

ระบบสารสนเทศเพื่อความเป็นทะเบียนและประมวลผล  
ส่วนงานทะเบียนและค่าธรรมเนียม

The Office Of The Register's Information System

Registration and Fee Section



นายวิวัฒน์ อุณสุทธิรักษ์ รหัส 37013307

นายตรีรัฐ สัจจา รหัส 37013293

วัน เดือน ปี - 1. ค.ค 2539  
เลขทะเบียน..... 038319  
เลขเรียกหนังสือ... T.39.339. ๑๗๖๖

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ออกแล้วเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปีการศึกษา 2539

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2539

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสารสนเทศของสำนักทะเบียนและประมวลผล ส่วนงานลงทะเบียนและค่าธรรมเนียม

ผู้จัดทำ

1. นายวิวัฒน์ อุนสุทธิรักษ์
2. นายตรีรัฐ สัจจา



..... อาจารย์ที่ปรึกษา

(วิบูลย์ พร้อมพานิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ระบบสารสนเทศของสำนักทะเบียนและประมวลผล

## ส่วนงานลงทะเบียนและค่าธรรมเนียม

วิวัฒน์ อุนสุทธิรักษ์

ตรีรัฐ ตัจจา

วิบูลย์ พร้อมพานิชย์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2539

### บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอ การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายแบบ ไคล์เอนท์ /เซิร์ฟเวอร์ เกี่ยวกับการจัดการฐานข้อมูล เพื่อใช้ในการสำหรับสำนักทะเบียนและประมวลผล ใน ส่วนของการลงทะเบียนเรียนของนักศึกษา

ในการพัฒนาระบบฐานข้อมูลดังกล่าวจะใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ของบริษัท Microsoft ที่ชื่อว่า Microsoft SQL Server และใช้ Visual Basic ในการพัฒนาโปรแกรมในส่วนของการติดต่อผู้ใช้งาน (Graphics User terface : GUI) ซึ่งระบบนี้จะทำงานภายใต้เครือข่ายของ Windows NT

สถาปัตยกรรม โครงสร้างของ โครงการนี้ ได้นำแนวความคิดของไคล์เอนท์/เซิร์ฟเวอร์ ที่มี ลักษณะแบบสามชั้นมาพัฒนาซึ่งจะแบ่งออกเป็น ชั้นที่หนึ่งจะเป็นส่วนติดต่อกับผู้ใช้งาน ชั้นที่ สองเป็นส่วนควบคุม

# **The Office Of The Register's Information System**

## **Registration and Fee Section**

Wiwat Unsuttirag

Treerath Satjar

Wiboon Prompanich Advisor

1996

### **ABSTRACT**

This thesis presents the development of an application on client/server network system. This application is relational database system for processing in office of central registrar and processing in students's registration.

During the implementation, this system uses Microsoft SQL Server to be a RDBMS ( Relational Database Management System ) and uses Microsoft Visual Basic for GUI (Graphics User Interface) programming. This system will be run on Windows NT networking.

This project is use Client/Server type 3 Tier architecture. First is user interface tier, second is business tier and final is data tier

## สารบัญ

เรื่อง	หน้าที่
บทที่ 1 บทนำ	1
บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับโครงการ	2
สถาปัตยกรรม Three-Tier	2
ARCHITECTING TIER 1 : CLIENT APPLICATIONS	4
ARCHITECTING TIER2 : BUSINESS SERVER	6
ARCHITECTING TIER 3 : DATA SERVERS	7
การออกแบบฐานข้อมูล	8
แนวความคิดแบบออบเจกต์โอเรียนเต็ล (Object Oriented)	9
แอปสแตร์กซ์	9
อินเซิร์แทนส์	12
ออบเจกต์โอเรียนเต็ลดีไซน์ (Object Oriented Design)	13
การออกแบบระบบ (System Design)	13
SQL Server	15
สถาปัตยกรรมของ SQL Server (Architectural Overview )	15
สถาปัตยกรรมของเซิร์ฟเวอร์ (Server Architecture)	16
ส่วนประกอบของเซิร์ฟเวอร์ (Server Components)	16
ส่วนประกอบของไคลเอ็นต์ (Client Components)	18
เครื่องมือช่วยในการพัฒนาระบบ	19
Databases in SQL Server	19
ระบบฐานข้อมูล (System Database)	20
User Database	21
พจนานุกรมข้อมูล (Data Dictionary)	21
ชนิดของข้อมูล (Data types)	22
Data Integrity และ Concurrency Control	24
สโตร์โพรซีเจอร์ (Stored Procedures)	25
Object Linking and Embedding (OLE)	27
รีโมทดาต้าออบเจกต์ (Remote Data Objects) (RDO)	30
การเข้าถึงข้อมูล (Accessing Data)	34

## สารบัญ(ต่อ)

เรื่อง	หน้าที่
บทที่ 3 การออกแบบและพัฒนาระบบ	36
ศึกษาระบบ	36
โครงสร้างของโครงการ	36
Object Hirarchie โดยรวมของระบบ	37
ARCHITECTING TIER 1 : CLIENT APPLICATIONS	37
รายละเอียดของแต่ละ Object ของส่วนการลงทะเบียน	38
รายละเอียดของแต่ละ Object ส่วนชำระค่าธรรมเนียม	41
ARCHITECTING TIER2 : BUSINESS SERVER	43
Object Hirarchie โดยรวมของระบบการลงทะเบียน	43
Object Hirarchie โดยรวมของระบบชำระค่าธรรมเนียม	44
ARCHITECTING TIER 3 : DATA SERVERS	45
ระบบการลงทะเบียน ตารางข้อมูลหลักที่เกี่ยวข้อง	45
ระบบชำระค่าธรรมเนียม ตารางข้อมูลหลักที่เกี่ยวข้อง	46
Tier2 และ Tier 3 RegMgr Object Interface ของระบบการลงทะเบียน	48
Tier2 และ Tier 3 RegMgr Object Interface ของ ระบบชำระค่าธรรมเนียม	49
รูปแสดงหน้าจอการทำงานของโปรแกรมส่วนระบบการลงทะเบียน	51
รูปแสดงหน้าจอการทำงานของโปรแกรมส่วนระบบการชำระค่าธรรมเนียม	63
บทที่ 4 ผลการทดลองแนวทางพัฒนาต่อและสรุป	71
กิตติกรรมประกาศ	72
หนังสืออ้างอิง	73

## สารบัญรูปลูกภาพ

รูป	หน้าที่
รูปแสดงออฟเจ็ค	11
รูปแสดง อินเซิร์ทแทนส์	12
รูป แสดงถึงความสัมพันธ์ระหว่างไครเอ็นต์ และ SQL Server	15
รูปแสดงส่วนประกอบของ SQL Server	16
รูปแสดง โครงสร้างของ SQL Server	19
รูปแสดง System และ User database	20
รูปแสดง RDO Object Model	30
รูปแสดง โครงสร้างของ โครงการงาน	36
รูปแสดง Object Hirarchie Tier 1	37
รูปแสดง Object Hirarchie Tier 2	43
รูปแสดงหน้าจอเข้าสู่ระบบการลงทะเบียน	51
รูปแสดงหน้าจอการลงทะเบียน	52
รูปแสดงหน้าจอขึ้นชั้นการลงทะเบียน	53
รูปแสดงหน้าจอการเปลี่ยน/เพิ่มวิชาเรียน	54
รูปแสดงหน้าจอรายงานการลงทะเบียน	55
รูปแสดงหน้าจอผลรายงานการลงทะเบียน	56
รูปแสดงหน้าจอตรวจสอบการลงทะเบียน	57
รูปแสดงหน้าจอผลการตรวจสอบ	58
รูปแสดงหน้าจอผลการเรียน	59
รูปแสดงหน้าจอการนำเข้าผลการเรียน	60
รูปแสดงหน้าจอรายงานผลการเรียน	61
รูปแสดงหน้าจอแสดงผลการเรียน	62
รูปแสดงหน้าจอระบบชำระค่าธรรมเนียม	63
รูปแสดงหน้าจชำระค่าธรรมเนียม	64
รูปแสดงหน้าจอนำเข้าอัตราค่าธรรมเนียม	65
รูปแสดงหน้าจอนำเข้าข้อมูลการยกเว้นค่าธรรมเนียม	66
รูปแสดงรูปหน้าจอตรวจสอบการชำระค่าธรรมเนียม	67
รูปแสดงหน้าจอผลการตรวจสอบ	68
รูปแสดงหน้าจอรายงานการเงิน	69
รูปแสดงหน้าจอผลรายงานการเงิน	70

## สารบัญตาราง

ตาราง	หน้าที่
ตารางแสดงรายละเอียดของ database catalog	21
รายละเอียดของ system catalog	22
Stores table	22
ตารางแสดง Properties ของ rdoEngine Object	31
ตาราง Methods ของ rdoEngine Object	31
ตาราง Properties ของ rdoEnvironment Object	32
ตาราง Method ของ rdoEnvironment Object	32
ตาราง Properties ของ rdoConnection Object	33
ตาราง Method ของ rdoConnection Object	34
ตาราง Properties ของ rdoTable Object	34
ตาราง Method ของ rdoTable Object	34
Register (ตารางวิชาที่ลงทะเบียน)	45
RegisterCard (ตารางใบลงทะเบียน)	45
Fee(การเก็บค่าธรรมเนียม)	46
FeeCost(ค่าธรรมเนียม)	46
FeeExcept(รายการยกเว้นค่าธรรมเนียม)	47
FeeExceptName(ชื่อรายการยกเว้นค่าธรรมเนียม)	47
FeeSubType(รายการค่าธรรมเนียมย่อย)	47
FeeType(ชนิดค่าธรรมเนียม)	47
Tier2 และ Tier 3 RegMgr Object Interface ของระบบการลงทะเบียน	48
Tier2 และ Tier 3 RegMgr Object Interface ของระบบชำระค่าธรรมเนียม	49

# บทที่ 1

## บทนำ

ในปัจจุบันการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายแบบไคลเอนต์/เซิร์ฟเวอร์ กำลังเป็นที่ถูกนำเสนอขึ้นมาแทนที่โปรแกรมประยุกต์ที่มีการประมวลผลแบบรวมศูนย์ ซึ่งช่วยแบ่งเบาการประมวลผลไปอยู่ที่ไคลเอนต์ และการพัฒนาโปรแกรมในแบบนี้ก็ได้มีการนำเสนอออกมาในหลายรูปแบบการพัฒนา และรูปแบบหนึ่งที่ถูกเสนอแนะขึ้นมาก็คือ สถาปัตยกรรมแบบ ทรี-เทียร์ (Three-Tiers Architecture) ซึ่งเป็นรูปแบบที่ถูกนำมาใช้ในการพัฒนาโครงการงานชิ้นนี้

ในการพัฒนาโครงการงานนี้เราเริ่มทำจากการศึกษาระบบการทำงานของสำนักทะเบียนและรวบรวมข้อมูลเพื่อนำมาวิเคราะห์และออกแบบโครงสร้างของโครงการงาน ซึ่งในการวิเคราะห์และออกแบบเราได้ใช้วิธีการของออบเจกต์โอเรียนเต้ดมาช่วยในการออกแบบ โดยโครงการงานนี้จะถูกแบ่งออกเป็นสามส่วนด้วยกัน ดังต่อไปนี้

1. ส่วนของการติดต่อกับผู้ใช้งาน (User Interface) ในส่วนนี้จะ เป็น โปรแกรมที่พัฒนาจากโปรแกรมภาษาวิซวลเบสิกของบริษัท ไมโครซอฟต์ ทำหน้าที่เป็นตัวติดต่อกับผู้ใช้งานในส่วนนี้จะเป็นส่วนที่ใช้แสดงผลกับรับข้อมูล ไม่มีการทำการคำนวณและจะไม่มีการเปลี่ยนแปลงข้อมูล โปรแกรมบ่อยครั้ง
2. ส่วนของกฎการควบคุม หรือ Business Rule ในส่วนนี้จะใช้ควบคุมกฎการทำงานที่ต้องการให้โปรแกรมทำงานเป็นตัวกลางประสานการทำงานระหว่างส่วนที่หนึ่งกับส่วนที่สามที่เป็นส่วนการจัดการฐานข้อมูล,คำนวณ,ประมวลผลข้อมูล,กำหนดขอบเขตการทำงานซึ่งอาจมีการเปลี่ยนแปลงกฎการทำงาน ในส่วนนี้ใช้ภาษาวิซวลเบสิกในการพัฒนา
3. ส่วนที่ควบคุมและจัดการเกี่ยวกับฐานข้อมูล(Data Base Management) เป็นส่วนที่จัดการฐานข้อมูลที่เกี่ยวข้องกับการลงทะเบียนและค่าธรรมเนียม โดยใช้โปรแกรมจัดการฐานข้อมูลที่มีชื่อว่า MS-SQL Server ของบริษัท ไมโครซอฟท์

## บทที่ 2

### ความรู้เบื้องต้นเกี่ยวกับโครงการ

#### สถาปัตยกรรม Three-Tier

เนื่องจาก Client/Server Model แบบ Two-Tier ยังมีข้อบกพร่องอย่างเช่น ยังไม่มีความสามารถในการขยายระบบ , ขาด Performance ที่ดี และยากต่อการจัดการ เป็นต้น ผู้ใช้จึงเปลี่ยนมาใช้ Three-Tier ซึ่งเป็นรูปแบบของ distributed-computing model หรือที่รู้จักกันดีชื่อของ Service Model ซึ่งเป็นการเปลี่ยนแปลงบทบาทของ server และ client และเพิ่มส่วนประกอบสถาปัตยกรรมใหม่เข้าไป คือ business service

Business service จะแยกตัวออกจาก data service และ client

**Three-Tier** ประกอบด้วย 3 ส่วน คือ

1. **User Service** ซึ่งจะเป็นส่วนที่ใช้ติดต่อกับ User และ โดยปกติแล้ว โปรแกรม Execute ของส่วนนี้จะฝังอยู่ใน Client Workstation การ design ที่ดีจะต้องทำให้ user สามารถ
  - เข้าใจการบริการในการใช้งาน ( Business Services ) ต่างๆ ได้
  - ใช้การบริการในการใช้งาน ได้อย่างมีประสิทธิภาพ

2. **Business Service** เป็นส่วนที่เชื่อมต่อส่วนของ User Service และ Data Service ซึ่งส่วนนี้จะคอยตอบสนองงานของ User และ Execute งานต่างๆ ของ User สร้างจาก procedure ปกติทั่วไป และ Business Rule ซึ่งส่วนนี้จะเป็นการไม่ให้ User ติดต่อกับ Database โดยตรงเพราะ Business Rules จะเปลี่ยนแปลงบ่อยกว่า Business Task แบ่งหน้าที่ออกเป็น 2 อย่าง คือ

- กำหนดหลักการในการทำงานขององค์กรและ โดย Data Service ไม่เปลี่ยนแปลง
- การเก็บรวบรวมโมเดลของ Business และ ก็แสดง โมเดลให้ดูที่ Client Applications โดย

ไม่มีความยุ่งยาก

3. **Data Services** เป็นการให้ นิยาม, บำรุง, เข้าถึง, และแก้ไขข้อมูล ซึ่งใน Three-Tier จะรวม Data Servers ซึ่งส่วนใหญ่จะเป็น SQL-based database management system

- Data Server จะจัดการข้อมูลดิบขององค์กรและควบคุมการติดต่อกันสำหรับ Business servers ซึ่งการตอบสนองขั้นแรกๆ ของ Data Services จะเป็นการเตรียม Business Services กับงานการติดต่อขั้นต่ำ และ รับประกันส่วนประกอบของข้อมูลซึ่งใน Business Server ใหม่จะใช้ Data Server แต่ยังไม่มียุทธวิธีในการที่บรรเทาการออกแบบ Database ที่ดี เพราะว่า - - Business Server จะให้ความยืดหยุ่นอย่างมากในการติดต่อกับข้อมูล แต่ไม่ได้หมายความว่า การออกแบบ Database จะทำอย่าง ไม่สมบูรณ์

## ข้อดีของสถาปัตยกรรมแบบ Three-Tier

1. สามารถทำการพัฒนาระบบได้เพราะ

- สถาปัตยกรรมเป็นแบบลำดับชั้นทำให้เมื่อเปลี่ยนแปลงการ Service หนึ่งอย่างจะไม่มีผลกับ Service อื่นอื่น

- ถ้ามีการเปลี่ยนการทำงานภายในของ Service แต่ไม่ได้เปลี่ยนที่ระบบการติดต่อกับผู้ใช้ Service อื่นก็ไม่จำเป็นต้องแก้ไข

- ถ้าต้องการเปลี่ยนระบบการติดต่อกับผู้ใช้ เราสามารถทำการ Design ใหม่จาก ระบบเดิมที่มีอยู่แล้ว

2. Business Servers สามารถใช้ร่วมกันได้

3. Business object และ Processes สามารถจะรักษาความปลอดภัยได้

เนื่องจาก Security สามารถย้ายออกจาก Data Services และ Client Application ไปเป็น Business Services Tier ซึ่ง Client Application จะไม่สามารถติดต่อโดยตรงไปยัง Data Services ดังนั้นจะมี Business Server ที่สามารถสร้าง Security ขึ้นมาได้ ซึ่ง Business server ไม่เพียงแต่สามารถควบคุมการเปลี่ยนแปลงและการเข้าถึงของข้อมูล แต่ยังต้องรู้ว่าข้อมูลจะถูกเข้าถึงได้อย่างไร และเปลี่ยนแปลงได้อย่างไร

อย่างไรก็ตาม Business Server สามารถใช้ Security ได้กับตัวอย่าง เช่น E-Mail

4. User สามารถจะสร้าง Client Applications ของตัวเองได้ เพราะ Business Rules สามารถตัดออกจาก Front-End ได้

5. Business Model จะมี Tier เป็นของตัวเอง

Business Object ประกอบด้วย

- คุณสมบัติ ( Properties )

- วิธีการ ( Method )

ซึ่งทั้ง 2 อย่างนี้ คือ attributes ของ objects

6. The Business Process Engine has a Tier

ใน Business Server จะมีการทำงานอยู่ 2 ชนิด คือ

- การทำงานที่มีโมเดลของ Business และการติดต่อตรงไปยัง Client Application

- การทำงานของ Process ที่จะรันพร้อมกันและทำงานแบบอัตโนมัติ

เช่น Order-processing object ที่จะตอบโต้กับ Client Applications ในการอนุญาตให้ User ดูข้อมูล Customer , ใ้ และ ลบ Order ซึ่งเป็นตัวอย่างของ Business Server ชนิดแรก

หลังจาก Order ถูกนำเข้า Order-processing object จะทำการติดต่อกับ Business process อื่นอื่นในการทำงาน.

## ตัวอย่างของการบริการในแต่ละ Tier

Tier	Service
Client application	Order processing
	Customer maintenance
	Product maintenance
Business servers	Order processing
	Accounts receivable
	Warehouse management
Data Servers	DBMS: product
	DBMS: customer
	DBMS: orders
	Messaging: Lotus Notes
Communications: EDI server	

## ARCHITECTING TIER 1 : CLIENT APPLICATIONS

### การเลือก Controlling application

ถ้าเราจะสร้าง application ขึ้นมาจะต้องมีฟังก์ชันที่เรียกว่า “Controlling client application” ซึ่งจะให้ user interface เบื้องต้นและทำงานร่วมกับ user services อื่นได้ The controlling client application จะไม่มีอยู่ใน OLE server เพราะโดยทั่วไปมันจะไม่ถูกควบคุมโดย applications ตัวอื่น แต่จะมีอยู่ใน OLE control, OLE automation servers และ OLE documents

The controlling application ต้อง

1. ให้ความสามารถในการติดต่อกับส่วนประกอบต่างๆ ใน user services และ business services ได้ เช่น OLE automation servers และ OLE controls

2. อนุญาตให้นักพัฒนาสร้างส่วนที่ใช้ติดต่อกับผู้ใช้ได้

3. มีภาษามาตรฐานสำหรับการใช้ในการพัฒนา

โดยภาษา Visual Basic มีหลักการทั้ง 3 ข้อนี้อย่างครบถ้วน

Client application 16 บิต และ 32 บิต

Visual basic 4.0 ได้เตรียมให้ผู้พัฒนาโปรแกรมสามารถสร้าง applications ทั้ง 16 บิต และ 32 บิต จาก source code อันเดียวกัน application แบบ 32 บิต ได้เตรียมข้อดีทั่วไปที่เหนือกว่า application แบบ 16 บิต อย่างไรก็ตามข้อดีส่วนใหญ่จะทำงานภายใต้ windows NT และ windows95

Applications แบบ 32 บิต จะต้องการ operating system แบบ 32 บิตในการทำงาน ถึงแม้ว่า application แบบ 32 บิตของ visual basic มีข้อดีมากมายที่เหนือกว่า application แบบ 16 บิต แต่ operating system แบบ 32 บิตจะยังไม่ถูกใช้โดยผู้ใช้ทุกคน

### Conditional Compilation

การพัฒนา client application ทั้ง 16 บิตและ 32 บิต จาก source code อันเดียวกัน มันยาก เพราะ client application แบบ 16 บิตจะไม่สามารถเรียกใช้ 32 บิต API ได้ มันจะเรียกใช้ได้แค่ 16 บิต API เท่านั้น ดังนั้นในการ compile เป็น .EXE จะไม่สามารถทำได้เพราะจะทำให้เกิด runtime error

แต่ visual basic สามารถสร้าง application แบบ 16 บิตและ 32 บิต จาก source code อันเดียวกันได้จากวิธีการที่เรียกว่า “Conditional compilation”

Conditional compilation จะอนุญาตให้เรารวมคำสั่งในโปรแกรมในการ compile เป็นไฟล์ .EXE โดยจะเป็นการสร้าง compilation constants เมื่อ compile application แล้ว visual basic จะตรวจสอบค่าของ compilation constants เพื่อจะรู้ว่าจะต้อง compile แบบใด

### ข้อดีของ application แบบ 32 บิต

- 1.เพิ่มความเร็วในการ execute โปรแกรม
- 2.มีการจัดการ resources ที่ดีกว่า
- 3.เพิ่มความมั่นคงของระบบ
- 4.เป็นการทำงานแบบ Preemptive multitasking

### Accessing other user services

Client application จะต้องมีความสามารถในการใช้ user services อื่นๆ ที่อยู่ใน ส่วนของ user services ซึ่ง visual basic client application จะเชื่อมต่อส่วนประกอบต่างๆ เข้าไว้ด้วยกัน เพื่อที่จะสร้างส่วนที่ผู้ใช้ติดต่อกับผู้ใช้

โดยใน User Service นี้จะมี Component มากมาย เช่น

- OLE Controls ( OCXes )
- OLE Automation Servers
- OLE Documents
- DLLs

## หลักการพัฒนา Client Application

1. รวบรวมความต้องการของ Client Application
2. ออกแบบ Client Application
3. ทดลองใช้ Application
4. ดำรงและติดตามผลของระบบที่เราได้ทำไปแล้ว
5. ตรวจสอบและแก้ไขแอปพลิเคชัน
6. ทำซ้ำในข้อ 4 และ 5 จนสำเร็จ

