

การออกแบบและพัฒนาโปรแกรมการจัดเก็บข้อมูลเพลงด้วยวิธีการเชิงวัตถุ

Design and Development Music Information System

By Object Oriented Approach

โดย

นางสาวเปรมรัตน์ พุฒสวัสดิ์

รหัส 42067102



\*H001826\*

อาจารย์ที่ปรึกษา

ดร.นพพร โชติกกำธร

วัน เดือน ปี.....	15 ส.ค. 2550
เลขทะเบียน.....	01826
เลขเรียกหนังสือ.....	ฉ.พ. ๑/13ก. 2544
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2544

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

<b>ชื่อหัวข้อ</b>	การออกแบบและพัฒนาระบบสารสนเทศข้อมูลเพลง
<b>นักศึกษา</b>	นางสาวเปรมรัตน์ พูลสวัสดิ์
<b>อาจารย์ที่ปรึกษา</b>	ดร.นพพร โชติภักดิ์
<b>ระดับการศึกษา</b>	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
<b>แขนงวิชา</b>	วิทยาการสารสนเทศ
<b>ปีการศึกษา</b>	2544

### บทคัดย่อ

โครงการพัฒนาระบบงานนี้ มีจุดประสงค์ในการออกแบบและพัฒนาระบบเพื่อนำไปใช้ประโยชน์สำหรับงานด้านการจัดรายการวิทยุอาทิเช่น งานด้านการจัดเก็บและค้นหาข้อมูลเพลง ข้อมูลศิลปินนักร้อง การจัดอันดับเพลง การเก็บข้อมูลการเล่นเกม ฯลฯ เพื่อให้มีประสิทธิภาพและความน่าเชื่อถือของข้อมูล โดยอยู่บนพื้นฐานของความสะดวกสบายต่อการใช้งานของผู้ใช้โดยแนวทางในการศึกษานั้นจะเป็นการวิเคราะห์และออกแบบระบบงานของบริษัทผู้ผลิตรายการวิทยุ โดยนำเอาหลักการเชิงวัตถุเข้ามาใช้ในการวิเคราะห์และออกแบบ และเลือกUML (Unified Modeling Language) ซึ่งเป็นโมเดลที่สนับสนุนแนวคิดเชิงวัตถุ มาใช้ในการวิเคราะห์และออกแบบระบบโดยอาศัย Rational Rose เป็นเครื่องมือในการสร้างไดอะแกรมต่าง ๆ เช่น Use Case Diagram, Sequence Diagram, Class Diagram และ State Diagram สำหรับในส่วนของการเขียนโปรแกรมเป็นลักษณะของการเขียนโปรแกรมแบบโครงสร้าง (Structure Programming) และฐานข้อมูลแบบ Relational Database โดยใช้ Visual Basic 6.0 ร่วมกับฐานข้อมูล MS Access 97

**Title** Design and Development Music Information System By Object Oriented Approach

**Student** Miss. Premrat Poolsawat

**Advisor** Dr.Nopporn Chotikakamthorn

**Level of Study** Master of Science in Information Technology

**Major** Information Science

**Academic Year**2001

### ABSTRACT

Music Information System Project designed and developed for Radio Station's Task like Store and Search Song Data, Artist Data, Chart, GamePlayer . for Improve performance , reliable of work and user friendly. By using Object Oriented Analysis and Design , all of system model are constructed in UML (Unified Modeling Language) format that support object oriented concept and use Rational Rose for represent diagrams. In the part of Implement base on Structure Programming and Relational Database by using Visual Basic 6.0 and MS Access 97.

## กิตติกรรมประกาศ

ในการจัดทำโครงการการออกแบบและพัฒนา โปรแกรมจัดเก็บข้อมูลเพลงด้วยวิธีการเชิง  
วัตถุนี้ ได้รับความสนับสนุนจากบุคคลหลาย ๆ ท่าน ทั้งทางด้านครอบครัวของผู้พัฒนาเองที่คอยดู  
แลเอาใจใส่ และให้กำลังใจ ตลอดการทำงาน ท่านอาจารย์ที่ปรึกษา คร.นพพร โชติกกำธร ที่ช่วย  
เสนอความคิดและให้คำแนะนำสำหรับการทำงาน และพี่ ๆ ที่สถานีวิทยุ ที่ให้ข้อมูลและข้อเสนอ  
แนะเพื่อเป็นแนวทางในการพัฒนาระบบ รวมทั้งเพื่อน ๆ พี่ ๆ IS8 ปกติ ที่ร่วมเรียนมาด้วยกัน แลก  
เปลี่ยนความคิดเห็นในการทำงานด้วยกัน ทางผู้พัฒนาเองก็ขอขอบพระคุณทุก ๆ ท่านไว้ ณ ที่นี้

เปรมรัตน์ พูลสวัสดิ์

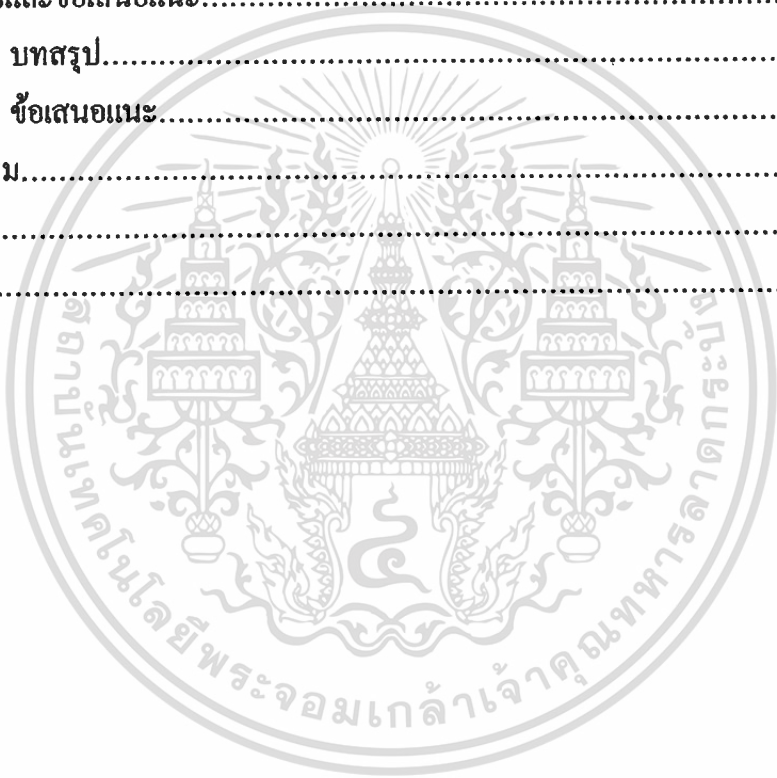


## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	VI
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาของระบบ.....	1
1.2 วัตถุประสงค์ของโครงการพัฒนาระบบงาน.....	1
1.3 ขอบเขตของโครงการพัฒนาระบบงาน.....	2
1.4 เนื้อหาในโครงการพัฒนาระบบงาน.....	2
2. Object-Oriented Methodology และ Unified Modeling Language.....	3
2.1 Object-Oriented คืออะไร.....	3
2.2 คำศัพท์และความหมาย.....	3
2.3 คุณสมบัติที่สำคัญของ Object-Oriented.....	6
2.4 Object-Oriented Methodology.....	7
2.5 ส่วนประกอบที่สำคัญของ Methodology.....	7
2.6 UML คืออะไร.....	8
2.7 ประวัติศาสตร์ของ UML.....	8
2.8 ประเภทของ UML Diagram.....	9
2.9 ความหมายของแต่ละ Diagram.....	11
3. การศึกษาการดำเนินการและปัญหาของระบบงานปัจจุบัน.....	17
3.1 การดำเนินงานของระบบงานปัจจุบัน.....	17
3.2 ปัญหาของระบบงานปัจจุบัน.....	19
4. การวิเคราะห์และออกแบบระบบใหม่.....	21
4.1 การวิเคราะห์ระบบ.....	21
4.2 การออกแบบระบบ.....	35

## สารบัญ (ต่อ)

	หน้า
5. การพัฒนาระบบและผลการทำงาน.....	48
5.1 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ.....	48
5.2 การพัฒนาระบบ.....	48
5.3 ผลการทำงาน.....	49
6. บทสรุปและข้อเสนอแนะ.....	50
6.1 บทสรุป.....	50
6.2 ข้อเสนอแนะ.....	51
บรรณานุกรม.....	52
ภาคผนวก.....	53
ประวัติผู้เขียน.....	70



## สารบัญภาพ

	หน้า
รูปที่ 2-1 ผังแสดงคำนิยามของ Object.....	3
รูปที่ 2-2 ตัวอย่าง Object ของคนที่ชื่อ "สมชาย".....	4
รูปที่ 2-3 ตัวอย่างส่วนประกอบภายใน Object ของคนที่ชื่อสมชาย.....	4
รูปที่ 2-4 ตัวอย่าง Static relation ของ Object.....	5
รูปที่ 2-5 ตัวอย่าง Dynamic relation ของ Object.....	5
รูปที่ 2-6 โครงสร้างส่วนประกอบของ System Development Methodology.....	7
รูปที่ 2-7 แผนภาพแสดงประวัติศาสตร์ของ UML.....	9
รูปที่ 2-8 แผนภาพแสดง UML Diagram ทั้งหมด 9 Diagrams.....	10
รูปที่ 2-9 สัญลักษณ์ของ Actor.....	11
รูปที่ 2-10 สัญลักษณ์ของ Use Case.....	12
รูปที่ 2-11 สัญลักษณ์ของ Communication.....	12
รูปที่ 2-12 สัญลักษณ์ของ Relationship.....	12
รูปที่ 2-13 สัญลักษณ์ของ Class.....	13
รูปที่ 2-14 สัญลักษณ์ของ Association relationship.....	13
รูปที่ 2-15 สัญลักษณ์ของ Aggregation relationship.....	14
รูปที่ 2-16 สัญลักษณ์ของ Depend on relationship.....	14
รูปที่ 2-17 ตัวอย่างความสัมพันธ์แบบ Generalization.....	14
รูปที่ 2-18 สัญลักษณ์ที่ใช้ภายใน Component Diagram.....	15
รูปที่ 2-19 สัญลักษณ์ต่าง ๆ ที่ใช้ใน Sequence Diagram.....	15
รูปที่ 2-20 สัญลักษณ์ต่าง ๆ ที่ใช้ใน State Diagram.....	16
รูปที่ 2-21 State Diagram แสดงการปิดเครื่อง.....	16
รูปที่ 4-1 Use Case Diagram ของระบบสารสนเทศข้อมูลเพลง.....	22
รูปที่ 4-2 Sequence Diagram ของ Login.....	27
รูปที่ 4-3 Sequence Diagram ของ Register.....	28
รูปที่ 4-4 Sequence Diagram ของ Play Song.....	28
รูปที่ 4-5 Sequence Diagram ของ Request Song.....	29
รูปที่ 4-6 Sequence Diagram ของ Get Chart.....	29

## สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4-7 Sequence Diagram ของ Search Song.....	30
รูปที่ 4-8 Sequence Diagram ของ Song Data.....	30
รูปที่ 4-9 Sequence Diagram ของ Artist Data.....	31
รูปที่ 4-10 Sequence Diagram ของ GamePlayer Data.....	31
รูปที่ 4-11 Class Diagram พื้นฐานของระบบสารสนเทศข้อมูลเพลง.....	32
รูปที่ 4-12 State Diagram ของการเข้าใช้งาน โปรแกรม .....	33
รูปที่ 4-13 State Diagram ของการเล่นเพลง.....	33
รูปที่ 4-14 State Diagram ของการเพิ่มข้อมูลลงในฐานข้อมูล.....	34
รูปที่ 4-15 State Diagram ของการจัดเตรียมเพลงเพื่อการจัดเก็บ.....	34
รูปที่ 4-16 Sequence Diagram ของ Login (ส่วนของการ Design).....	36
รูปที่ 4-17 Sequence Diagram ของ Register (ส่วนของการ Design).....	36
รูปที่ 4-18 Sequence Diagram ของ Play Song (ส่วนของการ Design).....	37
รูปที่ 4-19 Sequence Diagram ของ Search Song (ส่วนของการ Design).....	37
รูปที่ 4-20 Sequence Diagram ของ Request (ส่วนของการ Design).....	38
รูปที่ 4-21 Sequence Diagram ของ GetChart (ส่วนของการ Design).....	38
รูปที่ 4-22 Sequence Diagram ของ Song Data (ส่วนของการ Design).....	39
รูปที่ 4-23 Sequence Diagram ของ Artist Data (ส่วนของการ Design).....	39
รูปที่ 4-24 Sequence Diagram ของ GamePlayer Data (ส่วนของการ Design).....	40
รูปที่ 4-25 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FrmDJSchedule.....	40
และ Class FrmStation กับ Class อื่น ๆ	
รูปที่ 4-26 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FrmArtist.....	41
และ Class FrmSong กับ Class อื่น ๆ	
รูปที่ 4-27 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FrmRequest.....	41
และ Class FrmPlaySong กับ Class อื่น ๆ	
รูปที่ 4-28 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FrmGamePlayer.....	42
กับ Class อื่น ๆ	
รูปที่ 4-29 Class Diagram แสดงความสัมพันธ์ระหว่าง Class ต่าง ๆ ทั้งหมดในระบบ.....	42

## สารบัญภาพ (ต่อ)

หน้า

รูปที่ 4-30 ตัวอย่างการออกแบบหน้าจอของระบบที่สัมพันธ์กับ Class.....	43
รูปที่ 4-31 Class DJSchedule ที่สร้างขึ้นจากวิธีการ Reverse Engineering.....	44
และความสัมพันธ์ของ Class DJSchedule กับ FrmSchedule	
รูปที่ 4-32 ก) การ Generate Code จาก Class ที่สร้างขึ้นมา.....	45
รูปที่ 4-32 ข) การ Browse DDL เพื่อดู Schema ที่สร้างขึ้นมาจาก Class ที่เลือก.....	45
รูปที่ 4-32 ค) หน้าจอแสดงผลลัพธ์ของการสร้าง Schema จาก Class ที่เลือก.....	46
รูปที่ 4-33 โครงสร้างของฐานข้อมูลของ โปรแกรมจัดเก็บข้อมูลเพลง.....	47
รูปที่ 5-1 แบบจำลองการทำงานของ ADO Data Control.....	49
รูปที่ 6-1 แนวคิดเปรียบเทียบวิธีการวิเคราะห์และออกแบบระบบเชิงวัตถุกับแบบ โครงสร้าง.....	51



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาของระบบ

ในปัจจุบันเป็นที่ยอมรับกันว่าวิธีการเชิงวัตถุ (Object Oriented Approach) เป็นวิธีการที่ช่วยให้การออกแบบและพัฒนาระบบเป็นไปได้อย่างสะดวก ด้วยคุณสมบัติที่มีอยู่ในตัวของมันไม่ว่าจะเป็นทางด้านของการนำกลับมาใช้ใหม่ (Reusability) การถ่ายทอดคุณสมบัติ (inheritance) ฯลฯ จุดเด่นของวิธีการเชิงวัตถุนี้คือการที่มันเป็นวิธีการแบบ Bottom-up ที่จะมองจากส่วนเล็กๆ มองใน Module เล็กๆ ที่จะนำมาประกอบกันเป็นระบบใหญ่ ๆ ระบบหนึ่ง ซึ่งแต่ละ Module มันสามารถที่จะถูกนำไปใช้ในการสร้างระบบอื่นที่ต้องการการทำงานลักษณะเดียวกันได้ ทำให้มันง่ายต่อการที่เราจะสร้างระบบอื่น ๆ โดยอาศัย Module ที่เคยสร้างมาแล้ว จึงช่วยให้การพัฒนาระบบด้วยวิธีการเชิงวัตถุนี้รวดเร็วมากขึ้นกว่าวิธีการแบบโครงสร้าง (Structure Approach)

ในโครงการนี้ได้ทำการศึกษากระบวนการออกแบบและพัฒนาระบบที่เป็น Object - Oriented โดยนำมาประยุกต์ใช้กับระบบสารสนเทศข้อมูลเพลง ซึ่งการจัดรายการเพลงของสถานีวิทยุในปัจจุบันนั้น ส่วนใหญ่จะมีการดำเนินงานค้นหาและเล่นเพลงโดยอาศัยความชำนาญงานของนักจัดรายการเอง ซึ่งในแต่ละสถานีก็มีวิธีการหรือระบบการดำเนินงานที่ต่างกันไป การที่นักจัดรายการวิทยุที่เริ่มงานใหม่ซึ่งอาจเป็นผู้ที่เพิ่งเริ่มมาทำงานนี้ หรือเป็นผู้ที่เคยจัดรายการจากสถานีอื่นจะเรียนรู้ระบบการทำงานของสถานีนั้น ๆ จนคล่องตัว อาจจะเป็นการเสียเวลามาก และนอกจากนี้มีการใช้เทคโนโลยีสารสนเทศอย่างไม่เต็มรูปแบบ เช่น ใช้พิมพ์รายงานหรือเก็บเอกสารบางอย่าง ดังนั้น จึงเกิดแนวคิดในการที่จะทำการพัฒนาระบบสารสนเทศข้อมูลเพลง เพื่อที่จะพัฒนาการทำงานทางด้านการจัดรายการเพลงของสถานีวิทยุให้มีความคล่องตัวสูงขึ้นทั้งทางด้านการจัดเก็บและค้นหาข้อมูลต่าง ๆ ที่เกี่ยวข้อง เพื่อการทำงานที่มีประสิทธิภาพมากขึ้น และเป็นการนำเอาเทคโนโลยีที่มีอยู่มาใช้อย่างคุ้มค่า

### 1.2 วัตถุประสงค์ของโครงการพัฒนาระบบงาน

- 1) เพื่อศึกษาหลักการของแบบจำลองเชิงวัตถุ (Object Oriented Model)
- 2) เพื่อศึกษากระบวนการออกแบบและการพัฒนาระบบ โดยใช้เครื่องมือ (CASE TOOL) และภาษาที่สนับสนุนแบบจำลองเชิงวัตถุ
- 3) เพื่อการประยุกต์ใช้หลักการเชิงวัตถุในการออกแบบระบบสารสนเทศข้อมูลเพลง

- 4) ทำการพัฒนาระบบงานเพื่อช่วยเพิ่มประสิทธิภาพในการทำงานเกี่ยวกับข้อมูลเพลง

### 1.3 ขอบเขตของโครงการพัฒนาระบบงาน

- 1) วิเคราะห์และออกแบบระบบงาน โดยใช้แนวคิดเชิงวัตถุ (สร้าง UML Diagram ต่าง ๆ โดยการใช้โปรแกรม Rational Rose 2000 Enterprise Edition)
- 2) ออกแบบและสร้างฐานข้อมูลเชิงสัมพันธ์โดยใช้ MS Access 97
- 3) พัฒนาระบบงานตามที่ได้ทำการออกแบบไปสู่ Application โดยใช้ Visual Basic Programming

### 1.4 เนื้อหาในโครงการพัฒนาระบบงาน

เนื้อหาสำหรับโครงการพัฒนาระบบงานนี้ในบทแรกจะเป็นการกล่าวถึงความเป็นมา ความสำคัญและวัตถุประสงค์ในการทำโครงการพัฒนาระบบงาน ในบทที่ 2 จะกล่าวถึงแนวคิดเชิงวัตถุ ส่วนประกอบที่สำคัญของ Object-Oriented methodology และแบบจำลองที่ใช้ได้แก่ UML (Unified Modeling Language) บทที่ 3 เป็นส่วนของการศึกษาการดำเนินงานและปัญหาของระบบงานปัจจุบัน บทที่ 4 เป็นการวิเคราะห์และออกแบบระบบงานใหม่ บทที่ 5 เป็นส่วนของการพัฒนาระบบงานและผลการทำงานของระบบงานใหม่ บทที่ 6 เป็นส่วนของการสรุปและข้อเสนอแนะ และในภาคผนวกเป็นส่วนการติดตั้งและวิธีใช้งาน โปรแกรมจัดเก็บข้อมูลเพลง

## บทที่ 2

### Object -Oriented Methodology และ Unified Modeling Language

สำหรับเนื้อหาในบทนี้จะพูดถึงในส่วนของแนวคิดเชิงวัตถุและโมเดลจะนำมาที่ใช้ในการวิเคราะห์และออกแบบระบบคือ Unified Modeling Language (UML) ซึ่งมีรายละเอียดดังต่อไปนี้

#### 2.1 Object-Oriented คืออะไร

"Object-Oriented" เป็นแนวคิด หรือระเบียบวิธีคิด (paradigm) ของการสร้างหรือพัฒนาระบบงานหนึ่ง ๆ ดังคำพูดที่ว่า "OO is a technique for systems modeling" โดยจะมองระบบเป็นกลุ่มของวัตถุ (Object) ที่มีปฏิสัมพันธ์กัน หรือ "a number of Object that interaction" โดยการรวมเอาข้อมูล (information) และฟังก์ชันการทำงาน (Operation) เข้าไว้ด้วยกันในวัตถุ และกำหนด "วิธีการติดต่อกันระหว่างวัตถุ" (Interface) โดยถ้าวัตถุหนึ่งต้องการจะติดต่อกับอีกวัตถุหนึ่งจะต้องติดต่อผ่าน Interface ที่กำหนดไว้เท่านั้น

จากหลักการนี้จะพบว่ามีการเปลี่ยนแปลงแก้ไข หน้าที่การทำงาน และข้อมูลใด ๆ ในวัตถุหนึ่ง จะมีผลกระทบต่อวัตถุอื่น ๆ น้อยมาก ซึ่งจะทำให้การแก้ไขโปรแกรมทำได้สะดวกและรวดเร็วยิ่งขึ้น เป็นการลดต้นทุนในการบำรุงรักษาโปรแกรม นอกจากนี้ยังมีความสามารถในการสืบทอดคุณสมบัติ (Inheritance) จากวัตถุหนึ่งไปอีกวัตถุหนึ่งได้ ทำให้สามารถนำซอฟต์แวร์บางส่วนที่มีอยู่เดิมกลับมาใช้ใหม่ได้ (Reusability) โดยผู้พัฒนาไม่จำเป็นต้องทราบว่าซอฟต์แวร์บางส่วนที่นำกลับมาใช้นั้นเขียนขึ้นมาอย่างไร (How does it work?) เพียงแต่ทราบว่าซอฟต์แวร์ส่วนนั้นทำอะไร (What does it work ?) และทำการเพิ่มคุณสมบัติอื่น ๆ ที่ผู้พัฒนา ต้องการเข้าไป ทำให้การพัฒนาซอฟต์แวร์ใหม่ ๆ ทำได้รวดเร็วขึ้น[8]

#### 2.2 คำศัพท์ และความหมาย

- **Object** คือ วัตถุหรือสิ่งที่ประกอบด้วยสถานะ (State) หรือข้อมูล (Information) และฟังก์ชันการทำงาน(Operation) หรือพฤติกรรม (Behavior)

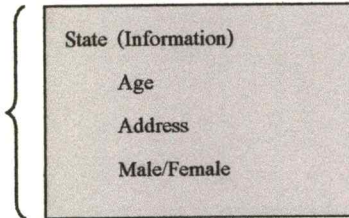
An Object is a characterized by  
A number of operations  
And a state which remembers the effect of these operations

รูปที่ 2-1 แสดงคำนิยามของ Object

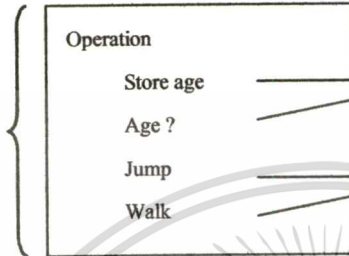
**ตัวอย่าง**

**สมชาย (Object ของคน)**

This part is **hidden** to us



The only part of the Object that we can see  
Is its **operation**.



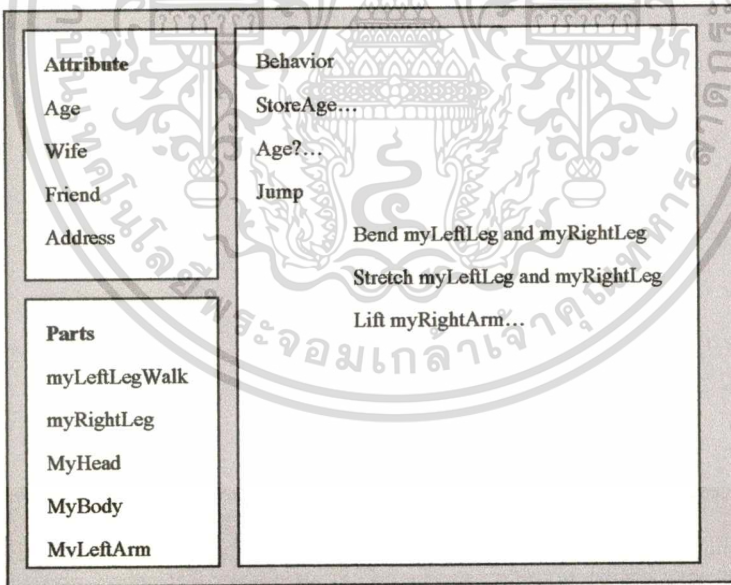
Operation which can affect or read the save info.

Operation which perform behaviors (affect internal info.)

**รูปที่ 2-2 ตัวอย่าง Object ของคนอื่นที่ชื่อ "สมชาย"**

ดังนั้นเมื่อเรามองเข้าไปใน Object จะประกอบด้วย Attribute, Part และ Behavior ดังรูป

สมชาย

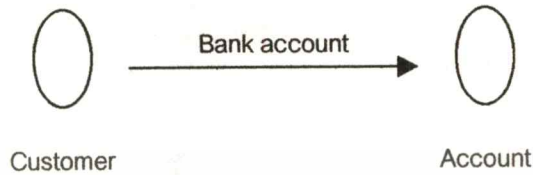


**รูปที่ 2-3 ตัวอย่างส่วนประกอบภายใน Object ของคนที่ชื่อสมชาย**

- **Message Passing**

ความเกี่ยวข้องกันระหว่าง Object แบ่งเป็น 2 ลักษณะคือ

1. **Static Relation** เช่น ลูกค้าเป็นเจ้าของบัญชี



รูปที่ 2-4 ตัวอย่าง Static relation ของ Object

2. **Dynamic Relation** เช่น ผู้ฝากเงินฝากเงินเข้าไปในบัญชี



รูปที่ 2-5 ตัวอย่าง Dynamic relation ของ Object

Dynamic Relation โดยความหมายของ OO จะหมายถึง Object หนึ่งจะกระตุ้น (Stimuli) ไปยังอีก Object หนึ่งส่วนในทาง Programming จะเรียกเป็นการส่ง Message ดังตัวอย่าง หากต้องการให้สมชาย jump ก็ต้องส่ง Stimuli ไปเมื่อสมชายได้รับก็จะกระทำ action ต่าง ๆ ได้แก่ส่ง stimuli ไปยัง arm และ leg

- **Class และ Instance**

Class คือ เซตของ Object

Instanceคือ Object ที่สร้างจาก Class

ตัวอย่าง Class ของคน ประกอบด้วย คนแต่ละคน โดยมีสมชายเป็น Instance ของ Class คน หรือ คนชื่อ "ประคิษฐ์" เป็น Instance ของ Class คนเช่นกัน โดยแต่ละ Instance จะมีความแตกต่างกันระหว่าง Object อื่น ๆ (มีเอกลักษณ์ หรือ identity)

## 2.3 คุณสมบัติที่สำคัญของ Object-Oriented

### ● Encapsulation และ Information hiding

ข้อมูล (Information) และหน้าที่การทำงานหรือพฤติกรรม (Operation/Behavior) จะถูกจับมัดรวมเข้าด้วยกัน (Encapsulation) ซึ่งจะสนับสนุนให้เกิด "การซ่อนข้อมูล" (Information hiding) และสิ่งที่เราจะเห็นได้ของ Object คือ Instance ของมัน

ทำให้การเข้าถึงข้อมูลจะกระทำโดยเรียกใช้ Operation ของ Object ขึ้นมาทำงาน ดังคำกล่าวที่ว่า "The only way to reaching the information in an Object is by using its operation."

