



ระบบสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์
Grade Checking Automation System by Telephony



นางสาวกฤตยา	เหมะฤดี	รหัส 45360021
นางสาววิภาณี	บุญสิงห์	รหัส 45360443
นายเกียรติชัย	เกตุดี	รหัส 45360658

15078850

ห้องสมุดคณะวิศวกรรมศาสตร์	
วันที่รับ.....	28 ส.ค. 2549
เลขทะเบียน.....	4900038
เลขเรียกหนังสือ.....	ป/ร.
มหาวิทยาลัยนเรศวร	
ก2765.	
1548	

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2548



ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ	ระบบสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์
ผู้ดำเนินโครงการ	นางสาวกฤตยา เหมะฤดี รหัสนิสิต 45360021 นางสาววิภาณี บุญสิงห์ รหัสนิสิต 45360443 นายเกียรติชัย เกตุดี รหัสนิสิต 45360658
อาจารย์ที่ปรึกษา	ดร. พนมขวัญ ริยะมงคล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2548

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะกรรมการการสอบโครงการวิศวกรรม

.....ประธานกรรมการ
(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ
(ดร.ธนิต มาลากร)

.....กรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)

หัวข้อโครงการ	ระบบสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์		
ผู้ดำเนินโครงการ	นางสาวกฤตยา	เหมาะฤดี	รหัสนิสิต 45360021
	นางสาววิภาณี	บุญสิงห์	รหัสนิสิต 45360443
	นายเกียรติชัย	เกตุดี	รหัสนิสิต 45360658
อาจารย์ที่ปรึกษา	ดร. พนมขวัญ ริยะมงคล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

บทคัดย่อ

โครงการนี้ได้ถูกพัฒนาขึ้น เพื่อสร้างระบบสอบถามผลการเรียนอัตโนมัติให้ นิสิตสามารถสอบถามข้อมูลผลการเรียนได้ผ่านทางโทรศัพท์ โดยข้อมูลผลการเรียนประกอบด้วย ผลการเรียนรายวิชา ผลการเรียนเฉลี่ยแต่ละภาคเรียน และผลการเรียนเฉลี่ยสะสม ระบบทั้งหมดถูกพัฒนาขึ้นโดยใช้ภาษาจาวาร่วมกับแอปพลิเคชันโปรแกรมมิ่งอินเทอร์เน็ตเฟสที่ช่วยในการเขียน โปรแกรมสำหรับ computer telephony (JTAPI) โปรแกรมที่พัฒนาขึ้นนี้สามารถใช้งานได้บนทุกแพลตฟอร์มที่มี Java Virtual Machine (JVM) และ JTAPI จากการทดลอง ระบบสามารถใช้งานได้ ทั้งระบบโทรศัพท์ที่ผ่านตู้สาขา (PABX) และระบบโทรศัพท์ปกติที่ไม่ผ่านตู้สาขา โดยทั้ง 2 ระบบโทรศัพท์โปรแกรมสามารถรับการสอบถามจากผู้ใช้ได้ทางโทรศัพท์ที่มีมือถือ โทรศัพท์โดยตรงและโทรศัพท์ที่ผ่าน PBX กล่าวโดยสรุป ระบบสามารถใช้งานได้จริงกับทุกรูปแบบการสอบถาม

Project Title	Grade Checking Automation System by Telephone		
Name	Miss Krittaya	Hamaroudee	ID. 45360021
	Miss Wiphanee	Boonsing	ID. 45360443
	Mr. Keattichai	Ketdee	ID. 45360658
Project Advisor	Dr. Panomkhawn Riyamongkol		
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2548		

.....

ABSTRACT

In this project we develop a system for students to inquire the information about their academic evaluation over the phone. The information includes the grades for each course, the average grade for a particular semester, and the average grade for all semesters. The software was developed entirely using Java together with the Java Telephony Application Programming Interface (JTAPI) in order to be able to install the program on any platform with Java Virtual Machine (JVM) and JTAPI. Base on our experiments, we have successfully used this system both with the telephone system with PABX and without PABX. On both occasions, the system was able to response to the requests regardless where the call was originate from. In short, our system is capable of handling all the real-world requests.

กิตติกรรมประกาศ

การจัดทำโครงการวิศวกรรมครั้งนี้สำเร็จลุล่วงด้วยดี เนื่องจากการแนะนำและช่วยเหลือจาก
อาจารย์ที่ปรึกษา ดร. พนมขวัญ ธิยะมงคล และ อาจารย์พงศ์พันธ์ กิจสนาโยธิน

ขอขอบคุณ ดร. สุรเดช จิตประไพกุลศาล และ ดร. ธนิต มาลากร ที่ให้คำปรึกษาในการทำ
โครงการและการจัดทำรูปเล่มโครงการ และขอขอบคุณอาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
ทุกท่าน ที่ได้สั่งสอนและให้ความรู้แก่เรา ซึ่งสามารถนำไปประยุกต์ใช้ในการทำโครงการนี้ได้
อย่างมากมาย และขอขอบคุณบริษัท sun ที่ได้พัฒนาแอปพลิเคชัน โปรแกรมมิ่งอินเทอร์เน็ตเฟสที่มีประ
โยชน์ไว้สำหรับให้นักพัฒนาได้นำมาใช้ในการพัฒนาโปรแกรมได้อย่างสะดวกมากยิ่งขึ้น



นายเกียรติชัย เกตุดี

นางสาวกฤตยา เหมะฤดี

นางสาววิภาณี บุญสิงห์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูปภาพ	ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของ โครงการงาน	1
1.2 วัตถุประสงค์ของโครงการงาน	1
1.3 ขอบข่ายของโครงการงาน	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	2
1.6 งบประมาณที่ใช้	3

บทที่ 2 หลักการและทฤษฎี

2.1 ทฤษฎีเกี่ยวกับโทรศัพท์	4
2.2 โมเด็ม	5
2.3 การเขียนโปรแกรมติดต่อกับโมเด็ม	6
2.4 ระบบฐานข้อมูล MySQL	7
2.5 JDBC (Java Data Base Connection)	11
2.6 การเชื่อมต่อระบบฐานข้อมูล MySQL และ JDBC drivers	15
2.7 Java Telephony API (JTAPI)	16
2.8 XTAPI	28

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนาระบบ

3.1 การทำงานของระบบ	31
3.2 ขอบเขตการใช้งานของระบบ	40
3.3 สิ่งที่ระบบต้องการ	40

บทที่ 4 ผลการทดลอง

4.1 การทดลองระบบสอบถามผลการเรียน	41
4.2 ผลการทดลองการทำการเปลี่ยนแปลงข้อมูลในฐานข้อมูลนิต	47
4.3 ผลการทดลองการใช้โทรศัพท์แบบต่างๆ ในการสอบถามผลการเรียน	49
4.4 ผลการทดลองการเชื่อมต่อโมเด็มกับระบบโทรศัพท์แบบต่างๆ	50

บทที่ 5 สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง	51
5.2 ปัญหาในการทดลอง	52
5.3 ปัญหาในการพัฒนา	52
5.4 ข้อเสนอแนะ	52

เอกสารอ้างอิง	53
---------------------	----

ประวัติผู้ทำโครงการ	54
---------------------------	----

สารบัญตาราง

ตารางที่	หน้า
1.1 แสดงขั้นตอนการดำเนินงาน	2
2.1 แสดงรุ่นของ JDBC ตามการพัฒนาของ JDK	12
2.2 แสดง package ทั้งหมดที่กำหนดไว้ใน JTAPI 1.3	19
2.3 แสดงรายชื่อโมดูลที่สามารถทำงานได้บน XTAPI	30
3.1 แสดงโครงสร้างตารางข้อมูล student ในฐานข้อมูล	35
3.2 แสดงโครงสร้างตารางข้อมูล subject ในฐานข้อมูล	36
3.3 แสดงโครงสร้างตารางข้อมูล grade ในฐานข้อมูล	37
5.1 แสดงผลการทดลองการเชื่อมต่อโมดูลกับระบบโทรศัพท์และการโทรติดต่อระบบ	51



สารบัญรูป

รูปที่	หน้า
2.1	โทรศัพท์แบบหมุน 4
2.2	การ Cross ความถี่ 4
2.3	แสดงการติดต่อระหว่างโปรแกรมกับระบบฐานข้อมูล 11
2.4	แสดง JTAPI Peer 17
2.5	แสดงการรวม API อื่นๆ ไว้ใน JTAPI 17
2.6	แสดงการทำงานของ JTAPI ที่มีสถานะแวดล้อมที่อยู่ต่างที่กัน 18
2.7	แสดง Core Plus Extension Architecture 18
2.8	แสดง JTAPI call model และ object ต่างๆ ใน call model 23
2.9	Connection State Transitions 25
2.10	Terminal Connection State Diagram 27
2.11	แสดงตัวอย่างของไฟล์ xtapi.inf 30
3.1	แสดงส่วนประกอบของระบบสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์ 31
3.2	แสดงขั้นตอนการตรวจสอบรหัสสนิสิตและรหัสผ่าน 32
3.3	แสดงขั้นตอนการสอบถามผลการเรียน 33
3.4	E-R Diagram แสดงโครงสร้างตารางฐานข้อมูลที่ใช้ในระบบ 35
3.5	แสดงโครงสร้างของตาราง student ใน MySql Command Line Client 36
3.6	แสดงโครงสร้างของตาราง subject ใน MySql Command Line Client 37
3.7	แสดงโครงสร้างของตาราง grade ใน MySql Command Line Client 38
4.1	แสดงหน้าต่างอินเตอร์เฟซการเลือกใช้โมเด็ม 41
4.2	แสดงหน้าต่างอินเตอร์เฟซเมื่อมีผู้ใช้ติดต่อเข้ามาในระบบ 42
4.3	แสดงการรับและตรวจสอบรหัสสนิสิตของระบบ 42
4.4	แสดงการรับและตรวจสอบรหัสผ่านของระบบ 43
4.5	แสดงหน้าต่างอินเตอร์เฟซเมื่อรหัสสนิสิตและรหัสผ่านถูกต้อง 43
4.6	แสดงหน้าต่างอินเตอร์เฟซเมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนรายวิชา 44
4.7	แสดงการรับและตรวจสอบรหัสรายวิชาของระบบ 44
4.8	แสดงหน้าต่างอินเตอร์เฟซเมื่อผู้ใช้เลือกรูปแบบการสอบถาม ผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ
4.9	แสดงการรับและตรวจสอบปีการศึกษาและภาคเรียนของระบบ 45

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.10 แสดงหน้าต่างอินเทอร์เน็ตเฟซเมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนเฉลี่ยสะสม	46
4.11 แสดงผลการเรียนเฉลี่ยสะสมของผู้ใช้	46
4.12 แสดงหน้าต่างอินเทอร์เน็ตเฟซเมื่อผู้ใช้ยกเลิกการใช้บริการหรือวางสาย	47
4.13 แสดงหน้าต่างอินเทอร์เน็ตเฟซของฐานข้อมูล Student	48
4.14 แสดงหน้าต่างอินเทอร์เน็ตเฟซของฐานข้อมูล Grade	48
4.15 แสดงหน้าต่างอินเทอร์เน็ตเฟซของฐานข้อมูล Subject	49



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

เนื่องจาก ปัจจุบันเป็นยุคแห่งข่าวสารข้อมูลและบริการ มีการนำเอาอินเทอร์เน็ตมาช่วยในการสืบค้นหรือสอบถามข้อมูลจากองค์กรต่างๆมากมาย ภายในมหาวิทยาลัยก็เช่นกัน มีการจัดทำระบบสอบถามผลการเรียนผ่านระบบเครือข่ายอินเทอร์เน็ต แต่ในความเป็นจริงแล้ว ผู้ที่จะใช้บริการคงจะไม่มีอินเทอร์เน็ตใช้กันทุกคน เนื่องจากไม่สามารถเข้าสู่เครือข่ายได้

ดังนั้น โครงการนี้จึงมีแนวคิดที่จะจัดทำระบบสอบถามผลการเรียนทางโทรศัพท์อัตโนมัติขึ้นมาเพื่ออำนวยความสะดวกในการสอบถามผลการเรียนได้อย่างถูกต้องและมีประสิทธิภาพ สำหรับผู้ที่ไม่สะดวกใช้บริการอินเทอร์เน็ต ซึ่งอุปกรณ์ที่ใช้นั้นได้แก่โทรศัพท์ ซึ่งเป็นอุปกรณ์ที่หาได้ง่าย และมีอยู่ทั่วไป ทำให้ผู้ใช้ลดข้อจำกัดทางด้านอุปกรณ์การสืบค้นได้

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างระบบสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์
2. เพื่อช่วยบริการข้อมูลข่าวสารอย่าง สะดวก รวดเร็ว
3. เพื่อเพิ่มวิธีในการรับข้อมูลข่าวสารของผู้ใช้บริการ

1.3 ขอบข่ายของโครงการ

1. ระบบสามารถรับ input จากผู้ใช้ทางปุ่มกดของโทรศัพท์ (สัญญาณ DTMF)
2. ระบบสามารถส่ง output เป็นเสียงเพื่อตอบกลับ ไปหาผู้ใช้ได้
3. ระบบสามารถเข้าไปค้นหาข้อมูลจากฐานข้อมูลได้
4. ระบบสามารถตอบผลการเรียนได้ดังนี้
 - 4.1 ผลการเรียนรายวิชา
 - 4.2 ผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ
 - 4.3 ผลการเรียนเฉลี่ยสะสม

1.4 ขั้นตอนการดำเนินงาน

การดำเนินงานของโครงการทำระบบสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์ที่มีกำหนดการทำงานดังนี้

ตารางที่ 1.1 แสดงขั้นตอนการดำเนินงาน

ระยะเวลา การดำเนินงาน	พ.ศ. 2547		พ.ศ. 2548										
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	
เขียนโครงสร้างการทำงาน	↔												
ศึกษาการเขียนโปรแกรม ติดต่อกับโมเด็ม		↔		↔									
ศึกษาการส่งเสียงออกไป ทางโมเด็ม				↔	↔								
ศึกษาการตรวจจับสัญญาณ ปุ่มกดจากโทรศัพท์							↔	↔					
ศึกษาการสร้างและการคิด ต่อฐานข้อมูล MySQL							↔	↔					
ออกแบบและสร้างโปรแกรม								↔	↔	↔			
ทดสอบระบบการทำงาน และแก้ไขข้อบกพร่อง									↔	↔	↔		
จัดทำรูปเล่มโครงการ													↔

1.5 ผลที่คาดว่าจะได้รับ

1. มีบริการระบบโทรศัพท์สอบถามผลการเรียนอัตโนมัติมาใช้งานได้จริงในคณะวิศวกรรมศาสตร์
2. ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมควบคุมโมเด็ม
3. ได้รับความรู้เกี่ยวกับการจัดการระบบฐานข้อมูล
4. ได้รับความรู้เกี่ยวกับการส่งไฟล์เสียงผ่านโมเด็ม

1.6 งบประมาณที่ใช้

ค่าวัสดุสำนักงาน	700	บาท
ค่าวัสดุอุปกรณ์คอมพิวเตอร์	1,500	บาท
ค่าถ่ายเอกสารและจัดทำรูปเล่มโครงการ	800	บาท
รวมค่าใช้จ่ายเป็นเงิน	3,000	บาท



บทที่ 2

หลักการและทฤษฎี

2.1 ทฤษฎีเกี่ยวกับโทรศัพท์

2.1.1 โทรศัพท์ที่ใช้งานกันอยู่ทั่วไป

โทรศัพท์ที่ใช้งานกันอยู่ทั่วไปแบ่งเป็น 2 ประเภท

2.1.1.1 ระบบโทรศัพท์แบบหมุน โทรศัพท์แบบหมุนจะ ใช้การขัดจังหวะการไหลของกระแสโดยใช้การเปิดปิดสวิตช์ ตามหมายเลขที่หมุน เช่น ถ้าหมุนหมายเลข 2 สวิตช์จะเปิดออก 2 ครั้งทำให้เกิด Pulse ออกไป 2 ลูก



รูปที่ 2.1 โทรศัพท์แบบหมุน

2.1.1.2 ระบบโทรศัพท์แบบกดปุ่ม (DTMF) โทรศัพท์แบบกดปุ่มจะทำการส่งสัญญาณแบบคลื่นความถี่ โดยจะใช้การส่งสัญญาณความถี่ 2 ค่า Modulate ออกไปพร้อมกันทำให้รู้ว่าผู้ใช้กดปุ่มอะไร เช่นหากมีการกดปุ่ม 1 จะทำการส่งความถี่ 697 Hz ,1209 Hz ออกไป

	1209 Hz	1477 Hz	1336 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

รูปที่ 2.2 การ Cross ความถี่

2.1.2 ส่วนประกอบอื่นๆของวงจรโทรศัพท์

2.1.2.1 วงจรกำเนิดเสียงเรียก (ringer) จะคอยส่งเสียงเมื่อมีการติดต่อเข้ามายังโทรศัพท์ ซึ่งวงจรนี้จะติดต่อกับชุมสายโดยตรง

2.1.2.2 วงจร Switch hook จะเป็นวงจรที่บอกว่าทางฝั่งผู้ใช้พร้อมที่จะใช้ หรืออยู่ในสถานะวางสาย ในขณะที่วางสายจะให้กระแสไหลไม่ครบวงจร ในขณะที่ยกสายจะทำให้กระแสครบวงจรหลังจากที่กระแสไหลครบวงจรที่ชุมสายก็จะส่งสัญญาณ Dial Tone ไปยังผู้ใช้เพื่อเป็นการบอกว่าจะสามารถกดหมายเลขได้

2.1.2.3 Microphone ใช้สำหรับรับ Input ของเสียงเพื่อส่งไปยังผู้รับ

2.2 โมเด็ม

โมเด็ม (Modem) ย่อมาจาก modulate (แปลงสัญญาณ) กับ demodulate (คืนรูปสัญญาณ) การส่งข้อมูลเป็นดังนี้

Computer (DTE) ↔ UART ↔ Modem ↔ NETWORK

การส่งข้อมูลเริ่มจาก คอมพิวเตอร์ส่งข้อมูลซึ่งถูกส่งแบบขนานไปยัง UART (Universal Asynchronous Receive Transmitter) หลังจากนั้น UART จะทำการส่งข้อมูลที่ได้รับมาและส่งออกไปในรูปอนุกรม โดยส่งบิตที่มีนัยสำคัญต่ำสุดก่อนไปให้โมเด็ม หลังจากนั้นข้อมูลแต่ละบิตจะถูกแปลง (Modulate) ให้เป็นสัญญาณเสียงที่มีความถี่อยู่ในย่านที่สามารถส่งผ่านสายโทรศัพท์ได้ก่อนที่จะส่งออกไป เมื่อโมเด็มฝ่ายผู้รับได้รับก็จะทำการ Demodulate ก่อนแล้วส่งให้คอมพิวเตอร์ทางฝ่ายผู้รับอีกครั้ง

ชนิดการส่งข้อมูลของ โมเด็ม มี 3 ชนิดคือ

1. แบบ simplex ส่งข้อมูลที่ทางเดียว
2. แบบ half duplex ส่งข้อมูลได้ 2 ทาง แต่ต้องไม่ใช้ในเวลาเดียวกัน
3. แบบ full duplex ส่งข้อมูลได้ 2 ทาง และอาจเป็นในเวลาเดียวกันก็ได้

ในการ Modulate จะทำการแปลง	ข้อมูล "0" เป็นความถี่ 1070 Hz
	ข้อมูล "1" เป็นความถี่ 1270 Hz

ในช่วงหลังโมเด็ม มีการพัฒนาให้มีความสามารถต่างๆเช่น

Voice Modem

เป็น โมเด็ม ที่มีความสามารถในการ Modulate สัญญาณข้อมูลในลักษณะรูปแบบเสียงไว้ด้วยซึ่งเรียกว่า Voice Modem โดยทำการเพิ่มวงจรการแปลงข้อมูลเสียงทั้งจากสัญญาณอนาลอกเป็นสัญญาณดิจิทัลและจากสัญญาณดิจิทัลเป็นสัญญาณอนาลอก