## ARCHITECTING TIER2 : BUSINESS SERVER

Business Object คือ application object ที่มีฟังก์ชันใน object เอง และ applications อื่นๆ สามารถเรียกใช้ได้ ซึ่งใน service model ในสถาปัตยกรรมแบบ three tier business object จะ map อยู่ในส่วน Second Tier

### ชนิดของ Business object มี 2 ชนิด

1. เป็นวัตถุที่สัมผัสได้ ( Tangible Object ) หรือสัมผัสไม่ได้ ( Intangible Object )
2. Business Event

เป็นต้น

Tangible Object เป็นวัตถุที่จับต้องได้ซึ่งรวมไปถึง Paper Base เช่น ใบสั่งซื้อ ใบร้องทุกข์

องค์การ

Intangible Object เป็นวัตถุที่จับต้องไม่ได้ รวมถึง Process และความสัมพันธ์ภายใน

Business Event เป็นเหตุการณ์ที่เกิดขึ้นจากการเปลี่ยนแปลง Object

### ข้อดีของ Business Object

1. มีความสามารถในการนำกลับมาใช้ใหม่ได้
2. สามารถปรับเปลี่ยนได้
3. สามารถบำรุงรักษาได้
4. มีความยืดหยุ่น
5. สามารถ Share กันได้
6. สามารถเรียกใช้จาก Tool ของ End User ได้
7. สามารถออกแบบการ Interface ได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ชนิดของ Business Services

### 1. Batch Business Service

- สามารถ Run ได้เมื่อต้องการ
- Operation สามารถถูกจัดการด้วยเวลาที่แน่นอนด้วยตัวจัดการของระบบ

### 2. Interactive Business

Business Server จะติดต่อบนแบบ Interactive กับ Client Application หลายๆตัวจะ Run ที่ Desktop ของ User

## ARCHITECTING TIER 3 : DATA SERVERS

Data service จะจัดการข่าวสารขององค์กร ซึ่งข่าวสารนี้จะเป็นที่รวบรวมของข้อมูลที่ใช้โดยชั้นของ business service ที่จะใช้ในการตัดสินใจและถูกควบคุมโดย business rules

Data service จะเป็นที่เก็บข้อมูลและความสัมพันธ์ระหว่างข้อมูล

### DBMS Data Services

ความหมายของคำต่างๆ ที่จำเป็น

Database คือ เป็นที่สะสมของข้อมูลที่มี ความสัมพันธ์ระหว่างข้อมูลและออกแบบให้ตรงกับความต้องการของ User แต่ละคน

Database management system ( DBMS ) เป็นซอฟต์แวร์ที่ใช้ในการสร้าง, บำรุงรักษา และเตรียมการควบคุมการเข้าถึงข้อมูลที่อยู่ใน Database

Relational database เป็น Database ที่เก็บสะสมที่แทน โดยความสัมพันธ์ต่างๆ หรือ Table

### The Relational Database Model

Relational Database จะมีความสัมพันธ์ของข้อมูลอยู่ 3 แบบภายในระหว่าง Table

- One-to-One ต้องการเพียง 1 record ใน Table ที่มีความสัมพันธ์กับอีก 1 record ใน primary table
- One-to-Many ต้องการเพียง 1 record หรือมากกว่าในการเชื่อมความสัมพันธ์กับ 1 record ใน primary table
- Many-to-Many อนุญาตให้ 1 record หรือมากกว่า ในการเชื่อมความสัมพันธ์กับ 1 record หรือ มากกว่าใน primary table

## การออกแบบฐานข้อมูล

โดยส่วนใหญ่จะใช้วิธี Normalization ซึ่งเป็นวิธีการจัดการความซ้ำซ้อนของข้อมูลใน Database ซึ่งจุดมุ่งหมายในการ Insert , delete and update record ภายใน Table โดยปราศจากข้อผิดพลาด

วิธีการ Normalization จะมีอยู่ 6 ขั้นตอนดังต่อไปนี้ คือ

1. First Normal Form ( 1 NF ) repeating group จะถูกขจัดทิ้ง
2. Second Normal Form ( 2 NF ) Partial functional dependencies จะถูกขจัดทิ้ง
3. Third Normal Form ( 3 NF ) Transitive dependencies จะถูกขจัดทิ้ง
4. Boyce-Codd Normal Form ( BCNF )
5. Fourth Normal Form ( 4 NF ) Multi-Valued dependencies จะถูกขจัดทิ้ง
6. Fifth Normal Form ( 5 NF )

### Using SQL to Access the Database

SQL มี Data definition language ( DDL ) ซึ่งเป็นภาษาที่ใช้ในการบรรยายเมื่อเราจะทำการสร้างหรือเปลี่ยนแปลง โครงสร้างของ Table ,Columns, Indexes หรืออื่นๆที่อยู่ใน Relational Database เช่น

- CREATE เป็นคำสั่งที่ใช้ในการสร้าง Table, Index, View และ Object อื่นๆใน Database
- ALTER จะเป็นการเปลี่ยนแปลงแก้ไขนิยามของ Table, Index, View และ Object อื่นๆ

ใน Database

- DROP เป็นคำสั่งที่เป็นการขจัด Table, Index, View หรือ Object อื่นๆ ใน Database
- SELECT เป็นคำสั่งที่ใช้ในการ Query ข้อมูลที่เราต้องการให้แสดงผล
- INSERT เป็นคำสั่ง Data manipulation ที่ใช้ในการแทรกข้อมูลเข้าสู่ Database
- DELETE เป็นคำสั่งที่ใช้ในการลบ Record ออกจาก Database

เป็นต้น

## แนวความคิดแบบออบเจกต์โอเรียนเต็ด (Object Oriented)

ต่อไปนี่เราจะมาศึกษาแนวความคิดที่กำลังได้รับความนิยมในวงการคอมพิวเตอร์ในปัจจุบัน ซึ่งแนวความคิดดังกล่าวนี้ก็คือแนวความคิดของออบเจกต์โอเรียนเต็ดนั่นเอง มันดูเหมือนว่าการใช้คำว่า “ออบเจกต์โอเรียนเต็ด” จะแสดงให้เห็นถึงความทันสมัย ความสามารถที่ดี และมีประโยชน์ต่อวงการสารสนเทศในปัจจุบันมาก อย่างไรก็ตามการนำแนวความคิดนี้มาใช้ให้เกิดประโยชน์อย่างมีประสิทธิภาพนั้นจำเป็นต้องมีความเข้าใจถึงแนวความคิดดังกล่าวอย่างถ่องแท้ด้วยเช่นเดียวกัน

คำว่า “ออบเจกต์โอเรียนเต็ด” ที่กำลังพูดถึงนี้ไม่ได้ถูกจำกัดอยู่เฉพาะกับการโปรแกรมมิ่งแบบออบเจกต์โอเรียนเต็ดเพียงอย่างเดียว แต่ยังรวมไปถึงปรัชญาในการวิเคราะห์ระบบ (System Analysis) การออกแบบระบบ (System Design) การออกแบบฐานข้อมูล (Database Design) และเรื่องอื่นๆที่เกี่ยวข้องในสาขาเดียวกันนี้ด้วย

ซึ่งในที่นี้เราจะแนะนำแนวความคิดพื้นฐานและคำเฉพาะที่ใช้ในระเบียบวิธีทางออบเจกต์โอเรียนเต็ด (Object Oriented Methods) โดยเราจะเริ่มต้นดูกันถึงเรื่องราวที่เป็นภูมิหลังหลังของแนวความคิด ดังต่อไปนี้

### อะไรคือระเบียบวิธีทางออบเจกต์โอเรียนเต็ด

ดังที่ได้กล่าวมาแล้วถึงคำว่า “ระเบียบวิธีทางออบเจกต์โอเรียนเต็ด” นี้หมายถึงหลายๆสิ่งเช่นเดียวกับคำว่าออบเจกต์โอเรียนเต็ด แต่ความหมายที่เฉพาะที่เรากำลังสนใจอยู่ก็จะได้แก่ในเรื่องของออบเจกต์โอเรียนเต็ด โปรแกรมมิ่ง, การออกแบบแบบออบเจกต์โอเรียนเต็ด, การวิเคราะห์แบบออบเจกต์โอเรียนเต็ด และฐานข้อมูลแบบออบเจกต์โอเรียนเต็ดหรือกล่าวอีกนัยหนึ่งก็คือทุกๆเรื่องที่เกี่ยวข้องกับการพัฒนาระบบที่ใช้หลักการของออบเจกต์โอเรียนเต็ดนั่นเอง

ในหัวข้อต่อไปเราจะแนะนำให้ทราบถึงคำเฉพาะที่ใช้ในระเบียบวิธีทางออบเจกต์โอเรียนเต็ด แต่ผู้อ่านควรตระหนักถึงคำเฉพาะเหล่านี้มีการนำไปใช้ในความหมายอื่นที่แตกต่างกันสำหรับผู้เขียนแต่ละคนเหตุที่เป็นเช่นนี้ก็เนื่องมาจากว่าเรื่องของออบเจกต์โอเรียนเต็ดนี้เป็นเรื่องที่เกิดขึ้นมาใหม่นั้นเอง

### คำศัพท์เฉพาะและแนวคิดพื้นฐานเบื้องต้น

แนวความคิดแบบออบเจกต์โอเรียนเต็ดไม่ว่าจะเป็นในเรื่องของการออกแบบ การโปรแกรม หรือ อื่นๆ จะถูกกำหนดอยู่ภายใต้คุณสมบัติอยู่ 2 คุณสมบัติ ซึ่งคุณสมบัติทั้งสองนั้นก็คือ

1. แอปสเตรกชัน (Abstraction)
2. อินฮีริเทนส์ (Inheritance)

ทั้งคำว่า แอปสเตรกชัน และ อินฮีริเทนส์ ต่างก็ร่วมกันปกปิดแนวความคิดที่สำคัญอื่นๆไว้ภายใต้มัน ซึ่งในหัวข้อต่อไปเราจะได้เปิดเข้าไปถึงแนวความคิดที่ถูกปิดบังไว้เหล่านี้ เพื่อเป็นการสร้างความเข้าใจถึงคำเฉพาะต่างๆที่จะต้องใช้กันในภายหลัง

## แอปสแตรกซ์

แอปสแตรกซ์ มีความหมายถึงการแทนคุณสมบัติที่สำคัญๆของบางสิ่งบางอย่างโดยไม่นำไปรวมรายละเอียดที่ไม่จำเป็น ในทางโปรแกรมมิ่งคำนี้จึงหมายถึงว่าออปเจกต์ควรถูกแทนที่ด้วยการรวมข้อมูลและ โพรเซสที่เกี่วข้องกับมันไว้ด้วยกันเพื่อเป็นตัวแทนของออปเจกต์นั้น

แนวความคิดพื้นฐานก็คือว่าออปเจกต์จะถูกกำหนดโดยกลุ่มของแอททริบิวต์ (Attributes) ที่ใช้ร่วมกันใช้แทนออปเจกต์นั้น แอททริบิวต์ดังกล่าวนี้อาจมีชื่อเรียกว่าเป็น อินสแตนส์วาริเอเบิล (Instance Variable) หรือ คลาสส์วาริเอเบิล (Class Variable) นอกจากกลุ่มของแอททริบิวต์แล้ว ออปเจกต์ก็จะถูกกำหนดไปพร้อมๆกันด้วยกลุ่มของ โพรซีเจอร์ (Procedure) ที่ได้รับอนุญาตที่สามารถปฏิบัติการเกี่ยวกับแอททริบิวต์ของออปเจกต์ได้ โดยปกติแล้วแอททริบิวต์ของออปเจกต์จะไม่สามารถเข้าถึงได้จากภายนอก ต้องเข้าถึงแอททริบิวต์เหล่านั้น โดยผ่านทางโพรซีเจอร์ของออปเจกต์นั้นเพียงอย่างเดียว

แนวความคิดที่ทำการผูกฟังกซ์ชันและข้อมูลเข้าด้วยกันนี้มีความสัมพันธ์ใกล้ชิดกับแนวความคิดเรื่องชนิดของข้อมูลในภาษาโปรแกรมมิ่งทั่วไป ในขณะที่ 3 เป็นอินสแตนส์ของข้อมูลชนิดจำนวนเต็ม และจำนวนเต็มถูกกำหนดให้มีแอททริบิวต์หนึ่งตัวคือค่าของมันเอง แต่อนุญาตให้ที่โอเปอเรชัน (Operation) ทางคณิตศาสตร์ได้หลายตัว ตัวเลขจำนวนจริงถูกกำหนดให้มีสิ่งต่างๆในลักษณะเดียวกัน แต่การอิมพลีเมนต์ (Implementation) ของการคูณสำหรับตัวเลขแบบ ฟลอยติงพอยต์ (Floating Point) จะแตกต่างกับการคูณตัวเลขจำนวนเต็มธรรมดา ในลักษณะนี้เราจึงสามารถจัดการกับออปเจกต์ที่ซับซ้อนเป็นต้นว่า ลูกข้าง ได้โดยกำหนดแอททริบิวต์ให้มันและกำหนดเมทอดส์ (Methods), โอเปอเรชัน หรือ เซอร์วิส (Service) ให้กับมัน ทั้งคำว่า เมทอดส์, โอเปอเรเตอร์ และเซอร์วิส นั้นเป็นคำที่ใช้เรียก โพรซีเจอร์ที่มีการปกปิดถึงการอิมพลีเมนต์จริงๆไว้

คำว่า คลาสส์ (Class) ในความหมายทางออปเจกต์โอเรียนเต็ด โปรแกรมมิ่งจะหมายถึงกลุ่มของออปเจกต์ที่ใช้แอททริบิวต์และเมทอดส์พื้นฐานร่วมกัน

ออปเจกต์มีมุมมองจากทั้งภายในและภายนอก มุมมองภายในจะบอกถึงสถานะ, การอิมพลีเมนต์ของออปเจกต์ ส่วนมุมมองภายนอกจะแสดงชื่อของเมทอดส์และชนิดของพารามิเตอร์ (Parameter) รูปต่อไปนี้เป็นสัญลักษณ์ที่ใช้ในการเขียนอธิบายออปเจกต์หนึ่งๆ

<b>Mail message</b>
Sender
Receive
Sender Address
Receiver Address
Date Sent
Date received
Route
Title
Text
Send
Present
File
Print

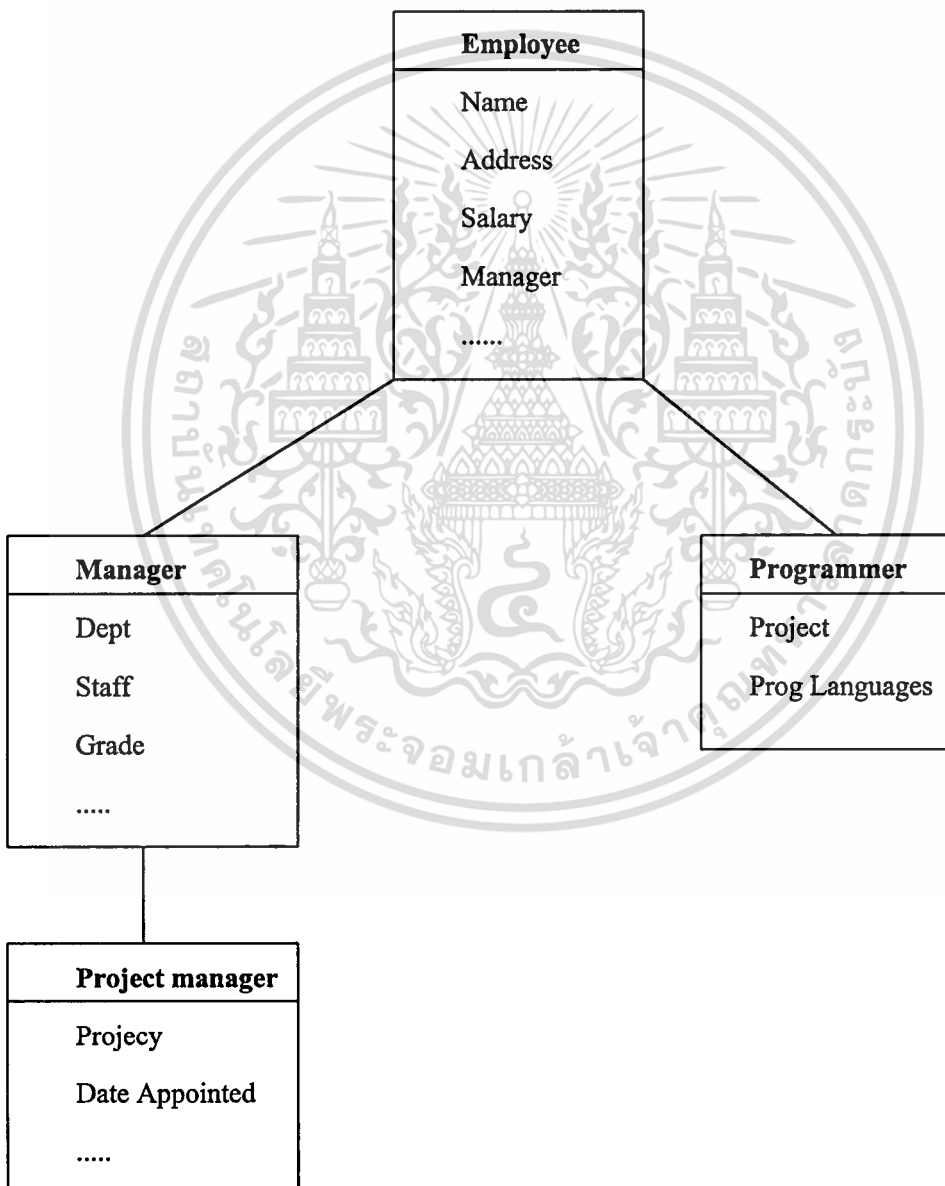
### รูปแสดงออฟเจ็ก

จากแนวความคิดในการปกปิดข้อมูลไม่ให้เห็นการติดต่อได้โดยตรงจากภายนอกของออฟเจ็กต์ การติดต่อระหว่างออฟเจ็กต์จึงเป็นการใช้ เมสเสจ ติดต่อกันระหว่างออฟเจ็กต์ เมสเสจหนึ่งจะประกอบด้วยแอดเดรส (Address) ของออฟเจ็กต์ที่จะส่งเมสเสจนั้นไปถึงและคำสั่งที่ประกอบด้วยชื่อของเมทอดส์และตามด้วยพารามิเตอร์ถ้ามี ถ้าออฟเจ็กต์ที่รับเมสเสจนั้นมีเมทอดส์ที่ต้องการตอบข้อมูลบางตัวก็สามารถส่งกลับไปให้ออฟเจ็กต์ที่ส่งเมสเสจนั้น

## อินเฮียริเทนส์

คุณสมบัติของออบเจกต์โอเรียนเต็ลอีกอันหนึ่งก็คือ อินเฮียริเทนส์ ซึ่งเป็นวิธีการในการจัดการกับความสัมพันธ์ทางโครงสร้างและความหมายระหว่างออบเจกต์และคลาสส์และตัดความซ้ำซ้อนของการที่ต้องเก็บข้อมูลหรือโพรซีเจอร์มากเกินไปที่จำเป็น

ออบเจกต์หนึ่งๆจะทำการสืบทอดคุณสมบัติหรืออินเฮียริเทนส์ของมันและเนื่องจากคลาสส์ก็เป็นออบเจกต์หนึ่ง ก็สามารุสืบทอดคุณสมบัติจากคลาสส์อื่นได้ด้วย ซึ่งคลาสส์นั้นเป็นซูเปอร์คลาสส์ของคลาสส์ที่ทำการอินเฮียริเทนส์คุณสมบัติมา จากการสืบทอดคุณสมบัติต่อกันนี้ จึงทำให้เกิดอินเฮียริเทนส์เน็ตเวิร์ค (Inheritance Network) ขึ้น



รูปแสดง อินเฮียริเทนส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ออฟเจ็กต์โอเรียนเต็ลดีไซน์ (Object Oriented Design)

เมื่อเราได้ทำการวิเคราะห์โครงสร้างและพฤติกรรมของออบเจ็กต์จนได้โมเดลต่าง ๆ ที่ใช้แทนระบบงานออกมาแล้ว ก็มาถึงขั้นตอนที่นำโมเดลเหล่านี้มาออกแบบระบบ โดยจะมีการจำแนกออฟเจ็กต์อื่น ๆ เพิ่มเติมที่จะต้องใช้ในระบบใหม่ แล้วทำการเชื่อมต่อกันเข้ากับออฟเจ็กต์ใน ส่วนเดิมเพื่อใช้สามารถทำงานร่วมกันได้

ภายหลังจากที่เราทำการออกแบบระบบใหม่ขึ้นมาได้แล้วเราก็จะนำระบบนั้นมาทำการอิมพลีเมนต์ ซึ่งการอิมพลีเมนต์นี้ไม่จำเป็นต้องใช้รักษาหรือฐานข้อมูลแบบออฟเจ็กต์โอเรียนเต็ลจริงๆ ก็ได้ เราสามารถนำระบบที่ออกแบบออฟเจ็กต์โอเรียนเต็ลนี้มาทำการปรับให้อยู่ในรูปแบบที่สามารถนำไปใช้การอิมพลีเมนต์ที่เราต้องการได้

ต่อไปนี้จะของแสดงตัวอย่างสัญลักษณ์ต่าง ๆ ที่ใช้ในวิธีการวิเคราะห์และออกแบบวิธีนี้ในหน้าต่อไป

## การออกแบบและการสร้างแบบจำลอง (Object-oriented Modeling and Design)

เป็นแนวคิดใหม่เพื่อการวิเคราะห์ปัญหาโดยการสร้างแบบจำลองอิงกับปัญหาในรูปแบบที่เป็นรูปธรรม วิธีการออกแบบที่นำมาประยุกต์ใช้มีชื่อว่า Object Modeling Technique(OMT) ซึ่งเป็นการแก้ปัญหาโดยเริ่มจากการ วิเคราะห์ (Analysis), ออกแบบ (Design) และประยุกต์ (Implementation) ขั้นตอนของ OMT เริ่มจากการออกแบบ แบบจำลองของปัญหา แล้วจึงเพิ่มรายละเอียดเข้าไปในแบบจำลองในขั้นตอนของการวิเคราะห์แบบจำลองของ OMT

### การวิเคราะห์

ขั้นตอนการวิเคราะห์เริ่มจากการเก็บรวบรวม ข้อกำหนดและลักษณะของปัญหา จากระบบและนำมาสร้างแบบจำลองทั้งสามดังนี้

