

การ인터เฟสด้วยเสียงเพื่อการสอบถามข้อมูลจากฐานข้อมูล

A SPEECH INTERFACE FOR QUERYING DATABASES



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

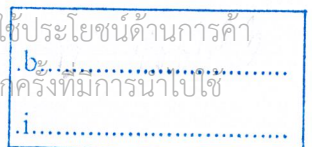
ปีการศึกษา 2546

เลขหมู่.....

เลขทะเบียน 51779.....

วัน,เดือน,ปี 29 ก.ค. 2547.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# A SPEECH INTERFACE FOR QUERYING DATABASES



**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF BECHELOR OF SCIENCE  
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE  
FACULTY OF SCIENCE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2003**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ      การอินเตอร์เฟสด้วยเสียงเพื่อการสอบถามข้อมูลจากฐานข้อมูล  
 A SPEECH INTERFACE FOR QUERYING DATABASES

ชื่อนักศึกษา      นายนเรนทร บำรุงกิจ      43050379  
                          นายภาคิน พัฒนพูนทอง      43050397  
                          นางสาวชนกานต์ แซ่ฮึ้ง      43050403

ภาควิชา      คณิตศาสตร์และวิทยาการคอมพิวเตอร์  
 สาขาวิชา      วิทยาการคอมพิวเตอร์  
 อาจารย์ที่ปรึกษา      ผศ.ดร.วีระ บุญจริง

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระ  
 จอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลัก  
 สูตรวิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2546

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ดร.นันทิกา เบนญะเทพานันท์	
กรรมการ	รศ.ธีรวัฒน์ ประกอบผล	
กรรมการและอาจารย์ที่ปรึกษา	ผศ.ดร.วีระ บุญจริง	

(ผู้ช่วยศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การอินเทอร์เน็ตด้วยเสียงเพื่อการสอบถามข้อมูลจากฐานข้อมูล	
ชื่อนักศึกษา	นายนเรนทร บำรุงกิจ	43050379
	นายภาทิน พัฒนพูนทอง	43050397
	นางสาวชนกานต์ แซ่อึ้ง	43050403
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	คณะวิทยาศาสตร์
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2546	
อาจารย์ที่ปรึกษา	ผศ.ดร.วีระ บุญจริง	

### บทคัดย่อ

โครงการนี้เป็นความพยายามหนึ่งในการพัฒนาระบบการสอบถามข้อมูลจากฐานข้อมูลด้วยเสียง ความพยายามนี้ใช้กรรมวิธีการพัฒนาระบบด้วยคอมพิวเตอร์ ดังนั้น งานหลักของโครงการนี้ได้แก่ การค้นหาคอมพิวเตอร์ที่รู้จักเสียงที่เหมาะสมแล้วทำการเชื่อมต่อคอมพิวเตอร์ที่เข้ากับคอมพิวเตอร์การสืบค้นข้อมูลจากฐานข้อมูลด้วยข้อความ ระบบการสอบถามข้อมูลด้วยเสียงที่พัฒนาขึ้นมีความสามารถตอบคำถามการสอบถามข้อมูลด้วยเสียงได้อย่างถูกต้อง โดยที่คำถามนั้นต้องเป็นคำถามที่เป็นไปตามไวยากรณ์ที่กำหนดไว้ล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Special Project Title</b>	A SPEECH INTERFACE FOR QUERYING DATABASES		
<b>Students</b>	Mr.Narentorn Bumrunakit	43050379	
	Mr.Pakin Pattanapontong	43050397	
	Miss.Chanakan Sae-Ung	43050403	
<b>Degree</b>	Bachelor of Science		
<b>Department</b>	Mathematics and Computer Science, Faculty of Science		
<b>Programme</b>	Computer Science		
<b>Academic Year</b>	2003		
<b>Special Project Advisor</b>	Assoc.Prof. Dr.Veera Boonjing		

### ABSTRACT

The project is an attempt to develop a speech interface for querying databases. This attempt is based on component-based software development methodology. Therefore, the major tasks of the project are searching for an appropriate speech recognition component and connecting this component to text database querying component. The developed speech interface for querying databases is capable of giving a correct answer to speech queries with limited predefined grammars.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการอินเทอร์เน็ตด้วยเสียงเพื่อการสอบถามข้อมูลจากฐานข้อมูล สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ผศ.ดร.วิระ บุญจริง อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการทำปัญหาพิเศษ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้ และชี้แนะการทำงานตลอดการทำปัญหาพิเศษนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณบิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจ และทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี รวมทั้งพี่ ๆ และเพื่อน ๆ ที่ได้ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ  
มีนาคม 2547



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของการรู้จำเสียง.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตของการศึกษา.....	1
1.4 ขั้นตอนการศึกษา.....	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
<b>บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....</b>	<b>3</b>
2.1 การพัฒนาระบบด้วยคอมพิวเตอร์.....	3
2.2 คอมพิวเตอร์สำเร็จรูป.....	4
2.3 คอมพิวเตอร์รู้จำเสียงสำเร็จรูป.....	6
<b>บทที่ 3 ระบบการสอบถามข้อมูลจากฐานข้อมูลด้วยเสียงพูด.....</b>	<b>8</b>
3.1 ส่วนประกอบของระบบ.....	8
3.2 คอมพิวเตอร์รู้จำเสียง(SAPI).....	11
3.3 ฐานข้อมูล(Database).....	14
3.4 ส่วนติดต่อกับผู้ใช้(User Interface).....	17
<b>บทที่ 4 การพัฒนาระบบ.....</b>	<b>21</b>
4.1 เครื่องมือที่ใช้ในการพัฒนา.....	21
4.2 การดำเนินงานของระบบ.....	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปผลและข้อเสนอแนะ.....	33
5.1 สรุปผลประสิทธิภาพการทำงานของระบบ.....	33
5.2 ข้อจำกัดของระบบที่พัฒนา.....	33
5.3 ข้อเสนอแนะ.....	33
 บรรณานุกรม.....	 34
 ภาคผนวก ก .....	 35
ภาคผนวก ข .....	52



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงรายละเอียดของไวยากรณ์พื้นฐานในการรู้จำเสียงของคอมโพเนนท์รู้จำเสียง.....	13
3.2 แสดงรายละเอียดตารางข้อมูลรายวิชา.....	15
3.3 แสดงรายละเอียดตารางข้อมูลผลการเรียน.....	16
3.4 แสดงรายละเอียดตารางข้อมูลนักศึกษา.....	16
3.5 แสดงรายละเอียดตารางรูปแบบเสียง.....	17
3.6 แสดงรายละเอียดตารางตัวอย่างของเสียง.....	17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
2.1 การเปรียบเทียบการพัฒนาแบบปกติเทียบกับการพัฒนาแบบใช้ COTS.....	4
2.2 การนำ COTS-Based มาใช้งาน.....	5
3.1 ภาพรวมของระบบการสอบถามข้อมูลด้วยเสียง.....	8
3.2 ระบบการฝึกเสียง.....	9
3.3 ระบบตัวอย่างการสอบถามข้อมูลจากฐานข้อมูล.....	11
3.4 ตัวอย่างไวยากรณ์พื้นฐานอย่างง่าย.....	14
3.5 ER-diagram ของฐานข้อมูลการสอบถามข้อมูลผลการเรียน.....	15
3.6 ER-diagram ของฐานข้อมูลการเก็บเสียงสั่งงาน.....	16
3.7 หน้าจอการทำงานของส่วนการฝึกเสียง.....	18
3.8 หน้าจอการทำงานของส่วนการเพิ่มเสียง.....	19
3.9 ส่วนของระบบการสอบถามข้อมูลผลการศึกษา.....	20
4.1 แสดงส่วนการเลือกกลุ่มคำที่มีเพื่อทำการจัดเก็บเสียงตัวอย่าง.....	27
4.2 แสดงเสียงที่มีการจัดเก็บไว้แล้วในกลุ่มคำที่เลือก.....	28
4.3 แสดงเสียงของแต่ละคนที่ได้ทำการจัดเก็บไว้ในฐานข้อมูลเสียง.....	29
4.4 แสดงส่วนของการเก็บเสียงจากผู้พูด.....	30
4.5 แสดงการเก็บเสียงที่ได้รับการเทรนลงในฐานข้อมูลเสียงตัวอย่าง.....	31
4.6 การเริ่มการรู้จำเสียงและเริ่มใส่คำด้วยคำพูด.....	31
4.7 แสดงการสั่ง “เลื่อน” เพื่อให้ เคอร์เซอร์ ไปยังช่องถัดไป.....	32
4.8 แสดงหน้าจอหลังจากการออกคำสั่งจาก “เลื่อน หรือ ตกลง”.....	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ด้วยปัจจุบันได้มีการนำเทคโนโลยีฐานข้อมูลมาใช้ในการจัดเก็บข้อมูลของหน่วยงานต่างๆ อย่างแพร่หลาย เพื่ออำนวยความสะดวกต่อผู้ใช้ข้อมูลให้สามารถเรียกค้นข้อมูลได้แบบทันทีทันใด แต่อย่างไรก็ตามการเรียกค้นข้อมูลจากฐานข้อมูลยังจำกัดอยู่ที่ภาษาฐานข้อมูลซึ่งไม่เป็นธรรมชาติต่อผู้ใช้ ดังนั้น หากการสอบถามข้อมูลสามารถทำได้โดยใช้เสียงพูดจักเป็นประโยชน์อย่างยิ่งต่อผู้ใช้ ด้วยเทคโนโลยีเสียงปัจจุบันทำให้การสอบถามข้อมูลจากฐานข้อมูลด้วยเสียงพูดเป็นไปได้

### 1.2 วัตถุประสงค์ของการศึกษา

เพื่อพัฒนาระบบสอบถามข้อมูลจากฐานข้อมูลด้วยเสียง โดยการเชื่อมต่อฐานข้อมูลกับคอมพิวเตอร์รู้จำเสียง

### 1.3 ขอบเขตของการศึกษา

ระบบฐานข้อมูลที่ใช้ในปัญหาพิเศษนี้ เป็นระบบฐานข้อมูลอย่างง่าย ที่เก็บผลการเรียนของนักศึกษา การพัฒนาระบบ เพื่อให้สามารถสอบถามข้อมูลจากฐานข้อมูลนี้ได้ด้วยเสียง จะใช้แนวคิดการพัฒนาระบบด้วยคอมพิวเตอร์ โดยเริ่มจากการกำหนดความต้องการคอมพิวเตอร์รู้จำเสียง แล้วทำการค้นหาคอมพิวเตอร์รู้จำเสียงในท้องตลาดที่เหมาะสม แล้วจึงนำมาเชื่อมต่อเข้ากับระบบฐานข้อมูล

### 1.4 ขั้นตอนการศึกษา

- 1.4.1 ศึกษาเทคโนโลยีการรู้จำเสียงที่มีการใช้งานอยู่และคัดเลือกมาใช้งาน
- 1.4.2 ศึกษาเทคโนโลยีที่คัดเลือกมาและนำมาประยุกต์ใช้กับระบบจำลอง
- 1.4.3 สร้างอินเตอร์เฟซของระบบจำลอง และสร้างฐานข้อมูลของระบบจำลองขึ้น
- 1.4.4 ทำการทดสอบแอปพลิเคชันที่สร้างขึ้น โดยตรวจสอบและแก้ไขปัญหาของระบบ
- 1.4.5 วิเคราะห์ระบบและพัฒนาให้เหมาะสมกับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ผลการศึกษา这不仅นอกจากจะได้รับระบบการสอบถามข้อมูลด้วยเสียงแล้ว ยังแสดงให้เห็นถึงความเป็นไปได้ของการผสมผสานเทคโนโลยีเพื่อการใช้งาน โดยการนำผลผลิตซอฟต์แวร์ที่มีอยู่มาสร้างผลิตภัณฑ์ใหม่ซึ่งสอดคล้องกับแนวความคิดการพัฒนาสมัยใหม่ที่ไม่ต้องเริ่มจากศูนย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# แนวความคิดพื้นฐานและทฤษฎีที่เกี่ยวข้อง

### 2.1 การพัฒนาระบบด้วยคอมโพเนนท์

การพัฒนาระบบด้วยคอมโพเนนท์(Component-Based Software Development: CBSD) กล่าวถึงการสร้างระบบขนาดใหญ่ด้วยซอฟต์แวร์ โดยการรวบรวมเอาซอฟต์แวร์ที่มีอยู่แล้วกลับมาใช้เพื่อเพิ่มประสิทธิภาพของซอฟต์แวร์โดยทำการเปลี่ยนแปลงและบำรุงรักษาระบบ ซึ่งเป็นการลดต้นทุนในการพัฒนาซอฟต์แวร์ การรวบรวมระบบ และลดต้นทุนในการบำรุงรักษาระบบแบบสไปรอล(Spiral) ของระบบที่เกี่ยวข้อง ซึ่งทำการสนับสนุนและอัปเดตส่วนต่างๆ ในระบบใหญ่ๆ โดยพื้นฐานของวิธีนี้จะมองว่าส่วนต่างๆ ของซอฟต์แวร์ขนาดใหญ่ จะมีการนำมาใช้บ่อยๆ จึงได้มีการนำส่วนของซอฟต์แวร์ที่เห็นว่ามีการใช้งานบ่อยครั้งมาเป็นพื้นฐานการนำไปใช้งาน ซึ่งการพัฒนาระบบด้วยคอมโพเนนท์ จะสนับสนุนการนำซอฟต์แวร์หรือคอมโพเนนท์ที่มีอยู่แล้วในท้องตลาดมาใช้มากกว่าที่จะเขียนขึ้นมาเองใหม่ทั้งหมด เพื่อเป็นการลดค่าใช้จ่ายในส่วนต่างๆ ในการผลิตซอฟต์แวร์ใหม่ๆ

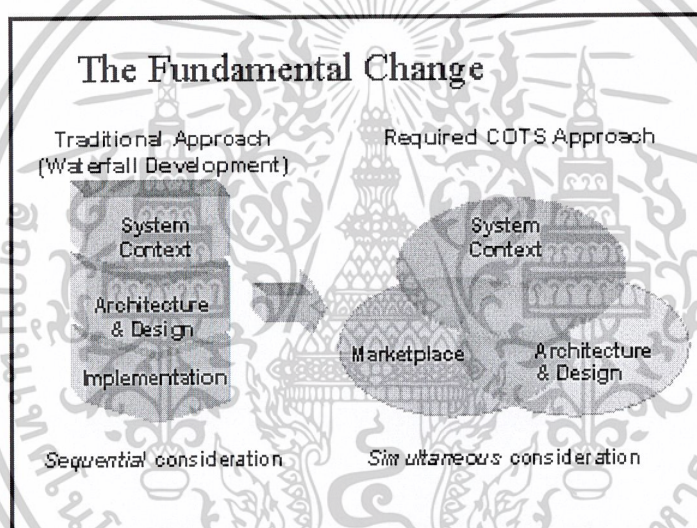
การพัฒนาระบบด้วยคอมโพเนนท์ ได้รวมทั้งสองส่วนไว้ด้วยกันคือ คอมโพเนนท์สำเร็จรูป (Commercial-off-the-shelf : COTS) และคอมโพเนนท์ที่ต้องการในระบบใหม่ ซึ่งการพัฒนาระบบด้วยคอมโพเนนท์จะทำให้เกิดประโยชน์ในหลายๆ ด้าน ดังนี้

- การเพิ่มในด้านคุณภาพ และความหลากหลายของผลิตภัณฑ์คอมโพเนนท์สำเร็จรูป
- ทางด้านเศรษฐกิจ จะลดค่าใช้จ่ายในการพัฒนาและการบำรุงรักษาระบบ
- เพิ่มเทคโนโลยีจากการรวบรวมระบบใหม่
- เพิ่มส่วนประกอบของซอฟต์แวร์ที่มีอยู่ซึ่งสามารถนำกลับมาใช้ในระบบใหม่ได้

การพัฒนาระบบด้วยคอมโพเนนท์จะเป็นการเปลี่ยนรูปแบบจาก “การเขียนโปรแกรม” เป็น “การประกอบโปรแกรม”

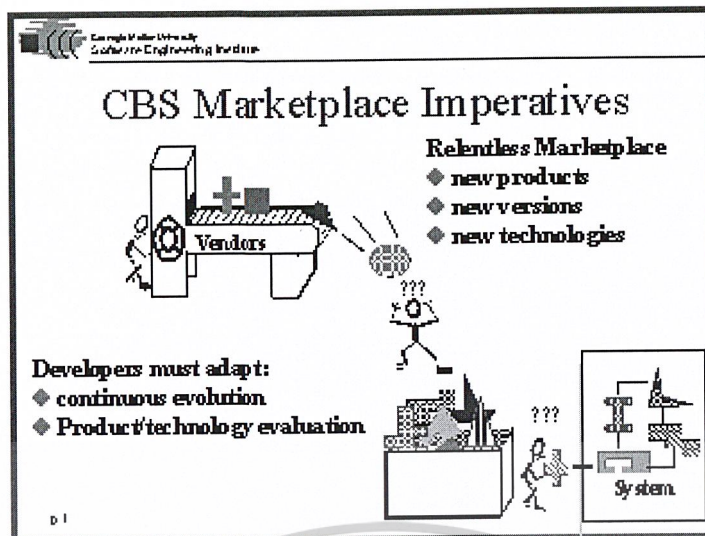
## 2.2 คอมโพเนนต์สำเร็จรูป

คอมโพเนนต์สำเร็จรูป(Component off-the-shelf Software : COTS) เป็นระบบที่กล่าวถึงการปรับปรุงเทคโนโลยี และในทางปฏิบัติคือใช้สำหรับประกอบกับคอมโพเนนต์ที่มีอยู่แล้วจะถูกนำไปใช้ในระบบซอฟต์แวร์ใหญ่ๆ และระบบที่มีใช้อยู่ปัจจุบัน โดยจากการทำระบบคอมโพเนนต์จุดประสงค์ในการทำก็เพื่อทำการเปลี่ยนวิศวกรรมซอฟต์แวร์บางส่วน โดยการพัฒนาซอฟต์แวร์แบบระบบคอมโพเนนต์นี้จะเปลี่ยนการทำวิศวกรรมซอฟต์แวร์จากแบบเดิมคือ“รับความต้องการมาแล้วมาสร้างระบบ”ไปเป็นแบบใหม่ที่ต้องอาศัยการพิจารณาลักษณะของระบบ(System context) หลายๆ ส่วน เช่น ความต้องการ ค่าใช้จ่าย ระยะเวลา สภาพแวดล้อมการทำงาน เป็นต้น พิจารณาความสามารถของผลิตภัณฑ์ในตลาดและสถาปัตยกรรมที่เปลี่ยนแปลงได้พร้อมๆ กัน



รูปที่ 2.1 การเปรียบเทียบการพัฒนาแบบปกติเทียบกับการพัฒนาแบบใช้คอมโพเนนต์สำเร็จรูป

จากการเปลี่ยนแปลงนี้จะส่งผลกับการทำงานในส่วนของวิศวกรรม เช่น การกำหนดความต้องการจะต้องเปลี่ยนไปเพื่อรองรับความต้องการของทั้งระบบ สถาปัตยกรรม และการออกแบบรวมทั้งการตลาดด้วย ซึ่งผลที่ได้รับก็คือกระบวนการทำงานและข้อตกลงต่างๆ ที่เหมาะสม



รูปที่ 2.2 การนำคอม โฟเนนท์สำเร็จรูปมาใช้งาน

สำหรับการทำการออกแบบ พัฒนา และการบำรุงรักษาระบบโดยใช้คอม โฟเนนท์สำเร็จรูป หรือทำการอัพเกรดระบบเก่าด้วยคอม โฟเนนท์สำเร็จรูปจะพบความยากในการพัฒนา ดังนี้

1. ผลผลิตของคอม โฟเนนท์สำเร็จรูป จะมีโครงสร้างของตัวเองซึ่งอาจจะไม่ตรงกับระบบที่ต้องการพัฒนา
2. ผลผลิตของคอม โฟเนนท์สำเร็จรูป จะมีข้อกำหนดในการใช้งานซึ่งอาจจะไม่ตรงกับแนวทางการทำงานของผู้ใช้

ดังนั้นในการออกแบบและสร้างระบบคอม โฟเนนท์สำเร็จรูป หรือการปรับปรุงระบบเก่า โดยการใช้ผลิตภัณฑ์ คอม โฟเนนท์สำเร็จรูป จะต้องสามารถตอบคำถามเหล่านี้ได้

1. เทคโนโลยีและผลิตภัณฑ์อะไรที่เหมาะสมที่สุด
2. ทำอย่างไร ที่จะแก้ไขผลิตภัณฑ์ที่ไม่ตรงกับระบบ
3. ทำอย่างไรในการควบคุมการผลิตต่างๆ เช่น ความน่าเชื่อถือ ระบบรักษาความปลอดภัย และ ประสิทธิภาพ เมื่อเกิดปัญหาความเข้ากันไม่ได้ของคอม โฟเนนท์
4. ทำการรวมผลิตภัณฑ์ของคอม โฟเนนท์สำเร็จรูปกับโค้ด ที่เป็นส่วนสำคัญในหลายๆ ระบบ ได้อย่างไร
5. จะได้รับประโยชน์อย่างไรจากการนำคอม โฟเนนท์มาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 คอมพิวเตอร์รู้จำเสียงสำเร็จรูป

ปัจจุบันได้มีผู้ผลิตหลายรายได้ให้ความสำคัญในเรื่องการใช้เสียงในการสั่งงานหรือป้อนข้อมูลให้กับระบบแทนการใช้อุปกรณ์ เช่น คีย์บอร์ดหรือเมาส์ จึงได้ทำการผลิตคอมพิวเตอร์รู้จำเสียงสำเร็จรูป(Speech Recognition COTS Software) เพื่อให้ผู้พัฒนาระบบต่างๆ สามารถนำคอมพิวเตอร์รู้จำเสียงรูปไปทำการพัฒนาระบบได้ง่ายมากขึ้น ซึ่งในการศึกษาคอมพิวเตอร์รู้จำเสียงของปัญหาพิเศษได้นำคอมพิวเตอร์รู้จำเสียงมาศึกษา ดังนี้

