

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็ก II  
ACCESS CONTROL USING MAGNETIC CARD II



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมระบบควบคุม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2542

เลขหมั.....  
เลขทะเบียน 36875  
วัน, เดือน, ปี 29 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
โดยไม่ได้รับอนุญาตจากสำนักหอสมุดกลาง หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อสำนักหอสมุดกลาง โทร. 0-2616-2543

ปริญญาโทปีการศึกษา 2542

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็ก

Access Control Using Magnetic Card

ผู้จัดทำ

1. นางสาวเจนจิรา อิงสกุล
2. นางสาวญาณิศา คศรีเดช



.....อาจารย์ที่ปรึกษา

( อาจารย์เกียรติวรรณ ทรงสัตย์ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็ก

นางสาวเจนจิรา อิงสกุล

นางสาวญาณิ คลธิเดช

อาจารย์เกียรติวรรณ ทรงสัตย์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2542

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการนำเสนอถึงการสร้างซอฟต์แวร์ เพื่อใช้ในการควบคุมระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็ก โดยได้ทำการศึกษาเกี่ยวกับภาษาที่ใช้ในการสร้างซอฟต์แวร์สำหรับควบคุมระบบ และการสร้างฐานข้อมูล ตลอดจนการติดต่อสื่อสารทางพอร์ทอนุกรมเพื่อทำการโอนย้ายข้อมูลระหว่างบอร์ดหลักและเครื่องคอมพิวเตอร์ ดังนั้นในการสร้างซอฟต์แวร์ที่ใช้ในการควบคุมจึงจำเป็นต้องแยกเป็น 2 ส่วน กล่าวคือ ซอฟต์แวร์ที่ใช้ในการติดต่อสื่อสารกับผู้ใช้ผ่านทางคอมพิวเตอร์ (User Interface) และซอฟต์แวร์บนไมโครคอนโทรลเลอร์ ซึ่งเป็นหน่วยควบคุมของบอร์ดหลัก

### ABSTRACT

This thesis is presented about software for access control system using magnetic card. By studying the language that use to create software for control system , database and communication through serial port for transfer data between main board and computer. So that software can separate into 2 parts. One is software that use for user interface and the another is software for microcontroller on main board.

สารบัญ

	หน้า
บทคัดย่อ	i
สารบัญ	ii
สารบัญรูปภาพ	v
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขต	1
1.4 วิธีการ	2
1.5 ประโยชน์	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 หลักการทำงาน	3
2.2 โปรแกรมสนับสนุนการติดต่อกับผู้ใช้	5
2.2.1 สิ่งแวดล้อมในการพัฒนาแอปพลิเคชันของเดสก์	5
2.2.1.1 หน้าต่างหลัก	5
2.2.1.2 หน้าต่างแอปเจกอินสเปคเตอร์	6
2.2.1.3 หน้าต่างฟอร์ม	8
2.2.1.4 หน้าต่างโค้ดอิดิเตอร์	8
2.2.2 การใช้คอมโพเนนต์เบื้องต้น	9
2.2.2.1 ความรู้พื้นฐานของคอมโพเนนต์	9
2.2.2.2 คุณสมบัติทั่วไปของคอมโพเนนต์	9
2.2.2.3 ประเภทของคอมโพเนนต์และคุณสมบัติของ	9
คอมโพเนนต์ต่าง	11
2.3 ไมโครคอนโทรลเลอร์ตระกูล MCS51	25
2.3.1 คุณสมบัติของ MCS51	25
2.3.2 การติดต่อสื่อสารผ่านทางพอร์ทอนุกรม	26
2.3.3 โครงสร้างการอินเตอร์รัปต์	28
2.4 โปรแกรมฟอร์เซตพ 2.95	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.4.1 เซลฟไฟล์	30
2.4.2 รายละเอียดของโปรแกรมฟอร์เซตพ	30
2.5 มาตรฐานในการสื่อสารข้อมูล	34
2.5.1 มาตรฐานการสื่อสารข้อมูลอนุกรม RS-232	34
บทที่ 3 การออกแบบ	37
3.1 ระบบฐานข้อมูลในเดสทอป	37
3.1.1 ดาต้าเบสเดสทอป (Database Desktop)	38
3.1.2 การสร้างแอปพลิเคชันด้านฐานข้อมูล	38
3.1.3 ระบบฐานข้อมูล	39
3.1.3.1 ฐานข้อมูลนักศึกษา	39
3.1.3.2 ฐานข้อมูลผู้ใช้	41
3.1.3.3 ฐานข้อมูลรหัสที่ระบบทำการสร้างขึ้น	42
3.1.3.4 ระบบรักษาความปลอดภัยของฐานข้อมูล	43
3.1.4 ระบบฐานข้อมูลบนบอร์ดหลัก	44
3.1.4.1 ฐานข้อมูลเก็บรายละเอียดของผู้ใช้	44
3.1.4.2 ฐานข้อมูลเก็บข้อมูลการใช้งานของผู้ใช้	45
3.2 ฟังก์ชันการทำงานของโปรแกรม	46
3.2.1 หน้าจอหลัก	49
3.2.2 ฟังก์ชันขอคู่มือ	50
3.2.2.1 ส่วนรายละเอียดแต่ละบุคคล	50
3.2.2.2 ส่วนบันทึกการใช้งาน	54
3.2.2.2.1 ฟังก์ชันคู่มือการใช้งานประจำวัน	54
3.2.2.2.2 ฟังก์ชันคู่มือการใช้งานย้อนหลัง	57
3.2.2.2.3 ฟังก์ชันคู่มือผู้ไม่ปิดเครื่อง	59
3.2.2.2.4 ฟังก์ชันคู่มือผู้ไม่มีสิทธิ์ใช้งาน	61
3.2.2.2.5 กราฟแสดงสถิติการใช้งาน	63
3.2.3 ฟังก์ชันแก้ไขเพิ่มเติมข้อมูล	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
3.2.3.1 ฟังก์ชันการเพิ่มผู้ใช้	65
3.2.3.2 ฟังก์ชันการแก้ไขข้อมูล	69
3.2.3.3 ฟังก์ชันการเปลี่ยนบัตรใหม่	71
3.2.3.4 ฟังก์ชันการลบข้อมูลผู้ใช้	72
3.2.4 ฟังก์ชันการถ่ายโอนข้อมูลเข้า (Upload)	75
3.2.5 ฟังก์ชันการถ่ายโอนข้อมูลออก (Download)	77
3.2.6 ฟังก์ชันเฮลพ	79
บทที่ 4 การทดลองและผลการทดลอง	80
4.1 ส่วนของโปรแกรมบนเครื่องคอมพิวเตอร์	80
4.1.1 ทำการทดลองส่งค่าออกทางพอร์ตอนุกรม	80
4.1.2 ทำการทดลองรับค่าจากพอร์ตอนุกรม	81
4.2 ส่วนของโปรแกรมบนบอร์ดหลัก	83
4.2.1 ส่วนของการติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ส่วนบุคคล	83
4.2.2 ส่วนของการทำงานกับฐานข้อมูล	85
4.3 การทดลองระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลและบอร์ดหลัก	86
บทที่ 5 บทวิจารณ์และสรุป	87
5.1 ปัญหาและแนวทางแก้ไข	87
5.2 แนวทางการพัฒนา	88
กิตติกรรมประกาศ	
เอกสารอ้างอิง	
ภาคผนวก ก. แสดงแผนภาพการทำงาน	
ภาคผนวก ข. แสดงโปรแกรมบนเครื่องคอมพิวเตอร์ส่วนบุคคล	

## สารบัญรูปภาพ

รูปที่	หน้า
2.1 แสดงรูปแบบการทำงานของระบบควบคุมการเปิดปิดด้วยบัตรแม่เหล็ก	3
2.2 แผนผังแสดงการทำงานของบอร์ดหลักในการเปิด-ปิดเครื่องคอมพิวเตอร์	4
2.3 แสดงหน้าต่างหลักของโปรแกรมเตลไฟ	5
2.4 แสดงเมนู	6
2.5 แสดง Speedbar	6
2.6 แสดง Component Palette	6
2.7 แสดงหน้าต่าง Object Inspector	7
2.8 แสดงหน้าต่างฟอร์ม	8
2.9 แสดงส่วน Code Editor	8
2.10 แสดงหน้าจอ DBGrid	17
2.11 แสดงหน้าจอไดอะล็อกบ็อก Report Settings	20
2.12 แสดงหน้าจอของ PrintDialog	23
2.13 แสดงหน้าจอของ PrinterSetup Dialog	23
2.14 แสดงตำแหน่งหน่วยความจำของโปรแกรมบริการอินเทอร์เน็ตแต่ละชนิดใน MCS51	29
2.15 แสดงหน้าต่างของโปรเจกต์ออกพจน	32
2.16 แสดงตัวอย่างหน้าต่างของเฮลพไฟล์	33
2.17 แสดง RS-232 Connector	34
2.18 แสดงการส่งและการรับสัญญาณระหว่าง DTE และ DCE ในมาตรฐาน RS-232	35
3.1 แสดงหน้าจอของโปรแกรม Database Desktop	38
3.2 แสดงหน้าจอการสร้างฐานข้อมูลนักศึกษาใน Database Desktop	39
3.3 แสดงหน้าจอการสร้างฐานข้อมูลผู้ใช้ใน Database Desktop	41
3.4 แสดงหน้าจอการสร้างฐานข้อมูลที่ระบบทำการสร้างขึ้นใน Database Desktop	42
3.5 แสดงหน้าจอการเชื่อมโยงข้อมูลระหว่างฐานข้อมูล	43
3.6 แสดงแผนผังการทำงานของโปรแกรมตอบสนองการทำงานของผู้ใช้	47
3.7 แสดงแผนผังการทำงานของโปรแกรมตอบสนองการทำงานของผู้ใช้ (ต่อ)	48
3.8 แสดงหน้าจอหลักของโปรแกรม	49
3.9 แสดงหน้าจอส่วนรายละเอียดแต่ละบุคคล	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.10 แผนภาพแสดงการค้นหา	53
3.11 หน้าจอแสดงส่วนบันทึกการใช้งาน	54
3.12 แสดงหน้าจอของฟังก์ชันข้อมูลผู้ใช้งานประจำวัน	55
3.13 แผนภาพแสดงการทำงานของฟังก์ชันขอข้อมูลผู้ใช้ประจำวัน	56
3.14 แสดงหน้าจอของฟังก์ชันข้อมูลผู้ใช้งานย้อนหลัง	57
3.15 แผนภาพแสดงการทำงานของฟังก์ชันขอข้อมูลผู้ใช้งานย้อนหลัง	58
3.16 แสดงหน้าจอของฟังก์ชันข้อมูลนักศึกษาที่ไม่ได้ทำการปิดเครื่อง	59
3.17 แผนภาพแสดงการทำงานของฟังก์ชันขอข้อมูลนักศึกษาที่ไม่ปิดเครื่อง	60
3.18 แสดงหน้าจอของฟังก์ชันข้อมูลนักศึกษาที่ไม่มีสิทธิใช้งาน	61
3.19 แผนภาพแสดงการทำงานของฟังก์ชันขอข้อมูลนักศึกษาที่ไม่มีสิทธิใช้งาน	62
3.20 แสดงหน้าจอกราฟแสดงข้อมูลผู้ใช้	63
3.21 แสดงหน้าจอกราฟแสดงจำนวนผู้ใช้งาน	63
3.22 แสดงหน้าจอกราฟแสดงจำนวนผู้ใช้แยกตามสถานะการใช้งาน	64
3.23 แสดงหน้าจอของฟังก์ชันการเพิ่มข้อมูลผู้ใช้	65
3.24 หน้าจอแสดงสถานะการขอติดต่อบอร์ดหลัก	66
3.25 หน้าจอแสดงให้ผู้ใช้ทำการรูดบัตร	66
3.26 แสดงหน้าจอสิ้นสุดการเพิ่มข้อมูล	67
3.27 หน้าจอแสดงการส่งข้อมูลผิดพลาด	67
3.28 แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลักในการเพิ่มข้อมูล	68
3.29 หน้าจอแสดงฟังก์ชันการแก้ไขข้อมูล	69
3.30 แสดงหน้าจอสถานะการรับ-ส่งข้อมูล	70
3.31 แสดงหน้าจอการแก้ไขข้อมูลเรียบร้อยแล้ว	70
3.32 แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลักในการแก้ไขข้อมูล	71
3.33 แสดงหน้าจอสิ้นสุดการเปลี่ยนบัตร	71
3.34 แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลักในการเปลี่ยนบัตร	72
3.35 แสดงฟังก์ชันการลบผู้ใช้	73
3.36 แสดงหน้าจอสิ้นสุดการลบข้อมูล	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.37 แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลักในการลบข้อมูล	74
3.38 แสดงหน้าจอฟังก์ชันการถ่ายโอนข้อมูลเข้า	75
3.39 แสดงหน้าจอไม่พบข้อมูลในฐานข้อมูล	76
3.40 แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลักในการถ่ายโอนข้อมูลเข้า	77
3.41 แสดงฟังก์ชันการถ่ายโอนข้อมูลออก	77
3.42 แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลักในการถ่ายโอนข้อมูลออก	78
4.1 แสดงหน้าจอโปรแกรมสำเร็จรูปที่ใช้ในการติดต่อสื่อสารทางพอร์ทอนุกรมของคอมพิวเตอร์	84

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมา

เนื่องจากในปัจจุบัน ไมโครคอนโทรลเลอร์ (MICROCONTROLLER) ได้เข้ามามีบทบาทสำคัญในงานต่าง ๆ มากยิ่งขึ้น ไม่ว่าจะเป็นในด้านการจัดการฐานข้อมูลต่าง ๆ การตรวจสอบข้อมูลหรือระบบรักษาความปลอดภัย สำหรับระบบควบคุมการเข้าออกด้วยบัตรแม่เหล็ก (Access Control Using Magnetic Card) จัดเป็นอีกตัวอย่างหนึ่ง ที่แสดงให้เห็นถึงการนำไมโครคอนโทรลเลอร์มาประยุกต์ใช้งานกับงานด้านการรักษาความปลอดภัย เนื่องจากไมโครคอนโทรลเลอร์ สามารถทำการประมวลผลข้อมูลจากฐานข้อมูลขนาดใหญ่ได้ภายในเวลาอันรวดเร็ว และมีประสิทธิภาพสูงกว่าระบบที่ใช้คนในการตรวจสอบ

ดังนั้น ในปฏิญานิพนธ์ฉบับนี้ จะเป็นการนำเสนอรายละเอียดของระบบควบคุมการเปิด-ปิดเครื่องคอมพิวเตอร์โดยใช้บัตรแม่เหล็กในการควบคุมห้องคอมพิวเตอร์รวม โดยทำการควบคุมที่แหล่งจ่ายไฟโดยตรง ซึ่งจะมีประโยชน์ในด้านการตรวจสอบข้อมูลของการเปิด และปิดคอมพิวเตอร์ โดยใช้ระบบจัดการฐานข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคล และความรู้ทางด้านไมโครคอนโทรลเลอร์มาประยุกต์ใช้งาน

#### 1.2 วัตถุประสงค์

การทำวิทยานิพนธ์เรื่องระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็กนี้ ได้กำหนดวัตถุประสงค์ไว้ดังนี้ คือ

1. ทำการออกแบบและจัดสร้างซอฟต์แวร์ (Software) ที่ทำการควบคุมระบบ และติดต่อใช้งานกับผู้ควบคุมระบบ
2. ทำการออกแบบและจัดสร้างระบบฐานข้อมูลของระบบ
3. สามารถติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) และไมโครคอนโทรลเลอร์ที่ใช้เป็นเครื่องเซิร์ฟเวอร์ (Server) บนบอร์ดหลัก ได้

#### 1.3 ขอบเขต

สำหรับหัวข้อวิทยานิพนธ์เรื่องระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็กนี้ ได้ทำการจัดแบ่งออกเป็น 2 ส่วน คือ ส่วนซอฟต์แวร์ และส่วนฮาร์ดแวร์ โดยที่แต่ละส่วนมีขอบเขตของการทำงานดังนี้

ส่วนซอร์ฟแวร์ เป็นส่วนโปรแกรมที่ทำการติดต่อกับผู้ใช้ ในการแสดงผลทางสถิติต่างๆ เป็นต้นว่า รายชื่อของนักศึกษาที่เข้ามาใช้งานในวันต่างๆ หรือ รายชื่อของนักศึกษาที่ไม่มีสิทธิ์ใช้งาน นอกจากนี้ในส่วนของซอร์ฟแวร์ยังต้องทำการติดต่อกับไมโครคอนโทรลเลอร์ ในการเปลี่ยนแปลงแก้ไขข้อมูลของฐานข้อมูลที่ใช้ในการควบคุมการเปิด - ปิดเครื่อง

ดังนั้นส่วนซอร์ฟแวร์จึงสามารถแยกออกเป็น 2 ส่วน คือซอร์ฟแวร์บนเครื่องคอมพิวเตอร์ ที่ใช้ในการติดต่อสื่อสารกับผู้ใช้ ( Use Interface ) และซอร์ฟแวร์บนไมโครคอนโทรลเลอร์ ซึ่งใช้ในการประมวลผลเพื่อควบคุมการเปิด - ปิดเครื่อง

ทางส่วนฮาร์ดแวร์ จะเป็นส่วนที่ทำการจัดสร้างบอร์ดหลัก และเครื่องเทอร์มินอล (Terminal) ขึ้นมาพร้อมทั้งทำการสร้างฟังก์ชันการทำงานพื้นฐานบนไมโครคอนโทรลเลอร์ เพื่อให้บอร์ดหลักสามารถควบคุมการทำงานของเครื่องเทอร์มินอลได้

#### 1.4 วิธีการ

ในการศึกษาและออกแบบระบบในส่วนของซอร์ฟแวร์นั้น ได้มีการศึกษาการเขียนโปรแกรมโดยใช้โปรแกรมเดลไฟ (Delphi 4) ซึ่งเป็น เครื่องมือพัฒนาแอปพลิเคชัน (Application) บนวินโดวส์ (Windows) ที่สามารถรองรับความสามารถต่าง ๆ ในวินโดวส์ได้อย่างครบถ้วน และโปรแกรมสำเร็จรูปในการจัดสร้างเฮल्पไฟล์ ( Help Files ) ในที่นี้เลือกใช้โปรแกรมฟอร์เฮल्प (ForcHelp2.95) เพื่อใช้ในการสร้างเฮल्पไฟล์ที่เป็นเครื่องมือช่วยสำหรับผู้เข้ามาใช้โปรแกรมในการควบคุมระบบ ให้สามารถทำความเข้าใจเกี่ยวกับโปรแกรมในการควบคุมระบบได้อย่างสะดวกและรวดเร็วยิ่งขึ้น นอกจากนี้ยังต้องมีการศึกษาเกี่ยวกับไมโครคอนโทรลเลอร์ (หลักการพื้นฐานและส่วนประกอบ) รวมทั้งการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์และไมโครคอนโทรลเลอร์เพื่อให้สามารถทำการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลและบอร์ดหลักได้

#### 1.5 ประโยชน์

ระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็กนี้ จะสามารถนำไปใช้งานเพื่อประโยชน์ในด้านการตรวจสอบข้อมูล ,การประมวลผล ,การตรวจสอบความปลอดภัย ,การทำงานร่วมกันภายใต้ทรัพยากรเดียวกัน ฯลฯ ซึ่งจะทำให้เกิดความสะดวกรวดเร็วและการทำงานอย่างมีประสิทธิภาพมากยิ่งขึ้น

## บทที่ 2

### หลักการและทฤษฎี

#### 2.1 หลักการทำงาน

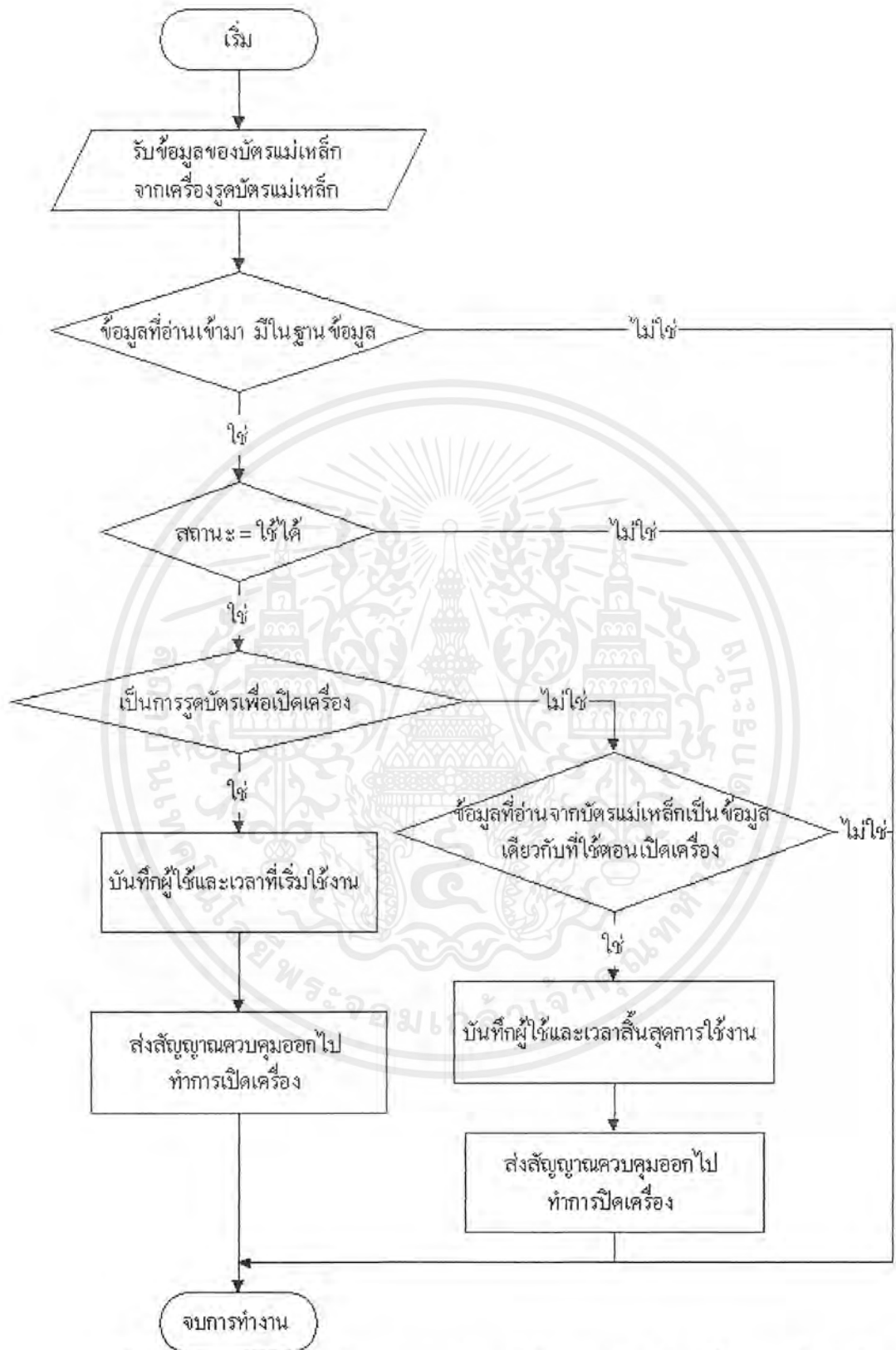


รูปที่ 2.1 แสดงรูปแบบการทำงานของระบบควบคุมการเปิดปิดด้วยบัตรแม่เหล็ก

หลักการการทำงานของระบบ คือเมื่อมีผู้ใช้ต้องการใช้งาน ผู้ใช้จะทำการรูดบัตรแม่เหล็กที่เครื่องรูดบัตร เครื่องรูดบัตรจะทำการอ่านข้อมูลขนาด 40 ไบต์จากบัตรแม่เหล็ก ส่งไปยังบอร์ดหลัก เพื่อให้บอร์ดหลักทำการประมวลผล ดังแผนผังในรูปที่ 2.2 นอกจากนี้เมื่อมีการรูดบัตรเพื่อทำการเปิดหรือปิดเครื่อง บอร์ดหลักจะทำการเก็บข้อมูลของผู้ใช้ เพื่อให้สามารถนำมาแสดงผลทางสถิติเป็นต้นว่า

- ข้อมูลผู้ใช้ประจำวัน
- ข้อมูลผู้ใช้อยอนหลัง
- ข้อมูลนักศึกษาที่ไม่มีสิทธิ์ใช้งาน
- ข้อมูลนักศึกษาที่ไม่ได้ทำการปิดเครื่อง

หรือ อาจจะแสดงจำนวนผู้ใช้ในรูปของกราฟ เพื่อความสะดวกในการเปรียบเทียบข้อมูล ดังนั้น สามารถสรุปได้ว่า บอร์ดหลักต้องทำการติดต่อกับเครื่องรูดบัตร เพื่อทำการอ่านข้อมูลซึ่งเป็นรหัสประจำตัวของผู้ใช้ขนาด 40 ไบต์ มาทำการประมวลผล ผ่านทางสายสัญญาณ RS-485 หลังจากนั้น ต้องมีการติดต่อกับคอมพิวเตอร์ ผ่านทางสายสัญญาณ RS-232-C เพื่อทำการเปิด-ปิด เครื่อง ตลอดจนการถ่ายโอนข้อมูลต่างๆ ดังรายละเอียดในบทที่ 3



รูปที่ 2.2 แผนผังแสดงการทำงานของบอร์ดหลักในการเปิด – ปิดเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

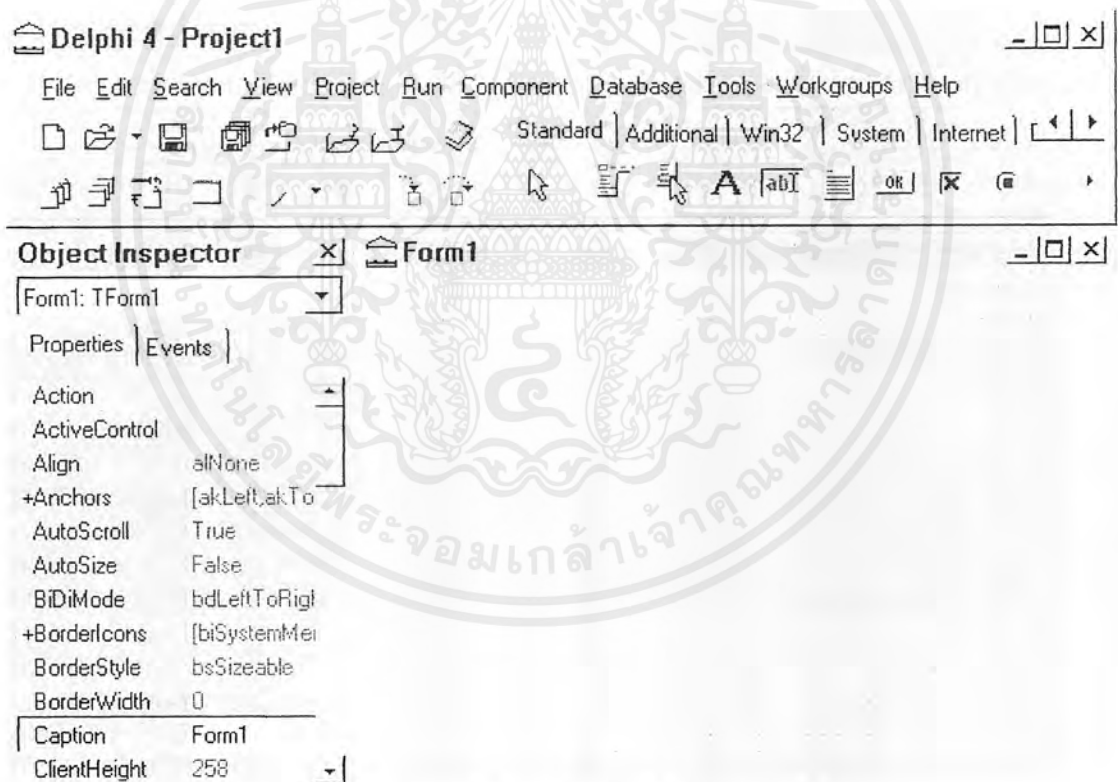
## 2.2 โปรแกรมสนับสนุนการติดต่อกับผู้ใช้

ในที่นี้ทางคณะผู้จัดทำได้เลือกใช้โปรแกรมเคลฟ เป็นโปรแกรมที่ใช้ติดต่อกับผู้ใช้ผ่านทางคอมพิวเตอร์ เนื่องจากเป็นเครื่องมือพัฒนาแอปพลิเคชันบนวินโดวส์ ที่มีประสิทธิภาพสูงสุดตัวหนึ่ง ซึ่งสามารถรองรับความสามารถต่างๆที่อยู่ในวินโดวส์ได้อย่างครบถ้วน และยังง่ายในการใช้งาน ทำให้สามารถพัฒนาแอปพลิเคชันต่างๆ ได้ในเวลาอันรวดเร็ว ไม่ว่าจะเป็นส่วนติดต่อกับผู้ใช้ ,การเชื่อมโยงฐานข้อมูล ตลอดจนการติดต่อสื่อสารผ่านทางพอร์ทอนุกรม โดยหลักการทำงานของโปรแกรมเคลฟ มีดังนี้

### 2.2.1 สิ่งแวดล้อมในการพัฒนาแอปพลิเคชันของเคลฟ

เมื่อทำการเรียกใช้โปรแกรมเคลฟ จะประกอบด้วยหน้าต่าง 4 หน้าต่างดังนี้

#### 2.2.1.1 หน้าต่างหลัก ( Main Window )



รูปที่ 2.3 แสดงหน้าต่างหลักของโปรแกรมเคลฟ

หน้าต่างหลักเป็นหน้าต่างแรกที่ปรากฏเมื่อเรียกใช้โปรแกรมเคลฟ ซึ่งประกอบไปด้วย 3 ส่วน คือ Menubar , Speedbar และ Component Palette โดยมีรายละเอียดดังนี้

- เมนู

File Edit Search View Project Run Component Database Tools Workgroups Help

รูปที่ 2.4 แสดงเมนู

เป็นส่วนหลักที่ใช้ในการทำงานของคำสั่งต่างๆ ซึ่งคำสั่งเกือบทั้งหมดสามารถเรียกใช้ผ่านเมนูได้

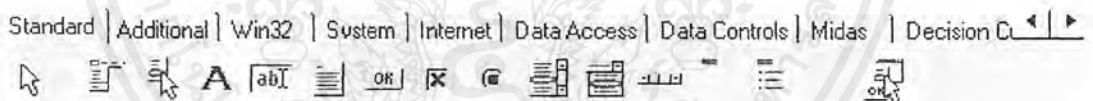
- Speedbar



รูปที่ 2.5 แสดง Speedbar

ประกอบไปด้วยปุ่มของคำสั่งที่ใช้งานบ่อยๆ สำหรับการทำงานของเคดไฟ ผู้ใช้สามารถใช้ในการเลือกกดเลือกปุ่มต่างๆ เพื่อเรียกคำสั่งเหล่านั้นมาใช้งาน

- Component Palette



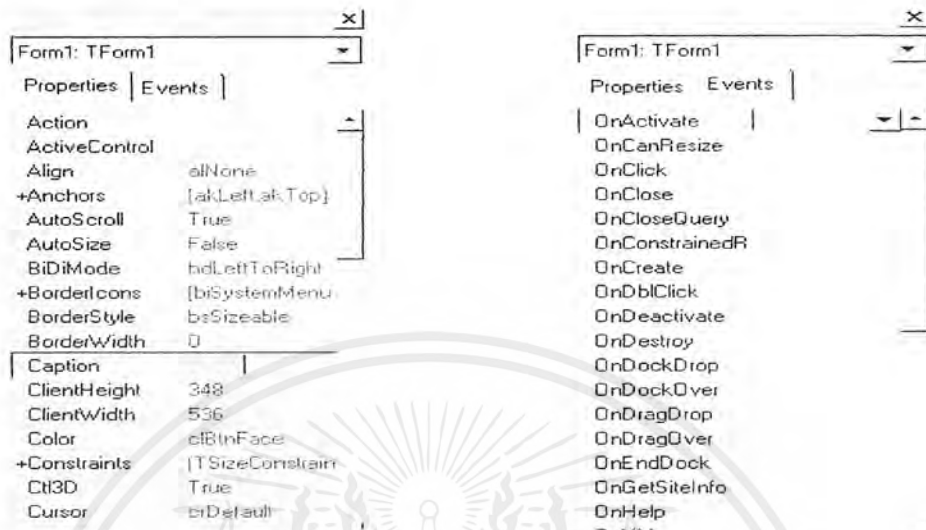
รูปที่ 2.6 แสดง Component Palette

ภายใน Component Palette จะมีไอคอนที่แสดงถึงคอมโพเนนต์ต่างๆ ในเคดไฟ โดยสามารถจัดแบ่งออกเป็นหลายกลุ่ม เช่น กลุ่ม Standard, กลุ่ม Additional ซึ่งในการเลือกใช้ ผู้ใช้ต้องทำการเลือกกลุ่มที่ต้องการใช้ก่อน โดยทำการเลือกที่ชื่อแท็บนั้นๆ จากนั้นทำการเลือกไอคอนที่ต้องการใช้งาน โดยการคลิกไปที่ตัวไอคอนนั้นมาวางลงบนฟอร์ม ในตำแหน่งที่ต้องการ

### 2.2.1.2 หน้าต่าง Object Inspector

Object Inspector เป็นหน้าต่างที่ใช้กำหนดคุณสมบัติ และ โปรซีเจอร์ที่ครอบคลุมเหตุการณ์ (Event) ต่างๆ ของคอมโพเนนต์ โดยภายใน Object Inspector จะประกอบไปด้วยแท็บ 2 แท็บดังแสดงในรูป 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงหน้าต่าง Object Inspector

- แท็บ Properties

จะแสดงคุณสมบัติ ( Properties ) ต่างๆของคอมโพเนนต์ โดยในแท็บนี้จะประกอบด้วยสองคอลัมน์ คือคอลัมน์ทางซ้าย จะเป็นส่วนแสดงชื่อของคุณสมบัติ และคอลัมน์ทางขวา จะเป็นส่วนแสดงค่าของคุณสมบัตินั้น ซึ่งผู้ใช้สามารถเป็นผู้กำหนดค่าเองได้

- แท็บ Events

เป็นแท็บที่ใช้กำหนดค่าต่างๆที่จะกระทำก็ต่อเมื่อเกิดเหตุการณ์หนึ่งๆขึ้นกับคอมโพเนนต์ โดยการกำหนดค่าต่างๆสามารถทำได้โดยการเขียนโปรแกรมตอบสนองเหตุการณ์ในหน้าต่างๆ

Code Editor

### 2.2.1.3 หน้าต่างฟอร์ม



#### รูปที่ 2.8 แสดงหน้าต่างฟอร์ม

เป็นส่วนที่ใช้ออกแบบส่วนที่ติดต่อกับผู้ใช้ ซึ่งสามารถนำคอมโพเนนต์ต่างๆ จาก Component Palette มาวางลงบนฟอร์ม โดยสามารถทำการเปลี่ยนขนาด, ย้ายตำแหน่ง, เพิ่ม หรือ ลบ คอมโพเนนต์ออกจากฟอร์มได้

### 2.2.1.4 หน้าต่าง Code Editor

```

Unit1
  TForm1
  Variables/Constants
  Uses
  interface
  uses
    Windows, Messages, SysUtils, Classes
  type
    TForm1 = class(TForm)
    private
      { Private declarations }
    public
      { Public declarations }
    end;
  var
    Form1: TForm1;
  
```

1: 1 Modified Insert

#### รูปที่ 2.9 แสดงส่วน Code Editor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Editor เป็น Text Editor สำหรับการเขียนโปรแกรมในเดสก์ท็อป เพื่อทำการเชื่อมต่อคอมโพเนนต์ต่างๆ ให้ทำงานร่วมกัน หรืออาจเป็นการเขียนโปรแกรมเพื่อตอบสนองเหตุการณ์จากแท็บ Event โดยลักษณะคำสั่งที่ใช้จะมีพื้นฐานมาจากภาษาปาสคาล ซึ่งแต่ละแท็บใน Code Editor จะแสดงถึงโปรแกรมในแต่ละฟอร์ม ดังนั้นจึงสามารถดูโปรแกรมของฟอร์มต่างๆ ได้ จากการคลิกเลือกแท็บ

## 2.2.2 การใช้คอมโพเนนต์เบื้องต้น

### 2.2.2.1 ความรู้พื้นฐานของคอมโพเนนต์

ในที่นี้จะขอพูดถึงคอมโพเนนต์ที่มีการใช้งานในระบบอย่างคร่าวๆ เท่านั้น ซึ่งในระบบนี้ใช้คอมโพเนนต์ต่างๆ ดังนี้

- Standard Component เป็นกลุ่มคอมโพเนนต์ที่เป็นมาตรฐานของวินโดวส์
- Data Access Component เป็นกลุ่มคอมโพเนนต์ที่ใช้ในการเข้าถึงฐานข้อมูล
- Data Control Component เป็นกลุ่มคอมโพเนนต์ที่ใช้ควบคุมฐานข้อมูล
- Qreport Component เป็นกลุ่มคอมโพเนนต์ที่ใช้สำหรับการทำรายงาน
- Dialogs Component เป็นกลุ่มคอมโพเนนต์ที่รวบรวมไดอะล็อกพื้นฐานที่ใช้ในการพัฒนาแอปพลิเคชัน
- Djan Component เป็นกลุ่มคอมโพเนนต์ที่ใช้ในการติดต่อสื่อสารทางพอร์ตอนุกรม

### 2.2.2.2 คุณสมบัติทั่วไปของคอมโพเนนต์

สำหรับคอมโพเนนต์โดยส่วนใหญ่จะมีคุณสมบัติที่เป็นคุณสมบัติพื้นฐาน ซึ่งสามารถแบ่งเป็นกลุ่มๆ ได้ ดังนี้

- คุณสมบัติที่เกี่ยวกับตำแหน่งและขนาด

คุณสมบัติ	ความหมาย
Height	กำหนดค่าความสูงของคอมโพเนนต์
Width	กำหนดค่าความกว้างของคอมโพเนนต์
Left	กำหนดตำแหน่งตามแนวนอนของคอมโพเนนต์
Top	กำหนดตำแหน่งตามแนวตั้งของคอมโพเนนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● คุณสมบัติที่เกี่ยวข้องกับลักษณะภายนอก

คุณสมบัติ	ความหมาย
BorderStyle	กำหนดกรอบให้คอมโพเนนต์ ถ้าเลือกเป็น <ul style="list-style-type: none"> <li>- bsSingle จะเกิดกรอบที่คอมโพเนนต์นั้น</li> <li>- bsNone จะไม่มีกรอบที่คอมโพเนนต์นั้น</li> </ul>
Color	กำหนดสีพื้นของคอมโพเนนต์ โดยสามารถทำการกำหนดได้เอง หรือเลือกจากสีตามค่ามาตรฐาน
Ctrl3D	ถ้ากำหนดให้เป็น True จะทำให้คอมโพเนนต์นั้นมีลักษณะเป็น 3 มิติ
Enable	ถ้ากำหนดให้เป็น True จะทำให้สามารถทำงานกับคอมโพเนนต์ได้ คือ มีการตอบสนองกับเมาส์ และคีย์บอร์ด
Font	กำหนด ชื่อ ,ขนาด ,รูปแบบ และสีของตัวอักษรที่ปรากฏในคอมโพเนนต์
Visible	ถ้ากำหนดให้เป็น True จะสามารถมองเห็นคอมโพเนนต์นั้น

● คุณสมบัติอื่นๆ

คุณสมบัติ	ความหมาย
Cursor	กำหนดลักษณะของเมาส์ขณะที่อยู่บนคอมโพเนนต์นั้น ซึ่งสามารถทำการกำหนดรูปไอคอนได้บน Control Panel
Hint	เป็นคุณสมบัติที่ใช้ในการเก็บข้อความที่แสดงเป็น Hint ซึ่งจะทำงานได้ก็ต่อเมื่อกำหนด ShowHint เป็น True ก่อน
PopupMenu	เป็นคุณสมบัติที่ใช้กำหนดคอมโพเนนต์ที่ใช้เป็นเมนูลอย ซึ่งจะปรากฏเมื่อมีการคลิกเมาส์ที่ปุ่มขวา แต่ต้องมีการสร้างคอมโพเนนต์ Popup Menu ก่อน
TabOrder	กำหนดลำดับการย้ายโฟกัส โดยใช้ Tab ซึ่งการเรียงลำดับจะเรียงจากน้อยไปมาก และจะทำงานก็ต่อเมื่อคุณสมบัติ TabStop เป็น True
TabStop	ถ้ากำหนดให้เป็น True จะมีการย้ายโฟกัสตามลำดับใน TabOrder

### 2.2.2.3 ประเภทของคอมโพเนนต์และคุณสมบัติของคอมโพเนนต์ต่างๆ

- Standard Component

[abI] อีดิท ( Edit )

เป็นเครื่องมือที่ใช้ในการรับข้อมูลจากผู้ใช้โปรแกรม และใช้แสดงผลข้อมูล แต่มีข้อจำกัดคือ สามารถรับข้อความได้เพียงครั้งละ 1 บรรทัดเท่านั้น

คุณสมบัติที่สำคัญของอีดิท

คุณสมบัติ	ความหมาย
Text	ใช้สำหรับเก็บค่าข้อความที่แสดงในอีดิท และเมื่อมีการแก้ไขข้อมูลในอีดิท ค่าข้อมูลที่ทำให้การเปลี่ยนแปลงจะสามารถตรวจสอบได้ทันที
MaxLength	ใช้กำหนดความยาวสูงสุดของข้อความที่แสดงในอีดิท
Modified	เป็นคุณสมบัติที่ใช้แสดงว่ามีการแก้ไขข้อความในอีดิทหรือไม่ ซึ่งเป็นคุณสมบัติที่จะพบเมื่อทำการเขียน โปรแกรม แต่จะไม่พบในหน้าต่าง Object Inspector
ReadOnly	ถ้าเป็น True จะแสดงว่าไม่สามารถทำการแก้ไขข้อมูลใดๆใน Text ได้

เมคคอดที่สำคัญของอีดิท

เมคคอด	ความหมาย
Clear	ใช้สำหรับลบข้อความใน Text ซึ่งมีวิธีการใช้ดังนี้ (ชื่ออีดิทที่ต้องการลบ).CLEAR;
ClearSelection	มีลักษณะคล้ายเมคคอด Clear แต่จะลบเฉพาะข้อความที่ถูกเลือกเท่านั้น มีวิธีการใช้ดังนี้ (ชื่ออีดิทที่ต้องการลบ).CLEARSELECTION

อีเวนต์ที่สำคัญของอีดิท

อีเวนต์	ความหมาย
OnChange	จะเป็นการตอบสนองเมื่อมีการเปลี่ยนแปลงข้อความในอีดิท

### ๑๕) ปุ่ม ( Button )

เป็นคอมโพเนนต์ที่ใช้ในการติดต่อกับผู้ใช้งาน จะมีลักษณะเป็นปุ่มให้คลิก และเมื่อมีการคลิกปุ่ม จะเกิดอีเวนต์ OnClick ขึ้น

คุณสมบัติที่สำคัญของปุ่ม

คุณสมบัติ	ความหมาย
Cancel	เป็นคุณสมบัติที่บอกว่าเมื่อมีการกด Esc ที่คีย์บอร์ด จะมาทำงานที่คำสั่งนี้
Caption	เป็นการแสดงข้อความบนปุ่ม โดยสามารถกำหนด Hot Key ได้โดยทำการใส่ "&" หน้าอักษรที่ต้องการใช้เป็น Hot Key
ModalResult	เป็นคุณสมบัติที่ใช้สำหรับกำหนดว่าเมื่อทำการปิดหน้าต่างนั้นๆ หน้าต่างหลัก จะรู้ว่าหน้าต่างที่ถูกปิดไปนั้นมีค่าสถานะเป็นอะไร

อีเวนต์ที่สำคัญของปุ่ม

อีเวนต์	ความหมาย
OnClick	จะเกิดอีเวนต์นี้เมื่อมีการคลิกปุ่ม
OnEnter	จะเกิดเมื่อมีการกำหนดค่าโฟกัสมาที่ปุ่มนี้ หรือใช้ปุ่มที่เปลี่ยนมาถึงปุ่มนี้
OnExit	ถ้าเดิมมีการกำหนดให้ค่าโฟกัสให้ปุ่มนี้ เมื่อมีการเลื่อนโฟกัสไปปุ่มอื่น จะเกิดอีเวนต์นี้ขึ้น

### ● Data Access Component


#### ↳ Data Source

เป็นคอมโพเนนต์ที่สำคัญ สำหรับงานด้านพัฒนาฐานข้อมูล เนื่องจาก DataSource เป็นตัวกลางในการเชื่อมต่อระหว่าง DataSet ( Table หรือ Query ) กับคอมโพเนนต์ประเภท Data Control ที่ใช้สำหรับการแสดงหรือแก้ไขข้อมูล

คอมโพเนนต์ประเภท Data Control ทุกตัว จะอ้างถึงออบเจกต์ DataSource ซึ่งออบเจกต์ DataSource จะไปอ้างถึง DataSet ที่ต้องการติดต่ออีกทีหนึ่ง โดยในการใช้ DataSource ผู้ใช้ต้องทำการกำหนดคุณสมบัติของ DataSet ให้ชี้ไปยังออบเจกต์ Table หรือ Query ที่ต้องการเพื่อเป็นการระบุที่มาของข้อมูลที่จะใช้

## คุณสมบัติที่สำคัญของ Data Source

คุณสมบัติ	ความหมาย
DataSet	ระบอบประเภท DataSource ที่อ้างถึงข้อมูลที่จะนำมาใช้

 ตาราง ( Table )

เป็นคอมโพเนนต์ที่ใช้ในการเข้าถึงตารางฐานข้อมูลโดยตรง โดยสามารถทำการแก้ไข, ค้นหา, คัดลอก และลบข้อมูลในตาราง สามารถกำหนดให้มีการกรองเอาเฉพาะข้อมูลที่ต้องการ โดยในการสร้างตาราง ผู้ใช้ต้องทำการกำหนดคุณสมบัติต่างๆ ของตารางดังนี้

## คุณสมบัติที่สำคัญของตาราง

คุณสมบัติ	ความหมาย
DatabaseName	สำหรับกำหนดชื่อของฐานข้อมูลที่ต้องการจะเชื่อมต่อ
TableName	ชื่อของตารางฐานข้อมูล
Active	ถ้ากำหนดเป็น True จะเป็นการเปิดใช้งานตาราง
ReadOnly	ถ้ากำหนดเป็น True จะทำให้ไม่สามารถแก้ไขข้อมูลในตารางได้
Exclusive	ถ้ากำหนดเป็น True จะเป็นการล็อกตารางไม่ให้ผู้อื่นสามารถเข้ามาใช้งานตารางนี้ได้
FieldValue	สำหรับอ้างถึงข้อมูลฟิลด์ต่างๆของเรคคอร์ด โดยมีรูปแบบการใช้งานดังนี้ <code>Table1.FieldValuel['CustNo'] = '1000';</code>
Fields	สำหรับอ้างถึงข้อมูลฟิลด์ต่างๆของเรคคอร์ดปัจจุบัน โดยการระบุลำดับของฟิลด์ โดยมีรูปแบบการใช้งานดังนี้ <code>Table1.Fields[0].AsString = '1000';</code>
IndesFieldName	สำหรับระบุชื่อฟิลด์ที่ใช้เป็นดัชนี (Index) โดยที่ดัชนีเหล่านี้ต้องมีการสร้างเอาไว้ในฐานข้อมูลอยู่แล้ว

