# การพัฒนาระบบสารสนเทศเพื่อการจัดการตารางสอน ของโรงเรียนดนตรี

# Development an Information System for Class Scheduling


โดย

นาย กรกต ขวัญเมือง

รหัส 44067440

*H002034*

อาจารย์ที่ปรึกษา

ผศ.ดร.นพพร โชติกกำธร

25 มี.ค. 2550
วัน เดือน ปี.....
02034
เลขทะเบียน.....
ส.พ. ก475ก 2546
เลขเรียกหนังสือ.....
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."


รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2546

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **ชื่อหัวข้อ** | การพัฒนาระบบสารสนเทศเพื่อการจัดการตารางสอนของโรงเรียนดนตรี |
| **นักศึกษา** | นายกรกต ขวัญเมือง |
| **อาจารย์ที่ปรึกษา** | ผศ.ดร.นพพร โชติกกำธร |
| **ระดับการศึกษา** | วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ |
| **แขนงวิชา** | วิทยาการสารสนเทศ |
| **ปีการศึกษา** | 2546 |

# บทคัดย่อ


วิทยาลัยดุริยางคศิลป์ มหาวิทยาลัยมหิดล เป็นโรงเรียนสอนดนตรี ที่อยู่ภายใต้มหาวิทยาลัยมหิดล มีแผนการที่จะนำระบบการจัดการเรียนการสอน และระบบการคิดค่าตอบแทนอาจารย์มาใช้ในการดำเนินงาน

โครงงานพัฒนาระบบสารสนเทศเพื่อการจัดการตารางสอนนำเสนอผลการวิเคราะห์ออกแบบ และพัฒนาระบบคอมพิวเตอร์ เพื่อรองรับการทำงานด้านการจัดตารางสอน โดยระบบที่ทำการพัฒนาขึ้นมานั้นประกอบด้วย 4 ระบบหลักๆ คือ ระบบการลงทะเบียน ระบบการจัดตารางเรียน ระบบการจองเวลาเรียน และระบบบันทึกเวลาเรียน โดยได้ทำการศึกษาขั้นตอนการทำงานของทั้ง 4 ระบบ ในวิทยาลัยดุริยางคศิลป์ มหาวิทยาลัยมหิดล และนำมาวิเคราะห์ออกแบบในเชิงวัตถุ โดยใช้ UML เป็นโมเดลในการออกแบบ ส่วนในการพัฒนาใช้ VB.NET เป็นเครื่องมือในการพัฒนา และใช้ Microsoft Access เป็นซอฟแวร์ระบบฐานข้อมูล โดยโปรแกรมที่สร้างขึ้นจะนำไปใช้รองรับการทำงานที่เกิดขึ้นในวิทยาลัยดุริยางคศิลป์ มหาวิทยาลัยมหิดลต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Title** | Development an Information System for Class Scheduling |
| **Student** | Mr. Goragot Kuanmuang |
| **Advisor** | Asst. Prof. Nopporn Chotikakamthorn |
| **Level of Study** | Master of Science in Information Technology |
| **Major** | Information Science |
| **Academic Year** | 2003 |


## Abstract

College of Music is a musical institute under Mahidol University. The institute has a plan to introduce a class scheduling system and computing teachers' wages. This project presents detail of the analysis, design and development of a computer system for class scheduling. The new system consists of 4 main subsystems namely the register subsystem, the class scheduling subsystem, the reservation subsystem and the class attendance recording subsystem. In this study, analysis and design was performed using UML. VB.NET was used to build software components and Microsoft Access was chosen as DBMS. The new system will be installed and used in the College of Music, Mahidol University.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณ ผศ.ดร.นพพร โชติกกำธร ซึ่งได้ให้คำปรึกษาและคำเสนอแนะต่างๆ จนกระทั่งโครงการพัฒนาระบบงานนี้เสร็จสมบูรณ์ และขอขอบคุณ รศ.ดร.สุกรี เจริญสุข ผู้อำนวยการวิทยาลัยดุริยางคศิลป์ มหาวิทยาลัยมหิดล และคุณวันดี จันทร์เจนจบ ที่ได้อำนวยความสะดวกในการติดต่อขอข้อมูลต่างๆ ที่ใช้ในการพัฒนาระบบ ตลอดจนเพื่อนๆ ที่ให้คำปรึกษาที่จะเป็นต่อการพัฒนาระบบงานในครั้งนี้ด้วย

นายกรกต ขวัญเมือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I
บทคัดย่อภาษาอังกฤษ..... II
กิตติกรรมประกาศ..... III
สารบัญ..... IV
สารบัญตาราง..... V
สารบัญภาพ..... VI
บทที่
1. บทนำ
1.1 ความเป็นมา..... 1
1.2 วัตถุประสงค์..... 1
1.3 ขอบเขตของการพัฒนาระบบงาน..... 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ..... 2
2. UML (Unified Modeling Language)
2.1 UML (Unified Modeling Language)..... 3
2.2 เครื่องมือที่ใช้ในการพัฒนา..... 14
3. ขั้นตอนการดำเนินงาน
3.1 การวิเคราะห์ระบบงาน..... 15
3.2 การออกแบบระบบงาน..... 16
4. การออกแบบฐานข้อมูล
4.1 ออกแบบฐานข้อมูล..... 30
4.2 ออกแบบหน้าจอและรายงาน..... 40
5. สรุปผลและเสอนแนะ..... 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่ หน้า

4.1 ห้อง..... 35
4.2 วิชา..... 35
4.3 ห้องเรียนวิชา..... 36
4.4 ใบบันทึกเวลาเรียน..... 36
4.5 ใบลงทะเบียน..... 36
4.6 นักเรียน.....37
4.7 ตารางเรียน.....37
4.8 ใบจอง.....38
4.9 อาจารย์.....38
4.10 อาจารย์สอนวิชา..... 39
4.11 พนักงาน..... 39
4.12 ใบลงทะเบียนสอบ..... 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

ภาพที่ หน้า


3.1 Use Case Diagram แสดงการทำงานของระบบ..... 17

3.2 Sequence diagram ของ Use Case ลงทะเบียน.....18

3.3 Sequence diagram ของ Use Case จองเวลาเรียน.....18

3.4 Sequence diagram ของ Use Case ตรวจสอบเวลาเรียน.....19

3.5 Sequence diagram ของ Use Case พิมพ์ตารางเรียน.....19

3.6 Sequence diagram ของ Use case ลงทะเบียนสอบ.....20

3.7 Sequence diagram ของ Use case บันทึกผลการสอบ.....20

3.8 Sequence diagram ของ Use case บันทึกเวลาเรียน.....21

3.9 Sequence diagram ของ Use case การจัดการข้อมูลพนักงาน..... 21

3.10 Sequence diagram ของ Use case การจัดการข้อมูลอาจารย์..... 22

3.11 Sequence diagram ของ Use case การจัดการข้อมูลห้อง..... 22

3.12 Sequence diagram ของ Use case การจัดการข้อมูลนักเรียน..... 23

3.13 Sequence diagram ของ Use case การจัดการข้อมูลวิชา..... 23

3.14 Activity diagram ของ Use case การลงทะเบียน.....24

3.15 Activity diagram ของ Use case การจองเวลาเรียน..... 25

3.16 Activity diagram ของ Use case การตรวจสอบเวลาเรียน..... 25

3.17 Activity diagram ของ Use Case พิมพ์ตารางเรียน.....26

3.18 Activity diagram ของ Use case ลงทะเบียนสอบ.....26

3.19 Activity diagram ของ Use Case บันทึกผลการสอบ..... 27

3.20 Activity diagram ของ Use case บันทึกเวลาเรียน..... 27

3.21 Statechart diagram ของนักเรียน.....29

3.22 Statechart diagram ของการจอง..... 29

3.23 Class Diagram แสดงความสัมพันธ์ระหว่าง Object..... 30

4.1 คลาสไดอะแกรมของอาจารย์ พนักงาน นักเรียน และ บุคคล.....30

4.2 ฐานข้อมูลของอาจารย์ พนักงาน นักเรียน และ บุคคล.....31

4.3 คลาสไดอะแกรมของอาจารย์ และ วิชา.....31

4.4 ER ไดอะแกรมของอาจารย์ และ วิชา.....31



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ(ต่อ)

ภาพที่ หน้า


4.5 คลาสไดอะแกรมของ วิชา และห้อง..... 32
4.6 ER ไดอะแกรมของ วิชา และห้อง..... 32
4.7 คลาสไดอะแกรมแสดงความสัมพันธ์ของ คลาสลงทะเบียน..... 33
4.8 ER ไดอะแกรมแสดงความสัมพันธ์ของ คลาสลงทะเบียน..... 33
4.9 ER-Diagram ของระบบ..... 34
4.10 หน้าจอใส่ชื่อและรหัสผ่าน..... 40
4.11 หน้าจอหลัก..... 41
4.12 หน้าจอบันทึกเวลาเรียน..... 41
4.13 หน้าจอลงทะเบียน..... 42
4.14 ใบเสร็จรับเงิน..... 42
4.15 สมุดลงเวลาเรียน..... 43
4.16 หน้าจอลงการจองเวลาเรียน..... 43
4.17 รายงานการจองเวลาเรียน..... 44
4.18 หน้าจอแสดงตารางสอนของอาจารย์ตามอาจารย์..... 44
4.19 หน้าจอแสดงตารางเรียนของห้อง..... 45
4.20 รายงานตารางสอนของอาจารย์..... 46
4.21 หน้าจอข้อมูลนักเรียน..... 47
4.22 หน้าจอข้อมูลอาจารย์..... 47
4.23 หน้าจอข้อมูลวิชา..... 48
4.24 หน้าจอข้อมูลห้อง..... 48



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมา

ในปัจจุบันคอมพิวเตอร์ได้ถูกนำมาช่วยในการทำงานต่างๆ มากมาย เนื่องจากมีความสะดวกรวดเร็ว สามารถทำงานที่มีความซับซ้อนได้เป็นอย่างดี ซึ่งเป็นผลให้ผู้ใช้ลดขั้นตอนและเวลาในการทำงาน

วิทยาลัยดุริยางคศิลป์ มหาวิทยาลัยมหิดล ได้เล็งเห็นความสำคัญทางด้านการจัดระบบคอมพิวเตอร์ โดยได้มีการจัดทำระบบเกี่ยวกับการลงทะเบียนนักเรียน และการคิดค่าตอบแทนให้กับอาจารย์ ซึ่งสามารถรองรับการทำงานได้ในระดับหนึ่ง อย่างไรก็ตามยังคงมีปัญหาในด้านอื่นๆ ที่เกิดขึ้น เช่น การค้นหาและจัดตารางการเรียนการสอนให้กับอาจารย์ผู้สอนและนักเรียน ซึ่งเป็นเรื่องที่สลับซับซ้อนและยุ่งยากในการจัดการ รวมทั้งการคิดค่าตอบแทนให้กับอาจารย์มีการประมวลผลที่ล่าช้าเป็นอย่างมาก ทำให้เกิดแนวคิดที่จะนำระบบสารสนเทศมาใช้ในการแก้ปัญหาต่างๆ เหล่านี้ โดยจะมีการจัดทำระบบจัดตารางการเรียนการสอน การเช็คเวลาเรียน และการปรับปรุงระบบการลงทะเบียนเรียน เพื่อให้การทำงานมีประสิทธิภาพ และประสิทธิผล

โดยรายงานฉบับนี้จะทำการออกแบบระบบการจัดตารางเรียน การตรวจสอบเวลาเรียน การลงทะเบียน การเก็บประวัตินักเรียน ครูผู้สอน และพนักงาน และการคิดค่าตอบแทนให้กับครูผู้สอน ซึ่งจะใช้ UML (Unified Modeling Language) ในการออกแบบระบบทั้งหมด เพื่อให้อยู่ในรูปแบบของออบเจ็ค (Object)

## 1.2 วัตถุประสงค์

การพัฒนาระบบการจัดตารางสอนและเวลาเรียนนี้ ได้กำหนดวัตถุประสงค์ของโครงการไว้ดังนี้

1.2.1 เพื่อลดขั้นตอนและความยุ่งยากในการค้นหาข้อมูลนักเรียนและครูผู้สอน

1.2.2 เพื่อใช้ในการตรวจสอบวิชา วันและเวลาที่มีการจองเรียน ซึ่งช่วยในการจัดตารางเรียนให้ง่ายยิ่งขึ้น

1.2.3 เพื่อลดปัญหาที่ซับซ้อนในการคิดค่าตอบแทนให้กับครูผู้สอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1.3 ขอบเขตของการพัฒนาระบบงาน**

การพัฒนาระบบสารสนเทศ เพื่อจัดการเรียนการสอน เป็นการพัฒนาเพื่อรองรับระบบการทำงานต่างๆ ดังนี้

1.3.1 ระบบการจองวิชาเรียน กรณีที่นักเรียนสนใจที่จะลงทะเบียนเรียน ซึ่งในวิชานั้นๆ ยังไม่สามารถจัดให้กับผู้เรียนได้ เนื่องจากการเรียนแบ่งออกเป็น การเรียนเดี่ยว การเรียนคู่ หรือ การเรียนกลุ่ม จึงต้องมีการตรวจสอบอย่างละเอียดเกี่ยวกับระดับความสามารถในการสอนของครูผู้สอน วัน เวลา และห้องเรียน

1.3.2 ระบบการลงทะเบียนของนักเรียน ซึ่งจัดเก็บข้อมูลเกี่ยวกับประวัติของนักเรียน และการชำระเงินค่าเรียนของนักเรียน

1.3.3 ระบบการจัดตารางเรียน ซึ่งจะจัดเก็บข้อมูลเวลาเรียน เพื่อสะดวกต่อการค้นหา

1.3.4 ระบบบันทึกเวลามาเรียนของนักเรียน เพื่อใช้ในการตรวจสอบเวลาเรียน และคิดค่าตอบแทนอาจารย์

**1.4 ประโยชน์ที่คาดว่าจะได้รับ**

1.4.1 นักเรียนจะได้รับความสะดวกมากขึ้นในการที่จะทำการลงทะเบียน

1.4.2 เจ้าหน้าที่สามารถค้นหาค้นหาข้อมูลนักเรียนและครูผู้สอนได้อย่างรวดเร็ว

1.4.3 เจ้าหน้าที่สามารถจัดตารางเรียนง่ายยิ่งขึ้น

1.4.4 ลดความผิดพลาดในการคิดค่าตอบแทนอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

## UML (Unified Modeling Language)

### 2.1 UML (Unified Modeling Language)

#### 2.1.1 กระบวนการพัฒนาซอฟแวร์เชิงวัตถุ

หลักการสำคัญของแนวคิดเชิงวัตถุ

ในการศึกษาแนวคิดเชิงวัตถุจำเป็นต้องทำความเข้าใจกับคำศัพท์ทุกคำที่จะอธิบายถึงต่อไปนี้