### 2.3.1 ตัวพัฒนาเสียงของบริษัทฟิลิปส์

ตัวพัฒนาเสียงของบริษัทฟิลิปส์(Philips Speech SDK)เป็นตัวพัฒนา(Software Development Kit)จึงสามารถนำไปพัฒนาร่วมกับ โปรเจกอื่นได้และตัวพัฒนาเสียงของบริษัทฟิลิปส์มีส่วนของโค้ดตัวอย่างให้สำหรับในการศึกษาและนำไปพัฒนา โดยคุณสมบัติหลักๆ ของตัวพัฒนาตัวนี้แบ่งเป็น 2 ส่วนคือ

- Dictation เป็นส่วนที่ทำหน้าที่แปลงเสียงเป็นข้อความ (Speech to Text)
- Command & Control เป็นคอมพิวเตอร์ที่สามารถทำให้ผู้ใช้ควบคุม โปรแกรมต่างๆ โดยใช้เสียงพูดได้

### 2.3.2 ตัวพัฒนาเสียงของบริษัทไมโครซอฟท์

ตัวพัฒนาเสียงของบริษัทไมโครซอฟท์(Microsoft Speech SDK:SAPI) เป็นตัวพัฒนาเสียงที่มีส่วนคล้ายกับตัวพัฒนาเสียงของบริษัทฟิลิปส์ตรงที่ต่างก็เป็นตัวพัฒนาเหมือนกันและมีคุณสมบัติหลักๆ ของตัวพัฒนาตัวนี้ 3 ส่วน คือ

- Command & Control เป็นคอมพิวเตอร์ที่สามารถทำให้ผู้ใช้ควบคุม โปรแกรมต่างๆ โดยใช้เสียงพูดได้
- Speech to Text ที่ทำหน้าที่แปลงเสียงเป็นข้อความ และ
- Text to Speech ที่ทำหน้าที่แปลงข้อความ เป็นเสียงพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากตัวพัฒนาเสียงของบริษัทฟิลิปส์และตัวพัฒนาเสียงของบริษัทไมโครซอฟท์ที่มีคุณสมบัติไม่แตกต่างกันมากนัก และในตัวพัฒนาเสียงของบริษัทไมโครซอฟท์เป็นคอมโพเนนต์ที่เป็นลักษณะของฟรีแวร์(Freeware) โดยมีส่วนการใช้งานคอมโพเนนต์ที่เหมาะสมกับการพัฒนาระบบของปัญหาพิเศษนี้ จึงทำการพิจารณาใช้งานตัวพัฒนาเสียงของบริษัทไมโครซอฟท์สำหรับการพัฒนาระบบอินเทอร์เน็ตเฟสด้วยเสียงเพื่อสอบถามฐานข้อมูล



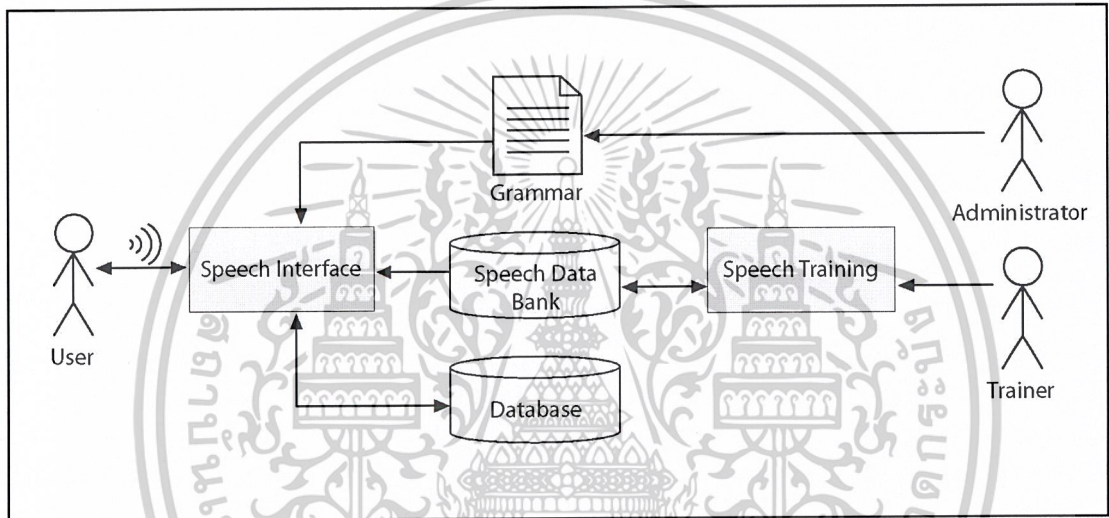
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

## ระบบการสอบถามข้อมูลจากฐานข้อมูลด้วยเสียงพูด

### 3.1 ส่วนประกอบของระบบ

ระบบการสอบถามข้อมูลจากฐานข้อมูลด้วยเสียงนั้นจะประกอบด้วย 3 ส่วนย่อยๆ ได้แก่ ส่วนของการฝึกเสียง(Speech Training) ส่วนของการสร้างไวยากรณ์(Grammar) และส่วนของตัวอย่างการสอบถามข้อมูล(Speech Interface) ซึ่งแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของระบบการสอบถามข้อมูลด้วยเสียง

Speech Interface

เป็นส่วนการสอบถามข้อมูลจากฐานข้อมูล ทำหน้าที่ในการสอบถามข้อมูลจากฐานข้อมูล

Speech Training

เป็นส่วนการฝึกเสียง ทำหน้าที่ในการฝึกให้ระบบสามารถรู้จำคำได้แม่นยำขึ้นรวมถึงการบันทึกเสียงลงในฐานข้อมูลการรู้จำเสียง

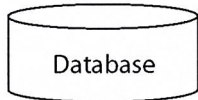
Grammar

เป็นส่วนของไฟล์ไวยากรณ์พื้นฐานของระบบการสอบถามข้อมูล ซึ่งมีหน้าที่สำคัญในการเป็นตัวบอกระบบให้มุ่งความสนใจไปยัง “กลุ่มคำ” เพียงกลุ่มเดียว เพื่อประโยชน์ด้าน ความเร็ว ความถูกต้อง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เป็นฐานข้อมูลที่เก็บรวบรวมคำพูดทั้งหมดของระบบ เช่น ในระบบการสอบถามข้อมูลผลการศึกษานั้นจะมีค่าต่างๆ เช่น 0-9 , เลื่อน , ตกลง เป็นต้น ซึ่งจะถูกเก็บรวบรวมไว้ในฐานข้อมูลของเสียงนี้



เป็นฐานข้อมูลของระบบที่จะนำมาทำการสอบถามข้อมูลเช่น ฐานข้อมูลผลการศึกษานักศึกษา ฐานข้อมูลประวัติพนักงาน เป็นต้น

ซึ่งรายละเอียดต่างๆ ของแต่ละส่วนมี ดังนี้

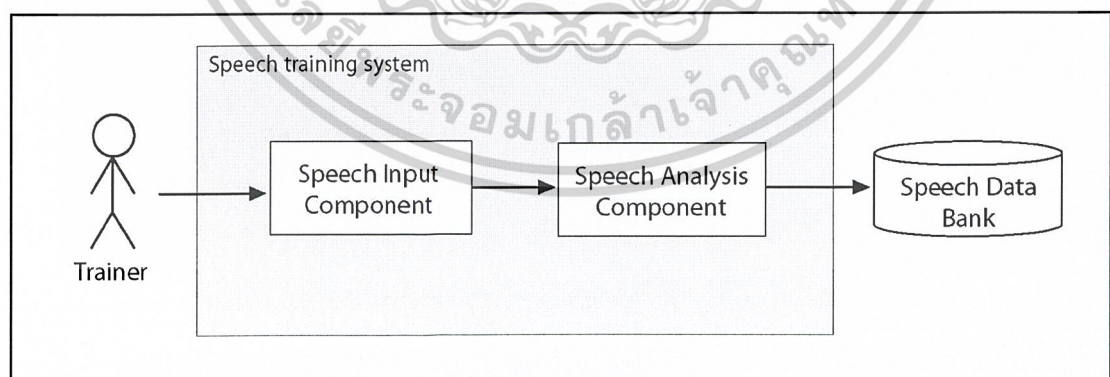
### 3.1.1 ส่วนของการฝึกเสียง

ในส่วนของการฝึกเสียงนั้นจะมีส่วนประกอบต่างๆ ดังนี้

3.1.1.1 ส่วนการรับเสียงพูด (Speech Input Component) ทำหน้าที่เป็นอินเตอร์เฟซในการรับเสียงพูดจากผู้ใช้ ทั้งการควบคุมคุณภาพเสียง การติดต่อกับส่วนควบคุมเสียงย่อยของวินโดวส์ (Windows sub sound system) สิ่งที่ได้จะเป็นคลื่นเสียงในรูปแบบดิจิทัล-ออดิโอ

3.1.1.2 ส่วนการวิเคราะห์และจำแนกเสียงพูด (Speech Analysis Component) ทำหน้าที่แปลงคลื่นเสียงในรูปแบบ ดิจิตอล-ออดิโอ ให้เป็นส่วยย่อยๆ ของเสียง (Phoneme)

3.1.1.3 ส่วนฐานข้อมูลเสียงพูด (Speech Data Bank) เป็นส่วนที่เก็บรวบรวมคำพูดทั้งหมดของระบบ เช่น คำว่า test ออกเสียงว่า [ t e a s t ]



รูปที่ 3.2 ระบบการฝึกเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 ส่วนของการสร้างไวยากรณ์

ในการสร้างไวยากรณ์ จะนำข้อมูลเสียงที่ได้ทำการฝึกในส่วนของการฝึกเสียงนั้น มาใช้ในการสร้างไวยากรณ์ที่ถูกต้องสำหรับระบบสอบถามข้อมูลที่เราต้องการ ซึ่งไวยากรณ์ที่สร้างขึ้นจะอยู่ในรูปแบบไฟล์ XML โดยที่รูปแบบไวยากรณ์ แท็กพื้นฐานที่ใช้ รายละเอียดต่างๆ ของแท็กพื้นฐาน และตัวอย่างไวยากรณ์อย่างง่าย จะทำการอธิบายในหัวข้อถัดไป

### 3.1.3 ส่วนของตัวอย่างการสอบถามข้อมูลจากฐานข้อมูล

ในส่วนของตัวอย่างที่จัดทำขึ้นในโครงการพิเศษนี้ คือ ระบบการสอบถามผลการศึกษาของนักศึกษา ซึ่งในระบบตัวอย่างนี้จะมีส่วนประกอบต่างๆ ดังนี้

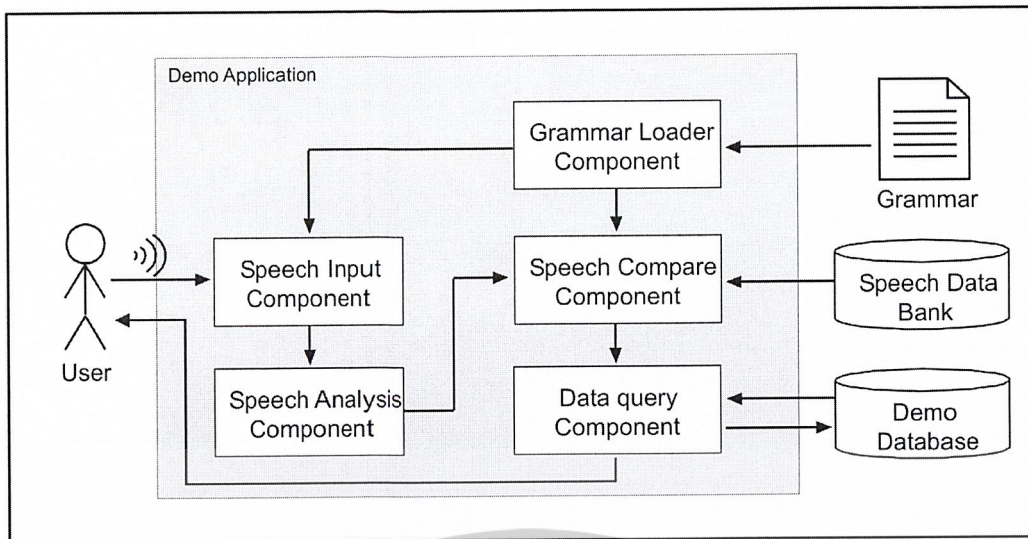
3.1.3.1 ส่วนการรับเสียงพูด การทำงานในส่วนนี้เหมือนกับส่วนการรับเสียงพูดของระบบการฝึกเสียง

3.1.3.2 ส่วนการวิเคราะห์และจำแนกเสียงพูด การทำงานในส่วนนี้เหมือนกับส่วนการวิเคราะห์และจำแนกเสียงพูดของระบบการฝึกเสียง

3.1.3.3 ส่วนการจัดเตรียมไวยากรณ์ (Grammar Loader Component) ทำหน้าที่อ่านไวยากรณ์จากไฟล์ ไวยากรณ์ เพื่อให้ระบบเตรียมการรู้จำเฉพาะกับกลุ่ม “คำ” นั้นๆ

3.1.3.4 ส่วนการเปรียบเทียบ (Speech Compare Component) ทำหน้าที่ ดึงข้อมูลวิเคราะห์ได้จาก “ส่วนการวิเคราะห์และจำแนกเสียงพูด” และ “ส่วนการจัดเตรียมไวยากรณ์” เพื่อมาเปรียบเทียบกับฐานข้อมูลเสียง

3.1.3.5 ส่วนการสอบถามฐานข้อมูล (Data Query Component) ทำหน้าที่เรียก “ชุดคำสั่ง SQL (SQL statement)” ที่ตรงกับ “ผลการเปรียบเทียบ” ไปสอบถามข้อมูลจากฐานข้อมูล แล้วนำผลลัพธ์ที่ได้ แสดงให้กับผู้ใช้



รูปที่ 3.3 ระบบตัวอย่างการสอบถามข้อมูลจากฐานข้อมูล

ส่วนย่อยทั้ง 3 ระบบ มีการใช้คอมโพเนนท์การรู้จำเสียงต่างๆ เช่น คอมโพเนนท์การรับเสียง หรือ คอมโพเนนท์การวิเคราะห์เสียง เป็นต้น ดังนั้นเมื่อพิจารณาแล้ว SAPI ได้มีการจัดเตรียมคอมโพเนนท์ต่างๆ ที่จำเป็นในการดำเนินการต่างๆ ข้างต้นไว้แล้ว ซึ่งมีส่วนช่วยในการสร้างระบบย่อยทั้ง 3 ระบบข้างต้น ซึ่งรายละเอียดของคอมโพเนนท์การรู้จำเสียงต่างๆ จะทำการอธิบายในหัวข้อต่อไป

### 3.2 คอมโพเนนท์รู้จำเสียง

คอมโพเนนท์รู้จำเสียง(SAPI) จะประกอบไปด้วยคอมโพเนนท์ในการทำงาน 2 ส่วน คือ ส่วนของการแปลงข้อความเป็นเสียงและส่วนของการรู้จำเสียง ดังที่ได้กล่าวมาข้างต้นในบทที่ 2 ซึ่งใน ครงงานพิเศษนี้เราจะใช้คอมโพเนนท์ของการรู้จำเสียง ซึ่งประกอบไปด้วยคอมโพเนนท์ต่างๆ ดังนี้

### 3.2.1 ISpRecognizer

อินเทอร์เฟซ ISpRecognizer เป็นตัวควบคุม "กลไกการทำงาน" (engine) ทั้งหมดของตัว "รู้จำเสียง" (SR : Speech Recognition)

อินเทอร์เฟซ ISpRecognizer ตัวหนึ่งเปรียบได้กับการรู้จำเสียงตัวหนึ่ง ซึ่งตัวโปรแกรมสามารถเชื่อมต่อกับออปเจกต์(Object) แต่ละตัวของ *recognizer* , *recognition\_contexts* ซึ่งตัวโปรแกรมสามารถควบคุมไวยากรณ์ที่จะนำไปใช้ได้ สามารถสั่งเริ่ม สั่งหยุดการรู้จำได้ รับเหตุการณ์(event) และแสดงผลลัพธ์จากการรู้จำได้ ตัวอินเทอร์เฟซ ISpRecognizer อนุญาตให้ควบคุมกลไกการทำงานของการรู้จำเสียง ในเรื่องการรับเสียง(audio input) ในตัวโปรแกรมทั่วไปไม่จำเป็นต้องเรียกกระบวนการการทำงาน(methods) ทุกกระบวนการในอินเทอร์เฟซนี้ เนื่องจากว่าคอมไพเลอร์ที่รู้จำเสียงได้ทำการจัดการเรื่องมาตรฐานการรับเสียงอยู่แล้ว

### 3.2.2 ISpRecoContext

อินเทอร์เฟซ ISpRecoContext ยอมให้ตัวโปรแกรมสร้าง functional views หลายๆ แบบ จากกลไกการทำงานของการรู้จำเสียงซึ่งออปเจกต์แต่ละตัวของ ISpRecoContext สามารถมุ่งความสนใจไปยังเหตุการณ์แต่ละเหตุการณ์ของการรู้จำเสียงเพื่อใช้ไวยากรณ์การรู้จำที่แตกต่างกันตามเหตุการณ์ที่ต่างกัน ในตัวโปรแกรมจำเป็นต้องมี ISpRecoContext อย่างน้อยหนึ่งตัวเพื่อรับเหตุการณ์ในการรู้จำเสียง และในตัวโปรแกรมสามารถสร้าง ISpRecoContext หลายๆตัวเพื่อจำแนกชนิดของเหตุการณ์ให้ตรงกับไวยากรณ์

### 3.2.3 ISpRecoGrammar

อินเทอร์เฟซ ISpRecoGrammar ยอมให้ตัวโปรแกรมจัดการกับ "คำ" และ "วลี" เพื่อให้กลไกการทำงานของการรู้จำเสียงนำไปทำการรู้จำเสียง

### 3.2.4 ISpRecoResult

อินเทอร์เฟซ ISpRecoResult ถูกใช้โดยตัวโปรแกรมเพื่อ "รับ" ข้อมูลเกี่ยวกับสมมติฐาน การรู้จำ และการรู้จำที่ผิดจากกลไกการทำงานของการรู้จำเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในส่วนของรูปแบบของไวยากรณ์นั้นจะขออธิบายเฉพาะไวยากรณ์พื้นฐานที่ใช้ในการทำโครงงานนี้ โดยรูปแบบไวยากรณ์ของคอมไพเลอร์ที่รู้จำเสียงนั้นจะประกอบไปด้วยแท็ก XML ซึ่งใช้ในการนิยามวลีหรือคำต่างๆที่กลไกการทำงานของกรรผู้จำเสียงใช้ในการรู้จำเสียง ซึ่งในส่วนของไวยากรณ์พื้นฐานจะมีรายละเอียด ดังที่แสดงในตารางที่ 3.1

ตารางที่ 3.1 แสดงรายละเอียดของไวยากรณ์พื้นฐานในการรู้จำเสียงของคอมไพเลอร์ที่รู้จำเสียง

แท็ก	รายละเอียด
<GRAMMAR>	เป็นแท็กเริ่มต้นของไวยากรณ์ที่ใช้ซึ่งจะมีแอตทริบิวต์ (attribute) ดังนี้ LANGID : เป็นการระบุภาษาที่ใช้ในไวยากรณ์
<DEFINE>	ใช้ในการนิยามกลุ่มของข้อความที่ใช้ในไวยากรณ์นั้นๆ ซึ่งจะมี <ID> เป็นองค์ประกอบ(element)ย่อย
<ID>	ใช้ในการประกาศ “กฎ(Rule)” ที่ใช้ในไวยากรณ์นั้นๆ ซึ่งจะมีแอตทริบิวต์ ดังนี้ NAME : เป็นการกำหนดชื่อของกฎที่ตั้ง VAL : เป็นการกำหนดค่าคงที่ที่กำหนดซึ่งจะมีความสัมพันธ์กับข้อความที่ระบุในไวยากรณ์ ซึ่งทั้ง NAME และ VAL เป็นแอตทริบิวต์ที่ต้องมี
<RULE>	เป็นแท็กที่ใช้ในการนิยามข้อความหรือคำต่างๆที่ อนุญาตให้ใช้ในการรู้จำเสียง ซึ่งจะมีแอตทริบิวต์ต่างๆ ที่ใช้งานดังนี้ NAME : เป็นชื่อของกฎที่ตั้งขึ้น ซึ่งจะเหมือนกับชื่อที่ตั้งใน <ID> TOPLEVEL : ค่าที่กำหนดให้ คือ “ACTIVE” กำหนดให้สามารถใช้งานกฎนั้นๆได้ ซึ่งในส่วนของกฎอ้างอิงไม่จำเป็นต้องใส่แอตทริบิวต์นี้ก็ได้นี้เนื่องจากกฎนั้นจะถูกอ้างอิงถึงโดยกฎอื่นอยู่แล้ว และ <RULE> จะมีองค์ประกอบย่อยต่างๆ ดังนี้ <RULEREF> , <P> หรือ <PHRASE> , <O> หรือ <OPT> , <LIST> หรือ <L>
<RULEREF>	ทำหน้าที่ในการดึงข้อมูลกฎจากกฎอื่นในไวยากรณ์เดียวกันหรือคนละไวยากรณ์มาใช้ ซึ่งจะมีแอตทริบิวต์ที่ใช้งานดังนี้ NAME : ชื่อของกฎที่สัมพันธ์กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<P> หรือ <PHRASE>	เป็นการกำหนดคำที่ต้องทำการใช้งานในการพูด <span style="float: right;">ซึ่งจะมี</span> แอดทริบิวต์ที่ใช้งาน คือ DISP : เป็นการกำหนดข้อความที่ใช้ในการแสดงผล
<O> หรือ <OPT>	เป็นการกำหนดคำที่สามารถใช้หรือไม่ใช้ก็ได้ในการพูด <span style="float: right;">ซึ่งจะมี</span> แอดทริบิวต์ที่ใช้งาน คือ DISP : เป็นการกำหนดข้อความที่ใช้ในการแสดงผล
<L> หรือ <LIST>	เป็นการกำหนดรายการของข้อความที่สามารถพูดได้ <span style="float: right;">ซึ่งมีใน</span> ส่วนขององค์ประกอบย่อมนั้นจะเป็นได้เฉพาะ <P> หรือ <PHRASE> เท่านั้น

```

<GRAMMAR LANGID="409">
  <RULE NAME="E-MAIL" TOPLEVEL="INACTIVE">
    <PHRASE>send new e-mail to</P>
    <RULEREFF NAME="ADDRESS_BOOK" PROPNAME="NAME"/>
  </RULE>
  <RULE NAME="ADDRESS_BOOK" DYNAMIC="TRUE">
    <PHRASE>placeholder</PHRASE>
  </RULE>
</GRAMMAR>