## เมดอดที่สำคัญของตาราง

เมดอด	ความหมาย
FieldByName	สำหรับดึงข้อมูลมาใช้ โดยระบุชื่อฟิลด์ มีรูปแบบการใช้งานดังนี้ <code>Table1.FieldByName('CustNo').AsString = '1000';</code> ลักษณะนี้เป็นการกำหนดค่าในฟิลด์ CustNo เหมือนคุณสมบัติ Field

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมคคอด	ความหมาย
Open	เปิดตารางขึ้นมาใช้งาน ( เหมือนการกำหนดคุณสมบัติ Active = True )
Close	ปิดตาราง ( เหมือนการกำหนดคุณสมบัติ Active = False )
Refresh	อ่านข้อมูลจากตารางใหม่ ใช้ในกรณีที่มีการแก้ไขข้อมูลจากที่อื่น
Append.Insert	เพิ่มเรคคอร์ดที่ตำแหน่งท้ายตาราง , เพิ่มเรคคอร์ดที่ตำแหน่งปัจจุบัน
Edit	เข้าสู่สถานะการแก้ไขข้อมูล
Delete	ลบเรคคอร์ดปัจจุบัน
Post	จัดเก็บข้อมูลที่เพิ่มหรือแก้ไข ถ้ามีการเลื่อนข้อมูลและข้อมูลถูกแก้ไขอยู่ เมคคอดนี้จะถูกเรียก โดยอัตโนมัติ
Cancel	ยกเลิกข้อมูลที่เพิ่มหรือแก้ไข
First	ไปยังเรคคอร์ดแรกของตาราง
Last	ไปยังเรคคอร์ดสุดท้ายของตาราง
Prior	ไปยังเรคคอร์ดก่อนหน้า
Next	ไปยังเรคคอร์ดถัดไป
FindKey	ค้นหาข้อมูลตามดัชนีที่ระบุไว้ในคุณสมบัติ IndexFieldName หรือ IndexFileName ถ้าค้นหาพบ จะเลื่อนไปยังเรคคอร์ดแรกที่พบ และคืนค่า True แต่ถ้าค้นหาไม่พบจะอยู่ที่เดิมและคืนค่า False มีรูปแบบดังนี้ <code>Table1.FindKey['1000'];</code>
FindNearest	เหมือน FindKey แต่ถ้าไม่พบข้อมูลที่ระบุ จะทำการหาข้อมูลใกล้เคียงแทน

การเรียงลำดับข้อมูลในคอม โพนেন্টตาราง

ปกติเมื่อเราใช้คอม โพนেন্টนี้ จะสามารถเข้าถึงข้อมูลในฐานข้อมูลได้ หลักจากที่ทำการเปิดตาราง เมื่อทำการเปิดตาราง ข้อมูลในตารางจะถูกจัดเรียงตามไพรมารีคีย์หรือไม่มีการจัดเรียงข้อมูลเลย ขึ้นอยู่กับประเภทฐานข้อมูลที่ใส่ แต่สำหรับตารางที่สกีการสร้างอินเด็กซ์ไว้ ผู้ใช้จะสามารถระบุให้มีการเรียงลำดับข้อมูลในตารางตามอินเด็กซ์ที่มีอยู่ได้ โดยใช้คุณสมบัติ IndexName ในการระบุชื่ออินเด็กซ์ไฟล์ หรือใช้คุณสมบัติ IndexsFieldNames ในการระบุชื่อฟิลด์หรือกลุ่มของชื่อฟิลด์ที่ใช้เป็นตัวเรียงข้อมูล และในกรณีที่ต้องการกำหนดให้ฟิลด์ที่ใช้ในการเรียงข้อมูลมากกว่าหนึ่งฟิลด์ ก็สามารถทำได้โดยคั่นชื่อฟิลด์เหล่านั้นด้วยเครื่องหมายเซมิโคลอน (:)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การค้นหาข้อมูลในตาราง

ในการค้นหาข้อมูลในตารางที่มีการสร้างอินเด็กซ์ไว้ ผู้ใช้สามารถทำการค้นหาได้โดยใช้เมธอด `FindKey` ,`FindNearest` ในการค้นหาข้อมูลได้ และสำหรับการค้นหาข้อมูลที่ไม่มีการสร้างอินเด็กซ์ไว้ ผู้ใช้ก็สามารถใช้เมธอด `Locate` ในการค้นหาได้

## การกรองข้อมูลในตาราง

ปกติเมื่อผู้ใช้เปิดตารางขึ้นเพื่อทำการใช้งาน จะเป็นการดึงข้อมูลทั้งหมดที่มีอยู่ขึ้นมาใช้งาน แต่ในกรณีที่ผู้ใช้ต้องการดึงข้อมูลมาใช้งานเพียงบางส่วน จึงจำเป็นต้องมีการกรองข้อมูลในตารางก่อน โดยวิธีที่ใช้ในการกรองข้อมูลมี 2 วิธี คือ

### 1. คุณสมบัติ Filter

ผู้ใช้สามารถระบุเงื่อนไขในการกรองข้อมูลในคุณสมบัติ Filter ได้ทันที โดยการ

ใช้คำสั่ง

`State = ( ค่าที่ต้องการกรอง )`

ซึ่งจะเป็นการระบุค่าในหน้าต่าง Object Inspector แต่การใช้คุณสมบัติ Filter ในการกรองข้อมูลนี้ต้องทำการกำหนดคุณสมบัติ Filtered เป็น True

### 2. อีเวนต์ OnFilterRecord

เนื่องจากการกรองข้อมูลด้วยคุณสมบัติ Filter เป็นการกรองอย่างง่าย ๆ จึงไม่สามารถสร้างเงื่อนไขการกรองข้อมูลที่ซับซ้อนได้ จึงต้องใช้อีเวนต์ `OnFilterRecord` เป็นตัวจัดการ โดยผู้ใช้จะต้องเขียนโปรแกรมการทำงานเมื่อเกิดอีเวนต์นี้ ในหน้าต่าง Code Explorer

และเมื่อใช้อีเวนต์นี้ในการกรอง ก็ยังมีความจำเป็นที่จะต้องกำหนดคุณสมบัติ Filtered ให้เป็น True ด้วย จึงจะสามารถทำการเรียกใช้อีเวนต์นี้ได้ โดยอีเวนต์นี้จะถูกเรียกใช้ทุกครั้งที่มีการเลื่อนไปยังเรคคอร์ดต่างๆของตาราง ซึ่งโปรแกรมที่ผู้ใช้เขียนขึ้นต้องมีการกำหนดค่าพารามิเตอร์ `Accept` ให้เป็น True ในเรคคอร์ดที่ต้องการให้แสดงผล และเป็น False ในเรคคอร์ดที่ไม่ต้องการให้มีการแสดงผล ดังตัวอย่างต่อไปนี้เป็นการกรองข้อมูล โดยเลือกให้มีการแสดงผลผู้ใช้ที่ใช้งานในปัจจุบัน

```
procedure TForm4.Table1FilterRecord(DataSet: TDataSet; var Accept: Boolean);
```

```
begin
```

```
    today := datetostr(now); // เป็นการอ่านค่าวันที่จากฐานข้อมูลเวลา
```

```
    accept := dataset['date'] = strtodate(today); // เป็นการกรองข้อมูลเฉพาะวันที่ปัจจุบันมาแสดงผล
```

```
end;
```



### Query

Query เป็นการเข้าถึงข้อมูลที่สำคัญอีกแบบหนึ่ง ซึ่งการเข้าถึงข้อมูลด้วยวิธีนี้ จำทำผ่านคำสั่งภาษา SQL ซึ่งภาษา SQL ที่ใช้ใน Query สามารถแยกได้ 2 ประเภท ดังนี้

1. คำสั่ง SELECT ซึ่งใช้ในการดึงข้อมูลจากฐานข้อมูลมาใช้งาน
2. คำสั่งอื่นๆ ที่ใช้ในการแก้ไขข้อมูล หรือ โครงสร้างข้อมูลในฐานข้อมูล ซึ่งคำสั่งเหล่านี้จะ

ไม่มีการดึงข้อมูลจากฐานข้อมูลมาใช้

คุณสมบัติที่สำคัญของ Query

คุณสมบัติ	ความหมาย
DatabaseName	สำหรับกำหนดชื่อฐานข้อมูลที่ต้องการเชื่อมต่อ
SQL	สำหรับใส่คำสั่งภาษา SQL
Active	ถ้ากำหนด Active เป็น True จะสามารถเปิดใช้งาน Query ซึ่งคำสั่งนี้จะสามารถใช้งานได้ต่อเมื่อมีการระบุคำสั่ง Select ในคุณสมบัติ SQL เท่านั้น
Params	แสดงพารามิเตอร์ที่อยู่ใน Query
RequistLive	ถ้ากำหนดเป็น True จะสามารถทำการแก้ไขข้อมูลได้
State	แสดงสถานะของ Query

เมธอดที่สำคัญของ Query

เมธอด	ความหมาย
Open	เปิด Query ขึ้นมาใช้งาน ( เหมือนกำหนดคุณสมบัติ Active เป็น True )
Close	ปิด Query ( เหมือนกำหนดคุณสมบัติ Active เป็น False )
ExecSQL	สั่งให้ทำงานตามคำสั่ง SQL ที่ระบุ
Refresh	อ่านข้อมูลจากฐานข้อมูลใหม่ ใช้ในกรณีที่ตารางมีการแก้ไขจากที่อื่น
Append, Insert	เพิ่มเรคคอร์ดที่ท้ายตาราง , เพิ่มเรคคอร์ดที่ตำแหน่งปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมคคอด	ความหมาย
Edit	แก้ไขข้อมูล ปกติเมคคอดนี้จะถูกเรียกใช้เองเมื่อมีการเปลี่ยนแปลงข้อมูลบนฟอร์ม
Delete	ลบเรคคอร์ดปัจจุบัน
Post	จัดเก็บข้อมูลที่เพิ่มหรือแก้ไข ถ้ามีการเลื่อนข้อมูล และมีข้อมูลที่ถูกแก้ไขอยู่
Cancel	ยกเลิกข้อมูลที่เพิ่มหรือแก้ไข
First	ไปยังเรคคอร์ดแรก
Last	ไปยังเรคคอร์ดสุดท้าย
Prior	ไปยังเรคคอร์ดก่อนหน้า
Next	ไปยังเรคคอร์ดถัดไป

● Data Control Component

DBGrid

DBGrid เป็นเหมือนตาราง สำหรับแสดงและแก้ไขข้อมูลในฐานข้อมูลซึ่ง DBGrid จะครอบคลุมการทำงานเกือบทั้งหมดในการจัดการกับข้อมูล ไม่ว่าจะเป็นการเลือกข้อมูล , การแสดงข้อมูล , การแก้ไขข้อมูล และการลบข้อมูล โดยภายใน DBGrid จะประกอบด้วยคีย์คอล์มต่างๆ ดังรูป

Id	Sex	Name	Year	Status	
▶ 39014007	นาย	กรกิจ เงามณีจกุล	4	ใช้ได้	(
39014022	นาย	กานต์ พฤษชากร	4	ใช้ไม่ได้	(
39014036	นาย	เกรียงศักดิ์ เท่าอาสาสงวงศ์	4	ใช้ไม่ได้	(
39014088	นาย	จุฑาพล คล้ายจำแรง	4	ใช้ไม่ได้	(
39014093	นางสาว	เจนจิรา อึ้งสกุล	4	ใช้ได้	(
39014101	นางสาว	ฉัตรดีแก้ว จิรไพโรจน์	4	ใช้ได้	(
39014118	นาย	ชวลิต ศรีสมยุปต์	4	ใช้ได้	(
39014123	นาย	ชاکริต กิจแก้ว	4	ใช้ได้	(
39014142	นางสาว	ภาณี ดลริเดช	4	ใช้ได้	(
40014073	นาย	จักรวาล วิจารณ์บุตร	3	ใช้ไม่ได้	(
40014074	นาย	จักรวาล ทัมเกตุ	3	ใช้ได้	(

รูปที่ 2.10 แสดงหน้าจอ DBGrid

แต่ละคอลัมน์จะอ้างอิงถึงฟิลด์ข้อมูล ซึ่งผู้ใช้สามารถกำหนดคุณสมบัติต่างๆของคอลัมน์เหล่านี้ได้ เช่น กำหนดฟิลด์ที่ความคุม , ลักษณะ , ตัวอักษร และสีที่ใช้แสดง เป็นต้น

ในการใช้งาน DBGrid ผู้ใช้ต้องกำหนดคุณสมบัติ DataSource ไปยังออปเจกต์ DataSource ที่อ้างอิงถึงข้อมูลที่ต้องการ

#### คุณสมบัติที่สำคัญของ DBGrid

คุณสมบัติ	ความหมาย
DataSource	ระบุออปเจกต์ DataSource ที่อ้างอิงถึงข้อมูลที่จะนำมาใช้ใน DBGrid
Columns	กลุ่มของออปเจกต์ DBGridColumn สำหรับกำหนดคุณสมบัติของแต่ละคอลัมน์ใน DBGrid
Options	สำหรับกำหนดรูปแบบการทำงานของ DBGrid ซึ่งเมื่อทำการดับเบิลคลิก ใน Object Inspector จะแสดงคุณสมบัติย่อยอื่นๆ
SelectedField	ใช้อ้างอิงฟิลด์ข้อมูลที่ถูกเลือกอยู่ใน DBGrid
SelectedIndex	ลำดับของฟิลด์ที่ถูกเลือกอยู่
ReadOnly	ถ้า ReadOnly เป็น True จะไม่สามารถทำการบันทึกหรือแก้ไขข้อมูลได้

#### DBEdit

เมื่อผู้ใช้สร้างออปเจกต์ DBEdit แล้ว ต้องทำการกำหนดคุณสมบัติ DataSource ของ DBEdit ไปยังออปเจกต์ DataSource ที่อ้างอิงถึงข้อมูลที่ต้องการ และกำหนดคุณสมบัติ DataField เป็นชื่อฟิลด์ที่ต้องการอ้างอิง

#### คุณสมบัติที่สำคัญของ DBEdit

คุณสมบัติ	ความหมาย
DataSource	ระบุออปเจกต์ DataSource อ้างอิงถึงข้อมูลที่ต้องการ
DataField	ชื่อของฟิลด์ที่ต้องการอ้างอิง
ReadOnly	ถ้า ReadOnly เป็น True จะไม่สามารถแก้ไขข้อมูลใน DBEdit ได้
EditText	เก็บข้อความที่แสดงอยู่บน DBEdit
Modified	ถ้ามีการแก้ไขข้อมูลใน DBEdit คุณสมบัติ Modified จะมีค่าเป็น True

- Qreport Component

Quick Report จะแบ่งรายงานออกเป็นส่วนๆ ตามลักษณะหน้าที่ และรูปแบบการแสดงผล เช่น ส่วนชื่อเรื่อง , หัวกระดาษ , ท้ายกระดาษ , หัวคอลัมน์ และส่วนรายละเอียดข้อมูล ซึ่งส่วนต่างๆเหล่านี้เรียกว่าแบนด์

ประเภทของแบนด์

แบนด์	รายละเอียด
หัวกระดาษ ( HasPageHeader )	จะพิมพ์ที่ส่วนบนสุดของกระดาษ ซึ่งจะทำการพิมพ์ทุกครั้งที่มีการขึ้นหน้าใหม่
หัวเรื่อง ( HasTitle )	เป็นแบนด์แรกที่ถูกพิมพ์ในรายงาน ( แต่อยู่หลังหัวกระดาษ ) และจะทำการพิมพ์เพียงครั้งเดียวเท่านั้น
หัวคอลัมน์ ( HasColumnHeader )	จะพิมพ์ด้านบนของแต่ละคอลัมน์ จะทำการพิมพ์หนึ่งครั้งในแต่ละหน้า หลังจากทำการพิมพ์แบนด์หัวกระดาษ และหัวเรื่องแล้ว
รายละเอียด ( HasDetail )	ข้อมูลแบนด์นี้จะถูกพิมพ์ทุกๆเรคคอร์ดของข้อมูล เป็นแบนด์ที่มีความสำคัญที่สุดเพราะใช้แสดงข้อมูลหลัก
ท้ายกระดาษ ( HasPageFooter )	ท้ายกระดาษ เป็นแบนด์ที่ถูกพิมพ์ที่ท้ายสุดของกระดาษในแต่ละหน้า

เมื่อผู้ใช้ทำการสร้างรายงานด้วย Quick Report จะได้ออปเจกต์ที่มีลักษณะเป็นกระดาษเปล่า ซึ่งเป็นตัวรายงานที่จะใช้ในการแสดงผล โดยผู้ใช้สามารถกำหนดคุณสมบัติเบื้องต้นของรายงานได้ โดยการดับเบิลคลิกที่ตัวรายงาน จะปรากฏไดอะล็อก Report Settings ดังรูป

คุณสมบัติของไดอะล็อก Report Settings

คุณสมบัติ	ความหมาย
Paper Size	กำหนดขนาดกระดาษ
Margins	กำหนดเส้นกั้นขอบ จำนวนคอลัมน์ และระยะห่างระหว่างคอลัมน์
Others	กำหนดรูปแบบของตัวอักษร และหน่วยวัด
Page Frame	กำหนดเส้นขอบรายงาน
Bands	กำหนดแบนด์ประเภทต่างๆ ที่ต้องการให้มีในรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X

Paper size

Width 210.00 Length 297.00 Portrait

Margins

Top 10.00 Left 10.00 Column space 0.00

Bottom 10.00 Right 10.00 Number of columns 1

Other

Font Angsana New Size 16 Units MM

Page frame

Top  Left Color  Change Frame width 1

Bottom  Right

Bands

	Length		Length
<input checked="" type="checkbox"/> Page header	10.58	<input type="checkbox"/> Page footer	
<input checked="" type="checkbox"/> Title	15.61	<input type="checkbox"/> Summary	
<input checked="" type="checkbox"/> Column header	10.58	<input checked="" type="checkbox"/> Print first page header	
<input checked="" type="checkbox"/> Detail band	10.58	<input checked="" type="checkbox"/> Print last page footer	

About QuickReport Preview Apply OK Cancel

รูปที่ 2.11 แสดงหน้าจอไอโอะบล็อกบล็อก Report Settings

### QuickRep

เป็นคอมโพเนนต์หลักที่ใช้ในการสร้างตัวรายงาน ดังนั้นข้อมูลที่จะถูกพิมพ์ในรายงานจะต้องอยู่ภายในออบเจกต์ QuickRep เท่านั้น โดยผู้ใช้สามารถกำหนดคุณสมบัติเบื้องต้นต่างๆของ QuickRep ได้โดยการกำหนดคุณสมบัติในไอโอะบล็อก Report Setting

#### คุณสมบัติของ QuickRep

คุณสมบัติ	ความหมาย
DataSet	กำหนดออบเจกต์ DataSet ซึ่งเข้าถึงข้อมูลที่จะนำมาใช้ในรายงาน
ReportTitle	ชื่อของรายงาน
Bands	กำหนดแบนด์ประเภทต่างๆที่จะมีใขรายงาน
Options	กำหนดลักษณะเพิ่มเติมของรายงาน ซึ่งประกอบด้วย <ul style="list-style-type: none"> <li>- FirstPageHeader ให้มีการพิมพ์แบนด์ประเภท PageHeader ในรายงาน</li> </ul>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าแรกหรือไม่ - LastPageFooter ให้มีการพิมพ์แบบค้ประเภท PageFooter ในรายงาน หน้าแรกหรือไม่
Page	กำหนดลักษณะของหน้ากระดาษรายงาน
PrinterSettings	กำหนดรูปแบบในการพิมพ์รายงาน
Units	หน่วยที่จะใช้กับรายงาน
ShowProgress	ในขณะที่แสดงหรือพิมพ์รายงานให้แสดงแถบการทำงานหรือไม่


#### เมคคอดของ Quick Report

เมคคอด	ความหมาย
Print	สั่งให้พิมพ์รายงานออกทางเครื่องพิมพ์
PrintPreview	สั่งให้แสดงรายงานทางจอภาพ
Prepare	สั่งให้ทำการสร้างรายงาน แต่ยังไม่มีการแสดงหรือพิมพ์รายงาน
PrinterSetup	แสดงไดอะล็อก Printer Setup สำหรับกำหนดคุณสมบัติต่างๆของเครื่องพิมพ์ ในการพิมพ์รายงาน
PrintBackground	สั่งให้พิมพ์รายงานออกทางเครื่องพิมพ์ ซึ่งการสั่งโดยใช้เมคคอดนี้ แอปพลิเคชันจะทำงานซ้อนทันที ในขณะที่เครื่องพิมพ์กำลังพิมพ์รายงาน

#### อีเวนต์ของ QuickRep

อีเวนต์	ความหมาย
BeforePrint	จะถูกเรียกใช้ก่อนที่จะพิมพ์รายงาน หรือปริ้วรายงาน
AfterPrint	จะถูกเรียกเมื่อรายงานได้ถูกพิมพ์เสร็จไปแล้ว ปกติอีเวนต์นี้จะถูกใช้ในการคืน resource ให้กับระบบ เมื่อมีการเรียกใช้เมคคอด PrintBackground
OnPreview	จะถูกเรียกใช้เพื่อเรียกตัวปริ้วที่สร้างขึ้นเอง
AfterPreview	จะถูกเรียกเมื่อตัวปริ้วถูกปิด
OnStartPage	จะถูกเรียกเมื่อมีการขึ้นหน้ากระดาษใหม่ในรายงาน
OnEndPage	จะถูกเรียกทุกครั้งที้หมดหน้ากระดาษ


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 QRSysData

เป็นคอมโพเนนต์ที่ใช้สำหรับการแสดงข้อมูลของระบบรายงาน เช่น วันที่ที่พิมพ์รายงาน เวลาขณะที่พิมพ์รายงาน หรือเลขหน้าของรายงาน

## คุณสมบัติ QRSysData


คุณสมบัติ	ความหมาย
Data	<p>สำหรับกำหนดรูปแบบของข้อมูลที่ต้องการดังนี้</p> <ul style="list-style-type: none"> <li>- qrsDate วันที่ปัจจุบัน</li> <li>- qrsDateTime วันและเวลาปัจจุบัน</li> <li>- qrsDetailCount จำนวนของข้อมูลภายใน DataSet (ไม่สามารถใช้กับ Query ได้)</li> <li>- qrsDetailNo ลำดับของข้อมูลที่ได้จาก DataSet</li> <li>- qrsPageNumber เลขที่หน้าปัจจุบัน</li> <li>- qrsReportTitle ชื่อของรายงาน</li> <li>- qrsTime เวลาปัจจุบัน</li> </ul>
Text	กำหนดข้อความที่จะแสดงหน้าข้อมูล

 QRDBText

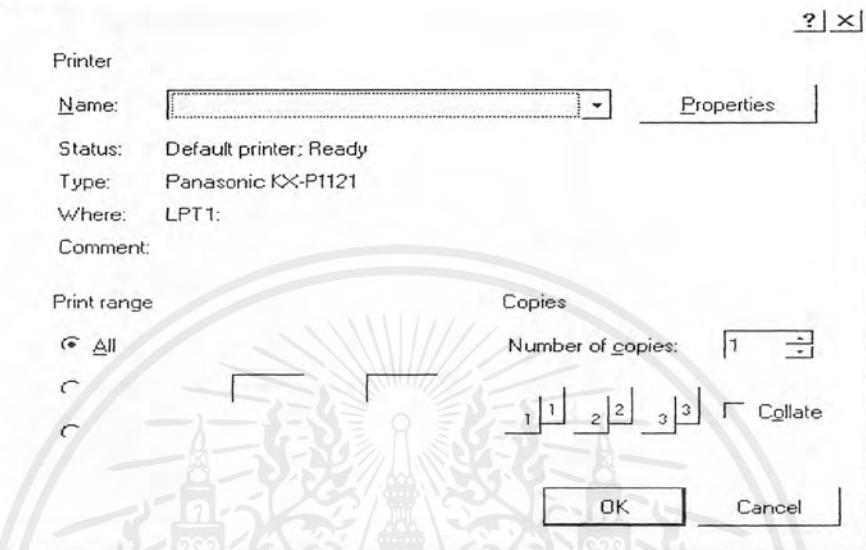
เป็นคอมโพเนนต์ที่ใช้ในการแสดงข้อมูลที่ได้จากฐานข้อมูลในรายงาน โดยออปเจกต์ QRDBText ทุกตัวจะดึงข้อมูลจากออปเจกต์ DataSet โดยผู้ใช้ต้องทำการกำหนดคุณสมบัติ DataSet ไปยังออปเจกต์ DataSet ที่เข้าถึงข้อมูลที่ต้องการ ซึ่งผู้ใช้อาจกำหนดคุณสมบัติ Format เพื่อกำหนดรูปแบบการแสดงผลของ QRDBText แต่ละตัว

## คุณสมบัติของ QRDBText


คุณสมบัติ	ความหมาย
DataSet	กำหนดออปเจกต์ DataSet ที่เข้าถึงข้อมูลที่ต้องการ
DataField	ชื่อฟิลด์ของข้อมูลที่ต้องการแสดง

 PrintDialog

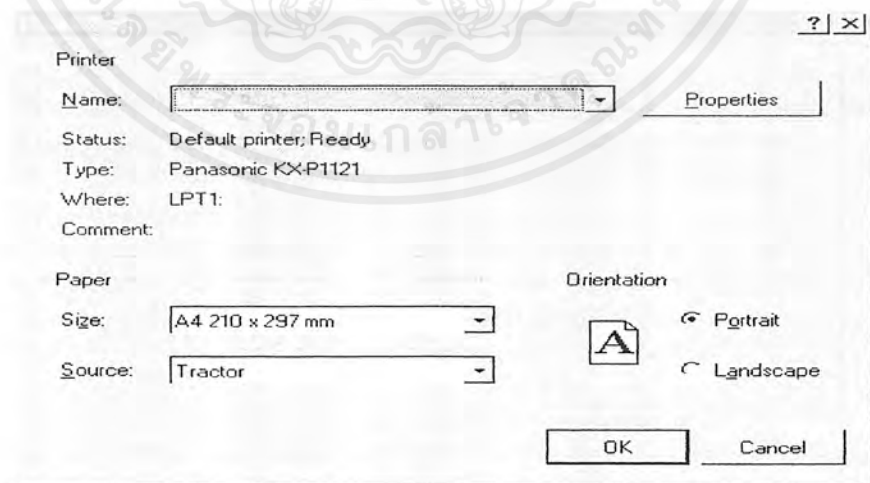
ใช้สำหรับแสดงหน้าจอ สำหรับตั้งเครื่องพิมพ์ให้ทำการพิมพ์ โดยจะมีลักษณะ หน้าจอ ดังรูปที่ 2.12



รูปที่ 2.12 แสดงหน้าจอของ PrintDialog

 PrinterSetupDialog

ใช้สำหรับแสดงหน้าจอเตรียมเครื่องพิมพ์ โดยเราสามารถสร้างกำหนดขนาดลักษณะ และที่เก็บกระดาษได้




รูปที่ 2.13 แสดงหน้าจอของ PrinterSetup Dialog

รายละเอียดของไดอะล็อกบ็อกแบบต่างๆ จะกล่าวถึงอีกครั้งในบทที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Dejan Component

 TComport

เป็นคอมโพเนนต์เพิ่มเติมที่ทางคณะผู้จัดทำ ได้ไปทำการดาวน์โหลดมาทางอินเทอร์เน็ต เพื่ออำนวยความสะดวกในการติดต่อสื่อสารทางพอร์ตอนุกรม มีคุณสมบัติดังนี้

คุณสมบัติ	ความหมาย
BaudRate	กำหนดความเร็วที่ใช้ในการรับส่งข้อมูล
DataBits	กำหนดจำนวนบิตข้อมูลที่ทำการส่ง
Parity	กำหนดบิตที่ใช้ในการตรวจสอบความถูกต้อง
Port	กำหนดพอร์ตที่ใช้ในการติดต่อ ( พอร์ต 1, 2, 2 หรือ 4 )
Stopbits	กำหนดจำนวนบิตจบข้อมูล
Connected	ถ้าเป็น True แสดงว่ามีการติดต่อพอร์ต

## เมธอดของ Tcomport

เมธอด	ความหมาย
WriteString	เป็นเมธอดที่ใช้ในการส่งคำสั่งออกจากพอร์ตอนุกรม
ReadString	เป็นเมธอดที่ใช้ในการรับคำสั่งทางพอร์ตอนุกรม
ShowPropForm	เป็นเมธอดที่ใช้แสดงคุณสมบัติต่าง ซึ่งผู้ใช้สามารถทำการกำหนดได้

## อีเวนต์ของ Tcomport

อีเวนต์	ความหมาย
OnOpen	อีเวนต์นี้จะถูกเรียกใช้เมื่อมีการเปิดพอร์ต
OnClose	อีเวนต์นี้จะถูกเรียกใช้เมื่อมีการปิดพอร์ต
OnRxChar	อีเวนต์นี้จะถูกเรียกใช้เมื่อมีข้อมูลเข้ามาในบัฟเฟอร์ของพอร์ตอนุกรม

## 2.3 ไมโครคอนโทรลเลอร์ตระกูล MCS51

### 2.3.1 คุณสมบัติของ MCS51

คุณสมบัติที่สำคัญ ๆ ของชิปไมโครคอนโทรลเลอร์ตระกูล MCS51 มีดังนี้

- ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ในชิปจำนวน 4 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป (RAM) อยู่ในชิปจำนวน 128 ไบต์ (ใน 8031, 8051) หรือ 256 ไบต์ (ในเบอร์ 8032, 8052)
- สามารถใช้หน่วยความจำสำหรับโปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้อย่างละ 64 กิโลไบต์ แยกจากกัน
- คำสั่งส่วนใหญ่ใช้เวลาทำงานเพียง 1 ไมโครวินาทีเมื่อใช้คริสตอลความถี่ 12 เมกะเฮิร์ตซ์
- มีพอร์ทที่สามารถรับหรือส่งข้อมูลได้ทั้ง 2 ทิศทาง จำนวน 4 พอร์ท ๆ ละ 8 บิต หรือสามารถใช้งานเป็นพอร์ทขนาด 1 บิตแยกจากกัน ทำให้เสมือนมีพอร์ทขนาด 1 บิตใช้งานรวมทั้งสิ้น 32 พอร์ท
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราเร็วในการรับและส่งข้อมูล (Baud rate) ได้ตั้งแต่ 300 ถึง 375 กิโลไบต์ต่อวินาที
- จัดลำดับความสำคัญของสัญญาณอินเทอร์รัปต์ (Interrupt Signal) ได้ 2 ระดับ
- มีรีจิสเตอร์ (Register) สำหรับใช้งานเป็นไทม์เมอร์ (Timer) หรือ เคาน์เตอร์ (Counter) เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิป หรือนับการเปลี่ยนแปลงสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ (Pulse) วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา (ในเบอร์ 8052 จะมี 3 ตัว)
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิตเพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น
- มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
- สามารถประมวลผลแบบบูลีนเพื่อใช้ในงานควบคุมโดยเฉพาะ
- ใช้โปรแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS48 (Upwardly compatible) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 การติดต่อสื่อสารผ่านทางพอร์ตอนุกรม (Serial Interface)

ไมโครคอนโทรลเลอร์ MCS51 มีซีเรียลพอร์ท (Serial Port) ซึ่งสามารถรับและส่งข้อมูลแบบอนุกรมได้ง่าย เพราะมีวงจรรีบ – ส่ง ข้อมูลอนุกรมอยู่ภายในชิพ ทำให้มีความสะดวกในการนำไปประยุกต์ใช้งานที่ต้องมีการติดต่อข้อมูลแบบอนุกรม ซีเรียลพอร์ทที่มีใน MCS51 สามารถทำงานได้ในแบบ ฟูลดูเพลก (Full Duplex) หมายความว่า มันสามารถรับและส่งข้อมูลได้พร้อม ๆ กัน โดยมีการบัฟเฟอร์ (Buffer) ในการรับข้อมูลให้ด้วย กล่าวคือ มันสามารถกำหนดการรับของไบต์ที่สองที่ถูกรับเข้ามา ก่อนที่ไบต์แรกซึ่งได้รับเข้ามาก่อนจะถูกอ่านจาก รีจิสเตอร์ รีจิสเตอร์ (Receive Register) (แต่ถ้าไบต์แรกยังไม่ถูกอ่านเมื่อเวลาที่การรับของไบต์ที่สองสิ้นสุดลงหนึ่งไบต์ในสองไบต์จะหายไป) ซีเรียลพอร์ทจริง ๆ แล้วประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต จำนวนสองตัว ซึ่งมีชื่อเรียกดังนี้ คือ รีจิสเตอร์ รีจิสเตอร์ และทรานสมิต รีจิสเตอร์ (Transmit Register) ซึ่งรีจิสเตอร์ทั้งสองมีตำแหน่งเดียวกันในสเปเชียลฟังก์ชันรีจิสเตอร์ (Special Function Register : SFR) คือ รีจิสเตอร์เอสบีพี (SBUF) โดยการเข้าถึงรีจิสเตอร์แต่ละตัว ซีพียูจะรู้ว่าผู้ใช้ต้องการติดต่อกับรีจิสเตอร์ตัวใด เพราะในการเขียนข้อมูลไปที่รีจิสเตอร์เอสบีพี จะหมายถึง โหลดค่าไปยัง ทรานสมิต รีจิสเตอร์ ส่วนการอ่านข้อมูลในรีจิสเตอร์เอสบีพี หมายถึงการรับข้อมูลจาก รีจิสเตอร์ รีจิสเตอร์

ซีเรียลพอร์ท ใน MCS51 สามารถทำงานในโหมดต่าง ๆ กันได้ถึง 4 แบบด้วยกัน เพื่อความเหมาะสมกับงานแต่ละงาน ดังนี้

- โหมด 0 : ข้อมูลแบบอนุกรมถูกรับเข้าและส่งออกผ่านทางขาอาร์เอ็กซ์ดี (RXD) ส่วนขาทีเอ็กซ์ดี (TXD) จะให้ชิพคล็อก (Shift Clock) ในโหมดนี้จะทำการรับส่งข้อมูลแบบ 8 บิต (8 Data Bits) โดยรับและส่งบิตต่ำก่อน (LSB First) ส่วนอัตราการรับส่งข้อมูล ถูกกำหนดได้แน่นอนที่ 1/12 ของความถี่ออสซิลเลเตอร์ที่ใช้ในระบบ
- โหมด 1 : ข้อมูลจำนวน 10 บิตถูกส่ง (ผ่านทางขาอาร์เอ็กซ์ดี) หรือถูกรับ (ผ่านทางขาอาร์เอ็กซ์ดี) โดยมี 1 บิตเริ่มต้น (Start Bit) (มีค่าเป็น 0 ) 8 บิตข้อมูล (รับและส่งบิตต่ำก่อน) และ 1 บิตจบ (Stop Bit) (มีค่าเป็น 1 ) ส่วนค่าอัตราการรับส่งข้อมูล สามารถเปลี่ยนแปลงได้
- โหมด 2 : ข้อมูลจำนวน 11 บิต ถูกส่ง (ผ่านทางขาทีเอ็กซ์ดี) หรือถูกรับ (ผ่านทางขาอาร์เอ็กซ์ดี) มี 1 บิตเริ่มต้น (มีค่าเป็น 0 ) 8 บิตข้อมูล (รับหรือส่งบิตต่ำก่อน) บิตที่ 9 เป็นบิตที่สามารถโปรแกรมได้ และ 1 บิตจบ (มีค่าเป็น 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าอัตราการรับส่งข้อมูลสามารถตั้งให้เป็น  $1/32$  หรือ  $1/64$  ของความถี่ออสซิลเลเตอร์ที่ใช้

- โหมด 3: ข้อมูลจำนวน 11 บิตถูกส่ง (ผ่านขาที่เอ็กซ์ดี) หรือถูกรับ (ผ่านขาอาร์เอ็กซ์ดี) โดยมี 1 บิตเริ่มต้น (มีค่าเป็น 0) 8 บิตข้อมูล (รับหรือส่งบิตต่าก่อน) บิตที่ 9 เป็นบิตที่สามารถโปรแกรมได้ และ 1 บิตจบ (มีค่าเป็น 1) ซึ่งจะเห็นว่า ในโหมด 3 จะเหมือนกับโหมด 2 ทุกอย่าง เว้นแต่ในโหมด 3 นี้สามารถเปลี่ยนแปลงค่าอัตราการรับส่งข้อมูลได้

สำหรับค่าของอัตราเร็วของการรับส่งข้อมูลจะมีค่าเท่าใดนั้น ก็ขึ้นอยู่กับการทำงานในแต่ละโหมดของซีเรียลพอร์ท ดังนี้

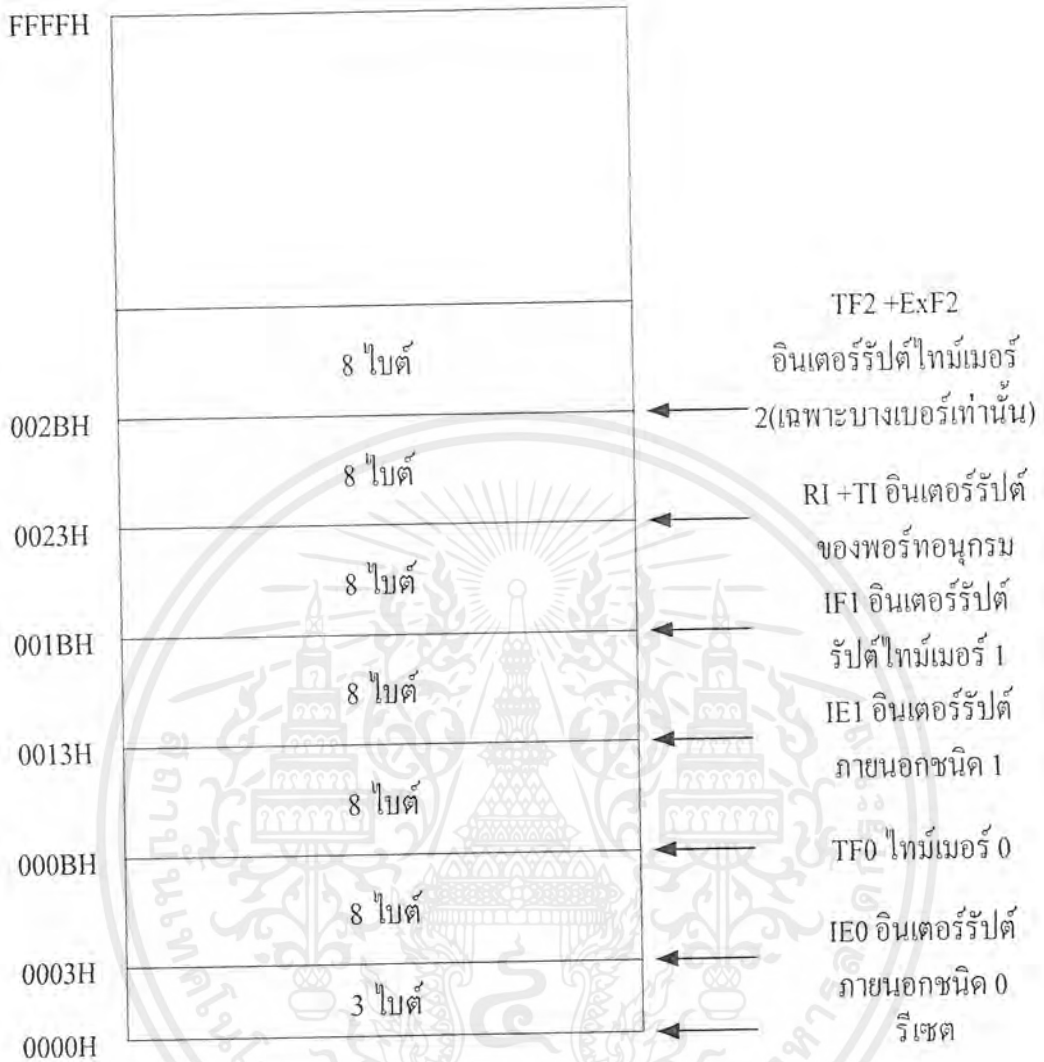
- ในโหมด 0: อัตราเร็วของการรับส่งข้อมูล = ความถี่ออสซิลเลเตอร์ที่ใช้ / 16
- ในโหมด 2: ค่าอัตราเร็วในการรับส่งข้อมูล ขึ้นอยู่กับค่าของบิต เอสมีอด (SMOD) ซึ่งอยู่ในรีจิสเตอร์พีคอน (PCON) โดยถ้าบิตเอสมีอดมีค่าเป็น 0 ค่าอัตราเร็วในการรับส่งข้อมูลจะมีค่าเป็น  $1/64$  ของความถี่ออสซิลเลเตอร์ที่ใช้ แต่ถ้าบิตเอสมีอดมีค่าเป็น 1 ค่าอัตราเร็วในการรับส่งข้อมูลจะมีค่าเป็น  $1/32$  ของความถี่ออสซิลเลเตอร์ที่ใช้
- โหมด 1 และ โหมด 3: ค่าอัตราเร็วในการรับส่งข้อมูลนั้น จะถูกกำหนดโดยอัตราเกิดโอเวอร์โฟลว์ในไทม์เมอร์ 1 (Timer1 Overflow Rate) แต่ถ้าเป็น 8052 ซึ่งมีไทม์เมอร์/เคาน์เตอร์ เพิ่มมาอีก 1 ตัว สามารถเอาไทม์เมอร์ 2 ที่มีเพิ่มมานี้เป็นตัวกำหนดอัตราเร็วของการรับส่งข้อมูลได้ ทำให้มีไทม์เมอร์ จำนวน 2 ตัว ที่สามารถนำมากำหนดอัตราเร็วในการรับส่งข้อมูล โดยอาจใช้ตัวใดตัวหนึ่งในการกำหนดอัตราเร็วสำหรับการรับข้อมูลและอีกตัวสำหรับการส่งข้อมูลได้

อัตราเร็วในการรับ ส่ง ข้อมูล	ความถี่ ออสซิลเลเตอร์	ค่าของบิต SMOD	ไทม์เมอร์ 1		
			C/T	โหมด	Reload Value
Mode 0 Max : 1 MHz	12 MHz	X	X	X	X
Mode 2 Max : 375K	12 MHz	1	X	X	X
Modes 1,3 : 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.059 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางแสดงการใช้ไทม์เมอร์ 1 กับ อัตราเร็วในการรับส่งข้อมูลที่ได้

### 2.3.3 โครงสร้างการอินเตอร์รัปต์

สำหรับ MCS51 สามารถรับสัญญาณอินเตอร์รัปต์ได้ถึง 5 ชนิด โดยจะเป็นสัญญาณอินเตอร์รัปต์ที่เกิดจากภายนอก 2 ชนิด และที่เกิดจากภายในชิปอีก 3 ชนิด เมื่อมีสัญญาณอินเตอร์รัปต์เกิดขึ้น MCS51 จะละการทำงานโปรแกรมที่กำลังทำอยู่และข้ามไปทำงานโปรแกรมบริการอินเตอร์รัปต์ (Interrupt Service Routine) ที่อยู่ในหน่วยความจำตำแหน่งต่าง ๆ ขึ้นอยู่กับชนิดของสัญญาณอินเตอร์รัปต์ ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 แสดงตำแหน่งหน่วยความจำของโปรแกรมบริการอินเทอร์เน็ตแต่ละชนิดใน MCS51

## 2.4 โปรแกรมฟอร์เฮลพ 2.95 (ForeHelp 2.95)

### 2.4.1 เฮลพไฟล์ (Help File)

เฮลพไฟล์ เป็นไฟล์ที่ถูกใช้โดยระบบของวินโดวส์ (Windows) ที่มีชื่อว่า วินเฮลพ (WinHelp) ซึ่งวินเฮลพจะทำการแสดงผลงานที่สร้างขึ้นของไฟล์นามสกุล เอช แอล พี (.HLP) ในหน้าจอของวินเฮลพ ซึ่งในการสร้างไฟล์นามสกุล เอช แอล พี นั้นมีวิธีการสร้างอยู่หลายทางด้วยกัน ตัวอย่างเช่น ใช้ โปรแกรมเวิร์ดโปรเซสเซอร์สร้างไฟล์นามสกุล อาร์ ที เอฟ (.RTF) แล้วใช้วินเฮลพคอมไพเลอร์ (WinHelp Compiler) แปลงไฟล์มาเป็นไฟล์นามสกุล เอช แอล พี ตามต้องการ หรือ อาจใช้โปรแกรมสำเร็จรูปที่เป็นเครื่องมือที่ใช้ในการสร้างไฟล์นามสกุล เอช แอล พี ในการสร้างเฮลพไฟล์ขึ้นมาก็ได้

### 2.4.2 รายละเอียดของโปรแกรมฟอร์เฮลพ

โปรแกรมฟอร์เฮลพ เป็นโปรแกรมสำเร็จรูปที่เป็นเครื่องมือในการสร้างเฮลพไฟล์ขึ้นมาตัวหนึ่งที่สามารถช่วยให้สร้างวินโดวส์เฮลพไฟล์ (Windows HelpFiles) ได้สะดวกและรวดเร็วยิ่งขึ้น

ในแต่ละโปรเจกต์ในการสร้างเฮลพไฟล์นั้น จะประกอบไปด้วยหน่วยย่อย (Unit) หลาย ๆ ส่วน คือ หัวข้อ (Topic) ซึ่งเป็นกลุ่มของข้อมูล อาจเปรียบได้กับหมวดหมู่ในหนังสือ ในแต่ละหัวข้ออาจจะประกอบไปด้วย ตัวหนังสือ (Text), รูปภาพ , และไฟล์ออดิโอ - วิชวล (Audio-visual Files) ซึ่งฟอร์เฮลพ จะแสดงหัวข้อเพียงหัวข้อเดียวในแต่ละครั้งที่แสดงหน้าต่างของฟอร์เฮลพ

ในแต่ละหัวข้อ จะต้องประกอบไปด้วยอย่างน้อยหนึ่งทีหมาย ซึ่งจะถูกสร้างโดยอัตโนมัติที่จุดเริ่มต้นของตัวอักษร และสามารถที่จะสร้างทีหมายเพิ่มเติมที่ใดก็ได้ในหัวข้อ เพื่อความแม่นยำในการไปถึงทีหมายที่ต้องการเมื่อเวลาที่ผู้ใช้ใช้ฮอตสปอตส์ (Hotspots) ในการที่จะไปยังทีหมาย

