



ระบบสารสนเทศของสำนักทะเบียนและประมวลผล  
The Office of The Register's Information System

โดย

นายจตุรนต์ กาญจนการุณ เลขประจำตัว 36014069  
นางสาวนุชนาฏ ตันติไพบุลย์วงศ์ เลขประจำตัว 36014228

วัน เดือน ปี - 1. ตุลาคม 2541  
เลขทะเบียน..... 038317  
เลขเรียกหนังสือ: T. 99. 89. 7. ๑๙๖๕

อาจารย์ที่ปรึกษา

อาจารย์วิบูลย์ พร้อมพานิชย์

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการ  
038317

ปริญญาานิพนธ์ปีการศึกษา 2539

ภาควิชา วิศวกรรมศาสตร์คอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสารสนเทศของสำนักทะเบียนและประมวลผล

ผู้จัดทำ

1. นายจตุรงค์ กาญจนการุณ รหัสประจำตัวนักศึกษา 36014069
2. นางสาวนุชนาฏ ตันติไพบุลย์วงศ์ รหัสประจำตัวนักศึกษา 36014228

  
.....อาจารย์ที่ปรึกษา  
( อาจารย์วิบูลย์ พร้อมพานิชย์ )

## ระบบสารสนเทศของสำนักทะเบียนกลางและประมวลผล

นายจตุรงค์ กาญจนการุณ  
น.ส.นุชนาฏ ดันติไพบูลย์วงศ์  
อาจารย์วิบูลย์ พร้อมพานิชย์ อาจารย์ที่ปรึกษา  
ปีการศึกษา 2539

### บทคัดย่อ

โครงการนี้ได้จัดทำขึ้นเพื่อพัฒนาโครงการ "ระบบสารสนเทศของสำนักทะเบียนและประมวลผล" ที่มีอยู่เดิมให้ดีขึ้น โดยใช้สถาปัตยกรรมทรีเทียร์ ซึ่งได้แบ่งการทำงานออกเป็น 3 ส่วน คือ ส่วนที่ติดต่อกับผู้ใช้, ส่วนที่เก็บข้อจำกัดของผู้ใช้งาน และส่วนที่เก็บและบริการข้อมูล โดยในสถาปัตยกรรมไคลเอ็นต์/เซิร์ฟเวอร์แบบเดิม ส่วนที่เก็บข้อจำกัดของผู้ใช้งานจะอยู่ในส่วนของไคลเอ็นต์ เมื่อมีการเปลี่ยนแปลงกฎเกณฑ์ของระบบต้องตามไปเปลี่ยนแปลงกฎเกณฑ์ที่เครื่องไคลเอ็นต์ทุกเครื่องเสมอไป แต่ในสถาปัตยกรรมแบบทรีเทียร์ได้มีการแยกส่วนนี้ออกมา ทำให้ผู้ใช้สามารถเปลี่ยนแปลงแก้ไขกฎเกณฑ์ต่างๆ ที่ส่วนกลางนี้ได้เลยโดยไม่ต้องยุ่งกับส่วนอื่นๆ ช่วยให้การดำเนินงานสะดวกขึ้น ทั้งการบำรุงรักษา, การเปลี่ยนแปลงกฎเกณฑ์ใหม่ และการพัฒนาระบบใหม่

ในการพัฒนาโครงการนี้ ได้ใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ของบริษัทไมโครซอฟท์ ที่ชื่อว่า "ไมโครซอฟท์ เอสคิวแอล เซิร์ฟเวอร์ เวอร์ชัน 6.5" เป็นส่วนที่เก็บและให้บริการข้อมูล และได้ใช้ "วิซวลเบสิก เวอร์ชัน 4.0 เอนเทอร์ไพรส์ อิดิชั่น" ในการเขียนโปรแกรมสำหรับส่วนที่ติดต่อกับผู้ใช้ รวมทั้งส่วนที่เก็บข้อจำกัดของผู้ใช้งาน โดยมีโครฟเวอร์โอคิบีซีเป็นโครฟเวอร์ที่ทำหน้าที่ติดต่อระหว่างส่วนแอปพลิเคชันและส่วนที่เก็บข้อมูล ซึ่งระบบนี้ทำงานภายใต้ระบบเครือข่ายท้องถิ่น (LAN)

# The Office of The Register's Information System

Jaturont Kanjanakaroon  
Nudchanard Tantiphaiboonwong  
Wiboon Prompanich Advisor  
1996

## Abstract

This project is the development version of the old project "The Office of The Register's Information System" which used Three-Tier Architecture, the new technology in Client/Server Programming today to design and develop the new system of the Registry by separated the system to 3 parts or 3 tiers : User Service, Business Service and Data Service which in the architecture of the old project, business service is in client part then when the business rule had changed, hundreds, sometimes thousands, of client workstation need to be upgrades with a new version of the front-end simultaneously, but in the new architecture which separated business service from client make an easy change to business rule, easy maintenance and easy to make new system.

During the implementation, this project used Microsoft® SQL Server™ Version 6.5 to be a RDBMS (Relation Database Management System), used Visual Basic® Enterprise Edition Version 4.0 for graphic user interface (GUI) and business service and used ODBC Driver (Open Database Connectivity) for connect between Client Application and Data Service and this system will be run on LAN (Local Area Network).

## สารบัญ

บทคัดย่อ		หน้า
บทที่ 1 บทนำ		1
บทที่ 2 ทฤษฎีและหลักการ		2
2.1	สถาปัตยกรรมแบบทิวเทียร์	2
2.2	สถาปัตยกรรมแบบทรีเทียร์	3
2.3	สถาปัตยกรรมเทียร์ที่หนึ่ง	5
2.3.1	กฎในการออกแบบส่วนบริการผู้ใช้	5
2.3.2	เครื่องมือที่ใช้ในการพัฒนาส่วนบริการผู้ใช้	5
2.3.3	การติดต่อระหว่างส่วนบริการผู้ใช้และ บิสซิเนส เซอร์วิส	5
2.2.4	วิธีการพัฒนาในส่วนของไคลเอ็นต์ แอปพลิเคชัน	7
2.4	สถาปัตยกรรมเทียร์ที่สอง	7
2.4.1	การออกแบบส่วนบิสซิเนส	8
2.4.2	การรวบรวมกฎเกณฑ์ข้อจำกัด	10
2.4.3	ข้อดีในการแยกส่วนบริการกฎเกณฑ์ข้อจำกัดออกจาก ส่วนไคลเอ็นต์	11
2.4.4	การทำงานกับส่วนบริการข้อมูล	11
2.4.4.1	การติดต่อกับส่วนบริการข้อมูลเชิงสัมพันธ์	12
2.4.4.2	รีโมท คาด้า ออบเจกต์	13
2.5	โอเพ่นคาด้าเบสคอนเนคตีวีดี (โอดีบีซี)	14
2.5.1	ส่วนประกอบของโอดีบีซี	15
2.5.2	ส่วนประกอบในการอินเตอร์เฟสกับโอดีบีซี	16
2.5.3	แอปพลิเคชัน	16
2.5.4	ไคร์ฟเวอร์แมนเนเจอร์	17
2.5.5	ไคร์ฟเวอร์	17
2.5.6	ชนิดของไคร์ฟเวอร์	17
2.5.7	ระดับข้อตกลงของโอดีบีซี	18
2.5.8	ข้อดีในการอินเตอร์เฟสกับโอดีบีซี	18

2.6	สถาปัตยกรรมเครือข่ายที่สาม	19
2.6.1	ระบบจัดการฐานข้อมูลส่วนบริการฐานข้อมูล	19
2.6.2	ฐานข้อมูลเชิงสัมพันธ์	19
2.62.1	ความสัมพันธ์แบบหนึ่ง ต่อ หนึ่ง	20
2.62.2	ความสัมพันธ์แบบหนึ่ง ต่อ กลุ่ม	20
2.62.3	ความสัมพันธ์แบบกลุ่ม ต่อ กลุ่ม	21
2.6.3	ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์	21
2.6.4	ฟังก์ชันการขึ้นต่อกัน	23
2.6.5	การออกแบบฐานข้อมูลรวมเชิงสัมพันธ์	24
2.6.6	การเขียนโปรแกรมส่วนการค้า เซอร์วิส โดยใช้ สตอร์ โพรซีเยอร์	25
2.7	ประโยชน์ของสถาปัตยกรรมแบบทรีเทียร์	27
2.8	การใช้วิซวลเบสิกในการพัฒนาสถาปัตยกรรมแบบทรีเทียร์	30
2.8.1	การใช้วิซวลเบสิกพัฒนาแอปพลิเคชันบนวินโดวส์ สำหรับส่วนการให้บริการผู้ใช้	31
2.8.2	การใช้วิซวลเบสิกพัฒนาแอปพลิเคชันบนวินโดวส์ สำหรับส่วนการให้บริการข้อมูล	32
2.8.3	ประโยชน์ของการใช้วิซวลเบสิกในการพัฒนาโปรแกรม ด้านฐานข้อมูล	33
2.9	โอเล่ (ออบเจกต์ ลิงค์กิง แอนด์ เอ็มเบดเดด)	33
2.9.1	โอเล่ ออบเจกต์	34
2.9.2	โอเล่ เซอร์วิส	35
2.9.3	โอเล่ ออโตเมชัน เซิร์ฟเวอร์	35
2.9.4	การควบคุม โอเล่ ออโตเมชัน ออบเจกต์	37
2.9.5	รีโมท ออโตเมชัน	39
2.9.6	โอเล่ คอนโทรล	40
<b>บทที่ 3</b>	<b>การดำเนินและพัฒนาระบบ</b>	<b>41</b>
3.1	การดำเนินงานในปัจจุบัน	41
3.1.1	ลักษณะของการทำงาน	41
3.1.2	ปัญหาในการทำงาน	43

3.2	วัตถุประสงค์ของโครงการ	44
3.3	ขอบเขตของโครงการ	44
3.3.1	ภาพรวมของระบบ	45
3.3.2	ความสามารถของระบบ	46
3.3.3	คุณสมบัติของระบบ	46
3.4	หลักการและวิธีการ	46
3.5	แผนงานในการพัฒนาระบบ	47
3.5.1	แผนงานในการพัฒนาระบบภาคเรียนที่ 1	47
3.5.2	แผนงานในการพัฒนาระบบภาคเรียนที่ 2	48
3.6	การพัฒนาระบบ	49
3.6.1	ความต้องการด้านฮาร์ดแวร์	49
3.6.2	ความต้องการด้านซอฟต์แวร์	52
3.7	รายละเอียดการทำงานของโครงการ	52
3.7.1	ระบบทะเบียนประวัตินักศึกษา	52
3.7.1.1	การทำงานของระบบ	54
3.7.1.2	การแก้ไขข้อมูลนิสิต	54
3.7.1.3	การฟื้นฟูสภาพนิสิต	54
3.7.1.4	รายงานที่ได้จากระบบ	55
3.7.2	ระบบข้อมูลหลักสูตรการศึกษา	55
3.7.2.1	การทำงานของระบบ	55
3.7.2.2	การแก้ไขข้อมูลเกี่ยวกับรายวิชาในระบบ	55
3.7.2.3	การยกเลิกวิชาในระบบ	56
3.7.2.4	รายงานที่ได้จากระบบ	56
3.8	การออกแบบระบบในส่วนของเทียร์ที่ 1	56
3.9	การออกแบบส่วนบิสซิเนส เซอร์วิส	66
3.10	การออกแบบส่วนออบเจกต์ อินเตอร์เฟส ของบิสซิเนส เซอร์วิส	68
3.11	การออกแบบส่วนดาต้า เซอร์วิส	69
3.12	การติดต่อกับผู้ใช้	71

<b>บทที่ 4 การทดลองและผลการทดลอง</b>	<b>73</b>
4.1 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการทดลอง	73
4.1.1 คอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบเครือข่าย	73
4.1.2 เครื่องไมโครคอมพิวเตอร์ที่ทำหน้าที่เป็นเทอร์มินอล ในระบบเครือข่ายจำนวน 2 เครื่อง	73
4.2 ความต้องการด้านฮาร์ดแวร์	74
4.2.1 ระบบปฏิบัติการที่ทำหน้าที่เป็นเซิร์ฟเวอร์	74
4.2.2 ระบบปฏิบัติการที่ทำหน้าที่เป็นเวิร์คสเตชัน	74
4.3 การทดลองและผลการทดลอง	74
4.3.1 ระบบการนำเข้าข้อมูลของประวัตินักศึกษาและข้อมูล โครงสร้างการศึกษา	74
<b>บทที่ 5 บทวิจารณ์และสรุป</b>	<b>84</b>
5.1 ความสามารถของโปรแกรม	84
5.2 ข้อจำกัดของ โปรแกรม	84
5.3 ปัญหาในการทำงาน	84
5.4 แนวทางในการพัฒนาต่อไป	85

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

## สารบัญรูปภาพ

รูปภาพ		หน้าที่
รูปที่ 2.1	แสดงเครื่องมือที่ใช้และกลไกที่ใช้ในการติดต่อแต่ละชั้นของเทียร์	4
รูปที่ 2.2	แสดงกลุ่มของส่วนประกอบ	8
รูปที่ 2.3	แสดงการจัดแบ่งเป็นกลุ่มของส่วนประกอบ	9
รูปที่ 2.4	แสดงตัวอย่างแต่ละชั้นของบิสซิเนส	10
รูปที่ 2.5	แสดงลำดับชั้นของ RDO ออบเจ็ค	13
รูปที่ 2.6	โครงสร้างแต่ละชั้นของ ODBC	15
รูปที่ 2.7	แสดงตัวอย่างการใช้งานจริงทีเรียร์ในองค์กร	28
รูปที่ 2.8	แสดงการไหลของข้อมูลในระบบ	29
รูปที่ 2.9	แสดงคอนโทรลพื้นฐานของวิซวลเบสิก	32
รูปที่ 3.1	ภาพรวมของระบบเดิม	42
รูปที่ 3.2	การติดต่อระหว่างไคลเ็นต์กับเซิร์ฟเวอร์	43
รูปที่ 3.3	ภาพรวมของระบบ	45
รูปที่ 3.4	ฟอร์มลือคอิน	56
รูปที่ 3.5	ฟอร์มซื้อเลือก	57
รูปที่ 3.6	ฟอร์มประวัตินิติศิต	58
รูปที่ 3.7	ฟอร์มประวัตินิติศิตส่วนการศึกษา	59
รูปที่ 3.8	ฟอร์มประวัตินิติศิตส่วนผู้ปกครอง	60
รูปที่ 3.9	ฟอร์มประวัตินิติศิตส่วนสถานภาพทางการศึกษา	61
รูปที่ 3.10	ฟอร์มซื้อเลือกของหลักสูตรการศึกษาและวิชา	62
รูปที่ 3.11	ฟอร์มหลักสูตรการศึกษา	63
รูปที่ 3.12	ฟอร์มวิชาเรียน	64
รูปที่ 3.13	ฟอร์มเพิ่ม-ลบวิชาเรียนในหลักสูตร	65
รูปที่ 3.14	แสดงเอนติตี้หลักๆที่เป็นออบเจ็คท์ในระบบ	67
รูปที่ 3.15	แสดงออบเจ็คท์โมเดลแบบลำดับชั้น	68
รูปที่ 3.16	แสดงแผนภาพในแอมของประวัตินักศึกษา	70
รูปที่ 3.17	แสดงโพล์ชาร์ตของหน้าจอป้อนรหัสผ่าน	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.18	แสดงโฟลว์ชาร์ตของหน้าจอเลือกประเภทการทำงาน	72
รูปที่ 4.1	หน้าจอเข้าสู่ระบบ	75
รูปที่ 4.2	เลือกประเภทการทำงานของประวัตินักศึกษา	75
รูปที่ 4.3	หน้าจอประวัตินักศึกษาในส่วนประวัติส่วนตัว	76
รูปที่ 4.4	หน้าจอประวัตินักศึกษาในส่วนประวัติการศึกษา	77
รูปที่ 4.5	หน้าจอประวัตินักศึกษาในส่วนของสถานภาพทางการศึกษา	78*
รูปที่ 4.6	หน้าจอประวัตินักศึกษาในส่วนประวัติผู้ปกครอง	79
รูปที่ 4.7	หน้าจอเลือกประเภทการทำงานของหลักสูตรการศึกษา	80
รูปที่ 4.8	หน้าจอแสดงรายละเอียดของหลักสูตรการศึกษา	81
รูปที่ 4.9	หน้าจอแสดงข้อมูลวิชาเรียนต่างๆไป	82
รูปที่ 4.10	หน้าจอการเลือกข้อมูลหลักสูตรวิชาที่ต้องแก้ไข	83

## สารบัญตาราง

ตารางที่ 2.1	แสดงหน้าที่การใช้งานและเครื่องมือที่ใช้ในแต่ละส่วนของทรีเทียร์	3
ตารางที่ 2.2	แสดงตัวอย่างการจัดการงานสั่งซื้อสินค้า	10
ตารางที่ 2.3	แสดงความสัมพันธ์ของเอนติตี้แบบหนึ่งต่อหนึ่ง	20
ตารางที่ 2.4	แสดงความสัมพันธ์ของเอนติตี้แบบหนึ่งต่อกลุ่ม	20
ตารางที่ 2.5	แสดงความสัมพันธ์ของเอนติตี้แบบกลุ่มต่อกลุ่ม	21
ตารางที่ 2.6	แสดงคีย์หลักกับคีย์นอก	22
ตารางที่ 2.7	แสดงฟังก์ชันการขึ้นต่อกัน	23
ตารางที่ 2.8	แสดงฟังก์ชันการขึ้นต่อกันอย่างเต็มที่	24
ตารางที่ 2.9	แสดงอินโพรเซสและเอทออฟโพรเซส เซิร์ฟเวอร์	34
ตารางที่ 2.10	แสดงรายละเอียดของโหนด ออโตเมชัน อินเตอร์เฟส	37
ตารางที่ 3.1	แสดงออบเจ็กต์ อินเตอร์เฟส	69

# บทที่ 1

## บทนำ

ในปัจจุบันสถาปัตยกรรมแบบ ไคลเอนต์/เซิร์ฟเวอร์ (Client/Server) ซึ่งมีการทำงานแยกออกเป็น 2 ส่วน คือ ส่วนผู้ขอใช้บริการหรือฟรอนต์เอนด์(Front-End or Client) กับส่วนผู้ให้บริการหรือแบคเอนด์(Back-End or Server) เป็นที่นิยมในปัจจุบันสำหรับการพัฒนาระบบต่างๆ เพราะมีเครื่องมือที่ช่วยในการพัฒนามากมายทำให้การพัฒนาโปรแกรมทำได้ง่าย และสามารถนำส่วนฟรอนต์เอนด์ไปใช้กับแบคเอนด์เครื่องใดก็ได้ที่มีมาตรฐานเดียวกัน ในโรงงานสำนักงานทะเบียนกลางก่อนหน้านี้ ได้มีการปรับปรุงจากโปรแกรมเดิมที่มีการทำงานแบบเท็กซ์โหมด (Text Mode) บนระบบปฏิบัติการดอส (DOS) มีลักษณะการทำงานแบบใช้งานคนเดียวและมีโปรแกรมการทำงานในแต่ละส่วนแยกออกจากกัน พัฒนามาเป็นแบบไคลเอนต์/เซิร์ฟเวอร์ที่ใช้ความสามารถของเทคโนโลยีการติดต่อผ่านสายสื่อสาร และความสามารถของโปรแกรมที่ใช้ช่วยในการพัฒนา มาพัฒนาระบบที่มีอยู่เดิมให้มีประสิทธิภาพดีขึ้น มีหน้าจอที่ติดต่อกับผู้ใช้เป็นแบบกราฟิกโหมด (Graphic Mode) ทำงานอยู่บนวินโดวส์และมีการทำงานเป็นแบบเรียลไทม์ (Real Time) ทำให้ได้รับข้อมูลที่ทันสมัยอยู่เสมอ จากการศึกษาที่ได้ศึกษาระบบที่มีอยู่พบว่าสถาปัตยกรรมแบบนี้ยังมีปัญหาคือ เมื่อมีการแก้ไขกฎเกณฑ์ใน โปรแกรมก็ต้องตามไปแก้ไขกฎเกณฑ์ที่เครื่อง ไคลเอนต์ทุกเครื่องเสมอไป ซึ่งถ้าเป็นระบบที่มีการแก้ไขกฎเกณฑ์ที่ใช้เสมอๆและมีเครื่อง ไคลเอนต์ในระบบเป็นจำนวนมากก็จะทำให้เสียค่าบำรุงรักษาสูงโดยไม่จำเป็น

จึงได้มีการพัฒนาสถาปัตยกรรมขึ้นมาใหม่เพื่อแก้ปัญหานี้ โดยการเพิ่มชั้น(Layer) เข้าไปแทรกระหว่างส่วนไคลเอนต์กับเซิร์ฟเวอร์ ซึ่งชั้นที่แทรกเข้าไปเราเรียกว่า "บิสซิเนส เซอร์วิส" (Business Service) เป็นส่วนที่ให้บริการกฎเกณฑ์ข้อจำกัดของระบบ ส่วนชั้นที่ติดต่อกับผู้ใช้เรียกใหม่ว่า "ส่วนให้บริการผู้ใช้" (User Service) และเรียกส่วนที่ให้บริการข้อมูลว่า "ส่วนให้บริการข้อมูล" (Data Service) เมื่อมีชั้นที่ทำหน้าที่เป็นตัวกลางของการติดต่อระหว่างผู้ให้บริการข้อมูล (Data Service ซึ่งก็คือ DBMS) กับส่วนให้บริการผู้ใช้ (User Service ซึ่งก็คือ Client Application) แล้ว เราก็สามารถเอากฎเกณฑ์ที่กำหนด เก็บไว้ในส่วนที่ให้บริการกฎเกณฑ์ข้อจำกัดได้ และเมื่อมีการแก้ไขจะสามารถแก้ไขที่ส่วนนี้ได้โดยไม่ต้องพัฒนาโปรแกรมทั้งหมดใหม่ ทำให้การพัฒนาโปรแกรมใหม่ รวมถึงการบำรุงรักษาทำได้โดยง่าย ซึ่งในการให้บริการของส่วนการเก็บข้อจำกัดของผู้ใช้ เราจะใช้เทคนิคของ โอเล่ ออโตเมชัน เซิร์ฟเวอร์ (OLE Automation Server) เป็นผู้ให้บริการ เราเรียกสถาปัตยกรรมแบบนี้ว่า "สถาปัตยกรรมทรีเทียร์"

## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 สถาปัตยกรรมทิวเทียร์ (Two-Tier Architecture)

โคล์เอ็นต์/เซิร์ฟเวอร์ แอปพลิเคชัน ส่วนใหญ่ในปัจจุบันมักจะถูกสร้างขึ้นด้วยภาษารุ่นที่ 4 (Fourth-Generation Language - 4GL) และถูกออกแบบเป็นสถาปัตยกรรมทิวเทียร์ มีส่วนการทำงานแยกเป็น 2 ส่วนคือ ฟรอนต์เอนด์กับแบคเอนด์ ซึ่งแต่ละส่วนมีลักษณะดังนี้

เทียร์	ลักษณะพิเศษ
1. ฟรอนต์เอนด์	ส่วนติดต่อกับผู้ใช้งานแบบกราฟิก(GUI Interface); คำสั่งภาษา SQL (SQL calls); โปรแกรมประยุกต์(Desktop application); ภาษาระดับ 4 (4GL)
2. แบคเอนด์	ผู้ให้บริการฐานข้อมูลเอสคิวแอล (SQL database server); ระบบปฏิบัติการแบบมัลติทาสกิ้ง (Multitasking operating system)

จากการศึกษา เราพบว่าสถาปัตยกรรมแบบทิวเทียร์นี้มีข้อจำกัดอยู่หลายประการ คือ

1. ความยากลำบากในการจัดการส่วนฟรอนต์เอนด์เพิ่มขึ้น โดยถ้ามีการเปลี่ยนแปลงแก้ไขฐานข้อมูลที่เซิร์ฟเวอร์ เครื่องโคล์เอ็นต์ทุกเครื่องจะต้องทำการปรับปรุงโปรแกรมให้ทันสมัยโดยให้ตรงกับข้อมูลใหม่ที่มีการเปลี่ยนแปลงนั้น เช่น การเปลี่ยนแปลงของคอลัมน์ในตาราง เป็นต้น

2. ความยากลำบากในการแบ่งใช้โปรเซสที่ใช้อยู่ร่วมกัน การออกแบบบิสซิเนสโปรเซสและโค้ด (Code) ใส่ในแอปพลิเคชัน ทำให้ไม่สามารถนำโปรเซสนั้นมาใช้ใหม่กับแอปพลิเคชันอื่นได้

#### 3. ความยากลำบากในการรักษาความปลอดภัย

3.1 ความยากในการดูแลความปลอดภัยในโครงสร้างเก็บข้อมูลของฐานข้อมูล  
เพราะการที่จะพัฒนาโปรแกรมส่วนฟรอนต์เอนด์ให้เรียกใช้ฐานข้อมูลได้นั้น จะต้องให้ผู้พัฒนาเข้าถึงโครงสร้างของตารางและความสัมพันธ์ของข้อมูลในตารางด้วย เพื่อการนำมาใช้ในการเขียนโปรแกรม อีกทั้งในองค์กรที่มีขนาดใหญ่จะเป็นไปไม่ได้เลยในการที่จะกำหนดให้สิทธิ์ที่เหมาะสมสำหรับทุกคนในการที่จะเข้าใช้ข้อมูล

3.2 ความปลอดภัยในส่วนของแอปพลิเคชันคือ ผู้ใช้สามารถเข้าถึงข้อมูลได้โดยใช้เครื่องมือชนิดอื่นแทนการผ่านฟรอนต์เอนด์แอปพลิเคชัน เช่น Microsoft Excel หรือคีย์เวิร์ดตัวอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 สถาปัตยกรรมทรีเทียร์ (Three-Tier Architecture)

สถาปัตยกรรมทรีเทียร์มีการพัฒนามาจากสถาปัตยกรรมไคลเอ็นต์/เซิร์ฟเวอร์แบบเดิม ที่แบ่งการทำงานจาก 2 ส่วน คือ ฟรอนต์เอนด์ กับ แบคเอนด์ มาเป็น 3 ส่วน โดยเพิ่มส่วนที่เรียกว่า "บิสซิเนส เซอร์วิส" ขึ้นมา ซึ่งแต่เดิมส่วน "บิสซิเนส เซอร์วิส" นี้ จะอยู่ทั้งในส่วนฟรอนต์เอนด์ กับ แบคเอนด์ ทำให้เกิดผลเสียต่อระบบส่วนรวม เพราะเมื่อมีการเปลี่ยนแปลงungskุณณ์ของระบบใหม่ จะต้องตามไปแก้ไขทุกส่วนที่เครื่องไคลเอ็นต์

โครงสร้างทั้ง 3 ส่วน มีดังนี้

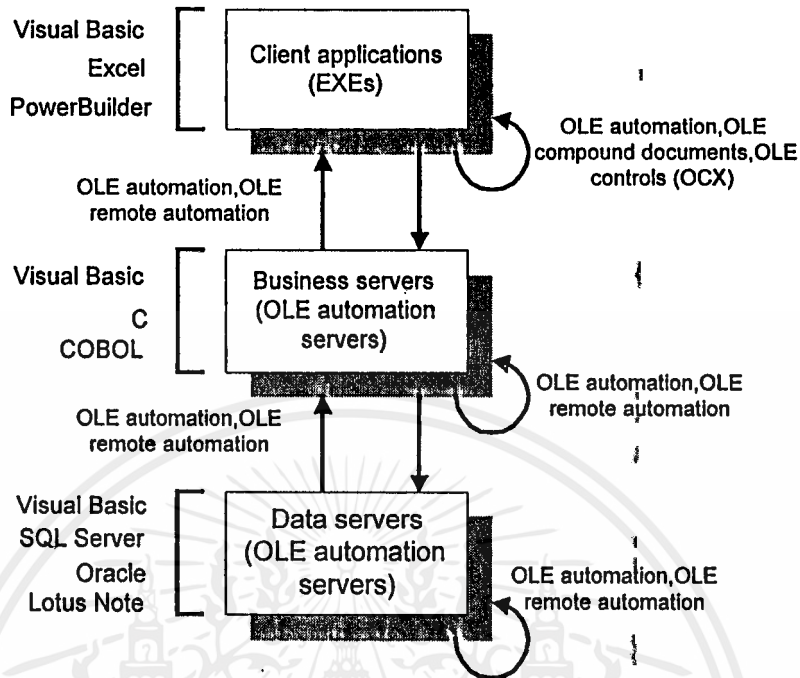
1. ส่วนบริการผู้ใช้ (User Service)
2. ส่วนบริการกฎเกณฑ์ข้อจำกัดของระบบ (Business Service)
3. ส่วนบริการข้อมูล (Data Service)

จากการแยกส่วนตามการให้บริการจึงอาจเรียก ทรีเทียร์อีกอย่างหนึ่งว่า "เซอร์วิส โมเดล" (Service Model) โดยแต่ละส่วนไม่จำเป็นต้องแยกออกจากกันเป็นส่วนๆจริงๆ (ไม่ใช่แบบ Physical แต่เป็น Logical คือไม่จำเป็นต้องแยกเป็น 3 เครื่อง) จะเป็นลักษณะของ "คอนเซ็ปชวล เลเยอร์" (Conceptual layer) คือแบ่งออกจากกันในแนวความคิดที่แต่ละส่วนทำงานแยกกัน ซึ่งส่วนที่เพิ่มเข้าไป (Business Service) อาจอยู่บนเครื่องไหนก็ได้ (อาจอยู่บนเครื่องไคลเอ็นต์หรือเครื่องเซิร์ฟเวอร์ก็ได้)

โดยแต่ละส่วนของทรีเทียร์จะมีลักษณะพิเศษ, ความรับผิดชอบ, กลไกที่ใช้ในการติดต่อและเครื่องมือที่ใช้ในการพัฒนาในแต่ละส่วนของทรีเทียร์ ดังตารางและรูปต่อไปนี้

เทียร์	ชนิดการบริการ	ลักษณะพิเศษ	ความรับผิดชอบ	เครื่องมือที่ใช้พัฒนา
1. ยูสเซอร์ เซอร์วิส (User Service)	โปรแกรมประยุกต์ (Client application)	ส่วนติดต่อกับผู้ใช้งาน (GUI Interface)	การนำเสนอ	ภาษาระดับ 4, เดสก์ทอป แอปพลิเคชัน
2. บิสซิเนส เซอร์วิส (Business Service)	บิสซิเนส เซอร์วิส (Business Service)	บิสซิเนส ออบเจกต์ พรอโทคอล & เมธอด	ข้อจำกัด, กฎเกณฑ์ของ ระบบ และความปลอดภัย	ภาษาระดับ 4 บางชนิด; โคบอล; ซี
3. คาด้า เซอร์วิส (Data Service)	ผู้ให้บริการข้อมูล (Data Servers)	ตัวจัดการฐานข้อมูล	ความถูกต้องของข้อมูล	Databases, Messaging System

ตารางที่ 2.1 แสดงหน้าทีการใช้งาน และเครื่องมือที่ใช้ในแต่ละส่วนของทรีเทียร์



รูปที่ 2.1 แสดงเครื่องมือที่ใช้และกลไกที่ใช้ในการติดต่อแต่ละชั้นของทรีเทียร์

รายละเอียดของแต่ละเทียร์ สามารถอธิบายได้ดังนี้

### 2.3 สถาปัตยกรรม เทียร์ที่หนึ่ง (Architecting Tier 1:)

#### ส่วนบริการผู้ใช้ (User Service หรือเรียกว่าส่วน Client application)

การทำงานในส่วนนี้จะเป็นการติดต่อกับผู้ใช้โดยตรง เพื่อแสดงข่าวสาร (information) และเก็บรวบรวมข้อมูล ส่วนนี้จะทำหน้าที่ในการเชื่อมต่อระหว่างผู้ใช้ (User) กับส่วนบริการกฎเกณฑ์ข้อจำกัดของระบบ โดยการควบคุมส่วนไคลเอ็นต์นี้ จะใช้แอปพลิเคชันที่เขียนขึ้นโดยวิซวลเบสิก ไคลเอ็นต์แอปพลิเคชันของโครงสร้างทรีเทียร์จะมีความแตกต่างกับทรีเทียร์คือ โปรแกรมแอปพลิเคชันของโครงสร้างแบบทรีเทียร์จะมีขนาดไม่ใหญ่ เพราะไม่ได้เก็บส่วนข้อจำกัดของระบบ (Business Logic) เอาไว้ เพียงแต่เก็บโค้ดสำหรับส่วนที่แสดงผลเท่านั้น (Presentation code)

สำหรับการพัฒนาเทียร์นี้สามารถใช้ภาษาระดับ 4 เช่น Visual Basic, Power Builder, Delphi, etc. และเดสก์ทอปแอปพลิเคชัน เช่น Ms Excel ในการพัฒนาได้ แต่ Visual Basic เป็นทางเลือกที่ดีที่สุด เพราะมีการสนับสนุน โอเล่ ออโตเมชัน และ โอเล่ คอนโทรลที่ดีกว่า

### 2.3.1 กฎในการออกแบบส่วนบริการผู้ใช้

2.3.1.1. ส่วนบริการผู้ใช้เป็นส่วนที่ติดต่อกับผู้ใช้โดยตรง ดังนั้นควรจะออกแบบให้มีลักษณะที่เข้าใจง่าย, สื่อความหมายดี จึงควรอยู่ในรูปแบบของกราฟิกเพื่อเก็บรวบรวมข้อมูลที่ผู้ใช้ป้อนเข้ามาและต้องแสดงข้อมูลให้ผู้ใช้เห็นได้ชัดเจน

2.3.1.2. ไม่ควรมีกฎเกณฑ์ของส่วนให้บริการกฎเกณฑ์อยู่ เพราะเมื่อมีการเปลี่ยนแปลงกฎเกณฑ์ของระบบจะได้ไม่ต้องตามไปเปลี่ยนแปลงกฎเกณฑ์ที่เครื่องอื่นๆอีก

2.3.1.3. เครื่องไคล์เอ็นต์แต่ละเครื่องสามารถสร้างส่วนที่ผู้ใช้ติดต่อกับผู้ใช้ได้ โดยไม่เกี่ยวข้องกับไคล์เอ็นต์เครื่องอื่น

### 2.3.2 เครื่องมือที่เลือกใช้ในการพัฒนาส่วนบริการผู้ใช้ ควรมีลักษณะดังนี้

2.3.2.1. ต้องมีความสามารถในการติดต่อกับส่วนประกอบ (Component) อื่นๆ ในเทียร์ส่วนที่ให้บริการผู้ใช้ (User Services Tier) และเทียร์ส่วนที่ให้บริการกฎเกณฑ์ของระบบ (Business Services Tier) ได้ เช่น โอล่า ออโตเมชัน เซิร์ฟเวอร์ (OLE Automation servers) และ โอล่าคอนโทรล (OLE Controls)