1. ออบเจ็กต์ (Object) ในแนวคิดเชิงวัตถุ ออบเจ็กต์ หมายถึงทั้งสิ่งที่จับต้องได้และจับต้องไม่ได้ เช่น บัญชีลูกค้า ปุ่ม เมนู ฐานข้อมูล รวมถึงเหตุการณ์ต่างๆ ซึ่งเป็นนามธรรม เป็นต้น
2. เอ็นแคปซูเลชั่น (Encapsulation)ความหมายโดยทั่วไปของเอ็นแคปซูเลชั่น คือการจัดกลุ่มองค์ความคิดที่คล้ายคลึงกันเข้าเป็นหน่วยเดียวกันเพื่อทำการอ้างถึงด้วยชื่อเดียวกัน ในเชิงซอฟแวร์โปรแกรมเมอร์มักพบว่าชุดคำสั่งที่คล้ายคลึงกันปรากฏอยู่หลายครั้งในตัวโปรแกรม ดังนั้นจึงเกิดแนวความคิดในการจับรูปแบบการซ้ำดังกล่าวเข้ามาอยู่ในสิ่งเดียวกันอันเป็นที่มาของเอ็นแคปซูเลชั่นนั่นเอง สิ่งที่ตามมานอกจากความง่ายในการทำความเข้าใจตัวโปรแกรมแล้วยังเป็นช่วยประหยัดการใช้พื้นที่หน่วยความจำอีกด้วย ในแนวคิดเชิงวัตถุ เอ็นแคปซูเลชั่น หมายถึงการรวบรวมโอเปอเรชั่นและแอตทริบิวต์เข้าเป็นหน่วยเดียวกันเพื่อที่ว่าแอตทวิบิวต์สามารถถูกเปลี่ยนแปลงได้อย่างเหมาะสมโดยผ่านโอเปอเรชั่น และเราจะเรียกผลที่เกิดจากการใช้งานเอ็นแคปซูเลชั่นว่า การซ่อนข้อมูล (Information Hiding)
3. คลาส (Class) กล่าวได้ว่าคลาส คือแม่พิมพ์ที่ประกอบไปด้วย ชื่อของคลาสเอง แอตทริบิวต์ (Attribute(s)) และโอเปอเรชั่น (Operation(s)) สำหรับใช้ในการสร้าง (Instantiate) ออบเจ็กต์ โดยทุกๆ ออบเจ็กต์ที่ถูกสร้างขึ้นมาจากคลาสเดียวกันจะมีโครงสร้างและพฤติกรรมที่เหมือนกัน กล่าวคือ มีโอเปอเรชั่นและมีแอตทวิบิวต์ที่เหมือนกันหากแต่ค่าในแอตทริบิวต์อาจแตกต่างกัน ความจริงคลาสคือสิ่งที่เกิดขึ้นจากความคิดของการเอ็นแคปซูเลชั่นนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. อินสแทนซ์ (Instance) สำหรับออบเจ็กต์ที่ถูกสร้างขึ้น Instantiated จากคลาส A จะเรียกออบเจ็กต์ดังกล่าวว่า เป็นอินสแทนซ์ของคลาส A
5. แอตทริบิวต์ (Attribute) คือคุณสมบัติ (Property) ของออบเจ็กต์ หรืออาจใช้แสดงถึงสถานะ (State) ของออบเจ็กต์ ณ เวลาใดเวลาหนึ่ง เช่น คลาสรถยนต์จะมีแอตทริบิวต์ได้ดังต่อไปนี้ เช่น รุ่น สี ความเร็ว เป็นต้น
6. ตัวดำเนินการหรือโอเปอเรชั่น (Operation) หรือเทธอด (Method) หมายถึงฟังก์ชันพฤติกรรม (Behavior) หรือบริการ ที่ออบเจ็กต์สามารถกระทำให้ได้ เช่น คลาสรถยนต์อาจจะมีโอเปอเรชั่น สตาร์ทเครื่องยนต์ ออกวิ่ง เบรก ดับเครื่องยนต์ เปิดไฟยกเลี้ยว เป็นต้น ทั้งนี้จุดประสงค์หลักของโอเปอเรชั่นก็เพื่อใช้ในการจัดการกับแอตทริบิวต์นั่นเอง
7. ลายเซ็น (Signature) ประกอบไปด้วย ชื่อของโอเปอเรชั่น พารามิเตอร์ของโอเปอเรชั่น และชนิดของข้อมูลที่ถูกส่งคืนจากโอเปอเรชั่น เช่น boolean sampleOperation(int input_parameter) เป็นต้น
8. ข้อความหรือเมสเสจ (Message) ประกอบไปด้วยชื่อของโอเปอเรชั่นและค่าพารามิเตอร์ต่างๆ ของโอเปอเรชั่นดังกล่าว โดยปกติในการอธิบายถึงหลักการเชิงวัตถุเราจะใช้ประโยชน์ที่ว่า "...ส่งเมสเสจไปยังออบเจ็กต์" นั่นหมายถึงการเรียกใช้งานโอเปอเรชั่นของออบเจ็กต์นั่นเอง
9. อินเทอร์เฟซ (Interface) คือชุดของลายเซ็นทั้งหมดของคลาสใดคลาสหนึ่ง ซึ่งจะแสดงถึงสิ่งที่ออบเจ็กต์ของคลาสดังกล่าวสามารถกระทำหรือตอบสนองได้ แน่นอนว่าก่อนที่เราจะทำการส่งเมสเสจไปยังออบเจ็กต์ใดๆ เราจะเป็นต้องทราบถึงอินเทอร์เฟซของออบเจ็กต์ดังกล่าวเสียก่อน
10. การสืบทอดคุณสมบัติ (Inheritance) เป็นวิธีการในการสร้างคลาสใหม่จากคลาสเดิมที่มีอยู่ ทั้งนี้คลาสที่สร้างขึ้นใหม่จะมีวัตถุประสงค์ในการทำงานที่เฉพาะเจาะจงมากยิ่งขึ้น ในที่นี้คลาสที่ถูกถ่ายคุณสมบัติจะถูกเรียกว่า ซูเปอร์คลาส (Superclass, Parent class) หรือเรียกคลาสที่ได้รับการสืบทอดคุณสมบัติว่าซับคลาส (Subclass, Child class, Derived class)
11. ความสัมพันธ์ระหว่างคลาสหรือออบเจ็กต์ (Relationship)

แบ่งได้เป็นสามประเภทคือ

- *ความสัมพันธ์แบบพึ่งพา (Dependency)* เมื่อฝ่ายถูกพึ่งพิงมีการเปลี่ยนแปลงจะก่อให้เกิดผลกระทบกับอีกฝ่ายหนึ่งที่เป็นฝ่ายพึ่งพิง เช่น คลาสข้อมูลลูกค้ากับคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัสดุ เมื่อลูกค้ามีการเปลี่ยนแปลงที่อยู่ พัสดุควรถูกส่งไปยังผู้รับได้อย่างถูกต้อง นั่นคือคลาสพัสดุพึ่งพาคลาสข้อมูลลูกค้านั่นเอง

- *ความสัมพันธ์แบบสืบทอดคุณสมบัติ (Inheritance)* เช่น คลาสพ่อกับคลาสลูก
- *ความสัมพันธ์แบบเกี่ยวพัน (Association)* โดยจะมีการกำหนดถึงบทบาทของแต่ละฝ่ายที่รวมความสัมพันธ์กัน เช่น คลาสนักเรียนกับคลาสวิชาที่เปิดสอนอาจมีความสัมพันธ์แบบเกี่ยวพันในรูปแบบของการลงทะเบียน หรือคลาสเครื่องยนต์กับคลาสรถยนต์จะมีความสัมพันธ์แบบเกี่ยวพันในรูปแบบขององค์ประกอบกล่าวคือรถยนต์จะประกอบไปด้วยเครื่องยนต์ เป็นต้น

12. การเปลี่ยนรูป (Polymorphism) หมายถึงการเปลี่ยนรูปร่างของออบเจ็กต์หนึ่งๆ ซึ่งในเชิงโปรแกรมมั่งจะเป็นการที่ตัวแปรออบเจ็กต์ ของคลาสใดคลาสหนึ่งสามารถเปลี่ยนรูปแบบไปจากคลาสเดิมได้ เช่น การสร้างออบเจ็กต์ ของซับคลาสและกำหนดตัวแปรออบเจ็กต์ ดังกล่าวให้เป็นประเภทซูเปอร์คลาสจะทำให้ตัวแปรออบเจ็กต์ เปลี่ยนรูปเป็นออบเจ็กต์ ของซูเปอร์คลาสแทน นอกจากนี้ยังรวมถึงการที่โอเปอเรชั่นเดียวกันมีพฤติกรรมที่แตกต่างกันเมื่อถูกใช้กับออบเจ็กต์ที่เกิดจากคนละคลาสกัน กล่าวอีกนัยหนึ่งคือออบเจ็กต์ที่เกิดจากต่างคลาสกันสามารถที่จะมีปฏิกิริยาตอบสนองต่อโอเปอเรชั่นชื่อเดียวกันได้อย่างแตกต่างกัน ตัวอย่างเช่น คลาสรูปสี่เหลี่ยมกับคลาสสามเหลี่ยมที่ได้รับการสืบทอดคุณสมบัติมาจากคลาสรูปหลายเหลี่ยนคลาสเดียวกัน ทั้งคู่อาจมีโอเปอเรชั่นที่มีชื่อว่า **"draw()"** เหมือนกัน แต่เมื่อมีการเรียกใช้โอเปอเรชั่นดังกล่าวของออบเจ็กต์ที่สร้างจากคลาสรูปสี่เหลี่ยมจะมีเป็นการวาดรูปสี่เหลี่ยม ต่างกับการโอเปอเรชั่น draw() ของรูปสามเหลี่ยมซึ่งจะเป็นการวดรูปสามเหลี่ยม ในแหล่งข้อมูลหลายๆ ที่กล่าวไว้ว่าคุณสมบัติเด่นของการเขียนโปรแกรมเชิงวัตถุมีอยู่ด้วยกันสามประการเรียกว่า **PIE** ซึ่งย่อมาจากคำว่า **Polymorphism, Inheritance** และ **Encapsulation** นั่นเอง

กล่าวโดยสรุป ข้อดีของการประยุกต์ใช้แนวคิดเชิงวัตถุในการพัฒนาระบบซอฟแวร์มีดังต่อไปนี้

- **สนับสนุนการพัฒนาระบบที่ซับซ้อน (Problem Decomposition)**

ในการพัฒนาโปรแกรมรูปแบบเดิม นักพัฒนาโปรแกรมจะทำงานกันที่ระดับฟังก์ชัน กล่าวคือตั้งแต่เริ่มการพัฒนาจนจบ จะเป็นการสร้างปรับปรุงแก้ไขรายละเอียดของฟังก์ชันต่างๆ ภายในตัวโปรแกรม แต่สำหรับแนวคิดเชิงวัตถุ นักพัฒนาจะทำการวิเคราะห์ออกแบบระบบกันใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับของออบเจ็กต์ซึ่งประกอบด้วยทั้งข้อมูลและฟังก์ชันภายในแต่ละออบเจ็กต์ ดังนั้นจึงกล่าวได้ว่าการอาศัยแนวคิดเชิงวัตถุจะช่วยให้นักพัฒนาสามารถสร้างโปรแกรมได้ง่ายขึ้นมาก

- **สนับสนุนการนำกลับมาใช้งานซ้ำอีกครั้ง (Promotion of Reusability)**

เนื่องจากแต่ละคลาสหรือออบเจ็กต์ที่กำหนดขึ้นนั้นจะมีความสมบูรณ์อยู่ในตัวมันเองบนพื้นฐานแนวคิดของแต่ละออบเจ็กต์เอง รวมทั้งยังเป็นอิสระจากสภาพแวดล้อมอื่น ดังนั้นแต่ละคลาสจึงง่ายต่อการนำกลับมาใช้งานปรับปรุงเพิ่มเติม การนำกลับมาใช้งานอาจอยู่ในรูปแบบของการสืบทอดคุณสมบัติระหว่างออบเจ็กต์หรือการใช้งานซอฟแวร์คอมโพเนเนต์ก็เช่นกัน ตัวอย่างเช่น เราสามารถสร้างออบเจ็กต์จากคลาสปุ่มเพื่อไปใส่ไว้ในโปรแกรมต่างๆ ได้ตามต้องการ นั่นคือคลาสปุ่มสามารถถูกนำกลับมาใช้งานได้

- **ปรับปรุงแก้ไขและบำรุงรักษาง่าย (Easy for change and Extensibility)**

เนื่องจากข้อมูลและฟังก์ชันการทำงานที่เกี่ยวข้องกับออบเจ็กต์หนึ่งๆ จะถูกรวบรวมอยู่ที่เดียวกัน (Encapsulation) การทำงานภายในของแต่ละออบเจ็กต์จะไม่เกี่ยวข้องพึ่งพิงกับโค้ดที่อยู่ภายนอกออบเจ็กต์ ดังนั้นนักพัฒนาสามารถทำการแก้ไขปรับปรุงรายละเอียดภายในของแต่ละคลาสได้โดยไม่กระทบต่อส่วนที่เรียกใช้งานภายนอกแต่อย่างใด นอกจากนี้ในการขยายระบบก็สามารถทำได้ง่ายๆ โดยการสร้างออบเจ็กต์หรือคลาสเพิ่มเติมลงไปในตัวโปรแกรมนั่นเอง

การเข้าถึงแอตทริบิวต์และโอเปอเรชั่นของคลาส (Visibility) ซึ่งปกติจะแบ่งได้เป็น 3 รูปแบบ คือ *พับลิก (Public), ไพรเวต (Private)* และ *โปรเท็กเท็ด (Protected)* โดยแอตทริบิวต์หรือโอเปอเรชั่นที่มีการเข้าถึงแบบ ไพรเวตจะต้องถูกเข้าถึงผ่านโอเปอเรชั่นเท่านั้น สำหรับโอเปอเรชั่นที่มีการเข้าถึงแบบพับลิกสามารถถูกเรียกใช้งานจากส่วนใดๆ ไม่ว่าจะจากภายนอกหรือภายในคลาสได้โดยตรง ส่วนแอตทริบิวต์หรือโอเปอเรชั่นที่มีการเข้าถึงแบบโปรเท็กเท็ดโดยปกติจะมีเพียงคลาสตัวมันเองและคลาสอื่นๆ ที่เป็นซับคลาสบนคลาส Hierarchy เดียวกันเท่านั้นที่จะเข้าถึงได้ ขอให้พิจารณาตัวอย่างจากรูปที่ 1.1 ซึ่งแสดงการเข้าถึงแอตทริบิวต์ประเภทต่างๆ ผ่านโอเปอเรชั่นที่ล้อมรอบอยู่ โดยแสดงไว้ด้วยสัญลักษณ์ลูกศร มีข้อระวังอยู่ประการหนึ่งคือการพัฒนาโปรแกรมเชิงวัตถุที่ดีจะต้องไม่มีการการเข้าถึงแอตทริบิวต์โดยตรงที่ไม่ผ่านโอเปอเรชั่นใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.1.2 ขั้นตอนของกระบวนการพัฒนาซอฟแวร์เชิงวัตถุ**

วัตถุประสงค์หลักของทุกกระบวนการพัฒนาซอฟแวร์คือ การแปลงความต้องการของผู้ใช้ให้เป็นระบบที่มีคุณภาพและสามารถใช้งานได้งานได้จริง อีกทั้งยังช่วยลดระยะเวลาที่ต้องทำการเขียนโปรแกรมและแก้ไขข้อผิดพลาดให้น้อยลงอีกด้วย

กระบวนการที่จะอธิบายต่อไปนี้เป็นวิธีการทั่วไปในการพัฒนาซอฟแวร์เชิงวัตถุที่ประกอบไปด้วย 5 ขั้นตอนสำคัญคล้ายคลึงกับหลายๆ กระบวนการพัฒนาซอฟแวร์เชิงวัตถุที่มีชื่อเสียงดังได้กล่าวไป และมีชื่อเรียกโดยทั่วไปว่า **Traditional OO Method** นอกจากนี้เช่นเดียวกันกระบวนการอื่นๆ Traditional OO Method ยังมีคุณสมบัติของการวนซ้ำและการเพิ่มเติมในแต่ละรอบ (Incremental Iterative Approach) โดยในแต่ละขั้นตอนจะถูกเรียกว่า **เฟส (Phase)** ซึ่งผลลัพธ์ที่ได้จากการพัฒนาในแต่ละเฟสจะต้องถูกนำไปใช้ในการพัฒนาต่อในเฟสถัดไป และมีเพียงเฟสแรกที่จะถูกดำเนินการในช่วงเริ่มต้นของการพัฒนาครั้งเดียวและในสี่เฟสที่เหลือจะถูกทำซ้ำ (Iteration) เพื่อทำการขยายต่อเติมฟังก์ชันของระบบจนกว่าจะได้ระบบที่สมบูรณ์ นั่นคือในแต่ละรอบของการวนซ้ำ สิ่งที่ได้จะเป็นเพียงต้นแบบ (Prototype) อันประกอบไปด้วยฟังก์ชันบางอย่างสอดคล้องกับความต้องการของผู้ใช้ และสั่งนี้เองที่ฝ่ายพัฒนาควรทำการส่งมอบให้กับผู้ใช้พิจารณาเป็นระยะๆ ในช่วงของการพัฒนา และในความเป็นจริงสิ่งที่เกิดขึ้นตามมาซึ่งมักหลีกเลี่ยงไม่ได้คือความจำเป็นในการดำเนินการเฟสแรกมากกว่าหนึ่งครั้งอันเนื่องมาจากการเปลี่ยนแปลงความต้องการของผู้ใช้นั่นเอง

ลักษณะดังกล่าวได้ถูกแสดงไว้ในรูปต่อไปนี้ ซึ่งมีรูปแบบที่คล้ายคลึงกับก้นหอยอันเป็นลักษณะของวงจรชีวิตในการพัฒนาซอฟแวร์แบบ Boehm (Boemh's Spiral Model)

**1. การวิเคราะห์ความต้องการของระบบหรือผู้ใช้ (Requirement Analysis)**

โครงการที่ถือว่าประสบความสำเร็จจะต้องตอบสนองต่อความต้องการและความคาดหวังของผู้ใช้ได้อย่างครบถ้วนหรือมากกว่าอันเป็นจุดสำคัญของทุกกระบวนการพัฒนาซอฟแวร์ การวิเคราะห์ความต้องการของผู้ใช้เป็นการค้นหาขอบข่ายของระบบและเป็นการเตรียมข้อมูลด้านความสามารถของระบบ (System Functions) จากมุมมองภายนอกที่จะต้องถูกทำการพัฒนาโดยไม่คำนึงถึงรายละเอียดหรือกรรมวิธีทางเทคนิคต่างๆ

ในความเป็นจริงการเริ่มต้นเฟสแรกนี้จะต้องทำในรูปของการกำหนดเป็นข้อตกลง (Agreement) ด้วยการเจรจากันระหว่างผู้ใช้งานกับผู้พัฒนา ซึ่งฝ่ายผู้พัฒนาจะต้องบันทึกความต้องการของผู้ใช้อย่างละเอียดที่สุดเท่าที่จะเป็นไปได้ เรียกขั้นตอนย่อยนี้ว่า **User Requirement Elicitation** อันหมายถึงการสัมภาษณ์ผู้ใช้งานระบบถึงความต้องการต่างๆ เฟสนี้จำเป็นอย่างมากโดยเฉพาะในการพัฒนาระบบด้านธุรกิจ แต่มักไม่จำเป็นสำหรับการสร้างระบบเชิงวิจัยพัฒนาซึ่งมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป้าหมายที่ชัดเจนอยู่แล้ว สำหรับในส่วนที่ไม่สามารถลงรายละเอียดของความต้องการได้ก็ควรกำหนดเป้าหมายให้ชัดเจนในส่วนนั้นๆ เพื่อป้องกันความเข้าใจที่ไม่ตรงกันระหว่างผู้ใช้และผู้พัฒนา