- Object Modeling
- Dynamic Modeling
- Functional Modeling

## การออกแบบระบบ (System Design)

การออกแบบระบบ (system design) เป็นการวางกลยุทธ์ในการแก้ปัญหาและจัดหาวิธีการแก้ไขให้กับระบบ รายละเอียดในขั้นตอนนี้ได้แก่ 1. การแบ่งระบบออกเป็นระบบย่อย (Subsystem) 2. จัดสรรระบบย่อยเหล่านี้ให้กับฮาร์ดและซอฟต์แวร์ 3. กำหนดแนวคิดนโยบายการตัดสินใจให้กับระบบเพื่อการออกแบบรายละเอียดของระบบ

ในขั้นตอนของการวิเคราะห์ จุดประสงค์ของในขั้นนี้คือพยายามที่จะกำหนดว่าเราจะต้องทำอะไรบ้างในระบบโดยไม่สนใจวิธีการทำ ในบางกลับกัน ในขั้นตอนของการออกแบบระบบจุดประสงค์ในขั้นนี้ คือพยายามหาวิธีการแก้ปัญหา โดยในขั้นแรกเริ่มในระดับสูงก่อน แล้วจึงค่อย ๆ เจาะลงไปรายละเอียด

ขั้นตอนต่าง ๆ ต่อไปนี้เป็นขั้นตอนที่ผู้ออกแบบระบบจะต้องทำ

- แแต่ระบบออกเป็นระบบย่อย
- กำหนดขั้นตอนที่จะต้องทำงานพร้อมกัน(concurrency)
- จัดสรรระบบย่อยให้แก่โปรเซสเซอร์
- เลือกวิธีการและการบริหารการจัดเก็บข้อมูล
- จัดสรรการให้ทรัพยากรณ์ของระบบโดยรวม
- เลือกวิธีการควบคุมของซอฟต์แวร์
- จัดการเงื่อนไขขอบ (Boundary condition)
- กำหนดผลได้ผลเสียของระบบ

### การออกแบบ ออฟเจกต์ (Object Design)

ในขั้นตอนนี้เป็นการเพิ่มรายละเอียดจากสองขั้นที่ได้กล่าวมา ฉะนั้นจึงขออธิบายเป็นหัวข้อคร่าว ๆ ดังต่อไปนี้

- 1.เพิ่มเติม โอเปอเรชั่นให้กับ ออฟเจกต์ model จาก โมเดลอื่น
  - เลือก โอเปอเรชั่น จาก Process ใน Functional model
  - เลือก Operation จาก เหตุการณ์ใน Dynamic model
2. ออกแบบ Algorithm สำหรับ โอเปอเรชั่น
  - เลือกใช้ Algorithm ที่สามารถ implement ได้ง่าย
  - กำหนด โครงสร้างข้อมูลให้เหมาะกับ Algorithms
  - กำหนด คณิตศาสตร์ภายใน และโอเปอเรชั่น ใหม่ถ้าจำเป็น
  - กำหนด หน้าที่ของแต่ละ โอเปอเรชั่น ให้ชัดเจน
3. กำหนดให้ การเข้าใช้ข้อมูลสามารถทำได้อย่างรวดเร็ว
  - เพิ่ม Association เข้าไปเพื่อความสะดวกในการเข้าใช้ข้อมูล
  - จัดระบบการทำงานใหม่เพื่อเพิ่มประสิทธิภาพ
  - เลี่ยงการใช้ Derived Value เพื่อเลี่ยงการคำนวณซ้ำของ Expression ที่ซับซ้อน
4. พัฒนาการควบคุมของ Software จากขั้นตอน System Design
5. จัดโครงสร้างของ คณิตศาสตร์ เพื่อเพิ่ม inheritance
6. ออกแบบการใช้งาน Association
  - วิเคราะห์เส้นทางของ Association หมายถึงการการประยุกต์ใช้ Pointer แทน Association ที่กำหนดโดยแบ่งเป็น Association แบบทางเดียว และ Association แบบสองทาง
7. กำหนด ออฟเจกต์ แอททริบิวท์ ให้ถูกต้อง
8. รวบรวม คณิตศาสตร์ และ Association ให้เป็น Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

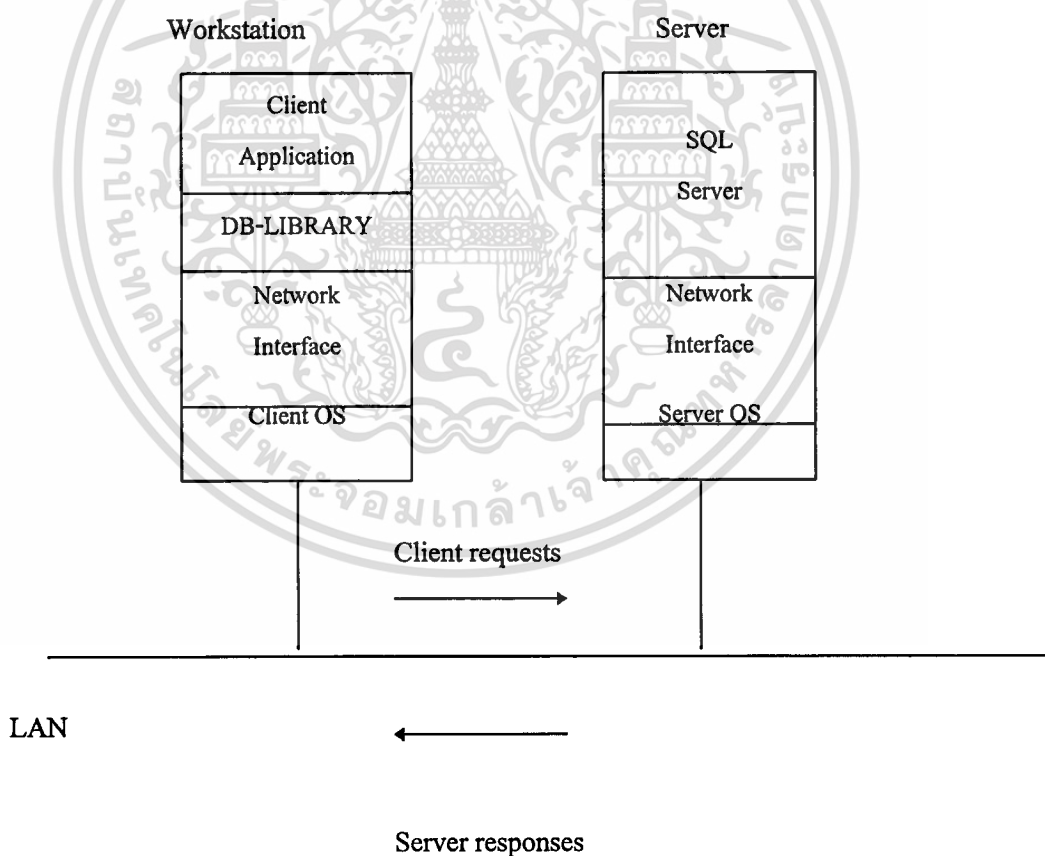
## SQL Server

### สถาปัตยกรรมของ SQL Server (Architectural Overview)

SQL Server คือ แหล่งบริการฐานข้อมูลแบบรีเลชันแนลในตัวโปรแกรม จะประกอบไปด้วยสถาปัตยกรรมของไคลเอ็นต์/เซิร์ฟเวอร์ในส่วนของ Server นั้นจะไม่ตอบสนองคำสั่งของ DBMS ที่เป็นมาตรฐาน ในส่วนของไคลเอ็นต์ ซึ่งถ้าการติดต่อกับผู้ใช้แล้วรูปแบบประยุกต์สามารถทำได้กับไคลเอ็นต์หลาย ๆ ชนิด ภาษาที่ใช้ติดต่อกันระหว่างไคลเอ็นต์และ SQL Server คือ TRANSACT-SQL ซึ่งก็คือ ภาษา SQL ที่ใช้ใน SQL Server

SQL Server เป็นการพัฒนาสำหรับ Online Transaction Processing (OLTP) คือ การรวมเป็นศูนย์กลางของผู้ใช้จำนวนมาก, ฐานข้อมูลและโปรแกรมประยุกต์ขนาดใหญ่

SQL Server สามารถใช้ในระบบ PC LAN และระบบ UNIX สำหรับ PC นั้นจะมี SQL Server สำหรับ OS/2, Windows NT และ SQL Server NLM สำหรับ PC UNIX คือ Sybase SQL Server



Client และ SQL Server

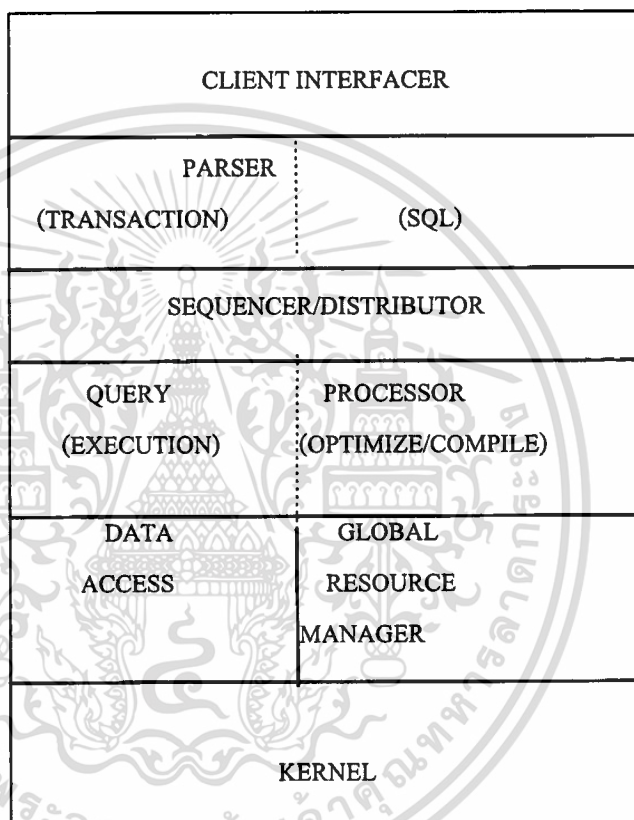
เอกสารนี้เป็นรูปแสดงถึงความสัมพันธ์ระหว่างโปรแกรมประยุกต์ของไคลเอ็นต์และ SQL Server เอกสารนี้เป็นเอกสารที่เผยแพร่โดยบริษัทผู้ผลิตเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สถาปัตยกรรมของเซิร์ฟเวอร์ (Server Architecture)

SQL Server ใช้สถาปัตยกรรมแบบ Single-process และ Multithreaded ใน Multithreaded นั้น คือ Single-process ของแต่ละ User ในระบบ Multiuser ที่มีการ Context Switching ที่ OS และสถาปัตยกรรมแบบ Multithreaded นี้ ยังเป็นการใช้หน่วยความจำอย่างมีประสิทธิภาพ

## ส่วนประกอบของเซิร์ฟเวอร์ (Server Components)

SQL Server ประกอบไปด้วยหลาย ๆ ส่วน ดังรูป



รูปแสดงส่วนประกอบของ SQL Server

## สถาปัตยกรรมของ SQL Server เบื้องต้น

1. Client interface
2. Sequencer/distributor
3. Query processor
4. Data management
5. Global resource manager
6. Kernel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Kernel** ที่แกนกลางของ SQL Server ก็คือ kernel ซึ่งทำหน้าที่หลาย ๆ อย่าง โดยทั่วไปจะตอบสนองกับระบบปฏิบัติการ kernel เป็นตัวกำหนดการทำงานและเรียงลำดับงานของงานต่อไปนี้ คือ context swiching, จัดการ buffer, disk cache, จัดการ I/O และคอมไพล์คำสั่ง queries

ใน SQL Server NT user task คือ การรวบรวม thread ของ server และ thread ที่มีการเปลี่ยนแปลงนั้นใช้ทำ schedule โดยแลกเปลี่ยนกับ function ของ OS ด้วยตัวเอง

SQL Server สามารถที่จะกำหนดลำดับความสำคัญของงานได้ มีความเสถียรภาพมากในสภาพแวดล้อมที่แตกต่างกัน

**Data Management and Global Resource Manager** จัดการข้อมูลและทรัพยากร ในส่วนนี้จะจัดการในการเก็บข้อมูล ค้นคืนข้อมูลจากฐานข้อมูลบนอินเด็กซ์และตารางใน query และจัดการ transaction รวมทั้งจัดการ transaction log เพื่อที่จะเรียกคืนได้เมื่อระบบหรือสื่อเกิดเสียหาย การจัดการทรัพยากรทั้งหมดจะทำกับหลาย ๆ งานที่อยู่ในบัฟเฟอร์ และที่เก็บข้อมูล เช่น จัดการเกี่ยวกับการ lock, จัดการ index, เรียงข้อมูล, ค้นคืนข้อมูล

**Query Processor** จะประกอบไปด้วย 2 ส่วนคือ optimizer และ execution

- Query optimizer ซึ่งอยู่ในรูปของการค้นคืนหรือแก้ไข ในส่วนนี้จำเป็นจะต้องมีในทุก DBMS ซึ่งจะให้ภาษา database เช่น SQL ในภาษา SQL นั้นจะต้องเจาะจงข้อมูลที่ต้องการค้นคืนหรือแก้ไข optimizer จะรับกลุ่มของ query tree และก็จะสร้างกลุ่มของ query plan ที่จะ execute โดย ส่วนของ execution

- Execution ในส่วนนี้จะตอบสนอง โดยการ execute optimal plan ที่ได้จากส่วน optimizer

Query optimizer ของ SQL Server จะให้ข้อมูลพื้นฐาน เพื่อคำนวณและประเมินสำหรับคำสั่งที่จะทำและยังให้ข้อมูลสถิติเกี่ยวกับค่าของ key

**Sequencer/Distributor**

Sequencer คือ ส่วนประกอบที่ตอบสนองการ execute แอปพลิเคชันที่ต้องการให้เหมาะสม โดยนำส่วนประกอบอื่น ๆ ที่เหมาะสมมาเรียงลำดับ sequencer จะยอมรับ sequence tree และเตรียมเพื่อ execute และจัดการการ execute จะกระทำกับหลาย ๆ ฟังก์ชัน ในขณะที่ execute รวมทั้งการ execute ใน store procedure, มองหาคำตอบ, รวบรวมแลกเปลี่ยน, ตรวจสอบและเรียกใช้ query processor เพื่อสร้าง plan

ส่วน distributor จะตอบสนองการกระจาย portions ของ SQL query databases แตกต่างกันใน Server เดียวกันหรือ Server อื่น

**Client Interface** ในส่วนนี้จะเป็นส่วนที่เชื่อมต่อแลกเปลี่ยนระหว่าง Client กับ Server ซึ่งจะมี 2 ส่วน คือ

SQL interface จะแลกเปลี่ยนกับทุก TRANSACT-SQL request interface ที่ใช้ข้อมูลและเรียกใช้ remote procedure ทั้งสองส่วนนี้จะยอมรับโดย Client request และจะเปลี่ยนคำสั่งเหล่านั้น

เป็นรูปแบบ internal form ซึ่งเรียกว่า sequence tree ที่ประกอบไปด้วย command tree และ procedural logic และในแต่ละ command tree สามารถประกอบไปด้วยหลาย ๆ query tree Client interface จะรวมทั้ง parser ที่ทำการวิเคราะห์ SQLstatement เพื่อตรวจสอบว่าคำสั่งนั้นถูกต้องหรือไม่

### ส่วนประกอบของไคลเอนต์ (Client Components)

Client application จะประกอบไปด้วยหลาย ๆ ส่วน SQL Server สามารถพัฒนาโปรแกรม client ได้หลาย ๆ platform การ interface สำหรับการเขียนโปรแกรม SQL Server จะให้ DB-LIBRARY

DB-LIBRARY คือ ระดับการเรียกใช้ interface นั่นคือ สามารถพัฒนาโปรแกรมแอปพลิเคชัน โดยใช้ภาษาโปรแกรมที่แตกต่างกันได้ เช่น C, PASCAL, COBOL โดยใช้ DB-LIBRARY

#### isql

isql เป็นแอปพลิเคชันที่ใช้ command-line เป็นตัวเชื่อมกับ SQL Server และยอมให้ผู้ใช้สามารถส่ง TRANSACT-SQL ไปยัง Server และดูผลลัพธ์ที่ได้ ซึ่งจะไม่มีการแก้ไข จัดรูปแบบ หรือ scrolling ถ้าเกิดกีย์พิมพ์จะให้คีย์ใหม่ให้ถูกต้อง ถ้า column หรือ rows ที่ได้กลับใหญ่หรือกว้างกว่าหน้าจอก็จะไปขึ้นบรรทัดใหม่

Server Administration Facility (SAF) คือ ffront-end client application อื่น ๆ ที่ได้จาก SQL Server จะเป็นรูปแบบของวินโดวส์, เมนู ง่ายต่อการใช้งาน สำหรับการควบคุม server ซึ่งต้องใช้ภาษา C และใช้ DB-LIBRARY interface ซึ่งอาจจะสามารถทำงานบน DDS หรือ OS/2 SAF จะเป็นเมนูสำหรับคำสั่งควบคุมและจัดการงาน เช่น การ log on ของผู้ใช้, เพิ่ม user หรือ backup ข้อมูล

SQL Administrator and SQL Object Manager คือ เครื่องมือที่ใช้ในงานควบคุมดูแลระบบ เช่น สร้าง device และ database กำหนด user และกลุ่ม user และจัดการ database โดยใช้รูปภาพในการ interface

Client Application อื่น ๆ โปรแกรมประยุกต์อื่นที่พัฒนาโดยอินเทอร์เฟสผ่าน DB-LIBRARY ที่สามารถติดต่อกับ server ได้ เช่น

#### End-User and เครื่องมือช่วยในการทำรายงาน

- Excel
- Lotus
- Forest and Trees
- SQR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**เครื่องมือช่วยในการพัฒนาระบบ**

- Object/1
- Visual Basic
- PowerBuilder
- SQL Windows

**Groupware**

- Saros Fileshare
- MS Electronic Mail Server (EMS)
- MS System Management Server (SMS)

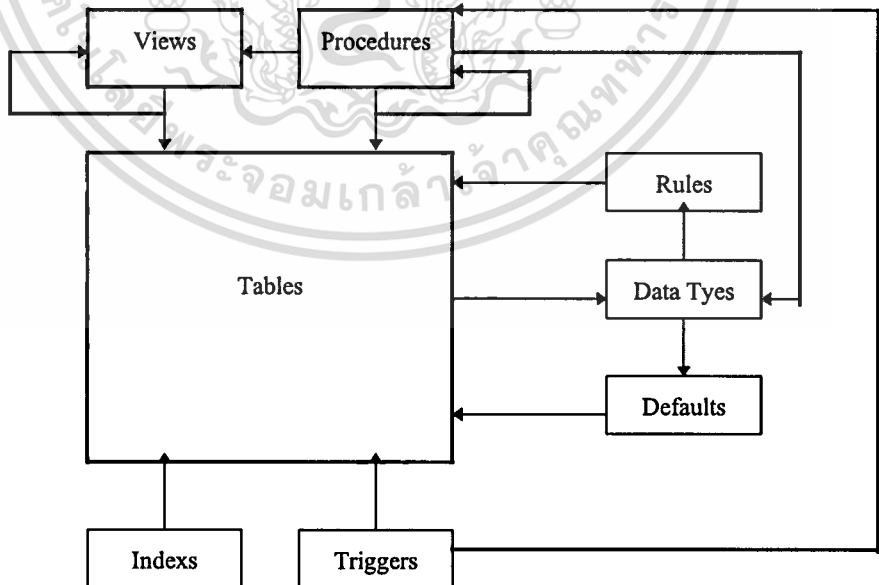
**Vertical Solutions**

- Peoplesoft HR
- Platinum Software

**Databases in SQL Server**

ทุก SQL Server ที่ติดตั้งแล้วสามารถมีได้หลาย ๆ database แต่ในความเป็นจริงจะจำกัดที่ทรัพยากรที่มีอยู่ใน SQL Server จะมี data base พื้นฐานอยู่ 2 ชนิด คือ

- System Databases จะถูกสร้างอยู่ที่ Server เวลาที่ติดตั้ง
- User Databases จะถูกสร้าง โดยผู้ใช้งาน



**รูปแสดง โครงสร้างของ SQL Server**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ระบบฐานข้อมูล (System Database)

Three databases จะถูกสร้างขึ้นเมื่อติดตั้ง SQL Server เป็น database ที่จะต้องใช้โดย Server สำหรับระบบของ database มาเรียกใช้ ประกอบไปด้วย databases ดังนี้

- master database (master)
- model database (model)
- temporary database (tempdb)

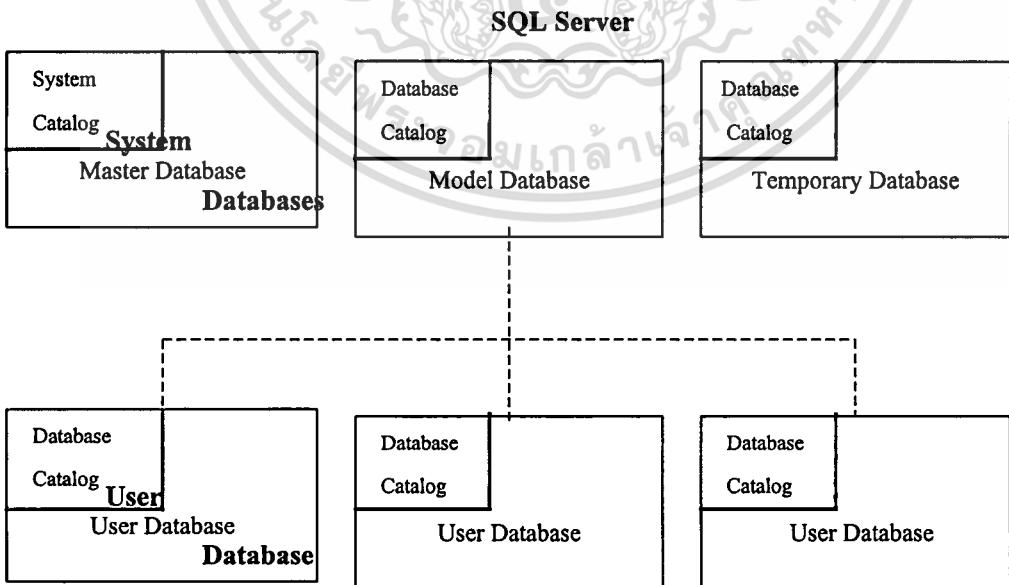
และอาจจะมี database ที่ 4 เพิ่มขึ้นมา เช่น pubs database แต่จะไม่ใช่ system database

**master database** ใช้เก็บข้อมูลข่าวสารเกี่ยวกับ Server database ทั้ง system และ user และควบคุมการทำงานของ Server เช่น login Id และ password, active process ชื่อและตำแหน่งของ database ข้อมูลเหล่านี้จะถูกเก็บในตารางที่เรียกว่า system tables ซึ่งเป็นส่วนหนึ่งของ data dictionary ของ Server master database ที่จะมี login ID เฉพาะ คือ Sa หรือ SA สำหรับผู้ควบคุมดูแลระบบ