"Abstract Data Type (ADT)" คือ Model ที่มี Operation ของมันที่จะกระทบต่อ Model

ข้อดี ของ Encapsulation คือ สามารถลดความซับซ้อนลง เนื่องจากสามารถใช้งานโดยรู้เพียงคุณลักษณะต่าง ๆ เท่านั้น โดยไม่ต้องรู้ถึงโครงสร้างภายใน

### ● Inheritance

การสืบทอดคุณสมบัติทั่ว ๆ ไปที่เหมือนกันระหว่าง Class

ตัวอย่าง Class ของผู้ชาย inherits Class ของคน

ข้อดี ของ Inheritance คือ

1. สามารถนำนิยามโดยทั่ว ๆ ไปมาใช้ใหม่ (Reuse common description)
2. การปรับปรุงแก้ไข ทำเพียงที่เดียวเท่านั้น (Modification can be made in only one place)
3. ลดความซ้ำซ้อนสำหรับโปรแกรมย่อยเพื่อให้ง่ายต่อการทำความเข้าใจ (Redundancy can be reduced leading to smaller models that are easier to understand)

"Overriding" คือคุณสมบัติที่สามารถนิยาม behavior หรือ information ใหม่ได้สำหรับ Class ที่สืบทอดมาจาก Super Class

### ● Polymorphism

โดยความหมายคือ "having many forms" หมายถึงการส่ง message เดียวกันให้กับ Object ที่ต่างกันจะแสดงพฤติกรรมที่ต่างกัน

ตัวอย่าง การบวกของ String และ int จะมีสัญลักษณ์เดี๋ยวกคือ + แต่ทำกับข้อมูลต่างชนิดกัน

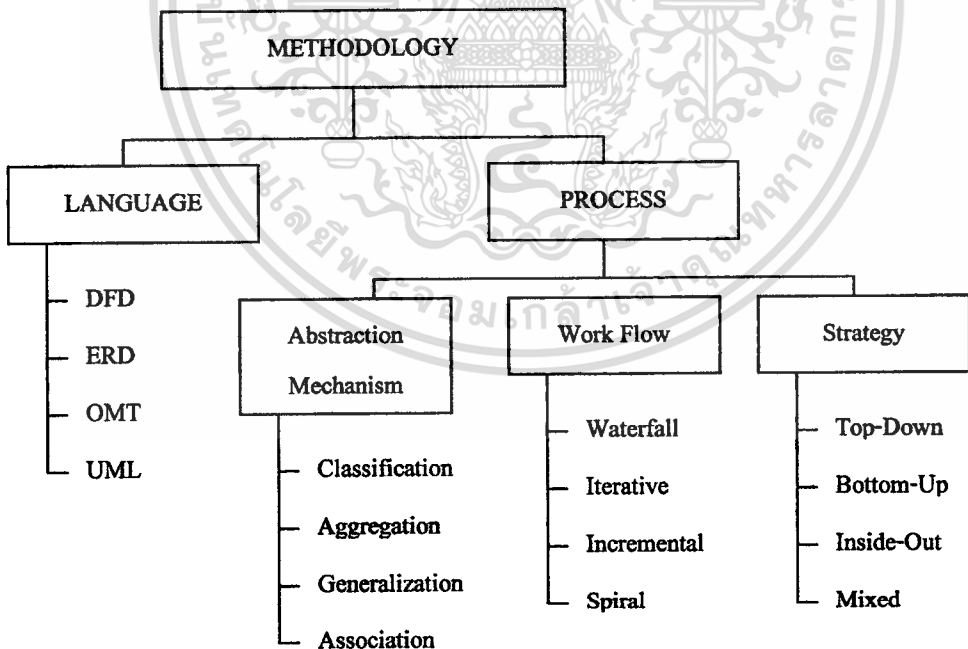
ข้อดี ของ Polymorphism คือ สนับสนุนการนำกลับมาใช้ใหม่ (Reuse) และมีการเปลี่ยนแปลงแก้ไขได้

## 2.4 Object-Oriented Methodology

การพัฒนากระบวนวิธีด้วย Object-Oriented Methodology นักวิเคราะห์ระบบจะต้องมีแนวคิดหรือระเบียบวิธีคิดในเชิงวัตถุที่ได้กล่าวมาแล้ว ตั้งแต่กระบวนการวิเคราะห์ ออกแบบ การเขียนโปรแกรม ไปจนถึงการทดสอบการทำงานของระบบ หรือ โปรแกรม (OO Analysis, OO Design, OO Programming และ OO testing) ยกตัวอย่างเช่น การวิเคราะห์ระบบงาน จะต้องวิเคราะห์ให้ได้ว่าควรมี Class อะไรบ้าง แต่ละ Class มี พฤติกรรมการสื่อสารระหว่างกันอย่างไร เกิดเหตุการณ์อะไรบ้าง เป็นต้น จึงจะได้ออกมาเป็นระบบที่ต้องการ ซึ่งต่างจากการวิเคราะห์และออกแบบระบบแบบดั้งเดิมที่มองแยกกันระหว่างการออกแบบในส่วนของคุณสมบัติ กับส่วนของกระบวนการทำงาน

## 2.5 ส่วนประกอบที่สำคัญของ Methodology

System Modeling Methodology โดยทั่วไปประกอบด้วย 2 ส่วน คือ Language และ Process Model โดย Language คือสัญลักษณ์ (notation) ที่ใช้ในการอธิบาย แสดงความหมายของระบบงาน ส่วน Process หมายถึง การบอกกล่าวว่าจะทำอะไรในขั้นตอนใดของการวิเคราะห์และออกแบบระบบงาน โดยสามารถแสดงได้ดังรูป



รูปที่ 2-6 โครงสร้างส่วนประกอบของ System Development Methodology

## 2.6 UML คืออะไร

UML (Unified Modeling Language) เป็นภาษาหรือรูปแบบจำลอง มาตรฐานที่ใช้ในการจำลองแบบในรายละเอียด (Blue Print) สำหรับการผลิตสร้าง (Construct) ซอฟต์แวร์ เช่นเดียวกับแบบจำลองทางธุรกิจ (Business Modeling) โดย UML จะเป็นการเสนอรูปแบบการปฏิบัติจริงในทางวิศวกรรมที่ดี (Best Engineering practices) อีกทั้งยังเหมาะกับระบบที่มีขนาดใหญ่และยุ่งยากซับซ้อนอีกด้วย

UML เป็นรูปแบบที่รวมเอาข้อดีของรูปแบบอื่น ๆ เข้ามาผสมผสานไว้ด้วยกัน คือ

- Data Modeling Concept (Entity Relationship Concept)
- Business Modeling (Work Flow)
- Object Modeling
- Component Modeling

นอกจากนี้ UML ยังสามารถใช้ได้ตลอดทั้งโครงการหรือการทำงาน เช่นเดียวกับ SDLC (Software Development Life Cycle)

## 2.7 ประวัติศาสตร์ของ UML

การกำหนดรูปแบบภาษาการวิเคราะห์และออกแบบเชิงวัตถุ เริ่มปรากฏในระหว่างกลางทศวรรษที่ 1970 ถึงปลายทศวรรษที่ 1980 มีการทดลองด้วยแนวความคิดที่แตกต่างกันอย่างแพร่หลายเกี่ยวกับการวิเคราะห์ และออกแบบเชิงวัตถุ และในช่วงระหว่างปี 1989-1994 ได้มีการเพิ่มจำนวนของรูปแบบภาษาจากเดิมซึ่งมีน้อยกว่า 10 รูปแบบไปเป็นมากกว่า 50 รูปแบบ และผู้ใช้งานจำนวนมากที่ใช้วิธีการเชิงวัตถุ (Object-Oriented) ก็ยังไม่พบกับวิธีที่ตนเองพอใจ

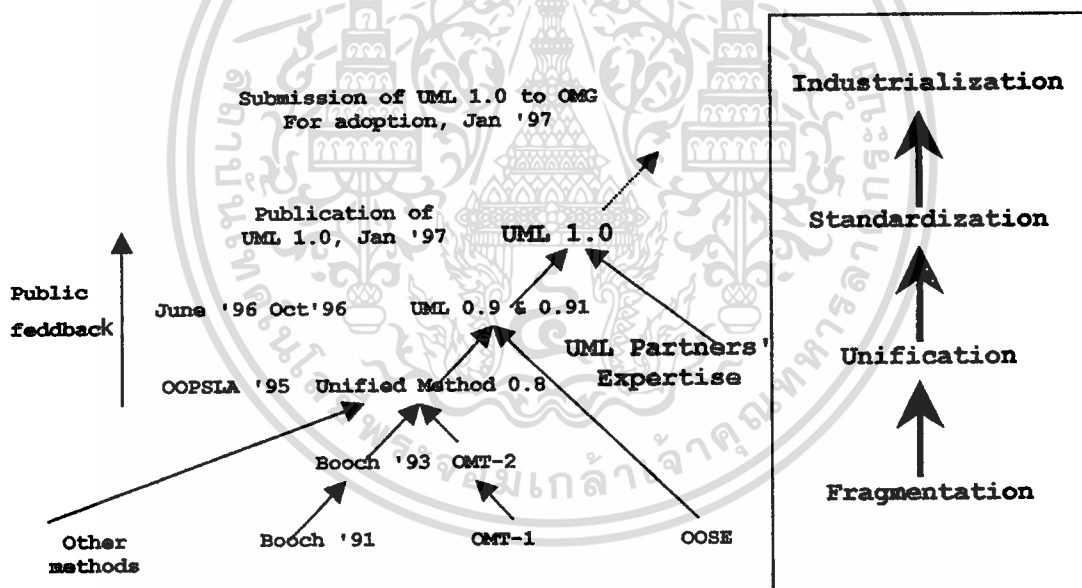
จนกระทั่งกลางทศวรรษที่ 1990 ได้เริ่มมีการพัฒนา UML ขึ้นในเดือนตุลาคม 1994 โดย Grady Booch และ Jim Rumbaugh ของ Rational Software Corporation ได้มีการรวมรูปแบบของ Booch และ OMT (Object Modeling Technique) เพื่อศึกษาวิธีการในเชิงวัตถุ หลังจากนั้นในเดือนตุลาคม 1995 ก็มีรูปแบบของ Unified Method 0.8 แต่ก็ไม่ยังประสบความสำเร็จ และในปี 1995 นั้น Ivar Jacobson ก็ได้นำรูปแบบนั้นมารวมกับวิธี OOSE (Object-Oriented Software Engineering) สุดท้ายในปี 1996 จึงมี UML 1.0 ออกมาเป็นมาตรฐานโดยการร่วมมือของหลาย ๆ บริษัทเช่น Digital Equipment Corp., I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, TI และ Unisys

ปัจจุบัน UML ได้รับการรองรับจาก OMG (Object Management Group) ให้เป็นมาตรฐานหนึ่ง และไม่มีใครเป็นเจ้าของ เป็นรูปแบบเปิด กฎเกณฑ์ที่ทำให้ UML ประสบความสำเร็จคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. UML เป็นสิ่งที่ทำแล้วได้ผลในหลาย ๆ วิธีที่เป็นลักษณะรูปแบบทางภาษาที่เคยเสนอมาในอดีต
2. UML เป็นการรวมรูปแบบที่ใช้ได้ในหลาย ๆ ระบบ (Business versus software) , ขั้นตอนการพัฒนาในระบบ (Development phase)

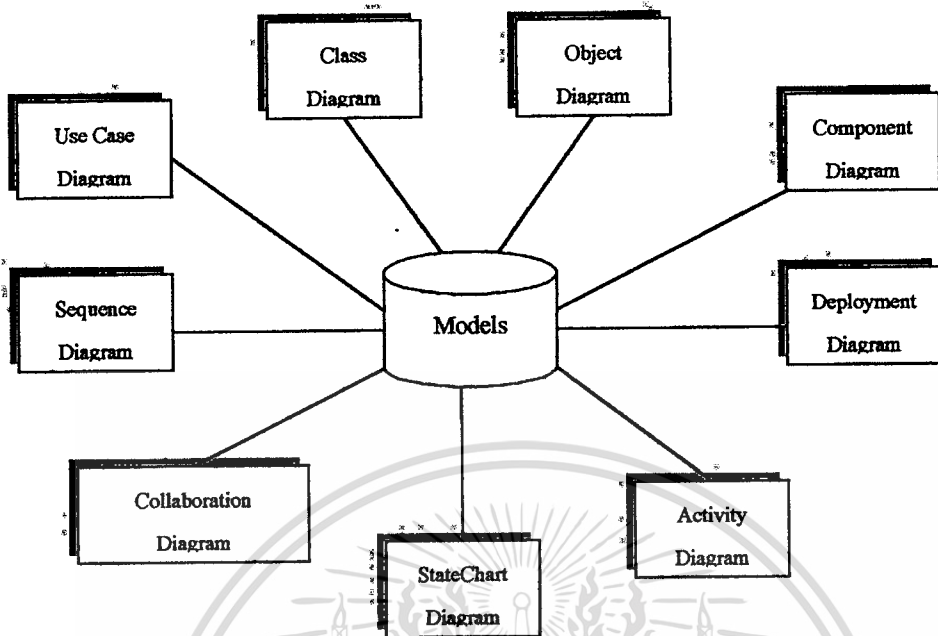
หลาย ๆ องค์กรได้มีการสนับสนุนมาตรฐานของ UML ตั้งแต่พื้นฐานรูปแบบทางภาษา (Modeling Language) ซึ่งเป็นวิธีการเชิงวัตถุ UML เป็นที่ยอมรับกันอย่างกว้างขวาง โดย UML 1.0 เป็นเวอร์ชันที่สามารถใช้ได้และมีเสถียรภาพ ในเดือนมกราคม 1997 เอกสารเกี่ยวกับ UML Version 1.0 ก็ได้รับการตอบสนองโดย Object Management Group (OMG) Analysis & Design Task Force's RFP -1 และในกลางปี 1997 OMG ได้มีการยอมรับมาตรฐานของ UML จึงทำให้หลาย ๆ องค์กรและผู้ผลิตได้ยอมรับมาตรฐาน UML จึงมีการใช้การสนับสนุนอื่น ๆ ทางด้าน UML มีเครื่องมือที่ช่วยในการพัฒนา ตลอดจนองค์กรที่ช่วยในการฝึกอบรม



รูปที่ 2-7 แผนภาพแสดงประวัติศาสตร์ของ UML

## 2.8 ประเภทของ UML Diagram

UML ได้กำหนด diagram ทั้งหมด 9 diagram ดังแสดงในรูป 2-8 เพื่อใช้ในการจำลองแบบระบบงานโดยการมองในแง่มุมต่าง ๆ (diagram แต่ละชนิด) เพื่อให้สามารถเข้าใจระบบงานให้มากที่สุด โดยผู้จำลองแบบไม่จำเป็นต้องใช้ทุก diagram โดยสามารถเลือกใช้ Diagram ที่เหมาะสม



รูปที่ 2-8 แผนภาพแสดง UML Diagram ทั้งหมด 9 Diagrams

โดย UML ทั้ง 9 Diagram สามารถจัดแบ่งออกเป็นกลุ่มต่าง ๆ ได้ดังนี้

### 2.8.1 Functional Model

Functional Model เป็นแบบจำลองที่ใช้ในการแสดงความต้องการของระบบทั้งหมด ช่วยในการอธิบายรายละเอียดหลัก ๆ ภายในวัตถุ แสดงให้เห็นการไหลของข้อมูลในแต่ละการทำงาน โดยจะสนใจเพียงแค่ว่ามีงานอะไรบ้างที่ต้องทำ เครื่องมือที่ใช้ในการแสดงความต้องการของระบบคือ Use Case Diagram ใช้ในการแสดงความต้องการของระบบทั้งหมดในลักษณะที่ผู้ใช้งานสามารถเข้าใจได้ง่าย โดยจะถูกนำไปใช้ต่อไปใน Phase ต่าง ๆ ของการวิเคราะห์และออกแบบระบบ

### 2.8.2 Object Model

Object Model เป็นแบบจำลองที่ใช้ในการแสดงโครงสร้างของระบบ โดยจะแสดงในรูปแบบของ Class ต่าง ๆ พิจารณาจากความต้องการของระบบ ที่แสดงอยู่ใน Functional Model เครื่องมือที่ใช้ในการแสดงโครงสร้างของระบบคือ Component Diagram แสดงถึงโครงสร้างและความสัมพันธ์ขององค์ประกอบต่าง ๆ ของซอฟต์แวร์ ส่วน Class Diagram แสดงถึงโครงสร้างและความสัมพันธ์ของ Class ส่วน Object Diagram แสดงถึงโครงสร้างและความสัมพันธ์ของ Instance และ Deployment Diagram แสดงถึงความสัมพันธ์ทางกายภาพระหว่างส่วนประกอบของฮาร์ดแวร์และซอฟต์แวร์

### 2.8.3 Dynamic Model

Dynamic Model เป็นแบบจำลองที่ใช้ในการแสดงถึงการทำงานระหว่าง Object ต่าง ๆ ตามการส่งข้อความ (Message) หรือเมื่อเหตุการณ์ (Event) ต่าง ๆ ได้เกิดขึ้น Object ในที่นี้หมายถึง Instance ที่สร้างขึ้นจาก Class ที่ได้ออกแบบไว้ใน Object Model มีคุณสมบัติและพฤติกรรมเช่นเดียวกับ Class ต้นแบบ ในการทำงานของระบบจะประกอบขึ้นจากการส่งข้อความไปมาระหว่าง Object เหล่านั้น เมื่อมีการทำงานไปเรื่อย ๆ แล้ว Object อาจจะมีการเปลี่ยนแปลงสถานะ (State) ไปตามเหตุการณ์ที่เกิดขึ้นได้ เพื่อให้เป็นไปตามที่กำหนดไว้ใน Function Model เครื่องมือที่ใช้คือ Sequence Diagram ที่จะแสดงให้เห็นถึงการแลกเปลี่ยนข่าวสารระหว่าง Object ต่าง ๆ, Collaboration Diagram ที่แสดงถึงความสัมพันธ์ระหว่าง Object ภายใน, Activity Diagram ที่แสดงถึงสถานะการทำงานในแต่ละ Action และ State Diagram แสดงวงจรชีวิตของ Object บอกถึงว่าเหตุการณ์ต่าง ๆ มีผลกระทบอะไรเกิดขึ้นบ้าง

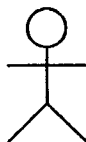
### 2.9 ความหมายของแต่ละ UML Diagram

ความหมายและสัญลักษณ์ต่าง ๆ ที่ใช้ใน UML Diagram ที่เลือกใช้ในระบบงานซึ่งประกอบไปด้วย 4 Diagram ได้แก่ Use Case Diagram, Class Diagram, Component Diagram และ Sequence Diagram

#### 2.9.1 Use Case Diagram

Use Case Diagram เป็นแนวความคิดของ Ivar Jacobson ซึ่ง OMG ได้รวมไว้ในมาตรฐานของ UML ด้วย และมีวิธีการอื่น ๆ นำแนวคิดนี้ไปใช้อย่างกว้างขวาง Use Case Diagram เป็นสิ่งที่ใช้ในการแสดงความต้องการของระบบทั้งหมดในลักษณะที่ผู้ใช้งานสามารถเข้าใจได้ง่าย โดยแสดงความสัมพันธ์ระหว่างผู้ใช้กับระบบต่าง ๆ ประกอบไปด้วยสัญลักษณ์ดังต่อไปนี้

- **Actor** คือสิ่งที่อยู่ภายนอกระบบที่จะพัฒนาขึ้นทั้งหมด ที่ต้องการแลกเปลี่ยนหรือส่งข้อมูลให้กับระบบ โดยที่ Actor อาจจะเป็นคน ระบบ หรือ โปรแกรมอื่น ๆ ก็ได้ โดยมากจะเป็นผู้เริ่มทำงานกับ Use Case เช่น ผู้ใช้ระบบ เป็นต้น



Human

รูปที่ 2-9 สัญลักษณ์ของ Actor

- Use Case จะเป็นตัวแทนงานที่เกิดขึ้นในขั้นตอนต่าง ๆ โดยจะใช้สัญลักษณ์เป็นรูปวงรีหรือวงกลมถ้า Use Case ใดมีกรอบสี่เหลี่ยมสีเทาล้อมรอบแสดงว่ามี Diagramย่อยที่ใช้อธิบายรายละเอียดของ Use Case ต่อไป



รูปที่ 2-10 สัญลักษณ์ของ Use Case

- **Communication** เป็นการแสดงความสัมพันธ์หรือการติดต่อสื่อสารกัน (การรับและให้ข้อมูลข่าวสารแก่กันและกัน) ระหว่าง Actor และ Use Case ซึ่งอาจจะเป็นการสื่อสารแบบทางเดียวหรือ 2 ทางก็ได้ แสดงด้วยสัญลักษณ์ที่มีหัวลูกศรสี่ด้าน



รูปที่ 2-11 สัญลักษณ์ของ Communication

- **Relationship** เป็นการแสดงความสัมพันธ์ระหว่าง Use Case กับ Use Case โดยใช้เส้นที่มีหัวลูกศรสี่ด้าน ความสัมพันธ์ระหว่าง Use Case มี 2 แบบ คือ Extends และ Uses



รูปที่ 2-12 สัญลักษณ์ของ Relationship

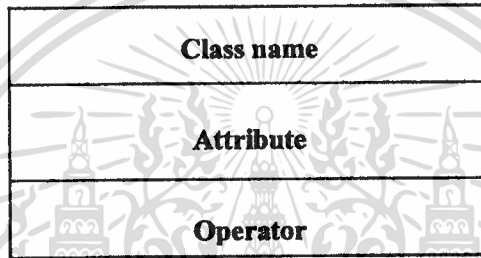
Extends เป็นการเพิ่มการทำงานให้กับ Use Case โดยการเรียกใช้จากอีก Use Case หนึ่ง ลูกศรจะออกจาก Use Case ที่ถูกเรียกใช้งานไปยัง Use Case ที่ต้องการเพิ่มเติมการทำงาน และจะมีความ <<Extends>> ระบุอยู่ข้าง ๆ เส้น

Uses เป็นการแสดงให้เห็นถึงการ Inherit หน้าที่การทำงานจาก Use Case หนึ่ง ไปอีก Use Case หนึ่ง ปลายลูกศรจะอยู่ที่ Use Case หลักที่จะถูกถ่ายทอดความสามารถออกไป โดยจะมีความ <<Uses>> ปรากฏอยู่ข้าง ๆ เส้น

## 2.9.2 Class Diagram

Class Diagram เป็น Model ชนิดหนึ่งที่เป็น Static Model จะเป็นการอธิบายถึง Class และความสัมพันธ์ระหว่าง Class ที่มีโครงสร้างของข้อมูลรวมถึงพฤติกรรมของข้อมูลที่แตกต่างกันซึ่ง Class 1 Class สามารถกำหนดทิศทางของการนำไปสร้างโปรแกรมและการสร้าง Class ใน Object Oriented ได้

Class ใช้สัญลักษณ์เป็นรูปสี่เหลี่ยม จะอธิบายถึงกลุ่มของ Object ต่าง ๆ ที่มีคุณลักษณะ (Attribute) หน้าที่การทำงาน (Operation) และความสัมพันธ์ (Relationship) ขั้นพื้นฐาน ซึ่งในแต่ละ Class ประกอบไปด้วยรายละเอียดพื้นฐาน 3 ส่วนดังรูป



รูปที่ 2-13 สัญลักษณ์ของ Class


- Class Name : แสดงชื่อของ Class ที่กำหนดในระบบ
- Attribute : เป็นการกำหนดคุณลักษณะทั้งหมดที่มีภายใน Class บอกถึงรายละเอียดของข้อมูล โดย Attribute ภายใน Class เดียวกันจะไม่ซ้ำกัน แต่ชื่อข้อมูลใน Class อาจจะไปซ้ำกับชื่อข้อมูลใน Class อื่นได้
- Operation : เป็นส่วนที่ใช้อธิบายว่าภายใน Class นั้นมี Method อะไรบ้าง มีการรับค่าชุดของตัวแปร (Argument) อะไรบ้าง หรือมีการส่งค่าออกไปหรือไม่
- Relationship : ความสัมพันธ์ระหว่าง Class โดย UML มีรูปแบบของความสัมพันธ์ระหว่าง Class อยู่ 4 แบบ ดังนี้
  1. Association: จะแสดงความสัมพันธ์ระหว่าง Class จะมีได้ทั้งทางเดียวและ 2 ทาง โดยมีสัญลักษณ์ดังนี้

ความสัมพันธ์แบบทางเดียว :

ความสัมพันธ์แบบสองทาง :

รูปที่ 2-14 สัญลักษณ์ของ Association relationship

2. Aggregation : เป็นรูปแบบพิเศษของ Association คือเป็นความสัมพันธ์ระหว่าง Whole และ Parts ของมัน คือ Whole จะประกอบไปด้วย Parts ต่าง ๆ ของมัน ดังนั้นการคงอยู่ของ Parts จะต้องขึ้นกับ Whole หรือจะเรียกความสัมพันธ์แบบนี้ว่า "Whole-Parts" ก็ได้โดยมีสัญลักษณ์ดังนี้

ความสัมพันธ์แบบ by Value :  \_\_\_\_\_

ความสัมพันธ์แบบ by Reference :  \_\_\_\_\_

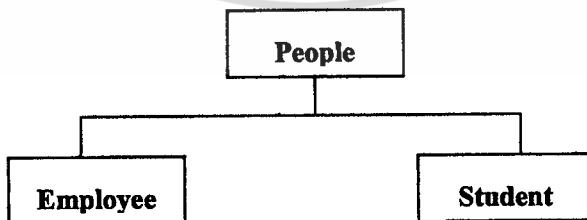
### รูปที่ 2-15 สัญลักษณ์ของ Aggregation relationship

3. Depends on : เป็นรูปแบบความสัมพันธ์แบบหนึ่งที่ใช้แสดงความสัมพันธ์กันระหว่าง Class 2 Class ในแง่ที่ Class หนึ่งเรียกใช้บริการของอีก Class หนึ่ง กล่าวคือ Class ของผู้ขอใช้บริการนั้นจะขึ้นอยู่กับบริการของ Class ของผู้ให้บริการ แต่ไม่มีการขึ้นต่อกันภายในโครงสร้างของ Class โดยมีสัญลักษณ์ดังนี้



### รูปที่ 2-16 สัญลักษณ์ของ Depend on relationship

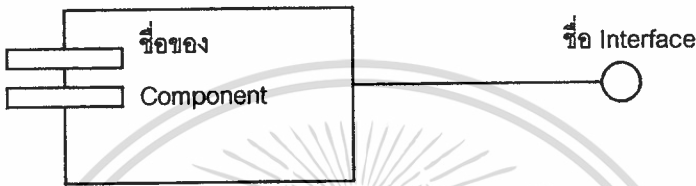
4. Generalization : ความสัมพันธ์รูปแบบนี้ใช้แสดงความสัมพันธ์ระหว่าง Class กับ Class ในแง่ที่ Class หนึ่งถ่ายทอดคุณสมบัติและโครงสร้างจากอีก Class หนึ่ง โดยเรียก Class ที่ถูกถ่ายทอดว่า Sub Class และเรียก Class ที่ทำการถ่ายทอดว่า Super Class โดยมีสัญลักษณ์ดังนี้



### รูปที่ 2-17 ตัวอย่างความสัมพันธ์แบบ Generalization

### 2.9.3 Component Diagram

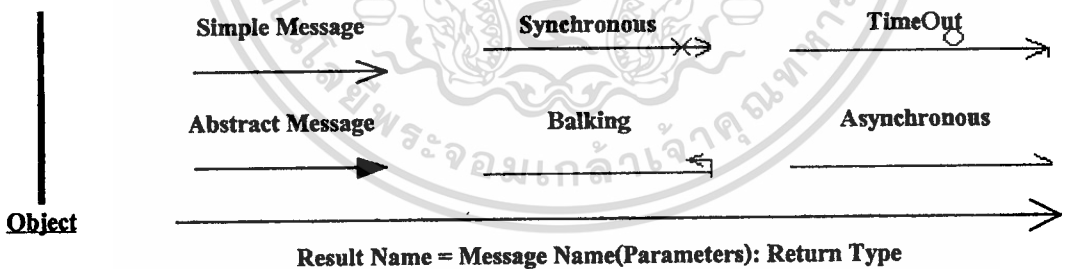
เป็น Diagram ที่แสดงถึงโครงสร้างและความสัมพันธ์ระหว่างองค์ประกอบต่าง ๆ ของซอฟต์แวร์ ซึ่งองค์ประกอบดังกล่าวอาจเป็น Source Program หรือ Library หรือ Executable Program ก็ได้ โดย Component Diagram จะเป็นแบบจำลองที่ประกอบด้วยองค์ประกอบต่าง ๆ ของระบบเชื่อมโยงกันโดยใช้ความสัมพันธ์ในลักษณะที่ขึ้นต่อกันในรูปของเส้นประที่มีหัวลูกศรชี้จาก Component ถูก ไปยัง Component หลัก ดังรูป



รูปที่ 2-18 สัญลักษณ์ที่ใช้ภายใน Component Diagram

### 2.9.4 Sequence Diagram

เป็น Diagram ที่แสดงให้เห็นถึงการทำงานระหว่าง Object ต่าง ๆ ตามการส่งข้อความ และเมื่อเหตุการณ์ต่างๆ ได้เกิดขึ้น ซึ่งผู้เขียนโปรแกรมจะใช้ Diagram ตัวนี้เป็นตัวช่วยเพื่อที่จะได้เขียนโปรแกรมให้ได้ตามที่ออกแบบไว้ สัญลักษณ์ที่ใช้ใน Sequence Diagram มีดังนี้



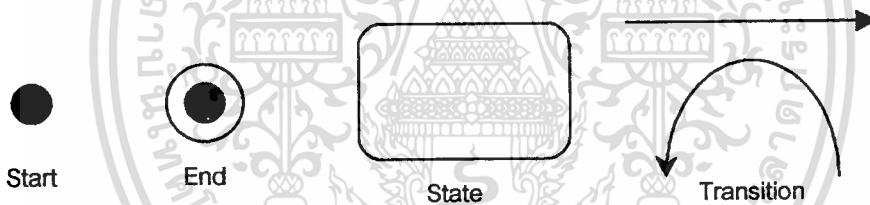
รูปที่ 2-19 สัญลักษณ์ต่างๆ ที่ใช้ใน Sequence Diagrams