```

รูปที่ 3.4 ตัวอย่างไวยากรณ์พื้นฐานอย่างง่าย

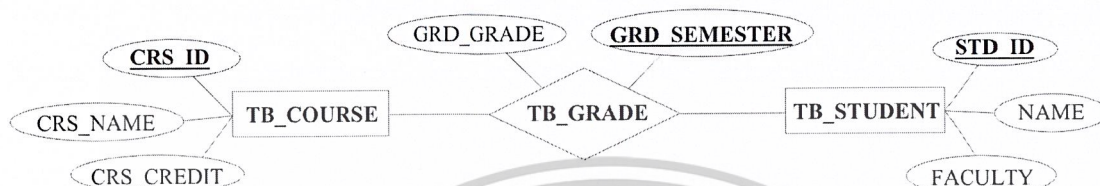
### 3.3 ฐานข้อมูล

ฐานข้อมูลที่นำมาใช้กับระบบที่พัฒนา แบ่งออกเป็นสองส่วน คือฐานข้อมูลการสอบถามข้อมูล ผลการเรียนของนักศึกษา และส่วนของฐานข้อมูลที่เก็บตัวอย่างเสียงที่นำมาใช้กับระบบที่พัฒนาขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 ฐานข้อมูลตัวอย่างการสอบถามข้อมูลผลการเรียนของนักศึกษา

ฐานข้อมูลการสอบถามข้อมูล ดังแสดงในรูปที่ 3.5 โดยฐานข้อมูลนี้จะทำการแบ่งออกเป็น 3 ตาราง คือ ตารางข้อมูลรายวิชา(TB\_COURSE) ตารางข้อมูลผลการเรียน(TB\_GRADE) และ ตารางข้อมูลนักศึกษา(TB\_STUDENT)



รูปที่ 3.5 ER-diagram ของฐานข้อมูลการสอบถามข้อมูลผลการเรียน

#### 3.3.1.1 ตารางข้อมูลรายวิชา

ตารางข้อมูลรายวิชา จะเก็บข้อมูลของรายวิชาของสถาบันที่มีการเรียนการสอน โดยเก็บรายละเอียดดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 แสดงรายละเอียดตารางข้อมูลรายวิชา

ลำดับที่	ชื่อคอลัมน์	ชนิดของข้อมูล	ความหมาย
1	CRS_ID (PK,FK)	Varchar(8)	CourseID: รหัสวิชา
2	CRS_NAME	Varchar(250)	CourseName: ชื่อวิชา
3	CRS_CREDIT	Smallint(3) unsigned	CourseCredit: หน่วยการเรียน

หมายเหตุ PK(Primary Key) คือคีย์หลัก, FK(Foreign Key) คือคีย์อ้างอิง

#### 3.3.1.2 ตารางข้อมูลผลการเรียน

ตารางข้อมูลผลการเรียน จะเก็บข้อมูลผลการเรียนของนักศึกษาที่ลงทะเบียนเรียน ในรายวิชาที่เปิดสอนของแต่ละเทอมและปีการศึกษา โดยเก็บรายละเอียดดังแสดงในตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 แสดงรายละเอียดตารางข้อมูลผลการเรียน

ลำดับที่	ชื่อคอลัมน์	ชนิดของข้อมูล	ความหมาย
1	STD_ID (PK,FK)	Varchar(8)	StudentID: รหัสนักศึกษา
2	CRS_ID (PK,FK)	Varchar(8)	CourseID: รหัสวิชา
3	GRD_SEMESTER (PK)	Varchar(6)	GradeSemester: ปีการศึกษา
4	GRD_GRADE	Char(2)	Grade: ผลการศึกษา

### 3.3.1.3 ตารางข้อมูลนักศึกษา

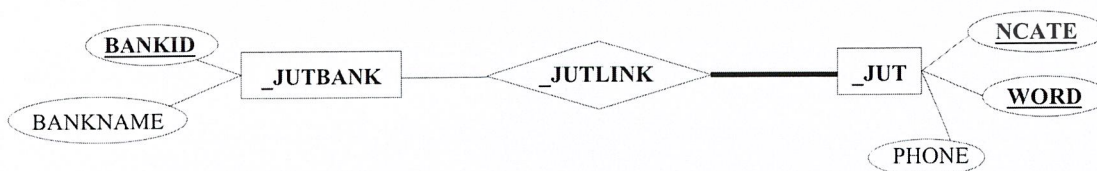
ตารางข้อมูลนักศึกษาจะทำการเก็บข้อมูลของนักศึกษาในสถาบัน โดยเก็บรายละเอียดดังตารางที่ 3.4

ตารางที่ 3.4 แสดงรายละเอียดตารางข้อมูลนักศึกษา

ลำดับที่	ชื่อคอลัมน์	ชนิดของข้อมูล	ความหมาย
1	STD_ID (PK,FK)	Varchar(8)	StudentID: รหัสนักศึกษา
2	NAME	Varchar(20)	Name: ชื่อนามสกุลนักศึกษา
3	FACULTY	Varchar(100)	Faculty: คณะ

### 3.3.2 ฐานข้อมูลตัวอย่างเสียง

ฐานข้อมูลตัวอย่างเสียงที่เก็บจะแบ่งออกเป็น 2 ส่วน คือส่วนของรูปแบบเสียงของการใช้งาน(\_JUTBANK) และส่วนของตัวอย่างเสียงตามแต่ละรูปแบบการใช้งาน(\_JUT) ซึ่งความสัมพันธ์ของ 2 ตาราง แสดงดังรูปที่ 3.6



รูปที่ 3.6 ER-diagram ของฐานข้อมูลการเก็บเสียงใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2.1 ตารางรูปแบบเสียง

ข้อมูลที่เก็บในตารางรูปแบบเสียงจะเป็นข้อมูลรูปแบบเสียงแต่ละระบบ โดยสามารถจำแนกรูปแบบเสียงที่เก็บได้ เช่น รูปแบบเสียงที่ใช้กับระบบสอบถามข้อมูลผลการศึกษา รูปแบบเสียงที่ใช้กับระบบการสอบถามข้อมูลโรงพยาบาล เป็นต้น โดยจะเก็บข้อมูลดังแสดงในตารางที่ 3.5

ตารางที่ 3.5 แสดงรายละเอียดตารางรูปแบบเสียง

ลำดับที่	ชื่อคอลัมน์	ชนิดของข้อมูล	ความหมาย
1	BANKID (PK)	Int(3)	BankID : รหัสประเภทเสียง
2	BANKNAME	Varchar(50)	BankName : ชื่อประเภทเสียง

### 3.3.2.2 ตารางตัวอย่างของเสียง

ข้อมูลที่เก็บในตารางตัวอย่างของเสียง เป็นข้อมูลเสียงที่มีในรูปแบบเสียงที่ต้องใช้งานทั้งหมด โดยจะเก็บข้อมูลดังแสดงในตารางที่ 3.6

ตารางที่ 3.6 แสดงรายละเอียดตารางตัวอย่างของเสียง

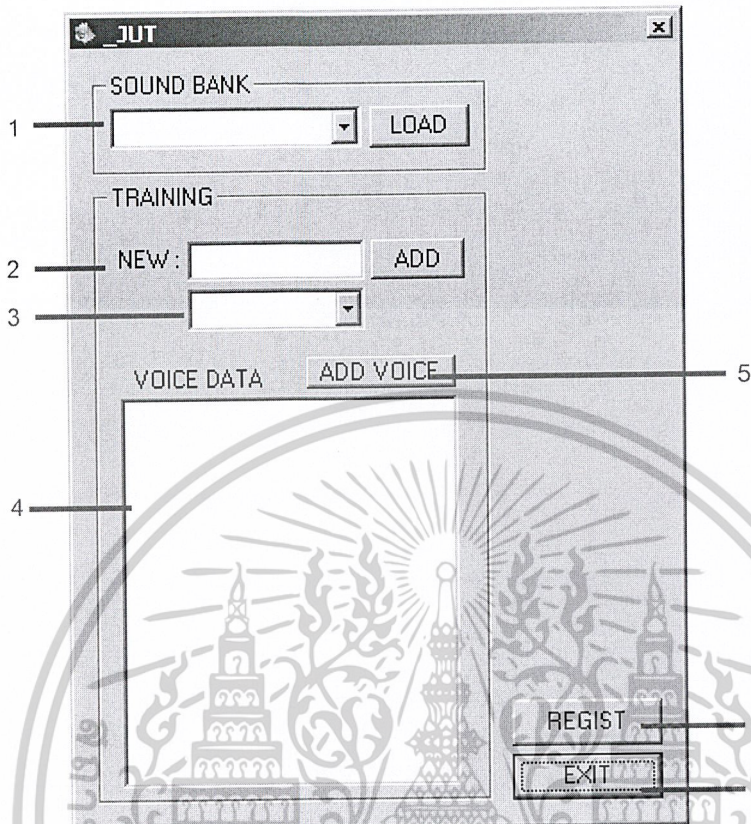
ลำดับที่	ชื่อคอลัมน์	ชนิดของข้อมูล	ความหมาย
1	BANKID (PK)	Int(3)	BankId : รหัสประเภทเสียง
2	NCATE (PK)	Varchar(8)	N Category : คำ
3	WORD (PK)	Varchar(8)	Word : ชื่อผู้พูด
4	PHONE	Varchar(50)	Phoneme : พยางค์ของคำที่เก็บ

## 3.4 ส่วนติดต่อกับผู้ใช้

ระบบจะแบ่งออกเป็น 2 ส่วน และในส่วนการติดต่อผู้ใช้ (User Interface) ก็จะแบ่งออกเป็น 2 ส่วนเช่นกัน ได้แก่ ส่วนการฝึกเสียง และ ส่วนของระบบการสอบถามข้อมูลผลการศึกษาของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

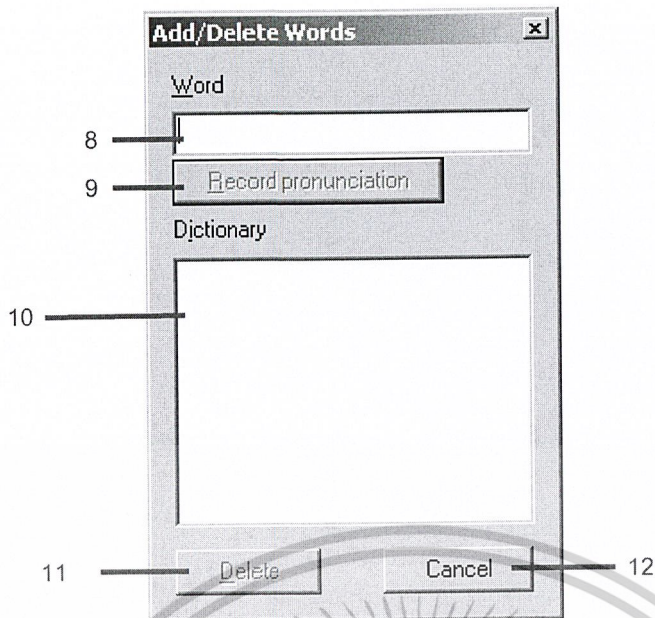
3.4.1 ส่วนการฝึกเสียง เป็นส่วนในการจัดการการฝึกเสียงที่ต้องการใช้ในระบบ ซึ่งแสดง  
 ดังรูปที่ 3.7 และ 3.8



รูปที่ 3.7 หน้าจอการทำงานของส่วนการฝึกเสียง

1. ฐานข้อมูลเสียงของแต่ละระบบ เมื่อคลิกปุ่ม “LOAD” ระบบจะทำการดึงข้อมูลของฐานข้อมูลเสียงในระบบต่างๆออกมาให้
2. ส่วนที่ใช้ในการสร้างคำใหม่ ในฐานข้อมูลเสียงที่เลือกไว้ โดยการพิมพ์ชื่อเพื่อระบุเสียงที่ต้องการเพิ่มแล้วคลิกปุ่ม “ADD”
3. ส่วนรายการเสียงที่มีในฐานข้อมูล
4. ส่วนรายการของผู้ที่ได้ทำการบันทึกเสียงของเสียงที่ได้ทำการเลือกจากข้อ 3
5. ส่วนของการเพิ่มการฝึกหัดของเสียงแต่ละเสียง ซึ่งจะอธิบายการทำงานของส่วนนี้ตามรูปที่ 3.8
6. ส่วนของการนำข้อมูลเสียงที่ได้เลือกไว้เก็บไว้ในฐานข้อมูลเสียง
7. ออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 หน้าจอการทำงานของส่วนการเพิ่มเสียง

8. ส่วนการอ้างอิงคำที่ใช้การฝึก
9. ปุ่มใช้ในการบันทึกเสียง
10. รายการเสียงที่ได้รับการฝึกแล้ว
11. ปุ่มที่ใช้ในการลบเสียง
12. ยกเลิก

### 3.4.2 ส่วนของระบบการสอบถามผลการศึกษาของนักศึกษา

ซึ่งจะแสดงให้เห็นดังรูปที่ 3.9 ซึ่งจะมีรายละเอียดดังนี้

1. ส่วนของข้อมูลต่างๆ ของนักศึกษา ซึ่งจะมี รหัสนักศึกษาและปีการศึกษา
2. ปุ่มแสดงผลการศึกษา
3. ปุ่มเลื่อนภาคการศึกษา
4. ช่องแสดงปีการศึกษาก่อนหน้า
5. ปุ่มเลื่อนภาคการศึกษาถัดไป
6. ส่วนทำการใช้งานไมโครโฟนในการรับเสียง
7. ส่วนแสดงชื่อนักศึกษา
8. ส่วนแสดงคณะของนักศึกษา
9. ส่วนแสดงผลการศึกษาในแต่ละปีการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 4

### การพัฒนาระบบ

#### 4.1 เครื่องมือที่ใช้ในการพัฒนาและการพัฒนาระบบ

ฐานข้อมูลที่ใช้ในการพัฒนาระบบไม่จำกัดว่าจะใช้ระบบการจัดการฐานข้อมูล(Database Management System : DBMS ) ตัวไหนซึ่งจะต้องสามารถใช้งานร่วมกับ ODBC ได้ เช่น ORACLE , Microsoft SQL , MySQL ฯลฯ ซึ่งระบบการสอบถามข้อมูลด้วยเสียงเป็นระบบที่มีขนาดเล็ก การใช้งานระบบการจัดการฐานข้อมูลขนาดใหญ่จึงไม่เหมาะสม ดังนั้นการพัฒนาระบบจึงเหมาะสมกับการใช้งาน MySQL ซึ่งเป็นระบบการจัดการฐานข้อมูลขนาดเล็ก ง่ายในการพัฒนา ดังนั้น ฐานข้อมูลทั้งหมดของระบบการสอบถามข้อมูลด้วยเสียงจึงเหมาะสมกับการใช้ MySQL

ระบบที่พัฒนาได้มีการแบ่งการทำงานออกเป็น 3 ส่วน

##### 4.1.1 ส่วนระบบการสอบถามข้อมูลการศึกษาของนักศึกษา

ส่วนระบบจำลองเป็นระบบที่ต้องใช้คอมพิวเตอร์หลักๆ ทุกตัวของคอมพิวเตอร์รู้จำเสียงของ SAPI ได้แก่ *Recognizer* , *RecoContext* , *RecoGrammar* และ *Lexicon* รวมทั้งคอมพิวเตอร์ที่ติดต่อกับฐานข้อมูล และการสร้างส่วนติดต่อผู้ใช้ที่ง่าย และพัฒนาได้เร็ว เมื่อพิจารณาแล้วเห็นได้ว่าทั้ง Visual C/C++ กับ Visual Basic มีขั้นตอนในการพัฒนาคอมพิวเตอร์ของ SAPI ใกล้เคียงกัน แต่การพัฒนาส่วนติดต่อผู้ใช้ของ Visual Basic สามารถพัฒนาได้ง่ายและเร็วกว่า จึงทำการใช้ Visual Basic ในการพัฒนาระบบส่วนนี้

ในการพัฒนาระบบการสอบถามข้อมูลการศึกษาของนักศึกษานั้น จะนำคอมพิวเตอร์ของ SAPI เข้ามาใช้งานในส่วนต่างๆ ดังแสดงในบทที่ 3 รูปที่ 3.3 ซึ่งจะมีหลักการพัฒนา ดังนี้

## ส่วนที่ 1 การประกาศตัวแปรที่จำเป็นในการใช้งาน ตามตัวอย่างโค้ด ต่อไปนี้

```
Public WithEvents RC As SpSharedRecoContext
Public Grammar As ISpeechRecoGrammar
Dim fRecoEnabled As Boolean
Dim fGrammarLoaded As Boolean
Dim RecoResult As ISpeechRecoResult
```

โดยที่ RC เป็นกลไกการทำงานหลักในการใช้งานการรู้จำเสียงซึ่งจะมีเหตุการณ์ต่างๆ  
รองรับในการทำงาน

*Grammar* เป็นไวยากรณ์ในการทำการรู้จำเสียง

*fRecoEnabled* เป็นตัวแปรที่ใช้ตรวจสอบการทำงานของกลไกการทำงานหลัก

*fGrammarLoaded* เป็นตัวแปรใช้ตรวจสอบการโหลดข้อมูลของไวยากรณ์

*RecoResult* เป็นผลลัพธ์ที่ได้จากการรู้จำเสียง

ส่วนที่ 2 การโหลดไวยากรณ์ของระบบ ซึ่งจะทำการโหลดไวยากรณ์จากไฟล์ sol.xml  
จากนั้นทำการตั้งค่าสถานะการทำงานของไวยากรณ์เป็น "Inactive" ไว้ก่อน ตามตัวอย่างโค้ดต่อไป  
นี้

```
Private Sub LoadDefaultCnCGrammar()
    ' First load attempt
    On Error Resume Next
    Grammar.CmdLoadFromFile "grammar\sol.xml", SLODynamic
    ' Second load attempt
    If Err Then
        On Error GoTo Err_CFGLoad
        Grammar.CmdLoadFromFile "grammar\sol.xml", SLODynamic
    End If

    Grammar.CmdSetRuleIdState 0, SGDSInactive
    fGrammarLoaded = True
Exit Sub
Err_CFGLoad:
    fGrammarLoaded = False
    MsgBox "Error"
Exit Sub
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ส่วนที่ 3** การเปลี่ยนสถานะการทำงานของไวยากรณ์โดยจะต้องเปลี่ยนจาก "Inactive" เป็น "Active" เพื่อเริ่มการรู้จำเสียง ตามตัวอย่างโค้ด ต่อไปนี้

```
Private Sub Active_Mic_Click()
    On Error GoTo ErrorHandler
    If fGrammarLoaded Then
        ' If recognition is currently not enabled, enable it.
        If Not fRecoEnabled Then
            Grammar.CmdSetRuleIdState 0, SGDSActive
            fRecoEnabled = True
        Else
            ' Deactivate either Dictation or Command and Control.
            Grammar.CmdSetRuleIdState 0, SGDSInactive
            fRecoEnabled = False
        End If
    End If
    Exit Sub
ErrorHandler:
    MsgBox "Failed to activate the grammar. It is possible
        that your audio device is used by other application.",
        vbOKOnly
End Sub
```

**ส่วนที่ 4** การรับเหตุการณ์ต่างๆ การเขียนการทำงานในการรับเหตุการณ์ต่างๆ ที่เกิดขึ้นขณะทำการรู้จำเสียงของระบบ โดยที่เหตุการณ์ต่างๆ ที่สำคัญมีดังนี้

**SRESoundStart** : เมื่อกลไกการทำงานสามารถตรวจการเริ่มต้นของข้อมูลเสียงที่เข้ามาได้

**SRESoundEnd** : เมื่อกลไกการทำงานสามารถตรวจการสิ้นสุดของข้อมูลเสียงที่เข้ามาได้

**SREPhraseStart** : เมื่อกลไกการทำงานสามารถเริ่มตรวจจับข้อความที่สามารถรู้จำได้

**SRERecognition** : เมื่อกลไกการทำงานได้ผลลัพธ์ที่ถูกต้องตามไวยากรณ์ที่กำหนดไว้

**SREHypothesis** : เมื่อกลไกการทำงานได้ข้อความที่น่าจะเป็นไปได้และตั้งสมมติฐานไว้

ซึ่งในการเขียนฟังก์ชันที่รองรับเหตุการณ์ต่างๆ จะเป็นดัง โค้ดตัวอย่าง ต่อไปนี้ โดยจะยกตัวอย่างมาเฉพาะเหตุการณ์ **SRERecognition** โดยการนำผลลัพธ์ที่ได้ใส่ไว้ใน **RecoResult** ซึ่งเป็นตัวแปรที่ใช้ในการเก็บผลลัพธ์จากการรู้จำเสียง

```
Private Sub RC_Recognition(ByVal StreamNumber As Long, ByVal StreamPosition As Variant, ByVal RecognitionType As SpeechLib.SpeechRecognitionType, ByVal Result As SpeechLib.ISpeechRecoResult) Set RecoResult = Result
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 5 การเชื่อมต่อฐานข้อมูล ในการเชื่อมต่อฐานข้อมูลจะใช้การทำงานในส่วน  
ของ ADO ของ Visual Basic 6.0 ในการสร้างการเชื่อมต่อและทำการสอบถามข้อมูลจากฐานข้อมูล  
โดยการส่งคำสั่งของ *SQL Statement* เพื่อทำการสอบถามข้อมูลจากฐานข้อมูล ซึ่งจะให้ผลลัพธ์  
ออกมาเป็น *RecordSet* ดังโค้ดตัวอย่าง ต่อไปนี้

```
Public rs As ADODB.recordSet
Public i As Integer
Public Function execSQL(sqlString As String) As ADODB.recordSet
    Dim recordSet As ADODB.recordSet
        connString = "DRIVER={MySQL ODBC 3.51
Driver};SERVER=localhost;UID=root;PWD=;DATABASE=SR;"

    Set conn = New ADODB.Connection
    conn.ConnectionString = connString
    conn.CursorLocation = adUseClient
    conn.Open

    Set recordSet = New ADODB.recordSet
    recordSet.Open sqlString, conn
    Set execSQL = recordSet
End Function
```