2.3.2.2. ต้องสามารถสร้างส่วนที่ติดต่อกับผู้ใช้ ให้ผู้ใช้สามารถใช้งานได้ง่าย

2.3.2.3. ภาษาที่ใช้เลือกใช้ในการพัฒนาควรเป็นภาษามาตรฐาน

โดยในระบบได้เลือกใช้ Visual Basic 4.0 Enterprise เป็นเครื่องมือในการพัฒนาส่วนบริการผู้ใช้นี้ นอกจากที่กล่าวมาแล้ว วิซวลเบสิก 4.0 ยังสามารถสร้างแอปพลิเคชันได้ทั้งแบบ 16 บิต และ 32 บิต จากแหล่งข้อมูลเดียวกันได้ แต่นิยมใช้แบบ 32 บิตเพราะมีข้อดีที่เหนือกว่าแบบ 16 บิตคือมีความเร็วในการเอ็กซีคิวท์ที่เพิ่มขึ้น สามารถจัดการรีซอร์สได้ดีกว่าและสามารถรันแอปพลิเคชันได้หลายตัวในเวลาเดียวกัน

### 2.3.3 การติดต่อระหว่างส่วนให้บริการผู้ใช้ และส่วนบิสซิเนส เซอร์วิส

ส่วนบิสซิเนส เซอร์วิส จะถูกสร้างให้เป็นโอล่า ออโตเมชัน เซิร์ฟเวอร์ ซึ่งจะเป็น "บิสซิเนส ออบเจ็กต์" โดยออบเจ็กต์ จะกำหนด เมธอด (Method) ที่เราเรียกว่าฟังก์ชันการทำงาน และพรอพเพอร์ตี้ ( Properties) คือค่าคุณสมบัติของออบเจ็กต์ไว้ในส่วนของ ไคล์เอ็นต์ แอปพลิเคชัน โดยทั้งสองส่วนนี้(ส่วนให้บริการผู้ใช้ และส่วนบิสซิเนส เซอร์วิส)ติดต่อกันได้โดยผ่านทาง "รีโมท ออโตเมชัน"

ตัวอย่างของการสร้างและเรียกใช้บิสซิเนส ออบเจกต์

```
Public Sub PlaceOrderTest()
```

'โพธิ์เซอร์ในการสร้างออร์เดอร์โดยมีการสร้างโอเล่อบเจกต์สำหรับออร์เดอร์ เมเนเจอร์

```
Dim loOrderMgr As Object 'กำหนดตัวแปรออบเจกต์
```

```
Set loOrderMgr = CreateObject("OrderMgr.Application") 'สร้างออบเจกต์ขึ้นมา
```

```
loOrderMgr.PlaceOrder Customer:=fiCustomerID,Product:=fiProduct, _
```

```
Quantity:=fiQuantity
```

'CustomerID,ProductID และ Order Quantity เป็นตัวแปรที่อยู่ในฟอร์มซึ่ง fiCustomerID,

'fiProduct,fiQuantity ได้ร้องขอให้เมธอดPlaceOrder ส่งค่าที่ได้จาก CustomerID,ProductID

'และ Order Quantity ไปให้ เพื่อส่งต่อไปกระบวนการต่อไป

```
End Sub
```

เราจะเห็นจากตัวอย่างว่า การติดต่อระหว่างส่วนไคลเอนต์ แอปพลิเคชัน กับส่วนบิสซิเนส เซิร์ฟเวอร์ สามารถทำได้ง่ายเพราะทุกๆบิสซิเนส ลอจิกจะอยู่ในเทียร์กลางหมด เทียร์แรกจึงไม่ต้อง มีโค้ดจำนวนมากเพื่อ โปรเซสข้อมูลที่จะรับเข้ามา

ตัวอย่างต่อไปแสดงถึงการที่จะขอเรียกดูสถานะของการสั่งของที่ได้สั่ง ไปแล้ว

```
Public Sub OrderStatusTest()
```

```
Dim loOrderMgr As Object
```

```
Dim liOrderStatus As Integer
```

```
Set loOrderMgr = CreateObject("OrderMgr.Application")
```

```
liOrderStatus = loOrderMgr.GetOrderStatus(OrderId := fiOrderId)
```

```
'Code to Display message box
```

```
End Sub
```

อะไรจะเกิดขึ้นเมื่อออร์เดอร์โปรเซสต้องเปลี่ยนแปลง เช่น ในกรณีที่มีการสั่งของนั้น ลูกค้า จะได้รับของก็ต่อเมื่อแผนกส่งของเห็นว่า ของที่สั่งมีอยู่ในคลังพอตามจำนวนที่สั่งไว้ แต่หากมีไม่พอ เช่นสั่ง 20 ชิ้น แต่มีของแค่ 16 ชิ้น แผนกส่งของก็จะรองนกว่าของจะครบจึงจะส่งแก่ลูกค้าซึ่งก็ ไม่รู้ว่าเมื่อไหร่ แทนที่จะส่งของที่มีอยู่ 16 ชิ้นไปให้ก่อน แล้วค่อยทยอยส่งเมื่อมีของมาเพิ่ม ตรงนี้เองที่สามารถไปเปลี่ยนแปลงลักษณะการทำงานได้ที่บิสซิเนส เซอร์วิส

### 2.3.4 วิธีการพัฒนาในส่วนของไคลเอนต์ แอปพลิเคชัน

2.3.4.1. **รวบรวมความต้องการว่าผู้ใช้ระบบต้องการอะไรจากส่วนไคลเอนต์ แอปพลิเคชัน** โดยอาจจะใช้การสัมภาษณ์จากผู้ใช้ระบบโดยตรงก็ได้ ซึ่งเป็นวิธีที่ดีที่สุด

2.3.4.2. **การออกแบบในส่วนไคลเอนต์ แอปพลิเคชัน** ต้องวางแผนในแต่ละหน้าจอ และวิธีการที่ในแต่ละหน้าจอจะติดต่อกับส่วนบิสซิเนส เซอร์วิส เทียร์ โดยการเชื่อมต่อระหว่างหน้าจอในการออกแบบ จะต้องดูจากความต้องการของผู้ใช้ด้วย

2.3.4.3. **การทดลองใช้ไคลเอนต์ แอปพลิเคชัน** ที่ได้ออกแบบแล้ว การทดลองใช้ส่วนที่ได้พัฒนาขึ้นเป็นต้นแบบ (Prototyping) นี้ ทำเพื่อให้รู้ถึงข้อผิดพลาดต่างๆ เพื่อจะได้แก้ไขโปรแกรมใช้งานจริงให้สมบูรณ์

2.3.4.4. **คุณสมบัติตอบสนองจากผู้ใช้**

2.3.4.5. **ปรับปรุงแก้ไข** เพื่อให้แอปพลิเคชันนั้นใช้งานได้ง่าย

## 2.4 สถาปัตยกรรม เทียร์ที่สอง (Architecting Tier 2:)

### ส่วนให้บริการลูกค้าของระบบ (Business Service)

ส่วนนี้จะทำหน้าที่เชื่อมต่อระหว่างส่วนให้บริการผู้ใช้งาน (User Service) และส่วนบริการข้อมูล (Data Services) ซึ่งจะตอบสนองการร้องขอจากผู้ใช้หรือส่วนบิสซิเนสของระบบอื่น

งานของส่วนนี้เป็นการดำเนินการที่ถูกกำหนดโดยความต้องการของแอปพลิเคชัน เช่น การใส่ข้อมูลการสั่งซื้อสินค้า หรือการพิมพ์รายชื่อลูกค้า โดยมีข้อจำกัดกฎเกณฑ์ของระบบที่เรียกว่า "บิสซิเนส รูล" (Business Rules) เป็นตัวกำหนดและควบคุมการทำงานของส่วนนี้

การออกแบบส่วนบิสซิเนส เซอร์วิส ใช้เทคนิคของโอเลอโตเมชันเซิร์ฟเวอร์ โดยขั้นแรกจะรวบรวมกระบวนการพื้นฐานต่าง ๆ ของระบบ ตามจุดประสงค์และเป้าหมายขององค์กรนั้น และกระบวนการพื้นฐานต่างๆเหล่านั้นจะถูกนำเสนอในรูปแบบของบิสซิเนส ออบเจกต์

โดยบิสซิเนส ออบเจกต์จะถูกแบ่งออกเป็น 2 ชนิด ได้แก่

1. ออบเจกต์ต่าง ๆ ที่มีตัวตนและไม่มีตัวตน
2. เหตุการณ์ธุรกิจ

ออบเจกต์ที่มีตัวตนได้แก่ เอกสารต่าง ๆ เช่น ใบส่งของ, ใบสั่งซื้อ, ใบเสนอขาย, ใบสัญญาต่างๆ หรือวัตถุที่จับต้องได้ สำหรับออบเจกต์ที่ไม่มีตัวตนได้แก่ โปรเซสและความสัมพันธ์ รวมทั้งเครดิต เป็นต้น

ส่วนเหตุการณ์ธุรกิจนั้นเป็นกิจกรรมที่เกิดจากผลของการเปลี่ยนออบเจกต์ หรือการกระทำเอหนึ่งที่ต้องการอีกการกระทำเข้ามาช่วยเพื่อผลประโยชน์ขององค์กร อนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ข้อดีของบิสซิเนสออบเจกต์

- 1.สามารถนำกลับมาใช้ได้ใหม่
- 2.สามารถย้ายส่วนบิสซิเนสลอจิกไปยังฮาร์ดแวร์ที่มีประสิทธิภาพแทนการนำไปไว้ยังไคลเอนต์ที่มีประสิทธิภาพต่ำ
- 3.ง่ายต่อการบำรุงรักษาบิสซิเนสออบเจกต์เพราะรวมอยู่บนเครื่องเดียวกัน
- 4.เพิ่มความยืดหยุ่นในการออกแบบแอปพลิเคชัน โดยการหลีกเลี่ยงการพัฒนาไคลเอนต์ แอปพลิเคชัน ที่มีขนาดใหญ่ มาเป็นการแบ่งส่วนของระบบออกมาเป็นส่วนย่อยๆแล้วแยกพัฒนาทีละส่วนย่อย ทำให้การพัฒนาโปรแกรมได้โดยง่าย และถูกต้อง

#### 2.4.1 การออกแบบส่วนบิสซิเนส เซอร์วิส

การออกแบบส่วนบิสซิเนส เซอร์วิสมีความจำเป็นพอๆกับการออกแบบส่วนฐานข้อมูล ส่วนของข้อมูลเชิงวัตถุ (Business Object) จะถูกผู้พัฒนาโปรแกรม (Developer) เรียกใช้เมื่อมีการสร้างโปรแกรม ดังนั้นส่วนของข้อมูลเชิงวัตถุต้องมีทุกบทบาทและหน้าที่ (Function) ที่จำเป็นในการสร้างโปรแกรมประยุกต์ในส่วนของผู้ใช้งาน

การออกแบบส่วนบิสซิเนส เซอร์วิส มี 4 ขั้นตอนดังนี้

##### 2.4.1.1 การกำหนดกลุ่มของส่วนประกอบที่มีอยู่จริงในองค์กร

(Identify logical Business Entities)

Orders	Line Item	Customer	Product
Order number Customer number Order date Order status Line items	Line number Product number Qty ordered Qty shipped	Customer number Name Address Contact person	Product number Description Vendor name

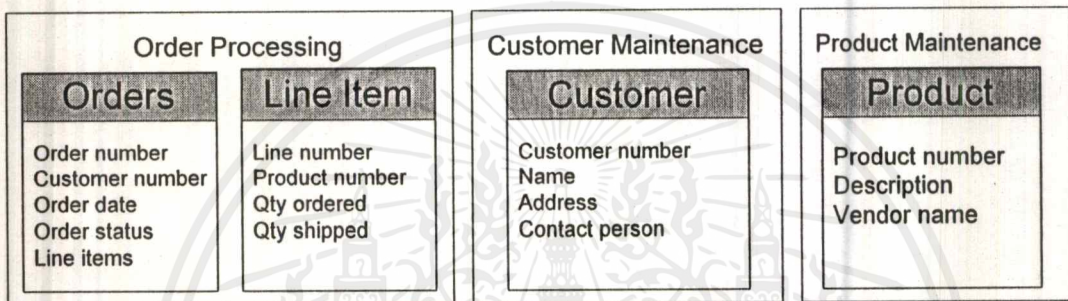
รูปที่ 2.2 แสดงกลุ่มของส่วนประกอบ

เริ่มจากการกำหนดบิสซิเนสเอนติตี้ ( Business Entities ) ที่มีอยู่ในองค์กร เช่น ใบสั่งซื้อ, ลูกค้า, สินค้า เป็นต้น ซึ่งบิสซิเนสเอนติตี้เหล่านี้ก็จะกลายเป็นบิสซิเนสออบเจกต์ต่างๆ หลังจากที่ได้กำหนดบิสซิเนสเอนติตี้ทั้งหมดที่มีอยู่ภายในองค์กรแล้ว ต่อไปก็หาแอตทริบิว ( Attributes ) ส่วนประกอบของออบเจกต์ และฟังก์ชันหรือเมธอดของแต่ละเอนติตี้ ซึ่งแอตทริบิวสามารถกำหนดขึ้นจากคุณลักษณะของเอนติตี้ เช่น เอนติตี้ของใบสั่งซื้อสินค้า ก็จะมีผู้สั่งซื้อ, วันที่สั่งซื้อ, สินค้าที่สั่งซื้อ เป็นต้น นอกจากนี้แอตทริบิวและเอนติตี้จะมีฟังก์ชันของมันเอง เช่น เอนติตี้ใบสั่งซื้ออาจจะมีฟังก์ชันเพิ่มข้อมูล, ลบข้อมูล, แก้ไขข้อมูลและการส่งพิมพ์เอกสารเป็นฟังก์ชันของเอนติตี้ นั้น ส่วนเอกสารเป็นเอกสารที่ส่งวนเวียนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นาเบเซประะยะขนด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของความสัมพันธ์ระหว่างเอนทิตีหรือกิจกรรมที่แต่ละเอนทิตีกระทำต่อกัน เช่น ลูกค้าสั่งซื้อสินค้าโดยกรอกใบสั่งซื้อ, ใบสั่งซื้อนั้นถูกอ่านและทำการจัดส่งสินค้าตามที่ลูกค้าต้องการ เป็นต้น

2.4.1.2 การจัดแบ่งกลุ่มของส่วนประกอบทางลอจิกคอล (Identify logical business entity groupings)



รูปที่ 2.3 แสดงการจัดแบ่งเป็นกลุ่มของส่วนประกอบ

หลังจากที่แอดทริบิวของบิสซิเนสออบเจกต์และฟังก์ชันของ บิสซิเนส ออบเจกต์ที่เป็นไปได้ทั้งหมดได้ถูกกำหนดขึ้นแล้ว เราจำเป็นต้องจับกลุ่มเอนทิตีที่เหมาะสมไว้ด้วยกันและแบ่งส่วนการทำงานให้ระบบมีประสิทธิภาพมากขึ้น เช่น จากตัวอย่างเราสามารถจัดเอนทิตีการสั่งซื้อกับชุดสินค้าที่ถูกสั่งซื้อให้อยู่ในกลุ่มเดียวกันได้เป็นกระบวนการการสั่งซื้อ เพราะมีการทำงานคล้ายๆกัน ส่วนเอนทิตีของลูกค้าและสินค้าที่ไม่มีความสัมพันธ์กันเราก็แยกไว้ต่างหาก

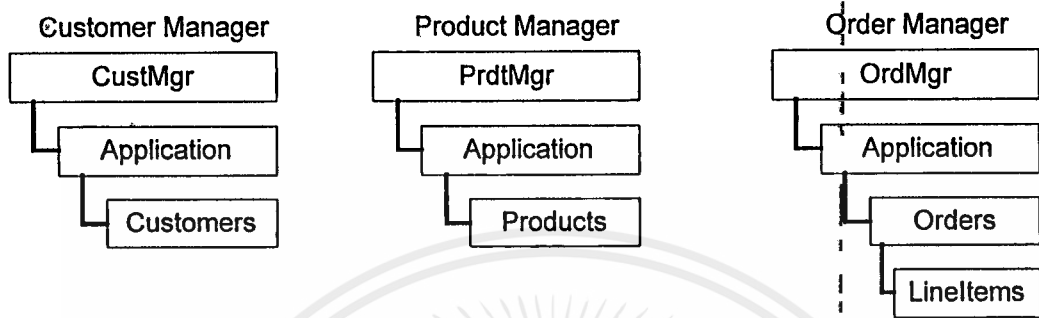
2.4.1.3 การจัดแบ่งกลุ่มของส่วนประกอบทางกายภาพ (Identify physical component)

วิธีการจัดแบ่งส่วนประกอบทางกายภาพ คือ การใช้เอนทิตีที่มีอยู่ในองค์กรมาจัดกลุ่มทำเป็นตารางในฐานข้อมูล โดยแบ่งตามหัวข้อหลัก ตัวอย่างเช่น องค์กรส่วนมากมักใช้ส่วนประกอบที่คล้ายคลึงกัน คือ กลุ่มของการสั่งซื้อสินค้า, กลุ่มของสินค้า, กลุ่มของลูกค้า, กลุ่มของผู้ขาย, กลุ่มของพนักงาน และอื่นๆ

2.4.1.4 การพัฒนาออบเจกต์แบบเป็นลำดับชั้น และการคิดต่อสำหรับแต่ละบิสซิเนส เซิร์ฟเวอร์ (Develop an object hierarchy and interface for each business server)

การออกแบบชั้นของออบเจกต์ จะออกแบบตามความสัมพันธ์ของออบเจกต์ ซึ่งการกำหนดออบเจกต์ในแอปพลิเคชัน ลำดับชั้นของออบเจกต์จะประกอบด้วยคุณสมบัติและวิธีการ (Properties and Methods) ซึ่งแอปพลิเคชันจะใช้ในการคุมแต่ง (manipulate) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจ็กต์ต่างๆในองค์กร การจะกำหนดการติดต่อกับส่วนที่เก็บข้อจำกัด สำคัญมากเพราะจะต้องแน่ใจว่าสิ่งที่กำหนดคนนั้นได้ครอบคลุมทุกอย่างที่ต้องการของออบเจ็กต์ วิธีที่ดีที่สุดที่จะทำคือ การถามผู้ใช้งานเลยว่าต้องการเห็นข้อมูลอะไรบ้าง



รูปที่ 2.4 แสดงตัวอย่างแต่ละชั้นของบิสซิเนส

2.4.2 การรวบรวมกฎเกณฑ์ข้อจำกัด (Encapsulating Business Rules)

กฎเกณฑ์ข้อจำกัดที่อยู่ในองค์กรจะถูกรวบรวมไว้ใน บิสซิเนส ออบเจ็กต์ซึ่งจะแสดงผ่านทางคุณสมบัติของออบเจ็กต์ และวิธีการเข้าถึงออบเจ็กต์นั้น

	Application	Orders	Order	LinelItems	LinelItem
<b>Collections:</b>	Orders		LinelItem		
<b>Member Objects:</b>		Item		Item	Parent
<b>Properties:</b>	Id UserId Password	Count	OrderID CustomerID OrderDate ShippedFlag RemoveFlag	Count	LinelID ProductID OrderQty ShipQty
<b>Methods:</b>	Initialize	LoadOrders UpdateOrders Add Remove		Add Remove	
<b>Data Tier:</b>		sp_get_orders	sp_add_order sp_delete_order		sp_add_lineitem

ตารางที่ 2.2 แสดงตัวอย่างของการจัดการงานตั้งชื่อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เมธอด (Methods)** แสดงถึงการกระทำที่โอเล่ ออโตเมชัน เซิร์ฟเวอร์ ใช้เพื่อควบคุมแอปพลิเคชัน เมธอดมักจะแสดงถึงการกระทำของบิสซิเนสออบเจกต์ที่มีความเกี่ยวข้องกัน ตัวอย่างเช่น Remove Method ของการสั่งซื้อสินค้า(Orders)เป็นการนำคำสั่งซื้อของออกจากระบบ ซึ่ง Remove จะประกอบด้วยการติดต่อของแอปพลิเคชันที่ใช้ในการควบคุมเพื่อลบคำสั่งซื้อ โดยไม่ต้องทราบถึงบิสซิเนส รูลที่ประยุกต์ใช้ในการทำงานเพื่อลบคำสั่งซื้อ

**คุณสมบัติ (Properties)** แสดงถึงลักษณะของโอเล่ ออโตเมชัน เซิร์ฟเวอร์ ออบเจกต์ ที่มีความหมายตรงตัวกับผู้ใช้,ออบเจกต์ หรือทั้งสองอย่าง คุณสมบัติมักถูกใช้เพื่อกำหนดว่าการกระทำใดควรจะเกิดขึ้นเมื่อเมธอดถูกเรียกใช้

#### 2.4.3 ข้อดีในการแยกส่วนบริการกฎเกณฑ์ข้อจำกัดของระบบออกมาจากส่วนไคลเอนต์

ในสถาปัตยกรรมแบบเดิม(Two-tier) ส่วนบิสซิเนสนี้จะอยู่ที่ส่วนไคลเอนต์ การย้ายส่วนนี้แยกมาต่างหากทำให้มีข้อดีหลายอย่างคือ

1. มีประโยชน์มากในกรณีที่มีการแก้ไขกฎเกณฑ์ของระบบ สามารถแก้ไขได้ทีละส่วนนี้โดยตรง ไม่ต้องไปแก้ที่เครื่องไคลเอนต์ทุกเครื่อง ( ในระบบเดิม หากมีการเปลี่ยนแปลงเมื่อใดแล้วจะต้องไปแก้ที่เครื่องไคลเอนต์ทุกๆเครื่อง ซึ่งเสียเวลามาก ยิ่งถ้าหากมีการเปลี่ยนแปลงกฎเกณฑ์ บ่อยๆแล้ว ก็จะทำให้เสียเวลามากขึ้นไปอีก )
2. การตรวจสอบระบบสามารถทำได้ง่าย โดยตรวจสอบที่ส่วนนี้โดยตรง
3. การวัดประสิทธิภาพของระบบสามารถทำได้จากส่วนกลางนี้
4. สามารถเพิ่มกฎเกณฑ์ข้อจำกัดได้จากส่วนกลางนี้

#### 2.4.4 การทำงานกับส่วนบริการข้อมูล

การจัดการกับระบบจัดการข้อมูลเชิงสัมพันธ์ (Relational Database Management System:RDBMS) เมื่อมีการทำงานกับส่วนให้บริการข้อมูลฐานข้อมูลเชิงสัมพันธ์ บิสซิเนส เซิร์ฟเวอร์จะใช้ฟังก์ชันการทำงานที่ผู้ให้บริการข้อมูล (Data Service) มีให้ ระบบจัดการฐานข้อมูลส่วนใหญ่ มักจะมีความสามารถในการให้บริการเข้าถึงข้อมูลในส่วนให้บริการ ได้โดยผ่านทางสตอร์ โพรซีเยอร์

สตอร์ โพรซีเยอร์ คือกลุ่มของคำสั่งภาษาเอสคิวแอล ที่ถูกแปลไว้ก่อน (Pre-compiled) แล้ว โดยผู้ให้บริการข้อมูล และมีแผนการทำงานที่ถูกกำหนดไว้ก่อนแล้ว(pre-determine execution plan) สตอร์ โพรซีเยอร์ จะให้ประโยชน์มากกว่าได้คอสคิวแอลที่ถูกฝังไว้ในบิสซิเนส เซิร์ฟเวอร์ เพราะจะช่วยให้ประสิทธิภาพและความสามารถในการควบคุมฐานข้อมูล มีเพิ่มมากขึ้น

ส่วนที่สำคัญที่ควรเข้าใจ คือ หลักการของกฎเกณฑ์ต่างๆของระบบ ควรจะอยู่ใน ส่วนบิตชีเนส เซิร์ฟเวอร์ ไม่ควรกระจายไปเก็บในส่วนเก็บกฎเกณฑ์ของระบบ และส่วนให้บริการข้อมูล เหตุผลคือ สตอร์ โพรซีเยอร์ ควรจะใช้สำหรับการจัดการกับข้อมูลแบบง่ายๆ เช่น การเรียกข้อมูล (retrieve) และปรับปรุงข้อมูลให้ทันสมัย (Update) สตอร์ โพรซีเยอร์ ไม่ได้ถูกสร้างขึ้นเพื่อการจัดการกับข้อมูล

ตัวอย่างการเรียกใช้สตอร์ โพรซีเยอร์ ในโปรแกรม เพื่อลบข้อมูล

```
db.BeginTrans
```

```
'open a handle to the delete orders stored procedure
```

```
Set qryDeleteOrder = db.CretePrepareStatement (_  
    " sp_Delete_Order ",_  
    " {call sp_DeleteOrder(?)})"
```

```
qryDeleteOrder.Close
```

#### 2.4.4.1 การติดต่อกับส่วนให้บริการข้อมูลเชิงสัมพันธ์ (Connecting to RDBMS Service)

เมื่อมีการสร้างโปรแกรมประยุกต์ให้ใช้งานกับฐานข้อมูลเชิงสัมพันธ์ ที่ทำงานเป็น แบบสถาปัตยกรรมทรีทีเยอร์ ต้องตระหนักถึงลักษณะสำคัญของการติดต่อกับฐานข้อมูล 4 อย่างคือ

1. การเข้าถึง โดยเน้นการมองเป็นออบเจ็คต์ (Object based access)
2. ประสิทธิภาพ (Solid performance)
- 3.สนับสนุนลักษณะสำคัญของฐานข้อมูลแบบปุ้ะเอียดและสามารถใช้เทคนิคขั้นสูงในการติดต่อกับฐานข้อมูลได้ (Support for advance database feature)
- 4.ความสามารถในการทำงานกับระบบจัดการฐานข้อมูลหลายชนิด (Ability to operate against multiple DBMS)

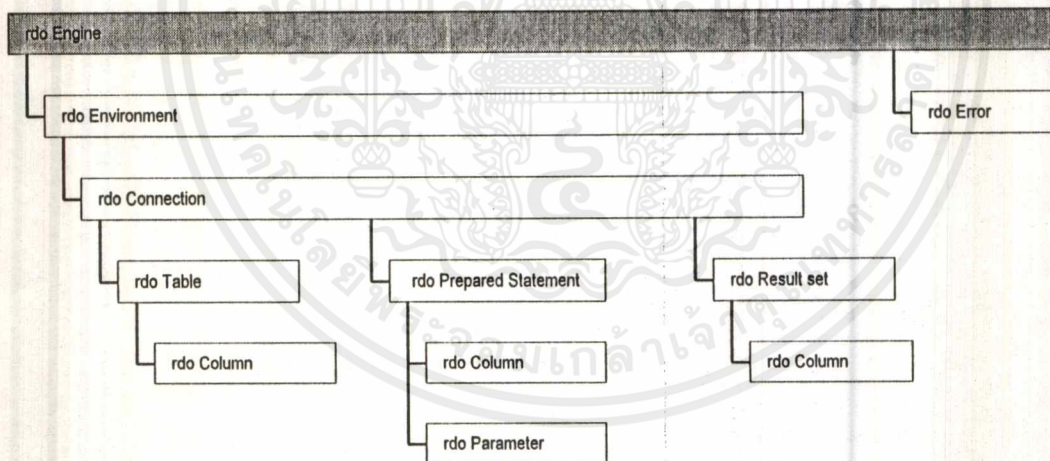
ซึ่งในปัจจุบันมีเพียงรีโมทดาต้าออบเจ็คต์ ( Remote Data Object ) เท่านั้นที่สนับสนุน ลักษณะสำคัญทั้ง 4 ประการดังกล่าว

#### 2.4.4.2 รีโมท คาด้า ออบเจ็กต์(Remote Data Object :RDO)

ด้วยความสามารถของวิซวลเบสิก 4.0 เอนเตอร์ไพรส์ อีดิชั่นมีเพียงรีโมท คาด้าออบเจ็กต์ (RDO) ที่สามารถสนับสนุน จุดประสงค์ที่ต้องการทั้ง 4 อย่างได้ข้างต้นได้ รีโมท คาด้า ออบเจ็กต์ มีความสามารถในการเข้าถึงฐานข้อมูลหลายๆชนิดที่สนับสนุนโอดีบีซี รีโมท คาด้า ออบเจ็กต์ มีลักษณะการติดต่อกับออบเจ็กต์ ซึ่งคล้ายกับ คาด้า แอกเซส ออบเจ็กต์ (Data Access Object : DAO) และอยู่ในโครงสร้างชั้นบนของโอดีบีซี สามารถเผยถึงลักษณะสำคัญของโอดีบีซี เอพีไอ(ODBC API)โดยไม่จำเป็นต้องรู้รายละเอียดส่วนลึกภายใน

##### ลักษณะสำคัญของ RDO

1. การเข้าถึงโอดีบีซี คาด้าซอร์ส (ODBC Data Source)
2. สนับสนุนฟังก์ชันการทำงานโดยเฉพาะของผู้ให้บริการ
3. การประมวลผลเพื่อเรียกใช้ข้อมูลแบบ อะซิงโครนัส
4. ต้องการหน่วยความจำน้อย
5. การเข้าถึงข้อมูลโดยมีโครงสร้างเป็นออบเจ็กต์



รูปที่ 2.5 แสดงลำดับชั้นของ RDO ออบเจ็กต์

(หมายเหตุ : rdo Engine เป็นออบเจ็กต์ ส่วน rdo ตัวอื่นๆด้านล่างเป็นทั้งออบเจ็กต์และคอลเลกชัน)

โครงสร้างแบบลำดับชั้นของ RDO เป็นโครงร่าง(framework) ที่แอปพลิเคชันใช้สำหรับถ่ายเทข้อมูลโดยผ่าน รีโมท โอดีบีซี คาด้าเบส ความสัมพันธ์ระหว่างออบเจ็กต์แสดงถึงโครงสร้างลอจิกคอลของฐานข้อมูล

โครงร่างของ RDO ประกอบด้วยออบเจ็กต์ 2 ชนิด คือ

- ออบเจ็กต์ที่จัดการการติดต่อกับรีโมท คาด้า ซอร์ส
- ออบเจ็กต์ที่อนุญาตให้โปรแกรมประยุกต์เข้าถึงได้โดยผ่านคาด้า ซอร์สเดียวกัน

ตัวอย่างการใช้ OpenConnection method เพื่อติดต่อใช้งานดาต้าซอร์ส

```
'set up the default environment
```

```
rdoEngine.rdoDefaultUser = UserId
```

```
rdoEngine.rdoDefaultPassword = Password
```

```
'open a connection to the database
```

```
Set db = rdoEngine.rdoEnvironment(0).openConnection(_  
    DATA_SOURCE,_  
    rdDriverNoPrompt,_  
    False)
```

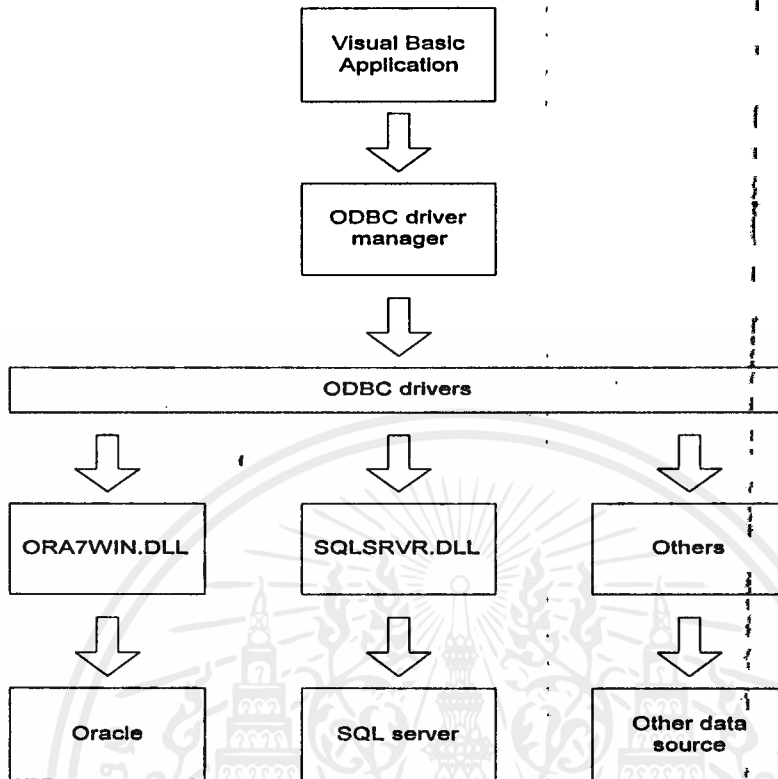
สำหรับรายละเอียดปลีกย่อยต่างๆของ RDO ออบเจกต์จะรวมอยู่ในภาคผนวก

## 2.5 โอเพ่นดาต้าเบสคอนเนคตีวิตี:โอดีบีซี (Open Database Connectivity,ODBC)

เป็นมาตรฐานการอินเตอร์เฟซซึ่งจะทำให้โปรแกรมประยุกต์สามารถเข้าถึงข้อมูลที่เกี่ยวข้องในระบบฐานข้อมูล ซึ่งถูกดูแลโดยระบบจัดการฐานข้อมูล(DBMS-Database Management System) โดยใช้คำสั่งภาษา SQL เป็นมาตรฐานในการติดต่อ โดยที่ก่อนที่จะมีการพัฒนาโอดีบีซีขึ้นมา นั้น การพัฒนาแอปพลิเคชันจะขึ้นกับซีบีเอ็มเอส ซึ่งจะมีลักษณะเป็น เอมเบด เอสคิวเอล (Embedded SQL) ประสิทธิภาพของเอ็มเบด เอสคิวเอลนี้ จะขึ้นอยู่กับฮาร์ดแวร์และสภาพแวดล้อมของระบบปฏิบัติการ ซึ่งจะทำให้ซอสโค้ดจะต้องถูกคอมไพล์ (Compile) ใหม่สำหรับแต่ละสภาพแวดล้อม

การที่แอปพลิเคชันใช้โอดีบีซีเป็นตัวติดต่อเพื่อการเข้าถึงข้อมูลนั้นจะทำให้แอปพลิเคชันสามารถเข้าถึงข้อมูลที่ถูกจัดเก็บโดยซีบีเอ็มเอสที่แตกต่างกันได้ ซึ่งทำให้ผู้พัฒนาแอปพลิเคชันสามารถนำแอปพลิเคชันที่พัฒนาแล้วไปใช้กับซีบีเอ็มเอสตัวใดก็ได้ โดยที่ผู้ใช้สามารถเพิ่มเติม โมดูล (Database Driver) ที่จะลิงก์แอปพลิเคชันให้ใช้กับซีบีเอ็มเอสที่ต้องการได้

โดยที่วิซวลเบสิก จะเตรียมโปรแกรมสำหรับการติดต่อกับฐานข้อมูลชนิดนั้น ที่เรียกว่า "ไดรฟ์เวอร์"(Driver) ไว้ให้ หากต้องการใช้งานข้อมูลชนิดอื่นๆ นอกจากนี้ ต้องหาไดรฟ์เวอร์สำหรับฐานข้อมูลชนิดนั้นมาติดตั้งเพิ่ม



รูปที่ 2.6 โครงสร้างแต่ละชั้นของ ODBC

**หมายเหตุ** DLL(Dynamiccally Linked Libraries)เป็นไคร์ฟเวอร์ที่ทำหน้าที่แปลงโอดีบีซีไปเป็นฟังก์ชันที่คาค้าซอร์ส สามารถเรียกใช้งานได้ซึ่งก็คือการแปลงจากวากยสัมพันธ์ของโอดีบีซี เอสคิวแอล (ODBC SQL Syntax) ไปเป็นวากยสัมพันธ์ (Syntax) ที่ คาค้าซอร์สเข้าใจ

