

# ระบบประนอมหนี้การเคหะแห่งชาติ

## National Housing Authority Compromised System

โดย

น.ส. อุษณีย์ เปี่ยมประถม

รหัส 41067028



\*H001608\*

อาจารย์ที่ปรึกษา

ผศ. บรรจง ปิยะธำรง

วัน เดือน ปี 22 S.A. 2549

เลขทะเบียน 01608

เลขเรียกหนังสือ

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชา โครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 2 ปีการศึกษา 2542  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

|                  |                                               |
|------------------|-----------------------------------------------|
| ชื่อหัวข้อ       | ระบบประนอมหนี้การเคหะแห่งชาติ                 |
| นักศึกษา         | น.ส. อุษณีย์ เปี่ยมประถม                      |
| อาจารย์ที่ปรึกษา | ผศ. บรรจง ปิยะธำรง                            |
| ระดับการศึกษา    | วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ |
| แขนงวิชา         | วิทยาการสารสนเทศ                              |
| ปีการศึกษา       | 2542                                          |

### บทคัดย่อ

การพัฒนาระบบสารสนเทศโดยอาศัยหลักวิธีในการพัฒนาระบบเชิงวัตถุ (Object Oriented Systems Development) โดยเริ่มตั้งแต่การวิเคราะห์ ไปจนถึงการออกแบบนั้น ถือเป็นแนวทางในการพัฒนาระบบแนวทางใหม่ ซึ่งในการพัฒนาระบบเชิงวัตถุนี้สามารถรองรับความซับซ้อนต่างๆ ของระบบ และสนับสนุนเทคนิคต่างๆ ในการพัฒนาโปรแกรม

ในโครงการนี้จะเป็นการพัฒนาระบบสารสนเทศ ของระบบงานประนอมหนี้ของการเคหะแห่งชาติ โดยอาศัยระเบียบวิธีของยูเอ็มแอล UML (Unified Modeling Language) เป็นเครื่องมือในการวิเคราะห์ และออกแบบแบบจำลองของระบบ และพัฒนาไปจนถึงการสร้างโปรแกรมของระบบ

|                       |                                               |
|-----------------------|-----------------------------------------------|
| <b>Title</b>          | National Housing Authority Compromised System |
| <b>Student</b>        | Miss Usanee Piampratom                        |
| <b>Advisor</b>        | Asst. Prof. Banjong Piyatamrong               |
| <b>Level of Study</b> | Master of Science in Information Technology   |
| <b>Major</b>          | Information Science                           |
| <b>Academic Year</b>  | 1999                                          |

## ABSTRACT

Developing an information system by using an object-oriented development methodology, which covered from analysis to designing is new approach. Object-oriented development can support complexity of system and software development techniques.

This project is object-oriented system development for the National Housing Authority Compromised System. UML (Unified Modeling Language) is used as a tool for analysis and designing model of system and developing up to implement phase.

## กิตติกรรมประกาศ

โครงการนี้สำเร็จลงได้ด้วยดีเพราะความช่วยเหลือจาก ผศ.บรรจง ปิยะธำรง อาจารย์ที่ปรึกษา ซึ่งได้ให้คำแนะนำและแนวทางในการดำเนินงานต่างๆแก่ผู้จัดทำตลอดมา ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย และนอกจากนี้ขอขอบพระคุณคณาจารย์ท่านอื่น ๆ ที่ได้ประสิทธิประสาทวิชาความรู้แก่ผู้จัดทำ จนสามารถนำความรู้เหล่านั้นมาใช้ในการทำโครงการนี้ให้ลุล่วงไปด้วยดี

และขอกราบขอบพระคุณคุณพ่อคุณแม่ของผู้จัดทำที่ได้อุปการะผู้จัดทำ และเป็นกำลังใจให้กับผู้จัดทำมาโดยตลอด

และท้ายที่สุดนี้ก็ต้องขอขอบคุณเพื่อน ๆ ที่ให้คำปรึกษา ข้อมูล รวมไปถึงการให้กำลังใจแก่ผู้จัดทำ

ผู้จัดทำ  
นางสาว อุษณีย์ เปี่ยมประดม



# สารบัญ

หน้า

|                                                  |     |
|--------------------------------------------------|-----|
| บทคัดย่อ.....                                    | I   |
| ABSTRACT.....                                    | II  |
| กิตติกรรมประกาศ.....                             | III |
| สารบัญ.....                                      | IV  |
| สารบัญภาพ.....                                   | X   |
| สารบัญตาราง.....                                 | XVI |
| บทที่ 1.....                                     | 1   |
| บทนำ.....                                        | 1   |
| 1.1 คำอธิบายระบบ.....                            | 1   |
| 1.1.1 วัตถุประสงค์.....                          | 2   |
| 1.1.2 แผนการดำเนินงาน.....                       | 2   |
| 1.1.3 ประโยชน์ที่คาดว่าจะได้รับ.....             | 2   |
| 1.1.4 ขอบเขตของระบบ.....                         | 2   |
| 1.2 โครงสร้างเนื้อหาของเอกสาร.....               | 3   |
| 1.2.1 บทที่ 1 บทนำ.....                          | 3   |
| 1.2.2 บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....            | 3   |
| 1.2.3 บทที่ 3 การวิเคราะห์ระบบ.....              | 3   |
| 1.2.4 บทที่ 4 การออกแบบระบบ.....                 | 3   |
| 1.2.5 บทที่ 5 สรุป.....                          | 3   |
| 1.2.6 ภาคผนวก.....                               | 3   |
| บทที่ 2.....                                     | 4   |
| ทฤษฎีที่เกี่ยวข้อง.....                          | 4   |
| 2.1 UML (Unified Modeling Language).....         | 4   |
| 2.1.1 การออกแบบเชิงสแตติก (Static Modeling)..... | 5   |

2.1.1.1 แผนผังยูสเคส (Use-Case Diagram).....5

2.1.1.2 แผนผังคลาส (Class Diagram).....7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

หน้า

|                                                               |    |
|---------------------------------------------------------------|----|
| 2.1.1.3 แผนผังออบเจ็กต์ (Object Diagram).....                 | 12 |
| 2.1.2 การออกแบบเชิงไดนามิก (Dynamic Model) .....              | 12 |
| 2.1.2.1 แผนผังซีควเอนส์ (Sequence Diagram).....               | 12 |
| 2.1.2.2 แผนผังคอลเลบอเรชัน (Collaboration Diagram).....       | 13 |
| 2.1.2.3 แผนผังสเตท (State Diagram).....                       | 14 |
| 2.1.2.4 แผนผังกิจกรรม (Activity Diagram) .....                | 14 |
| 2.1.3 โครงสร้างภายนอก(Physical Architecture) .....            | 15 |
| 2.1.3.1 แผนผังองค์ประกอบ (Component Diagram).....             | 15 |
| 2.1.3.2 แผนผังดีพลอยเมนต์ (Deployment Diagram).....           | 16 |
| 2.2 การวิเคราะห์เชิงวัตถุ.....                                | 17 |
| 2.2.1 ออกแบบ Activity Diagram .....                           | 17 |
| 2.2.2 หา Actor ที่เกี่ยวข้องกับระบบ.....                      | 17 |
| 2.2.3 ออกแบบ Use case Diagram .....                           | 17 |
| 2.2.4 ออกแบบ Interaction Diagram .....                        | 18 |
| 2.2.5 การสร้าง Class Diagram.....                             | 18 |
| 2.2.5.1 การกำหนดความสัมพันธ์ระหว่าง คลาส .....                | 19 |
| 2.2.5.2 การกำหนด Attribute.....                               | 21 |
| 2.2.5.3 การกำหนด Method.....                                  | 21 |
| 2.2.6 การทำซ้ำ (Iterate) และการแก้ไขให้ดีขึ้น (Refine).....   | 21 |
| 2.3 การออกแบบเชิงวัตถุ .....                                  | 21 |
| 2.3.1 Object-Oriented Design Axioms .....                     | 22 |
| 2.3.2 การออกแบบ คลาส .....                                    | 23 |
| 2.3.3 การออกแบบ Access Layer.....                             | 24 |
| 2.3.4 การออกแบบ View Layer .....                              | 24 |
| 2.3.5 การทำซ้ำและแก้ไขปรับปรุงให้ดีขึ้นการออกแบบทั้งหมด ..... | 25 |

## สารบัญ(ต่อ)

หน้า

|                                                                                                      |    |
|------------------------------------------------------------------------------------------------------|----|
| การวิเคราะห์ระบบ .....                                                                               | 26 |
| 3.1 คำอธิบายระบบ .....                                                                               | 26 |
| 3.1.1 ความต้องการของระบบ .....                                                                       | 26 |
| 3.2 Activity Diagram .....                                                                           | 27 |
| 3.3 Use-Case Diagram .....                                                                           | 28 |
| 3.3.1 Actor .....                                                                                    | 28 |
| 3.3.2 Use Case .....                                                                                 | 28 |
| 3.3.3 Use case Diagram .....                                                                         | 28 |
| 3.4 Class Diagram.....                                                                               | 30 |
| 3.5 Interaction Diagram .....                                                                        | 30 |
| 3.5.1 Sequence Diagram ของ Search Contract's Details Use Case<br>(ซึ่งใช้เลขที่สัญญาในการค้นหา)..... | 31 |
| 3.5.2 Sequence Diagram ของ Search Contract's Details Use Case<br>(ซึ่งใช้ชื่อลูกค้าในการค้นหา).....  | 32 |
| 3.5.3 Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้เลขที่สัญญาในการค้นหา).....   | 33 |
| 3.5.4 Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้ชื่อลูกค้า ในการค้นหา).....   | 34 |
| 3.5.5 Sequence Diagram ของ Update Old Contract Use Case .....                                        | 35 |
| 3.5.6 Sequence Diagram ของ Create New Contract Use Case.....                                         | 36 |
| 3.5.7 Sequence Diagram ของ Create Compromised Contract Use Case.....                                 | 37 |
| 3.5.8 Sequence Diagram ของ Create Contract 25 years Use Case.....                                    | 38 |
| 3.5.9 Sequence Diagram ของ Create Report Use Case .....                                              | 39 |
| บทที่ 4 .....                                                                                        | 40 |
| การออกแบบระบบ.....                                                                                   | 40 |

## สารบัญ(ต่อ)

หน้า

|                                                                |    |
|----------------------------------------------------------------|----|
| 4.1.1 Attribute ของแต่ละ คลาส.....                             | 41 |
| 4.1.2 Method ของแต่ละ คลาส.....                                | 43 |
| 4.1.3 การออกแบบความสัมพันธ์ .....                              | 46 |
| 4.1.4 การออกแบบ Attribute.....                                 | 46 |
| 4.1.5 การออกแบบ Method.....                                    | 49 |
| 4.2 Access Layer.....                                          | 49 |
| 4.3 View Layer.....                                            | 51 |
| 4.4 การสร้างระบบฐานข้อมูล.....                                 | 53 |
| 4.4.1 การติดตั้ง DBMS.....                                     | 53 |
| 4.4.2 การสร้าง Table ลงใน Database เพื่อเก็บข้อมูลของระบบ..... | 54 |
| 4.4.2.1 การกำหนด Model สำหรับการสร้าง Table.....               | 54 |
| 4.5 การออกแบบยูสเซอร์อินเตอร์เฟส .....                         | 64 |
| 4.5.1 Class MainUI.....                                        | 64 |
| 4.5.1.1 เมนูค้นหาข้อมูล.....                                   | 65 |
| 4.5.1.2 เมนูจัดทำสัญญา.....                                    | 65 |
| 4.5.1.3 เมนูจัดทำรายงาน.....                                   | 65 |
| 4.5.2 Class Search UI.....                                     | 66 |
| 4.5.2.1 ค้นหาข้อมูลรายละเอียดสัญญา.....                        | 66 |
| 4.5.2.2 แสดงหน้าจอค้นหารายละเอียดขอยกค่างชำระ.....             | 67 |
| 4.5.3 Class Create Contract UI .....                           | 67 |
| 4.5.3.1 แสดงหน้าจอจัดทำสัญญาใหม่ .....                         | 67 |
| 4.5.3.2 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ 10 ปี .....             | 68 |
| 4.5.3.3 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี .....               | 69 |
| บทที่ 5 .....                                                  | 71 |
| การสร้างระบบ .....                                             | 71 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

### 5.1 การพัฒนาบนฝั่ง Server.....

71

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

หน้า

|                                                                                                      |    |
|------------------------------------------------------------------------------------------------------|----|
| 5.2 การพัฒนาบนฝั่ง Client .....                                                                      | 75 |
| 5.2.1 การสร้างส่วนรับข้อมูลและแสดงผล (Form) .....                                                    | 75 |
| 5.2.2 การสร้างส่วนรายงาน (Report) .....                                                              | 77 |
| 5.2.3 การรวมส่วนต่างๆ ที่พัฒนาเข้าด้วยกัน (Integration).....                                         | 77 |
| บทที่ 6 .....                                                                                        | 78 |
| คู่มือระบบ.....                                                                                      | 78 |
| 6.1 ความต้องการของระบบ (System Requirement) .....                                                    | 78 |
| 6.2 แบบจำลองของกระบวนการทางธุรกิจ.....                                                               | 79 |
| 6.3 Use-Case Diagram .....                                                                           | 80 |
| 6.3.1 Actor .....                                                                                    | 80 |
| 6.3.2 Use Case .....                                                                                 | 80 |
| 6.3.3 Use case Diagram .....                                                                         | 80 |
| 6.4 Interaction Diagram .....                                                                        | 82 |
| 6.4.1 Sequence Diagram ของ Search Contract's Details Use Case<br>(ซึ่งใช้เลขที่สัญญาในการค้นหา)..... | 82 |
| 6.4.2 Sequence Diagram ของ Search Contract's Details Use Case<br>(ซึ่งใช้ชื่อลูกค้าในการค้นหา).....  | 83 |
| 6.4.3 Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้เลขที่สัญญาในการค้นหา).....   | 84 |
| 6.4.4 Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้ชื่อลูกค้า ในการค้นหา).....   | 85 |
| 6.4.5 Sequence Diagram ของ Update Old Contract Use Case .....                                        | 86 |
| 6.4.6 Sequence Diagram ของ Create New Contract Use Case.....                                         | 87 |
| 6.4.7 Sequence Diagram ของ Create Compromised Contract Use Case.....                                 | 88 |
| 6.4.8 Sequence Diagram ของ Create Contract 25 years Use Case.....                                    | 89 |
| 6.4.9 Sequence Diagram ของ Create Report Use Case .....                                              | 90 |



## สารบัญ(ต่อ)

หน้า

|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| 6.5 Class Diagram.....                                                                 | 90  |
| 6.5.1 Class.....                                                                       | 90  |
| 6.5.2 Attribute ของแต่ละ คลาส.....                                                     | 91  |
| 6.5.3 Method.....                                                                      | 94  |
| 6.5.4 Class Diagram in Application Logic .....                                         | 108 |
| 6.5.5 Class Diagram in Access Layer .....                                              | 109 |
| 6.5.5.1 Schema.....                                                                    | 109 |
| 6.5.5.2 Class Diagram in Access Layer.....                                             | 110 |
| 6.5.5.3 ความสัมพันธ์ ระหว่าง Class Diagram in Access Layer และ Application Logic ..... | 111 |
| 6.5.6 Class Diagram in View Layer.....                                                 | 112 |
| 6.5.6.1 View Layer .....                                                               | 112 |
| 6.5.6.2 ความสัมพันธ์ ระหว่าง Class Diagram in Access Layer และ Application Logic ..... | 113 |
| บทที่ 7 .....                                                                          | 114 |
| คู่มือผู้ใช้งาน .....                                                                  | 114 |
| 7.1 วิธีการเข้าสู่โปรแกรม.....                                                         | 114 |
| 7.2 เมนูค้นหาข้อมูล.....                                                               | 115 |
| 7.2.1 ค้นหารายละเอียดสัญญา.....                                                        | 116 |
| 7.2.2 ค้นหารายละเอียดค่างชำระ .....                                                    | 120 |
| 7.3 เมนูจัดทำสัญญา .....                                                               | 121 |
| 7.3.1 จัดทำสัญญาใหม่ .....                                                             | 121 |
| 7.3.2 จัดทำสัญญาประนอมหนี้ .....                                                       | 126 |
| 7.3.3 จัดทำสัญญาเช่าซื้อ 25 ปี.....                                                    | 129 |
| 7.4 เมนูจัดทำรายงาน.....                                                               | 133 |



## สารบัญ(ต่อ)

|                                                 | หน้า |
|-------------------------------------------------|------|
| 7.3.2 รายงานบลูกหนี้เข้าซื้อค้างชำระ .....      | 134  |
| 7.3.3 รายงานอายุประนอมหนี้ลูกหนี้ค้างชำระ.....  | 136  |
| 7.5 เมนูออกจากระบบ .....                        | 136  |
| 7.6 เมนูหน้าต่าง.....                           | 136  |
| บทที่ 8 .....                                   | 138  |
| สรุป .....                                      | 138  |
| 8.1 ปัญหาต่างๆ ที่เกิดในการพัฒนาระบบ .....      | 139  |
| 8.2 เปรียบเทียบกับการพัฒนาระบบในแนวทางเดิม..... | 139  |
| 8.3 แนวทางในการพัฒนาระบบ .....                  | 139  |
| บรรณานุกรม .....                                | 141  |
| ภาคผนวก .....                                   | 142  |
| ก. ตัวอย่างรายงานของระบบ.....                   | 142  |
| ข. ตัวอย่าง Store Procedure.....                | 146  |
| ประวัติผู้เขียน .....                           | 157  |

## สารบัญภาพ

หน้า

### รูปที่

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| 1. รูปที่ 2.1 แสดงตัวอย่าง Use-Case Diagram.....                                     | 5  |
| 2. รูปที่ 2.2 แสดงความสัมพันธ์แบบ Extends Relationships .....                        | 6  |
| 3. รูปที่ 2.3 แสดงความสัมพันธ์แบบ Uses Relationships.....                            | 7  |
| 4. รูปที่ 2.4 แสดงสัญลักษณ์แทน คลาส .....                                            | 7  |
| 5. รูปที่ 2.5 แสดงความสัมพันธ์แบบ Qualified Association .....                        | 8  |
| 6. รูปที่ 2.6 แสดงความสัมพันธ์แบบ OR-Association .....                               | 9  |
| 7. รูปที่ 2.7 แสดงความสัมพันธ์แบบ Ternary Association.....                           | 9  |
| 8. รูปที่ 2.8 แสดงความสัมพันธ์แบบ Shared Aggregation.....                            | 10 |
| 9. รูปที่ 2.9 แสดงความสัมพันธ์แบบ Composition Aggregation .....                      | 10 |
| 10. รูปที่ 2.10 แสดงตัวอย่างความสัมพันธ์แบบ Generalization ภายใน Class Diagram ..... | 11 |
| 11. รูปที่ 2.11 แสดงตัวอย่างความสัมพันธ์แบบ Dependency ภายใน Class Diagram.....      | 11 |
| 12. รูปที่ 2.12 แสดงตัวอย่างความสัมพันธ์แบบ Refinement ภายใน Class Diagram .....     | 11 |
| 13. รูปที่ 2.13 แสดงการเปรียบเทียบระหว่าง Class Diagram กับ Object Diagram.....      | 12 |
| 14. รูปที่ 2.14 แสดงตัวอย่าง Sequence Diagram.....                                   | 13 |
| 15. รูปที่ 2.15 แสดงตัวอย่าง Collaboration Diagram .....                             | 13 |
| 16. รูปที่ 2.16 แสดงตัวอย่าง State Diagram .....                                     | 14 |
| 17. รูปที่ 2.17 แสดงตัวอย่าง Activity Diagram.....                                   | 15 |
| 18. รูปที่ 2.18 แสดงตัวอย่าง Component Diagram .....                                 | 16 |
| 19. รูปที่ 2.19 แสดงตัวอย่าง Deployment Diagram .....                                | 16 |
| 20. รูปที่ 2. 20 แสดงความสัมพันธ์ระหว่าง Corollary กับ Axiom .....                   | 23 |
| 21. รูปที่ 3. 1 แสดง Activity Diagram ของระบบประนอนมนี้ .....                        | 27 |
| 22. รูปที่ 3. 2 แสดง Use Case Diagram ของระบบประนอนมนี้ .....                        | 29 |
| 23. รูปที่ 3. 3 แสดง Class Diagram ของ ระบบประนอนมนี้ของการทะเลาะแห่งชาติ .....      | 30 |
| 24. รูปที่ 3. 4 แสดง Sequence Diagram ของ Search Contract's Details Use Case         |    |

## สารบัญภาพ(ต่อ)

หน้า

รูปที่

|                                                                                                                    |    |
|--------------------------------------------------------------------------------------------------------------------|----|
| 25. รูปที่ 3. 5 แสดง Sequence Diagram ของ Search Contract's Details Use Case<br>(ซึ่งใช้ชื่อลูกค้าในการค้นหา)..... | 32 |
| 26. รูปที่ 3. 6 แสดง Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้เลขที่สัญญาในการค้นหา).....  | 33 |
| 27. รูปที่ 3. 7 แสดง Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้ชื่อลูกค้าในการค้นหา).....   | 34 |
| 28. รูปที่ 3. 8 แสดง Sequence Diagram ของ Update Old Contract Use Case .....                                       | 35 |
| 29. รูปที่ 3. 9 แสดง Sequence Diagram ของ Create New Contract Use Case.....                                        | 36 |
| 30. รูปที่ 3. 10 แสดง Sequence Diagram ของ Create Compromised Contract Use Case.....                               | 37 |
| 31. รูปที่ 3. 11 แสดง Sequence Diagram ของ Create Contract 25 Years Use Case .....                                 | 38 |
| 32. รูปที่ 3. 12 แสดง Sequence Diagram ของ Create Report Use Case .....                                            | 39 |
| 33. รูปที่ 4. 1 แสดง Class Diagram ของระบบประนอมหนี้ .....                                                         | 45 |
| 34. รูปที่ 4. 2 แสดงความสัมพันธ์ระหว่าง Business Layer และ Access Layer.....                                       | 50 |
| 35. รูปที่ 4. 3 แสดงคลาสใน Access Layer .....                                                                      | 51 |
| 36. รูปที่ 4. 4 แสดงคลาสใน View Layer .....                                                                        | 52 |
| 37. รูปที่ 4. 5 แสดงความสัมพันธ์ระหว่าง Business Layer และ View Layer.....                                         | 53 |
| 38. รูปที่ 4. 6 แสดงการ Import Oracle8 Data Type .....                                                             | 55 |
| 39. รูปที่ 4. 7 แสดงการสร้าง Component พร้อมกับการกำหนดชื่อ .....                                                  | 55 |
| 40. รูปที่ 4. 8 แสดงการกำหนดคุณสมบัติของ Component (Component Specification).....                                  | 56 |
| 41. รูปที่ 4. 9 แสดงการ Assign Class ที่ Map.....                                                                  | 56 |
| 42. รูปที่ 4. 10 แสดงการกำหนด Stereotype ของ Class เป็น ObjectType .....                                           | 57 |
| 43. รูปที่ 4. 11 แสดงการกำหนดชนิดข้อมูลของ Attribute.....                                                          | 57 |
| 44. รูปที่ 4. 12 แสดงการสร้างฐานข้อมูลโดยใช้ WiZard .....                                                          | 58 |
| 45. รูปที่ 4. 13 แสดง Wizard เลือกฐานข้อมูลเป็น Relational Database.....                                           | 58 |
| 46. รูปที่ 4. 14 แสดง Wizard ในการ Map (Data Type Description).....                                                | 59 |

## สารบัญภาพ(ต่อ)

หน้า

### รูปที่

|                                                                                                    |    |
|----------------------------------------------------------------------------------------------------|----|
| 47. รูปที่ 4. 15 แสดง Wizard ในการ Map (การกำหนด Column).....                                      | 59 |
| 48. รูปที่ 4. 16 แสดง Wizard ของการ Map (การกำหนด Indices / Primary Key) .....                     | 60 |
| 49. รูปที่ 4. 17 แสดง Wizard ของการ Map (การกำหนด Foreign Key).....                                | 60 |
| 50. รูปที่ 4. 18 แสดง Wizard ของการ Map (การเรียงลำดับ Attribute) .....                            | 61 |
| 51. รูปที่ 4. 19 แสดง Schema Diagramของระบบประนอมหนี้ .....                                        | 62 |
| 52. รูปที่ 4. 20 แสดง Entity Relatioship ของระบบประนอมหนี้ .....                                   | 63 |
| 53. รูปที่ 4. 21 แสดงหน้าจอเมนูหลังของระบบการประนอมหนี้ .....                                      | 64 |
| 54. รูปที่ 4. 22 แสดงเมนูย่อยสำหรับการค้นหาข้อมูล .....                                            | 65 |
| 55. รูปที่ 4. 23 แสดงเมนูย่อยสำหรับการจัดทำสัญญา.....                                              | 65 |
| 56. รูปที่ 4. 24 แสดงเมนูย่อยสำหรับการจัดทำรายงาน .....                                            | 66 |
| 57. รูปที่ 4. 25 แสดงหน้าจอค้นหารายละเอียดสัญญา.....                                               | 66 |
| 58. รูปที่ 4. 26 แสดงหน้าจอค้นหารายละเอียดขอยอดค้างชำระ .....                                      | 67 |
| 59. รูปที่ 4. 27 แสดงหน้าจอจัดทำสัญญาใหม่.....                                                     | 68 |
| 60. รูปที่ 4. 28 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ 10 ปี.....                                         | 69 |
| 61. รูปที่ 4. 29 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี.....                                           | 70 |
| 62. รูปที่ 5. 1 แสดงตัวอย่าง ส่วนของ Parameter หรือ ค่าตัวแปรต่างๆ ใน Program Specification        | 72 |
| 63. รูปที่ 5. 2 แสดงตัวอย่าง ส่วนของ Function ต่างๆ ใน Program Specification .....                 | 73 |
| 64. รูปที่ 5. 3 แสดงตัวอย่าง ส่วนของ Program Body .....                                            | 73 |
| 65. รูปที่ 5. 4 แสดงตัวอย่าง ส่วนของ Program Body .....                                            | 74 |
| 66. รูปที่ 5. 5 6 แสดงตัวอย่าง ส่วนของ Program Body .....                                          | 74 |
| 67. รูปที่ 5. 7 แสดงตัวอย่างการเรียกใช้ Store Procedure ในฝั่ง Server จาก Form ในฝั่ง Client ..... | 76 |
| 68. รูปที่ 5. 8 แสดงตัวอย่างการเรียกใช้ Store Procedure ในฝั่ง Server จาก Form ในฝั่ง Client ..... | 76 |
| 69. รูปที่ 6. 1 แสดง Activity Diagram ของระบบประนอมหนี้ .....                                      | 79 |
| 70. รูปที่ 6. 2 แสดง Use Case Diagram ของระบบประนอมหนี้ .....                                      | 81 |
| 71. รูปที่ 6. 3 แสดง Sequence Diagram ของ Search Contract's Details Use Case                       |    |



## สารบัญภาพ(ต่อ)

หน้า

รูปที่

|                                                                                                                    |     |
|--------------------------------------------------------------------------------------------------------------------|-----|
| (ซึ่งใช้เลขที่ สัญญาในการค้นหา) .....                                                                              | 82  |
| 72. รูปที่ 6. 4 แสดง Sequence Diagram ของ Search Contract's Details Use Case<br>(ซึ่งใช้ชื่อลูกค้าในการค้นหา)..... | 83  |
| 73. รูปที่ 6. 5 แสดง Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้เลขที่สัญญาในการค้นหา) ..... | 84  |
| 74. รูปที่ 6. 6 แสดง Sequence Diagram ของ Search Remaining Amount Use Case<br>(ซึ่งใช้ชื่อลูกค้าในการค้นหา).....   | 85  |
| 75. รูปที่ 6. 7 แสดง Sequence Diagram ของ Update Old Contract Use Case .....                                       | 86  |
| 76. รูปที่ 6. 8 แสดง Sequence Diagram ของ Create New Contract Use Case .....                                       | 87  |
| 77. รูปที่ 6. 9 แสดง Sequence Diagram ของ Create Compromised Contract Use Case.....                                | 88  |
| 78. รูปที่ 6. 10 แสดง Sequence Diagram ของ Create Contract 25 Years Use Case .....                                 | 89  |
| 79. รูปที่ 6. 11 แสดง Sequence Diagram ของ Create Report Use Case .....                                            | 90  |
| 80. รูปที่ 6. 12 แสดง Algorithm ของ getAmountbuy Method.....                                                       | 94  |
| 81. รูปที่ 6. 13 แสดง Algorithm ของ getInterestRate Method.....                                                    | 95  |
| 82. รูปที่ 6. 14 แสดง Algorithm ของ getInterestTotal Method.....                                                   | 96  |
| 83. รูปที่ 6. 15 แสดง Algorithm ของ verifyContractID Method .....                                                  | 97  |
| 84. รูปที่ 6. 16 แสดง Algorithm ของ getContractDetil Method .....                                                  | 98  |
| 85. รูปที่ 6. 17 แสดง Algorithm ของ calCreditTerm Method.....                                                      | 99  |
| 86. รูปที่ 6. 18 แสดง Algorithm ของ calCreditAmount Method .....                                                   | 100 |
| 87. รูปที่ 6. 19 แสดง Algorithm ของ calRemainAmount Method.....                                                    | 101 |
| 88. รูปที่ 6. 20 แสดง Algorithm ของ createContract Method.....                                                     | 102 |
| 89. รูปที่ 6. 21 แสดง Algorithm ของ verifyCustomerName Method .....                                                | 104 |
| 90. รูปที่ 6. 22 แสดง Algorithm ของ getCustomerDetail Method.....                                                  | 104 |
| 91. รูปที่ 6. 23 แสดง Algorithm ของ getMaxTermI Method.....                                                        | 105 |
| 92. รูปที่ 6. 24 แสดง Algorithm ของ getPaymentDate Method .....                                                    | 106 |



## สารบัญภาพ(ต่อ)

หน้า

### รูปที่

|                                                                                                                 |     |
|-----------------------------------------------------------------------------------------------------------------|-----|
| 93. รูปที่ 6. 25 แสดง Algorithm ของ getProjectDetail Method .....                                               | 107 |
| 94. รูปที่ 6. 26 แสดง Algorithm ของ getModelDetail Method.....                                                  | 107 |
| 95. รูปที่ 6. 27 แสดง Class Diagram ใน Application Logic .....                                                  | 108 |
| 96. รูปที่ 6. 28 แสดง Schema ของระบบฐานข้อมูล โดยใช้ Class Diagram.....                                         | 109 |
| 97. รูปที่ 6. 29 แสดง Class Diagram ในส่วนของ Access Layer .....                                                | 110 |
| 98. รูปที่ 6. 30 แสดงความสัมพันธ์ระหว่าง Class Diagram in Access Layer และ Application Logic .....              | 111 |
| 99. รูปที่ 6. 31 แสดง Class Diagram ในส่วนของ View Layer.....                                                   | 112 |
| 100. รูปที่ 6. 32 แสดงความสัมพันธ์ระหว่าง Class Diagram in View Layer และ Application Logic .....               | 113 |
| 101. รูปที่ 7. 2 แสดงการ Login เข้าสู่ระบบ .....                                                                | 114 |
| 102. รูปที่ 7. 3 แสดงหน้าจอเมนูหลักของระบบ .....                                                                | 115 |
| 103. รูปที่ 7. 4 แสดงเมนูย่อยในเมนูค้นหาข้อมูล .....                                                            | 115 |
| 104. รูปที่ 7. 5 แสดงหน้าจอค้นหารายละเอียดสัญญา.....                                                            | 116 |
| 105. รูปที่ 7. 6 แสดงการกรอกเลขที่สัญญาเพื่อใช้ในการค้นหาข้อมูล.....                                            | 117 |
| 106. รูปที่ 7. 7 แสดงหน้าจอแสดงรายละเอียดของสัญญา หลังจาก กดปุ่ม “ค้นหา” .....                                  | 117 |
| 107. รูปที่ 7. 8 แสดงการกรอก ชื่อ-สกุล ลูกค้า เพื่อใช้ในการค้นหาข้อมูล.....                                     | 118 |
| 108.รูปที่ 7. 9 แสดงหน้าจอแสดงรายการสัญญาของลูกค้าหลังจากทำการค้นหา ชื่อ-สกุล ลูกค้าพบ .....                    | 119 |
| 109. รูปที่ 7. 10 แสดงหน้าจอแสดงรายละเอียดของสัญญา หลังจาก กดปุ่ม “เลือก”.....                                  | 119 |
| 110. รูปที่ 7. 11 แสดงหน้าจอบอกข้อผิดพลาด<br>(ในกรณีที่ไม่มีพบเลขที่สัญญา หรือ ชื่อ-สกุลลูกค้า ที่ต้องการ)..... | 120 |
| 111. รูปที่ 7. 12 แสดงหน้าจอค้นหารายละเอียดขอยอดค้างชำระ .....                                                  | 120 |
| 112. รูปที่ 7. 13 แสดงเมนูย่อยของเมนูจัดทำสัญญา.....                                                            | 121 |
| 113. รูปที่ 7. 14 แสดงหน้าจอจัดทำสัญญาใหม่.....                                                                 | 122 |

## สารบัญญภาพ(ต่อ)

หน้า

รูปที่

|                                                                                                                                 |     |
|---------------------------------------------------------------------------------------------------------------------------------|-----|
| 114. รูปที่ 7. 15 แสดงหน้าจอจัดทำสัญญาใหม่ หลังจาก ค้นหาข้อมูลสัญญาเก่าเรียบร้อยแล้ว<br>(กรณีค้นหาจาก “เลขที่สัญญา”) .....      | 123 |
| 115. รูปที่ 7. 16 แสดงหน้าจอจัดทำสัญญาใหม่ หลังจาก ค้นหาข้อมูลสัญญาเก่าเรียบร้อยแล้ว<br>(กรณีค้นหาจาก “ชื่อ-สกุล ลูกค้า”) ..... | 123 |
| 116. รูปที่ 7. 17 แสดงหน้าจอเตือน ให้กรอกเลขที่สัญญาใหม่ หลังจาก กดปุ่ม “จัดทำสัญญาใหม่”<br>.....                               | 124 |
| 117. รูปที่ 7. 18 แสดงหน้าจอจัดทำสัญญาใหม่<br>(หลังจาก กดปุ่ม “OK” ในหน้าจอเตือน ให้กรอกเลขที่สัญญาใหม่) .....                  | 124 |
| 118. รูปที่ 7. 19 แสดงหน้าจอกรอกเลขที่สัญญาใหม่ .....                                                                           | 125 |
| 119. รูปที่ 7. 20 แสดงหน้าจอเตือนว่า ต้องการบันทึกใช่หรือไม่.....                                                               | 125 |
| 120. รูปที่ 7. 21 แสดงหน้าจอเตือนว่า การบันทึกครั้งนี้ สมบูรณ์หรือไม่.....                                                      | 126 |
| 121. รูปที่ 7. 22 แสดงหน้าจอจัดทำสัญญาประนอมหนี้.....                                                                           | 126 |
| 122. รูปที่ 7. 23 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ หลังจากค้นหารายละเอียดสัญญาแล้ว<br>(ค้นหาจาก “เลขที่สัญญา”) .....              | 127 |
| 123. รูปที่ 7. 24 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ หลังจากค้นหารายละเอียดสัญญาแล้ว<br>(ค้นหาจาก “ชื่อ-สกุล ลูกค้า”).....          | 128 |
| 124. รูปที่ 7. 25 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ หลังจากกดปุ่ม “คำนวณเงินต้องงวด” .....                                         | 129 |
| 125. รูปที่ 7. 26 แสดงหน้าจอเตือน ให้กรอกอัตราดอกเบี้ยสัญญาประนอมหนี้และจำนวนงวด....                                            | 129 |
| 126. รูปที่ 7. 27 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี .....                                                                      | 130 |
| 127. รูปที่ 7. 28 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี หลังจาก ค้นหารายละเอียดสัญญาแล้ว<br>(ค้นหาจาก “เลขที่สัญญา”).....          | 131 |
| 128. รูปที่ 7. 29 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี หลังจาก ค้นหารายละเอียดสัญญาแล้ว<br>(ค้นหาจาก “ชื่อ-สกุล ลูกค้า”).....     | 131 |
| 129. รูปที่ 7. 30 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี หลังจากกดปุ่ม “คำนวณเงินต้องงวด” .....                                     | 132 |
| 130. รูปที่ 7. 31 แสดงเมนูย่อยในเมนูจัดทำรายงาน.....                                                                            | 133 |

## สารบัญภาพ(ต่อ)

หน้า

รูปที่

|                                                                                                                   |     |
|-------------------------------------------------------------------------------------------------------------------|-----|
| 131. รูปที่ 7. 32 แสดงหน้าจอแสดงความก้าวหน้าของรายงาน .....                                                       | 133 |
| 132. รูปที่ 7. 33 แสดงตัวอย่างรายงานงบลูกหนี้ค้างชำระ .....                                                       | 134 |
| 133. รูปที่ 7. 34 แสดงหน้าจอรับค่าตัวแปร รหัสโครงการ และรหัสโมเดล<br>(เพื่อจัดทำรายงาน งบลูกหนี้ค้างชำระ).....    | 134 |
| 134. รูปที่ 7. 35 แสดงหน้าจอที่กรอกค่า รหัสโครงการ และ รหัสโมเดล<br>(เพื่อจัดทำรายงาน งบลูกหนี้ค้างชำระแล้ว)..... | 135 |
| 135. รูปที่ 7. 37 แสดงตัวอย่างรายงานงบลูกหนี้ค้างชำระ .....                                                       | 135 |
| 136. รูปที่ 7. 38 แสดงตัวอย่างรายงานอายุประนอมหนี้ลูกหนี้ค้างชำระ .....                                           | 136 |
| 137. รูปที่ 7. 39 แสดงเมนูย่อยในเมนูหน้าต่าง .....                                                                | 137 |

# สารบัญตาราง

หน้า

ตารางที่

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| 1. ตารางที่ 1. 1 แสดงเงื่อนไข และหลักเกณฑ์ในระบบสัญญาประนอมหนี้ .....                   | 1  |
| 2. ตารางที่ 2. 1 แสดงรายละเอียดเงื่อนไข และหลักเกณฑ์ต่างๆในการประนอมหนี้.....           | 26 |
| 3. ตารางที่ 4. 1 แสดงคลาสของระบบประนอมหนี้.....                                         | 40 |
| 4. ตารางที่ 4. 2 แสดงชื่อพร้อมคำอธิบายแอทริบิวต์ ที่อยู่ในคลาสประนอมหนี้.....           | 41 |
| 5. ตารางที่ 4. 3 แสดงชื่อพร้อมคำอธิบายเมทอด ที่อยู่ในคลาสของระบบประนอมหนี้.....         | 43 |
| 6. ตารางที่ 4. 4 แสดงชื่อพร้อมชนิดข้อมูลแอทริบิวต์ ที่อยู่ในคลาสของระบบประนอมหนี้ ..... | 46 |
| 7. ตารางที่ 6. 1 แสดง ชื่อคลาส และคำอธิบาย ของแต่ละคลาส ในระบบประนอมหนี้.....           | 90 |
| 8. ตารางที่ 6. 2 แสดงชื่อพร้อมคำอธิบายแอทริบิวต์ ที่อยู่ในคลาสประนอมหนี้.....           | 91 |

# บทที่ 1

## บทนำ

### 1.1 คำอธิบายระบบ

ระบบประนอมหนี้เป็นส่วนหนึ่งของระบบบัญชีลูกหนี้ของการเคหะแห่งชาติ โดยในการดำเนินงานของการเคหะในส่วนของระบบบัญชีลูกหนี้ นั้น เมื่อลูกหนี้มีการค้างชำระค่าเช่าซื้อ ทาง การเคหะแห่งชาติจะมีการดำเนินงานในส่วนของการประนอมหนี้ให้กับลูกหนี้ที่ค้างชำระเหล่านั้น ซึ่งมีขั้นตอนในการประนอมหนี้ต่างๆ เช่น ทำสัญญาเช่าซื้อใหม่ โดยจะทำการกำหนดค่าเช่าซื้อ ต่องวด และระยะเวลาในการชำระใหม่

ซึ่งการเคหะแห่งชาติ ได้กำหนดหลักเกณฑ์ และวิธีปฏิบัติเกี่ยวกับการประนอมหนี้ ดังนี้

ตารางที่ 1. 1 แสดงเงื่อนไข และหลักเกณฑ์ในระบบสัญญาประนอมหนี้

| หลักเกณฑ์                                                                                                                                                                                                                                                                                                                                  | วิธีปฏิบัติ                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| มีเงื่อนไขดังนี้<br>1.ให้ผู้ค้างชำระ ประนอมหนี้ได้ในระยะเวลา ไม่เกิน 10 ปี และค่าเช่าซื้อส่วนที่เหลือให้ ทำสัญญาเช่าซื้อใหม่ผ่อนชำระไม่เกิน 25 ปี<br>2. กำหนดให้ ในส่วนการผ่อนชำระไม่เกิน 25 ปี อัตราผ่อนชำระ เท่ากับ อัตราผ่อนชำระ ค่าเช่าซื้อเดิมในช่วง 5 ปีแรก หลังจากนั้น ปรับอัตราเช่าซื้อเพิ่มทุกๆ 5 ปี จนครบอายุ สัญญาไม่เกิน 25 ปี | การทำสัญญา<br>ทำสัญญาใหม่ ให้แยกเป็น 2 สัญญา คือ สัญญาประนอมหนี้ไม่เกิน 10 ปี และ สัญญาเช่าซื้อไม่เกิน 25 ปี<br>การคำนวณอัตราผ่อนชำระค่าเช่าซื้อ<br>1. สัญญาประนอมหนี้ไม่เกิน 10 ปี<br>คำนวณอัตราผ่อนชำระ โดยใช้อัตรา ดอกเบี้ยคงที่ 7 %<br>2. สัญญาเช่าซื้อไม่เกิน 25 ปี<br>คำนวณอัตราผ่อนชำระค่าเช่าซื้อให้เท่ากับอัตราค่าเช่าซื้อเดิม ในช่วง 5 ปีแรก หลังจากนั้น ปรับอัตราเช่าซื้อเพิ่มทุกๆ 5 ปี จนครบอายุสัญญาไม่เกิน 25 ปี |



โดยในการพัฒนาระบบประណอหนึ่ของการเคหะแห่งชาติในโครงการนี้ จะใช้การพัฒนา  
ระบบเชิงวัตถุ (Object-Oriented System Development) โดยใช้ระเบียบวิธีของยูเอ็มแอล UML  
(Unified Modeling Language) เป็นเครื่องมือในการวิเคราะห์ และออกแบบแบบจำลอง (Model)  
ของระบบ ในส่วนของฐานข้อมูลจะทำการสร้างฐานข้อมูลเชิงสัมพันธ์ของระบบโดยใช้ระบบจัด  
การฐานข้อมูลอราเคิล (Oracle Database Management System) และพัฒนาส่วนที่ติดต่อกับผู้ใช้  
ด้วย Oracle Developer

#### 1.1.1 วัตถุประสงค์

1. แก้ไขปรับปรุงขั้นตอนการทำงานระบบประណอหนึ่ให้มีความสะดวกและรวดเร็ว
2. จัดการข้อมูลเกี่ยวกับระบบประណอหนึ่ให้มีความถูกต้องทันสมัย

#### 1.1.2 แผนการดำเนินงาน

1. ศึกษาความต้องการของผู้ใช้ระบบ
2. วิเคราะห์และออกแบบระบบโดยสร้างแบบจำลองระบบ
3. พัฒนาโปรแกรมตามแบบจำลองที่ได้ออกแบบไว้

#### 1.1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถอำนวยความสะดวกให้กับผู้ใช้ ใช้งานได้ง่าย และรวดเร็วยิ่งขึ้น
2. ข้อมูลเกี่ยวกับระบบมีความถูกต้องทันสมัย

#### 1.1.4 ขอบเขตของระบบ

ระบบประណอหนึ่นั้นจะสามารถทำงานต่างๆ ได้ดังนี้

- 1.สรุปยอดค้างชำระเดิมของลูกหนึ่ที่จะทำการประណอหนึ่แต่ละราย โดยสรุปว่ายังคง  
เหลือค่าเช่าซื้อที่ต้องชำระทั้งหมดเป็นจำนวนเท่าไร ก่อนทำการประណอหนึ่
2. จัดทำสัญญาใหม่ ทั้งในส่วนสัญญาประណอหนึ่ไม่เกิน 10 ปี และ สัญญาเช่าซื้อไม่  
เกิน 25 ปี
3. คำนวณค่าเช่าซื้อต้องวด โดยใช้อัตราดอกเบี้ยตามหลักเกณฑ์ และวิธีปฏิบัติเกี่ยวกับ  
การประណอหนึ่
4. จัดทำรายงานต่างๆ ซึ่งได้แก่
  1. สรุปรายงานลูกหนึ่ประណอหนึ่
  2. รายงานลูกหนึ่ค้างชำระเปรียบเทียบแยกตามโครงการ

## 1.2 โครงสร้างเนื้อหาของเอกสาร

### 1.2.1 บทที่ 1 บทนำ

กล่าวถึงภาพรวมของโครงการ ซึ่งประกอบไปด้วยวัตถุประสงค์ แผนการดำเนินงาน ผลที่คาดว่าจะได้รับ และขอบเขตของระบบ

### 1.2.2 บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

กล่าวถึงทฤษฎีที่เกี่ยวข้อง ที่นำมาใช้ในการดำเนินงานในโครงการนี้ ซึ่งจะประกอบไปด้วยกระบวนการของการวิเคราะห์และออกแบบเชิงวัตถุ และระเบียบวิธีต่างๆของ UML (Unified Modeling Language)

### 1.2.3 บทที่ 3 การวิเคราะห์ระบบ

กล่าวถึงการวิเคราะห์ระบบประเภทย่อยนี้ ซึ่งเป็นการวิเคราะห์ระบบตามระเบียบของ UML และแสดงแบบจำลองของระบบที่ได้ออกแบบขึ้นมา

### 1.2.4 บทที่ 4 การออกแบบระบบ

กล่าวถึงการออกแบบระบบประเภทย่อยนี้ ซึ่งเป็นการออกแบบระบบตามระเบียบของ UML และแสดงแบบจำลองของระบบที่ได้ออกแบบขึ้นมา

### 1.2.5 บทที่ 5 การสร้างระบบ

กล่าวถึงวิธีการพัฒนาซอฟต์แวร์ของระบบประเภทย่อยนี้ โดยการพัฒนาจะเป็นไปตามสิ่งที่ได้จากการวิเคราะห์และออกแบบ

### 1.2.5 บทที่ 5 สรุป

กล่าวถึงปัญหาต่างๆที่พบ และแสดงการเปรียบเทียบระหว่างการพัฒนาแบบเชิงวัตถุกับการพัฒนาระบบแบบเดิม

### 1.2.6 ภาคผนวก

แสดงถึงรูปแบบรายงานต่างๆ ของระบบประเภทย่อยนี้ของการเคหะแห่งชาติ

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 UML (Unified Modeling Language)

Interaction Diagram (Object-Oriented) ซึ่งการพัฒนาระบบสารสนเทศในปัจจุบันนิยมสร้างโปรแกรมในเชิงวัตถุ ซึ่ง UML เป็นภาษาที่ใช้ในการวิเคราะห์และออกแบบที่ช่วยสนับสนุนการออกแบบในเชิงวัตถุ การวิเคราะห์และออกแบบระบบ โดยใช้ UML นั้นจะเป็นการออกแบบแบบจำลอง (Model) ขึ้นมาเพื่อใช้อธิบายระบบที่ต้องการพัฒนา โดยแบบจำลองใน UML แบ่งได้ทั้งเป็น 2 ชนิด คือ แบบจำลองเชิงสถิต (Static Model) และ แบบจำลองเชิงไดนามิก (Dynamic Model) โดยการออกแบบเชิงสถิตจะเป็นการออกแบบภาพรวมของระบบนั้นๆ เช่น โครงสร้างของคลาส (Class) เป็นอย่างไร โดยจะไม่มีมีการเปลี่ยนแปลงของโครงสร้างนั้นๆ ส่วนการออกแบบเชิงไดนามิก จะเป็นการออกแบบเพื่อแสดงถึงพฤติกรรมของระบบว่ามีการเปลี่ยนแปลงอย่างไร ณ เวลาใดเวลาหนึ่งซึ่งเมื่อเวลาเปลี่ยนไปพฤติกรรมนั้นๆ อาจมีการเปลี่ยนแปลงได้

ใน UML นั้นจะมีแผนผัง (Diagram) ที่ใช้เพื่อแสดงการรายละเอียดของระบบ ทั้งหมด 9 แผนผัง โดยในแผนผังทั้ง 9 แผนผังนี้ จะแบ่งเป็น 2 ส่วนใหญ่ๆ คือ

- การออกแบบเชิงสถิต โดยจะประกอบด้วย
  - Use Case Diagram
  - Class Diagram
  - Object Diagram
- การออกแบบเชิงไดนามิก โดยจะประกอบด้วย
  - State Diagram
  - Sequence Diagram
  - Collaboration Diagram
  - Activity Diagram
  - Component Diagram
  - Deployment Diagram

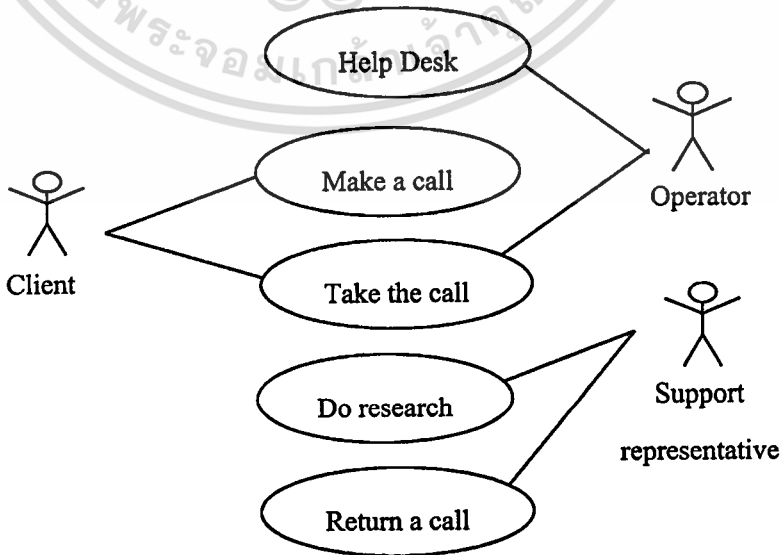
2.1.1 การออกแบบเชิงสถิติก (Static Modeling)

การออกแบบเชิงสถิติก จะเป็นการอธิบายถึงโครงสร้างโดยรวมของระบบ โดยจะมีการออกแบบแบบจำลองในสองมุมมอง คือในมุมมองภายนอกระบบ โดยการใช้ Use-Case Diagram และในมุมมองภายในระบบ ซึ่งจะใช้ Class Diagram และ Object Diagram เพื่อการแสดงโครงสร้างของระบบโดยสมบูรณ์ จึงใช้ไคอะแกรมทั้ง 3 ไคอะแกรมช่วยในการแสดงโครงสร้างแบบของระบบ

2.1.1.1 แผนผังยูสเคส (Use-Case Diagram)

Use-Case Diagram เป็นมุมมองภายนอกระบบ จะแสดงความสัมพันธ์โดยรวมของระบบ คือ แสดงความต้องการของระบบ ว่า ในระบบนั้น มีการดำเนินงาน หรือ มีกิจกรรมอะไรเกิดขึ้นบ้าง โดยภายใน แผนผังยูสเคส หนึ่งๆ จะประกอบด้วย

- 1. แอ็คเตอร์ (Actor) ซึ่งจะใช้สัญลักษณ์เป็นรูปคน โดย แอ็คเตอร์ จะหมายถึงบุคคล หรือสิ่งต่างๆ ที่ เข้ามาเกี่ยวข้องกับระบบ หรือ เข้ามาใช้ระบบ ซึ่งแอ็คเตอร์จะมีการส่งและรับข่าวสาร (Message) กับระบบหรือมีการแลกเปลี่ยนข้อมูลกับระบบ
- 2. ยูสเคส (Use Case) จะใช้สัญลักษณ์เป็นวงรี โดย ยูสเคส จะแสดงถึงกิจกรรมหลักๆ ที่เกิดขึ้นภายในระบบนั้น ซึ่งอาจจะเป็นกิจกรรมที่เป็นการติดต่อกันระหว่างผู้ใ้กับระบบ หรือ ระหว่างระบบกับระบบ



3. ความสัมพันธ์ (Relationships) ซึ่งเป็นความสัมพันธ์ระหว่างแอ็คเตอร์กับยูสเคส หรือระหว่างยูสเคสด้วยกันเอง ซึ่งสามารถแบ่งความสัมพันธ์ได้เป็น 2 แบบ คือ

- Extends Relationships

ซึ่งเป็นความสัมพันธ์แบบซูเปอร์คลาส (Super Class) กับซับคลาส (Sub Class) ยูสเคสหนึ่ง เอ็ซเทน (extend) อีกยูสเคสหนึ่ง ซึ่งจะหมายความว่า ยูสเคสแรกนั้น เป็นซับคลาส ส่วนยูสเคสหลังจะเป็น ซูเปอร์คลาส ซึ่งจะหมายถึงว่า ซับคลาส จะมีกิจกรรมทุกอย่างกิจกรรม เหมือนกับที่ ซูเปอร์คลาส มี และจะมีกิจกรรมเพิ่มเติมอื่นๆ ที่นอกเหนือออกมาอีก โดยจะใช้สัญลักษณ์แทน ดังในรูปที่ 2.2

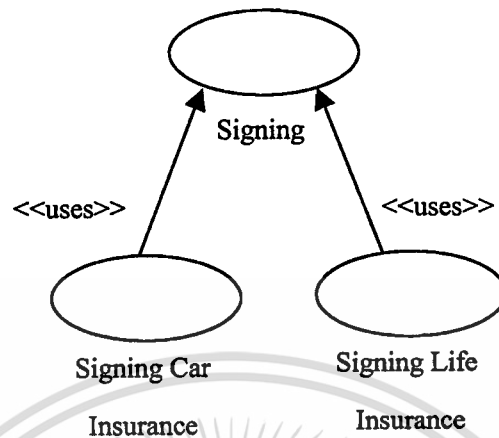


รูปที่ 2.2 แสดงความสัมพันธ์แบบ Extends Relationships

- Uses Relationships

ในกรณีที่ ยูสเคส หลายๆ ยูสเคส มีการกิจกรรมอย่างใดอย่างหนึ่งที่เหมือนกัน สามารถที่จะปรับปรุงกิจกรรมนั้นๆ มาสร้างเป็น ยูสเคสใหม่ อีกยูสเคส หนึ่งได้ แล้วให้ ยูสเคส อื่นๆ มาเรียกใช้ ยูสเคสใหม่ นั้นได้ โดยแสดงตัวอย่างความสัมพันธ์ ดังรูปที่ 2.3





## รูปที่ 2.3 แสดงความสัมพันธ์แบบ Uses Relationships

### 2.1.1.2 แผนผังคลาส (Class Diagram)

เมื่อทำการสร้าง แผนผังยูสเคส เรียบร้อยแล้ว ขั้นตอนต่อไปในการออกแบบ คือ การสร้างแผนผังคลาส (Class Diagram) ซึ่งเป็นส่วนที่ระบุว่า ในแบบจำลองนี้มี คลาส อะไรประกอบอยู่บ้าง พร้อมทั้งแสดงความสัมพันธ์แบบต่างๆ ระหว่างคลาส ด้วย

คลาสใช้สัญลักษณ์สี่เหลี่ยมผืนผ้า ภายในมีองค์ประกอบ 3 ส่วน คือ ส่วนบนจะเป็นส่วนของชื่อ (class name) แสดงเป็นตัวหนาจัดอยู่กึ่งกลาง ส่วนกลางจะเป็นส่วนของคุณลักษณะประจำ (attribute) พร้อมด้วยการกำหนดประเภทและความสามารถในการมองเห็น ซึ่งจะใช้ (+) แทน Public (-) แทน Private และ (#) แทน Protect ) และส่วนล่างจะเป็นส่วนของพฤติกรรม (Operation) พร้อมด้วยพารามิเตอร์และประเภทของค่าที่มีการส่งคืน และยังรวมถึงแสดงความสามารถในการมองเห็นด้วย หรืออาจจะแสดงในลักษณะของชื่ออย่างเดี่ยว

| Class name                        |
|-----------------------------------|
| + Attribute name : Attribute type |
| + Operation name (Parameter)      |

## รูปที่ 2.4 แสดงสัญลักษณ์แทน คลาส

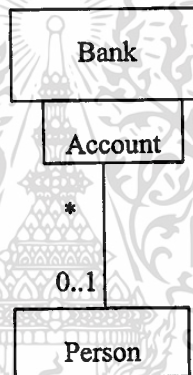
ความสัมพันธ์ระหว่างคลาสที่ใช้กันสามารถแบ่งได้เป็น 4 ความสัมพันธ์หลักๆ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นความสัมพันธ์แบบสองทางระหว่างคลาส แสดงถึงความเกี่ยวข้องกันระหว่างคลาส ใช้สัญลักษณ์เส้นตรงและมีชื่อของความสัมพันธ์กำกับด้วย โดยแบ่งเป็นความสัมพันธ์ย่อยได้ดังนี้