#### 4.1.2 ส่วนการฝึกสอนระบบ

ส่วนฝึกสอนระบบต้องใช้คอมโพเนนต์ *Recognizer* และคอมโพเนนต์ *DisplayUI* เป็น  
หลัก คอมโพเนนต์ *Recognizer* เป็นคอมโพเนนต์ในการสร้างส่วนรับเสียง ตัดคำ และการทำให้  
เป็น *Phoneme* คอมโพเนนต์ *DisplayUI* เป็นคอมโพเนนต์ของ SAPI ที่มีไว้เพื่อการเพิ่มเสียงเข้า  
ไปใน *Registry*

SAPI มีตัวอย่างการใช้งานคอมโพเนนต์ *Recognizer* และคอมโพเนนต์ *DisplayUI* เป็น  
Visual C/C++ อยู่แล้วจึงง่ายต่อการศึกษาและพัฒนาาระบบการฝึกสอน เพราะฉะนั้นจึงทำให้  
Visual C/C++ ในการพัฒนาส่วนนี้

ในการพัฒนาส่วนการฝึกสอนนี้ จะใช้คอมโพเนนต์ *DisplayUI* ที่ได้กล่าวไว้ข้างต้น ซึ่ง  
คอมโพเนนต์นี้จะทำหน้าที่ทั้งรับเสียงและนำ "คำ" ที่ได้ใส่ลงในฐานข้อมูลเสียง

```

void CSR_SPtrainingDlg::OnAddDelete()
{
    HRESULT hr;

    // Get the word nearest the beginning of the selection
    // If these calls fail, we just won't get the word.

    WCHAR *pwszNewWord = NULL;

    ULONG ulDataSize = pwszNewWord ?
        (sizeof(WCHAR) * wcslen( pwszNewWord )) : 0;

    USES_CONVERSION;
    hr = g_cpEngine->DisplayUI( m_hWnd, NULL, SPDUI AddRemoveWord,
pwszNewWord, ulDataSize );

    if ( pwszNewWord )
    {
        free( pwszNewWord );
    }
}