- Object : คือ Instance ของ Class ที่ทำหน้าที่รับ-ส่งข้อความ (Message) จาก Object อื่น ๆ และทำการตอบสนองตามข้อความนั้น ๆ เพื่อให้เกิดการทำงานในขั้นตอนต่าง ๆ ของระบบ โดยจะใช้สัญลักษณ์เป็นเส้นตรงแนวตั้งและมีชื่อของ Object และ Class กำหนดอยู่ด้านต่าง
- Message : เป็นข้อความที่ส่งไปมาระหว่าง Object ที่ถูกเรียก ซึ่ง Message จะประกอบไปด้วยส่วนประกอบต่าง ๆ ดังต่อไปนี้
  - Result Name : คือตัวแปรที่จะจัดเก็บค่าที่ได้จากการส่งข้อความ (Message)

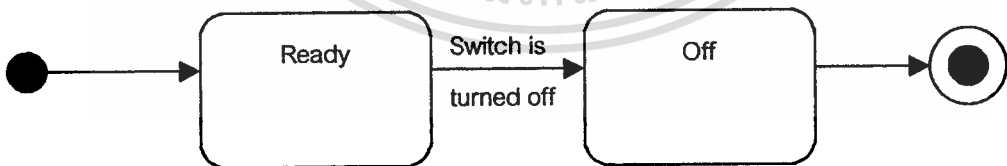
- Parameters : คือ Argument ต่าง ๆ ที่ส่งให้กับ Object เพื่อทำงานตาม Method ที่กำหนดด้วย Message นั้น ๆ
- Return Type : คือประเภทของข้อมูลที่ส่งกลับมาหลังจากการทำงานของ Method นั้น ๆ ถ้า Method นั้นกำหนดว่ามีการส่งค่ากลับมา
- Constraint : คือเงื่อนไขการส่ง Message

### 2.9.5 State Diagram

State Diagram เป็นส่วนที่ใช้ประกอบคำอธิบาย Class โดยจะแสดงทุก ๆ สถานะที่เป็นไปได้และเหตุการณ์ที่ทำให้ Object ต่าง ๆ ใน Class เกิดการเปลี่ยนแปลง โดยเหตุการณ์ที่ทำให้เปลี่ยนแปลงอาจเกิดจาก Object อื่น ๆ ส่ง Message มาเช่น เวลาเปลี่ยนไป หรือมีเงื่อนไขใดๆ เกิดขึ้น โดยการเปลี่ยนแปลงสถานะเรียกว่า Transition ซึ่ง State Diagram ไม่จำเป็นจะต้องวาดขึ้นสำหรับทุก Class แต่จะใช้แสดงสำหรับ Class ซึ่งมีการกำหนดสถานะไว้ชัดเจนและเมื่อการกระทำได้รับผลหรือถูกเปลี่ยนแปลงไปในสถานะต่างๆ สัญลักษณ์ต่าง ๆ ที่ใช้ใน State Diagram แสดงดังรูป 2-20 และตัวอย่าง State Diagram แสดงดังรูป 2-21



รูปที่ 2-20 สัญลักษณ์ต่าง ๆ ที่ใช้ใน State Diagram



รูปที่ 2-21 State Diagram แสดงการปิดเครื่อง

สำหรับในส่วนของการวิเคราะห์และออกแบบระบบงานใหม่โดยอาศัย Diagrams ต่าง ๆ ของ Unified Modeling Language (UML) ดังที่ได้กล่าวมาแล้วข้างต้นนั้น จะนำเสนอในส่วนของ บทที่ 4 ในเรื่องของการวิเคราะห์และออกแบบระบบใหม่ต่อไป

## บทที่ 3

### การศึกษาการดำเนินงานและปัญหาของระบบงานปัจจุบัน

สำหรับในบทนี้เราจะพูดถึงการดำเนินงานและปัญหาของระบบงานในปัจจุบันว่ามีรูปแบบการทำงานอย่างไร มีงานอะไรบ้างที่ต้องทำ และในการทำงานในรูปแบบที่ใช้อยู่ในปัจจุบันนั้นเกิดปัญหาอะไรขึ้นมาบ้าง เพื่อที่จะได้นำเอาข้อมูลเหล่านี้ไปใช้ในการวิเคราะห์และออกแบบระบบงานใหม่ที่จะทำการพัฒนาขึ้นมา

#### 3.1 การดำเนินงานของระบบงานปัจจุบัน

การจัดรายการเพลงของสถานีวิทยุส่วนมากนั้นจะเป็นการจัดรายการสด กระจายเสียง ณ ขณะนั้นมีการโต้ตอบกับผู้ฟังแบบทันทีทันใด ซึ่งจะประกอบไปด้วยการเล่นเพลงสลับกับการเปิดสปอตโฆษณา หรือจึงเกิดเพลง โดยทั้งหมดจะต้องเป็นการดำเนินงานของนักจัดรายการวิทยุ และในบางครั้งเพื่อให้การทำงานทันกับเวลา ทันกับความต้องการก็จะต้องมีผู้ช่วยจัดรายการวิทยุ ช่วยดำเนินรายการด้วย ซึ่งการทำงานทั้งหมดเป็นการทำด้วยมือทั้งสิ้น

โดยสามารถแบ่งการดำเนินงานออกเป็น 3 ส่วนที่สำคัญคือ การจัดเก็บเพลง การค้นหาเพลงเพื่อเล่นเพลง การเปิดสปอตโฆษณา

#### 3.1.1 การจัดเก็บเพลงหรือข้อมูลเพลง

อัลบั้มเพลงต่าง ๆ จะถูกจัดเก็บไว้ในชั้นวางแผ่น CD ภายในห้องจัดรายการวิทยุ โดยไม่มีการจัดเก็บเพลงลงในคอมพิวเตอร์ หรือเล่นเพลงผ่านโปรแกรมคอมพิวเตอร์ใด ๆ โดยจะมีการแบ่งชั้นวางแผ่น CD เป็น 2 ส่วน คือ

- ชั้นวางแผ่น CD ใหม่ หรือแผ่น CD ที่มีการเปิดบ่อย ๆ ซึ่งจะอยู่ใกล้กับนักจัดรายการวิทยุ
- ชั้นวาง CD อื่นนอกจากที่กล่าวข้างต้น ซึ่งจะตั้งอยู่รอบห้อง

หากมีแผ่น CD ออกใหม่ก็จะทำการจัดเก็บลง CD ใหม่ที่ลงในชั้นวางแผ่น CD ใหม่ทุกครั้งและย้ายแผ่นที่เปิดน้อยลงไปยังชั้นวาง CD อื่นแทน

โดยในแต่ละสถานีก็จะมีวิธีการจัดวางลำดับของ CD บนชั้นวางที่ต่างกันขึ้นอยู่กับแนวทางในการจัดรายการของสถานีวิทยุนั้น ๆ เช่น สถานีเพลงฮิตก็อาจเรียงตามวันที่ออกอัลบั้ม สถานีเพลงรักก็อาจเรียงตามชื่ออัลบั้มหรือชื่อศิลปิน เป็นต้น

ข้อมูลเพลงด้านต่าง ๆ นั้นนักจัดรายการวิทยุจะทราบได้โดยการอ่านจากปกอัลบั้มหรือเอกสารต่าง ๆ ทั่ว ๆ ไป แต่จะไม่มีกรบันทึกหรือจัดเก็บข้อมูลไว้เป็นเรื่องเป็นราว

การบันทึกอันดับเพลงฮิต การขอเพลง หรือลำดับของการเปิดเพลง จะเป็นการบันทึกลงในกระดาษซึ่งทำด้วยมือทั้งหมด ไม่มีรูปแบบของเอกสารที่เป็นมาตรฐานเดียวกัน โดยขึ้นอยู่กับกรบันทึกของนักจัดรายการแต่ละคน แล้วนำมาสรุปรวมกันในแต่ละช่วงวันหรือสัปดาห์

### 3.1.2 การค้นหาและเล่นเพลง

นักจัดรายการวิทยุจะอาศัยความชำนาญในการจำว่าเพลงที่ต้องการค้นหาเป็นของศิลปินคนใด อยู่ในอัลบั้มใด และตำแหน่งในการเก็บอัลบั้มอยู่ที่ใด ซึ่งจะทำการค้นหาจากชั้นวาง CD ใหม่ก่อน และจะต้องหาดำแหน่งของ CD บนชั้นวางอีกครั้งเนื่องจากตำแหน่งบนชั้นอาจถูกเปลี่ยนเมื่อมีแผ่นใหม่ ๆ เข้ามา จึงไม่สามารถใช้การจำตำแหน่งได้ อาจจะมีการจำลักษณะของปกอัลบั้มหรือตัวหนังสือบนแผ่น CD แทน

การจัดรายการเพลงจะมี 2 ลักษณะคือ

- มีการโต้ตอบกับผู้ฟัง และเปิดเพลงตามที่ผู้ฟังขอเข้ามา โดยนักจัดรายการวิทยุจะเป็นผู้ค้นหาแผ่นเพลงและเล่นเพลงเอง พร้อมทั้งเป็นคนเปิดจิงเกิ้ลและสปอตโฆษณาเองทั้งหมด ลักษณะการทำงานเป็นแบบโต้ตอบทันทีทันใด (Interactive) และเป็นการทำด้วยมือทั้งหมด
- ช่วงการเปิดเพลงต่อเนื่องซึ่งจะไม่มีการเล่นเกมส์ หรือพูดคุยโต้ตอบกับคนฟังเลย โดยนักจัดรายการวิทยุเป็นผู้จัดรายการเพลงที่ต้องการเปิดเป็นชุดต่อเนื่องลงในกระดาษไว้และเมื่อถึงเวลาออกอากาศ ผู้ช่วยจัดรายการจะเป็นคนเปลี่ยนแผ่น CD ตามรายการนั้น ลักษณะการทำงานเป็นไปตามแผนที่วางไว้ และเป็นการทำด้วยมือเองทั้งหมดเช่นกัน

### 3.1.3 การเปิดสปอตโฆษณาหรือจิงเกิ้ลเพลง

สปอตโฆษณา คือ โฆษณาที่ขึ้นระหว่างการจัดรายการเพลงในแต่ละช่วง

จิงเกิ้ลเพลง คือ เสียงเพลง เสียงพูด หรือเสียงร้องสั้น ๆ เพื่อประกอบการจัดรายการวิทยุในแต่ละช่วงการจัดรายการ

โดยการเปิดสปอตโฆษณา หรือจิงเกิ้ลเพลงในปัจจุบัน จะอาศัยการเล่นจากเครื่องเล่น CD ที่สามารถสลับการเปิดระหว่างแผ่น CD ได้ 4 แผ่น ซึ่งนักจัดรายการวิทยุจะต้องเป็นคนเลือกสปอตโฆษณา หรือจิงเกิ้ลเพลงที่ต้องการและจัดวางแผ่น CD เอง รวมถึงการตั้งตำแหน่งในการเล่นแต่ละสปอตโฆษณา หรือจิงเกิ้ลเพลงเองทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 ปัญหาของระบบงานปัจจุบัน

เนื่องจากระบบการจัดรายการเพลง จะเกี่ยวข้องกับเพลงจำนวนมาก และการทำงานในปัจจุบันยังคงเป็นการทำงานด้วยมือทั้งหมด ดังนั้นการค้นหาข้อมูลที่ต้องการจากข้อมูลจำนวนมาก การปรับปรุงข้อมูล บันทึกข้อมูลใหม่ ย่อมทำได้ลำบาก เสียเวลา ไม่สะดวกในการทำงาน และยังมีโอกาสเกิดความผิดพลาดได้สูงอีกด้วย ซึ่งสามารถแยกเป็นข้อ ๆ ได้ดังนี้

- เมื่อเวลาผ่านไป เพลงใหม่ ๆ จะมีจำนวนมากขึ้น หรือ เพลงเก่าอาจมีการนำมาทำใหม่ในจังหวะที่ต่างออกไป หรือเพลงที่มีชื่อเหมือนกันมีมากขึ้น ทำให้การค้นหาเพลงทำได้ลำบากมากขึ้น และใช้เวลาในการค้นหามากขึ้น
- ห้องจัดรายการมีขนาดจำกัด และเมื่อแผ่น CD มีจำนวนมากขึ้นเรื่อย ๆ ทำให้ห้องไม่พอที่จะวางแผ่น CD ทั้ง จึงจำเป็นต้องคัดเลือกแผ่น CD ที่เปิดน้อยมากออกจากชั้นเก็บทำให้เพลงเก่า ๆ หายไปจากระบบ
- ต้องคอยสลับตำแหน่งและจดจำตำแหน่งใหม่ในการจัดเก็บแผ่น CD อย่างสม่ำเสมอ และใช้เวลามากในการปรับปรุงข้อมูล จัดเรียงแผ่น CD ต่าง ๆ ที่มีอยู่ในระบบทั้งหมด
- เนื่องจากในแต่ละสถานีวิทยุมีรูปแบบการจัดเรียงแผ่น CD ที่ต่างกันทำให้นักจัดรายการวิทยุมือใหม่ หรือนักจัดรายการวิทยุที่ประจำอยู่สถานีหนึ่ง เมื่อไปจัดรายการอีกสถานีหนึ่ง จะต้องเรียนรู้รูปแบบการจัดวาง CD ของสถานีวิทยุนั้น ๆ ใหม่ ซึ่งทำให้เสียเวลาและเกิดความสับสนได้
- ในบางครั้งการขอเพลงจากผู้ฟังอาจจะไม่ทราบชื่อเพลง ทราบเพียงเนื้อร้องในบางท่อนหรือข้อมูลบางอย่าง ซึ่งต้องอาศัยความสามารถเฉพาะตัวของนักจัดรายการวิทยุในการค้นหาเพลงให้ได้ตามต้องการ
- ความต้องการของนักจัดรายการวิทยุ ไม่ได้มีเพียงแค่ค้นหาเพลงได้รวดเร็วเท่านั้นยังต้องการที่จะทราบข้อมูลที่เกี่ยวข้องกับเพลงนั้น ๆ เช่นเนื้อร้อง ข้อมูลศิลปิน หรือข้อมูลทางด้านดนตรีอื่น ๆ เพื่อที่จะใช้ในการพูดคุย เล่นเกมส์หรือตอบคำถาม
- การจัดเก็บข้อมูลของการเปิดเพลงหรือขอเพลงเพื่อนำมาจัดอันดับความนิยมของระบบงานปัจจุบันนั้นยังคงเป็นการบันทึกด้วยมือ ทำให้เสียเวลา ยุ่งยากต่อการสรุปอันดับยอดนิยมการจัดเก็บเป็นประวัติและการค้นหา
- นักจัดรายการวิทยุจะต้องทำงาน ณ ขณะนั้นตลอดเวลาไม่สามารถตั้งโปรแกรมการเล่นเพลงไว้ล่วงหน้าได้

- เนื่องจากการเปิดสล็อตโฆษณา หรือจึงเกิดเพลง นักจัดรายการวิทยุจะต้องเป็นคนทำเองทั้งหมด รวมถึงการตั้งตำแหน่งในการเล่นใหม่ในแต่ละสล็อตโฆษณา หรือจึงเกิด ทำให้เสียเวลาในการตั้งตำแหน่งและจัดวางแผ่นตามที่ต้องการ
- การจัดเก็บข้อมูลการเปิดเพลง ขอเพลง เพลงยอดนิยม และการจัดรายการเป็นเอกสารทั้งสิ้นซึ่งเมื่อเวลาผ่านไปเมื่อเอกสารมีจำนวนมากขึ้นจะทำให้ค้นหาข้อมูลที่ต้องการได้ลำบาก และเอกสารอาจมีการชำรุด หรือสูญหายได้
- ไม่สามารถรองรับการทำงานและความต้องการในการจัดเก็บ ค้นหา และเล่นเพลงได้อย่างรวดเร็วและทันเวลาเพียงพอ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การวิเคราะห์และออกแบบระบบใหม่

สำหรับเนื้อหาในบทนี้นั้นจะเป็นเรื่องที่ต้องเนื่องมาจากการศึกษาการดำเนินงานและปัญหาของระบบงานในปัจจุบัน ซึ่งในส่วนของทฤษฎีการวิเคราะห์ระบบนั้นจะเป็นการศึกษาเพื่อให้ทราบ ว่า Problem Domain ของระบบคืออะไร มีความต้องการในด้านใด เพื่อเป็นการทำความเข้าใจใน รายละเอียดของปัญหาต่าง ๆ เหล่านั้น ส่วนขั้นตอนของการออกแบบจะเป็นการสร้าง Model เพื่อที่จะแก้ปัญหาใน Problem Domain ที่ได้จากการวิเคราะห์ และเราจะนำเอาการออกแบบนี้ไปใช้เป็น แนวทางในการพัฒนาโปรแกรมต่อไป

#### 4.1 การวิเคราะห์ระบบ (Object-Oriented Analysis)

##### 4.1.1 ความต้องการของระบบ

จากการทำงานในรูปแบบเดิมดังที่กล่าวมาในบทที่ผ่านมาทำให้เกิดความต้องการในการพัฒนาระบบงานสารสนเทศข้อมูลเพลง เพื่อที่จะรองรับการทำงานในส่วนต่าง ๆ ที่มีความเกี่ยวข้องกับการจัดรายการวิทยุ โดยสามารถจัดหมวดหมู่ของงานที่จะดำเนินการได้ดังนี้

##### 4.1.1.1 การดำเนินรายการ

ในการดำเนินรายการวิทยุ งานหลักที่สำคัญสำหรับนักจัดรายการวิทยุคือจะต้องดูแล ควบคุมการเปิดเพลง จึงเกิด และสเปคโฆษณา ให้เหมาะสมตามแนวทางของแต่ละสถานีได้วางไว้ ซึ่งสำหรับระบบที่เราจะพัฒนาขึ้นมาจะช่วยให้นักจัดรายการวิทยุสามารถที่จะสร้างชุดเล่นเพลง อัตโนมัติ โดยไม่ต้องทำการควบคุมตลอดเวลา ซึ่งจะช่วยลดข้อผิดพลาดในการควบคุมการเปิด เพลง จึงเกิด และ สเปคโฆษณาได้

##### 4.1.1.2 การจัดเก็บข้อมูลที่เกี่ยวข้อง

ในเรื่องของการจัดเก็บข้อมูลที่เกี่ยวข้องนั้นเป็นส่วนที่ออกแบบเพิ่มเติมเพื่อที่จะให้นักจัด รายการวิทยุสามารถที่จะมีแหล่งข้อมูลเพื่อที่จะใช้ประกอบการจัดรายการได้อย่างมีประสิทธิภาพ ทั้งการจัดเก็บข้อมูลใหม่ ๆ, แก้ไขข้อมูลที่มีการเปลี่ยนแปลงและการค้นหาข้อมูล โดยข้อมูลต่าง ๆ ที่จัดเก็บนั้นสามารถแบ่งออกเป็น

- ข้อมูลเกี่ยวกับเพลง โดยจะเป็นการเก็บรายละเอียดเกี่ยวกับเพลงต่าง ๆ ในส่วนของ ศิลปินผู้ร้อง, เนื้อเพลง, ผู้แต่งคำร้อง, ผู้แต่งทำนอง

- ข้อมูลเกี่ยวกับศิลปิน โดยจะเป็นการเก็บรายละเอียดต่าง ๆ ของศิลปิน เช่นชื่อ ที่ติดต่อ เบอร์ติดต่อ สังกัด ผลงานที่ผ่านมา เป็นต้น
- ข้อมูลเกี่ยวกับผู้ร่วมเล่นเกมส์ เนื่องจากในบางครั้งรายการวิทยุจะมีการเปิดโอกาสให้ผู้ฟัง ได้โทรศัพท์เข้ามาร่วมเล่นเกมส์กับทางรายการ และมีการมอบของรางวัล ดังนั้นจะมีการจัดทำส่วนของการจัดเก็บข้อมูลของผู้ร่วมเล่นเกมส์เพื่อสะดวกในการตรวจสอบ การเล่นเกมส์ และการรับของรางวัลได้

#### 4.1.1.3 การจัดอันดับเพลง

ระบบจะมีการจัดเก็บรายการขอเพลงจากผู้ฟัง ไว้ในฐานข้อมูลและแสดงอันดับเพลงในแต่ละสัปดาห์ได้ เพื่อช่วยให้รูปแบบของการบันทึกข้อมูลเป็นมาตรฐานเดียวกัน และความถูกต้องแม่นยำมากขึ้น ช่วยลดเวลาในการจัดอันดับเพลง

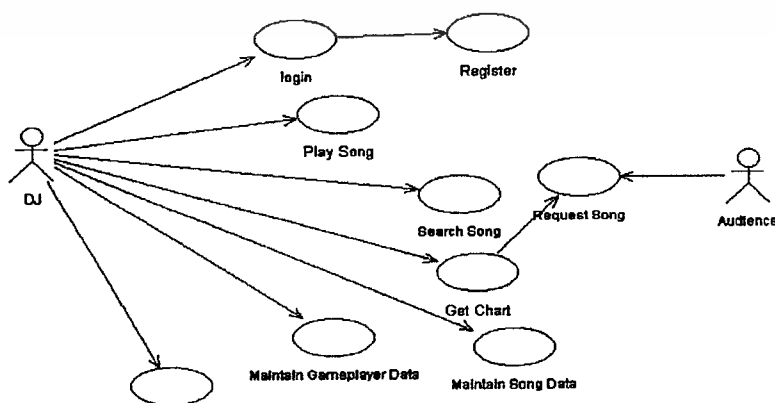
#### 4.1.1.4 การทำรายงาน

นอกจากงานหลัก ๆ ดังกล่าวข้างต้นนั้น ระบบจะต้องสามารถออกรายงานต่าง ๆ เช่น รายงานการขอเพลงในแต่ละวัน, รายการอันดับเพลงประจำสัปดาห์, รายงานผู้ได้รับของรางวัลในการเล่นเกมส์,

ซึ่งจากความต้องการต่าง ๆ ของระบบดังที่ได้กล่าวมานั้นเราจะนำไปใช้เป็นข้อมูลในการวิเคราะห์ระบบด้วยหลักการเชิงวัตถุในหัวข้อถัดไป

#### 4.1.2 Use Case Diagram

เราสามารถที่จะสร้างแบบจำลองของระบบสารสนเทศข้อมูลเพลงในรูปแบบของ Use Case Diagram โดยเราจะพิจารณาจากการทำงานของระบบว่ามีงานอะไรบ้าง แต่ละงานก็คือแต่ละ Use Case นั้นเอง ซึ่งจะได้ Use Case Diagram ดังแสดงในรูป 4-1



รูปที่ 4-1 Use Case Diagram ของระบบสารสนเทศข้อมูลเพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.3 Use Case Description

จาก Use Case Diagram เราสามารถที่จะทำการอธิบายรายละเอียด ขั้นตอนการทำงานหรือ ขั้นตอนการใช้งาน (Scenario) ของแต่ละ Use Case ได้ดังนี้

#### 4.1.3.1 Use Case Login

**Actors :** นักจัดรายการวิทยุ(Disk Jockey : DJ)

**Description :** DJ จะทำการเข้าสู่ระบบโดยการ Login

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ต้องการเรียกใช้งานระบบ จะต้องทำการ Login เข้าสู่ระบบและใส่ชื่อผู้ใช้และรหัสผ่าน
2. กดปุ่ม OK เพื่อยืนยันการเข้าระบบ
3. ถ้ารหัสผ่านถูกต้องตามที่ได้ Register ไว้ระบบก็จะอนุญาตให้ DJ เข้ามาใช้ระบบได้
4. ถ้ารหัสผ่านไม่ถูกต้องตามที่ได้ Register ไว้ระบบก็จะให้ทำการเข้าระบบใหม่
5. ถ้า DJ ยังไม่มีการ Register ไว้กับระบบ จะต้องทำการ Register ก่อน โดยเลือกปุ่ม Register

#### 4.1.3.2 Use Case Register

**Actors :** นักจัดรายการวิทยุ(Disk Jockey : DJ)

**Description :** DJ จะทำการลงทะเบียนเพื่อเข้าใช้ระบบ

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ต้องการทำการ Login เข้าสู่ระบบแต่ยังไม่ได้มีการ Register จะต้องทำการเลือก Register โดยกดที่ปุ่ม Register ระบบจะแสดงหน้าการลงทะเบียนนักจัดรายการ
2. DJ ทำการเพิ่มข้อมูลชื่อ, ชื่อผู้ใช้ระบบ, รหัสผ่าน, ช่วงเวลาที่จัด และกดปุ่ม OK เพื่อทำการบันทึกข้อมูลการลงทะเบียนในฐานข้อมูล
3. DJ สามารถเลือกทำการแก้ไขข้อมูลได้โดยการกดปุ่ม Update โดยจะต้องทำการยืนยันรหัสผ่านก่อนที่จะทำการแก้ไขข้อมูล

#### 4.1.3.3 Use Case Play Song

**Actors:** นักจัดรายการวิทยุ(Disk Jockey : DJ)

**Description:** DJ จะทำการควบคุมการเปิดเพลง สร้างรายการเล่นเพลง

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ทำการ Login เข้าสู่ระบบและทำการเลือกที่เมนูเล่นเพลง (Play Song)
2. ระบบจะแสดงหน้าจอเล่นเพลง
3. DJ สามารถเลือกเล่นเพลงที่เก็บอยู่ในหน่วยความจำของเครื่องคอมพิวเตอร์ โดยกดปุ่ม Add เพื่อเพิ่มเพลงเข้าในรายการเล่นเพลง (PlayList)
4. DJ สามารถเลือกสร้างชุดเล่นเพลงเพื่อเก็บไว้เล่นล่วงหน้าได้โดยการเพิ่มเพลงเข้าใน PlayList แล้วกดปุ่ม Save list จากนั้นทำการตั้งชื่อชุดเล่นเพลง นั้น ๆ เมื่อต้องการเล่นชุดเล่นเพลงก็เล่นก็เลือกปุ่ม Load List แล้วเลือกชุดเล่นเพลงที่ต้องการ

#### 4.1.3.4 Use Case Search Song

**Actors:** นักจัดรายการวิทยุ(Disk Jockey : DJ)

**Description:** DJ จะทำการค้นหาเพลงจากระบบ

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ทำการ Login เข้าสู่ระบบและทำการเลือกที่เมนูค้นหาเพลง (Search Song)
2. ระบบจะแสดงหน้าจอค้นหาเพลง
3. DJ สามารถเลือกค้นหาเพลงจากชื่อเพลงหรือชื่อศิลปิน จากนั้นก็ใส่ข้อมูลที่ต้องการค้นหา กดปุ่มค้นหา
4. ระบบจะทำการค้นหาข้อมูลของเพลงนั้น ๆ และแสดงรายละเอียดของเพลงบนหน้าจอการค้นหา
5. DJ สามารถเลือกการเปิดเพลงแบบอัตโนมัติ โดยเลือกที่เพลงที่ค้นหาพบตามต้องการ ระบบจะเล่นเพลงให้แบบอัตโนมัติ(สำหรับการขอเพลงแบบเปิดทันที)

#### 4.1.3.5 Use Case Request Song

**Actors:** นักจัดรายการวิทยุ(Disk Jockey : DJ), ผู้ฟังรายการวิทยุ (Audience)

**Description:** DJ จะทำการบันทึกการขอเพลงจากผู้ฟังเพื่อใช้เป็นข้อมูลในการจัดอันดับเพลง

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ทำการ Login เข้าสู่ระบบและทำการเลือกที่เมนูรายการขอเพลง (Request Song)
2. ระบบจะแสดงหน้าจอรายการขอเพลง
3. DJ สามารถรายชื่อเพลงที่ขอจากรายการเพลงที่มีอยู่ในระบบ (เนื่องจากทุกเพลงที่ขอจะต้องมีการเก็บอยู่ในระบบอยู่แล้ว)และกดปุ่มบันทึกเพื่อทำการบันทึกการขอเพลงลงในฐานข้อมูล
4. DJ สามารถเลือกทำการ Clear รายการขอเพลงเพื่อการบันทึกการขอเพลงครั้งใหม่ได้โดยการกดปุ่ม Clear
5. DJ สามารถเลือกการพิมพ์รายงาน โดยกดปุ่มพิมพ์เพื่อแสดงรายงานการขอเพลงประจำวันได้
6. DJ สามารถเลือกให้ระบบแสดงอันดับเพลงได้ โดยการเลือกปุ่มอันดับเพลง (Get Chart)

#### 4.1.3.6 Use Case Get Chart

**Actors :** นักจัดรายการวิทยุ(Disk Jockey : DJ)

**Description :** DJ จะทำการจัดอันดับเพลงยอดนิยมขึ้นมาจากรายการขอเพลงจากผู้ฟังรายการ

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ทำการ Login เข้าสู่ระบบและทำการเลือกที่เมนูรายการขอเพลงเลือกปุ่มจัดอันดับเพลง (Get Chart)
2. ระบบจะแสดงหน้าจออันดับเพลงยอดนิยม
3. DJ ทำการใส่อันดับเพลงที่ต้องการให้แสดงอันดับเพลง จากนั้นกดปุ่ม แสดงรายการ เพื่อให้ระบบแสดงรายการเพลงยอดนิยมที่มีผู้ขอมามากที่สุดตามที่กำหนดอันดับไว้ (โดยการเรียกใช้ข้อมูลจากรายการขอเพลง)
5. DJ สามารถเลือกทำการพิมพ์รายงานอันดับเพลงยอดนิยมได้โดยการกดปุ่ม พิมพ์
6. DJ สามารถเลือกทำการบันทึกผลการขอเพลงโดยการกดปุ่ม Save