### 1.1 Qualified Association

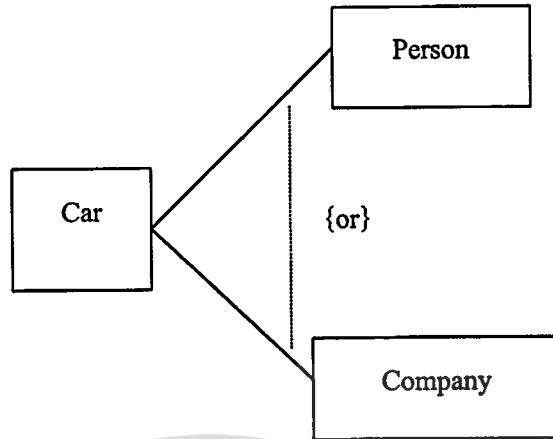
เป็นความสัมพันธ์ระหว่างออบเจกต์ 2 ออบเจกต์ ซึ่งสามารถเป็นได้ทั้งความสัมพันธ์แบบ one-to-many หรือ many-to-many และนอกจากนี้ความสัมพันธ์แบบนี้จะเป็นการระบุว่า ระหว่างออบเจกต์นั้นๆ สัมพันธ์กันที่แอททริบิวต์(Attribute)อะไร โดยแสดงสัญลักษณ์ ดังรูปที่ 2.5



รูปที่ 2.5 แสดงความสัมพันธ์แบบ Qualified Association

### 1.2 OR-Association

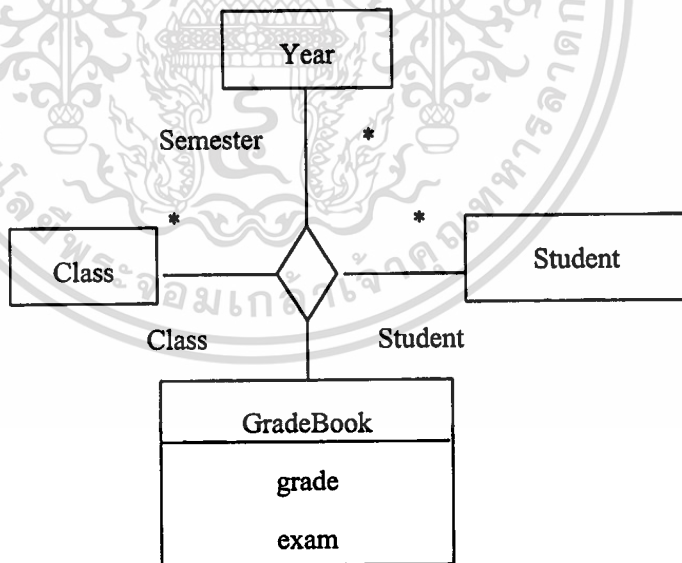
เป็นความสัมพันธ์ที่เกิดขึ้นระหว่างออบเจกต์หนึ่ง กับอีก 2 ออบเจกต์ โดยออบเจกต์แรกนั้น จะต้องเลือกว่า จะมีความสัมพันธ์กับออบเจกต์ใดเพียงออบเจกต์เดียวเท่านั้น ในระหว่างออบเจกต์อีก 2 ออบเจกต์ที่เหลือ โดยจะมีความสัมพันธ์กับทั้งสองออบเจกต์พร้อมๆ กันไม่ได้ โดยแสดงสัญลักษณ์ ดังรูปที่ 2.6



รูปที่ 2.6 แสดงความสัมพันธ์แบบ OR-Association

### 1.3 Ternary Association

เป็นความสัมพันธ์ระหว่างคลาส ตั้งแต่ 2 คลาสขึ้นไป ซึ่งสามารถแสดงความสัมพันธ์ได้ดังรูปที่ 2.7



รูปที่ 2.7 แสดงความสัมพันธ์แบบ Ternary Association

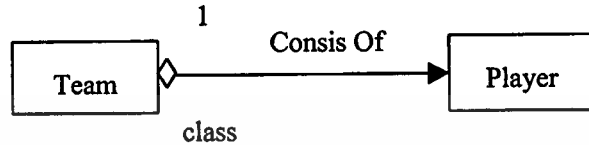
## 2. Aggregation

เป็นความสัมพันธ์ที่แสดงถึงส่วนประกอบต่างๆ ภายในคลาส โดยสามารถแบ่งได้เป็น 2 แบบ คือ

### 2.1 Shared Aggregation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

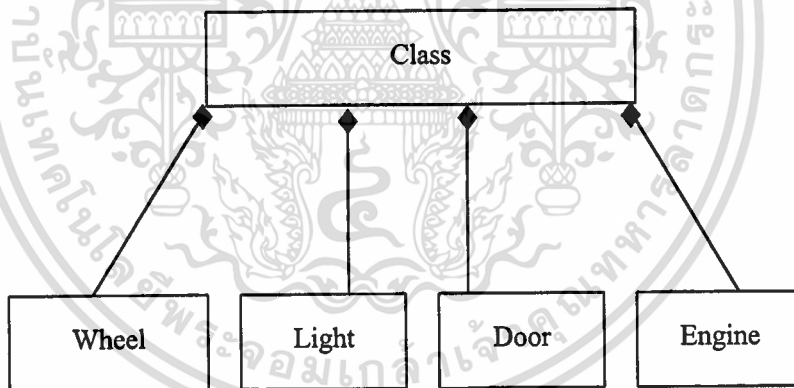
เป็นการแสดงถึงส่วนประกอบของ คลาส แบบหลวมๆ ซึ่งหมายความว่า คลาสแรก อาจไม่จำเป็นต้องมี อีกคลาส เป็นส่วนประกอบก็ได้ ดังตัวอย่างในรูปที่ 2.8



รูปที่ 2.8 แสดงความสัมพันธ์แบบ Shared Aggregation

## 2.2 Composition Aggregation

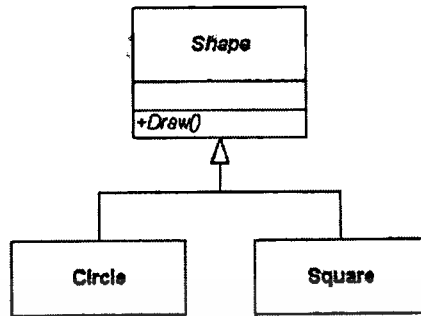
เป็นการแสดงถึงส่วนประกอบของคลาสนั้นๆ โดยที่คลาสแรก ต้องมี อีกคลาสหนึ่งเป็นส่วนประกอบเสมอ จึงจะเป็นคลาสขึ้นมาได้



รูปที่ 2.9 แสดงความสัมพันธ์แบบ Composition Aggregation

## 3. Generalization

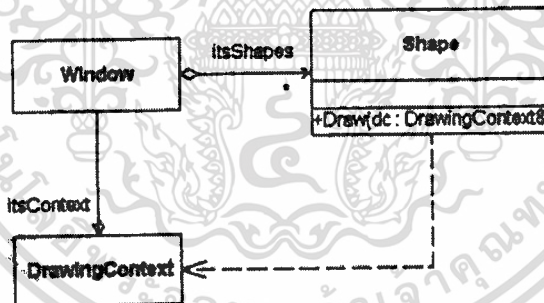
เป็นความสัมพันธ์ ที่แบ่งระหว่างซูปเปอร์คลาส กับซับคลาส (Superclass/Subclass) หรือเป็นความสัมพันธ์ที่บางครั้งเรียกว่าความสัมพันธ์แบบ "is-a relationship" ใช้สัญลักษณ์ลูกศรหัวปิดที่ชี้ไปยังซูปเปอร์คลาส แสดงถึงการถ่ายทอดทั้งคุณลักษณะและพฤติกรรมจากซูปเปอร์คลาสทั้งหมดไปยังซับคลาส โดยแสดงดังรูปที่ 2.10



รูปที่ 2.10 แสดงตัวอย่างความสัมพันธ์แบบ Generalization ภายใน Class Diagram

#### 4. Dependency

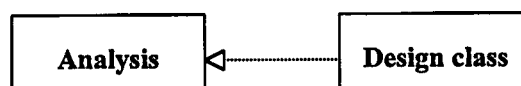
เป็นความสัมพันธ์ในลักษณะที่มีการขึ้นตรงต่อกัน คือเมื่อมีการแก้ไขสิ่งหนึ่งจะมีผลกับอีกสิ่งหนึ่งที่เกิดขึ้นต่อกัน (Dependent) ใช้สัญลักษณ์เป็นเส้นประและลูกศรหัวเปิด ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 แสดงตัวอย่างความสัมพันธ์แบบ Dependency ภายใน Class Diagram

#### 5. Refinement

เป็นความสัมพันธ์ของรูปแบบการอธิบาย 2 แบบซึ่งอธิบายในสิ่งๆ เดียวกัน แต่คนละมิติกัน ใช้สัญลักษณ์เป็นเส้นประและลูกศรหัวปิด ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 แสดงตัวอย่างความสัมพันธ์แบบ Refinement ภายใน Class Diagram

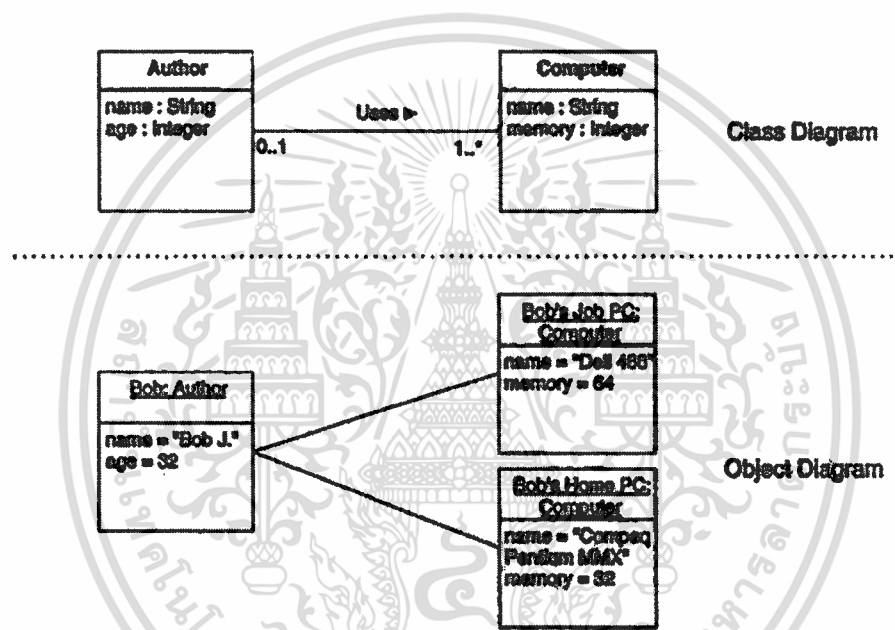
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2.1.1.3 แผนผังออบเจกต์ (Object Diagram)

ในส่วน of แผนผังออบเจกต์ (Object Diagram) นี้จะเป็นส่วนที่แสดงรายละเอียดและความสัมพันธ์ระหว่าง ออบเจกต์ (Object) ในแต่ละคลาส

โดยสัญลักษณ์ที่ใช้แทน ออบเจกต์จะเป็นสัญลักษณ์เดียวกับ ที่ใช้แทน คลาส แต่ในส่วนชื่อของออบเจกต์นั้นจะมีการขีดเส้นใต้ ซึ่งเป็นส่วนที่แตกต่างจากคลาส



รูปที่ 2.13 แสดงการเปรียบเทียบระหว่าง Class Diagram กับ Object Diagram

2.1.2 การออกแบบเชิงไดนามิก (Dynamic Model)

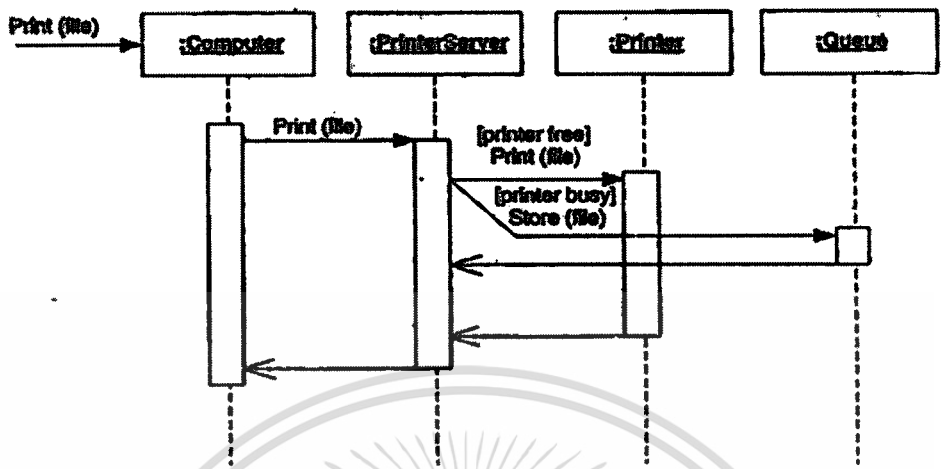
เมื่อทำการสร้าง Class Diagram แล้วขั้นตอนต่อไปเป็นการสร้างแบบจำลองไดนามิก (Dynamic Model) ซึ่งแบบจำลองไดนามิกแบ่งได้ดังนี้

2.1.2.1 แผนผังซีควเอนซ์ (Sequence Diagram)

เป็นแผนผังที่อธิบายถึงความสัมพันธ์ของออบเจกต์ ในเวลาต่างๆ ว่า มีการเปลี่ยนแปลงอย่างไร คือ จะมีเรียงลำดับออบเจกต์ตามแนวนอน และส่วนในแนวตั้งมีลำดับการส่งผ่านข้อความ (Message) จากออบเจกต์หนึ่งไปยังออบเจกต์หนึ่ง

ใน Sequence Diagram สัญลักษณ์สี่เหลี่ยมผืนผ้าแทนออบเจกต์หรือคลาส สัญลักษณ์เส้นประในแนวตั้งเรียกว่า เส้นชีวิตของออบเจกต์ (Object 's Lifeline) เป็นการ

เอกสารนี้เป็นเอกสารกำหนดขอบเขตการทำงานของออบเจกต์นั้น ตัวอย่างแสดงได้ดังรูปที่ 2.14 ประโยชน์ด้านการค้าไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

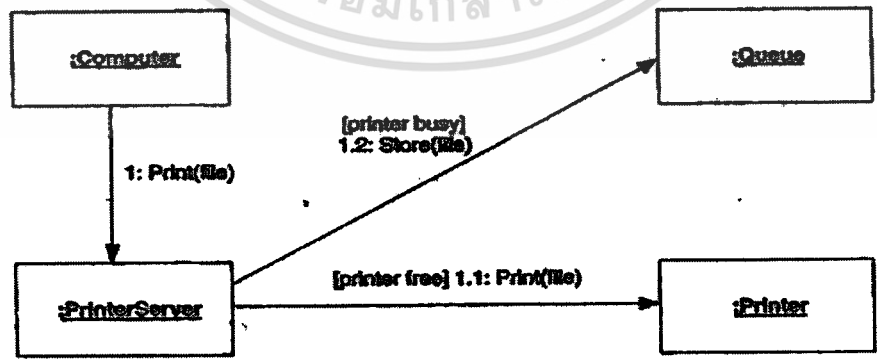


รูปที่ 2.14 แสดงตัวอย่าง Sequence Diagram

2.1.2.2 แผนผังคอแลบอเรนซ์ (Collaboration Diagram)

เป็นแผนผังที่อธิบายถึงออบเจกต์และความสัมพันธ์กันระหว่างออบเจกต์ในลักษณะการส่งข้อความไปมาระหว่างผู้ส่ง(sender) และผู้รับ(receiver) โดยไม่มีลำดับสัญลักษณ์ที่ใช้ ใน Collaboration Diagram เหมือนกับ Sequence Diagram

การทำงานของ Collaboration Diagram จะคล้ายกับ Sequence Diagram ดังนั้นจึงสามารถเลือกแสดงได้ในรูปแบบใดรูปแบบหนึ่ง กล่าวคือทั้งสองแผนผังสามารถเปลี่ยนแปลงรูปแบบระหว่างกันได้

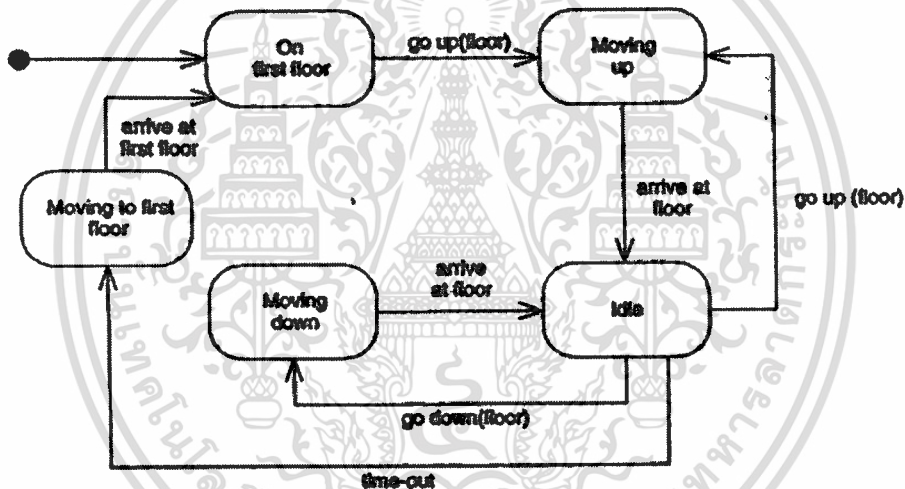


รูปที่ 2.15 แสดงตัวอย่าง Collaboration Diagram

### 2.1.2.3 แผนผังสถานะ (State Diagram)

อธิบายถึงวงจรชีวิต (Life Cycle) ของออบเจกต์ โดยจะเขียน State Diagram เฉพาะในส่วนของคลาสที่มีความซับซ้อน เพื่อช่วยให้สามารถกำหนดสถานะ (State) ของคลาสได้ชัดเจนยิ่งขึ้น

โดยมีสัญลักษณ์วงกลมทึบเป็นจุดเริ่มต้นของแผนผังนี้ หรือ กล่าวได้ว่าออบเจกต์ถูกสร้างขึ้นที่จุดนี้ ส่วนสัญลักษณ์วงกลมทึบที่มีวงกลมล้อมรอบเป็นจุดสิ้นสุด ถูกสรรระหว่างสถานะ เป็นการแสดงถึง การเปลี่ยนแปลงที่เกิดขึ้นระหว่างสถานะต่างๆ ของออบเจกต์ ตัวอย่างของ State Diagram แสดงได้ดังรูปที่ 2.16



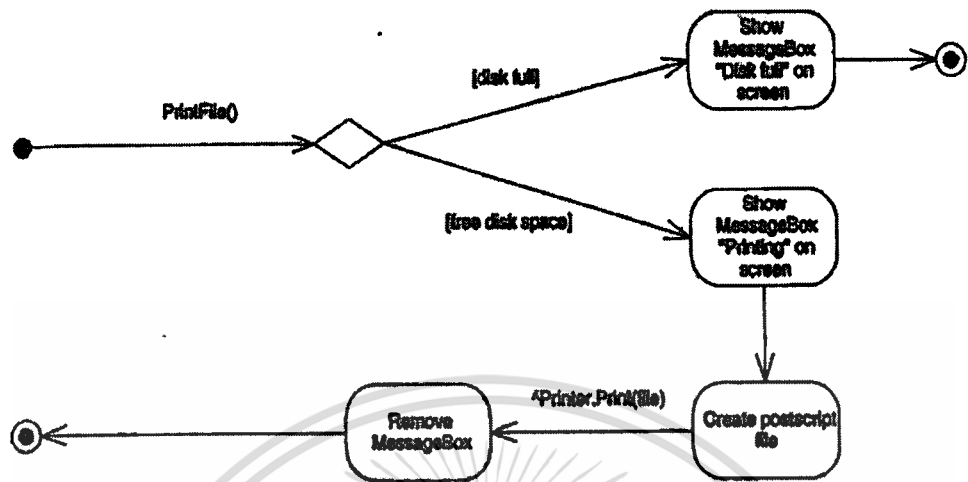
รูปที่ 2.16 แสดงตัวอย่าง State Diagram

### 2.1.2.4 แผนผังกิจกรรม (Activity Diagram)

เป็นแผนผังที่แสดงถึงการกระทำ (Action) และผลลัพธ์ของการกระทำนั้น โดยจะเน้นถึงขั้นตอนดำเนินการ (Operation/Method) และกิจกรรม (Activity) ในออบเจกต์นั้น

Activity Diagram คล้ายกับ State Diagram เพียงแต่มีจุดประสงค์ที่ต่างกัน คือ Activity Diagram เน้นที่การกระทำ และผลลัพธ์ ซึ่งผลลัพธ์ที่ได้ ก็คือการเปลี่ยนแปลงสถานะของออบเจกต์ ใน State Diagram นั้นเอง

สัญลักษณ์ที่ใช้ใน Activity Diagram เหมือนกับ State Diagram แต่มีสัญลักษณ์เพิ่มขึ้นมาคือรูปสี่เหลี่ยมขนมเปียกปูน ซึ่งแสดงถึงการตัดสินใจ โดยแสดงตัวอย่างของ Activity Diagram ในรูปที่ 2.17



รูปที่ 2.17 แสดงตัวอย่าง Activity Diagram

2.1.3 โครงสร้างภายนอก(Physical Architecture)

อธิบายถึงแต่ละส่วนของซอฟต์แวร์และฮาร์ดแวร์ โดยมีการแปลง (Mapping) จากโครงสร้างภายใน (Logical Architecture) มาเป็นโครงสร้างภายนอก (Physical Architecture) สามารถแบ่งแผนผังได้ 2 แผนผังคือ

2.1.3.1 แผนผังองค์ประกอบ (Component Diagram)

แสดงถึงหลักการของโครงสร้างภายนอก และหน้าที่ที่ถูกกำหนดโดยโครงสร้างภายใน เช่น คลาส ออบเจกต์ และความสัมพันธ์ระหว่างออบเจกต์

Component Diagram แสดงถึงไฟล์ที่มีในระบบ หรือในสถานะแวดล้อมของการพัฒนาระบบ

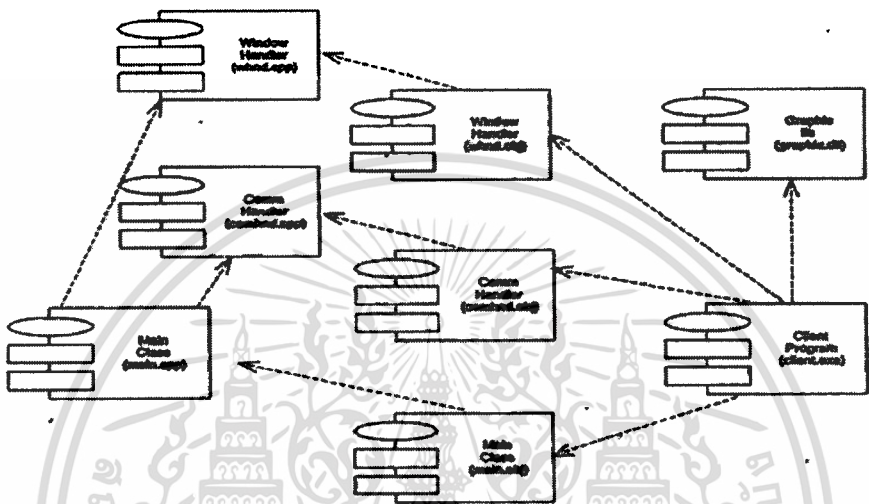
สัญลักษณ์ที่ใช้ใน Component Diagram คือ คอมโพเนนท์ จะแทนด้วยรูปสี่เหลี่ยมผืนผ้าที่มีรูปร่าง และสี่เหลี่ยมผืนผ้าเล็กๆสองรูปอยู่ทางด้านซ้าย ส่วนเส้นประที่มีลูกศรหัวเปิดแสดงถึงการเชื่อมต่อกันระหว่างคอมโพเนนท์

โดยภายในคอมโพเนนท์ มีส่วนประกอบได้ดังนี้

- 1. Source component  
เป็นโค้ดไฟล์ก่อนทำการคอมไพล์โปรแกรม
- 2. Binary component  
เป็นออบเจกต์โค้ดไฟล์ที่ทำการคอมไพล์เรียบร้อยแล้ว และรวมถึงไลบรารีไฟล์

3. Executable component

เป็นไฟล์ ที่ทำการลิงค์ ทุกๆ ไบนารีคอมโพเนนท์เข้าไว้ด้วยกัน

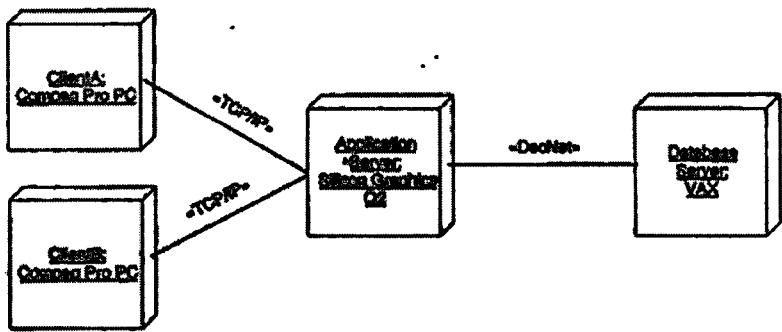


รูปที่ 2.18 แสดงตัวอย่าง Component Diagram

2.1.3.2 แผนผังดีพลอยเมนต์ (Deployment Diagram)

เป็นการพยากรณ์สถาปัตยกรรมของระบบ ขณะโปรแกรมกำลังทำงาน (Run-time) รวมถึง โปรโตคอล (Protocol) ในการเชื่อมต่อระหว่างโหนด (Node) ซึ่งได้แก่ โปรเซสเซอร์ (Processor) อุปกรณ์ต่างๆ และซอฟต์แวร์ ต่างๆ ไว้ด้วยกัน

สัญลักษณ์ที่ใช้ คือ โหนด จะแทนด้วย กล่องสี่เหลี่ยม ส่วนสัญลักษณ์เส้นทึบที่ลากเชื่อมกันระหว่างโหนด แทนการเชื่อมโยง โดยมีป้ายกำกับบอกถึงโปรโตคอลในการเชื่อมโยงด้วย





## 2.2 การวิเคราะห์เชิงวัตถุ

ในขั้นตอนการวิเคราะห์ระบบนั้น วัตถุประสงค์หลักของการวิเคราะห์คือการกำหนดขอบเขต และความต้องการของระบบ นอกจากนี้จะต้องระบุถึงสิ่งที่ระบบต้องทำตามความจำเป็นและความต้องการของผู้ใช้งาน

ขั้นตอนในการวิเคราะห์ระบบ ตามระเบียบวิธีของ UML นั้น ประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. ออกแบบ Activity Diagram
2. หา Actor ที่เกี่ยวข้องกับระบบ
3. ออกแบบ Use case Diagram
4. ออกแบบ Interaction Diagram
5. สร้าง Class Diagram
6. การทำซ้ำ (Iterate) และการแก้ไขให้ดีขึ้น (Refine)

### 2.2.1 ออกแบบ Activity Diagram

ออกแบบระบบงานของระบบ โดยการสร้าง Activity Diagram ซึ่งเป็นแผนผังที่แสดงถึง ภาพรวมของกิจกรรมทางธุรกิจที่เกิดขึ้นทั้งหมดในระบบ เพื่อให้เกิดความเข้าใจถึง ความต้องการต่างๆ ของผู้ใช้งาน

### 2.2.2 หา Actor ที่เกี่ยวข้องกับระบบ

Actor หมายถึงบทบาทของผู้ใช้งานที่เกี่ยวข้องกับระบบ ซึ่งผู้ใช้งาน 1 คน อาจมีได้หลายบทบาทที่เกี่ยวข้องกับระบบนั้นๆ การกำหนด Actor นั้นสามารถจะได้อมาโดยการตอบคำถามดังต่อไปนี้

- ใครคือผู้ใช้ระบบ หรือใครคือผู้รับผลลัพธ์ที่ได้จากระบบ
- ใครมีผลต่อระบบ หรือกลุ่มใดที่มีความจำเป็นต่อระบบ
- ฮาร์ดแวร์อะไร หรือมีระบบอื่นระบบใด ที่มาใช้ระบบ
- Application ของระบบแก้ปัญหาของใคร
- ผู้ใช้งานใช้ระบบอย่างไร สิ่งใดที่ผู้ใช้งานเหล่านั้นทำกับระบบ

### 2.2.3 ออกแบบ Use case Diagram

โดยพิจารณาว่า ระบบต้องมีกิจกรรมใดบ้าง เพื่อให้บริการแก่ Actor ของระบบ

Use case จะแทนด้วยหน้าที่ ที่ระบบสามารถทำงาน โดยให้พิจารณาจากสิ่งที่ Actor

ต้องการจะทำ

โดยการหา Use case จะหาได้จากขั้นตอนต่อไปนี้คือ

1. หางานและหน้าที่ที่ Actor จะปฏิบัติ หรืองานที่ระบบต้องการให้ Actor ปฏิบัติ
2. กำหนดชื่อให้กับ Use Case ชื่อที่ใช้ควรสื่อความหมายให้ชัดเจนถึงกิจกรรม หรือหน้าที่ของ Use Case นั้น โดยทั่วไปชื่อ Use Case มักขึ้นต้นด้วยคำกริยา หรือการกระทำ
3. อธิบาย Use Case อย่างย่อๆ โดยใช้คำที่ผู้ใช้งานคุ้นเคย เพราะจะทำให้คำอธิบายไม่คลุมเครือน้อยที่สุด

ในระหว่างที่ทำการออกแบบ Use Case นั้น อาจจะต้องมีการเปลี่ยนแปลง Actor ได้ ซึ่งการเปลี่ยนแปลงต่างๆนี้ ควรกระทำอย่างระมัดระวัง

ในการออกแบบ Use Case นั้น ควรมีการออกแบบในลักษณะของการนำกลับมาใช้ใหม่ (Reused) ซึ่งสามารถใช้ความสัมพันธ์ ระหว่าง Use Case แบบ Uses และ Extend ได้

#### 2.2.4 ออกแบบ Interaction Diagram

Interaction Diagram จะแสดงถึงรายละเอียดต่างๆ ของแต่ละกิจกรรมหรือหน้าที่ ซึ่ง Interaction Diagram นั้นจะแสดงถึง รายละเอียดของแต่ละ Use Case

ซึ่งในระเบียบวิธีของ UML นั้น มีรูปแบบแผนผังที่ใช้ในการออกแบบ Interaction Diagram อยู่ 2 ชนิด คือ Sequence Diagram และ Collaboration Diagram

Sequence Diagram แสดงถึงการโต้ตอบของกลุ่มของ ออบเจกต์ ตามลำดับเวลา

Collaboration Diagram จะแสดงความสัมพันธ์ระหว่างกลุ่มของ ออบเจกต์

ซึ่งก็หมายความว่า สามารถใช้ แผนผังใดแผนผังหนึ่งในการอธิบายเหตุการณ์ที่เกิดขึ้นของ Use Case หนึ่งๆ ได้ หรืออาจใช้ทั้งสองแผนผังร่วมกัน

โดย Use Case 1 แผนผังจะสามารถนำมาเขียน Interaction Diagram ได้ 1 แผนผัง

#### 2.2.5 การสร้าง Class Diagram

คือ การจำแนก ออบเจกต์ ต่างๆ ให้มารวมกลุ่มเป็น คลาส โดยนำข้อมูลที่ได้จากการวิเคราะห์ภาพรวมของความต้องการระบบ โดย Use Case และ Interaction Diagram ในข้างต้น มาใช้ประกอบ โดยจะต้องมีการระบุคุณสมบัติของแต่ละคลาส

การตั้งชื่อให้ คลาส เป็นสิ่งสำคัญประการหนึ่ง โดยการตั้งชื่อ คลาส ควรคำนึงถึงสิ่งต่อไปนี้

- ควรเป็นคำนามเอกพจน์ เป็นคำนามหรือเป็นคำนามที่มีส่วนขยายด้วยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ควรเป็นชื่อที่ผู้ใช้งานพอใจ หรือเป็นคำศัพท์ที่ผู้ใช้งานคุ้นเคย โยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ควรจะสื่อความหมายด้วยตัวเอง และสามารถอ่านแล้วเข้าใจได้

### 2.2.5.1 การกำหนดความสัมพันธ์ระหว่าง คลาส

โดยทั่วไปความสัมพันธ์ระหว่าง คลาส จะแบ่งเป็น 3 ประเภท คือ Association, Super-Sub Structure และ Aggregation and a-part of structure

#### Association

จะแทนด้วยการติดต่อบetween 2 คลาส หรือมากกว่า การกำหนดความสัมพันธ์แบบ Association จะเริ่มจากการวิเคราะห์หาการโต้ตอบระหว่าง คลาส โดยในการกำหนดความสัมพันธ์ จะต้องพิจารณาว่า ถ้าหากว่า ออบเจกต์ มีหน้าที่สำหรับทำงานใดงานหนึ่ง และขาดข้อมูลที่เป็นในการปฏิบัติงานแล้ว ออบเจกต์ อาจจะต้องมีการติดต่อ หรือมีความสัมพันธ์กับ ออบเจกต์ อื่นๆ ที่มีข้อมูล ให้ทำงานนั้นแทน

Wirfs-Brock, Wilkerson and Wiener ได้ให้คำถามที่สามารถช่วยในการกำหนดความสัมพันธ์แบบ Association ได้ดังนี้คือ

- คลาส สามารถทำงานให้เสร็จได้ด้วยตัวเองได้หรือไม่
- ถ้าไม่ แล้ว คลาส นั้นต้องการอะไร หรือ สามารถให้คลาสอื่นใดทำงานนั้นแทนได้หรือไม่

โดยทั่วไปความสัมพันธ์แบบ Association จะตรงกับคำกริยา (Verb) หรือคำบุพบท (Preposition) เช่น Part of หรือ Next to เป็นต้น

ในการออกแบบความสัมพันธ์นั้น ควรมีการพิจารณาถึงความสัมพันธ์ที่ไม่จำเป็นออก เช่น ความสัมพันธ์แบบ Ternary Association ซึ่งเป็นความสัมพันธ์ที่มีความซับซ้อน และหากเป็นไปได้ ควรแปลงความสัมพันธ์แบบ Ternary Association ให้เป็นความสัมพันธ์แบบ Binary Association

#### Super-Sub Structure

หรือที่ทราบกันเป็น Generalization Hierarchy เป็นความสัมพันธ์ที่แสดงถึงการความสัมพันธ์ที่ถ่ายทอดคุณลักษณะของคลาส คืออนุญาตให้ ออบเจกต์หนึ่งสามารถสร้างมาจาก ออบเจกต์ อื่นๆ ได้

รูปแบบและแนวทางสำหรับการกำหนดความสัมพันธ์แบบ Super-Sub Relationship (A Generalization)

- Top-Down เป็นการมองที่ชื่อของคลาส โดยมองค่านามที่มีคำขยายค่านามประกอบอยู่

- Bottom-Up เป็นการมอง Attribute และ Method ของแต่ละ คลาส ที่เหมือนกัน แล้วรวมกลุ่มเป็น Common Attribute และ Common Method (ใน Super Class)
- Reusability เป็นการนำโครงสร้างที่ได้นั้น มากกลับมาใช้ใหม่
- Multiple Inheritance เป็นรูปแบบความสัมพันธ์ที่ให้คลาส สามารถมี Parent ได้มากกว่าหนึ่งคลาส ซึ่งเป็นรูปแบบที่ควรหลีกเลี่ยง เพราะเป็นรูปแบบที่มีความซับซ้อนและในทางปฏิบัติ เนื่องจากในหลาย Ancestor อาจมีชื่อของ Method เหมือนกันก็ได้ ดังนั้นอาจทำให้เกิดมีการสับสนในการเรียกใช้ Method ว่า จะเป็น Method ของ Parent ไหน

#### Aggregation and a-part of structure

เป็นความสัมพันธ์ที่แทนด้วย คลาส ที่ประกอบด้วย คลาส อื่นๆ หลาย คลาส ที่มีพฤติกรรมที่ไม่เหมือนกัน หรือแตกต่างกันอย่างมากเช่น คลาสรถยนต์ประกอบไปด้วย คลาส หลายๆ คลาส ซึ่งหนึ่งในนั้นอาจเป็น คลาสวิทยุ และจะเห็นอย่างชัดเจนว่าพฤติกรรมของ คลาสรถยนต์และ คลาสวิทยุนั้นเป็นพฤติกรรมที่ไม่เหมือนกันเลย

ความสัมพันธ์แบบ Aggregation จะมีลักษณะสำคัญหลักๆ อยู่ 2 ประการคือ

- Transitivity เป็นคุณลักษณะที่เกิดขึ้นคือ ถ้า A เป็นส่วนหนึ่งของ B และ B เป็นส่วนหนึ่งของ C แล้ว A ก็เป็นส่วนหนึ่งของ C ด้วย เช่น คาร์บูเรเตอร์เป็นส่วนหนึ่งของเครื่องยนต์ และเครื่องยนต์เป็นส่วนหนึ่งของรถยนต์ ดังนั้น คาร์บูเรเตอร์ ก็เป็นส่วนหนึ่งของรถยนต์ด้วย
- Antisymmetry เป็นคุณลักษณะของความสัมพันธ์ที่บอกว่า ถ้า A เป็นส่วนหนึ่งของ B แล้ว B จะไม่เป็นส่วนหนึ่งของ A ตัวอย่างเช่น เครื่องยนต์เป็นส่วนหนึ่งของรถยนต์ แต่รถยนต์ไม่เป็นส่วนหนึ่งของเครื่องยนต์ เป็นต้น

Coad and Yourdon ได้ให้แนวทางในการกำหนดความสัมพันธ์แบบ Aggregation

ไว้ดังนี้

- Assembly เป็นลักษณะของการรวมกันโดยการสร้างมาจากส่วนต่างๆ เช่น ซุปหัวหอมจะเป็นการรวมกันของ หัวหอม แป้ง เนยก้อน และอื่นๆ เป็นต้น
- Container เป็นลักษณะของการรวมกัน แต่ไม่ใช่ลักษณะที่ถูกสร้างขึ้นด้วยจากส่วนต่างๆ เช่น บ้านจะประกอบไปด้วย เฟอร์นิเจอร์ และเครื่องมือต่างๆ เป็นต้น

- Collection-Member เป็นลักษณะการรวมแบบแนวความคิด (Conceptual) เช่น ทีมฟุตบอลจะประกอบไปด้วยผู้เล่น เป็นต้น

### 2.2.5.2 การกำหนด Attribute

Attribute ของ คลาส ในระบบ จะหมายถึง ข้อมูลที่จำเป็นต่อระบบ ซึ่งอาจใช้คำถาม “ข้อมูลเกี่ยวกับอะไรที่เราต้องการเก็บและสืบค้น” ช่วยในการค้นหา Attribute ของ คลาส ได้

การที่จะได้ Attribute มานั้นอาจใช้แนวทางต่างๆ ต่อไปนี้ ช่วยในการกำหนด Attribute ได้

- โดยปกติแล้ว Attribute ส่วนใหญ่จะตรงกับค่านามที่ตามด้วยคำบุพบท เช่น ราคา ของชุป ซึ่งในบางกรณีก็อาจจะตรงกับคำขยายค่านาม (Adjective) หรือคำขยายกริยา (Adverb) ก็ได้
- พยายามให้ คลาส นั้นมีความเรียบง่าย คือกำหนด Attribute แค่สามารถกำหนดสถานะของ ออบเจกต์ ได้ก็พอ
- ไม่กำหนด Attribute ที่เกิดมาจาก Attribute อื่นๆ (Derived Attribute) เช่น อายุ เพราะค่าของมันจะได้มาจากการคำนวณกับ Attribute วันเกิด ซึ่ง Derived Attribute นี้ควรจะถูกกำหนดเป็น Method
- อย่าพยายามกำหนด Attribute เกินความจำเป็น เพราะสามารถกำหนดเพิ่มได้ในการทำซ้ำต่อๆ ไปได้

### 2.2.5.3 การกำหนด Method

ในออบเจกต์ จะต้องมีการกำหนดบริการของแต่ละออบเจกต์ไว้ด้วย ซึ่งบริการ หรือ หน้าที่ต่างๆ ที่ ออบเจกต์มีจะเรียกว่า Method

### 2.2.6 การทำซ้ำ (Iterate) และการแก้ไขให้ดีขึ้น (Refine)

เป็นการทำซ้ำเพื่อเป็นการเพิ่มสิ่งที่ถูกต้องที่สุด ในทุกๆ ขั้นตอนของการพัฒนานั้น สามารถจะทำซ้ำได้โดยการทำซ้ำนั้นจะเป็นการแก้ไขให้ถูกต้องและสิ่งที่เหมาะสมที่สุด และในการพัฒนาจะค่อยๆ เพิ่มความถูกต้องและความสมบูรณ์ขึ้นมาเรื่อยๆ

## 2.3 การออกแบบเชิงวัตถุ

กระบวนการการออกแบบเชิงวัตถุประกอบไปด้วยขั้นตอนต่างๆ ดังนี้

### 1 ใช้ Axiom ช่วยในการออกแบบ Class

- แก้ไขปรับปรุงให้ Class Diagram สมบูรณ์ขึ้นโดยการเพิ่มรายละเอียดลงไป เช่น กำหนด ชนิด หรือความสามารถในการมองเห็นของ Attributes และ Method รวมถึงกำหนดโปรโตคอล (Protocol) ที่จะใช้ในการติดต่อกับคลาสนั้น
- ทำซ้ำและแก้ไขปรับปรุงให้ดีขึ้นอีกครั้ง



2. การออกแบบ Access Layer ซึ่งเป็นชั้นที่มีหน้าที่ในการเข้าถึงข้อมูลของระบบ

- การสร้าง คลาส แบบ Mirror (Mirror Class) โดยพิจารณาว่า คลาสไหนที่มี การติดต่อกับฐานข้อมูล ให้สร้างเป็น Mirror Class ไว้ใน Access Layer
- กำหนดสร้างความสัมพันธ์ใน Layer
- ทำให้คลาส และความสัมพันธ์มีความชัดเจนมากยิ่งขึ้น
- ทำซ้ำและแก้ไขปรับปรุงให้ดีขึ้นอีกครั้ง

3. การออกแบบ View Layer เป็นการออกแบบคลาสที่ทำหน้าที่ติดต่อระหว่างผู้ใช้งาน กับระบบ (User Interface)

- การออกแบบ User Interface ในชั้น Macro Level
- การออกแบบ User Interface ในชั้น Micro Level
- ทดสอบการใช้งานและความพอใจของผู้ใช้งาน
- ทำซ้ำและแก้ไขปรับปรุงให้ดีขึ้นอีกครั้ง

4. การทำซ้ำและแก้ไขปรับปรุงให้ดีขึ้นการออกแบบทั้งหมด

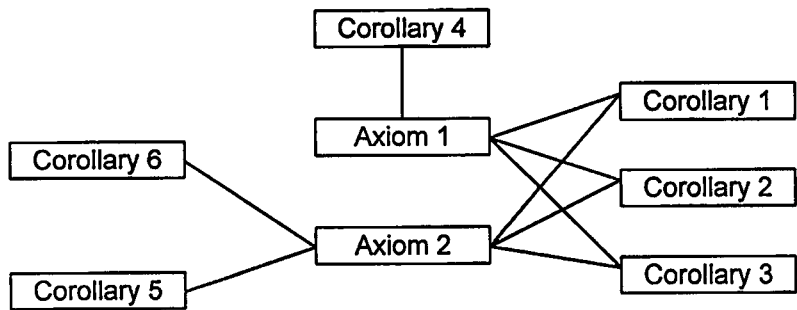
การออกแบบเชิงวัตถุที่ได้นั้นควรจะต้องมีการทำซ้ำ (Iterative) หลายๆ ครั้ง นานเท่าที่คิดว่า คลาส ของ ออบเจกต์ นั้นได้ผ่านการเรียนรู้เรียบร้อยแล้ว และเต็มใจกับการออกแบบ หรือจนกระทั่งรู้สึกพอใจกับกระบวนการที่ผ่านมา

### 2.3.1 Object-Oriented Design Axioms

Axiom คือความจริงที่ไม่ต้องพิสูจน์ซึ่งไม่มีข้อยกเว้น ในการออกแบบครั้งนี้จะใช้ Axiom 2 ข้อ ซึ่งข้อ 1 เป็นข้อที่จัดการกับความสัมพันธ์ระหว่างส่วนประกอบต่างๆ ของระบบ (เช่น Class, Software Component และ Requirement) และข้อที่ 2 เป็นข้อที่จัดการกับความซับซ้อนของการออกแบบ ซึ่งโดยสรุปคือ

1. The independence axiom เป็นพิจารณาความเป็นอิสระของส่วนประกอบ ต่างๆ
2. The information axiom เป็นพิจารณาข้อมูลของการออกแบบให้มีน้อยที่สุด

จาก Axiom ในการออกแบบทั้ง 2 อาจนำไปสู่ที่มาของ Corollary ดังรูปที่ 2.20 ซึ่ง Corollary คือส่วนขยายที่ต่อมาจาก Axiom หรือส่วนขยายที่มาจากส่วนอื่นๆ ที่ได้ผ่านการ พิสูจน์มาแล้ว Corollary อาจมีส่วนช่วยในการตัดสินใจในการออกแบบด้วย เพราะ Corollary เหล่านั้นได้นำมาใช้ในสถานการณ์จริงซึ่งใช้อย่างง่ายกว่า Axiom หรืออาจเรียก Corollary ว่าเป็นกฎในการออกแบบ แต่ทั้งหมดนั้นมีที่มาจาก Axiom ทั้ง 2 นั่นเอง



รูปที่ 2. 20 แสดงความสัมพันธ์ระหว่าง Corollary กับ Axiom

ใช้ Corollary เหล่านี้เป็นหลักเกณฑ์ในการใช้ออกแบบ คลาส

Corollary 1 ออกแบบให้มีข้อมูลน้อยที่สุดและลักษณะเป็น Uncoupled ซึ่งมีจุดประสงค์หลักคือการทำให้อยู่ในออบเจกต์ และ Software Component มีลักษณะเกี่ยวข้องกันมากที่สุด

Corollary 2 ให้มีวัตถุประสงค์เดียว ซึ่งแต่ละคลาส ต้องมีวัตถุประสงค์เดียวที่ชัดเจน ซึ่งอาจจะอธิบายอย่างง่ายๆ ได้โดยใช้ไม่กี่ประโยค

Corollary 3 ให้จำนวนคลาส ที่มีความธรรมดา (Simple) มากๆ เพื่อที่จะนำกลับมาใช้ใหม่ได้ (Reusability)

Corollary 4 ให้มีการแทนกันอย่างลงตัว (Strong Mapping) เป็นการบอกถึงความสัมพันธ์ระหว่างออบเจกต์ ที่ได้มาจากการวิเคราะห์ กับ ออบเจกต์ ที่ใช้ในการออกแบบ ซึ่งในการวิเคราะห์และออกแบบเชิงวัตถุขึ้นอยู่กับพื้นฐานของการสร้างแบบจำลองเดียวกัน ซึ่งในขั้นการออกแบบจะเป็นเพียงการเพิ่มรายละเอียดและยังคงรักษาความจำเป็นที่เหมือนกันเอาไว้

Corollary 5 ให้เป็นมาตรฐาน เป็นการสนับสนุนให้มีมาตรฐานในการออกแบบ ซึ่งใช้ในการติดต่อสื่อสารระหว่าง Component และการนำคลาส หรือ Component กลับมาใช้ใหม่ (Reusing)

Corollary 6 ให้การออกแบบรวมกับการถ่ายทอดไปด้วยกัน ซึ่งก็คือ Common Method ต้องย้ายเป็นอยู่ใน Super-Class เพราะอาจมีผลต่อคำสั่งของโปรแกรมได้ แต่ละขั้นตอนของกระบวนการการออกแบบเชิงวัตถุ จะมีรายละเอียดดังต่อไปนี้

### 2.3.2 การออกแบบ คลาส

เป็นการนำ คลาส ที่ได้มาจากการวิเคราะห์แก้ไขปรับปรุงให้ดีขึ้น โดยการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า กำหนดความสามารถในการมองเห็น (Visibility) ทั้ง Attribute และ Method กำหนดไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทข้อมูลและขนาดของ Attribute ที่ตรงกับภาษาที่ใช้ในการพัฒนาระบบ รวมถึงอาจมีการกำหนดค่าเริ่มต้นให้กับ Attribute บางกรณีอาจต้องมีการเพิ่ม บาง Attribute ขึ้นมาสำหรับรองรับการสร้างระบบด้วย

ในส่วนของ Method จะมีการกำหนดรูปแบบการเรียกใช้ (Protocol) ซึ่งจะรวมถึงการกำหนดความสามารถในการมองเห็น กำหนด Algorithm ในแต่ละ Method โดยอาจใช้ Activity Diagram หรือ State Diagram ช่วยในการแสดง Algorithm ดังกล่าว และจะใช้ Corollary 1 เข้ามาช่วยในการออกแบบ Method ด้วยเพื่อต้องการให้ Method มีความเป็นหนึ่งเดียวมากที่สุด (คือทำงานเพียงหน้าที่เดียว) และลดความซับซ้อนของ Message และจำนวนของการรับ-ส่ง Message ด้วย

บางกรณีอาจมีการใช้ Package เพื่อเป็นการแบ่งกลุ่มของ คลาส ที่ทำหน้าที่เกี่ยวข้องกัน โดยปกติแล้ว Package สามารถใช้เป็นที่รวมกันของ Model Element แม้กระทั่ง Package อื่นๆ อีก เพื่อเป็นการอธิบายระบบในระดับที่สูงขึ้น

### 2.3.3 การออกแบบ Access Layer

เป็นการออกแบบ คลาส ที่ทำหน้าที่เป็นผู้เข้าถึงข้อมูลของระบบ ซึ่งอาจใช้ขั้นตอนดังนี้

1. นำทุก คลาส มาสร้างเป็น คลาส ใหม่ (Mirror Class) ที่ทำหน้าที่เป็นผู้เข้าถึงข้อมูลของระบบ เช่น มี คลาส1 และอาจสร้าง คลาส ใหม่เป็น คลาส1DB เป็นต้น
2. กำหนดความสัมพันธ์ของ คลาส ที่สร้างขึ้นใหม่ ซึ่งอาจใช้ความสัมพันธ์ในลักษณะเดิมก็ได้
3. ทำให้ คลาส และความสัมพันธ์เรียบง่ายและชัดเจน โดยการตัด คลาส ที่ซ้ำกันทิ้ง หรือ คลาส ที่ไม่จำเป็น หรือแม้แต่โครงสร้างที่ไม่จำเป็น เช่น คลาส ที่ให้บริการเหมือนกัน เป็นต้น
4. ทำซ้ำและแก้ไขปรับปรุงให้ดีขึ้น

### 2.3.4 การออกแบบ View Layer

เป็นการออกแบบ คลาส ที่ทำหน้าที่เป็น User Interface ระหว่างผู้ใช้งานกับตัวระบบ ซึ่งอาจใช้แนวทางตามขั้นตอนดังนี้

การออกแบบ User Interface ในขั้น Macro Level เป็นการกำหนด คลาส ที่มีหน้าที่โต้ตอบระหว่างผู้ใช้งานกับระบบ สามารถหาได้จากการวิเคราะห์ Use Case ที่ถูกพัฒนาขึ้นในขั้นการวิเคราะห์ และ Sequence Diagram หรือ Collaboration Diagram สามารถช่วยใน

การกำหนด คลาส ด้วย หลังจากได้ คลาส แล้วจึงมากำหนดความสัมพันธ์ระหว่าง คลาส ที่ ได้มาใหม่ และอาจมีการทำซ้ำแก้ไขปรับปรุงก็ได้

การออกแบบ User Interface ในขั้น Micro Level อาจจะใช้หลักการต่อไปนี้ เป็น แนวทางในการช่วยออกแบบ User Interface ที่ได้มา

- ทำให้ชัดเจนเรียบง่าย (ประยุกต์ Corollary 2) ทำให้ User Interface เรียบง่ายและชัดเจน จนผู้ที่ไม่รู้เรื่องเกี่ยวกับเครื่องมือและกลไกของระบบสามารถทำงานได้
- ทำให้เป็นธรรมชาติ (ประยุกต์ Corollary 4) ผู้ใช้งานปกติจะสามารถคาดเดาเหตุการณ์ที่จะเกิดขึ้นกับ User Interface ได้ โดยเสมือนไม่ได้ทำงานกับระบบอยู่
- ทำให้ผู้ใช้งานเป็นผู้ควบคุม Software (ประยุกต์ Corollary 1) อาจเป็นการเตรียมงานที่เป็นอัตโนมัติเพื่ออำนวยความสะดวกให้กับผู้ใช้งาน หรืออาจเป็นการเพิ่มคำแนะนำหากผู้ใช้งานมีการทำผิดพลาด เป็นต้น ยังรวมถึงทำให้ User Interface มีลักษณะเหมือนกันทั้งระบบ

หลังจากสร้าง User Interface จึงมาถึงขั้นการทดสอบการใช้งานและทดสอบความพอใจของผู้ใช้งานต่อไป ซึ่งจะเป็นการนำ User Interface ที่ได้มาให้ผู้ใช้งานของระบบเป็นผู้ประเมิน และสุดท้ายทำซ้ำแก้ไขปรับปรุงให้ดีขึ้น

### 2.3.5 การทำซ้ำและแก้ไขปรับปรุงให้ดีขึ้นการออกแบบทั้งหมด

การออกแบบเชิงวัตถุเป็นกระบวนการที่ทำซ้ำ (Iterative Process) ซึ่งหลังจากผ่านแต่ละขั้นของการออกแบบอาจจะเป็นการค้นพบสิ่งที่เกี่ยวกับการการสร้างอย่างมากก็ได้

### บทที่ 3

## การวิเคราะห์ระบบ

### 3.1 คำอธิบายระบบ

#### 3.1.1 ความต้องการของระบบ

ระบบประνομหนี่เป็นส่วนหนึ่งของระบบบัญชีลูกหนี่ของการเคหะแห่งชาติ โดยในการดำเนินงานของการเคหะในส่วนของระบบบัญชีลูกหนี่นั้น เมื่อลูกหนี่มีการค้างชำระค่าเช่าซื้อ ทาง การเคหะแห่งชาติจะมีการดำเนินงานในส่วนของการประνομหนี่ให้กับลูกหนี่ที่ค้างชำระเหล่านั้น ซึ่งมีขั้นตอนในการประνομหนี่ต่างๆ เช่น ทำสัญญาเช่าซื้อใหม่ โดยจะทำการกำหนดค่าเช่าซื้อ ต่องวด และระยะเวลาในการชำระใหม่

ซึ่งการเคหะแห่งชาติ ได้กำหนดหลักเกณฑ์ และวิธีปฏิบัติเกี่ยวกับการประνομหนี่ ดังนี้

ตารางที่ 2. 1 แสดงรายละเอียดเงื่อนไข และหลักเกณฑ์ต่างๆในการประνομหนี่

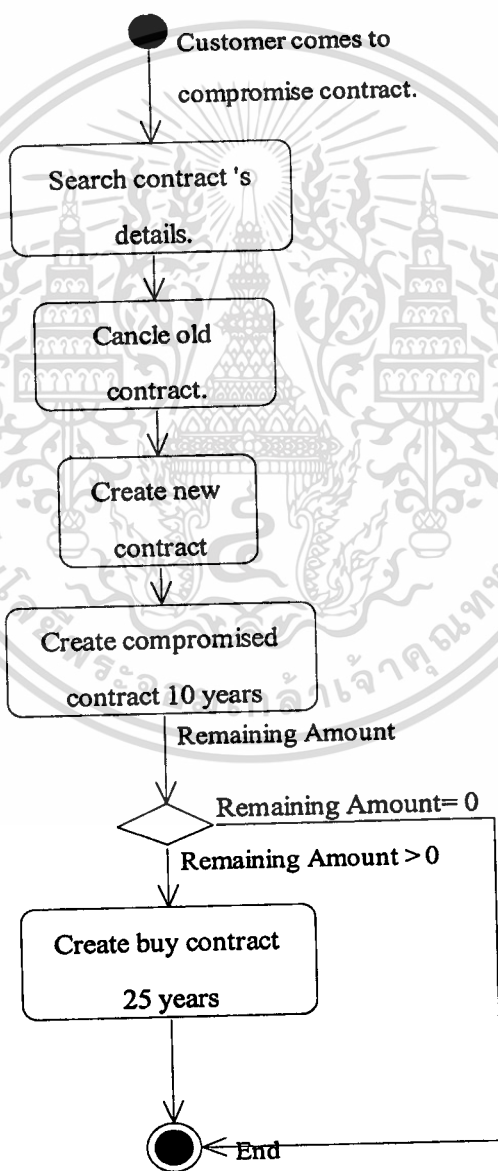
| หลักเกณฑ์                                                                                                                                                                                                                                                                                                                                  | วิธีปฏิบัติ                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| มีเงื่อนไขดังนี้<br>1.ให้ผู้ค้างชำระ ประνομหนี่ได้ในระยะเวลา ไม่เกิน 10 ปี และค่าเช่าซื้อส่วนที่เหลือให้ ทำสัญญาเช่าซื้อใหม่ผ่อนชำระไม่เกิน 25 ปี<br>2. กำหนดให้ ในส่วนการผ่อนชำระไม่เกิน 25 ปี อัตราผ่อนชำระ เท่ากับ อัตราผ่อนชำระ ค่าเช่าซื้อเดิมในช่วง 5 ปีแรก หลังจากนั้น ปรับอัตราเช่าซื้อเพิ่มทุกๆ 5 ปี จนครบอายุ สัญญาไม่เกิน 25 ปี | การทำสัญญา<br>ทำสัญญาใหม่ ให้แยกเป็น 2 สัญญา คือ สัญญาประνομหนี่ไม่เกิน 10 ปี และ สัญญาเช่าซื้อไม่เกิน 25 ปี<br>การคำนวณอัตราผ่อนชำระค่าเช่าซื้อ<br>1. สัญญาประνομหนี่ไม่เกิน 10 ปี<br>คำนวณอัตราผ่อนชำระ โดยใช้อัตรา ดอกเบี้ยคงที่ 7 %<br>2. สัญญาเช่าซื้อไม่เกิน 25 ปี<br>คำนวณอัตราผ่อนชำระค่าเช่าซื้อให้เท่า กับอัตราค่าเช่าซื้อเดิม ในช่วง 5 ปีแรก หลัง จากนั้น ปรับอัตราเช่าซื้อเพิ่มทุกๆ 5 ปี จน ครบอายุสัญญาไม่เกิน 25 ปี |