Detect DTMF

โมเด็มชนิดนี้สามารถจะตรวจสอบสัญญาณการกดปุ่มจากผู้ใช้อีกฝั่งได้

2.3 การเขียนโปรแกรมติดต่อกับโมเด็ม

มี 3 วิธีที่นิยมใช้ดังนี้

- ใช้ AT Command
- ใช้ API เช่น JTAPI , TAPI
- ใช้ Component

2.3.1 AT Command

คำสั่งที่ใช้ควบคุมโมเด็ม ได้รับการพัฒนาโดยบริษัท Hayes ซึ่งเป็นบริษัทผลิตโมเด็มชั้นนำของโลก โดยนำชุดคำสั่งนี้ไปใช้กับโมเด็ม ของตนในเวลาต่อมาชุดคำสั่งนี้ก็เริ่มแพร่หลายใช้กับโมเด็ม จากทุกบริษัทจนมีการรับรองเป็นมาตรฐานของคำสั่งควบคุมการทำงานของ โมเด็มชุดคำสั่งเหล่านี้มีชื่อเรียกว่า Hayes Command หรือ AT Command มีลักษณะดังนี้ ATDT 123456 เป็นคำสั่งที่สั่งให้โมเด็มโทรออกไปที่ เบอร์ 123456 , ATH จะสั่งให้วางหูโทรศัพท์ โดยคำสั่งเหล่านี้ไม่ได้สั่งให้โมเด็ม โดยตรง แต่จะสั่งให้ Driver ที่ควบคุม โมเด็ม ก่อน แล้ว Driver จึงจะนำคำสั่งนั้นไปส่งงาน โมเด็มอีกทีหนึ่ง

ข้อดี - ง่ายในการสร้างโปรแกรมขนาดเล็ก

- สามารถใช้คำสั่งที่เป็นคำสั่งเฉพาะของโมเด็ม แต่ละรุ่นได้

ข้อเสีย - ต้องทำการเขียน โปรแกรมกับ port โดยตรง

2.3.2 API

เป็นเครื่องมือที่ถูกสร้างขึ้นมาเพื่อใช้ในการติดต่อกับโมเด็ม ได้ง่ายขึ้นแทนแบบ AT Command ที่พบเห็นได้บ่อยคือ JTAPI จาก Sun เพื่อใช้งานกับภาษา JAVA และ TAPI จาก Microsoft โดยผู้ใช้ทำการเขียนโปรแกรมกับ API นั้นไม่ต้องอ้างถึงในระดับ physical เช่นต้องอ้างอิงกับ port ที่ใช้ต่อกับโมเด็มนั้น ผู้เขียนโปรแกรมจะอ้างอิงอุปกรณ์ทุกอย่างผ่าน API ไม่ว่าจะเป็นการค้นหา โมเด็ม หรือ Sound device ของ โมเด็ม ตัวนั้นๆ

2.3.2.1 JTAPI (Java Telephony API) JTAPI เป็นอินเตอร์เฟซในการเขียนโปรแกรมสำหรับ computer telephony

ซึ่งถูกพัฒนาโดยบริษัท SUN (Sun Micro-System Co., LTD.) JTAPI ถูกออกแบบมาเพื่อให้สามารถเขียน โปรแกรมได้ง่ายแต่มีคุณลักษณะต่างๆครบ ซึ่งสามารถใช้ในการเขียนโปรแกรมได้หลากหลายตั้งแต่โปรแกรม call center ไปจนถึงพัฒนาเว็บเพจ และยังสนับสนุนการควบคุมการเรียกจากแบบบุคคลที่หนึ่งและสาม (first และ third party call control) แอปพลิเคชันที่ถูกพัฒนาจาก JTAPI สามารถนำไปใช้ในแพลตฟอร์มและระบบโทรศัพท์ต่างๆกันได้ เนื่องจาก JTAPI ได้ใช้ภาษา Java ซึ่งโปรแกรมที่พัฒนาโดยภาษานี้จะสามารถรันได้โดยไม่ขึ้นกับแพลตฟอร์มใดๆ

ความต้องการของผู้พัฒนา JTAPI คือต้องการให้การพัฒนาโปรแกรมทำได้ง่ายและโปรแกรมสามารถทำงานได้บนแพลตฟอร์มต่างๆ โดยการเขียนโปรแกรมเพียงครั้งเดียว JTAPI ช่วยให้ผู้พัฒนาไม่จำเป็นต้องรู้หรือเข้าใจกลไกบางอย่างของ IP telephony สำหรับการสร้างแอปพลิเคชันได้ในระดับกว้าง ตั้งแต่เดสทอปแอปพลิเคชันไปจนถึง call center รวมทั้ง JTAPI ยังได้ลดความแตกต่างระหว่างการควบคุมการเรียกจากบุคคลที่หนึ่งและสาม และความแตกต่างระหว่างการควบคุมการเรียก (call center) และการควบคุมสื่อมีเดีย (media control) นอกจากนี้ผู้พัฒนาโปรแกรมสามารถที่พัฒนาโดยใช้ JTAPI อยู่บน telephony API อื่นๆ ได้ เช่น TAPI และ TSAPI

2.3.3 Component

Component ส่วนใหญ่ที่มี จะเป็น Component ที่ใช้งานกับภาษา Visual Basic และทำงานบนระบบ Windows ซึ่งส่วนใหญ่จะมีการกำหนดราคาขายและมีให้เลือกใช้หลายรุ่น ซึ่งก็รวมทั้งรุ่นที่ไม่เสียค่าใช้จ่ายในการทดลองใช้ด้วยแต่ก็มักจะมีการกำหนดระยะเวลาในการใช้งาน Component เช่น 15 วัน 1 เดือน หรือ 3 เดือน เป็นต้น โดย Component นั้นก็จะทำงานบน TAPI ต่างๆ อีกที่

ข้อดี - มีรุ่นฟรีให้ทดลองใช้ แม้จะมีการจำกัดระยะเวลาการใช้งาน

- ใช้งานง่ายและมีเอกสารประกอบการใช้งานพร้อมตัวอย่างการใช้งานอย่างครบถ้วน

ข้อเสีย - จำกัดเวลาทำให้ไม่สามารถนำมาใช้ในการทำงานจริงได้ ถ้าจะใช้ในระยะเวลาานาน

จะต้องเสียค่าใช้จ่ายในการซื้อ Component

- ความสามารถในการทำงานจำกัด

2.4 ระบบฐานข้อมูล MySQL

MySQL เป็นโปรแกรมฐานข้อมูลที่ใช้จัดเก็บข้อมูล โปรแกรมหนึ่ง ทำงานในลักษณะ Client-Server โดย MySQL เป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่ง Database Management System (DBMS) ที่ทำงานอยู่บนพื้นฐานของโมเดลข้อมูลเชิงสัมพันธ์ เรียกว่า “ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS)” คือ สามารถทำงานกับตารางข้อมูลหลายตารางพร้อมๆ กัน โดยสามารถแสดงความสัมพันธ์ของตารางเหล่านั้นด้วย field ที่ใช้ร่วมกัน โดยใช้ ภาษา SQL (Query Language) ในการทำงานกับ ระบบฐานข้อมูล MySQL มีคำสั่งพื้นฐานดังนี้

2.4.1 การสร้างเทเบิลและจัดการข้อมูล

ฐานข้อมูลเชิงสัมพันธ์จะประกอบด้วยเทเบิล (Table) ซึ่งเป็นตารางสำหรับบันทึกข้อมูลเก็บไว้ภายใต้โครงสร้างของเทเบิลที่ได้รับการออกแบบตามความต้องการใช้งานของระบบงานต่างๆ

2.4.1.1 การสร้างเทเบิล (Create Table)

การสร้างเทเบิลด้วย SQL สามารถทำได้ด้วยคำสั่ง CREAT TABLE ซึ่งจะกำหนดลักษณะของข้อมูลเป็นคอลัมน์ (Column) และมีรูปแบบดังนี้

รูปแบบ

```

CREATE TABLE <table_name>
(<column_name_1> <data_type> [NOT NULL][UNIQUE],
 <column_name_2> <data_type> [NOT NULL][UNIQUE],
...
...
...
<column_name_n> <data_type> [NOT NULL][UNIQUE],
[PRIMARY KEY (column_name),]
[FOREIGN KEY (column_name) REFERENCES table_name]);

```

คำอธิบาย

- คำว่า CREATE TABLE เป็นคำสำคัญหรือคีย์เวิร์ดที่ต้องมีทุกครั้ง และตามด้วยชื่อของเทเบิลที่ต้องการสร้าง

- แต่ละบรรทัดที่เหลือในวงเล็บ เป็นการกำหนดชื่อของคอลัมน์ตามด้วยประเภทของข้อมูล
- สำหรับแต่ละเทเบิลต้องมีอย่างน้อย 1 คอลัมน์
- รายละเอียดของแต่ละคอลัมน์ให้คั่นด้วยเครื่องหมาย , (จุลภาค หรือ comma)
- ทุกคำสั่งจะปิดท้ายด้วยเครื่องหมาย ; (semicolon)
- PRIMARY KEY หมายถึง การกำหนดคอลัมน์ที่เป็นคีย์หลักของเทเบิลที่สร้างขึ้น
- FOREIGN KEY หมายถึง การกำหนดคอลัมน์ที่เป็นคีย์นอกของเทเบิลที่สร้างขึ้นมา ซึ่งจะมี

ความสัมพันธ์กับคีย์หลักของเทเบิลที่อ้างถึง หลังคำว่า REFERENCES

- คำว่า NOT NULL ใ้ต่อท้ายคอลัมน์ใดคอลัมน์นั้นๆจะต้องมีค่าเสมอ
- คำว่า UNIQUE ใ้ต่อท้ายคอลัมน์ใด ข้อมูลในแต่ละแถวของคอลัมน์นั้นๆจะต้องไม่ซ้ำกัน

2.4.1.2 การเปลี่ยนโครงสร้างของเทเบิล (Alter Table)

บางครั้งอาจต้องมีการเปลี่ยน โครงสร้างข้อมูลในเทเบิล เนื่องจากการออกแบบที่ผิดพลาดในตอนต้นหรือจากการเปลี่ยนความต้องการของผู้ใช้งาน ในกรณีที่เทเบิลมีข้อมูลอยู่แล้วควรระวังเป็นพิเศษ เพราะการแก้ไขโครงสร้างอาจมีผลกระทบต่อข้อมูลเหล่านั้น ในการแก้ไขโครงสร้างของเทเบิลทำได้ด้วยคำสั่ง ALTER TABLE โดยมีรูปแบบดังนี้

รูปแบบ

```

ALTER TABLE <table_name>
[MODIFY] [<column_name> <data_type>],
[ADD] [<column_name> <data_type>];

```

2.4.1.3 การลบเทเบิลออกจากฐานข้อมูล (Drop Table)

ในการลบเทเบิลที่มีอยู่ออกจากฐานข้อมูล ทำได้โดยใช้คำสั่ง DROP TABLE ดังนี้

รูปแบบ DROP TABLE <table_name>;

2.4.1.4 การป้อนข้อมูลแต่ละแถวลงในเทเบิล (Insert)

หลังจากสร้างเทเบิลตามโครงสร้างที่ออกแบบไว้ ในแต่ละเทเบิลจะยังไม่มีข้อมูลอยู่เลย ในการป้อนข้อมูลแต่ละแถวลงในเทเบิล จะใช้คำสั่ง INSERT INTO โดยมีรูปแบบดังนี้

รูปแบบ INSERT INTO <table_name>
VALUES (<value_1>, <value_2>, ...);

คำอธิบาย

- คำว่า INSERT INTO เป็นคีย์เวิร์ดที่ต้องมีทุกครั้ง และตามด้วยชื่อของเทเบิลที่ต้องการใส่ข้อมูลลงไป
- คำว่า VALUES เป็นคีย์เวิร์ดที่ต้องมีทุกครั้ง และตามด้วยข้อมูลของแต่ละแถว ซึ่งประกอบด้วยข้อมูลในทุกคอลัมน์เรียงตามลำดับจากซ้ายไปขวา เหมือนอย่างตอนที่สร้างเทเบิลด้วยคำสั่ง CREATE TABLE
- คั่นข้อมูลแต่ละคอลัมน์ด้วยเครื่องหมาย , (comma)
- รายละเอียดของแต่ละคอลัมน์ให้คั่นด้วยเครื่องหมาย , (จุลภาค หรือ comma)

ในบางครั้งข้อมูลบางแถวอาจมีไม่ครบทุกคอลัมน์ การป้อนข้อมูลลักษณะนี้สามารถทำได้โดยมีรูปแบบดังนี้

รูปแบบ INSERT INTO <table_name><(list of column_name)>
VALUES <(list of matching values)>;

คำอธิบาย

- ในการป้อนข้อมูลแต่ละแถว บางครั้งอาจต้องการป้อนข้อมูลเพียงบางคอลัมน์เท่านั้น
- คำว่า INSERT INTO เป็นคีย์เวิร์ดที่ต้องมีทุกครั้ง และตามด้วยชื่อของเทเบิลที่ต้องการป้อนข้อมูลเข้าไป จากนั้นตามด้วยชื่อของคอลัมน์เฉพาะที่ต้องการป้อนข้อมูลเท่านั้น ซึ่งจะเขียนอยู่ในวงเล็บ
- คำว่า VALUES เป็นคีย์เวิร์ดที่ต้องมีทุกครั้งและตามด้วยข้อมูลที่เรียงให้ตรงกับชื่อคอลัมน์ที่กำหนดข้างต้นตามลำดับจากซ้ายไปขวา
- ในคอลัมน์ที่ระบุ NOT NULL จะต้องใส่ข้อมูลด้วยเสมอ ไม่สามารถจะละได้

2.4.1.5 การใส่ข้อมูลโดยใช้ข้อมูลจากเทเบิลอื่นใหม่

ความสามารถที่เพิ่มขึ้นของคำสั่ง INSERT INTO โดยการแทรก (embed) คำสั่ง SELECT เพิ่มเข้าไปในคำสั่ง INSERT INTO ในอีกรูปแบบหนึ่งได้ดังนี้

รูปแบบ INSERT INTO <table_name>
 SELECT <(list of column_name)>
 FROM <table_name>
 WHERE <(column_name)> = value ;

2.4.1.6 การแก้ไขข้อมูลที่มีอยู่ (Update)

หลังจากป้อนข้อมูลเข้าไปในเทเบิลแล้ว ในกรณีที่เราต้องการแก้ไขข้อมูลที่มีอยู่แล้ว ทำได้โดยใช้คำสั่ง UPDATE ดังนี้

รูปแบบ UPDATE <table_name>
 SET <column_name> = new_value
 [WHERE <condition>] ;

2.4.1.7 การลบข้อมูลที่มีอยู่ (Delete)

การลบข้อมูลออก 1 แถว หรือหลายแถวจากเทเบิลหนึ่ง สามารถใช้คำสั่ง DELETE โดยมีรูปแบบดังนี้

รูปแบบ DELETE FROM <table_name>
 WHERE <condition>;

2.4.2 การสอบถามข้อมูลด้วย SELECT

คำสั่ง SELECT ประกอบด้วยส่วนพื้นฐานสำคัญ 2 ส่วนคือ SELECT และ FROM โดยมีรูปแบบดังนี้

รูปแบบ SELECT column_name, list of column_names, or *
 FROM table_name, or list of table_names ;

คำอธิบาย

- SELECT เป็นคำสั่งที่ใช้เลือกเอาเฉพาะข้อมูลตามคอลัมน์ที่กำหนดท้ายคำสั่ง SELECT มาแสดงผล

- กรณีที่ต้องการแสดงทุกคอลัมน์ ให้ใช้ * แทนทุกคอลัมน์

- FROM ใช้ระบุชื่อเทเบิลที่ใช้เลือกข้อมูล

ดังนั้นในการสอบถามข้อมูลด้วยคำสั่ง SELECT จะต้องกำหนดชื่อคอลัมน์และชื่อเทเบิลให้

ชัดเจน

2.4.3 การสอบถามข้อมูลแบบมีเงื่อนไข (SELECT... WHERE)

การกำหนดเงื่อนไขเพื่อให้เห็นเฉพาะแถวที่ตรงตามเงื่อนไขที่ต้องการเท่านั้น โดยในคำสั่ง SELECT ให้เพิ่มประโยคคำสั่งย่อย WHERE เพื่อกำหนดเงื่อนไขในการค้นหาข้อมูลที่ต้องการในเทเบิล โดยมีรูปแบบดังนี้

```
รูปแบบ      SELECT    <column_name>
              FROM      <table_name>
              WHERE     condition;
```

2.5 JDBC (Java Data Base Connection)

JDBC ถูกพัฒนาโดยบริษัท Sun Microsystem Inc. แบ่งออกเป็น 2 ส่วนคือ JDBC api เป็นชุดของ Class และ Interfaces เพื่อให้โปรแกรมภาษาจาวาใช้ติดต่อกับระบบฐานข้อมูล อีกส่วนหนึ่งคือข้อกำหนดของ JDBC drivers เพื่อให้บริษัทผู้ผลิตระบบฐานข้อมูลนำไปสร้าง drivers ที่ติดต่อกับฐานข้อมูลของเขา ทำให้โปรแกรมไม่ผูกติดกับระบบฐานข้อมูลที่ใช้ รวมทั้งไม่จำเป็นต้องทราบรายละเอียดวิธีการติดต่อกับระบบฐานข้อมูลนั้น การติดต่อกันระหว่างโปรแกรมกับระบบฐานข้อมูล แสดงได้ดังในรูปร่างด้านล่าง



รูปที่ 2.3 แสดงการติดต่อกันระหว่างโปรแกรมกับระบบฐานข้อมูล

ผู้ผลิตฐานข้อมูลแต่ละประเภทจะสร้าง drivers สำหรับระบบฐานข้อมูลนั้น โดย Drivers จะแปลคำสั่งจาก JDBC เป็นคำสั่งสำหรับเข้าใช้งานระบบฐานข้อมูล ส่วนโปรแกรมภาษาจาวา จะส่งคำสั่งสำหรับติดต่อกับระบบฐานข้อมูลเป็น SQL ไปที่ JDBC แล้ว JDBC เปลี่ยนคำสั่งนั้นเป็นคำสั่งสำหรับ drivers

การติดตั้ง drivers สำหรับระบบฐานข้อมูลนั้นอาจทำได้โดยโปรแกรมติดตั้งเอง หรือจะระบุ drivers เมื่อโปรแกรมทำงานก็ได้ ดังนั้นโปรแกรมที่จะใช้ระบบฐานข้อมูลจึงเป็นอิสระจากระบบฐานข้อมูล คือเราสามารถสร้างโปรแกรมได้โดยไม่ต้องทราบเลยว่า ระบบฐานข้อมูลที่จะใช้จริงๆเป็นยี่ห้อใด และหากเปลี่ยนระบบฐานข้อมูลก็ไม่ต้องเขียนโปรแกรมใหม่ทั้งหมด เพราะโปรแกรมจะติดต่อออกไปเป็น SQL ผ่านทาง JDBC ที่เป็นมาตรฐาน อีกทั้งโปรแกรมหนึ่งอาจมี drivers ติดตั้งอยู่ได้หลายตัว ทำให้โปรแกรมหนึ่งสามารถติดต่อกับระบบฐานข้อมูลได้หลายตัวพร้อมกัน

JDBC รุ่น 1.0 ถูกเพิ่มเข้ามาในภาษาจาวา ตอนที่ เป็น JDK 1.1 ต่อมา มีการพัฒนาเปลี่ยนแปลง อีกสองครั้งใน JDK รุ่นถัดมา แสดงได้ดังนี้

ตารางที่ 2.1 แสดงรุ่นของ JDBC ตามการพัฒนาของ JDK

JDBC	JDK
1.0	1.1
2.0	1.2
3.0	1.4

ไม่ว่าจะเป็น JDBC รุ่นใดก็ตาม จะต้องมียกเลิกพื้นฐาน 3 อย่างคือ

1. เริ่มต้นติดต่อกับระบบฐานข้อมูล
2. ส่งคำสั่ง SQL ไประบบฐานข้อมูล
3. การรับผลลัพธ์จากการใช้คำสั่ง SQL แล้วทำการประมวลผล