#### 4.1.3.7 Use Case Maintain Song Data

**Actors :** นักจัดรายการวิทยุ (Dick Jockey :DJ)

**Description :** DJ ทำการจัดเก็บ,แก้ไข,ค้นหา ข้อมูลเพลงเช่น ชื่อเพลง, ชื่อศิลปิน, ความยาวเพลง, เนื้อเพลง, ผู้แต่งคำร้อง, ผู้แต่งทำนอง, ตำแหน่งเพลงในหน่วยความจำ

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ทำการ Login เข้าสู่ระบบและทำการเลือกที่เมนู ข้อมูลเพลง
2. ระบบแสดงหน้าจอข้อมูลเพลง
3. DJ ทำการใส่ข้อมูลของเพลงที่ต้องการบันทึกเข้าระบบ แล้วกดปุ่มบันทึกรายการ
4. DJ สามารถยกเลิกการทำรายการได้โดยการกดปุ่ม ยกเลิก
5. DJ สามารถทำการแก้ไขเปลี่ยนแปลงรายการได้โดยการกดปุ่ม แก้ไข
6. DJ สามารถทำการค้นหาข้อมูลเพลงโดยค้นหาจาก ชื่อเพลง, ชื่อศิลปิน ,เนื้อร้องบางส่วน โดยการใส่ข้อมูลที่ต้องการค้นหาในช่องข้อมูลที่ต้องการค้นหา แล้วกดปุ่ม ค้นหา

#### 4.1.3.8 Use Case Maintain Artist Data

**Actors :** นักจัดรายการวิทยุ (Dick Jockey :DJ)

**Description :** DJ ทำการจัดเก็บ,แก้ไข,ค้นหา ข้อมูลศิลปินเพลงเช่น ชื่อศิลปิน, สถานที่ติดต่อ, เบอร์ติดต่อ, สังกัด,ประวัติผลงาน

**Basic Flow :**

1. Use Case นี้เริ่มต้นเมื่อ DJ ทำการ Login เข้าสู่ระบบและทำการเลือกที่เมนู ข้อมูลศิลปิน
2. ระบบแสดงหน้าจอข้อมูลศิลปิน
3. DJ ทำการใส่ข้อมูลของศิลปินที่ต้องการบันทึกเข้าระบบ แล้วกดปุ่มบันทึกรายการ
4. DJ สามารถยกเลิกการทำรายการได้โดยการกดปุ่ม ยกเลิก
5. DJ สามารถทำการแก้ไขเปลี่ยนแปลงรายการได้โดยการกดปุ่ม แก้ไข
6. DJ สามารถทำการค้นหาข้อมูลศิลปินโดยค้นหาจาก ชื่อศิลปิน,สังกัด,ชื่อเพลง โดยการใส่ข้อมูลที่ต้องการค้นหาในช่องข้อมูลที่ต้องการค้นหา แล้วกดปุ่ม ค้นหา

#### 4.1.3.9 Use Case Maintain GamePlayer Data

**Actors :** นักจัดรายการวิทยุ (Dick Jockey :DJ)

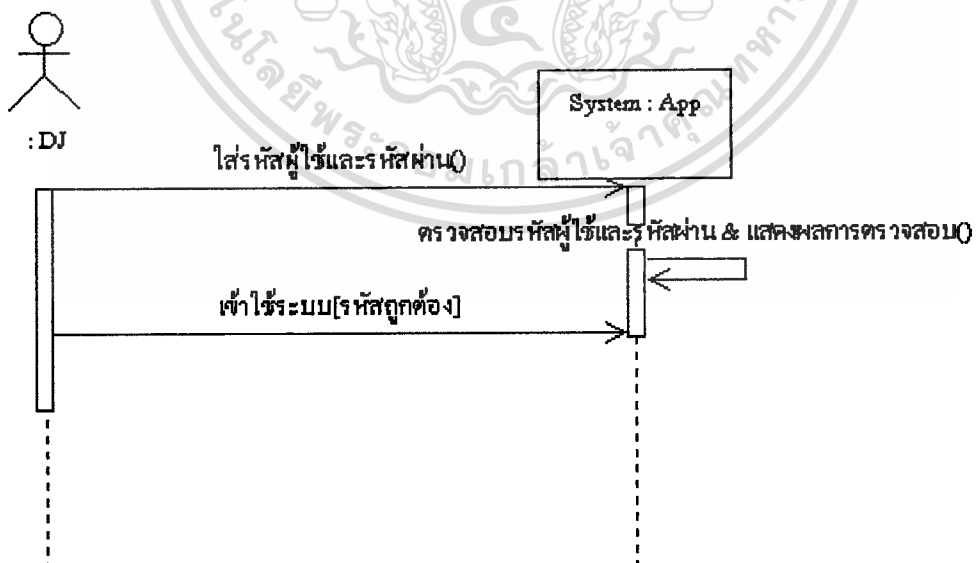
**Description :** DJ ทำการจัดเก็บ,แก้ไข,ค้นหา ข้อมูลผู้เล่นเกมส์เช่น ชื่อผู้เล่นเกมส์, สถานที่ติดต่อ, เบอร์ติดต่อ,บันทึกการรับของรางวัล

**Basic Flow :**

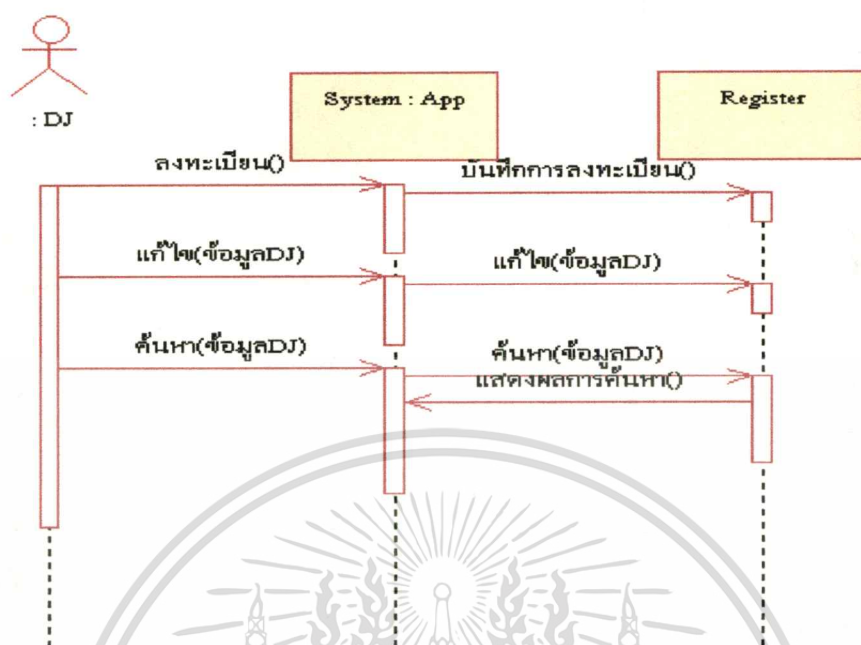
1. Use Case นี้เริ่มต้นเมื่อ DJ ทำการ Login เข้าสู่ระบบและทำการเลือกที่เมนู ข้อมูลผู้เล่นเกมส์
2. ระบบแสดงหน้าจอข้อมูลผู้เล่นเกมส์
3. DJ ทำการใส่ข้อมูลของผู้เล่นเกมส์ที่ต้องการบันทึกเข้าระบบ แล้วกดปุ่มบันทึกรายการ
4. DJ สามารถยกเลิกการทำรายการได้ โดยการกดปุ่ม ยกเลิก
5. DJ สามารถทำการแก้ไขเปลี่ยนแปลงรายการได้ โดยการกดปุ่ม แก้ไข
6. DJ สามารถทำการค้นหาข้อมูลผู้เล่นเกมส์ โดยค้นหาจาก ชื่อผู้เล่นเกมส์,วันที่เล่นเกมส์ โดยการใส่ข้อมูลที่ต้องการค้นหาในช่องข้อมูลที่ต้องการค้นหา แล้วกดปุ่ม ค้นหา

#### 4.1.4 Sequence Diagram

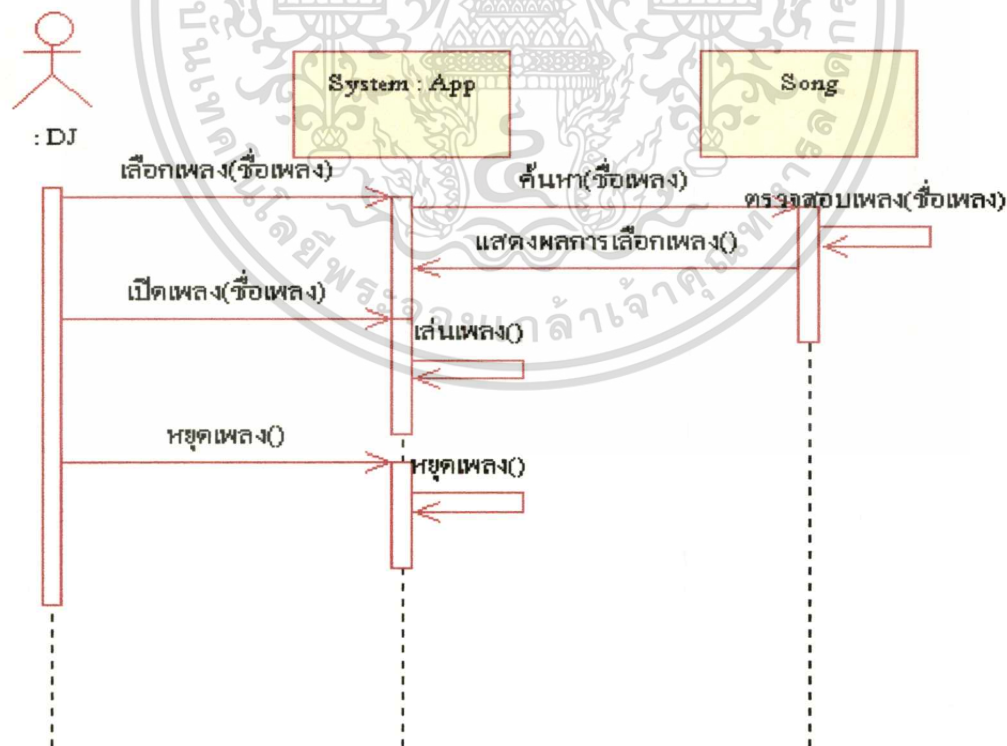
โดย Sequence Diagram ในขั้นตอนนี้จะเป็นการมองระบบเป็น Black Block และอธิบายว่า Actor จะทำอะไรกับระบบได้บ้าง ซึ่งจะสอดคล้องกับ Use Case Diagram ดังรูป 4-2 ถึง 4-10



รูปที่ 4-2 Sequence Diagram ของ Login

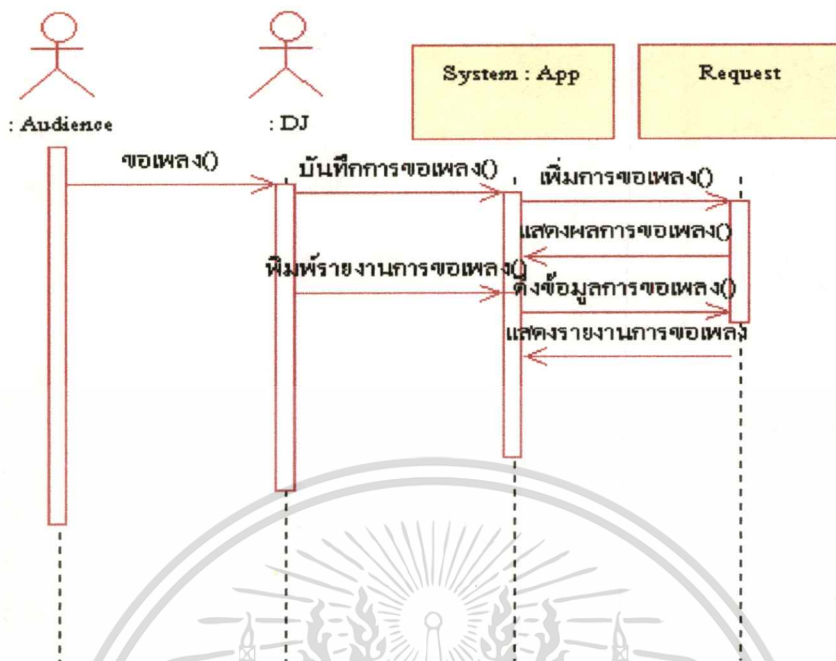


รูปที่ 4-3 Sequence Diagram ของ Register

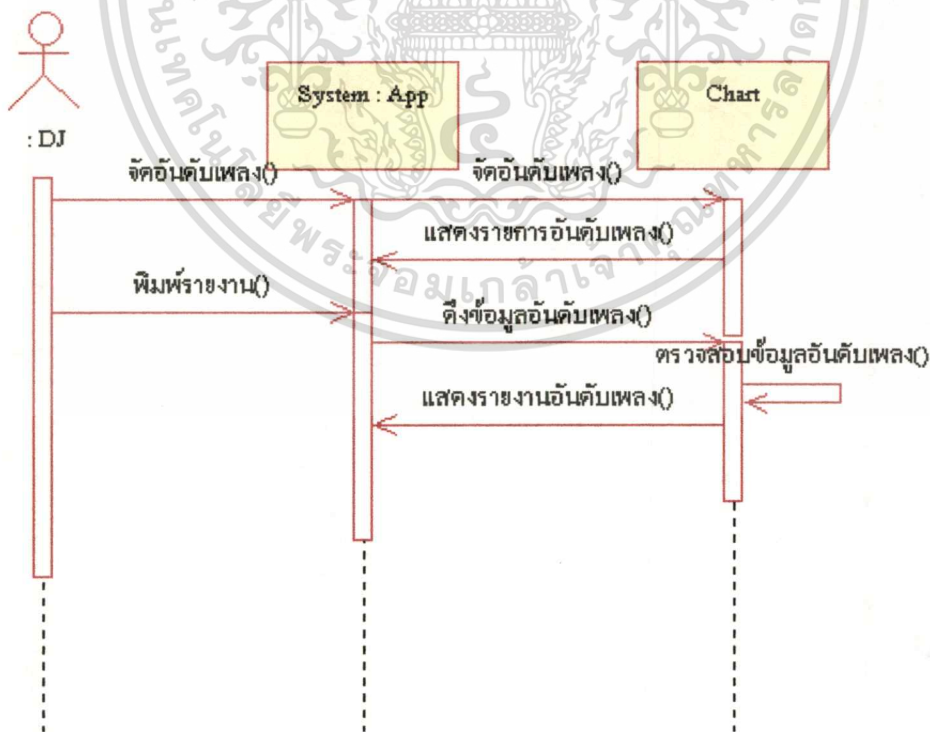


รูปที่ 4-4 Sequence Diagram ของ Play Song

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

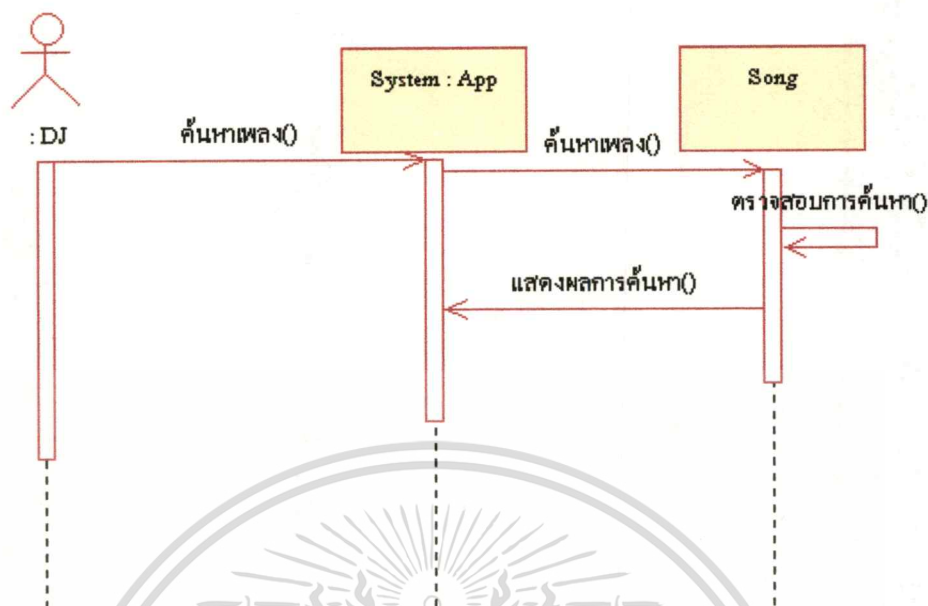


รูปที่ 4-5 Sequence Diagram ของ Request Song

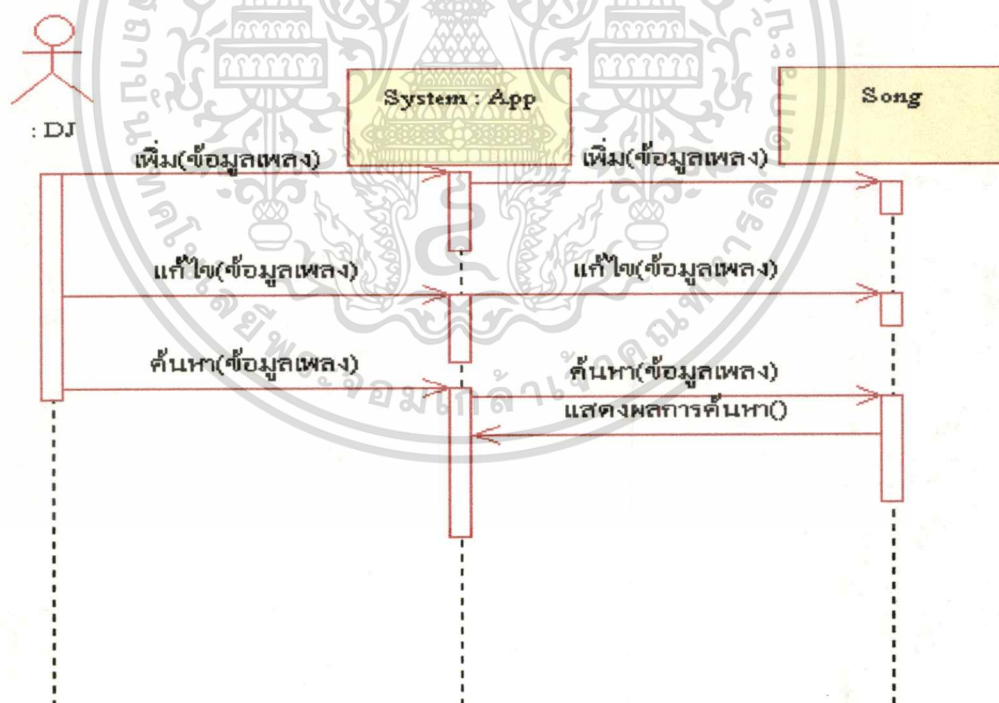


รูปที่ 4-6 Sequence Diagram ของ Get Chart

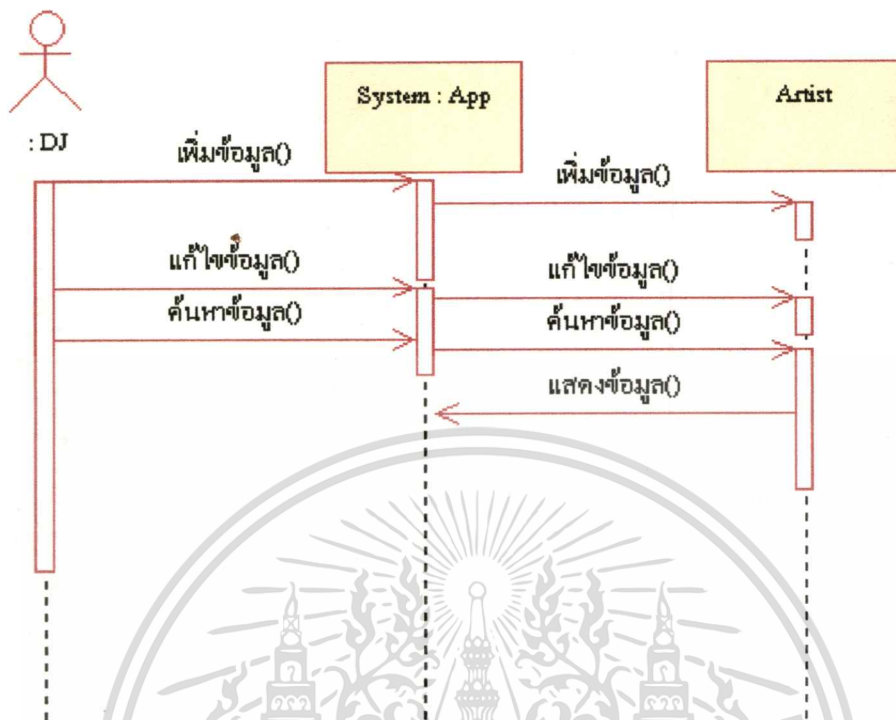
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



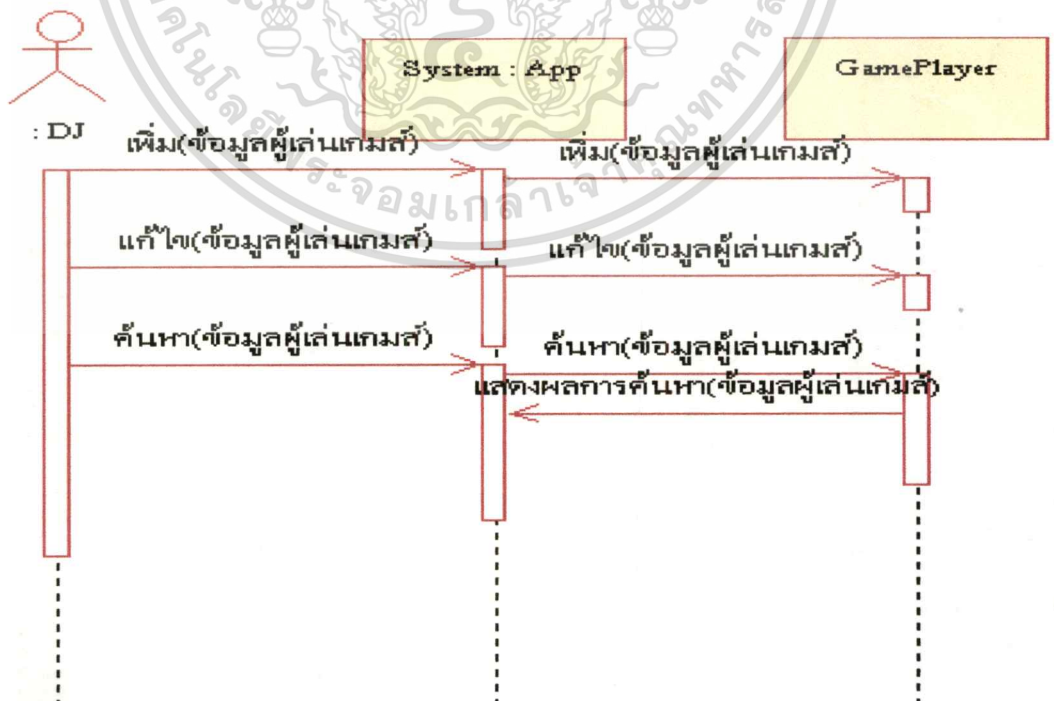
รูปที่ 4-7 Sequence Diagram ของ Search Song



รูปที่ 4-8 Sequence Diagram ของ Song Data



รูปที่ 4-9 Sequence Diagram ของ Artist Data

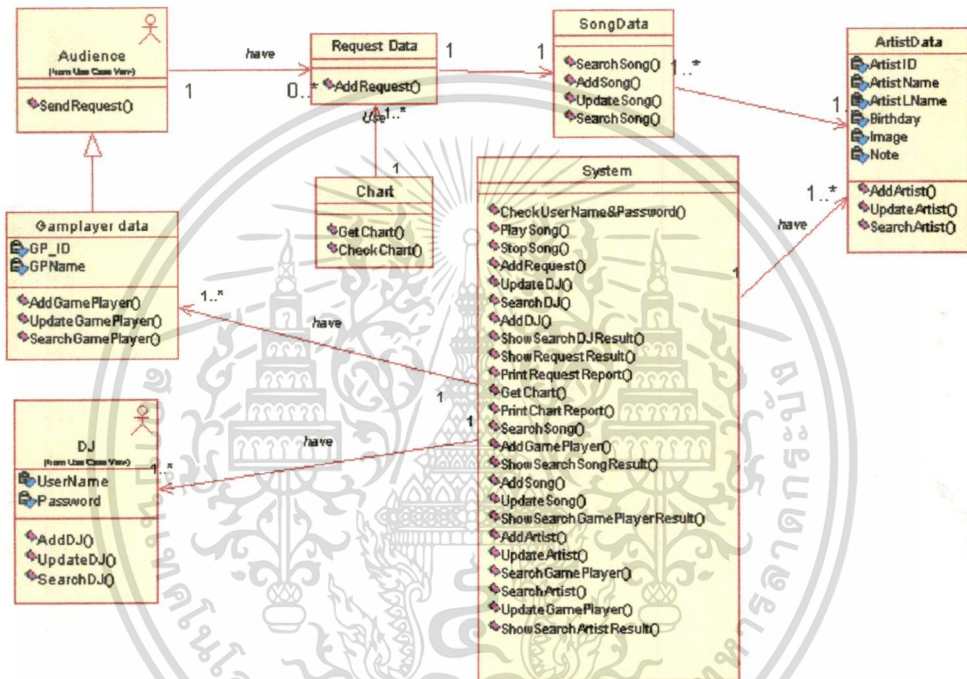


รูปที่ 4-10 Sequence Diagram ของ GamePlayer Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.5 Class Diagram

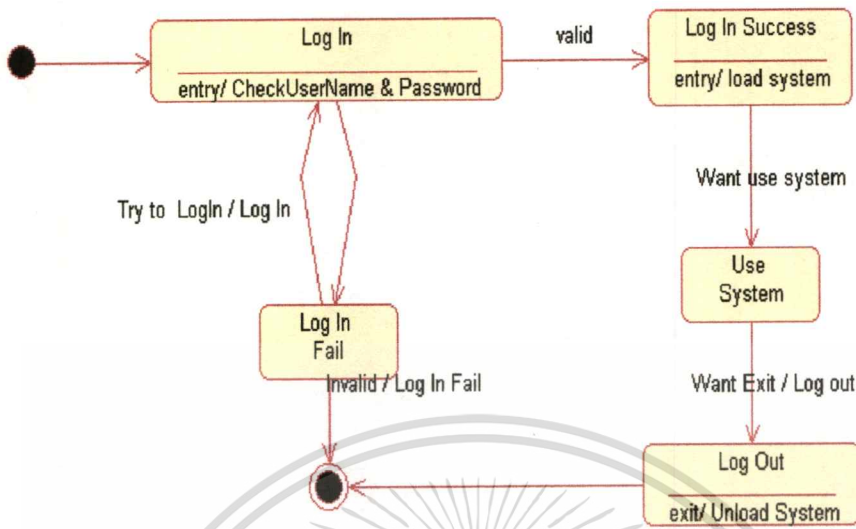
จากการวิเคราะห์เพื่อทำการหา Class ที่มีอยู่ใน Problem Domain ของระบบ โดยใช้หลักการของ Abstraction เพื่อบรรยายความสัมพันธ์ของ Class ต่าง ๆ จะทำให้ได้ Class Diagram พื้นฐานของระบบสารสนเทศข้อมูลเพลงดังรูป 4-11



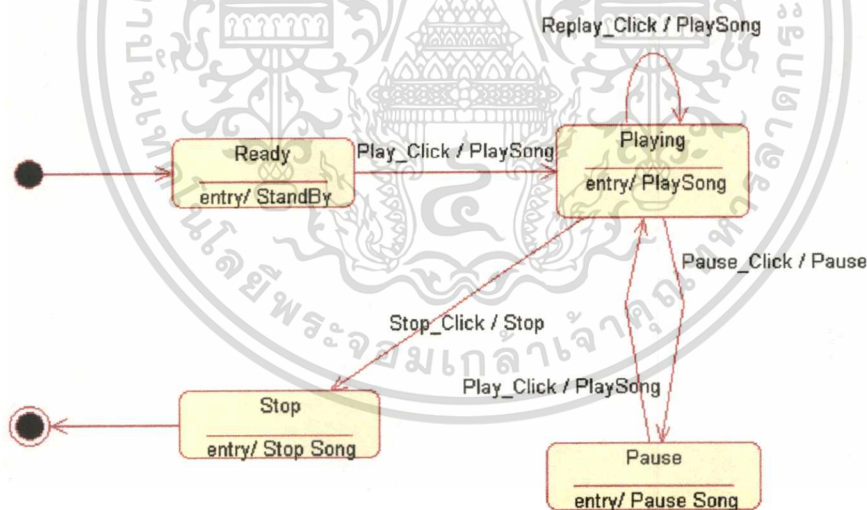
รูปที่ 4-11 Class Diagram พื้นฐานของระบบสารสนเทศข้อมูลเพลง