**model database** เป็นตัวแบบสำหรับการสร้าง user database อื่น ๆ database ที่สร้างขึ้นใหม่ จะคัดลอกมาจาก model ซึ่งขนาดจะสามารถเปลี่ยนแปลงได้ model database สามารถแก้ไขได้ทั้งขนาดและรายละเอียด สำหรับการสร้าง user database ใหม่

**temporary database** จะใช้เก็บข้อมูลชั่วคราวที่ถูกสร้างขึ้นเนื่องจากการทำงานของ Server หรือว่าจากการสร้างโดยผู้ใช้งาน temporary database จะถูกใช้ร่วมกับทุก ๆ database บน server

จากรูปจะแสดงความแตกต่างของแต่ละ database



รูปแสดง System และ User database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## User Database

งานเบื้องต้นของ Server คือ จัดการกับ User Database จากมุมมองของข้อมูล และจัดการข้อมูล User database ถูกสร้างโดยผู้ใช้ด้วยคำสั่ง CREATE DATABASE ผู้ที่สร้างจะต้องเป็นคนที่ได้สิทธิจากผู้ควบคุมระบบ ทุก User Database จะถูกสร้างจาก master database การสร้าง database จะเป็นการจัดสรรเนื้อที่เก็บข้อมูลและข้อจำกัดของการทำงาน user database ประกอบไปด้วยกลุ่มของ system table ซึ่งเป็นส่วนหนึ่งของ master system table จะเก็บ login ID และ Password สำหรับ user โดยเฉพาะ คือ dbo ซึ่งสามารถกำหนดสิทธิของ user แต่ละรูปแบบได้ การ drop database หรือลบ database ออกจากระบบ database จะถูก drop ด้วยคำสั่ง DROP DATABASE

## พจนานุกรมข้อมูล (Data Dictionary)

SQL Server ก็เหมือน ๆ กับ DBMS อื่น ๆ ซึ่งมี data dictionary หรือ catalog ซึ่ง catalog นี้จะประกอบไปด้วย system table หลาย ๆ ตัว ทุก ๆ system database และ user database บน server จะมีกลุ่มของ system table 13 table ที่ใช้เก็บข้อมูลเฉพาะ ข้อมูลนั้นเกี่ยวกับ database เช่น user, สิทธิของ user กลุ่มของ system table นี้ จะเรียกว่า database catalog

master database จะประกอบไปด้วย database catalog table 13 table ที่ใช้เก็บข้อมูลของ Server เช่น login Id และ Password, อุปกรณ์และ server database อื่น ๆ กลุ่มของ system table เหล่านี้เรียกว่า system catalog

## ตารางแสดงรายละเอียดของ database catalog

System table	รายละเอียด
sysalternates	เก็บ server login ID และ map กับ database user
syscolumns	เป็นข้อมูลเกี่ยวกับทุก columns ของตารางหรือ views, parameters ใน stored
syscomments	procedures
sysdepends	กำหนดรูปแบบของ views,rules,defaults,trigger, และ stored procedures
sysindexes	ขึ้นอยู่กับระหว่าง stored procedures, views, triggers และ stored อื่น ๆ
syskeys	รายละเอียดของ clustered, nonclustered index และตารางที่ไม่มี indexes
syslogs	รายละเอียดของ primary และ foreign keys
sysobjects	Transaction logรายละเอียดของ Tables,views,rules,defaults,triggers,procedures,log
	temporary objects
sysprocedures	ข้อกำหนดของ Stores procedures,triggers,views,rules และ defaults
sysprotects	ข้อมูลเกี่ยวกับสิทธิของผู้ใช้ database
syssegments	รายละเอียดของ segments
systypes	รายละเอียดของระบบและข้อกำหนดของผู้ใช้ที่กำหนดเอง
sysusers	รายละเอียดของ database user

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้วงใน ใต้เงื่อนไขใด ๆ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รายละเอียดของ system catalog

System table	รายละเอียด
syscharsets	รายละเอียดของกลุ่มตัวอักษรและวิธีการเรียงข้อมูล
sysconfigures	รายละเอียดของ user-settable configuration parameters
syscurconfigs	ข้อมูลปัจจุบันของ configuration parameters ที่กำลังใช้อยู่
sysdatabases	รายละเอียดของ database แต่ละตัวบน server
sysdevices	รายละเอียดของ database devices และ dump devices สำหรับ server
syslanguages	รายละเอียดของภาษาเข้าไบบน server (ภาษาอื่นที่ไม่ใช่อเมริกัน)
syslocks	รายละเอียดของการ lock
syslogins	รายละเอียดคนบน server login account
sysmessages	รายละเอียดคนบน server ข้อความแสดงความผิดพลาดของระบบและเตือน
sysprocesses	รายละเอียดของ active processes
sysremotelogins	รายละเอียดคนบน remote users authorized ไปยัง access server
sysservers	รายละเอียดคนบน remote server ซึ่งรวมทั้ง remote stored procedure
sysusages	รายละเอียดของ disk ที่กำหนดให้ database

### ชนิดของข้อมูล (Data types)

#### Stores table

Store_id	Stor_name	Stor_address	City	State	Zip
Char(4)	Varchar(40)	Varchar(40)	Varchar(20)	Char(2)	Char(5)
not null	null	null	null	null	null
7066	Barnum's	567 Pasadena Ave.	Tustin	CA	92789
7067	News and Brews	577 First St.	Los Gatos	CA	96745
7131	Doc-U-Mat: Quality Laundry and Books	24-A Avrogado Way	Remulade	WA	98014
8042	Bookbeat	679 Carson St.	Portland	OR	98076
6380	Eric the Read Books	788 Catamaugus Ave.	Seattle	WA	98056
7896	Fricative Bookship	89 Madison Ave.	Fremont	CA	90019

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ชนิดของข้อมูล (Data type)

data type ของ Column แสดงให้เห็นชนิดของข้อมูลใน Column เช่น Column Store-id มี data type เป็น char(4) สามารถเก็บตัวอักษรของข้อมูล 4 ตัว datatype ของ Column จะถูกกำหนดโดยผู้ใช้ในส่วนของ table definition

### Nulls

ค่า nulls คือ ข้อมูลที่ไม่รู้ค่าซึ่งแตกต่างจากค่าอื่น ๆ ที่เป็นไปได้ เช่น ถู่มือส่วนตัว จะมีหมายเลขโทรศัพท์เป็น phone\_num สำหรับคนที่ไม่มีหมายเลขโทรศัพท์จะต้องเป็นค่า null ซึ่งอาจจะแทนด้วย 000-0000 SQL server สนับสนุน true null values และยอมให้ผู้ใช้กำหนดค่า null values เองได้

### User-defined data type

SQL server สนับสนุนข้อมูลที่ใช้สามารถกำหนดขึ้นเอง

### Rules and Defaults

rule คือ วัตถุประสงค์ที่ให้บริการซึ่งจะจำกัดโดยกฎของข้อมูลที่กำหนดโดย datatype rule จะใช้เป็นกฎข้อบังคับของระบบ เช่น เงินเดือนของพนักงานจะต้องมากกว่า 5,000 บาท

defaults ยอมให้ผู้ใช้กำหนดข้อมูลที่ server โดยอัตโนมัติในการทำ insert column

### Index

คือ จุดประสงค์ที่จะทำให้เข้าถึงข้อมูลใน table อย่างรวดเร็ว SQL server ใช้ Btree เป็นโครงสร้างของ index key value สำหรับ index จะถูกเก็บไว้ใน index pages และ row ของข้อมูลจะถูกเก็บไว้ใน Data page SQL server ใน 1 page จะมี 2 k-byte ของ data ที่อ่านจาก disk

Index สามารถทำเป็นแบบ Dense คือทุกคีย์จะถูกแสดงใน index หรือ nondense หมายถึงเลือกแต่คีย์ที่แน่นอนเท่านั้นของแต่ละ page SQL server ยอมให้ผู้ใช้สามารถกำหนด cluster index และ non-cluster index

### Stored Procedures

SQL server สามารถสร้าง store procedures ซึ่งก็คือ ชื่อของ precompiled กลุ่มของ TRANSACT-SQL compile จาก data dictionary จากนั้นบน procedure สามารถ execute ง่าย ๆ โดยชื่อที่ compile จากการเรียกมาจาก dictionary และ execute การสร้าง store procedures ทำได้โดยใช้คำสั่ง CREATE PROCEDURE

### Triggers

คือ ชนิดของ stored procedures ที่ใช้เฉพาะงาน ซึ่งจะทำงานโดยอัตโนมัติเมื่อมีการเรียกใช้เกิดขึ้น เหตุการณ์ที่จะเกิดขึ้น โดยทั่วไปจะได้จากการแก้ไขตารางการ insert, update และ delete ข้อมูล

## Data Integrity และ Concurrency Control

ในระบบที่ใช้คนเดียวผู้ใช้สามารถอ่านและแก้ไขข้อมูลได้โดยไม่ต้องสนใจว่าข้อมูลนั้นจะถูกเปลี่ยนแปลงโดยผู้อื่นในเวลาเดียวกัน แต่ในระบบที่มีผู้ใช้ได้หลายคน(Multi User) ผู้ใช้อาจจะใช้ข้อมูลพร้อมกันในเวลาเดียวกัน

เพราะฉะนั้นหน้าที่ของระบบ database ที่ใช้หลายคน ต้องทำให้แน่ใจว่า Transaction ทั้งหลายของผู้ใช้แต่ละคนจะต้องถูกต้องเมื่อมีการใช้งานพร้อมกัน

### Share และ Exclusive locks

- Share lock คือ ข้อมูลที่ถูก lock สามารถที่จะใช้ร่วมกับผู้อื่น หรือผู้อื่นอาจมาใช้ข้อมูลร่วมได้แต่ไม่สามารถแก้ไขข้อมูลนั้นได้

- Exclusive lock คือการ lock ไว้ใช้คนเดียวไม่ยอมให้ผู้อื่นมาใช้ร่วม exclusive lock จะใช้เมื่อมีการแก้ไขข้อมูลแต่ไม่ให้ผู้มาแก้ไข

- Update lock ใช้สำหรับ UPDATE และ DELETE ซึ่งมีข้อจำกัดน้อยกว่า Exclusive lock ใน SQL Server การ lock จะเป็นแบบ granularity ของ page หรือ ตาราง การ lock ที่ระดับ page จะใหญ่กว่าการ Lock ที่ระดับ row ในหนึ่ง page SQL server กำหนดเนื้อที่บน disk 2 Kbyte ซึ่งประกอบไปด้วยหลายๆ Row การ lock page จะมี Over head น้อยกว่าการ lock row เมื่อจำนวน lock page ใน table มีมากขึ้นก็อาจจะทำการ lock table ก็ได้

### Repeatable Read Consistency

ใน SQL server จะมีการแก้ปัญหา repeatable read consistency โดยใช้ HOLDLOCK ใช้กับตาราง หรือ VIEW ซึ่งจะทำ share lock และจะ lock จนกระทั่ง Transaction ทำงานเสร็จ

### Dead lock และ Live lock

dead lock เกิดขึ้นเมื่อ 2 process พยายามที่จะใช้ทรัพยากร ที่ ถูก lock จาก process อื่นไว้ SQL server จะมีการตรวจสอบการเกิด Dead lock และทำการแก้ปัญหาโดยอัตโนมัติ โดยจะเลือก process หนึ่งทำให้เกิด Dead lock และทำการ กำจัด process นั้น process นั้นจะ roll back และก็จะปล่อย lock ซึ่ง process ที่จะเลือกนั้นจะต้องเป็นตัวที่มี CPU Time น้อยที่สุด

การเกิด dead lock ที่เกิดจาก share lock และ exclusive lock เรียกว่า Live lock หรือ starvation ซึ่งเป็นไปได้ที่ Share lock lock page อยู่แล้ว Exclusive lock ต้องการที่จะ lock page เดียวกัน ก็จะต้องรอให้ share lock ปลด lock ก่อน แต่ว่า share lock เองก็รอให้ Exclusive ปลด lock ทำให้เกิด Live lock

SQL Server จะทำการตรวจสอบและแก้ปัญหาโดยใช้ Demand Lock หลังจาก share lock ที่ 4 lock อยู่ เมื่อ exclusive lock ต้องการจะ lock SQL server จะคืน exclusive ไปยัง demand lock จนกว่า share lock จะปลด lock ถึงจะเข้า lock ได้

## สโตร์โพรซีเจอร์(Stored Procedures)

สโตร์โพรซีเจอร์เป็นชื่อของกลุ่มพริคอมไพล์เลอร์ของ TRANSACT SQL ซึ่งจะทำให้การเอ็กซ์ซิคิวต์ได้ง่ายโดยการเรียงชื่อของสโตร์โพรซีเจอร์

## การสร้างและลบสโตร์โพรซีเจอร์(Creating and Dropping Stored Procedures)

การสร้างสโตร์โพรซีเจอร์ทำได้โดยใช้คำสั่ง CREATE PROCEDURE และการลบทำได้โดยใช้คำสั่ง DROP PROCEDURE

หนึ่งโพรซีเจอร์ประกอบไปด้วย ชื่อโพรซีเจอร์, พารามิเตอร์และส่วนของ SQL Statement

## รูปแบบ

```
CREATE PROCEDURE[<owner>]<procedure>[;<number>]
    [((<parameter><data-type>[=<default>]
    [<parameter><data-type>[=<default>]]--D)]
    [WITH RECOMPILE] AS
    <sql-statement>
```

### ตัวอย่างที่ 1

```
create proc sp-users as
select users=names from sysusers
output
```

ถ้าต้องการให้มีพารามิเตอร์ สามารถทำได้ ดังนี้

### ตัวอย่างที่ 2

```
create procedure author_info @last_name varchar(30) as
select au_fname, au_lname, state, phone, contract
from authors
where au_lname = @last_name
```

Output :

@last\_name คือ พารามิเตอร์เป็นแบบตัวอักษร ขนาด 30 ตัวอักษร

### ตัวอย่างที่ 3

```

create procedure titles_type_range
    @type varchar(15), @high money, @low money = 0 as
select title, type, price
from titles
where type = @type
and price between @high and !low

output :

```

### การลบสโตร์โพรซีเจอร์(Dropping Stored Procedures)

การลบสโตร์โพรซีเจอร์สามารถทำได้โดยใช้คำสั่ง DROP PROCEDURE

รูปแบบ : **DROP PROCEDURE [<owner>.]<procedure>**  
**[, [<owner>.]<procedure>]\_\_**

### ตัวอย่าง

```
drop procedure author_info
```

## Object Linking and Embedding (OLE)

OLE (ออกเสียงว่า “Oh-Lay”) จะมองเป็น Object และ Component ที่ใช้งานร่วมกัน OLE เป็นภาษาอิสระที่อยู่ในรูปของ Binary

OLE จะ execute บน Windows platform และ Macintosh platform

OLE based on Component Object Model (COM) โดย COM ถือเป็นหัวใจของ OLE ซึ่งจะใช้สร้าง Services ต่างๆ และ OLE เป็น Client/Server model โดย

- Client ได้แก่ Object user
- Server ได้แก่ Object implementer

COM จะมีมาตรฐานสำหรับการติดต่อสื่อสารระหว่าง Objects และสามารถสร้าง Object ได้จากหลายภาษา COM จะกำหนด Object interface ,วิธีสร้าง Object และวิธีทำลาย Object รวมถึงการจัดการข้อผิดพลาดและการทำ Mememory management

### In-Process กับ Out-Of-Process

เมื่อ OLE object 2 objects ติดต่อกัน ฝ่าย Client ก็จะส่งสัญญาณ Request ไปที่ ฝ่าย Server แล้ว Server ก็จะทำหน้าที่ 1 ใน 2 อย่างคือ

- In-Process server จะ run ที่ Client
  - Out-Of-Process server จะ run ที่ Server
- การนำไป implement สามารถสรุปได้ดังนี้

<i>Server Type</i>	<i>OLE Client</i>	<i>OLE Server</i>
In-Process	EXE or DLL	DLL
Out-Of-Process	EXE or DLL	EXE

### OLE Services

#### 1. Marshaling

เป็น Process ที่ใช้สำหรับการติดต่อสื่อสารของ OLE object ประกอบด้วย -Proxy ซึ่งอยู่ทาง Client ทำหน้าที่นำ Function call และ Parameters ที่อยู่ ในรูปแบบมาตรฐานของ OLE ส่งไปให้ทาง Server

-Stub ซึ่งอยู่ทาง Server จะทำหน้าที่รับ Function call และ Parameters ที่ส่งมาจากทาง Client มา convert ให้อยู่ในรูปแบบที่ Server สามารถเข้าใจได้

#### 2. Structured Storage

ให้ OLE เก็บ Documents ที่บรรจุ Documents อื่นๆ หรือเรียกว่า Compound document (Compound file) ซึ่งประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Stream คือ ข้อมูลที่ไม่มีโครงสร้าง

- Storage คล้ายๆ Directory ซึ่งคอยจัดการและจัดแจง Streams และ Storages ให้เป็น โครงสร้างแบบ Hierarchy

### 3. Moniker

ใช้อ้างอิง Document ที่ link อยู่ใน Compound document

### 4. Uniform Data Transfer (UDT)

เป็นกลไกในการส่งผ่านข้อมูลจาก Client ไปยัง Server

### 5. Drag and Drop

เป็นการ cut และ paste พร้อมกันใน 1 step

### 6. Compound document processing

เป็นการรวมนำเอา Documents จากหลายๆ Application มารวมกันเป็น Compound document

### 7. Linking

เป็นการเชื่อมต่อ Client object เพื่อให้สามารถอ้างอิงเป็น Server document object ได้

### 8. Embedding

เป็นการซ่อนไฟล์หลายๆ ไฟล์ อยู่ในไฟล์เดียว เรียกว่า Compound file โดยถ้าจะแก้ไข Embedded document จะต้องออกจาก Container document ก่อน

### 9. In-Place Activation

เป็นกระบวนการที่ยอมให้ User แก้ไข Embedded document โดยไม่ต้องออกจาก Container application ก่อน

### 10. OLE Automation Server (OA)

เป็นกลไกในการที่ Application หนึ่ง control หรือ automate อีก Application หนึ่ง มีชื่อที่ใช้ใน Client และ Server ที่มีความหมายเหมือนกับคำว่า OLE Automation Server ได้แก่

- Client - OLE automation client
  - Controlling application
  - OLE automation controller
- OLE client component
- Server - OLE automation server
  - OLE object application

การที่จะ manipulate กับ OLE automation server นั้นต้อง manipulate กับ OLE automation

interface ซึ่งประกอบด้วย เอกสารนี้เขียนขึ้นเพื่อให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ลักษณะของ Object เช่น Font
  - Method Operations ที่ Object สามารถกระทำได้ เช่น print
  - Property Member object - Object ที่ขึ้นกับ Object อื่น
  - Collection - เป็น Array ของ Member object
- OLE automation server สามารถ run ได้ทั้งแบบ
- Local คือ run ที่ Machine ที่เป็น Client
  - Remote คือ run ที่ Machine อื่นๆ ที่ไม่ใช่ Client

## 11. OLE Control (OCX)

เป็นรูปแบบพิเศษของ OLE automation object

OLE control ทำหน้าที่

- วาดโดยตรงที่ Client application
- Support การกำหนด Property หลายๆ อย่างได้อย่างรวดเร็ว
- Detect licensinf in a design versus runtime environment
- Support an extended interface that allows it to fire events
- Support data “binding”

OLE control นั้นจะทำงานใน Client application และ support การทำงานแบบ Outside-in activation

OLE control สามารถ implement ได้เป็นแบบ DLL เท่านั้น

## 12. Distributed Component Object Model (DCOM)

เป็นการติดต่อสื่อสารของ OLE ผ่านระบบเครือข่าย บางครั้งเรียกว่า “Distributed OLE”

## 13. Nile

OLE สำหรับ Database

## รีโมทดาต้าออบเจกต์ (Remote Data Objects) (RDO)

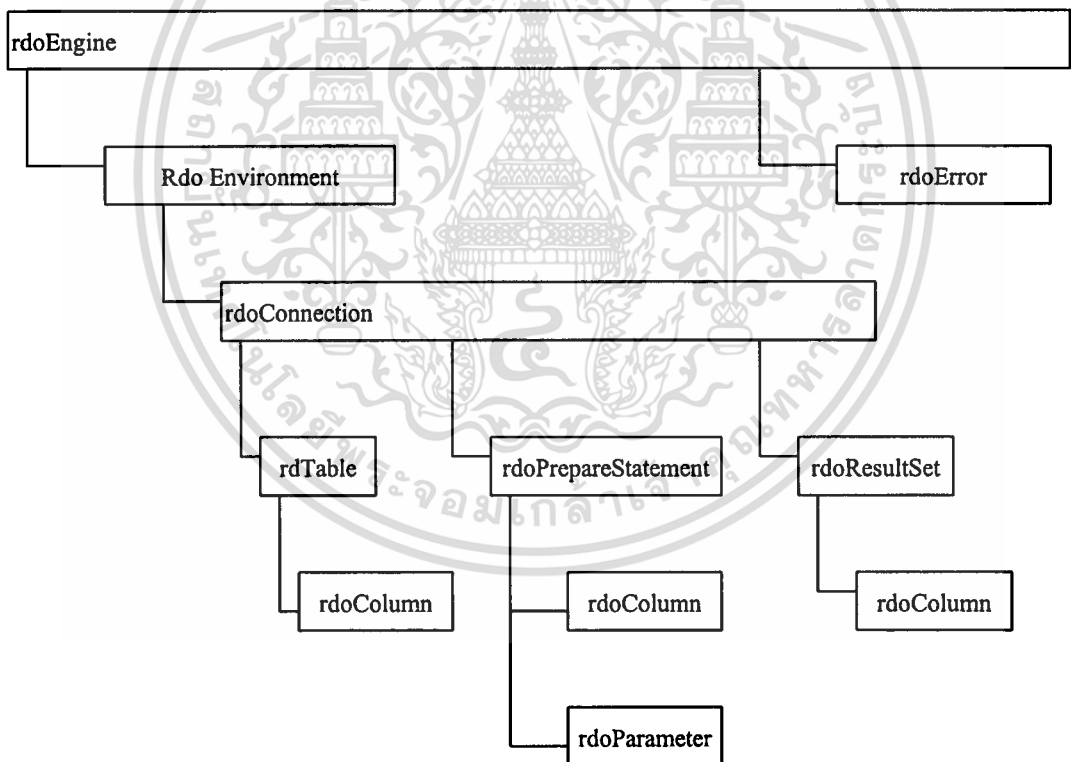
วิซวลเบสิก 4 ได้แนะนำวิธีการเข้าถึงข้อมูลที่มีประสิทธิภาพและสามารถสนับสนุน ODBC ของดาต้าเบสหลาย ๆ ชนิด และง่ายต่อการเรียนรู้ในการเชื่อมต่อออบเจกต์

