

ระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย

A Document Database System for English Learning and English-Thai

Translation

นายชนากร	อุปละ	รหัส 46380022
นายอธิปัติ	สุขสันต์	รหัส 46380045
นายเอกชัย	ทาชมภู	รหัส 46380056

ห้องสมุดคณะวิศวกรรมศาสตร์  
วันที่รับ...../7 เม.ย. 2553  
เลขทะเบียน.....14941688  
เลขเรียกหนังสือ.....ฟ/ร.  
มหาวิทยาลัยนเรศวร ๖๒318

2550

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550



## ใบรับรองโครงการนิสิต

หัวข้อโครงการ	ระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย		
ผู้ดำเนินโครงการ	นายชนกร อุปละ	รหัส 46380022	
	นายอธิปัติ สุขสันต์	รหัส 46380045	
	นายเอกชัย ทาชมภู	รหัส 46380056	
อาจารย์ที่ปรึกษา	ดร. อัครพันธ์ วงศ์กั้งแห		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์  
คณะกรรมการสอบโครงการนิสิต

.....ประธานกรรมการ

(ดร.อัครพันธ์ วงศ์กั้งแห)

.....กรรมการ

(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ

(อาจารย์จิราพร พุกสุข)

หัวข้อโครงการ	ระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย		
ผู้ดำเนินโครงการ	นายธนากร	อุปละ	รหัส 46380022
	นายอธิปิติ	สุขสันต์	รหัส 46380045
	นายเอกชัย	พาชนฎ	รหัส 46380056
อาจารย์ที่ปรึกษา	ดร. อัครพันธ์ วงศ์กังแห		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

#### บทคัดย่อ

โครงการนี้เป็นการพัฒนาโปรแกรมระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย พัฒนาโปรแกรมโดยใช้ Microsoft Visual Studio 2005 และเชื่อมต่องานข้อมูลโดยใช้ Microsoft Office Access 2003 เพื่ออำนวยความสะดวกในการค้นหาคำศัพท์ภาษาอังกฤษ จึงแบ่งการทำงานออกเป็น 3 ส่วน คือ ส่วนที่หนึ่ง การค้นหาคำศัพท์ในพจนานุกรมหลัก จะแสดงการอ่านออกเสียง ความหมายจำแนกตามชนิดของคำ ตัวอย่างการใช้ คำเหมือน และคำตรงข้าม ส่วนที่สอง การค้นหาคำศัพท์ในพจนานุกรมส่วนบุคคล จะแบ่งคำศัพท์ตามปีที่ยังบันทึก พร้อมทั้งความหมายจำแนกตามชนิดของคำและบันทึกความเข้าใจของผู้ใช้ ส่วนที่สาม การค้นหาตัวอย่างประโยคของคำศัพท์ในฐานข้อมูลเอกสาร จะแสดงตัวอย่างประโยคภาษาอังกฤษพร้อมทั้งเอกสารอ้างอิง

ผลที่ได้จากการทำโครงการนี้ คือ โปรแกรมสามารถค้นหาคำศัพท์ภาษาอังกฤษพร้อมทั้งตัวอย่างประโยค ซึ่งเป็นประโยชน์ต่อการเรียนรู้คำศัพท์ภาษาอังกฤษ อีกทั้งยังสามารถปรับปรุงคำศัพท์ให้ทันสมัยและเข้าใจได้ง่ายขึ้น

**Project title** A Document Database System for English Learning and English-Thai  
Translation

<b>Name</b>	Mr. Thanakorn Upala	ID. 46380022
	Mr. Athipat Suksan	ID. 46380045
	Mr. Ekkachai Tachompoo	ID. 46380056

**Project advisor** Dr. Akaraphunt Vongkunghae

**Major** Computer Engineering.

**Department** Electrical and Computer Engineering.

**Academic year** 2007

### Abstract

This project is a development of database system program for English learning and English-Thai translation. The program is developed by using Microsoft Visual Studio 2005 and connected with a database software by Microsoft Office Access 2003.

The program is divided into three parts. First, vocabulary search module in an original dictionary showing pronunciations, meanings, examples, synonym and antonym. Second, vocabulary search module in a personal dictionary, categorized by the year of record and the additional user's notes. Last, example sentences search module in the document database, showing the example sentences in English and their reference.

The outcome of this project is a program that can search the meaning of vocabulary, the example sentences. It is very useful for English learning. Users can add and revise vocabularies to update the dictionary for his own understanding.

## กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลงได้ด้วยความอนุเคราะห์อันดียิ่งจาก-ดร.อักรพันธ์ วงศ์กัณฑ์-ที่ได้กรุณาเป็นอาจารย์ที่ปรึกษาให้แนวคิดช่วยเหลือให้ความเอาใจใส่ตลอดและตลอดเวลาอันแสนมีค่าเพื่อตรวจแก้ไขข้อบกพร่องต่างๆจนโครงการสำเร็จลุล่วงไปด้วยดี ทางผู้จัดทำจึงใคร่ขอขอบคุณอาจารย์เป็นอย่างสูงไว้ ณ ที่นี้-

ขอขอบคุณฝ่ายการเงินของคณะวิศวกรรมศาสตร์มหาวิทยาลัยนเรศวร ที่ช่วยอำนวยความสะดวกในการเบิกจ่ายงบประมาณในการดำเนินโครงการ

ขอขอบคุณทุกๆท่านที่มีได้เอ่ยนามในที่นี้ ที่มีส่วนร่วมช่วยในโครงการนี้ให้สำเร็จลุล่วงไปด้วยดี

คณะผู้จัดทำโครงการ

นายชนากร อู่ปละ

นายอธิปัตติ สุขสันต์

นายเอกชัย ทาชมภู



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	3
1.6 งบประมาณที่ใช้.....	3
บทที่ 2 หลักการและทฤษฎี	
2.1 ระบบฐานข้อมูล (Database System).....	4
2.2 วงจรชีวิตของการพัฒนาระบบฐานข้อมูล (Database Life Cycle).....	8
2.3 อี-อาร์โมเดล.....	16
2.4 การจัดการฐานข้อมูลด้วยภาษา SQL.....	21
2.5 การเขียนโปรแกรมโดยใช้ Visual Basic.NET ติดต่อฐานและจัดการฐานข้อมูล ด้วย ADO.NET.....	32
บทที่ 3 วิธีการดำเนินงาน	
3.1 การรวบรวมและวิเคราะห์ความต้องการ.....	45
3.2 แบบจำลองการทำงานของโปรแกรม.....	48
3.3 การออกแบบฐานข้อมูล.....	50
3.4 การออกแบบโปรแกรมเพื่อใช้งานร่วมกับระบบฐานข้อมูล.....	53

## สารบัญ (ต่อ)

	หน้า
3.5 แบบจำลองการออกแบบโปรแกรม.....	70
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	
4.1 โปรแกรมส่วนของการค้นหาคำศัพท์.....	75
4.2 โปรแกรมส่วนของการเพิ่มคำศัพท์.....	79
4.3 โปรแกรมส่วนของการแก้ไขคำศัพท์.....	82
4.4 โปรแกรมส่วนของการลบคำศัพท์.....	84
4.5 โปรแกรมส่วนของฐานข้อมูลเอกสาร.....	85
4.6 ผลการทดลอง.....	88
<b>บทที่ 5 บทสรุป</b>	
5.1 สรุปผลการทดลอง.....	90
5.2 จุดเด่น, จุดด้อย และข้อจำกัดของโครงการที่ทำขึ้น.....	91
5.3 แนวทางการพัฒนาเพิ่มเติม.....	92
เอกสารอ้างอิง.....	93
ประวัติผู้ดำเนินโครงการ.....	94

## สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน.....	3
2.1 Arithmetic Operators.....	21
2.2 Logical Operators.....	22
2.3 Comparison Operators.....	23
2.4 Wildcard.....	23
2.5 Concatenation Operators.....	24
2.6 Aggregate function.....	27
2.7 ค่าอาร์กิวเมนต์ต่างๆที่ใช้ติดต่อกับฐานข้อมูล.....	39
3.1 ฐานข้อมูลชนิดของคำศัพท์ (PartOfSpeech).....	52
3.2 ฐานข้อมูลคำศัพท์ในพจนานุกรมหลัก (Original).....	52
3.3 ฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมหลัก (Original_detail).....	53
3.4 ฐานข้อมูลคำศัพท์ในพจนานุกรมส่วนบุคคล (Personal).....	53
3.5 ฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมส่วนบุคคล (Personal_detail).....	53
4.1 ตารางแสดงผลการทดลอง.....	89
5.1 ตารางสรุปผลการทดลอง.....	90



## สารบัญรูป

รูปที่	หน้า
2.1 การติดต่อระหว่างผู้ใช้กับฐานข้อมูล โดยมี DBMS เป็นตัวกลาง.....	4
2.2 ส่วนประกอบทั้ง 5 ส่วนของโปรแกรม DBMS.....	6
2.3 ขั้นตอนในการพัฒนาระบบสารสนเทศ System Development Life Cycle (SDLC).....	9
2.4 ขั้นตอนการออกแบบฐานข้อมูลตาม DBLC.....	11
2.5 ขั้นตอนการออกแบบฐานข้อมูลทั้ง 3 ระดับ.....	13
2.6 ความสัมพันธ์ระหว่างขั้นตอนของ SDLC และ DBLC.....	15
2.7 เอนทิตีนักศึกษา.....	16
2.8 เอนทิตีอ่อนแอ.....	17
2.9 Property ของเอนทิตีนักศึกษา.....	17
2.10 Composite Property.....	18
2.11 Key Property.....	18
2.12 Multi-Valued Property.....	19
2.13 Derived Property.....	20
2.14 Subtype และ Supertype.....	20
2.15 กลุ่มคลาสใน ADO.NET ที่ทำงานร่วมกับฐานข้อมูล.....	32
2.16 กลุ่มคลาสที่ทำงานร่วมกับฐานข้อมูลแบบเปิดการเชื่อมต่อ (connected database) แยกตามประเภทของ Data Provider.....	33
2.17 การทำงานใน layer แบบ connected database.....	34
2.18 การทำงานใน layer แบบ disconnected database.....	35
2.19 ภาพรวมการทำงานทั้งระบบของคลาสใน layer ของกลุ่มคลาส DataSet.....	35
2.20 เปรียบเทียบ DataSet กับ RecordSet ใน ADO.....	36
2.21- คุณสมบัติของ DataSet ใน ADO.....	36
3.1 แผนภาพระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย.....	47
3.2 System Requirement Model ของระบบฐานข้อมูลพจนานุกรมหลัก.....	48
3.3 System Requirement Model ของระบบฐานข้อมูลพจนานุกรมส่วนบุคคล.....	49
3.4 System Requirement Model ของระบบฐานข้อมูลเอกสาร.....	50
3.5 อี-อาร์ โคอะแกรมแสดงโครงสร้างฐานข้อมูลพจนานุกรม.....	51

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.6 Context Diagram ของระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย...54	54
3.7 Process Decomposition Diagram ของระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปล ภาษาอังกฤษ-ไทย.....55	55
3.8 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบค้นหาคำศัพท์.....56	56
3.9 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบค้นหาคำศัพท์.....57	57
3.10 โครงสร้างการเขียนโปรแกรมแสดงระบบค้นหาคำศัพท์.....58	58
3.11 โครงสร้างการเขียนโปรแกรมแสดงระบบค้นหาตัวอย่างประโยค.....59	59
3.12 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบเพิ่มคำศัพท์.....60	60
3.13 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบเพิ่มคำศัพท์.....61	61
3.14 โครงสร้างการเขียนโปรแกรมแสดงระบบเพิ่มคำศัพท์.....62	62
3.15 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบแก้ไขคำศัพท์.....63	63
3.16 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบแก้ไขคำศัพท์.....64	64
3.17 โครงสร้างการเขียนโปรแกรมแสดงระบบแก้ไขคำศัพท์.....65	65
3.18 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบลบคำศัพท์.....66	66
3.19 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบลบคำศัพท์.....66	66
3.20 โครงสร้างการเขียนโปรแกรมแสดงระบบลบคำศัพท์.....67	67
3.21 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบเพิ่มเอกสาร.....68	68
3.22 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบเพิ่มเอกสาร.....68	68
3.23 โครงสร้างการเขียนโปรแกรมแสดงระบบเพิ่มเอกสาร.....69	69
3.24 โครงสร้างการจัดเรียงตัวอย่างประโยคให้ตำแหน่งคำศัพท์ที่ค้นหาตรงกัน.....70	70
3.25 แบบจำลองการใช้งานโปรแกรม.....71	71
3.26 แบบจำลองการสร้างคำศัพท์ในฐานข้อมูลพจนานุกรมหลัก.....72	72
3.27 แบบจำลองการสร้างคำศัพท์ในฐานข้อมูลพจนานุกรมส่วนบุคคล.....73	73
3.28 แบบจำลองการเพิ่มเอกสารในฐานข้อมูลเอกสาร.....74	74
4.1 โปรแกรมหลักสำหรับการค้นหาคำศัพท์.....75	75
4.2 การค้นหาคำศัพท์โดยเลือกชนิดของคำในพจนานุกรมหลัก.....76	76
4.3 การค้นหาคำศัพท์โดยเลือกปีที่บันทึกในพจนานุกรมส่วนบุคคล.....77	77

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4 การค้นหาตัวอย่างประโยคในฐานข้อมูลเอกสาร.....	78
4.5 การแสดงเอกสารอ้างอิงของตัวอย่างประโยค.....	78
4.6 โปรแกรมส่วนของการเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมหลัก.....	79
4.7 ตัวอย่างของการเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมหลัก.....	80
4.8 โปรแกรมส่วนของการเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมส่วนบุคคล.....	81
4.9 ตัวอย่างของการเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมส่วนบุคคล.....	81
4.10 โปรแกรมส่วนของการแก้ไขคำศัพท์ฐานข้อมูลพจนานุกรมหลัก.....	82
4.11 การเลือกปีที่ยกเลิกเพื่อทำการแก้ไขคำศัพท์ในฐานข้อมูลพจนานุกรมส่วนบุคคล.....	83
4.12 โปรแกรมส่วนของการแก้ไขคำศัพท์ฐานข้อมูลพจนานุกรมส่วนบุคคล.....	83
4.13 การยืนยันการลบ.....	84
4.14 แสดงผลโปรแกรมได้ลบคำศัพท์เรียบร้อยแล้ว.....	84
4.15 เมนูการเพิ่มเอกสาร.....	85
4.16 แบบฟอร์มการเพิ่มเอกสาร.....	85
4.17 การเลือกไฟล์เอกสารที่ต้องการเพิ่มลงในฐานข้อมูล.....	86
4.18 แสดงผลการเพิ่มเอกสารเมื่อเอกสารเข้าสู่ระบบ.....	86
4.19 การลบเอกสาร.....	87
4.20 การค้นหาตัวอย่างประโยค.....	87
4.21 การแสดงตัวอย่างประโยคโดยจัดชิดซ้าย.....	87
4.22 การแสดงตัวอย่างประโยค โดยจัดตามตำแหน่งคำศัพท์.....	88

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบันนี้ ภาษาอังกฤษเป็นภาษาสากลที่มีความสำคัญมากในการติดต่อสื่อสาร ช่วยในการเรียนรู้และเปิดโลกทัศน์ในการเรียนรู้ และมีความจำเป็นในการใช้ชีวิตประจำวันไม่ว่าจะเป็น การใช้ อินเทอร์เน็ต หนังสือเรียนภาษาอังกฤษ การใช้คอมพิวเตอร์ เป็นต้น คนไทยส่วนใหญ่มีปัญหา เกี่ยวกับการใช้ภาษาอังกฤษ โดยเฉพาะการนำคำศัพท์ภาษาอังกฤษไปใช้ให้ถูกต้องตามหลัก ไวยากรณ์ ซึ่งในหนังสือพจนานุกรมและพจนานุกรมอิเล็กทรอนิกส์ ภาษาอังกฤษ-ไทย มีตัวอย่าง ประโยคภาษาอังกฤษ ไม่เพียงพอต่อการทำความเข้าใจ ทำให้ผู้ใช้ ใช้คำศัพท์ได้ไม่ถูกต้องอย่างที่ ควรจะเป็น

ในหนังสือพจนานุกรมและพจนานุกรมอิเล็กทรอนิกส์ภาษาอังกฤษ-ไทยทั่วไป จะช่วยให้ทราบ แต่เพียงความหมายของคำศัพท์และหน้าที่ของคำศัพท์นั้นๆ และอาจจะมีตัวอย่างบ้างเล็กน้อย ซึ่งยังไม่เพียงพอต่อการทำความเข้าใจคำศัพท์คำนั้น ๆ เนื่องจากคำศัพท์บางคำมีหลายความหมาย ขึ้นอยู่กับหน้าที่ของคำและเนื้อเรื่องของประโยคที่อ่านด้วย เช่น ตัวอย่างของคำศัพท์คำว่า Capital (แคปพิเทิล) ถ้าเป็นคำนาม (Noun) จะมีความหมายดังนี้ คือ เงินลงทุน, เมืองหลวง, ตัวอักษรตัวใหญ่, ยอดเสา ถ้าเป็นคำคุณศัพท์ (Adjective) จะมีความหมายดังนี้ คือ ดีเยี่ยม, วิเศษ, สำคัญมาก, สมควร ประหารชีวิต

ตัวอย่างประโยค เช่น

[1] The investor had too little capital to take chances.

- ผู้ลงทุนมีเงินลงทุนน้อยเกินกว่าที่จะฉวยโอกาสนี้ไว้ได้

[2] The capital of the ancient column was richly decorated.

- ยอดสูงสุดสุดของเสาหินโบราณตกแต่งประดับประดาอย่างเต็มที่

[3] Albany is the capital of New York

- อัลบานี เป็นเมืองหลวงของนิวยอร์ก

เมื่อคำศัพท์ภาษาอังกฤษคำเดียวกันแต่ชนิดของคำแตกต่างกันก็จะทำให้ความหมายของคำศัพท์นั้น เปลี่ยนไป หรือถ้าเป็นคำชนิดเดียวกัน เช่นเป็นคำนาม แต่ในเนื้อเรื่องหรือเนื้อหาที่กล่าวถึงไม่ เหมือนกัน ก็อาจจะทำให้คำศัพท์มีความหมายไม่ตรงกันได้

ดังนั้นถ้าหากผู้ใช้ได้เห็นตัวอย่างประโยคที่หลากหลาย จะช่วยให้ผู้ใช้สามารถเรียนรู้และ วิเคราะห์ได้ด้วยตนเอง ก็จะทำให้มีความเข้าใจคำศัพท์คำนั้น ได้มากขึ้น โดยโปรแกรมที่สร้างขึ้นจะ มีการสร้างฐานข้อมูลสำหรับเก็บตัวอย่างประโยคภาษาอังกฤษที่อยู่ในไฟล์เอกสารต่าง ๆ รวมทั้ง

ระบบฐานข้อมูลการแปลภาษาอังกฤษ-ไทย (A Document Database System for English Learning and English-Thai Translation)

โปรแกรมนี้จะช่วยอำนวยความสะดวกแก่ผู้ใช้ ทั้งยังสามารถสร้างพจนานุกรมส่วนบุคคล เฉพาะของผู้ใช้เองได้และคาดหวังว่า ระบบฐานข้อมูลที่พัฒนาขึ้นนี้จะเป็นเครื่องมือเพื่อช่วยสร้าง พจนานุกรมภาษาให้ทันสมัย สามารถก้าวทันการเปลี่ยนแปลงของภาษาซึ่งพัฒนาตามสังคมได้

## 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างโปรแกรมค้นหาความหมายของคำศัพท์ภาษาอังกฤษ และตัวอย่างประโยค ภาษาอังกฤษ เพื่อช่วยให้ผู้ใช้โปรแกรมสามารถใช้คำศัพท์ภาษาอังกฤษได้ถูกต้องตาม หลัก ไวยากรณ์ภาษาอังกฤษ
2. เพื่อสร้างโปรแกรมที่สามารถเพิ่มฐานข้อมูลได้โดยผู้ใช้ และผู้ใช้สามารถสร้างฐานข้อมูล เพื่อใช้เป็นพจนานุกรมส่วนบุคคลของตนเองได้ อีกทั้งยังสามารถสร้างฐานข้อมูลสำหรับจัดเก็บ เอกสารได้
3. เพื่อประยุกต์ใช้ Search Engine
4. เพื่อให้การค้นหาความหมายของคำศัพท์ภาษาอังกฤษ และตัวอย่างประโยคในฐานข้อมูล นั้นค้นหาได้อย่างรวดเร็ว

## 1.3 ขอบข่ายของงานของโครงการ

1. สร้าง Search Engine ที่มีการติดต่อผู้ใช้ผ่าน GUI (Graphic User Interface)
2. สร้างฐานข้อมูลสำหรับเก็บข้อมูลคำศัพท์ แบ่งเป็น 2 ส่วน คือฐานข้อมูลพจนานุกรมหลัก และพจนานุกรมส่วนบุคคล
3. สร้างฐานข้อมูลเอกสารสำหรับเก็บตัวอย่างประโยคภาษาอังกฤษที่อยู่ในไฟล์เอกสารต่างๆ
4. สามารถค้นหาข้อมูลของคำศัพท์ ในส่วนของ ฐานข้อมูลพจนานุกรมหลัก และพจนานุกรม ส่วนบุคคลได้
5. สามารถค้นหาตัวอย่างประโยคของคำศัพท์ภาษาอังกฤษจากไฟล์เอกสารได้
6. สามารถเพิ่ม แก้ไข และลบ ข้อมูลคำศัพท์ภาษาอังกฤษ ทั้งในส่วนของฐานข้อมูล พจนานุกรมหลักและฐานข้อมูลพจนานุกรมส่วนบุคคล ได้
7. สามารถเพิ่มและลบ เอกสารในส่วนของฐานข้อมูลเอกสารได้
8. สามารถเปิดดูเอกสารอ้างอิงของแต่ละประโยคได้

## 1.4 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	พ.ศ. 2550							
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.
1. ศึกษาการเขียนโปรแกรม	←		→					
2. ศึกษาเกี่ยวกับ Data Structure	←	→						
3. ศึกษาเกี่ยวกับ Database	←	→						
4. ศึกษาไวยากรณ์ภาษาอังกฤษ	←	→						
5. ออกแบบฐานข้อมูล			←	→				
6. ออกแบบโปรแกรม				←	→			
7. สร้างโครงสร้างฐานข้อมูล				←	→			
8. เขียนโปรแกรมติดต่อ ฐานข้อมูล					←	→		
9. ทดสอบและโปรแกรม					←	→		
12. สรุปผลและจัดเตรียมเอกสาร							←	→

## 1.5 ผลที่คาดว่าจะได้รับ

1. โปรแกรมสามารถค้นหาคำศัพท์ได้อย่างรวดเร็ว และมีประสิทธิภาพ
2. โปรแกรมสามารถแสดงตัวอย่างประโยคภาษาอังกฤษ ได้ถูกต้องตรงกับคำศัพท์ที่ค้นหา
3. สามารถช่วยพัฒนาทักษะภาษาอังกฤษของผู้ใช้ โดยการวิเคราะห์คำศัพท์
4. ระบบฐานข้อมูลสามารถจัดเก็บคำศัพท์และเอกสารจำนวนมากได้
5. โปรแกรมสามารถพัฒนาต่อได้

## 1.6 งบประมาณที่ใช้

- |                                 |                  |
|---------------------------------|------------------|
| 1. ค่าหนังสือประกอบการทำโครงการ | 1,000 บาท        |
| 2. ค่าตลับหมึกพิมพ์             | 1,000 บาท        |
| 3. ค่ากระดาษสำหรับพิมพ์         | 500 บาท          |
| 4. ค่าเช่าเล่มโครงการ           | 500 บาท          |
| รวมทั้งสิ้น                     | <u>3,000 บาท</u> |

(สามพันบาทถ้วน)

## บทที่ 2

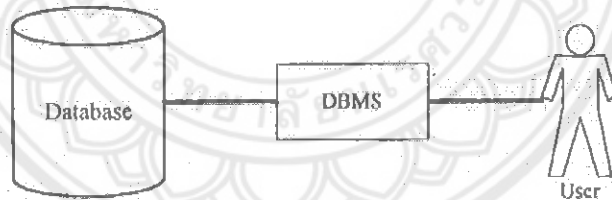
# หลักการและทฤษฎี

### 2.1 ระบบฐานข้อมูล (Database System)

ชีวิตประจำวันในปัจจุบัน ไม่ว่าจะดำเนินการใดๆ มนุษย์จะต้องเกี่ยวข้องกับข้อมูลอย่างใดอย่างหนึ่งเสมอ เช่น การติดต่อราชการที่จะต้องใช้ข้อมูลจากบัตรประชาชนหรือสำมะโนประชากร การติดต่อกับธนาคารที่จะต้องใช้ข้อมูลจากสมุดเงินฝาก เป็นต้น โดยเฉพาะอย่างยิ่ง เมื่อเทคโนโลยีของโลกได้พัฒนาขึ้น จนกระทั่งปัจจุบันที่มีการใช้คอมพิวเตอร์กันอย่างกว้างขวาง ข้อมูลในด้านต่างๆ ซึ่งในอดีตจัดเก็บอยู่บนกระดาษได้ถูกนำมาจัดเก็บไว้ในคอมพิวเตอร์แทน โดยในยุคเริ่มต้นจะจัดเก็บข้อมูลต่างๆ อยู่ในแต่ละแฟ้มข้อมูล แต่เมื่อปริมาณข้อมูลมีจำนวนเพิ่มมากขึ้น ประกอบกับความต้องการใช้ข้อมูลมีเพิ่มขึ้น รวมทั้งข้อมูลได้เปลี่ยนไปเป็นปัจจัยที่มีผลต่อการแข่งขันทางธุรกิจ การจัดเก็บข้อมูลจึงได้เปลี่ยนไป และเกิดคำว่า “ฐานข้อมูล” ขึ้นแทน

#### 2.1.1 Database Management System (DBMS)

ในการติดต่อกับข้อมูลภายในฐานข้อมูลของผู้ใช้ จะต้องกระทำผ่าน โปรแกรมที่เรียกว่า Database Management System (DBMS) ดังรูป



รูปที่ 2.1 การติดต่อระหว่างผู้ใช้งานกับฐานข้อมูล โดยมี DBMS เป็นตัวกลาง

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้งานกับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆ ภายในฐานข้อมูล ซึ่งต่างจากระบบแฟ้มข้อมูลที่หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อกับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มคำสั่ง DML หรือ DDL หรือจะด้วยโปรแกรมต่างๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกโปรแกรม DBMS นำมาแปล (compile) เป็นการกระทำ (Operation) ต่างๆ ภายใต้คำสั่งนั้นๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่างๆ ภายในโปรแกรม DBMS ที่ทำหน้าที่ในการแปลคำสั่งไปเป็นการกระทำต่างๆ ที่จะกระทำกับข้อมูลนั้น ประกอบด้วยส่วนการทำงานต่าง ๆ ดังนี้

### 1. Database Manager

เป็นส่วนที่ทำหน้าที่กำหนดการกระทำต่างๆ กับส่วน File Manager เพื่อไปกระทำกับข้อมูล ที่เก็บอยู่ในฐานข้อมูล (File Manager เป็นส่วนที่ทำหน้าที่บริหารและจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูล ในระดับกายภาพ)

### 2. Query Processor

เป็นส่วนที่ทำหน้าที่แปลงประโยคคำสั่งของ Query Language ให้อยู่ในรูปแบบของคำสั่งที่ Database Manager เข้าใจ

### 3. Data Manipulation Language Precompiler

เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DML ให้อยู่ในรูปแบบที่ส่วน Application Programs Object Code จะนำไปเข้ารหัสเพื่อส่งต่อไปยังส่วน Database Manager ในการแปลประโยคคำสั่งของกลุ่มคำสั่ง DML ของส่วน Data Manipulation Language Precompiler นี้ต้องทำงานร่วมกับส่วน Query Processor

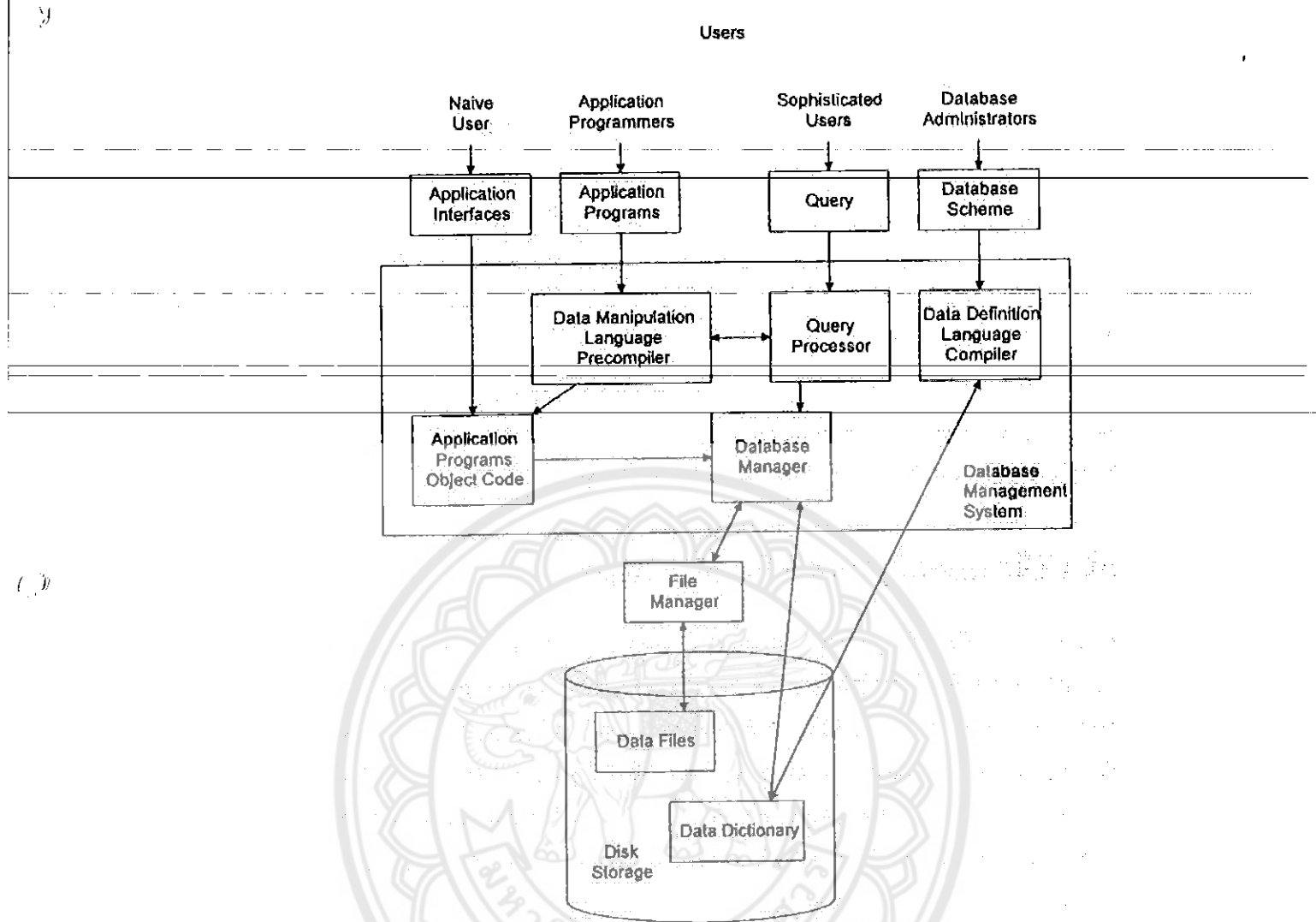
### 4. Data Definition Language Precompiler

เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DDL ให้อยู่ในรูปแบบของ MetaData ที่เก็บอยู่ในส่วน Data dictionary ของฐานข้อมูล (MetaData ได้แก่ รายละเอียดที่บอกถึงโครงสร้างต่างๆ ของข้อมูล)

### 5. Application Programs Object Code

เป็นส่วนที่ทำหน้าที่แปลงคำสั่งต่างๆ ของโปรแกรม รวมทั้งคำสั่งในกลุ่มคำสั่ง DML ที่ส่งต่อมาจากส่วน Data Manipulation Language Precompiler ให้อยู่ในรูปของ Object Code ที่จะส่งต่อไปให้ Database Manager เพื่อกระทำกับข้อมูลในฐานข้อมูล





รูปที่ 2.2 ส่วนประกอบทั้ง 5 ส่วนของโปรแกรม DBMS

โปรแกรม DBMS นี้ได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาทางด้าน Data Independence ที่ไม่มีในระบบแฟ้มข้อมูล ดังนั้นจึงมีความเป็นอิสระจากทั้งตัว Hardware และตัวข้อมูลในฐานข้อมูล กล่าวคือ โปรแกรม DBMS จะมีการทำงานที่ไม่ขึ้นกับรูปแบบ (Platform) ของตัว Hardware ที่นำมาใช้กับระบบฐานข้อมูล รวมทั้งมีรูปแบบในการอ้างถึงข้อมูลที่ไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูล ด้วยการใช้ Query Language ในการติดต่อกับข้อมูลในฐานข้อมูลแทนคำสั่งของภาษาคอมพิวเตอร์ในยุคที่ 3 ส่งผลให้ผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูล โดยไม่จำเป็นต้องทราบถึงประเภทของข้อมูลหรือขนาดของข้อมูลนั้น หรือสามารถกำหนดลำดับที่ของ Field ในการแสดงผลได้โดยไม่ต้องคำนึงถึงลำดับที่จริงของ Field นั้น

### 2.1.2 หน้าที่ของ DBMS

สำหรับหน้าที่ของโปรแกรม DBMS มีดังนี้

1. ทำหน้าที่แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ
2. ทำหน้าที่ในการนำคำสั่งต่างๆ ซึ่งได้รับการแปลแล้ว ไปสั่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ข้อมูล (Retrieve) การจัดเก็บข้อมูล (Update) การลบข้อมูล (Delete) การเพิ่มข้อมูล (Add) เป็นต้น
3. ทำหน้าที่ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะคอยตรวจสอบว่า คำสั่งใดที่สามารถทำงานได้ และคำสั่งใดที่ไม่สามารถทำงานได้
4. ทำหน้าที่รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ
5. ทำหน้าที่เก็บรายละเอียดต่างๆ เกี่ยวกับข้อมูลภายในฐานข้อมูลไว้ใน Data Dictionary ซึ่งรายละเอียดเหล่านี้จึงมักจะถูกเรียกว่า “ข้อมูลของข้อมูล” (Metadata)
6. ทำหน้าที่ควบคุมให้ฐานข้อมูลทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

### 2.1.3 ประโยชน์ของฐานข้อมูล

การให้นำข้อมูลที่มีความสัมพันธ์กันมาใช้ร่วมกันเป็นฐานข้อมูลนั้น จะก่อให้เกิดประโยชน์ดังนี้

1. สามารถลดความซ้ำซ้อนของข้อมูล (Data Redundancy) โดยไม่จำเป็นต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันไว้ในระบบแฟ้มข้อมูลของแต่ละหน่วยงานเหมือนเช่นเดิม แต่สามารถนำข้อมูลมาใช้ร่วมกันในคุณลักษณะ Integrated แทน
2. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูล (Data Inconsistency) เนื่องจากไม่ต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันในหลายแฟ้มข้อมูล ดังนั้นการแก้ไขข้อมูลในแต่ละชุด จะไม่ก่อให้เกิดค่าที่แตกต่างกันได้
3. แต่ละหน่วยงานในองค์กร สามารถใช้ข้อมูลร่วมกันได้
4. สามารถกำหนดให้ข้อมูลมีรูปแบบที่เป็นมาตรฐานเดียวกันได้ เพื่อให้ผู้ใช้ข้อมูลในฐานข้อมูลชุดเดียวกัน สามารถเข้าใจและสื่อสารถึงความหมายเดียวกัน
5. สามารถกำหนดระบบความปลอดภัยให้กับข้อมูลได้ โดยการกำหนดระดับความสามารถในการเรียกใช้ข้อมูลของผู้ใช้แต่ละคน ให้แตกต่างกันตามความรับผิดชอบ
6. สามารถรักษาความถูกต้องของข้อมูลได้ โดยระบบกฎเกณฑ์ในการควบคุมความผิดพลาดที่อาจขึ้นจากการป้อนข้อมูลผิด
7. สามารถตอบสนองต่อความต้องการใช้ข้อมูลในหลายรูปแบบ
8. ทำให้ข้อมูลเป็นอิสระจากโปรแกรมที่ใช้งานข้อมูลนั้น (Data Independence) ซึ่งส่งผลให้ผู้พัฒนาโปรแกรมสามารถแก้ไขโครงสร้างของข้อมูล โดยไม่กระทบต่อโปรแกรมที่เรียกใช้งาน

ข้อมูลนั้น เช่น ในกรณีที่ต้องการเปลี่ยนขนาดของ Field สำหรับระบบเพิ่มข้อมูล จะกระทำได้อย่างยาก เนื่องจากต้องเปลี่ยนแปลงตัวโปรแกรมที่อ้างถึง Field นั้นทั้งหมด ซึ่งต่างจากการใช้ระบบฐานข้อมูลที่มีการอ้างถึงข้อมูล จะไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูล จึงไม่ส่งผลให้ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้นมากนัก

## 2.2 วงจรชีวิตของการพัฒนาระบบฐานข้อมูล (Database Life Cycle)

ฐานข้อมูลนับเป็นส่วนที่สำคัญสำหรับระบบงานสารสนเทศที่ใช้คอมพิวเตอร์ในการประมวลผล (Computer Base Information System) เนื่องจากเป็นส่วนที่ใช้จัดเก็บข้อมูลต่างๆ ซึ่งใช้เป็น Input ระบบงานสารสนเทศในการออกแบบระบบงานสารสนเทศ จึงต้องให้ความสำคัญกับการออกแบบฐานข้อมูลเช่นเดียวกับการออกแบบในส่วนประมวลผล

### 2.2.1 วงจรชีวิตของการพัฒนาระบบงานสารสนเทศ (System Development Life Cycle)

ในการพัฒนาระบบงานสารสนเทศโดยทั่วไป จะดำเนินตามขั้นตอนต่างๆ ที่กำหนดไว้ใน System Development Life Cycle (SDLC) แต่เนื่องจาก SDLC มีอยู่ด้วยกันหลายวิธี (Methodology) ดังนั้นจำนวนและรายละเอียดของขั้นตอนต่างๆ จึงแตกต่างกันไปตาม Methodology ของ SDLC ที่นักพัฒนาระบบงานสารสนเทศเลือกใช้ แต่อย่างไรก็ตามขั้นตอนต่างๆ ของแต่ละ Methodology ก็ไม่ได้แตกต่างกันอย่างสิ้นเชิง เนื่องจาก Methodology ของ SDLC ส่วนใหญ่ จะยึดแนวทางมีการแก้ปัญหาของ Federick Taylor ที่เรียกว่า Scientific Management เป็นหลัก ดังนั้นขั้นตอนต่างๆ ในการพัฒนาระบบงานสารสนเทศของ SDLC ที่กล่าวถึงในที่นี้ จึงเป็นขั้นตอนหลักๆ ที่พบอยู่ใน Methodology ต่างๆ ของ SDLC ซึ่งประกอบด้วยขั้นตอนต่างๆ ดังนี้