### 3.2 Activity Diagram

Activity Diagram ช่วยในการสร้างภาพรวมของการดำเนินงานในระบบประណหมนี้ ดังรูป

ที่ 3.1



### 3.3 Use-Case Diagram

#### 3.3.1 Actor

Officer :เจ้าหน้าที่ผู้ใช้งานระบบ

Debtor : ลูกหนี้ของการเคหะ (ลูกค้ำที่ค้างชำระ และต้องการทำการประนอมหนี้)

#### 3.3.2 Use Case

Search Contract 's Details (ค้นหารายละเอียดสัญญา) :

ทำการค้นหารายละเอียดของสัญญาที่ต้องการทำการประนอมหนี้

Search Remaining Amount (ค้นหารายละเอียดการค้างชำระ)

ทำการค้นหารายละเอียดการค้างชำระของสัญญาที่ต้องการทำการประนอมหนี้

Update Old Contract (ยกเลิกสัญญาเดิม) :

ทำการยกเลิกสัญญาเช่าซื้อเดิม

Create New Contract (จัดทำสัญญาใหม่) :

จัดทำสัญญาเช่าซื้อใหม่

Create Compromised Contract (จัดทำสัญญาประนอมหนี้) :

จัดทำสัญญาประนอมหนี้ โดยจะทำการคำนวณ จำนวนงวดที่ต้องชำระ และค่าเช่าซื้อต่องวดใหม่ โดยกำหนดระยะเวลาในการใช้สัญญานี้ 10 ปี และกำหนดอัตราดอกเบี้ยคงที่ 7 %

Create Contract 25 Years (จัดทำสัญญาเช่าซื้อ 25 ปี) :

จัดทำสัญญาเช่าซื้อ 25 ปี โดยจะทำการคำนวณ จำนวนงวดที่ต้องชำระ และค่าเช่าซื้อต่องวดใหม่ โดยกำหนดระยะเวลาในการใช้สัญญานี้ไม่เกิน 25 ปี และกำหนดอัตราดอกเบี้ยให้เพิ่มขึ้น 5 % ทุกๆ 5 ปี

Calculate payment per term (คำนวณจำนวนงวด และค่าเช่าซื้อต่องวด) :

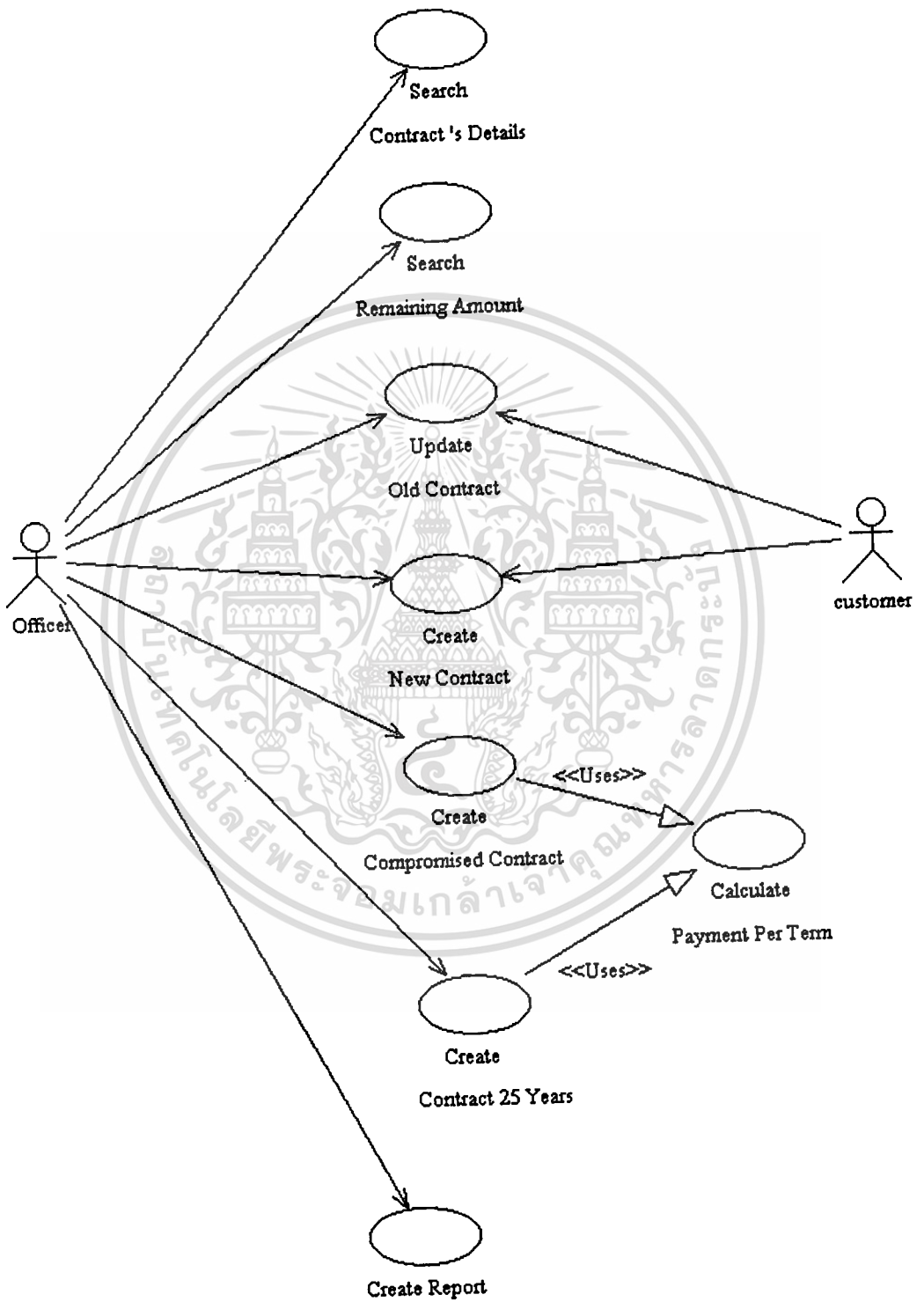
ทำการคำนวณจำนวนงวด และค่าเช่าซื้อต่องวด

Create Report (จัดทำรายงานต่างๆ) :

จัดทำรายงาน โดยจะเป็นการทำรายงาน เกี่ยวกับ ลูกหนี้เช่าซื้อค้างชำระ

#### 3.3.3 Use case Diagram

จาก Actor และ Use case ทั้งหมดที่ได้สามารถนำมาสร้างเป็นไดอะแกรมซึ่งแสดงไว้ดังรูป

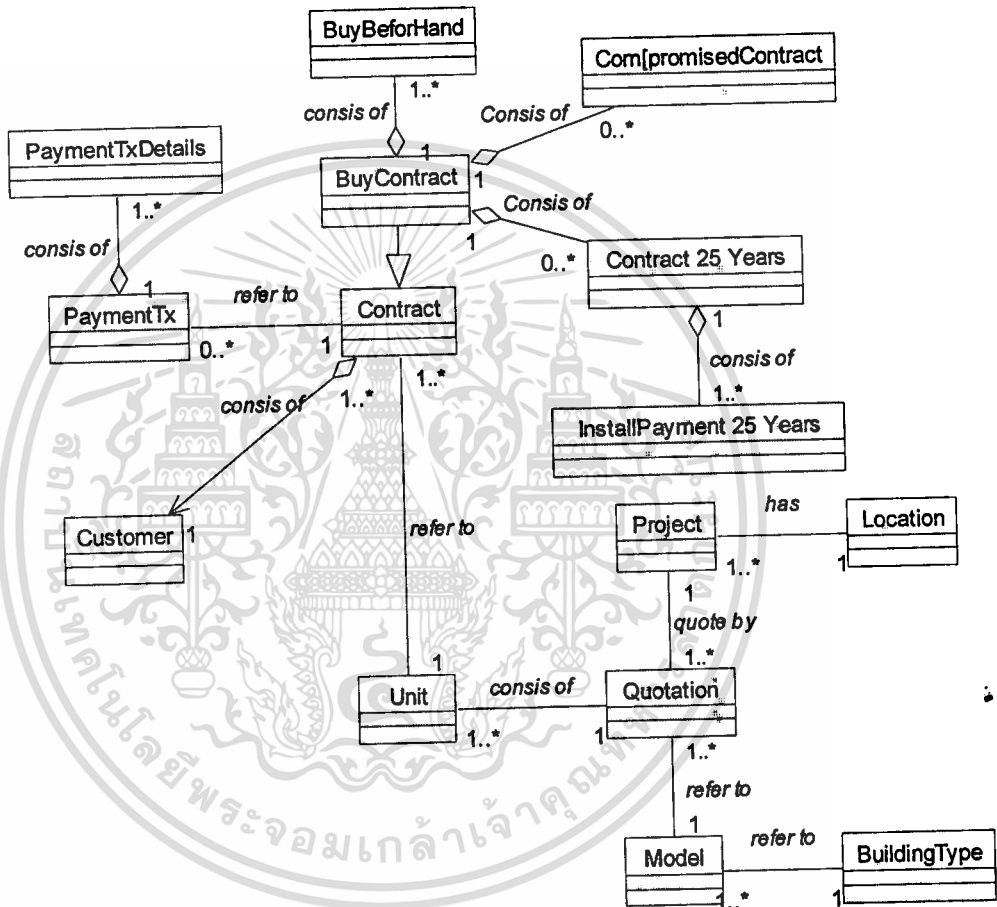


รูปที่ 3. 2 แสดง Use Case Diagram ของระบบประนอมหนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 Class Diagram

จาก Use Case Diagram ที่ออกแบบมาได้นั้น สามารถนำมาเป็นแนวทางในการออกแบบแผนผังคลาสได้ ดังแผนผังคลาสในรูปที่ 3.3

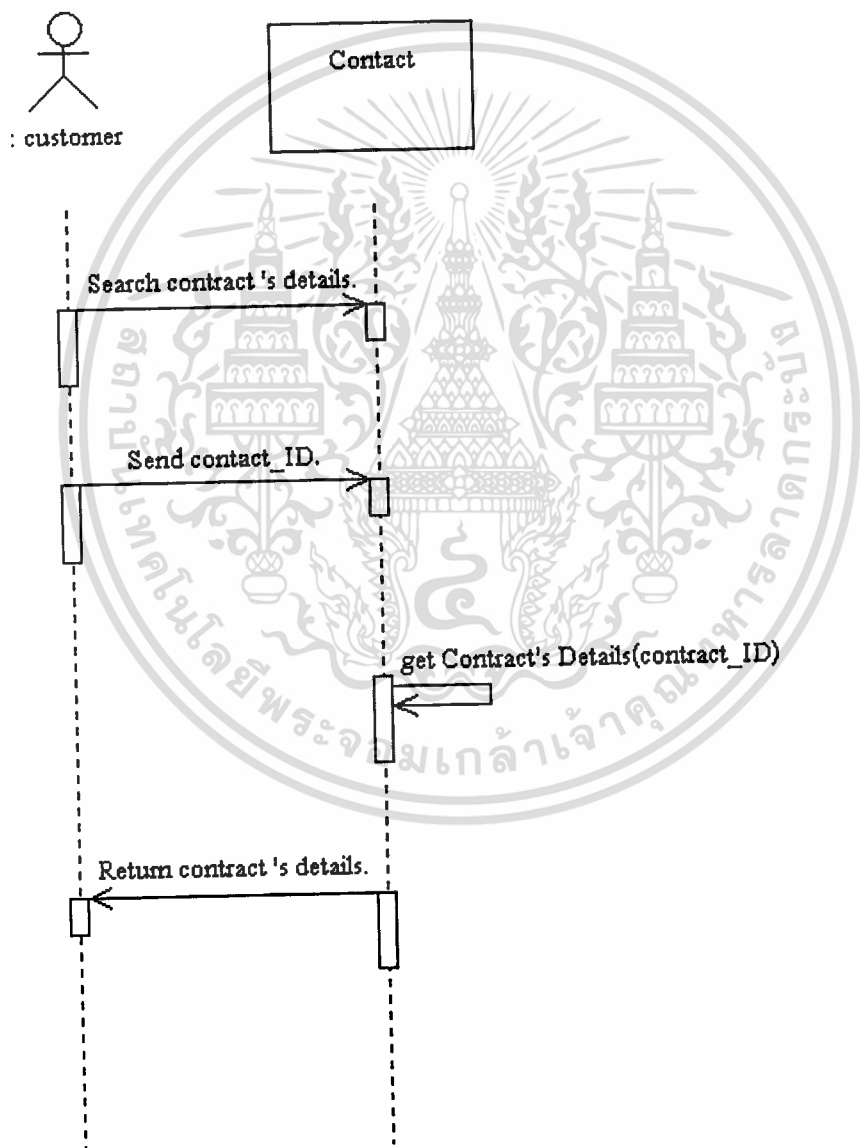


รูปที่ 3.3 แสดง Class Diagram ของระบบระบรณมหนี้ของการเคหะแห่งชาติ

### 3.5 Interaction Diagram

ซึ่งจะใช้ Sequence Diagram เป็นหลักในการแสดงเหตุการณ์ที่เกิดขึ้นเพื่อต้องการแสดงให้เห็นถึงลำดับเหตุการณ์ที่เกิดขึ้น ในแต่ละ Use Case โดยจะเรียงลำดับการแสดงเหตุการณ์ของแต่ละ Use Case ตามลำดับความสำคัญของระบบ

3.5.1 Sequence Diagram ของ Search Contract’s Details Use Case (ซึ่งใช้เลขที่สัญญาในการค้นหา)

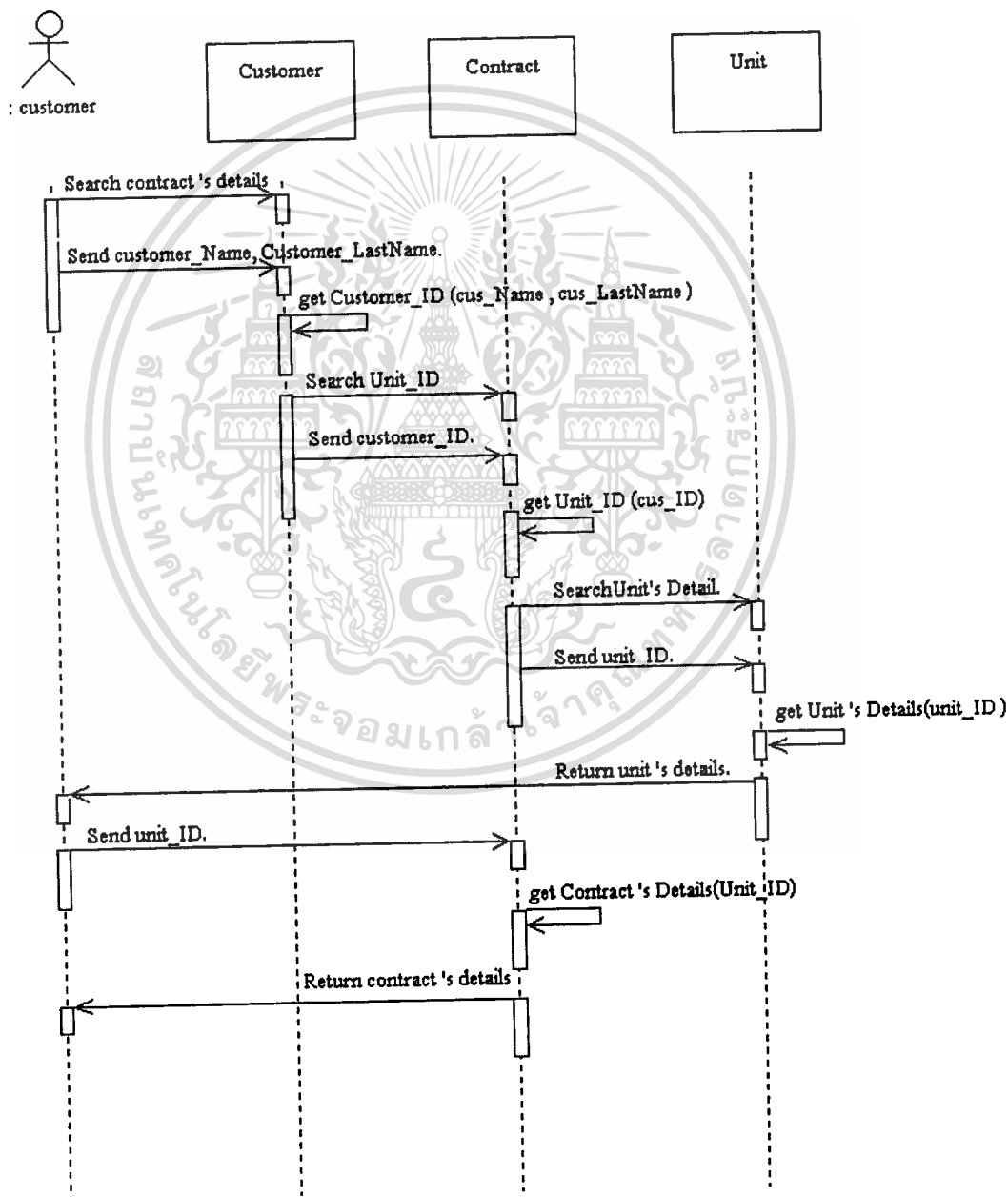


รูปที่ 3. 4 แสดง Sequence Diagram ของ Search Contract’s Details Use Case (ซึ่งใช้ เลขที่สัญญาในการค้นหา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



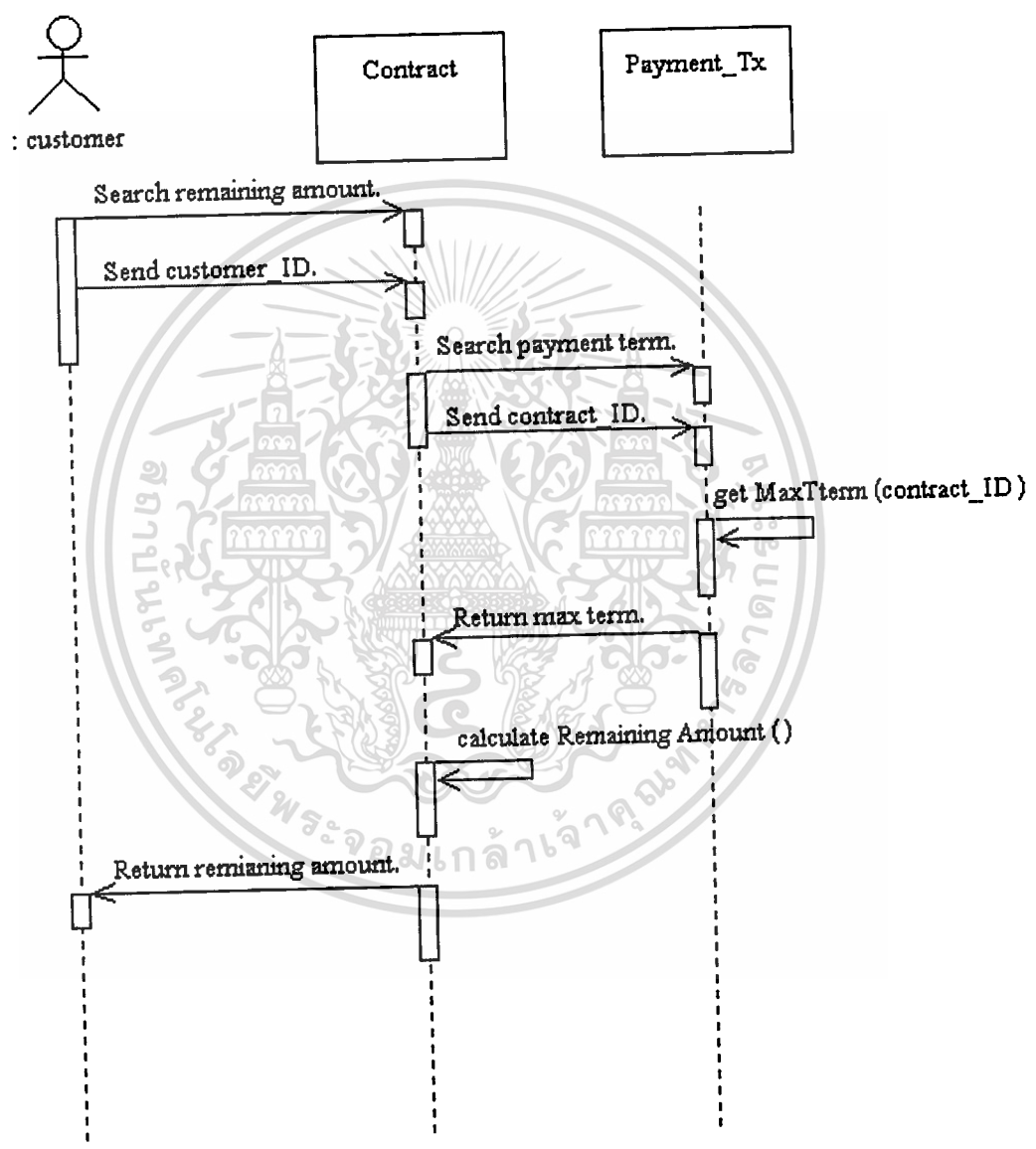
### 3.5.2 Sequence Diagram ของ Search Contract's Details Use Case (ซึ่งใช้ชื่อลูกค้าในการค้นหา)



รูปที่ 3. 5 แสดง Sequence Diagram ของ Search Contract's Details Use Case (ซึ่งใช้ชื่อลูกค้าในการค้นหา)

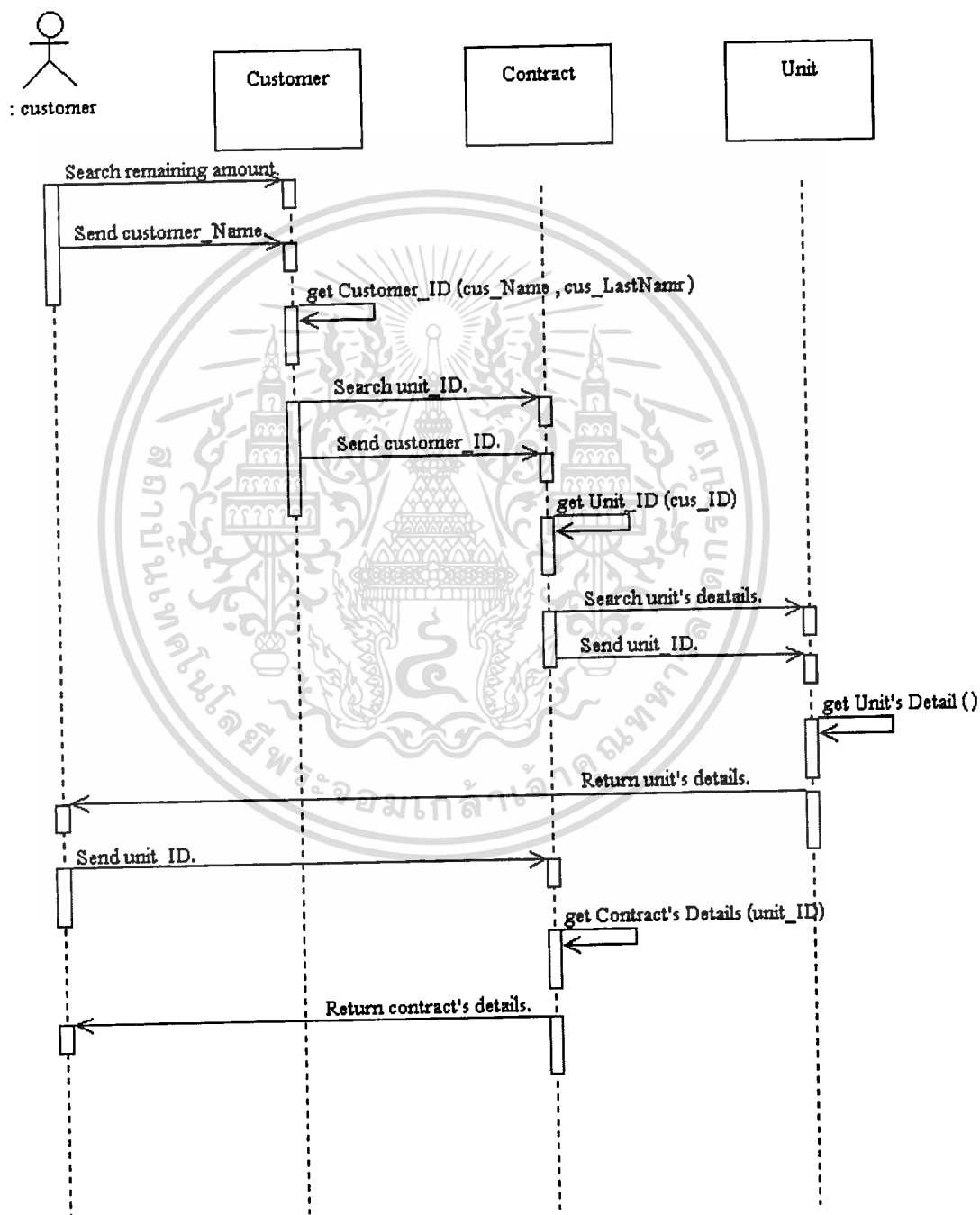
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้เลขที่  
สัญญาในการค้นหา)



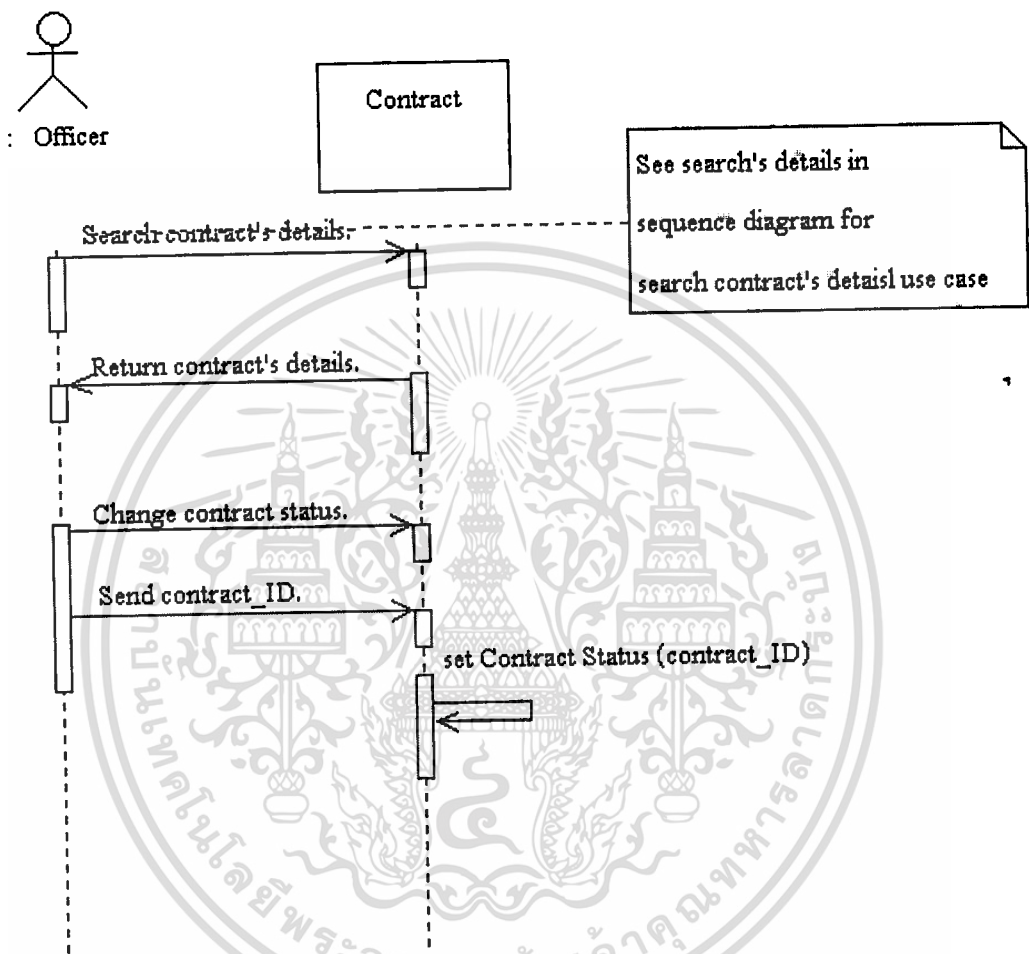
รูปที่ 3. 6 แสดง Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้เลขที่  
สัญญาในการค้นหา)

### 3.5.4 Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้ข้อมูลค่าในการค้นหา)



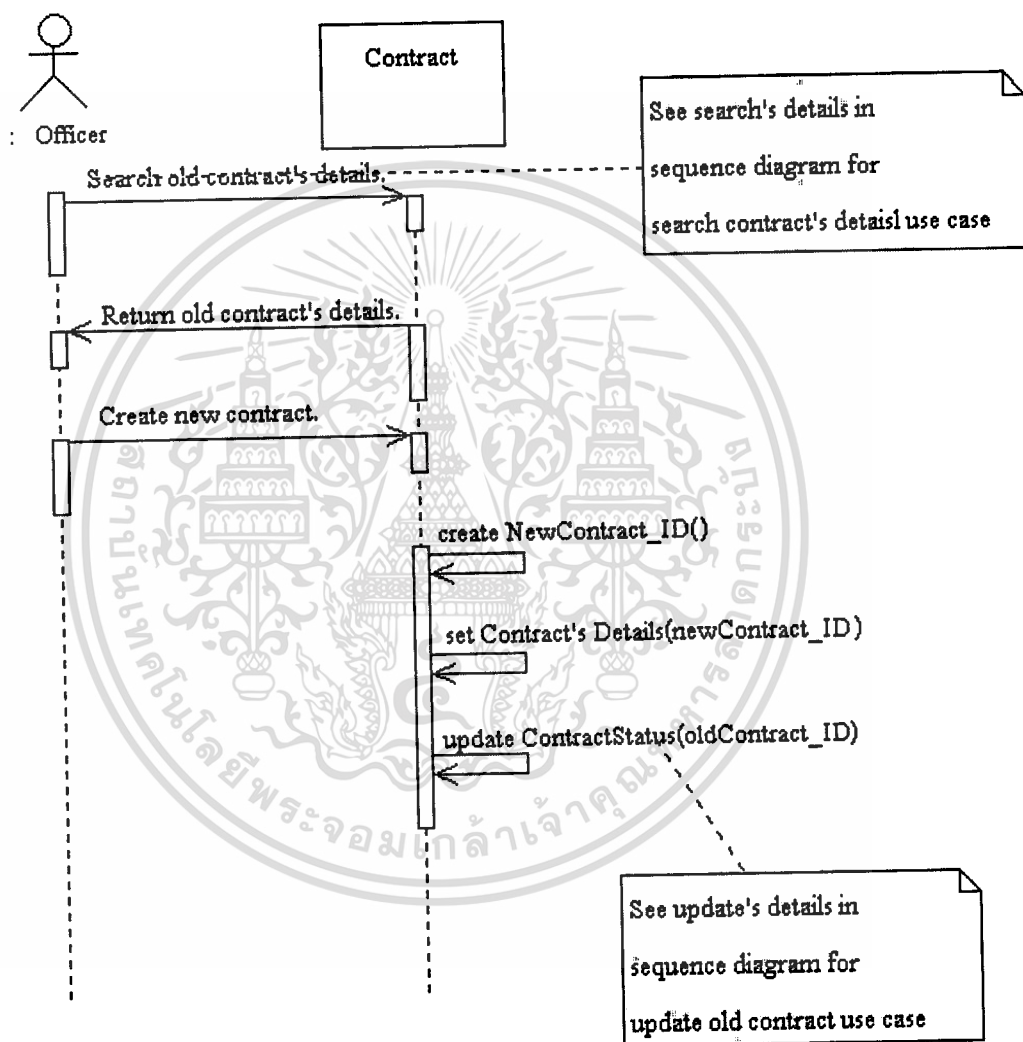
รูปที่ 3.7 แสดง Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้ข้อมูลค่าในการค้นหา)

3.5.5 Sequence Diagram ของ Update Old Contract Use Case



รูปที่ 3. 8 แสดง Sequence Diagram ของ Update Old Contract Use Case

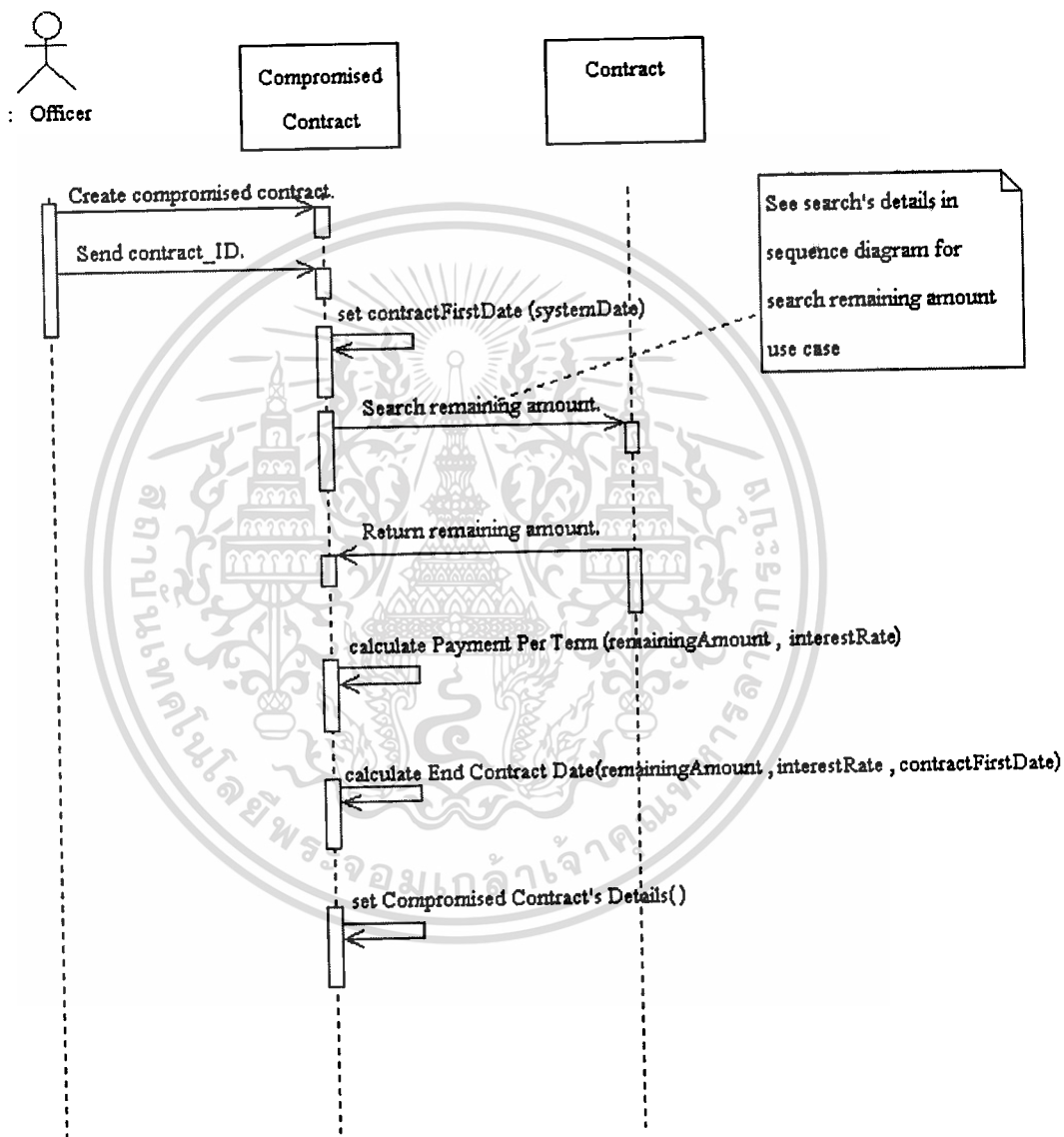
### 3.5.6 Sequence Diagram ของ Create New Contract Use Case



รูปที่ 3. 9 แสดง Sequence Diagram ของ Create New Contract Use Case

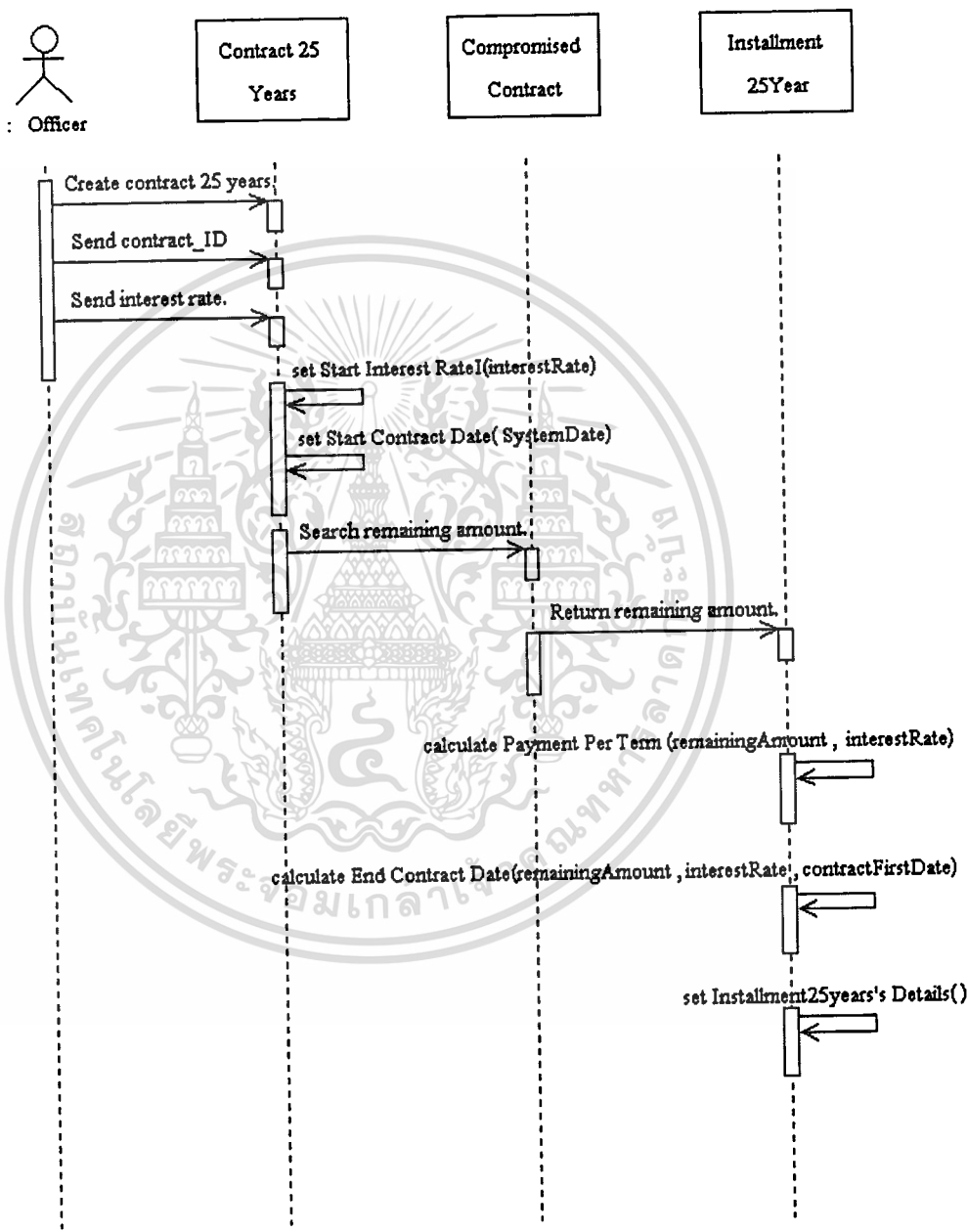


### 3.5.7 Sequence Diagram ของ Create Compromised Contract Use Case



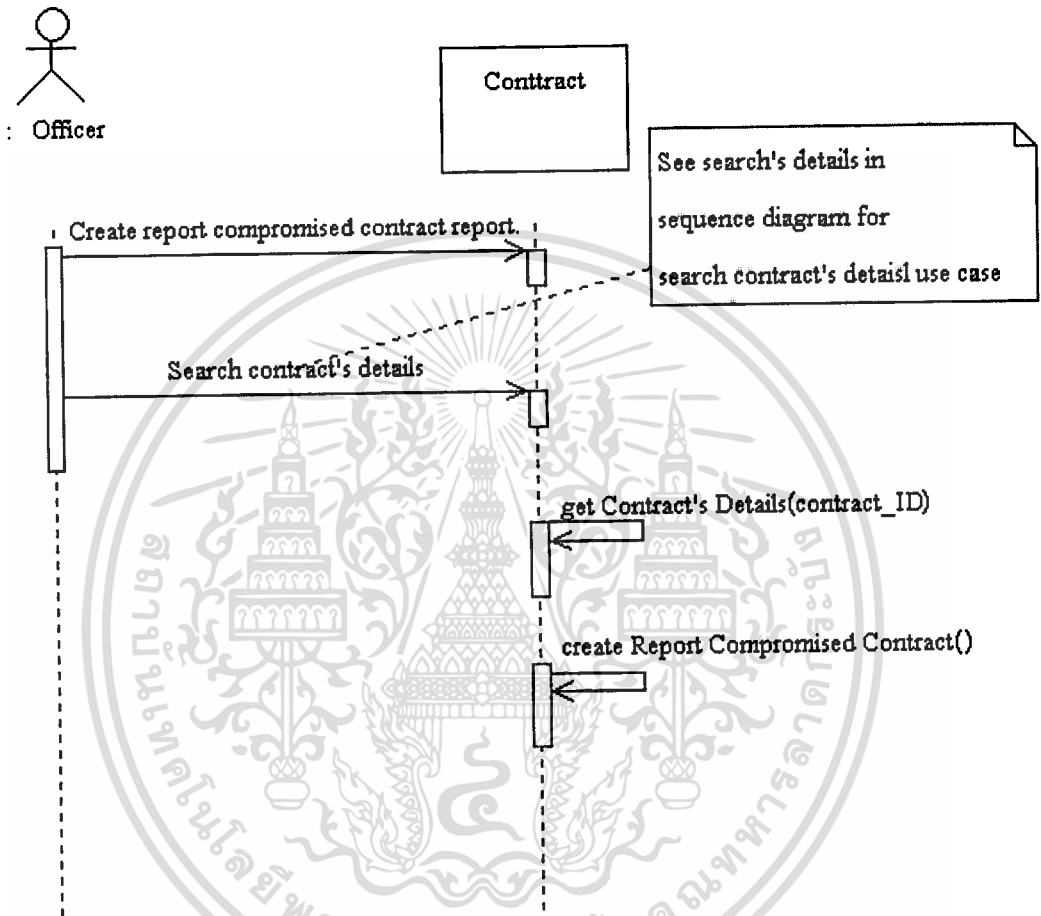
รูปที่ 3. 10 แสดง Sequence Diagram ของ Create Compromised Contract Use Case

3.5.8 Sequence Diagram ของ Create Contract 25 years Use Case



รูปที่ 3. 11 แสดง Sequence Diagram ของ Create Contract 25-Years Use Case

### 3.5.9 Sequence Diagram ของ Create Report Use Case



รูปที่ 3. 12 แสดง Sequence Diagram ของ Create Report Use Case

บทที่ 4

การออกแบบระบบ

ในส่วนการออกแบบ คือ การออกแบบ และปรับปรุงแผนผังคลาสที่สร้างไว้ในขั้นตอนวิเคราะห์ โดยจะต้องมีการกำหนดรายละเอียดต่างๆภายในคลาส เช่น กำหนดรายละเอียดแอตทริบิวต์ของคลาส กำหนดการดำเนินการของคลาส คลาสที่ปรับปรุง และกำหนดรายละเอียดต่างๆแล้วนั้นจะเป็นคลาสที่อยู่ในบิสซิเนสเลเยอร์ (Business Layer) ซึ่งในส่วนของการออกแบบนี้ยังจะต้องมีการออกแบบแผนผังคลาสเพิ่มเติมอีก ซึ่งได้แก่ แผนผังคลาสซึ่งเป็นส่วนที่ติดต่อกับฐานข้อมูลของระบบ คือ คลาสใน แอคเซสเลเยอร์ (Access Layer) และคลาสในส่วนนี้จะทำหน้าที่ติดต่อกับผู้ใช้โดยตรง คือ คลาสที่อยู่ในวิวเลเยอร์ (View Layer )

4.1 กำหนดรายละเอียดในคลาสในบิสซิเนสเลเยอร์

จากในส่วนขั้นตอนการวิเคราะห์ในตอนต้น สามารถจำแนกคลาสได้ดังนี้

ตารางที่ 4. 1 แสดงคลาสของระบบประนอมหนี้

| ชื่อคลาส                  | วัตถุประสงค์                                        |
|---------------------------|-----------------------------------------------------|
| 1. BuildingType           | ประเภทสิ่งปลูกสร้าง                                 |
| 2. BuyContract            | สัญญาเช่าซื้อ                                       |
| 3. BuyDownPayment         | เงินคาวนของการเช่าซื้อ                              |
| 4. CompromisedContract    | สัญญาประนอมหนี้ไม่เกิน 10 ปี                        |
| 5. Contract               | ข้อมูลสัญญา                                         |
| 6. Contract 25 Years      | สัญญาเช่าซื้อไม่เกิน 25 ปี                          |
| 7. Customer               | ข้อมูลลูกค้า                                        |
| 8. InstallPayment 25Years | รายละเอียดการจ่ายเงินแต่ละงวดของสัญญาเช่าซื้อ 25 ปี |
| 9. Location               | รายละเอียดสถานที่ตั้ง                               |
| 10. Model                 | โมเดล                                               |
| 11. PaymentTx             | Transaction การรับชำระเงินแต่ละงวด                  |
| 12. PaymentTxD            | รายละเอียดการชำระเงินแต่ละงวด                       |



| ชื่อคลาส      | วัตถุประสงค์     |
|---------------|------------------|
| 13. Project   | ข้อมูลโครงการ    |
| 14. Quotation | ข้อมูลการขาย     |
| 15. Unit      | ข้อมูลหน่วยอาศัย |

4.1.1 Attribute ของแต่ละ คลาส

โดยจะแสดงชื่อ Attribute พร้อมคำอธิบายของแต่ละ Attribute ดังนี้

ตารางที่ 4. 2 แสดงชื่อพร้อมคำอธิบายแอทริบิวต์ ที่อยู่ในคลาสประนอมหนี้

| ชื่อคลาส               | แอทริบิวต์               | คำอธิบาย                       |
|------------------------|--------------------------|--------------------------------|
| 1. BuildingType        | buildingTypeID           | รหัสประเภทสิ่งปลูกสร้าง        |
|                        | buildingTypeName         | ชื่อประเภทสิ่งปลูกสร้าง        |
| 2. BuyContract         | buyInterestRate          | อัตราดอกเบี้ยต่อการเช่าซื้อ    |
|                        | buyPrincipal             | เงินต้นของการเช่าซื้อ          |
|                        | buyAmount                | จำนวนเงินทั้งหมดของการเช่าซื้อ |
|                        | downPaymentAmount        | จำนวนเงินดาวน์ที่จ่ายเป็นก้อน  |
| 3. BuyDownPayment      | From Term Down-Payment   | งวดที่เริ่มต้นจ่ายเงินดาวน์    |
|                        | toTermDownPaymentt       | งวดที่สิ้นสุดจ่ายเงินดาวน์     |
|                        | pricePerTermDown-Payment | จำนวนเงินดาวน์ที่ชำระแต่ละงวด  |
| 4. CompromisedContract | interestRate             | อัตราดอกเบี้ย                  |
|                        | termPayment              | จำนวนงวดที่ต้องชำระทั้งหมด     |
|                        | installPayment           | จำนวนเงินที่จ่ายต่องวด         |
|                        | startContractDate        | วันที่ทำสัญญาประนอมหนี้        |
| 5. Contract            | contractID               | เลขที่สัญญา                    |
|                        | contractDate             | วันที่ทำสัญญา                  |



| ชื่อคลาส                 | แอททริบิวต์       | คำอธิบาย                                                                                                                                                                                                       |
|--------------------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | contractStatus    | สถานะของสัญญา <ul style="list-style-type: none"> <li>U : ยังใช้งานอยู่</li> <li>A : ยกเลิกเมื่อสิ้นสุดสัญญา</li> <li>T:หยุดเมื่อคำนวณงวดสัญญาใหม่</li> <li>C : เปลี่ยนสัญญา</li> <li>X : ประณอมหนี้</li> </ul> |
|                          | paymentType       | ประเภทการชำระเงิน <ul style="list-style-type: none"> <li>C : ผ่านสช. (เงินสด)</li> <li>B : ผ่านธนาคาร</li> </ul>                                                                                               |
|                          | pricePerTerm      | เงินที่ต้องชำระต่องวด                                                                                                                                                                                          |
|                          | term              | จำนวนงวดที่ต้องชำระ                                                                                                                                                                                            |
|                          | contractType      | ประเภทสัญญา <ul style="list-style-type: none"> <li>R : สัญญาเช่า</li> <li>B : สัญญาเช่าซื้อ</li> <li>L : สัญญาสินเชื่อต่อเติมอาคาร</li> </ul>                                                                  |
|                          | contractIDNew     | เลขที่สัญญาใหม่                                                                                                                                                                                                |
| 6. Contract 25 Years     | startIntarestRate | อัตราดอกเบี้ยเริ่มต้นทำสัญญา 25 ปี                                                                                                                                                                             |
|                          | startContractDate | วันที่ทำสัญญา 25 ปี                                                                                                                                                                                            |
| 7. Customer              | customerID        | เลขที่บัตรประจำตัวประชาชน                                                                                                                                                                                      |
|                          | title             | คำนำหน้าชื่อ                                                                                                                                                                                                   |
|                          | firstName         | ชื่อ                                                                                                                                                                                                           |
|                          | lastName          | สกุล                                                                                                                                                                                                           |
|                          | regisAddress      | ที่อยู่ตามบัตรประชาชน                                                                                                                                                                                          |
|                          | currentAddress    | ที่อยู่ปัจจุบัน                                                                                                                                                                                                |
|                          | telephone         | หมายเลขโทรศัพท์                                                                                                                                                                                                |
| 8.InstallPayment 25Years | formTermPayment   | งวดที่เริ่มต้นจ่ายค่าเช่าซื้อ                                                                                                                                                                                  |
|                          | toTermPayment     | งวดที่สิ้นสุดการจ่าย                                                                                                                                                                                           |
|                          | installPayment    | จำนวนเงินที่จ่ายต่องวด                                                                                                                                                                                         |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต  
 ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ชื่อคลาส       | แอททริบิวต์  | คำอธิบาย                         |
|----------------|--------------|----------------------------------|
| 9. Location    | LocationID   | รหัสสถานที่ตั้ง                  |
|                | LocationName | ชื่อที่ตั้ง                      |
| 10 Model       | ModelID      | รหัสโมเดล                        |
|                | ModelName    | ชื่อโมเดล                        |
| 11 PaymentTx   | PaymentTxID  | รหัสสำหรับแต่ละ Transaction      |
|                | PaymentDate  | วันที่ของ Transaction ที่รับชำระ |
|                | PaymentTerm  | งวดที่ชำระ                       |
|                | ContractID   | เลขที่สัญญา                      |
| 12 PaymentTxID | PaymentTxID  | รหัสสำหรับแต่ละ Transaction      |
|                | AccountID    | รหัสบัญชี เช่น 0520              |
|                | PaymentPrice | จำนวนเงินที่ชำระต่อรายการบัญชี   |
| 13 Project     | ProjectID    | รหัสโครงการ                      |
|                | ProjectName  | ชื่อโครงการ                      |
| 14. Quotation  | ProjectID    | รหัสโครงการ                      |
|                | ModelID      | รหัสโมเดล                        |
| 15. Unit       | ProjectID    | รหัสโครงการ                      |
|                | ModelID      | รหัสโมเดล                        |
|                | HouseNo      | เลขที่บ้าน                       |

4.1.2 Method ของแต่ละ คลาส

โดยจะแสดงชื่อ Method พร้อมคำอธิบายของแต่ละ Method ดังนี้

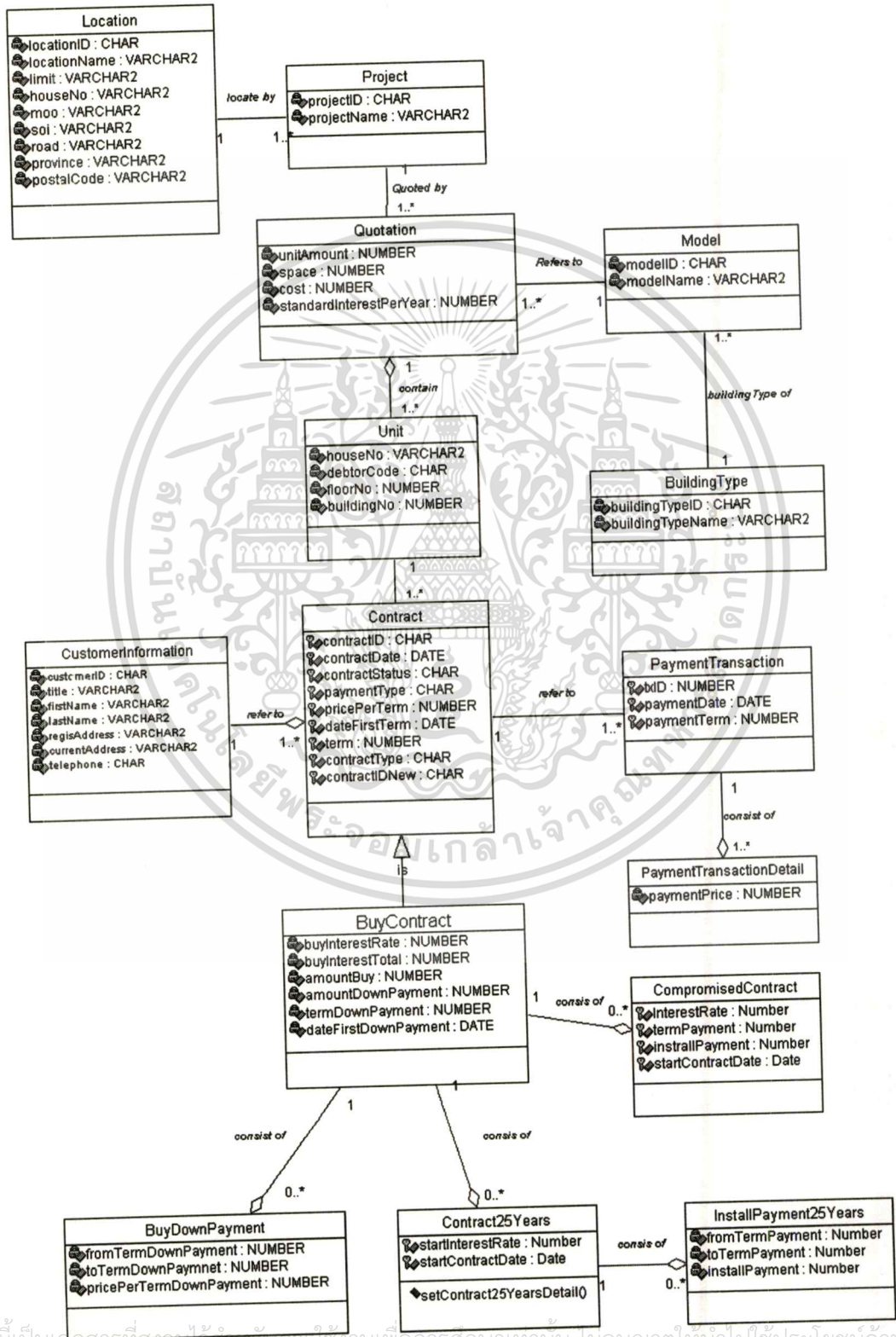
ตารางที่ 4. 3 แสดงชื่อพร้อมคำอธิบายเมทอด ที่อยู่ในคลาสของระบบประนอมหนี้

| ชื่อคลาส               | แอททริบิวต์           | คำอธิบาย                            |
|------------------------|-----------------------|-------------------------------------|
| 1. BuyContract         | getAmountBuy( )       | แสดงข้อมูลราคาเช่าซื้อทั้งหมด       |
| 2. CompromisedContract | calculatePricePerTerm | คำนวณจำนวนเงินที่ต้องจ่ายต่อ<br>งวด |



| ชื่อคลาส                 | แอททริบิวต์                  | คำอธิบาย                           |
|--------------------------|------------------------------|------------------------------------|
| 3. Contract              | getContractDetail            | แสดงรายละเอียดข้อมูลสัญญา          |
|                          | verifyContractID             | ตรวจสอบเลขที่สัญญา                 |
|                          | calculateRemaining<br>Amount | คำนวณยอดเงินคงเหลือ                |
|                          | calculateCreditTerm          | คำนวณจำนวนงวดที่ค้างชำระ           |
|                          | calculateCreditAmount        | คำนวณจำนวนเงินที่ค้างชำระ          |
|                          | createContract               | สร้างสัญญาใหม่                     |
| 4. Contract 25 Years     | CreateContract25Years        | สร้างสัญญาเช่าซื้อ 25 ปี           |
| 5. Customer              | getCustomerDetails           | แสดงข้อมูลลูกค้า                   |
|                          | verifyCustomerName           | ตรวจสอบชื่อลูกค้า                  |
| 6. InstallPayment25Years | calculateInstallTermPayment  | คำนวณจำนวนงวดที่ต้องจ่าย           |
| 7. Model                 | getModelDetail               | แสดงรหัสและชื่อโมเดลทั้งหมด        |
| 8. PaymentTx             | getMaxTerm                   | แสดงงวดสุดท้ายที่ชำระ              |
|                          | getPaymentDate               | แสดงวันที่ชำระจ่ายล่าสุด           |
| 9. PaymentTxID           | getPaymentPrice              | แสดงจำนวนเงินที่ชำระในแต่ละสัปดาห์ |
| 10. Project              | getProjectDetail             | แสดงรหัสและชื่อโครงการ             |
| 11. Unit                 | getUnitDetail                | แสดงรายละเอียดหน่วยอาศัย           |