JDBC ทั้งสามรุ่นสนับสนุนยกเลิกสำหรับใช้งานขั้นพื้นฐานเหมือนกัน แต่ JDBC รุ่นหลังจะเพิ่ม กลไกบางอย่างที่รุ่นก่อนหน้าไม่มี นับเริ่มตั้งแต่ JDBC 2.0 เป็นต้นมา คลาสและ interface ของ JDBC ถูกแบ่งออกเป็นสอง packages หนึ่งคือ "java.sql" มีกลไกขั้นพื้นฐาน ซึ่งเป็นข้อกำหนดว่า drivers ของระบบฐานข้อมูลยี่ห้อใดที่จะใช้งานกับโปรแกรมภาษาจาวาต้องสนับสนุน "java.sql" ทั้งหมด สองคือ "javax.sql" ซึ่งเรียกว่า JDBC optional package ที่มีกลไกที่ drivers ของระบบฐานข้อมูลที่ระบบ j2ee ต้องสนับสนุน แต่ระบบฐานข้อมูลทั่วไปอาจจะไม่สนับสนุนก็ได้

2.5.1 ลักษณะที่สำคัญของ JDBC

JDBC เป็น low-level Interface ซึ่งสามารถเรียกใช้คำสั่ง SQL ได้โดยตรงซึ่งสามารถใช้งานได้ อย่างมีประสิทธิภาพ มีขนาดเล็ก และง่ายต่อการใช้กว่าตัวติดต่อ Data Base ชนิดอื่น ๆ แต่ก็ยังมีออกแบบสำหรับใช้งานแบบ High-Level Interface ด้วย ซึ่ง High Level Interface คือ User Friendly ซึ่ง High-level API ของ JDBC ที่ถูกพัฒนาซึ่งมีดังนี้

1. JDBC ต้องการคำสั่ง SQL ที่จะทำการส่งเป็น String ไปยัง Method ของภาษา Java ทำให้ Programmer สามารถใช้ตัวแปรในการรับส่งคำสั่ง SQL ได้
2. มีการแปลง relation DataBase Table ไปเป็น Java Class ซึ่งเรียกว่า Object/Relational ซึ่งแต่ละแถวของ Table จะถูกแปลงไปเป็น Instant ของ Class และแต่ละ Column จะเป็น attribute ของ Class ซึ่ง Programmer สามารถดำเนินการต่าง ๆ คล้ายด้วยว่า Table เป็น Class หนึ่ง ใน Java

2.5.2 การเปรียบเทียบระหว่าง JDBC และ ODBC

Microsoft ODBC API สามารถใช้ Application ในการติดต่อกับ DataBase ได้เกือบทุกประเภท เช่นกัน มีคำถามว่าทำไมไม่ใช้ ODBC ในการติดต่อฐานข้อมูลในภาษา Java ซึ่งคำตอบก็คือ สามารถใช้ ODBC ในการติดต่อฐานข้อมูลกับ Java ได้ แต่จะเป็นการดีถ้ามีการใช้ JDBC-ODBC Bridge ซึ่งเป็นรูปแบบหนึ่งของ JDBC ซึ่งสามารถสรุปคำตอบคำถามนี้ได้ว่า

1. ODBC ไม่มีความสะดวกในการใช้งานได้โดยตรงจาก Java เพราะมีการใช้ C Interface ในการเรียกคำสั่งจาก Java แล้วแปลงเป็น code ภาษา C มีผลกระทบหลายเรื่อง เช่น Security , implement , robustness ของ Application
2. ตัวแปรที่มีการแปลงจาก ODBC (API ไปเป็น Java API จะทำให้เกิดปัญหาขึ้น เช่น Java ไม่มี Pointer ในขณะที่ C มีการใช้งาน Pointer
3. ODBC ยากต่อการเรียนรู้ เพราะมีการผสมผสานคุณสมบัติง่ายและยากเข้าไว้ด้วยกันและมีการนำเอา option ต่าง ๆ มารวบรวมกันไว้โดยที่เราต้องศึกษา object ทั้งหมด ของ ODBC เพื่อที่ใช้คุณสมบัติแยกย่อยต่าง ๆ ไป แต่ JDBC มีการออกแบบมาเพื่อการใช้งานให้ได้ง่าย จะมี option ต่างหากเพื่อใช้ในงานที่ยากต่างหาก ซึ่งอยู่ในส่วนของ Advanced Programmer จะต้องศึกษาต่อไป
4. Java และ JDBC จะมีการใช้งานร่วมกันได้เพราะต่างก็เขียนขึ้นมาด้วย Code ภาษา Java ซึ่งต่างจาก ODBC การจัดการ ODBC Driver ต้องมีการ ติดตั้งลงในทุก ๆ เครื่องที่มีการใช้งาน ในขณะที่ JDBC Driver มีการเขียนขึ้น ใน code ภาษา Java JDBC code จะถูกรวมอยู่ใน Application โดยอัตโนมัติ ซึ่งจะมีความปลอดภัยกับข้อมูลในการใช้งานในระบบ Network

2.5.3 ประเภทของ JDBC Driver

เมื่อโปรแกรมภาษาจาวาต้องการติดต่อกับระบบฐานข้อมูลใด จะต้องใช้ drivers สำหรับระบบฐานข้อมูลนั้น เพื่อทำหน้าที่เชื่อมต่อระหว่างโปรแกรมกับฐานข้อมูล ซึ่งแต่ละระบบฐานข้อมูลจะมีความแตกต่างกันไปในแต่ละประเภท ดังนั้น drivers ที่สร้างขึ้นสำหรับต่อกับ JDBC จึงถูกสร้างขึ้นด้วยวิธีที่ต่างกันไป ซึ่งส่งผลถึงสภาพแวดล้อมในการทำงานด้วย เราสามารถแบ่ง drivers ตามวิธีที่ถูกสร้างขึ้นมาได้ 4 ประเภท ดังนี้

2.5.3.1 JDBC-ODBC Bridge Driver

เป็น drivers ที่ถูกสร้างขึ้นมาโดยใช้ native code เพื่อทำหน้าที่เป็นตัวกลางระหว่างโปรแกรมกับ ODBC drivers ข้อดีของการใช้ bridge drivers คือ ODBC drivers สามารถต่อกับระบบฐานข้อมูลได้หลายประเภท โดยเฉพาะที่ทำงานได้ใน Microsoft Windows แต่ข้อเสียคือ มีการแปลคำสั่งเพิ่มขึ้นอีกระดับหนึ่ง ทำให้ทำงานช้าเมื่อเปรียบเทียบกับ drivers ประเภทอื่น และเครื่องด้าน clients ต้องมีการติดตั้งโปรแกรมของ ODBC รวมทั้งต้องทำ configuration ให้มองเห็น databases ที่จะใช้ ซึ่งจุดประสงค์ที่แท้จริงในการสร้าง JDBC-ODBC driver คือเพื่อให้ผู้เขียน โปรแกรมทดลองเปลี่ยน ODBC เป็น JDBC แต่ในปัจจุบัน ระบบฐานข้อมูลเกือบทุกประเภทมี drivers ประเภทอื่นที่จะกล่าวต่อไป ซึ่ง

ทำงานได้เร็วกว่า ละเชื่อถือได้มากกว่า ดังนั้น ควรใช้ JDBC-ODBC driver ในการทดลองหรือเรียนรู้เท่านั้น ส่วนในการใช้งานจริงควรใช้ drivers ประเภทอื่น

2.5.3.2 Native API partly-Java Driver

เป็น drivers ที่ถูกสร้างขึ้นมาโดยใช้ native code แล้วนำมาสร้างเป็นคลาสของภาษาจาวา ในการใช้งานจะติดตั้งทั้ง native code และคลาสของจาวาที่เครื่องฝั่ง clients ทุกตัวที่ใช้งานเช่นเดียวกับแบบแรก แต่เนื่องจากไม่ต้องผ่าน driver ตัวกลางอย่าง ODBC และมีบางส่วนที่ทำงานด้วย native code จึงทำงานได้เร็วกว่าแบบอื่นๆ ยกเว้น Native-protocol pure Java Driver ตัวอย่าง driver ประเภทนี้ได้แก่ Oracle Call Interface (OCI) driver

2.5.3.3 JDBC-Net pure Java Driver

เป็น drivers ที่สร้างด้วยภาษาจาวาทั้งหมด โดย drivers จะแปลคำสั่งจาก JDBC เป็นคำสั่งที่ใช้งานผ่าน network ไปที่ server ของระบบฐานข้อมูล จากนั้น server จะแปลคำสั่งที่ได้รับเป็นคำสั่งสำหรับเชื่อมต่อกับฐานข้อมูล ข้อดีคือ drivers ประเภทนี้ถูกสร้างขึ้นด้วยภาษาจาวาทั้งหมด จึงสามารถถูกดาวน์โหลดไปยังเครื่องฝั่ง client โดยอัตโนมัติ แต่ก็มีข้อเสียคือ เพิ่มความยุ่งยากในการจัดการด้านความปลอดภัย ตัวอย่าง driver ประเภทนี้ได้แก่ Web logic ของ Tengah

2.5.3.4 Native protocol pure Java Driver

เป็น drivers ที่สร้างด้วยภาษาจาวาทั้งหมดเหมือน JDBC-Net pure Java Driver แต่ drivers จะทำหน้าที่ติดต่อกับระบบฐานข้อมูลโดยตรง ทำให้มีประสิทธิภาพการทำงานมาก เรามักเรียก drivers ประเภทนี้ว่า thin drivers ระบบฐานข้อมูลที่มี drivers ประเภทนี้ได้แก่ Oracle , Sybase , Informix , IBM , DB2 , Inprise Interbase และ Microsoft SQL Server

ข้อแนะนำในการเลือก drivers ให้เหมาะสมกับงาน มีดังนี้

1. ถ้าระบบของเราใช้ระบบฐานข้อมูลหนึ่งอยู่แล้ว เราก็ต้องใช้ drivers ที่มีมากับระบบฐานข้อมูลนั้น ซึ่งโดยปกติเกือบทุกประเภทจะให้ฟรี และอาจมีรุ่นปรับปรุงใหม่ให้ดาวน์โหลดได้เรื่อยๆ หากไม่จำเป็นไม่ควรใช้ drivers ที่ผลิตโดยบริษัทที่ไม่ใช่ผู้ผลิตระบบฐานข้อมูลนั้น
2. การเลือกประเภทของ drivers มีผลเฉพาะในตอนใช้งานโปรแกรม แต่ในระหว่างพัฒนาโปรแกรมเราจะใช้ drivers ชนิดใดก็ได้ หากในตอนใช้งานเราต้องการความเร็วควรเลือก driver ประเภทที่เป็น native API หรือ native protocol หากเราไม่สามารถติดตั้งโปรแกรมที่เครื่องของ clients ควรเลือกใช้ drivers ประเภทที่เป็น pure Java เพื่อให้สามารถดาวน์โหลดไปที่เครื่องของ clients ได้ง่าย

2.5.4 ข้อดีของ JDBC

1. ใช้งานง่าย การตั้งค่าไม่ยุ่งยาก
2. สามารถทำงานร่วมกับภาษา Java ได้เป็นอย่างดี เพราะ Driver ที่เขียนขึ้นนั้นส่วนมากจะพัฒนาจากภาษา Java และมีการ compile เป็น class file ทำให้ใช้งานง่าย
3. สามารถติดต่อฐานข้อมูลได้เกือบทุกประเภท ซึ่งส่วนใหญ่จะเป็นฐานข้อมูลแบบเชิงสัมพันธ์
4. ไฟล์ Driver มีขนาดเล็กกะทัดรัด
5. สามารถใช้งานได้หลายระบบปฏิบัติการ
6. เป็น low-level Interface และสามารถใช่ High-level interface ด้วยถ้าต้องการ
7. การใช้งานจะเป็นเชิง Object-Oriented
8. Driver แต่ละตัวมีมาตรฐานใกล้เคียงกัน

2.6 การเชื่อมต่อระบบฐานข้อมูล MySQL และ JDBC drivers

สำหรับระบบฐานข้อมูล MySQL จะต้องดาวน์โหลดไฟล์สำหรับ JDBC driver ของ MySQL จาก WWW.mysql.org เป็นไฟล์อย่างเช่น mysql-connector-java-3.1.8-bin.zip (ซึ่งเป็นรุ่น 3.0.9) แล้วทำการ unzip จะได้ไดเรกทอรี mysql-connector-java-3.1.8-bin ที่มีไฟล์ mysql-connector-java-3.0.9-stable-bin.jar ซึ่งเป็นไฟล์ jar ที่มี JDBC driver สำหรับ MySQL จะต้องกำหนดค่า classpath ให้ชี้มาที่ไฟล์นี้ แต่เราจะทำงานที่ไดเรกทอรี c:\project ดังนั้นเพื่อความสะดวก จึงก๊อปปี้ไฟล์นี้ไปที่ไดเรกทอรีที่ทำงาน และกำหนด classpath โดยเปลี่ยนไดเรกทอรีไปที่ c:\project แล้วพิมพ์

```
Set classpath=mysql-connector-java-3.1.8 -bin.jar;
```

คลาสที่เป็น JDBC driver ของ MySQL ชื่อ com.mysql.jdbc.Driver
สังเกตว่า database url สำหรับ MySQL จะอยู่ในรูปแบบ

```
Jdbc:mysql://[hostname][,failoverhost...][:port]/[dbname][?param1=value1][&param2=value2].....
```

ระบบฐานข้อมูลโดยทั่วไปมีขั้นตอนการสร้างการติดต่อ รวมทั้งการส่งคำสั่ง SQL ออกไป และรับผลลัพธ์กลับมาเหมือนกัน เพียงมีชื่อ drivers และวิธีอ้างถึงชื่อ databases ที่แตกต่างกันไปในแต่ละประเภท

2.7 Java Telephony API (JTAPI)

Java Telephony API (JTAPI) เป็น interface ที่ช่วยในการเขียนโปรแกรมแบบ object-oriented สำหรับการเขียนโปรแกรมที่เป็น computer telephony ด้วยภาษา Java ซึ่งประกอบด้วยกลุ่มของ class และ interface ที่ประกอบกันขึ้นเป็น Java extension package อยู่ในชื่อ javax.* โดย JTAPI เป็นส่วนติดต่อที่อยู่ระหว่างโปรแกรม กับระบบโทรศัพท์

JTAPI สนับสนุนการทำงานของผู้ใช้ได้อย่างกว้างขวาง ตั้งแต่ผู้พัฒนา call center application ไปจนถึงนักออกแบบ web page อีกทั้งยังสนับสนุนการควบคุมการเรียกจากแบบบุคคลที่หนึ่ง (first-party call control) และแบบบุคคลสาม (third-party call control) JTAPI ถูกออกแบบมาให้ทำให้การเขียนโปรแกรมมีความง่ายและไม่ซับซ้อน อย่างไรก็ตาม JTAPI ก็ยังมีคุณลักษณะที่สำคัญที่จะนำไปใช้ในการพัฒนาโปรแกรมที่เป็น computer telephony ที่มีความซับซ้อนได้

ในความเป็นจริงแล้ว Java Telephony API เป็นกลุ่มของ API โดยจะมี core API ที่ให้บริการรูปแบบพื้นฐานของการโทรศัพท์และคุณสมบัติเบื้องต้นของโทรศัพท์ เช่น การวางโทรศัพท์ การรับโทรศัพท์ เป็นต้น โดย core API ถูกล้อมรอบด้วย API อื่นๆ ที่ให้บริการคุณสมบัติต่างๆ ที่เป็นลักษณะเฉพาะของโทรศัพท์ เช่น call center เป็นต้น โดยโครงสร้างของ core และ package อื่นๆ จะอธิบายภายหลัง

application ที่เขียนขึ้น โดยใช้ Java Telephony API นั้นสามารถใช้งานได้หลาย platform และระบบโทรศัพท์ที่แตกต่างกันได้ อีกทั้งยังสามารถทำงานร่วมกับ computer telephony API แบบอื่นๆ ได้ เช่น Sun Microsystem's SunXTL, Microsoft and Intel's TAPI, Novell and Lucent's TSAPI และ IBM's CallPath เป็นต้น

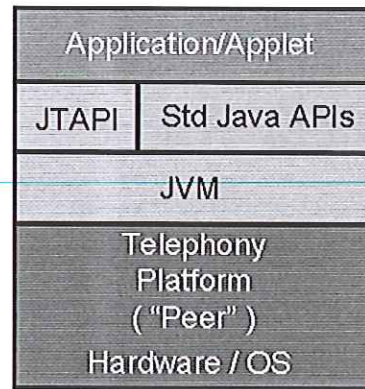
2.7.1 สถาปัตยกรรมของ JTAPI

JTAPI สามารถนำไปใช้งานได้ทั้งในระบบโทรศัพท์ที่มีเครือข่ายที่เป็นส่วนตัวและเครือข่ายที่เป็นสาธารณะ สามารถทำงานข้ามรูปแบบของระบบโทรศัพท์ได้ เพื่อที่จะให้ได้ถึงคุณสมบัตินี้ JTAPI ได้ใช้ความคิดในเชิงสถาปัตยกรรม 2 แบบ คือ

1. JTAPI Peer
2. Core Plus Extension Architecture

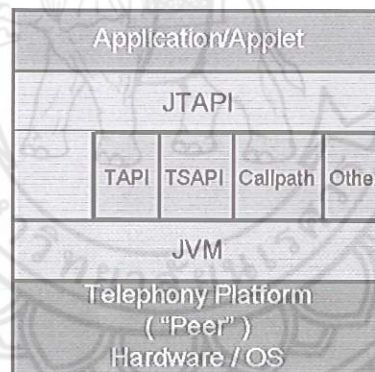
2.7.1.1 JTAPI Peer

การพัฒนาโปรแกรมของ JTAPI สำหรับระบบโทรศัพท์ที่เฉพาะเจาะจงนั้นถูกเรียกว่า peer ตัวอย่างของ peer ในระบบโทรศัพท์ เช่น PBX systems, media servers, call center servers เป็นต้น ซึ่ง JTAPI, standard Java APIs and the JVM (Java Virtual Machine) นั้นถูกพัฒนาขึ้นที่ส่วนบนของระบบโทรศัพท์ ดังรูปที่ 2.4 โดย peer นั้นแสดงถึงระบบปฏิบัติการและ hardware ของรูปแบบนั้น และ application เข้าถึงคุณสมบัติของ peer ผ่านทางการเขียนโปรแกรมติดต่อกับ JTAPI และ JVM



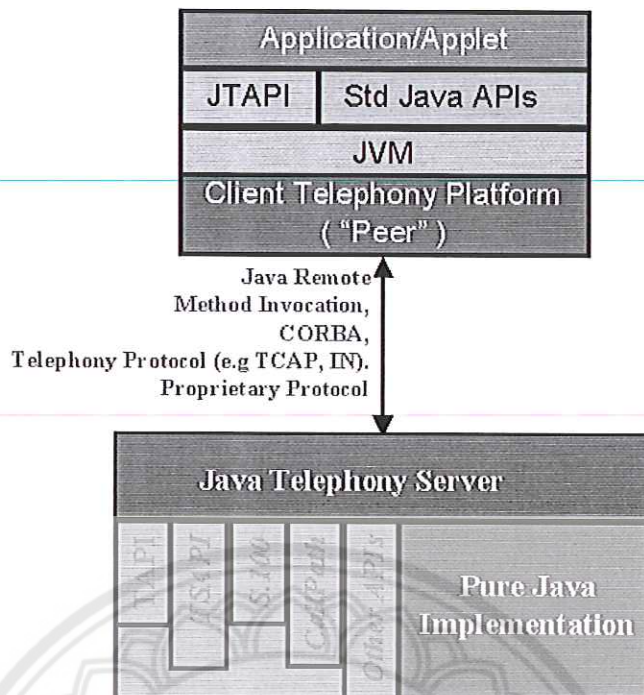
รูปที่ 2.4 แสดง JTAPI Peer

ในรูปที่ 2.4 นั้นแสดงให้เห็นถึงความไม่ซับซ้อน, จุดสำคัญ, และแบบแผนของ JTAPI ในการพัฒนาโปรแกรม แต่ JTAPI ไม่ได้ถูกจำกัดอยู่เพียงรูปแบบนี้เท่านั้น รูปแบบของ JTAPI peer สามารถมี telephony API ที่เป็นมาตรฐานแบบอื่นๆ เช่น TAPI, TSAPI หรือ API ที่ถูกเขียนขึ้น โดยเฉพาะ เช่น Nortel Networks Meridian Link, IBM Callpath, Lucent Passageway รวมอยู่ด้วยได้ ดังรูปที่ 2.5 โดย JTAPI จะทำงานอยู่บน API เหล่านี้