โอดีบีซี เป็นมาตรฐานสำหรับการสร้างไคร์ฟเวอร์ของฐานข้อมูลตามรูปที่ 2.6 โดย แอปพลิเคชันโปรแกรมที่ใช้คือ วิวทเบสิก จะใช้งานโอดีบีซี ผ่านคำสั่งมาตรฐานคือ โอดีบีซี เอพีไอ (ODBC API:Open Database Connectivity Application Program Interface) โอดีบีซี ไคร์ฟเวอร์จะใช้คาค้าซอร์สตามแอปพลิเคชันโปรแกรม โดยใช้คำสั่งของโอดีบีซี (ODBC Command) และสามารถใช้งานคาค้าซอร์สอื่นๆที่มี โอดีบีซี ไคร์ฟเวอร์ ได้โดยไม่ต้องปรับเปลี่ยนโค้ด

### 2.5.1 ส่วนประกอบของโอดีบีซี (ODBC Compenents)

2.5.1.1 แอปพลิเคชัน(Application)คือโปรแกรมที่ทำการใช้งานข้อมูลจากแหล่งของข้อมูล ซึ่งเขียนขึ้นโดยภาษาโปรแกรมมิ่งต่างๆไป มีหน้าที่หลักในการเข้าถึงข้อมูลภายในฐานข้อมูล โดยเรียกใช้ฟังก์ชันของโอดีบีซี,การส่งคำสั่งเอสคิวแอล และการรับข้อมูลผลลัพธ์

2.5.1.2 ไดรฟ์เวอร์ แมเนเจอร์ ( Driver Manager) เป็นส่วนที่จัดการเกี่ยวกับการเรียกใช้ .DLL ต่างๆ ตามความต้องการของแอปพลิเคชันโปรแกรม โดยจะทำการโหลดไดรฟ์เวอร์ในนามของแอปพลิเคชัน

2.5.1.3 ไดรฟ์เวอร์ (Driver) ในที่นี้หมายถึง .DLL ต่างๆ ซึ่งทำหน้าที่รับคำสั่ง SQL จากแอปพลิเคชัน แล้วส่งคำสั่ง SQL ต่อไปยังดาต้าซอร์ส (Data source) ด้วยชุดคำสั่งที่ดาต้าซอร์สสามารถเข้าใจได้ แล้วรอรับผลตอบสนอง (Result) เพื่อส่งต่อไปยังแอปพลิเคชัน นอกจากนี้ยังสามารถแก้ไขปรับปรุงรูปแบบการร้องขอจากแอปพลิเคชันเพื่อให้สอดคล้องกับดีบีเอ็มเอส

2.5.1.4 ดาต้าซอร์ส (Data source) ประกอบด้วยข้อมูลที่ผู้ใช้ต้องการจะเข้าถึง รวมไปถึงระบบปฏิบัติการ (OperatingSystem : OS) ที่เกี่ยวข้อง ,ดีบีเอ็มเอส(DBMS)และเน็ตเวิร์คแพลตฟอร์ม (Network Platform ) (ถ้ามี)

## 2.5.2 ส่วนประกอบในการอินเตอร์เฟสกับ ODBC

2.5.2.1 ไลบรารี(Libraries)ของฟังก์ชันซึ่งจะทำให้แอปพลิเคชันติดต่อกับดีบีเอ็มเอสเพื่อที่จะเอ็กรหัสคิวรี่คำสั่งแล้วดึงข้อมูลออกมา

2.5.2.2 ไวยากรณ์ของเอสคิวแอลซึ่งยึดถือตามมาตรฐานของ X/Open และ SQL Access Group(SAG)SQL CAE Specification(1992)

2.5.2.3 มาตรฐานรหัสข้อผิดพลาด (Error code)

2.5.2.4 วิธีมาตรฐานในการติดต่อและการล็อกเข้าดีบีเอ็มเอส

2.5.2.5 มาตรฐานประเภทของข้อมูล

## 2.5.3 แอปพลิเคชัน จะทำงานต่างๆผ่าน โอดีบีซีอินเตอร์เฟสดังต่อไปนี้

1. ทำการร้องขอการติดต่อไปยังดาต้าซอร์ส
2. ส่งเอสคิวแอลไปยังดาต้าซอร์ส
3. กำหนดเนื้อที่และรูปแบบของข้อมูล เพื่อการประมวลผลคำสั่งเอสคิวแอล
4. ร้องขอผลลัพธ์
5. ประมวลผลข้อผิดพลาดที่อาจจะเกิดขึ้น
6. รายงานผลลัพธ์กลับไปยังผู้ใช้
7. ร้องขอการคอมมิต(Commit) และการ โรลแบค (Rollback) สำหรับการควบคุม

ทรานแซคชัน

8. ยกเลิกการติดต่อจากดาต้าซอร์ส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5.4 ไดรฟ์เวอร์แมเนเจอร์

ไดรฟ์เวอร์แมเนเจอร์เป็นไดนามิกลิงก์ไลบรารี (Dynamic-link library-DLL) จุดประสงค์หลักของไดรฟ์เวอร์แมเนเจอร์ คือ การโหลดไดรฟ์เวอร์ โดยจะทำตามลำดับดังต่อไปนี้

1. ใช้ ODBC.INI ไฟล์ แมพ(map) ชื่อของคาคำขอส เพื่อค้นหาไดรฟ์เวอร์ดีแอลแอล

2. ทำการประมวลผลโอดีบีซี อินิเชียลไลเซชันคอล (ODBC Initialization calls)

3. จัดการ โอดีบีซีฟังก์ชันสำหรับแต่ละไดรฟ์เวอร์

4. จัดการพารามิเตอร์ให้ถูกต้องสำหรับการเรียกใช้โอดีบีซี

## 2.5.5 ไดรฟ์เวอร์

ไดรฟ์เวอร์เป็นดีแอลแอลที่อิมพลีเมนต์โอดีบีซีฟังก์ชันคอลและทำการติดต่อกับข้อมูลที่คาคำขอส โดยไดรฟ์เวอร์แมเนเจอร์จะทำการโหลดไดรฟ์เวอร์เมื่อแอปพลิเคชันได้เรียกใช้ฟังก์ชัน SQLBrowseConnect SQLConnect หรือ SQLDriverConnect โดยไดรฟ์เวอร์จะทำตามขั้นตอนดังต่อไปนี้เพื่อตอบรับการเรียกโอดีบีซีฟังก์ชันของแอปพลิเคชัน

1. สร้างการติดต่อไปยังคาคำขอส

2. ส่งการร้องขอไปยังคาคำขอส

3. แปลงข้อมูลให้เป็นรูปแบบอื่นถ้าเป็นการเรียกโดยแอปพลิเคชัน

4. ส่งผลลัพธ์กลับไปยังแอปพลิเคชัน

5. ถ้าเกิดข้อผิดพลาดขึ้นจะมีการ ส่ง ข้อผิดพลาด กลับไปยังแอปพลิเคชันที่เรียก

6. ทำการประกาศและจัดการเกี่ยวกับเคอร์เซอร์ (Cursors) ถ้าจำเป็น

7. อินิเชียลทรานแซกชันถ้าคาคำขอสนั้นๆต้องการ

## 2.5.6 ชนิดของไดรฟ์เวอร์

โอดีบีซีกำหนดชนิดของไดรฟ์เวอร์ไว้ 2 รูปแบบคือ

2.5.6.1 ซิงเกิลเทียร์ (Single-tier) ไดรฟ์เวอร์ชนิดนี้จะมีหน้าที่เอ็กซีคิวทีฟฟังก์ชันที่ส่งมาจากโอดีบีซี รวมทั้งคำสั่งเอ็สคิวแอลด้วย ดังนั้นไดรฟ์เวอร์ชนิดนี้จึงมีหน้าที่บางส่วนของคาคำขอสอยู่ด้วย

2.5.6.2 มัลติเทียร์ (Multiple-tier) ไดรฟ์เวอร์ชนิดนี้จะมีหน้าที่เอ็กซีคิวทีฟเฉพาะฟังก์ชันที่ส่งมาจากโอดีบีซีเท่านั้น ส่วนคำสั่งเอ็สคิวแอล จะส่งไปให้กับตัวคาคำขอส

ซึ่งระบบหนึ่งอาจจะประกอบด้วยไดรฟ์เวอร์ทั้งสองชนิดเลขก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5.7 ระดับข้อตกลงของโอดีบีซี (ODBC Conformance Levels)

ในการใช้มาตรฐาน โอดีบีซีในการเชื่อมต่อนั้นจะต้องมีข้อตกลงมาตรฐานเพื่อที่จะให้คนพัฒนาส่วนของไครฟ์เวอร์ สามารถครอบคลุมฟังก์ชันต่างๆ ที่จำเป็นได้ครบ และทางด้านแอปพลิเคชันสามารถเลือกใช้ไครฟ์เวอร์ตามฟังก์ชันมาตรฐานได้โดยง่าย

โดยระดับข้อตกลงของไครฟ์เวอร์แบ่งออกเป็น

### 2.5.7.1 ODBC API ซึ่งแบ่งย่อยออกได้เป็น

- CORE API อ้างอิงตามมาตรฐานของ X/OPEN & SQL Access Group Call Level Interface
- Level1 API
- Level2 API

ซึ่งสองระดับท้ายสุดถือว่าเป็นระดับ เอ็กเทนเคดฟังก์ชันนอลลิตี (Extended Functionality)

### 2.5.7.2 ODBC SQL Grammar และ ODBC SQL datatypes แบ่งออกเป็น

- Minimum SQL Grammar ซึ่งเป็นระดับพื้นฐาน
- Core SQL Grammar ยึดตามมาตรฐานที่ระบุใน X/OPEN & SQL Access Group SQL CAE Specification (1992) Extended SQL Grammar Support Extension SQL ของดีบีเอ็มเอสทั่วไป

## 2.5.8 ข้อดีในการอินเตอร์เฟซกับโอดีบีซี

2.5.8.1 สามารถกำหนดคำสั่งเอสคิวแอลได้ทั้งในซอร์สโค้ด (Source Code) ของแอปพลิเคชันนั้น และขณะกำลังทำงาน (Run time)

2.5.8.2 ซอร์สโค้ดเดียวกันสามารถเข้าถึงข้อมูลที่เก็บอยู่บนดีบีเอ็มเอสคนละตัวกันได้

2.5.8.3 โพรโทคอลในการสื่อสารข้อมูล (Data Communication Protocol) ระหว่างแอปพลิเคชันแต่ละตัวไม่มีความสำคัญ

2.5.8.4 สามารถปรับแต่งรูปแบบของข้อมูลที่รับและส่งระหว่างแอปพลิเคชันกับตัวโอดีบีซีได้ตามต้องการ

สรุปคือ ความสามารถของโอดีบีซีนี้เองที่ใช้ติดต่อไปยังฐานข้อมูลในระบบอื่นๆ จะช่วยให้โปรแกรมของเราเป็นอิสระจากฐานข้อมูลมากขึ้น โดยที่เราสามารถสร้างโปรแกรมขึ้นมา



ทดสอบกับฐานข้อมูลจำลองในเครื่องของเราเองก่อนได้ จากนั้นเมื่อนำไปใช้จริงกับฐานข้อมูลที่อยู่บนเครื่องเซิร์ฟเวอร์ก็เพียงแต่เปลี่ยนที่คำสั่ง โอเพนดาต้าเบส (Opendatabase) เพียงคำสั่งเดียวก็เสร็จ

## 2.6 สถาปัตยกรรม เทียร์ที่สาม (Architecting Tier 3:)

### ส่วนบริการข้อมูล (Data Service)

ส่วนนี้จะจัดการการร้องขอข้อมูลจากส่วนบิสซิเนส เซอร์วิส โดยมีระบบจัดการฐานข้อมูล (DBMS : Database Management System) เป็นตัวจัดการข้อมูล, ให้บริการข่าวสารข้อมูล, ดูแลรักษา ความถูกต้องของข้อมูลภายในองค์กร ลักษณะการจัดเก็บข้อมูลก็ไม่แตกต่างจากในระบบเดิม (ทูเทียร์) โดยข้อมูลจะถูกจัดเก็บไว้ในหน่วยเก็บข้อมูล (data storage) และถูกเรียกใช้โดยชั้นของ ส่วนให้บริการกฎเกณฑ์ข้อจำกัดของระบบ (Business Service) ซึ่งข้อมูลที่เก็บไว้ที่ชั้นนี้จะมี ความสัมพันธ์กับบิสซิเนส ออบเจ็ค

ข้อมูลไม่จำเป็นจะต้องถูกจัดเก็บแค่ภายในระบบจัดการฐานข้อมูลเท่านั้น มันสามารถอยู่ บนเมนเฟรม, อิมเมจ เซิร์ฟเวอร์, อินเทอร์เน็ต หรือแหล่งข้อมูลอย่างอื่นก็ได้ แต่ใน ทรีเทียร์ แหล่ง ข้อมูลจะถูกจัดเก็บ และจัดการโดยคิวบีเอ็มเอส เมื่อเทคโนโลยีมีการพัฒนาขึ้น เราอาจจะเห็นการจัด เก็บแหล่งข้อมูล ในรูปแบบอื่นที่ดี และมีความยืดหยุ่นมากกว่านี้

### 2.6.1 ระบบจัดการฐานข้อมูลของส่วนบริการข้อมูล (DBMS Data Service)

**ฐานข้อมูล :** เป็นกลุ่มของข้อมูล ที่มีความสัมพันธ์ในระดับลอจิคอล (logical) และ ถูกออกแบบเพื่อให้ครอบคลุมความต้องการของผู้ใช้หลายๆคน

**ระบบจัดการฐานข้อมูล :** เป็นแอปพลิเคชันที่ใช้เพื่อสร้าง, บำรุงรักษา และควบคุม การเข้าถึงข้อมูลข่าวสารที่ถูกจัดเก็บในฐานข้อมูล

**ฐานข้อมูลเชิงสัมพันธ์ :** เป็นฐานข้อมูลที่ประกอบไปด้วยข้อมูลที่ถูกจัดเก็บตาม ความสัมพันธ์ของข้อมูลในรูปแบบของตาราง

### 2.6.2 ฐานข้อมูลเชิงสัมพันธ์

โครงสร้างฐานข้อมูลเชิงสัมพันธ์ที่มีความสัมพันธ์ระหว่างเอนติตี้แบ่งเป็น 3 ชนิด ดังนี้คือ

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

### 2.6.2.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างเอนทิตี หมายถึง เมื่อเอนทิตีหนึ่ง มีข้อมูลของคีย์หลักค่าหนึ่ง ค่าข้อมูลดังกล่าวก็จะมีความสัมพันธ์กับค่าข้อมูลของคีย์หลักของอีก เอนทิตีหนึ่งเพียงค่าเดียวเท่านั้น เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่าง เอนทิตีนักเรียน กับ เอนทิตีผู้ปกครอง เป็นแบบหนึ่งต่อหนึ่งแล้ว หมายความว่า การที่เราจะอ้างอิงถึงนักเรียนคนใดคนหนึ่งก็จะสามารถอ้างอิงถึงผู้ปกครองได้เพียงคนเดียวเท่านั้น และในทางตรงกันข้ามก็ต้องเป็นจริง ด้วย คือ เมื่อเราอ้างอิงถึงผู้ปกครองคนใดคนหนึ่งแล้วก็สามารถอ้างอิงถึงนักเรียนได้เพียงคนเดียว เท่านั้น

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	b
C	c

ตารางที่ 2.3 ตารางแสดงความสัมพันธ์ของเอนทิตีแบบหนึ่งต่อหนึ่ง

### 2.6.2.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตี หมายถึง เมื่อเอนทิตีหนึ่ง มีข้อมูลของคีย์หลักค่าหนึ่ง ค่าข้อมูลดังกล่าว ก็จะมีความสัมพันธ์กับค่าข้อมูลของคีย์หลักของอีก เอนทิตีหนึ่งได้หลายค่า เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่าง เอนทิตีนักเรียนกับเอนทิตีผู้ปกครอง เป็นแบบ หนึ่งต่อกลุ่มแล้ว จะหมายความว่า การที่เราอ้างอิงถึงนักเรียนคนใดคนหนึ่ง ก็จะ สามารถ อ้างอิงถึงผู้ปกครองคนใดคนหนึ่งได้เพียงคนเดียวเท่านั้น และในทางตรงกันข้ามก็หมายความว่า เมื่อเราอ้างอิงถึงผู้ปกครองคนใดคนหนึ่งแล้วก็สามารถอ้างอิงถึงนักเรียนได้เพียงคนเดียว เท่านั้น แต่ผู้ปกครองที่เราอ้างอิงเป็นคนละคนกันก็จะอ้างอิงถึงนักเรียนคนเดียวกันก็เป็นได้

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	a
C	c

ตารางที่ 2.4 ตารางแสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม

### 2.6.2.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตี หมายถึง ค่าข้อมูลของคีย์หลักของเอนทิตีหนึ่งที่แตกต่างกันอาจอ้างอิงถึงค่าข้อมูลของคีย์หลักของอีกเอนทิตีหนึ่งได้ค่าเดียวหรือหลายค่าก็ได้ เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่างเอนทิตีนักเรียนกับเอนทิตีผู้ปกครองเป็นแบบกลุ่มต่อกลุ่มแล้วก็จะหมายความว่า การที่เราอ้างอิงถึงนักเรียนคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงผู้ปกครองคนเดียวกันได้ และในทางกลับกัน การที่เราอ้างอิงถึงผู้ปกครองคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงนักเรียนคนเดียวกันก็ได้

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	a
C	c
C	d

ตารางที่ 2.5 ตารางแสดงความสัมพันธ์ของเอนทิตีแบบกลุ่มต่อกลุ่ม

นอกจากนี้ เอนทิตียังมีเอนทิตีบางประเภทที่การอ้างอิงด้วมัน ได้อย่างสมบูรณ์นั้น จะต้องอ้างอิงเอนทิตีอื่นเสมอ เช่น หากระบบฐานข้อมูลของบุคลากรในโรงเรียนมีเอนทิตีของลูกนักรการาร โรงด้วยแล้ว การอ้างอิงถึงเอนทิตีนี้ได้จะต้องอ้างอิงถึงเอนทิตีนักรการาร โรงด้วยเสมอ เอนทิตีประเภทนี้เราเรียกว่า เอนทิตีชนิดอ่อน (Weak entity)

### 2.6.3 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์

#### กฎของความคงสภาพ (Integrity rule)

กฎของความคงสภาพของโมเดลเชิงสัมพันธ์ (Relational model) เป็นทฤษฎีที่ช่วยยืนยันความถูกต้องของความสัมพันธ์ระหว่างข้อมูลว่า รีเลขันใดที่เป็นไปตามกฎของความคงสภาพนี้แล้วย่อมจะมีความสัมพันธ์ระหว่างข้อมูลอย่างถูกต้องอยู่ตลอดเวลา ไม่ว่ารีเลขันนั้นจะมีการเปลี่ยนแปลงแก้ไขข้อมูลไปในรูปแบบใดก็ตาม

กฎของความคงสภาพมีความหมายอยู่ 2 ลักษณะ คือ กฎความคงสภาพของเอนทิตี (entity integrity rule) และกฎความคงสภาพของการอ้างอิง (referential integrity rule) ดังอธิบายได้ดังนี้

กฎความคงสภาพของเอนติตี้ กล่าวว่า

“แอตทริบิวต์ทุกตัวที่เป็นส่วนของคีย์หลักจะไม่อนุญาตให้มีค่าว่าง” หมายความว่า คีย์หลักของทุกรีเลชันจะไม่สามารถเก็บค่าข้อมูลที่เป็นค่าว่างได้ เหตุผลของข้อกำหนดนี้ก็คือ เพื่อให้การเข้าถึงข้อมูลในแถวใดๆ ของรีเลชันมีความเป็นไปได้เสมอ เพราะถ้าคีย์หลักของแถวใดมีค่าข้อมูลเป็นค่าว่างแล้ว ก็จะส่งผลให้การเข้าถึงข้อมูลในแถวนั้นไม่สามารถกระทำได้อย่างแน่นอน

กฎความคงสภาพของการอ้างอิง กล่าวว่า

“ถ้าเรามีรีเลชัน R2 ซึ่งมี FK เป็นคีย์นอกที่อ้างอิงถึงคีย์หลัก PK ในรีเลชัน R1 สำหรับทุกค่าของ FK ใน R2 จะต้อง

ก. มีค่าเท่ากับค่า PK ในแถวใดแถวหนึ่งในรีเลชัน R1

ข. มีค่าของแอตทริบิวต์ทุกตัวใน FK เป็นค่าว่าง”

หมายความว่า แอตทริบิวต์ใดๆ ที่เป็นคีย์หลักของรีเลชันหนึ่ง เมื่อมีการนำแอตทริบิวต์นั้นไปเป็นคีย์นอกของอีกรีเลชันหนึ่ง การเป็นคีย์นอกของแอตทริบิวต์นั้นจะต้องมีโดเมนเป็นโดเมนเดียวกันกับแอตทริบิวต์ที่เป็นคีย์หลัก ทั้งนี้ ก็เพื่อให้การนำรีเลชันมาใช้งานร่วมกัน (การนำรีเลชันมา join กัน) กระทำได้อย่างถูกต้อง คือ ทุกแอตทริบิวต์ที่เป็นคีย์นอกจะต้องมีข้อมูลซ้ำกับข้อมูลของแอตทริบิวต์ที่เป็นคีย์หลักอย่างแน่นอน แต่อาจมีบางค่าข้อมูลของแอตทริบิวต์ที่เป็นคีย์หลักเป็นข้อมูลไม่อยู่ในโดเมนของแอตทริบิวต์ที่เป็นคีย์นอกก็ได้ นั่นคือ โดเมนของคีย์นอกจะต้องเล็กกว่าหรือเท่ากับโดเมนของคีย์หลักเสมอ

รีเลชัน R1

คีย์หลักของ R1	คีย์อื่นๆของ R1
A	1
B	2
C	3
D	4

รีเลชัน R2

คีย์หลักของ R2	คีย์นอกของ R1
a	A
b	B
c	B
d	C

ตารางที่ 2.6 ตารางแสดงคีย์หลักกับคีย์นอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.4 ฟังก์ชันการขึ้นต่อกัน (Functional dependency)

ฟังก์ชันการขึ้นต่อกันเป็นข้อกำหนดที่ช่วยให้เราเห็นถึงความสัมพันธ์ของแอตทริบิวต์ต่างๆ ที่อยู่ในรีเลชัน ทั้งนี้เพราะแอตทริบิวต์ต่างๆ ที่อยู่ในเอนทิตีเดียวกัน จะมีความสัมพันธ์กันเอง โดยที่ความสัมพันธ์นี้อาจเกี่ยวข้องหรือไม่เกี่ยวกับความสัมพันธ์ที่มันมีต่อกีย์หลักของเอนทิตีนั้นก็เป็นที่ได้ ซึ่งการที่แอตทริบิวต์เหล่านั้นมีความสัมพันธ์กันเองจะเป็นสิ่งที่เราต้องพิจารณาแยกออกเป็นรีเลชันย่อยๆ เพราะแอตทริบิวต์ของแต่ละรีเลชันก็ควรจะมีสัมพันธ์กับคีย์หลักของรีเลชันของตนเองเท่านั้น

กำหนดรีเลชัน R ถ้ามีแอตทริบิวต์ Y ของ R เป็นฟังก์ชันที่ขึ้นต่อแอตทริบิวต์ X ของ R เราสามารถเขียนแทนได้ด้วยสัญลักษณ์

$$R.X \text{ ----} \rightarrow R.Y$$

อ่านว่า R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y

หรือ R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y

หรือ R.Y ขึ้นอยู่กับ R.X

นิยาม R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y ก็ต่อเมื่อ ทุกค่าข้อมูลของแอตทริบิวต์ X ใน R จะมีค่าข้อมูลของแอตทริบิวต์ Y ใน R ได้เพียงค่าเดียวเสมอ โดยที่แอตทริบิวต์ X และ Y อาจจะมีคีย์แบบรวม (composite key) ก็ได้

รีเลชัน R

X	Y
a	1
b	2
a	1
b	1

ตารางที่ 2.7 ตารางแสดงฟังก์ชันการขึ้นต่อกัน

นิยาม R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y อย่างเต็มที่ (R.Y fully functionally dependent on R.X) ก็ต่อเมื่อ R.Y มีฟังก์ชันการขึ้นอยู่กับ R.X และไม่ขึ้นอยู่กับข้อมูลเพียงบางส่วนของ R.X โดยที่แอตทริบิวต์ X และ Y อาจจะเป็นคีย์แบบรวมก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีเลชัน R

X		Y
A	a	1
B	b	2
A	a	1
B	c	3

ตารางที่ 2.8 ตารางแสดงฟังก์ชันการขึ้นต่อกันอย่างเต็มที่

### 2.6.5 การออกแบบฐานข้อมูลรวมแบบรีเลชันแนล

การจัดเก็บข้อมูลจำนวนมาก โดยใช้เทคโนโลยีฐานข้อมูลช่วยในการจัดเก็บต้องคำนึงถึงวิธีการออกแบบระบบฐานข้อมูล และระบบจัดการฐานข้อมูล (DBMS) ที่เหมาะสม

ปัจจุบันวิธีการออกแบบฐานข้อมูลที่รู้จักกันแพร่หลายได้แก่ วิธีนอร์มัลไลเซชัน (Normalization) ซึ่งเป็นวิธีที่มีแนวคิดในการปรับปรุงคุณสมบัติของรีเลชันเป็นขั้นตอน อย่างมีระบบเพื่อลดโอกาสที่จะเกิดความผิดพลาดเนื่องจากการเปลี่ยนแปลงข้อมูล การออกแบบฐานข้อมูลด้วยวิธีนอร์มัลไลเซชันนี้ เริ่มต้นด้วยการ กำหนดยูนิเวอร์แซลรีเลชัน (Universal Relation) พร้อมทั้งกำหนดความสัมพันธ์ระหว่างแอตทริบิวต์ (Attribute) ต่างๆ ของยูนิเวอร์แซลรีเลชันนั้นในรูปของฟังก์ชันแนล ดีเพนเดนซ์ (Functional Dependency) มัลติแวลู ดีเพนเดนซ์ (Multivalued Dependency) และจอยน์ดีเพนเดนซ์ (Join Dependency) แล้วทำให้ รีเลชันในแต่ละขั้นตอนมีคุณสมบัติตรงตามที่กำหนดไว้ ตั้งแต่ขั้นตอนที่ 1 ถึงขั้นตอนที่ 5 ผลที่ได้ ในขั้นตอนสุดท้ายจะได้ Fifth Normal Form (5NF)

ถึงแม้วิธีนอร์มัลไลเซชันนี้จะเป็นวิธีที่มีขั้นตอนที่เป็นระบบ แต่ก็ยังเป็นวิธีที่ค่อนข้างยุ่งยากและซับซ้อน โดยเฉพาะอย่างยิ่งเมื่อระบบงานมีขนาดใหญ่ แอตทริบิวต์มีจำนวนมาก การออกแบบด้วยวิธี นอร์มัลไลเซชันจึงเป็นเรื่องลำบาก

ข้อเสนอการออกแบบระบบฐานข้อมูลรวมแบบรีเลชันแนล (Integrated Relational Database System) โดยใช้ในแอม (NIAM : Nijssen's Information System Analysis Method) เป็นเครื่องมือในการออกแบบ เนื่องจากแนวคิดที่ให้ Conceptual Schema มีพื้นฐานมาจากโครงสร้างภาษาธรรมชาติ ใช้รูปประโยคที่มี ประธาน กริยา กรรม วิธีแสดงรูปแบบความสัมพันธ์ของข้อมูล เป็นแบบจำลองที่มีความหมาย และมีเครื่องหมายแสดงความสัมพันธ์ของข้อมูล และข้อจำกัดของข้อมูลได้อย่างชัดเจน นอกจากนั้น ยังสามารถแปลง Conceptual Schema เป็น Relational Database

Schema ซึ่งจะอยู่ในรูปของ Fifth Normal Form และเนื่องจากวิธีการนี้ใช้รูปสัญลักษณ์แสดงความสัมพันธ์ของข้อมูลทำให้ง่ายต่อการเข้าใจ ดังนั้นการใช้วิธีนี้จึงเป็นการสะดวกในการออกแบบฐานข้อมูลของระบบงานใหญ่ๆ

เนื่องจากเป็นที่ยอมรับกันโดยทั่วไปแล้วว่าระบบฐานข้อมูลรีเลชันแนลเป็นเครื่องมือที่เหมาะสม ในการช่วยพัฒนาระบบสารสนเทศ (Information System) โดยเฉพาะอย่างยิ่งระบบสารสนเทศที่มีจุดประสงค์ ในการสนับสนุนการจัดการ (Management Information System : MIS) ดังนั้นการออกแบบฐานข้อมูลรวม สำหรับองค์กรจึงเป็นสิ่งสำคัญอย่างยิ่ง เพราะฐานข้อมูลที่ได้จะเป็นโครงสร้างข้อมูลหลัก สำหรับรองรับระบบสารสนเทศ ดังกล่าว

#### 2.6.6 การเขียนโปรแกรมส่วนดาต้า เซอร์วิสโดยใช้ สตอร์โพรซีเจอร์ (Stored Procedure)

สตอร์ โพรซีเจอร์ คือ กลุ่มของคำสั่งภาษาเอสคิวแอลที่ถูกแปลไว้ก่อนแล้ว (Pre-compiled) โดยผู้ให้บริการข้อมูล และจะมีแผนการทำงานที่ถูกกำหนดไว้ก่อนล่วงหน้าแล้ว (pre-determine execution plan) สตอร์ โพรซีเจอร์ จะให้ประโยชน์มากกว่าโค้ดเอสคิวแอลที่ถูกฝังไว้ใน บิสซิเนส เซิร์ฟเวอร์ เพราะจะช่วยให้ประสิทธิภาพและความสามารถในการควบคุมฐานข้อมูล มีเพิ่มมากขึ้น

ประโยชน์ของการใช้ สตอร์ โพรซีเจอร์

- เพิ่มประสิทธิภาพของระบบ
- เพิ่มความถูกต้องของข้อมูล (data integrity)

##### 1) เพิ่มประสิทธิภาพของระบบ ( performance )

สตอร์ โพรซีเจอร์เป็นคำสั่งของทรานแซก เอสคิวแอล (transact SQL) ที่ถูกคอมไพล์ในตอนที่เอ็กซีคิวครั้งแรกแล้วเก็บไว้ในฐานข้อมูล มีลักษณะเป็นยูนิท (unit) และสตอร์ โพรซีเจอร์จะถูกเอ็กซีคิวเมื่อมีการเรียกใช้ (กล่าวคือมีการเอ็กซีคิวครั้งต่อไปจะไม่ต้องถูกคอมไพล์ซ้ำอีก ทำให้รวดเร็วขึ้น ซึ่งต่างจากเอสคิวแอล สเตทเมนต์ (SQL Statement) ที่ถูกเขียนไว้ในโปรแกรมซึ่งโค้ดเหล่านี้ต้องถูกคอมไพล์ทุกครั้งที่มีการเรียกใช้ โปรแกรม)

##### 2) เพิ่มความถูกต้องของข้อมูล

การใช้สตอร์ โพรซีเจอร์ ทำให้การเปลี่ยนแปลงข้อมูลไม่กระทบกระเทือนถึงไพรมารี คีย์ (primary key) และการเปลี่ยนแปลงข้อมูลใดๆจะต้องผ่านการตรวจสอบ (Check) จากเงื่อนไขใน สตอร์ โพรซีเจอร์ก่อนว่าถูกต้องตามที่กำหนดไว้หรือไม่

## ตัวอย่าง Stored Procedure

```
if exists (select * from sysobjects where id = object_id('
dbo.sp_Add_OrderHdr') and sysstat & 0xf = 4)
    drop procedure dbo.sp_Add_OrderHdr.
GO
```

```
/* Object: Stored Procedure dbo.sp_Add_OrderHdr */
```

```
create procedure sp_Add_OrderHdr
@OrderID int = 0 OUTPUT,
@CustomerID int,
@OrderDate varchar(25) ,
@ShippedFlag char(1) = "N"

as
begin
/** Get the next order number if none passed **/
if @OrderID = 0
begin
    select @OrderID = max(OrderID) + 1 from OrderHdr
end
/** for the first order check for the null **/
if @OrderID = NULL
    select @OrderID = 100000000
insert into OrderHdr (OrderID, OrderDate, ShippedFlag,
CustomerID)
values(
@OrderID,
@OrderDate,
@ShippedFlag,
@CustomerID)
end
GO
```

## 2.7 ประโยชน์ของสถาปัตยกรรมแบบทรีเทียร์

### 2.7.1 การจัดการระบบมีประสิทธิภาพมากขึ้น

เนื่องจากการใช้โครงสร้างแบบเป็นเลเยอร์(Layer) ทำให้การเปลี่ยนแปลงที่เกิดขึ้นในส่วนใดส่วนหนึ่ง ไม่กระทบกระเทือนต่อส่วนบริการส่วนอื่น แต่ในสถาปัตยกรรมแบบทูเทียร์ ซึ่งส่วนฟรอนต์เอนด์ และแบคเอนด์จะมีความสัมพันธ์กันอย่างใกล้ชิด เมื่อมีการเปลี่ยนแปลงเกิดขึ้น การเปลี่ยนแปลงจะต้องเปลี่ยนแปลงที่ทุกๆเครื่องไคลเอนด์

### 2.7.2 กฎเกณฑ์ข้อบังคับสามารถถูกแบ่งปันได้

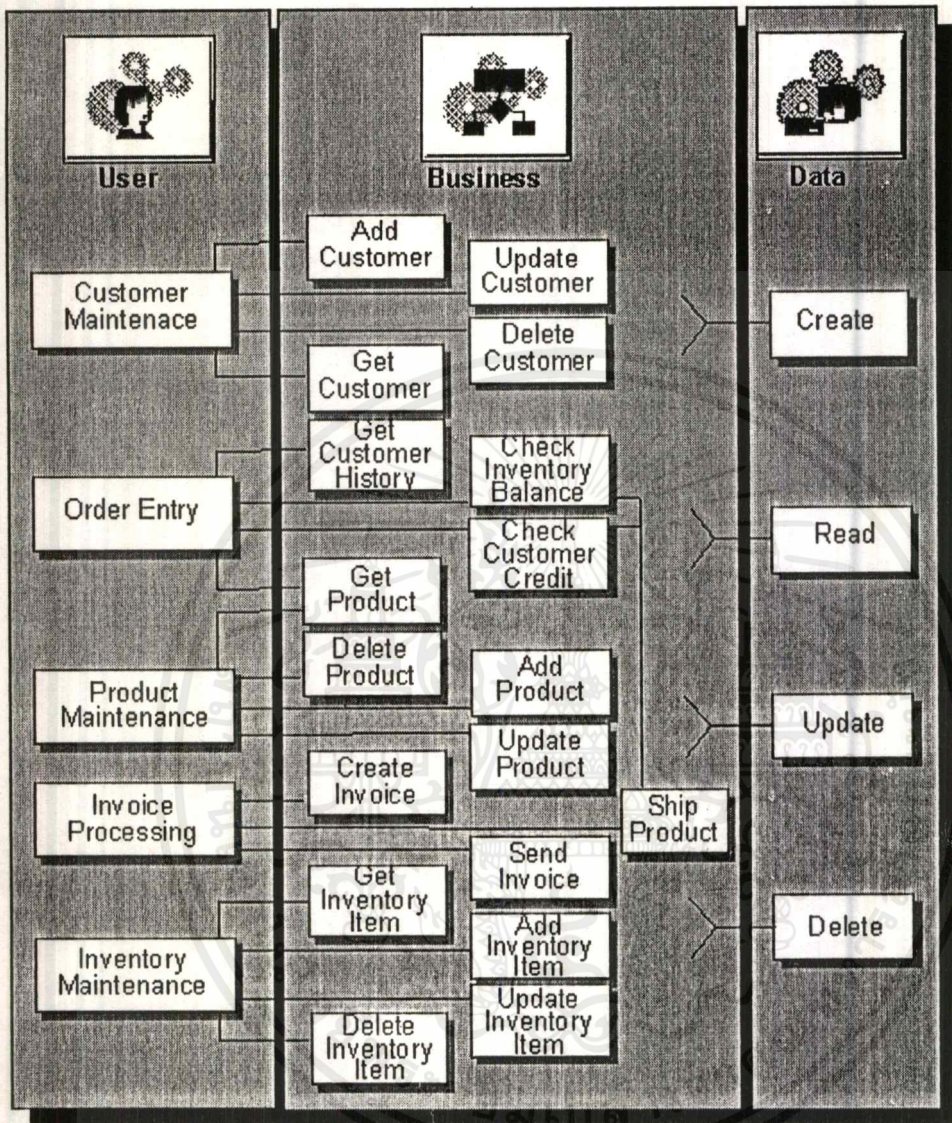
ส่วนกฎเกณฑ์ข้อบังคับไม่มีการกำหนดขอบเขต ทำให้ใช้ได้ในทุกๆแพลตฟอร์ม แอปพลิเคชัน (Desktop Application) โดยเมื่อแอปพลิเคชันจากไคลเอนด์หลายๆตัวมีคำสั่งการใช้งานเข้ามา ส่วนบริการกฎเกณฑ์ข้อบังคับไม่จำเป็นต้องสร้างโค้ดซ้ำซ้อนกัน และไม่ต้องเข้าไปดูแลในทุกๆส่วน เพราะโค้ดสามารถใช้ร่วมกันได้ (code reuse)