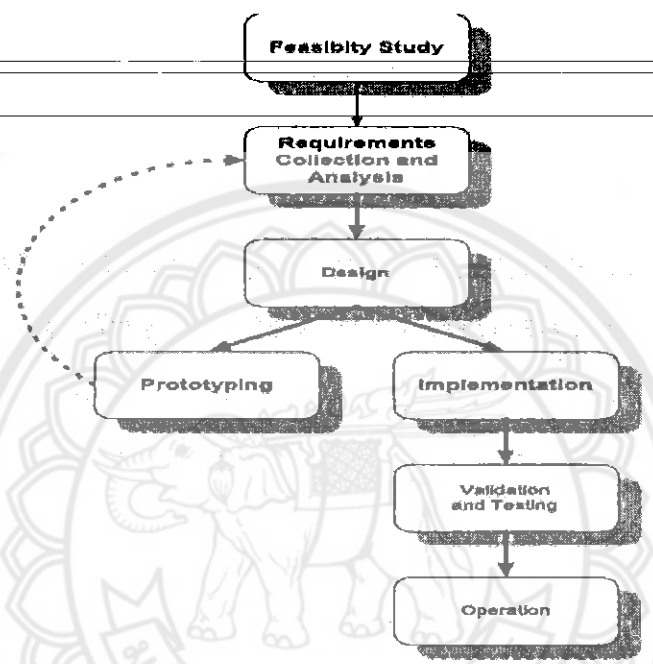
1. Feasibility Study เป็นขั้นตอนที่เกี่ยวข้องกับการประเมินต้นทุนของทางเลือกต่างๆ ของการพัฒนาระบบงานสารสนเทศ เพื่อพิจารณาเลือกทางเลือกในการพัฒนาระบบงานสารสนเทศที่มีความคุ้มค่ามากที่สุด

2. Requirement Collection and Analysis เป็นขั้นตอนในการจัดเก็บรวบรวมความต้องการจากผู้ใช้งานวิเคราะห์ เพื่อจำแนกถึงปัญหาและความต้องการออกเป็นกลุ่ม เพื่อใช้กำหนดขอบเขตให้กับระบบงานสารสนเทศที่จะพัฒนาขึ้น

3. Design เป็นขั้นตอนที่นำเอาปัญหา และความต้องการด้านต่างๆ ที่จำแนกไว้ในขั้นตอนที่ 2 มาใช้ในการออกแบบระบบงานสารสนเทศ

4. Prototyping เป็นขั้นตอนที่นำเอาส่วนต่างๆ ที่ได้ออกแบบไว้ในขั้นตอนที่ 3 มาพัฒนาเป็นต้นแบบของระบบงาน เพื่อนำไปทดลองใช้หาข้อผิดพลาดของระบบงานก่อนนำไปใช้งาน ในกรณีที่มีข้อผิดพลาดเกิดขึ้น รายละเอียดของข้อผิดพลาดต่างๆ จะถูกนำไปเป็นข้อมูลสำหรับขั้นตอนที่ 2 ได้ใหม่

- 5. Implementation เป็นขั้นตอนที่นำเอาระบบงานสารสนเทศที่พัฒนาเสร็จเรียบร้อยแล้วไปทดลองใช้งาน
- 6. Validation and Testing เป็นขั้นตอนของการตรวจสอบความถูกต้องที่พัฒนาขึ้น
- 7. Operation เป็นขั้นตอนสุดท้าย ซึ่งแน่ใจแล้วว่า ระบบงานสารสนเทศที่พัฒนาขึ้น สามารถทำงานได้อย่างถูกต้อง จึงเริ่มนำข้อมูลต่างๆ มาใช้งานจริง



รูปที่ 2.3 ขั้นตอนในการพัฒนาระบบสารสนเทศ System Development Life Cycle (SDLC)

ในแต่ละขั้นตอนของการพัฒนาระบบงานสารสนเทศตาม SDLC นี้ จะไม่ได้แยกออกจากกันอย่างชัดเจน แต่ผลของการทำงานในขั้นตอนหนึ่ง จะสามารถส่งผลกระทบต่อการทำงานในขั้นตอนที่ผ่านมาได้ ซึ่งข้อมูลที่สะท้อนกลับมาระหว่างขั้นตอนเหล่านี้ สามารถนำไปใช้ปรับปรุงและแก้ไขข้อผิดพลาดในการออกแบบของขั้นตอนที่ผ่านมาได้เป็นอย่างดี

ในการพัฒนาระบบงานสารสนเทศขึ้นใช้งาน Zachman (1987) ได้กล่าวถึงสิ่งที่ต้องพิจารณาถึงไว้ 3 ส่วนด้วยกัน คือ ข้อมูล ส่วนประมวลผล และระบบเครือข่ายคอมพิวเตอร์ โดยทั่วไปข้อมูลจะเป็นส่วนที่นักพัฒนาระบบงานสารสนเทศ จะต้องให้ความสำคัญมากกว่าส่วนอื่นๆ เนื่องจากข้อมูลเป็นทั้ง Input และ Output ของส่วนประมวลผล

ดังนั้น จึงมีความจำเป็นที่จะต้องออกแบบในส่วนของคุณสมบัติก่อนที่จะออกแบบในส่วนประมวลผล ถึงแม้ว่าในแง่ความเป็นจริงแล้ว การออกแบบทั้งในส่วนของคุณสมบัติและส่วนประมวลผลมักจะดำเนินการควบคู่กันไป เพื่อเป็นการตรวจสอบความถูกต้องและความสมบูรณ์ของ

และส่วน แต่อย่างไรก็ตาม การออกแบบในส่วนของคุณข้อมูลก็มักจะเริ่มต้นก่อนส่วนประมวลผลเสมอ

ในขั้นตอนของการนำไปใช้งาน (Implementation) ก็เช่นเดียวกัน ส่วนของคุณข้อมูลจะต้องถูกพัฒนาจนสามารถนำไปใช้งานได้จริงก่อน จึงจะสามารถเริ่มต้นพัฒนาในส่วนของโปรแกรมต่างๆ ที่ใช้ในการประมวลผลได้ เนื่องจากโปรแกรมโดยทั่วไปมักจะต้องการอ้างอิงถึงส่วนของคุณข้อมูลสำหรับส่วนระบบเครือข่ายคอมพิวเตอร์ โดยทั่วไปมักจะถูกออกแบบ และพัฒนาหลังจาก 2 ส่วนข้างต้น เนื่องจาก ระบบเครือข่ายคอมพิวเตอร์จะเกี่ยวข้องกับการออกแบบในระดับกายภาพ (Physical Design) ที่เป็นการออกแบบในขั้นตอนสุดท้าย เพื่อนำระบบสารสนเทศที่พัฒนาขึ้นไปใช้งานจริง

### 2.2.2 วงจรชีวิตของการพัฒนาระบบฐานข้อมูล (Database Life Cycle)

วงจรชีวิตของการพัฒนาระบบฐานข้อมูล (Database Life Cycle) หรือที่เรียกอย่างย่อว่า DBLC เป็นขั้นตอนที่กำหนดขึ้น เพื่อใช้เป็นแนวทางในการพัฒนาระบบฐานข้อมูลขึ้นใช้งาน ซึ่งประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. Database Initial study เป็นขั้นตอนแรกของการพัฒนาระบบฐานข้อมูลขึ้นใช้งาน ในขั้นตอนนี้ผู้พัฒนาระบบฐานข้อมูลจะต้องวิเคราะห์ความต้องการต่างๆ ของผู้ใช้ เพื่อกำหนดจุดมุ่งหมาย ปัญหา ขอบเขต และกฎระเบียบต่างๆ ของระบบฐานข้อมูลที่จะพัฒนาขึ้น เพื่อใช้เป็นแนวทางในการออกแบบฐานข้อมูลในขั้นตอนต่อไป

2. Database design เป็นขั้นตอนที่นำเอารายละเอียดต่างๆ ที่ได้จากการวิเคราะห์ในขั้นตอนแรก มากำหนดเป็นแนวทางในการออกแบบฐานข้อมูลขึ้นใช้งานซึ่งแบ่งออกเป็น 3 ระดับ คือการออกแบบฐานข้อมูล ในระดับ Conceptual, Logical และ Physical

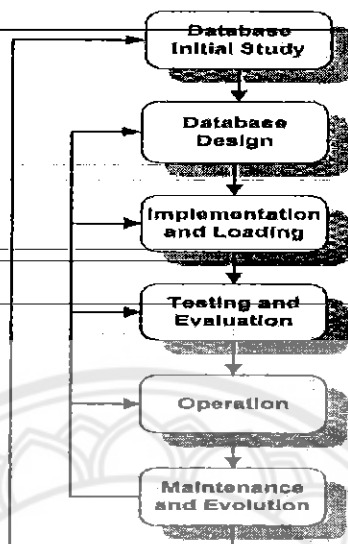
3. Implementation and Loading เป็นขั้นตอนที่นำเอาโครงสร้างต่างๆ ของระบบฐานข้อมูลที่ได้จากการออกแบบในขั้นตอน Database Design มาสร้างเป็นตัวฐานข้อมูลที่จะใช้เก็บข้อมูลจริง รวมทั้งแปลงข้อมูลของระบบงานเดิมให้สามารถนำมาใช้งานในระบบฐานข้อมูลที่พัฒนาขึ้นใหม่ ในกรณีที่ระบบเดิมมีการใช้คอมพิวเตอร์ในการประมวลผล

4. Testing and Evaluation เป็นขั้นตอนของการทดสอบระบบฐานข้อมูลที่พัฒนาขึ้นเพื่อหาข้อผิดพลาดต่างๆ รวมทั้งทำการประเมินความสามารถของระบบฐานข้อมูลนั้นเพื่อนำไปใช้เป็นแนวทางในการปรับปรุง ให้ระบบฐานข้อมูลที่พัฒนาขึ้นสามารถรองรับความต้องการของผู้ใช้ในด้านต่างๆ ได้อย่างถูกต้องและครบถ้วน

5. Operation เป็นขั้นตอนที่นำเอาระบบฐานข้อมูลที่พัฒนาขึ้นเสร็จเรียบร้อยแล้ว ไปใช้งานจริง

6. Maintenance and Evolution เป็นขั้นตอนที่เกิดขึ้นระหว่างการใช้งานระบบฐานข้อมูลจริง เพื่อบำรุง รักษาให้ระบบฐานข้อมูลทำงานได้อย่างมีประสิทธิภาพ รวมทั้งเป็นขั้นตอนของการแก้ไข

และปรับปรุงระบบฐานข้อมูลในกรณีที่มีการเพิ่มหรือเปลี่ยนแปลงความต้องการของผู้ใช้ที่ส่งผลกระทบต่อระบบฐานข้อมูล



รูปที่ 2.4 ขั้นตอนการออกแบบฐานข้อมูลตาม DBLC

การทำงานของแต่ละขั้นตอนในการออกแบบฐานข้อมูลตามวงจรชีวิตของการพัฒนาระบบฐานข้อมูลนี้จะมีลักษณะเช่นเดียวกับวงจรชีวิตของการพัฒนาระบบสารสนเทศ กล่าวคือ รายละเอียดที่ได้จากแต่ละขั้นตอนการพัฒนาระบบฐานข้อมูล สามารถที่จะสะท้อนกลับไปยังการทำงานในขั้นตอนก่อนหน้า ซึ่งจะช่วยปรับปรุง และแก้ไขข้อผิดพลาดในการออกแบบของขั้นตอนที่ผ่านมาได้เป็นอย่างดี

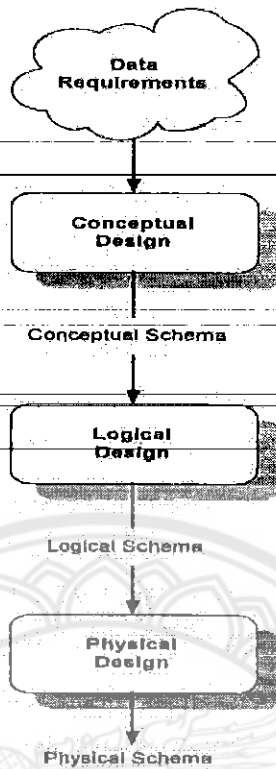
### 2.2.3 ขั้นตอนของการออกแบบฐานข้อมูล

การออกแบบฐานข้อมูล สามารถแบ่งออกได้เป็น 3 ขั้นตอน ดังนี้

- 1- การออกแบบฐานข้อมูลในระดับ **Conceptual** การออกแบบฐานข้อมูลในระดับนี้ จะเป็นการกำหนดโครงร่าง (Schema) เริ่มต้น ที่มีจุดมุ่งหมายเพื่ออธิบายถึงโครงสร้างหลักๆ ของข้อมูลภายในระบบฐานข้อมูล โดยไม่คำนึงถึงฐานข้อมูลที่จะนำมาใช้ว่าจะมีโครงสร้างข้อมูลเป็นแบบ Hierarchical หรือ Network หรือ Relational ดังนั้นผลลัพธ์ที่ได้จากการออกแบบในระดับนี้ จึงเป็นแบบจำลองของข้อมูลที่ประกอบด้วยโครงร่างที่อยู่ในรูปของแนวความคิดซึ่งยังไม่สามารถนำไปใช้งานได้จริง ดังนั้น แบบจำลองของข้อมูลที่ได้จากการออกแบบในขั้นตอนนี้ จึงมักจะถูกเรียกว่า Conceptual Schema แต่อย่างไรก็ตามการออกแบบในระดับนี้กลับมีความสำคัญ เนื่องจากโครงสร้างที่ได้จากการออกแบบในขั้นตอนนี้ จะถูกนำไปใช้ในขั้นตอนอื่นๆ ต่อไปดังนั้นหนังสือทางด้านระบบฐานข้อมูลส่วนใหญ่ จึงมักจะกล่าวถึงการออกแบบฐานข้อมูลในระดับนี้

2. การออกแบบฐานข้อมูลในระดับ **Logical** การออกแบบในระดับนี้ จะเป็นระดับที่ต่อเนื่องมาจากระดับ **Conceptual** กล่าวคือ การออกแบบฐานข้อมูลในระดับนี้ จะอาศัยโครงร่างที่จากการออกแบบในระดับ **Conceptual** มาปรับปรุงให้มีโครงร่างที่เป็นไปตามโครงสร้างข้อมูลของฐานข้อมูลที่จะนำมาใช้งาน โดยจะยังไม่คำนึงถึงผลิตภัณฑ์ทางด้านฐานข้อมูลที่จะนำมาใช้กับระบบฐานข้อมูลที่ออกแบบขึ้นนั้น การออกแบบฐานข้อมูลในขั้นตอนนี้ จำเป็นที่จะต้องปรับปรุงโครงร่างบางอย่างใน **Conceptual Schema** ให้สอดคล้องกับโครงสร้างข้อมูลของฐานข้อมูล ที่จะนำมาใช้งาน ทั้งนี้ก็เนื่องมาจากบางโครงร่างที่ได้ออกแบบไว้ใน **Conceptual Schema** จะไม่สามารถนำมาใช้กับโครงสร้างข้อมูลของฐานข้อมูลที่เลือกมา ใช้งานได้ นอกจากนี้ ในระบบงานที่มีขนาดใหญ่ ซึ่งมักจะนิยมแยกความต้องการของผู้ใช้ออกเป็นความต้องการย่อยๆ แล้วจึงนำแต่ละความต้องการนั้นไปกำหนดเป็น **Conceptual Schema** ซึ่งจะส่งผลให้ระบบงานนั้น จะประกอบด้วย **Conceptual Schema** มากกว่า 1 โครงร่าง ดังนั้น การออกแบบในขั้นตอนนี้ จึงต้องมีการนำเอาแต่ละ **Conceptual Schema** นั้น มาประกอบกันด้วยการออกแบบในขั้นตอนนี้ เนื่องจากเป็นขั้นตอนก่อนที่ จะนำเอาโครงร่างที่ออกแบบขึ้น ไปสร้างเป็นฐานข้อมูลจริง ดังนั้นในขั้นตอนนี้จึงต้องมีการตรวจสอบความถูกต้องของโครงร่างที่ออกแบบขึ้นกับส่วนประมวลผลต่างๆ ที่ออกแบบไว้ รวมทั้งจะต้องแปลงโครงร่างต่างๆ ให้อยู่ในรูปของ **Relation** ในกรณีที่ฐานข้อมูลที่ใช้ มีโครงสร้างข้อมูลแบบ **Relational**

3. การออกแบบฐานข้อมูลในระดับ **Physical** การออกแบบในระดับนี้ จะเป็นขั้นตอนสุดท้ายของการออกแบบฐานข้อมูล การออกแบบฐานข้อมูลในขั้นตอนนี้ จะเป็นการปรับปรุงโครงสร้างของโครงร่างที่ออกแบบขึ้นเช่นเดียวกัน แต่การปรับปรุงโครงสร้างของการออกแบบฐานข้อมูลในขั้นตอนนี้ จะเป็นการนำเอาโครงร่างที่ได้จากการออกแบบในระดับ **Logical** มาปรับปรุงโครงสร้างให้เป็นไปตามโครงสร้างของผลิตภัณฑ์ทางด้านฐานข้อมูลที่จะนำมาใช้งานแทน เนื่องจากแต่ละผลิตภัณฑ์จะมีโครงสร้างในรายละเอียดที่แตกต่างกัน เช่น ประเภทของข้อมูล โครงสร้างในการจัดเก็บ และวิธีการในการเข้าถึงข้อมูล เป็นต้น สำหรับผลลัพธ์ที่ได้จากการออกแบบในระดับนี้ จะ ได้แก่ โครงสร้างของระบบฐานข้อมูลที่สามารถนำไปใช้ในการสร้างตัวฐานข้อมูลจริง



รูปที่ 2.5 ขั้นตอนการออกแบบฐานข้อมูลทั้ง 3 ระดับ

#### 2.2.4 ความสัมพันธ์ระหว่าง SDLC และ DBLC

ขั้นตอนต่างๆ ที่กำหนดไว้ใน SDLC ถูกกำหนดขึ้นเพื่อใช้ในการพัฒนาระบบงานสารสนเทศ แต่เนื่องจากระบบงานสารสนเทศจะต้องประกอบด้วยส่วนประมวลผลและข้อมูล ดังนั้นในการพัฒนาระบบงานสารสนเทศ จึงต้องดำเนินการพัฒนาทั้งในส่วนของประมวลผล และส่วนของข้อมูล กล่าวคือผู้พัฒนาระบบงานสารสนเทศ จะดำเนินงานตามขั้นตอนต่างๆ ของ SDLC เป็นหลัก แต่ถ้าการดำเนินงานภายในขั้นตอนนั้นๆ เกี่ยวข้องกับการพัฒนาในส่วนของข้อมูลแล้ว ผู้พัฒนาจะต้องดำเนินงานตามขั้นตอนที่กำหนดไว้ใน DBLC ประกอบไปด้วย ซึ่งเมื่อนำทั้ง 2 วงจรชีวิตมาเปรียบเทียบกันแล้ว จะพบว่ามียุทธศาสตร์ต่างๆ ดังนี้

1. การดำเนินงานพัฒนาระบบงานสารสนเทศ จะเริ่มต้นจากขั้นตอน Feasibility study ของ SDLC เป็นลำดับแรก

2. การดำเนินงานพัฒนาระบบงานสารสนเทศในขั้นตอนที่ 2 ได้แก่ ขั้นตอน Requirement Collection and Analysis ของ SDLC แต่เนื่องจากการวิเคราะห์ความต้องการของผู้ใช้ที่รวบรวมมาภายในขั้นตอนนี้ มีจุดมุ่งหมายอยู่ 2 ข้อ ดังนี้



ก. เพื่อวิเคราะห์ถึงปัญหาต่างๆที่เกิดขึ้นในระบบงานเดิม และกำหนดขอบเขตของระบบงานใหม่ที่จะพัฒนาขึ้น ซึ่งรายละเอียดของการวิเคราะห์ในส่วนนี้จะถูกนำไปใช้ออกแบบส่วนประมวลผลต่างๆ ภายในระบบงานใหม่

ข. เพื่อวิเคราะห์ถึงรายละเอียดของข้อมูลต่างๆ ที่ปรากฏอยู่ภายในระบบ ก็เพื่อที่จะนำไปใช้ในการออกแบบส่วนของข้อมูลภายในระบบงานใหม่ ซึ่งจุดมุ่งหมายในข้อนี้ ได้แก่ การดำเนินงานในขั้นตอน Database Initial Study ของ DBLC ดังนั้น ภายในขั้นตอนนี้จึงต้องดำเนินการคู่กับการดำเนินงานในขั้นตอน Database Initial Study ของ DBLC

3. การดำเนินงานพัฒนาระบบงานสารสนเทศในขั้นตอนที่ 3 ได้แก่ ขั้นตอน Design ของ SDLC แต่ด้วยเหตุผลเดียวกับขั้นตอน Requirement Collection and Analysis ภายใต้นั้นนี้ จึงต้องออกแบบทั้งในส่วนประมวลผล และส่วนของข้อมูล ดังนี้ ภายในขั้นตอนนี้จึงต้องดำเนินการคู่กับการดำเนินงานในขั้นตอน Database Design ของ DBLC ซึ่งภายในขั้นตอน Database Design ของ DBLC นี้จะต้องดำเนินงาน 3 ขั้นตอน คือ การออกแบบฐานข้อมูลในระดับ Conceptual, Logical และ Physical

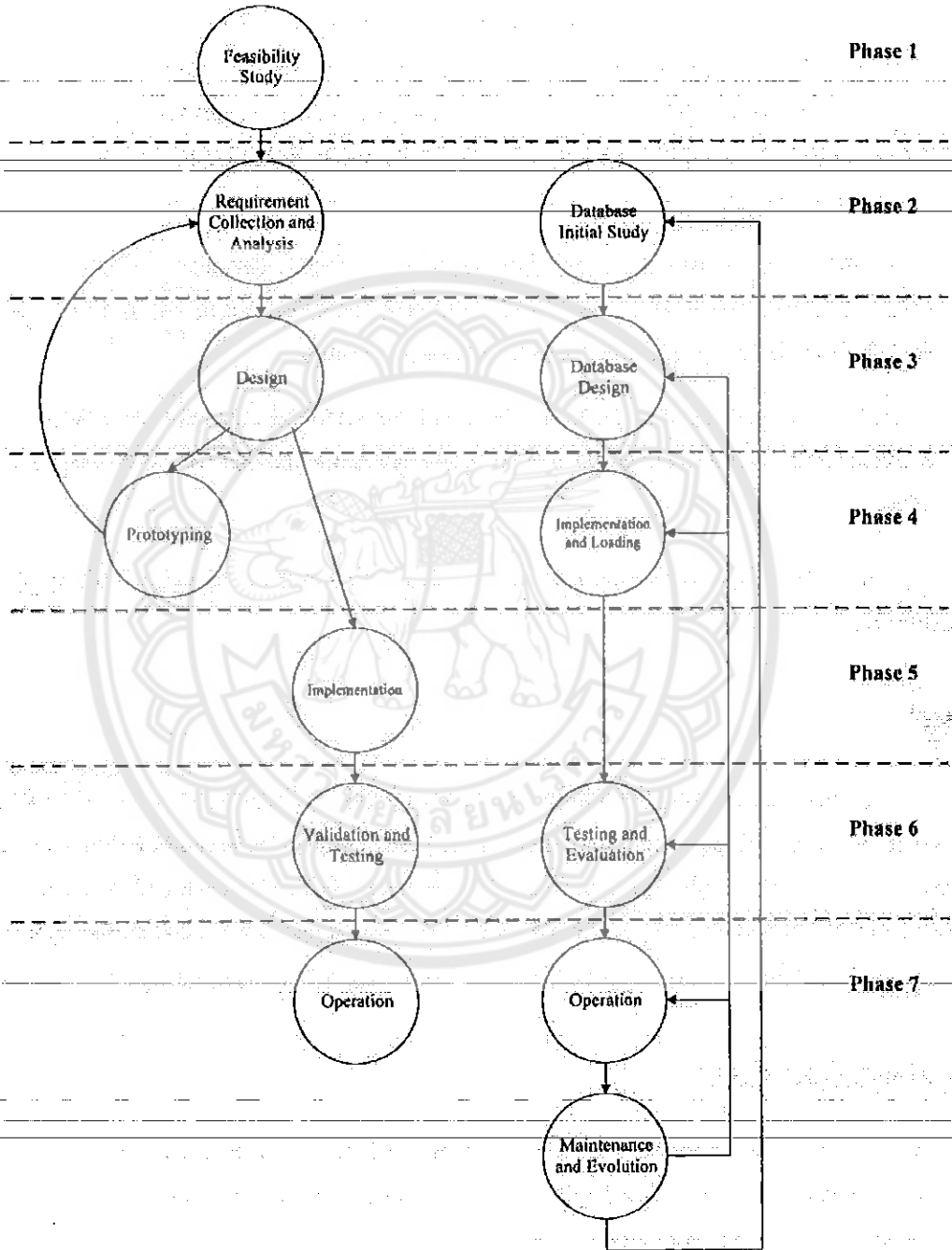
4. การดำเนินงานพัฒนาระบบงานสารสนเทศในขั้นตอนที่ 4 ได้แก่ ขั้นตอน Prototyping ของ SDLC แต่เนื่องจากต้นแบบของโปรแกรม (Prototype) ที่นำเสนอต่อผู้ใช้ จะต้องอาศัยข้อมูลประกอบ ดังนั้น ภายในขั้นตอนนี้จึงต้องดำเนินการคู่กับการดำเนินในขั้นตอน Implementation and Lading ของ DBLC เพื่อที่จะได้นำโครงสร้างข้อมูลที่ได้ออกแบบไว้ มาจัดทำเป็นฐานข้อมูล และสามารถนำไปใช้ประกอบกับต้นแบบของโปรแกรม

5. การดำเนินงานพัฒนาระบบงานสารสนเทศในขั้นตอนที่ 5 ได้แก่ Implementation ของ SDLC ซึ่งเป็นขั้นตอนในการนำระบบงานสารสนเทศที่พัฒนาขึ้น ทั้งในส่วนของโปรแกรมและข้อมูลไปทดลองใช้งาน

6. การดำเนินงานพัฒนาระบบงานสารสนเทศในขั้นตอนที่ 6 ได้แก่ Validation and Testing ของ SDLC แต่เนื่องจากระบบงานสารสนเทศจะมีความถูกต้องได้ ระบบสารสนเทศนั้น จะต้องมีส่วนประมวลผลที่ทำงานได้อย่างถูกต้อง และมีข้อมูลที่ครบถ้วน รวมทั้งมีข้อมูลที่สามารถตอบสนองต่อการประมวลผลต่างๆ ได้เป็นอย่างดี ดังนั้น ในการตรวจสอบความถูกต้องของระบบสารสนเทศ จึงต้องดำเนินการคู่กับ ขั้นตอน Testing and Evaluation ของ DBLC

7. การดำเนินงานพัฒนาระบบงานสารสนเทศในขั้นตอนสุดท้าย ได้แก่ ขั้นตอน Operation ของ SDLC แต่เนื่องจากการนำระบบงานสารสนเทศไปใช้งาน จะต้องดำเนินการติดตั้งทั้งในส่วนของโปรแกรม และจัดเตรียมข้อมูลต่างๆ ดังนั้น ในขั้นตอนนี้ จึงต้องดำเนินการคู่กับขั้นตอน Operation ของ DBLC

8. ส่วนขั้นตอน Maintenance and Evolution ของ DBLC จะเป็นขั้นตอนที่ไม่เกี่ยวข้องกับการพัฒนาระบบงานสารสนเทศ เนื่องจากขั้นตอนนี้เป็นของการบำรุงรักษาฐานข้อมูล เมื่อนำระบบงานสารสนเทศไปใช้งานจริงแล้ว



รูปที่ 2.6 ความสัมพันธ์ระหว่างขั้นตอนของ SDLC และ DBLC

## 2.3 อี-อาร์โมเดล

อี-อาร์โมเดล หรือ Entity-Relationship Model เป็นแบบจำลองข้อมูลที่มีความนิยมอย่างมากสำหรับนำมาใช้เพื่อการออกแบบฐานข้อมูลในระดับแนวคิด โดยอี-อาร์โมเดลเป็นผลงานการพัฒนาของ Peter Pin Shan Chen จาก Massachusetts Institute of Technology ในปี ค.ศ.1976 อี-อาร์โมเดล เป็นแบบจำลองข้อมูลซึ่งแสดงถึงโครงสร้างของฐานข้อมูลที่เป็นอิสระจากซอฟต์แวร์ที่จะใช้ในการพัฒนาฐานข้อมูล รวมทั้งรายละเอียดและความสัมพันธ์ระหว่างข้อมูลในระบบในลักษณะที่เป็นภาพรวม ทำให้เป็นประโยชน์อย่างมากต่อการรวบรวมและวิเคราะห์รายละเอียด ตลอดจนความสัมพันธ์ของข้อมูลต่างๆ โดยอี-อาร์โมเดลมีการใช้สัญลักษณ์ต่างๆ ที่เรียกว่า Entity Relationship Diagram หรือ อี-อาร์โคอะแกรม แทนรูปแบบของข้อมูลเชิงตรรกะขององค์กร จึงทำให้บุคลากรที่เกี่ยวข้องกับระบบฐานข้อมูล สามารถเข้าใจลักษณะของข้อมูลและความสัมพันธ์ระหว่างข้อมูล ได้ง่ายและถูกต้องตรงกัน ระบบที่ได้รับการออกแบบจึงมีความถูกต้องและสอดคล้องกับวัตถุประสงค์ขององค์กร ทั้งนี้ อี-อาร์โมเดลประกอบด้วย 4 ส่วน

### 2.3.1 เอนทิตี (Entity)

เอนทิตี หมายถึง สิ่งที่น่าสนใจสามารถระบุได้ในความเป็นจริง และต้องการเก็บรวบรวมข้อมูลที่เกี่ยวข้องด้วยไว้ในฐานข้อมูล โดยตัวอย่างของเอนทิตีประเภทต่าง ๆ เช่น บุคคล สถานที่ สิ่งของ ในอี-อาร์โคอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้า (Rectangle) แทนหนึ่งเอนทิตี โดยมีชื่อของเอนทิตีนั้น ๆ กำกับอยู่ภายใน เอนทิตีสามารถจำแนกได้เป็น 2 ประเภท คือ เอนทิตีปกติ (regular entity) และเอนทิตีอ่อนแอ (weak entity)

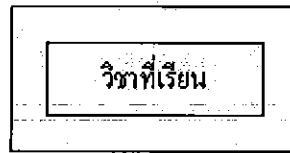
1. เอนทิตีปกติ (regular entity หรือ strong entity) หมายถึง เอนทิตีที่สนใจและต้องการจัดเก็บข้อมูลที่เกี่ยวข้องไว้ในระบบฐานข้อมูล ซึ่งการคงอยู่ของเอนทิตีนี้ไม่เกี่ยวข้องกับเอนทิตีอื่น โดยเอนทิตีนี้สามารถมีคุณสมบัติ Identity ได้ตัวเอง ในอี-อาร์โคอะแกรม สัญลักษณ์ที่ใช้แทนเอนทิตีปกติเป็นเช่นเดียวกับเอนทิตี ดังนั้น เมื่อมีกล่าวถึงเอนทิตีใด ๆ โดยไม่มีการระบุรายละเอียดอื่น จึงหมายความว่าเอนทิตีปกตินั้นเอง

นักศึกษา

รูปที่ 2.7 เอนทิตีนักศึกษา

2. เอนทิตีอ่อนแอ (weak entity) หมายถึง เอนทิตีที่มีการคงอยู่เกี่ยวข้องกับเอนทิตีอื่นในระบบฐานข้อมูล โดยเอนทิตีอื่นที่มีความสัมพันธ์กับเอนทิตีนี้เรียกว่า Parent Entity หรืออาจกล่าวได้ว่า เอนทิตีอ่อนแอจะไม่มีคามหมายหรือไม่สามารถปรากฏในฐานข้อมูลได้ หากปราศจาก Parent Entity ที่มีความสัมพันธ์เกี่ยวข้องกัน ซึ่งสมาชิกของเอนทิตีอ่อนแอจะสามารถมีคุณสมบัติ Identity

ได้ก็ต่อเมื่ออาศัย Property ใด Property หนึ่งของเอนทิตีปกติมาประกอบกับ Property ของเอนทิตีนั้นๆ ในอี-อาร์โคอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้าสองรูปซ้อนกัน (double rectangle) แทนหนึ่งเอนทิตีต่ออันแอ โดยมีชื่อของเอนทิตีต่ออันแอ นั้นๆ กำกับอยู่ภายใน



รูปที่ 2.8 เอนทิตีต่ออันแอ

### 2.3.2 Property

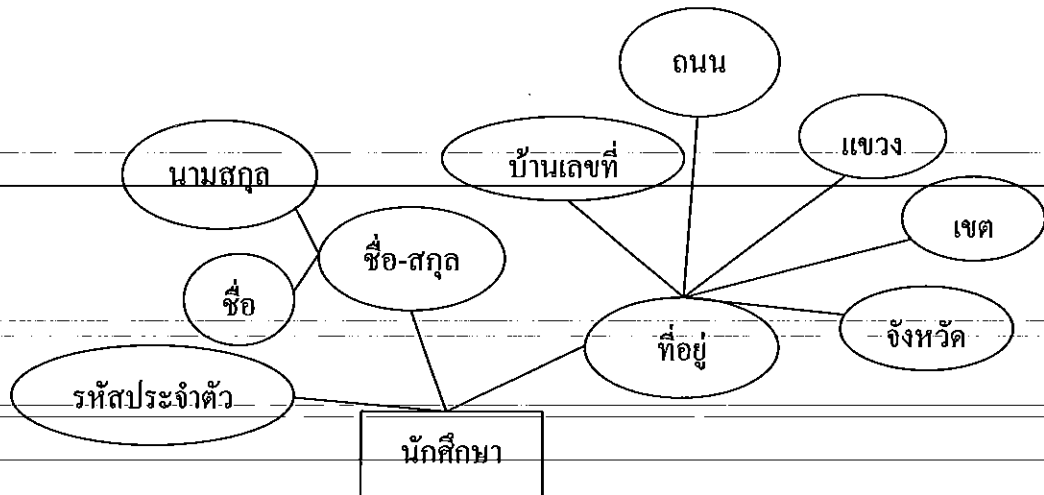
Property หมายถึง ข้อมูลที่แสดงคุณสมบัติหรือคุณลักษณะของเอนทิตีหรือความสัมพันธ์เช่น Property ของเอนทิตีพนักงานประกอบด้วย รหัสประจำตัว ชื่อ-สกุล เพศ รหัสคณะ เป็นต้น ในอี-อาร์โคอะแกรมใช้สัญลักษณ์รูปวงรี แทน Property และเชื่อมต่อกับเอนทิตีที่มี Property นั้นด้วยเส้นตรง Property สามารถจำแนกได้เป็น 6 ประเภท คือ

1. **Simple Property** หมายถึง Property ที่ไม่สามารถแบ่งแยกย่อยลงไปได้อีก เช่น Property ชื่อ นามสกุล และเพศ เป็นต้น ในอี-อาร์โคอะแกรม สัญลักษณ์ที่ใช้แทน Simple Property เป็นเช่นเดียวกับ Property



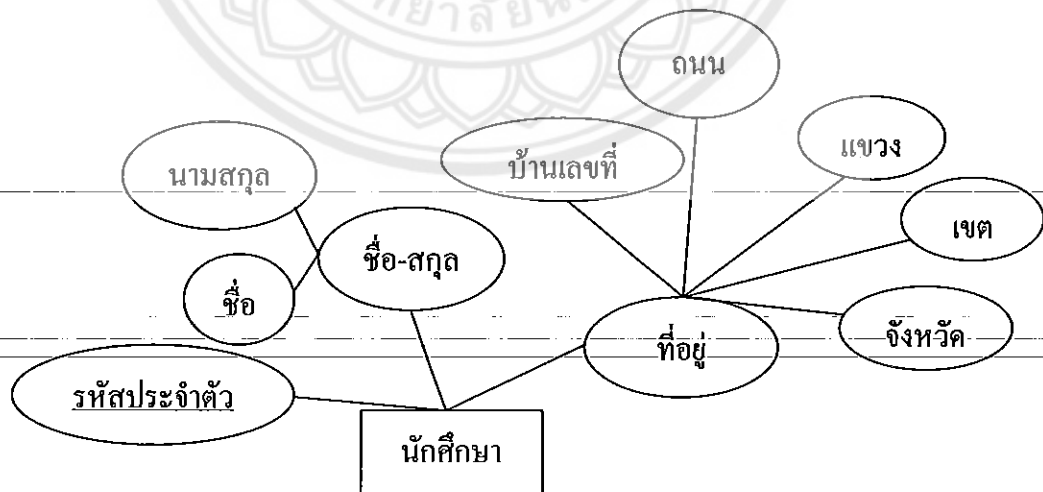
รูปที่ 2.9 Property ของเอนทิตีนักศึกษา

2. **Composite Property** หมายถึง Property ที่มีลักษณะตรงข้ามกับ Simple Property คือสามารถแบ่งแยกย่อยลงไปได้อีก เช่น Property ชื่อ-สกุล สามารถแบ่งได้ 2 Simple Property คือ Property ชื่อ และนามสกุล หรือ Property ที่อยู่ สามารถแบ่งได้ 5 Simple Property คือ Property บ้านเลขที่ ถนน แขวง เขต และจังหวัด เป็นต้น ในอี-อาร์โคอะแกรม ใช้สัญลักษณ์รูปวงรีที่มีชื่อของ Property นั้นกำกับอยู่ภายในแทนหนึ่ง Composite Property และเชื่อมต่อกับ Simple Property ที่จำแนกออกไปด้วยเส้นตรง



รูปที่ 2.10 Composite Property

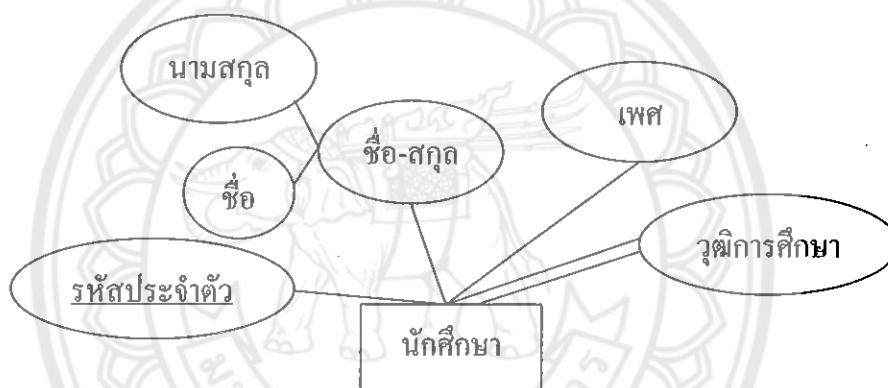
3. **Key Property** หมายถึง Property หรือกลุ่มของ Property ที่มีค่าของข้อมูลในแต่ละสมาชิกของ เอนทิตีไม่ซ้ำกัน ทำให้สามารถระบุความแตกต่างของแต่ละสมาชิกในเอนทิตีได้ เช่น เอนทิตี นักศึกษาประกอบด้วย Property รหัสประจำตัว ชื่อ-สกุล และที่อยู่ โดย Property ที่สามารถบอกความแตกต่างของ นักศึกษาแต่ละคนได้ คือ รหัสประจำตัว ดังนั้น Property รหัสประจำตัวจึงเป็น Key Property ของเอนทิตี นักศึกษา เป็นต้น ใช้สัญลักษณ์รูปวงรีซึ่งภายในกำกับด้วยชื่อของ Property ที่มีการขีดเส้นใต้แทน Key Property และเชื่อมต่อกับเอนทิตีที่มี Property นั้นด้วยเส้นตรง