รูปที่ 2.5 แสดงการรวม API อื่นๆ ไว้ใน JTAPI

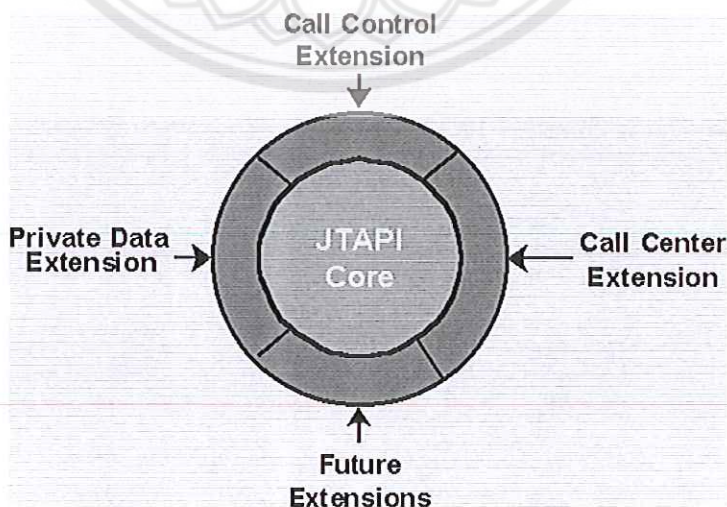
สถาปัตยกรรมของ JTAPI ยังอนุญาตให้ JTAPI สามารถทำงานในสถานะแวดล้อมที่อยู่ต่างที่กันได้ ดังรูปที่ 2.6



รูปที่ 2.6 แสดงการทำงานของ JTAPI ที่มีสถานะแวดล้อมที่อยู่ต่างที่กัน

2.7.1.2 Core Plus Extension Architecture

เป็นการกำหนดจำนวน package ที่น้อยที่สุดที่แสดงถึงหน้าที่ของ core ที่ JTAPI สามารถจะมีได้ โดยหน้าที่ของ core จะให้บริการเกี่ยวกับหน้าที่ของสร้างและตอบในระบบการเรียกจากแบบบุคคลที่สอง (two party call) สำหรับ application ที่ต้องการเพิ่มการโทรหรือการควบคุมการไหลของข้อมูล ทำได้โดยการเพิ่ม package ที่ขยายออกมาจากการติดต่อกับ core โดยรวมอยู่ในการส่วนเพิ่มเติมของ JTAPI เป็นเหมือนความต้องการ ดังรูปที่ 2.7



รูปที่ 2.7 แสดง Core Plus Extension Architecture

package ที่ขยายออกมาไม่ได้จำกัดเฉพาะการขยายออกมาจากหน้าที่ในปัจจุบันของ core เท่านั้น แต่ในความเป็นจริง package ส่วนใหญ่ขยายหน้าที่ของ core และเพิ่มหน้าที่โดยไม่ใช่หน้าที่ในปัจจุบันของ core เช่น media package ใน JTAPI 1.3 สามารถพัฒนาโดยไม่เกี่ยวข้องกับ core

2.7.2 Packages ของ JTAPI

Java Telephony API ประกอบขึ้นจากกลุ่มของ package ที่เขียนด้วยภาษา Java แต่ละ package จะให้บริการในส่วนเฉพาะของหน้าที่สำหรับรูปแบบของ computer-telephony application โดยการพัฒนาโปรแกรมโดย JTAPI ส่วนใหญ่มักจะต้องการ core package ส่วน package อื่นๆ เป็นส่วนเพิ่มเติม ในการพัฒนา telephony server ก็จะเลือก package ที่สนับสนุนรูปแบบและ hardware ของระบบ

package ทั้งหมดของ JTAPI ที่อยู่ในลำดับชั้นที่ชื่อ javax.telephony โดย package ที่กำหนดไว้ใน JTAPI 1.3 มีดังนี้

ตารางที่ 2.2 แสดง package ทั้งหมดที่กำหนดไว้ใน JTAPI 1.3

Core package	javax.telephony
Call Control	javax.telephony.callcontrol
Call Center	javax.telephony.callcenter
Media	javax.telephony.media
Mobile	javax.telephony.mobile
Phone	javax.telephony.phone
Private Data	javax.telephony.privatedata

2.7.2.1 Core

javax.telephony package หรือ core package เป็นส่วนหลักที่ใช้ในติดต่อกับส่วนอื่นๆ ให้บริการโครงสร้างพื้นฐานของรูปแบบการโทรและคุณสมบัติเบื้องต้นของโทรศัพท์ โดยคุณสมบัติเหล่านี้รวมทั้งการวางโทรศัพท์, การรับโทรศัพท์ และการหยุดการติดต่อ telephony application ที่ไม่มีความซับซ้อนมักจะต้องการใช้เพียง core package เท่านั้นที่นำไปใช้ในงานเหล่านั้น โดยไม่ต้องการเกี่ยวข้องกับรายละเอียดของ package อื่นๆ

core package ประกอบด้วย

- วิธีการควบคุมการโทรพื้นฐาน
 - createCall()
 - connect()
 - answer()
 - disconnect()
- ความสามารถสำหรับแต่ละวิธีการเหล่านี้
- การเปลี่ยนแปลงของ state ทั่วไปที่เกิดจากเหตุการณ์ที่ตรงกัน

2.7.2.2 Call Control

javax.telephony.callcontrol package ขยายมาจาก core package โดยให้บริการในการควบคุมการโทรที่มีความสามารถสูงขึ้นและมีรายละเอียดรูปแบบของ state ในการโทรมากขึ้น โดยเพิ่มคุณสมบัติการโทร ดังนี้

- conference calling
- transfer
- forwarding
- call hold
- call join
- message waiting
- do not disturb
- call park & pickup

2.7.2.3 Call Center

javax.telephony.callcenter package ให้บริการกับ application ที่มีความต้องการที่จะทำงานเกี่ยวกับระบบ call center โดยคุณสมบัติที่จำเป็นในการควบคุมระบบ call center ที่มีขนาดใหญ่ ตัวอย่างของคุณสมบัตินี้ ได้แก่

- ACD agent support
- Routing
- Automated Call Distribution (ACD)
- Predictive Calling
- Application Data (associated with telephony objects.)

2.7.2.4 Media

javax.telephony.media package ให้บริการ application ให้มีความสามารถในการใช้ media stream ที่เกี่ยวข้องกับโทรศัพท์ สำหรับ JTAPI 1.3 ใช้ media API แบบใหม่โดยการเพิ่มรายละเอียดของ ECTF S.410 ใน version นี้ได้ถูกออกแบบมาให้มีความสมบูรณ์เพื่อจะมาแทนที่ media package ที่กำหนดไว้ใน JTAPI 1.2 โดยจะทำการจับคู่รูปแบบของ ECTF ของ CTI server ที่สนับสนุน multi-vendor application และการแบ่งทรัพยากรในสถานะแวดล้อม แต่ก็ยังสามารถทำงานได้บนสถานะแวดล้อมที่มีผู้ใช้คนเดียวได้ โดยยังแสดงถึง object-oriented API ที่เป็นพื้นฐานของ ECTF S.100 API อยู่ด้วย

media package ของ JTAPI 1.3 นั้นใช้สถาปัตยกรรมของทรัพยากรในการที่จะให้บริการ media service สำหรับ

- Players
- Recorders
- Signal Detectors
- Signal Generators

โดยยอมให้มีการเพิ่มทรัพยากรใหม่ๆ เข้าไปในคุณสมบัติของ JTAPI ได้ง่ายทรัพยากรสำหรับ fax server และ automatic speech recognition อยู่ใน JTAPI 1.3 นี้ด้วย

class ทั้งหมดใน media package ของ JTAPI 1.2 ถูกแทนที่ด้วย JTAPI 1.3 โดย media package ใหม่นี้ได้กำหนดให้มีความแน่ใจว่าไม่มีชื่อที่ซ้อนกัน อนุญาตให้ทั้งสอง package อยู่ร่วมกันสำหรับช่วงเวลาที่มีการเปลี่ยนแปลง

2.7.2.5 Mobile

javax.telephony.mobile package ให้บริการคุณสมบัติที่ต้องการสำหรับ application ที่ทำงานบนโทรศัพท์มือถือหรือบนเครือข่ายโทรศัพท์มือถือ mobile package ขยายออกมาจาก core package โดยมีหน้าที่เกี่ยวกับเครือข่ายโทรศัพท์มือถือ ดังนี้

- Mobile Call Control
- Mobile Network Access and Control
- Media Services for Mobile Applications

รูปแบบของโทรศัพท์ที่สนับสนุน application เหล่านี้จากการให้บริการของ JTAPI โดยโทรศัพท์มือถือยังรวมถึง

- Individual mobile telephone sets (wireless)
- Individual mobile data-only devices (pagers)
- GSM, TDMA and CDMA mobile stations

2.7.2.6 Phone

javax.telephony.phone package ขอมให้ application ควบคุมคุณสมบัติในระดับ physical ของอุปกรณ์ hardware ของโทรศัพท์ ในการพัฒนาจะอธิบายถึง terminal เป็นเหมือนการรวมกันของส่วนประกอบ โดยส่วนประกอบแต่ละชนิดจะมีการติดต่อกับ package นี้ ซึ่งให้บริการในการติดต่อกับแบบอย่าง ดังนี้

- buttons
- displays
- lamps
- ringers
- hook switches
- speakers
- microphones

2.7.2.7 Private Data

javax.telephony.privatepackage ขอมให้ application สามารถติดต่อโดยตรงกับข้อมูลภายใต้ hardware switch โดยข้อมูลเหล่านี้ถูกใช้ใน โครงสร้างของ switch เพื่อที่จะทำการเลือกการทำงานเฉพาะ ซึ่ง application มักจะใช้ package นี้ในการเก็บข้อมูลไว้เป็นส่วนหนึ่งของ object ใน Java Telephony API

2.7.3 Call Model

call model ประกอบด้วย ส่วนหลัก 6 ส่วนในการควบคุมการ โทร, เหตุการณ์ในการ โทรและการเพิ่มความสามารถให้กับการ โทร ซึ่งสามารถอธิบายได้ดังนี้

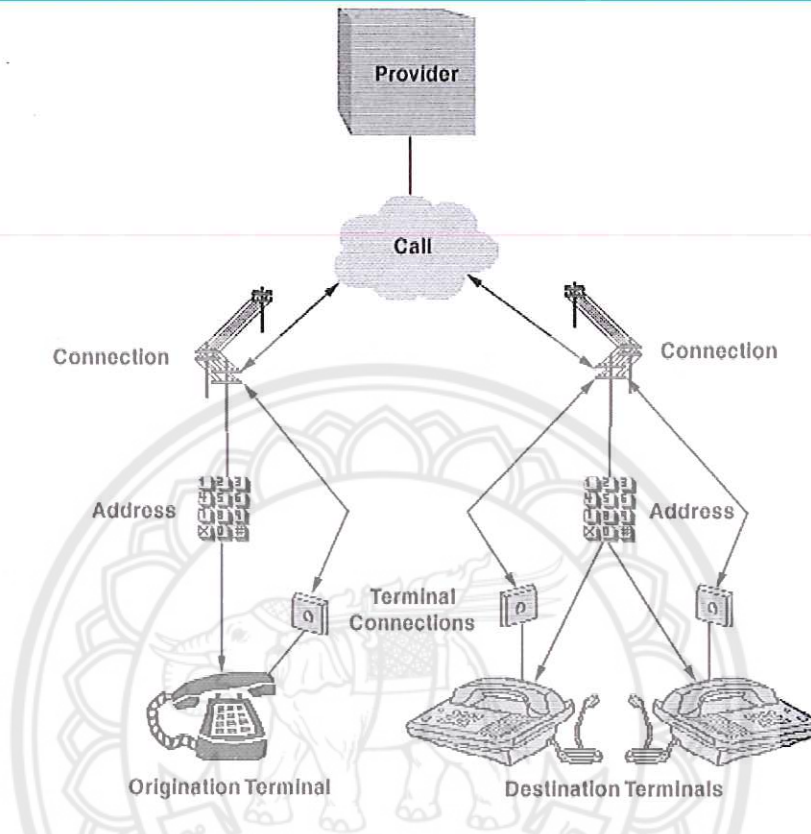
2.7.3.1 Core Objects

call model ที่ใช้ใน JTAPI ประกอบด้วยส่วนในการควบคุมการ โทร 6 ส่วน คือ

- Provider
- Call
- Address
- Connection
- Terminal
- TerminalConnection

โดย object เหล่านี้ถูกกำหนดให้ใช้ใน core package แต่ละ object นั้นแสดงถึง physical หรือ logical ที่มีอยู่จริงในระบบโทรศัพท์ จุดประสงค์ของ call model คือการอธิบายถึงการโทรศัพท์และจุดสิ้นสุดที่เกี่ยวข้องกับการโทรศัพท์ object เหล่านี้มีความเกี่ยวข้องกับส่วนอื่นๆ ไม่ทางใดก็ทางหนึ่ง ดังที่อธิบายไว้ในเรื่อง core package

ผังรูปที่ 2.8 แสดงถึง JTAPI call model และ object ที่เป็นส่วนประกอบของ call model ซึ่งจะได้อธิบายแต่ละ object ดังนี้



รูปที่ 2.8 แสดง JTAPI call model และ object ต่างๆ ใน call model

2.7.3.1.1 Provider

Provider เป็นตัวแทนของโปรแกรมที่เป็น service provider ซึ่งจะควบคุมการติดต่อ PBX กับ server, โทรศัพท์/fax card ในคอมพิวเตอร์ หรือเทคโนโลยีในระบบ computer networking เช่น IP โดย Provider จะซ้อนรูปแบบของระบบย่อยของโทรศัพท์ และอนุญาตให้ Java application และ applet ทำการติดต่อกับระบบย่อยของโทรศัพท์ ในอุปกรณ์ที่เป็นอิสระจากส่วนอื่นๆ

2.7.3.1.2 Call

Call แสดงถึงการโทรที่มีการไหลของข้อมูลระหว่าง service provider และผู้ที่มีส่วนร่วมในการโทร การโทรศัพท์ประกอบด้วย Call object และ connection ซึ่งจะมีมากกว่าหนึ่ง connection ก็ได้ เช่น ในแบบ two-party call มีหนึ่ง Call object และสอง ในการประชุมทางโทรศัพท์ก็จะมีสาม connection หรือมากกว่าที่เกี่ยวข้องกับหนึ่ง

2.7.3.1.3 Address

Address object แสดงถึงหมายเลขโทรศัพท์ แทน logical endpoint ของการโทรซึ่งแตกต่างจาก physical endpoint ในความเป็นจริงหนึ่ง address สามารถมีได้หลาย physical endpoint

2.7.3.1.4 Connection

Connection object เป็นรูปแบบการติดต่อกันระหว่าง Call object และ Address object ความสัมพันธ์นี้เกี่ยวกับมุมมอง logical เพราะว่าเป็นความเกี่ยวข้องกับความสัมพันธ์ของ Call และ Address Connection objects มักจะอยู่ใน state ใด state หนึ่งที่แสดงถึงความสัมพันธ์ระหว่าง Call และ Address ในขณะนั้น ซึ่ง state เหล่านี้จะได้อธิบายภายหลัง

2.7.3.1.5 Terminal

Terminal object แสดงถึงอุปกรณ์เช่น โทรศัพท์รวมถึงคุณสมบัติของอุปกรณ์นั้น แต่ละ Terminal object มักจะมีหนึ่ง Address หรือมากกว่าหนึ่ง Address ก็ได้ที่เกี่ยวข้องกับมัน เช่นในระบบโทรศัพท์ของบริษัทต่างๆ Terminal object เป็น physical endpoint ของการโทรศัพท์เนื่องจากมีลักษณะเป็นส่วนของ hardware

2.7.3.1.6 TerminalConnection

TerminalConnection objects นั้นเป็นรูปแบบของความสัมพันธ์ระหว่าง Connection และ physical endpoint ของ Call (Terminal object) โดยความสัมพันธ์นี้ เป็นความรู้ในมุมมองของ physical ของ Connection TerminalConnection อธิบายถึง state ของความสัมพันธ์ระหว่าง Connection และ Terminal. ซึ่ง state ของ TerminalConnection จะได้อธิบายในภายหลัง

2.7.3.2 Capabilities

สถานะแวดล้อมของระบบโทรศัพท์นั้นกว้างมากในความสามารถที่จะทำงานได้ การเปลี่ยนแปลงที่เกิดขึ้นในระบบโทรศัพท์มักจะมีผลกระทบในการพิจารณาในด้านเทคนิคหรือการบำรุงรักษา แต่ JTAPI ไม่ได้สิ่งไหนที่มีผลจากการเปลี่ยนแปลง อะไรที่ JTAPI ทำการให้บริการกลไกสำหรับการหา application ที่สามารถทำงานได้บนสถานะแวดล้อมปัจจุบัน กลไกที่นำมาจาก telephony API ต่างๆ นั้นเรียกว่า Capabilities

javax.telephony.capabilities package นั้นรวมกลไกที่อนุญาตให้ application ชักถามการกระทำที่จะเกิดขึ้น โดย Capabilities มี 2 รูปแบบ คือ

- **Static** แสดงถึงการพัฒนาหรือแบบเฉพาะ object ของ call model ซึ่งสนับสนุนวิธีการใดๆ
- **Dynamic** แสดงถึงการกระทำที่อนุญาตบน object ของ call model ในแต่ละ state ขณะนั้นของ call model

Capabilities เป็นวิธีการแบบ one-to-one ถ้าวิธีการไม่ได้ถูกพัฒนาในสถานะแวดล้อมทั้งหมด capability จะอนุญาตให้ application ตรวจสอบว่ามันสามารถที่จะทำงานในเวลานี้หรือไม่ ในทุกๆ core object นั้นเกี่ยวข้องกับ capability object ดังนี้

4900038

- ProviderCapabilities ยธ.
- CallCapabilities ก2765
- AddressCapabilities 2578.
- ConnectionCapabilities
- TerminalCapabilities
- TerminalConnectionCapabilities

เพราะว่า call model object ใน package ที่ขยายออกมานั้นสนับสนุนการเพิ่มวิธีการ package เหล่านี้ก็จะมีความเกี่ยวข้องกับ capability object ในลักษณะเหมือนกันกับวิธีการเหล่านี้เป็นอย่างดี