RDO คือ กลุ่มของออบเจกต์ที่สามารถให้แอปพลิเคชันเข้าถึงข้อมูล RDR ถูกนำมาใช้งานที่ thin layer ที่อยู่บนสุดของ ODBC API

### คุณลักษณะของ RDO

- Access to ODBC data Sources
- สนับสนุนสำหรับฟังก์ชันเฉพาะของ Server
- อะซิงโครนัสกับการทำ Query
- ใช้หน่วยความจำไม่มาก
- เข้าถึงข้อมูลอย่างมีระเบียบ

### RDO Object Model



รูปแสดง RDO Object Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## rdoEngine

rdoEngine object แสดง remote data source ออบเจกต์นี้จะถูกสร้างขึ้นเองโดยอัตโนมัติ เมื่อมีการอ้างอิง RDO เพราะ rdoEngine เป็นระดับบนสุดของออบเจกต์ rdoEngine ใช้กำหนดค่า rdoEnvironment

### ตารางแสดง Properties ของ rdoEngine Object

Property	อธิบาย
rdoDefaultCursorDriver	ชนิดของเคอร์เซอร์ที่สร้าง : ODBC,server-side
rdoDefaultErrorTheshold	ค่าของ Errorthreshold property ใช้โดย rdoPreparedStatement Objects
rdoDefaultLoginTimeout	จำนวนค่าที่ ODBC รอที่พยายามจะ connect
rdoDefaultPassword	datasource
rdoDefaultUser	Password ที่ใช้เมื่อสร้าง rdoEnvironment ขึ้นใหม่
rdoVersion	UserId ใช้เมื่อสร้าง rdoEnvironment ขึ้นใหม่ 3 ตัวอักษร RDO library version number

### ตาราง Methods ของ rdoEngine Object

Method	อธิบาย
rdoCreateEnvironment	สร้าง new connection remote datasource และ rdoEnvironment object
rdoRegisterDatasource	เก็บข้อมูลการติดต่อไว้ที่ WindowRegister

เมื่อ rdo Engine object ถูกสร้าง มันจะสร้าง default environment, rdoEnvironment (0) โดยใช้ rdoDefault parameter เมื่อโปรแกรมต้องการจะเปลี่ยนแปลงก็ให้ Set ใน default properties ของ rob Environment (0) หรือ rdoEngine ก่อนที่จะสร้าง connection

ODBC data Source สามารถกำหนดให้โปรแกรมเรียกใช้ โดยใช้ rdoEngine โดยวิธี rdoRegisterDataSourcemethod นี้ ต้องการสร้าง data source ครั้งเดียวเท่านั้น

โดย rdoRegisterDataSource Method ต้องการ 4 อาร์กิวเมนต์

- dsName ชื่อที่ใช้ใน OpenConnection method ใช้อ้างอิงในการสร้าง dataSource
- drive ชื่อของ ODBC driver แต่ไม่ใช่ ODBC Dll file
- Silent ข้อมูล ชนิดบูลีนแสดง ODBC ไดอะล็อกบ็อก
- attributes ข้อความหรือ keyword ที่เพิ่มเติมใน ODBC.INI

เอกสารที่สงวนลิขสิทธิ์ของไมโครซอฟท์ จำกัด การใช้งานโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## rdoEnvironment

rdoEnvironment Object นี้แสดงกลุ่มของค่าตัวเบสคอนเนคชั่นเฉพาะ UserId และตอบสนองต่อ ODBC rdoEnvironment หนึ่ง object สามารถสนับสนุนได้หลาย database connection เพราะ ODBC สนับสนุนกับ environment เพียงหนึ่งเดียวกับแอปพลิเคชัน

เมื่อต้องการเชื่อมต่อหลาย ๆ connection โดยใช้ rdoEnvironment เดียว properties เช่น UserName และ Password จะถูกแชร์ระหว่าง connection

rdoEnvironment object จะถูกสร้างโดย rdoCreateEnvironment ซึ่งเป็น method ของ rdoEngine object

## ตัวอย่าง

```
Dim objEnv as rdoEnvironment
Set objEnv = rdoCreateEnvironment ("KUDB","Sa","database")
```

## ตาราง Properties ของ rdoEnvironment Object

Property	อธิบาย
Count	จำนวนของ rdoEnvironment object เกิดขึ้นใน Collection
CursorDriver	ชนิดของเคอร์เซอร์ที่สร้าง : ODBC หรือ server-side
hEv	ODBC environment handle ที่สามารถผ่าน ODBC API
LoginTimeOut	จำนวนของวินาทีที่ ODBC driver รอก่อนจะ time out
Name	ชื่อที่สัมพันธ์กับ rdoEnvironment
Password	password ใช้ในการสร้าง rdoEnvironment
UserName	User name ของ rdoEnvironment object user

## ตาราง Method ของ rdoEnvironment Object

Method	อธิบาย
BeginTrans	เริ่มต้นทรานแซคชั่น
Close	ปิด rdoEnvironment และเปิด rdoConnection, rollback
CommitTrans	Complete current Transaction Save และเปลี่ยนแปลง
Item	ให้ค่า rdoEnvironment collection ของ member ต้อง
OpenConnection	เปิดการ connection ไปยัง ODBC dataSource และให้ Object ของ data
RollbackTrans	Complete current transaction, ยกเลิกข้อมูลที่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OpenConnection เป็น method ที่ใช้ทำ connection กับ datasource method นี้จะมี 4 อาร์กิวเมนต์ ดังนี้

- dsName ชื่อของ register data source ที่เริ่มต้น connect
- prompt ชื่อกำหนดเมื่อ user ต้องการเพิ่มอาร์กิวเมนต์ผ่าน ODBC connect dialog กำหนด rdDriverNoPrompt user ให้เห็น ODBC dialog
- readonly ชื่อกำหนดของ user เพื่อจะ update ข้อมูล
- connect สตริงที่กำหนดพารามิเตอร์แก่ ODBC driver

### ตัวอย่าง

```
rdoEngin.rdoDefaultUser = UserId
rdoEngine.rdoDefaultPassword = Password
Set db = rdoEngine.rdoEnvironments (0) .openConnection ( _
    DATA_SOURCE,_
    rdDriverNoPrompt,_
    False)
```

### rdoConnection

rdoConnection object เป็นการเชื่อมต่อกับ data source แบบ physical

หนึ่ง rdoConnection object จะสร้าง result ของ rdoEnvironment.OpenConnection method

### ตาราง Properties ของ rdoConnection Object

Property	อธิบาย
AsyncCheckInterval	จำนวนเวลามิลิวินาทีที่ RDO คอยก่อนที่จะตรวจสอบสถานะสมบูรณ์ของ
Connect	asynchronous quere Read_Only string ที่ให้แก่ ODBC connect_string ค่านี้ใช้โดย
Count	OpenConnection method
hDbc	จำนวนของ rdoConnection object current ใน collection
Name	ODBC connection handle ตอบสนองกับ rdoConnection นี้
QueryTimeOut	ชื่อ Data Source ที่ใช้สำหรับการ connect
Row Affected	เวลาของ ODBC ที่รอเมื่อเกิด error ในการทำ query
StillExecuting	จำนวน row ที่เกิดจาก Execute method
Transactions	ข้อมูล Boolean แสดงเมื่อ query run จบ
Updatable	Boolean แสดงเมื่อ rdoConnection สนับสนุน transaction
Version	Boolean แสดงเมื่อ Object เปลี่ยนแปลงข้อมูลVersion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเผยแพร่ต่อหน่วยงานอื่นโดยไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับความยินยอม

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตาราง Method ของ rdoConnection Object

Method	อธิบาย
BeginTrans	เริ่ม Transaction ใหม่
Cancel	ยกเลิก query ที่ run ใน asynchronous mode
Close	หยุดการ connect
CommitTrans	จบ current transaction, save ข้อมูลที่เปลี่ยนแปลง
CreatePreparedStatement	สร้าง rdoPreparedStatement Object
Execute	สั่งให้ query หรือ SQL statement Run โดยไม่ต้อง Return Row
Item	Return สมาชิกของ rdoConnection Collection
OpenResultset	สร้าง rdoResultset object, ใช้โดย execute query และ SQL statement จะ return data
RollbackTrans	End current transaction, ยกเลิกข้อมูลที่เปลี่ยนแปลง

### การเข้าถึงข้อมูล(Accessing Data)

#### rdoTable

rdoTable object แสดงตารางหรือ View ใน database, rdoTables Collection จะรวบรวม

รวมข้อมูลบนทุก Table ที่มีใน database

### ตาราง Properties ของ rdoTable Object

Property	อธิบาย
Count	จำนวน rdoTable Object ใน Collection
Name	ชื่อ data Base Table
RowCount	จำนวน row ที่มีใน table
Type	String ที่แสดงชนิดของ table, view
Updatable	Boolean แสดง table เปลี่ยนแปลง, หรือไม่เปลี่ยนแปลง

### ตาราง Method ของ rdoTable Object

Property	อธิบาย
Item	Return จำนวนของ rdoTable Collection
OpenResultSet	สร้าง rdoResultset object ที่เก็บทุก row จาก rdoTable
Refresh	Retrieve list ของชื่อ table จาก database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## rdoResultset

เมื่อ query หรือ SQL Statement run และ return record จาก database record นี้ จะถูกเก็บใน rdoResult object

เมื่อสร้าง rdoResultset object สามารถใช้ argument ของ OpenResultset 4 แบบดังนี้

- Forward Only      Row ของ resultset สามารถ access และ update โดยขอบ pointer ที่เคลื่อนจากหน้าไปหลัง
- Static Type      Copy ของ row ใน resultset ใช้
- Keyset\_Type      Copy ของ row ใน resultset ใช้
- Keyset-Type      dynamic result set ที่ใช้ add, update หรือ delete row จาก table ความยาวจำกัด
- Dynamic-Type      dynamic resultset สามารถใช้ add,update หรือ delete row จาก table row ความยาวไม่จำกัด

## rdoColumn

rdoColumn Object แสดง column ของ database table

## ใช้ Stored Procedures

### - rdoPreparedStatement

rdoPreparedStatement object ใช้แสดง stored query defination หรือ พารามิเตอร์ ของ stored procedure รวมทั้ง SQL Statements

### - rdoParameter

rdoParameter Object ให้วิธีการที่ application สามารถกำหนด query parameter ก่อนที่จะทำการ query

## ตัวอย่าง

```
Set qryGetorders = db.CreatPrepare Statement ( _
    "SP_Get_Orders",_
    "{Call SP_Get_Orders (?,?)}")
qryGetOrders.rdoParameters(0).Value = CustomerId
qryGetOrders.rdoParameters(1).Value = ShippedFlag
Set rstGetOrders = qryGetOrders.OpenResultset CrdOpenForwardOnly,
    rdConcurReadOnly)
```

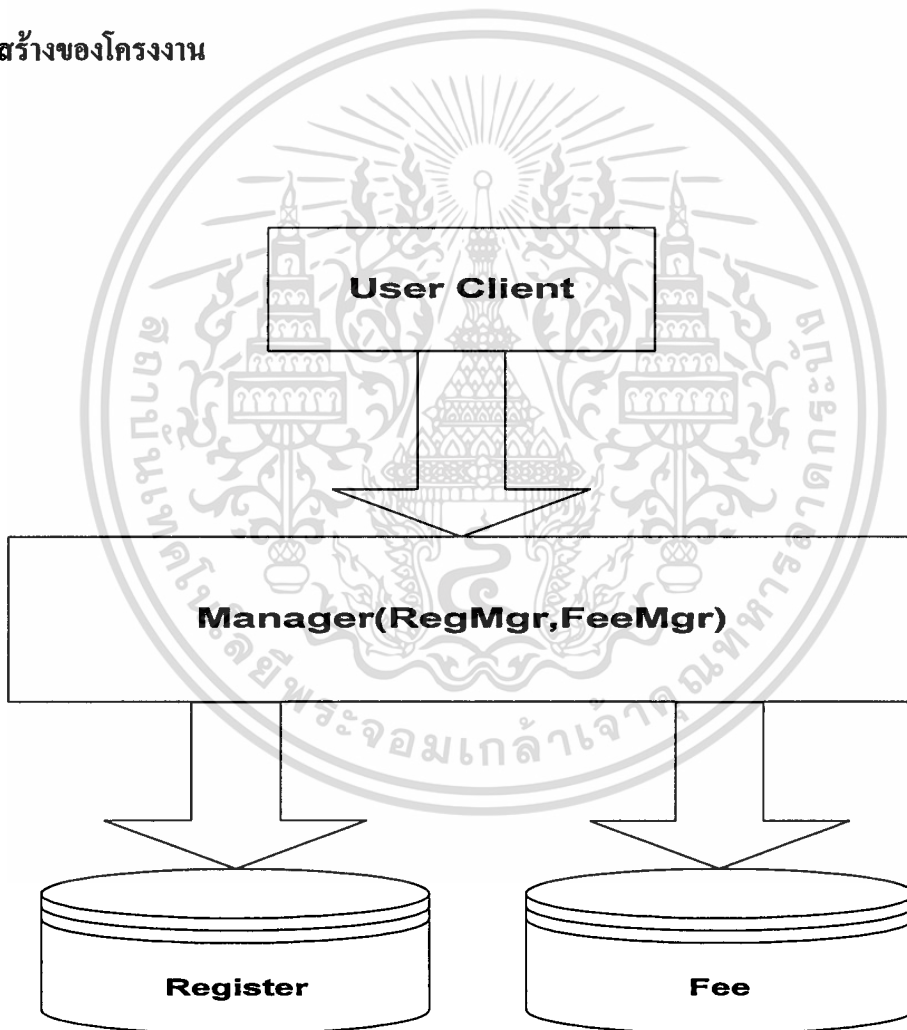
### บทที่ 3

#### การออกแบบและพัฒนาระบบ

##### ศึกษาระบบ

ทำการศึกษาระบบการทำงานของสำนักทะเบียนในส่วนของการลงทะเบียนและการจัดเก็บค่าธรรมเนียม, ขั้นตอนการลงทะเบียน, ข้อมูลที่ต้องการเก็บว่ามีอะไรบ้างจากนั้นนำข้อมูลที่ได้มาทำการวิเคราะห์และออกแบบโครงสร้างของโครงการและกำหนดขั้นตอนการทำงานต่าง ๆ ในโครงการนี้ นำวิธีการของออปเจ็กต์ โอเรียนเต็ล มาใช้และเขียนเป็น object Hirarchie

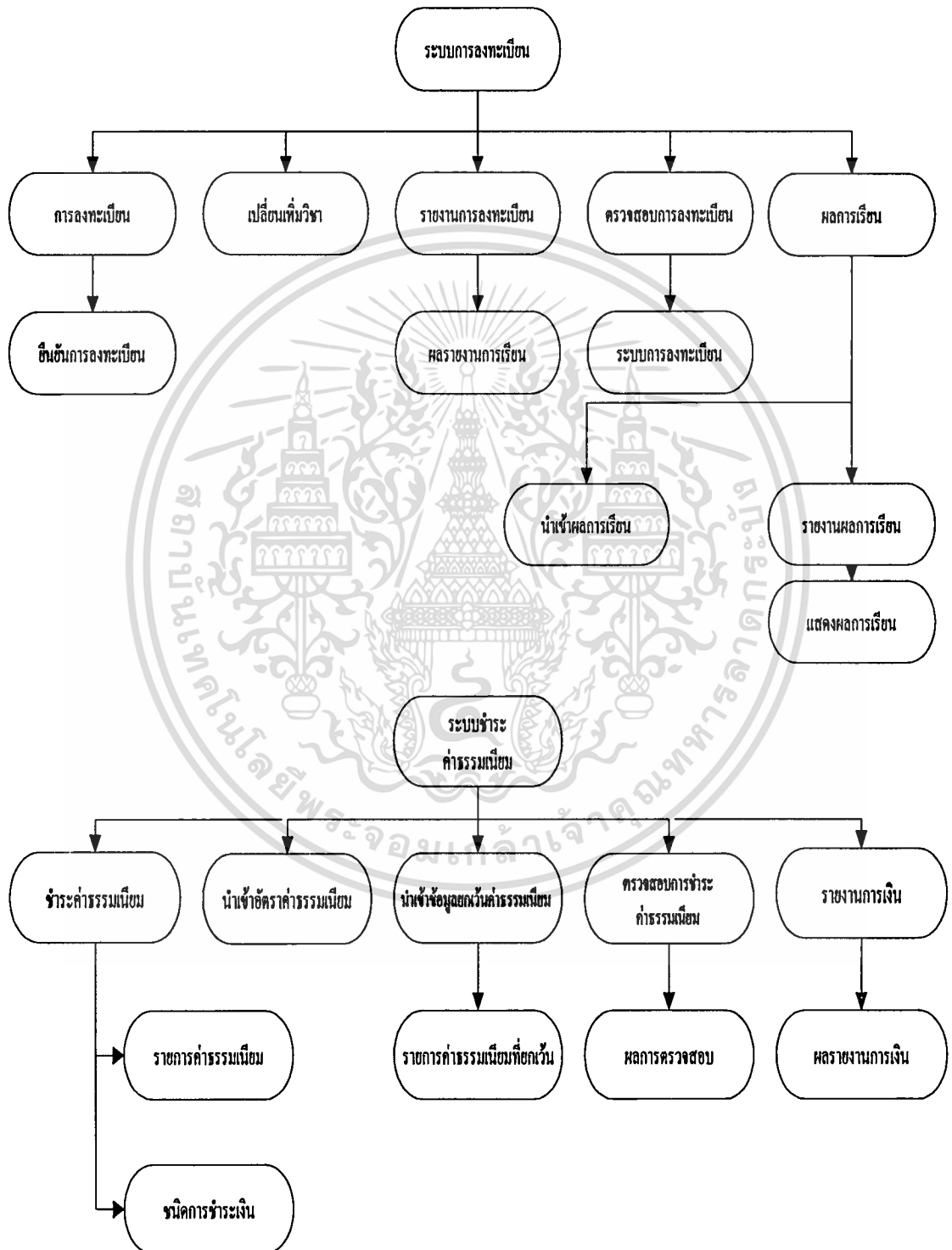
##### โครงสร้างของโครงการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Object Hirarchie โดยรวมของระบบสารสนเทศของสำนักทะเบียนและประมวลผล

### ARCHITECTING TIER 1 : CLIENT APPLICATIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รายละเอียดของแต่ละ Object ของส่วนการลงทะเบียน

การลงทะเบียน	วิชาที่ลงทะเบียน	ยืนยันการลงทะเบียน
รัศนศึกษา ชื่อ นามสกุล คณะ ภาควิชา สาขาวิชาเอก อาจารย์ที่ปรึกษา วิทยาเขต ปีการศึกษา ภาคเรียน วิชาที่ลงทะเบียน	รหัสวิชา ชื่อวิชา ประเภทการลงทะเบียน หมู่บรรยาย หมู่ปฏิบัติ	รัศนศึกษา ชื่อ นามสกุล คณะ ภาควิชา สาขาวิชาเอก อาจารย์ที่ปรึกษา วิทยาเขต ปีการศึกษา ภาคเรียน วิชาที่ลงทะเบียน
Loadข้อมูลนักศึกษา Loadข้อมูลวิชาที่ลงทะเบียน เพิ่มวิชาเข้า ลบวิชาออก ยืนยันการลงทะเบียน ยกเลิกข้อมูล ระบบการลงทะเบียน รายการหลัก		คำนวณหน่วยกิตรวม Updateข้อมูล ยกเลิกข้อมูล

## ผลการเรียน

ผลการเรียน
นำเข้าผลการเรียน
รายงานผลการเรียน
พิมพ์ใบแสดงผลการเรียน
พิมพ์ใบรับรองผลการเรียน
รายการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เปลี่ยนเพิ่มวิชา

<b>เปลี่ยนเพิ่มวิชา</b>
รหัสนักศึกษา
ชื่อ
นามสกุล
คณะ
ภาควิชา
สาขาวิชาเอก
อาจารย์ที่ปรึกษา
วิทยาเขต
ปีการศึกษา
ภาคเรียน
วิชาที่ลงทะเบียน
<b>เพิ่มวิชาเข้า</b>
ลบวิชาออก
Update ข้อมูล
ยกเลิกข้อมูล
ระบบการลงทะเบียน
รายการหลัก

<b>วิชาที่ลงทะเบียน</b>
รหัสวิชา
ชื่อวิชา
ประเภทการลงทะเบียน
หมู่บรรยาย
หมู่ปฏิบัติ

## รายงานการลงทะเบียน

<b>รายงานการลงทะเบียน</b>
ปีการศึกษา
ภาคเรียนที่
คณะ
ภาควิชา
ชนิดของการเรียงข้อมูล
<b>กำหนดคณะภาค, ชนิด</b>
<b>ค้นคืนข้อมูล</b>
แสดงผลรายงาน
ยกเลิก
ระบบลงทะเบียน
รายการหลัก

<b>ผลรายงานการลงทะเบียน</b>
ปีการศึกษา
ภาคเรียนที่
คณะ
ภาควิชา
รหัสนักศึกษา
ชื่อนักศึกษา
วิชาที่ลงทะเบียน
<b>พิมพ์ออกเครื่องพิมพ์</b>
ระบบการลงทะเบียน
รายการหลัก

<b>วิชาที่ลงทะเบียน</b>
รหัสวิชา
ชื่อวิชา
ประเภทการลงทะเบียน
หมู่บรรยาย
หมู่ปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตรวจสอบการลงทะเบียน

ตรวจสอบการลงทะเบียน
ปีการศึกษา
ภาคเรียนที่
คณะ
ภาควิชา
ชนิดการเรียงข้อมูล
กำหนดคณะภาค, ชนิด
ตรวจสอบ
แสดงผลการตรวจสอบ
ยกเลิกการตรวจสอบ
ระบบการลงทะเบียน
รายการหลัก

ผลรายงานการตรวจสอบ
ปีการศึกษา
ภาคเรียนที่
คณะ
ภาควิชา
รหัสนักศึกษา
ชื่อ-นามสกุล นักศึกษา
พิมพ์ออกเครื่องพิมพ์
ระบบการลงทะเบียน
รายการหลัก

## เข้าผลการเรียน

## รายงานผลการเรียน

นำเข้าผลการเรียน
คณะ
ภาควิชา
ปีการศึกษา
ภาคเรียนที่
รหัสวิชา
ชื่อวิชา
รหัสอาจารย์ผู้สอน
ชื่ออาจารย์ผู้สอน
รหัสนักศึกษา
รหัสนักศึกษา
หน่วยกิต
ผลการเรียน
Loadข้อมูล
นำข้อมูลเข้าเพิ่ม
Updateฐานข้อมูล
ยกเลิกข้อมูล
ระบบการลงทะเบียน
รายการหลัก