รูปที่ 2.11 Key Property

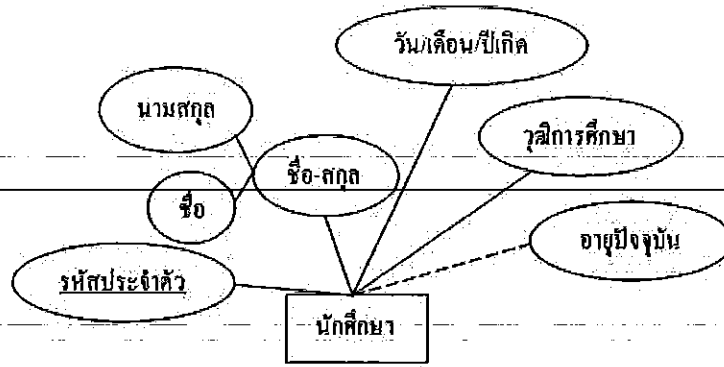
4. **Single-Valued Property** หมายถึง Property ที่มีค่าของข้อมูลในแต่ละสมาชิกของ เอนทิตีได้เพียง ค่าเดียว เช่น บุคคลหนึ่งคนมีเพศเดียว Property เพศจึงเป็น Single-Valued Property หรือพนักงานหนึ่งคนมีเงินเดือนเพียงค่าเดียว Property เงินเดือนจึงเป็น Single-Valued Property เป็นต้น ทั้งนี้ในอี-อาร์ไออะแกรม สัญลักษณ์ที่ใช้แทน Single-Valued Property จะเป็นเช่นเดียวกับ Property

5. **Multi-Valued Property** หมายถึง Property ที่มีลักษณะตรงข้ามกับ Single-Valued Property โดยเป็น Property ที่สามารถมีค่าของข้อมูลในแต่ละสมาชิกของเอนทิตีได้หลายค่า เช่น บุคคลหนึ่งคนอาจมีวุฒิการศึกษาได้หลายระดับ Property วุฒิการศึกษาจึงเป็น Multi-Valued Property หรือบ้านหลังหนึ่งอาจมีหลายเบอร์โทรศัพท์ Property เบอร์โทรศัพท์จึงเป็น Multi-Valued Property เป็นต้น ใช้สัญลักษณ์รูปวงรีซึ่งภายในกำกับด้วยชื่อของ Property เชื่อมต่อกับเอนทิตีที่มี Property นั้นด้วยเส้นขนาน



รูปที่ 2.12 Multi-Valued Property

6. **Derived Property** หมายถึง Property ที่ค่าของข้อมูลในแต่ละสมาชิกของเอนทิตีได้มาจากการนำค่าของข้อมูลใน Property อื่นที่มีอยู่ในแต่ละสมาชิกของเอนทิตีมาทำการคำนวณ ซึ่งโดยทั่วไปไม่จำเป็นต้องจัดเก็บ Property ประเภทนี้ไว้ในระบบฐานข้อมูล เนื่องจาก Property ประเภทนี้มีการเปลี่ยนแปลงค่าของ ข้อมูลในแต่ละสมาชิกของเอนทิตีทุกครั้งเมื่อมีการเปลี่ยนแปลงค่าของข้อมูลใน Property ที่ถูกนำมาคำนวณของแต่ละสมาชิกในเอนทิตี เช่น Property อายุปัจจุบันสามารถคำนวณได้จาก Property วัน/เดือน/ปีเกิด เป็นต้น ในอี-อาร์ไออะแกรม ใช้สัญลักษณ์รูปวงรีที่มีชื่อของ Property นั้นกำกับอยู่ภายในแทนหนึ่ง Derived Property และเชื่อมต่อกับเอนทิตีที่มี Property นั้นด้วยเส้นประ



รูปที่ 2.13 Derived Property

### 2.3.3 ความสัมพันธ์

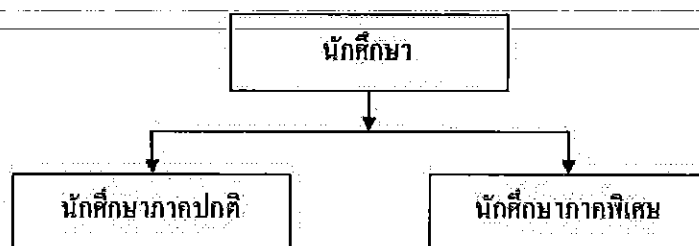
ความสัมพันธ์ หมายถึง เอนทิตีที่แสดงความสัมพันธ์ระหว่างสองเอนทิตีขึ้นไป ซึ่งโดยทั่วไปเป็นความสัมพันธ์ระหว่างเอนทิตีที่มี Property ร่วมกัน โดยแต่ละความสัมพันธ์จะถูกระบุด้วยชื่อที่อธิบายถึงความสัมพันธ์นั้นๆ เช่นความสัมพันธ์สังกัดแสดงความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีคณะ เป็นต้น ในอี-อาร์ไคอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัด (diamond) ที่มีชื่อของความสัมพันธ์นั้นกำกับอยู่ภายในแทนหนึ่งความสัมพันธ์ และเชื่อมต่อกับเอนทิตีที่เกี่ยวข้องกับความสัมพันธ์นั้นด้วยเส้นตรง สามารถแบ่งความสัมพันธ์ได้เป็น 3 ประเภท คือ

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one relationship)
2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one to many relationship)
3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many relationship)

### 2.3.4 Subtype และ Supertype

Supertype คือ เอนทิตีที่สามารถแยกออกเป็นเอนทิตีย่อยๆ ได้ โดยแต่ละสมาชิกในเอนทิตีย่อยนั้นสามารถมีคุณสมบัติ Identity ได้ด้วยตัวเอง เช่น เอนทิตีนักศึกษา

Subtype คือ เอนทิตีย่อยที่แยกออกมาจาก-Supertype เช่น-เอนทิตีนักศึกษากาปกติและนักศึกษากาพิเศษ



รูปที่ 2.14 Subtype และ Supertype

## 2.4 การจัดการฐานข้อมูลด้วยภาษา SQL

### 2.4.1 รู้จักภาษา SQL

SQL ย่อมาจาก Structure Query Language เป็นภาษาที่ใช้ร่วมกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System – RDBMS) ที่ได้รับความนิยมมากที่สุดภาษาหนึ่ง พัฒนาโดยบริษัท IBM Research ในปี ค.ศ. 1974 เรียกภาษานี้ว่า SEQUEL ย่อมาจาก Structured-English Query Language เนื่องจากโครงสร้างภาษาจะคล้ายคลึงกับประโยคภาษาอังกฤษมาก หลังจากนั้นได้ถูกพัฒนาอย่างต่อเนื่องและนำไปใช้เป็นภาษาสำหรับระบบจัดการฐานข้อมูลของบริษัทต่างๆอย่างแพร่หลาย จนต้องกำหนดมาตรฐาน ANSI-SQL ขึ้น โดยความร่วมมือระหว่าง ANSI (American Nation Standards Institute) และ ISO (International Standards Organization) เพื่อให้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ต่างระบบกันสามารถใช้ชุดคำสั่งร่วมกันได้ มาตรฐานที่ใช้ปัจจุบันคือ ANSI SQL-92

คำสั่งในภาษา SQL จะแบ่งออกเป็น 2 กลุ่มใหญ่ๆคือ กลุ่มคำสั่งจัดการข้อมูลในฐานข้อมูล (Data Manipulation Language – DML) และกลุ่มคำสั่งที่ใช้จัดการโครงสร้างฐานข้อมูล (Data Definition Language – DDL)

### 2.4.2 โอเปอเรเตอร์ที่ใช้ในภาษา SQL

โอเปอเรเตอร์ (Operator) หรือตัวดำเนินการที่ใช้ร่วมกับภาษา SQL ประกอบด้วยโอเปอเรเตอร์ที่ใช้คำนวณทางคณิตศาสตร์ (Arithmetic Operators), ตรวจสอบค่าทางตรรกะ (Logical Operators), เปรียบเทียบค่า (Comparison Operators) และเชื่อมข้อความ (Concatenation Operators)

ตารางที่ 2.1 Arithmetic Operators

โอเปอเรเตอร์	หน้าที่	ตัวอย่าง	ผลลัพธ์
+	บวก	2+2	4
-	ลบ	2-3	-1
-	กลับบวกเป็นลบ, ลบเป็นบวก	-(1+3)	-4
*	คูณ	21*3	63
/	หารธรรมดา	11/4	2.75
\	หารปัดเศษทิ้ง	11\4	2
Mod	หารเอาผลลัพธ์เฉพาะเศษ(modulo)	11 Mod 4	3
^	ยกกำลัง	(-5)^3	-125



ตารางที่ 2.2 Logical Operators

โอเปอเรเตอร์	ความหมาย	ตัวอย่าง	ผลลัพธ์
Not	กลับเป็นตรงข้าม	Nor False	True
		Not True	False
And	และ	False And False	False
		False And True	False
		True And False	False
		True And True	True
Or	หรือ	False Or False	False
		False Or True	True
		True Or False	True
		True Or True	True
Xor	Exclusive Or	False Xor False	False
		False Xor True	True
		True Xor False	True
		True Xor True	False
Eqv	Equivalent	False Eqv False	True
		False Eqv True	False
		True Eqv False	False
		True Eqv True	True
Imp	Imply	False Imp False	True
		False Imp True	True
		True Imp False	False
		True Imp True	True

ตารางที่ 2.3 Comparison Operators

โอเปอเรเตอร์	ความหมาย	ตัวอย่าง	ผลลัพธ์
<	น้อยกว่า	5<5	False
<=	น้อยกว่าหรือเท่ากับ	4.5<=5	True
>	มากกว่า	4>4	False
>=	มากกว่าหรือเท่ากับ	4>=4	True
=	เท่ากับ	2=2.5	False
<>	ไม่เท่ากับ	4<>	True
Is	ใช่หรือไม่? (ใช้ตรวจสอบ Null)	A Is Null	True ถ้า A เป็น Null
Is Not	ไม่ใช่? (ใช้ตรวจสอบ Null)	A Is Not Null	True ถ้า A ไม่เป็น Null
Like	ใช้เปรียบเทียบข้อความใดๆกับรูปแบบ (pattern) ที่กำหนดด้วยสัญลักษณ์ wildcard	Like "sc%"	ข้อความที่ขึ้นต้นด้วย sc เช่น science, screen, school เป็นต้น
In	อยู่ในชุดที่กำหนดหรือไม่	"Avon" In ("Glos", "Som", "Avon")	True
Between ... And	อยู่ในช่วงที่กำหนดหรือไม่	100 Between 200 And 300	False

ตารางที่ 2.4 Wildcard

สัญลักษณ์	ความหมาย	ตัวอย่าง	ผลลัพธ์
%	แทนตัวอักษรหรือตัวเลขใดๆ (0-9) จำนวนกี่ตัวก็ได้	Like "Happy%"	ข้อความใดๆที่ขึ้นต้นด้วย Happy เช่น HAPPY, Happy New Year
_	แทนตัวอักษรหรือตัวเลขใดๆ เพียง 1 ตัวเท่านั้น	Like "b_"	อักขระตัวที่ 2 และ 3 จะเป็นตัวอักษรหรือตัวเลขก็ได้ เช่น boy, b12, Bx9

ตารางที่ 2.4 (ต่อ)

สัญลักษณ์	ความหมาย	ตัวอย่าง	ผลลัพธ์
#	แทนตัวเลขใดๆเพียง1ตัว	Like "12#0#"	อักขระตัวที่ 3 และ 5 จะเป็นตัวเลขเท่านั้น เช่น 12007, 12904
[ ]	แทนตัวอักษรหรือตัวเลขใดๆที่ ระบุไว้ใน [ ]	Like "s[o, u, 1-3]n"	อักขระตัวที่ 2 ต้องเป็น o, u, 1, 2 หรือ 3 เช่น son, SUN, s1n, S2N
[! ]	อักษรหรือตัวเลขใดๆก็ได้ ยกเว้นที่ระบุต่อจาก เครื่องหมาย ! ใน [ ]	Like "a[!a-c]%" Like "a[!a, c]%"	อักขระตัวที่ 2 ต้อง ไม่ใช่ a, b หรือ c เช่น aunt, AirPlane, ant, a888 อักขระตัวที่ 2 ต้อง ไม่ใช่ a หรือ c เช่น audio, abstract, a9xy เป็นต้น

ตารางที่ 2.5 Concatenation Operators

สัญลักษณ์	ความหมาย	ตัวอย่าง	ผลลัพธ์
&	เชื่อมข้อความกับข้อความหรือ ข้อความกับตัวเลขเข้าด้วยกัน	"Happy" & " " & "Birthday" 31 & "ธันวาคม" & 2545	"Happy Birthday"  "31 ธันวาคม 2545"
+	เชื่อมข้อความกับข้อความ เท่านั้น	"Happy" + " " + "Birthday" 31 + "ธันวาคม" + 2545	"Happy Birthday"  แสดงข้อความ ผิดพลาด

14941688

2.4.3 คำสั่ง SELECT

เป็นคำสั่งที่ใช้สอบถามหรือคัดลอกข้อมูล (Data Query) จากหนึ่งเทเบิลหรือหลายเทเบิลที่มีความสัมพันธ์กัน โดยกำหนดเงื่อนไขเพื่อค้นหาข้อมูลที่ต้องการ รวมทั้งสามารถเรียงลำดับ จัดกลุ่มข้อมูล และใช้ฟังก์ชันคำนวณหาค่าผลสรุปของข้อมูลในแต่ละกลุ่มได้

รูปแบบ **SELECT** [predicate] [\* | tablename. \* | [tablename.] fieldname1 [AS alias1] [, [tablename.]fieldname2 [AS alias2] [ , ... ] ] }

2/5  
82315

**FROM** tablename [ , ... ] [IN externaldatabase]

2550

**WHERE** Criteria

เมื่อ	predicate	หมายถึงคำสั่งย่อย ALL, TOP, DISTINCT และ DISTINCTROW ใช้กำหนดจำนวนเรคอร์ดหรือรูปแบบการแสดงผล ถ้าไม่กำหนดจะหมายถึง ALL คือแสดงทุกเรคอร์ด
	tablename	หมายถึงชื่อเทเบิลที่จะนำมาเลือกข้อมูล
	*	หมายถึงให้เลือกข้อมูลจากทุกฟิลด์ของเทเบิลนั้น
	fieldname1, fieldname2	หมายถึงชื่อฟิลด์ที่จะนำมาเลือกข้อมูล
	alias1, alias2, ...	หมายถึงกำหนดข้อความที่จะให้ แสดงที่หัวคอลัมน์ของเทเบิลแทน แสดงด้วยชื่อฟิลด์ (ไม่ทำให้ชื่อฟิลด์ในโครงสร้างเทเบิลเปลี่ยนแปลง) ใช้ร่วมกับคำสั่ง AS
	externaldatabase	หมายถึงชื่อไฟล์ฐานข้อมูลภายนอก (ไม่ได้เปิดใช้งานขณะนั้น) ซึ่งเก็บเทเบิลที่ระบุในอาร์กิวเมนต์ tablename
	criteria	หมายถึงเงื่อนไขที่ใช้เลือกข้อมูลจากเทเบิล

1. เลือกข้อมูลไม่ซ้ำกันด้วย **DISTINCT** และ **DISTINCTROW**

การเลือกข้อมูลจากเทเบิลอาจพบรายการที่แสดงข้อมูลซ้ำกันมากกว่า 1 รายการได้ หากต้องการให้แสดงรายการข้อมูลที่ไม่ซ้ำกันเพียงรายการเดียว จะต้องกำหนดคำสั่งย่อย **DISTINCT** และ **DISTINCTROW** ไว้หลังคำสั่ง **SELECT**

รูปแบบ **SELECT** {**DISTINCT**|**DISTINCTROW**} {\*-|tablename. \* | [tablename.] fieldname1 [AS alias1] [, [tablename.]fieldname2 [AS alias2] [ , ... ] ] }

**FROM** tablename [ , ... ] [IN externaldatabase]

**WHERE** Criteria

<b>DISTINCT</b>	ถ้าข้อมูล que เลือกมามีรายการข้อมูลซ้ำกัน (หมายถึงซ้ำกันเฉพาะฟิลด์ที่เลือกมเท่านั้น ไม่จำเป็นต้องซ้ำกันทุกฟิลด์ในเรคอร์ดนั้น) และไม่ต้องการให้แสดงข้อมูลซ้ำกันเลย ให้ใช้คำสั่งนี้กรองรายการที่ซ้ำออกไป
<b>DISTINCTROW</b>	ถ้าข้อมูล que เลือกมามีรายการข้อมูลซ้ำกันทั้งรายการ (ซ้ำกันทุกฟิลด์ใน

เทเบิลนั้นด้วย) ให้ใช้คำสั่งนี้กรองรายการที่ซ้ำออกไป

## 2. เรียงลำดับข้อมูลด้วย ORDER BY

สามารถนำข้อมูลที่ได้จาก ประโยคคำสั่ง SELECT มาเรียงลำดับจากน้อยไปมาก (Ascending) หรือจากมากไปน้อย (Descending) โดยกำหนดฟิลด์ที่เรียงลำดับข้อมูลไว้หลังคำสั่ง ORDER BY

รูปแบบ SELECT fieldlist

FROM tablename

WHERE criteria

[ORDER BY fieldname1 [ASC | DESC] [,fieldname2 [ASC | DESC] ] [, ...]]

เมื่อ	fieldlist	หมายถึงรายชื่อฟิลด์ที่ต้องการนำข้อมูลมาแสดง
	tablename	หมายถึงชื่อเทเบิลที่เป็นแหล่งข้อมูล
	criteria	หมายถึงเงื่อนไขที่ใช้เลือกข้อมูลจากเทเบิล
	fieldname1, fieldname2,...	หมายถึงชื่อฟิลด์ที่จะนำข้อมูลมาเรียงลำดับ
	ASC	หมายถึงเรียงลำดับข้อมูลจากน้อยไปมาก (ระบุหรือไม่ก็ได้)
	DESC	หมายถึงเรียงลำดับจากมากไปน้อย (ต้องระบุ)

## 3. เลือกข้อมูลจากหลายเทเบิลที่สัมพันธ์กัน

การเลือกข้อมูลที่มาจากเทเบิลมากกว่า 1 เทเบิล และข้อมูลในแต่ละเทเบิล มีความสัมพันธ์กัน โดยความสัมพันธ์นั้นถูกเชื่อมโยงจาก Primary Key ของเทเบิลหนึ่ง ไปยัง Foreign Key ของอีกเทเบิลหนึ่งนั้น

ก. **INNER JOIN** ข้อมูลใดๆในเทเบิลหนึ่งสามารถจับคู่กับข้อมูลในอีกเทเบิลหนึ่งตามเงื่อนไขที่กำหนด เรียกฟิลด์ที่ใช้ในการจับคู่ว่า Joined field

ข. **LEFT OUTER JOIN** ข้อมูลทั้งหมดในเทเบิลทางซ้ายของความสัมพันธ์จะถูกนำมาจับคู่กับเทเบิลที่อยู่ทางขวาของความสัมพันธ์ โดยเรคอร์ดในเทเบิลทางซ้ายที่หาคู่ได้จะแสดงข้อมูลตามที่กำหนด ส่วนเรคอร์ดที่หาคู่ไม่ได้จะแสดงค่าว่าง (Null) เรียกการ JOIN แบบนี้ได้อีกอย่างหนึ่งว่า LEFT JOIN

ค. **RIGHT OUTER JOIN** ข้อมูลทั้งหมดในเทเบิลทางขวาของความสัมพันธ์จะถูกนำมาจับคู่กับเทเบิลที่อยู่ทางซ้ายของความสัมพันธ์ โดยเรคอร์ดในเทเบิลทางขวาที่หาคู่ได้จะแสดงข้อมูลตามที่กำหนด ส่วนเรคอร์ดที่หาคู่ไม่ได้จะแสดงค่าว่าง เรียกการ JOIN แบบนี้ได้อีกอย่างหนึ่งว่า RIGHT JOIN

รูปแบบ SELECT ... FROM tablename1

{INNER JOIN | LEFT JOIN | RIGHT JOIN} tablename2 ON tablename1.fieldname1

comppor tablename2.fieldname2

## WHERE criteria

เมื่อ	tablename1, tablename2	หมายถึงชื่อเทเบิลที่นำข้อมูลมาจับคู่กัน
	fieldname1, fieldname2	หมายถึงชื่อฟิลด์ที่จะนำมาจับคู่ โดยฟิลด์ทั้ง 2 ต้องเก็บข้อมูลประเภทเดียวกัน (ชื่อฟิลด์ไม่จำเป็นต้องเหมือนกัน)
	compopr	หมายถึงตัวดำเนินการเปรียบเทียบ (Comparison Operator) ที่ใช้ในนิพจน์เงื่อนไข เช่น =, <, >, <=, >=, และ <

## 4. คำนวณค่าผลสรุปด้วย Aggregate Function

เป็นฟังก์ชันที่สามารถนำมาใช้ร่วมกับคำสั่ง SELECT เพื่อคำนวณค่าผลสรุปข้อมูล เช่น ค่าผลรวม (Sum), ค่าเฉลี่ย (Average), ค่าสูงสุด/ต่ำสุด (Max/Min), คำนับจำนวน (Count) และค่าทางสถิติเช่น ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation) และค่าความแปรปรวน (Variance)

รูปแบบ SELECT function(fieldname) AS virtualField...

เมื่อ	function	หมายถึงชื่อฟังก์ชันผลสรุปที่จะนำมาใช้ร่วมกับคำสั่ง SELECT
	fieldname	หมายถึงชื่อฟิลด์ที่ต้องการนำข้อมูลมาคำนวณค่าผลสรุป
	virtualField	หมายถึงชื่อฟิลด์เสมือนที่ใช้เก็บผลลัพธ์จากการคำนวณ (ฟิลด์เสมือนจะเก็บผลลัพธ์ชั่วคราวเท่านั้นและไม่มีผลกับโครงสร้างเทเบิล)

## ตารางที่ 2.6 Aggregate function

ฟังก์ชัน	การทำงาน
Sum(expr)	คำนวณหาผลรวมของข้อมูลในฟิลด์หรือพจน์ที่กำหนดให้เมื่อ expr คือชื่อฟิลด์แบบตัวเลขหรือนิพจน์ที่นำมาหาผลรวม ตัวอย่างเช่น Sum(Salary) As Total คำนวณหาผลรวมของเงินเดือนพนักงานทุกคนแล้วเก็บไว้ในฟิลด์เสมือนชื่อ Total
Count (expr)	นับจำนวนเรคอร์ดตามเงื่อนไขที่กำหนดให้ เมื่อ expr คือชื่อฟิลด์ที่ใช้เป็นเงื่อนไขในการนับจำนวน - ถ้ากำหนดฟิลด์เงื่อนไขมากกว่า 1 ฟิลด์ ให้คั่นแต่ละฟิลด์ด้วยเครื่องหมาย & - ถ้าข้อมูลในฟิลด์เงื่อนไขมีค่าว่างในเรคอร์ดใด เรคอร์ดนั้นจะ ไม่ถูกนับรวม - ถ้าต้องการให้เรคอร์ดที่มีค่าว่างถูกนับรวมด้วย ให้ใส่เครื่องหมาย * แทนชื่อฟิลด์ Count(*) นับจำนวนเรคอร์ดทั้งหมด (รวมเรคอร์ดที่เป็นค่าว่าง) Count(CustomerID&CustomerName) นับเฉพาะเรคอร์ดที่มีค่าในฟิลด์ CustomerID หรือ CutomerName ไม่เป็นค่าว่าง

ตารางที่ 2.6 (ต่อ)

ฟังก์ชัน	การทำงาน
Count	นับจำนวนเรคอร์ดตามเงื่อนไขที่กำหนดให้
(expr)	เมื่อ expr คือชื่อฟิลด์ที่ใช้เป็นเงื่อนไขในการนับจำนวน <ul style="list-style-type: none"> <li>- ถ้ากำหนดฟิลด์เงื่อนไขมากกว่า 1 ฟิลด์ ให้คั่นแต่ละฟิลด์ด้วยเครื่องหมาย &amp;</li> <li>- ถ้าข้อมูลในฟิลด์เงื่อนไขมีค่าว่างในเรคอร์ดใด เรคอร์ดนั้นจะไม่ถูกนับรวม</li> <li>- ถ้าต้องการให้เรคอร์ดที่มีค่าว่างถูกนับรวมด้วย ให้ใส่เครื่องหมาย * แทนชื่อฟิลด์</li> </ul> ตัวอย่างเช่น
	Count(*) นับจำนวนเรคอร์ดทั้งหมด (รวมเรคอร์ดที่เป็นค่าว่าง) Count(CustomerID&CustomerName) นับเฉพาะเรคอร์ดที่มีค่าในฟิลด์ CustomerID หรือ CustomerName ไม่เป็นค่าว่าง
Min(expr)	หาข้อมูลที่มีค่าต่ำสุดในฟิลด์ที่กำหนดให้ เมื่อ expr คือชื่อฟิลด์หรือนิพจน์ที่ต้องการนำมาหาข้อมูลที่มีค่าต่ำสุด ตัวอย่างเช่น Min(Salary) นำข้อมูลในฟิลด์ Salary มาหาอัตราเงินเดือนต่ำสุด Min(Quantity *UnitPrice) นำผลคูณระหว่างจำนวนสินค้ากับราคาสินค้าต่อหน่วยมาหามูลค่าสินค้าต่ำสุด
Max (expr)	หาข้อมูลที่มีค่าสูงสุดในฟิลด์ที่กำหนดให้ เมื่อ expr คือชื่อฟิลด์หรือนิพจน์ที่ต้องการนำมาหาข้อมูลที่มีค่าสูงสุด ตัวอย่างเช่น Max (Salary) นำข้อมูลในฟิลด์ Salary มาหาอัตราเงินเดือนสูงสุด Max(Quantity *UnitPrice) นำผลคูณระหว่างจำนวนสินค้ากับราคาสินค้าต่อหน่วยมาหามูลค่าสินค้าสูงสุด
Avg(expr)	หาค่าเฉลี่ยจากข้อมูลในฟิลด์ที่กำหนดให้ เมื่อ expr คือชื่อฟิลด์หรือนิพจน์ที่ต้องการนำมาหาค่าเฉลี่ย ตัวอย่างเช่น Avg(Salary) นำข้อมูลในฟิลด์ Salary มาหาเงินเดือนเฉลี่ยของพนักงาน Avg(Quantity *UnitPrice) นำผลคูณระหว่างจำนวนสินค้ากับราคาสินค้าต่อหน่วยมาหามูลค่าเฉลี่ย

ตารางที่ 2.6 (ต่อ)

ฟังก์ชัน	การทำงาน
StDev(expr)	หาค่าส่วนเบี่ยงเบนมาตรฐานของตัวอย่าง (Sample) ข้อมูลประชากรในฟิลด์ที่กำหนดให้ (ถ้าจำนวนเรคอร์ดทั้งหมดน้อยกว่า 2 เรคอร์ด จะได้ผลลัพธ์เป็นค่าว่าง (Null)) เมื่อ expr คือชื่อฟิลด์แบบตัวเลขหรือนิพจน์ที่นำมาหาค่าส่วนเบี่ยงเบนมาตรฐาน
StDevP(expr)	หาค่าส่วนเบี่ยงเบนมาตรฐานของข้อมูลประชากร (Population) ในฟิลด์ที่กำหนดให้ (ถ้าจำนวนเรคอร์ดเป็นศูนย์หรือไม่ปรากฏเรคอร์ดที่จะนำมาคำนวณ จะได้ผลลัพธ์เป็นค่าว่าง) เมื่อ expr คือชื่อฟิลด์แบบตัวเลขหรือนิพจน์ที่นำมาหาค่าส่วนเบี่ยงเบนมาตรฐาน
Var(expr)	หาค่าความแปรปรวนของตัวอย่างข้อมูลประชากรในฟิลด์ที่กำหนดให้ (ถ้าเรคอร์ดทั้งหมดน้อยกว่า 2 เรคอร์ด จะได้ผลลัพธ์เป็นค่าว่าง)
VarP(expr)	หาค่าความแปรปรวนของข้อมูลประชากรในฟิลด์ที่กำหนดให้ (ถ้าเรคอร์ดทั้งหมดน้อยกว่า 2 เรคอร์ด จะได้ผลลัพธ์เป็นค่าว่าง) เมื่อ expr คือชื่อฟิลด์แบบตัวเลขหรือนิพจน์ที่นำมาหาค่าความแปรปรวน

### 5. จัดกลุ่มข้อมูลด้วย Group By

เป็นคำสั่งที่ใช้จัดกลุ่มข้อมูล เพื่อคำนวณค่าผลสรุปในแต่ละกลุ่มเปรียบเทียบกัน เช่น คำนวณ หาจำนวนพนักงานแต่ละแผนกพร้อมอัตราเงินเดือนสูงสุด/ต่ำสุดของแผนก คำนวณหายอดรวมเงินเดือนและเงินเดือนเฉลี่ยของพนักงานแต่ละแผนก เป็นต้น คำสั่ง Group By จะใช้ร่วมกับฟังก์ชันผลสรุป โดยมีรูปแบบการใช้ดังนี้

รูปแบบ SELECT fieldlist

FROM tablename

WHERE criteria

[GROUP BY groupfieldlist]

เมื่อ fieldlist หมายถึงชื่อฟิลด์ที่จะนำมาเลือกข้อมูลหรือคำนวณค่าด้วยฟังก์ชันผลสรุป  
tablename หมายถึงชื่อเทเบิลที่เป็นแหล่งข้อมูล  
criteria หมายถึงเงื่อนไขที่ใช้เลือกข้อมูลจากเทเบิล  
groupfieldlist หมายถึงชื่อฟิลด์ที่ใช้จัดกลุ่มข้อมูล



## 6. กำหนดเงื่อนไขจัดกลุ่มข้อมูลด้วย HAVING

การจัดกลุ่มข้อมูลด้วยคำสั่ง GROUP BY นั้น สามารถกำหนดเงื่อนไขเพิ่มเติมเพื่อแสดงขอบ-เขตข้อมูลที่จะนำมาคำนวณค่าผลสรุปแบบเจาะจงมากขึ้นโดยใช้คำสั่ง HAVING ซึ่งมีรูปแบบการใช้ดังนี้

```
รูปแบบ SELECT fieldlist
        FROM tablename
        WHERE criteria
        [GROUP BY groupfieldlist]
        [HAVING condition]
```

เมื่อ condition หมายถึงนิพจน์เงื่อนไขที่ใช้จัดกลุ่มข้อมูล โดยกำหนดได้ไม่เกิน 40 เงื่อนไข และแต่ละเงื่อนไขต้องเชื่อมด้วย โอเปอเรเตอร์ And หรือ Or

### 2.4.4 ปรับปรุงข้อมูลด้วยคำสั่ง UPDATE

เป็นคำสั่งที่ใช้ปรับปรุงหรือแก้ไขข้อมูลในเทเบิล โดยสามารถทำได้ทีละเรคอร์ด หรือหลายๆ เรคอร์ดหรือทุกเรคอร์ดในเทเบิลพร้อมกัน

```
รูปแบบ UPDATE tablename SET fieldname1 = value1 [, fieldname2 = value2, ...]
```

```
WHERE criteria
```

เมื่อ tablename หมายถึงชื่อเทเบิลที่ต้องการปรับปรุงข้อมูล  
 fieldname1, fieldname2, ... หมายถึงชื่อฟิลด์ที่จะนำข้อมูลมาแก้ไขหรือกำหนดค่าใหม่  
 value1, value2, ... หมายถึงข้อมูลที่จะนำไปแทนที่ข้อมูลเดิมของฟิลด์ ที่ระบุไว้ในอาร์กิวเมนต์ fieldname  
 criteria หมายถึงเงื่อนไขที่ใช้ค้นหาเรคอร์ดที่จะนำมาแก้ไขข้อมูล

### 2.4.5 เพิ่มเรคอร์ดใหม่ด้วยคำสั่ง INSERT

เป็นคำสั่งที่ใช้เพิ่มข้อมูลเรคอร์ดใหม่ในเทเบิล โดยสามารถกำหนดข้อมูลที่จะเพิ่มไว้ในคำสั่ง INSERT INTO โดยตรง หรือจะใช้ร่วมกับคำสั่ง SELECT...FROM... เพื่อดึงข้อมูลจาก เทเบิลอื่นมาใช้ก็ได้

#### 1. กำหนดข้อมูลในคำสั่ง INSERT INTO

เป็นวิธีที่ใช้สำหรับเพิ่มข้อมูลเรคอร์ดใหม่ที่ละเรคอร์ด (single-record) โดยกำหนดข้อมูลแต่ละฟิลด์ลงในคำสั่ง INSERT INTO โดยตรง

```
รูปแบบ INSERT INTO tablename [(fieldname1 [, fieldname2 [, ...] )]
```

```
VALUES (value1 [, value2 [, ...] )]
```

เมื่อ tablename หมายถึงชื่อเทเบิลที่ต้องการเพิ่มข้อมูลเรคอร์ดใหม่  
 fieldname1, fieldname2, ... หมายถึงชื่อฟิลด์สำหรับรับข้อมูลที่จะเพิ่มใหม่

value1, value2, ... หมายถึงข้อมูลที่กำหนดให้แต่ละฟิลด์ โดยลำดับที่ของข้อมูล จะต้องตรงกับลำดับที่ของฟิลด์รับด้วย

## 2. ดึงข้อมูลจากคำสั่งเทเบิลอื่นด้วยคำสั่ง SELECT...FROM...

เป็นวิธีที่ใช้คัดลอกข้อมูล 1 เรคอร์ดหรือหลายๆเรคอร์ด (multiple-record) พร้อมกันจากเทเบิลหนึ่งมาสร้างเป็นเรคอร์ดใหม่ในอีกเทเบิลหนึ่ง ชื่อฟิลด์และชนิดข้อมูลของทั้งสองเทเบิลต้องเหมือนกัน

รูปแบบ INSERT INTO tablename [IN externaldatabase] [(fieldname1 [, fieldname2 [, ...]])]