นอกจากการค้นหาความสามารถที่ต้องมีอยู่ในตัวระบบแล้ว ผู้พัฒนายังจำเป็นต้องทำการสำรวจถึงความต้องการในแง่อื่นๆ ด้วย (Nonfunctional Requirements) ตัวอย่างเช่น ประสิทธิภาพช่วงเวลาในการตอบสนอง ส่วนอินเทอร์เฟซ การใช้งานร่วมกับระบบเดิม เป็นต้น อย่างไรก็ตามหากมองในภาพที่กว้างขึ้นจะพบว่าการดำเนินการพัฒนาระบบจะต้องเกี่ยวข้องกับงบประมาณ ระยะเวลาในการส่งมอบผลงานแต่ละช่วง การทำข้อสัญญา และการบริหารโครงการ เป็นต้น

**2. การวิเคราะห์ระบบ (Domain Analysis)**

การวิเคราะห์ระบบ (Domain Analysis) หรือ OOA (OO Analysis) เป็นการวิเคราะห์โครงสร้าง (Structure) และพฤติกรรม (Behavior) ของระบบที่จะทำการพัฒนาซึ่งจะถูกนำไปกำหนดรายละเอียดเชิงเทคนิคในเฟสการออกแบบและจะถูกสร้างจริงเป็นลำดับต่อไปในอิมพลีเมนต์เตชั่นเฟส ดังนั้นในเฟสนี้จะไม่มีการพิจารณาถึงประเด็นด้านเทคนิคและรายละเอียดในการอิมพลีเมนต์ระบบ กิจกรรมในเฟสนี้จะเป็นการทำความเข้าใจกับระบบที่กำลังจะพัฒนาในสองด้าน กล่าวคือ

- **ด้านการวิเคราะห์โครงสร้าง (Structure)** เป็นการทำความเข้าใจกับระบบที่กำลังพัฒนาเพื่อค้นหาคลาสและความสัมพันธ์ระหว่างคลาสในระบบ ซึ่งผลที่ได้เหล่านี้จะถูกนำไปอิมพลีเมนต์สร้างเป็นระบบขึ้นจริงในเฟสถัดๆ ไป
- **ด้านการวิเคราะห์พฤติกรรม (Behavior)** เนื่องจากออบเจ็กต์ที่เกิดจากคลาสต่างๆ ในโครงสร้างของระบบจะทำงานร่วมกันเพื่อให้ระบบโดยรวมสามารถตอบสนองต่อการใช้งานของผู้ใช้ได้ในทุกฟังก์ชัน กิจกรรมในระหว่างการปฏิบัติงานร่วมกันของแต่ละออบเจ็กต์จำเป็นต้องถูกทำความเข้าใจเพื่อนำไปอิมพลีเมนต์เป็นขั้นตอนการทำงานของแต่ละฟังก์ชันของระบบได้

**3. การออกแบบระบบ (design) หรือ OOD (OO Design)**

เป็นการคิดค้นวิธีแก้ไขปัญหาหรือพิจารณารายละเอียดเชิงเทคนิคเพื่อเตรียมที่จะอิมพลีเมนต์ระบบขึ้นจริง ซึ่งจะเป็นการนำผลการวิเคราะห์จากเฟสที่สองมาทำการแก้ไขเพิ่มเติมรายละเอียดเชิงเทคนิคเพื่อสามารถที่จะถูกนำไปสร้างขึ้นเป็นระบบซอฟแวร์จริงได้อย่างสมบูรณ์ตัวอย่างของกิจกรรมที่เกิดขึ้นในเฟสนี้คือ

- การเลือกใช้ภาษาโปรแกรมคอมพิวเตอร์ที่จะใช้ในการพัฒนาระบบ
- การออกแบบฐานข้อมูล โดยเฉพาะในส่วนที่เรียกว่า Conceptual Schema ซึ่งนิยมใช้ ER โมเดลเข้าช่วย
- การออกแบบสถาปัตยกรรมระบบ จำนวนเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การออกแบบลักษณะการเชื่อมต่อของเครือข่าย
- การนำกลับมาใช้งานของคลาสไลบรารีหรือซอฟแวร์คอมโพเนนต์ที่มีอยู่
- การออกแบบหน้าตาของตัวโปรแกรม (User Interface)
- รวมถึงการจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งานระบบ เช่น ระบบควรมีการจัดการอย่างไรหากผู้ใช้ป้อนข้อมูลผิดประเภท เป็นต้น

ดังนั้นในเฟสนี้จำเป็นต้องอาศัยความรู้ทางเทคนิคและเทคโนโลยีต่างๆ ที่มีอยู่เพื่อสามารถเลือกได้อย่างเหมาะสม

**4. การสร้างโปรแกรมระบบ (construction, Coding, Implementation)**

หลังจากที่ระบบถูกออกแบบไว้อย่างสมบูรณ์พร้อมที่จะถูกนำไปอิมพลีเมนต์จริงโดยโปรแกรมเมอร์ ในเฟสนี้สิ่งที่เกิดขึ้นจากการออกแบบทั้งหมดจะถูกแปลงไปสู่ระบบจริง โดยส่วนใหญ่กิจกรรมในเฟสนี้จะเป็นการสร้างโปรแกรมหรือการอิมพลีเมนต์ระบบอันเป็นขั้นตอนของ OOP (OO Programming) ที่คุ้นเคยกันดี ซึ่งจะต้องอาศัยความรู้ในตัวภาษาโปรแกรมที่ใช้ในการสร้างโค้ด ในขั้นตอนนี้จะถูกดำเนินการโดยทีมโปรแกรมเมอร์ภายหลังจากได้รับข้อมูลการออกแบบที่เพียงพอจากฝ่ายวิเคราะห์ออกแบบ

**5. การทดสอบระบบ (Testing)**

เป็นการทดสอบความถูกต้องของระบบที่พัฒนาเพื่อค้นหาข้อผิดพลาดเชิงเทคนิค และการตรวจสอบความสอดคล้องกับความต้องการที่ถูกระบุอยู่ในความต้องการของผู้ใช้งาน ทั้งนี้การค้นพบข้อผิดพลาดจะถือว่าเป็นความสำเร็จของการดำเนินงานในเฟสนี้มิใช่เป็นความล้มเหลวแต่อย่างใด นอกจากนี้ยังเป็นการประเมินความสมบูรณ์ของระบบว่าจำเป็นต้องทำการวิเคราะห์ออกแบบเพิ่มเติมอีกครั้งหรือไม่ โดยปกติในเฟสนี้จะต้องมีการจัดเตรียมข้อมูลที่จะใช้สำหรับการทดสอบและการประเมินผลลัพธ์เรียกว่า **Test Cases** ซึ่งจะใช้ในการตรวจสอบส่วนต่างๆ ของระบบในทุกแง่มุมของการทำงานทั้งหมดที่เป็นไปได้ แต่ละ Test Case จะประกอบไปด้วยกิจกรรมในการทดสอบการป้อนข้อมูลและการทดสอบผลลัพธ์รวมถึงความคาดหวังจากการทำงานของระบบว่าน่าจะเป็นที่พึงพอใจแก่ผู้ใช้งานจริงหรือไม่ ผลของการทดสอบจะถูกบันทึกลงรายงานการทดสอบซึ่งรวมถึงการบรรยายรายละเอียดข้อผิดพลาดที่ปรากฏเพื่อทำการแก้ไขต่อไปโดยโปรแกรมเมอร์

ในเฟสนี้สิ่งที่ทำให้นักพัฒนาปวดหัวอยู่เสมอคือข้อผิดพลาด (Error) ซึ่งแบ่งได้เป็นสองด้านได้แก่

1. *Functional* เป็นความผิดพลาดในระดับฟังก์ชันการทำงานระบบ ซึ่งมักเกิดขึ้นบ่อยครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. *Nonfunctional* เป็นความผิดพลาดที่ไม่เกี่ยวข้องกับฟังก์ชันการทำงานของระบบที่พัฒนาโดยตรง เช่น การที่ประสิทธิภาพการทำงานของระบบต่ำเกินไป ช่วงเวลาในการตอบสนองนานเกินไป เป็นต้น

### 2.1.3 อะไรคือภาษาโมเดลลิ่ง (Modeling Language)

ยูเมแอล เป็นภาษาสัญลักษณ์รูปภาพมาตรฐาน (Standard Modeling Language) สำหรับใช้ในการสร้างโมเดลเชิงวัตถุ การใช้งานภาษายูเอ็มแอลนอกจากจะต้องเข้าใจในแนวคิดเชิงวัตถุแล้วยังจำเป็นต้องมีพื้นฐานความเข้าใจเกี่ยวกับวิชวลโมเดลลิ่ง (Visual Modeling) ด้วยเช่นกัน

โมเดลลิ่ง เป็นวิธีการวิเคราะห์ออกแบบ (Analysis and Design) อย่างหนึ่งที่เน้นการใช้งานโมเดลเป็นหลัก ซึ่งโมเดลที่สร้างขึ้นมาจะสามารถช่วยให้เข้าใจปัญหาได้ง่ายขึ้น อีกทั้งยังสามารถนำโมเดลมาเป็นเครื่องมือในการสื่อสารถ่ายทอดความคิดกับบุคคลอื่นๆ ที่เกี่ยวข้องในโครงการได้ เช่น ลูกค้า นักวิเคราะห์ นักออกแบบ เป็นต้น ส่ววิชวลโมเดลลิ่งคือการโมเดลสิ่งที่ใช้สัญลักษณ์รูปภาพในการสร้างโมเดลของระบบที่จะพัฒนาเพื่อประโยชน์ที่คล้ายคลึงกันในการทำความเข้าใจกับความต้องการของลูกค้า การออกแบบระบบที่เป็นไปได้อย่างชัดเจนขึ้น และการบำรุงรักษาระบบที่ง่ายยิ่งขึ้น โมเดลเกิดขึ้นโดยการนำเสนอส่วนต่างๆ ของระบบแต่เพียงส่วนที่สำคัญโดยไม่คำนึงถึงรายละเอียดที่ไม่สำคัญซึ่งวิธีการนี้ได้ถูกใช้โดยเหล่าสถาปนิก นักแกะสลัก จิตรกร วิศวกร มาเป็นเวลานานหลายพันปี

เช่นเดียวกัน ในการพัฒนาระบบซอฟแวร์ที่ซับซ้อน นักพัฒนาจำเป็นต้องทำความเข้าใจกับมุมมองด้านต่างๆ ของระบบก่อนทำการพัฒนาจริง วิธีการหนึ่งในการลบล้างข้อจำกัดของมนุษย์ในการทำความเข้าใจกับระบบทั้งหมดในทุกด้านก็คือการสร้างโมเดลอันเปรียบเสมือนพิมพ์เขียวที่แสดงถึงภาพรวมของระบบทั้งหมด โมเดลที่สร้างขึ้นนี้จะต้องถูกวาดขึ้นด้วยสัญลักษณ์ที่แม่นยำ เน้นความสอดคล้องกับความต้องการของผู้ใช้งานระบบเป็นสำคัญ ในส่วนของรายละเอียดต่างๆ จะค่อยๆ ถูกเพิ่มเติมลงไปในตัวโมเดล และในที่สุดโมเดลก็จะถูกนำไปอิมพลีเมนต์ขึ้นเป็นระบบจริง

ผลที่ได้จากการทำโมเดลลิ่งก็คือโมเดลซึ่งจะเกิดขึ้นในแต่ละขั้นตอนของการพัฒนาระบบ เราเรียกโมเดลเหล่านี้ว่า**โมเดลผลลัพธ์** อันได้แก่

- ขั้นตอนการวิเคราะห์ความต้องการของผู้ใช้งานระบบ โมเดลที่ได้คือ **Requirement Analysis Model** ซึ่งก็คือ **Requirement Specification** นั่นเอง
- ขั้นตอนการวิเคราะห์ตัวระบบ โมเดลที่ได้คือ **Analysis Model**
- ขั้นตอนการออกแบบระบบ โมเดลที่ได้คือ **Design Model**
- ขั้นตอนการพัฒนาระบบ โมเดลที่ได้คือตัวโปรแกรม

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามโมเดลแรกนี้จะถูกสร้างขึ้นด้วยภาษายูเอ็มแอล จากภาพหลังจากได้โมเดลผลลัพธ์กระบวนการถัดไปคือ การใช้เครื่องมือในการเขียนภาษาโปรแกรมต่างๆ ทำการสร้างระบบซอฟแวร์ขึ้นจริงโดยใช้โมเดลผลลัพธ์ที่สร้างขึ้นทั้งหมด

กล่าวโดยสรุปการทำโมเดลลิ่งจะก่อให้เกิดประโยชน์หลายอย่างดังได้กล่าวไป ซึ่งในความเป็นจริงแล้วเราพบว่าการสร้างโมเดล เช่น โมเดลบ้าน ผู้ออกแบบสามารถนำไปคุยกับลูกค้าเพื่อแสดงถึงแนวความคิดในการออกแบบได้อย่างชัดเจนและง่ายกว่าการอธิบายด้วยวิธีการอื่น ทั้งนี้เนื่องจากวัตถุประสงค์หลักของโมเดลก็คือช่วยให้สามารถทำความเข้าใจกับปัญหาอันนำมาซึ่งวิธีการแก้ไขปัญหานั่นเอง นอกจากนี้การสร้างโมเดลยังช่วยลดค่าใช้จ่ายในการพัฒนาได้เป็นอย่างมาก กว่าคือตัวโมเดลจะถูกทดลองสร้างขึ้นก่อนการสร้างระบบจริงเพื่อศึกษาหรือค้นหาข้อบกพร่องรวมถึงการแก้ไขเปลี่ยนแปลงส่วนต่างๆ ซึ่งจะเป็นไปได้ไม่ง่ายและสิ้นเปลืองเป็นอย่างมากหากต้องทำกับระบบจริงที่พัฒนาขึ้นแล้ว

ถึงจุดนี้อาจสรุปเบื้องต้นได้ว่ายูเอ็มแอลเป็นภาษาที่ใช้ในการสร้างโมเดลเชิงวัตถุอันมีพื้นฐานอยู่บนหลักการวิชวลโมเดลลิ่งและหลักการเชิงวัตถุนั่นเอง

### 2.1.4 ความเป็นมาของภาษายูเอ็มแอล

ยูเอ็มแอลถูกเริ่มต้นครั้งแรกในปี 1994 ที่บริษัท Rational Software โดย Grady Booch และ James Rumbaugh วัตถุประสงค์เบื้องต้นในการร่วมงานกันระหว่างทั้งสองคนเป็นการพัฒนากระบวนการพัฒนาซอฟแวร์เชิงวัตถุที่เป็นหนึ่งเดียวกัน (Unified Method) โดยนำเอาวิธีของแต่ละคนคือวิธีของ Booch และวิธี OMT มารวมกันและปรับปรุงใหม่ ต่อมาในปี 1995 Ivar Jacobson ผู้พัฒนากระบวนการ OOSE หรือ Objectory ได้เข้าร่วมกับโครงการดังกล่าวซึ่งในครั้งนี้เป็นการสร้างภาษาโมเดลขึ้นใหม่เรียกว่า **Unified Modeling Language (UML)** และทั้งสามขนานนามตังเองว่า **Three Amigos**

เนื่องจากวิธีการพัฒนาซอฟแวร์เชิงวัตถุของทั้งสามมีชื่อเสียงอยู่แล้วในช่วงเวลานั้น ดังนั้นภาษายูเอ็มแอลที่ถูกพัฒนาขึ้นใหม่นี้จึงกลายมาเป็นที่นิยมใช้อย่างแพร่หลายเช่นกัน ในปี 1996 มีหลายบริษัทที่ขอเข้าร่วมในการพัฒนาภาษายูเอ็มแอลซึ่งได้แก่ บริษัทดิจิตอลอีควิปเมนต์ บริษัทเอชพี บริษัทไอบีเอ็ม รวมทั้งบริษัทออราเคิล บริษัทไมโครซอฟต์ และบริษัทอื่นๆ อีกหลายบริษัท ต่อมาในปี 1997 ยูเอ็มแอลรุ่นที่ 1.1 ก็ได้ถูกเสนอให้กับหน่วยงาน OMG (Object Management Group) ซึ่งได้ถูกกำหนดให้ภาษาโมเดลมาจรฐานในที่สุด จากนั้นภาษายูเอ็มแอลถูกพัฒนาต่อโดย OMG ซึ่งในปัจจุบันยูเอ็มแอลที่ถูกเผยแพร่ออกสู่สาธารณชนคือรุ่นที่ 1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.1.5 ข้อดีของยูเอ็มแอล**

