

การพัฒนาระบบบัญชีลูกหนี้เช่า การเคหะแห่งชาติ

Developing a rental account system for National Housing Authority of Thailand

โดย

นาย ภูสิทธิ์ รัตนปิยะสุนทร

รหัส 41067004



H001624

อาจารย์ที่ปรึกษา

ผศ. บรรจง ปิยะธำรง

วัน เดือน ปี..... 22 ส.ค. 2549
เลขทะเบียน..... 01624
เลขเรียกหนังสือ.....
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2542
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาระบบบัญชีลูกหนี้เช่าการเคหะแห่งชาติ
นักศึกษา	นาย ภูสิทธิ์ รัตนปิยะสุนทร
อาจารย์ที่ปรึกษา	ผศ. บรรจง ปิยะธำรง
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2542

บทคัดย่อ

โครงการนี้ เป็นการพัฒนาระบบสารสนเทศ โดยใช้หลักวิธีในการพัฒนาระบบแบบเชิงวัตถุ ตั้งแต่การวิเคราะห์ ไปจนถึงการออกแบบ และนำสิ่งที่ออกแบบได้มาพัฒนาเป็นระบบที่ใช้งานได้จริง โดยใช้เครื่องมือและระบบฐานข้อมูลเชิงสัมพันธ์ การออกวิเคราะห์และออกแบบเชิงวัตถุ นั้นจะกระทำโดยอาศัยระเบียบวิธีของยูเอ็มแอล ทั้งภาษา ระเบียบวิธี และเครื่องมือช่วยการวิเคราะห์ระบบ ในโครงการนี้จะพัฒนาระบบงาน โดยอาศัยกรณีศึกษาของระบบบัญชีลูกหนี้เช่า การเคหะแห่งชาติ

Title Developing a rental account system for National Housing Authority of Thailand

Student Mr. Phoosith Ratpiyasoonorn

Advisor Assoc.Prof Banjong Piyathamrong

Level of Study Master of Science in Information Technology

Major Information Science

Academic Year 1999

ABSTRACT

This project's main objective is to developing an information system by using an object-oriented development methodology, which covered from analysis to designing. The result of design then developing into a working system by using a relational database system and relational database tools. The object-oriented analysis and design is based on UML in language, methodology and tools. This project is using a case study from a rental account system for rental customer of National Housing Authority of Thailand.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลงได้ด้วยดี ก็เพราะความกรุณาจาก ผศ.บรรจง ปิยะธำรง อาจารย์ที่ปรึกษา ที่กรุณาให้คำแนะนำ และชี้แนวทางแก่ข้าพเจ้าตลอดมา ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย และยังขอขอบพระคุณคณาจารย์ท่านอื่น ๆ ที่ได้ประสิทธิประสาทวิชาความรู้แก่ข้าพเจ้า จนได้ใช้เป็นศาสตร์ประกอบการทำโครงการนี้ได้ลุล่วงดังประสงค์

ที่สำคัญที่สุด ข้าพเจ้าต้องขอกราบขอบพระคุณคุณพ่อคุณแม่ของข้าพเจ้า ที่ได้อุปการะเลี้ยงดูข้าพเจ้า และเป็นกำลังใจให้กับข้าพเจ้าเสมอมา ทั้งในช่วงเวลาที่ลำบากที่สุด และช่วงเวลาที่ดีที่สุดในท้ายที่สุดนี้ก็ต้องขอขอบคุณเพื่อน ๆ ทั้งที่อยู่ในกลุ่มพัฒนาระบบเดียวกัน และเพื่อน ๆ ทั้งหมด ที่ได้ให้ทั้งคำปรึกษา ข้อมูล รวมไปถึงกำลังใจแก่ข้าพเจ้า

นายภูสิทธิ์ รัตนปิยะสุนทร

สารบัญ

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ.....	III
สารบัญ	IV
สารบัญภาพ	V
บทที่	
1. บทนำ.....	1
2. ทฤษฎีที่เกี่ยวข้อง	3
2.1 ระเบียบวิธีรวม และภาษารวม.....	3
2.2 การพัฒนาระบบเชิงวัตถุ	5
2.3 ระเบียบวิธีรวม	6
2.4 การออกแบบ.....	11
2.5 การออกแบบส่วนติดต่อกับฐานข้อมูลและส่วนติดต่อกับผู้ใช้.....	13
3. การวิเคราะห์และออกแบบระบบงาน	16
3.1 การศึกษาความเป็นไปได้.....	16
3.2 การวิเคราะห์ระบบ.....	17
3.3 การออกแบบระบบ	27
3.4 การออกแบบส่วนติดต่อกับผู้ใช้.....	30
3.5 การออกแบบซีเควนซ์โคอะแกรม.....	45
3.6 การสร้างระบบ.....	55
4. สรุปผลการพัฒนา.....	61
บรรณานุกรม.....	63
ภาคผนวก	
ก. คู่มือพัฒนาระบบ	64
ก.1 ยูสเคสโคอะแกรม	64
ก.2 คลาสโคอะแกรม.....	65
ก.3 ซีเควนซ์โคอะแกรม	72
ก.4 แอคทिवิตี้โคอะแกรมของบริการของคลาส	87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

ภาคผนวก

ข. คู่มือการใช้ระบบ	104
ข.1 การติดตั้งทางด้านเซิร์ฟเวอร์	104
ข.2 การติดตั้งทางด้านไคลเอ็นท์	106
ข.3 การใช้งานระบบงาน	106
ข.4 การใช้ระบบย่อยหนึ่งถึงกำหนดชำระ	108
ข.5 การใช้ระบบย่อยรายละเอียดสัญญา	111
ข.6 ระบบย่อยหน่วยภาษีว่าง	112
ข.7 ระบบย่อยการชำระเงินประจำวัน	117
ข.8 ระบบย่อยหนี้เสีย	119
ข.9 ระบบย่อยลูกหนี้หมดอายุสัญญา	120
ข.10 การทำงานกับรายงาน	122
ค. โปรแกรมต้นฉบับ	123
ค.1 สคริปต์สร้างตาราง	123
ค.2 โปรแกรมต้นฉบับของสโตร์โพรซีเจอร์	127
ประวัติผู้เขียน	170

สารบัญภาพ

หน้า

ภาพที่

2.1 ความสัมพันธ์ระหว่างบุคคลกับแอกเตอร์	7
2.2 ตัวอย่างยูสเคสไคอะแกรม	7
2.3 ซีเควนซ์ไคอะแกรมเพื่อช่วยหาคลาส	9
2.4 การสร้างตารางให้กับออปเจต	13
2.5 การสร้างตารางจากความสัมพันธ์ชนิดคลาสร้อย	13
3.1 ยูสเคสไคอะแกรมของระบบลูกหนี้เช่า	17
3.2 แอคติวิตีไคอะแกรมของระบบบัญชีลูกหนี้ และส่วนที่เกี่ยวกับระบบลูกหนี้เช่า	18
3.3 ซีเควนซ์ไคอะแกรมของยูสเคสสัญญาถึงกำหนดชำระ	20
3.4 ซีเควนซ์ไคอะแกรมของยูสเคส รายละเอียดสัญญา	21
3.5 ซีเควนซ์ไคอะแกรมของยูสเคสหน่วยอาศัยที่ว่าง	22
3.6 ซีเควนซ์ไคอะแกรมของยูสเคสการชำระเงินประจำวัน	23
3.7 ซีเควนซ์ไคอะแกรมของยูสเคสนี้เสีย	24
3.8 ซีเควนซ์ไคอะแกรมของยูสเคสลูกหนี้หมดอายุสัญญา	25
3.9 คลาสไคอะแกรมที่คัดออกมาจากซีเควนซ์ไคอะแกรมระดับการวิเคราะห์ระบบ	26
3.10 คลาสไคอะแกรมระดับออกแบบสุดท้าย (ส่วนธุรกิจ) แสดงเฉพาะแอทริบิว	29
3.11 คลาสส่วนติดต่อกับผู้ใช้	30
3.12 ส่วนติดต่อกับผู้ใช้ที่เป็นเมนูหลัก	31
3.13 ส่วนอินเตอร์เฟซของระบบงานย่อยที่นำข้อมูลการชำระเงินเข้าระบบ	32
3.14 คลาสไคอะแกรมส่วนติดต่อกับผู้ใช้เพื่อรับข้อมูลการชำระเงินเข้าระบบ	33
3.15 ส่วนติดต่อกับผู้ใช้ของระบบย่อยรายละเอียดสัญญา	34
3.16 คลาสไคอะแกรมของระบบย่อย รายละเอียดสัญญา	35
3.17 อินเตอร์เฟซหน่วยพักอาศัยว่าง ตามเขตควบคุม	36
3.18 หน่วยพักอาศัยว่างตามโครงการ	36
3.19 หน้าต่างระบบย่อยรายละเอียดหน่วยอาศัยที่ว่าง	37
3.20 คลาสไคอะแกรมของระบบย่อยหน่วยอาศัยที่ว่าง	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.21 อินเทอร์เน็ตของระบบย่อย การชำระเงินประจำวัน	38
3.22 คลาสไดอะแกรมของระบบย่อยการชำระเงินประจำวัน	39
3.23 อินเทอร์เน็ตของระบบย่อยหนี้เสีย	40
3.24 คลาสไดอะแกรมของระบบย่อยหนี้เสีย	40
3.25 อินเทอร์เน็ตของระบบย่อยลูกหนี้ถึงกำหนดชำระ	41
3.26 อินเทอร์เน็ตของระบบย่อยหนี้ถึงกำหนดชำระตาม โครงการ	42
3.27 คลาสไดอะแกรมของระบบย่อยหนี้ถึงกำหนดชำระ	43
3.28 อินเทอร์เน็ตของระบบย่อยลูกหนี้หมดอายุสัญญา	43
3.29 คลาสไดอะแกรมของระบบย่อยลูกหนี้หมดอายุสัญญา	44
3.30 ซีเควนซ์ไดอะแกรมของหน้าต่างระบบถึงกำหนดชำระตามเขตปกครอง	46
3.31 ซีเควนซ์ไดอะแกรมของหน้าต่างหนี้ถึงกำหนดชำระตาม โครงการ	46
3.32 ซีเควนซ์ไดอะแกรมของหน้าต่างรายละเอียดสัญญา	47
3.33 ซีเควนซ์ไดอะแกรมของหน้าต่างหน่วยพักอาศัยที่ว่างตามเขต	48
3.34 ซีเควนซ์ไดอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตาม โครงการ	49
3.35 ซีเควนซ์ไดอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตามหน่วยพักอาศัย	49
3.36 ซีเควนซ์ไดอะแกรมของหน้าต่างการชำระเงินประจำวัน	50
3.37 ซีเควนซ์ไดอะแกรมของหน้าต่างหนี้เสีย	51
3.38 ซีเควนซ์ไดอะแกรมของหน้าต่างลูกหนี้หมดอายุสัญญา	52
3.39 รายงานหนี้ถึงกำหนดชำระ	53
3.40 รายงานรายละเอียดสัญญา	53
3.41 รายงานหน่วยอาศัยที่ว่าง	53
3.42 รายงานหน่วยอาศัยที่ว่างประวัติการทำสัญญา	54
3.43 รายงานการชำระเงินประจำวัน	54
3.44 รายงานหนี้เสีย	54
3.45 รายงานลูกหนี้หมดอายุสัญญา	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
3.46 โครงสร้างคลาสโลเคชั่น	55
3.47 โครงสร้างคลาสโปรเจค.....	56
3.48 โครงสร้างคลาสโมเดล.....	56
3.49 โครงสร้างคลาสบิวคิงไทป์	56
3.50 โครงสร้างคลาสคอมโพเนนท์ไอเทม.....	56
3.51 โครงสร้างคลาสคอมโพเนนท์.....	56
3.52 โครงสร้างคลาสโควเทชั่น	56
3.53 โครงสร้างคลาสนูนิต.....	57
3.54 โครงสร้างคลาสเรนท์คอนแทรค	57
3.55 โครงสร้างคลาสคัสทอเมอร์	57
3.56 โครงสร้างคลาสคอนแทรค	58
3.57 โครงสร้างคลาสเพเมนท์	58
3.58 โครงสร้างคลาสเพเมนท์คิตเทล.....	58
3.59 โครงสร้างคลาสแอดเคาท์	59
3.60 โครงสร้างสถาปัตยกรรมของระบบ.....	59
ก.1 ยูสเคสไดอะแกรมระบบบัญชีลูกหนี้	64
ก.2 คลาสไดอะแกรมระบบบัญชีลูกหนี้.....	65
ก.3 โครงสร้างคลาสโลเคชั่น.....	66
ก.4 โครงสร้างคลาสโปรเจค	66
ก.5 โครงสร้างคลาสโมเดล	66
ก.6 โครงสร้างคลาสบิวคิงไทป์.....	67
ก.7 โครงสร้างคลาสคอมโพเนนท์.....	67
ก.8 โครงสร้างคลาสคอมโพเนนท์ไอเทม	68
ก.9 โครงสร้างคลาสโควเทชั่น	68
ก.10 โครงสร้างคลาสนูนิต	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

ก.11	โครงสร้างคลาสคอนแทรก.....	69
ก.12	โครงสร้างคลาสเรนท์คอนแทรก.....	70
ก.13	โครงสร้างคลาสคัสทอเมอร์.....	70
ก.14	โครงสร้างคลาสเพเมนต์.....	71
ก.15	โครงสร้างคลาสเพเมนต์คีเทล.....	71
ก.16	โครงสร้างคลาสแอกเคพท์.....	72
ก.17	คลาสไดอะแกรมของหน้าต่างอินเทอร์เน็ตเฟส.....	73
ก.18	หน้าต่างระบบย่อยหนึ่งถึงกำหนดชำระ.....	73
ก.19	ซีควเอนซ์ไดอะแกรมของหน้าต่างระบบหนึ่งถึงกำหนดชำระตามเขตปกครอง.....	74
ก.20	อินเทอร์เน็ตเฟสของระบบย่อยหนึ่งถึงกำหนดชำระตาม โครงการ.....	75
ก.21	ซีควเอนซ์ไดอะแกรมของหน้าต่างหนึ่งถึงกำหนดชำระตาม โครงการ.....	75
ก.22	ส่วนติดต่อกับผู้ใช้ของระบบย่อยรายละเอียดสัญญา.....	76
ก.23	ซีควเอนซ์ไดอะแกรมของหน้าต่างรายละเอียดสัญญา.....	77
ก.24	อินเทอร์เน็ตเฟสหน่วยพักอาศัยว่าง ตามเขตควบคุม.....	78
ก.25	ซีควเอนซ์ไดอะแกรมของหน้าต่างหน่วยพักอาศัยที่ว่างตามเขต.....	78
ก.26	หน่วยพักอาศัยว่างตาม โครงการ.....	79
ก.27	ซีควเอนซ์ไดอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตาม โครงการ.....	80
ก.28	หน้าต่างระบบย่อยรายละเอียดหน่วยอาศัยที่ว่าง.....	81
ก.29	ซีควเอนซ์ไดอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตามหน่วยพักอาศัย.....	81
ก.30	อินเทอร์เน็ตเฟสของระบบย่อย การชำระเงินประจำวัน.....	82
ก.31	ซีควเอนซ์ไดอะแกรมของหน้าต่างการชำระเงินประจำวัน.....	83
ก.32	อินเทอร์เน็ตเฟสของระบบย่อยหนึ่งเสีย.....	84
ก.33	ซีควเอนซ์ไดอะแกรมของหน้าต่างหนึ่งเสีย.....	84
ก.34	อินเทอร์เน็ตเฟสของระบบย่อยลูกหนี้หมดอายุสัญญา.....	85
ก.35	ซีควเอนซ์ไดอะแกรมของหน้าต่างลูกหนี้หมดอายุสัญญา.....	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

หน้า

ภาพที่

ก.36 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสแอคเคาท์.....	87
ก.37 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสบิวคิงไทป์.....	88
ก.38 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสคอมโพเนนท์ไอเทม	88
ก.39 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสคอมโพเนนท์.....	89
ก.40 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสคอนแทรก.....	89
ก.41 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสคัสทอเมอร์	93
ก.42 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสโลเคชั่น	94
ก.43 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสโมเดล	95
ก.44 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสเพเมนต์.....	95
ก.45 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสเพเมนต์ดีเทล	97
ก.46 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสโปรเจก.....	98
ก.47 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสเรนท์คอนแทรก.....	100
ก.48 แอคทีวิตี้ โคอะแกรมของเมธอร์ดของคลาสนูนิต	100
ข.1 การใช้โปรแกรม “SQL Plus 8.0” สร้างโครงสร้างข้อมูล.....	105
ข.2 การใช้โปรแกรม “Net8 Easy Config” สร้างการเชื่อมต่อ	105
ข.3 หน้าต่างโปรแกรมหลัก	106
ข.4 หน้าต่างระบบย่อยใส่ข้อมูลการชำระเงิน	107
ข.5 หน้าต่างระบบย่อยหนี้ถึงกำหนดชำระ.....	108
ข.6 รายงานหนี้ถึงกำหนดชำระ	109
ข.7 หน้าต่างรายละเอียดลูกหนี้ถึงกำหนดชำระของโครงการ	110
ข.8 หน้าต่างระบบย่อยรายละเอียดสัญญา.....	111
ข.9 รายงานรายละเอียดสัญญา	112
ข.10 หน้าต่างระบบย่อยหน่วยอาศัยว่าง.....	113
ข.11 รายงานหน่วยอาศัยว่าง	114
ข.12 หน้าต่างหน่วยอาศัยว่างตามโครงการ.....	115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
ข.13 หน้าต่างรายละเอียดหน่วยอาศัย.....	116
ข.14 รายงานประวัติการทำสัญญา	116
ข.15 หน้าต่างระบบขอยการชำระเงินประจำวัน.....	117
ข.16 รายงานการชำระเงิน	118
ข.17 หน้าต่างระบบขอยหนี้เสีย.....	119
ข.18 รายงานหนี้เสีย.....	120
ข.19 หน้าต่างระบบขอยลูกหนี้หมดอายุสัญญา.....	121
ข.20 รายงานลูกหนี้หมดอายุสัญญา.....	122



บทที่ 1

บทนำ

โครงการนี้เป็นการโครงการพัฒนาระบบสารสนเทศโดยใช้วิธีการเชิงวัตถุ ที่ใช้กรณีศึกษา และข้อมูลจากระบบบัญชีลูกหนี้เช่าของการเคหะแห่งชาติ ระบบนี้เป็นระบบย่อยจากระบบรวมคือระบบรับบัญชีลูกหนี้ ซึ่งมีส่วนประกอบอื่น คือ ระบบรับชำระ, ระบบตั้งหนี้, ระบบลูกหนี้เช่าซื้อ, ระบบที่ดินส่วนเกิน และระบบประណอมหนี้

ระบบงานเดิมของการเคหะแห่งชาติจะเป็นระบบที่ทำงานบนเครื่องคอมพิวเตอร์แบบรวมศูนย์ และสื่อสารกับระบบย่อยอื่น ๆ ผ่านทางรายงานและการส่งแผ่นดิสก์ นอกจากนั้นบางระบบงาน ก็เป็นระบบที่ทำงานด้วยมือและกระดาษอยู่ ดังนั้นโครงการนี้จึงเป็นการออกแบบระบบใหม่หมด เพื่อเป็นการรองรับความต้องการใหม่ ๆ และขยายส่วนงานเพิ่มขึ้นด้วย

ระบบงานของการเคหะชาตินั้น จะเป็นระบบที่เริ่มด้วยการตั้งหนี้ขึ้นมา เพื่อเป็นบัญชีหลักอ้างอิงในการกำหนดบัญชีหนี้ โดยการตั้งหนี้นี้จะกระทำพร้อม ๆ กับการสร้างโครงการใหม่ ๆ ของการเคหะแห่งชาติ หลังจากนั้นเมื่อมีการซื้อขายหน่วยพักอาศัย ก็จะมีการสร้างเป็นสัญญาขึ้นมา ซึ่งก็จะมีรายละเอียดต่าง ๆ ของการชำระเงิน เช่นจำนวนงวด และจำนวนเงินที่ต้องจ่ายต่องวด ข้อมูลเหล่านี้บางส่วนจะถูกส่งไปให้ฝ่ายที่มีหน้าที่รับชำระ ซึ่งอาจจะหมายถึงธนาคาร หรืออาจจะหมายถึง เจ้าหน้าที่ของการเคหะแห่งชาติประจำเขตต่าง ๆ เมื่อลูกค้ามาชำระเงินประจำงวด ข้อมูลการชำระก็จะถูกส่งมาที่ระบบคอมพิวเตอร์ส่วนกลางซึ่ง ระบบรับชำระก็จะนำข้อมูลการชำระเข้าสู่ระบบ จากนั้นระบบลูกหนี้เช่า และ ระบบลูกหนี้เช่าซื้อ ก็จะประมวลผล และให้รายงานตามที่กำหนด ในขณะที่เดียวกัน ระบบประណอมหนี้ และ ระบบที่ดินส่วนเกิน ก็จะทำหน้าที่ของคนถ้าเกิดกรณีที่ต้องใช้งานขึ้น