สำหรับ ฮอตสปอตส์ นั้นมีอยู่หลายชนิดด้วยกัน ตัวอย่างเช่น การกระโดด (Jump) เมื่อผู้ใช้ใช้การกระโดด วินเฮลพจะทำการแสดงหัวข้อใหม่ที่หน้าต่างเฮลพแทนหน้าจอของหัวข้อเก่า ส่วนเมื่อผู้ใช้เรียกใช้ ป๊อปอัพ (Popup) นั้น หน้าต่างเฮลพจะยังคงแสดงหัวข้อเดิมอยู่ และจะมีหัวข้อที่เกี่ยวข้องที่ต้องการของคุณั้นแสดงให้ผู้ใช้ได้เห็นอยู่ในหน้าต่างเล็ก ๆ เหนือหน้าต่างเฮลพ และเมื่อผู้ใช้ทำการกดปุ่มเมาส์อีกครั้งหนึ่ง หน้าต่างเล็ก ๆ นั้นก็จะหายไป

การใช้งานโปรแกรมฟอร์เฮลพนั้น จะมีขั้นตอนคือ ต้องทำการสร้างไฟล์นามสกุล เอช พี เจ และไฟล์นามสกุล อาร์ ที เอฟ ขึ้นมาก่อนแล้วจึงทำการแปลงเป็นไฟล์นามสกุล เอช แอล พี อีกทีหนึ่ง

ในแต่ละโปรเจกของเฮลพไฟล์อาจจะประกอบไปด้วยไฟล์นามสกุลต่าง ๆ ดังต่อไปนี้

- ไฟล์ฐานข้อมูล (Database File) เป็นไฟล์ที่มีนามสกุล ดี บี เอ (.DBA) โปรแกรมฟอร์เฮลพ จะทำการจัดการเฮลพไฟล์เหมือนกับเป็นกลุ่มของไฟล์ฐานข้อมูล และเนื่องด้วยเหตุผลนี้จึงทำให้ไม่สามารถทำการลบหรือเคลื่อนย้ายไฟล์ใด ๆ ในกลุ่มได้

- เทม โพรารี ไฟล์ (Temporary Files) เป็นไฟล์ที่อยู่ในรูปแบบของ FHXXXXXX.DBA (โดยที่ XXXXXX เป็นตัวเลขที่ถูกสุ่มขึ้นมา) จะเป็นไฟล์ที่เป็นกลุ่มของฐานข้อมูลที่ใช้ในเฮลพไฟล์ จะเกิดขึ้นเมื่อมีการแก้ไขฐานข้อมูลที่ได้ทำการคัดลอกมา (Copy of the database)

- แบคอัพ ไฟล์ (Backup Files) เป็นไฟล์ที่มีนามสกุล เอฟ เอช บี (.FHB)

- กราฟฟิกไฟล์ (Graphic Files) เป็นไฟล์ที่มีนามสกุล บี เอ็ม พี (.BMP), เอ็ม อาร์ บี (.MRB), ดับบลิว เอ็ม เอฟ (.WMF), เอส เอช จี (.SHG) คือ ไฟล์พวกรูปภาพกราฟฟิกต่าง ๆ ซึ่งจะสามารถแทรกเข้าไปในเฮลพไฟล์เพื่อให้แสดงในเฮลพไฟล์ได้

- ออดิโอ และวิดีโอ ไฟล์ (Audio and Video Files) เป็นไฟล์ที่มีนามสกุล เอ วิ ไอ (.AVI) สามารถแทรกไฟล์ประเภทนี้เข้าไปในเฮลพไฟล์เพื่อให้แสดงในเฮลพไฟล์ได้

- โปรเจกไฟล์ และริช เทก ฟอร์แมตไฟล์ (Project and Rich Text Format Files) เป็นไฟล์ที่มีนามสกุล เอช พี เจ และนามสกุล อาร์ ที เอฟ ตามลำดับ โปรแกรมฟอร์เฮลพจะให้ใช้ไฟล์นามสกุล เอช พี เจ ได้ทั้งที่สร้างด้วยโปรแกรมฟอร์เฮลพเอง และที่สร้างโดยซอฟต์แวร์อื่น ๆ และในการเปิดไฟล์นามสกุลเอช พี เจ นั้น ก็จะต้องมีไฟล์นามสกุล อาร์ ที เอฟ อยู่ด้วยเสมอ

- เฮลพไฟล์ (Help Files) เป็นไฟล์ที่มีนามสกุล เอช แอล พี เกิดจากการแปลงของวินเฮลพคอมไพเลอร์ ซึ่งในการที่จะแปลงไฟล์มาเป็นไฟล์นามสกุล เอช แอล พี ได้ นั้น จำเป็นที่จะต้องมีไฟล์นามสกุล เอช พี เจ และ นามสกุล อาร์ ที เอฟ เป็นไฟล์อินพุทให้แก่วินเฮลพคอมไพเลอร์ด้วย

- คอนเทนท์ไฟล์ (Contents File) เป็นไฟล์ที่มีนามสกุล ซี เอ็น ที (.CNT) จะเกิดขึ้นเมื่อมีการสร้างคอนเทนท์สำหรับโปรเจกนั้น ๆ

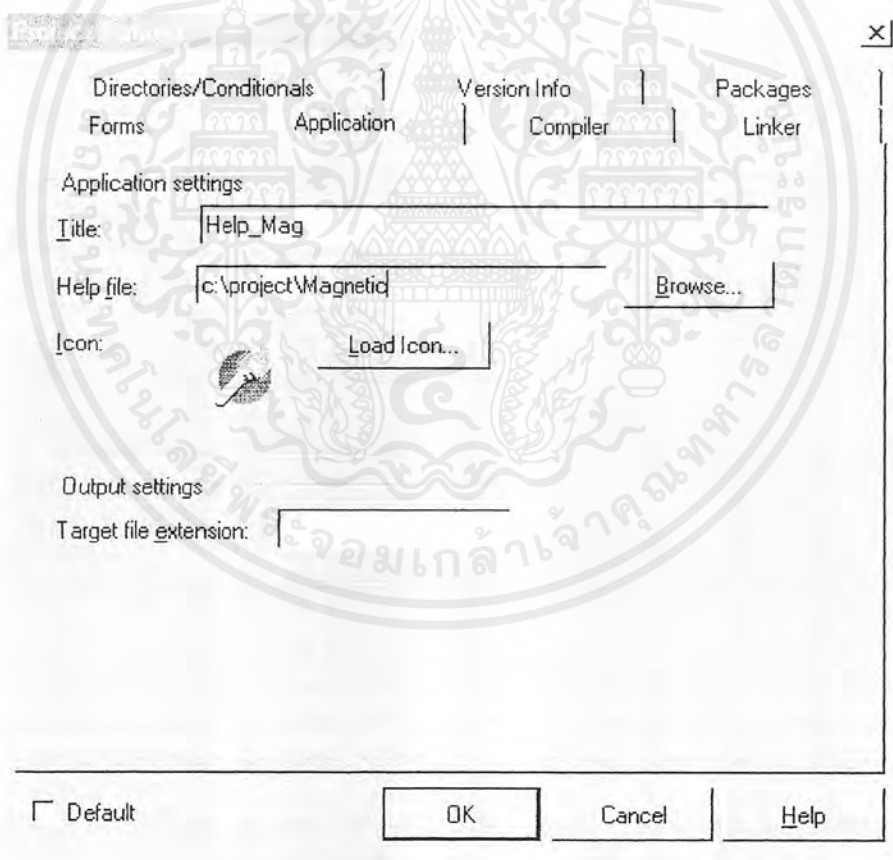
- ฟอร์เฮลพ และวินเฮลพ แอคเซซอรี ไฟล์ (ForeHelp and WinHelp Accessory Files) โปรแกรมฟอร์เฮลพ จะทำการสร้าง ไฟล์นามสกุล ที เอ เอฟ (.TAF) และไฟล์นามสกุล แอล

โอ จี (.LOG) ที่มีชื่อเดียวกันกับชื่อของฐานข้อมูล โปรแกรมฟอร์เฮลทจะใช้ไฟล์เหล่านี้ภายในเท่านั้น และจะทำการลบทิ้งก่อนที่จะออกจากโปรแกรมฟอร์เฮลท

การเชื่อมต่อเฮลทไฟล์ เข้ากับ โปรเจกในเคลไฟล์

สำหรับการสร้างเฮลท ในโปรแกรมที่สร้างโดยใช้โปรแกรมภาษาเคลไฟล์ จะใช้ไฟล์นามสกุล เฮล เอช พี ที่ทำการแปลงมาได้มาเชื่อมต่อกับ โปรแกรมหลักที่ใช้ภาษาเคลไฟล์เขียนได้เลย โดยมีขั้นตอนดังนี้

เมื่อเราทำการสร้างเฮลทไฟล์โดยใช้โปรแกรมฟอร์เฮลท 2.95 จนได้ไฟล์ที่มีนามสกุล เฮล เอช พี มาเรียบร้อยแล้ว ก็ให้นำไฟล์นั้น ไปบันทึกไว้ที่ไดเรกทอรี (Directory) เดียวกันกับโปรแกรมหลัก แล้วทำการเพิ่มไฟล์นามสกุล เฮล เอช พี นั้น เข้าไปในโปรแกรมหลัก โดยเลือกที่เมนูโปรเจก / ออพชั่น จะปรากฏหน้าต่างของ โปรเจกออพชั่นขึ้นมา ให้เลือก แท็บชื่อว่ามีชื่อว่า แอปพลิเคชัน (Application) แล้วทำการเลือกชื่อไฟล์เป็นไฟล์ที่ได้ทำการสร้างไว้



รูปที่ 2.15 แสดงหน้าต่างของ โปรเจกออพชั่น

จากนั้น ให้ใช้ คอมโพเนนต์แอปพลิเคชัน (ซึ่งจะถูกสร้างขึ้นโดยอัตโนมัติเมื่อมีการสร้างฟอร์มใด ๆ ขึ้นมา) ช่วยในการแสดงเฮลปไฟล์

ตัวอย่างเช่น หากเราต้องการให้เมื่อกดปุ่มที่ 1 แล้วแสดงหน้าต่างของเฮลปไฟล์ โดยที่เฮลปไฟล์นั้นมีชื่อว่า 'TRY1.HLP' และมีหมายเลขในการกระโดดไปถึงซึ่งมีค่าเป็นสตริง (JumpID:String) เป็น 'Contents' (คือ จะให้แสดงหน้าต่างที่มีค่าหมายเลขในการกระโดดไปถึง เป็น 'Contents' ของไฟล์ 'TRY1.HLP') เราจะสามารถใช้คำสั่งซึ่งแสดงได้ดังโค้ดต่อไปนี้