SELECT [source.] fldname1 [, fldname2 [, ...]] FROM tableexpression

เมื่อ	tablename	หมายถึงชื่อเทเบิลที่ต้องการเพิ่มข้อมูลเรคอร์ดใหม่
	fieldname1, fieldname2, ...	หมายถึงชื่อฟิลด์ที่ใช้รับข้อมูลใหม่
	externaldatabase	หมายถึงชื่อไฟล์ฐานข้อมูลที่เป็นเจ้าของเทเบิลที่กำหนดใน tablename (ต้องระบุชื่อไฟล์ฐานข้อมูลเสมอถ้าเทเบิลต้นทางและปลายทางไม่อยู่ในไฟล์ฐานข้อมูลเดียวกัน)
	source	หมายถึงชื่อเทเบิลต้นทางที่จะนำข้อมูลมาใช้
	fldname1, fldname2, ...	หมายถึงชื่อฟิลด์ในเทเบิลต้นทาง
	tableexpression	หมายถึงชื่อเทเบิลหรือกลุ่มเทเบิลที่มีความสัมพันธ์กันจากการใช้คำสั่งจับคู่ (JOIN)

### 2.4.6 ลบเรคอร์ดด้วยคำสั่ง DELETE

เป็นคำสั่งที่ใช้ลบข้อมูลที่ละเรคอร์ดหรือหลายๆเรคอร์ดพร้อมกันออกจากเทเบิล

รูปแบบ DELETE [\*] FROM tablename WHERE criteria

เมื่อ	tablename	หมายถึงชื่อเทเบิลที่ต้องการลบเรคอร์ดข้อมูลออกไป
	criteria	หมายถึงเงื่อนไขที่ใช้ค้นหาเรคอร์ดที่ต้องการลบ

## 2.5 การเขียนโปรแกรมโดยใช้ Visual Basic.NET ติดต่อฐานและจัดการฐานข้อมูลด้วย

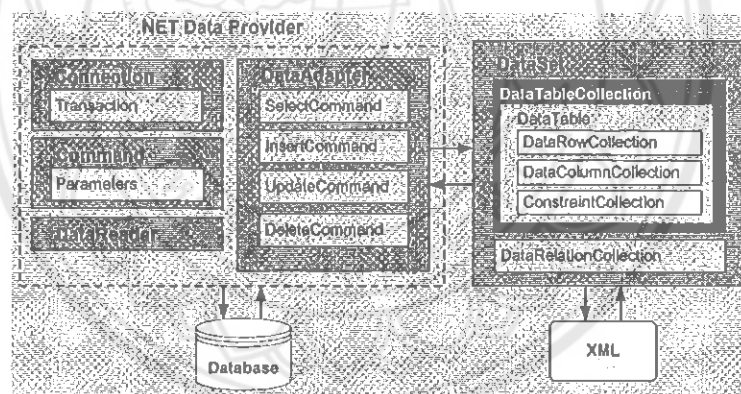
### ADO.NET

#### 2.5.1 โครงสร้างการทำงานใน ADO.NET

ผู้ที่เคยใช้ Visual Basic 6.0 พัฒนาโปรแกรมด้านฐานข้อมูลมาก่อนจะทราบว่า การใช้เทคโนโลยี ADO (ActiveX Data Object) เป็นเครื่องมือในการติดต่อและจัดการฐานข้อมูลในแบบ connected database ก็จะต้องเปิดการเชื่อมต่อกับฐานข้อมูลไว้ตลอดเวลาใช้งานซึ่งจะทำให้การเพิ่ม ลบ แก้ไขหรือการกระทำใดๆ ก็ตามที่เกี่ยวข้องกับข้อมูลจะเป็นการกระทำกับข้อมูลจริงในไฟล์ฐานข้อมูลนั้นโดยตรง ซึ่งมีข้อดีคือทำให้ข้อมูลถูกปรับปรุงให้ทันสมัยและถูกต้องอยู่

ตลอดเวลา แต่ข้อเสียคือ ถ้ามีผู้ใช้ฐานข้อมูลเป็นจำนวนมากและพร้อมๆกันหรือการประมวลผลข้อมูลที่มีปริมาณมากจนทำให้ระยะเวลาที่เชื่อมต่อกับ server นานเกินไป ซึ่งจะทำให้อัตราการสิ้นเปลืองทรัพยากรบนเครื่อง server สูง ส่งผลให้ประสิทธิภาพโดยรวมของระบบลดลงได้ ดังนั้นใน Visual Studio.NET จึงได้นำเทคโนโลยีตัวใหม่ล่าสุดคือ ADO.NET ที่พัฒนาจาก ADO มาจัดการข้อมูลแบบ connected database คือสามารถจัดการข้อมูลในฐานข้อมูลโดยไม่ต้องการเชื่อมต่อไว้ตลอดเวลาเหมือนใน ADO เนื่องจากข้อมูลจะถูกอ่านจากฐานข้อมูลมาเก็บไว้ในหน่วยความจำก่อน จากนั้นจะตัดการเชื่อมต่อกับฐานข้อมูล ทำให้การกระทำใดๆ หลังจากนี้จะเป็นการกระทำกับข้อมูลในหน่วยความจำเท่านั้น เมื่อเสร็จสิ้นกระบวนการแล้วจึงจะบันทึกกลับลงฐานข้อมูล ซึ่งจะช่วยประหยัดทรัพยากรของระบบและเพิ่มประสิทธิภาพการทำงานของ server (ที่จริงใน ADO เองก็มีความสามารถในการทำ connected database ได้ แต่โปรแกรมเมอร์จะต้องเป็นผู้ควบคุมด้วยตนเองว่าจะให้มีการเชื่อมต่อหรือยกเลิกการเชื่อมต่อเมื่อใด ซึ่งต่างกับ ADO.NET ที่จะตัดการเชื่อมต่อให้โดยอัตโนมัติ เมื่อไม่มีความจำเป็นต้องใช้ฐานข้อมูลในแบบ on-line และจะเปิดการเชื่อมต่อให้ใหม่เมื่อจำเป็นต้องทำงานร่วมกับฐานข้อมูลจริง)

### 1. คลาสใน ADO.NET



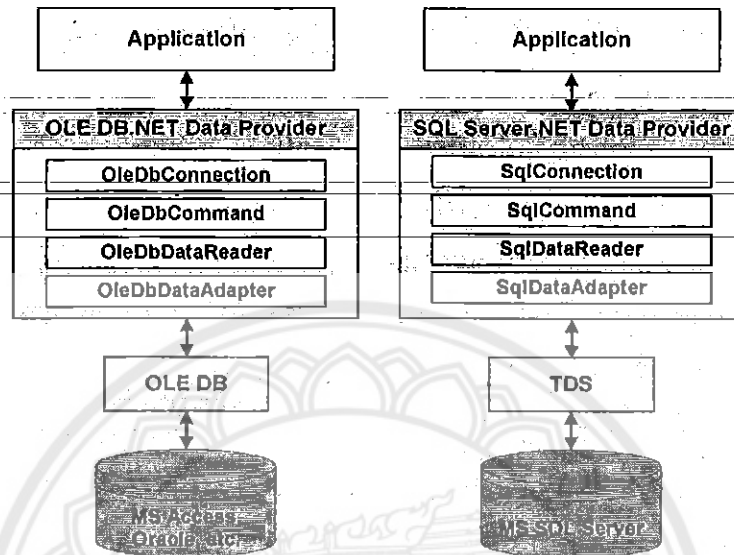
รูปที่ 2.15 กลุ่มคลาสใน ADO.NET ที่ทำงานร่วมกับฐานข้อมูล

เนื่องจากการทำงานร่วมกับฐานข้อมูลด้วย ADO.NET สามารถทำได้ทั้งแบบ connected database และ disconnected database จึงอาจแบ่งคลาสใน ADO.NET ที่ใช้ทำงานร่วมกับฐานข้อมูลออกได้เป็น 2 กลุ่มคือ

กลุ่มที่ 1 เป็นกลุ่มคลาสที่ทำงานร่วมกับฐานข้อมูลแบบเปิดการเชื่อมต่อ (connected database) แยกออกเป็น 2 กลุ่มตามประเภทของ Data Provider

- กลุ่ม SQL Server .NET Data Provider ใช้ติดต่อและทำงานร่วมกับไฟล์ฐานข้อมูล Microsoft SQL Server ประกอบด้วยคลาส SqlCommand, SqlDataReader และ SqlDataAdapter

- กลุ่ม OLE DB .NET Data Provider ใช้ติดต่อและทำงานร่วมกับไฟล์ฐานข้อมูลเช่น Microsoft Access, Oracle เป็นต้น ประกอบด้วยคลาส OleDbCommand, OleDbDataReader และ OleDbDataAdapter



รูปที่ 2.16 กลุ่มคลาสที่ทำงานร่วมกับฐานข้อมูลแบบเปิดการเชื่อมต่อ (connected database) แยกตามประเภทของ Data Provider

คลาสทั้งสองกลุ่มนี้ทำหน้าที่เดียวกัน แต่ที่แยกจากกันก็เพื่อให้เกิดประสิทธิภาพในการทำงานสูงสุด เนื่องจากกลุ่มคลาส Sql นั้นสามารถสื่อสารกับฐานข้อมูล SQL Server ได้โดยตรงอยู่แล้วในระดับ low-level ผ่านโปรโตคอล Tabular Data Stream (TDS) ในขณะที่กลุ่มคลาส OLE DB Providers ถ้านำคลาสในกลุ่มนี้มาใช้ร่วมกับฐานข้อมูล Microsoft SQL Server เวอร์ชัน 7.0 หรือหลังจากนี้จะทำให้ความเร็วและประสิทธิภาพในการเข้าถึงข้อมูลลดลงได้

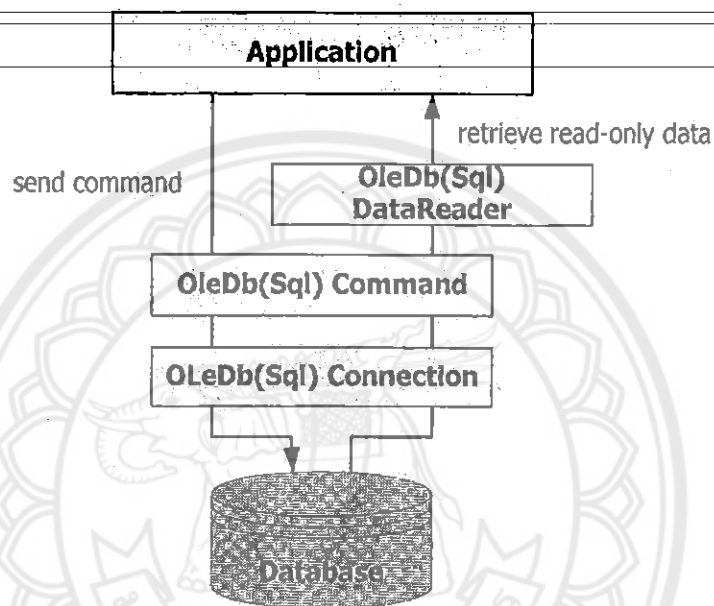
กลุ่มที่ 2 เป็นกลุ่มคลาส DataSet ที่ทำงานร่วมกับข้อมูลแบบตัดการเชื่อมต่อกับฐานข้อมูล disconnected database

การทำงานใน layer แรกซึ่งเป็นส่วนที่ทำงานแบบเชื่อมต่อกับฐานข้อมูลตลอดเวลา นั้นจะเหมาะกับการที่ทำงานที่ปริมาณข้อมูลไม่มากนักและประมวลผลข้อมูลไม่ซับซ้อนจนทำให้สิ้นเปลืองหน่วยความจำและทรัพยากรอื่นๆ บนเครื่อง server มากเกินไป คลาสแต่ละตัวจะทำหน้าที่ดังนี้

- คลาส connection ทำหน้าที่ติดต่อกับฐานข้อมูล SQL Server หรือ OLE DB เช่นเดียวกับออบเจ็กต์ Connection ใน ADO

- คลาส command จะใช้ร่วมกับออบเจ็กต์ Parameter ในการประมวลผลคำสั่ง SQL, Query, หรือ Stored Procedure ในแบบที่สามารถส่งผลลัพธ์คืนกลับมาหรือไม่ส่งคืนก็ได้เช่นเดียวกับการใช้ออบเจ็กต์ Command ใน ADO

- คลาส Data Reader จะใช้สำหรับการดึงข้อมูลแบบที่ละเอียดกว่าจากฐานข้อมูลบน Server cursor จะเป็นแบบ forward-only คือเดินหน้าอย่างเดียวถอยหลังไม่ได้และไม่สามารถแก้ไขข้อมูลได้ด้วย (read-only) ทำให้ความเร็วในการเข้าถึงข้อมูลสูง



รูปที่ 2.17 การทำงานใน layer แบบ connected database

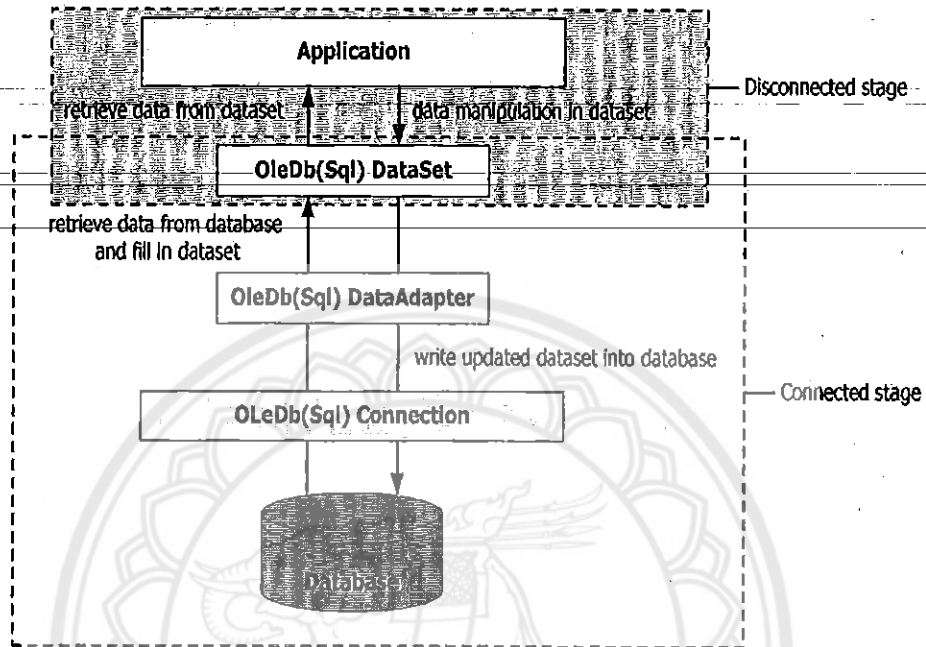
การทำงานใน layer ถัดมาคือส่วนที่ทำงานในแบบ disconnected database จะใช้ในกรณีที่มีปริมาณข้อมูลมีจำนวนมากและการเข้าถึงข้อมูลมีความซับซ้อนขึ้นเช่น การดึงข้อมูลแบบเป็นกลุ่ม, การเรียงลำดับ หรือการกรองข้อมูล เป็นต้น คลาสที่เกี่ยวข้องกับการทำงานใน layer นี้ก็คือคลาส

DataSet และ DataAdapter

- คลาส DataAdapter ประกอบด้วยออบเจ็กต์ command 4 ตัว คือ SelectCommand, InsertCommand, UpdateCommand และ DeleteCommand โดยออบเจ็กต์ SelectCommand จะทำหน้าที่คัดเลือกข้อมูลจากฐานข้อมูลมาสร้างชุดข้อมูลหรือ dataset ซึ่งจะเก็บไว้ในหน่วยความจำของเครื่องไคลเอนต์ หลังจากนั้นจะตัดการเชื่อมต่อกับฐานข้อมูลจริงบนเครื่องเซิร์ฟเวอร์

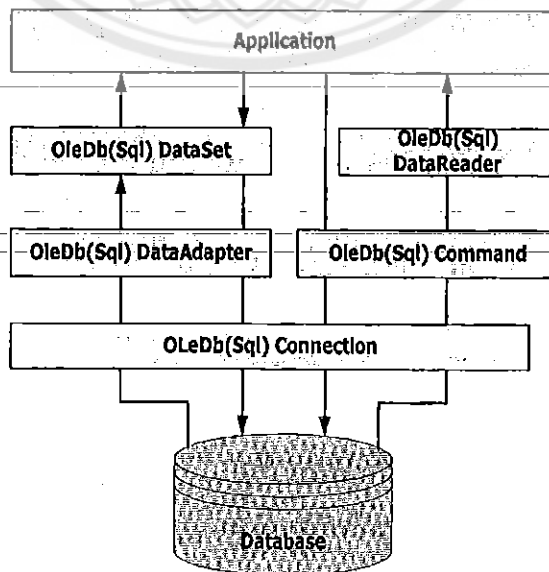
- คลาส DataSet ใช้แทนชุดข้อมูลในหน่วยความจำ การกระทำใดๆหลังตัดการเชื่อมต่อกับฐานข้อมูลเช่น เพิ่มเรคอร์ด ลบเรคอร์ด แก้ไขข้อมูล จะเป็นการกระทำกับข้อมูลใน dataset เท่านั้น ไม่ได้กระทำกับข้อมูลในฐานข้อมูลจริง ซึ่งต่างกับออบเจ็กต์ Recordset ใน ADO ที่การกระทำใดๆ กับข้อมูลในกลุ่มเรคอร์ด (recordset) จะมีผลต่อข้อมูลในฐานข้อมูลด้วย หลัง

ปรับปรุงแก้ไขข้อมูลใน dataset แล้ว จะใช้ออบเจ็กต์ InsertCommand, UpdateCommand และ DeleteCommand ของคลาส DataAdapter บันทึกข้อมูลในหน่วยความจำกลับลงฐานข้อมูลอีกครั้งหนึ่ง



รูปที่ 2.18 การทำงานใน layer แบบ disconnected database

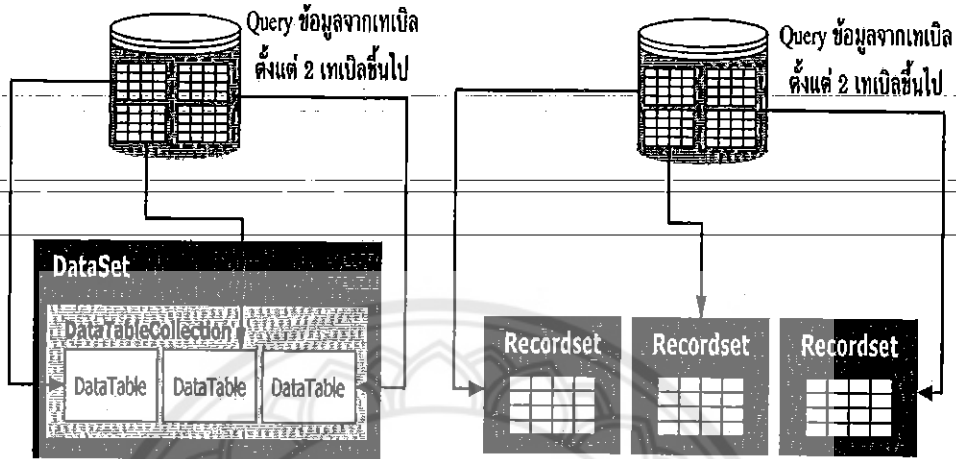
ถ้านำคลาสใน layer ทั้งสองมาประกอบกันเพื่อมองภาพรวมการทำงานทั้งระบบ จะแสดงได้ด้วยแผนภาพดังนี้



รูปที่ 2.19 ภาพรวมการทำงานทั้งระบบของคลาสใน layer ของกลุ่มคลาส DataSet

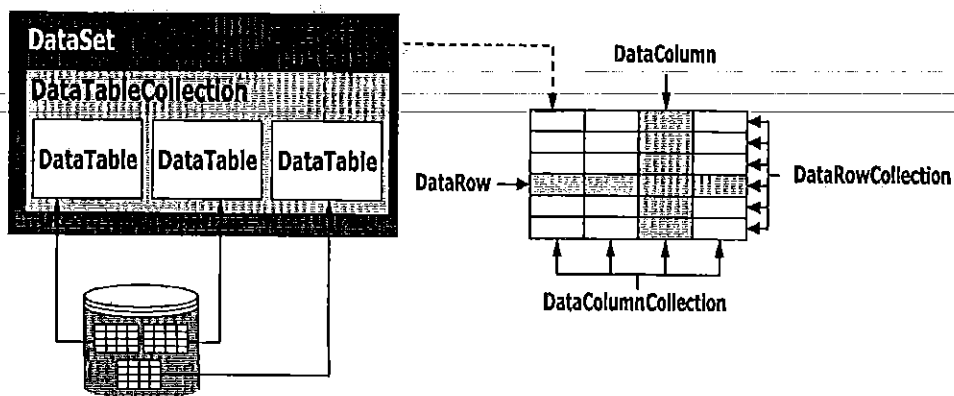
## 2. เปรียบเทียบ DataSet กับ RecordSet ใน ADO

สำหรับคลาส DataSet นี้ถูกนำมาใช้แทนออบเจ็กต์ Recordset ใน ADO โดยที่มีคุณสมบัติที่แตกต่างจาก Recordset ซึ่งจะนำมากล่าวบางส่วนที่สำคัญคือ



รูปที่ 2.20 เปรียบเทียบ DataSet กับ RecordSet ใน ADO

- เนื่องจากใน 1 DataSet จะประกอบด้วย DataTable ตั้งแต่ 1 ตั้งแต่หนึ่งตัวขึ้นไป โดยกลุ่ม DataTable ทั้งหมดจะเก็บอยู่ใน DataTableCollection จึงสามารถแสดงผลลัพธ์ (resultsets) ได้มากกว่า 1 ผลลัพธ์ใน DataSet เดียวกัน ถ้าเปรียบเทียบกับ ADO จะเห็นว่าออบเจ็กต์ Recordset 1 ตัว จะแทนข้อมูลจาก 1 เทเบิล (หรือ Query) เท่านั้น ถ้าต้องการนำข้อมูลจากหลายๆ เทเบิลมาสร้าง Recordset จะต้องนำเทเบิลเหล่านั้นมาจับคู่เพื่อสร้างข้อมูลจากความสัมพันธ์ระหว่างกันก่อน ถ้านำข้อมูลจากเทเบิลที่ไม่มีความสัมพันธ์กันเลยมาใช้ จะต้องสร้างออบเจ็กต์ Recordset เท่ากับจำนวนเทเบิลนั้น จากที่กล่าวมาว่าใน 1 DataSet อาจประกอบด้วย DataTable มากกว่า 1 ตัว จึงสามารถสร้างความสัมพันธ์ระหว่าง DataTable เหล่านี้ได้เช่นกันด้วยคลาส DataRelation



รูปที่ 2.21 คุณสมบัติของ DataSet ใน ADO

- DataTable แต่ละตัวใน DataSet ประกอบด้วยกลุ่มของ DataRow (DataRowCollection) และแต่ละกลุ่ม DataColumn (DataColumnCollection) โดย 1 DataRow จะหมายถึงข้อมูล 1 เรคอร์ด ส่วน 1 DataColumn จะหมายถึงข้อมูล 1 ฟิลด์ของ DataTable นั้น นอกจากนี้ในแต่ละ DataTable ยังเห็นข้อมูลโครงสร้างเช่น ชื่อคอลัมน์ (ColumnName), ชนิดของข้อมูล (Data Type), คุณสมบัติการเพิ่มค่าแบบอัตโนมัติ (AutoIncrement), อนุญาตให้มีการรับค่าว่าง (AllowDBNull) เป็นต้น รวมทั้งส่วนของ ConstraintCollection ที่เกี่ยวกับกฎเกณฑ์และเงื่อนไขต่างๆ ที่ใช้ควบคุมการคงสภาพความถูกต้องของข้อมูลเช่น การกำหนดว่าเมื่อลบเรคอร์ดแม่ (parent record) ให้เรคอร์ดลูกทั้งหมด (child record) ต้องถูกลบออกไป หรือกำหนดให้ข้อมูลในฟิลด์นั้นจะต้องไม่มีค่าซ้ำกัน (UniqueConstraint) เป็นต้น

- การทำงานร่วมกับ DataSet จะเป็นอิสระจากการเชื่อมต่อกับฐานข้อมูลบนเครื่อง server โดยจะทำหน้าที่เป็น memory-resident database คือเก็บข้อมูลจากเทเบิลไว้ในหน่วยความจำของเครื่อง client โดยจำลองโครงสร้างและรูปแบบการเก็บข้อมูลจริงในฐานข้อมูลบน database server

- เนื่องจากข้อมูลใน DataSet จะถูกเก็บบนหน่วยความจำของเครื่อง client จึงไม่มีการนำ server cursor มาใช้เหมือนกับการใช้ออบเจ็กต์ Recorder ใน ADO

- การปรับปรุงไขข้อมูลจะทำให้เฉพาะในหน่วยความจำซึ่งไม่มีผลต่อข้อมูลในฐานข้อมูลจริง ถ้าต้องการบันทึกข้อมูลที่แก้ไขหรือเพิ่มใหม่ลงฐานข้อมูลจริงจะต้องทำผ่านเมธอด update() ของคลาส DataAdapter เพื่อประมวลผลคำสั่ง InsertCommand, UpdateCommand และ DeleteCommand ดังกล่าวแล้ว ซึ่งจะต่างกับการใช้ออบเจ็กต์ Recordset ใน ADO ที่การแก้ไขและบันทึกข้อมูลจะสามารถทำกับฐานข้อมูลจริงได้โดยตรง ทำให้การทำงานใน layer ที่ 2 ของ ADO.NET ด้วย DataSet จะใช้หน่วยความจำบนเครื่องสูงและความเร็วในการเข้าถึงข้อมูลจะต่ำกว่าการใช้ DataReader ใน layer แรก

### 2.5.2 เนมสเปซที่ใช้งานกับฐานข้อมูล

เนื่องจากคลาสต่างๆที่ใช้งานใน .NET-Framework มีอยู่เป็นจำนวนมาก จึงต้องจำแนกคลาสไว้เป็นกลุ่มเพื่อเรียกใช้ได้ง่าย ซึ่งจะเรียกกลุ่มของคลาสต่างๆนี้ว่า เนมสเปซ (namespace) ซึ่งในที่นี้จะขอกกล่าวถึงเนมสเปซบางตัวที่มีความสำคัญและจำเป็นต่อการทำงานร่วมกับ ADO.NET

- System.Data เป็นเนมสเปซ ที่ประกอบด้วย คลาสพื้นฐานที่ใช้ร่วมกันระหว่างแอปพลิเคชันต่างๆ บนแพลตฟอร์ม .NET โดยไม่ขึ้นกับชนิดของแหล่งข้อมูล (data source) หรือภาษาคอมพิวเตอร์ที่ใช้พัฒนาแอปพลิเคชันนั้น เช่น คลาส DataSet, DataTable, DataRow, DataRelation เป็นต้น

- System.Data.SqlClient ประกอบด้วยคลาสที่ใช้ในการติดต่อและจัดการข้อมูลฐานข้อมูล Microsoft SQL Sever เช่น คลาส SqlConnection, SqlCommand, SqlDataReader, SqlDataAdapter



- System.Data.OleDb ประกอบด้วยคลาสที่ใช้ติดต่อและจัดการข้อมูลในฐานข้อมูลที่เชื่อมผ่าน OLE DB Provider เช่น คลาส OleDbConnection, OleDbDataReader, OleDbDataAdapter

### 1. การเรียกใช้คลาสใน ADO.NET

การนำคลาสใน ADO.NET มาใช้จะทำให้ได้ 2 วิธีคือ

- เรียกใช้ผ่านคอนโทรลทูลบ็อกซ์ Data ของ VB.NET IDE โดยเขียนโปรแกรมควบคุมการทำงานเพิ่มเติมเล็กน้อยถ้าเปรียบเทียบกับเครื่องมือที่ทำงานร่วมกับ ADO ใน VB 6.0 ก็อาจเปรียบเทียบได้กับคอนโทรล ADO Data และ DataEnvironment ที่ช่วยให้สามารถเชื่อมต่อและจัดการข้อมูลได้อย่างรวดเร็ว

- เรียกใช้ผ่านโปรแกรมที่เขียนขึ้นเองทั้งหมดตั้งแต่ขั้นตอนการเชื่อมต่อกับฐานข้อมูลด้วยคลาส Connection, การอ่านข้อมูลจากฐานข้อมูลมาไว้ในหน่วยความจำด้วยคลาส DataAdapter เพื่อทำงานบนแบบ memory-resident database ด้วยคลาส DataSet จนถึงขั้นตอนการบันทึกข้อมูลทั้งหมดกลับลงฐานข้อมูลอีกครั้ง

#### 2.5.3 ติดต่อกับฐานข้อมูลด้วยคลาส Connection

การเข้าถึงและการจัดการข้อมูลในฐานข้อมูล จะต้องสร้างการเชื่อมต่อกับฐานข้อมูลนั้นก่อน สำหรับภาษา VB.NET จะใช้คำสั่ง Imports อ้างอิงเนมสเปซไว้ที่บรรทัดบนสุดของโปรแกรมเพื่อเรียกใช้คลาสที่เหมาะสมกับประเภทของฐานข้อมูล ถ้าใช้ภาษาอื่นจะแตกต่างกันไป เช่น C# ใช้ Using เป็นต้น

##### 1. สร้างออบเจกต์จากคลาส Connection

รูปแบบที่แสดงข้างล่างเป็นรูปแบบคำสั่งที่ใช้ติดต่อกับฐานข้อมูล Microsoft Access ด้วย OleDbConnection ถ้าต้องการติดต่อกับฐานข้อมูล Microsoft SQL Server ให้ใช้รูปแบบเดียวกันโดยเปลี่ยนจาก OleDbConnection เป็น SqlConnection

รูปแบบที่ 1      Dim objConn As OleDbConnecton

objConn = New OleDbConnection(strConn)

รูปแบบที่ 2      Dim objConn As OleDbConnecton = New OleDbConnection(strConn)

เมื่อ objConn หมายถึงออบเจกต์ที่สร้างจากคลาส Connection.

strConn หมายถึง connection string หรือประโยคข้อความที่ประกอบด้วย อาร์กิวเมนต์ต่างๆที่ใช้ติดต่อกับฐานข้อมูล โดยอาร์กิวเมนต์แต่ละตัวจะคั่นด้วยเครื่องหมาย ; ในที่นี้จะขอกล่าวถึงอาร์กิวเมนต์ที่สำคัญและควรทราบดังนี้

ตารางที่ 2.7 ค่าอาร์กิวเมนต์ต่างๆที่ใช้ติดต่อกับฐานข้อมูล

อาร์กิวเมนต์	ค่าที่กำหนด
Provider	ระบุชื่อผู้ให้บริการเชื่อมต่อกับฐานข้อมูล
DataSource	ระบุชื่อฐานข้อมูลที่ต้องการติดต่อกับ (ถ้าเป็นฐานข้อมูล SQL Server ให้ระบุเป็นชื่อของ Server หรือถ้าฐานข้อมูลนั้นอยู่บนเครื่อง local ให้กำหนดด้วยชื่อเครื่องนั้นหรือถ้าติดตั้ง MSDE จาก Microsoft .NET Framework SDK ที่ให้มาพร้อมกับชุดโปรแกรม Microsoft Visual Studio.NET ไว้บนเครื่อง จะใส่ (local)\NetSDK แทนชื่อเครื่องก็ได้
UserID	ระบุชื่อผู้ใช้บริการที่ต้องการติดต่อกับฐานข้อมูล
Password	รหัสผ่านของผู้ใช้บริการ
Initial Catalog	ระบุชื่อฐานข้อมูล (ใช้ในกรณีเชื่อมต่อกับฐานข้อมูล SQL Server) กรณีเชื่อมต่อกับฐานข้อมูล SQL Server ให้ระบุว่าจะใช้ NT Security (Trusted Connection) หรือ SQL Server Security (Standard Security) ในการตรวจสอบ (authentication) การเข้าใช้ฐานข้อมูล (ค่าเริ่มต้นหรือค่าดีฟอลต์ของอาร์กิวเมนต์นี้คือ SQL Server Security ซึ่งมีค่าเป็น 'False' <ul style="list-style-type: none"> <li>- ถ้าใช้ NT Security ให้กำหนดค่าเป็น 'True' หรือ 'SSPI' (Security Support Provider Interface) มีความหมายว่าให้ใช้ข้อมูลที่ log on เข้าสู่ Windows NT/2000/XP ในการเชื่อมต่อกับฐานข้อมูล จึงไม่ต้องกำหนด User ID และ Password อีก (แต่กรณีจะทำได้ก็ต่อเมื่อ account นั้นได้รับอนุญาตให้เข้าถึงฐานข้อมูลนั้นด้วย)</li> </ul> ถ้าใช้ SQL Server Security ให้กำหนดข้อมูลที่ใส่ log on เข้าสู่ฐานข้อมูลนั้นไว้ในอาร์กิวเมนต์ User ID และ Password

## 2. เปิดการเชื่อมต่อกับฐานข้อมูลด้วยเมธอด Open()

หลังจากสร้าง connection string หรือประโยคข้อความที่ใช้เชื่อมต่อกับฐานข้อมูลเสร็จแล้ว จะต้องทำการ active หรือเปิดการเชื่อมต่อกับฐานข้อมูลนั้นเมธอด Open() ซึ่งมีรูปแบบดังนี้

รูปแบบ ObjConn.Open()

เมื่อ ObjConn หมายถึงออบเจ็กต์ที่สร้างจากคลาส Connection

### 3. ปิดการเชื่อมต่อกับฐานข้อมูลด้วยเมธอด Close()

หลังเสร็จสิ้นการทำงานร่วมกับฐานข้อมูล ให้ปิดการเชื่อมต่อกับฐานข้อมูลนั้นเพื่อคืน

หน่วยความจำและทรัพยากรที่ถูกเรียกใช้ให้กับระบบโดยใช้เมธอด Close()

รูปแบบ ObjConn.Close()

เมื่อ ObjConn หมายถึงออบเจ็กต์ที่สร้างจากคลาส Connection

#### 2.5.4 แสดงข้อมูลด้วยคลาส DataReader

หลังเชื่อมต่อกับฐานข้อมูลแล้ว ถ้าต้องการนำข้อมูลมาแสดงให้ใช้คลาส Command ร่วมกับเมธอด ExecuteReader() สร้างและประมวลผลคำสั่ง Select เพื่อคัดเลือกข้อมูลที่ต้องการและส่งผลลัพธ์ที่ได้มายัง DataReader หลังจากนั้นจึงใช้เมธอด Read() ดึงข้อมูลจาก DataReader มาแสดงผลทีละเรคอร์ดโดยใช้ลูป While...End , While, Do While...Loop หรือ For...Next เป็นต้น

##### 1. สร้างออบเจ็กต์จากคลาส DataReader

รูปแบบ Dim objdtReader As OleDbDataReader

เมื่อ objdtReader หมายถึงออบเจ็กต์ที่สร้างจากคลาส DataReader

##### 2. ประมวลผลด้วยคำสั่ง ExecuteReader()

เป็นเมธอดที่ใช้ประมวลผลคำสั่ง SQL ร่วมกับออบเจ็กต์ Command โดยนำผลลัพธ์ที่ได้มาเก็บไว้ใน DataReader (ดูรายละเอียดการสร้างออบเจ็กต์ Command และประโยคคำสั่ง SQL ได้ในหัวข้อประมวลผลคำสั่ง SQL ด้วยคลาส Command)

รูปแบบ ObjdtReader = objCmd.ExecuteReader()

เมื่อ objdtReader หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataReader

objCmd หมายถึง ออบเจ็กต์ที่สร้างจากคลาส Command

### 3. เข้าถึงข้อมูลแต่ละเรคอร์ดด้วยเมธอด Read()

เป็นเมธอดที่ใช้ดึงข้อมูลจาก DataReader มาแสดงผลแบบทีละเรคอร์ด

รูปแบบ objdtReader.Read()

เมื่อ objdtReader หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataReader

### 4. เข้าถึงข้อมูลแต่ละฟิลด์ด้วยพรีอเพอริตี้ Item()

รูปแบบที่ 1 objdtReader.Item(field)

รูปแบบที่ 2 objdtReader(field)

เมื่อ objdtReader หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataReader

field หมายถึง ชื่อฟิลด์หรือลำดับที่ฟิลด์นั้นในเรคอร์ด (ฟิลด์แรกมีลำดับที่เป็น 0 เรียงตามลำดับไปจนถึงลำดับที่ n)

ถ้าจะสร้างออบเจ็กต์จากคลาส SqlDataReader ให้ใช้รูปแบบข้างต้นเช่นเดียวกันแต่เปลี่ยนจาก OleDbDataReader เป็น SqlDataReader

### 2.5.5 ประมวลผลคำสั่ง SQL ด้วยคลาส Command

สามารถใช้คำสั่ง SQL ในกลุ่ม DML จัดการข้อมูลในฐานข้อมูลได้ ซึ่งประกอบด้วยคำสั่ง Insert (เพิ่มเรคอร์ดใหม่), Update (ปรับปรุงเรคอร์ด) และ Delete (ลบเรคอร์ด)

#### 1. สร้างออบเจ็กต์และประโยคคำสั่ง SQL จากคลาส Command

รูปแบบที่ 1 Dim objCmd As OleDbCommand

```
objCmd = New OleDbCommand (sqlCmd, varConn)
```

รูปแบบที่ 2 Dim objCmd As OleDbCommand = New OleDbCommand (sqlCmd, varConn)

เมื่อ objCmd หมายถึง ออบเจ็กต์ที่สร้างจากคลาส Command  
sqlCmd หมายถึง ประโยคคำสั่ง SQL  
objConn หมายถึง ออบเจ็กต์ที่สร้างจากคลาส Connection ที่จะนำไปใช้  
สร้างการติดต่อกับฐานข้อมูล

หลังสร้างออบเจ็กต์ Command และประโยคคำสั่ง SQL แล้ว ให้เลือกเมธอดที่จะนำมาใช้ประมวลผลคำสั่งเพื่อให้ได้ผลลัพธ์ที่ต้องการ ในที่นี้จะแสดงรูปแบบการใช้เมธอด ExecuteNonQuery() และเมธอด ExecuteScalar()

#### 2. ประมวลผลคำสั่งด้วยเมธอด ExecuteNonQuery()

เป็นเมธอดที่ใช้ประมวลผลคำสั่งเช่น เพิ่มเรคอร์ดใหม่ ลบเรคอร์ด แก้ไขเรคอร์ด โดยไม่ส่งผลลัพธ์เป็นเรคอร์ดข้อมูลกลับคืนมา (Non-Query)

กรณีไม่ส่งผลลัพธ์กลับมา

รูปแบบ objCmd.ExecuteNonQuery()

เมื่อ objCmd หมายถึง ออบเจ็กต์ที่สร้างจากคลาส Command

กรณีส่งผลลัพธ์เป็นจำนวนเรคอร์ดที่ถูกประมวลผลกลับคืนมา

รูปแบบ varRecAffected = objCmd.ExecuteNonQuery()

เมื่อ varRecAffected หมายถึง ตัวแปรเก็บจำนวนเรคอร์ดที่ถูกประมวลผล

objCmd หมายถึง ออบเจ็กต์ที่สร้างจากคลาส Command

#### 3. ประมวลผลคำสั่งด้วยเมธอด ExecuteScalar()

เป็นเมธอดที่ส่งผลลัพธ์ที่ได้จากการประมวลผลคำสั่ง SQL กลับคืนมาเพียงค่าเดียวเช่นผลลัพธ์ที่ได้จากการประมวลผลฟังก์ชันผลสรุป (Aggregate Function)

รูปแบบ varResult = objCmd.ExecuteScalar()

เมื่อ varResult หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากการประมวลผลกลุ่มเรคอร์ด

objCmd หมายถึง ออบเจ็กต์ที่สร้างจากคลาส Command  
 การสร้างออบเจ็กต์จากคลาส SqlDataReader ให้ใช้รูปแบบข้างต้นเช่นเดียวกันแต่เปลี่ยนจาก  
 OleDbDataReader เป็น SqlDataReader

### 2.5.6 ทำงานแบบ disconnected database ด้วยคลาส DataSet และ DataAdapter

ในหัวข้อนี้จะแสดงวิธีการทำงานในแบบ disconnected database โดยใช้ dataAdapter ส่งคำสั่ง  
 Select ไปยังฐานข้อมูลเพื่อประมวลผล แล้วนำผลลัพธ์ที่ได้ทั้งหมดมาสร้างเทเบิล dataset ด้วยเม-  
 ธอด Fill() โดยส่วนของ dataset นี้จะถูกเก็บไว้ในหน่วยความจำ (memory-resident database) เพื่อ  
 นำไปใช้งานต่อไปโดยตัดการเชื่อมต่อกับฐานข้อมูลจริง

#### 1. สร้างออบเจ็กต์จากคลาส DataAdapter

รูปแบบ Dim objDataAdapter As OleDbDataAdapter

objDataAdapter = New OleDbDataAdapter(sqlCmd, objConn)

เมื่อ objDataAdapter หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataAdapter  
 objConn หมายถึง ออบเจ็กต์ที่สร้างจากคลาส Connection  
 sqlCmd หมายถึง ประโยคคำสั่งที่นำมาสร้างเทเบิล dataset

#### 2. สร้างออบเจ็กต์จากคลาส DataSet

รูปแบบ Dim objDataSet As DataSet

เมื่อ objDataSet หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataSet

#### 3. สร้างออบเจ็กต์จากคลาส DataTable

รูปแบบ Dim objDataTable As DataTable

objDataTable = objectSet.Tables(dataname)

เมื่อ objDataTable หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataTable  
 objDataSet หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataSet  
 dataname หมายถึง ชื่อเทเบิลใน dataset นั้น

#### 4. สร้างออบเจ็กต์จากคลาส DataRow

เป็นออบเจ็กต์ที่แทนข้อมูลแต่ละแถวใน DataTable

รูปแบบ

รูปแบบที่ 1 Dim objDataRow As DataRow

objDataRow = objDataSet.Tables(dataname).Rows(i)

รูปแบบที่ 2 Dim objDataRow As DataRow

objDataTable.Rows(i)

เมื่อ objDataRow หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataRow  
 objDataTable หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataTable

objdtSet หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataSet

### 5. เข้าถึงข้อมูลแต่ละเรคอร์ดด้วยหรือพเพอร์รี่ Rows()

เป็นหรือพเพอร์รี่ที่ใช้ในการเข้าถึงข้อมูลแต่ละเรคอร์ดใน DataTable

รูปแบบ 1 objdtTable.Rows(i)

รูปแบบ 2 objdtSet.Tables(dataname).rowx(i)

เมื่อ objdtSet หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataTable

objdtTable หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataTable

i หมายถึง ลำดับที่ของเรคอร์ดในชุดข้อมูล โดยเรคอร์ดแรกจะเริ่มที่ลำดับ 0 เรียงตามลำดับ ไปจนถึงลำดับที่ n

### 6. เข้าถึงข้อมูลแต่ละฟิลด์ด้วยหรือพเพอร์รี่ Item()

เป็นหรือพเพอร์รี่ที่ใช้ในการเข้าถึงข้อมูลแต่ละฟิลด์ของแต่ละเรคอร์ดใน DataTable

รูปแบบ 1 objdtTable.Rows(i).Item(fiels)