จะเห็นได้ว่าระบบลูกหนี้เช่านี้จะเป็นระบบที่รับข้อมูลที่สร้างโดยระบบรับชำระหนี้ และนำข้อมูลมาประมวลผลเพื่อสร้างรายงานตามที่ต้องการ

การวิเคราะห์, ออกแบบ และพัฒนาระบบในโครงการนี้ จะกระทำโดยใช้การวิเคราะห์และออกแบบโดยเชิงวัตถุ คืออ้างอิงตามมาตรฐาน UML (Unified Modeling Language) คือออกแบบระบบเป็นออบเจกต์ที่ทำงานร่วมกันเพื่อให้ได้ระบบที่ทำงานได้จริง การออกแบบนั้นก็เช่นกัน ก็จะ

ออกแบบเป็นเชิงวัตถุ หลังจากนั้น ก็จะนำสิ่งที่ออกแบบได้ไปพัฒนาด้วยเครื่องมือที่ช่วยการพัฒนา ระบบโดยใช้ระบบฐานข้อมูลแบบเชิงสัมพันธ์

สังเกตได้ว่า ในช่วงการออกแบบ และมาสู่การพัฒนาระบบนั้นจำเป็นจะต้องมีการแปลง จากการพัฒนาเชิงวัตถุ มาเป็นการพัฒนาระบบโดยใช้ฐานข้อมูลเชิงสัมพันธ์เพื่อแสดงรูปแบบข้อมูล ในเชิงวัตถุ จำเป็นต้องมีการออกแบบเป็นพิเศษเพื่อกิจกรรมนี้

วิธีการออกแบบแบบนี้จะมีข้อดี คือสามารถได้ข้อดีของการออกแบบแบบเชิงวัตถุคือ มีความคล่องตัวในการเปลี่ยนแปลงมากกว่า ข้อมูลมีความเป็นนามธรรม (Abstract) มากขึ้น เมื่อมีความเปลี่ยนแปลงที่ส่วนหนึ่ง การเปลี่ยนแปลงนั้นก็จะมีผลกระทบต่อส่วนอื่น หรือถ้ามีการจะ น้อยที่สุด トラバドที่ส่วนสำหรับการติดต่อสื่อสารระหว่างแอปเจคนั้นยังคงเดิม นอกจากข้อดีเหล่านี้แล้ว ก็ยังมีข้อดีอื่น ๆ ของการออกแบบระบบเชิงวัตถุคือ สามารถนำระบบที่ออกแบบไว้แล้วกลับมา ใช้ใหม่ได้ โดยไม่จำเป็นต้องออกแบบ พัฒนา และทดสอบหาข้อผิดพลาดใหม่ ทำให้ระบบที่ พัฒนาขึ้น มีข้อผิดพลาดน้อยที่สุด ส่งผลต่อความน่าเชื่อถือของระบบ และมีระยะในการพัฒนาสั้นลง

นอกจากนี้การใช้ระบบฐานข้อมูลเชิงสัมพันธ์ในการพัฒนาระบบก็มีข้อดีคือ มีเครื่องมือ และระบบให้เลือกใช้มากมาย ทำให้สามารถเลือกทางเลือกที่ดีที่สุดทั้งทางด้านเทคนิค และทาง เศรษฐศาสตร์ นอกจากนั้น เนื่องจากเทคโนโลยีฐานข้อมูลเชิงสัมพันธ์ได้รับการพัฒนามาเป็นระยะ เวลานาน ทำให้เครื่องและระบบที่มีอยู่ในปัจจุบันมีประสิทธิภาพสูง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ระเบียบวิธีรวม และภาษารวม

เมื่อการโปรแกรมเชิงวัตถุได้รับความนิยมแพร่หลายในยุคหลัง ๆ ทำให้เกิดความต้องการระเบียบวิธีการวิเคราะห์ ออกแบบ และพัฒนาระบบในแนวทางเชิงวัตถุด้วยเช่นกัน ในช่วงแรกก็มีหลาย ๆ แนวความคิดเกิดขึ้น แต่ละแนวความคิดก็มีวิธี และเครื่องมือเฉพาะ ทำให้เกิดความสับสนขึ้น ไม่ว่าจะ เป็นความสับสนในระเบียบวิธีพัฒนาเองที่มีหลายมาตรฐาน และความสับสนเมื่อเกิดความจำเป็นต้องนำมาตรฐานเหล่านั้นมารวมกันกำหนดเป็นมาตรฐานภาษากลาง

จนในปี 1995 จึงเกิดความคิดในการรวมระเบียบวิธีทั้งหมดเข้าด้วยกัน โดยรวมเอาข้อดีและลักษณะเด่นของวิธีทั้งหมดเข้าด้วยกัน และเรียกเป็นชื่อใหม่ว่า ยูเอ็มแอล (UML - Unified Modeling Language) (Eriksson, 1998) มีวัตถุประสงค์ของการออกแบบคือ ต้องการให้เป็นภาษาสำหรับสร้างแบบจำลองให้กับระบบที่จะพัฒนา โดยพยายามทำให้สิ่งที่ออกแบบกับสิ่งที่พัฒนาขึ้นมีความใกล้เคียงกันมากที่สุด รองรับระบบงานที่ซับซ้อนได้ และสุดท้ายคือมีสื่อกลางในการสื่อสารที่สามารถเข้าใจได้ทั้งโดยมนุษย์และโดยเครื่องคอมพิวเตอร์

ส่วนประกอบที่สำคัญของยูเอ็มแอลที่ใช้ในการอธิบายระบบที่สมบูรณ์แบบจำเป็นต้องประกอบด้วย สิ่งแรกคือ มุมมอง (View) ซึ่งจะมีหลายมุมมองสำหรับหลายบุคคลที่เกี่ยวข้อง ภายในมุมมองจะต้องประกอบจากแผนผัง (Diagram) หลาย ๆ แผนผังรวมกัน ภายในแต่ละแผนผังก็จะมีองค์ประกอบหลาย ๆ องค์ประกอบรวมกัน เช่นออปเจก คลาส และความสัมพันธ์ระหว่างกัน

โดยมุมมองที่สำคัญของ UML จะมีอยู่ 5 มุมมองด้วยกัน ประกอบด้วย

1. มุมมองยูสเคส (Use Case View) เป็นมุมมองของหน้าที่ของระบบ ซึ่งบุคคลภายนอกที่ใช้ระบบจะมองเห็น ใช้สื่อสารระหว่างผู้ออกแบบระบบและผู้ใช้ระบบ (รวมทั้งผู้ตัดสินใจสร้างระบบ)

2. มุมมองลอจิกคอลล (Logical View) จะเป็นมุมมองที่ให้รายละเอียดว่าแต่ละหน้าที่ของระบบในยูสเคสนั้นสามารถสร้างได้อย่างไร หมายความว่ารวมทั้งทางด้าน สเตตติค และทางไดนามิกส์ มุมมองนี้จะเป็นมุมมองที่ใช้มากในการออกแบบระบบ

3. มุมมองของส่วนประกอบ (Component View) จะบรรยายว่าระบบจะถูกแยกออกเป็นโมดูลย่อยอย่างไร และขึ้นต่อกันอย่างไรบ้าง อาจจะรวมไปถึงทรัพยากรที่ใช้ และความคืบหน้าในการพัฒนาด้วย

4. มุมมองคอนเคอเรนซ์ (Concurrent View) เป็นมุมมอง นอนฟังก์ชันนัล (Non Functional) ที่บอกว่าสามารถแยกระบบเป็น โพรเซส (Process) ย่อย ๆ ได้อย่างไรเพื่อเพิ่มประสิทธิภาพให้กับระบบ

5. มุมมองดีพลอยเมนต์ (Deployment View) จะเป็นมุมมองที่บอกว่าจะสามารถสร้างระบบที่ใช้งานจริงได้อย่างไร ประกอบด้วยอะไรบ้างทั้งทางฮาร์ดแวร์และซอฟต์แวร์

โดยลักษณะแล้ว มุมมองจะบ่งบอกถึงระดับ หรือสถานะของการพัฒนาระบบว่าอยู่ในขั้นตอนใด เช่นอยู่ในขั้นตอนของการวิเคราะห์ ออกแบบ หรือพัฒนาระบบ โดยสิ่งเดียวกันสามารถจะอยู่ในมุมมองที่ต่างกัน ได้ โดยขึ้นอยู่กับว่าใครเป็นผู้ใช้มุมมองนั้น และแต่ละมุมมองก็จะมีลักษณะสะท้อนความต้องการของระดับนั้น เช่นมุมมองที่ระดับคอม โปเนนท์ก็จะรองรับการจัดวางระบบว่าชิ้นส่วนไหนของระบบควรจะถูกออกแบบและจัดวางอยู่ที่ส่วนใด ส่วนมุมมองยูสเคสก็จะเน้นที่ความต้องการของระบบ จึงเหมาะสมกับขั้นตอนการวิเคราะห์ระบบเป็นต้น

ในแต่ละมุมมองก็ต้องประกอบด้วยแผนผังซึ่งใน UML มีทั้งหมด 9 แผนผัง คือ

1. แผนผังยูสเคส (Use Case Diagram) จะเป็นแผนผังที่แสดงหน้าที่ของระบบ และผู้ที่เกี่ยวข้อง สามารถบอกถึงขอบเขตของระบบได้ว่าส่วนใดถูกรวมเป็นส่วนหนึ่งของระบบ ส่วนใดเป็นส่วนภายนอกที่ระบบจะต้องติดต่อดูด้วย

2. แผนผังคลาส (Class Diagram) จะแสดงคลาสที่มีอยู่ในระบบ และความสัมพันธ์ต่อกัน

3. แผนผังออบเจกต์ (Object Diagram) จะเป็นแผนผังที่แสดงออบเจกต์ ซึ่งเป็นอินสแตนซ์ (Instance) หนึ่งของคลาส ที่เวลาหนึ่ง ๆ จึงเป็นตัวอย่างระบบ

ต่อไปนี้เป็นแผนผังในกลุ่มไดนามิกส์

4. แผนผังสเตท (State Diagram) เป็นส่วนเสริมแผนผังคลาส ที่บอกถึงสถานะต่าง ๆ ที่คลาสสามารถเป็นได้ โดยจะเปลี่ยนไปตามเหตุการณ์ที่เกิด

5. แผนผังซีควเอนซ์ (Sequence Diagram) จะเป็นแผนผังที่แสดงลำดับการสื่อสารระหว่างออบเจกต์ตามลำดับเวลา

6. แผนผังคอลเลบอเรต (Collaboration Diagram) จะเป็นแผนผังที่บอกปฏิสัมพันธ์ระหว่างออบเจกต์ สามารถทำหน้าที่ได้เหมือนแผนผังซีควเอนซ์ โดยแผนผังซีควเอนซ์จะเน้นเส้นเวลา แต่แผนผังคอลเลบอเรตจะเน้นความสัมพันธ์ระหว่างออบเจกต์

7. แผนผังกิจกรรม (Activity Diagram) จะเป็นแผนผังที่แสดงการเกิดขึ้นของกิจกรรมภายในระบบ เงื่อนไข การตัดสินใจ เส้นทางที่เลือก กิจกรรมที่เป็นผลสืบเนื่อง

8. แผนผังองค์ประกอบ (Component Diagram) จะแสดงระบบหรือโปรแกรมที่อยู่ในรูปแบบในความเป็นจริง โดยแผนผังนี้จะให้ข้อมูลในทางปฏิบัติของคลาส ความสัมพันธ์ รูน ความขึ้นต่อกัน

9. แผนผังดีพลอยเมนต์ (Deployment Diagram) จะแสดงสถาปัตยกรรมของระบบในทางปฏิบัติ ทั้งในด้านฮาร์ดแวร์ และซอฟต์แวร์

โดยเนื้อแท้แล้ว แต่ละชนิดแผนผังก็จะมีลักษณะไวยากรณ์ของตนเอง ว่ามีวิธีการใช้อย่างไร เหมือนกับเป็นลักษณะทางภาษาอย่างหนึ่ง และแต่ละแผนผังก็มีความสามารถในการอธิบายส่วนที่ต่างกัน ไปของระบบขึ้นอยู่กับการใช้งาน โดยแผนผังที่จำเป็นนั้นก็มียูสเคส คลาส และซีควเอนซ์หรือคอแลบอเรชัน ส่วนอื่น ๆ นั้นจะเป็นส่วนอธิบายเพิ่มเติมเพื่อให้การอธิบายระบบมีความสมบูรณ์ยิ่งขึ้น

ด้วยเครื่องมือช่วยการวิเคราะห์และออกแบบระบบของยูเอ็มแอลในปัจจุบันนั้น ถ้าผู้ออกแบบสามารถใช้เครื่องมือเหล่านี้ และสามารถจะเขียนโดยใช้ไวยากรณ์โดยถูกต้องตามหลักไวยากรณ์นั้น เครื่องมือที่มีความสามารถให้ความสะดวกได้หลายอย่างเช่น ตรวจสอบความถูกต้องระหว่างขั้นตอนการพัฒนา ระบบ ตรวจสอบความขัดแย้งกันเองภายในระบบ เช่นการที่มีกิจกรรมหนึ่ง แต่ไม่มีระบบมารองรับ หรือการมีกิจกรรมชนกันเป็นต้น นอกจากนั้นท้ายที่สุดเครื่องมือยังมีความสามารถในการสร้างโครงสร้างของแอปพลิเคชัน หรือ โครงสร้างฐานข้อมูลให้อีกด้วย

2.2 การพัฒนาระบบเชิงวัตถุ

ด้วยเครื่องมือที่มีคือยูเอ็มแอลนั้น ยังไม่สามารถพัฒนาระบบออกมาเป็นระบบที่ทำงานได้จริง จำเป็นต้องมีส่วนประกอบอีกส่วนหนึ่งที่มีความสำคัญอย่างยิ่ง นั่นคือระเบียบวิธีที่จะใช้วิเคราะห์และออกแบบระบบ ซึ่งการพัฒนาเชิงวัตถุนั้นก็มียุทธวิธีของตัวเองเช่นเดียวกัน ซึ่งก็มีหลายระเบียบวิธี แต่ในที่นี้จะเน้นกล่าวถึงระเบียบวิธีที่ได้จากการประมวลผลหลาย ๆ วิธีเข้าด้วยกัน เรียกว่าระเบียบวิธีรวม หรือ Unified Approach

ระเบียบวิธีรวม หรือ Unified Approach นั้น เป็นระเบียบวิธีที่มิใช่เป็นแบบทางเดียว นั่นคือเมื่อเรากระทำสิ่งใดสิ่งหนึ่งไปแล้ว เราสามารถย้อนกลับมาปรับปรุงสิ่งที่ออกแบบไปแล้วได้ ทำให้เราสามารถปรับปรุงสิ่งที่เราออกแบบไปได้เรื่อย ๆ ทำให้เหมาะกับปัญหาที่มีความไม่แน่นอนสูง เช่นมีความต้องการที่ไม่แน่นอน ยังไม่ทราบแน่ชัด (Bahrami, 1999:39-54)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนการวิเคราะห์ระบบนั้น วัตถุประสงค์หลักก็คือ พยายามหาขอบเขตของระบบงาน และความต้องการของระบบ ดังนั้นสิ่งที่ต้องทำก็คือ

1.หา Actor ที่มีทั้งหมดในระบบ ซึ่ง Actor หลายถึงสิ่งที่อยู่นอกระบบที่จำเป็นต้องติดต่อกับระบบหมายรวมทั้งคน และระบบอื่น

2.สร้างระบบงานของระบบ โดยใช้ Activity Diagram โดยแผนผังระบบที่สร้างขึ้นนี้จะสร้างขึ้นเพื่อเป็นจุดอ้างอิงถึงกิจกรรมที่มีทั้งหมดในระบบ ลักษณะจะคล้ายโฟลว์ชาร์ต

3.สร้างยูสเคสไดอะแกรม เพื่อระบุกิจกรรมที่ระบบจะให้บริการ โดยอ้างอิงจาก Activity Diagram ว่ามีกิจกรรมใดที่ระบบต้องให้บริการบ้าง และเพื่อบอกว่า Actor แต่ละตัวใช้บริการใดของระบบบ้าง

4.สร้างอินเตอร์แอกชันไดอะแกรม เพื่อนำแต่ละกิจกรรมที่ระบบให้บริการมาลงรายละเอียดว่ากิจกรรมนั้นต้องทำอะไรบ้างในระหว่างออปเจกเพื่อทำให้งานสำเร็จลง

5.ระบุคลาส เป็นการนำออปเจกมารวมกลุ่มเป็นคลาสและระบุคุณสมบัติของแต่ละคลาส และฟังก์ชันที่แต่ละคลาสให้บริการ

ต่อจากนั้นก็นำข้อมูลที่วิเคราะห์ได้มาออกแบบซึ่งวัตถุประสงค์หลักก็คือให้รายละเอียดในส่วนต่าง ๆ ของคลาส ให้รายละเอียดของส่วนติดต่อกับผู้ใช้ ซึ่งมีขั้นตอนดังต่อไปนี้

1. ออกแบบคลาส คือให้รายละเอียด ถึงแอทริบิว, ฟังก์ชัน, ความสัมพันธ์ระหว่างคลาส รวมไปถึง โครงสร้างอื่น ๆ โปรโตคอล และข้อกำหนดอื่น ๆ

2.ออกแบบชั้นที่ใช้ติดต่อ คือส่วนที่ทำหน้าที่เป็นส่วนติดต่อกับ Actor

3.ออกแบบโปรโตไทป์และส่วนติดต่อกับผู้ใช้ เพื่อให้เป็นส่วนที่ให้ผู้ใช้งานได้ค้นหาความต้องการว่าตรงกับที่ดึงเอาไว้หรือไม่

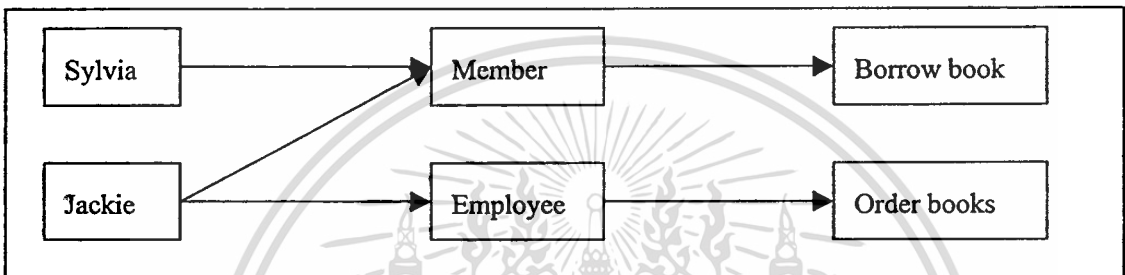
4.เป็นการให้ผู้ผู้ใช้ได้ทดลองติดต่อกับ โปรโตไทป์เพื่อทดสอบความต้องการว่าถูกต้องหรือไม่ และดูว่าระบบมีพฤติกรรมตรงกับที่ต้องการหรือไม่

5.ทบทวนและปรับแก้ระบบ หลังจากที่ใช้ได้ประเมินระบบแล้ว ก็นำเอาสิ่งที่ต้องการเพิ่มเติมนั้นมาออกแบบระบบเพิ่ม แต่ถ้าผู้ใช้ไม่มีความต้องการเพิ่ม ก็นำเอาโปรโตไทป์นั้นมาเป็นส่วนหนึ่งของสัญญาว่าขอบเขตของระบบนั้นมีขอบเขตแค่ไหน

2.3 ระเบียบวิธีรวม

ในหัวข้อที่แล้วได้ให้ภาพรวมว่า การพัฒนาระบบเชิงวัตถุนั้นมีลำดับขั้นตอนเป็นอย่างไร ด้านกว้าง แต่หัวข้อนี้จะป็นรายละเอียดทฤษฎีการพัฒนาาระบบเชิงวัตถุโดยใช้ระเบียบวิธีรวม

อย่างที่กล่าวไปแล้วว่าระเบียบวิธีรวมนั้นจะต้องเริ่มต้นด้วยการมองหาแอกเตอร์ในระบบ โดยแอกเตอร์นั้นสามารถเป็นอะไรก็ได้ตั้งแต่คน เครื่องจักรอื่นที่เกี่ยวข้อง หรือแม้แต่ระบบคอมพิวเตอร์อื่น สิ่งที่คุณวิเคราะห์ต้องทำก็คือมองหาว่าขอบข่ายของระบบนั้นอยู่ที่ใด โดยการพยายามระบุแอกเตอร์ก็จะไปเป็นบอกรขอบเขตไปในตัวว่า สิ่งที่เป็นแอกเตอร์นั้นเป็นสิ่งที่อยู่นอกระบบ ลักษณะของแอกเตอร์ที่ดีก็คือแอกเตอร์ที่ทำหน้าที่บอกรขอบเขตของสิ่งนั้น (Bahrami, 1999: 134) แต่ละคน ๆ เดียวอาจจะมีหลายบทบาท ซึ่งก็ควรจะกลายเป็นคนละแอกเตอร์ มิฉะนั้นระบบก็จะเป็นระบบที่ออกแบบมาขึ้นกับตัวบุคคลแทนที่จะเป็นระบบที่มีความเป็นสากลในการใช้งาน



