# การพัฒนาระบบสารสนเทศระบบลูกหนี้เช่าซื้อการเคหะแห่งชาติ

# Information System Development of National Housing Authority Hire Purchase System


โดย

นางสาวนิตยา เปล่งเจริญศิริชัย

รหัส 41067076

*H001609*

อาจารย์ที่ปรึกษา

ผศ. บรรจง ปิยธำรง


วัน เดือน ปี 22 ธ.ค. 2549

เลขทะเบียน 01609

เลขเรียกหนังสือ .....

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."


รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2542

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **ชื่อหัวข้อ** | การพัฒนาระบบสารสนเทศระบบลูกหนี้เช่าซื้อการเคหะแห่งชาติ |
| **นักศึกษา** | นางสาวนิตยา เปล่งเจริญศิริชัย |
| **อาจารย์ที่ปรึกษา** | ผศ. บรรจง ปิยธำรง |
| **ระดับการศึกษา** | วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ |
| **แขนงวิชา** | วิทยาการสารสนเทศ |
| **ปีการศึกษา** | 2542 |

# บทคัดย่อ

ปัจจุบันมีเทคโนโลยีต่างๆ ที่ใช้ในการพัฒนาระบบสารสนเทศมากขึ้น มีการใช้วิธีการวิเคราะห์และออกแบบระบบหลายวิธี วิธีการในอดีตยังมีความซับซ้อนและยุ่งยากในการนำไปพัฒนาระบบเพื่อใช้งาน จึงได้มีการใช้วิธีการพัฒนาแบบใหม่ที่เป็นการวิเคราะห์และออกแบบระบบเชิงออบเจ็คทำให้มีการใช้งานง่ายขึ้น พัฒนาการเขียนโปรแกรมที่ดี และการวิเคราะห์ในแต่ละขั้นตอนมีความเข้าใจมากขึ้น ซึ่งมีเครื่องมือในการวิเคราะห์และออกแบบระบบเชิงวัตถุที่นิยมใช้คือ UML (Unified Modeling Language) ซึ่งเป็นการทำแบบจำลองเชิงวัตถุและไดอะแกรมที่ช่วยในการพัฒนาระบบได้รวดเร็วขึ้น ในระบบงานลูกหนี้เช่าซื้อการเคหะแห่งชาติมีการวิเคราะห์และออกแบบระบบโดยใช้ UML ที่ช่วยออกแบบระบบแล้วจึงสร้างโปรแกรมเพื่อนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Title** | Information System Development of National Housing Authority Hire Purchase System |
| **Student** | Miss Nittaya Plengjaroensirichai |
| **Advisor** | Asst. Prof. Banjong Piyatamrong |
| **Level of Study** | Master of Science in Information Technology |
| **Major** | Information Science |
| **Academic Year** | 1999 |

# ABSTRACT

In the present, technology approachs use to developing information system. Traditional Approach is complexity and difficult to system development, so object orient approach is used to system analysis and design .UML is object oriented model of system analysis and design that uses easily and good programming . UML is used to analysis and design Hire Purchase of National Housing Authority to developing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า


บทคัดย่อ ..... I

ABSTRACT ..... II

สารบัญ ..... III

สารบัญตาราง ..... VII

สารบัญภาพ ..... VIII

บทที่ 1 บทนำ ..... 1

1.1 คำอธิบายโครงการ ..... 1

1.1.1 วัตถุประสงค์ ..... 1

1.1.2 แผนการดำเนินงาน ..... 1

1.1.3 ผลที่คาดว่าจะได้รับ ..... 1

1.2 โครงสร้างเนื้อหาของเอกสาร ..... 2

1.2.1 บทที่ 1 บทนำ ..... 2

1.2.2 บทที่ 2 ทฤษฎีที่เกี่ยวข้อง ..... 2

1.2.3 บทที่ 3 การวิเคราะห์และออกแบบ ..... 2

1.2.4 บทที่ 4 การสร้างระบบ ..... 2

1.2.5 บทที่ 5 สรุป ..... 2

1.2.6 บทที่ 6 คู่มือระบบ ..... 2

1.2.7 บทที่ 7 คู่มือการใช้งาน ..... 2

1.2.8 ภาคผนวก ก ..... 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง ..... 3

2.1 UML (Unified Modeling Language) ..... 3

2.1.1 Static Modeling ..... 4

2.1.1.1 Use-Case Diagram ..... 4

2.1.1.2 Class Diagram ..... 5

2.1.1.3 Object Diagram ..... 10

2.1.2 Dynamic Model ..... 10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.1 Sequence Diagram ..... 10
2.1.2.2 Collaboration Diagram ..... 11
2.1.2.3 State Diagram ..... 12
2.1.2.4 Activity Diagram ..... 13
2.1.3 โครงสร้างภายนอก ..... 13
2.1.3.1 Component Diagram ..... 13
2.1.3.2 Deployment Diagram ..... 14
2.2.1 การวิเคราะห์ระบบ ..... 15
2.2.1.1 หา actor ของระบบ ..... 15
2.2.1.2 ทำแอคติวิตี้ ไดอะแกรม ..... 16
2.2.1.3 พัฒนา use case ..... 16
2.2.1.4 เตรียม interaction diagram ..... 16
2.2.1.5 การแยกประเภท ..... 16
2.2.1.6 การกระทำซ้ำเพื่อปรับแต่งระบบ ..... 17
2.2.2 การออกแบบระบบ ..... 17
2.2.2.1 การใช้ Axioms ในการออกแบบ ..... 17
2.2.2.2 การออกแบบ Access Layer ..... 19
2.2.2.3 การออกแบบ View Layer ..... 19
2.2.2.4 การทำซ้ำและปรับปรุงการออกแบบทั้งหมด ..... 20
บทที่ 3 การวิเคราะห์และออกแบบ ..... 22
3.1 แบบจำลองของกระบวนการทางธุรกิจ ..... 22
3.2 Use Case Diagram ของระบบ ..... 22
3.2.1 Actor ..... 23
3.2.2 Use Case ..... 23
3.3 อินเทอร์แอกชั่น ไดอะแกรม (Interaction diagram) ..... 24
3.3.1 Use Case DownPayment ..... 24
3.3.2 Use Case Daily BuyPayment ..... 25
3.3.3 Use Case Debtor BuyContract ..... 26
3.3.4 Use Case OverduePayment ..... 27
3.3.5 Use Case DuePayment ..... 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การกำหนด Class Diagram ในขั้นตอนการวิเคราะห์ ....29
3.4.1 การกำหนด Class ของระบบ ....29
3.4.2 ความสัมพันธ์ของ Class ....30
3.4.3 Attribute ของแต่ละ Class ....31
3.4.4 Method ของแต่ละ Class ....34
3.5 การออกแบบ Class Diagram ในขั้นตอนการออกแบบ ....38
3.6 การออกแบบ Access Layer ....43
3.7 การออกแบบ View Layer ....46
บทที่ 4 การสร้างระบบ ....51
4.1 การพัฒนาบนฝั่ง Server ....51
4.1.1 การติดตั้ง DBMS ....51
4.1.2 การสร้างตารางลงใน Database เพื่อเก็บข้อมูลของระบบ ....51
4.1.3 การสร้าง Store Procedure ใน Database ....75
4.2 การพัฒนาบนฝั่ง Client ....75
บทที่ 5 สรุป ....76
5.1 สรุปผลการวิเคราะห์และออกแบบ ....76
5.2 การพัฒนาระบบเชิงวัตถุ ....76
บทที่ 6 คู่มือระบบ ....78
6.1 ขอบเขต (Scope) ของระบบ ....78
6.1.1 ผู้อ่าน (Audience) ....78
6.1.2 องค์กร (Organization) ....78
6.2 ความต้องการของระบบ (System Requirement) ....78
6.3 แบบจำลองของกระบวนการทางธุรกิจ ....78
6.4 แบบจำลอง Use Case ....79
6.4.1 Actor ....79
6.4.2 Use Case ....79
6.4.3 Use Case Diagram ....80
6.5 Interaction Diagram ....80
6.5.1 Sequence Diagram ของ Use Case DownPayment ....80
6.5.2 Sequence Diagram ของ Daily BuyPayment Use Case ....81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.3 Sequence Diagram ของ Use Case DebtorBuyContract....82
6.5.4 Sequence Diagram ของ Use Case OverduePayment....82
6.5.5 Sequence Diagram ของ Use Case DuePayment....83
6.6 Class Diagram....83
6.6.1 Class....83
6.6.2 ความสัมพันธ์ของ Class....84
6.6.3 Attribute....85
6.6.4 Method....88
6.6.5 Class Diagram in Application Logic....92
6.6.6 Class Diagram in Access Layer....93
6.6.7 Class Diagram in View Layer....94
บทที่ 7 คู่มือผู้ใช้งาน....95
7.1 การเข้าสู่โปรแกรม....95
7.2 รายละเอียดลูกหนี้เช่าซื้อ....96
7.3 การรับชำระเงิน....98
7.4 การรับชำระเงินล่วงหน้า....100
7.5 การค้างชำระค่าเช่าซื้อ....102
7.6 ลูกหนี้ถึงกำหนดชำระเงิน....104
7.7 การออกจากโปรแกรม....106
บรรณานุกรม....107
ภาคผนวก ก....108
ประวัติผู้เขียน....114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

หน้า

ตารางที่


1. ตารางที่ 3.1 แสดง Class ของระบบเช่าซื้อ....29
2. ตารางที่ 3.2 แสดง Attribute ของคลาสต่างๆ....31
3. ตารางที่ 3.3 แสดงรายละเอียด Attribute ของ Class ต่างๆ....38
4. ตารางที่ 4.1 ฐานข้อมูล Location....70
5. ตารางที่ 4.2 ฐานข้อมูล Project....70
6. ตารางที่ 4.4 ฐานข้อมูล BuyDownPayment....70
7. ตารางที่ 4.5 ฐานข้อมูล Quotation....71
8. ตารางที่ 4.6 ฐานข้อมูล Unit....71
9. ตารางที่ 4.7 ฐานข้อมูล Paymenttransaction....71
10. ตารางที่ 4.8 ฐานข้อมูล Contract....72
11. ตารางที่ 4.9 ฐานข้อมูล Model....73
12. ตารางที่ 4.10 ฐานข้อมูล PaymentTransactionDetail....73
13. ตารางที่ 4.11 ฐานข้อมูล CustomerInformation....73
14. ตารางที่ 4.12 ฐานข้อมูล Insurance....73
15. ตารางที่ 4.13 ฐานข้อมูล BuyContract....74
16. ตารางที่ 4.14 ฐานข้อมูล AccountInformation....74
17. ตารางที่ 6.1 แสดง Class ของระบบเช่าซื้อ....83
18. ตารางที่ 6.2 แสดงรายละเอียดของ Attribute....85


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

หน้า

รูปที่


1. รูปที่ 2.1 แสดง Use-Case Diagram....4
2. รูปที่ 2.2 แสดง Class Diagram....6
3. รูปที่ 2.3 แสดงส่วนประกอบของ Class....6
4. รูปที่ 2.4 แสดงความสัมพันธ์แบบ Association....6
5. รูปที่ 2.5 แสดงความสัมพันธ์แบบไบนารี....7
6. รูปที่ 2.6 แสดงความสัมพันธ์รหว่าง 2 Class ขึ้นไป....7
7. รูปที่ 2.7 แสดงความสัมพันธ์แบบออ....8
8. รูปที่ 2.8 แสดงความสัมพันธ์แบบ Aggregation....8
9. รูปที่ 2.9 แสดงความสัมพันธ์แบบ Composition....8
10. รูปที่ 2.10 แสดงความสัมพันธ์แบบ Generalization....9
11. รูปที่ 2.11 แสดงความสัมพันธ์แบบ Dependency....9
12. รูปที่ 2.12 แสดงความสัมพันธ์แบบ Refinement....10
13. รูปที่ 2.13 แสดงตัวอย่าง Object....10
14. รูปที่ 2.14 แสดง Sequence Diagram....11
15. รูปที่ 2.15 แสดง Collaboration Diagram....12
16. รูปที่ 2.16 แสดง State Diagram....12
17. รูปที่ 2.17 แสดง Activity Diagram....13
18. รูปที่ 2.18 แสดง Component Diagram....14
19. รูปที่ 2.19 แสดง Deployment Diagram....14
20. รูปที่ 2.20 แสดงความแตกต่างระหว่างผู้ใช้กับ actor....15
21. รูปที่ 2.21 แสดง Axiom และ Corollary....18
22. รูปที่ 3.1แสดง Activity Diagram ของระบบเช่าซื้อ....22
23. รูปที่ 3.2 แสดง Use Case Diagram ของระบบเช่าซื้อ....23
24. รูปที่ 3.3 แสดง Sequence Diagram ของ Use Case DownPayment....24
25. รูปที่ 3.4 แสดง Collaboration Diagram ของ Use Case DownPayment....25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

26. รูปที่ 3.5 แสดง Sequence Diagram ของ Use Case DailyBuyPayment .....25
27. รูปที่ 3.6 แสดง Collaboration Diagram ของ Use Case Daily BuyPayment.....26
28. รูปที่ 3.7 แสดง Sequence Diagram ของ Use Case DebtorBuyContract .....26
29. รูปที่ 3.8 แสดง Collaboration Diagram ของ Use Case DebtorBuyContract.....27
30. รูปที่ 3.9 แสดง Sequence Diagram ของ Use Case OverduePayment .....27
31. รูปที่ 3.10 แสดง Collaboration Diagram ของ Use Case OverduePayment.....28
32. รูปที่ 3.11 แสดง Sequence Diagram ของ Use Case DuePayment .....28
33. รูปที่ 3.12 แสดง Collaboration Diagram ของ Use Case DuePayment.....29
34. รูปที่ 3.13 แสดง Class Diagram ของระบบเช่าซื้อ.....37
35. รูปที่ 3.14 แสดงความสัมพันธ์ของ Class ในขั้นตอนการออกแบบ .....41
36. รูปที่ 3.15 แสดง Class Diagram จากการออกแบบเพิ่มเติม.....42
37. รูปที่ 3.16 แสดง Class ที่เหมาะสมกับฐานข้อมูลแบบ Relational .....44
38. รูปที่ 3.17 แสดงความสัมพันธ์ของ Class ใน Access Layer กับ Business Layer.....45
39. รูปที่ 3.18 แสดงความสัมพันธ์ระหว่าง Class ที่เป็น User Interface.....46
40. รูปที่ 3.19 แสดงความสัมพันธ์ระหว่าง View Layer กับ Business Layer.....47
41. รูปที่ 3.20 แสดง GUI ที่เป็นหน้าจอหลักของระบบ.....48
42. รูปที่ 3.21 แสดง GUI ที่เป็นหน้าจอของรายลูกหนี้เช่าซื้อ.....48
43. รูปที่ 3.22 แสดง GUI ของการชำระเงินล่วงหน้า.....49
44. รูปที่ 3.23 แสดง GUI ของการค้างชำระเงินค่าเช่าซื้อ .....50
45. รูปที่ 3.24แสดง GUI ของลูกหนี้ถึงกำหนดชำระ.....50
46. รูปที่ 3.25 แสดง GUI ของการรับชำระเงินค่าเช่าซื้อประจำวัน.....51
47. รูปที่ 3.26 แสดง GUI ของรายละเอียดเพิ่มเติมการรับชำระเงินค่าเช่าซื้อประจำวัน .....51
48. รูปที่ 4.1แสดงการ Import Oracle8 Data Type.....52
49. รูปที่ 4.2 แสดงการสร้าง Component.....53
50. รูปที่ 4.3 แสดงการตั้งชื่อ Component.....53
51. รูปที่ 4.4 แสดงการกำหนด Component.....54
52. รูปที่ 4.5 แสดงการเปลี่ยนรูป Component .....54
53. รูปที่ 4.6 แสดงการ Assign Class .....55
54. รูปที่ 4.7 แสดงการเลือก Stereotype เป็น Object Type .....55
55. รูปที่ 4.8 แสดงรายละเอียดของ Attribute.....56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

56. รูปที่ 4.9 แสดงการกำหนดขนาดของข้อมูล ..... 57
57. รูปที่ 4.10 แสดงการเลือก Data Type Creation Wizard ..... 58
58. รูปที่ 4.11 แสดงหน้าจอ Data Type Creation Wizard ..... 58
59. รูปที่ 4.12 แสดงหน้าจอการใส่รายละเอียด Class ใหม่ ..... 59
60. รูปที่ 4.13 แสดงการเลือก Attribute ..... 59
61. รูปที่ 4.14 แสดงหน้าจอการสร้าง Primary Key ..... 60
62. รูปที่ 4.15 แสดงหน้าจอ Foreign Key ..... 61
63. รูปที่ 4.16แสดงหน้าจอ รายละเอียดของ Attribute ที่อยู่ในตาราง ..... 61
64. รูปที่ 4.17 แสดงหน้าจอเลือก Class ที่จะแปลง ..... 62
65. รูปที่ 4.18 แสดงผลลัพธ์ของการสร้าง Script DDL ..... 63
66. รูปที่ 4.19 แสดงการ Login เพื่อต่อเข้ากับ DBMS ..... 63
67. รูปที่ 4.20 แสดง Entity Relationship ..... 69
68. รูปที่ 6.1 แสดงกระบวนการของระบบเช่าซื้อ ..... 79
69. รูปที่ 6.2 แสดง Use Case Diagram ของระบบเช่าซื้อ ..... 80
70. รูปที่ 6.3 แสดง Sequence Diagram ของ Use Case DownPayment ..... 81
71. รูปที่ 6.4 แสดง Sequence Diagram ของ Use Case Daily BuyPayment ..... 81
72. รูปที่ 6.5 แสดง Sequence Diagram ของ Use Case DebtorBuyContract ..... 82
73. รูปที่ 6.6 แสดง Sequence Diagram ของ Use Case OverduePayment ..... 82
74. รูปที่ 6.7 แสดง Sequence Diagram ของ Use Case Duepayment ..... 83
75. รูปที่ 6.8 แสดง Activity Diagram ของ getdatecontract method ..... 89
76. รูปที่ 6.9 แสดง Activity Diagram ของ calculatetermMax method ..... 89
77. รูปที่ 6.10 แสดง Activity Diagram ของ verifycustomerid method ..... 91
78. รูปที่ 6.11 แสดง Class Diagram ในส่วนของ ApplicationLogic ..... 92
79. รูปที่ 6.12 แสดง Schema ของระบบฐานข้อมูลโดยใช้ Class Diagram ..... 93
80. รูปที่ 6.13 แสดง Class Diagram ในส่วนของ View Layer ..... 94
81. รูปที่ 7.1 แสดงการ Login เข้าสู่ระบบ ..... 95
82. รูปที่ 7.2 แสดงหน้าต่างหลักของโปรแกรม ..... 95
83. รูปที่ 7.3 แสดงหน้าจอรายละเอียดลูกหนี้เช่าซื้อ ..... 96
84. รูปที่ 7.4 แสดงรายละเอียดลูกหนี้เช่าซื้อ ..... 97
85. รูปที่ 7.5 แสดงหน้าจอความก้าวหน้าการประมวลผลการพิมพ์รายละเอียดลูกหนี้เช่าซื้อ ..97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

86. รูปที่ 7.6 แสดงรายละเอียดลูกหนี้ที่พร้อมจะพิมพ์ ....98
87. รูปที่ 7.7 แสดงหน้าจอการรับชำระเงิน ....98
88. รูปที่ 7.8 แสดงรายละเอียดการรับชำระเงิน....99
89. รูปที่ 7.9 แสดงหน้าจอรายละเอียดการรับชำระเงินเพิ่มเติม ....99
90. รูปที่ 7.10 แสดงรายละเอียดการรับชำระเงินที่พร้อมจะพิมพ์....100
91. รูปที่ 7.11 แสดงหน้าจอการรับชำระเงินล่วงหน้า ....101
92. รูปที่ 7.12 แสดงรายละเอียดการรับชำระเงินล่วงหน้า....101
93. รูปที่ 7.13 แสดงรายละเอียดการรับชำระเงินล่วงหน้าพร้อมจะพิมพ์ ....102
94. รูปที่ 7.14 แสดงหน้าจอการค้างชำระค่าเช่าซื้อ ....103
95. รูปที่ 7.15 แสดงรายละเอียดการค้างชำระค่าเช่าซื้อ....103
96. รูปที่ 7.16 แสดงรายละเอียดการค้างชำระค่าเช่าซื้อที่พร้อมจะพิมพ์....104
97. รูปที่ 7.17 แสดงหน้าจอลูกหนี้ถึงกำหนดชำระ ....105
98. รูปที่ 7.18 แสดงรายละเอียดลูกหนี้ถึงกำหนดชำระ ....105
99. รูปที่ 7.19 แสดงรายละเอียดลูกหนี้ถึงกำหนดชำระที่พร้อมจะพิมพ์ ....106
100. รูปที่ ก.1 แสดงรายงานรายละเอียดลูกหนี้เช่าซื้อ ....109
101. รูปที่ ก.2 แสดงรายงานการรับชำระค่าเช่าซื้อ....110
102. รูปที่ ก.3 แสดงรายงานการรับชำระเงินล่วงหน้า ....111
103. รูปที่ ก.4 แสดงรายงานลูกหนี้ค้างชำระ....112
104. รูปที่ ก.5 แสดงรายงานลูกหนี้ถึงกำหนดชำระ ....113

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

# บทนำ

## 1.1 คำอธิบายโครงการ

โครงการนี้เป็นโครงการ การพัฒนาระบบรับชำระของการเคหะแห่งชาติ ซึ่งจะใช้การพัฒนาระบบเชิงวัตถุ (Object-Oriented System Development) และใช้ UML (Unified Modeling Language) ช่วยในการสร้างแบบจำลอง (Model) ของระบบลูกหนี้เช่าซื้อ และใช้ DBMS เป็น Oracle เวอร์ชั่น 8 และพัฒนา Front End ด้วย Oracle Developer เวอร์ชั่น 6

ระบบบัญชีลูกหนี้เช่าซื้อ การเคหะแห่งชาติ มีการแบ่งเป็นระบบย่อยๆ โดยระบบตั้งหนี้มีหน้าที่ในการตั้งโครงการต่างๆ ระบบรับชำระมีหน้าที่ในการรับชำระเงินต่างๆ ระบบลูกหนี้เช่าซื้อเป็นระบบหนึ่งของระบบบัญชีลูกหนี้การเคหะแห่งชาติ โดยจะมีหน้าที่ควบคุมตรวจสอบการทำบัญชีลูกหนี้เช่าซื้อ จัดทำรายละเอียดลูกหนี้เช่าซื้อตามสัญญาซึ่งลูกหนี้จะเช่าซื้อบ้าน

### 1.1.1 วัตถุประสงค์

1. ปรับปรุงขั้นตอนการทำงานในระบบลูกหนี้เช่าซื้อให้ใช้งานได้สะดวก รวดเร็วขึ้น
2. บริหารข้อมูลที่ใช้ในระบบลูกหนี้เช่าซื้อให้มีความถูกต้อง ทันสมัยในการนำไปใช้งาน

### 1.1.2 แผนการดำเนินงาน

1. ศึกษาและทำความเข้าใจในระบบลูกหนี้เช่าซื้อ
2. วิเคราะห์และออกแบบระบบโดยใช้ UML
3. พัฒนาระบบโดยใช้แบบจำลองที่ได้ออกแบบไว้

### 1.1.3 ผลที่คาดว่าจะได้รับ

1. ข้อมูลมีความถูกต้อง รวดเร็ว และทันสมัยตลอดเวลา
2. ข้อมูลมีความถูกต้องและทันสมัยในการนำไปใช้งาน
3. สามารถทราบรายละเอียดเกี่ยวกับลูกหนี้ได้อย่างรวดเร็ว ถูกต้อง และทันทีเมื่อต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 โครงสร้างเนื้อหาของเอกสาร

**1.2.1 บทที่ 1 บทนำ**

เป็นบทที่กล่าวถึงภาพรวมของโครงการ ซึ่งประกอบไปด้วยวัตถุประสงค์ แผนการดำเนินงาน ผลที่คาดว่าจะได้รับ

**1.2.2 บทที่ 2 ทฤษฎีที่เกี่ยวข้อง**

เป็นบทที่กล่าวถึงทฤษฎีที่เกี่ยวข้องกับโครงการนี้ซึ่งจะประกอบไปด้วยกระบวนการของการวิเคราะห์และออกแบบเชิงวัตถุ

**1.2.3 บทที่ 3 การวิเคราะห์และออกแบบ**

เป็นบทที่อธิบายการวิเคราะห์และออกแบบหลังจากผ่านการศึกษาระบบมาแล้ว โดยมีการแสดงเป็นแบบจำลองตามของระบบ

**1.2.4 บทที่ 4 การสร้างระบบ**

เป็นบทที่กล่าวถึงวิธีการพัฒนาซอฟต์แวร์ระบบเช่าซื้อ ซึ่งจะเป็นการสร้างซอฟต์แวร์ตามสิ่งที่ได้จากการวิเคราะห์และออกแบบ

**1.2.5 บทที่ 5 สรุป**

เป็นบทที่กล่าวถึงผลสรุปการวิเคราะห์และออกแบบ และการพัฒนาระบบเชิงวัตถุ

**1.2.6 บทที่ 6 คู่มือระบบ**

เป็นบทที่เกี่ยวข้องผู้ดูแลระบบหรือผู้ที่จะพัฒนาระบบเพื่อศึกษาระบบเพิ่มเติมต่อไป

**1.2.7 บทที่ 7 คู่มือการใช้งาน**

เป็นบทที่เกี่ยวข้องกับผู้ใช้งานระบบเพื่อเป็นคู่มือในการใช้งาน

**1.2.8 ภาคผนวก**

เป็นบทที่แสดงรูปแบบรายงานต่างๆ ที่มีในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# ทฤษฎีที่เกี่ยวข้อง

## 2.1 UML (Unified Modeling Language)

ในการพัฒนาระบบจะมีการวิเคราะห์และออกแบบระบบแล้วจึงจะมีการสร้างระบบ วิธีการวิเคราะห์และออกแบบระบบวิธีหนึ่งที่นิยมคือการใช้ UML UML เป็นภาษาที่ใช้ในการสร้างแบบจำลอง (Modeling Language) และกระบวนการต่างๆ สำหรับการพัฒนาระบบที่ใช้เทคโนโลยีเชิงวัตถุ

UML ประกอบด้วยส่วนหลักๆ ดังนี้

1. มุมมอง (View) เป็นการแสดงมุมมองในด้านต่างๆ ของระบบ ซึ่งจะประกอบไปด้วยไดอะแกรมต่างๆ ในแต่ละมุมมอง
2. Diagram เป็นรูปภาพที่ใช้อธิบายมุมมองของระบบ ซึ่ง UML มีทั้งหมด 9 Diagram ต่อไปนี้
   - Use Case Diagram
   - Class Diagram
   - Object Diagram
   - State Diagram
   - Sequence Diagram
   - Collaboration Diagram
   - Activity Diagram
   - Component Diagram
   - Deployment Diagram
3. ส่วนประกอบของแบบจำลอง (Model element) เป็นส่วนที่ใช้ภายในไดอะแกรม เช่น Class Object หรือความสัมพันธ์ เป็นต้น
4. วิธีทั่วไป (General mechanism) ใช้ในการเพิ่มประสิทธิภาพให้กับ UML เช่นการเพิ่มหมายเหตุ (Note) หรือการตกแต่ง (Adornment) ส่วนประกอบของแบบจำลองในไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

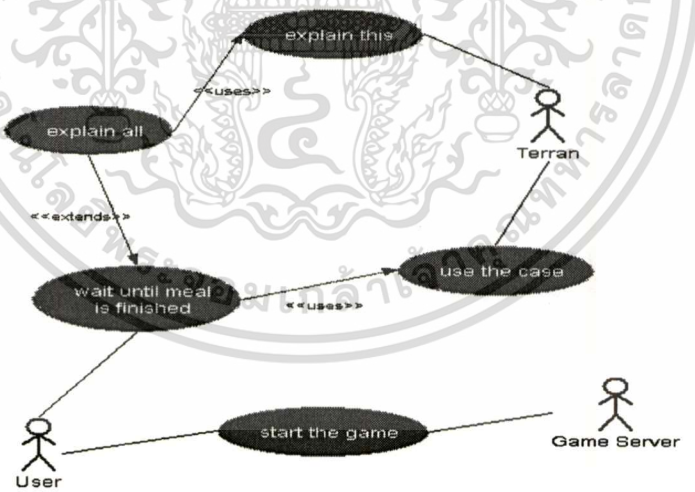
ระบบทุกระบบจะมีโครงสร้างแบบสแตติก (Static Structure) และแบบไดนามิก (Dynamic Behavior) UML สามารถรองรับได้ทั้งโครงสร้างแบบสแตติกและแบบไดนามิกของระบบ โครงสร้างแบบสแตติกจะใช้ Class Diagram และ Object Diagram และโครงสร้างแบบไดนามิกใช้ State Diagram Sequence Diagram Collaboration Diagram และ Activity Diagram อธิบายพฤติกรรมระบบ

**2.1.1 Static Modeling**

Static Modeling จะเป็นลักษณะแทนโครงสร้างของระบบมี Class Diagram Usecase Diagram และ Object Diagram โดย Class Diagram จะอธิบายถึงสิ่งที่อยู่ในระบบและความสัมพันธ์ระหว่างสิ่งเหล่านั้น ส่วน Object Diagram จะเป็นมุมมองตัวอย่างของ Class Diagram ที่จะบอกว่าระบบทำอะไรเป็นลักษณะแบบจำลองในมุมมองที่อยู่ในระบบ ส่วน Use-Case Diagram จะมีการสร้างแบบจำลองในมุมมองจากภายนอกระบบ

**2.1.1.1 Use-Case Diagram**

Use-Case Diagram เป็นแผนผังที่แสดงความต้องการของระบบ โดยระบุถึงกิจกรรมที่มีในระบบ มีการดำเนินงานอะไรเกิดขึ้นบ้างในระบบ แสดงดังรูปที่ 2.1




**รูปที่ 2.1 แสดง Use-Case Diagram**

ส่วนประกอบของ Use-Case Diagram

1. Actor เป็นส่วนที่ติดต่อกับระบบ อาจเป็นผู้ใช้ ฮาร์ดแวร์ภายนอกหรือระบบต่างๆ ที่อยู่ภายนอก โดยจะใช้สัญลักษณ์รูปคนและมีชื่อใต้รูป ซึ่งมีการส่งข้อมูล รับข้อมูลหรือมีการแลกเปลี่ยนข้อมูลกับระบบ actor จะต่างจากผู้ใช้ระบบ (user) โดยผู้ใช้ระบบคนหนึ่งอาจแสดงได้หลายบทบาท (หลาย actor ) ในที่ติดต่อระหว่างผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

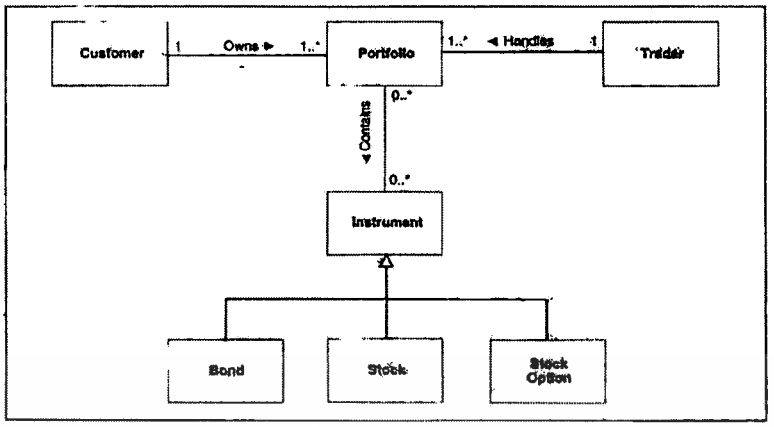
กับระบบหรือระหว่างระบบกับระบบเพื่อหาความต้องการของระบบ actor อาจเป็นเพียงส่งอินพุตเข้าสู่ระบบ หรือรับเอาท์พุตจากระบบ หรือเป็นทั้งอินพุตและเอาท์พุตให้กับระบบ

2. Use case เป็นกิจกรรมหรืองานต่างๆที่เกิดขึ้นในระบบที่มีการติดต่อกันระหว่างผู้ใช้กับระบบ โดยใช้สัญลักษณ์วงรีและมีชื่ออยู่ในวงรีหรือใต้วงรี ซึ่งอาจเป็นกิจกรรมที่ติดต่อระหว่างผู้ใช้กับระบบหรือระหว่างระบบกับระบบเพื่อหาความต้องการของผู้
3. ความสัมพันธ์ภายใน Use-Case Diagram เป็นความสัมพันธ์ระหว่าง actor กับยูสเคส หรือ use case กับ use case ซึ่งสามารถแบ่งได้ 3 แบบ คือ
   - ความสัมพันธ์ระหว่าง Use-Cases ด้วยกันจะมี 2 ลักษณะคือ Use และ Extend ความสัมพันธ์แบบ Use จะแทนด้วยลูกศรหัวปิดไม่ทึบที่ชี้ไปยัง Use Case ที่จะใช้พร้อมกับมี <<Uses>> กำกับ เป็นความสัมพันธ์ระหว่าง use case กับ use case เกิดขึ้นเมื่อ use case หนึ่งต้องใช้กิจการรมบางอย่างจากอีก use case หนึ่ง หรือมีกิจกรรมที่ร่วมกัน ส่วนความสัมพันธ์แบบ Extend จะแทนด้วยลูกศรหัวปิดไม่ทึบที่ชี้ไปยัง Use Case ที่เป็นจุดเริ่ม (Base) พร้อมกับมี <<Extends>> กำกับเป็นความสัมพันธ์ระหว่าง use case กับ use case ในกรณีที่มีกิจกรรมคล้ายๆกัน แต่มีหน้าที่เพิ่มเติมขึ้นเป็นกิจกรรมที่มีความเฉพาะกว่า จึงเป็นลักษณะคล้ายกับ Superclass และ Subclass
   - ความสัมพันธ์ระหว่าง Actor มีเพียงความสัมพันธ์แบบ Generalization จะแทนด้วยลูกศรหัวปิดที่ชี้ไปยัง Actor ที่เป็นลักษณะของ Super Class
   - ความสัมพันธ์ระหว่าง Actor กับ Use Case มีเพียงความสัมพันธ์แบบ Association ซึ่งแทนด้วยลูกศรหัวเปิด หรือไม่มีหัวลูกศรในกรณีของทั้ง Actor และ Use Case มีการความสัมพันธ์แบบ 2 ทาง ซึ่งปลาย (หาง)

**2.1.1.2 Class Diagram**

เป็นไดอะแกรมที่ทำหลังจากสร้าง Use Case Diagramแล้ว เป็นไดอะแกรมหลักของการวิเคราะห์ระบบที่แสดงโครงสร้างแบบสแตติกของแบบจำลอง เป็นที่รวมของ Class และความสัมพันธ์ระหว่าง Class แสดงดังรูปที่ 2.2 Class จะใช้สัญลักษณ์สี่เหลี่ยมผืนผ้า ซึ่งประกอบด้วยชื่อ Class Attribute และ Operation ดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

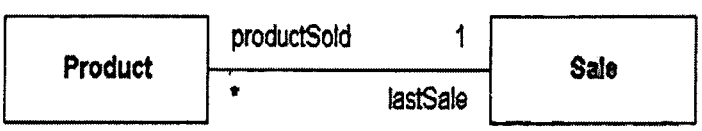



**รูปที่ 2.2 แสดง Class Diagram**

**รูปที่ 2.3 แสดงส่วนประกอบของ Class**

ความสัมพันธ์ระหว่าง Class ที่ใช้กันสามารถแบ่งได้เป็น 4 ความสัมพันธ์หลักๆ คือ

1. Association เป็นความสัมพันธ์แบบสองทางระหว่าง Class แสดงถึงความเกี่ยวข้องกันระหว่าง Class เป็นการแสดงการติดต่อระหว่าง Class ซึ่งอาจจะเป็นการติดต่อระหว่าง Class ที่มีความสัมพันธ์กันหรือเป็นการอ้างถึงไปยังอีก Class หนึ่ง โดยจะมีการตั้งชื่อของความสัมพันธ์ เช่น next to เป็นต้น และมีการแสดงบทบาทของ Class ที่มีความสัมพันธ์กัน ใช้สัญลักษณ์เส้นตรงและมีชื่อของความสัมพันธ์กำกับด้วย ดังแสดงในรูปที่ 2.4

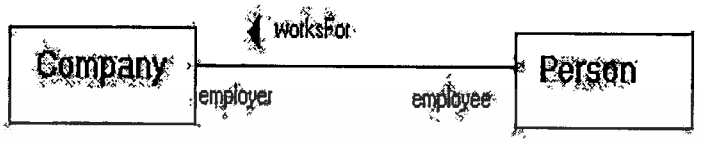



**รูปที่ 2.4 แสดงความสัมพันธ์แบบ Association**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

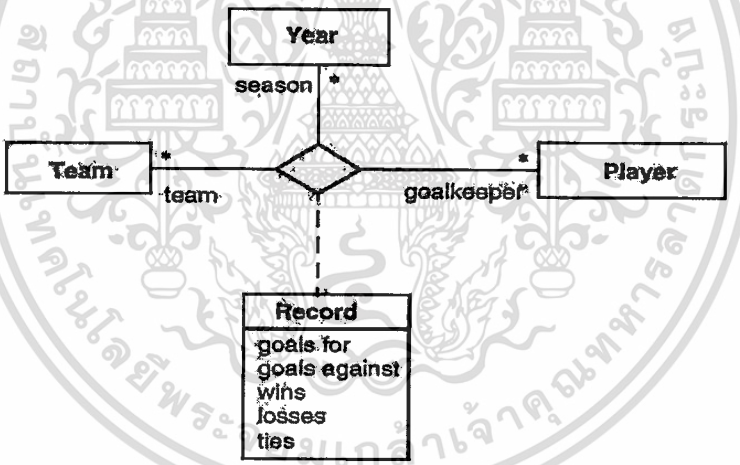
ลักษณะความสัมพันธ์แบบ association ยังมีหลายแบบ เช่น

- ความสัมพันธ์แบบไบนารี (Binary association) เป็นความสัมพันธ์ระหว่าง Class 2 Class ดังรูป 2.5




**รูปที่ 2.5 แสดงความสัมพันธ์แบบไบนารี**

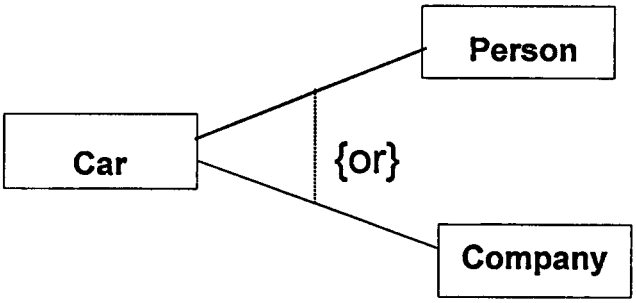
- ความสัมพันธ์ระหว่าง 2 Class ขึ้นไป (N-Ary association) ซึ่งแสดงดังรูป 2.6




**รูปที่ 2.6 แสดงความสัมพันธ์รหว่าง 2 Class ขึ้นไป**

- ความสัมพันธ์แบบออ (OR association) เป็นความสัมพันธ์ระหว่าง Class หนึ่งกับ Class อื่นๆ ดังรูป 2.7 โดย Class สามารถมีความสัมพันธ์กับ Class แรกหรือ Class ที่สองได้เพียง Class เดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