รูปแบบ 2 objdtSet.Tables(dataname).Rows(i).Item(field)

รูปแบบ 3 objdtRow.Item(field)

เมื่อ objdtTable หมายถึงออบเจ็กต์ที่สร้างจากคลาส DataTable

objdtRow หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataRow

objdtSet หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataSet

field หมายถึง ชื่อฟิลด์หรือลำดับที่ของฟิลด์นั้นในเรคอร์ด (ฟิลด์แรกจะมีลำดับที่เป็น 0 เรียงตามลำดับ ไปจนถึงลำดับที่ n)

### 2.5.7 กรองและเรียงลำดับข้อมูลใน DataTable ด้วยคลาส DataView

สามารถใช้หรือพเพอร์รี่ DefaultView ของคลาส DataTable ไปแสดงผลด้วยคลาส DataView ซึ่งจะแสดงข้อมูลได้เพียงอย่างเดียวไม่ได้ (read-only) โดยมีหรือพเพอร์รี่ RowFilter และหรือพเพอร์รี่ Sort ที่ช่วยให้สามารถกรองข้อมูลตามเงื่อนไขที่กำหนดและจัดเรียงลำดับข้อมูลจากน้อยไปมากหรือมากไปน้อยพร้อมก็นำผลลัพธ์มาแสดงได้ทันที

#### 1. สร้างออบเจ็กต์จากคลาส DataView

รูปแบบ Dim objdtView As DataView

เมื่อ objdtView หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataView

#### 2. แสดงผลข้อมูลด้วยหรือพเพอร์รี่ DefaultView

เป็นหรือพเพอร์รี่ของ DataTable ที่ใช้ในกรณีต้องการนำข้อมูลมาแสดงผลด้วยคลาส

DataView

รูปแบบ objdtView = objdtSet.Tables(datname).DefaultView

เมื่อ objdtView หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataView

objdtSet	หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataSet
dataname	หมายถึงชื่อเทเบิลใน dataset ที่จะนำมาแสดงผลด้วย DataView

### 3. กรองข้อมูลด้วยพร็อพเพอร์ตี้ RowFilter

เป็นเรื่อพเพอร์ตี้ในคลาส DataView ที่นำไปใช้เพื่อกรองข้อมูลภายใต้เงื่อนไขที่กำหนด

รูปแบบ ObjdtView.RowFilter = criteria

เมื่อ	objdtView	หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataView
	criteria	หมายถึง เงื่อนไขที่ใช้ในการกรองข้อมูล

### 4. เรียงลำดับข้อมูลด้วยพร็อพเพอร์ตี้ Sort

รูปแบบ ObjdtView.Sort = fieldame

เมื่อ	objdtView	หมายถึงออบเจ็กต์ที่สร้างจากคลาส DataView
	fieldname	หมายถึง ชื่อฟิลด์ที่จะนำข้อมูลมาเรียงลำดับ ถ้าต้องการเรียงลำดับจากมากไปน้อยให้ใส่ DESC ต่อท้ายชื่อฟิลด์ โดยเว้นวรรคอย่างน้อย 1 ช่อง

## 2.5.8 บันทึกข้อมูลใน DataSet ลงฐานข้อมูล

เนื่องจากการทำงานกับ DataSet จะเป็นแบบ disconnecter database การเปลี่ยนแปลงใดๆ ที่เกิดขึ้นกับข้อมูลใน DataSet นั้นจะยังไม่ถูกบันทึกลงฐานข้อมูลทันที จะต้องใช้คลาส CommandBuilder สร้าง InseertCommand, UpdateCommand และ DeleteCommand ก่อน(เนื่องจากการทำงานใดๆ กับ Dataset จะเป็นได้ทั้งการเพิ่มเรคอร์ดใหม่ ปรับปรุงข้อมูลในเรคอร์ดเก่า และลบเรคอร์ด) หลังจากนั้นจึงใช้เมธอด Update() ของ DataAdapter บันทึกข้อมูลใน DataSet ที่มีการเปลี่ยนแปลงกลับลงฐานข้อมูลจริงอีกครั้งหนึ่งด้วย Dommand ทั้งสามตัวข้างต้น

### 1. สร้างออบเจ็กต์จากคลาส DammandBuilder

รูปแบบ Dim objCmdBuidler As OleCommndBuilder

ObjCmdBuidler = New OleDbCommandBuilder(objdtAdapter)

เมื่อ	objCmdBuidler	หมายถึง ออบเจ็กต์ที่สร้างจากคลาส CommandBuilder
	objdtAdapter	หมายถึง ออบเจ็กต์ที่สร้างจากคลาส DataAdapter

### 2. บันทึกข้อมูลลงฐานข้อมูลด้วยเมธอด Update()

รูปแบบ objdtAdapter.Update(objdtSet, dataname)

เมื่อ	objdtAdapter	หมายถึงออบเจ็กต์ที่สร้างจากคลาส DataAdapter
	objdtSet	หมายถึงออบเจ็กต์ที่สร้างจากคลาส DataSet
	dataname	หมายถึงชื่อที่ตั้งให้กับเทเบิลใน Dataset ที่จะนำมาบันทึกลงฐานข้อมูล

## บทที่ 3

### วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงลำดับขั้นตอน และระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย (A Document Database System for English Learning and English-Thai Translation) ประกอบไปด้วย การรวบรวม และวิเคราะห์ความต้องการ, การออกแบบฐานข้อมูล, การออกแบบและพัฒนาโปรแกรมโดยใช้ Microsoft Visual Studio 2005 โปรแกรมด้วยภาษา Visual Basic .NET ติดต่อฐานข้อมูลโดยใช้ Microsoft Office Access 2003

#### 3.1 การรวบรวมและวิเคราะห์ความต้องการ

ระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย (A Document Database System for English Learning and English-Thai Translation) ต้องการระบบคอมพิวเตอร์ที่มีความสามารถในการจัดการข้อมูลคำศัพท์ภาษาอังกฤษและข้อมูลเอกสารตัวอย่างประโยค โดยระบบที่ต้องการ มีรายละเอียดความต้องการดังนี้

- ในการค้นหาคำศัพท์ภาษาอังกฤษ จะค้นหาได้ครั้งละ 1 คำเท่านั้น จะไม่สามารถค้นหา กลุ่มคำหรือวลี (Phrase) ได้
- ในการค้นหาคำศัพท์ภาษาอังกฤษแต่ละครั้ง ต้องสามารถแสดงความหมายของคำศัพท์, คำอ่าน (Pronunciation), คำที่มีความหมายคล้ายกัน (Synonym), คำที่มีความหมายตรงข้าม (Antonym), ตัวอย่างการใช้ (Example Sentence), พจนานุกรมส่วนบุคคล, ตัวอย่างประโยค และเอกสารตัวอย่างประโยคทั้งหมด ออกมาพร้อมกัน
- ในการแสดงคำศัพท์ในพจนานุกรมส่วนบุคคลจะต้องแสดงปี (ค.ศ.) ที่บันทึก พร้อมทั้งความหมายตามชนิดของคำศัพท์ (Part of speech) และบันทึกความเข้าใจของผู้ใช้ (User's note)
- สามารถเพิ่มเอกสารตัวอย่างประโยคที่เป็นไฟล์สกุล .txt ได้เท่านั้น
- ในการแสดงตัวอย่างประโยคจะต้องเน้นคำศัพท์ ที่ค้นหาให้สังเกตเห็นได้ง่าย
- จะสามารถเปิดเอกสารอ้างอิงของตัวอย่างประโยคแต่ละประโยคได้
- ในการแสดงความหมายของคำศัพท์ในส่วนของพจนานุกรมหลัก (Original dictionary) จะสามารถเลือกแสดงตามชนิดของคำได้
- สามารถสร้างคำศัพท์ แก้ไข และลบข้อมูลคำศัพท์ได้ทั้งในส่วนของพจนานุกรมหลัก (Original dictionary) และพจนานุกรมส่วนบุคคล (Personal dictionary) ได้
- ในการสร้างคำศัพท์ จะไม่สามารถเพิ่มคำซ้ำซ้อนกัน ทั้งในส่วนของพจนานุกรมหลัก (Original dictionary) และพจนานุกรมส่วนบุคคล (Personal dictionary)



เพื่อเป็นการตอบสนองความต้องการดังกล่าวข้างต้น สามารถแบ่งการทำงานของระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย ออกเป็น 3 ระบบย่อย ดังนี้

### 1. ระบบฐานข้อมูลพจนานุกรมหลัก (Original dictionary)

เป็นระบบเพื่อจัดเก็บข้อมูลคำศัพท์ภาษาอังกฤษ ซึ่งใช้เป็นคำศัพท์หลัก และสามารถจัดการข้อมูลเหล่านั้นได้ ไม่ว่าจะเป็นการเพิ่ม แก้ไข ลบ อีกทั้งยังสามารถค้นหาคำศัพท์ โดยระบุชนิดของคำได้ เพื่อแสดงความหมายของคำศัพท์ซึ่งจำแนกตามชนิดของคำ ตัวอย่างการใช้คำศัพท์ คำอ่าน หรือการออกเสียง คำที่มีความหมายคล้าย และ คำที่มีความหมายตรงข้าม

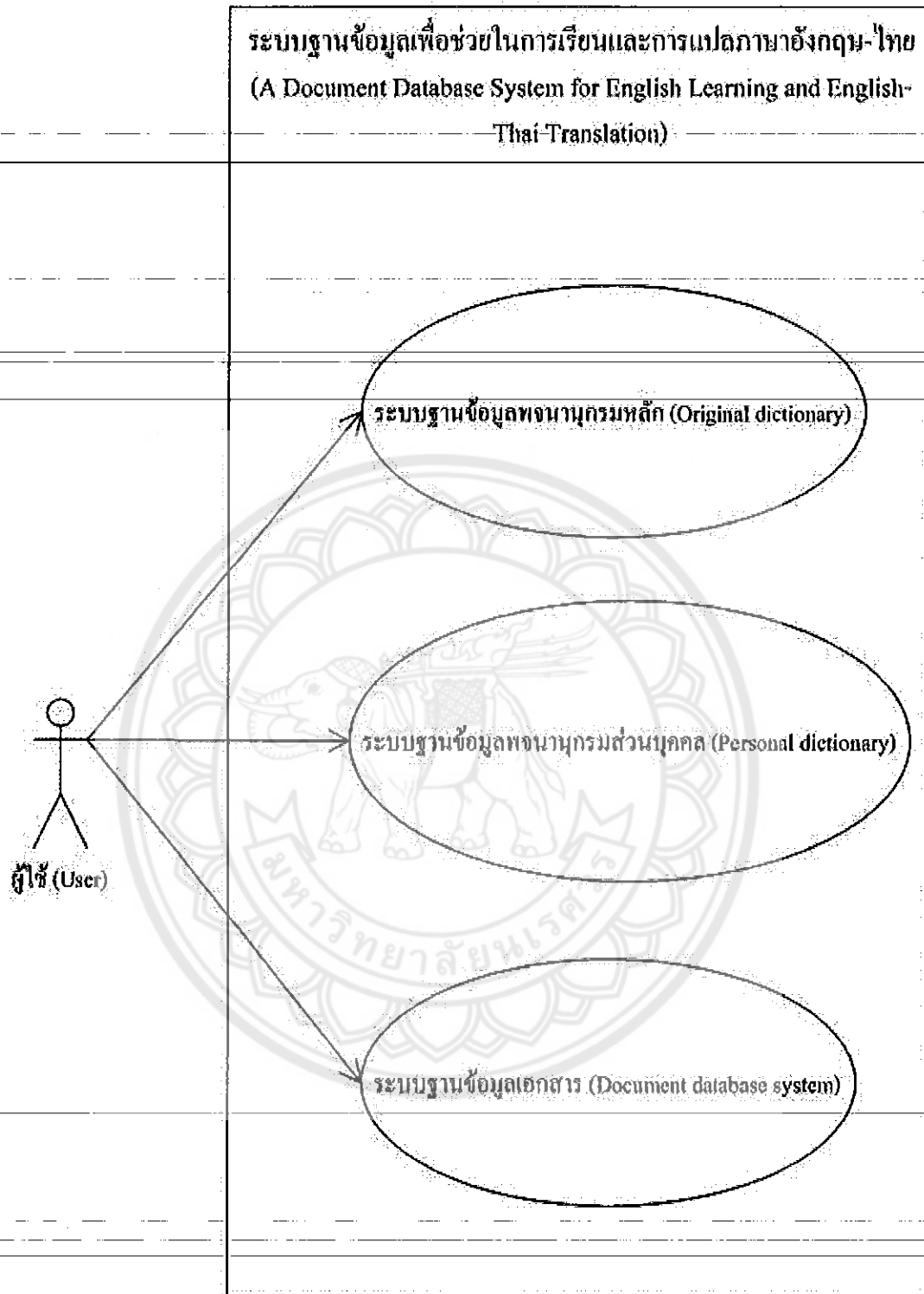
### 2. ระบบฐานข้อมูลพจนานุกรมส่วนบุคคล (Personal dictionary)

เป็นระบบเพื่อจัดเก็บข้อมูลคำศัพท์ภาษาอังกฤษของผู้ใช้ และสามารถจัดการข้อมูลทั้งหมด เช่นเดียวกับพจนานุกรมหลัก ที่สามารถเพิ่ม แก้ไข ลบ และสามารถค้นหาคำศัพท์ โดยระบุปีที่บันทึกคำศัพท์ ความหมายของคำศัพท์ และบันทึกความเข้าใจของผู้ใช้

### 3. ระบบฐานข้อมูลเอกสาร (Document database system)

เป็นระบบเพื่อจัดเก็บเอกสารภาษาอังกฤษ สามารถค้นหาประโยคภาษาอังกฤษจากเอกสารทั้งหมดในฐานข้อมูลเพื่อนำมาแสดงเป็นตัวอย่างประโยค โดยทำการเน้นคำศัพท์ที่ค้นหาเพื่อจะได้สังเกตลักษณะการใช้คำศัพท์ของแต่ละประโยค อีกทั้งยังสามารถเปิดดูเอกสารอ้างอิงของแต่ละประโยคได้ง่าย

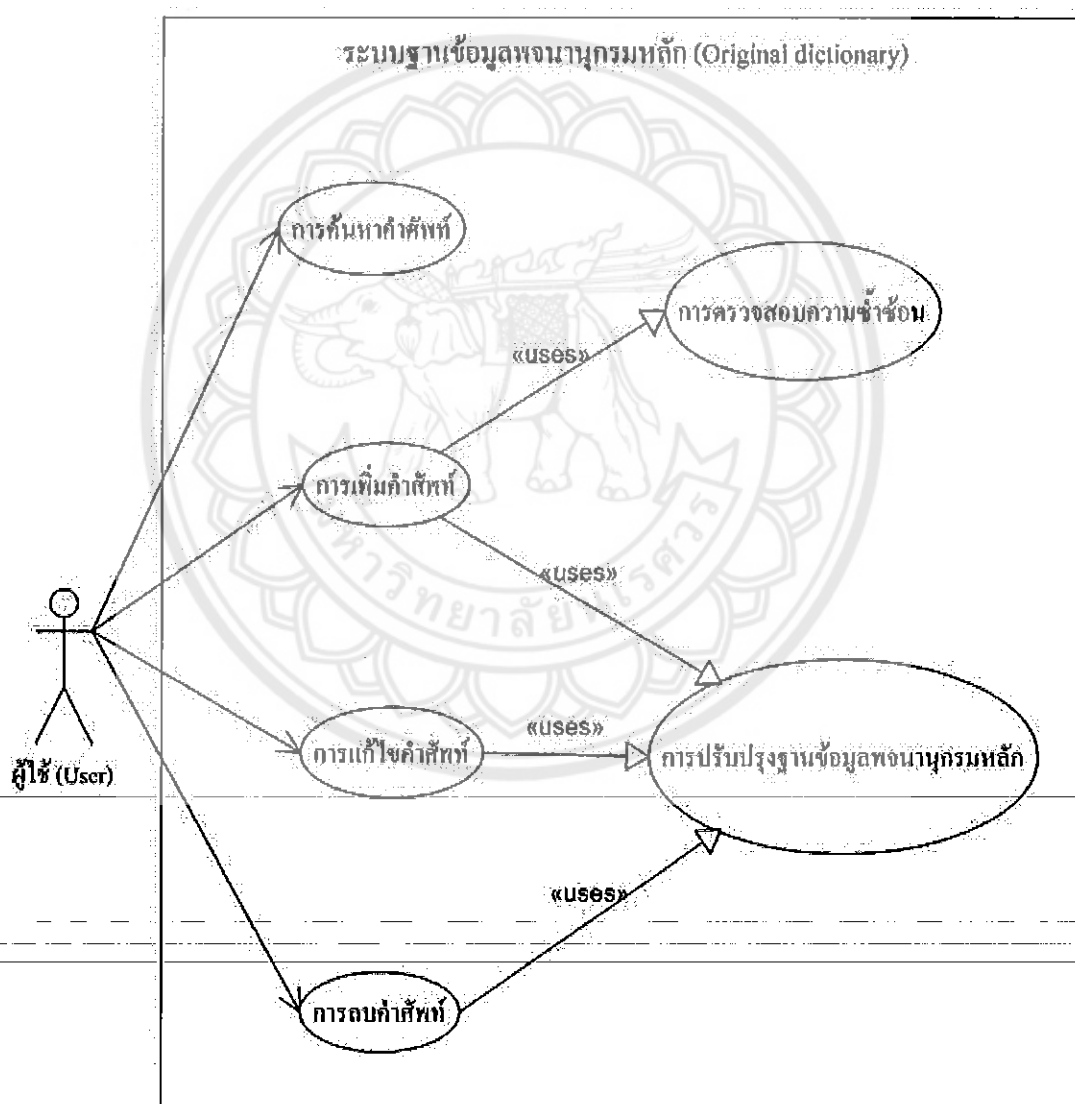
จะเห็นว่าทั้ง 3 ระบบย่อยของ ระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย นี้สามารถครอบคลุมความต้องการของผู้ใช้ได้ ดังรูป 3.1



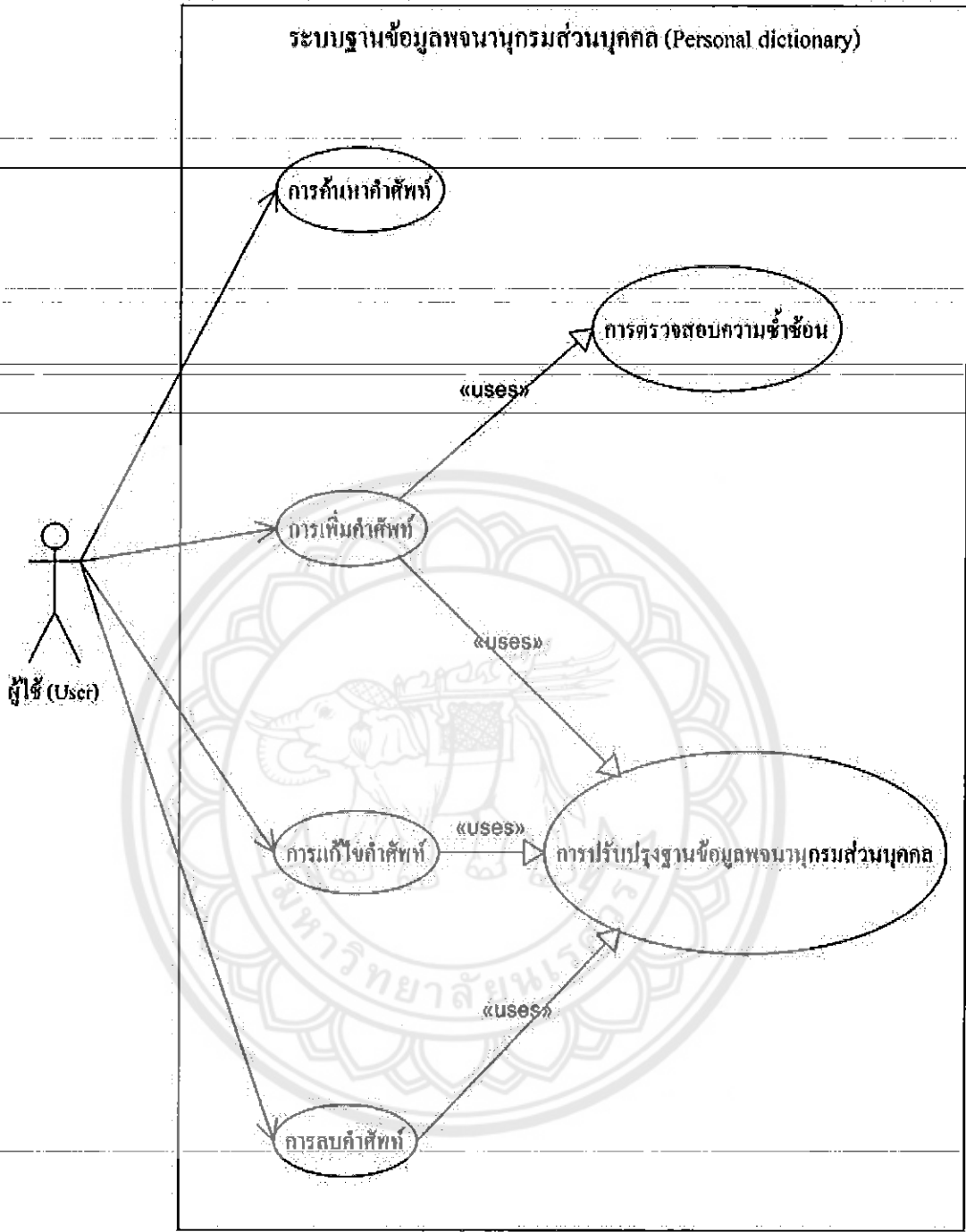
รูปที่ 3.1 แผนภาพระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย

### 3.2 แบบจำลองการทำงานของโปรแกรม

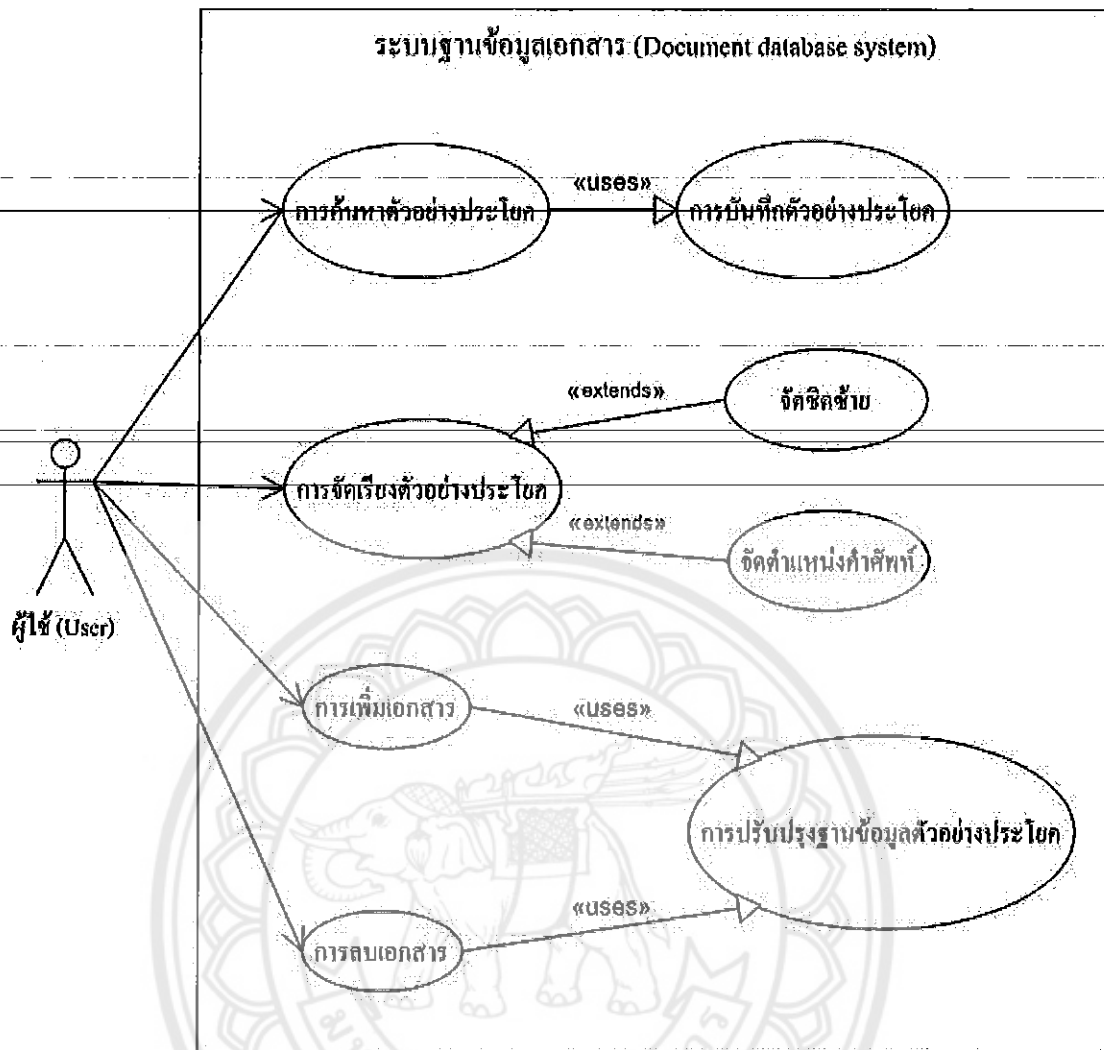
จากการศึกษาและและรวบรวมความต้องการของระบบ สามารถแบ่งระบบการทำงานของระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย ออกได้เป็น 3 ระบบ คือ ระบบฐานข้อมูลพจนานุกรมหลัก (Original dictionary), ระบบฐานข้อมูลพจนานุกรมส่วนบุคคล (Personal dictionary) และระบบฐานข้อมูลเอกสาร (Document database system) เพื่อให้สามารถเข้าใจได้ง่าย จึงได้สร้าง System Requirement Model อธิบายภาพรวมของระบบย่อยทั้ง 3 ระบบ ดังต่อไปนี้



รูปที่ 3.2 System Requirement Model ของระบบฐานข้อมูลพจนานุกรมหลัก



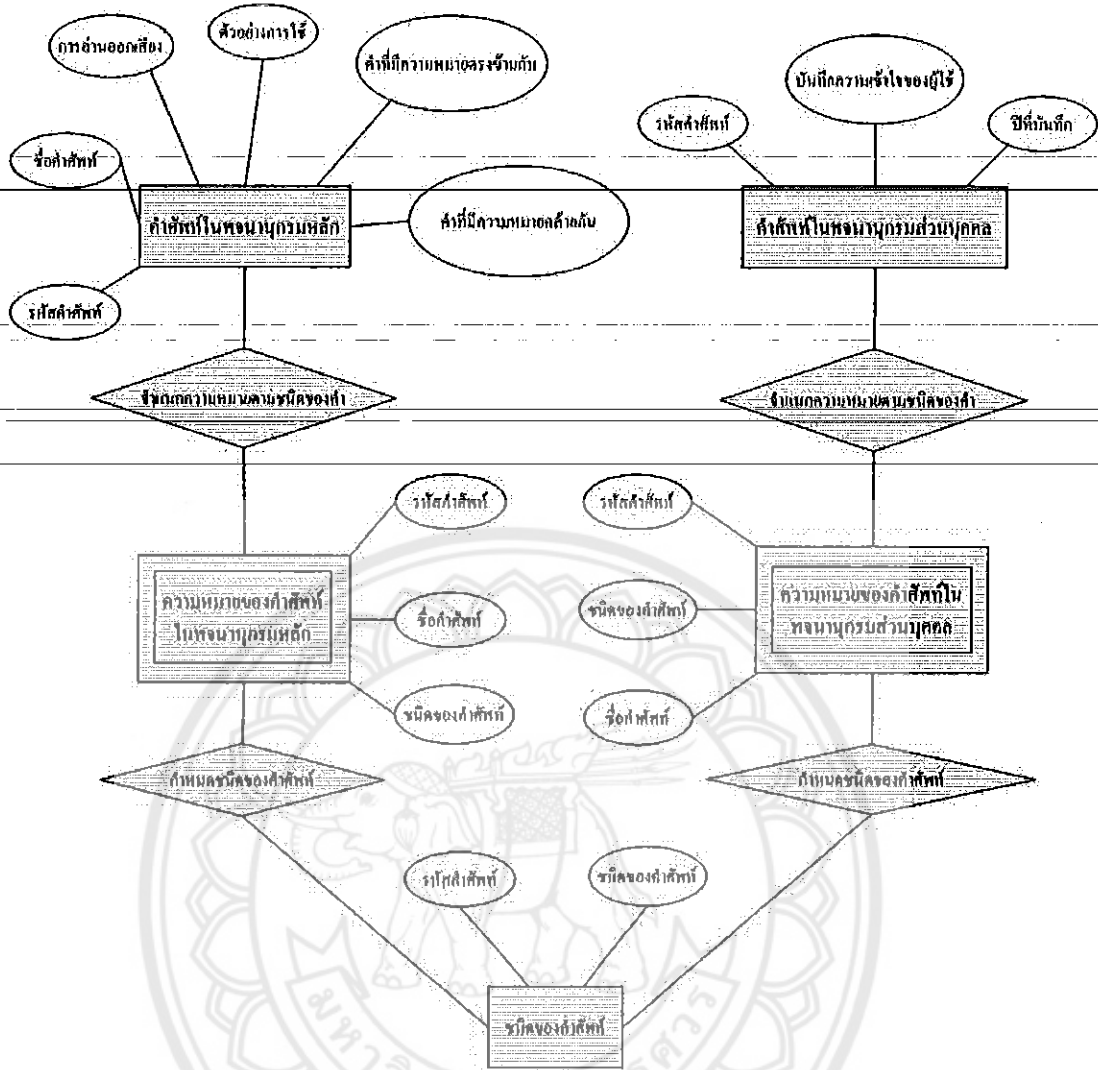
รูปที่ 3.3 System Requirement Model ของระบบฐานข้อมูลพจนานุกรมส่วนตัว



รูปที่ 3.4 System Requirement Model ของระบบฐานข้อมูลเอกสาร

### 3.3 การออกแบบฐานข้อมูล

จากการวิเคราะห์ระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย (A Document Database System for English Learning and English-Thai Translation) สามารถใช้ อี-อาร์โมเดล ในการออกแบบฐานข้อมูลพจนานุกรม



รูปที่ 3.5 อี-อาร์ไอโคแกรม แสดงโครงสร้างฐานข้อมูลพจนานุกรม

ก่อนที่จะมีการสร้างโปรแกรม ได้ออกแบบในส่วนของฐานข้อมูลทั้งหมด 5 ตารางคือ

- |                    |   |
|--------------------|---|
| 1. PartOfSpeech    | ฐานข้อมูลชนิดของคำศัพท์                         |
| 2. Original        | ฐานข้อมูลคำศัพท์ในพจนานุกรมหลัก                 |
| 3. Original_detail | ฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมหลัก      |
| 4. Personal        | ฐานข้อมูลพจนานุกรมส่วนบุคคล                     |
| 5. Personal_detail | ฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมส่วนบุคคล |

ตารางที่ 3.1 ฐานข้อมูลชนิดของคำศัพท์ (PartOfSpeech)

ID	PartOfSpeech
----	--------------

ตารางรายละเอียด ฐานข้อมูลชนิดของคำศัพท์

Field	Type	Size (Byte)	Null	Key	Description
ID	Long Integer	8	No	pk	รหัสคำศัพท์
PartOfSpeech	String	2	No	-	ชนิดของคำศัพท์

ตารางที่ 3.2 ฐานข้อมูลคำศัพท์ในพจนานุกรมหลัก (Original)

ID	Vocabulary	Pronunciation	Example	Synonym	Antonym
----	------------	---------------	---------	---------	---------

ตารางรายละเอียดฐานข้อมูลคำศัพท์ในพจนานุกรมหลัก

Field	Type	Size (Byte)	Null	Key	Description
ID	Long Integer	8	No	pk	รหัสคำศัพท์
Vocabulary	String	2	No	-	ชื่อคำศัพท์
Pronunciation	String	2	No	-	การอ่านออกเสียง
Example	String	2	Yes	-	ตัวอย่างการใช้
Synonym	String	2	Yes	-	คำที่มีความหมายคล้ายกัน
Antonym	String	2	Yes	-	คำที่มีความหมายตรงกันข้าม

ตารางที่ 3.3 ฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมหลัก (Original\_detail)

ID_Voc	Pos	Mean
--------	-----	------

ตารางรายละเอียดฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมหลัก

Field	Type	Size (Byte)	Null	Key	Description
ID_Voc	Long Integer	8	No	pk	รหัสคำศัพท์
Pos	String	2	No	-	ชนิดของคำศัพท์
Mean	String	2	No	-	ความหมายของคำศัพท์

ตารางที่ 3.4 ฐานข้อมูลคำศัพท์ในพจนานุกรมส่วนบุคคล (Personal)

ID	Vocabulary	YearSave	UserNote
----	------------	----------	----------

ตารางรายละเอียดฐานข้อมูลคำศัพท์ในพจนานุกรมส่วนบุคคล

Field	Type	Size (Byte)	Null	Key	Description
ID	Long Integer	8	No	pk	รหัสคำศัพท์
Vocabulary	String	2	No	-	ชื่อคำศัพท์
YearSave	String	2	No	-	ปีที่บันทึก
UserNote	String	2	Yes	-	บันทึกความเข้าใจของผู้ใช้

ตารางที่ 3.5 ฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมส่วนบุคคล (Personal\_detail)

ID	Pos	Mean
----	-----	------

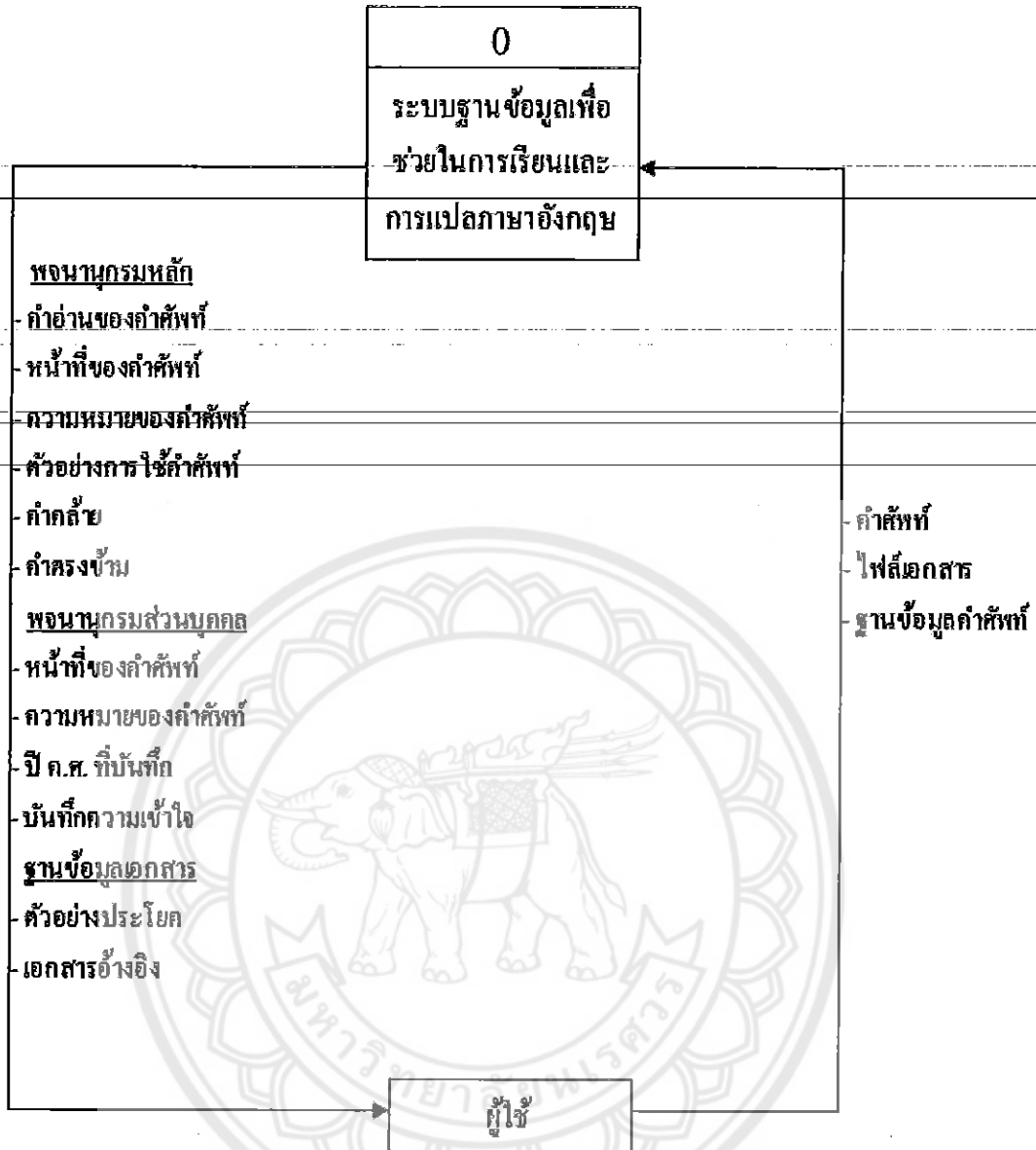
ตารางรายละเอียดฐานข้อมูลความหมายของคำศัพท์ในพจนานุกรมส่วนบุคคล

Field	Type	Size (Byte)	Null	Key	Description
ID	Long Integer	8	No	pk	รหัสคำศัพท์
Pos	String	2	No	-	ชนิดของคำศัพท์
Mean	String	2	No	-	ความหมายของคำศัพท์