รูปที่ 2.1 ความสัมพันธ์ระหว่างบุคคลกับแอกเตอร์

เมื่อได้แอกเตอร์แล้วก็สร้างเป็นยูสเคสไดอะแกรมเพื่อระบุถึงบริการที่ระบบมีให้ สิ่งนี้จะเป็นสิ่งที่กำหนดเป็นเป้าหมายของการออกแบบ ว่าตามเป้าหมายแล้วสิ่งที่ต้องการได้เป็นผลลัพธ์นั้นคือระบบที่มีบริการและความสามารถอะไรบ้าง โดยในการสร้างยูสเคสนั้นอาจจะสร้างแอกทิวิตี้ไดอะแกรม สร้างแผนผังระบบงานที่มีอยู่ก็เป็นได้ เพื่อช่วยในการค้นหาว่าที่จุดใดของแอกทิวิตี้ไดอะแกรมบ้างที่เป็นจุดที่เป็นบริการที่ผู้ใช้ระบบต้องการ

ลักษณะของยูสเคสก็คือจะเป็นจุดบริการที่ระบบมีให้ โดยยูสเคสอาจจะไปเรียกใช้ยูสเคสอื่นก็ได้ เพื่อเป็นการขยายระบบดังรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างยูสเคสไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างยูสเคสในรูปที่ 2.2 แสดงให้เห็นบริการที่ระบบตัวอย่างมีให้ ซึ่งมี 6 บริการด้วยกัน นั่นคือ บริการยืมหนังสือ บริการยืมข้ามห้องสมุด บริการคืนหนังสือ บริการทำวิจัย บริการอ่านหนังสือพิมพ์ และบริการจัดซื้อ โดยบริการยืมหนังสือ บริการคืนหนังสือ และบริการยืมข้ามห้องสมุด จะไปเรียกใช้บริการตรวจสอบบัตรต่ออีกช่วงหนึ่ง ซึ่งไดอะแกรมนี้ก็จะสามารถแสดงได้ตามวัตถุประสงค์แล้ว ระบบนี้จะให้บริการอะไรบ้าง และสิ่งใดเป็นส่วนหนึ่งของระบบ ส่วนใดอยู่ภายนอกระบบ สุดท้ายก็สามารถบอกถึงการสื่อสารได้ว่าระบบต้องมีการสื่อสารกับสิ่งใดบ้าง

หลังจากนั้นก็จะเป็นกระบวนการสรรหาคลาสที่เหมาะสม ซึ่งกระบวนการนี้สามารถกระทำได้หลายวิธี คือ การค้นหาค่านาม การหาคลาสมาตรฐาน การใช้ยูสเคส การหาบริการและผู้ร่วมงาน (Bahrami, 1999)

โดยแต่ละวิธีมีรายละเอียดคือ วิธีการค้นหาค่านามนั้นจะเป็นการเอาเอกสารปัญหา หรือจากการสอบถามกับผู้ใช้รวบรวมข้อมูลทั้งหมด และเลือกเอาเฉพาะค่านามออกมาจากเอกสารหรือข้อความเหล่านั้น และจัดแยกค่านามเหล่านั้นเป็นหมวดหมู่ เช่นมีความเป็นไปได้สูง คลุมเครือ และไม่เกี่ยวข้อง วิธีการนี้จะทำให้ผู้วิเคราะห์สามารถเลือกได้ว่า ค่านามใดบ้างที่มีความน่าจะเป็นที่จะกลายเป็นคลาสที่ดีในอนาคต โดยลักษณะของคลาสที่ดี ก็คือมีหน้าที่ชัดเจน ไม่มีหน้าที่ที่ซ้ำกันกับคลาสอื่น เป็นคลาสที่มีหลาย ๆ คุณสมบัติเพราะคลาสที่มีเพียงแอทริบิวเดียว จริง ๆ อาจจะเป็นเพียงแอทริบิวของคลาสอื่น และให้ระวางค่านามที่มีคำวิเศษณ์ขยายอยู่ เพราะเพียงคำขยายที่ต่างกันเล็กน้อยอาจจะหมายถึงคลาสที่แตกต่างกันมาก ก็มีโอกาสนั้นไปได้

วิธีการที่สองคือการใช้คลาสมาตรฐาน โดยมาตรฐานเหล่านี้สร้างขึ้นจากประสบการณ์ของผู้มีประสบการณ์ในการพัฒนาระบบเชิงวัตถุ โดยจะเสนอแนวทางในการมองหาคลาสที่ดี โดยแบ่งเป็นหมวดหมู่ เช่นคลาสที่เป็นแนวคิดหลักของระบบ เช่นถ้าเป็นระบบที่ทำงานเกี่ยวกับการประเมินประสิทธิภาพ คลาสที่ดีที่เป็นแนวคิดหลักของระบบก็คือคลาสของประสิทธิภาพ ต่อมาก็คือคลาสที่เป็นเหตุการณ์ หรือบ่งถึงระยะเวลา คลาสที่บอกถึงความเป็นองค์กร หรือการรวมกันของกลุ่มคน คลาสของคนที่บ่งบอกถึงหน้าที่ที่แตกต่างกันไป คลาสของสถานที่ และคลาสของสิ่งที่จับต้องได้ เช่นเครื่องมืออุปกรณ์ต่าง ๆ เป็นต้น วิธีการนี้จะให้มองหาตามหัวข้อที่ได้จำแนกไว้แล้ว และมองหาไปในระบบว่ามีสิ่งใดบ้างที่น่าจะเป็นคลาสที่น่าสนใจ หรือเป็นตัวเลือกที่น่าสนใจที่จะเป็นคลาสที่ดีในอนาคต

วิธีการต่อมาคือการใช้ยูสเคส วิธีการนี้เป็นวิธีการที่ดูจะเป็นหลักเป็นฐาน และให้ความน่าเชื่อถือมากที่สุด เพราะวิธีการนั้นตรงไปที่ปัญหา นั่นคือจากยูสเคสแต่ละยูสเคสนั้น ผู้วิเคราะห์ก็จะพัฒนาออกมาให้เห็นถึงกิจกรรมที่ดำเนินอยู่ภายในยูสเคสว่ามีอะไรเกิดขึ้นบ้าง โดยสามารถแสดงได้ในรูปของซีควเอนซ์ไดอะแกรมในระดับของการวิเคราะห์ เมื่อแต่ละยูสเคสได้รับการพัฒนาซีเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคาน์เตอร์ไอทีเข้ามาแล้ว แต่ละซีเคาน์เตอร์ไอทีจะเป็นการแสดงผลการสื่อสารกับระหว่าง
แอปเจตต่าง ๆ ที่อยู่ในระบบ เช่นกระบวนการคิดเงินนั้น จะต้องมีข้อมูลผ่านไปที่ใครบ้างเป็นต้น

ลักษณะของซีเคาน์เตอร์ไอทีเป็นไปดังรูป 2.3 ซึ่งจะเห็นถึงลักษณะของกิจกรรมที่เกิดขึ้น
ในยูสเคส แต่ละยูสเคส เมื่อสังเกตจะเห็นได้ว่ากิจกรรมที่เกิดขึ้นจะเป็นการบังคับไปในตัวว่า จะ
ต้องอาศัยแอปเจตอะไรบ้างในการทำงานให้ลุล่วง ซึ่งแอปเจตเหล่านั้นนั่นเองที่จะเป็นตัวแทนที่ดีของ
คลาสในอนาคต ดังนั้นวิธีนี้ก็จะเป็นการรวบรวมเอาแอปเจตที่ปรากฏในแต่ละซีเคาน์เตอร์ไอที
มารวบรวม รวมไปถึงบริการที่มีให้ และเกิดแต่กันในเรื่องของแอปเจตก็สามารถกลายเป็นโครง
สร้างของคลาสที่ดี ก็จะกลายเป็นเมธอดของคลาสในอนาคตนั่นเอง



รูปที่ 2.3 ซีเคาน์เตอร์ไอทีเพื่อช่วยหาคลาส

วิธีการสุดท้ายจะเป็นวิธีที่เหมือนจะเป็นวิธีสำหรับสอนการมองหาคลาสมากกว่าวิธีที่จะใช้
งานได้จริง เพราะวิธีการนี้จะเป็นการมองหาคลาส โดยการระบุนำที่หรือบริการให้กับคลาสต้น
ทางก่อน และพยายามมองว่าเพื่อจะบรรลุบริการนั้น คลาสนั้นจะต้องอาศัยความช่วยเหลือจากคลาส
ใดบ้าง ซึ่งจะช่วยหาคลาสใหม่ ๆ เข้ามาในระบบ

เมื่อได้คลาสแล้ว ก็จะเป็นระบุถึงรายละเอียดของคลาส เพื่อให้ครบสมบูรณ์เป็นคลาสไอที
แกรมที่สมบูรณ์ ซึ่งสิ่งแรกที่จะต้องระบุก็คือความสัมพันธ์ระหว่างคลาส

ความสัมพันธ์ระหว่างคลาสนั้นสามารถแบ่งได้กว้างเป็น 3 แบบ คือการสื่อสารปกติ ความ
สัมพันธ์แบบคลาสที่เฉพาะเจาะจงลงไป และความสัมพันธ์แบบที่เป็นส่วนประกอบกัน โดยการ
มองหาความสัมพันธ์นั้นสามารถมองได้จากเอกสารที่ระบุปัญหาของระบบเช่นกัน โดยความ
เอกสารนี้เป็นเอกสารที่ส่งวันไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัมพันธ์ แบบที่เป็นการสื่อสารกันนั้น มักจะปรากฏเป็นรูปคำกริยาที่สื่อกันระหว่างนามสองตัว หรือความสัมพันธ์แบบสื่อสารกันเนื่องจากมีตำแหน่งที่ต่างกัน ก็จะอยู่ในรูปคำกริยาที่ตามด้วยคำบุพบท สิ่งที่ควรหลีกเลี่ยงอย่างยิ่งในการสร้างความสัมพันธ์ก็คือ ไม่ควรมีความสัมพันธ์เกินที่ละ 3 คลาส และถ้ามีก็ควรแตกออกเป็นความสัมพันธ์ระหว่างคลาส 2 คลาสที่มีหลายคู่ความสัมพันธ์ และไม่ควรมีความสัมพันธ์ที่เกิดขึ้นเพราะข้อจำกัดของเครื่องมือพัฒนาที่ใช้ มิฉะนั้นระบบที่ออกแบบได้จะเป็นระบบที่ขึ้นกับเครื่องมือ หรือขั้นตอนการพัฒนา และท้ายที่สุดก็จะเป็นการออกแบบที่ไม่ดี เพราะไม่มีความคล่องตัวเพียงพอ และสุดท้ายคือการสื่อสารที่เกิดขึ้นไม่ควรมีลักษณะที่เป็นเฉพาะตัวมากเกินไป การสื่อสารที่เฉพาะเกินไปอาจจะสามารถแตกออกเป็นความสัมพันธ์ทั่วไปหลาย ๆ ความสัมพันธ์ได้ ซึ่งถ้าทำได้ดังนั้น ก็ควรจะทำ เพราะความสัมพันธ์ที่เฉพาะเจาะจงเกินไป จะทำให้ความซับซ้อนของระบบเพิ่มขึ้น โดยไม่จำเป็น

ต่อไปจะเป็นความสัมพันธ์แบบเฉพาะเจาะจง (Super Class / Sub Class) ความสัมพันธ์ลักษณะนี้คือ คลาสที่เป็นคลาสย่อยจะเป็นคลาสที่มีเฉพาะมากกว่าคลาสดั้งเดิม เช่นความสัมพันธ์ระหว่าง หมอ กับหมอฟัน หรือรถสี่ล้อ กับรถยนต์ส่วนบุคคล ลักษณะการค้นหาความสัมพันธ์แบบนี้สามารถทำได้หลายวิธี คือ มองหาจากเอกสารปัญหา โดยหาลักษณะคำนามที่มีการใช้คำขยายต่อกันไปหลาย ๆ ทอดในภาษาอังกฤษ ลักษณะการขยายก็เป็นการเฉพาะเจาะจงขึ้นนั่นเอง วิธีการต่อไปก็คือเป็นรวบรวมคลาสนี้ที่มีขนาดเล็ก ๆ แล้วหาลักษณะที่เหมือน ลักษณะร่วม แล้วผลัดกันลักษณะร่วมเหล่านั้นให้กลายเป็นคลาสดั้งเดิมนั้นเอง วิธีการที่สามก็คือคล้าย ๆ กับวิธีการที่สองแต่จะต่างกัน คือ จะพยายามหาลักษณะร่วมกันให้มากที่สุด และผลัดกันให้มากที่สุด ผลัดกันทุกสิ่งที่มีลักษณะร่วมกันเป็นลักษณะแผนภูมิต้นไม้ โดยหวังผลก็คือ จะทำให้สามารถมีการนำสิ่งที่ร่วมกันนั้นกลับมาใช้ให้มากที่สุด

ลักษณะต่อมาคือลักษณะความสัมพันธ์แบบเป็นส่วนประกอบ เช่น คลาสของรถยนต์จะต้องประกอบด้วย ล้อ ประตู เป็นต้น โดยลักษณะความสัมพันธ์แบบนี้มีสองประการ คือ เป็นแรงและแบบอ่อน แบบแรงนั้นหมายถึงว่า คลาสหลักนั้นประกอบด้วยคลาสย่อย ๆ จริง จะขาดสิ่งใดไปไม่ได้ แต่แบบอ่อนหมายถึง คลาสใหญ่เป็นที่รวบรวมของคลาสย่อย เช่นนักเรียน กับห้องเรียน เป็นต้น

เมื่อได้ความสัมพันธ์ระหว่างคลาสแล้ว สิ่งต่อไปที่จะต้องกำหนดก็คือแอทริบิว และเมธอดที่คลาสนั้นมีให้บริการ โดยลักษณะของแอทริบิวที่ดีคือ มักจะเป็นคำนามต่อท้ายด้วยบุพบท เช่น ราคา ของ บ้าน ราคา ก็จะเป็นแอทริบิวที่ดี ในการออกแบบแอทริบิวนั้น ควรจะคำนึงถึงความง่ายเข้าใจ คือไม่ควรใส่แอทริบิวมากเกินไป และในบางครั้งในเอกสารปัญหาอาจจะไม่ได้มีข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครบทุกอย่าง ผู้วิเคราะห์ออกแบบระบบควรจะไปศึกษาถึงงานด้านนั้น เพราะแเทริบิวหลาย ๆ ตัว อาจจะซ่อนอยู่ และด้วยอาศัยความรู้ในงานด้านนั้น จึงจะสามารถพบได้

ลักษณะต่อมา คือ ไม่ควรมีแเทริบิวที่เป็นอนุพันธ์ของแเทริบิวอื่น เช่น การที่อายุสามารถคำนวณได้จากวันเกิด และวันที่ปัจจุบัน ก็ไม่ควรให้อายุเป็นแเทริบิว เพราะจะทำให้ไม่ค่าที่ไม่แน่นอน และเปลี่ยนแปลงอยู่ตลอดเวลา และมีค่าเดือนที่สำคัญคือไม่ควรมุ่งหาแเทริบิวมากเกินไป จนไม่มีความคืบหน้าในด้านอื่น เพราะอาจจะมีแเทริบิวอื่น ๆ เพิ่มมาภายหลังก็เป็นสิ่งที่เป็นไปได้

ต่อไปจะเป็นพยายามมองหาเมธอร์ดของคลาส ซึ่งจะสังเกตเห็นได้อย่างชัดเจนว่า เมธอร์ดนั้นเป็นสิ่งที่มองหาได้ไม่ยาก โดยดูจากการสื่อสารที่เกิดขึ้นระหว่างออปเจกที่อยู่ในซีควนซ์ไคอะแกรม ที่ได้สร้างออกมาแล้วนั่นเอง

2.4 การออกแบบ

เมื่อเสร็จขั้นตอนการวิเคราะห์แล้ว ก็ถึงขั้นตอนการออกแบบ โดยการออกแบบจะมีหลายระดับ เช่นที่ระดับคลาสก็จะเป็นการให้รายละเอียดมากขึ้น ซึ่งถ้าเป็นแเทริบิวก็จะเป็นการกำหนดชนิดแเทริบิว ถ้าเป็นเมธอร์ดก็จะเป็นการกำหนด โปรโตคอลการสื่อสาร การอินเตอร์เฟส และพารามิเตอร์ที่รับส่งกันนั่นเอง

ลักษณะการออกแบบที่ดีคือ ลักษณะแรก จะต้องทำให้แต่ละโมดูลมีความขึ้นต่อกันน้อยที่สุด เพราะถ้าโมดูลมีการขึ้นต่อกันมาก การจะหยาบโมดูลโคมาใช้ใหม่ ก็จะไม่สามารถทำได้อย่างอิสระ และถ้ามีการเปลี่ยนแปลงเกิดขึ้นก็จะได้รับผลกระทบในวงกว้าง

ลักษณะต่อมาก็คือ แต่ละคลาสควรจะมีวัตถุประสงค์เดียว ซึ่งเป็นลักษณะการออกแบบที่ทำให้การนำกลับมาใช้ใหม่ทำได้อย่างมีประสิทธิภาพ เพราะออปเจกหรือคลาสที่ออกแบบไปแล้วนั้นมีวัตถุประสงค์เดียวที่มีความเป็นสากล ซึ่งจะส่งผลให้มีความขึ้นต่อโมดูลอื่นน้อยลงด้วยเพราะถ้ามีวัตถุประสงค์น้อย ก็มีความจำเป็นจะต้องสื่อสารกับคนอื่นน้อยด้วยเช่นกัน

ลักษณะที่สามก็คือ ควรออกแบบให้มีลักษณะคลาสเป็นคลาสเล็ก ๆ จำนวนมาก ๆ มากกว่าจะเป็นคลาสใหญ่ ๆ ที่มีจำนวนน้อย ๆ การที่ออกแบบคลาสมาเป็นลักษณะนี้จะทำให้สามารถนำคลาสดกลับมาใช้ใหม่ได้ง่าย และการนำคลาสดเข้ามาเพิ่มเติมความสามารถก็จะสามารถทำได้เป็นระบบ นอกจากนั้นแล้ว การที่เป็นคลาสเล็ก ๆ ยังทำให้การดูแลรักษา สามารถทำได้ง่าย เพราะโครงสร้างของคลาสดก็จะง่าย ๆ และมีความชัดเจน ทำให้โอกาสที่จะเกิดข้อผิดพลาดลดลง ไปจนถึง การนำมาสร้างเป็นระบบใหญ่ ก็จะสามารถจัดการ ได้เป็นอย่างดี เพราะไม่มีความซับซ้อนในการสื่อสาร

ลักษณะที่ดีคือ การวิเคราะห์และการออกแบบควรจะมี ความเกี่ยวเนื่องกัน นั่นหมายถึงว่า การออกแบบที่ดีนั้นควรมีที่มา ที่สอดคล้องกับระบบที่ได้วิเคราะห์เอาไว้แล้ว โครงสร้างและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมควรจะมีลักษณะที่สอดคล้องกัน มิใช่เป็น โครงสร้างที่ต่างกันเหมือนไม่ใช่ระบบเดียวกัน จะทำให้การดูแลรักษา และการปรับเปลี่ยนในอนาคตประสบความลำบาก เพราะโครงสร้างที่ ออกแบบไม่ได้สะท้อนปัญหาที่แท้จริง เพียงแต่สามารถทำงานได้บรรลุเท่านั้น

ลักษณะต่อมาคือ การออกแบบควรมีมาตรฐาน (Bahrami, 1999:208) การมีมาตรฐานทำให้ลดความซับซ้อนลง เพราะการสร้างมาตรฐานก็เหมือนกับเป็นการจัดระเบียบองค์กรแบบหนึ่ง ซึ่งเป็นการเพิ่มประสิทธิภาพในการทำงานสร้าง และพัฒนาระบบงาน การค้นหาข้อผิดพลาด หรือ การเพิ่มเติมความสามารถก็จะสามารถทำได้อย่างมีประสิทธิภาพ เพราะทุกอย่างมีมาตรฐานกำหนดไว้หมดแล้ว รวมไปถึงการนำเอาความสามารถเก่ามาใช้ใหม่ก็สามารถทำได้ดี เพราะอยู่ในมาตรฐาน เดียว มิฉะนั้นแล้ว หลาย ๆ กรณีก็จะต้องเริ่มต้นนับหนึ่งใหม่ทั้งหมด