#### 4.1.6 State Diagram

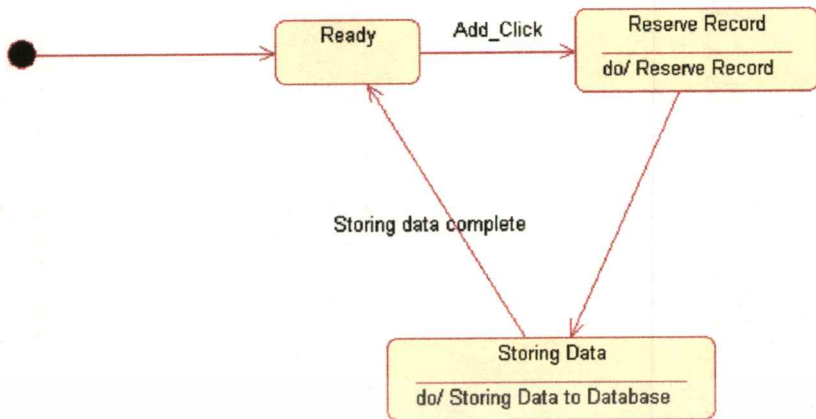
สำหรับ State Diagram ในส่วนของโปรแกรมจัดเก็บข้อมูลเพลงนี้จะแสดงในส่วนของ การเข้าใช้โปรแกรม (Login) ซึ่งจะช่วยให้มองเห็นถึงการเปลี่ยนแปลงสถานะในการเข้าใช้โปรแกรม, ในส่วนของการเล่นเพลง, ส่วนของการเพิ่มข้อมูล และส่วนของการจัดเตรียมเพลงเพื่อจัดเก็บ ซึ่งในส่วนของการจัดเตรียมเพลงนี้ เนื่องจากรูปแบบของเพลงโดยทั่วไปที่ใช้เปิดในสถานีวิทยจะเป็นลักษณะของไฟล์ .CDA ซึ่งหากจะนำมาจัดเก็บลงในเครื่องคอมพิวเตอร์จะทำให้กินเนื้อที่มากจึงต้องมีการแปลงไฟล์จาก .CDA ไปเป็นไฟล์ .MP3 ซึ่งจะช่วยให้ขนาดของไฟล์ให้เล็กลงได้อย่างมาก โดยจะทำการเปลี่ยนสถานะจาก .CDA เป็น .WAV โดยโปรแกรม Ripper และจาก .WAV ไปเป็น .MP3 โดยโปรแกรม Encoder เพื่อใช้ในการจัดเก็บต่อไป ซึ่ง State Diagrams ต่าง ๆ นี้แสดงดังรูป 4-12 ถึง 4-15



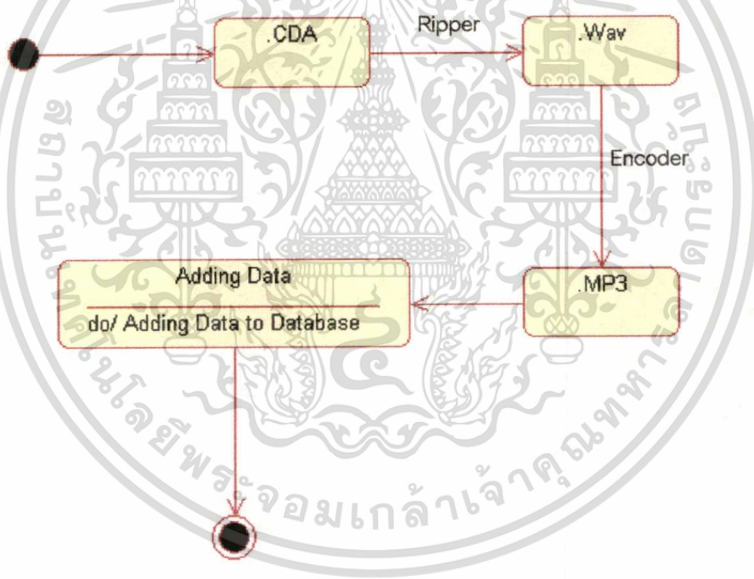
รูปที่ 4-12 State Diagram ของการเข้าใช้งานโปรแกรม



รูปที่ 4-13 State Diagram ของการเล่นเพลง



รูปที่ 4-14 State Diagram ของการเพิ่มข้อมูลลงในฐานข้อมูล



รูปที่ 4-15 State Diagram ของการจัดเตรียมเพลงเพื่อการจัดเก็บ

## 4.2 การออกแบบระบบ (Object-Oriented Design)

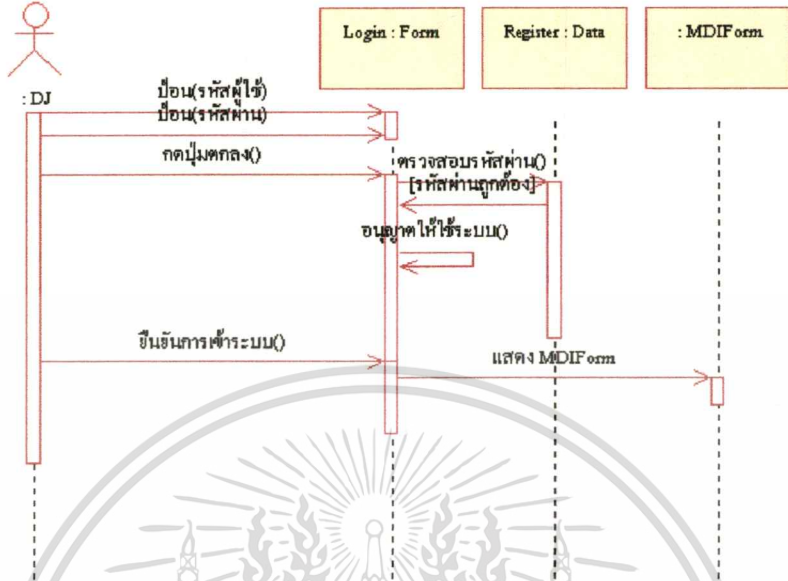
ในขั้นตอนของการออกแบบระบบนี้จะประกอบไปด้วย 4 ขั้นตอนคือ

- การปรับปรุง Diagrams ต่าง ๆ ที่ได้จาก Object-Oriented Analysis ให้ดีขึ้น มีประสิทธิภาพมากขึ้น เรียกขั้นตอนนี้ว่าการทำ Refinement
- การออกแบบโครงสร้างของ Software หรือ Application Architecture Design ซึ่งใน UML จะนำเสนอในรูปแบบของ Component Diagram
- การออกแบบโครงสร้างของ Hardware หรือ System Architecture Design ซึ่งใน UML จะนำเสนอในรูปแบบของ Deployment Diagram
- การออกแบบฐานข้อมูลหรือ Persistent Data Design หรือ Conceptual Database Design ซึ่งในส่วนของ Object-Oriented Design ยังไม่มี Diagram ใดที่ใช้เพื่อแสดงส่วนนี้โดยเฉพาะ แต่ก็มีหลักการในการที่จะแปลง Class Diagram ให้เป็น E-R Diagram และสร้าง Table ใน Relation Database ดังจะ ได้กล่าวถึงต่อไป

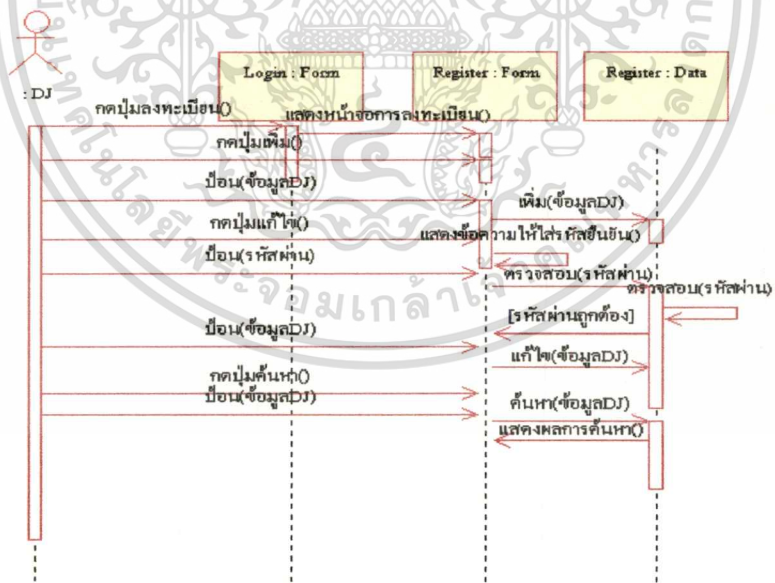
### 4.2.1 การ refinement

#### 4.2.1.1 Sequence Diagram ในส่วนของการออกแบบ

เป็นการพิจารณา Sequence Diagram จากส่วนของการวิเคราะห์แล้วนำมาใช้ออกแบบระบบให้มีความสมบูรณ์มากยิ่งขึ้นเช่นมีการออกแบบให้มีหน้าจอของระบบขึ้นมาเพื่อรองรับการทำงานในส่วนต่าง ๆ ซึ่งจะทำให้เกิด Class ของระบบเพิ่มขึ้นมาอีกด้วย ซึ่งหากสังเกตในส่วนของ Message ที่เกิดขึ้นในแต่ละขั้นตอนมันก็คือการเรียกใช้ Operation ของ Class นั้น ๆ นั่นเอง เช่น จากรูป 4-16 นี้ Message ป้อน(รหัสผู้ใช้) กับ ป้อน(รหัสผ่าน) เป็นการส่งค่ารหัสผู้ใช้งานกับรหัสผ่าน ไปยัง Login Form , ส่วนใน Message ตรวจสอบรหัสผ่าน() คือการที่ Login Form ไปเรียกใช้ Operation ตรวจสอบรหัส() จาก Register Data ซึ่งเป็นส่วนที่เก็บข้อมูลการลงทะเบียนของนักจัดรายการวิทยุ เป็นต้น Sequence Diagram ที่ได้ทำการปรับปรุงเพิ่มเติมแสดงดังรูป 4-16 ถึง 4-24

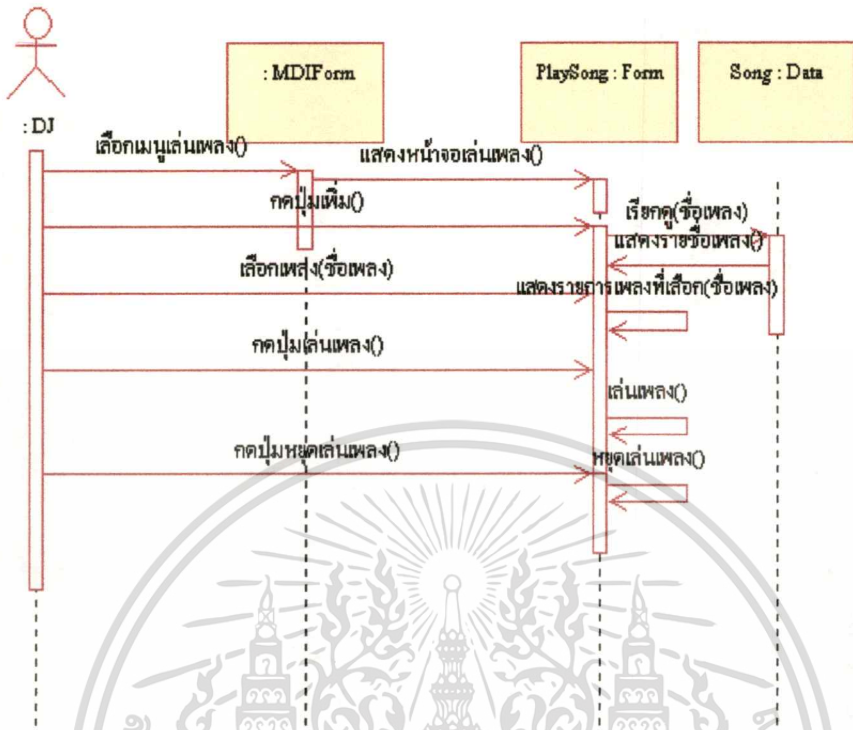


รูปที่ 4-16 Sequence Diagram ของ Login (ส่วนของการ Design)

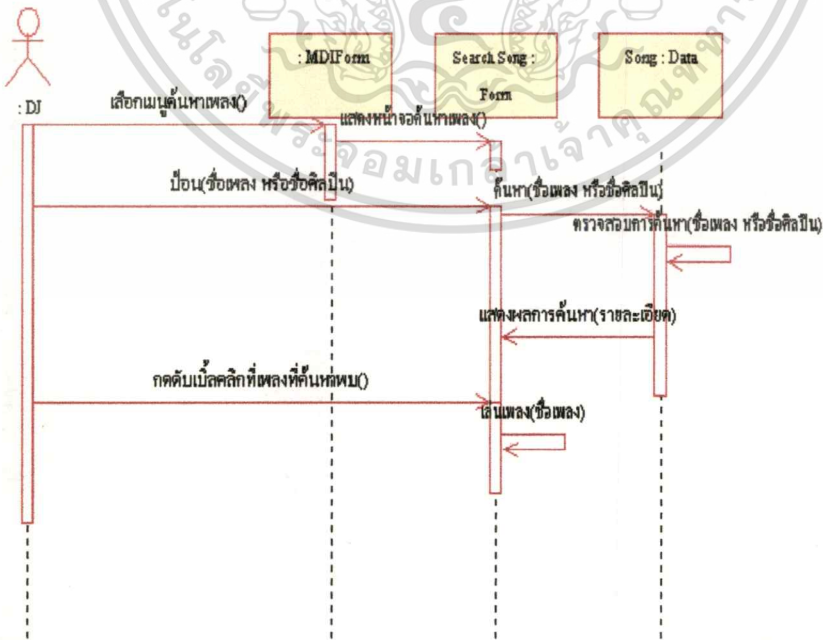


รูปที่ 4-17 Sequence Diagram ของ Register (ส่วนของการ Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

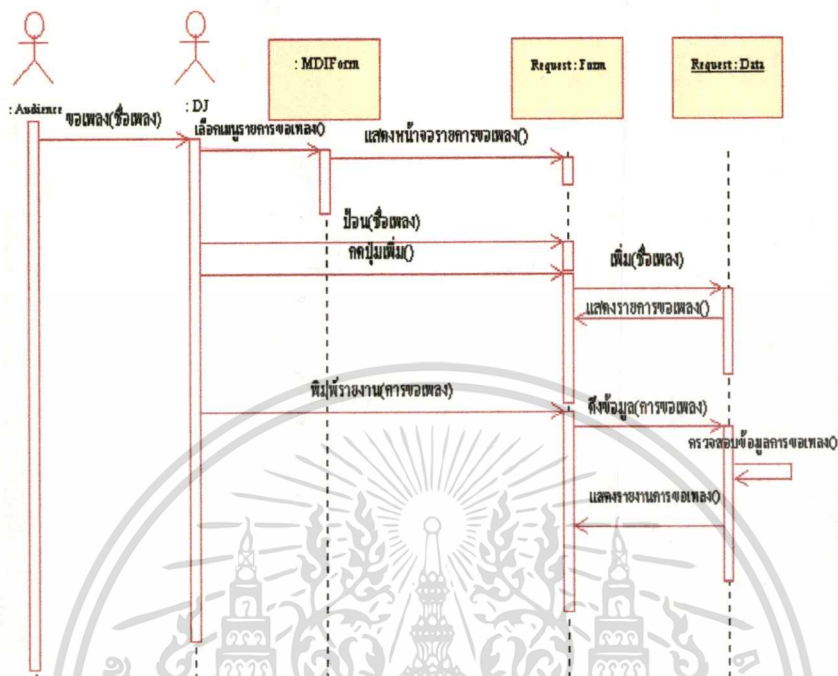


รูปที่ 4-18 Sequence Diagram ของ Play Song (ส่วนของการ Design)

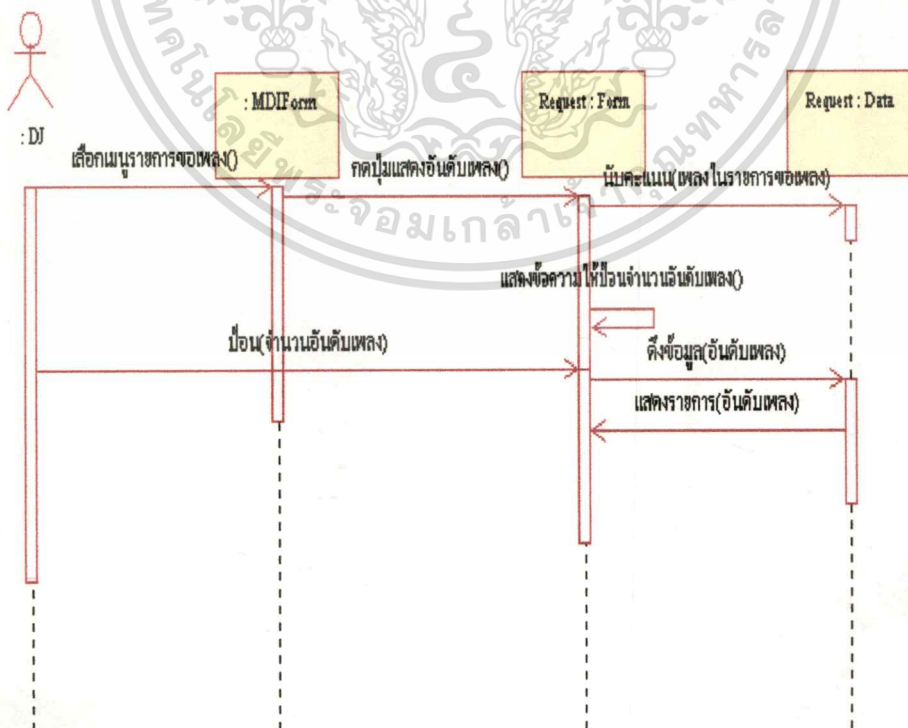


รูปที่ 4-19 Sequence Diagram ของ Search Song (ส่วนของการ Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

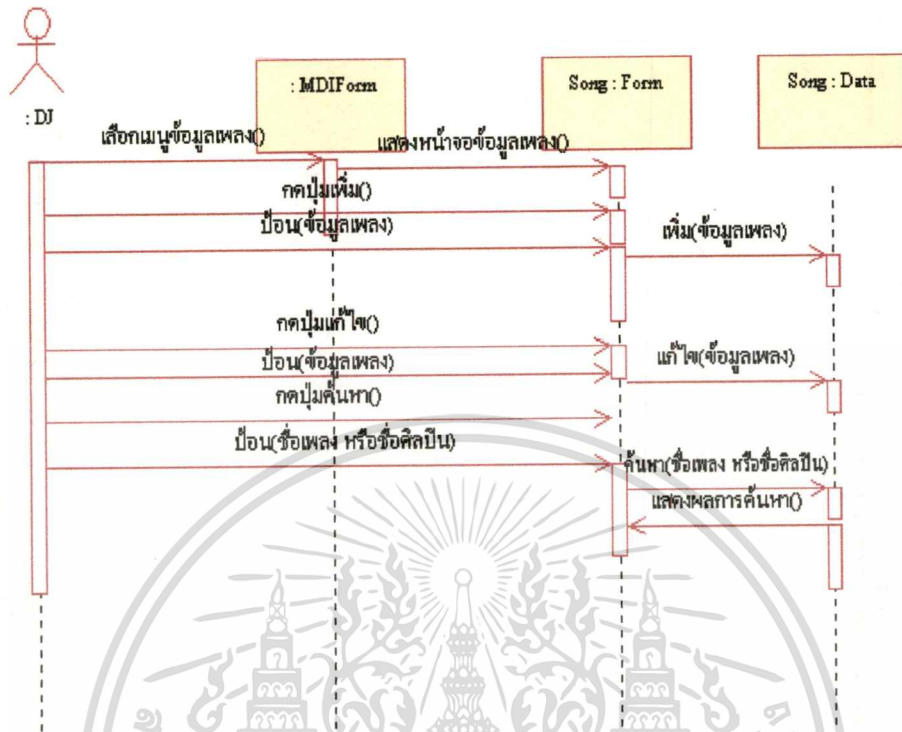


รูปที่ 4-20 Sequence Diagram ของ Request (ส่วนของการ Design)

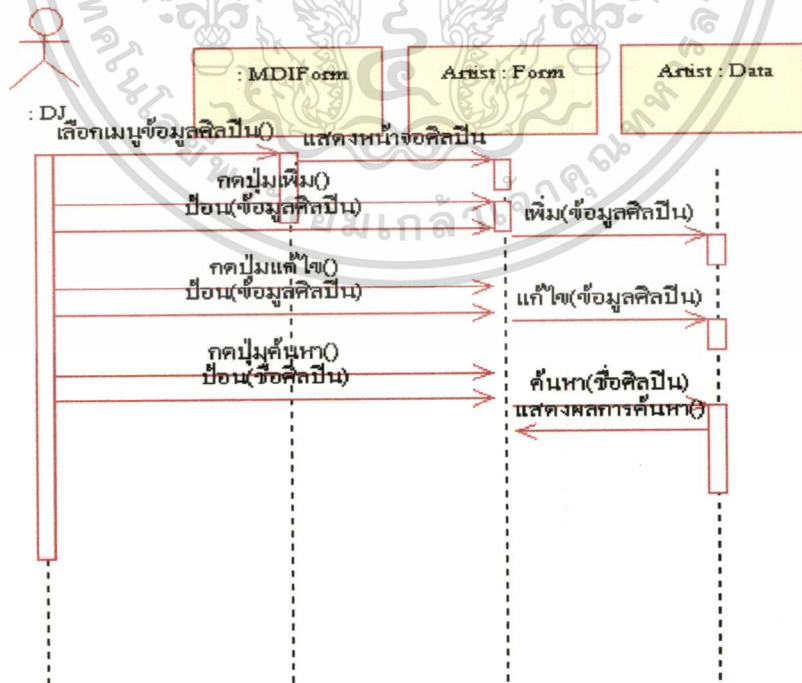


รูปที่ 4-21 Sequence Diagram ของ GetChart (ส่วนของการ Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

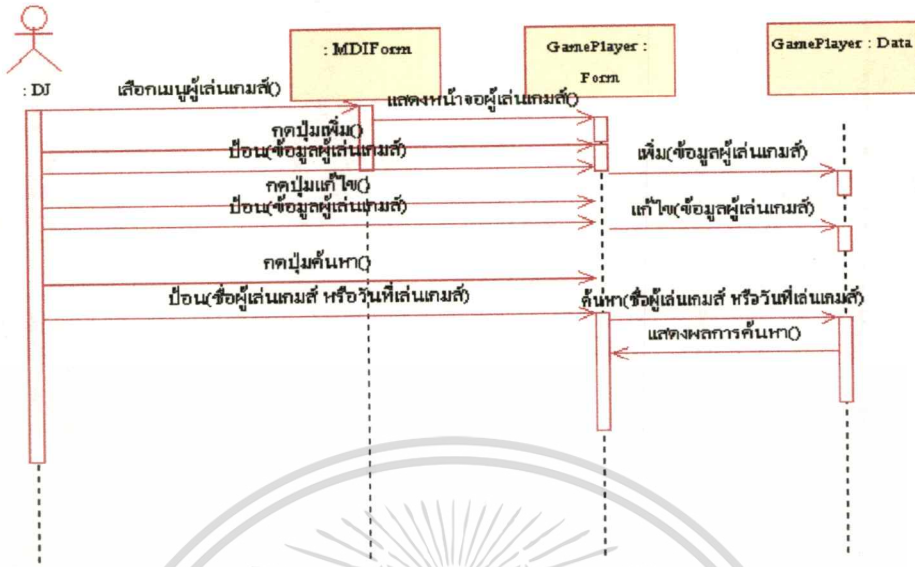


รูปที่ 4-22 Sequence Diagram ของ Song Data (ส่วนของการ Design)



รูปที่ 4-23 Sequence Diagram ของ Artist Data (ส่วนของการ Design)

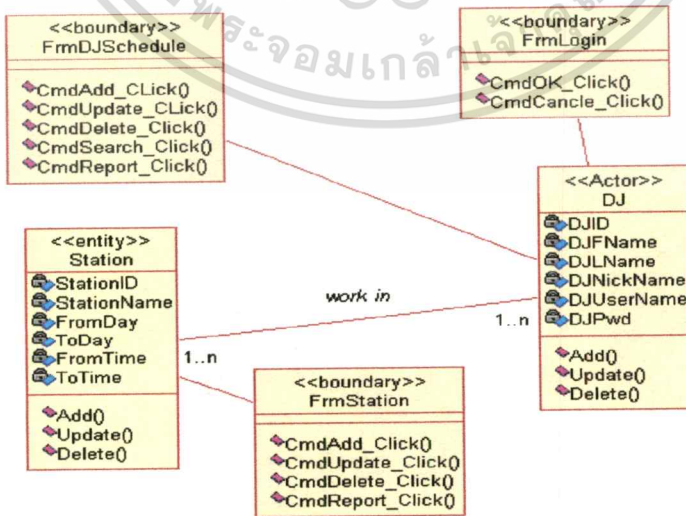
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-24 Sequence Diagram ของ GamePlayer Data (ส่วนของการ Design)

4.2.1.2 การ Refinement Class Diagram

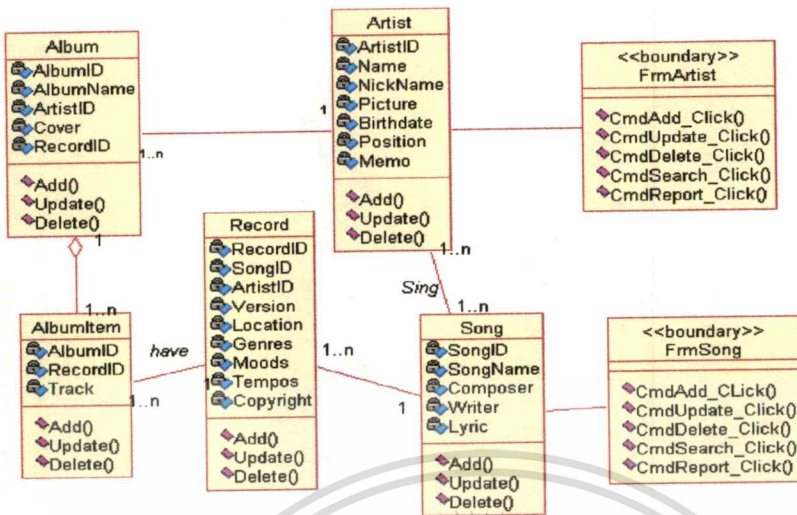
จากการทำ Refinement ในส่วนของ Sequence Diagram จะพบว่าเราจะมีส่วนของหน้าจอต่าง ๆ เพิ่มเข้ามาเพื่อที่จะรองรับการทำงานในส่วนต่าง ๆ และในส่วนของการ Refinement Class Diagram นี้เราก็จะทำการสร้าง Class Diagram ที่มีความสมบูรณ์ยิ่งขึ้นโดยแสดงส่วนของ Attribute และ Operations ของ Class ให้ครบถ้วน รวมถึงการสร้าง Class ของ Interface ที่เพิ่มเข้ามาจากขั้นตอนการ Refinement Sequence Diagram ด้วย



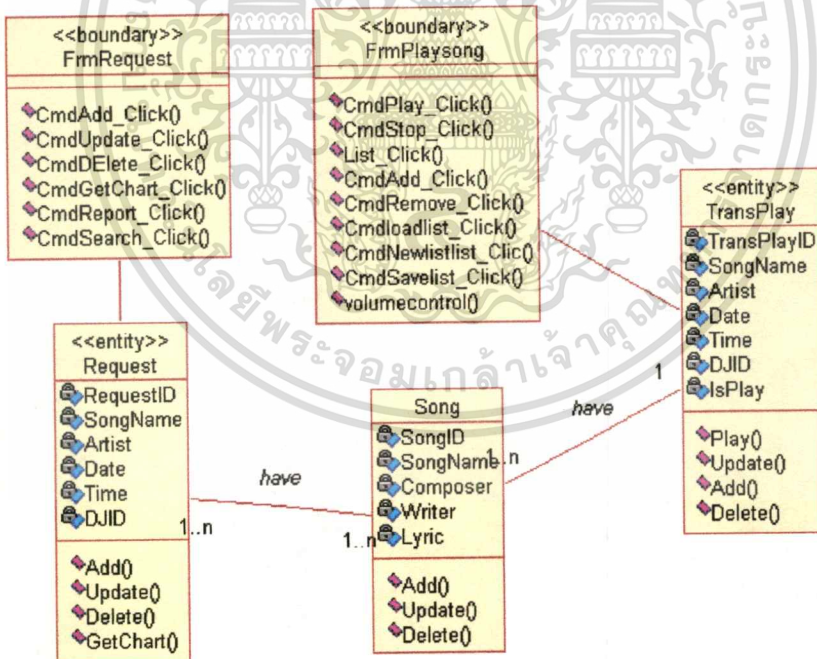
รูปที่ 4-25 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FrmDJSchedule