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    application.HelpFile:='try1.hlp';
    application.HelpJump('contents');
end;
```

ผลที่เกิดขึ้นคือ เมื่อทำตามขั้นตอนดังที่กล่าวมาแล้ว เมื่อทำการกดปุ่มที่ 1 ก็จะทำให้ปรากฏหน้าต่างของเฮลปไฟล์แสดงให้เห็น ได้ตามต้องการ



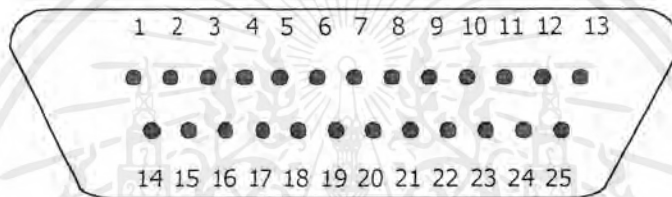
รูปที่ 2.16 แสดงตัวอย่างหน้าต่างของเฮลปไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 มาตรฐานในการสื่อสารข้อมูล

### 2.5.1 มาตรฐานการสื่อสารข้อมูลอนุกรม RS-232

มาตรฐานการสื่อสารข้อมูลอนุกรม RS-232 นั้นเป็นมาตรฐานที่ใช้กันกว้างขวางที่สุด ถูกพัฒนาขึ้นโดย The Electronics Industries Association (EIA) และแบบที่นิยมใช้ทั่วไปคือ RS-232-C มาตรฐานที่เห็นได้เด่นชัดมากที่สุดคือ จำนวนสายที่มี 25 เส้น (RS-232 Standard) ระหว่างอุปกรณ์ที่ใช้สายที่ 2 สำหรับเอาท์พุท (Data Terminal Equipment : DTE) และ อุปกรณ์ที่ใช้สายที่ 2 สำหรับอินพุท (Data Communication Equipment : DCE) (บางที่เรียกว่า DB-25 Cable) ซึ่งสายแต่ละเส้นจะมีหน้าที่ในการสื่อสารแตกต่างกันไป ดังรูปที่ 2.17



รูปที่ 2.17 แสดง RS-232 Connector

ตัวอย่างเช่น DTE คือ เครื่องไมโครคอมพิวเตอร์ (PC) และ DCE คือ โมเด็ม ซึ่งอุปกรณ์ทั้งสองจะไม่ส่งข้อมูลถ้าอุปกรณ์อีกเครื่องไม่ต้องการรับข้อมูลนั้น เมื่อ DCE เชื่อมต่อเข้ากับโครงข่าย (Network) ซึ่งมี DTE เชื่อมต่ออยู่ด้วย DCE ต้องรู้ว่าเมื่อไรที่ DTE พร้อมในการส่งข้อมูล (ดูรูปที่ 2.17 ประกอบ) เมื่อ DTE ส่งสัญญาณ DTR (Data Terminal Ready ที่เวลา  $t_1$ ) ต่อมา DCE จะตอบสนองต่อสัญญาณโดยเชื่อมต่อตัวเองเข้ากับโครงข่าย ต่อมา DCE จะส่งสัญญาณ DSR (Data Set Ready) เพื่อตอบกลับไปว่าพร้อมสำหรับรับข้อมูล (ที่เวลา  $t_2$ ) ตอนนี้อุปกรณ์ทั้งสองก็พร้อมในการติดต่อสื่อสารกันแล้ว

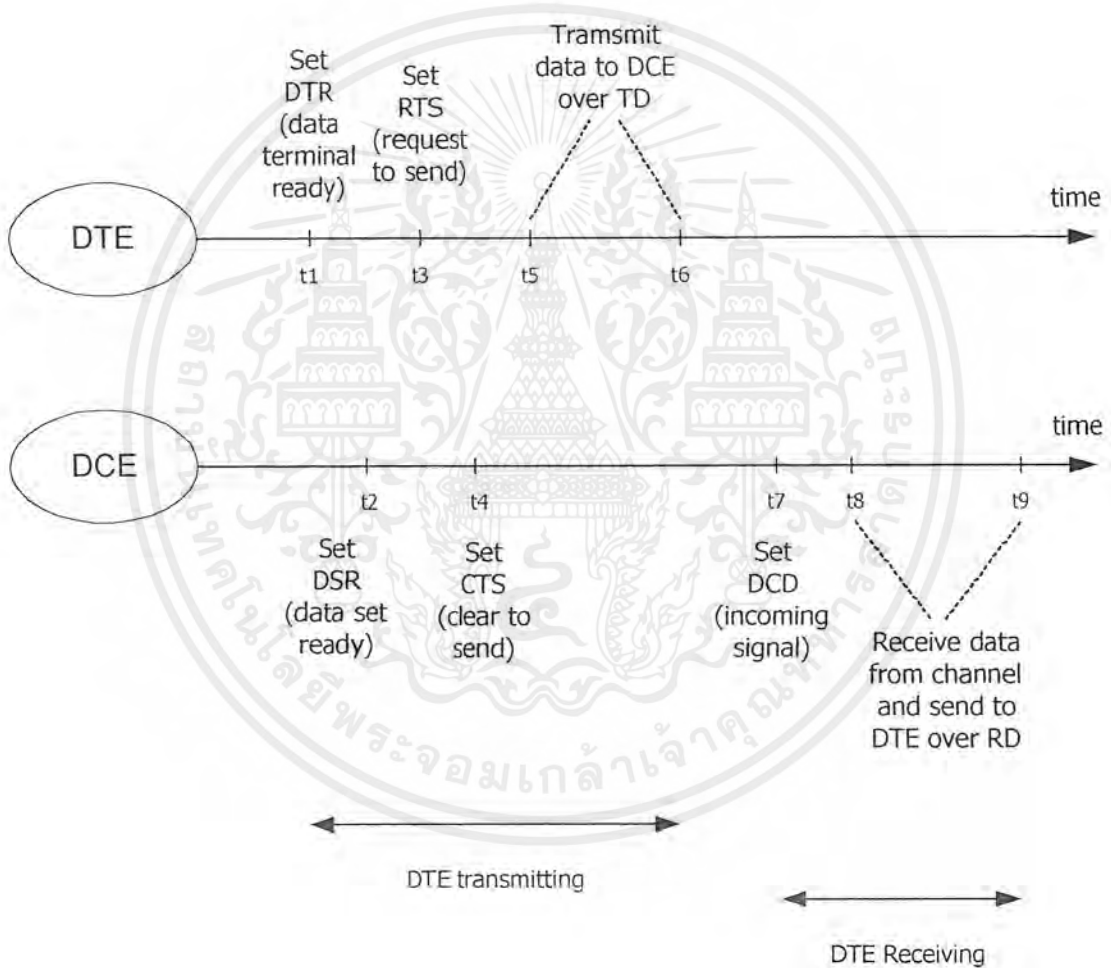
หาก DTE ขออนุญาตในการส่งข้อมูลให้แก่ DCE จะมีการส่งสัญญาณ RTS (Request To Send) ช่วงนี้ระบบจะมีทิศทางการสื่อสารเป็นแบบฮาร์ฟดูเพลกซ์ (Half-Duplex) ในการตอบรับสัญญาณ RTS นั้น DCE จะเข้าสู่โหมดทรานสมิต (Transmit) หมายความว่ามันพร้อมที่จะส่งข้อมูลออกไปสู่โครงข่ายแล้ว และส่งสัญญาณ CTS (Clear To Send) และสุดท้ายแล้ว DTE ก็ส่งข้อมูลออกไปตามสายสัญญาณ TD (Transmit Data)

เมื่อ DCE ตรวจพบสัญญาณเข้ามาจากโครงข่าย มันจะส่งสัญญาณ DCD (Incoming Signal) DCE จะส่งสัญญาณเหล่านี้ไปยัง DTE โดยใช้สายสัญญาณ RD (Receive Data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสัญญาณทางไฟฟ้าของมาตรฐาน RS-232 นั้น ได้มีการกำหนดคุณลักษณะของ สัญญาณทางไฟฟ้าที่ถูกใช้ในการเชื่อมต่ออนุกรมโดยตรง มีเพียงสองลักษณะคือ สเปส (Space) แสดงถึงไบนารี 0 หรือแรงดันไฟฟ้าบวก และ มาร์ค (Mark) แสดงถึงไบนารี 1 หรือแรงดันไฟฟ้าลบ

บนสายข้อมูล (เช่นสาย 2 และ 3) แรงดันไฟฟ้าบวกแสดงถึงค่าลอจิก 0 และแรงดันไฟฟ้าลบแสดงถึงค่าลอจิก 1 บนสายแฮนด์เช็คกิ้ง (เช่น DTR และ DSR) แรงดันไฟฟ้าบวกแสดงว่าส่งข้อมูลได้ ส่วนแรงดันไฟฟ้านบหมายถึงหยุดส่งข้อมูล



รูปที่ 2.18

แสดงการส่งและการรับสัญญาณระหว่าง DTE และ DCE ในมาตรฐาน RS-232

แรงดันไฟฟ้าบวก (สถานะสเปส) อยู่ระหว่าง +5 ถึง +15 โวลต์ สำหรับเอาต์พุต และระหว่าง +3 และ +15 โวลต์สำหรับอินพุต ความแตกต่างมีไว้เพื่อกรณีที่แรงดันไฟฟ้าสูญหายเนื่องจากความยาวของสายสัญญาณ ในทำนองเดียวกัน แรงดันไฟฟ้าลบ (สถานะมาร์ค) ถูกกำหนดไว้ระหว่าง -5 ถึง -15 โวลต์สำหรับเอาต์พุต และ -3 ถึง -15 โวลต์สำหรับอินพุต

โดยทั่วไป ตัวเชื่อมต่อ (Connector) กับพอร์ท RS-232 ที่ใช้กันจะมีสายต่อไม่ถึง 25 เส้น เราเรียกว่าเป็นสับเซตของ RS-232 (RS-232 Subsets) ส่วนมากจะมีสาย 8 หรือ 9 เส้น และมีผู้นิยมใช้กันอย่างแพร่หลาย เพราะผู้ใช้ไม่ต้องการใช้ความสามารถของมาตรฐาน RS-232 อย่างเต็มที่ ซึ่งพวกเขาต้องการที่จะสื่อสารเพียงเพื่อติดต่อและรับส่งข้อมูลเท่านั้น

มาตรฐาน RS-232 สามารถส่งข้อมูลได้ 20,000 บิตต่อวินาที (bps) โดยระยะทางหรือความยาวของสายประมาณ 50 ฟุต



### บทที่ 3

#### การออกแบบ

#### 3.1 ระบบฐานข้อมูลในเดสก์ท็อป

ฐานข้อมูลมาตรฐานที่เดสก์ท็อปใช้เป็นตัวเก็บข้อมูล คือ Paradox และ dBASE แต่เดสก์ท็อปสามารถทำงานกับฐานข้อมูลภายนอกอื่นๆ ไม่ว่าจะเป็นไฟล์ฐานข้อมูลทั่วไป เช่น Microsoft Access , Microsoft FoxPro ซึ่งต่อไปรายชื่อในรายงานฉบับนี้จะเรียกฐานข้อมูลนี้ว่า ฐานข้อมูลภายใน ( ฐานข้อมูลภายใน คือ ฐานข้อมูลที่อยู่ในรูปแบบของไฟล์สำหรับเก็บข้อมูล ปกติจะอยู่ภายในเครื่องเดียวกันกับแอปพลิเคชันที่ใช้งาน เช่น Paradox , dBASE , MS Access เป็นต้น )

โปรแกรมเดสก์ท็อปมีการเชื่อมต่อฐานข้อมูลผ่านระบบที่ชื่อว่า Borland Database Engine ( BDE ) ซึ่งเป็นตัวจัดการระบบทางด้านฐานข้อมูลทั้งหมด

ศัพท์ทางด้านฐานข้อมูลที่ควรรู้

คำศัพท์	ความหมาย
ฐานข้อมูล ( Database )	คือ กลุ่มของข้อมูลที่มีความสัมพันธ์กัน ซึ่งผู้ใช้สามารถบันทึก , เปลี่ยนแปลง และสืบค้นกลับมาใช้งานได้อย่างรวดเร็วและเป็นระบบ
ตาราง ( Table )	คือ ที่สำหรับเก็บข้อมูล ซึ่งในฐานข้อมูลอาจมีตารางได้หลายตาราง โดยแต่ละตารางจะประกอบด้วยคอลัมน์ที่เรียกว่า ฟิวด์ และแถวเรียกว่า เรคคอร์ด
อินเด็กซ์ ( Index )	คือ คำนีของตาราง ซึ่งสามารถมองอินเด็กซ์เหมือนกับคีย์ที่อยู่ตอนท้ายของหนังสือ ซึ่งเป็นตัวช่วยค้นหาข้อมูลในตาราง
ไพรมารีคีย์ ( Primary key )	คือ ฟิวด์ หรือกลุ่มของฟิวด์ในตาราง ที่ถูกใช้ในการอ้างอิงถึงข้อมูลแต่ละเรคคอร์ด ซึ่งทุกๆเรคคอร์ดจะมีไพรมารีคีย์ที่ไม่ซ้ำกัน
SQL( Structure Query Language )	คือ ภาษาที่ใช้ในการจัดการกับข้อมูลในฐานข้อมูล
คิวรี ( Query )	คือ การสอบถามข้อมูลจากฐานข้อมูล ปกติเราจะเขียนคิวรี โดยใช้คำสั่งภาษา SQL
BDE ( Borland Database Engine )	เป็นระบบที่ใช้ในการเชื่อมต่อกับฐานข้อมูล ซึ่งงานด้านฐานข้อมูลต่างๆของเดสก์ท็อปทำผ่าน BDE ทั้งหมด
ไดรเวอร์ ( Driver )	เป็นตัวกลางที่ช่วยในการเชื่อมต่อจาก BDE ไปยังฐานข้อมูล ซึ่งสามารถเชื่อมต่อไปยังฐานข้อมูลทุกประเภทที่มีไดรเวอร์ในการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำศัพท์	ความหมาย
แอสเลียส ( Alias )	เป็นตัวเก็บคุณสมบัติต่างๆ ในการเชื่อมต่อไปยังฐานข้อมูล เช่น ประเภทฐานข้อมูล , ตำแหน่งที่เก็บฐานข้อมูล , ชื่อฐานข้อมูล เป็นต้น

### 3.1.1 Database Desktop



รูปที่ 3.1 แสดงหน้าจอของโปรแกรม Database Desktop

เป็นโปรแกรมสำหรับจัดการเกี่ยวกับงานทางด้านฐานข้อมูลต่างๆ ไม่ว่าจะเป็นการสร้าง , การแสดง และการแก้ไขตาราง และอินเด็กซ์ของฐานข้อมูล รวมไปถึงการแก้ไขข้อมูลในตาราง และการสร้างไฟล์คำสั่ง SQL ซึ่งโดยมากจะใช้ Database Desktop ในการสร้างตาราง และ อินเด็กซ์ ตลอดจนการกรอกข้อมูลเริ่มต้น

### 3.1.2 การสร้างแอปพลิเคชันด้านฐานข้อมูล

ขั้นตอนการสร้างแอปพลิเคชันด้านฐานข้อมูลมีดังนี้

1. สร้างแอเลียสที่เชื่อมต่อไปยังฐานข้อมูลที่ต้องการ
2. สร้างตารางฐานข้อมูลและอินเด็กซ์
3. สร้างแอปพลิเคชันใหม่ในเคลไพ
4. กำหนดการเข้าถึงข้อมูลโดยใช้คอมโพเนนต์ประเภท Data Access
5. ออกแบบและสร้างหน้าจอส่วนติดต่อกับผู้ใช้ โดยใช้คอมโพเนนต์ประเภท Data Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดลไฟได้แบ่งคอมโพเนนต์ทางด้านฐานข้อมูลออกเป็น 2 ประเภทใหญ่ๆ คือ

1. Data Access เป็นคอมโพเนนต์ ที่ใช้ในการเข้าถึงฐานข้อมูลในลักษณะต่าง ๆ ซึ่งลักษณะการเข้าถึงข้อมูลที่มีจะถูกใช้เป็นประจำคือ การใช้ตาราง และคอมโพเนนต์คิวรี
2. Data Control เป็นคอมโพเนนต์ที่ใช้ในการควบคุมการติดต่อกับผู้ใช้ในรูปแบบต่างๆ เช่น ช่องสำหรับกรอกข้อมูล เป็นต้น

### 3.1.3 ระบบฐานข้อมูล

ผู้จัดทำได้เลือกใช้ฐานข้อมูลภายในของเดลไฟ เนื่องจากสะดวกในการใช้ และง่ายต่อการพัฒนาโปรแกรมในอนาคต โดยฐานข้อมูลที่สร้างขึ้นมีดังนี้

#### 3.1.3.1 ฐานข้อมูลนักศึกษา

Field roster:

	Field Name	Type	Size	Key
1		A	9	*
2	Year	A	8	*
3	Sex	A	6	
4	Name	A	40	
5	Status	A	10	
6	Reason	A	50	
7	Code	A	3	
8	Mag	A	40	

Table properties:

Validity Checks

1. Required Field

2. Minimum value:

3. Maximum value:

4. Default value:

5. Picture:

Enter a field name up to 25 characters long.

Pack Table      Assist...

Save      Save As...      Cancel      Help

รูปที่ 3.2 แสดงหน้าจอการสร้างฐานข้อมูลนักศึกษาใน Database Desktop

ในการออกแบบฐานข้อมูลผู้ใช้ ทางคณะผู้จัดทำได้กำหนดรูปแบบของฟิลด์ที่ใช้ในการเก็บข้อมูลดังนี้

ฟิลด์	ชนิดข้อมูล	ขนาด	จุดประสงค์
Id	String	9	ใช้เก็บรหัสประจำตัวนักศึกษา
Sex	String	6	ใช้เก็บค่านำหน้าชื่อ
Name	String	40	ใช้เก็บชื่อ - สกุลนักศึกษา
Year	String	8	ใช้เก็บชั้นปี ของนักศึกษา
Status	String	10	ใช้เก็บสถานะภาพการใช้งาน
Reason	String	50	ใช้เก็บเหตุผลในกรณีที่ไม่มีสิทธิ์ใช้งาน
Code	String	3	ใช้เก็บรหัส 3 ไบต์ที่ระบบทำการสร้างขึ้น
Mag	String	40	ใช้เก็บข้อมูลจากบัตรแม่เหล็กขนาด 40 ไบต์

หมายเหตุ

Id คือ รหัสนักศึกษาตามบัตรประจำตัวนักศึกษา

Code คือ รหัสประจำตัวนักศึกษาขนาด 3 ไบต์ ที่คอมพิวเตอร์จะสร้างขึ้นเมื่อทำการเพิ่มผู้ใช้ เพื่อใช้ในการติดต่อกับบอร์ดหลักแทนการส่งข้อมูลจากบัตรแม่เหล็กขนาด 40 ไบต์

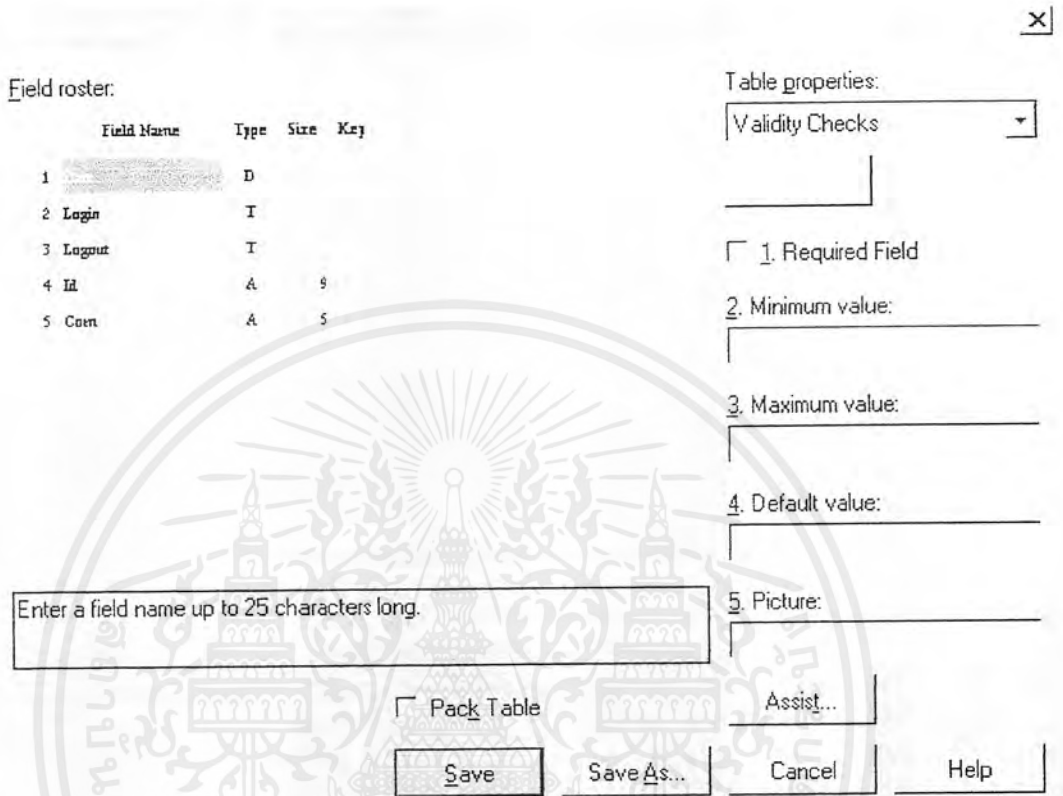
Mag คือ ข้อมูลที่อ่านได้จากบัตรแม่เหล็กขนาด 40 ไบต์

โดยในที่นี้ได้กำหนดให้ฟิลด์ Id และ ฟิลด์ Year เป็นคีย์ที่ใช้ในการจัดเรียงข้อมูลในฐานข้อมูล ( การกำหนดคีย์ทำได้โดยใส่เครื่องหมาย \* ในช่องคีย์ ดังรูปที่ 3.1 ) นอกจากนั้นยังได้มีการสร้างอินเด็กซ์ เพื่ออำนวยความสะดวกดังนี้

อินเด็กซ์	
Idxid	จัดเรียงข้อมูลตาม Id
Idxname	จัดเรียงข้อมูลตาม Name
Idxyear	จัดเรียงข้อมูลตาม year และ code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.1.3.2 ฐานข้อมูลผู้ใช้



รูปที่ 3.3 แสดงหน้าจอการสร้างฐานข้อมูลผู้ใช้ใน Database Desktop

ในการออกแบบฐานข้อมูลผู้ใช้ ทางคณะผู้จัดทำได้กำหนดรูปแบบของฟิลด์ที่ใช้ในการเก็บข้อมูลดังนี้

ฟิลด์	ชนิดข้อมูล	ขนาด	จุดประสงค์
Date	Date	-	ใช้เก็บวันที่ทำการใช้งาน
Login	Time	-	ใช้เก็บเวลาที่เริ่มทำการใช้งาน
Logout	Time	-	ใช้เก็บเวลาสิ้นสุดการใช้งาน
Id	String	9	ใช้เก็บรหัสนักศึกษาที่ทำการใช้งาน
Com	string	5	ใช้เก็บหมายเลขคอมพิวเตอร์ที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเด็กซ์ที่ใช้งานมีดังนี้

อินเด็กซ์	
Idxdate	จัดเรียงข้อมูลตาม date
Idxuse	จัดเรียงข้อมูลตาม date , login และ com

### 3.1.3.3 ฐานข้อมูลรหัสที่ระบบทำการสร้างขึ้น

Field roster:

Field Name	Type	Size	Key
1	A	3	*

Table properties:

Validity Checks

- 1. Required Field
- 2. Minimum value:
- 3. Maximum value:
- 4. Default value:
- 5. Picture:

Pack Table

Assist...

Save Save As... Cancel Help

Enter a field name up to 25 characters long.

รูปที่ 3.4 แสดงหน้าจอการสร้างฐานข้อมูลรหัสที่ระบบทำการสร้างขึ้นใน Database Desktop

ในฐานข้อมูลนี้ทำการสร้างขึ้นเพื่อเก็บรหัส 3 ไบต์ของผู้ใช้ที่ถูกลบออกจากฐานข้อมูล เพื่อนำรหัสเหล่านี้กลับมาใช้อีกครั้ง เมื่อมีการเพิ่มผู้ใช้ นั่นคือ จะมีการเพิ่มข้อมูลเข้าฐานข้อมูล เมื่อมีการลบผู้ใช้ออกจากฐานข้อมูลนักศึกษา และจะมีการลบข้อมูลออกจากฐานข้อมูลนี้เมื่อมีการเพิ่มผู้ใช้เข้ามาในฐานข้อมูลนักศึกษา โดยในฐานข้อมูลนี้จะมีฟิลด์เพียงฟิลด์เดียวคือ รหัส 3 ไบต์ และทำการจัดเรียงข้อมูลในฐานข้อมูลตามรหัสนี้จากน้อย ไป มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งานบางครั้งจำเป็นต้องมีการเชื่อมโยงข้อมูลระหว่างฐานข้อมูล โดยในการเชื่อมโยงนี้สามารถทำได้โดยการกำหนดคีย์ที่จะใช้ในการเชื่อมโยงให้ตรงกัน ในที่นี้ทางคณะผู้จัดทำกำหนดให้ใช้ฟิลด์ Id ในการเชื่อมโยง ดังรูป

รูปที่ 3.5 แสดงหน้าจอการเชื่อมโยงข้อมูลระหว่างฐานข้อมูล

จากรูปเป็นการหาฟิลด์ Name ในฐานข้อมูลนักศึกษา เพื่อให้คอมพิวเตอร์ที่อ้างอิงกับฐานข้อมูลผู้ใช้สามารถอ้างอิงข้อมูลจากฟิลด์ Name ได้ (ฐานข้อมูลผู้ใช้ไม่มีฟิลด์ Name) โดยในการเชื่อมโยงจะใช้ฟิลด์ Id ในการเชื่อมโยง

#### 3.1.3.4 ระบบรักษาความปลอดภัยของระบบฐานข้อมูล

เนื่องจากการกระทำการใด ๆ ตามฟังก์ชันการแก้ไขและเปลี่ยนแปลงข้อมูล จะเป็นการกระทำกรกับข้อมูลในฐานข้อมูลโดยตรง ดังนั้นจึงจำเป็นที่จะต้องมึระบบรักษาความปลอดภัยสำหรับฐานข้อมูล โดยในที่นี้คณะผู้จัดทำได้ทำการกำหนดรหัส (PASSWORD) ขึ้น เมื่อเริ่มต้นของระบบ ผู้ที่มีหน้าที่ดูแลระบบจะได้รับรหัสเริ่มต้น (Initial Password) ซึ่งมีได้สูงสุด 3 คน เนื่องจากระบบมีรหัสเพียง 3 รหัสเท่านั้น

ดังนั้นในการเข้าใช้ฟังก์ชันแก้ไข – เปลี่ยนแปลงข้อมูล ผู้ดูแลต้องกรอกรหัสก่อน ถ้ารหัสที่ทำการกรอกไม่ถูกต้อง ระบบจะไม่เข้าสู่ฟังก์ชันแก้ไข – เปลี่ยนแปลงข้อมูล นอกจากนี้ผู้ดูแลยังสามารถเปลี่ยนรหัสได้ตามต้องการ

### 3.1.4 ระบบฐานข้อมูลบนบอร์ดหลัก

ข้อมูลบนบอร์ดหลัก จะมีการเก็บข้อมูลในรูปแบบของรหัสแอสกี (ASCII Code) แทนแต่ละตัวอักษร ตัวอย่างเช่น หากต้องการเก็บข้อมูลรหัสของผู้ใช้ 3 หลัก ที่มีค่า '032' ข้อมูลที่ถูกเก็บจะถูกเก็บอยู่ในหน่วยความจำภายนอกของบอร์ดหลัก เป็นข้อมูลที่มีขนาด 3 ไบต์ เรียงกัน มีค่าเป็น 30H 33H 32H ตามลำดับ

สำหรับระบบฐานข้อมูลบนบอร์ดหลัก จะมีอยู่ 2 ฐานข้อมูล ได้แก่ ฐานข้อมูลเก็บรายละเอียดของผู้ใช้ และฐานข้อมูลเก็บข้อมูลการใช้งานของผู้ใช้ ดังมีรายละเอียดของแต่ละอัน ดังต่อไปนี้

#### 3.1.4.1 ฐานข้อมูลเก็บรายละเอียดของผู้ใช้

ในการเก็บข้อมูลนั้น จะทำการแยกเก็บข้อมูลของแต่ละชั้นปีออกจากกันเพื่อความสะดวกและรวดเร็วในการค้นหาข้อมูล โดยแต่ละชั้นปีจะมีสถานที่เก็บข้อมูลอยู่ในตำแหน่งที่ต่าง ๆ กันไป โดยทั้งหมดจะมีจำนวนชั้นปีอยู่ 5 ชั้นปี ได้แก่ ชั้นปีที่ 2 ,ชั้นปีที่ 3 ,ชั้นปีที่ 4,ปริญญาโท และอาจารย์ ซึ่งรายละเอียดของการเก็บข้อมูลของแต่ละชั้นปีมีดังต่อไปนี้

ชื่อฟิลด์ (Field Name)	ขนาด (ไบต์)
Mag	40
Code	3
Status	1

รายละเอียดของแต่ละฟิลด์ มีดังต่อไปนี้

1. ฟิลด์ Mag เป็นฟิลด์ที่ใช้ในการเก็บค่ารหัสของบัตรแม่เหล็ก 40 ไบต์ ที่อ่านมาได้จากเครื่องรูดบัตรแม่เหล็ก โดยจะเก็บอยู่ในลักษณะของรหัสแอสกี 40 ตัว
2. ฟิลด์ Code เป็นฟิลด์ที่ใช้ในการเก็บค่ารหัสของผู้ใช้ มีขนาด 3 ไบต์ ซึ่งเป็นค่ารหัสที่ทางโปรแกรมบนเครื่องคอมพิวเตอร์หลักทำการสร้างขึ้นมาให้แก่ผู้ใช้แต่ละคน เพื่อความสะดวกในการค้นหาและเก็บข้อมูล โดยจะเก็บอยู่ในลักษณะของรหัสแอสกี 3 ตัว
3. ฟิลด์ Status เป็นฟิลด์ที่ใช้ในการแสดงสถานะการใช้งานของแต่ละชั้นปี และสามารถบอกได้ถึงชั้นปีของผู้ใช้อีกด้วย เพราะข้อมูลที่ทำการจัดเก็บจะอยู่ในรูปของข้อมูลที่เป็นรหัสแอสกี 1 ไบต์ ที่ถ้าหากมีสถานะเป็นใช้ไม่ได้ ค่าในฟิลด์นี้จะมีค่าเป็น 0 แต่ถ้าหากมีสถานะเป็น

ใช้ได้ ค่าในฟิลด์นี้จะมีค่าเป็น 2 ,3 ,4 ,5 ,6 ตามแต่ว่าผู้ใช้นั้นอยู่ในชั้นปีที่ 2 ,3 ,4 ,ปริญญาโท หรือเป็นอาจารย์

### 3.1.4.2 ฐานข้อมูลเก็บข้อมูลการใช้งานของผู้ใช้

เมื่อผู้ใช้เข้ามาใช้งานระบบโดยการรูดบัตรแม่เหล็ก ข้อมูลของผู้ใช้พร้อมทั้งวัน เวลา และหมายเลขคอมพิวเตอร์ที่ใช้ จะถูกเก็บบันทึกลงฐานข้อมูลเพื่อส่งต่อให้เครื่องคอมพิวเตอร์หลักใช้ในการขอและตรวจสอบรายละเอียดการใช้งานได้ โดยจะมีรายละเอียดของรูปแบบการเก็บข้อมูลดังต่อไปนี้

ชื่อฟิลด์ (Field Name)	ขนาด (ไบต์)
Code	3
Date	6
Login	6
Logout	6
Com	2

สำหรับรายละเอียดของแต่ละฟิลด์ มีดังต่อไปนี้

1. ฟิลด์ Code เป็นฟิลด์ที่ใช้เก็บค่ารหัสของผู้ใช้ ที่ถูกสร้างขึ้นมาโดยโปรแกรมในเครื่องคอมพิวเตอร์หลักและถูกส่งมาเก็บอยู่ในฐานข้อมูลเก็บรายละเอียดของผู้ใช้งาน ซึ่งเป็นค่ารหัสแอสกีขนาด 3 ไบต์

2. ฟิลด์ Date เป็นฟิลด์ที่ใช้ในการเก็บค่าวันที่ที่ผู้ใช้ได้ทำการเข้ามาใช้งานระบบ โดยจะเก็บอยู่ในรูปแบบของ รหัสแอสกีขนาด 6 ไบต์ แสดงถึง วันที่ 2 ไบต์ ,เดือน 2 ไบต์ และ ปี 2 ไบต์

3. ฟิลด์ Login เป็นฟิลด์ที่ใช้ในการเก็บค่าเวลาที่ผู้ใช้ได้ทำการเข้ามาใช้งานระบบ โดยจะเก็บอยู่ในรูปแบบของ รหัสแอสกีขนาด 6 ไบต์ แสดงถึง ชั่วโมงที่เข้ามาใช้งาน 2 ไบต์ ,นาฬิกาที่เข้ามาใช้งาน 2 ไบต์ และวินาทีที่เข้ามาใช้งาน 2 ไบต์

4. ฟิลด์ Logout เป็นฟิลด์ที่ใช้ในการเก็บค่าเวลาที่ผู้ใช้เลิกใช้งานและทำการรูดบัตรเพื่อออกจากระบบ โดยจะเก็บอยู่ในรูปของ รหัสแอสกีขนาด 6 ไบต์ แสดงถึง ชั่วโมงที่ออกจากระบบ 2 ไบต์ ,นาฬิกาที่ออกจากระบบ 2 ไบต์ และวินาทีที่ออกจากระบบ 2 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

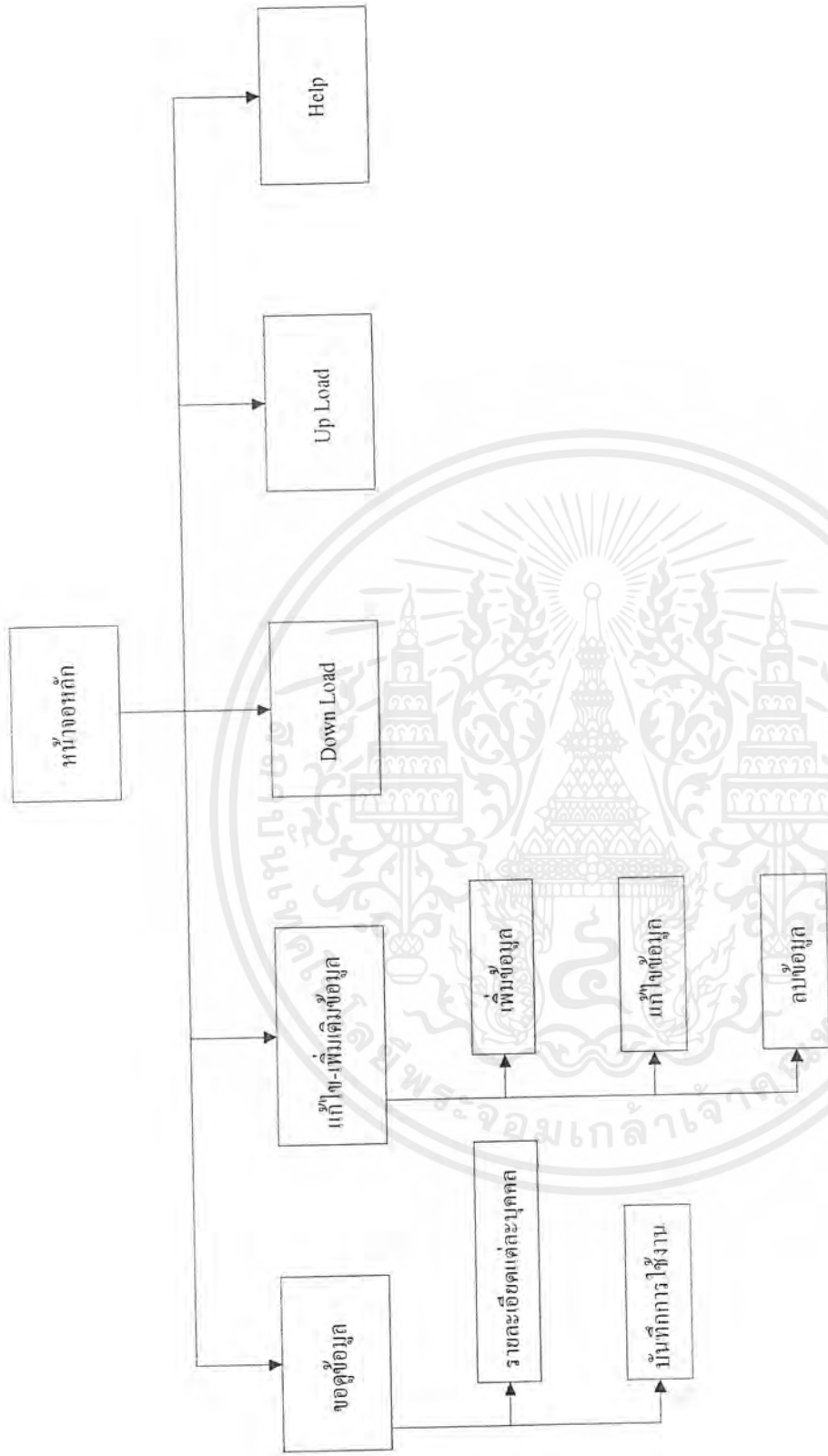
5. ไฟล์ Com เป็นไฟล์ที่ใช้ในการเก็บค่าหมายเลขคอมพิวเตอร์ที่ผู้ใช้ได้ทำการ  
 ระบุบัตรเพื่อเข้ามาใช้งานระบบ โดยจะเก็บเป็นรหัสแอสกี แสดงถึงหมายเลขของ  
 คอมพิวเตอร์ ขนาด 2 ไบต์

### 3.2 ฟังก์ชันการทำงานของโปรแกรม

ฟังก์ชันการทำงานสามารถแสดงในรูปแบบของแผนผังได้ดังรูปที่ 3.6 และ 3.7 โดยมีรายละเอียดของแต่ละฟังก์ชัน ดังนี้

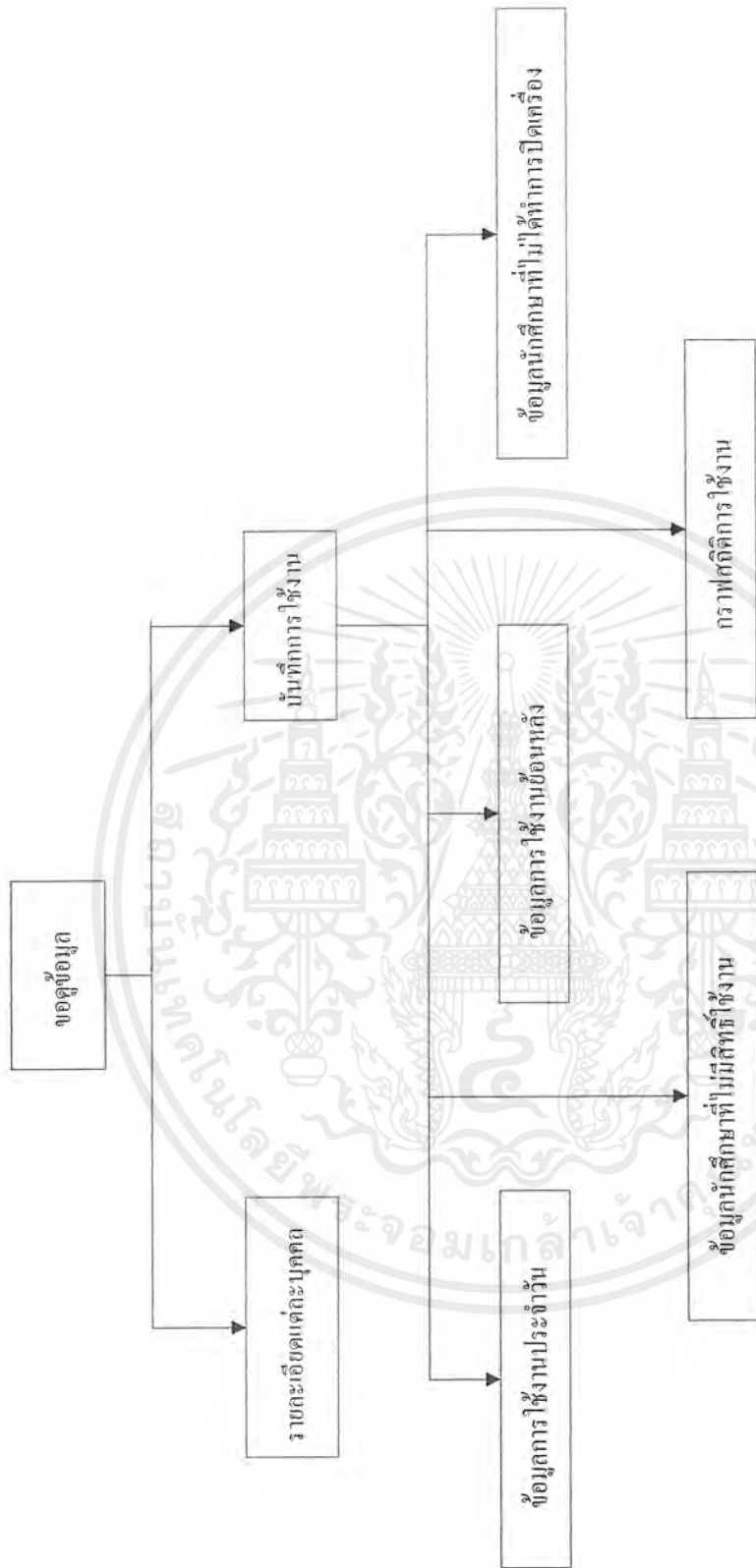


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงแผนผังการทำงานของโปรแกรมตอบสนองการทำงานของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงแผนผังการทำงานของโปรแกรมตอบสนองการทำงานของผู้ใช้ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 หน้าจอหลัก



รูปที่ 3.8 แสดงหน้าจอหลักของโปรแกรม

หน้าจอจะเป็นหน้าจอแสดงฟังก์ชันการทำงานทั้งหมดของ โปรแกรม ซึ่งผู้ใช้สามารถทำการเลือกใช้งานได้โดยการคลิกที่ปุ่มของฟังก์ชันที่ต้องการใช้งาน โดยแต่ละฟังก์ชันมีรายละเอียดดังนี้

#### 3.2.2 ฟังก์ชันขอลูข้อมูล

ฟังก์ชันนี้สามารถแบ่งออกเป็น 2 ส่วน คือ

1. ส่วนรายละเอียดแต่ละบุคคล เป็นส่วนที่จะทำการค้นหา และแสดงข้อมูลของนักศึกษา จากฐานข้อมูลนักศึกษา
2. ส่วนบันทึกการใช้งาน เป็นส่วนแสดงฟังก์ชันย่อยของฟังก์ชันขอลูข้อมูล ดังแสดงในรูปที่ 3.7

โดยผู้ใช้สามารถทำการเลือกใช้งานได้จากแท็บบนฟอรม์ ซึ่งแต่ละส่วนมีรายละเอียดดังนี้

##### 3.2.2.1 ส่วนรายละเอียดแต่ละบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดแต่ละบุคคล | บันทึกการใช้งาน |

ค้นหาข้อมูล

ชื่อ-สกุล

รหัส

ค้นหา

ยกเลิก

รายละเอียด

รหัส

39014001

ชื่อ-สกุล

นริศ ดอริเดช

ชั้นปี

2

สถานะ

ไร้ได้

เหตุผล

ดูข้อมูลคนใหม่

กลับหน้าจอหลัก

ID	NAME	YEAR
39014001	นริศ ดอริเดช	2
39014007	กรกิต เงามะบุญกุล	4
39014022	กณต์ พงกมากร	4
39014036	เกรียงศักดิ์ เหล้าอาภาสุวรรณ	4
39014038	ชวติลา ศรีณชยุปต์	4
39014093	เจนจิรา สิงสกุล	4
39014101	ฉัตรแก้ว จิริไพโรจน์	4

### รูปที่ 3.9 แสดงหน้าจอส่วนรายละเอียดแต่ละบุคคล

ฟังก์ชันในส่วนนี้จะเป็นการนำเอารายละเอียดของนักศึกษา มาแสดงบนหน้าจอ ซึ่งในฟังก์ชันของการค้นหาข้อมูลนี้ สามารถแบ่งออกเป็น 3 ส่วนใหญ่ๆ ดังนี้

1. ส่วนของการค้นหาข้อมูล ในการค้นหาข้อมูล จะสามารถแยกกรณีที่ทำการค้นหาได้ดังนี้

#### 1.1 ทำการค้นหาจากชื่อ-สกุล

ในกรณีนี้ผู้ใช้สามารถกรอกแค่ชื่อเพียงอย่างเดียวได้ ข้อมูลที่ได้จากการค้นหา จะแสดงในตารางข้อมูลใกล้เคียง เช่นเดียวกับกรณีที่ชื่อซ้ำ หรือ หาชื่อไม่พบ ซึ่งถ้าเป็นกรณีไม่พบชื่อผู้ใช้ โปรแกรมจะค้นหาชื่อใกล้เคียงขึ้นมาแสดงผล

#### 1.2 ทำการค้นหาจากรหัสนักศึกษา (ฟิลด์ Id)

ในกรณีนี้จะไม่มีการแสดงผลในตารางข้อมูลใกล้เคียง เนื่องจากถ้าไม่พบรหัสในฐานข้อมูลจะแสดง โดอะล๊อคบด็อบคอบคอบผู้ใช้ทันทีว่า ไม่มีผู้ใช้นี้ในฐานข้อมูลผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ทำการค้นหาจากทั้งชื่อ-สกุล และรหัส

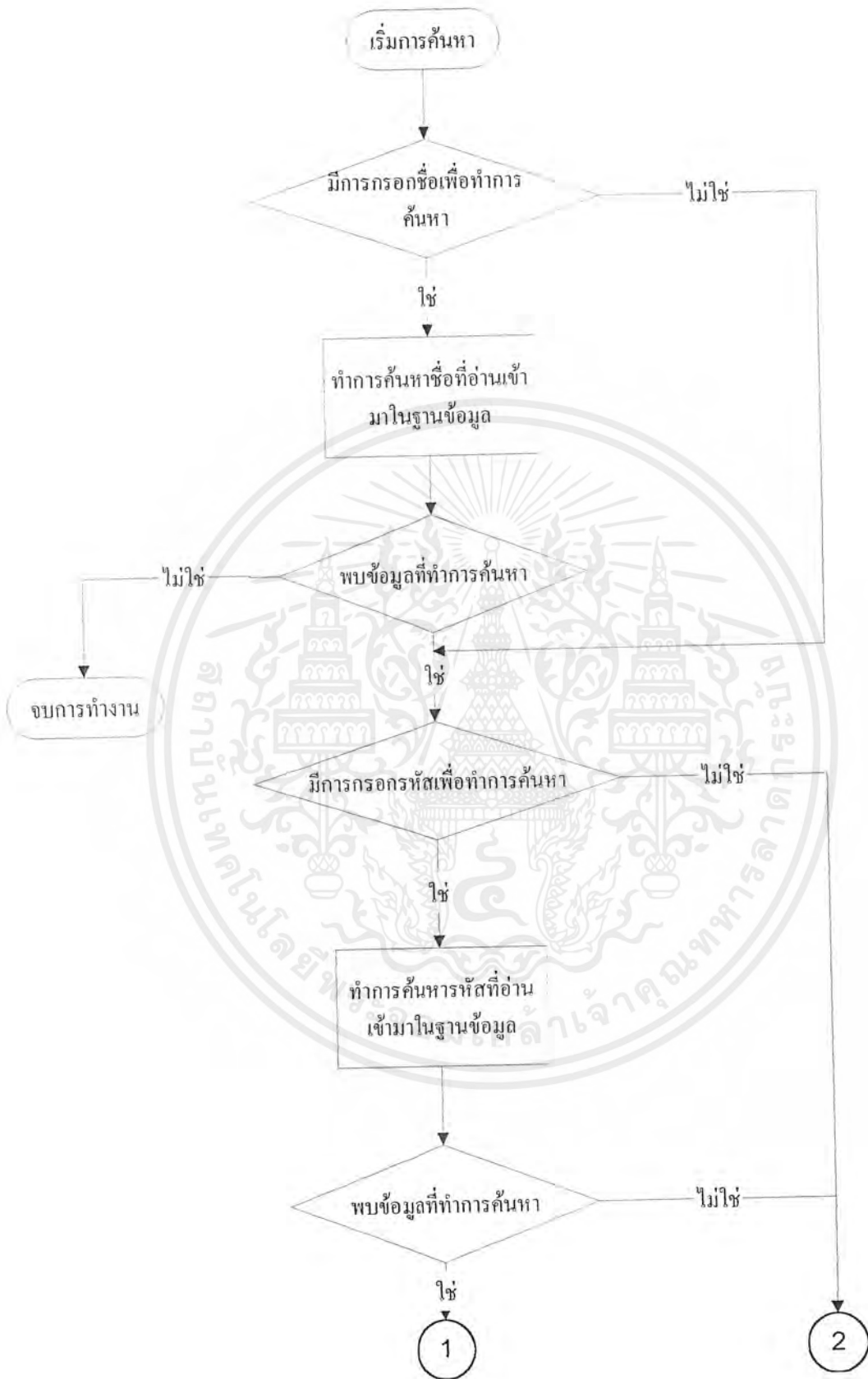
ในกรณีนี้ชื่อที่ทำการค้นหาได้ต้องตรงกับชื่อที่ได้จากการค้นหาโดยใช้รหัส ถ้าไม่ตรงหรือไม่พบ โปรแกรมจะแสดงไอคอนบล็อกล็อกบอกผู้ใช้ทันทีว่า ไม่พบผู้ใช้นี้ในฐานะข้อมูลผู้ใช้

## 2. ส่วนของการแสดงผล สำหรับโปรแกรมนี จะมีส่วนแสดงผล 2 ส่วนด้วยกัน กล่าวคือ

2.1 ส่วนแสดงรายละเอียด ในส่วนนี้จะเป็นการแสดงรายละเอียดส่วนตัวของบุคคลที่ทำการค้นหา ดังต่อไปนี้

- รหัส
- ชื่อ - สกุล
- ชั้นปี
- สถานะ : ( ใช้ได้ / ใช้ไม่ได้ )
- เหตุผล : ในกรณีที่สถานะถูกกำหนดเป็น “ใช้ไม่ได้”

2.2 ส่วนตารางแสดงข้อมูลของผู้ใช้ทั้งหมด ในส่วนนี้จะนำเสนอในรูปแบบของตาราง โดยข้อมูลที่นำมาแสดงจะเป็นข้อมูลของนักศึกษาที่มีรายชื่ออยู่ในฐานข้อมูล และเพื่อเป็นการอำนวยความสะดวกแก่ผู้ใช้ กล่าวคือ ผู้ใช้สามารถทำการเลือกดูรายชื่อที่ต้องการหาจากตารางได้ อีกทางหนึ่ง โดยทำการคลิกที่ชื่อของบุคคลที่ต้องการดูข้อมูล รายละเอียดทั้งหมดจะถูกนำมาแสดงในส่วนแสดงรายละเอียด สำหรับการเลื่อนข้อมูลในตารางสามารถทำได้ 2 วิธี คือ เลื่อนจากสกอลบาร์ (Scroll Bar) หรือทำการเลื่อนจาก นาวีเกเตอร์ (Navigator) ที่อยู่บริเวณด้านบนของตาราง โดยข้อมูลทั้งหมดในฐานข้อมูลจะเรียงตามรหัสนักศึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แผนภาพแสดงการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.2 ส่วนบันทึกการใช้งาน

ในส่วนนี้จะเสมือนเป็นเมนูของฟังก์ชันย่อย ที่ใช้ในการเลือกแสดงผลทางสถิติของการใช้งานต่างๆ ดังรูปที่ 3.11 โดยแต่ละฟังก์ชันย่อยมีรายละเอียดดังนี้



รูปที่ 3.11 หน้าจอแสดงส่วนบันทึกการใช้งาน

#### 3.2.2.2.1 ฟังก์ชันดูข้อมูลการใช้งานประจำวัน

ฟังก์ชันนี้เป็นการใช้คอมพิวเตอร์ DBGrid ในการแสดงจำนวนผู้ใช้งานประจำวัน ซึ่งข้อมูลที่นำมาแสดงเป็นข้อมูลที่อยู่ในฐานข้อมูลผู้ใช้ โดยข้อมูลที่แสดงมีดังนี้

- เวลาที่เริ่มใช้งาน
- เวลาสิ้นสุดการใช้งาน
- หมายเลขเครื่องคอมพิวเตอร์ที่ถูกใช้งาน
- รหัสนักศึกษา
- ชื่อ - สกุล
- ชั้นปี



	2	13:00	14:52	39014101	จัดตั้งแก้ว จิรโพธิ์โรจน์	4
24/10/42	5	10:00	12:00	39014101	จัดตั้งแก้ว จิรโพธิ์โรจน์	4
24/10/42	5	11:00	13:00	39014142	ญาติ ศจรเดช	4

### รูปที่ 3.12 แสดงหน้าจอของฟังก์ชันข้อมูลผู้ใช้งานประจำวัน

ข้อมูลที่นำมาแสดงจะถูกกรองเฉพาะวันที่ปัจจุบัน โดยใช้โอเนต OnFilterRecord ซึ่งโปรแกรมจะสามารถรู้วันที่ปัจจุบันได้จากการดึงฐานเวลาในคอมพิวเตอร์ ดังรูปที่ 3.13

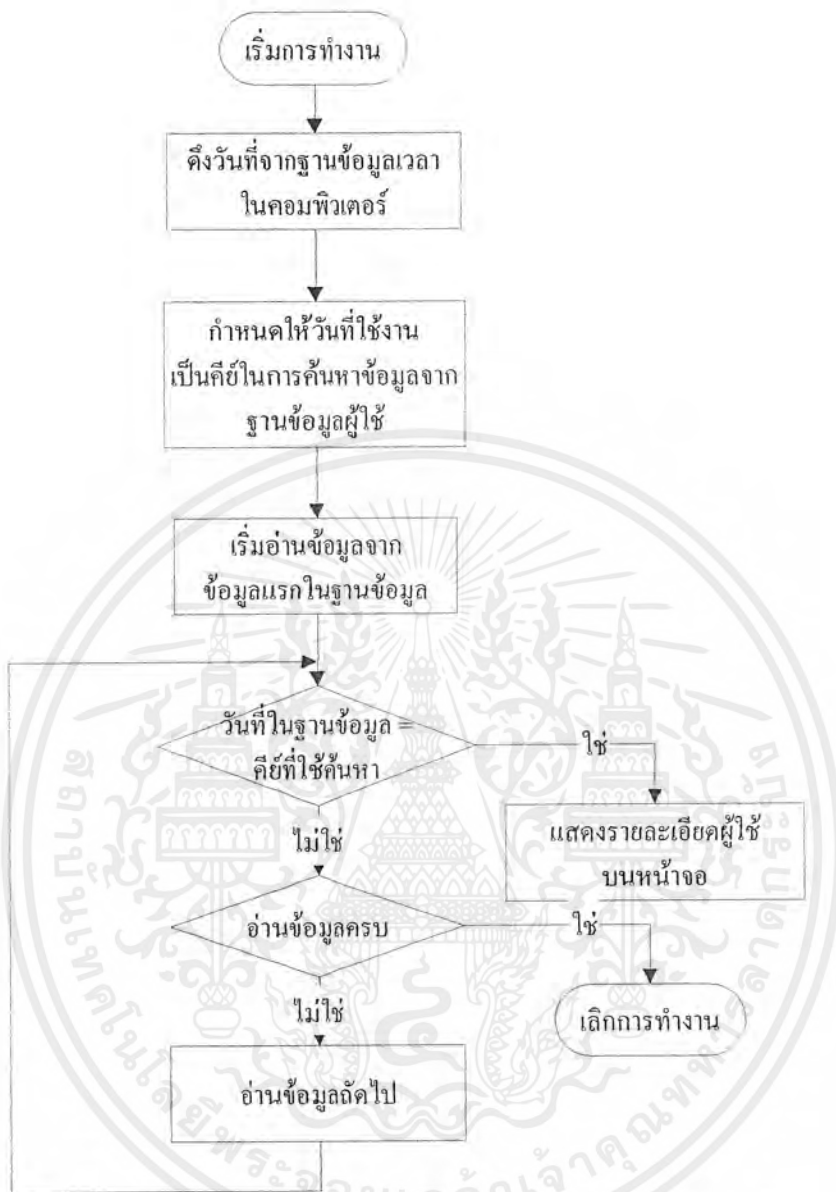
นอกจากนี้ยังมีฟังก์ชันเสริมเข้ามา เพื่ออำนวยความสะดวกในกรณีที่ต้องการพิมพ์ข้อมูลออกทางเครื่องพิมพ์ โดยมี 2 ฟังก์ชัน คือ

- ฟังก์ชันสำหรับดูข้อมูลก่อนพิมพ์

เป็นการใช้คอมโพเนนต์ Quick Report มาใช้แสดงผลข้อมูลก่อนพิมพ์ ดังนั้นข้อมูลที่นำมาแสดง จะอยู่ในรูปของรายงาน ซึ่งมีรูปแบบตามที่โปรแกรมเอาไว้ใน Quick Report

- ฟังก์ชันสำหรับสั่งพิมพ์ข้อมูล

เป็นการใช้คอมโพเนนต์ PrintDialog ดังได้กล่าวรายละเอียดไว้ในบทที่ 2



รูปที่ 3.13 แผนภาพแสดงการทำงานของฟังก์ชันขอข้อมูลผู้ใช้ประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.2.2.2 ฟังก์ชันดูข้อมูลการใช้ย้อนหลัง

\_ | ✕ | X |

ข้อมูลผู้ใช้ประจำวันที่ 24/1/43

Date	Login	Logout	Dept ID	NAME	YEAR
	14:30:12	15:23:45	15	40014074 จักรวรรค์ ทิมเกตุ	3
24/1/43	14:30:12	15:23:45	15	40014074 จักรวรรค์ ทิมเกตุ	3

Today: 12/3/43

เมนู

- 🖨️ พิมพ์
- 📄 ดูข้อมูลงานศึกษา
- 📄 กลุ่มคนโหวตใช้บัตรประชาชน
- 📄 ดูข้อมูลย้อนหลัง

รูปที่ 3.14 แสดงหน้าจอของฟังก์ชันข้อมูลผู้ใช้งานย้อนหลัง

ฟังก์ชันนี้จะมีหลักการทำงานคล้าย ๆ ฟังก์ชันการขอข้อมูลผู้ใช้ประจำวัน นั่นคือมีการกรองข้อมูล โดยใช้วันที่ใช้งานเป็นตัวกรอง แต่ในฟังก์ชันนี้จะทำการรับค่าวันที่จากปฏิทิน ไม่ใช่เป็นการดึงวันที่จากฐานข้อมูลเวลา โดยในลักษณะนี้ผู้ใช้สามารถเลือกวันที่ได้จากปฏิทินทันที โดยทำการคลิกที่วันที่ที่ต้องการดังรูปที่ 3.15

ในส่วนของคุณสมบัติที่นำมาแสดงผล หรือฟังก์ชันย่อยต่างๆ จะเหมือนกับฟังก์ชันการขอข้อมูลผู้ใช้ประจำวัน



รูปที่ 3.15 แผนภาพแสดงการทำงานของฟังก์ชันขอคู่มือข้อมูลผู้ใช้ย้อนหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.2.3 ฟังก์ชันดูข้อมูลนักศึกษาที่ไม่ได้ทำการปิดเครื่อง

	1	10.25	39014101	ฉัตรแก้ว จิรไพโรจน์	4
17/10/42	3	11.00	39014093	เจนจิรา อึ้งสกุล	4
18/10/42	2	13.00	39014142	ภาณี ดลริเดช	4

รูปที่ 3.16 แสดงหน้าจอของฟังก์ชันข้อมูลนักศึกษาที่ไม่ได้ทำการปิดเครื่อง

การทำงานในฟังก์ชันนี้ ยังคงมีการกรองข้อมูลในฐานข้อมูลอยู่ ( ในที่นี้คือฐานข้อมูลผู้ใช้ที่มีการดึงข้อมูลบางส่วนมาจากฐานข้อมูลนักศึกษา ) โดยใช้ฟิลด์ Logout เป็นตัวกรอง นั่นคือ จะมีการแสดงผลเฉพาะผู้ใช้ที่ไม่มีข้อมูลของการรูดบัตรเพื่อทำการปิดเครื่อง โดยในการกรองลักษณะนี้ ถือเป็นการกรองข้อมูลที่ไม่ซ้ำซ้อน ( ต่างจาก 2 แบบแรก ) ดังนั้นจึงสามารถทำการกรองโดยใช้คุณสมบัติ Filter ได้ดังรูปที่ 3.17

ลักษณะข้อมูลที่นำมาแสดงผล และฟังก์ชันต่างๆเหมือนของฟังก์ชันการดูข้อมูลผู้ใช้ประจำวัน

ฟังก์ชันนี้ทำการสร้างขึ้นมาเพื่อให้ผู้ดูแลสามารถทราบได้ว่า ผู้ใช้คนใดที่ไม่ได้ทำการปิดเครื่อง เพื่อจะได้ดำเนินการตักเตือน หรือสามารถระงับสิทธิ์การใช้งานได้ โดยเข้าไปแก้ไขฐานข้อมูลนักศึกษา ในฟิลด์สถานะ



รูปที่ 3.17 แผนภาพแสดงการทำงานของฟังก์ชันขอคู่มือข้อมูลนักศึกษาที่ไม่ปิดเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.2.3 ฟังก์ชันดูข้อมูลนักศึกษาที่ไม่มีสิทธิ์ใช้งาน

ข้อมูลนักศึกษาที่ไม่มีสิทธิ์ใช้งาน

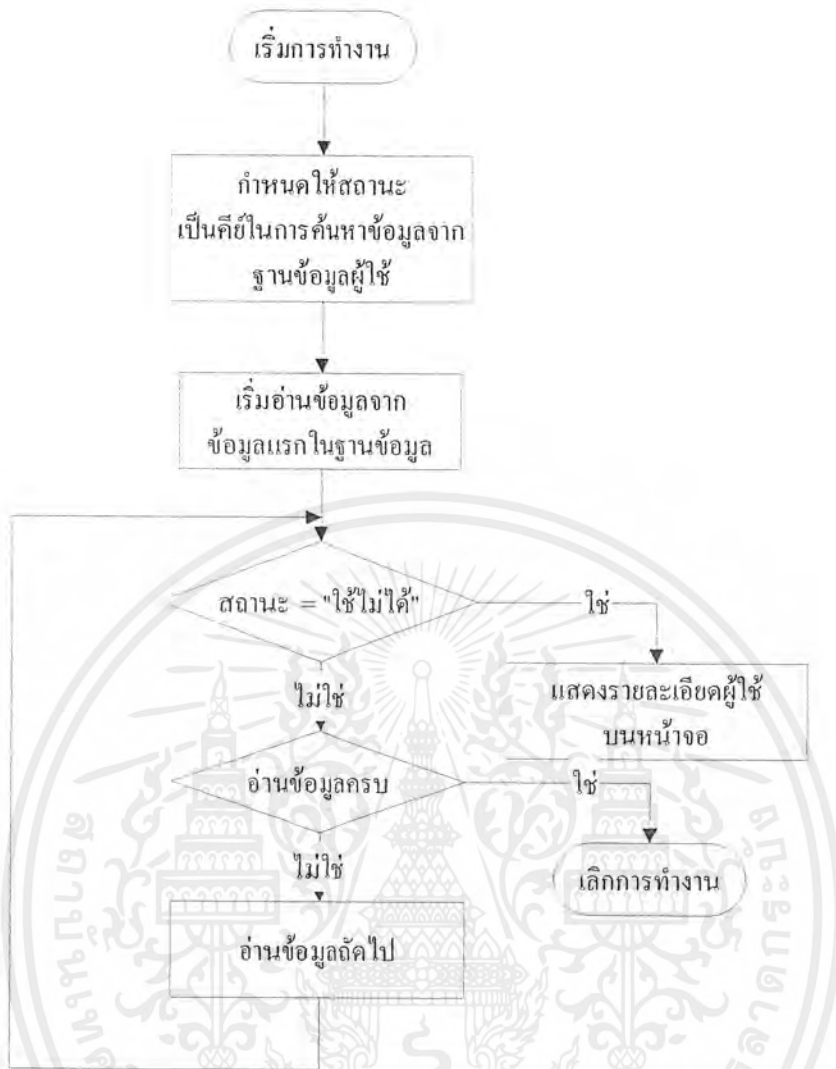
ID	NAME	YEAR	STATUS	REASON
	เกรียงศักดิ์ เหล่าลาภวงศ์	4	ใช้ไม่ได้	
39014101	ฉัตรแก้ว จรุงโรจน์	4	ใช้ไม่ได้	

รูปที่ 3.18 แสดงหน้าจอของฟังก์ชันขอข้อมูลนักศึกษาที่ไม่มีสิทธิ์ใช้งาน

ฟังก์ชันนี้เป็นการแสดงข้อมูลของนักศึกษาที่ไม่มีสิทธิ์ใช้งานพร้อมทั้งเหตุผลที่ทำให้ไม่มีสิทธิ์ใช้งาน โดยฟังก์ชันนี้จะอ้างอิงจากฐานข้อมูลนักศึกษาเท่านั้น ( ฟังก์ชันอื่นต้องอ้างอิงทั้งฐานข้อมูลผู้ใช้และฐานข้อมูลนักศึกษา ) แต่ก็ยังจำเป็นที่จะต้องมีการกรองข้อมูลอยู่ โดยในฟังก์ชันนี้จะใช้ฟิลด์สถานะเป็นตัวกรอง นั่นคือ จะแสดงข้อมูลนักศึกษาที่มีฟิลด์สถานะเป็น “ใช้ไม่ได้” เท่านั้น ลักษณะการกรองจึงไม่ซับซ้อน ดังนั้นสามารถเลือกกรองข้อมูลโดยใช้คุณสมบัติ Filter ได้ดังรูปที่

3.19

ฟังก์ชันเพิ่มเติมอื่น ๆ จะมีลักษณะเหมือนของฟังก์ชันขอข้อมูลผู้ใช้ประจำวัน



รูปที่ 3.19 แผนภาพแสดงการทำงานของฟังก์ชันขอข้อมูลนักศึกษาที่ไม่มีสิทธิ์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.2.5 กราฟแสดงสถิติการใช้งาน

Graph - ๑๑ x

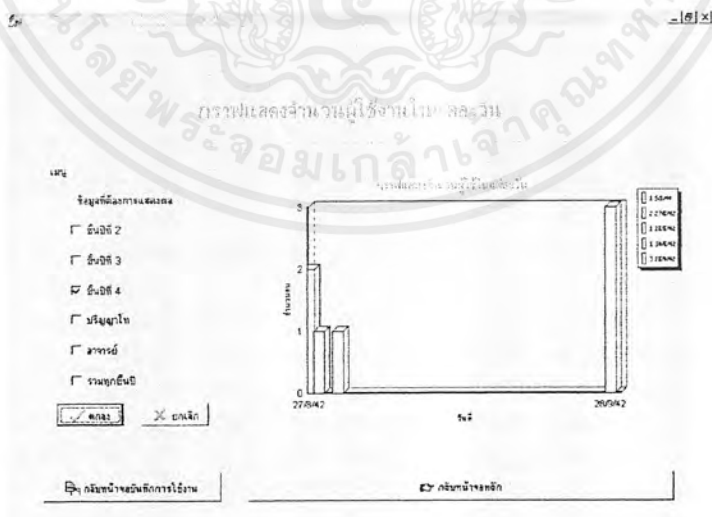
๕

๕

#### รูปที่ 3.20 แสดงหน้าจอกราฟแสดงข้อมูลผู้ใช้

สำหรับฟังก์ชันนี้เป็นฟังก์ชันที่แสดงข้อมูลในรูปแบบของกราฟแก่ผู้ใช้ เพื่อความง่ายและความสะดวกในการวิเคราะห์ข้อมูล โดยที่กราฟที่สามารถแสดงให้ดูได้จะอยู่ในรูปของกราฟแท่ง ในที่นี้สามารถดูกราฟได้ 2 ลักษณะคือ

##### 1. กราฟแสดงจำนวนผู้ใช้ในแต่ละวัน

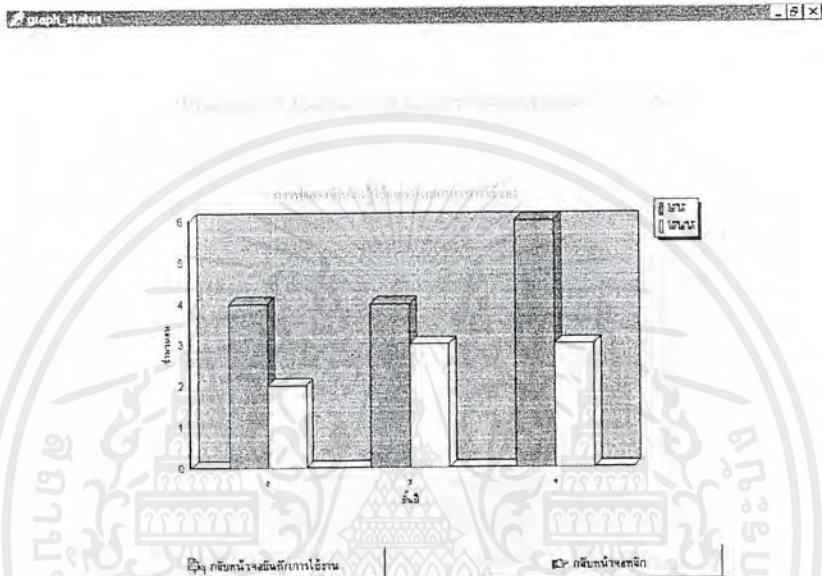


#### รูปที่ 3.21 แสดงหน้าจอกราฟแสดงจำนวนผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับฟังก์ชันนี้จะสามารถเลือกรูปแบบการแสดงผลของกราฟได้ว่าต้องการขอข้อมูลการใช้งานของชั้นปีใดบ้าง จากเมนูด้านข้างของหน้าจอ โดยกราฟที่นำมาแสดง จะเป็นการแสดงสถิติการใช้ภายในระยะเวลา 1 เดือน

## 2. กราฟแสดงจำนวนผู้ใช้แยกตามสถานะการใช้งาน



รูปที่ 3.22 แสดงหน้าจอกราฟแสดงจำนวนผู้ใช้แยกตามสถานะการใช้งาน

เป็นการแสดงข้อมูลการเปรียบเทียบจำนวนผู้ใช้ที่มีสถานะใช้ได้ และใช้ไม่ได้ของแต่ละชั้น

ปี

### 3.2.3 ฟังก์ชันแก้ไข - เพิ่มเติมข้อมูล

ฟังก์ชันนี้สามารถแบ่งออกเป็น 3 ฟังก์ชันย่อย ซึ่งผู้ใช้สามารถเลือกใช้งานได้โดยเลือกคลิกที่แท็บด้านบน โดยฟังก์ชันย่อยประกอบด้วย

- ฟังก์ชันการเพิ่มผู้ใช้
- ฟังก์ชันการแก้ไขข้อมูล
- ฟังก์ชันการลบผู้ใช้

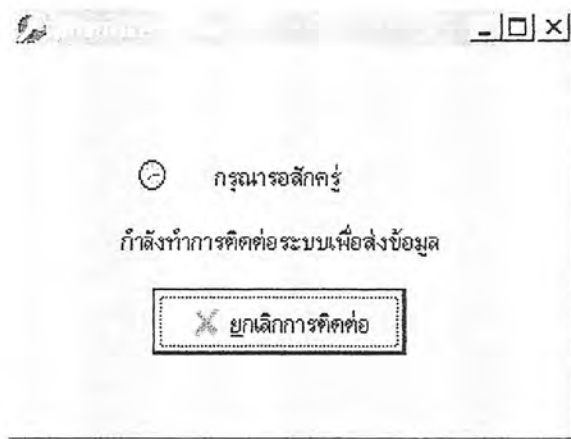
#### 3.2.3.1 ฟังก์ชันการเพิ่มผู้ใช้

ID	NAME	YEAR	ST
39014001	นลินี ดลริเดช	2	ใช้
39014007	กรกิจ เวเนบุรกุล	4	ใช้
39014022	กานต์ พุกษากร	4	ใช้
39014036	เกรียงศักดิ์ เหล่าอาภาวงศ์	4	ใช้
39014038	ชวจิต ศรีณยุปต์	4	ใช้
39014093	เจนจิรา สิงสกุล	4	ใช้
39014101	ฉัตรนภัท จิรไพโรจน์	4	ใช้
39014142	ญาณี ดลริเดช	4	ใช้
39014238	ธนาภาวดี สิงหนิม	4	ใช้

รูปที่ 3.23 แสดงหน้าจอของฟังก์ชันการเพิ่มข้อมูลผู้ใช้

ในการเพิ่มข้อมูลผู้ใช้ จะเป็นการเพิ่มข้อมูลเข้าฐานข้อมูลนักศึกษา โดยในขั้นแรกต้องทำการกรอกรายละเอียดให้ครบ และเมื่อทำการกดปุ่ม “เพิ่ม” จะเป็นการส่งสัญญาณไปติดต่อกับบอร์ดหลัก ในขณะที่โปรแกรมจะแสดงหน้าจอ ดังรูปที่ 3.24 และเมื่อทำการติดต่อได้จะแสดงหน้าจอเพื่อให้ผู้ใช้ทำการรูดบัตร ดังรูปที่ 3.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



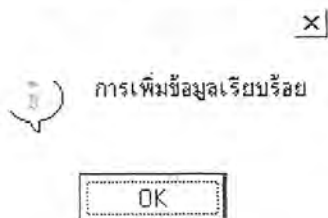
รูปที่ 3.24 หน้าจอแสดงสถานะการขอติดต่อบอร์ดหลัก



รูปที่ 3.25 หน้าจอแสดงให้ผู้ใช้ทำการรูดบัตร

เมื่อผู้ใช้ทำการรูดบัตร หน้าจอจะมีแถบวิ่งปรากฏขึ้น และบอร์ดหลักจะทำการอ่านข้อมูลขนาด 40 ไบต์จากเครื่องอ่านบัตรแม่เหล็ก เพื่อทำการเก็บข้อมูล และทำการส่งข้อมูลนั้นมายังคอมพิวเตอร์ เมื่อคอมพิวเตอร์ได้รับข้อมูลขนาด 40 ไบต์ จะทำการสร้างรหัสประจำตัวผู้ใช้งานขนาด 3 ไบต์ขึ้น เพื่อใช้ในการติดต่อกับบอร์ดหลักในการทำงานครั้งต่อไป

เมื่อบอร์ดหลักรับข้อมูลขนาด 3 ไบต์ บอร์ดหลักจะทำการตรวจสอบความถูกต้องและส่งสัญญาณยกเลิกการติดต่อในกรณีที่ข้อมูลที่ทำการส่งถูกต้อง ในกรณีนี้เมื่อได้รับสัญญาณยกเลิกการติดต่อ จะทำการเพิ่มข้อมูลผู้ใช้เข้าฐานข้อมูล และแสดงหน้าจอดังรูปที่ 3.26



รูปที่ 3.26 แสดงหน้าจอสิ้นสุดการเพิ่มข้อมูล

ส่วนในกรณีที่การส่งข้อมูลไปยังบอร์ดหลักผิดพลาด บอร์ดหลักจะทำการส่งสัญญาณการส่งข้อมูลผิดพลาด โปรแกรมจะแสดงหน้าจอดังรูปที่ 3.27 ถ้าผู้ใช้เลือกปุ่ม OK ระบบจะเริ่มทำการติดต่อกันใหม่ แต่ถ้าเลือกปุ่ม Cancel ระบบจะเลิกทำการติดต่อกับบอร์ดหลัก



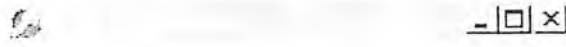
รูปที่ 3.27 หน้าจอแสดงการส่งข้อมูลผิดพลาด



รูปที่ 3.28 แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอรืและบอร์ดหลัก ในการเพิ่มข้อมูล



ในขณะที่ทำการส่งข้อมูล โปรแกรมจะแสดงหน้าจอดังรูป



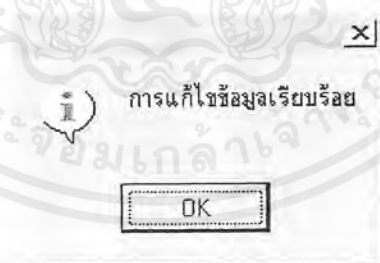
กรณารอส่งข้อมูล

กำลังทำการส่งข้อมูล

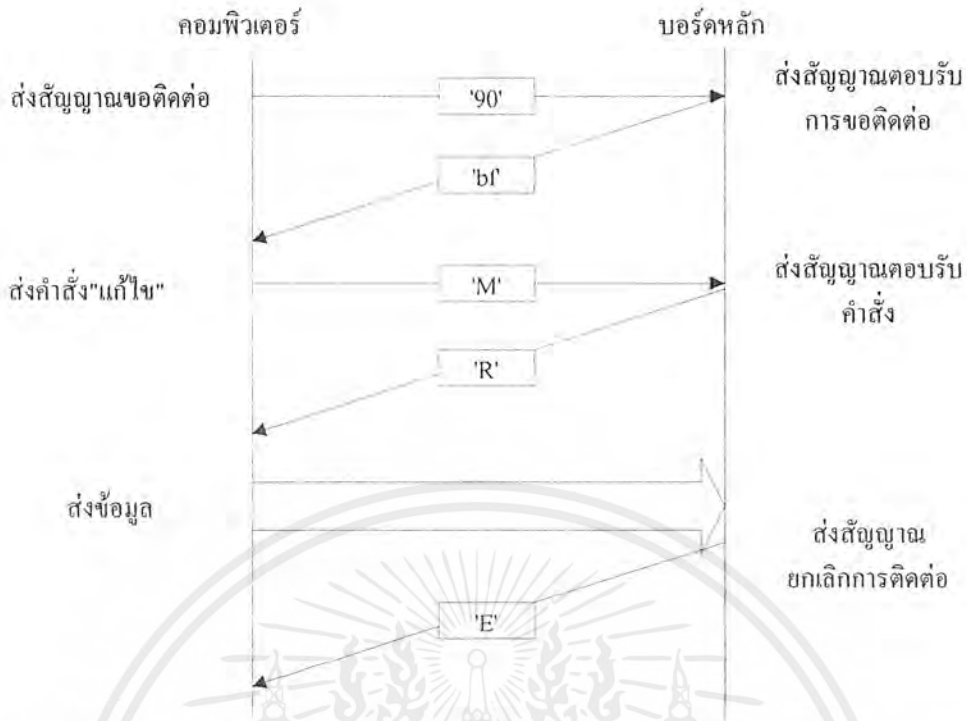
X ยกเลิกการส่งข้อมูล

รูปที่ 3.30 แสดงหน้าจอสถานะการรับ-ส่งข้อมูล

เมื่อบอร์ดหลักรับข้อมูลเรียบร้อยแล้ว จะทำการตรวจสอบความถูกต้องของข้อมูลด้วยวิธีผลรวม ถ้าการส่งข้อมูลถูกต้อง บอร์ดหลักจะทำการแก้ไขฐานข้อมูลตามข้อมูลที่ส่งไปให้ และจะส่งสัญญาณยกเลิกการติดต่อกลับมา นั่นคือ การแก้ไขเรียบร้อยแล้ว โปรแกรมจะแสดงหน้าจอดังรูปที่ 3.31 แต่ในกรณีที่การส่งข้อมูลผิดพลาด บอร์ดหลักจะทำการส่งสัญญาณการส่งข้อมูลผิดพลาดกลับมา เหมือนกรณีการเพิ่มผู้ใช้



รูปที่ 3.31 แสดงหน้าจอการแก้ไขข้อมูลเรียบร้อยแล้ว

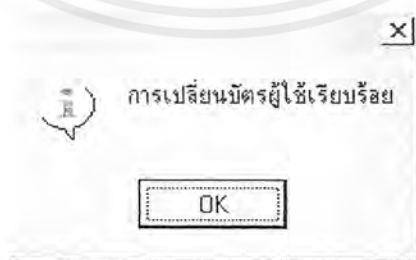


รูปที่ 3.32

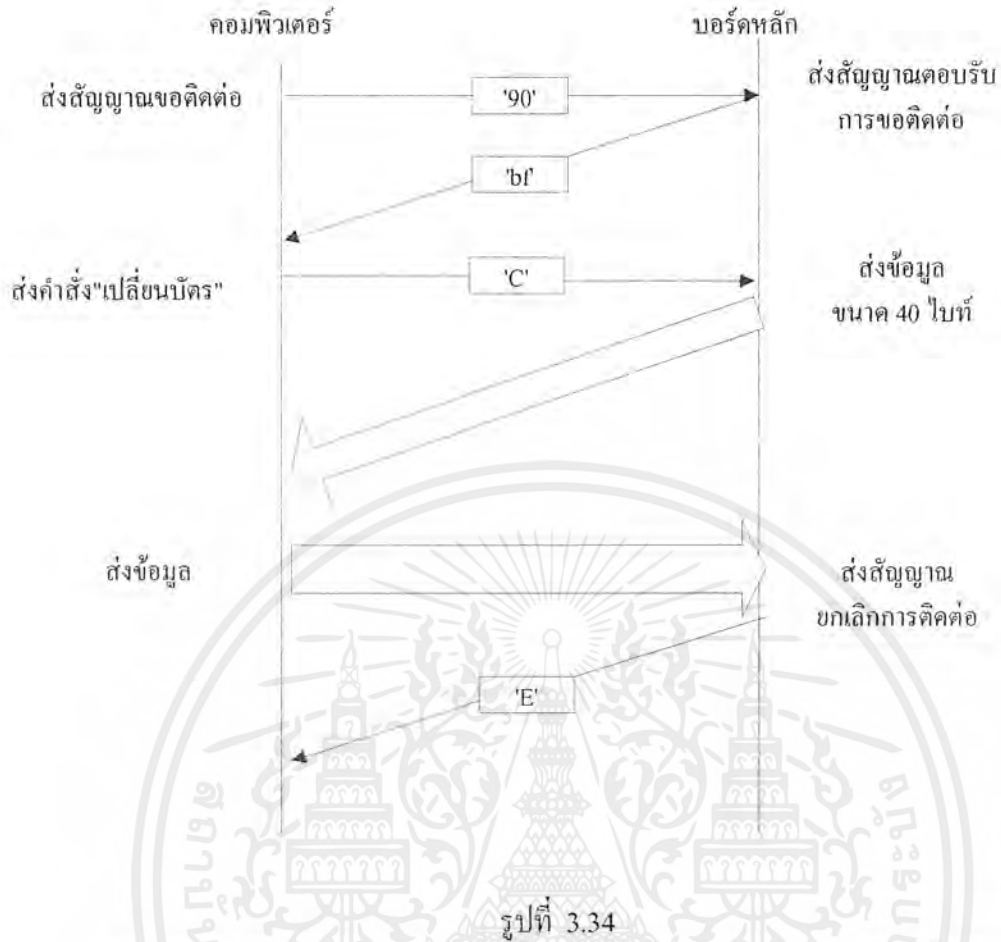
แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเคอร์และบอร์ดหลัก ในการแก้ไขข้อมูล

### 3.2.3.3 ฟังก์ชันการเปลี่ยนบัตรใหม่

ในกรณีที่ทำการเปลี่ยนบัตรแม่เหล็กใหม่ โปรแกรมจะทำการติดต่อกับบอร์ดหลักเหมือนในกรณีของการเพิ่มผู้ใช้ โดยฟังก์ชันนี้จะแตกต่างจากฟังก์ชันการเพิ่มผู้ใช้ ตรงคำสั่งที่ใช้ในการติดต่อกับบอร์ดหลัก และในฟังก์ชันนี้โปรแกรมไม่จำเป็นต้องสร้างรหัสประจำตัวผู้ใช้ขนาด 3 ไบต์ใหม่ เพียงแต่ทำการแก้ไขข้อมูลขนาด 40 ไบต์ในฐานข้อมูลนักศึกษาเท่านั้น



รูปที่ 3.33 แสดงหน้าจอสิ้นสุดการเปลี่ยนบัตร



แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลัก ในการเปลี่ยนบิต

#### 3.2.3.4 ฟังก์ชันการลบข้อมูลผู้ใช้

ฟังก์ชันนี้จะเป็นฟังก์ชันสำหรับการลบข้อมูลผู้ใช้ออกจากฐานข้อมูลนักศึกษา โดยรูปแบบการทำงานจะเหมือนในส่วนของฟังก์ชันการแก้ไขข้อมูล แต่เมื่อบอร์ดหลักรับข้อมูลเรียบร้อยแล้วและส่งสัญญาณยกเลิกการติดต่อกลับมา โปรแกรมจะทำการลบข้อมูลผู้ใช้ออกจากฐานข้อมูลทันที และแสดงหน้าจอสิ้นสุดการลบข้อมูลดังรูปที่ 3.37

เพิ่มข้อมูล | แก้ไขข้อมูล | ลบข้อมูล

ค้นหาข้อมูล

ชื่อ-สกุล

รหัส

ค้นหา

ยกเลิก

ลบข้อมูล

รหัส: 39014001 ชื่อ-สกุล: นลินี ตริเตช

ชั้นปี: 2 สถานะ: ไล่ได้

เหตุผล

ลบ

ยกเลิก

ข้อมูลใกล้เคียง

ID	NAME
39014001	นลินี ตริเตช
39014007	กรกิจ เชาบเมฆกุล
39014022	กานต์ พฤกษการ
39014036	เกรียงศักดิ์ เหล่าลาภสงวณ
39014038	ชวติต ตริณยศุภัต
39014093	เจนจิรา อึ้งสกุล
39014101	ศัตตเมศวร์ ศิริไพโรจน์

ID	NAME
39014001	นลินี ตริเตช
39014007	กรกิจ เชาบเมฆกุล
39014022	กานต์ พฤกษการ
39014036	เกรียงศักดิ์ เหล่าลาภสงวณ

ปุ่มพิมพ์เอกสาร

รูปที่ 3.35 แสดงฟังก์ชันการลบผู้ใช้



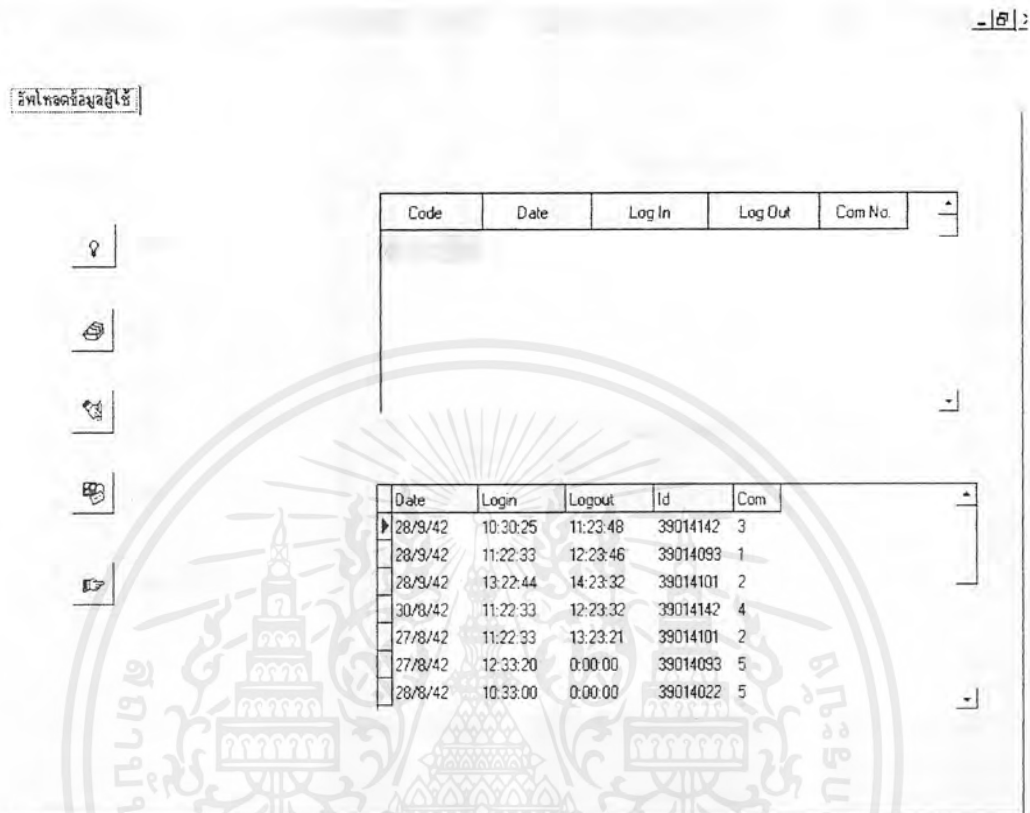
รูปที่ 3.36 แสดงหน้าจอสิ้นสุดการลบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.4 ฟังก์ชันการถ่ายโอนข้อมูลเข้า ( Upload )



รูปที่ 3.38 แสดงหน้าจอฟังก์ชันการถ่ายโอนข้อมูลเข้า

ฟังก์ชันนี้จะเป็นการถ่ายโอนข้อมูลจากบอร์ดหลักมาเก็บในฐานข้อมูลผู้ใช้ เนื่องจากบอร์ดหลักมีหน่วยความจำที่จำกัด และเพื่อเป็นข้อมูลในการแสดงผลทางสถิติของโปรแกรม โดยข้อมูลที่ทำการถ่ายโอนเข้ามามีรูปแบบดังนี้

จำนวนคน	'a'	รหัส 3 ไบต์	'#'	วันที่ใช้งาน	'#'	เวลาเริ่มใช้	'#'
เวลาสิ้นสุด	'#'	หมายเลขเครื่อง	'#'	ตัวตรวจสอบ ความถูกต้อง	'z'		

หมายเหตุ	a	เป็น ไบต์เริ่มต้นข้อมูล
	#	เป็น ไบต์คั่นข้อมูล
	z	เป็น ไบต์สิ้นสุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

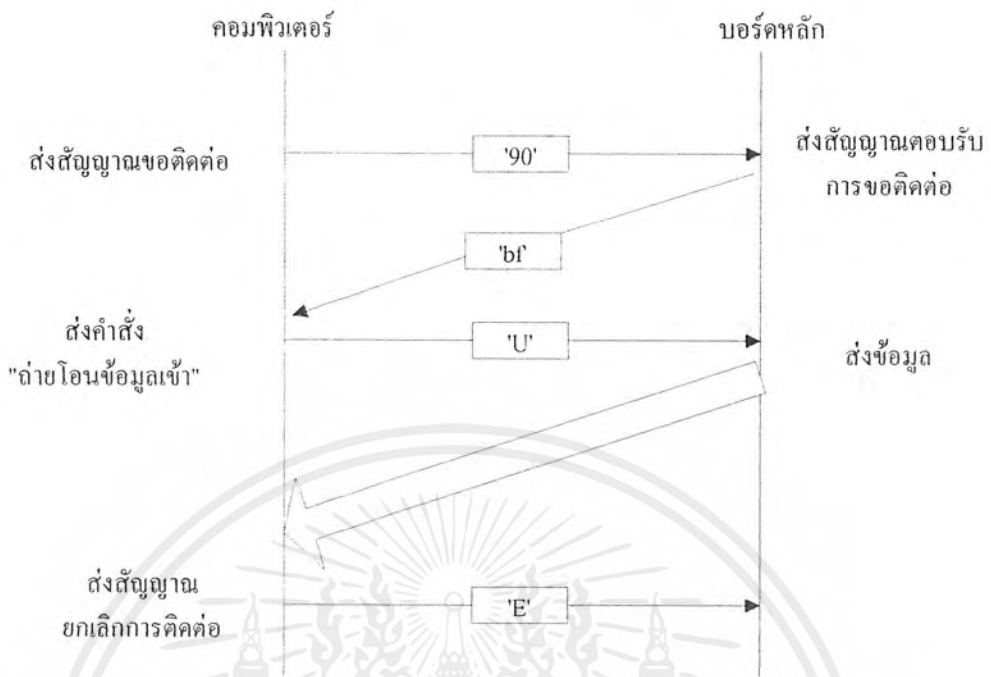
ในขั้นแรกของการใช้ฟังก์ชัน ผู้ใช้ต้องทำการติดต่อพอร์ทอนุกรมก่อน โดยการคลิกที่ปุ่มติดต่อพอร์ท เมื่อได้รับสัญญาณตอบรับการติดต่อ ขั้นตอนที่ต่อไปจะเป็นการส่งสัญญาณคำสั่งไปยังบอร์ดหลัก หลังจากนั้นจะทำการรับข้อมูลที่บอร์ดหลักถ่ายโอนมาให้ เพื่อตรวจสอบความถูกต้อง และทำการแยกข้อมูลของผู้ใช้แต่ละคนตามฟิลด์ต่างๆ เพื่อนำมาทำการแสดงผลบนหน้าจอ แต่ในกรณีที่การส่งข้อมูลผิดพลาด โปรแกรมจะแสดงหน้าจอดังรูปที่ 3.27

ในขั้นตอนแรกที่กำลังมาเป็นเพียงการรับข้อมูลมาแสดงผลเท่านั้น ยังไม่ได้ทำการเก็บข้อมูลลงในฐานข้อมูลผู้ใช้ ถ้าต้องการจัดเก็บในฐานข้อมูล ผู้ใช้ต้องทำการคลิกที่ปุ่ม “เก็บข้อมูลเข้าฐานข้อมูล” ซึ่งโปรแกรมจะทำการค้นหาข้อมูลของผู้ใช้จากฐานข้อมูลนักศึกษา ก่อน โดยใช้รหัส 3 ไบต์เป็นคีย์ในการค้นหา ถ้าไม่พบข้อมูลในฐานข้อมูลนักศึกษาจะแสดงหน้าจอดังรูปที่ 3.39



รูปที่ 3.39

แต่ในกรณีที่มิข้อมูลในฐานข้อมูล โปรแกรมจะทำการจัดเก็บข้อมูลที่รับมาเข้าฐานข้อมูลผู้ใช้ และส่งสัญญาณยกเลิกการติดต่อไปยังบอร์ดหลัก เมื่อบอร์ดหลักได้รับสัญญาณดังกล่าวจะทำการลบฐานข้อมูลจากหน่วยความจำภายในทั้งหมด เพื่อที่จะนำมาจัดเก็บข้อมูลการใช้งานในครั้งต่อไป



รูปที่ 3.40

แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์คหลัก ในการถ่ายโอนข้อมูลเข้า

3.2.5 ฟังก์ชันการถ่ายโอนข้อมูลออก (Download)

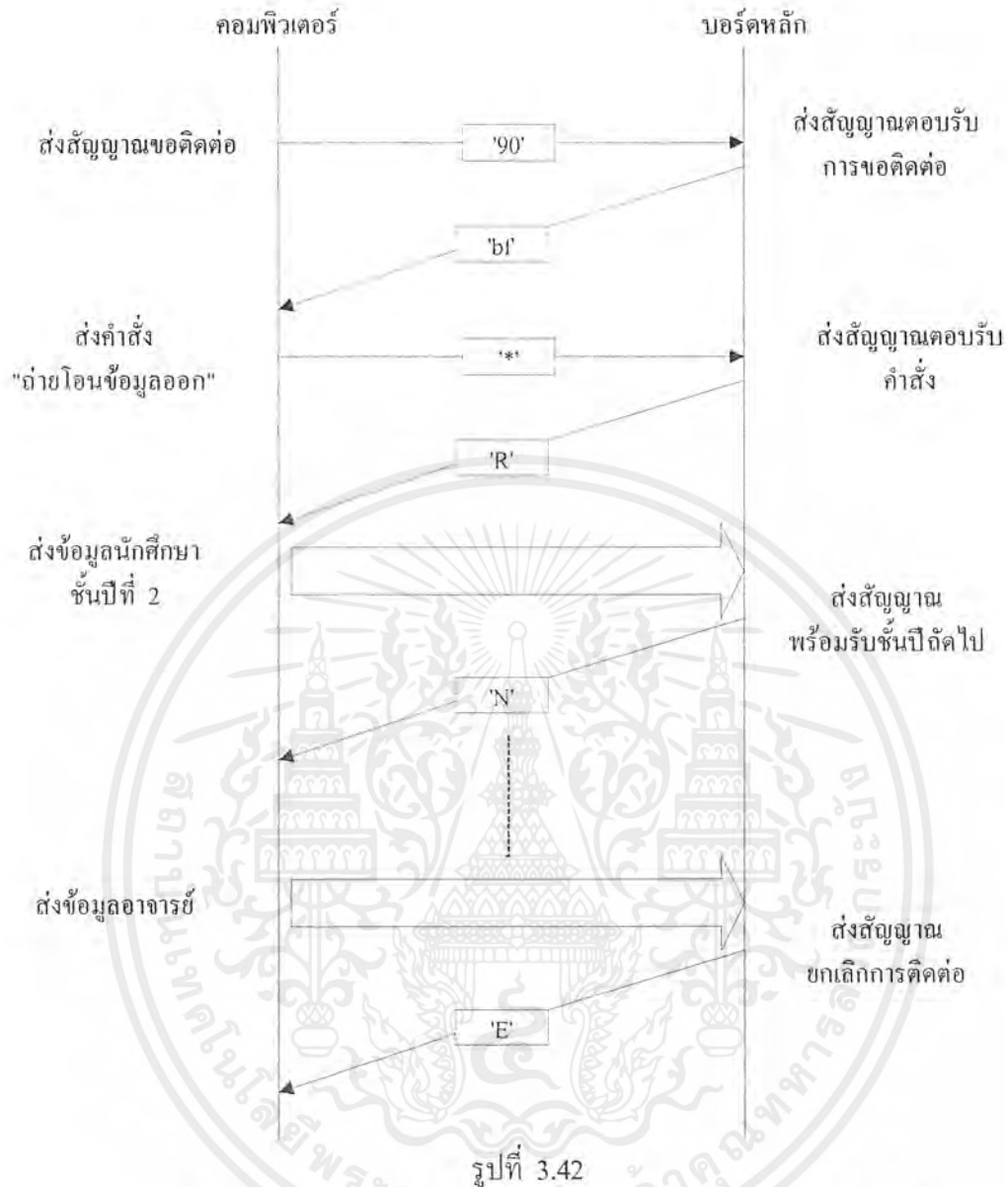
เมนู

รายชื่อของนักศึกษา

Id	Sex	Name	Year	Status
39014007	นาย	กรกิจ เจาเนอญจกุล	4	ใช้ได้
39014022	นาย	กานต์ พุกษากร	4	ใช้ไม่ได้
39014036	นาย	เกรียงศักดิ์ เพ็ญลาภาสุวรรณท์	4	ใช้ไม่ได้
39014088	นาย	จุฑาพล คล้ายจำเริญ	4	ใช้ไม่ได้
39014093	นางสาว	เจนจิรา สิงขกุล	4	ใช้ได้
39014101	นางสาว	ฉัตรแก้ว จิโรโรจน์	4	ใช้ได้
39014118	นาย	ชวฉัตร ศรีผยงยุรัตน์	4	ใช้ได้
39014123	นาย	ชاکกริต กิจนแก้ว	4	ใช้ได้
39014142	นางสาว	ญาณี ดลจริตเดช	4	ใช้ได้
40014073	นาย	จักรวาล วิจารณ์บุตร	3	ใช้ไม่ได้
40014074	นาย	จักรวาล ทัพเกษ	3	ใช้ได้

รูปที่ 3.41 แสดงฟังก์ชันการถ่ายโอนข้อมูลออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงการส่งสัญญาณติดต่อระหว่างคอมพิวเตอร์และบอร์ดหลัก ในการถ่ายโอนข้อมูลออก

จากรูปที่ 3.48 จะเห็นว่าในขั้นแรกต้องทำการส่งสัญญาณขอทำการติดต่อกับบอร์ดหลักก่อน โดยการกดปุ่ม "ติดต่อพอร์ท" พร้อมทั้งรอสัญญาณตอบรับการขอติดต่อ และเมื่อผู้ใช้ทำการกดปุ่มส่งข้อมูล โปรแกรมจะทำการส่งข้อมูลโดยมีรูปแบบของข้อมูลดังนี้

'Y'	จำนวนคน	ข้อมูลในชั้นปี
-----	---------	----------------

หมายเหตุ Y เป็นไบนารีเริ่มต้นการส่งข้อมูลของชั้นปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นคือ โปรแกรมจะทำการส่งข้อมูลไปครั้งละชั้นปี เมื่อได้รับสัญญาณพร้อมรับชั้นปีถัดไป จึงจะทำการส่งข้อมูลต่อจนครบ โดยรูปแบบของข้อมูลในแต่ละชั้นปีเป็นดังนี้



หมายเหตุ

a	เป็น ไบต์เริ่มต้นข้อมูล
#	เป็น ไบต์คั่นข้อมูล
z	เป็น ไบต์สิ้นสุดข้อมูล

ฟังก์ชันนี้มีไว้เพื่อทำการส่งข้อมูลจากคอมพิวเตอร์ไปยังฐานข้อมูลของบอร์ดหลัก ในกรณีที่ข้อมูลมีการเปลี่ยนแปลง หรือข้อมูลในฐานข้อมูลของบอร์ดหลักสูญหาย ซึ่งบอร์ดหลักจะทำการรับข้อมูลครั้งละ 1 ชั้นปี มาทำการตรวจสอบข้อมูล และนำข้อมูลที่ได้ไปเก็บในหน่วยความจำ ถ้าในกรณีที่การส่งข้อมูลถูกต้อง บอร์ดหลักจะส่งสัญญาณพร้อมรับชั้นปี ถัดไปมายังคอมพิวเตอร์ เพื่อให้ทำการส่งข้อมูลของชั้นปีถัดไปให้จนครบ แต่ในกรณีที่การส่งข้อมูลผิดพลาด บอร์ดหลักจะส่งสัญญาณการส่งข้อมูลผิดพลาดมายังคอมพิวเตอร์ และ โปรแกรมจะแสดงหน้าจอการส่งข้อมูลผิดพลาดดังรูปที่ 3.27

### 3.2.6 Hclp

ฟังก์ชันนี้จัดทำขึ้นเพื่อช่วยเหลือผู้ใช้ในการใช้โปรแกรม โดยแบ่งออกเป็นส่วนต่างๆดังนี้

- ระบบฐานข้อมูล
- รายละเอียดของโปรแกรม
- วิธีการใช้งาน โปรแกรม

โดยรายละเอียดในแต่ละส่วน จะเหมือนกับรายละเอียดที่ได้กล่าวไว้แล้วในบทที่ 2 และบทที่ 3

## บทที่ 4

### การทดลองและผลการทดลอง

ในส่วนของการทดลองของโปรแกรม จะเป็นการทดลองเกี่ยวกับการติดต่อสื่อสารผ่านทางพอร์ทอนุกรม ดังที่ได้กล่าวไว้แล้วในบทที่ 1 ว่าส่วนของโปรแกรมสามารถแยกออกเป็น 2 ส่วน คือ โปรแกรมบนเครื่องคอมพิวเตอร์ และโปรแกรมบนบอร์ดหลัก

#### 4.1 การทดลองของโปรแกรมบนเครื่องคอมพิวเตอร์

##### 4.1.1 ทำการทดลองส่งค่าออกทางพอร์ทอนุกรม

4.1.1.1 ทำการทดลองส่งข้อมูลที่เป็นสตริงออกทางพอร์ทอนุกรมโดยใช้คำสั่ง WriteString ซึ่งมีรูปแบบคำสั่งดังนี้