### 2.7.3 กฎเกณฑ์ข้อบังคับของระบบและโปรเซสมีความปลอดภัยขึ้น

การดูแลรักษาความปลอดภัยถูกย้ายจากส่วนบริการข้อมูลไปที่ส่วนบริการกฎเกณฑ์ข้อบังคับของระบบทำให้ส่วนไคลเอนด์ไม่ติดต่อกับส่วนบริการข้อมูลโดยตรง เราสามารถควบคุมแอปพลิเคชันที่ไปเรียกใช้ข้อมูลจากส่วนบริการข้อมูลได้ ไม่เพียงแต่สามารถควบคุมว่าข้อมูลใดถูกเปลี่ยน และเข้าไปใช้งานได้ที่เท่านั้น แต่ยังสามารถรู้ได้ว่าข้อมูลถูกเปลี่ยน และเข้าไปใช้งานอะไรได้ด้วย

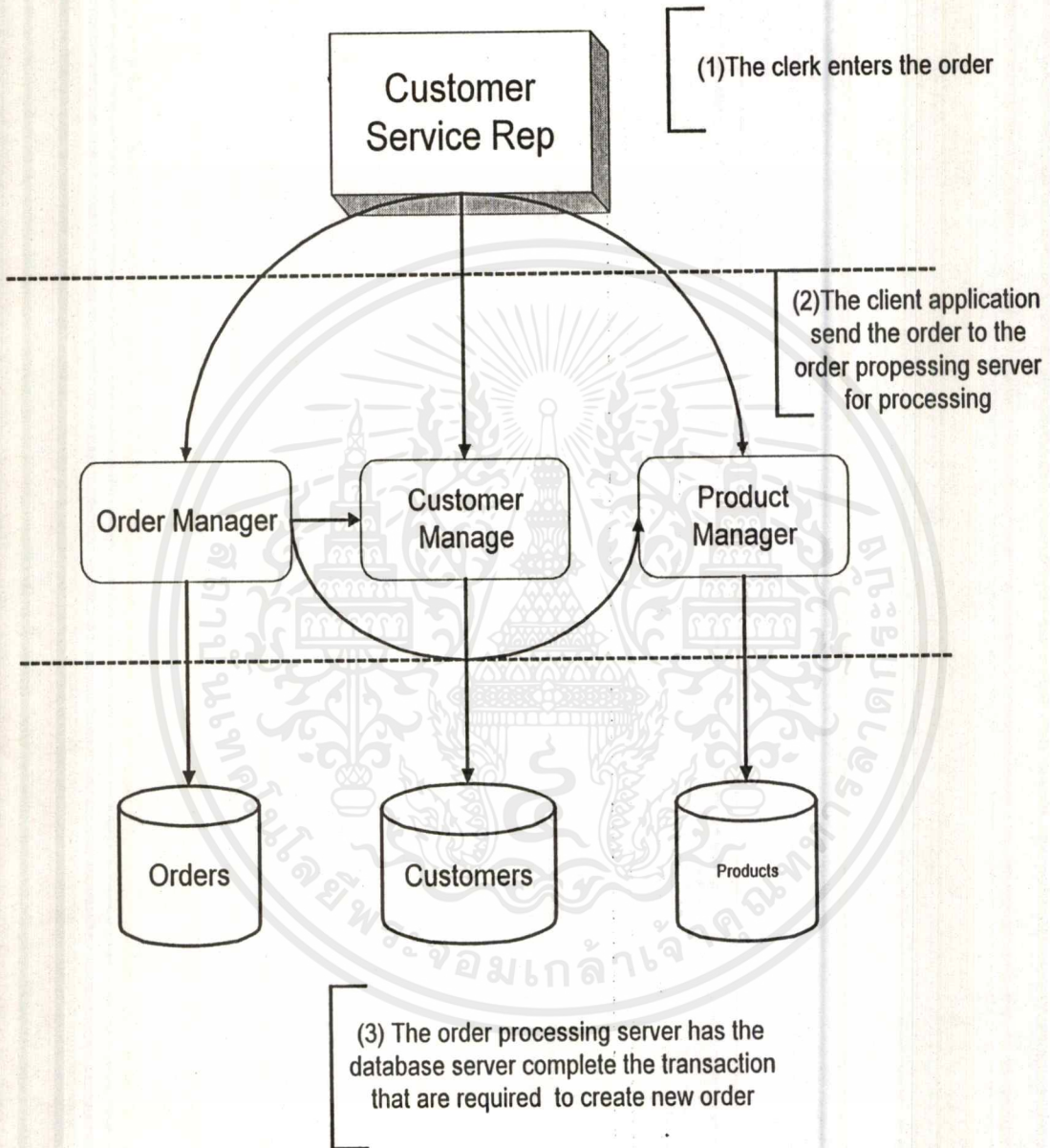
### 2.7.4 ผู้ใช้สามารถสร้างแอปพลิเคชันของตนเองได้

เนื่องจากกฎเกณฑ์ข้อบังคับของระบบถูกนำออกจากส่วนฟรอนต์เอนด์และนำไปใส่ในส่วนบริการกฎเกณฑ์ข้อบังคับของระบบ ดังนั้นส่วนของไคลเอนด์จึงไม่ต้องมีกฎเกณฑ์ข้อบังคับของระบบที่ซับซ้อนอีกต่อไป ส่วนไคลเอนด์จึงมีเพียงแต่ส่วนที่เก็บแอปพลิเคชันของระบบเท่านั้น ทำให้ผู้ใช้สามารถสร้างแอปพลิเคชันของตนเองได้ถ้าเพียงรู้ขอบเขตที่มีให้ในส่วนบิสซิเนส เซอร์วิสเท่านั้น



รูปที่ 2.7 แสดงตัวอย่างการใช้งานจริงของทรีเทียร์ในองค์กร

ตัวอย่างการใช้งานสถาปัตยกรรมแบบทรีเทียร์



รูปที่ 2.8 แสดงการไหลของข้อมูลในระบบ

จากรูป มีขั้นตอนการทำงานดังนี้

1. เสิร์ชรับรายการจากลูกค้าและป้อนข้อมูลผ่านส่วนแอปพลิเคชันที่บริการลูกค้า (Customer Service Application) โดยหน้าจออาจจะเป็นรูปแบบที่ประกอบด้วย ชื่อลูกค้า, สินค้า, รายละเอียดสินค้า หรืออื่นๆ

2. เมื่อป้อนข้อมูลเข้ามาแล้วส่วนไคลเอนต์จะส่งรายละเอียดการสั่งซื้อสินค้าไปยังส่วนที่รับผิดชอบการสั่งซื้อสินค้า ( Order-processing business server ) โดยส่วนนี้จะตรวจสอบรายละเอียดการสั่งซื้อสินค้าโดยการติดต่อกับลูกค้าและส่วนที่รับผิดชอบกฎเกณฑ์ข้อบังคับของระบบ (Business server) ถ้าข้อมูลที่ได้รับถูกต้อง ส่วนที่จัดการการสั่งซื้อสินค้า (Order server) จะทำการนำข้อมูลใหม่ที่ได้ไปเก็บที่ส่วนเก็บข้อมูล (Database server)

3. ส่วนเก็บข้อมูล (Data server manager) จะจัดการกับรายการที่ส่วนจัดการการสั่งซื้อสินค้า (Order manager business server) ส่งมาโดยนำข้อมูลที่ได้รับมาไปเก็บ

## 2.8 การใช้วิซวลเบสิกในการพัฒนาสถาปัตยกรรมแบบทรีเทียร์

การใช้วิซวลเบสิกเป็นเครื่องมือสำหรับพัฒนาโปรแกรมบนวินโดวส์นั้น วิซวลเบสิก 4.0 รุ่นเอนเทอร์ไพรส์ ได้เพิ่มส่วนที่จัดการฐานข้อมูลเข้ามาด้วย โดยฐานข้อมูลที่ใช้กับวิซวลเบสิกนี้เป็นฐานข้อมูลภายนอกประเภท โอดีบีซี ก็จะเป็นฐานข้อมูลที่อยู่บนเครื่องอื่น โปรแกรมของวิซวลเบสิกจะติดต่อกับฐานข้อมูลชนิดนี้โดยผ่านไดรฟ์เวอร์ (Driver) โอดีบีซี (ODBC) ที่ได้ติดตั้งไว้แล้ว สำหรับการติดตั้ง โอดีบีซี อ่านได้ในบทที่ 3

วิซวลเบสิก เอนเทอร์ไพรส์ อีดิชั่น ถูกออกแบบให้มีความยืดหยุ่นในการใช้งาน โดยการเพิ่มสิ่งเหล่านี้เข้าไป

- การสร้างส่วนสนับสนุนสำหรับส่วนบิสซิเนสลอจิก(business logic)โดยการใช้สถาปัตยกรรมแบบทรีเทียร์(Three-tierd architecture) แทนสถาปัตยกรรมแบบทูเทียร์(Two-tierd)

- มีความยืดหยุ่นในการรวมข้อมูล (encapsulated) เข้าไว้ด้วยกันเพื่อให้สามารถใช้ส่วนประกอบต่างๆ ได้อีก (reusable component)

- การเข้าถึงฐานข้อมูลได้อย่างมีประสิทธิภาพ ด้วยวิธีที่เหมาะสมที่สุดในการเข้าถึงข้อมูลที่ต้องการ

- ไม่จำเป็นต้องเรียนรู้และใช้คำสั่งที่แตกต่างกันใหม่หมด เพื่อจัดการกับฐานข้อมูลที่ไม่เหมือนกัน ช่วยให้โปรแกรมที่สร้างขึ้นมาสำหรับทำงานกับฐานข้อมูลชนิดหนึ่งสามารถปรับเปลี่ยนไปใช้ฐานข้อมูลอีกชนิดหนึ่งได้โดยง่าย โดยไม่ต้องแก้ไขโปรแกรม เพราะวิซวลเบสิกจะแยกส่วนของข้อมูลกับการทำงานของโปรแกรมออกเป็นอิสระจากกันได้

โดยการใช้วิซวลเบสิกสำหรับการพัฒนาสถาปัตยกรรมแบบทรีเทียร์นี้จะถูกนำมาใช้ใน 2 ส่วนด้วยกันคือส่วนบริการผู้ใช้ และส่วนบิสซิเนส เซอร์วิส

## 2.8.1 การใช้ Visual Basic พัฒนาแอปพลิเคชันบนวินโดวส์สำหรับส่วนการให้บริการผู้ใช้

จากความต้องการในปัจจุบันที่มีการนำเอาระบบคอมพิวเตอร์มาใช้ทำงานต่างๆ มากขึ้น ซึ่งงานในแต่ละระบบย่อมประกอบไปด้วย ผู้ใช้ (User) ที่แตกต่างกันออกไปทั้งในแง่ของ ประสบการณ์ในการใช้คอมพิวเตอร์และความชำนาญ สาเหตุนี้เองทำให้ผู้พัฒนาระบบพยายามที่จะออกแบบระบบให้สามารถติดต่อกับผู้ใช้ได้ที่แตกต่างกันนั้นได้ โดยมองที่ความสามารถ ทางด้านการติดต่อกับผู้ใช้ (User Interface) ที่ดี เข้าใจง่าย ใช้งานสะดวกเป็นสิ่งสำคัญ

โดยการติดต่อกับผู้ใช้ แบ่งเป็น 2 ประเภทใหญ่ๆ คือ

1. Command User Interface
2. Graphics User Interface

ซึ่งแต่ละประเภทมีความเหมาะสมในการนำไปใช้งานที่แตกต่างกันกล่าวคือ Command User Interface เหมาะกับระบบงานที่ผู้ใช้มีความชำนาญในการใช้คีย์บอร์ด และคอมพิวเตอร์เป็นอย่างดี

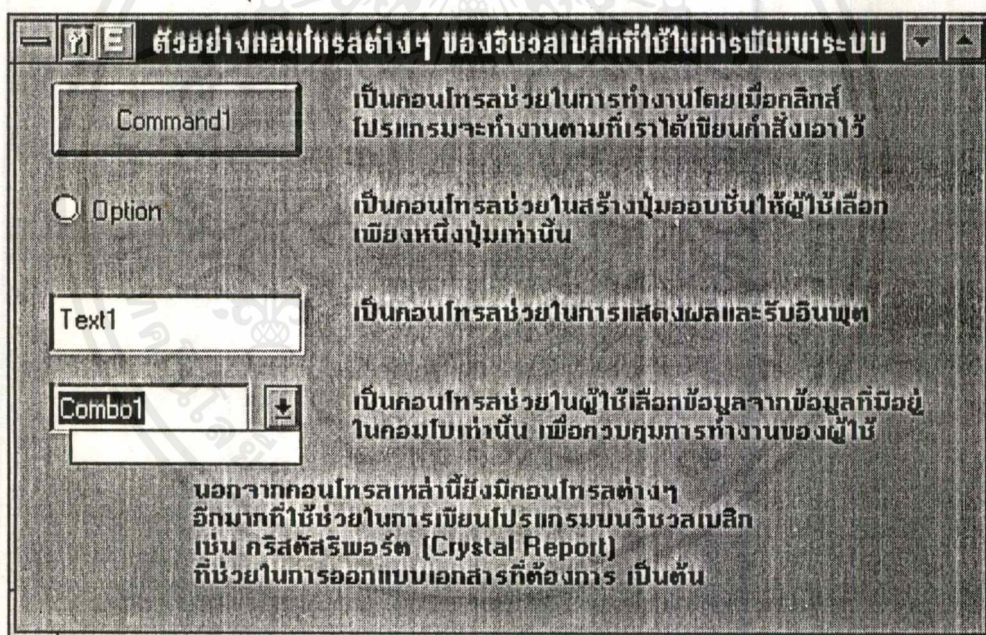
ส่วน Graphics User Interface นั้นเหมาะกับระบบงานที่ผู้ใช้ไม่มีความชำนาญในการใช้คอมพิวเตอร์นักหรือระบบงานที่ต้องการความสะดวกรวมทั้งความสวยงามของหน้าจอต่างๆ ด้วยเหตุนี้เองทำให้ผู้พัฒนาระบบส่วนใหญ่ในปัจจุบันออกมาทำ User Interface แบบกราฟิก (Graphics) มากขึ้น ทำให้เกิดการพัฒนาซอฟต์แวร์ที่ช่วยในการเขียน โปรแกรมด้านนี้ มากขึ้น

จากการพัฒนาระบบปฏิบัติการจากดอส (DOS) ไปสู่วินโดวส์ (Windows) การพัฒนาแอปพลิเคชันต่างๆ ซึ่งเดิมที่เคยใช้เครื่องมือในการพัฒนาเพียงแค่ แอสเซมบลี, ซี-คอมไพเลอร์, ปาสคาล-คอมไพเลอร์ ฯลฯ กับโปรแกรมเอดิเตอร์ ก็เพียงพอแล้วสำหรับแอปพลิเคชันบนดอสซึ่งมีสถานะแวดล้อมที่ไม่ซับซ้อนอะไรมากนัก แต่สำหรับในวินโดวส์ซึ่งมีสถานะแวดล้อมแบบกราฟิก, เป็นระบบมัลติทาสกิ้ง (Multitasking) ฯลฯ เครื่องมือ (Tool) ในการพัฒนาที่เคยใช้กันอยู่ คงจะไม่เหมาะสมที่จะนำมาใช้ในการพัฒนาแอปพลิเคชันบนวินโดวส์ (Windows Application) เพราะจะทำให้ใช้ทรัพยากรทั้งกำลังคนและเวลาเป็นอย่างมาก จากจุดนี้เองจึงเป็นจุดเริ่มต้นของ แนวความคิดในการสร้างเครื่องมือต่างๆ เพื่อนำมาช่วยสำหรับการพัฒนาแอปพลิเคชันบนวินโดวส์ ให้สะดวกและง่ายยิ่งขึ้น

ระบบช่วยเขียนโปรแกรม Microsoft Visual for Windows ถือได้ว่าเป็นความก้าวหน้าที่สำคัญที่เกี่ยวข้องกับการเขียนแอปพลิเคชันที่ทำงานบนวินโดวส์ ใน Visual Basic ประกอบไปด้วยเอนจิน (Engine) ที่ทำงานด้วย อีเวนต์และเครื่องมือช่วยออกแบบที่ใช้งานได้ง่าย ซึ่งด้วยลักษณะเหล่านี้ทำให้สามารถได้รับประโยชน์จากสภาพแวดล้อมกราฟิกของวินโดวส์ได้อย่างเต็มที่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้สามารถสร้างแอปพลิเคชันที่มีสมรรถนะสูงได้อย่างรวดเร็ว โดยเฉพาะอย่างยิ่งสำหรับงานทางด้าน Graphic User Interface

Visual Basic ยังเป็นหนึ่งในภาษาคอมพิวเตอร์ภาษาแรกที่สนับสนุน Event-Driven programming ซึ่งเป็นรูปแบบการเขียนโปรแกรมที่เหมาะสมกับ Graphic User Interface เป็นอย่างยิ่ง ใน Event-driven programming แทนที่จะเขียนโปรแกรมที่แสดงขั้นตอนทุกๆ ขั้นตอนออกมาเรียงตามลำดับ ผู้เขียนโปรแกรมกลับเขียนโปรแกรมที่ตอบสนองต่อกิจกรรมที่ผู้ใช้ทำเช่น การเลือกคำสั่ง การคลิกหรือกดปุ่มเมาส์ในวินโดว์ การเคลื่อนย้ายเมาส์ แทนที่จะเขียนโปรแกรมขนาดใหญ่เพียงโปรแกรมเดียว ผู้เขียนโปรแกรมจะสามารถสร้างโปรแกรมเป็นกลุ่มของโปรแกรมย่อยๆ ซึ่งจะเกิดการดำเนินงานโดยอีเวนต์ที่ผู้ใช้เริ่มขึ้นและด้วย Visual Basic ทำให้สามารถเขียนแอปพลิเคชันเช่นนั้นได้ด้วยความรวดเร็วและง่ายขึ้น อาทิเช่น



รูปที่ 2.9 แสดงคอนโทรลพื้นฐานของวิซวลเบสิก

## 2.8.2 การใช้ Visual Basic พัฒนาแอปพลิเคชันบนวินโดวส์สำหรับส่วนการให้บริการข้อมูล

วิซวลเบสิกได้นำเอาหลักการของ ออบเจ็กต์มาประยุกต์ใช้ รวมทั้งในเรื่องที่เกี่ยวกับฐานข้อมูลด้วย ดังนั้นทุกองค์ประกอบของฐานข้อมูลในวิซวลเบสิก ไม่ว่าจะเป็นตัวฐานข้อมูล (Database) ,ตาราง(Table),ฟิลด์(Field) หรือคอลัมน์(Column) และอินเด็กซ์(Index) จะมีลักษณะเป็นออบเจ็กต์หมด ทุกออบเจ็กต์ที่เกี่ยวกับฐานข้อมูลในวิซวลเบสิก จะประกอบไปด้วยส่วนสำคัญ 2 ส่วน คือ "พรอพเพอร์ตี้" (Property) และ "เมธอด" (Method: ฟังก์ชันการทำงานกับออบเจ็กต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งที่ใช้ในการจัดการกับโครงสร้างฐานข้อมูลก็คือ การเรียกใช้ฟังก์ชันการทำงาน (Method) ที่เกี่ยวข้องกับออบเจกต์ของฐานข้อมูลที่ต้องการนั่นเอง ซึ่งในระบบทรีทีเยอร์จะเลือกใช้ rdoObject เป็นตัวจัดการออบเจกต์ของฐานข้อมูล

### 2.8.3 ประโยชน์ของการใช้วิซวลเบสิกในการพัฒนาโปรแกรมด้านฐานข้อมูล

ลดข้อจำกัดของการพัฒนางานโดยใช้แฟ้มข้อมูลธรรมดาในการจัดการกับข้อมูลดังนี้

#### 2.8.3.1) โปรแกรมเป็นอิสระจากฐานข้อมูลที่ใช้

- เพราะว่าเมื่อ ไม่มีความแตกต่างของฐานข้อมูล หมายความว่าไม่ว่าจะเป็นฐานข้อมูลชนิดใด ก็จะใช้ชุดคำสั่งในการอ่าน เขียน จัดการกับฐานข้อมูลต่างๆ ได้เหมือนกัน จะแตกต่างกันก็เพียงแต่ส่วนขยายของคำสั่งสำหรับการเปิด หรือสร้างฐานข้อมูลเท่านั้น

- ทำให้ไม่ต้องเรียนรู้คำสั่งที่แตกต่างกันหลายๆแบบ

- การใช้ชุดคำสั่งย่อย (module) ร่วมกันก็เป็นไปได้มาก

- การสร้างระบบงานในลักษณะ โคล์เอ็นด์/เซิร์ฟเวอร์ ก็ง่ายขึ้น สามารถทำการพัฒนาระบบบนเครื่องพีซี เครื่องเดียวก็ได้ แล้วค่อยเปลี่ยนค่าที่กำหนดในเรื่องฐานข้อมูลเท่านั้น ก็สามารถนำไปใช้กับข้อมูลบนเครื่องอื่นได้ทันที

#### 2.8.3.2) ข้อมูลเดิมที่มีอยู่แล้วในฐานข้อมูลชนิดอื่นๆสามารถนำมาใช้ใน วิซวลเบสิกได้ โดยไม่จำเป็นต้องสร้างใหม่ให้ซ้ำซ้อนกัน

2.8.3.3) แยกส่วนติดต่อกับผู้ใช้ออกจากคำสั่ง

## 2.9 โอลี (OLE: Object Linking and Embedding)

โอลี เป็นเหมือนภาษาที่ทำให้แอปพลิเคชันหนึ่งสามารถติดต่อกับแอปพลิเคชันอื่นๆ ได้เพื่อความสะดวกในการทำงานร่วมกัน การใช้ โอลี ทำให้แอปพลิเคชันของวิซวลเบสิก สามารถแสดงและควบคุมแอปพลิเคชันอื่นได้ ทำให้ผู้ใช้สามารถทำงานได้อย่างสะดวกรวดเร็วและมากขึ้น ผู้ใช้โอลี จะรู้สึกเหมือนกับการทำงานบนแอปพลิเคชันเพียงตัวเดียว แต่มีความสามารถของแอปพลิเคชันอื่นรวมอยู่ด้วย เช่น ใช้โปรแกรมของ ไมโครซอฟท์ เวิร์ด (Microsoft Word) อยู่แต่มีรูปจากโปรแกรม วิสิโอ(Visio) ภายใน เวิร์ด โดยการใช้งาน ผู้ใช้ไม่จำเป็นต้องเกี่ยวข้องกับการจัดการหรือเปลี่ยนไป เปลี่ยนมาระหว่างแอปพลิเคชัน งานต่างๆจะทำได้โดยใช้ความสามารถของโอลี ซึ่งจะสามารถแสดงข้อมูลจากแอปพลิเคชันอื่นๆ และแก้ไขข้อมูลเหล่านั้นจากภายในแอปพลิเคชันได้

ไมโครซอฟท์ สนับสนุนการใช้งาน โอลี บน วินโดวส์ แพลตฟอร์ม (Windows Platforms ) ซึ่งมี Windows 3.x, Windows 95, Windows NT โอลี ได้กลายเป็นส่วนหนึ่งของ Windows Open System Architecture (WOSA) ซึ่งเป็นการตกลงระหว่าง Windows Application

ต่างๆว่าควรจะทำงานให้เข้ากันได้อย่างไร โอล่ เป็นพื้นฐานมาจาก Component Object Model (COM) โดยจะกำหนดหน้าที่ที่สำคัญๆของ โอล่ ซึ่งเซอร์วิสอื่นสร้างขึ้นมา

โอล่ จะกำหนดมาตรฐานบนแอฟลิเคชัน คอมโพเนนต์ว่าจะติดต่อกันอย่างไร เพราะเมื่อแอฟลิเคชัน ถูกพัฒนาด้วยภาษาโปรแกรมมิ่ง ที่แตกต่างกันแล้วจำเป็นต้องทำงานด้วยกัน แอฟลิเคชัน เหล่านั้นจำเป็นต้องติดต่อกันด้วยภาษาซึ่งเรียกว่า "โอล่"

### 2.9.1 โอล่ ออบเจ็กต์

การพัฒนาโปรแกรมโดยใช้โอล่จะต้องมองสิ่งต่างๆเป็นออบเจ็กต์ ซึ่งอาจจะแตกต่างจากที่เคยพบในการเขียนโปรแกรมแบบออบเจ็กต์โอเรียนเตด (Object - Oriented Programming) โอล่ออบเจ็กต์สามารถเป็นได้ทั้งเอกสารของเวิร์ดโปรเซสซิ่ง (Word Processing) , เซลแต่ละเซลล์ในสเปรดชีต (SpreadSheet), ส่วนจัดการใบส่งสินค้า, ตัวจัดการฐานข้อมูล, รูปเคลื่อนไหว(Animation), โปรแกรมประยุกต์ทั้งโปรแกรมหรือแม้กระทั่งฮาร์ดแวร์ แพลตฟอรม์ (Hardware Platform) ล้วนเป็นออบเจ็กต์ทั้งสิ้น โดยจะถือว่าสิ่งเกี่ยวกับคอมพิวเตอร์เป็นออบเจ็กต์ทั้งสิ้น ยกตัวอย่างเช่น ไคลเอนต์/เซิร์ฟเวอร์ จะเป็นการโต้ตอบระหว่าง โอล่ ออบเจ็กต์ 2 ตัว ซึ่ง ไคลเอนต์เป็นออบเจ็กต์ที่ส่งการร้องขอ(Request) ส่วนเซิร์ฟเวอร์จะเป็นออบเจ็กต์ที่ตอบสนองต่อการร้องขอ

ในการติดต่อระหว่างโอล่ออบเจ็กต์ 2 ออบเจ็กต์ ออบเจ็กต์หนึ่งจะมีการร้องขอให้อีกออบเจ็กต์หนึ่งทำงานให้ ออบเจ็กต์ดังกล่าวจะถูกเรียกว่า โอล่ไคลเอนต์ (OLE Client) และอีกออบเจ็กต์หนึ่งจะถูกเรียกว่า โอล่เซิร์ฟเวอร์ (OLE Server)

โอล่เซิร์ฟเวอร์ตัวหนึ่งๆ สามารถเป็นได้ทั้ง อินโปรเซสเซิร์ฟเวอร์หรือเอาท์ออฟโปรเซสเซิร์ฟเวอร์ ซึ่งอินโปรเซสเซิร์ฟเวอร์จะถูกจัดเก็บอยู่ในรูปแบบไฟล์ DLL และเอาท์ออฟโปรเซสเซิร์ฟเวอร์จะจัดเก็บอยู่ในรูปแบบไฟล์ที่สามารถเอ็กซิกิวทีฟได้

อินโปรเซสเซิร์ฟเวอร์จะวิ่งอยู่ในแอดเดรสสเปซ (Address Space) เดียวกันกับโอล่ไคลเอนต์ ในขณะที่เอาท์ออฟโปรเซสเซิร์ฟเวอร์จะวิ่งอยู่ในแอดเดรสสเปซของมันเอง ดังนั้นการร้องขอแบบ อินโปรเซส จะคล้ายกับฟังก์ชันคอลของ DLL และการร้องขอแบบเอาท์ออฟโปรเซส จะคล้ายกับฟังก์ชันคอลของ DDE (Dynamic Data Exchange)

ชนิดของเซิร์ฟเวอร์	โอล่ไคลเอนต์	โอล่เซิร์ฟเวอร์
อินโปรเซส	EXE หรือ DLL ไฟล์	DLL ไฟล์
เอาท์ออฟโปรเซส	EXE หรือ DLL ไฟล์	EXE ไฟล์

ตาราง 2.9 แสดงอินโปรเซส (In - Process) และเอาท์ออฟโปรเซส (Out of - Process) เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9.2 โอล่ เซอร์วิส ที่น่าสนใจ

**แดร็ก แอนด์ ดรอป (Drag and Drop) :** เป็นการเคลื่อนย้ายข้อมูลระหว่างแอปพลิเคชัน คล้ายๆกับการ คัท(cut) และการ เพซท์ (Paste) แต่จะทำงานได้เพียง 1 ชุดเป็ โดยการให้ผู้ใช้เลือกอบเจ็กต์ ในแอปพลิเคชันตัวหนึ่ง โดยการกดปุ่มเมาส์ ข้างซ้ายทิ้งไว้ที่อบเจ็กต์นั้น แล้วทำการลากเมาส์(Drag) ไปยังแอปพลิเคชันหนึ่ง และวาง อบเจ็กต์ในตำแหน่งที่ต้องการ (Drop)

**โอล่ ด็อคคิวเมนต์ (OLE Document) :** อาจกล่าวได้ว่าเป็นคอมพิวเตอร์คือคิวเมนต์ ซึ่งประกอบไปด้วย ข้อมูลจากหลายๆ แอปพลิเคชัน เช่น อาจจะเป็น เวิร์ด ที่ประกอบไปด้วย รูปภาพ ที่สร้างจาก วิสิโอ(Visio)

**การเชื่อมโยง (Linking) :** โอล่ สามารถนำอบเจ็กต์ไปเชื่อมโยงกับแอปพลิเคชันหนึ่งๆ การเชื่อมโยงจะเป็นการแทรกตัวถือตำแหน่ง(เก็บตัวชี้ตำแหน่ง) ไม่ใช่ข้อมูลจริงๆ ส่วนข้อมูลจริง จะถูกเก็บอยู่นอก คอมพิวเตอร์ ด็อคคิวเมนต์ ยกตัวอย่างเช่น เมื่อเชื่อมโยงกราฟิก ไปไว้ในเวิร์ด ข้อมูลที่ถูกเชื่อมโยงจะถูกเก็บไว้ในไฟล์อีกไฟล์หนึ่ง ส่วน โอล่ จะเก็บเฉพาะเส้นทางที่จะไปยังข้อมูลจริงๆ แอปพลิเคชันที่ถูกเชื่อมโยงจะใช้งานได้โดยอัตโนมัติ โดยผู้ใช้สามารถแก้ไขกราฟิกได้โดยการใช้แอปพลิเคชันที่เกี่ยวข้องกัน และเมื่อมีการเปลี่ยนแปลงกราฟิกในส่วนซอร์สแอปพลิเคชัน ก็จะมีการเปลี่ยนแปลงในแอปพลิเคชันทั้งหมดที่มีการเชื่อมโยง หรือถ้าซอร์ส แอปพลิเคชัน เกิดการย้ายไปที่อื่น การเชื่อมโยงก็จะหายไปด้วย

**การฝัง (Embedding) :** โอล่สามารถฝังอบเจ็กต์ไว้ในแอปพลิเคชันได้ โดยการฝังอบเจ็กต์นี้จะเป็นการแทรกข้อมูลที่แท้จริงสำหรับอบเจ็กต์ที่ถูกฝังเอาไว้ ข้อมูลจะถูกเก็บอยู่ในคอมพิวเตอร์ไฟล์ ต่างจากการเชื่อมโยงที่เก็บเฉพาะตัวชี้ว่า ข้อมูลอยู่ที่ไหน

## 2.9.3 โอล่ ออโตเมชัน เซิร์ฟเวอร์ (OLE Automation Server)

โอล่ ออโตเมชัน เซิร์ฟเวอร์ เป็นกลไกที่ช่วยให้สามารถเรียกใช้แอปพลิเคชันที่ต้องการเพื่อการทำงานที่จะเรียกใช้ และควบคุม โอล่ อบเจ็กต์ ภายในแอปพลิเคชันนั้นๆอีกทอดหนึ่ง โดยสามารถใช้คุณสมบัติและฟังก์ชันการทำงานของ โอล่ อบเจ็กต์ ที่มีอยู่ในแอปพลิเคชันนั้นๆได้

เป็นส่วนที่ได้รับการพัฒนาอย่างมากในวิซวลเบสิก 4.0 ซึ่งแอปพลิเคชันที่สร้างขึ้นสามารถทำงานเป็นเซิร์ฟเวอร์บริการแอปพลิเคชันที่ทำงานในระบบเดียวกัน ระบบอื่น หรือบนเน็ตเวิร์กได้ โดยลักษณะการเขียน โปรแกรมจะใช้ลักษณะของคลาส โมดูลสำหรับกำหนดคปรอพเพอดี และเมธอดซึ่งแอปพลิเคชันอื่นเรียกใช้ได้โดยผ่านโอล่ ( การใช้ OLE ทำให้แอปพลิเคชันเป็นโอล่ เซิร์ฟเวอร์และแอปพลิเคชันอื่นสามารถเรียกใช้บริการได้ )

จากการที่วิซวลเบสิกเพิ่มลักษณะการใช้โอเล่ให้มีประสิทธิภาพมากขึ้น แทนการเขียนโปรแกรมทั้งหมดไว้ในไฟล์เดียว ทำให้สามารถแยกโปรแกรมออกเป็น โอเล่ เซิร์ฟเวอร์หลายๆ ส่วนได้ ซึ่งมีประโยชน์ดังนี้คือ