รายงานผลการเรียน
คณะ
ภาควิชา
ปีการศึกษา
ภาคเรียนที่
รหัสนักศึกษา
รหัสนักศึกษา
Loadข้อมูลคณะ,ภาค
เลือกนักศึกษาที่ต้องการ
Loadผลการเรียน
แสดงผลการเรียน
ระบบการลงทะเบียน
รายการหลัก

แสดงผลการเรียน
รหัสนักศึกษา
ชื่อ
สาขาวิชาเอก
ภาควิชา
คณะ
ปีการศึกษา
ภาคเรียนที่
รหัสวิชา
ชื่อวิชา
หน่วยกิต
เกรด
เกรดเฉลี่ย
เกรดเฉลี่ยสะสม
สถานะภาพ
คำนวณเกรดเฉลี่ย
พิมพ์ผลออกเครื่องพิมพ์
ระบบการลงทะเบียน
รายการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รายละเอียดของแต่ละ Object ส่วนชำระค่าธรรมเนียม

ชำระค่าธรรมเนียม
รหัสนักศึกษา
ชื่อนักศึกษา
สาขาวิชาเอก
ภาควิชา
คณะ
วิทยาเขต
ปีการศึกษา
ภาคเรียน
รายการค่าธรรมเนียม
การคิดค่าธรรมเนียม
ชนิดค่าธรรมเนียม
แสดงข้อมูลค่าธรรมเนียม
Updateข้อมูลค่าธรรมเนียม
ระบบชำระค่าธรรมเนียม

รายการค่าธรรมเนียม
รายการที่ 1
รายการที่ 2
รายการที่ 3
รายการที่ 4
รายการที่ 5
รายการที่ 6
รายการที่ 7
รายการที่ 8
รายการที่ 9
รายการที่ 10
รายการที่ 11
รายการที่ 12

ชนิดการชำระเงิน
ชนิดชำระเงิน
ธนาคาร
สาขา
เลขที่
Updateชนิดการชำระเงิน

## นำเข้าข้อมูลอัตราค่าธรรมเนียม

นำเข้าข้อมูลอัตราค่าธรรมเนียม
ปีการศึกษา
ภาคเรียนที่
ประเภทค่าธรรมเนียม
ประเภทค่าธรรมเนียมย่อย
นิสิตที่รับเข้าตั้งแต่ปี
ถึงปี
จำนวนเงิน
เพิ่มรายการ
ลบรายการ
Updateข้อมูล
ยกเลิกข้อมูล
ระบบชำระค่าธรรมเนียม
รายการหลัก

## นำเข้าข้อมูลการยกเว้นค่าธรรมเนียม

นำเข้าข้อมูลการยกเว้นค่าธรรมเนียม
ปีการศึกษา
ภาคเรียน
ประเภทนักศึกษาที่ยกเว้น
รายการค่าธรรมเนียมที่ยกเว้น
Updateข้อมูลยกเว้นค่าธรรมเนียม
ยกเลิกข้อมูล
ระบบชำระค่าธรรมเนียม
รายการหลัก

รายการค่าธรรมเนียมที่ยกเว้น
รายการที่ 1
รายการที่ 2
:
:
รายการที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตรวจสอบการชำระค่าธรรมเนียม

ตรวจสอบการชำระค่าธรรมเนียม
ปีการศึกษา
ภาคเรียน
ชนิดการเรียงข้อมูล
ตรวจสอบ
ยกเลิกการตรวจสอบ
แสดงผลการตรวจสอบ
ระบบชำระค่าธรรมเนียม
รายการหลัก

ผลการตรวจสอบ
ปีการศึกษา
ภาคเรียน
คณะ
ภาควิชา
รหัสนักศึกษา
ชื่อ-นามสกุล
พิมพ์ออกเครื่องพิมพ์
ระบบชำระค่าธรรมเนียม
รายการหลัก

## รายงานการเงิน

ประเภทรายงาน
ประเภทรายการ
รายงานตามประเภท
ระบบชำระค่าธรรมเนียม
รายการหลัก

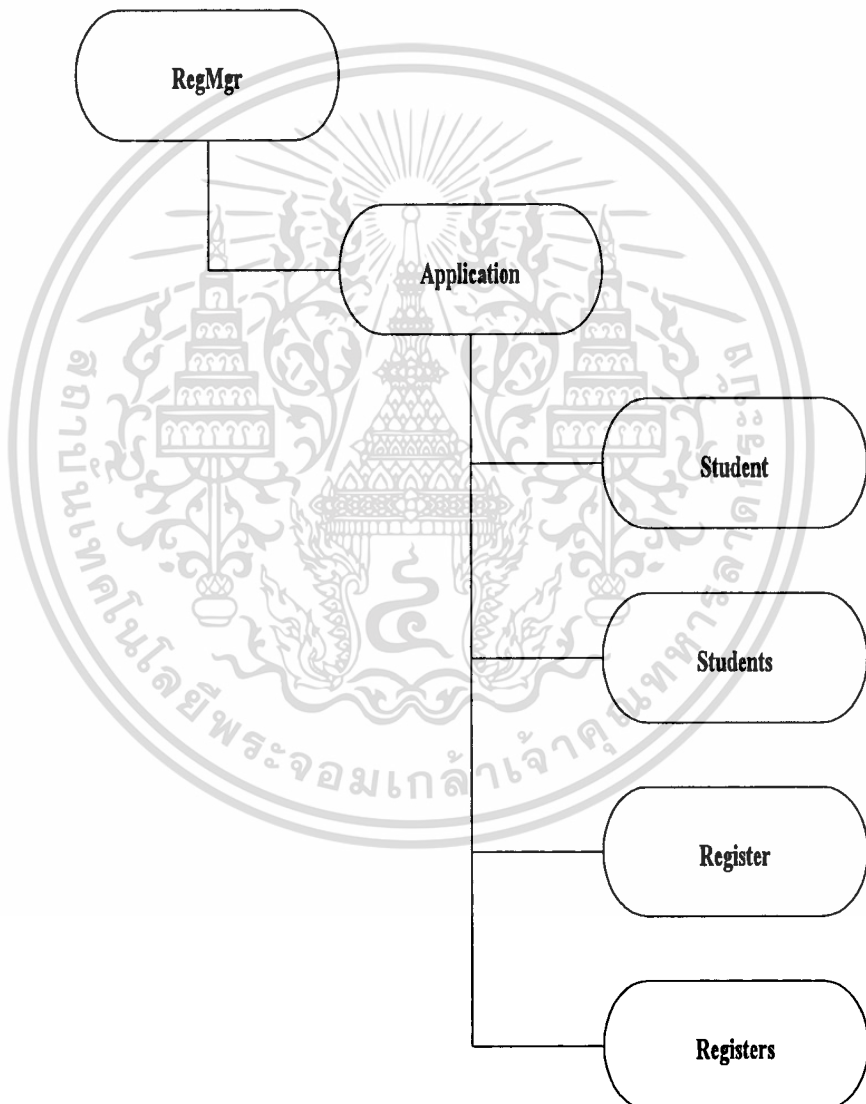
รายงานการเงิน
ปีการศึกษา
คณะ
จำนวนเงิน
พิมพ์ออกเครื่องพิมพ์
ระบบการลงทะเบียน
รายการหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ARCHITECTING TIER2 : BUSINESS SERVER

### ระบบการลงทะเบียน

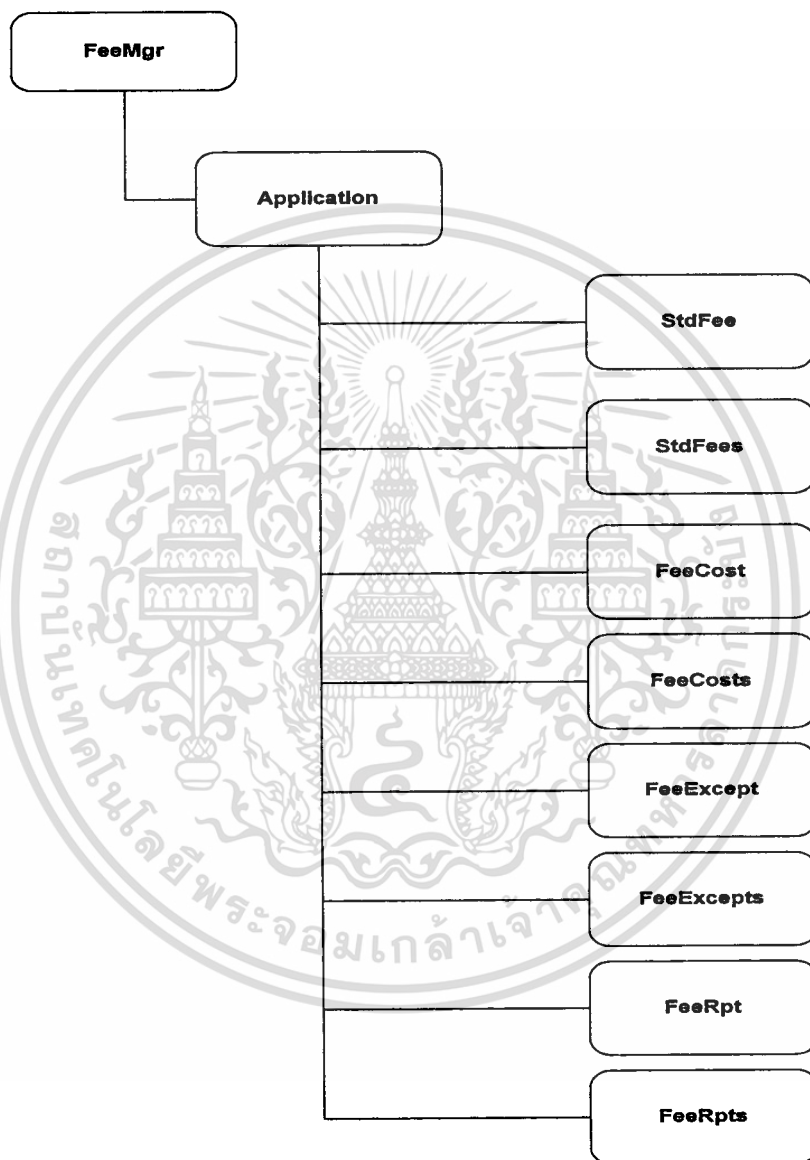
#### Object Hirarchie โดยรวมของระบบการลงทะเบียน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบชำระค่าธรรมเนียม

### Object Hirarchie โดยรวมของระบบชำระค่าธรรมเนียม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ARCHITECTING TIER 3 : DATA SERVERS

ระบบการลงทะเบียน ตารางข้อมูลหลักที่เกี่ยวข้อง

Register (ตารางวิชาที่ลงทะเบียน)

Column Name	Datatype	Size	Key
Regis_No	int	4	pk
Regis_Subj_Id	char	6	pk
Regis_Lab_Group_No char	char	2	
Regis_Lec_Group_No char	char	2	
Subj_Regis_Type	char	1	
Regis_Status	char	1	
Regis_subj_Grade	char	2	
Regis_Flag	char	1	

RegisterCard (ตารางใบลงทะเบียน)

Column Name	Datatype	Size	Key
Regis_No	int	4	pk
Regis_Std_Id	char	8	
Regis_Year	char	4	
Regis_Term	char	1	
Regis_Type	char	1	
Regis_Date	char	10	
Regis_Pay_Type	char	50	
Std_Pay_Type	char	50	
Bill_Book_No	char	4	
Bill_Order_No	char	4	
Bill_Date	char	20	
Fee_Type1	int	4	
::	::	:	
Fee_Type12	int	4	
Paid_Bank_Name	char	20	
Paid_Bank_Branch	char	30	
Paid_Bank_Check_No	int	4	

## ระบบชำระค่าธรรมเนียม ตารางข้อมูลหลักที่เกี่ยวข้อง

### Fee(การเก็บค่าธรรมเนียม)

Column Name	Datatype	Size	Key
Subj_Id	char	6	pk
Fee_Year_Begin	char	4	pk
Fee_Term_Begin	char	1	pk
Fee_Lecture	int	4	
Fee_Lab	int	4	
Atstd_Begin	char	4	
Atstd_End	char	4	

### FeeCost(ค่าธรรมเนียม)

Column Name	Datatype	Size	Key
Fee_Type_No	int	4	pk
Fee_Year	char	4	pk
Fee_Term	char	4	pk
Fee_Sub_Type_No	int	4	pk
Std_Type_No	char	2	pk
Std_Faculty_Id	char	2	pk
Std_Year_Enter1	char	4	pk
Std_Year_Enter2	char	4	
Fee_Cost	int	4	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**FeeExcept(รายการยกเว้นค่าธรรมเนียม)**

Column Name	Datatype	Size	Key
Std_Pay_Type_No	char	4	pk
Term	char	1	pk
Year	char	4	pk
Fee_Type_No1	int	4	
::	::	:	
Fee_Type_No12	int		4

**FeeExceptName(ชื่อรายการยกเว้นค่าธรรมเนียม)**

Column Name	Datatype	Size	Key
Std_Except_Type_No	char	4	pk
Except_Type_Name	char	50	

**FeeSubType(รายการค่าธรรมเนียมย่อย)**

Column Name	Datatype	Size	Key
Fee_Type_No	int	4	pk
Fee_Sub_Type_No	int	4	pk
Fee_Sub_Type_Name	char	50	
For_Whom	char	50	

**FeeType(ชนิดค่าธรรมเนียม)**

Column Name	Datatype	Size	Key
Fee_Type_No	int	4	pk
Fee_Type_Name	char	50	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Tier2 และ Tier 3 RegMgr Object Interface ของระบบการลงทะเบียน

	Application	Students	Student	Registers	Register
Collection	Students Registers				
Member Object		Item		Item	
Properties	Id UserId	Count	StdId StdName StdSname Faculty Depart Major Teacher Campus	Count	SubjId SubjName Subjtype LecGroupNo LabGroupNo Grade Year Term RemoveFlag
Methods:	Initialize	Getstudent GetFaculty GetDepart Add Move CollectionGet	PropertyGet	LoadReg RptReg ChkReg UpdReg Add Remove CollectionGet CollectionSet	PropertyGet PropertySet
Data Tier or Tier 3		sp_Get_Student sp_Get_Faculty sp_Get_Depart		sp_Load_Reg sp_Rpt_Reg sp_Chk_Reg sp_Add_Reg sp_Del_Reg sp_Upd_Reg	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Tier2 และ Tier 3 RegMgr Object Interface ของระบบชำระค่าธรรมเนียม

	Application	StdFees	StdFee	FeeCosts	FeeCost
Collection	StdFees FeeCosts FeeExcepts FeeRpts				
MemberObj		Item		Item	
Properties	Id UserId Password	Count	StdId StdName StdSname Major Depart Faculty Campus FeeType 1 : : : FeeType 12 StdType PayType BankName BankBranch BankChkNo	Count	FeeTypeNo FeeTypeName FeeSubTypeNo FeeSubTypeNam e YearBegin YearEnd Cost
Methods:	Initialize	GetStdFee GetFeeType UpdateCost ChkFee Add Remove CollectionGet CollectionSet	PropertyGet PropertySet	GetFeeTypeName GetFeeSubTypeName UpdateFeeSub Add Remove CollectionGet CollectionSet	PropertyGet PropertySet
Data Tier : Tier 3		SP_Get_Std SP_Get_Fee SP_Update_Fee SP_Chk_Fee		SP_Get_FeeName SP_Get_FeeSubTypName SP_Update_FeeType	

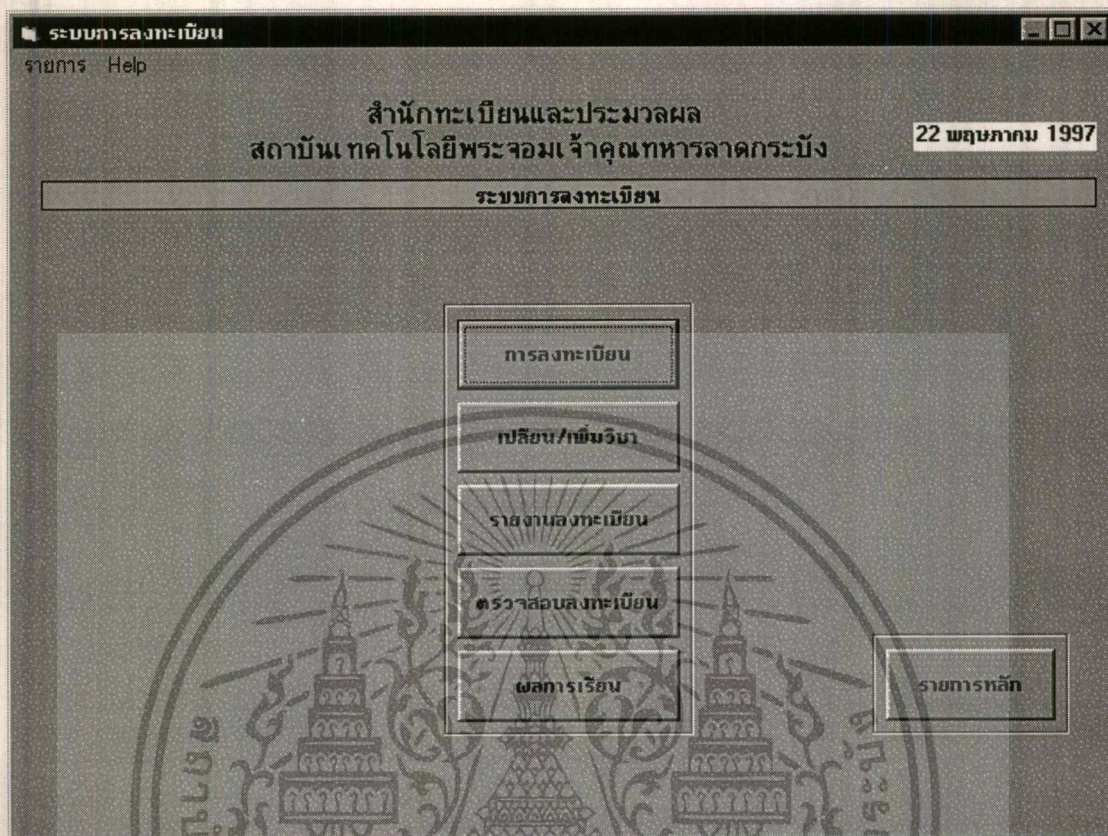
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Tier2 และ Tier 3 RegMgr Object Interface ของระบบชำระค่าธรรมเนียม(ต่อ)**

	<b>FeeExcept</b>	<b>FeeExcept</b>	<b>FeeRpts</b>	<b>FeeRpt</b>
Collections				
MemberObj	Item		Item	
Properties	Count	ExceptTypeName FeeExceptType 1 : FeeExceptType 12	Count	Faculty TypeRpt Year Cost
Methods :	FeeExceptUpdate GetStdType  CollectionGet CollectionSet	PropertyGet PropertySet	FeeRpt  CollectionGet	PropertyGet
Data Tier : Tier 3	SP_Update_FeeExcept SP_Get_StdType		SP_FeeRpt	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รูปแสดงหน้าจอการทำงานของโปรแกรมส่วนระบบการลงทะเบียน



### รูปแสดงหน้าจอเข้าสู่ระบบการลงทะเบียน

เมื่อเรียก โปรแกรมขึ้นมาจะเข้าสู่หน้าจอนี้ หน้าทีขอแต่ละปุ่มมีดังนี้

- การลงทะเบียน เมื่อคลิกปุ่มนี้จะเข้าสู่การลงทะเบียนเรียน
- เปลี่ยน/เพิ่มวิชา เมื่อคลิกปุ่มนี้จะเข้าสู่หน้าจอเปลี่ยนเพิ่มวิชาที่ลงทะเบียน
- รายงานการลงทะเบียน เมื่อคลิกปุ่มนี้จะเข้าสู่หน้าจอรายงานการลงทะเบียนที่ลง
- ตรวจสอบการลงทะเบียน เมื่อคลิกปุ่มนี้จะเข้าสู่หน้าจอตรวจสอบการลงทะเบียนว่ามี  
นักศึกษาคนใดที่ยังไม่ได้ลงทะเบียน
- ผลการเรียน เมื่อคลิกปุ่มนี้จะเข้าสู่หน้าจอผลการเรียน นำเข้าเกรดและรายงาน
- รายการหลัก เมื่อคลิกปุ่มนี้จะออกจาก โปรแกรม

การลงทะเบียน

สำนักงานทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
การลงทะเบียน

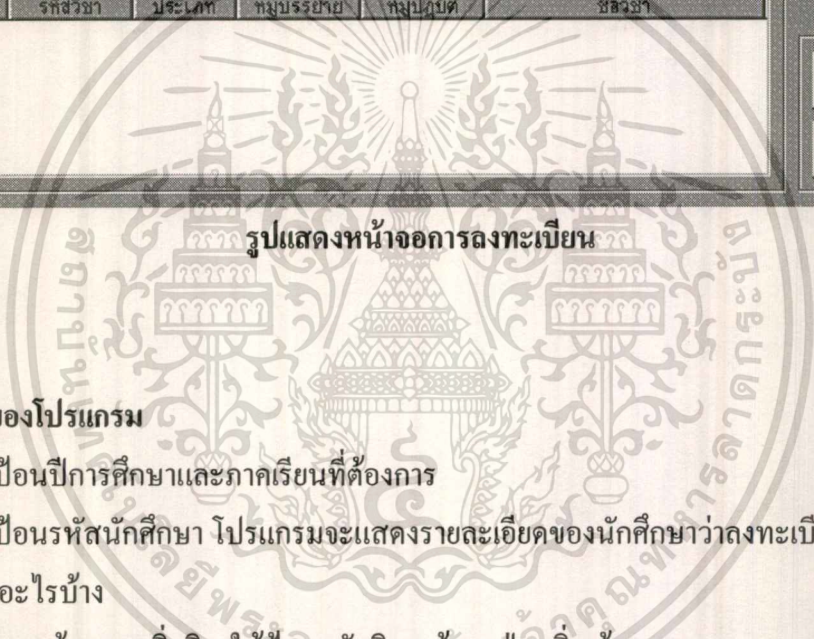
22 พฤษภาคม 1997

ปีการศึกษา :  ภาคเรียนที่ :

ข้อมูลนิสิต  
รหัสนักศึกษา :  ชื่อ :   
คณะ :  สาขาวิชาเอก :   
ภาควิชา :  อาจารย์ที่ปรึกษา :   
วิทยาเขต :

รายละเอียดวิชาที่ทำการลงทะเบียน  
รหัสวิชา :  ชื่อวิชา :   
หมู่บรรยาย :  ประเภทการลงทะเบียน  
หมู่ปฏิบัติ :   กิตถหน่วยกิต  ไม่กิตถหน่วยกิต