ลักษณะสุดท้ายคือพยายามสนับสนุนการใช้การสืบทอดให้มากที่สุด คือพยายามนำส่วน ประกอบเก่ามาใช้ใหม่ หรือเพิ่มความสามารถเข้าไป แทนที่จะเริ่มพัฒนาสิ่งใหม่ทั้งหมด การนำสิ่ง เก่ามาใช้ใหม่มีข้อดีคือ ทำให้ย่นระยะเวลาในการพัฒนาลงเพราะไม่ได้เริ่มจากความไม่มีอะไรเลย ในขณะที่สิ่งที่ผลิตออกมาได้ก็มีความน่าเชื่อถือมากกว่า เพราะระบบเก่าที่ถูกนำกลับมาใช้ใหม่ ก็เป็น ระบบที่ผ่านการใช้งานมาแล้ว ระยะเวลาหนึ่ง ได้ผ่านการทดสอบความน่าเชื่อถือแล้ว ดังนั้นการนำ กลับมาใช้ใหม่ ก็สามารถยืนยันความถูกต้องของการทำงานได้โดยไม่ต้องทดสอบส่วนเก่านั้นอีก เป็นการเพิ่มทั้งประสิทธิภาพ และความน่าเชื่อถือของระบบไปพร้อม ๆ กัน

ดังนั้นในการออกแบบคลาสจากส่วนเดิมก็มีหน้าที่เพียงดูแลการจัดแบ่งคลาสให้มีหน้าที่ และ แนวคิดตามหลักการออกแบบที่ได้กล่าวไปแล้ว สิ่งที่ต้องทำก็คือจัดแบ่งขอบเขตการงานระหว่าง คลาส ออกแบบชนิดของตัวแปรที่เป็นข้อมูลภายใน และออกแบบโปรโตคอลสำหรับทำหน้าที่เป็น อินเทอร์เฟซสำหรับให้บริการกับคลาสอื่น

โดยลักษณะของบริการที่คลาสที่ให้สามารถสรุปออกมาได้กว้าง ๆ ก็คือ เป็นผู้สร้าง คือเป็น บริการที่ให้บริการสร้างออปเจกใหม่ในคลาสนั้นขึ้นมา เป็นบริการทำลายออปเจกเก่าที่ใช้งานเสร็จ แล้ว บริการแปลงค่าต่าง ๆ เพื่อให้สามารถทำงานสอดคล้องกันระหว่างสองคลาสที่ต่างกัน ได้ ให้ บริการคัดลอกข้อมูลระหว่างออปเจก ให้บริการตั้งค่าของออปเจก ให้บริการให้ค่าของออปเจก ให้ บริการติดต่อกับกับอุปกรณ์ภายนอกที่คลาสนั้นมีหน้าที่เชื่อมต่ออยู่ด้วย และให้บริการงานที่เกี่ยวข้อง กับระบบงานจริงของระบบ เช่นระบบการรับเงินก็มีหน้าที่สรุปยอดเงินด้วย เป็นต้น

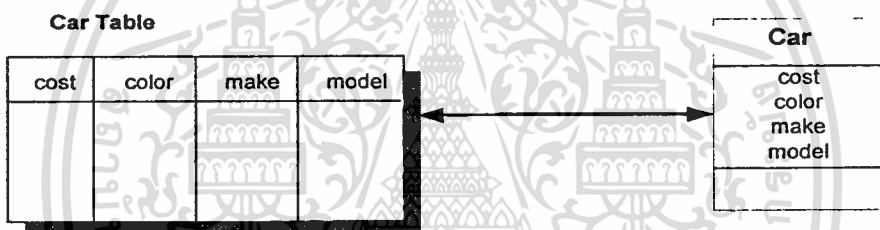
ข้อควรระวังในการออกแบบก็คือ ควรจะรักษาแนวคิดของคลาสเอาไว้ให้ได้ คือถ้ากำหนด ในตอนต้นแล้วว่าจะให้คลาสไหนทำหน้าที่อะไร ก็ควรจะรักษาแนวคิดนั้นเอาไว้ให้ได้ เพราะเมื่อ การพัฒนาผ่านไปเรื่อย ๆ คลาสก็มีแนวโน้มที่จะเพิ่มความสามารถขึ้นเรื่อย ๆ นั้นหมายความว่า คลาส มีแนวโน้มที่จะโตขึ้นเรื่อย ๆ ซึ่งทำให้มีความเป็นไปได้สูงที่คลาสนั้นจะสูญเสียจุดยืนไป ไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานในหน้าที่ที่เกินบทบาทดั้งเดิมไปมาก โดยถ้าคลาสใดมีขนาดใหญ่มาก ๆ ก็เป็นไปได้ที่ผู้ออกแบบจะแยกคลาสนั้นออกเป็นคลาสย่อย ๆ ลง โดยพยายามหาจุดยื่นให้กับแต่ละคลาสต่อไป ด้วยวิธีการนี้จะยังคงทำให้การออกแบบระบบสามารถเป็นไปอย่างมีมาตรฐานได้ และได้ระบบงานที่ไม่มี ความสับสนแปรปรวนในตัวเอง

2.5 การออกแบบส่วนติดต่อกับฐานข้อมูล และส่วนติดต่อกับผู้ใช้

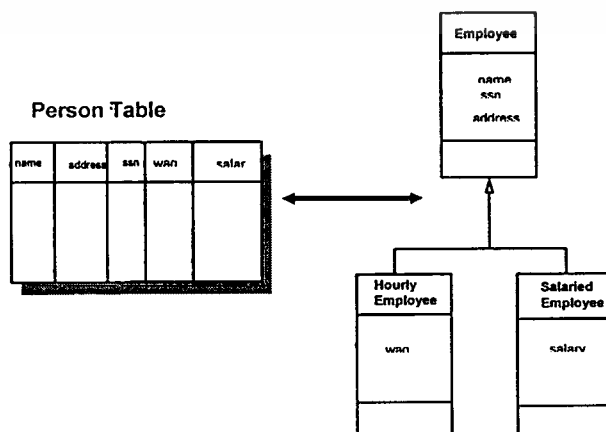
ตามแนวคิดการออกแบบส่วนติดต่อกับฐานข้อมูลแล้ว จะออกแบบให้มีคลาสเฉพาะอีก คลาสหนึ่งขึ้นมาเฉพาะสำหรับติดต่อกับฐานข้อมูล โดยจะสร้างเฉพาะคลาสที่ต้องการจัดเก็บข้อมูล ของคนลงฐานข้อมูลเท่านั้น

โดยแนวคิดแล้วคลาสที่ต้องการจัดเก็บข้อมูลของคนลงฐานข้อมูลนั้น จะสามารถสร้างตารางในฐานข้อมูลเชิงสัมพันธ์ได้หนึ่งตารางต่อหนึ่งคลาสพอดี ดังรูปที่ 2.4



รูปที่ 2.4 การสร้างตารางให้กับแอปเจค

นอกจากการสร้างตารางแบบหนึ่งต่อหนึ่งดังนี้แล้ว ยังมีความสัมพันธ์ในรูปแบบอื่น ๆ เช่น หนึ่งตาราง สำหรับมากกว่า 1 คลาส โดยคลาสเหล่านั้นจำเป็นต้องมีบางส่วนของข้อมูลร่วมกัน แบบที่สามคือ แบบหนึ่งตารางต่อหลายคลาสที่มีความสัมพันธ์แบบลักษณะเฉพาะ คือมีคลาสต้นฉบับกับคลาสย่อยที่มีลักษณะเฉพาะเจาะจงลงไป ดังรูปที่ 2.5



รูปที่ 2.5 การสร้างตารางจากความสัมพันธ์ชนิดคลาสย่อย

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของบริษัทฯ ขอสงวนสิทธิ์ในเนื้อหา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการสร้างตารางชนิดนี้ก็จะเป็นที่ฐานในการแปลงจากระบบเชิงวัตถุไปเป็นระบบฐานข้อมูลเชิงสัมพันธ์ และเหลือสิ่งสุดท้ายที่จะต้องพิจารณา ก็คือความสัมพันธ์ระหว่างคลาสที่เป็นความสัมพันธ์แบบการสื่อสาร เมื่อได้รับการแปลงให้อยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์จำเป็นจะต้องได้รับการดูแลความสัมพันธ์นั้นด้วย ซึ่งถ้าจะพิจารณากันแล้วความสัมพันธ์นั้นก็สามารถแปลงให้อยู่ในรูปของพอร์เนจ็ลยีของฐานข้อมูลเชิงสัมพันธ์ได้ นั่นคือถ้ามีความสัมพันธ์กัน เช่นแบบหนึ่งต่อหลาย ๆ ออปเจก ก็ให้มีการเอาไพรมารีจ็ลยีของคลาสต้นทางมาเป็นพอร์เนจ็ลยีของคลาปลายทางเพียงเท่านี้ก็สามารรถำรงข้อกำหนดและความสัมพันธ์ระหว่างกันไว้ได้แล้ว

ต่อไปจะเป็นหลักวิธีในการออกแบบส่วนติดต่อกับฐานข้อมูล คือขั้นแรกนั้น ดูว่าในจำนวนคลาสทั้งหมด มีคลาสใดบ้างที่ต้องการเก็บข้อมูลลงฐานข้อมูล ทุก ๆ คลาสที่ต้องการเก็บข้อมูลลงฐานข้อมูลก็จัดการสร้างคลาสที่เหมือน ๆ กันขึ้นมาเพื่อทำการนี้โดยเฉพาะ โดยใช้คลาสที่สร้างขึ้นใหม่ขึ้นไปอยู่ในกลุ่มคลาสที่จะใช้เข้าถึงฐานข้อมูล

ขั้นต่อมา ก็เป็นการให้ความสัมพันธ์ระหว่างคลาสที่ใช้เข้าถึงข้อมูลด้วยกัน ว่ามีความสัมพันธ์กันแบบใด โดยมากก็จะมีความสัมพันธ์เหมือนกับคลาสต้นฉบับ และก็จะมีความสัมพันธ์กับคลาสต้นฉบับในแบบ การสื่อสารกัน หรือการเป็นส่วนหนึ่งของคลาสต้นฉบับ ขั้นต่อมา ก็เป็นการปรับปรุงโมเดลที่เกิดขึ้น โดยวัตถุประสงค์ก็คือต้องการลดความซ้ำซ้อน คือถ้ามีคลาสใดที่ทำหน้าที่เหมือนกันก็จัดการลดรูปให้เหลือเพียงคลาสเดียวเท่านั้น

ส่วนเมธอร์คของคลาสเหล่านั้นก็จะถูกแยกออกเป็นอีกคลาสหนึ่งเหมือนกันเช่นกัน โดยลักษณะแล้ว คลาสที่สร้างขึ้นใหม่ทั้งสองชุด จะมีลักษณะคือคลาสที่เป็นแอทริบิวก็จะมีเมธอร์ค ซึ่งคลาสนี้ก็จะกลายเป็นตารางในฐานข้อมูลนั่นเอง และคลาสิกอีกชุดก็คือคลาสที่มีแต่เมธอร์คแต่ไม่มีแอทริบิว ซึ่งคลาสนี้ก็จะทำหน้าที่ติดต่อกับฐานข้อมูลนั่นเอง

คลาสิกทั้งสองชุดที่สร้างขึ้นใหม่นั้นจะเรียกว่าแอกเซสคลาสิก หรือเป็นคลาสที่ใช้เข้าถึงฐานข้อมูลนั่นเอง โดยคลาสิกพวกนี้จะถูกเรียกใช้งานโดยคลาสิกจริงอีกที

หลังจากจุดนี้คลาสที่ทำหน้าที่เป็นตารางในฐานข้อมูลจำเป็นต้องได้รับการนอร์มอลไลซ์ เพื่อป้องกันความผิดปกติที่อาจจะเกิดขึ้น

ต่อไปจะเป็นการออกแบบส่วนติดต่อกับผู้ใช้ ซึ่งมีลักษณะคล้ายกับส่วนติดต่อกับข้อมูลในบางแง่ก็จะเป็นการพิจารณาว่าส่วนใดบ้างที่ต้องการติดต่อกับผู้ใช้ เมื่อเจอส่วนนั้นแล้วก็จะเป็นการรายละเอียดในส่วนนั้นลงไป

ดังนั้นขั้นตอนการออกแบบส่วนติดต่อกับผู้ใช้ก็จะมีขั้นตอนคือ ขั้นแรกต้องดูก่อนว่ามีส่วนใดบ้างที่ต้องการติดต่อกับผู้ใช้ ซึ่งสามารถดูได้หลายที่ เช่นอาจจะดูว่ามีคลาสใดบ้างที่ต้องการติดต่อกับผู้ใช้ หรืออาจจะดูจากยูสเคสและซีเควนซ์ไดอะแกรม ว่ามีส่วนใดบ้างที่ระบบจะต้องการเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อกับแอดเดอร ที่จุดนั้นเองก็กำหนดไว้ว่าให้เป็นจุดที่ต้องการติดต่อกับผู้ใช้ กระบวนการนี้เรียกว่ากระบวนการออกแบบทางมหภาค

ต่อไปก็จะเป็นออกแบบในรายละเอียดว่าแต่ละส่วนติดต่อกันมีนั้นมีรายละเอียดเป็นอย่างไร คือว่าส่วนติดต่อนั้นจะต้องติดต่อกับคลาสใดบ้างเพื่อให้บรรลุถึงกิจกรรม เมื่อได้ในรายละเอียดแล้วก็จัดการสร้างโปรโตไทป์ขึ้นมาเพื่อเป็นต้นแบบ

ขั้นต่อไปก็เป็นการทดสอบเพื่อดูความพอใจของผู้ใช้ว่าต้องการปรับเปลี่ยนสิ่งใดหรือไม่ และสุดท้ายก็จะเป็นทบทวนปรับเปลี่ยนไปเรื่อย ๆ จนกว่าจะได้ระบบที่ดีที่พอใจผู้ใช้

การออกแบบในระดับมหภาคนั้นมีขั้นตอนย่อยคือ ดูว่าแต่ละคลาสมีคลาสใดบ้างที่ต้องการการติดต่อกับผู้ใช้ โดยอย่างที่กล่าวไปแล้วว่าสามารถดูได้จากยูสเคสและซีควเอนซ์ไดอะแกรม เมื่อได้คลาสที่ต้องการติดต่อกันแล้ว ก็สร้างคลาสสำหรับติดต่อกันขึ้นมา และให้ความสัมพันธ์ระหว่างกัน และระหว่างคลาสสเปคติดของมัน

การออกแบบภาคจุลภาคนั้นเป็นการให้รายละเอียดจริงในส่วนที่ทำหน้าที่ติดต่อกัน มีกฎช่วยในการออกแบบคือ ข้อแรกต้องทำให้การติดต่อนั้นเรียบง่าย เข้าใจง่าย และใช้ง่ายที่สุด ความซับซ้อนไม่ช่วยให้การใช้งานระบบดีขึ้น กฎข้อต่อมาคือทำให้ผู้ใช้สามารถเห็นทุก ๆ สิ่งทุก ๆ อย่างภายในหน้าต่างนั้น ไม่ว่าจะเกิดเป็นกระบวนการที่เกิดขึ้น หรือข้อมูลที่มีการเปลี่ยนแปลง จำเป็นจะต้องเห็นทันที ไม่มีส่วนใดที่ควรให้ผู้ใช้คาดเดาเอาเอง เพราะจะทำให้ความรู้สึกละเลยต่อการใช้งาน และอาจทำให้เกิดความผิดพลาดจากการคาดเดาได้

ต่อมาคือทุก ๆ กิจกรรมที่เกิดขึ้นจะต้องทำให้ผู้ใช้สามารถควบคุมได้ จะต้องไม่มีกิจกรรมใดที่ผู้ใช้ต้องการจะควบคุม แต่ไม่สามารถทำได้ เช่นการยกเลิกกิจกรรมเป็นต้น ซึ่งหมายความว่าส่วนติดต่อของเราจะต้องยอมให้เกิดความผิดพลาดได้ และสามารถยกเลิกได้ และควรมีการตอบสนองทันทีที่เกิดเหตุการณ์ขึ้น และส่วนติดต่อนั้นควรมีความคงเส้นคงวาของการใช้สัญลักษณ์และข้อความเพื่อทำให้ผู้ใช้เกิดความเคยชิน

ข้อแนะนำในการออกแบบอื่นเช่น สิ่งที่สำคัญควรวางไว้ที่มุมซ้ายบน และสิ่งไม่สำคัญที่มุมขวาล่างเหมือนการอ่านหนังสือ ข้อความที่จะแจ้งควรจะเป็นทางแนะนำ และเป็นทางบวก กระชับและเข้าใจได้ง่าย ปุ่มควบคุมควรอยู่ในตำแหน่งที่ทำให้การกดปุ่มไม่บังข้อมูลที่อยู่ในหน้าต่าง ไม่ควรให้สี หรือใช้ฟอนต์มากเกินไป เพราะจะเป็นการหันเหความสนใจผู้ใช้ออกจากงาน และใช้ความพยายามส่วนใหญ่ไปในการทำความเข้าใจระบบการสื่อสารที่โปรแกรมพยายามจะสร้าง

บทที่ 3

การวิเคราะห์และออกแบบระบบงาน

3.1 ศึกษาความเป็นไปได้

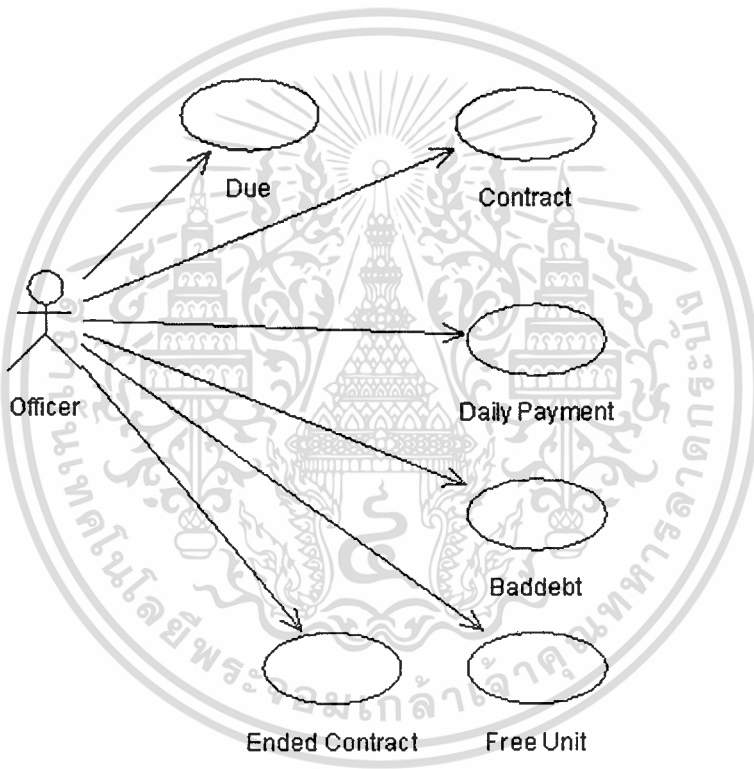
การศึกษความเป็นไปได้ในที่นี้นั้น จะเป็นการศึกษาความเป็นไปได้ในแง่ของวิทยาการเท่านั้น เพราะความเป็นไปได้ในด้านอื่น ๆ เช่นทางด้านเศรษฐศาสตร์นั้นอยู่นอกเหนือจากขอบข่ายของการศึกษาในโครงการนี้ เนื่องจากเป็นสถานะจำลองที่สร้างขึ้น โดยใช้กรณีศึกษาที่มีอยู่จริง และจากการสอบถามผู้ใช้ ทำให้ทราบว่าระบบนี้เป็นระบบที่มีอยู่ก่อนแล้ว ความต้องการที่เพิ่มขึ้นนั้นอยู่ในรูปแบบของการปรับปรุงข้อมูล และความสามารถต่าง ๆ ให้เข้ากับสภาพแวดล้อมใหม่ที่เปลี่ยนไป เช่นทางด้านฮาร์ดแวร์ ระบบปฏิบัติการ การสื่อสาร และด้านอื่น ๆ เป็นต้น

ทำให้เกิดความท้าทายใหม่คือ ถ้าทำการวิเคราะห์และออกแบบระบบใหม่ทั้งหมดเป็นในเชิงวัตถุจะมีความเป็นไปได้เพียงใด ในตอนแรกที่ทำการศึกษา ก็เห็นว่า การวิเคราะห์และออกแบบในเชิงวัตถุนั้นมีความเป็นไปได้สูง เพราะวิทยาการทางการวิเคราะห์และออกแบบเชิงวัตถุได้ถูกพัฒนาไปมากแล้วเช่นกัน ไม่ว่าจะเป็นในเรื่องของระเบียบวิธี หรือเป็นในเรื่องของเครื่องมือ แต่ปัญหาก็คือ ความเป็นไปได้ในการพัฒนาระบบที่ใช้งานได้จริง และมีประสิทธิภาพเพียงพอต่อความต้องการของผู้ใช้นั้นยังเป็นเรื่องที่น่าสงสัย เพราะเครื่องมือที่พัฒนาระบบฐานข้อมูลเชิงวัตถุนั้น ยังไม่เป็นที่แพร่หลาย และการพัฒนาระบบในลักษณะนี้อาจมีความเสี่ยงสูง เช่นความเสี่ยงที่เครื่องมือที่มีจะไม่สามารถทำกิจกรรมที่ต้องการได้ ในขณะที่เครื่องมือการพัฒนาทั่วไปที่เป็นเชิงกระบวนการนั้น มีให้เลือกหลากหลายทำให้มีความสามารถหลากหลายด้วยเช่นกัน