```

#### 4.1.3 ส่วนการสร้างไวยากรณ์

ในการสร้างไวยากรณ์สำหรับระบบการสอบถามข้อมูลจากฐานข้อมูลนั้น ต้องทำการสร้างไวยากรณ์ในรูปแบบของไฟล์ XML ที่กำหนดใน SAPI โดยการสร้างไวยากรณ์นั้นสามารถใช้โปรแกรมสร้างไฟล์ข้อความ(Text Editor) ทั่วไปได้ เช่น Notepad หรือ Editplus เป็นต้น โดยรูปแบบการสร้างไวยากรณ์จะเป็นดังที่กล่าวมาในบทที่แล้ว ซึ่งไวยากรณ์ที่ใช้ในการทำระบบการสอบถามข้อมูลผลการศึกษานักศึกษาในปัญหาพิเศษนี้จะแสดงได้ดังต่อไปนี้

```

<Grammar LANGID="409">
  <DEFINE>
    <ID NAME="rank" VAL="5"/>
  </DEFINE>
  <RULE NAME="rank" TOPLEVEL="ACTIVE">
    <O DISP="0">_JUTtest00</O>
    <O DISP="1">_JUTtest11</O>
    <O DISP="2">_JUTtest22</O>
    <O DISP="3">_JUTtest33</O>
    <O DISP="4">_JUTtest44</O>
    <O DISP="5">_JUTtest55</O>
    <O DISP="6">_JUTtest66</O>
    <O DISP="7">_JUTtest77</O>
    <O DISP="8">_JUTtest88</O>
    <O DISP="9">_JUTtest99</O>
    <O DISP="ok">_JUTtestok</O>
    <O DISP="tab">_JUTtesttab</O>
    <O DISP="again">_JUTtestagain</O>
    <O DISP="next">_JUTtestnext</O>
    <O DISP="prev">_JUTtestprev</O>
    <O DISP="end">_JUTtestend</O>
  </RULE>
</Grammar>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 การดำเนินงานของระบบ

### 4.2.1 ภาพรวมของระบบ

ลักษณะการทำงานของระบบจะถูกแบ่งออกเป็น 2 ส่วนด้วยกัน คือ “การฝึกหัดระบบกับการเตรียมไวยากรณ์” และ “การทำงานของโปรแกรมตัวอย่าง” ซึ่งขั้นตอนการทำงาน มีดังนี้

#### 4.2.1.1 ผู้ใช้ระบบทำการฝึกหัดระบบ ซึ่งมีขั้นตอนในการฝึกหัดระบบอยู่ในหัวข้อ 4.2.2.1

คำต่างๆ ที่ใช้ใน โปรแกรมตัวอย่าง มีดังนี้

- ศูนย์ (ศูนย์)	หมายถึง	0
- หนึ่ง (หนึ่ง)	หมายถึง	1
- สอง (สอง)	หมายถึง	2
- สาม (สาม)	หมายถึง	3
- สี่ (สี่)	หมายถึง	4
- ห้า (ห้า)	หมายถึง	5
- หก (หก)	หมายถึง	6
- เจ็ด (เจ็ด)	หมายถึง	7
- แปด (แปด)	หมายถึง	8
- เก้า (เก้า)	หมายถึง	9
- เลื่อน (เลื่อน)	หมายถึง	เลื่อนตำแหน่งในการกรอกข้อมูล
- ตรวจสอบ (ตรวจ)	หมายถึง	การตรวจสอบผลการศึกษา
- ถัดไป (ต่อ-ไป)	หมายถึง	การดูผลการศึกษาในปีการศึกษาถัดไป
- ก่อนหน้า (ก่อน-หน้า)	หมายถึง	การดูผลการศึกษาในปีการศึกษาก่อน
- อีกครั้ง (อีก-ครั้ง)	หมายถึง	การป้อนข้อมูลใหม่
- จบ (จบ)	หมายถึง	จบการทำงานของระบบ

#### 4.2.1.2 ผู้ใช้นำข้อมูลเสียงจากฐานข้อมูลมาสร้างไวยากรณ์ ในรูปแบบไฟล์ sol.xml

#### 4.2.1.3 นำไวยากรณ์ที่ได้ ไปใช้ใน โปรแกรมตัวอย่าง

#### 4.2.1.4 ในส่วนของโปรแกรมตัวอย่างจะมีการทำงานอยู่ในหัวข้อ 4.2.2.2

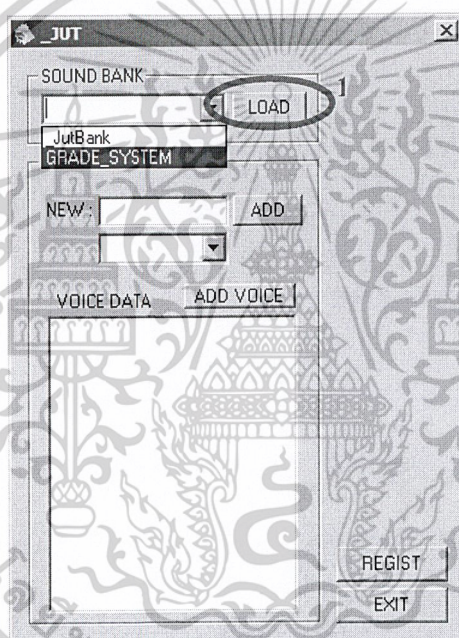
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ ผู้ใช้สามารถป้อนข้อมูลแบบปกติโดยไม่ต้องใช้การพูดได้ และ เนื่องจากข้อมูลการฝึกระบบยังมีน้อยและการออกเสียงของแต่ละคนนั้นไม่เหมือนกัน ทำให้ผลการทำงานอาจเกิดข้อผิดพลาดได้

## 4.2.2 การใช้งานระบบ

### 4.2.2.1 ขั้นตอนการใช้งานส่วนการฝึกสอนระบบ

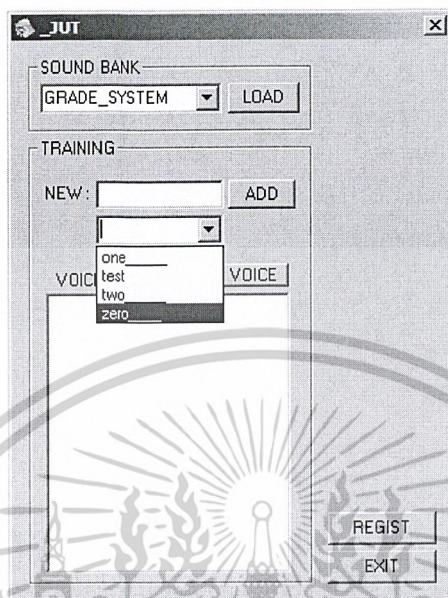
1. เมื่อเข้าใช้งานส่วนของการฝึกสอนระบบ เริ่มแรกจะต้องทำการเลือกกลุ่มคำที่มีเพื่อการจัดเก็บเสียงตัวอย่าง โดยเริ่มจากการคลิกปุ่ม LOAD (หมายเลข 1) ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 แสดงส่วนการเลือกกลุ่มคำที่มีเพื่อทำการจัดเก็บเสียงตัวอย่าง

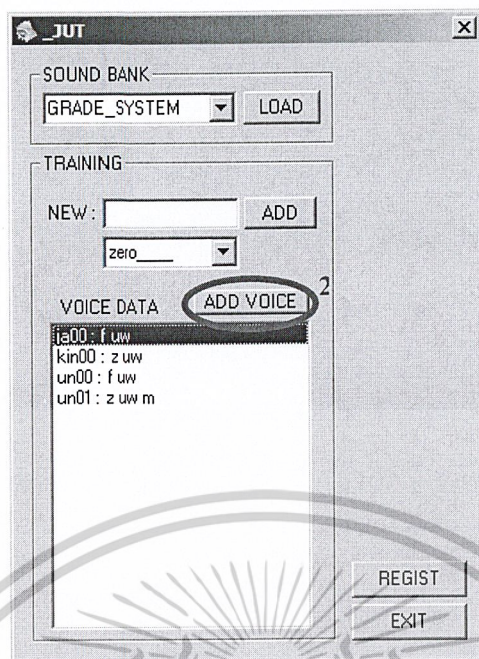
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากนั้นเมื่อทำการเลือกกลุ่มคำที่ต้องการ จะสามารถตรวจสอบได้ว่ามีเสียงใดบ้างที่มีการจัดเก็บไว้แล้ว ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 แสดงเสียงที่มีการจัดเก็บไว้แล้วในกลุ่มคำที่เลือก

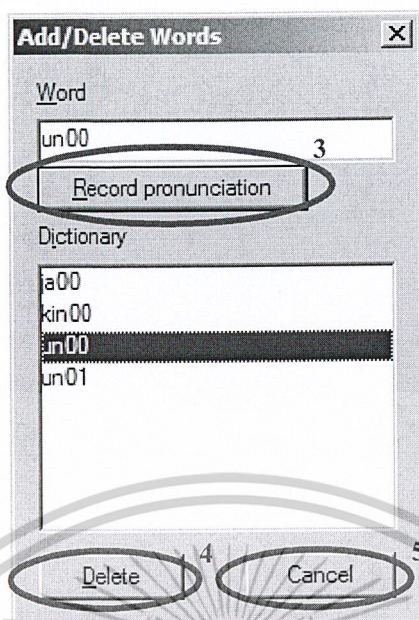
3. เมื่อต้องการเพิ่มเสียงเข้าไปยัง “คำ” ใดๆ สามารถเลือกจากคอรอบคาวน์ลิสต์ได้ เมื่อทำการเลือก “คำ” ที่ต้องการจัดเก็บได้แล้ว ตัวโปรแกรมจะแสดงข้อมูลเสียงของแต่ละคนที่ได้มีการจัดเก็บไว้แล้วในฐานข้อมูลออกมา ดังแสดงในรูปที่ 4.2 ซึ่งสามารถฟังเสียงที่มีการเก็บไว้ในฐานข้อมูลเสียงของแต่ละคนได้โดยทำการคลิกเลือกจากส่วนที่แสดงขึ้นมาในรูปที่ 4.3



รูปที่ 4.3 แสดงเสียงของแต่ละคนที่ได้ทำการจัดเก็บไว้ในฐานข้อมูลเสียง

จากนั้นเมื่อต้องการเก็บเสียงจากคำที่เลือกไว้เพิ่มขึ้นจากเดิม ด้วยเสียงของผู้เทรน (trainee) คนใหม่ ก็จะมีการเพิ่มเสียงโดยเมื่อคลิก ADD VOICE (หมายเลข 2) จะปรากฏหน้าจอ ดังแสดงในรูปที่ 4.4 ซึ่งหน้าจอนี้จะต้องทำการกำหนดชื่อของผู้เทรนลงในช่องคำ (WORD) จากนั้นคลิก Record pronunciation (หมายเลข 3) จะทำการเก็บเสียงผู้พูดจากไมโครโฟน และทำการแสดงผลของเสียงที่ทำการเก็บไว้ในส่วนของ Dictionary เพื่อสามารถเลือกฟังเสียงที่ทำการเทรน ลงในระบบ ซึ่งสามารถทำการลบเสียงที่เทรนไว้แล้วได้ โดยคลิก Delete (หมายเลข 4) และสามารถปิดส่วนการรับเสียงเพิ่มจากการเทรนได้โดยคลิก Cancel (หมายเลข 5)

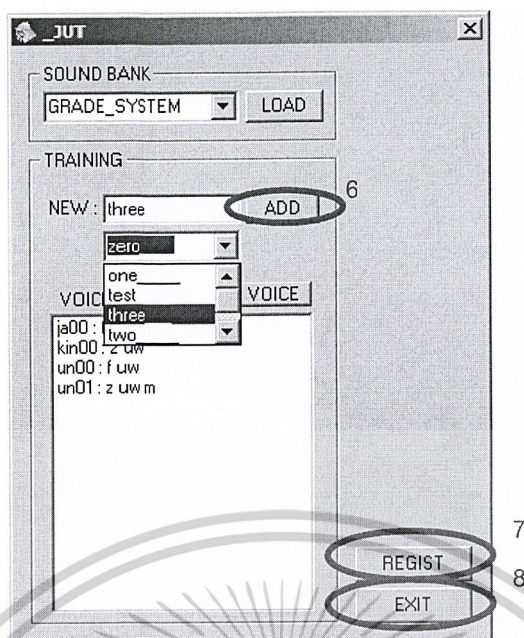
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงส่วนของการเก็บเสียงจากผู้พูด

เมื่อต้องการให้มีการเทรนเสียง "คำ" ใหม่เพื่อใช้ในระบบ จะสามารถทำการเพิ่ม "คำ" ได้โดยคลิก ADD (หมายเลข 6) เพื่อกำหนด "คำ" ที่ต้องการทำการเทรนเสียงลงในฐานข้อมูล ดังแสดงในรูปที่ 4.5 และสามารถทำการเทรนเสียงได้ตามขั้นตอนที่กล่าวมาแล้วข้างต้นได้

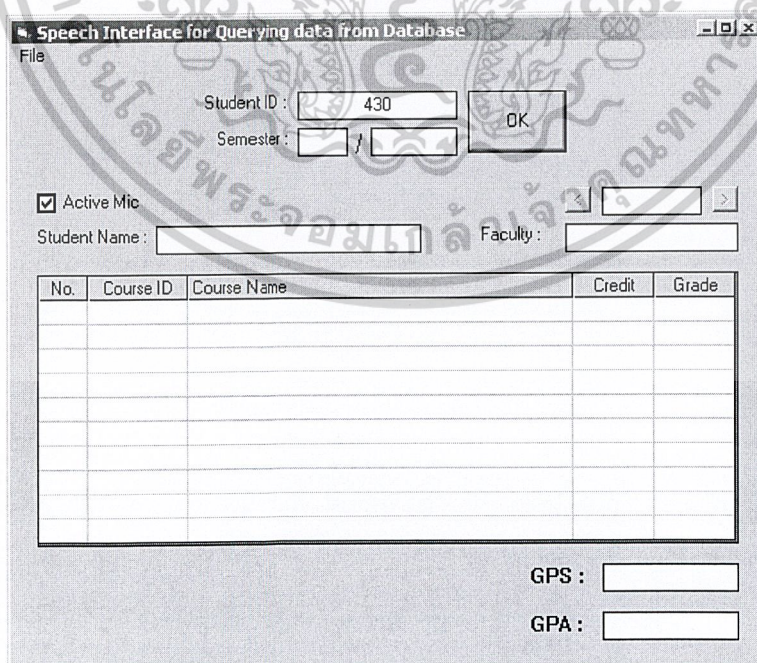
4.เมื่อทำการเทรนระบบด้วยเสียงจนเสร็จสิ้น จะต้องทำการนำเสียงที่เทรนเก็บลงในฐานข้อมูลเสียงที่มีการเก็บข้อมูลจริง โดยทำการคลิก REGIST (หมายเลข 7) เพื่อส่งข้อมูลเสียงลงฐานข้อมูล และปิดโปรแกรมการเทรนระบบด้วยเสียงด้วยการคลิกโดยคลิก EXIT (หมายเลข 8)



รูปที่ 4.5 แสดงการเก็บเสียงที่ได้รับการทรนกลงในฐานข้อมูลเสียงตัวอย่าง

#### 4.2.2.1 ขั้นตอนการใช้งานส่วนการฝึกสอนระบบ

- เมื่อต้องการให้ระบบเริ่มการรู้จำเสียง ให้คลิกที่  Active Mic แล้วรูปจะเปลี่ยนเป็น  Active Mic เมื่อเปลี่ยนรูปแล้วสามารถพูดตัวเลขในช่อง Student ID ได้ ซึ่งเป็นช่องเดียวที่จำเป็นต้องใส่ค่า ไม่ว่าจะจากไมโครโฟน หรือ จากคีย์บอร์ด



รูปที่ 4.6 การเริ่มการรู้จำเสียงและเริ่มใส่ค่าด้วยคำพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อต้องการใส่ข้อมูลลงในช่องอื่น ให้พูดคำว่า “เลื่อน” แล้วเคอร์เซอร์ จะเลื่อน

ไปช่องอื่น

รูปที่ 4.7 แสดงการสั่ง “เลื่อน” เพื่อให้ เคอร์เซอร์ ไปยังช่องถัดไป

3. เมื่อคิดว่าข้อมูลที่ให้ระบบเพียงพอแล้ว ( ปกติใส่ StudentID ก็เพียงพอแล้ว ) ให้พูดคำว่า “ตกลง” แล้วระบบ จะ แสดงผลการสอบถามกลับมาให้เรา

รูปที่ 4.8 แสดงหน้าจอหลังจากการออกคำสั่งจาก “เลื่อน หรือ ตกลง”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผลประสิทธิภาพการทำงานของระบบ

ลักษณะของระบบที่พัฒนาขึ้น เป็นการสร้างอินเทอร์เน็ตเพื่อรับคำสั่งการทำงานด้วยเสียง โดยจะรับข้อมูลเข้าที่เป็นเสียงพูดของผู้ใช้งานระบบผ่านไมโครโฟน โดยทำการป้อนข้อมูลต่างๆ ด้วยเสียง ในการสอบถามข้อมูลผลการเรียน ระบบจะทำการตรวจสอบข้อมูลเสียงที่ป้อนเข้ามากับข้อมูลเสียงที่เก็บไว้ในฐานข้อมูลเสียง และตรวจสอบกับไวยากรณ์ว่ามีความถูกต้องในการตั้งงาน และแสดงผลของข้อมูลเสียงที่ป้อนเข้าไปอย่างถูกต้อง ในส่วนของการประมวลผลเมื่อทำการสอบถามข้อมูล สามารถติดต่อกับฐานข้อมูล ค้นหาและนำมาแสดงผลได้อย่างถูกต้อง

#### 5.2 ข้อจำกัดของระบบที่พัฒนา

1. จำนวนเสียงที่เป็นข้อมูลเข้าของระบบมีจำนวนที่แน่นอน ตามรูปแบบการสั่งงานที่เป็นไปได้ทั้งหมด โดยข้อมูลเสียงที่เข้ามาในระบบจะต้องเป็นคำที่กำหนดอยู่ใน ไวยากรณ์เท่านั้น
2. รูปแบบของคำสั่งในการสอบถามข้อมูลจำกัดอยู่เพียงแค่การสอบถามข้อมูลเท่านั้น ไม่สามารถกำหนดให้ระบบทำคำสั่งอื่นๆ ที่ทำงานกับฐานข้อมูลได้

#### 5.3 ข้อเสนอแนะ

1. เพิ่มจำนวนข้อมูลเสียงที่เข้ามาในระบบให้มากขึ้น เพื่อให้คำสั่งในการทำงานของระบบมีความหลากหลายมากขึ้น
2. รูปแบบของคำสั่งในการสอบถามข้อมูลควรจะสามารถทำงานร่วมกับฐานข้อมูลได้หลายอย่าง เช่น สามารถทำการแก้ไขข้อมูล หรือลบข้อมูลได้ เป็นต้น โดยไม่จำเป็นที่ผู้ใช้จะต้องเข้าใจคำสั่งของการทำงานกับฐานข้อมูลได้ เพียงแค่ใช้เสียงสั่งงานแทนก็พอ

## บรรณานุกรม

ยุทธนา ตีลาศวัฒน์กุล. 2544. **คู่มือการเขียนโปรแกรมและใช้งาน Visual C++ 6.0 ฉบับโปรแกรมเมอร์**. พิมพ์ครั้งที่ 1. นนทบุรี : อินโฟเพรส.

ยุทธนา ตีลาศวัฒน์กุล. **คู่มือการเขียนโปรแกรม Microsoft, Advanced Visual C++ Version 6.0 ฉบับ Database Programming**. กรุงเทพฯ : ชัคเซสมิเดีย.

Carnegie Mellon University. 2004. **CBS Overview**. [online]. Available :

<http://www.sei.cmu.edu/cbs/overview.html>

Carnegie Mellon University. 2004. **Component-Based Software Development/COTS**

**Integration , Purpose and Origin**. [online]. Available :

<http://www.sei.cmu.edu/cbs/overview.html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบปฏิบัติการที่ต้องการ

Microsoft Windows 98 / Me / 2000 / XP

## ซอฟต์แวร์ที่ต้องการ

Microsoft Speech SDK version 5.1

MySQL Server and Clients version 3.23.49

MySQL Connector / ODBC version 3.51.06

โปรแกรมการฝึกเสียง

โปรแกรมระบบการสอบถามข้อมูลผลการเรียนของนักศึกษา

## ฮาร์ดแวร์ที่ต้องการ

Pentium III 533 MHz หรือสูงกว่า

หน่วยความจำ 128 MB

เนื้อที่บนฮาร์ดดิสก์ 10 MB หรือสูงกว่า

การ์ดแสดงผล 8 MB หรือสูงกว่า

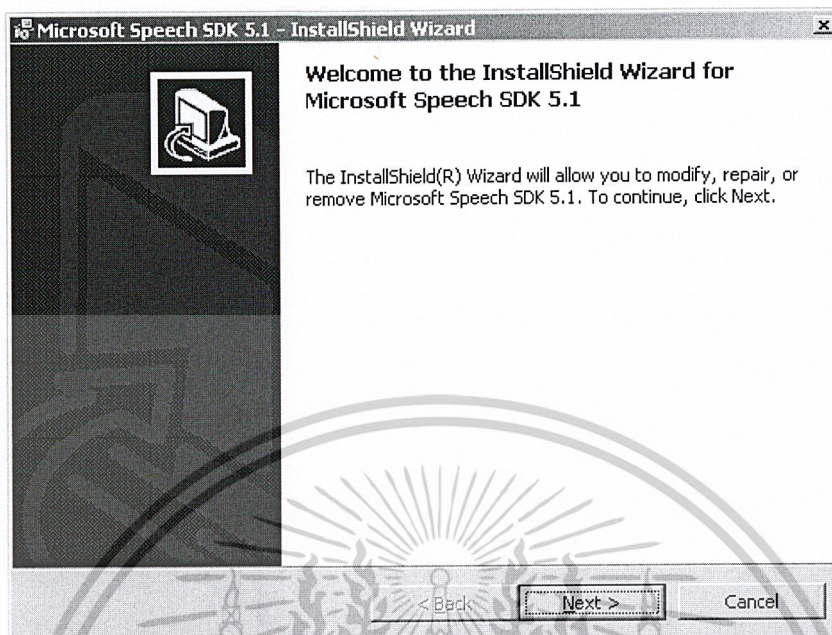
การ์ดเสียง ที่สามารถเชื่อมต่อไมโครโฟนได้

ไมโครโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การติดตั้ง Microsoft Speech SDK version 5.1

1. ดับเบิลคลิกไฟล์ setup.exe ในโฟลเดอร์ SpeechSDK51 จะปรากฏหน้าต่างดังรูป ก.1



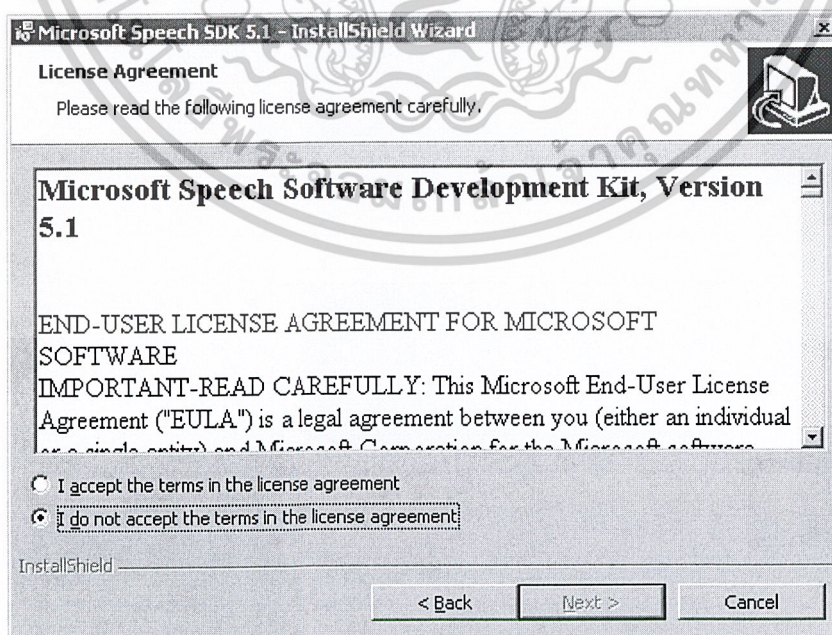
รูปที่ ก.1 หน้าจอการติดตั้ง Microsoft Speech SDK version 5.1

2. คลิก Next ไปเรื่อยๆ จะพบหน้าต่างดังรูป ก.2 , ก.3 และ ก.4

โดยที่ ก.2 ทำการเลือก I agree the terms in the license agreement แล้วคลิก Next

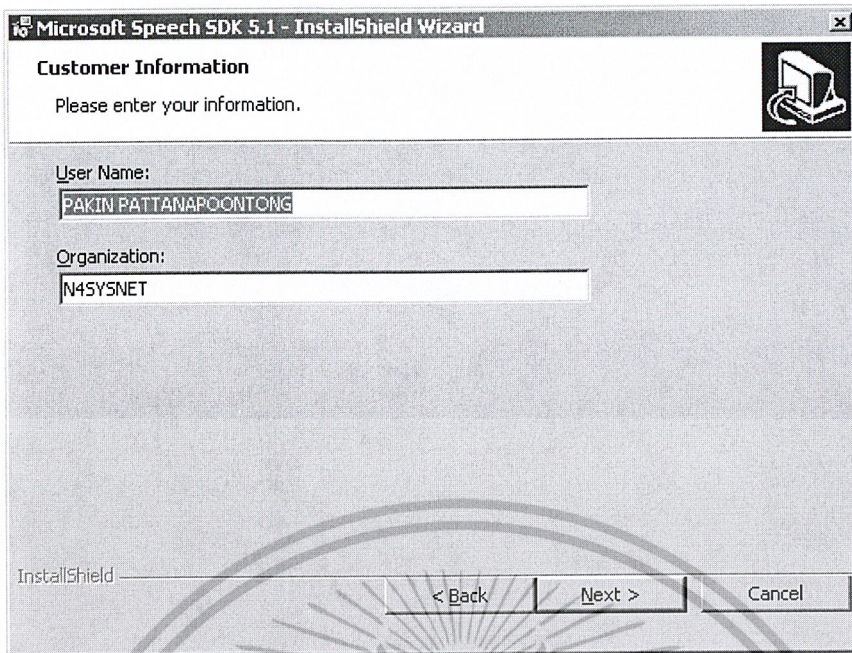
ก.3 ทำการใส่ชื่อและชื่อขององค์กร ซึ่งจะใส่อะไรก็ได้

ก.4 ทำการเลือกตำแหน่งในการติดตั้ง

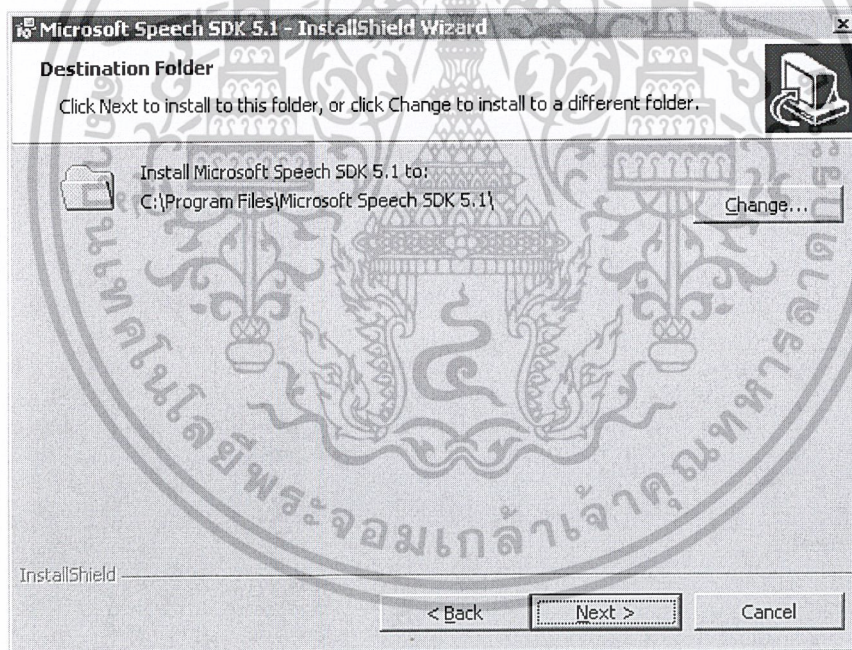


รูปที่ ก.2 หน้าจอแสดงรายละเอียดเกี่ยวกับลิขสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



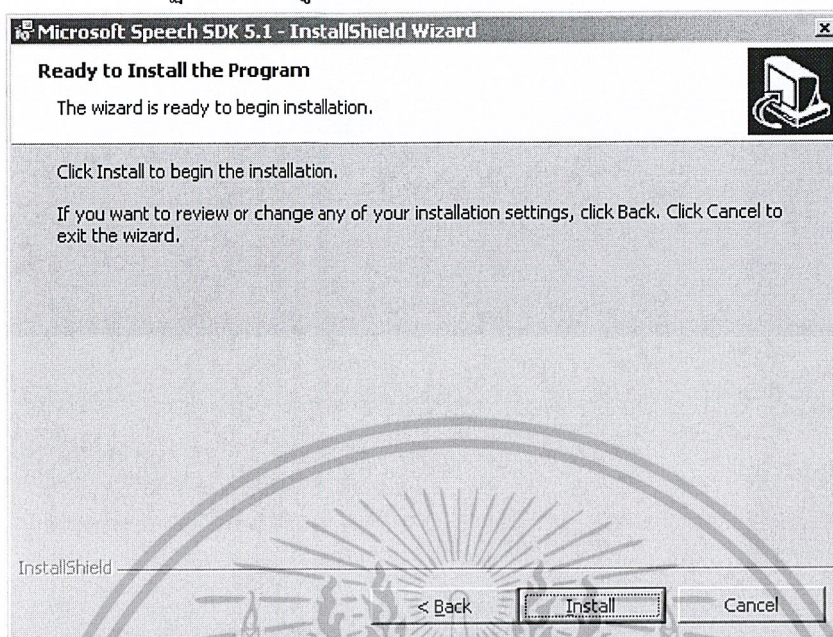
รูปที่ ก.3 หน้าจอแสดงชื่อและชื่อองค์กร



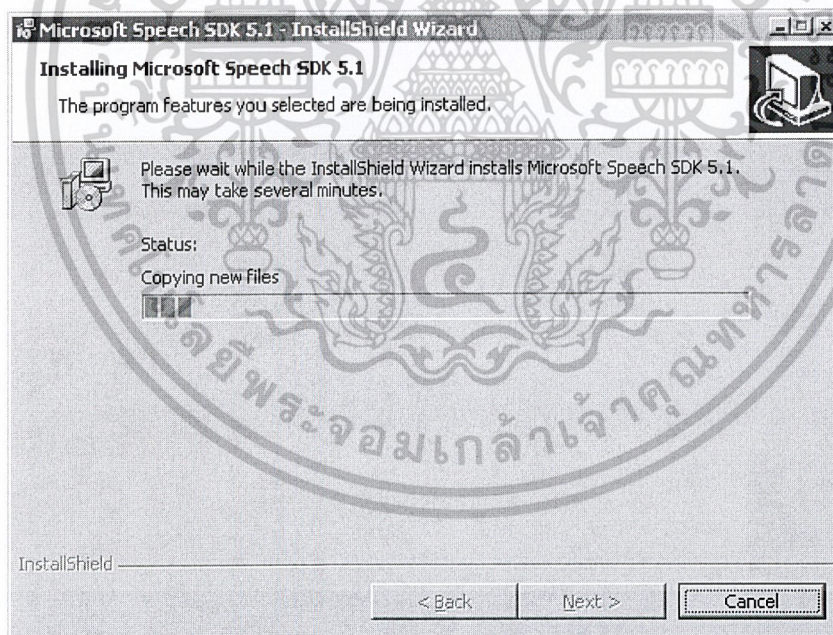
รูปที่ ก.4 หน้าจอแสดงตำแหน่งในการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อกำหนดการติดตั้งแล้วจะปรากฏหน้าจอตั้งรูป ก.5 แล้วทำการคลิก Install เพื่อทำการเริ่มการติดตั้ง โปรแกรม จะปรากฏหน้าจอตั้งรูปที่ ก.6



รูปที่ ก.5 คลิก Install เพื่อเริ่มการติดตั้ง



รูปที่ ก.6 หน้าจอแสดงการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อติดตั้งเสร็จเรียบร้อยแล้วจะปรากฏหน้าจอจดังรูป ก.7 ทำการคลิก Finish เพื่อสิ้นสุดการติดตั้ง

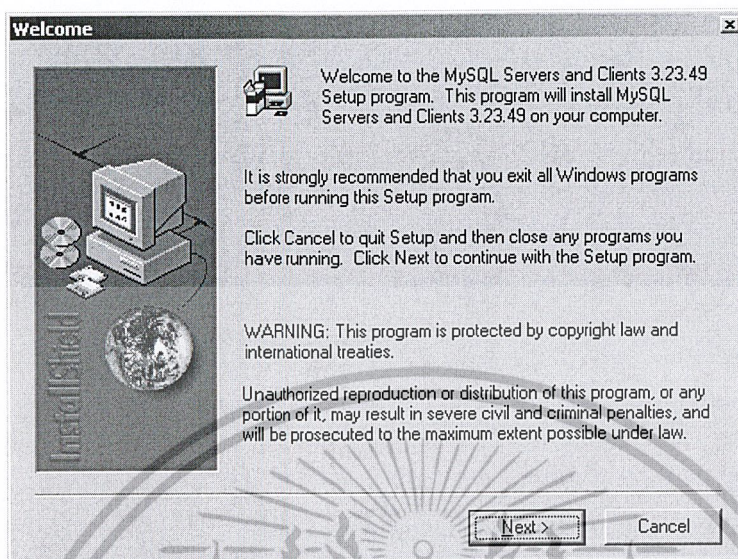


รูปที่ ก.7 สิ้นสุดการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การติดตั้ง MySQL Server and Clients version 3.23.49

1. ดับเบิลคลิกที่ไฟล์ SETUP.EXE ในโฟลเดอร์ mysql-2.23.49-win จะปรากฏหน้าต่างดังรูปที่ ก.8



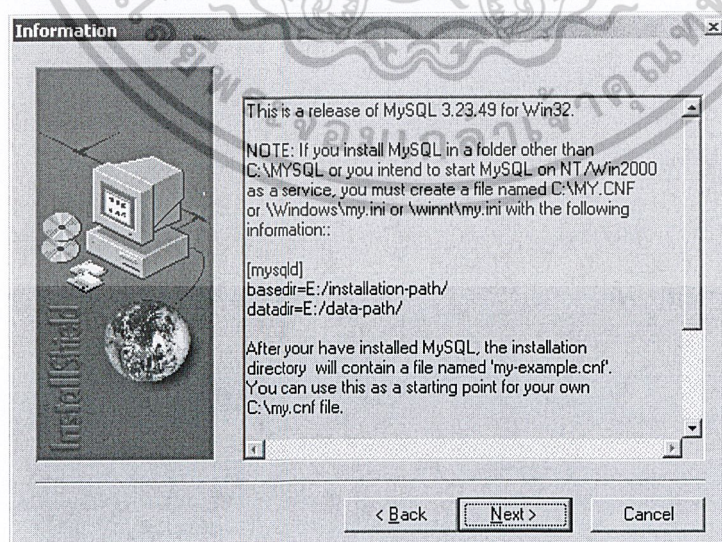
รูปที่ ก.8 หน้าจอการติดตั้ง MySQL Server and Clients version 3.23.49

2. คลิก Next ไปเรื่อยจะพบหน้าจอดังรูปที่ ก.9 , ก.10 , ก.11 และ ก.12

โดยที่ ก.10 เป็นการเลือกตำแหน่งที่ต้องการติดตั้ง โดยสามารถเลือกได้โดยการกดปุ่ม Browse

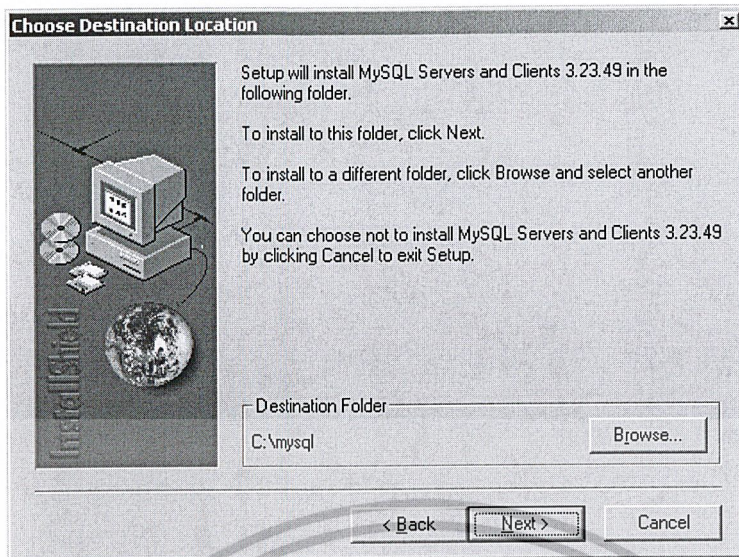
ก.11 เป็นการเลือกการรูปแบบติดตั้ง โดยในที่นี้จะเลือกแบบ Typical และเมื่อกด Next ก็จะทำกรติดตั้งโปรแกรม

ก.12 สิ้นสุดการติดตั้ง

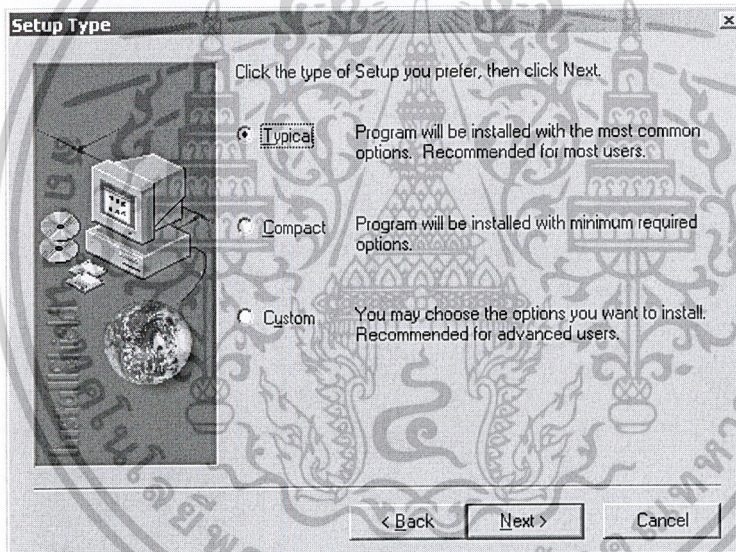


รูปที่ ก.9 หน้าจอการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

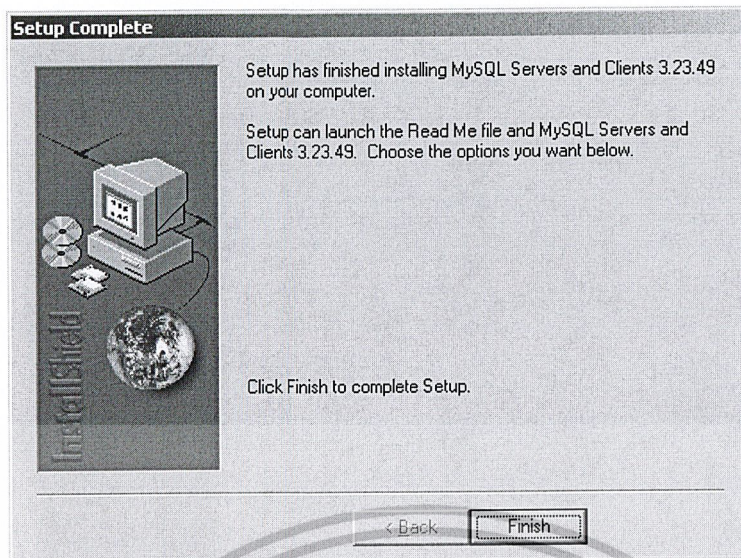


รูปที่ ก.10 การเลือกตำแหน่งในการติดตั้ง



รูปที่ ก.11 การเลือกรูปแบบการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.12 สิ้นสุดการติดตั้ง

3. ทำการติดตั้ง Service ของ MySQL ดังรูปที่ ก.13 และ ก.14

โดยที่ ก.13 ทำการเข้าไปที่โฟลเดอร์ bin ของ MySQL

ก.14 ทำการติดตั้ง Service โดยใช้คำสั่ง `mysqld-nt -install`



รูปที่ ก.13 เข้าไปที่โฟลเดอร์ bin ของ MySQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

C:\WINNT\system32\cmd.exe
C:\>cd mysql
C:\mysql>cd bin
C:\mysql\bin>mysqld-nt --install
Service successfully installed.
C:\mysql\bin>_

```

รูปที่ ก.14 ติดตั้ง Service

4. ทำการรัน Service ของ MySQL โดยเข้าไปที่ Start > Setting > Control panel > Services จากนั้นหา Service ชื่อ MySQL แล้ว Start Service ดังรูป ก.15 หรือทำการ Restart วินโดวส์