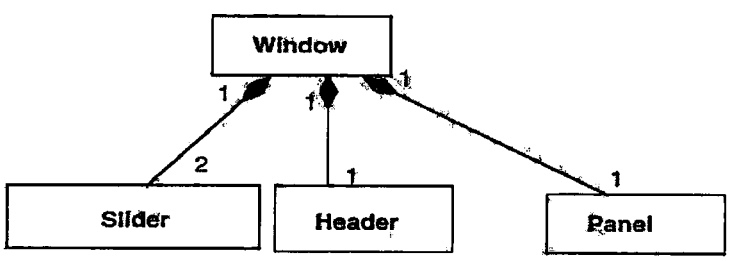



**รูปที่ 2.7 แสดงความสัมพันธ์แบบออ**

- ความสัมพันธ์แบบ Aggregation เป็นความสัมพันธ์เป็นความสัมพันธ์แบบรวม โดย Class หนึ่งเป็นส่วนหนึ่งของอีก Class หนึ่ง (Whole-Part) จะใช้สัญลักษณ์ลูกศรมีหัวรูปสี่เหลี่ยมขนมเปียกปูนไม่ทึบอยู่ด้านที่เป็นทั้งหมด แสดงดังรูป 2.8

**รูปที่ 2.8 แสดงความสัมพันธ์แบบ Aggregation**

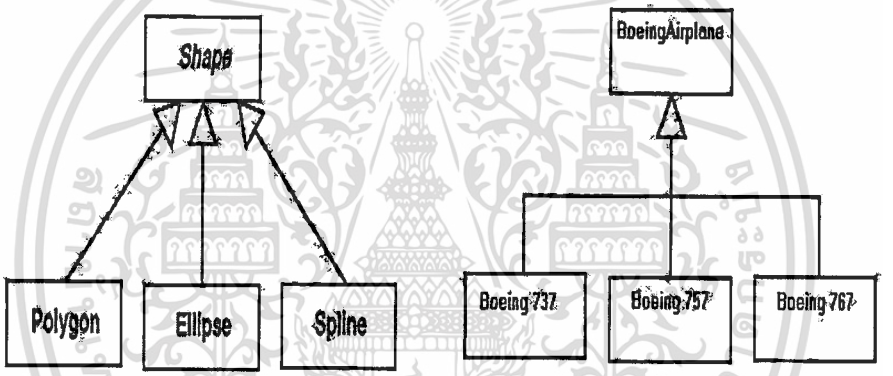
- ความสัมพันธ์แบบ Composition เป็นความสัมพันธ์รูปแบบหนึ่งที่หลายๆ Class ย่อยประกอบกันเป็น Class หนึ่ง ซึ่งถ้าขาด Class ย่อย Class ใด Class หนึ่งจะทำให้ Class ไม่สมบูรณ์ โดยมีส่วนทั้งหมดและส่วนที่เป็นบางส่วน จะใช้สัญลักษณ์ลูกศรที่มีหัวเป็นรูปสี่เหลี่ยมขนมเปียกปูนทึบอยู่ทางด้านที่เป็นทั้งหมด แสดงดังรูป 2.9




**รูปที่ 2.9 แสดงความสัมพันธ์แบบ Composition**

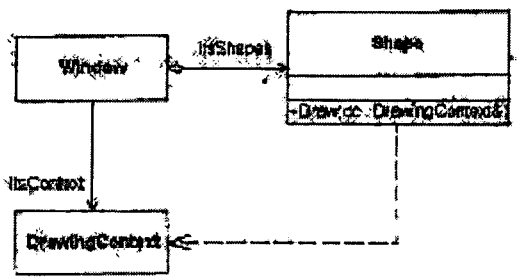
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Generalization เป็นความสัมพันธ์ที่แบ่งระหว่าง Superclass กับ Subclass เป็นความสัมพันธ์ระหว่าง Class ในลักษณะ Superclass กับ Subclass โดยใช้สัญลักษณ์ลูกศรหัวปิดไม่ทึบชี้จาก Subclass ไปยัง Superclass ดังรูป 2.10 โดยแบ่งได้ 2 แบบคือ
   - ความสัมพันธ์แบบแยกกัน (Separate target style) เป็นความสัมพันธ์แบบ Subclass/Superclass ที่แต่ละ Subclass ไม่มีการใช้คุณสมบัติรวมกัน
   - ความสัมพันธ์แบบใช้ร่วมกัน (Shared target style) เป็นความสัมพันธ์แบบ Subclass/Superclass ที่ Subclass ต่างๆ มีการใช้คุณสมบัติร่วมกัน




**รูปที่ 2.10 แสดงความสัมพันธ์แบบ Generalization**

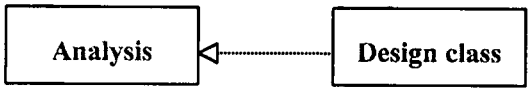
3. Dependency เป็นความสัมพันธ์ในลักษณะที่มีการขึ้นตรงต่อกัน คือเมื่อมีการแก้ไขสิ่งหนึ่งจะมีผลกับอีกสิ่งหนึ่งที่ขึ้นต่อกัน (Dependent) ใช้สัญลักษณ์เป็นเส้นประและลูกศรหัวเปิด ดังแสดงในรูปที่ 2.11




**รูปที่ 2.11 แสดงความสัมพันธ์แบบ Dependency**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Refinement เป็นความสัมพันธ์ของรูปแบบการอธิบาย 2 แบบซึ่งอธิบายในสิ่งๆ เดียวกัน แต่คนละมิติกัน ใช้สัญลักษณ์เป็นเส้นประและลูกศรหัวปิด ดังแสดงในรูปที่ 2.12




**รูปที่ 2.12 แสดงความสัมพันธ์แบบ Refinement**

**2.1.1.3 Object Diagram**

Object Diagram แสดงรายละเอียดของ Object และความสัมพันธ์ระหว่าง Object ซึ่งเป็น Instance ของ Class และเป็นตัวอย่างของ Class Diagram ที่ซับซ้อนเพื่อช่วยให้เกิดความเข้าใจได้ง่ายขึ้น Object Diagram มีลักษณะคล้ายกับ Class Diagram แต่ต่างกันตรงที่ชื่อของ Object จะขีดเส้นใต้ไว้ที่ชื่อของ Object หรือเป็นชื่อที่นำหน้าชื่อของ Class และคั่นด้วย ":" จะได้รูปแบบคือ "ชื่อ Object : ชื่อ Class" การตั้งชื่อของ Object อาจจะใช้ชื่อของ Class หรือใช้ชื่อใหม่ได้ แสดงดังรูปที่ 2.13

**รูปที่ 2.13 แสดงตัวอย่าง Object**

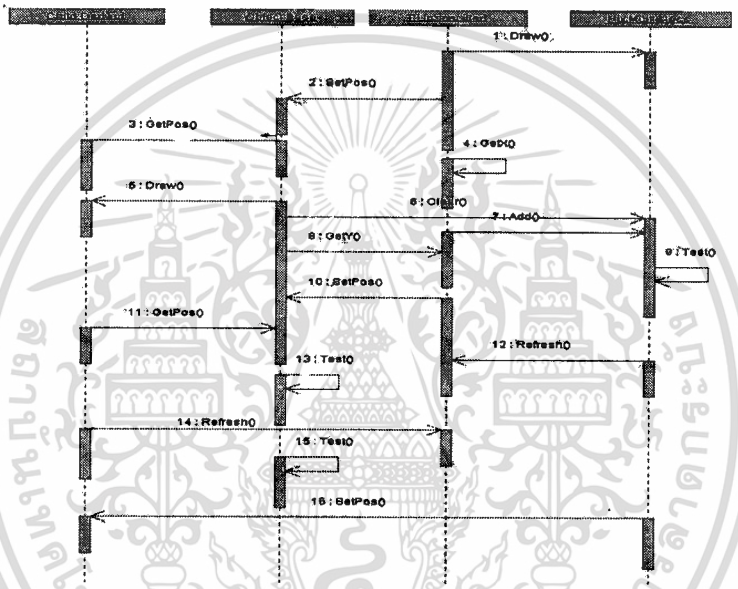
**2.1.2 Dynamic Model**

เป็น Model ที่สามารถอธิบายพฤติกรรมของระบบโดยการนำเสนอตัวอย่างของเหตุการณ์ที่เกิดขึ้นภายในระบบนั้นๆ รวมถึงเป็นการนำเสนอวิธีการของสิ่งที่มีอยู่ภายในระบบว่ามีการติดต่อสื่อสารกันอย่างไร โดยมีการส่งข้อความ (Message) ให้กันและกัน ซึ่ง Dynamic Model นี้คล้ายกับเป็นการเพิ่มรายละเอียดของระบบเพื่อให้เหมาะสมกับการสร้างหรือพัฒนาต่อไป

**2.1.2.1 Sequence Diagram**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sequence Diagram เป็นลำดับของงานที่เกิดขึ้น เป็นแบบจำลองแบบไดนามิก ที่อธิบายความสัมพันธ์ของ Class ที่เกิดขึ้นตามลำดับเวลา Sequence Diagram มี 2 มิติ คือแนวนอนจะแทนด้วย Object ต่างๆ เป็นรูปสี่เหลี่ยมผืนผ้าและแนวตั้งจะแทนด้วยเวลาเรียกว่าแกนเวลา (lifeline) เป็นเส้นประ (dash line ) เพื่อกำหนดขอบเขตการทำงานของ Object โดยจะมีการส่งผ่านข้อความจาก Object หนึ่งไปยังอีก Object หนึ่งตามลำดับของเวลาที่ผ่านไป ดังรูป 2.14 แต่ละข้อความแสดงเป็นลูกศรระหว่างแกนเวลาของ 2 Object

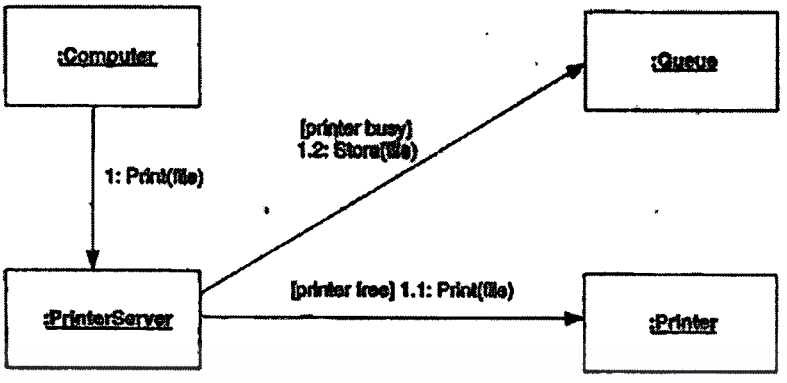



**รูปที่ 2.14 แสดง Sequence Diagram**

**2.1.2.2 Collaboration Diagram**

Collabolation Diagram เป็นที่รวมของ Object ที่สัมพันธ์กันในลักษณะข้อความ และความสัมพันธ์กันระหว่าง Object มีการแลกเปลี่ยนข้อความไปมาระหว่าง Object เพื่อให้ได้ผลลัพธ์ที่ต้องการ ลำดับถูกแสดงเป็นหมายเลขข้อความดังรูป 2.15 Collaboration Diagram มีการทำงานเหมือนกับ Sequence Diagram แต่มีรูปแบบไดอะแกรมที่ต่างกัน Collaboration Diagram เน้นที่ Object ที่ส่งข้อความไม่มีลำดับเวลา ส่วน Sequence Diagram จะเน้นเวลาของข้อความที่ส่งไปมา ดังนั้นสามารถเลือกแสดงได้ในรูปแบบใดรูปแบบหนึ่ง จึงเรียกว่า Interaction Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

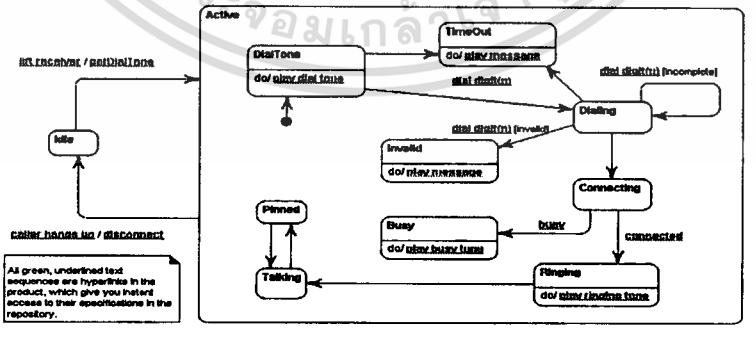



**รูปที่ 2.15 แสดง Collaboration Diagram**

โดยทั่วไปแล้วอาจใช้ Sequence Diagram หรือ Collaboration Diagram ในการแสดงเหตุการณ์หนึ่งๆ ของ Use Case ได้ ขึ้นอยู่กับความเหมาะสมและสถานการณ์

**2.1.2.3 State Diagram**

State Diagram แสดงเป็นลำดับของสถานะที่ Object นั้นทำงานในช่วงชีวิตของมันเพื่อตอบสนองต่อข้อความและภายนอกที่มากระตุ้นแสดงดังรูปที่ 2.16 ส่วนใหญ่จะเขียนในกรณีที่ระบบมีความซับซ้อน เป็นตัวช่วย Sequence Diagram เพื่อให้ทราบสถานะชัดเจนมากขึ้น สัญลักษณ์จุดเริ่มต้นของ State Diagram คือวงกลมทึบ จุดสิ้นสุดแทนด้วยสัญลักษณ์วงกลมทึบเล็กๆอยู่ในอีกวงกลมหนึ่งซึ่งอาจไม่จำเป็นต้องมีจุดสิ้นสุดก็ได้ สถานะแสดงเป็นรูปสี่เหลี่ยมขอบมน ใช้ลูกศรแสดงถึงการเปลี่ยนแปลงระหว่างสถานะ หรือแสดงเหตุการณ์ที่ทำให้สถานะนั้นเปลี่ยน

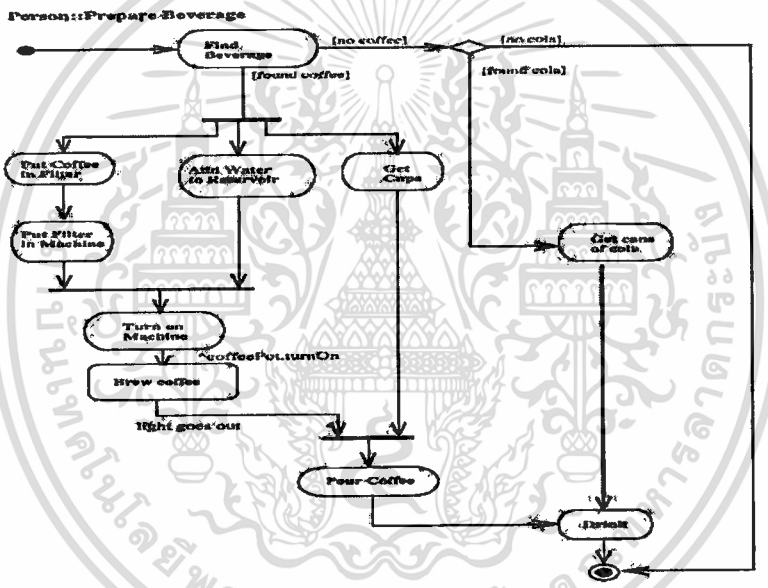



**รูปที่ 2.16 แสดง State Diagram**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.1.2.4 Activity Diagram**

Activity Diagram แสดงถึงสถานะที่เป็นกิจกรรม ที่แสดงให้เห็นประสิทธิภาพของ Operation และมีการส่งผ่านที่ถูก Trigger โดย Operation แสดงการกระทำต่างๆ และผลลัพธ์ที่เกิดขึ้นในรูปแบบของสถานะของ Object ซึ่งต่างจาก State Diagram Activity Diagram Activity Diagram ถูกใช้เป็นแบบจำลองการทำงานของระบบทั้งหมดเพื่อให้มองการส่งผ่านและการทำงานใน use case หรือระหว่าง Class รูปแบบ Activity Diagram จะคล้ายกับ State Diagram กิจกรรมแสดงสัญลักษณ์เป็นรูปสี่เหลี่ยมมนซึ่งมีชื่อของ Operation และมีการกำหนดเงื่อนไขโดยใช้รูปสี่เหลี่ยมขนมเปียกปูนแสดงดังรูป 2.17




**รูปที่ 2.17 แสดง Activity Diagram**

**2.1.3 โครงสร้างภายนอก**

เป็นการอธิบายส่วนซอฟต์แวร์ (software) และฮาร์ดแวร์(hardware) โดยมีการแปลงจากโครงสร้างภายในมาเป็นโครงสร้างภายนอก ซึ่งจะเป็นการเปลี่ยนรูปแบบของ Class ให้อยู่ในรูปแบบของ Components, Process และคอมพิวเตอร์ในโครงสร้างภายนอก โดยมีไดอะแกรมคือ Component Diagram และ Deployment Diagram

**2.1.3.1 Component Diagram**

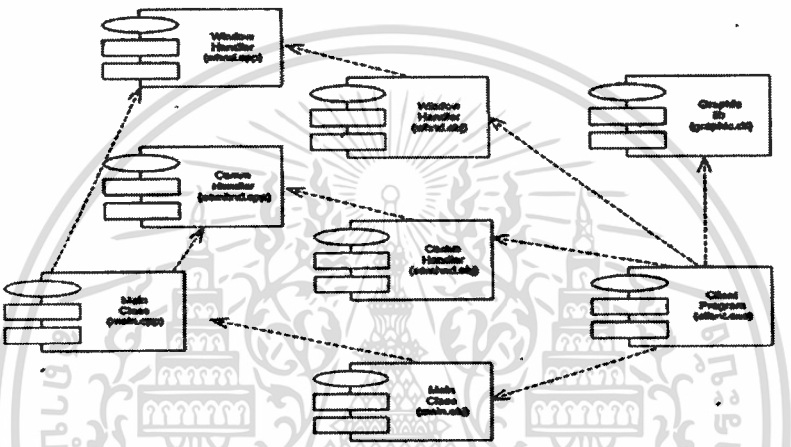
Component แสดงหลักการของโครงสร้างภายนอก และหน้าที่ที่ถูกกำหนดโดยโครงสร้างภายใน และแสดงองค์ประกอบของซอฟต์แวร์ที่ขึ้นต่อกัน โดยใน Component มีส่วนประกอบคือ

- source code components เป็นโค้ดไฟล์ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- binary code components เป็น Object โค้ดไฟล์
- executable components เป็น Execute ไฟล์

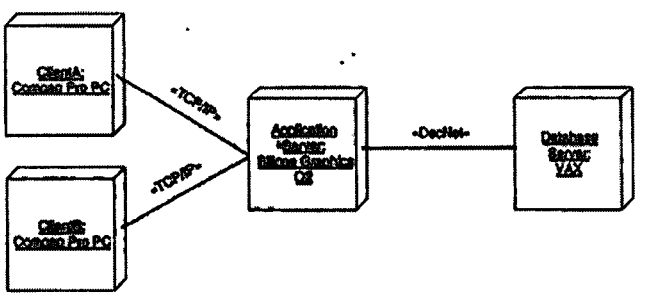
Component Diagram เป็นแบบจำลองของ component ที่ติดต่อกันโดยความสัมพันธ์แบบ Dependency Component Diagram จะมีโครงสร้างภายนอกแสดงดังรูปที่ 2.18 มีการใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้าที่มีรูปวงรีและสี่เหลี่ยมผืนผ้าเล็กๆสองอันอยู่ทางด้านซ้าย แทน Component และใช้สัญลักษณ์เส้นประที่มีลูกศรหัวเปิดแทนการเชื่อมต่อกันระหว่าง Component



**รูปที่ 2.18 แสดง Component Diagram**

**2.1.3.2 Deployment Diagram**

Deployment Diagram แสดงถึงสถาปัตยกรรมของระบบ ที่อธิบายลักษณะโปรแกรมในช่วง run-time รวมถึงการติดต่อสื่อสารกันระหว่างโหนด (Node) ซึ่งเป็นโครงสร้างที่รวมโปรเซสเซอร์ อุปกรณ์ต่างๆ และซอฟต์แวร์ ต่างๆไว้ด้วยกัน โหนดอาจจะมีอินสแตนของ Component ที่แสดงให้เห็นว่า Component นี้ทำงานอยู่บนโหนด โหนดใช้สัญลักษณ์ที่ใช้กล่องสี่เหลี่ยม และใช้สัญลักษณ์เส้นทึบแทนการเชื่อมโยงกันระหว่างโหนด ซึ่งแสดงดังรูปที่ 2.19



**รูปที่ 2.19 แสดง Deployment Diagram**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 การพัฒนาระบบเชิงวิเคราะห์

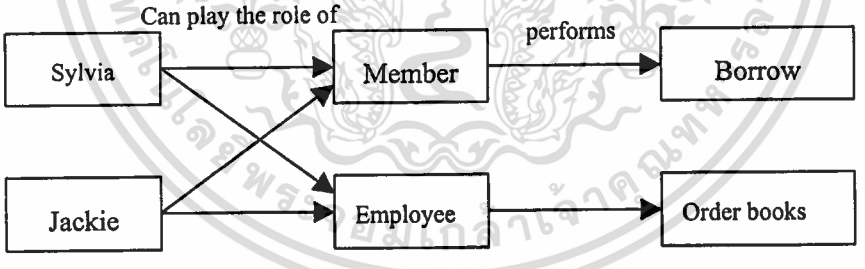
ในการพัฒนาระบบเชิงวิเคราะห์จะแตกต่างจากการวิเคราะห์ระบบแบบเก่าซึ่งมีพื้นฐานอยู่บนฟังก์ชั่น (function) และโพรซีเยอร์(procedure) การพัฒนาระบบเชิงวิเคราะห์จะมีการสร้างโม ดูล (Module) ที่พร้อมจะใช้งานได้ทำงานภายในตัวเองได้ หรือเรียกว่า Object ทำให้สามารแก้ไขหรือนำกลับไปใช้ได้อีก วิธีการแบบรวม(unified approach, UA) เป็นวิธีการในการพัฒนาระบบเชิง Object และมีการกำหนดขั้นตอนในการพัฒนาระบบ คือ วิเคราะห์ ออกแบบ สร้างระบบและทดสอบ ซึ่งแต่ละขั้นตอนหลักจะมีขั้นตอนย่อยๆ อีก

### 2.2.1 การวิเคราะห์ระบบ

การวิเคราะห์ระบบเชิงวัตถุ เกี่ยวข้องกับการหาความต้องการของระบบ หา Class และความสัมพันธ์ระหว่าง Class การที่จะเข้าใจระบบจะมีการหาผู้ใช้ระบบหรือ actor ว่าใครคือ actor และเขาใช้ระบบอย่างไร โดยมีการหา use case ซึ่งเป็นลักษณะการติดต่อระหว่างผู้ใช้กับระบบที่หาความต้องการของผู้ใช้ระบบโดยการใช้วิธีการต่างๆในการหาข้อมูล เช่น พิจารณาจากเอกสารที่มีอยู่ การสัมภาษณ์ การทำแบบสอบถาม และการสังเกต ในการวิเคราะห์ระบบจะมีขั้นตอนดังนี้

#### 2.2.1.1 หา actor ของระบบ

การหา actor ของระบบจะมีการหาว่าใครคือผู้ใช้ระบบ actor เป็นการแทนบทบาทของผู้ใช้ระบบ ผู้ใช้ระบบอาจจะแสดงได้หลายบทบาทแต่ actor สามารถแทนผู้ใช้หนึ่งคนและแสดงบทบาทเดียวในแต่ละ use case ซึ่งแสดงดังรูปที่ 2.20




**รูปที่ 2.20 แสดงความแตกต่างระหว่างผู้ใช้กับ actor**

การหา actor อาจเกิดจากคำถามดังนี้

- ใครเป็นคนใช้ระบบ
- ใครมีผลกระทบต่อระบบ
- ระบบภายนอกระบบไหนที่ใช้ระบบทำงาน
- ปัญหาอะไรที่ระบบเข้าไปทำเพื่อแก้ปัญหา
- ผู้ใช้ใช้ระบบอย่างไร

actor ไม่เพียงแต่เป็นผู้ใช้ระบบเท่านั้น อาจจะเป็นระบบภายนอกที่ต้องการข้อมูลจากระบบ หรือเป็นฮาร์ดแวร์ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.1.2 ทำแอคติวิตี้ ไดอะแกรม**

การทำแอคติวิตี้ ไดอะแกรมเพื่อใช้พัฒนาแบบจำลองขั้นตอนทางธุรกิจอย่างง่ายๆ ทำให้เราเข้าใจระบบอย่างคร่าวๆ ทำให้มีเป้าหมายในการพัฒนา use case ต่อไป

**2.2.1.3 พัฒนา use case**

การพัฒนา use case โดยหาว่าผู้ใช้ทำอะไรกับระบบและติดต่อกับระบบอย่างไร ขั้นตอนในการทำ use case มีดังนี้

1 แต่ละ actor หางานและหน้าที่ที่ actor ทำ หรือระบบนั้นต้องการให้ actor ทำ use case ควรจะตอบสนองต่องานนั้นอย่างชัดเจน

2 การตั้งชื่อของ use case ควรจะตั้งชื่อในลักษณะที่อธิบายหน้าที่ของ use case ทั่วๆ ไป ชื่อควรจะแสดงถึงการกระทำ มักจะอยู่ในรูปแบบของกิริยา (verb) เช่น borrows หรือ กิริยากับคำนาม เช่น Borrow books

3 อธิบายการทำงานของ use case อย่างย่อๆ ในรูปแบบที่เข้าใจง่าย

ผู้ใช้ระบบหลายคนสามารถเป็น actor ที่ต่างกัน และสามารถแสดงได้บทบาทเดียว ดังนั้นในแต่ละ use case จะมี actor หลักซึ่งควรจะมีการทำดังนี้

- แยกผู้ใช้ออกจาก actor
- แยก actor ออกจาก actor อื่น (แยกหน้าที่ของแต่ละ actor )
- แยก use case ที่มี actor เริ่มต้นที่แตกต่างกัน หรือมีพฤติกรรมที่ต่างกัน

**2.2.1.4 เตรียม Interaction Diagram**

การเตรียม Interaction Diagram เป็นการหา Sequence Diagram และ Collaboration Diagram ที่อธิบายถึงวิธีการทำงานร่วมกัน ของกลุ่มของ Object เพื่อให้ได้งานหนึ่งๆ มา ซึ่งจะแสดงถึงรูปแบบพฤติกรรมของ Object ที่โต้ตอบกันใน Use Case หนึ่ง

**2.2.1.5 การแยกประเภท**

การแยกประเภทเป็นการพัฒนา Class ไดอะแกรมของเชิงสแตติกโดยจะมีการกระทำดังนี้

- หา Class เป็นกลไลสำคัญในการแยกประเภทของ Object เป็นลักษณะเฉพาะของโครงสร้างพฤติกรรม และใช้อธิบาย Object มีการใช้วิธีการของ use case driven ในการหา Class
- หาความสัมพันธ์ระหว่าง Class (relationships) ซึ่งมีหลายความสัมพันธ์คือ
  - ความสัมพันธ์แบบ association จะมีรูปแบบ เช่น next to contained in order from talk to เป็นต้น บางครั้งความสัมพันธ์ไม่สอดคล้องกับระบบ ดังนั้นจึงมีการกำจัดความสัมพันธ์ที่ไม่จำเป็น คือ
    - ความสัมพันธ์ในขณะสร้างระบบ (implementation association) เช่น User Interface อยู่ในช่วงของการออกแบบ ยังไม่ต้องใช้ตอนนี้
    - ควรหลีกเลี่ยงความสัมพันธ์ระหว่าง Class หลาย Class (Ternary association)
  - ความสัมพันธ์แบบ generalization มีวิธีการในการหาหลายวิธีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Top-down เป็นลักษณะแบบประกอบด้วยหลายคุณศัพท์ (adjective) ในชื่อของ Class ถ้ามีลักษณะเฉพาะควรจะพิจารณาเป็น Subclass
- Bottom-up เป็น Class ที่มี Attribute หรือ Method (method) คล้ายกัน ทำให้สามารถรวมเป็น Attribute หรือ Method ไปเป็น Abstract Class
- การย้าย Attribute และ Method ให้อยู่ในชั้นที่สูงที่สุดเท่าที่เป็นไปได้ (Reusability)
- หลีกเลี่ยงการใช้การถ่ายทอดจากหลาย Class (Multiple inheritance)

- ความสัมพันธ์แบบประกอบกันจะมีลักษณะดังนี้
  - องค์ประกอบที่ถูกสร้างขึ้นจากหลายส่วนประกอบเป็น Class ข้างบน (Assembly)
  - ความสัมพันธ์แบบ Container เช่น บ้านจะมีเฟอร์นิเจอร์บรรจุอยู่
  - ความสัมพันธ์แบบ Collection-Member เช่นทีมฟุตบอลเป็นที่รวมของนักฟุตบอล
- หา Attribute มีลักษณะดังนี้
  - Attribute ติดต่อกับคำนามตามด้วยบุพบท เช่น cost of
  - ทำ Class ให้ง่าย มี Attribute เพียงพอในการระบุสถานะของ Object
  - Attribute จะต้องมีการอธิบายครบสมบูรณ์ในปัญหา
  - ละเว้น Attribute ที่ได้มา เช่น ไม่ควรใช้เรื่องเวลาที่เลยมาจากการสั่งซื้อ แต่ควรใช้เวลาของการสั่งซื้อ หรืออายุซึ่งควรจะเก็บวันเกิดดีกว่า
  - ไม่ควรมี Attribute ที่มากเกินไป ไม่แน่ใจยังไม่ต้องเติม เพราะยังสามารถเติม Attribute เข้าไปใหม่ได้
- หา Method ซึ่งจะทำการตอบสนองต่อการคิวรี การอ่าน การเขียนที่เกี่ยวกับการจัดการค่าของ Attribute

#### 2.2.1.6 การกระทำซ้ำเพื่อปรับแต่งระบบ

การทำซ้ำเพื่อที่จะปรับปรุงแก้ไขความถูกต้องของระบบเพื่อให้ระบบมีความถูกต้องมากขึ้น

### 2.2.2 การออกแบบระบบ

การออกแบบระบบหลังจากที่มีการวิเคราะห์ระบบแล้ว เป็นการออกแบบ Class ที่กำหนดในขั้นตอนการวิเคราะห์ มีการกำหนด attribute method และความสัมพันธ์ระหว่าง Class และมีการทำ User Interface ซึ่งจะมีการหา Object และ Class เพิ่มเติม การออกแบบที่ดีคือการทำให้ง่ายและชัดเจนในการสร้างระบบ การออกแบบระบบประกอบด้วยขั้นตอนดังนี้

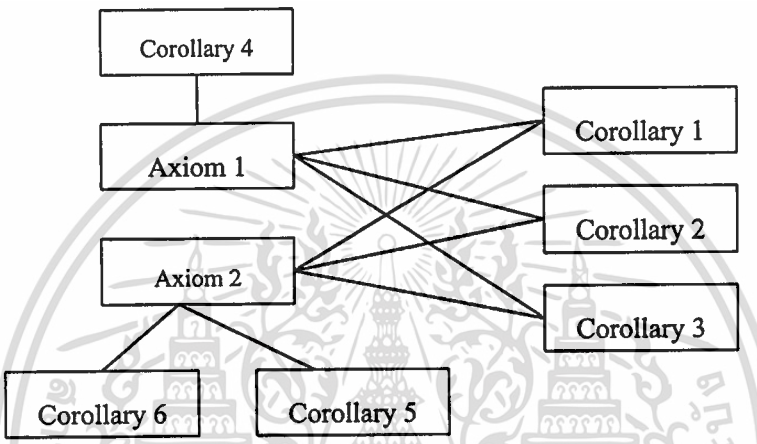
#### 2.2.2.1 การใช้ Axioms ในการออกแบบ

Axiom เป็นความจริงพื้นฐานที่พิสูจน์แล้วว่าเป็นความจริง และไม่มีค่ายกเว้น ส่วนCorollary เป็นผลสืบเนื่องจาก Axiom และพิสูจน์แล้วว่าจริง Axiom ที่ออกแบบมี 2 ข้อ คือ

1. Axiom1 เป็นความพยายามทำให้ส่วนประกอบต่างๆ มีอิสระจากกันมากสุด
2. Axiom2 ในแต่ละ Class ควรมีข้อมูลที่จำเป็นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Axiom ในการออกแบบทั้ง 2 อาจนำไปสู่ที่มาของ Corollary ดังรูปที่ 2.21 ซึ่ง Corollary คือส่วนขยายที่ต่อมาจาก Axiom หรือส่วนขยายที่มาจากส่วนอื่นๆ ที่ได้ผ่านการพิสูจน์มาแล้ว Corollary อาจมีส่วนช่วยในการตัดสินใจในการออกแบบด้วย เพราะ Corollary เหล่านั้นได้นำมาใช้ในสถานการณ์จริงซึ่งใช้ง่ายกว่า Axiom หรืออาจเรียก Corollary ว่าเป็นกฎในการออกแบบ แต่ทั้งหมดนั้นมีที่มา มาจาก Axiom ทั้ง 2 นั่นเอง




**รูปที่ 2.21 แสดง Axiom และ Corollary**

Corollary ที่เกิดขึ้นจะใช้เป็นหลักเกณฑ์ในการใช้ออกแบบ Class มีดังนี้

- Corollary 1 เกิดจาก Axiom 1 และ Axiom 2 โดยออกแบบให้มีข้อมูลน้อยที่สุดและลักษณะเป็น Uncoupled เป็นการใส่ข้อมูลที่จำเป็นเท่านั้นใน Class ที่จำเป็นเท่านั้น
- Corollary 2 เกิดจาก Axiom 1 และ Axiom 2 โดย Class แต่ละ Class ที่สร้างขึ้นควรมีจุดประสงค์เดียว
- Corollary 3 เกิดจาก Axiom 1 และ Axiom 2 โดยมี Class ที่ง่ายๆ จำนวนมากไม่มี Method มากจะช่วยในการนำกลับไปใช้ได้อีก
- Corollary 4 เกิดจาก Axiom 1 โดยให้มีการแทนกันอย่างลงตัว (Strong Mapping) เป็นการบอกถึงความสัมพันธ์ระหว่าง Object ที่ได้มาจากการวิเคราะห์ กับ Object ที่ใช้ในการออกแบบ ซึ่งในขั้นการออกแบบจะเป็นเพียงการเพิ่มรายละเอียดและยังคงรักษาความจำเป็นที่เหมือนกันเอาไว้
- Corollary 5 เกิดจาก Axiom 2 โดยให้มีมาตรฐานในการออกแบบ ซึ่งใช้ในการติดต่อสื่อสารระหว่าง Component และสามารถนำกลับมาใช้ใหม่ได้
- Corolllary 6 เกิดจาก Axiom 2 โดยให้การออกแบบร่วมกับการถ่ายทอดไปด้วยกัน Method ที่ซับซ้อนควรอยู่ที่ Superclass

มีการนำ Axiom มาประยุกต์ในการออกแบบ Class Attribute และ Method ความสัมพันธ์ โครงสร้าง และโปรโตคอลโดยมีการกระทำดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับปรุง Class ไดอะแกรม โดยการเพิ่มรายละเอียดของ Class Diagramประกอบด้วยขั้นตอนดังนี้

- ปรับปรุง Attribute
- ออกแบบ Method และโปรโตคอลเพิ่มเติม
- ปรับปรุงความสัมพันธ์ระหว่าง Class
- ปรับปรุงการถ่ายทอด Class
- การปรับปรุงและทำซ้ำ

**2.2.2.2 การออกแบบ Access Layer**

เป็นการออกแบบ Class ที่ทำหน้าที่เป็นผู้เข้าถึงข้อมูลของระบบ ซึ่งอาจใช้ขั้นตอนดังนี้

- สร้าง Class ใหม่ (mirror class) จากทุก Class ทางที่สร้างขึ้นมา จะทำหน้าที่ถึงข้อมูลของระบบ
- กำหนดความสัมพันธ์ระหว่าง Class ในAccess Layer
- ทำ Class และความสัมพันธ์ระหว่าง Class ให้มีความง่ายเพื่อกำจัด Class และโครงสร้างที่ซ้ำซ้อน
  - Class ที่ซ้ำซ้อนคือ อย่ามี Class 2 Class ที่แสดงกิจกรรมที่เหมือนกัน ให้เลือกเพียงหนึ่ง Class และกำจัดอีก Class ไป
  - Class ที่ประกอบด้วย Method เดียว หรือ 2 Method ควรรวมกับ Class ที่มีอยู่ได้
- ปรับปรุงและทำซ้ำ

หลังจากที่ได้ Class ที่สมบูรณ์แล้วจะมีการนำไปทำเป็นโครงสร้างในฐานข้อมูลแบบRelational โดยมีการแปลงจาก Class ไปเป็นตารางในฐานข้อมูล ซึ่งการแปลงก็มีหลายแบบ เช่น

- หนึ่ง Class แปลงไปเป็นหนึ่งตาราง
- หลาย Class แปลงไปเป็น 1 ตาราง
- Class ที่เป็นการถ่ายทอดแปลงไปเป็น 1 ตาราง
- Class ที่เป็นการถ่ายทอดแปลงไปเป็นหลายตาราง

**2.2.2.3 การออกแบบ View Layer**

- ออกแบบ User Interface ในระดับ Macro เป็นการหา Object ของ Vieew Layer หรือกำหนด Class ที่มีหน้าที่โต้ตอบระหว่างผู้ใช้งานกับระบบ โดยหาได้จากการวิเคราะห์ Use Case ในขั้นการวิเคราะห์ และ Sequence Diagram หรือ Collaboration Diagram
- ออกแบบ User Interface ระดับ Micro อาจจะใช้หลักการต่อไปนี้ เป็นแนวทางในการช่วยออกแบบ User Interface ที่ได้มา
  - ทำให้ชัดเจนเรียบง่าย (ประยุกต์ Corollary 2) ทำให้ User Interface เรียบง่ายและชัดเจน จนผู้ที่ไม่รู้เรื่องเกี่ยวกับเครื่องมือและกลไกของระบบสามารถทำงานได้
  - ทำให้เป็นธรรมชาติ (ประยุกต์ Corollary 4) ผู้ใช้งานปกติจะสามารถคาดเดาเหตุการณ์ที่จะเกิดขึ้นกับ User Interface ได้ โดยเสมือนไม่ได้ทำงานกับระบบอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำให้ผู้ใช้งานเป็นผู้ควบคุม Software (ประยุกต์ Corollary 1) อาจเป็นการเตรียมงานที่เป็นอัตโนมัติ เพื่ออำนวยความสะดวกให้กับผู้ใช้งาน หรืออาจเป็นการเพิ่มคำแนะนำหากผู้ใช้มีการทำผิดพลาด เป็นต้น ยังรวมถึงทำให้ User Interface มีลักษณะเหมือนกันทั้งระบบซึ่ง
- ทดสอบการใช้งานและผู้ใช้พอใจ
- ปรับปรุงและแก้ไข

**2.2.2.4 การทำซ้ำและปรับปรุงการออกแบบทั้งหมด**



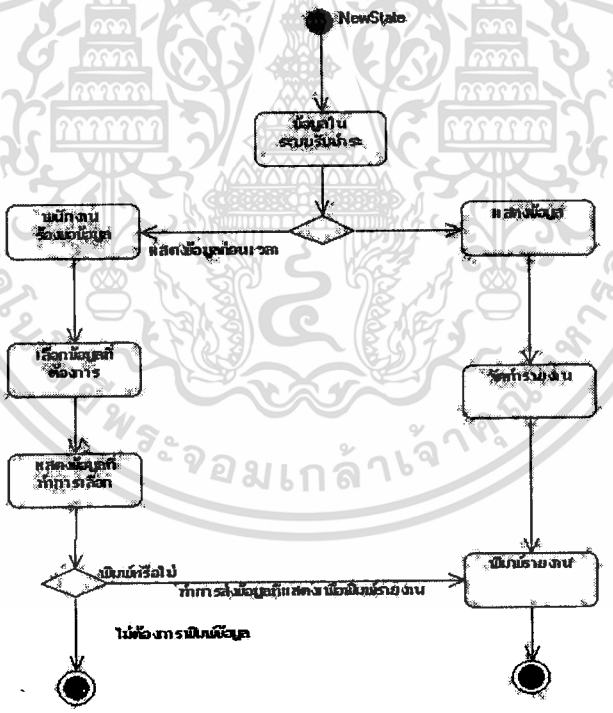
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3