ยูเอ็มแอลมีข้อดีหลายประการดังต่อไปนี้

1. เป็นภาษารูปภาพมาตรฐาน (Standard Visual Modeling Language) หรือภาษาสากลที่ใช้ในการพัฒนาซอฟแวร์เชิงวัตถุและสามารถใช้ในการแลกเปลี่ยนโมเดลได้อย่างสื่อความหมายรวมถึงการจัดสร้างเอกสารการวิเคราะห์ออกแบบระบบ โดยเฉพาะในการสร้างระบบขนาดใหญ่ซึ่งต้องอาศัยการทำงานเป็นทีม สมาชิกในทีมจำเป็นต้องมีการประสานงานกันเป็นอย่างดีในทิศทางเดียวกัน การประยุกต์ใช้ยูเอ็มแอลจะทำให้ผลของการวิเคราะห์ออกแบบระบบในขั้นตอนต่างๆ สามารถถูกแลกเปลี่ยนระหว่างผู้ร่วมงานภายในทีมด้วยกันได้โดยแต่ละฝ่ายจะสามารถทำความเข้าใจยูเอ็มแอลโมเดลอย่างรวมเร็วและตรงกัน
2. สามารถนำเสนอและสนับสนุนหลักการเชิงวัตถุได้อย่างครบถ้วนชัดเจน ทำให้นักพัฒนาระบบสามารถทำความเข้าใจกับปัญหาและค้นพบวิธีแก้ไขได้อย่างรวดเร็วและง่ายยิ่งขึ้น
3. ไม่ผูกติดกับภาษาโปรแกรมภาษาใดภาษาหนึ่ง กล่าวคือโมเดลที่ถูกสร้างขึ้นจากภาษามาตรฐานยูเอ็มแอลนี้ สามารถถูกแปลงไปเป็นระบบจริงที่ถูกสร้างขึ้นด้วยภาษาโปรแกรมเชิงวัตถุใดๆ ก็ได้
4. เป็นภาษาที่ง่ายต่อการทำความเข้าใจ ผู้ที่ทำการศึกษาหรือนำไปใช้งานไม่จำเป็นต้องมีความรู้อื่นใดนอกจากแนวคิดเชิงวัตถุ ไม่ว่าจะเป็นความรู้ด้านการคำนวณหรือความรู้ด้านอื่นๆ ก็ตาม
5. สามารถถูกแปลงเป็นภาษาที่ใช้ในการสร้างระบบขึ้นจริงได้อย่างอัตโนมัติ จึงเป็นการช่วยลดภาระ เวลา และค่าใช้จ่ายการพัฒนาระบบได้เป็นอย่างมาก ในปัจจุบันมีเครื่องมือพัฒนาที่มีความสามารถดังกล่าวมากมาย
6. สนับสนุนการขยายปรับปรุงระบบ เนื่องจากการทำงานกับภาษายูเอ็มแอลเป็นการทำงานที่ระดับแนวคิดเชิงวัตถุและวิธีการแก้ปัญหาเป็นสำคัญ การเพิ่มเติมแก้ไขระบบสามารถกระทำได้กับโมเดลก่อนลงมือพัฒนาเพิ่มเติมจริง ซึ่งแน่นอนว่าจะง่ายกว่าการเริ่มต้นทำการเปลี่ยนแปลงที่ซอร์สโค้ด
7. ในขณะเดียวกันยูเอ็มแอลยังถูกใช้ในการบันทึกความคิดของนักพัฒนาในลักษณะของเอกสารที่พร้อมจะถูกนำมาทำความเข้าใจหรือสานต่ออีกครั้งได้อย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.1.6 องค์ประกอบของยูเอ็มแอล**

องค์ประกอบของตัวภาษายูเอ็มแอลมี 3 ส่วนใหญ่ๆ คือ

- **สัญลักษณ์ทั่วไป (Things)** คือสัญลักษณ์พื้นฐานที่ถูกใช้งานในการสร้างไดอะแกรมยูเอ็มแอลต่างๆ โดยแบ่งเป็นหมวดย่อยๆ ดังนี้
  - *หมวดโครงสร้าง (Structural)* ได้แก่ ยูสเคส คลาส อินเทอร์เฟซ คอมโพเนนต์ คอลแลบอเรชั่น และโหนด
  - *หมวดพฤติกรรม (Behavioral)* คือส่วนที่เป็นไดนามิอของยูเอ็มแอล ซึ่งได้แก่ อินเตอร์แอ็กชั่น สเตตแมชชีน
  - *หมวดการจัดกลุ่ม (Grouping)* เพื่อใช้ในการรวบรวมองค์ประกอบต่างๆ ในโมเดลให้เหมาะสม ได้แก่ แพ็กเกจ
  - *หมวดคำอธิบายประกอบ (Annotational)* ได้แก่ โน้ต (Note)
- **ความสัมพันธ์ (Relationships)** มี 3 ชนิดคือ
  - *ความสัมพันธ์แบบพึ่งพา (Dependency Relationship)*
  - *ความสัมพันธ์แบบเกี่ยวพัน (Association Relationship)*
  - *ความสัมพันธ์แบบเจเนอรัลไลเซชั่น (Generalization Relationship)* หรือความสัมพันธ์แบบไม่เจาะจง ได้แก่ ความสัมพันธ์แบบสืบทอดคุณสมบัติ (Inheritance)
- **ไดอะแกรมต่างๆ (Diagrams)**

ในส่วนของไดอะแกรมจะประกอบไปด้วย 8 ไดอะแกรม ให้เลือกใช้ตามความเหมาะสม โดยในแต่ละไดอะแกรมจะเปรียบเสมือนมุมมองในด้านต่างๆ ของระบบที่กำลังพัฒนาซึ่งจะช่วยให้การวิเคราะห์ออกแบบเป็นไปได้อย่างมีประสิทธิภาพและง่ายดายมากยิ่งขึ้น

1. *ยูสเคสไดอะแกรม (Use Case Diagram)* ใช้ในการโมเดลฟังก์ชันการทำงานของระบบ
2. *คลาสไดอะแกรม (Class Diagram)* ใช้ในการโมเดลคลาสต่างๆ ที่จำเป็นในระบบ
3. *แอ็กทิวิตี้ไดอะแกรม (Activity Diagram)* มีหลักการเดียวกับโฟลว์ชาร์ด (Flowchart)
4. *สเตตชาร์ตไดอะแกรม (Statechart Diagram)* ใช้สำหรับแสดงถึงสถานะของออบเจ็กต์ในระหว่างการทำงาน
5. *คอลแลบอเรชั่นไดอะแกรม (Collaboration Diagram)* ใช้แสดงการทำงานร่วมกันของออบเจ็กต์ในระบบ
6. *ซีเควนซ์ไดอะแกรม (Sequence Diagram)* ใช้ในการโมเดลกิจกรรมต่างๆ ที่เกิดขึ้นกับออบเจ็กต์ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. *คอมโพเนนต์ไดอะแกรม (Component Diagram)* ใช้สำหรับสร้างโมเดลของคอมโพเนนต์ในระบบ
8. *ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)* ใช้แสดงการติดตั้งใช้งานส่วนประกอบต่างๆ ของระบบ

**2.2 เครื่องมือที่ใช้พัฒนา**

เครื่องมือที่ใช้พัฒนาระบบการจัดการตารางสอนของโรงเรียนดนตรีนี้ คือ VB.NET หรือ VisualBasic.NET ซึ่งเป็นการนำเอา Visual Basic 6 มาพัฒนาด้วยการเพิ่มเติมความสามารถของเทคโนโลยี .NET Framework เข้าไป เพื่อทำให้ VB.NET เป็นภาษาคอมพิวเตอร์ที่สามารถพัฒนาโปรแกรมที่ใช้งานอยู่บน Web และบนระบบปฏิบัติการ Windows ได้ และยังทำการสนับสนุนการเขียนโปรแกรมในลักษณะของ Object Oriented เพิ่มขึ้นด้วย เทคโนโลยี .NET ที่บริษัท Microsoft พัฒนาขึ้นมีจุดมุ่งหมายเพื่อสนับสนุนแนวคิดแบบ Reusable Code และช่วยลดช่องว่างในความแตกต่างด้านต่างๆ ที่เกิดขึ้นในการพัฒนาโปรแกรม เช่นความแตกต่างของภาษาที่ใช้พัฒนา ความแตกต่างระหว่างแนวคิดที่ใช้พัฒนาโปรแกรมที่อยู่บน Windows และบน Web เป็นต้น ดังนั้นไม่ว่าภาษาใดก็ตามที่ทำงานอยู่ภายใต้ .NET Framework จึงสามารถใช้ทรัพยากรต่างๆ ของ .NET Framework ได้เช่นเดียวกันทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# ขั้นตอนการดำเนินงาน

การวิเคราะห์และออกแบบระบบการจัดการตารางสอนและเวลาเรียน นี้จะศึกษาขั้นตอนการทำงานและปัญหาอุปสรรคที่เกิดขึ้นของระบบงานเดิมที่ใช้อยู่ในปัจจุบัน จากนั้นจะเป็นการวิเคราะห์และออกแบบระบบงานใหม่โดยนำเสนอความสัมพันธ์ของแต่ละขั้นตอน โดยใช้ โมเดลมาตรฐานของ UML โดยรายละเอียดจะได้กล่าวถึงต่อไป

## 3.1 การวิเคราะห์ระบบงาน

ลักษณะการเรียนของวิทยลัยดุริยางคศิลป์มหาวิทยาลัยมหิดลมีลักษณะเป็นการที่นักเรียนสามารถเลือกเรียนปฏิบัติเครื่องดนตรีได้ตามความสนใจ และตามเกณฑ์อายุของหลักสูตรนั้นๆ กำหนด นักเรียนสามารถเลือกเวลาเรียนได้ตามสะดวก แต่ทั้งนี้ต้องสอดคล้องกับความสะดวกของอาจารย์ด้วย ซึ่งนักเรียนที่มีประสบการเรียนมาแล้วก็สามารถทำการทดสอบ (Audition) เพื่อที่จะจัดระดับชั้นเรียนที่เหมาะสมได้

ในการลงทะเบียน 1 ภาคเรียนมี 12 ครั้ง หรือ 3 เดือน การเรียนปกติ เรียนสัปดาห์ละ 1 ครั้ง ครั้งละ 60 หรือ 30 นาที ตามประเภทของการเรียน การชำระค่าเล่าเรียนเป็นภาคเรียน และค่าลงทะเบียนแรกเข้าของนักศึกษาใหม่ 100 บาท ซึ่ง ระบบการทำงานต่างของวิทยลัยดุริยางคศิลป์มหาวิทยาลัยมหิดลมีขั้นตอนการทำงานดั้งนี้

**3.1.1 การจองวิชาเรียน** คือการที่นักเรียนไม่สามรถลงทะเบียนได้เนื่องจากไม่มีเวลา หรืออาจารย์ที่ต้องการ ซึ่งมีขั้นตอนการทำงานดังนี้

3.1.1.1 กรณีนักเรียนเก่ากลับมาเรียนใหม่ หรือ นักเรียนใหม่ที่มาเรียนวิชาพื้นฐาน

- นักเรียนทำการตรวจสอบเวลาเรียนกับเจ้าหน้าที่
- ถ้าไม่มีเวลาที่ต้องการก็ทำการจองช่วงเวลาเรียนที่ต้องการ

3.1.1.2 กรณีนักเรียนใหม่ ที่ต้องทำการทดสอบความรู้

- ทำการทดสอบความรู้
- อาจารย์ผู้ทดสอบทำการกำหนดวิชาเรียนและระดับการเรียนโดยดูจากผลการสอบ
- ตรวจสอบเวลาเรียนที่ต้องการ
- ถ้าไม่มีเวลาที่ต้องการก็ทำการจองช่วงเวลาเรียนที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.1.2 การลงทะเบียนของนักเรียน** คือนักเรียนต้องมาทำการบันทึกเวลาเรียน ซึ่งจะประกอบด้วยการกำหนด วิชา วัน เวลา อาจารย์ และห้องเรียน ก่อน แล้วค่อยไปจ่ายเงินลงทะเบียน ซึ่งจะมีขั้นตอนต่างๆ ดังต่อไปนี้

- นักเรียนทำการตรวจสอบเวลาเรียนกับเจ้าหน้าที่
- ถ้าสามารถเลือกเวลาเรียนที่ต้องการได้ก็ทำการบันทึกเวลาเรียนกับเจ้าหน้าที่
- จ่ายเงินค่าลงทะเบียน
- รับใบเสร็จค่าลงทะเบียน และ สมุดลงเวลาเรียน

**3.1.3 การจัดตารางเรียน** คือการนำบันทึกเวลาเรียนในขั้นตอนการลงทะเบียนมาจัด ตารางเรียนว่า อาจารย์คนไหนสอนวิชาอะไร เวลาอะไร วันไหน เริ่มเมื่อไหร่ถึงเมื่อไหร่ โดยมีขั้นตอนต่างๆ ดังนี้

- จากการบันทึกเวลาเรียนของขั้นตอนการลงทะเบียน จะเก็บบันทึกเวลาเรียนเอาไว้
- พนักงานเลือกดูรายการสอนของอาจารย์แต่ละคน โดยระบุชื่ออาจารย์ และวิชา

**3.1.4 การบันทึกเวลามาเรียนของนักเรียน** เพื่อใช้ในการตรวจสอบเวลาเรียน และคิดค่าตอบแทนอาจารย์ มีขั้นตอนการทำงานดังนี้

- นักเรียนนำสมุดลงเวลาเรียนมาให้พนักงานเพื่อทำการลงเวลามาเรียน
- พนักงานประทับตราแล้วบันทึกข้อมูลลงในคอมพิวเตอร์

## 3.2 การออกแบบระบบงาน

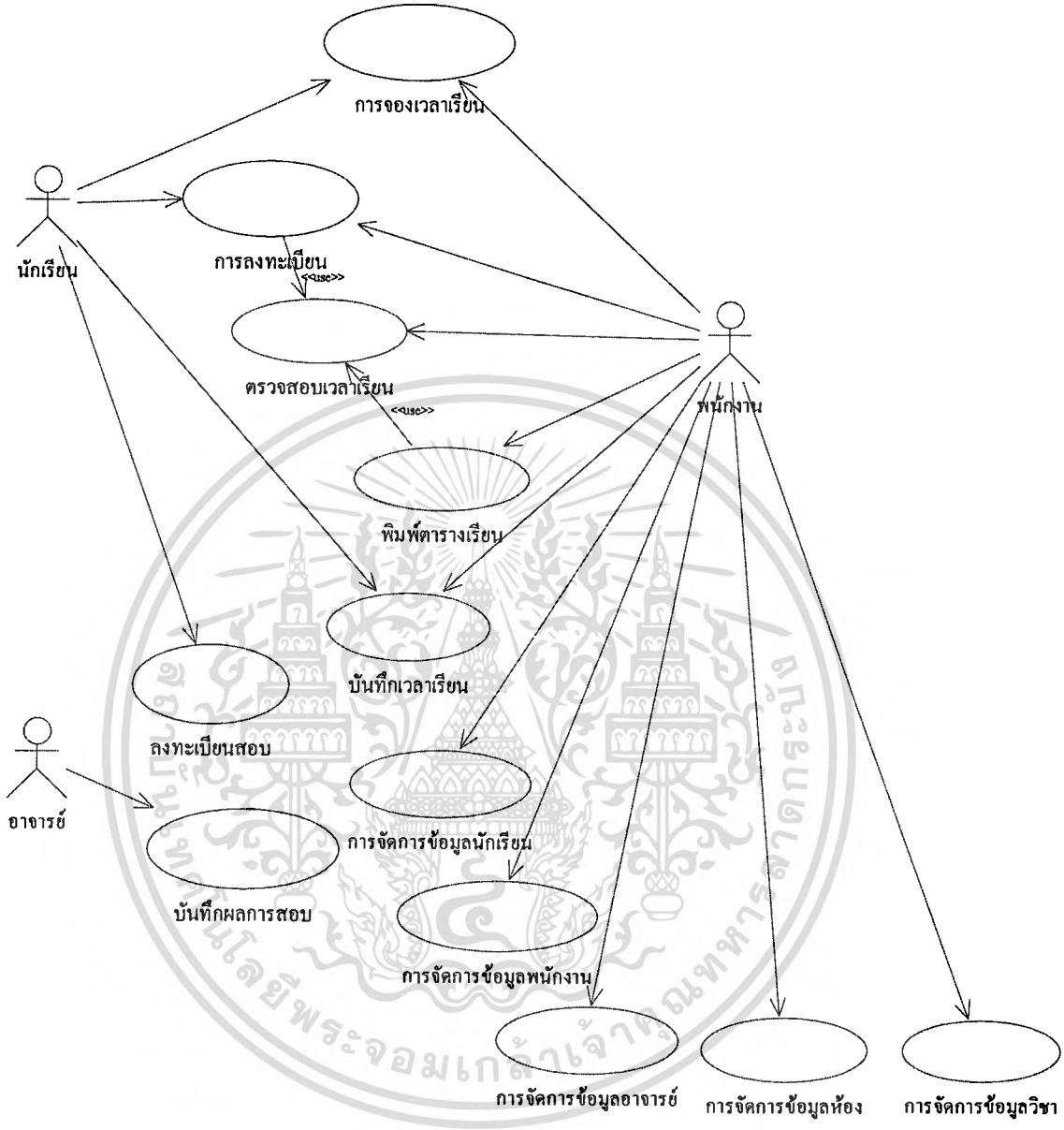
จากการวิเคราะห์ระบบงานของวิทยาลัยดุริยางคศิลป์มหาวิทยาลัยมหิดล นำขั้นตอนการทำงานต่าง ๆ มาวิเคราะห์ระบบโดยใช้ Use Case Diagram, Sequence Diagram และ Class Diagram เพื่อแสดงขั้นตอนการทำงาน และ การออกแบบการทำงานของระบบ

**3.2.1 Use Case Diagram** จากการวิเคราะห์ระบบงานของวิทยาลัยดุริยางคศิลป์มหาวิทยาลัยมหิดล นั้นพบว่า มีผู้ที่เกี่ยวข้องกับการทำงานของระบบ คือ พนักงาน นักเรียน และอาจารย์ ซึ่งเราจะนำมาเป็น Actor ของระบบ และจากขั้นตอนการทำงานต่างๆที่เกิดขึ้นเราสามารถนำมาเขียนเป็น Use Case Diagram ได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.