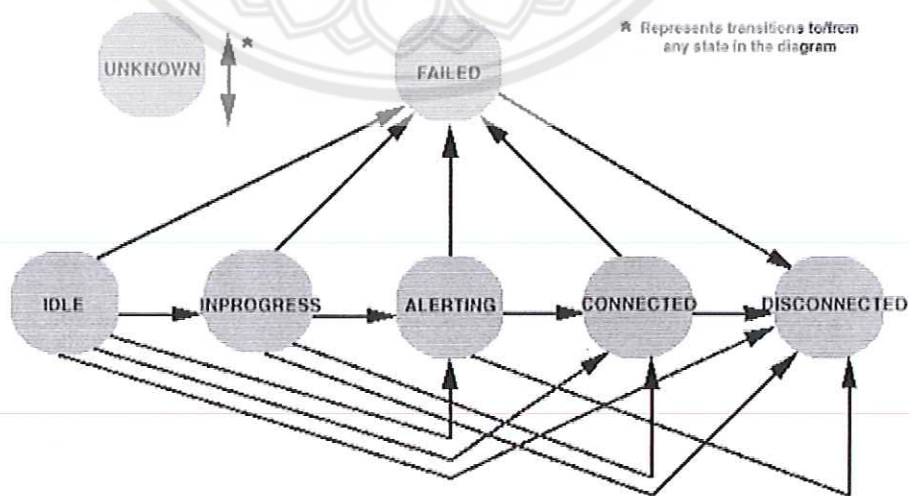
2.7.3.3 Events

JTAPI application นั้นต้องการที่จะรู้ state ที่เปลี่ยนไปใน call model object นั้นที่ไม่ได้เป็นผลของสิ่งต่างๆที่ application ทำ เช่น application ต้องการที่จะรู้ว่าว่ามีโทรศัพท์โทรเข้ามาถึงหรือยัง เพื่อที่จะได้ทำการเปลี่ยนมุมมองของผู้ใช้เกี่ยวกับ terminal บน graphical user interface

JTAPI สนับสนุน event model 2 แบบ คือ

- Listener Event Model
- Observer Event Model

2.7.4 Connection States



รูปที่ 2.9 Connection State Transitions

- **IDLE state**

IDLE state เป็น is state เริ่มต้นสำหรับการมี Connection object ใหม่ โดยปกติ Connection จะเปลี่ยนจาก IDLE state ไปสู่ state อื่นอย่างรวดเร็ว Connection ใน IDLE state แสดงถึงว่ามีการเข้าร่วมใน party ของ telephone call ในรูปแบบใดรูปแบบหนึ่ง ที่ใช้ได้กับ Connection ใน state นี้

- **INPROGRESS state**

INPROGRESS state แสดงถึง telephone call ในขณะนั้นกำลังจะไปถึงตำแหน่งของปลายทาง

- **ALERTING state**

ALERTING state แสดงถึงปลายทางของ telephone call พร้อมที่จะมีการโทรเข้าได้

- **CONNECTED state**

CONNECTED state แสดงถึง party นั้นทำการ active เป็นส่วนหนึ่งของ telephone call Connection ใน CONNECTED state นำไปใช้ใน party ที่กำลังสนทนาอยู่กับ party อื่นๆ ในการโทร หรือถูกติดต่อไปถึงเสียง

- **DISCONNECTED state**

DISCONNECTED state แสดงถึง party นั้นไม่ได้เป็นส่วนหนึ่งของ telephone call ไม่มี method ใดที่ใช้ได้กับ Connection ใน DISCONNECTED state

- **FAILED state**

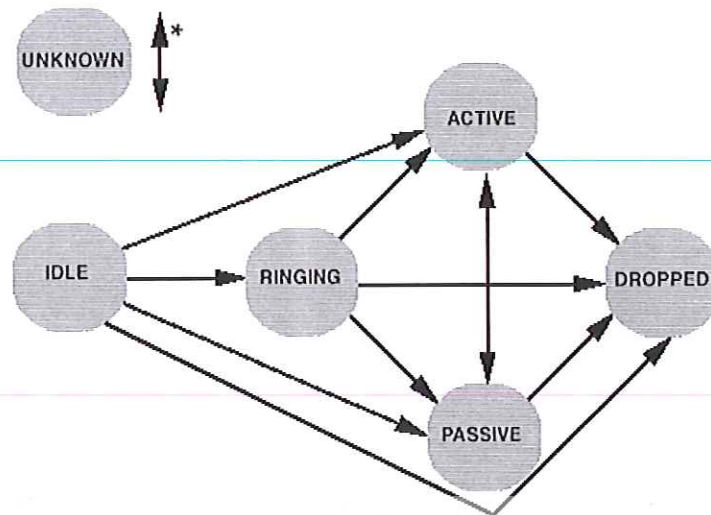
FAILED state แสดงถึง telephone call เกิดข้อบกพร่องในการติดต่อกับปลายทาง ตัวอย่างเช่น ถ้ามี application ใช้คำสั่ง Call.connect() เพื่อจะทำการติดต่อจาก telephone call ไปยัง party ที่สายไม่ว่าง Connection ที่เกี่ยวข้องกับ party นั้นก็จะถูกเปลี่ยนไปอยู่ในไปอยู่ใน FAILED state

- **UNKNOWN state**

UNKNOWN state แสดงถึง Provider ไม่สามารถหา state ให้กับ Connection ในเวลานั้นได้ Connection ก็จะเปลี่ยน state ไปสู่ UNKNOWN state ได้ตลอด นอกจากว่า Connection อยู่ใน DISCONNECTED หรือ FAILED state ผลจาก method ต่างๆ บน Connection ใน state นี้ไม่สามารถคาดเดาได้

2.7.5 TerminalConnection Object States

TerminalConnection object แสดงถึงความสัมพันธ์ระหว่าง Terminal และ Connection ดังที่ได้กล่าวไว้ก่อนหน้านี้ object เหล่านี้แสดงถึง physical view ของ Call อธิบายถึง physical endpoint ของ Terminal ที่เป็นส่วนหนึ่งของ telephone call เช่นเดียวกับ Connection objects TerminalConnection object มีกลุ่มของ state ของมันเอง และการเปลี่ยนแปลง state ดังรูปที่ 2.10 การเปลี่ยนแปลงของ state อธิบายได้ ดังนี้



รูปที่ 2.10 Terminal Connection State Diagram

- **IDLE state**

IDLE state เป็น state เริ่มต้นสำหรับ TerminalConnection object ทั้งหมด มีความหมายเช่นเดียวกับ IDLE state ใน Connection object

- **ACTIVE state**

ACTIVE state แสดงถึง Terminal นั้น active เป็นส่วนหนึ่งของ telephone call

- **RINGING state**

RINGING state แสดงถึง Terminal กำลังส่งสัญญาณไปถึงผู้ใช้ที่กำลังโทรเข้าในขณะที่ Terminal

- **DROPPED state**

DROPPED state แสดงถึง Terminal เคยเป็นส่วนหนึ่งของ telephone call แต่ถูกหยุดแล้ว DROPPED state เป็น state สุดท้ายของทุก TerminalConnections.

- **PASSIVE state**

PASSIVE state แสดงถึง Terminal นั้นเป็นส่วนหนึ่งของ telephone call แต่ไม่ได้ทำการ , but active TerminalConnection ใน PASSIVE state แสดงถึงทรัพยากรบน Terminal ที่เคยถูกใช้ใน telephone call หลายๆ Package ให้บริการคุณสมบัติของ Terminals เพิ่มขึ้นในการรวมกับการโทรต่างๆ ใน PASSIVE state

- UNKNOWN state

UNKNOWN state แสดงถึง Provider ไม่สามารถหา state ในเวลาปัจจุบันให้กับ Terminal Connection ได้ มีลักษณะเดียวกับ UNKNOWN state ใน Connection object's UNKNOWN state.

2.7.6 Interface MediaCallObserver

MediaCallObserver ขยายออกมาจาก CallObserver interface และรายงานผลของเหตุการณ์ที่เกี่ยวข้องไปยัง MediaTerminalConnection object เหตุการณ์สำหรับ object นี้ถูกรายงานไปยัง ใน core package เพราะว่าเหตุการณ์ของ TerminalConnection ถูกรายงานไปยัง CallObserver object

Interface นี้ไม่มี method ใดๆ เลย เหตุการณ์ทั้งหมดสำหรับ MediaTerminalConnection object จะถูกรายงานไปยัง method ที่ชื่อ callChangedEvent() ใน CallObserver interface เหตุการณ์ของ Media TerminalConnection ขยายมาจาก TermConnEv interface (ขยายจาก CallEv interface)

2.7.6.1 callChangedEvent()

จะรายงานเหตุการณ์ทั้งหมดที่เกิดขึ้นที่เกี่ยวกับ Call object โดย method ส่ง array ของ CallEv objects ในรูปแบบของ arguments ที่มีลักษณะเช่นเดียวกับรายการของเหตุการณ์ที่แสดงถึงการเปลี่ยนแปลงของ Call object ไปยัง Connection และ TerminalConnection object ที่เกี่ยวข้องกับ Call object นี้

2.8 XTAPI

XTAPI เป็นการทำให้ JTAPI มีความง่ายมากขึ้น ให้สามารถนำมาใช้งานได้จริง ผู้ที่มีความรู้ที่สามารถทำงานกับ JTAPI โดยตรงมีไม่มาก จึงได้มีการพัฒนา XTAPI ขึ้น โดยมีเป้าหมายในการให้บริการ Service Provider Interface สำหรับ JTAPI XTAPI มีความต้องการที่จะให้บริการ code พื้นฐานสำหรับการพัฒนา JTAPI XTAPI เป็นลิขสิทธิ์ภายใต้ GPL license แต่อย่างไรก็ตามก็ยังมีช่องโหว่ที่ยังทำผู้ทำธุรกิจเกี่ยวกับโทรศัพท์ที่สามารถใช้งาน XTAPI

ในปัจจุบัน XTAPI ให้บริการเกี่ยวกับหน้าที่ใน Core และ Media package XTAPI ยังกำหนดความรู้เกี่ยวกับ IXTAPI Service Provider Interface ที่ซึ่งมีไม่มาก แต่ก็เพียงพอสำหรับหน้าที่ใน Core และ Media package XTAPI มี Service Provider Interface 2 แบบที่ได้สร้างขึ้นแล้ว คือ provider ที่ทำงานผ่าน JNI-Bridge ของ Microsoft's TAPI และ provider ที่ทำงานโดยตรงกับ serial modem ผ่านทาง JavaComm API ซึ่ง provider สามารถทำงานได้ทั้งบน Windows, Solaris และ Linux

2.8.1 Setting up XTAPI

ในที่นี้จะพูดถึง provider 2 แบบ คือ

1. Microsoft's TAPI
2. Serial Voice Modems.

2.8.1.1 MSTAPI

ถ้าหากใช้ MSTAPI provider ไม่ต้องทำการตั้งค่าใดๆ เพียงแต่ต้องแน่ใจว่า xtapi.dll สามารถจะพบได้โดย xtapi.jar ดังนั้น xtapi.dll ควรจะอยู่ใน path หรือ ใน directory ที่ทำงาน

XTAPI จะทำงานได้ดีภายใต้ระบบปฏิบัติการของ Microsoft แบบ 32-bit โดยมีข้อกำหนด ดังนี้

- Window 95 จะต้องทำการ update ด้วย Unimodem/V driver ซึ่งสามารถหาได้จาก Microsoft

- เฉพาะ voice modem ที่ XTAPI สนับสนุนเท่านั้นที่จะสามารถทำงานได้
- โดยปกติ voice modem ไม่สามารถทำงานได้ภายใต้ NT 4.0 ด้วย MSTAPI provider NT 4.0 ไม่ให้บริการเกี่ยวกับความสามารถของ voice modem

- ระบบปฏิบัติการ NT ใหม่ ๆ เช่น Windows 2000 และ XP สามารถทำงานได้ดี
- จะต้องสร้าง provider ที่เป็นการอ้างอิงถึง MSTAPI provider เช่น

```
○ myprovider = peer.getProvider(null);
```

การทำงานกับข้อมูลเสียงได้โดยการใช้ไฟล์ .wav ที่ต้องอยู่ในมาตรฐาน g.711และในการทำงานกับ MSTAPI provider ต้องมีการกำหนด class path ดังนี้

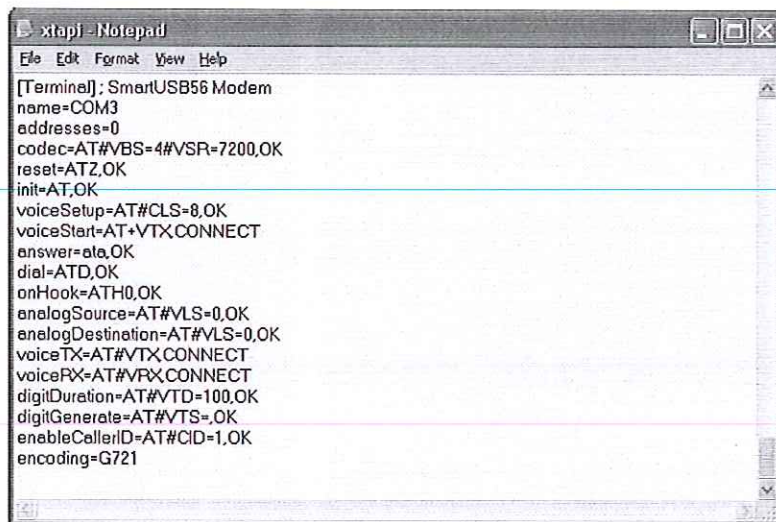
```
class path = ././xtapi.jar;./mstapisp.jar;./jtapi.jar;
```

2.8.1.2 Serial Provider

Serial provide ต้องการไฟล์ชื่อ xtapi.inf ซึ่งต้องเติมคุณสมบัติของโมเด็มที่ใช้ให้ถูกต้อง ในการใช้งาน Serial provide ต้องสร้าง provider เช่น

```
○ myprovider = peer.getProvider("Serial");
```

ส่วนมากไม่ใช่โมเด็มทั้งหมดที่ส่งค่าไฟล์ .inf สำหรับ Windows รูปแบบของ xtapi.inf มีรากฐานแตกต่างจากการใช้ใน Windows ซึ่งไฟล์ .inf ใน Windows สามารถจะหาได้ใน WinInf แต่ค่าต่างๆ เหล่านี้จะมีผลอย่างมากในการใช้ภายใต้ Linux xtapi.inf มีส่วนของแต่ละ JTAPI Terminal ที่มีค่าในรูปแบบของ Command,Response มีลักษณะดังรูปที่ 2.11



```

xtapi - Notepad
File Edit Format View Help
[Terminal]: SmartUSB56 Modem
name=COM3
addresses=0
codec=AT#VBS=4#VSR=7200,OK
reset=ATZ,OK
init=AT,OK
voiceSetup=AT#CLS=8,OK
voiceStart=AT+VTX.CONNECT
answer=ata,OK
dial=ATD,OK
onHook=ATH0,OK
analogSource=AT#VLS=0,OK
analogDestination=AT#VLS=0,OK
voiceTX=AT+VTX.CONNECT
voiceRX=AT#VRX.CONNECT
digitDuration=AT#VTD=100,OK
digitGenerate=AT#VTS=,OK
enableCallerID=AT#CID=1,OK
encoding=G721

```

รูปที่ 2.11 แสดงตัวอย่างของไฟล์ xtapi.inf

การทำงานกับข้อมูลเสียงได้โดยการใช้ 16-Bit Linear mono PCM 8Khz ที่เป็นไฟล์ .au และใน
การทำงานกับ Serial provider ต้องมีการกำหนด class path ดังนี้

```
class path = ./;./xtapi.jar;./serialsp.jar;./audiosp.jar;./jtapi.jar;
```

2.8.2 XTAPI Working Modem List

ตารางที่ 2.3 แสดงรายชื่อโมเด็มที่สามารถทำงานได้บน XTAPI

Modem	Win9x	Win2K	WinXP	Linux	ChipSet
Creative Labs Modem Blaster DI5630 (HCF)	M-S	M-S	M-S	X	?
Creative Labs Modem Blaster DE56251[1]	?-?	X-S	X-S	S	?
US Robotics Voice Faxmodem 5605	?-?	X-S	X-S	S	?
Zoom 2985 USB External	?-?	?-?	?-S	S	Lucent - Venus
Generic HSP56 MicroModem	?-?	?-?	M-S	?	PCTel HSP56
Generic	?-?	?-?	M-S	?	Intel V.92 HaM
Internal MSP3880SP-U	M-?	?-?	?-?	?	?
Smart USB (VD56US)	M-?	?-?	?-?	?	?
Momenta V.90 USB	M-?	?-?	?-?	?	?
USR Message Serial	M-?	?-?	?-?	?	?

หมายเหตุ : M=STAPI Provider (M) , S=Serial Provider (S) , unknown (?) , doesn't work (X)

บทที่ 3

การออกแบบและพัฒนาระบบ

3.1 การทำงานของระบบ

การทำงานของระบบได้ออกแบบให้ประกอบด้วยขั้นตอนการทำงาน 4 ส่วน คือ

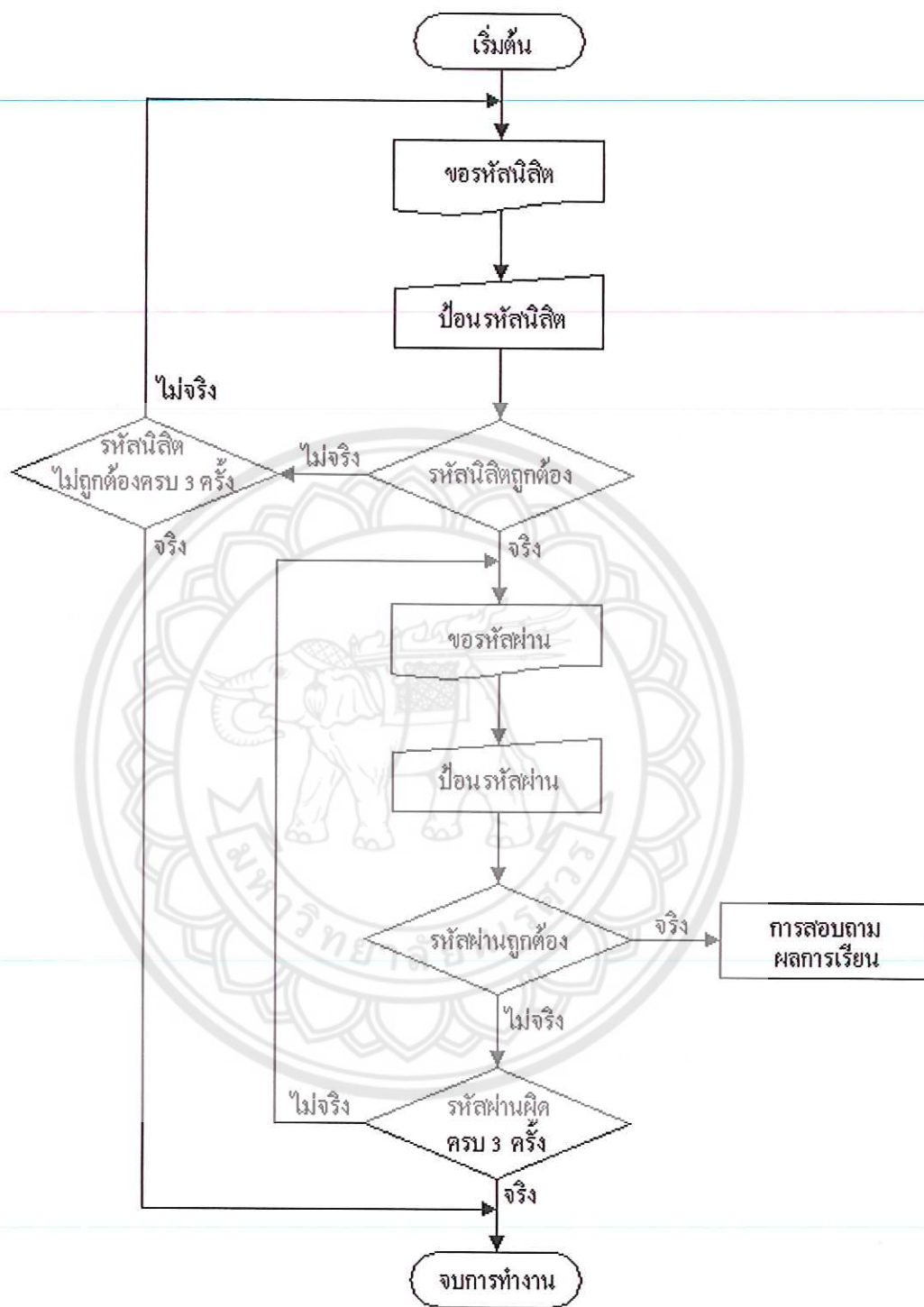
- การควบคุมขั้นตอนการทำงาน
- การจัดการฐานข้อมูล
- การจัดการข้อมูลเสี่ยง
- การจัดการเกี่ยวกับระบบโทรศัพท์