ดังนั้นจึงมีแนวความคิดต่อมา ก็คือ ถ้าเป็นระบบผสม คือมีการวิเคราะห์และออกแบบในเชิงวัตถุ แต่เมื่อถึงขั้นการพัฒนาระบบจริง ให้ใช้วิทยาการในการพัฒนาระบบแบบระบบฐานข้อมูลเชิงสัมพันธ์ทั่วไป ซึ่งเครื่องมือช่วยวิเคราะห์และออกแบบระบบในปัจจุบันก็มีความสามารถในการให้แนวทางว่า การเชื่อมสองระบบเข้าด้วยกันนั้นจะเป็นในแนวทางใด เอกสารอ้างอิงที่แนวแนวคิดการเชื่อมสองวิธีการเข้าด้วยกันนั้นก็สามารถหาได้ ดังนั้นแนวทางนี้จึงมีความเป็นไปได้จึงมีสูงมาก

3.2 การวิเคราะห์ระบบ

จะเห็นได้ว่าระบบนี้จะเป็นระบบย่อยที่ทำงานร่วมกับระบบย่อยอื่น ๆ เพื่อให้ได้งานเป็นระบบรวมคือระบบบัญชีลูกหนี้ ดังนั้นจึงสามารถสร้างภาพ ขอบข่ายของระบบได้คือ มีส่วนติดต่อกับภายนอกคือ ผู้ใช้ระบบ คือหมายถึงเจ้าหน้าที่ที่ควบคุมดูแลระบบลูกหนี้เช่า และอีกด้านก็คือข้อมูลที่ได้รับมาจาก ระบบรับชำระแต่เนื่องจากข้อมูลจากระบบรับชำระนั้น อยู่ในฐานข้อมูลเดียวกันกับระบบลูกหนี้เช่า ซึ่งก็คือเป็นฐานข้อมูลเดียวกัน ดังนั้นจึงถึงว่าไม่มีความจำเป็นต้องแปลง หรือมีตัวกลางที่ทำหน้าที่ติดต่อ ดังนั้นจึงมองว่าข้อมูลนั้นก็เป็นส่วนหนึ่งของระบบเช่นกัน ทำให้สามารถสร้างยูสเคสไดอะแกรม ที่บอกถึงบริการหรือกิจกรรมที่ระบบมีให้ดังรูปที่ 3.1



รูปที่ 3.1 ยูสเคสไดอะแกรมของระบบลูกหนี้เช่า

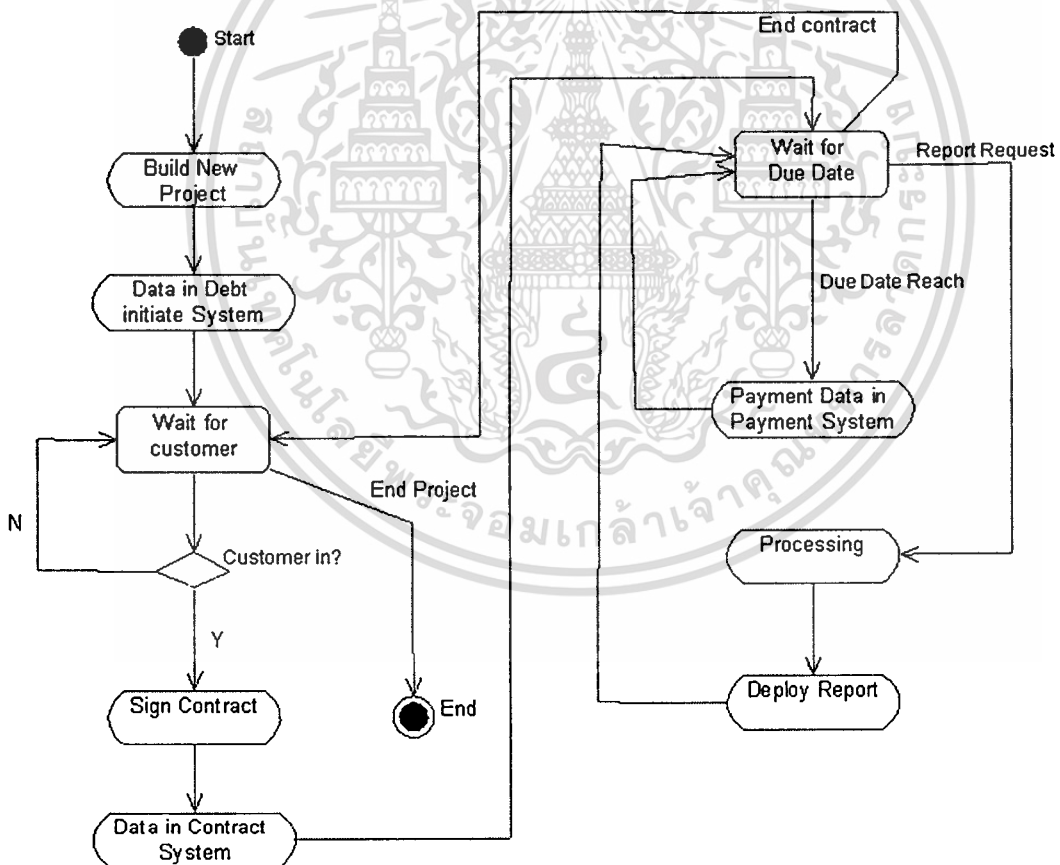
ลักษณะกิจกรรมของการจัดการลูกหนี้เช่าทั้งหมดก็คือ ขึ้นแรกเมื่อมีโครงการสร้างขึ้น ก็จำเป็นต้องใส่ข้อมูลโครงการเข้าไปในระบบ ซึ่งเป็นส่วนที่ระบบตั้งนี้จะต้องทำ รวมทั้งการขอเลขที่บ้าน จากส่วนงานที่เกี่ยวข้องเพื่อใส่เป็นส่วนหนึ่งของข้อมูลโครงการ และเมื่อมีลูกค้ายมาทำการจองแล้ว ก็จะถูกสร้างเป็นสัญญาขึ้นเป็นส่วนงานของระบบการทำสัญญา เมื่อถึงกำหนดจ่ายเงิน ลูกค้าก็จะไปจ่ายที่สถานที่รับชำระ ซึ่งก็คือ ธนาคาร หรือหน่วยงานของการเคหะประจำเขต เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สข. ข้อมูลการชำระเงินก็จะถูกรับเข้าระบบโดยผ่านทางระบบรับชำระ และข้อมูลก็จะถูกส่งมาที่ ศูนย์กลางเพื่อเข้าสู่ฐานข้อมูลของระบบ และให้ข้อมูลนั้นแก่ระบบย่อยอื่น ๆ ที่เกี่ยวข้องนำข้อมูลไป ใช้อีกที ซึ่งก็คือ ระบบลูกหนี้เข้าที่ทำการพัฒนาก็จะให้ข้อมูลเกี่ยวกับลูกหนี้เข้า ระบบอื่น ๆ ก็จะนำ ข้อมูลมาประมวลผลตามงานของตน เช่นระบบลูกหนี้เข้าซื้อ ระบบประนอมหนี้ ระบบที่ดินส่วน เกิน เป็นต้น

ที่จุดนี้เอง เมื่อเจ้าหน้าที่ที่ต้องการข้อมูลการชำระเงิน หรือข้อมูลอื่น ๆ ที่เกี่ยวกับลูกหนี้ เข้า ก็จะติดต่อกับระบบลูกหนี้เข้าเพื่อขอสืบค้นข้อมูล หรือรายงานที่เกี่ยวข้อง ระบบลูกหนี้เข้าก็จะ ไปหาข้อมูล ไม่ว่าจะจากการดึงขึ้น การทำสัญญา หรือการชำระเพื่อให้ได้ข้อมูลที่ผู้ใช้ต้องการ

สามารถสร้างเป็นแอกทิวิตี้ไดอะแกรม ตามมาตรฐานของยูเอ็มแอล ของกิจกรรมรวมของ ระบบบัญชีลูกหนี้ได้ดังต่อไปนี้



รูปที่ 3.2 แอกทิวิตี้ไดอะแกรมของระบบบัญชีลูกหนี้ และส่วนที่เกี่ยวกับระบบลูกหนี้เข้า

เมื่อทราบกิจกรรมการทำงานรวมคร่าว ๆ แล้ว ก็จะเป็นการเริ่มวิเคราะห์ระบบ ซึ่งวิธีการในการวิเคราะห์ระบบนั้นมีหลายวิธีด้วยกัน ไม่ว่าจะเป็นการวิเคราะห์โดยอาศัยคำถามจากเอกสารที่เกี่ยวข้องกับระบบ วิธีการมองหาคลาสพื้นฐานจากหมวดหมู่ที่มีผู้จัดเอาไว้แล้ว การหาคลาสจากกิจกรรม และการมองหาคลาสจากกิจกรรมของคลาสซึ่งจะบอกถึงคลาสอื่นที่ป็นผู้ร่วมงาน ซึ่งในการพัฒนาระบบนี้จะเป็นการวิเคราะห์ระบบ ในแนวทางการวิเคราะห์โดยการอาศัยการวิเคราะห์กิจกรรมของระบบ หรือการวิเคราะห์ทางยูสเคสนั่นเอง

การศึกษาข้อมูลปัญหาของระบบทำให้สามารถสร้างยูสเคสขึ้นมาได้ 6 ยูสเคสด้วยกัน ดังที่ได้แสดงไว้แล้ว แต่ละยูสเคสนั้นก็จะสามารถสร้างเป็นผังการสื่อสาร โดยแสดงออกด้วยซีเควนซ์ไดอะแกรม ซึ่งแต่ละซีเควนซ์ไดอะแกรมก็จะแสดงการสื่อสารที่เกิดขึ้นระหว่างออปเจกอะไรบางอย่าง ทำให้สามารถบอกได้ว่ามีคลาสอะไรบางอย่างที่มีความจำเป็นต้องมีในระบบ รวมไปถึงบริการที่จะต้องให้มี รวมไปถึงข้อความที่จะต้องส่ง ไปหาออปเจกอื่นเพื่อทำให้งานลุล่วงไปได้

ด้วยวิธีการนี้เมื่อได้ซีเควนซ์ไดอะแกรมในระดับการวิเคราะห์แล้ว ก็จะสามารถระบุคลาสในระดับการวิเคราะห์ แอทริบิว และบริการที่แต่ละคลาสที่ให้ไว้ ซึ่งข้อมูลจากคลาสไดอะแกรม และซีเควนซ์ไดอะแกรมจะเป็นเครื่องมือหลักในการวิเคราะห์ระบบ ก่อนจะส่ง ไปสู่การออกแบบระบบต่อไป

จากการศึกษาข้อมูลของระบบการเช่านั้น ทำให้ทราบว่า ยูสเคสทั้ง 6 ของระบบนั้นประกอบด้วย การทำงานเกี่ยวกับลูกหนี้ที่ถึงกำหนดชำระเงิน รายละเอียดของสัญญา ข้อมูลของหน่วยอาศัยที่ว่าง ข้อมูลการชำระเงินประจำวัน ข้อมูลเกี่ยวกับหนี้เสีย และข้อมูลลูกหนี้หมดอายุสัญญา

ซึ่งรายละเอียดนั้นก็คือ ในการทำงานเกี่ยวกับลูกหนี้ที่ถึงกำหนดชำระนั้น ข้อมูลที่ผู้ใช้งานอยากทราบและเกี่ยวข้องก็มี จำนวนหน่วยอาศัยที่ตรงกับเงื่อนไข ค่าเช่าของหน่วยอาศัย จำนวนเงินรวมของแต่ละโครงการ ซึ่งสามารถสืบค้นได้ตามเขตการดูแล และสืบค้นต่อไปได้ตามโครงการ

ต่อมาคือรายละเอียดของสัญญา สิ่งที่ต้องการทราบก็มี อายุของสัญญา ดอกเบี้ยมาตรฐานที่ทำเป็นสัญญา ค่าเช่าที่ตกลงกันไว้ เงื่อนไขการประกันค่าเสียหาย และข้อมูลเกี่ยวกับลูกค้า

การทำงานเกี่ยวกับหน่วยอาศัยที่ว่างนั้นจะทำงานกับข้อมูลที่ประกอบด้วย รหัสของหน่วยอาศัยที่ว่าง โครงการของหน่วยอาศัยนั้น และประวัติการเช่าก่อนหน้า

ต่อมาคือข้อมูลเกี่ยวกับการชำระเงินประจำวัน ซึ่งเป็นการให้ข้อมูลการทำงานประจำวันอย่างละเอียด ซึ่งต้องประกอบด้วย หมายเลขของทรานแซคชันของข้อมูล หมายเลขของหน่วยอาศัย ซึ่งผูกติดกับหมายเลขสัญญาที่ทำกัน ไปจนถึงวันที่จ่าย ค่าเช่า และข้อมูลว่าการจ่ายเงินนั้นเป็นการจ่ายสำหรับงวดที่เท่าใด

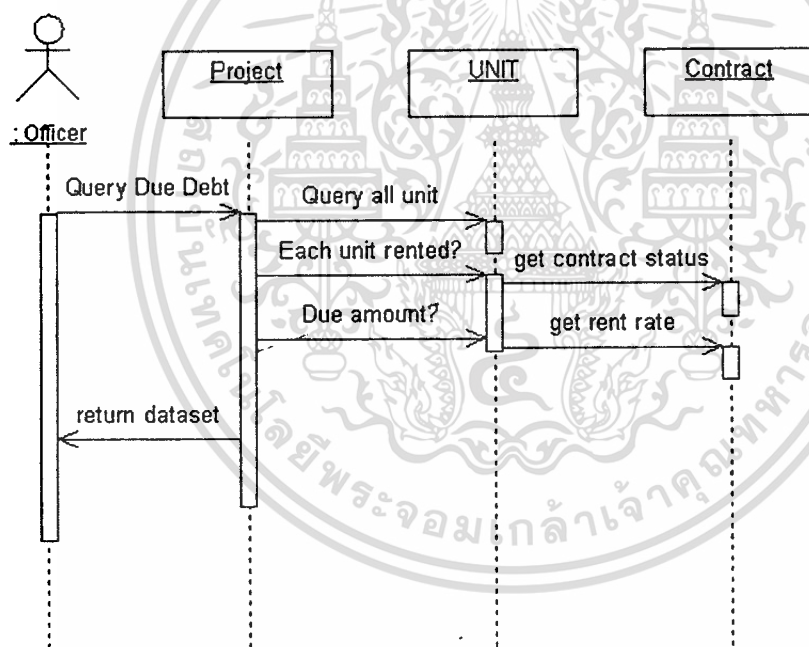
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาเป็นระบบเกี่ยวกับหนี้เสีย หรือหนี้ค้างชำระ ซึ่งก็ต้องการข้อมูลคือ หมายเลขโครงการ หมายเลขหน่วยอาศัย แบบของหน่วยอาศัย และจำนวนหนี้เสีย ซึ่งแบ่งเป็นหนี้เสียจากค่าเช่า และหนี้เสียจากค่าประกัน

และสุดท้ายคือระบบลูกหนี้หมดอายุสัญญา ซึ่งมีข้อมูลที่ต้องการคือ หมายเลขหน่วยอาศัย เลขที่สัญญา วงคแรกของค่าเช่า ค่าเช่า สถานะสัญญา ข้อมูลลูกค้า และค่าเช่า ค่าประกันค้างชำระ

ต่อไปจะเป็นการแสดงให้เห็นถึงซีเควนซ์ไดอะแกรม ซึ่งแต่ละซีเควนซ์ไดอะแกรม จะเป็นการอธิบายถึงแต่ละยูสเคส ว่าแต่ละยูสเคสนั้นมีกิจกรรมภายในอะไรบ้าง ซึ่งจะแสดงให้เห็นทั้งหมดทุกยูสเคสที่มีในระบบ และท้ายที่สุดก็จะได้ข้อมูลสำหรับนำไปสร้างเป็นคลาสไดอะแกรม

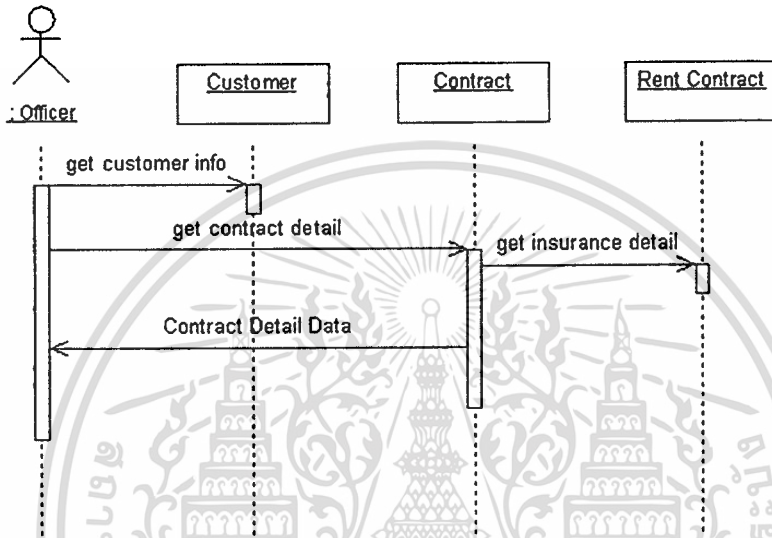
จากข้อมูลจากยูสเคสทั้งหมด 5 รายการ ซึ่งเป็นระดับการวิเคราะห์ระบบ นั้นทำให้สามารถสร้างเป็น ซีเควนซ์ไดอะแกรมในระดับการวิเคราะห์ระบบได้ดังต่อไปนี้คือ



รูปที่ 3.3 ซีเควนซ์ไดอะแกรมของยูสเคสสัญญาถึงกำหนดชำระ

จากรูปที่ 3.3 จะแสดงให้เห็นถึงกิจกรรมที่กระทำ ก็คือผู้ใช้จะร้องขอข้อมูล หนี้ที่ถึงกำหนดชำระ ซึ่งในกรณีนี้จะเป็นหนี้ที่เกิดจากการเช่าอย่างเดียวนั่น โดยร้องขอข้อมูลไปที่ออฟเจต โปรเจก จากนั้นออฟเจตโปรเจกก็จะร้องขอข้อมูลรายการหน่วยอาศัยทั้งหมดที่อยู่ในโครงการของตนไปที่ออฟเจตยูนิต จากนั้นก็จะเอาข้อมูลหน่วยอาศัยที่ได้มาสอบถามถึงแต่ละหน่วยอาศัยว่าถูกเช่าไปหรือยัง โดยสอบถามไปยังออฟเจตยูนิต และออฟเจตยูนิตก็จะสอบถามต่อไปยังออฟเจตคอนเทกสตันเป็นเอกสารที่ส่งจนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

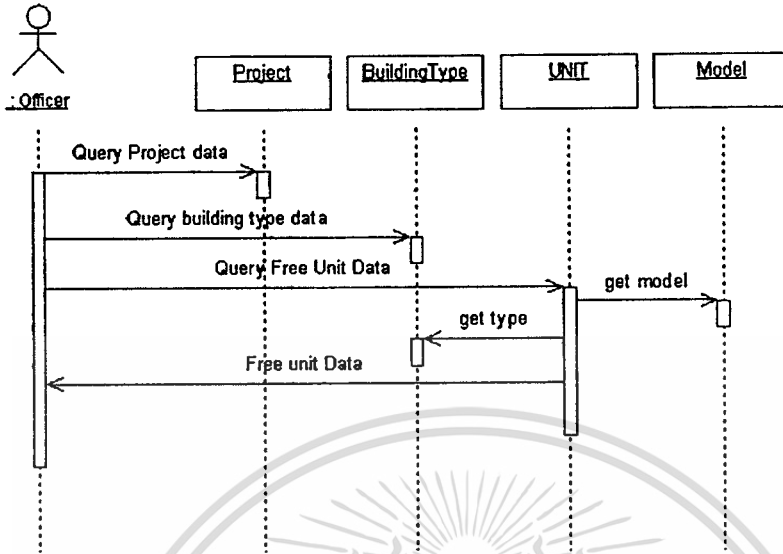
แทรก ออปเจคคอนแทรกก็จะส่งสถานะของสัญญานั้นก็กลับมา ออปเจคโปรเจคก็จะรู้ว่าหน่วยอาศัยนั้นถูกเช่าอยู่หรือเปล่า ถ้าถูกเช่าอยู่ก็แสดงว่าถึงกำหนดชำระ ก็จะสอบถามราคาเช่าต่อไป ซึ่งออปเจคคอนแทรกก็จะส่งค่าเช่าซึ่งเป็นสิ่งที่ตกลงกันไว้ในสัญญากลับมาให้ ดังนั้นท้ายที่สุดแล้วออปเจคโปรเจคก็จะสามารถคำนวณได้ว่า ภายในโครงการของคนนั้น มีกี่หน่วยอาศัยที่ถูกเช่าอยู่ และเป็นเงินค่าเช่ารวมกันเท่าใด ส่งข้อมูลทั้งหมดคืนไปให้เจ้าหน้าที่ที่สอบถาม