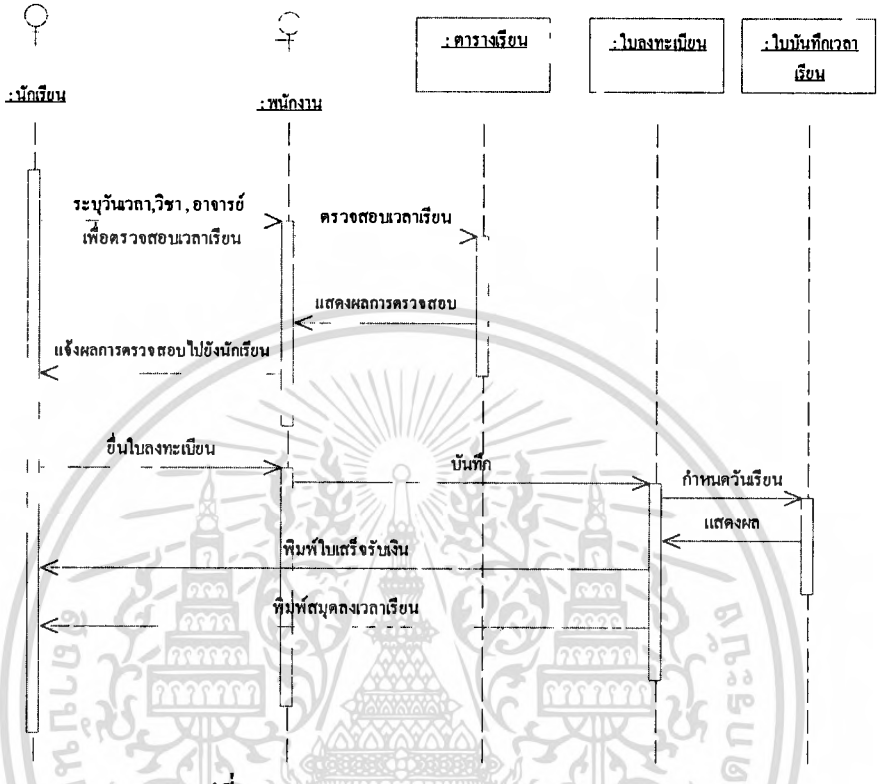

**รูปที่ 3.1** Use Case Diagram แสดงการทำงานของระบบ

### 3.2.2 Interaction Diagram

ในการออกแบบการแสดงเหตุการณ์ต่างๆที่เกิดขึ้นใน Use Case Diagram แต่ละอันนั้นเราจะใช้ Sequence diagram, Activity diagram และ StateChart diagram ในการแสดงเหตุการณ์ต่างที่เกิดขึ้น

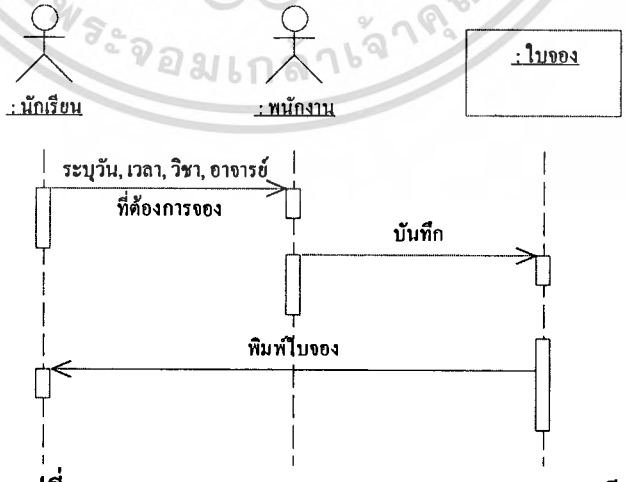
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2.2.1 Sequence diagram ของ Use case ลงทะเบียน** คือขั้นตอนการทำงานของระบบลงทะเบียนตามที่ได้ทำการวิเคราะห์ระบบของวิทยาลัยดุริยางคศิลปมหาวิทยาลัยมหิดล




**รูปที่ 3.2** Sequence diagram ของ Use case ลงทะเบียน

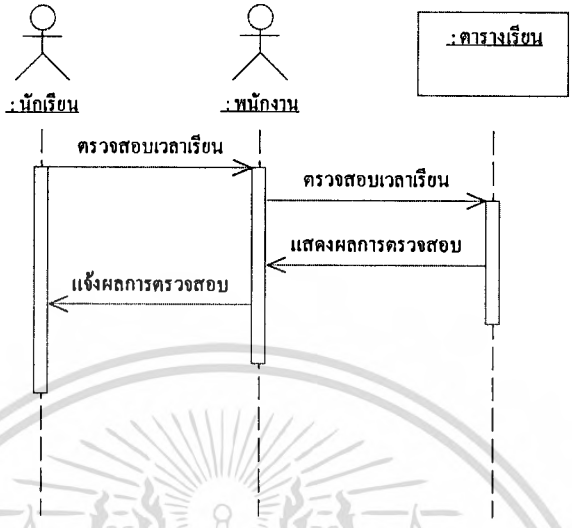
**3.2.2.2 Sequence diagram ของ Use case จองเวลาเรียน**




**รูปที่ 3.3** Sequence diagram ของ Use Case จองเวลาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.3 Sequence diagram ของ Use case ตรวจสอบเวลาเรียน



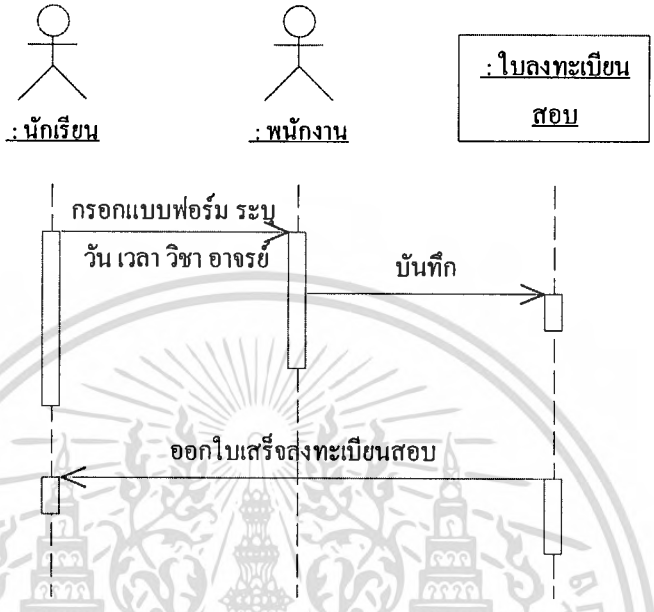

**รูปที่ 3.4** Sequence diagram ของ Use case ตรวจสอบเวลาเรียน

### 3.2.2.4 Sequence diagram ของ Use case พิมพ์ตารางเรียน

**รูปที่ 3.5** Sequence diagram ของ Use Case พิมพ์ตารางเรียน

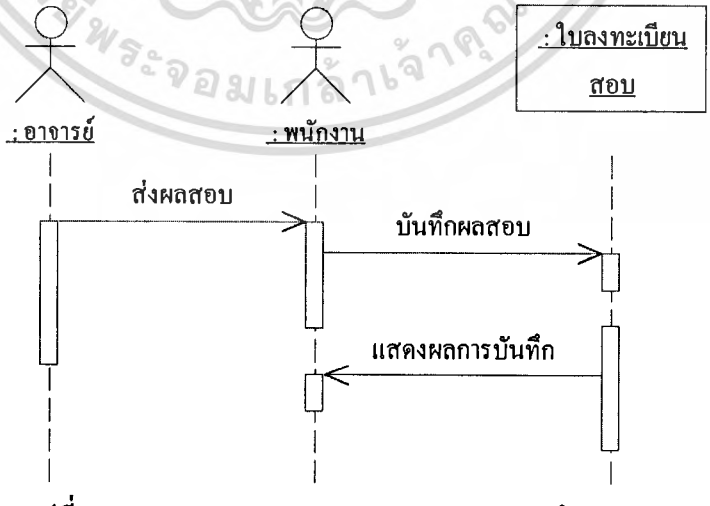
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.5 Sequence diagram ของ Use case ลงทะเบียนสอบ




**รูปที่ 3.6** Sequence diagram ของ Use case ลงทะเบียนสอบ

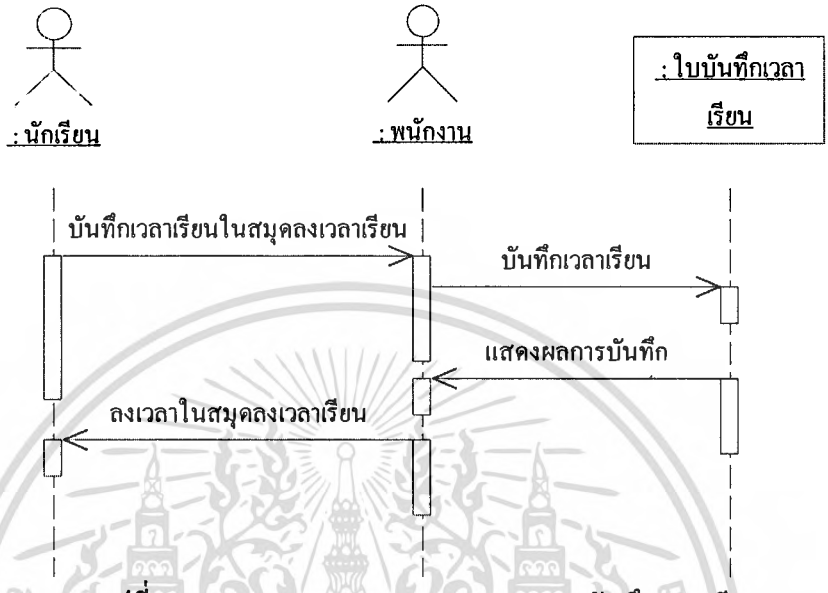
### 3.2.2.6 Sequence diagram ของ Use case บันทึกผลการสอบ




**รูปที่ 3.7** Sequence diagram ของ Use case บันทึกผลการสอบ

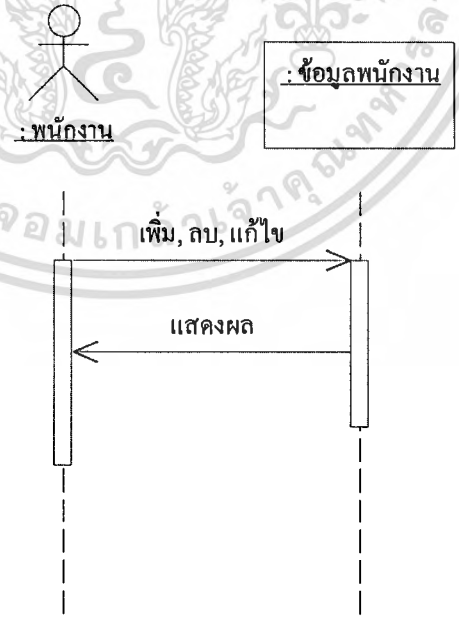
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.7 Sequence diagram ของ Use case บันทึกเวลาเรียน




**รูปที่ 3.8** Sequence diagram ของ Use case บันทึกเวลาเรียน

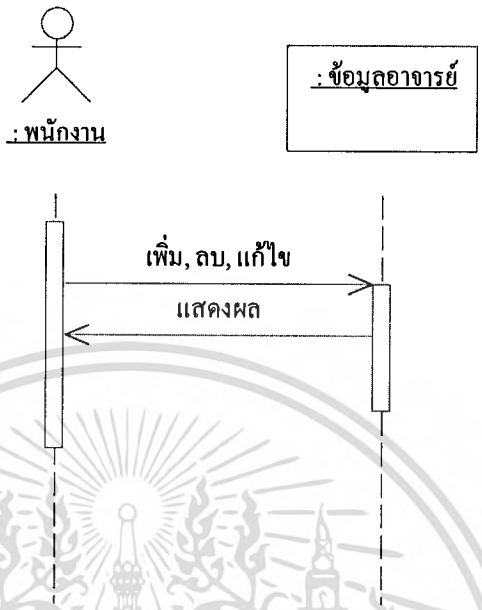
### 3.2.2.8 Sequence diagram ของ Use case การจัดการข้อมูลพนักงาน




**รูปที่ 3.9** Sequence diagram ของ Use case การจัดการข้อมูลพนักงาน

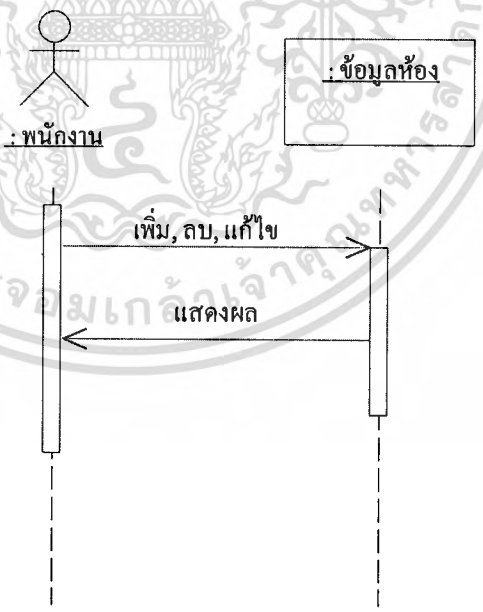
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2.2.9 Sequence diagram ของ Use case การจัดการข้อมูลอาจารย์**




**รูปที่ 3.10** Sequence diagram ของ Use case การจัดการข้อมูลอาจารย์

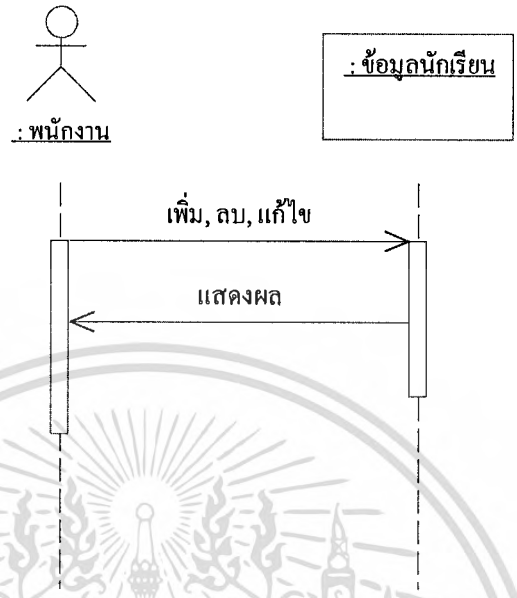
**3.2.2.10 Sequence diagram ของ Use case การจัดการข้อมูลห้อง**




**รูปที่ 3.11** Sequence diagram ของ Use case การจัดการข้อมูลห้อง

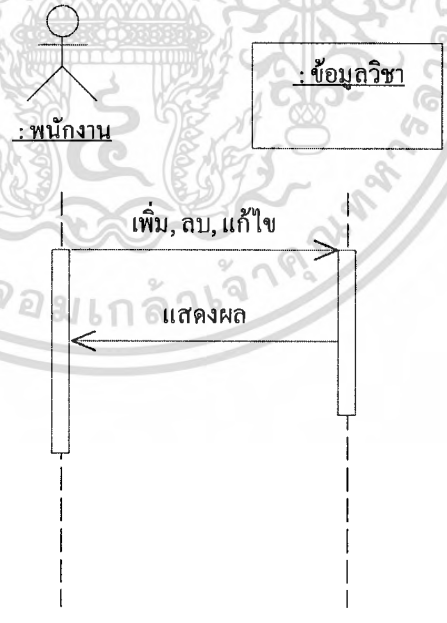
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.11 Sequence diagram ของ Use case การจัดการข้อมูลนักเรียน




