลัก ๆ อยู่ 2 ข้อคือ

3.2.1 กำหนดวัตถุประสงค์

- คำนวณจำนวนเงินค่าล่วงเวลา แบบอัตโนมัติ
- ตรวจสอบสิทธิ์การทำงานล่วงเวลา
- ตรวจสอบเวลาในการทำงานล่วงเวลา
- หาอัตราล่วงเวลาของแต่ละรายการค่าล่วงเวลา
- บันทึกรายการค่าล่วงเวลา
- ออกรายงานค่าล่วงเวลารายบุคคล
- ออกรายงานสรุปค่าล่วงเวลารายเดือนส่งแผนกเงินเดือน

จากวัตถุประสงค์ของระบบข้างต้น สามารถช่วยการทำงานเกี่ยวกับค่าล่วงเวลาของพนักงานธุรการให้มีประสิทธิภาพและได้ข้อมูลค่าล่วงเวลาที่ต้องการ ในเรื่องต่อไปนี้

- ลดความผิดพลาดในการคำนวณจำนวนชั่วโมงการทำงานล่วงเวลา ของทั้งเดือน
- ลดความผิดพลาดของจำนวนเงินค่าล่วงเวลา
- ลดความผิดพลาดของการคิดอัตราค่าล่วงเวลา
- สามารถส่งรายการค่าล่วงเวลาให้แผนกเงินเดือนได้ รวดเร็วขึ้น

3.2.2 ลักษณะสำคัญของระบบ

- ข้อมูลหลักของระบบค่าล่วงเวลา
 - ข้อมูลผู้ปฏิบัติงาน
 - ข้อมูลวันหยุด

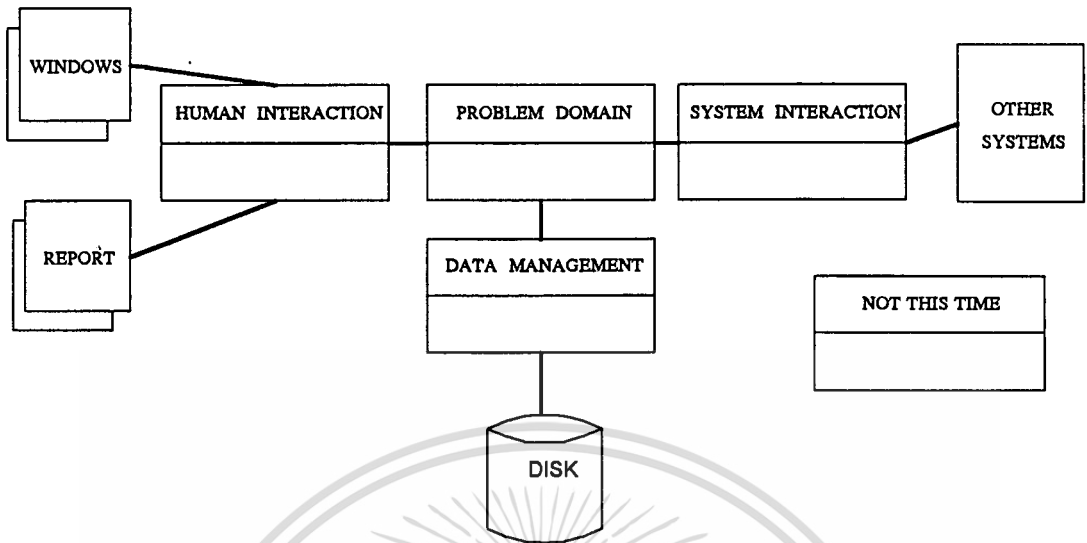
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อมูลอัตราค่าล่วงเวลา
- ข้อมูลกลุ่มเวลาเข้างาน
- ข้อมูลค่าล่วงเวลา
- ลักษณะสำคัญของการทำงานล่วงเวลา
 - การหาอัตราค่าล่วงเวลา
 - การคำนวณจำนวนเงินค่าล่วงเวลา และจำนวนชั่วโมงค่าล่วงเวลารายเดือน
 - ตรวจสอบสิทธิในการทำงานล่วงเวลา
- ลักษณะสำคัญของระบบในแง่การนำไปใช้ประโยชน์เชิงวิเคราะห์
 - มีจำนวนผู้ปฏิบัติงานกี่คน และเป็นใครบ้าง ที่ทำงานล่วงเวลาเกินสิทธิ์เพื่อนำไปพิจารณาในรายละเอียดว่า มีเหตุผลอะไรที่ต้องทำงานล่วงเวลาเกินสิทธิ์ ซึ่งถ้ามีเหตุผลเพียงพอ จะได้เพิ่มจำนวนชั่วโมงในการทำงานล่วงเวลา
 - มีจำนวนผู้ปฏิบัติงานกี่คน และเป็นใครบ้าง ที่ไม่ได้ทำงานล่วงเวลา เพื่อนำไปประเมินประสิทธิภาพในการทำงาน
- ลักษณะสำคัญของระบบในการติดต่อกับระบบอื่น (ระบบนี้ไม่มีการติดต่อกับระบบอื่น)

3.2.3 วิเคราะห์และออกแบบระบบ เป็นการออกแบบส่วนประกอบต่าง ๆ ของ

Object Model ซึ่งมี 5 ส่วน ดังต่อไปนี้คือ

- 1) Problem Domain (PD)
- 2) Human Interaction (HI)
- 3) Data Management (DM)
- 4) System Interaction (SI) สำหรับระบบนี้จะไม่มี เพราะไม่ได้มีการติดต่อกับระบบอื่น
- 5) Not This time (NT)



รูปที่ 3.1 ส่วนประกอบของ Object Model และ ส่วนประกอบอื่น ๆ ที่เกี่ยวข้อง

1. Problem Domain (PD) มี 4 ขั้นตอนคือ

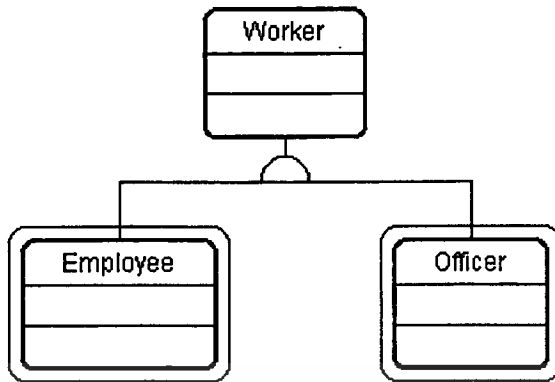
- 1.1 วิเคราะห์หา Object ที่ใช้ในระบบ
- 1.2 พิจารณาความสัมพันธ์ของแต่ละ PD Object
- 1.3 วิเคราะห์ความรับผิดชอบของแต่ละ PD Object
- 1.4 วิเคราะห์การทำงานของ PD Object ในระบบ

1.1 วิเคราะห์หา Object ที่ใช้ในระบบ ซึ่งมี 3 ส่วนใหญ่ ๆ ดังนี้

- ผู้กระทำ และผู้มีส่วนร่วม
- สถานที่
- สิ่งของ

ผู้กระทำและผู้มีส่วนร่วม

- ผู้กระทำ คือ ผู้ที่กระทำแล้วก่อให้เกิดข้อมูลในระบบ ซึ่งได้แก่ พนักงาน (Officer) และลูกจ้าง (Employee) ซึ่งมีคุณสมบัติเหมือนกัน แต่ใช้อัตราค่าสว่างเวลาในการคำนวณต่างกัน ดังนั้นจึง สามารถนำเอาคุณสมบัติที่เหมือนกันสร้างเป็น Class ขึ้นมา โดยใช้ชื่อว่า ผู้ปฏิบัติงาน (Worker) ดังนั้น Object ที่ชื่อว่า Employee และ Officer จะ Inherit จาก Worker Class (ความหมายของสัญลักษณ์ต่าง ๆ ที่ใช้จะแสดงในภาคผนวก ก)



รูปที่ 3.2 Class และ Object ของผู้กระทำ

- ผู้มีส่วนร่วม จะมี 2 กลุ่มคือ
1. เป็นผู้ที่เกี่ยวข้องกับระบบ ในแง่การติดตาม ตรวจสอบ อนุมัติ ซึ่งได้แก่ หัวหน้าของแต่ละหน่วยงาน (Head of workers) ซึ่งจะทำหน้าที่อนุมัติการทำงานล่วงเวลาของผู้ใต้บังคับบัญชาของตน
 2. เป็นผู้ถูกประเมินประสิทธิภาพ การทำงานของระบบค่าล่วงเวลา ซึ่งได้แก่ พนักงานธุรการ(Clerk) เพื่อประเมินประสิทธิภาพ การทำงานในระบบค่าล่วงเวลา
- ในการทำงานของระบบนี้ ผู้มีส่วนร่วม ทั้งข้อ 1 และ ข้อ 2 จะยังไม่ครอบคลุมถึง ดังนั้น ผู้มีส่วนร่วมทั้ง 2 คน จะอยู่ในส่วนของ Not This Time(NT)



รูปที่ 3.3 สรุป PD และ NT ของผู้กระทำและผู้มีส่วนร่วม

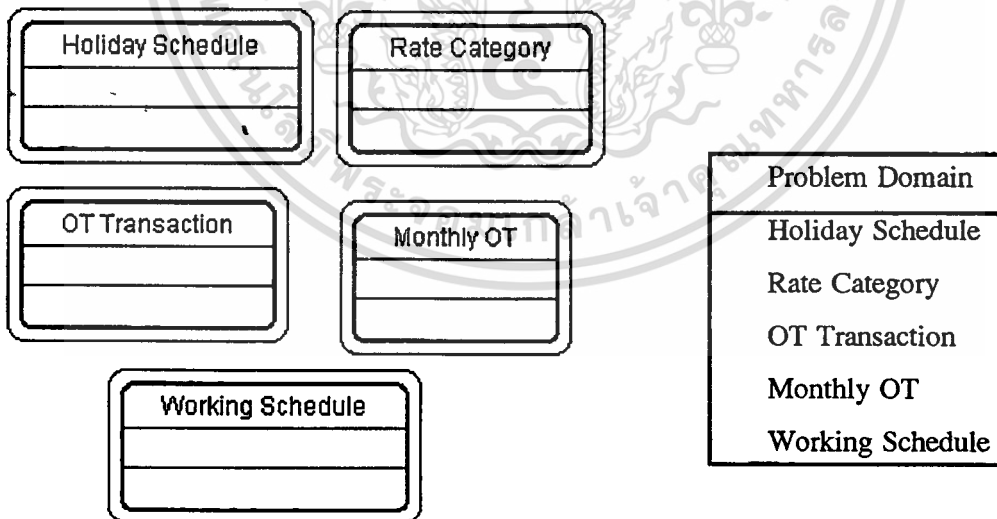
สถานที่ คือสถานที่ที่มี Object อยู่ ซึ่งได้แก่ หน่วยงานสังกัด (Department)



รูปที่ 3.4 สรุป Object และ PD ของ สถานที่

สิ่งของ

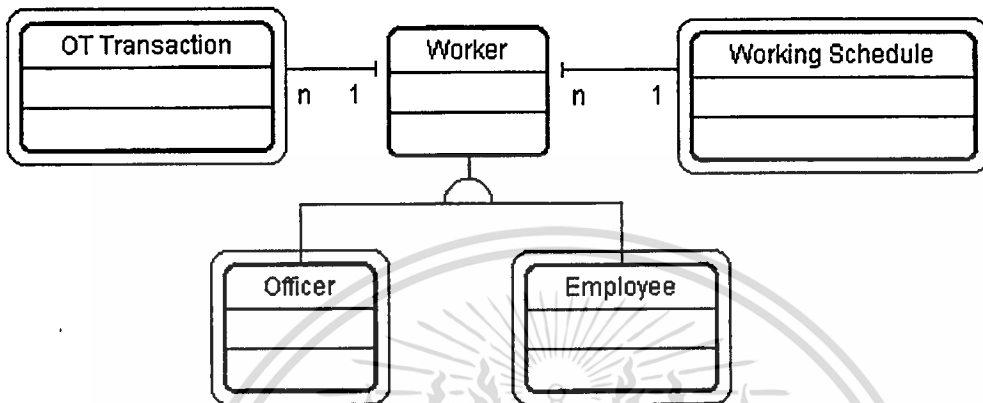
- สิ่งของที่มีตัวตนที่ใช้ในระบบ ซึ่งได้แก่ ตารางวันหยุด (Holiday Schedule), ตารางเวลาการทำงาน (Working Schedule)
- สิ่งของที่เก็บข้อมูลเกี่ยวกับตัวเลข ที่ถูกนำมาใช้ในระบบ ซึ่งได้แก่ อัตราค่าล่วงเวลา (Rate Category)
- เหตุการณ์หรือรายการที่เกิดขึ้น (Transaction) ที่สนใจในระบบค่าล่วงเวลา ซึ่งได้แก่ รายการค่าล่วงเวลา(OT transaction), ค่าล่วงเวลารายเดือน (Monthly OT)



รูปที่ 3.5 สรุป Object และ PD ของ สิ่งของ

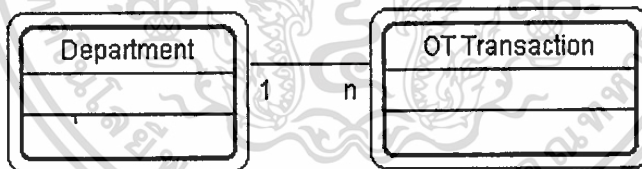
1.2 พิจารณาความสัมพันธ์ของแต่ละ PD Object เพื่อตรวจสอบความครบถ้วนถูกต้อง ของ Object ในระบบ

ผู้กระทำ กับ Transaction



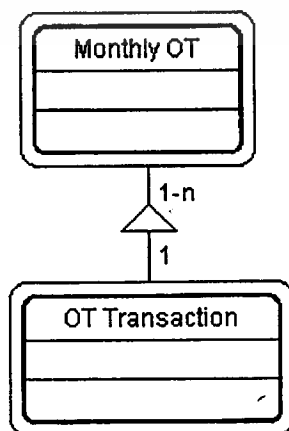
รูปที่ 3.6 ความสัมพันธ์ระหว่างผู้กระทำกับ Transaction

สถานที่ กับ Transaction



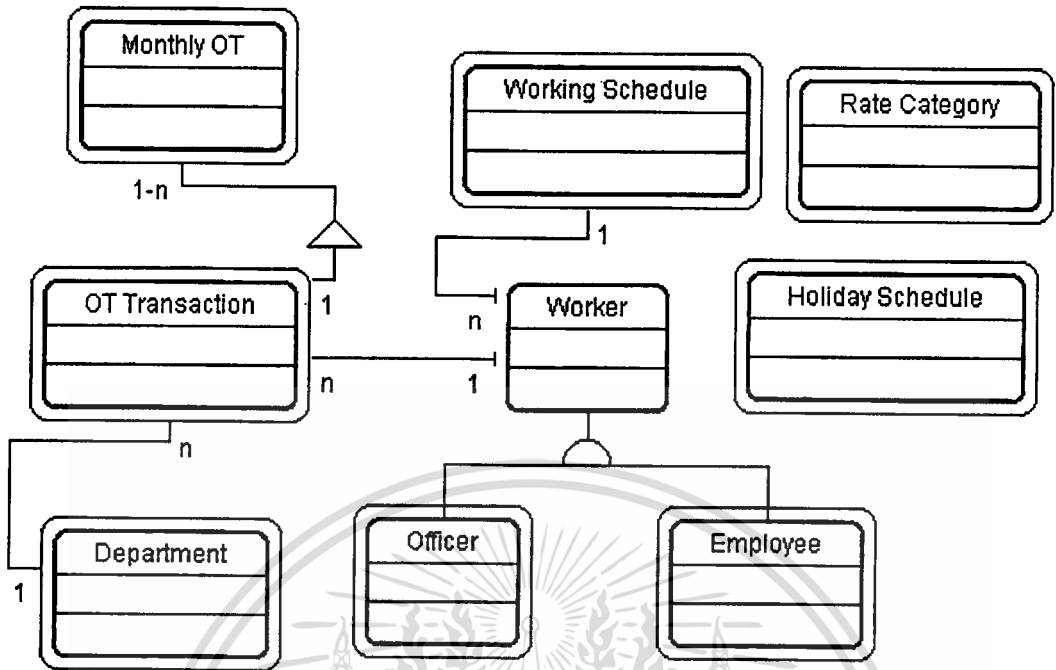
รูปที่ 3.7 ความสัมพันธ์ระหว่างสถานที่กับ Transaction

Transaction กับ Transaction



รูปที่ 3.8 ความสัมพันธ์ระหว่าง Transaction กับ Transaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อผู้ดูแลเห็นาใบเซอร์เบียนด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงภาพรวม Object และความสัมพันธ์ระหว่างกัน

1.3 วิเคราะห์ความรับผิดชอบของแต่ละ PD Object ใน 3 หัวข้อดังนี้

- “what I know” หมายถึง Object รู้จักอะไรบ้างเกี่ยวกับ Object นั้น ซึ่งก็คือ Attribute ของ Object นั้นเอง
- “who I know” หมายถึง Object รู้จัก Object ไตบ้าง
- “what I do” หมายถึง Object ทำอะไรบ้าง ทั้งที่เกี่ยวกับตัวเองและ Object อื่น ซึ่งก็คือ ส่วนที่เป็น Service ของ Object นั้นเอง

ผู้กระทำและผู้มีส่วนร่วม ซึ่งได้แก่ Worker, Employee, Officer

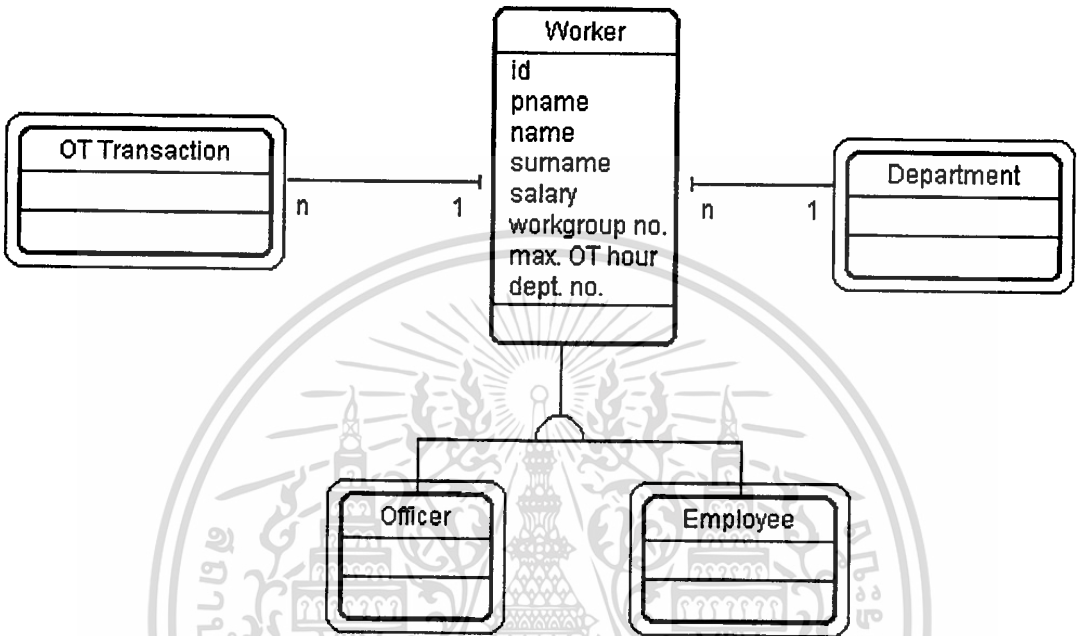
- “what I know”

หมายเลขผู้ปฏิบัติงาน	(id)
คำนำหน้า	(pname)
ชื่อ	(name)
นามสกุล	(surname)
เงินเดือน	(salary)
กลุ่มทำงาน	(workgroup no.)
จำนวนชม. ล่วงเวลาสูงสุด	(max. OT hour)
รหัสสังกัด	(dept. no.)

- “who I know” : Department และ OT Transaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- “what I do” : ซึ่งได้แก่ งานพื้นฐานของทุก Object ที่กระทำกับ Object ของตัวเอง และจะไม่นำไปแสดง ในส่วนที่เป็น service ของ Object เพราะถือว่ามีโดยนัย ใน Object ทุกตัว ซึ่งมีการทำงานดังต่อไปนี้ คือ เพิ่ม, แก้ไข, ลบ, ค้นหา, แรก, ก่อนหน้า, ถัดไป, สุดท้าย



รูปที่ 3.10 แสดง Attribute และความสัมพันธ์ ของ Object ผู้กระทำ

สถานที่ (Department)

- “what I know”

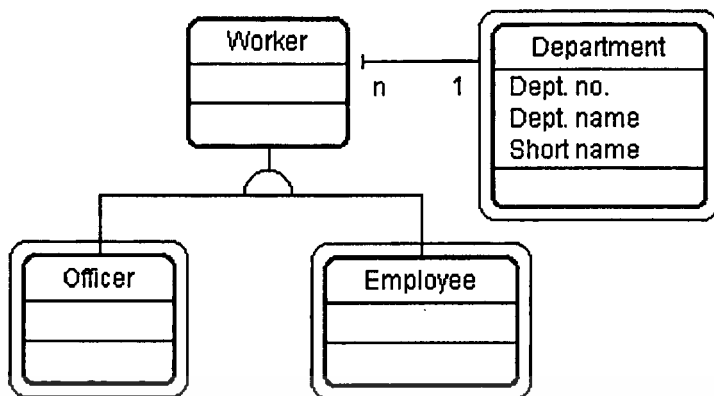
รหัสสังกัด (Dept. no.)

ชื่อสังกัด (Dept. name)

ชื่อย่อสังกัด (Short name)

- “who I know” : Worker

- “what I do” : ซึ่งได้แก่ งานพื้นฐาน



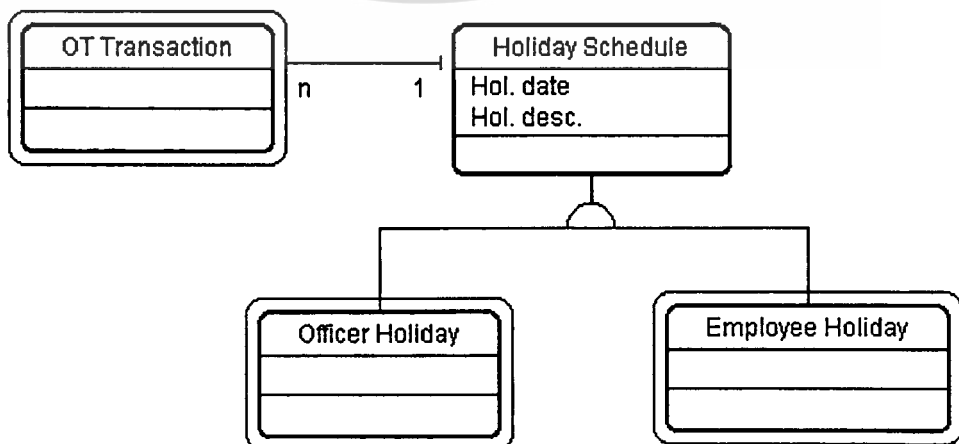
รูปที่ 3.11 แสดง Attribute และความสัมพันธ์ ของ Object สถานที่

สิ่งของ

1. สิ่งของที่มีตัวตน

- Holiday Schedule เนื่องจาก ตารางวันหยุดของพนักงานและลูกจ้าง จะไม่เหมือนกัน ดังนั้น ควรจะแยกเป็นคนละ Object โดยมี Class Holiday Schedule ซึ่งมี Attribute ที่ใช้ร่วมกัน และสามารถ Inherit มาให้กับ Object ชื่อ Employee Holiday และ Officer Holiday ได้

- “what I know”
 - วันหยุด (Hol. Date)
 - ชื่อวันหยุด (Hol. Desc.)
- “who I know” : OT Transaction
- “what I do” : ซึ่งได้แก่ งานพื้นฐาน



รูปที่ 3.12 แสดง Attribute และความสัมพันธ์ ของ Object ชื่อ Holiday Schedule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Working Schedule

- “what I know”

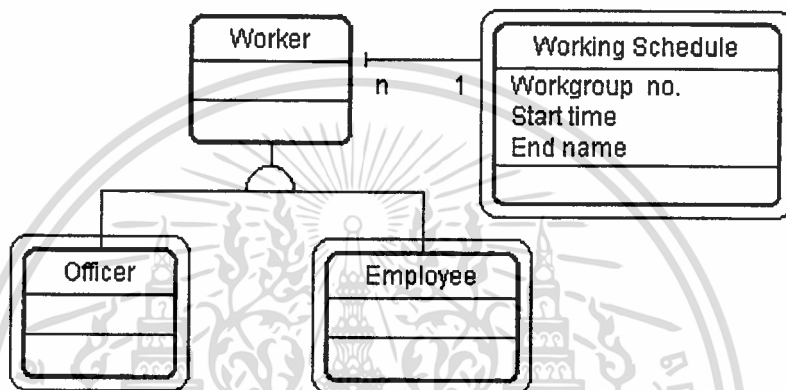
รหัสกลุ่มทำงาน (Workgroup no.)

เวลาเริ่มงาน (Start time)

เวลาเลิกงาน (End time)

- “who I know” : Worker

- “what I do” : ซึ่งได้แก่ งานพื้นฐาน



รูปที่ 3.13 แสดง Attribute และความสัมพันธ์ ของ Object ชื่อ Working Schedule

2. สิ่งของที่เก็บข้อมูลเกี่ยวกับตัวเลขที่ถูกนำมาใช้ในระบบ (Rate Category)

- “what I know”

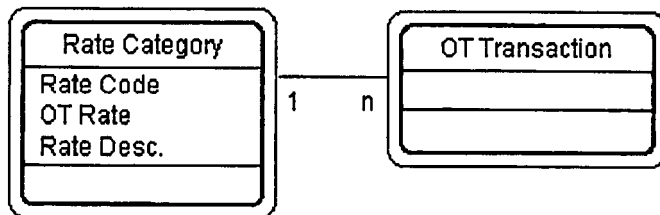
รหัสอัตรา (Rate Code)

อัตราค่าล่วงเวลา (OT Rate)

คำบรรยายค่าล่วงเวลา (Rate Desc.)