จาก Attribute และ Method ของ คลาส ที่ได้มา สามารถจะวาดเป็นไดอะแกรมได้รูปที่ 4.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 4.1 แสดง Class Diagram ของระบบประเมินหนี้



4.1.3 การออกแบบความสัมพันธ์

ความสัมพันธ์ของ Class Diagram จากรูปที่ 3.13 นั้น มีความสัมพันธ์แบบ Binary รวมอยู่ด้วย ซึ่งเป็นความสัมพันธ์แบบสองทาง หรือหมายความว่าระหว่างทั้ง 2 คลาส จะรู้ข้อมูลซึ่งกันและกันและในการ Implement นั้นจะต้องสร้าง Attribute ในการอ้างอิงไว้ทั้ง 2 ฝ่าย ซึ่งอาจไม่มีความจำเป็นเลย ทั้งนี้ขึ้นอยู่กับความต้องการของระบบด้วย ส่วนความสัมพันธ์อื่นๆ ในไดอะแกรมมีความชัดเจนแล้ว

จาก Class Diagram ในรูปที่ 3.13 นี้จะมีการกำหนดความสัมพันธ์ใหม่อีกครั้งเพื่อให้เหมาะสมกับการออกแบบ Attribute ต่อไป ความสัมพันธ์แบบ Binary ใดที่ไม่มีความจำเป็นให้ใช้ ความสัมพันธ์ทางเดียวแทน ซึ่งความสัมพันธ์แบบทางเดียวนี้สามารถอธิบายได้คือ คลาส ใด คลาส หนึ่งรู้ข้อมูลอีก คลาส ฝ่ายเดียว

จาก แผนผังต่างๆ ที่วิเคราะห์ออกมาได้นี้ ทำให้สามารถกำหนดขอบเขต และเงื่อนไขต่างๆที่จำเป็น ของระบบประนอมหนี้ของการเคหะแห่งชาติ ได้

และจากขั้นตอนการวิเคราะห์ โดยการสร้างแผนผังต่างๆที่ผ่านมาในข้างต้นนั้น จะนำไปใช้ในการออกแบบระบบเชิงวัตถุต่อไป

4.1.4 การออกแบบ Attribute

เป็นการออกแบบ Attribute ให้เหมาะสมกับการพัฒนา ซึ่งจะหมายถึงการกำหนดประเภทข้อมูล ขนาดของข้อมูล ค่าเริ่มต้นและความสามารถในการเข้าถึงของแต่ละ Attribute อาจรวมถึงเพิ่มหรือลดจำนวน Attribute

ในระบบประนอมหนี้ที่ออกแบบนี้ มีรายละเอียดของ ประเภท ขนาด ของข้อมูลดังต่อไปนี้

ตารางที่ 4. 4 แสดงชื่อพร้อมชนิดข้อมูลแอทริบิวต์ ที่อยู่ในคลาสของระบบประนอมหนี้

| ชื่อคลาส          | แอทริบิวต์             | คำอธิบาย     |
|-------------------|------------------------|--------------|
| 1. BuildingType   | buildingTypeID         | CHAR(2)      |
|                   | buildingTypeName       | VARCHAR2(50) |
| 2. BuyContract    | buyInterestRate        | NUMBER(5,2)  |
|                   | buyPrincipal           | NUMBER(11,2) |
|                   | buyAmount              | NUMBER(11,2) |
|                   | downPaymentAmount      | NUMBER(11,2) |
| 3. BuyDownPayment | From Term Down-Payment | NUMBER(5,0)  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภายในเท่านั้น ไม่ควรเปิดเผยให้บุคคลภายนอกได้รับทราบ  
ไม่ว่ากรณีใดๆก็ตามหากมีให้คัดลอกเอกสารนี้โดยไม่ได้รับอนุญาตจากผู้จัดทำเอกสารฯ จะถือว่าผิดกฎหมาย



| ชื่อคลาส                        | แอททริบิวต์              | คำอธิบาย      |
|---------------------------------|--------------------------|---------------|
|                                 | toTermDownPaymenyt       | NUMBER(5,0)   |
|                                 | pricePerTermDown-Payment | NUMBER(9,2)   |
| <b>4. CompromisedContract</b>   | interestRate             | NUMBER(5,2)   |
|                                 | termPayment              | NUMBER(5,0)   |
|                                 | installPayment           | NUMBER(9,2)   |
|                                 | startContractDate        | DATE(8)       |
| <b>5. Contract</b>              | contractID               | CHAR(8)       |
|                                 | contractDate             | DATE(8)       |
|                                 | contractStatus           | CHAR(1)       |
|                                 | paymentType              | CHAR(1)       |
|                                 | pricePerTerm             | NUMBER(9,2)   |
|                                 | term                     | NUMBER(5,0)   |
|                                 | contractType             | CHAR(1)       |
|                                 | contractIDNew            | CHAR(8)       |
| <b>6. Contract 25 Years</b>     | startIntarestRate        | NUMBER(5,2)   |
|                                 | startContractDate        | DATE(8)       |
| <b>7. Customer</b>              | customerID               | CHAR(13)      |
|                                 | title                    | VARCHAR2(30)  |
|                                 | firstName                | VARCHAR2(20)  |
|                                 | lastName                 | VARCHAR2(30)  |
|                                 | regisAddress             | VARCHAR2(255) |
|                                 | currentAddress           | VARCHAR2(255) |
|                                 | telephone                | CHAR(15)      |
| <b>8.InstallPayment 25Years</b> | formTermPayment          | NUMBER(5,0)   |
|                                 | toTermPayment            | NUMBER(5,0)   |
|                                 | installPayment           | NUMBER(9,2)   |
| <b>9. Location</b>              | locationID               | CHAR(4)       |
|                                 | locationName             | VARCHAR2(50)  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะภายในเท่านั้น ไม่อนุญาตให้เผยแพร่ภายนอกองค์กร

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ชื่อคลาส       | แอทริบิวต์               | คำอธิบาย      |
|----------------|--------------------------|---------------|
|                | houseNo                  | VARCHAR2(9)   |
|                | moo                      | VARCHAR2(2)   |
|                | soi                      | VARCHAR2(3)   |
|                | road                     | VARCHAR2(30)  |
|                | province                 | VARCHAR2(30)  |
|                | postalCode               | VARCHAR2(8)   |
| 10. Model      | modelID                  | CHAR(2)       |
|                | modelName                | VARCHAR2(20)  |
| 11. PaymentTx  | txID                     | NUMBER(15,0)  |
|                | paymentDate              | DATE(8)       |
|                | paymentTerm              | NUMBER(5,0)   |
| 12. PaymentTxD | paymentPrice             | NUMBER(9,2)   |
| 13. Project    | projectID                | CHAR(4)       |
|                | projectName              | VARCHAR2(100) |
| 14. Quotation  | unitAmount               | NUMBER(5,0)   |
|                | space                    | NUMBER(5,2)   |
|                | cost                     | NUMBER(11,2)  |
|                | StandardInterestPer-Year | NUMBER(5,2)   |
| 15. Unit       | houseNo                  | VARCHAR2(9)   |
|                | debtorCode               | CHAR(14)      |
|                | buildingNo               | NUMBER(3,0)   |

#### 4.1.5 การออกแบบ Method

เป็นการออกแบบบริการ หรือหน้าที่ต่างๆที่คลาสนั้นควรจะมีกำหนด Protocol ในการติดต่อสื่อสารระหว่าง คลาส ซึ่งเป็นการกำหนด Visibility Parameter และค่าที่ส่งกลับคืนของ Method และอาจรวมถึงการออกแบบ Algorithm ในแต่ละ Method ด้วยก็ได้ ในบางกรณีอาจต้องมีการเพิ่ม Method ที่อำนวยความสะดวกในการสร้างระบบและการทำงานของระบบด้วย สำหรับ Method ที่เพิ่มขึ้นมาใหม่จะมีคำอธิบายประกอบ อยู่ข้างๆ

การเพิ่มรายละเอียดลงไป ใน คลาส ที่ได้จากการวิเคราะห์ และเป็นการสร้างแบบจำลองบน CASE Tool จึงมีความจำเป็นที่จะต้องใช้สัญลักษณ์ตาม CASE Tool ด้วย โดยส่วนมากแล้ว CASE Tool ที่ใช้นี้จะมีสัญลักษณ์เหมือนกับสัญลักษณ์ที่ใช้ใน UML แต่สัญลักษณ์ในส่วนที่เป็น Visibility นั้นสามารถแสดงรายละเอียดได้ดังต่อไปนี้

แทน Public

แทน Private

แทน Protect

#### 4.2 Access Layer

ในขั้นตอนนี้เป็นการออกแบบ คลาส ที่ทำหน้าที่ในการติดต่อกับฐานข้อมูลจริงของระบบ และรวมถึงการสร้างฐานข้อมูลให้กับระบบอีกด้วย

ในโครงการนี้เป็นการออกแบบระบบฐานข้อมูลเพื่อให้เหมาะสมกับระบบฐานข้อมูลแบบ Relational ของ Oracle ซึ่งเป็น DBMS ที่ใช้ในระบบ

เนื่องจากระบบฐานข้อมูลที่ใช้ในการพัฒนาระบบนี้เป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ส่วนการวิเคราะห์และออกแบบที่ผ่านมาเป็นการออกแบบเชิงวัตถุ ซึ่งทฤษฎีทั้งสองส่วนนี้แตกต่างกัน เพราะระบบฐานข้อมูลเชิงสัมพันธ์ นั้นเป็นระบบฐานข้อมูลที่สามารถเข้าถึงข้อมูลได้โดยตรง (ในที่นี้จะไม่นับถึงสิทธิของผู้ใช้งาน) แต่ทฤษฎีของการออกแบบเชิงวัตถุ นั้นโดยปกติแล้ว จะไม่สามารถเข้าถึงข้อมูลได้โดยตรง จะเข้าถึงข้อมูลผ่านทาง Interface ของ ออบเจกต์

ในระบบฐานข้อมูลเชิงสัมพันธ์ นั้น Schema ต่างๆ จะถูกสร้างเป็น Table ซึ่ง Table จะประกอบไปด้วยแถว (Row) และหลัก (Column) และแต่ละ Column ก็จะมีชื่อและประเภทของข้อมูลประเภทเดียวกัน

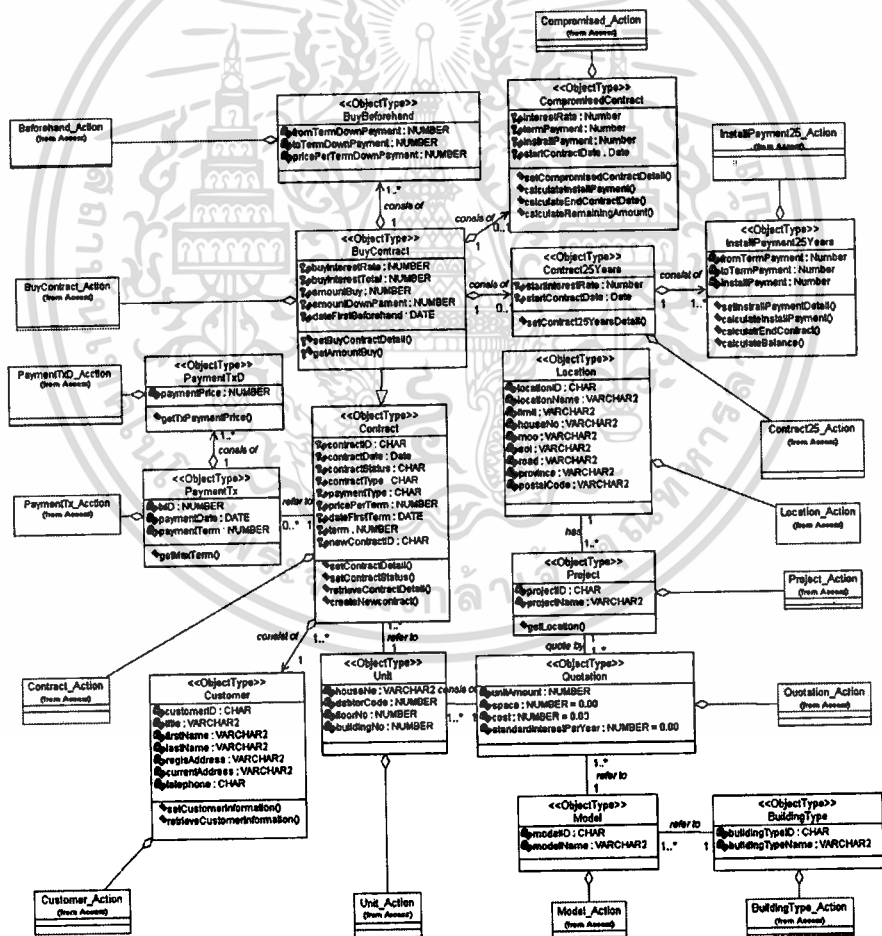
ส่วนในรูปแบบของ ออบเจกต์ นั้นอาจเทียบได้ว่าคลาสนี้ 1 คลาส เป็น Table, Object 1 Object เป็น Row ส่วน Method เป็น Store Procedure ในระบบฐานข้อมูลเชิงสัมพันธ์

จากความแตกต่างดังกล่าวนี้ จึงต้องมีการปรับแบบจำลองที่ออกแบบมาให้มีความเหมาะสมกับระบบฐานข้อมูลที่จะสร้าง โดยอิงตามลักษณะของระบบฐานข้อมูล โดยจะมีการปรับ คลาส ไม่ว่าที่มันใด ๆ ทั้งนั้น ออกใหม่ให้สอดคล้องกับเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ที่ได้จากการออกแบบมาแล้วนั้น กำหนดความสัมพันธ์ของ คลาส เหล่านั้นและทำการเพิ่ม Attribute ที่ทำหน้าที่เป็น Primary Key และ Foreign Key เพื่อเทียบให้เป็น Table ในระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งในส่วนของการสร้างฐานข้อมูลนี้จะกล่าวต่อไปในหัวข้อการสร้างฐานข้อมูล

ซึ่งในการออกแบบคลาสในแอคเซสเลเยอร์นั้น เพื่อใช้เป็นส่วนในการติดต่อกับฐานข้อมูลของระบบนั้น จะเป็นการสร้าง Mirror CLass กับ คลาสที่อยู่ในบิสซิเนสเลเยอร์ โดยการจะตั้งชื่อคลาสในแอคเซสเลเยอร์ให้สอดคล้องกับ คลาสในบิสซิเนสเลเยอร์ กล่าวคือ ชื่อคลาสในแอคเซสเลเยอร์จะเหมือนใน บิสซิเนสเลเยอร์ แต่จะต่อท้ายด้วย \_Action เพื่อแสดงการดำเนินการ ดังแสดงตัวอย่างความสัมพันธ์ในรูปที่ 4.2

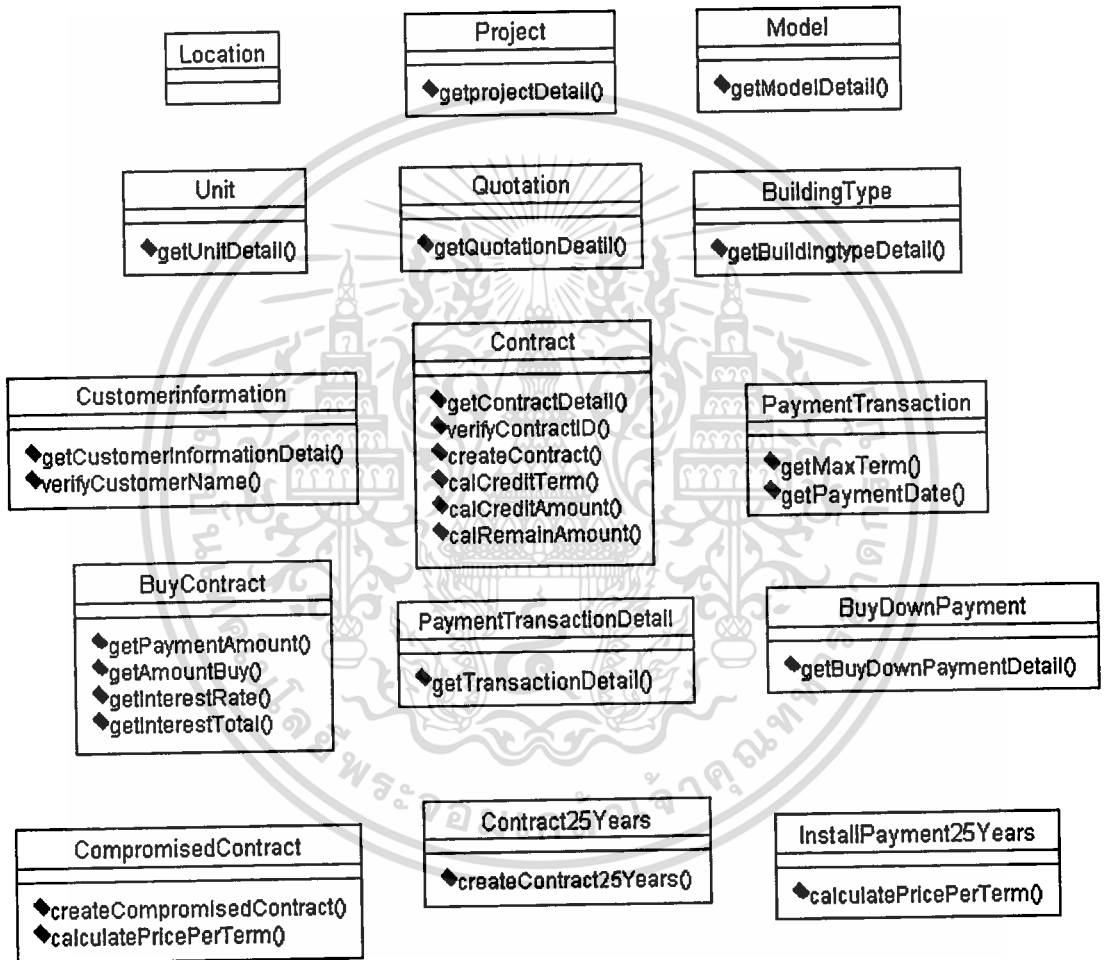


รูปที่ 4.2 แสดงความสัมพันธ์ระหว่าง Business Layer และ Access Layer

ส่วน ของ คลาส ใน Access Layer นี้ ในการสร้างระบบจะสร้างเป็น Store Procedure ไว้ใน DBMS และเพื่อยังรักษาทฤษฎีของ Object-Oriented ไว้ ดังนั้น คลาสในส่วน ของ Access Layer

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้จึงประกอบไปด้วย ส่วนของ Method ต่างๆ ที่จะใช้ในการติดต่อกับ ฐานข้อมูลจริง เท่านั้น โดยจะไม่มีในส่วนของ Attribute ของคลาส ดังแสดงตัวอย่าง คลาสใน Access Layer ดังรูปที่ 4.3



รูปที่ 4. 3 แสดงคลาสใน Access Layer

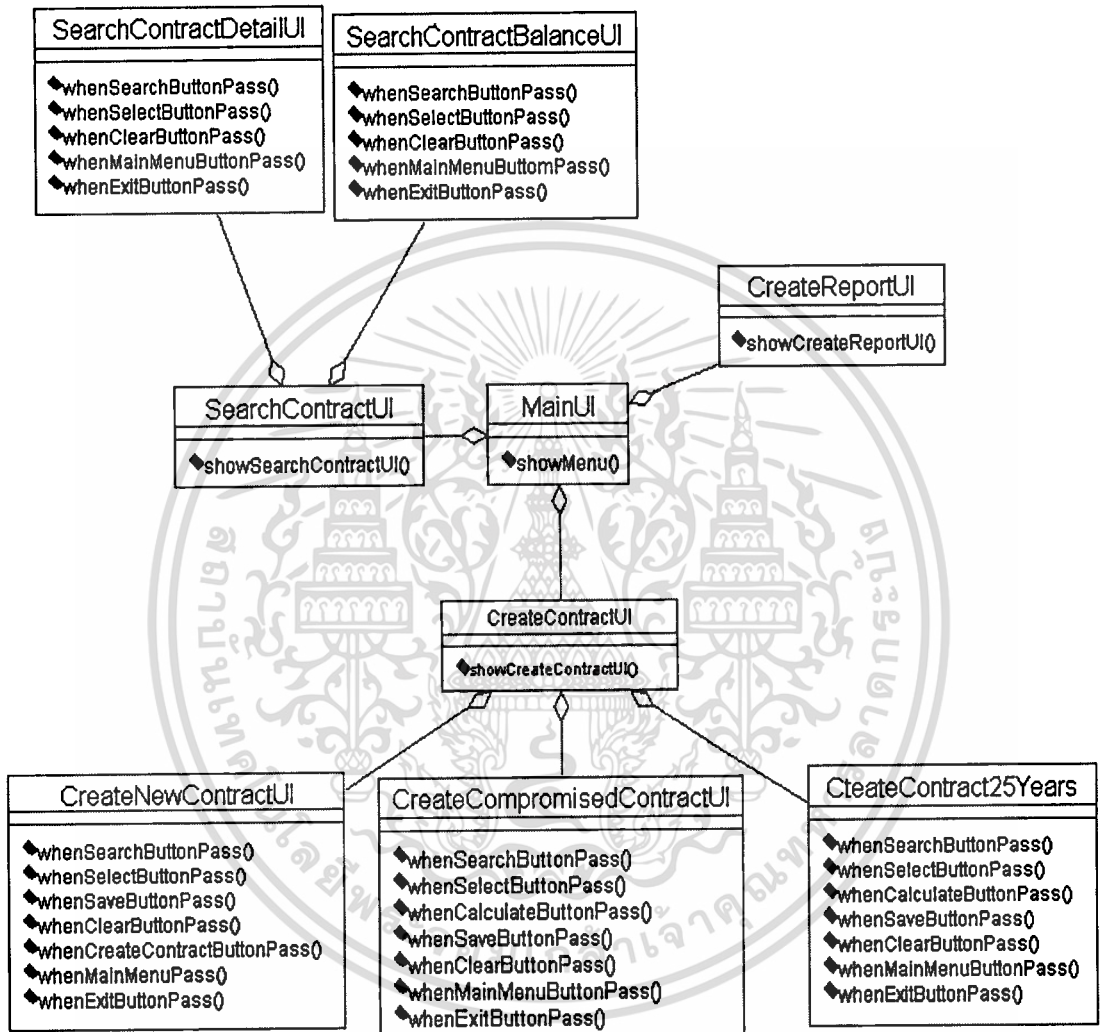
#### 4.3 View Layer

เป็นการออกแบบ คลาส ที่ทำหน้าที่เป็นส่วนที่ติดต่อกับผู้ใช้งาน ซึ่งเป็น Interface ระหว่างผู้ใช้งานกับระบบ หรือเป็น User Interface (UI) ของระบบ และเมื่อได้ คลาสที่ต้องการแล้ว นำ คลาส เหล่านั้นมาพัฒนาต่อ เป็นในรูปแบบของ Graphic User Interface (GUI)

ในการออกแบบจะเป็นการกำหนด คลาส ที่ทำหน้าที่เป็น Interface ของระบบ ซึ่งอาจจะ

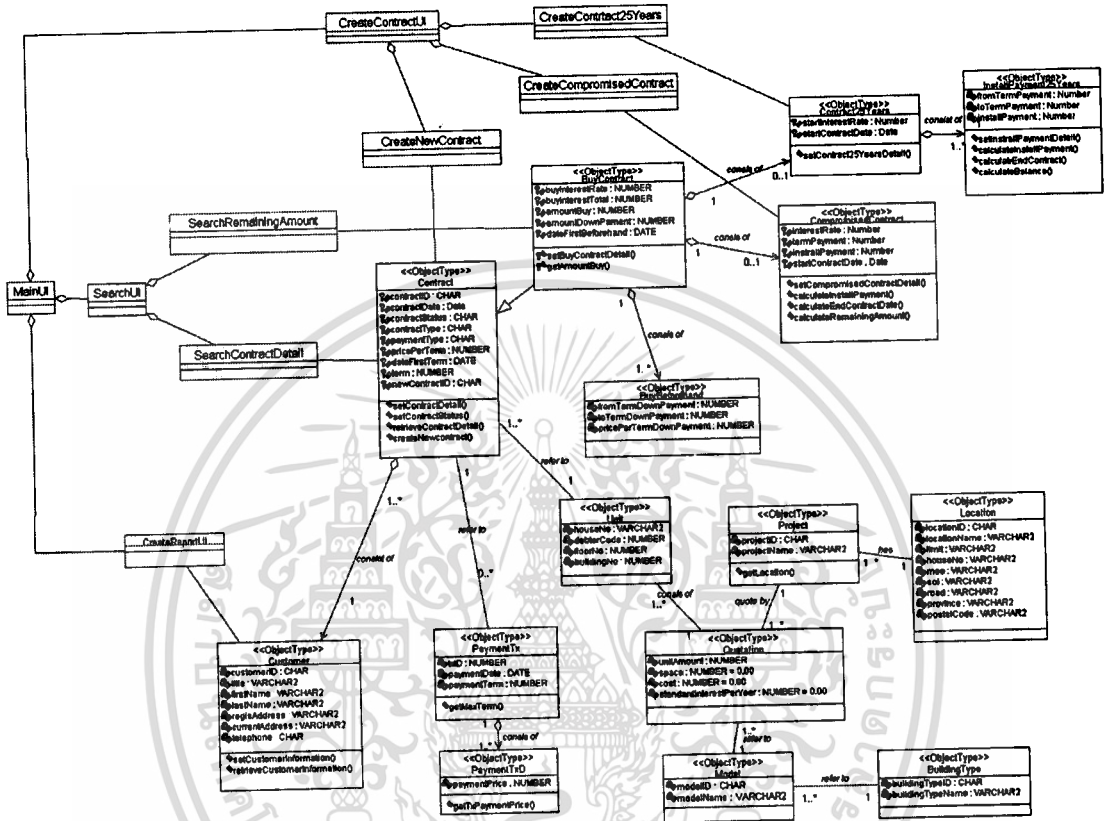


ดังนั้นสามารถแสดงโคแอมพลาสที่ทำหน้าที่เป็น User Interface ได้ ดังรูปที่ 4.4นี้คือ



รูปที่ 4.4 แสดงคลาสใน View Layer

ส่วนในรูปที่ 4.5 จะเป็นการแสดงความสัมพันธ์ระหว่าง Business Layer และ View Layer



รูปที่ 4.5 แสดงความสัมพันธ์ระหว่าง Business Layer และ View Layer

## 4.4 การสร้างระบบฐานข้อมูล

### 4.4.1 การติดตั้ง DBMS

DBMS ที่ใช้เป็น Oracle Server เวอร์ชัน 8 ซึ่งก็เปรียบเสมือนเป็นการติดตั้งโปรแกรมแบบทั่วไปเหมือนกับโปรแกรมอื่นๆ (สามารถหาได้ตามคู่มือ DBMS ที่มีได้) หลังจากติดตั้ง DBMS เรียบร้อยแล้ว ก็ทำการสร้าง Database ให้กับระบบงาน ซึ่งคล้ายกับเป็นการกำหนดขนาดของ Database เพื่อให้สามารถรองรับส่วนต่างๆ ต่อไป (เช่น Table, Store Procedure เป็นต้น) ต่อจากการสร้าง Database จะเป็นการสร้าง User ของที่ใช้งานกับ Database (พร้อมกับการกำหนดสิทธิ์ที่เหมาะสม) เพื่อเป็น User ของระบบงาน

จากนั้นจะเป็นการกำหนดทางเชื่อมต่อ (หรือที่เรียกว่า Host String) เพื่อให้ผู้พัฒนาเข้าถึง Database ได้ โดยใช้เครื่องมือที่มีมาให้คือ Net Easy Config ช่วยในการกำหนด และขณะนี้ Database ในฝั่ง Server พร้อมรับบริการ แต่ยังไม่มีโครงสร้างฐานข้อมูลของระบบอยู่

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.2 การสร้าง Table ลงใน Database เพื่อเก็บข้อมูลของระบบ

การสร้าง Table ของระบบจะอาศัย Rational Rose 2000 เป็นเครื่องมือช่วยในการสร้างฐานข้อมูล ซึ่งปกติจะต้องมาสร้างฐานข้อมูลขึ้นมาด้วยตัวเอง ซึ่งเครื่องมือนี้จะอาศัยแบบจำลองที่มีการออกแบบไว้เป็นข้อมูลในการสร้าง Table บนฝั่ง Server หรือเรียกว่า Forward Engineering ซึ่งมีวิธีการดังนี้

##### 4.4.2.1 การกำหนด Model สำหรับการสร้าง Table

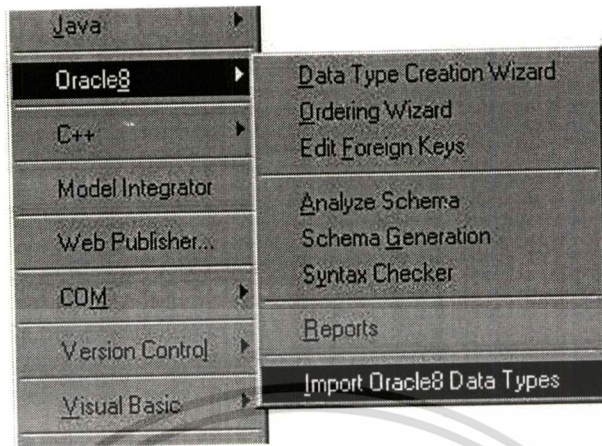
เนื่องจากการพัฒนาครั้งนี้มีการ Implement บนระบบฐานข้อมูลที่เป็นแบบ Relational จึงต้องมีการ Map จาก Class ที่ได้รับการออกแบบซึ่งเป็นรูปแบบของ Object ไปเป็น Class ที่สนับสนุนการสร้าง Schema แบบ Relational บน Database จะเป็นลักษณะ Class ที่ไม่มี Method และจะต้องมีการกำหนดลักษณะ Class ที่ต้องการสร้าง Schema ให้เหมาะสมในการสร้าง

การ Map มี 2 วิธีคือ ใช้ Wizard (Data Type Creation Wizard) ช่วย และการสร้าง Class ขึ้นมาใหม่ด้วยตัวเอง และการ Map ในที่นี้จะอาศัย Wizard เป็นเครื่องมือช่วยในการ Map เพื่อความสะดวกและรวดเร็ว รวมถึงการเก็บข้อมูลการกำหนดใน Class Diagram ได้ดีกว่า ส่วนวิธีการ Map ใน Software สามารถทำได้หลายรูปแบบมากมาย ในส่วนที่นำเสนอจะเป็นเพียงรูปแบบเดียวเท่านั้น

การใช้ Wizard ใน Rational Rose ช่วยในการ Map Wizard จะทำการ Map จาก Class ที่มี Stereotype เป็น Object Types หรือ Relational Tables เท่านั้น

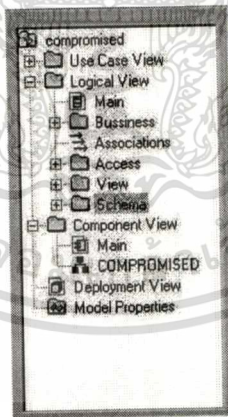
ก่อนที่จะเริ่มต้นนั้นควรที่จะกำหนด Class Diagram ให้มีความละเอียดเพียงพอเสียก่อน เพราะหากทำการ Map ทาง Wizard จะทำการ Map คุณสมบัติที่กำหนดไว้ใน Class Diagram มาด้วย

1. เริ่มต้นจากการ Map Class โดยนำ Class Diagram ของขั้นการออกแบบมาเป็นจุดเริ่มต้น หากกรณีที่ใช้ Framework ของ Oracle8 อยู่แล้วให้ข้ามขั้นนี้ไป แต่กรณีที่ไม่ได้ใช้ Framework ของ Oracle8 ต้องมีการ Import ชนิดข้อมูลของ Oracle8 ที่ Rational Rose รู้จักก่อน



รูปที่ 4.6 แสดงการ Import Oracle8 Data Type

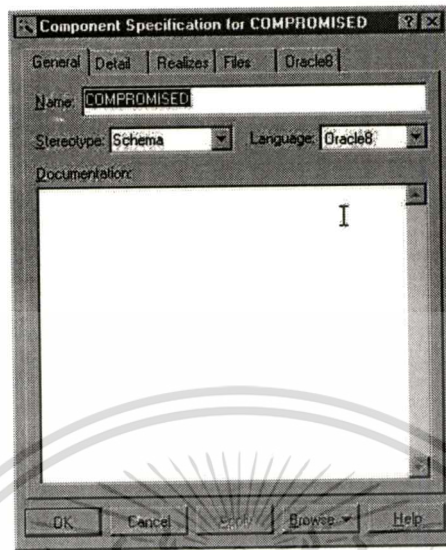
2. ขั้นตอนมาสร้าง Component พร้อมกับตั้งชื่อ แสดงไว้ดังรูปที่ 4.6 และชื่อของ Component จะใช้แทน Owner ของ Table ในการสร้าง Schema เพราะโดยปกติการสร้าง Table ใน Oracle นั้นจะสร้างตามสิทธิของ User และเมื่อสร้าง Table ได้ User ที่เป็นผู้สร้างจะถือเป็น Owner



รูปที่ 4.7 แสดงการสร้าง Component พร้อมกับการกำหนดชื่อ

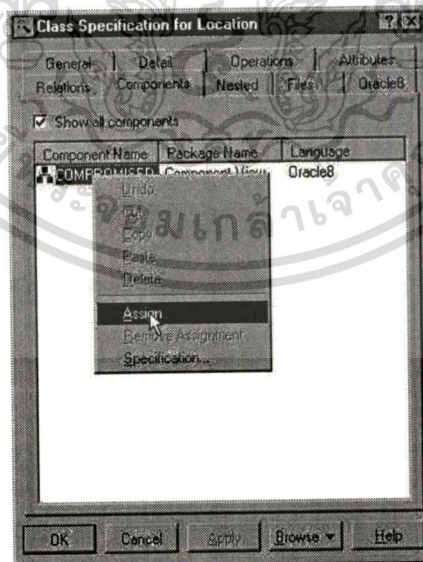
3. กำหนดคุณสมบัติของ Component (Component Specification) โดยให้ Language เป็น Oracle8 (แล้ว Apply) หลังจากการ Apply กำหนด Stereotype ให้เป็น Schema





รูปที่ 4. 8 แสดงการกำหนดคุณสมบัติของ Component (Component Specification)

4. หลังจากนั้นเลือก Tab Sheet (ยังอยู่ใน Component Specification) Realizes แล้วเลือก Class ที่ต้องการให้ Map โดยการ Assign เมื่อ Assign ได้จะขึ้นเครื่องหมายถูกสีแดงทับ Class ที่ Assign ได้



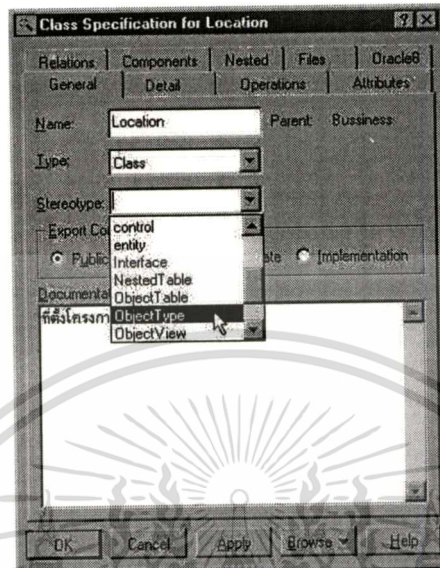
รูปที่ 4. 9 แสดงการ Assign Class ที่ Map

5. จากนั้นมาที่ Class Specification ของ Class ที่ถูก Map สังเกตว่ามี Tab Sheet ชื่อ Oracle8ปรากฏเพิ่มขึ้นมา กำหนด Stereotype เป็น ObjectType ซึ่งขั้นตอนนี้หมายความว่าให้ Class นี้มีชนิดเป็น ObjectType

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

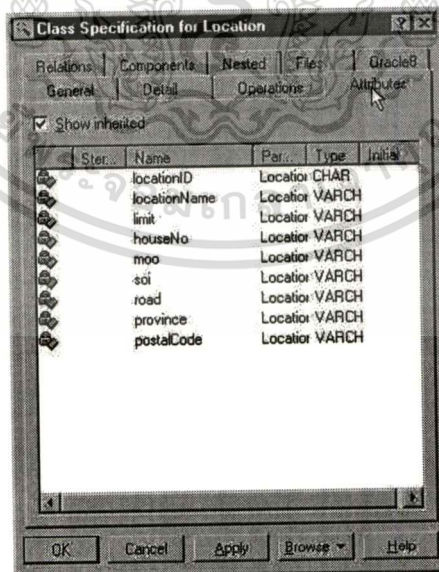
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่โดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 4. 10 แสดงการกำหนด Stereotype ของ Class เป็น ObjectType

6. เปลี่ยนมาที่ Tab Sheet ชื่อ Attributes แล้ว กำหนด Attribute Specification เพื่อกำหนด ชนิดของข้อมูลของ Attribute ซึ่งต้องเป็นข้อมูลประเภทที่ Oracle รู้จักด้วย (Scalar Type) และกำหนด Attribute Specification ของ Attribute ทุกตัว

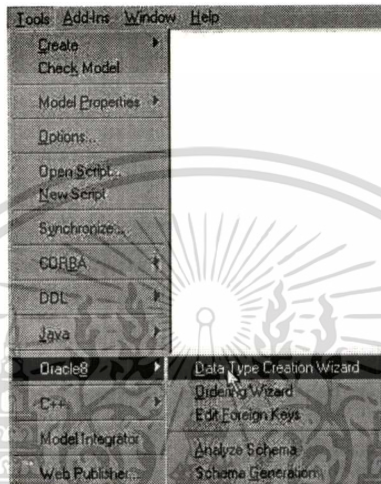


รูปที่ 4. 11 แสดงการกำหนดชนิดข้อมูลของ Attribute

- 7 จากนั้นทำซ้ำขั้นตอน 5 จนถึงขั้นตอนที่ 6 เพื่อเป็นการกำหนด Class Specification ที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อระบุการกำหนด Class Diagram ที่จะนำมา Map เท่านั้น ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

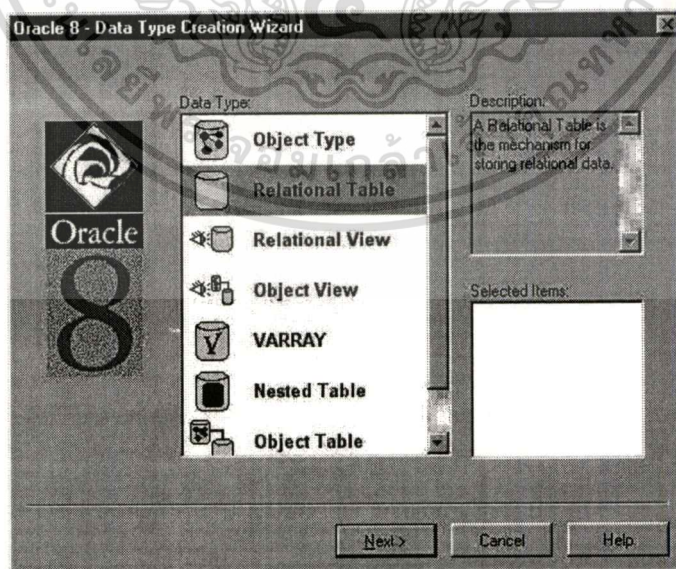
หลังจากนั้นใช้ Wizard ของ Rational Rose 2000 ช่วยในการสร้าง Class เหล่านี้ขึ้นมาใหม่ โดยมีขั้นตอนต่างๆ ดังนี้

1. เลือกที่เมนูบาร์ Tool -> oracle -> Data Type Creation Wizard ตามลำดับ ดังรูป



รูปที่ 4. 12 แสดงการสร้างฐานข้อมูลโดยใช้ Wizard

2. เลือกให้ฐานข้อมูลเป็น Relational Database ดังรูป

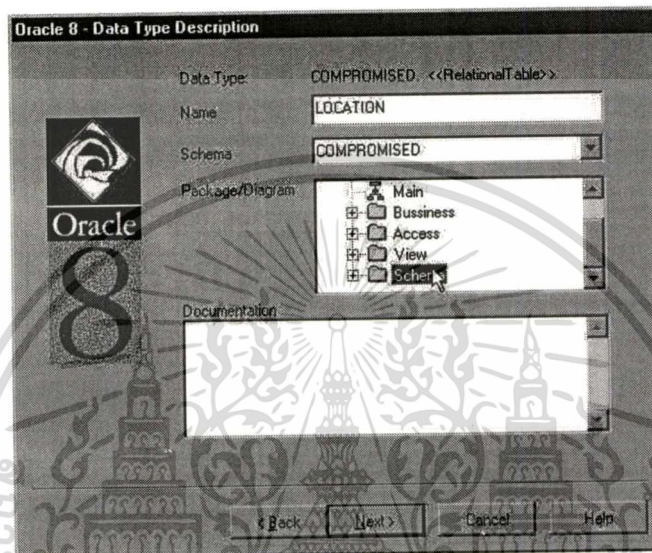


รูปที่ 4. 13 แสดง Wizard เลือกฐานข้อมูลเป็น Relational Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

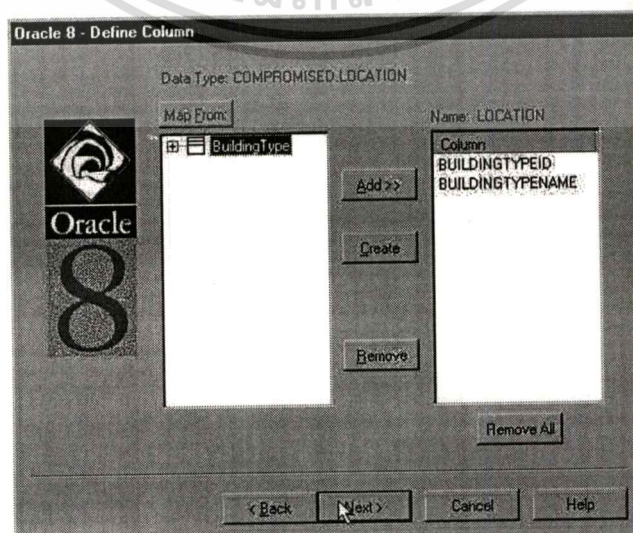


3. ใส่ชื่อของ Class ที่จะสร้างขึ้นใหม่ พร้อมกับเลือก Schema ที่มีให้ในช่อง Schema และระบุตำแหน่งที่ให้สร้าง Class ใหม่นี้ เก็บไว้ในที่ใด ในช่อง Package/Diagram ได้

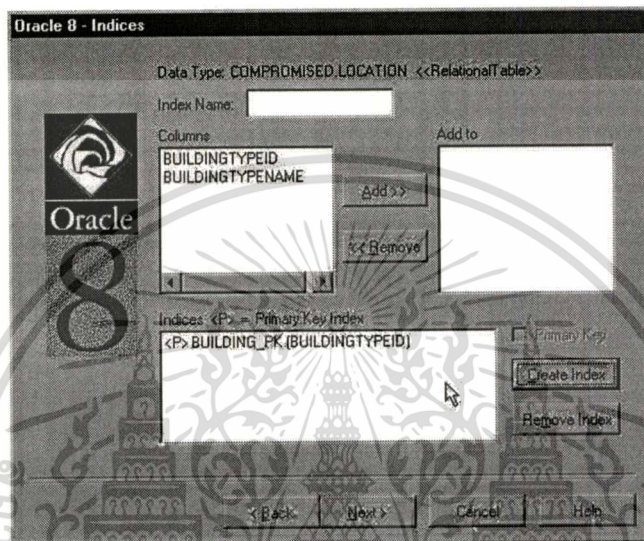


รูปที่ 4. 14 แสดง Wizard ในการ Map (Data Type Description)

4. และหน้าจอต่อไปจะเป็นการกำหนด Attribute ให้แก่ Class ที่จะสร้างใหม่ ซึ่งสามารถทำได้ทั้งสร้างใหม่เองหรือทำการ Map มาจากสิ่งที่มีอยู่แล้ว และในที่นี้จะเป็นการสร้างจากสิ่งที่มีอยู่แล้วคือ Map Form

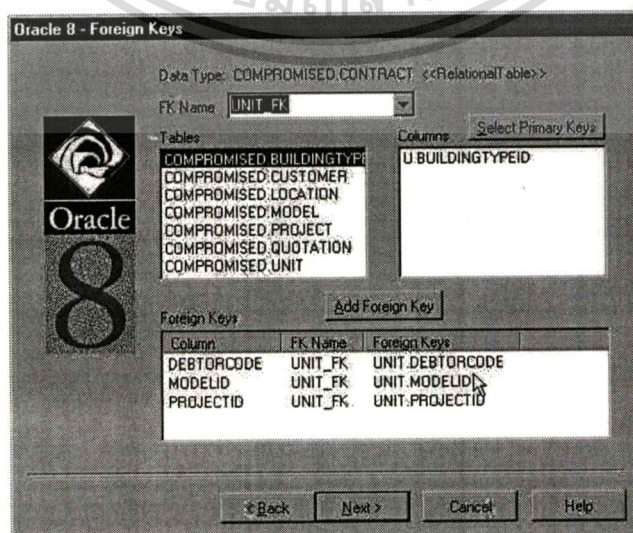


5. กำหนด Index ซึ่งสามารถมีได้หลาย index และอาจกำหนดให้ Index ทำหน้าที่เป็น Primary Key ได้ ชื่อของ Index ที่สร้างขึ้น บน Oracle DBMS จะทำหน้าที่เป็น Constraint ของ Column ดังนั้นการตั้งชื่อของ Index จะต้อง Unique



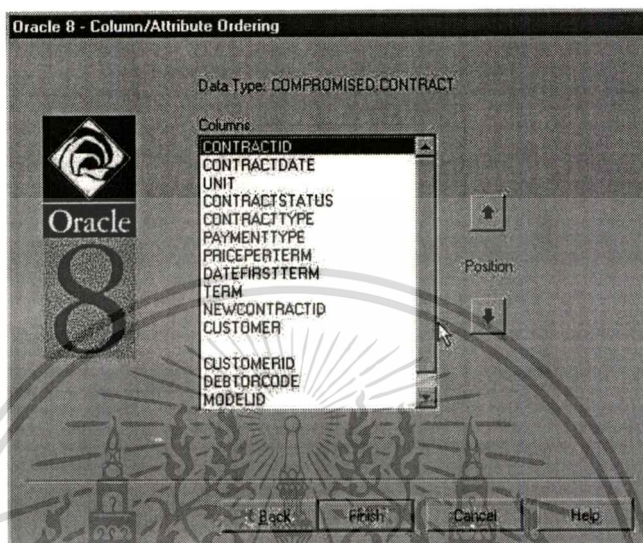
รูปที่ 4. 16 แสดง Wizard ของการ Map (การกำหนด Indices / Primary Key)

6. กำหนด Foreign Key โดยกำหนดชื่อ Foreign Key เมื่อเลือก Foreign Key แล้วจะมีชื่อของ Column นั้น ปรากฏอยู่ในช่อง Column ซึ่งอาจมีตัวอักษร P และ U นำหน้าชื่อ Column ซึ่ง P แทนด้วย Primary Key และ U แทนด้วย Column ที่ Unique





7. ขั้นต่อไปเป็นการเรียงลำดับของ Attribute ของ Class ที่สร้างใหม่ และเป็นขั้นตอนสุดท้ายในการ Map



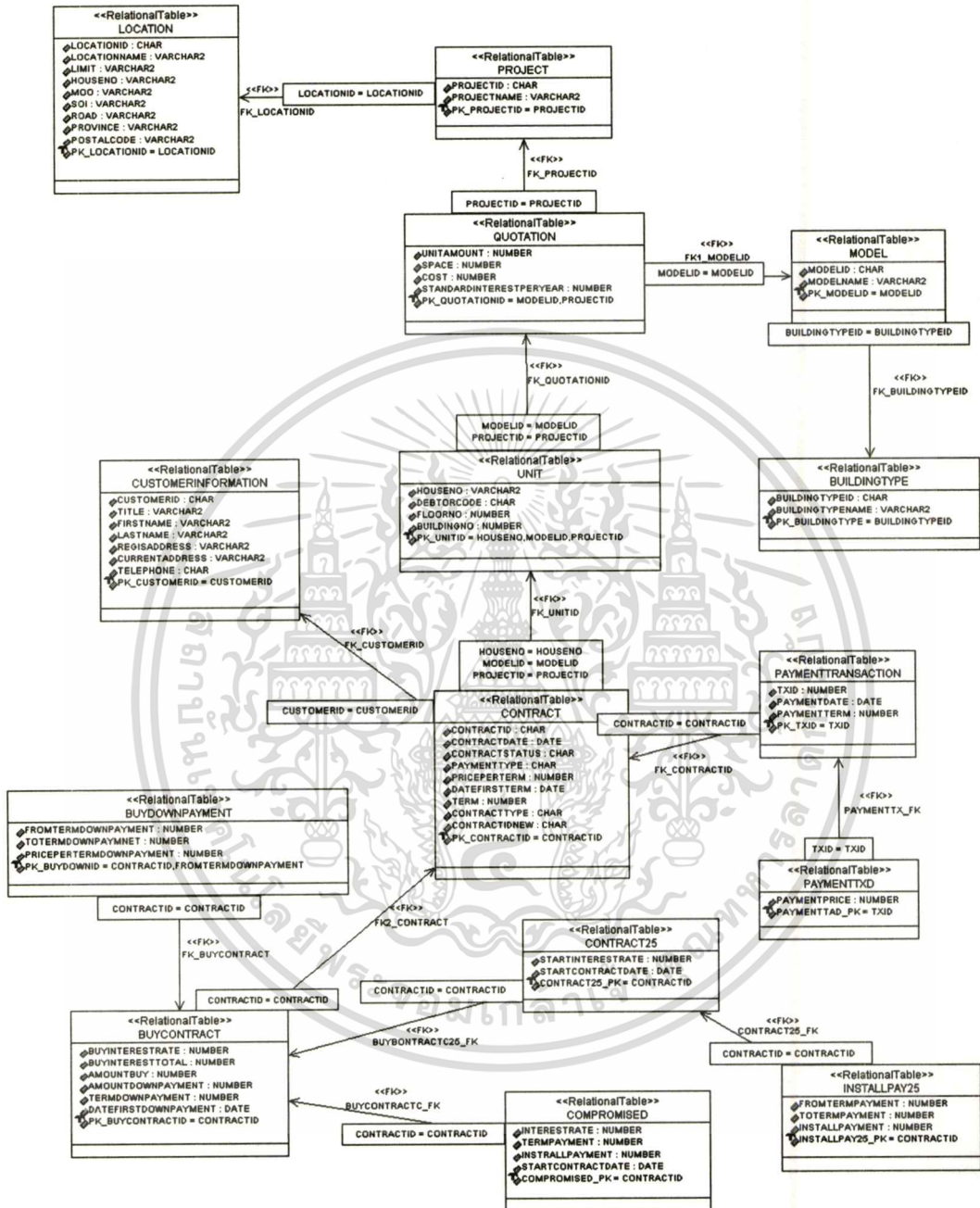
รูปที่ 4. 18 แสดง Wizard ของการ Map (การเรียงลำดับ Attribute)

8. ทำซ้ำจนกว่าจะได้โครงสร้างของ Class ที่ต้องการ

เนื่องจากการใช้ระบบฐานข้อมูลเป็นแบบ Relational จึงมีความจำเป็นในการปฏิบัติตามหลักเกณฑ์และทำให้ได้ประสิทธิภาพสูงสุดในสภาพแวดล้อมที่ทำงานอยู่ และสิ่งหนึ่งที่ควรปฏิบัติคือการ Normalization กับโครงสร้างของ Table ในระบบ เพื่อมิให้เกิดปัญหาความซ้ำซ้อน ของข้อมูลในระบบ

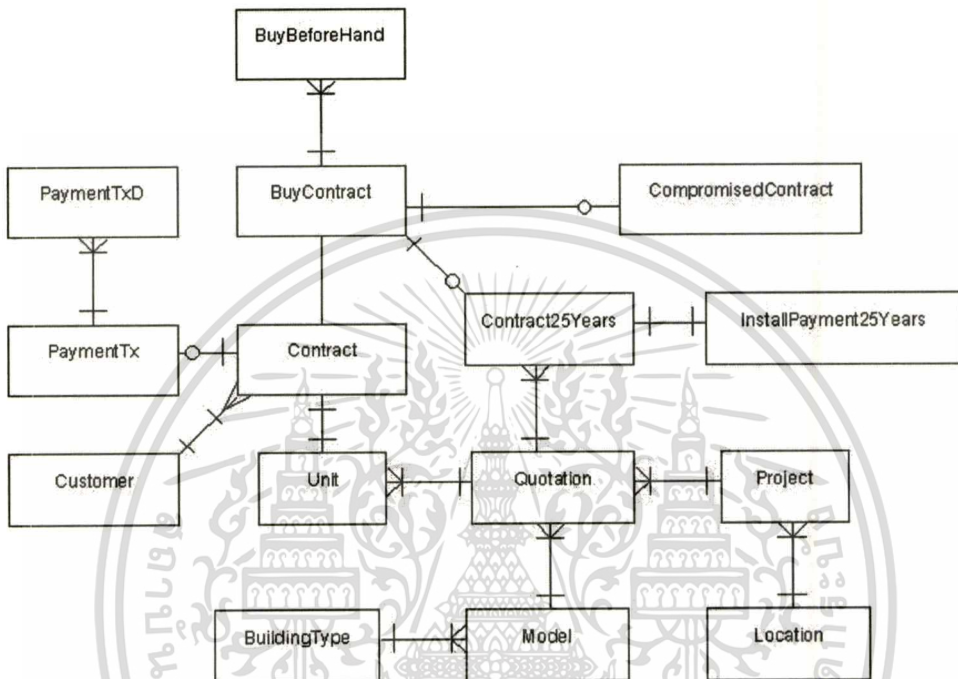
หลังจากผลที่ได้ นั้นมีความจำเป็นอย่างมากที่จะตรวจสอบว่าโครงสร้าง Class ที่ได้นั้น เกิดปัญหาเรื่องการซ้ำซ้อนของข้อมูลหรือไม่ หากยังก็จัดการทำ Normalize ซึ่งอาจเป็นการเพิ่มหรือลด Class ที่ Map ได้และ/หรือ Attribute ที่ Map ได้เช่นกัน หลังจากทำการ สร้าง Class ตามขั้นตอนข้างต้น แล้วจะได้ โค้ดแอมของ schema ดังรูปที่ 4.18





รูปที่ 4. 19 แสดง Schema Diagramของระบบประนองหนี้

หลังจากที่ได้ schema ออกมาแล้วนั้น จะต้องมีการตรวจสอบเพิ่มเติมว่า โครงสร้าง Class ที่ได้นั้น มีความซ้ำซ้อนหรือไม่ โดยนำโครงสร้างที่ได้ มาเขียนเป็น Entity Relationship ได้ดังนี้