สรุปรายวิชาที่เปลี่ยนแปลง

ลำดับที่	รหัสวิชา	ประเภท	หมู่บรรยาย	หมู่ปฏิบัติ	ชื่อวิชา
					

เพิ่มเข้า  
ลบออก  
ตกลง  
ยกเลิก  
ออก  
รายการ

### รูปแสดงหน้าจอการลงทะเบียน

#### การทำงานของโปรแกรม

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- ป้อนรหัสนักศึกษา โปรแกรมจะแสดงรายละเอียดของนักศึกษาว่าลงทะเบียน วิชาอะไรบ้าง
- หากต้องการเพิ่มวิชาให้ป้อนรหัสวิชาแล้วกดปุ่ม เพิ่มเข้า
- หากต้องการลบวิชาออกให้คลิกที่รายละเอียดวิชาที่ต้องการลบแล้วกดปุ่มลบออก
- กดปุ่มต้องลงจะเข้าสู่ ขั้นตอนการลงทะเบียน หากไม่ตกลงกดปุ่มยกเลิก
- กดปุ่มออก ถ้าต้องการกลับสู่ระบบการลงทะเบียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยื่นยื่นการลงทะเบียน

สำนักงานทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

ยื่นยื่นการลงทะเบียน

เลขประจำตัวนิสิต :  นีอนิสิต :

คณะ :  สาขาวิชาเอก :

ภาควิชา :  อาจารย์ที่ปรึกษา :

ปีการศึกษา :  ภาคเรียนที่ :

รายละเอียดการลงทะเบียน

ลำดับที่	รหัสวิชา	ประเภท	หมู่บรรยาย	หมู่ปฏิบัติ	ชื่อวิชา

จำนวนหน่วยกิตรวม

หน่วยกิตรวม  หน่วย

### รูปแสดงหน้าจอยื่นยื่นการลงทะเบียน

หน้าจอนี้จะแสดงรายละเอียดของวิชาที่ลงทะเบียน ให้ทำการตรวจสอบที่ลงทะเบียนว่าถูกต้องหรือไม่ หากไม่ถูกต้องให้กดปุ่ม ยกเลิก จะกลับสู่การลงทะเบียน ให้ทำการแก้ไขให้ถูกต้อง ถ้าถูกต้องแล้ว กดปุ่มตกลง โปรแกรมจะทำการ Update ข้อมูลลงฐานข้อมูล

เปลี่ยน/เพิ่มวิชาเรียน

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เปลี่ยน/เพิ่มวิชาเรียน

22 พฤษภาคม 1997

ปีการศึกษา :  ภาคเรียนที่ :

ข้อมูลนิสิต  
รหัสนักศึกษา :  ชื่อ :   
คณะ :  สาขาวิชาเอก :   
ภาควิชา :  อาจารย์ที่ปรึกษา :   
วิทยาเขต :

รายละเอียดวิชาที่ทำการลงทะเบียน  
รหัสวิชา :  ชื่อวิชา :   
หมู่บรรยาย :   
หมู่ปฏิบัติ :  ประเภทการลงทะเบียน  
 กิตยหน่วยกิต  ไม่กิตยหน่วยกิต

สรุปรายวิชาที่เปลี่ยนแปลง

ลำดับที่	รหัสวิชา	ประเภท	หมู่บรรยาย	หมู่ปฏิบัติ	ชื่อวิชา

เพิ่มเข้า  
ลบออก  
ตกลง  
ยกเลิก  
ออก  
รายการ

### รูปแสดงหน้าจอการเปลี่ยน/เพิ่มวิชาเรียน

#### การทำงานของโปรแกรม

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- ป้อนรหัสนักศึกษา โปรแกรมจะแสดงรายละเอียดของนักศึกษาว่าลงทะเบียน วิชาอะไรบ้าง
- หากต้องการ เพิ่มวิชาให้ป้อนรหัสวิชาแล้วกดปุ่ม เพิ่มเข้า
- หากต้องการลบวิชาออกให้คลิกที่รายละเอียดวิชาที่ต้องการลบแล้วกดปุ่มลบออก
- กดปุ่มตกลง โปรแกรมจะบันทึกหลักฐานข้อมูล หากไม่ตกลงกดปุ่มยกเลิก
- กดปุ่มออก ถ้าต้องการกลับสู่ระบบการลงทะเบียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานการลงทะเบียน

สำนักงานทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

รายงานการลงทะเบียน

ปีการศึกษา :  ภาคเรียนที่ :

ตกลง  
ยกเลิก

เรียงตามรหัสนักศึกษา  
เรียงตามรหัสวิชา  
เรียงตามสาขาวิชา  
เรียงตามภาควิชา  
เรียงตามคณะ

ระบุภาค ทั้งหมด  
คณะ :

ภาควิชา :

ออก  
รายการ

### รูปแสดงหน้าจอรายงานการลงทะเบียน

#### การทำงานของโปรแกรม

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- ถ้าต้องการให้รายงานทุกคณะและทุกภาค กดปุ่ม ทั้งหมดแล้วกดปุ่มเรียงตามที่ต้องการ
- ถ้าต้องการระบุภาคและคณะ กดปุ่ม ระบุภาค แล้วเลือกคณะและภาค
- กดปุ่มต้องลงจะแสดงผลรายงานการลงทะเบียน หากไม่ตกลงกดปุ่มยกเลิก
- กดปุ่มออก ถ้าต้องการกลับสู่ระบบการลงทะเบียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

ผลรายงานการลงทะเบียน

สำนักงานทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

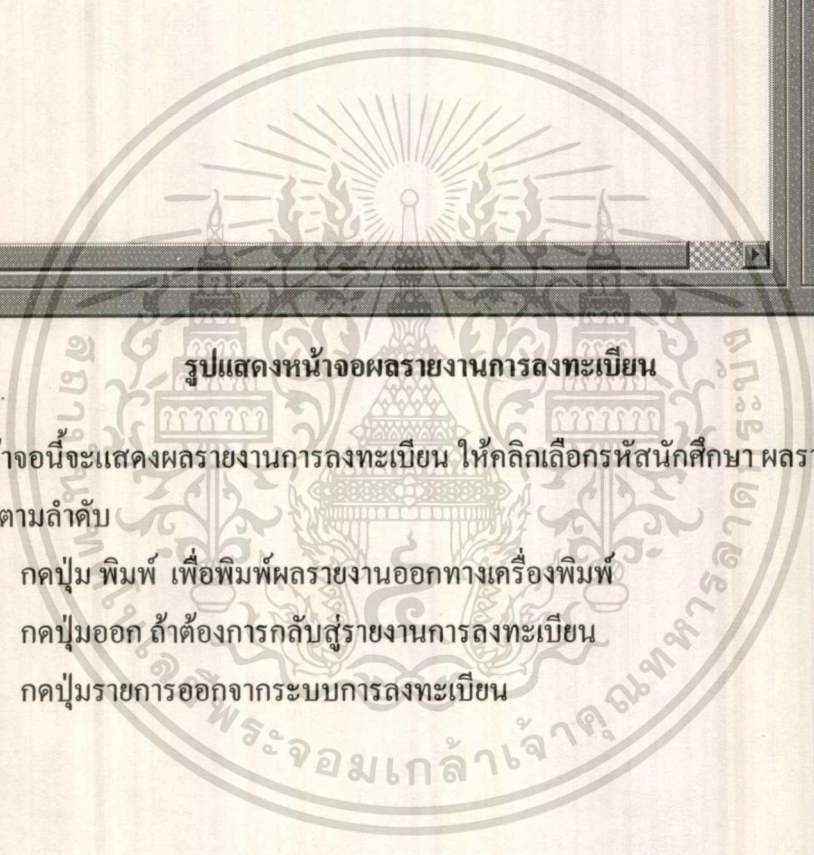
ผลรายงานการลงทะเบียน

ปีการศึกษา :  ภาคเรียนที่ :

คณะ :  ภาควิชา :

รหัสนักศึกษา :  ชื่อนักศึกษา :

เพิ่ม

ลำดับที่	รหัสวิชา	ประเภท	กลุ่มบรรยาย	กลุ่มปฏิบัติ	ชื่อวิชา
					

ออก

รายการ

### รูปแสดงหน้าจอผลรายงานการลงทะเบียน

หน้าจอนี้จะแสดงผลรายงานการลงทะเบียน ให้คลิกเลือกรหัสนักศึกษา ผลรายงานจะแสดงที่ list ตามลำดับ

- กดปุ่ม พิมพ์ เพื่อพิมพ์ผลรายงานออกทางเครื่องพิมพ์
- กดปุ่มออก ถ้าต้องการกลับสู่รายงานการลงทะเบียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบการลงทะเบียน

สำนักงานทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

ตรวจสอบการลงทะเบียน

ปีการศึกษา :  ภาคเรียนที่ :

ตกลง  
ยกเลิก

ตามรหัสนักศึกษา  
ตามสาขาวิชาเอก  
ตามภาควิชา  
ตามคณะ

ระบุภาค ทั้งหมด

คณะ :

ภาควิชา :

ออก  
รายการ

### รูปแสดงหน้าจอตรวจสอบการลงทะเบียน

หน้าจอนี้เป็นการตรวจสอบการลงทะเบียนว่ามีใครที่ยังไม่ได้ลงทะเบียน

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- ถ้าต้องการให้ตรวจสอบทุกคณะและทุกภาค กดปุ่ม ทั้งหมด แล้วกดปุ่มเรียงตาม
- ถ้าต้องการระบุภาคและคณะ กดปุ่ม ระบุภาค แล้วเลือกคณะและภาค
- กดปุ่มต้องลงจะแสดงผลการตรวจสอบการลงทะเบียน หากไม่ตกลงกดปุ่มยกเลิก
- กดปุ่มออก ถ้าต้องการกลับสู่ระบบการลงทะเบียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

รายงานชื่อนักศึกษาที่ยังไม่ได้ลงทะเบียน

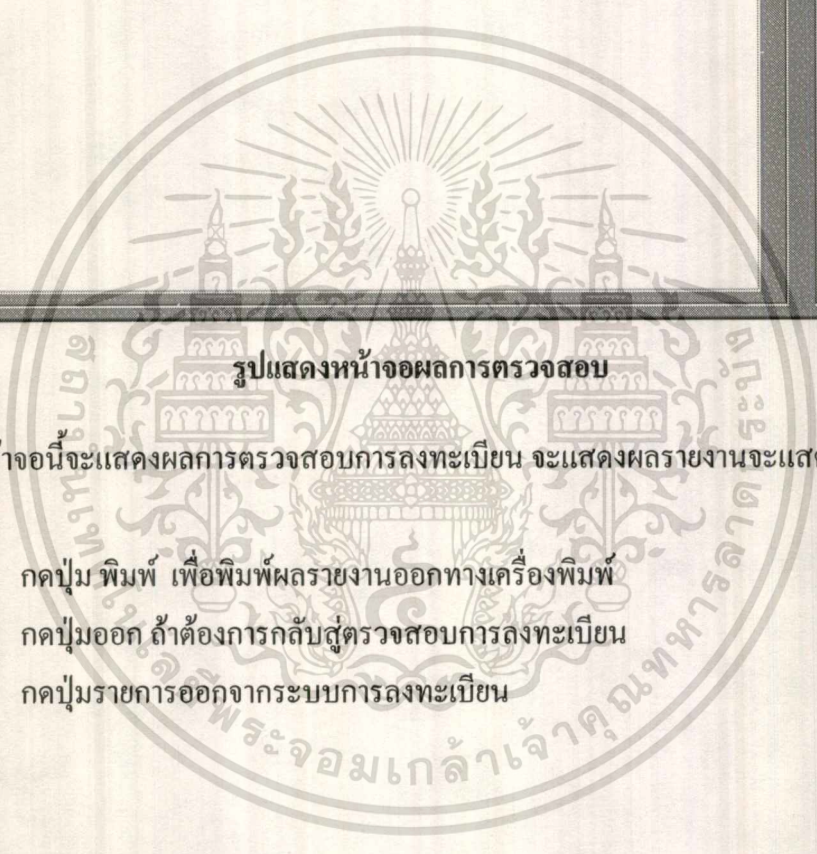
สำนักงานทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

22 พฤษภาคม 1997

รายงานชื่อนักศึกษาที่ยังไม่ได้ลงทะเบียน

ปีการศึกษา  ภาคเรียนที่

คณะ :  สาขาวิชา :

ลำดับที่	รหัสนักศึกษา	ชื่อ	นามสกุล นักศึกษา
			

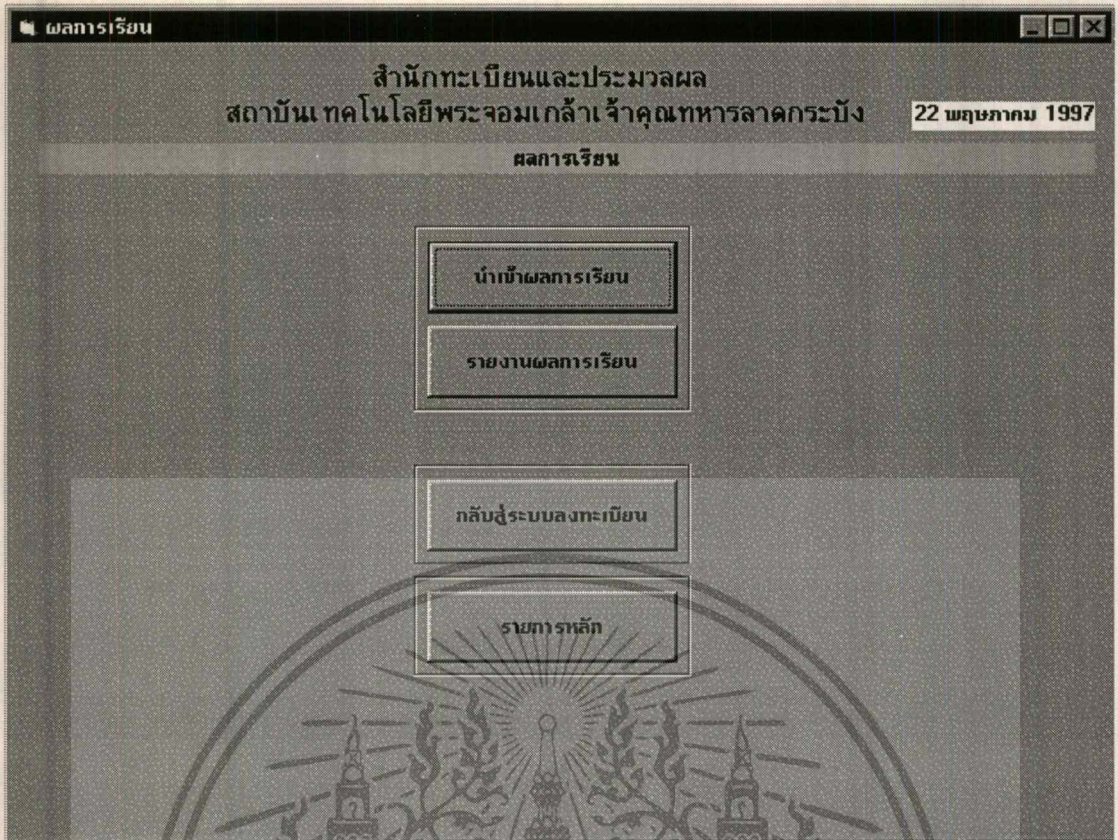
### รูปแสดงหน้าจอผลการตรวจสอบ

หน้าจอนี้จะแสดงผลการตรวจสอบการลงทะเบียน จะแสดงผลรายงานจะแสดงที่ list ตาม

ลำดับ

- กดปุ่ม พิมพ์ เพื่อพิมพ์ผลรายงานออกทางเครื่องพิมพ์
- กดปุ่มออก ถ้าต้องการกลับสู่ตรวจสอบการลงทะเบียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### รูปแสดงหน้าจอผลการเรียน

- กดปุ่ม นำเข้าผลการเรียน จะเข้าสู่หน้าจอนำเข้าผลการเรียน
- กดปุ่ม รายงานผลการเรียนจะเข้าสู่หน้าจอรายงานผลการเรียน
- กดปุ่ม กลับสู่ระบบลงทะเบียน จะกลับไปยังหน้าจอ ระบบการลงทะเบียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าเข้าสู่ผลการเรียน

สำนักงานทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

หน้าเข้าสู่ผลการเรียน

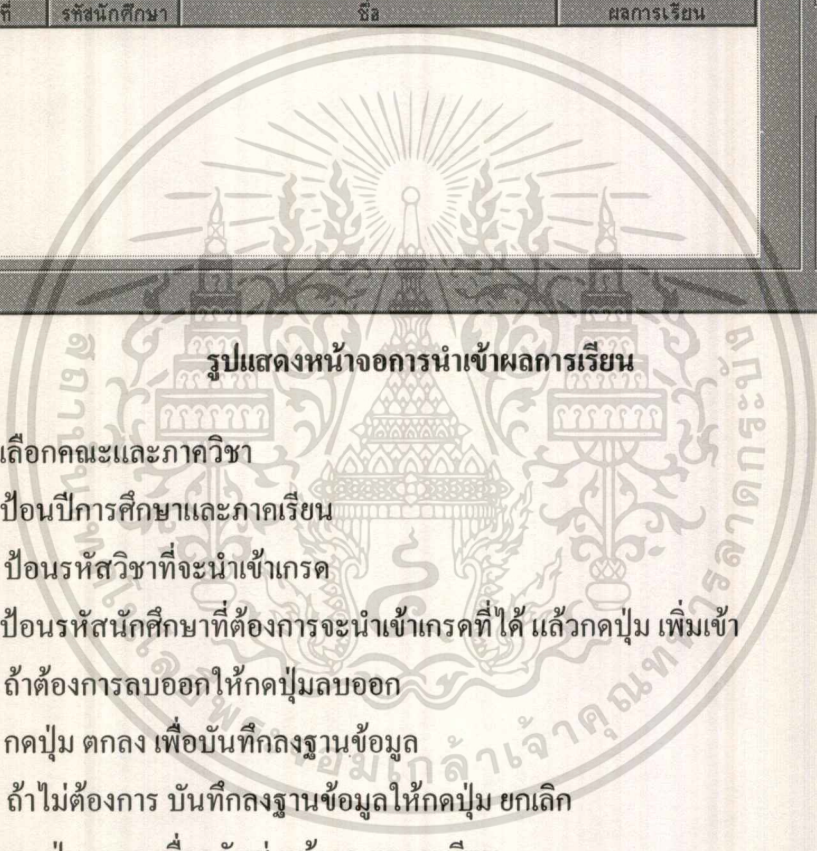
คณะ :  ภาควิชา :

ปีการศึกษา :  ภาคเรียนที่ :

รหัสวิชา :  ชื่อวิชา :

รหัสนักศึกษา :  ชื่อ :

ผลการเรียน :  หน่วยกิต :

ลำดับที่	รหัสนักศึกษา	ชื่อ	ผลการเรียน
			

ปุ่ม:

### รูปแสดงหน้าจอการนำเข้าผลการเรียน

- เลือกคณะและภาควิชา
- ป้อนปีการศึกษาและภาคเรียน
- ป้อนรหัสวิชาที่จะนำเข้าเกรด
- ป้อนรหัสนักศึกษาที่ต้องการจะนำเข้าเกรดที่ได้ แล้วกดปุ่ม เพิ่มเข้า
- ถ้าต้องการลบออกให้กดปุ่มลบออก
- กดปุ่ม ตกลง เพื่อบันทึกถาวรข้อมูล
- ถ้าไม่ต้องการ บันทึกถาวรข้อมูลให้กดปุ่ม ยกเลิก
- กดปุ่ม ออก เพื่อกลับสู่ หน้าจอ ผลการเรียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานผลการเรียน

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

รายงานผลการเรียน

คณะ :  สาขาวิชา :

ปีการศึกษา :  ภาคเรียนที่ :

รหัสนักศึกษา :  ชื่อ :

รหัสนักศึกษา      รหัสนักศึกษา ที่ต้องการ

>>  
>  
<  
<<

ตกลง

ออก

รายการ

### รูปแสดงหน้าจอรายงานผลการเรียน

- เลือกคณะและภาควิชา
- ป้อนปีการศึกษาและภาคเรียน
- ป้อนรหัสนักศึกษา ที่ต้องการ ถ้าต้องการหลายคนให้คลิกที่ รายการรหัสนักศึกษา แล้วกดปุ่ม > ถ้าต้องการหมด กด >>
- กดปุ่ม ตกลง จะเข้าสู่หน้าจอแสดงผลรายงาน
- กดปุ่ม ออก เพื่อกลับสู่ หน้าจอ ผลการเรียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผลการเรียน

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

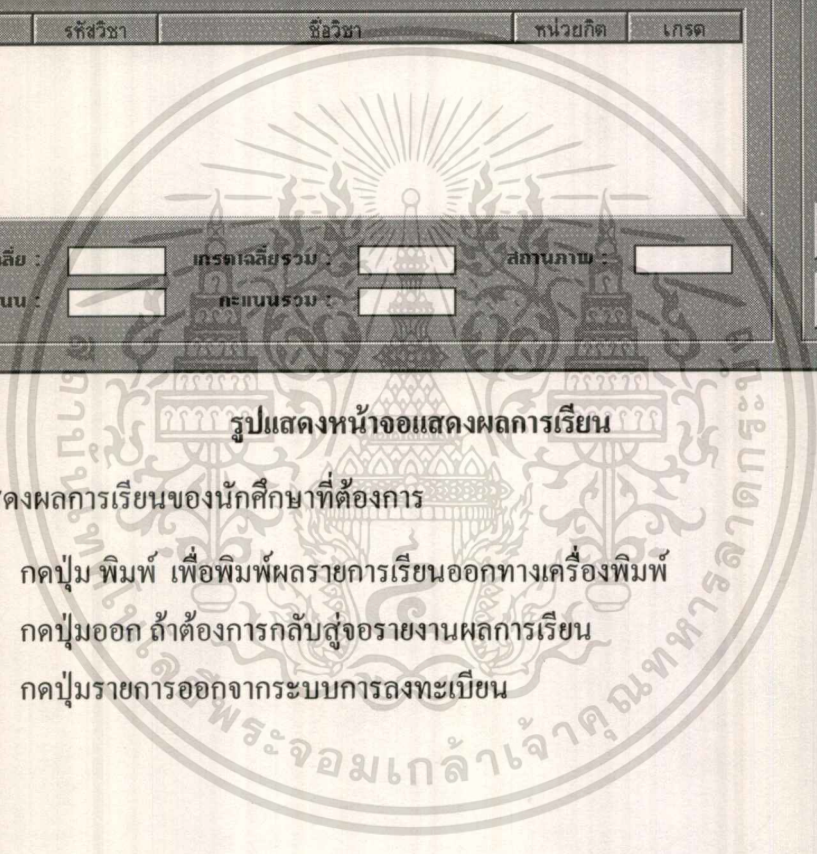
แสดงผลการเรียน

รหัสนักศึกษา :  ชื่อ :

สาขาวิชาเอก :  ภาควิชา :

คณะ :

ปีการศึกษา :  ภาคเรียนที่ :

ลำดับที่	รหัสวิชา	ชื่อวิชา	หน่วยกิต	เกรด
				

เกรดเฉลี่ย :  เกรดเฉลี่ยรวม :  สถานภาพ :