- “who I know” : OT Transaction

- “what I do” : ซึ่งได้แก่ งานพื้นฐาน



รูปที่ 3.14 แสดง Attribute และความสัมพันธ์ ของ Object ชื่อ Rate Category

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เหตุการณ์หรือรายการที่เกิดขึ้น

- OT Transaction

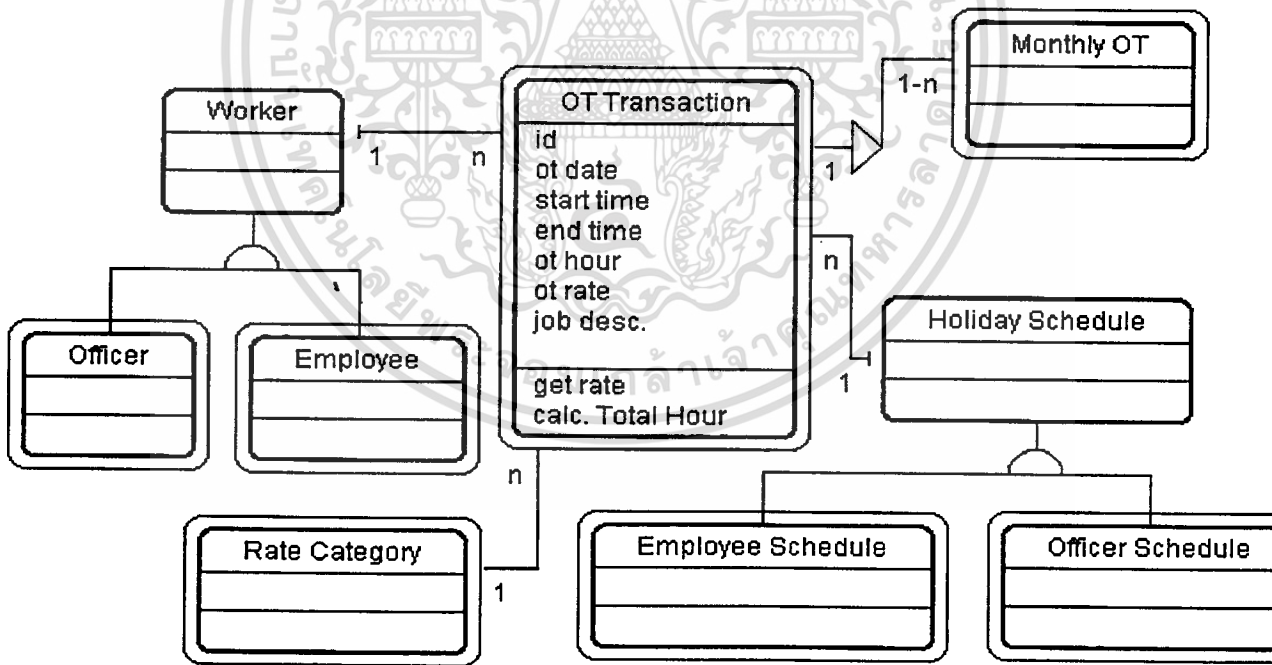
- “what I know”

หมายเลขผู้ปฏิบัติงาน	(Id)
วันที่ทำล่วงเวลา	(OT Date)
เวลาเริ่มต้น	(Start Time)
เวลาสิ้นสุด	(End Time)
ชั่วโมงทำงานล่วงเวลา	(OT Hour)
อัตราค่าล่วงเวลา	(OT Rate)
งานที่ทำ	(Job Desc.)

- “who I know” : Monthly OT, Worker, Rate Category, Holiday Schedule

- “what I do” : ซึ่งได้แก่

1. งานพื้นฐาน
2. หาอัตราค่าล่วงเวลา (get rate)
3. คำนวณจำนวนชั่วโมงทำงานล่วงเวลา (Calc. Total Hour)



รูปที่ 3.15 แสดง Attribute และความสัมพันธ์ ของ Object ชื่อ OT Transaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Monthly OT

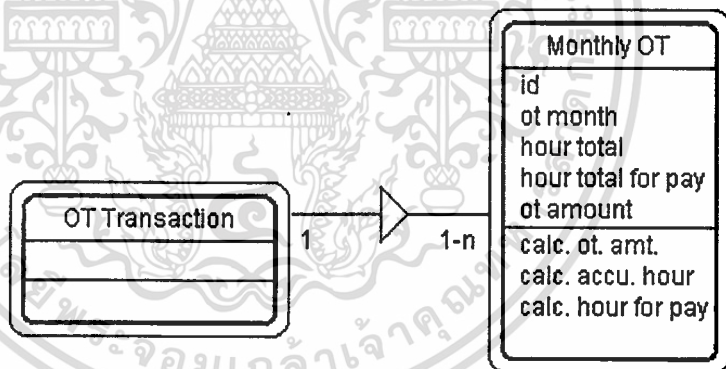
- “what I know”

หมายเลขผู้ปฏิบัติงาน	(Id)
เดือนปี ของข้อมูล	(OT Month)
จำนวนช.ม. รวม	(Hour Total)
จำนวนช.ม. คิดเงิน	(Hour Total for Pay)
จำนวนเงินค่าล่วงเวลา	(OT Amount)

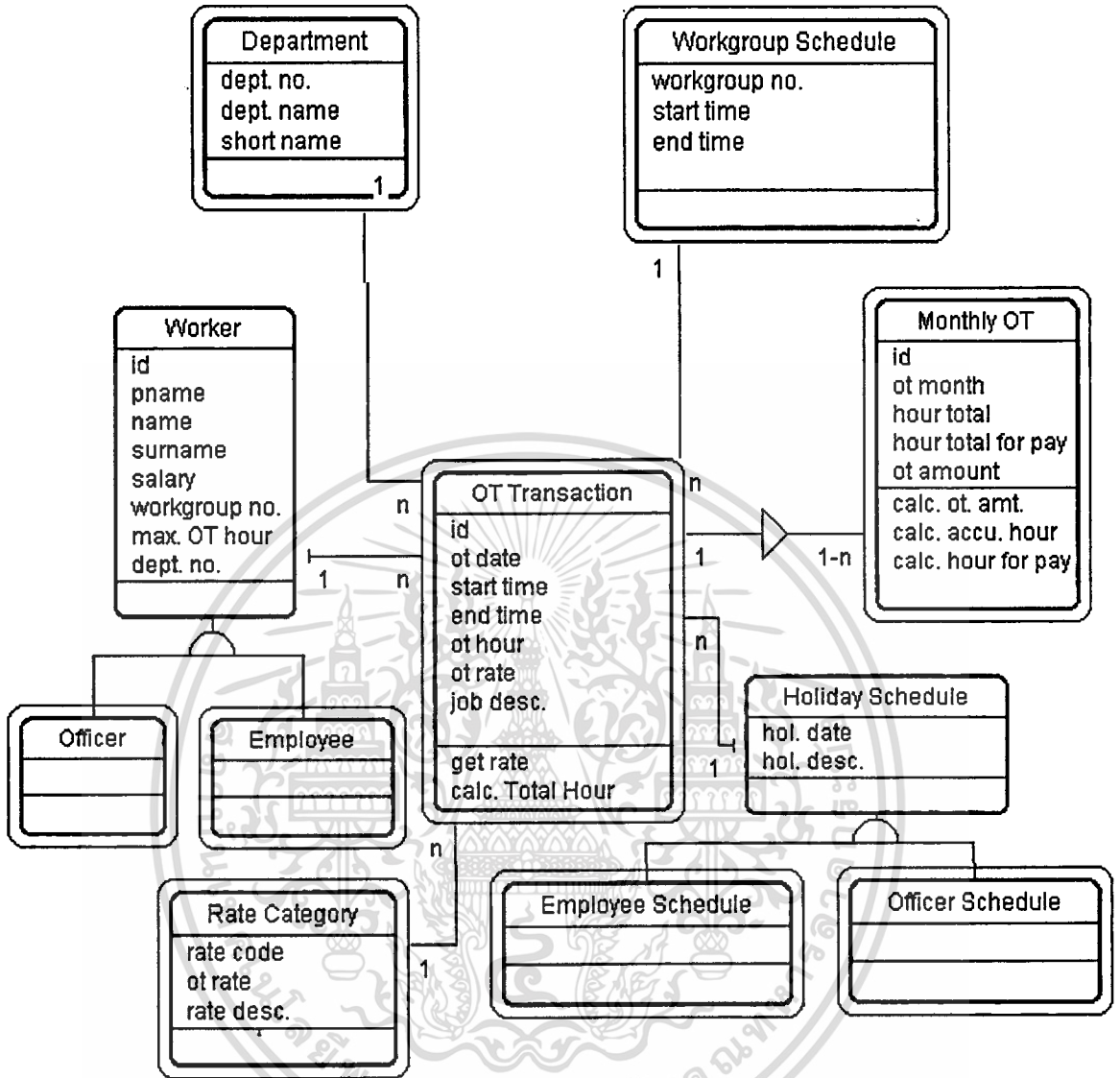
- “who I know” : OT Transaction

- “what I do” : ซึ่งได้แก่

1. งานพื้นฐาน
2. คำนวณจำนวนเงินล่วงเวลา (Calc. Ot. Amt.)
3. คำนวณชั่วโมงสะสมล่วงเวลา ซึ่งไม่คำนึงถึงอัตราค่าล่วงเวลา (Calc. Accu. Hour)
4. คำนวณชั่วโมงคิดเงิน ซึ่งคำนึงถึงอัตราค่าล่วงเวลา เพื่อนำไปคิดเงินค่าล่วงเวลา (Calc. Hour for Pay)



รูปที่ 3.16 แสดง Attribute และความสัมพันธ์ ของ Object ชื่อ Monthly OT



รูปที่ 3.17 แสดง Attribute และความสัมพันธ์ ของ Object ทั้งหมด ในระบบ

1.4 วิเคราะห์การทำงานของ PD Object ในระบบ

สาเหตุในการวิเคราะห์

- เพื่อหา Object เพิ่มเติม (ถ้ามี)
- เพื่อถ่วงน้ำหนักความรับผิดชอบ
- เพื่อเข้าใจการทำงานของระบบ
- เพื่อให้เกิดความสมบูรณ์ ของ Object Model
- เพื่อทดสอบ Object Model

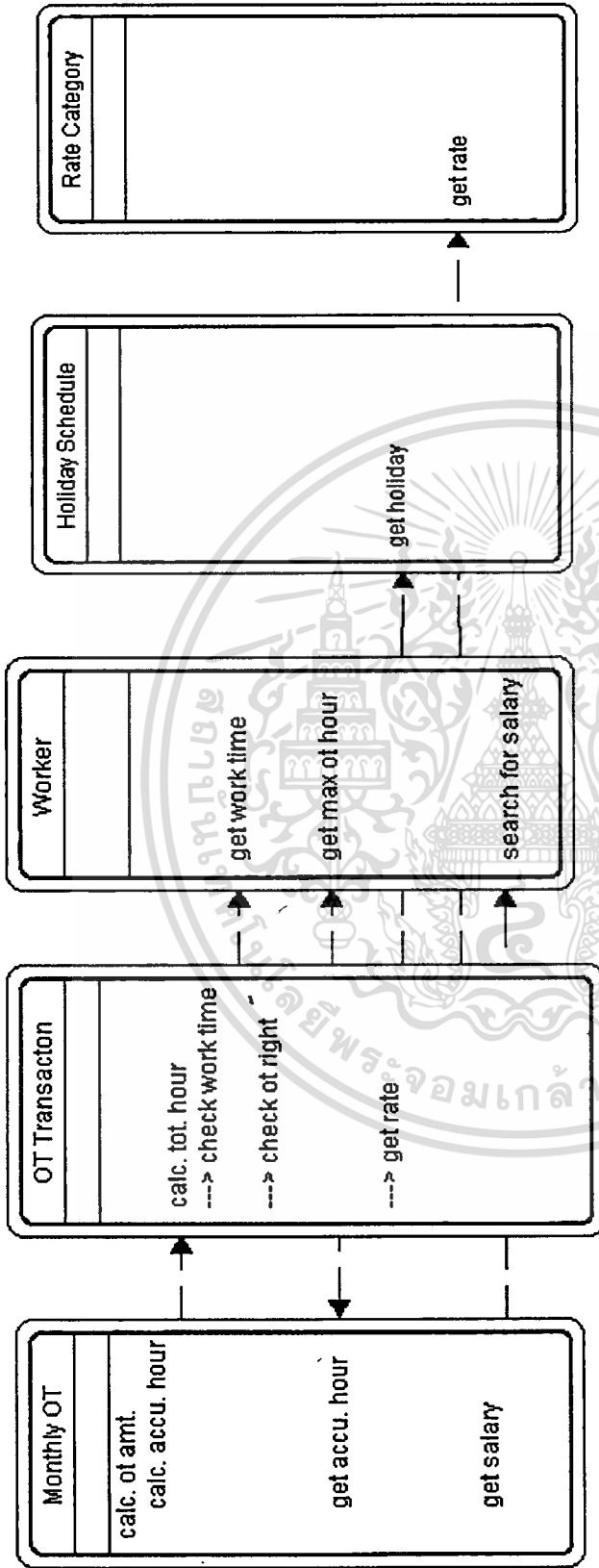
การวิเคราะห์ ควรจะเลือกการทำงาน ในส่วนที่มีความสำคัญของระบบนี้ ขึ้นมา ซึ่ง

ได้แก่ การคำนวณค่าล่วงเวลาเป็นต้น

จากรูปที่ 3.18 นี้ จะเห็นว่าเมื่อวิเคราะห์ทำให้เห็น Service ที่จำเป็นของระบบเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1.8 แสดงการวิเคราะห์การทำงานของ PD Object ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.18 นำส่วนของ Service มาเขียนเป็น method พร้อม Message ที่จะส่งไปยัง Object ต่าง ๆ ได้ดังนี้

1. Montlyly OT Object

- CalcOtAmt(;amount)
- CalcAccuHour(;hour total, hour total for pay)
- GetAccuHour(id;hour total)
- GetSalary(;salary)

2. OT Transaction Object

- CalcTotHour(id;hour total)
- CheckWorkTime(start time,end time;result)
- CheckOtRight(result)
- GetRate(;Ot rate)

3. Worker Object

- GetWorkTime(id;start time,end time)
- GetMaxOtHour(id;max ot hour)
- SearchForSalary(id;salary)

4. Holiday Schedule

- GetHoliday(Ot date;type of date)

5. Rate Category

- GetRate(Ot date,id,type of date;ot rate)

2. Human Interaction (HI) มี 5 ขั้นตอนคือ

2.1 วิเคราะห์หา Windows ที่ใช้ในระบบ

2.2 วิเคราะห์หา Report ที่ใช้ในระบบ

2.3 วิเคราะห์หา Windows ที่ใช้เป็นเมนู

2.4 พิจารณาความรับผิดชอบของแต่ละ HI Object

2.5 วิเคราะห์การทำงานของ HI Object ที่มีลักษณะสำคัญ

2.1 วิเคราะห์หา Windows ที่ใช้ในระบบ มีขั้นตอนดังนี้

1. เพิ่ม Windows ของแต่ละ Object ที่ได้ออกแบบไว้ในข้อ 1 โดยไม่รวมส่วนที่เป็น Transaction Object ตามรายละเอียดข้างล่างนี้

เลขที่	ชื่อของ Window	คำอธิบาย
1	FormEmployee	Window บันทึกลูกจ้าง
2	FormOfficer	Window บันทึกพนักงาน
3	FormWorkGroup	Window บันทึกตารางเวลาทำงาน
4	FormRate	Window บันทึกอัตรา
5	FormHolEmp	Window บันทึกวันหยุดลูกจ้าง
6	FormHolOffi	Window บันทึกวันหยุดพนักงาน
7	FormDept	Window บันทึกสังกัด

ตารางที่ 2.1 Windows จากการออกแบบในขั้นตอน PD ยกเว้น Transaction Object

2. เพิ่ม Window ส่วนที่เหลือที่ไม่ได้เกิดจากการออกแบบในขั้นตอน PD
- เพิ่ม Logon Window เมื่อระบบต้องการกำหนดสิทธิ์การใช้งาน
 - เพิ่ม System Setup Window เพื่อสร้าง User และสิทธิการใช้งาน
 - เพิ่ม Window ที่เกี่ยวข้องกับการทำให้ระบบค่าล่วงเวลา สอดคล้องกับความต้องการของผู้ใช้ ซึ่งได้แก่ Overtime Entry Window
 - พิจารณาสິงที่จำเป็นต้องเพิ่มเติมในระบบ เพื่อให้การใช้ระบบงาน มีความสะดวกและรวดเร็ว ซึ่งได้แก่การขึ้นเดือนใหม่ เพื่อให้บันทึกค่าล่วงเวลา ได้สะดวก และรวดเร็ว โดยบันทึกเฉพาะวันที่ ส่วนเดือนและปีจะเปลี่ยนเมื่อขึ้นเดือนใหม่ เพียงครั้งเดียว

เลขที่	ชื่อของ Window	คำอธิบาย
8	FormLogin	Window สำหรับ Login เข้าระบบ
9	FormClerk	Window บันทึกผู้ใช้งานในระบบ
10	FormOtTrans	Window บันทึกค่าล่วงเวลา
11	FormNewMonth	Window สำหรับขึ้นเดือนใหม่ เพื่อให้บันทึกค่าล่วงเวลา ได้สะดวก และรวดเร็ว โดยบันทึกเฉพาะวันที่ ส่วนเดือนและปีจะเปลี่ยนเมื่อขึ้นเดือนใหม่ เพียงครั้งเดียว ด้วย Window นี้

ตารางที่ 2.2 Windows ส่วนที่ต้องใช้งานเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 วิเคราะห์หา Report ที่ใช้ในระบบ

1. พิจารณาความต้องการของบุคคลที่เกี่ยวข้องกับระบบ ว่าต้องการรายงานอะไรบ้าง ซึ่งในระบบนี้ ผู้ปฏิบัติงานและหัวหน้าหน่วยงาน ต้องการรายงานเพื่อตรวจสอบการทำงานล่วงเวลาแบบรายวัน ก่อนสรุปเป็นรายเดือน ดังนั้นจะมีรายงานเพิ่มอีก 1 รายงาน คือ รายงานค่าล่วงเวลาแบบรายวันแยกตามบุคคล
2. พิจารณา Report ที่ระบบนี้ต้องการใช้งาน ซึ่งมี 1 Report คือ รายงาน ลว. 1 ซึ่งเป็นรายงานสรุปค่าล่วงเวลารายเดือนแยกตามหน่วยงาน

เลขที่	ชื่อของ Window	คำอธิบาย
12	FormPrintDailyOt	Window สำหรับเลือกประเภทการออกรายงานรายบุคคล
13	FormRptDailyOt	Window ออกรายงานค่าล่วงเวลารายบุคคล
14	FormRptMonthlyOt	Window ออกรายงานสรุปค่าล่วงเวลา

ตารางที่ 2.3 Windows ที่ได้จากการวิเคราะห์หาความต้องการ Report

2.3 วิเคราะห์หา Windows สำหรับใช้เป็นเมนู

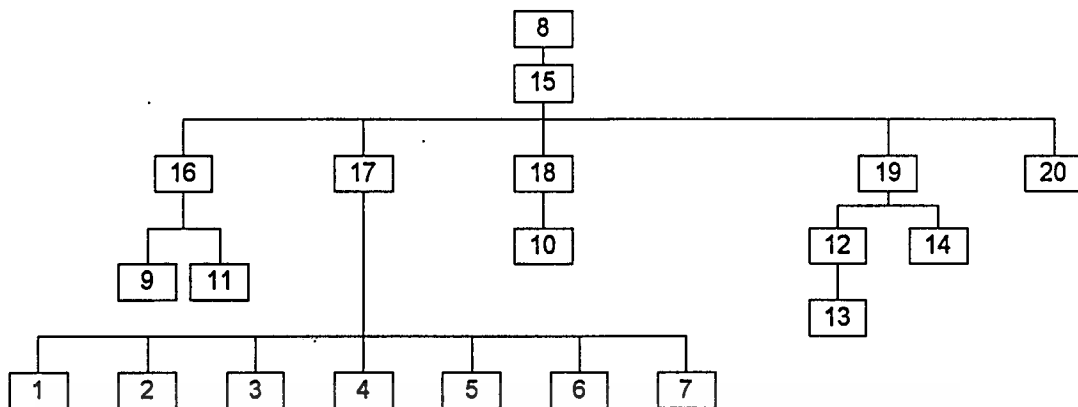
เลขที่	ชื่อของ Window	คำอธิบาย
15	FormMain	เมนูหลักของระบบค่าล่วงเวลา
16	MenuSetup	เมนูย่อยของการกำหนดระบบ
17	MenuMaster	เมนูย่อยของการบันทึกข้อมูลหลัก
18	MenuOt	เมนูย่อยของการบันทึกค่าล่วงเวลา
19	MenuReport	เมนูย่อยของการออกรายงาน
20	MenuLogoff	เมนูย่อยของการออกจากระบบ

ตารางที่ 2.4 Windows สำหรับเป็นเมนู

จากการวิเคราะห์หา HI ข้างต้น สามารถนำ HI ทั้งหมดมาสรุปความสัมพันธ์ได้

ดังรูปที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงโครงสร้าง HI ของระบบค่าล่วงเวลา

2.4 พิจารณาความรับผิดชอบของแต่ละ HI Object

ในการพิจารณาความรับผิดชอบของแต่ละ HI Object จะพิจารณาเฉพาะ Object ที่มีความสำคัญ โดยดู 3 แ่ง เหมือนกับการทำ PD Object คือ “what I know”, “who I know” และ “what I do” ดังมี Object ดังต่อไปนี้

1. บันทึกลูกจ้าง

- “what I know” มักจะประกอบด้วย field ที่ใช้ในการค้นหา และ field ที่เป็น Data Entry ซึ่งได้แก่ หมายเลขผู้ปฏิบัติงาน, คำนำหน้า, ชื่อ, นามสกุล, เงินเดือน, กลุ่มเข้างาน, จำนวน ชม. ล่วงเวลาสูงสุด, รหัสสังกัด
- “who I know” : Workgroup Schedule, Department
- “what I do” : 1. งานพื้นฐานของทุก Object ที่กระทำกับ Object ของตัวเอง ซึ่งได้แก่ เพิ่ม, แก้ไข, ลบ, แสดง, แรก, ก่อนหน้า, ถัดไป และสุดท้าย
- 2. เก็บข้อมูลของ Employee (Commit)
- 3. ยกเลิกการทำงาน กับ Employee (Cancel)
- 4. กลับสู่เมนู (Close)

FormEmployee
id
pname
name
surname
salary
workgroup no.
max. OT hour
dept. no.
Basic Job
Commit
Cancel
Close

รูปที่ 3.20 แสดง FormEmployee Object

2. บันทึกพนักงาน

- “what I know” : หมายเลขผู้ปฏิบัติงาน, คำนำหน้า, ชื่อ, นามสกุล, เงินเดือน, กลุ่ม
เข้างาน, จำนวน ชม. ล่วงเวลาสูงสุด, รหัสสังกัด
- “who I know” : Workgroup Schedule, Department
- “what I do” : 1. งานพื้นฐาน
2. เก็บข้อมูลของ Officer (Commit)
3. ยกเลิกการทำงาน กับ Officer (Cancel)
4. กลับสู่เมนู (Close)

FormOfficer
id
pname
name
surname
salary
workgroup no.
max. OT hour
dept. no.
Basic Job
Commit
Cancel
Close

รูปที่ 3.21 แสดง FormOfficer Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. บันทึกตารางเวลาการทำงาน

- “what I know” : รหัสกลุ่มเข้างาน, เวลาเริ่มงาน, เวลาเลิกงาน
- “who I know” : ไม่มี
- “what I do” : 1. งานพื้นฐาน
 2. เก็บข้อมูลของ Workgroup Schedule (Commit)
 3. ยกเลิกการทำงาน กับ Workgroup Schedule (Cancel)
 4. กลับสู่เมนู (Close)

FormWorkgroup
workgroup no.
start time
end time
Basic Job
Commit
Cancel
Close

รูปที่ 3.22 แสดง FormWorkGroup Object

4. บันทึกอัตรา

- “what I know” : รหัสอัตรา, อัตราค่าล่วงเวลา, ค่าบรรยาย
- “who I know” : ไม่มี
- “what I do” : 1. งานพื้นฐาน
 2. เก็บข้อมูลของ Rate Category (Commit)
 3. ยกเลิกการทำงาน กับ Rate Category (Cancel)
 4. กลับสู่เมนู (Close)

FormRate
rate code
ot rate
rate desc.
Basic Job
Commit
Cancel
Close

รูปที่ 3.23 แสดง FormRate Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. บันทึกวันหยุดลูกจ้าง

- “what I know” : วัน เดือน ปี ของวันหยุด และ คำบรรยายของวันหยุด
- “who I know” : ไม่มี
- “what I do” : 1. งานพื้นฐาน
 2. เก็บข้อมูลของ Employee Schedule (Commit)
 3. ยกเลิกการทำงาน กับ Employee Schedule (Cancel)
 4. กลับสู่เมนู (Close)

FormHolEmp
hol. date
hol. desc.
Basic Job
Commit
Cancel
Close

รูปที่ 3.24 แสดง FormHolEmp Object

6. บันทึกวันหยุดพนักงาน

- “what I know” : วัน เดือน ปี ของวันหยุด และ คำบรรยายของวันหยุด
- “who I know” : ไม่มี
- “what I do” : 1. งานพื้นฐาน
 2. เก็บข้อมูลของ Officer Schedule (Commit)
 3. ยกเลิกการทำงาน กับ Officer Schedule (Cancel)
 4. กลับสู่เมนู (Close)