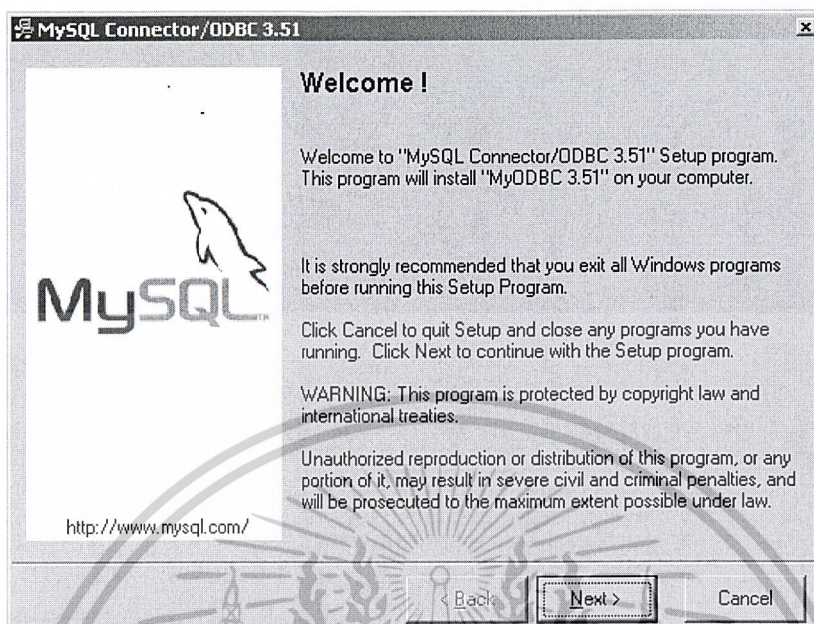
Name	Description	Status	Startup Type	Log On As
IPSEC Policy Agent	Manages I...	Started	Automatic	LocalSystem
Kerberos Key Distri...	Generates ...	Disabled	Automatic	LocalSystem
License Logging Ser...		Started	Automatic	LocalSystem
Logical Disk Manager	Logical Disk...	Started	Automatic	LocalSystem
Logical Disk Manage...	Administrat...	Manual	Automatic	LocalSystem
Machine Debug Man...	Supports lo...	Started	Automatic	LocalSystem
Messenger	Sends and ...	Started	Automatic	LocalSystem
Microsoft Search	Creates ful...	Started	Automatic	LocalSystem
MSSQLSERVER		Manual	Automatic	LocalSystem
MSSQLServerADHel...		Manual	Automatic	LocalSystem
MySQL		Started	Automatic	LocalSystem
Net Logon	Supports p...	Manual	Automatic	LocalSystem
NetMeeting Remote...	Allows aut...	Manual	Automatic	LocalSystem
Network Connections	Manages o...	Started	Manual	LocalSystem
Network DDE	Provides n...	Manual	Automatic	LocalSystem
Network DDE DSDM	Manages s...	Manual	Automatic	LocalSystem
NT LM Security Sup...	Provides s...	Started	Manual	LocalSystem
NVIDIA Driver Help...		Started	Automatic	LocalSystem
OracleOraHome81C...		Manual	Automatic	LocalSystem
Performance Logs a...	Configures...	Manual	Automatic	LocalSystem
Plug and Play	Manages d...	Started	Automatic	LocalSystem

รูปที่ ก.15 Service ของ MySQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

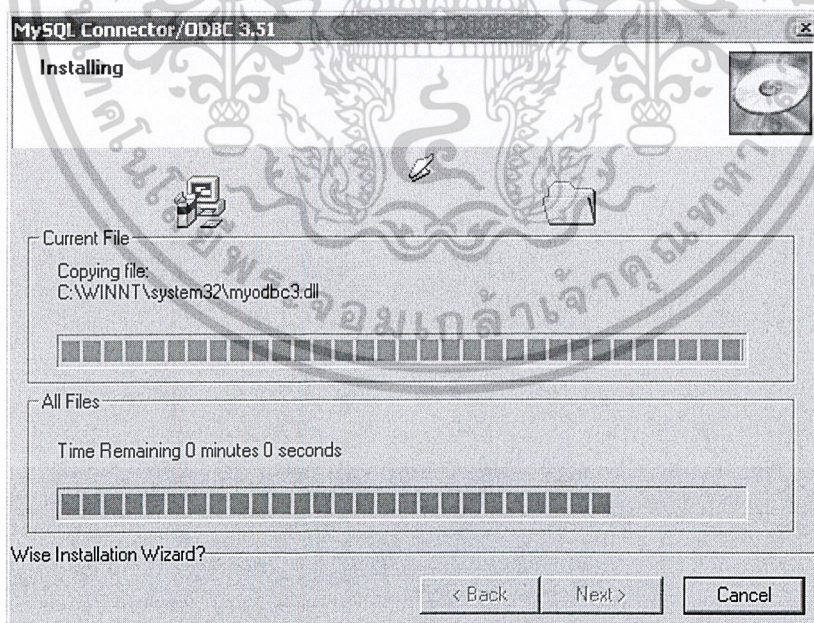
## การติดตั้ง MySQL Connector / ODBC version 3.51.06

1. ทำการดับเบิลคลิกไฟล์ MyODBC-3.51.06.exe จะปรากฏหน้าจอตั้งรูป ก.16



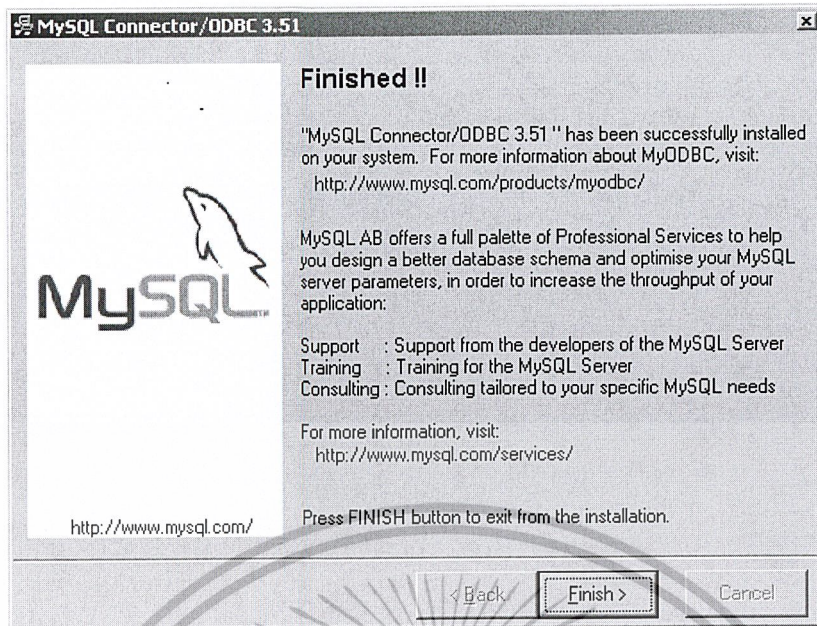
รูปที่ ก.16 หน้าจอการติดตั้ง MySQL Connector / ODBC version 3.51.06

2. คลิก Next ไปเรื่อยๆ ก็จะทำการติดตั้ง ตามรูปที่ ก.17 เมื่อติดตั้งเรียบร้อยแล้วจะปรากฏหน้าจอตั้งรูปที่ ก.18 จากนั้นคลิก Finish เพื่อสิ้นสุดการติดตั้ง



รูปที่ ก.17 หน้าจอแสดงการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

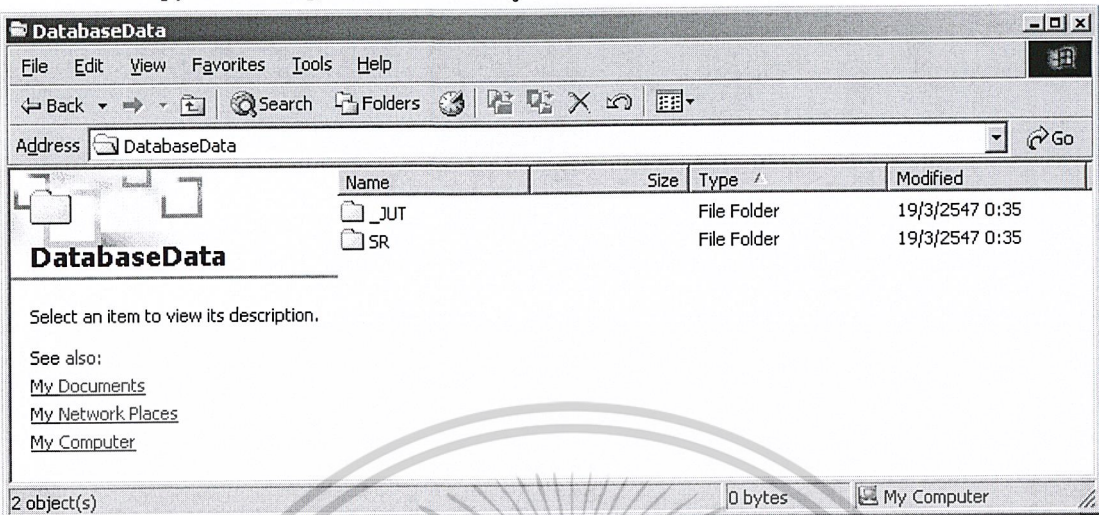


รูปที่ ก.18 หน้าจอสิ้นสุดการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

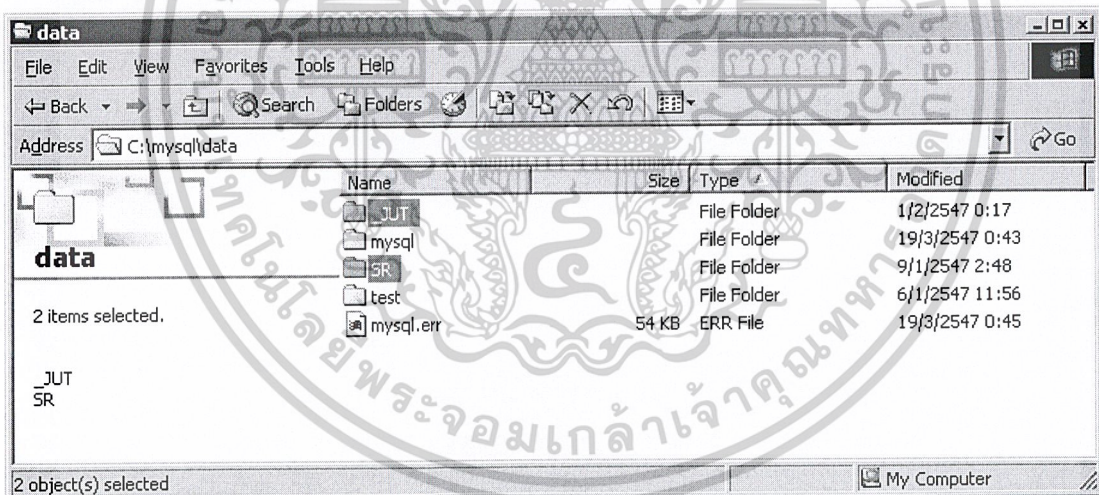
## การติดตั้งฐานข้อมูลที่ใช้ในการทำปัญหาพิเศษนี้

1. ทำการ Copy โฟลเดอร์ \_JUT และ SR ที่อยู่ในโฟลเดอร์ Database Data ดังรูป ก.19



รูปที่ ก.19 โฟลเดอร์ Database Data ต้องทำการ Copy \_JUT และ SR

2. ทำการนำโฟลเดอร์ที่ Copy มาไป Paste ไว้ที่โฟลเดอร์ data ของ MySQL (ในที่นี้คือ C:\mysql\data\ ) ดังรูป ก.20

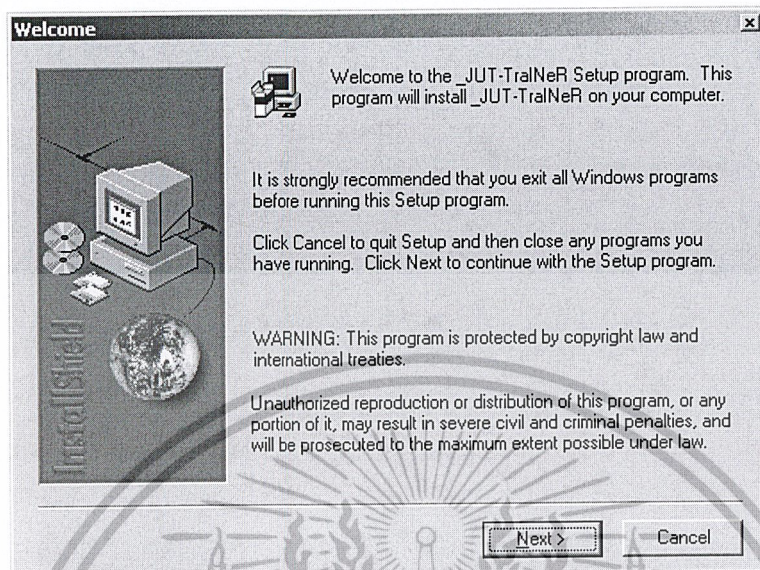


รูปที่ ก.20 ทำการ Paste ข้อมูลที่ Copy มาใน C:\mysql\data\

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

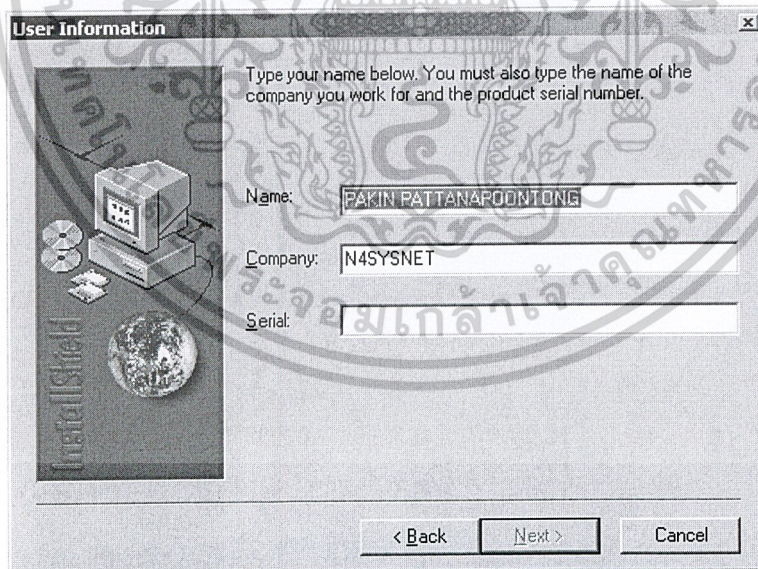
## การติดตั้งโปรแกรมการฝึกเสียง

1. ทำการดับเบิลคลิกไฟล์ setup.exe ในโฟลเดอร์ \_JUT-TraiNeR จะปรากฏหน้าจอตั้งรูป ก.21



รูปที่ ก.21 หน้าจอการติดตั้ง โปรแกรมการฝึกเสียง

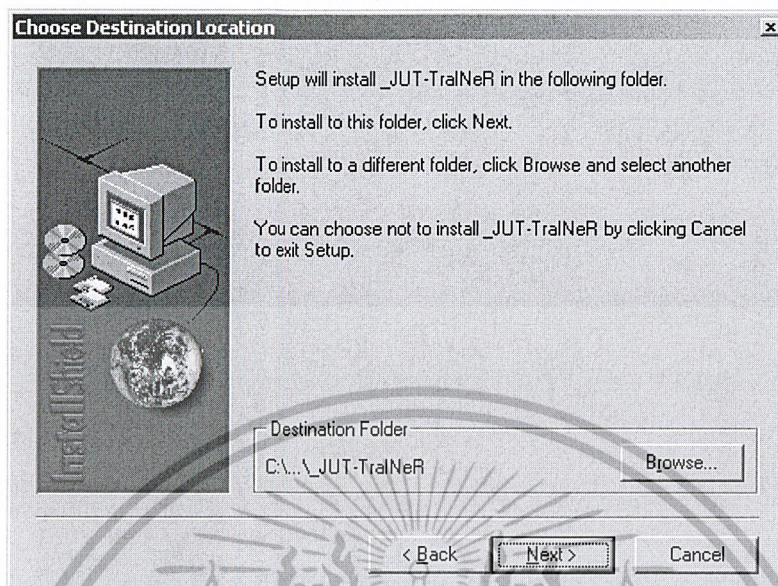
2. คลิก Next จะพบกับหน้าจอใส่รายละเอียดของผู้ใช้ ดังรูป ก.22 ใส่ Serial จากนั้นคลิก Next โดยที่ Serial number คือ \_Jut-1234



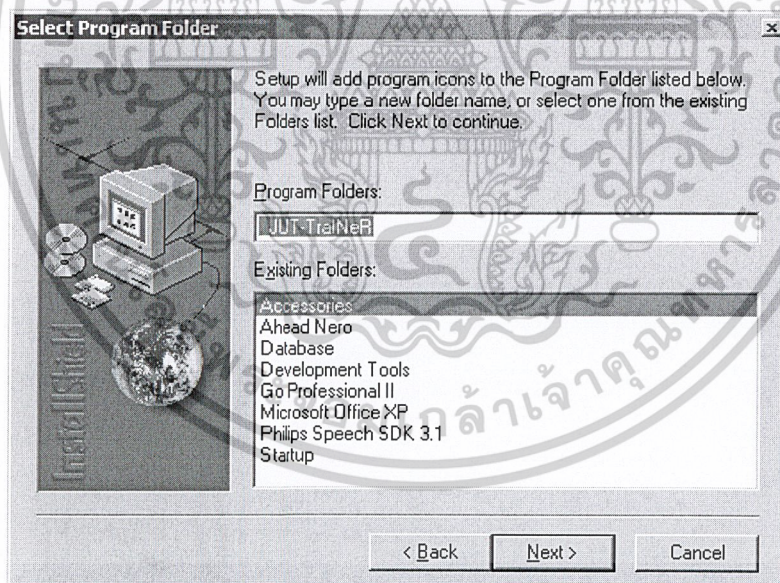
รูปที่ ก.22 หน้าจอใส่รายละเอียดของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จะพบหน้าจอจดังรูป ก.23 ให้ทำการเลือกตำแหน่งในการติดตั้ง และ รูป ก.24 ทำการสร้างเมนูที่ Start เมนู



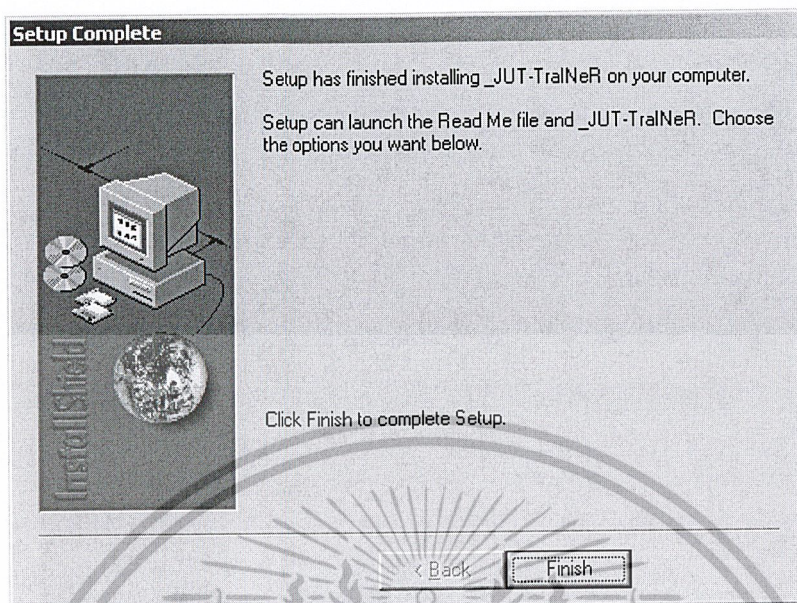
รูปที่ ก.23 ทำการเลือกตำแหน่งในการติดตั้ง



รูปที่ ก.24 ทำการสร้างเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อติดตั้งเสร็จเรียบร้อยแล้วจะปรากฏหน้าจอ ดังรูป ก.25 จากนั้นคลิก Finish เพื่อยืนยันการติดตั้ง

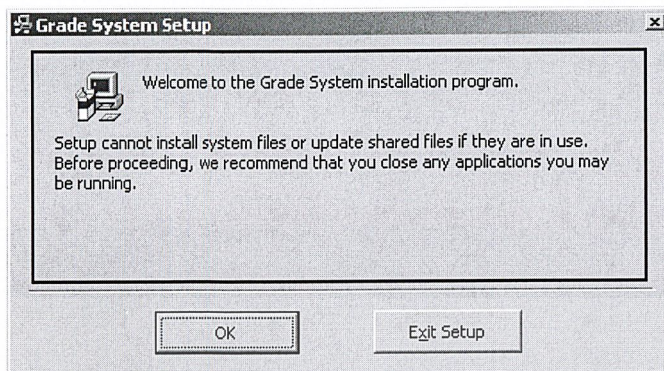


รูปที่ ก.25 สิ้นสุดการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

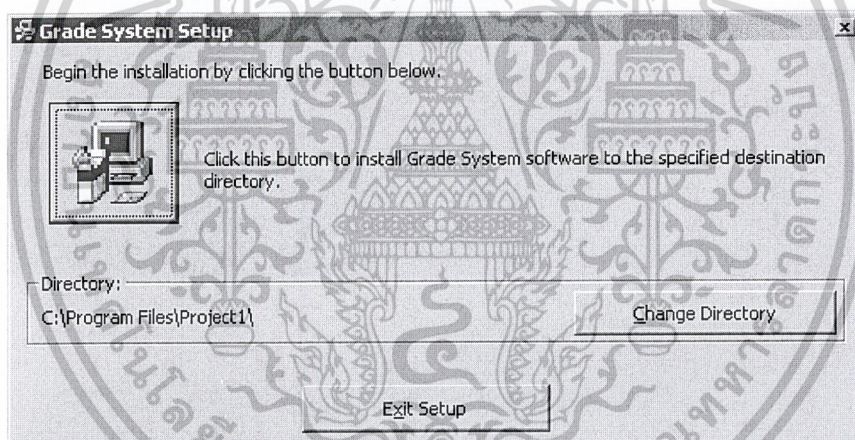
## การติดตั้งโปรแกรมระบบการสอบถามข้อมูลการศึกษาของนักศึกษา