และ Class FrmStation กับ Class อื่น ๆ

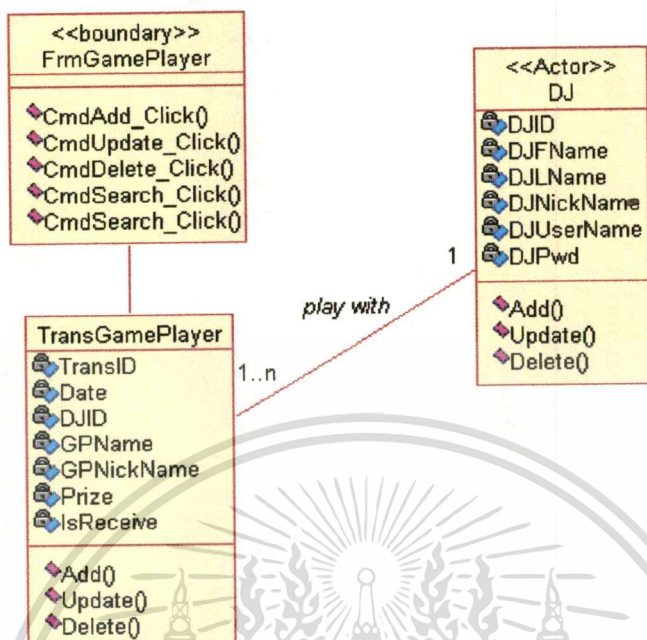
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนเพื่อการค้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



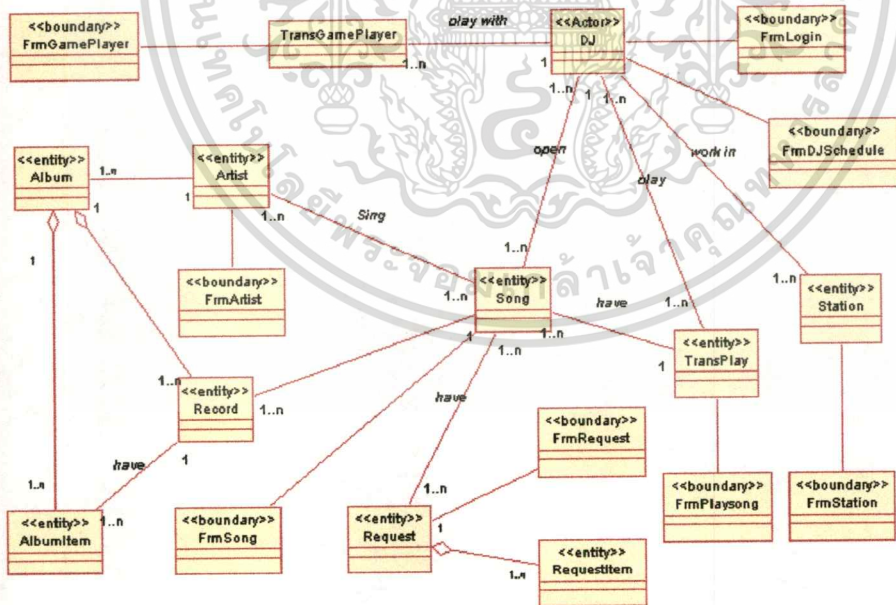
รูปที่ 4-26 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FmArtist และ Class FmSong กับ Class อื่น ๆ



รูปที่ 4-27 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FmRequest และ Class FmPlaySong กับ Class อื่น ๆ



รูปที่ 4-28 Class Diagram (Interface Class) แสดงความสัมพันธ์ของ Class FrmGamePlayer กับ Class อื่น ๆ

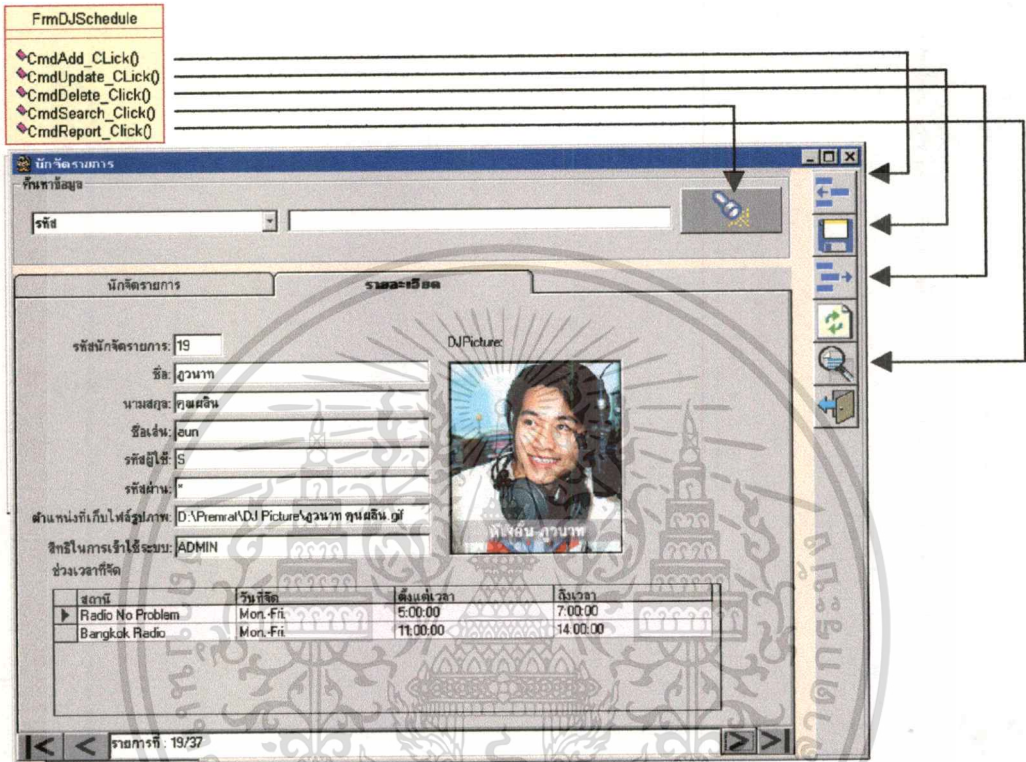


รูปที่ 4-29 Class Diagram แสดงความสัมพันธ์ระหว่าง Class ต่าง ๆ ทั้งหมดในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

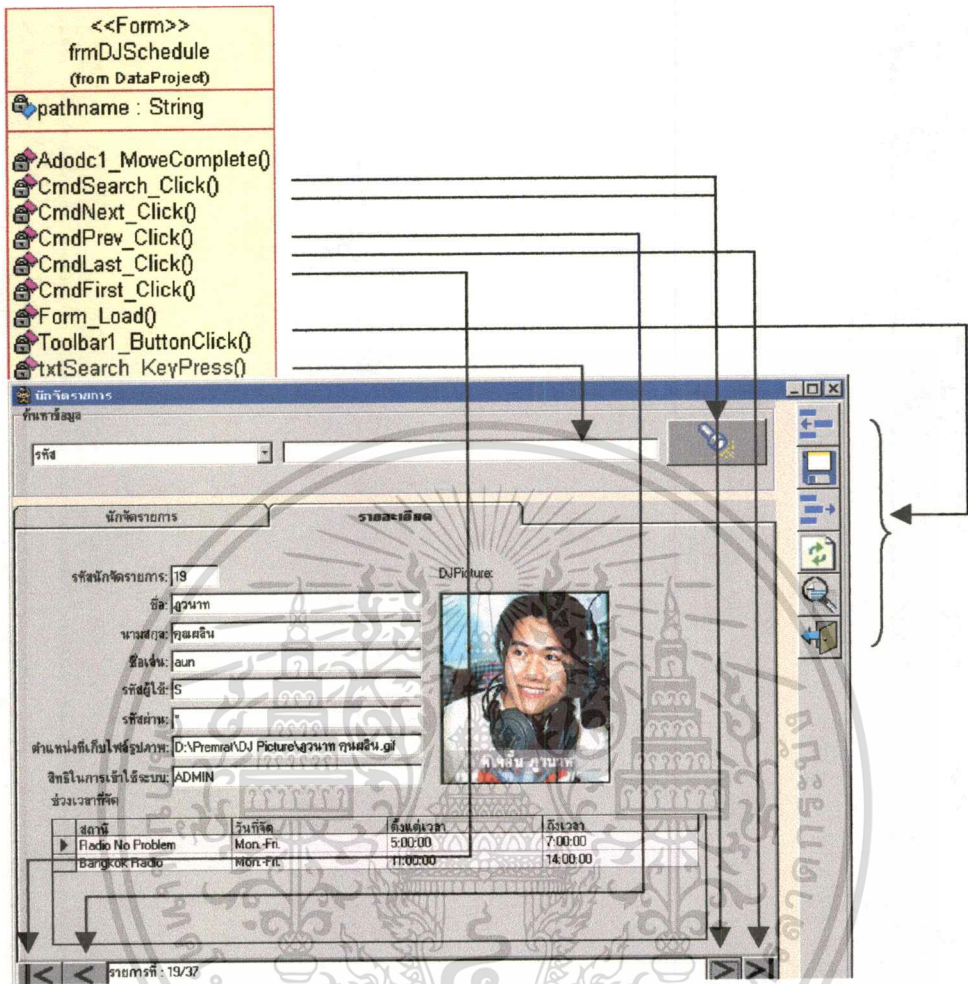
### 4.2.1.3 การออกแบบหน้าจอของระบบ

ในส่วนของการออกแบบหน้าจอที่ใช้ในระบบนั้นจะอ้างอิงส่วนของ Class ที่เป็น interface ต่าง ๆ ดังรูป



รูปที่ 4-30 ตัวอย่างการออกแบบหน้าจอของระบบที่สัมพันธ์กับ Class

จากรูปที่ 4-30 จะพบว่าในส่วนของ Operation ของ Class จะเป็นส่วนที่นำมาใช้เป็นการดำเนินการที่เกิดขึ้นในหน้านั้น ๆ เช่น CmdAdd\_Click() คือการกระทำที่เกิดขึ้นเมื่อมีการกดปุ่ม Add ที่หน้านี้ ในทางกลับกัน เราสามารถที่จะสร้างหน้าจอจาก Visual Basic เพื่อใช้ในการแสดงตัวอย่างหน้าจอให้แก่ User เพื่อที่จะให้ User พิจารณาว่าตรงตามที่ต้องการหรือไม่ เมื่อออกแบบหน้าจอเสร็จแล้วจึงเลือกใช้โปรแกรม Rational Rose ที่เป็น Add-Ins ใน Visual Basic ทำการ reverse Engineering กลับมาเป็น Class ของหน้านั้น ๆ ก็ได้ โดยจากตัวอย่างเดียวกันนี้ ถ้าใช้วิธีการ reverse Engineering จะได้ Class ดังรูป 4-31

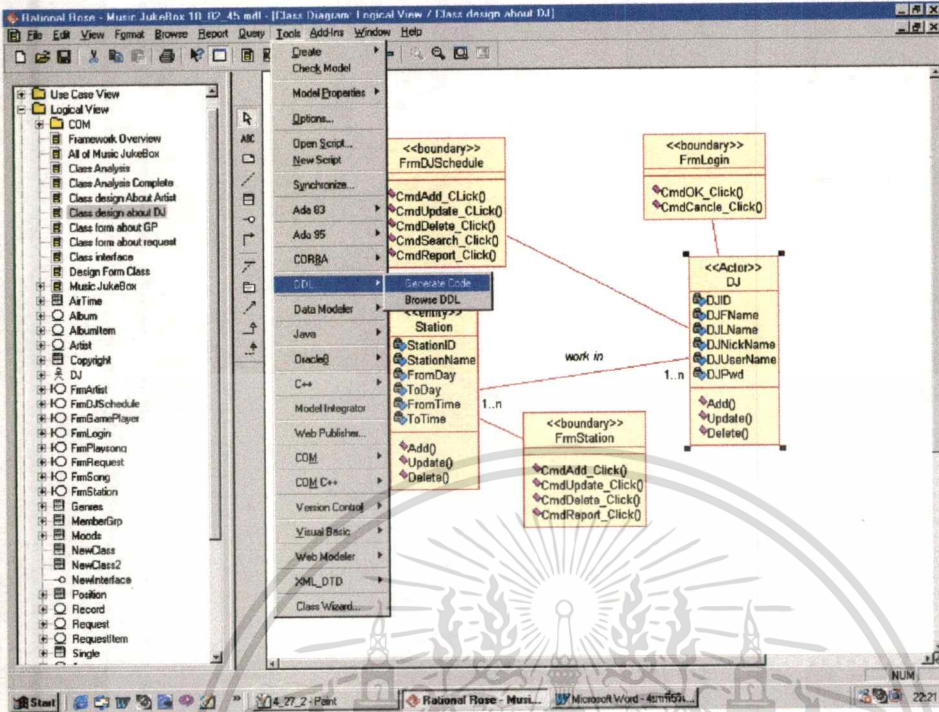


รูปที่ 4-31 Class DJSchedule ที่สร้างขึ้นจากวิธีการ Reverse Engineering และความสัมพันธ์ของ ClassDJSchedule กับ FrmDJSchedule

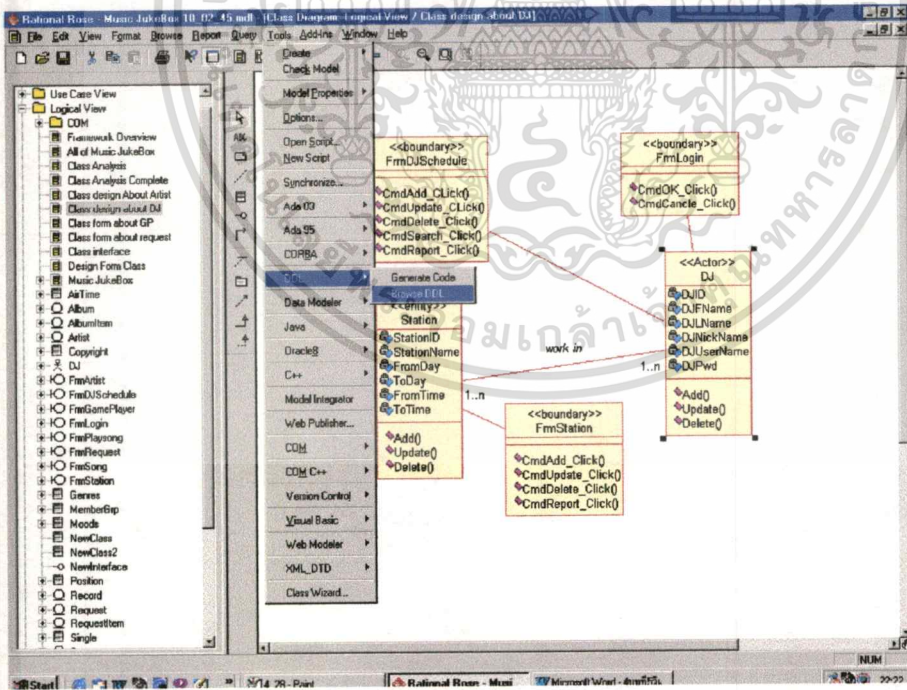
#### 4.2.2 การแปลง Class Diagram เป็น Database

จาก Class Diagram ที่สร้างขึ้นเราสามารถที่จะนำไปใช้ในการแปลงเป็นตารางในฐานข้อมูลได้โดยใช้เครื่องมือในการสร้าง Database Schemas ของ Rational Rose ซึ่งในที่นี้เลือกการสร้างในลักษณะของ DDL (Data Definition Language) โดยมีวิธีการดังนี้

- เลือก Class ที่ต้องการสร้างแล้วไปที่เมนู Tool→DDL→Generate Code ดังรูปที่ 4-32 ก)
- จากนั้นเลือกเมนู Tool →Browse DDL ดังรูปที่ 4-32 ข)
- จะปรากฏหน้าจอของ Schemas ที่ถูกสร้างขึ้นมาดังรูปที่ 4-32 ค)



รูปที่ 4-32 ก) การ Generate Code จาก Class ที่สร้างขึ้นมา



รูปที่ 4-32 ข) การ Browse DDL เพื่อดู Schemas ที่สร้างขึ้นจาก Class ที่เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Untitled - Notepad
File Edit Search Help
CREATE TABLE DJ
(DJID          CHAR(10)  NOT NULL,
DJFNAME       CHAR(50)  NOT NULL,
DJLNAME       CHAR(50)  NOT NULL,
DJNICKNAME    CHAR(50)  NOT NULL,
DJUSERNAME    CHAR(50)  NOT NULL,
DJPWD         CHAR(50)  NOT NULL,
PRIMARY KEY = DJID)

CREATE TABLE STATION
(STATIONID     CHAR(10)  NOT NULL,
STATIONNAME   CHAR(50)  NOT NULL,
FROMDAY       DATE(8)   NOT NULL,
TODAY         DATE(8)   NOT NULL,
FROMTIME      TIME(8)   NOT NULL,
TOTIME        TIME(8)   NOT NULL,
PRIMARY KEY = STATIONID)