FormHolOffi
hol. date
hol. desc.
Basic Job
Commit
Cancel
Close

รูปที่ 3.25 แสดง FormHolOffi Object

7. บันทึกสังกัด

- “what I know” : รหัสสังกัด, ชื่อสังกัด, ชื่อย่อสังกัด
- “who I know” : ไม่มี
- “what I do” : 1. งานพื้นฐาน
 2. เก็บข้อมูลของ Department (Commit)
 3. ยกเลิกการทำงาน กับ Department (Cancel)
 4. กลับสู่เมนู (Close)

FormDept
dept. no.
dept. name
short name
Basic Job
Commit
Cancel
Close

รูปที่ 3.26 แสดง FormDept Object

8. เข้าสู่ระบบ

- “what I know” : ชื่อพนักงานธุรการ, รหัสผ่าน
- “who I know” : Clerk
- “what I do” : 1. logon (ตรวจสอบสิทธิ์การใช้งาน)
 2. ยกเลิกการทำงาน (Cancel)

Logon Window
clerk name
password
logon
cancel

รูปที่ 3.27 แสดง Logon Window Object

9. บันทึกผู้ใช้งานในระบบ

- “what I know” : ชื่อผู้ใช้งาน, รหัสผ่าน
- “who I know” : ไม่มี
- “what I do” : 1. งานพื้นฐาน

2. เก็บข้อมูลของ Clerk (Commit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3. ยกเลิกการทำงาน กับ Clerk (Cancel)
- 4. กลับสู่เมนู (Close)

FormClerk
clerk name
clerk password
Basic Job
Commit
Cancel
Close

รูปที่ 3.28 แสดง FormClerk Object

10. บันทึกค่าล่วงเวลา

- “what I know” : รหัสผู้ปฏิบัติงาน, วัน เดือน ปี ที่ทำงานล่วงเวลา, เวลาเริ่มต้นทำล่วงเวลา, เวลาสิ้นสุดทำล่วงเวลา, จำนวนชม ล่วงเวลา, อัตราค่าล่วงเวลา, งานที่ทำล่วงเวลา
- “who I know” : Monthly OT, Worker, Holiday Schedule, Department, Workgroup Schedule, Rate Category
- “what I do” : 1. งานพื้นฐาน
2. เก็บข้อมูล ของ OT Transaction (Commit)
3. ยกเลิกการทำงานกับ OT Transaction (Cancel)
4. กลับสู่เมนู (Close)

FormOTTrans
id
ot date
start time
end time
ot hour
ot rate
job desc.
Basic Job
Commit
Cancel
Close

รูปที่ 3.29 แสดง FormOtTrans Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ขึ้นเดือนใหม่

- “what I know” : ไม่มี
- “who I know” : MonthStatus
- “what I do” : 1. Update ข้อมูลของ MonthStatus
2. ยกเลิกการทำงานกับ MonthStatus (Cancel)

FormNewMonth
Update Cancel

รูปที่ 3.30 แสดง FormNewMonth Object

12. เลือกประเภทการพิมพ์รายงานค่าล่วงเวลารายบุคคล

- “what I know” : รหัสผู้ปฏิบัติงาน
- “who I know” : Worker
- “what I do” : 1. กำหนดเงื่อนไขในการพิมพ์
2. ยกเลิกการพิมพ์ (Cancel)

FormPrintDailyOt
id
Set Printing Condition Cancel

รูปที่ 3.31 แสดง FormPrintTrans Object

13. ออกรายงานค่าล่วงเวลารายบุคคล

- “what I know” : รหัสผู้ปฏิบัติงาน, ชื่อผู้ปฏิบัติงาน, วันที่ทำงานล่วงเวลา, เวลาเริ่มต้นทำล่วงเวลา, เวลาสิ้นสุดทำล่วงเวลา, จำนวนชม ล่วงเวลา, อัตราค่าล่วงเวลา, งานที่ทำล่วงเวลา
- “who I know” : Ot Transaction, Worker, Department
- “what I do” : 1. ออกรายงานรายบุคคล

FormRptDailyOt
id
pname
name
surname
ot date
start time
end time
ot hour
ot rate
job desc.
Print Report

รูปที่ 3.32 แสดง FormRptDailyOt Object

14. ออกรายงานสรุปค่าล่วงเวลา

- “what I know” : รหัสผู้ปฏิบัติงาน, ชื่อผู้ปฏิบัติงาน, สังกัดผู้ปฏิบัติงาน, วันที่ทำงาน
ล่วงเวลา, จำนวนเงินค่าล่วงเวลา
- “who I know” : Monthly OT, Worker, Department
- “what I do” : 1. ออกรายงานสรุปแยกตามสังกัด

FormRptMonthlyOt
id
pname
name
surname
dept. name
ot month
ot amount
Print Report

รูปที่ 3.33 แสดง FormRptMonthlyOt Object

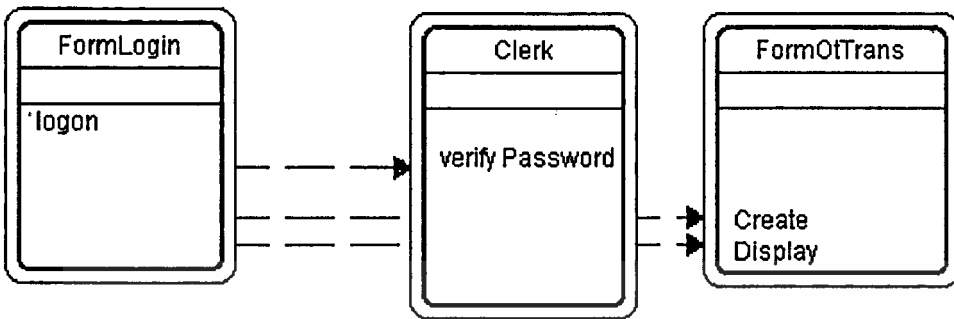
2.5 วิเคราะห์การทำงานของ HI Object ที่มีลักษณะสำคัญ

ขั้นตอนหรือเหตุการณ์ที่สำคัญในระบบค่าล่วงเวลา คือ ขั้นตอนในการสร้างข้อมูลค่า
ล่วงเวลา ซึ่งประกอบด้วยขั้นตอนดังนี้

1. เข้าระบบ
2. บันทึกค่าล่วงเวลา
3. เลิกการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เข้าสู่ระบบ



รูปที่ 3.34 แสดงการวิเคราะห์การ เข้าสู่ระบบ

จากรูปที่ 3.22 สามารถนำส่วนของ Service มาเขียนเป็น method พร้อม Message ที่จะส่งไปยัง Object ต่าง ๆ ได้ดังนี้

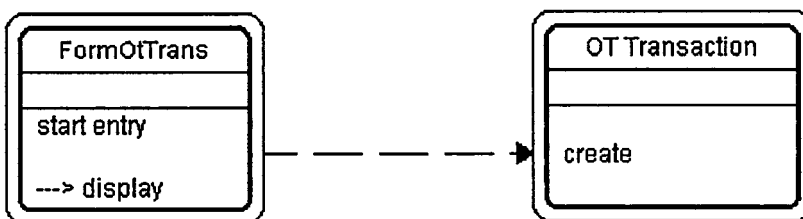
1. FormLogon
 - logon
2. Clerk
 - verifyPassword(clerk name, clerk password; result)
3. FormOtTrans
 - create
 - display

2. บันทึกค่าล่วงเวลา แบ่งออกเป็น 2 เหตุการณ์ดังนี้

2.1 การเริ่มต้นบันทึกค่าล่วงเวลา

2.2 การสรุปค่าล่วงเวลา

2.1 การเริ่มต้นบันทึกค่าล่วงเวลา



รูปที่ 3.35 แสดงการวิเคราะห์การเริ่มต้นบันทึกค่าล่วงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.23 สามารถนำส่วนของ Service มาเขียนเป็น method พร้อม Message ที่จะส่งไปยัง Object ต่าง ๆ ได้ดังนี้

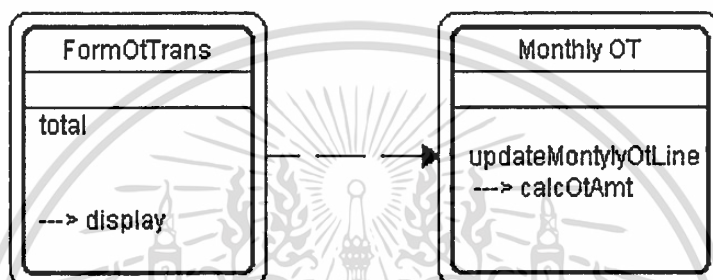
1. FormOtTrans

- Start entry
- display

2. Ot Transaction

- create(OT Transaction)

2.2 การสรุปค่าล่วงเวลา



รูปที่ 3.36 แสดงการวิเคราะห์การสรุปค่าล่วงเวลา

จากรูปที่ 3.24 สามารถนำส่วนของ Service มาเขียนเป็น method พร้อม Message ที่จะส่งไปยัง Object ต่าง ๆ ได้ดังนี้

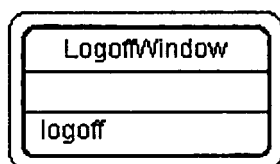
1. FormOtTrans

- total
- display

2. Montly OT

- updateMontlyOtLine
- calcOtAmt(;amount)

3. ออกจากระบบ



รูปที่ 3.37 แสดงการวิเคราะห์การออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Data Management (DM)

จากการวิเคราะห์ระบบข้างต้น จะได้ Object และ Attribute ของระบบทั้งหมด ในการทำขั้นตอน Data Management คือ การนำ PD Object มาทำเป็น DM Object นั้นเอง ถ้า PD Object ใด มี Parent Class ก็จะใช้ Parent Class นั้นมาเป็น DM Object และในแต่ละ DM Object จะมี Attribute เหมือนกับ PD Object ซึ่งมี DM Object ดังต่อไปนี้

3.1 Worker DM

3.2 Clerk DM

3.3 Working Schedule DM

3.4 Rate Category DM

3.5 Holiday Schedule DM

3.6 Department DM

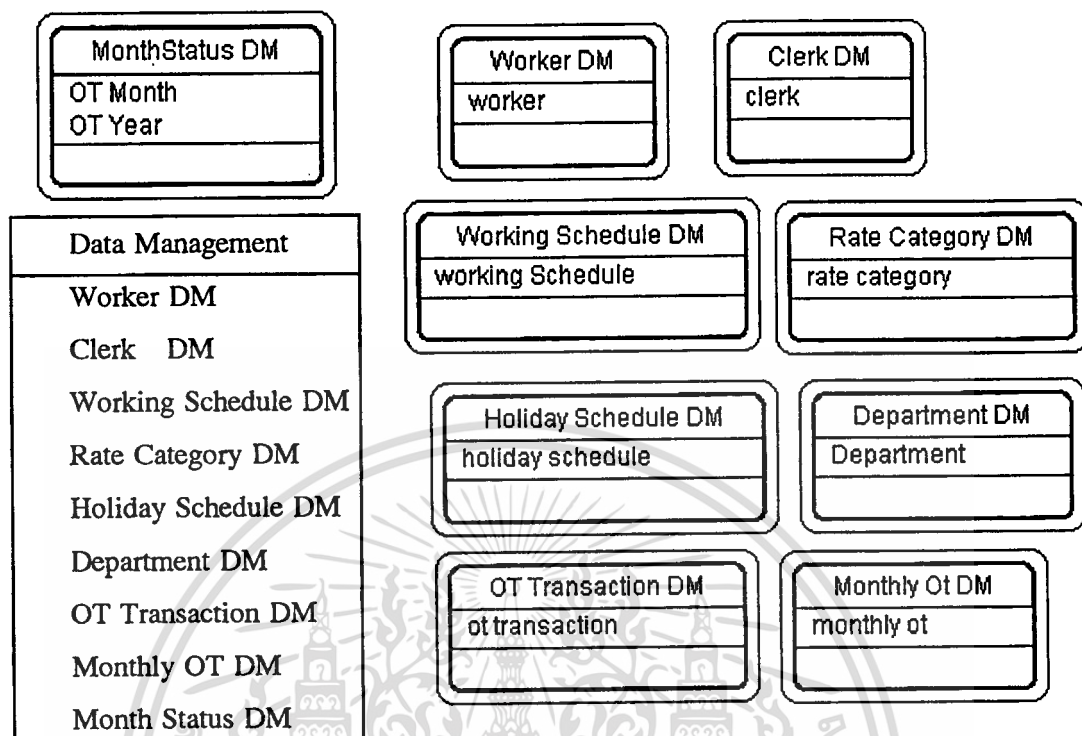
3.7 OT Transaction DM

3.8 Monthly OT DM

และในแต่ละ DM Object จะมี attribute เหมือนกับ PD Object ยกเว้น Worker DM และ Holiday Schedule DM ซึ่งจะมี attribute เพิ่มเติม ดังนี้

- Worker DM เพิ่ม attribute ประเภท(type) เพื่อให้สามารถแยกออกได้ว่าเป็นข้อมูล ของ Object ที่ชื่อว่า Officer หรือ Employee
- Holiday Schedule เพิ่ม attribute ประเภท(type) เหมือนกับ Worker DM เพื่อให้สามารถแยกออกได้ว่าเป็นข้อมูลของ Object ที่ชื่อว่า Officer Holiday หรือ Employee Holiday

นอกจากนี้ ในขั้นตอนของ HI จะเกิด HI Object ชื่อ ขึ้นเดือนใหม่ ซึ่งก่อให้เกิด MonthStatus DM ขึ้น โดยประกอบด้วย attribute เดือนของค่าล่วงเวลา (OT Month) และ ปีของค่าล่วงเวลา (OT Year)



รูปที่ 3.38 Data Management Object

หลังจากเราทำการศึกษาการออกแบบฐานข้อมูลแบบ Object-Oriented (OODB) เนื่องจากการศึกษาครั้งนี้ เราไม่ได้ใช้ Object-Oriented Database Management System(OODBMS) เพราะในปัจจุบัน DBMS ยังไม่สามารถสนับสนุนการจัดการฐานข้อมูลแบบ Object-Oriented ได้ ดังนั้น ผู้ศึกษาจึงจำเป็นต้องใช้ Relational Database Management System(RDBMS) คือ InterBase แทน โดยนำ DM Object มาแปลงเป็น Table(Entity) ดังแสดงในตัวอย่างรูปที่ 3.39 และตารางที่ 3.1 จะแสดงการสรุปรายชื่อ Table ของ DM Object

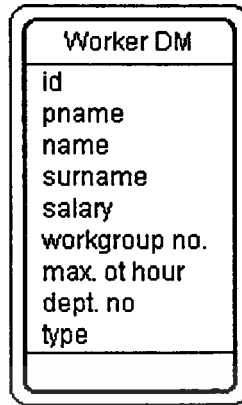


Table : Worker(ผู้ปฏิบัติงาน)

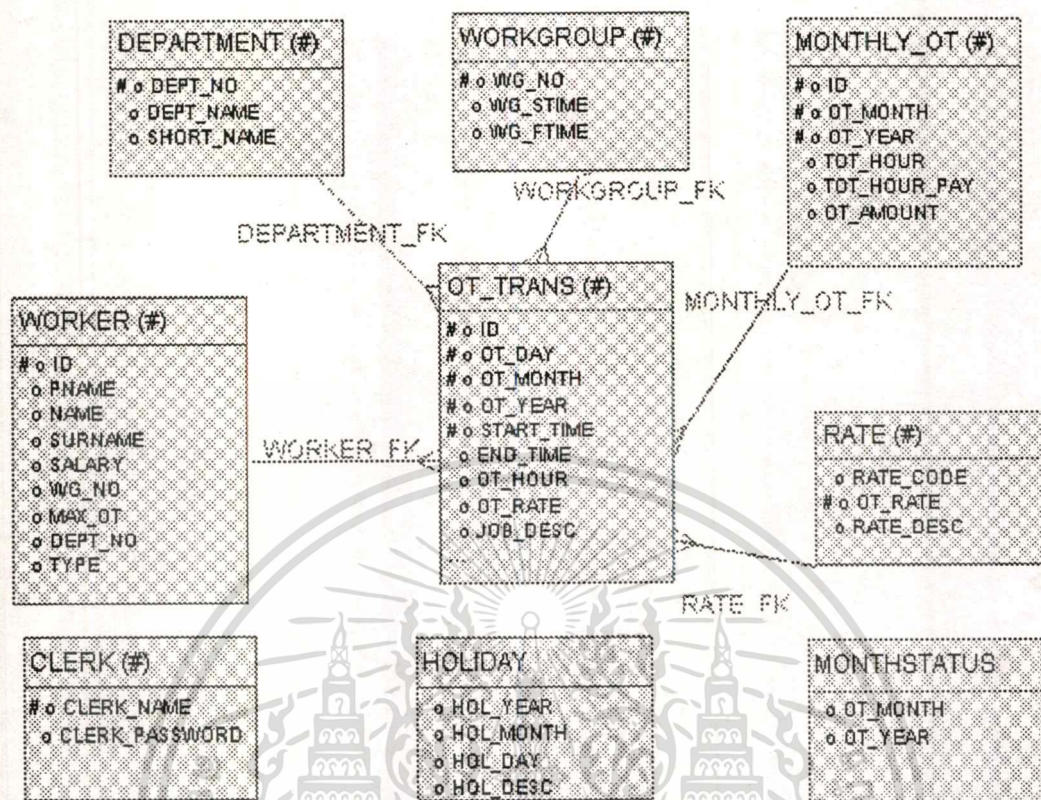
ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
id	หมายเลขผู้ปฏิบัติงาน	char	6	
pname	คำนำหน้า	char	12	
name	ชื่อ	char	15	
surname	นามสกุล	char	20	
salary	เงินเดือน	integer	6	
wg_no	กลุ่มเข้างาน	char	1	
max_ot	จำนวนช.ม ล่วงเวลาสูงสุด	integer	3	
dept_no	รหัสสังกัด	char	7	
type	ประเภทผู้ปฏิบัติงาน	char	1	มีค่า 1 = พนักงาน 2 = ลูกจ้าง

รูปที่ 3.39 แสดงการเปลี่ยน Object เป็น Table

Data Management	ชื่อของ Table
Worker DM	Worker
Clerk DM	Clerk
Working Schedule DM	WorkGroup
Rate Category DM	Rate
Holiday Schedule DM	Holiday
Department DM	Dept
OT Transaction DM	OT_Trans
Monthly OT DM	Monthly_OT
Month Status DM	MonthStatus

ตารางที่ 3.1 แสดงรายชื่อ Table ของแต่ละ Data Management

เมื่อทำการเปลี่ยน Object เป็น Table แล้ว ในขั้นตอนต่อไปคือการสร้างความสัมพันธ์ระหว่าง Table โดยแสดง Table และความสัมพันธ์ระหว่าง Table ในรูปที่ 3.40 ส่วนรายละเอียด Attribute ของ Table ทั้งหมด จะแสดงในภาคผนวก ข



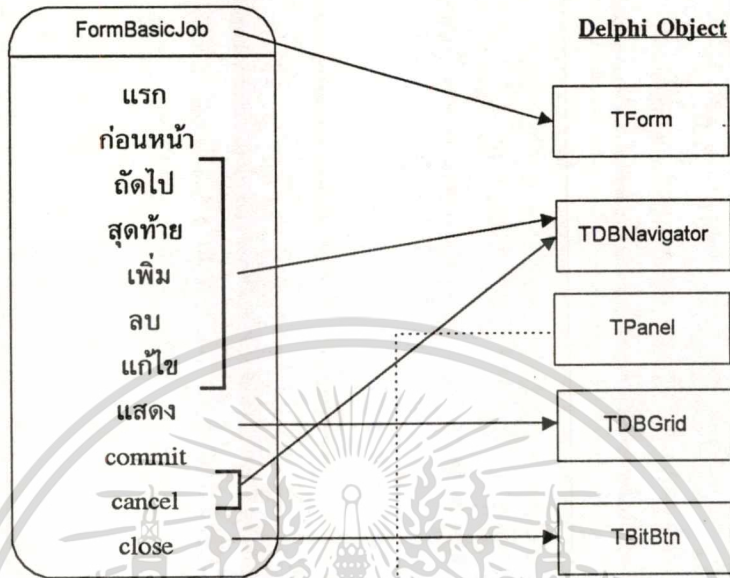
รูปที่ 3.40 แสดง Table และความสัมพันธระหว่าง Table ของระบบค่าล่วงเวลา

3.3 การพัฒนาโปรแกรม

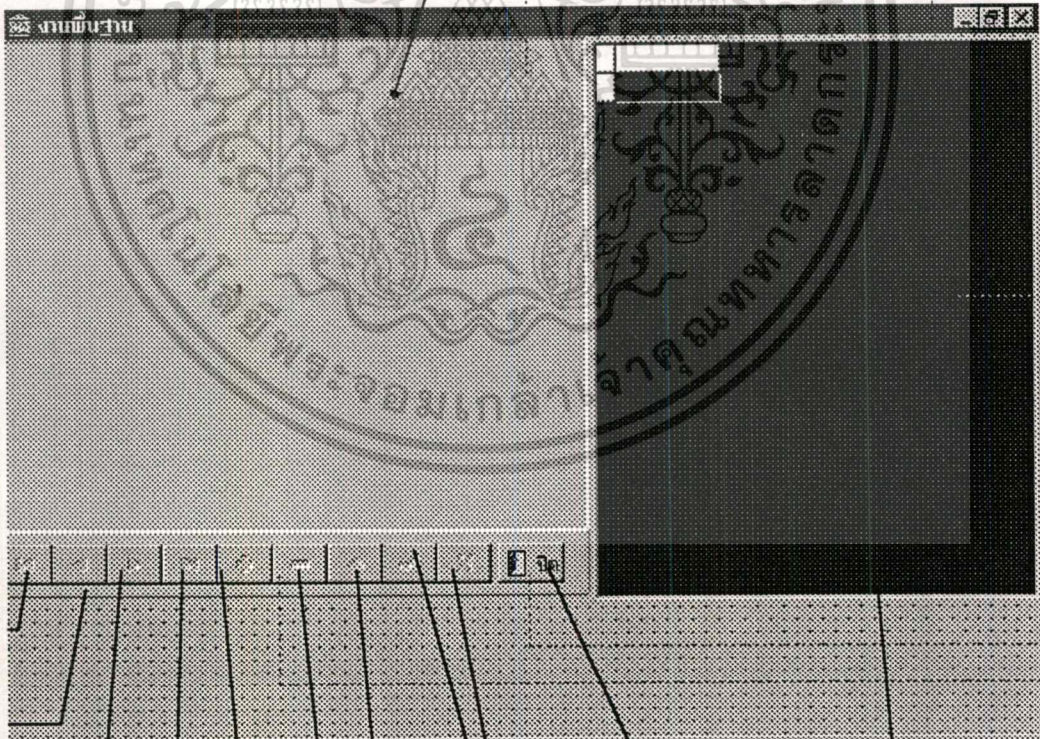
หลังจากออกแบบระบบและออกแบบฐานข้อมูลแล้ว จะได้ Class ,Object ต่าง ๆ ตลอดจน Method ทั้งหมด ขั้นตอนต่อไปคือ การพัฒนาโปรแกรมโดยใช้ OOP โดยกรณีศึกษาที่เราใช้โปรแกรม Delphi ซึ่ง Object แต่ละตัวของระบบ จะถูกสร้างขึ้นโดยใช้ Object ที่มีอยู่แล้วของ Delphi (ใน Delphi บางทีจะเรียก Object เหล่านี้ว่า component), จาก Third party ซึ่งเขียนโดย Ray Konopka และที่ได้สร้างขึ้นเอง นำมาประกอบกันจนกลายเป็น Object ของระบบ

การพัฒนาโปรแกรม มีหลักการหนึ่งที่เป็นสิ่งสำคัญคือ inheritance ดังนั้น ก่อนที่เราจะพัฒนาโปรแกรม ตาม HI Object เราควรพิจารณาส่วนที่มีลักษณะเหมือนกัน ว่ามีอะไรบ้าง จากนั้นสร้างส่วนนั้น เป็น Class ต้นแบบ เพื่อให้ Object อื่น ๆ นำไปใช้พัฒนาได้ง่ายและสะดวกมากยิ่งขึ้น ซึ่งในระบบค่าล่วงเวลานี้ จะสามารถสร้าง Class ต้นแบบ ได้ดังนี้(รูป Object ทั้งหมดที่ใช้ในการพัฒนา จะแสดงไว้ในภาคผนวก คและวิธีการใช้งานของระบบ จะแสดงอยู่ในภาคผนวก ง)

1) FormBasicJob(งานพื้นฐาน) ซึ่งประกอบด้วย เพิ่ม, แก้ไข, ลบ, แสดง, แรก, ก่อนหน้า, ถัดไป และสุดท้าย



(ใช้สร้างพื้นที่เพื่อจัดกลุ่มของ Object)