- โอเล่ เซิร์ฟเวอร์ จะถูกโหลดเมื่อต้องการใช้ และอัน โหลดเมื่อไม่ต้องการใช้
- โอเล่ เซิร์ฟเวอร์ แบบ เอาท์-ออฟ โพรเซส (Out-of-process) ทำงานแบบ 32 บิต บนวินโดวส์ 95 หรือเอ็นที และสามารถเรียกใช้จากแอฟลิเคชันแบบ 16 บิต ได้
- โอเล่ เซิร์ฟเวอร์ สามารถแยกกันคิบั๊กเมื่อมีข้อผิดพลาด และนำกลับมาใช้ซ้ำได้ ช่วยให้ประหยัดเวลาไม่ต้องเขียน โปรแกรมส่วนนั้นขึ้นมาใหม่อีก
- ประโยชน์ที่เห็นได้ชัดอีกข้อหนึ่งคือ ประสิทธิภาพในการทำงานของแอฟลิเคชันจะเพิ่ม เช่น ในระหว่างที่ใช้ VBTERM คำนวณโหลดไฟล์จากการติดต่อระยะไกล ซึ่งปกติมักทำงานได้ค่อนข้างช้า ดังนั้นในช่วงนั้นแอฟลิเคชันอื่นก็สามารถเรียกใช้ VBTERM ต่อได้ทันที จึงเท่ากับเป็นการใช้ประโยชน์จากมัลติทาสกิงของวินโดวส์ 95 ได้อย่างเต็มที่

โอเล่ ออโตเมชัน เซิร์ฟเวอร์ ถูกสร้างโดยใช้โครงสร้างใหม่ที่ถูกแนะนำในวิซวลเบสิก 4.0 คือ คลาสโมดูล (Class Modules) เหมือนกับมาตรฐาน .BAS modules เป็น โค้ดโมดูลที่ไม่มี GUI แต่มี ฟังก์ชัน, โพรซีเจอร์ และ แวริเอเบิล (variable) โดยเรากำหนด คลาสโมดูลเป็น ออบเจกต์คลาส ทุกๆแวริเอเบิล และฟังก์ชันในคลาสโมดูล สามารถกำหนดความเป็น "พับลิค" (Public: ใช้ได้ทั่วไป) หรือ "ไพรเวท" (Private: ใช้ได้เฉพาะใน คลาส โมดูลนั้น)

แอฟลิเคชัน สามารถถูก คอมไพล์ เป็น โอเล่ ออโตเมชัน เซิร์ฟเวอร์ ด้วยการเลือก ออบชัน (Option) จาก ทูล (Tool) เมนู และเลือก โอเล่ เซิร์ฟเวอร์ ใน โปรเจกต์ แทป (Project Tab) เมื่อ แอฟลิเคชันถูกคอมไพล์เป็น โอเล่ เซิร์ฟเวอร์ มันสามารถจะถูกเรียกได้ทันที โดยแอฟลิเคชันอื่นๆผ่านทาง โอเล่ ออโตเมชัน โดยทุกๆฟังก์ชันที่ ดีแคลร์(declare) เป็นพับลิคจะถูกมองเป็นเมธอด และทุกๆแวริเอเบิลและพรอพเพอร์ตี้ที่ ดีแคลร์เป็นพับลิคจะถูกมองเป็นพรอพเพอร์ตี้

โอเล่ เซิร์ฟเวอร์ แบบ เอาท์-ออฟโพรเซส จะเป็น โปรแกรมที่ทำงานได้เองตัวหนึ่ง ที่ให้บริการ โอเล่กับโปรแกรมอื่นๆ งานที่ทำของ โปรแกรมจะใช้เนื้อที่สแต็กของตัวเอง เมื่อใดก็ตามที่ไคลเอ็นต์ใช้โอเล่ ออบเจกต์โพรเซสข้ามจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์ โดยมีจุดเด่นคือทำงานแบบอะซิงโครนัส การแสดงฟอร์มเป็นแบบนอนโมเดล ป้องกันข้อผิดพลาดที่จะเกิดในเซิร์ฟเวอร์ ซึ่งอาจทำให้โปรแกรมหยุดทำงานได้ ติดต่อกับโปรแกรมได้ทุกชนิดทั้ง 16 บิตและ 32 บิต แต่มีข้อเสียคือ ความเร็วตอนเริ่มต้นจะต้องโหลด คีแอลแอล (DLL) หลายๆตัว และระหว่างการทำงานจะต้องใช้ซีพียูไหลหลายรอบ ทำให้การใช้งานแบบนี้ช้ากว่า แบบอิน-โพรเซส

โอเล่ เซิร์ฟเวอร์ แบบอิน-โพรเซส จะเป็นคิแอลแอลที่ให้บริการ โอเล่กับโปรแกรมการทำงานจะใช้สแต็กและเนื้อที่ของไคล์เอ็นด์ตลอดจึงไม่ต้องเสียเวลาข้ามไปข้ามมาระหว่างเซิร์ฟเวอร์กับไคล์เอ็นด์ การทำงานจึงเร็วกว่า แบบเอาท์-ออฟโพรเซส แต่สำหรับ อิน-โพรเซสทำงานเป็นแบบ 32 บิตเท่านั้นและไม่สามารถใช้โมเดลฟอร์มได้

แอตทริบิวต์(attribute) ของ โอเล่ ออโตเมชัน อินเตอร์เฟส มี พรอพเพอร์ตี้, เมธอด, อีเวนต์(Event),เมมเบอร์ ออบเจกต์(Member object) และคอลเลกชัน(Collection)

แอตทริบิวต์	คำอธิบาย
พรอพเพอร์ตี้	คุณสมบัติของออบเจกต์
เมธอด	กระบวนการที่ออบเจกต์สามารถทำได้
อีเวนต์	การกระทำต่อออบเจกต์
เมมเบอร์ออบเจกต์	ออบเจกต์ย่อยที่ถูกสะสมในออบเจกต์ เช่น ตัวผลิตภัณฑ์ (item) ในสายผลิตภัณฑ์(LinItem)
คอลเลกชัน	ชนิดของออบเจกต์ที่เก็บรวบรวมกลุ่มของออบเจกต์ที่มีความสัมพันธ์กันไว้ เช่น ตารางที่เก็บแถวของข้อมูลไว้

ตาราง 2.10 แสดงรายละเอียดของโอเล่ออโตเมชันอินเตอร์เฟส

## 2.9.4 การควบคุม โอเล่ ออโตเมชัน ออบเจกต์

2.9.4.1 การสร้าง โอเล่ ออโตเมชัน ออบเจกต์โดยเริ่มจากการประกาศตัวแปรและอ้างอิงถึง โอเล่ ออโตเมชัน ออบเจกต์นั้น

ในการประกาศตัวแปรเชิงออบเจกต์ทำได้โดยการใช้คำสั่ง

`Dim OleObj As Object` หรือ

`Dim OleObj As class`

และทำการอ้างอิงถึง โอเล่ ออบเจกต์ ที่ประกาศตัวแปรแล้ว ทำได้โดยใช้คำสั่ง

`Set OleObj = New class` หรือ

`Set OleObj = CreateObject ("class")` หรือ

`Set OleObj = GetObject ("", "class")` หรือ

`Set OleObj = GetObject ([pathname], [classserver])`

โดย	<i>OleObj</i>	คือ ชื่อของตัวแปร
	<i>class</i>	คือ ชื่อของคลาสนั้น(เปลี่ยนแปลงได้ที่พรอพเพอร์ตี้ของคลาส)
	<i>pathname</i>	คือ ตำแหน่งของคลาสเซิร์ฟเวอร์ไฟล์
	<i>classserver</i>	คือ ชื่อของคลาสเซิร์ฟเวอร์

#### 2.9.4.2 การเข้าถึง โอล่ ออโตเมชัน ออบเจกต์ (Accessing an OLE Automation Object)

- โดยใช้พรอพเพอร์ตี้ของออบเจกต์ มีรูปแบบของคำสั่งดังนี้

`object.property = value`

โดยที่	<i>object</i>	คือ	ชื่อออบเจกต์
	<i>property</i>	คือ	พรอพเพอร์ตี้ต่าง ๆ ของออบเจกต์
	<i>value</i>	คือ	ค่าที่เป็นไปได้ของพรอพเพอร์ตี้

- โดยใช้เมธอดของออบเจกต์ มีรูปแบบของคำสั่งดังนี้

`variable = object.method ( arg1, arg2, .... )`

โดยที่	<i>variable</i>	คือ	ชื่อตัวแปร
	<i>method</i>	คือ	กระบวนการที่ทำได้
	<i>arg</i>	คือ	ค่าต่างๆ ที่ต้องการใช้ในเมธอด

- โดยใช้ เมมเบอร์ ออบเจกต์ และคอลเล็กชันของออบเจกต์ มีรูปแบบของคำสั่งดังนี้

`object.container(member).Mproperty = value` หรือ

`variable = object.container(member).Mmethod`

โดยที่	<i>container</i>	คือ	ชื่อคอลเล็กชัน
	<i>member</i>	คือ	ตำแหน่งของ เมมเบอร์ ออบเจกต์
	<i>Mproperty</i>	คือ	พรอพเพอร์ตี้ของ เมมเบอร์ ออบเจกต์
	<i>Mmethod</i>	คือ	กระบวนการของ เมมเบอร์ ออบเจกต์

#### 2.9.4.3 การทำลาย โอล่ ออโตเมชัน ออบเจกต์ ( Freeing Resources )

เมื่อไม่ต้องการใช้ โอล่ ออโตเมชัน ออบเจกต์ต่อไป ให้ทำการทำลายออบเจกต์นั้น เพื่อคืนทรัพยากรที่ออบเจกต์นั้นใช้ให้ว่าง ทำให้เกิดประสิทธิภาพในการทำงาน โดยใช้คำสั่ง

`Set objvar = Nothing`

โดยที่ *objvar* คือ ชื่อตัวแปรเชิงออบเจกต์

โอเล่ ออโตเมชัน เซิร์ฟเวอร์ ที่รันบนเครื่องเดียวกันเรียก โลกอล (Local) ถ้าต่างเครื่องกันเรียก รีโมท (Remote) โดย รีโมท ออโตเมชัน สามารถรันได้ทั้ง โลกอล และรีโมท

### 2.9.5 รีโมท ออโตเมชัน (Remote Automation หรือเรียกอีกอย่างว่า Remote OLE Server)

ในวิซวลเบสิก 4.0 เอนเตอร์ไพรส์อิดิชั่น จะทำการติดต่อผ่านระบบเครือข่ายจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่งได้ด้วยการติดต่อผ่านทางโอเล่ ออโตเมชัน (OLE Automation) ที่เหมือนกัน โดยที่โอเล่ ออโตเมชันจะตอบสนองการร้องขอจากเครื่องไคลเอนต์ผ่านทางรีโมท ออโตเมชัน ซึ่งเพิ่มเข้ามาในวิซวลเบสิก 4.0 เอนเตอร์ไพรส์ อิดิชั่น โดยรีโมท ออโตเมชัน เซิร์ฟเวอร์ หรือ รีโมท โอเล่ เซิร์ฟเวอร์ สามารถเรียกใช้รีซอร์ส (resource) ของเครื่องคอมพิวเตอร์เครื่องอื่นบนเน็ตเวิร์กได้

เหตุผลที่ใช้รีโมทออโตเมชันของวิซวลเบสิกและโอเล่ ในการพัฒนาส่วนของแอปพลิเคชันสำหรับระบบไคลเอนต์-เซิร์ฟเวอร์ในระดับเอนเตอร์ไพรส์ คือ

- รีโมท ออโตเมชันให้การติดต่อทางโปรแกรมมิ่งที่เหมือนกัน ทำให้ง่ายในการเข้าถึงจากโอเล่ของไคลเอนต์

- แนวความคิดที่ยุ่งยากซับซ้อนสามารถถูกจำกัด อธิบายและบำรุงรักษาได้โดยตรงจากภาษาของวิซวลเบสิกมากกว่าเอสคิวแอล วิซวลเบสิกให้ความสามารถในการทดสอบ แก้ไขและควบคุมซอร์สโค้ดที่มากกว่า

- ช่วยลดภาระของเซิร์ฟเวอร์ ซึ่งโอเล่ ออโตเมชันสามารถคำนวณงานที่ใหญ่จากเครื่องเคสทอปไม่ต้องไปโหลดเซิร์ฟเวอร์

- เพราะเป็นการติดต่อระหว่างคอมโพเนนท์ที่เหมือนกันทั้งการเอ็กซีคิวต์ในโลคอลและรีโมท คอมโพเนนท์จึงสามารถเคลื่อนย้ายระหว่างเครื่องโดยไม่ต้องทำการคอมไพล์คอมโพเนนท์หรือไคลเอนต์ของมันใหม่อีกครั้ง

รีโมท ออโตเมชันไม่ทำงาน กับอิน-โพรเซส เซิร์ฟเวอร์ (in-process server) โดยทั่วไปแล้ว โอเล่ ออโตเมชัน บิสซิเนส เซิร์ฟเวอร์ ที่จะต้องถูกเข้าถึงโดย ยูสเซอร์ เซอร์วิส ต้องสร้างเป็น เอ้าท์-ออฟโพรเซส แต่การเข้าถึงแบบอิน-โพรเซส จะเร็วกว่าการเข้าถึงแบบ เอ้าท์-ออฟโพรเซส ดังนั้นโดยทั่วไปแล้ว ยูสเซอร์ เซอร์วิส (เช่น Common logon dialog) เราจะเอาไว้บนเครื่องเดียวกันกับ ไคลเอนต์ และต้องถูกสร้างเป็น อิน-โพรเซส

### 2.9.6 โอลี คอนโทรล (OLE Control)

- โอลี คัสตอม คอนโทรล (OCX) เป็นรูปแบบที่พิเศษ ของ โอลี ออโตเมชัน ออบเจกต์
- OCX เป็น DLL ชนิดพิเศษที่ใช้ในการพัฒนาแอปพลิเคชัน ของภาษาวิซวลเบสิก เพื่อรองรับการทำงานจากระบบปฏิบัติการแบบ 32 บิต ซึ่งเพิ่มเข้ามาในวิซวลเบสิก เวอร์ชัน 4.0 โดยที่เวอร์ชันเดิมยังเป็น VBX ที่เป็นแบบ 16 บิต อยู่
- มีลักษณะคล้าย โอลี ออโตเมชัน คือมี พรอพเพอร์ตี้, เมธอด, เมมเบอร์ ออบเจกต์ และคอลเลกชัน
- ตัวอย่างของ OCX เช่น เท็กซ์บ็อกซ์ , ปุ่มคำสั่ง และ ลิสต์บ็อกซ์ ซึ่งจะถูกใช้ โดยทั่วไปในการสร้างกราฟิกอินเทอร์เฟซของไคลเอ็นต์ แอปพลิเคชัน (User Service)

## บทที่ 3

### การดำเนินการและพัฒนาระบบ

#### 3.1 การดำเนินงานในปัจจุบัน

##### 3.1.1 ลักษณะของการทำงาน

เดิมสำนักทะเบียนและประมวลผล ประสบกับปัญหาในการดำเนินงาน เนื่องจากไม่มีความสะดวกในการเข้าถึงข้อมูลบางระดับที่จัดเก็บอยู่ที่สำนักบริการคอมพิวเตอร์ เพื่อทำการแก้ไขการลงทะเบียนเรียนของนิสิตให้ทันเหตุการณ์ หรือการเรียกใช้ข้อมูลเพื่อตรวจสอบความถูกต้องก่อนออกหลักฐานทางการศึกษาต่างๆ นอกจากนี้ข้อมูลบางอย่างซึ่งเป็นที่ต้องการของหน่วยงานต่างๆ สำนักทะเบียนและประมวลผลไม่สามารถออกได้รวดเร็ว ทันความต้องการของผู้ใช้ในระดับต่างๆที่เกี่ยวข้อง

จุดมุ่งหมายของโครงการเดิม นอกจากแก้ปัญหาที่พบได้ในการดำเนินงานในปัจจุบัน และเพิ่มประสิทธิภาพของระบบย่อยๆอื่นแล้ว ระบบฐานข้อมูลที่เหมาะสมของสำนักทะเบียนและประมวลผล มีความจำเป็นต้องได้รับการพัฒนาขึ้นมาเป็นระบบสารสนเทศหรือระบบทะเบียนนิสิตเพื่อการจัดการการศึกษาให้สอดคล้องกับงานของฝ่ายต่างๆภายในสำนักงานที่ต้องเกี่ยวข้องกับงานทะเบียนประวัตินิสิต และการลงทะเบียนเรียนของนิสิตโดยมีวัตถุประสงค์ที่สำคัญคือ

1. เพื่อช่วยให้ผู้บริหาร และผู้ปฏิบัติงาน เข้าใจกระบวนการปฏิบัติงานได้ชัดเจนขึ้น และสามารถเพิ่มประสิทธิภาพให้กระบวนการปฏิบัติงานที่รับผิดชอบ ให้มีความกระชับ และคล่องตัวขึ้น

2. เพื่อลดขั้นตอนกระบวนการปฏิบัติงานประจำวัน

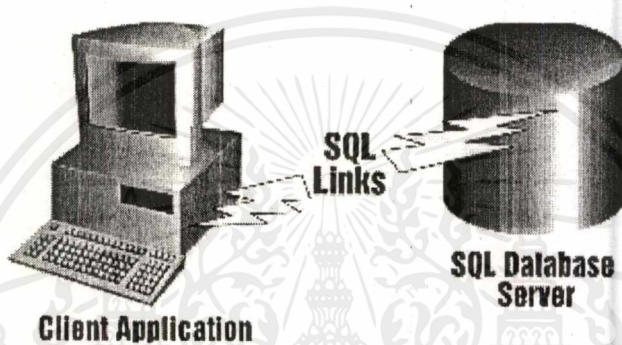
3. เพื่อจัดเตรียมองค์กรให้มีขีดความสามารถในการสร้างงาน และการรองรับงานทะเบียนประวัตินิสิต และการลงทะเบียนเรียนของนิสิตที่จะขยายตัวเพิ่มขึ้นในอนาคต

##### ขอบเขตของงานเดิม

โครงการเดิมเป็นการเขียน โปรแกรมระบบทะเบียนขึ้นมาแทนระบบเดิมที่มีอยู่ โดยที่ต้องมีความสามารถที่ดีกว่าเดิมในด้าน ความเร็วของการทำงานออกเอกสารต่างๆ ความสะดวกในการบริหารข้อมูล เนื่องจากปัจจุบันการทำงานของสำนักทะเบียนและประมวลผลประสบกับปัญหาในการดำเนินงาน เนื่องจากไม่มีความสะดวกในการเข้าถึงข้อมูลบางระดับที่จัดเก็บอยู่ที่สำนักบริการคอมพิวเตอร์ เพื่อทำการแก้ไขการลงทะเบียนเรียนของนิสิตให้ทันเหตุการณ์ หรือการเรียก

ใช้ข้อมูลเพื่อตรวจสอบความถูกต้องก่อนออกหลักฐานทางการศึกษาต่างๆ นอกจากนั้นข้อมูลบางอย่างซึ่งเป็นที่ต้องการของหน่วยงานต่างๆ สำนักทะเบียนและประมวลผลไม่สามารถออกได้อย่างรวดเร็ว ทันความต้องการของผู้ใช้ในระดัปล่างๆ ที่เกี่ยวข้อง

โปรแกรมระบบงานทะเบียนเดิม มีลักษณะการทำงานเป็นอิสระ ไม่เชื่อมต่อกับระบบอื่นใด แต่ข้อมูลอาจจะถูกเก็บไว้ห่างจากคอมพิวเตอร์ที่ใช้งานซึ่งอาจจะต้องใช้ระบบเครือข่ายต่างๆมาเกี่ยวข้อง ลักษณะโดยรวมของระบบเป็นดังภาพ



รูปที่ 3.1 ภาพรวมของระบบเดิม

ซึ่งลักษณะโดยทั่วไปของระบบเดิมมีดังนี้

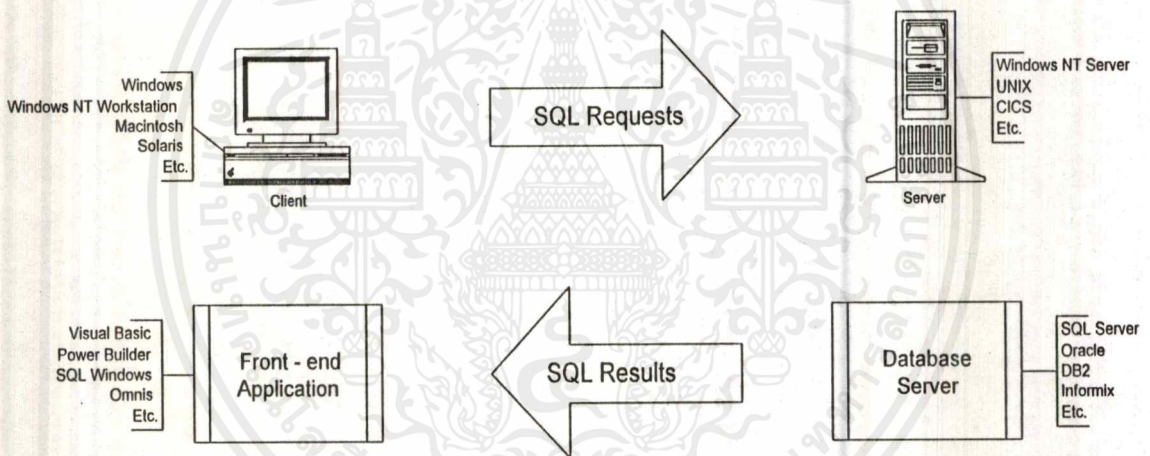
- โปรแกรมนี้จะทำการรับคำสั่งจากฟรอนต์เอนด์ ซึ่งเป็นโปรแกรมที่มีการทำงานภายใต้สภาพแวดล้อมของวินโดวส์ โปรแกรมจะต้องมีใช้งานที่ง่าย และสามารถทำความเข้าใจได้ง่าย ข้อมูลที่ป้อนเข้าสู่ระบบที่มีความสัมพันธ์กันควรป้อนเพียงครั้งเดียวเพื่อลดความซ้ำซ้อน โดยระบบจะต้องรองรับงานจำนวนมากได้เพราะระบบจะต้องบริการกับเทอร์มินอลหลายตัวพร้อมกัน และต้องเป็นการกระทำแบบเรียลไทม์ เพราะจะต้องมีการแก้ไขเปลี่ยนแปลงข้อมูลตลอดเวลา ความปลอดภัยของข้อมูลนั้นจะต้องมีการป้องกันตามสิทธิที่ได้รับของผู้ใช้แต่ละคน เอกสารที่ได้จะต้องเข้าใจง่าย รวมไปถึงความเร็วในการพิมพ์ข้อมูลต้องเหมาะสม

- โปรแกรมทำงานภายใต้สภาพแวดล้อมของวินโดวส์ที่มีการตั้งค่าความละเอียดของจอตามที่กำหนดไว้เท่านั้น รวมไปถึงชนิดของระบบปฏิบัติการและฮาร์ดแวร์ต้องเป็นไปตามที่กำหนดเท่านั้น

### 3.1.2 ปัญหาในการทำงาน

ในโครงการเดิมคือ "ระบบสารสนเทศของสำนักทะเบียนและประมวลผล" ได้ใช้สถาปัตยกรรมการให้บริการข้อมูลแบบ ไคลเอนต์-เซิร์ฟเวอร์ (Client / Server) ในการพัฒนาระบบ ซึ่งเป็นรูปแบบที่นิยมใช้ในปัจจุบัน เพราะการพัฒนาโปรแกรมทำได้ง่าย และสามารถนำส่วนฟรอนต์เอนด์ (Front-end ) ไปใช้กับเซิร์ฟเวอร์ใดก็ได้ที่มีมาตรฐานเดียวกัน แต่จากการศึกษาพบว่าข้อจำกัดของโครงการเดิมมีดังนี้

1. โปรแกรมเดิมเป็นการทำงานแบบทิวเทียร์ (Two-Tier) โดยส่วนฟรอนต์เอนด์จะทำการร้องขอข้อมูล(คำสั่ง SQL)ไปยังส่วนที่เก็บข้อมูลหรือเซิร์ฟเวอร์( โดยเซิร์ฟเวอร์จะเป็นที่เก็บและให้บริการข้อมูล) เมื่อเซิร์ฟเวอร์ได้รับคำสั่ง(SQL Code)แล้วจะทำการประมวลผล แล้วส่งผลลัพธ์ กลับไปยังส่วนฟรอนต์เอนด์ (ซึ่งก็คือไคลเอนต์)



รูปที่ 3.2 การติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์

ซึ่งการประมวลผลแบบนี้ มีข้อเสีย คือ

1.1 ความยากในการจัดการเปลี่ยนแปลงส่วนฟรอนต์เอนด์ เนื่องจากการเปลี่ยนแปลงโปรแกรมใหม่ หากมีการเปลี่ยนแปลงแก้ไขฐานข้อมูลที่เซิร์ฟเวอร์ เช่น การเพิ่ม Column ใน Table เป็นต้น เครื่องไคลเอนต์ทุกเครื่องจะต้องทำการเปลี่ยนแปลงโปรแกรมใหม่ ให้ตรงกับข้อมูลใหม่ที่มีการเปลี่ยนแปลงในเซิร์ฟเวอร์ ยิ่งระบบที่มีการแก้ไขกฎเกณฑ์ที่ใช้เสมอๆ จะทำให้เสียค่าบำรุงรักษาสูงโดยไม่จำเป็น

1.2 ความยากในการใช้โปรแกรมร่วมกัน การเขียนโปรแกรมโดยเอาส่วนบิตชีเนตโปรแกรมและคำสั่ง (Code) ใส่ใน แอปพลิเคชัน ทำให้โปรแกรม นั้นไม่สามารถนำมาใช้ใหม่กับแอปพลิเคชันอื่นได้

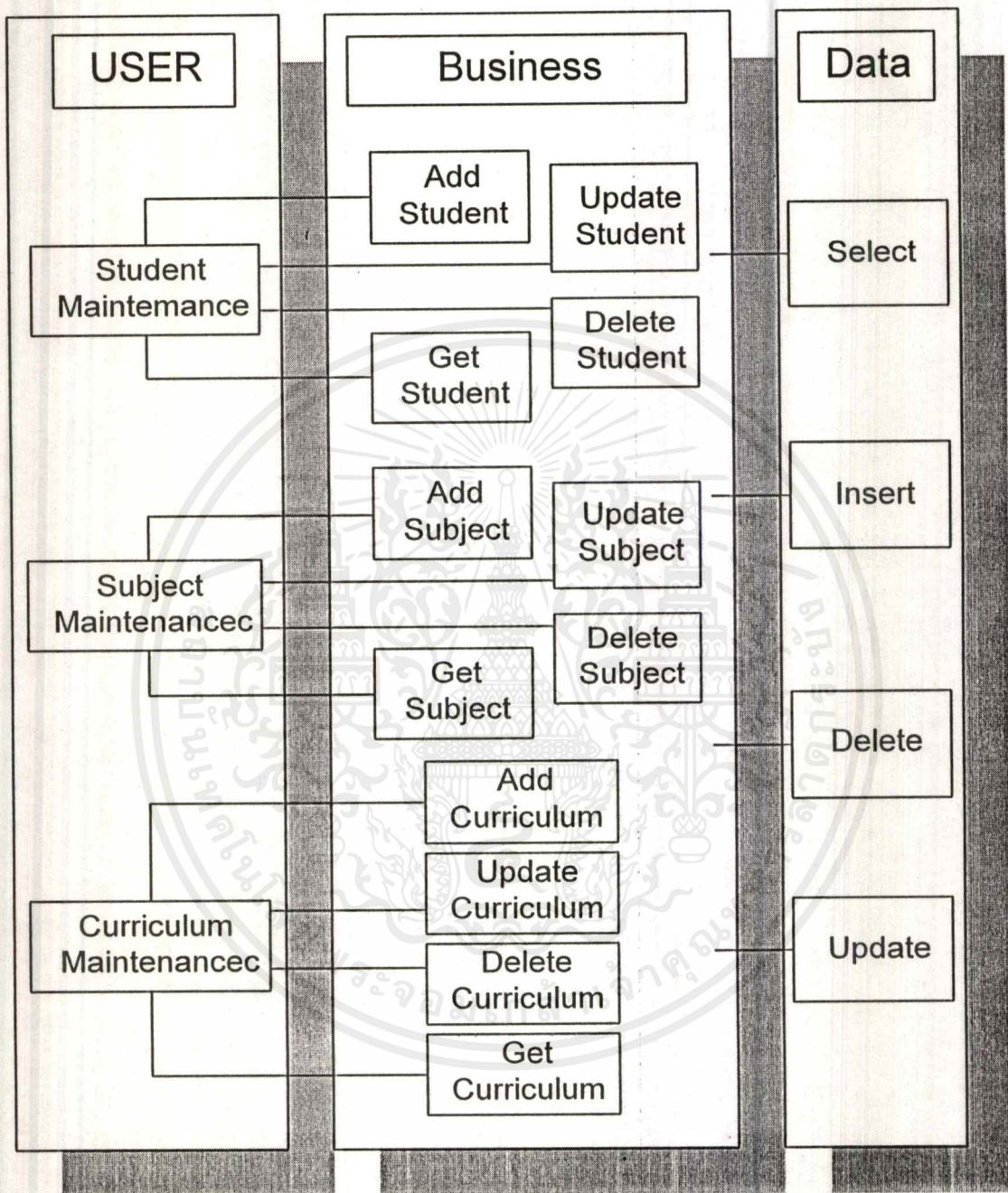
### 3.2 วัตถุประสงค์ของโครงการ

ในระบบเดิม เมื่อมีการแก้ไขกฎเกณฑ์ในโปรแกรมจะต้องตามไปแก้กฎเกณฑ์ที่เครื่องไคลด์เอนต์ทุกเครื่องเสมอไป ยิ่งระบบที่มีการแก้ไขกฎเกณฑ์ที่ใช้เสมอๆ จะทำให้เสียค่าบำรุงรักษาสูงโดยไม่จำเป็น เราสามารถแก้ปัญหาได้โดยการเพิ่มชั้น(Layer) เข้าไปแทรกระหว่างส่วนฟรอนต์เอนต์กับแบคเอนต์ (Back-end) ซึ่งชั้นที่แทรกเข้าไปเราเรียกว่า "บิสซิเนส เซอร์วิส" (Business Service) เป็นส่วนที่เก็บข้อจำกัดของผู้ใช้ ส่วนชั้นที่ติดต่อกับผู้ใช้เรียกว่า "ส่วนบริการผู้ใช้" (User Service) และเรียกส่วนที่ให้บริการข้อมูลว่า "ส่วนบริการข้อมูล" (Data Service) เมื่อมีชั้นที่ทำหน้าที่เป็นตัวกลางของการติดต่อระหว่างผู้ให้บริการเดิม (Data Service ซึ่งก็คือ DBMS) และส่วนให้บริการผู้ใช้ (User Service ซึ่งก็คือ Application) แล้ว เราก็สามารถเอากฎเกณฑ์ที่กำหนด เก็บไว้ในส่วนนี้ได้ (ส่วนบิสซิเนส) และเมื่อมีการแก้ไขจะสามารถแก้ไขที่ส่วนนี้ได้เลยโดยไม่ต้องพัฒนาโปรแกรมทั้งหมดใหม่ ทำให้การพัฒนาโปรแกรมใหม่ รวมถึงการบำรุงรักษาทำได้โดยง่าย ซึ่งในการให้บริการของส่วนการเก็บข้อจำกัดของผู้ใช้ เราจะใช้ โอเล่ ออโตเมชัน เซิร์ฟเวอร์ (OLE Automation Server) เป็นผู้ให้บริการ

### 3.3 ขอบเขตของโครงการ

สำหรับโครงการนี้ จะเป็นการทำงานในส่วนของทะเบียนประวัตินิติ และหลักสูตร ของสำนักทะเบียนกลาง หน้าที่ของส่วนนี้คือจัดเก็บข้อมูลประวัติของนิติ และสิ่งอื่นที่เกี่ยวข้อง เช่น บิดา , ประวัติการศึกษา เป็นต้น โดยระบบการจัดเก็บข้อมูลประวัติของนิติ และหลักสูตรนี้จะใช้สถาปัตยกรรมแบบทรีเทียร์ในการพัฒนา ซึ่งทำให้การบำรุงรักษาแก้ไขใหม่ในภายหลังสามารถทำได้ง่าย

## 3.3.1 ภาพรวมของระบบ



รูปที่ 3.3 ภาพรวมของระบบ

จากรูป ระบบจะประกอบด้วยโปรแกรมที่ติดต่อกับผู้ใช้ 3 ส่วน คือ Student Maintenance, Subject Maintenance และ Curriculum Maintenance ซึ่งโปรแกรมต่างๆเหล่านี้จะติดต่อกับส่วนให้บริการกฎเกณฑ์ตามชนิดของการทำงานต่างๆ เช่น การเพิ่ม,ลบ และส่วนให้บริการกฎเกณฑ์จะทำการติดต่อกับส่วนให้บริการข้อมูลอีกทีหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 ความสามารถของระบบ

#### ระบบมีความสามารถดังนี้

- สามารถค้นหาข้อมูลจากฐานข้อมูล รวมไปถึงการลบ,เพิ่ม และแก้ไขข้อมูล
- สามารถเป็นตัวรับข้อมูลเพื่อลดขั้นตอนของงานทะเบียน รวมไปถึงลดเวลาการทำงาน และจะต้องสามารถเก็บข้อมูลได้ทั้งภาษาไทย และภาษาอังกฤษ
- สามารถรับข้อมูลตามกฎเกณฑ์ของทางมหาวิทยาลัยได้
- เพิ่มระบบเสริมตามต้องการได้โดยง่าย
- สามารถเปลี่ยนแปลงกฎเกณฑ์ในระบบได้ง่าย โดยไม่จำเป็นต้องเปลี่ยนแปลงกฎเกณฑ์ที่โปรแกรมของผู้ใช้งาน การเปลี่ยนแปลงจะทำให้ส่วนให้บริการกฎเกณฑ์ (Business Service) เพียงอย่างเดียว ทำให้ลดค่าใช้จ่ายในการพัฒนาโปรแกรม

### 3.3.3 คุณสมบัติของระบบ

งานในส่วนของผู้ให้บริการข้อมูล (Data Service) นั้น จะต้องสามารถรองรับการทำงานพื้นฐานของระบบได้ เช่น การเพิ่มเติม, แก้ไข, ลบ, หรือการค้นหาข้อมูล(Query) ได้ และสำหรับข้อมูลบางส่วนที่สำคัญสามารถทำการสำรองข้อมูล(Backup)ได้ โดยจะมีระบบความปลอดภัยในการทำงาน ทั้งในส่วนขอระบบฐานข้อมูลเองโดยสามารถตั้งค่าได้โดยใช้ความสามารถของระบบจัดการฐานข้อมูล และในส่วนขอระบบงาน ผู้ใช้แต่ละคนจะมีสิทธิ (Permission) ในการเข้าสู่ระบบที่ต่างๆ กันตาม User Login ซึ่งสิทธิของผู้ใช้จะขึ้นอยู่กับหน้าที่รับผิดชอบของแต่ละคน