1. ดับเบิลคลิกที่ไฟล์ setup.exe ในโฟลเดอร์ GradeSystem จะปรากฏหน้าจอ ดังรูป ก.26



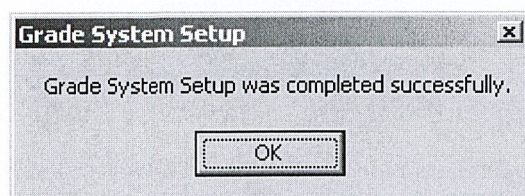
รูปที่ ก.26 หน้าจอการติดตั้ง

2. คลิก OK เพื่อเริ่มต้นการติดตั้ง จะปรากฏหน้าจอ ดังรูป ก.27 เพื่อทำการเลือกตำแหน่งที่จะติดตั้งโปรแกรม



รูปที่ ก.27 หน้าจอการเลือกตำแหน่งในการติดตั้ง

3. ทำการติดตั้ง เมื่อติดตั้งเสร็จเรียบร้อยจะปรากฏหน้าจอ ดังรูป ก.28 เพื่อยืนยันการติดตั้ง



รูปที่ ก.28 สิ้นสุดการติดตั้ง

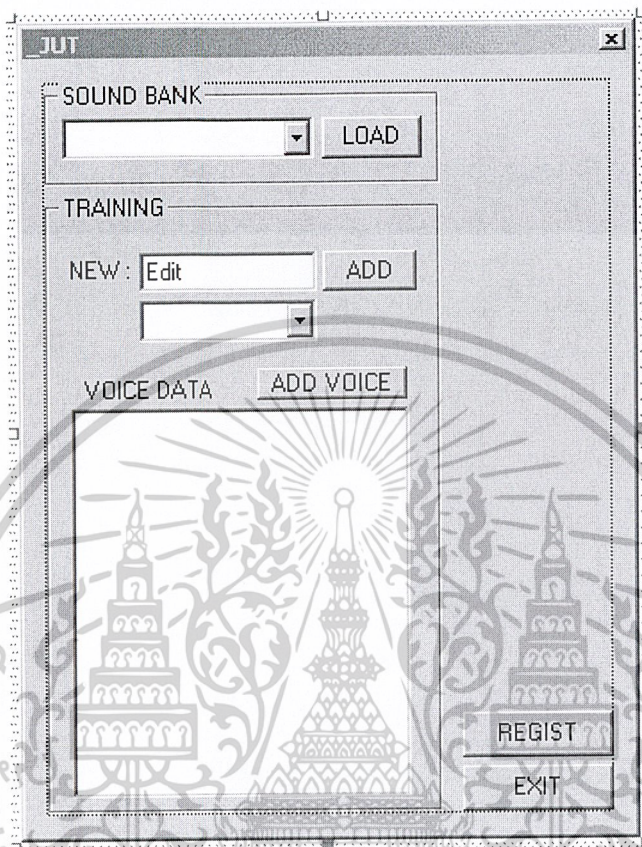
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมการฝึกเสียง

ในส่วนของโปรแกรมการฝึกเสียงนั้น จะใช้ Visual C/C++ ในการพัฒนา โดยสร้างโปรเจกแบบ MFC ซึ่งจะมีการสร้างไดอะล็อก ดังรูป ข.1



รูปที่ ข.1 ไดอะล็อกในโปรแกรมการฝึกเสียง

### Source Code จากไฟล์ SR\_SptrainingDlg.cpp

```
// SR_SptrainingDlg.cpp : implementation file
//

#include "stdafx.h"
#include "SR_Sptraining.h"
#include "SR_SptrainingDlg.h"
#include
<afxdb.h>//////////////////////////////////////

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// SAPI use
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "sp_sr_gram.h"
#define WM_RECOEVENT WM_USER+1

static CString strCombo="";
static CString nbankid,ncate;
////////////////////////////////////
// CSR_SPtrainingDlg dialog

CSR_SPtrainingDlg::CSR_SPtrainingDlg(CWnd* pParent /*=NULL*/)
: CDialog(CSR_SPtrainingDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CSR_SPtrainingDlg)
    m_trnew = _T("");
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in
Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CSR_SPtrainingDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CSR_SPtrainingDlg)
    DDX_Control(pDX, IDC_LIST1, m_lvoice);
    DDX_Control(pDX, IDC_TRCOMBOX, m_trbox);
    DDX_Control(pDX, IDC_SBCOMBOBOX, m_sbbox);
    DDX_Text(pDX, IDC_EDIT1, m_trnew);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CSR_SPtrainingDlg, CDialog)
    //{{AFX_MSG_MAP(CSR_SPtrainingDlg)
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_WM_DESTROY()
    ON_BN_CLICKED(IDC_BUTTON1, OnAddDelete)
    ON_BN_CLICKED(IDC_LOADSB, OnLoadsb)
    ON_BN_CLICKED(IDC_ADDTR, OnAddtr)
    ON_CBN_SELCHANGE(IDC_SBCOMBOBOX, OnSelchangeSbcombobox)
    ON_CBN_SELCHANGE(IDC_TRCOMBOX, OnSelchangeTrcombox)
    ON_BN_CLICKED(IDC_BUTTON2, OnRegist)
    //}}AFX_MSG_MAP
    ON_MESSAGE(WM_RECOEVENT, OnRecoEvent)
END_MESSAGE_MAP()

////////////////////////////////////
// CSR_SPtrainingDlg message handlers

BOOL CSR_SPtrainingDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    if (!InitializeSapi())
    {
        FinishSapi();
        PostMessage(WM_CLOSE);
    }
    // Set the icon for this dialog. The framework does this
automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here

    return TRUE; // return TRUE unless you set the focus to a control
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

// If you add a minimize button to your dialog, you will need the code below  
 // to draw the icon. For MFC applications using the document/view model,  
 // this is automatically done for you by the framework.

```
void CSR_SPtrainingDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user
// drags the minimized window.
HCURSOR CSR_SPtrainingDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

bool CSR_SPtrainingDlg::InitializeSapi()
{
    if (FAILED(CoInitialize(NULL)))
    {
        AfxMessageBox("Error starting COM");
        return false;
    }

    HRESULT hRes = g_cpEngine.CoCreateInstance(CLSID_SpSharedRecognizer);

    if (FAILED(hRes))
    {
        AfxMessageBox("Error starting SAPI");
        return false;
    }

    hRes = g_cpEngine->CreateRecoContext(&g_cpRecoCtxt);

    if (FAILED(hRes))
    {
        AfxMessageBox("Error creating context");
        return false;
    }

    hRes = g_cpRecoCtxt->SetNotifyWindowMessage(m_hWnd, WM_RECOEVENT, 0,
0);

    if (FAILED(hRes))
    {
        AfxMessageBox("Error creating notification window");
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return false;
    }

    hRes = g_cpRecoCtxt->SetInterest(SPFEI(SPEI_RECOGNITION), SPFEI
(SPEI_RECOGNITION));

    if (FAILED(hRes))
    {
        AfxMessageBox("Error creating interest...seriously");
        return false;
    }

    hRes = g_cpRecoCtxt->CreateGrammar(0, &g_cpCmdGrammar);
    if (FAILED(hRes))
    {
        AfxMessageBox("Error creating grammar");
        return false;
    }

    hRes = g_cpCmdGrammar->LoadCmdFromResource(
        NULL,
        MAKEINTRESOURCEW(IDR_SAPI0),
        L"SRGRAMMAR",
        MAKELANGID(LANG_NEUTRAL, SUBLANG_NEUTRAL), SPLO_DYNAMIC);

    if (FAILED(hRes))
    {
        AfxMessageBox("Error creating grammar rules");
        return false;
    }

    // Set rules to active, we are now listening for commands
    hRes = g_cpCmdGrammar->SetRuleState(NULL, NULL, SPRS_ACTIVE );
    if (FAILED(hRes))
    {
        AfxMessageBox("Error setting rule state");
        return false;
    }

    return true;
}

LRESULT CSR_SPtrainingDlg::OnRecoEvent(WPARAM wParam, LPARAM lParam)
{
    // Event helper class
    CSpEvent event;

    // Loop processing events while there are any in the queue
    while (event.GetFrom(g_cpRecoCtxt) == S_OK)
    {
        // Look at recognition event only
        switch (event.eEventId)
        {
            case SPEI_RECOGNITION:
                ExecuteCommand(event.RecoResult());
                break;
        }
    }

    return 0;
}

void CSR_SPtrainingDlg::ExecuteCommand(ISpPhrase *pPhrase)
{
    SPPHRASE *pElements;

    UINT uType = 0;
    COLORREF crShape = RGB(0,0,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Get the phrase elements, one of which is the rule id we specified in
// the grammar. Switch on it to figure out which command was
recognized.

```

```

if (SUCCEEDED(pPhrase->GetPhrase(&pElements)))
{
    switch ( pElements->Rule.ulId )
    {
        case VID_Commands:
        {
            switch(pElements->pProperties->vValue.ulVal )
            {
                case VID_QuitCmd:    { break; }
                case VID_AboutCmd:   { break; }
                case VID_Gen5Cmd:    { break; }
            }
        } break;
        case VID_MainDraw:
        {
            const SPPHRASEPROPERTY *pProp = pElements->
pProperties;

            while (pProp) {
                switch(pProp->vValue.ulVal )
                {
                    case VID_Square: break;
                    case VID_Circle: break;
                    case VID_Triangle: break;
                    case VID_Red: break;
                    case VID_Green: break;
                    case VID_Blue: break;
                    case VID_Black: break;
                    case VID_White: break;
                    case VID_Yellow: break;
                    case VID_Gray: break;
                }
                pProp = pProp->pNextSibling;
            } break;
        }
    }
    // Free the pElements memory which was allocated for us
    ::CoTaskMemFree (pElements);
}
}

```

```

void CSR_SPtrainingDlg::FinishSapi ()
{
    // Release grammar, if loaded
    if (g_cpCmdGrammar) {
        g_cpCmdGrammar.Release();
    }

    // Release recognition context, if created
    if (g_cpRecoCtxt) {
        g_cpRecoCtxt->SetNotifySink (NULL);
        g_cpRecoCtxt.Release();
    }

    // Release recognition engine instance, if created
    if (g_cpEngine) {
        g_cpEngine.Release();
    }
}

```

```

void CSR_SPtrainingDlg::OnDestroy ()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    CDialog::OnDestroy();

    FinishSapi();
}

void CSR_SPtrainingDlg::OnAddDelete()
{
    if ( (nbankid != "") && (ncate != "") )
    {
        HRESULT hr;
        WCHAR *pwszNewWord = NULL;

        ULONG ulDataSize = pwszNewWord ?
            (sizeof(WCHAR) * wcslen( pwszNewWord )) : 0;
        cleardata();

        USES_CONVERSION;
        hr = g_cpEngine->DisplayUI(
            m_hWnd, NULL, SPDUI_AddRemoveWord, pwszNewWord,
ulDataSize );

        if ( pwszNewWord )
        {
            free(pwszNewWord);
        }

        voiceinbase();
        getvoice();
    }
}

void CSR_SPtrainingDlg::OnCancel()
{
    // TODO: Add extra cleanup here
    FinishSapi();

    CDialog::OnCancel();
}

void CSR_SPtrainingDlg::OnLoads()
{
    // TODO: Add your control notification handler code here
    CDatabase m_dbStd;

    TCHAR buffer[256];
    CString vat_values;

    m_dbStd.Open( _T( "_jut" ), FALSE, true, _T( "ODBC;UID=root" ), TRUE );
    CRecordset m_pSet(&m_dbStd);
    // select bankname in combobox
    sprintf( buffer, _T( "SELECT BANKNAME FROM _JUTBANK" ) );
    m_pSet.Open(CRecordset::forwardOnly, _T(buffer) );

    int m=0;
    while( !m_pSet.IsEOF( ) )
    {
        m_pSet.GetFieldValue( m, vat_values );
        m_sbbox.AddString(vat_values);
        m_pSet.MoveNext( );
    }

    m_pSet.Close();
    // select word from bankname
    m_dbStd.Close();
}

void CSR_SPtrainingDlg::OnAddtr()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    char buffer[255];

    UpdateData(true);
    if ( m_trnew != "" )
    {
        int lng = 8 - m_trnew.GetLength();
        if (lng >= 0 )
        {
            sprintf(buffer,"%s",m_trnew);
            for(int il=0;il<lng;il++)
            {
                sprintf(buffer,"%s%c",buffer,'_');
            }
        } else
        {
            sprintf( buffer,"%s", m_trnew.Left(8) );
        }

        m_trbox.AddString(buffer);
    }
}

void CSR_SPtrainingDlg::OnSelchangeSbcombobox()
{
    CDatabase m_dbStd;
    char buffer[255];
    CString vat_values;
    int j=0;

    m_dbStd.Open( _T( "_jut" ), FALSE, true, _T( "ODBC;UID=root" ), TRUE );
    CRecordset m_pSet(&m_dbStd);

    int nCurrentSelected=m_sbbox.GetCurSel();
    m_sbbox.GetLBText(nCurrentSelected,strCombo);

    sprintf( buffer, _T( "SELECT DISTINCT BANKID FROM _JUTBANK WHERE
BANKNAME LIKE '%s'" ),strCombo);
    m_pSet.Open(CRecordset::forwardOnly,_T(buffer) );

    // change StrCombo from BANKNAME to BANKID
    while( !m_pSet.IsEOF() )
    {
        m_pSet.GetFieldValue(j, strCombo);
        m_pSet.MoveNext();
    }
    m_pSet.Close();

    nbankid = strCombo;

    // get CATEGORY from _JUT where StrCombo = BANKID
    sprintf( buffer, _T( "SELECT DISTINCT ncate FROM _JUT WHERE
BANKID=%s" ),strCombo);
    m_pSet.Open(CRecordset::forwardOnly,_T(buffer) );

    while( !m_pSet.IsEOF( ) )
    {
        m_pSet.GetFieldValue( j, vat_values );
        m_trbox.AddString(vat_values);
        m_pSet.MoveNext( );
    }

    m_pSet.Close();
    m_dbStd.Close();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

// get voice from user or application word and display in VOICE DATA ListBox
void CSR_SPtrainingDlg::getvoice(void)
{
    m_lvoice.ResetContent();

    char vword[255],vphone[255];
    char buffer[255];

    HRESULT hr;
    WCHAR getphone[255];
    SPWORDLIST spwordlist;
    memset(&spwordlist, 0, sizeof(spwordlist));
    DWORD dwCookie = 0;
    DWORD dwGeneration;
    hr = g_cpLexicon.CoCreateInstance(CLSID_SpLexicon);
    hr = g_cpLexicon->GetGeneration(&dwGeneration);

    LANGID langidUS = MAKELANGID(LANG_ENGLISH, SUBLANG_ENGLISH_US);
    CComPtr<ISpPhoneConverter> cpPhoneConv;
    hr = SpCreatePhoneConverter(langidUS, NULL, NULL, &cpPhoneConv);

    while (SUCCEEDED(hr = g_cpLexicon->GetWords(eLExTYPe_USER |
eLExTYPe_APP, &dwGeneration, &dwCookie, &spwordlist)))
    {
        for (SPWORD *pword = spwordlist.pFirstWord;
            pword != NULL;
            pword = pword->pNextWord)
        {
            for (SPWORDPRONUNCIATION *pwordpron = pword-
>pFirstWordPronunciation;
                pwordpron != NULL;
                pwordpron = pwordpron->pNextWordPronunciation)
            {
                // DoSomethingWith(pwordpron->ePartOfSpeech,
pwordpron->pszPronIPA);
                hr = cpPhoneConv->IdToPhone(pwordpron->
szPronunciation,getphone);
                sprintf(vphone,"%S",getphone);

                // get word and phone for display in VOICE DATA ListBox
                sprintf(vword,"%S",pword->pszWord);
                sprintf(buffer,"%s : %s",vword,vphone);
                m_lvoice.AddString(_T(buffer)); // add to voice data
                ListBox
            }
        }

        if (hr == S_OK) { break; } // nothing more to retrieve
    }

    //free all the buffers
    CoTaskMemFree(spwordlist.pvBuffer);

    if (g_cpLexicon)
    {
        g_cpLexicon.Release();
    }
}

// clear all voice from user or application word
void CSR_SPtrainingDlg::clsvoice(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    HRESULT hr;
    SPWORDLIST spwordlist;
    memset(&spwordlist, 0, sizeof(spwordlist));
    DWORD dwCookie = 0;
    DWORD dwGeneration;
    hr = g_cpLexicon.CoCreateInstance(CLSID_SpLexicon);
    hr = g_cpLexicon->GetGeneration(&dwGeneration);

    while (SUCCEEDED(hr = g_cpLexicon->GetWords(eLExTYPe_USER |
eLExTYPe_APP, &dwGeneration, &dwCookie, &spwordlist)))
    {
        for (SPWORD *pword = spwordlist.pFirstWord;
            pword != NULL;
            pword = pword->pNextWord)
        {
            hr = g_cpLexicon->RemovePronunciation(pword->
pszWord,pword->LangID,pword->pFirstWordPronunciation->ePartOfSpeech,NULL);
        }

        if (hr == S_OK) { break; } // nothing more to retrieve
    }

    //free all the buffers
    CoTaskMemFree(spwordlist.pvBuffer);

    if (g_cpLexicon)
    {
        g_cpLexicon.Release();
    }

    m_lvoice.ResetContent();
}

void CSR_SPtrainingDlg::OnSelchangeTrcombox()
{
    if ( (nbankid != "") )
    {
        clsvoice();

        int nCurrentSelected=m_trbox.GetCurSel();
        m_trbox.GetLBText(nCurrentSelected,ncate);

        basevoice();
        getvoice();
    }
}

void CSR_SPtrainingDlg::basevoice()
{
    CDatabase m_dbStd;
    char buffer[255];
    CString vat_word;
    CString vat_phone;
    int j=0;

    HRESULT hr;

    LANGID langidUS = MAKELANGID(LANG_ENGLISH, SUBLANG_ENGLISH_US);
    CComPtr<ISpPhoneConverter> cpPhoneConv;
    CComPtr<ISpLexicon> g_addLexicon;
    hr = g_addLexicon.CoCreateInstance(CLSID_SpLexicon);
    hr = SpCreatePhoneConverter(langidUS, NULL, NULL, &cpPhoneConv);
    WCHAR pphone[SP_MAX_PRON_LENGTH];
    WCHAR pword[100];

    SPPHONEID wszId[SP_MAX_PRON_LENGTH];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CRecordset m_pSet(&m_dbStd);

sprintf( buffer, _T( "SELECT word,phone FROM _JUT WHERE BANKID=%s AND
ncate LIKE '%s'",nbankid,ncate);
m_pSet.Open(CRecordset::forwardOnly,_T(buffer) );

while( !m_pSet.IsEOF() )
{
    m_pSet.GetFieldValue(j, vat_word);
    m_pSet.GetFieldValue(1, vat_phone);

    MultiByteToWideChar(CP_ACP,MB_ERR_INVALID_CHARS,vat_word,vat_word.GetL
ength()*sizeof(CString),pword,50);

    MultiByteToWideChar(CP_ACP,MB_ERR_INVALID_CHARS,vat_phone,vat_phone.Ge
tLength()*sizeof(CString),pphone,50);
    hr = cpPhoneConv->PhoneToId(pphone, wszId);
    hr = g_addLexicon->AddPronunciation((const WCHAR *)pword,
langidUS, SPPS_Noun, wszId);

    // for debug
/*
    if ( hr == S_OK ) { MessageBox("Function completed
successfully."); }
    else if ( hr == E_INVALIDARG ) { MessageBox("At least one of
the parameters is not valid."); }
    else if ( hr == SP_ALREADY_IN_LEX ) { MessageBox("The same
pronunciation of the word already exists in the user lexicon."); }
    else if ( hr == SPERR_APPLEX_READ_ONLY ) { MessageBox("Cannot
add a word to application lexicon."); }
    else if ( hr == SPERR_UNINITIALIZED ) { MessageBox("The
interface has not been initialized."); }
    else if ( hr == E_OUTOFMEMORY ) { MessageBox("Exceeded
available memory."); }
*/
    m_pSet.MoveNext();
}
m_pSet.Close();
}

// clear all word from database where bankid and ncate
void CSR_SPtrainingDlg::cleardata(void)
{
    CDatabase m_dbStd;
    char buffer[255];

    m_dbStd.Open( _T( "_jut" ), FALSE, true, _T( "ODBC;UID=root" ),TRUE );

    sprintf( buffer, _T( "DELETE FROM _JUT WHERE BANKID=%s AND ncate LIKE
'%s'",nbankid,ncate);
    m_dbStd.ExecuteSQL( _T(buffer) );
    //m_pSet.Open(CRecordset::forwardOnly,_T(buffer) );

    m_dbStd.Close();
}

// send all the voice to database
void CSR_SPtrainingDlg::voiceinbase(void)
{
    CDatabase m_dbStd;
    m_dbStd.Open( _T( "_jut" ), FALSE, true, _T( "ODBC;UID=root" ),TRUE );

    char vword[255],vphone[255];
    char buffer[255];

    HRESULT hr;
    WCHAR getphone[255];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SPWORDLIST spwordlist;
memset(&spwordlist, 0, sizeof(spwordlist));
DWORD dwCookie = 0;
DWORD dwGeneration;
hr = g_cpLexicon.CoCreateInstance(CLSID_SpLexicon);
hr = g_cpLexicon->GetGeneration(&dwGeneration);