รูปที่ 4. 20 แสดง Entity Relationship ของระบบประนอมหนี้

เนื่องจากในฐานข้อมูลของระบบประนอมหนี้ มีลักษณะฐานข้อมูลแบบ ฐานข้อมูลเชิงสัมพันธ์ จึงจำเป็นต้องพิจารณาต่อไปว่า Class Diagram ที่ได้มานี้ เหมาะสมกับฐานข้อมูลเชิงสัมพันธ์หรือไม่ นั่นคือ ตาราง ที่ได้ออกมานั้นจะต้องเป็น Third-Normal form ตามทฤษฎีการวิเคราะห์และออกแบบระบบในรูปแบบ Relational ซึ่งหากพิจารณาแล้ว พบว่า ตารางที่ได้ออกมานั้นอยู่ในระดับ 1NF แล้ว แต่จะต้องมีการพิจารณาต่อไปในส่วนของ 2NF และ 3NF

**1NF :** ทุก ๆ attribute ใน Table ต้องเป็น atomic value (data item is atomic value)

คือ attribute แต่ละตัวใน Table จะแสดงถึงค่าเพียงค่าเดียว มีความหมายบ่งชี้ถึงสิ่งๆ เดียว

**2NF :** จะต้องเป็น 1NF และ ทุกๆ attribute ใน Table ที่ไม่ได้เป็น Primary Key จะต้องขึ้น อยู่กับ Primary Key อย่างสมบูรณ์ (all nonkey is full depend on Primary Key)

**3 NF** หมายถึง ทุกค่าในตารางที่ไม่ได้เป็นคีย์ในตารางจะไม่มี การขึ้นต่อกันระหว่างค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้แบบเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ภายในตารางคือจะไม่มี Transitive FD ในตาราง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากการพิจารณา Attribute แต่ละตัวพบว่า ตารางที่ได้ทำการ Map จาก Class ที่ออกแบบในเชิงวัตถุ มาเป็นตารางซึ่งเป็นโครงสร้างฐานข้อมูลแบบ Relational นี้เป็น 3NF เป็นอย่างน้อย

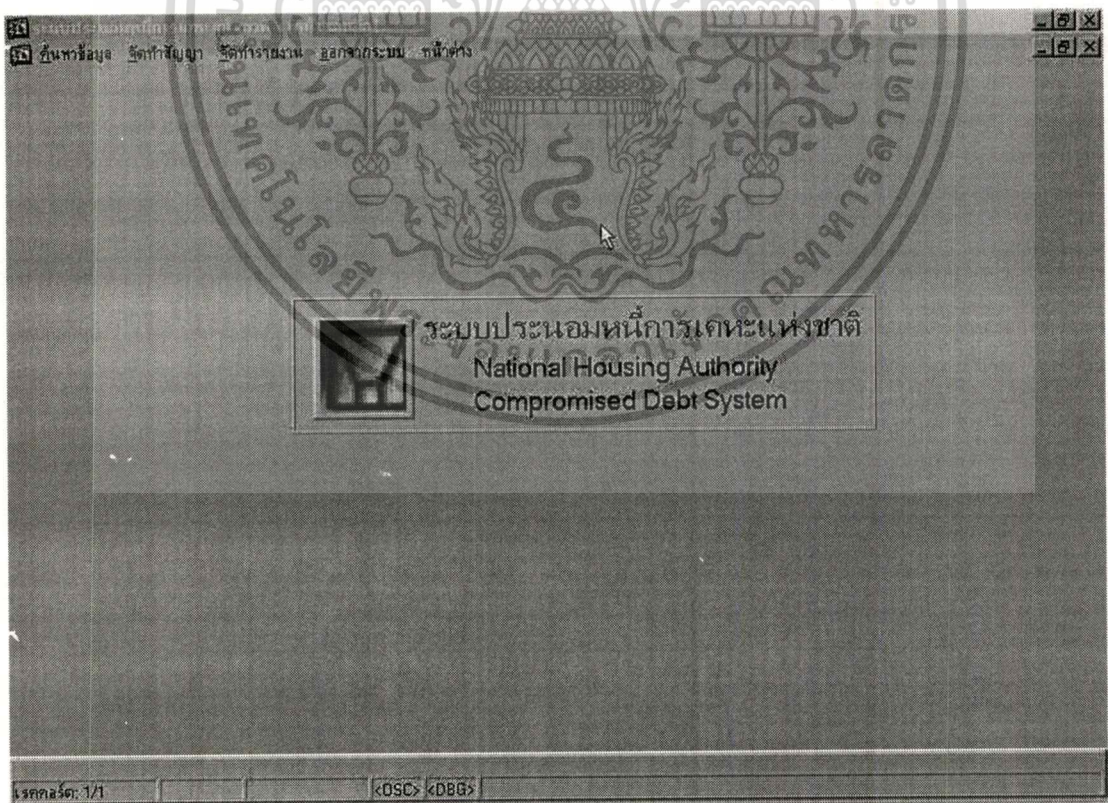
#### 4.5 การออกแบบยูสเซอร์อินเตอร์เฟซ

หลังจากได้ คลาส ที่เป็น User Interface แล้วต่อไปจะเป็นการกำหนดความสัมพันธ์ของคลาส เหล่านี้เพื่อให้เห็นโครงสร้างของ User Interface ชัดขึ้น

จากความสัมพันธ์และ คลาส ที่ได้จะใช้ในการออกแบบและสร้าง Prototype ขึ้นมาเป็นแบบรูปภาพ (GUI) โดยสิ่งที่จะแสดงใน GUI นี้อาจจะพิจารณาจาก Attribute ที่อยู่ใน คลาส ที่มีความสัมพันธ์กันใน Business Layer ช่วยก็ได้ ซึ่ง Prototype ที่ได้มานั้นอาจเป็นต้นแบบในการช่วยพัฒนาต่อไปได้

การสร้าง Prototype ในช่วงการออกแบบนี้จะใช้ Oracle Developer 6 ในการสร้างซึ่งจะได้ GUI ดังนี้

##### 4.5.1 Class MainUI



รูปที่ 4. 21 แสดงหน้าจอเมนูหลังของระบบการประเมินหนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ใน GUI นี้จะเป็นส่วนของหน้าจอหลักในการทำงานของผู้ใช้งาน โดยมีการทำงานหลักเป็นเพียงการเรียกให้ GUI อื่นขึ้นมาเท่านั้น

รายละเอียดใน Class MainUI นั้น จะประกอบไปด้วยเมนูต่างๆเพื่อไปเรียกหน้าจอต่างๆขึ้นมาทำงาน โดยแบ่งเมนูเป็นหมวดหมู่ได้แก่ เมนูค้นหาข้อมูล เมนูจัดทำสัญญา เมนูจัดทำรายงาน เมนูออกจากระบบงาน และ เมนูหน้าต่าง

#### 4.5.1.1 เมนูค้นหาข้อมูล

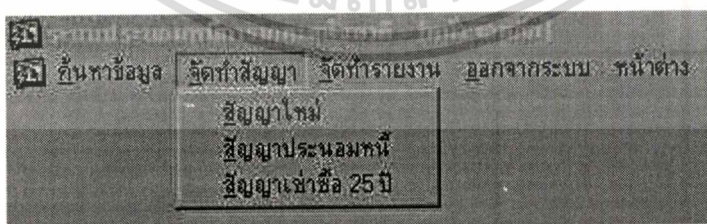
เป็นเมนูสำหรับการค้นหาข้อมูลซึ่งจะประกอบด้วยการค้นหาข้อมูลของรายละเอียดสัญญา และข้อมูลรายละเอียดข้อตกลงชำระของแต่ละสัญญา



รูปที่ 4. 22 แสดงเมนูย่อยสำหรับการค้นหาข้อมูล

#### 4.5.1.2 เมนูจัดทำสัญญา

เป็นเมนูสำหรับการจัดทำสัญญาใหม่ในกรณีที่ต้องการทำการประนอมหนี้ จะต้องมีการจัดทำสัญญาใหม่ ซึ่งจะประกอบด้วยการจัดทำสัญญาใหม่ การจัดทำสัญญาประนอมหนี้ และการจัดทำสัญญาเช่าซื้อปกติ 25 ปี



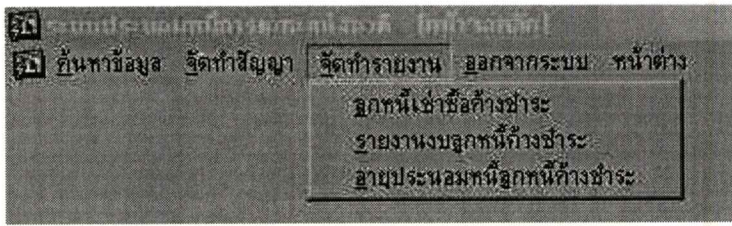
รูปที่ 4. 23 แสดงเมนูย่อยสำหรับการจัดทำสัญญา

#### 4.5.1.3 เมนูจัดทำรายงาน

เป็นเมนูสำหรับการจัดทำรายงาน ซึ่งจะประกอบรายงานต่างๆที่เกี่ยวกับระบบประนอมหนี้ โดยจะประกอบไปด้วย รายงานลูกหนี้ค้างชำระค่าเช่าซื้อ ลูกหนี้ค้างชำระค่าเช่าซื้อแยกตามโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 4. 24 แสดงเมนูย่อยสำหรับการจัดทำรายงาน

#### 4.5.2 Class Search UI

ในส่วนของ Class Search UI นี้ จะประกอบไปด้วย หน้าจอเกี่ยวกับการค้นหาข้อมูลต่างๆ ของสัญญา โดยใน Class Search UI นี้จะประกอบด้วย 2 หน้าจอหลัก คือ หน้าจอค้นหาข้อมูลรายละเอียดสัญญา และหน้าจอค้นหาข้อมูลยอการค้าชำระ

##### 4.5.2.1 ค้นหาข้อมูลรายละเอียดสัญญา

หน้าจอค้นหาข้อมูลรายละเอียดสัญญานี้ จะเป็นส่วนของการค้นหาข้อมูลรายละเอียดสัญญา ซึ่งในการค้นหานี้ จะสามารถค้นหาได้จาก เลขที่สัญญา และค้นหาจาก ชื่อ-สกุลลูกค้า รายละเอียดต่างๆที่จะแสดงในหน้าจอนี้ เช่น วันที่ทำสัญญา ประเภทสัญญา สถานะของสัญญา เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 4. 25 แสดงหน้าจอค้นหารายละเอียดสัญญา

ในการนี้แต่ละฟังก์ชัน อีกทั้งยังมีเพิ่มเติมเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



#### 4.5.2.2 แสดงหน้าจอกันหารายละเอียดยอดค้ำชำระ

หน้าจอกันหาข้อมูลรายละเอียดยอดค้ำชำระ จะเป็นส่วนของการค้นหาข้อมูลเกี่ยวกับยอดค้ำชำระของสัญญา ซึ่งในการค้นหานี้ จะสามารถค้นหาได้จาก เลขที่สัญญา และค้นหาจาก ชื่อ-สกุลลูกค้า โดยในรายละเอียดยอดค้ำชำระนั้น จะแสดงข้อมูลต่างๆ เช่น วันที่ชำระล่าสุด งวดล่าสุด ที่ชำระ จำนวนงวดที่ค้ำชำระ จำนวนเงินที่ค้ำชำระ เป็นต้น

รูปที่ 4. 26 แสดงหน้าจอกันหารายละเอียดยอดค้ำชำระ

#### 4.5.3 Class Create Contract UI

ในส่วนของ Class Create Contract UI นี้ จะเป็นส่วนของการจัดทำสัญญาใหม่ ซึ่งจะประกอบไปด้วย 3 หน้าจอหลัก คือ หน้าจอจัดทำสัญญาใหม่ หน้าจอจัดทำสัญญาประนอมหนี้ และ หน้าจอจัดสัญญาเช่าซื้อ 25 ปี

##### 4.5.3.1 แสดงหน้าจอจัดทำสัญญาใหม่

เป็นหน้าจอที่ทำงานเกี่ยวกับการจัดทำสัญญาใหม่ โดยในการดำเนินงานของหน้านี้ คือ จะมีการค้นหาข้อมูล ของสัญญาเก่าขึ้นมาก่อน โดยในการค้นหา จะค้นหาได้จาก เลขที่สัญญา หรือ ชื่อ-สกุลลูกค้า และเมื่อได้รายละเอียดของสัญญาเก่าแล้ว ต่อไปจะเป็นการจัดทำสัญญาใหม่ โดยจะไม่วางเงื่อนไขทางอื่น อีกทางห้ามให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



มีการกำหนดเลขที่สัญญาใหม่ และทำการบันทึกข้อมูลของสัญญาใหม่ ลงในฐานข้อมูล และนอกจากนี้การบันทึกยังเป็นการแก้ไขข้อมูลสัญญาเก่า โดยจะแก้ไขสถานะของสัญญาเก่าเป็น “ประนอมหนี้” (ดังนั้นการบันทึกจะเป็นการ insert ข้อมูลสัญญาใหม่ และ เป็นการ update ข้อมูลสัญญาเก่า)

ระบบประมวลผลเอกสารสัญญา (เอกสารสัญญา)

บันทึกข้อมูล จัดทำสัญญา จัดทำรายงาน ดูเอกสารระบบ หน้าต่าง

จัดทำสัญญาใหม่

วันที่ 28 เม.ย. 2000

☒ ค้นหาจากเลขที่สัญญา

☐ ค้นหาจากชื่อ-สกุลลูกค้า

ชื่อ

สกุล

เลขที่สัญญา ชื่อโครงการ แบบที่อยู่อาศัย บ้านเลขที่

เลือก

รายละเอียดลูกค้า

เลขประจำตัว

บัตรประชาชน

ชื่อ

สกุล

รายละเอียดสัญญาเก่า

รหัสผ่านอาศัย

วันที่ทำสัญญา

ประเภทสัญญา

สถานะสัญญา

ประเภทการชำระเงิน

วันที่จ่ายเงินงวดแรก

จำนวนงวด

จำนวนงวดต่อปี

ดำเนินการ

จัดทำสัญญาใหม่

เลขที่สัญญาใหม่

เปลี่ยนสถานะสัญญาก่อนบันทึก

พิมพ์สัญญา

พิมพ์สัญญาใหม่

บันทึก

หน้าจอหลัก

จบการทำงาน

ทำสัญญาประนอมหนี้

เวอร์ชัน: 1/1

<ESC> <ENTER>

รูปที่ 4. 27 แสดงหน้าจอจัดทำสัญญาใหม่

#### 4.5.3.2 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ 10 ปี

เป็นหน้าจอที่จัดทำสัญญาประนอมหนี้ โดยจะมีการคำนวณค่าเช่าซื้อต่องวด ของสัญญาประนอมหนี้นั้นๆ ขั้นตอนในการดำเนินงานของหน้าจอนี้ คือ จะทำการค้นหาข้อมูล ของสัญญาที่จะทำการประนอมหนี้ขึ้นมา และจากนั้น จะทำการคำนวณค่าเช่าซื้อต่องวดของสัญญาประนอมหนี้ นั้น โดยจะต้องมีการกำหนดค่า อัตราดอกเบี้ย และจำนวนงวดทั้งหมดที่ต้องการเสียก่อน



ก.ก. - กองบังคับการตำรวจจราจร - กองบัญชาการตำรวจนครบาล

ก.ก. ค้นหาข้อมูล จัดทำสัญญา จัดทำรายงาน ออกจากระบบ หน้าต่าง

**จัดทำสัญญาประนอมหนี้ 10 ปี**

วันที่ 28 เม.ย. 2000

☒ ค้นหาจากเลขที่สัญญา ค้นหา

☐ ค้นหาจากชื่อ-สกุลลูกค้า

ชื่อ

สกุล

| เลขที่สัญญา | ชื่อโครงการ | แบบที่อยู่อาศัย | บ้านเลขที่ | เลือก |
|-------------|-------------|-----------------|------------|-------|
|             |             |                 |            |       |
|             |             |                 |            |       |
|             |             |                 |            |       |
|             |             |                 |            |       |
|             |             |                 |            |       |

รายละเอียดเงินต้นสัญญาใหม่

จำนวนเงินที่ต้องชำระ  บาท

อัตราดอกเบี้ยทั้งหมด  บาท

รายละเอียดการชำระเงินคงงวด

อัตราดอกเบี้ยประนอมหนี้  เปอร์เซ็นต์

จำนวนงวดที่ต้องการ

จำนวนเงินคงงวด  บาท

จำนวนเงินคงงวด

ทำสัญญาเสร็จร้อย 25 ปี

เวอร์ชัน: 1/1      <RSC> <DBG>

รูปที่ 4. 28 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ 10 ปี

#### 4.5.3.3 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี

เป็นหน้าจอที่จัดทำสัญญาเช่าซื้อ 25 ปี โดยจะมีการคำนวณค่าเช่าซื้อต่องวดของสัญญานั้นๆ ขั้นตอนในการดำเนินงานของหน้าจอนี้ คือ จะทำการค้นหาข้อมูลของสัญญาขึ้นมา และจากนั้น จะทำการคำนวณค่าเช่าซื้อต่องวดของสัญญานั้น โดยจะต้องมีการกำหนดค่า อัตราดอกเบี้ย และจำนวนงวดทั้งหมดที่ต้องการเสียก่อน



โปรแกรมประมวลผลข้อมูลระบบงาน...  
 ขั้นตอนการปฏิบัติงาน จัดทำสัญญา จัดทำรายงาน ออกจากระบบ หน้าต่าง

### จัดทำสัญญาเช่าซื้อปกติ 25 ปี

วันที่ 28 เม.ย. 2000

☒ ค้นหาจากเลขที่สัญญา

ค้นหา

☒ ค้นหาจากชื่อ-สกุลลูกค้า

ชื่อ

สกุล

| เลขที่สัญญา | ชื่อโครงการ | แบบที่อยู่อาศัย | บ้านเลขที่ |
|-------------|-------------|-----------------|------------|
|             |             |                 |            |
|             |             |                 |            |
|             |             |                 |            |
|             |             |                 |            |
|             |             |                 |            |

เลือก

รายละเอียดการชำระเงินคงงวด

|                                                                                                            |                                                                                                    |
|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <p>จำนวนเงินที่ต้องชำระ <input type="text"/> บาท</p> <p>อัตราดอกเบี้ย <input type="text"/> เปอร์เซ็นต์</p> | <p>จำนวนงวดที่ต้องชำระ <input type="text"/> งวด</p> <p>จำนวนเงินคงงวด <input type="text"/> บาท</p> |
|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|

คำนวณเงินคงงวด

บันทึก

ล้างหน้าจอ

หน้าจอหลัก

จบการทำงาน

รหัสเครื่อง: 1/1      ROSE <DEBUG>

รูปที่ 4. 29 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี

## บทที่ 5

### การสร้างระบบ

ในช่วงการสร้างระบบจะเป็นการเขียนโปรแกรมจริงตามที่ได้มีการออกแบบไว้ และการออกแบบที่ได้มานั้นเป็นลักษณะการทำงานแยกกันหลายส่วน หรือเป็นที่รู้จักกันในสถาปัตยกรรม Client/Server ซึ่งโดยรวมแล้วที่ออกแบบไว้จะแบ่งเป็น 2 ส่วนใหญ่ คือส่วนที่เกี่ยวกับ User Interface และส่วนของ Business Layer (Application Logic) จะถูกพัฒนาไว้ในฝั่ง Client และส่วนที่เกี่ยวกับ Access Layer จะถูกพัฒนาไว้บนฝั่ง Server

เครื่องมือที่ใช้ในการพัฒนา

- Rational Rose 2000 (CASE Tool)
- Oracle Developer 6
- Oracle Server 8

#### 5.1 การพัฒนานฝั่ง Server

จากการออกแบบ Access Layer ซึ่งเป็นกลุ่ม คลาส ที่ทำงานอยู่ในส่วนของฝั่ง Server โดยในการสร้างระบบฝั่ง Server นี้ จะจากการสร้างระบบฐานข้อมูลของระบบประนอมหนี้จาก Schema ที่ออกแบบได้ และทำการสร้าง Store Procedure ไปเก็บไว้บนฝั่ง Server โดยจะมีการขั้นตอนในการติดตั้งฝั่ง Server ดังนี้

- การติดตั้ง DBMS
- การสร้าง Table ลงใน Database เพื่อเก็บข้อมูลของระบบ
- การสร้าง Store Procedure ใน Database

ในส่วนของการติดตั้ง DBMS และการสร้าง Table นั้น ได้กล่าวไว้แล้วในส่วนของการสร้างฐานข้อมูลในบทที่ 4

ส่วนในการสร้าง Store Procedure จาก คลาส ในกลุ่ม Access Layer จะสร้างเป็น Package 1 Package ต่อ คลาส 1 คลาส ใน Access Layer

การสร้าง Store Procedure ของระบบประนอมหนี้จะอาศัย Oracle Developer 6 ช่วยในการสร้าง โดยจะให้ Developer 6 ทำการติดต่อไปยัง Server และสร้าง Store Procedure บนฝั่ง Server ผ่านทาง Developer 6 และใช้ภาษา PL/SQL สร้าง Store Procedure ของระบบ

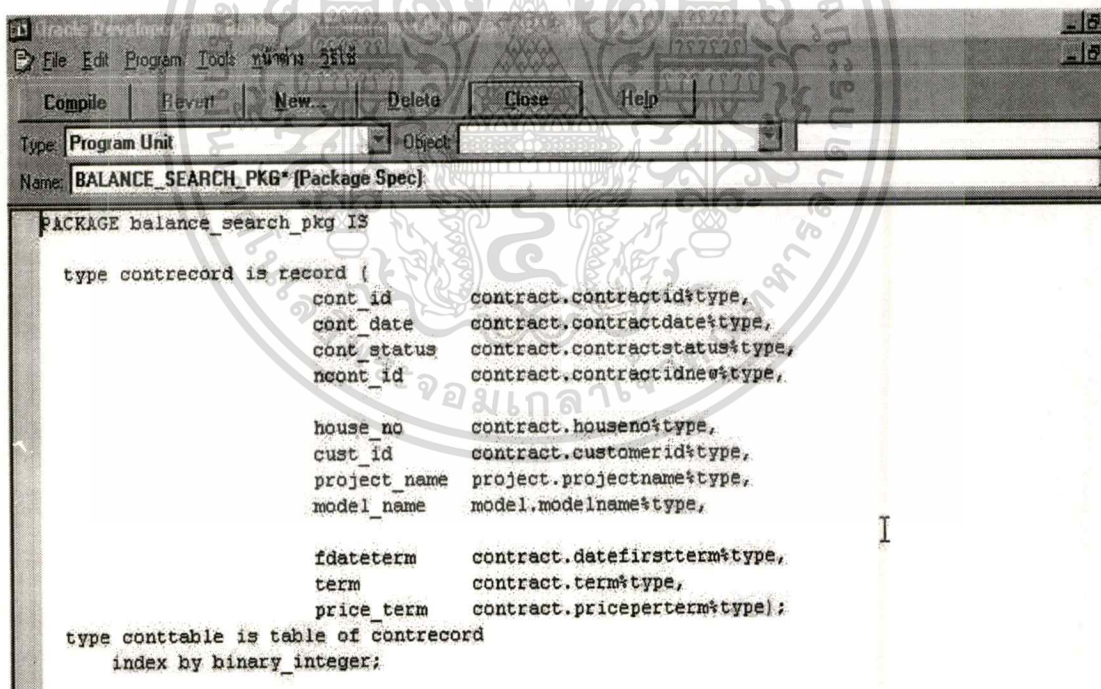


Package ของ Oracle นั้นจะถูกแบ่งเป็น 2 ส่วนคือส่วนที่เป็น Program Specification และ Program Body ซึ่ง Program Specification จะเปรียบเสมือนเป็น Interface สำหรับใช้งานและภายในจะมีเพียง ชื่อ Function และ/หรือ Procedure และ Parameter ต่างๆ พร้อมกับ ค่าที่ต้องการจะส่งกลับ และส่วน Program Body จะเป็นรายละเอียด Code โปรแกรม ที่มี Function และ/หรือ Procedure ตรงกับที่ประกาศไว้ในส่วนของ Program Specification และอาจมี Function / Procedure เพิ่มภายใน Program Body ได้ แต่จะ Interface ผ่านตรงๆ ไม่ได้

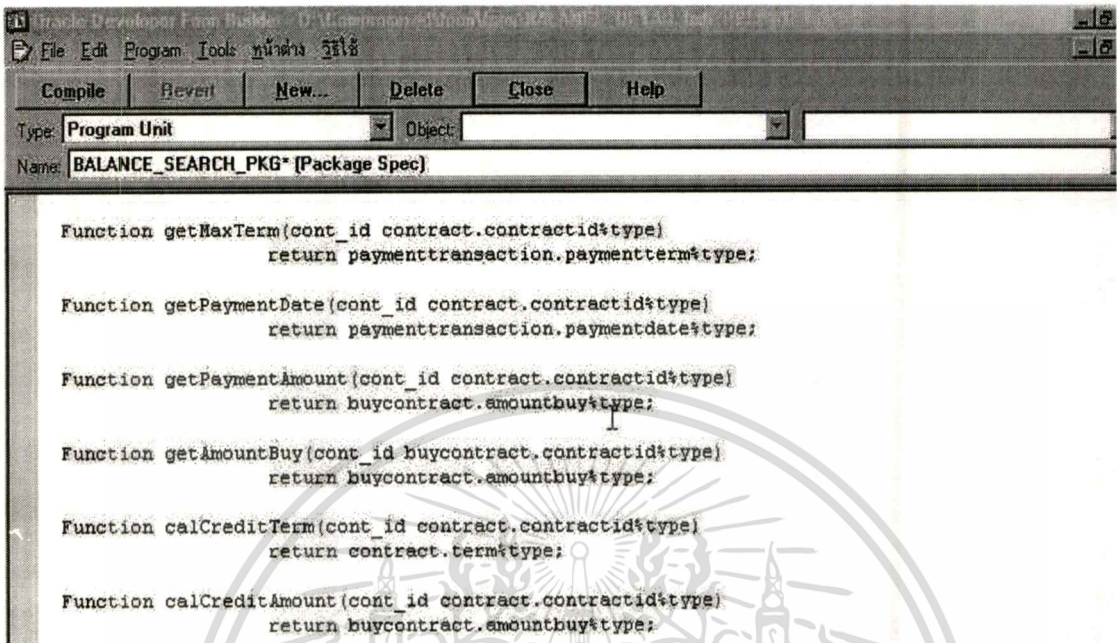
Program Specification ใน Package นั้นเปรียบเป็นการประกาศ Visibility เป็น Public ของ Object-Oriented

Program Body ใน Package ก็เปรียบเป็นการประกาศ Visibility เป็น Private ใน Object-Oriented

ตัวอย่างของ Program Specification นั้นจะแสดงไว้ในรูปที่ 5.1 และ รูปที่ 5.2



รูปที่ 5.1 แสดงตัวอย่าง ส่วนของ Parameter หรือ ค่าตัวแปรต่างๆ ใน Program Specification



```

Oracle Developer Form Builder  11.0.0.4.0.0  11.0.0.4.0.0  11.0.0.4.0.0  11.0.0.4.0.0
File Edit Program Tools  หน้าต่าง  วิเคราะห์
Compile Revert New... Delete Close Help
Type: Program Unit  Object:
Name: BALANCE_SEARCH_PKG* (Package Spec)

Function getMaxTerm(cont_id contract.contractid%type)
    return paymenttransaction.paymentterm%type;

Function getPaymentDate(cont_id contract.contractid%type)
    return paymenttransaction.paymentdate%type;

Function getPaymentAmount(cont_id contract.contractid%type)
    return buycontract.amountbuy%type;

Function getAmountBuy(cont_id buycontract.contractid%type)
    return buycontract.amountbuy%type;

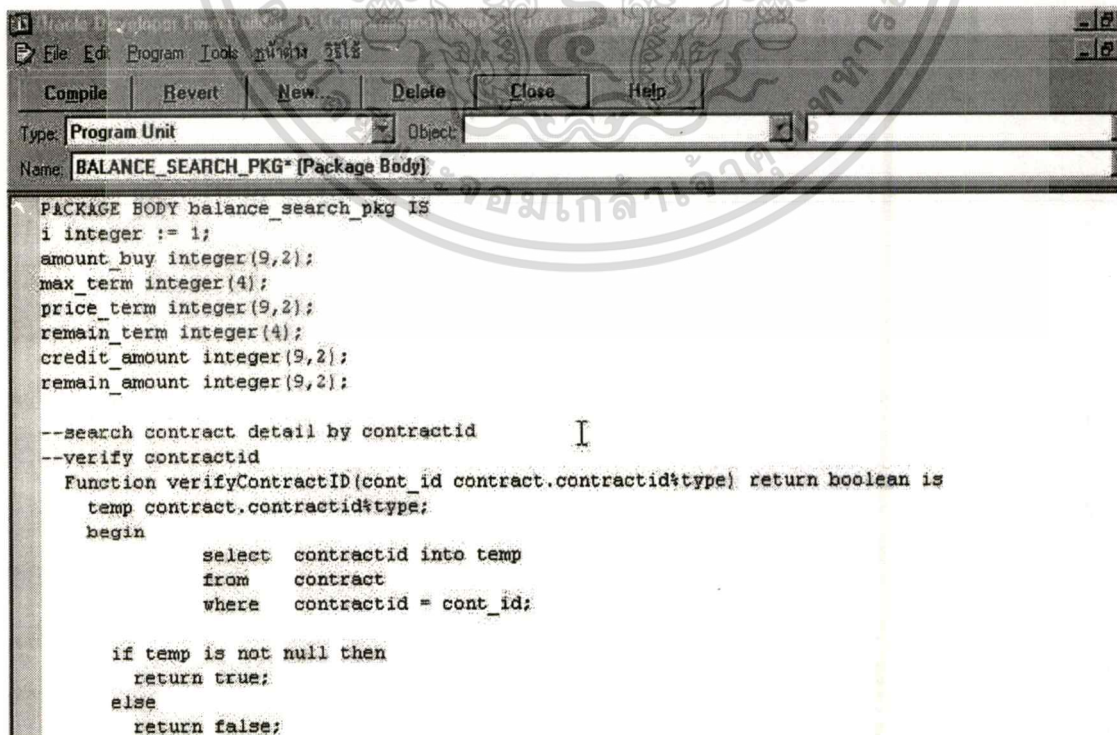
Function calCreditTerm(cont_id contract.contractid%type)
    return contract.term%type;

Function calCreditAmount(cont_id contract.contractid%type)
    return buycontract.amountbuy%type;

```

รูปที่ 5.2 แสดงตัวอย่าง ส่วนของ Function ต่างๆ ใน Program Specification

ตัวอย่างของ Program Body จะแสดงดังรูปที่ 5.3 และ รูปที่ 5.4



```

Oracle Developer Form Builder  11.0.0.4.0.0  11.0.0.4.0.0  11.0.0.4.0.0  11.0.0.4.0.0
File Edit Program Tools  หน้าต่าง  วิเคราะห์
Compile Revert New... Delete Close Help
Type: Program Unit  Object:
Name: BALANCE_SEARCH_PKG* (Package Body)

PACKAGE BODY balance_search_pkg IS
i integer := 1;
amount_buy integer(9,2);
max_term integer(4);
price_term integer(9,2);
remain_term integer(4);
credit_amount integer(9,2);
remain_amount integer(9,2);

--search contract detail by contractid
--verify contractid
Function verifyContractID(cont_id contract.contractid%type) return boolean is
temp contract.contractid%type;
begin
    select contractid into temp
    from contract
    where contractid = cont_id;

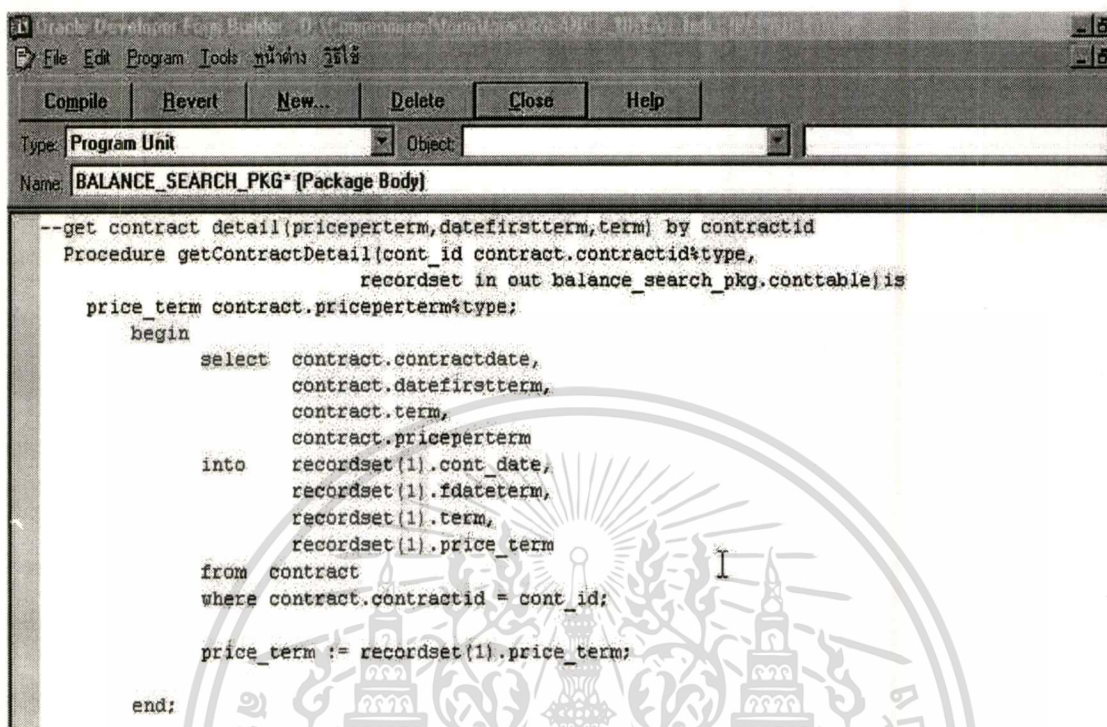
    if temp is not null then
        return true;
    else
        return false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 5.3 แสดงตัวอย่าง ส่วนของ Program Body และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





```

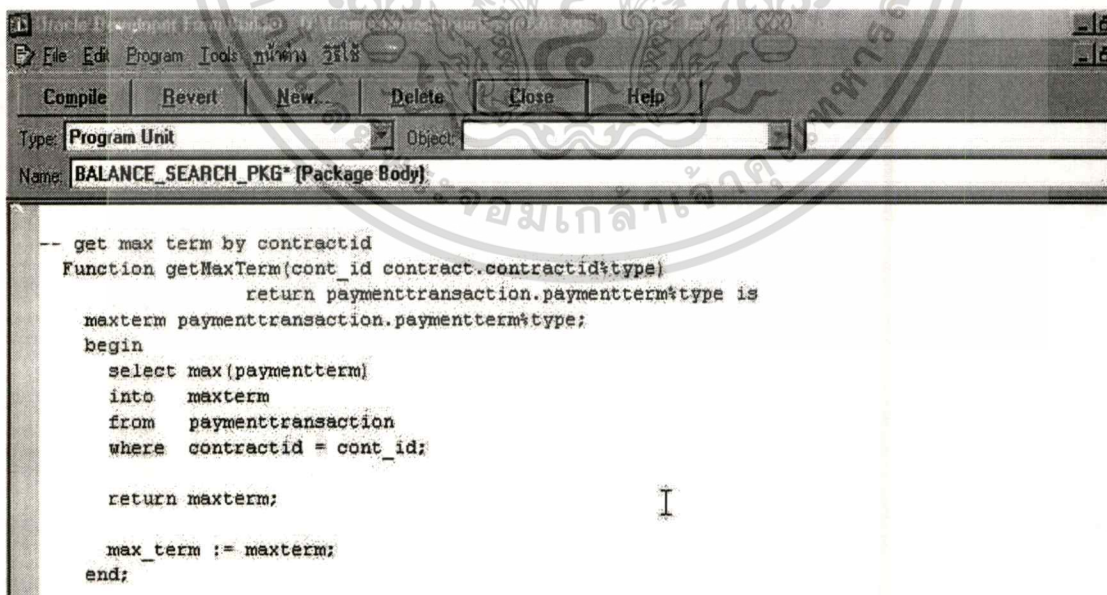
--get contract detail(priceperterm,datefirstterm,term) by contractid
Procedure getContractDetail(cont_id contract.contractid%type,
                           recordset in out balance_search_pkg.conttable)is
  price term contract.priceperterm%type;
  begin
    select  contract.contractdate,
            contract.datefirstterm,
            contract.term,
            contract.priceperterm
    into    recordset(1).cont_date,
            recordset(1).fdateterm,
            recordset(1).term,
            recordset(1).price term
    from    contract
    where   contract.contractid = cont_id;

    price_term := recordset(1).price_term;

  end;

```

รูปที่ 5.4 แสดงตัวอย่าง ส่วนของ Program Body



```

-- get max term by contractid
Function getMaxTerm(cont_id contract.contractid%type)
  return paymenttransaction.paymentterm%type is
  maxterm paymenttransaction.paymentterm%type;
  begin
    select max(paymentterm)
    into   maxterm
    from   paymenttransaction
    where  contractid = cont_id;

    return maxterm;