```
WriteString (str,true);
```

เมื่อกำหนดตัวแปร str เป็นตัวแปรที่ใส่ข้อมูลสตริงที่ต้องการส่งค่าออก

โดยในที่นี้ได้ใช้เครื่องไมโครคอมพิวเตอร์แผ่นพิมพ์เดี่ยว (Single Board Microcomputer) รุ่น Jazz31 V2.0 เป็นเครื่องมือที่นำมาใช้ทดสอบการรับค่า เนื่องจากเครื่องนี้สามารถทำการรับ-ส่งค่าแบบอนุกรมได้

ขั้นตอนการทดลอง

1. เตรียมเครื่อง Jass31 V2.0 ให้พร้อมที่จะทำการรับค่าโดยใช้ Fn2 RXB แล้วใส่ตำแหน่งหน่วยความจำแรกที่ใช้ในการเก็บค่า ในที่นี้กำหนดเป็น 9000H สังเกตได้ว่าจะมีจุดสีแดงขึ้นบนหน้าจอ แสดงว่าเครื่องพร้อมที่จะทำการรับข้อมูลแล้ว

2. ทำการติดต่อพอร์ทและส่งข้อมูลที่เป็นสตริงออกทางพอร์ท โดยให้ลองส่งค่า 'a' ออกทางพอร์ท

3. ทำการตรวจสอบค่าที่ส่งโดยเข้าไปอ่านที่หน่วยความจำของเครื่อง Jass31 V2.0 โดยใช้ฟังก์ชัน DMEM แล้วใส่ค่าตำแหน่งเริ่มต้น ซึ่งในที่นี้คือ 9000H

จากการทดลองพบว่าเมื่อทำการส่งค่า 'a' ออกทางพอร์ท เมื่อไปทำการตรวจสอบค่าที่ตำแหน่งหน่วยความจำ 9000H ของ Jass31 V2.0 พบว่าเป็นค่า 61 H ซึ่งเป็นรหัสแอสกี (ASCII CODE) ของตัวอักษร 'a' นั่นเอง

4.1.1.2 ทำการทดลองส่งข้อมูลที่เป็นสตริงออกทางพอร์ทอนุกรมโดยใช้คำสั่ง WriteString แต่ในขั้นตอนนี้จะทดลองทำการส่งข้อมูลขนาดหลายตัวอักษร เนื่องจากในการทำงานจริงนั้น ข้อมูลที่ทำการส่งจะมีขนาดยาวมาก

จากการทดลองพบว่า ด้วยคอมพิวเตอร์ที่ใช้ในการส่ง สามารถทำการส่งข้อมูลได้ยาวโดยไม่จำกัดตัวอักษร แม้ว่าข้อมูลที่ทำการส่งนั้นจะมีขนาดยาวกว่าขนาดของบัฟเฟอร์ของพอร์ทอนุกรมก็ตาม

#### 4.1.2 ทำการทดลองรับค่าจากพอร์ทอนุกรม

4.1.2.1 ทำการทดลองรับข้อมูลสตรงจากพอร์ทอนุกรม โดยใช้คำสั่ง ReadString ซึ่งมีรูปแบบคำสั่งดังนี้

```
ReadString(str,inque,true);
```

เมื่อกำหนดตัวแปร str เป็นตัวแปรที่จะทำการรับข้อมูลสตรงที่ทำการรับเข้ามา โดยในที่นี้ได้ใช้เครื่อง Jazz31 V2.0 เป็นตัวที่นำมาใช้ทดสอบการส่งค่า เนื่องจากเครื่องนี้สามารถทำการรับ-ส่งค่าแบบอนุกรมได้

1. ทำการเตรียมเครื่อง Jazz31 V2.0 ในการส่งข้อมูลโดยทำการใส่ข้อมูลที่ต้องการส่งไว้ในหน่วยความจำ ( สามารถทำการเลือกตำแหน่งหน่วยความจำที่ต้องการเก็บค่าข้อมูลได้ โดยใช้ฟังก์ชัน DMEM แล้วตามด้วยตำแหน่งหน่วยความจำแรกที่ต้องการ ) โดยในการป้อนข้อมูลต้องป้อนเป็นข้อมูลฐาน 16 ตามรหัสแอสกี
  2. ทำการติดต่อพอร์ทอนุกรม
  3. ทำการส่งข้อมูลจากทางเครื่อง Jazz31 V2.0 โดยใช้ฟังก์ชัน Fn2 TXB ตามด้วยตำแหน่งหน่วยความจำเริ่มต้น และหน่วยความจำสุดท้ายที่ต้องการส่ง
  4. แสดงผลที่รับมาบนหน้าจอเพื่อตรวจสอบความถูกต้อง
- จากการทดลองพบว่า สามารถทำการรับค่าที่ส่งมาได้ถูกต้อง โดยไม่จำกัดความยาวเช่นกัน

4.1.2.2 ทำการทดลองรับข้อมูลคาร์ดินัล ( Cardinal ) จากพอร์ทอนุกรม โดยใช้คำสั่ง Read ซึ่งมีรูปแบบคำสั่งดังนี้