# การวิเคราะห์และออกแบบ

## 3.1 แบบจำลองของกระบวนการทางธุรกิจ

ระบบเช่าซื้อเป็นระบบที่ทำการแสดงรายละเอียดของลูกค้า และการรับชำระเงินในระบบเช่าซื้อและทำการออกรายงาน ในการพัฒนาแบบจำลองของระบบเช่าซื้อเชิงธุรกิจจะมีการใช้ Activity Diagram มาช่วยในการอธิบายกิจกรรมโดยรวมของระบบ ซึ่งแสดงดังรูปที่ 3.1




**รูปที่ 3.1แสดง Activity Diagram ของระบบเช่าซื้อ**

## 3.2 Use Case Diagram ของระบบ

การหา Actor และ Use Case ของระบบนั้นจะต้องผ่านการศึกษาความต้องการของระบบ และผ่านการทำซ้ำหลายต่อหลายรอบ พร้อมกับกำหนด Actor และ Use Case ไปด้วยกันในแต่ละรอบด้วยซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจจะมีการปรับปรุงเปลี่ยนแปลง Actor และ Use Case ด้วย Actor และ Use Case ของระบบมีดังนี้

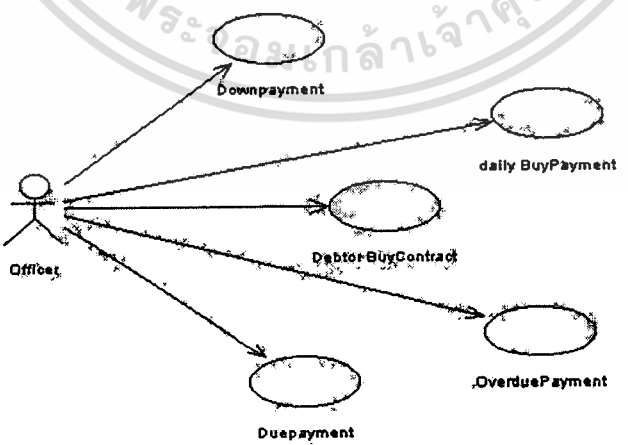
**3.2.1 Actor**

- Officer พนักงานของระบบเช่าซื้อ ทำหน้าที่ในการตรวจสอบการรับชำระเงินของระบบเช่าซื้อ และทำการออกรายงาน

**3.2.2 Use Case**

- DownPayment (การรับชำระเงินล่วงหน้า) มีหน้าที่ในการตรวจสอบการรับชำระเงินล่วงหน้าประจำวันของลูกหนี้
- Daily BuyPayment (การรับชำระค่าเช่าซื้อ) มีหน้าที่ในการตรวจสอบการรับชำระค่าเช่าซื้อประจำวันของลูกหนี้
- DebtorBuyContract (สัญญาของลูกหนี้เช่าซื้อ) มีหน้าที่แสดง ประวัติ และสัญญาของลูกค้าที่ทำการเช่าซื้อ
- OverduePayment (การค้างชำระค่าเช่าซื้อ) มีหน้าที่ในการตรวจสอบการค้างชำระของลูกค้าเป็นรายเดือน
- DuePayment (ค่าเช่าซื้อถึงกำหนดชำระ) ทำหน้าที่ตรวจสอบเมื่อครบกำหนดชำระจะแสดงลูกค้าที่ต้องชำระเงินที่เกี่ยวข้องกับในสัญญาเป็นประจำเดือน

หลังจากที่กำหนด Actor และ Use Case แล้วจึงทำการสร้างความสัมพันธ์ระหว่าง Actor และ Use Case ซึ่งแสดงเป็น Use Case Diagram ดังรูปที่ 3.2




**รูปที่ 3.2 แสดง Use Case Diagram ของระบบเช่าซื้อ**

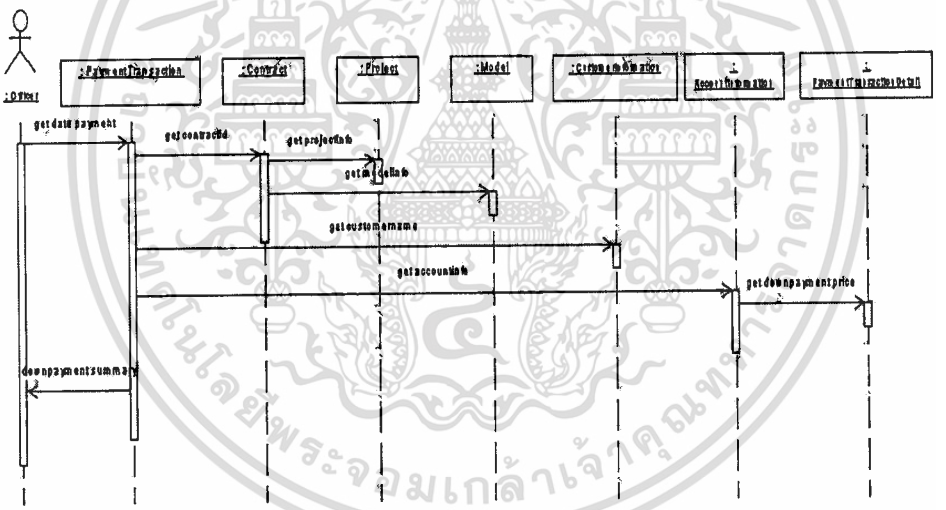
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.3 อินเทอร์แอกชั่น ไดอะแกรม (Interaction diagram)

ในการแสดงเหตุการณ์หรือหน้าที่การทำงานของระบบเช่าซื้อ จะพิจารณาแต่ละ Use Case ที่ได้กำหนดไว้แล้วนำมาแสดงลำดับเหตุการณ์ต่างๆ ที่เกิดขึ้นใน Use Case นั้น ซึ่งจะใช้อินเตอร์แอกชั่น ไดอะแกรมในการแสดงเหตุการณ์ที่เกิดขึ้นซึ่งมี 2 ไดอะแกรมคือ Sequence Diagram และ Collaboration Diagram โดยสามารถเลือกไดอะแกรมใดก็ได้ในการแสดงหรือจะใช้ทั้งสองไดอะแกรมช่วยในการแสดงก็ได้ ซึ่งจะพิจารณาในแต่ละ Use Case ดังนี้

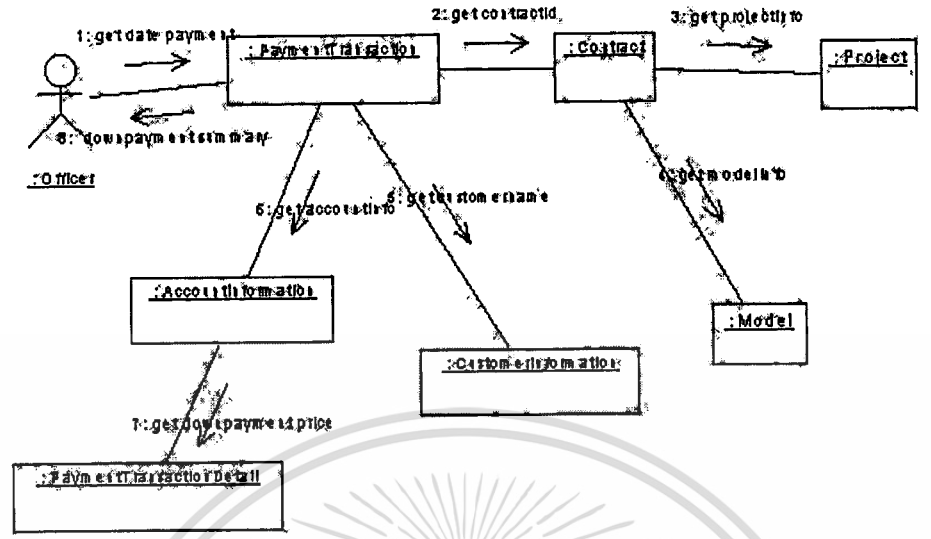
### 3.3.1 Use Case DownPayment

- Sequence Diagram ของ Use Case DownPayment แสดงลำดับการเหตุการณ์ของการชำระเงินล่วงหน้าแสดงในรูปที่ 3.3
- Collaboration Diagram ของ DownPayment แสดงดังรูปที่ 3.4




**รูปที่ 3.3 แสดง Sequence Diagram ของ Use Case DownPayment**

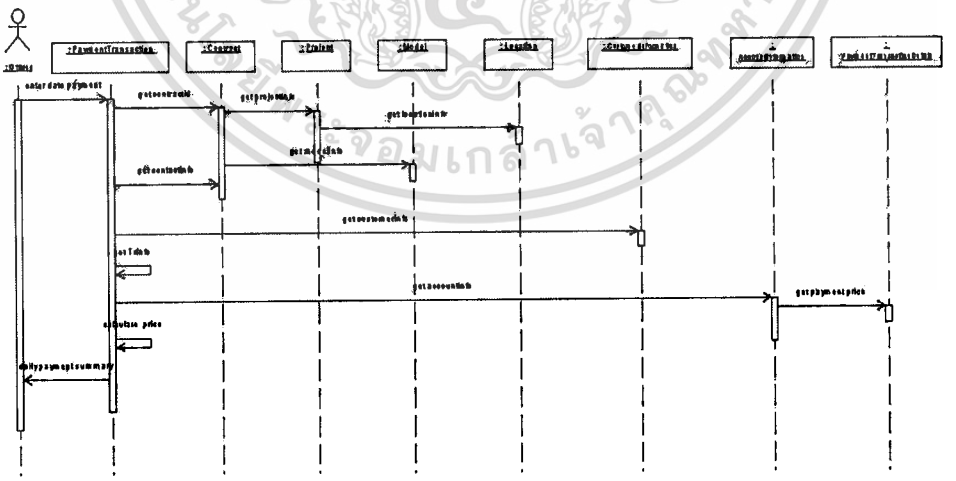
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.4 แสดง Collaboration Diagram ของ Use Case DownPayment**

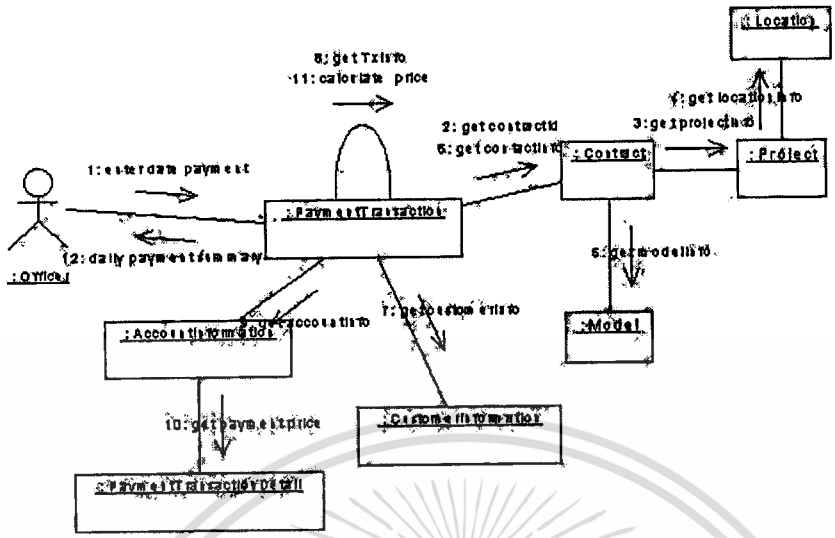
### 3.3.2 Use Case Daily BuyPayment

- Sequence Diagram ของ Use Case Daily BuyPayment แสดงลำดับการเหตุการณ์ของการรับชำระค่าเช่าซื้อแสดงในรูปที่ 3.5
- Collaboration Diagram ของ Use Case Daily BuyPayment แสดงดังรูปที่ 3.6



**รูปที่ 3.5 แสดง Sequence Diagram ของ Use Case DailyBuyPayment**

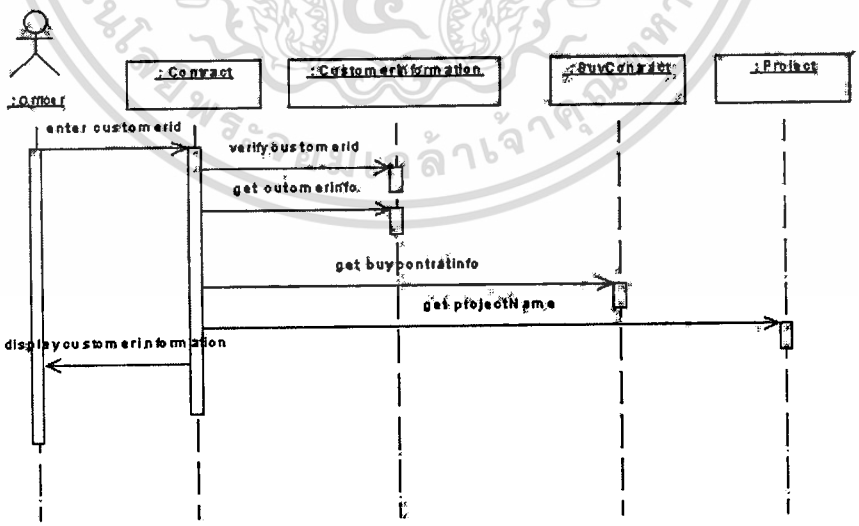
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.6 แสดง Collaboration Diagram ของ Use Case Daily BuyPayment**

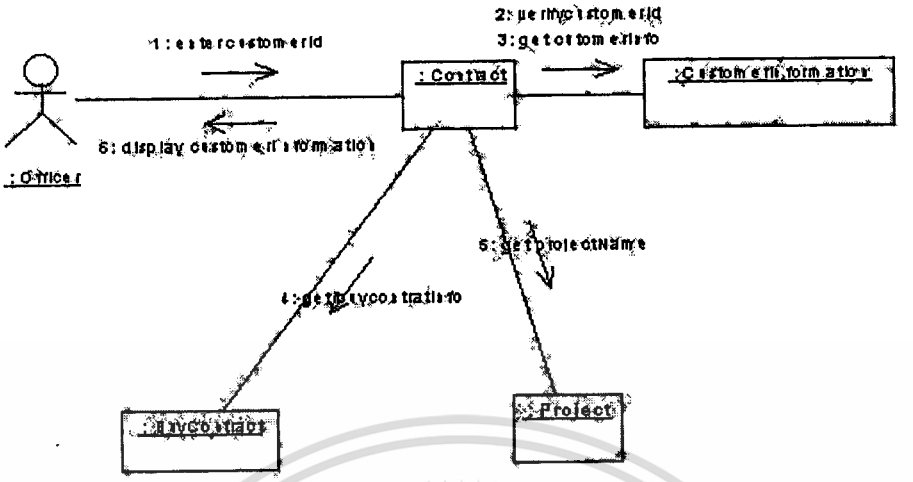
### 3.3.3 Use Case Debtor BuyContract

- Sequence Diagram ของ Use Case DebtorBuyContract แสดงลำดับการเหตุการณ์ของการสัญญาของลูกค้าที่เช่าซื้อแสดงในรูปที่ 3.7
- Collaboration Diagram ของ Use Case DebtorBuyContract แสดงดังรูปที่ 3.8




**รูปที่ 3.7 แสดง Sequence Diagram ของ Use Case DebtorBuyContract**

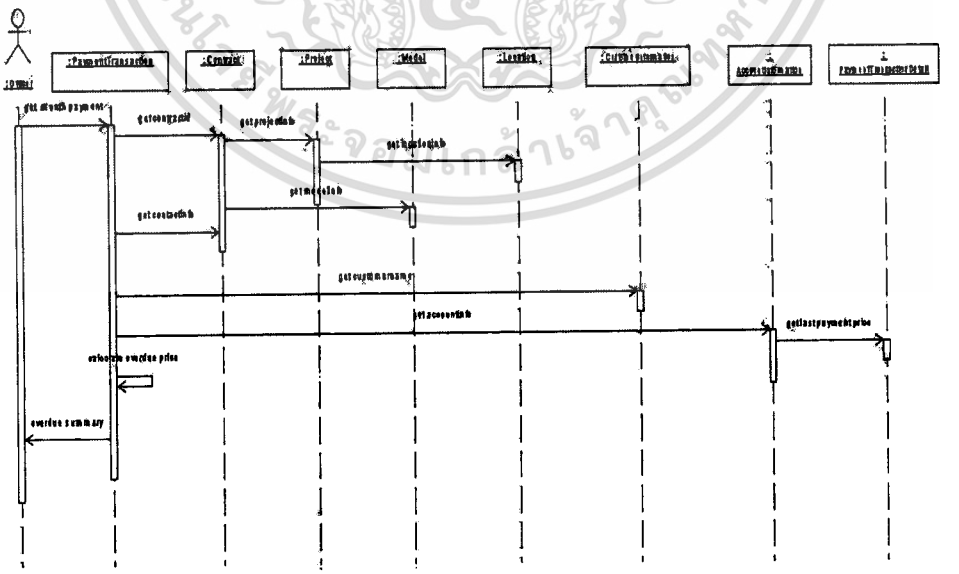
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.8 แสดง Collaboration Diagram ของ Use Case DebtorBuyContract**

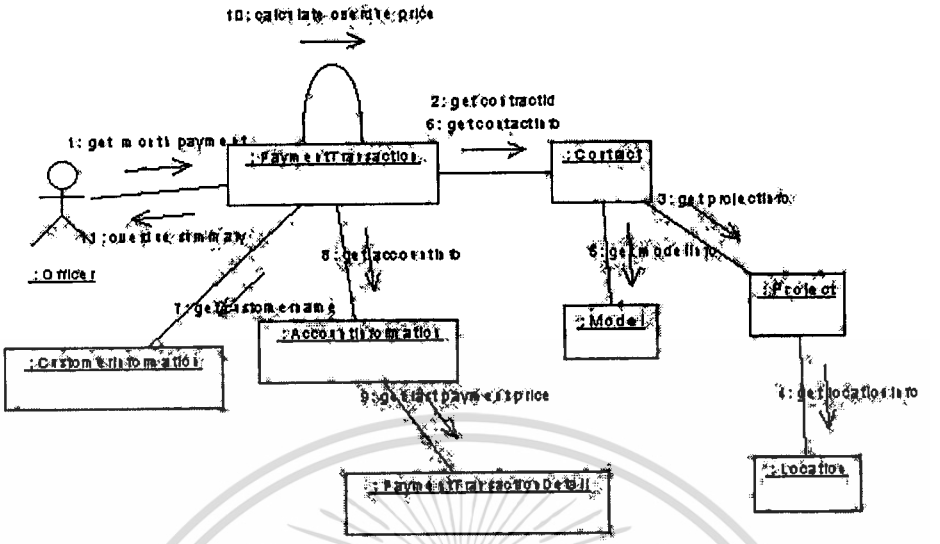
### 3.3.4 Use Case OverduePayment

- Sequence Diagram ของ Use Case OverduePayment แสดงลำดับการเหตุการณ์ของการค้างชำระค่าเช่าซื้อแสดงในรูปที่ 3.9
- Collaboration Diagram ของ Use Case OverduePayment แสดงดังรูปที่ 3.10



**รูปที่ 3.9 แสดง Sequence Diagram ของ Use Case OverduePayment**

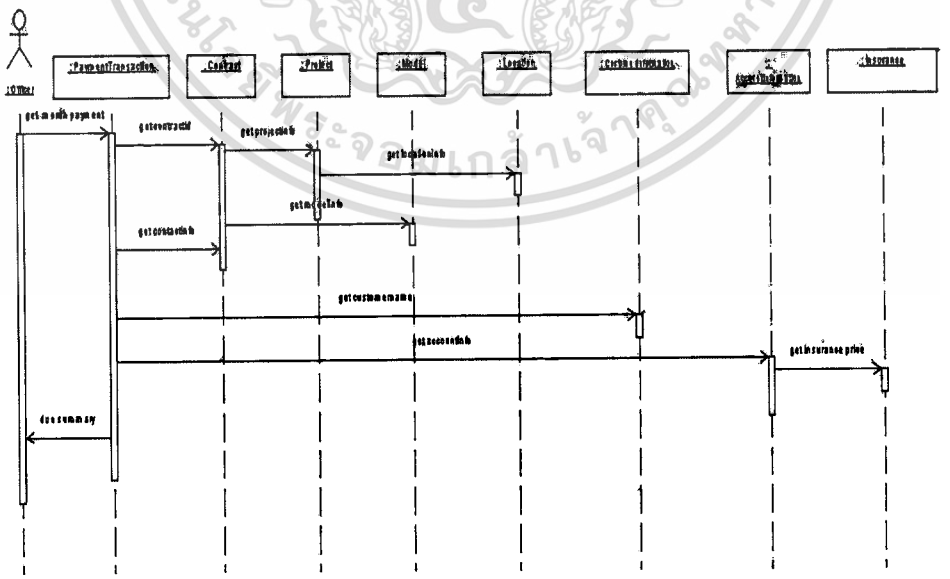
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.10 แสดง Collaboration Diagram ของ Use Case OverduePayment**

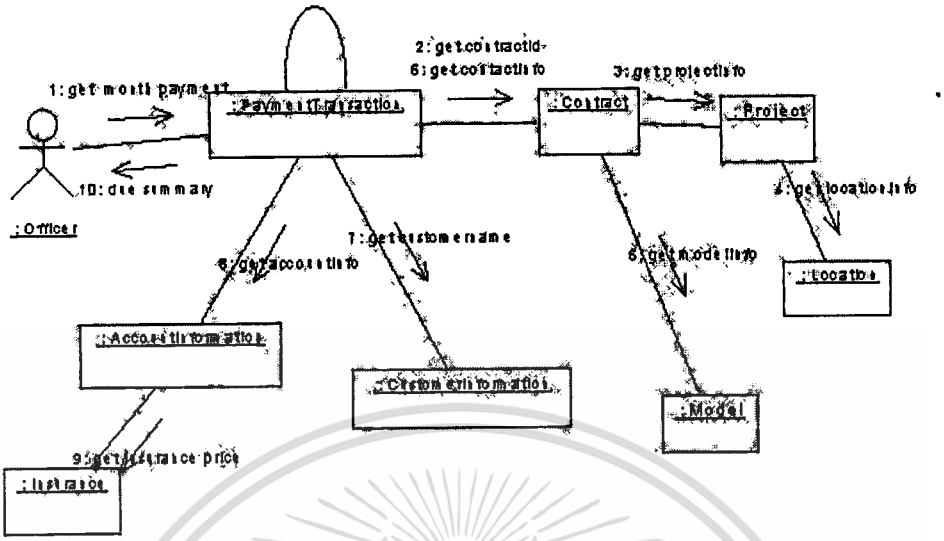
### 3.3.5 Use Case DuePayment

- Sequence Diagram ของ Use Case DuePayment แสดงลำดับการเหตุการณ์ของค่าเช่าซื้อเมื่อถึงกำหนดชำระแสดงในรูปที่ 3.11
- Collaboration Diagram ของ Use Case DuePayment แสดงดังรูปที่ 3.12



**รูปที่ 3.11 แสดง Sequence Diagram ของ Use Case DuePayment**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.12 แสดง Collaboration Diagram ของ Use Case DuePayment**

## 3.4 การกำหนด Class Diagram ในขั้นตอนการวิเคราะห์

### 3.4.1 การกำหนด Class ของระบบ

เมื่อทำการวิเคราะห์ Use Case Diagram และ Interaction Diagram ทำให้สามารถกำหนด Class ได้ดังตารางที่ 3.1

**ตารางที่ 3.1 แสดง Class ของระบบเช่าซื้อ**

| ชื่อ Class | ความหมาย |
|---|---|
| Location | รายละเอียดเกี่ยวกับที่ตั้งสำนักงานเคหะชุมชน |
| Project | รายละเอียดเกี่ยวกับโครงการ |
| Model | รายละเอียดเกี่ยวกับโมเดล |
| Building Type | รายละเอียดเกี่ยวกับสิ่งปลูกสร้าง |
| Unit | รายละเอียดเกี่ยวกับหน่วยอาศัย |
| Quotation | รายละเอียดข้อมูลขายโครงการ |
| BuyContract | รายละเอียดสัญญาเช่าซื้อ |
| CustomerInformation | รายละเอียดลูกค้า |
| Insurance | รายละเอียดค่าประกันการเช่าซื้อ |
| BuyDownPayment | รายละเอียดการจ่ายเงินดาวน์ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.1 แสดง Class ของระบบเช่าซื้อ(ต่อ)**

| | |
|---|---|
| AccountInformation | เป็นรายการบัญชีที่ใช้ระบบรับชำระ ซึ่งเป็นข้อมูลที่ต้องการในการบันทึกการรับชำระแต่ละรายการ |
| Contract | เป็นสัญญาของลูกหนี้ ที่ได้ทำไว้กับทางการเคหะ ซึ่งเป็นข้อมูลที่ต้องการในการรับชำระ |
| PaymentTransaction | เป็น Transaction การรับชำระของระบบ ซึ่งเป็นการเก็บรายละเอียดการรับชำระในแต่ละครั้ง |
| PaymentTransactionDetail | เป็นรายละเอียดของแต่ละ Transaction ของการรับชำระ |

**3.4.2 ความสัมพันธ์ของ Class**

- Contract กับ PaymentTransaction สัญญา (Contract) ของลูกหนี้ทุกสัญญาที่มีการทำไว้กับทางการเคหะ ลูกหนี้ที่ถือสัญญานั้นๆ จะต้องมีการจ่ายเงิน (PaymentTransaction) เกิดขึ้น และสัญญาหนึ่งๆ จะต้องมีการจ่ายเงินหนึ่งครั้งหรือมากกว่าหนึ่งครั้งขึ้นไป
- PaymentTransactionDetail กับ PaymentTransaction ในหนึ่งครั้งของการชำระเงินของลูกหนี้จะมีรายละเอียดการชำระเงิน (Payment-TransactionDetail) เกิดขึ้น เพราะการชำระเงินนั้นอาจมีการชำระได้ตั้งแต่หนึ่งรายการขึ้นไปภายในหนึ่งครั้ง หรือการชำระเงินหนึ่งรายการหรือหลายๆ รายการนั้นจะเกิดขึ้นภายในครั้งหนึ่งๆ เท่านั้นและหากไม่เกิดขึ้นก็จะไม่มีรายละเอียดรายการที่ชำระ
- PaymentTransactionDetail กับ AccountInformation การชำระแต่ละรายการที่ชำระก็จะมีการบอกถึงชนิดบัญชี (AccountInformation) ที่ลูกหนี้ชำระมาซึ่งรายการหนึ่งก็จะบอกถึงชนิดบัญชีหนึ่งเท่านั้นจึงทำให้เกิดความสัมพันธ์แบบ Association
- Location กับ Project ในเขตของสำนักงานเคหะชุมชนหนึ่ง มีหลายโครงการที่อยู่อาศัย ซึ่งเป็นความสัมพันธ์แบบ Association
- BuildingType กับ Model โมเดลแต่ละโมเดลจะอยู่ในประเภทสิ่งปลูกสร้างหนึ่งเท่านั้น เช่น โมเดล F1 เป็นสิ่งปลูกสร้างประเภท บ้านเดี่ยว เป็นความสัมพันธ์แบบ Association
- Project กับ Quotation โครงการหนึ่งจะมีการเสนอขายหลายรายการ ซึ่งรายการหนึ่งสำหรับแต่ละโมเดล เป็นความสัมพันธ์แบบ Association

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Quotation กับ Model ข้อมูลการขายแต่ละรายการจะอ้างถึงโมเดล เป็นความสัมพันธ์แบบ Association
- Quotation กับ Unit ในข้อมูลการขายหนึ่งรายการหรือ แต่ละโมเดล จะต้องประกอบด้วยหน่วยอาศัยหลายหน่วย เป็นความสัมพันธ์ Aggregation
- Contract กับ Unit ในสัญญา 1 สัญญาจะมีหน่วยอาศัยได้ 1 หน่วยอาศัย และ 1 หน่วยอาศัยจะมีอยู่ในใบสัญญาได้ หลายใบ เป็นความสัมพันธ์แบบ Association
- BuyContract กับ Contract สัญญาเช่าซื้อเป็นสัญญาที่มาจากสัญญา มีรายละเอียดเพิ่มเติมเกี่ยวกับการเช่าซื้อ เป็นความสัมพันธ์แบบ Generalization
- BuyDownPayment กับ BuyContract สัญญาเช่าซื้อประกอบด้วยเงินดาวน์หลายงวดและสัญญาเช่าซื้ออาจไม่มีเงินดาวน์ก็ได้ เป็นความสัมพันธ์แบบประกอบกัน
- Insurance กับ Contract สัญญา 1 สัญญามีค่าประกันได้หลายค่า และค่าประกันจะอยู่ในสัญญา 1 สัญญา เป็นความสัมพันธ์แบบ Aggregation
- Insurance กับ AccountInformation ค่าประกันจะมีบัญชี 1 บัญชี และ 1บัญชีจะมีจะมีค่าประกันได้หลาย ค่า เป็นความสัมพันธ์แบบ Aggregation
- Customerinformation กับ Contract ลูกหนี้ 1 คนมีสัญญาได้หลายสัญญา และสัญญา 1 สัญญามีลูกหนี้ได้คนเดียว เป็นความสัมพันธ์แบบ Aggregation

### 3.4.3 Attribute ของแต่ละ Class

Class แต่ละ Class จะมี Attribute ต่างๆ ดังตารางที่ 3.2

**ตารางที่ 3.2 แสดง Attribute ของคลาสต่างๆ**

| Class AccountInformation | |
|---|---|
| AccountID | รหัสบัญชี เช่น 0520 |
| AccountName | ชื่อบัญชี เช่นค่าเช่า, เช่าซื้อ (Install), ค่าบริการ (Service),ค่าประกัน (Insurance), ภาษี (Tax), ค่าที่ดินส่วนเกิน (Ex-land), สินเชื่อ (Loan), ค่าปรับ (ReSchedule), อื่นๆ (Other) |
| AccountDescription | คำอธิบายเพิ่มเติมแต่ละชื่อบัญชี |
| Class Contract | |
| ContractID | เลขที่สัญญา |
| ContractDate | วันที่ทำสัญญา |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่3.2 แสดง Attribute ของคลาสต่างๆ(ต่อ)**

| | |
|---|---|
| ContractStatus | สถานะสัญญาใช้ U แทนสัญญาที่ใช้งาน, T แทนสัญญาที่รอการปรับปรุงเปลี่ยนแปลง และ A แทนสัญญาที่มีการชำระครบแล้ว |
| PaymentType | ประเภทการชำระเงิน ใช้ C แทนชำระเงินสด และ B แทนชำระผ่านธนาคาร |
| PricePerTerm | เงินที่ต้องจ่ายต่องวด |
| DateFirstTerm | วันที่ทำการจ่ายเงินงวดแรก |
| Term | จำนวนงวดที่ต้องจ่าย |
| ContractType | ประเภทของสัญญาใช้ B แทนสัญญาเช่าซื้อ, R แทนสัญญาเช่า และ L แทนสัญญากู้เงิน |
| ContractIDNew | รหัสสัญญาที่เก็บไว้กรณีเปลี่ยนสัญญา |
| Class PaymentTransaction | |
| txID | รหัสสำหรับแต่ละ Transaction |
| PaymentDate | วันที่ของ Transaction ที่รับชำระ |
| PaymentTerm | งวดที่ชำระ |
| Class PaymentTransactionDetail | |
| PaymentPrice | จำนวนเงินที่ชำระต่อรายการบัญชี |
| Class Location | |
| LocationID | รหัสสำนักงานเคหะชุมชน |
| LocationName | ชื่อสำนักงานเคหะชุมชน |
| Limit | เขต |
| HouseNo | บ้านเลขที่ |
| Moo | หมู่ที่ |
| Soi | ซอย |
| Road | ถนน |
| Province | จังหวัด |
| PostalCode | รหัสไปรษณีย์ |
| Class Project | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.2 แสดง Attribute ของคลาสต่างๆ(ต่อ)**

| | |
|---|---|
| ProjectID | รหัสโครงการ |
| ProjectName | ชื่อโครงการ |
| Class Model | |
| ModelID | รหัสโมเดล |
| ModelName | ชื่อโมเดล |
| Class Quoation | |
| UnitAmount | จำนวนหน่วยอาศัย |
| Space | เนื้อที่ต่อหน่วย(ตร.ว.) |
| Cost | ราคาขายเงินสดต่อหน่วย(บาท) |
| StandardInterest | อัตราดอกเบี้ยการผ่อนชำระค่าเช่าซื้อต่อปี(%) |
| Class BuildingType | |
| BuildingTypeID | รหัสประเภทสิ่งปลูกสร้าง |
| BuildingTypeName | ชื่อประเภทสิ่งปลูกสร้าง |
| Class Unit | |
| HouseNo | บ้านเลขที่ |
| DebtorCode | รหัสลูกหนี้ |
| FloorNo | หมายเลขชั้น |
| BuildingNo | หมายเลขอาคาร |
| Class BuyContract | |
| BuyInterestRate | อัตราดอกเบี้ยของการเช่าซื้อ |
| BuyInterestTotal | ดอกเบี้ยทั้งหมดของการเช่าซื้อ |
| AmountBuy | จำนวนเงินทั้งหมดของการเช่าซื้อ |
| AmountDownPayment | จำนวนเงินดาวทั้งหมด |
| TermDownPayment | จำนวนงวดที่ต้องจ่ายเงินดาวน์ |
| DateFirstDownPayment | วันที่จ่ายเงินดาวน์งวดแรก |
| Class BuyDownPayment | |
| FromtermDownPayment | เงินดาวน์งวดที่เริ่มจ่าย |
| ToTermDownPayment | เงินดาวน์งวดสุดท้ายที่จ่าย |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.2 แสดง Attribute ของคลาสต่างๆ(ต่อ)**

| | |
|---|---|
| PricePerTermDownPayment | เงินดาวน์ที่ต้องจ่ายแต่ละงวด |
| Class Insurance | |
| InsurancePrice | จำนวนเงินค่าประกันต่างๆ |
| Class CustomerInformation | |
| CustomerID | รหัสลูกค้า |
| Title | คำนำหน้าชื่อ |
| FirstName | ชื่อ |
| LastName | นามสกุล |
| RegisAddress | ที่อยู่ตามบัตรประชาชน |
| CurrentAddress | ที่อยู่ปัจจุบัน |
| Telephone | เบอร์โทรศัพท์ |

### 3.4.4 Method ของแต่ละ Class

ในแต่ละ Class มีการดำเนินการดังนี้

- AccountInformation
  - getAccountName(accountid :CHAR) return CHAR;
    ค้นหาชื่อบัญชีจากรหัสบัญชี
- Paymenttransaction
  - getDateContract(paymentdate :DATE,tableOfContractID : TAB_TMP);
    ค้นหารายละเอียดสัญญาจากวันที่จ่ายเงิน
  - verifyDate(paymentdate :DATE) return BOOLEAN;
    ตรวจสอบวันที่จ่ายเงิน
  - gettermdown(contractid :CHAR) return number;
    ค้นหางวดที่จ่ายเงินดาวน์จากเลขที่สัญญา
  - calculatetermMax(contractid :CHAR) return number;
    ค้นหาจำนวนเทอมมากสุดที่จ่ายเงินของเลขที่สัญญา
- Paymenttransactiondetail
  - getPayListContract(paymentdate :DATE, contractid :CHAR,tableOfAccount :TAB_TMP);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหารายละเอียดรายการการจ่ายเงินจากวันที่จ่ายเงินและเลขที่สัญญา

- getPayListContractDown(paymentdate : DATE, contractid : CHAR ,tableOfAccountid :TAB_TMP);

  ค้นหารายละเอียดการจ่ายเงินชำระล่วงหน้าจากวันที่จ่ายเงินและเลขที่สัญญา

- Contract
  - getCustContract(contractID :CHAR,tableOfContractID : TAB_TMP);

    ค้นหารายละเอียดของสัญญาจากเลขที่สัญญา
  - getNewCont(tableOfContractID :TAB_TMP,tableOfContractID :TAB_TMP, projectID :CHAR,modelID :CHAR);

    ค้นหาชุดข้อมูลของสัญญาจากชุดข้อมูลเดิมโดยเปรียบเทียบกับรหัสโครงการและรหัสโมเดล
  - gethouseid(contractID :CHAR)return CHAR;

    ค้นหาเลขที่บ้านจากเลขที่สัญญา
  - getcontractid(tableOfcontractID :CHAR);

    ค้นหาเลขที่สัญญาทั้งหมด
  - getbuyterm(contractid :CHAR) return number;

    ค้นหาจำนวนเทอมการจ่ายเงินค่าเช่าซื้อจากเลขที่สัญญา
  - getpaytype(contractid :CHAR) return CHAR;

    ค้นหาประเภทของการจ่ายเงินจากเลขที่สัญญา
- Customerinformation
  - getCustlistcustomer(customerID :CHAR,tableOfCustomerID :TAB_TMP);

    หา List (ชุดข้อมูล) ของลูกค้าจากเลขที่ลูกค้า
  - verifycustomerid(customerID :CHAR) return BOOLEAN;

    ตรวจสอบรหัสลูกค้าว่าถูกต้องหรือไม่
  - getfname(contractID :CHAR) return CHAR;

    ค้นหาชื่อของลูกค้าจากรหัสสัญญา
  - getlname(contractID :CHAR) return CHAR;

    ค้นหานามสกุลลูกค้าจากรหัสสัญญา
- Insurance
  - getinsureP(contractid :CHAR) return CHAR;

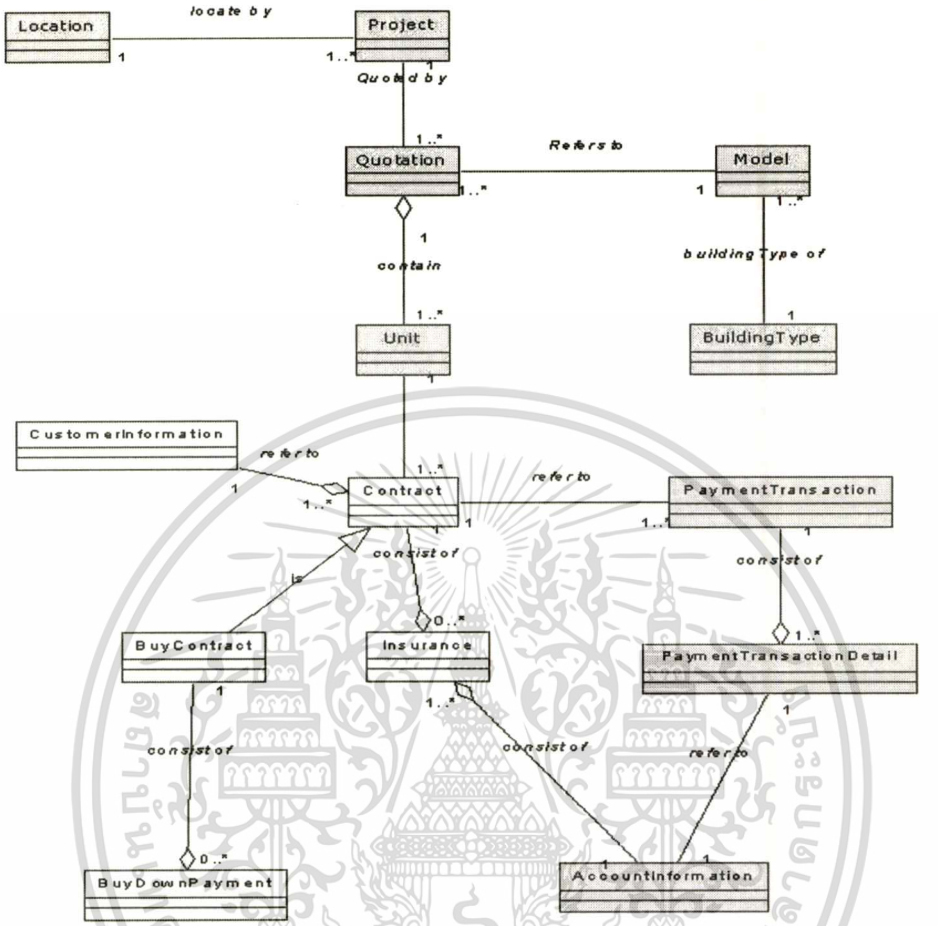
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหาค่าประกันจากเลขที่สัญญา