รายการแรก
รายการก่อนหน้า

รายการถัดไป
รายการสุดท้าย

เพิ่ม
ลบ

แก้ไข

Cancel
Commit

ปิด window

ส่วนแสดงรายการข้อ
ทั้งหมดของ Object

รูปที่ 3.41 แสดงการสร้าง FormBasicJob

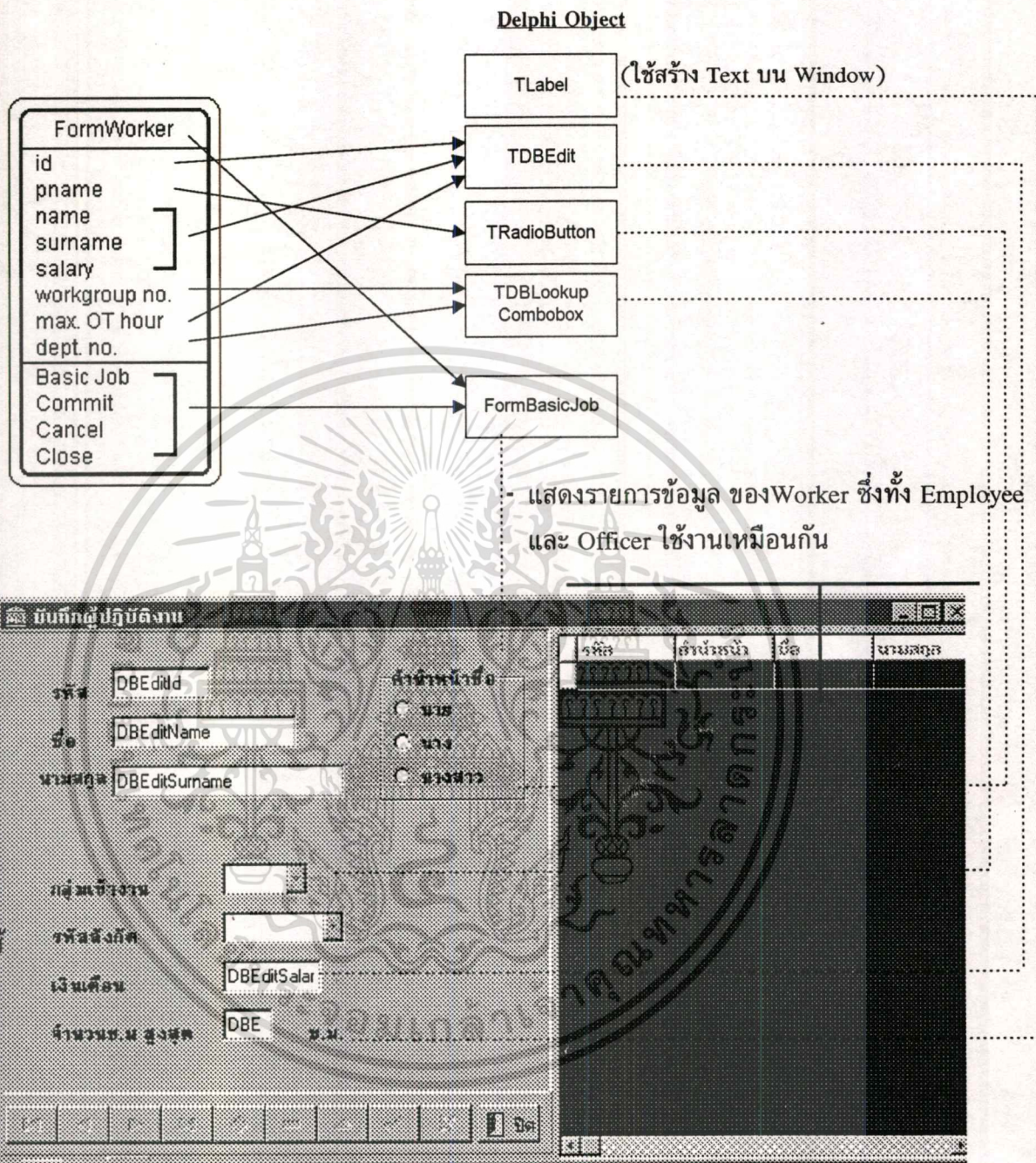
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Class ที่พบในขั้นตอนการออกแบบ Problem Domain(PD)

2.1 FormWorker นำมาสร้างเป็น HI Object ชื่อ FormWorker เพื่อสร้างเป็น Class ต้นแบบให้กับ HI Object ชื่อ FormEmployee (บันทึกลูกจ้าง) และ FormOfficer (บันทึกพนักงาน) โดยมีรายละเอียดดังนี้

1. งานพื้นฐาน ใช้ inherit จาก FormBasicJob Class
2. ชื่อ Attribute ที่จะบันทึกข้อมูล ใช้ Object ของ Delphi ชื่อ TLabel
3. การรับค่าจากการบันทึกข้อมูล ใช้ Object ของ Delphi ชื่อ TEdit
4. การรับค่าของ Attribute เข้าไปที่ Database โดยตรง ใช้ Object ของ Delphi ชื่อ TDBEdit
5. ใช้เพื่อให้สามารถเข้าถึงข้อมูล ของ InterBase ได้ โดยใช้ Non Visual Object ของ Delphi ชื่อ TDataSource
6. ใช้เพื่อให้สามารถเข้าถึงข้อมูล ของ InterBase ได้ และใช้ระบุว่า Table อะไรที่จะใช้งาน โดยใช้ Non Visual Object ของ Delphi ชื่อ TTable

นอกจากนี้ ต้องเขียน Method ต่าง ๆ เพิ่มเติม โดยดูจากการออกแบบไว้ในขั้นตอน Problem Domain แต่สำหรับ Object ที่เป็นการบันทึกข้อมูลหลัก เช่น บันทึกวันหยุดพนักงาน , บันทึกพนักงาน, บันทึกสังกัด ฯลฯ จะไม่มี Method เพิ่มเติม แต่จะมีเฉพาะงานพื้นฐาน ซึ่งใช้ Object ของ Delphi ชื่อ DBNavigation และ DBGrid เป็นตัวจัดการ ในรูปที่ 3.42 จะแสดงการสร้าง FormWorker จากการ inherit Object จาก Delphi



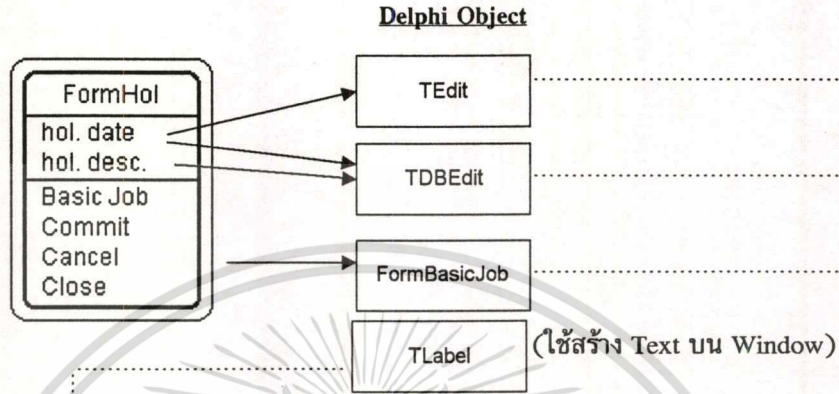
รายการข้อมูล
ของ Worker
ซึ่ง Employee
และ Officer ใช้
ส่วนนี้เหมือน
กัน

รูปที่ 3.42 แสดงการสร้าง FormWorker

2.2 FormHol นำมาสร้างเป็น HI Object ชื่อ FormHol เพื่อสร้างเป็น Class
ต้นแบบ ให้กับ HI Object ชื่อ FormHolEmp (บันทึกวันหยุดลูกจ้าง) และ
FormHolOffi (บันทึกวันหยุดพนักงาน) โดยมีวิธีการสร้างคล้าย FormWorker

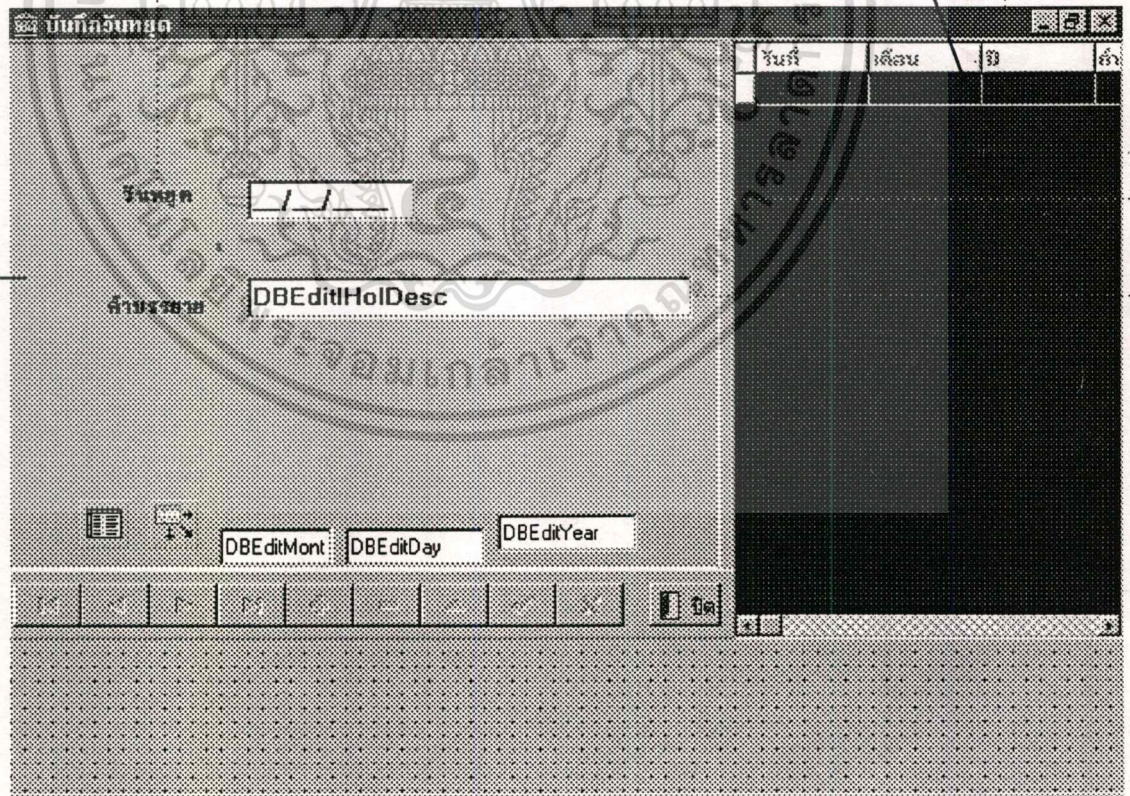
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 FormHol นำมาสร้างเป็น HI Object ชื่อ FormHol เพื่อสร้างเป็น Class ต้นแบบ ให้กับ HI Object ชื่อ FormHolEmp (บันทึกวันหยุดลูกจ้าง) และ FormHolOffi (บันทึกวันหยุดพนักงาน) โดยมีวิธีการสร้างคล้าย FormWorker



แสดงรายการข้อมูลของ Holiday ซึ่ง Employee Holiday และ Officer Holiday จะใช้ข้อมูลส่วนนี้ เหมือนกัน

รายการข้อมูลของ Holiday ซึ่ง Employee Holiday และ Officer Holiday จะใช้ข้อมูลส่วนนี้ เหมือนกัน

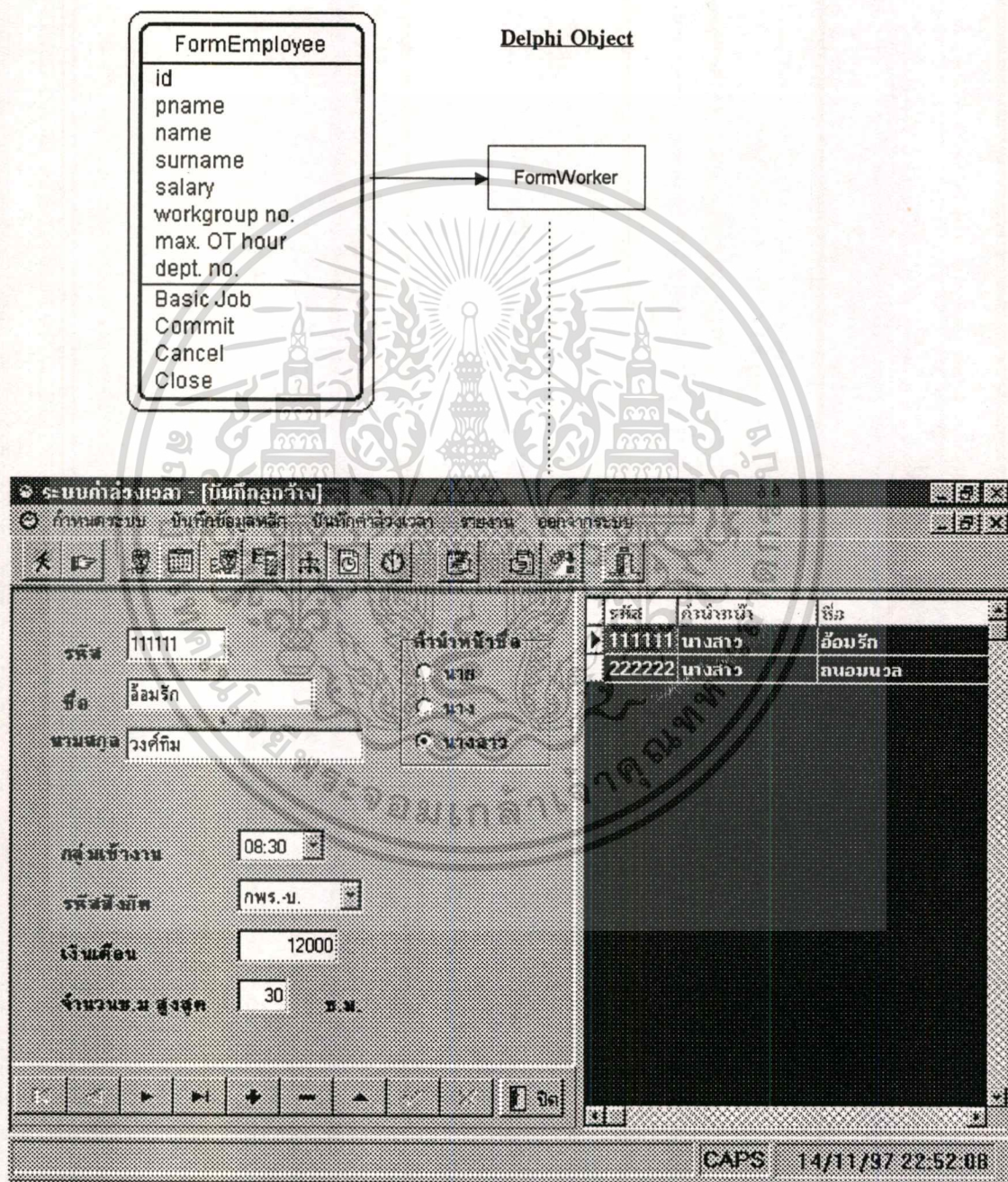


รูปที่ 3.43 แสดงการสร้าง FormHol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเราได้สร้าง Class เพิ่มเติม เพื่อใช้ในการ Inherit แล้ว ขั้นตอนต่อมาคือนำ HI Object แต่ละตัว มาพัฒนาโปรแกรมด้วย Delphi ซึ่งมีรายละเอียดดังต่อไปนี้

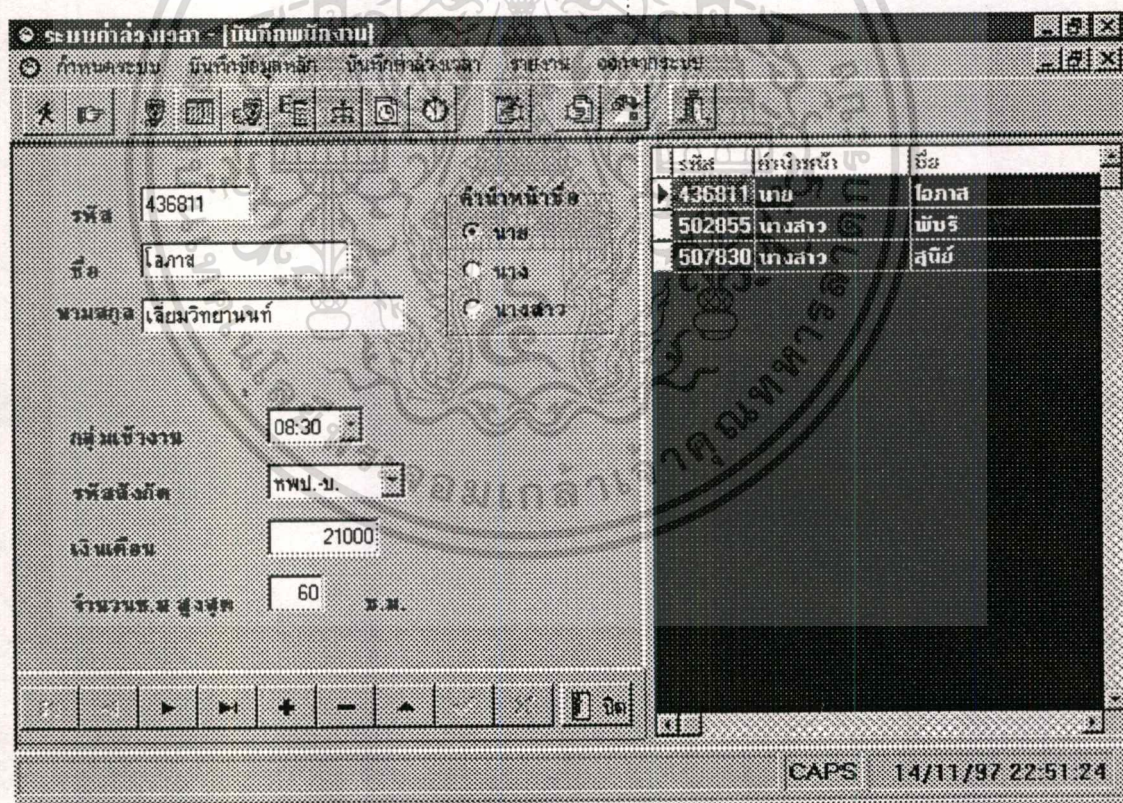
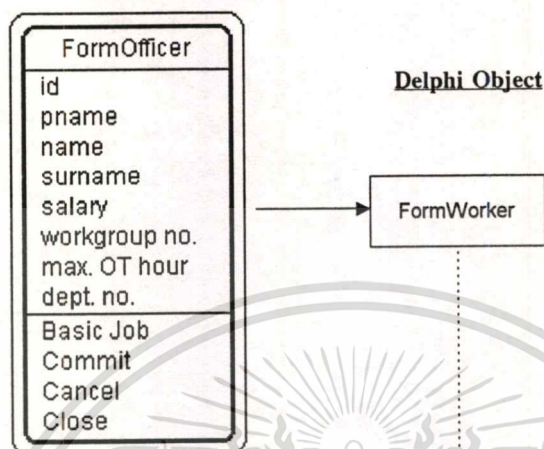
1. FormEmployee (Window บันทึกลูกจ้าง)จะถูกสร้างขึ้น โดยใช้ FormWorker classและเพิ่มเติมการให้ค่า ประเภทของผู้ปฏิบัติงาน ให้ค่าเท่ากับ 2



รูปที่ 3.44 แสดงการสร้าง FormEmployee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่จำกัดสิทธิ์ในสิ่งอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

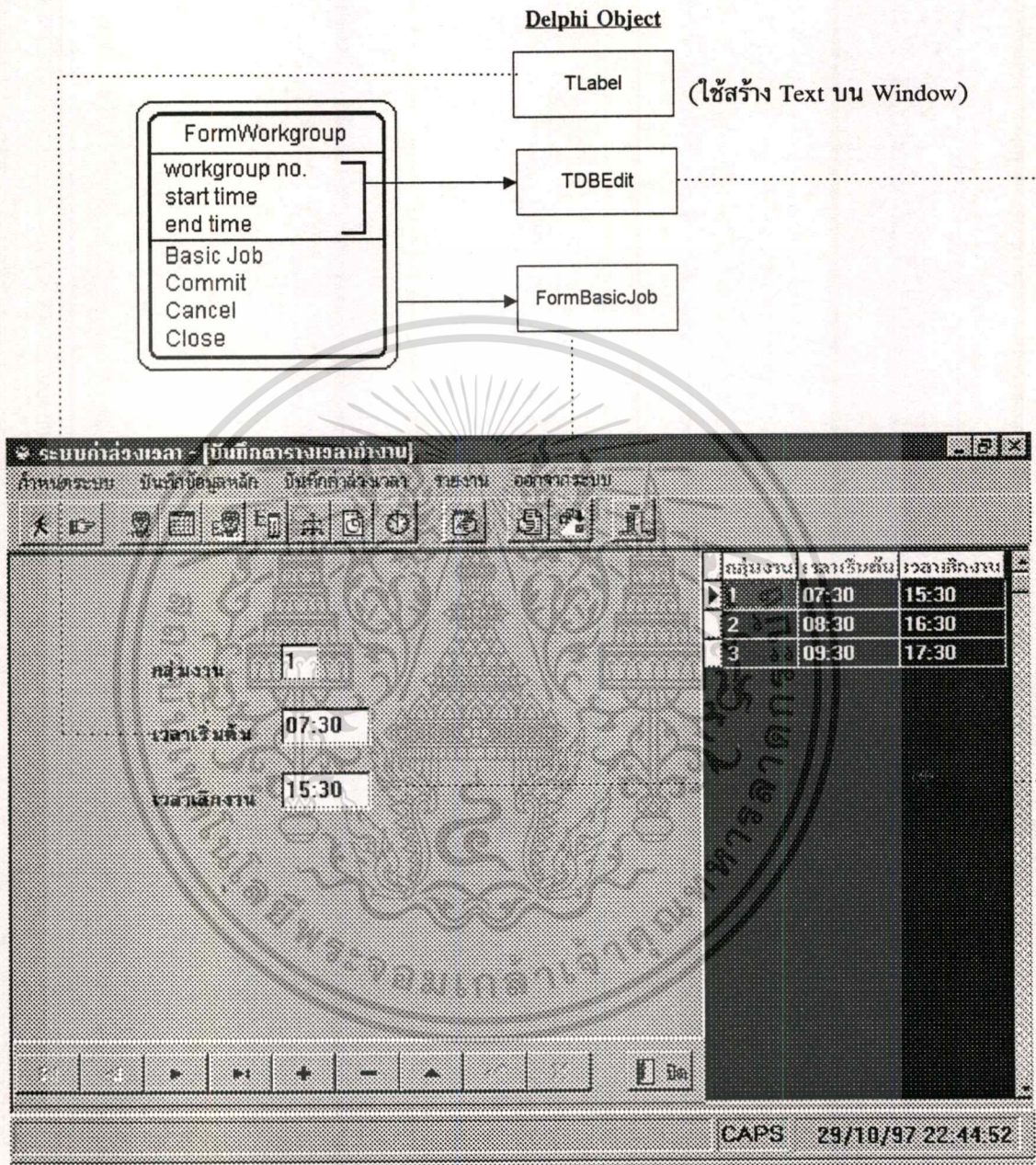
2. FormOfficer (Window บันทึกพนักงาน) จะถูกสร้างขึ้น โดยใช้ FormWorker class และเพิ่มเติมการให้ค่า ประเภทของผู้ปฏิบัติงาน ให้ค่าเท่ากับ 1



รูปที่ 3.45 แสดงการสร้าง FormOfficer

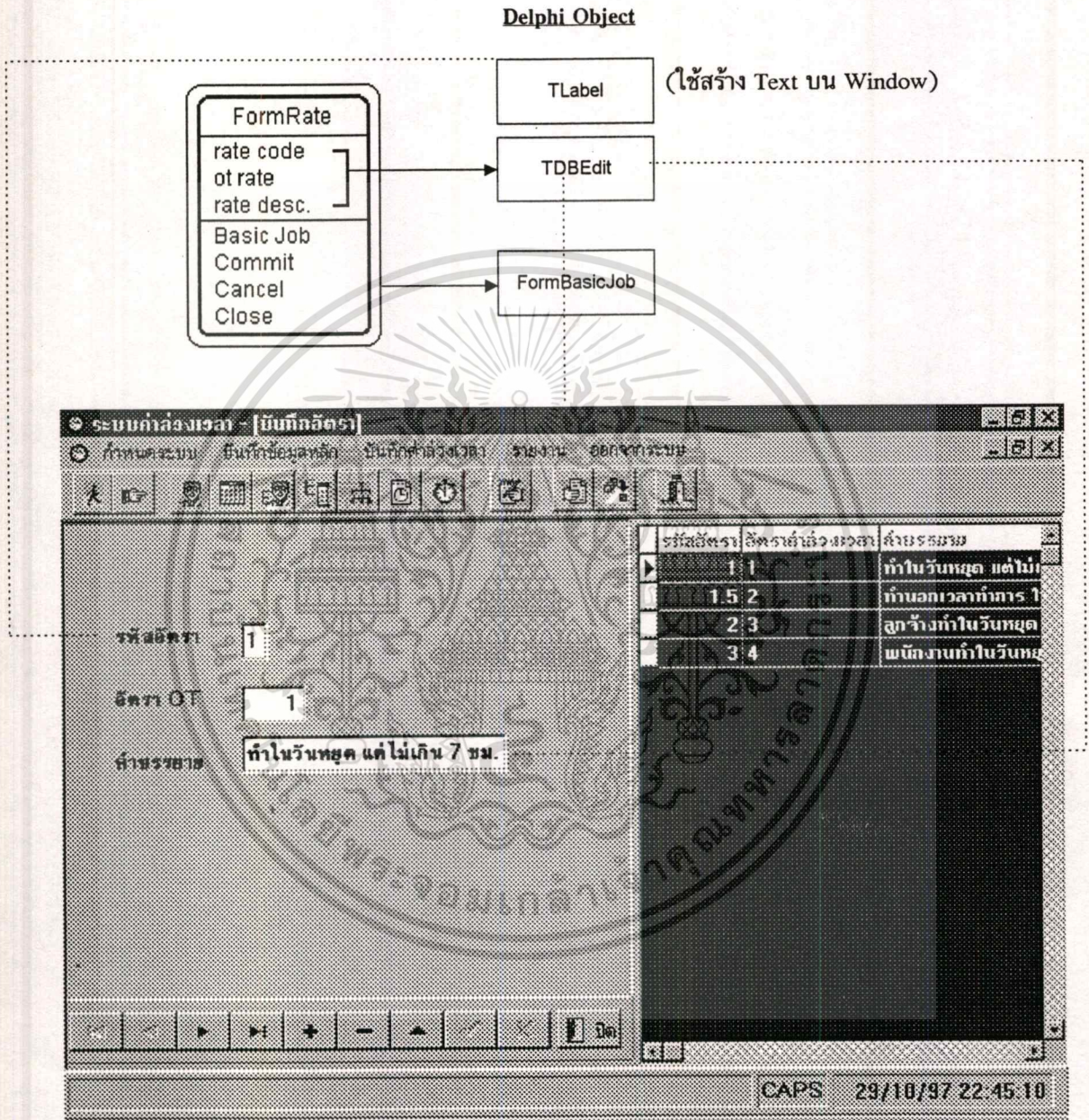
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. FormWorkGroup (Window บันทึกตารางเวลาทำงาน) จะมีวิธีการสร้างคล้ายกับ FormWorker ตามรูปข้างล่างนี้