### 3.4 หลักการและวิธีการในการพัฒนาระบบ

ศึกษาความต้องการของระบบเดิม และทำการเปลี่ยนแปลงบางส่วนของระบบให้ดีขึ้นโดย

- ในส่วนของหน้าจอที่ติดต่อกับผู้ใช้ มีการเปลี่ยนแปลงใหม่ โดยลดจำนวนปุ่มการใช้งานลง เพื่อให้การทำงานเร็วขึ้น แต่ให้สิ่งที่จำเป็นสำหรับการใช้งานครบถ้วน
- กฎเกณฑ์ของระบบจะถูกย้ายจากส่วนแอปพลิเคชันไปยังส่วนที่เรียกว่า "Business Service" ซึ่งจะทำการเปลี่ยนแปลงแก้ไขโปรแกรมในส่วนขอกฎเกณฑ์ขอระบบ ไม่จำเป็นต้องพัฒนาโปรแกรมทั้งหมดใหม่
- ศึกษาส่วนการจัดเก็บข้อมูลในฐานข้อมูลที่จะนำมาใช้งาน
- เขียนสตอร์ โพรซีเจอร์ เพื่อให้ส่วนบริการกฎเกณฑ์(Business Service) เรียกใช้งาน ทำให้การทำงานกับข้อมูลที่มีอยู่ มีประสิทธิภาพดีขึ้น

### 3.5 แผนงานในการพัฒนาระบบ

ในภาคเรียนที่ 1 ได้แบ่งการดำเนินงานเป็น 2 ส่วน คือ

- ศึกษาและวิเคราะห์การทำงานของระบบเดิม
- ศึกษาโครงสร้างของระบบที่จะทำใหม่และศึกษาเครื่องมือที่จะช่วยในการพัฒนา

ในภาคเรียนที่ 2 ได้แบ่งการดำเนินงานเป็น 2 ส่วน คือ:

- ออกแบบระบบใหม่โดยอ้างอิงการทำงานเดิม
- ตรวจสอบความถูกต้องของระบบงานที่ได้ออกแบบไว้
- เขียนโปรแกรมในรายละเอียดส่วนต่างๆ
- ทำการทดสอบระบบงานที่ได้ออกแบบไว้รวมไปถึงการติดตั้งระบบ
- จัดทำเอกสารคู่มือการใช้งาน

โดยมีรายละเอียดขั้นตอนการดำเนินงาน ดังนี้

#### 3.5.1 แผนงานในการพัฒนาระบบ ภาคเรียนที่ 1

- วิเคราะห์ความต้องการของระบบโดยทำการศึกษาโครงสร้างการบริหารของมหาวิทยาลัยรวมทั้งหน่วยงานที่เกี่ยวข้องเพื่อใช้เป็นประโยชน์ในการวางแผนงานขั้นต่อไป

- ศึกษากฎเกณฑ์และระเบียบเกี่ยวกับการจัดการศึกษาเบื้องต้นในภาพรวมเพื่อใช้เป็นประโยชน์ในการวางแผนงานขั้นต่อไป ซึ่งจะทำให้เราทราบกฎเกณฑ์พื้นฐานและคุณสมบัติที่ใช้ในการเป็นนิสิตของมหาวิทยาลัย รวมไปถึงด้านการบริหารงานการศึกษาด้วย

- เนื่องจากระบบมีขนาดใหญ่และทำงานในหลายๆด้านจึงทำการแบ่งแยกระบบออกเป็นส่วนต่างๆและศึกษาความรับผิดชอบ กรรมวิธี และระยะเวลาของการทำงานในส่วนต่างๆ

- กำหนดคุณสมบัติของคอมพิวเตอร์ที่จะนำมาใช้ในระบบรวมถึงคุณสมบัติของซอฟต์แวร์

- ดำรงในรายละเอียดของระบบงานที่ทำ คือ งานทะเบียนประวัติ นิสิต และหลักสูตรการศึกษา

โดยทั้งหมดนี้จะมีการเข้าไปศึกษาระบบเดิมที่ใช้อยู่ว่ามีขั้นตอนการปฏิบัติอย่างไรบ้าง มีความต้องการเอกสารอะไรบ้าง มีลักษณะเป็นอย่างไร และหาวิธีการออกแบบระบบเพื่อแก้ไขปัญหาที่เกิดขึ้นรวมถึงเพิ่มประสิทธิภาพการทำงานให้ได้ดีตามวัตถุประสงค์ของระบบที่กำหนดไว้

- ศึกษารหัสข้อมูลของระบบเดิม ซึ่งรหัสข้อมูลนี้ จะถูกใช้อ้างอิงในการเขียนโปรแกรม

#### แกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ศึกษารายงานที่เกี่ยวกับแต่ละระบบ โดยอาศัยเอกสารเดิมที่มีอยู่
- ศึกษารายละเอียดของฐานข้อมูล หรือ Data Dictionary

### 3.5.2 แผนงานในการพัฒนาระบบ ภาคเรียนที่ 2

- ออกแบบส่วนเก็บกฎเกณฑ์ของระบบโดยอ้างอิงจากการทำงานของระบบเดิม
- ทำการออกแบบส่วนติดต่อผู้ใช้ทางหน้าจอของแต่ละระบบ โดยยังคงยึดถือความต้องการของผู้ใช้เดิม แต่จะมีการเปลี่ยนแปลงหน้าจอเพื่อเพิ่มประสิทธิภาพของโปรแกรมให้ดีขึ้น
- เขียนโปรแกรมตามที่ได้ออกแบบโดยแบ่งงานออกเป็นส่วนๆ แยกกันเขียนโปรแกรม โดยอาศัย Data Dictionary ของระบบเดิม เป็นตัวกำหนดมาตรฐานและยึดเป็นหลักในการเขียนโปรแกรม โดยการเขียนโปรแกรมนี้จะประกอบไปด้วย

1. การเขียนสตอร์ โพรซีเจอร์ ในส่วนให้บริการข้อมูลซึ่งจะทำหน้าที่ในการเพิ่ม,ลบ,เปลี่ยนแปลงข้อมูล และแสดงผลข้อมูล

2. การเขียนโปรแกรมส่วนให้บริการเก็บกฎเกณฑ์ข้อจำกัด (Business Service) ซึ่งมี 2 ส่วนคือ ส่วนเก็บรายละเอียดของออบเจกต์ในระบบและส่วนบริหารออบเจกต์นั้นๆ

3. การเขียนโปรแกรมในส่วนที่ติดต่อกับผู้ใช้งาน โดยโปรแกรมส่วนนี้จะไปเรียกใช้งานโปรแกรมส่วนที่ให้บริการเก็บกฎเกณฑ์ข้อจำกัดอีกครั้งหนึ่ง

- ทำการทดสอบระบบงานที่ได้ออกแบบไว้รวมไปถึงการติดตั้งระบบ โดยจะทำการทดสอบในแต่ละระบบย่อยโดยใช้ข้อมูลสมมติมาทำการทดสอบ หลังจากนั้นจะทำการทดสอบระบบรวมโดยทดลองทำงานจริง และอาจมีการปรับปรุงการออกแบบในด้านต่างๆ ไม่ว่าจะเป็นการออกแบบฐานข้อมูล Algorithm ในการทำงานเพื่อให้ระบบมีประสิทธิภาพสูงสุด เมื่อระบบมีประสิทธิภาพตามที่ต้องการแล้วจึงทำการติดตั้งระบบเพื่อใช้จริง

- จัดทำเอกสารคู่มือการใช้งาน

เป็นการนำเอกสารในขั้นตอนต่างๆมารวมกันและทำการเรียบเรียงและเพิ่มเติมส่วนประกอบอื่นๆ เพื่อจัดทำเอกสารเกี่ยวกับระบบซึ่งได้แก่

1. User document ซึ่งเป็นเอกสารสำหรับผู้ใช้งาน

2. System document ซึ่งเป็นเอกสารที่รวบรวมข้อมูลเกี่ยวกับระบบสำหรับผู้ควบคุมระบบ และใช้เป็นเอกสารอ้างอิงเพื่อการพัฒนาต่อไป

### 3.6 การพัฒนาระบบ

#### 3.6.1 ความต้องการด้านฮาร์ดแวร์

ระบบที่ใช้งานมีลักษณะการทำงานเป็นไคลเอนต์ เซิร์ฟเวอร์ เพราะฉะนั้น อุปกรณ์คอมพิวเตอร์ที่นำมาใช้ในระบบ ควรมีคอมพิวเตอร์ 1 ตัว ทำหน้าที่เป็นเซิร์ฟเวอร์ และคอมพิวเตอร์อีกจำนวนหนึ่งทำหน้าที่เป็นไคลเอนต์ โดยเครื่องคอมพิวเตอร์ที่ใช้ควรจะมีคุณลักษณะดังนี้

##### 1. เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์หรือผู้ให้บริการ

- หน่วยประมวลผลกลาง (CPU) เพนเทียม 75 Mhz ขึ้นไปหรือเทียบเท่า
- หน่วยความจำหลัก (Main Memory) ขนาดไม่ต่ำกว่า 32 เมกะไบต์ และสามารถขยายได้ในแผงวงจรแม่ (Mother Board) ไม่ต่ำกว่า 136 เมกะไบต์ โดยไม่ต้องเปลี่ยนหรือเพิ่มหน่วยประมวลผล
- หน่วยความจำความเร็วสูง (Cache Memory) ชนิดภายในหน่วยประมวลผลกลาง ขนาดไม่น้อยกว่า 16 กิโลไบต์ และชนิดภายนอกหน่วยประมวลผลกลางขนาดไม่น้อยกว่า 256 กิโลไบต์
- ระบบขับจานแม่เหล็ก (Magnetic Disk Subsystem) โดยใช้มาตรฐานการเชื่อมต่อแบบ SCSI-II พร้อมแผงควบคุม อย่างน้อย 2 ไครฟ์ และมีความจุรวมกันเมื่อฟอร์แมต (formatted) เสร็จแล้วไม่น้อยกว่า 1 กิกะไบต์ และต้องสามารถขยายได้ไม่น้อยกว่า 6 กิกะไบต์ภายในเครื่อง
- สามารถสนับสนุน SCSI-II DISK ARRAY
- ระบบเทปแม่เหล็ก (Tape Drive) โดยใช้มาตรฐานการเชื่อมต่อแบบ SCSI-II จำนวน 1 ชุด ขนาดความจุไม่ต่ำกว่า 2 กิกะไบต์
- เครื่องอ่านซีดีรอม โดยใช้มาตรฐานการเชื่อมต่อแบบ SCSI-II จำนวน 1 ชุด มีอัตราการถ่ายข้อมูล (Data transfer rate) อย่างน้อย 600 กิโลไบต์ต่อวินาทีขึ้นไป
- เครื่องขับจานแม่เหล็กอย่างอ่อน (Floppy Disk Drive) ขนาด 3.5 นิ้ว ความจุไม่ต่ำกว่า 1.44 เมกะไบต์ จำนวน 1 หน่วย
- มีช่องสื่อสารแบบอนุกรม (Serial Port) แบบ RS-232C ไม่น้อยกว่า 2 ช่อง และช่องสื่อสารแบบขนาน (Parallels Ports) อย่างน้อย 1 ช่อง
- มีช่องเสียบสำหรับแผงวงจรเสริม (Expansion Slots) ไม่น้อยกว่า 9 ช่อง ประกอบด้วย PCI 2 ช่อง EISA 7 ช่อง

- มีอุปกรณ์เชื่อมต่อกับระบบเครือข่าย (Network Interface Card) แบบ Ethernet ซึ่งสามารถใช้กับระบบเคเบิลแบบ Unshielded Twisted Pair (RJ-45) ตามมาตรฐาน IEEE 802.3 10BaseT จำนวน 1 ชุด

- แผงวงจรแสดงผลทางกราฟฟิก (Graphic Board) แบบ SVGA ซึ่งสามารถแสดงรายละเอียดได้ไม่น้อยกว่า 1,024 x 768 pixels ต่อเชื่อมกับระบบคอมพิวเตอร์ในระบบ Graphics Accelerated Local Bus แบบ PCI

- จอภาพสีชนิดรายละเอียดสูง ขนาดไม่ต่ำกว่า 14 นิ้ว ตามเส้นทแยงมุม ซึ่งสามารถใช้แสดงภาพที่ได้รับสัญญาณจากแผงวงจรแสดงผลกราฟฟิก รายละเอียดได้ไม่น้อยกว่า 1,024 x 768 pixels แบบ Non-interlace

- คีย์บอร์ด (Keyboard) ที่มีอักษรภาษาไทย/อังกฤษ ตัวเลขและเครื่องหมาย สัญลักษณ์พิเศษอย่างน้อย 101 คีย์

- มีอุปกรณ์ป้อนคำสั่งแบบเมาส์ (Mouse) ที่มีปุ่ม 2 ปุ่ม จำนวน 2 ชุด

## 2. เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอนต์

- หน่วยประมวลผลกลาง 486DX2 -66 เมกะเฮิร์ต ขึ้นไปหรือเทียบเท่า

- มีหน่วยความจำหลัก ขนาดไม่ต่ำกว่า 16 เมกะไบต์ และสามารถขยายได้ในแผงวงจรแม่ขนาดไม่ต่ำกว่า 128 เมกะไบต์ โดยไม่ต้องเปลี่ยนหรือเพิ่มหน่วยประมวลผลกลาง

- หน่วยความจำความเร็วสูง ชนิดภายในหน่วยประมวลผลกลาง ขนาดไม่น้อยกว่า 16 กิโลไบต์ และชนิดภายนอกหน่วยประมวลผลกลางขนาดไม่น้อยกว่า 256 กิโลไบต์

- ระบบขับจานแม่เหล็ก โดยใช้มาตรฐานการเชื่อมต่อแบบ IDE ความจุเมื่อฟอร์แมตเสร็จแล้วไม่น้อยกว่า 340 เมกะไบต์

- เครื่องขับจานแม่เหล็กอย่างอ่อน ขนาด 3.5 นิ้ว ความจุไม่ต่ำกว่า 1.44 เมกะไบต์ จำนวน 1 หน่วย

- มีช่องสื่อสารแบบอนุกรมแบบ RS-232C ไม่น้อยกว่า 2 ช่อง และช่องสื่อสารแบบขนานอย่างน้อย 1 ช่อง

- มีช่องเสียบสำหรับแผงวงจรเสริม แบบ 16 บิต ไม่น้อยกว่า 3 ช่อง

- มีอุปกรณ์เชื่อมต่อกับระบบเครือข่ายแบบ Ethernet ซึ่งสามารถใช้กับระบบเคเบิลแบบ Thick Ethernet (AUI) ตามมาตรฐาน IEEE 802.3 และระบบเคเบิลแบบ Thick Ethernet (BNC) ตามมาตรฐาน IEEE 802.3 และ Unshielded Twisted Pair (RJ-45) ตามมาตรฐาน IEEE 802.3

โดยมีความเร็วในการส่งข้อมูลไม่น้อยกว่า 10 เมกะบิตต่อวินาที และผ่านมาตรฐาน FCC Class B

- ระบบแสดงผลประกอบด้วย Graphics Accelerated Local Bus พร้อมความ  
จำอย่างน้อย 1 เมกะไบต์ และจอภาพสีขนาดไม่ต่ำกว่า 14 นิ้วตามเส้นทแยงมุม รายละเอียดได้ไม่  
น้อยกว่า 800 x 600 pixels 256 สี แบบ Non-interlace

- คีย์บอร์ด ที่มีอักษรภาษาไทย/อังกฤษ ตัวเลขและเครื่องหมายสัญลักษณ์  
พิเศษอย่างน้อย 101 คีย์

- มีอุปกรณ์ป้องกันคำสั่งแบบเม้าส์ที่มีปุ่ม 2 ปุ่ม จำนวน 2 ชุด

3. อุปกรณ์รวมข้อมูลบนเครือข่ายในระบบเคเบิล แบบ UTP (Ethernet Concentra-  
tor) จำนวน 1 ชุด

4. เครื่องพิมพ์เลเซอร์เพื่อใช้งาน เป็นเครื่องพิมพ์สำหรับระบบเครือข่าย จำนวน 1  
ชุด ซึ่งมีลักษณะ

- มีความละเอียดในการพิมพ์ไม่น้อยกว่า 600 x 600 จุดต่อนิ้ว

- มีหน่วยความจำไม่น้อยกว่า 16 เมกะไบต์

- มีความเร็วในการพิมพ์ 20 แผ่นต่อนาที

- มีอุปกรณ์เชื่อมต่อเข้ากับเครื่อง ไมโครคอมพิวเตอร์ในระบบเครือข่าย เพื่อ  
ใช้งานสำหรับเป็นเครื่องพิมพ์สำหรับระบบเครือข่าย

5. เครื่องควบคุมกระแสไฟฟ้าและสำรองไฟฟ้า (Uninterruptable Power Supply)  
เป็นแบบ On-line UPS ขนาดไม่ต่ำกว่า 3 KVA

### 3.6.2 ความต้องการด้านซอฟต์แวร์

#### 1. เครื่องที่ทำหน้าที่เป็นเซิร์ฟเวอร์

- ระบบปฏิบัติการไมโครซอฟต์วินโดวส์เอ็นที เซิร์ฟเวอร์ เวอร์ชัน 3.51 ขึ้นไป  
(Microsoft Windows NT Server 3.51)

- ระบบจัดการฐานข้อมูล ไมโครซอฟต์ เอสคิวเอล เซิร์ฟเวอร์ เวอร์ชัน 6.5  
ขึ้นไป (Microsoft SQL Server 6.5)

- โปรแกรมพัฒนาระบบ ไมโครซอฟต์ วิซวลเบสิก เวอร์ชัน 4.0 รุ่นเอ็นเทอร์  
ไพรซ์ (MicroSoft Visual Basic Version 4.0 Enterprise Edition)

#### 2. เครื่องที่ทำหน้าที่เป็น WorkStation

- ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 95 รุ่นภาษาไทย(Microsoft  
Windows 95 Thai Edition)

- โปรแกรมพัฒนาระบบ ไมโครซอฟต์ วิซวลเบสิก เวอร์ชัน 4.0 รุ่นเอ็นเทอร์  
ไพรซ์ (MicroSoft Visual Basic Version 4.0 Enterprise Edition)

### 3.7 รายละเอียดการทำงานของโครงการ

#### 3.7.1 ระบบทะเบียนประวัตินักศึกษา

ข้อมูลที่นำเข้าระบบ สามารถแบ่งออกได้ดังนี้

- ชื่อ และชื่อสกุลภาษาไทย แยกเป็นสองส่วน
- ชื่อ และชื่อสกุลภาษาอังกฤษ แยกเป็นสองส่วน
- เพศ
- คำนำนาม เช่น นาย นาง นางสาว ยศ ราชทินนาม
- วัน/เดือน/ปี เกิด
- จังหวัดภูมิลำเนา
- วุฒิเดิม ( ก่อนเข้าศึกษา )
- ศาสนา
- เชื้อชาติ
- ที่อยู่ เลขที่
- ตรอก/ชอย
- ถนน
- แขวง

- เขต
- จังหวัด
- โทรศัพท์
- รหัสไปรษณีย์
- วันที่/เดือน/ปี ที่เข้าศึกษา
- ประเภทของการรับเข้า ประกอบด้วย
  - โควต้าโครงการส่งเสริมโอกาสการศึกษาต่อในมหาวิทยาลัยเกษตรศาสตร์
  - โควต้าโครงการนักกีฬาดีเด่น
  - โควต้าโครงการกระทรวงมหาดไทย นิสิตไทยอิสลาม
  - โควต้าโครงการ พ.ศ.ว.ท.
  - โควต้าโครงการ ส.ศ.ว.ท.
  - โครงการคณะวิศวกรรมศาสตร์ภาคพิเศษ
  - โครงการคณะอุตสาหกรรมเกษตร ภาคพิเศษ
  - โครงการศึกษาต่ออีกหนึ่งปริญญา คณะวิศวกรรมศาสตร์
  - โครงการพิเศษ โรงเรียนสัตวแพทย์ กรมปศุสัตว์
  - โครงการข้าราชการลาศึกษาต่อ
  - อื่น ๆ
  - สถาบันสมทบ
  - วิทยาลัยชลประทาน
  - โรงเรียนสัตวแพทย์ กรมปศุสัตว์ (เทคนิคการสัตวแพทย์)
  - ทุนการศึกษารัฐบาลระดับปริญญาตรี นิสิตต่างประเทศ
  - สอบผ่านทางทบวงมหาวิทยาลัย
  - รับโอนจากสถาบันการศึกษาอื่น
  - รหัสคณะ
  - รหัสภาควิชา
  - รหัสสาขาวิชาเอก
  - รหัสอาจารย์ที่ปรึกษา
  - วิทยาเขต

### 3.7.1.1 การทำงานของระบบ มีดังนี้

- ข้อมูลของนิสิตใหม่จะถูกนำเข้าและเก็บลงในระบบฐานข้อมูลจากทางจอภาพ ( on line ) ในกรณีของนิสิตโควตาประเภทต่าง ๆ นิสิตในโครงการคณะวิศวกรรมศาสตร์ ภาคพิเศษ และนิสิตในโครงการคณะอุตสาหกรรมเกษตร ภาคพิเศษ เป็นต้น ยกเว้น นิสิตที่สอบผ่านทบวงมหาวิทยาลัย ข้อมูลจะมาในรูปแบบทฤษ์ไฟล์ (Text File) จากทบวงมหาวิทยาลัย

- ระบบจะออกรหัสประจำตัวให้กับนิสิตใหม่โดยจัดเรียงชื่อ - นามสกุลตาม พจนานุกรมแยกตามคณะ ประเภทวิชาและประเภทของการรับเข้า

- เลขประจำตัวนิสิตประกอบด้วยเลข 8 หลัก คือ ตัวเลขสองหลักแรก เป็นปีที่เข้าศึกษา ตัวเลขสองหลักถัดมา เป็นรหัสคณะ ตัวเลขสามตัวถัดมา เป็นหมายเลขลำดับของนิสิตในคณะนั้น ๆ ตัวเลขตัวสุดท้ายเป็น check digit

- ระบบจะจัดทำ / พิมพ์ชื่อ ที่อยู่ของผู้ปกครองนิสิตลงบนสติกเกอร์ ( sticker ) สำหรับติดหน้าซองจดหมาย หรือบนด้านหลังเอกสารที่ต้องจัดส่งไปให้ผู้ปกครองนิสิต โดยมีตำแหน่งตรงกับกรอบกระดาษใส่ที่เปิดเป็นช่องไว้ที่หน้าซองจดหมาย

### 3.7.1.2 การแก้ไขข้อมูลของนิสิต

การทำงานของระบบ มีดังนี้

การแก้ไขข้อมูลของนิสิตจะทำในลักษณะ on-line เพื่อแก้ไขเปลี่ยนแปลงข้อมูลในฐานข้อมูลการเปลี่ยนข้อมูลของนิสิตจะทำได้ทุกหัวข้อ ยกเว้นเลขประจำตัวของนิสิตเท่านั้น การขอและเก็บหลักฐานในการเปลี่ยนข้อมูล เช่น หลักฐานการเปลี่ยนชื่อหรือที่อยู่ จะเก็บเป็นแฟ้มเอกสาร

### 3.7.1.3 การฟื้นฟูสภาพนิสิต

ข้อมูลที่นำเข้าสู่ระบบ สามารถแบ่งออกได้ดังนี้

- รหัสประจำตัวนิสิต
- วันที่/เดือน/ปี และภาคการศึกษาที่ฟื้นฟูสภาพนิสิต
- สาเหตุ

การทำงานของระบบ มีดังนี้

ข้อมูลต่าง ๆ จะถูกนำเข้าในระบบในลักษณะ on-line และระบบจะทำการเปลี่ยนสภาพของนิสิตในระบบ

### 3.7.1.4 รายงานที่ได้จากระบบ

ระบบจะต้องสามารถออกรายงานได้ทั้งทางจอภาพและทางเครื่องพิมพ์ โดยสามารถพิมพ์รายงานต่าง ๆ ได้ดังนี้

- รายละเอียดของนิสิตจัดเรียงตามรหัสประจำตัวนิสิต
- รายละเอียดของนิสิตจัดเรียงตามชื่อและนามสกุล
- รายละเอียดของนิสิตจัดเรียงตามคณะ/ภาควิชา/สาขาวิชาเอก/ชั้นปี
- รายละเอียดของนิสิตจัดเรียงตามปี/ภาคการศึกษาที่นิสิตเข้ารายงานตัว
- รายละเอียดของนิสิตจัดเรียงตามปี/ภาคการศึกษาที่นิสิตพ้นสภาพ

### 3.7.2 ระบบข้อมูลหลักสูตรการศึกษา สามารถแบ่งออกได้ดังนี้

การนำรายละเอียดของวิชาเข้าระบบ

- รหัสวิชา
- ชื่อวิชา ( ภาษาไทย )
- ชื่อวิชา ( ภาษาอังกฤษ )
- คณะ
- หมู่เรียน บรรยายและ/หรือปฏิบัติการ
- จำนวนหน่วยกิตรวมของวิชา
- จำนวนหน่วยกิตของวิชา ( บรรยาย )
- ค่าลงทะเบียนหน่วยกิตละ
- จำนวนหน่วยกิตของวิชา ( ปฏิบัติการ )
- ค่าลงทะเบียนหน่วยกิตละ
- ปี/ภาคการศึกษาที่เริ่มใช้

#### 3.7.2.1 การทำงานของระบบ มีดังนี้

รายละเอียดของวิชาจะถูกจัดเข้าระบบฐานข้อมูลในลักษณะ on-line

#### 3.7.2.2 การแก้ไขข้อมูลเกี่ยวกับวิชาในระบบ มีการทำงานดังนี้

ข้อมูลสำหรับวิชาจะสามารถเปลี่ยนแปลงแก้ไขในลักษณะ on-line ตาม

การค้นหาคำหรือชื่อวิชา

### 3.7.2.3 การยกเลิกวิชาในระบบ

ข้อมูลที่น่าเข้าระบบ สามารถแบ่งออกได้ดังนี้

- รหัสวิชา
- ปี/ภาคการศึกษาที่เลิกใช้

### 3.7.2.4 รายงานที่ได้จากระบบ

ระบบจะทำรายงานตามความต้องการต่อไปนี้ โดยสามารถออกรายงานได้ทั้งทางจอภาพและเครื่องพิมพ์

- รายละเอียดของวิชาโดยจัดเรียงตามรหัสวิชา
- รายละเอียดของวิชาโดยจัดเรียงตามรายชื้อวิชา
- ชื่อและรายละเอียดของวิชาโดยสังเขป โดยจัดเรียงตามคณะ
- ชื่อและรายละเอียดของวิชาโดยสังเขป โดยจัดเรียงตามปี/ภาคการศึกษาที่ใช้
- ชื่อและรายละเอียดของวิชาโดยสังเขป โดยจัดเรียงตามปี/ภาคการศึกษาที่เลิกใช้
- ชื่อและรายละเอียดของวิชาโดยสังเขป โดยจัดเรียงตามวิชาที่มีการปฏิบัติ
- ชื่อและรายละเอียดของวิชาโดยสังเขป โดยจัดเรียงตามวิชาที่ไม่มี การปฏิบัติ การ
- ชื่อและรายละเอียดของวิชาโดยสังเขป โดยจัดเรียงตามวิชาที่มีและไม่มี การปฏิบัติการ

## 3.8 การออกแบบระบบในส่วนของผู้ใช้ที่ 1 (User Service)

เริ่มจากการที่เราออกแบบฟอร์มล็อกอิน(Login) ซึ่งจะคอยรับข้อมูลว่ามีผู้ใดที่ต้องการจะเข้ามาใช้งานในระบบ โดยการออกแบบจะออกแบบให้มีลักษณะพื้นฐานดังนี้

รหัสผ่านเข้าสู่ระบบ	
โปรดใส่ชื่อและรหัสผ่านเข้าสู่ระบบ	
ชื่อผู้ใช้งาน	<input type="text"/>
รหัสผ่าน	<input type="password"/>
<input type="button" value="ตกลง"/>	
<input type="button" value="ยกเลิก"/>	

รูปที่ 3.4 ฟอร์มล็อกอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในองค์กรเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟอร์มล๊อคอินที่ออกแบบไว้ เราจะทำการคอมไพล์เป็นดีแอลแอลไฟล์ (.DLL) ซึ่งจะเป็นอินโพรเซส เซิร์ฟเวอร์ และทำการลงทะเบียนไว้ที่เครื่องโดยใช้คำสั่ง Regsvr32.exe LogonMgr.dll ซึ่งสามารถให้แอปพลิเคชันตัวอื่นใช้ได้ LogonMgr.dll ร่วมกันได้ในการเป็นฟอร์มล๊อคอินของแต่ละระบบ

หน้าจอที่เป็นส่วนติดต่อแรกของโปรแกรม จะเป็นข้อเลือกให้ผู้ใช้สามารถเข้าไปใช้งานยังส่วนต่างๆ ที่แตกต่างกันได้ ดังนี้



รูปที่ 3.5 ฟอร์มข้อเลือก

ประวัตินิสิต

โปรแกรมสำหรับสำนักทะเบียนกลางและประมวลผล

วันที่ 8/4/2540  
UserID

ประวัติส่วนตัว

รหัสประจำตัวนักศึกษา:  เพศชาย  เพศหญิง

นาย  ชื่อไทย:  นามสกุลไทย:

Mr.  ชื่ออังกฤษ:  นามสกุลอังกฤษ:

เชื้อชาติ:  สัญชาติ:  ศาสนา:

วันเกิด:  /  /  กลุ่มเลือด:  ส่วนสูง:  น้ำหนัก:

ที่อยู่ปัจจุบัน

บ้านเลขที่:  ตรอก/ซอย:  ถนน:

แขวง/ตำบล:  เขต/อำเภอ:  จังหวัด:

รหัสไปรษณีย์:  เบอร์โทรศัพท์:

ที่อยู่ตามทะเบียนบ้าน

บ้านเลขที่:  ตรอก/ซอย:  ถนน:

แขวง/ตำบล:  เขต/อำเภอ:  จังหวัด:

รหัสไปรษณีย์:  เบอร์โทรศัพท์:

ประวัติการศึกษา

ผู้ปกครอง

สถานภาพศึกษา

เพิ่มข้อมูล

ลบข้อมูล

แก้ไขข้อมูล

ตรวจสอบข้อมูล

ยกเลิก

กลับสู่รายการหลัก

รูปที่ 3.6 ฟอรัมประวัตินิสิต

หน้าที่ของหน้าจอฟอรัมประวัตินิสิต

- การรับข้อมูลในส่วนของประวัตินิสิต มาเก็บไว้ในฐานข้อมูล
- การดึงข้อมูลจากฐานข้อมูลเกี่ยวกับประวัตินิสิตออกมาแสดงผล
- การแก้ไขข้อมูลในส่วนของประวัตินิสิต
- การลบข้อมูลในส่วนของประวัตินิสิต

ประวัติส่วนการศึกษา

โปรแกรมสำหรับสำนักทะเบียนกลางและประมวลผล

วันที่ 8/4/2540  
Userid

ประวัติการศึกษา

วุฒิเดิม(ก่อน เข้าศึกษา)

โรงเรียน

จังหวัด  คณะแผนก ผลิต

ประเภทการรับเข้า

สอบผ่านทบวงมหาวิทยาลัย

โควต้าโครงการส่งเสริมการศึกษาต่อในมหาวิทยาลัยเกษตรศาสตร์

สถาบันสมทบ

ทนการศึกษารัฐบาลระดับปริญญาตรี นิสิตต่างประเทศ

รับโอนจากสถาบันการศึกษาอื่น

วัน/เดือน/ปี ที่เข้ารับการศึกษ  /  /

วัน/เดือน/ปี ที่จบการศึกษ  /  /

ประวัติส่วนตัว

ผู้ปกครอง

สถานภาพศึกษา

รหัส

ตรวจสอบ

แก้ไข

ยกเลิก

กลับสู่รายการหลัก

รูปที่ 3.7 ฟอรัมประวัตินิสิตส่วนการศึกษา

หน้าที่ของหน้าจอฟอรัมประวัตินิสิตส่วนการศึกษา

- การรับข้อมูลในส่วนของประวัตินิสิตส่วนการศึกษา มาเก็บไว้ในฐานข้อมูล
- การดึงข้อมูลจากฐานข้อมูลเกี่ยวกับประวัตินิสิตส่วนการศึกษาออกมาแสดงผล
- การแก้ไขข้อมูลในส่วนของประวัตินิสิตส่วนการศึกษา
- การลบข้อมูลในส่วนของประวัตินิสิตส่วนการศึกษา

รายละเอียดผู้ปกครอง

วันที่ 8/4/2540

### โปรแกรมสำนักทะเบียนกลางและประมวลผล

**ผู้ปกครอง**

ชื่อผู้ปกครอง  อาชีพ/ตำแหน่ง

ความสัมพันธ์  โทรศัพท์ที่ทำงาน

สถานที่ทำงาน  โทรสาร

ที่อยู่

รหัสไปรษณีย์  โทรศัพท์ที่บ้าน

**บิดามารดา**

ชื่อบิดา  อาชีพ/ตำแหน่ง

ที่อยู่  โทรศัพท์ที่ทำงาน

สถานที่ทำงาน  โทรศัพท์ที่บ้าน

ชื่อมารดา  อาชีพ/ตำแหน่ง

ที่อยู่  โทรศัพท์ที่ทำงาน

สถานที่ทำงาน  โทรศัพท์ที่บ้าน

รหัส

ประวัติส่วนตัว

ประวัติการศึกษา

สถานภาพการศึกษา

เพิ่มข้อมูล

ลบข้อมูล

แก้ไขข้อมูล

ตรวจสอบ

ยกเลิก

กลับสู่รายการหลัก

### รูปที่ 3.8 ฟอรัมประวัตินิติสดส่วนผู้ปกครอง

หน้าที่ของหน้าจอฟอรัมประวัตินิติสดส่วนผู้ปกครอง

- การรับข้อมูลในส่วนของประวัตินิติสดส่วนผู้ปกครอง มาเก็บไว้ในฐานข้อมูล
- การดึงข้อมูลจากฐานข้อมูลเกี่ยวกับประวัตินิติสดส่วนผู้ปกครองออกมาแสดงผล
- การแก้ไขข้อมูลในส่วนของประวัตินิติสดส่วนผู้ปกครอง
- การลบข้อมูลในส่วนของประวัตินิติสดส่วนผู้ปกครอง

สถานภาพทางการศึกษา

โปรแกรมสำนักทะเบียนกลางและประมวลผล วันที่ 8/4/2540

ชั้นปีที่  รหัสประจำตัวนักศึกษา

คณะ

ภาควิชา

สาขาวิชาเอก

สาขาวิชาโท

หลักสูตร

รหัสอาจารย์ที่ปรึกษา

ชื่อไทย

นามสกุลไทย

ประวัติส่วนตัว

ประวัติการศึกษา

ผู้ปกครอง

ตกลง

ยกเลิก

กลับสู่รายการหลัก

รูปที่ 3.9 ฟอรัมประวัตินิสัยส่วนสถานภาพทางการศึกษา

หน้าที่ของหน้าจอฟอรัมประวัตินิสัยส่วนสถานภาพทางการศึกษา

- การรับข้อมูลในส่วนของสถานภาพทางการศึกษา มาเก็บไว้ในฐานข้อมูล
- การดึงข้อมูลจากฐานข้อมูลเกี่ยวกับสถานภาพทางการศึกษาออกมาแสดงผล
- การแก้ไขข้อมูลในส่วนของสถานภาพทางการศึกษา
- การลบข้อมูลในส่วนของสถานภาพทางการศึกษา

นอกจากฟอร์มส่วนทะเบียนประวัตินิติแล้ว ยังมีส่วนของหลักสูตรการศึกษาและวิชา ซึ่งมีฟอร์มแรกที่เข้าไปใช้งานดังนี้