**รูปที่ 3.12** Sequence diagram ของ Use case การจัดการข้อมูลนักเรียน

### 3.2.2.12 Sequence diagram ของ Use case การจัดการข้อมูลวิชา




**รูปที่ 3.13** Sequence diagram ของ Use case การจัดการข้อมูลวิชา

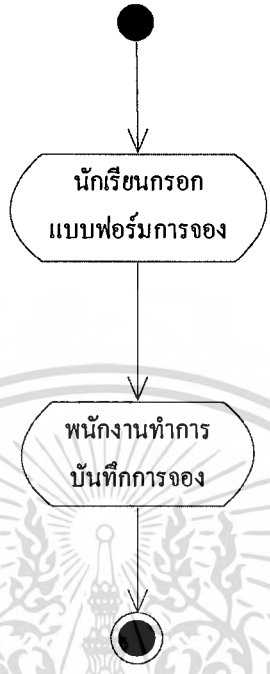
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.13 Activity diagram ของ Use case การลงทะเบียน

**รูปที่ 3.14** Activity diagram ของ Use case การลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.14 Activity diagram ของ Use case การจองเวลาเรียน



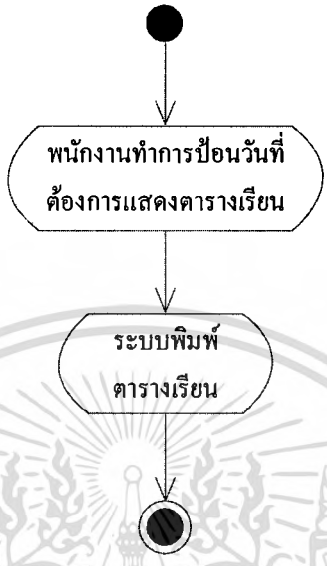

**รูปที่ 3.15** Activity diagram ของ Use case การจองเวลาเรียน

### 3.2.2.15 Activity diagram ของ Use case การตรวจสอบเวลาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

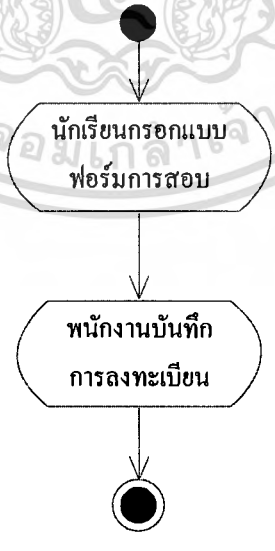
**รูปที่ 3.16** Activity diagram ของ Use case การตรวจสอบเวลาเรียน

### 3.2.2.16 Activity diagram ของ Use case พิมพ์ตารางเรียน




**รูปที่ 3.17** Activity diagram ของ Use Case พิมพ์ตารางเรียน

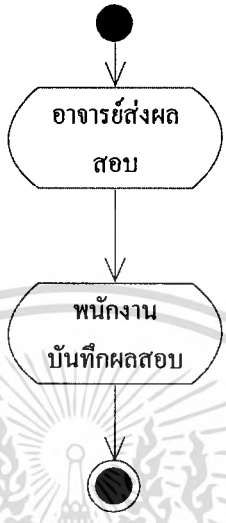
### 3.2.2.17 Activity diagram ของ Use case ลงทะเบียนสอบ




**รูปที่ 3.18** Activity diagram ของ Use case รลงทะเบียนสอบ

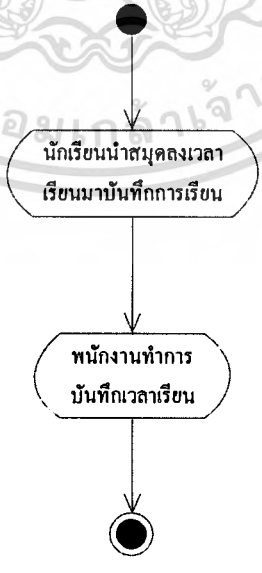
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.18 Activity diagram ของ Use case บันทึกผลการสอบ




**รูปที่ 3.19** Activity diagram ของ Use Case บันทึกผลการสอบ

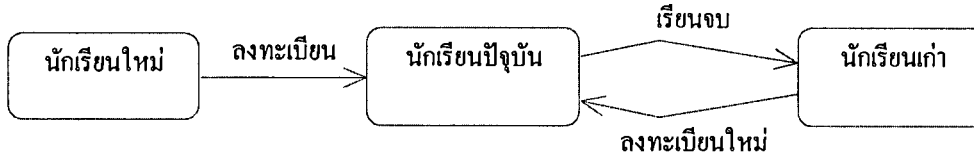
### 3.2.2.19 Activity diagram ของ Use case บันทึกเวลาเรียน




**รูปที่ 3.20** Activity diagram ของ Use case บันทึกเวลาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2.2.20 Startchart diagram ของนักเรียน**




**รูปที่ 3.21** Statechart diagram ของนักเรียน

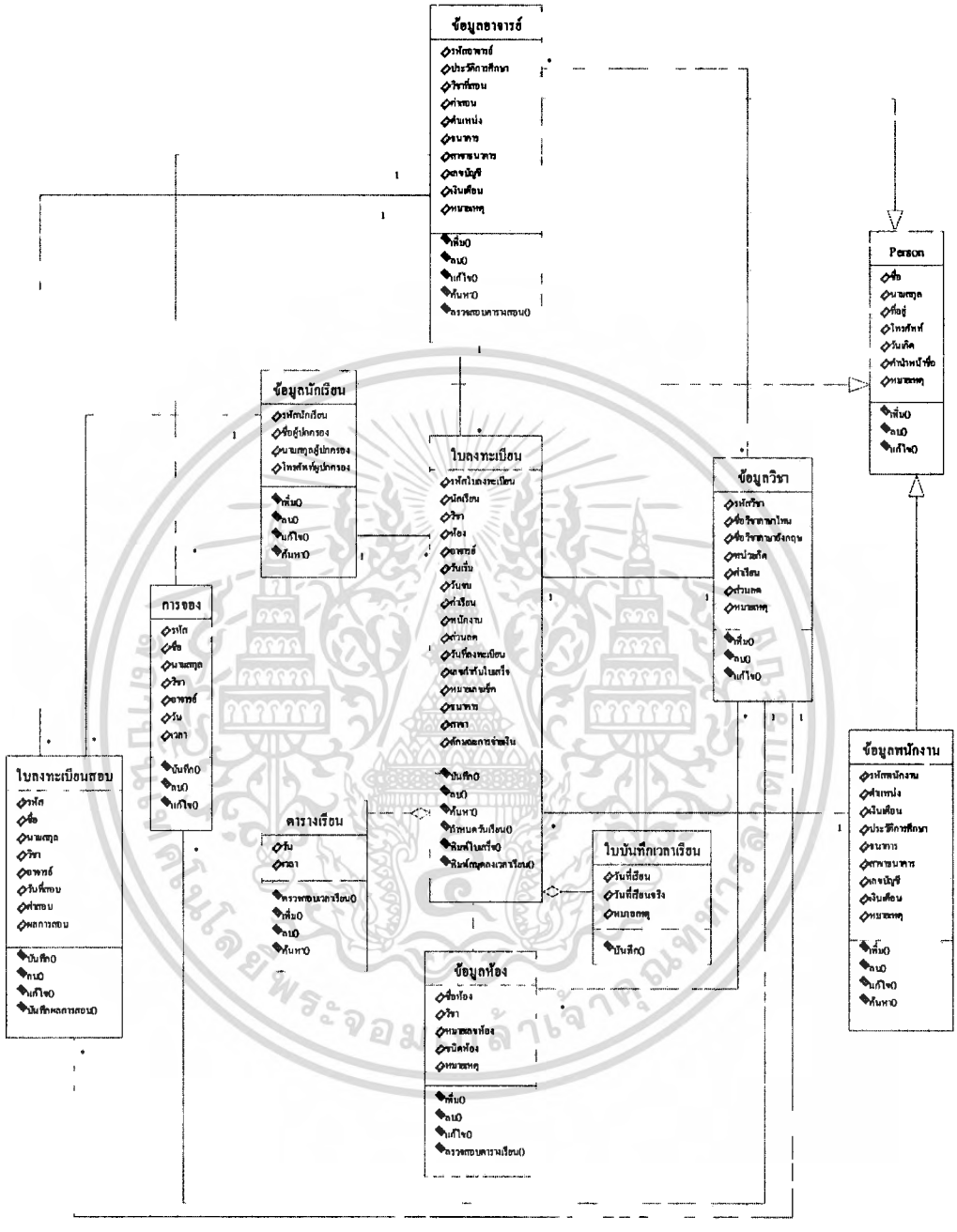
**3.2.2.21 Startchart diagram ของการจอง**

**รูปที่ 3.22** Statechart diagram ของการจอง

**3.2.3 Class Diagram**

จาก Sequence Diagram ซึ่งแสดงขั้นตอนการทำงานและละขั้นตอนของระบบเราจะพบว่าในแต่ละขั้นตอนการทำงานจะมี Class การทำงาน ที่ทำงานร่วมกันอยู่ในแต่ละระบบ ซึ่งเราสามารนำความสัมพันธ์ของแต่ละ Class มาเขียนเป็น Class Diagram ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.23** Class Diagram แสดงความสัมพันธ์ระหว่าง Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

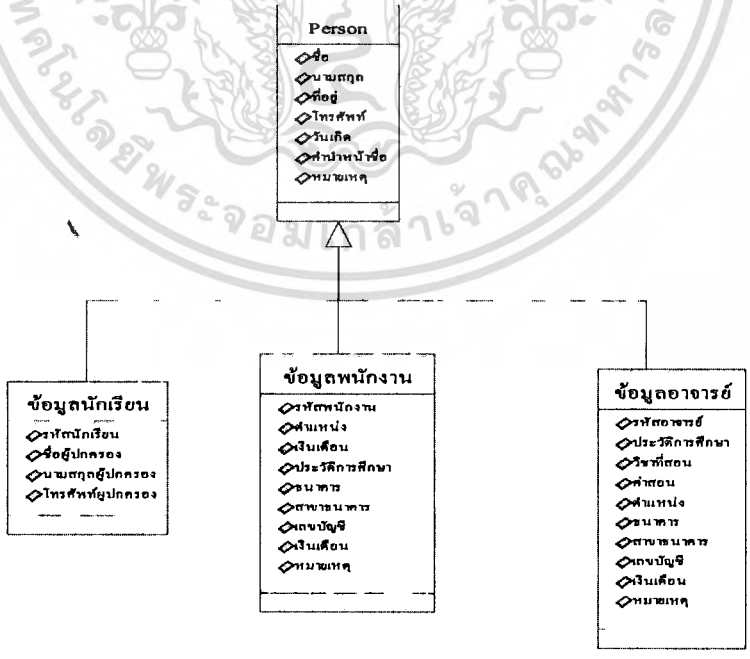
# บทที่ 4
# การออกแบบฐานข้อมูล

## 4.1 ออกแบบฐานข้อมูล

ในการเก็บข้อมูลต่าง ๆ ของระบบ ใช้ฐานข้อมูลแบบสัมพันธ์(Relational Database Management System) โดยเราจัดเก็บข้อมูลลงใน Microsoft Access เพราะฉะนั้นก็ต้องทำการออกแบบฐานข้อมูลเพื่อรองรับการทำงานที่เกิดขึ้น ซึ่งเราจะทำการออกแบบจาก Class Diagram ที่เราได้ทำการออกแบบไว้ในขั้นตอนการดำเนินงาน โดยเราสามารถแปลงเป็น Class Diagram เป็น ER-Diagram โดยนำ attribute ต่างๆ ของแต่ละ class มาเป็น fields ต่างๆ ของแต่ละตาราง ได้ดังนี้

4.1.1 ออกแบบฐานข้อมูล โดยดูจากสัมพันธ์ระหว่าง คลาสอาจารย์ คลาสพนักงาน คลาสนักเรียน และ คลาสบุคคล เป็นความสัมพันธ์แบบสืบทอดซึ่งเราสามารถนำมาออกแบบเป็น ฐานข้อมูลแบบสัมพันธ์ได้ดังนี้

Class Diagram




**รูปที่ 4.1** คลาสไดอะแกรมของอาจารย์ พนักงาน นักเรียน และ บุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

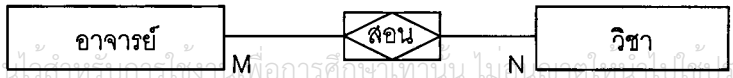
เขียนเป็น ER-Diagram ได้ 3 ตารางดังนี้

| พนักงาน | อาจารย์ | นักเรียน |
|---|---|---|
| - รหัส | - รหัส | - รหัส |
| - ชื่อ | - ชื่อ | - ชื่อ |
| - นามสกุล | - นามสกุล | - นามสกุล |
| - ที่อยู่ | - ที่อยู่ | - ที่อยู่ |
| - โทรศัพท์ | - โทรศัพท์ | - โทรศัพท์ |
| - ตำแหน่ง | - ตำแหน่ง | - ชื่อผู้ปกครอง |
| - เงินเดือน | - การศึกษา | - นามสกุลผู้ปกครอง |
| - การศึกษา | | - โทรศัพท์ผู้ปกครอง |

**รูปที่ 4.2** ฐานข้อมูลของอาจารย์ พนักงาน นักเรียน และ บุคคล

4.1.2 ออกแบบฐานข้อมูลระหว่าง คลาสอาจารย์ คลาสวิชา ซึ่งอาจารย์ 1 คน สามารถสอนได้หลายวิชา และ 1 วิชามีอาจารย์สอนได้หลายคน

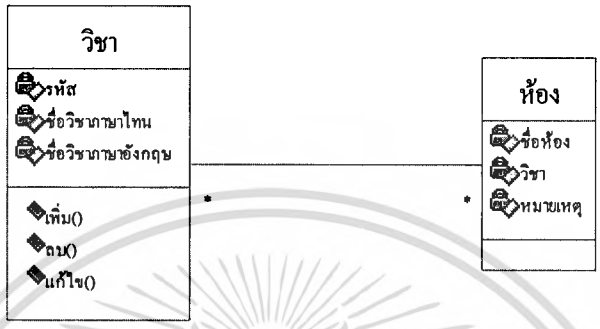
**รูปที่ 4.3** คลาสไดอะแกรมของอาจารย์ และ วิชา



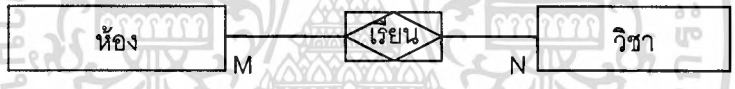

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.4** ER ไดอะแกรมของอาจารย์ และ วิชา

4.1.4 ออกแบบฐานข้อมูลระหว่าง คลาสห้อง คลาสวิชา ซึ่งห้อง 1 ห้อง สามารถสอนได้หลายวิชา และ 1 วิชาสอนได้หลายห้อง



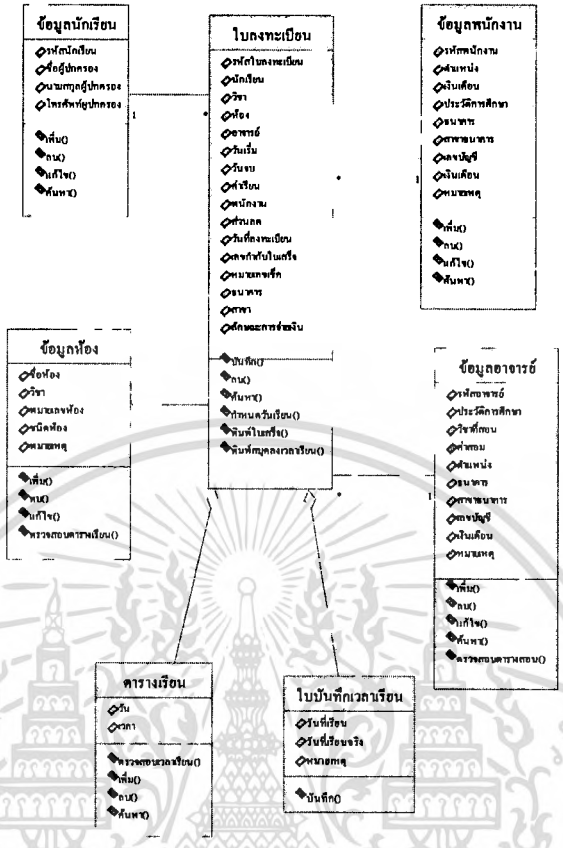

**รูปที่ 4.5** คลาสไดอะแกรมของ วิชา และห้อง



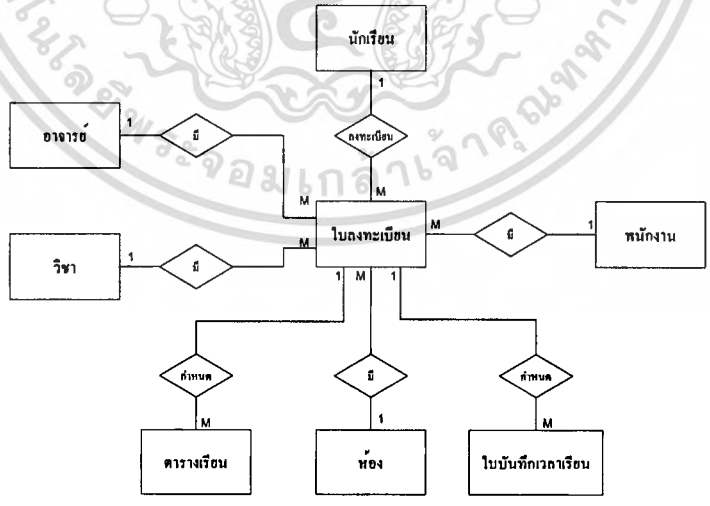

**รูปที่ 4.6** ER ไดอะแกรมของ วิชา และห้อง

4.1.4 ออกแบบฐานข้อมูลระหว่าง คลาสนักเรียน คลาสลงทะเบียน ซึ่งห้อง 1 ห้อง สามารถสอนได้หลายวิชา และ 1 วิชาสอนได้หลายห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