รูปที่ 3.4 ซีควเอนซ์ไดอะแกรมของยูสเคส รายละเอียดสัญญา

จากรูปที่ 3.4 จะแสดงยูสเคสที่สองคือ ซีควเอนซ์ไดอะแกรมของยูสเคสรายละเอียดสัญญา ซึ่งมีกิจกรรมที่เกิดขึ้นคือ เจ้าหน้าที่จะสอบถามไปที่ออปเจคคัสทอเมอร์เพื่อขอข้อมูลลูกค้า และทราบว่าต้องการรายละเอียดสัญญาของลูกค้าคนไหน จากนั้นก็จะสอบถามไปที่ออปเจคคอนแทรกเพื่อขอรายละเอียดสัญญาของลูกค้าคนนั้น ออปเจคคอนแทรกก็จะถามไปยังออปเจคเรนท์คอนแทรก ซึ่งเป็นออปเจคที่เฉพาะลงไปอีก ถึงรายละเอียดสัญญาเพิ่มเติม ซึ่งเป็นลักษณะเฉพาะของสัญญาเช่า ซึ่งกรณีนี้ก็ข้อมูลของการประกันการเช่า เมื่อออปเจคคอนแทรกได้ข้อมูลทั้งหมดแล้วซึ่งรวบรวมได้ทั้งจากออปเจคตัวเองและออปเจคเรนท์คอนแทรก ก็ส่งข้อมูลคืนไปให้เจ้าหน้าที่

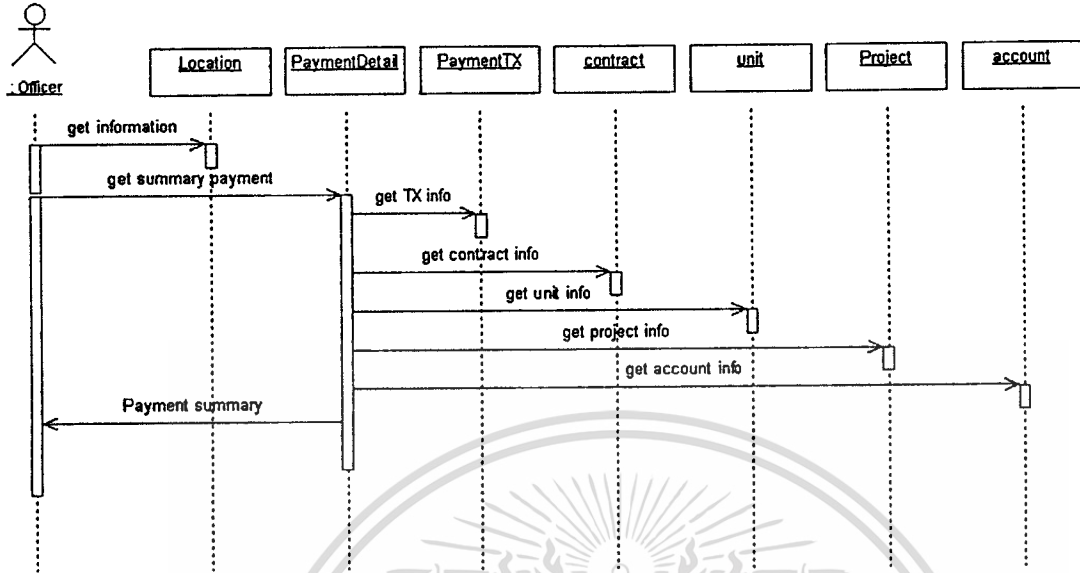
ลักษณะความสัมพันธ์ระหว่างออปเจคคอนแทรก และออปเจคเรนท์คอนแทรคนั้นจะเป็นความสัมพันธ์แบบคลาสย่อย ซึ่งเป็นการคลาสที่ให้รายละเอียดเฉพาะลงไปอีก ดังนั้นในการออกแบบจึงเป็นคลาสที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และคีย์ของคลาสแม่ก็จะเป็นฟอร์เรนจ์คีย์และคีย์ของคลาสลูกด้วยเช่นกัน



รูปที่ 3.5 ซีควเอนซ์ไดอะแกรมของยูสเคสหน่วยอาศัยที่ว่าง

จากรูปที่ 3.5 ซึ่งเป็นซีควเอนซ์ไดอะแกรมของยูสเคสข้อมูลหน่วยอาศัยที่ยังว่างนั้น เป็นยูสเคสที่เจ้าหน้าที่ต้องการร้องขอ เพื่อจะทราบว่า มีหน่วยอาศัยใดบ้างที่ยังว่างอยู่ เพื่อให้ลูกค้าใหม่สามารถเลือกได้ตามต้องการ โดยกิจกรรมจะเริ่มต้น โดยการออกแบบนั้นตั้งใจจะให้ผู้ใช้งานสามารถเลือกค้นข้อมูลได้ตามโครงการ และแบบของหน่วยอาศัยที่ต้องการ ดังนั้นในซีควเอนซ์ไดอะแกรมนั้นเจ้าหน้าที่จะต้องส่งข้อความไปสอบถามข้อมูลโครงการ จากออปเจกโพรเจกก่อน และส่งข้อความไปสอบถามข้อมูลแบบของหน่วยอาศัยจากออปเจกยูนิต เมื่อตัดสินใจได้แล้วว่าต้องการจะดูข้อมูลของโครงการใด และแบบใด ก็จะส่งข้อความไปที่ออปเจก หน่วยอาศัยว่าต้องการข้อมูลหน่วยอาศัยที่ว่างที่อยู่ในโครงการ และแบบของหน่วยอาศัยที่ต้องการ

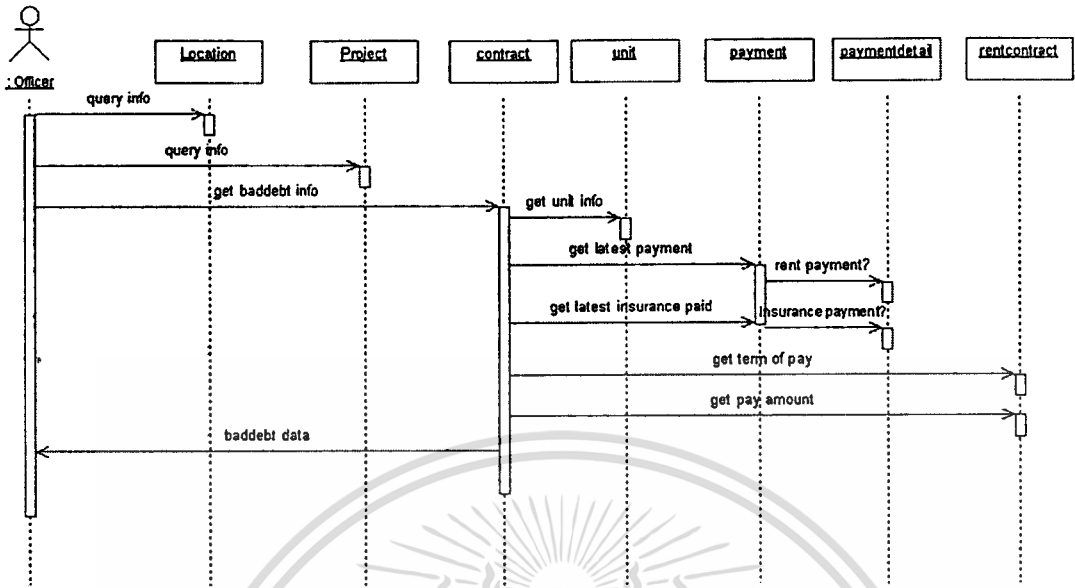
ออปเจกยูนิต เมื่อได้รับการร้องขอดังนั้น ก็จะไปค้นหาภายในตัวเองว่า ภายในหน่วยอาศัยทั้งหมดนั้น มีหน่วยอาศัยใดบ้างที่อยู่ในโครงการที่ต้องการ เมื่อได้รายการแล้ว ก็จะไปสอบถามที่ออปเจกโมเดล ว่า หน่วยอาศัยนั้น มีโมเดลเป็นแบบใด และเมื่อได้โมเดลแล้ว ก็จะส่งคำร้องไปสอบถามออปเจกชนิดหน่วยอาศัยต่อไปว่า โมเดลที่ได้นั้นเป็นชนิดแบบใด และเทียบว่าตรงกับที่เจ้าหน้าที่ต้องการหรือไม่ เมื่อได้ข้อมูลตามที่ผู้ใช้ต้องการแล้ว ก็จะส่งรายการทั้งหมดที่เรียกค้นได้คืนเป็นข้อมูลไปให้ผู้ใช้



รูปที่ 3.6 ซีควเอนซ์ไดอะแกรมของยูสเคสการชำระเงินประจำวัน

จากรูปที่ 3.6 ซึ่งเป็นซีควเอนซ์ที่ค่อนข้างยาว เป็นกิจกรรมของยูสเคสการชำระเงินประจำวัน ซึ่งวัตถุประสงค์ก็เพื่อให้เจ้าหน้าที่สามารถตรวจสอบข้อมูลการรับชำระประจำวัน หรือประจำเดือน ซึ่งสามารถเลือกดูตามเขตของหน่วยงานได้ ดังนั้นขั้นแรกสุดเจ้าหน้าที่ต้องตรวจสอบกับเขตควบคุมก่อนว่าต้องการจะเลือกดูข้อมูลของเขตการควบคุมใด หลังจกนั้นก็ส่งคำร้องไปหาออปเจคเพเมนต์ดีเทล ซึ่งเป็นออปเจคที่เก็บรายการย่อยทั้งหมดเป็นตัว ๆ เช่นค่าน้ำ ค่าไฟ ค่าเช่า ออปเจคนี้จะส่งคำร้องต่อไปหาออปเจคเพเมนต์ ซึ่งจะรวมกลุ่มรายการที่เป็นบิลเดียวกันเป็นกลุ่ม ๆ เพื่อให้ได้ข้อมูลว่าการจ่ายเงินนั้น ๆ เป็นของสัญญาใด ซึ่งเมื่อได้ข้อมูลการจ่ายแล้ว สิ่งที่จะต้องทำต่อไปก็คือพยายามเทียบว่า การจ่ายเงินนั้นสังกัดอยู่ในเขตควบคุมใด

ซึ่งความพยายามจะหาว่าการจ่ายเงินนั้นสังกัดอยู่ในเขตควบคุมใต้นั้นจำเป็นต้องสอบถามต่อไปหลาย ๆ ทอด ซึ่งก็คือ ต้องสอบถามไปที่ออปเจคคอนแทรค ว่าสัญญานั้นเป็นของหน่วยอาศัยใด และสอบถามไปที่ออปเจคยูนิทว่า หน่วยอาศัยนั้นอยู่ในโครงการใด และสุดท้ายสอบถามไปที่ออปเจคโครงการว่าโครงการนั้นอยู่ในเขตปกครองใด เมื่อทราบแล้วว่าสังกัดหน่วยควบคุมใด ก็จะนำไปเทียบกับสิ่งที่ผู้ใช้เลือกว่า ไซเขตปกครองที่ผู้ใช้ต้องการหรือไม่ ท้ายที่สุดออปเจคเพเมนต์ดีเทลก็จะรวบรวมข้อมูลตามที่ใช้ต้องการส่งเป็นข้อมูลกลับไปให้ผู้ใช้



รูปที่ 3.7 ซีเควนซ์ไดอะแกรมของยูสเคสหนี้เสีย

จากรูปที่ 3.7 ซึ่งเป็นซีเควนซ์ไดอะแกรมของยูสเคสหนี้เสีย เป็นยูสเคสที่ตั้งใจให้เจ้าหน้าที่สามารถตรวจสอบได้ว่า ถึงงวดการชำระเงินปัจจุบันนั้น มีหนี้เสียเท่าใดบ้าง ก็หน่วยอาศัย ซึ่งแบ่งตาม เขตควบคุมและตามโครงการ ดังนั้นในซีเควนซ์ไดอะแกรม ก็จะมีลักษณะเช่นเดิม คือเจ้าหน้าที่จะทำการสืบค้นไปที่ออฟเจคโลเคชั่น และออฟเจคโปรเจคเพื่อหา เขตควบคุม และโครงการที่ต้องการหาข้อมูล จากนั้นก็จะส่งคำร้อง ไปหาออฟเจคคอนแทรกเพื่อขอข้อมูลหนี้เสีย ดังนั้นในกรณีนี้ออฟเจคคอนแทรกก็จะทำหน้าที่เป็นผู้ประสานงานกับออฟเจคอื่น เพื่อให้ได้มาซึ่งข้อมูลหนี้เสียทั้งหมด ตามเขตและ โครงการที่ต้องการ

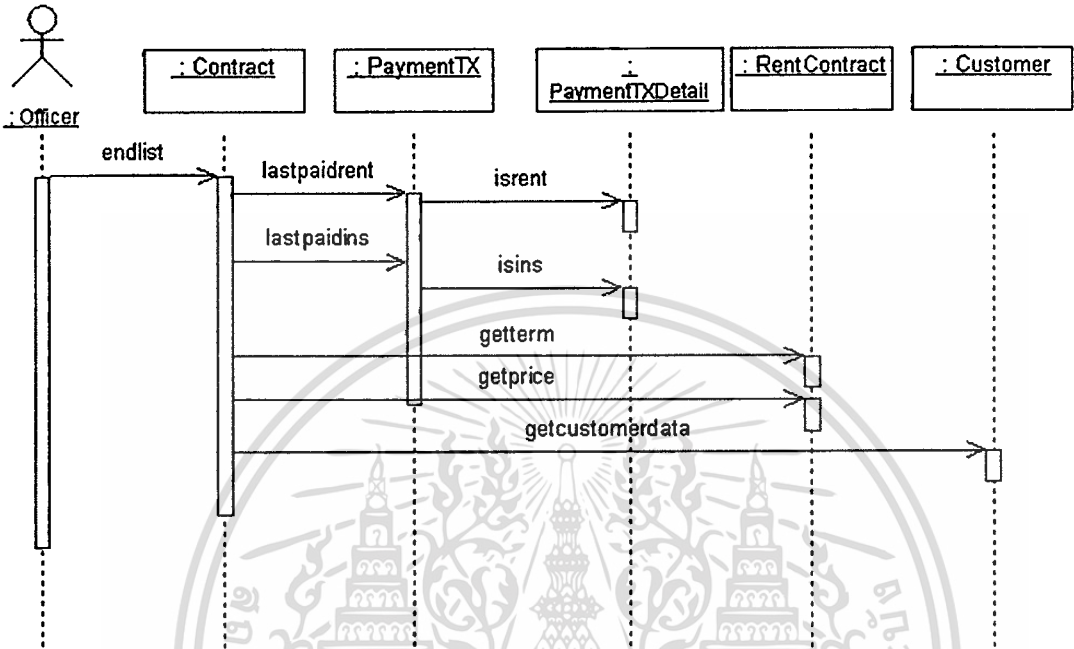
เมื่อออฟเจคคอนแทรกได้รับคำร้องแล้ว ก็จะดูก่อนว่า แต่ละสัญญานั้นสังกัดโครงการที่ต้องการหรือไม่ โดยตรวจสอบโครงการที่สังกัดไปที่ออฟเจค ยูนิท เมื่อเป็นโครงการที่ต้องการแล้ว ก็จะดูว่าแต่ละสัญญานั้นเป็นหนี้เสียหรือไม่ โดยตรวจสอบไปที่ออฟเจค เพเมนต์ ว่าการจ่ายค่าเช่าครั้งสุดท้ายนั้นเป็นเมื่อใด ออฟเจคการเพเมนต์ ก็จะตรวจสอบทุก ๆ ทรานแซคชันจากล่าสุดย้อนหลังไปเรื่อย ๆ ของสัญญานั้นว่าเป็นการจ่ายค่าเช่าหรือไม่ โดยสอบถามไปที่ออฟเจค เพเมนต์ทีเทลอีกที ถ้าเป็นการชำระค่าเช่า ก็จะไ้รายการ การชำระล่าสุด

จากนั้นออฟเจคคอนแทรกก็จะสอบถามอีกว่า การชำระค่าประกันการเช่าครั้งล่าสุดนั้นเป็นเมื่อใด ด้วยวิธีการเดียวกัน ก็จะไ้ข้อมูลการชำระค่าประกันล่าสุดมา เมื่อได้ข้อมูลการชำระเงินล่าสุดแล้ว ก็จะสามารถคำนวณได้ว่า เป็นหนี้เสีย หรือไม่ ถ้าเป็นหนี้เสีย ค้างอยู่ที่งวด เป็นจำนวนเงินเท่าใด โดยสามารถตรวจสอบจำนวนงวดของค่าประกัน กับจำนวนค่าประกัน ไปที่ออฟเจคเรนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่หรือใช้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนแทรก ทั้งหมดนี้ก็จะสามารถสร้างชุดข้อมูลหนี้เสียของ โครงการนั้นส่งคืน ไปให้เจ้าหน้าที่ผู้ สอบถามได้



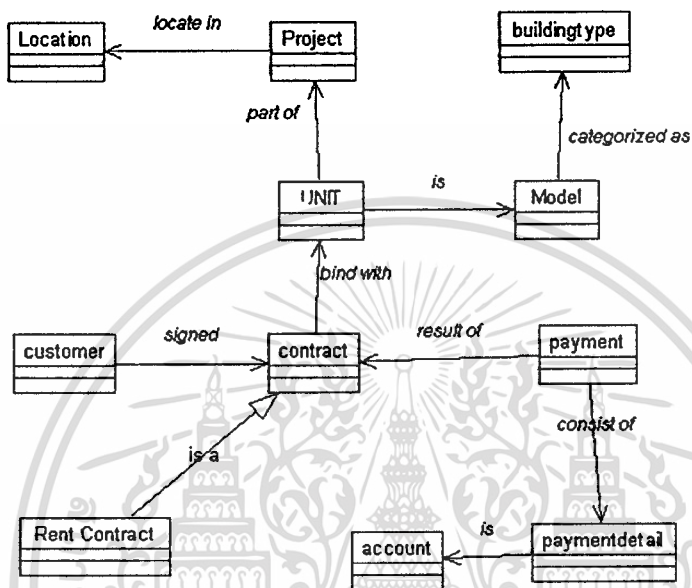
รูปที่ 3.8 ซีเควนซ์ไดอะแกรมของยูสเคสลูกหนี้หมดอายุสัญญา

ต่อไปรูปที่ 3.8 จะเป็นซีเควนซ์ไดอะแกรมของยูสเคสลูกหนี้หมดอายุสัญญาซึ่งเป็นยูสเคสสุดท้ายของระบบ โดยวัตถุประสงค์ของยูสเคสนี้ก็คือต้องการให้เจ้าหน้าที่ที่สามารถตรวจสอบได้ว่าในงวดการชำระเงินปัจจุบันจะมีลูกหนี้รายใดเป็นจะต้องชำระเงินเป็นครั้งสุดท้ายบ้าง ก็จะหมดสัญญาในงวดต่อไป เพื่อจะได้สามารถทำการแจ้ง หรือจัดการกับหนี้ค้างชำระให้เสร็จสิ้น ดังนั้นลักษณะการสื่อสารก็คือ เจ้าหน้าที่ที่จะส่งคำร้องไปยังออปเจคคอนแทรก เพื่อขอให้สร้างชุดข้อมูลให้ โดยออปเจคคอนแทรกจะเป็นผู้ประสานงานในกรณีนี้ ดังนั้นในขั้นแรกออปเจคคอนแทรกจะดูข้อมูลของตัวเองก็จะทราบได้ว่ามีสัญญาใดบ้างที่จะหมดอายุในงวดเดือนนี้ และก็จะถามไปยังออปเจคเพเมนต์ เพื่อขงงวดสุดท้ายที่จ่ายค่าเช่าและค่าประกัน เพื่อจะทราบได้ว่าสัญญานั้นมีหนี้ค้างจ่ายหรือไม่ ออปเจคเพเมนต์ก็จะมองหาการจ่ายเงินครั้งสุดท้ายของสัญญานั้น และก็จะถามไปยังออปเจคเพเมนต์ดีเทลเพื่อถามว่าการจ่ายเงินครั้งนั้นมีค่าเช่า และค่าประกันหรือไม่ ก็จะทราบการจ่ายครั้งสุดท้าย จากนั้นออปเจคคอนแทรกก็จะสอบถามไปยังออปเจคเรนท์คอนแทรกเพื่อสอบถามค่าประกัน และงวดที่ต้องจ่ายค่าประกัน เพื่อประกอบการคำนวณหนี้ค้างชำระ จากนั้นก็จะสอบถามไปยังออปเจคคัสทอเมอร์เพื่อขอข้อมูลลูกค้า ดังนั้นออปเจคคอนแทรกก็จะได้รับข้อมูลครบทั้งหมดสร้างเป็นข้อมูลคืนไปให้เจ้าหน้าที่ดังประสงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้รายละเอียดกิจกรรมที่เกิดขึ้นทั้งหมดภายในระบบแล้ว ถึงขั้นนี้ก็จะได้ภาพคร่าว ๆ ว่าระบบนี้จะต้องประกอบด้วยคลาสอะไรบ้าง และภายในคลาสนั้นจะต้องมีบริการอะไรให้บ้าง ซึ่งสามารถวิเคราะห์ได้ออกมาดังภาพที่ 3.9