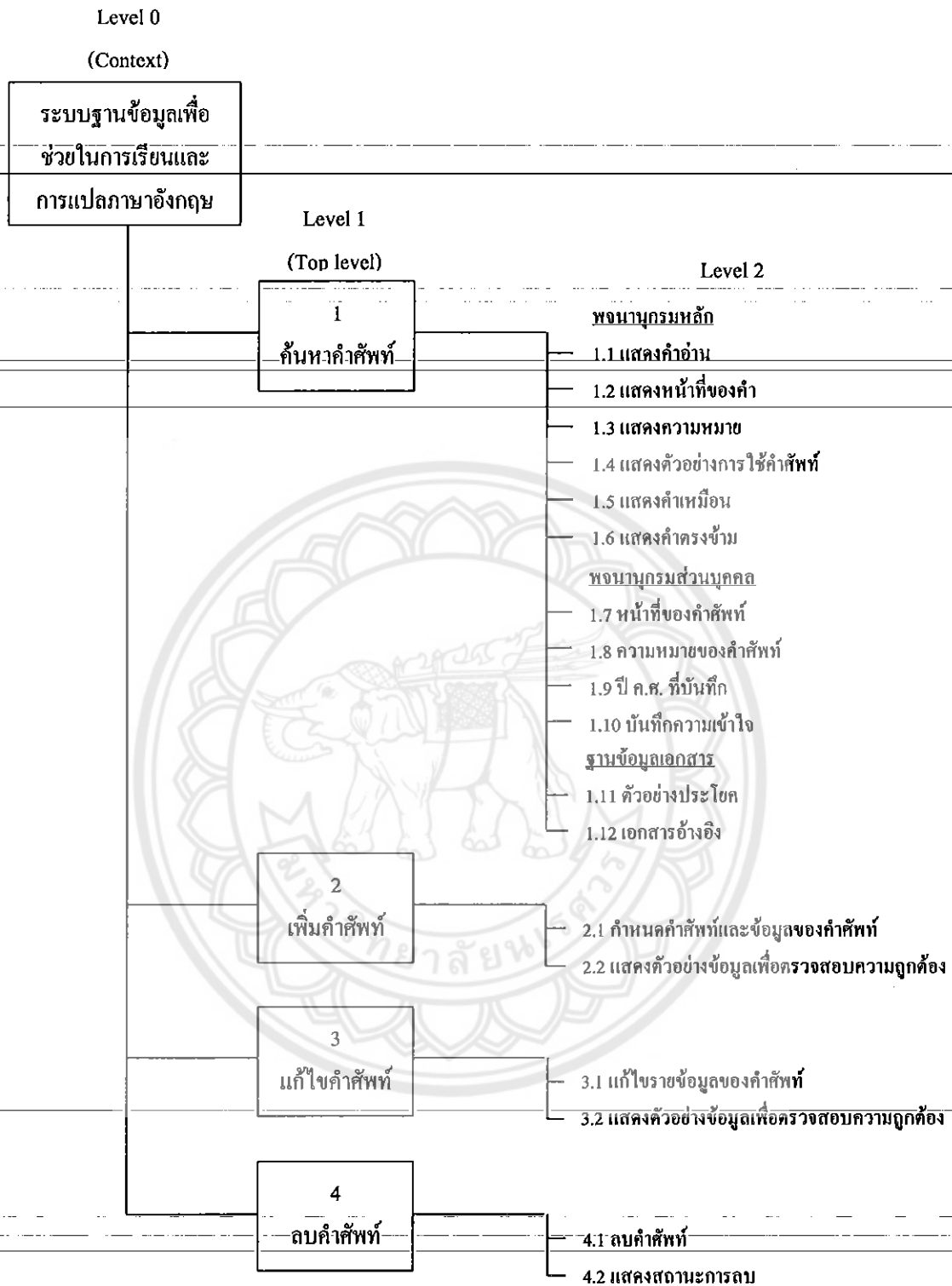
### 3.4 การออกแบบโปรแกรมเพื่อใช้งานร่วมกับระบบฐานข้อมูล

เพื่อให้สามารถเข้าใจได้ง่ายในการออกแบบโปรแกรมให้สอดคล้องกับฐานข้อมูล ทางผู้จัดทำโครงการจึงได้ออกแบบ Context Diagram, แผนภาพ Process Decomposition Diagram, แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) และ โครงสร้างการเขียนโปรแกรม (Flow Chart) ขึ้น ซึ่งจะทำให้เห็นความสัมพันธ์ระหว่างโปรแกรมกับข้อมูลที่เกี่ยวข้องว่าข้อมูลดังกล่าวมาจากส่วนใดและข้อมูลถูกส่งไปที่ส่วนใด





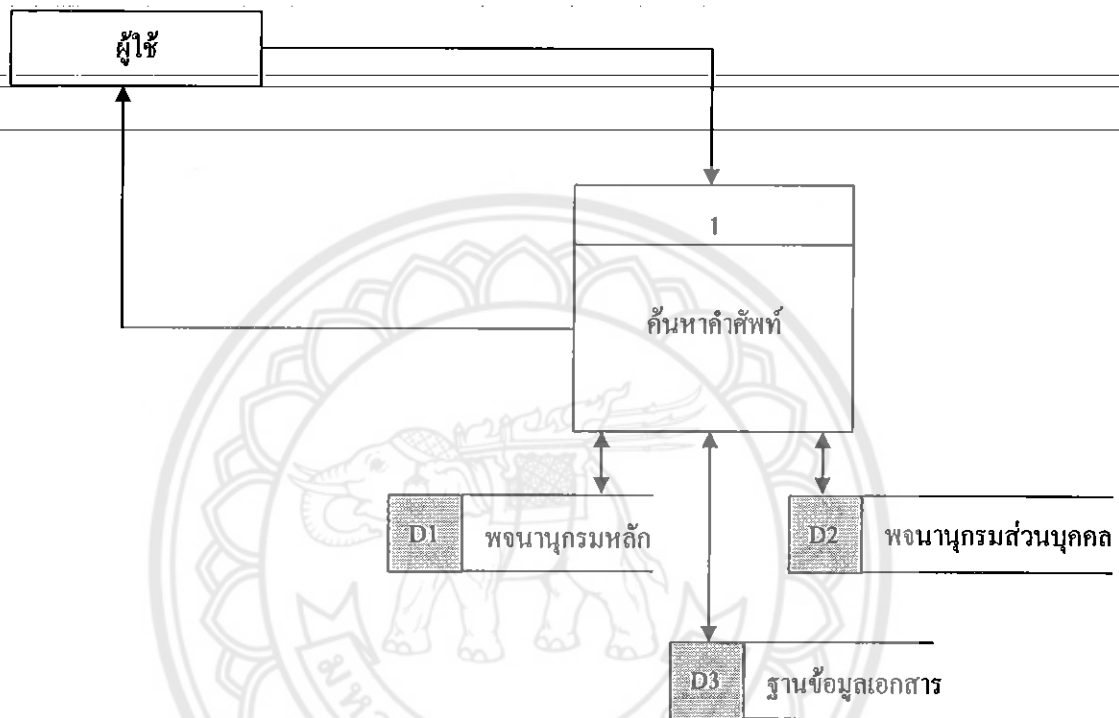
รูปที่ 3.6 Context Diagram ของระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-  
ไทย



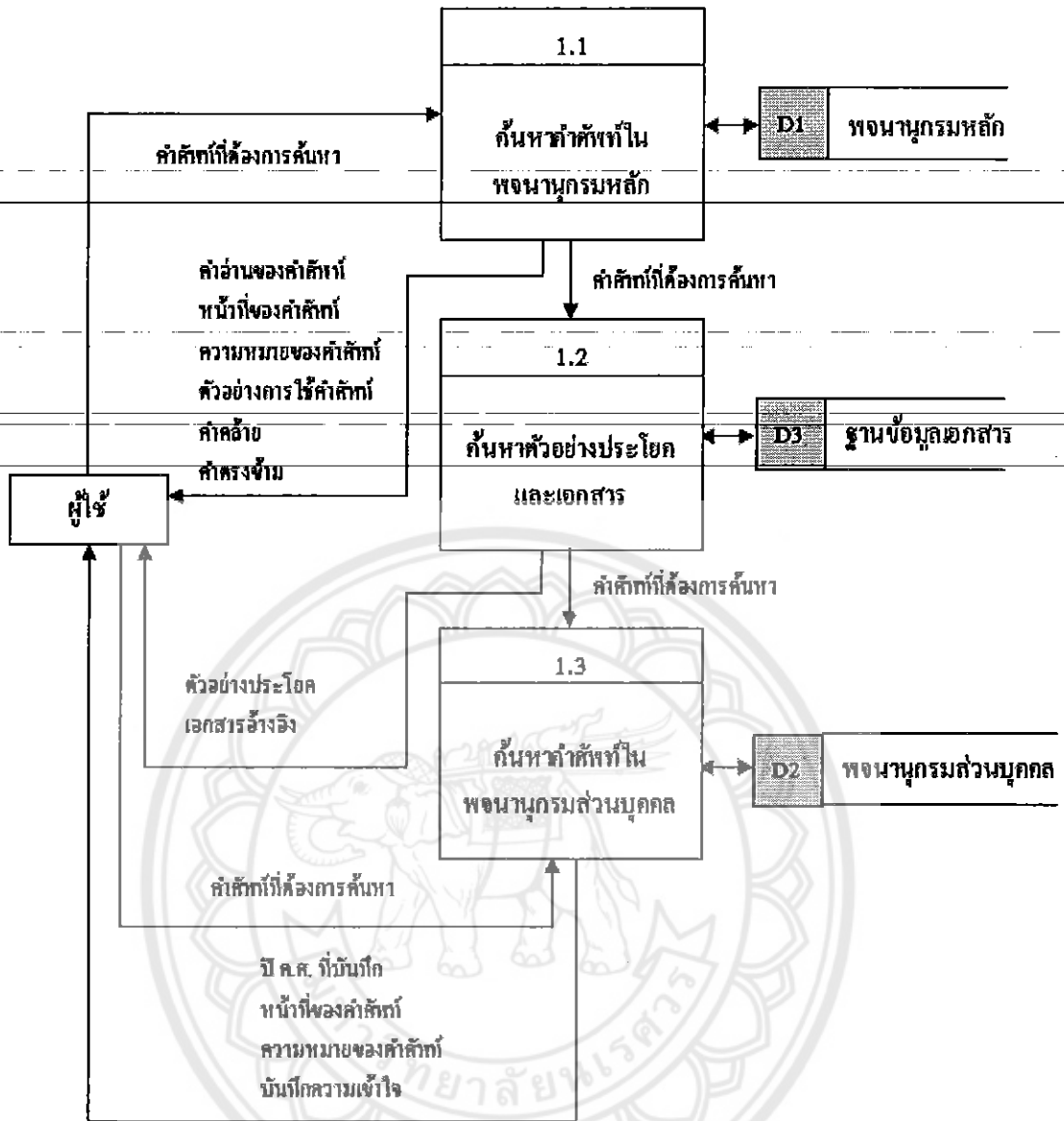
รูปที่ 3.7 Process Decomposition Diagram ของระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย

### 3.4.1 แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) และโครงสร้างการเขียนโปรแกรม (Flow Chart) ของระบบค้นหาคำศัพท์

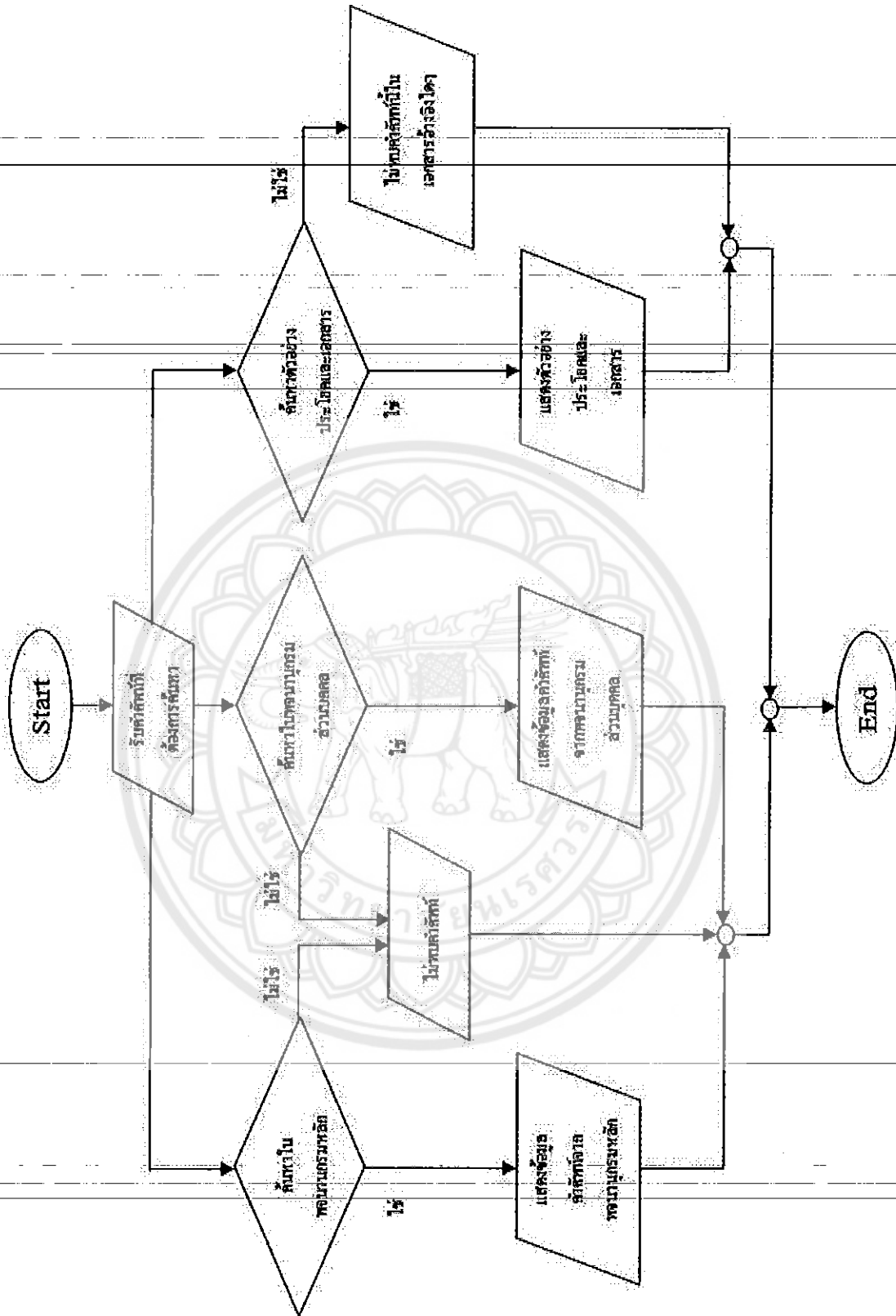
แผนภาพกระแสข้อมูลของระบบค้นหาคำศัพท์ จะรับคำศัพท์จากผู้ใช้ แล้วทำการค้นหาในฐานข้อมูลพจนานุกรมหลัก ฐานข้อมูลพจนานุกรมส่วนบุคคล และฐานข้อมูลเอกสาร แล้วแสดงผลข้อมูลคำศัพท์



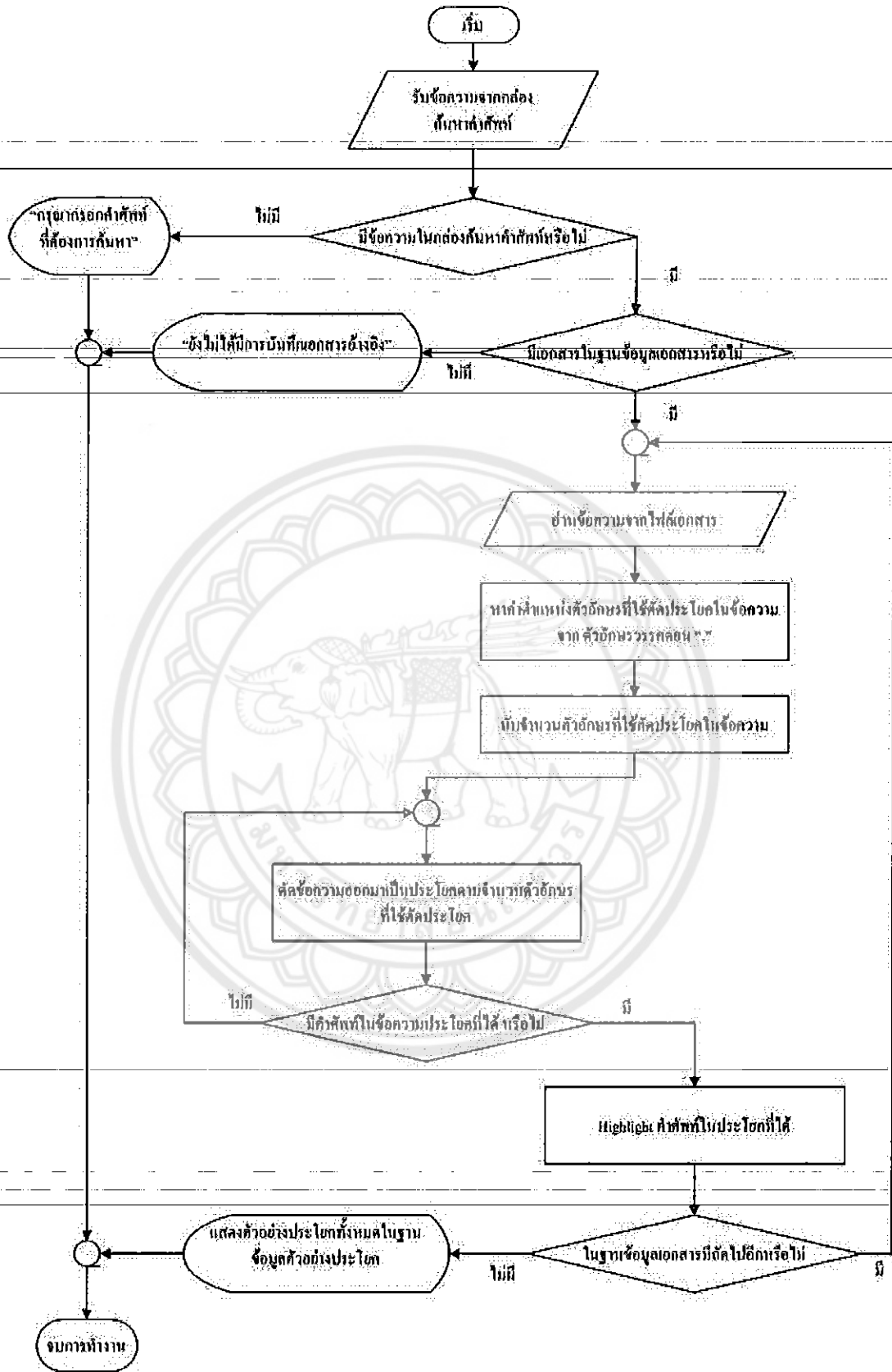
รูปที่ 3.8 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบค้นหาคำศัพท์



รูปที่ 3.9 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบค้นหาคำศัพท์



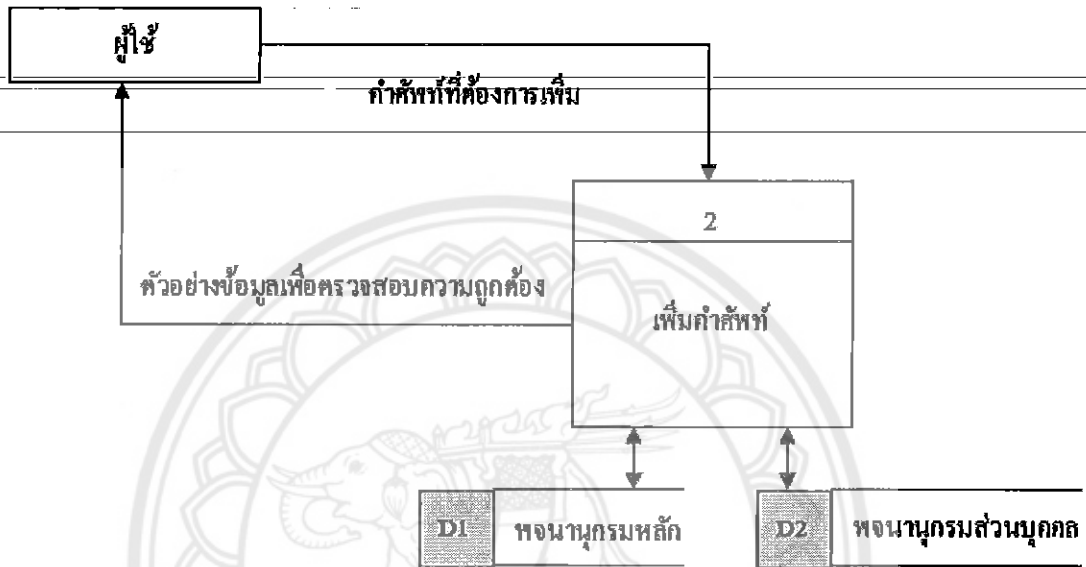
รูปที่ 3.10 โครงสร้างการเขียนโปรแกรมแสดงระบบค้นหาศัพท์



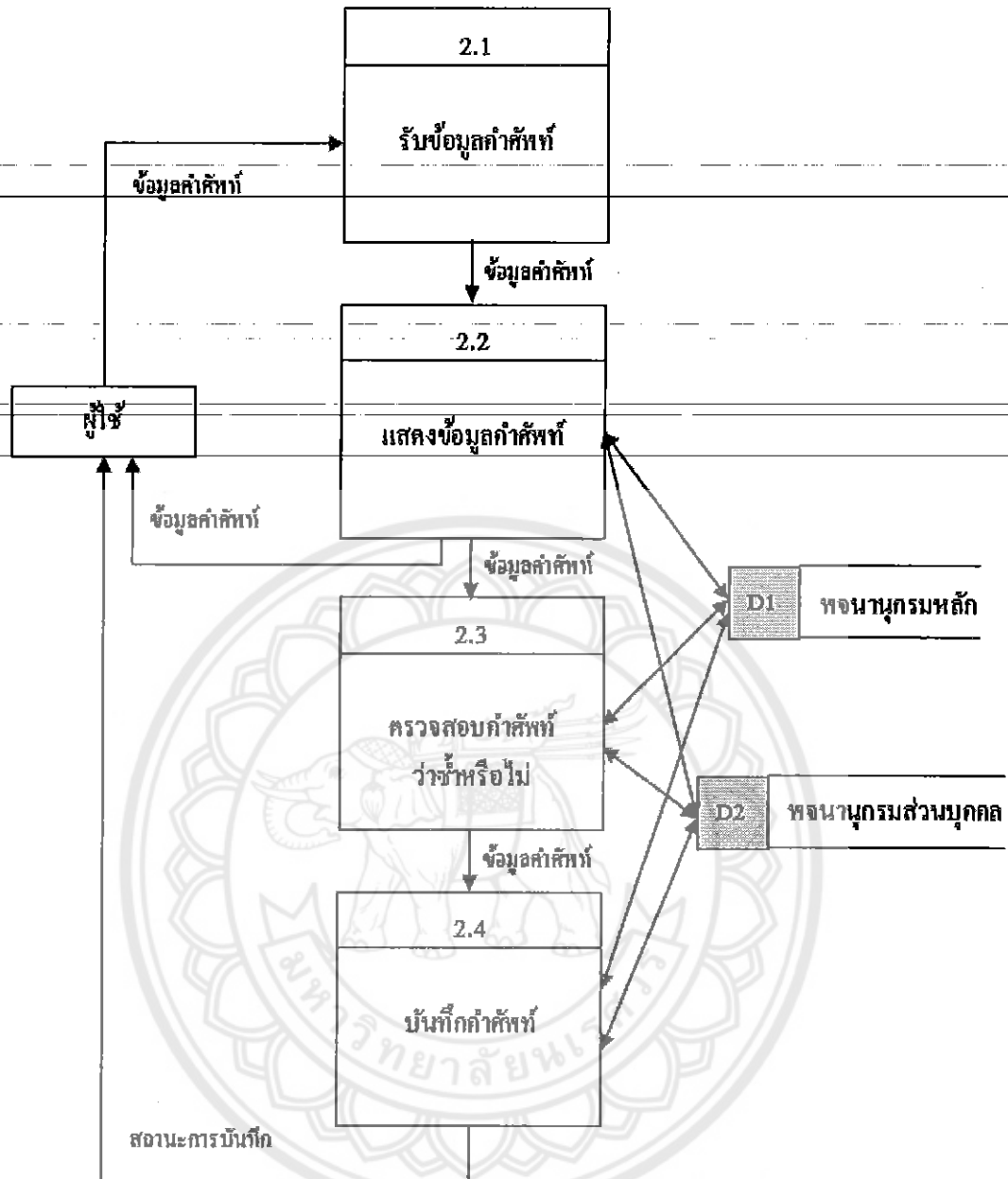
รูปที่ 3.11 โครงสร้างการเขียน โปรแกรมแสดงระบบค้นหาตัวอย่างประโยค

### 3.4.2 แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) และโครงสร้างการเขียนโปรแกรม (Flow Chart) ของระบบเพิ่มคำศัพท์

แผนภาพกระแสข้อมูลของระบบเพิ่มคำศัพท์ จะรับข้อมูลคำศัพท์ที่ผู้ใช้ต้องการเพิ่มลงในฐานข้อมูลพจนานุกรมหลักหรือฐานข้อมูลพจนานุกรมส่วนบุคคล แล้วแสดงตัวอย่างข้อมูลเพื่อตรวจสอบความถูกต้อง

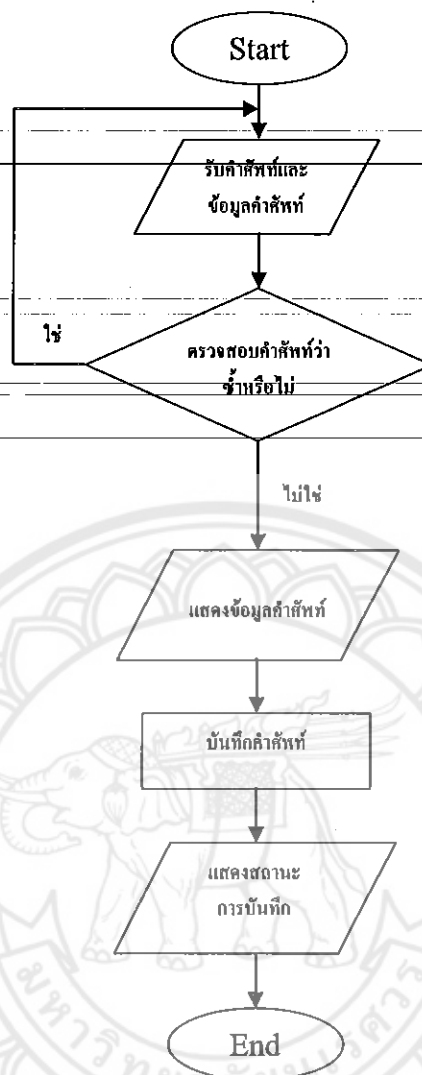


รูปที่ 3.12 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบเพิ่มคำศัพท์



รูปที่ 3.13 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบเพิ่มคำศัพท์

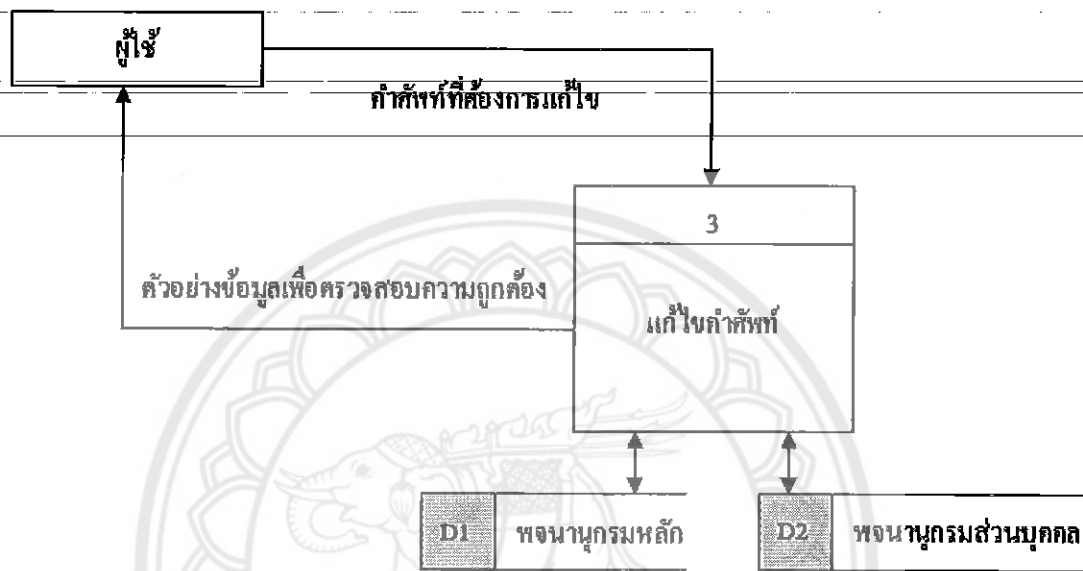




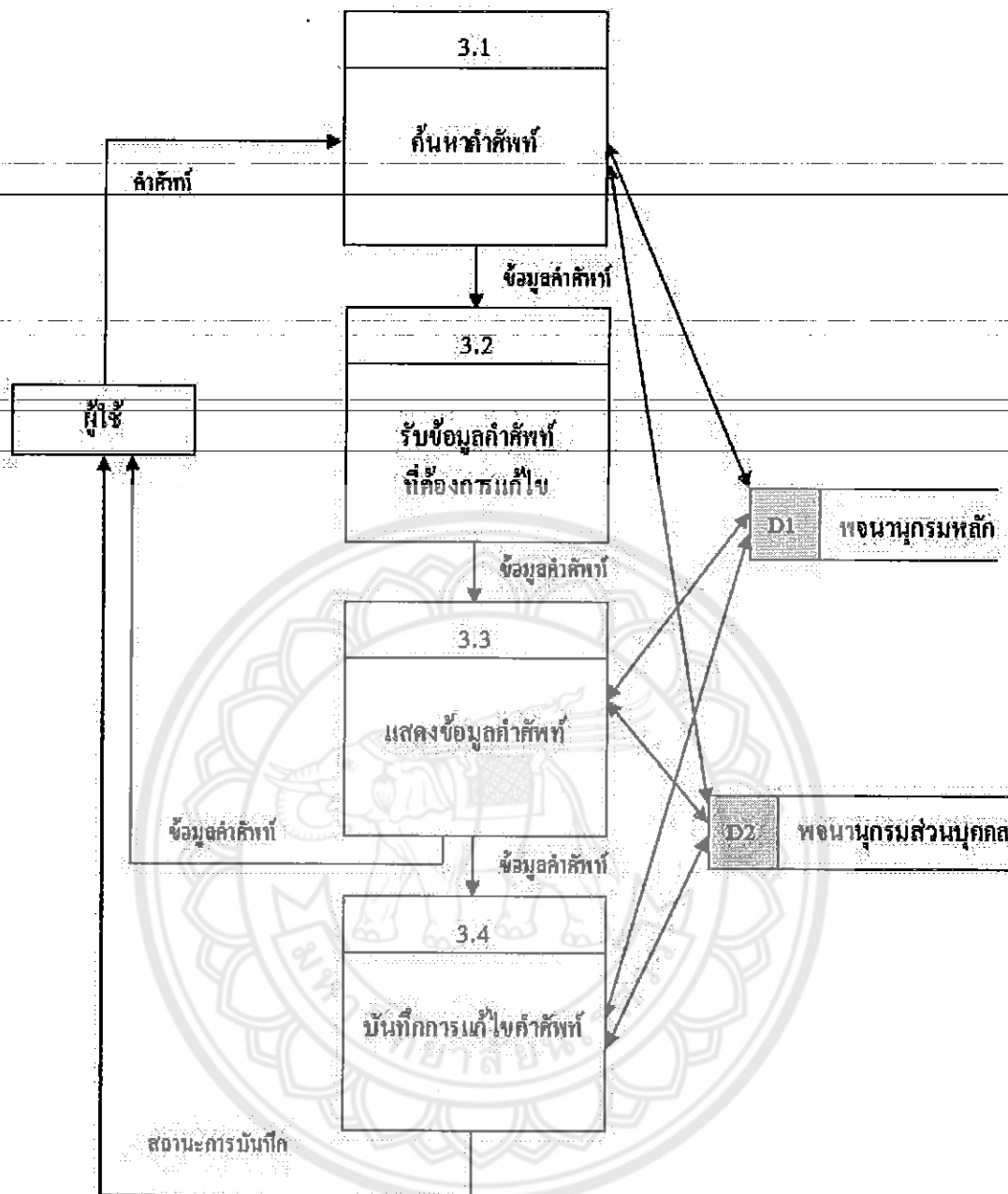
รูปที่ 3.14 โครงสร้างการเขียนโปรแกรมแสดงระบบเพิ่มคำศัพท์

### 3.4.3 แผนผังกระแสการไหลของข้อมูล (Data Flow Chart) และโครงสร้างการเขียนโปรแกรม (Flow Chart) ของระบบแก้ไขคำศัพท์

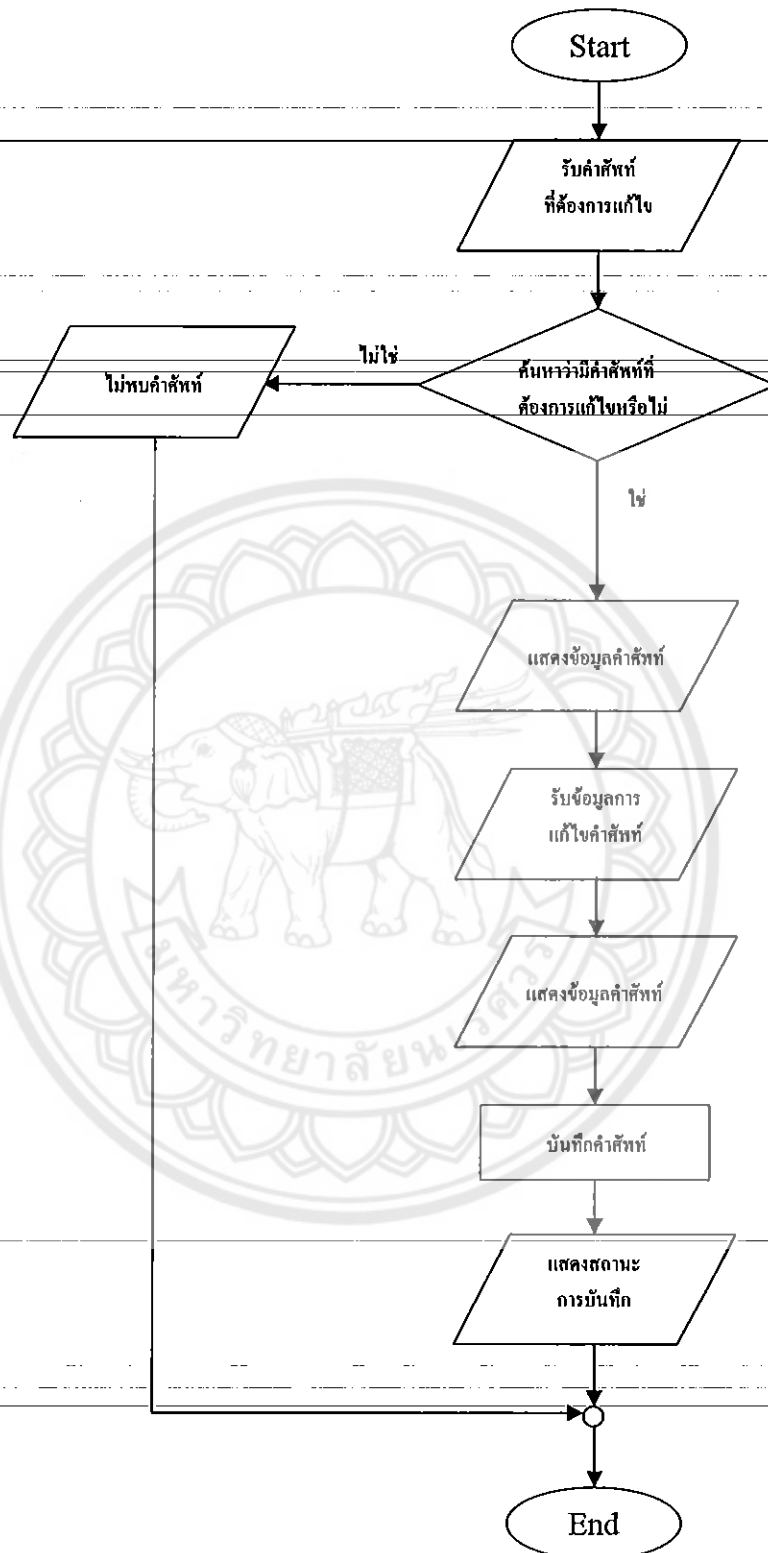
แผนภาพกระแสข้อมูลของระบบแก้ไขคำศัพท์ จะรับคำศัพท์ที่ต้องการแก้ไขจากผู้ใช้แล้วทำการค้นหาในฐานข้อมูลพจนานุกรมหลักหรือฐานข้อมูลพจนานุกรมส่วนบุคคล เมื่อพบว่าไม่มีคำศัพท์คำนั้น จะให้ผู้ใช้ทำการแก้ไขข้อมูลคำศัพท์ แล้วแสดงตัวอย่างข้อมูลเพื่อตรวจสอบความถูกต้อง



รูปที่ 3.15 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบแก้ไขคำศัพท์



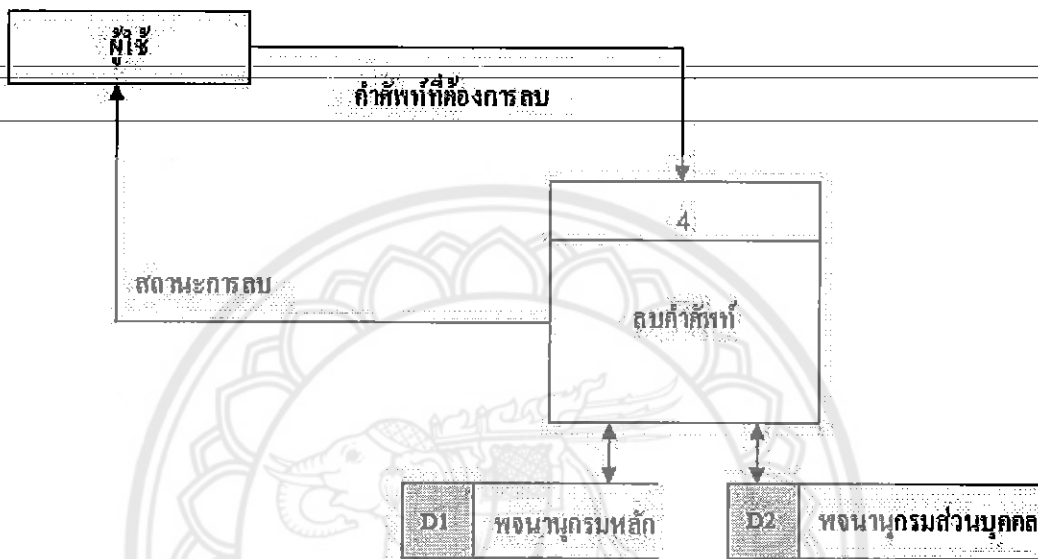
รูปที่ 3.16 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบแก้ไขคำศัพท์