รูปที่ 3.10 ฟอร์มข้อเลือกของหลักสูตรการศึกษาและวิชา

ฟอร์มนี้ จะช่วยให้ผู้ใช้งานสามารถเลือกได้ว่า จะเข้าไปใช้งานแอปพลิเคชันในส่วนใด ซึ่งมีแอปพลิเคชันที่ทำหน้าที่หลักๆอยู่ 2 อย่าง คือ

- แอปพลิเคชันที่ดูแลเรื่องหลักสูตรการศึกษา
- แอปพลิเคชันที่ดูแลเรื่องวิชาเรียนต่างๆ

หลักสูตรการศึกษา

โปรแกรมสำนักทะเบียนกลางและประมวลผล

วันที่ 8/4/2540  
User id

เพิ่มข้อมูลหลักสูตร

รหัสของหลักสูตร

ชื่อหลักสูตร(ไทย)

ชื่อหลักสูตร(อังกฤษ)

อักษรย่อชื่อหลักสูตร(ไทย)

อักษรย่อชื่อหลักสูตร(อังกฤษ)

รหัสสคณ

รหัสภาควิชา

ศึกษาทั่วไป

วิชาเฉพาะ

เลือกเสรี

ฝึกงาน

โครงสร้างหลักสูตร

เพิ่มหลักสูตร

แก้ไขหลักสูตร


ลบหลักสูตร

ตรวจสอบ

ตกลง

ยกเลิก

กลับสู่รายการหลัก

รหัสวิชา	ชื่อวิชา	กลุ่มวิชา	จำนวนหน่วยกิตภาค...	จำนวนหน่วยกิตภาค
				

จำนวนหน่วยกิตรวมของหลักสูตร

### รูปที่ 3.11 φόρμหลักสูตรการศึกษา

หน้าที่ของหน้าจอฟอร์มหลักสูตรการศึกษา

- การรับข้อมูลในส่วนของหลักสูตรการศึกษา มาเก็บไว้ในฐานข้อมูล
- การดึงข้อมูลจากฐานข้อมูลเกี่ยวกับหลักสูตรการศึกษาออกมาแสดงผล
- การแก้ไขข้อมูลในส่วนของหลักสูตรการศึกษา
- การลบข้อมูลในส่วนของหลักสูตรการศึกษา

ระบบการแก้ไขวิชาเรียน

วันที่ 8/4/2540  
Userid

### โปรแกรมสำนักทะเบียนกลางและประมวลผล

รายละเอียดของวิชา

รหัสวิชา  ชั้นปีที่เรียน

ชื่อวิชา(ไทย)  ปีที่เริ่มใช้

ชื่อวิชา(อังกฤษ)  ปีที่เลิกใช้

คณะ  ภาควิชา

รายละเอียด

Course Description

หมวดวิชา

ศึกษาทั่วไป

วิชาเฉพาะ

การฝึกงาน

เลือกเสรี

หน่วยกิตบรรยาย  หน่วยกิตปฏิบัติ

ภาคการศึกษาที่เริ่มใช้

ภาคการศึกษาที่เลิกใช้

กลุ่มวิชา

ความหมาย

เพิ่มวิชา

แก้ไขวิชา

ลบวิชา

ตรวจสอบ

ตกลง

ยกเลิก

กลับสู่รายการหลัก

รูปที่ 3.12 ฟอรัมวิชาเรียน

## หน้าที่ของหน้าจอฟอร์มวิชาเรียน

- การรับข้อมูลในส่วนของวิชาเรียน มาเก็บไว้ในฐานข้อมูล
- การดึงข้อมูลจากฐานข้อมูลเกี่ยวกับวิชาเรียนออกมาแสดงผล
- การแก้ไขข้อมูลในส่วนของวิชาเรียน
- การลบข้อมูลในส่วนของวิชาเรียน

เพิ่ม-ลบ วิชาในหลักสูตร

วันที่ 8/4/2540

โปรแกรมสำหรับทะเบียนกลางและประมวลผล

เพิ่ม-ลบ วิชาในหลักสูตร

รหัสหลักสูตร

รหัสวิชา

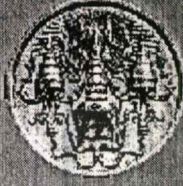
ชื่อวิชา

กลุ่มวิชา

ชั้นปีที่เรียน

ภาคการศึกษา

หมวดวิชาทั่วไป  
 หมวดวิชาเฉพาะ  
 หมวดวิชาเลือกเสรี  
 หมวดวิชาฝึกงาน



รูปที่ 3.13 ฟอรั่มเพิ่ม-ลบวิชาเรียนในหลักสูตร

หน้าที่ของหน้าจอฟอรั่มเพิ่ม-ลบวิชาเรียนในหลักสูตร

- การรับข้อมูลในส่วนของวิชาเรียนในแต่ละหลักสูตร มาเก็บไว้ในฐานข้อมูล
- การดึงข้อมูลจากฐานข้อมูลเกี่ยวกับวิชาเรียนในแต่ละหลักสูตรออกมาแสดงผล
- การลบข้อมูลของวิชาเรียนในหลักสูตร

### 3.9 การออกแบบส่วนบิสซิเนส เซอร์วิส

กำหนดบิสซิเนส เอนตี้ของระบบ โดยมีเอนตี้หลักๆ ที่เป็นออบเจกต์ดังนี้

ประวัตินักศึกษา	ประวัติการศึกษา	ประวัติผู้ประกอบการ
เลขประจำตัว	วุฒิเดิม	ชื่อ(ผู้ประกอบการ)
นามสกุล (ไทย)	โรงเรียน	อาชีพ
ชื่อ (อังกฤษ)	จังหวัด	สถานที่ทำงาน
นามสกุล (อังกฤษ)	คะแนนเฉลี่ย	โทรศัพท์
คำนำหน้า (ไทย)	วันที่เข้าศึกษา	โทรสาร
คำนำหน้า (อังกฤษ)	วันที่จบการศึกษา	ความสัมพันธ์
วันเดือนปี เกิด	ประเภทการรับเข้า	บ้านเลขที่
เพศ		ชอย
เชื้อชาติ		ถนน
สัญชาติ		แขวง
ศาสนา		เขต
กลุ่มเลือด		จังหวัด
ส่วนสูง		ชื่อ(บิดา)
น้ำหนัก		อาชีพ(บิดา)
บ้านเลขที่ (ปัจจุบัน)		สถานที่ทำงาน(บิดา)
ชอย (ปัจจุบัน)		โทรศัพท์ที่ทำงาน(บิดา)
ถนน (ปัจจุบัน)		ที่อยู่(บิดา)
แขวง (ปัจจุบัน)		โทรศัพท์ที่บ้าน(บิดา)
เขต (ปัจจุบัน)		ชื่อ(มารดา)
จังหวัด (ปัจจุบัน)		อาชีพ(มารดา)
รหัสไปรษณีย์ (ปัจจุบัน)		สถานที่ทำงาน(มารดา)
โทรศัพท์ (ปัจจุบัน)		โทรศัพท์ที่ทำงาน(มารดา)
โทรสาร (ปัจจุบัน)		ที่อยู่(มารดา)
บ้านเลขที่ (ตามทะเบียนบ้าน)		โทรศัพท์ที่บ้าน(มารดา)
ชอย (ตามทะเบียนบ้าน)		
ถนน (ตามทะเบียนบ้าน)		
แขวง (ตามทะเบียนบ้าน)		
เขต (ตามทะเบียนบ้าน)		
จังหวัด (ตามทะเบียนบ้าน)		
รหัสไปรษณีย์ (ตามทะเบียนบ้าน)		
โทรศัพท์ (ตามทะเบียนบ้าน)		

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานภาพทางการศึกษา	หลักสูตรวิชาเรียนทั่วไป	รายละเอียดของหลักสูตรวิชา
ชั้นปีที่ เลขประจำตัวนักศึกษา คณะ ภาควิชา สาขาวิชาเอก สาขาวิชาโท หลักสูตร ชื่อคณะ ชื่อภาควิชา ชื่อสาขาวิชาเอก ชื่อสาขาวิชาโท ชื่อหลักสูตร รหัสอาจารย์ที่ปรึกษา ชื่ออาจารย์ นามสกุลอาจารย์	รหัสวิชา ชื่อวิชา(ไทย) ชื่อวิชา(อังกฤษ) ชั้นปีที่เรียน คณะ ภาควิชา รายละเอียดวิชา(ไทย) รายละเอียดวิชา(อังกฤษ) หมวดวิชา วิชาบังคับผ่าน	รหัสหลักสูตร ชื่อหลักสูตร(ไทย) ชื่อหลักสูตร(อังกฤษ) อักษรย่อ(ไทย) อักษรย่อ(อังกฤษ) รหัสคณะ ชื่อคณะ รหัสภาควิชา ชื่อภาควิชา จำนวนหน่วยกิตรวม วิชาในหมวดวิชาต่างๆ โครงสร้างหลักสูตร

<b>คณะ</b> รหัสคณะ ชื่อคณะ(ไทย) ชื่อคณะ(อังกฤษ)	<b>ภาควิชา</b> รหัสภาควิชา รหัสคณะ ชื่อภาควิชา(ไทย) ชื่อภาควิชา(อังกฤษ)	<b>สัญชาติ</b> รหัส ความหมาย	<b>เชื้อชาติ</b> รหัส ความหมาย
<b>ศาสนา</b> รหัส ความหมาย	<b>ประเภทการรับเข้า</b> รหัสการรับเข้า ชื่อการรับเข้า ประเภทการรับเข้า	<b>เขต</b> รหัสเขต ชื่อเขต ชื่อจังหวัด	<b>จังหวัด</b> รหัสจังหวัด ชื่อจังหวัด

รูปที่ 3.14 แสดงเอนิตีหลักๆ ที่เป็นออบเจกต์ในระบบ

### 3.10 การออกแบบส่วนออบเจ็กต์อินเทอร์เน็ต ของบิสซิเนส เซอร์วิส

เมื่อได้ออกแบบออบเจ็กต์ต่างๆของระบบแล้ว เราก็จะได้บิสซิเนสออบเจ็กต์ที่จะดูแลแต่ละหน้าที่หลักๆ ดังนี้

StdMgr	ดูแลส่วนประวัตินักศึกษา
StdMgr2	ดูแลประวัตินักศึกษาส่วนประวัติการศึกษา
ParMgr	ดูแลประวัตินักศึกษาส่วนผู้ปกครอง
CurricuMgr	ดูแลหลักสูตรการศึกษา
CurriSbMgr	ดูแลวิชาในแต่ละหลักสูตร
SubjMgr	ดูแลวิชาเรียน
SubGpMgr	ดูแลหมวดวิชา
DepMgr	ดูแลภาควิชา
FacMgr	ดูแลคณะ

นอกจากนี้ยังมีส่วนบิสซิเนสออบเจ็กต์ที่จะดูแลส่วนย่อยๆอีกมากมาย เช่น

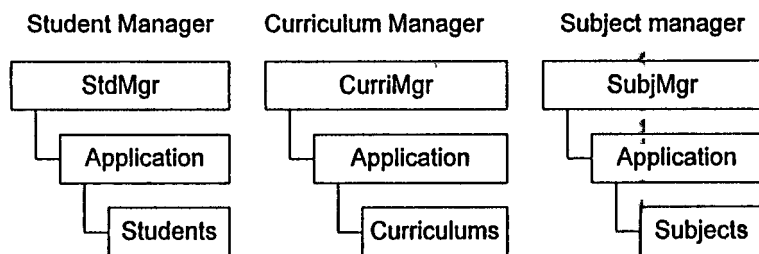
CampMgr	ดูแลมหาวิทยาลัยต่างสาขา
BorderMgr	ดูแลอำเภอ
ProvMgr	ดูแลจังหวัด
ReliMgr	ดูแลศาสนา
DadMgr	ดูแลบิดา
MomMgr	ดูแลมารดา

ฯลฯ

ซึ่งนับว่าเป็นส่วนที่เล็กมากในระบบจึงไม่ขอกล่าวถึง

ขั้นต่อไปคือการสร้างออบเจ็กต์โมเดลแบบลำดับชั้น(Object Hierarchy Model) ซึ่ง

มีตัวอย่างดังนี้คือ



รูปที่ 3.15 แสดงออบเจ็กต์โมเดลแบบลำดับชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเราก็จะทำการออกแบบส่วนออบเจ็กต์อินเทอร์เน็ตเฟส ซึ่งในที่นี้จะขอยก  
ออบเจ็กต์อินเทอร์เน็ตเฟสบางตัวให้เป็นตัวอย่างดังนี้

### SubjMgr

	Application	Subjects	Subject
<b>Collections:</b>	Subjects		
<b>Member objects:</b>	Item		
<b>Properties</b>	Id	Count	รหัสวิชา (Subj_Id)
	UserId		ชื่อวิชา(ไทย) (Subj_Name_Thai)
	Password		ชื่อวิชา(อังกฤษ) (Subj_Name_Eng)
			ชั้นปีที่เรียน (Year_study)
			คณะ (Faculty_Id)
			ภาควิชา (Subj_Dept_Id)
			รายละเอียดวิชา (ไทย)(Subj_Descrip_Thai)
			รายละเอียดวิชา(อังกฤษ) (Subj_Descrip_Eng)
			หมวดวิชา (SubGroup_Type_Id)
			วิชาบังคับผ่าน (Pre_Subj_Id)
<b>Methods:</b>	Initialize	LoadSubjs	
		UpdateSubjs	
		Add	
		Remove	
<b>( Performance tuning )</b>		CollectionGet	PropertyGet
<b>:</b>		CollectionSet	PropertySet
<b>Data tier:</b>	sp_get_subj	sp_add_subj	
		sp_delete_subj	
		sp_update_subj	

ตาราง 3.1 แสดงออบเจ็กต์อินเทอร์เน็ตเฟส

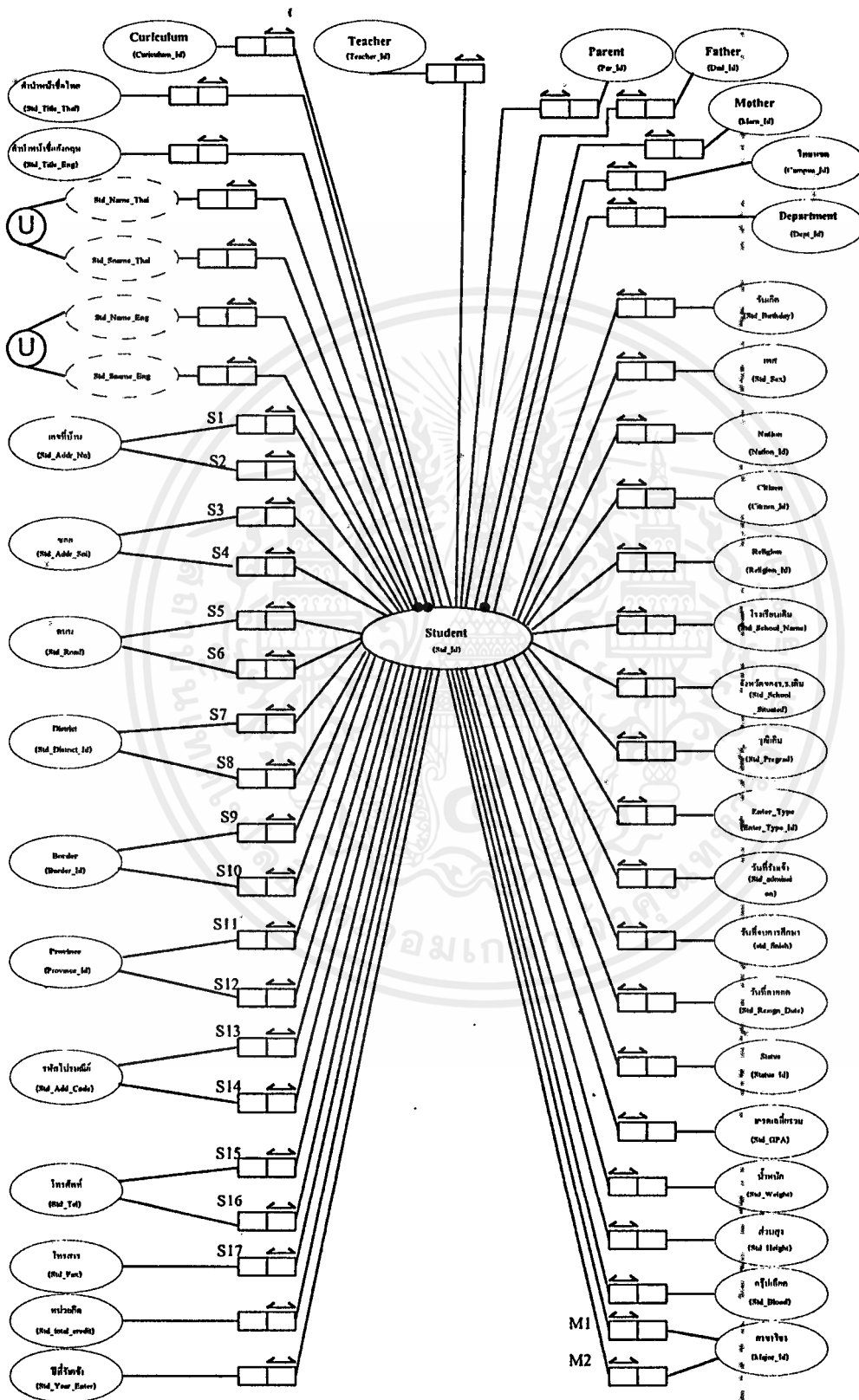
### 3.11 การออกแบบส่วนดาต้า เซอร์วิส

การออกแบบในส่วนนี้ ก็คือการออกแบบตารางในการจัดเก็บฐานข้อมูลนั่นเอง ซึ่งการออกแบบสามารถทำได้โดยวิธีใดวิธีหนึ่ง เช่น เอนติตี้ รีเลชันเนล โมเดล, ในแอมู เป็นต้น

จะขอยกตัวอย่างการออกแบบจากฐานข้อมูลที่มีอยู่แล้วแต่เดิมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพ NIAM ของระบบงานทะเบียนประวัตินิติคดี



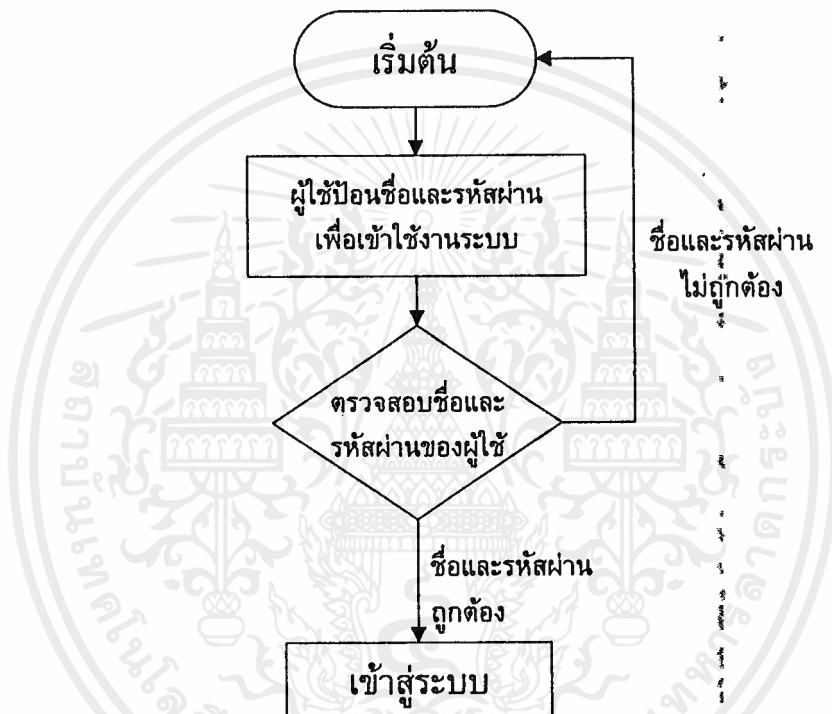
รูปที่ 3.16 แสดงแผนภาพไนแอมของประวัตินักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.12 การติดต่อกับผู้ใช้

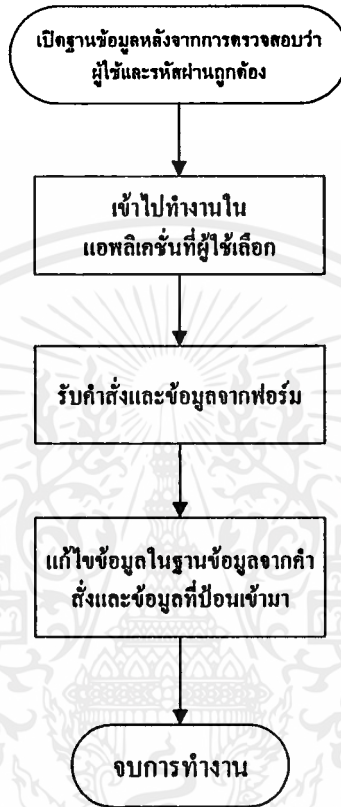
#### 3.12.1 หน้าจอป้อนรหัสผ่านเพื่อติดต่อกับระบบ

##### การทำงานของโปรแกรม



รูปที่ 3.17 แสดง โฟลว์ชาร์ตของหน้าจอป้อนรหัสผ่าน

### 3.12.2 หน้าจอในการทำงานส่วนต่างๆ



รูปที่ 3.18 แสดงโฟลว์ชาร์ตของหน้าจอเลือกประเภทการทำงาน

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการทดลอง

##### 4.1.1 คอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบเครือข่าย (Server)

1. มีหน่วยประมวลผลกลาง เพนเทียม 75 เมกะเฮิร์ต
2. มีหน่วยความจำหลัก ขนาดไม่ต่ำกว่า 32 เมกะไบต์
3. มีหน่วยความจำชนิดความเร็วสูงภายในหน่วยประมวลผลกลางขนาดไม่น้อยกว่า 16 กิโลไบต์และภายนอกประมวลผลกลางขนาดไม่น้อยกว่า 256 กิโลไบต์
4. ระบบขับจานแม่เหล็ก โดยใช้มาตรฐานการเชื่อมต่อ (Interface) แบบ SCSI-II ความจุ 2 กิกะไบต์ (formatted)
5. เครื่องอ่านซีดีรอม (CD-ROM) มีความจุ 600 เมกะไบต์ โดยใช้มาตรฐานการเชื่อมต่อ SCSI-II จำนวน 1 ชุด
6. ฟลอปปี ดิสก์ไดรฟ์ (Floppy Disk Drive) ขนาด 3.5 นิ้ว ความจุไม่ต่ำกว่า 1.44 เมกะไบต์ จำนวน 1 หน่วย
7. จอภาพขาวดำ ขนาด 14 นิ้ว ความละเอียด 600 x 480 พิกเซล (pixel)
8. คีย์บอร์ด ที่มีอักษรภาษาไทย/อังกฤษ ตัวเลข และเครื่องหมายสัญลักษณ์พิเศษอย่างน้อย 101 คีย์
9. มีอุปกรณ์ป้อนคำสั่งแบบเมาส์ (Mouse) ที่มี 2 ปุ่ม จำนวน 1 ชุด

##### 4.1.2 เครื่องไมโครคอมพิวเตอร์ที่ทำหน้าที่เป็นเทอร์มินอล (Terminal) ในระบบเครือข่าย จำนวน 2 เครื่อง

1. หน่วยประมวลผลกลาง 486DX-66
2. มีหน่วยความจำหลัก ขนาด 16 เมกะไบต์ 1 เครื่อง และ 12 เมกะไบต์ 1 เครื่อง
3. ระบบขับจานแม่เหล็ก โดยใช้มาตรฐานการเชื่อมต่อแบบ IDE มีความจุ 420 เมกะไบต์
4. ฟลอปปี ดิสก์ไดรฟ์ ขนาด 3.5 นิ้ว ความจุไม่ต่ำกว่า 1.44 เมกะไบต์ จำนวน 1 หน่วย
5. จอภาพสีขนาด 14 นิ้ววัดตามแนวทแยงมุม แสดงภาพได้ที่มีความละเอียด 600 x 480 พิกเซล 256 สีแบบ นอน อินเตอร์เฟส (Non-interlace)
6. คีย์บอร์ดที่มีอักษรภาษาไทย/อังกฤษ ตัวเลข และเครื่องหมายสัญลักษณ์พิเศษอย่างน้อย 101 คีย์

7. มีอุปกรณ์ป้อนคำสั่งแบบเมาส์ ที่มี 2 ปุ่ม จำนวน 1 ชุด
8. เครื่องพิมพ์ EPSON LQ 570 จำนวน 1 เครื่อง

#### 4.2 ความต้องการทางด้านซอฟต์แวร์

1. ในส่วนบริการผู้ใช้ (User Service) ใช้ วิชาเว็บลิกเป็นตัวพัฒนา (Developer)
2. ในส่วนบริการกฎเกณฑ์ระบบ (Business Service) ใช้วิชาเว็บลิกเป็นตัวพัฒนา (Developer)
3. ในส่วนบริการข้อมูล (Data Service) จะใช้ เอสคิวแอล เซิร์ฟเวอร์ (SQL Server) จัดการงานในส่วนฐานข้อมูล

##### 4.2.1 ระบบปฏิบัติการที่ทำหน้าที่เป็นเซิร์ฟเวอร์

1. ระบบปฏิบัติการ Microsoft Windows NT version 3.51
2. โปรแกรมระบบจัดการฐานข้อมูล Microsoft SQL Server version 6.5
3. โอเอส ออโตเมชัน เซิร์ฟเวอร์

##### 4.2.2 ระบบปฏิบัติการที่ทำหน้าที่เป็นเวิร์คสเตชัน

1. ระบบปฏิบัติการ Windows 95 รุ่นภาษาไทย
2. MicroSoft Visual Basic version 4.0 Enterprise Edition

#### 4.3 การทดลองและผลจากการทดลอง

##### 4.3.1 ระบบการนำข้อมูลเข้าของประวัตินักศึกษาและข้อมูลโครงสร้างการศึกษา

###### ข้อมูลนำเข้า

- ข้อมูลประวัติการศึกษาและส่วนตัวของนักศึกษา, ผู้ปกครอง, อาจารย์
- ข้อมูลหลักสูตรการศึกษา

###### ผลการทดลอง

- จากหน้าจอการจัดการเกี่ยวกับข้อมูลประเภทต่างๆ ที่ได้ออกแบบไว้ทำให้การป้อนข้อมูลเข้าสู่ระบบสามารถทำได้หลายวิธี ไม่ว่าจะเป็นการพิมพ์ตัวข้อมูลเข้าสู่ระบบ การเลือกข้อมูลจากรายชื่อที่มีให้แล้ว หรือเป็นการเลือกทางเลือกที่กำหนดไว้ โดยที่เมื่อได้ทำการป้อนข้อมูลเข้าไปที่ช่องต่างๆ แล้วพร้อมกับการยืนยันที่ปุ่มตกลง ข้อมูลทั้งหมดจะถูกบันทึกลงไปที่ตำแหน่งของข้อมูล นอกจากนี้การมีโหมดการทำงานได้แก่ การลบข้อมูล การเพิ่มข้อมูล การแก้ไขข้อมูล สามารถทำให้การใช้ข้อมูลพร้อมๆ กันในหลายการทำงานเป็นไปได้สะดวกขึ้นเพราะไม่ต้องการการล็อกของข้อมูลที่ไม่จำเป็น

เริ่มจากการเข้าระบบโดยผ่านหน้าจอล็อกอิน โดยมีหน้าจอดังนี้

รหัสผ่านเข้าสู่ระบบ

โปรดใส่ชื่อและรหัสผ่านเข้าสู่ระบบ

ชื่อผู้ใช้งาน :

รหัสผ่าน :

ตกลง

ยกเลิก

รูปที่ 4.1 หน้าจอเข้าสู่ระบบ

ระบบทะเบียนประวัติ

ยินดีต้อนรับสู่ระบบงานการศึกษา  
กรุณาเลือกกรายการที่ท่านต้องการ

ประวัติส่วนตัว

ประวัติการศึกษา

ออกจากโปรแกรม

วันที่ 8/4/2540

01:07 NUM CAPS INS

รูปที่ 4.2 หน้าจอเลือกประเภทการทำงานของประวัตินักศึกษา

การทดลอง

- เลือกการทำงานจากหน้าจอ

ประวัติส่วนตัว

ประวัติการศึกษา

และ ออกจากการทำงาน

ผลการทดลอง

- สามารถเข้าสู่ระบบทะเบียนประวัติได้ ทั้งประวัติส่วนตัวและประวัติการศึกษา

พร้อมทั้งส่วนประกอบบนหน้าจอเลือกการทำงาน เช่น นาฬิกา สามารถทำงานได้จริง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติสภ

วันที่ 8/4/2540  
Userid

โปรแกรมสำหรับสำนักทะเบียนกลางและประมวลผล

ประวัติส่วนตัว

รหัสประจำตัวนักศึกษา:   เพศชาย  เพศหญิง

นาย  ชื่อไทย:  นามสกุลไทย:

Mr.  ชื่ออังกฤษ:  นามสกุลอังกฤษ:

เชื้อชาติ  สัญชาติ  ศาสนา

วันเกิด  /  /  กลุ่มเลือด  ส่วนสูง  น้ำหนัก

ที่อยู่ปัจจุบัน

บ้านเลขที่  ตรอก/ซอย  ถนน

แขวง/ตำบล  เขต/อำเภอ  จังหวัด

รหัสไปรษณีย์  เบอร์โทรศัพท์

ที่อยู่ตามทะเบียนบ้าน

บ้านเลขที่  ตรอก/ซอย  ถนน

แขวง/ตำบล  เขต/อำเภอ  จังหวัด

รหัสไปรษณีย์  เบอร์โทรศัพท์

ประวัติการศึกษา

ผู้ปกครอง

สถานภาพศึกษา

เพิ่มข้อมูล

ลบข้อมูล

แก้ไขข้อมูล

ตรวจสอบข้อมูล

ยกเลิก

กลับสู่รายการหลัก

รูปที่ 4.3 หน้าจอประวัตินักศึกษาในส่วนประวัติส่วนตัว

#### การทดลอง

- ใส่อข้อมูลตามที่ปรากฏในส่วนของหน้าจอ
- เลือกประเภทการทำงาน ว่าต้องการจะทำงานในส่วนใดต่อไป

#### ผลการทดลอง

- พบปัญหาเรื่องตัวแปรคนละชนิด แต่ได้แก้ไขเรียบร้อยแล้ว
- ความเร็วในการทำงานอยู่ในระดับที่เหมาะสม
- ความเร็วในการโหลดตอนเริ่มต้นของโปรแกรม อยู่ในระดับพอใช้ ไม่ช้ามาก

(ความช้าเกิดจากสาเหตุที่โปรแกรมต้องไปโหลดข้อมูลจากฐานข้อมูลมาเก็บไว้ก่อน)

ประวัติส่วนบุคคล

โปรแกรมสำหรับสำนักทะเบียนกลางและประมวลผล

วันที่ 8/4/2540  
Userid

ประวัติการศึกษา

วุฒิเดิม(ก่อนเข้าศึกษา)

โรงเรียน

จังหวัด  คณะแผนกเฉลี่ย

ประเภทการรับเข้า

สอบผ่านทบวงมหาวิทยาลัย

โควต้าโครงการส่งเสริมการศึกษาต่อในมหาวิทยาลัยเกษตรศาสตร์

สถาบันสมทบ

ทุนการศึกษารัฐบาลระดับปริญญาตรี นิสิตต่างประเทศ

รับโอนจากสถาบันการศึกษาอื่น

วัน/เดือน/ปี ที่เข้ารับการศึกษา  /  /

วัน/เดือน/ปี ที่จบการศึกษา  /  /

ประวัติส่วนตัว

ผู้ปกครอง

สถานภาพศึกษา

รหัส

ตรวจสอบ

แก้ไข

ยกเลิก

กลับสู่รายการหลัก

รูปที่ 4.4 หน้าจอประวัตินักศึกษาในส่วนประวัติการศึกษา

## การทดลอง

- ใส่ข้อมูลตามที่ปรากฏในส่วนของหน้าจอ
- เลือกประเภทการทำงาน ว่าต้องการจะทำงานในส่วนใดต่อไป

## ผลการทดลอง

- ความเร็วในการทำงานอยู่ในระดับที่ค่อนข้างเร็วเพราะรายละเอียดน้อย
- ความเร็วในการโหลดตอนเริ่มต้นของฟอร์ม อยู่ในระดับพอใช้

สถานภาพทางการศึกษา

โปรแกรมสำนักทะเบียนกลางและประมวลผล วันที่ 8/4/2540

ชั้นปีที่  รหัสประจำตัวนักศึกษา

คณะ

ภาควิชา

สาขาวิชาเอก

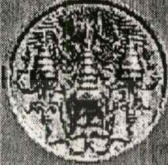
สาขาวิชาโท

หลักสูตร

รหัสอาจารย์ที่ปรึกษา

ชื่อไทย

นามสกุลไทย



ประวัติส่วนตัว

ประวัติการศึกษา

ผู้ปกครอง

ตกลง

ยกเลิก

กลับสู่รายการหลัก

#### รูปที่ 4.5 หน้าจอประวัตินักศึกษาในส่วนของสถานภาพทางการศึกษา

##### การทดลอง :

- ใส่ข้อมูลตามที่ปรากฏในส่วนของหน้าจอ
- เลือกประเภทการทำงาน ว่าต้องการจะทำงานในส่วนใดต่อไป

##### ผลการทดลอง

- ความเร็วในการทำงานอยู่ในระดับพอใช้ ความล่าช้ามีสาเหตุมาจากการที่ฟอร์มนี้ทำงานกับการเรียกใช้ข้อมูลจากหลายๆตาราง
- ความเร็วในการโหลดตอนเริ่มต้นของฟอร์ม อยู่ในระดับพอใช้

รายละเอียดผู้ปกครอง

โปรแกรมสำนักทะเบียนกลางและประมวลผล วันที่ 8/4/2540

ผู้ปกครอง

ชื่อผู้ปกครอง  อาชีพ/ตำแหน่ง

ความสัมพันธ์  โทรศัพท์ที่ทำงาน

สถานที่ทำงาน  โทรสาร

ที่อยู่

รหัสไปรษณีย์  โทรศัพท์ที่บ้าน

บิดามารดา

ชื่อบิดา  อาชีพ/ตำแหน่ง

ที่อยู่  โทรศัพท์ที่ทำงาน

สถานที่ทำงาน  โทรศัพท์ที่บ้าน

ชื่อมารดา  อาชีพ/ตำแหน่ง

ที่อยู่  โทรศัพท์ที่ทำงาน

สถานที่ทำงาน  โทรศัพท์ที่บ้าน

ประวัติส่วนตัว

ประวัติการศึกษา

สถานภาพการศึกษา

รหัส

เพิ่มข้อมูล

ลบข้อมูล

แก้ไขข้อมูล

ตรวจสอบ

ยกเลิก

กลับสู aray การหลัก

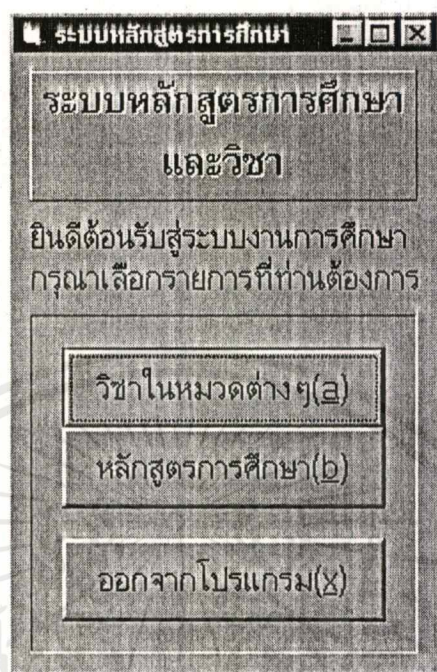
รูปที่ 4.6 หน้าจอประวัตินักศึกษาในส่วนประวัติผู้ปกครอง

**การทดลอง**

- ใส่ข้อมูลตามที่ปรากฏในส่วนของหน้าจอ
- เลือกประเภทการทำงาน ว่าต้องการจะทำงานในส่วนใดต่อไป

**ผลการทดลอง**

- ความเร็วในการทำงานอยู่ในระดับพอใช้ เนื่องจากต้องใช้ข้อมูลจาก 2 ตารางจึงอาจจะใช้เวลาไปนิดหน่อย แต่ไม่อาจเรียกว่าช้าได้
- ความเร็วในการโหลดตอนเริ่มต้นของฟอร์ม อยู่ในระดับพอใช้



รูปที่ 4.7 หน้าจอเลือกประเภทการทำงานของหลักสูตรการศึกษา

#### การทดลอง

- เลือกการทำงานจากหน้าจอ  
วิชาในหมวดต่างๆ  
หลักสูตรการศึกษา  
และ ออกจากการทำงาน

#### ผลการทดลอง

- สามารถเข้าสู่ระบบหลักสูตรการศึกษาได้ ทั้งส่วนวิชาเรียนและหลักสูตรการ

ศึกษา

หลักสูตรการศึกษา

โปรแกรมสำหรับทะเบียนกลางและประมวลผล

วันที่ 8/4/2540  
Userid

เพิ่มข้อมูลหลักสูตร

รหัสของหลักสูตร

ชื่อหลักสูตร(ไทย)

ชื่อหลักสูตร(อังกฤษ)

อักษรย่อชื่อหลักสูตร(ไทย)

อักษรย่อชื่อหลักสูตร(อังกฤษ)

รหัสคณะ

รหัสภาควิชา

ศึกษาทั่วไป

วิชาเฉพาะ

เลือกเสรี

ฝึกงาน

โครงสร้างหลักสูตร

เพิ่มหลักสูตร

แก้ไขหลักสูตร

ลบหลักสูตร

ตรวจสอบ

ตกลง

ยกเลิก

กลับสู่รายการหลัก

รหัสวิชา	ชื่อวิชา	กลุ่มวิชา	จำนวนหน่วยกิตภาค...	จำนวนหน่วยกิตภาค...