รูปที่ 3.46 แสดงการสร้าง FormWorkgroup

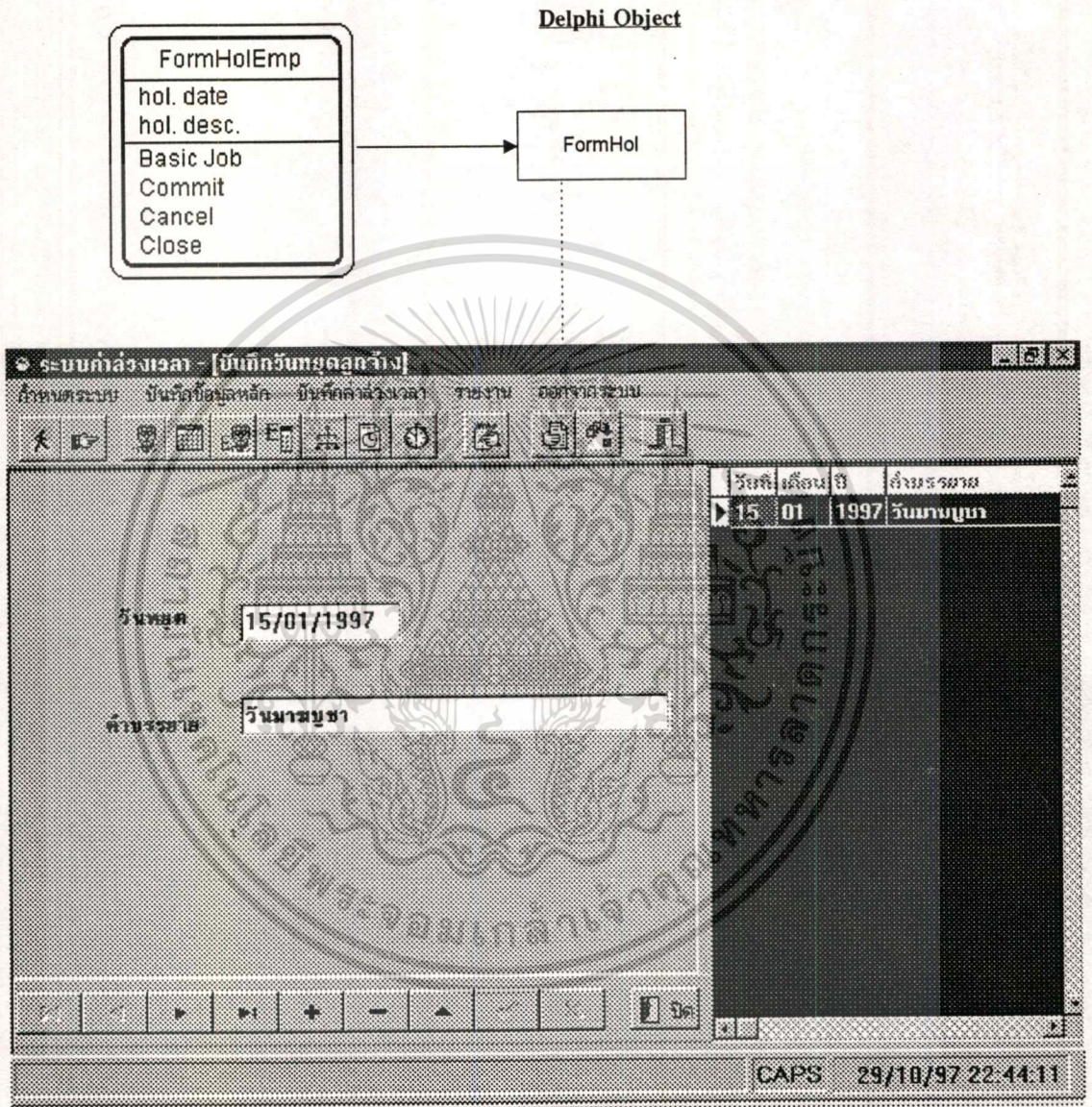
4. FormRate (Window บันทึกอัตรา) จะมีวิธีการสร้างคล้ายกับ FormWorker ตามรูปข้างล่างนี้



รูปที่ 3.47 แสดงการสร้าง FormRate

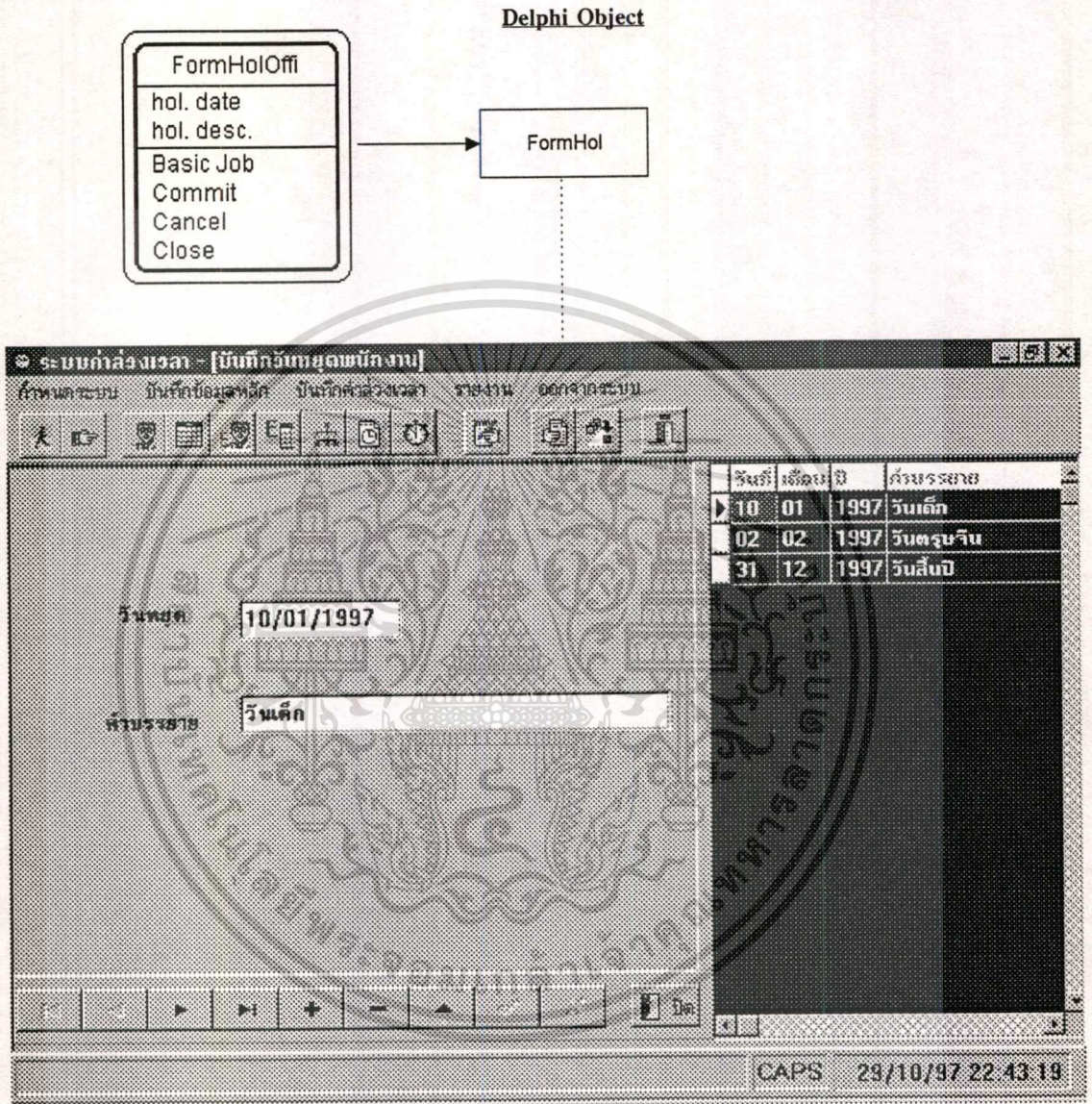
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. FormHolEmp (Window บันทึกวันหยุดลูกจ้าง) จะถูกสร้างขึ้น โดยใช้ FormHol class และเพิ่มเติมการให้ค่า ประเภทของผู้ปฏิบัติงาน ให้ค่าเท่ากับ 2



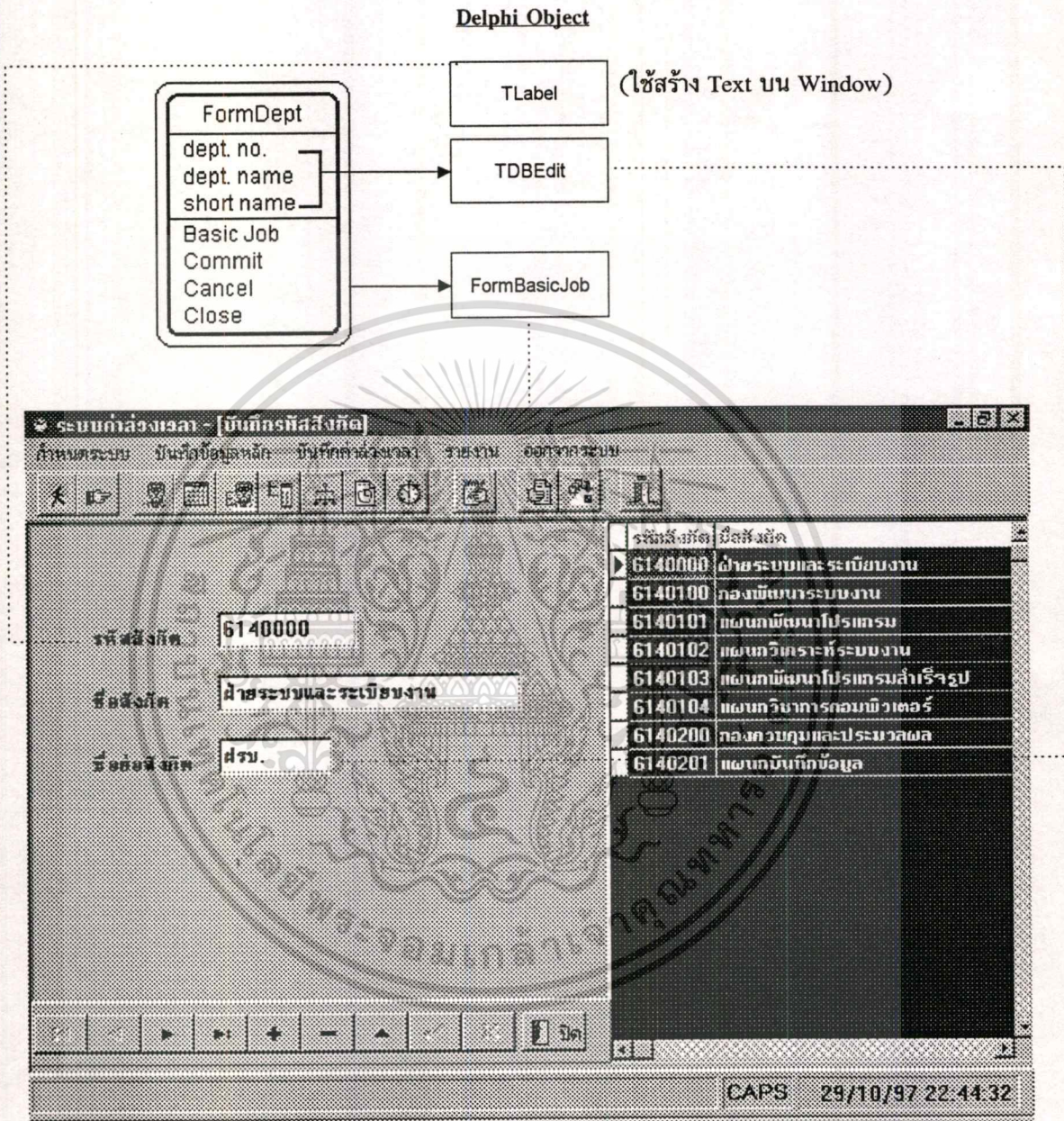
รูปที่ 3.48 แสดงการสร้าง FormHolEmp

6. FormHolOffi (Window บันทึกวันหยุดพนักงาน) จะถูกสร้างขึ้น โดยใช้ FormHol class และเพิ่มเติมการให้ค่า ประเภทของผู้ปฏิบัติงาน ให้ค่าเท่ากับ 1



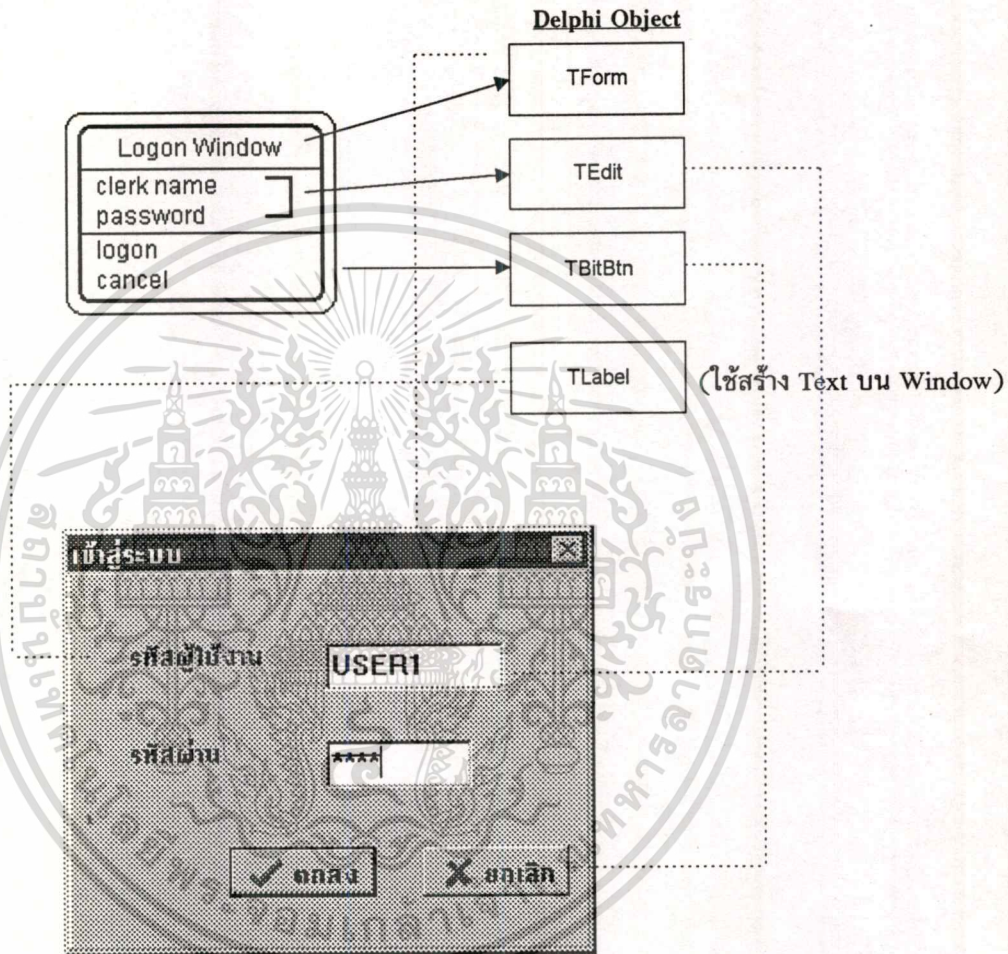
รูปที่ 3.49 แสดงการสร้าง FormHolOffi

7. FormDept (Window บันทึกสังกัด) จะมีวิธีการสร้างคล้ายกับ FormWorker ตามรูปข้างล่างนี้



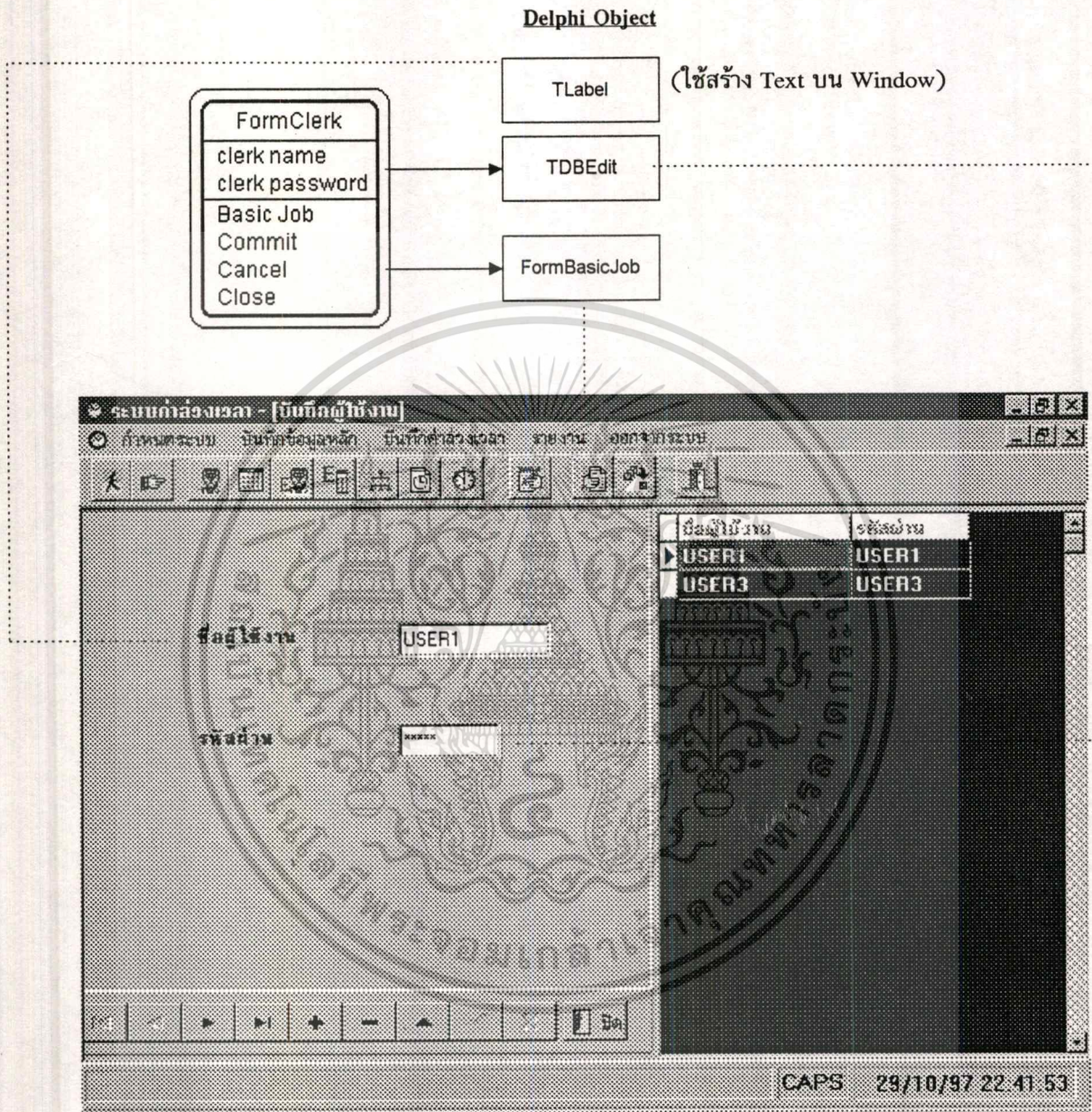
รูปที่ 3.50 แสดงการสร้าง FormDept

8. FormLogin (Window สำหรับ Login เข้าสู่ระบบ) การสร้าง FormLogin นี้ จะมี Non Visual Object ของ Delphi ชื่อ Tdatabase ด้วย เพื่อติดต่อกับ Database Server และสร้างส่วนที่เหลือ ตามรูปข้างล่างนี้ หลังจากนั้นเพิ่มเติม Method ตามที่ได้ออกแบบไว้ในขั้นตอน Human Interaction



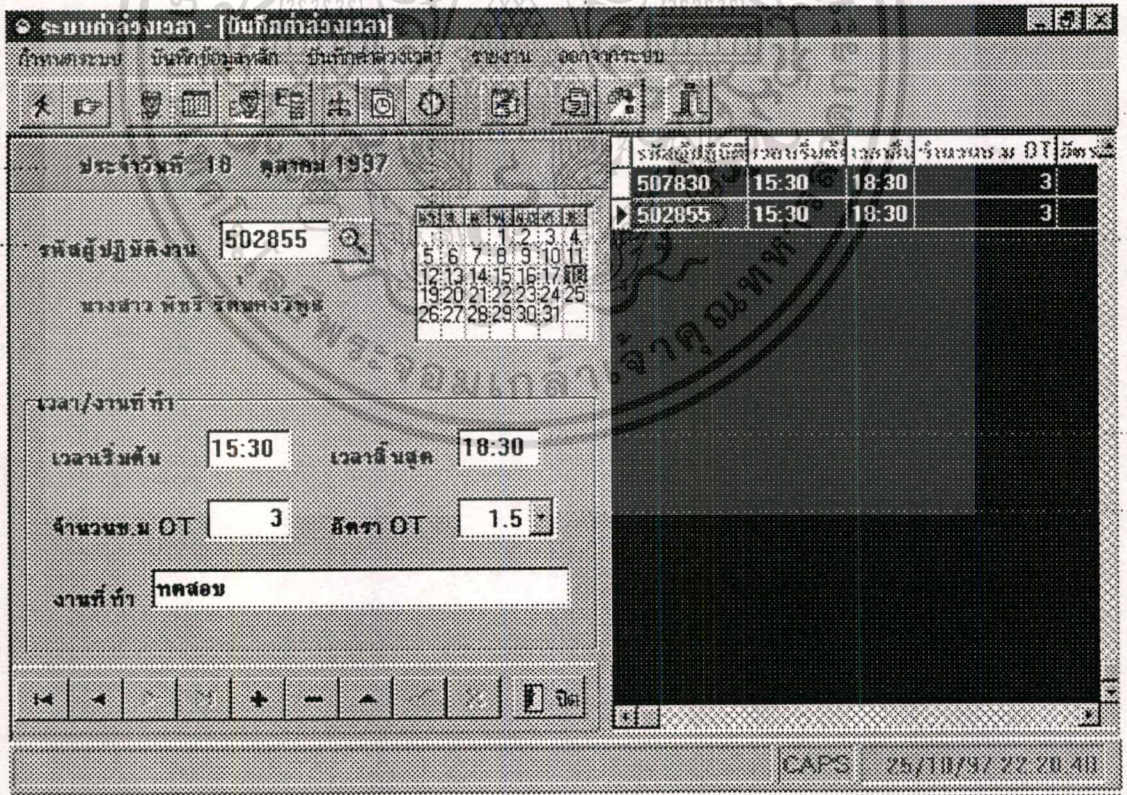
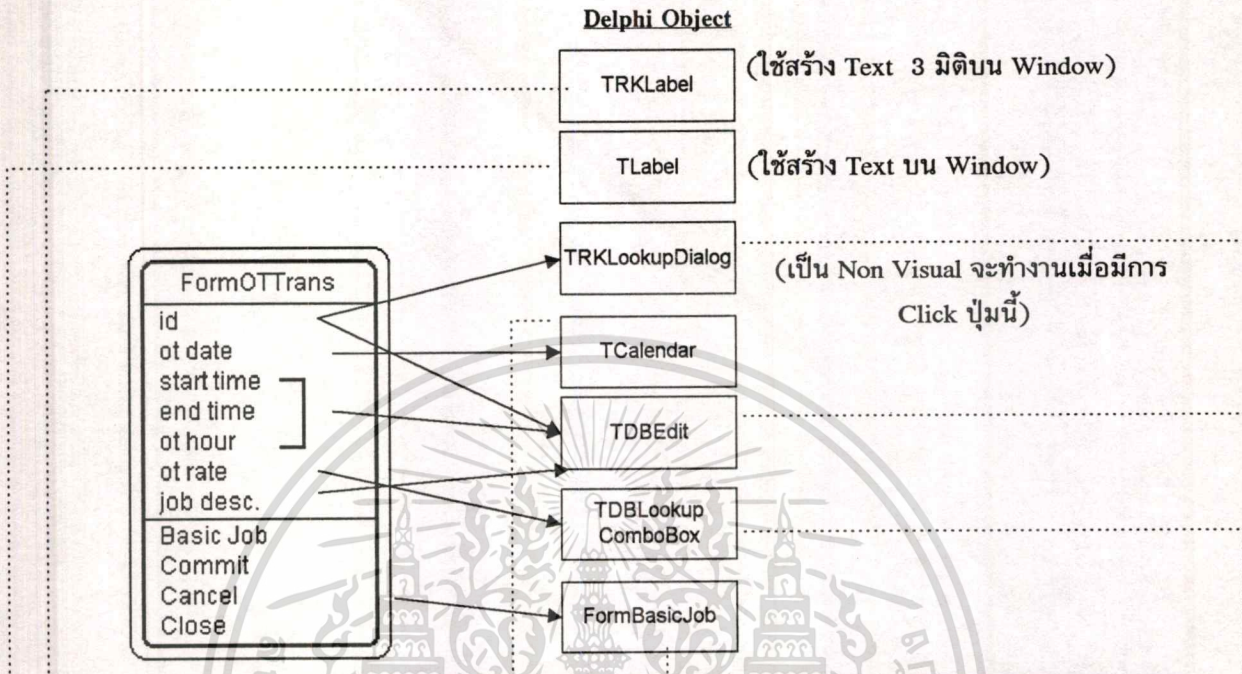
รูปที่ 3.51 แสดงการสร้าง FormLogin