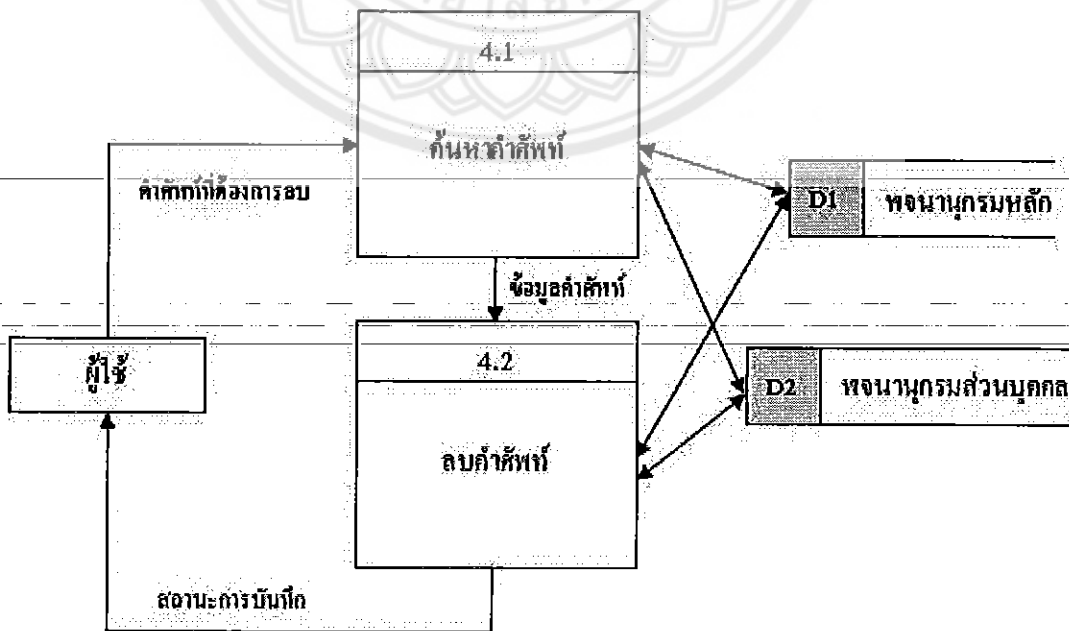
รูปที่ 3.17 โครงสร้างการเขียนโปรแกรมแสดงระบบแก้ไขคำศัพท์

3.4.4 แผนผังกระแสการไหลของข้อมูล (Data Flow Chart) และโครงสร้างการเขียนโปรแกรม (Flow Chart) ของระบบลบคำศัพท์

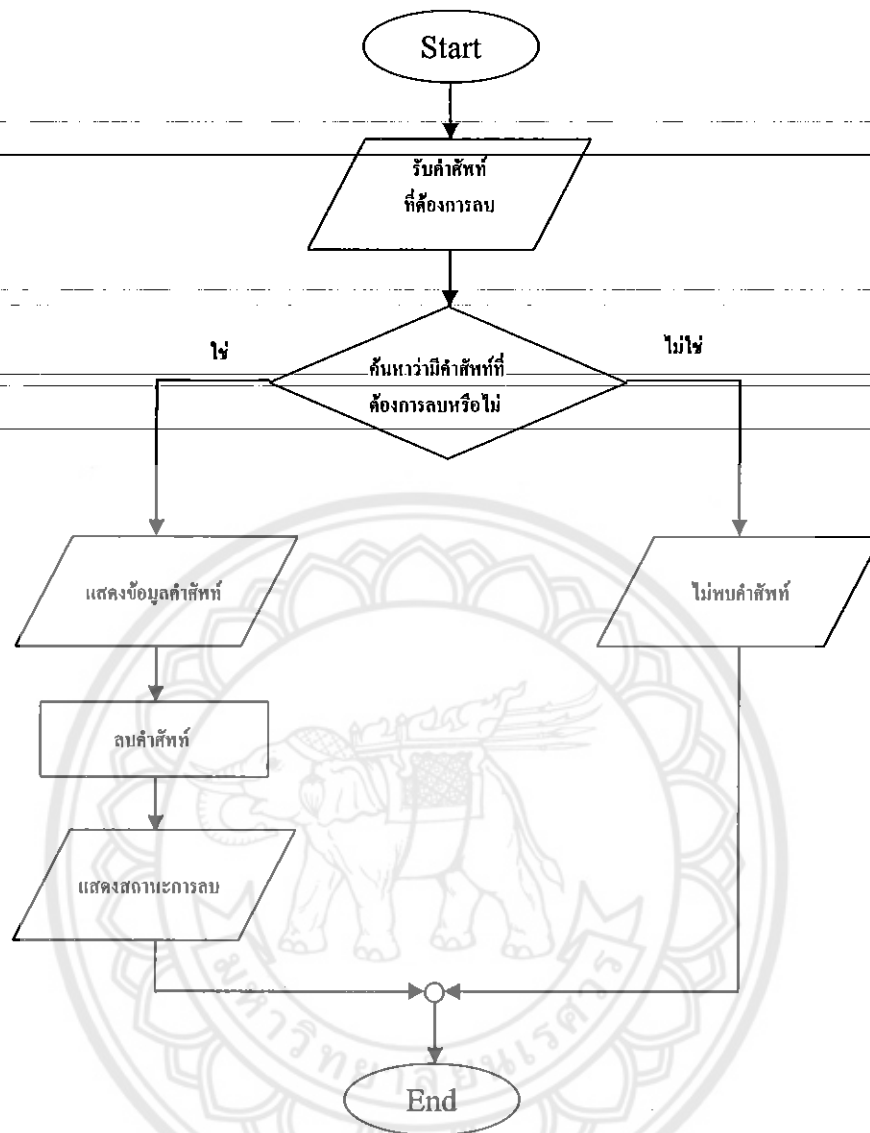
แผนภาพกระแสข้อมูลของระบบลบคำศัพท์ จะรับคำศัพท์ที่ต้องการลบจากผู้ใช้ แล้วทำการค้นหาในฐานข้อมูลพจนานุกรมหลักหรือฐานข้อมูลพจนานุกรมส่วนบุคคล เมื่อพบว่ามีคำศัพท์คำนั้น ระบบก็จะทำการลบคำศัพท์ และแสดงสถานะของการลบให้ผู้ใช้ทราบ



รูปที่ 3.18 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบลบคำศัพท์



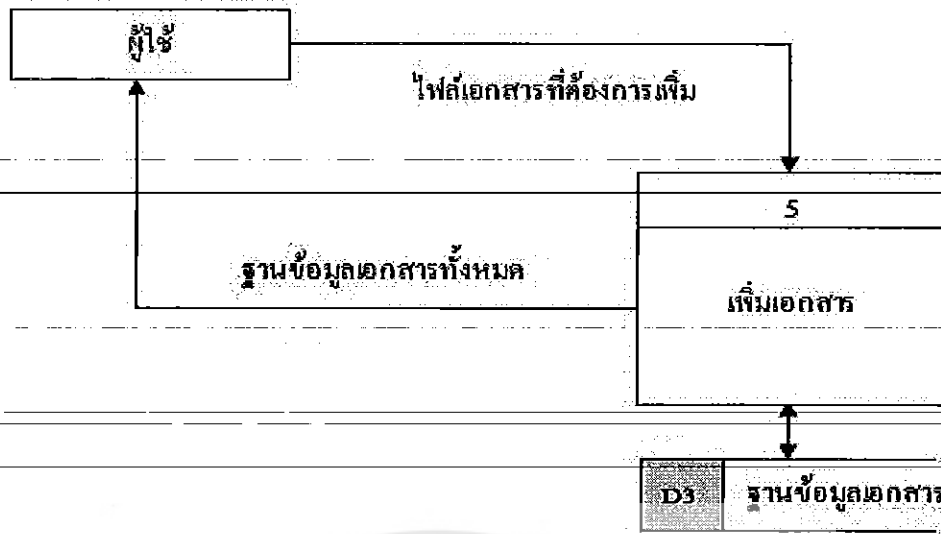
รูปที่ 3.19 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบลบคำศัพท์



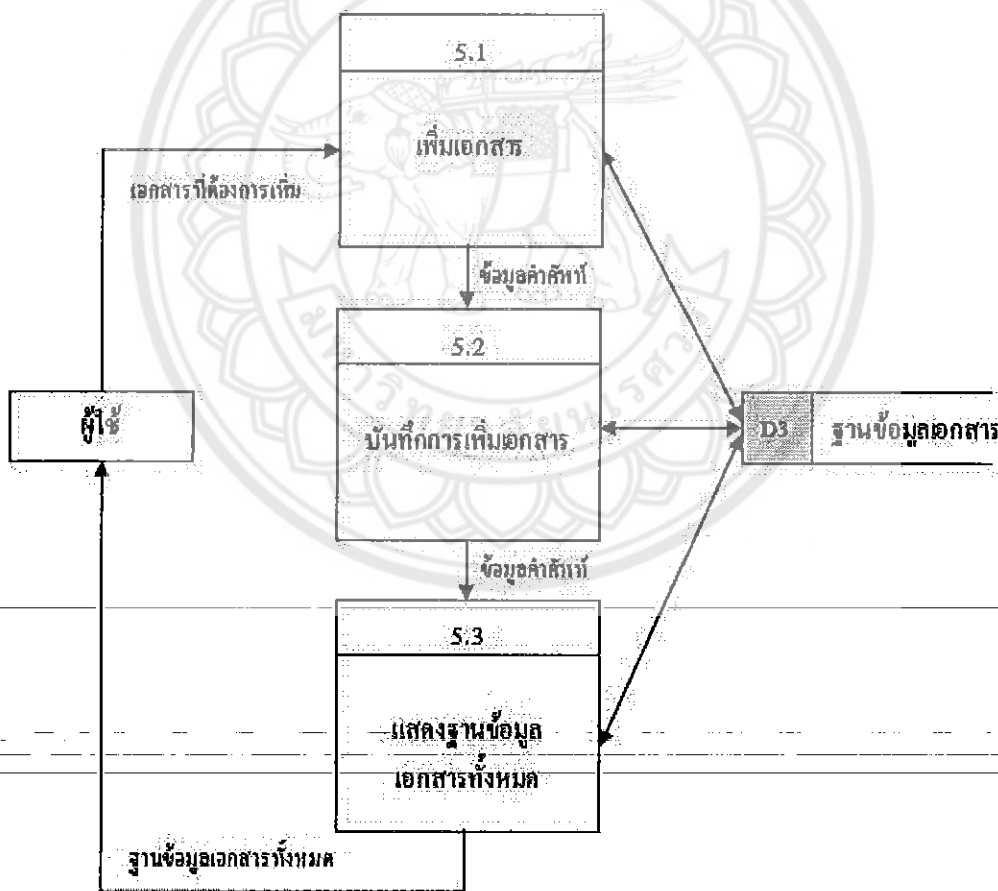
รูปที่ 3.20 โครงสร้างการเขียนโปรแกรม แสดงระบบลบคำศัพท์

### 3.4.5 แผนผังกระแสการไหลของข้อมูล (Data Flow Chart) และโครงสร้างการเขียนโปรแกรม (Flow Chart) ของระบบเพิ่มเอกสาร

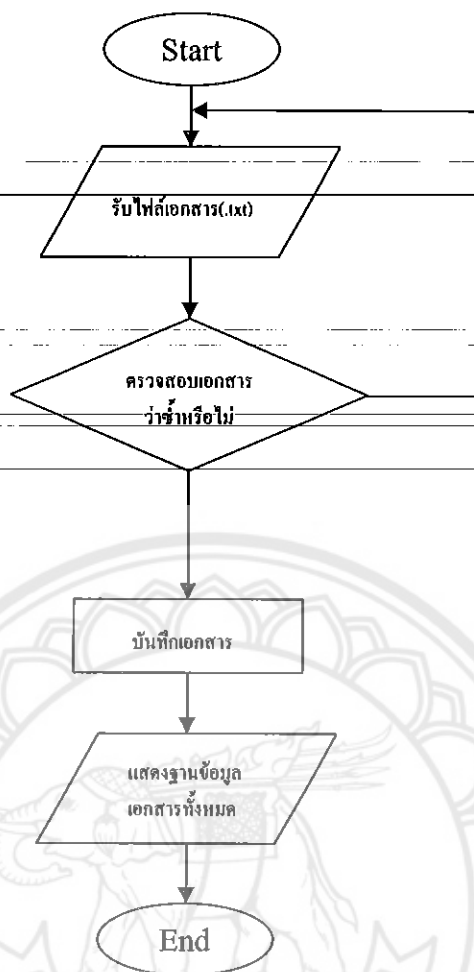
แผนภาพกระแสข้อมูลของระบบเพิ่มเอกสาร จะรับไฟล์เอกสาร (\*.txt) ที่ต้องการเพิ่มจากผู้ใช้ แล้วทำการเพิ่มไฟล์เอกสารลงในฐานข้อมูลเอกสาร และแสดงข้อมูล



รูปที่ 3.21 แผนภาพกระแสข้อมูลระดับที่ 1 (DFD-Level 1) ระบบเพิ่มเอกสาร



รูปที่ 3.22 แผนภาพกระแสข้อมูลระดับที่ 2 (DFD-Level 2) ระบบเพิ่มเอกสาร

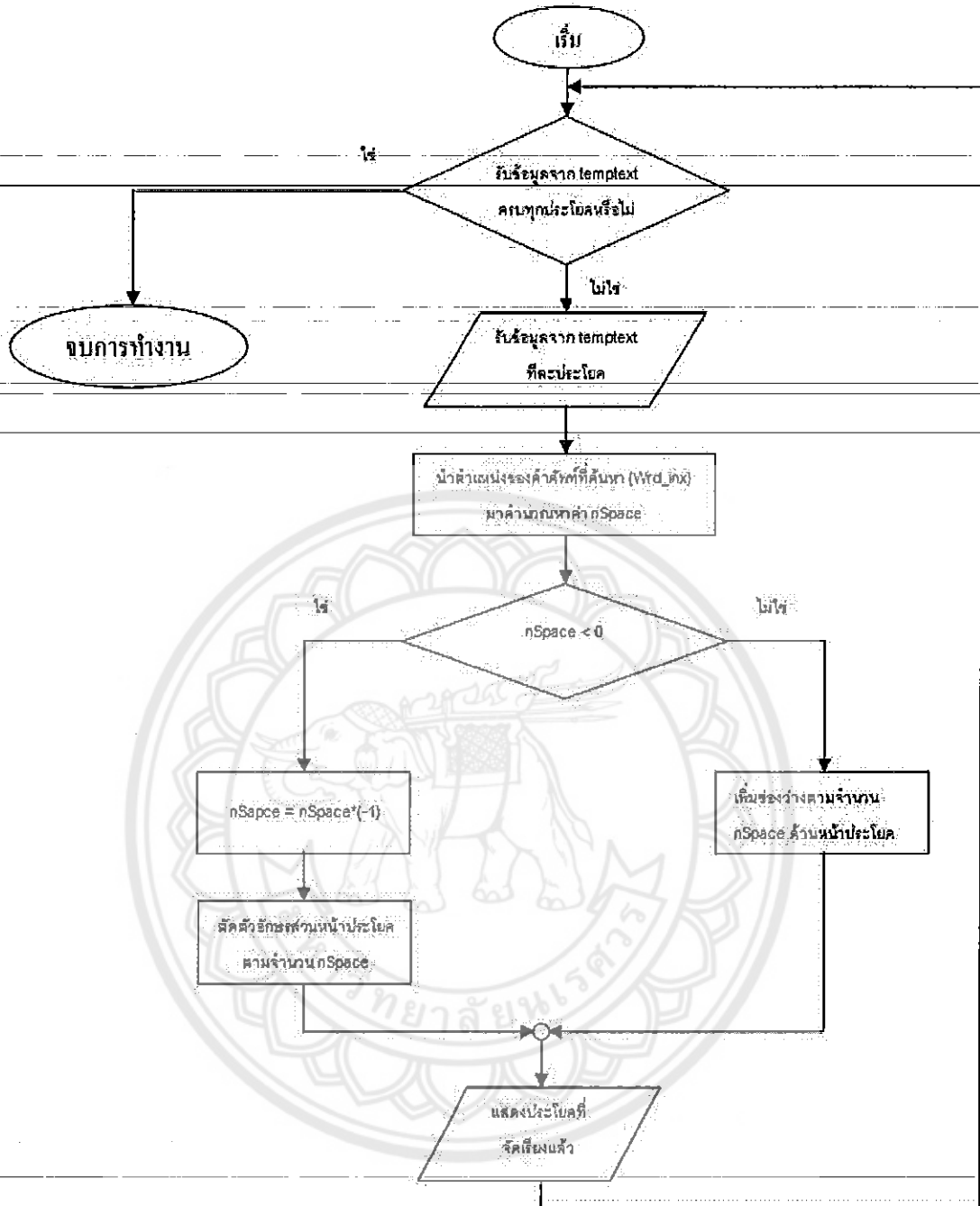


รูปที่ 3.23 โครงสร้างการเขียนโปรแกรม แสดงระบบเพิ่มเอกสาร

### 3.4.6 โครงสร้างการเขียนโปรแกรม (Flow Chart) ของการจัดเรียงตัวอย่างประโยค

ในการแสดงตัวอย่างประโยคเพื่อให้ สังเกตคำศัพท์ได้ง่ายจึง ได้มีการจัดเรียงให้ตำแหน่งของ คำศัพท์ที่ค้นหาอยู่ตรงกันทุกประโยค ซึ่งมีโครงสร้างการทำงาน ดังรูปที่ 3.24





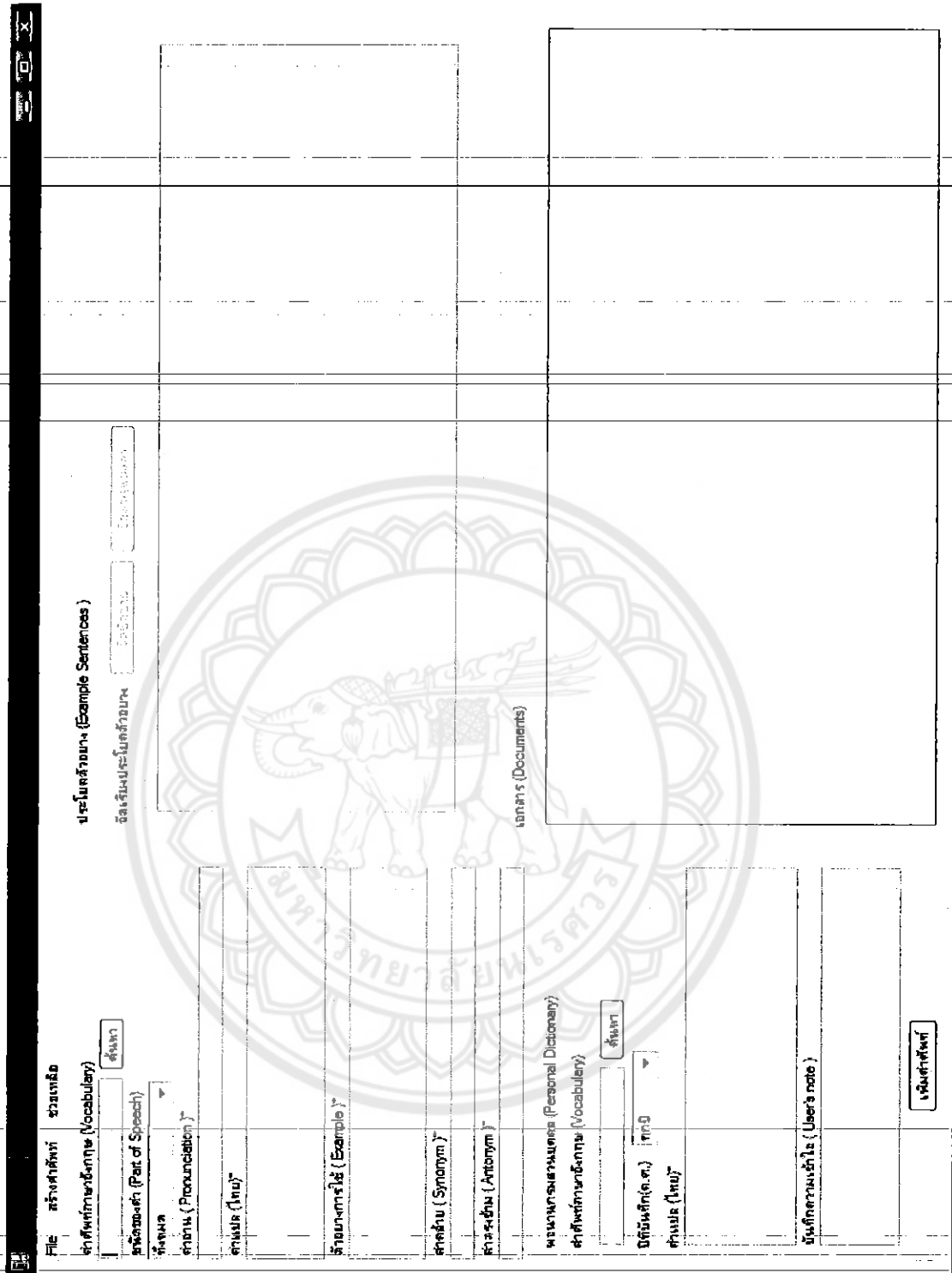
รูปที่ 3.24 โครงสร้างการจัดเรียงตัวอย่างประโยคให้ตำแหน่งคำศัพท์ที่ค้นหาตรงกัน

### 3.5 แบบจำลองการออกแบบโปรแกรม

กระบวนการการพัฒนาโครงการใดๆ นั้นมีวัตถุประสงค์เพื่อทำการสนับสนุนการทำงานของ  
ผู้ใช้โปรแกรมเป็นหลัก หรือสนับสนุนความต้องการของผู้ใช้ดังนี้

#### 3.5.1 รูปแบบการใช้งานโปรแกรม

ผู้จัดทำโครงการได้ออกแบบการใช้งานโปรแกรมออกเป็น 3 ส่วน คือ พจนานุกรมหลัก,  
พจนานุกรมส่วนบุคคล, ตัวอย่างประโยคและเอกสาร ดังรูปที่ 3.25



รูปที่ 3.25 แบบจำลองการใช้งาน โปรแกรม

### 3.5.2 รูปแบบการบันทึกข้อมูล

ผู้จัดทำโครงการได้ออกแบบการบันทึกข้อมูลในฐานข้อมูลมีทั้งหมด 3 ส่วนคือ การสร้างหรือเพิ่มคำศัพท์ในฐานข้อมูลพจนานุกรมหลัก ดังรูปที่ 3.26 และ การสร้างหรือเพิ่มคำศัพท์ใน

ฐานข้อมูลพจนานุกรมส่วนบุคคล คังรูปที่ 3.27 และการเพิ่มเอกสารอ้างอิงหรือเอกสารนำมาใช้เป็นตัวอย่างประโยคในฐานข้อมูลเอกสาร คังรูปที่ 3.28

สร้างคำศัพท์

สร้างคำศัพท์ภาษาอังกฤษ

คำศัพท์ (ภาษาอังกฤษ)

คำอ่าน (Pronunciation)

ชนิดของคำ (Part of Speech)

- นาม (Noun)
- กริยา (Verb)
- คำกริยาวิเศษณ์ (Adverb)
- คำคุณศัพท์ (Adjective)
- บุพบท (Preposition)
- สันธาน (Conjunction)
- อทาน (Interjection)

ตัวอย่างการใช้ (Example)

คำสลับ (Synonym)

คำตรงข้าม (Antonym)

แสดงผลการสร้างคำศัพท์

คำแปล (ไทย)

คำแปล (ไทย)

คำแปล (ไทย)

คำแปล (ไทย)

คำแปล (ไทย)

คำแปล (ไทย)

คำแปล (ไทย)

คำแปล (ไทย)

ตัวอย่างการใช้ (Example)

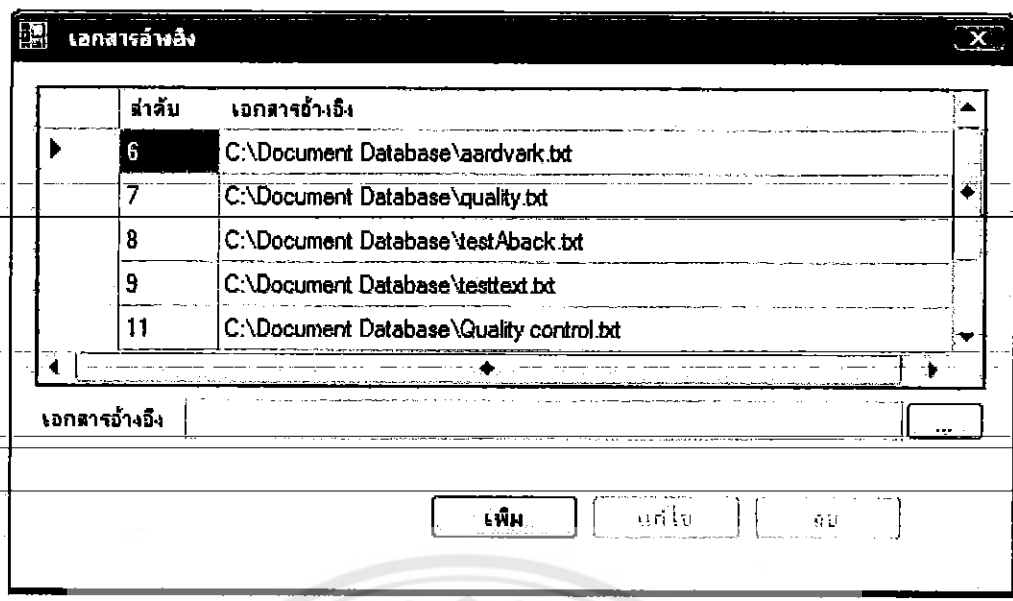
สร้างคำศัพท์

ลบคำศัพท์

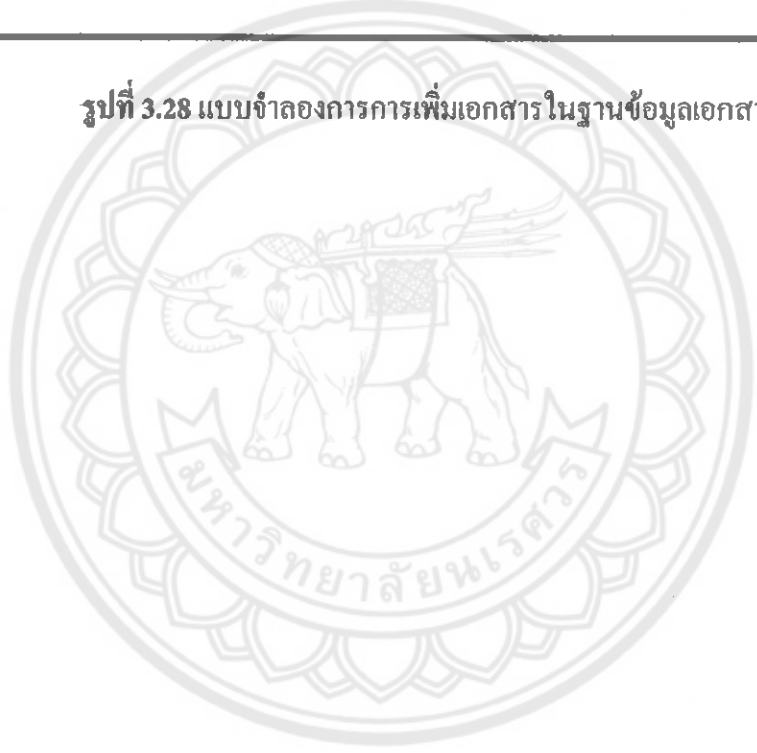
ยกเลิก

รูปที่ 3.26 แบบจำลองการสร้างคำศัพท์ในฐานข้อมูลพจนานุกรมหลัก





รูปที่ 3.28 แบบจำลองการเพิ่มเอกสารในฐานข้อมูลเอกสาร



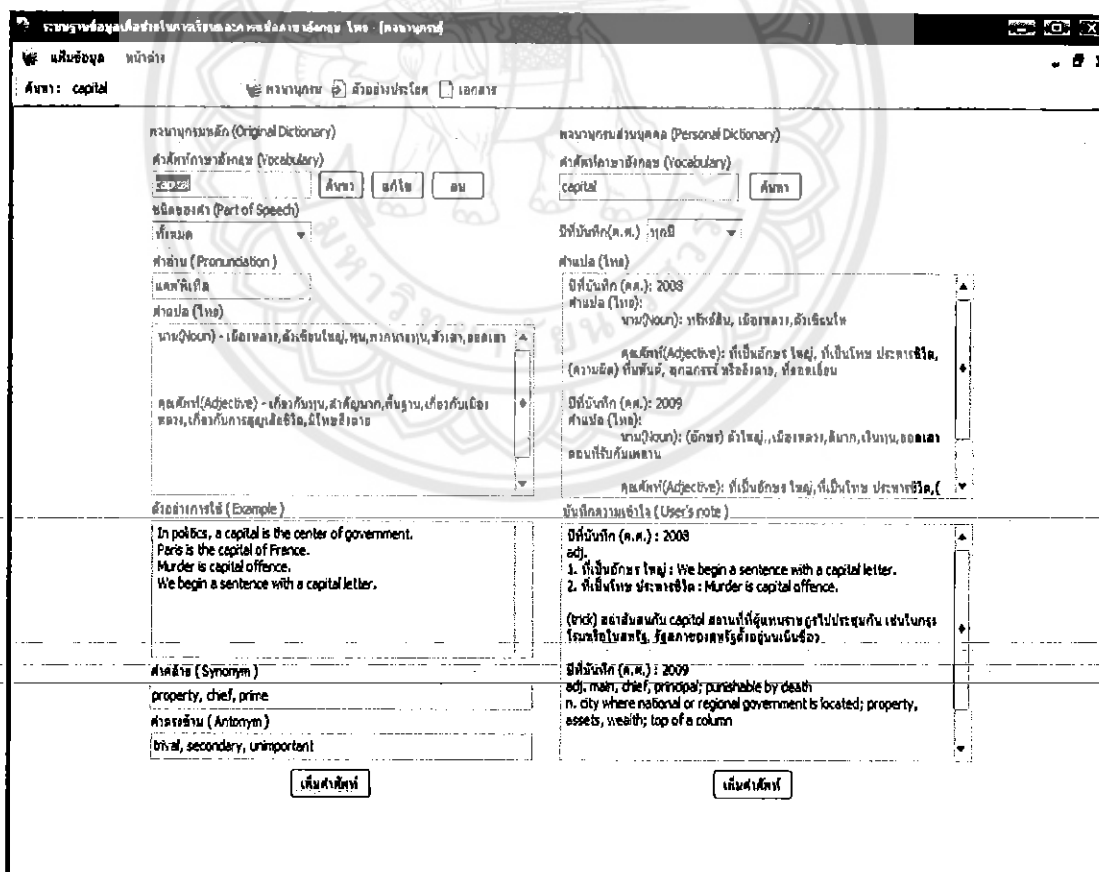
## บทที่ 4

### การทดลองและผลการทดลอง

จากบทที่ 3 ซึ่งได้ทำการออกแบบและเขียนโปรแกรม อันต่อไปจึงเป็นการทดลองการใช้งาน โดยบทที่ 4 นี้จะเป็นการแสดงผลการทดลอง อธิบายวิธีการใช้งาน โปรแกรมระบบฐานข้อมูล เพื่อช่วยในการเรียนและการแปลภาษาอังกฤษ-ไทย และการแก้ไขข้อผิดพลาดใน โปรแกรม ดังนี้

#### 4.1 โปรแกรมส่วนของการค้นหาคำศัพท์

การค้นหาคำศัพท์จะสามารถค้นหาได้โดยป้อนคำศัพท์ไว้ที่ช่องคำศัพท์ภาษาอังกฤษ (Vocabulary) จากหน้าต่างของโปรแกรมหลัก แล้วทำการค้นหาโดยกดปุ่ม Enter ที่เป็นพิมพ์ ก็จะสามารถค้นหาคำศัพท์จากทั้ง 3 ส่วนพร้อมกัน จากการค้นหาคำศัพท์ตัวอย่างคำว่า “capital” ดังรูปที่ 4.1



รูปที่ 4.1 โปรแกรมหลักสำหรับการค้นหาคำศัพท์

### การค้นหาคำศัพท์ได้ผลดังนี้

1. ฐานข้อมูลพจนานุกรมหลัก แสดงข้อมูลคำศัพท์ ได้แก่ คำอ่าน, ชนิดของคำ, ความหมายของคำศัพท์แต่ละชนิด, ตัวอย่างการใช้คำศัพท์, คำคล้าย และ คำตรงข้าม อีกทั้งยังสามารถเลือกค้นหาคำศัพท์โดยเลือกชนิดของคำศัพท์ ดังรูปที่ 4.2

#### พจนานุกรมหลัก (Original Dictionary)

##### คำศัพท์ภาษาอังกฤษ (Vocabulary)

capital

ค้นหา

แก้ไข

ลบ

##### ชนิดของคำ (Part of Speech)

ทั้งหมด

ทั้งหมด

นาม(Noun)

สรรพนาม(Pronoun)

กริยา(Verb)

กริยาวิเศษณ์(Adverb)

คุณศัพท์(Adjective)

บุพบท(Preposition)

สันธาน(Conjunction)

อุทาน(Interjection)

คุณศัพท์(Adjective) - เกษวาทย์, สำคัญมาก, พื้นฐาน, เกี่ยวกับเมือง

หลวง, เกี่ยวกับการสูญเสียชีวิต, มีโทษถึงตาย

##### ตัวอย่างการใช้ (Example)

In politics, a capital is the center of government.  
Paris is the capital of France.  
Murder is capital offence.  
We begin a sentence with a capital letter.

##### คำคล้าย (Synonym)

property, chief, prime

##### คำตรงข้าม (Antonym)

trivial, secondary, unimportant

เพิ่มคำศัพท์

### รูปที่ 4.2 การค้นหาคำศัพท์โดยเลือกชนิดของคำในพจนานุกรมหลัก

2. ฐานข้อมูลพจนานุกรมส่วนบุคคล แสดงข้อมูลคำศัพท์ ได้แก่ ปีที่บันทึก, ความหมายของคำศัพท์แต่ละชนิด และ บันทึกความเข้าใจของผู้ใช้ อีกทั้งยังสามารถเลือกค้นหาคำศัพท์โดยเลือกปีที่บันทึกคำศัพท์ ดังรูปที่ 4.3

พจนานุกรมส่วนบุคคล (Personal Dictionary)

คำศัพท์ภาษาอังกฤษ (Vocabulary)

capital

ค้นหา

ปีที่บันทึก(ค.ศ.)

ทุกปี

สำเนา (ไทย)

ทุกปี

ปีที่บันทึก (ค.ศ.): 2009

สำเนา (ไทย): 2008

นาม( 2007

นาม( 2006

ค.ศ. 2005

(ความผิด) ที่มที่ 2004

2003

เมืองหลวง, ลัวเขียนไท

เป็นอักษรใหญ่, ที่เป็นโทษ ประหารชีวิต,  
ถึงตาย, ที่ยอดเขียม

ปีที่บันทึก (ค.ศ.): 2009

สำเนา (ไทย):

นาม(Noun): (อักษร) ตัวใหญ่,,เมืองหลวง, ตีมาก, เงินทุน, ยอดเสา  
ตอนที่รับกับเพดาน

คุณศัพท์(Adjective): ที่เป็นอักษร ใหญ่, ที่เป็นโทษ ประหารชีวิต,(

บันทึกความเข้าใจ (User's note)

ปีที่บันทึก (ค.ศ.): 2008

adj.

1. ที่เป็นอักษร ใหญ่ : We begin a sentence with a capital letter.

2. ที่เป็นโทษ ประหารชีวิต : Murder is capital offence.

(trick) อย่าสับสนกับ capitol สถานที่ที่ผู้แทนราษฎรไปประชุมกัน เช่นในกรุง  
โรมหรือในสหรัฐ, รัฐสภาของสหรัฐตั้งอยู่บนเนินเขิว

ปีที่บันทึก (ค.ศ.): 2009

adj. main, chief, principal; punishable by death

n. city where national or regional government is located; property,  
assets, wealth; top of a column

เพิ่มคำศัพท์

## รูปที่ 4.3 การค้นหาคำศัพท์โดยเลือกปีที่บันทึกในพจนานุกรมส่วนบุคคล

3. ฐานข้อมูลเอกสาร แสดงตัวอย่างประโยคภาษาอังกฤษ และเอกสาร ต่างๆที่ถูกจัดเก็บไว้ใน ฐานข้อมูลเอกสาร เมื่อค้นพบคำศัพท์ที่ต้องการค้นหาในแต่ละประโยค จะนำประโยคนั้นๆมาแสดง และทำการเน้นคำศัพท์ให้เห็นได้ชัดเจน (Highlight) โดยมีรูปแบบ คือ “ คำศัพท์ ” ดังรูปที่ 4.4 อีก ทั้งยังแสดงเอกสารอ้างอิงของแต่ละประโยค ซึ่งเมื่อมีการเปิดเอกสารอ้างอิงใดๆ ก็จะได้แสดงไฟล์ เอกสารอ้างอิงนั้นขึ้นมา พร้อมทั้งทำการเน้นคำศัพท์ที่ค้นหาทุกคำในเอกสารนั้นด้วย ดังรูปที่ 4.5



ศัพท์ (ภาษาไทย)	จำนวนประโยค	รวม	คำศัพท์:	วลี/วลีอนุประโยค
capital	5		บริบทอื่น	วลี/วลีอนุประโยค
1	In economics a capital or capital goods or real capital refers to items of extensive value.			Capital (economics).txt
2	The capital goods may be acquired with money or financial capital.			Capital (economics).txt
3	In finance and accounting the capital generally refers to financial wealth, especially that used			Capital (economics).txt
4	The capital in narrow and broad uses In classical economics, capital is one of three (or four, in			Capital (economics).txt
5	There was the further clarification that capital is a stock.			Capital (economics).txt

### รูปที่ 4.4 การค้นหาตัวอย่างประโยคในฐานข้อมูลเอกสาร

**จัดการข้อมูลเอกสาร**

เอกสาร (Documents)

Capital (economics).txt

In economics a capital or capital goods or real capital refers to items of extensive value. The term can also be applied to the amount of wealth a person controls or is capable of controlling.  
 The capital goods may be acquired with money or financial capital. In finance and accounting the capital generally refers to financial wealth, especially that used to start or maintain a business, sometimes referred to as Cash flow.

**The capital in narrow and broad uses**  
 In classical economics, capital is one of three (or four, in some formulations) factors of production. The others are land, labour and (in some versions) organisation, entrepreneurship, or management. Goods with the following features are capital:

- It can be used in the production of other goods (this is what makes it a factor of production).
- It was produced, in contrast to "land," which refers to naturally occurring resources such as geographical locations and minerals.
- It is not used up immediately in the process of production unlike raw materials or intermediate goods. (The significant exception to this is depreciation allowance, which like intermediate goods, is treated as a business expense.)

These distinctions of convenience carried over to neoclassical economics with little change in formal analysis for an extended period. There was the further clarification that capital is a stock. As such, its value can be estimated at a point in time, say December 31. By contrast, investment, as production to be added to the capital stock, is described as taking place over time ("per year"), thus a flow.

Earlier illustrations often described capital as physical items, such as tools, buildings, and vehicles that are used in the production process. Since at least the 1960s economists have increasingly focused on broader forms of capital. For example, investment in skills and education can be viewed as building up human capital or knowledge capital, and investments in intellectual property can be viewed as building up intellectual capital. These terms lead to certain questions and controversies discussed in those articles.

Human development theory describes human capital as being composed of distinct social, imitative and creative elements:

- Social capital is the value of network trusting relationships between individuals in an economy.
- Individual capital which is inherent in persons, protected by societies, and trades labor for trust or money. Close parallel concepts are 'talent', 'ingenuity', 'leadership', 'trained bodies', or 'innate skills' that cannot reliably be reproduced by using any combination of any of the others above. In traditional economic analysis individual capital is more usually called labour.