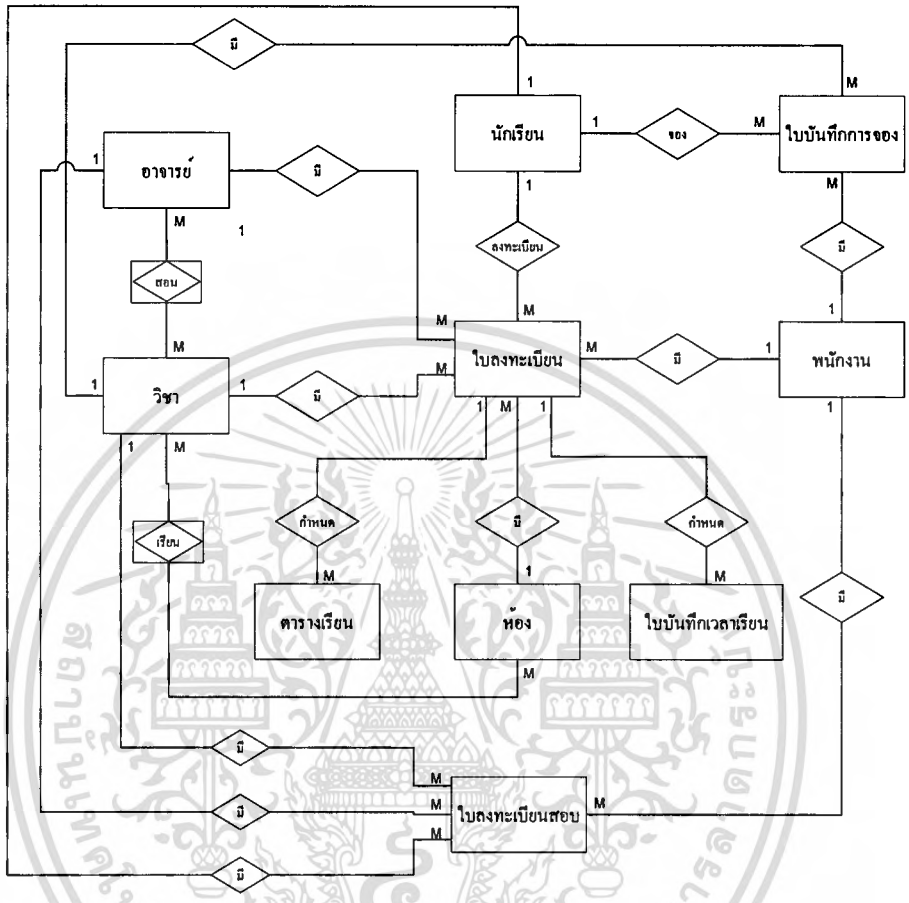

**รูปที่ 4.7** คลาสไดอะแกรมแสดงความสัมพันธ์ของ คลาสลงทะเบียน




**รูปที่ 4.8** ER ไดอะแกรมแสดงความสัมพันธ์ของ คลาสลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Class Diagram ทั้งหมดเราสามารถนำมาเขียน ER-Diagram ได้ดังนี้




**รูปที่ 4.9** ER-Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก ER – Diagram จะมีตารางทั้งหมด 12 ตารางดังนี้

**ตารางที่ 4.1** ห้อง (Room)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| R_id | Text (5) | หมายเลขห้อง | PK |
| R_name | Text (30) | ชื่อห้อง | |
| R_Type | Text (5) | ชนิดของห้อง | |
| R_rmk | Text (50) | หมายเหตุ | |
| | | | |

**ตารางที่ 4.2** วิชา (Crouse)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| C_id | Text (10) | รหัสวิชา | PK |
| C_Tname | Text (30) | ชื่อวิชาไทย | |
| C_Ename | Text (5) | ชื่อวิชาอังกฤษ | |
| C_Credit | Number(1) | หน่วยกิต | |
| C_Price | Currency | ค่าเรียน | |
| C_DisPrice | Currency | ส่วนลด | |
| C_rmk | Text (50) | หมายเหตุ | |
| | | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.3** ห้องเรียนวิชา (RoomCruose)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| R_id | Text (5) | หมายเลขห้อง | PK,FK |
| C_id | Text (10) | รหัสวิชา | PK,FK |
| | | | |

**ตารางที่ 4.4** ใบบันทึกเวลาเรียน (SaveSh)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| S_id | Text (10) | รหัสนักเรียน | PK |
| Date | Date | วันที่เรียน | PK |
| DateOfST | Date | วันที่มาเรียนจริง | PK |
| RMK | Text(100) | หมายเหตุ | |
| | | | |

**ตารางที่ 4.5** ใบลงทะเบียน (Regis)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| Reg_id | Text (10) | รหัสใบลงทะเบียน | PK |
| S_id | Text (10) | รหัสนักเรียน | FK |
| C_id | Text(10) | รหัสวิชา | FK |
| R_id | Text(10) | รหัสห้อง | FK |
| T_id | Text(10) | รหัสอาจารย์ | FK |
| SS_date | Date | วันเริ่มเรียน | |
| SE_date | Date | วันเรียนจบ | |
| CostAmt | Number | ค่าเรียน | |
| Sf_id | Text(10) | รหัสพนักงาน | FK |
| Reg_disc | Number | ส่วนลด | |
| RegDate | Date | วันที่ลงทะเบียน | |
| SequenceNO | Number | เลขกำกับใบเสร็จ | |
| CheckNo | Number | หมายเลขเช็ค | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Field | Data Type | Description | Remark |
|---|---|---|---|
| Bank | Text (30) | ธนาคาร | |
| Branch | Text (40) | สาขา | |
| TypeOfPay | Text (2) | ลักษณะการจ่ายเงิน | |
| | | | |

**ตารางที่ 4.6** นักเรียน (Student)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| S_id | Text (10) | รหัสนักเรียน | PK |
| S_Title | Text (3) | คำนำหน้าชื่อ | |
| S_Name | Text (30) | ชื่อ | |
| S_Surname | Text (30) | นามสกุล | |
| S_Address | Text (80) | ที่อยู่ | |
| S_Tel | Text (20) | โทรศัพท์ | |
| S_adm | Text(30) | ชื่อผู้ปกครอง | |
| S_admTel | Text (20) | โทรศัพท์ | |
| S_birth | Date | วันเกิด | |
| S_rmk | Text(50) | หมายเหตุ | |
| | | | |

**ตารางที่ 4.7** ตารางเรียน (Schedule)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| Reg_id | Text (10) | หมายเลขใบลงทะเบียน | PK |
| Sc_Date | Text(15) | วันเรียน (จันทร์-อาทิตย์) | PK |
| Sc_time | Number | เวลาเรียน | PK |
| | | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8** ใบจอง (Reserve)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| res_id | Text (10) | รหัสใบจอง | PK |
| res_name | Text (10) | ผู้จอง | |
| c_id | Text (10) | รหัสวิชา | FK |
| t_id | Text (10) | รหัสอาจารย์ | FK |
| res_date | Date | วันเรียน | |
| res_time | Time | เวลาเรียน | |
| sf_id | Text (10) | รหัสพนักงาน | FK |
| Rmk | Text (50) | หมายเหตุ | |

**ตารางที่ 4.9** อาจารย์ (Teacher)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| T_id | Text(10) | รหัสอาจารย์ | |
| T_title | Text(3) | คำนำหน้าชื่อ | |
| T_Name | Text(30) | ชื่อ | |
| T_Surname | Text(30) | นามสกุล | |
| T_Add | Text(50) | ที่อยู่ | |
| T_Tel | Text(20) | โทรศัพท์ | |
| T_Pos | Text(20) | ตำแหน่ง | |
| T_Stu | Text(100) | ประวัติการศึกษา | |
| T_Bank | Text(20) | ธนาคาร | |
| T_Branch | Text(20) | สาขา | |
| T_Acc | Text(15) | เลขบัญชี | |
| T_salary | Number | เงินเดือน | |
| Rmk | Text(80) | หมายเหตุ | |
| | | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.10** อาจารย์สอนวิชา (TeacherCourse)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| T_id | Text (10) | รหัสอาจารย์ | PK,FK |
| C_id | Text (10) | รหัสวิชา | PK,FK |
| C_Cost | Number | ค่าสอนต่อชั่วโมง | |
| | | | |

**ตารางที่ 4.11** พนักงาน (Staff)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| Sf_id | Text(10) | รหัสพนักงาน | |
| Sf_title | Text(3) | คำนำหน้าชื่อ | |
| Sf_Name | Text(30) | ชื่อ | |
| Sf_Surname | Text(30) | นามสกุล | |
| Sf_Add | Text(50) | ที่อยู่ | |
| Sf_Tel | Text(20) | โทรศัพท์ | |
| Sf_Pos | Text(20) | ตำแหน่ง | |
| Sf_Stu | Text(100) | การศึกษา | |
| Sf_Bank | Text(20) | ธนาคาร | |
| Sf_Branch | Text(20) | สาขา | |
| Sf_Acc | Text(15) | เลขบัญชี | |
| T_salary | Number | เงินเดือน | |
| Rmk | Text(80) | หมายเหตุ | |
| | | | |

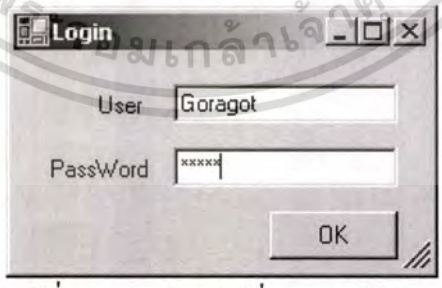
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.12** ใบลงทะเบียนสอบ(RegTest)

| Field | Data Type | Description | Remark |
|---|---|---|---|
| TestID | Text (10) | รหัสใบสอบ | PK,FK |
| TestName | Text (30) | ผู้สอบ | PK,FK |
| TestSurname | Text (30) | นามสกุลผู้สอบ | |
| c_id | Text(10) | รหัสวิชา | FK |
| TestDate | Date | วันที่สอบ | |
| t_id | Text(10) | รหัสอาจารย์ | FK |
| TestRest | Text(500) | ผลการสอบ | |
| TestRMK | Text(500) | หมายเหตุ | |
| TestPrice | Number | ค่าสอบ | |
| Sf_id | Text(10) | รหัสพนักงาน | FK |
| | | | |

## 4.2 ออกแบบหน้าจอและรายงาน

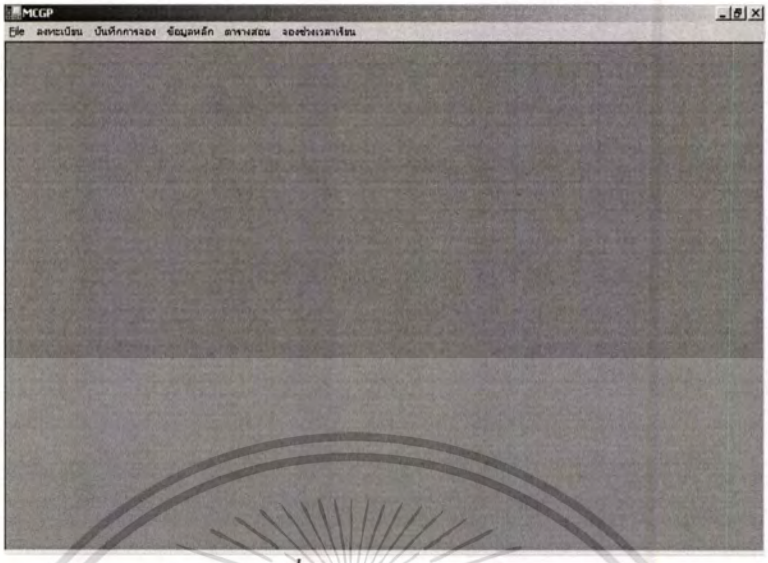
เมื่อเข้าสู่ระบบผู้ใช้ต้องใส่ชื่อและรหัสผ่าน




**รูปที่ 4.10** หน้าจอใส่ชื่อและรหัสผ่าน

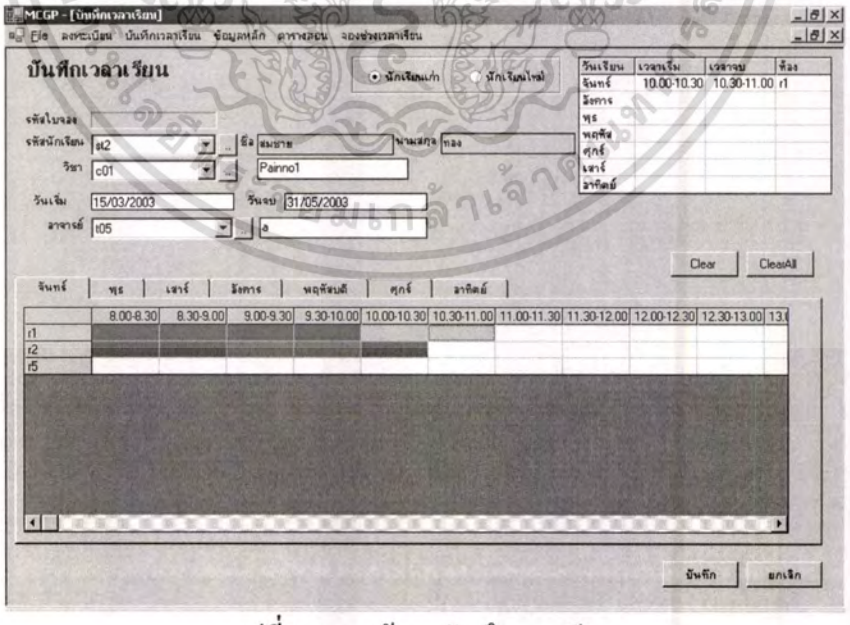
เมื่อใส่รหัสผ่านถูกต้องก็จะเข้าสู่หน้าจอหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



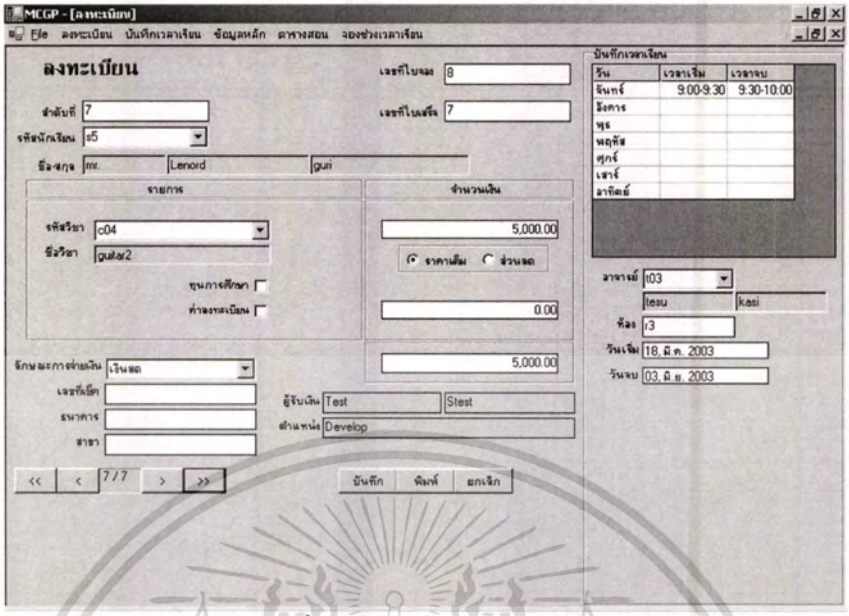

**รูปที่ 4.11** หน้าจอหลัก

1. ระบบลงทะเบียนซึ่งจะประกอบด้วยหน้าจอ บันทึกเวลาเรียน หน้าจอลงทะเบียน ใบเสร็จรับเงิน และสมุดลงเวลาเรียน




**รูปที่ 4.12** หน้าจอบันทึกเวลาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.13** หน้าจอลงทะเบียน

**ใบเสร็จรับเงิน**

เลขที่ 00175 ลำดับ 08

ชื่อ นายสมปอง ทอง

ที่อยู่ 123/456 xxxxxxxxxxxxxx

| ลำดับ | รายการ | ราคา (บาท) |
|---|---|---|
| 1 | ค่าลงทะเบียนเรียนวิชา เปียโนขั้นต้น | 5,150.00 |
| | **รวม** | 5,150.00 |
| **(ห้าพันหนึ่งร้อยห้าสิบบาทถ้วน)** | | |

_____

( )