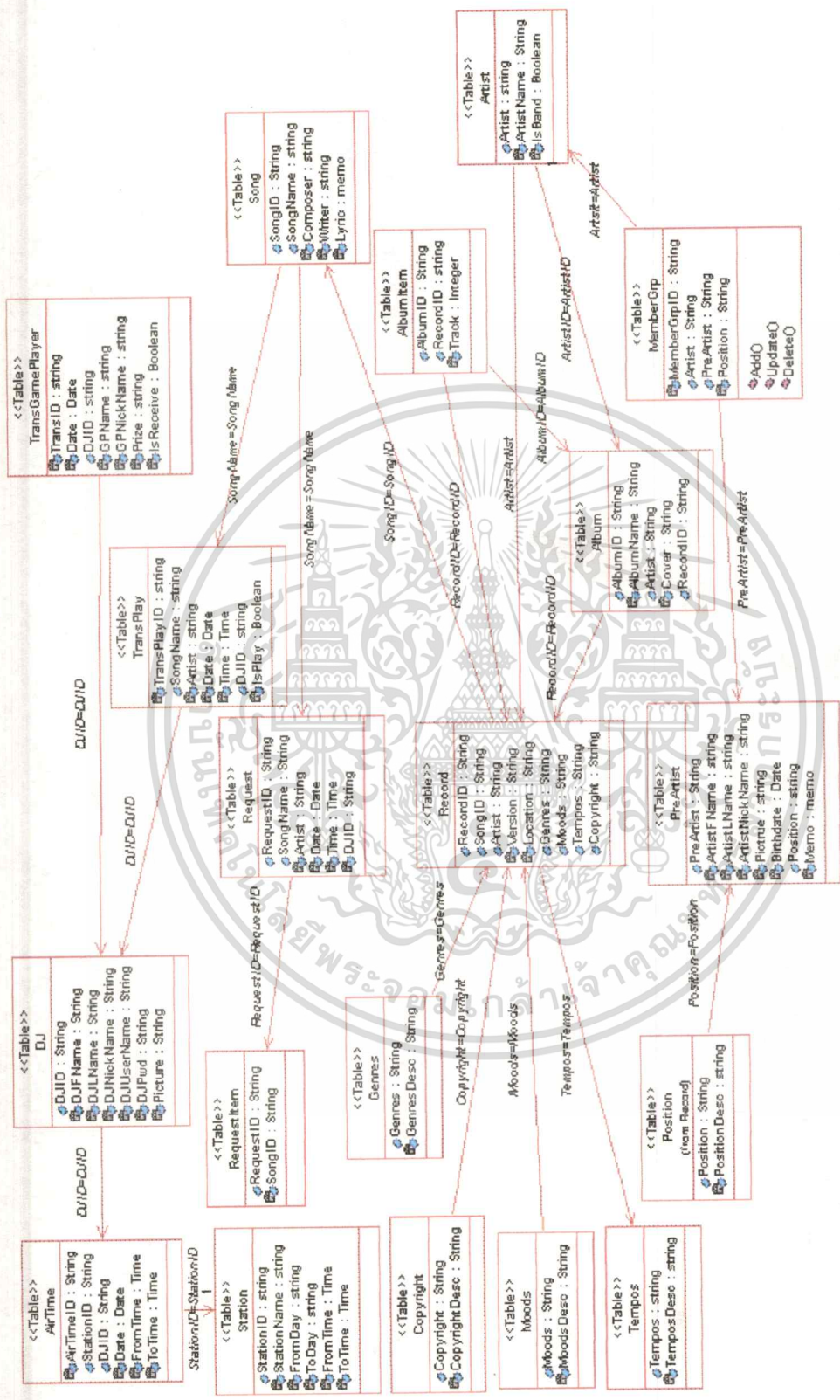
CREATE TABLE DJ_STATION
(DJID          CHAR(10),
STATIONID     CHAR(10),
PRIMARY KEY = (DJID,STATIONID)
FOREIGN KEY = DJID REFERENCES DJ,
FOREIGN KEY = STATIONID REFERENCES STATION)

```

รูปที่ 4-32 ค) หน้าจอแสดงผลของการสร้าง Schemas จาก Class ที่เลือก

ซึ่งเราสามารถนำเอา Code ที่ได้นี้ไปใช้ในการสร้างตารางในฐานข้อมูลของเราต่อไปได้ ดัง

รูปที่ 4-33 เป็นการแสดง โครงสร้างของฐานข้อมูลของโปรแกรมจัดเก็บข้อมูลเพลง ที่ได้สร้างขึ้น โดยอ้างอิงจาก Database Schemas ที่ได้จากการ Generate DDL Code โดยนำเสนอในรูปแบบของ Class Diagram ซึ่งจะนำเอาโครงสร้างดังกล่าวนี้ไปใช้ในการสร้างฐานข้อมูลในลักษณะของ ๓ Relation Database ต่อไป



รูปที่ 4-33 โครงสร้างของฐานข้อมูลของโปรแกรมจัดเก็บข้อมูลเพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### การพัฒนาระบบและผลการทำงาน

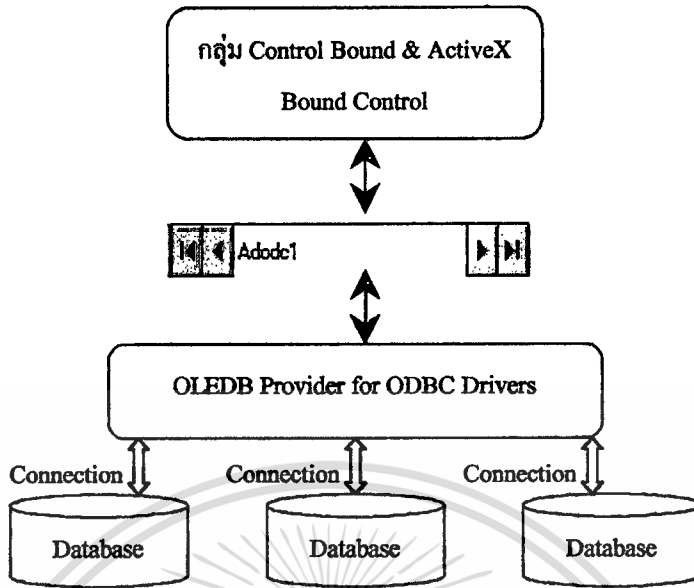
ในบทนี้จะกล่าวถึงเรื่องของฮาร์ดแวร์และซอฟต์แวร์ที่นำมาใช้ในการพัฒนาระบบ รวมถึงแนวทางในการพัฒนาระบบและผลของการทำงานของระบบที่ได้พัฒนาขึ้นมาดังมีรายละเอียดต่อไปนี้

#### 5.1 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

- ฮาร์ดแวร์
  - หน่วยประมวลผลกลาง เพนเทียม 4 1.7 GHz (CPU Pentium 4 1.7 GHz)
  - หน่วยความจำหลัก 100 MB (Main Memory 100 MB)
  - ฮาร์ดดิสก์ 10 GB (Harddisk 10 GB)
  - CD-ROM
- ซอฟต์แวร์
  - ระบบปฏิบัติการ windows 98
  - Ms Access 97
  - Visual Basic Version 6.0
  - Audio Extractor Version 1.0 (ใช้แปลงไฟล์ .CDA เป็น .WAV)
  - CDMaster32 (ใช้แปลงไฟล์ .WAV เป็น .MP3)

#### 5.2 การพัฒนาระบบ

ในส่วนของการเชื่อมต่อกับฐานข้อมูลซึ่งเป็นแบบ Relational Database ที่ได้ทำการแปลงมาจาก Class Diagram ในส่วนของ Object Oriented Design แล้วนั้น จะทำการติดต่อโดยผ่านทาง ActiveX Data Object หรือ ADO ที่ทำหน้าที่เป็นตัวกลางระหว่าง OLE Provider กับ Visual Basic ซึ่งจะมีรูปแบบการติดต่อกับฐานข้อมูลเป็นลักษณะการ Connection เข้ากับฐานข้อมูล ซึ่งวิธีในการจัดการกับข้อมูลต่าง ๆ ในฐานข้อมูลเช่น การเพิ่ม, แก้ไข, ลบข้อมูลและค้นหาข้อมูลจะกระทำผ่านทาง Method ของออบเจ็ค Recordset โดยแบบจำลองการทำงานของ ADO แสดงดังรูป 5-1



รูปที่ 5-1 แบบจำลองการทำงานของ ADO Data Control

### 5.3 ผลการทำงาน

ในส่วนของผลการทำงานของระบบที่ได้ทำการพัฒนาขึ้นมา นั้นสามารถสรุปผลการทำงานได้ดังต่อไปนี้

- สามารถทำการตรวจสอบสิทธิในการเข้าใช้งาน(Login)ของผู้ใช้แต่ละคนได้อย่างถูกต้อง
- ส่วนของการจัดเก็บข้อมูลสามารถทำการจัดเก็บข้อมูลตามโครงสร้างของฐานข้อมูลที่ได้ออกแบบไว้
- สามารถแสดงผลของข้อมูลและค้นหาข้อมูลที่ต้องการได้
- การเล่นเพลงโปรแกรมสามารถการเล่นเพลงได้อย่างต่อเนื่องและสามารถจัดเก็บเป็นรายการเล่นเพลงรวมทั้งตั้งเวลาเล่นเพลงล่วงหน้าได้ แต่ยังคงขาดในส่วนของการพัฒนาตัวควบคุมความถี่ของเสียง (Equalizer) ทำให้ผู้ใช้ไม่สามารถปรับแต่งเสียงทุ่มหรือแหลมได้
- สามารถแสดงผลการขอเพลงและการเปิดเพลงรวมทั้งออกรายงานต่าง ๆ ได้

## บทที่ 6

### บทสรุปและข้อเสนอแนะ

สำหรับโครงการนี้มีจุดประสงค์เพื่อการศึกษาแนวคิดในการออกแบบและพัฒนาระบบ โดยใช้หลักการเชิงวัตถุ ระบบงานที่พัฒนานี้เป็นลักษณะงาน ที่ช่วยสนับสนุนการจัดการรายการวิทยุ โดยหน้าที่หลัก ๆ เช่นการจัดเก็บข้อมูลเพื่อการค้นหาและเล่นเพลง โดยแนวทางการศกษาเริ่มจากการวิเคราะห์และออกแบบระบบตามหลักการเชิงวัตถุโดยใช้ UML เป็นเครื่องมือช่วยในการวิเคราะห์และออกแบบระบบ ในส่วนของการพัฒนาโปรแกรมนั้นได้ใช้ Visual Basic version 6.0 เป็นเครื่องมือในการพัฒนาระบบ บนฐานข้อมูล MS Access 97

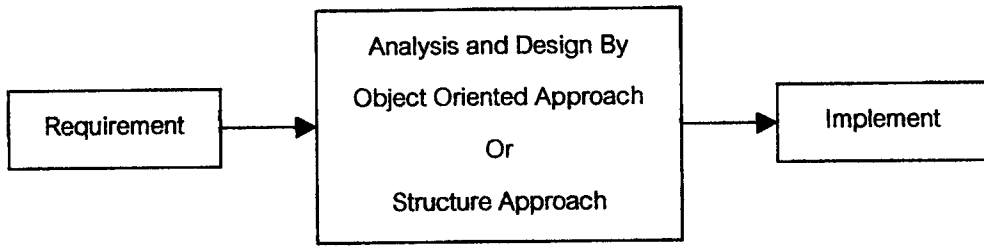
จากผลการศึกษาพบว่าวิธีการเชิงวัตถุที่นำมาใช้ในการวิเคราะห์และออกแบบระบบนี้เป็นแนวคิดที่สามารถทำความเข้าใจได้ง่ายทั้งในแง่ของผู้วิเคราะห์และผู้ใช้งาน โดยอาศัยการมองสิ่งต่าง ๆ ในระบบเป็นวัตถุหนึ่ง ๆ และพิจารณาว่าวัตถุต่าง ๆ เหล่านั้นมีคุณสมบัติ (Attribute) อะไรที่จะบ่งชี้ถึงตัวมันเองได้ และมีความสามารถ (Method Or Operation) ในการที่จะทำอะไร ได้บ้าง จากนั้นก็ทำการวิเคราะห์ถึงความสัมพันธ์ของวัตถุต่าง ๆ ที่เกี่ยวข้องอยู่ในระบบ เมื่อผ่านขั้นตอนการวิเคราะห์เรียบร้อยแล้วก็จะพิจารณาว่าระบบที่จะพัฒนานั้นจะต้องมีการปรับเปลี่ยนแก้ไขอะไรบ้าง จากนั้นก็ทำการออกแบบการทำงานของระบบและสร้างหน้าจอการใช้งานที่สัมพันธ์กับที่ได้ออกแบบไว้ เมื่อเราทำการวิเคราะห์และออกแบบระบบเสร็จเรียบร้อยแล้ว ก็จะนำเอาการออกแบบที่ได้ไปเป็นพื้นฐานสำหรับการพัฒนาระบบต่อไป

นอกจากนี้แนวคิดในเชิงวัตถุยังมีประโยชน์ในการ Maintenance ไม่ว่าจะเป็นคุณสมบัติในการนำกลับมาใช้ใหม่ (Reusability) ที่สามารถช่วยให้นำ Object ที่พัฒนาแล้วไปใช้งานกับระบบอื่น ๆ ได้ เพราะแนวคิดดังกล่าวนี้สามารถแก้ไขเพิ่มเติมโปรแกรมได้อย่างสะดวกรวดเร็ว และไม่ส่งผลกระทบต่อส่วนอื่น ๆ ของระบบ

#### 6.1 บทสรุป

สำหรับการพัฒนาระบบงานนั้นไม่ว่าเราจะทำการวิเคราะห์และออกแบบระบบด้วยวิธีการในรูปแบบใด ๆ ก็ตาม ผลของการดำเนินงานนั้น ๆ จะต้องนำไปสู่ระบบงานที่ตรงกับความต้องการของผู้ใช้ โดยเมื่อเปรียบเทียบวิธีการแบบ Structure Approach และ Object Oriented Approach จะเห็นได้ว่าเราจะมี Business Objective เป็นตัวที่จะกำหนด Problem Domain ของระบบที่จะทำ จากนั้นเราจะวิเคราะห์และออกแบบระบบไปตามวิธีการของโมเดลต่าง ๆ ที่เลือกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-1 แนวคิดเปรียบเทียบวิธีการวิเคราะห์และออกแบบระบบเชิงวัตถุกับแบบโครงสร้าง

โดยวิธีการวิเคราะห์และออกแบบระบบโดยวิธีการเชิงวัตถุนี้มีการใช้ Diagram ที่หลากหลายสำหรับการนำเสนอข้อมูลในส่วนต่าง ๆ ของระบบ ทำให้สามารถที่จะทำความเข้าใจได้ง่าย เช่น Use Case Diagram จะช่วยให้ทั้งทางด้านของ User และผู้วิเคราะห์เกิดความเข้าใจตรงกันว่าในระบบที่จะพัฒนาขึ้นมานั้นจะต้องรองรับการทำงานในส่วนต่าง ๆ ที่แสดงใน Use Case Diagram โดยผู้วิเคราะห์อาจขอความร่วมมือจาก User ในการที่จะให้วาด Use Case Diagram เพื่อแสดงขอบเขตของงานขึ้นมาให้ รวมทั้ง User เองก็สามารถที่จะเข้าใจ Diagram โดยที่ไม่ต้องอาศัยความรู้เกี่ยวกับการพัฒนาระบบงาน หรือในส่วนของการทำงาน Reverse Engineering ก็ช่วยให้สามารถสร้าง Diagrams ที่สัมพันธ์กับ Prototype ที่สร้างขึ้นในลักษณะของ Demo Program ซึ่งช่วยให้ User มองเห็นโครงสร้างของระบบงานได้ชัดเจนยิ่งขึ้นและยังช่วยลดเวลาในส่วนของการออกแบบระบบไปได้มาก

## 6.2 ข้อเสนอแนะ

โปรแกรมจัดเก็บข้อมูลเพลงที่ได้ทำการพัฒนาขึ้นมานี้จะช่วยให้การทำงานทางด้านการจัดรายการวิทยุมีประสิทธิภาพมากขึ้นทั้งในเรื่องของการจัดเก็บและค้นหาเพลง รวมถึงให้ข้อมูลต่าง ๆ ที่เกี่ยวข้องกับการทำงาน แต่ในด้านของการนำเอาหลักการเชิงวัตถุมาใช้นั้นเป็นเพียงส่วนของการวิเคราะห์และออกแบบเชิงวัตถุ แต่ในส่วนของการนำข้อมูลที่ใช้ยังคงเป็นฐานข้อมูลแบบ Relation Database ดังนั้นหากในอนาคตมีผู้สนใจที่จะทำการพัฒนาระบบงานนี้ต่อไปควรพิจารณาศึกษาในแง่ของการใช้ Object Oriented Database Management System (OODBMS) เพื่อให้เกิดความสมบูรณ์ในรูปแบบเชิงวัตถุมากขึ้น อันจะก่อให้เกิดประโยชน์และประสิทธิภาพในการทำงานตามคุณสมบัติของ Object Oriented Approach ได้อย่างสมบูรณ์ยิ่งขึ้น รวมไปถึงส่วนของการพัฒนาไปในแนวทางของ Web Application ซึ่งก็จะช่วยให้การทำงานเปิดกว้างและเป็นประโยชน์มากยิ่งขึ้นต่อไปในอนาคต

## บรรณานุกรม

กิตติ ภัคศิวิวัฒน์กุล และกิตติพงษ์ กลมกล่อม. 2544. UML วิเคราะห์และออกแบบระบบเชิงวัตถุ.

กรุงเทพฯ: ไทยเจริญการพิมพ์.

เบญจรงค์ นิตยพัฒน์. 2543. การพัฒนาห้องสมุดดิจิทัลโดยใช้ UML. กรุงเทพฯ:

คณะเทคโนโลยีสารสนเทศ: สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Bahrami, Ali. 1999. **Object Oriented Systems Development**: McGraw-Hill.

Cetus team. 1996-2001. **Architecture and Design : Unified Modeling Language (UML)**.

[online]. Available: [http://www.cetus-links.org/oo\\_uml.html](http://www.cetus-links.org/oo_uml.html).

Eriksson, Hans-Erik and Penker, Magnus. 1998. **UML Toolkit**: John Wiley & Sons.

Jeffrey, L Whitten. et. al. 2001. **Systems Analysis and Design Methods**. 5<sup>th</sup> ed. McGraw-Hill.

Tkach, Daniel and Puttick Richard. 1996. **Object Technology in Application Development**.

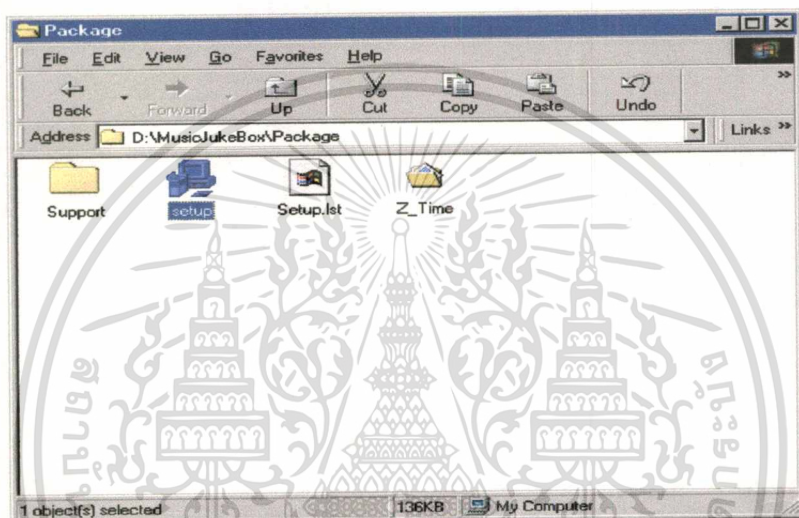
2<sup>nd</sup> ed. CA: Addison-Wesley.

## ภาคผนวก

### การติดตั้งและใช้งานโปรแกรมจัดเก็บข้อมูลเพลง

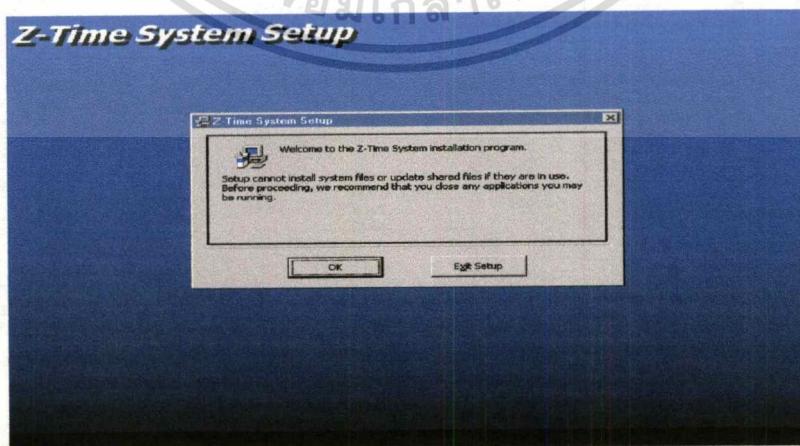
#### 1. การติดตั้งโปรแกรมจัดเก็บข้อมูลเพลง มีรายละเอียดขั้นตอนการติดตั้งดังต่อไปนี้

##### 1.1 ทำการเลือกไฟล์ Setup.exe จาก Package ดังรูปที่ ก-1



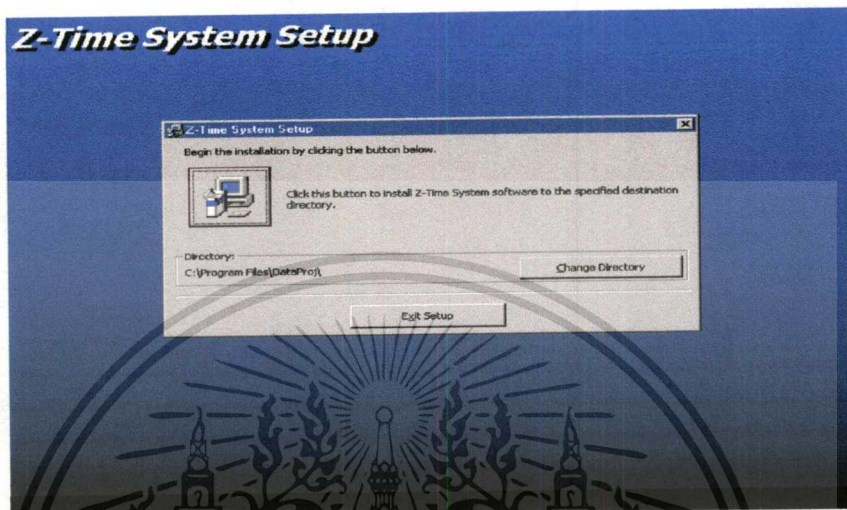
รูปที่ ก-1 หน้าจอเริ่มต้นการติดตั้งโปรแกรมจัดเก็บข้อมูลเพลง

##### 1.2 หลังจากเลือก Setup.exe แล้วจะปรากฏหน้าจอให้ยืนยันการติดตั้ง ดังรูปที่ ก-2 ให้เลือก OK เพื่อยืนยันการติดตั้งโปรแกรม



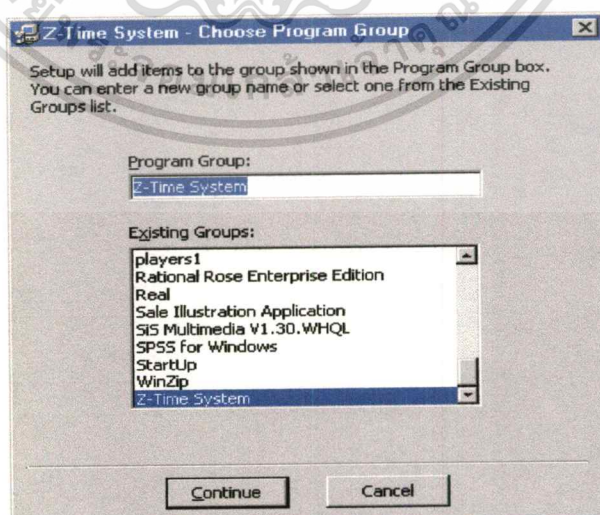