- Location
  - getlocationname(projectID :CHAR) return CHAR ;
    ค้นหาชื่อสำนักงานการเคหะจากรหัสโครงการ
  - getprovname(projectID :CHAR) return CHAR;
    ค้นหาชื่อจังหวัดจากรหัสโครงการ
- Project
  - getprojectname(projectID :CHAR) return CHAR;
    ค้นหาชื่อโครงการจากรหัสโครงการ
  - getprojectlist(tableOfProjectID);
    ค้นหารายละเอียดชุดข้อมูลของโครงการ
- Model
  - getmodelname(modelID :CHAR) return CHAR;
    ค้นหาชื่อโมเดลจากรหัสโมเดล
  - getlistModel(projectID :CHAR,tableOfModelID);
    ค้นหารายละเอียดของชุดขุอมูลโมเดลจากรหัสโครงการ
- Buildingtype
  - getBuildingname(modelid :CHAR) return CHAR;
    ค้นหาประเภทอาคารจากรหัสโมเดล

หลังจากกำหนดความสัมพันธ์ระหว่าง Class แล้วจะแสดงความสัมพันธ์ดังรูปที่3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.13 แสดง Class Diagram ของระบบเช่าซื้อ**

จากรูป 3.13 มีการใช้ Class ที่มาจากระบบต่างๆ ดังนี้

- ระบบรับชำระ จะประกอบด้วย Class ดังนี้
  - Class Paymenttransaction
  - Class PaymenttransactionDetail
  - Class AccountInformation
- ระบบตั้งหนี้ ประกอบด้วย Class ดังนี้
  - Class Location
  - Class Project
  - Class Model
  - Class Quotation
  - Class Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Class Buildingtype

- ระบบลูกหนี้สินเชื่อ เป็นระบบมีการทำสัญญา ประกอบด้วย Class ดังนี้
  - Class Contract
  - Class BuyContract
  - Class CustomerInformation
  - Class BuyDownPayment
  - Class Insurance

## 3.5 การออกแบบ Class Diagram ในขั้นตอนการออกแบบ

**หลังจากที่ทำการวิเคราะห์ระบบแล้วจึงจะทำการออกแบบระบบเช่าซื้อ ซึ่งขั้นตอนในการออกแบบระบบจะมีการปรับปรุงแก้ไข Class ไดอะแกรมที่ได้จากการวิเคราะห์ระบบ หลังจากนั้นจึงทำการออกแบบ Access Layer และการออกแบบ View Layer ต่อไป**

การออกแบบ Attribute จะมีการกำหนดรายละเอียดของ Attribute เพิ่มเติมเช่น มีการกำหนดประเภท ขนาดของ Attribute ดังตารางที่ 3.3

**ตารางที่ 3.3 แสดงรายละเอียด Attribute ของ Class ต่างๆ**

| Class AccountInformation | |
|---|---|
| AccountID | CHAR(4) |
| AccountName | VARCHAR2(30) |
| AccountDescription : | VARCHAR2(255) |
| Class Contract | |
| ContractID | CHAR(8) |
| ContractDate | DATE |
| ContractStatus | CHAR(1) |
| PaymentType | CHAR(1) |
| PricePerTerm | NUMBER(11,2) |
| DateFirstTerm | DATE |
| term | NUMBER(5) |
| ContractType | CHAR(1) |
| ContractIDNew | CHAR(8) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.3 แสดงรายละเอียด Attribute ของ Class ต่างๆ(ต่อ)**

| Class PaymentTransaction | |
|---|---|
| txID | NUMBER(15) |
| paymentDate | DATE |
| paymentTerm | NUMBER(5) |
| Class PaymentTransactionDetail | |
| paymentPrice | NUMBER(11,2) |
| Class Location | |
| LocationID | CHAR(4) |
| LocatinName | VARCHAR2(50) |
| Limit | VARCHAR2(2) |
| HouseNo | VARCHAR2(9) |
| Moo | VARCHAR2(2) |
| Soi | VARCHAR2(3) |
| Road | VARCHAR2(30) |
| Province | VARCHAR2(30) |
| PostalCode | VARCHAR2(8) |
| Class Project | |
| ProjectID | CHAR(4) |
| ProjectName | VARCHAR2(100) |
| Class Model | |
| ModelID | CHAR(2) |
| ModelName | VARCHAR2(20) |
| Class Quoation | |
| unitAmount = 0 | NUMBER(3) |
| space = 0.00 | NUMBER(5,2) |
| cost = 0.00 | NUMBER(11,2) |
| StandardInterest | NUMBER(4,2) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

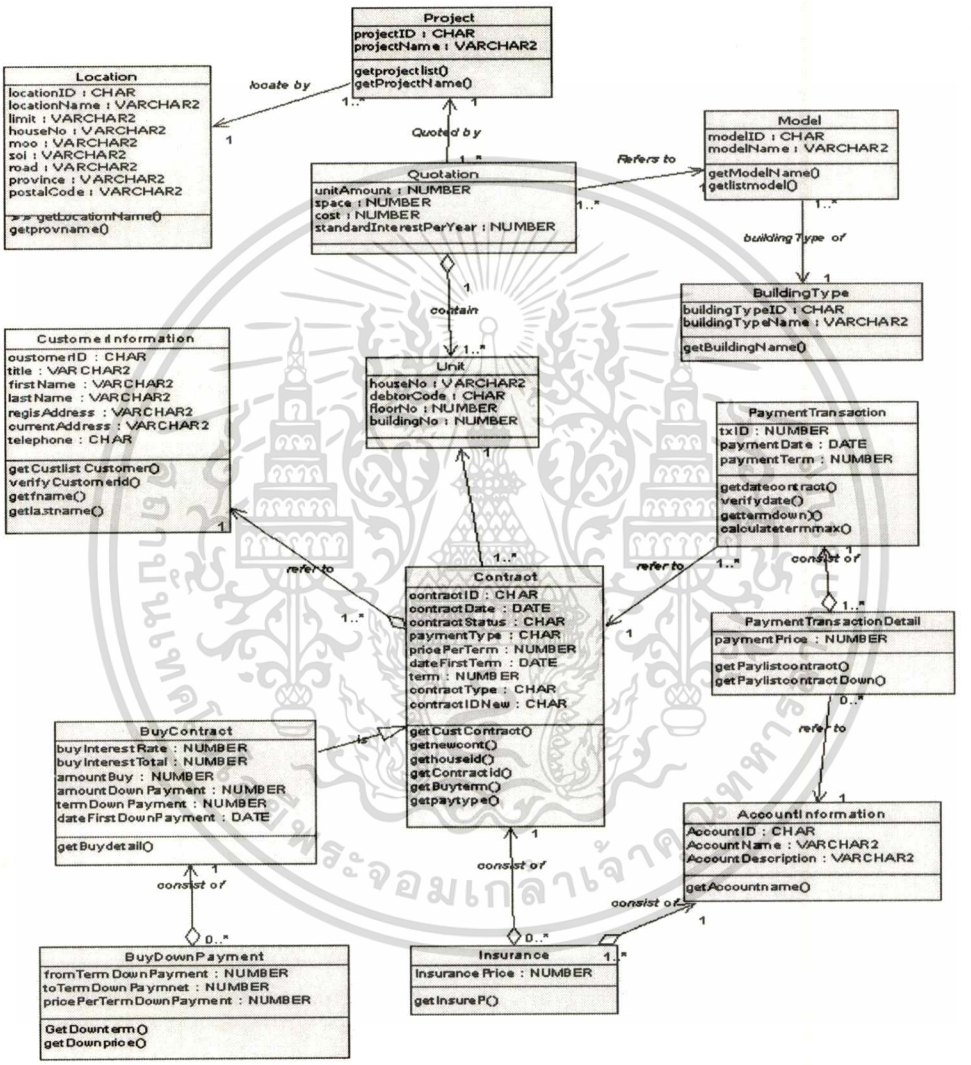
| Class BuildingType | |
|---|---|

**ตารางที่ 3.3 แสดงรายละเอียด Attribute ของ Class ต่างๆ(ต่อ)**

| | |
|---|---|
| BuildingTypeID | CHAR(2) |
| BuildingTypeName | VARCHAR2(50) |
| Class Unit | |
| HouseNo | VARCHAR2(9) |
| DebtorCode | CHAR(14) |
| BuildingNo | NUMBER(2) |
| Class BuyContract | |
| BuyInterestRate | NUMBER(4,2) |
| buyInterestTotal | NUMBER(7,2) |
| AmountBuy | NUMBER(9,2) |
| AmountDownPayment | NUMBER(9,2) |
| TermDownPayment | NUMBER(3,0) |
| dateFirstDownPayment | DATE |
| Class BuyDownPayment | |
| fromtermDownPayment | NUMBER(3,0) |
| toTermDownPayment | NUMBER(3,0) |
| pricePerTermDownPayment | NUMBER(7,2) |
| Class Insurance | |
| InsurancePrice | NUMBER(7,2) |
| Class CustomerInformation | |
| customerID | CHAR(13) |
| title | VARCHAR2(10) |
| firstName | VARCHAR(20) |
| lastName | VARCHAR(30) |
| regisAddress | VARCHAR(255) |
| currentAddress | VARCHAR(255) |
| telephone | CHAR(15) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

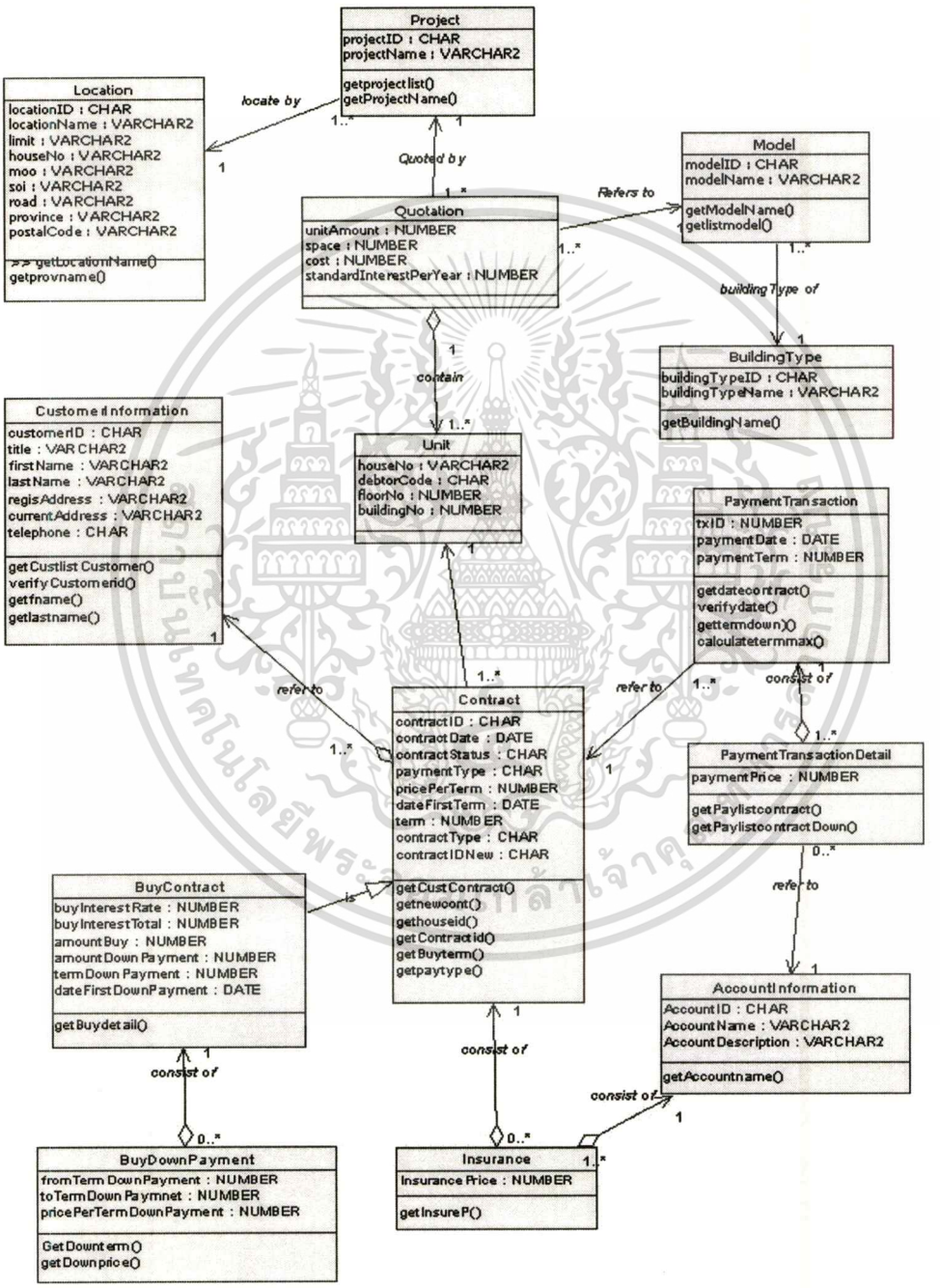
หลังจากทำการออกแบบ Attribute แล้วทำการออกแบบความสัมพันธ์ของ Class จะสามารถแสดงได้ดังรูปที่ 3.14




**รูปที่ 3.14 แสดงความสัมพันธ์ของ Class ในขั้นตอนการออกแบบ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นจะมีการกำหนดรายละเอียดเพิ่มเติมของ Class Diagram ซึ่งแสดงดังรูปที่ 3.15




**รูปที่ 3.15 แสดง Class Diagram จากการออกแบบเพิ่มเติม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 การออกแบบ Access Layer

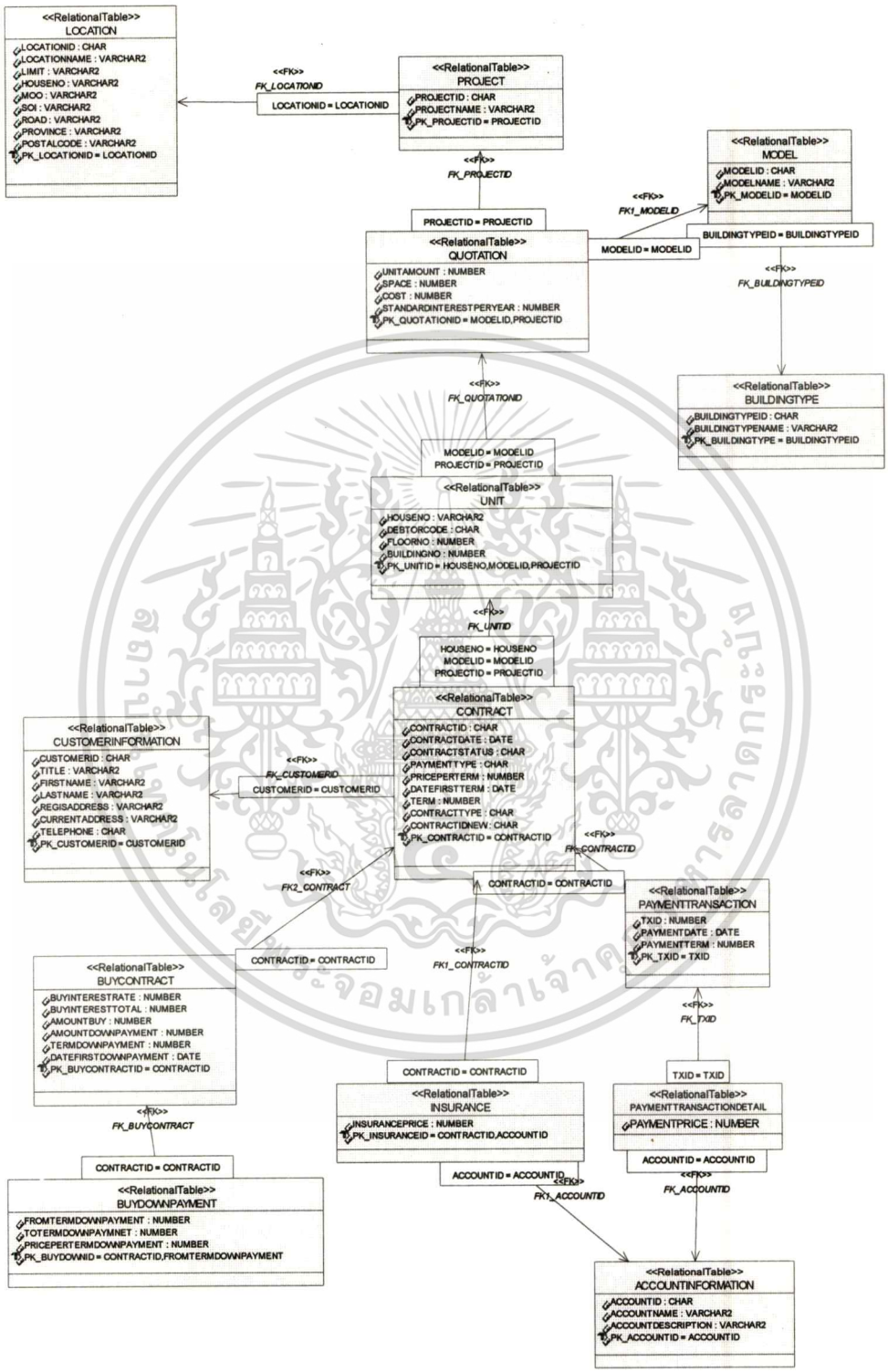
การออกแบบ Access Layer เป็นการออกแบบ Class ที่ทำหน้าที่ในการติดต่อกับข้อมูลต่างๆ ในฐานข้อมูลของระบบรวมถึงการสร้างฐานข้อมูลให้กับระบบ Access Layer สามารถที่จะแปลงการร้องขอข้อมูลจากทางธุรกิจไปเป็นโปรโตคอลที่เหมาะสมในการเข้าถึงข้อมูล ในระบบเช่าซื้อจะเป็นการออกแบบ Access Layer ให้เหมาะสมกับฐานข้อมูลแบบ Relational ของ Oracle ซึ่งฐานข้อมูลแบบ Relational จะมีการเข้าถึงข้อมูลได้โดยตรง แต่จากการออกแบบเชิงวัตถุการจะเข้าถึงข้อมูลต้องผ่าน interface ของ Object ดังนั้นในฐานข้อมูลแบบ Relational Class จะมีการทำเป็นโครงสร้างฐานข้อมูลเป็นลักษณะตาราง มี Attribute ของแต่ละตาราง และ Method ต่างๆ จะถูกแปลงไปเป็น Store Procedure ส่วน Object คือค่าในตารางแต่ละแถว

ในการออกแบบหลังจากที่มีการกำหนด Attribute ในแต่ละ Class และความสัมพันธ์ระหว่าง Class จะมีการกำหนด Primary Key และ Foreign Key เพื่อใช้ในการแปลงจาก Class ไปเป็นตารางของฐานข้อมูลแบบ Relational ซึ่งแสดงดังรูปที่ 3.16

จากรูปที่ 3.16 จะใช้สัญลักษณ์ Qualifier (ซึ่งเป็นสัญลักษณ์รูปสี่เหลี่ยมเล็กที่อยู่ติดกับตัว Class) ของ Class Diagram ใน UML แทนด้วย Foreign Key และแสดงชื่อของความสัมพันธ์ระหว่างแต่ละ Class พร้อมกับ Stereotype เป็น FK ด้วย สำหรับ Qualifier ที่มีการกำหนด Attribute มากกว่า 1 ตัวแสดงว่า Attribute เหล่านั้นเป็น Foreign Key เช่นกัน และยังมีการใช้สัญลักษณ์รูปที่คล้ายกับเครื่องมือในการขุด ( ) ซึ่งจะแสดงถึง Attribute สำหรับการสร้างและในที่นี้ให้กำหนดให้เป็น Index ที่ใช้ร่วมกับ Primary Key ใน Class

การทำงานสภาวะแวดล้อมใดๆ ก็ตามควรจะพยายามทำงานอยู่ในสภาวะแวดล้อมนั้นอย่างมีประสิทธิภาพ และเมื่อมีการทำงานอยู่ในสภาวะแวดล้อมของระบบฐานข้อมูลแบบ Relational จึงจำเป็นที่ต้องพิจารณา Class Diagram (หลังจากการปรับ) นั้นว่าเกิดปัญหาความซ้ำซ้อนของข้อมูลหรือไม่ หากเกิดปัญหาความซ้ำซ้อนของข้อมูลจะต้องมีการทำ Normalize ให้เหมาะสมด้วย สำหรับรูปที่ 3.16 นี้ไม่เกิดปัญหาความซ้ำซ้อนของข้อมูล

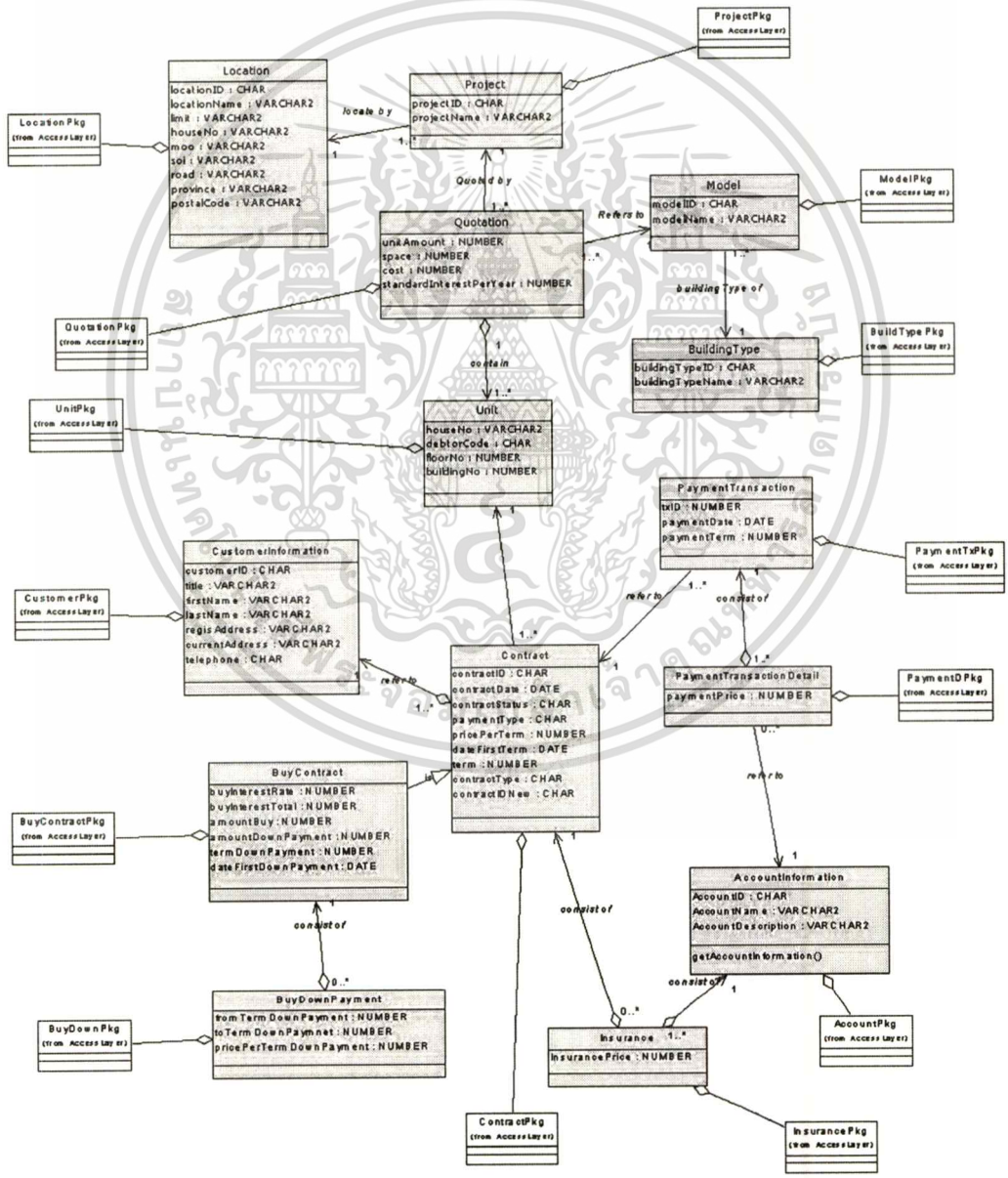
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.16 แสดง Class ที่เหมาะสมกับฐานข้อมูลแบบ Relational**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้าง Class ที่ทำหน้าที่เข้าถึงฐานข้อมูลจะมีการย้าย Method จาก Class เดิมที่มีการเข้าถึงฐานข้อมูลไปไว้ยัง Class ใหม่ที่อยู่ในส่วนของ Access Layer ส่วน Class ใดที่ทำงานได้โดยไม่ต้องใช้ข้อมูลในฐานข้อมูลก็จะคงเดิมไว้ Method ที่อยู่ใน Access Layer จะถูกสร้างเป็น Store Procedure ที่เป็นลักษณะ package ของแต่ละ Class ความสัมพันธ์ระหว่าง Access Layer กับ Class Diagram ที่มีการย้าย Method (Business Layer) จะเป็นลักษณะแบบ Aggregation ซึ่งแสดงดังรูปที่ 3.17




รูปที่ 3.17 แสดงความสัมพันธ์ของ Class ใน Access Layer กับ Business Layer

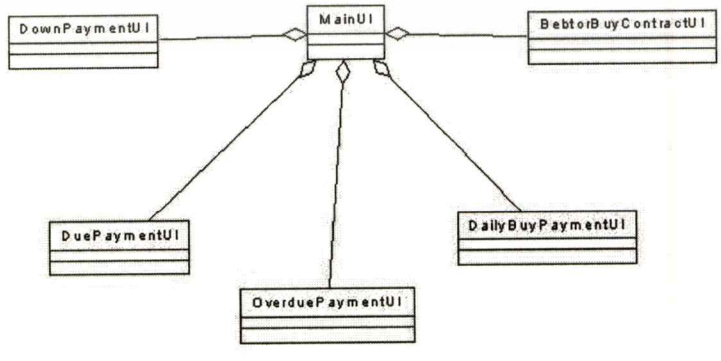
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.7 การออกแบบ View Layer

**การออกแบบ View Layer หรือ User Interface เป็นการสร้าง Class ใหม่ที่ทำหน้าที่เป็น interface ระหว่างผู้ใช้กับระบบโดยจะอาศัย Use Case Diagram ในขั้นตอนการวิเคราะห์มาช่วยในการออกแบบ Class ที่ใช้เป็น User Interface ได้ดังนี้**

- DebtorBuyContractUI
- DownPaymentUI
- OverduePaymentUI
- DuePaymentUI
- DailyBuyPaymentUI
- MainUI

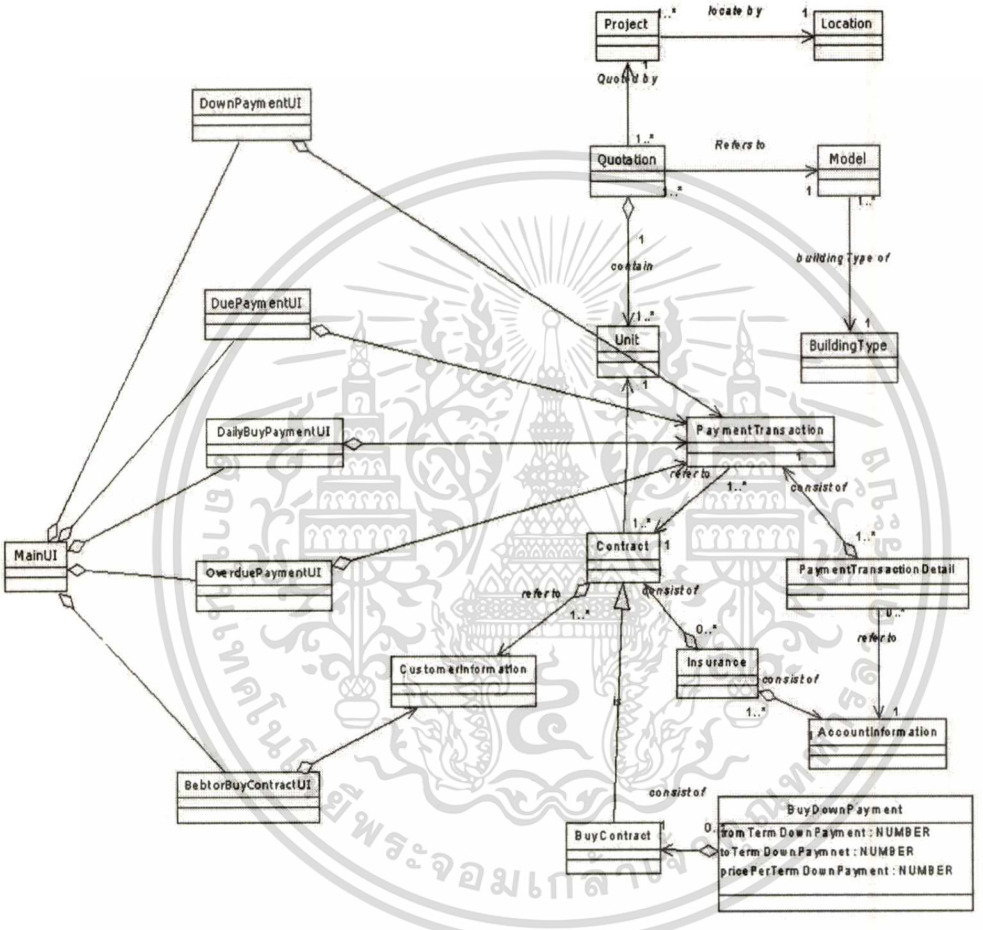
จาก Use Case Diagram ทำให้สามารถออกแบบ User Interface หลาย Class โดย Class DebtorBuyContract ทำหน้าที่เป็น User Interface ทำหน้าที่แสดงสัญญาของลูกหนี้เช่าซื้อ Class DownPaymentUI เป็น User Interface ที่ทำหน้าที่แสดงการชำระเงินล่วงหน้าของลูกหนี้เช่าซื้อ Class OverduePaymentUI เป็น User Interface ทำหน้าที่แสดงการค้างชำระของลูกหนี้เช่าซื้อ Class DuePaymentUI เป็น User Interface ที่ทำหน้าที่แสดงลูกหนี้ถึงกำหนดชำระ Class DailyBuyPayment เป็น User Interface ที่ทำหน้าที่แสดงการรับชำระเงินค่าเช่าซื้อ แต่การที่แสดง Class ต่างๆ นี้จำเป็นจะต้องมี Class ที่มาเชื่อมโยง Class เหล่านี้เข้าด้วยกันจึงได้มีการสร้าง Class ใหม่ขึ้นมาอีก 1 Class คือ Class MainUI ซึ่งจะช่วยให้การติดต่อระหว่าง Class ต่างๆ มีความสะดวกมากขึ้น หลังจากนั้นจึงทำการกำหนดความสัมพันธ์ของ Class ต่างๆ เข้าด้วยกัน ซึ่งแสดงดังรูปที่ 3.18




**รูปที่ 3.18 แสดงความสัมพันธ์ระหว่าง Class ที่เป็น User Interface**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ระหว่าง Class ที่ได้ในขั้นการออกแบบ View Layer นี้กับ Class ที่ได้ในออกแบบช่วงต้นจะเป็นความสัมพันธ์แบบ Aggregation ของ Class ทำงานร่วมกันซึ่งแสดงไว้ในรูปที่ 3.19

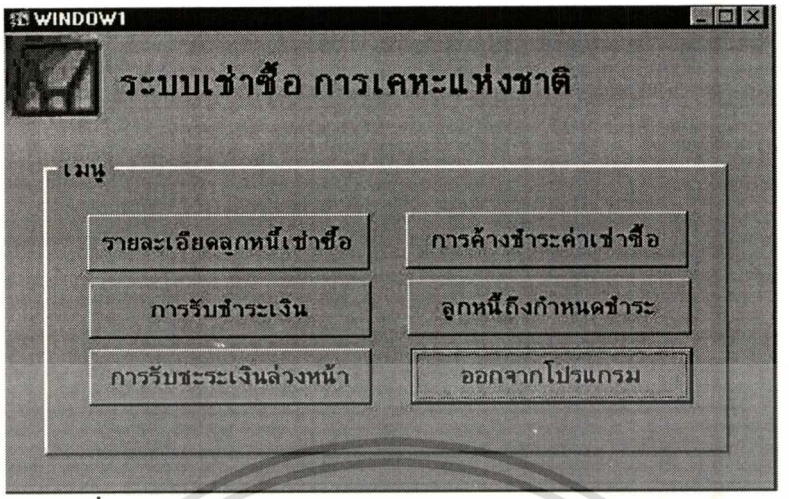



**รูปที่ 3.19 แสดงความสัมพันธ์ระหว่าง View Layer กับ Business Layer**

จากความสัมพันธ์และ Class ที่ได้จะใช้ในการออกแบบและสร้าง Prototype ขึ้นมาเป็นแบบ GUI โดยสิ่งที่จะแสดงใน GUI นี้อาจจะพิจารณาจาก Attribute ที่อยู่ใน Class ที่มีความสัมพันธ์กันใน Business Layer ก็ได้ การออกแบบหน้าจอการใช้งานของระบบดังนี้

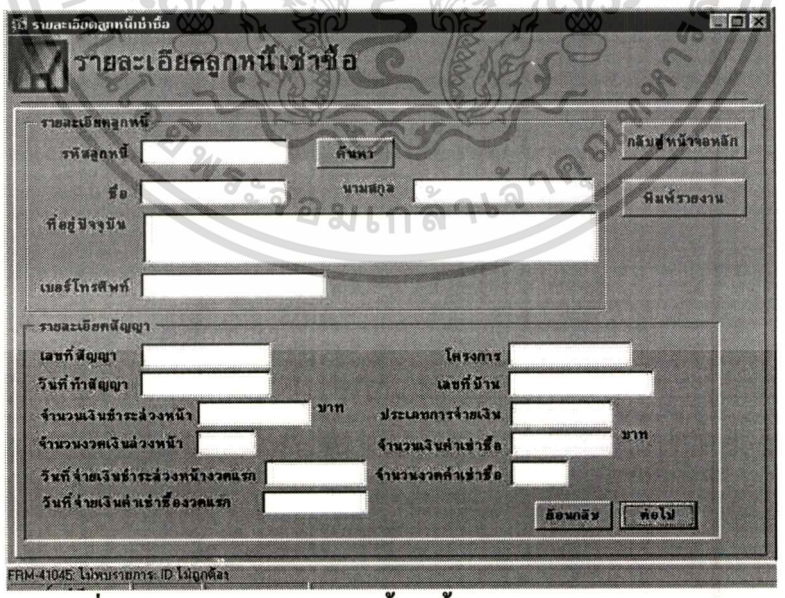
- Class MainUI จะสร้างเป็น GUI ได้ดังรูปที่ 3.20 ซึ่งเป็น GUI ของส่วนหน้าจอหลักของการใช้งานโดยจะเป็นตัวเชื่อมกับ GUI อื่นๆ ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.20 แสดง GUI ที่เป็นหน้าจอหลักของระบบ**

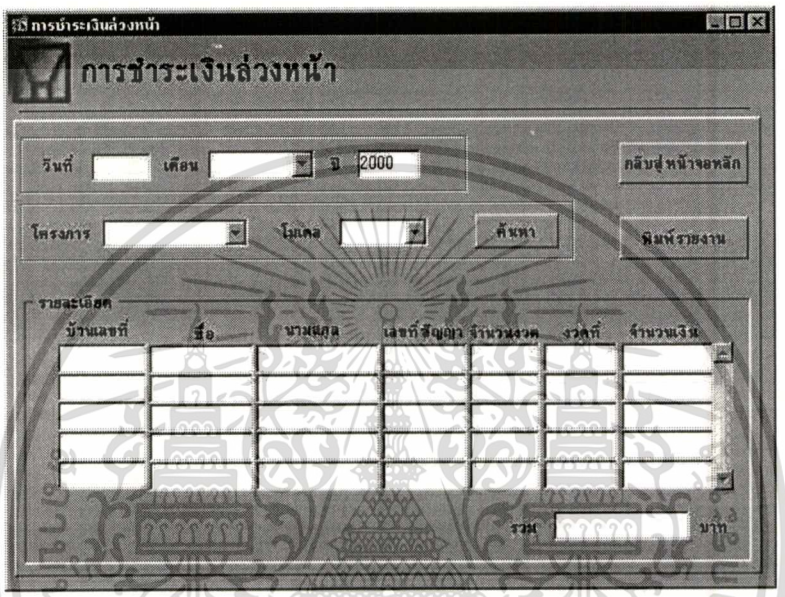
- Class DebtorBuyContractUI จะสร้างเป็น GUI ได้ดังรูปที่ 3.21 เป็น GUI ที่ทำหน้าที่แสดงรายละเอียดของลูกหนี้และรายละเอียดสัญญาเช่าซื้อของลูกหนี้เช่าซื้อ โดยจะใส่รหัสลูกหนี้เข้าไป และทำการค้นหา จะได้รายละเอียดที่อยู่ของลูกหนี้และสัญญาเช่าซื้อของลูกหนี้เช่าซื้อ ซึ่งลูกหนี้คนหนึ่งอาจมีสัญญาเช่าซื้อได้หลายสัญญา หลังจากนั้น หลังจากนั้นถ้าต้องการข้อมูลก็สั่งพิมพ์ได้




**รูปที่ 3.21 แสดง GUI ที่เป็นหน้าจอของรายลูกหนี้เช่าซื้อ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

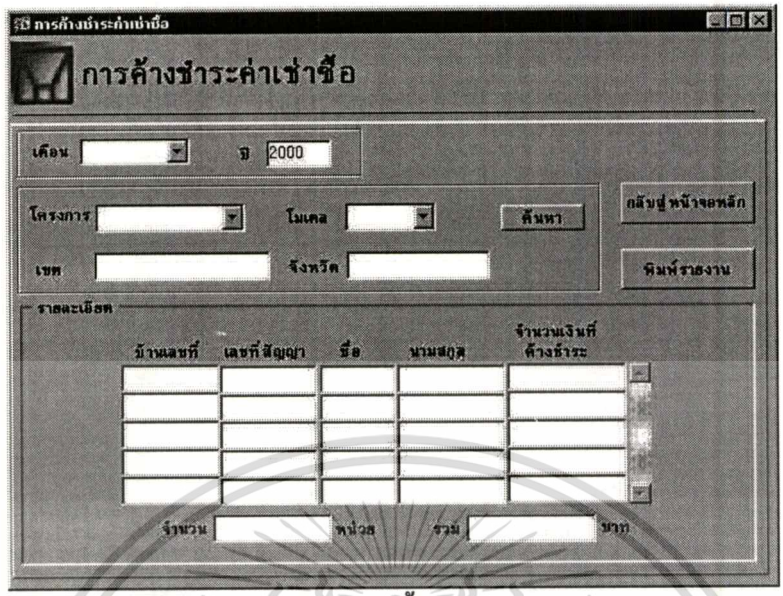
- Class DownPaymentUI จะสร้างเป็น GUI ได้ดังรูปที่ 3.22 เป็น GUI ที่ทำหน้าที่แสดงรายละเอียดการรับชำระเงินล่วงหน้า โดยสามารถเลือกวันที่ในการแสดงผล และเลือกโปรเจ็ค กับโมเดลเพื่อดูรายละเอียดการชำระเงินล่วงหน้าของลูกหนี้ และสามารถพิมพ์รายงานแสดงรายละเอียดการชำระเงินได้