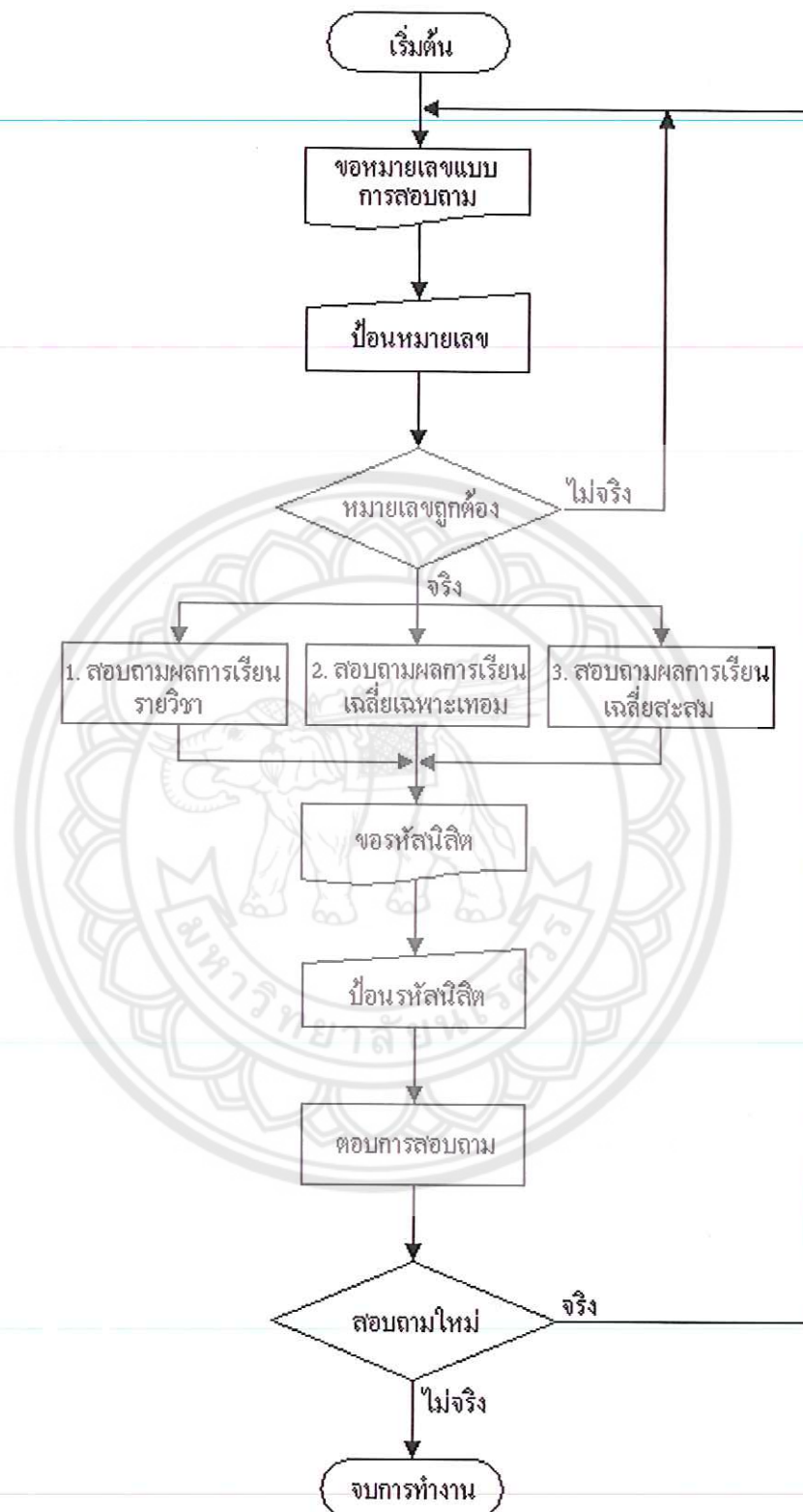
รูปที่ 3.1 แสดงส่วนประกอบของระบบสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์

3.1.1 การควบคุมขั้นตอนการทำงาน

การควบคุมขั้นตอนการทำงานเป็นส่วนที่เรียกใช้การทำงานของส่วนอื่นๆ เพื่อที่จะทำให้ระบบสามารถให้บริการการสอบถามจากผู้ใช้อย่างถูกต้อง เป็นลำดับขั้นตอน ดังรูปที่ 3.2 และ รูปที่ 3.3



รูปที่ 3.2 แสดงขั้นตอนการตรวจสอบรหัสนิติและรหัสผ่าน



รูปที่ 3.3 แสดงขั้นตอนการสอบถามผลการเรียน

เมื่อผู้ใช้งานโทรศัพท์ที่ติดต่อเข้ามาใช้งานระบบได้แล้ว ระบบจะทำการร้องขอรหัสสนิติด ผู้ใช้จะต้องทำการป้อนรหัสสนิติดซึ่งเป็นตัวเลขจำนวน 8 หลัก เมื่อระบบได้รับรหัสสนิติดแล้วระบบจะทำการตรวจสอบความถูกต้องของรหัสสนิติดที่ได้รับกับข้อมูลที่อยู่ในฐานข้อมูล ถ้ารหัสสนิติดที่ผู้ใช้ป้อนเข้ามานั้นถูกต้องระบบจะทำงานในขั้นตอนต่อไป แต่ถ้ารหัสสนิติดที่ผู้ใช้ป้อนเข้ามานั้นไม่ถูกต้องระบบจะทำการร้องขอรหัสสนิติดใหม่ ซึ่งผู้ใช้จะสามารถป้อนรหัสสนิติดผิดได้สองครั้ง ถ้าครั้งที่สามยังผิดอีกระบบจะหยุดการติดต่อกับผู้ใช้ทันที

เมื่อผ่านการตรวจสอบรหัสสนิติดแล้ว ระบบจะทำการร้องขอรหัสผ่านซึ่งสามารถเป็นได้ทั้งตัวเลข เครื่องหมาย “#” และ เครื่องหมาย “*” จำนวน 8 หลัก เมื่อระบบได้รับรหัสผ่านแล้วระบบจะทำการตรวจสอบความถูกต้องของรหัสผ่าน โดยมีรูปแบบของการตรวจสอบเช่นเดียวกับการตรวจสอบความถูกต้องของรหัสสนิติด

เมื่อผ่านขั้นตอนของการตรวจสอบความถูกต้องของรหัสสนิติดและรหัสผ่านแล้ว ระบบจะร้องขอให้ผู้ใช้เลือกรูปแบบการสอบถามที่ผู้ใช้ต้องการ โดยมีรูปแบบการสอบถาม ดังนี้

1. การสอบถามผลการเรียนรายวิชา

ระบบจะร้องขอรหัสวิชาที่ต้องการสอบถาม ผู้ใช้ต้องป้อนรหัสวิชาที่เป็นตัวเลข 6 หลัก เมื่อระบบได้รับรหัสวิชาแล้ว ระบบจะทำการค้นหาข้อมูลจากฐานข้อมูล โดยระบบจะบอกรหัสวิชาและผลการเรียนของรายวิชานั้น

2. การสอบถามผลการเรียนเฉลี่ยเฉพาะภาคเรียน

ระบบจะทำการร้องขอให้ผู้ใช้ป้อนปีการศึกษาและภาคเรียนที่ต้องการ ข้อมูลปีการศึกษาเป็นตัวเลข 4 หลักและภาคเรียนเป็นตัวเลข 1 หลัก โดยป้อนข้อมูลติดกัน เช่น 25451 หมายถึง ปีการศึกษา 2545 ภาคเรียนที่ 1

3. การสอบถามผลการเรียนเฉลี่ยสะสม

ระบบจะทำการบอกผลการเรียนสะสมตั้งแต่ผู้ใช้เริ่มเข้าศึกษาจนถึงปัจจุบัน

เมื่อผู้ใช้เลือกสอบถามแล้ว ระบบจะดำเนินการตามแบบสอบถามที่ผู้ใช้ต้องการ และเมื่อสิ้นสุดของกระบวนการตามแบบสอบถามแล้ว ผู้ใช้สามารถที่จะเลือกการสอบถามแบบอื่นหรือเลือกการหยุดให้บริการแบบสอบถามก็ได้

3.1.2 การจัดการฐานข้อมูล

ส่วนนี้จะทำหน้าที่ในการค้นหาข้อมูลจากฐานข้อมูล โดยใช้ระบบฐานข้อมูล MySQL ซึ่งสามารถทำงานได้โดยไม่ขึ้นกับระบบปฏิบัติการ และสามารถทำงานกับตารางข้อมูลได้หลายตารางพร้อมๆ กัน

การติดต่อกับระบบฐานข้อมูล จะทำการติดต่อผ่านทาง Java Data Base Connection (JDBC) ทำให้โปรแกรมไม่ผูกติดกับระบบฐานข้อมูลที่ใช้ หากเปลี่ยนระบบฐานข้อมูลก็ไม่ต้องเขียน โปรแกรมใหม่ทั้งหมด เพียงเปลี่ยน drivers ให้ตรงกับระบบฐานข้อมูลที่ใช้เท่านั้น

3.1.2.1 การออกแบบและสร้างฐานข้อมูล

โครงสร้างตารางฐานข้อมูลที่ใช้ในระบบ เริ่มจากการสร้างฐานข้อมูลชื่อ gradechecking โดยใช้คำสั่ง

```
mysql> CREATE DATABASE gradechecking;
```



รูปที่ 3.4 E-R Diagram แสดงโครงสร้างตารางฐานข้อมูลที่ใช้ในระบบ

ระบบจะทำการค้นหาข้อมูลจากฐานข้อมูลนี้ ซึ่งออกแบบให้ประกอบด้วย 3 ตาราง ดังนี้

(1) ตาราง student

ใช้ในการเก็บข้อมูลของนิสิต และรหัสผ่านในการเข้าใช้ระบบของนิสิต มีโครงสร้างดังนี้

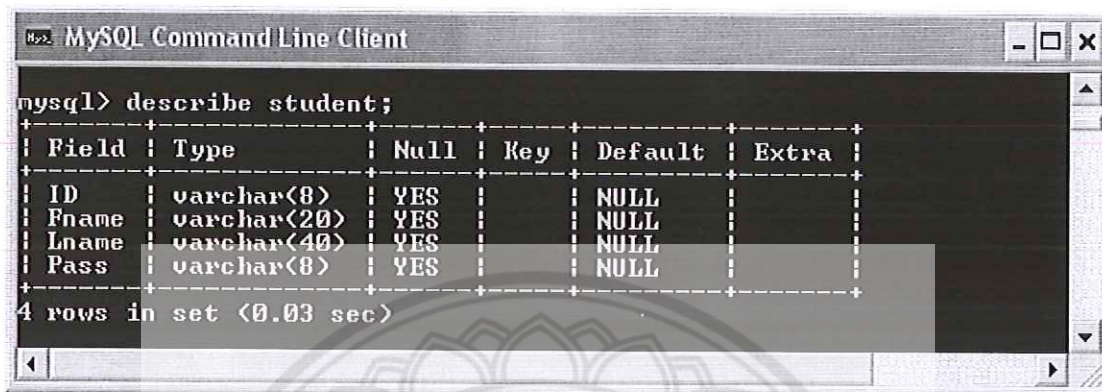
ตารางที่ 3.1 ตารางแสดงโครงสร้างตารางข้อมูล student ในฐานข้อมูล

ชื่อคอลัมน์	ชนิด	รายละเอียด
ID	varchar(8)	รหัสประจำตัวของนิสิต
Fname	varchar(20)	ชื่อของนิสิต
Lname	varchar(40)	นามสกุลของนิสิต
Pass	varchar(8)	รหัสผ่านในการเข้าใช้ระบบ

สร้างตาราง student โดยใช้คำสั่ง

```
mysql> CREATE TABLE student (ID VARCHAR(8), Fname VARCHAR(20),
-> Lname VARCHAR(20), Pass VARCHAR(8));
```

ซึ่งจะได้โครงสร้างของตาราง student ใน MySQL Command Line Client ดังรูปที่ 3.5



รูปที่ 3.5 แสดงโครงสร้างของตาราง student ใน MySQL Command Line Client

(2) ตาราง subject

ใช้ในการเก็บข้อมูลของรายวิชาต่างๆ มีโครงสร้างดังนี้

ตารางที่ 3.2 ตารางแสดงโครงสร้างตารางข้อมูล subject ในฐานข้อมูล

ชื่อคอลัมน์	ชนิด	รายละเอียด
ID	varchar(6)	รหัสรายวิชา
Subject	varchar(40)	ชื่อรายวิชา
Credit	int(1)	จำนวนหน่วยกิต

สร้างตาราง subject โดยใช้คำสั่ง

```
mysql> CREATE TABLE subject (ID VARCHAR(6), Subject VARCHAR(40),
-> Credit INT(1));
```

ซึ่งจะได้โครงสร้างของตาราง subject ใน MySQL Command Line Client ดังรูปที่ 3.6

```

mysql> describe subject;
+----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| ID | varchar(6) | YES | | NULL | |
| Subject | varchar(40) | YES | | NULL | |
| Credit | int(1) | YES | | NULL | |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.05 sec)

```

รูปที่ 3.6 แสดงโครงสร้างของตาราง subject ใน MySQL Command Line Client

(3) ตาราง grade

ใช้ในการเก็บผลการเรียนในแต่ละวิชาของนิสิต รวมทั้งปีการศึกษา และภาคเรียนที่ลงทะเบียนเรียน มีโครงสร้างดังนี้

ตารางที่ 3.3 ตารางแสดง โครงสร้างตารางข้อมูล grade ในฐานข้อมูล

ชื่อคอลัมน์	ชนิด	รายละเอียด
StdID	varchar(8)	รหัสนิสิต
SubID	varchar(6)	รหัสรายวิชา
Year	varchar(4)	ปีการศึกษาที่ลงทะเบียนเรียน
Term	varchar(1)	ภาคเรียนที่ลงทะเบียนเรียน
Grade	varchar(2)	ผลการเรียน

สร้างตาราง grade โดยใช้คำสั่ง

```

mysql> CREATE TABLE grade (StdID VARCHAR(8), SubID VARCHAR(6),
-> Year VARCHAR(4), Term VARCHAR(1), Grade VARCHAR(2));

```

ซึ่งจะได้โครงสร้างของตาราง grade ใน MySQL Command Line Client ดังรูปที่ 3.7

```
mysql> describe grade;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| StdID | varchar(8) | YES |     | NULL    |       |
| SubID | varchar(6) | YES |     | NULL    |       |
| Year  | varchar(4) | YES |     | NULL    |       |
| Term  | char(1)   | YES |     | NULL    |       |
| Grade | char(2)   | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

รูปที่ 3.7 แสดงโครงสร้างของตาราง grade ใน MySQL Command Line Client

3.1.2.2 การใช้ข้อมูลในฐานะข้อมูล

ในระบบมีการค้นหาข้อมูลในฐานะข้อมูลโดยใช้ในการตรวจสอบความถูกต้องและเพื่อใช้ในการหาคำตอบที่ผู้ใช้ต้องการ

(1) การตรวจสอบความถูกต้องของรหัสนิสิต

ใช้ข้อมูลในตาราง student ในการตรวจสอบความถูกต้องของรหัสนิสิต โดยมีรูปแบบประโยคในภาษา SQL ที่ใช้ในการตรวจสอบ ดังนี้

```
mysql> SELECT ID FROM student where ID = 'รหัสนิสิต';
```

(2) การตรวจสอบความถูกต้องของรหัสผ่าน

ใช้ข้อมูลในตาราง student ในการตรวจสอบความถูกต้องของรหัสผ่านของนิสิตแต่ละคน มีรูปแบบของประโยคในภาษา SQL ที่ใช้ในการตรวจสอบ ดังนี้

```
mysql> SELECT Pass FROM student where ID = 'รหัสนิสิต';
```

หลังจากนั้น นำข้อมูลที่ได้มาเปรียบเทียบกับ รหัสผ่านที่นิสิตทำการป้อนเข้ามาว่าตรงกันหรือไม่

(3) การหาเกรดของรายวิชา

ใช้ข้อมูลในตาราง grade โดยมีรูปแบบของประโยคในภาษา SQL ที่ใช้ในการค้นหาเกรดของรายวิชาที่นิสิตต้องการทราบ ดังนี้

```
mysql> SELECT Grade FROM grade where ID = 'รหัสนิสิต'
```

```
-> and SubID = 'รหัสรายวิชา';
```


(4) การหาเกรดเฉลี่ยในภาคเรียนที่ต้องการ

ในการหาเกรดเฉลี่ยในแต่ละภาคเรียนนั้น ใช้ข้อมูลใน 2 ตาราง คือ subject และ grade โดยรับข้อมูลที่นิสิตป้อนเข้ามา 3 ค่า คือ รหัสนิสิต, ปีการศึกษา และภาคเรียน มาใช้ในการหาข้อมูล มีรูปแบบของประโยคในภาษา SQL ดังนี้

```
mysql> SELECT subject.Credit , grade.Grade
-> FROM subject, grade where subject.ID = grade.SubID -> and StdID =
'รหัสนิสิต' and Year = 'ปีการศึกษา'
-> and Term = 'ภาคเรียน';
```

โดยจะได้ข้อมูลออกมา คือ จำนวนหน่วยกิต และ เกรดของแต่ละวิชาในเทอมนั้น ซึ่งสามารถนำข้อมูลเหล่านี้มาหาเกรดเฉลี่ยในภาคเรียนนั้นได้

(5) การหาเกรดเฉลี่ยสะสม

ใช้ข้อมูลใน 2 ตาราง คือ subject และ grade ในการหาเกรดเฉลี่ยสะสม โดยใช้รูปแบบของประโยคในภาษา SQL ดังนี้