LANGID langidUS = MAKELANGID(LANG_ENGLISH, SUBLANG_ENGLISH_US);
CComPtr<ISpPhoneConverter> cpPhoneConv;
hr = SpCreatePhoneConverter(langidUS, NULL, NULL, &cpPhoneConv);

while (SUCCEEDED(hr = g_cpLexicon->GetWords(eLEXTYPE_USER |
eLEXTYPE_APP, &dwGeneration, &dwCookie, &spwordlist)))
{
    for (SPWORD *pword = spwordlist.pFirstWord;
        pword != NULL;
        pword = pword->pNextWord)
    {
        for (SPWORDPRONUNCIATION *pwordpron = pword-
>pFirstWordPronunciation;
            pwordpron != NULL;
            pwordpron = pwordpron->pNextWordPronunciation)
        {
            hr = cpPhoneConv->IdToPhone(pwordpron->
szPronunciation, getphone);
            sprintf(vphone, "%S", getphone);

            sprintf(vword, "%S", pword->pszWord);
            sprintf(buffer, "INSERT INTO
_JUT(bankid,ncate,word,phone)
VALUES(%s,'%s','%s','%s')", nbankid,ncate,vword,vphone);
            m_dbStd.ExecuteSQL( _T(buffer) );
        }
        if (hr == S_OK) { break; } // nothing more to retrieve
    }

    m_dbStd.Close(); // close the database connection;
    //free all the buffers
    CoTaskMemFree(spwordlist.pvBuffer);

    if (g_cpLexicon)
    {
        g_cpLexicon.Release();
    }
}

```

```

void CSR_SPtrainingDlg::OnRegist()
{
    if ( nbankid != "" )
    {
        clsvoice();
        CDatabase m_dbStd;
        char buffer[255];
        CString vat_word;
        CString vat_phone;
        CString vat_cate;
        CString vat_all;
        int j=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HRESULT hr;

LANGID langidUS = MAKELANGID(LANG_ENGLISH, SUBLANG_ENGLISH_US);
CComPtr<ISpPhoneConverter> cpPhoneConv;
CComPtr<ISpLexicon> g_addLexicon;
hr = g_addLexicon.CoCreateInstance(CLSID_SpLexicon);
hr = SpCreatePhoneConverter(langidUS, NULL, NULL,
&cpPhoneConv);
WCHAR pphone[SP_MAX_PRON_LENGTH];
WCHAR pword[100];

SPPHONEID wszId[SP_MAX_PRON_LENGTH];

m_dbStd.Open( _T( "jut" ), FALSE, true, _T( "ODBC;UID=root"
),TRUE );
CRecordset m_pSet(&m_dbStd);

sprintf( buffer, _T( "SELECT ncate,word,phone FROM _JUT WHERE
BANKID=%s" ),nbankid);
m_pSet.Open(CRecordset::forwardOnly,_T(buffer) );

while( !m_pSet.IsEOF() )
{
    m_pSet.GetFieldValue(j, vat_cate);
    m_pSet.GetFieldValue(1, vat_word);
    m_pSet.GetFieldValue(2, vat_phone);
    sprintf(buffer, "_JUT%s%s",vat_cate,vat_word);
    vat_all = buffer;

    MultiByteToWideChar(CP_ACP,MB_ERR_INVALID_CHARS,vat_all,vat_all.GetLen
gth()*sizeof(CString),pword,50);

    MultiByteToWideChar(CP_ACP,MB_ERR_INVALID_CHARS,vat_phone,vat_phone.Ge
tLength()*sizeof(CString),pphone,50);
    hr = cpPhoneConv->PhoneToId(pphone, wszId);
    hr = g_addLexicon->AddPronunciation((const WCHAR *)
pword, langidUS, SPPS Noun, wszId);

    // for debug
    /* if ( hr == S_OK ) { MessageBox("Function completed
successfully."); }
    else if ( hr == E_INVALIDARG ) { MessageBox("At least
one of the parameters is not valid."); }
    else if ( hr == SP_ALREADY_IN_LEX ) { MessageBox("The
same pronunciation of the word already exists in the user lexicon."); }
    else if ( hr == SPERR_APPLEX_READ_ONLY ) { MessageBox
("Cannot add a word to application lexicon."); }
    else if ( hr == SPERR_UNINITIALIZED ) { MessageBox("The
interface has not been initialized."); }
    else if ( hr == E_OUTOFMEMORY ) { MessageBox("Exceeded
available memory."); }
    */
    m_pSet.MoveNext();
}
m_pSet.Close();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมระบบการสอบถามข้อมูลผลการศึกษานักศึกษา

ในส่วนของระบบการสอบถามข้อมูลผลการศึกษานักศึกษานั้น จะใช้ Visual Basic 6.0 ในการพัฒนา โดยสร้างโปรเจกแบบ Standard EXE ซึ่งจะมีการสร้างฟอร์ม ดังรูป ข.2

รูปที่ ข.2 ฟอร์มการขอระบบการสอบถามข้อมูลผลการศึกษานักศึกษา (form1.frm)

### Source Code จากฟอร์ม form1.frm

```

'-----
'
' This application is Grade System Application
' For Project A Speech Interface For Querying database
' SAPI 5.1. The main object used here for the RecoContext is RC.
'
' Copyright @ 2001 Microsoft Corporation All Rights Reserved.
'
'-----

Option Explicit

Public WithEvents RC As SpSharedRecoContext 'The main shared Recognizer
Context
Public Grammar As ISpeechRecoGrammar 'Command and Control interface
Dim indent As Integer 'Sets indent level for small
output window
Dim fRecoEnabled As Boolean 'Is recognition enabled
Dim fGrammarLoaded As Boolean 'Is a grammar loaded
Dim RecoResult As ISpeechRecoResult 'Recognition result interface

Private Sub Exit_Click()
End
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub prevSem_Click()
    semester.Text = getSemesterFromDB(hidePrev.Text)
    syear.Text = getYearFromDB(hidePrev.Text)
    resetFlexGrid
    OK_Click
End Sub
Private Sub nextSem_Click()
    semester.Text = getSemesterFromDB(hideNext.Text)
    syear.Text = getYearFromDB(hideNext.Text)
    resetFlexGrid
    OK_Click
End Sub
Private Sub OK_Click()
    Dim sid, sem, yyy, sss As String
    Dim allSemester, flag As Boolean
    Dim gps, gpa As Double
    Dim credit, tcredit As Integer
    resetFlexGrid
    StdName.Text = ""
    StdFac.Text = ""
    sid = StudentID.Text
    sem = semester.Text
    yyy = syear.Text
    allSemester = False
    If sid = "" Or Not IsNumeric(sid) Then
        If sid = "" Then
            MsgBox "Please input Student ID"
        Else
            MsgBox "all character in Student ID should be number"
        End If
    Else
        Set rs = execSQL("SELECT * FROM TB_Student WHERE Std_id='" & sid &
        """)
        If rs.RecordCount <= 0 Then
            Exit Sub
        End If
        StdName.Text = rs.Fields("Std_name")
        StdFac.Text = rs.Fields("Std_faculty")
        If sem = "" Or yyy = "" Then
            allSemester = True
        Else
            allSemester = False
            If IsNumeric(sem) Or IsNumeric(yyy) Then
                Else
                    MsgBox "all character in Student ID or Semester or Year
                    should be number"
                End If
            End If

            ' get semester index -----
            query = "SELECT Grd_semester FROM TB_Grade WHERE Std_id='" & sid &
            "" GROUP BY Grd_semester ORDER BY Grd_semester"

            Set rs = execSQL(query)
            If rs.RecordCount > 0 Then
                If allSemester Then
                    sem = getSemesterFromDB(rs.Fields("Grd_semester"))
                    yyy = getYearFromDB(rs.Fields("Grd_semester"))
                    rs.MoveNext
                    hidePrev.Text = ""
                    If Not rs.EOF Then
                        hideNext.Text = rs.Fields("Grd_semester")
                    End If
                    checkEnabledPrevNext
                    semester.Text = sem
                    syear.Text = yyy
                Else
                    flag = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

While (flag)
  If (rs.EOF) Then
    flag = False
  ElseIf rs.Fields("Grd_semester") = getSemesterForDB(sem,
yyy) Then
    flag = False
  Else
    hidePrev.Text = rs.Fields("Grd_semester")
    rs.MoveNext
  End If
Wend
If rs.EOF Then
  MsgBox "No match semester"
ElseIf (rs.BOF) Then
  hidePrev.Text = ""
Else
  rs.MoveNext
  If rs.EOF Then
    hideNext.Text = ""
  Else
    hideNext.Text = rs.Fields("Grd_semester")
  End If
End If
If hidePrev = (yyy & "/" & sem) Then
  hidePrev.Text = ""
End If
checkEnabledPrevNext
End If
Else
  MsgBox "You have never take any course."
End If

' Execute query -----
query = "SELECT g.Crs_id,c.Crs_name,c.Crs_credit,g.Grd_grade" _
& " FROM TB_Course c,TB_Grade g" _
& " WHERE g.Std_id='" & sid & "' AND g.Crs_id=c.Crs_id AND
g.Grd_semester='" & yyy & "/" & sem & "'" _
& " ORDER BY g.Crs_id"
Set rs = execSQL(query)
If rs.RecordCount > 0 Then
  i = 1
  gps = 0#
  tcredit = 0
  While Not rs.EOF
    myFlex.Row = i
    'No.-----
    myFlex.Col = 0
    myFlex.Text = CStr(i)
    'Crs_id-----
    myFlex.Col = 1
    myFlex.Text = rs.Fields("Crs_id")
    'Crs_name-----
    myFlex.Col = 2
    myFlex.Text = rs.Fields("Crs_name")
    'Crs_credit-----
    myFlex.Col = 3
    myFlex.Text = rs.Fields("Crs_credit")
    credit = CInt(rs.Fields("Crs_credit"))
    tcredit = tcredit + credit
    'Grd_grade-----
    myFlex.Col = 4
    myFlex.Text = rs.Fields("Grd_grade")
    gps = gps + multCreditAndGrade(rs.Fields("Grd_grade"),
credit)
    i = i + 1
    rs.MoveNext
  Wend
  gps = gps / CDBl(tcredit)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        End If
        txtGPA.Text = CStr(calGPA(sid))
        txtGPS.Text = CStr(gps)
        txtSem.Text = sem & "/" & yyy
    End If
    StudentID.SetFocus
    chkStudentID.Value = 1
    chkSemester.Value = 0
    chkSyear.Value = 0
End Sub
Private Sub Form_Load()
' Set datagrid
myFlex.Cols = 5
myFlex.Rows = 11
myFlex.Row = 0
myFlex.Col = 0
myFlex.Text = "No."
myFlex.ColWidth(0) = 500
myFlex.ColAlignment(0) = 3
myFlex.Col = 1
myFlex.Text = "Course ID"
myFlex.ColWidth(1) = 1000
myFlex.ColAlignment(1) = 3
myFlex.Col = 2
myFlex.Text = "Course Name"
myFlex.ColWidth(2) = 3847
myFlex.Col = 3
myFlex.Text = "Credit"
myFlex.ColWidth(3) = 800
myFlex.ColAlignment(3) = 3
myFlex.Col = 4
myFlex.Text = "Grade"
myFlex.ColWidth(4) = 800
myFlex.ColAlignment(4) = 3
prevSem.Enabled = False
nextSem.Enabled = False
StudentID.Text = ""
semester.Text = ""
syear.Text = ""
chkStudentID.Value = 1
chkSemester.Value = 0
chkSyear.Value = 0

' Set up error handler
On Error GoTo Err_SAPILoad

' Initialize globals
indent = 0
fRecoEnabled = False
fGrammarLoaded = False

' Create the Shared Reco Context by default
Set RC = New SpSharedRecoContext

' Create grammar objects
LoadGrammarObj

' Attempt to load the default .xml file and set the RuleId State to
Inactive until
' the user starts recognition.
LoadDefaultCnCGrammar

.....
.....

    chkStudentID.Value = 0
    chkSemester.Value = 0
    chkSyear.Value = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Exit Sub
```

```
Err_SAPILoad:
```

```
MsgBox "Error loading SAPI objects! Please make sure SAPI5.1 is  
correctly installed.", vbCritical
```

```
' Exit_Click  
Exit Sub
```

```
End Sub
```

```
' This subroutine creates the Grammar object and sets the states to  
inactive  
' until the user is ready to begin recognition.
```

```
Private Sub LoadGrammarObj()  
Set Grammar = RC.CreateGrammar(1)
```

```
' Load Dictation but set it to Inactive until user selects "Dictation"  
radio  
' button.
```

```
Grammar.DictationLoad "", SLOStatic  
Grammar.DictationSetState SGDSInactive
```

```
End Sub
```

```
' This subroutine attempts to load the default English .xml file. It will  
prompt the
```

```
' user to load a valid .xml file if it cannot find sol.xml in either of  
the 2  
' specified paths.
```

```
Private Sub LoadDefaultCnCGrammar()  
' First load attempt  
On Error Resume Next  
Grammar.CmdLoadFromFile "sol.xml", SLODynamic
```

```
' Second load attempt  
If Err Then  
On Error GoTo Err_CFGLoad  
Grammar.CmdLoadFromFile "sol.xml", SLODynamic  
End If
```

```
' Set rule state to inactive until user clicks Recognition button  
Grammar.CmdSetRuleIdState 0, SGDSInactive
```

```
fGrammarLoaded = True
```

```
Exit Sub
```

```
Err_CFGLoad:
```

```
fGrammarLoaded = False  
MsgBox "Error"  
Exit Sub
```

```
End Sub
```

```
' Activates or Deactivates either Command and Control or Dictation based  
on the
```

```
' current state of the Recognition button.
```

```
Private Sub Active_Mic_Click()  
' ActivateMic.Value = Checked
```

```
On Error GoTo ErrorHandler
```

```
' First make sure a valid .xml file is loaded if the user is selecting C&C  
If fGrammarLoaded Then
```

```
' If recognition is currently not enabled, enable it.  
If Not fRecoEnabled Then
```

```
' Determine if user wants to activate Dictation or Command and
```

```
Control
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Grammar.CmdSetRuleIdState 0, SGDSActive
        fRecoEnabled = True
    Else
        ' Deactivate either Dictation or Command and Control.
        Grammar.CmdSetRuleIdState 0, SGDSInactive
        fRecoEnabled = False
    End If
    StudentID.SetFocus
    chkStudentID.Value = 1
    chkSemester.Value = 0
    chkYear.Value = 0
End If
Exit Sub

ErrorHandler:
    MsgBox "Failed to activate the grammar. It is possible that your audio
device is used by other application.", vbOKOnly
End Sub

' The following subroutines are event handlers that get called when the SR
engine
' fires events.

' Adaption event handler
Private Sub RC_Adaption(ByVal StreamNumber As Long, ByVal StreamPosition As
Variant)
End Sub
' Audio Level event handler
Private Sub RC_AudioLevel(ByVal StreamNumber As Long, ByVal StreamPosition
As Variant, ByVal AudioLevel As Long)
End Sub
' Bookmark event handler
Private Sub RC_Bookmark(ByVal StreamNumber As Long, ByVal StreamPosition As
Variant, ByVal EventData As Variant, ByVal Options As
SpeechLib.SpeechBookmarkOptions)
End Sub
' End Stream event handler
Private Sub RC_EndStream(ByVal StreamNumber As Long, ByVal StreamPosition As
Variant, ByVal StreamReleased As Boolean)
End Sub
' Engine Private event handler
Private Sub RC_EnginePrivate(ByVal StreamNumber As Long, ByVal
StreamPosition As Variant, ByVal lParam As Variant)
End Sub
' False Recognition event handler
Private Sub RC_FalseRecognition(ByVal StreamNumber As Long, ByVal
StreamPosition As Variant, ByVal Result As SpeechLib.ISpeechRecoResult)
End Sub
' Hypothesis event handler
Private Sub RC_Hypothesis(ByVal StreamNumber As Long, ByVal StreamPosition
As Variant, ByVal Result As SpeechLib.ISpeechRecoResult)
End Sub
' Recognition event handler
Private Sub RC_Recognition(ByVal StreamNumber As Long, ByVal StreamPosition
As Variant, ByVal RecognitionType As SpeechLib.SpeechRecognitionType, ByVal
Result As SpeechLib.ISpeechRecoResult)
    Set RecoResult = Result
    recogCommand (RecoResult.PhraseInfo.GetText())
End Sub
' Recognition For Other Context event handler
Private Sub RC_RecognitionForOtherContext(ByVal StreamNumber As Long, ByVal
StreamPosition As Variant)
End Sub
' Recognizer State Change event handler
Private Sub RC_RecognizerStateChange(ByVal StreamNumber As Long, ByVal
StreamPosition As Variant, ByVal NewState As
SpeechLib.SpeechRecognizerState)
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

' Request UI event handler
Private Sub RC_RequestUI(ByVal StreamNumber As Long, ByVal StreamPosition As
Variant, ByVal UIType As String)
End Sub
' Sound End event handler
Private Sub RC_SoundEnd(ByVal StreamNumber As Long, ByVal StreamPosition As
Variant)
End Sub
' Sound Start event handler
Private Sub RC_SoundStart(ByVal StreamNumber As Long, ByVal StreamPosition
As Variant)
End Sub
' Stream Start event handler
Private Sub RC_StartStream(ByVal StreamNumber As Long, ByVal StreamPosition
As Variant)
End Sub
Private Sub deleteText()
    If chkStudentID.Value = 1 Then
        StudentID.Text = ""
    ElseIf chkSemester.Value = 1 Then
        semester.Text = ""
    ElseIf chkSyear.Value = 1 Then
        syear.Text = ""
    End If
End Sub
Private Sub deleteChar()
    If chkStudentID.Value = 1 Then
        StudentID.Text = Left(StudentID.Text, (Len(StudentID.Text) - 1))
    ElseIf chkSemester.Value = 1 Then
        semester.Text = ""
    ElseIf chkSyear.Value = 1 Then
        syear.Text = Left(syear.Text, (Len(syear.Text) - 1))
    End If
End Sub
Private Sub insertText(ByVal recoText As String)
    If chkStudentID.Value = 1 Then
        StudentID.Text = StudentID.Text & recoText
    ElseIf chkSemester.Value = 1 Then
        semester.Text = recoText
    ElseIf chkSyear.Value = 1 Then
        syear.Text = syear.Text & recoText
    End If
End Sub
Private Sub recogCommand(ByVal recogText As String)
    If recogText = "ok" Then
        Call OK Click
    ElseIf recogText = "delete1" Then
        Call deleteChar
    ElseIf recogText = "delete2" Then
        Call deleteText
    ElseIf recogText = "tab" Then
        Call tabCommand
    Else
        insertText (recogText)
    End If
End Sub
Private Sub tabCommand()
    If chkStudentID.Value = 1 Then
        semester.SetFocus
        semester.SelStart = 0
        semester.SelLength = Len(semester.Text)
        chkStudentID.Value = 0
        chkSemester.Value = 1
        chkSyear.Value = 0
    ElseIf chkSemester.Value = 1 Then
        syear.SetFocus
        syear.SelStart = 0
        syear.SelLength = Len(syear.Text)
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        chkStudentID.Value = 0
        chkSemester.Value = 0
        chkSyear.Value = 1
    ElseIf chkSyear.Value = 1 Then
        StudentID.SetFocus
        StudentID.SelStart = 0
        StudentID.SelLength = Len(StudentID.Text)
        chkStudentID.Value = 1
        chkSemester.Value = 0
        chkSyear.Value = 0
    End If
End Sub
Private Sub checkEnabledPrevNext()
    If hidePrev.Text = "" Then
        prevSem.Enabled = False
    Else
        prevSem.Enabled = True
    End If

    If hideNext.Text = "" Then
        nextSem.Enabled = False
    Else
        nextSem.Enabled = True
    End If
End Sub
Function getSemesterFromDB(ByVal sems As String) As String
    getSemesterFromDB = Right(sems, 1)
End Function
Function getYearFromDB(ByVal sems As String) As String
    getYearFromDB = Left(sems, 4)
End Function
Function getSemesterForDB(ByVal s As String, ByVal y As String) As String
    getSemesterForDB = y & "/" & s
End Function
Function multCreditAndGrade(ByVal g As String, ByVal c As Integer) As Double
    Dim cal As Double
    If g = "A" Then
        cal = (4#) * c
    ElseIf g = "B+" Then
        cal = (3.5) * c
    ElseIf g = "B" Then
        cal = (3#) * c
    ElseIf g = "C+" Then
        cal = (2.5) * c
    ElseIf g = "C" Then
        cal = (2#) * c
    ElseIf g = "D+" Then
        cal = (1.5) * c
    ElseIf g = "D" Then
        cal = (1#) * c
    Else
        cal = 0#
    End If
    multCreditAndGrade = cal
End Function
Sub resetFlexGrid()
    Dim i, j As Integer
    For i = 1 To 9
        For j = 0 To 4
            myFlex.Row = i
            myFlex.Col = j
            myFlex.Text = ""
        Next j
    Next i
End Sub
Function calGPA(ByVal id As String) As Double
    Dim gpa As Double
    Dim tc, c As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim r As ADODB.recordSet
Set r = execSQL("SELECT c.Crs_credit,g.Grd_grade FROM TB_Course
c,TB_Grade g WHERE g.Std_id='" & id & "' AND g.Crs_id=g.Crs_id")
gpa = 0#
tc = 0
While Not r.EOF
    c = CInt(r.Fields("Crs_credit"))
    tc = tc + c
    gpa = gpa + multCreditAndGrade(r.Fields("Grd_grade"), c)
    r.MoveNext
Wend
calGPA = gpa / CDBl(tc)
End Function
Private Sub againCommand()
' Set datagrid
myFlex.Cols = 5
myFlex.Rows = 11
myFlex.Row = 0
myFlex.Col = 0
myFlex.Text = "No."
myFlex.ColWidth(0) = 500
myFlex.ColAlignment(0) = 3
myFlex.Col = 1
myFlex.Text = "Course ID"
myFlex.ColWidth(1) = 1000
myFlex.ColAlignment(1) = 3
myFlex.Col = 2
myFlex.Text = "Course Name"
myFlex.ColWidth(2) = 3847
myFlex.Col = 3
myFlex.Text = "Credit"
myFlex.ColWidth(3) = 800
myFlex.ColAlignment(3) = 3
myFlex.Col = 4
myFlex.Text = "Grade"
myFlex.ColWidth(4) = 800
myFlex.ColAlignment(4) = 3
prevSem.Enabled = False
nextSem.Enabled = False
StudentID.Text = ""
semester.Text = ""
year.Text = ""
chkStudentID.Value = 1
chkSemester.Value = 0
chkYear.Value = 0
StudentID.SetFocus
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้