คะแนน :  คะแนนรวม :

พิมพ์

ออก

รายการ

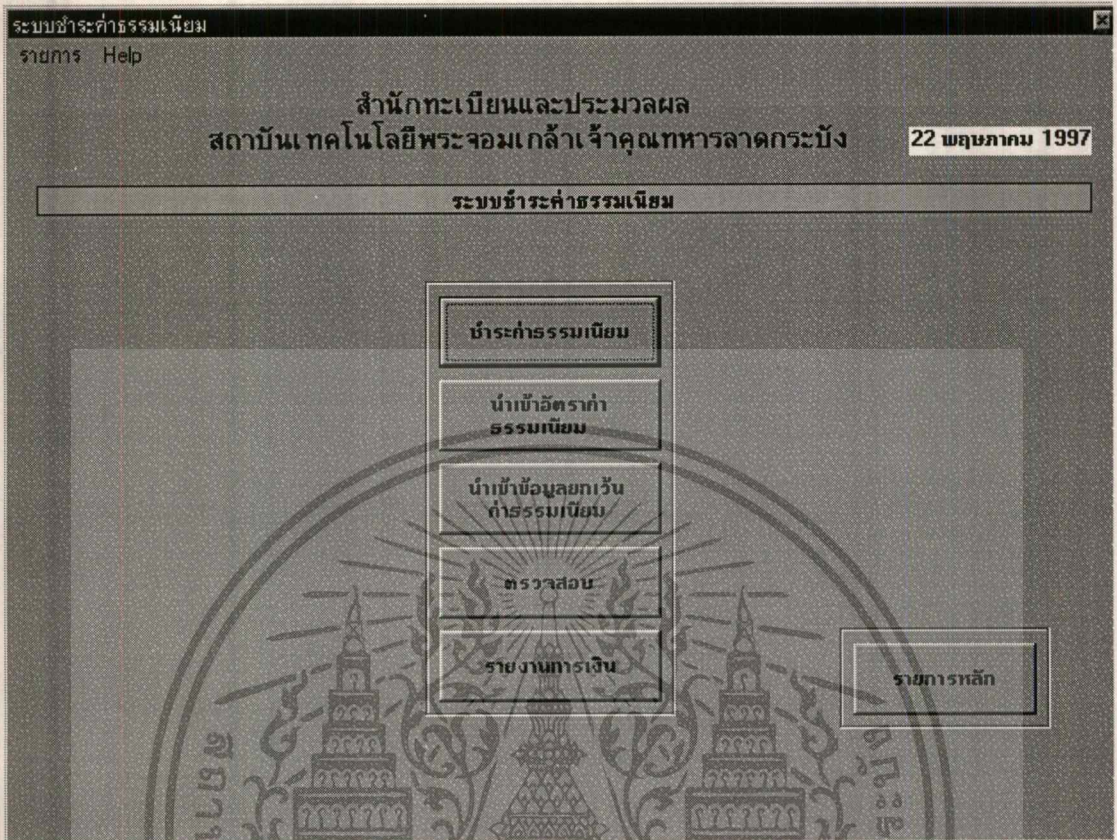
### รูปแสดงหน้าจอแสดงผลการเรียน

หน้าจอแสดงผลการเรียนของนักศึกษาที่ต้องการ

- กดปุ่ม พิมพ์ เพื่อพิมพ์ผลรายการเรียนออกทางเครื่องพิมพ์
- กดปุ่มออก ถ้าต้องการกลับสู่จอรายงานผลการเรียน
- กดปุ่มรายการออกจากระบบการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รูปแสดงหน้าการทำงานของโปรแกรมส่วนระบบการชำระค่าธรรมเนียม



### รูปแสดงหน้าจอระบบชำระค่าธรรมเนียม

เมื่อเรียกโปรแกรมขึ้นมาจะเข้าสู่หน้าจอนี้ หน้าทีขอแต่ละปุ่มมีดังนี้

- ชำระค่าธรรมเนียม เมื่อกดปุ่มนี้จะเข้าสู่หน้าจอการชำระค่าธรรมเนียม
- นำเข้าอัตราค่าธรรมเนียม เมื่อกดปุ่มนี้จะเข้าสู่หน้าจอนำเข้าอัตราค่าธรรมเนียม
- นำเข้าอัตราค่าธรรมเนียม เมื่อกดปุ่มนี้จะเข้าสู่หน้าจอนำเข้าอัตราค่าธรรมเนียม
- ตรวจสอบ เมื่อกดปุ่มนี้จะเข้าสู่หน้าจอตรวจสอบการชำระค่าธรรมเนียมว่ามี

นักศึกษาคนใดที่ยังไม่ได้ชำระค่าธรรมเนียม

- รายงานการเงิน เมื่อกดปุ่มนี้จะเข้าสู่หน้าจอรายงานการเงิน
- รายการหลัก เมื่อกดปุ่มนี้จะออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชำระค่าธรรมเนียม

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

ชำระค่าธรรมเนียม

ปีการศึกษา :  เทอม :   
 รหัสนักศึกษา :  ชื่อ :   
 สาขาวิชาเอก :  ภาควิชา :   
 คณะ :  วิทยาเขต :

รายละเอียดค่าธรรมเนียม

รายการค่าธรรมเนียม	จำนวนเงิน
ค่าธรรมเนียมแรกเข้า	
ค่าน่วยกิต	
ค่าบำรุงห้องสมุด	
ค่าบำรุงมหาวิทยาลัย	
ค่าบำรุงพิเศษสำหรับนักศึกษาต่างประเทศ	
ค่าบำรุงกิจการรมนิสิต	
ค่าประกันของเสียหาย	
ค่าบัตรประจำตัวนิสิต	
ค่าธรรมเนียมพิเศษคณะ	
ค่าธรรมเนียมพิเศษรายวิชา	
ค่าปรับ	
ค่าอื่น ๆ	
รวม	

การคิดค่าธรรมเนียม

ปกติ  
 กรณีพิเศษ

ชำระเงินโดย

เงินสด  ชำระที่ธนาคาร  
 ตัวแลกเงินธนาคาร  ของค้างชำระ

### รูปแสดงหน้าชำระค่าธรรมเนียม

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- ป้อนรหัสนักศึกษา จะแสดงรายละเอียดค่าธรรมเนียมที่ต้องชำระ
- เลือกชำระแบบใด
- กดปุ่ม ตกลง เพื่อบันทึกข้อมูล
- ถ้าไม่ต้องการบันทึกกดปุ่ม ยกเลิก
- กดปุ่ม ออก เพื่อกลับสู่ระบบชำระค่าธรรมเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำเข้าข้อมูลอัตราค่าธรรมเนียม

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

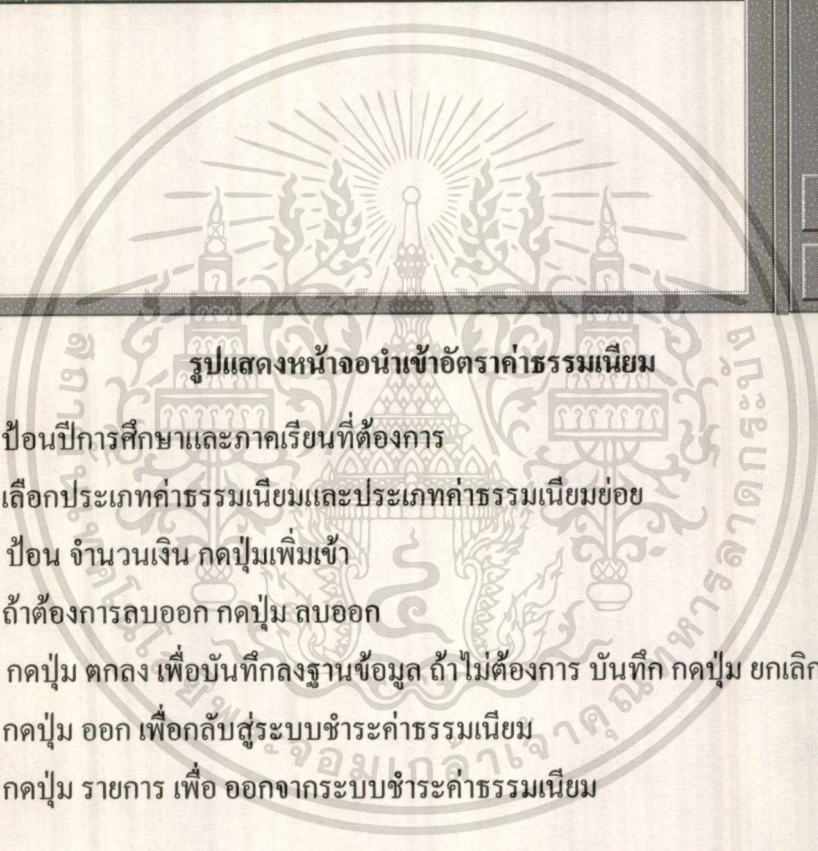
นำเข้าข้อมูลอัตราค่าธรรมเนียม

ปีการศึกษา :  ภาคเรียนที่ :

รายละเอียดอัตราค่าธรรมเนียม

ประเภทค่าธรรมเนียม :  ประเภทค่าธรรมเนียมย่อย :  จำนวนเงิน :

แสดงรายละเอียดชำระค่าธรรมเนียม

ลำดับที่	นิสิตที่รับเข้าตั้งแต่ปี	ถึงปี	จำนวนเงิน
			

เพิ่มเข้า  
ลบออก  
ตกลง  
ยกเลิก  
ออก  
รายการ

รูปแสดงหน้าจอนำเข้าอัตราค่าธรรมเนียม

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- เลือกประเภทค่าธรรมเนียมและประเภทค่าธรรมเนียมย่อย
- ป้อน จำนวนเงิน กดปุ่มเพิ่มเข้า
- ถ้าต้องการลบออก กดปุ่ม ลบออก
- กดปุ่ม ตกลง เพื่อบันทึกหลักฐานข้อมูล ถ้าไม่ต้องการ บันทึก กดปุ่ม ยกเลิก
- กดปุ่ม ออก เพื่อกลับสู่ระบบชำระค่าธรรมเนียม
- กดปุ่ม รายการ เพื่อ ออกจากระบบชำระค่าธรรมเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำเข้าข้อมูลการยกเว้นค่าธรรมเนียม

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

นำเข้าข้อมูลการยกเว้นค่าธรรมเนียม

ปีการศึกษา :  ภาคเรียนที่ :

ประเภทนักศึกษาที่ยกเว้นค่าธรรมเนียม :

ค่าธรรมเนียมที่ยกเว้น

ค่าธรรมเนียมแรกเข้า  ค่าประกันของเสียหาย

ค่าหน่วยกิต  ค่าบัตรประจำตัวนิสิต

ค่าบำรุงห้องสมุด  ค่าธรรมเนียมพิเศษคณะ

ค่าบำรุงมหาวิทยาลัย  ค่าธรรมเนียมพิเศษรายวิชา

ค่าบำรุงพิเศษสำหรับนักศึกษาต่างประเทศ  ค่าปรับ

ค่าบำรุงกิจกรรมนิสิต  ค่าอื่น ๆ

ตกลง

ยกเลิก

ออก

รายการ

### รูปแสดงหน้าจอนำเข้าข้อมูลการยกเว้นค่าธรรมเนียม

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- เลือกประเภทนักศึกษาที่ได้รับการยกเว้นค่าธรรมเนียม
- เลือกค่าธรรมเนียมที่ยกเว้น
- กดปุ่ม ตกลง เพื่อบันทึกลงฐานข้อมูล ถ้าไม่ต้องการ บันทึก กดปุ่ม ยกเลิก
- กดปุ่ม ออก เพื่อกลับสู่ระบบชำระค่าธรรมเนียม
- กดปุ่ม รายการ เพื่อ ออกจากระบบชำระค่าธรรมเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายชื่อนักศึกษาที่ยังไม่ได้ชำระค่าธรรมเนียม

สำนักทะเบียนและประมวลผล  
สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

ตรวจสอบการชำระค่าธรรมเนียม

ปีการศึกษา :  ภาคเรียนที่ :

ตกลง  
ยกเลิก

ตามรหัสนักศึกษา  
ตามสาขาวิชา  
ตามภาควิชา  
ตามคณะ

ทั้งหมด  
ระบุภาควิชา

คณะ :

ภาควิชา :

ออก  
รายการ

### รูปแสดงหน้าจอตรวจสอบการชำระค่าธรรมเนียม

หน้าจอนี้เป็นการตรวจสอบการชำระค่าธรรมเนียมว่ามีใครที่ยังไม่ได้ชำระค่าธรรมเนียม

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- ถ้าต้องการ ให้ตรวจสอบทุกคณะและทุกภาค กดปุ่ม ทั้งหมด แล้วกดปุ่มเรียงตาม
- ถ้าต้องการระบุภาคและคณะ กดปุ่ม ระบุภาค แล้วเลือกคณะและภาค
- กดปุ่มต้องลงจะแสดงผลการตรวจสอบหากไม่ตกลงกดปุ่มยกเลิก
- กดปุ่ม ออก เพื่อกลับสู่ระบบชำระค่าธรรมเนียม
- กดปุ่ม รายการ เพื่อ ออกจากระบบชำระค่าธรรมเนียม

รายชื่อนักศึกษาที่ยังไม่ได้ชำระค่าธรรมเนียม

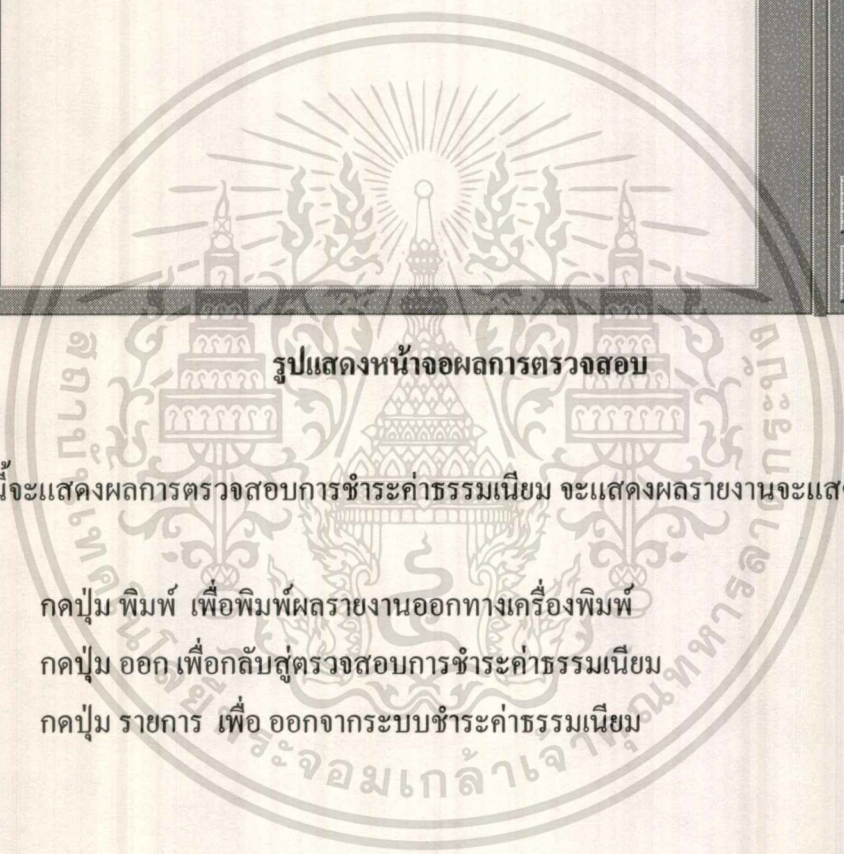
สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

ผลการตรวจสอบ

ปีการศึกษา :  ภาคเรียนที่ :

คณะ :  ภาควิชา :

รายชื่อนักศึกษาที่ยังไม่ได้ชำระค่าธรรมเนียม

ลำดับที่	รหัสนักศึกษา	ชื่อ-นามสกุล นักศึกษา
		

### รูปแสดงหน้าจอผลการตรวจสอบ

หน้าจอนี้จะแสดงผลการตรวจสอบการชำระค่าธรรมเนียม จะแสดงผลรายงานจะแสดงที่ list ตามลำดับ

- กดปุ่ม พิมพ์ เพื่อพิมพ์ผลรายงานออกทางเครื่องพิมพ์
- กดปุ่ม ออก เพื่อกลับสู่ตรวจสอบการชำระค่าธรรมเนียม
- กดปุ่ม รายการ เพื่อ ออกจากระบบชำระค่าธรรมเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทรายงาน

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 22 พฤษภาคม 1997

ประเภทรายงาน

ปีการศึกษา :  ภาคเรียนที่ :

ค่าธรรมเนียมการลงทะเบียนล่าช้า

ค่าธรรมเนียมพิเศษคณะ

การจัดสรรค่าธรรมเนียมแรกเข้า

การจัดสรรค่าบำรุงมหาวิทยาลัย

การจัดสรรค่านายกิต

การชำระเงินการลงทะเบียน KU2

ออก

รายการหลัก

### รูปแสดงหน้าจอรายงานการเงิน

- ป้อนปีการศึกษาและภาคเรียนที่ต้องการ
- กดปุ่มรายการต่างๆ ที่ต้องการ
- กดปุ่ม ออก เพื่อกลับสู่ระบบชำระค่าธรรมเนียม
- กดปุ่ม รายการ เพื่อ ออกจากระบบชำระค่าธรรมเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

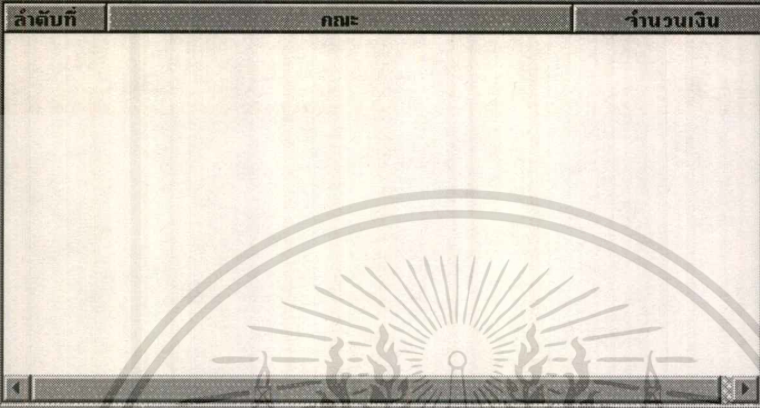
รายงานการเงิน

สำนักทะเบียนและประมวลผล  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานการเงิน

ปีการศึกษา :  ภาคเรียนที่ :  คณะ :

ปีการศึกษา

ลำดับที่	คณะ	จำนวนเงิน
		

พิมพ์

ออก

รายการ

รูปแสดงหน้าจอผลรายงานการเงิน

- เลือกคณะที่ต้องการ
- กดปุ่ม พิมพ์ เพื่อพิมพ์ผลรายงานออกทางเครื่องพิมพ์
- กดปุ่ม ออก เพื่อกลับสู่รายงานการเงิน
- กดปุ่ม รายการ เพื่อ ออกจากระบบชำระค่าธรรมเนียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลองแนวทางพัฒนาต่อและสรุป

#### ผลการทดลอง

จากการทดลองป้อนข้อมูลให้แก่โปรแกรมจะด้ผลว่าข้อมูลที่ทำการเรียกใช้จากฐานข้อมูลสามารถเรียกดูได้ดี แต่ความเร็วอาจช้าไปบ้างเนื่องจาก ใน Tier 1 และ Tier ที่ 2 นั้น โปรแกรมจะ Run อยู่ที่เดียวกันและใช้ Windows 95 เป็น Operating System และเครื่องที่ใช้มีขีดความสามารถไม่เพียงพอทำให้ระบบช้าลง

และจากการทดลองการ Insert Delete และ Update ยังใช้งานไม่ค่อยดีบ้างครั้งทำได้ บ้างครั้งทำไม่ได้เนื่องจากตัว โปรแกรมยังมีโครงสร้างการทำงานยังไม่ดีนัก และลจิกทาง SQL ยังไม่ค่อยถูกต้องนักจึงทำให้มีอาการผิดพลาด

#### แนวทางพัฒนาต่อ

โปรแกรมระบบสารสนเทศสำหรับสำนักทะเบียนในส่วนการลงทะเบียนและส่วนระบบชำระค่าธรรมเนียมนี้สิ่งที่จะต้องปรับปรุงคือระบบยังไม่ค่อยถูกต้องนัก ต้องทำการศึกษาระบบให้ละเอียดมากกว่านี้

ในส่วนการลงทะเบียนนั้นยังไม่มีการตรวจสอบวิชาบังคับผ่าน ไม่มีการตรวจสอบว่าจบการศึกษาแล้ว การนำเข้าผลการเรียนยังไม่ค่อยดีนัก และการตรวจสอบยังไม่ค่อยได้ผลที่น่าพอใจ

#### สรุป

โครงการระบบสารสนเทศสำหรับสำนักทะเบียนนี้ได้แบ่งออกเป็นส่วนย่อย ๆ หลายส่วนเพื่อให้ง่ายต่อการศึกษา โครงการนี้ไม่ได้ทำเพื่อนำไปใช้งานจริง ๆ แต่เป็นการนำความรู้ที่ได้เรียนมาในวิชาต่าง ๆ ในสายวิศวกรรมศาสตร์คอมพิวเตอร์มาประยุกต์ใช้ และเป็นประสบการณ์ในการพัฒนาระบบฐานข้อมูลต่อไปในอนาคตให้ดียิ่งขึ้น

## กิติกรรมประกาศ

โครงการระบบสารสนเทศของสำนักทะเบียนและประมวลผลส่วนงานลงทะเบียนและค่าธรรมเนียมนี้จะไม่ประสบความสำเร็จลงได้หากไม่มีบุคคลและองค์กรเหล่านี้

### ขอขอบคุณ

1. อาจารย์ วิบูลย์ พร้อมพานิชย์ ที่ให้คำปรึกษาและคอยสอน
2. คุณชนพัตร ช่วยสุข ที่ให้ยืมอุปกรณ์ในการทำงาน
3. บริษัท ไอโซเน็ต จำกัด ที่ได้ให้ซอฟต์แวร์และฮาร์ดแวร์ในการศึกษา
4. ฝ่ายสารสนเทศ โนโลยีสารสนเทศ ธนาคาร สหธนาคาร จำกัด(มหาชน) ที่ให้ใช้อุปกรณ์ในการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หนังสืออ้างอิง

1. Alope Nath, "The Guide to SQL Server Second Edition", Addison-Craig Wesley Pubshing Company, 1995
2. Goren, "Visual Basic 4 Enterprise", Que Corporation, 1996
3. Ian Sommerville, "Software Engineering Fifth Edition", Addison-Craig Wesley Pubshing Company, 1995
4. James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen, "Object Oriented Modeling and Design", Prentice Hall, 1991