```
Read (int,inque,true);
```

เมื่อกำหนดตัวแปร int เป็นตัวแปรที่จะทำการรับข้อมูลคาร์ดินัล โดยในที่นี้ได้ใช้เครื่อง Jazz31 V2.0 เป็นตัวที่นำมาใช้ทดสอบการส่งค่า เนื่องจากเครื่องนี้สามารถทำการรับ-ส่งค่าแบบอนุกรมได้

### ขั้นตอนการทดลอง

1. ทำการเตรียมเครื่อง Jazz31 V2.0 ในการส่งข้อมูลโดยทำการใส่ข้อมูลที่ต้องการส่งไว้ในหน่วยความจำ ( สามารถทำการเลือกตำแหน่งหน่วยความจำที่ต้องการเก็บค่าข้อมูลได้ โดยใช้ฟังก์ชัน DMEM แล้วตามด้วยตำแหน่งหน่วยความจำแรกที่ต้องการ ) โดยในการป้อนข้อมูลต้องป้อนเป็นข้อมูลฐาน 16 ตามรหัสแอสกี

2. ทำการติดต่อพอร์ตอนุกรม

3. ทำการส่งข้อมูลจากทางเครื่อง Jazz31 V2.0 โดยใช้ฟังก์ชัน Fn2 TXB ตามด้วยตำแหน่งหน่วยความจำเริ่มต้น และหน่วยความจำสุดท้ายที่ต้องการส่ง

4. แสดงผลที่รับมาบนหน้าจอเพื่อตรวจสอบความถูกต้อง

จากการทดลองพบว่า สามารถทำการรับค่าที่ส่งมาได้ถูกต้อง แต่ความยาวในการรับส่งจะถูกจำกัดด้วยขนาดของตัวแปร int เนื่องจากในเคสไฟ ค่าตัวแปรคาร์คินัลจะมีขนาดตั้งแต่ 0 ถึง 4294967295 หรือ 32 บิต ดังนั้นถ้ามีการส่งข้อมูลที่มีขนาดเกิน 32 บิต จะปรากฏผลดังรูป

### ข้อมูลที่ทำการส่ง

ตำแหน่งหน่วยความจำ	ค่าที่ป้อน	ค่าฐาน 10	ค่าฐาน 2
9000H	30	48	00110000
9001H	31	49	00110001
9002H	32	50	00110010
9003H	33	51	00110011
9004H	34	52	00110100

### ข้อมูลที่รับเข้ามาแสดงผล

เมื่อทำการส่งข้อมูลตั้งแต่ตำแหน่ง 9000H – 9004H ข้อมูลที่รับเข้ามาควรจะเป็น

00110000 00110001 00110010 00110011 00110100

แต่ข้อมูลที่ได้รับเข้ามากลับเป็น

00110011 00110010 00110001 00110000 00110100

จะเห็นว่าคอมพิวเตอร์จะทำการรับข้อมูลได้ที่ละ 32 บิต ( 4 ไบต์ ) และข้อมูลที่รับเข้ามาจะเรียงจากหลังไปหน้า เป็นลักษณะนี้ตลอด

จากปัญหาดังกล่าวทำให้ผู้จัดทำเลือกใช้การรับข้อมูลที่เป็นสตรีง เนื่องจากไม่มีข้อจำกัดในเรื่องของขนาดข้อมูล

#### 4.2 ส่วนของโปรแกรมบนบอร์ดหลัก

สำหรับส่วนของโปรแกรมบนบอร์ดหลักนี้ จะแบ่งออกได้เป็น 3 ส่วนหลัก ๆ คือ ส่วนของการติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ส่วนบุคคล ,ส่วนของการทำงานกับฐานข้อมูล และส่วนที่เป็นฟังก์ชันการทำงานพื้นฐานของบอร์ดหลัก ซึ่งสำหรับส่วนที่ 3 นี้ จะเป็นส่วนที่ทางกลุ่มฮาร์ดแวร์ได้เป็นผู้จัดสร้างขึ้น เพื่อสำหรับการนำมาเรียกใช้ในการติดต่อจัดการกับข้อมูลในฐานข้อมูลได้สะดวกยิ่งขึ้น

##### 4.2.1 ส่วนของการติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ส่วนบุคคล

จะเป็นส่วนที่จัดการทำการส่งข้อมูลให้แก่เครื่องคอมพิวเตอร์ส่วนบุคคล และรับข้อมูลจากเครื่องคอมพิวเตอร์ส่วนบุคคล โดยข้อมูลที่ทำการจัดส่งไปยังเครื่องคอมพิวเตอร์ส่วนบุคคลนั้น ได้แก่ข้อมูลเกี่ยวกับบันทึกการเข้ามาใช้งานของผู้ใช้ระบบ ส่วนข้อมูลที่ทำการรับมาจากเครื่องคอมพิวเตอร์ส่วนบุคคลนั้น จะเป็นข้อมูลที่ประกอบไปด้วยรายละเอียดของผู้ใช้ ซึ่งมีรายละเอียดดังในฐานข้อมูลที่ได้อีกกล่าวมาแล้วข้างต้น

สำหรับส่วนของโปรแกรมในส่วนนี้ จะมีหน้าที่ในการสร้างรูปแบบของข้อมูลที่ได้มีการรับส่ง และตรวจเช็คความถูกต้องของข้อมูลที่ได้ทำการรับส่ง หากมีการผิดพลาดของข้อมูลเกิดขึ้นในการรับส่งข้อมูลก็จะมีแจ้งเตือนแก่ผู้ใช้เครื่องคอมพิวเตอร์ส่วนบุคคลถึงความผิดพลาดที่เกิดขึ้นเพื่อให้ผู้ใช้เครื่องคอมพิวเตอร์ส่วนบุคคลได้ทำการติดต่อกลับมาใหม่

ในการส่งข้อมูลการเข้ามาใช้งานของผู้ใช้ระบบจากบอร์ดหลัก ให้แก่เครื่องคอมพิวเตอร์ส่วนบุคคล มีขั้นตอนคือ เมื่อทางเครื่องคอมพิวเตอร์ส่วนบุคคลทำการติดต่อมายังบอร์ดหลัก และทางบอร์ดหลักได้ทำการตรวจเช็คและพบว่า เป็นการขอข้อมูลการเข้ามาใช้งานของผู้ใช้ระบบ ทางบอร์ดหลักจะทำการจัดส่งข้อมูลในฐานข้อมูลไปให้แก่เครื่องคอมพิวเตอร์ส่วนบุคคล โดยมีการเพิ่มเติมรูปแบบของข้อมูลให้เป็นไปตามข้อตกลงในการสื่อสารที่ได้กำหนดไว้

เมื่อได้ทำการทดลองส่งข้อมูลโดยใช้ตัวโปรแกรมสำเร็จรูปที่ใช้ในการติดต่อสื่อสารทางพอร์ทอนุกรมของคอมพิวเตอร์ เป็นตัวรับค่าจากบอร์ดหลักเพื่อตรวจเช็คความถูกต้องของรูปแบบในการจัดส่งและข้อมูลที่ส่งมา (โดยที่ข้อมูลในฐานข้อมูลที่ได้ทำการจัดส่งมานั้น เป็นข้อมูลที่ได้ทำการสมมติขึ้นและเก็บไว้อยู่ในฐานข้อมูลตามที่กำหนด)

เมื่อให้ข้อมูลในฐานข้อมูลที่ได้ทำการสมมติและบันทึกไว้ เป็น

โค้ด 3 ไบต์	วันที่ 6 ไบต์	เวลาเข้า 6 ไบต์	เวลาออก 6 ไบต์	หมายเลข คอม 2 ไบต์
102	110143	093212	104231	02
031	110143	095503	104831	06
089	110143	102336	110232	07
114	120143	091220	114301	06



รูปที่ 4.1

แสดงหน้าจอโปรแกรมสำเร็จรูปที่ใช้ในการติดต่อสื่อสารทางพอร์ทอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ทางโปรแกรมทางเครื่องคอมพิวเตอร์ส่วนบุคคลได้รับจะมีข้อมูลดังนี้

004a102#110143#093212#104231#02#{za031#110143#095503#104831#06#-za089#110143#102336#110232#07#-za114#120143#091220#114303#06#-zE

สำหรับการรับข้อมูลผู้ใช้จากเครื่องคอมพิวเตอร์ส่วนบุคคลมาเก็บยังฐานข้อมูลในเครื่องเซิร์ฟเวอร์นั้น เริ่มต้นขึ้นด้วยการที่เครื่องคอมพิวเตอร์ส่วนบุคคลส่งสัญญาณมาที่บอร์ดหลัก และเมื่อบอร์ดหลักได้ทำการตรวจเช็คพบว่าเป็นการขอส่งข้อมูลรายละเอียดผู้ใช้จากเครื่องคอมพิวเตอร์มายังบอร์ดหลักแล้ว เครื่องเซิร์ฟเวอร์ก็จะส่งสัญญาณพร้อมรับกลับไปยังเครื่องคอมพิวเตอร์ส่วนบุคคล เพื่อให้เครื่องคอมพิวเตอร์ส่วนบุคคลทำการส่งข้อมูลตามรูปแบบการรับส่งมายังบอร์ดหลัก เมื่อบอร์ดหลักทำการรับข้อมูลมาเรียบร้อยแล้วก็จะทำการตรวจเช็คความถูกต้องของข้อมูล หากข้อมูลที่ส่งมาถูกต้องก็จะทำการเก็บข้อมูลลงหน่วยความจำภายนอก แต่หากข้อมูลที่ส่งมามีความผิดพลาด ทางบอร์ดหลักก็จะทำการส่งสัญญาณแจ้งกลับไปยังเครื่องคอมพิวเตอร์ส่วนบุคคลเพื่อแสดงถึงความผิดพลาดที่เกิดขึ้น

ในการทดลองรับข้อมูลของบอร์ดหลักนี้ ได้ทำการทดลองกับโปรแกรมสำเร็จรูปในการส่งข้อมูลผ่านพอร์ตสื่อสารอนุกรมเช่นเดียวกัน โดยที่ได้เขียนโปรแกรมทดลองบนบอร์ดหลักเพิ่มเติมจากการตรวจสอบเลข ๆ โดยถ้าหากตรวจเช็คข้อมูลที่ได้รับมาเรียบร้อยแล้วและมีค่าถูกต้องตามต้องการ ก็จะให้บอร์ดหลักทำการส่งข้อมูลที่ได้รับมานั้นกลับไปยังเครื่องคอมพิวเตอร์ส่วนบุคคลเพื่อแสดงผลของข้อมูล และทำการตรวจเช็คด้วยสายตาอีกครั้งหนึ่ง ว่าการรับส่งข้อมูลนั้นไม่มีการผิดพลาด

#### 4.2.2 ส่วนของการทำงานกับฐานข้อมูล

ในการทำงานกับฐานข้อมูล แบ่งเป็น การอ่านข้อมูลจากฐานข้อมูลเพื่อส่ง ไปให้แก่เครื่องคอมพิวเตอร์ส่วนบุคคล และการรับข้อมูลจากเครื่องคอมพิวเตอร์ส่วนบุคคลมาเก็บบันทึกไว้ในฐานข้อมูล

สำหรับในการรับข้อมูลจากเครื่องคอมพิวเตอร์ส่วนบุคคลมาเก็บบันทึกไว้ในฐานข้อมูลนั้น เมื่อได้ทำการรับข้อมูลจากเครื่องคอมพิวเตอร์ส่วนบุคคลมาและทำการตรวจเช็คความถูกต้องเรียบร้อยแล้ว ก็จะมีการบันทึกข้อมูลเหล่านั้นลงบนฐานข้อมูลตามตำแหน่งของข้อมูลที่ได้มีการออกแบบไว้

การทดลองเพื่อตรวจสอบความถูกต้องของการเก็บบันทึกข้อมูล ทำโดย เมื่อมีการส่งค่าข้อมูลมาจากเครื่องคอมพิวเตอร์ส่วนบุคคล และทำการตรวจเช็คความถูกต้อง พร้อมทั้งบันทึก

ข้อมูลลงบนฐานข้อมูลเรียบร้อยแล้ว ก็ได้ทำการส่งข้อมูลจากตำแหน่งที่ทำการบันทึกไว้นั้น ไปยังเครื่องคอมพิวเตอร์ส่วนบุคคลและได้ใช้โปรแกรมสำเร็จรูปในการติดต่อสื่อสารผ่านทางพอร์ทอนุกรมเป็นตัวรับค่าและแสดงค่า แล้วทำการตรวจเช็คความถูกต้องด้วยสายตา ว่าค่าที่ได้ทำการบันทึกลงบนฐานข้อมูลนั้นไม่มีความผิดพลาดเกิดขึ้น

#### 4.3 การทดลองระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลและบอร์ดหลัก

เมื่อได้ทำการตรวจเช็คโดยการแยกส่วนระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคล และบอร์ดหลักเรียบร้อยแล้ว ก็ได้มีการนำเครื่องคอมพิวเตอร์ส่วนบุคคลและบอร์ดหลักมาต่อใช้งานร่วมกันตามฟังก์ชันต่าง ๆ ที่ได้สร้างขึ้นมา ผลปรากฏว่าในครั้งแรกที่ได้มีการทดลอง ในเรื่องของการรับส่งข้อมูลไม่เกิดปัญหาใด ๆ แต่ในการเก็บข้อมูลเข้าฐานข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคล ไม่สามารถทำได้ เนื่องจากในการสมมติข้อมูลในฐานข้อมูลบนบอร์ดหลักนั้น ไม่เข้า (Match) กับสมมติข้อมูลในฐานข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคล ตัวอย่างเช่น ในส่วนของโค้ดของผู้ใช้งาน ข้อมูลที่ได้สมมติไว้ในหน่วยความจำของบอร์ดหลัก ไม่มีปรากฏอยู่ในฐานข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคล ดังนั้น จึงไม่สามารถทำการเก็บข้อมูลที่รับมาและตรวจเช็คเรียบร้อยแล้วว่าถูกต้องลงฐานข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคลได้ จึงได้มีการแก้ไขโดยการสมมติข้อมูลในหน่วยความจำของบอร์ดหลักใหม่ให้เข้ากันกับข้อมูลในฐานข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคล และหลังการปรับปรุงการทดลอง ก็สามารถพบว่า การทำงานระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลและบอร์ดหลักสามารถทำงานได้ตรงตามต้องการ

## บทที่ 5

### บทวิจารณ์และสรุป

ปฏิญานิพนธ์ฉบับนี้ เกี่ยวกับระบบควบคุมการเข้าออกด้วยบัตรแม่เหล็ก ( Access Control Using Magnetic Card ) โดยเป็นส่วนหนึ่งของโปรแกรมเท่านั้น ซึ่งการประยุกต์ใช้งานจริงในปัจจุบัน ส่วนโปรแกรมจะสามารถทำการพัฒนาได้ในหลายรูปแบบ เพื่อให้เหมาะกับระบบนั้นๆมากที่สุด ดังนั้นในวิทยานิพนธ์ฉบับนี้จะขกกล่าวถึง ปัญหาและแนวทางแก้ไข และ แนวทางในการพัฒนา เพื่อให้ง่ายต่อการพัฒนาโปรแกรมนี้ในอนาคต

#### 5.1 ปัญหาและแนวทางแก้ไข

5.1.1 การรันแอปพลิเคชันที่เขียนด้วยเคลไฟ จะสามารถทำงานได้บนเครื่องที่มีส่วนประกอบดังต่อไปนี้ เป็นอย่างน้อย

- Window 95/98 หรือ Window NT
- CPU 80486 หรือ Pentium หรือสูงกว่า
- หน่วยความจำ 8 M ( แนะนำ 16 MB สำหรับ Windows 95/98 และ 32 MB Windows NT )
- จอภาพ VGA หรือจอภาพที่มีความละเอียดสูงกว่า
- Mouse
- เนื้อที่บนฮาร์ดดิสก์ที่ต้องการขึ้นอยู่กับรูปแบบของการติดตั้ง และเวอร์ชันของเคลไฟ โดยสามารถอธิบายตามตารางได้ดังนี้

Version	Full Install	Compact Install
Standard	93 MB	40 MB
Professional	121 MB	48 MB
Client / Server	154 MB	58 MB

ซึ่งในขณะที่ทำการติดตั้ง จะใช้เนื้อที่หน่วยความจำอีก 10 MB

ซึ่งจะเกิดปัญหาว่า คอมพิวเตอร์ของทางภาควิชาอาจจะไม่สามารถใช้งาน โปรแกรมนี้ได้

5.1.2 การนำแอปพลิเคชันที่เขียนด้วยเคลฟ มาทำงานบนเครื่องคอมพิวเตอร์ที่ไม่มีโปรแกรมเคลฟนั้น สามารถทำได้ แต่ถ้าโปรแกรมมีส่วนของฐานข้อมูลมาเกี่ยวข้อง ต้องหาโปรแกรมช่วยมารองรับการทำงานเกี่ยวกับฐานข้อมูลด้วย

5.1.3 ปัญหาทางโปรแกรม เกี่ยวกับการติดต่อสื่อสารทางพอร์ทอนุกรม เนื่องจากหนังสือคู่มือการใช้งานเคลฟ มักไม่มีการแนะนำเกี่ยวกับการติดต่อสื่อสารทางพอร์ท ทำให้ยากลำบากต่อการศึกษา และการโปรแกรม

แนวทางแก้ไข ต้องทำการไปค้นหาข้อมูลเกี่ยวกับการติดต่อทางอินเทอร์เน็ต

5.1.4 ปัญหาทางด้านฐานข้อมูล เนื่องจากทางคณะผู้จัดทำไม่มีความรู้ทางด้านฐานข้อมูลมาก่อน ไม่ว่าจะเป็นชนิดของฐานข้อมูล หรือรูปแบบต่างๆของฐานข้อมูล

แนวทางแก้ไข ต้องทำการศึกษาเพิ่มเติม

## 5.2 แนวทางการพัฒนา

5.2.1 กรณีที่ระบบขาดไฟเลี้ยง หรือไฟตก เนื่องจากการใช้งานของระบบเกี่ยวข้องกับฐานข้อมูลเป็นจำนวนมาก กรณีอาจสร้างความเสียหายให้แก่ระบบฐานข้อมูลได้ จึงจำเป็นต้องมีระบบไฟสำรองให้แก่ระบบเพื่อป้องกันเหตุการณ์ต่างๆ

5.2.2 ทำการตั้งเวลาเพื่อให้มีการถ่ายโอนข้อมูลโดยอัตโนมัติในเวลาดังกล่าว เพื่อป้องกันในกรณีที่ผู้ควบคุมระบบลืมมาทำการถ่ายโอนข้อมูล ทำให้หน่วยความจำบนบอร์ดหลักเต็ม หรือข้อมูลบนฐานข้อมูลของบอร์ดหลัก ไม่ถูกต้อง

5.2.3 ทำการตั้งเวลาที่จะอนุญาตให้มีการใช้งานได้ (Time Zone) เพื่อป้องกันการมาใช้งานในช่วงเวลาที่ไม่เหมาะสม หรือในวันหยุดต่างๆ

## กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์เกียรติวรรณ ทรงสัจย์ ที่ให้คำปรึกษา และให้การสนับสนุนในด้านต่างๆด้วยดีมาตลอด และขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมระบบควบคุมทุกท่าน ที่ให้การประสิทธิ์ประสาทวิชา ตลอดจนความห่วงใยเสมอมา

ขอขอบคุณ พี่พงศธร มณีวัฒนา สำหรับบอร์ด Jazz , พี่นวเทพ ชนาพรรณ และพี่ฐานานามประดิษฐ์ สำหรับเครื่อง Pro 100 ที่ทำให้ project นี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณ พี่ธีรวัฒน์ วงศ์วิณะ สำหรับคำปรึกษาต่างๆ และโปรแกรม ForeHelp

ขอบคุณเพื่อนๆภาควิชาวิศวกรรมระบบควบคุมทุกคน สำหรับความช่วยเหลือและกำลังใจต่างๆ

ท้ายที่สุดขอกราบขอบพระคุณคุณพ่อ คุณแม่ ที่ทำให้มีวันนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

1. กนก กุศุมลย์นุกูล , “คู่มือการเขียนโปรแกรม Delphi 4” , บริษัทซัคเซสมีเดียจำกัด , 422 หน้า , 2542.
2. จรณิต แก้วกั้งวาล , “การออกแบบและจัดการฐานข้อมูล” , ซีเอ็ดยูเคชั่น , 214 หน้า , 2536.
3. จิรศักดิ์ เหลืองอุไร , “คัมภีร์การใช้งาน การสื่อสารอนุกรมบน PC” , ซีเอ็ดยูเคชั่น , 372 หน้า , 2538.
4. นุกูล กระจาย , “การเขียนโปรแกรมแบบวิซวลด้วย Delphi 4” , ซีเอ็ดยูเคชั่น , 496 หน้า , 2542.
5. ยืน ภู่วรรณ , “การสื่อสารข้อมูลและไมโครคอมพิวเตอร์เน็ตเวิร์ค” , ซีเอ็ดยูเคชั่น , 243 หน้า , 2532.
6. สมศักดิ์ ศรีขจรเกียรติ , “Delphi 4 Step by Step” , บีบีไอไฟล์ พับลิชชิง , 304 หน้า , 2542.
7. ศุนทร วิฑูสุรพจน์ , “การโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051” , ซีเอ็ดยูเคชั่น , 2537.
8. Kent Resdorff , “SAMS Tech Yourself Borland Delphi 4 in 21 Days” , Sams Publishing , 918 p. , 1998.



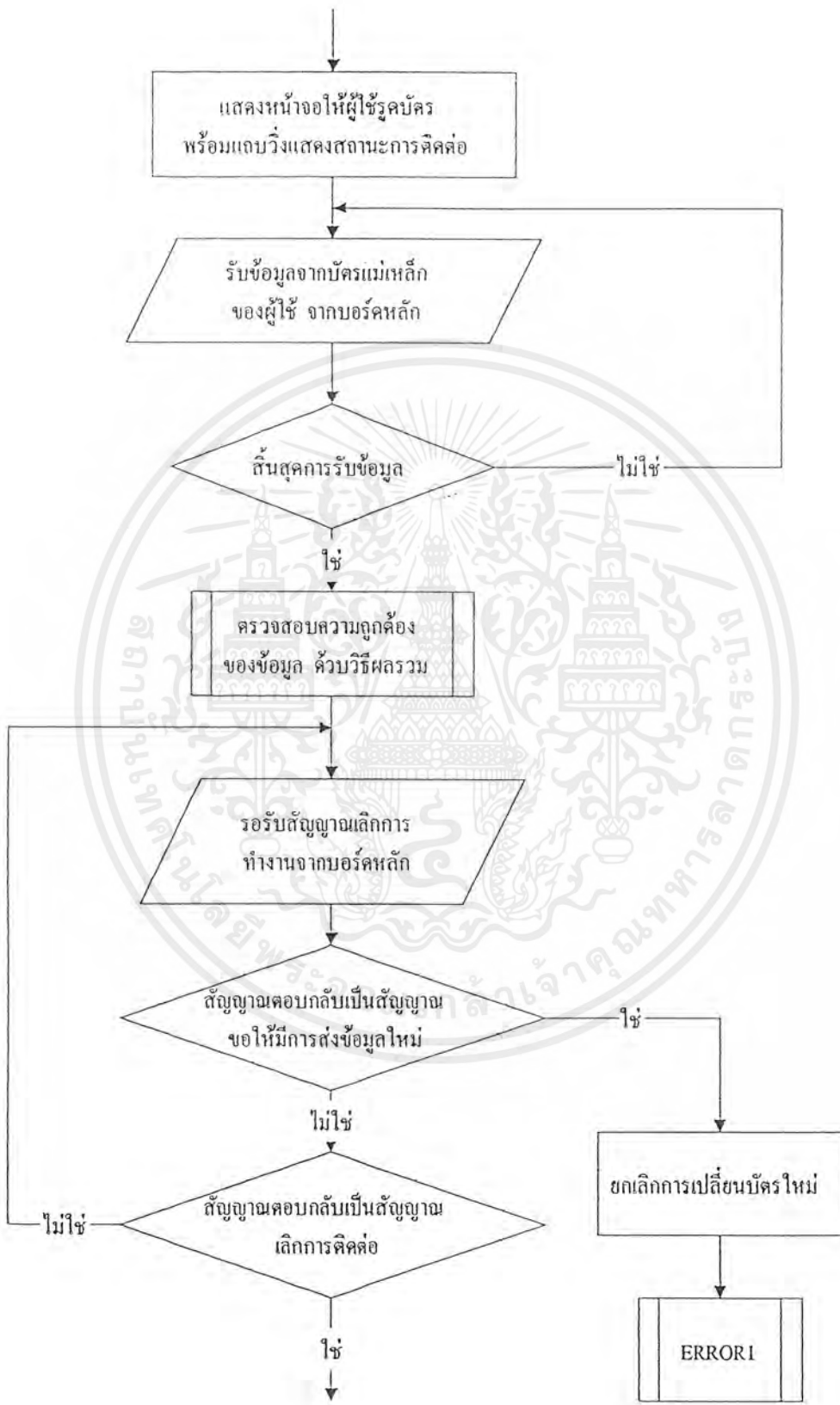
ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# แผนภาพแสดงการเพิ่มข้อมูลผู้ใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



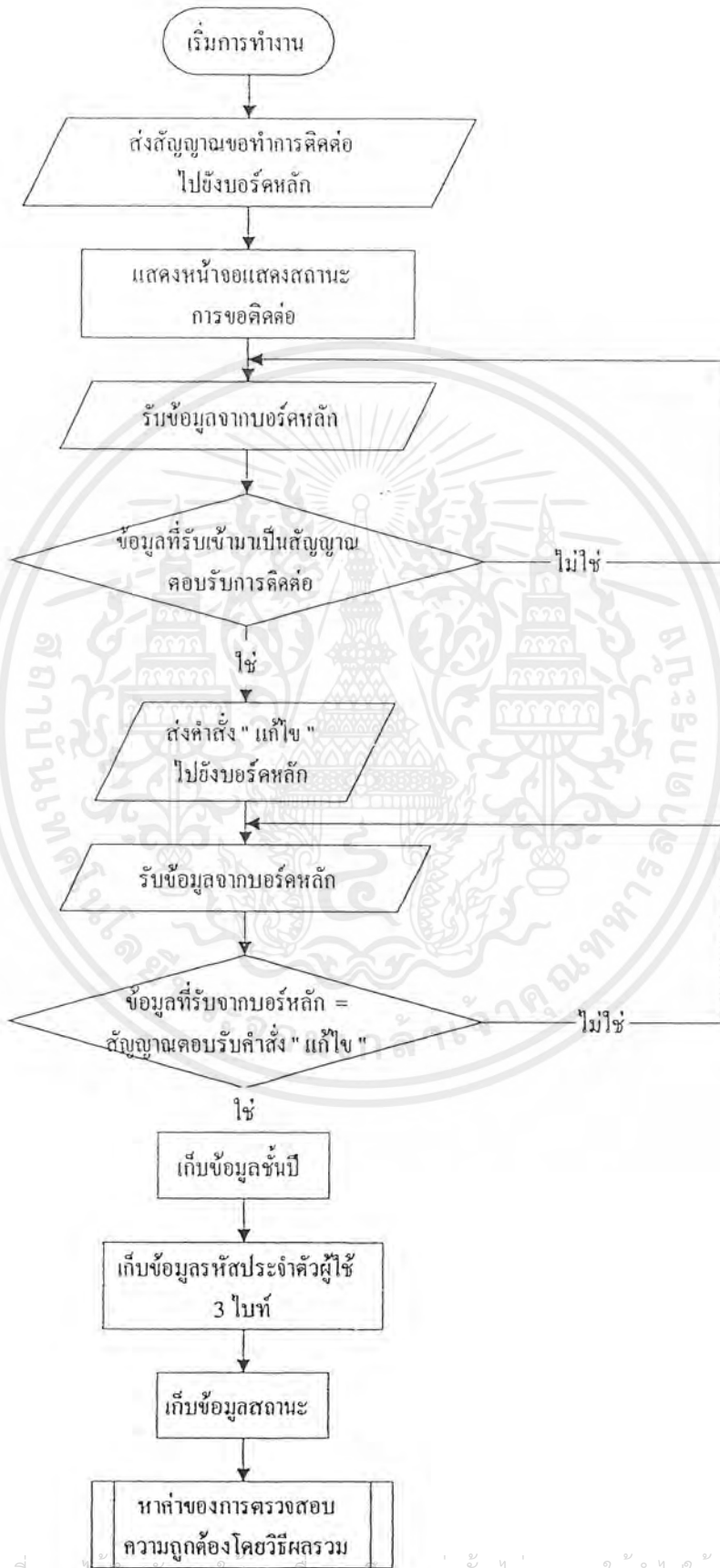
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ERROR

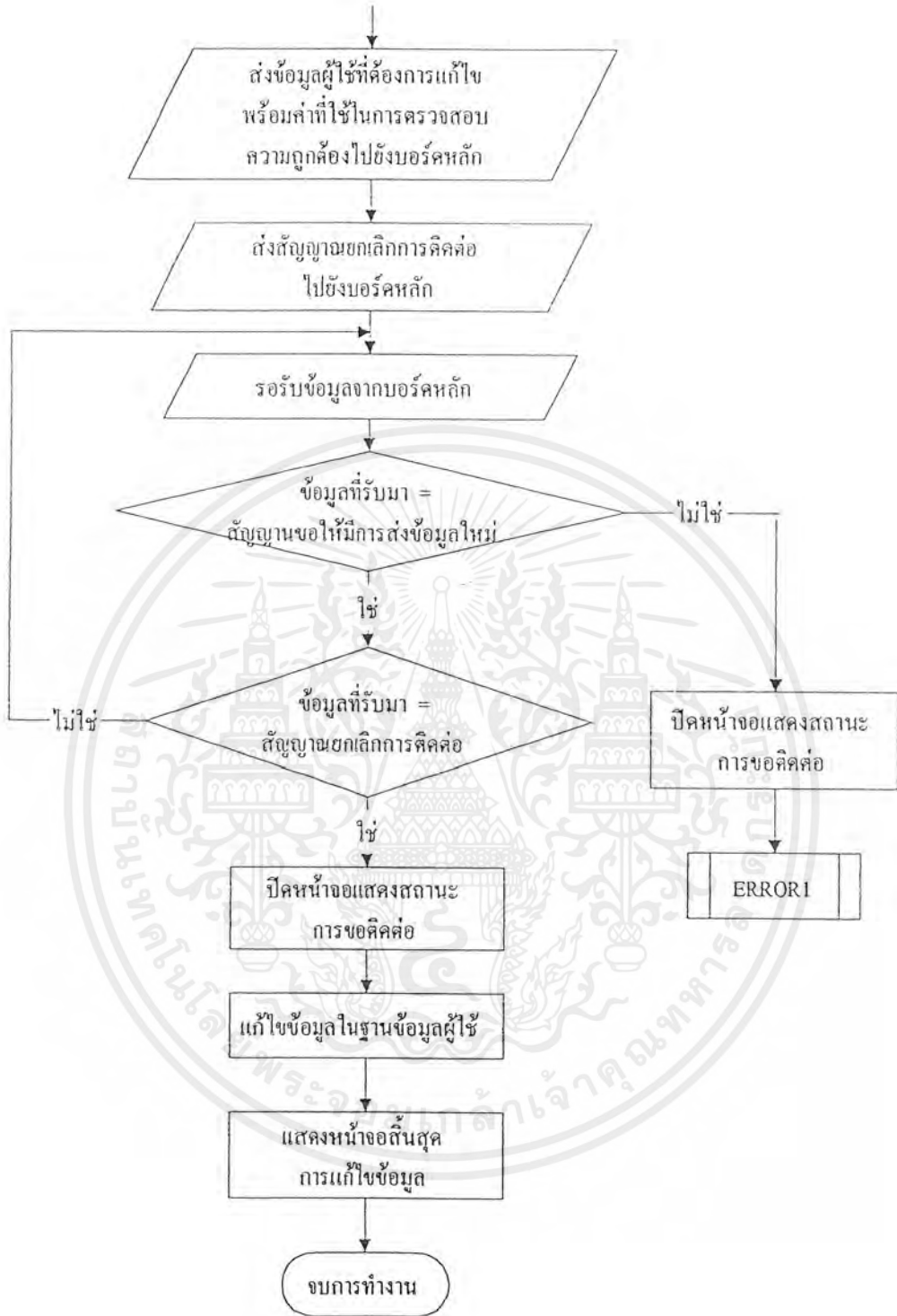


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพการแก้ไขข้อมูล

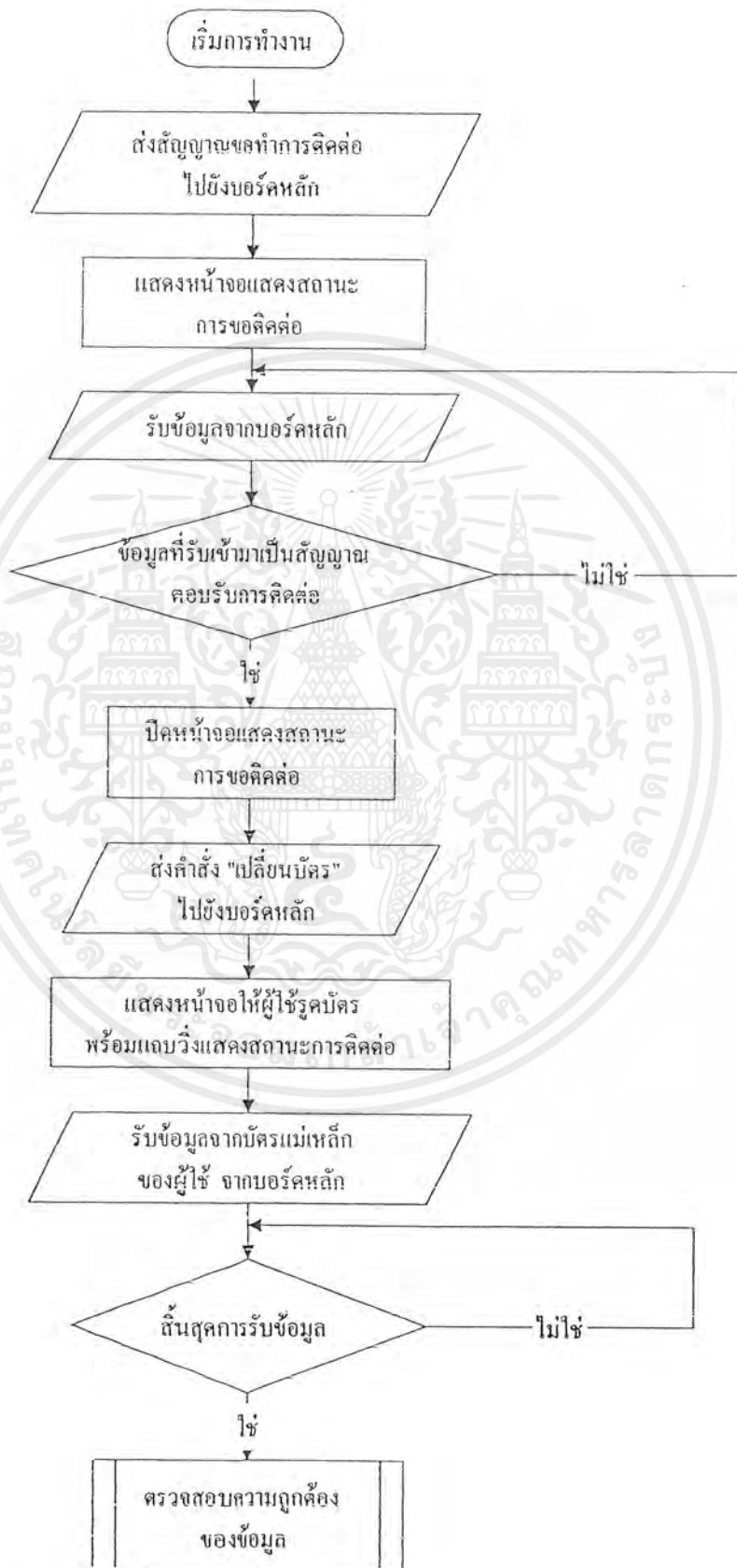


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

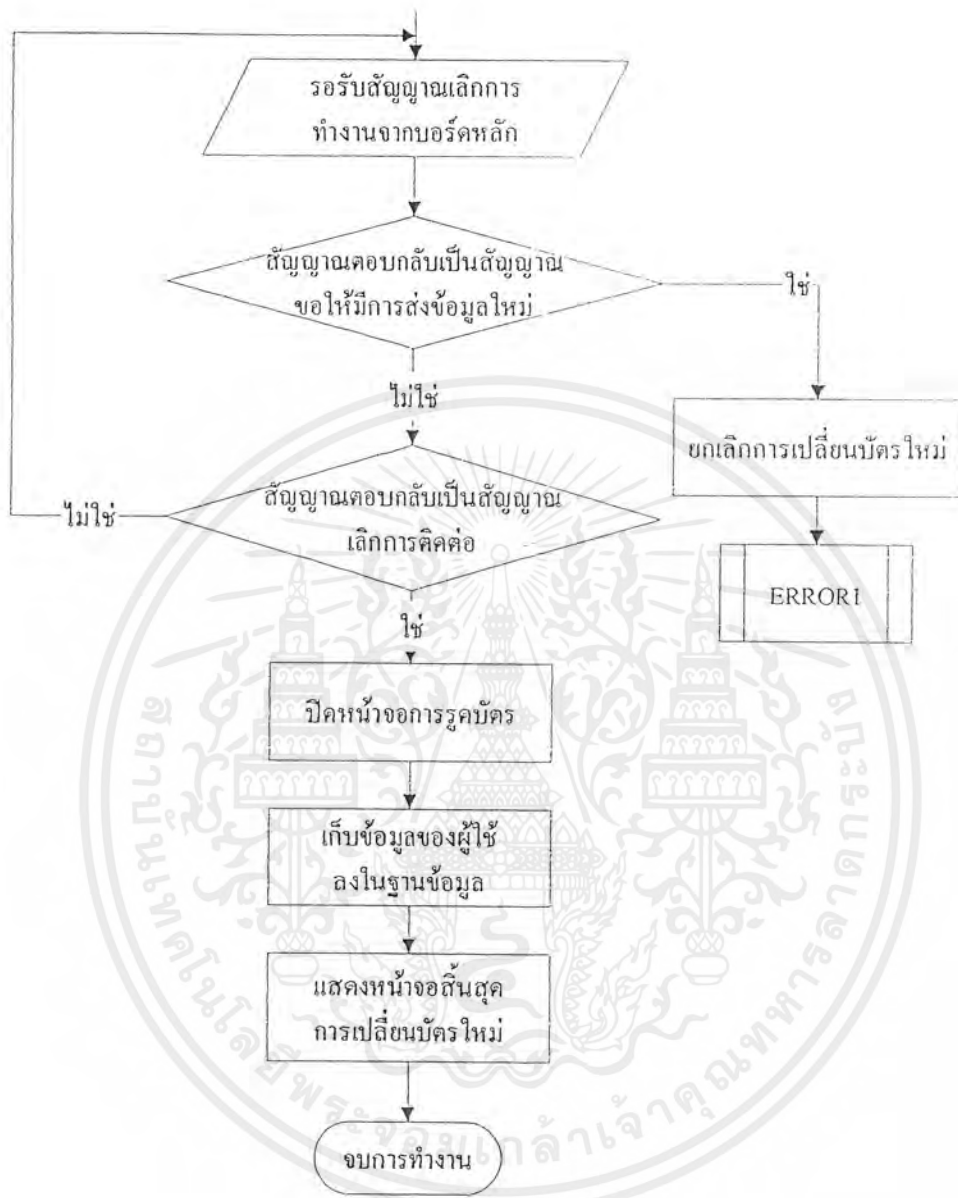


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพแสดงการเปลี่ยนบัตร

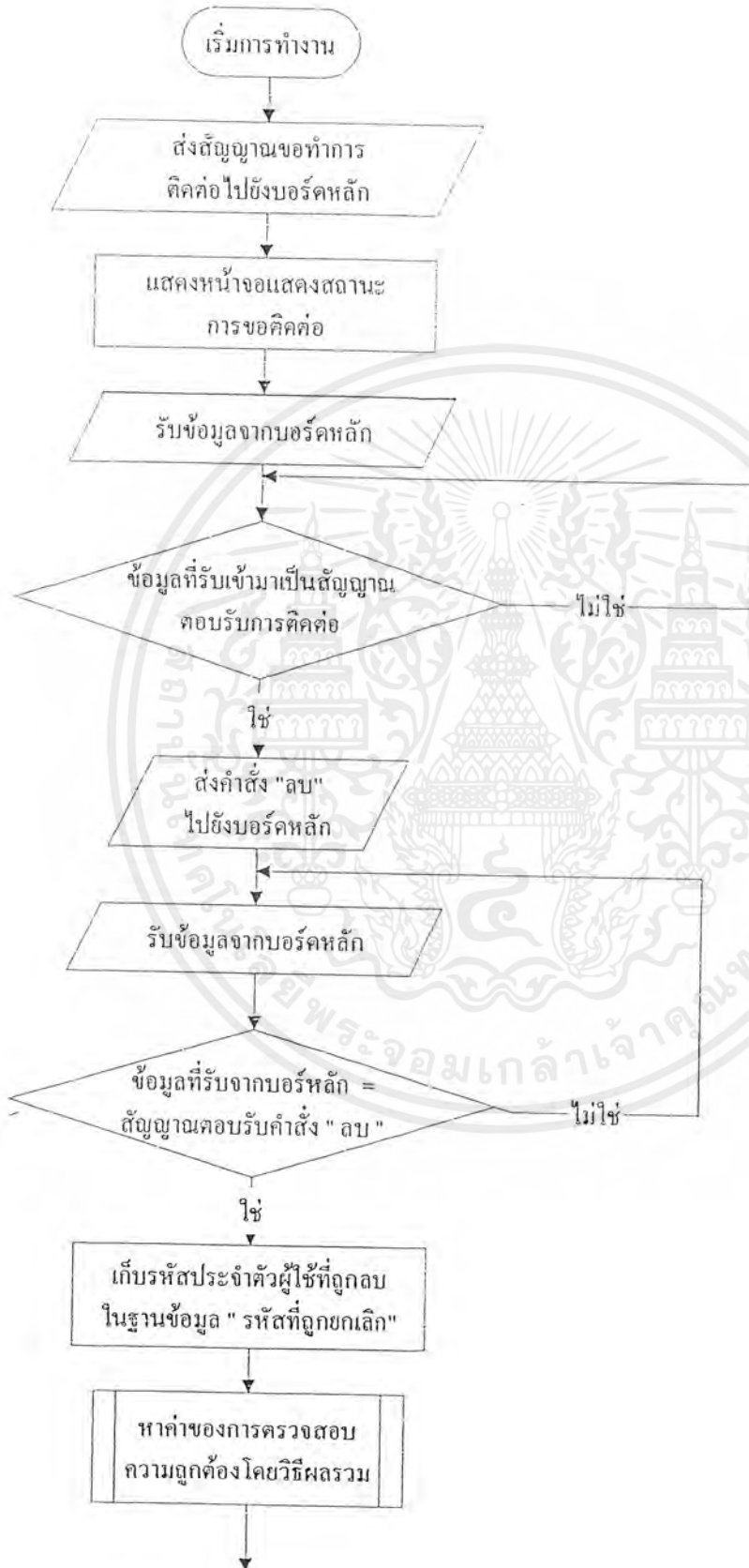


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

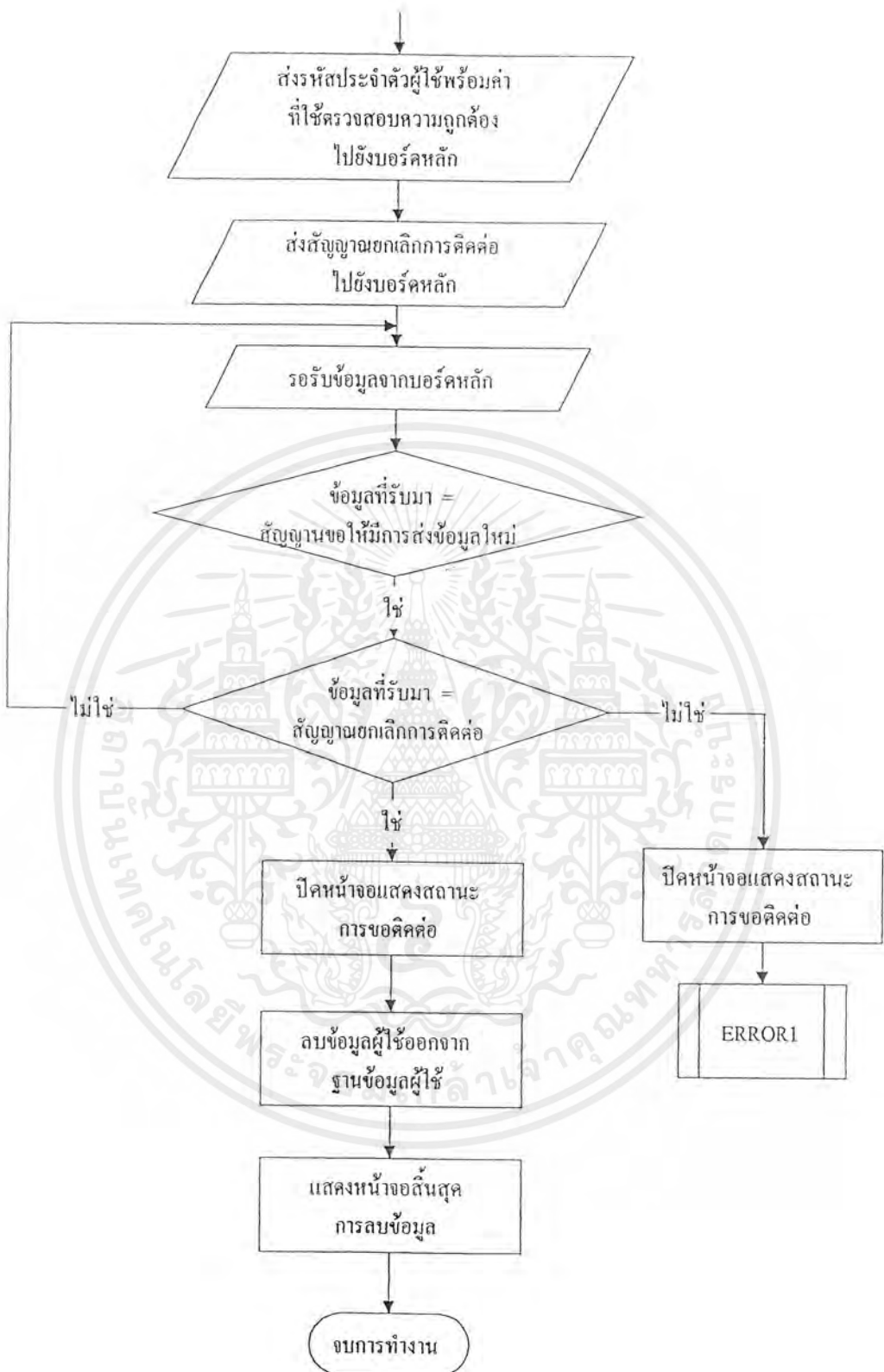


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพการลบข้อมูล

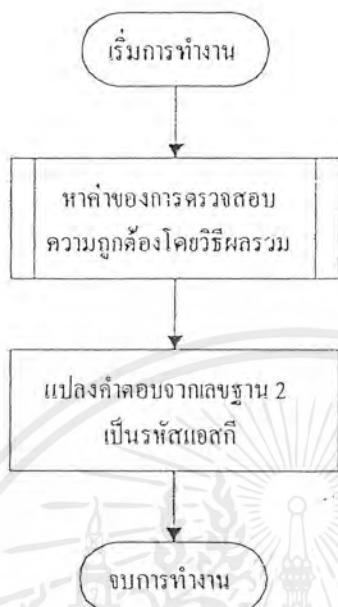


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



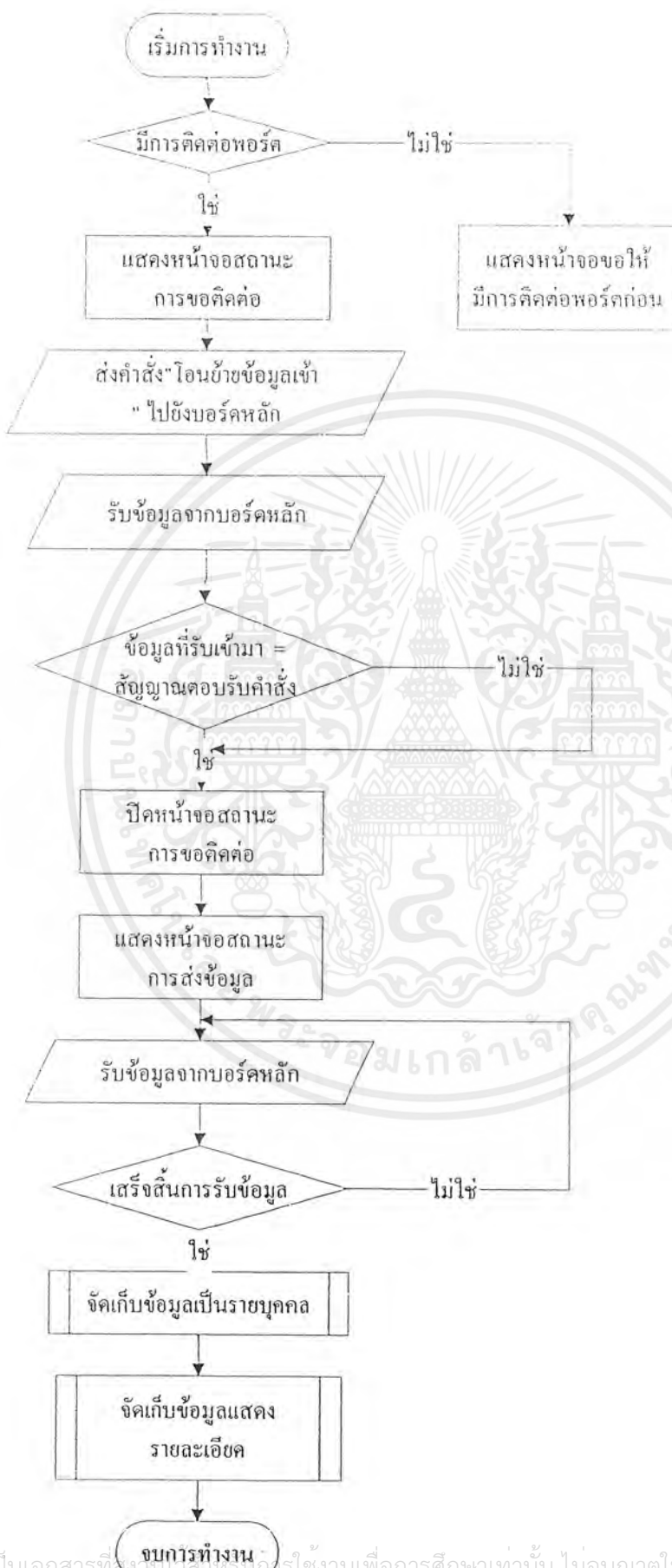
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพการสร้างข้อมูลที่ใช้เป็นค่าตรวจสอบความถูกต้อง



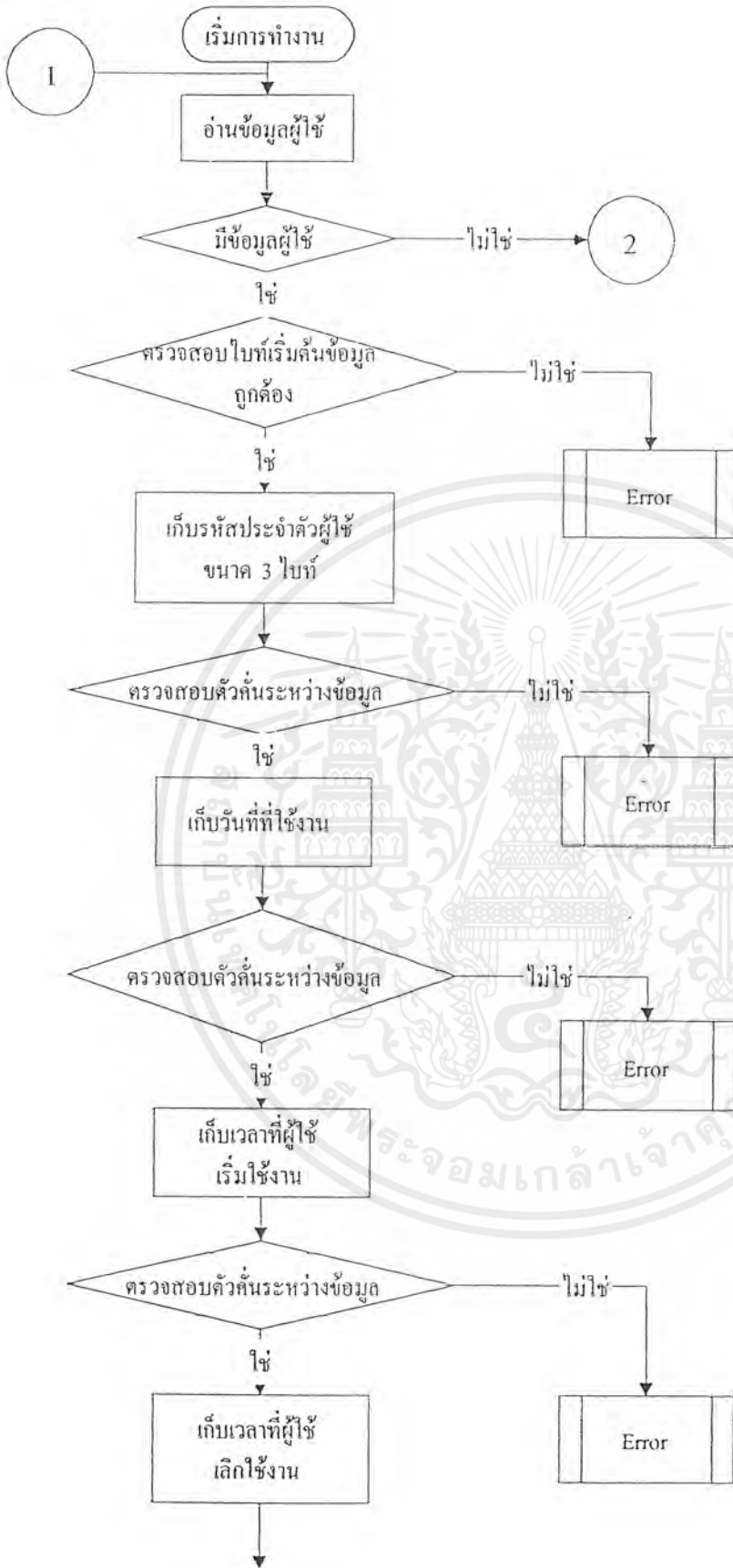
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพแสดงการโอนย้ายข้อมูลเข้า

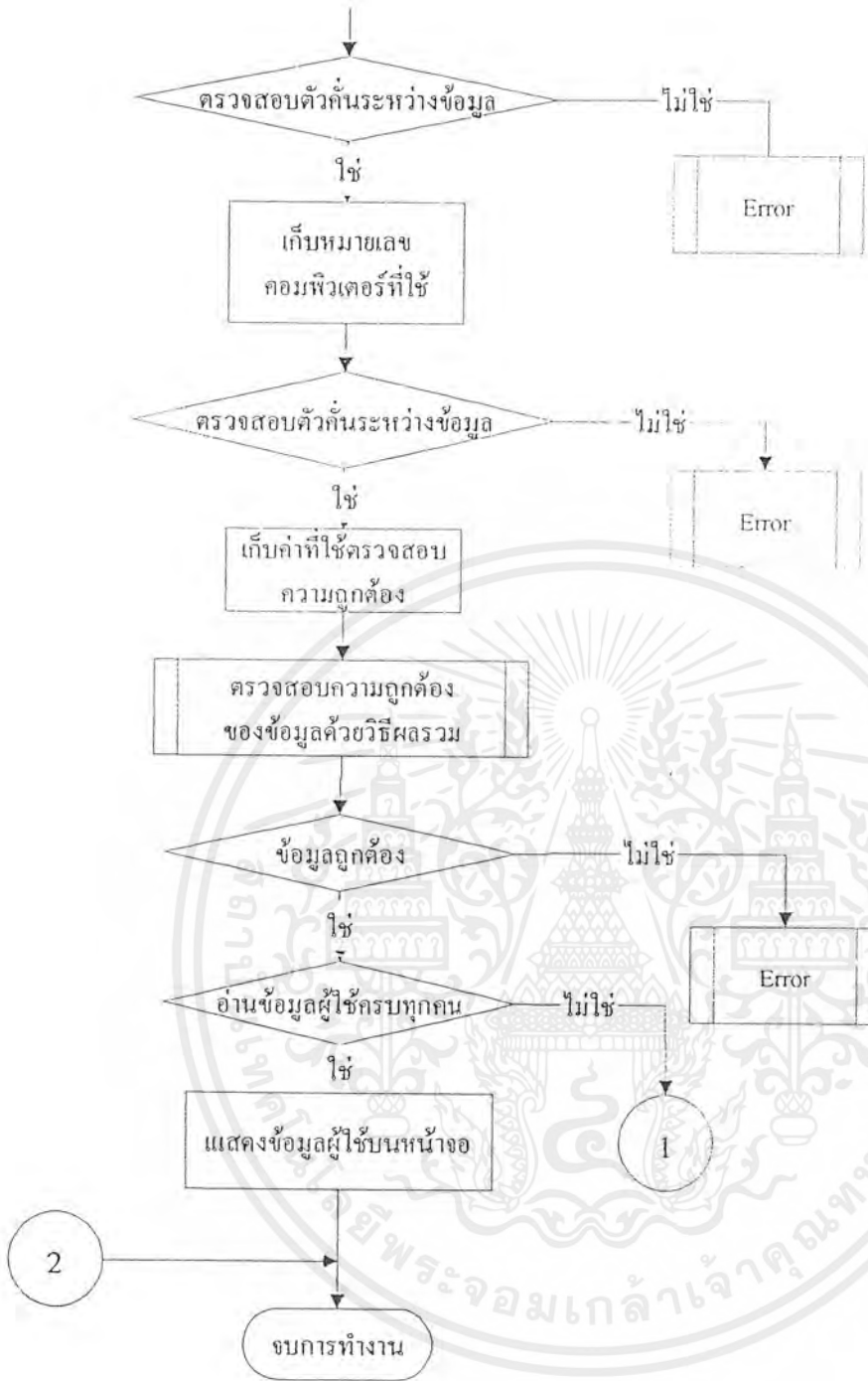


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

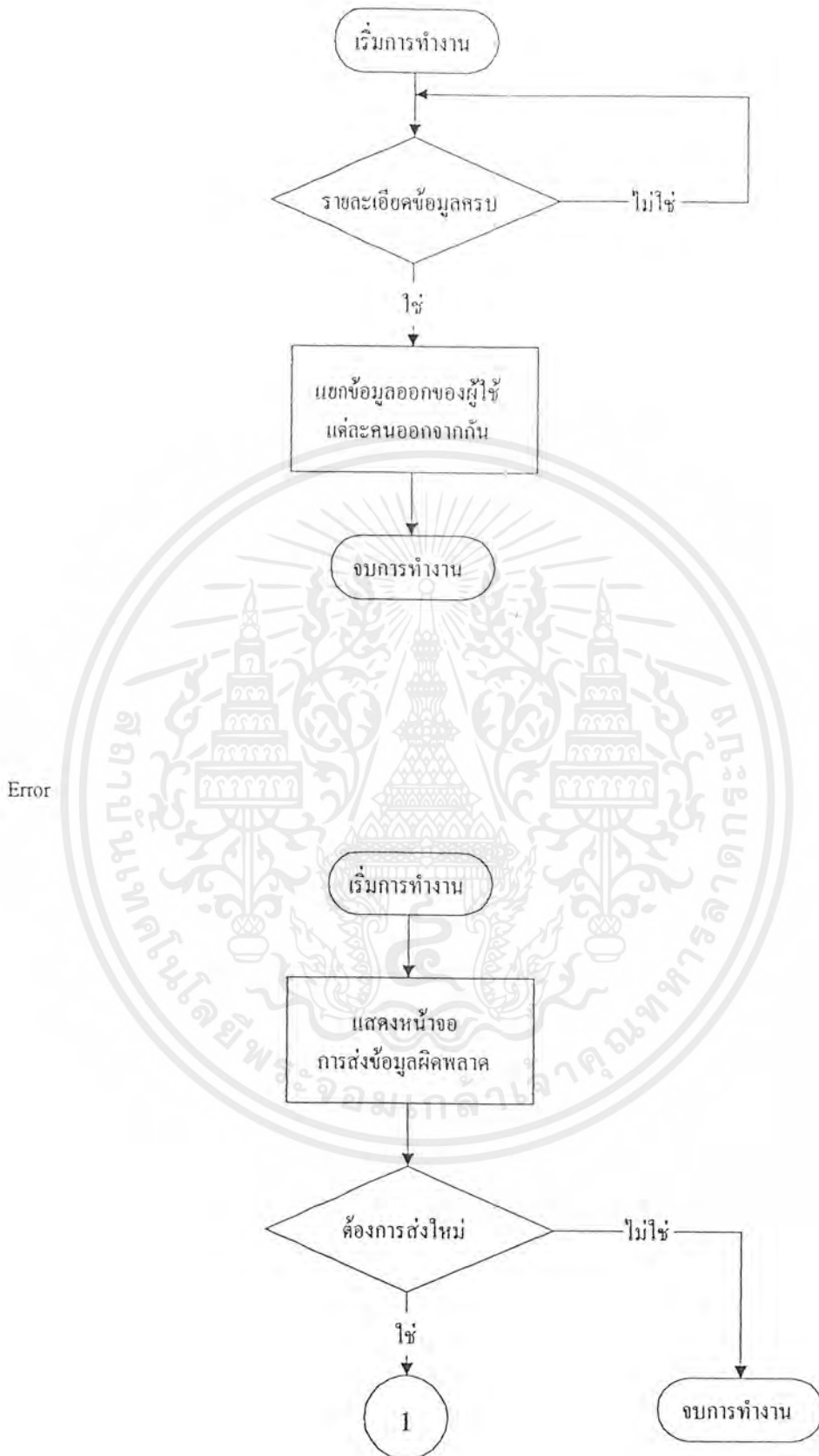
Field



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



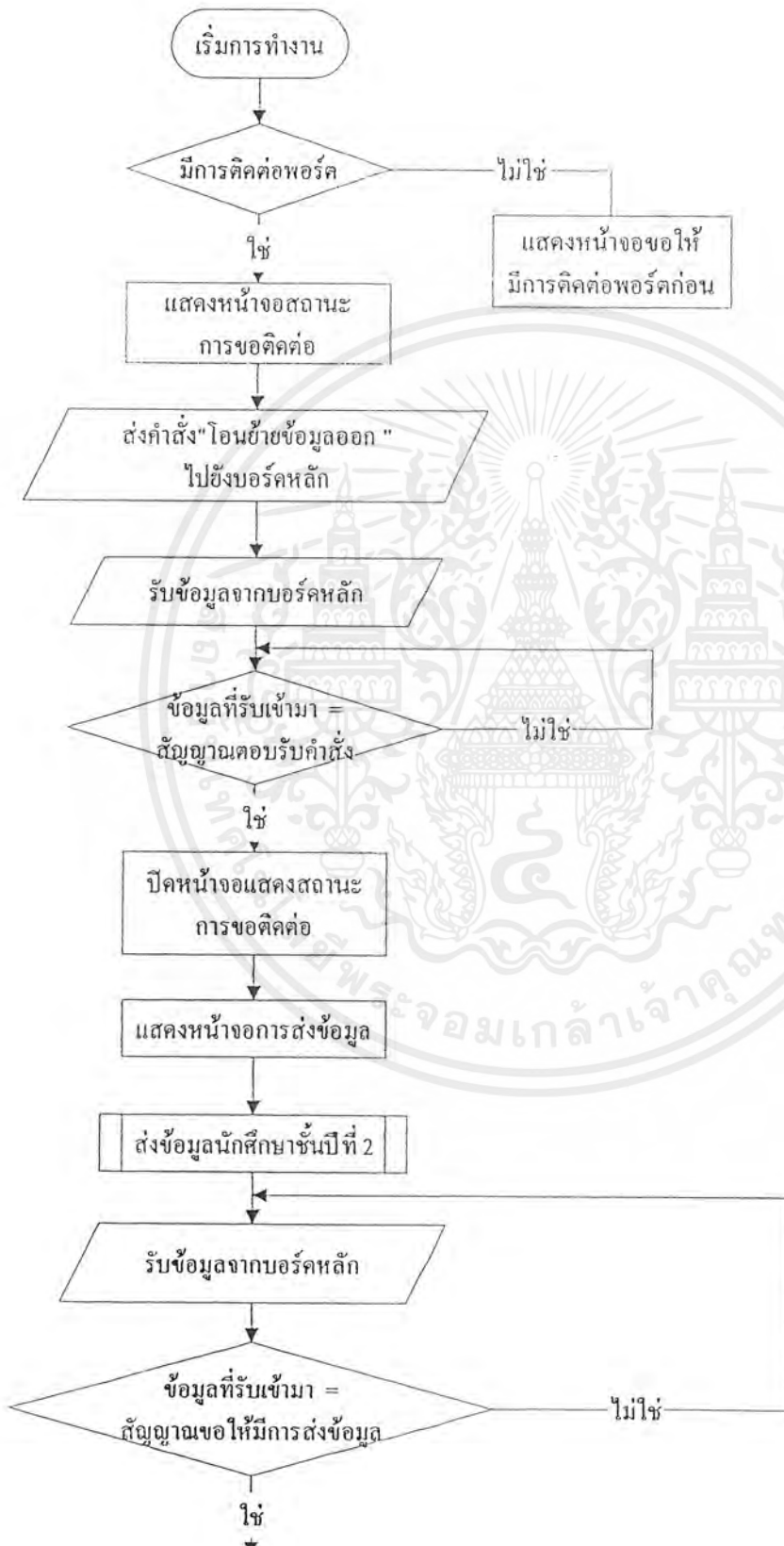
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

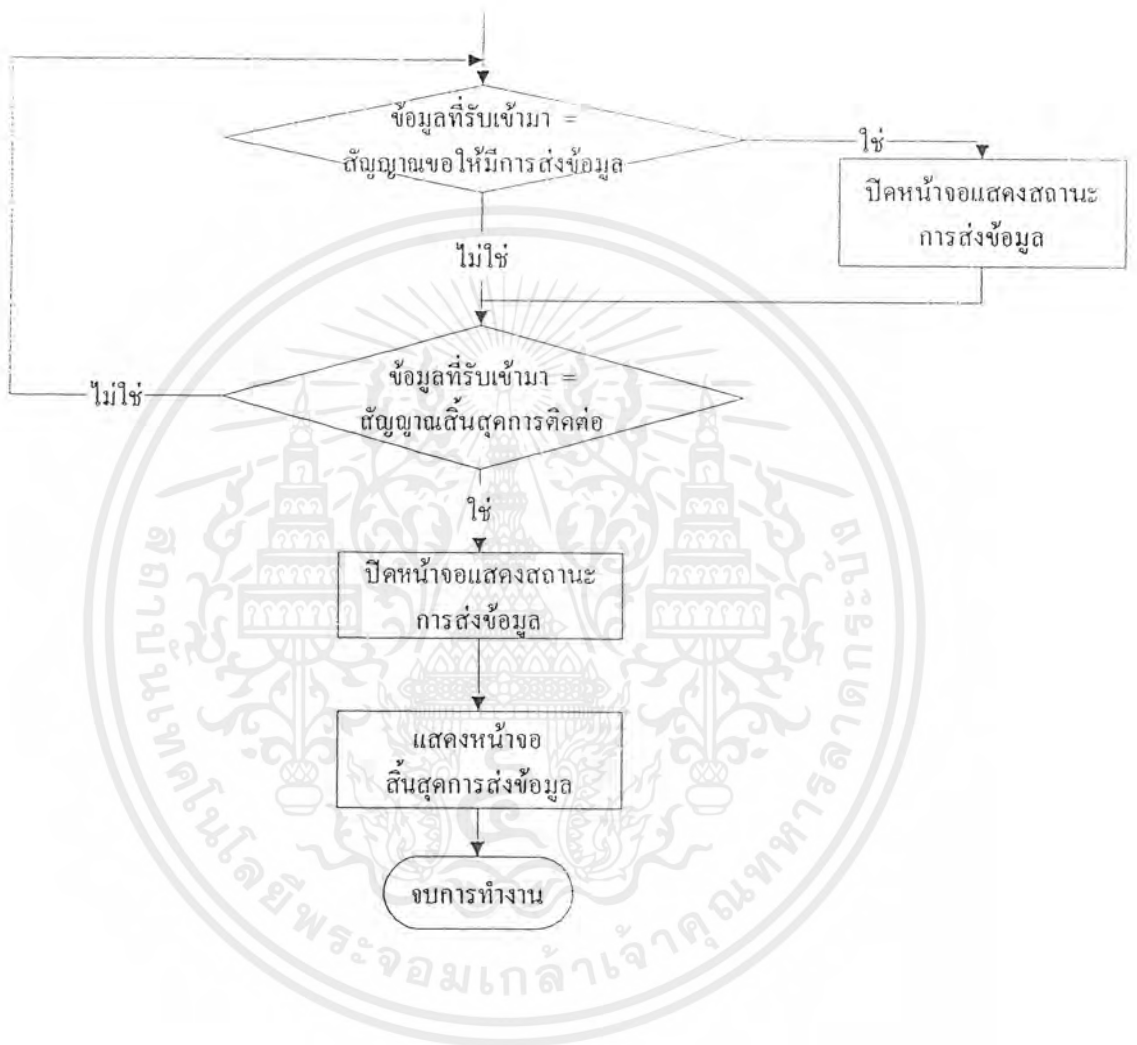
# แผนภาพแสดงการถ่ายโอนข้อมูลออก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit cannotuse_ok;

interface
    procedure SpeedButton3Click(Sender:
    TObject);
    procedure SpeedButton4Click(Sender:
    TObject);
    procedure SpeedButton2Click(Sender:
    TObject);
    procedure SpeedButton1Click(Sender:
    TObject);
private
    { Private declarations }
type
    public
    { Public declarations }
    end;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Table1: TTable;
    Table1Id: TStringField;
    Table1Sex: TStringField;
    Table1Name: TStringField;
    Table1Year: TStringField;
    Table1Reason: TStringField;
    Table1Status: TStringField;
    PrintDialog1: TPrintDialog;
    procedure BitBtn1Click(Sender: TObject);
    implementation
    uses view_ok, main_ok, cannotuse_preview;
    {$R *.DFM}
    procedure TForm7.BitBtn1Click(Sender:
    TObject);
    begin
        form7.Hide;
        form2.show;
    end;
    procedure TForm7.SpeedButton3Click
    (Sender: TObject);
    begin
        form7.Hide;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    form2.show;
end;

procedure TForm7.SpeedButton4Click
(Sender: TObject);
begin
    form7.hide;
    form1.show;
end;

```

```

procedure TForm7.SpeedButton2Click
(Sender: TObject);
begin
    form13.quickrep1.Preview;
end;

procedure TForm7.SpeedButton1Click
(Sender: TObject);
begin
    if printdialog1.execute then
        form13.print;
end;

end.