```
mysql> SELECT subject.Credit , grade.Grade
-> FROM subject, grade where subject.ID = grade.SubID
-> and StdID = 'รหัสนิสิต';
```

ซึ่งจะได้ข้อมูลของจำนวนหน่วยกิตและ เกรดของทุกวิชาที่นิสิตเรียน แล้วนำข้อมูลเหล่านี้มาหาเกรดเฉลี่ยสะสม

3.1.3 การจัดการข้อมูลเสียง

ในการทำงานส่วนนี้ เป็นการจัดการข้อมูลที่ได้จากฐานข้อมูลให้อยู่ในรูปแบบของข้อมูลเสียงก่อนที่จะส่งกลับไปยังผู้ใช้เพื่อให้ผู้ใช้เข้าใจได้ ซึ่งข้อมูลเสียงที่ใช้เป็นไฟล์ .wav ในมาตรฐาน g.711 (CCITT u-Law 8 kHz, 8 Bit, Mono) โดยมีข้อมูลเสียงที่ใช้ในการสื่อสารกับผู้ใช้มี 2 รูปแบบ คือ

(1) ข้อมูลเสียงที่บอกให้ผู้ใช้งานทราบถึงขั้นตอนการทำงาน

เป็นส่วนที่ระบบบอกผู้ใช้งานให้ทราบถึงขั้นตอนการทำงาน เพื่อให้ผู้ใช้สามารถใช้งานระบบได้อย่างถูกต้อง

(2) ข้อมูลเสียงที่ใช้บอกผลการเรียน

ข้อมูลเสียงที่ประกอบด้วยเสียงของตัวเลข 0-9 , จุด , บวก และตัวอักษรภาษาอังกฤษ (A,B,C,D,F,I,W,S,U,P)

3.1.4 การจัดการเกี่ยวกับระบบโทรศัพท์

ระบบสอบถามผลการเรียน อัตโนมัตินทางโทรศัพท์นี้ใช้โมเด็ม ในการเชื่อมต่อคอมพิวเตอร์เข้ากับระบบโทรศัพท์ ทำให้เราต้องสามารถส่งงานโมเด็มได้ โดยเราใช้ JTAPI ในการจัดการการติดต่อกับระบบโทรศัพท์โดยมีสิ่งที่ต้องการส่งการให้โมเด็มทำงานดังนี้

- การตรวจจับสัญญาณการเรียกเข้า
- การรับสาย-วางสาย เพื่อเริ่มและสิ้นสุดการติดต่อ
- การส่งสัญญาณเสียงเพื่อแจ้งขั้นตอนการทำงานและคำตอบที่ผู้ใช้ต้องการกลับไปยังผู้ใช้งาน
- การตรวจจับสัญญาณการกดปุ่ม โทรศัพท์จากผู้ใช้งาน

3.2 ขอบเขตการใช้งานของระบบ

ผู้ใช้งานสามารถโทรศัพท์เข้ามาใช้งานได้ตลอดเวลาที่ระบบเปิดให้บริการอยู่ โดยจะสามารถทำการติดต่อได้ครั้งละหนึ่งคู่สาย

ซึ่งผู้ใช้งานสามารถติดต่อกับระบบด้วยวิธีการดังนี้

1. ฟังเสียงจากโทรศัพท์เพื่อทราบรายละเอียด ข้อมูล และขั้นตอนการทำงาน
2. กดปุ่มโทรศัพท์เพื่อเลือกตัวเลือกในการทำงาน และป้อนข้อมูลที่ระบบต้องการ เพียงเท่านั้นผู้

ใช้งานก็สามารถใช้งานระบบได้

โดยมีแบบการสอบถามให้ผู้ใช้สามารถใช้งานระบบดังนี้

1. สอบถามผลการเรียนรายวิชา
2. สอบถามผลการเรียนเฉลี่ยในภาคเรียนที่ต้องการ
3. สอบถามผลการเรียนเฉลี่ยสะสม

โดยระบบจะสามารถทำงานได้บนทุกระบบปฏิบัติการ เช่น Windows , Linux

3.3 สิ่งที่ระบบต้องการ

1. ข้อมูลเสียงที่ใช้ในการติดต่อกับผู้ใช้งานผ่านทางระบบ โทรศัพท์
2. ระบบการจัดการฐานข้อมูลที่สามารถใช้งานผ่านทางตัวจัดการ JDBC ได้
3. เลขหมายโทรศัพท์เพื่อเป็นการติดต่อระหว่างระบบและระบบโทรศัพท์
4. ระบบต้องการโมเด็ม สำหรับการแปลงสัญญาณเพื่อส่งข้อมูลผ่านไปในระบบโทรศัพท์

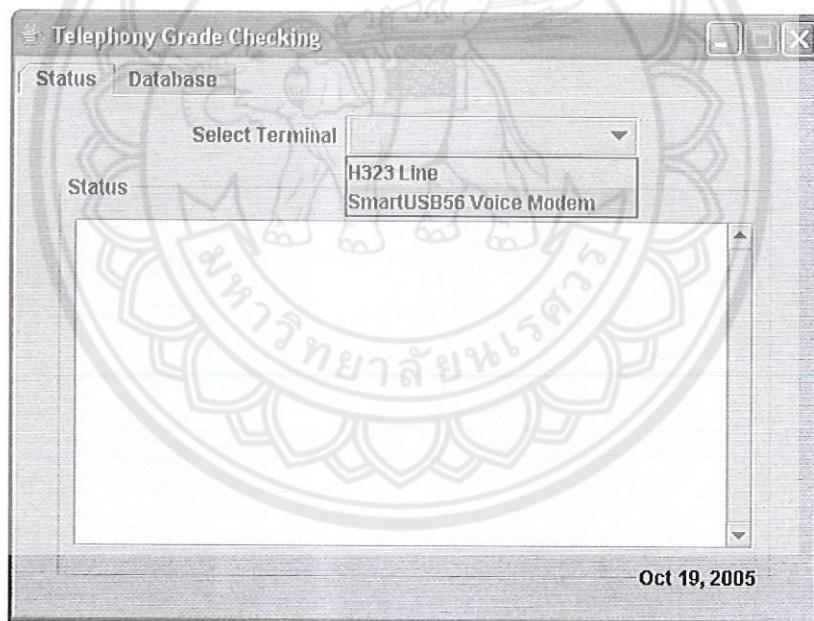
บทที่ 4

ผลการทดลอง

จากการพัฒนาระบบในบทที่แล้ว จะสามารถทำการทดลองสอบถามผลการเรียนอัตโนมัติทางโทรศัพท์ ซึ่งสามารถใช้ได้ทั้งโทรศัพท์บ้าน โทรศัพท์ภายใน และโทรศัพท์มือถือ ในการสอบถาม โดยทำการทดลองบนระบบปฏิบัติการ Microsoft Window XP เชื่อมต่อกับ SmartUSB56 Voice Modem 1 ตัว ผ่านทางเลขหมายโทรศัพท์บ้าน โดยได้ทำการทดลองในรูปแบบต่างๆ ดังนี้

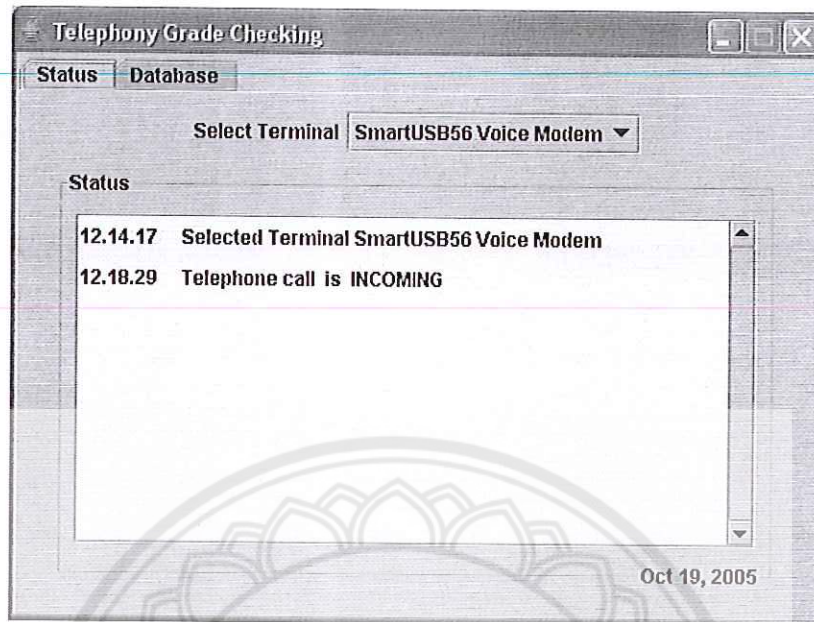
4.1 การทดลองระบบสอบถามผลการเรียน

ในการใช้งานระบบสอบถามผลการเรียน เมื่อระบบเริ่มทำงานจะสามารถเห็นการทำงานของระบบผ่านทางหน้าต่างอินเทอร์เน็ตเฟส โดยเริ่มแรกจะต้องทำการเลือกใช้โมเด็ม แสดงได้ดังรูปที่ 4.1



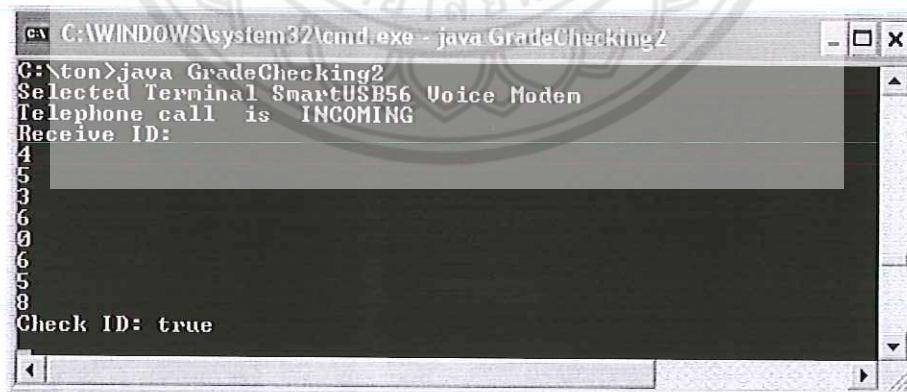
รูปที่ 4.1 แสดงหน้าต่างอินเทอร์เน็ตเฟสการเลือกใช้โมเด็ม

เมื่อมีผู้ใช้ติดต่อเข้ามาในระบบ หน้าต่างอินเทอร์เน็ตเฟสจะแสดงได้ดังรูปที่ 4.2



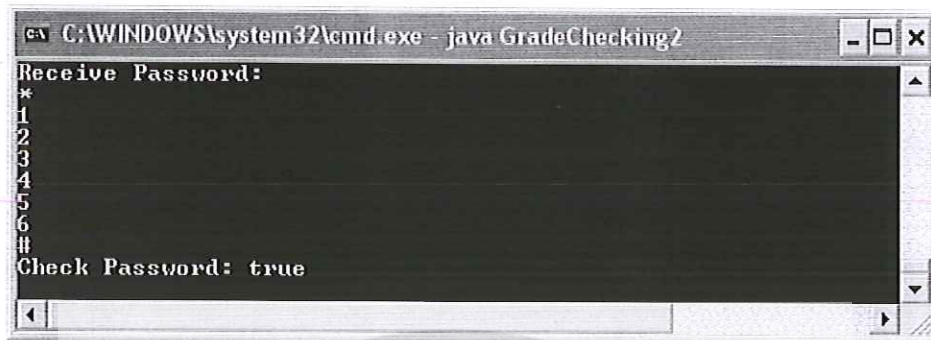
รูปที่ 4.2 แสดงหน้าต่างอินเทอร์เน็ตเฟสเมื่อมีผู้ใช้ติดต่อเข้ามาในระบบ

หลังจากที่ผู้ใช้ได้ติดต่อเข้ามาในระบบ ระบบจะร้องขอรหัสนิสิตและตรวจสอบรหัสนิสิตที่ผู้ใช้กดทางปุ่มโทรศัพท์ โดยระบบจะรับสัญญาณการกดมาทีละตัวและนำไปตรวจสอบกับฐานข้อมูลที่ได้นับทีละตัว ถ้าตรวจพบรหัสนิสิตในฐานข้อมูล ระบบจะแสดงผลว่ารหัสนิสิตถูกต้อง แสดงได้ดังรูปที่ 4.3



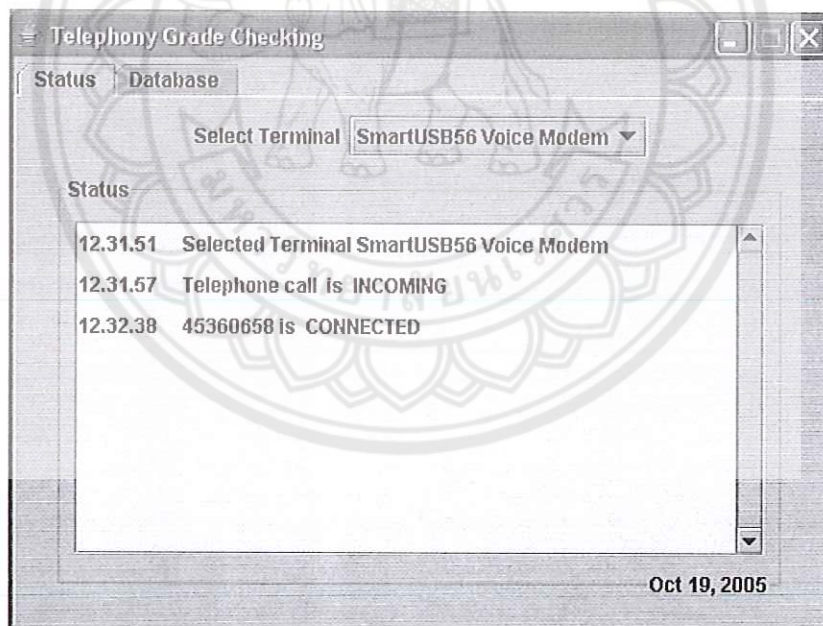
รูปที่ 4.3 แสดงการรับและตรวจสอบรหัสนิสิตของระบบ

เมื่อรหัสสนิติดถูกต้อง ระบบจะร้องขอรหัสผ่านและตรวจสอบรหัสผ่านที่ผู้ใช้กดทางปุ่ม โทรศัพท์ โดยระบบจะรับสัญญาณการกดมาทีละตัวและนำไปตรวจสอบกับฐานข้อมูลที่ได้นบันทึกไว้ ถ้าตรวจสอบพบรหัสผ่านในฐานข้อมูล ระบบจะแสดงผลว่ารหัสผ่านถูกต้อง แสดงได้ดังรูปที่ 4.4



รูปที่ 4.4 แสดงการรับและตรวจสอบรหัสผ่านของระบบ

เมื่อรหัสสนิติดและรหัสผ่านที่ระบบร้องขอถูกต้อง หน้าต่างอินเทอร์เน็ตเฟชจะแสดงผลการติดต่อกับระบบของผู้ใช้ โดยจะแสดงรหัสสนิติดของผู้ที่ติดต่อเข้ามาในระบบ ดังรูปที่ 4.5

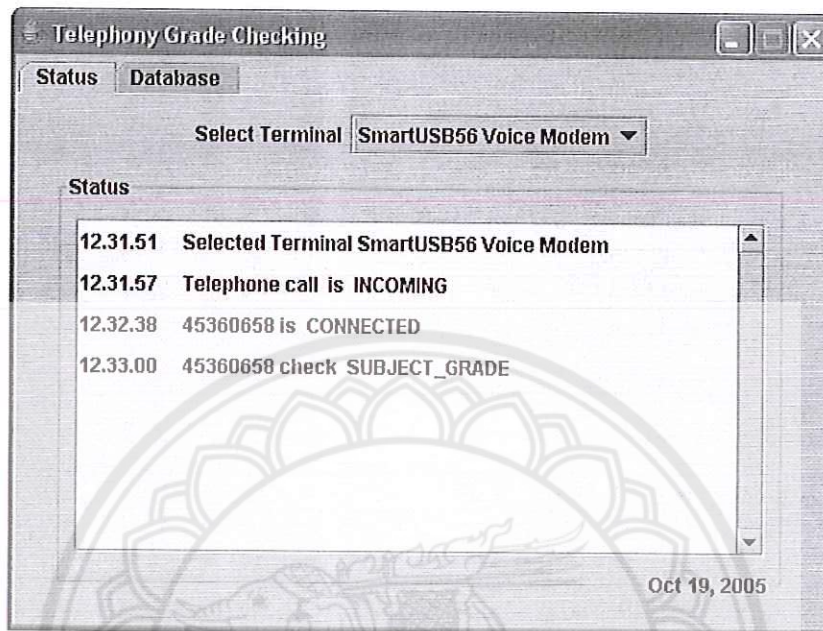


รูปที่ 4.5 แสดงหน้าต่างอินเทอร์เน็ตเฟชเมื่อรหัสสนิติดและรหัสผ่านถูกต้อง

เมื่อผ่านขั้นตอนของการตรวจสอบความถูกต้องของรหัสสนิติดและรหัสผ่านแล้ว ระบบจะร้องขอให้ผู้ใช้เลือกรูปแบบการสอบถามที่ผู้ใช้ต้องการ โดยมีรูปแบบการสอบถามให้ผู้ใช้เลือก 3 แบบ ดังนี้

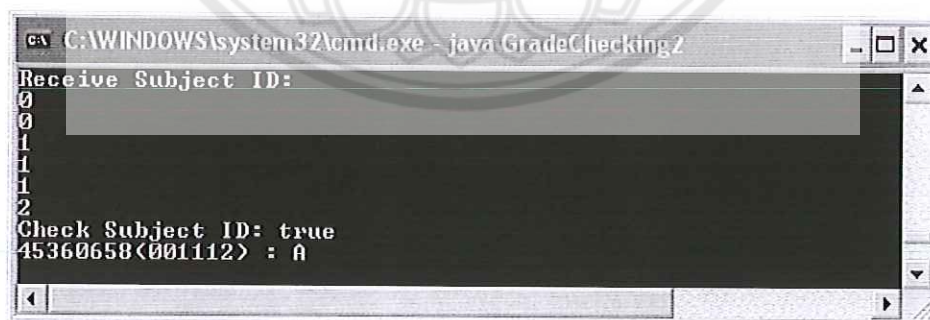
1. สอบถามผลการเรียนรายวิชา

เมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนรายวิชา ระบบจะแสดงหน้าต่างอินเทอร์เน็ตเฟชดัง
รูปที่ 4.6



รูปที่ 4.6 แสดงหน้าต่างอินเทอร์เน็ตเฟชเมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนรายวิชา

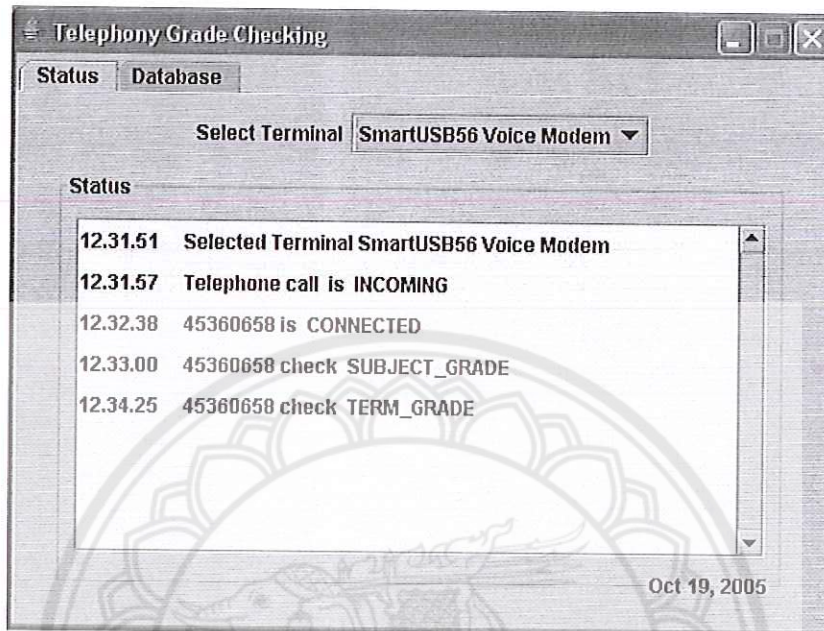
หลังจากผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนรายวิชา ระบบจะร้องขอรหัสรายวิชา และตรวจสอบรหัสรายวิชากับฐานข้อมูล ถ้ารหัสรายวิชาถูกต้อง ระบบจะแสดงรหัสรายวิชาที่ต้องการสอบถามและผลการเรียนของรายวิชานั้น ดังรูปที่ 4.7



รูปที่ 4.7 แสดงการรับและตรวจสอบรหัสรายวิชาของระบบ

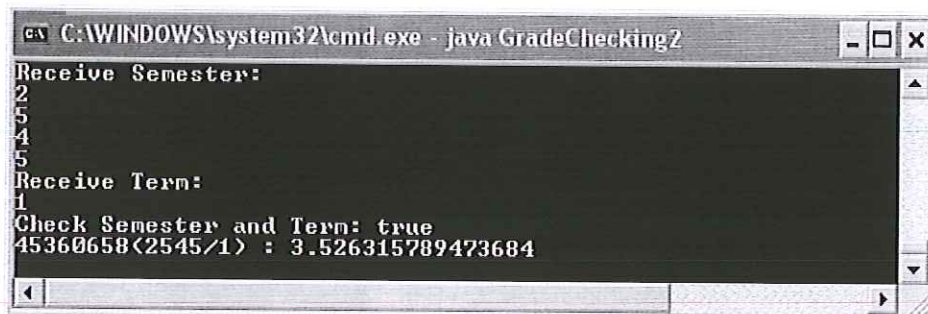
2. สอบถามผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ

เมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ ระบบจะแสดงหน้าต่างอินเทอร์เน็ต ดังรูปที่ 4.8



รูปที่ 4.8 แสดงหน้าต่างอินเทอร์เน็ตเมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ

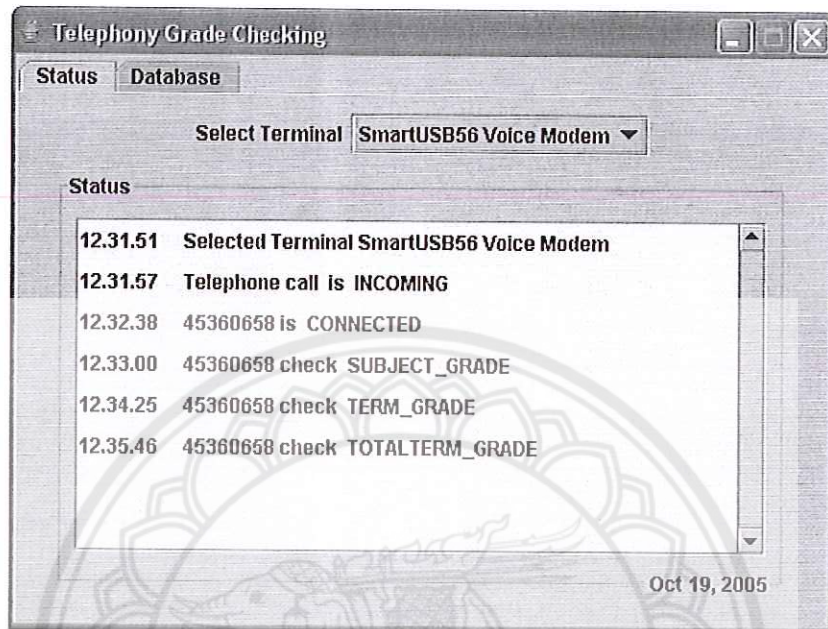
หลังจากผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ ระบบจะร้องขอปีการศึกษาและภาคเรียนที่ผู้ใช้บริการต้องการสอบถาม และทำการตรวจสอบปีการศึกษาและภาคเรียนกับฐานข้อมูล ถ้าพบปีการศึกษาและภาคเรียนที่ผู้ใช้สอบถาม ระบบจะแสดงผลการเรียนของภาคเรียนนั้น ดังรูปที่ 4.9



รูปที่ 4.9 แสดงการรับและตรวจสอบปีการศึกษาและภาคเรียนของระบบ

3. สอบถามผลการเรียนเฉลี่ยสะสม

เมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนเฉลี่ยสะสม ระบบจะแสดงหน้าต่างอินเทอร์เน็ตเฟซ ดังรูปที่ 4.10



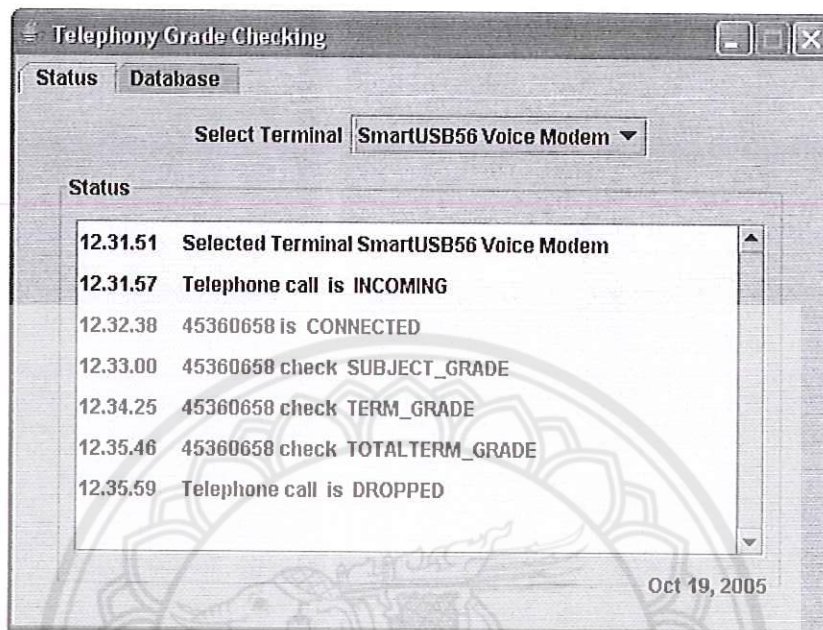
รูปที่ 4.10 แสดงหน้าต่างอินเทอร์เน็ตเฟซเมื่อผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนเฉลี่ยสะสม

หลังจากผู้ใช้เลือกรูปแบบการสอบถามผลการเรียนเฉลี่ยสะสม ระบบจะแสดงผลการเรียนเฉลี่ยสะสม ดังรูปที่ 4.11



รูปที่ 4.11 แสดงผลการเรียนเฉลี่ยสะสมของผู้ใช้

เมื่อสิ้นสุดของกระบวนการตามแบบสอบถามแล้ว ผู้ใช้สามารถที่จะเลือกการสอบถามแบบอื่น หรือเลือกการยกเลิกใช้บริการแบบสอบถามก็ได้ เมื่อผู้ใช้เลือกการยกเลิกใช้บริการหรือวางสายจะแสดง หน้าต่างอินเทอร์เน็ตเฟสดังรูปที่ 4.12



รูปที่ 4.12 แสดงหน้าต่างอินเทอร์เน็ตเฟสเมื่อผู้ใช้ยกเลิกการให้บริการหรือวางสาย

4.2 ผลการทดลองการทำการเปลี่ยนแปลงข้อมูลในฐานข้อมูลนิสิต

ในการทำงานของระบบ จะสามารถทำการเปลี่ยนแปลงข้อมูลนิสิตได้ ผ่านทางหน้าต่างอินเทอร์เน็ตเฟสการเปลี่ยนแปลงฐานข้อมูล โดยในส่วนนี้จะแบ่งการทำงานของฐานข้อมูลออกเป็น 3 ส่วน ได้แก่