**รูปที่ 3.22 แสดง GUI ของการชำระเงินล่วงหน้า**

- Class OverduePaymentUI จะสร้างเป็น GUI ได้ดังรูปที่ 3.23 โดยจะแสดงรายละเอียดของการค้างชำระค่าเช่าซื้อของลูกหนี้เพื่อนำไปใช้ในการวิเคราะห์หนี้ว่าควรจะทำอย่างไรต่อไป โดยจะมีการแสดงรายเดือน มีการแสดงโครงการและโมเดล และรายละเอียดการค้างชำระ และสามารถพิมพ์รายงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



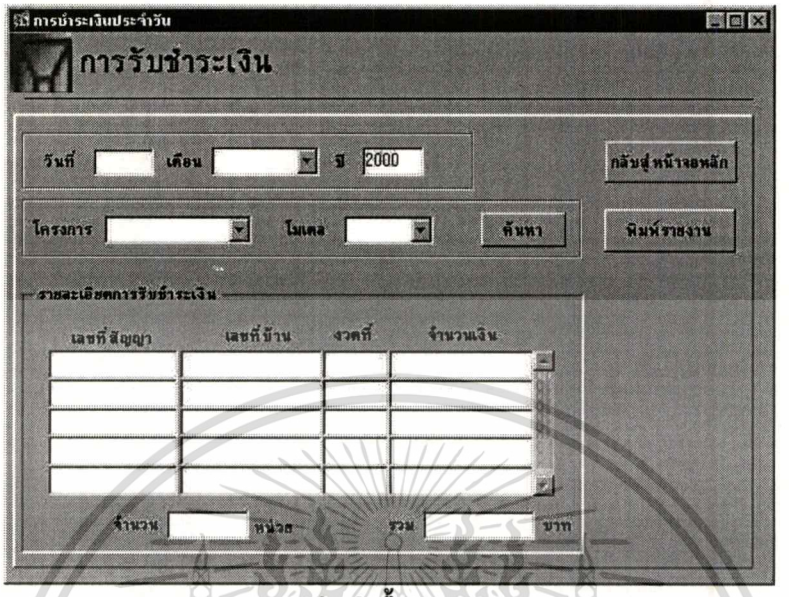

**รูปที่ 3.23 แสดง GUI ของการค้างชำระเงินค่าเช่าซื้อ**

- Class DuePaymentUI จะสร้างเป็น GUI ได้ดังรูปที่ 3.24 จะแสดงรายละเอียดที่จะชำระเงินค่าเช่าซื้อเมื่อถึงกำหนดชำระ ซึ่งสามารถพิมพ์รายงานได้

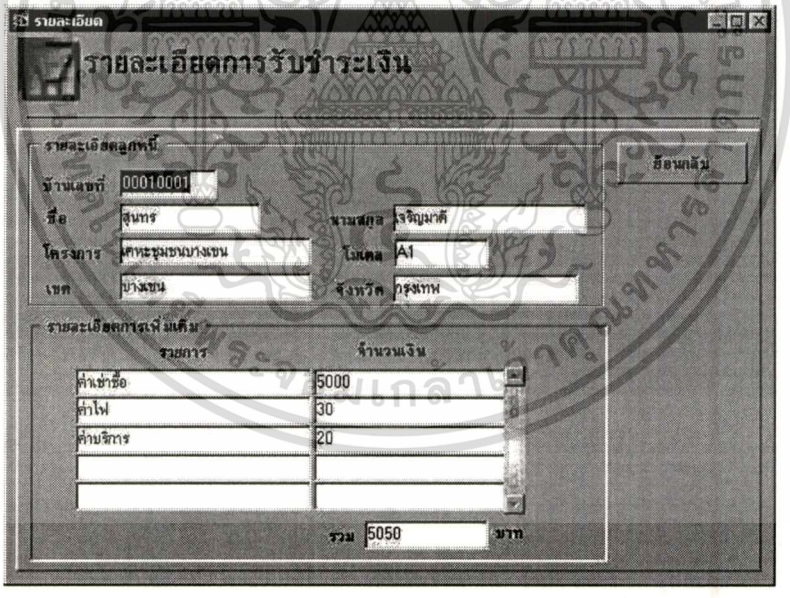
**รูปที่ 3.24แสดง GUI ของลูกหนี้ถึงกำหนดชำระ**

- Class DailyBuyPayment จะสร้างเป็น GUI ได้ดังรูปที่ 3.25 และ 3.26 โดยจะแสดงรายละเอียดการรับชำระเงินเช่าซื้อซึ่งสามารถแสดงตามวันได้ โดยแสดงโครงการและรายละเอียดการรับชำระเงินค่าเช่าซื้อ โดยสามารถพิมพ์รายงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.25 แสดง GUI ของการรับชำระเงินค่าเช่าซื้อประจำวัน**




**รูปที่ 3.26 แสดง GUI ของรายละเอียดเพิ่มเติมการรับชำระเงินค่าเช่าซื้อประจำวัน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# การสร้างระบบ

ในช่วงการสร้างระบบจะเป็นการเขียนโปรแกรมตามที่ได้มีการออกแบบไว้ ซึ่งมีการออกแบบเป็น 2 ส่วนคือ ส่วนของ User Interface และส่วนของ Application Login ซึ่งอยู่ที่ฝั่ง Client และส่วนที่เกี่ยวกับ Access Layer จะถูกพัฒนาไว้บนฝั่ง Server หลังจากที่ได้ GUI ที่เป็นหน้าจอต่างๆ แล้วจะทำการเขียนโปรแกรมสร้างระบบตามที่ออกแบบไว้ โดยจะมีการแปลง Class ที่ได้ออกแบบไว้ไปเป็นโครงสร้างฐานข้อมูลแบบ Relational ของระบบโดยใช้เครื่องมือ Rose2000 ในการออกแบบ Class และแปลงไปเป็นโครงสร้างฐานข้อมูล มีการใช้ Oracle server 8 ในการเป็น DBMS จัดการกับฐานข้อมูล ส่วนการพัฒนา GUI จะใช้ Developer 6

## 4.1 การพัฒนาบนฝั่ง Server

จากการออกแบบ Access Layer จะเป็นกลุ่ม Class ที่ทำงานอยู่ในส่วนของฝั่ง Server ซึ่งจะเป็นงานที่เริ่มจากการสร้างระบบฐานข้อมูลของระบบรับชำระจาก Schema มาในออกแบบ และทำการสร้าง Store Procedure ไว้บนฝั่ง Server โดยจะมีวิธีการดังต่อไปนี้

### 4.1.1 การติดตั้ง DBMS

การติดตั้ง DBMS เพื่อจัดการกับฐานข้อมูลจะใช้ Oracle server 8 โดยจะทำการติดตั้งที่ฝั่ง server เมื่อลง Oracle แล้วจะมีการสร้างฐานข้อมูล กำหนดขนาดฐานข้อมูล สร้าง user ของระบบที่มีสิทธิ์ในการจัดการกับฐานข้อมูล แล้วจึงทำการเชื่อมต่อกับผู้ใช้งานของระบบเพื่อให้สามารถจัดการและเข้าถึงฐานข้อมูลโดยใช้เครื่องมือ Net Easy Config เพื่อกำหนดการเชื่อม

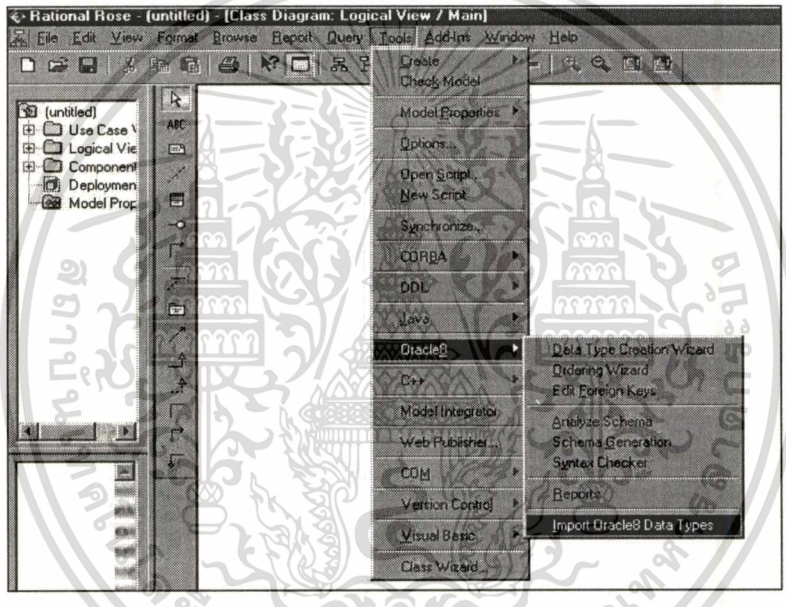
### 4.1.2 การสร้างตารางลงใน Database เพื่อเก็บข้อมูลของระบบ

การสร้างตารางของระบบจะอาศัย Rational Rose 2000 เป็นเครื่องมือช่วยในการสร้างฐานข้อมูล ซึ่งปกติจะต้องมาสร้างฐานข้อมูลขึ้นมาด้วยตัวเอง ซึ่งเครื่องมือนี้จะอาศัยแบบจำลองที่มีการออกแบบไว้เป็นข้อมูลในการสร้าง ตาราง บนฝั่ง Server หรือเรียกว่า Forward Engineering ซึ่งจะมีการกำหนด Model สำหรับการสร้างตารางแล้วจึงทำการสร้างตาราง โดยแปลงจาก Class ที่ได้ทำการออกแบบซึ่งเป็นรูปแบบของ Object ไปเป็น Class ที่สนับสนุนการสร้าง Schema แบบ Relational บน Database จะเป็นลักษณะ Class ที่ไม่มี Method และจะต้องมีการกำหนดลักษณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class ที่ต้องการสร้าง Schema ให้เหมาะสมในการสร้าง หลังจากนั้นจึงทำการแปลง Class โดยมีวิธีการแปลงดังนี้

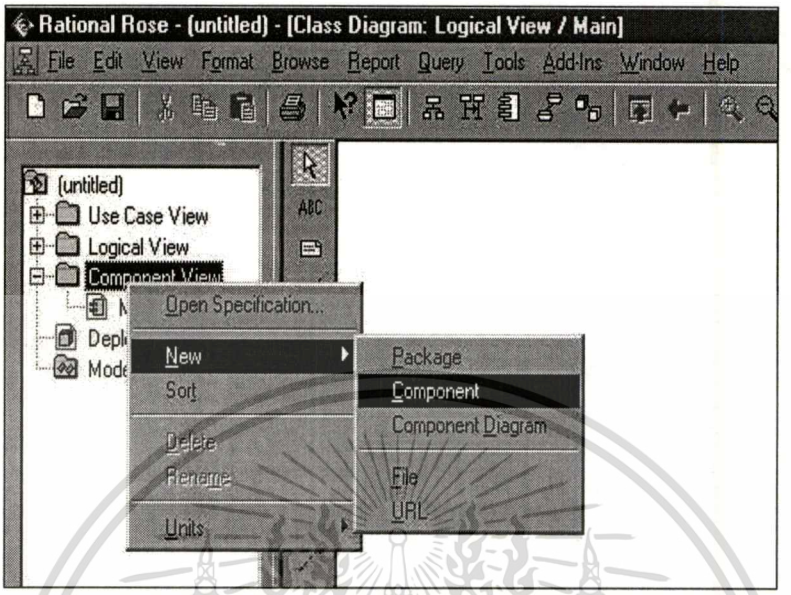
1. การแปลงจาก Class ที่ออกแบบไว้ไปเป็นตารางในฐานข้อมูลของ Oracle ถ้าไม่ได้เลือกรูปแบบของ Oracle ตั้งแต่เริ่มต้นการออกแบบจะต้องทำการ Import ชนิดข้อมูลของ Oracle ก่อน โดยเลือกเมนู tools แล้วเลือก Oracle8 แล้วเลือก Import Oracle8 Data types แสดงดังรูปที่ 4.1 แต่ถ้าเลือกรูปแบบของ Oracle ตั้งแต่ต้นก็ไม่ต้องทำการ Import ในขั้นตอนนี้



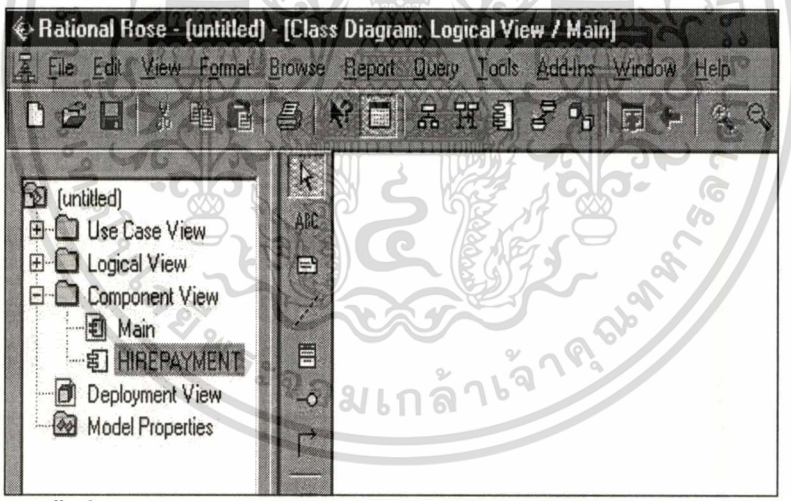

**รูปที่ 4.1 แสดงการ Import Oracle8 Data Type**

2. ทำการสร้าง Component และตั้งชื่อซึ่งจะใช้แทนชื่อเจ้าของตารางนั้น แสดงดังรูปที่ 4.2 และรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



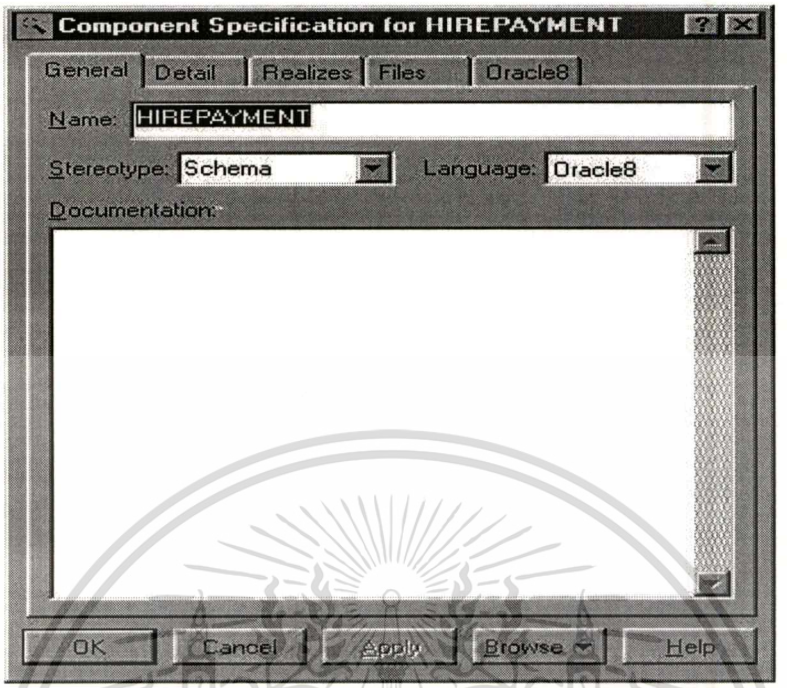

**รูปที่ 4.2 แสดงการสร้าง Component**



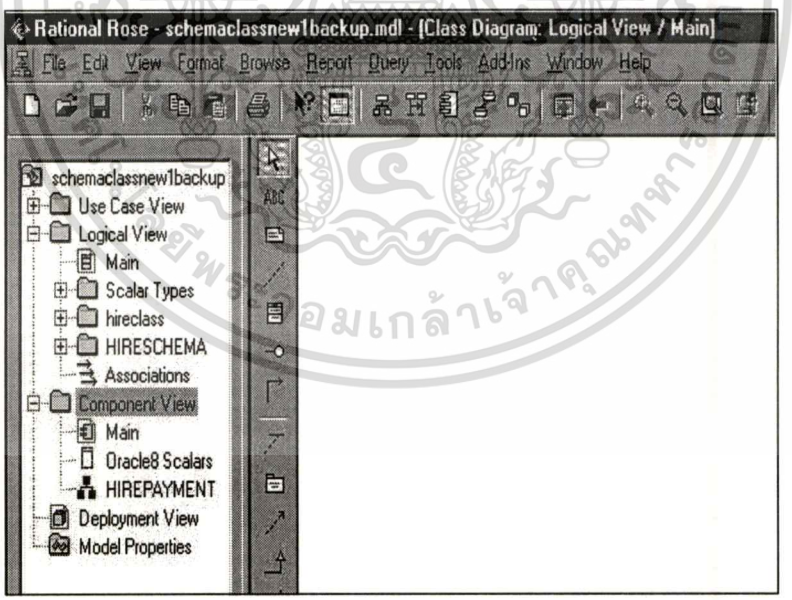

**รูปที่ 4.3 แสดงการตั้งชื่อ Component**

3. กำหนด Component ให้ Language เป็น Oracle8 โดยคลิกขวาที่ชื่อของ Component เลือก Open Specification แล้วเลือก language ให้เป็น Oracle8 แล้วกดปุ่ม Apply แล้วเลือก Sterotype เป็น Schema ซึ่งแสดงดังรูปที่ 4.4 แล้วกดปุ่ม OK รูปของ Component จะเปลี่ยนไปดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.4 แสดงการกำหนด Component**

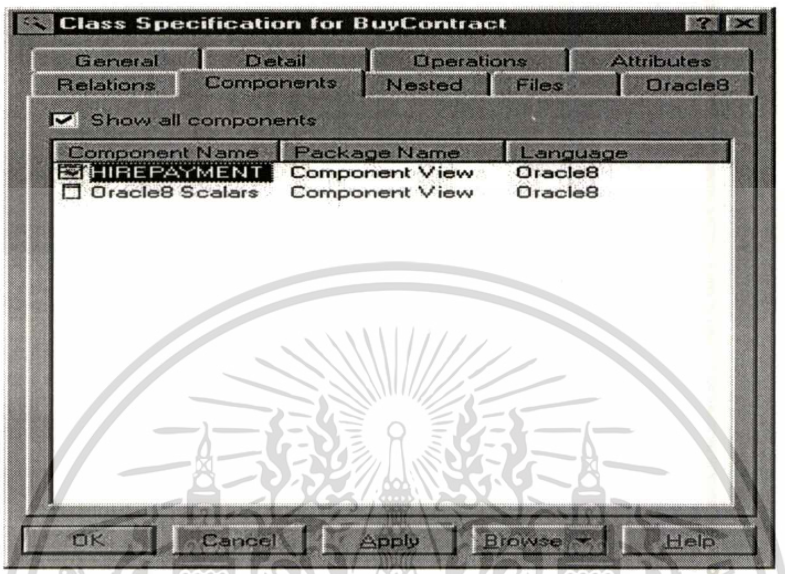



**รูปที่ 4.5 แสดงการเปลี่ยนรูป Component**

4. หลังจากนั้นมาที่ Class ที่ต้องการจะแปลงโดยเลือก Class ที่ต้องการแปลง แล้วคลิกขวาเลือก Open Specification มาที่ Tab Component แล้วเลือก Component ที่ต้องการ แล้วคลิกขวาที่ Component ที่ต้องการ แล้วเลือก Assign เพื่อให้ Class นั้น

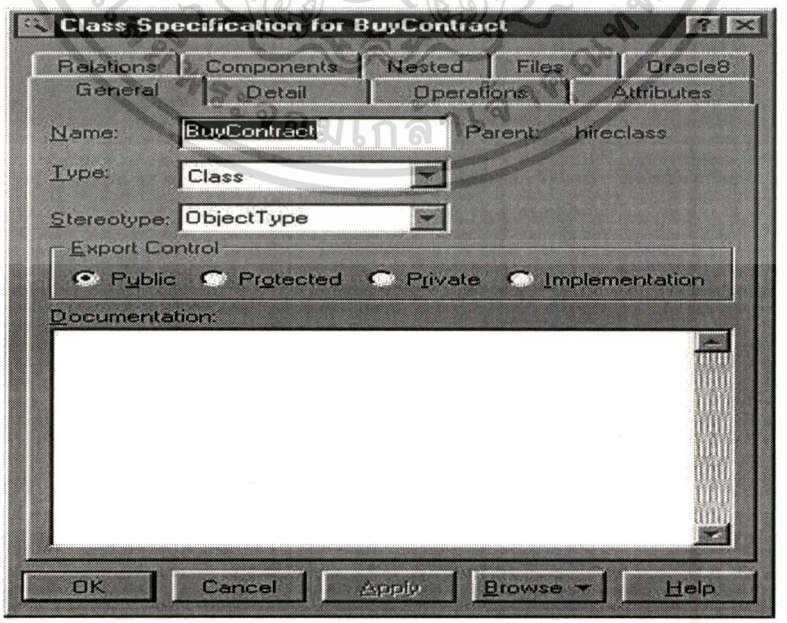
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นส่วนหนึ่งของ Component นั้นเพื่อจะแปลงไปเป็นตาราง แสดงดังรูปที่ 4.6 จะพบว่ามี Tab Oracle8 เกิดขึ้นมาใหม่




**รูปที่ 4.6 แสดงการ Assign Class**

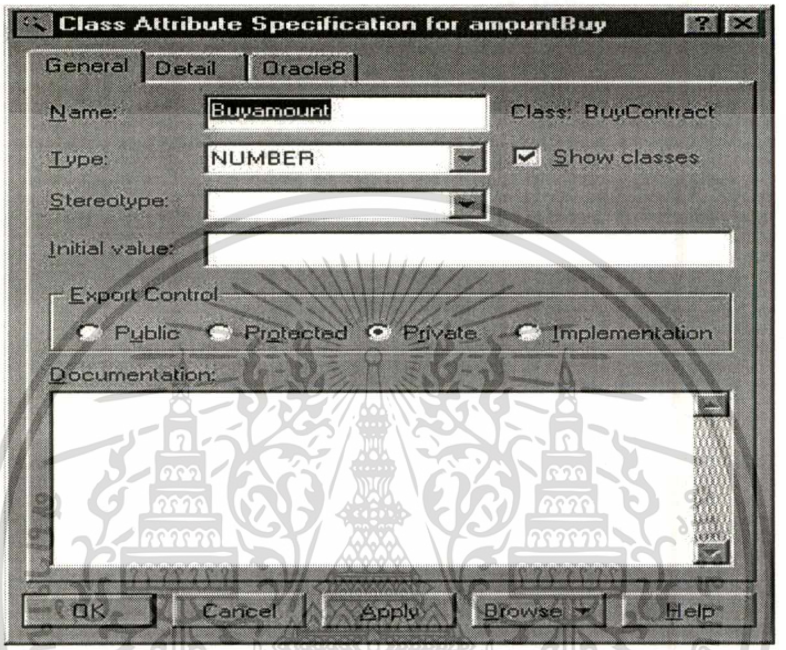
5. หลังจากนั้นมาที่ Tab General เลือก Stereotype เป็น ObjectType แสดงดังรูปที่ 4.7




**รูปที่ 4.7 แสดงการเลือก Stereotype เป็น Object Type**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

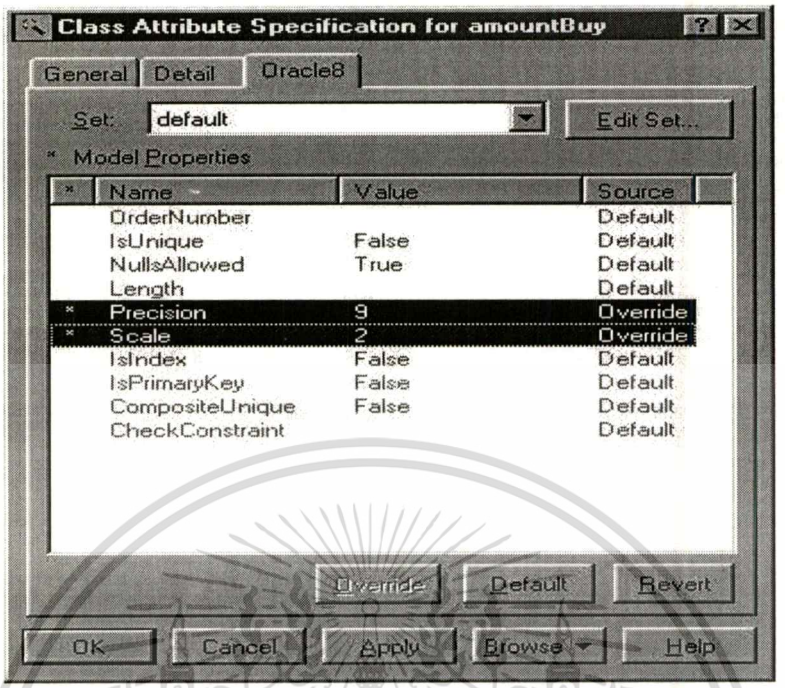
6. หลังจากนั้นมาที่ Tab Attribute ซึ่ง Attribute ของ Class นั้นจะต้องมีการกำหนดชนิดและขนาดของข้อมูล โดยดับเบิลคลิก (Double Click) ที่ Attribute ที่จะกำหนดขนาดและชนิดของข้อมูลจะแสดงรายละเอียดดังรูปที่ 4.8




**รูปที่ 4.8 แสดงรายละเอียดของ Attribute**

7. หลังจากนั้นมาที่ Tab Oracle8 เพื่อกำหนดขนาดของข้อมูลโดยถ้ามีชนิดของข้อมูลเป็นประเภท Char จะกำหนดขนาดที่ length แต่ถ้าเป็นประเภท Number จะกำหนดขนาดที่ precision ซึ่งเป็นขนาดของตัวเลขจำนวนเต็มก่อนจุดทศนิยม และ scale ซึ่งเป็นขนาดของจำนวนทศนิยมหลังจำนวนเต็ม ดังแสดงดังรูปที่ 4.9 แล้วกดปุ่ม OK

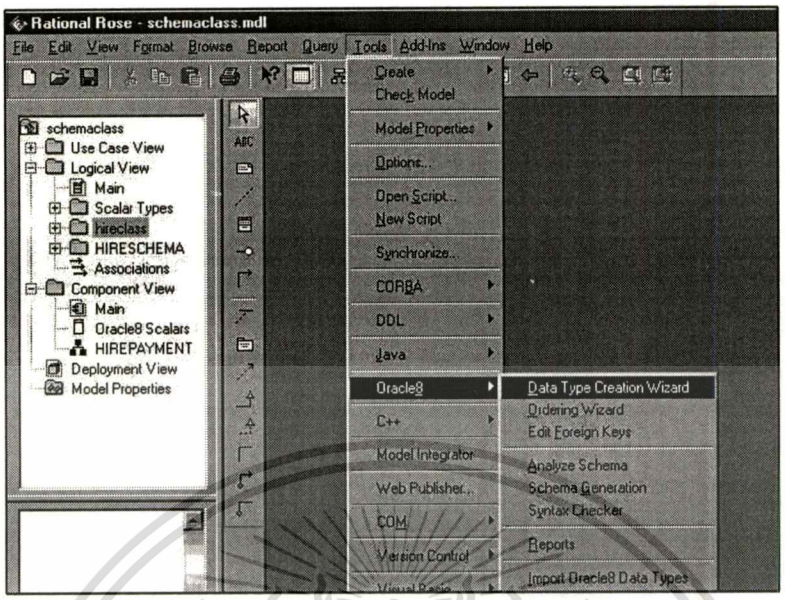
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



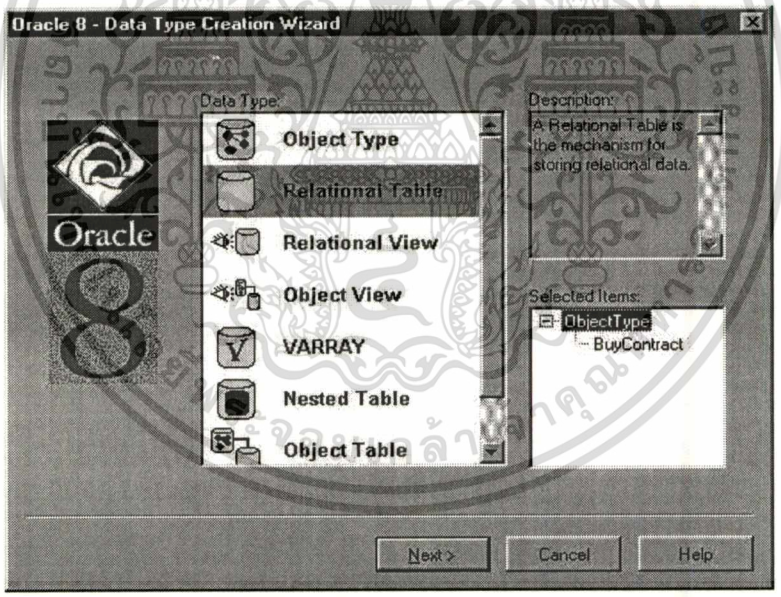

**รูปที่ 4.9 แสดงการกำหนดขนาดของข้อมูล**

8. ทำการกำหนด Class ทุก Class ที่ต้องการให้เป็นส่วนหนึ่งของ Component เพื่อแปลงไปเป็นตาราง โดยทำตั้งแต่ขั้นตอนที่ 4 ถึงขั้นตอนที่ 7 จนครบทุก Class
9. เมื่อทำการแปลง Class ต่างๆ ให้เป็น ObjecTtype เพื่อให้เป็นส่วนหนึ่งของ Component เรียบร้อยแล้วจะมีการกำหนด Class ขึ้นมาใหม่เพื่อใช้แปลงเป็นตารางในฐานข้อมูลโดยเลือก Class ที่ต้องการแล้วไปที่เมนู tools เลือก Oracle8 แล้วเลือก Data Type Creation Wizard แสดงดังรูปที่ 4.10 ซึ่งจะขึ้นหน้าจอ Data Type Creation Wizard แสดงดังรูปที่ 4.11 เลือก Relational Table แล้วกดปุ่ม Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



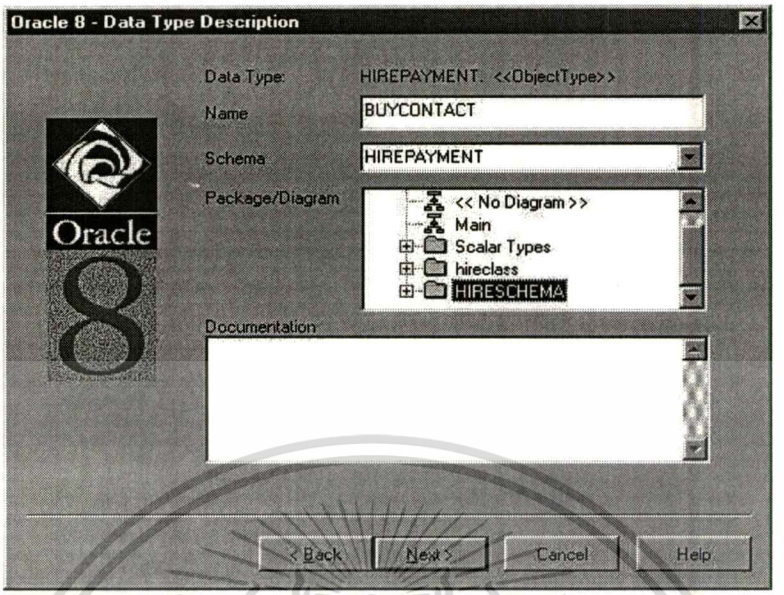

**รูปที่ 4.10 แสดงการเลือก Data Type Creation Wizard**




**รูปที่ 4.11 แสดงหน้าจอ Data Type Creation Wizard**

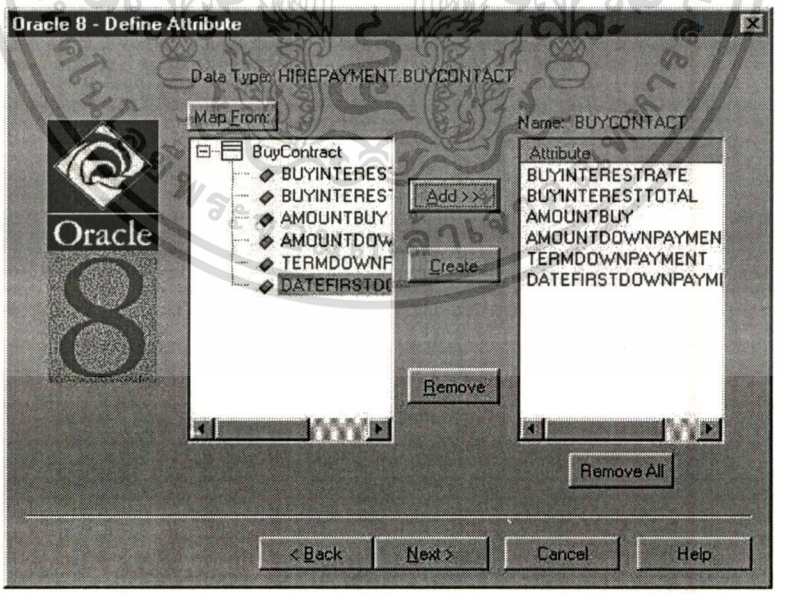
10. หลังจากกดปุ่ม Next จะแสดงหน้าจอดังรูปที่ 4.12 แล้วใส่ชื่อของ Class แล้วเลือก Schema ที่ต้องการและระบุตำแหน่งที่จะให้ Class นี้อยู่ซึ่งเป็น Package หรือ Diagram แล้วกดปุ่ม Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.12 แสดงหน้าจอการใส่รายละเอียด Class ใหม่**

11. หลังจากนั้นทำการเลือก Attribute ให้กับ Class ใหม่ โดยเลือก Attribute ที่ต้องการ แล้วกดปุ่ม Add แสดงดังรูปที่ 4.13 หลังจากนั้นกดปุ่ม Next

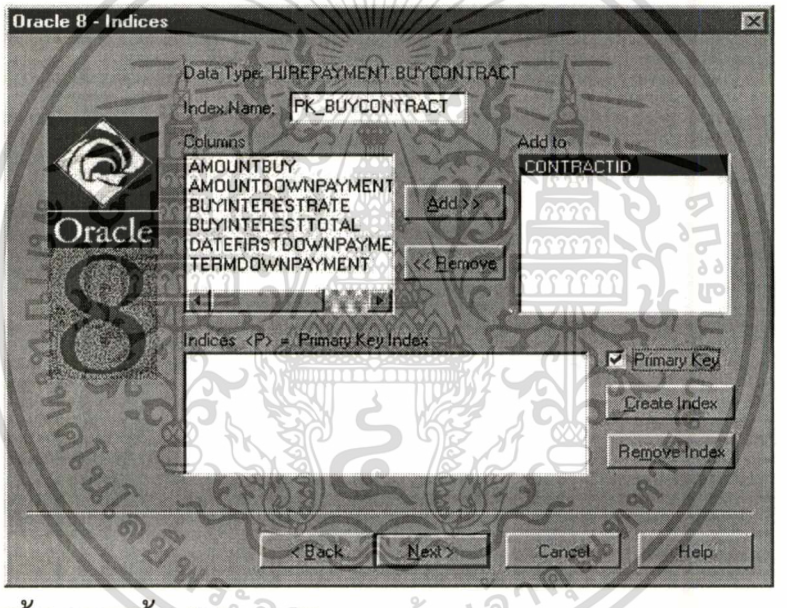



**รูปที่ 4.13 แสดงการเลือก Attribute**

12. หลังจากนั้นจะเข้าสู่หน้าจอที่จะสร้าง Primary Key แสดงดังรูปที่ 4.14 หลังจากนั้นเลือก Index ที่ต้องการ และตั้งชื่อ Index ถ้าต้องการให้เป็น Primary Key

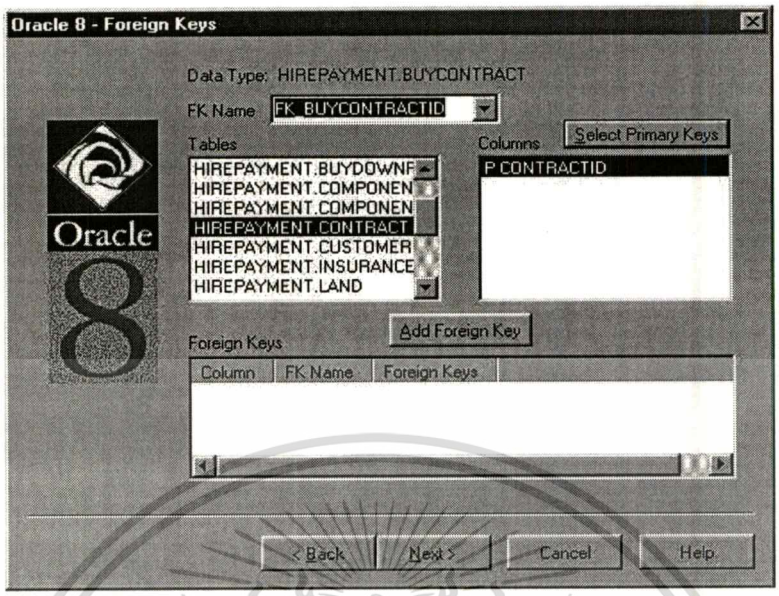
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็เช็คเลือก Primary Key แล้วจึง กดปุ่ม Create Index จะได้ Attribute ที่เป็น Primary Key แต่ถ้ามี Foreign Key ที่จะใช้เป็น Primary Key ก็ให้กดปุ่ม Next ไปสู่หน้าจอเลือก Foreign Key ก่อนโดยยังไม่ต้องทำตรงหน้าจอ Primary Key ซึ่งแสดงดังรูปที่ 4.15 ใส่ชื่อ Foreign Key และเลือก Attribute จากตารางอื่นที่ต้องการจะให้เป็น Foreign Key แล้วกดปุ่ม Add Foreign Key หลังจากนั้นจึงย้อนกลับไปหาหน้าจอสร้าง Primary Key เพื่อเลือก Attribute ที่จะเป็น Primary Key ของตารางนี้ แต่ถ้าไม่มี Foreign Key ก็ให้กดปุ่ม Next ของหน้าจอ Foreign Key ผ่านไปเลย



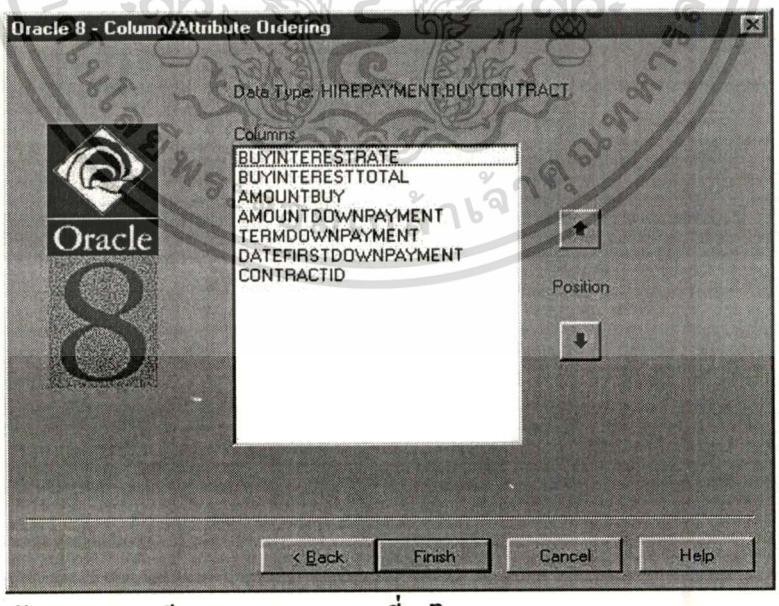

**รูปที่ 4.14 แสดงหน้าจอการสร้าง Primary Key**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.15 แสดงหน้าจอ Foreign Key**