ผู้รับเงิน

**รูปที่ 4.14** ใบเสร็จรับเงิน

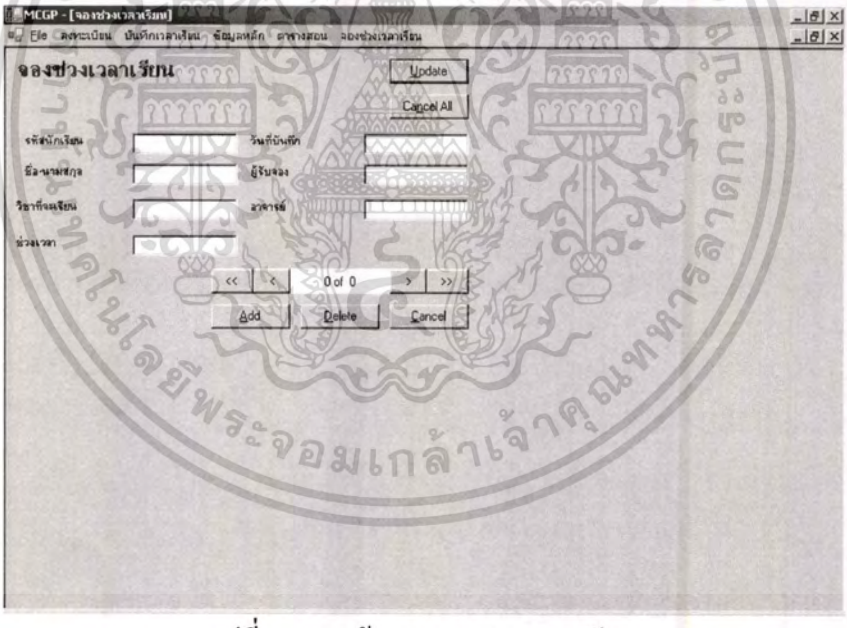
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**สมุดลงเวลาเรียน**

| | |
|---|---|
| ครั้งที่ 1 (1/1/2003)..... | ครั้งที่ 7 (12/2/2003)..... |
| ครั้งที่ 2 (8/1/2003)..... | ครั้งที่ 8 (19/2/2003)..... |
| ครั้งที่ 3 (15/1/2003)..... | ครั้งที่ 9 (26/2/2003)..... |
| ครั้งที่ 4 (22/1/2003)..... | ครั้งที่ 10 (5/3/2003)..... |
| ครั้งที่ 5 (29/1/2003)..... | ครั้งที่ 11 (12/3/2003)..... |
| ครั้งที่ 6 (5/2/2003)..... | ครั้งที่ 12 (19/3/2003)..... |

**รูปที่ 4.15** สมุดลงเวลาเรียน

2. ระบบการจอง ซึ่งจะประกอบด้วยหน้าจอการจองเวลาเรียน และรายงานการจองเวลาเรียน




**รูปที่ 4.16** หน้าจอลงการจองเวลาเรียน

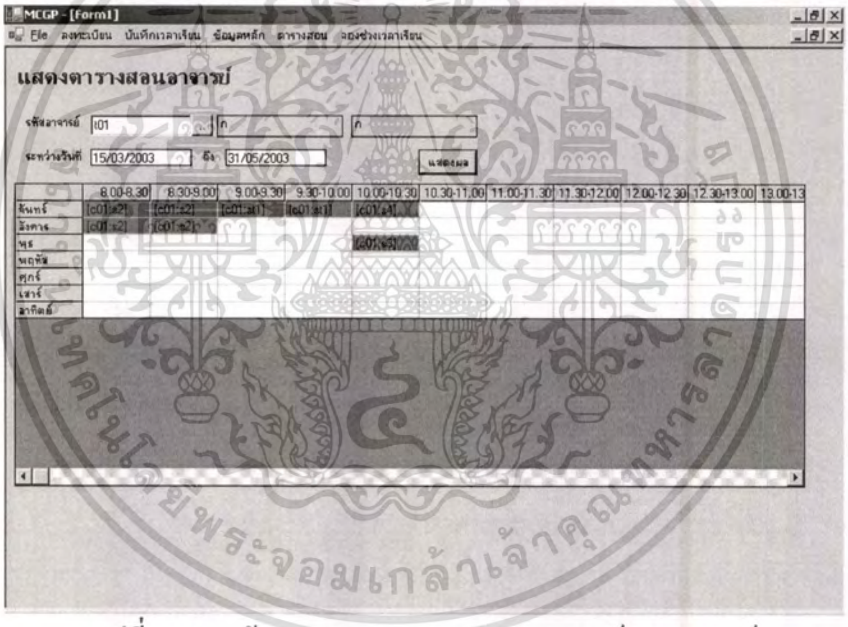
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รายงานการจองเวลาเรียน**
**วันที่ 15/03/2003**

| รหัส | นักเรียน | วัน | ช่วงเวลา | วิชา | อาจารย์ |
|---|---|---|---|---|---|
| 0001 | สมชาย ดดดดดด | จันทร์ | 9.00-12.00 | เปียโน | อาจารย์ก |
| 0A0215 | สมปอง ท | จันทร์ | 17.00-18.00 | กีต้าร์ | อาจารย์ข |
| 0001 | John Max | จันทร์ | 13.00-16.00 | เปียโน | อาจารย์ก |

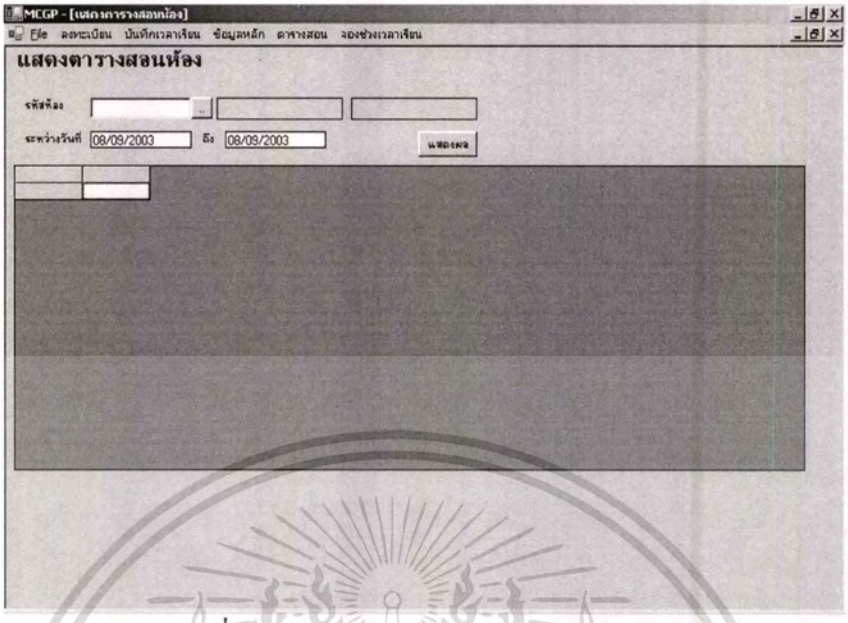
**รูปที่ 4.17** รายงานการจองเวลาเรียน

3. ระบบการจัดตารางเรียน จะประกอบไปด้วยหน้าจอแสดงตารางเรียนตามอาจารย์




**รูปที่ 4.18** หน้าจอแสดงตารางสอนของอาจารย์ตามอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.19** หน้าจอแสดงตารางเรียนของห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางสอนอาจารย์**

วันที่ 15/03/2003

T01254

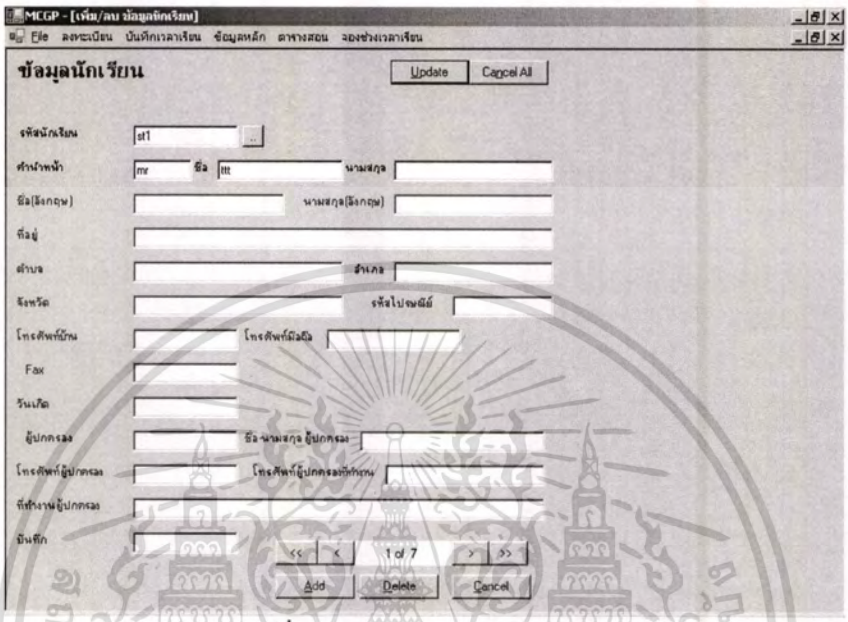
อาจารย์ กก กกกก

| | | | |
|---|---|---|---|
| 8.00-8.30 | วิชา C01<br>นักเรียน สมชาย<br>เริ่ม 15/03/2003<br>จบ 18/06/2003 | 16.30-17.00 | |
| 8.30-9.00 | วิชา C01<br>นักเรียน สมชาย ก<br>เริ่ม 15/03/2003<br>จบ 18/06/2003 | 17.00-17.30 | |
| 9.00-9.30 | | 17.30-18.00 | |
| 9.30-10.00 | | 18.00-18.30 | |
| 10.00-10.30 | | 18.30-19.00 | |
| 10.30-11.00 | | 19.00-19.30 | |
| 11.00-11.30 | | 19.30-20.00 | |
| 11.30-12.00 | วิชา C02 | 20.00-20.30 | |
| 12.00-12.30 | วิชา C02<br>นักเรียน สมชาย<br>เริ่ม 15/03/2003<br>จบ 18/06/2003 | 20.30-21.00 | |
| 12.30-13.00 | | | |
| 13.00-13.30 | | | |
| 13.30-14.00 | | | |
| 14.00-15.30 | | | |
| 15.30-16.00 | | | |
| 16.00-16.30 | | | |

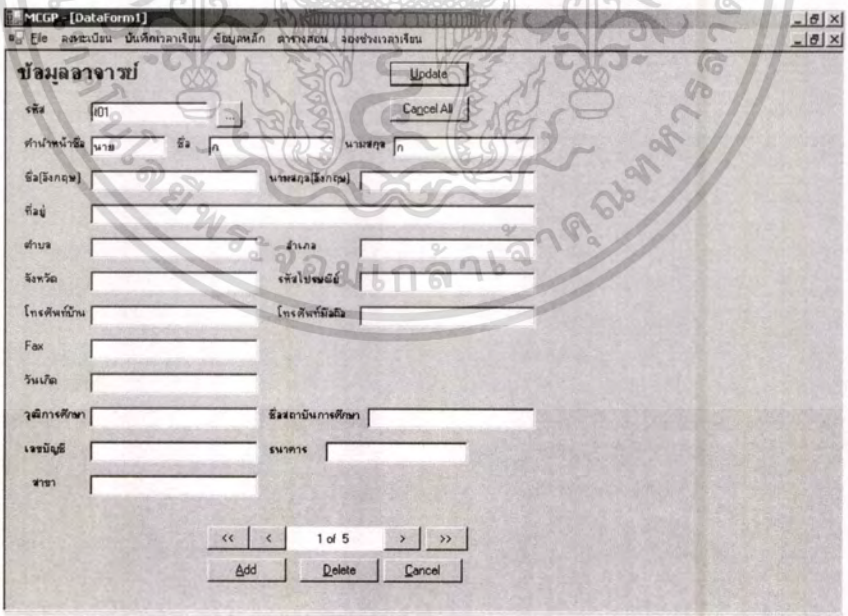
**รูปที่ 4.20** รายงานตารางสอนของอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หน้าจอป้อนข้อมูลหลัก ประกอบด้วย หน้าจอข้อมูลนักเรียน หน้าจอข้อมูลอาจารย์ หน้าจอข้อมูลวิชา และ หน้าจอข้อมูลห้อง

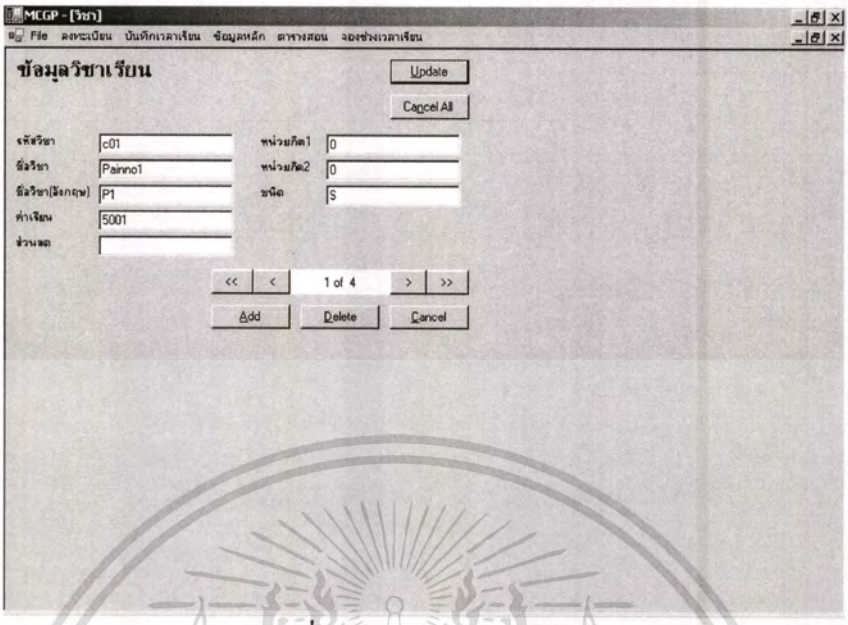



**รูปที่ 4.21** หน้าจอข้อมูลนักเรียน

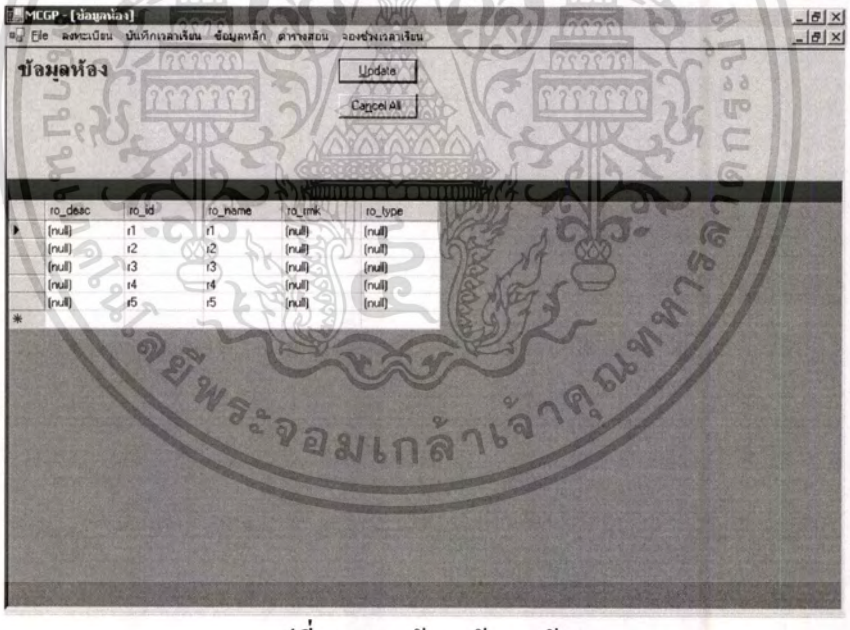



**รูปที่ 4.22** หน้าจอข้อมูลอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.23** หน้าจอข้อมูลวิชา




**รูปที่ 4.24** หน้าจอข้อมูลห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
## สรุปผลและข้อเสนอแนะ

การพัฒนาระบบการจัดการตารางสอนของโรงเรียนดนตรีนี้ยังเป็นการพัฒนาการจัดตารางเรียนตารางสอนและเวลาเรียนเพื่อลดขั้นตอน์การทำงานที่ซับซ้อนและการจัดเก็บข้อมูลอย่างเป็นระบบ เพื่อใช้ข้อมูลต่างๆ ในการบันทึก ค้นหา แก้ไข ลบ ได้อย่างสะดวก และลดข้อผิดในการทำงานที่เกิดขึ้นในการทำงาน ซึ่งถ้าจะนำไปพัฒนาต่อไปก็อาจจะนำการออกแบบระบบเพื่อทำการลงทะเบียนผ่าเวบไซค์หรือสามารถทำการตรวจสอบเวลาเรียนและข้อมูลต่างผ่านเวบไซค์ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

ชาลี วรกุลพิพัฒน์ และ เทพฤทธิ์ บัณฑิตวัฒนาวงศ์ . 2545 **UML ภาษามาตรฐานเพื่อผู้พัฒนาซอฟต์แวร์.** กรุงเทพ. ซีเอ็ดยูเคชั่น.

Ali Bahrami, **Object Oriented system development**, McGRAW-HILL INTERNATIONAL, 1999.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้