9. FormClerk (Window บันทึกผู้ใช้งานในระบบ) จะมีวิธีการสร้างคล้ายกับ FormWorker ตามรูปข้างล่างนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

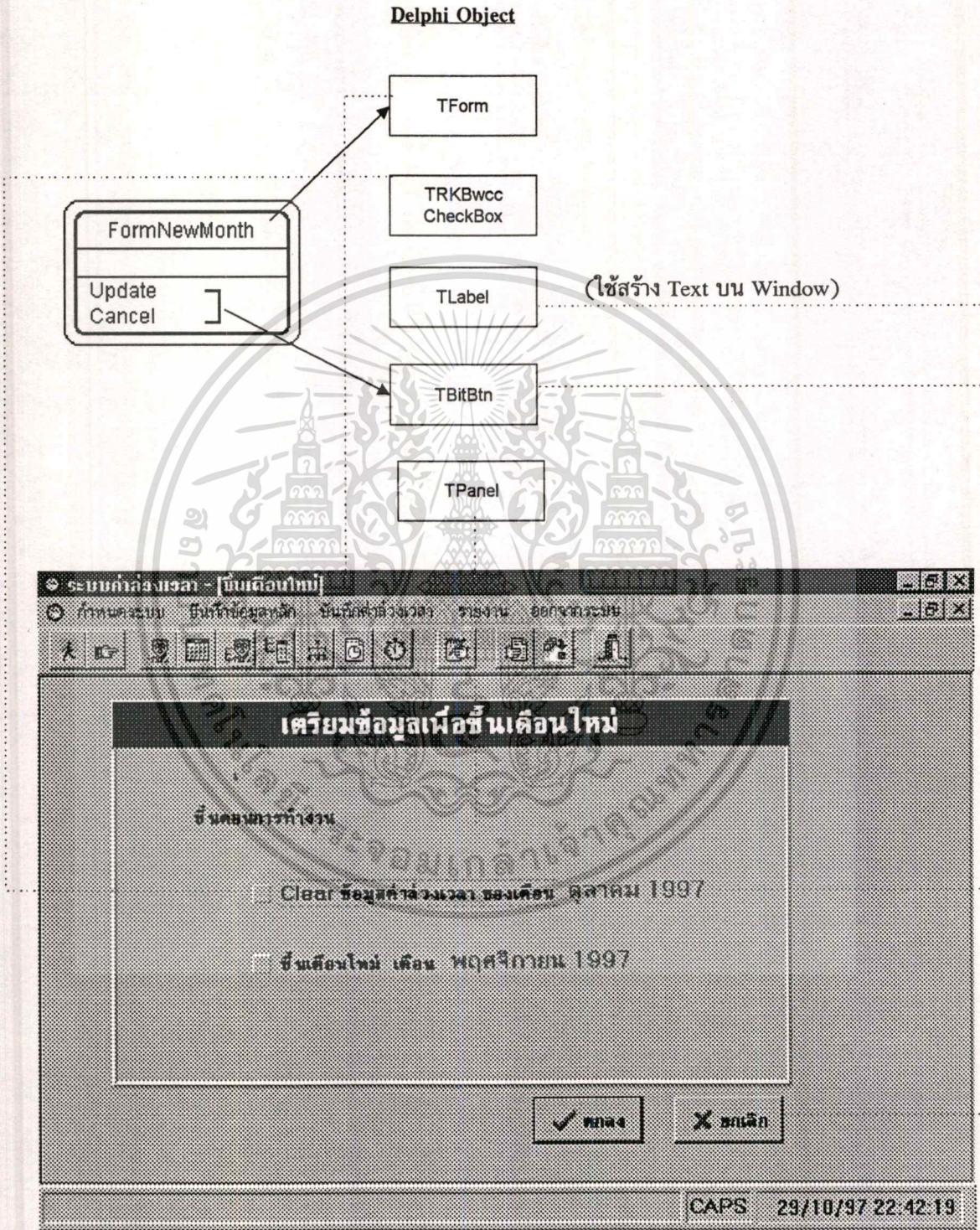
10. FormOtTrans (Window บันทึกค่าล่วงเวลา) จะมีวิธีการสร้างคล้ายกับ FormWorker ตามรูปข้างล่างนี้



รูปที่ 3.53 แสดงการสร้าง FormOtTrans

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

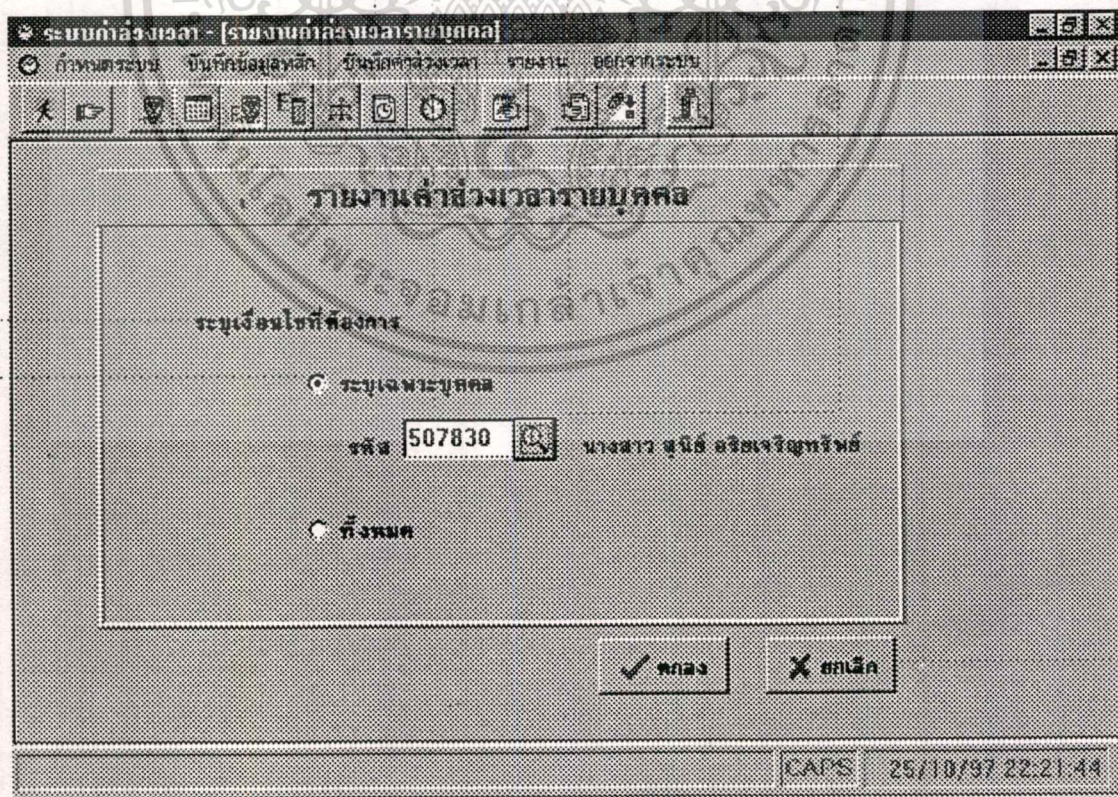
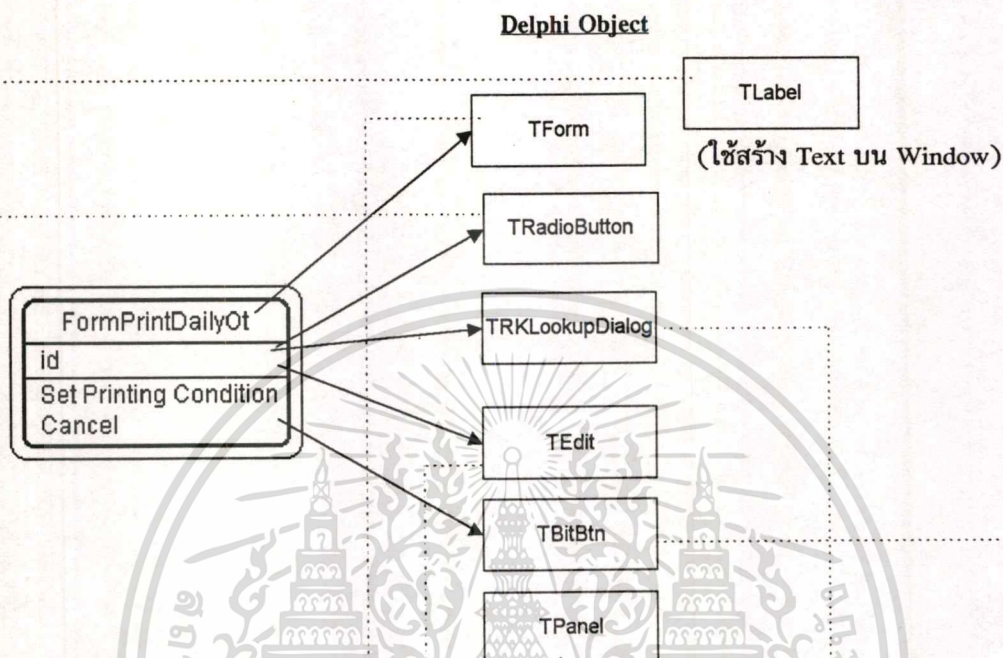
11. FormNewMonth (Window สำหรับขึ้นเดือนใหม่) จะมีวิธีการสร้างคล้ายกับ FormWorker ตามรูปข้างล่างนี้



รูปที่ 3.54 แสดงการสร้าง FormNewMonth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

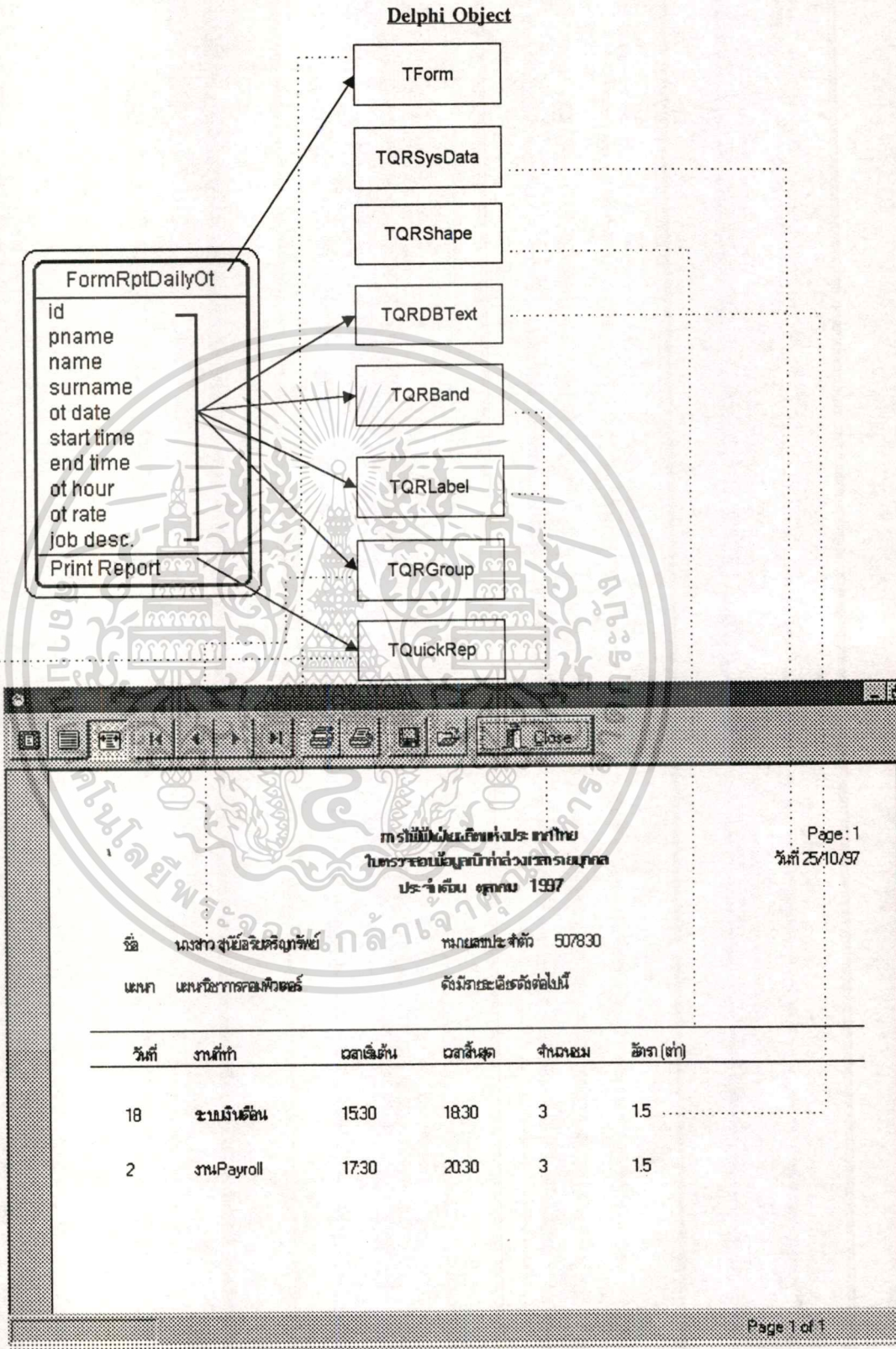
12. FormPrintDailyOt (Window สำหรับเลือกประเภทการออกรายงานรายบุคคล) สร้างตามรายละเอียดข้างล่างนี้



รูปที่ 3.55 แสดงการสร้าง FormPrintDailyOt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. FormRptDailyOt (Window ออกรายงานค่าล่วงเวลารายบุคคล)



รูปที่ 3.56 แสดงการสร้าง FormRptDailyOt

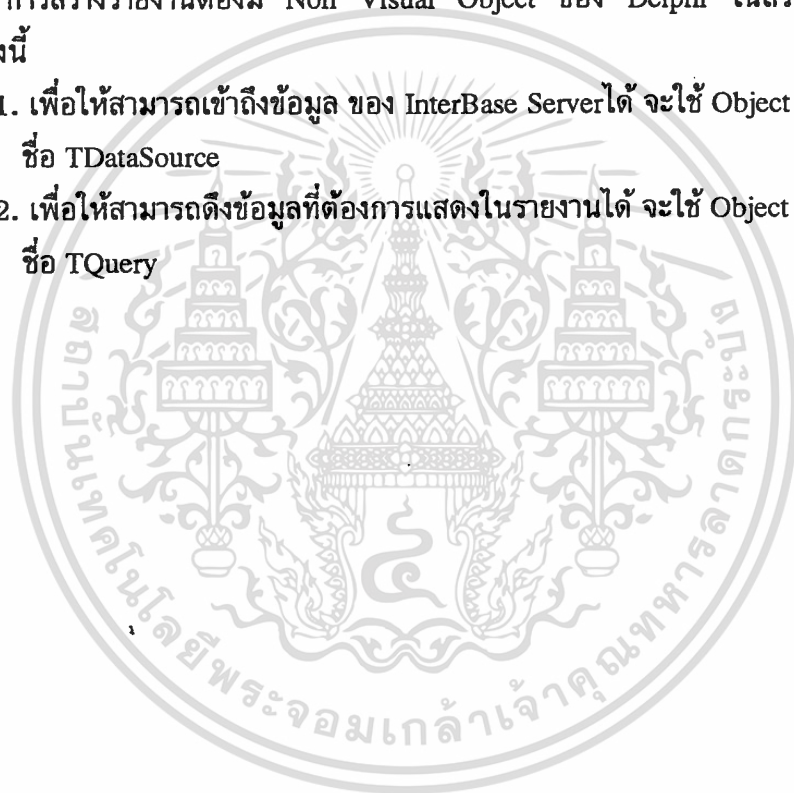
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างตามรูปร่างต้น ได้ใช้ Visual Object ของ Delphi ใน กลุ่ม QuickRpt Component Page ดังนี้

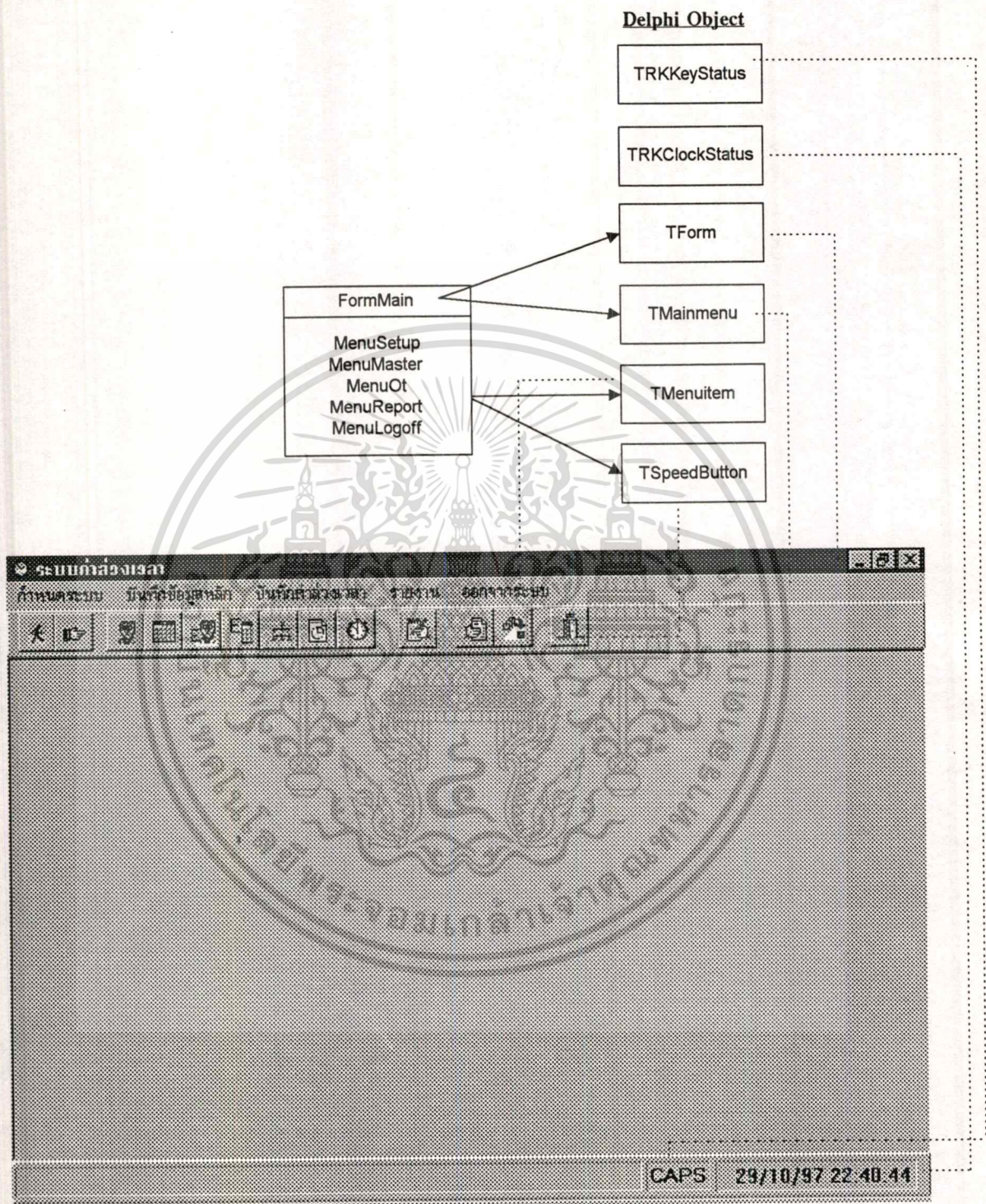
1. คำบรรยาย ใช้ TQRLabel Object
 2. การสร้าง Header ,Footer ใช้ TQRBand Object
 3. เส้น สีเหลี่ยม และรูปแบบอื่น ๆ ใช้ TQRShape Object
 4. วันเดือนปี และ เลขที่หน้า ใช้ TQRSysData
 5. ค่าใน Database ใช้ TQRDbText
 6. การจัดกลุ่มเพื่อพิมพ์รายงาน ใช้ TQRGroup เช่นพิมพ์รายงานรายบุคคล
- การสร้างรายงานต้องมี Non Visual Object ของ Delphi ในส่วนที่จะติดต่อกับ

Database ดังนี้

1. เพื่อให้สามารถเข้าถึงข้อมูล ของ InterBase Serverได้ จะใช้ Object ของ Delphi ชื่อ TDataSource
2. เพื่อให้สามารถดึงข้อมูลที่ต้องการแสดงในรายงานได้ จะใช้ Object ของ Delphi ชื่อ TQuery



15. Windows ที่ใช้เป็นเมนู



รูปที่ 3.58 แสดงการสร้าง Window ที่ใช้เป็นเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง Window ที่ใช้เป็นเมนู มีรายละเอียดดังนี้

1. Mainmenu จะสร้างโดยใช้ Object ชื่อ TMainMenu
2. Menu ย่อย อื่น ๆ ได้แก่ MenuSetup, MenuMaster, MenuOt, MenuReport, MenuLogoff จะสร้างจาก Object ชื่อ TMenuItem
3. การสร้าง Toolbar ทั้งหมดจะใช้ Object ชื่อ TSpeedbutton
4. การสร้าง StatusBar ใช้ TRkKeyStatus เพื่อแสดงสถานะของ Keyboard และใช้ TRkClockStatus เพื่อแสดงวันที่และเวลา ของเครื่องคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

บทสรุปและข้อเสนอแนะ

4.1 บทสรุป

การดำเนินการศึกษานี้ เป็นกรณีศึกษา การพัฒนาระบบคำสั่งเวลาด้วยหลักการแบบ Object-Oriented Paradigm โดยใช้ เทคนิคของ Object Modeling Technique (OMT) ของ Peter Coad เป็นพื้นฐานการวิเคราะห์และออกแบบระบบ และใช้ Delphi ในการพัฒนาโปรแกรม ต้นแบบ บนฐานข้อมูล InterBase และใช้เครื่องคอมพิวเตอร์ Pentium 133 MHz

การดำเนินการได้เริ่มจากการศึกษาความต้องการ ตลอดจน ปัญหาต่าง ๆ ของระบบ ในปัจจุบัน เพื่อเป็นแนวทางและเป็นประโยชน์ในการพัฒนาระบบคำสั่งเวลา จากนั้นเป็นขั้นตอนการวิเคราะห์และออกแบบระบบ ด้วยวิธีการของ OMT ซึ่งเป็นขั้นตอนการทำงาน แบบ Step-by-Step ผลการวิเคราะห์และออกแบบระบบ จะได้ Object, Class, attribute, method, ความสัมพันธ์ระหว่าง Object, Service ระหว่าง Object ตลอดจน โครงสร้างของฐานข้อมูล จากนั้นใช้หลักของ GUI มาทำการพัฒนาโปรแกรม สำหรับขั้นตอนการพัฒนาโปรแกรม ได้ทำการพัฒนาเพียงบางส่วนของระบบ เพื่อศึกษาปัญหาและความเป็นไปได้ในการพัฒนาเต็มรูปแบบต่อไป

ผลการศึกษาพบว่า การนำเอาวิธีการของ Object-Oriented มาใช้ในงานด้านวิเคราะห์ และออกแบบนั้น เป็นแนวคิด สามารถทำความเข้าใจได้ง่าย เพราะการวิเคราะห์และออกแบบ ทุกขั้นตอนจะพิจารณา จากสิ่งที่อยู่ในระบบเหมือนกับเป็นคำนามหรือ Object ตัวหนึ่ง (เช่น คน, สิ่งของ, สถานที่ ฯลฯ) ดังนั้นขั้นตอนต่อไป ในการวิเคราะห์ ก็จะนำสิ่งเหล่านั้น มาวิเคราะห์ ว่าสิ่งนั้น ประกอบด้วยอะไรบ้าง, รู้จัก Object อะไรบ้าง, มีการทำอะไรบ้าง และสุดท้าย แต่ละตัวมีความสัมพันธ์กันอย่างไร ซึ่งการวิเคราะห์และออกแบบเช่นนี้ เปรียบเสมือนการวิเคราะห์จากสิ่งที่เราพบเห็นอยู่ในชีวิตประจำวัน จึงเป็นการง่ายสำหรับผู้พัฒนาระบบใหม่ ๆ ที่จะพัฒนา นอกจากนี้ยังสามารถนำผลการวิเคราะห์และออกแบบข้างต้น ไปพัฒนาโปรแกรมแบบ OOP ได้ทันที อย่างมีประสิทธิภาพและสอดคล้องกัน เพราะใช้แนวคิดที่คล้ายกัน อันได้แก่ Encapsulation, Inheritance, Polymorphism, Reuse เป็นต้น

OOP เป็นเทคนิคการเขียนโปรแกรม ซึ่งมี Object เป็นส่วนประกอบที่สำคัญ โดยยึดหลักการในการรวบรวม Object และการดำเนินการกับ Object ที่เหมาะสม ส่วน Procedural Programming เป็นการสร้างโปรแกรมจากการนำคำสั่งย่อย ๆ มาประกอบกัน ซึ่งดูเหมือนง่าย แต่การเขียนโปรแกรมให้อ่านเข้าใจง่ายและหาข้อผิดพลาดได้ง่าย ขึ้นอยู่กับโปรแกรมเมอร์ ว่ามีความชำนาญหรือความถนัดแบบใด ซึ่งมีเพียงเจ้าของโปรแกรมเท่านั้น ที่จะทำความเข้าใจได้ง่าย ต่างกับ OOP ซึ่งโปรแกรมเมอร์ทุกคนจะสามารถทำความเข้าใจกับโปรแกรมของเดิมได้ และเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่สามารถแก้ไขได้ง่าย ดังนั้นการพัฒนาโปรแกรมแบบ OOP จะช่วยลดปัญหาเรื่องการบำรุงรักษา โปรแกรมได้ส่วนหนึ่ง