รูปที่ 3.9 คลาสไดอะแกรมที่คัดออกมาจากซีควเอนซ์ไดอะแกรมระดับการวิเคราะห์ระบบ

จากรูปที่ 3.9 จะเป็นคลาสไดอะแกรมขั้นแรกที่ได้จากการดึงเอาอุปเจดที่มีกิจกรรมออกมาจากซีควเอนซ์ไดอะแกรม คลาสไดอะแกรมจะแสดงความสัมพันธ์ระหว่างกันของเหล่าคลาสด้วย ซึ่งคลาสเหล่านี้จะเป็นแกนหลัก ให้ผู้ออกแบบสามารถใช้เป็นจุดเริ่มต้นในการออกแบบระบบที่มีรายละเอียด และสร้างเป็นระบบต่อไป

จะสังเกตความสัมพันธ์ได้ว่ามีเพียงสองชนิดเท่านั้น คือความสัมพันธ์แบบการสื่อสารธรรมดา แสดงความเกี่ยวข้องกันเกี่ยวกับการแลกเปลี่ยนข้อมูลซึ่งกันและกัน และความสัมพันธ์แบบที่สองคือความสัมพันธ์แบบเฉพาะเจาะจงกับคลาสทั่วไป ซึ่งเป็นการแสดงถึงคลาทย่อยที่ให้รายละเอียดในแต่ละกรณีไป และแต่ละคลาสนั้นจะมีหน้าที่เพียงบทบาทเดียว คือเป็นตัวแทนของสิ่งใดสิ่งหนึ่งในระบบไป จะไม่มีคลาสที่ออกแบบมาเพื่อกิจกรรมเฉพาะกิจ ดังนั้นจะสังเกตได้ว่าคลาสไดอะแกรมนี้จะมีลักษณะคล้ายกับอ็อบเจกต์ไดอะแกรมในระดับหนึ่ง เว้นแต่ว่าอ็อบเจกต์ไดอะแกรมไม่มีความสัมพันธ์ที่หลากหลายเท่ากับคลาสไดอะแกรม

3.3 การออกแบบระบบ

สำหรับการออกแบบระบบนั้นจะเริ่มต้นจากการนำเอาคลาสโคอะแกรมเบื้องต้นที่ได้จากการวิเคราะห์ระบบมาเป็นต้นแบบสำหรับออกแบบ และให้รายละเอียด เช่นความสัมพันธ์ระหว่างคลาส แอทริบิว และบริการที่เหล่าคลาสจะมีให้แก่ระบบ

ในความเป็นจริงแล้ว คลาสโคอะแกรมที่ได้มานั้น สามารถสะท้อนความต้องการของระบบได้เป็นอย่างดี ดังนั้นงานการออกแบบซึ่งก็คือ การทบทวนและปรับปรุงเพื่อให้สนองความต้องการได้ดียิ่งขึ้นนั้น จึงมีสิ่งต้องทำเพียงเล็กน้อย ซึ่งจากที่ออกแบบไว้ สิ่งที่ต้องทำเพิ่มก็คือ การเพิ่มส่วนอธิบายรายละเอียดของโมเดลที่เป็นแบบของหน่วยอาศัยว่ามีส่วนประกอบอย่างไรบ้าง เช่นจำนวนห้องนอน จำนวนห้องน้ำ และการเพิ่มส่วนอธิบายการใช้จ่ายว่าค่าใช้จ่ายต่าง ๆ นั้น เป็นอะไรบ้าง เป็นรหัสบัญชีค่าใช้จ่ายเท่าไรเท่านั้น

นอกจากนั้นจะมีคลาสพิเศษที่เพิ่มขึ้นมาคือคลาสของราคา เนื่องจากลักษณะพิเศษของระบบงานของการเคหะ คือ หน่วยอาศัยแต่ละโมเดล จะมีราคาต่างกัน ไป และโมเดลเดียวกัน หากแต่อยู่คนละโครงการซึ่งก็คือคนละทำเล ก็จะมีราคาไม่เท่ากัน ดังนั้นคลาสของราคาจึงเป็นคลาสพิเศษที่ช่วยมาให้คำอธิบายกับปัญหานี้ โดยจะเป็นแอตโซซิเอตคลาส ระหว่าง คลาสโมเดล และ คลาสโครงการ

ส่วนอื่น ๆ ก็เพียงให้รายละเอียดของแอทริบิว ว่าแต่ละคลาสนั้นต้องการแอทริบิวอะไรบ้าง ซึ่งมีรายละเอียดดังต่อไปนี้

คลาสเขตควบคุมจะมีรายการแอทริบิวคือ รหัสของเขตควบคุม ชื่อเขตควบคุม ถนนที่ตั้งเลขหมู่บ้านของที่ตั้ง ซอย แขวง เขต จังหวัด และรหัสไปรษณีย์

คลาสโครงการ มีแอทริบิวคือ รหัสโครงการ และชื่อโครงการ คลาสราคาที่เป็นแอตโซซิเอตคลาสดังที่กล่าวมาแล้ว จะมีแอทริบิวคือ พื้นที่ใช้สอย ราคา และอัตราดอกเบี้ยที่จะใช้ในกรณีซื้อผ่อน ต่อมาคือคลาสหน่วยอาศัย มีแอทริบิวคือ รหัสหน่วยอาศัย ซึ่งก็คือบ้านเลขที่ รหัสตึกหนึ่งซึ่งเป็นรหัสพิเศษที่ใช้อ้างอิงกันภายในหน่วยงานของการเคหะแห่งชาติ ตำแหน่งชั้นที่อยู่ และเลขที่อาคารที่อยู่

ต่อมาคือคลาสโมเดล มีแอทริบิวคือ รหัสโมเดล และชื่อโมเดล คลาสชนิดหน่วยอาศัยจะมีแอทริบิวคือ รหัสของชนิดหน่วยอาศัย และชื่อของชนิด เช่นบ้านเดี่ยว ห้องเดี่ยว เป็นต้น

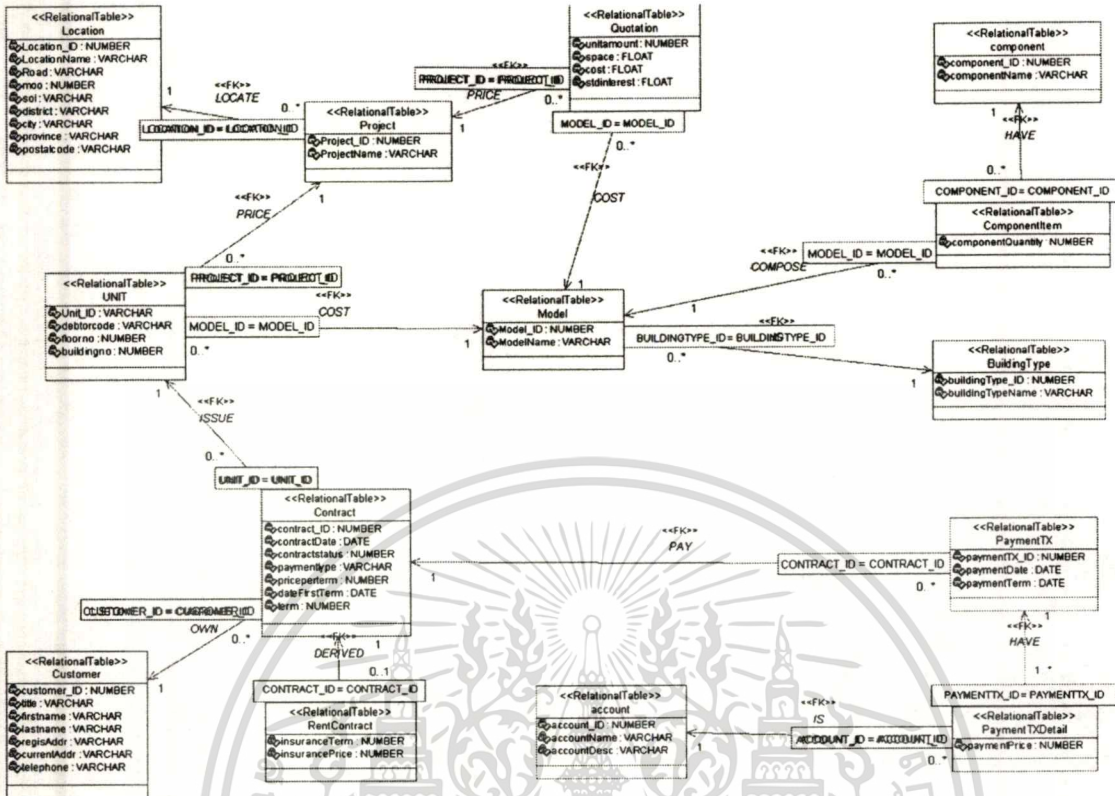
นอกจากนั้นส่วนที่เพิ่มเติมเข้ามาในขั้นการออกแบบก็คือ ส่วนอธิบายคุณลักษณะของโมเดลว่ามีส่วนประกอบอย่างไร จะทำโดยคลาสคอมโพเนนท์ไอเทม (Component Item) ซึ่งมีแอทริบิวคือ ปริมาณของส่วนประกอบนั้น และคลาสคอมโพเนนท์ (Component) มีแอทริบิวคือ รหัสส่วนประกอบ และชื่อส่วนประกอบ

ต่อมาเป็นคลาสที่สำคัญคือคลาส สัญญา มีแอทริบิวคือ รหัสสัญญา วันที่ทำสัญญา สถานะของสัญญา ชนิดการชำระเงิน ราคาคงงวด แรกของสัญญา และจำนวนงวด และคลาสสัญญาเช่า ซึ่งเป็นคลาสสัญญาที่เจาะจงลงไปอีก เกิดจากความสัมพันธ์แบบคลาสเจาะจงกับคลาสทั่วไป (Specific / Generalize) มีแอทริบิว อธิบายลักษณะพิเศษที่มีเฉพาะในสัญญาเช่า ก็คือข้อมูลการประกันการเช่า คือจำนวนงวดที่ต้องผ่อนค่าประกันการเช่า และค่างวดต่อการผ่อนหนึ่งครั้ง ต่อมาเป็นคลาสของลูกค้า มีแอทริบิวคือ รหัสลูกค้า คำนำหน้าชื่อ ชื่อ นามสกุล ที่อยู่ตามทะเบียน ที่อยู่จริง และเบอร์โทรศัพท์

คลาสต่อไปเป็นคลาสที่มีความสำคัญอีกเช่นกัน คือคลาสการชำระเงิน ซึ่งแต่ละข้อมูลในคลาสนี้ จะหมายถึงการชำระเงินแต่ละครั้ง ทำให้มีแอทริบิวคือ รหัสทรานแซกชัน วันที่จ่าย และงวดที่จ่าย ซึ่งแต่ละทรานแซกชันนี้อาจจะเป็นไปได้ที่จ่ายหลาย ๆ อย่างพร้อมกัน เช่น ในการจ่ายเงินหนึ่งครั้ง อาจจะมีการจ่ายค่าเช่า ค่าประกัน ค่าน้ำ และค่าไฟ ดังนั้น แต่ละรายการจะถูกอธิบายด้วยคลาสต่อมาคือ คลาสรายละเอียดการชำระเงิน ซึ่งมีแอทริบิวคือ ราคาที่จ่ายและรหัสค่าใช้จ่าย : ซึ่งถูกอธิบายโดยอีกคลาส คือคลาสรหัสบัญชีค่าใช้จ่าย มีแอทริบิวคือ รหัสบัญชีค่าใช้จ่าย ชื่อบัญชี และคำอธิบายของบัญชีนั้น ๆ

ทั้งหมดก็เป็นรายละเอียดของคลาสทั้งหมดที่มีในระบบนี้ เนื่องจากระบบนี้ถูกวางแผนให้จะต้องสร้างบนระบบฐานข้อมูลของโอราเคิล และสร้างระบบติดต่อกับผู้ใช้โดยอาศัยเครื่องมือคือ เดเวลลอปเปอร์ 6 ทำให้ต้องออกแบบระบบเป็นฐานข้อมูลแบบเชิงสัมพันธ์ ตามหลักการออกแบบจะสามารถแปรจากหนึ่งคลาสได้เป็นหนึ่งตารางพอดี ดังนั้นคลาสที่ออกแบบจึงต้องเตรียมโครงสร้างให้กับการแปลงชนิดนี้ด้วย ดังนั้นโดยทั่วไป คลาสก็จะมีรหัสของคลาสทำหน้าที่เป็นไพรมารีคีย์ของตารางนั้น และทุก ๆ ความสัมพันธ์ก็จะถูกสร้างโดยการเอาไพรมารีคีย์มาเป็นฟอเรนจ์คีย์ของคลาส นอกจากนั้นแล้วแอทริบิวอื่น ๆ ของคลาสก็จะเป็นเพียงแอทริบิวที่ให้คำอธิบายลักษณะของแต่ละข้อมูลเท่านั้น มิได้มีความสำคัญในลักษณะโครงสร้างความสัมพันธ์ใด ๆ

ทำให้ท้ายที่สุดแล้ว สามารถได้คลาสไดอะแกรมสุดท้ายที่ผ่านการแก้ไขในการปรับโครงสร้างดังรูปที่ 3.10 ซึ่งประกอบไปด้วยส่วนประกอบที่สำคัญ เช่นการบรรยายลักษณะความสัมพันธ์ว่ามีความสัมพันธ์เป็นชนิดใด และมีลักษณะความสัมพันธ์ระหว่างข้อมูลเป็นอย่างไร เช่นหนึ่งข้อมูลแต่หลายข้อมูล เป็นต้น มีแอทริบิวของคลาสแต่ละคลาสทั้งหมด มีความสัมพันธ์และแอทริบิวที่เป็นตัวเชื่อมความสัมพันธ์ รวมไปถึงชนิดของข้อมูลที่แต่ละแอทริบิวนั้นเป็น ทำให้เป็นการลงรายละเอียดที่ขึ้นกับเครื่องมือที่ใช้ในการพัฒนาด้วย



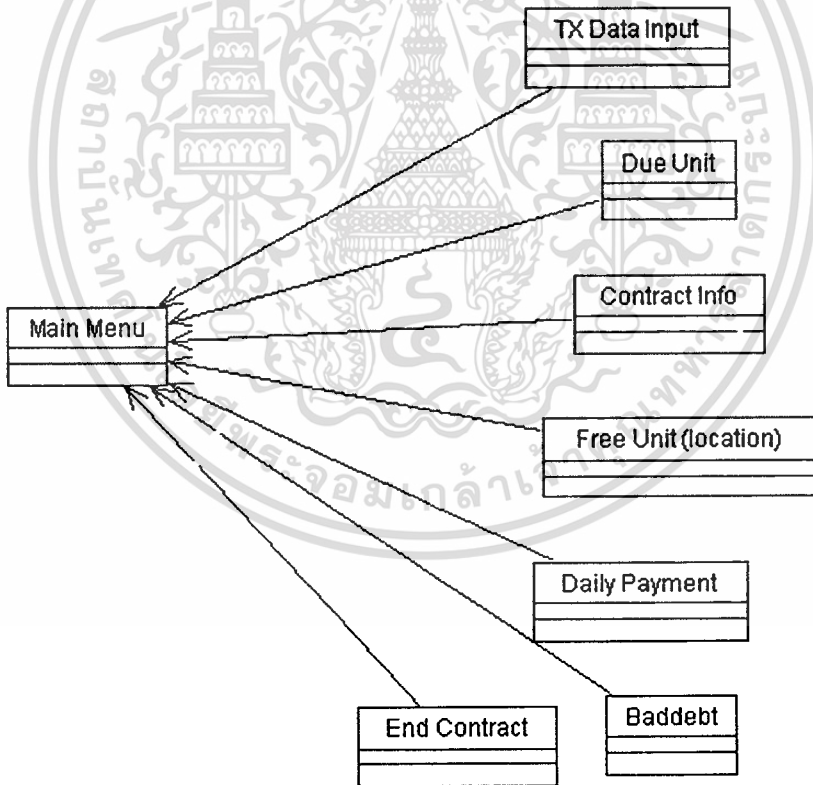
รูปที่ 3.10 คลาสไดอะแกรมระดับออกแบบสุดท้าย (ส่วนธุรกิจ) แสดงเฉพาะแอทริบิวต์

รูปที่ 3.10 จะเป็นคลาสไดอะแกรมสุดท้ายของส่วนธุรกิจ ที่สามารถสร้างคำสั่ง SQL เพื่อนำไปสร้าง Schema ให้กับระบบฐานข้อมูลได้ทันที ดังนั้นแต่ละแอทริบิวต์นั้น ก็ต้องได้รับการออกแบบชนิดของข้อมูลที่ใช้ด้วย โดยจำเป็นต้องเป็นชนิดข้อมูลพื้นฐานข้อมูลโอราเคิลรองรับ และแต่ละคลาสขณะนี้ยังไม่มีบริการที่ชัดเจน (method) เนื่องจากต้องได้รับการออกแบบในลำดับต่อไป

ต่อไปจะเป็นการออกแบบทางด้านไดนามิกส์คือกิจกรรมที่เกิดขึ้นจริง โดยต้องอ้างอิงตามยูสเคส และซีเควนซ์ไดอะแกรมระดับวิเคราะห์ที่ได้แสดงไว้แล้ว แต่จะมีการให้รายละเอียดที่ลงถึงระดับที่สามารถพัฒนาไปใช้งานได้จริง ดังนั้นซีเควนซ์ไดอะแกรม หรือกิจกรรมที่เกิดขึ้นนั้นจำเป็นต้องสอดคล้องกับกิจกรรมจริงที่เกิดขึ้นนั้นหมายความว่า สิ่งแรกที่จะต้องได้รับการออกแบบก่อนคือส่วนติดต่อกับผู้ใช้ จากนั้นจึงเป็นการให้รายละเอียดส่วนติดต่อกับผู้ใช้ ก่อนจะนำไปสู่ซีเควนซ์ไดอะแกรมของแต่ละหน้าต่างที่ติดต่อกับผู้ใช้ว่าจำเป็นต้องส่งข้อมูลอะไรระหว่างกันบ้าง และจำเป็นต้องใช้ข้อมูลอะไรบ้าง จำเป็นจะต้องไปเรียกจากใคร และข้อมูลที่ได้รับกลับมาเพื่อจะแสดงผลเป็นอะไร

3.4 การออกแบบส่วนติดต่อกับผู้ใช้

ตามหลักการออกแบบส่วนติดต่อกับผู้ใช้นั้น ทางทฤษฎีกำหนดไว้ว่าจะต้องมีการออกแบบสองระดับ คือระดับมหภาค จะดูภาพกว้างว่า มีส่วนใดบ้างที่ต้องติดต่อกับผู้ใช้ และระดับจุลภาค คือการให้รายละเอียดแต่ละส่วนที่ทางระดับมหภาคได้ค้นพบนั้นควรมีรายละเอียดอย่างไรบ้าง และต้องติดต่อกันเองอย่างไร ติดต่อกับคลาสที่เป็นแกนหลักทางธุรกิจอย่างไร ซึ่งจากการออกแบบนั้น ทำให้ได้ภาพคร่าว ๆ คือ ระบบนี้จะมีด้วยกัน 5 ยูสเคสซึ่งนั่นหมายถึง 5 บริการที่มีให้ดังนั้นจึงมีส่วนติดต่อกับผู้ใช้หลัก ๆ 5 ส่วน และส่วนที่เป็นส่วนเชื่อมโยงกัน ทำหน้าที่เป็นเมนูหลักอีกหนึ่งส่วน เพิ่มส่วนพิเศษอีกส่วนที่อยู่นอกเหนือความต้องการของระบบ แต่ก็มีความจำเป็นคือ ส่วนที่ทำหน้าที่นำข้อมูลเข้า รวมทั้งสิ้นมีด้วยกัน 7 ส่วนติดต่อ ซึ่งสามารถสร้างคลาสไคอะแกรมของส่วนติดต่อได้ดังต่อไปนี้



รูปที่ 3.11 คลาสส่วนติดต่อกับผู้ใช้

รูปที่ 3.11 จะเป็นคลาสไดอะแกรมส่วนติดต่อกับผู้ใช้ที่ได้จากการออกแบบระดับมหภาค ซึ่งโดยทั่วไป หนึ่งคลาสก็คือหน้าตาต่างนั่นเอง แต่ละคลาสก็จะมีเอทริบิวเป็นคุณสมบัติของหน้าตาต่าง เช่น ขนาด สี หรือชื่อ ส่วนประกอบภายใน ซึ่งมีจำเป็นต้องอธิบาย ณ ที่นี้ เนื่องจากเป็นคุณสมบัติเฉพาะของแต่ละเครื่องมือพัฒนาระบบ ส่วนบริการที่ให้หรือเมธอร์ค่านั้นก็จะเป็นส่วนบริการที่รองรับเมื่อเกิดเหตุการณ์ขึ้น เช่นเมื่อผู้ใช้กดปุ่ม หรือเลือกตัวเลือก เป็นต้น