```

```

unit card1;

interface

uses
    Windows, Messages, SysUtils, Classes,
    Graphics, Controls, Forms, Dialogs,
    StdCtrls, Buttons, ComCtrls, Comms;

type
    TForm10 = class(TForm)
        ProgressBar1: TProgressBar;
        Label1: TLabel;
        BitBtn1: TBitBtn;
        procedure BitBtn1Click(Sender: TObject);
        procedure FormClose(Sender: TObject; var
            Action: TCloseAction);
        procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
        code : string[3];
    end;

var
    Form10: TForm10;

implementation

uses edit_ok;

{$R *.DFM}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm10.BitBtn1Click(Sender: TObject);
begin
    form10.close;
end;

procedure TForm10.FormClose(Sender: TObject; var Action: TCloseAction):
begin
    progressbar1.position:=0;
    form3.enabled := true;
end;

procedure TForm10.FormShow(Sender: TObject);
begin
    progressbar1.min := 0;
    progressbar1.max := 50000;
    progressbar1.step := 1;
end;

end.

unit data_storage;

interface

uses
    Windows, Messages, SysUtils, Classes,
    Graphics, Controls, Forms, Dialogs,
    StdCtrls, Grids, DBGrids, Db, DBTables;

type
    TForm18 = class(TForm)
        Memo1: TMemo;
        Memo2: TMemo;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        DataSource1: TDataSource;
        Table1: TTable;
        DBGrid1: TDBGrid;
        Button1: TButton;
        Table1Code: TStringField;
    procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form18: TForm18;

implementation

uses DownLoad;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{SR *.DFM}
```

```
procedure TForm18.Button1Click(Sender:
```

```
TObject);
```

```
begin
```

```
    form18.close;
```

```
    form20.show;
```

```
end;
```

```
end.
```

```
unit DownLoad;
```

```
interface
```

```
uses
```

```
    Windows, Messages, SysUtils, Classes,
```

```
    Graphics, Controls, Forms, Dialogs,
```

```
    Db, DBTables, Comms, StdCtrls, Buttons,
```

```
    ComCtrls, Grids, DBGrids;
```

```
type
```

```
    TForm20 = class(TForm)
```

```
        PageControl1: TPageControl;
```

```
        GroupBox1: TGroupBox;
```

```
        SpeedButton1: TSpeedButton;
```

```
        SpeedButton2: TSpeedButton;
```

```
        SpeedButton3: TSpeedButton;
```

```
        SpeedButton4: TSpeedButton;
```

```
        Label1: TLabel;
```

```
        Label2: TLabel;
```

```
        Label3: TLabel;
```

```
        Label4: TLabel;
```

```
        ComPort1: TComPort;
```

```
        DataSource1: TDataSource;
```

```
        Table1: TTable;
```

```
        DBGrid1: TDBGrid;
```

```
        Table1Id: TStringField;
```

```
        Table1Sex: TStringField;
```

```
        Table1Name: TStringField;
```

```
        Table1Year: TStringField;
```

```
        Table1Status: TStringField;
```

```
        Table1Reason: TStringField;
```

```
        Table1Code: TStringField;
```

```
        StaticText1: TStaticText;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Table1Mag: TStringField;
DataSource2: TDataSource;
Table2: TTable;
Memo1: TMemo;
procedure Initial;
procedure ComPort1Close(Sender:
TObject);
procedure SpeedButton1Click(Sender:
TObject);
procedure ComPort1Open(Sender:
TObject);
procedure ComPort1RxChar(Sender:
TObject; InQue: Integer);
procedure SpeedButton4Click(Sender:
TObject);
procedure Convert( x : integer ; y : integer;
var num : integer);
procedure ConvertAscii ( x : integer; var ch
: char);
procedure Count ( var x : integer);
procedure MakeData ( i : integer );
procedure FindStatus( var status2 :
integer);
procedure MakeData2 ;
procedure SpeedButton2Click(Sender:
TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form20: TForm20;
connect,send : boolean;
year : string[8];
tyear : integer;
implementation
uses main_ok,
data_storage,UpLoad_Send_Status,UpLoad_
Send_data, edit_ok;
{$R *.DFM}
procedure TForm20.Initial;
begin
form18.memo1.clear;
connect := false;
send := false;
comport1.close;
end;
procedure TForm20.ComPort1Close(Sender:
TObject);
begin
label1.caption := 'ตัดต่อพอร์ต';
end;
procedure TForm20.SpeedButton1Click
(Sender: TObject);
begin
if comport1.connected then
comport1.close
else
comport1.Open;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm20.ComPort1Open(Sender: TObject);
begin
  label1.caption := 'ยกเลิกการติดต่อพอร์ต';
  comport1.WriteString('90',true); // ส่งสัญญาณขอส่งข้อมูล
  if (str = 'R') and connect then // สัญญาณตอบรับการส่งข้อมูลจาก 51
  begin
    send := true;
    form17.close;
    form20.enabled := true;
  end
else
  begin
    if (stop = 'E') and send then
    begin
      form20.enabled := true;
      form19.close;
      messagedlg('การส่งข้อมูลเรียบร้อยแล้ว',mtinformation,[mbOK],0);
    end
    else
    begin
      if (stop = '!') and send then
      begin
        form19.close;
        if messagedlg ('การส่งข้อมูลผิดพลาด ต้องทำการติดต่อใหม่',mterror,[mbOK,mbCancel],0) = mrCancel then
          form20.enabled := true;
        else
          begin
            comport1.writestring
            ('90',true); // ส่งสัญญาณขอติดต่อ 51 ใหม่
          end
        end
      end
    end
  end
end;

procedure TForm20.ComPort1RxChar
(Sender: TObject; InQue: Integer);
const
  free = 'bf';
var
  str,temp : string;
  size : integer;
  stop : char;
begin
  comport1.readstring(str,inque,true);
  temp := 'aa';
  stop := 'a';
  if str <> "" then
  begin
    size := length(str);
    temp := copy(str,size-1,2);
    stop := str[1];
  end;
  if temp = free then // สัญญาณตอบรับการติดต่อจาก 51
  begin
    connect := true;
  end
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    send := false;
    form17.form := 'down';
    form17.show;
end;
end;
else
begin
    if (str = 'N') and send then
    begin
        if tyear <= 6 then
        begin
            form18.memo1.clear;
            tyear := tyear+1;
            makedata(tyear);
            form18.memo1.text :=
form18.memo1.text + '?';
            comport1.writestring
(form18.memo1.text,true);
            memo1.text :=
memo1.text + form18.memo1.text;
        end;
        else
            messagedlg('การส่งข้อ
มูลผิดพลาด',mterror,[mbok],0);
        end;
    end;
end;
end;
end;
end;

procedure TForm20.SpeedButton4Click
(Sender: TObject): // กลับหน้าจอหลัก
begin
    initial;
    form20.close;
    form1.show;
end;

procedure TForm20.Convert ( x : integer; y :
integer ;var num : integer);
var
    str : string;
    size,i : integer;
begin
    str := inttostr(x);
    size := length(str);
    while size < y do
    begin
        str := '0' + str;
        size := size+1;
    end;
    for i := 1 to y do
    begin
        x := strtoint(copy(str,i,1)) + 48; //
convert to decimal
        num := num + x;
    end;
end;

procedure TForm20.ConvertAscii (x : integer;
var ch : char);
var
    ans1,ans2,size,i,j : integer;
    ans : string;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    ans := "";
    while x <> 0 do
        begin
            ans1 := x div 2;
            ans2 := x mod 2;
            ans := intostr(ans2) + ans;
            x := ans1;
        end;
    end;
    size := length(ans);
    if size > 7 then
        ans := copy(ans,size-6,7);
        ans1 := strtoint(ans[7]);
        size := 1;
        for i := 6 downto 1 do
            begin
                ans2 := 1;
                for j := 1 to size do
                    ans2 := (strtoint(ans[j])*2)*ans2;
                ans1 := ans1 + ans2;
                size := size+1;
            end;
        end;
        ch := chr(ans1);
    end;

    procedure TForm20.Count( var x : integer);
    var
        temp : string[8];
    begin
        x := 1;
        table2.next;
        temp := table2['year'];
        while (not table2.eof) and (temp = year) do
            begin
                x := x+1;
                table2.next;
                temp := table2['year'];
            end;
        end;
        size := length(ans);
        if size > 7 then
            ans := copy(ans,size-6,7);
            ans1 := strtoint(ans[7]);
            size := 1;
            for i := 6 downto 1 do
                begin
                    ans2 := 1;
                    for j := 1 to size do
                        ans2 := (strtoint(ans[j])*2)*ans2;
                    ans1 := ans1 + ans2;
                    size := size+1;
                end;
            end;
            ch := chr(ans1);
        end;

        procedure TForm20.MakeData( i : integer );
        var
            result : boolean;
            x : integer;
            temp : string[3];
        begin
            case i of
                2,4 : year := intostr(i);
                5 : year := 'ป.โท';
                6 : year := 'อาจารย์';
            end;
            table2.open;
            table2.indexfieldnames := 'Year';
            table2.setkey;
            table2['year'] := year;
            result := table2.gotokey;
            if result then
                begin
                    count(x);
                    temp := intostr(x);
                    while length(temp) <> 3 do
                        temp := '0'+temp;
                    form18.Memo1.text :=
                        form18.memo1.text + 'Y' + intostr(i) + temp ;
                end;
            end;
        end;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        makedata2:
    end
    else
        form18.memo1.text :=
form18.memo1.text + 'Y' + inttostr(i) + '000';
    end;

procedure TForm20.FindStatus ( var status2 :
integer);
var
    ch : char;
begin
    ch := year[1];
    case ch of
        '2' : status2 := 2;
        '3' : status2 := 3;
        '4' : status2 := 4;
        'ป' : status2 := 5;
        'อ' : status2 := 6;
    end;
end;

procedure TForm20.MakeData2 ;
var
    temp : string[8];
    mag : string[40];
    code : string[3];
    status : string[10];
    j,num,status2,year2 : integer;
    ch : char;
begin
    temp := table2['year'];
    while temp = year do
        begin
            num := 0;
            mag := table2['mag'];
            for j := 1 to 40 do
                begin
                    ch := mag[j];
                    num := num + ord(ch);
                end;
            code := table2['code'];
            convert(strtoint(code),3,num);
            status := table2['status'];
            if status = 'ใช้ไม่ได้' then
                status2 := 0
            else
                status2 := 1;
            num := num + status2+48;
            ConvertAscii(num,ch);
            form18.memo1.text :=
form18.Memo1.text +
'a'+mag+'#'+code+'#'+inttostr
(status2)+'#'+ch+'z';
            table2.next;
            if not table2.eof then
                temp := table2['year']
            else
                temp := '0';
            end;
        end;
    end;

procedure TForm20.SpeedButton2Click
(Sender: TObject); // ส่งข้อมูล
begin
    if send then
        begin
            form18.Memo1.clear;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

form20.enabled := false;
form19.form := 'down';
form19.show;
tyear := 2;
makedata(tyear);
form18.memo1.text :=
form18.memo1.text + '?';

comport1.writestring(form18.memo1.text,true)
;
memo1.text := form18.memo1.text;
{for i := 2 to 6 do
begin
MakeData(i);
end;
form18.memo1.Text := form18.Memo1.text
+ '?';
comport1.writestring(form18.memo1.text,true)
;
memo1.text := form18.memo1.text; }
end
else
messagedlg('ยังไม่ได้ทำการติดต่อ
พอร์ต',mterror,[mbOK],0);
end;

end.

```

```

unit edit_ok;

interface

uses
Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
StdCtrls, Grids, DBGrids, DBCtrls, Mask, Db,
DBTables, ComCtrls, ExtCtrls,
Buttons, Comms;

type
TForm3 = class(TForm)
PageControl1: TPageControl;
TabSheet1: TTabSheet;
TabSheet2: TTabSheet;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
Edit1: TEdit;
Edit2: TEdit;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
ComboBox1: TComboBox;
ComboBox2: TComboBox;
TabSheet3: TTabSheet;
GroupBox1: TGroupBox;
Panel1: TPanel;
Panel2: TPanel;
Edit3: TEdit;
Edit4: TEdit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BitBtn3: TBitBtn;  
 BitBtn4: TBitBtn;  
 DBGrid2: TDBGrid;  
 GroupBox3: TGroupBox;  
 GroupBox4: TGroupBox;  
 Panel3: TPanel;  
 Panel4: TPanel;  
 Edit5: TEdit;  
 Edit6: TEdit;  
 BitBtn7: TBitBtn;  
 BitBtn8: TBitBtn;  
 Label11: TLabel;  
 Label12: TLabel;  
 Label13: TLabel;  
 Label14: TLabel;  
 Label15: TLabel;  
 DBEdit3: TDBEdit;  
 DBEdit4: TDBEdit;  
 BitBtn9: TBitBtn;  
 BitBtn10: TBitBtn;  
 DBGrid3: TDBGrid;  
 BitBtn11: TBitBtn;  
 BitBtn12: TBitBtn;  
 DBEdit5: TDBEdit;  
 BitBtn13: TBitBtn;  
 Label16: TLabel;  
 DBGrid4: TDBGrid;  
 Label17: TLabel;  
 DBGrid5: TDBGrid;  
 DataSource1: TDataSource;  
 Table1: TTable;  
 BitBtn14: TBitBtn;  
 DataSource2: TDataSource;  
 DBEdit7: TDBEdit;  
 Table2: TTable;  
 DBEdit8: TDBEdit;  
 DBEdit9: TDBEdit;  
 ComboBox3: TComboBox;  
 Table1ID: TStringField;  
 Table1SEX: TStringField;  
 Table1NAME: TStringField;  
 Table1YEAR: TStringField;  
 Table1STATUS: TStringField;  
 Table1REASON: TStringField;  
 Table1CODE: TStringField;  
 Table2ID: TStringField;  
 Table2SEX: TStringField;  
 Table2NAME: TStringField;  
 Table2YEAR: TStringField;  
 Table2STATUS: TStringField;  
 Table2REASON: TStringField;  
 Table2CODE: TStringField;  
 Table1Mag: TStringField;  
 DataSource3: TDataSource;  
 Table3: TTable;  
 ComPort1: TComPort;  
 GroupBox2: TGroupBox;  
 Label5: TLabel;  
 Label6: TLabel;  
 Label7: TLabel;  
 Label8: TLabel;  
 Label10: TLabel;  
 BitBtn5: TBitBtn;  
 BitBtn6: TBitBtn;  
 DBEdit1: TDBEdit;  
 DBEdit2: TDBEdit;  
 DBEdit6: TDBEdit;  
 DBEdit12: TDBEdit;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DBEdit10: TDBEdit;
DBComboBox1: TDBComboBox;
DBComboBox2: TDBComboBox;
SpeedButton1: TSpeedButton;
procedure BitBtn2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure TabSheet1DragDrop(Sender,
Source: TObject; X, Y: Integer);
procedure PageControl1Change(Sender:
TObject);
procedure FindCode;
procedure Insert;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure Panel1Click(Sender: TObject);
procedure Panel2Click(Sender: TObject);
procedure TabSheet2DragDrop(Sender,
Source: TObject; X, Y: Integer);
procedure BitBtn6Click(Sender: TObject);
procedure Back;
procedure BitBtn11Click(Sender: TObject);
procedure BitBtn12Click(Sender: TObject);
procedure TabSheet3DragDrop(Sender,
Source: TObject; X, Y: Integer);
procedure Panel3Click(Sender: TObject);
procedure Panel4Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn13Click(Sender: TObject);
procedure AddCode;
procedure Delete;
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);

procedure BitBtn14Click(Sender: TObject);
procedure Table2FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
procedure TabSheet3Exit(Sender:
TObject);
procedure Table3FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
procedure Modify;
procedure BitBtn5Click(Sender: TObject);
procedure ConvertBi( var x : integer);
procedure ConvertAscii ( x : integer ; var
ch : char);
procedure Error;
procedure FindStatus ( var year2 : integer);
procedure ComPort1RxChar(Sender:
TObject; InQue: Integer);
procedure FormClose(Sender: TObject; var
Action: TCloseAction);
procedure ComPort1Open(Sender:
TObject);
procedure SpeedButton1Click(Sender:
TObject);
procedure FormShow(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
procedure Clear;
end;
var
Form3: TForm3;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

click_name,click_id,result_n,result_jd,result ;
boolean;
showmess,start,check : boolean;
code : string[3];
data,str : string;
change : boolean;
year : string[8];

implementation

uses main_ok, card1, data_storage.
UpLoad_Send_Status.
UpLoad_Send_data;

{$R *.DFM}

procedure TForm3.FormCreate(Sender:
TObject);
begin
table1.open;
table1.Refresh;
table1.first;
pagecontrol1.activepage := tabsheet1;
edit1.clear;
edit2.clear;
combobox1.text:='2';
combobox2.text:='ใช้ได้';
edit3.hide;
edit4.hide;
edit5.hide;
edit6.hide;
end;

procedure TForm3.TabSheet1DragDrop
(Sender, Source: TObject; X, Y: Integer);
begin
table1.Open;
table1.refresh;
table1.First;
edit1.clear;
edit2.clear;
combobox1.text:='2';
combobox2.text:='ใช้ได้';
end;

procedure TForm3.PageControl1Change
(Sender: TObject);
begin
table1.first;
if pagecontrol1.activepage = tabsheet2
then
begin
edit3.Clear;
edit3.hide;
edit4.clear;
edit4.hide;
dbgrid2.show;
dbgrid5.Hide;
label17.hide;
tabsheet2.Enabled := true;
end;
if pagecontrol1.activepage= tabsheet3
then
begin
edit5.clear;
edit5.hide;
edit6.clear;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

edit6.hide;
dbgrid3.Show;
dbgrid4.hide;
label16.hide;
tabsheet3.enabled := true;
end;
end;

procedure TForm3.Clear;
begin
    table1.cancel;
    edit1.clear;
    edit2.clear;
    combobox1.text:='2';
    combobox2.text:='ใช้ได้';
end;

procedure TForm3.BitBtn2Click(Sender:
TObject); {CANCEL ADD}
begin
    clear;
end;

procedure TForm3.FindCode;
var
    str : string[3];
    size : integer;
begin
    table3.open;
    table3.first;
    str := table3['code'];
    if str = '999' then
    begin
        table1.open;
        table1.indexname := 'idxcode';
        table1.last;
        code := table1.fieldvalues['code'];
        code := inttostr(strtoint(code) + 1);
        size := length(code);
        while size < 3 do
        begin
            code := '0' + code;
            form10.code := code;
            size := size + 1;
        end;
        table1.indexname := 'idxid';
    end
    else
    begin
        code := str;
        form10.code := code;
        table3.delete;
        table3.close;
    end;
end;

Procedure TForm3.Insert;
begin
    if (edit1.text <> "") and (edit2.text <> "")
        and ( combobox1.text <> "") and
        (combobox2.text<>") then
    begin
        table1.append;
        table1.fieldvalues['id']:=edit1.text;
        table1.fieldvalues['name']:=edit2.text;

        table1.fieldvalues['year']:=combobox1.text;
        table1.fieldvalues['status']:=combobox2.text;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

table1.fieldvalues['sex']:=combobox3.text;
    table1.fieldvalues['code']:= code;
    table1.fieldvalues['mag'] := data;
    table1.post; // เก็บข้อมูลผู้ใช้ใหม่
    table1.indexname := 'idxid';
    form10.close;
    messagedlg('การเพิ่มผู้ใช้เรียบร้อย',mtinformation,[mbOK],0);
    form3.enabled := true;
    edit1.clear;
    edit2.clear;
    combobox1.text:='2';
    combobox2.text:='ใช้ได้';
end;
form3.enabled := true;
end;

procedure TForm3.BitBtn1Click(Sender:
TObject); {ADD}
begin
    if (edit1.text <>") and (edit2.text <>")
    and (combobox1.text<>") and
    (combobox2.text<>")then
    begin
        FindCode; // หา code ให้ผู้ใช้ใหม่
        start := false;
        if not comport1.connected then
            comport1.open
        else
            begin
                comport1.writestring('90',true);
                form3.enabled := false;
                form17.form := 'edit';
                form17.show;
            end;
    end;
end;
else
    messagedlg('กรุณากรอกข้อมูลให้ครบ',mterror,[mbOK],0);
end;
end;

(For Tab2)
procedure TForm3.BitBtn3Click(Sender:
TObject); {button SEARCH click}
begin
    showmess := false;
    if click_name then
    begin
        if edit3.text = " then
            click_name := false
        else
            begin
                table1.indexname := 'idxname';
                table1.setkey;
                table1.FieldValues['name'] :=
                edit3.text;
                result_n := table1.gotokey;
                if not result_n then
                    begin
                        table2.close;
                        table2.open;
                        dbedit1.dataSource :=
                        datasource2;
                        dbedit2.datasource :=
                        datasource2;
                        dbcombox1.dataSource :=
                        datasource2;
                    end;
            end;
        end;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dbcombobox2.datasource :=
datasource2;
dbedit6.datasource :=
datasource2;
dbedit12.datasource :=
datasource2;
dbedit10.datasource :=
datasource2;
result_n := true;
messagedlg('สามารถดูข้อมูลใกล้เคียงได้จากตารางข้อมูลใกล้เคียง',mtcustom,
[mbok],0);
showmess := true;
dbgrid2.hide;
dbgrid5.show;
label17.show;
end;
end;
if click_id then
begin
if edit4.text = '' then
click_id := false ;
table1.indexname := 'idxid';
table1.SetKey;
table1.FieldValues['id'] := edit4.text;
result_id := table1.gotokey;
end;
if click_name and click_id then
begin
result := result_n and result_id;
result_n := result_n and not
(click_id);
result_id := result_id and not
(click_name);
if (not (result or result_n or result_id))
and not showmess then
begin
messagedlg('ไม่พบข้อมูลในฐานข้อมูล',mtinformation,[mbok],0);
messagedlg('สามารถดูข้อมูลใกล้เคียงได้จากตารางข้อมูลใกล้เคียง',mtcustom,
[mbok],0);
dbgrid2.hide;
dbgrid5.show;
label17.show;
end
end
else
if click_id then
if not result_id then
begin
messagedlg('ไม่พบข้อมูลในฐานข้อมูล',mtinformation,[mbok],0);
dbgrid2.hide;
dbgrid5.hide;
label17.Hide;
end;
end;
procedure TForm3.BitBtn4Click(Sender:
TObject); {button CANCEL SEARCH click}
begin
table1.first;
edit3.clear;
edit3.hide;
edit4.clear;
edit4.hide;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

label17.hide;
dbgrid5.Hide;
dbgrid2.show;
dbedit1.datasource := datasource1;
dbedit2.DataSource := datasource1;
dbedit6.datasource := datasource1;
dbcombobox1.datasource :=
datasource1;
dbcombobox2.datasource :=
datasource1;
dbedit10.datasource := datasource1;
end;

procedure TForm3.Panel1Click(Sender:
TObject); {panel NAME click}
begin
edit3.show;
click_name := true;
end;

procedure TForm3.Panel2Click(Sender:
TObject); {panel ID click}
begin
edit4.show;
click_id := true;
end;

procedure TForm3.TabSheet2DragDrop
(Sender, Source: TObject; X, Y: Integer);
begin
table1.open;
table1.refresh;
table1.first;
edit3.hide;

edit4.hide;
end;

procedure TForm3.BitBtn6Click(Sender:
TObject); {CANCEL EDIT table}
begin
table1.Cancel;
dbedit1.datasource := datasource1;
dbedit2.DataSource := datasource1;
dbedit6.datasource := datasource1;
dbcombobox1.datasource :=
datasource1;
dbcombobox2.datasource :=
datasource1;
dbedit10.datasource := datasource1;
end;

procedure TForm3.Back;
begin
table1.first;
form3.Hide;
form1.show;
comport1.close;
end;

procedure TForm3.BitBtn11Click(Sender:
TObject); {back to main}
begin
back;
end;

procedure TForm3.BitBtn12Click(Sender:
TObject); {back to main}
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

--back; click_name := false
end; else
begin
procedure TForm3.TabSheet3DragDrop table1.indexname :='idxname';
(Sender, Source: TObject; X, Y: Integer); table1.setkey;
begin table1.FieldValues['name'] :=
edit5.text;
table1.open; edit5.text;
table1.refresh; result_n := table1.gotokey;
table1.first; if not result_n then
edit5.hide; begin
edit6.hide; table2.Close;
end; table2.open;
procedure TForm3.Panel3Click(Sender: dbedit3.datasource :=
TObject); {panel name click} dbedit4.datasource :=
begin dbedit5.datasource :=
edit5.show; datasource2;
click_name := true; dbedit7.datasource :=
end; datasource2;
procedure TForm3.Panel4Click(Sender: dbedit8.datasource :=
TObject); {panel id click} datasource2;
begin dbedit9.datasource :=
edit6.show; datasource2;
click_id := true; result_n := true;
end; messagedlg('สามารถดูข้อมูลใกล้เคียงได้จากตารางข้อมูลใกล้เคียง',mtcustom,
procedure TForm3.BitBtn7Click(Sender: [mbok],0);
TObject); {SEARCH} showmess := true;
begin dbgrid3.hide;
showmess := false; dbgrid4.show;
if click_name then label16.show;
begin end;
if edit5.text = " then end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
if click_id then
begin
  if edit6.text = "" then
    click_id := false ;
    table1.indexname := 'idxid';
    table1.SetKey;
    table1.FieldValues['id'] := edit6.text;
    result_id := table1_gotokey;
end;
if click_name and click_id then
begin
  result := result_n and result_id;
  result_n := result_n and not
(click_id);
  result_id := result_id and not
(click_name);
  if (not (result or result_n or result_id))
and not showmess then
  begin
    messagedlg('ไม่พบข้อมูลในฐานข้อมูล',mtinformation,[mbok],0);
    messagedlg('สามารถดูข้อมูลใกล้เคียงได้จากตารางข้อมูลใกล้เคียง',mtcustom,[mbok],0);
    dbgrid3.hide;
    dbgrid4.show;
    label16.show;
  end
end
else
  if click_id then
    if not result_id then
      begin
        messagedlg('ไม่พบข้อมูลในฐานข้อมูล',mtinformation,[mbok],0);
        dbgrid3.hide;
        dbgrid4.hide;
        label16.Hide;
      end
    end;
  procedure TForm3.BitBtn8Click(Sender: TObject); {CANCEL SEARCH}
  begin
    table1.first;
    edit5.clear;
    edit5.hide;
    edit6.clear;
    edit6.hide;
    click_name := false;
    click_id := false;
    dbgrid3.show;
    dbgrid4.hide;
    label16.hide;
    dbedit3.datasource := datasource1;
    dbedit4.datasource := datasource1;
    dbedit5.datasource := datasource1;
    dbedit7.datasource := datasource1;
    dbedit8.datasource := datasource1;
    dbedit9.datasource := datasource1;
  end;
  procedure TForm3.BitBtn13Click(Sender: TObject); {back to main}
  begin
    back;
  end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm3.AddCode:
begin
    table3.open;
    table3.append;
    table3['code'] := code;
    table3.post;
    table3.close;
end;

procedure TForm3.Delete;
begin
    AddCode;
    table1.delete;
    table1.first;
    dbgrid3.show;
    dbgrid4.hide;
    label16.hide;
    form19.close;
    messagedlg('การลบข้อมูลเรียบร้อยแล้ว',mtinformation,[mbOK],0);
    form3.enabled := true;
end;

procedure TForm3.BitBtn9Click(Sender:
TObject); {button DELETE click}
begin
    code := table1.fieldvalues['code'];
    if not comport1.connected then
        comport1.Open
    else
        begin
            comport1.writestring('90',true); // ส่ง
สัญญาณขอติดต่อ 51
            form3.enabled := false;
        end;
end;

form17.form := 'edit';
form17.show;
end;
end;

procedure TForm3.BitBtn10Click(Sender:
TObject);{button CANCEL DELETE click}
begin
    table1.first;
    dbedit1.clear;
    dbedit2.clear;
    dbedit3.clear;
    dbedit4.clear;
    dbedit3.datasource := datasource1;
    dbedit4.datasource := datasource1;
    dbedit5.datasource := datasource1;
    dbedit7.datasource := datasource1;
    dbedit8.datasource := datasource1;
    dbedit9.datasource := datasource1;
end;

procedure TForm3.BitBtn14Click(Sender:
TObject);
begin
    form10.show;
end;

procedure TForm3.Table2FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var index : string[2];
begin
    if pagecontrol1.activepage = tabsheet2
    then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

x := ans1;
end;
size := length(ans);
if size > 7 then
  x := strtoint(copy(ans,size-6,7))
else
  x := strtoint(ans);
end;

procedure TForm3.ConvertAscii (x : integer;
var ch : char);
var
  ans1,ans2,size,i,j : integer;
  ans : string;
begin
  ans := "";
  while x <> 0 do
    begin
      ans1 := x div 2;
      ans2 := x mod 2;
      ans := inttostr(ans2) + ans;
      x := ans1;
    end;
  size := length(ans);
  if size > 7 then
    ans := copy(ans,size-6,7);
  ans1 := strtoint(ans[7]);
  size := 1;
  for i := 6 downto 1 do
    begin
      ans2 := 1;
      for j := 1 to size do
        ans2 := (strtoint(ans[ij])*2)*ans2;
        ans1 := ans1 + ans2;
        size := size+1;
      end;
      ch := chr(ans1);
    end;
  end;

procedure TForm3.Error;
begin
  form10.close;
  form19.close;
  if messagedlg('การส่งข้อมูลผิดพลาด
ต้องการส่งใหม่',mterror,[mbOK,mbCancel],0)
= mrCancel then
    begin
      form3.enabled := true;
      form3.clear;
    end
  else
    begin // ติดต่อใหม่อีกครั้ง
      form19.close;
      form10.close;
      form3.comport1.writestring('90',true);
      // ขอติดต่อกับ 51 ใหม่
      form17.form := 'edit';
      form17.show;
    end;
  end;

procedure TForm3.FindStatus ( var year2 :
integer);
var
  ch : char;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ch := year[1];
case ch of
  '2' : year2 := 2;
  '3' : year2 := 3;
  '4' : year2 := 4;
  'ป' : year2 := 5;
  'อ' : year2 := 6;
end;
end;

procedure TForm3.ComPort1RxChar(Sender:
TObject; InQue: Integer);
const
  free = 'bf';
  stop = 'e';
var
  str,temp : string;
  status : string[9];
  i,size,num,sum51,code2,status2,year2 :
integer;
  ch,stop1 : char;
begin
  comport1.readstring(str,inque,true);
  temp := 'aa';
  stop1 := 'a';
  if str <> "" then
  begin
    size := length(str);
    temp := copy(str,size-1,2);
    stop1 := str[1];
  end;
  if pagecontrol1.activepage = tabsheet1
  then
  begin
    if temp = free then
    begin
      start := true;
      data := "";
      num := 0;
      comport1.writestring('A',true); // ส่ง
      คำสั่ง "เพิ่ม"
      form17.close;
      form10.show;
    end
    else
    begin
      if start then
      begin
        data := data + str;
        form10.progressBar1.stepby(2);
        size := length(data);
        if size > 40 then
        begin
          num := 0;
          form10.progressBar1.stepby(2);
          str := copy(data,1,40); // เก็บ
          40 byte
          form10.progressBar1.stepby(2);
          for i:= 1 to 40 do
          begin
            ch := data[i];
            num := num + ord(ch);
          end;
          form10.progressBar1.stepby(2);
          ch := data[41];
          form10.progressBar1.stepby(3);
        end;
      end;
    end;
  end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ch = '#' then
    begin
form10.progressbar1.stepby(4);
        ch := data[43];
        ord(ch);
form10.progressbar1.stepby(5);
        if ch = stop then
            begin
                combobox1.text;
                status := table1
                ['status'];
form10.progressbar1.stepby(6);
                ch := data[42];
                then
form10.progressbar1.stepby(7);
                sum51 := ord(ch);
                status2 := 0
                else
form10.progressbar1.stepby(9);
                convertbi(sum51);
                (status2);
                status2 := 1;
                findstatus(year2);
                convertasci
                (code2+status2+48+year2+48,ch);
form10.progressbar1.stepby(10);
                converbi(num);
                str :=
                'a'+code+'#'+inttostr(status2)+'#'+inttostr
                (year2)+'#'+ch+'z';
form10.progressbar1.stepby(10);
                start := false;
                comport1.writestring
                error
                (str,true);
            else
                end;
            begin
                end
                data := str; // ข้อมูล
            else
                error;
            while
                end
form10.progressbar1.position <> 50000 do
                else
                error;
form10.progressbar1.stepby(10);
                end;
                code2 := 0;
                end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
begin
    if stop1 = '' then
        error
    else
begin
    if stop1 = 'E' then
        insert;
    end;
end;
end;
end
else
begin
    if pagecontrol1.activepage = tabsheet2
then
begin
    if not change then
begin
    if temp = free then
begin
        comport1.writestring('M',true);
// ส่งคำสั่งแก้ไข
        start := true;
        form17.close;
        form19.form := 'edit';
        form19.show;
    end
    else
begin
        if start then
begin
            if str = 'R' then
begin
                year :=
                    dbcombobox1.text;
                ch := year[1];
                case ch of
                    '2' : year := '2';
                    '3' : year := '3';
                    '4' : year := '4';
                    'ป' : year := '5';
                    'อ' : year := '6';
                end;
                code := dbedit10.text;
                code2 := 0;
                for i := 1 to 3 do
begin
                    ch := code[i];
                    code2 := code2 +
                        ord(ch);
                end;
                status :=
                    dbcombobox2.text;
                if status = 'ใช้ไม่ได้' then
                    status2 := 0
                else
                    status2 := 1;
                year2 := strtoint(year);
                convertascii
                    (code2+status2+48+year2+48,ch);
                str :=
                    'a'+code+'#'+inttostr(status2)+'#+year+'#'+c
                    h+'z';
                comport1.writestring(str,true);
            end
        else
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if stop1 = '!' then
    error
else
    begin
        if stop1 = 'E' then
            begin
                form19.Close;
                messagedlg('การ
แก้ไขเรียบร้อย', mtinformation, [mbOK].0);
                form3.enabled :=
true;
            end;
        end;
    end;
end;
end;
end;
end;
else
begin // เปลี่ยนบัตรใหม่
    if temp = free then
        begin
            start := true;
            data := "";
            num := 0;
            comport1.writestring
('C', true); // ส่งคำสั่ง เปลี่ยนบัตร
            form17.close;
            form10.show;
        end
    else
        begin
            if start then
                begin
                    data := data + str;
                    form10.progressbar1.stepby(2);
                    size := length(data);
                    if size > 40 then
                        begin
                            num := 0;
                            form10.progressbar1.stepby(2);
                            str := copy(data, 1, 40);
                            // เก็บ 40 byte
                            form10.progressbar1.stepby(2);
                            for i:= 1 to 40 do
                                begin
                                    ch := data[i];
                                    num := num + ord
(ch);
                                    form10.progressbar1.stepby(2);
                                end;
                                ch := data[41];
                                form10.progressbar1.stepby(3);
                                if ch = '#' then
                                    begin
                                        form10.progressbar1.stepby(4);
                                        ch := data[43];
                                        form10.progressbar1.stepby(5);
                                        if ch = stop then
                                            begin
                                                form10.progressbar1.stepby(6);
                                                ch := data[42];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
form10.progressbar1.stepby(7);
sum51 := ord table1['status'];
status :=
(ch);
if status = 'ใช้
ไม่ได้' then
form10.progressbar1.stepby(9);
status2 := 0
convertbi else
(sum51);
status2 := 1;
findstatus(year2);
form10.progressbar1.stepby(10);
convertasci
convertbi(num); (code2+status2+48+year2+48,ch);
str :=
form10.progressbar1.stepby(10); 'a'+code+'#'+inttostr(status2) + '#'+year
+'#'+ch+'z';
if sum51 <>
num then
error
comport1.writestring(str,true);
else
start := false;
begin
end;
data := str; //
end
else
while
error;
form10.progressbar1.position <> 50000 do
end
else
form10.progressbar1.stepby(10);
error;
code :=
end;
end
dbedit10.text;
code2 := 0;
else
for i := 1 to 3
begin
do
if stop1 = 'I' then
error
begin
ch :=
else
code[i];
begin
code2 :=
if stop1 = 'E' then
begin
code2 + ord(ch);
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        end;
    end;
end;

end;
end;
end;
end;

procedure TForm3.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
    comport1.Close;
end;

procedure TForm3.ComPort1Open(Sender:
TObject);
begin
    comport1.writestring('90',true);
    form3.enabled := false;
    form17.form := 'edit';
    form17.show;
end;

procedure TForm3.SpeedButton1Click
(Sender: TObject);
begin
    change := true;
    if not comport1.connected then
        comport1.open
    else
        begin
            comport1.writestring('90',true);
            form3.enabled := false;
            form17.form := 'edit';
            form17.show;
        end;
end;

procedure TForm3.FormShow(Sender:
TObject);
begin
    table1.open;
    table1.Refresh;
    table1.first;
    pagecontrol1.activepage := tabsheet1;
    edit1.clear;
    edit2.clear;
    combobox1.text := '2';
    combobox2.text := 'ใช้ได้';
    edit3.hide;
    edit4.hide;
    edit5.hide;
    edit6.hide;
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit graph_status;
interface
uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  Buttons, TeEngine, Series, ExtCtrls,
  TeeProcs, Chart, DBChart, Db,
  DBTables;

type
  TForm15 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    DataSource2: TDataSource;
    Query2: TQuery;
    DBChart1: TDBChart;
    Series1: TBarSeries;
    Series2: TBarSeries;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    Panel1: TPanel;
    procedure SpeedButton1Click(Sender:
TObject);
    procedure SpeedButton2Click(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form15: TForm15;
implementation
uses view_ok, main_ok;

{$R *.DFM}

procedure TForm15.SpeedButton1Click
(Sender: TObject);
begin
  form15.Hide;
  form2.show;
end;

procedure TForm15.SpeedButton2Click
(Sender: TObject);
begin
  form15.hide;
  form1.show;
end;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit graph1;                                implementation

interface                                    uses pastday_ok, view_ok,
                                              main_ok,show_graph, graph_status;

uses
  Windows, Messages, SysUtils, Classes,    ($R *.DFM)
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, TeeProcs, TeEngine,   {procedure TForm8.BitBtn1Click(Sender:
  Chart, mxgraph, Grids, mxgrid,          TObject);
  ExtCtrls, mxpivsrc, mxDB, Db, DBTables,  begin
  mxtables, mxstore, Series;              form8.hide;
                                              form2.show;
                                              end;}

type
  TForm8 = class(TForm)
    Panel1: TPanel;                          procedure TForm8.SpeedButton1Click
    Panel2: TPanel;                          (Sender: TObject);
    SpeedButton1: TSpeedButton;             begin
    SpeedButton2: TSpeedButton;             form8.hide;
    Label1: TLabel;                         form14.show;
    Label2: TLabel;                         end;
    {procedure BitBtn1Click(Sender: TObject);}
    procedure SpeedButton1Click(Sender:      procedure TForm8.SpeedButton2Click
TObject);                                  (Sender: TObject);
    procedure SpeedButton2Click(Sender:      begin
TObject);                                  form8.hide;
private                                     form15.show;
  { Private declarations }                  end;
public
  { Public declarations }                   end.
end;

var
  Form8: TForm8;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

password in 'password.pas' {Form21};

program mag_ok;

uses
  Forms,
  main_ok in 'main_ok.pas' {Form1},
  view_ok in 'view_ok.pas' {Form2},
  data_storage in 'data_storage.pas'
{Form18},
  DownLoad in 'DownLoad.pas' {Form20},
  edit_ok in 'edit_ok.pas' {Form3},
  card1 in 'card1.pas' {Form10},
  oneday_ok in 'oneday_ok.pas' {Form4},
  UpLoad in 'UpLoad.pas' {Form16},
  UpLoad_Send_data in
'UpLoad_Send_data.pas' {Form19},
  UpLoad_Send_Status in
'UpLoad_Send_Status.pas' {Form17},
  graph1 in 'graph1.pas' {Form8},
  graph_status in 'graph_status.pas' {Form15},
  show_graph in 'show_graph.pas' {Form14},
  cannotuse_preview in
'cannotuse_preview.pas' {Form13},
  notclose_ok in 'notclose_ok.pas' {Form6},
  notclose_preview in 'notclose_preview.pas'
{Form12},
  oneday_preview in 'oneday_preview.pas'
{Form9},
  pastday_ok in 'pastday_ok.pas' {Form5},
  pastday_preview in 'pastday_preview.pas'
{Form11},
  cannotuse_ok in 'cannotuse_ok.pas'
{Form7},

  {SR *.RES}

begin
  Application.Initialize;
  Application.HelpFile := 'C:\Program
Files\Borland\Delphi4\Magnetic\Try1.hlp';
  Application.Title := 'Access Control Using
Magnetic Card';
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm18, Form18);
  Application.CreateForm(TForm20, Form20);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm10, Form10);
  Application.CreateForm(TForm4, Form4);
  Application.CreateForm(TForm16, Form16);
  Application.CreateForm(TForm19, Form19);
  Application.CreateForm(TForm17, Form17);
  Application.CreateForm(TForm8, Form8);
  Application.CreateForm(TForm15, Form15);
  Application.CreateForm(TForm14, Form14);
  Application.CreateForm(TForm13, Form13);
  Application.CreateForm(TForm6, Form6);
  Application.CreateForm(TForm12, Form12);
  Application.CreateForm(TForm9, Form9);
  Application.CreateForm(TForm5, Form5);
  Application.CreateForm(TForm11, Form11);
  Application.CreateForm(TForm7, Form7);
  Application.CreateForm(TForm21, Form21);
  Application.Run;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit main_ok;
interface
uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, Db, DBTables;
type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    procedure SpeedButton6Click(Sender:
  TObject);
    procedure SpeedButton1Click(Sender:
  TObject);
    procedure SpeedButton4Click(Sender:
  TObject);
    procedure SpeedButton5Click(Sender:
  TObject);
    procedure SpeedButton3Click(Sender:
  TObject);
    procedure SpeedButton2Click(Sender:
  TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  var
    Form1: TForm1;
  implementation
    uses view_ok, edit_ok, card1, UpLoad,
    DownLoad, oneday_ok, password ;
    {$R *.DFM}
    procedure TForm1.SpeedButton6Click
    (Sender: TObject);
    begin
      form1.close;{exit}
    { table1.close; }
    end;
    procedure TForm1.SpeedButton1Click
    (Sender: TObject);
    begin
      form1.hide;
      form4.tabsheet := 1;
      form2.show;{view}
    end;
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.SpeedButton4Click
(Sender: TObject);
begin
  form1.hide;
  form21.show;
end;

procedure TForm1.SpeedButton5Click
(Sender: TObject);
begin
  application.HelpFile := 'try1.hlp';
  application.helpjump('contents');
end;

procedure TForm1.SpeedButton3Click
(Sender: TObject);
begin
  form1.hide;
  form16.show;
end;

procedure TForm1.SpeedButton2Click
(Sender: TObject);
begin
  form1.hide;
  form20.show;
end;

end.

unit notclose_ok;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  Db, Grids, DBGrids, DBTables, StdCtrls,
  Buttons;

type
  TForm6 = class(TForm)
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    Table2: TTable;
    GroupBox1: TGroupBox;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Table2ID: TStringField;
    Table2SEX: TStringField;
    Table2NAME: TStringField;
    Table2YEAR: TStringField;
    Table2STATUS: TStringField;
    Table2REASON: TStringField;
    Table2CODE: TStringField;
    Table1: TTable;
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Table1ID: TStringField;
Table1NAME: TStringField;
Table1YEAR: TStringField;
PrintDialog1: TPrintDialog;
Table1Date: TDateField;
Table1Login: TTimeField;
Table1Logout: TTimeField;
Table1Com: TStringField;
procedure BitBtn1Click(Sender: TObject);
procedure Table1FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
procedure SpeedButton3Click(Sender:
TObject);
procedure SpeedButton4Click(Sender:
TObject);
procedure SpeedButton2Click(Sender:
TObject);
procedure SpeedButton1Click(Sender:
TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form6: TForm6;

implementation

uses view_ok, main_ok, notclose_preview;

{$R *.DFM}

procedure TForm6.BitBtn1Click(Sender:
TObject);
begin
  form6.hide;
  form2.show;
end;

procedure TForm6.Table1FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
begin
  accept := timetostr(datasets['logout'])=
'0:00:00';
end;

procedure TForm6.SpeedButton3Click
(Sender: TObject);
begin
  form6.hide;
  form2.show;
end;

procedure TForm6.SpeedButton4Click
(Sender: TObject);
begin
  form6.hide;
  form1.show;
end;

procedure TForm6.SpeedButton2Click
(Sender: TObject);
begin
  form12.quickrep1.preview;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm6.SpeedButton1Click
```

```
(Sender: TObject);
```

```
begin
```

```
    if printdialog1.execute then
```

```
        form12.print;
```

```
end;
```

```
end.
```

```
unit notclose_preview;
```

```
interface
```

```
uses
```

```
    Windows, Messages, SysUtils, Classes,
```

```
    Graphics, Controls, Forms, Dialogs,
```

```
    ExtCtrls, QuickRpt, QrCtrls, Db, DBTables;
```

```
type
```

```
    TForm12 = class(TForm)
```

```
        QuickRep1: TQuickRep;
```

```
        PageHeaderBand1: TQRBand;
```

```
        TitleBand1: TQRBand;
```

```
        ColumnHeaderBand1: TQRBand;
```

```
        DetailBand1: TQRBand;
```

```
        QRLabel1: TQRLabel;
```

```
        QRSysData1: TQRSysData;
```

```
        QRLabel2: TQRLabel;
```

```
        QRSysData2: TQRSysData;
```

```
        QRLabel3: TQRLabel;
```

```
        QRLabel4: TQRLabel;
```

```
        QRLabel5: TQRLabel;
```

```
        QRLabel6: TQRLabel;
```

```
        QRLabel7: TQRLabel;
```

```
        QRLabel8: TQRLabel;
```

```
        QRLabel9: TQRLabel;
```

```
        QRDBText1: TQRDBText;
```

```
        DataSource1: TDataSource;
```

```
        DataSource2: TDataSource;
```

```
        Table2: TTable;
```

```
        QRDBText2: TQRDBText;
```

```
        QRDBText3: TQRDBText;
```

```
        QRDBText4: TQRDBText;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

QRDBText5: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
Table2ID: TStringField;
Table2SEX: TStringField;
Table2NAME: TStringField;
Table2YEAR: TStringField;
Table2STATUS: TStringField;
Table2REASON: TStringField;
Table2CODE: TStringField;
Table1: TTable;
Table1ID: TStringField;
Table1NAME: TStringField;
Table1YEAR: TStringField;
QRDBText8: TQRDBText;
Table1sex: TStringField;
Table1Date: TDateField;
Table1Login: TTimeField;
Table1Logout: TTimeField;
Table1Com: TStringField;
procedure Table1FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form12: TForm12;

implementation

{$R *.DFM}

```

```

procedure TForm12.Table1FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
begin
  accept := timetostr(datasel['logout']) =
'0:00:00';
end;
end.

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit oneday_ok:
    Table1ID: TStringField;
    Table1NAME: TStringField;
    Table1YEAR: TStringField;
    PnntDialog1: TPrintDialog;
    Table1Date: TDateField;
    Table1Login: TTimeField;
    Table1Logout: TTimeField;
    Table1Com: TStringField;
    procedure SpeedButton4Click(Sender:
        TObject);
    procedure SpeedButton3Click(Sender:
        TObject);
    procedure Table1FilterRecord(DataSet:
        TDataSet; var Accept: Boolean);
    procedure SpeedButton2Click(Sender:
        TObject);
    procedure SpeedButton1Click(Sender:
        TObject);
    procedure Label5Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    tabsheet : integer;
end;

var
    Form4: TForm4;
    today : string[8];
    year : string[3];
    size : integer;

implementation

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

uses view_ok, main_ok, oneday_preview;

($R *.DFM)

procedure TForm4.SpeedButton4Click
(Sender: TObject);
begin
    form4.hide;
    form1.show;
end;

procedure TForm4.SpeedButton3Click
(Sender: TObject);
begin
    form4.hide;
    tabsheet := 2;
    form2.show;
end;

procedure TForm4.Table1FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
begin
    today := datetostr(now);
    size := length(today);
    year := copy(today,size-1,2);
    if strtoint(year) < 40 then
    begin
        year := inttostr(strtoint(year)+43);
        size := length(year);
        if size > 2 then
            year := copy(year,2,2);
    end;
end;

size := length(today);
today := copy(today,1,size-2) + year;
end;
accept := dataset['date'] = strtodate
(today);

procedure TForm4.SpeedButton2Click
(Sender: TObject);
begin
    form9.quickrep1.preview;
end;

procedure TForm4.SpeedButton1Click
(Sender: TObject);
begin
    if printdialog1.execute then
        form9.print;
end;

procedure TForm4.Label5Click(Sender:
TObject);
begin
    { today := datetostr(now);
    size := length(today);
    year := copy(today,size-1,2);
    if strtoint(year) < 40 then
    begin
        year := inttostr(strtoint(year)+43);
        size := length(year);
        if size > 2 then
            year := copy(year,2,2);
    end;
}
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    today := copy(today,1_size-2) + year;
end; }
end;

procedure TForm4.FormShow(Sender:
TObject):
begin
    table1.open;
    labsheet := 2;
end;
end.