Further classifications of capital that have been used in various theoretical or applied uses include:

- Financial capital which represents obligations; and is liquidated as money for trade; and owned by legal entities. It is in the form of capital assets; traded in financial markets. Its market value is not based on the historical accumulation of money invested but on the perception by the market of its expected revenues and of the risk entailed.
- Natural capital which is inherent in ecologies and protected by communities to support life, e.g. a river which provides farms with water.
- Infrastructural capital is non-natural support systems (e.g. clothing, shelter, roads, personal computers) that minimize need for new social trust, instruction, and natural resources. (Almost all of this is manufactured, leading to the older term manufactured capital, but some arises from interactions with natural capital, and so it makes more sense to describe it in terms of its appreciation/depreciation

### รูปที่ 4.5 การแสดงเอกสารอ้างอิงของตัวอย่างประโยค

## 4.2 โปรแกรมส่วนของการเพิ่มคำศัพท์

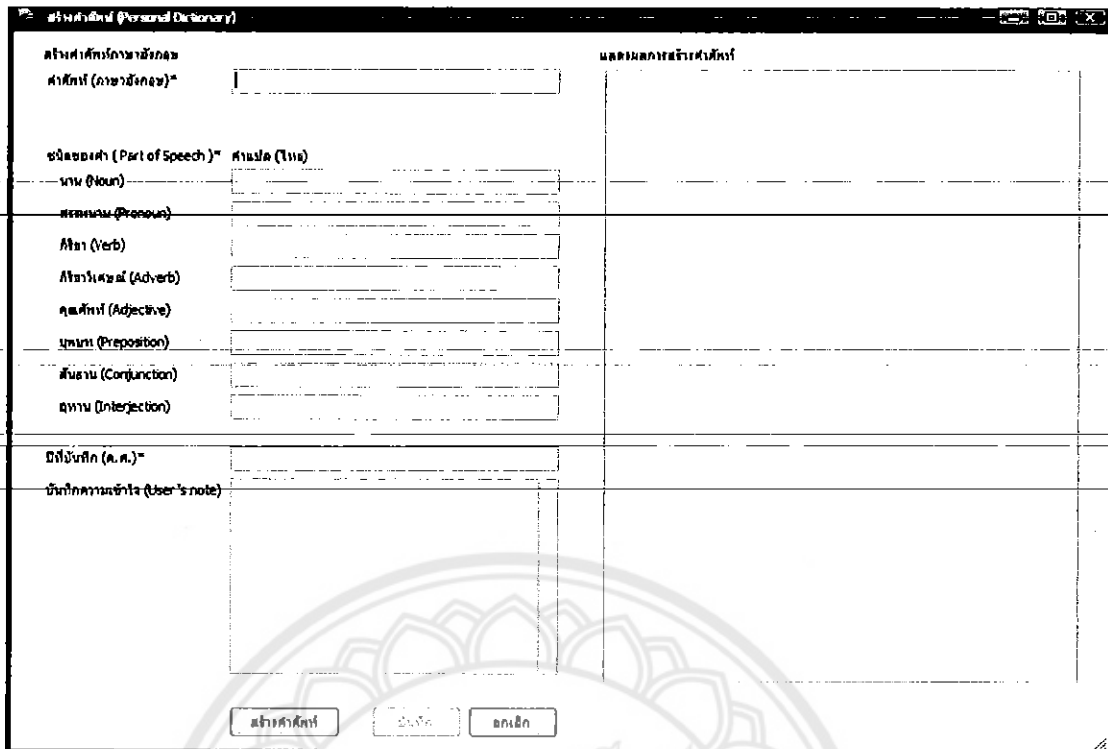
การเพิ่มคำศัพท์มีอยู่ 2 ส่วน คือ ฐานข้อมูลพจนานุกรมหลัก และ พจนานุกรมส่วนบุคคล

### 4.2.1 การเพิ่มคำศัพท์ในฐานข้อมูลพจนานุกรมหลัก

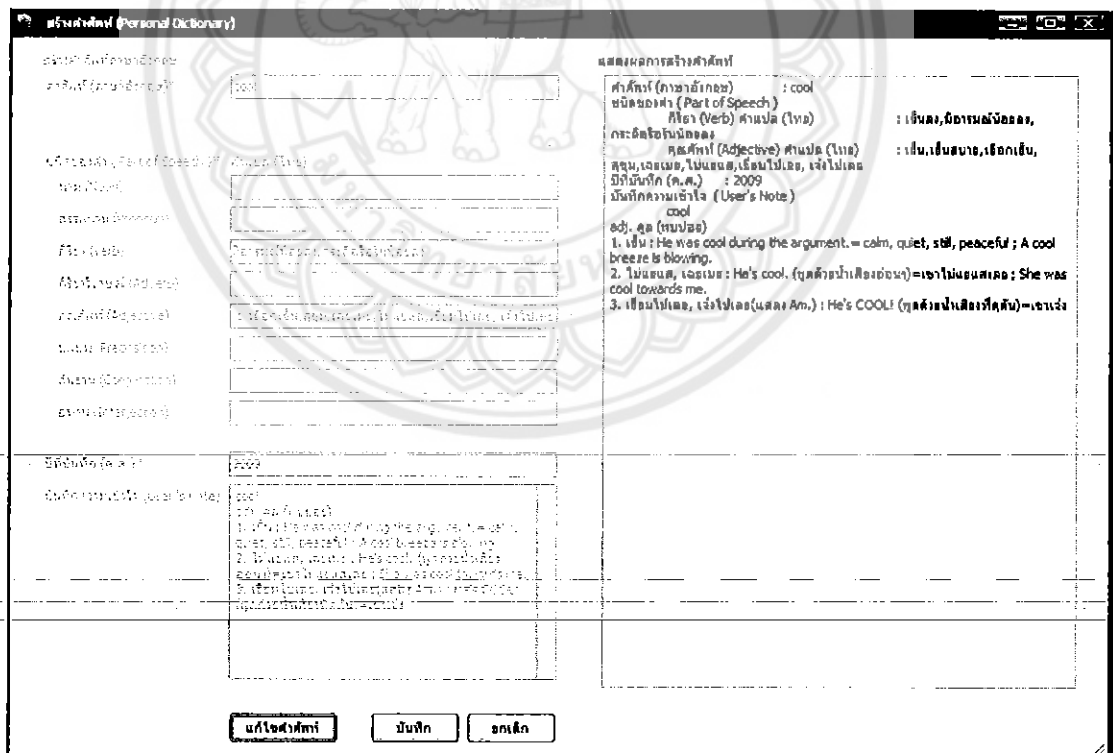
รูปที่ 4.6 โปรแกรมส่วนของการเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมหลัก

จากรูปที่ 4.6 เป็นแบบฟอร์มสำหรับเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมหลัก ต้องป้อนข้อมูลคำศัพท์ต่างๆลงไป ได้แก่ ชื่อคำศัพท์, คำอ่าน และความหมายของคำที่จำแนกตามชนิดของคำ ส่วนที่มีหรือไม่มีก็ได้ คือ ตัวอย่างการใช้, คำคล้าย และ คำตรงข้าม เมื่อป้อนข้อมูลเสร็จก็กดปุ่ม สร้างคำศัพท์ ซึ่งโปรแกรมจะแสดงข้อมูลทั้งหมดที่ป้อนลงไป เพื่อให้ตรวจสอบความถูกต้องของข้อมูลก่อนที่จะทำการบันทึกคำศัพท์ ตัวอย่างดังรูปที่ 4.7 ถ้าพบว่าป้อนข้อมูลคำศัพท์ผิดพลาดในการป้อนข้อมูลแล้วต้องการแก้ไขก็กดปุ่ม แก้ไขคำศัพท์ และเมื่อแน่ใจว่าข้อมูลคำศัพท์ถูกต้องแล้ว ก็สามารถบันทึกข้อมูลในฐานข้อมูลพจนานุกรมหลักโดยกดปุ่ม บันทึก





รูปที่ 4.8 โปรแกรมส่วนของการเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมส่วนบุคคล



รูปที่ 4.9 ตัวอย่างของการเพิ่มคำศัพท์ฐานข้อมูลพจนานุกรมส่วนบุคคล

### 4.3 โปรแกรมส่วนของการแก้ไขคำศัพท์

การแก้ไขคำศัพท์มีอยู่ 2 ส่วน คือ ฐานข้อมูลพจนานุกรมหลัก และ ฐานข้อมูลพจนานุกรมส่วนบุคคล

#### 4.3.1 การแก้ไขคำศัพท์ในฐานข้อมูลพจนานุกรมหลัก

เมื่อมีการค้นหาคำศัพท์แล้วพบข้อมูลคำศัพท์นั้น โปรแกรมจะยอมให้ผู้ใช้สามารถแก้ไขคำศัพท์นั้นได้ โดยจะมีปุ่มแก้ไขปรากฏขึ้นมา ถ้าหากต้องการแก้ไขก็กดปุ่ม แก้ไข ซึ่งจะแสดงหน้าต่างแบบฟอร์ม สร้างคำศัพท์ ขึ้นมาดังรูปที่ 4.10

ศัพท์คำศัพท์ (Original Dictionary)

สร้างคำศัพท์ภาษาอังกฤษ

คำศัพท์ (ภาษาอังกฤษ)\*

คำอ่าน (Pronunciation)\*

ชนิดของคำ (Part of Speech)\*

นาม (Noun)

สรรพนาม (Pronoun)

กิริยา (Verb)

กิริยาวิเศษณ์ (Adverb)

คุณศัพท์ (Adjective)

บุพบท (Preposition)

สันธาน (Conjunction)

อุทาน (Interjection)

วลีตัวอย่างใช้ (Example)\*

คำพ้อง (Synonym)

คำตรงข้าม (Antonym)

รูปที่ 4.10 โปรแกรมส่วนของการแก้ไขคำศัพท์ฐานข้อมูลพจนานุกรมหลัก

จากรูปที่ 4.10 เป็นแบบฟอร์มสร้างคำศัพท์ฐานข้อมูลพจนานุกรมหลัก ซึ่งจะนำข้อมูลของคำศัพท์ที่มีอยู่ในฐานข้อมูลออกมาแสดง เพื่อให้แก้ไขข้อมูลคำศัพท์ต่างๆที่พบความผิดพลาดได้สะดวก เมื่อแก้ไขเรียบร้อยแล้วก็กดปุ่ม สร้างคำศัพท์ โปรแกรมจะแสดงข้อมูลทั้งหมดที่ป้อนลงไป เพื่อให้ตรวจสอบความถูกต้องของข้อมูลอีกครั้งก่อนที่จะทำการบันทึกคำศัพท์

#### 4.3.2 การแก้ไขคำศัพท์ในฐานข้อมูลพจนานุกรมส่วนบุคคล

ในพจนานุกรมส่วนบุคคลผู้ใช้จะแก้ไขข้อมูลคำศัพท์ได้ก็ต่อเมื่อได้ทำการค้นหาคำศัพท์โดยจะต้องเลือกปีที่บันทึกคำศัพท์ที่ต้องการแก้ไข ดังรูปที่ 4.11 เมื่อต้องการแก้ไขก็กดปุ่ม แก้ไข ซึ่งจะแสดงแบบฟอร์มขึ้นมา ดังรูปที่ 4.12

## พจนานุกรมส่วนบุคคล (Personal Dictionary)

## คำศัพท์ภาษาอังกฤษ (Vocabulary)

capital

ค้นหา

แก้ไข

ลบ

ปีที่บันทึก(ค.ศ.) 2008

## คำแปล (ไทย)

ปีที่บันทึก(ค.ศ.): 2008

คำแปล (ไทย):

นาม(Noun): ทรัสต์สิน, เมืองหลวง, ตัวเขียนใหญ่

คุณศัพท์(Adjective): ที่เป็นอักษรใหญ่, ที่เป็นโทษ ประหารชีวิต, (ความคิด) ที่มึนตึ๋น, ฤกษ์งูหรือหรือถึงคาบ, ฤกษ์งูเขียน

## บันทึกความเข้าใจ (User's note)

ปีที่บันทึก(ค.ศ.): 2008

adj.

1. ที่เป็นอักษรใหญ่ : We begin a sentence with a capital letter.

2. ที่เป็นโทษ ประหารชีวิต : Murder is capital offence.

( trick ) อย่างสัมพันธ์กับ capital สถานที่ที่ผู้แทนราษฎรไปประชุมกัน เช่นในกรุงโรมหรือในสหรัฐ, รัฐสภาของสหรัฐตั้งอยู่บนเนินเขาว

เพิ่มคำศัพท์

รูปที่ 4.11 การเลือกปีที่บันทึกเพื่อทำการแก้ไขคำศัพท์ในฐานข้อมูลพจนานุกรมส่วนบุคคล

พจนานุกรมส่วนบุคคล (Personal Dictionary)	
คลังคำศัพท์ภาษาอังกฤษ คำศัพท์ (ภาษาอังกฤษ)*	capital
ชนิดของคำ (Part of Speech)**	คำแปล (ไทย)
นาม (Noun)	ทรัสต์สิน, เมืองหลวง, ตัวเขียนใหญ่
สรรพนาม (Pronoun)	
กริยา (Verb)	
กริยาวิเศษณ์ (Adverb)	
คุณศัพท์ (Adjective)	ที่เป็นอักษรใหญ่, ที่เป็นโทษ ประหารชีวิต, (ความคิด) ที่มึนตึ๋น
บุพบท (Preposition)	
สันธาน (Conjunction)	
อุทาน (Interjection)	
ปีที่บันทึก(ค.ศ.)*	2008
บันทึกความเข้าใจ (User's note)	adj. 1. ที่เป็นอักษรใหญ่ : We begin a sentence with a capital letter. 2. ที่เป็นโทษ ประหารชีวิต : Murder is capital offence.  ( trick ) อย่างสัมพันธ์กับ capital สถานที่ที่ผู้แทนราษฎรไปประชุมกัน เช่นในกรุงโรมหรือในสหรัฐ, รัฐสภาของสหรัฐตั้งอยู่บนเนินเขาว
<div style="text-align: right;"> <input type="button" value="ลบคำศัพท์"/> <input type="button" value="แก้ไข"/> <input type="button" value="บันทึก"/> </div>	

รูปที่ 4.12 โปรแกรมส่วนของการแก้ไขคำศัพท์ฐานข้อมูลพจนานุกรมส่วนบุคคล

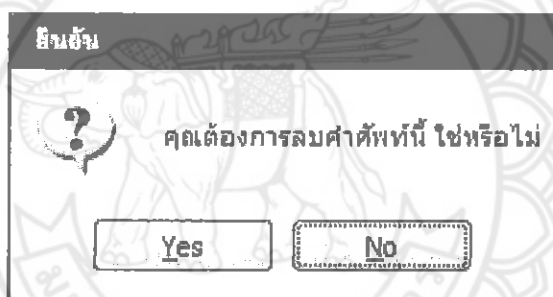
จากรูปที่ 4.12 เป็นแบบฟอร์มสร้างคำศัพท์ฐานข้อมูลพจนานุกรมส่วนบุคคล ซึ่งจะนำข้อมูลของคำศัพท์ที่มีอยู่ในฐานข้อมูลออกมาแสดง เพื่อให้แก้ไขข้อมูลคำศัพท์ต่างๆที่พบความผิดพลาดได้สะดวก เมื่อแก้ไขเรียบร้อยแล้วก็กดปุ่ม สร้างคำศัพท์ โปรแกรมจะแสดงข้อมูลทั้งหมดที่ป้อนลงไป เพื่อให้ตรวจสอบความถูกต้องของข้อมูลอีกครั้งก่อนที่จะทำการบันทึกคำศัพท์

#### 4.4 โปรแกรมส่วนของการลบคำศัพท์

การลบคำศัพท์มีอยู่ 2 ส่วน คือ ฐานข้อมูลพจนานุกรมหลัก และ พจนานุกรมส่วนบุคคล ดังต่อไปนี้

##### 4.4.1 การลบคำศัพท์ในฐานข้อมูลพจนานุกรมหลัก

เมื่อมีการค้นหาคำศัพท์แล้วพบข้อมูลคำศัพท์นั้น โปรแกรมจะขอมให้ผู้ใช้สามารถลบคำศัพท์นั้นได้ โดยจะมีปุ่มลบปรากฏขึ้นมา ถ้าหากต้องการลบก็กดปุ่ม ลบ ซึ่งจะแสดงข้อความเพื่อให้ผู้ใช้ยืนยันการลบคำศัพท์นั้น ดังรูปที่ 4.13



รูปที่ 4.13 การยืนยันการลบ

จากรูปที่ 4.13 เมื่อกดปุ่มลบ จะแสดงข้อความยืนยันการลบ หากต้องการลบให้กดปุ่ม Yes ถ้าไม่ต้องการลบกดปุ่ม No และเมื่อกดปุ่ม Yes จะแสดงดังรูปที่ 4.14 คำศัพท์นั้นก็จะถูกลบออกจากฐานข้อมูลพจนานุกรมหลักทันที

ลบคำศัพท์ ✕



ลบข้อมูลเรียบร้อยแล้ว

OK

รูปที่ 4.14 แสดงผลโปรแกรมได้ลบคำศัพท์เรียบร้อยแล้ว

#### 4.4.2 การลบคำศัพท์ในฐานะข้อมูลพจนานุกรมส่วนบุคคล

ในพจนานุกรมส่วนบุคคลผู้ใช้จะแก้ไขข้อมูลคำศัพท์ได้ก็ต่อเมื่อได้ทำการค้นหาคำศัพท์โดยจะต้องเลือกปีที่บันทึกคำศัพท์ที่ต้องการลบเช่นเดียวกับการแก้ไขคำศัพท์ ดังรูปที่ 4.11 เมื่อต้องการลบคำศัพท์ก็กดปุ่ม ลบ ซึ่งจะแสดงข้อความเพื่อให้ผู้ใช้ยืนยันการลบคำศัพท์นั้น ดังรูปที่ 4.13 และจะแสดงผลการลบดังรูปที่ 4.14 เช่นเดียวกันกับการลบคำศัพท์ในฐานะข้อมูลพจนานุกรมหลัก

#### 4.5 โปรแกรมส่วนของฐานข้อมูลเอกสาร

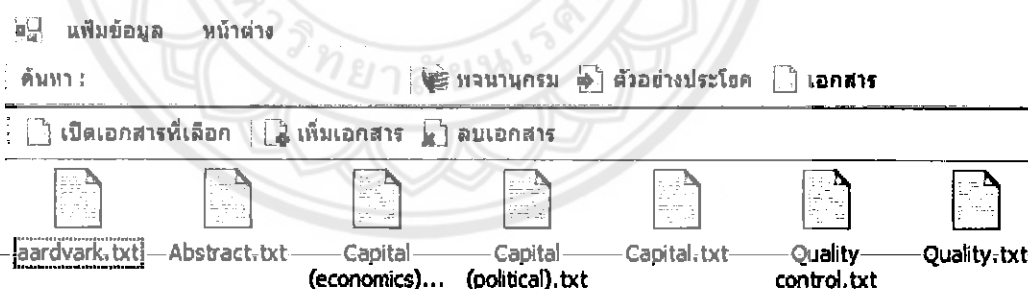
ฐานข้อมูลเอกสารใช้เก็บเอกสาร เพื่อนำมาเป็นตัวอย่างประโยคภาษาอังกฤษ ดังนั้นจำเป็นต้องมีส่วนของการเพิ่มและลบ เอกสาร อีกทั้งยังมีส่วนแสดงตัวอย่างประโยคพร้อม

##### 4.5.1 การเพิ่มเอกสาร มีขั้นตอนดังต่อไปนี้

1. ไปที่เมนู เพิ่มข้อมูล แล้วเลือก เพิ่มข้อมูล จากนั้นไปที่ เอกสาร ดังรูปที่ 4.15



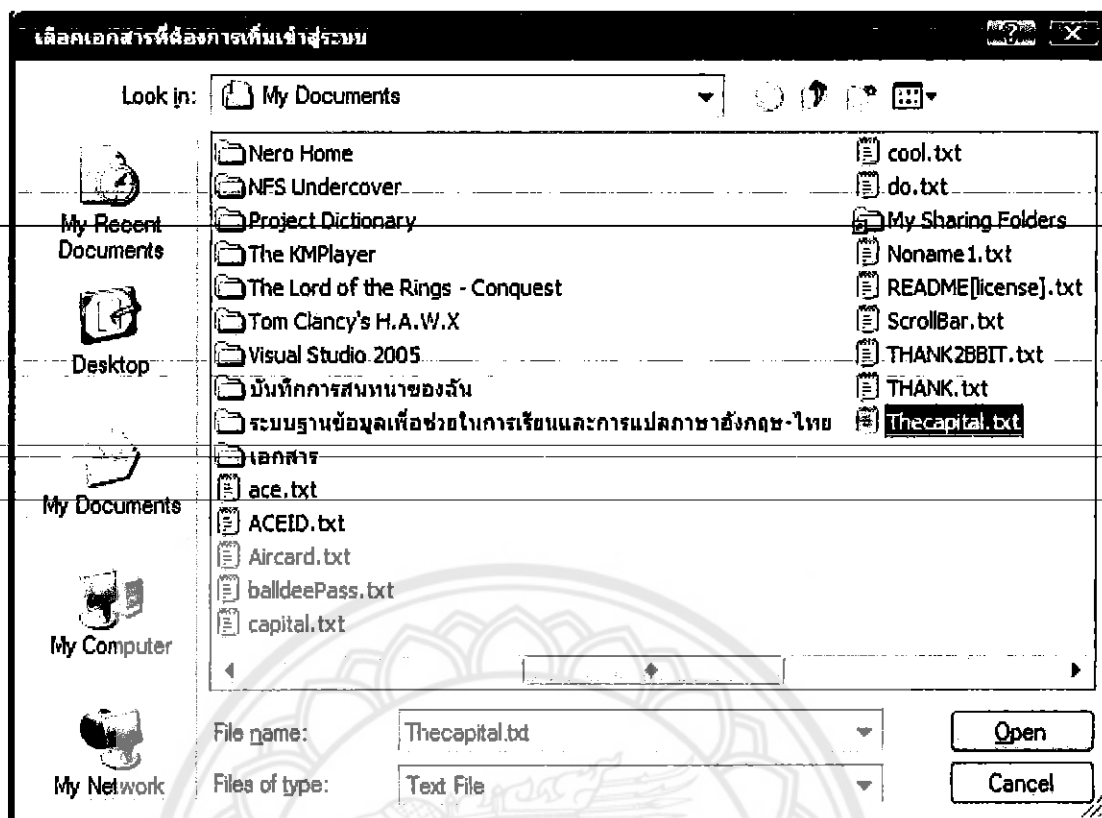
รูปที่ 4.15 เมนูการเพิ่มเอกสาร



รูปที่ 4.16 แบบฟอร์มการเพิ่มเอกสาร

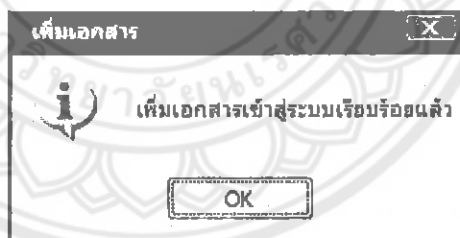
2. จากรูปที่ 4.16 เป็นหน้าต่างจัดการเอกสาร เมื่อต้องการเพิ่มเอกสาร ให้กดปุ่ม เพิ่มเอกสาร โปรแกรมจะเรียกหน้าต่าง "เลือกเอกสารที่ต้องการเพิ่มเข้าสู่ระบบ" ขึ้นมา เพื่อเลือกไฟล์เอกสารที่ต้องการ เมื่อเลือกไฟล์ที่ต้องการเพิ่มได้แล้วก็กดปุ่ม Open ดังรูปที่ 4.17





รูปที่ 4.17 การเลือกไฟล์เอกสารที่ต้องการเพิ่มลงในฐานข้อมูล

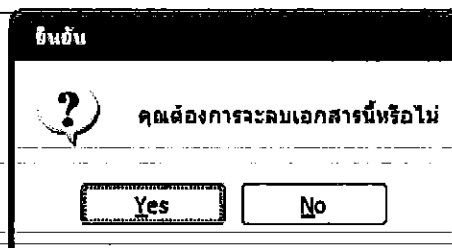
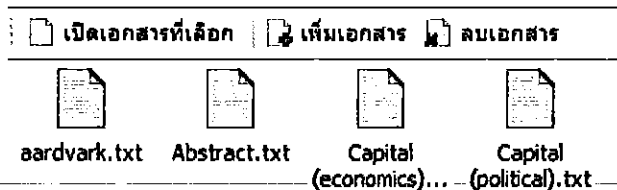
3. หลังจากเพิ่มข้อมูลเอกสารไปแล้ว จะมีข้อความแสดงผลการเพิ่มเอกสาร ดังรูปที่ 4.18



รูปที่ 4.18 แสดงผลการเพิ่มเอกสารเมื่อเอกสารเข้าสู่ระบบ

#### 4.5.2 การลบเอกสาร

จากรูปที่ 4.16 เป็นหน้าต่างจัดการเอกสาร เมื่อต้องการลบเอกสาร ให้เลือกไฟล์เอกสาร แล้วกดปุ่ม ลบ โปรแกรมก็จะแสดงปุ่มยืนยันเพื่อลบเอกสาร ดังรูปที่ 4.19



รูปที่ 4.19 การลบเอกสาร

#### 4.5.3 การเปิดเอกสาร

จากรูปที่ 4.16 เป็นหน้าต่างจัดการเอกสาร เมื่อต้องการเปิดเอกสารจากไฟล์ในฐานข้อมูล ให้เลือกไฟล์เอกสาร แล้วคลิกปุ่ม เปิดเอกสารที่เลือก โปรแกรมก็จะแสดงเอกสารขึ้นมา เอกสารที่แสดงขึ้นมาสามารถแก้ไขและบันทึกได้

#### 4.5.4 การแสดงตัวอย่างประโยค

ในการแสดงตัวอย่างประโยค จะต้องป้อนคำศัพท์ภาษาอังกฤษและกำหนดจำนวนประโยคที่ต้องการจะแสดง แล้วคลิกปุ่ม ค้นหา ดังรูปที่ 4.20

คำศัพท์ (ภาษาอังกฤษ)	จำนวนประโยค	ค้นหา
capital	5	ค้นหา

รูปที่ 4.20 การค้นหาตัวอย่างประโยค

การแสดงผลตัวอย่างประโยคนั้นมีสองรูปแบบคือ จัดชิดซ้ายและจัดตามตำแหน่งคำศัพท์ ซึ่งการจัดชิดซ้าย ตัวอย่างประโยคทุกประโยคจะอยู่ชิดซ้าย ดังรูปที่ 4.21 และอีกแบบคือการจัดเรียงตามคำศัพท์ ดังรูปที่ 4.22

คำศัพท์ (ภาษาอังกฤษ)	จำนวนประโยค	ค้นหา	จัดเรียงประโยค:	จัดเรียงตามคำศัพท์
capital	5	ค้นหา	จัดเรียงประโยค:	จัดเรียงตามคำศัพท์
1	In economics a capital or capital goods or real capital refers to items of extensive value.			Capital (economics).txt
2	The capital goods may be acquired with money or financial capital.			Capital (economics).txt
3	In finance and accounting the capital generally refers to financial wealth, especially that used			Capital (economics).txt
4	The capital in narrow and broad usesIn classical economics, capital is one of three (or four, in			Capital (economics).txt
5	There was the further clarification that capital is a stock.			Capital (economics).txt

#### รูปที่ 4.21 การแสดงตัวอย่างประโยค โดยจัดชิดซ้าย

คำศัพท์ (ภาษาอังกฤษ)	จำนวนประโยค	ค้นหา	จัดเรียงประโยค:	จัดเรียงตามคำศัพท์
capital	5	ค้นหา	จัดเรียงประโยค:	จัดเรียงตามคำศัพท์
1	In economics a capital or capital goods or real capital refers to items			Capital (economics).txt
2	The capital goods may be acquired with money or financial or			Capital (economics).txt
3	In finance and accounting the capital generally refers to financial wealth, especially;			Capital (economics).txt
4	The capital in narrow and broad usesIn classical economics,			Capital (economics).txt
5	There was the further clarification that capital is a stock.			Capital (economics).txt

#### รูปที่ 4.22 การแสดงตัวอย่างประโยค โดยจัดตามตำแหน่งคำศัพท์

จากรูปที่ 4.22 เป็นการแสดงตัวอย่างประโยค โดยจัดเรียงตามคำศัพท์ที่ค้นหา ระบบจะทำการจัดตำแหน่งคำศัพท์ที่ค้นหาคำแรกของประโยคแต่ละประโยคให้มีตำแหน่งตรงกันทุกประโยค เพื่อให้สังเกตลักษณะการใช้คำศัพท์ภาษาอังกฤษได้ง่าย

#### 4.6 ผลการทดลอง

โปรแกรมระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปล ภาษาอังกฤษ-ไทยที่พัฒนาขึ้น ได้ผลการทำงานของ โปรแกรมดังตารางต่อไปนี้

ตารางที่ 4.1 ตารางแสดงผลการทดลอง

ระบบ	ผลการทดลอง
ฐานข้อมูลพจนานุกรมหลัก (Original dictionary)	<ol style="list-style-type: none"> <li>1. สามารถค้นหาคำศัพท์โดยเลือกชนิดของคำได้</li> <li>2. สามารถแสดงข้อมูลคำศัพท์ ได้แก่ การอ่านออกเสียง, ความหมายของคำศัพท์โดยจำแนกตามชนิดของคำ, ตัวอย่างการใช้คำศัพท์, คำคล้าย และคำตรงข้ามได้</li> <li>3. สามารถเพิ่มหรือสร้างคำศัพท์ได้โดยที่คำศัพท์ไม่ซ้ำซ้อนกัน</li> <li>4. สามารถแก้ไขคำศัพท์ได้</li> <li>5. สามารถลบคำศัพท์ได้</li> </ol>
ฐานข้อมูลพจนานุกรมส่วนบุคคล (Personal dictionary)	<ol style="list-style-type: none"> <li>1. สามารถค้นหาคำศัพท์โดยเลือกปีที่บ้านทีกได้</li> <li>2. สามารถแสดงข้อมูลคำศัพท์ ได้แก่ ความหมายของคำศัพท์โดยจำแนกตามชนิดของคำ, ปีที่บ้านทีก และบันทึกความเข้าใจของผู้ใช้ได้</li> <li>3. สามารถเพิ่มหรือสร้างคำศัพท์ซ้ำกันโดยที่ ปีที่บ้านทีกไม่ซ้ำกันได้</li> <li>4. สามารถแก้ไขคำศัพท์ตามปีที่บ้านทีกได้</li> <li>5. สามารถลบคำศัพท์ตามปีที่บ้านทีกได้</li> </ol>
ฐานข้อมูลเอกสาร (Document database)	<ol style="list-style-type: none"> <li>1. สามารถค้นหาตัวอย่างประโยคจากเอกสารได้</li> <li>2. สามารถจัดเรียงตัวอย่างประโยคให้ตำแหน่งของคำศัพท์ที่ค้นหาตรงกันทุกประโยคได้</li> <li>3. สามารถเพิ่มเอกสารเพื่อใช้เป็นตัวอย่างประโยคได้</li> <li>4. สามารถลบเอกสารออกจากฐานข้อมูลได้</li> <li>5. สามารถดูเอกสารอ้างอิงของตัวอย่างประโยคได้</li> </ol>

## บทที่ 5

### สรุปผลการทดลอง

โปรแกรมระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปล ภาษาอังกฤษ-ไทย มีระบบอำนวยความสะดวกในการค้นหาคำศัพท์ ผู้ใช้ค้นหาคำศัพท์ในครั้งเดียวก็สามารถทราบข้อมูลคำศัพท์ในพจนานุกรมหลักและพจนานุกรมส่วนบุคคล อีกทั้งยังมีตัวอย่างประโยค ซึ่งตัวอย่างประโยคมีความสำคัญมากในพจนานุกรม ทั้งนี้ก็เพื่อให้ผู้ใช้สามารถเรียนรู้การใช้คำศัพท์ด้วยตัวเองได้ โดยมีตัวอย่างประโยคจำนวนมาก ผู้ใช้จะเห็นว่าใช้คำศัพท์ในประโยคนั้นอย่างไรและมีการเชื่อมกับคำอื่นอย่างไรให้ถูกต้องตามหลักไวยากรณ์ และเมื่อผู้ใช้ศึกษาคำศัพท์เข้าใจแล้วก็สามารถบันทึกไว้ได้เพื่อเป็นประโยชน์ต่อผู้ใช้ ซึ่งยังสามารถเรียนรู้ความเปลี่ยนแปลงทางภาษาได้ เนื่องจากมีการระบุปีที่บันทึก

ผู้จัดทำโครงการใช้โปรแกรม Visual Studio 2005 ในการพัฒนาโปรแกรม โดยเลือกใช้ภาษา Visual Basic.Net ในการออกแบบและเขียน โปรแกรม เนื่องจากจุดเด่นที่สำคัญของ Visual Basic.Net อยู่ที่ เป็นภาษาเชิงวัตถุที่ช่วยให้สามารถพัฒนาแอปพลิเคชันขนาดใหญ่และซับซ้อน ได้สะดวก อีกทั้งยังสร้างฐานข้อมูลด้วยโปรแกรม Microsoft Office Access 2003 ในการจัดการฐานข้อมูลเนื่องจาก เทคโนโลยี ADO.NET ของ Visual Basic.Net สามารถติดต่อและจัดการกับฐานข้อมูลที่สร้างจาก Microsoft Office Access 2003 ได้ง่ายและสามารถประมวลผลข้อมูลที่มีปริมาณมากได้อย่างมีประสิทธิภาพ

#### 5.1 สรุปผลการทดลอง

จากการทดลองใช้โปรแกรมระบบฐานข้อมูลเพื่อช่วยในการเรียนและการแปล ภาษาอังกฤษ-ไทย โปรแกรมสามารถแสดงผลได้เป็นที่น่าพอใจสามารถทำได้ตามที่ได้กำหนดความต้องการไว้สรุปผลได้ดังตารางต่อไปนี้

ตารางที่ 5.1 ตารางสรุปผลการทดลอง

ขอบเขต	ได้	ไม่ได้	หมายเหตุ
1. สร้าง Search Engine ที่มีการติดต่อผู้ใช้ผ่าน GUI (Graphic User Interface)	/		
2. สร้างฐานข้อมูลสำหรับเก็บข้อมูลคำศัพท์แบ่งเป็น 2 ส่วน คือฐานข้อมูลพจนานุกรมหลัก และพจนานุกรมส่วนบุคคล	/		

ตารางที่ 5.1(ต่อ)

ขอบเขต	ได้	ไม่ได้	หมายเหตุ
3. สร้างฐานข้อมูลเอกสารสำหรับเก็บตัวอย่างประโยคภาษาอังกฤษที่อยู่ในไฟล์เอกสารต่างๆ	/		ไฟล์เอกสาร สกุลไฟล์ .txt
4. สามารถค้นหาข้อมูลของคำศัพท์ ในส่วน ของฐานข้อมูลพจนานุกรมหลัก และ พจนานุกรมส่วนบุคคล	/		
5. สามารถค้นหาตัวอย่างประโยคของคำศัพท์ ภาษาอังกฤษจากไฟล์เอกสาร	/		
6. สามารถเพิ่ม แก้ไข และลบ ข้อมูลคำศัพท์ ภาษาอังกฤษทั้งในส่วนของฐานข้อมูล พจนานุกรมหลักและฐานข้อมูลพจนานุกรม ส่วนบุคคล	/		การแก้ไขคำศัพท์ ส่วนบุคคลต้องเลือก ค้นหา ตามปีที่บันทึก
7. สามารถเพิ่มและลบ เอกสารในส่วนของ ฐานข้อมูลเอกสาร	/		
8. สามารถเปิดดูเอกสารอ้างอิงของแต่ละ ประโยค	/		ต้องค้นหาคำศัพท์ ก่อน จึง จะ พบ ตัวอย่างประโยค
9. สามารถจัดตำแหน่งของคำศัพท์ที่ค้นหาใน ตัวอย่างประโยคแต่ละประโยคให้ตรงกันได้	/		

## 5.2 จุดเด่น จุดด้อย และข้อจำกัดของโครงการที่ทำขึ้น

## จุดเด่นคือ

1. ได้ทราบข้อมูลคำศัพท์รวดเร็วเหมาะสมกับความต้องการ
2. ค้นหาคำศัพท์เพียงครั้งเดียว จะได้ข้อมูลจากทั้ง 3 ระบบ คือ ระบบฐานข้อมูลพจนานุกรมหลัก, ระบบฐานข้อมูลพจนานุกรมส่วนบุคคล และระบบฐานข้อมูลเอกสาร
3. เพิ่มความสะดวกในการเรียนรู้คำศัพท์ภาษาอังกฤษ
4. เพิ่มข้อมูลคำศัพท์ได้ในปริมาณมาก
5. ได้ศึกษาความเปลี่ยนแปลงของคำศัพท์

## เอกสารอ้างอิง

- [1] สุรสิทธิ์ คิวประสพศักดิ์, นันทนีแขวงโสภา. อินไซต์ Visual Basic.NET. กรุงเทพมหานคร : บริษัทโปรวิชั่น จำกัด. 2537.
- [2] โอภาส เขียมสิริวงศ์. การวิเคราะห์และออกแบบระบบ. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด. 2549.
- [3] ดร.วิทย์ เทียงนุรณเบรรณ. พจนานุกรมอังกฤษ-ไทย. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด. 2541.
- [4] กิตติ ภักดีวัฒนะกุล, จำลอง ครัวอุตสาหกรรม. ระบบฐานข้อมูล (Database System). พิมพ์ครั้งที่ 9. กรุงเทพมหานคร : บริษัท เกทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด. 2550.
- [5] รัชชัย จำลอง. มือใหม่ Access 2003 ใช้งานอย่างมืออาชีพ. กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด. 2549.



## ประวัติผู้เขียนโครงการ



ชื่อ นายชนากร อุปละ  
 ภูมิลำเนา บ้านเลขที่ 22 หมู่ 4 ต.เมืองพาน อ.พาน จ.เชียงราย  
 57120

## ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนปลายจาก โรงเรียนพานพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์  
 มหาวิทยาลัยนเรศวร

อีเมลล์ cojiro\_tanako@hotmail.com



ชื่อ นายอธิปิติ สุขสันต์  
 ภูมิลำเนา บ้านเลขที่ 54 หมู่ 10 ต.เชียงบาน อ.เชียงคำ จ.พะเยา  
 56110

## ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนปลายจาก โรงเรียนเชียงคำวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์  
 มหาวิทยาลัยนเรศวร

อีเมลล์ modcomx@yahoo.com



ชื่อ นายเอกชัย ทาชมภู  
 ภูมิลำเนา บ้านเลขที่ 12 หมู่ 3 ต.ห้วยไคร้ อ.แม่สาย  
 จ.เชียงราย 57220

## ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนปลายจาก โรงเรียนแม่จันวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์  
 มหาวิทยาลัยนเรศวร

อีเมลล์ hextor\_at@hotmail.com