13. หลังจากนั้นจะแสดงหน้าจอรายละเอียดของ Attribute ที่จะอยู่ในตารางดังรูปที่ 4.16 และสามารถจัดลำดับของ Attribute ได้ แล้วจึงกดปุ่ม Finish จบจบขั้นตอนการทำ Class ใหม่



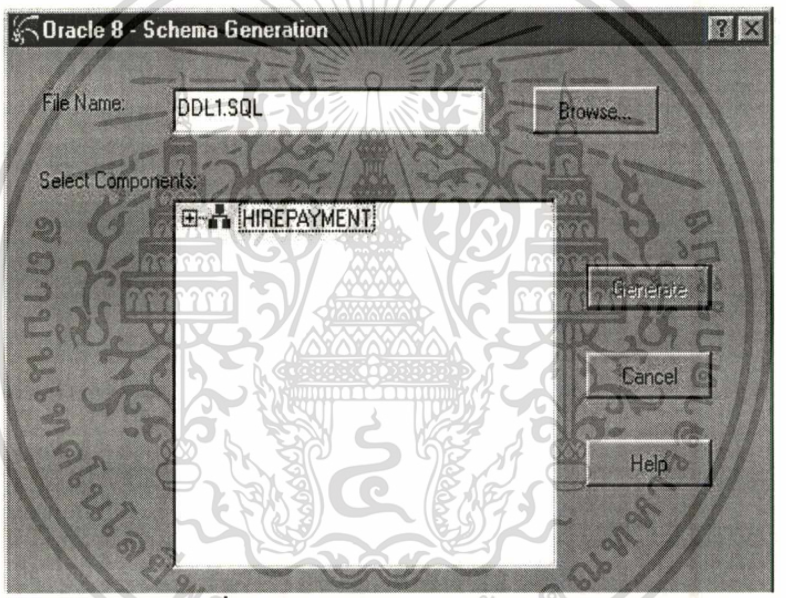

**รูปที่ 4.16แสดงหน้าจอ รายละเอียดของ Attribute ที่อยู่ในตาราง**

14. ทำซ้ำตั้งแต่ขั้นตอนที่ 9 ถึงขั้นตอนที่ 13 จนครบจำนวน Class ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่ใช้ฐานข้อมูลแบบ Relational จะต้องมีการจัดการเกี่ยวกับ Normalization เพื่อไม่ให้เกิดความซ้ำซ้อนของข้อมูล แต่จาก Class ที่จะทำการแปลงไปเป็นฐานข้อมูลพบว่าไม่เกิดความซ้ำซ้อนของข้อมูล หลังจากที่ทำการสร้าง Class ใหม่ที่เป็นส่วนหนึ่งของ Component แล้วเพื่อใช้ในการสร้างตารางแล้วจะมีการสร้างตารางในฐานข้อมลดังนี้

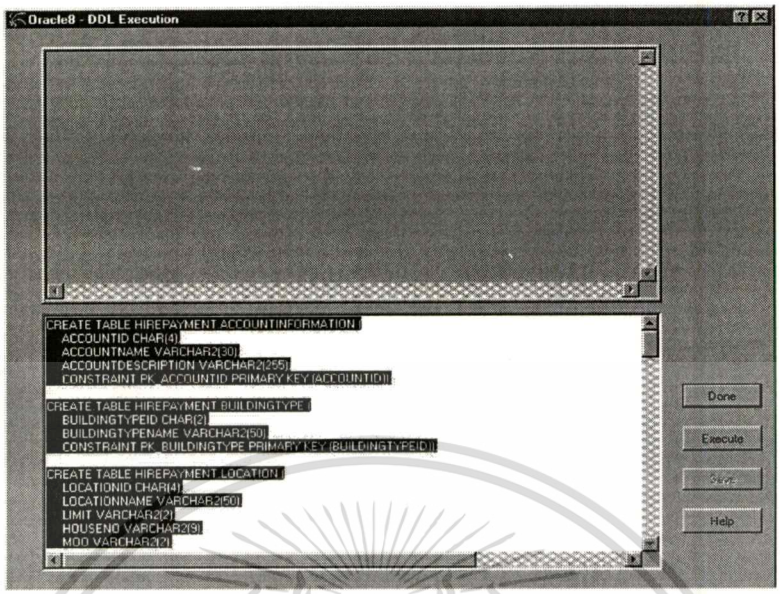
1. เลือกเมนู tools แล้วเลือก Oracleแล้วเลือก Schema Generation จะขึ้นหน้าจอของการแปลง Class ให้เป็นฐานข้อมูลแบบ Relational แสดงดังรูปที่ 4.17 ช่อง "File Name" จะเป็นช่องสำหรับใส่ที่อยู่และชื่อของ File ที่เก็บ Script DDL (Data Definition Language)




**รูปที่ 4.17 แสดงหน้าจอเลือก Class ที่จะแปลง**

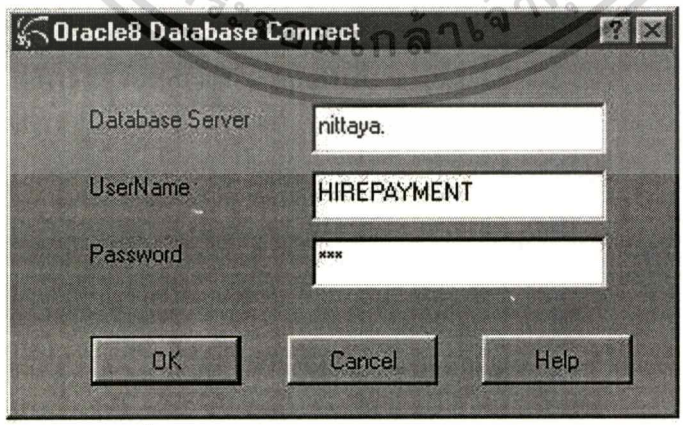
2. เลือก Class ใน Component ที่จะแปลง หลังจากนั้นกดปุ่ม Generate จะได้ผลลัพธ์ของการสร้าง Script DDL แสดงดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.18 แสดงผลลัพธ์ของการสร้าง Script DDL**

3. เมื่อได้ผลลัพธ์ของการสร้าง Script นี้ แล้วต้องการสร้าง Table ลงบน DBMS ในฝั่ง Server ก็เริ่มสร้างโดยกดปุ่ม Execute ซึ่งจะแสดงดังรูปที่ 4.26 หรือหากยังไม่สร้าง Table ในทันที ก็อาจจะนำ Script File ที่ได้นี้ไป Run บน DBMS ภายหลังได้ จากรูปที่ 4.19 จะมีการ Login เข้าสู่ DBMS เพื่อนำ Script File ไปทำงานบน DBMS (เพื่อสร้าง Table บน DBMS) ซึ่งจะได้ Table ที่ยังไม่มีข้อมูลอยู่ใน Database และพร้อมจะเก็บข้อมูลของระบบได้




**รูปที่ 4.19 แสดงการ Login เพื่อต่อเข้ากับ DBMS**

สุดท้ายผลลัพธ์ที่ได้ในขั้นนี้คือมี Table เปล่าๆ ที่ต้องการอยู่ใน Database และพร้อมจะเก็บข้อมูลของระบบได้ ซึ่งคำสั่งในการสร้างตารางของตารางในฐานข้อมูลแสดงได้ดังนี้ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CREATE TABLE HIREPAYMENT.ACCOUNTINFORMATION (
    ACCOUNTID CHAR(4),
    ACCOUNTNAME VARCHAR2(30),
    ACCOUNTDESCRIPTION VARCHAR2(255),
    CONSTRAINT PK_ACCOUNTID PRIMARY KEY (ACCOUNTID));

CREATE TABLE HIREPAYMENT.BUILDINGTYPE (
    BUILDINGTYPEID CHAR(2),
    BUILDINGTYPENAME VARCHAR2(50),
    CONSTRAINT PK_BUILDINGTYPE PRIMARY KEY (BUILDINGTYPEID));

CREATE TABLE HIREPAYMENT.LOCATION (
    LOCATIONID CHAR(4),
    LOCATIONNAME VARCHAR2(50),
    LIMIT VARCHAR2(2),
    HOUSENO VARCHAR2(9),
    MOO VARCHAR2(2),
    SOI VARCHAR2(3),
    ROAD VARCHAR2(30),
    PROVINCE VARCHAR2(30),
    POSTALCODE VARCHAR2(8),
    CONSTRAINT PK_LOCATIONID PRIMARY KEY (LOCATIONID));

CREATE TABLE HIREPAYMENT.CUSTOMERINFORMATION (
    CUSTOMERID CHAR(13),
    TITLE VARCHAR2(10),
    FIRSTNAME VARCHAR2(20),
    LASTNAME VARCHAR2(30),
    REGISADDRESS VARCHAR2(255),
    CURRENTADDRESS VARCHAR2(255),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TELEPHONE CHAR(15),
    CONSTRAINT PK_CUSTOMERID PRIMARY KEY (CUSTOMERID));

CREATE TABLE HIREPAYMENT.PROJECT (
    PROJECTID CHAR(4),
    PROJECTNAME VARCHAR2(100),
    LOCATIONID CHAR(4),
    CONSTRAINT PK_PROJECTID PRIMARY KEY (PROJECTID),
    CONSTRAINT FK_LOCATIONID FOREIGN KEY(LOCATIONID
    REFERENCES HIREPAYMENT.LOCATION(LOCATIONID));

CREATE TABLE HIREPAYMENT.MODEL (
    MODELID CHAR(2),
    MODELNAME VARCHAR2(20),
    BUILDINGTYPEID CHAR(2),
    CONSTRAINT PK_MODELID PRIMARY KEY (MODELID),
    CONSTRAINT FK_BUILDINGTYPEID FOREIGN KEY(BUILDINGTYPEID)
    REFERENCES HIREPAYMENT.BUILDINGTYPE(BUILDINGTYPEID));

CREATE TABLE HIREPAYMENT.QUOTATION (
    PROJECTID CHAR(4),
    MODELID CHAR(2),
    UNITAMOUNT NUMBER(3,0),
    SPACE NUMBER(5,2),
    COST NUMBER(11,2),
    STANDARDINTERESTPERYEAR NUMBER(4,2),
    CONSTRAINT PK_QUOTATIONID PRIMARY KEY
    (MODELID,PROJECTID),
    CONSTRAINT FK1_MODELID FOREIGN KEY(MODELID) REFERENCES
    HIREPAYMENT.MODEL(MODELID),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CONSTRAINT FK_PROJECTID FOREIGN KEY(PROJECTID)
    REFERENCES  HIREPAYMENT.PROJECT(PROJECTID));

CREATE TABLE HIREPAYMENT.UNIT (
    PROJECTID CHAR(4),
    MODELID CHAR(2),
    HOUSENO VARCHAR2(9),
    DEBTORCODE CHAR(14),
    FLOORNO NUMBER(2,0),
    BUILDINGNO NUMBER(2,0),
    CONSTRAINT PK_UNITID PRIMARY KEY
    (HOUSENO,MODELID,PROJECTID),
    CONSTRAINT FK_QUOTATIONID FOREIGN
    KEY(MODELID,PROJECTID) REFERENCES HIREPAYMENT.QUOTATION
    (MODELID,PROJECTID));

CREATE TABLE HIREPAYMENT.CONTRACT (
    CONTRACTID CHAR(8),
    CONTRACTDATE DATE,
    CONTRACTSTATUS CHAR(1),
    PAYMENTTYPE CHAR(1),
    PRICEPERTERM NUMBER(11,2),
    DATEFIRSTTERM DATE,
    TERM NUMBER(5,0),
    CONTRACTTYPE CHAR(1),
    CONTRACTIDNEW CHAR(8),
    CUSTOMERID CHAR(13),
    HOUSENO VARCHAR2(9),
    MODELID CHAR(2),
    PROJECTID CHAR(4),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CONSTRAINT PK_CONTRACTID PRIMARY KEY (CONTRACTID),
    CONSTRAINT FK_CUSTOMERID FOREIGN KEY(CUSTOMERID)
    REFERENCES HIREPAYMENT.CUSTOMERINFORMATION
    (CUSTOMERID),
    CONSTRAINT FK_UNITID FOREIGN
    KEY(HOUSENO,MODELID,PROJECTID) REFERENCES
    HIREPAYMENT.UNIT(HOUSENO,MODELID,PROJECTID));

CREATE TABLE HIREPAYMENT.PAYMENTTRANSACTION (
    TXID NUMBER(15,0),
    PAYMENTDATE DATE,
    PAYMENTTERM NUMBER(5,0),
    CONTRACTID CHAR(8),
    CONSTRAINT PK_TXID PRIMARY KEY (TXID),
    CONSTRAINT FK_CONTRACTID FOREIGN KEY(CONTRACTID)
    REFERENCES HIREPAYMENT.CONTRACT(CONTRACTID));

CREATE TABLE HIREPAYMENT.BUYCONTRACT (
    CONTRACTID CHAR(8),
    BUYINTERESTRATE NUMBER(4,2),
    BUYINTERESTTOTAL NUMBER(7,2),
    AMOUNTBUY NUMBER(9,2),
    AMOUNTDOWNPAYMENT NUMBER(9,2),
    TERMDOWNPAYMENT NUMBER(3,0),
    DATEFIRSTDOWNPAYMENT DATE,
    CONSTRAINT PK_BUYCONTRACTID PRIMARY KEY (CONTRACTID),
    CONSTRAINT FK2_CONTRACT FOREIGN KEY(CONTRACTID)
    REFERENCES HIREPAYMENT.CONTRACT(CONTRACTID));

CREATE TABLE HIREPAYMENT.INSURANCE (
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