จำนวนหน่วยกิตรวมของหลักสูตร

#### รูปที่ 4.8 หน้าจอแสดงรายละเอียดของหลักสูตรการศึกษา

##### การทดลอง

- ใส่ข้อมูลตามที่ปรากฏในส่วนของหน้าจอ
- เลือกประเภทการทำงาน ว่าต้องการจะทำงานในส่วนใดต่อไป

##### ผลการทดลอง

- ส่วนลิสต์วิวยังค่อนข้างช้า
- ความเร็วในส่วนอื่นอยู่ในระดับที่เหมาะสม
- ความเร็วในการโหลดตอนเริ่มต้นของโปรแกรม อยู่ในระดับพอใช้ ไม่ช้ามาก

( ความช้าเกิดจากสาเหตุที่โปรแกรมต้องไปโหลดข้อมูลจากฐานข้อมูลมาเก็บไว้ก่อน )

ระบบการศึกษาระดับปริญญาตรี

โปรแกรมสำหรับลงทะเบียนกลางและประมวลผล

วันที่ 8/4/2540  
Userid

รายละเอียดของวิชา

รหัสวิชา  ชั้นปีที่เรียน

ชื่อวิชา(ไทย)  ปีที่เริ่มใช้

ชื่อวิชา(อังกฤษ)  ปีที่เลิกใช้

คณะ  ภาควิชา

รายละเอียด

Course Description

หมวดวิชา

ศึกษาทั่วไป

วิชาเฉพาะ

การฝึกงาน

เลือกเสรี

หน่วยกิตบรรยาย  หน่วยกิตปฏิบัติ

ภาคการศึกษาที่เริ่มใช้

ภาคการศึกษาที่เลิกใช้

กลุ่มวิชา

ความหมาย

เพิ่มวิชา

แก้ไขวิชา

ลบวิชา

ตรวจสอบ

ตกลง

ยกเลิก

กลับสู่รายการหลัก

รูปที่ 4.9 หน้าจอแสดงข้อมูลวิชาเรียนต่างๆไป

**การทดลอง**

- ใส่ข้อมูลตามที่ปรากฏในส่วนของหน้าจอ
- เลือกประเภทการทำงาน ว่าต้องการจะทำงานในส่วนใดต่อไป

**ผลการทดลอง**

- ความเร็วของการทำงานอยู่ในระดับที่เหมาะสม
- ยังไม่มีข้อมูลในส่วนของรายละเอียดวิชาในฐานข้อมูล จึงดึงข้อมูลออกมาแสดงผลไม่ได้

ผลไม่ได้

แอป-ลบ วิชาในหลักสูตร

วันที่ 8/4/2540

## โปรแกรมสำหรับลงทะเบียนกลางและประมวลผล

เพิ่ม-ลบ วิชาในหลักสูตร

รหัสหลักสูตร

รหัสวิชา

ชื่อวิชา

กลุ่มวิชา

ชั้นปีที่เรียน

ภาคการศึกษา

หมวดวิชาทั่วไป  
 หมวดวิชาเฉพาะ  
 หมวดวิชาเลือกเสรี  
 หมวดวิชาฝึกงาน

รูปที่ 4.10 หน้าจอการเลือกข้อมูลหลักสูตรวิชาที่ต้องแก้ไข

#### การทดลอง

- ใส่ข้อมูลตามที่ปรากฏในส่วนหน้าจอ
- เลือกประเภทการทำงาน ว่าต้องการจะทำงานในส่วนใดต่อไป

#### ผลการทดลอง

- ความเร็วของการทำงานอยู่ในระดับที่เหมาะสม มีการใช้ข้อมูลจากหลายตาราง

จึงอาจทำให้เสียเวลาบ้าง

## บทที่ 5

### บทวิจารณ์ และสรุป

#### 5.1 ความสามารถของโปรแกรม

- 1.สามารถจัดเก็บ,แก้ไข,เพิ่มเติม,ตรวจสอบ,ลบและพิมพ์ประวัติส่วนตัวนักศึกษา
- 2.สามารถจัดเก็บ,แก้ไข,เพิ่มเติม,ตรวจสอบ,ลบและพิมพ์ประวัติการศึกษานักศึกษา
- 3.สามารถจัดเก็บ,แก้ไข,เพิ่มเติม,ตรวจสอบ,ลบและพิมพ์ประวัติผู้ปกครองของนักศึกษา
- 4.สามารถจัดเก็บ,แก้ไข,เพิ่มเติม,ตรวจสอบ,ลบและพิมพ์ประวัตินักศึกษาในส่วนของสถานภาพทางการศึกษา
- 5.สามารถจัดเก็บ,แก้ไข,เพิ่มเติม,ตรวจสอบ,ลบและพิมพ์หลักสูตรการศึกษา
- 6.ข้อมูลต่างๆจะแสดงในรูปแบบกราฟิก ( Graphic ) ซึ่งใช้งานได้สะดวกกว่าเท็กซ์โหมด (Text Mode)
- 7.ระบบทำงานบน ระบบเครือข่าย( Network ) ซึ่งผู้ที่มีสิทธิในการเข้าถึงข้อมูลได้สามารถใช้โปรแกรมนี้เรียกใช้ข้อมูลได้ ทำให้ลดภาระให้แก่สำนักทะเบียนและประมวลผล รวมถึงความรวดเร็วในการรับข้อมูลมากขึ้นด้วย

#### 5.2 ข้อจำกัดของโปรแกรม

- 1.เนื่องจากระบบเป็นโครงสร้างแบบทรีเทียร์ซึ่งเป็นสถาปัตยกรรมแบบใหม่ ซึ่งมีความใหม่ประกอบกับเวลาที่จำกัดซึ่งอาจทำให้ศึกษาขั้นตอนการทำงาน รวมไปถึงหลักการต่างๆ อาจคลาดเคลื่อนไปบ้าง
- 2.เออร์เรอร์เมสเสจ (Error massage) นั้นยังไม่มีการจัดการที่ดีพอ บางครั้งเออร์เรอร์เมสเสจที่บอกแก่ผู้ใช้ก็ดูเหมือนจะไม่มีประโยชน์เพราะเป็นการบอกข้อผิดพลาดโดยรวมเท่านั้น

#### 5.3 ปัญหาในการทำงาน

- 1.เครื่องที่ใช้ทำหน้าที่เป็นผู้ให้บริการ(Server)มีประสิทธิภาพยังไม่น่าเชื่อถือเพราะเกิดปัญหาบ่อยจากระบบฮาร์ดแวร์ของเครื่อง เช่น อยู่ดีๆ เครื่องก็แฮงค์(Hang) และรีบูตตัวเองใหม่อีกทั้งความเร็วและหน่วยความจำของเครื่องควรมีการอัปเดต

- 2.เครื่องที่ใช้พัฒนาควรจะเร็วกว่านี้ เพราะในปัจจุบัน โปรแกรมและระบบปฏิบัติการใหม่ใช้รีซอร์สค่อนข้างเยอะ หากไม่มีการอัปเดตจะทำให้ผู้พัฒนารุ่นต่อไปทำงานค่อนข้างลำบาก เช่น ในการพัฒนาทรีเทียร์ บางคราจะต้องโหลดคิวเวลเบสิกใช้งานพร้อมกันถึง 8-9 ตัว

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หนังสือที่ใช้ค้นคว้ามิน้อย ค้นคว้าค่อนข้างลำบาก อีกทั้งผู้รู้เทคโนโลยีนี้ก็ยังมีน้อย เพราะยังเป็นเทคโนโลยีที่ใหม่ ทำให้เกิดความลำบากในการศึกษาโครงการ ระหว่างที่ศึกษาโครงการนี้ ผู้ทำโครงการต้องติดต่อกับปรึกษาเรื่องนี้กับผู้รู้ในต่างประเทศบ่อยมาก และแต่ละท่านต่างก็มองเทคโนโลยีในเรื่องนี้ไปกันคนละคอนเซ็ปต์ ทำให้ต้องเกิดความลังเลในการตัดสินใจ
3. ข้อมูลในฐานข้อมูลมีน้อย ทำให้การทดลองลำบากเพราะต้องอาศัยข้อมูลมาก
4. ระบบบางอย่างในการเก็บข้อมูลยังไม่มี ความชัดเจน เช่น วิชาเรียนก่อน-หลัง

#### 5.4 แนวทางการพัฒนาต่อไป

1. เนื่องจากความเร็วของระบบยังช้าอยู่ ซึ่งเมื่อเป็น โครงสร้างแบบทรีทีเรียร์น่าจะเร็วขึ้น อาจเป็นเพราะยังพัฒนาไม่ดีพอ ดังนั้นควรจะมีการพัฒนาการเขียนโปรแกรมและศึกษาระบบให้มีประสิทธิภาพมากกว่านี้
2. ควรจะมีการพัฒนาในเรื่องของข้อผิดพลาด(Error message) ให้สามารถอธิบายสาเหตุที่เกิดปัญหาให้ละเอียดกว่านี้
3. ควรมีการศึกษาสถาปัตยกรรมแบบทรีทีเรียร์ให้มากยิ่งขึ้น และทำการพัฒนาจนระบบมีประสิทธิภาพมากขึ้น จะพบว่าสถาปัตยกรรมแบบนี้ จะช่วยแก้ปัญหาดังที่กล่าวในบทแรกๆ และช่วยประหยัดในเรื่องของค่าใช้จ่ายที่ต้องซื้อฮาร์ดแวร์ราคาแพงมาใช้
4. ควรมีการศึกษาและทำความเข้าใจระบบใหม่ โดยการไปหาข้อมูลจากของจริง แล้วเขียนรายละเอียดที่ศึกษาใหม่ให้ครบถ้วนเพื่อการพัฒนาในขั้นต่อไปจะสามารถทำได้ง่ายและเข้าใจ และควรที่จะออกแบบระบบฐานข้อมูลใหม่ด้วย

## ภาคผนวก

### ประเภทของข้อมูลใน Visual Basic (Data type in Visual Basic version 4.0)

Data type	Storage size	Range
Byte	1 bytes	0 to 255
Boolean	2 bytes	True or False
Integer	2 bytes	-32,768 to 32,767
Long (long integer)	4 bytes	-2,147,483,648 to 2,147,483,647
Single (single-precision floating-point)	4 bytes	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values.
Double (double-precision floating-point)	8 bytes	-1.79769313486232E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values.
Currency (scaled integer)	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Date	8 bytes	January 1, 100 to December 31, 9999.
Object	4 bytes	Any Object reference.
String (variable-length)	10 bytes + string length	0 to approximately 2 billion (approximately 65,400 for Microsoft Windows version 3.1 and earlier)
String (fixed-length)	Length of string	1 to approximately 65,400.
Variant (with numbers)	16 bytes	Any numeric value up to the range of a Double.
Variant (with numbers)	22 bytes + string length	Same range as for variable-length String.
User-defined (using Type)	Number required by elements	The range of each element is the same as the range of its data type.

## รายละเอียดต่างๆ เกี่ยวกับการใช้ RDO ออบเจ็กต์

### ตารางแสดงคุณสมบัติของ rdoEngine ออบเจ็กต์

คุณสมบัติ	คำอธิบาย
rdoDefaultCursorDriver	ชนิดของเคอร์เซอร์ที่ถูกสร้าง : ODBC ; Server side;Server side if supported,else ODBC
rdoDefaultErrorThreshole	ค่าของคุณสมบัติ ErrorThreshole ซึ่งถูกใช้โดย rdoPrepareStatement ออบเจ็กต์
rdoDefaultLoginTimeout	เป็นเวลา ที่ ODBC รอขณะที่กำลังพยายามทำการติดต่อกับแหล่งข้อมูล(หน่วยเป็นวินาที)
rdoDefaultPassword	พาสเวิร์ดที่ถูกใช้เมื่อทำการสร้าง rdoEnvironment ใหม่ (ถ้าในขณะนั้นยังไม่มีกำหนดพาสเวิร์ด)
rdoDefaultUser	UserId ที่ถูกใช้เมื่อทำการสร้าง rdoEnvironment ใหม่ (ถ้าในขณะนั้นยังไม่มีกำหนด UserId)
rdoVersion	เวอร์ชันของ rdo ไบเบรารี ซึ่งมีขนาด 5 ตัวอักษร

### ตารางแสดงเมธอดของ rdoEngine ออบเจ็กต์

เมธอด	คำอธิบาย
rdoCreateEnvironment	สร้างการติดต่อใหม่ไปยังคาค้า ซอร์ส และ rdoEnvironment ออบเจ็กต์
rdoRegisterDataSource	ทำการรีจิสเตอร์ข้อมูลของคาค้า ซอร์ส ลงในวินโดว รีจิสทรี

### ตารางแสดงพรีอเพอติวีของ rdoEnvironment ออบเจ็กต์

พรีอเพอติวี	คำอธิบาย
Count	จำนวนของ rdoEnvironment ออบเจ็กต์ที่อยู่ในคอลเลกชัน
CursorDriver	ชนิดของเคอร์เซอร์ที่สามารถสร้างได้ : ODBC หรือ Server - side ขึ้นกับ ODBC Driver ที่สนับสนุน
hEnv	การจัดการของ ODBC Environment ซึ่งสามารถผ่านค่าไปยัง ODBC API ได้
LoginTimeout	เวลาในหน่วยวินาที ที่ ODBC Driver manager รอ จนกระทั่งหมดเวลา เมื่อทำการติดต่อกับ

Name	ชื่อที่เกี่ยวข้องกับ rdoEnvironment
Password	พาสเวิร์ดที่ใช้ในการสร้าง rdoEnvironment , เป็นพรอพเพอร์ตี้ที่เขียนได้เพียงครั้งเดียว
UserName	ชื่อผู้ใช้ที่ใช้ในการสร้าง rdoEnvironment , เป็นพรอพเพอร์ตี้ที่เขียนได้เพียงครั้งเดียว

ตารางแสดงเมธอดของ rdoEnvironment ออบเจ็กต์

เมธอด	คำอธิบาย
BeginTrans	เริ่มทรานสแอคชัน
Close	ปิด rdoEnvironment และ ทุก ๆ rdoConnection ที่ได้เปิดไว้ , และทำการ Rolling Back ทรานสแอคชันที่ไม่สมบูรณ์
CommitTrans	จบทรานสแอคชัน , ทำการเก็บค่าต่าง ๆ ที่ถูกเปลี่ยนแปลงไป
Item	ค่าของเมมเบอร์ของ rdoEnvironment คอลเลคชันที่ส่งกลับมา
OpenConnection	เริ่มทำการเชื่อมต่อไปยังดาต้า ซอร์ส
RollbackTrans	ยกเลิกการเปลี่ยนแปลงที่ได้ทำมา

ตารางแสดงพรอพเพอร์ตี้ของ rdoConnection ออบเจ็กต์

พรอพเพอร์ตี้	คำอธิบาย
AsyncCheckInterval	เวลาในหน่วยมิลลิวินาที ที่ RDO รอ ก่อนที่จะตรวจสอบสถานะของการดึงข้อมูล
Connect	เก็บค่าสตริง ( String ) ที่ใช้เมื่อเริ่มการติดต่อ
Count	จำนวนของ rdoConnection ออบเจ็กต์ในคอลเลคชัน
hDbc	ค่า Handle ในการติดต่อของ rdoConnection
Name	ชื่อของดาต้า ซอร์สที่ต้องการติดต่อ
QueryTimeout	เวลาในหน่วยวินาที ที่ ODBC รอ ก่อนที่จะหมดเวลาในการดึงข้อมูล
RowsAffected	จำนวนแถวของข้อมูลที่ถูกดึงออกมา
StillExecuting	บอกว่าขณะนี้กำลังทำการดึงข้อมูลอยู่
Transactions	บอกถึงการสนับสนุนการใช้ทรานสแอคชันใน rdoConnection
Updatable	บอกถึงความสามารถในการเปลี่ยนแปลงค่าในออบเจ็กต์
Version	เวอร์ชันของดาต้า ซอร์ส

ตารางแสดงเมธอดของ rdoConnection ออบเจ็กต์

เมธอด	คำอธิบาย
BeginTrans	เริ่มทรานสแอคชั่น
Cancel	ยกเลิกการดึงข้อมูล
Close	ยกเลิกการติดต่อ
CommitTrans	จบทรานสแอคชั่น , ทำการเก็บค่าต่าง ๆ ที่ถูกเปลี่ยนแปลงไป
CreatePreparedStatement	สร้าง rdoPreparedStatement ตัวใหม่
Execute	การทำงานตามชุดคำสั่งของ SQL
Item	ส่งกลับค่าเมมเบอร์ของ rdoConnection คอลเลคชั่น
OpenResultSet	สร้าง rdoResultSet ตัวใหม่
RollbackTrans	ยกเลิกการเปลี่ยนแปลงที่ได้ทำมา

ตารางแสดงพรอพเพอร์ตี้ของ rdoTable ออบเจ็กต์

พรอพเพอร์ตี้	คำอธิบาย
Count	จำนวนของ rdoTable ออบเจ็กต์ในคอลเลคชั่น
Name	ชื่อของตารางฐานข้อมูล
RowCount	จำนวนแถวของข้อมูลในตาราง
Type	เป็นสตริงซึ่งบอกถึงชนิดของออบเจ็กต์ : Table , View , Alias , ...
Updatable	บอกถึงความสามารถในการเปลี่ยนแปลงค่าในตาราง , ปกติจะไม่อนุญาตให้ทำการเปลี่ยนแปลงได้

ตารางแสดงเมธอดของ rdoTable ออบเจ็กต์

เมธอด	คำอธิบาย
Item	ส่งกลับค่าเมมเบอร์ของ rdoConnection คอลเลคชั่น
OpenResultSet	สร้าง rdoResultSet ตัวใหม่
Refresh	ทำการดึงข้อมูลเดิมนั้นมาใหม่อีกครั้ง

ตารางแสดงพรีอเพอติวของ rdoResultSet ออบเจ็กต์

พรีอเพอติว	คำอธิบาย
AbsolutePosition	เป็นตัวชี้ตำแหน่งของข้อมูล
BOF	บอกถึงแถวแรกของข้อมูลที่ต้องการ
Bookmark	เป็นตัวตั้งค่า หรือส่งกลับค่าแถวของข้อมูลที่ต้องการเข้าถึง
Bookmarkable	เป็นตัวที่บอกว่าสามารถใช้ Bookmark ได้หรือไม่
Count	จำนวนของ rdoResultSet ออบเจ็กต์ในคอลเลกชัน
EOF	บอกถึงแถวสุดท้ายของข้อมูลที่ต้องการ
hStmnt	Handle ของ ODBC สเตจเม้นท์ ( Statement )
LastModified	ส่งกลับค่าแถวที่มีการเปลี่ยนแปลงครั้งหลังสุด
LockEdits	บอกถึงชนิดของการล็อกในระหว่างการเขียน
Name	เป็นสตริงซึ่งเก็บตัวอักษร 256 ตัวแรกของ SQL query
PercentPosition	บอกตำแหน่งของแถวในรูปของเปอร์เซ็นต์
Restartable	บอกถึงการสนับสนุน รีควิรี ( Requery ) เมธอด ของ rdoResultSet
RowCount	จำนวนแถวที่ดึงออกมา
StillExecuting	เป็นตัวบอกว่าการดึงข้อมูลเสร็จเรียบร้อยแล้วหรือยัง
Transactions	บอกถึงการสนับสนุนทรานแซกชันของ rdoResultSet
Type	ชนิดของ rdoResultSet ออบเจ็กต์ที่สร้าง
Updatable	ค่าบูลีนซึ่งบ่งบอกว่าตารางที่ใช้อยู่สามารถแก้ไขได้

ตารางแสดงเมธอดของ rdoResultSet ออบเจ็กต์

เมธอด	คำอธิบาย
AddNew	จัดเตรียมแถวของข้อมูลใหม่
Cancel	ยกเลิก และคืนค่าเดิมให้แก่ rdoResultSet ออบเจ็กต์
CancelUpdate	ยกเลิกการแก้ไข rdoResultSet ด้วยการลบข้อมูลในบัฟเฟอร์ ( Buffer ) ออก
Close	ยกเลิก rdoResultSet ออบเจ็กต์
Delete	ลบแถวข้อมูลปัจจุบันจาก rdoResultSet ออบเจ็กต์
Edit	การทำการเปลี่ยนแปลงค่าในแถวข้อมูลใน rdoResultSet ออบเจ็กต์ โดยการเปลี่ยนแปลงนี้ จะทำในส่วนของบัฟเฟอร์ ซึ่งจะมีผลต่อฐานข้อมูลเมื่อใช้เมธอด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือมีการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ขออนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Update
GetRows	ทำการคัดลอกแถวข้อมูลจาก rdoResultSet ใส่งในตัวแปรอาร์เรย์ 2 มิติ
Item	ส่งกลับค่าที่เฉพาะของ rdoResultSet ออบเจ็กต์ ใน rdoResultSet คอลเลกชัน
MoreResults	ทำการเคลียร์ rdoResultSet เดิม เพื่อเตรียม rdoResultSet ใหม่
Move	เลื่อนตัวชี้แถวสัมพันธ์กับตำแหน่งแถวปัจจุบัน ซึ่งถ้ามีค่าเป็นบวก จะทำการเลื่อนไปข้างหน้า และถ้ามีค่าเป็นลบ จะเลื่อนกลับไปข้างหลัง
MoveFirst	เลื่อนตัวชี้แถวไปยังแถวแรกของ rdoResultSet
MoveLast	เลื่อนตัวชี้แถวไปยังแถวสุดท้ายของ rdoResultSet
MoveNext	เลื่อนตัวชี้แถวไปยังแถวถัดไปของ rdoResultSet
MovePrevious	เลื่อนตัวชี้แถวไปยังแถวก่อนหน้าใน rdoResultSet
Requery	ทำการดึงข้อมูลเดิมขึ้นมาใหม่อีกครั้ง
Update	บันทึกสิ่งที่เปลี่ยนแปลงที่อยู่ในบัฟเฟอร์

ตารางแสดงพารามิเตอร์และเมธอดของ rdoColumn ออบเจ็กต์

พารามิเตอร์และเมธอด	คำอธิบาย
AllowZeroLength	คุณสมบัติที่บ่งบอกว่าคอลัมน์นี้สนับสนุนข้อมูลที่มีค่าเป็นแนว ( Null ) ได้หรือไม่
Attributes	เป็นคุณสมบัติที่บ่งชี้ถึงคุณลักษณะของคอลัมน์ เช่น ขนาดของคอลัมน์ เป็นต้น
ChunkRequired	เป็นคุณสมบัติที่บ่งชี้ว่าต้องการจะใช้เมธอด GetChunk หรือไม่ ในการดึงข้อมูลจากคอลัมน์
Count	จำนวนของ rdoColumn ออบเจ็กต์ ในคอลเลกชัน
Name	เก็บชื่อของคอลัมน์ใน rdoColumn ออบเจ็กต์
OrdinalPosition	ค่าตำแหน่งของคอลัมน์ใน rdoColumn คอลเลกชัน
Required	บอกว่าคอลัมน์ใด ๆ ต้องการข้อมูลที่ไม่มีค่าเป็นแนว
Size	บอกถึงจำนวนตัวอักษรที่มากที่สุดที่สามารถรับได้ สำหรับคอลัมน์ประเภทสตริง แต่ถ้าเป็นนิวมอรัล ( Numeric ) ก็จะบอกถึงจำนวนไบต์ที่มากที่สุดที่คอลัมน์รับได้
SourceColumn	ชื่อของคอลัมน์ในฐานข้อมูล

SourceTable	ชื่อตารางในฐานข้อมูล
Type	ชนิดของข้อมูลในคอลัมน์
Updatable	บอกว่าค่าในคอลัมน์สามารถเปลี่ยนแปลงได้
Value	แสดงข้อมูลปัจจุบันในคอลัมน์
AppendChunk	เป็นเมธอดที่ทำการเพิ่มข้อมูลลงในคอลัมน์
ColumnSize	เป็นเมธอดที่ส่งค่าจำนวนไบต์ใน rdoColumn
GetChunk	เป็นเมธอดที่ทำการส่งค่าใน rdoColumn ออบเจ็กต์

ตารางแสดงพารามิเตอร์ของ rdoPreparedStatement ออบเจ็กต์

พารามิเตอร์	คำอธิบาย
BrdThreshold	แสดงถึงคอลัมน์ที่มีขนาดใหญ่ที่สุด ซึ่งจะถูกกำหนดขอบเขตโดย ODBC
Connect	สตริงข้อมูลที่เก็บข้อมูลการเชื่อมต่อกับฐานข้อมูล
Count	จำนวนของ rdoPreparedStatement ออบเจ็กต์ ที่มีอยู่ใน rdoPreparedStatement คอลเลกชัน
ErrorThreshold	กำหนดระดับความรุนแรงของข้อผิดพลาด
hStmt	Handle ของ ODBC สเตจเมนต์
KeysetSize	เจาะจงจำนวนแถวข้อมูลในคีย์เซตบัฟเฟอร์ ( Key set buffer )
LockType	ชนิดของรีซัลท์เซตที่กำลังถูกใช้อยู่
LogMessages	ตำแหน่งของ ODBC เทรซไฟล์ ( Trace File ) ที่ถูกใช้โดย ODBC ไดรฟ์เวอร์ เพื่อทำการบันทึกการกระทำที่เกิดขึ้น
MaxRows	จำนวนแถวข้อมูลที่มากที่สุด ที่ได้จากการคิวรี หากมีค่าเป็น -1 จะไม่จำกัดจำนวนแถวของข้อมูล
Name	ชื่อของพรีเพร์สเตจเมนต์ ( Prepare Statement )
QueryTimeout	เวลาในหน่วยวินาที ที่ ODBC รอ ก่อนการเจนเนอเรท ( generate ) ไทม์เอาต์เออเรอร์ ( Timeout Error ) ขณะที่กำลังรอให้การคิวรีนั้นเสร็จสมบูรณ์
RowsAffected	จำนวนแถวข้อมูลที่เกิดจากการใช้เมธอดเอกซิคิวทีฟครั้งล่าสุด
RowsetSize	จำนวนแถวข้อมูลที่เกิดจากการใช้คีย์เซตเคอร์เซอร์
SQL	ที่เก็บ SQL สเตจเมนต์ที่ถูกใช้ในการกำหนดชุดคำสั่งในการคิวรี ที่ถูกเอกซิคิวทีฟไปแล้ว

StillExecuting	เป็นตัวบอกว่าการดึงข้อมูลเสร็จเรียบร้อยแล้วหรือยัง
Type	ชนิดของการคิวรี
Updatable	บอกถึงการแก้ไขข้อมูลในรีซอลเซต ว่าสามารถทำได้หรือไม่

ตารางแสดงเมธอดของ rdoPreparedStatement ออบเจ็กต์

เมธอด	คำอธิบาย
Cancel	ยกเลิกการคิวรี
Close	การลบ rdoPreparedStatement ออบเจ็กต์ ออกจากคอลเลกชัน
Execute	การทำงานตามชุดคำสั่ง
Item	ทำการส่งค่า rdoPreparedStatement ออบเจ็กต์ที่เจาะจง จากคอลเลกชัน
OpenResultSet	ทำการสร้าง rdoResultSet ออบเจ็กต์ขึ้นมาใหม่ เพื่อทำการเพิ่มออบเจ็กต์ให้กับ rdoResultSet คอลเลกชัน

ตารางแสดงพารามิเตอร์ของ rdoParameter ออบเจ็กต์

พารามิเตอร์	คำอธิบาย
Count	จำนวนของ rdoParameter ออบเจ็กต์ ภายใน rdoParameter คอลเลกชัน
Direction	บ่งบอกวิธีที่พารามิเตอร์ถูกใช้โดยการคิวรี ในทิศทางกร ให้ - รับ ค่า
Name	ชื่อของพารามิเตอร์
Type	ชนิดของข้อมูลในพารามิเตอร์
Value	พารามิเตอร์ที่ทำการเซตค่าเริ่มต้น ในกับพารามิเตอร์

ตารางแสดงพารามิเตอร์และเมธอดของ rdoError ออบเจ็กต์

พารามิเตอร์และเมธอด	คำอธิบาย
Count	แสดงจำนวนของข้อผิดพลาดที่เกิดขึ้น ใน rdoError คอลเลกชัน
Description	เก็บคำข้อความที่บรรยายถึงข้อผิดพลาด
HelpContext	แสดงถึงค่า คอนเทนต์ไอดี ในเฮลป์ไฟล์ ( Help File ) ในการบอกรายละเอียดของข้อผิดพลาด
HelpFile	ค่าที่เก็บตำแหน่งของเฮลป์ไฟล์
Number	แสดงถึง เออเรอร์นัมเบอร์ ( error number )

Source	บอกถึงตำแหน่งที่เกิดข้อผิดพลาด
SQLRetCode	บอกสถานะว่ามีข้อผิดพลาดของการเอกซิกิวท์ ( Execute ) ชุดคำสั่ง SQL หรือไม่
SQLState	บอกสถานะข้อผิดพลาดของการเอกซิกิวท์ ( Execute ) ชุดคำสั่ง SQL

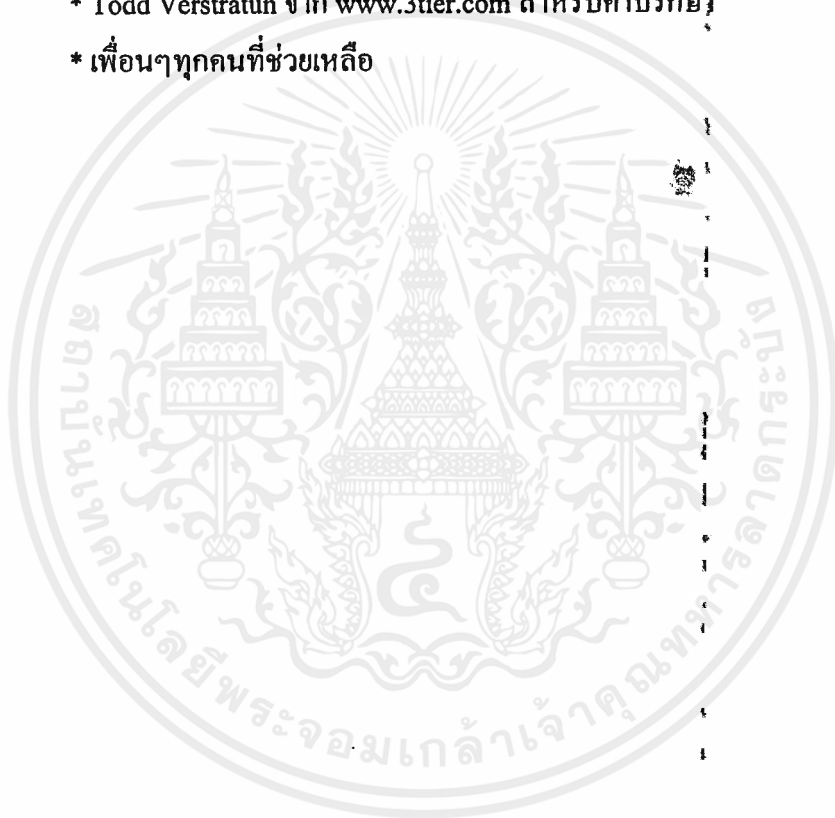
ตารางแสดงพรอพเพอร์ตี้และเมธอดของ rdoError ออบเจ็กต์	
พรอพเพอร์ตี้และเมธอด	คำอธิบาย
Clear	เมธอดที่ทำการเคลียร์ rdoError คอลเลกชัน
Item	ทำการส่งค่า rdoError ออบเจ็กต์ที่เจาะจง จากคอลเลกชัน



## กิตติกรรมประกาศ

วิทยานิพนธ์นี้ได้รับความช่วยเหลือจากบุคคลต่าง ๆ มากมาย จนสำเร็จลุล่วงได้ด้วยดีทางคณะผู้จัดทำจึงขอขอบพระคุณบุคคลต่างๆ เหล่านี้

- \* บุพการีที่ส่งเสริมให้ลูกหลานได้เรียนจนสำเร็จ
- \* อาจารย์วิบูลย์ พร้อมพานิชย์ ในการให้คำปรึกษาตลอดช่วงระยะเวลาการทำงาน
- \* Todd Verstratun จาก [www.3tier.com](http://www.3tier.com) สำหรับคำปรึกษา
- \* เพื่อนๆ ทุกคนที่ช่วยเหลือ



## หนังสืออ้างอิง

1. ราบินเคอร์ ศรีกิจจาภรณ์, "คู่มือการใช้งาน Visual Basic สำหรับวินโดวส์", ซีเอ็ด  
ยูเคชั่น, 407 หน้า, 2538
2. วิสาร กำจรเวทย์, "Visual Basic ฉบับ Database", โปรวิชั่น, 196 หน้า, 2539
3. Craig Goren, "Visual Basic 4 Enterprise", QUE, 974 p., 1996
4. Microsoft, "Microsoft Visual Basic", Microsoft Press, 1064 p.,