    max_term := maxterm;
  end;

```

รูปที่ 5.5 แสดงตัวอย่าง ส่วนของ Program Body

## 5.2 การพัฒนาบนฝั่ง Client

การพัฒนาบนฝั่ง Client นั้นเป็นการสร้าง UI (User Interface) โดยนำ UI ที่ได้จากขั้นตอนการออกแบบมาพัฒนาต่อจนเสร็จสมบูรณ์ ซึ่งในระบบประนีประนอมนี้จะเป็นการสร้าง UI บนระบบปฏิบัติการ MS Windows ทำให้สามารถแสดง UI เป็นแบบรูปภาพได้ (GUI : Graphic User Interface) และจะใช้ Oracle Developer 6 ช่วยในการพัฒนา GUI นี้ขึ้นมา

ภายใน Oracle Developer 6 มีเครื่องมือที่ช่วยอำนวยความสะดวกในการสร้าง UI หลายตัว โดยหลักๆ ก็เป็นเครื่องมือในการสร้าง Form, Report, Query, Graphic, Procedure, Project, Schema และ Translation ซึ่งที่จำเป็นอย่างยิ่งในการใช้พัฒนาค้างนี้คือเครื่องมือในการสร้าง Form และเครื่องมือในการสร้าง Report

### 5.2.1 การสร้างส่วนรับข้อมูลและแสดงผล (Form)

ในเครื่องมือการสร้าง Form จะมีองค์ประกอบหลักๆ ในการสร้าง Form ดังนี้คือ

- Window เป็นหน้าต่างในการแสดงผล
- Canvas เป็นส่วนที่เปรียบเสมือนพื้นที่สำหรับวางและจัดเรียง Component ต่างๆ
- Data Block ซึ่งเป็นส่วนในการเก็บ Component ต่างๆ ในการรับและแสดงผล เช่น Text Box, Command Button หรือ Check Box เป็นต้น รวมถึงอาจใช้ในการรวมกลุ่มของ Component ที่มีหน้าที่คล้ายๆ กัน
- Program Unit เป็นส่วนในการเก็บ Code โปรแกรมที่ใช้เฉพาะใน Form นั้นๆ อาจเป็น Package, Procedure หรืออาจเป็น Function เป็นต้น
- Trigger เป็นเหมือนเหตุการณ์ต่างๆ ที่จะเกิดขึ้น เช่น When-Button-Pressed จะเกิดขึ้นเมื่อปุ่มถูกกด เป็นต้น หรือที่เรียกว่า Event ในเครื่องมืออื่นๆ

ในส่วนของการใช้งานจะต้องมีการใช้ Canvas ร่วมกับ Window จึงนับว่าเป็น Form หนึ่ง เพราะ Window เป็นเหมือนกับหน้าต่างเปล่าๆ แล้วนำ Canvas ที่เปรียบเหมือนผ้าใบมาจึงบนหน้าต่าง ดังนั้นจะมีลักษณะคล้ายกับ Form หนึ่ง Form

การพัฒนาจะเริ่มต้นจากการนำ UI แต่ละอันที่ได้จากขั้นตอนการออกแบบมาพัฒนาต่อจนกระทั่งมีหน้าที่ครบสมบูรณ์ตรงตามความต้องการของผู้ใช้ โดยอาศัย Algorithm ของแต่ละ Method ที่ออกแบบมา

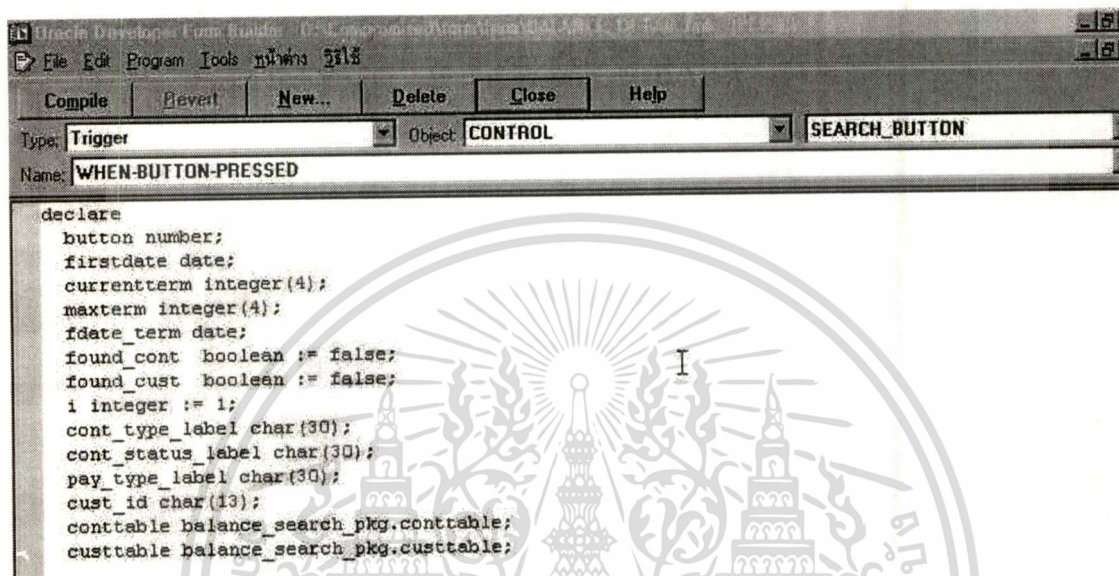
การพัฒนาจะเริ่มต้นจากการนำ UI แต่ละอันที่ได้จากขั้นตอนการออกแบบมาพัฒนาต่อจนกระทั่งมีหน้าที่ครบสมบูรณ์ตรงตามความต้องการของผู้ใช้ โดยอาศัย Algorithm ของแต่ละ Method

ที่ออกแบบมา เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

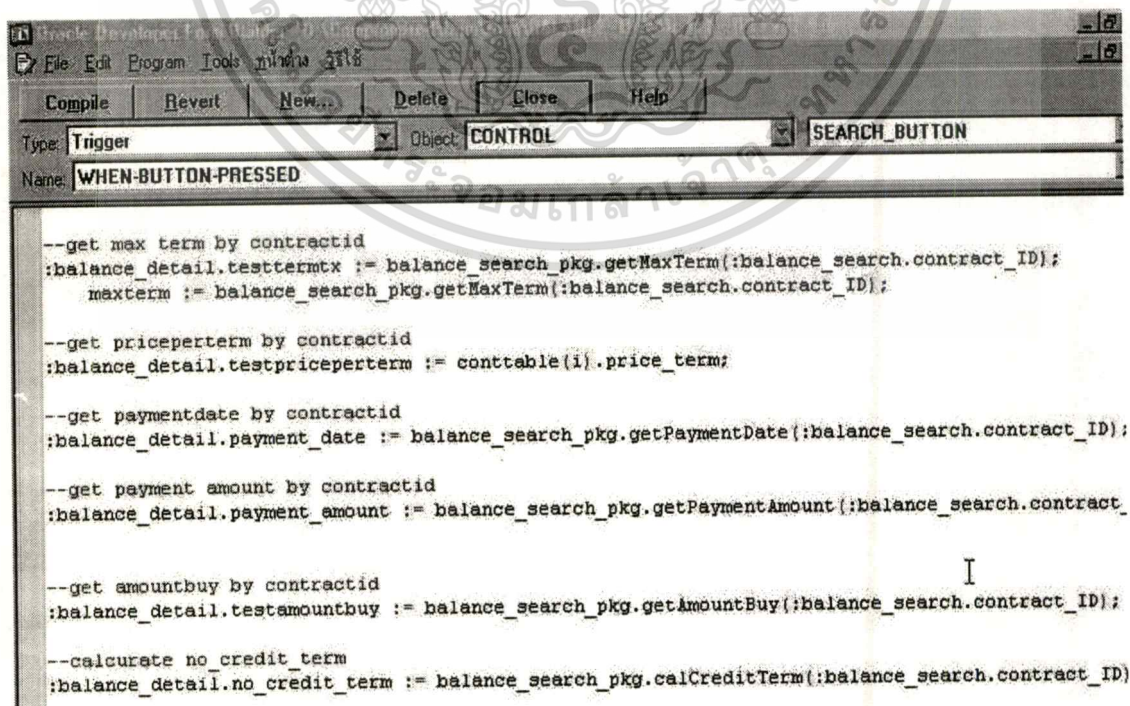
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตัวอย่างการเรียกใช้ Store Procedure ที่เก็บไว้ในส่วน Server โดยการเรียกใช้นี้ จะเรียกใช้ผ่าน User Interface ซึ่งสร้างมาจากหน้าจอ Form ต่างๆ ของ Oracle Developer 6



รูปที่ 5.7 แสดงตัวอย่างการเรียกใช้ Store Procedure ในฝั่ง Server จาก Form ในฝั่ง Client



รูปที่ 5.8 แสดงตัวอย่างการเรียกใช้ Store Procedure ในฝั่ง Server จาก Form ในฝั่ง Client

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ในการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.2.2 การสร้างส่วนรายงาน (Report)

ในเครื่องการสร้าง Report จะมีองค์ประกอบหลักๆ ในการสร้าง Report ดังนี้คือ

- Data Model เป็นส่วนที่ใช้ในการกำหนดรูปแบบชุดของข้อมูลสำหรับสร้างรายงาน หรือเปรียบเป็นที่ในการสืบค้นข้อมูลมาสำหรับการทำรายงาน
- Layer Model เป็นส่วนที่ใช้ในการจัดรูปแบบของรายงานที่ต้องการ โดยจะแสดงข้อมูลที่ Data Model กำหนด

จะสร้างรายงานโดยใช้รูปแบบของรายงานที่แสดงอยู่ในภาคผนวกของเอกสาร

### 5.2.3 การรวมส่วนต่างๆ ที่พัฒนาเข้าด้วยกัน (Integration)

การรวมส่วนต่างๆ ที่พัฒนาเข้าด้วยกัน (Integration) จะเป็นการเชื่อมโยงสิ่งที่พัฒนาเข้าด้วยกัน คือทำให้ทุกสิ่งที่พัฒนามีทางเข้าถึงได้เพียงทางเดียวและทำให้เป็น Application ที่สามารถใช้งานได้ตรงตามความต้องการของผู้ใช้ และในที่นี้คือการเชื่อมประสานกันระหว่าง Form และรายงานที่พัฒนาขึ้นมาให้มีการใช้รายงานโดยผ่าน Form ที่ออกแบบสำหรับเลือกพิมพ์รายงาน

หลังจากนั้น เมื่อมีการเข้าถึงใน Form หรือรายงานต่างๆ จากทางเข้าเพียงทางเดียวแล้ว ต่อไปจะเป็นการคล้ายกับการ Compile เพื่อทำให้เป็น Application อย่างสมบูรณ์ ซึ่งอาจจะใช้เครื่องมือในการสร้าง Project เข้ามาช่วยทำในส่วนนี้ก็ได้



## บทที่ 6

### คู่มือระบบ

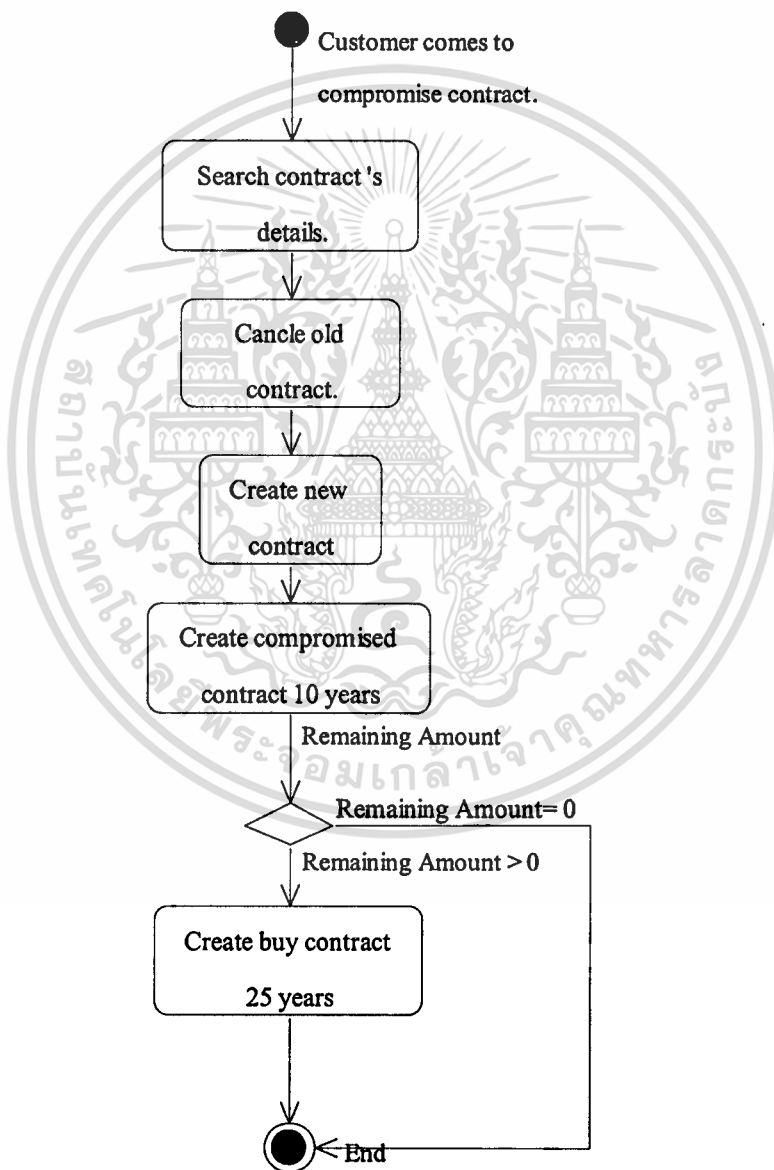
#### 6.1 ความต้องการของระบบ (System Requirement)

เมื่อลูกหนี้มีการค้างชำระค่าเช่าซื้อ ทางการเคหะจะจัดการดำเนินการในส่วนของการประนอมหนี้ โดยขั้นตอน และเงื่อนไขต่างๆในส่วนการประนอมหนี้ คือ

- เมื่อลูกหนี้มีการค้างชำระเกินข้อกำหนดที่ทางการเคหะกำหนดไว้ ต้องทำการเรียกลูกหนี้รายนั้นๆมาทำการประนอมหนี้
- เมื่อลูกหนี้มาทำการประนอมหนี้ ทางการเคหะ จะทำการจัดทำสัญญาเช่าซื้อใหม่ โดยจะต้องมีการแก้ไขข้อมูลในส่วนของสัญญาเดิมด้วย คือ ต้องมีการแก้ไขสถานะของสัญญาเดิม ให้มีสถานะสัญญาเป็น “ประนอมหนี้”
- จากนั้นลูกหนี้จะต้องมีการทำสัญญาอีก 2 สัญญา คือ สัญญาประนอมหนี้ และสัญญาเช่าซื้อปกติ 25 ปี
- ระบบประนอมหนี้ของการเคหะจะต้องมีการเก็บรวบรวมข้อมูล ในส่วนของการทำสัญญาใหม่ สัญญาประนอมหนี้ และสัญญาเช่าซื้อปกติ 25 ปี ไว้ เพื่อใช้ในการดำเนินงานในส่วนต่างๆต่อไป เช่น การจัดทำรายงานอายุประนอมหนี้
- ในส่วนของสัญญาประนอมหนี้ จะเก็บรายละเอียดเกี่ยวกับ รายละเอียดสัญญาประนอมหนี้ เช่น รายละเอียดจำนวนเงินชำระต่องวด จำนวนงวดทั้งหมดที่ชำระ อัตราดอกเบี้ยสัญญาประนอมหนี้ ดังนั้นในส่วนนี้จะมีการคำนวณ จำนวนเงินชำระต่องวด โดยในขณะที่ดำเนินงานพัฒนาระบบนี้ เงื่อนไขในการชำระ อัตราดอกเบี้ยของสัญญาประนอมหนี้ จะมีค่าคงที่ 7 เปอร์เซ็นต์
- ในส่วนการทำสัญญาเช่าซื้อ 25 ปี จะมีการเก็บข้อมูล และรายละเอียดต่างๆ เช่นเดียวกับ สัญญาประนอมหนี้ แต่ในส่วนของอัตราดอกเบี้ยของสัญญาเช่าซื้อ 25 ปี นั้น จะมีค่า เริ่มต้นเท่ากับค่าเช่าซื้อเม และหลังจากนั้นจะเพิ่มขึ้นทุกๆ 5 ปี ซึ่งเงื่อนไขนี้เป็นเงื่อนไขในขณะที่ทำการพัฒนาระบบครั้งนี้

## 6.2 แบบจำลองของกระบวนการทางธุรกิจ

แบบจำลองของกระบวนการทางธุรกิจของระบบประนอมหนี้ ซึ่งจะแสดงกระบวนการของการประนอมหนี้ตามลำดับ



รูปที่ 6.1 แสดง Activity Diagram ของระบบประนอมหนี้

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยทางบริษัทฯ เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.3 Use-Case Diagram

### 6.3.1 Actor

Officer :เจ้าหน้าที่ผู้ใช้งานระบบ

Debtor : ลูกหนี้ของการเคหะ (ลูกค้ำที่ค้างชำระ และต้องการทำการประนอมหนี้)

### 6.3.2 Use Case

Search Contract 's Details (ค้นหารายละเอียดสัญญา) :

ทำการค้นหารายละเอียดของสัญญาที่ต้องการทำการประนอมหนี้

Search Remaining Amount (ค้นหารายละเอียดการค้างชำระ)

ทำการค้นหารายละเอียดการค้างชำระของสัญญาที่ต้องการทำการประนอมหนี้

Update Old Contract (ยกเลิกสัญญาเดิม) :

ทำการยกเลิกสัญญาเช่าซื้อเดิม

Create New Contract (จัดทำสัญญาใหม่) :

จัดทำสัญญาเช่าซื้อใหม่

Create Compromised Contract (จัดทำสัญญาประนอมหนี้) :

จัดทำสัญญาประนอมหนี้ โดยจะทำการคำนวณ จำนวนงวดที่ต้องชำระ และค่าเช่าซื้อต่องวดใหม่ โดยกำหนดระยะเวลาในการใช้สัญญานี้ 10 ปี และกำหนดอัตราดอกเบี้ยคงที่ 7 %

Create Contract 25 Years (จัดทำสัญญาเช่าซื้อ 25 ปี) :

จัดทำสัญญาเช่าซื้อ 25 ปี โดยจะทำการคำนวณ จำนวนงวดที่ต้องชำระ และค่าเช่าซื้อต่องวดใหม่ โดยกำหนดระยะเวลาในการใช้สัญญานี้ไม่เกิน 25 ปี และกำหนดอัตราดอกเบี้ยให้เพิ่มขึ้น 5 % ทุกๆ 5 ปี

Calculate payment per term (คำนวณจำนวนงวด และค่าเช่าซื้อต่องวด) :

ทำการคำนวณจำนวนงวด และค่าเช่าซื้อต่องวด

Create Report (จัดทำรายงานต่างๆ) :

จัดทำรายงาน โดยจะเป็นการทำรายงาน เกี่ยวกับ ลูกหนี้เช่าซื้อค้างชำระ

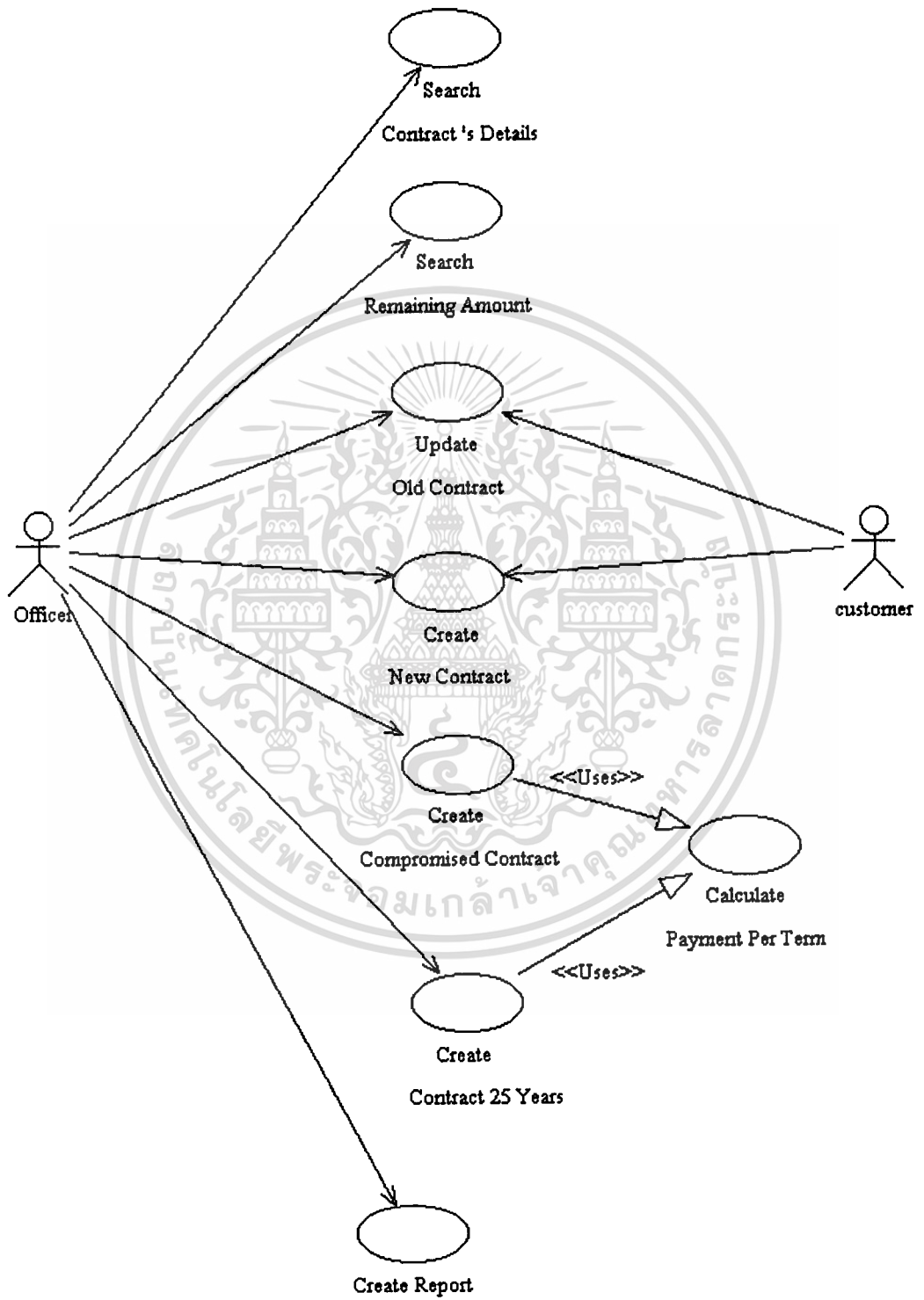
### 6.3.3 Use case Diagram

จาก Actor และ Use case ทั้งหมดที่ได้สามารถนำมาสร้างเป็นไดอะแกรมซึ่งแสดงไว้ดังรูป

ที่ 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

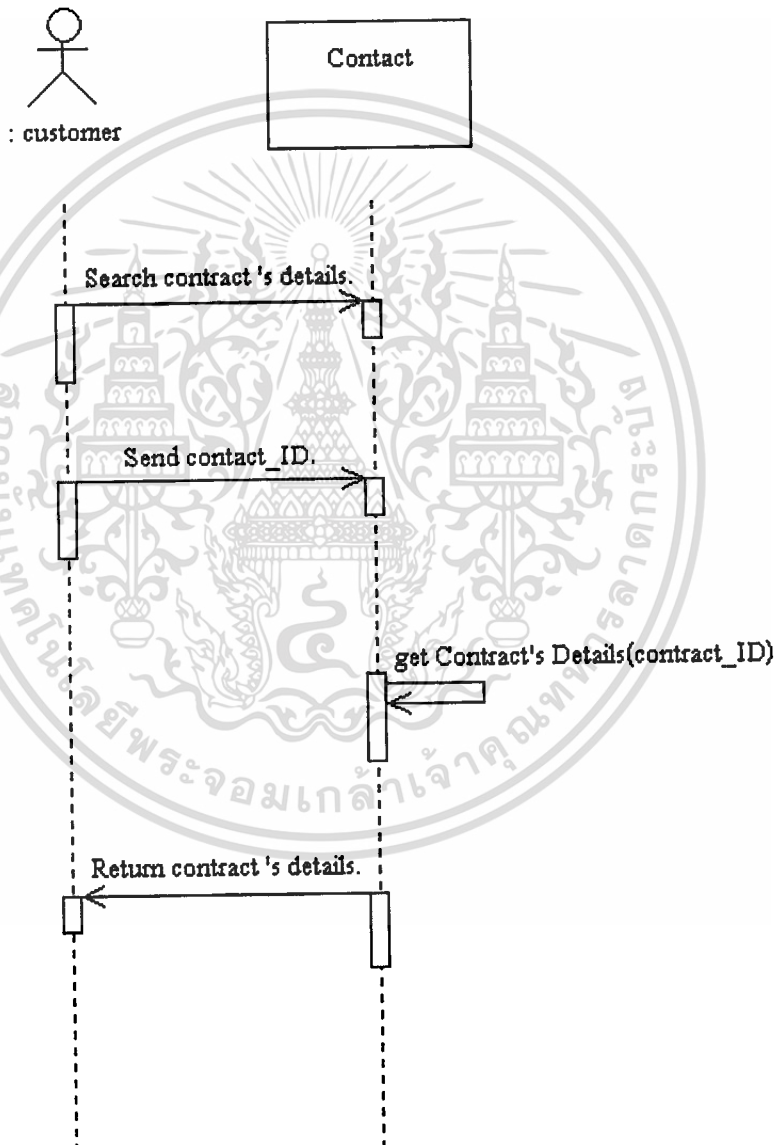




รูปที่ 6. 2 แสดง Use Case Diagram ของระบบประนอมหนี้

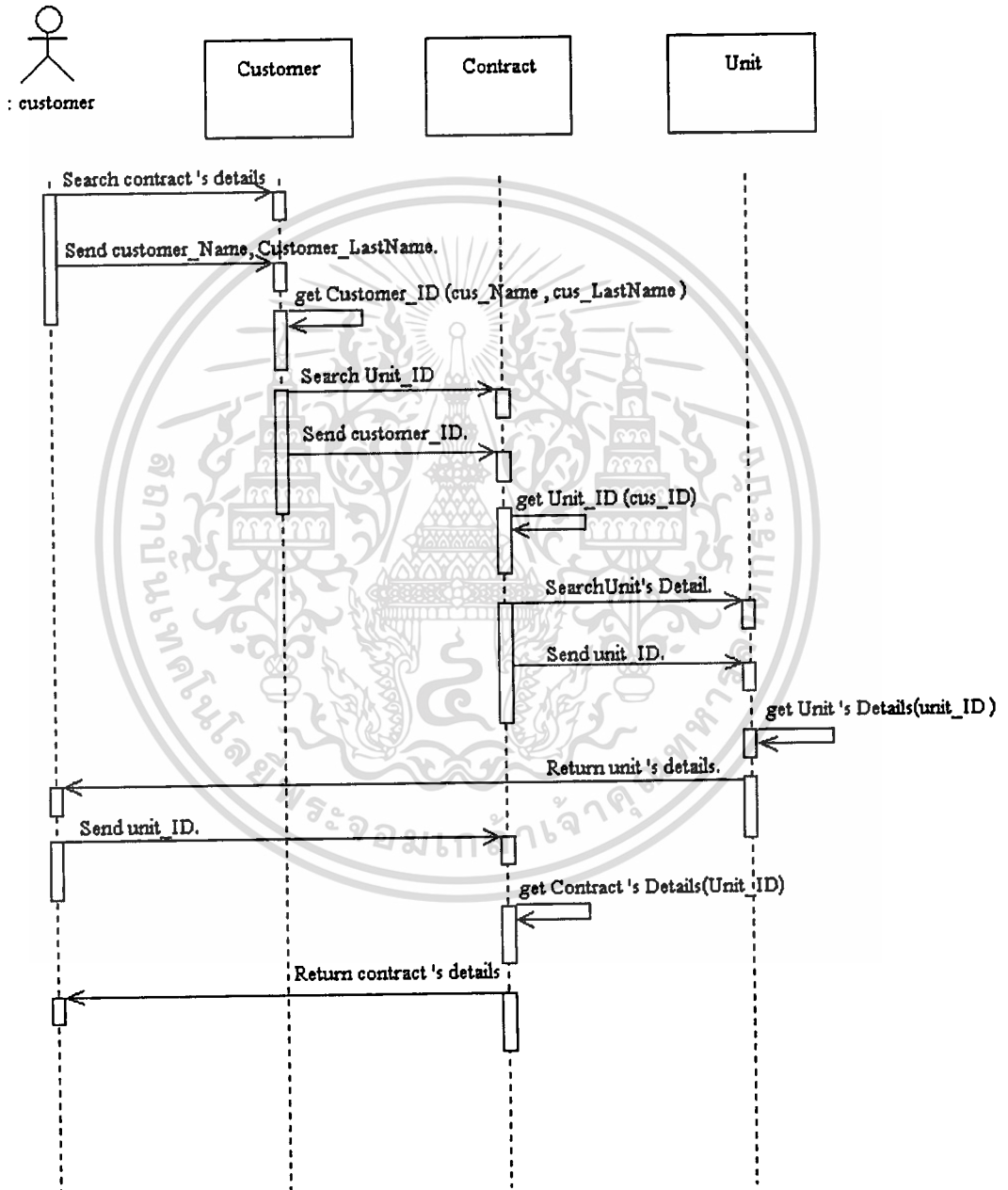
## 6.4 Interaction Diagram

### 6.4.1 Sequence Diagram ของ Search Contract's Details Use Case (ซึ่งใช้เลขที่สัญญาในการค้นหา)



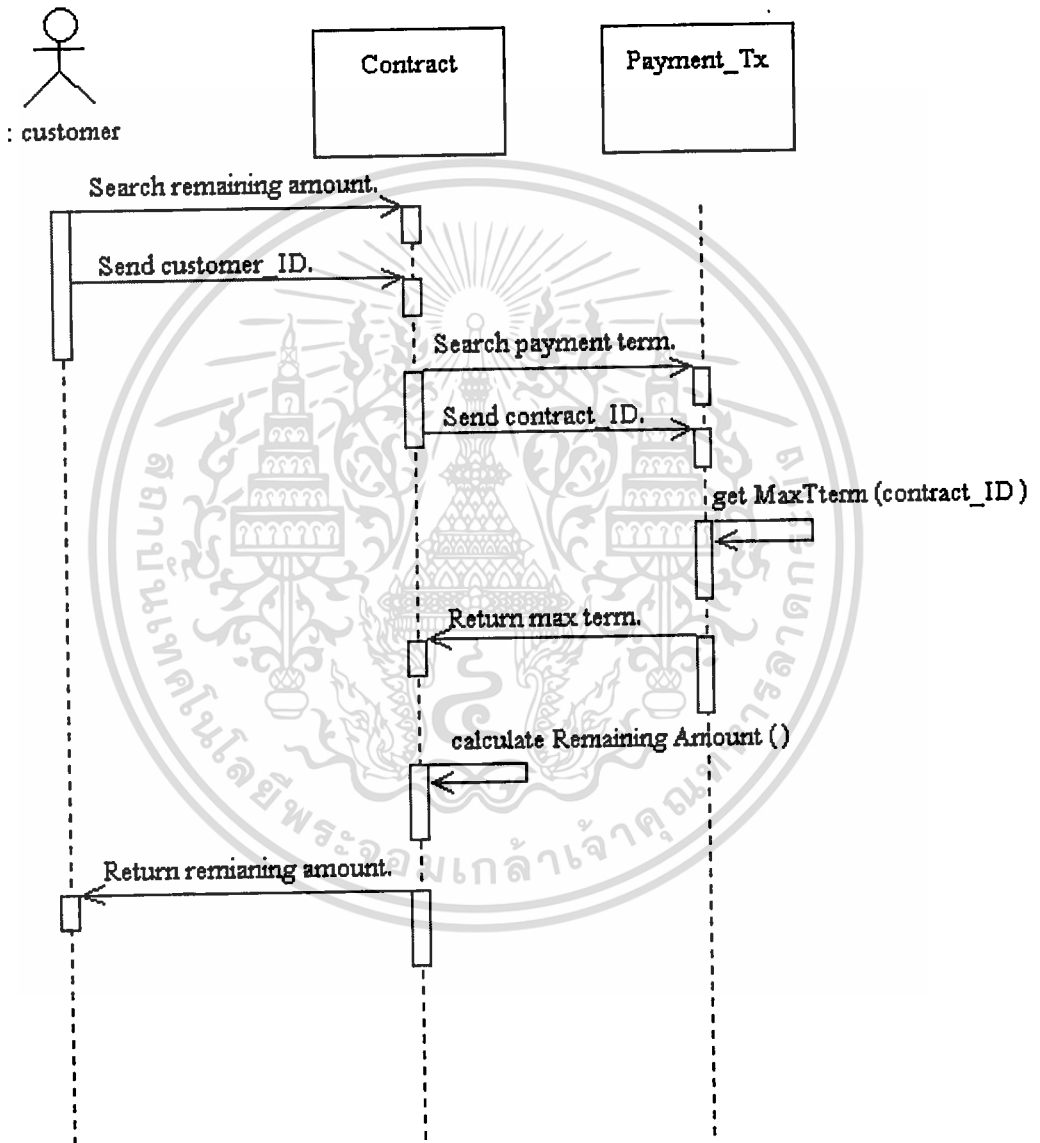
รูปที่ 6. 3 แสดง Sequence Diagram ของ Search Contract's Details Use Case (ซึ่งใช้ เลขที่สัญญาในการค้นหา)

#### 6.4.2 Sequence Diagram ของ Search Contract's Details Use Case (ซึ่งใช้ข้อมูลค้นหาในการค้นหา)



รูปที่ 6. 4 แสดง Sequence Diagram ของ Search Contract's Details Use Case (ซึ่งใช้ข้อมูลค้นหาในการค้นหา)

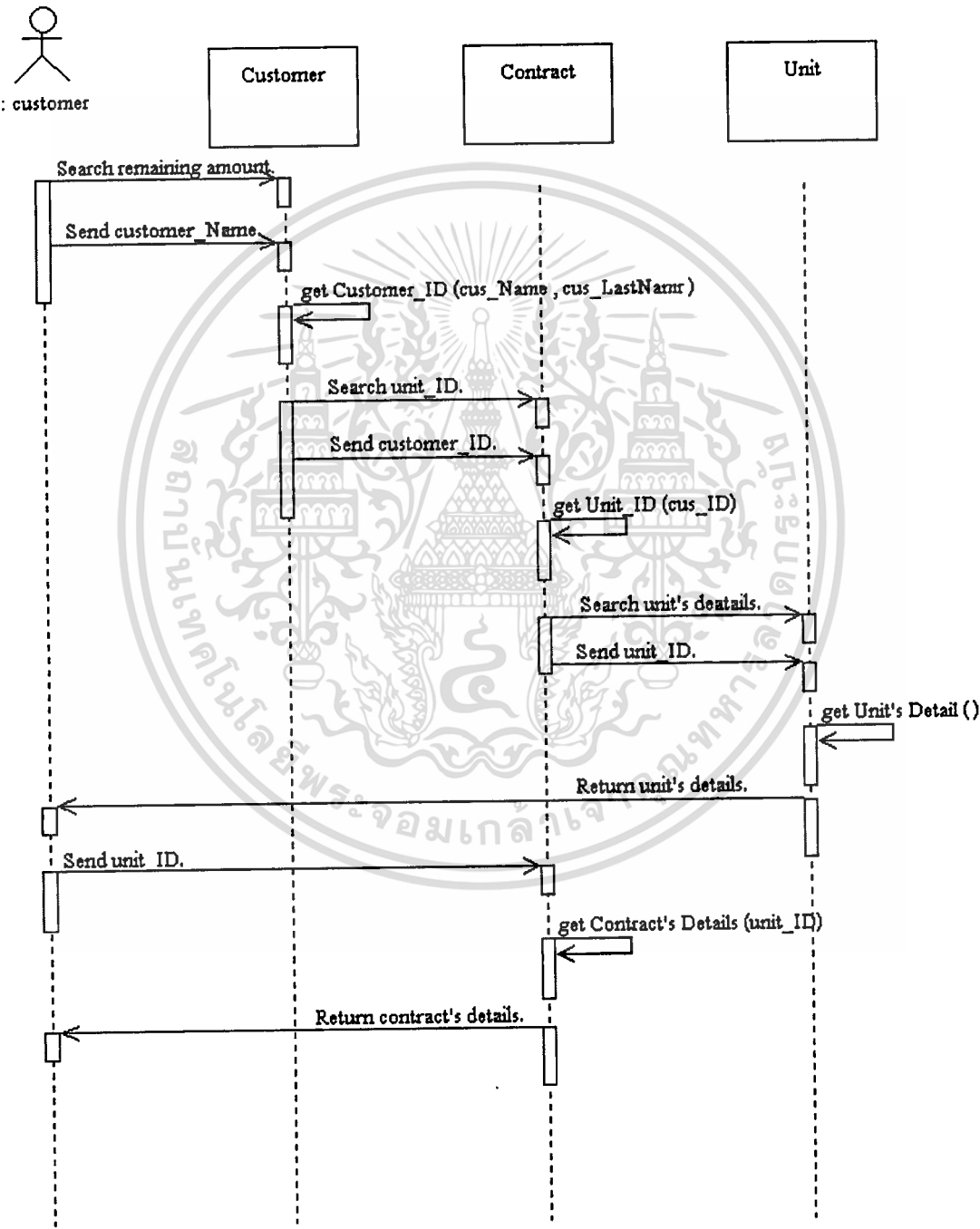
#### 6.4.3 Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้เลขที่สัญญาในการค้นหา)



รูปที่ 6. 5 แสดง Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้เลขที่สัญญาในการค้นหา)



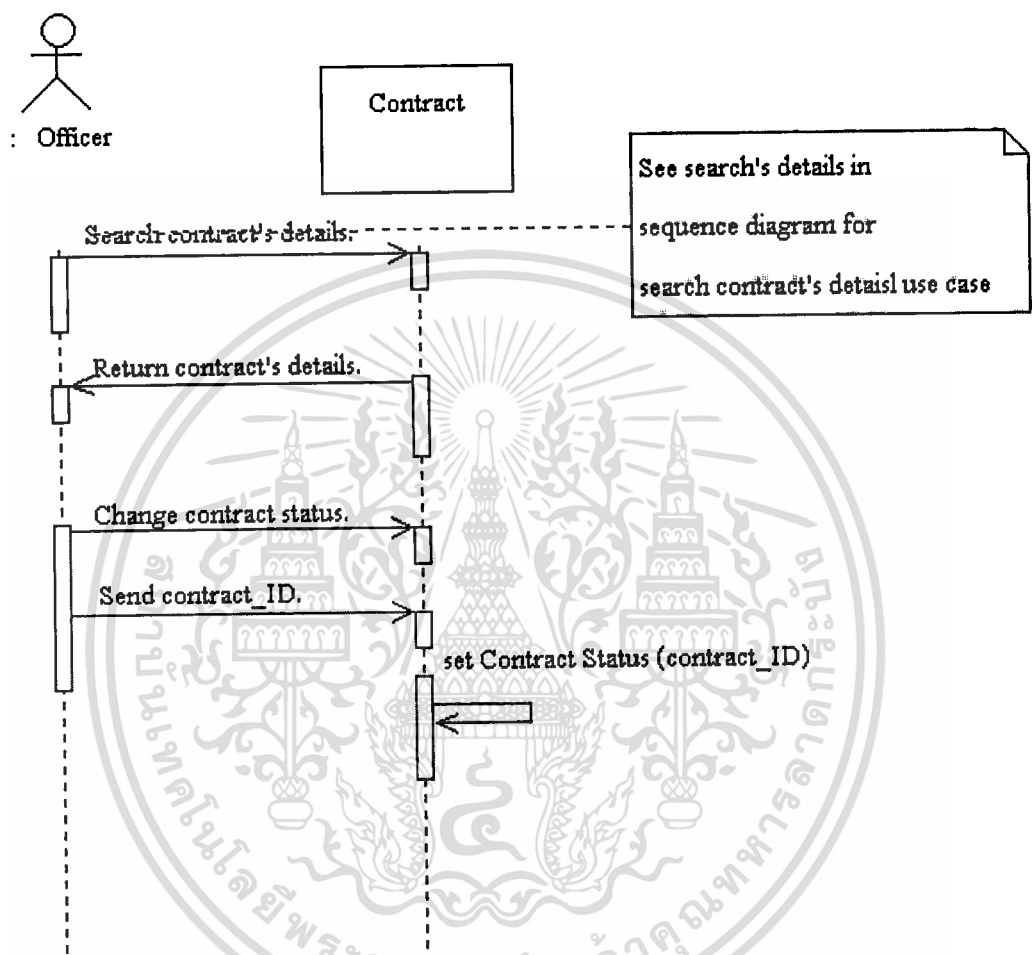
6.4.4 Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้ข้อมูลค่าในการค้นหา)



รูปที่ 6. 6 แสดง Sequence Diagram ของ Search Remaining Amount Use Case (ซึ่งใช้ข้อมูลค่าในการค้นหา)

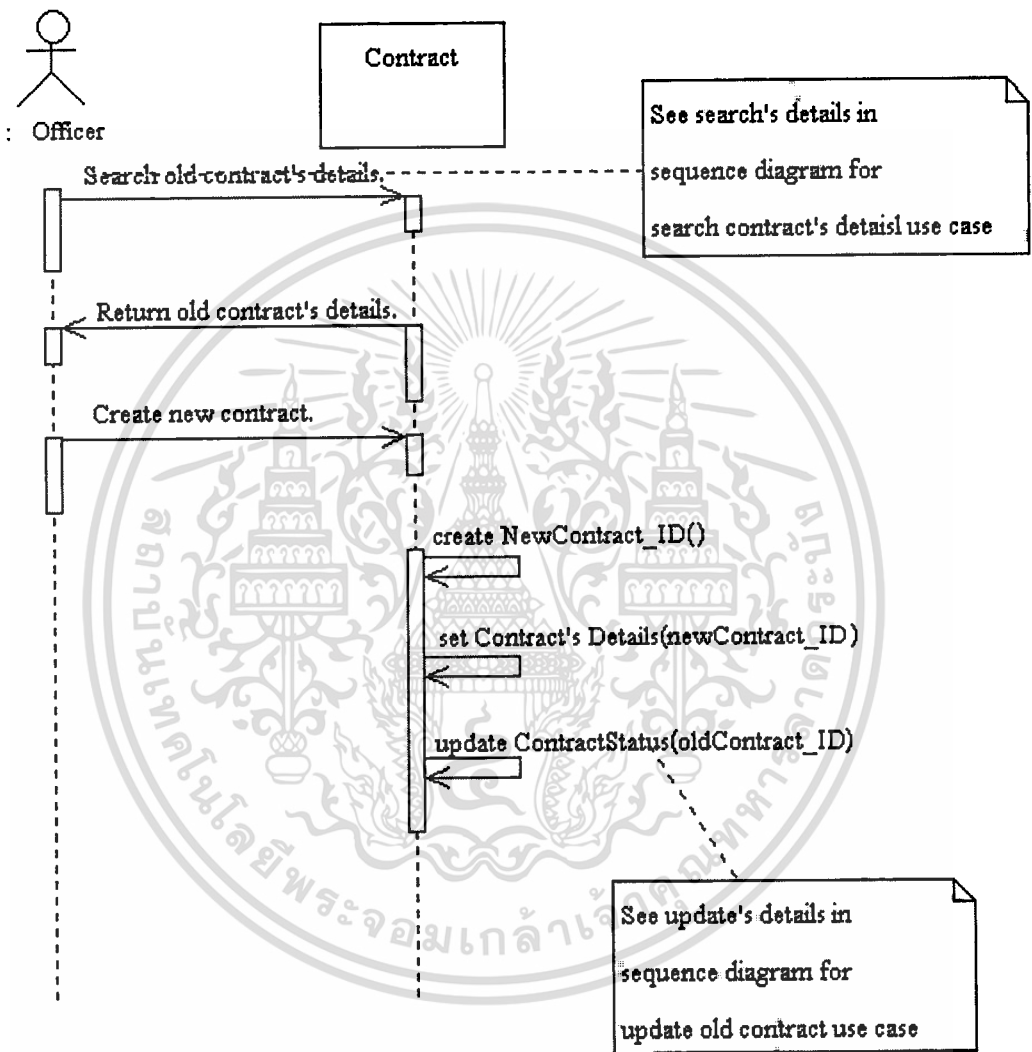
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.5 Sequence Diagram ของ Update Old Contract Use Case



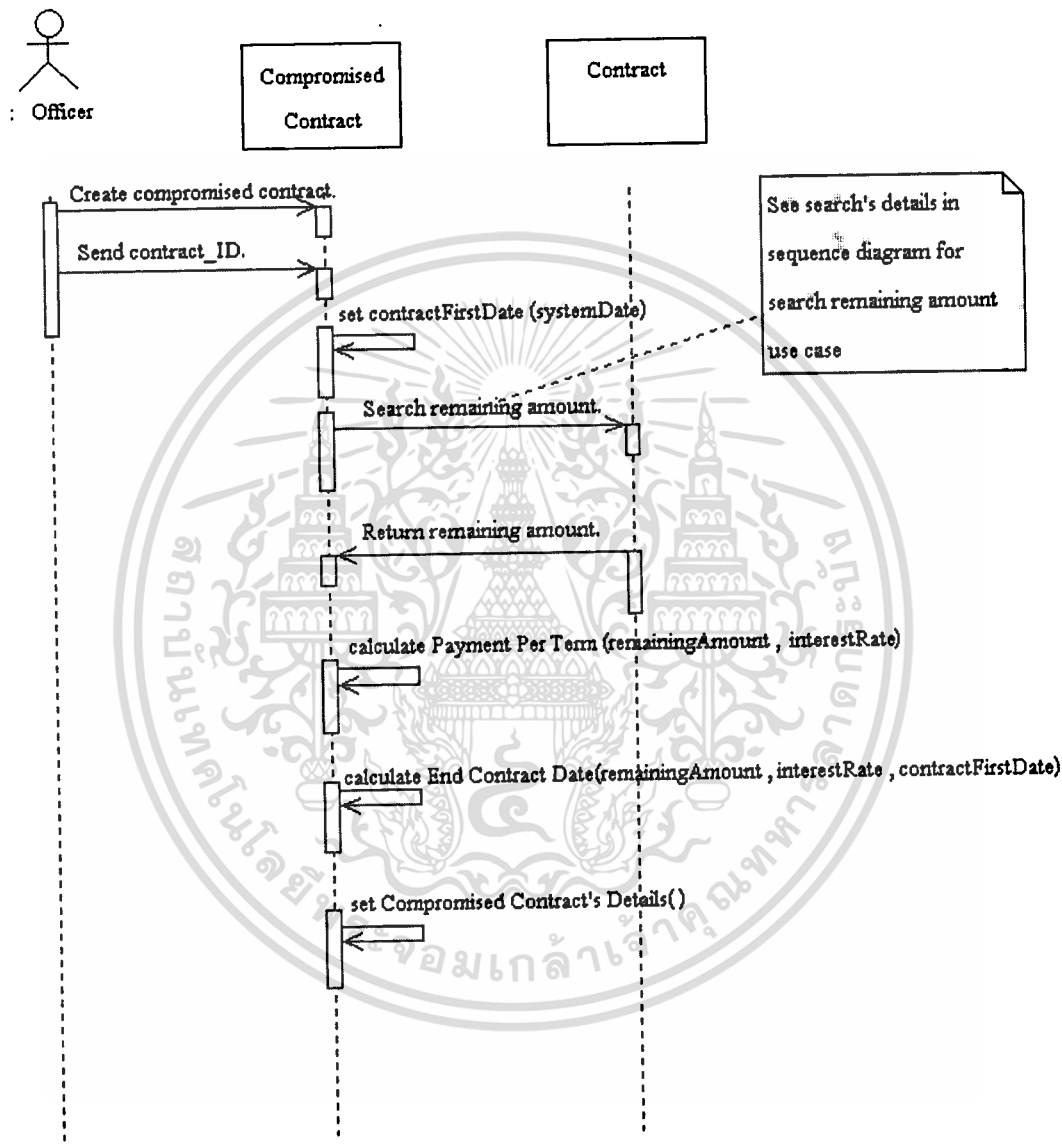
รูปที่ 6. 7 แสดง Sequence Diagram ของ Update Old Contract Use Case

6.4.6 Sequence Diagram ของ Create New Contract Use Case



รูปที่ 6. 8 แสดง Sequence Diagram ของ Create New Contract Use Case

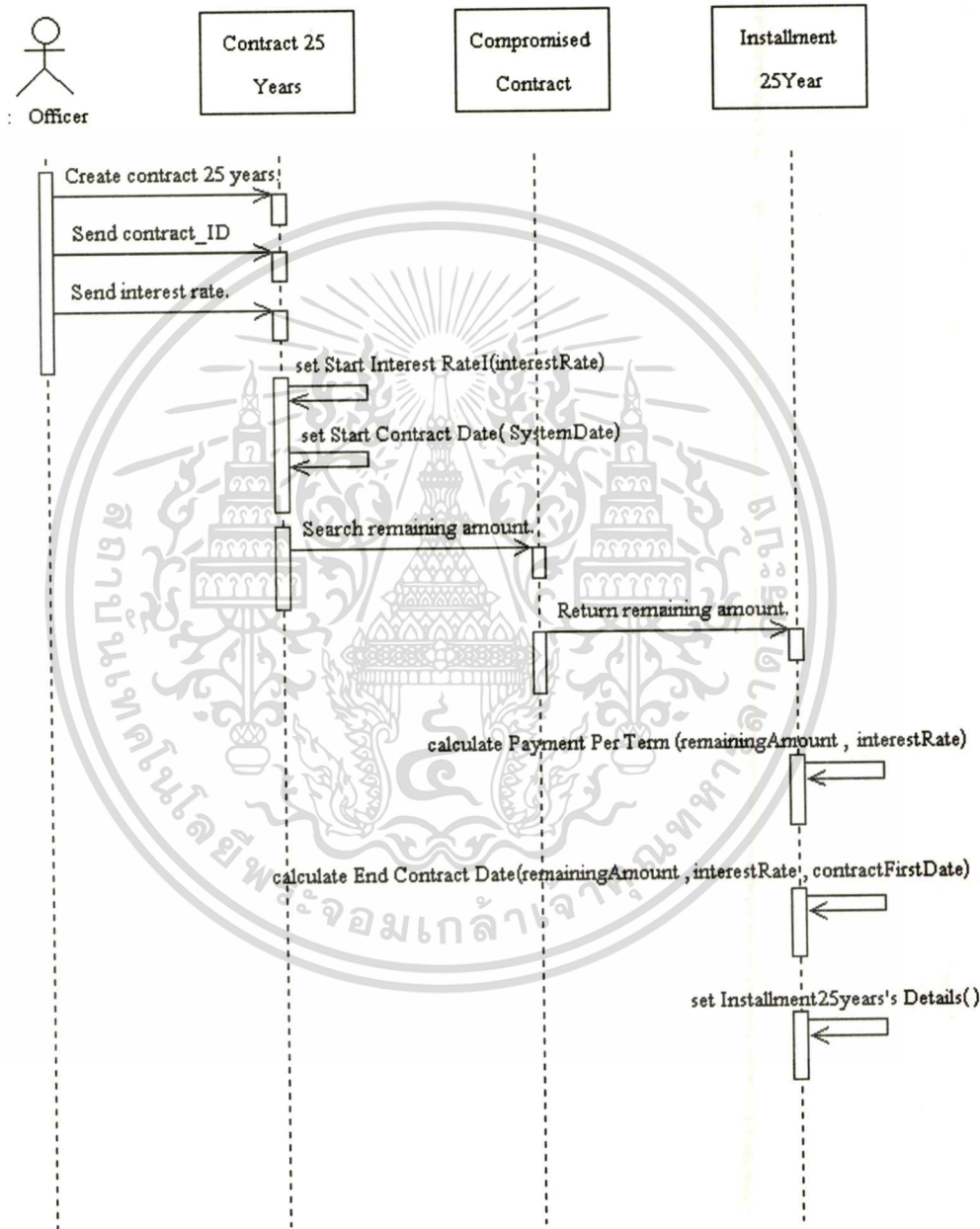
6.4.7 Sequence Diagram ของ Create Compromised Contract Use Case



รูปที่ 6. 9 แสดง Sequence Diagram ของ Create Compromised Contract Use Case

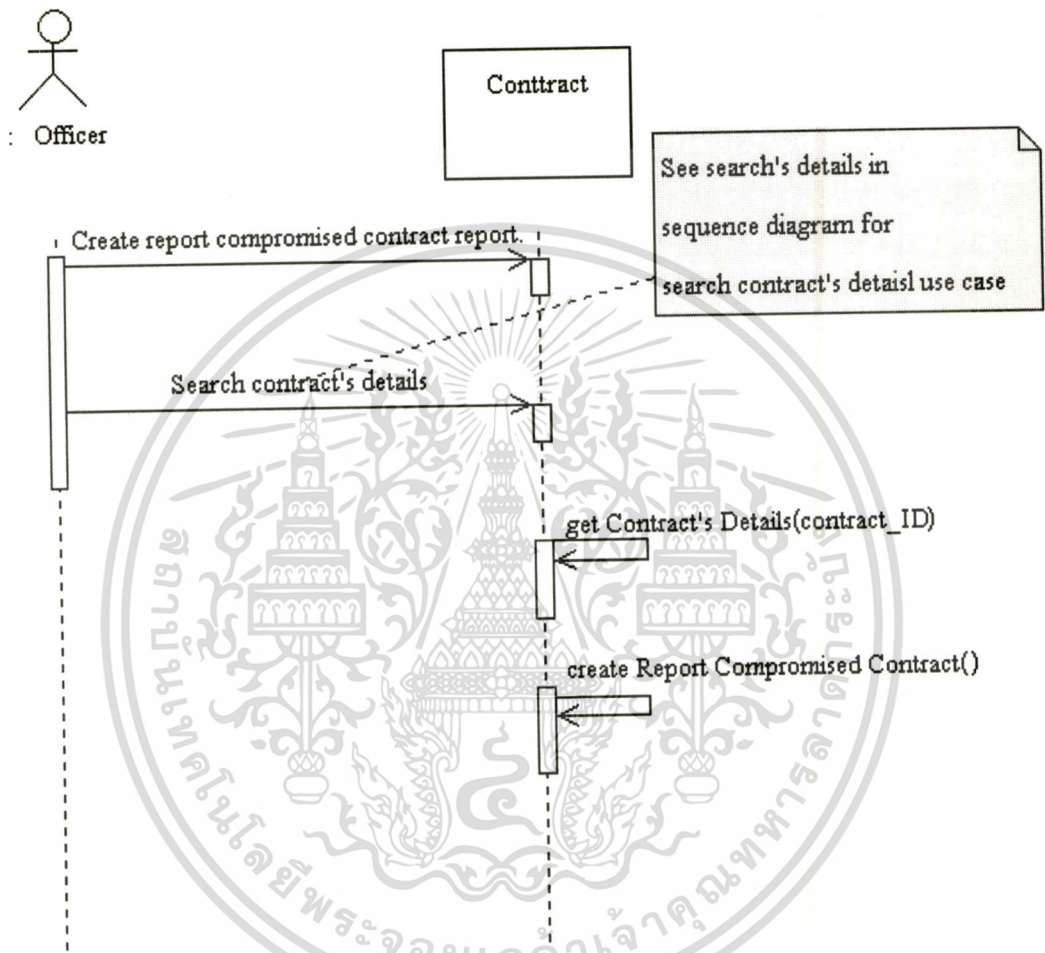


6.4.8 Sequence Diagram ของ Create Contract 25 years Use Case



รูปที่ 6. 10 แสดง Sequence Diagram ของ Create Contract 25 Years Use Case

6.4.9 Sequence Diagram ของ Create Report Use Case



รูปที่ 6. 11 แสดง Sequence Diagram ของ Create Report Use Case

6.5 Class Diagram

6.5.1 Class

ตารางที่ 6. 1 แสดง ชื่อคลาส และคำอธิบาย ของแต่ละคลาส ในระบบประนอนหนี้

| ชื่อคลาส          | วัตถุประสงค์                |
|-------------------|-----------------------------|
| 1. BuildingType   | ประเภทสิ่งปลูกสร้าง         |
| 2. BuyContract    | สัญญาเช่าซื้อ               |
| 3. BuyDownPayment | เงินค้ำประกันของการเช่าซื้อ |

| ชื่อคลาส                  | วัตถุประสงค์                                        |
|---------------------------|-----------------------------------------------------|
| 4. CompromisedContract    | สัญญาประนอมหนี้ไม่เกิน 10 ปี                        |
| 5. Contract               | ข้อมูลสัญญา                                         |
| 6. Contract 25 Years      | สัญญาเช่าซื้อไม่เกิน 25 ปี                          |
| 7. Customer               | ข้อมูลลูกค้า                                        |
| 8. InstallPayment 25Years | รายละเอียดการจ่ายเงินแต่ละงวดของสัญญาเช่าซื้อ 25 ปี |
| 9. Location               | รายละเอียดสถานที่ตั้ง                               |
| 10. Model                 | โมเดล                                               |
| 11. PaymentTx             | Transaction การรับชำระเงินแต่ละงวด                  |
| 12. PaymentTxD            | รายละเอียดการชำระเงินแต่ละงวด                       |
| 13. Project               | ข้อมูลโครงการ                                       |
| 14. Quotation             | ข้อมูลการขาย                                        |
| 15. Unit                  | ข้อมูลหน่วยอาศัย                                    |

6.5.2 Attribute ของแต่ละ คลาส

โดยจะแสดงชื่อ Attribute พร้อมคำอธิบายของแต่ละ Attribute ดังนี้

ตารางที่ 6. 2 แสดงชื่อพร้อมคำอธิบายแอทริบิวต์ ที่อยู่ในคลาสประนอมหนี้

| ชื่อคลาส        | แอทริบิวต์        | คำอธิบาย                       |
|-----------------|-------------------|--------------------------------|
| 1. BuildingType | buildingTypeID    | รหัสประเภทสิ่งปลูกสร้าง        |
|                 | buildingTypeName  | ชื่อประเภทสิ่งปลูกสร้าง        |
| 2. BuyContract  | buyInterestRate   | อัตราดอกเบี้ยต่อการเช่าซื้อ    |
|                 | buyPrincipal      | เงินต้นของการเช่าซื้อ          |
|                 | buyAmount         | จำนวนเงินทั้งหมดของการเช่าซื้อ |
|                 | downPaymentAmount | จำนวนเงินดาวน์ที่จ่ายเป็นก้อน  |



| ชื่อคลาส               | แอทริบิวต์               | คำอธิบาย                                                                                                                                                                                                           |
|------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3. BuyDownPayment      | From Term Down-Payment   | งวดที่เริ่มต้นจ่ายเงินค่างวด                                                                                                                                                                                       |
|                        | toTermDownPayment        | งวดที่สิ้นสุดจ่ายเงินค่างวด                                                                                                                                                                                        |
|                        | pricePerTermDown-Payment | จำนวนเงินค่างวดที่ชำระแต่ละงวด                                                                                                                                                                                     |
| 4. CompromisedContract | interestRate             | อัตราดอกเบี้ย                                                                                                                                                                                                      |
|                        | termPayment              | จำนวนงวดที่ต้องชำระทั้งหมด                                                                                                                                                                                         |
|                        | installPayment           | จำนวนเงินที่จ่ายต่องวด                                                                                                                                                                                             |
|                        | startContractDate        | วันที่ทำสัญญาประนอมหนี้                                                                                                                                                                                            |
| 5. Contract            | contractID               | เลขที่สัญญา                                                                                                                                                                                                        |
|                        | contractDate             | วันที่ทำสัญญา                                                                                                                                                                                                      |
|                        | contractStatus           | สถานะของสัญญา <ul style="list-style-type: none"><li>● U : ยังใช้งานอยู่</li><li>● A : ยกเลิกเมื่อสิ้นสุดสัญญา</li><li>● T:หยุดเมื่อคำนวณงวดสัญญาใหม่</li><li>● C : เปลี่ยนสัญญา</li><li>● X : ประนอมหนี้</li></ul> |
|                        | paymentType              | ประเภทการชำระเงิน <ul style="list-style-type: none"><li>● C : ผ่านสช. (เงินสด)</li><li>● B : ผ่านธนาคาร</li></ul>                                                                                                  |
|                        | pricePerTerm             | เงินที่ต้องชำระต่องวด                                                                                                                                                                                              |
|                        | term                     | จำนวนงวดที่ต้องชำระ                                                                                                                                                                                                |
|                        | contractType             | ประเภทสัญญา <ul style="list-style-type: none"><li>● R : สัญญาเช่า</li><li>● B : สัญญาเช่าซื้อ</li><li>● L : สัญญาสินเชื่อดำเนินอาคาร</li></ul>                                                                     |
|                        | contractIDNew            | เลขที่สัญญาใหม่                                                                                                                                                                                                    |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



| ชื่อคลาส                   | แอทริบิวต์        | คำอธิบาย                           |
|----------------------------|-------------------|------------------------------------|
| 6. Contract 25 Years       | startIntarestRate | อัตราดอกเบี้ยเริ่มต้นทำสัญญา 25 ปี |
|                            | startContractDate | วันที่ทำสัญญา 25 ปี                |
| 7. Customer                | customerID        | เลขที่บัตรประจำตัวประชาชน          |
|                            | title             | คำนำหน้าชื่อ                       |
|                            | firstName         | ชื่อ                               |
|                            | lastName          | สกุล                               |
|                            | regisAddress      | ที่อยู่ตามบัตรประชาชน              |
|                            | currentAddress    | ที่อยู่ปัจจุบัน                    |
|                            | telephone         | หมายเลขโทรศัพท์                    |
| 8. Install Payment 25Years | formTermPayment   | งวดที่เริ่มต้นจ่ายค่าเช่าซื้อ      |
| 9. Location                | LocationID        | รหัสสถานที่ตั้ง                    |
|                            | LocationName      | ชื่อที่ตั้ง                        |
| 10 Model                   | ModelID           | รหัส โมเดล                         |
|                            | ModelName         | ชื่อ โมเดล                         |
| 11 PaymentTx               | PaymenTxID        | รหัสสำหรับแต่ละ Transaction        |
|                            | PaymentDate       | วันที่ของ Transaction ที่รับชำระ   |
|                            | PaymentTerm       | งวดที่ชำระ                         |
|                            | ContractID        | เลขที่สัญญา                        |
| 12 PaymentTxID             | PaymentTxID       | รหัสสำหรับแต่ละ Transaction        |
|                            | AccountID         | รหัสบัญชี เช่น 0520                |
|                            | PaymentPrice      | จำนวนเงินที่ชำระต่อรายการบัญชี     |
| 13 Project                 | ProjectID         | รหัสโครงการ                        |
|                            | ProjectName       | ชื่อโครงการ                        |
| 14. Quotation              | ProjectID         | รหัสโครงการ                        |
|                            | ModelID           | รหัสโมเดล                          |

| ชื่อคลาส | แอททริบิวต์ | คำอธิบาย    |
|----------|-------------|-------------|
| 15. Unit | ProjectID   | รหัสโครงการ |
|          | ModelID     | รหัสโมเดล   |
|          | HouseNo     | เลขที่บ้าน  |

6.5.3 Method

โดยจะแสดงชื่อ Method, Parameter พร้อมคำอธิบายของแต่ละ Method และ Algorithm สำหรับ Method ที่มีความซับซ้อนดังนี้

Class BuyContract

Public Operations:

getAmountbuy (contractID : CHAR)

แสดงราคาเช่าซื้อทั้งหมด

ในการค้นหา ราคาเช่าซื้อทั้งหมด จะต้องทำการ select ราคาเช่าซื้อทั้งหมด จาก ตาราง BuyContract โดยจะใช้ contractID เป็นคีย์ในการค้นหา

Function getAmountBuy(cont\_id contract.contractid%type)  
return paymenttransaction.paymentterm%type;

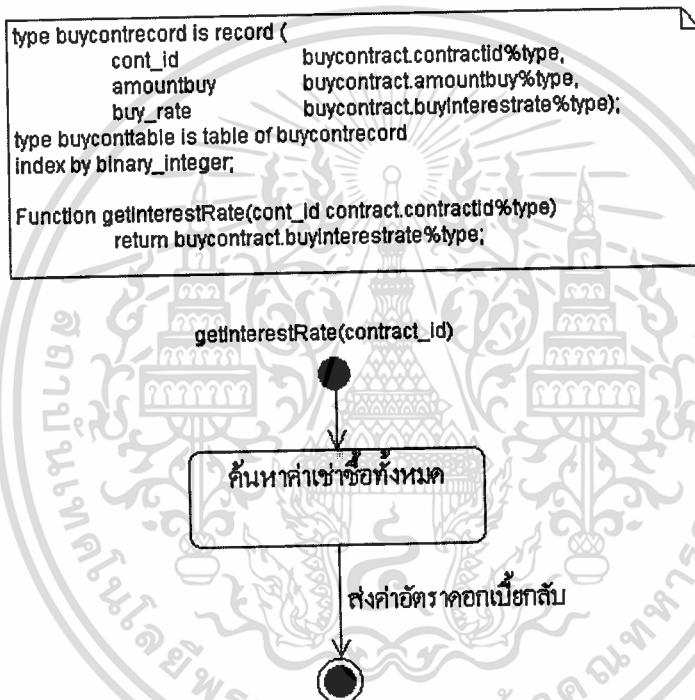


รูปที่ 6. 12 แสดง Algorithm ของ getAmountbuy Method

getInterestRate (contractID : CHAR)

แสดงอัตราดอกเบี้ยสัญญาเช่าซื้อ

ในการค้นหา อัตราดอกเบี้ยสัญญาเช่าซื้อ จะต้องทำการ select อัตราดอกเบี้ยเช่าซื้อ จากตาราง BuyContract โดยจะใช้ contractID เป็นคีย์ในการค้นหา



รูปที่ 6. 13 แสดง Algorithm ของ getInterestRate Method

getInterestTotal (contractID : CHAR)

แสดงจำนวนดอกเบี้ยทั้งหมดของสัญญาเช่าซื้อ

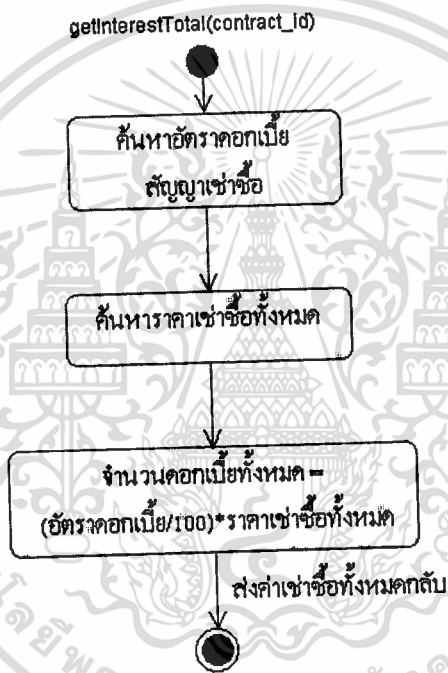
ในการค้นหา จำนวนดอกเบี้ยทั้งหมด จะต้องทำการ select อัตราดอกเบี้ยเช่าซื้อ และ ราคาเช่าซื้อทั้งหมดจากตาราง BuyContract โดยจะใช้ contractID เป็นคีย์ในการค้นหา จากนั้นจึงทำการคำนวณหาค่า ดอกเบี้ยทั้งหมด

```

type buycontractrecord is record (
    cont_id          buycontract.contractid%type,
    amountbuy        buycontract.amountbuy%type,
    buy_rate         buycontract.buyinterestrate%type);
type buyconttable is table of buycontractrecord
index by binary_integer;