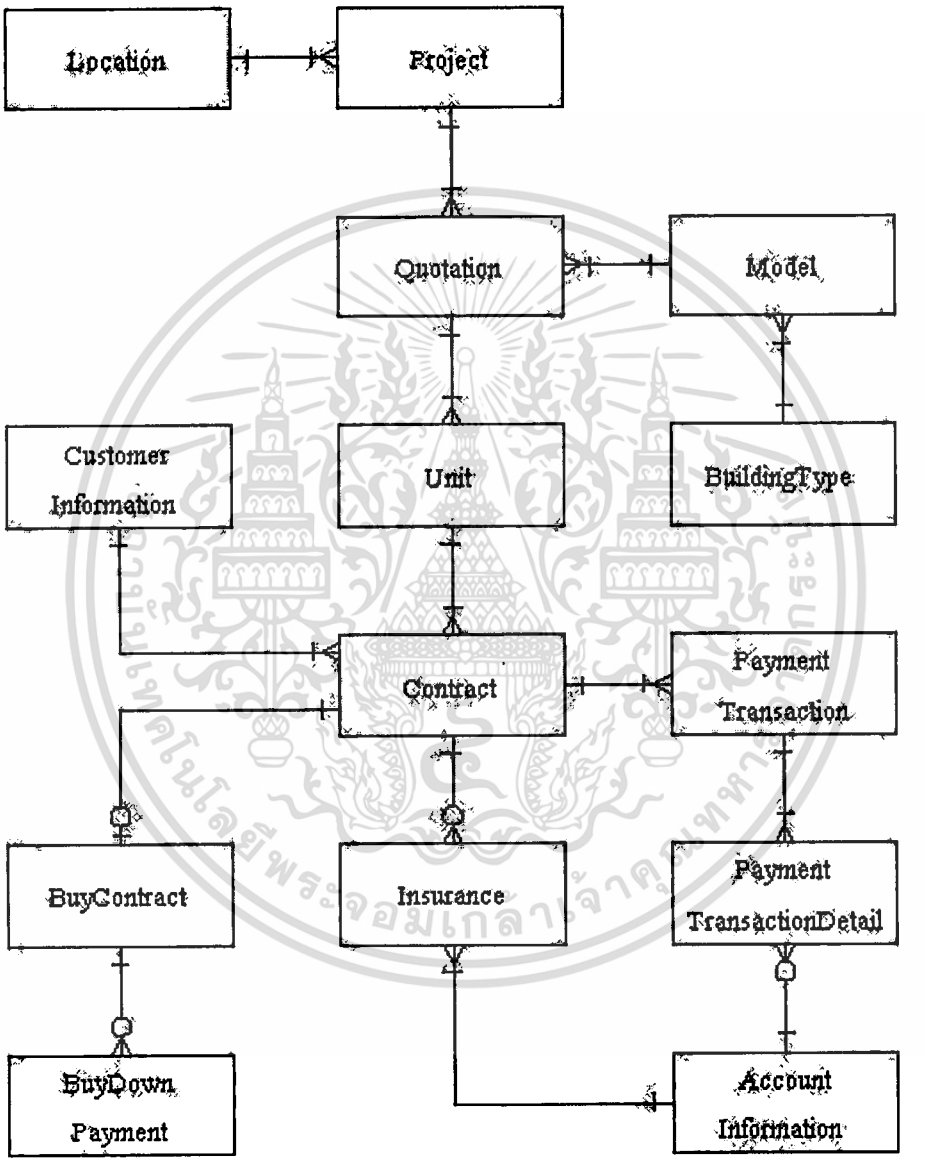
```
    CONTRACTID CHAR(8),
    ACCOUNTID CHAR(4),
    INSURANCEPRICE NUMBER(7,2),
    CONSTRAINT PK_INSURANCEID PRIMARY KEY
    (CONTRACTID,ACCOUNTID),
    CONSTRAINT FK1_CONTRACTID FOREIGN KEY(CONTRACTID)
    REFERENCES HIREPAYMENT.CONTRACT(CONTRACTID),
    CONSTRAINT FK1_ACCOUNTID FOREIGN KEY(ACCOUNTID)
    REFERENCES HIREPAYMENT.ACCOUNTINFORMATION(ACCOUNTID));

CREATE TABLE HIREPAYMENT.BUYDOWNPAYMENT (
    CONTRACTID CHAR(8),
    FROMTERMDOWNPAYMENT NUMBER(3,0),
    TOTERMDOWNPAYMNET NUMBER(3,0),
    PRICEPERTERMDOWNPAYMENT NUMBER(7,2),
    CONSTRAINT PK_BUYDOWNID PRIMARY KEY
    (CONTRACTID,FROMTERMDOWNPAYMENT),
    CONSTRAINT FK_BUYCONTRACT FOREIGN KEY(CONTRACTID)
    REFERENCES HIREPAYMENT.BUYCONTRACT(CONTRACTID));

CREATE TABLE HIREPAYMENT.PAYMENTTRANSACTIONDETAIL (
    TXID NUMBER(15,0),
    ACCOUNTID CHAR(4),
    PAYMENTPRICE NUMBER(11,2),
    PAYMENTTRANSACTIONDETAIL_ID NUMBER(5,0),
    PRIMARY KEY (PAYMENTTRANSACTIONDETAIL_ID),
    CONSTRAINT FK_TXID FOREIGN KEY(TXID) REFERENCES
    HIREPAYMENT.PAYMENTTRANSACTION(TXID),
    CONSTRAINT FK_ACCOUNTID FOREIGN KEY(ACCOUNTID)
    REFERENCES HIREPAYMENT.ACCOUNTINFORMATION(ACCOUNTID));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคำสั่งในการสร้างตารางจะพบว่า เมื่อนำไปสร้างเป็นตารางแล้วจะได้ 3NF ทำให้ไม่เกิดความซ้ำซ้อนของข้อมูลมีการกำหนด Primary Key และ Foreign Key ของแต่ละตาราง ซึ่งแสดงเป็น ER (Entity Relationship) ดังรูปที่ 4.20




**รูปที่ 4.20 แสดง Entity Relationship**

จากรูปที่ 4.20 จะพบว่าแต่ละตารางที่ได้ ไม่เกิดการซ้ำซ้อน เป็น 3NF จึงไม่จำเป็นต้องทำ Normalization อีก โดยแต่ละตารางจะมี Attribute ที่เป็น Primary Key และ Foreign Key ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.1 ฐานข้อมูล Location**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| LocationID (PK) | CHAR(4) NOT NULL | รหัสสำนักงานเคหะชุมชน |
| LocatinName | VARCHAR2(50) | ชื่อสำนักงานเคหะชุมชน |
| Limit | VARCHAR2(2) | เขต |
| HouseNo | VARCHAR2(9) | บ้านเลขที่ |
| Moo | VARCHAR2(2) | หมู่ที่ |
| Soi | VARCHAR2(3) | ซอย |
| Road | VARCHAR2(30) | ถนน |
| Province | VARCHAR2(30) | จังหวัด |
| PostalCode | VARCHAR2(8) | รหัสไปรษณีย์ |

**ตารางที่ 4.2 ฐานข้อมูล Project**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ProjectID (PK) | CHAR(4) NOT NULL | รหัสโครงการ |
| ProjectName | VARCHAR2(100) | ชื่อโครงการ |
| LocationID (FK1) | CHAR(4) NOT NULL | รหัสสำนักงานเคหะชุมชน |

**ตารางที่ 4.3 ฐานข้อมูล Building Type**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| BuildingTypeID (PK) | CHAR(2) NOT NULL | รหัสประเภทสิ่งปลูกสร้าง |
| BuildingTypeName | VARCHAR2(50) | ชื่อประเภทสิ่งปลูกสร้าง |

**ตารางที่ 4.4 ฐานข้อมูล BuyDownPayment**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ContractID(PK,FK) | CHAR(8) NOT NULL | เลขที่สัญญา |
| FromtermDownPayment | NUMBER(3,0) | เงินดาวน์งวดที่เริ่มจ่าย |
| ToTermDownPayment | NUMBER(3,0) | เงินดาวน์งวดสุดท้ายที่จ่าย |
| PricePerTermDownPayment | NUMBER(7,2) | เงินดาวน์ต้องจ่ายแต่ละงวด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.5 ฐานข้อมูล Quotation**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ProjectID (PK,FK) | CHAR(4) NOT NULL | รหัสโครงการ |
| ModelID (PK,FK) | CHAR(2) NOT NULL | รหัสโมเดล |
| UnitAmount | NUMBER(3) | จำนวนหน่วยอาศัย |
| Space | NUMBER(5,2) | เนื้อที่ต่อหน่วย (ตร.ว) |
| Cost | NUMBER(11,2) | ราคาขายเงินสดต่อหน่วย (บาท) |
| StandardInterest | NUMBER(4,2) | อัตราดอกเบี้ยการผ่อนชำระค่าเช่าซื้อต่อปี (%) |

**ตารางที่ 4. 6 ฐานข้อมูล Unit**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ProjectID (PK,FK) | CHAR(4) NOT NULL | รหัสโครงการ |
| ModelID (PK,FK) | CHAR(2) NOT NULL | รหัสโมเดล |
| HouseNo (PK) | VARCHAR2(9) NOT NULL | บ้านเลขที่ |
| DebtorCode | CHAR(14) | รหัสลูกหนี้ |
| BuildingNo | NUMBER(2) | หมายเลขอาคาร |
| FloorNO | CHAR(2,0) | ชั้น |

**ตารางที่ 4.7 ฐานข้อมูล Paymenttransaction**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| txID (PK) | NUMBER(15) NOT NULL | รหัสสำหรับแต่ละ Transaction |
| PaymentDate | DATE | วันที่ของ Transaction ที่รับชำระ |
| PaymentTerm | NUMBER(5) | งวดที่ชำระ |
| ContractID(FK) | CHAR(8) NOT NULL | เลขที่สัญญา |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 ฐานข้อมูล Contract**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ContractID (PK) | CHAR(8) NOT NULL | เลขที่สัญญา |
| ContractDate | DATE | วันที่ทำสัญญา |
| ContractStatus | CHAR(1) | สถานะสัญญาใช้ U แทนสัญญาที่ใช้งาน, T แทนสัญญาที่รอการปรับปรุงเปลี่ยนแปลง และ A แทนสัญญาที่มีการชำระครบแล้ว |
| PaymentType | CHAR(1) | ประเภทการชำระเงิน ใช้ C แทนชำระเงินสด และ B แทนชำระผ่านธนาคาร |
| PricePerTerm | NUMBER(11,2) | เงินที่ต้องจ่ายต่องวด |
| DateFirstTerm | DATE | วันที่ทำการจ่ายเงินงวดแรก |
| term | NUMBER(5) | จำนวนงวดที่ต้องจ่าย |
| ContractType | CHAR(1) | ประเภทของสัญญาใช้ B แทนสัญญาเช่าซื้อ, R แทนสัญญาเช่า และ L แทนสัญญากู้เงิน |
| ContractIDNew | CHAR(8) | รหัสสัญญาที่เก็บไว้กรณีเปลี่ยนสัญญา |
| CustomerID(FK) | CHAR(8) NOT NULL | รหัสลูกค้า |
| HouseNo(FK) | VARCHAR2(9) NOT NULL | บ้านเลขที่ |
| ProjectID(FK) | CHAR(4) NOT NULL | รหัสโครงการ |
| ModelID(FK) | CHAR(2) NOT NULL | รหัสโมเดล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 ฐานข้อมูล Model**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ModelID (PK3) | CHAR(2) NOT NULL | รหัสโมเดล |
| ModelName | VARCHAR2(20) | ชื่อโมเดล |
| BuildingTypeID (FK4) | CHAR(2) NOT NULL | รหัสประเภทสิ่งปลูกสร้าง |

**ตารางที่ 4.10 ฐานข้อมูล PaymentTransactionDetail**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| AccountID(PK,FK) | CHAR(4) NOT NULL | รหัสบัญชี |
| txID(PK,FK) | NUMBER(15) NOT NULL | รหัสแต่ละ Transaction |
| PaymentPrice | NUMBER(11,2) | จำนวนเงินที่ชำระต่อรายการบัญชี |

**ตารางที่ 4.11 ฐานข้อมูล CustomerInformation**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| CustomerID(PK) | CHAR(13) NOT NULL | รหัสลูกค้า |
| Title | VARCHAR2(10) | คำนำหน้าชื่อ |
| FirstName | VARCHAR(20) | ชื่อ |
| LastName | VARCHAR(30) | นามสกุล |
| RegisAddress | VARCHAR(255) | ที่อยู่ตามบัตรประชาชน |
| CurrentAddress | VARCHAR(255) | ที่อยู่ปัจจุบัน |
| Telephone | CHAR(15) | เบอร์โทรศัพท์ |

**ตารางที่ 4.12 ฐานข้อมูล Insurance**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ContractID(PK,FK) | CHAR(8) NOT NULL | เลขที่สัญญา |
| AccountID(PK)(FK) | CHAR(4) NOT NULL | รหัสบัญชี |
| InsurancePrice | NUMBER(7,2) | จำนวนเงินค่าประกันต่างๆ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.13 ฐานข้อมูล BuyContract**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| ContractID(PK,FK) | CHAR(8) NOT NULL | เลขที่สัญญา |
| BuyInterestRate | NUMBER(4,2) | อัตราดอกเบี้ยของการเช่าซื้อ |
| BuyInterestTotal | NUMBER(7,2) | ดอกเบี้ยทั้งหมดของการเช่าซื้อ |
| AmountBuy | NUMBER(9,2) | จำนวนเงินทั้งหมดของการเช่าซื้อ |
| AmountDownPayment | NUMBER(9,2) | จำนวนเงินดาวทั้งหมด |
| TermDownPayment | NUMBER(3,0) | จำนวนงวดที่ต้องจ่ายเงินดาวน์ |
| DateFirstDownPayment | DATE | วันที่จ่ายเงินดาวน์งวดแรก |

**ตารางที่ 4.14 ฐานข้อมูล AccountInformation**

| ชื่อ Attribute | ชนิดข้อมูล | คำอธิบาย |
|---|---|---|
| AccountID (PK) | CHAR(4) NOT NULL | รหัสบัญชี เช่น 0520 |
| AccountName | VARCHAR2(30) | ชื่อบัญชี เช่นค่าเช่า, เช่าซื้อ (Install),ค่าบริการ (Service),ค่าประกัน (Insurance),ภาษี(Tax), ค่าที่ดินส่วนเกิน(Ex-land), สินเชื่อ(Loan),ค่าปรับ (ReSchedule),อื่นๆ (Other) |
| AccountDescription : | VARCHAR2(255) | คำอธิบายเพิ่มเติมแต่ละชื่อบัญชี |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.1.3 การสร้าง Store Procedure ใน Database**

เป็นการสร้าง Store Procedure จาก Class ในกลุ่ม Access Layer โดยจะสร้างเป็น Package 1 Package ต่อ Class 1 Class ใน Access Layer หรือสามารถจะนำ Component Diagram ในขั้นตอนการออกแบบมาเป็น Diagram สำหรับการอ้างอิงได้เช่นกัน

การสร้าง Store Procedure ของระบบเช่าซื้อจะอาศัย Oracle Developer 6 ช่วยในการสร้าง โดยจะทำการติดต่อไปยัง Server และสร้าง Store Procedure บนฝั่ง Server และใช้ภาษา PL/SQL สร้าง Store Procedure ของระบบ

Package ของ Oracle นั้นจะถูกแบ่งเป็น 2 ส่วนคือส่วนที่เป็น Program Specification และ Program Body ซึ่ง Program Specification จะเปรียบเสมือนเป็น Interface สำหรับใช้งานและภายในจะมีเพียง ชื่อ Function และ/หรือ Procedure และ Parameter ต่างๆ พร้อมกับ ค่าที่ต้องการจะส่งกลับ และส่วน Program Body จะเป็นรายละเอียด Code โปรแกรม ที่มี Function และ/หรือ Procedure ตรงกับที่ประกาศไว้ในส่วนของ Program Specification และอาจมี Function / Procedure เพิ่มภายใน Program Body ได้ แต่จะ Interface ผ่านตรงๆ ไม่ได้ Program Specification ใน Package นั้นเปรียบเป็นการประกาศ Visibility เป็น Public ของ Object-Oriented และส่วน Program Body ใน Package ก็เปรียบเป็นการประกาศ Visibility เป็น Private ใน Object-Oriented

## 4.2 การพัฒนาบนฝั่ง Client

การพัฒนาบนฝั่ง Client นั้นเป็นการสร้าง UI (User Interface) โดยนำ UI ที่ได้จากขั้นการออกแบบมาพัฒนา โดยการพัฒนา GUI จะใช้ Oracle Developer 6 โดยใช้ในการสร้าง Form และ Report

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

# สรุป

## 5.1 สรุปผลการวิเคราะห์และออกแบบ

จากการวิเคราะห์และออกแบบระบบลูกหนี้เช่าซื้อของการเคหะแห่งชาติ จะมีขั้นตอนในการทำงานโดยศึกษาระบบงาน ศึกษาปัญหาของระบบ รวบรวมรายละเอียดความต้องการของระบบโดยโดยใช้วิธีการที่มีอยู่ เช่น การรวบรวมเอกสาร เป็นต้น เมื่อวิเคราะห์ หลังจากทำการวิเคราะห์ระบบแล้ว จึงทำการออกแบบระบบ โดยมีการใช้วิธีการออกแบบเชิงวัตถุที่มีการใช้ Diagram ต่างๆ มาช่วย และมีขั้นตอนการทำงานที่สามารถย้อนกลับได้ การแก้ไขทำได้ง่าย เมื่อดำเนินการพัฒนาระบบพบว่ามีปัญหาที่เกิดขึ้นคือในการวิเคราะห์ และออกแบบระบบโดยใช้วิธีการเชิงวัตถุซึ่งแตกต่างกับการใช้วิธีเดิมที่มีอยู่ทำให้ผู้พัฒนาระบบยังไม่มีความชำนาญในการใช้เครื่องมือ ทำให้มีข้อผิดพลาดบ้างในการพัฒนาระบบ และจากการวิเคราะห์และออกแบบระบบเป็นเชิงวัตถุมีการใช้เครื่องมือในการสร้าง Diagram ต่างๆ และทำการสร้าง Class ก่อนที่จะแปลงไปเป็นโครงสร้างฐานข้อมูลแบบ Relational ของ Oracle และมีการพัฒนาระบบโดยใช้ Developer ซึ่ง Rose มีความสามารถในการใช้งานได้ระดับหนึ่ง และการติดต่อสื่อสารของระบบเป็นการเรียกใช้ procedure ต่างๆ และมีการสื่อสารกันระหว่าง Class ในลักษณะที่เป็นแบบ Object ทำให้การใช้วิธีการ join ของหลายๆตารางใช้งานไม่ได้

## 5.2 การพัฒนาระบบเชิงวัตถุ

ในการพัฒนาระบบเชิงวัตถุเป็นการพัฒนาระบบที่ดีกว่าในการพัฒนาระบบแบบเดิม เนื่องจากในแต่ละขั้นตอนมีลักษณะการทำงานที่ต่อเนื่องกัน ทำให้การเปลี่ยนแปลงจากขั้นตอนหนึ่งไปยังอีกขั้นตอนหนึ่งทำได้ง่าย มีความต่อเนื่องเข้าใจได้เพราะมีการใช้แบบจำลองที่มีลักษณะเหมือนกันตลอดจึงทำให้ต่อเนื่องกัน ในขณะที่วิธีการแบบเดิมในแต่ละขั้นตอนจะไม่ต่อเนื่องเพราะมีการใช้แบบจำลองที่ต่างกันในแต่ละขั้นตอน ทำให้ต้องทำความเข้าใจในแต่ละแบบจำลอง ในลักษณะของ Object จะเป็นการจัดการกับข้อมูลของตนเอง และมีวิธีการทำงานภายในตัวเองได้เนื่องจากมีทั้งข้อมูล และ Method ในตัวเอง ดังนั้นเมื่อทำการเปลี่ยนแปลงภายในก็จะไม่มีผลกระทบเกิดขึ้น ยังคงใช้งานได้เหมือนเดิม รวมทั้งมีเรื่องของการนำกลับมาใช้ใหม่ได้ ทำให้การพัฒนาระบบมีความรวดเร็วขึ้น รวมทั้งปัจจุบันแนวคิดเรื่องของ Object มีมากขึ้น มีเครื่องมือใช้งานมากขึ้น เช่น Rational Rose เป็นต้น ทำให้การพัฒนาระบบเชิงวัตถุมีแนวโน้มที่จะมีการใช้งานมากขึ้น ดังนั้นในการพัฒนาระบบเช่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ซื้อ การเคหะแห่งชาติโดยใช้วิธีการเชิงวัตถุจึงสามารถทำได้ ซึ่งยังต้องมีการปรับปรุงวิธีการอีกบ้างเพื่อให้ได้ประสิทธิภาพที่ดีขึ้น**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6

# คู่มือระบบ

## 6.1 ขอบเขต (Scope) ของระบบ

### 6.1.1 ผู้อ่าน (Audience)

เป็นผู้ที่มีหน้าที่ดูแลรักษาระบบที่ถูกพัฒนาขึ้น และจะต้องมีความรู้ด้านการพัฒนาระบบเล็กน้อย และโดยเฉพาะอย่างยิ่งต้องอ่านสัญลักษณ์ต่างๆ ของ UML (Unified Modeling Language) ได้

### 6.1.2 องค์กร (Organization)

พนักงานของระบบเช่าซื้อเป็นหน่วยงานหนึ่งของการเคหะแห่งชาติ

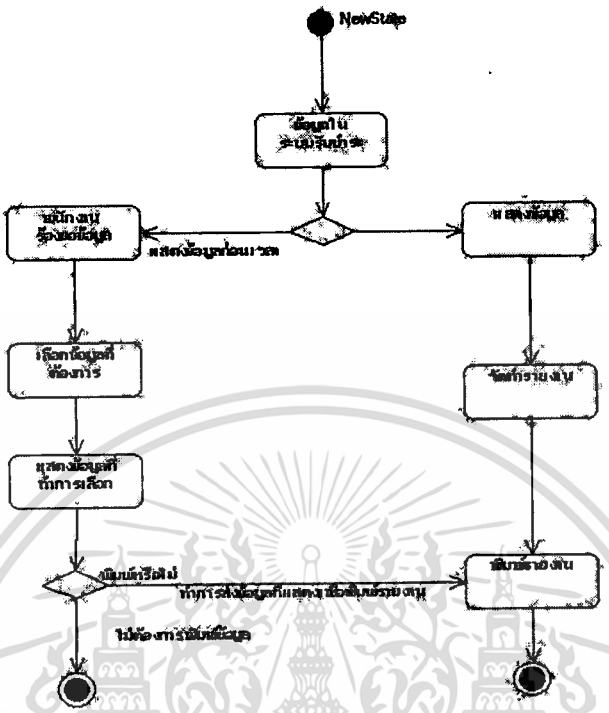
## 6.2 ความต้องการของระบบ (System Requirement)

ระบบสามารถค้นหาข้อมูลเกี่ยวกับการเช่าซื้อและสามารถออกรายงานการเช่าซื้อโดยสามารถดูรายละเอียดลูกค้าของการเช่าซื้อ การรับชำระเงิน การรับชำระเงินล่วงหน้า การค้างชำระลูกหนี้ถึงกำหนดชำระแล้วทำการออกรายงานได้

## 6.3 แบบจำลองของกระบวนการทางธุรกิจ

แบบจำลองของกระบวนการทางธุรกิจของระบบเช่าซื้อจะแสดงดังรูปที่ 6.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 6.1 แสดงกระบวนการของระบบเช่าซื้อ**

## 6.4 แบบจำลอง Use Case

### 6.4.1 Actor

- Officer พนักงานของระบบเช่าซื้อ ทำหน้าที่ในการตรวจสอบการรับชำระเงินของระบบเช่าซื้อ และทำการออกรายงาน

### 6.4.2 Use Case

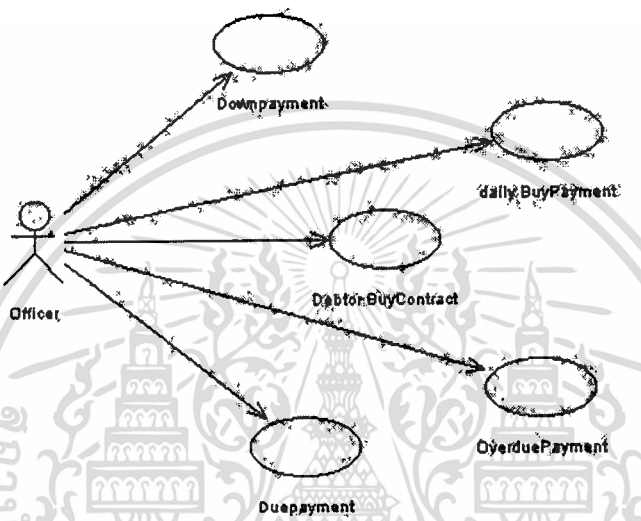
- DownPayment (การรับชำระเงินล่วงหน้า) มีหน้าที่ในการตรวจสอบการรับชำระเงินล่วงหน้าประจำวันของลูกหนี้
- Daily BuyPayment (การรับชำระค่าเช่าซื้อ) มีหน้าที่ในการตรวจสอบการรับชำระค่าเช่าซื้อประจำวันของลูกหนี้
- DebtorBuyContract (สัญญาของลูกหนี้เช่าซื้อ) มีหน้าที่แสดง ประวัติ และสัญญาของลูกค้าที่ทำการเช่าซื้อ
- OverduePayment (การค้างชำระค่าเช่าซื้อ) มีหน้าที่ในการตรวจสอบการค้างชำระของลูกค้าเป็นรายเดือน
- DuePayment (ค่าเช่าซื้อถึงกำหนดชำระ) ทำหน้าที่ตรวจสอบเมื่อครบกำหนดชำระจะแสดงลูกค้าที่ต้องชำระเงินที่เกี่ยวข้องกับในสัญญาเป็นประจำเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use case ทั้งหมดจะมีผู้เริ่มต้นคือ พนักงานของระบบเช่าซื้อ ทำหน้าที่ค้นหารายละเอียดของข้อมูล และทำการออกรายงาน

**6.4.3 Use Case Diagram**

จะแสดง Use Case Diagram ดังรูปที่ 6.2



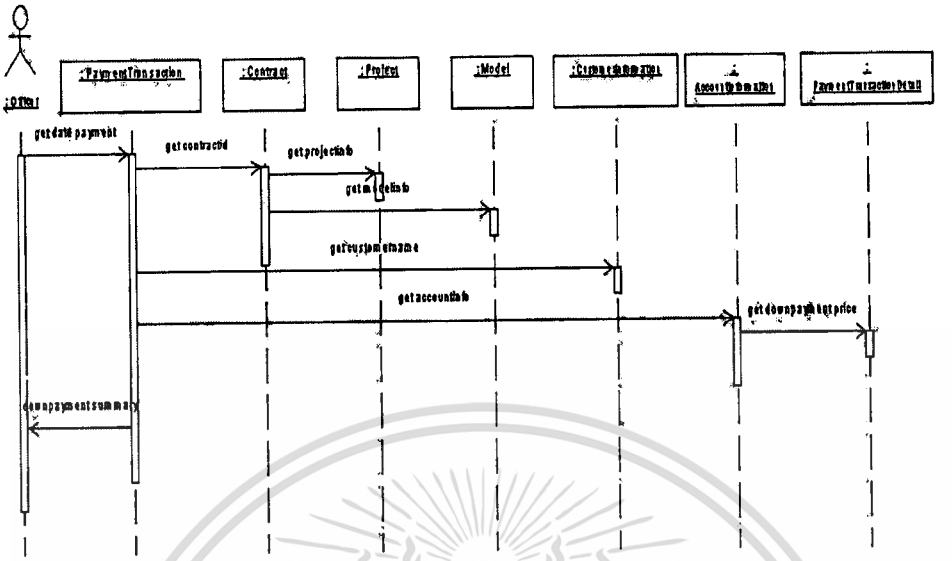

**รูปที่ 6.2 แสดง Use Case Diagram ของระบบเช่าซื้อ**

## 6.5 Interaction Diagram

แสดง Sequence Diagram ของแต่ละ Use Case ดังนี้

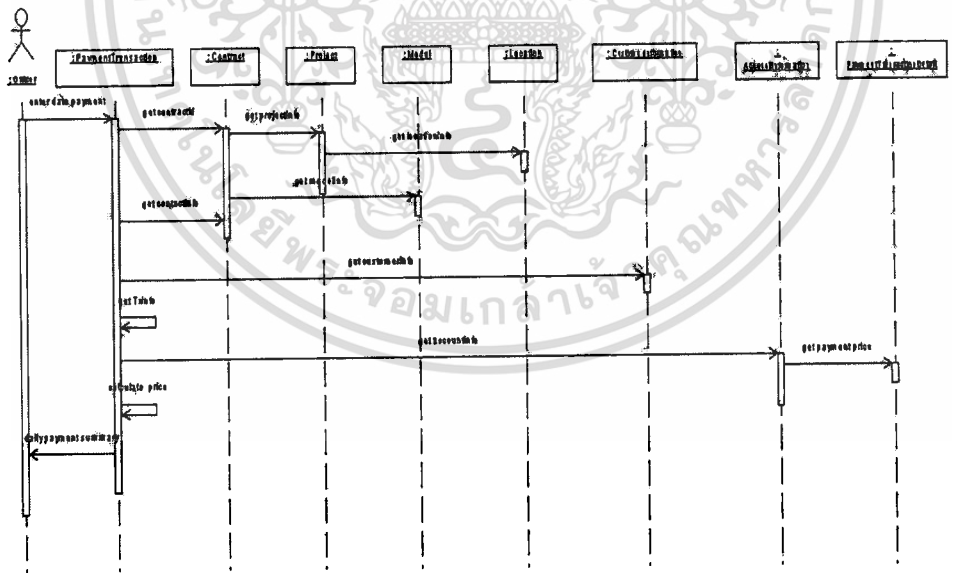
**6.5.1 Sequence Diagram ของ Use Case DownPayment**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 6.3 แสดง Sequence Diagram ของ Use Case DownPayment**

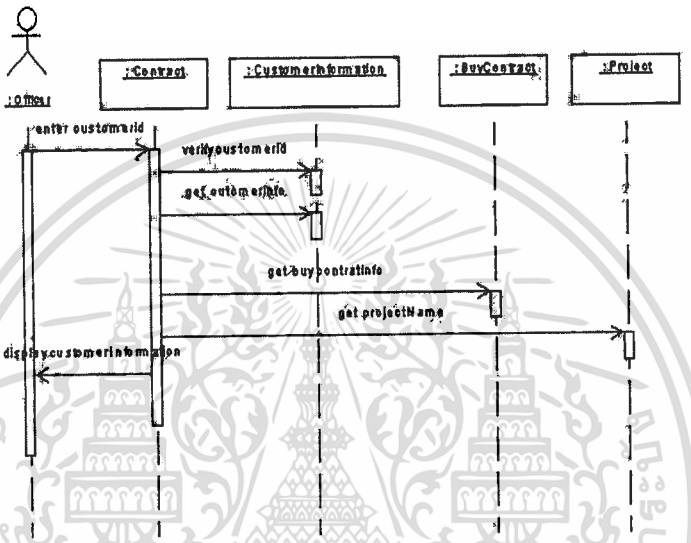
### 6.5.2 Sequence Diagram ของ Daily BuyPayment Use Case



**รูปที่ 6.4 แสดง Sequence Diagram ของ Use Case Daily BuyPayment**

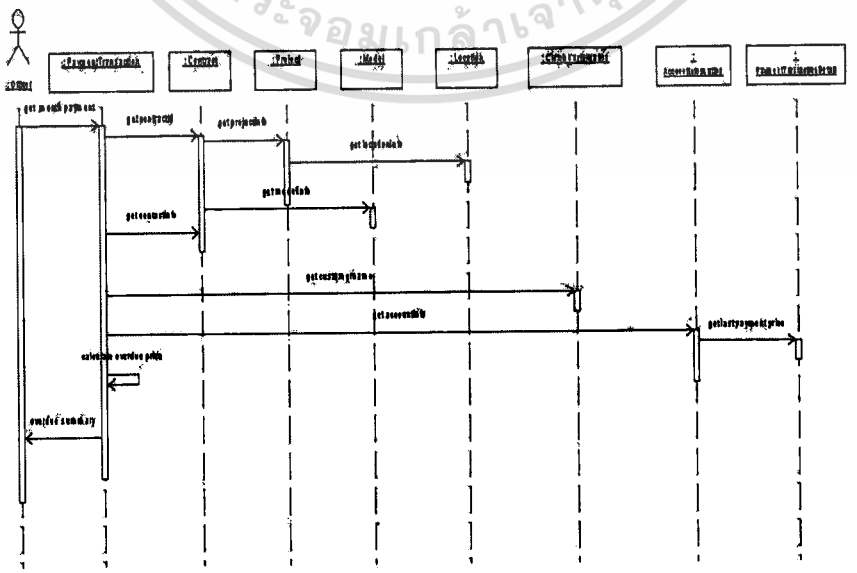
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.5.3 Sequence Diagram ของ Use Case DebtorBuyContract




**รูปที่ 6.5 แสดง Sequence Diagram ของ Use Case DebtorBuyContract**

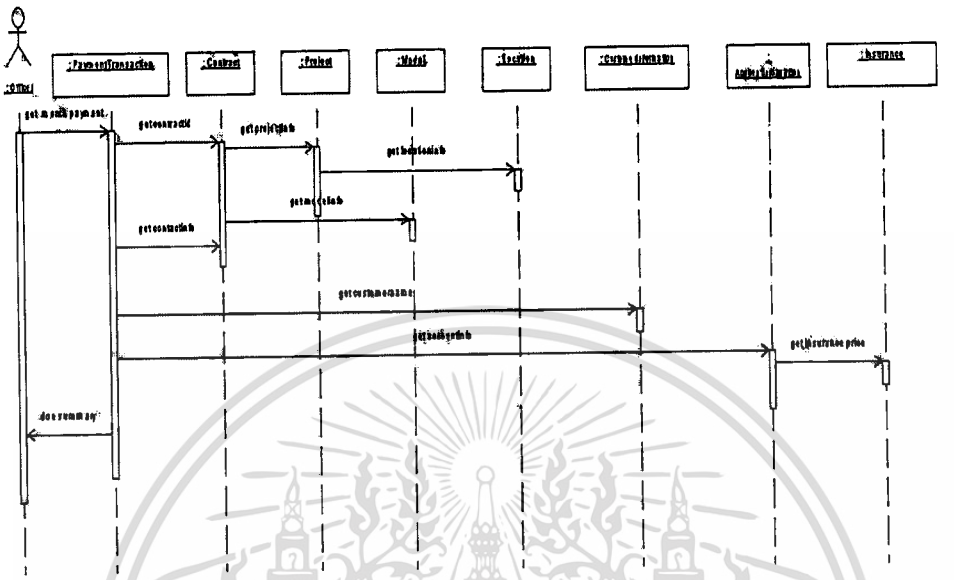
### 6.5.4 Sequence Diagram ของ Use Case OverduePayment



**รูปที่ 6.6 แสดง Sequence Diagram ของ Use Case OverduePayment**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.5.5 Sequence Diagram ของ Use Case DuePayment



**รูปที่ 6.7 แสดง Sequence Diagram ของ Use Case Duepayment**

## 6.6 Class Diagram

### 6.6.1 Class

**ตารางที่ 6.1 แสดง Class ของระบบเช่าซื้อ**

| ชื่อ Class | ความหมาย |
|---|---|
| Location | รายละเอียดเกี่ยวกับที่ตั้งสำนักงานเคหะชุมชน |
| Project | รายละเอียดเกี่ยวกับโครงการ |
| Model | รายละเอียดเกี่ยวกับโมเดล |
| Building Type | รายละเอียดเกี่ยวกับสิ่งปลูกสร้าง |
| Unit | รายละเอียดเกี่ยวกับหน่วยอาศัย |
| Quotation | รายละเอียดข้อมูลขายโครงการ |
| BuyContract | รายละเอียดสัญญาเช่าซื้อ |
| CustomerInformation | รายละเอียดลูกค้า |
| Insurance | รายละเอียดค่าประกันการเช่าซื้อ |
| BuyDownPayment | รายละเอียดการจ่ายเงินดาวน์ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 6.1 แสดง Class ของระบบเช่าซื้อ(ต่อ)**

| | |
|---|---|
| AccountInformation | เป็นรายการบัญชีที่ใช้ระบบรับชำระ ซึ่งเป็นข้อมูลที่ต้องการในการบันทึกการรับชำระแต่ละรายการ |
| Contract | เป็นสัญญาของลูกหนี้ ที่ได้ทำไว้กับทางการเคหะ ซึ่งเป็นข้อมูลที่ต้องการในการรับชำระ |
| PaymentTransaction | เป็น Transaction การรับชำระของระบบ ซึ่งเป็นการเก็บรายละเอียดการรับชำระในแต่ละครั้ง |
| PaymentTransactionDetail | เป็นรายละเอียดของแต่ละ Transaction ของการรับชำระ |

### 6.6.2 ความสัมพันธ์ของ Class

- Contract กับ PaymentTransaction สัญญา (Contract) ของลูกหนี้ทุกสัญญาที่มีการทำไว้กับทางการเคหะ ลูกหนี้ที่ถือสัญญานั้นๆ จะต้องมีการจ่ายเงิน (PaymentTransaction) เกิดขึ้น และสัญญาหนึ่งๆ จะต้องมีการจ่ายเงินหนึ่งครั้งหรือมากกว่าหนึ่งครั้งขึ้นไป
- PaymentTransactionDetail กับ PaymentTransaction ในหนึ่งครั้งของการชำระเงินของลูกหนี้จะมีรายละเอียดการชำระเงิน (Payment-TransactionDetail) เกิดขึ้น เพราะการชำระเงินนั้นอาจมีการชำระได้ตั้งแต่หนึ่งรายการขึ้นไปภายในหนึ่งครั้ง หรือการชำระเงินหนึ่งรายการหรือหลายๆ รายการนั้นจะเกิดขึ้นภายในครั้งหนึ่งๆ เท่านั้นและหากไม่เกิดขึ้นก็จะไม่มีรายละเอียดรายการที่ชำระ
- PaymentTransactionDetail กับ AccountInformation การชำระแต่ละรายการที่ชำระก็จะมีการบอกถึงชนิดบัญชี (AccountInformation) ที่ลูกหนี้ชำระมาซึ่งรายการหนึ่งก็จะบอกถึงชนิดบัญชีหนึ่งเท่านั้นจึงทำให้เกิดความสัมพันธ์แบบ Association
- Location กับ Project ในเขตของสำนักงานเคหะชุมชนหนึ่ง มีหลายโครงการที่อยู่อาศัย ซึ่งเป็นความสัมพันธ์แบบ Association
- BuildingType กับ Model โมเดลแต่ละโมเดลจะอยู่ในประเภทสิ่งปลูกสร้างหนึ่งเท่านั้น เช่น โมเดล F1 เป็นสิ่งปลูกสร้างประเภท บ้านเดี่ยว เป็นความสัมพันธ์แบบ Association
- Project กับ Quotation โครงการหนึ่งจะมีการเสนอขายหลายรายการ ซึ่งรายการหนึ่งสำหรับแต่ละโมเดล เป็นความสัมพันธ์แบบ Association

เอกสารนี้เป็นเอกสารที่ [illegible] ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Quotation กับ Model ข้อมูลการขายแต่ละรายการจะอ้างถึงโมเดล เป็นความสัมพันธ์แบบ Association
- Quotation กับ Unit ในข้อมูลการขายหนึ่งรายการหรือ แต่ละโมเดล จะต้องประกอบด้วยหน่วยอาศัยหลายหน่วย เป็นความสัมพันธ์ Aggregation
- Contract กับ Unit ในสัญญา 1 สัญญาจะมีหน่วยอาศัยได้ 1 หน่วยอาศัย และ 1 หน่วยอาศัยจะมีอยู่ในใบสัญญาได้ หลายใบ เป็นความสัมพันธ์แบบ Association
- BuyContract กับ Contract สัญญาเช่าซื้อเป็นสัญญาที่มาจากสัญญา มีรายละเอียดเพิ่มเติมเกี่ยวกับการเช่าซื้อ เป็นความสัมพันธ์แบบ Generalization
- BuyDownPayment กับ BuyContract สัญญาเช่าซื้อประกอบด้วยเงินดาวน์หลายงวดและสัญญาเช่าซื้ออาจไม่มีเงินดาวน์ก็ได้ เป็นความสัมพันธ์แบบประกอบกัน
- Insurance กับ Contract สัญญา 1 สัญญามีค่าประกันได้หลายค่า และค่าประกันจะอยู่ในสัญญา 1 สัญญา เป็นความสัมพันธ์แบบ Aggregation
- Insurance กับ AccountInformation ค่าประกันจะมีบัญชี 1 บัญชี และ 1บัญชีจะมีจะมีค่าประกันได้หลาย ค่า เป็นความสัมพันธ์แบบ Aggregation
- Customerinformation กับ Contract ลูกหนี้ 1 คนมีสัญญาได้หลายสัญญา และสัญญา 1 สัญญามีลูกหนี้ได้คนเดียว เป็นความสัมพันธ์แบบ Aggregation

### 6.6.3 Attribute

Attribute ของแต่ละ Class จะแสดงโดยจะแสดงชื่อ Attribute ประเภทข้อมูล พร้อมคำอธิบายของแต่ละ Attribute ดังตารางที่ 6.2

**ตารางที่ 6.2 แสดงรายละเอียดของ Attribute**

| Class AccountInformation | | |
|---|---|---|
| AccountID | CHAR(4) | รหัสบัญชี เช่น 0520 |
| AccountName | VARCHAR2(30) | ชื่อบัญชี เช่นค่าเช่า, เช่าซื้อ (Install), ค่าบริการ (Service),ค่าประกัน (Insurance), ภาษี (Tax), ค่าที่ดินส่วนเกิน (Ex-land), สินเชื่อ (Loan), ค่าปรับ (ReSchedule), อื่นๆ (Other) |