1. ฐานข้อมูล Student

จะทำหน้าที่ในการเก็บข้อมูลนิสิต สามารถทำการเพิ่มข้อมูลนิสิต ค้นหา นิสิต แก้ไขข้อมูลนิสิต และลบข้อมูลนิสิตออกจากฐานข้อมูลได้ โดยในส่วนของการแสดงผลจะประกอบด้วยรหัสนิสิต รหัสผ่าน และชื่อ-นามสกุลของนิสิต ดังรูปที่ 4.13

รูปที่ 4.13 แสดงหน้าต่างอินเทอร์เน็ตเฟซของฐานข้อมูล Student

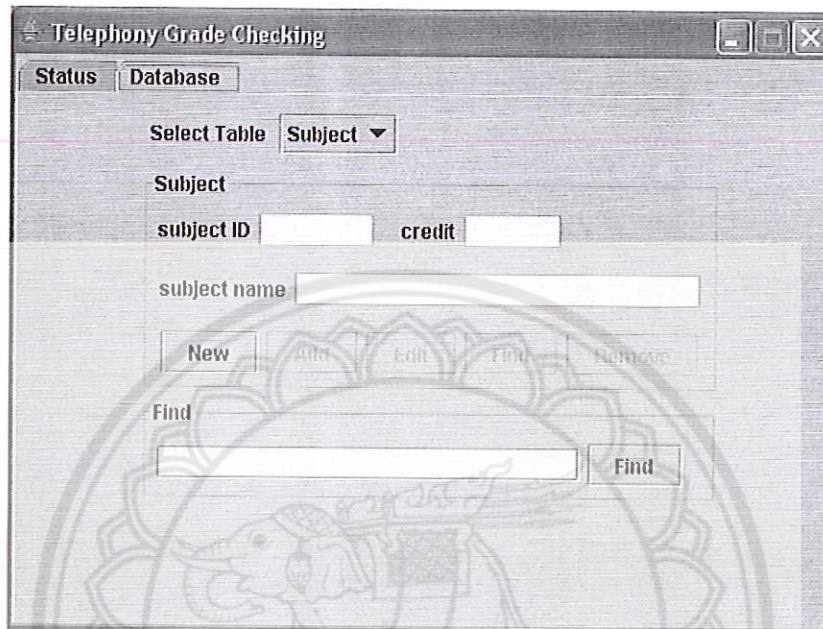
2. ฐานข้อมูล Grade

จะทำหน้าที่ในการเก็บข้อมูลผลการเรียนของนิสิต สามารถทำการเพิ่มข้อมูลผลการเรียน ค้นหาผลการเรียนรายวิชา แก้ไขและลบข้อมูลผลการเรียนของนิสิตออกจากฐานข้อมูลได้ โดยในส่วนของการแสดงผลจะประกอบด้วยรหัสนิสิต รหัสรายวิชา ปีการศึกษา ภาคเรียน และผลการเรียนที่ได้รับ ดังรูปที่ 4.14

รูปที่ 4.14 แสดงหน้าต่างอินเทอร์เน็ตเฟซของฐานข้อมูล Grade

3. ฐานข้อมูล Subject

จะทำหน้าที่ในการเก็บข้อมูลรายวิชา สามารถทำการเพิ่ม แก้ไขและลบข้อมูลได้ รวมทั้งสามารถค้นหาชื่อรายวิชาและจำนวนหน่วยกิตของแต่ละรายวิชาได้ โดยในส่วนของผลการแสดงผลจะประกอบด้วยรหัสรายวิชา ชื่อรายวิชา และจำนวนหน่วยกิตของรายวิชานั้น ดังรูปที่ 4.15



รูปที่ 4.15 แสดงหน้าต่างอินเทอร์เน็ตเฟสของฐานข้อมูล Subject

ในการทดลอง ภายหลังจากที่มีการเปลี่ยนแปลงข้อมูลในฐานข้อมูล เมื่อได้ทำการโทรศัพท์ติดต่อเข้ามาใช้งาน ระบบสามารถค้นหาข้อมูลจากฐานข้อมูลและบอกข้อมูลผลการเรียนได้ถูกต้องตามที่ได้ทำการเปลี่ยนแปลง

4.3 ผลการทดลองการใช้โทรศัพท์แบบต่างๆ ในการสอบถามผลการเรียน

ในการสอบถามผลการเรียน สามารถใช้โทรศัพท์แบบต่างๆ ในการสอบถาม เพื่อให้ได้ผลการเรียนที่ต้องการ โดยจากการทดลองสามารถใช้ทั้งระบบโทรศัพท์ที่ผ่านตู้สาขา (PABX) และระบบโทรศัพท์ปกติที่ไม่ผ่านตู้สาขา โดยทั้ง 2 ระบบโทรศัพท์โปรแกรมสามารถรับการสอบถามจากผู้ใช้ได้ทั้งทางโทรศัพท์ใช้สายตามบ้านหรือโทรศัพท์มือถือ

4.4 ผลการทดลองการเชื่อมต่อโมเด็มกับระบบโทรศัพท์แบบต่างๆ

ในการเชื่อมต่อโมเด็มกับระบบโทรศัพท์ เพื่อให้ระบบสอบถามผลการเรียนสามารถให้บริการสอบถามผลการเรียนผ่านทางโทรศัพท์ได้ ได้ทำการทดลองเชื่อมต่อโมเด็มกับระบบโทรศัพท์ 2 ระบบ คือ เชื่อมต่อโมเด็มกับเลขหมายโทรศัพท์ที่ผ่านตู้สาขา (PABX) และเชื่อมต่อโมเด็มกับเลขหมายโทรศัพท์ที่โทรศัพท์ปกติที่ไม่ผ่านตู้สาขา จากการทดลอง สามารถเชื่อมต่อ โมเด็มกับระบบโทรศัพท์ได้ทั้ง 2 รูปแบบ



บทที่ 5

สรุปผลและข้อเสนอแนะ

โครงการนี้แสดงให้เห็นถึงการพัฒนาระบบโดยใช้แอปพลิเคชันโปรแกรมมิ่งอินเทอร์เน็ตเฟชมาช่วยในการพัฒนาให้สามารถเขียนโปรแกรมเพื่อควบคุมและติดต่อกับอุปกรณ์ที่ได้ทำการเชื่อมต่อ โดยสามารถควบคุมการทำงานของโมเด็มในการรับสัญญาณที่ส่งมาทางโทรศัพท์และนำสัญญาณที่ได้มาประมวลผลและแสดงผลออกไปเป็นสัญญาณเสียงตอบกลับผ่านทางโทรศัพท์

ภาษาที่ใช้ในการพัฒนาระบบทั้งหมด คือ ภาษาจาวา พัฒนามาบน JTAPI (Java Telephony Application Programming Interface) เป็นอินเทอร์เน็ตเฟชในการเขียนโปรแกรมสำหรับ computer telephony ซึ่งถูกพัฒนาโดยบริษัท SUN (Sun Micro-System Co., LTD.) ทำให้สามารถเขียนโปรแกรมได้ง่ายและมีคุณลักษณะต่างๆครบถ้วน สามารถนำไปใช้ในแพลตฟอร์มและระบบโทรศัพท์ต่างๆกันได้

5.1 สรุปผลการทดลอง

จากการทดลองการเชื่อมต่อระบบ (Server) กับระบบโทรศัพท์และทดลองใช้โทรศัพท์โทรติดต่อบริษัท สามารถทำการเชื่อมต่อระบบได้ 2 รูปแบบ และการโทรติดต่อบริษัทสามารถโทรผ่านโทรศัพท์ได้ 3 แบบแสดงในตารางที่ 5.1 ดังนี้

ตารางที่ 5.1 แสดงผลการทดลองการเชื่อมต่อโมเด็มกับระบบโทรศัพท์และการโทรติดต่อบริษัท

การเชื่อมต่อโมเด็มกับระบบโทรศัพท์ที่ทำการทดลอง	การโทรติดต่อบริษัทที่ทำการทดลอง
1. Server ต่อกับระบบโทรศัพท์ที่มี PBX	1. โทรผ่านโทรศัพท์มือถือ 2. โทรผ่านโทรศัพท์โดยตรง 3. โทรผ่านโทรศัพท์ที่ผ่าน PBX
2. Server ต่อกับระบบโทรศัพท์ที่ไม่มี PBX	1. โทรผ่านโทรศัพท์มือถือ 2. โทรผ่านโทรศัพท์โดยตรง 3. โทรผ่านโทรศัพท์ที่ผ่าน PBX

ในขั้นตอนของการสอบถามผลการเรียน ได้ทำการทดลองการสอบถามผลการเรียนรายวิชาการ สอบถามผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ และการสอบถามผลการเรียนเฉลี่ยสะสม หลังจากที่ได้โทรติดต่อเข้าไปยังเลขหมายปลายทางของระบบแล้ว ระบบจะสามารถรับสัญญาณโทรศัพท์ที่ติดต่อเข้ามา โดยบนการทำงานของระบบจะแสดงผลผ่านทางอินเทอร์เน็ตเฟชของโปรแกรม ซึ่งจะแสดงการทำงานของระบบในแต่ละขั้นตอน และระบบจะส่งไฟล์เสียงตอบรับกลับไปยังโทรศัพท์ของผู้ใช้ โดยการติด

ต่อสื่อสารกับระบบ ผู้ใช้สามารถติดต่อกับระบบได้โดยการกดปุ่มบนเครื่องโทรศัพท์ เพื่อป้อนข้อมูลรหัสชนิดและรหัสผ่านที่ระบบร้องขอเข้าสู่ระบบ และเพื่อสอบถามผลการเรียนที่ต้องการ ผลการทดลองที่ได้ คือ ระบบสามารถทำงานได้ตามขั้นตอน สามารถส่งเสียงได้ถูกต้องตรงตามที่ได้ทำการบันทึกไว้ และสามารถดึงข้อมูลจากฐานข้อมูลนิติมาใช้ในระบบได้อย่างถูกต้อง โดยมีการแบ่งรูปแบบการสอบถามออกเป็น 3 รูปแบบ ได้แก่ ผลการเรียนรายวิชา ผลการเรียนเฉลี่ยภาคเรียนที่ต้องการ ผลการเรียนเฉลี่ยสะสม

ในการทดลองการเปลี่ยนแปลงฐานข้อมูลนิติ โดยการเพิ่มจำนวนนิติและการแก้ไขผลการเรียน จากนั้นทำการทดลองกับระบบอีกครั้ง เพื่อทดสอบว่าระบบสามารถแสดงผลได้ถูกต้องตรงตามข้อมูลที่ได้ทำการเปลี่ยนแปลงไว้หรือไม่ ผลการทดลองที่ได้ คือ ระบบสามารถแสดงผลโดยส่งเสียงไปยังโทรศัพท์ได้อย่างถูกต้องตรงตามที่ได้เปลี่ยนแปลงไว้ในฐานข้อมูลนิติ

5.2 ปัญหาในการทดลอง

ในช่วงที่ผู้ใช้ติดต่อเข้าใช้ระบบ ขณะที่ระบบมีการส่งสัญญาณเสียงบอกขั้นตอนการสอบถามต่างๆในระบบ ผู้ใช้จะไม่สามารถกดปุ่มโทรศัพท์ก่อนที่จะเสียงที่ส่งออกไปเล่นจบได้ ทำให้ผู้ใช้ต้องเสียเวลาในการรอที่จะเลือกรูปแบบการสอบถามหรือทำงานในขั้นตอนถัดไปของระบบ

5.3 ปัญหาในการพัฒนา

ในด้านของคุณภาพเสียงที่ใช้ในระบบนั้น ยังมีคุณภาพไม่ดีเท่าที่ควร เนื่องจากได้ทำการบันทึกเสียงโดยใช้อุปกรณ์ที่ไม่ได้มาตรฐาน ทำให้เสียงที่ทำการบันทึกนั้นยังไม่มีคุณภาพที่ดีมากนัก และในการส่งเสียงไปยังโทรศัพท์นั้น มีข้อจำกัดในด้าน Service Provider ทำให้เสียงที่ส่งกลับไปยังผู้ใช้บริการมีคุณภาพเสียงไม่ดีเท่ากับที่ได้ทำการบันทึกไว้

5.3 ข้อเสนอแนะ

1. เนื่องจากการทำงานของระบบ ได้กำหนดเป็นขั้นตอนที่ตายตัวไม่สามารถเปลี่ยนแปลงขั้นตอนการทำงานได้ ทำให้ยากในการนำไปประยุกต์ใช้กับระบบสอบถามอัตโนมัติรูปแบบอื่นๆ ดังนั้นควรจะทำให้โปรแกรมมีความยืดหยุ่น สามารถกำหนดการทำงานกับข้อมูลแบบต่างๆ ได้โดยไม่ต้องแก้ไขโปรแกรมมากนัก เพื่อให้สะดวกในการนำไปประยุกต์ใช้กับระบบสอบถามอัตโนมัติอื่นๆ
2. ในการเชื่อมต่อระบบกับสายโทรศัพท์ที่ได้พัฒนาขึ้นมา สามารถให้บริการได้เพียงครั้งละ 1 คู่สาย ควรจะมีการปรับปรุงให้สามารถให้บริการได้หลายคู่สายในเวลาเดียวกัน
3. ในการบันทึกเสียงควรบันทึกเสียงด้วยอุปกรณ์ที่มีคุณภาพ

เอกสารอ้างอิง

- [1] วีรศักดิ์ ชิ่งถาวร. **Java Programming Volume I**. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2547
- [2] วีรศักดิ์ ชิ่งถาวร. **Java Programming Volume II**. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2547
- [3] วีรศักดิ์ ชิ่งถาวร. **Java Programming Volume III**. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2547
- [4] วรณวิภา ดิถะสิริ. **คู่มือเรียน SQL ด้วยตัวเอง**. กรุงเทพมหานคร : โปรวิชั่น. 2545
- [5] สาริตพงศ์ พุทธิประเสริฐ, สินชัย กมลภิวังศ์. ศึกษกร วุฒิสัททิฏลกิจ. “โปร โคคคตลมาตรฐาน สำหรับอินเทอร์เน็ตเทลโฟนนี่ (Internet Telephony Protocols). ” [Online]. Available : http://www.nectec.or.th/ntj/No10/papers/No10_tutor_1.pdf
- [6] [Online]. Available : <http://xtapi.sourceforge.net/>
- [7] [Online]. Available : <http://java.sun.com/products/jtapi/>