Function getInterestTotal(cont_id contract.contractid%type)
return buycontract.amountbuy%type;

```



รูปที่ 6. 14 แสดง Algorithm ของ getInterestTotal Method

Class Contract

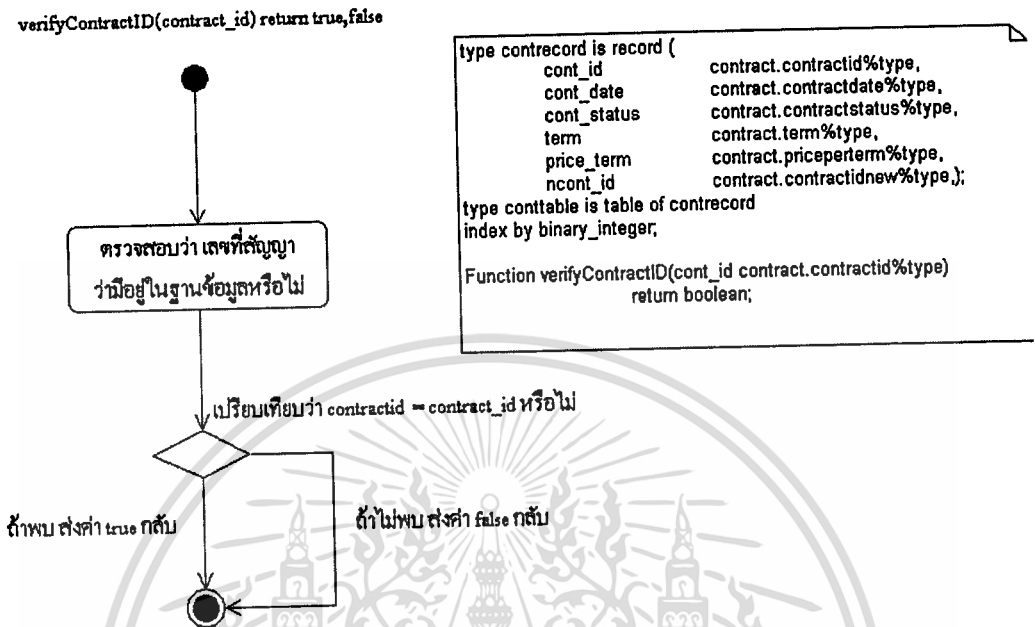
Public Operations:

verifyContractID (contractID : CHAR)

ตรวจสอบเลขที่สัญญา

ตรวจสอบเลขที่สัญญาที่ผู้ใช้ป้อนเข้ามาว่า มีอยู่ในฐานข้อมูลหรือไม่



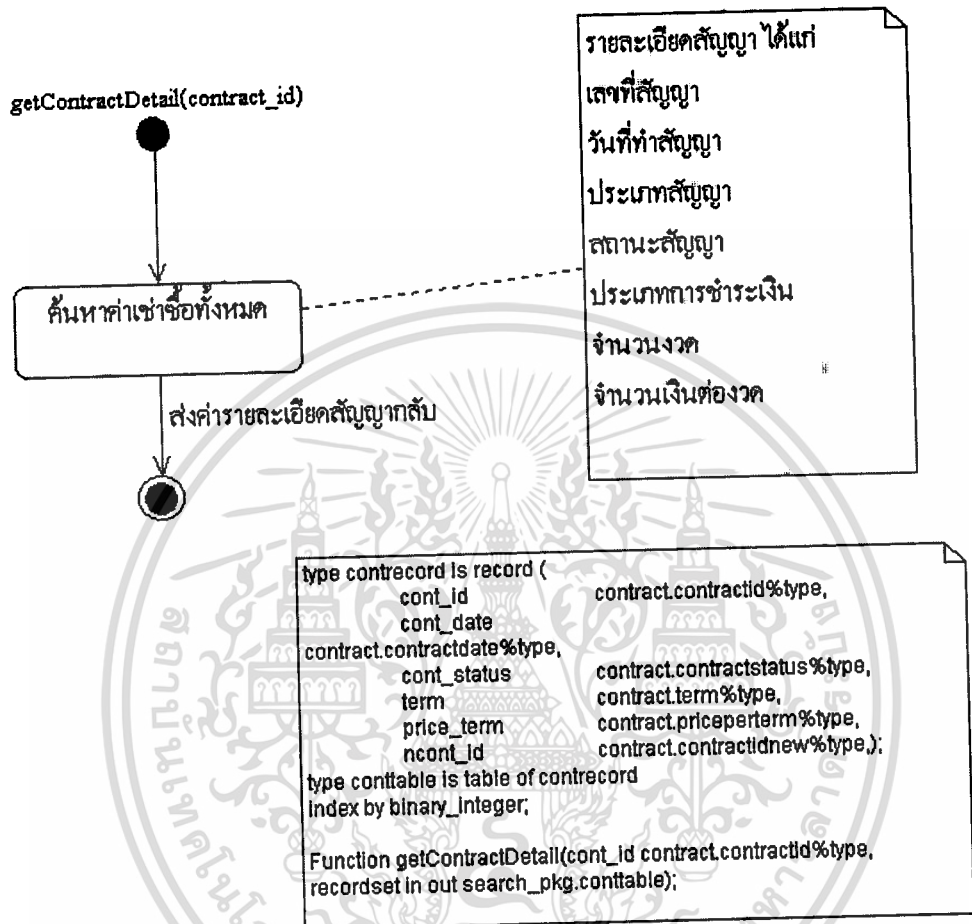


รูปที่ 6. 15 แสดง Algorithm ของ verifyContractID Method

getContractDetail (contractID : CHAR)

แสดงรายละเอียดสัญญา

ค้นหารายละเอียดสัญญา จาก ตาราง Contract โดยจะทำการค้นหารายละเอียดสัญญา ซึ่งได้แก่ เลขที่สัญญา วันที่ทำสัญญา ประเภทสัญญา สถานะสัญญา ประเภทการชำระเงิน จำนวนงวด จำนวนค่าเช่าซื้อต่องวด และเลขที่สัญญาใหม่ (ในกรณีที่มีการเปลี่ยนสัญญา) โดยในการค้นหา จะทำการ select ค่าของรายละเอียดสัญญา จาก ตารางสัญญา โดยใช้ เลขที่สัญญาเป็นคีย์ในการค้นหา

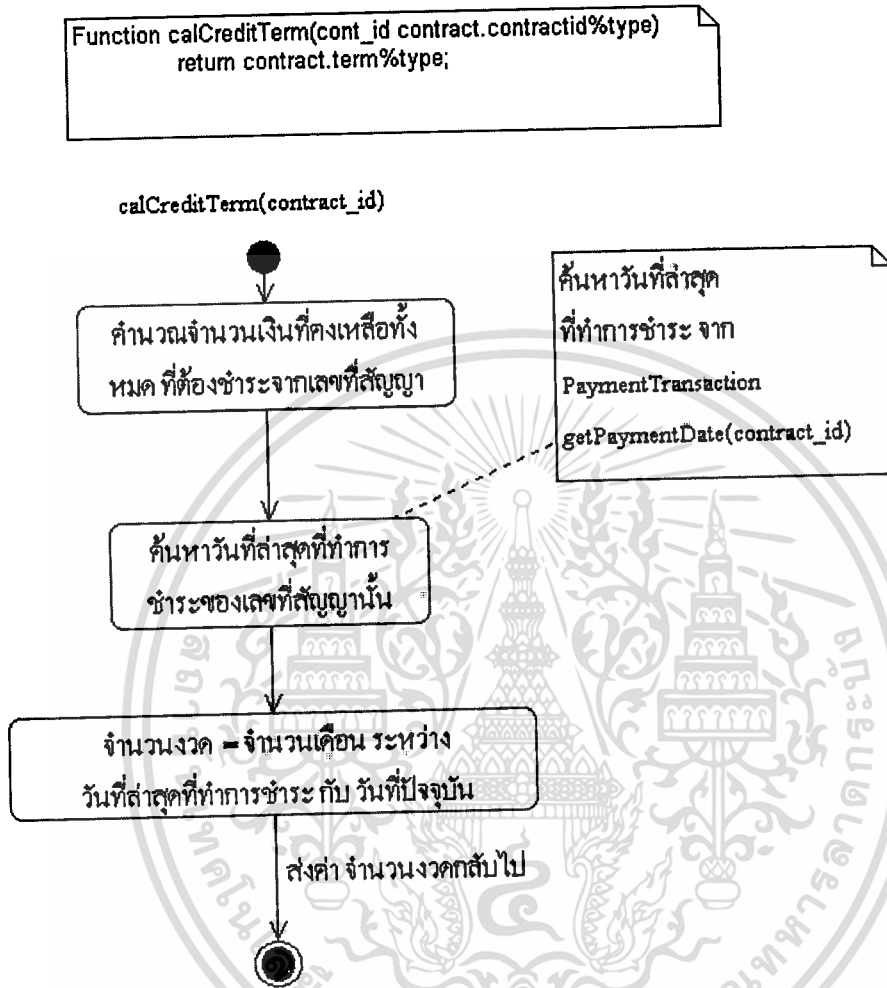


รูปที่ 6. 16 แสดง Algorithm ของ getContractDetil Method

calCreditTerm (contractID : CHAR)

คำนวณจำนวนงวดที่ค้างชำระ

การคำนวณงวดค้างชำระ จะทำการ เรียก Function getPaymentDate ของตาราง PaymentTransaction เพื่อหาวันที่ล่าสุดที่ชำระ จากนั้นจะทำการหาจำนวนเดือนระหว่างวันที่ที่จ่ายล่าสุด กับ วันที่ปัจจุบัน

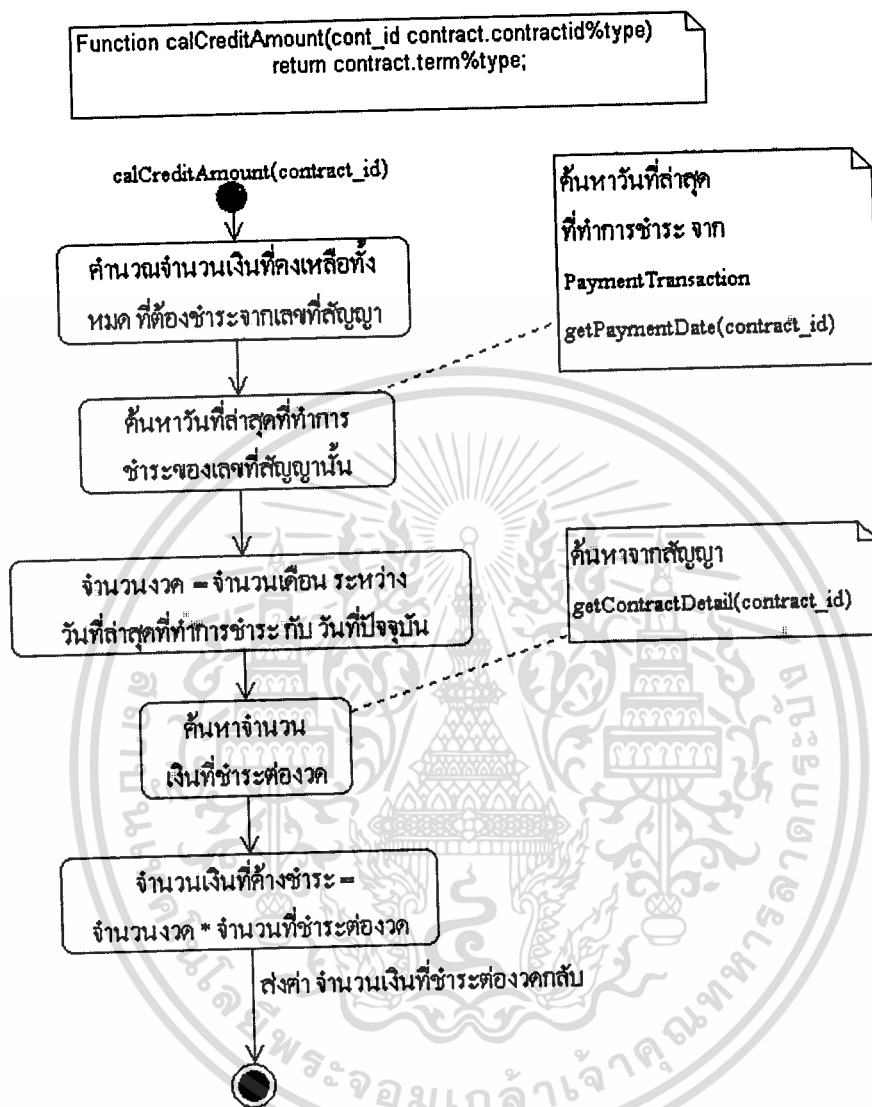


รูปที่ 6. 17 แสดง Algorithm ของ calCreditTerm Method

calCreditAmount (contractID : CHAR)

คำนวณจำนวนเงินที่ค้างชำระ

การคำนวณจำนวนค้างชำระ จะทำการเรียก Function getPaymentDate ของตาราง PaymentTransaction จากนั้นจะทำการหาจำนวนเดือนระหว่างวันที่ที่จ่ายล่าสุด กับ วันที่ปัจจุบัน เพื่อให้ได้งวดค้างชำระ จากนั้นทำการ หาค่าเข้าซื้อต้องงวดจากตาราง Contract แล้ว นำมาคำนวณ หาค่าจำนวนเงินที่ค้างชำระ



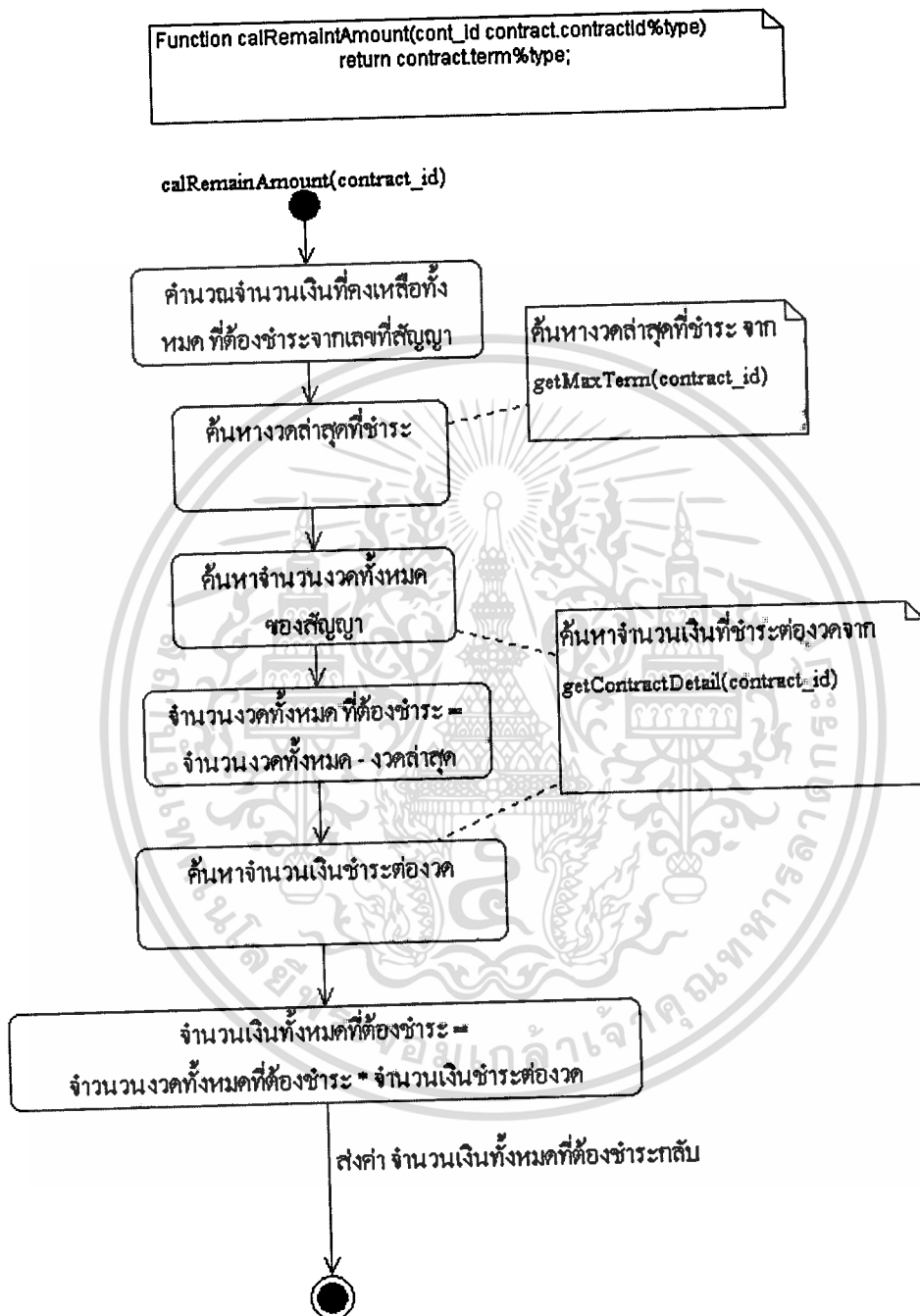
รูปที่ 6. 18 แสดง Algorithm ของ calCreditAmount Method

calRemainAmount (contractID : CHAR)

คำนวณจำนวนทั้งหมดที่ต้องชำระ

การคำนวณจำนวนทั้งหมดที่ต้องชำระ จะทำการเรียก Function getMaxTerm ของตาราง PaymentTransaction เพื่อหาจำนวนงวดล่าสุดที่ชำระ จากนั้นทำการ หาค่าเช่าซื้อต้องงวด และ จำนวนงวดทั้งหมด จากตาราง Contract เพื่อมาทำการคำนวณหา จำนวนทั้งหมดที่ต้องชำระ





รูปที่ 6. 19 แสดง Algorithm ของ calRemainAmount Method

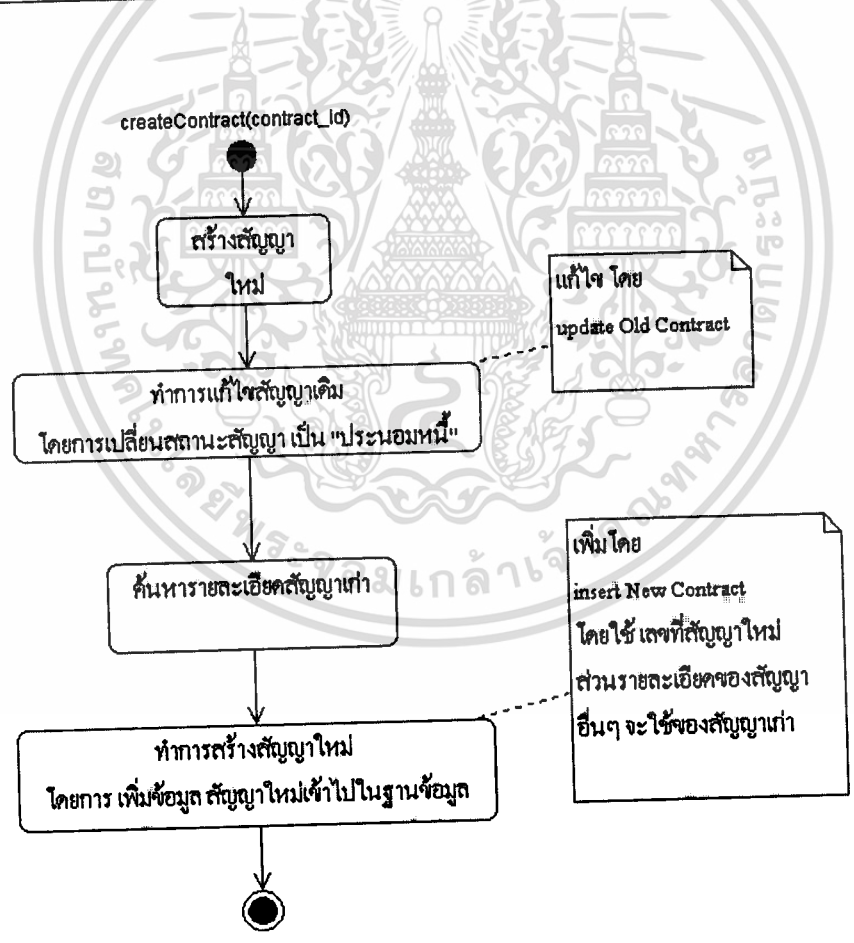
createContract (contractID : CHAR)

สร้างสัญญาใหม่

ในการสร้างสัญญาใหม่ จะทำการ แก้ไข สถานะสัญญาเดิม และทำการเพิ่มข้อมูลเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าสัญญาใหม่ลงในฐานะข้อมูล

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
type contracord is record (  
    cont_id          contract.contractid%type,  
    cont_date        contract.contractdate%type,  
    price_term       contract.priceperterm%type,  
    term             contract.term%type,  
    cont_status      contract.contractstatus%type,  
    ncont_id         contract.contractid%type,  
    cust_id          contract.customerid%type,  
    proj_id          contract.projectid%type,  
    mod_id           contract.modelid%type,  
    house            contract.houseno%type);  
  
type conttable is table of contracord  
index by binary_integer;  
  
Procedure changelDContract(cont_id contract.contractid%type,  
    ncont_id contract.contractidnew%type,  
    cont_status contract.contractstatus%type,  
    cont_date contract.contractdate%type);
```



รูปที่ 6. 20 แสดง Algorithm ของ createContract Method

## Compromised Contract

### Public Operations:

CreateCompromised (contractID : CHAR)

สร้างสัญญาประนอมหนี้

ในการสร้างสัญญาประนอมหนี้ จะทำการเพิ่มข้อมูลสัญญาประนอมหนี้ ซึ่ง จะทำการเพิ่มข้อมูลรายละเอียดสัญญา ดังนี้ เลขที่สัญญา วันที่ทำสัญญา จำนวนเงินที่ชำระต่องวด จำนวนงวด และอัตราดอกเบี้ยประนอมหนี้ ลงในตาราง Compromised ในฐานข้อมูล

## Contract25Years

### Public Operations:

CreateContract25Years (contractID : CHAR)

สร้างสัญญาเช่าซื้อ 25 ปี

ในการสร้างสัญญาเช่าซื้อ 25 ปี จะทำการทำการเพิ่มข้อมูลสัญญาเช่าซื้อ 25 ปี ซึ่ง จะทำการเพิ่มข้อมูลรายละเอียดสัญญา ดังนี้ เลขที่สัญญา วันที่ทำสัญญา จำนวนงวด และ อัตราดอกเบี้ยเริ่มต้น ลงในตาราง Contract 25 Years ในฐานข้อมูล

## CustomerInformation

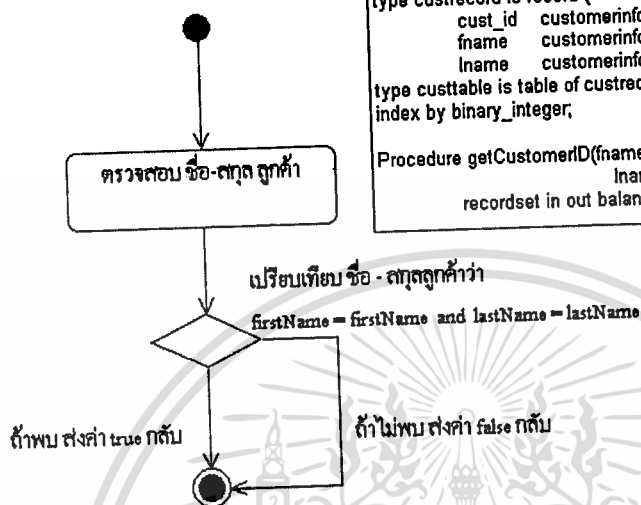
### Public Operations:

verifyCustomerName (firstName : CHAR , lastName : CHAR)

ตรวจสอบชื่อลูกค้า

ตรวจสอบชื่อลูกค้าที่ผู้ใช้ป้อนเข้ามาว่า มีอยู่ในฐานข้อมูลหรือไม่ โดยผู้ใช้จะต้องป้อน ทั้ง ชื่อ-สกุล ลูกค้า เพื่อจะมาทำการตรวจสอบ

verifyCustomerName(firstName,lastName)



```

type custrecord is record (
    cust_id    customerinformation.customerid%type,
    fname     customerinformation.firstname%type,
    lname     customerinformation.lastname%type);
type custtable is table of custrecord
index by binary_integer;

Procedure getCustomerID(fname customerinformation.firstname%type,
    lname customerinformation.lastname%type,
    recordset in out balance_search_pkg.custtable);
  
```

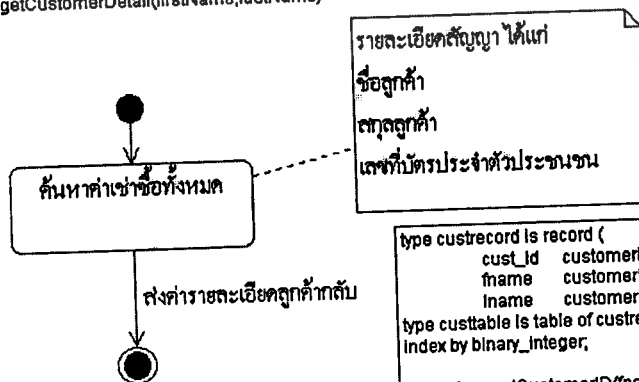
รูปที่ 6. 21 แสดง Algorithm ของ verifyCustomerName Method

getCustomerDetail (firstName : CHAR , lastName : CHAR)

แสดงรายละเอียดลูกค้า

ค้นหารายละเอียดลูกค้า จาก ตาราง CustomerInformation โดยจะทำการค้นหารายละเอียดลูกค้า ซึ่งได้แก่ รหัสลูกค้า โดยในการค้นหา จะทำการ select ค่าของรายละเอียดสัญญา จาก ตาราง CustomerInformation โดยใช้ ชื่อ-สกุล ลูกค้า เป็นคีย์ในการค้นหา

getCustomerDetail(firstName,lastName)



รายละเอียดสัญญา ได้แก่  
ชื่อลูกค้า  
สกุลลูกค้า  
เลขที่บัตรประจำตัวประชาชน

```

type custrecord is record (
    cust_id    customerinformation.customerid%type,
    fname     customerinformation.firstname%type,
    lname     customerinformation.lastname%type);
type custtable is table of custrecord
index by binary_integer;

Procedure getCustomerID(fname customerinformation.firstname%type,
    lname customerinformation.lastname%type,
    recordset in out balance_search_pkg.custtable);
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่า **รูปที่ 6. 22 แสดง Algorithm ของ getCustomerDetail Method**



PaymentTransaction

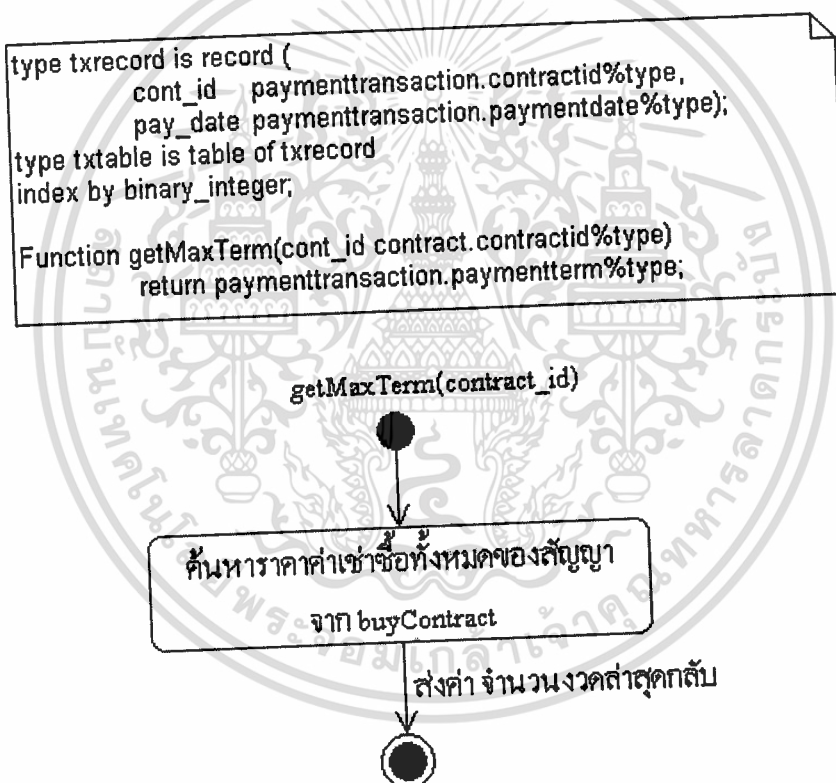
*Public Operations:*

getMaxTerm (contractID : CHAR)

แสดงจำนวนงวดล่าสุดที่ชำระ

ทำการค้นหาจำนวนงวดล่าสุดที่ชำระ โดยจะทำการ Select งวดล่าสุดจาก ตาราง

PaymentTransaction



รูปที่ 6.23 แสดง Algorithm ของ getMaxTerm Method

getPaymentDate (contractID : CHAR)

แสดงวันที่ล่าสุดที่ชำระ

ทำการค้นหาวันที่ล่าสุดที่ชำระ โดยจะทำการ Select งวดล่าสุดจาก ตาราง

PaymentTransaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function getPaymentDate(cont_id contract.contractid%type)  
return paymenttransaction.paymentterm%type;
```

getPaymentDate(contract\_id)

ค้นหาราคาค่าเช่าซื้อทั้งหมดของสัญญา  
จาก buyContract

ส่งค่าวันที่ล่าสุดกลับ

รูปที่ 6. 24 แสดง Algorithm ของ getPaymentDate Method

Project

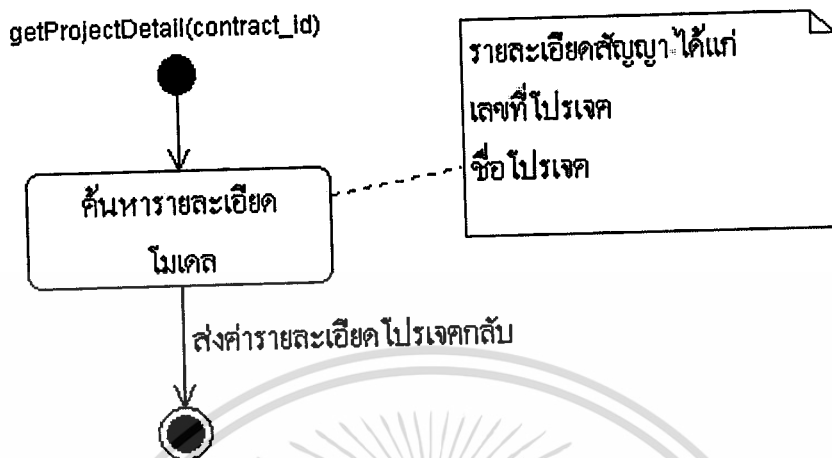
Public Operations:

getProjectDetail (contractID : CHAR)

แสดงรายละเอียดโครงการ

ทำการค้นรายละเอียดโครงการได้แก่ รหัสโครงการ และ ชื่อโครงการ จาก ตาราง

Project



รูปที่ 6. 25 แสดง Algorithm ของ `getProjectDetail Method`

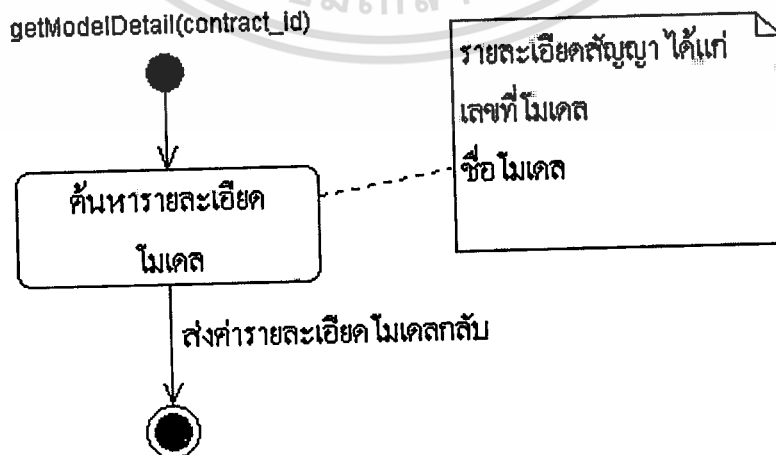
Model

*Public Operations:*

`getModelDetail (contractID : CHAR)`

แสดงรายละเอียด โมเดล

ทำการค้นรายละเอียดโมเดล ได้แก่ รหัสโมเดล และ ชื่อโมเดล จาก ตาราง Model

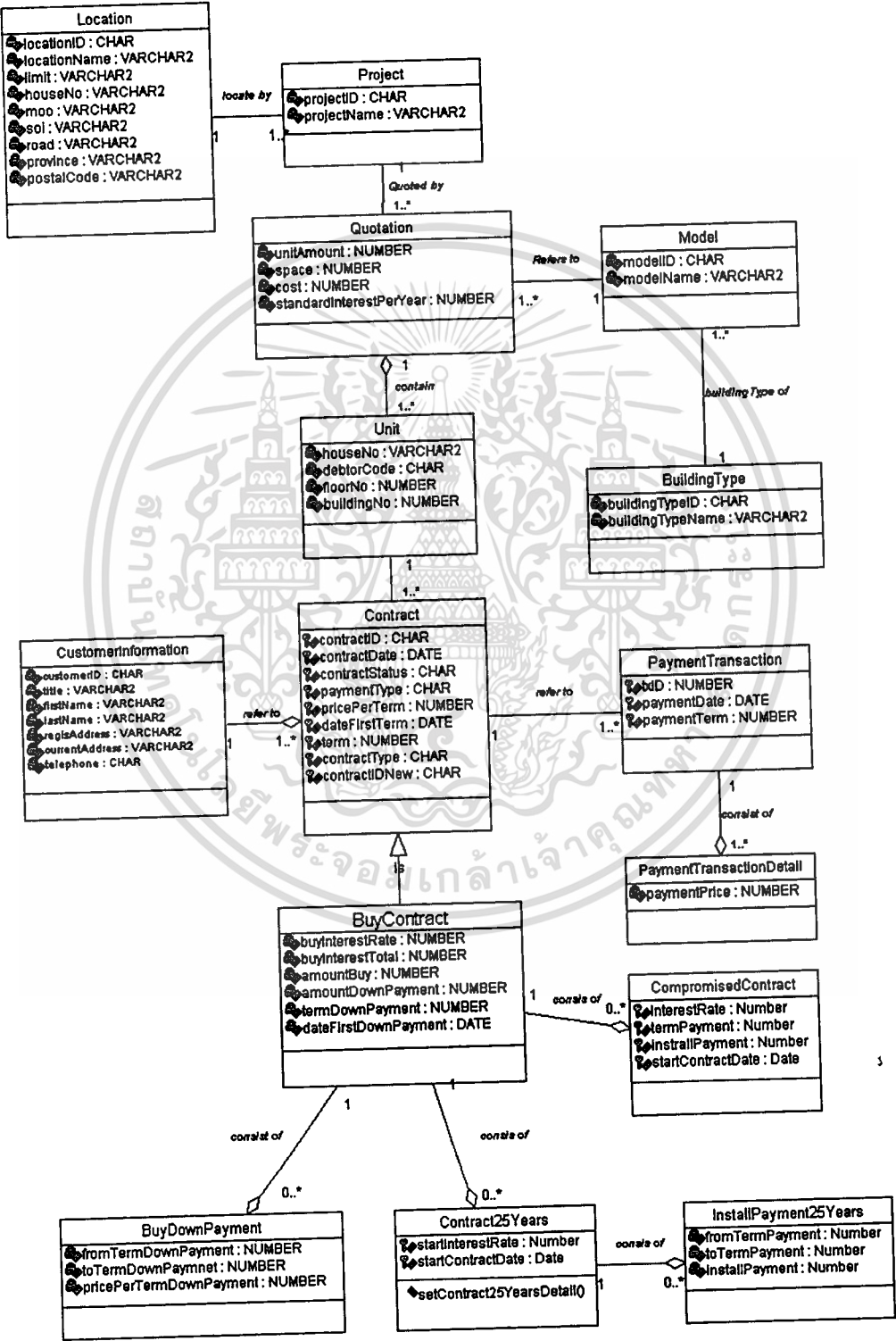


รูปที่ 6. 26 แสดง Algorithm ของ `getModelDetail Method`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่อนุญาตให้เผยแพร่ข้อมูลนี้ไปยังบุคคลภายนอก และข้อมูลนี้รวมถึงข้อมูลลิขสิทธิ์ที่สงวนไว้ใช้

6.5.4 Class Diagram in Application Logic

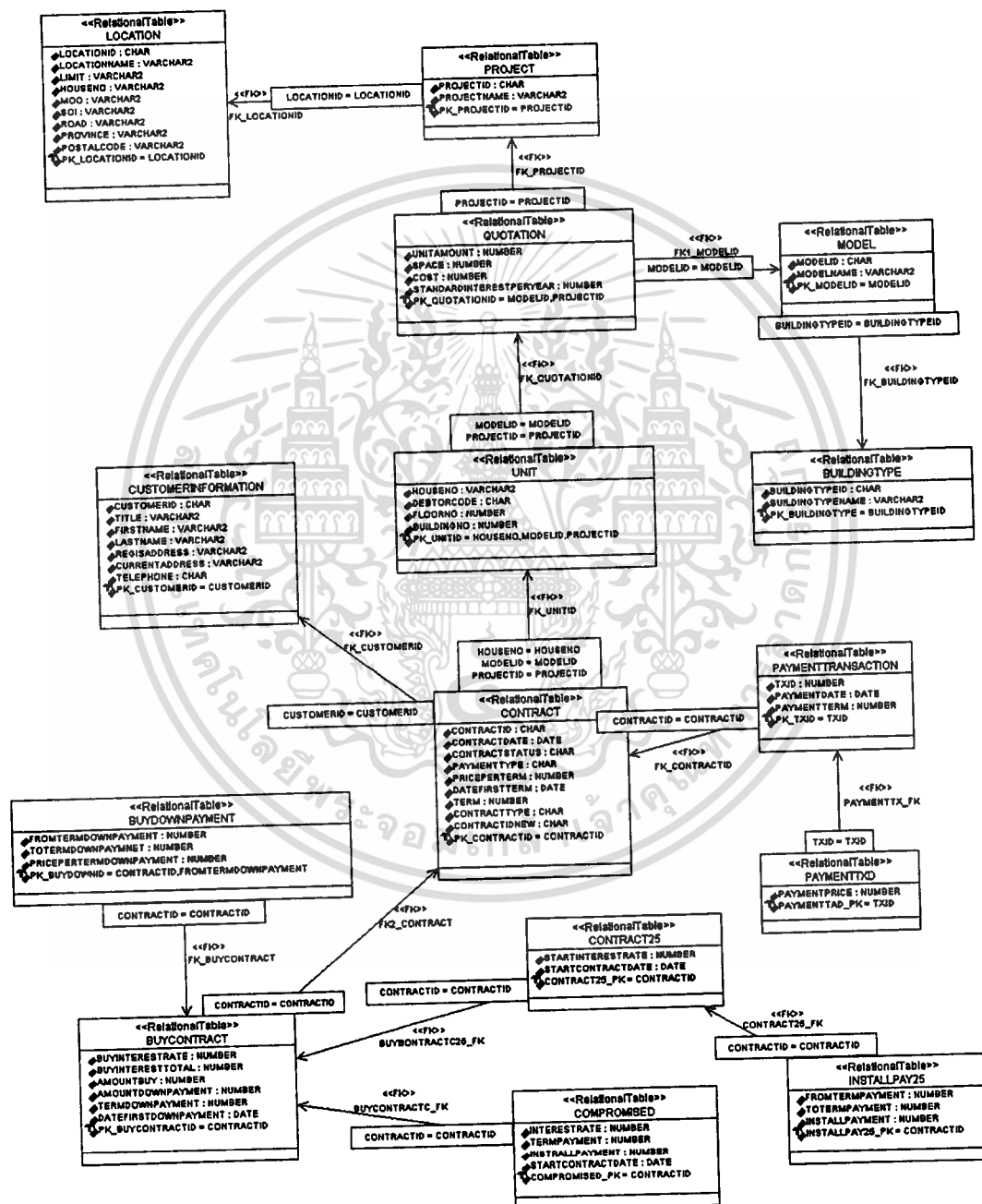


รูปที่ 6. 27 แสดง Class Diagram ใน Application Logic นั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## 6.5.5 Class Diagram in Access Layer

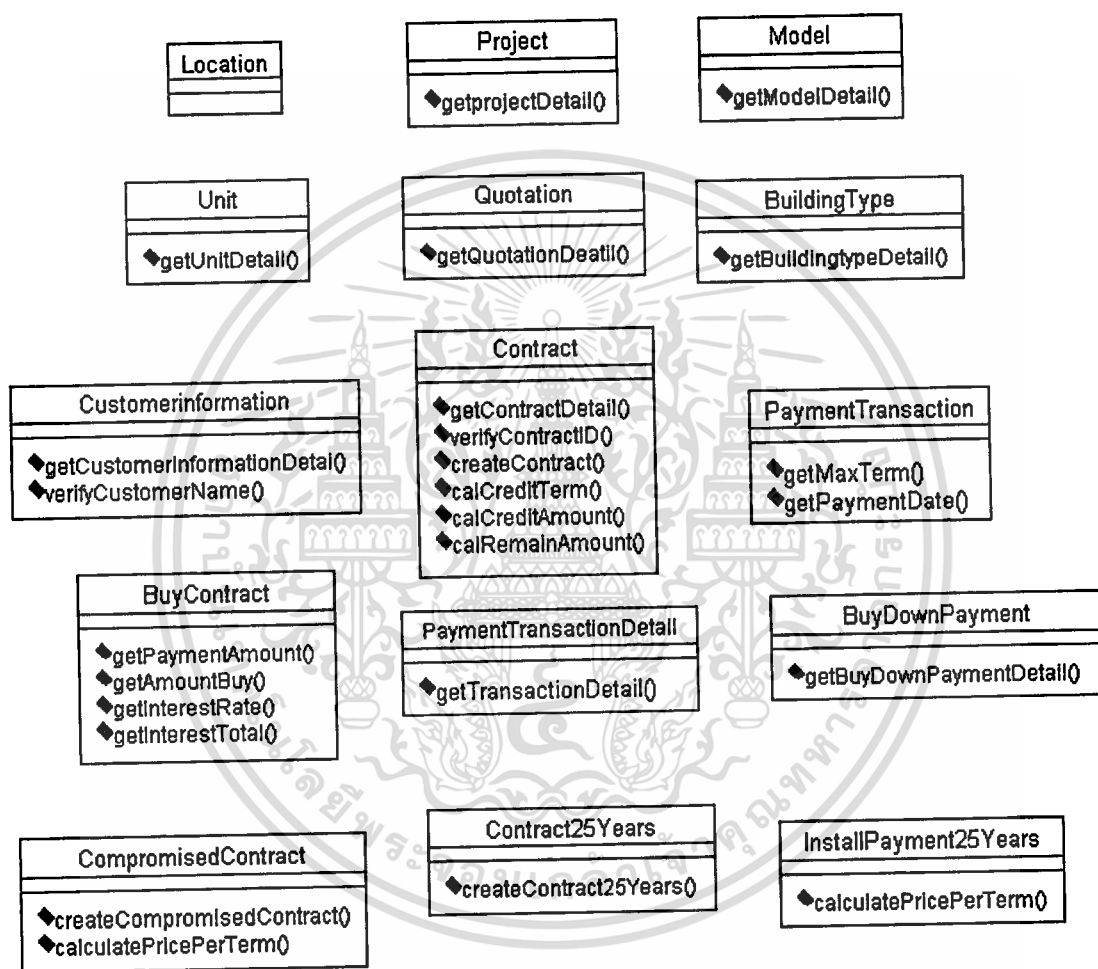
### 6.5.5.1 Schema



รูปที่ 6. 28 แสดง Schema ของระบบฐานข้อมูลโดยใช้ Class Diagram

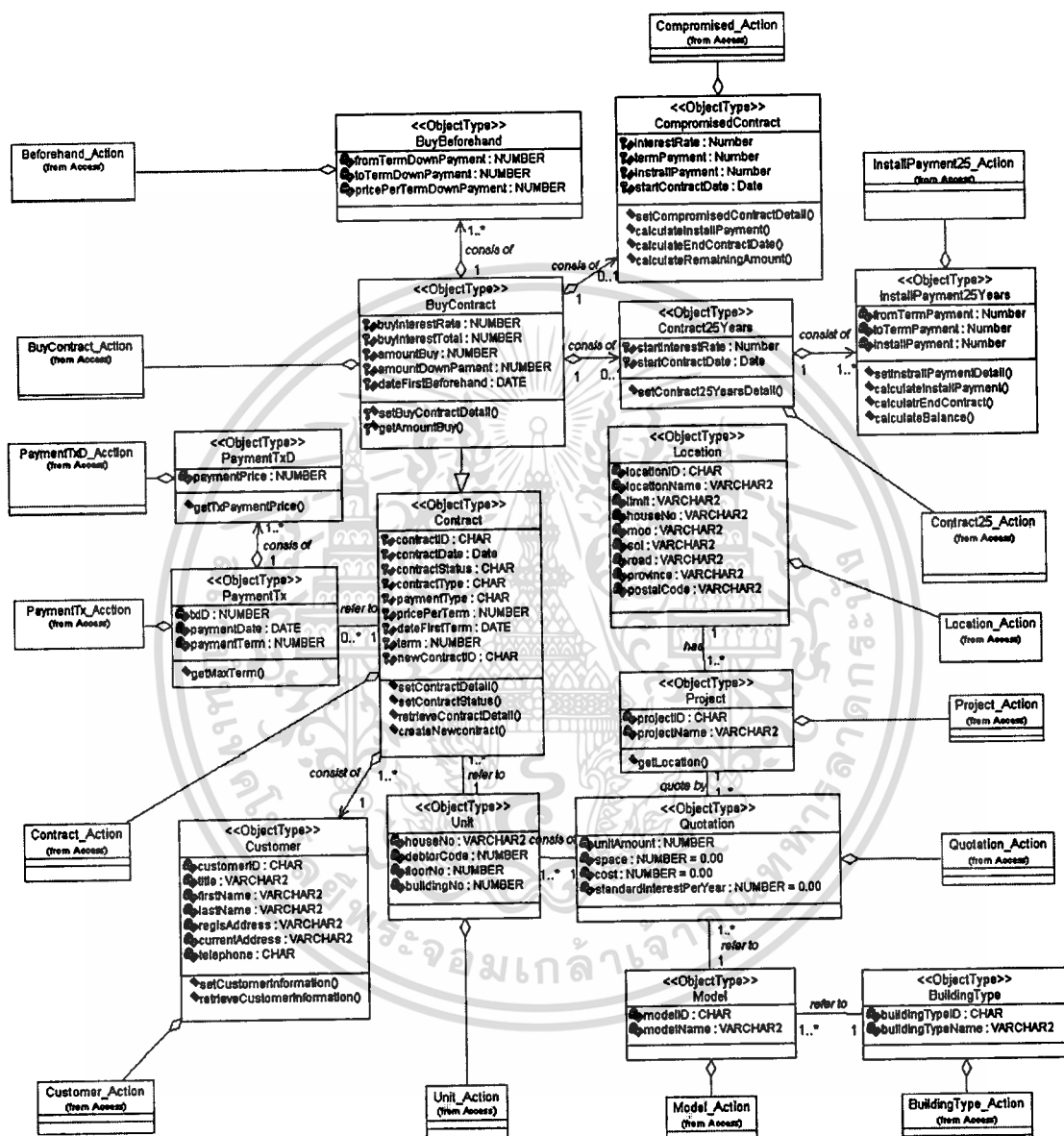
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.5.5.2 Class Diagram in Access Layer



รูปที่ 6. 29 แสดง Class Diagram ในส่วนของ Access Layer

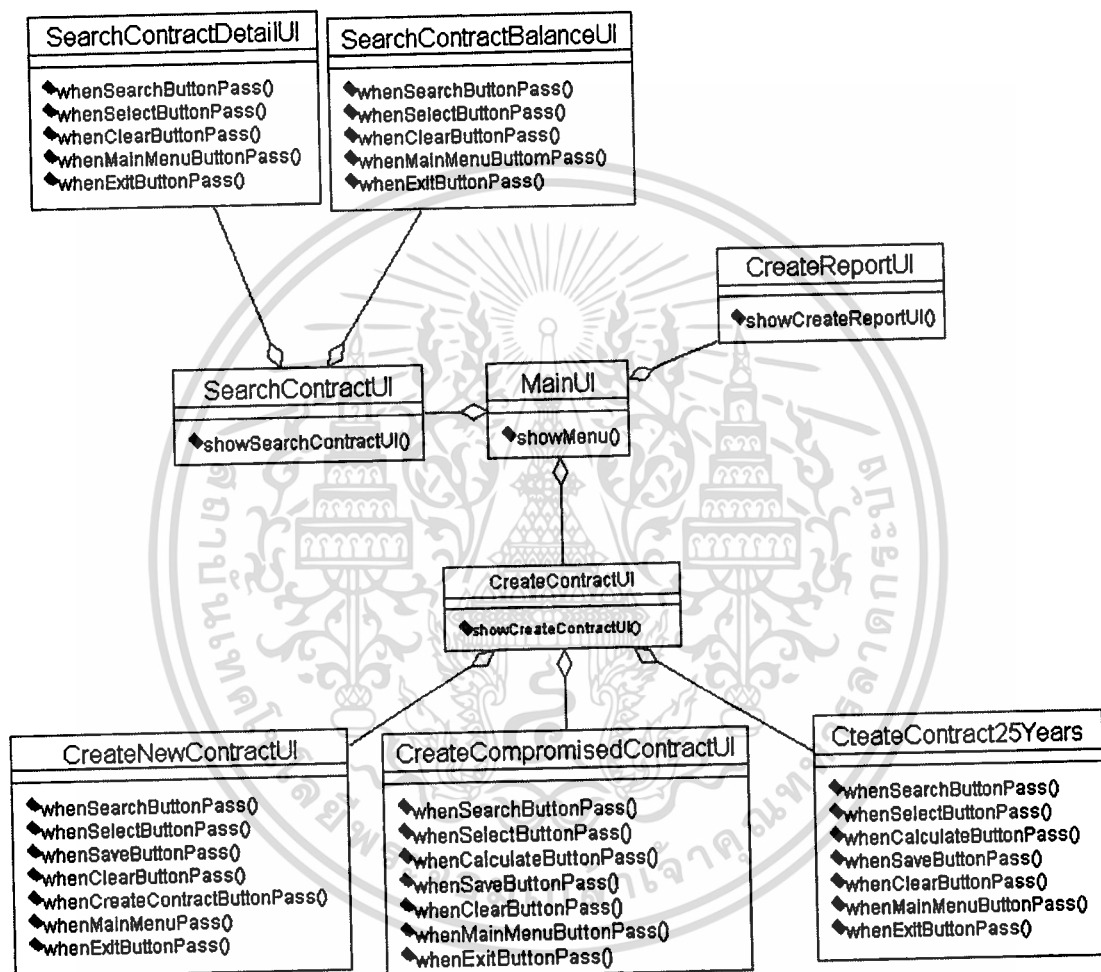
### 6.5.5.3 ความสัมพันธ์ ระหว่าง Class Diagram in Access Layer และ Application Logic



รูปที่ 6. 30 แสดงความสัมพันธ์ระหว่าง Class Diagram in Access Layer และ Application Logic

## 6.5.6 Class Diagram in View Layer

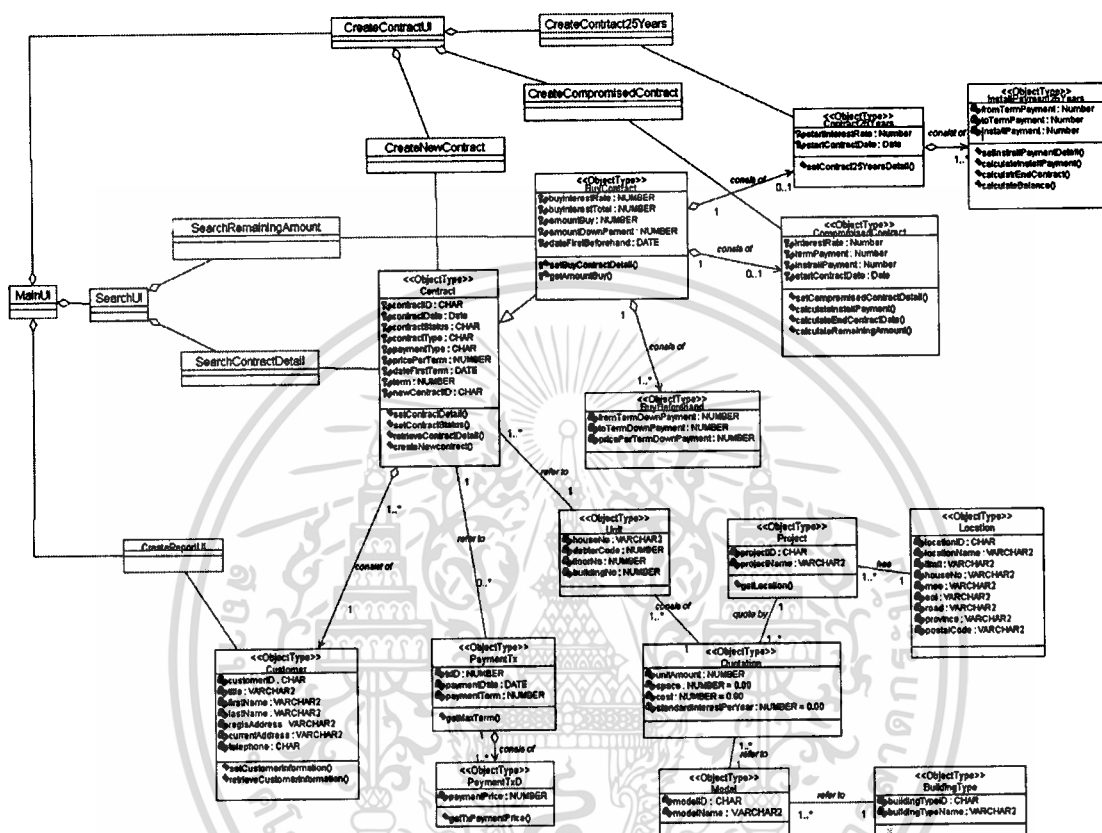
### 6.5.6.1 View Layer



รูปที่ 6. 31 แสดง Class Diagram ในส่วนของ View Layer



### 6.5.6.2 ความสัมพันธ์ ระหว่าง Class Diagram in Access Layer และ Application Logic



รูปที่ 6.32 แสดงความสัมพันธ์ระหว่าง Class Diagram in View Layer และ Application Logic

## บทที่ 7

### คู่มือผู้ใช้งาน

#### 7.1 วิธีการเข้าสู่โปรแกรม

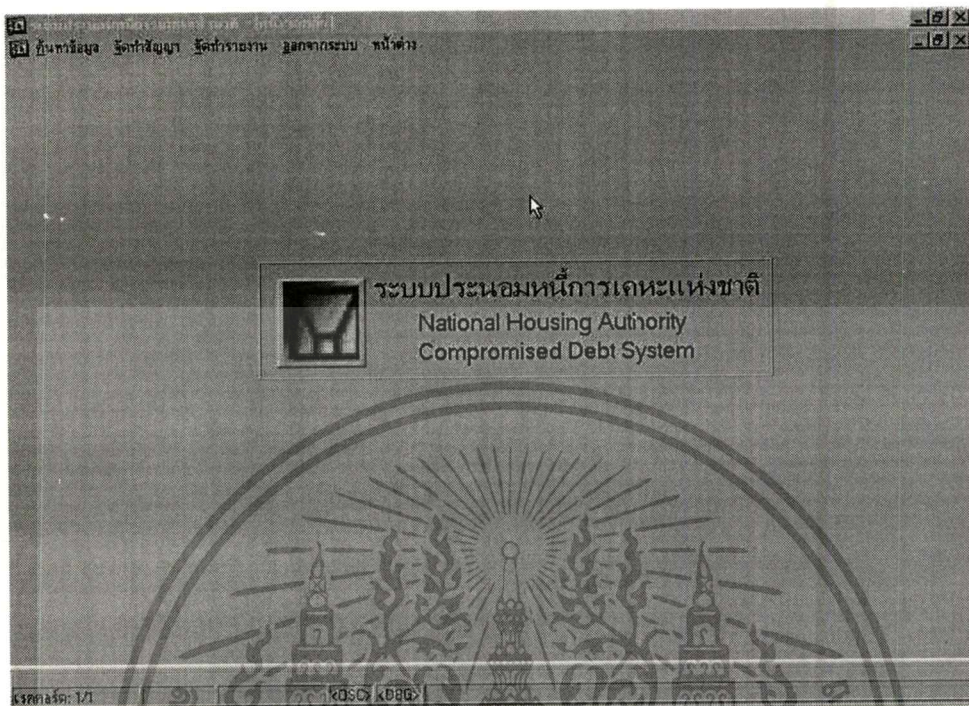
ผู้ใช้งานจะสามารถทำงานกับโปรแกรมนี้ได้ ต้องติดตั้งชุดโปรแกรมที่เรียกว่า Oracle Client ลงบนเครื่องที่ต้องการทำงานเสียก่อน โดยขั้นตอนแรกเมื่อเริ่มเข้าสู่โปรแกรม จะปรากฏหน้าจอ ดังรูปที่ 7.1 ซึ่งเป็นรูปภาพที่แสดงการ Login เข้าสู่ระบบ



รูปที่ 7.1 แสดงการ Login เข้าสู่ระบบ

เมื่อนำจอปรากฏดังรูปที่ 7.1 ผู้ใช้จะต้องกรอก ชื่อผู้ใช้งาน (User Name) รหัสผ่าน (Password) และฐานข้อมูล (Host String) ลงไป แล้ว กดปุ่มถูกต้อง เมื่อทั้งหมดถูกต้องก็จะสามารถเข้าสู่โปรแกรมได้และจะแสดงหน้าจอขึ้นมาดังรูปที่ 7.2 ซึ่งเป็นภาพที่แสดงหน้าจอหลักของโปรแกรม





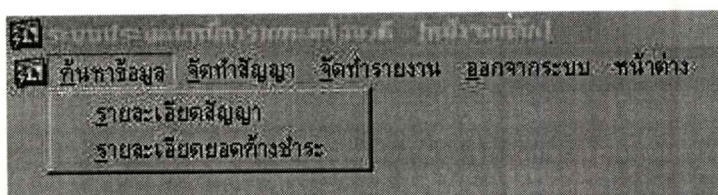
รูปที่ 7.2 แสดงหน้าจอเมนูหลักของระบบ

เมื่อปรากฏหน้าจอ เมนูหลักของระบบแล้ว ผู้ใช้จะสามารถใช้งานระบบได้ทันที โดยในหน้าจอเมนูหลักนี้ จะมีส่วนของการทำงาน แยกออกเป็น เมนูหลัก 5 เมนู ได้แก่

- 1 เมนูค้นหาข้อมูล เป็นเมนูที่ใช้สำหรับค้นหาข้อมูลต่างๆของสัญญา
2. เมนูจัดทำสัญญา เป็นเมนูที่ใช้สำหรับจัดทำสัญญาใหม่เมื่อมีการประนอมหนี้
3. เมนูจัดทำรายงาน เป็นเมนูที่ใช้สำหรับจัดทำรายงานต่างๆของระบบประนอมหนี้
4. เมนูออกจากระบบ เป็นเมนูที่ใช้สำหรับออกจากระบบ
5. เมนูหน้าต่าง เป็นเมนูที่ใช้เลือก และบอก สถานะของหน้าต่าง

## 7.2 เมนูค้นหาข้อมูล

จากหน้าจอหลัก (รูปที่ 7.2) เมื่อทำการเลือกเมนูค้นหาข้อมูล จะมีเมนูย่อย 2 เมนู คือ เมนูรายละเอียดสัญญา และ เมนูรายละเอียดค่างชำระ ดังแสดงในรูปที่ 7.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 7.3 แสดงเมนูย่อยในเมนูค้นหาข้อมูล

ไม่มีการผลิตหรือจำหน่ายในรูปแบบสิ่งพิมพ์แต่อย่างใด และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 7.2.1 ค้นหารายละเอียดสัญญา

เมื่อเลือกที่เมนูย่อยรายละเอียดสัญญาปรากฏหน้าจอดังรูปที่ 7.4

รูปที่ 7.4 แสดงหน้าจอค้นหารายละเอียดสัญญา

จากหน้าค้นหารายละเอียดสัญญานี้ จะประกอบไปด้วยส่วนการทำงาน 2 ส่วน คือ ส่วนบน คือส่วนของการค้นหา และส่วนล่างเป็นส่วนของการแสดงรายละเอียด (ซึ่งรูปแบบนี้จะใช้กับทุกๆหน้าจอของโปรแกรมนี้)

เริ่มต้นใช้งาน ผู้ใช้จะต้องทำการค้นหาก่อนว่า ต้องการดูรายละเอียดของสัญญา เลขที่เท่าไร หรือ สัญญาของใคร โดยในการค้นหาทำได้ 2 แบบ คือ ค้นหาจากเลขที่สัญญา หรือ ค้นหาจากชื่อ-สกุลลูกค้า

ขั้นตอนในการค้นหารายละเอียดสัญญา โดยใช้ “เลขที่สัญญา” ค้นหา

1. กรอกเลขที่สัญญา ดังในรูปที่ 7.5



รูปที่ 7.5 แสดงการกรอกเลขที่สัญญาเพื่อใช้ในการค้นหาข้อมูล

- เมื่อกรอก เลขที่สัญญา เรียบร้อยแล้ว ปุ่ม “ค้นหา” จะสามารถทำงานได้ (enable) ให้ทำการกดปุ่มค้นหา เพื่อทำการค้นหาเลขที่สัญญานั้น ซึ่งถ้าค้นหาพบจะปรากฏ รายละเอียดสัญญา ในส่วนล่างของหน้าจอ ดังรูปที่ 7.6

เอกสารนี้เป็นเอกสารที่แจ้งหนี้หรือใบแจ้งหนี้ที่ออกให้เพื่อใช้ในการชำระหนี้ เมื่อผู้ยื่นให้เห็น ใบแจ้งหนี้จะเปลี่ยนด้านการค้า

รูปที่ 7.6 แสดงหน้าจอแสดงรายละเอียดของสัญญา หลังจาก กดปุ่ม “ค้นหา” สารทุกครั้งที่มีการนำไปใช้



3. ถ้าต้องการ ค้นหารายละเอียดของสัญญาต่อไป ให้กดปุ่มล่าง เพื่อทำการล้างข้อมูลบนหน้าจอ ก่อนทำการค้นหาใหม่

ขั้นตอนในการค้นหารายละเอียดสัญญา โดยใช้ “ชื่อ-สกุล ลูกค้า” ค้นหา

1. กรอก ชื่อสกุล-ลูกค้า แล้วกด “Enter” ดังในรูป 7.7

The screenshot shows a software interface for searching contract details. The window title is "รายละเอียดสัญญา" (Contract Details). The date is set to "01.พ.ค. 2000". The interface is divided into several sections:

- ค้นหาจากเลขที่สัญญา (Search by Contract Number):** Includes a text box for "เลขที่สัญญา" (Contract Number) and a "ค้นหา" (Search) button.
- ค้นหาจากชื่อ-สกุลลูกค้า (Search by Name and Surname of Supplier):** Includes text boxes for "ชื่อ" (Name), "สกุล" (Surname), and "เลขที่สัญญา" (Contract Number), and a "ค้นหา" (Search) button.
- รายละเอียดสัญญา (Contract Details):** A table with columns for "เลขที่สัญญา" (Contract Number), "ชื่อโครงการ" (Project Name), "แบบสัญญา" (Contract Type), and "วันที่" (Date).
- รายละเอียดสัญญา (Contract Details):** A section with text boxes for "วันที่สัญญา" (Contract Date), "ประเภทสัญญา" (Contract Type), "สถานะสัญญา" (Contract Status), and "เลขที่สัญญาใหม่" (New Contract Number).
- รายละเอียดสัญญา (Contract Details):** A section with text boxes for "ประเภทสัญญา" (Contract Type), "วันที่สัญญา" (Contract Date), "จำนวนเงิน" (Amount), and "จำนวนเงินต่องวด" (Amount per installment).
- รายละเอียดสัญญา (Contract Details):** A section with text boxes for "วันที่สัญญา" (Contract Date), "ประเภทสัญญา" (Contract Type), "สถานะสัญญา" (Contract Status), and "เลขที่สัญญาใหม่" (New Contract Number).

The interface also includes a large circular watermark of the Thai coat of arms in the background.