| AccountDescription : | VARCHAR2(255) | คำอธิบายเพิ่มเติมแต่ละชื่อบัญชี |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 6.2 แสดงรายละเอียด Attribute ของ Class ต่างๆ(ต่อ)**

| | | |
|---|---|---|
| Class Contract | | |
| ContractID | CHAR(8) | เลขที่สัญญา |
| ContractDate | DATE | วันที่ทำสัญญา |
| ContractStatus | CHAR(1) | สถานะสัญญาใช้ U แทนสัญญาที่ใช้งาน, T แทนสัญญาที่รอการปรับปรุงเปลี่ยนแปลง และ A แทนสัญญาที่มีการชำระครบแล้ว |
| PaymentType | CHAR(1) | ประเภทการชำระเงิน ใช้ C แทนชำระเงินสด และ B แทนชำระผ่านธนาคาร |
| pricePerTerm | NUMBER(11,2) | เงินที่ต้องจ่ายต่องวด |
| dateFirstTerm | DATE | วันที่ทำการจ่ายเงินงวดแรก |
| term | NUMBER(5) | จำนวนงวดที่ต้องจ่าย |
| contractType | CHAR(1) | ประเภทของสัญญาใช้ B แทนสัญญาเช่าซื้อ, R แทนสัญญาเช่า และ L แทนสัญญากู้เงิน |
| ContractIDNew | CHAR(8) | รหัสสัญญาที่เก็บไว้กรณีเปลี่ยนสัญญา |
| Class PaymentTransaction | | |
| txID | NUMBER(15) | รหัสสำหรับแต่ละ Transaction |
| paymentDate | DATE | วันที่ของ Transaction ที่รับชำระ |
| paymentTerm | NUMBER(5) | งวดที่ชำระ |
| Class PaymentTransactionDetail | | |
| paymentPrice | NUMBER(11,2) | จำนวนเงินที่ชำระต่อรายการบัญชี |
| Class Location | | |
| LocationID | CHAR(4) | รหัสสำนักงานเคหะชุมชน |
| LocatinName | VARCHAR2(50) | ชื่อสำนักงานเคหะชุมชน |
| Limit | VARCHAR2(2) | เขต |
| HouseNo | VARCHAR2(9) | บ้านเลขที่ |
| Moo | VARCHAR2(2) | หมู่ที่ |
| Soi | VARCHAR2(3) | ซอย |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 6.2 แสดงรายละเอียด Attribute ของ Class ต่างๆ(ต่อ)**

| | | |
|---|---|---|
| Road | VARCHAR2(30) | ถนน |
| Province | VARCHAR2(30) | จังหวัด |
| PostalCode | VARCHAR2(8) | รหัสไปรษณีย์ |
| Class Project | | |
| ProjectID | CHAR(4) | รหัสโครงการ |
| ProjectName | VARCHAR2(100) | ชื่อโครงการ |
| Class Model | | |
| ModelID | CHAR(2) | รหัสโมเดล |
| ModelName | VARCHAR2(20) | ชื่อโมเดล |
| Class Quoation | | |
| unitAmount = 0 | NUMBER(3) | จำนวนหน่วยอาศัย |
| space = 0.00 | NUMBER(5,2) | เนื้อที่ต่อหน่วย(ตร.ว.) |
| cost = 0.00 | NUMBER(11,2) | ราคาขายเงินสดต่อหน่วย(บาท) |
| StandardInterest | NUMBER(4,2) | อัตราดอกเบี้ยการผ่อนชำระค่าเช่าซื้อต่อปี(%) |
| Class BuildingType | | |
| BuildingTypeID | CHAR(2) | รหัสประเภทสิ่งปลูกสร้าง |
| BuildingTypeName | VARCHAR2(50) | ชื่อประเภทสิ่งปลูกสร้าง |
| Class Unit | | |
| HouseNo | VARCHAR2(9) | บ้านเลขที่ |
| DebtorCode | CHAR(14) | รหัสลูกหนี้ |
| FloorNo | NUMBER(2) | หมายเลขชั้น |
| BuildingNo | NUMBER(2) | หมายเลขอาคาร |
| Class BuyContract | | |
| BuyInterestRate | NUMBER(4,2) | อัตราดอกเบี้ยของการเช่าซื้อ |
| buyInterestTotal | NUMBER(7,2) | ดอกเบี้ยทั้งหมดของการเช่าซื้อ |
| AmountBuy | NUMBER(9,2) | จำนวนเงินทั้งหมดของการเช่าซื้อ |
| AmountDownPayment | NUMBER(9,2) | จำนวนเงินดาวทั้งหมด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 6.2 แสดงรายละเอียด Attribute ของ Class ต่างๆ(ต่อ)**

| | | |
|---|---|---|
| TermDownPayment | NUMBER(3,0) | จำนวนงวดที่ต้องจ่ายเงินดาวน์ |
| dateFirstDownPayment | DATE | วันที่จ่ายเงินดาวน์งวดแรก |
| Class BuyDownPayment | | |
| fromtermDownPayment | NUMBER(3,0) | เงินดาวน์งวดที่เริ่มจ่าย |
| toTermDownPayment | NUMBER(3,0) | เงินดาวน์งวดสุดท้ายที่จ่าย |
| pricePerTermDownPayment | NUMBER(7,2) | เงินดาวน์ที่ต้องจ่ายแต่ละงวด |
| Class Insurance | | |
| InsurancePrice | NUMBER(7,2) | จำนวนเงินค่าประกันต่างๆ |
| Class CustomerInformation | | |
| customerID | CHAR(13) | รหัสลูกค้า |
| title | VARCHAR2(10) | คำนำหน้าชื่อ |
| firstName | VARCHAR(20) | ชื่อ |
| lastName | VARCHAR(30) | นามสกุล |
| regisAddress | VARCHAR(255) | ที่อยู่ตามบัตรประชาชน |
| CurrentAddress | VARCHAR(255) | ที่อยู่ปัจจุบัน |
| Telephone | CHAR(15) | เบอร์โทรศัพท์ |

**6.6.4 Method**

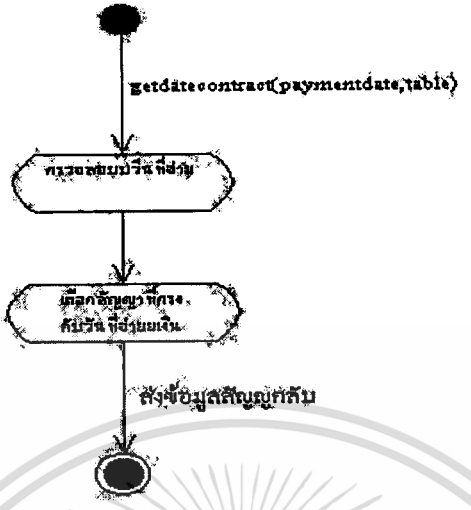
ในแต่ละ Class จะมี method ต่างๆ โดยจะแสดงชื่อ Method, Parameter พร้อมคำอธิบายของแต่ละ Method ดังนี้

- AccountInformation
  - getAccountName(accountid :CHAR) return CHAR;

    **ค้นหาชื่อบัญชีจากรหัสบัญชี**
- Paymenttransaction
  - getDateContract(paymentdate :DATE,tableOfContractID : TAB_TMP);

    **ค้นหารายละเอียดสัญญาจากวันที่จ่ายเงิน แสดงดังรูปที่ 6.8**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 6.8 แสดง Activity Diagram ของ getdatecontract method**

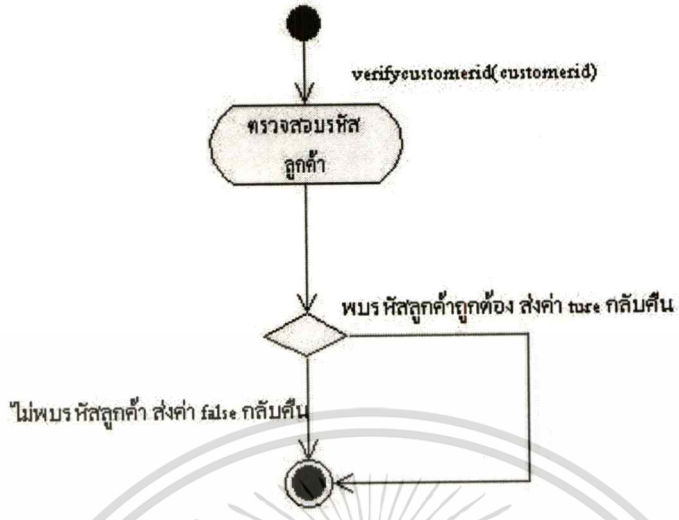
- verifyDate(paymentdate :DATE) return BOOLEAN;
  ตรวจสอบวันที่จ่ายเงิน
- gettermdown(contractid :CHAR) return number;
  ค้นหางวดที่จ่ายเงินดาวน์จากเลขที่สัญญา
- calculatetermMax(contractid :CHAR) return number;
  ค้นหาจำนวนเทอมมากสุดที่จ่ายเงินของเลขที่สัญญา แสดงดังรูปที่ 6.9

**รูปที่ 6.9 แสดง Activity Diagram ของ calculatetermMax method**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Paymenttransactiondetail
  - getPayListContract(paymentdate :DATE, contractid :CHAR,tableOfAccount :TAB_TMP);
    ค้นหารายละเอียดรายการการจ่ายเงินจากวันที่จ่ายเงินและเลขที่สัญญา
  - getPayListContractDown(paymentdate : DATE, contractid : CHAR ,tableOfAccountid :TAB_TMP);
    ค้นหารายละเอียดการจ่ายเงินชำระล่วงหน้าจากวันที่จ่ายเงินและเลขที่สัญญา
- Contract
  - getCustContract(contractID :CHAR,tableOfContractID : TAB_TMP);
    ค้นหารายละเอียดของสัญญาจากเลขที่สัญญา
  - getNewCont(tableOfContractID :TAB_TMP,tableOfContractID :TAB_TMP, projectID :CHAR,modelID :CHAR);
    ค้นหาชุดข้อมูลของสัญญาจากชุดข้อมูลเดิมโดยเปรียบเทียบกับรหัสโครงการและรหัสโมเดล
  - gethouseid(contractID :CHAR)return CHAR;
    ค้นหาเลขที่บ้านจากเลขที่สัญญา
  - getcontractid(tableOfcontractID :CHAR);
    ค้นหาเลขที่สัญญาทั้งหมด
  - getbuyterm(contractid :CHAR) return number;
    ค้นหาจำนวนเทอมการจ่ายเงินค่าเช่าซื้อจากเลขที่สัญญา
  - getpaytype(contractid :CHAR) return CHAR;
    ค้นหาประเภทของการจ่ายเงินจากเลขที่สัญญา
- Customerinformation
  - getCustlistcustomer(customerID :CHAR,tableOfCustomerID :TAB_TMP);
    หา List (ชุดข้อมูล) ของลูกค้าจากเลขที่ลูกค้า
  - verifycustomerid(customerID :CHAR) return BOOLEAN;
    ตรวจสอบรหัสลูกค้าว่าถูกต้องหรือไม่ แสดง Activity Diagram ดังรูปที่ 6.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 6.10 แสดง Activity Diagram ของ verifycustomerid method**

- getfname(contractID :CHAR) return CHAR;
  ค้นหาชื่อของลูกค้าจากรหัสสัญญา
- getlname(contractID :CHAR) return CHAR;
  ค้นหานามสกุลลูกค้าจากรหัสสัญญา

- Insurance
  - getinsureP(contractid :CHAR) return CHAR;
    ค้นหาค่าประกันจากเลขที่สัญญา
- Location
  - getlocationname(projectID :CHAR) return CHAR ;
    ค้นหาชื่อสำนักงานการเคหะจากรหัสโครงการ
  - getprovname(projectID :CHAR) return CHAR;
    ค้นหาชื่อจังหวัดจากรหัสโครงการ
- Project
  - getprojectname(projectID :CHAR) return CHAR;
    ค้นหาชื่อโครงการจากรหัสโครงการ
  - getprojectlist(tableOfProjectID);
    ค้นหารายละเอียดชุดข้อมูลของโครงการ
- Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- getmodelname(modelID :CHAR) return CHAR;

  ค้นหาชื่อโมเดลจากรหัสโมเดล

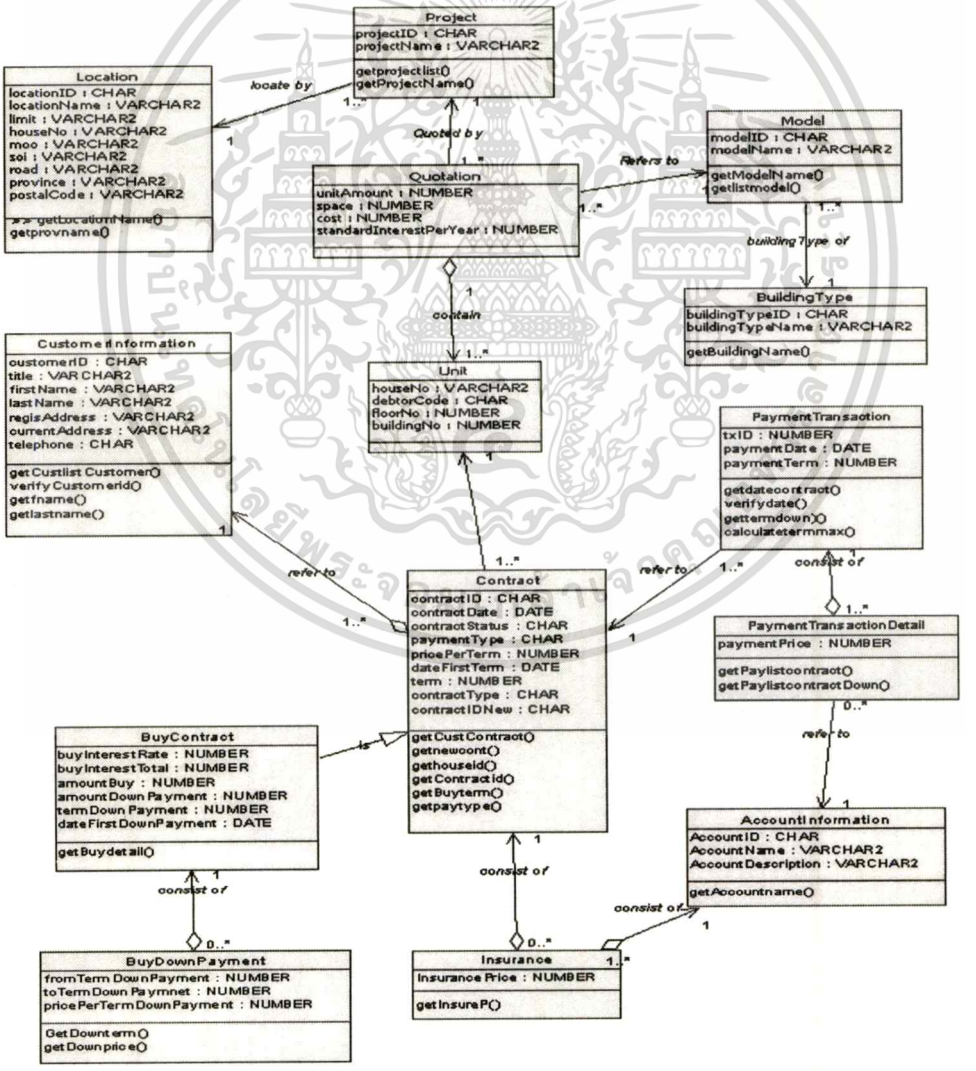
- getlistModel(projectID :CHAR,tableOfModelID);

  ค้นหารายละเอียดของชุดขุอมูลโมเดลจากรหัสโครงการ

- Buildingtype
  - getBuildingname(modelid :CHAR) return CHAR;

    ค้นหาประเภทอาคารจากรหัสโมเดล

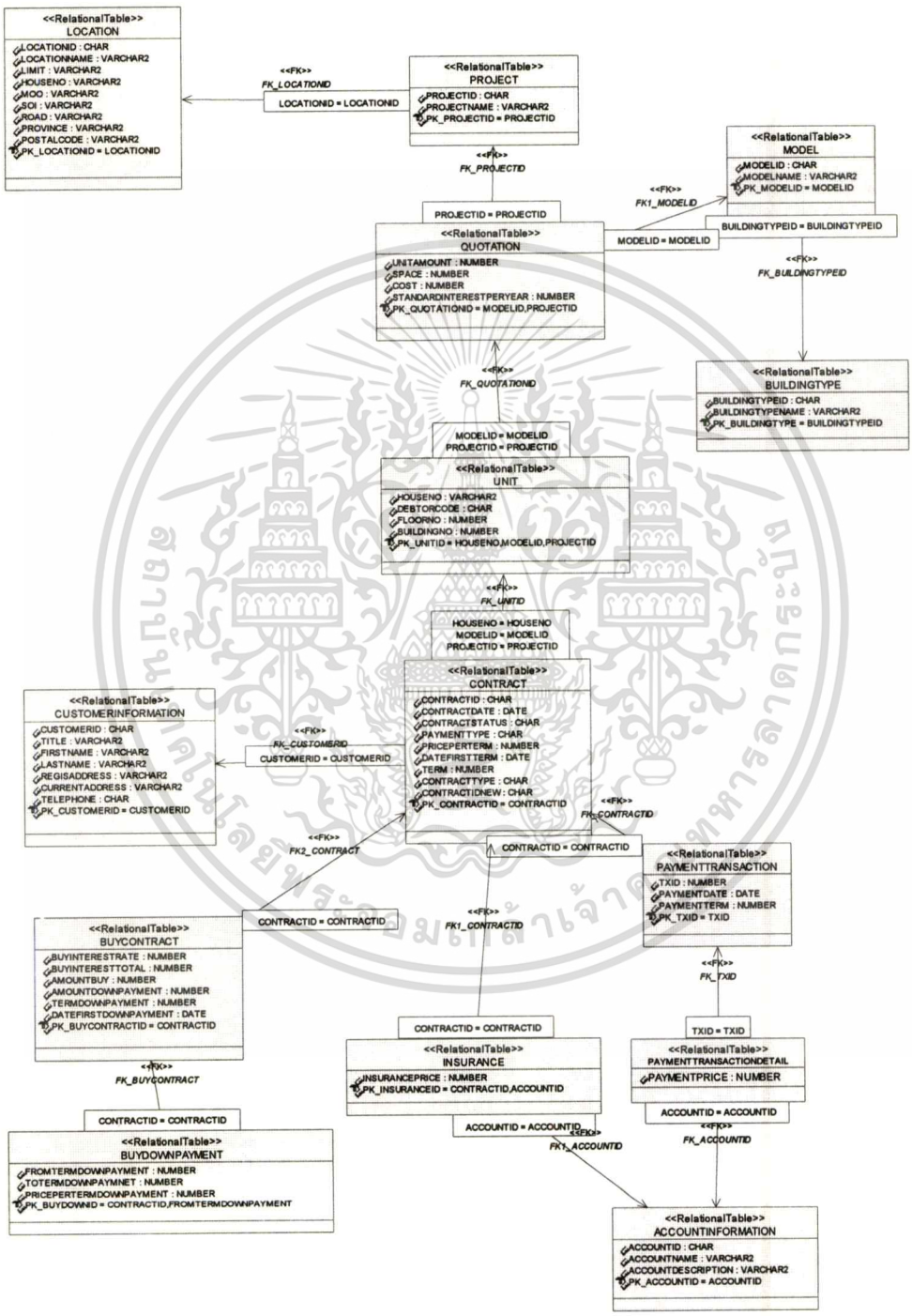
### 6.6.5 Class Diagram in Application Logic




**รูปที่ 6.11 แสดง Class Diagram ในส่วนของ Application Logic**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

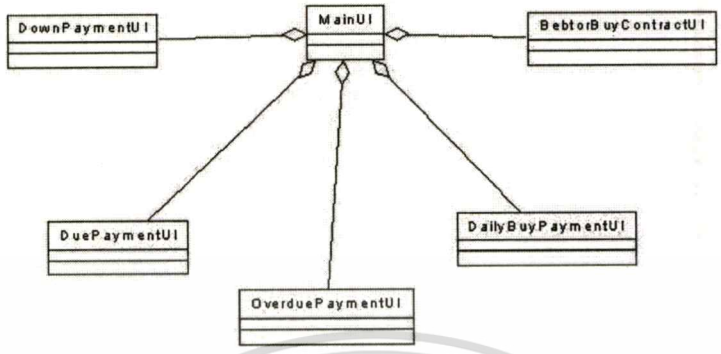
### 6.6.6 Class Diagram in Access Layer




**รูปที่ 6.12 แสดง Schema ของระบบฐานข้อมูลโดยใช้ Class Diagram**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.6.7 Class Diagram in View Layer




**รูปที่ 6.13 แสดง Class Diagram ในส่วนของ View Layer**

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 7

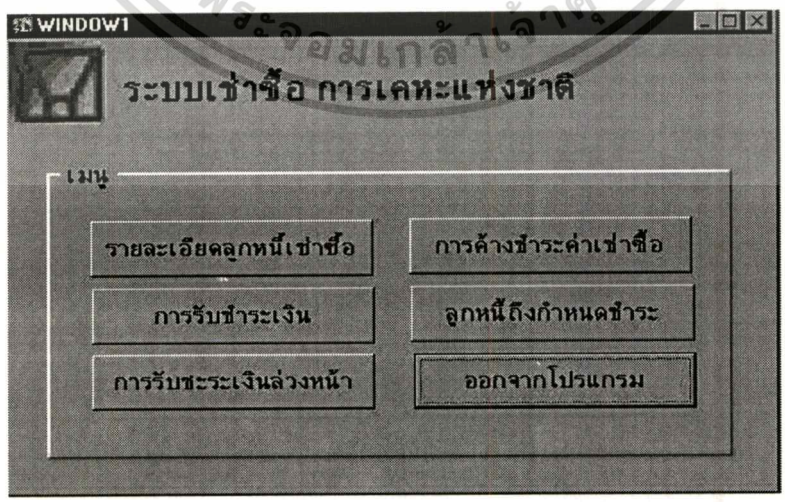
# คู่มือผู้ใช้งาน

## 7.1 การเข้าสู่โปรแกรม

การเข้าสู่โปรแกรมจะต้องมีการ login เข้าสู่ระบบก่อน โดยจะต้องมีการติดตั้ง Oracle Client ลงบนเครื่องที่ต้องการทำงานก่อนซึ่งแสดงเป็นหน้าจอเข้าสู่ระบบดังรูปที่ 7.1

**รูปที่ 7.1 แสดงการ Login เข้าสู่ระบบ**

จากรูปที่ 7.1 จะมีการใส่ชื่อผู้ใช้ (User Name) รหัสผ่าน (Password) และฐานข้อมูล (Host String) แล้วกดปุ่ม "เชื่อมต่อ" เพื่อเข้าสู่โปรแกรมของระบบซึ่งแสดงหน้าจอหลักดังรูปที่ 7.2



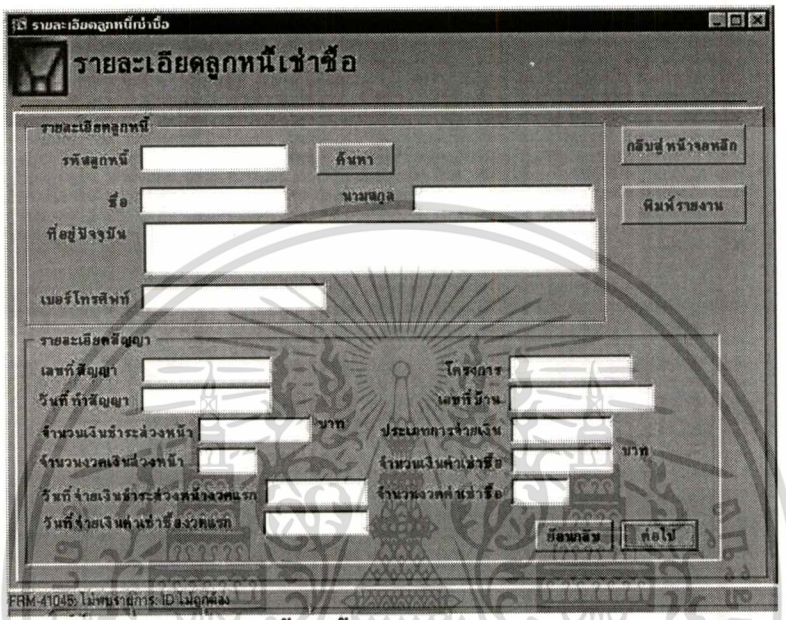

**รูปที่ 7.2 แสดงหน้าต่างหลักของโปรแกรม**

จากรูปที่ 7.2 สามารถเลือกเมนูที่ต้องการหรือออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.2 รายละเอียดลูกหนี้เช่าซื้อ

จากรูปที่ 7.2 เป็นหน้าจอหลักของโปรแกรม กดปุ่ม "รายละเอียดลูกหนี้เช่าซื้อ" ซึ่งจะแสดงหน้าจอรายละเอียดลูกหนี้เช่าซื้อดังรูปที่ 7.3

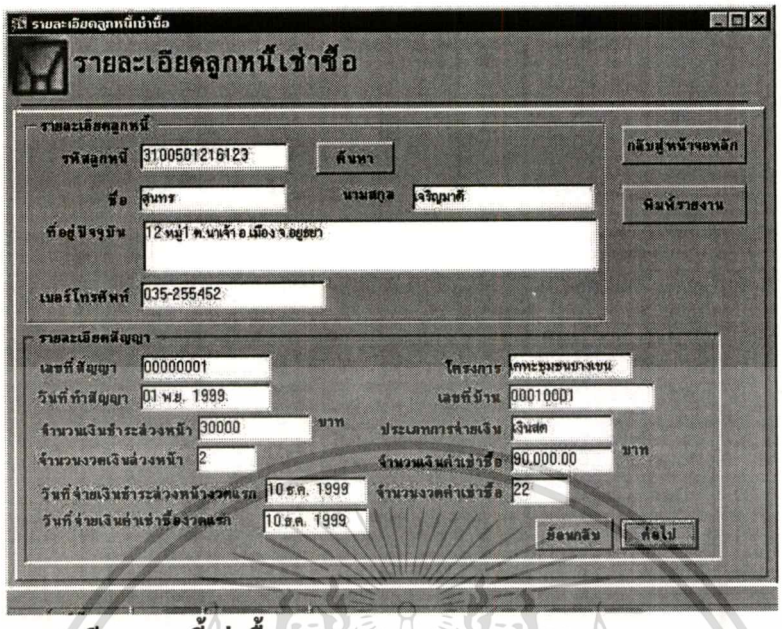



**รูปที่ 7.3 แสดงหน้าจอรายละเอียดลูกหนี้เช่าซื้อ**

จากรูป 7.3 มีการทำงานดังนี้

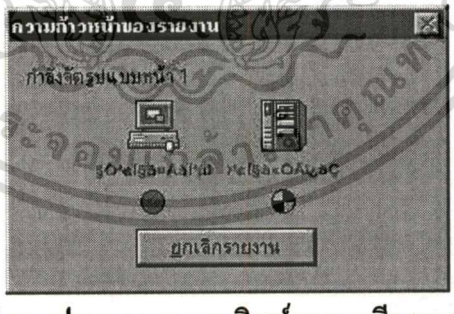
- ทำการใส่รหัสลูกค้าที่ต้องการ แล้วกดปุ่ม "ค้นหา" จะมีการแสดงรายละเอียดของลูกค้า และรายละเอียดของสัญญาเช่าซื้อของลูกค้าซึ่งอาจมีหลายสัญญา แสดงดังรูปที่ 7.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 7.4 แสดงรายละเอียดลูกหนี้เช่าซื้อ**

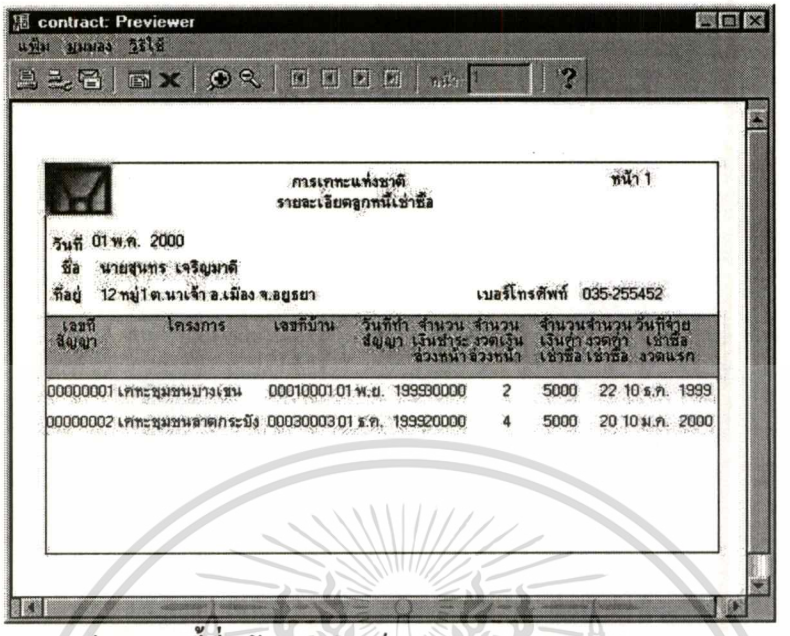
- กดปุ่ม "ต่อไป" เมื่อต้องการดูสัญญาถัดไป
- กดปุ่ม "ย้อนกลับ" เมื่อต้องการดูสัญญาย้อนกลับ
- ถ้าต้องการพิมพ์รายงาน กดปุ่ม "พิมพ์รายงาน" จะขึ้นหน้าจอแสดงความก้าวหน้าในการประมวลผลการพิมพ์ดังรูปที่ 7.5




**รูปที่ 7.5 แสดงหน้าจอความก้าวหน้าการประมวลผลการพิมพ์รายละเอียดลูกหนี้เช่าซื้อ**

- หลังจากนั้นจะขึ้นหน้าจอแสดงรูปแบบรายงานรายละเอียดลูกหนี้เช่าซื้อดังรูปที่ 7.6 แล้วจึงพิมพ์รายงาน
- ถ้าต้องการออกจากหน้าจอรายละเอียดลูกหนี้เช่าซื้อกดปุ่ม "กลับสู่หน้าจอหลัก"

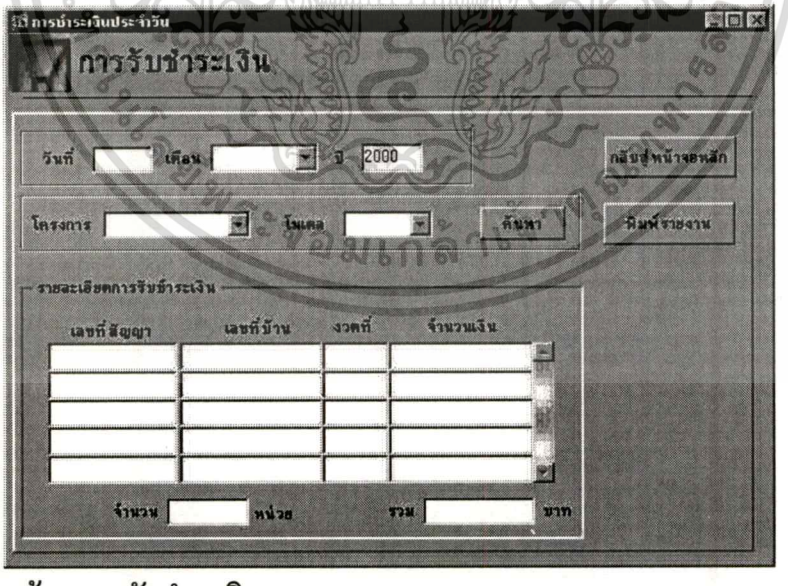
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 7.6 แสดงรายละเอียดลูกหนี้ที่พร้อมจะพิมพ์**

## 7.3 การรับชำระเงิน

จากรูปที่ 7.2 กดปุ่ม “การรับชำระเงิน” จะแสดงหน้าจอการรับชำระเงินดังรูปที่ 7.7



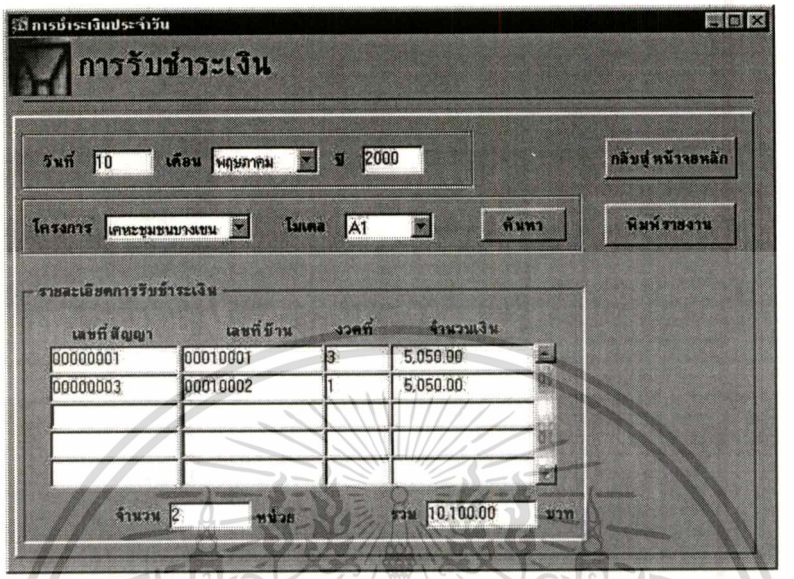

**รูปที่ 7.7 แสดงหน้าจอการรับชำระเงิน**

จากรูป 7.7 จะมีขั้นตอนแสดงรายละเอียดการรับชำระเงินดังนี้

- ใส่วันที่ เลือกเดือน และใส่ปีที่ต้องการแสดงรายละเอียดการรับชำระเงิน
- เลือกโครงการและโมเดลที่ต้องการให้แสดงรายละเอียด

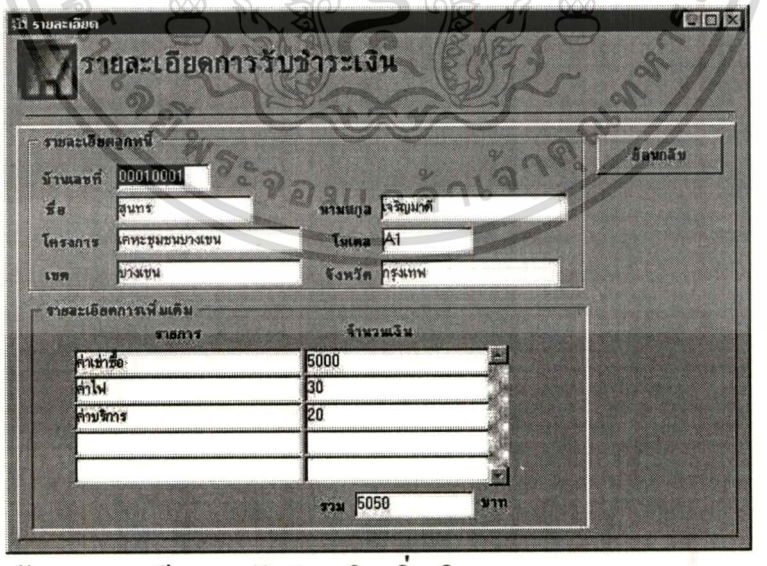
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กดปุ่ม “ค้นหา” จะแสดงรายละเอียดการรับชำระเงินเช่าซื้อแสดงดังรูปที่ 7.8




**รูปที่ 7.8 แสดงรายละเอียดการรับชำระเงิน**

- เมื่อต้องการดูรายละเอียดเพิ่มเติมสามารถดับเบิลคลิกที่เลขที่สัญญาที่ต้องการดูรายละเอียดเพิ่มเติม ซึ่งจะแสดงหน้าจอรายละเอียดการรับชำระเงินดังรูปที่ 7.9

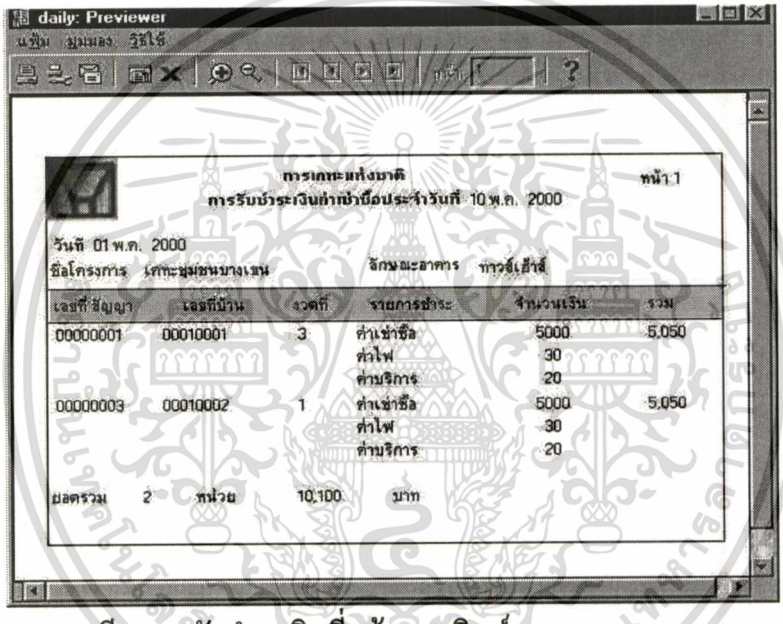



**รูปที่ 7.9 แสดงหน้าจอรายละเอียดการรับชำระเงินเพิ่มเติม**

- จากรูปที่ 7.9 จะมีการแสดงรายละเอียดเกี่ยวกับลูกหนี้ที่จ่ายค่าเช่าซื้อบ้าน และแสดงรายละเอียดรายการที่จ่ายเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อต้องการย้อนกลับไปที่หน้าจอการรับชำระเงิน ให้กดปุ่ม "ย้อนกลับ" ก็จะกลับสู่หน้าจอดังรูปที่ 7.8
- จากรูปที่ 7.8 เมื่อต้องการพิมพ์รายละเอียดข้อมูลให้กดปุ่ม "พิมพ์รายงาน" ก็จะขึ้นหน้าจอแสดงความก้าวหน้าการประมวลผลการพิมพ์ดังรูปที่ 7.5
- หลังจากนั้นจะขึ้นหน้าจอแสดงรูปแบบรายงานรายละเอียดการรับชำระเงินค่าเช่าซื้อดังรูปที่ 7.10 แล้วจึงพิมพ์รายงาน
- ถ้าต้องการออกจากหน้าจอการรับชำระเงินให้กดปุ่ม "กลับสู่หน้าจอหลัก"

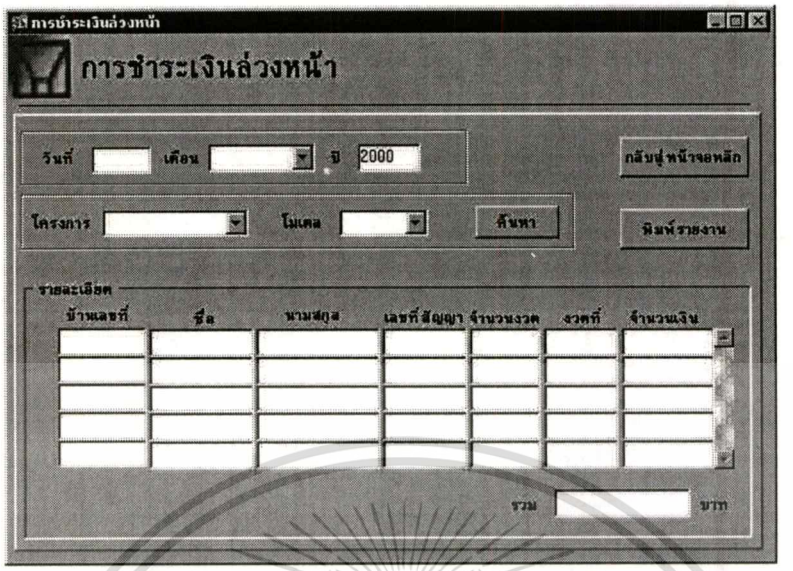



**รูปที่ 7.10 แสดงรายละเอียดการรับชำระเงินที่พร้อมจะพิมพ์**

## 7.4 การรับชำระเงินล่วงหน้า

จากหน้าจอหลักรูปที่ 7.2 กดปุ่ม "การรับชำระเงินล่วงหน้า" จะแสดงดังรูปที่ 7.11

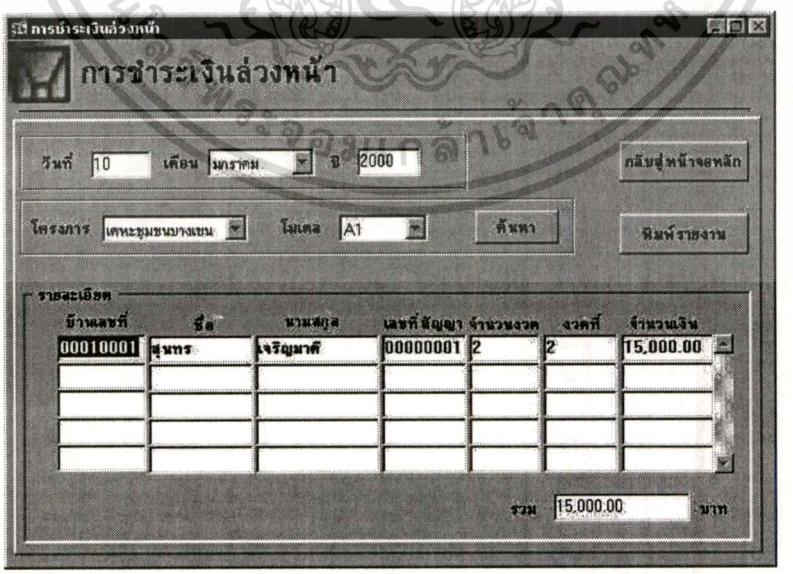
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 7.11 แสดงหน้าจอการรับชำระเงินล่วงหน้า**

จากรูป 7.11 จะมีขั้นตอนแสดงรายละเอียดการรับชำระเงินดังนี้

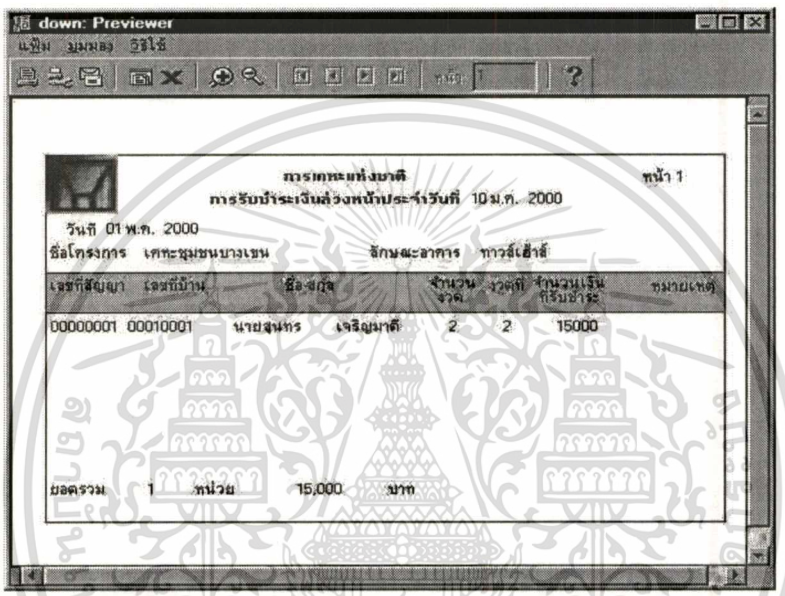
- ใส่วันที่ เลือกเดือน และใส่ปีที่ต้องการแสดงรายละเอียดการรับชำระเงินล่วงหน้า
- เลือกโครงการและโมเดลที่ต้องการให้แสดงรายละเอียด
- กดปุ่ม "ค้นหา" จะแสดงรายละเอียดการรับชำระเงินล่วงหน้าจะแสดงดังรูปที่ 7.12




**รูปที่ 7.12 แสดงรายละเอียดการรับชำระเงินล่วงหน้า**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อต้องการพิมพ์รายละเอียดข้อมูลให้กดปุ่ม “พิมพ์รายงาน” ก็จะขึ้นหน้าจอแสดงความก้าวหน้าในการประมวลผลการพิมพ์ดังรูปที่ 7.5
- หลังจากนั้นจะขึ้นหน้าจอแสดงรูปแบบรายงานการชำระเงินล่วงหน้าดังรูปที่ 7.13 แล้วจึงพิมพ์รายงาน
- ถ้าต้องการออกจากหน้าจอการรับชำระเงินให้กดปุ่ม ”กลับสู่หน้าจอหลัก”

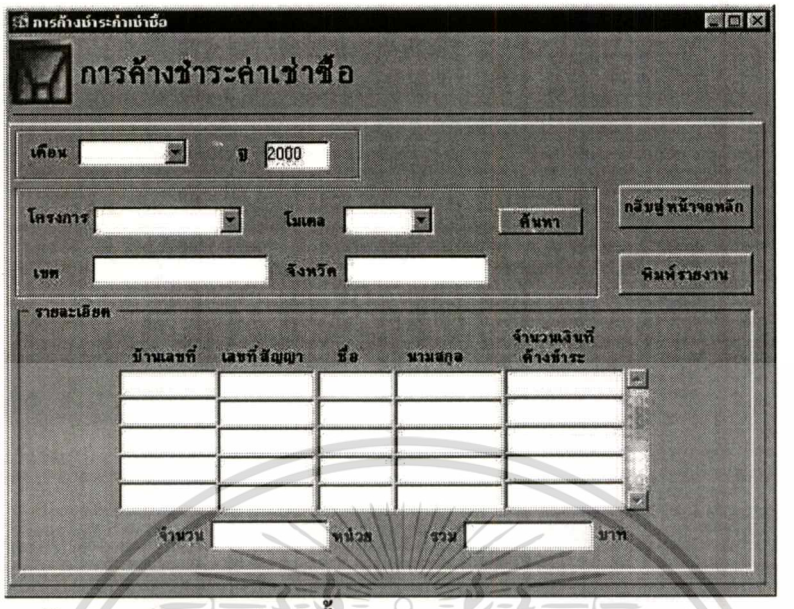



**รูปที่ 7.13 แสดงรายละเอียดการรับชำระเงินล่วงหน้าพร้อมจะพิมพ์**

## 7.5 การค้างชำระค่าเช่าซื้อ

จากหน้าจอหลักรูปที่ 7.2 กดปุ่ม “การค้างชำระค่าเช่าซื้อ” จะแสดงหน้าจอดังรูปที่ 7.14

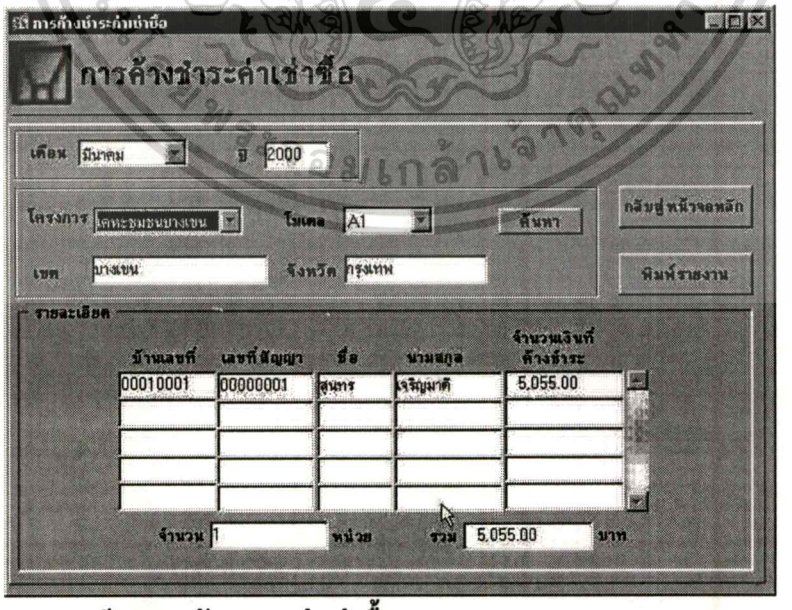
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 7.14 แสดงหน้าจอการค้างชำระค่าเช่าซื้อ**

จากรูป 7.14 จะมีขั้นตอนแสดงรายละเอียดการค้างชำระค่าเช่าซื้อดังนี้

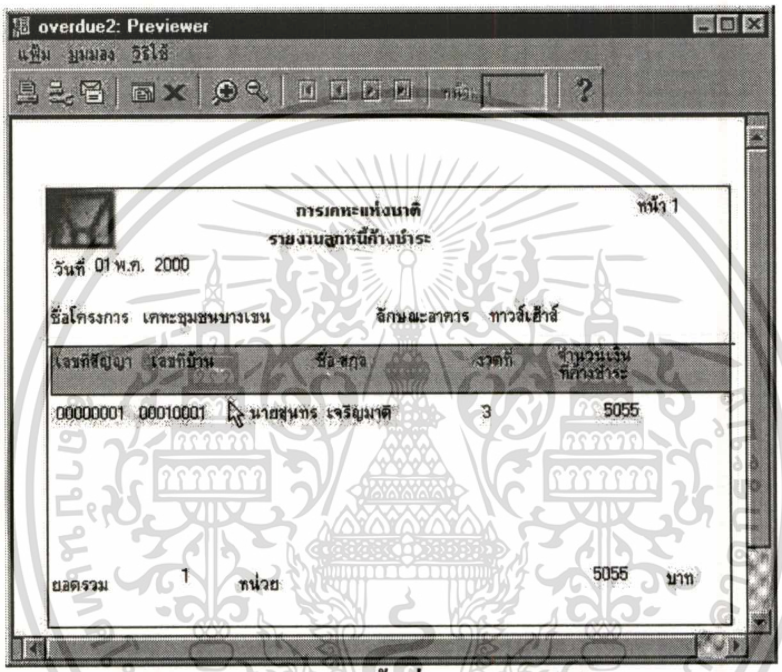
- เลือกเดือน และใส่ปีที่ต้องการแสดงรายละเอียดการรับชำระเงิน
- เลือกโครงการและโมเดลที่ต้องการให้แสดงรายละเอียด
- กดปุ่ม "ค้นหา" จะแสดงรายละเอียดการค้างชำระเงินค่าเช่าซื้อดังรูปที่ 7.15




**รูปที่ 7.15 แสดงรายละเอียดการค้างชำระค่าเช่าซื้อ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อต้องการพิมพ์รายละเอียดข้อมูลให้กดปุ่ม “พิมพ์รายงาน” ก็จะขึ้นหน้าจอแสดงความก้าวหน้าในการประมวลผลการพิมพ์ดังรูปที่ 7.5
- หลังจากนั้นจะขึ้นหน้าจอแสดงรูปแบบรายงานรานละเอียดการค้างชำระค่าเช่าซื้อดังรูปที่ 7.16 แล้วจึงพิมพ์รายงาน
- ถ้าต้องการออกจากหน้าจอการรับชำระเงินให้กดปุ่ม ”กลับสู่หน้าจอหลัก”

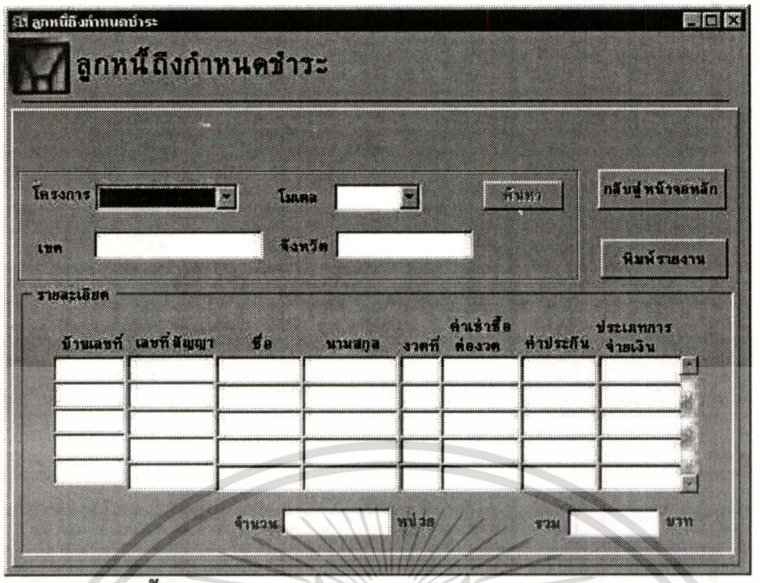



**รูปที่ 7.16 แสดงรายละเอียดการค้างชำระค่าเช่าซื้อที่พร้อมจะพิมพ์**

## 7.6 ลูกหนี้ถึงกำหนดชำระเงิน

จากรูปที่ 7.2 กดปุ่ม “ลูกหนี้ถึงกำหนดชำระ” จะแสดงหน้าจอดังรูปที่ 7.17

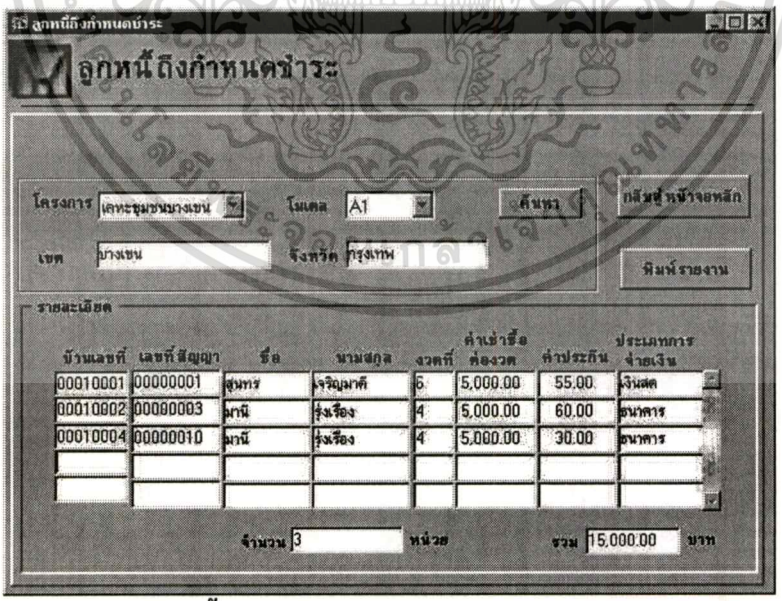
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 7.17 แสดงหน้าจอลูกหนี้ถึงกำหนดชำระ**

จากรูป 7.17 จะมีขั้นตอนแสดงรายละเอียดลูกหนี้ถึงกำหนดชำระดังนี้

- เลือกโครงการและโมเดลที่ต้องการให้แสดงรายละเอียด
- กดปุ่ม "ค้นหา" จะแสดงรายละเอียดลูกหนี้ถึงกำหนดชำระแสดงดังรูปที่ 7.18

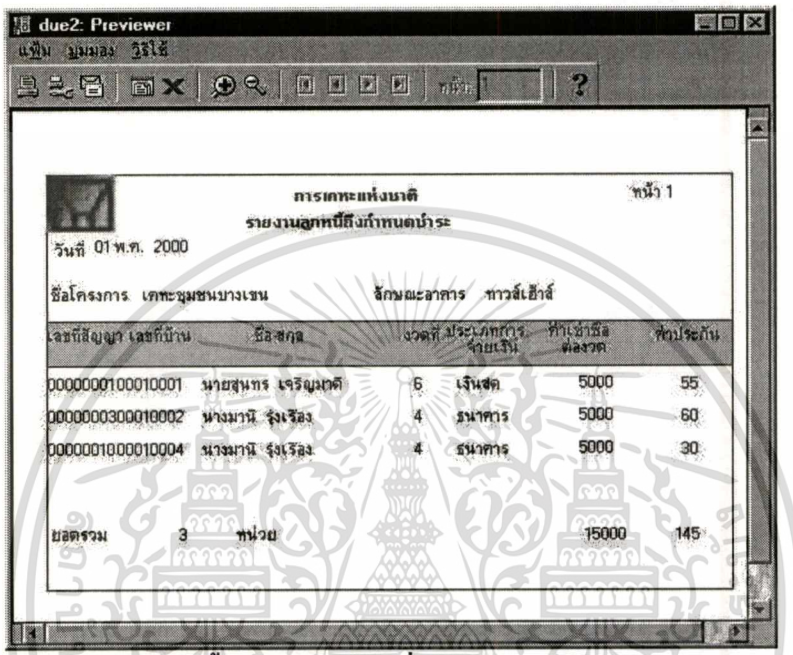



**รูปที่ 7.18 แสดงรายละเอียดลูกหนี้ถึงกำหนดชำระ**

- เมื่อต้องการพิมพ์รายละเอียดข้อมูลให้กดปุ่ม "พิมพ์รายงาน" ก็จะขึ้นหน้าจอแสดงความก้าวหน้าในการประมวลผลการพิมพ์ดังรูปที่ 7.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากนั้นจะขึ้นหน้าจอแสดงรูปแบบรายงานรานละเอียดการค้างชำระค่าเช่าซื้อ ดังรูปที่ 7.19 แล้วจึงพิมพ์รายงาน
- ถ้าต้องการออกจากหน้าจอการรับชำระเงินให้กดปุ่ม "กลับสู่หน้าจอหลัก"




**รูปที่ 7.19 แสดงรายละเอียดลูกหนี้ถึงกำหนดชำระที่พร้อมจะพิมพ์**

## 7.7 การออกจากโปรแกรม

เมื่อต้องการออกจากโปรแกรม จากรูปที่ 7.2 ให้กดปุ่ม "ออกจากโปรแกรม"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

Bahrami, Ali. 1998. **Object Oriented Systems Development using Unified Modeling Language**. Singapore: McGraw-Hill.

Dittman, Kevin C. 1997. **Object Modeling Systems Analysis and Design**. USA: Irwin McGraw-Hill.

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

Quatrani, Terry. 1998. **Visual Modeling with Rational Rose and UML**. USA: Wesley Longman.

Lulushi, Albert. 1998. **Developing Oracle Forms Application.** USA: Prentice Hall PTR.

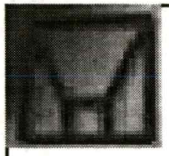
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างแบบพิมพ์ที่ใช้ในการเคหะแห่งชาติ



การเคหะแห่งชาติ
รายละเอียดลูกหนี้เช่าซื้อ



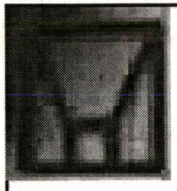
วันที่ 02 พ.ค. 2000

ชื่อ นายสุนทร เจริญมาดี

ที่อยู่ 12 หมู่1 ต.นาเจ้า อ.เมือง จ.อยุธยา เบอร์โทรศัพท์ 035-255452

| เลขที่สัญญา | โครงการ | เลขที่บ้าน | วันที่ทำสัญญา | จำนวนเงินชำระล่วงหน้า | จำนวนงวดเงินล่วงหน้า | จำนวนเงินค่าเช่าซื้อ | จำนวนงวดค่าเช่าซื้อ | วันที่จ่ายเช่าซื้องวดแรก |
|---|---|---|---|---|---|---|---|---|
| 00000001 | เคหะชุมชนบางเขน | 00010001 | 01 พ.ย. 1999 | 30000 | 2 | 5000 | 22 | 10 ธ.ค. 1999 |
| 00000002 | เคหะชุมชนลาดกระบัง | 00030003 | 01 ธ.ค. 1999 | 20000 | 4 | 5000 | 20 | 10 ม.ค. 2000 |

รูปที่ 4.2 แสดงรายงานเสนอแนะผู้ใช้



**การเคหะแห่งชาติ** หน้า 1

**การรับชำระเงินค่าเช่าซื้อประจำวันที่** 10 พ.ค. 2000

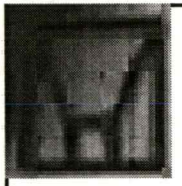
วันที่ 02 พ.ค. 2000

ชื่อโครงการ เคหะชุมชนบางเขน ลักษณะอาคาร ทาวน์เฮ้าส์

| เลขที่สัญญา | เลขที่บ้าน | งวดที่ | รายการชำระ | จำนวนเงิน | รวม |
|---|---|---|---|---|---|
| 00000001 | 00010001 | 3 | ค่าเช่าซื้อ | 5000 | 5,050 |
| | | | ค่าไฟ | 30 | |
| | | | ค่าบริการ | 20 | |
| 00000003 | 00010002 | 1 | ค่าเช่าซื้อ | 5000 | 5,050 |
| | | | ค่าไฟ | 30 | |
| | | | ค่าบริการ | 20 | |
| ยอดรวม | 2 | หน่วย | 10,100 | บาท | |

ลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รูปที่ 3 แสดงรายงานการรับชำระเงินล่วงหน้า



**การเคหะแห่งชาติ** หน้า 1

**การรับชำระเงินล่วงหน้าประจำวันที่** 10 เม.ย. 2000

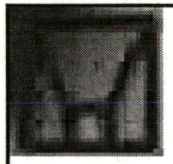
วันที 02 พ.ค. 2000

ชื่อโครงการ เคหะชุมชนบางเขน ลักษณะอาคาร บ้านเดี่ยว

| เลขที่สัญญา | เลขที่บ้าน | ชื่อ-สกุล | | จำนวนงวด | งวดที่ | จำนวนเงินที่รับชำระ | หมายเหตุ |
|---|---|---|---|---|---|---|---|
| 00000005 | 00010003 | นางมานี | รุ่งเรือง | 5 | 1 | 10000 | |

ยอดรวม 1 หน่วย 10,000 บาท

รูปที่ ง.4 ตัวอย่างรายงานที่ผู้ใช้งานสร้างขึ้น



**การเคหะแห่งชาติ** หน้า 1

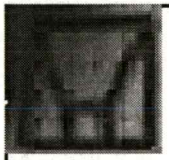
**รายงานลูกหนี้ค้างชำระ**

วันที่ 02 พ.ค. 2000

ชื่อโครงการ เคหะชุมชนบางเขน ลักษณะอาคาร ทาวน์เฮ้าส์

| เลขที่สัญญา | เลขที่บ้าน | ชื่อ-สกุล | งวดที่ | จำนวนเงินที่ค้างชำระ |
|---|---|---|---|---|
| 00000001 | 00010001 | นายสุนทร เจริญมาดี | 3 | 5055 |
| ยอดรวม | 1 หน่วย | | | 5055 บาท |

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามนำไปใช้ในการค้า



**การเคหะแห่งชาติ** หน้า 1

**รายงานลูกหนี้ถึงกำหนดชำระ**

วันที่ 02 พ.ค. 2000

ชื่อโครงการ เคหะชุมชนลาดกระบัง ลักษณะอาคาร ทาวน์เฮ้าส์

| เลขที่สัญญา | เลขที่บ้าน | ชื่อ-สกุล | งวดที่ | ประเภทการจ่ายเงิน | ค่าเช่าซื้อต่องวด | ค่าประกัน |
|---|---|---|---|---|---|---|
| 0000000200030003 | | นายสุนทร เจริญมาดี | 5 | เงินสด | 5000 | 55 |
| 0000000800030001 | | นางมานี รุ่งเรือง | 2 | ธนาคาร | 5000 | 40 |
| ยอดรวม | 2 | หน่วย | | | 10000 | 95 |

ระบบสารสนเทศเพื่อการบริหารงานของการเคหะแห่งชาติ ระบบงานสารสนเทศเพื่อการบริหารงานของการเคหะแห่งชาติ

# ประวัติผู้เขียน

| | | |
|---|---|---|
| ชื่อ | นางสาวนิตยา เปล่งเจริญศิริชัย | |
| วันเดือนปีที่เกิด | 14 กรกฎาคม 2517 | |
| ที่อยู่ | 186 หมู่ 2 ต. ลำพยา อ.เมือง จ.นครปฐม 73000 | |
| ประวัติการศึกษา | ระดับมัธยมศึกษาตอนต้น | โรงเรียนราชินีบูรณะ |
| | ระดับมัธยมศึกษาตอนปลาย | โรงเรียนพระปฐมวิทยาลัย |
| | ระดับอุดมศึกษา | มหาวิทยาลัยศิลปากร |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้