นอกจากประโยชน์ในขั้นตอนการออกแบบและวิเคราะห์ระบบงาน ตลอดจนถึงขั้นตอนการพัฒนาโปรแกรมแล้ว การใช้แนวคิด Object-Oriented ยังมีประโยชน์ในช่วงการ Maintenance และการนำ Object ที่พัฒนาแล้ว ไปใช้กับงานในระบบอื่น ๆ ต่อไปได้อีกด้วย เพราะแนวคิดดังกล่าว สามารถแก้ไขเพิ่มเติมโปรแกรมได้สะดวก รวดเร็ว และไม่มีผลกระทบต่อส่วนอื่นของระบบ ตลอดจนถึงผู้พัฒนาโปรแกรมคนอื่น สามารถจะเข้าใจการออกแบบและพัฒนาโปรแกรม ได้ดีกว่าการเขียนโปรแกรมในแบบเดิม (Structured)

แต่อย่างไรก็ตาม ในการใช้แนวคิดเรื่อง Object-Oriented เพื่อการพัฒนาระบบงานครั้งนี้ ก็ได้พบปัญหาบางประการ และมีแนวทางที่สามารถแก้ไขได้ ดังมีรายละเอียดดังนี้

1. ในการวิเคราะห์และออกแบบ โดยใช้ Object-Oriented เพื่อให้สามารถ Reuse ได้ในอนาคต จะใช้เวลาในการออกแบบนานและต้องคำนึงถึงความต้องการในอนาคต ซึ่งอาจจะถูกต้องหรือไม่ก็ได้ ดังนั้น การออกแบบและวิเคราะห์ เพื่อให้ได้ประโยชน์ดังกล่าว ควรพิจารณาและเปรียบเทียบระหว่างผลที่จะได้รับและสิ่งที่ต้องลงทุนด้วย
2. การวิเคราะห์และออกแบบ เช่นนี้ ถ้าใช้ Diagram แบบ Object-Oriented เป็นสื่อกับผู้ใช้งาน อาจจะทำความเข้าใจได้ยากกว่าแบบเดิม ซึ่งมีทั้ง Work Flow และ Data Flow Diagram ที่ผู้ใช้งานคุ้นเคยอยู่ นอกจากนี้ ในการวิเคราะห์และออกแบบ ควรต้องมีเครื่องมือที่ช่วยในการเก็บข้อมูลและรายละเอียดของระบบ เพื่อใช้ในการยืนยันความต้องการกับผู้ใช้งาน ว่าถูกต้องตามที่ได้ตกลงกันไว้ ดังนั้นถ้าจะใช้เทคนิคแบบ Object-Oriented ให้สมบูรณ์มากขึ้น ควรจะมีเครื่องมือ ที่สร้าง Document ต่าง ๆ เช่น เก็บข้อมูลความต้องการของผู้ใช้งาน, เก็บข้อมูลรายละเอียดของระบบ เป็นต้น
3. ในขณะวิเคราะห์และออกแบบด้วยเทคนิค Object-Oriented ควรจะยึด แนวคิด ของ Object-Oriented (Object, Class, Encapsulation, Inheritance, Polymorphism, Reuse เป็นต้น) เพราะบางครั้ง ผู้พัฒนาระบบ อาจจะติดกับการ ออกแบบในแนวคิดแบบเดิม ๆ ทำให้การวิเคราะห์และออกแบบ ไม่ได้เป็นไปตามแนวคิดแบบ Object-Oriented

4.1 ข้อเสนอแนะ

1. ในกรณีศึกษาครั้งนี้ ยังขาดการใช้ Object-Oriented Database เพราะในปัจจุบันยังไม่มี Object-Oriented Database Management System(OODBMS) ที่แท้จริง ดังนั้นถ้าในอนาคตมี OODBMS ก็ควรลองเลือกใช้ เพื่อให้การพัฒนาระบบแบบ Object-Oriented มีความสมบูรณ์มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การพัฒนาโปรแกรม เราสามารถใช้โปรแกรมอื่น ๆ ที่มีลักษณะแบบ OOP ได้ เช่น C++, Java เป็นต้น โดยเฉพาะ Java เพราะเป็นการพัฒนาโปรแกรมแบบ Open system หรืออาจจะใช้โปรแกรมที่ไม่เป็นแบบ OOP ก็ได้ เช่น Visual Fox, Visual Basic แต่อาจจะทำให้ขาดประโยชน์บางข้อของ OOP ไป เช่น Reuse
3. ในกรณีศึกษาครั้งนี้ ใช้ระบบงานที่ค่อนข้างเล็กและไม่ซับซ้อน จึงอาจจะเห็นประโยชน์ของการพัฒนาระบบด้วยแนวคิดแบบนี้ไม่มากนัก ดังนั้นจึงควรมีการนำไปศึกษากรณีที่ระบบมีขนาดใหญ่และซับซ้อนกว่านี้ เพื่อจะได้เห็นข้อดีและข้อเสียมากขึ้น ก่อนที่จะนำไปใช้พัฒนาระบบจริง



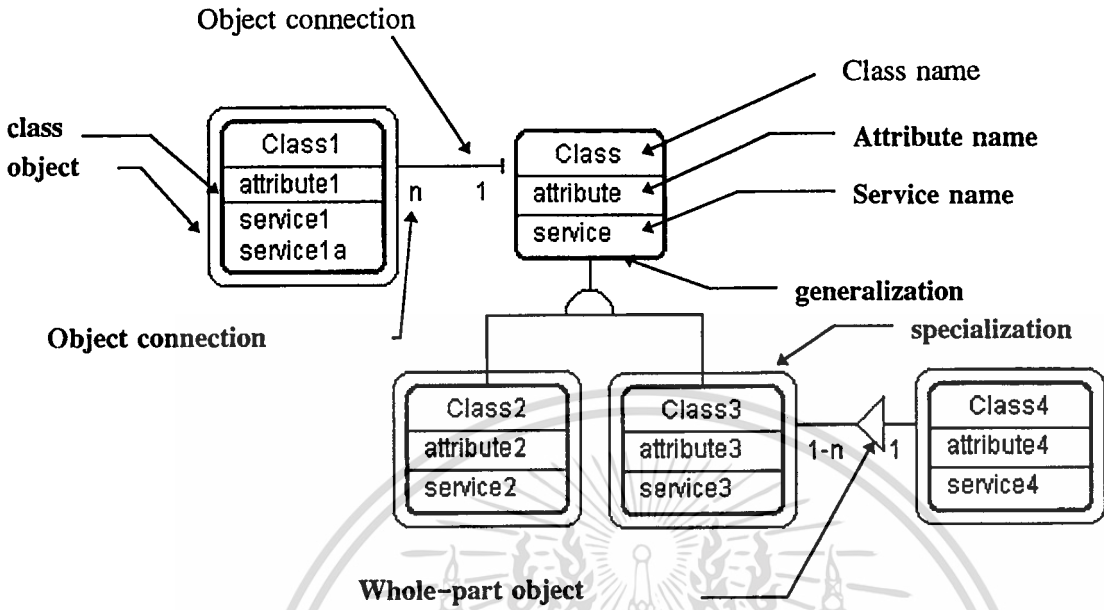
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- ขจรศักดิ์ คันธพนิต. “แนวทางในการพัฒนาโปรแกรม” บิซิเนส คอมพิวเตอร์ แมกะซีน. ปีที่ 5, ฉบับที่ 51(พฤษภาคม 2536): 160-165.
- บุญเลิศ เอี่ยมทัศนาศ. Delphi. กรุงเทพฯ:บริษัท ซีเอ็ดดูเคชั่น จำกัด(มหาชน),2539.
- ปิยดา จิตต์จำนงค์. การออกแบบระบบสารสนเทศบุคลากรสำหรับข้าราชการ โดย Object-Oriented paradigm, วิทยานิพนธ์ วิทยาศาสตร์มหาบัณฑิต. กรุงเทพฯ:บัณฑิตวิทยาลัย มหาวิทยาลัยมหิดล 2539.
- ประชา ตระการศิลป์, Client/Server Systems Development, ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- ราบินเดอร์ ศรีกิจจาภรณ์. “Object-Oriented Programming” บิซิเนส คอมพิวเตอร์ แมกะซีน. ปีที่ 5, ฉบับที่ 51(พฤษภาคม 2536): 160-165.
- รอม หิรัญพฤษ. “ฐานข้อมูลเชิงวัตถุ Object-Oriented Database” บิซิเนส คอมพิวเตอร์ แมกะซีน. ปีที่ 5, ฉบับที่ 51(พฤษภาคม 2536): 160-165.
- Aksit, Mehmet and Ludewijk Bergmans. Obstacles in Object-Oriented Software Development. Netherland:University of twente, 1992.
- Coad, Peter, David North and Mark Mayfield. Object Models, Second Edition. New Jersey:Prentice-Hall Inc.,1997.
- Graham, Ian. Object Oriented Methods, Second Edition. London:Addison-Wesley Publishing Company Inc., 1994.
- Jacobson, Ivan. Object Oriented Software Engineering. London:ACM Press,1992.
- Konopka, Ray. Developing Custom Delphi 3 Components. Arizona:An International Thomson Publishing company, 1997.
- Osier, Dan, Steve Grobman and Steve Batson. Teach Yourself Delphi 2 in 21 Days. Indiana : Sams Publishing,1996.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ประเภทของการ connect

- 0-1
- 0-1
- 1
- 1
- n
- 0 or more
- 1-n
- 1 or more
- ordered n
- ordered collection
- (XOR A) n
- XOR across connections labeled "A"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table : Worker(ผู้ปฏิบัติงาน)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
id	หมายเลขผู้ปฏิบัติงาน	char	6	
pname	คำนำหน้า	char	12	
name	ชื่อ	char	15	
surname	นามสกุล	char	20	
salary	เงินเดือน	integer	6	
wg_no	กลุ่มเข้างาน	char	1	
max_ot	จำนวนชม.ม. ล่วงเวลาสูงสุด	integer	3	
dept_no	รหัสสังกัด	char	7	
type	ประเภทผู้ปฏิบัติงาน	char	1	มีค่า 1 = พนักงาน 2 = ลูกจ้าง

Table : Clerk(พนักงานธุรการ:ผู้ใช้ระบบ)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
clerk_name	ชื่อผู้ใช้ระบบ	char	15	
clerk_password	รหัสผ่าน	char	10	

Table : Workgroup(กลุ่มเข้างาน)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
wg_no	รหัสกลุ่มเข้างาน	char	1	
wg_stime	เวลาเริ่มงาน	char	5	
wg_ftime	เวลาเลิกงาน	char	5	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table : Rate(อัตราค่าล่วงเวลา)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
rate_code	รหัสอัตรา	char	1	มีค่า 1= ทำในวันหยุด แต่ไม่เกิน 7 ช.ม. 2= ทำในวันทำการ 3= ลูกจ้างทำในวันหยุดและเกิน 7 ช.ม 4= พนักงาน ทำในวันหยุด และ เกิน 7 ช.ม
ot_rate	อัตราค่าล่วงเวลา	numeric	2,1	
rate_desc	คำบรรยาย	char	30	

Table : Holiday(วันหยุด)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
hol_year	ปีของวันหยุด	char	4	
hol_month	เดือนของวันหยุด	char	2	
hol_day	วันที่ของวันหยุด	char	2	
hol_desc	คำบรรยายของวันหยุด	char	20	

Table : Department(สังกัด)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
dept_no	รหัสสังกัด	char	7	
dept_name	ชื่อสังกัด	char	35	
short_name	ชื่อย่อสังกัด	char	15	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table : Ot_trans(รายการค่าล่วงเวลา)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
id	รหัสผู้ปฏิบัติงาน	char	1	
ot_day	วันที่ทำงานล่วงเวลา	char	2	
ot_month	เดือนที่ทำงานล่วงเวลา	char	2	
ot_year	ปีที่ทำงานล่วงเวลา	char	4	
start_time	เวลาเริ่มต้นทำล่วงเวลา	char	5	
end_time	เวลาสิ้นสุดทำล่วงเวลา	char	5	
ot_hour	จำนวนช.ม ล่วงเวลา	numeric	4,2	
ot_rate	อัตราค่าล่วงเวลา	numeric	2,1	
job_desc	งานที่ทำล่วงเวลา	char	30	

Table : Monthly_Ot(สรุปค่าล่วงเวลารายเดือน)

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
id	รหัสผู้ปฏิบัติงาน	char	1	
ot_year	ปีที่ทำงานล่วงเวลา	char	4	
ot_month	เดือนที่ทำงานล่วงเวลา	char	2	
tot_hour	จำนวนช.ม รวม	numeric	5,2	
tot_hour_pay	จำนวนช.ม รวมที่จะจ่าย ค่าล่วงเวลา	numeric	5,2	
ot_amount	จำนวนเงินค่าล่วงเวลา	numeric	8,2	

Table : MonthStatus(เดือนปีของข้อมูล)

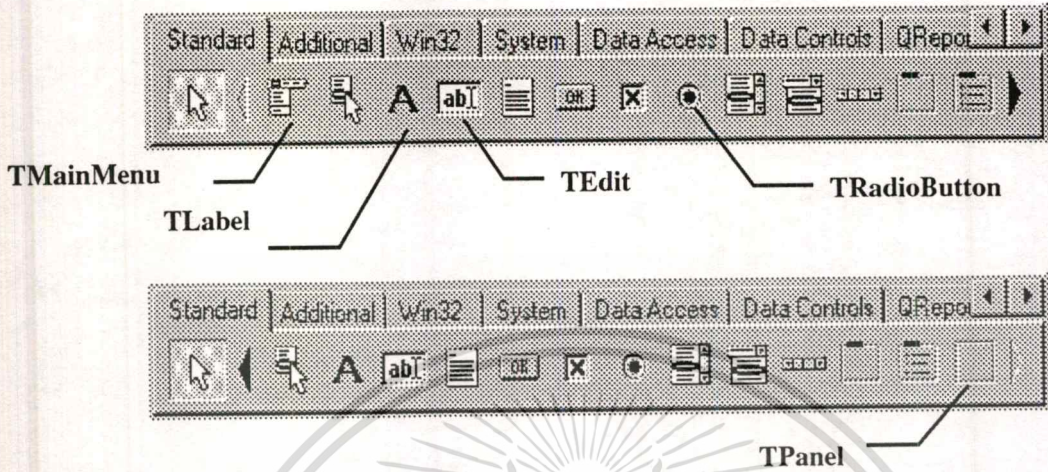
ชื่อ Field	คำอธิบาย	ประเภท	ความยาว	หมายเหตุ
ot_month	เดือนของข้อมูล	char	2	
ot_year	ปีของข้อมูล	char	4	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

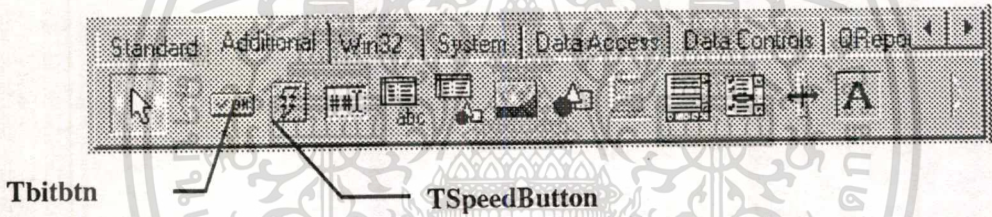


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

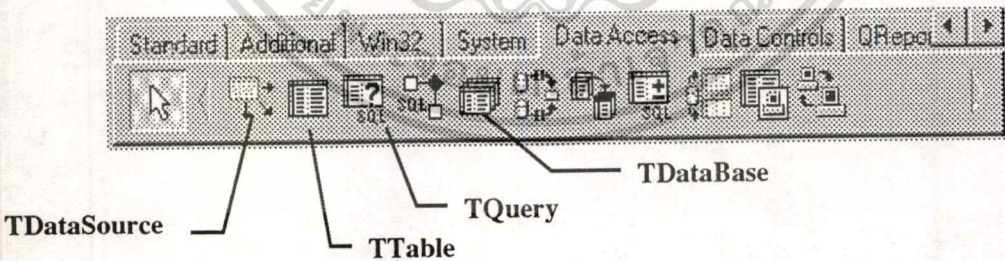
1) Standard Component Page



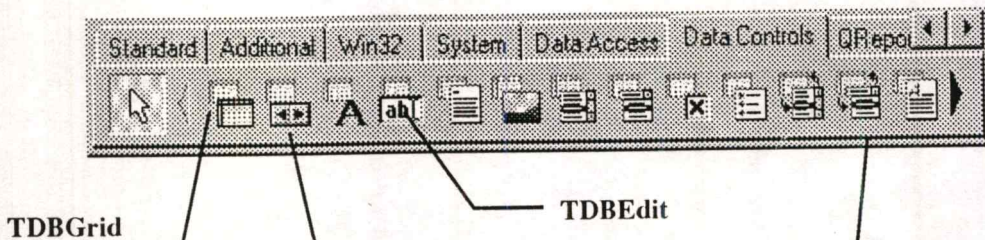
2. Additional Component Page



3. Data Access Component Page

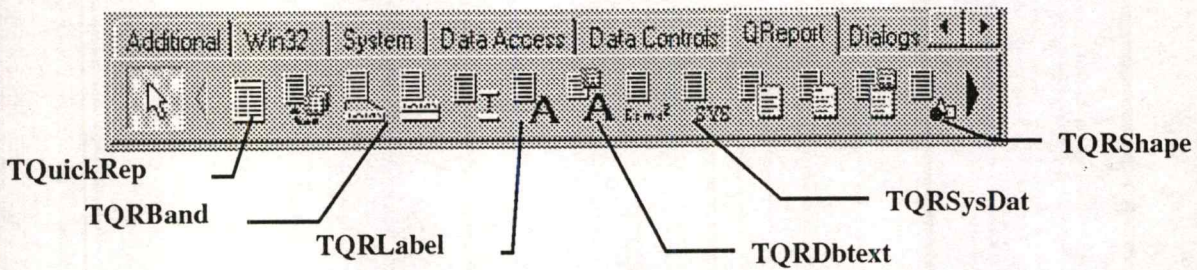


4. Data Controls Component Page

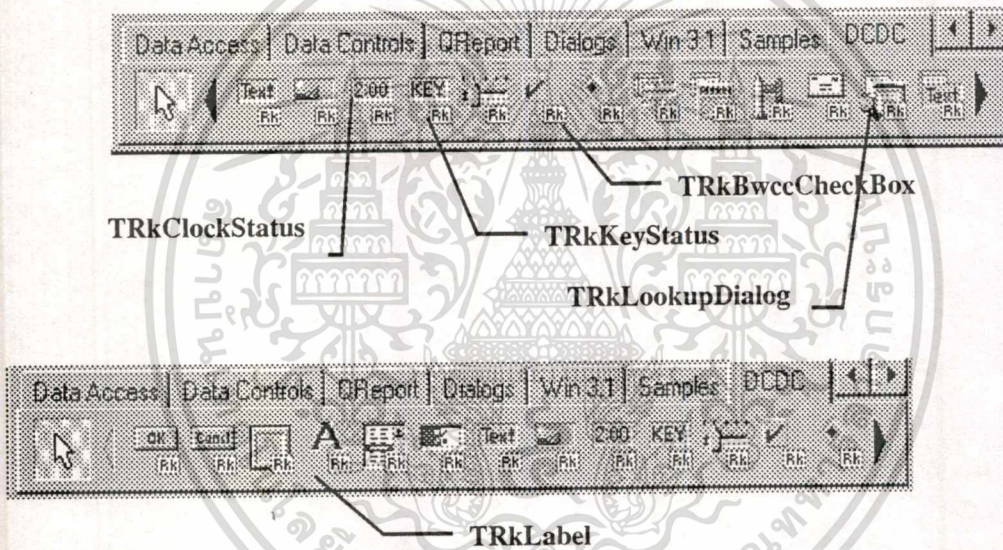


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในของมหาวิทยาลัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สามารถใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องสงวนลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

5. QuickRpt Component Page

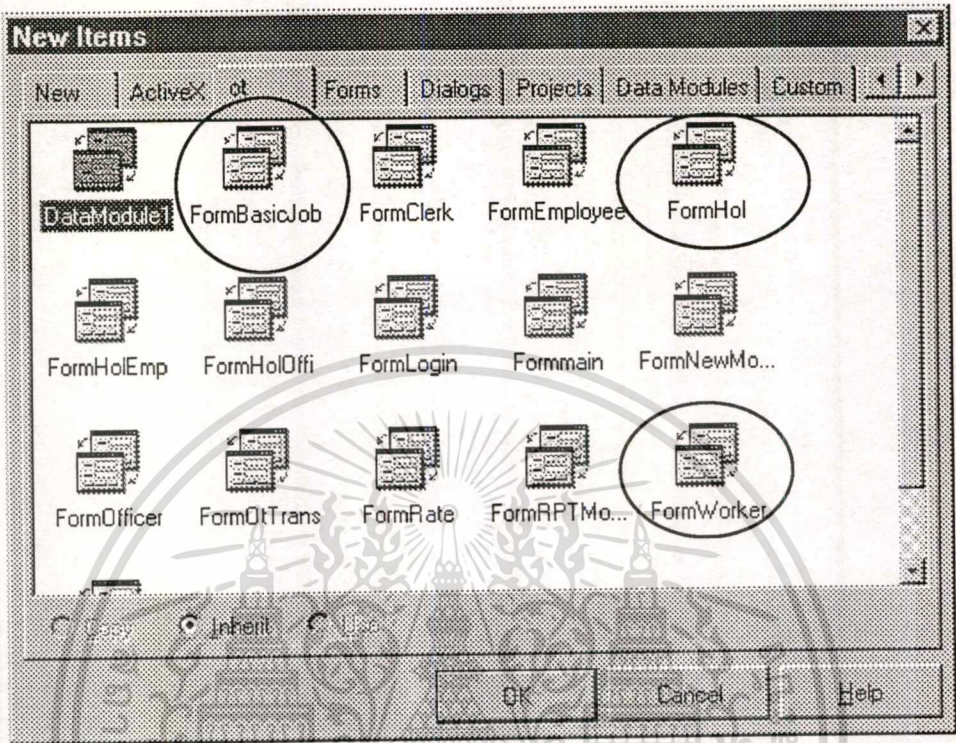


6. DCDC Component Page ซึ่งเป็นส่วนเพิ่มเติม ของ Third Party โดย Ray Konopka

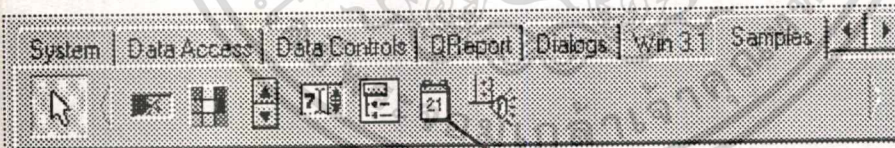


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. Class ที่สร้างขึ้นใช้ภายใน Project การเรียกใช้งาน ของส่วนนี้ เรียกใช้โดยที่เมนู File/New



8. Samples Component Page



TCalendar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



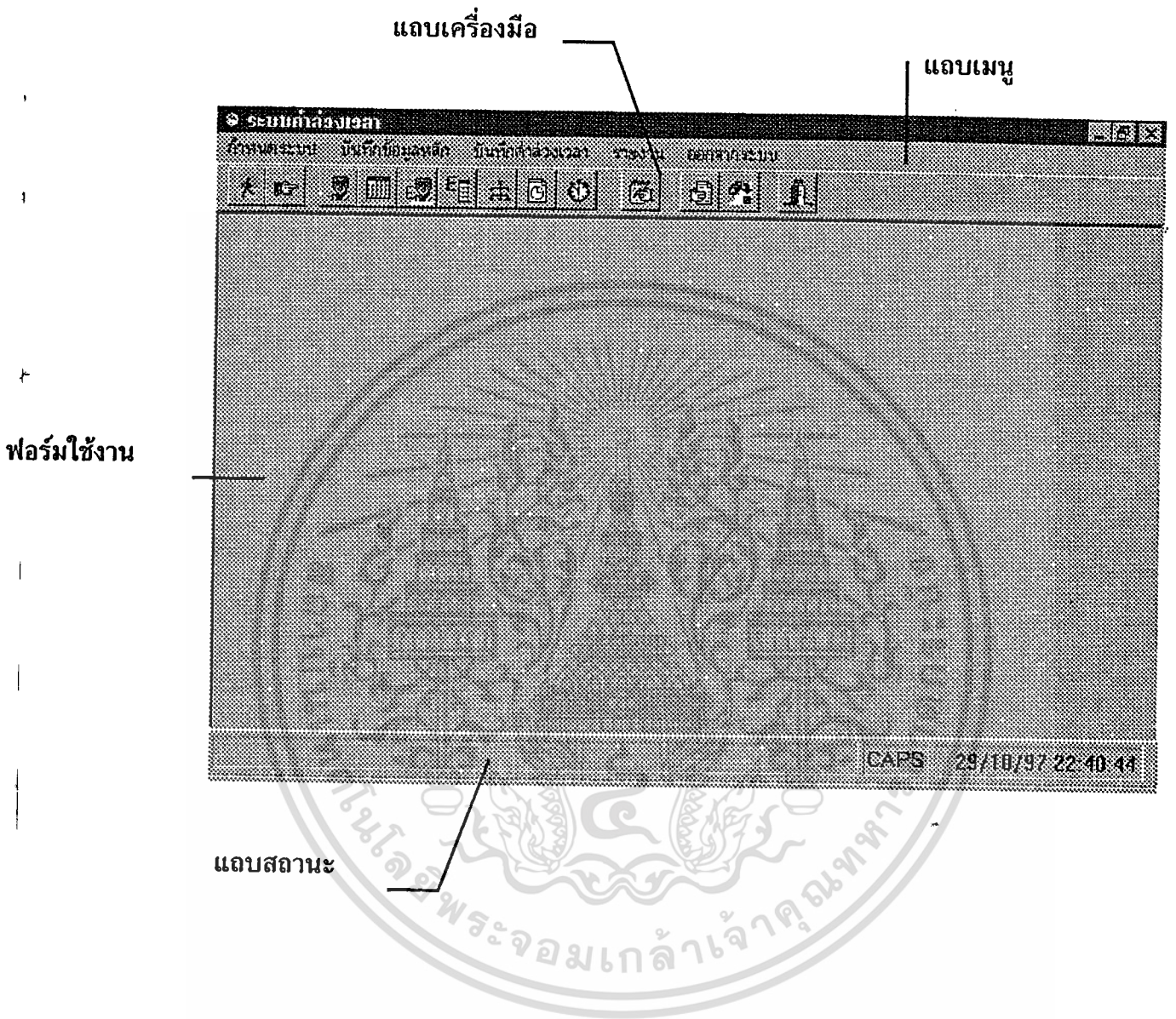
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดโปรแกรม ค่าล่วงเวลา ประกอบด้วย การทำงานต่าง ๆ ดังนี้