รูปที่ ก-2 หน้าจอให้ทำการยืนยันการติดตั้งโปรแกรม

- 1.3 หลังจากที่ยืนยันการติดตั้งโปรแกรมแล้วจะปรากฏหน้าจอเพื่อให้เลือกที่จะเปลี่ยนตำแหน่งในการติดตั้ง (Change Directory) หากไม่ต้องการเปลี่ยนให้เลือกที่ Button to install ดังรูปที่ ก-3 เพื่อทำการติดตั้งโปรแกรม



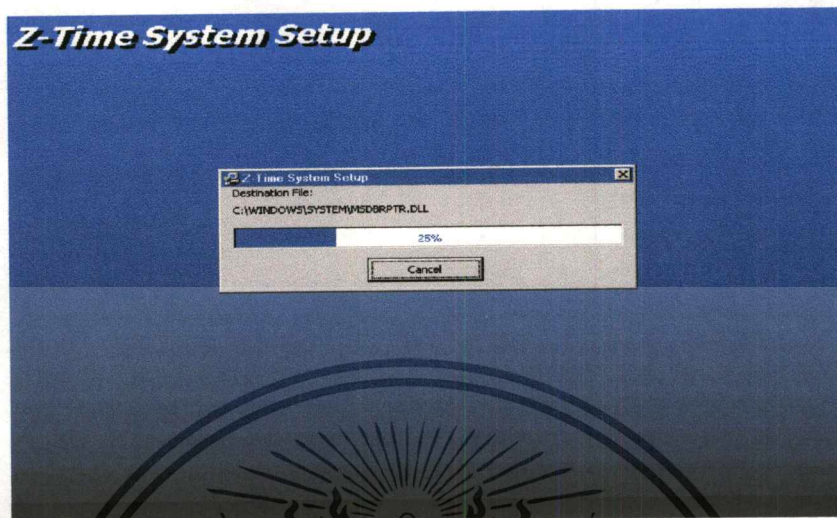
รูปที่ ก-3 หน้าจอให้เลือกกำหนดตำแหน่งที่ต้องการติดตั้งโปรแกรม

- 1.4 จากนั้นจะปรากฏหน้าจอในการแสดงโปรแกรมในกลุ่มของโปรแกรมที่มีการติดตั้งอยู่ก่อนแล้ว ดังรูปที่ ก-4 ซึ่งในที่นี่กำหนด Default เป็น Z-Time System ซึ่งจะเป็นชื่อที่ปรากฏในการเรียกใช้งานโปรแกรม หลังจากทำการกำหนดเรียบร้อยแล้วให้กดปุ่ม Continue เพื่อทำการติดตั้งโปรแกรมต่อไป



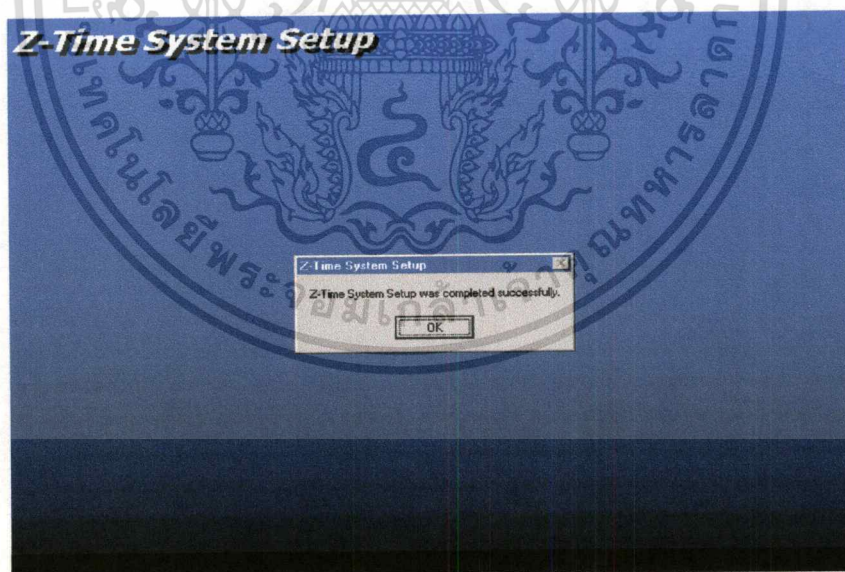
รูปที่ ก-4 หน้าจอการกำหนดกลุ่มของโปรแกรม

1.5 จากนั้นโปรแกรมจะทำการติดตั้ง ไฟล์ต่าง ๆ โดยอัตโนมัติ ดังรูปที่ ก-5



รูปที่ ก-5 หน้าจอขณะทำการติดตั้งไฟล์

1.6 เมื่อทำการติดตั้งเสร็จเรียบร้อยแล้วจะปรากฏหน้าจอแสดงการสิ้นสุดการติดตั้งดังรูปที่ ก-6



รูปที่ ก-6 หน้าจอเมื่อทำการติดตั้งเสร็จสมบูรณ์

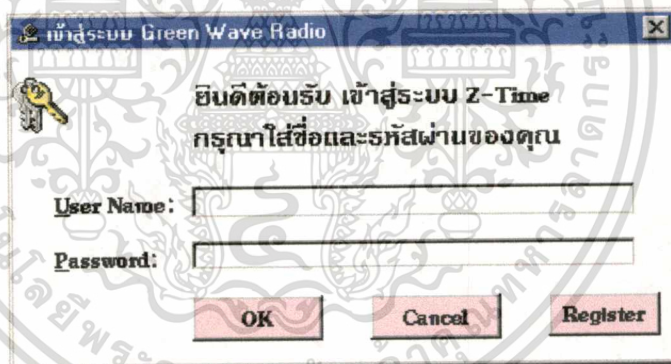
## 2. การใช้งานโปรแกรมจัดเก็บข้อมูลเพลง

- 2.1 เมื่อทำการเข้าสู่โปรแกรมจะปรากฏหน้าจอให้เลือกสถานีที่ต้องการใช้งานดังรูปที่ ก-7



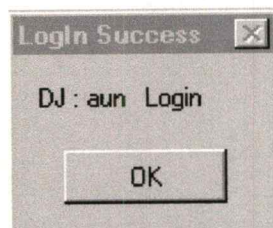
รูปที่ ก-7 หน้าจอเริ่มต้นในการใช้งานโปรแกรม

- 2.2 เมื่อเลือกสถานีที่ต้องการใช้งานแล้วจะปรากฏหน้าจอให้ทำการ Login เพื่อตรวจสอบสิทธิในการใช้งานโปรแกรม ดังรูปที่ ก-8



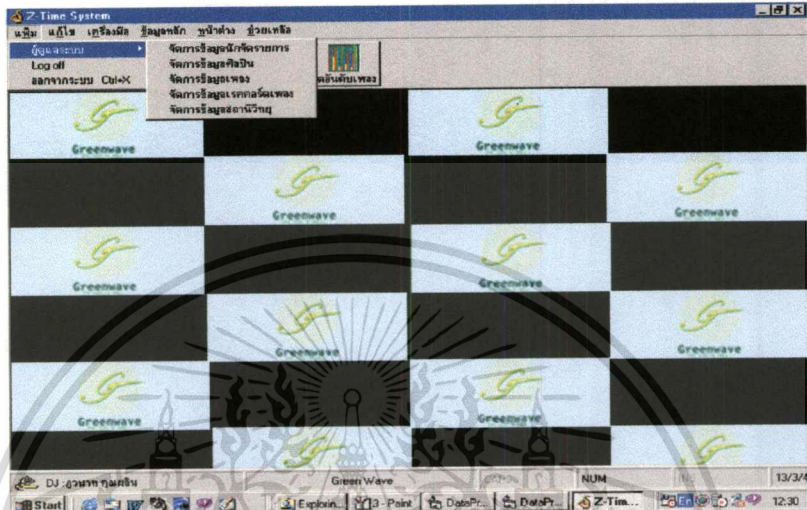
รูปที่ ก-8 หน้าจอการ Login เพื่อใช้งาน โปรแกรม

- 2.3 หลังจากที่ทำกร Login หากข้อมูลถูกต้อง โปรแกรมจะอนุญาตให้ผู้ใช้ใช้งานระบบ โดยแสดงการยืนยันด้วย Message box ดังรูปที่ ก-9



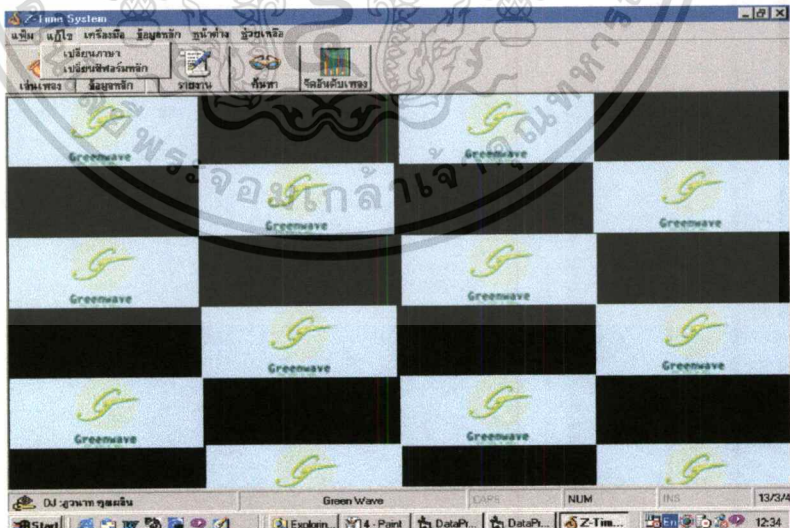
รูปที่ ก-9 หน้าจอผลของการ Login

- 2.4 เมื่อทำการ login เข้าโปรแกรมเรียบร้อยแล้ว หากผู้ใช้มีสถานะเป็น Admin ของระบบ ในส่วนของเมนูผู้ดูแลระบบ (Administrator) ซึ่งอยู่ในเมนูเพิ่มเติม จะสามารถใช้งาน ได้เพื่อให้ผู้ดูแลระบบทำการแก้ไขเพิ่มเติมข้อมูลในระบบ ดังรูปที่ ก-10



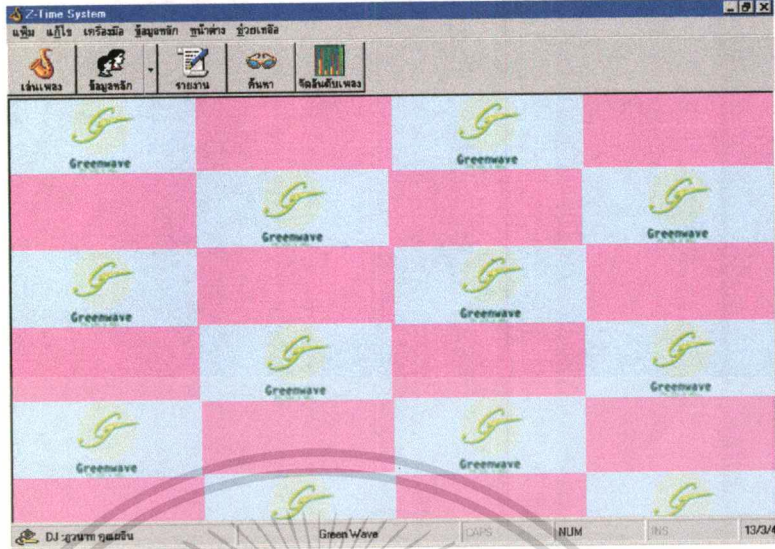
รูปที่ ก-10 หน้าจอแสดงเมนูผู้ดูแลระบบกรณีผู้ใช้มีสถานะเป็น Admin

- 2.5 ที่เมนูแก้ไขจะเป็นส่วนของการเพิ่มเติมการทำงานเพื่อให้ผู้ใช้สามารถที่จะเปลี่ยนแปลงภาษาและสีที่แสดงในหน้าจอหลัก ดังรูปที่ ก-11 และ ก-12



รูปที่ ก-11 หน้าจอแสดงเมนูแก้ไข

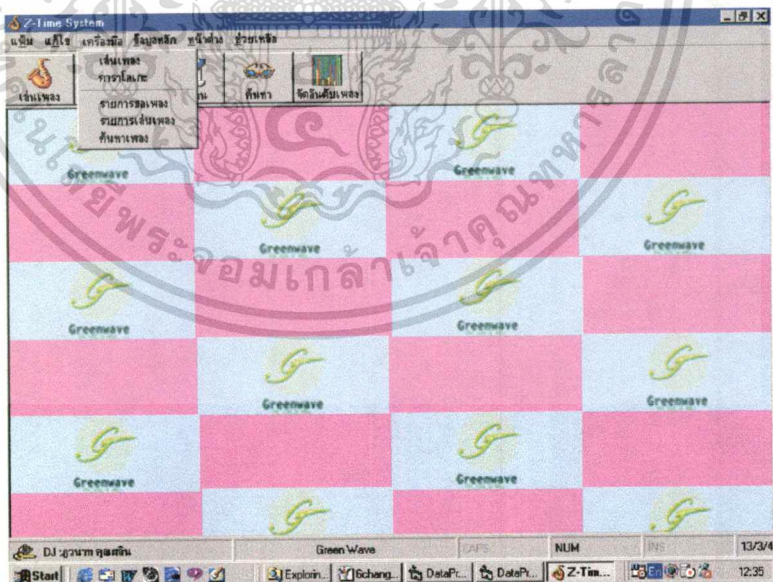
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-12 หน้าจอแสดงผลของการเปลี่ยนสีหน้าจอหลัก


2.6

ที่เมนูเครื่องมือจะเป็นส่วนของเครื่องมือต่าง ๆ ที่ใช้งานการทำงานเช่นเล่นเพลง, คาราโอเกะ, บันทึกรายการขอเพลง, รายการเปิดเพลงและการค้นหาเพลง ดังรูปที่ ก-13



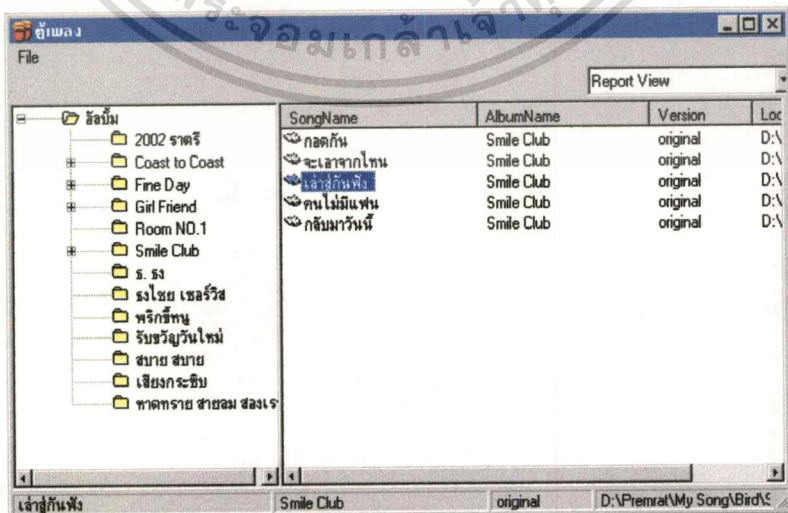
รูปที่ ก-13 หน้าจอแสดงเมนูเครื่องมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.7 ในส่วนของการเล่นเพลงเมื่อทำการเลือกที่เมนูเครื่องมือ > เล่นเพลง หรือที่ Toolbar เล่นเพลง จะปรากฏหน้าจอการเล่นเพลงดังรูปที่ ก-14 ซึ่งสามารถเลือกเพลงที่ต้องการเปิดได้จากตู้เพลงที่เก็บเพลงต่าง ๆ ที่จัดเก็บอยู่ในฐานข้อมูลมาแสดงโดยเลือกที่ปุ่ม Add ดังรูปที่ ก-15 เมื่อทำการเลือกเพลงที่ต้องการเรียบร้อยรายการเพลงที่ต้องการจะปรากฏอยู่ใน List ที่หน้าจอเล่นเพลงผู้ใช้สามารถทำการเล่นเพลงได้โดยกดที่ปุ่ม  หรือกดดับเบิลคลิกที่เพลง เพื่อทำการเล่นเพลง ดังรูปที่ ก-16



รูปที่ ก-14 หน้าจอแสดงเครื่องมือเล่นเพลง



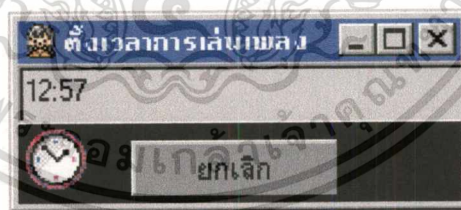
รูปที่ ก-15 หน้าจอตู้เพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



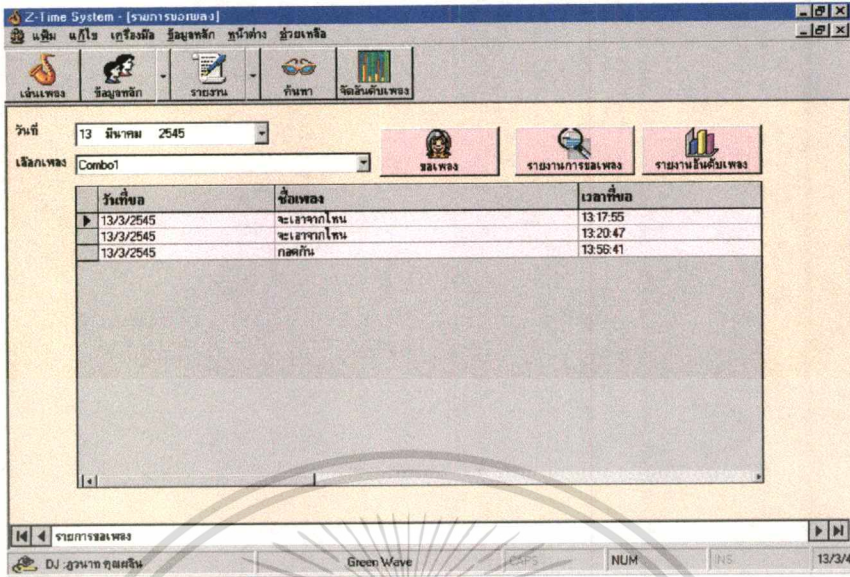
รูปที่ ก-16 หน้าจอขณะทำการเล่นเพลง

- 2.8 ผู้ใช้สามารถทำการบันทึกรายการเล่นเพลงที่ต้องการได้โดยการกด Save list หรือเรียกรายการเล่นเพลงที่ทำการบันทึกไว้โดยกด Load list และหากผู้ใช้อยังไม่ต้องการทำการเล่นเพลงทันที สามารถที่จะทำการตั้งเวลาในการเล่นเพลงได้โดยการกดที่ Set AutoPlay จะปรากฏหน้าจอตั้งรูปที่ ก-17 เพื่อให้ทำการตั้งเวลาและระบบจะทำการเล่นเพลงเมื่อถึงเวลาตามที่ตั้งไว้โดยอัตโนมัติ



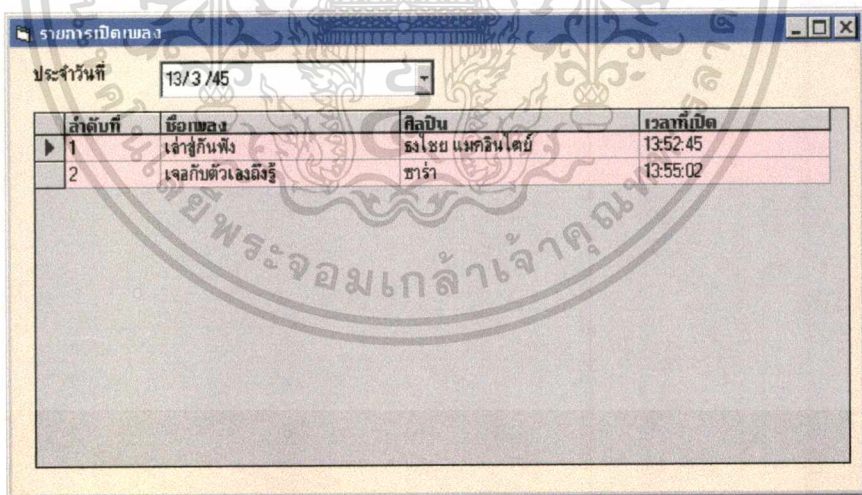
รูปที่ ก-17 หน้าจอการตั้งเวลาเล่นเพลง

- 2.9 ในกรณีที่ผู้ใช้ทำการขอเพลงผู้ใช้สามารถทำการบันทึกข้อมูลการขอเพลงได้โดยการเลือกที่เมนูเครื่องมือ > รายการขอเพลง ซึ่งจะปรากฏหน้าจอให้ทำการบันทึกรายการขอเพลงดังรูปที่ ก-18



รูปที่ ก-18 หน้าจอรายการขอเพลง

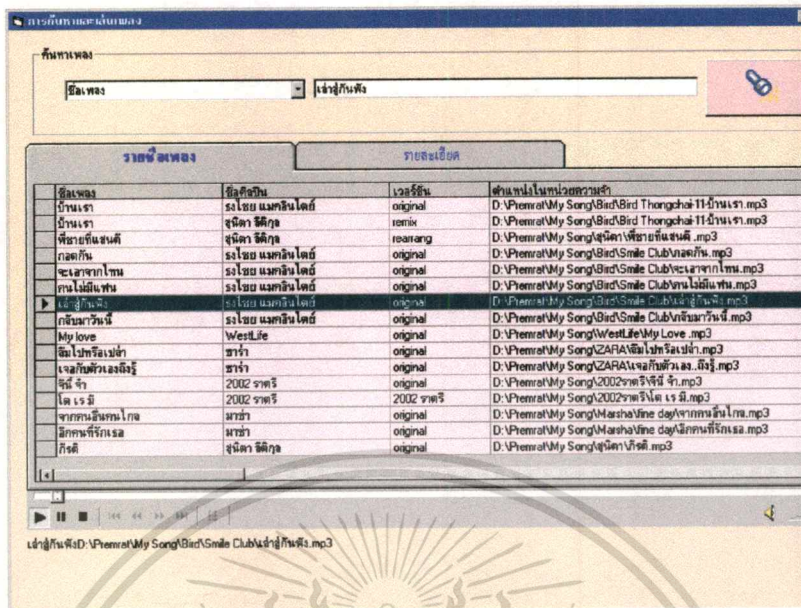
- 2.10 ที่เมนูเครื่องมือ > รายการเปิดเพลง จะเป็นการแสดงรายการเพลงที่เปิดไปแล้ว ดังรูปที่ ก-19



รูปที่ ก-19 หน้าจอรายการเปิดเพลง

- 2.11 ที่เมนูเครื่องมือ > ค้นหาเพลง จะเป็นส่วนของการค้นหาและเล่นเพลงแบบทันที โดยจะแสดงรายละเอียดของเพลงเช่น ชื่อเพลง, ผู้แต่งคำร้อง, ผู้แต่งทำนอง, เนื้อเพลง ฯลฯ และผู้ใช้สามารถเลือกดับเบิลคลิกที่ชื่อเพลงที่ต้องการเพื่อเล่นเพลง

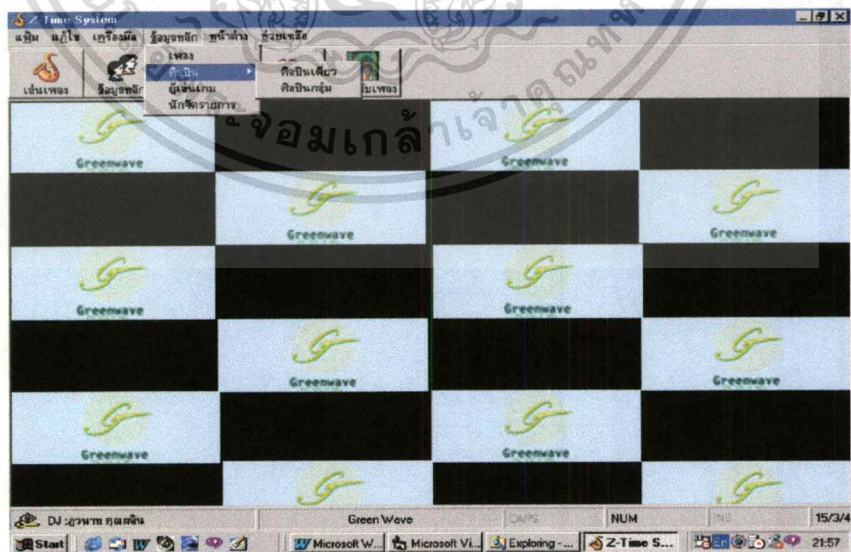




รูปที่ ก-22 หน้าจอการเล่นเพลงแบบทันทีจากหน้าจอค้นหาและเล่นเพลง

2.12

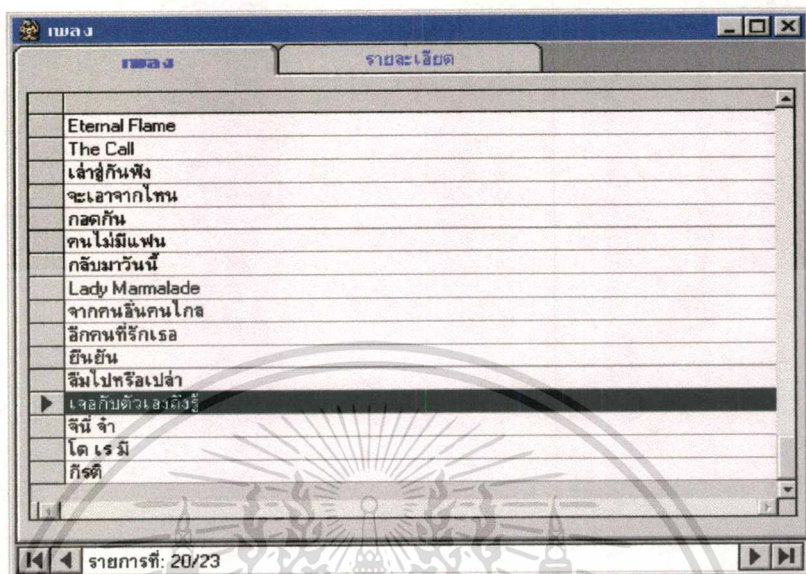
ที่เมนูข้อมูลหลัก จะเป็นเมนูสำหรับผู้ใช้ที่มีสถานะเป็น User เข้ามาใช้ดูและค้นหาข้อมูลซึ่งจะไม่สามารถทำการแก้ไข เพิ่มเติมข้อมูลต่างๆ ได้ โดยเมนูย่อยประกอบไปด้วย ข้อมูลเพลง, ข้อมูลศิลปินเดี่ยว, ข้อมูลศิลปินกลุ่ม, ข้อมูลผู้เล่นเกมส์ และข้อมูลนักจัดรายการ ดังรูปที่ ก-23



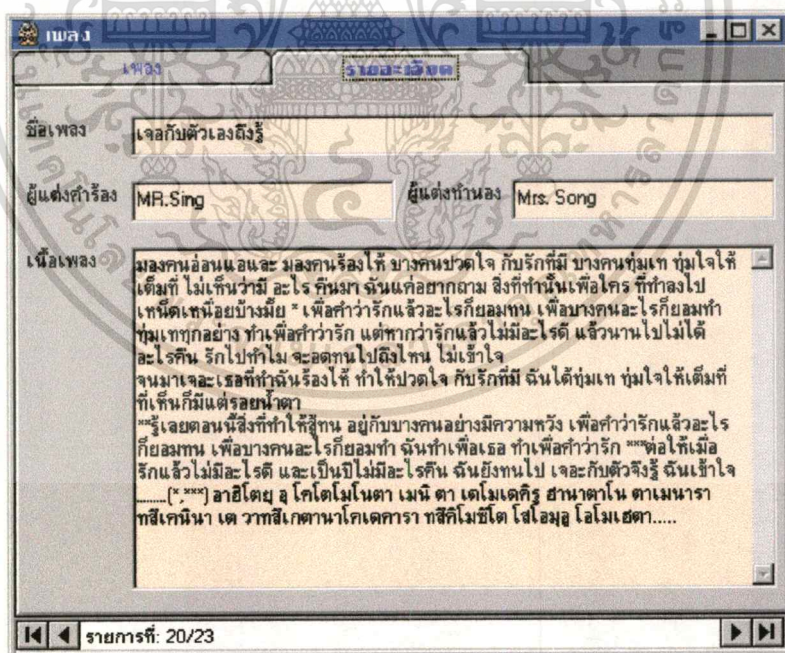
รูปที่ ก-23 หน้าจอเมนูข้อมูลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.13 ที่เมนูข้อมูลหลัก > ข้อมูลเพลง จะแสดงข้อมูลเกี่ยวกับเพลงต่าง ๆ ที่มีอยู่ในฐานข้อมูลพร้อมทั้งรายละเอียดของเพลงนั้น ๆ ดังแสดงในรูป ก-24 และ ก-25

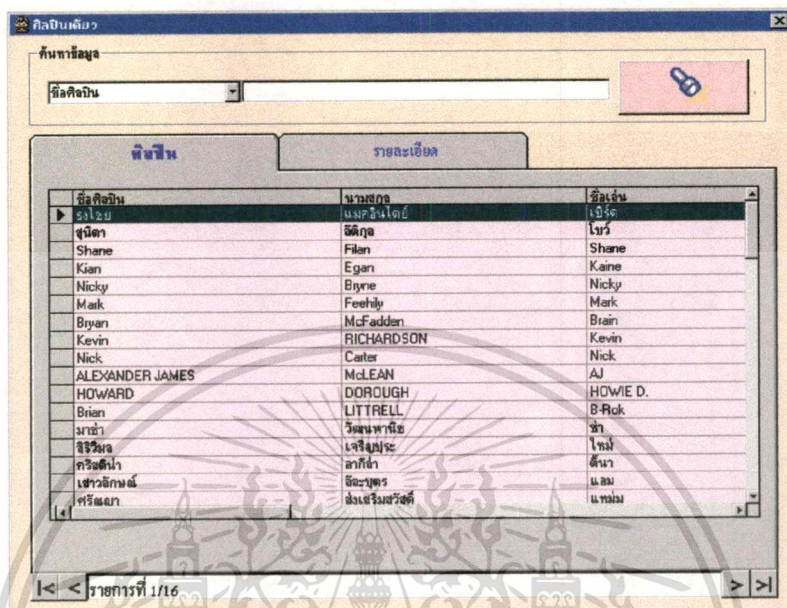


รูปที่ ก-24 หน้าจอข้อมูลเพลง

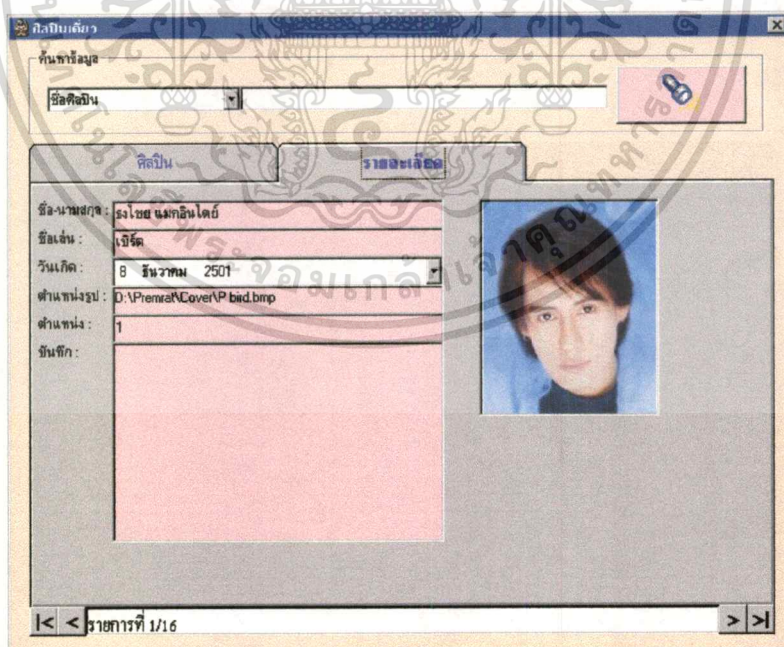


รูปที่ ก-25 หน้าจอรายละเอียดข้อมูลเพลง

- 2.14 ที่เมนูข้อมูลหลัก > ศิลปิน>ศิลปินเดี่ยว จะเป็นการแสดงข้อมูลของศิลปินเดี่ยวทั้งหมดที่อยู่ในฐานข้อมูล ดังรูปที่ ก-26 และ ก-27

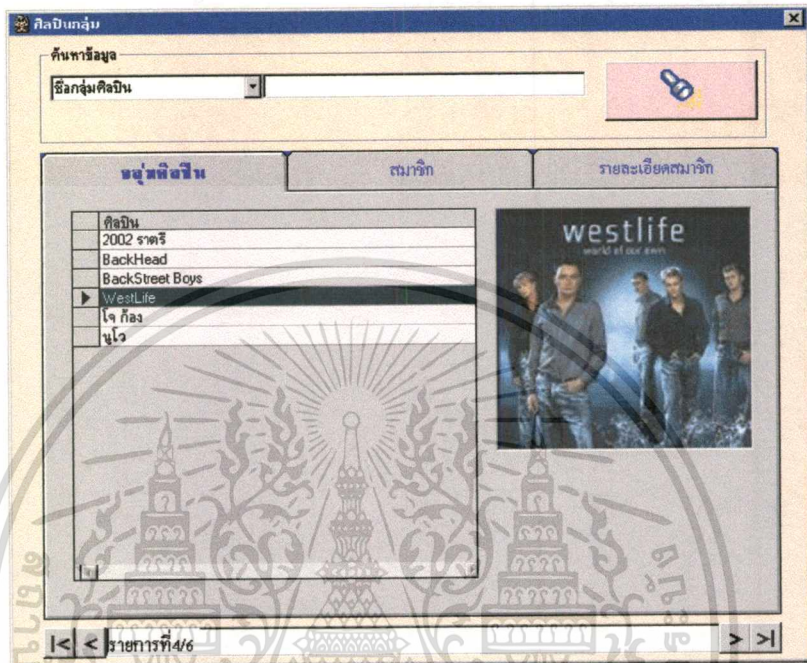


รูปที่ ก-26 หน้าจอแสดงรายชื่อศิลปินเดี่ยว

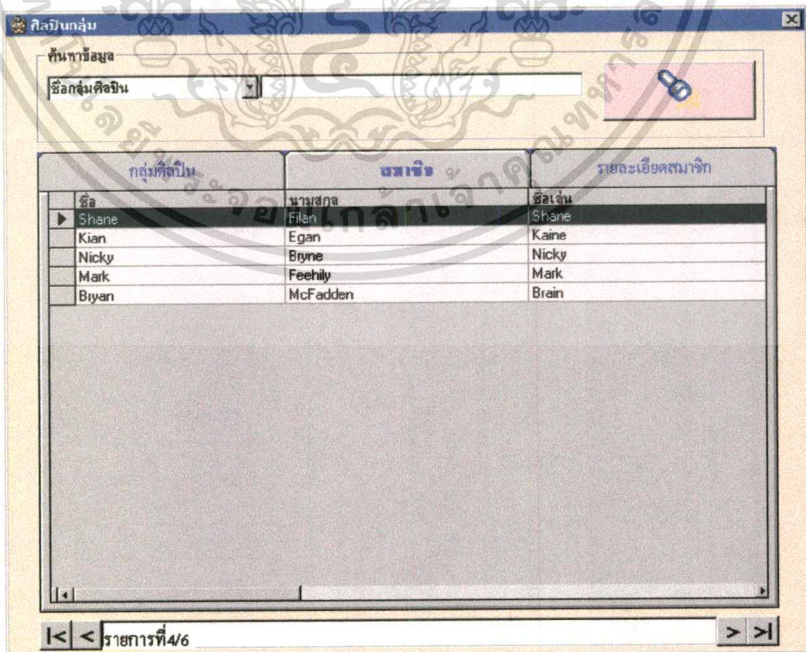


รูปที่ ก-27 หน้าจอแสดงรายละเอียดของศิลปินเดี่ยว

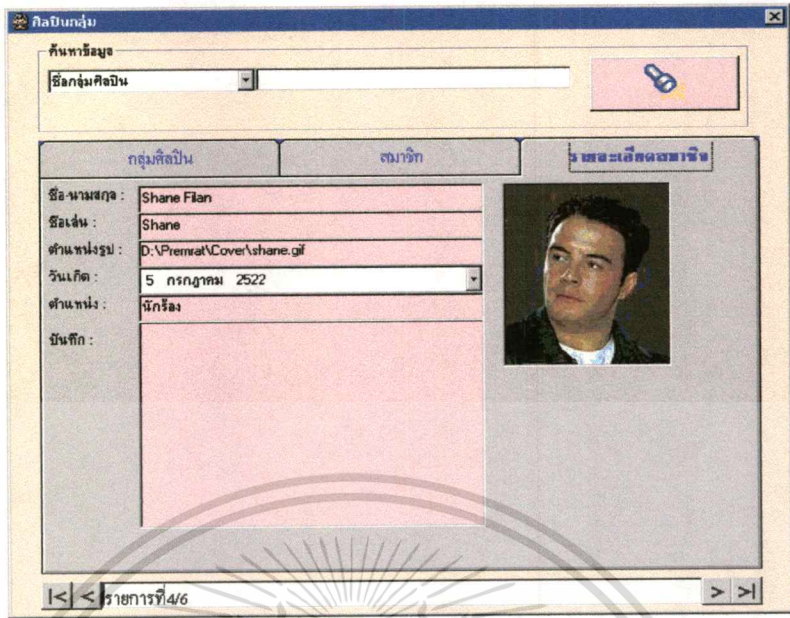
- 2.15 ที่เมนูข้อมูลหลัก > ศิลปิน>ศิลปินกลุ่ม จะเป็นการแสดงข้อมูลของศิลปินกลุ่มและสมาชิกในกลุ่ม โดยประกอบไปด้วย Tab ศิลปินกลุ่ม, Tab สมาชิก และ Tab รายละเอียดสมาชิก ดังรูป ก-28 ถึง ก-30



รูปที่ ก-28 หน้าจอข้อมูลศิลปิน

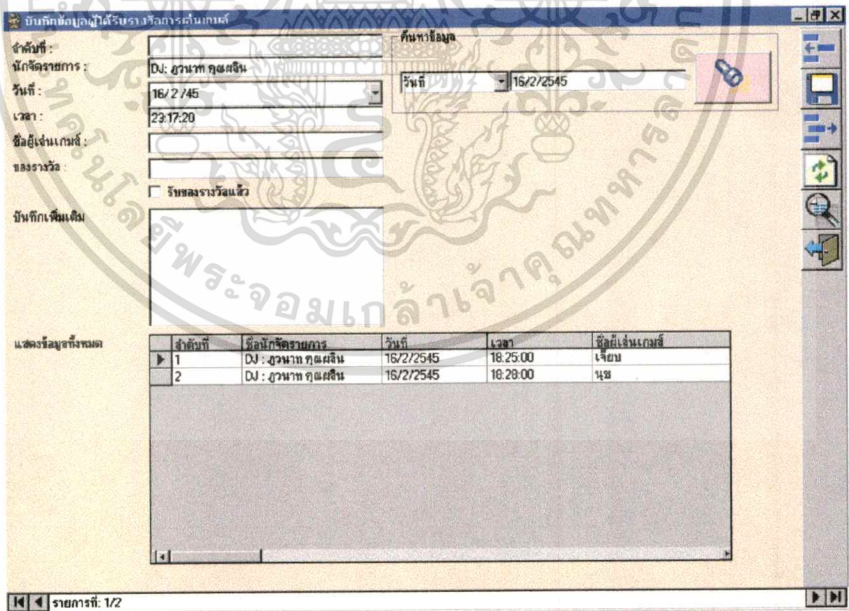


รูปที่ ก-29 หน้าจอแสดงสมาชิกในกลุ่ม



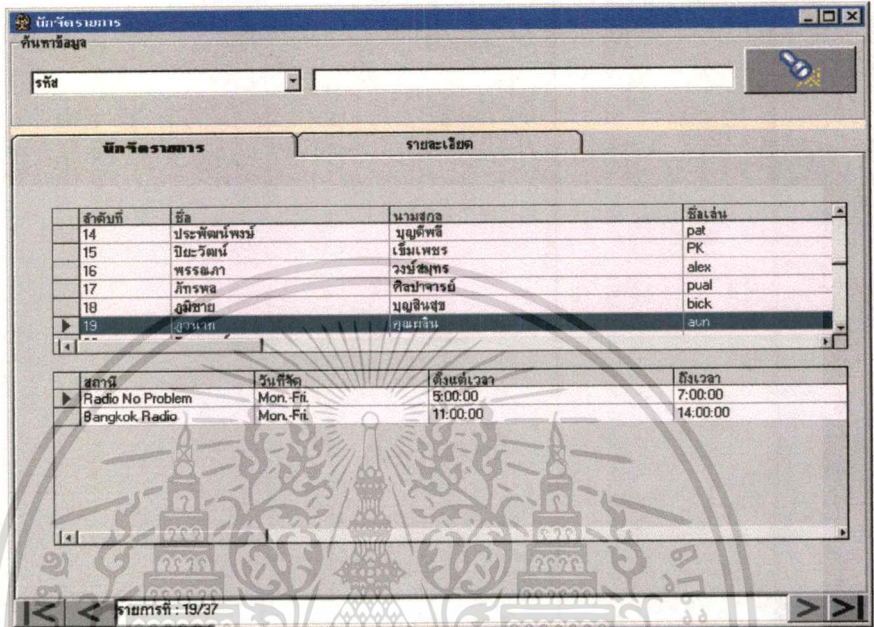
รูปที่ ก-30 หน้าจอแสดงรายละเอียดของสมาชิกในกลุ่ม

2.16 ที่เมนูข้อมูลหลัก > ข้อมูลผู้เล่นเกมส์ จะเป็นส่วนของการบันทึกข้อมูลของผู้ที่ได้  
รับรางวัลจากการร่วมเล่นเกมส์ ดังรูปที่ ก-31

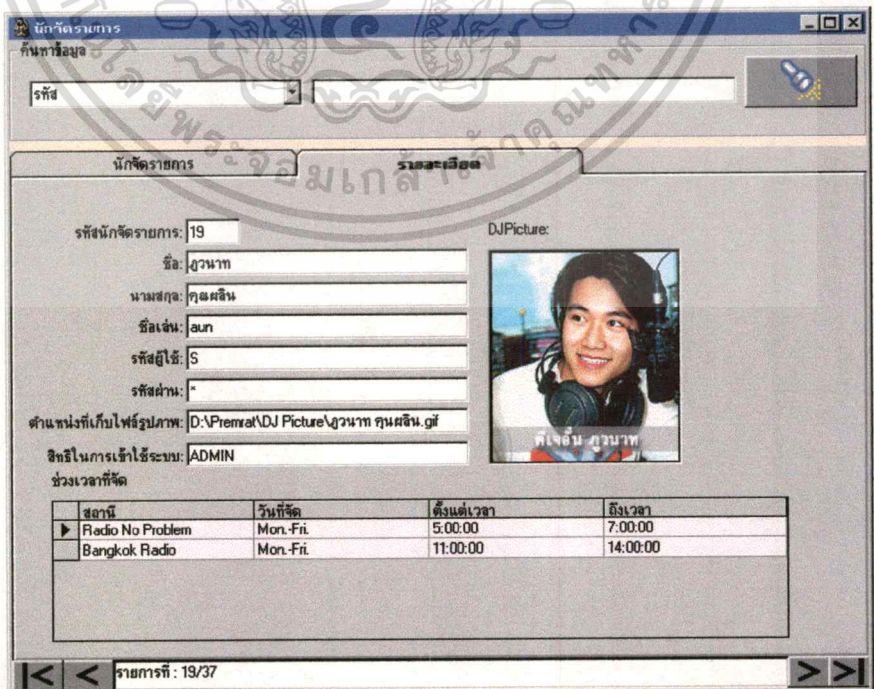


รูปที่ ก-31 หน้าจอข้อมูลผู้เล่นเกมส์

- 2.17 ที่เมนูข้อมูลหลัก > นักจัดรายการ จะเป็นส่วนที่แสดงข้อมูลต่าง ๆ และรายละเอียดของนักจัดรายการแต่ละคน รวมทั้งส่วนของช่วงเวลาและสถานีที่จัดรายการ ดังรูปที่ ก-32 และ ก-33



รูปที่ ก-32 หน้าจอข้อมูลนักจัดรายการ



รูปที่ ก-33 หน้าจอรายละเอียดนักจัดรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิได้อนุญาตให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.18 ในส่วนของรายงานต่าง ๆ เช่นรายงานการขอเพลง, รายงานข้อมูลผู้เล่นเกมส์ ฯลฯ ผู้ใช้สามารถเลือกได้จากส่วนของรายงานที่ Toolbar ซึ่งตัวอย่างหน้าจอรายงานการขอเพลงและรายงานข้อมูลผู้เล่นเกมส์ แสดงดังรูป ก-34 และ ก-35 ตามลำดับ

รายงานการขอเพลง				
ประจำเดือน	DJ	วันที่ขอ	เวลา	ชื่อเพลง
March 2002				
	DJ : ภูวนาท คุณผลิน	13/3/02	19:17:55	จะเอาจากไหน
	DJ : ภูวนาท คุณผลิน	13/3/02	19:20:47	จะเอาจากไหน
	DJ : ภูวนาท คุณผลิน	13/3/02	19:56:41	กอดกัน

รูปที่ ก-34 หน้าจอรายงานการขอเพลง

รายงานข้อมูลผู้เล่นเกมส์							
ประจำเดือน	ลำดับที่	ชื่อสมาชิก	วันที่	เวลา	ผู้เล่นเกม	รางวัล	รับแล้ว
February 2002							
	1	DJ : ภูวนาท คุณผลิน	16/2/02	18:25:00	เจียม	Gift Set จาก PCT	<input type="checkbox"/>
	2	DJ : ภูวนาท คุณผลิน	16/2/02	18:28:00	นุช	Contact Len	<input type="checkbox"/>

รูปที่ ก-35 หน้าจอรายงานผู้เล่นเกมส์

## ประวัติผู้เขียน

ชื่อผู้เขียน

นางสาวเปรมรัตน์ พูลสวัสดิ์

วันเดือนปีเกิด

28 เมษายน 2520

สถานที่เกิด

จังหวัดกรุงเทพมหานคร

วุฒิการศึกษาระดับปริญญาตรี

วท.บ.(วิทยาการคอมพิวเตอร์)

สถานที่สำเร็จการศึกษา

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยหัวเฉียวเฉลิมพระเกียรติ

ปีที่สำเร็จการศึกษา

ปีการศึกษา 2541



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้