รูปที่ 7.7 แสดงการกรอก ชื่อ-สกุล ลูกค้า เพื่อใช้ในการค้นหาข้อมูล

2. เมื่อกรอก ชื่อ-สกุล ลูกค้า และกด “Enter” เรียบร้อยแล้ว ปุ่ม “ค้นหา” จะสามารถทำงานได้ (enable) ให้ กดปุ่มค้นหา เพื่อทำการค้นหาเลขที่สัญญาทั้งหมด ของลูกค้ารายนั้น โดยจะแสดงรายละเอียดสัญญา ซึ่งได้แก่ เลขที่สัญญา ชื่อโครงการ ชื่อโมเดล บ้านเลขที่ ในส่วนบนด้านซ้ายของหน้าจอ ดังรูปที่ 7.8



ระบบประมวลผลการชำระหนี้  
ยื่นชำระหนี้ ชำระหนี้เงิน วัตถุประสงค์การชำระหนี้ ระบบ หน้าต่าง  
รายงาน: รายงานหนี้เงิน

### รายละเอียดสัญญา

วันที่ 01 พ.ค. 2000

☐ ค้นหาจากเลขที่สัญญา

เลขที่สัญญา:

☐ ค้นหาจากชื่อ-สกุลลูกค้า

ชื่อ:

สกุล:

| เลขที่สัญญา | ชื่อโครงการ     | แบบที่อยู่อาศัย | บ้านเลขที่ |
|-------------|-----------------|-----------------|------------|
| 00000004    | ศูนย์ 1/5       | D2              | 000000001  |
| 00000007    | ศูนย์ 1/5       | D1              | 000000001  |
| 00000010    | วังสกลดิษฐ์ใหม่ | H1              | 000000001  |
|             |                 |                 |            |
|             |                 |                 |            |

รายละเอียดสัญญา

วันที่ทำสัญญา:

ประเภทสัญญา:

สถานะสัญญา:

เลขที่สัญญาใหม่:

ประเภทการชำระเงิน:

วันที่ชำระเงินงวดแรก:

จำนวนงวด:

จำนวนเงินต่องวด:

ล้างหน้าจอ

หน้าจอหลัก

จบการทำงาน

FBM-41045: ไม่พบรายการ ID ไม่ถูกต้อง  
เลขที่รหัส: 4/4

รูปที่ 7.8 แสดงหน้าจอแสดงรายการสัญญาของลูกค้า หลังจากทำการค้นหา ชื่อ-สกุล ลูกค้าพบ

3. ทำการเลือกสัญญาที่ต้องการ(โดยคลิกเมาส์ที่เลขที่สัญญาที่ต้องการ) แล้วกดปุ่ม “เลือก” เพื่อทำการค้นหารายละเอียดของสัญญานั้น ซึ่งจะปรากฏหน้าจอแสดงรายละเอียดสัญญา หลังจากกดปุ่มเลือก ดังรูปที่ 7.9

ระบบประมวลผลการชำระหนี้  
ยื่นชำระหนี้ ชำระหนี้เงิน วัตถุประสงค์การชำระหนี้ ระบบ หน้าต่าง  
รายงาน: รายงานหนี้เงิน

### รายละเอียดสัญญา

วันที่ 01 พ.ค. 2000

☐ ค้นหาจากเลขที่สัญญา

เลขที่สัญญา:

☒ ค้นหาจากชื่อ-สกุลลูกค้า

ชื่อ:

สกุล:

| เลขที่สัญญา | ชื่อโครงการ     | แบบที่อยู่อาศัย | บ้านเลขที่ |
|-------------|-----------------|-----------------|------------|
| 00000004    | ศูนย์ 1/5       | D2              | 000000001  |
| 00000007    | ศูนย์ 1/5       | D1              | 000000001  |
| 00000010    | วังสกลดิษฐ์ใหม่ | H1              | 000000001  |
|             |                 |                 |            |
|             |                 |                 |            |

รายละเอียดสัญญา

วันที่ทำสัญญา: 25 ตุลาคม 1999

ประเภทสัญญา: สัญญาเช่าซื้อ

สถานะสัญญา: ชำระหนี้แล้ว

เลขที่สัญญาใหม่:

ประเภทการชำระเงิน: เงินสด

วันที่ชำระเงินงวดแรก: 01 พฤศจิกายน 1999

จำนวนงวด: 100 งวด

จำนวนเงินต่องวด: 45,000.00 บาท

ล้างหน้าจอ

หน้าจอหลัก

จบการทำงาน

เลขที่รหัส: 1/1

เอกสารนี้เป็นเอกสารที่แจ้งหนี้เงินให้กับผู้ชำระหนี้เพื่อใช้ในการหักภาษีเงินได้ เมื่อผู้ชำระหนี้ได้ชำระเงินเรียบร้อยแล้ว

รูปที่ 7.9 แสดงหน้าจอแสดงรายละเอียดของสัญญา หลังจาก กดปุ่ม “เลือก” เอกสารทุกครั้งที่มีการนำไปใช้





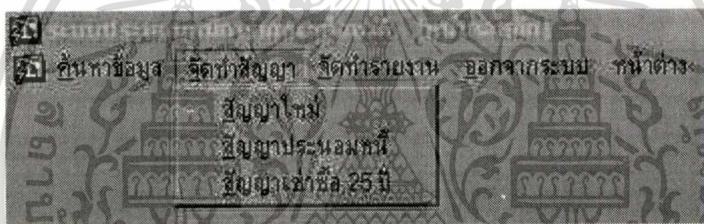


1. กรอกข้อมูลที่จะใช้ในการค้นหา ซึ่งได้แก่ เลขที่สัญญา หรือ ชื่อ-สกุล ลูกค้า
2. กดปุ่ม “ค้นหา” เพื่อค้นหารายละเอียดขอดีค้างชำระ
3. หรือ ถ้าใช้ ชื่อ-สกุล ลูกค้า ในการค้นหา จะต้องมีการเลือก เลขที่สัญญาที่ต้องการก่อน โดยการคลิกที่ เลขที่สัญญาที่ต้องการ แล้วจึง กดปุ่ม “เลือก” เพื่อค้นหารายละเอียดขอดีค้างชำระ

### 7.3 เมนูจัดทำสัญญา

จากหน้าจอหลัก (รูปที่ 7.2) เมื่อทำการเลือกเมนูจัดทำสัญญา จะมีเมนูย่อย 3 เมนู คือ เมนูจัดทำสัญญาใหม่ เมนูจัดทำสัญญาประนอมหนี้ และเมนูจัดทำสัญญาเช่าซื้อ 25 ปี ดังแสดงในรูปที่

7.12



รูปที่ 7.12 แสดงเมนูย่อยของเมนูจัดทำสัญญา

#### 7.3.1 จัดทำสัญญาใหม่

เมื่อเลือกที่เมนูย่อยจัดทำสัญญาใหม่ปรากฏหน้าจอ ดังรูปที่ 7.13

รูปที่ 7.13 แสดงหน้าจอจัดทำสัญญาใหม่

เริ่มต้นใช้งาน ผู้ใช้จะต้องทำการค้นหารายละเอียดสัญญาเก่าเสียก่อน โดยขั้นตอนหรือวิธีการค้นหานี้ จะเหมือนกับที่กล่าวไว้แล้ว ในส่วนของการค้นหาข้อมูลรายละเอียดสัญญา กล่าวคือ การค้นหาจะสามารถค้นหาได้ 2 แบบ คือ ค้นหาจากเลขที่สัญญา หรือ ค้นหาจาก ชื่อ-สกุลลูกค้า

ในส่วนของการจัดทำสัญญาใหม่นี้ เมื่อทำการค้นหาเลขที่สัญญา หรือชื่อ-สกุล ลูกค้า เรียบร้อยแล้ว จะปรากฏรายละเอียดดังรูปที่ 7.14 ในกรณีที่ค้นหาด้วย “เลขที่สัญญา” และดังรูปที่ 7.15 ในกรณีที่ค้นหาด้วย “ชื่อ-สกุล ลูกค้า”



โปรแกรมระบบการศึกษาระดับมัธยมศึกษา  
ชั้นประถมศึกษา จัดทำสัญญา จัดทำรายงาน ลอกจากระบบ หน้าต้น

จัดทำสัญญาใหม่ วันที่ 01 พ.ค. 2000

☒ ค้นหาจากเลขที่สัญญา ☐ ค้นหาจากชื่อ-สกุลลูกค้า

เลขที่สัญญา 00000001 ค้นหา

เลขที่สัญญา ชื่อโครงการ แผนที่อยู่อาศัย บ้านเลขที่

รายละเอียดลูกค้า

เลขประจำตัวบัตรประชาชน 1234567891111

ชื่อ ปิลาชัย

สกุล ภาณุสิทธิ์

รายละเอียดสัญญา

รหัสหน่วยอาศัย 0001H1000000001

วันที่ทำสัญญา 05/01/2000

ประเภทสัญญา สัญญาเช่าซื้อ

สถานะสัญญา ปะยอมหนี้

ประเภทการชำระเงิน ค่าผ่อนชำระ

วันที่จ่ายงวดแรก 05 พ.ค. 2000

จำนวนงวด 24

จำนวนเงินที่ต้องรับ 20,000.00

ดำเนินการจัด

ทำสัญญาใหม่

ยืนยัน

หน้าจอหลัก

จบการทำงาน

ทำสัญญาในระบบคอมพิวเตอร์

กรกฎสิทธิ์ 1/1

KDS< > DBG<

รูปที่ 7.14 แสดงหน้าจอจัดทำสัญญาใหม่ หลังจาก ค้นหาข้อมูลสัญญาเก่าเรียบร้อยแล้ว(กรณีค้นหาจาก “เลขที่สัญญา”)

โปรแกรมระบบการศึกษาระดับมัธยมศึกษา  
ชั้นประถมศึกษา จัดทำสัญญา จัดทำรายงาน ลอกจากระบบ หน้าต้น

จัดทำสัญญาใหม่ วันที่ 01 พ.ค. 2000

☒ ค้นหาจากเลขที่สัญญา ☐ ค้นหาจากชื่อ-สกุลลูกค้า

เลขที่สัญญา ค้นหา

เลขที่สัญญา ชื่อโครงการ แผนที่อยู่อาศัย บ้านเลขที่

00000004 ต.พชร 1/5 D2 00000001

00000007 ต.พชร 1/5 D1 00000001

00000010 รังสิตศิลาใหม่ H1 00000001

รายละเอียดลูกค้า

เลขประจำตัวบัตรประชาชน 2000000000000

ชื่อ ปัญญา

สกุล พรขจรกิจกุล

รายละเอียดสัญญา

รหัสหน่วยอาศัย 001005000000001

วันที่ทำสัญญา 25/10/1999

ประเภทสัญญา สัญญาเช่าซื้อ

สถานะสัญญา ยังคงใช้อยู่

ประเภทการชำระเงิน เงินสด

วันที่จ่ายงวดแรก 01 พ.ย. 1999

จำนวนงวด 100

จำนวนเงินที่ต้องรับ 45,000.00

ดำเนินการจัด

ทำสัญญาใหม่

ยืนยัน

หน้าจอหลัก

จบการทำงาน

ทำสัญญาในระบบคอมพิวเตอร์

กรกฎสิทธิ์ 1/1

KDS< > DBG<

รูปที่ 7.15 แสดงหน้าจอจัดทำสัญญาใหม่ หลังจาก ค้นหาข้อมูลสัญญาเก่าเรียบร้อยแล้ว(กรณีค้นหาจาก “ชื่อ-สกุลลูกค้า”)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หลังจากทำการค้นหา รายละเอียดสัญญาเก่า เรียบร้อยแล้ว ขั้นตอนต่อไป คือ การจัดทำสัญญาใหม่ โดยมีขั้นตอนการจัดทำสัญญาใหม่นี้

1. กดปุ่ม “จัดทำสัญญาใหม่”

เมื่อกดปุ่ม จัดทำสัญญาใหม่แล้ว จะปรากฏหน้าจอเตือนให้กรอกเลขที่สัญญาใหม่ ดังรูปที่ 7.16



รูปที่ 7.16 แสดงหน้าจอเตือน ให้กรอกเลขที่สัญญาใหม่ หลังจาก กดปุ่ม “จัดทำสัญญาใหม่”

2. เมื่อขึ้นหน้าจอเตือน ดังรูปที่ 7.16 นั้น ให้กดปุ่ม “OK” จากนั้นจะปรากฏหน้าจอ ดังรูปที่ 7.17

รูปที่ 7.17 แสดงหน้าจอจัดทำสัญญาใหม่ หลังจาก กดปุ่ม “OK” ในหน้าจอเตือนให้กรอกเลขที่

เอกสารสัญญาใหม่สารที่ส่งจนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้











เริ่มต้นใช้งาน ผู้ใช้จะต้องทำการค้นหารายละเอียดสัญญา คือ จำนวนเงินที่ต้องชำระทั้งหมด และดอกเบี้ยทั้งหมด เสียก่อน โดยขั้นตอนหรือวิธีการค้นหานี้ จะเหมือนกับที่กล่าวไว้แล้ว ในส่วนของการค้นหาข้อมูลรายละเอียดสัญญา กล่าวคือ การค้นหาจะสามารถค้นหาได้ 2 แบบ คือ ค้นหาจากเลขที่สัญญา หรือ ค้นหาจาก ชื่อ-สกุลลูกค้า

ในส่วนของการจัดทำสัญญาประนอมหนี้นี้ เมื่อทำการค้นหารายละเอียดสัญญา เรียบร้อยแล้ว จะปรากฏรายละเอียดดังรูปที่ 7.22 ในกรณีที่ค้นหาด้วย “เลขที่สัญญา” และดังรูปที่ 7.23 ในกรณีที่ค้นหาด้วย “ชื่อ-สกุล ลูกค้า”

[illegible]

รูปที่ 7. 22 แสดงหน้าจการจัดทำสัญญาประνομหนี หลังจาก ค้นหารายละเอียดสัญญาแล้ว  
(ค้นหาจาก “เลขที่สัญญา”)



**จัดทำสัญญาประนอมหนี้ 10 ปี**

วันที่ 01 ม.ค. 2001

☐ ค้นหาจากเลขที่สัญญา

☒ ค้นหาจากชื่อ-สกุลลูกค้า

ชื่อลูกค้า \_\_\_\_\_

สกุล \_\_\_\_\_

| เลขที่สัญญา | ชื่อโครงการ   | หน่วยย่อย | บ้านเลขที่ |
|-------------|---------------|-----------|------------|
| 00000004    | ธนบุรี 1/5    | D2        | 000000001  |
| 00000007    | ธนบุรี 1/5    | D1        | 000000001  |
| 00000010    | รังสิตฟ้าใหม่ | H1        | 000000001  |
|             |               |           |            |
|             |               |           |            |

รายละเอียดหนี้ค้ำประกัน (ใหม่)

จำนวนเงินที่ค้ำประกัน 4,275,000.00 บาท

อัตราดอกเบี้ยทั้งหมด .00 บาท

รายละเอียดการชำระเงินต่องวด

อัตราดอกเบี้ยประนอมหนี้ \_\_\_\_\_ เปอร์เซ็นต์

จำนวนงวดที่ต้องการ \_\_\_\_\_

จำนวนเงินต่องวด .00 บาท

คำนวณเงินต่องวด

ทำสัญญาฉบับที่ 25

เวอร์ชัน: 1/1

รูปที่ 7. 23 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ หลังจาก ค้นหารายละเอียดสัญญาแล้ว  
(ค้นหาจาก “ชื่อ-สกุล ลูกค้า”)

หลังจากทำการค้นหา รายละเอียดสัญญา เรียบร้อยแล้ว ขั้นตอนต่อไป คือ การจัดทำสัญญาประนอมหนี้ โดยมีขั้นตอนการจัดทำสัญญาใหม่นี้

1. กรอกค่าอัตราดอกเบี้ยประนอมหนี้ (ในขณะที่ทำการพัฒนาอยู่นี้ เงื่อนไขของอัตราดอกเบี้ยประนอมหนี้ คือ ใช้อัตราดอกเบี้ยคงที่ 7 %)
2. กรอก จำนวนงวดที่ต้องการ
3. เมื่อกรอกค่าต่างๆเรียบร้อยแล้ว
4. กดปุ่ม “คำนวณเงินต่องวด” จะปรากฏจำนวนเงินต่องวดที่คำนวณได้ ดังหน้าจอในรูปที่ 7.24



รูปที่ 7. 24 แสดงหน้าจอจัดทำสัญญาประนอมหนี้ หลังจากกดปุ่ม “คำนวณเงินต้องงวด”

5. จากนั้น ถ้าต้องการ บันทึกข้อมูล ให้กดปุ่ม “บันทึก” จะปรากฏ หน้าจอเตือน คล้ายกับ ที่กล่าวมาแล้ว ในส่วนของการบันทึก ใน หน้าจอจัดทำสัญญาใหม่
6. ในกรณีที่ ยังไม่ได้ กรอกข้อมูล อัตราดอกเบี้ยสัญญาประนอมหนี้ และ จำนวนงวดที่ต้องการ ให้ครบ จะมีหน้าจอเตือนให้กรอก อัตราดอกเบี้ยสัญญาประนอมหนี้ และ จำนวนงวดที่ต้องการ ดังแสดงในรูปที่ 7.25



รูปที่ 7. 25 แสดงหน้าจอเตือนให้กรอก อัตราดอกเบี้ยสัญญาประนอมหนี้ และ จำนวนงวด

ในกรณีที่ต้องการจะทำ สัญญาใหม่ต่อไป ให้ทำการล้างข้อมูลต่างๆ บนหน้าจอก่อน โดยการกดปุ่ม “ล้าง” แล้วจึงดำเนินการจัดทำสัญญาใหม่ต่อไปได้

### 7.3.3 จัดทำสัญญาเช่าซื้อ 25 ปี

เอกสารนี้เป็นเอกสารที่ส่งวันไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
เมื่อเลือกที่เมนูย่อยจัดทำสัญญาเช่าซื้อ 25 ปี จะปรากฏหน้าจอ ดังรูปที่ 7.26



รูปที่ 7. 26 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี

เริ่มต้นใช้งาน ผู้ใช้จะต้องทำการค้นหารายละเอียดสัญญา คือ จำนวนเงินที่ต้องชำระทั้งหมด เสียก่อน โดยขั้นตอนหรือวิธีการค้นหานี้ จะเหมือนกับที่กล่าวไว้แล้ว ในส่วนของการค้นหาข้อมูลรายละเอียดสัญญา กล่าวคือ การค้นหาจะสามารถค้นหาได้ 2 แบบ คือ ค้นหาจากเลขที่สัญญา หรือ ค้นหาจาก ชื่อ-สกุลลูกค้า

ในส่วนของการจัดทำสัญญาเช่าซื้อ 25 ปีนี้ เมื่อทำการค้นหารายละเอียดสัญญา เรียบร้อยแล้ว จะปรากฏรายละเอียดดังรูปที่ 7.27 ในกรณีที่ค้นหาด้วย “เลขที่สัญญา” และดังรูปที่ 7.28 ในกรณีที่ค้นหาด้วย “ชื่อ-สกุล ลูกค้า”



**กรมส่งเสริมการค้าระหว่างประเทศ**

ใบทราเชอริส อุตสาหกรรม อุตสาหกรรม ออกจากระบบ หน้าต่าง

กรมส่งเสริมการค้าระหว่างประเทศ

## จัดทำสัญญาเช่าซื้อปกติ 25 ปี

วันที่ 01 พ.ค. 2000

☒ ค้นหาจากเลขที่สัญญา

**ค้นหา**

00000001

☐ ค้นหาจากชื่อ-สกุลลูกค้า

ชื่อ \_\_\_\_\_

สกุล \_\_\_\_\_

| เลขที่สัญญา | ชื่อโครงการ | แบบก่อสร้าง | บ้านเลขที่ |
|-------------|-------------|-------------|------------|
|             |             |             |            |
|             |             |             |            |
|             |             |             |            |
|             |             |             |            |
|             |             |             |            |

**เลือก**

รายละเอียดการชำระเงินต่องวด

|                      |                   |                     |           |
|----------------------|-------------------|---------------------|-----------|
| จำนวนเงินที่ต้องชำระ | 7,000.00 บาท      | จำนวนงวดที่ต้องจ่าย | _____     |
| อัตราดอกเบี้ย        | _____ เปอร์เซ็นต์ | จำนวนเงินคงเหลือ    | _____ บาท |

**กำหนดวงเงินสินเชื่อรวม**

บันทึก

สร้างหน้างาน

พิมพ์ออก

จบการทำงาน

กรณีสืบค้น: 1/1      KDSO KDBSO

รูปที่ 7. 27 แสดงหน้าจอดีทำสัญญาเช่าซื้อ 25 ปี หลังจาก คำนวณรายละเอียดสัญญาแล้ว  
(ค้นหาจาก “เลขที่สัญญา”)

**จัดทำสัญญาเช่าซื้อปกติ 25 ปี**

วันที่ 01 พค. 2006

☐ ซื้อจากธนาคารที่สัญญา  
☒ ซื้อจากบริษัทผู้ให้เช่า

ชื่อ สัญญา [ ]  
 ชื่อ ผู้เช่า [ ]  
 ที่อยู่ [ ]

| เลขที่สัญญา | ชื่อโครงการ  | แบบที่อยู่อาศัย | บ้านเลขที่ |
|-------------|--------------|-----------------|------------|
| 00000004    | ตบจว 1/5     | D2              | 000000001  |
| 00000007    | ตบจว 1/5     | D1              | 000000001  |
| 00000010    | วังสราภิรมย์ | H1              | 000000001  |
|             |              |                 |            |
|             |              |                 |            |

จำนวนเงินที่ต้องชำระ 7,000.00 บาท  
 อัตราดอกเบี้ย [ ] เปอร์เซ็นต์

จำนวนงวดที่ต้องชำระ [ ]  
 จำนวนเงินต่องวด 00 บาท

จำนวนเงินที่ต้องจ่าย  
 จำนวนเงินที่ต้องรับ  
 จำนวนเงินที่ต้องจ่าย

**รูปที่ 7. 28 แสดงหน้าจอบันทึกทำสัญญาเช่าซื้อ 25 ปี หลังจาก คำนวณรายละเอียดสัญญาแล้ว**  
เอกสารประกอบแฟ้มการคำนวณเงินผ่อนชำระเพื่อกู้ยืมเงินเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ (บันทึกจาก “ซื้อ-สกุล ลูกค้า”)  
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากท่านไม่เห็นด้วยกับเงื่อนไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หลังจากทำการค้นหา รายละเอียดสัญญา เรียบร้อยแล้ว ขั้นตอนต่อไป คือ การจัดทำสัญญาเช่าซื้อ 25 ปี โดยมีขั้นตอนการจัดทำสัญญาใหม่ดังนี้

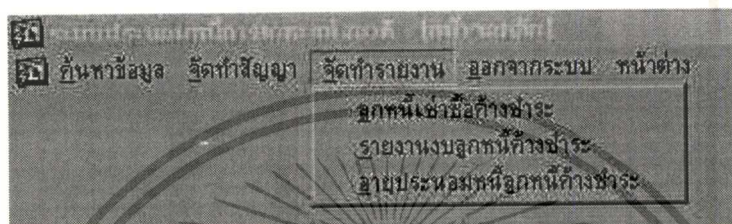
- 1 กรอกค่าอัตราดอกเบี้ยสัญญาเช่าซื้อ 25 ปี
- 2 กรอก จำนวนงวดที่ต้องการ
- 3 เมื่อกรอกค่าต่างๆเรียบร้อยแล้ว
- 4 กดปุ่ม “คำนวณเงินต้องงวด” จะปรากฏจำนวนเงินต้องงวดที่คำนวณได้ ดังหน้าจอในรูปที่ 7.29

รูปที่ 7. 29 แสดงหน้าจอจัดทำสัญญาเช่าซื้อ 25 ปี หลังจากกดปุ่ม “คำนวณเงินต้องงวด”

- 5 จากนั้น ถ้าต้องการ บันทึกข้อมูล ให้กดปุ่ม “บันทึก” จะปรากฏ หน้าจอเตือน คล้ายกับที่กล่าวมาแล้ว ในส่วนของการบันทึก ใน หน้าจอจัดทำสัญญาใหม่
- 6 ในกรณีที่ ยังไม่ได้ กรอกข้อมูล อัตราดอกเบี้ยสัญญาเช่าซื้อ 25 ปี และ จำนวนงวดที่ต้องการ ให้ครบ จะมีหน้าจอเตือนให้กรอก อัตราดอกเบี้ยสัญญาเช่าซื้อ 25 ปี และ จำนวนงวดที่ต้องการ เหมือนกับ หน้าจอเตือน ในการจัดทำสัญญาประนอมหนี้

## 7.4 เมนูจัดทำรายงาน

จากหน้าจอหลัก (รูปที่ 7.2) เมื่อทำการเลือกเมนูจัดทำรายงาน จะมีเมนูย่อย 3 เมนู คือ เมนูรายงานลูกหนี้เข้าซื้อค้างชำระ รายงานงบลูกหนี้ค้างชำระ รายงานอายุลูกหนี้ค้างชำระ ดังแสดงในรูปที่ 7.30



รูปที่ 7. 30 แสดงเมนูย่อยในเมนูจัดทำรายงาน

### 7.3.1 รายงานลูกหนี้เข้าซื้อค้างชำระ

เมื่อเลือกที่เมนูย่อยรายงานลูกหนี้เข้าซื้อค้างชำระแล้ว จะเป็นการเรียกรายงานลูกหนี้ค้างชำระมาแสดงผล โดยจะปรากฏหน้าจอ ดังรูปที่ 7.31 ซึ่งเป็นหน้าจอแสดงความก้าวหน้าของรายงาน และ รายงานในรูปที่ 7.32 เป็นหน้าจอรายงานลูกหนี้เข้าซื้อค้างชำระ



รูปที่ 7. 31 แสดงหน้าจอแสดงความก้าวหน้าของรายงาน





### รายงานลูกหนี้ค้างชำระค่าเช่าซื้อ

ชื่อโครงการ ขุมขบรรมกลั

โมเดล H1

| ชื่อ_สกุล         | เลขที่สัญญา | บ้านเลขที่ | จำนวนงวด<br>ที่ค้างชำระ<br>[งวด] | จำนวนเงิน<br>ที่ค้างชำระ<br>[บาท] |
|-------------------|-------------|------------|----------------------------------|-----------------------------------|
| นาย ทยา จันทนชาติ | 000000009   | 100000000  | 2                                | 80,000.00                         |

รูปที่ 7.32 แสดงตัวอย่างรายงานงบลูกหนี้ค้างชำระ

#### 7.3.2 รายงานงบลูกหนี้เข้าซื้อค้างชำระ

เมื่อเลือกเมนูย่อยรายงานงบลูกหนี้ค้างชำระ จะแสดงหน้าจอ ดังรูป 7.33 เพื่อให้กรอกค่าตัวแปร หรือ พารามิเตอร์ สำหรับออกรายงาน

รูปที่ 7.33 แสดงหน้าจอรับค่าตัวแปร รหัสโครงการ และ รหัสโมเดล เพื่อจัดทำรายงาน งบลูกหนี้ค้างชำระ

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับใช้ในการศึกษา และการวิจัยเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ในการตัดสินใจทางธุรกิจ หรือ การตัดสินใจอื่นใด และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณการกร : รหัสโครงการ

รหัสโครงการ: 0010

รหัสโมเดล: 001

รูปที่ 7.34 แสดงหน้าจอที่กรอกค่า รหัสโครงการ และ รหัสโมเดล เพื่อจัดทำรายงาน บงกชนี้ค้างชำระแล้ว

จากการส่งค่าพารามิเตอร์ไปนั้น จะได้ผลรายงาน ดังรูปที่ 7.35

report2 - Preview

การเคหะแห่งชาติ  
905 ถนนพหลโยธิน คลองจั่น เขตบางกะปิ  
กทม. 10240

หน้า 1  
วันที่ 01-พฤษภาคม 2008

รายงานงบผูกหนี้ค้างชำระค่าเช่าซื้อ

ชื่อโครงการ: ริงสิตฟ้าใหม่  
ชื่อโมเดล: H1

| บ้านเลขที่ | จำนวนงวดที่ค้างชำระ | จำนวนเงินที่ค้างชำระ | ค้าง 1-2 เดือน | ค้าง 3-6 เดือน | ค้าง 7-12 เดือน | ค้าง 13-24 เดือน | ค้าง 25 เดือนขึ้นไป |
|------------|---------------------|----------------------|----------------|----------------|-----------------|------------------|---------------------|
| 000000001  | 2                   | 90000                | 90000          | 0              | 0               | 0                | 0                   |
| 000000002  | 2                   | 100000               | 100000         | 0              | 0               | 0                | 0                   |
| 000000003  | 2                   | 60000                | 60000          | 0              | 0               | 0                | 0                   |
| 000000004  | 1                   | 20000                | 20000          | 0              | 0               | 0                | 0                   |
|            | 7                   | 270000               | 270000         | 0              | 0               | 0                | 0                   |

RealPlayer 7 Basic

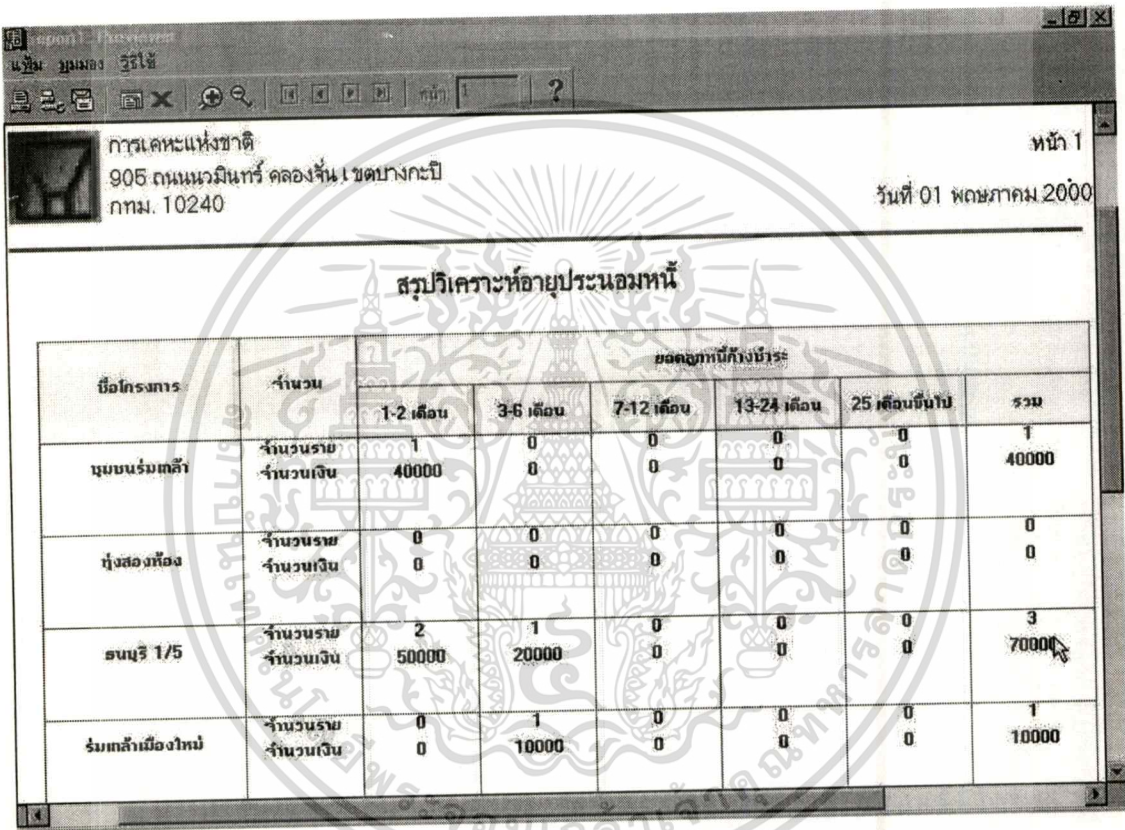
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 7.35 แสดงตัวอย่างรายงานงบผูกหนี้ค้างชำระ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 7.3.3 รายงานอายุประนอมหนี้ลูกหนี้ค้างชำระ

เมื่อเลือกที่เมนูย่อยรายงานอายุประนอมหนี้ลูกหนี้ค้างชำระแล้ว จะเป็นการเรียกรายงานอายุประนอมหนี้ลูกหนี้ค้างชำระมาแสดงผล ดังหน้าจอรายงานอายุประนอมหนี้ลูกหนี้ค้างชำระรูปที่ 7.36



| ชื่อโครงการ       | จำนวน     | ยอดลูกหนี้ค้างชำระ |           |            |             |                | รวม   |
|-------------------|-----------|--------------------|-----------|------------|-------------|----------------|-------|
|                   |           | 1-2 เดือน          | 3-6 เดือน | 7-12 เดือน | 13-24 เดือน | 25 เดือนขึ้นไป |       |
| ชุมชนร่มเกล้า     | จำนวนราย  | 1                  | 0         | 0          | 0           | 0              | 1     |
|                   | จำนวนเงิน | 40000              | 0         | 0          | 0           | 0              | 40000 |
| ทุ่งสองห้อง       | จำนวนราย  | 0                  | 0         | 0          | 0           | 0              | 0     |
|                   | จำนวนเงิน | 0                  | 0         | 0          | 0           | 0              | 0     |
| ธนบุรี 1/5        | จำนวนราย  | 2                  | 1         | 0          | 0           | 0              | 3     |
|                   | จำนวนเงิน | 50000              | 20000     | 0          | 0           | 0              | 70000 |
| ร่มเกล้าเมืองใหม่ | จำนวนราย  | 0                  | 1         | 0          | 0           | 0              | 1     |
|                   | จำนวนเงิน | 0                  | 10000     | 0          | 0           | 0              | 10000 |

รูปที่ 7. 36 แสดงตัวอย่างรายงานอายุประนอมหนี้ลูกหนี้ค้างชำระ

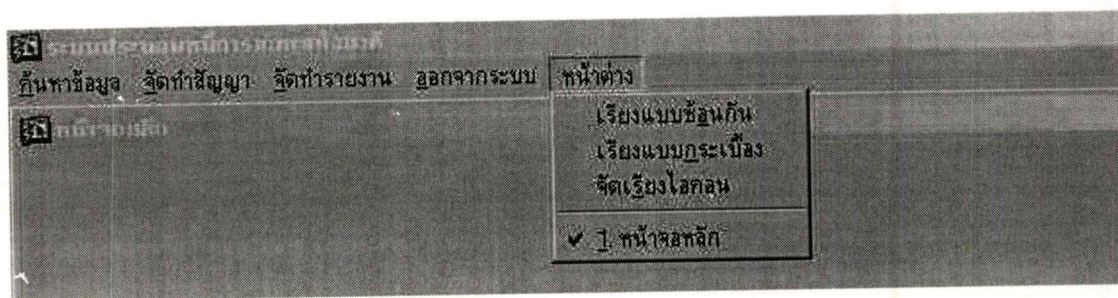
### 7.5 เมนูออกจากระบบ

เมนูจัดการระบบ โดยจะเป็นการสิ้นสุดการทำงานหรือออกจากโปรแกรมในส่วนของระบบ

### 7.6 เมนูหน้าต่าง

เมนูนี้เป็นเมนูที่ระบบการจัดการภายใน Oracle developer 6 แสดงขึ้นมาให้เอง ซึ่งประกอบด้วยเมนูย่อยที่ใช้ในการจัดการการทำงานของ Window ขณะ Run โปรแกรม ดังรูปที่ 7.37 โดยเมนูย่อย 3 ตัวบนใช้ในการจัดการ Window ส่วนเมนูย่อยตัวสุดท้ายจะแสดง Window ที่กำลัง





รูปที่ 7. 37 แสดงเมนูย่อยในเมนูหน้าต่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

### สรุป

จากการทำการพัฒนาระบบสารสนเทศในเชิงวัตถุ โดยใช้ระบบประนีประนอมหนึ่งของการเคหะแห่งชาติเป็นกรณีศึกษาครั้งนี้ เป็นการพัฒนาระบบตั้งแต่ขั้นตอนของการศึกษาความเป็นไปได้ในการพัฒนาระบบ การวิเคราะห์ระบบ การออกแบบระบบ และสุดท้ายการนำโครงสร้างของระบบที่ได้ออกแบบนั้นมาทำการพัฒนาเป็นระบบที่ใช้งานได้จริง ดังนั้นสามารถที่จะสรุปการทำโครงการได้ดังนี้

ในการพัฒนาระบบนั้นมีวัตถุประสงค์ที่จะ เลือกใช้วิธีการพัฒนาระบบในขั้นตอนของการวิเคราะห์และออกแบบเป็นไปในเชิงวัตถุ ส่วนการลงมือพัฒนาจริงนั้นปลายทางคือระบบฐานข้อมูลแบบเชิงสัมพันธ์ ซึ่งทำให้ประสบปัญหาในการนำวิธีการพัฒนาระบบทั้ง 2 วิธีการนี้มาผสมผสานกันโดยยังสามารถคงไว้ซึ่งหลักการของการพัฒนาทั้ง 2 วิธี

จากแนวความคิดในขั้นตอนวิเคราะห์และออกแบบระบบแบบวัตถุนั้นมีข้อดี คือ สามารถที่จะมองเห็น Object ต่างๆ ที่ออกแบบขึ้นมา โดย Object ต่างๆ นี้ จะมีความสมบูรณ์ในตัวเอง โดยเมื่อต้องการปรับเปลี่ยนก็สามารถที่จะทำได้โดยไม่มีผลกระทบต่อส่วนอื่นๆ ซึ่งเป็นคุณสมบัติอย่างหนึ่งของ Object-Oriented และยังสามารถนำสิ่งที่พัฒนาไว้แล้วมาใช้ซ้ำได้

แต่การที่แปลงจากโครงสร้างที่ถูกออกแบบมาในเชิงวัตถุให้มีโครงสร้างที่เหมาะสมกับฐานข้อมูลเชิงสัมพันธ์และยังคงสามารถที่จะยังคงคุณสมบัติของ Object-Oriented ได้นั้น ก็ไม่ใช่เรื่องง่ายนัก เนื่องจากเครื่องมือแต่ละประเภทที่ถูกพัฒนาขึ้นมาก็มีขีดความสามารถและความเหมาะสมในการใช้งานเฉพาะตัว

สำหรับ Oracle 8i นั้น ในการทำงานจริงๆ แล้วยังไม่เป็นระบบฐานข้อมูลเชิงวัตถุที่สมบูรณ์ เพียงแต่มีฟังก์ชันการทำงานรองรับการทำงานในเชิงวัตถุ นั่นก็คือ Store Procedure

อีกประการหนึ่งเนื่องจากการพัฒนาเชิงวัตถุมีแนวความคิด Encapsulation data นั่นก็คือจะอนุญาตเฉพาะ Class ของคนเท่านั้นที่จะสามารถเข้าถึงข้อมูลของตนเองได้ ดังนั้น จึงไม่สามารถใช้ประสิทธิภาพและความสามารถของภาษา SQL ได้อย่างเต็มที่ตัวอย่างเช่นเรื่องของการ Join ไม่สามารถที่จะนำมาใช้ได้ ด้วยเหตุที่การ Join เป็นวิธีการเข้าถึงข้อมูลโดยตรงไปยัง Table หรือ Sub-Query ก็ไม่สามารถใช้ได้เช่นกัน

## 8.1 ปัญหาต่างๆ ที่เกิดในการพัฒนาระบบ

เนื่องจากการพัฒนาระบบเชิงวัตถุนี้เป็นการพัฒนาแบบ Iterative คือเป็นลักษณะของการทำซ้ำ รวมถึงการแก้ไขปรับปรุงให้ดีขึ้น ทำให้ต้องมีการแก้ไขรายละเอียดต่างๆ อยู่มากและตลอดเวลา ทำให้มีความต้องการสำหรับการนำเครื่องมือเข้ามาช่วยในการพัฒนา

ในส่วนของการสร้างระบบครั้งนี้มีปัญหาใหญ่ๆ ที่พบอย่างชัดเจนคือเรื่องของการพัฒนาบนระบบฐานข้อมูลแบบ Relational หากยังคงรักษาทฤษฎีของ Object-Orient แล้วจะใช้ประสิทธิภาพของภาษา SQL ได้ไม่เต็มประสิทธิภาพ เช่นในเรื่องของการ Join จะไม่ได้ใช้เลยเพราะการ Join เป็นวิธีการเข้าถึงข้อมูลโดยตรงของอีก Table หนึ่งทันที หรือแม้แต่ Sub-Query ก็ไม่ได้ใช้เช่นกัน และยังรวมถึงไม่ได้ใช้ประสิทธิภาพ DBMS ของ Oracle ในส่วนของ Query Optimizer เลย

## 8.2 เปรียบเทียบกับการพัฒนาระบบในแนวทางเดิม

การพัฒนาระบบเชิงวัตถุมีการแสดงระบบที่พัฒนาในระดับที่สูงกว่าการพัฒนาระบบในแนวทางเดิม เนื่องจากจะแสดงอยู่ในระดับ ออบเจกต์ เพราะภายใน ออบเจกต์ จะมีทั้งข้อมูลและหน้าที่ของตัวเองที่ทำงานร่วมกัน

ในแต่ละขั้นตอนของการพัฒนาระบบเชิงวัตถุ จะมีความกลมกลืนกันมาก กล่าวคือการเปลี่ยนจากขั้นหนึ่งไปยังอีกขั้นหนึ่งนั้น เป็นไปอย่างเรียบง่าย เพราะมีการใช้ภาษาที่เหมือนกันตลอดการพัฒนา และภาษาในที่นี้ก็คือแบบจำลองนั่นเอง ซึ่งการใช้แบบจำลองเดียวกันตลอดในการพัฒนาจะมีผลทำให้เวลาในการพัฒนานั้นมีความรวดเร็วยิ่งขึ้น ขณะที่การพัฒนาในแนวทางเดิมนั้น ในแต่ละขั้นของการพัฒนาจะมีรูปแบบและวิธีที่ต่างกันไป

ในการพัฒนาเชิงวัตถุนี้จะมีการสนับสนุนการนำของที่มีอยู่กลับมาใช้ใหม่ (Reusability) เพราะจะนำ ออบเจกต์ แทนด้วยโลกของความเป็นจริง และแต่ละ ออบเจกต์ นั้นสามารถอยู่ได้ด้วยตัวเอง เมื่อนำ ออบเจกต์ แทนด้วยโลกของความเป็นจริง ก็แสดงว่าเป็นการไม่ผูกมัดว่าจะต้องอยู่ในเพียงระบบนี้เท่านั้น กล่าวคือ การออกแบบนั้นจะพิจารณาถึงการให้ใช้งานได้ทั่วไป

## 8.3 แนวทางในการพัฒนาระบบ

การพัฒนาระบบเชิงวัตถุนี้เป็นแนวทางล่าสุดในการพัฒนาระบบ ซึ่งยังไม่มีข้อพิสูจน์ที่แน่ชัดว่าพัฒนาระบบเชิงวัตถุจะเป็นแนวทางสำหรับการพัฒนาระบบที่ดีที่สุด แต่ก็ยังเป็นแนวทางหนึ่งของการพัฒนาระบบที่สามารถรองรับความซับซ้อนที่เกิดขึ้นได้ และมีการใช้แบบจำลองที่เหมือนกันตลอดทั้งการพัฒนานั้นคือ UML (Unified Modeling Language) และยังมีการสนับสนุนการ Reuse อย่างเต็มที่ ซึ่งทำให้มีผลกับระยะเวลาของการพัฒนานั้นเร็วขึ้น



สำหรับ UML เองนั้นมีส่วนช่วยในการพัฒนาเป็นอย่างมาก เนื่องจากมีแบบจำลองที่สามารถสนับสนุนสภาวะแวดล้อมในการพัฒนาระบบเชิงวัตถุทุกๆ ด้าน ไม่ว่าจะเป็นด้านการรวบรวมความต้องการ ด้านการวิเคราะห์ ด้านการออกแบบ หรือแม้แต่ด้านการสร้างระบบ และยิ่งในปัจจุบันมีเครื่องมือที่สนับสนุนการพัฒนาระบบเชิงวัตถุเพิ่มขึ้นเรื่อยๆ และมีผลทำให้การพัฒนาระบบเชิงวัตถุมีแนวโน้มเป็นการพัฒนาระบบในอนาคตต่อไปได้



## บรรณานุกรม

- Bahrami, Ali. 1998. Object Oriented Systems Development using Unified Modeling Language. Singapore: McGraw-Hill.
- Coad, P. and Yourdon, E. 1991. Object-Oriented Design. New Jersey: Yourdon Press.
- Dittman, Kevin C. 1997. Object Modeling Systems Analysis and Design. Irwin McGraw-Hill.
- Eliksson, Hans-Erik and Magnus Penker. 1998. UML Toolkit. John Wiley & Sons.
- Montgomery, Stephen L. 1998. Building Object-Oriented Software. McGraw-Hill.
- Quatrani, Terry. 1998. Visual Modeling with Rational Rose and UML. Wesley Longman.
- Wirfs-Brock, Rebecca., Brian Wilerson And Lauren Wiener. 1992. Designing Object-Oriented Software. New Jersey: Prentice-Hall.

## ภาคผนวก

### ก. ตัวอย่างรายงานของระบบ

ตัวอย่างรายงานของระบบประนอมหนี้ ซึ่งจะประกอบด้วย สรุปรายการลูกหนี้ค้างชำระค่าเช่าซื้อ รายงานงบลูกหนี้ค้างชำระค่าเช่าซื้อ และ รายงานวิเคราะห์อายุประนอมหนี้ โดยแสดงรายละเอียดต่างๆ ดังนี้







การเคหะแห่งชาติ

905 ถนนนวมินทร์ คลองจั่น เขตบางกะปิ กทม. 10240

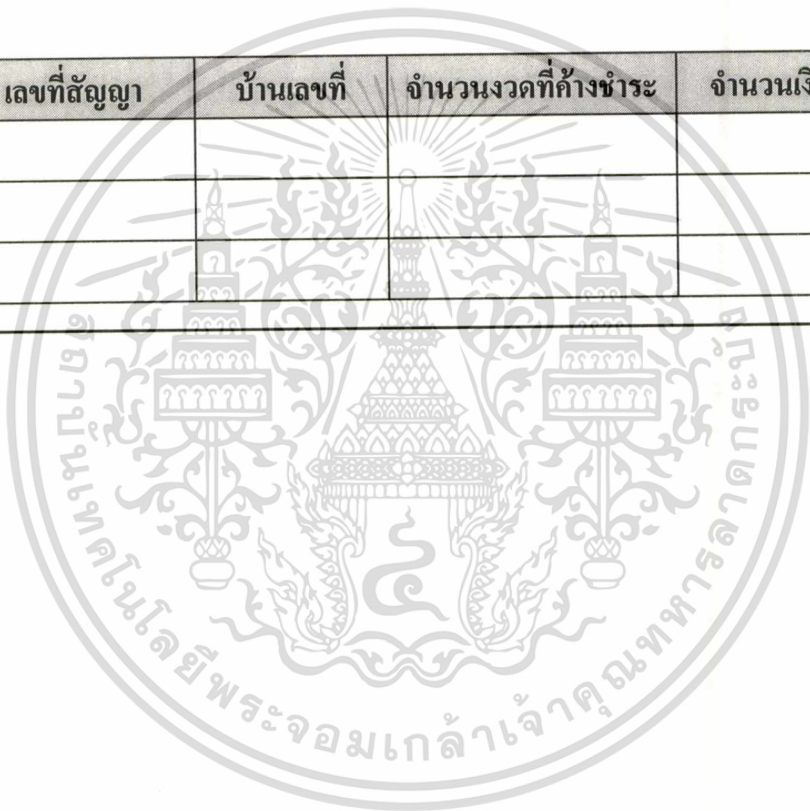
วันที่ออกรายงาน 10/11/2542

รายงานลูกหนี้ค้างชำระค่าเช่าซื้อ

ชื่อโครงการ

โมเดล

| ชื่อ-สกุล | เลขที่สัญญา | บ้านเลขที่ | จำนวนงวดที่ค้างชำระ | จำนวนเงินที่ค้างชำระ |
|-----------|-------------|------------|---------------------|----------------------|
|           |             |            |                     |                      |
|           |             |            |                     |                      |
|           |             |            |                     |                      |





การเคหะแห่งชาติ

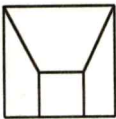
905 ถนนนวมินทร์ คลองจั่น เขตบางกะปิ กทม. 10240

วันที่ออกรายงาน 10/11/2542

รายงานงบผูกพันค้ำชำระค่าเช่าซื้อ

ชื่อโครงการ  
โมเดล

| บ้านเลขที่ | จำนวนงวด<br>ที่ค้ำชำระ | จำนวนเงิน<br>ที่ค้ำชำระ | ค้ำ<br>1-2<br>เดือน | ค้ำ<br>3-6<br>เดือน | ค้ำ<br>7-12<br>เดือน | ค้ำ<br>13-24<br>เดือน | ค้ำ<br>25 เดือน<br>ขึ้นไป |
|------------|------------------------|-------------------------|---------------------|---------------------|----------------------|-----------------------|---------------------------|
|            |                        |                         |                     |                     |                      |                       |                           |
|            |                        |                         |                     |                     |                      |                       |                           |
|            |                        |                         |                     |                     |                      |                       |                           |
|            |                        |                         |                     |                     |                      |                       |                           |



การเคหะแห่งชาติ  
905 ถนนนวมินทร์ คลองจั่น เขตบางกะปิ กทม. 10240

วันที่ออกรายงาน 10/11/2542

รายงานสรุปวิเคราะห์อายุประนอมหนี้

ชื่อโครงการ  
โมเดล

| ชื่อโครงการ | จำนวน     | ยอดลูกหนี้ค้างชำระ |                   |                    |                     |                        |     |
|-------------|-----------|--------------------|-------------------|--------------------|---------------------|------------------------|-----|
|             |           | ค้าง<br>1-2 เดือน  | ค้าง<br>3-6 เดือน | ค้าง<br>7-12 เดือน | ค้าง<br>13-24 เดือน | ค้าง 25<br>เดือนขึ้นไป | รวม |
|             | จำนวนรวม  |                    |                   |                    |                     |                        |     |
|             | จำนวนเงิน |                    |                   |                    |                     |                        |     |
|             | จำนวนรวม  |                    |                   |                    |                     |                        |     |
|             | จำนวนเงิน |                    |                   |                    |                     |                        |     |
|             | จำนวนรวม  |                    |                   |                    |                     |                        |     |
|             | จำนวนเงิน |                    |                   |                    |                     |                        |     |



## ข. ตัวอย่าง Store Procedure

Store Procedure Balance\_Search\_Pkg

Program Specification

PACKAGE balance\_search\_pkg IS

type contrecord is record (

cont\_id contract.contractid%type,  
cont\_date contract.contractdate%type,  
cont\_status contract.contractstatus%type,  
ncont\_id contract.contractidnew%type,  
house\_no contract.houseno%type,  
cust\_id contract.customerid%type,  
project\_name project.projectname%type,  
model\_name model.modelname%type,  
fdaterm contract.datefirstterm%type,  
term contract.term%type,  
price\_term contract.priceperterm%type);

type conttable is table of contrecord

index by binary\_integer;

type custrecord is record (

cust\_id customerinformation.customerid%type,  
fname customerinformation.firstname%type,  
lname customerinformation.lastname%type);

type custtable is table of custrecord

index by binary\_integer;

type buycontrecord is record (

cont\_id buycontract.contractid%type,  
amountbuy buycontract.amountbuy%type);

type buyconttable is table of buycontrecord

index by binary\_integer;

type txrecord is record (

cont\_id            paymenttransaction.contractid%type,

pay\_date        paymenttransaction.paymentdate%type);

type txtable is table of txrecord

index by binary\_integer;

Function verifyContractID(cont\_id contract.contractid%type)

return boolean;

Function getMaxTerm(cont\_id contract.contractid%type)

return paymenttransaction.paymentterm%type;

Function getPaymentDate(cont\_id contract.contractid%type)

return paymenttransaction.paymentdate%type;

Function getPaymentAmount(cont\_id contract.contractid%type)

return buycontract.amountbuy%type;

Function getAmountBuy(cont\_id buycontract.contractid%type)

return buycontract.amountbuy%type;

Function calCreditTerm(cont\_id contract.contractid%type)

return contract.term%type;

Function calCreditAmount(cont\_id contract.contractid%type)

return buycontract.amountbuy%type;

Function calRemainAmount(cont\_id contract.contractid%type)

```
return buycontract.amountbuy%type;
```

```
Function verifyCustomerName(fname customerinformation.firstname%type,  
                             lname customerinformation.lastname%type)  
  
return boolean;
```

```
Procedure getCustomerID(fname customerinformation.firstname%type,  
                        lname customerinformation.lastname%type,  
                        recordset in out balance_search_pkg.custtable);
```

```
Procedure getProjectDetail(cust_id contract.customerid%type,  
                           recordset in out balance_search_pkg.conttable);
```

```
Procedure getContractDetail(cont_id contract.contractid%type,  
                             recordset in out balance_search_pkg.conttable);
```

```
END;
```

Program Body

```
PACKAGE BODY balance_search_pkg IS
```

```
i integer := 1;
```

```
amount_buy integer(9,2);
```

```
max_term integer(4);
```

```
price_term integer(9,2);
```

```
remain_term integer(4);
```

```
credit_amount integer(9,2);
```

```
remain_amount integer(9,2);
```

```
--search contract detail by contractid
```

```
--verify contractid
```

```
Function verifyContractID(cont_id contract.contractid%type) return boolean is
```



```

temp contract.contractid%type;

begin

    select  contractid into temp
    from    contract
    where   contractid = cont_id;

    if temp is not null then
        return true;
    else
        return false;
    end if;
Exception
    When no_data_found then
        return false;
end;

```

--get contract detail(priceperterm,datefirstterm,term) by contractid

```

Procedure getContractDetail(cont_id contract.contractid%type,
    recordset in out balance_search_pkg.conttable)is
    price_term contract.priceperterm%type;
begin
    select  contract.contractdate,
            contract.datefirstterm,
            contract.term,
            contract.priceperterm
    into    recordset(1).cont_date,
            recordset(1).fdateterm,
            recordset(1).term,
            recordset(1).price_term
    from    contract
    where   contract.contractid = cont_id;

```

```

price_term := recordset(1).price_term;
end;

```

-- get max term by contractid

Function getMaxTerm(cont\_id contract.contractid%type)

```

    return paymenttransaction.paymentterm%type is
maxterm paymenttransaction.paymentterm%type;
begin
    select  max(paymentterm)
    into    maxterm
    from    paymenttransaction
    where   contractid = cont_id;
    return maxterm;
    max_term := maxterm;
end;

```

--get paymentdate by contractid

Function getPaymentDate(cont\_id contract.contractid%type)

```

    return paymenttransaction.paymentdate%type is
pay_date date;
begin
    select  max(paymentdate)
    into    pay_date
    from    paymenttransaction
    where   contractid = cont_id;
    return pay_date;
end;

```

--get payment amount by contractid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function getPaymentAmount(cont\_id contract.contractid%type)

return buycontract.amountbuy%type is

maxterm paymenttransaction.paymentterm%type;

price\_term contract.priceperterm%type;

pay\_amount integer(9,2);

begin

select max(paymentterm)

into maxterm

from paymenttransaction

where contractid = cont\_id;

select contract.priceperterm

into price\_term

from contract

where contract.contractid = cont\_id;

pay\_amount := maxterm\*price\_term;

return pay\_amount;

end;

--get amount buy by contractid

Function getAmountBuy(cont\_id buycontract.contractid%type)

return buycontract.amountbuy%type is

amount\_buy buycontract.amountbuy%type;

begin

select amountbuy

into amount\_buy

from buycontract

where contractid = cont\_id;

return amount\_buy;

end;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



--calcurate NO. credit term by contractid

Function calCreditTerm(cont\_id contract.contractid%type)

return contract.term%type is

fdate\_term contract.datefirstterm%type;

tx\_date paymenttransaction.paymentdate%type;

maxterm paymenttransaction.paymentterm%type;

rterm integer;

begin

--get paymentdate

```
select  max(paymentdate)
into    tx_date
from    paymenttransaction
where   contractid = cont_id;
```

--cal NO. month between

```
rterm := months_between(sysdate,tx_date);
return rterm;
remain_term := rterm;
```

end;

--calcurate CreditAmount by contractid

Function calCreditAmount(cont\_id contract.contractid%type)

return buycontract.amountbuy%type is

credit\_amount integer(9,2);

price\_term contract.priceperterm%type;

rterm integer(4);

tx\_date date;

begin

```
select  max(paymentdate)
into    tx_date
from    paymenttransaction
```

```

where contractid = cont_id;

select contract.priceperterm

into price_term

from contract

where contract.contractid = cont_id;

rterm := months_between(sysdate,tx_date);

credit_amount := price_term*rterm;

return credit_amount;

end;

```

--calcurate remain amount

Function calRemainAmount(cont\_id contract.contractid%type)

return buycontract.amountbuy%type is

credit\_amount integer(9,2);

price\_term contract.priceperterm%type;

maxterm paymenttransaction.paymentterm%type;

amount\_buy buycontract.amountbuy%type;

remain\_amount integer(9,2);

begin

select contract.priceperterm

into price\_term

from contract

where contract.contractid = cont\_id;

select max(paymentterm)

into maxterm

from paymenttransaction

where contractid = cont\_id;

select amountbuy

into amount\_buy

from buycontract

```

where    contractid = cont_id;

credit_amount := price_term*maxterm;

remain_amount := amount_buy-credit_amount;

return remain_amount;

end;

```

--search contract detail by customer name

--verify firstname and lastname

```

Function verifyCustomerName(fname customerinformation.firstname%type,
                             lname customerinformation.lastname%type)
return boolean is
temp1 customerinformation.firstname%type;
temp2 customerinformation.lastname%type;
begin
select  firstname , lastname
into    temp1 , temp2
from    customerinformation
where   firstname = fname and
        lastname = lname;
if      temp1 is not null and
        temp2 is not null then
        return true;
else
        return false;
end if;
Exception
When no_data_found then
return false;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



--search customerid by customer name

```

Procedure getCustomerID(fname customerinformation.firstname%type,
                        lname customerinformation.lastname%type,
                        recordset in out balance_search_pkg.custtable)is
begin
    select  customerid
    into    recordset(1).cust_id
    from    customerinformation
    where   firstname = fname and
           lastname = lname;
end;
```

--search project detail by customerid

```

Procedure getProjectDetail(cust_id contract.customerid%type,
                           recordset in out balance_search_pkg.conttable) is
cursor c1(ccus_id contract.customerid%type) is
select  contract.contractid,
        project.projectname,
        model.modelname,
        contract.houseno
from    contract,
        project,
        model
where   customerid = cust_id and
        contract.projectid =project.projectid and
        contract.modelid = model.modelid;
i binary_integer := 1;
begin
```

```

    open c1(cust_id);
```

```

    loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fetch c1
into recordset(i).cont_id,
recordset(i).project_name,
recordset(i).model_name,
recordset(i).house_no;
exit when c1%notfound;
i := i+1;
end loop;
close c1;
end;
END;
```



ประวัติผู้เขียน

ชื่อ นางสาว อุษณีย์ เปี่ยมประถม  
เกิด 15 สิงหาคม 2517

ประวัติการศึกษา

มัธยม                      ราชินีบน  
อุดมศึกษา                มหาวิทยาลัยมหิดล

ประวัติการทำงาน

เจ้าหน้าที่ห้องแลป บริษัทสเปเชียลแลป จำกัด