- การเข้าทำงาน
- การบันทึกผู้มีสิทธิใช้งานและการขึ้นเดือนใหม่
- บันทึกค่าล่วงเวลา
- บันทึกข้อมูลหลัก
 - เพิ่มพนักงาน
 - เพิ่มลูกจ้าง
 - เพิ่มตารางเวลาทำงาน
 - เพิ่มสังกัด
 - เพิ่มวันหยุดพนักงาน
 - เพิ่มวันหยุดลูกจ้าง
 - เพิ่มอัตราค่าล่วงเวลา
- ออกรายงาน
 - รายงานการเบิกค่าล่วงเวลา
 - รายงานสรุปค่าล่วงเวลา
- การจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

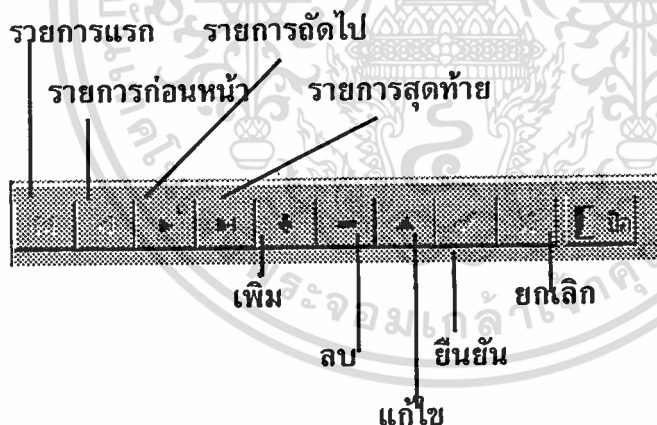
รายละเอียดของจอภาพในการทำงานของระบบงาน จะแสดงหลังจากผ่านการ Login เข้าสู่ระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วย 4 ส่วนดังนี้

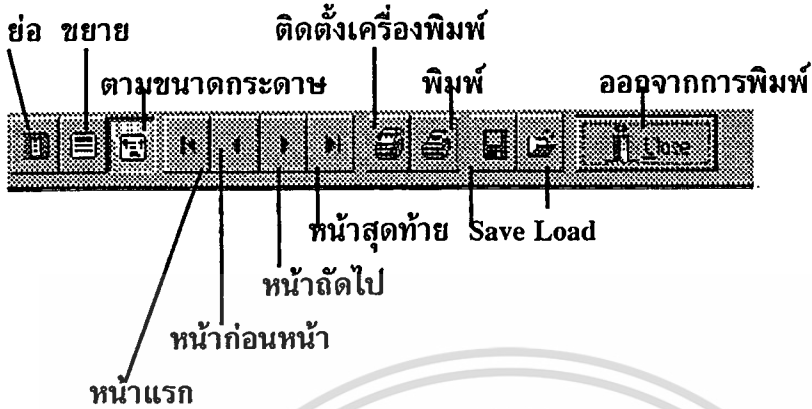
- ส่วนรายการข้อมูล เป็นส่วนที่แสดงรายการปัจจุบัน เพื่อทำการบันทึกหรือแก้ไขข้อมูล
- ส่วนควบคุมรายการข้อมูล มี 9 คำสั่งให้เลือกคือ รายการแรก,รายการก่อนหน้า, รายการถัดไป, รายการสุดท้าย, เพิ่ม,ลบ, แก้ไข,ตกลงและยกเลิก
รายการแรก เพื่อไปยังข้อมูลรายการแรก
รายการก่อนหน้า เพื่อไปยังข้อมูลก่อนหน้า
รายการถัดไป เพื่อไปยังข้อมูลถัดไป
รายการสุดท้าย เพื่อไปยังข้อมูลสุดท้าย
เพิ่ม เพื่อระบุว่าต้องการจะเพิ่มรายการ
ลบ เพื่อระบุว่าต้องการลบรายการนี้
แก้ไข เพื่อระบุว่าต้องการแก้ไขรายการปัจจุบัน
ตกลง เพื่อยืนยันการบันทึกหรือแก้ไขข้อมูล
ยกเลิก เพื่อยกเลิกการบันทึกหรือแก้ไขข้อมูล



- ส่วนแสดงรายการข้อมูล เป็นส่วนที่จะแสดงรายการข้อมูลทั้งหมด ถ้ากรณีข้อมูลมีมากกว่าขนาดของส่วนรายการข้อมูล จะสามารถเลื่อนดูข้อมูลทั้งหมดได้ โดยใช้แถบเลื่อน(Scroll Bar)
- ออกจากการทำงาน เพื่อออกจากการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

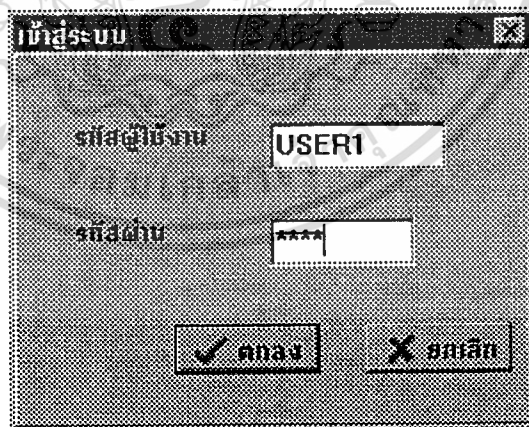
คำอธิบายแถบเครื่องมือการพิมพ์



การใช้คำสั่งของระบบคำสั่งวงเวลา มี 2 แบบคือ

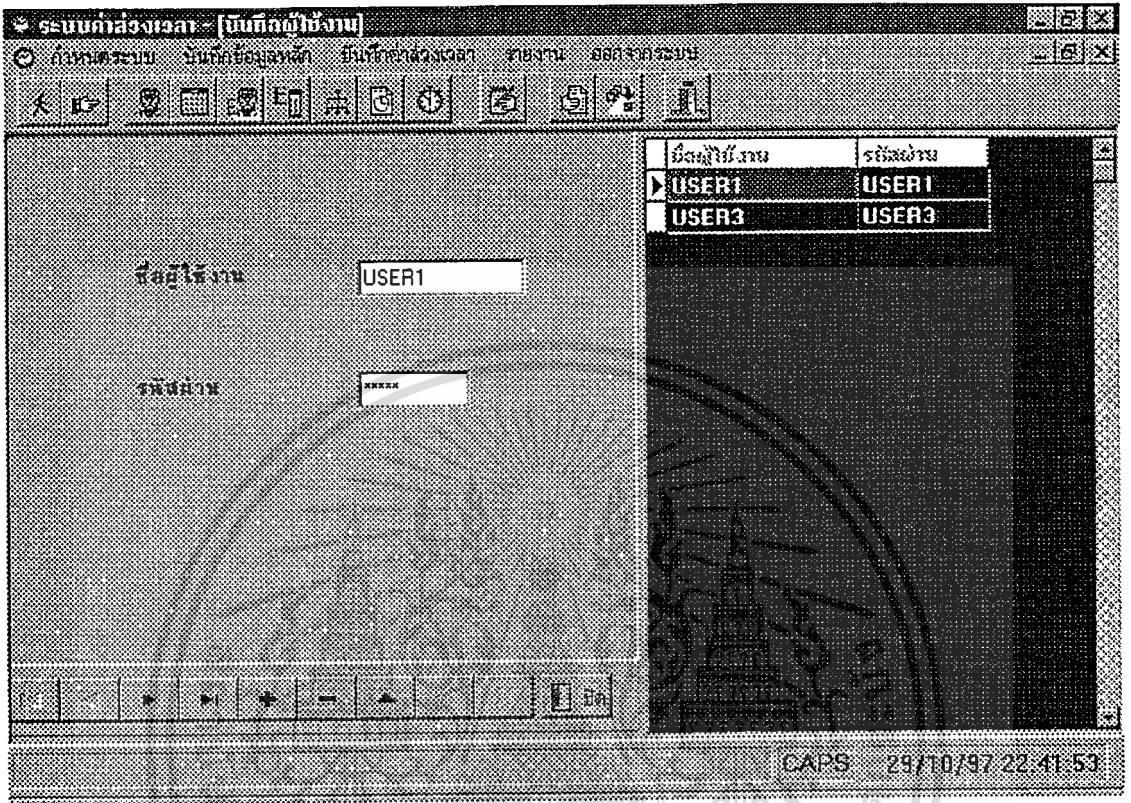
- แถบเมนู(Menu Bar) Click เม้าส์ ณ ตำแหน่งเมนูคำสั่งที่ต้องการ เพื่อเปิดเมนู
- แถบเครื่องมือ(Tool Bar) Click เม้าส์ ณ บริเวณ Icon เครื่องมือที่ต้องการ

การเข้าทำงาน โดยใส่รหัสผู้ใช้งานและรหัสผ่านให้ถูกต้อง จากนั้นกดปุ่ม ตกลงเพื่อยืนยันการทำงาน ถ้าไม่ต้องการทำงานต่อ กดปุ่ม ยกเลิก



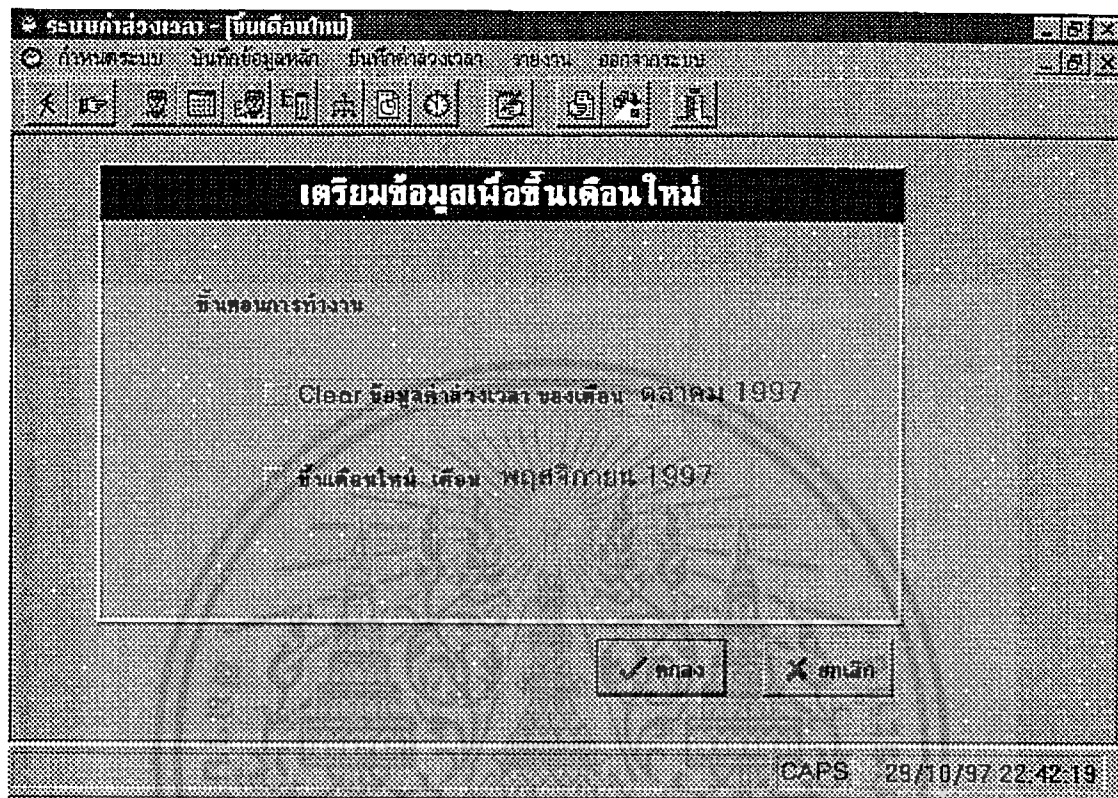
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกผู้ใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นเดือนใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกพนักงาน

ระบบกำลังเวลา - [บันทึกพนักงาน]

กำหนดระบบ บันทึกข้อมูลผลัด บันทึกค่าล่วงเวลา รายงาน ออกจากระบบ

รหัส 436811

ชื่อ โสภส

สถานศึกษา เสียมวิทยายนนท์

กลุ่มเช้างาน 08:30

รหัสสังกัด ทพป.บ.

เงินเดือน 21000

จำนวนวัน สูงสุด 60 วัน

ตำแหน่งงาน

นาย
 นาง
 นางสาว

รหัส	ค่านักเข้า	ชื่อ
436811	นาย	โสภส
502855	นางสาว	บันรี
507830	นางสาว	สุณีย์

CAPS 14/11/97 22:51:24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกวันหยุดพนักงาน

ระบบกำลังแสดง: บันทึกวันหยุดพนักงาน

กำหนดระบบ บันทึกข้อผิดพลาด บันทึกสิ่งผิดพลาด รายงาน ออกจากระบบ

วันที่หยุด: 10/01/1997

จำนวนวัน: วันเด็ก

วันที่	เดือน	ปี	จำนวน
10	01	1997	วันเด็ก
02	02	1997	วันตรุษจีน
31	12	1997	วันสิ้นปี

CAPS 29/10/97 22:43:19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกลูกจ้าง

ระบบคำนวณเวลา - [บันทึกลูกจ้าง]

ภาพหน้าจอระบบ บันทึกข้อมูลลูกจ้าง บันทึกข้อมูลช่วงเวลา รายงานงาน ออกจากระบบ

รหัส: 111111

ชื่อ: อ้อมรัก

นามสกุล: วงศ์กิม

ตำแหน่งประจำชื่อ

นาย

นาง

นางสาว

กลุ่มเข้างาน: 08:30

รหัสมีบัตร: กพร.บ.

เงินเดือน: 12000

จำนวนชม สูงสุด: 30 ชม.

รหัส	ตำแหน่ง	ชื่อ
111111	นางสาว	อ้อมรัก
222222	นางสาว	กนกนวล

CAPS 14/11/97 22:52:00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกทรัพย์สินกัก

ระบบกล้องเวลา - [บันทึกทรัพย์สินกัก]

วิเทศระบบ - บันทึบบันทึก - บันทึกทรัพย์สินกัก - ระบบงาน - 000จากระบบ

รหัสสังกัด: 6140000

ชื่อสังกัด: ฝ่ายระบบและระเบียบงาน

ชื่อย่อยสังกัด: ผรบ.

รหัสสังกัด	ชื่อสังกัด
6140000	ฝ่ายระบบและระเบียบงาน
6140100	กองทัพนโยบายระบบงาน
6140101	แผนกนิเทศาภิบาล
6140102	แผนกวิเคราะห์ระบบงาน
6140103	แผนกนิเทศาภิบาลโปรแกรมสำเร็จรูป
6140104	แผนกวิชาการคอมพิวเตอร์
6140200	กองควบคุมและประมวลผล
6140201	แผนกบันทึกข้อมูล

CAPS 29/10/97 22:44:32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกตารางเวลาทำงาน

ระบบคำนวณเวลา - บันทึกตารางเวลาทำงาน

กำหนดระบบ: บันทึกข้อมูลหลัก บันทึกตารางเวลา รายงาน ออกจากระบบ

กลุ่มงาน:

เวลาเริ่มต้น:

เวลาเลิกงาน:

สัปดาห์งาน	เวลาเริ่มต้น	เวลาเลิกงาน
1	07:30	15:30
2	08:30	16:30
3	09:30	17:30

CAPS 29/10/97 22:44:52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกอัตรา

ระบบกำลังเวลา - บันทึกอัตรา

กำหนดระบบ: บันทึกข้อมูลหลัก บันทึกข้อมูลรอง รายงาน ออกจากระบบ

รหัสอัตรา	อัตราค่าจ้างเวลา	ค่าปรับ
1	1	ทำในวันหยุด แต่ไม่
1.5	2	ทำนอกเวลาทำการ 1
2	3	ลูกจ้างทำในวันหยุด
3	4	พนักงานทำในวันหยุด

รหัสอัตรา: 1

อัตรา OT: 1

ค่าปรับ: ทำในวันหยุด แต่ไม่เกิน 7 ชม.

CAPS 29/10/97 22:45:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกค่าล่วงเวลา

ระบบค่าล่วงเวลา - [บันทึกค่าล่วงเวลา]

กำหนดระบบ บันทึกข้อมูลหลัก บันทึกค่าล่วงเวลา รายงาน ออกจากระบบ

ประวัติวันที่ 10 ตุลาคม 1997

รหัสผู้มีสิทธิ์งาน 502855

นางสาว นิลวิ วัฒนวงษ์วิเศษ

เวลา/งานที่ทำ

เวลาเริ่มต้น 15:30 เวลาสิ้นสุด 18:30

จำนวนชม. OT 3 อัตรา OT 1.5

งานที่ทำ ทาสอย

รหัสผู้มีสิทธิ์	เวลาเริ่มทำ	เวลาถึง	จำนวนชม. OT	อัตรา
507830	15:30	18:30	3	
502855	15:30	18:30	3	

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

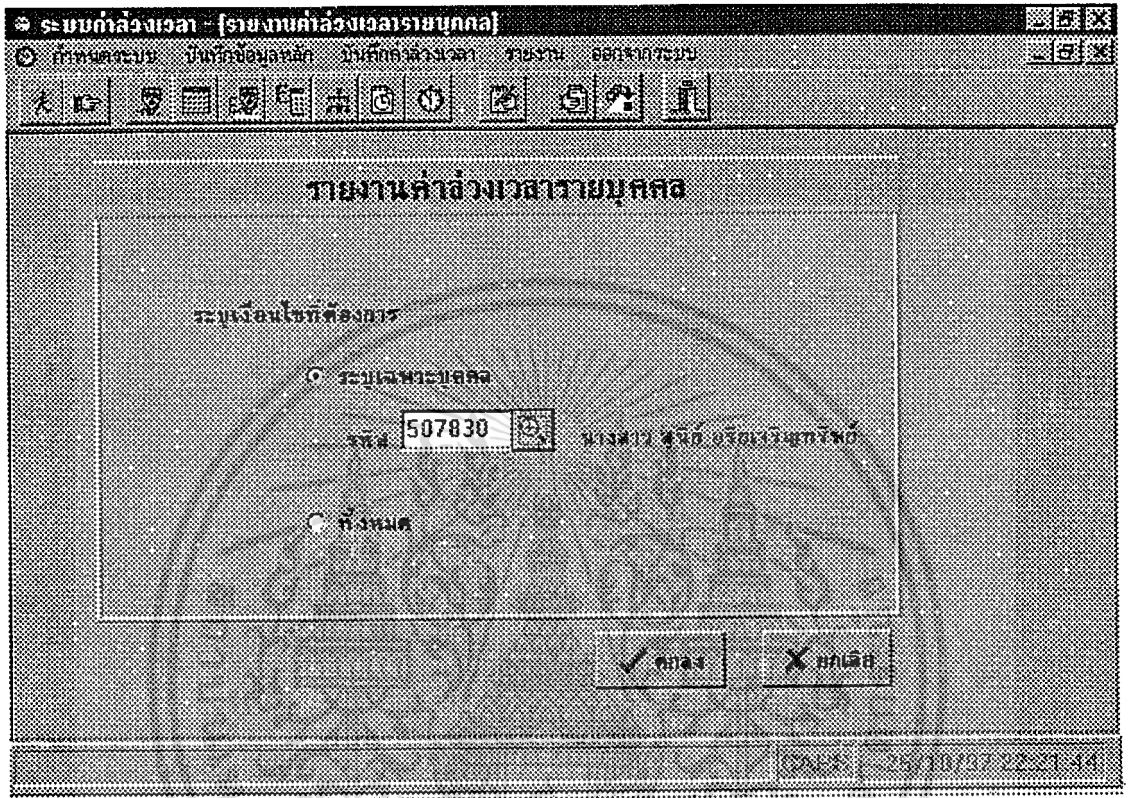
17/10/97 22:20:41

วิธีใช้งานเพิ่มเติม

1. การเลือกวันที่ เลือกโดย Click เม้าส์ ที่ช่องวันที่ที่ต้องการบันทึก ในปฏิทิน และในส่วนของการแสดงรายการทั้งหมด จะแสดงรายการเฉพาะข้อมูลของวันที่ที่เลือกในปฏิทิน เท่านั้น
2. การเลือกอัตราค่าล่วงเวลา จะมี Drop-down list ซึ่งจะแสดงข้อมูลอัตราค่าล่วงเวลาทั้งหมด ให้เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกรายงานค่าล่วงเวลารายบุคคล สามารถระบุเงื่อนไขที่ต้องการพิมพ์ได้ว่าจะพิมพ์ทั้งหมดหรือเฉพาะผู้ปฏิบัติงานบางคน



ซึ่งหลังจากเลือก ตกลง การออกรายงานแล้ว จะปรากฏ Window เหมือนการ Preview รายงานก่อนพิมพ์ และสามารถ Setup Printer ได้ด้วย ดังภาพที่จะแสดงต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วันที่	งานที่ทำ	เวลาเริ่มต้น	เวลาสิ้นสุด	จำนวนชม	อัตรา(ค่า)
18	ขานเงินเดือน	15:30	18:30	3	1.5
2	งานPayroll	17:30	20:30	3	1.5

กรมส่งเสริมการค้าระหว่างประเทศ
ในพระราชบัญญัติว่าด้วยการสงวนสิทธิ
ประจําเดือน ตุลาคม 1997

Page: 1
วันที่ 25/10/97

ชื่อ นางสาว ชุมนวีระศรีภูมิรักษ์ หมายเลขประจำตัว 507830
แผนก แผนกวิชาการคอมพิวเตอร์ จังหวัดยะลา ๖๖๐๐๐๖

Page 1 of 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกรายงานสรุปค่าล่วงเวลา

การไม่มีใบอนุญัตติขออนุญาต				Page: 1	
ในกรณีการมีค่าล่วงเวลาที่เกิดจากกรณีพิเศษ				วันที่ 29/10/97	
ประจำปีเดือน ตุลาคม 1997					
กองพัฒนาระบบงาน			ขอมติค่าล่วงเวลาและสิทธิการเบิกเงินทดแทน		
ชื่อ	หมายเลขประจำตัว	จำนวนขอมติค่าล่วงเวลาทั้งสิ้น	จำนวนเงิน		
นาย โอสถ	ศิริวิทยากร	436811	12	1636.36	
นางสาว พิชิ	จิตพรวิฑู	502855	165	1714.29	

Page 1 of 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปขั้นตอนการทำงานของระบบ

- กรณีมีการเริ่มเดือนใหม่ หรือครั้งแรกในการใช้งาน ต้องมีการบันทึกเพิ่มหลักต่าง ๆ ให้เรียบร้อย ซึ่งมีดังนี้
 - เพิ่มพนักงาน
 - เพิ่มลูกจ้าง
 - เพิ่มตารางเวลาทำงาน
 - เพิ่มสังกัด
 - เพิ่มวันหยุดพนักงาน
 - เพิ่มวันหยุดลูกจ้าง
 - เพิ่มอัตราค่าล่วงเวลา
- ในระหว่างเดือน บันทึกค่าล่วงเวลา ,การเปลี่ยนแปลงต่าง ๆในเพิ่มหลัก หรือ ออกรายงานเพื่อการตรวจสอบในเรื่องต่าง ๆ
- เมื่อสิ้นเดือน
 - ออกรายงานค่าล่วงเวลารายบุคคล เพื่อส่งให้ผู้ปฏิบัติงานตรวจสอบความถูกต้อง
 - ออกรายงานสรุปค่าล่วงเวลา ส่งแผนกเงินเดือน
 - ขึ้นเดือนใหม่ (จะไม่สามารถบันทึกข้อมูลเดือนใหม่ได้ ถ้าไม่ทำการเตรียมเพิ่มสำหรับเดือนใหม่ก่อน)

ประวัติผู้เขียน

นางสาวสุนีย์ อริยเจริญทรัพย์ เกิดเมื่อวันที่ 2 ตุลาคม พ.ศ. 2511 ที่จังหวัด กรุงเทพฯ จบการศึกษาปริญญาตรี สาขาวิชาบัญชี จากคณะบริหารธุรกิจ สถาบันเทคโนโลยีราชมงคล เข้าทำงานที่การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย เมื่อวันที่ 1 ตุลาคม พ.ศ. 2533 ในแผนกพัฒนาโปรแกรม โดยทำงานเกี่ยวกับการเขียนโปรแกรม ประมาณ 5 ปี จากนั้นได้ย้ายมาทำงานที่ แผนกวิชาการคอมพิวเตอร์ จนถึงปัจจุบัน โดยทำงานเกี่ยวกับการศึกษาเทคโนโลยีสมัยใหม่ เพื่อเผยแพร่ความรู้ให้กับหน่วยงานอื่น ๆ ต่อไป