```

```

unit oneday_preview;

interface

uses
    Windows, Messages, SysUtils, Classes,
    Graphics, Controls, Forms, Dialogs,
    Db, QrCtrls, QuickRpt, ExtCtrls, DBTables;

type
    TForm9 = class(TForm)
        DataSource1: TDataSource;
        DataSource2: TDataSource;
        Table2: TTable;
        QuickRep1: TQuickRep;
        PageHeaderBand1: TQRBand;
        TitleBand1: TQRBand;
        ColumnHeaderBand1: TQRBand;
        DetailBand1: TQRBand;
        QRLabel1: TQRLabel;
        QRSysData1: TQRSysData;
        QRLabel2: TQRLabel;
        QRSysData2: TQRSysData;
        QRLabel4: TQRLabel;
        QRLabel5: TQRLabel;
        QRLabel6: TQRLabel;
        QRLabel7: TQRLabel;
        QRLabel8: TQRLabel;
        QRLabel3: TQRLabel;
        Table2ID: TStringField;
        Table2SEX: TStringField;
        Table2NAME: TStringField;
        Table2YEAR: TStringField;
        Table2STATUS: TStringField;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Table2REASON: TStringField;           {$R*.DFM)
Table2CODE: TStringField;
Table1: TTable;                        procedure TForm9.Table1FilterRecord
Table1ID: TStringField;                (DataSet: TDataSet;
Table1NAME: TStringField;              var Accept: Boolean);
Table1YEAR: TStringField;              var
QRDBText1: TQRDBText;                  size : integer;
QRDBText2: TQRDBText;                  begin
QRDBText3: TQRDBText;                  today_preview := datetostr(now);
QRDBText4: TQRDBText;                  size := length(today_preview);
QRDBText5: TQRDBText;                  year := copy(today_preview,size-1,2);
QRDBText6: TQRDBText;                  if strtoint(year) < 40 then
QRDBText7: TQRDBText;                  begin
Table1sex: TStringField;                year := inttostr(strtoint(year)+43);
Table1Date: TDateField;                 size := length(year);
Table1Login: TTimeField;                if size > 2 then
Table1Logout: TTimeField;               year := copy(year,2,2);
Table1Com: TStringField;                size := length(today_preview);
procedure Table1FilterRecord(DataSet:   today_preview := copy
TDataSet; var Accept: Boolean);          (today_preview,1,size-2) + year;
private                                  end;
  { Private declarations }               accept := dataset['date'] = today_preview;
public                                    end;
  { Public declarations }
end;                                       end.

var
Form9: TForm9;
today_preview : string[10];
year : string[3];
size : integer;

```

#### implementation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit password;

interface
.
uses
  Windows, Messages, SysUtils, Classes
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Mask, Buttons, ComCtrls, Db,
  DBTables;

type
  TForm21 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    MaskEdit1: TMaskEdit;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    MaskEdit2: TMaskEdit;
    MaskEdit3: TMaskEdit;
    MaskEdit4: TMaskEdit;
    DataSource1: TDataSource;
    Table1: TTable;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
  procedure initial;
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  implementation
  uses main_ok, edit_ok;
  {$R *.DFM}
  procedure TForm21.initial;
  begin
    maskedit1.text := "";
    maskedit2.text := "";
    maskedit3.text := "";
    maskedit4.text := "";
  end;
  procedure TForm21.BitBtn2Click(Sender:
  TObject);
  begin
    form21.Hide;
    form1.show;
  end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm21.BitBtn1Click(Sender:
TObject);
var
    result : boolean;
begin
    table1.open;
    table1.setkey;
    table1['passw'] := maskedit1.text;
    result := table1.gotokey;
    if result then
    begin
        form21.hide;
        form3.show;
    end
    else
    begin
        messagedlg(' PASSWORD ไม่ถูก
ต้อง',mterror,[mbOK],0);
        initial;
    end;
    table1.close;
end;

```

```

procedure TForm21.BitBtn4Click(Sender:
TObject);
begin
    pagecontrol1.activepage := tabsheet1;
end;

```

```

procedure TForm21.BitBtn3Click(Sender:
TObject);
begin
    table1.open;
    table1.setkey;

```

```

    table1['passw'] := maskedit2.text;
    result := table1.gotokey;
    if result then
    begin
        if maskedit3.text = maskedit4.text then
        begin
            table1.delete;
            table1.append;
            table1['passw'] := maskedit3.text;
            table1.post;
            table1.close;
            messagedlg('การเปลี่ยน
PASSWORD เรียบร้อย' mtinformation,
[mbOK],0);
            initial;
        end
        else
        begin
            messagedlg('กรุณากรอก
PASSWORD ใหม่อีกครั้ง', mtinformation,
[mbOK],0);
            initial;
        end;
    end;

```

```

end
else
begin
    messagedlg(' PASSWORD ไม่ถูก
ต้อง',mterror,[mbOK],0);
    initial;
end;
end;

```

```

procedure TForm21.FormClose(Sender:
TObject; var Action: TCloseAction);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    table1.close;
end;

procedure TForm21.FormShow(Sender:
TObject);
begin
    pagecontrol1.activepage := tabsheet1;
    initial;
end;

end.

unit pastday_ok;

interface

uses
    Windows, Messages, SysUtils, Classes,
    Graphics, Controls, Forms, Dialogs,
    QrCtrls, QuickRpt, ExtCtrls, Db, DBTables,
    StdCtrls, Buttons, Grids,
    DBGrids, ComCtrls;

type
    TForm5 = class(TForm)
        DBGrid1: TDBGrid;
        DataSource1: TDataSource;
        DataSource2: TDataSource;
        MonthCalendar1: TMonthCalendar;
        GroupBox1: TGroupBox;
        SpeedButton1: TSpeedButton;
        SpeedButton2: TSpeedButton;
        SpeedButton3: TSpeedButton;
        SpeedButton4: TSpeedButton;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        Label6: TLabel;
        Table2: TTable;
        Table2ID: TStringField;
        Table2SEX: TStringField;
        Table2NAME: TStringField;
        Table2YEAR: TStringField;
        Table2STATUS: TStringField;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Table2REASON: TStringField;
Table2CODE: TStringField;
Table1: TTable;
Table1ID: TStringField;
Table1NAME: TStringField;
Table1YEAR: TStringField;
PrintDialog1: TPrintDialog;
Table1sex: TStringField;
Table1Date: TDateField;
Table1Com: TStringField;
Table1Login: TTimeField;
Table1Logout: TTimeField;
procedure SpeedButton3Click(Sender:
TObject);
- procedure SpeedButton4Click(Sender:
TObject);
procedure Table1FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
procedure MonthCalendar1Click(Sender:
TObject);
procedure FormActivate(Sender: TObject);
procedure SpeedButton2Click(Sender:
TObject);
procedure SpeedButton1Click(Sender:
TObject);

private
  { Private declarations }
public
  { Public declarations }
check : boolean;

end;
var
Form5: TForm5;
day : string[8];
day_c : string[8];
size : integer;
year : string[3];

implementation

uses view_ok, graph1, main_ok,
pastday_preview;

{$R *.DFM}

procedure TForm5.SpeedButton3Click
(Sender: TObject);
begin
form5.hide;
form2.show;
check := false;
end;

procedure TForm5.SpeedButton4Click
(Sender: TObject);
begin
form5.hide;
form1.show;
check := false;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm5.Table1FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
begin
    day := datetostr(monthcalendar1.date);
    size := length(day);
    year := copy(day,size-1,2);
    if strtoint(year) < 40 then
        begin
            year := inttostr(strtoint(year)+43);
            size := length(year);
            if size > 2 then
                year := copy(year,2,2);
            size := length(day);
            day := copy(day,1,size-2) + year;
        end;
    accept := dataset['date'] = strtodate(day);
end;

```

```

procedure TForm5.MonthCalendar1Click
(Sender: TObject);
var
    i : integer;
    d,m : string[2];
    temp : string[8];
begin
    check := false;
    day := datetostr(monthcalendar1.date);
    size := length(day);
    year := copy(day,size-1,2);
    if strtoint(year) < 40 then
        begin
            year := inttostr(strtoint(year)+43);

```

```

size := length(year);
if size > 2 then
    year := copy(year,2,2);
size := length(day);
day := copy(day,1,size-2) + year;
end;
table1.filtered := true;
label5.caption := day;
table1.close;
table1.open;
form11.check := true;
end;
procedure TForm5.FormActivate(Sender:
TObject);
begin
    day := datetostr(now);
    day_c := day;
    size := length(day);
    year := copy(day,size-1,2);
    if strtoint(year) < 40 then
        begin
            year := inttostr(strtoint(year)+43);
            size := length(year);
            if size > 2 then
                year := copy(year,2,2);
            size := length(day);
            day := copy(day,1,size-2) + year;
        end;
    label5.caption := day;
    monthcalendar1.date := strtodate(day_c);
    table1.close;
    table1.open;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit pastday_preview;

procedure TForm5.SpeedButton2Click
(Sender: TObject);
begin
    form11.qrlabel3.caption :=
label5.caption;
    form11.quickrep1.preview;
end;

```

```

procedure TForm5.SpeedButton1Click
(Sender: TObject);
begin
    if printdialog1.execute then
        form11.print;
end;

end.

```

```

interface

uses
    Windows, Messages, SysUtils, Classes,
    Graphics, Controls, Forms, Dialogs,
    QrCtrls, Db, DBTables, QuickRpt, ExtCtrls,
    StdCtrls;

```

```

type
    TForm11 = class(TForm)
        QuickRep1: TQuickRep;
        DataSource1: TDataSource;
        Table1: TTable;
        Table2: TTable;
        DataSource2: TDataSource;
        PageHeaderBand1: TQRBand;
        TitleBand1: TQRBand;
        ColumnHeaderBand1: TQRBand;
        DetailBand1: TQRBand;
        QRLabel1: TQRLabel;
        QRSysData1: TQRSysData;
        QRLabel2: TQRLabel;
        QRSysData2: TQRSysData;
        QRLabel4: TQRLabel;
        Table1Date: TDateField;
        Table1Login: TTimeField;
        Table1Logout: TTimeField;
        Table1Id: TStringField;
        Table1Com: TStringField;
        Table1name: TStringField;
        Table1year: TStringField;
        Table1sex: TStringField;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

QRLabel5: TQRLabel;
QRLabel6: TQRLabel;
QRLabel7: TQRLabel;
QRLabel8: TQRLabel;
QRLabel9: TQRLabel;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
QRDBText5: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRLabel3: TQRLabel;
Table2Id: TStringField;
Table2Sex: TStringField;
Table2Name: TStringField;
Table2Year: TStringField;
Table2Status: TStringField;
Table2Reason: TStringField;
Table2Code: TStringField;
procedure FormClick(Sender: TObject);
procedure Table1FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
private
  { Private declarations }
public
  { Public declarations }
  check : boolean;
  day_p : string[8];
end;

var
  Form11: TForm11;
  size : integer;
implementation
  uses pastday_ok;
  {$R *.DFM}
  procedure TForm11.Table1FilterRecord
  (DataSet: TDataSet;
  var Accept: Boolean);
  begin
    if check then
      begin
        day := qrlabel3.caption;
        size := length(day);
        year := copy(day,size-1,2);
        if strtoint(year) < 40 then
          begin
            year := inttostr(strtoint(year)+43);
            size := length(year);
            if size > 2 then
              year := copy(year,2,2);
            size := length(day);
            day := copy(day,1,size-2) + year;
          end;
        accept := dataset['date'] = strtodate
          (day);
        end;
      end;
  end;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit show_graph;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, TeEngine, Series, ExtCtrls,
  TeeProcs, Chart, DBChart,
  Db, DBTables, Grids, DBGrids, Mask,
  DBCtrls;

type
  TForm14 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    DBChart1: TDBChart;
    Series1: TBarSeries;
    Series2: TBarSeries;
    Series3: TBarSeries;
    Series4: TBarSeries;
    Series5: TBarSeries;
    Series6: TBarSeries;
    DataSource2: TDataSource;
    Query2: TQuery;
    DataSource3: TDataSource;
    Query3: TQuery;
    DataSource4: TDataSource;
    Query4: TQuery;
    DataSource5: TDataSource;
    Query5: TQuery;
    DataSource6: TDataSource;
    Query6: TQuery;
    GroupBox1: TGroupBox;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
    CheckBox5: TCheckBox;
    Label2: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    Panel1: TPanel;
    CheckBox6: TCheckBox;
    Label1: TLabel;
    Query1COUNT: TIntegerField;
    Query1Year: TStringField;
    Query1Date: TDateField;
    Query2COUNT: TIntegerField;
    Query2Year: TStringField;
    Query2Date: TDateField;
    procedure BitBtn2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender:
    TObject);
    procedure SpeedButton2Click(Sender:
    TObject);
    procedure FormHide(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var
    Action: TCloseAction);
    procedure FormShow(Sender: TObject);
    procedure Query1FilterRecord(DataSet:
    TDataSet; var Accept: Boolean);
    procedure Query2FilterRecord(DataSet:
    TDataSet; var Accept: Boolean);
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    procedure Query3FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
    procedure Query4FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
    procedure Query5FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
    procedure Query6FilterRecord(DataSet:
TDataSet; var Accept: Boolean);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form14: TForm14;

implementation

uses view_ok, main_ok;

{$R *.DFM}

procedure TForm14.BitBtn2Click(Sender:
TObject);
begin
    checkbox1.checked := false;
    checkbox2.checked := false;
    checkbox3.checked := false;
    checkbox4.checked := false;
    checkbox5.checked := false;
    checkbox6.checked := false;
    series1.active := false;
    series2.active := false;
    series3.active := false;
    series4.active := false;
    series5.active := false;
    series6.active := false;
end;

procedure TForm14.SpeedButton1Click
(Sender: TObject);
begin
    form14.hide;
    form2.show;
end;

procedure TForm14.SpeedButton2Click
(Sender: TObject);
begin
    form14.hide;
    form1.show;
end;

procedure TForm14.FormHide(Sender:
TObject);
begin
checkbox1.checked := false;
checkbox2.checked := false;
checkbox3.checked := false;
checkbox4.checked := false;
checkbox5.checked := false;
checkbox6.checked := false;
series1.active := false;
series2.active := false;
series3.active := false;
series4.active := false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

series5.active := false;
series6.active := false;

end;

procedure TForm14.BitBtn1Click(Sender:
TObject);
begin
    series1.Active := checkbox1.checked;
    series2.Active := checkbox2.checked;
    series3.Active := checkbox3.checked;
    series4.Active := checkbox4.checked;
    series5.Active := checkbox5.checked;
    series6.active:=checkbox6.Checked;
end;

procedure TForm14.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
    checkbox1.checked := false;
    checkbox2.checked := false;
    checkbox3.checked := false;
    checkbox4.checked := false;
    checkbox5.checked := false;
    checkbox6.Checked := false;
    series1.active := false;
    series2.active := false;
    series3.active := false;
    series4.active := false;
    series5.active := false;
    series6.active := false;
end;

procedure TForm14.FormShow(Sender:
TObject);
begin
    query1.open;
    query2.Open;
    query3.Open;
    query4.Open;
    query5.open;
    query6.Open;
    checkbox1.checked := false;
    checkbox2.checked := false;
    checkbox3.checked := false;
    checkbox4.checked := false;
    checkbox5.checked := false;
    series1.active := false;
    series2.active := false;
    series3.active := false;
    series4.active := false;
    series5.active := false;
    series6.active := false;
end;

procedure TForm14.Query1FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var
    today,day,temp,start,yar,date : string;
    i,size,month : integer;
    ch : char;
begin
    today := datetostr(now);
    size := length(today);
    yar := copy(today,size-1,2);
    if strtoint(yar) < 40 then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  year := inttostr(strtoint(year)+43);
  size := length(year);
  if size > 2 then
    year := copy(year,2,2);
  size := length(today);
  today := copy(today,1,size-2) + year;
end;
i := pos('/',today);
size := length(today);
day := copy(today,1,i-1);
temp := copy(today,i+1,size-i);
i := pos('/',temp);
size := length(temp);
month := strtoint(copy(temp,1,i-1));
year := copy(temp,i+1,size-i);
start := day + '/' + inttostr(month-1) + '/' +
year;
if month = 1 then
  start := day + '/12/' + inttostr(strtoint
(year)-1);
if month = 12 then
  start := day + '/1/' + inttostr(strtoint
(year)+1);
if query1[count(*)] then
begin
date := datetostr(query1['date']);
accept := strtodate(date) > strtodate
(start);
end;
end;

```

```

procedure TForm14.Query2FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var
today,day,temp,start,year,date : string;
i,size,month : integer;
ch : char;
begin
today := dateostr(now);
size := length(today);
year := copy(today,size-1,2);
if strtoint(year) < 40 then
begin
year := inttostr(strtoint(year)+43);
size := length(year);
if size > 2 then
year := copy(year,2,2);
size := length(today);
today := copy(today,1,size-2) + year;
end;
i := pos('/',today);
size := length(today);
day := copy(today,1,i-1);
temp := copy(today,i+1,size-i);
i := pos('/',temp);
size := length(temp);
month := strtoint(copy(temp,1,i-1));
year := copy(temp,i+1,size-i);
start := day + '/' + inttostr(month-1) + '/' +
year;
if month = 1 then
start := day + '/12/' + inttostr(strtoint
(year)-1);
if month = 12 then
start := day + '/1/' + inttostr(strtoint
(year)+1);
if query1[count(*)] then
begin
date := datetostr(query1['date']);
accept := strtodate(date) > strtodate
(start);
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

start := day + '/1/' + inttostr(strtoint
(year)+1);
if query2['count(*)'] then
begin
date := datetostr(query2['date']);
accept := strtodate(date) > strtodate
(start);
end;
end;

```

```

procedure TForm14.Query3FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var
today,day,temp,start,year,date : string;
i,size,month : integer;
ch : char;
begin
today := datetostr(now);
size := length(today);
year := copy(today,size-1,2);
if strtoint(year) < 40 then
begin
year := inttostr(strtoint(year)+43);
size := length(year);
if size > 2 then
year := copy(year,2,2);
size := length(today);
today := copy(today,1,size-2) + year;
end;
i := pos('/',today);
size := length(today);

```

```

day := copy(today,1,i-1);
temp := copy(today,i+1,size-i);
i := pos('/',temp);
size := length(temp);
month := strtoint(copy(temp,1,i-1));
year := copy(temp,i+1,size-i);
start := day + '/' + inttostr(month-1) + '/' +
year;
if month = 1 then
start := day + '/12/' + inttostr(strtoint
(year)-1);
if month = 12 then
start := day + '/1/' + inttostr(strtoint
(year)+1);
if query3['count(*)'] > 0 then
begin
date := datetostr(query3['date']);
accept := strtodate(date) > strtodate
(start);
end;
end;

```

```

procedure TForm14.Query4FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var
today,day,temp,start,year,date : string;
i,size,month : integer;
ch : char;
begin
today := datetostr(now);
size := length(today);
year := copy(today,size-1,2);
if strtoint(year) < 40 then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  year := inttostr(strtoint(year)+43);
  size := length(year);
  if size > 2 then
    year := copy(year,2,2);
  size := length(today);
  today := copy(today,1,size-2) + year;
end;
i := pos('/',today);
size := length(today);
day := copy(today,1,i-1);
temp := copy(today,i+1,size-i);
i := pos('.',temp);
size := length(temp);
month := strtoint(copy(temp,1,i-1));
year := copy(temp,i+1,size-i);
start := day + '/' + inttostr(month-1) + '/' +
year;
if month = 1 then
  start := day + '/12/' + inttostr(strtoint
(year)-1);
if month = 12 then
  start := day + '/1/' + inttostr(strtoint
(year)+1);
if query4['count(*)'] > 0 then
begin
  date := datetostr(query4['date']);
  accept := strtodate(date) > strtodate
(start);
end;
end;
end;

```

```

procedure TForm14.Query5FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var
  today,day,temp,start,year,date : string;
  i,size,month : integer;
  ch : char;
begin
  today := datetostr(now);
  size := length(today);
  year := copy(today,size-1,2);
  if strtoint(year) < 40 then
  begin
    year := inttostr(strtoint(year)+43);
    size := length(year);
    if size > 2 then
      year := copy(year,2,2);
    size := length(today);
    today := copy(today,1,size-2) + year;
  end;
  i := pos('/',today);
  size := length(today);
  day := copy(today,1,i-1);
  temp := copy(today,i+1,size-i);
  i := pos('.',temp);
  size := length(temp);
  month := strtoint(copy(temp,1,i-1));
  year := copy(temp,i+1,size-i);
  start := day + '/' + inttostr(month-1) + '/' +
year;
if month = 1 then
  start := day + '/12/' + inttostr(strtoint
(year)-1);
if month = 12 then
  start := day + '/1/' + inttostr(strtoint
(year)+1);
if query5['count(*)'] > 0 then
begin
  date := datetostr(query5['date']);
  accept := strtodate(date) > strtodate
(start);
end;
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    start := day + '/1/' + inttostr(strtoint
(year)+1);
    if query5['count(*)'] then
    begin
        date := datetostr(query5['date']);
        accept := strtodate(date) > strtodate
(start);
    end;
end;

```

```

    day := copy(today,1,i-1);
    temp := copy(today,i+1,size-i);
    i := pos('/',temp);
    size := length(temp);
    month := strtoint(copy(temp,1,i-1));
    year := copy(temp,i+1,size-i);
    start := day + '/' + inttostr(month-1) + '/' +
year;
    if month = 1 then
        start := day + '/12/' + inttostr(strtoint
(year)-1);
    if month = 12 then
        start := day + '/1/' + inttostr(strtoint
(year)+1);
    if query6['count(*)'] then
    begin
        date := datetostr(query6['date']);
        accept := strtodate(date) > strtodate
(start);
    end;
end;
end;

```

```

procedure TForm14.Query6FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var
    today,day,temp,start,year,date : string;
    i,size,month : integer;
    ch : char;
begin
    today := datetostr(now);
    size := length(today);
    year := copy(today,size-1,2);
    if strtoint(year) < 40 then
    begin
        year := inttostr(strtoint(year)+43);
        size := length(year);
        if size > 2 then
            year := copy(year,2,2);
        size := length(today);
        today := copy(today,1,size-2) + year;
    end;
    i := pos('/',today);
    size := length(today);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit UpLoad;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, Db, DBTables,
  Grids, DBGrids, ComCtrls,
  Comms;

type
  TForm16 = class(TForm)
    TabControl1: TTabControl;
    DBGrid1: TDBGrid;
    GroupBox1: TGroupBox;
    SpeedButton1: TSpeedButton;
    Label1: TLabel;
    SpeedButton2: TSpeedButton;
    Label2: TLabel;
    SpeedButton3: TSpeedButton;
    Label3: TLabel;
    SpeedButton4: TSpeedButton;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    DataSource1: TDataSource;
    ComPort: TComPort;
    SpeedButton5: TSpeedButton;
    Label7: TLabel;
    StringGrid1: TStringGrid;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    StaticText3: TStaticText;
    StaticText4: TStaticText;
    StaticText5: TStaticText;
    DataSource2: TDataSource;
    Table2: TTable;
    Table1: TTable;
    procedure SpeedButton1Click(Sender:
      TObject);
    procedure ComportOpen(Sender:
      TObject);
    procedure ComportClose(Sender:
      TObject);
    procedure error;
    procedure Convert( var x : integer; y :
      integer);
    procedure ConvertBi( var x : integer);
    procedure CheckBreak( ch : char ; var
      check : boolean);
    procedure Field;
    procedure MakeList;
    procedure Initial;
    procedure ComportRxChar(Sender:
      TObject: InQue: Integer );
    procedure SpeedButton5Click(Sender:
      TObject);
    procedure SpeedButton4Click(Sender:
      TObject);
    procedure SpeedButton2Click(Sender:
      TObject);
    procedure FormShow(Sender: TObject);
    procedure SpeedButton3Click(Sender:
      TObject);
  private
    { Private declarations }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public
    { Public declarations }
end;

var
    Form16: TForm16;
    row,column,q : integer;
    source : string;

    label1.caption := 'ยกเลิกการติดต่อพอร์ต';
    comport.WriteString('90',true); // ส่ง
    สัญญาณขอให้ 51 ส่งข้อมูล
    form16.enabled := false;
    form17.form := 'up';
    form17.show;
end;

procedure TForm16.ComportClose (Sender:
TObject );
begin
    label1.caption := 'ติดต่อพอร์ต';
end;

procedure TForm16.error;
begin
    if messagedlg('การส่งข้อมูลผิดพลาด
ต้องการส่งใหม่',mterror,[mbOK,MbCancel],0 )
= mrCancel then
    begin
        initial;
        form16.enabled := true;
        form19.Close;
    end
    else
    begin
        initial;
        comport.writestring('90',true); // ส่ง
        สัญญาณขอให้ 51 ส่งข้อมูลใหม่
        form16.enabled := false;
        form19.close;
        form17.show;
    end;
end;

procedure TForm16.ComportOpen (Sender:
TObject );
begin
    ($R *.DFM)

    procedure TForm16.SpeedButton1Click
    (Sender: TObject);
    begin
        if Comport.Connected then
        begin
            Comport.Close;
            source := "";
        end
        else
            Comport.Open;
        end;

    procedure TForm16.ComportOpen (Sender:
TObject );
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm16.Convert ( var x : integer;
y : integer);
var
  str : string;
  size,i : integer;
begin
  str := inttostr(x);
  x := 0;
  size := length(str);
  while size < y do
  begin
    str := '0' + str;
    size := size+1;
  end;
  for i := 1 to y do
    x := strtoint(copy(str,i,1)) + x + 48; //
convert to decimal
end;

```

```

procedure TForm16.ConvertBi (var x :
integer);
var
  ans1,ans2,size : integer;
  ans : string;
begin
  ans := "";
  while x <> 0 do
  begin
    ans1 := x div 2;
    ans2 := x mod 2;
    ans := inttostr(ans2) + ans;
    x := ans1;
  end;
  size := length(ans);

```

```

  if size > 7 then
    x := strtoint(copy(ans,size-6,7))
  else
    x := strtoint(ans);
end;

```

```

procedure TForm16.CheckBreak ( ch : char ;
var check : boolean);
begin
  if ch <> '#' then
  begin
    error;
    check := false;
  end
  else
    check := true;
end;

```

```

procedure TForm16.Field;
const
  start = 'a';
  stop = 'z';
var
  data,temp : string;
  day,month,year,hr,min,sec,com : string[2];
  code : string[3];
  date,login,logout : string[8];

  code2,day2,month2,year2,hr2,min2,sec2,com
2 : integer;
  line,i,j,num,sum51 : integer;
  ch : char;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

check_all := true;
row := 1;
column := 0;
num := 0;
line := 0;
data := form18.memo2.lines[line];
while (data <> "") and check_all do
begin
temp := copy(data,1,1);
if temp <> start then
check_all := false
else
begin
i := 2;
code := copy(data,i,3);
code2 := strtoint(code);
convert(code2,3);
i := i+3;
ch := data[i];
i := i+1;
CheckBreak(ch,check_all);
if check_all then
begin
day := copy(data,i,2);
day2 := strtoint(day);
convert(day2,2);
ch := day[1];
if ch = '0' then
day := day[2];
month := copy(data,i+2,2);
month2 := strtoint(month);
convert(month2,2);
ch := month[1];
if ch = '0' then
month := month[2];
year := copy(data,i+4,2);
year2 := strtoint(year);
convert(year2,2);
date := day + '/' + month + '/'
+ year;
i := i+6;
ch := data[i];
CheckBreak(ch,check_all);
i := i+1;
num := code2 + day2 +
month2 + year2 ;
for j := 1 to 2 do
begin
if check_all then
begin
hr := copy(data,i,2);
hr2 := strtoint(hr);
convert(hr2,2);
min := copy(data,i+2,2);
min2 := strtoint(min);
convert(min2,2);
sec := copy(data,i+4,2);
sec2 := strtoint(sec);
convert(sec2,2);
i := i+6;
num := num + hr2 +
min2 + sec2;
end;
ch := data[i];
checkbreak(ch,check_all);
if check_all then
begin
i := i+1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

end;
form18.memo1.clear;
form18.memo2.clear;
end;

procedure TForm16.MakeList ;
var
size1,size2,i,j : integer;
source_temp : string;

begin
j := 1;
size1 := (length(source)-1) div 31;
size2 := (length(source)-1) mod 31;
if (size2 <> 0) or (size1 = 0) then
messedlg('การส่งข้อมูลผิดพลาด',mterror,[mbOK],0)
else
begin
for i:= 1 to size1 do
begin
source_temp := copy(source,j,31);
form18.memo2.Lines.add
(source_temp);
j := j + 31;
end;
field:
end;
end;

procedure TForm16.Initial; // เมื่อสิ้นสุดการส่ง
โน้ตแต่ละครั้ง
begin
check_start := false;
check_data := false;
check_stop := false;

form18.memo1.clear;
form18.memo2.clear;
end;

procedure TForm16.ComportRxChar(Sender:
TObject; InQue: Integer);
const
free = 'bf';
var
temp : string[2];
Str,temp1 : String;
size : integer;
stop : char;
begin
comport.ReadString (str,inque,true);
temp1 := 'aa';
stop := 'a';
if str <> '' then
begin
size := length(str);
temp1 := copy(str,size-1,2);
stop := str[1];
end;
if not check_start then
begin
if temp1 = free then
begin
comport.WriteString('U',true); // ส่ง
สัญญาณพร้อมรับข้อมูล
check_start := true;
source := "";
check_all := false;
form17.close;
form19.form := 'up';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

form19.show; // แสดงหน้าจอ "กำลังส่ง
ข้อมูล"
end ;
{else
begin
form16.enabled := true;
form17.Close;
if messagedlg('การติดต่อผิดพลาด
ต้องการติดต่อใหม่',mterror,[mbOk,mbCancel],
0) = mrCancel then
comport.close
else
begin
initial;
comport.writestring('90',true); //
ส่งสัญญาณขอให้ 51 ส่งข้อมูลใหม่
form16.enabled := false;
form19.close;
form17.show;
end;
end; }
end
else // ส่วนการรับข้อมูล
begin
if not check_stop then
begin
if source = " then
begin
if str <> " then
begin
q := strtoint(copy(str,1,3)); // บอกร
จำนวน user
if q = 0 then
begin
messagedlg('ไม่มีข้อมูลใน
หน่วยความจำ',mtinformation,[mbOK],0);
comport.writestring('E',true);
form19.hide;
end
else
begin
str := copy(str,4,length(str)-3);
form18.memo1.text :=
form18.memo1.text + str;
source := form18.memo1.text;
end;
end;
else
begin
if str <> " then
begin
form18.memo1.text :=
form18.memo1.text + str;
source := form18.memo1.text;
size := length(source);
temp := copy(source,size-1,1);
if size > (q*31) then
begin
temp := copy(source,
(q*31)+1,2);
if temp = 'E' then
begin
check_stop := true;
form19.close;
form16.enabled := true;
initial;
makelist;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

error;
check_all := true;
end;
end
else
begin
id := table2.fieldvalues['id'];
table2.close;
with StringGrid1 do
begin
date := cells[1,i];
login := cells[2,i];
logout := cells[3,i];
com := cells[4,i];
end;
table1.open;
table1.append;
table1['id'] := id;
table1['date'] := strtodate(date);
table1['login'] := strtotime(login);
table1['logout'] := strtotime(logout);
table1['com'] := com;
table1.post;
end;
end;
end;

procedure TForm16.SpeedButton3Click
(Sender: TObject);
var
ij : integer;
begin
for i := 1 to row-1 do
for j := 0 to 4 do
with StringGrid1 do
cells[j,i] := '';
end;
end;
end;

procedure TForm16.FormShow(Sender:
TObject);
begin
table1.open;
table2.open;
initial;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit UpLoad_Send_data;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls;

type
  TForm19 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    BitBtn1: TBitBtn;
    Image1: TImage;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    form : string[5];
  end;

var
  Form19: TForm19;

implementation

uses edit_ok, UpLoad, DownLoad;

{$R *.DFM}

procedure TForm19.BitBtn1Click(Sender:
TObject);
begin
  if form = 'up' then
  begin
    form19.close;
    form16.enabled := true;
  end
  else
  begin
    if form = 'edit' then
    begin
      form19.close;
      form3.enabled := true;
    end
    else
    begin
      form19.close;
      form20.enabled := true;
    end;
  end;
end;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit UpLoad_Send_Status;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls;

type
  TForm17 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    BitBtn1: TBitBtn;
    Image1: TImage;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    form : string[5];
  end;

var
  Form17: TForm17;

```

```

procedure TForm17.BitBtn1Click(Sender:
TObject);
begin
  if form = 'up' then
  begin
    form17.close;
    form16.enabled := true;
    form16.ComPort.close;
  end
  else
  if form = 'down' then
  begin
    form17.close;
    form20.enabled := true;
    form20.comport1.close;
  end
  else
  if form = 'edit' then
  begin
    form17.close;
    form3.enabled := true;
  end;
end;
end.

```

implementation

uses UpLoad, DownLoad, edit\_ok;

{\$R \*.DFM}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit view_ok;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, DBCtrls, Mask, Db, DBTables,
  ComCtrls, ExtCtrls, Grids,
  DBGrids, Buttons;

type
  TForm2 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    DataSource1: TDataSource;
    Table1: TTable;
    DBNavigator1: TDBNavigator;
    GroupBox1: TGroupBox;
    Panel1: TPanel;
    Panel2: TPanel;
    Edit1: TEdit;
    Edit2: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    GroupBox2: TGroupBox;
    DBGrid1: TDBGrid;
    BitBtn3: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;

    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    BitBtn4: TBitBtn;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    SpeedButton5: TSpeedButton;
    Label10: TLabel;
    Label11: TLabel;
    DBGrid2: TDBGrid;
    SpeedButton6: TSpeedButton;
    Label12: TLabel;
    DataSource2: TDataSource;
    Table2: TTable;
    DBEdit6: TDBEdit;
    Table1ID: TStringField;
    Table1SEX: TStringField;
    Table1NAME: TStringField;
    Table1YEAR: TStringField;
    Table1STATUS: TStringField;
    Table1REASON: TStringField;
    Table1CODE: TStringField;
    Table2Id: TStringField;
    Table2Sex: TStringField;
    Table2Name: TStringField;
    Table2Year: TStringField;
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Table2Status: TStringField;           { Private declarations }
Table2Reason: TStringField;          public
Table2Code: TStringField;            { Public declarations }
procedure Button1Click(Sender: TObject); end;
procedure TabSheet1DragDrop(Sender,
Source: TObject; X, Y: Integer);      var
    procedure PageControl1Change(Sender:
TObject);                             Form2: TForm2;
    procedure TabSheet2DragDrop(Sender,
Source: TObject; X, Y: Integer);      click_name,click_id,result,result_n,result_id :
                                        boolean;
    procedure BitBtn3Click(Sender: TObject); showmess : boolean;
    procedure Panel1Click(Sender: TObject); implementation
    procedure Panel2Click(Sender: TObject); uses main_ok, oneday_ok, pastday_ok,
    procedure BitBtn1Click(Sender: TObject); notclose_ok, cannotuse_ok, graph1
    procedure BitBtn2Click(Sender: TObject); {,report,report_unuse};
    procedure BitBtn4Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject); {$R *.DFM}
    procedure SpeedButton5Click(Sender: TObject); procedure TForm2.Button1Click(Sender:
TObject); {back to Main}
    procedure SpeedButton4Click(Sender: TObject); begin
    procedure SpeedButton2Click(Sender: TObject); table1.cancel;
    procedure SpeedButton3Click(Sender: TObject); table1.first;
    procedure SpeedButton6Click(Sender: TObject); form2.hide;
    procedure Table2FilterRecord(DataSet: TDataSet; var Accept: Boolean); form1.show;
    procedure FormClose(Sender: TObject; var Action: TCloseAction); end;
    procedure FormShow(Sender: TObject); procedure TForm2.TabSheet1DragDrop
(Sender, Source: TObject; X, Y: Integer); begin
    private                               table1.Open;
                                        table1.Refresh;
                                        table1.First;
                                        edit1.clear;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

edit1.hide;
edit1.hide;
edit2.hide;
end;

procedure TForm2.PageControl1Change
(Sender: TObject);
begin
    table1.Open;
    table1.Refresh;
    table1.First;
    edit1.clear;
    edit1.hide;
    edit2.clear;
    edit2.Hide;
    dbedit1.datasource := datasource1;
    dbedit2.datasource := datasource1;
    dbedit3.datasource := datasource1;
    dbedit4.datasource := datasource1;
    dbedit5.datasource := datasource1;
    dbedit6.datasource := datasource1;
end;

procedure TForm2.TabSheet2DragDrop
(Sender, Source: TObject; X, Y: Integer);
begin
    table1.Open;
    table1.Refresh;
    table1.First;
end;

procedure TForm2.BitBtn3Click(Sender:
TObject); {back to main}
begin
    edit1.clear;
    edit1.hide;
    edit2.clear;
    edit2.hide;
    table1.cancel;
    table1.first;
    form2.hide;
    form1.show;
end;

procedure TForm2.Panel1Click(Sender:
TObject); {panel name click}
begin
    edit1.show;
    click_name := true;
end;

procedure TForm2.Panel2Click(Sender:
TObject); {panel id click}
begin
    edit2.show;
    click_id := true;
end;

procedure TForm2.BitBtn1Click(Sender:
TObject); {button search click}
begin
    dbedit1.datasource := datasource1;
    dbedit2.datasource := datasource1;
    dbedit3.datasource := datasource1;
    dbedit4.datasource := datasource1;
    dbedit5.datasource := datasource1;
    dbedit6.datasource := datasource1;
    showmess := false;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if click_name then
begin
    if edit1.text = "" then
        click_name := false
    else
        begin
            table1.indexname := 'idxname';
            table1.setkey;
            table1.FieldValues['name'] :=
edit1.text;
            result_n := table1.gotokey;
            if not result_n then
                begin
                    table2.close;
                    table2.open;
                    dbedit1.DataSource :=
datasource2;
                    dbedit2.datasource :=
datasource2;
                    dbedit3.datasource :=
datasource2;
                    dbedit4.datasource :=
datasource2;
                    dbedit5.datasource :=
datasource2;
                    dbedit6.datasource :=
datasource2;
                    result_n := true;
                    {messagedlg('ไม่พบข้อมูลในฐาน
ข้อมูล',mtinformation,[mbok],0); }
                    messagedlg('สามารถดูข้อมูลใกล้เคียง
ได้จากตารางข้อมูลใกล้เคียง',mtcustom,
[mbok],0);
                    showmess := true;
                end;
            end;
        end;
    end;
end;

dbgrid1.hide;
dbgrid2.show;
label11.show;

end;

if click_id then
begin
    if edit2.text = "" then
        click_id := false ;
        table1.indexname := 'idxid';
        table1.SetKey;
        table1.FieldValues['id'] := edit2.text;
        result_id := table1.gotokey;
        end;
        if click_name and click_id then
            begin
                result := result_n and result_id;
                result_n := result_n and not
(click_id);
                result_id := result_id and not
(click_name);
                if (not (result or result_n or result_id))
and not showmess then
                    begin
                        messagedlg('ไม่พบข้อมูลในฐานข้อ
มูล',mtinformation,[mbok],0);
                        messagedlg('สามารถดูข้อมูลใกล้เคียง
ได้จากตารางข้อมูลใกล้เคียง',mtcustom,
[mbok],0);
                        dbgrid1.hide;
                        dbgrid2.show;
                        label11.show;
                        {clgrid;}
                    end;
            end;
        end;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end
    end
    else
        if click_id then
            if not result_id then
                begin
                    messagedlg('ไม่พบข้อมูลในฐานข้อมูล', mtInformation, [mbok], 0);
                    dbgrid1.show;
                    dbgrid2.hide;
                    label11.Hide;
                    (clgrid);
                end
            end;
        procedure TForm2.BitBtn2Click(Sender:
        TObject): {cancel search }
        begin
            table1.first;
            edit1.clear;
            edit1.hide;
            edit2.clear;
            edit2.hide;
            click_name := false;
            click_id := false;
            dbgrid1.show;
            dbgrid2.hide;
            label11.hide;

            edit1.clear;
            edit1.hide;
            edit2.clear;
            edit2.hide;
            dbgrid1.show;
            dbgrid2.hide;
            label11.hide;
            click_name := false;
            click_id := false;
        end;
        procedure TForm2.SpeedButton1Click
        (Sender: TObject);
        begin
            form2.hide;
            form4.show;{data in one day}
            {table1.filtered := true;
            table1.filter := 'date=30/8/42';}
        end;
        procedure TForm2.SpeedButton5Click
        (Sender: TObject);
        begin
            form2.Hide;
            form1.show;{back to main}
        end;
        procedure TForm2.SpeedButton4Click
        (Sender: TObject);
        begin
            form2.hide;
            form7.show;{data can not use}
        end;
    end;
    procedure TForm2.BitBtn4Click(Sender:
    TObject);
    begin
        table1.first;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm2.SpeedButton2Click
(Sender: TObject);
begin
    form2.hide;
    form5.show;{data in past}
end;

procedure TForm2.SpeedButton3Click
(Sender: TObject);
begin
    form2.hide;
    form6.show;{data not close}
end;

procedure TForm2.SpeedButton6Click
(Sender: TObject);
begin
    form2.hide;
    form8.show;
end;

procedure TForm2.Table2FilterRecord
(DataSet: TDataSet;
var Accept: Boolean);
var index : string[2];
begin
    index := copy(edit1.text,1,2);
    accept := copy(dataset['name'],1,2) =
index;
end;

procedure TForm2.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
    dbedit1.datasource := datasource1;
    dbedit2.datasource := datasource1;
    dbedit3.datasource := datasource1;
    dbedit4.datasource := datasource1;
    dbedit5.datasource := datasource1;
    dbedit6.datasource := datasource1;
end;

procedure TForm2.FormShow(Sender:
TObject);
begin
    table1.Open;
    table1.Refresh;
    table1.First;
    edit1.clear;
    edit1.hide;
    edit2.clear;
    edit2.hide;
    dbgrid1.show;
    dbgrid2.hide;
    label11.hide;
    if form4.tabsheet = 1 then
        pagecontrol1.activepage := tabsheet1
    else
        pagecontrol1.activepage := tabsheet2;
end;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้