ต่อไปจะเป็นการออกแบบระดับจุลภาค คือการให้รายละเอียดว่าแต่ละส่วนติดต่อกับผู้ใช้นั้นจะต้องมีรายละเอียดอย่างไรบ้าง ซึ่งสามารถแสดงออกมาได้ในรูปของหน้าตาส่วนติดต่อกับผู้ใช้ที่ออกแบบไว้ ซึ่งส่วนติดต่อกับผู้ใช้ที่ออกแบบไว้ก็จะมีผลย้อนกลับมาเปลี่ยนแปลงคลาสไดอะแกรมของส่วนติดต่อกับผู้ใช้ที่อื่นที่เช่นกัน

การออกแบบส่วนติดต่อกับผู้ใช้ที่เป็นเมนูหลัก นั้นผลจากการออกแบบให้หน้าตาคร่าว ๆ มีส่วนประกอบดังภาพที่ 3.12

รูปที่ 3.12 ส่วนติดต่อกับผู้ใช้ที่เป็นเมนูหลัก

จากรูปที่ 3.12 จะเห็นว่าส่วนติดต่อกับผู้ใช้แรกนั้น จะเป็นส่วนที่ทำหน้าที่เป็นศูนย์กลางการควบคุม เพื่อจะเรียกใช้ระบบย่อยอื่นต่อไป ทำให้มีความสัมพันธ์กับอินเตอร์เฟสคลาสอื่น เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปดังรูปที่ 3.11 คือสัมพันธ์กับทุกอินเทอร์เฟซคลาสอื่น แต่ไม่มีความสัมพันธ์กับคลาสทางระบบงานจริง หรือคลาสทางธุรกิจเลย ลักษณะการทำงานคือจะมีส่วนที่บอกว่า ปัจจุบันผู้ใช้ใด กำลังเชื่อมต่อกับฐานข้อมูลใด และผู้ใช้สามารถเลือกที่จะติดต่อกับฐานข้อมูลในชื่อผู้ใช้ใหม่ได้

นอกจากนั้นก็จะเป็นการกดปุ่มเพื่อเรียกระบบงานย่อยที่ต้องการขึ้นมาทำงาน ซึ่งในระบบนี้จะมีระบบงานย่อยทั้งหมด 7 ระบบงานด้วย เพิ่มเติมจากยูสเคสที่มี 6 ยูสเคส และสุดท้ายคือมีปุ่มที่ให้ผู้ใช้งานสามารถเลือกที่จะออกจากระบบได้

ต่อไปจะเป็นส่วนติดต่อกับผู้ใช้เพื่อใส่ข้อมูลการชำระเงินค่าเช่าเข้าสู่ระบบ ซึ่งระบบงานนี้เป็นส่วนเพิ่มเติมจากระบบงานจริง เพื่อให้สามารถมีข้อมูลเข้ามาประมวลผลรวมได้ จึงจำเป็นต้องมีซึ่งทำให้มีหน้าตาดังรูปที่ 3.13

The screenshot shows a web application window titled "นำข้อมูลการรับชำระเช่า" (Rent Payment Information Entry). The window is divided into several sections:

- เขตควบคุม (Control Area):** Includes fields for "รหัสเขต" (Area Code) with value "1" and "ชื่อเขต" (Area Name) with value "bangkai region". Buttons: "ก่อนหน้า" (Previous), "ต่อไป" (Next).
- ลูกค้า (Customer):** Includes fields for "ชื่อ" (Name) with value "Frijob" and "นามสกุล" (Surname) with value "Capra". Buttons: "ก่อนหน้า" (Previous), "ต่อไป" (Next).
- รายละเอียดสัญญา (Contract Details):** Includes fields for "ค่าเช่า" (Rent) with value "1000" and "เลขที่สัญญา" (Contract No.) with value "1001000". Buttons: "ก่อนหน้า" (Previous), "ต่อไป" (Next).
- ข้อมูลการชำระเงิน (Payment Information):** Includes fields for "วันที่ชำระเงิน" (Payment Date) with value "01 03 1999", "เลขที่ชำระเงิน" (Payment No.) with value "1", and "งวดที่จ่าย" (Payment Term) with value "03 1999". Buttons: "ก่อนหน้า" (Previous), "ต่อไป" (Next).
- รายการ (List):** A table with columns: "จำนวนเงิน" (Amount), "เลขที่ชำระเงิน" (Payment No.), and "รายการที่ชำระ" (Payment Item).

จำนวนเงิน	เลขที่ชำระเงิน	รายการที่ชำระ
1000	1	rent
21	1	insurance
10	1	water

Buttons at the bottom right: "จัดเก็บข้อมูล" (Save Data), "กลับสู่รายการหลัก" (Return to Main List).

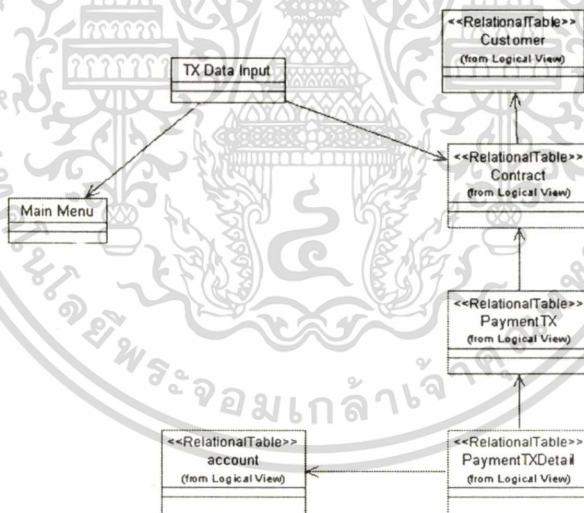
รูปที่ 3.13 ส่วนอินเทอร์เฟซของระบบงานย่อยที่นำข้อมูลการชำระเงินเข้าระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.13 จะเห็นว่าส่วนติดต่อนี้จะเริ่มต้นจากที่ผู้ใช้จะต้องเลือกชื่อลูกค้าที่จะชำระเงินก่อน และถัดไปทางด้านขวา ก็จะเลือกสัญญาที่เป็นของลูกค้าคนนั้น ว่าต้องการจะชำระเงินของสัญญาฉบับใด เพราะลูกค้าหนึ่งรายสามารถจะมีสัญญาได้มากกว่าหนึ่งสัญญา หรือพูดอีกอย่างคือลูกค้าหนึ่งรายสามารถจะเช่าได้มากกว่า 1 หน่วยพักอาศัย

หลังจากนั้น ก็จะใส่ข้อมูลการชำระเงินที่กลางหน้าต่าง ซึ่งประกอบด้วย วันที่ชำระเงิน รหัสบิลที่ชำระเงิน งวดที่จ่าย ด้านล่างสุดก็จะสามารถใส่รายการการชำระเงินได้ว่า เป็นจำนวนเงินเท่าใด และเป็นค่าใช้จ่ายอะไร ดังนั้นทำให้ส่วนติดต่อกับผู้ใช้จำเป็นต้องติดต่อกับหรือมีความสัมพันธ์กับคลาสทางธุรกิจ หลายคลาสด้วยกัน นั่นคือ ลูกค้า สัญญา การชำระเงิน รายละเอียดการชำระเงิน และบัญชี ทำให้มีความสัมพันธ์ดังรูปที่ 3.14

จากรูปก็จะเห็นความสัมพันธ์ระหว่างคลาสที่ทำหน้าที่เป็นอินเทอร์เฟส กับคลาสที่เป็นระบบงานจริงที่มีการติดต่อสื่อสารกันอย่างไร เส้นเชื่อมหมายถึง มีการแลกเปลี่ยนข้อมูลกันเกิดขึ้น เช่นมีการร้องขอข้อมูล ส่งข้อมูลกลับ ส่วนที่ทำหน้าที่แสดงผลก็จะเป็นส่วนอินเทอร์เฟสนั่นเอง



รูปที่ 3.14 คลาสไดอะแกรมส่วนติดต่อกับผู้ใช้เพื่อรับข้อมูลการชำระเงินเข้าระบบ

ต่อไปจะเป็นส่วนติดต่อกับผู้ใช้ของระบบย่อยรายละเอียดของสัญญา ซึ่งมีส่วนติดต่อกับผู้ใช้ที่มีรายละเอียดดังรูป 3.15

WINDOW1

รายละเอียดสัญญา

ลูกค้า			
รหัสลูกค้า	100	ที่อยู่ปัจจุบัน	47/1 Petkasame
สำเนาชื่อ	Mr.	ที่อยู่ตามทะเบียน	64/2 Sukhumwit
ชื่อ	Frijob	โทรศัพท์	3488383
นามสกุล	Capra		
		ก่อนหน้า	ต่อไป

รายละเอียดสัญญา			
รหัสสัญญา	1001000	รหัสหน่วยอาศัย	100001
วันที่ทำสัญญา	01 ม.ค. 1999	ค่าเช่า	1000
สถานะสัญญา	1	จำนวนงวดประกัน	3
วิธีการชำระเงิน	cash	ค่าประกัน	21
งวดแรก	01 ก.พ. 1999		
งวดสุดท้าย	01 มี.ค. 2003		
จำนวนงวด	50	สัญญาก่อนหน้า	สัญญาต่อไป
พิมพ์รายงาน	<input type="checkbox"/> พิมพ์ลูกค้าทั้งหมด	กลับรายการหลัก	

รูปที่ 3.15 ส่วนติดต่อกับผู้ใช้ของระบบย่อยรายละเอียดสัญญา

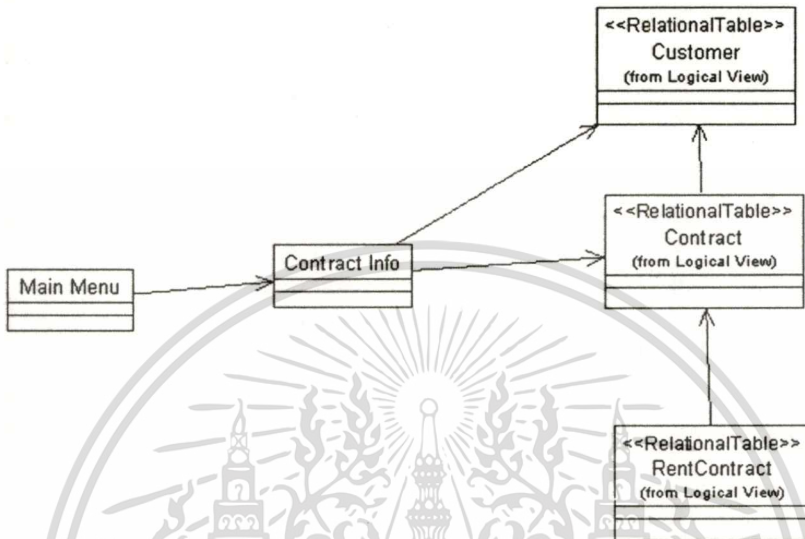
ส่วนนี้จะให้ข้อมูลเกี่ยวกับลูกค้า ซึ่งเป็นข้อมูลส่วนตัวเช่น ชื่อ นามสกุล สถานที่ติดต่อ เบอร์โทรศัพท์ เป็นต้น และด้านล่างก็จะเป็นสัญญาที่ลูกค้าท่านนั้นทำการเช่าแห่งชาติ ซึ่งภายในสัญญาก็จะมีรายละเอียด เช่นวิธีการชำระเงิน จำนวนงวดที่จะต้องจ่าย วันที่งวดเริ่มต้น และวันที่งวดสิ้นสุดสัญญา เป็นต้น โดยการทำงานแล้ว ผู้ใช้จะสามารถเลือกรายชื่อลูกค้าที่ด้านบนก่อน จากนั้นก็จะเลือกสัญญาที่ด้านล่าง โดยสามารถเลือกดูสัญญาต่อไป หรือก่อนหน้าของลูกค้าท่านเดียวกันได้โดยกดที่ปุ่มเพื่อเลือก หลังจากนั้นก็สามารถที่จะพิมพ์ข้อมูลออกเป็นรายงานได้โดยกดที่ปุ่มพิมพ์รายงาน โดยสามารถจะเลือกพิมพ์รายงานสัญญาของลูกค้าทั้งหมดทุกคนได้โดยเลือกที่ช่องเช็คบอซ์ โดยหน้าที่นี้ออกแบบมาเพื่อให้ผู้ใช้เกิดความสะดวกในการพิมพ์รายการสัญญาทั้งหมดที่มี เพื่อจะไม่ต้องเลือกลูกค้าทีละราย และพิมพ์รายงานทีละราย

หลังจากเสร็จสิ้นการทำงานแล้ว สามารถกดปุ่ม กลับรายการหลัก เพื่อกลับสู่หน้าต่างที่ทำงานที่เป็นเมนูควบคุมการทำงานทั้งหมดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนติดต่อกับผู้ใช้นี้จะทำงานสัมพันธ์กับคลาสที่เป็นระบบงานอยู่ไม่กี่คลาสเท่านั้น นั่นคือ คลาสลูกค้า คลาสสัญญา และคลาสสัญญาเช่า ทำให้มีคลาสไดอะแกรมดังรูปที่ 3.16



รูปที่ 3.16 คลาสไดอะแกรมของระบบย่อย รายละเอียดสัญญา

ต่อมาเป็นอินเตอร์เฟซของระบบย่อยหน่วยอาศัยที่ว่าง ซึ่งเป็นอินเตอร์เฟซที่พิเศษกว่าอินเตอร์เฟซอื่น ๆ เพราะจากการวิเคราะห์ระดับจุดภาค ทำให้ออกแบบได้รายละเอียดว่า จะต้องมียระบบหน้าต่างย่อยถึง 3 หน้าต่างด้วยกัน ซึ่งเป็นอินเตอร์เฟซที่ทำให้ผู้ใช้สามารถสอบถามรายละเอียดย่อย ไปเรื่อย ๆ คือ ในหน้าต่างแรก จะเป็นหน้าต่างที่รายงานหน่วยพักอาศัยที่ว่าง โดยจะสรุปเป็นหมวดหมู่ตามเขตควบคุม สามารถสรุปออกมาได้ทีละเขตควบคุมนั้น มีกี่โครงการ และแต่ละโครงการ มีหน่วยพักอาศัยว่างเท่าใด

หน้าต่างต่อมา จะเป็นการเจาะรายละเอียดไปในแต่ละโครงการ ว่าแต่ละโครงการนั้นมีหน่วยอาศัยว่างกี่หน่วย แต่ละหน่วยมีลักษณะประเภทอาคารเป็นอย่างไร และมีตำแหน่งอย่างไร และหน้าต่างสุดท้ายจะเป็นข้อมูลรายละเอียดของแต่ละหน่วยพักอาศัย ว่ามีราคาเท่าใด มีประวัติการเช่าเป็นอย่างไร มีอัตราดอกเบี้ยเป็นอย่างไร และคุณสมบัติอย่างละเอียดของหน่วยอาศัยเป็นอย่างไร ซึ่งชุดของหน้าต่างชุดนี้จะเป็นดังรูป 3.17 ถึงรูปที่ 3.19

WINDOW1

หน่วยอาศัยว่าง

เขตควบคุม

รหัสเขต 2 ตำบล bangrak
 ชื่อเขต North Bangkok อำเภอ bangkorleam
 หมู่ 7 จังหวัด bangkok
 ถนน sukhumvit รหัสไปรษณีย์ 10990
 ซอย nana

ก่อนหน้า ต่อไป

หน่วยอาศัยว่างตามโครงการ

รหัสโครงการ	ชื่อโครงการ	รหัสเขต	หน่วยอาศัยว่าง
2	More Happy	2	9

พิมพ์รายงาน พิมพ์ทุกหน่วยควบคุม รายละเอียด กลับรายการหลัก

รูปที่ 3.17 อินเทอร์เน็ตหน่วยพักอาศัยว่าง ตามเขตควบคุม

WINDOW1

หน่วยอาศัยว่างตามโครงการ

โครงการ

รหัสโครงการ 0 ชื่อโครงการ Happy Village ก่อนหน้า ต่อไป

ชนิดอาคาร

รหัสชนิดอาคาร 1 ชื่อชนิดอาคาร standard house ก่อนหน้า ต่อไป

หน่วยอาศัย

รหัสหน่วยอาศัย	ชนิดอาคาร	ชั้นที่	อาคารที่
100004	standard house	0	0
101002	standard house	0	0
101003	standard house	0	0

พิมพ์รายงาน พิมพ์ทุกโครงการ รายละเอียด กลับรายการก่อนหน้า

รูปที่ 3.18 หน่วยพักอาศัยว่างตามโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WINDOW1

หน่วยอาศัยว่าง - รายละเอียดหน่วยอาศัย

หน่วยอาศัย

รหัสหน่วยอาศัย: 102002 ชั้นที่:

ชื่อร้านค้า: american house ลากรที่:

ชนิดอาคาร: standard house

คุณสมบัติของโมเดล

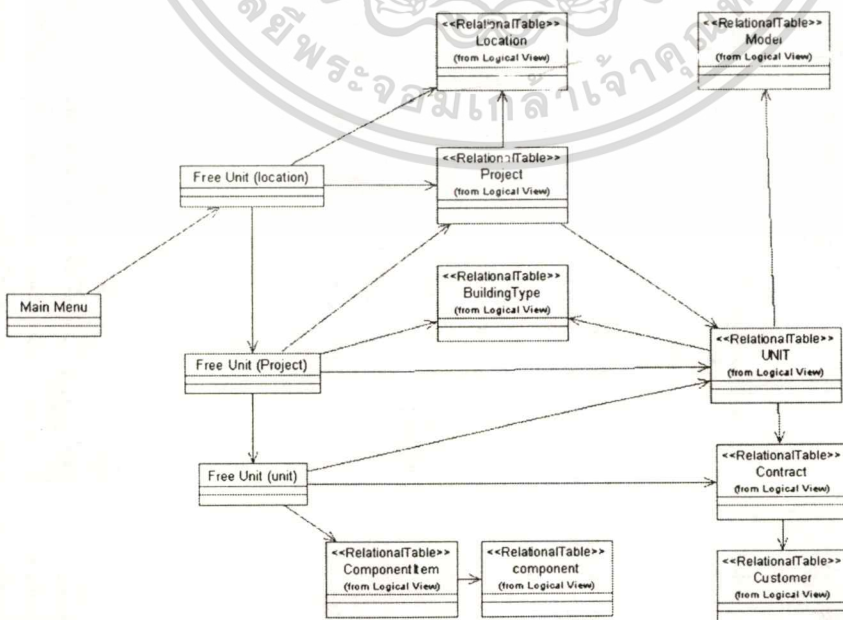
จำนวน	ส่วนประกอบ
4	bedroom
3	bathroom
3	floor
1	restroom

ประวัติสัญญา

เลขที่สัญญา	สถานะ	งวดแรก	งวดสุดท้าย	ท่านำชื่อ	ชื่อลูกค้า	นามสกุล
920100	2	01 มี.ค. 1995	01 เม.ย. 1999	Mr. Frijob	Capra	
901001	2	01 พ.ค. 1996	01 มี.ย. 2000	Mr. Alexandre	Dumas	

รูปที่ 3.19 หน้าต่างระบบย่อยรายละเอียดหน่วยอาศัยที่ว่าง

และรูปที่ 3.20 จะเป็นคลาสไดอะแกรมความสัมพันธ์ของอินเตอร์เฟสระบบย่อยนี้



รูปที่ 3.20 คลาสไดอะแกรมของระบบย่อยหน่วยอาศัยที่ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.17 ถึงรูปที่ 3.19 นั้นจะแสดงให้เห็นการทำงานของระบบย่อยนี้ คือ ที่ระดับแรก ผู้ใช้จะต้องเลือกก่อนว่าต้องการจะดูข้อมูลของเขตควบคุมใด เมื่อเลือกได้แล้วที่ด้านล่างของหน้าต่าง ก็จะแสดงรายละเอียดของแต่ละโครงการ โดยสรุปว่าเหลือที่พักอาศัยเท่าใด

ถ้าผู้ใช้สนใจโครงการใดเป็นพิเศษก็จะเอามาใส่ไปเลือกที่โครงการนั้นและกดปุ่ม แสดงรายละเอียด ซึ่งก็จะเป็นการเปิดไปสู่หน้าต่างต่อไป ซึ่งจะแสดงรายละเอียดของหน่วยพักอาศัยว่างของโครงการนั้นว่า ชนิดอาคารใด เหลือหน่วยพักอาศัยว่าเป็นจำนวนเท่าใด และแต่ละหน่วยมีที่ตั้งอยู่ที่ใด มีบ้านเลขที่เท่าใด

ถ้าผู้ใช้สนใจหน่วยพักอาศัยใดเป็นพิเศษ ก็จะเลือกที่หน่วยพักอาศัยนั้น และกดปุ่ม แสดงรายละเอียดอีกเช่นกัน ซึ่งที่หน้าต่างต่อไป รายละเอียดทุกอย่างเกี่ยวกับหน่วยพักอาศัยนั้นก็แสดงออกมา ไม่ว่าจะเป็นรายละเอียดของแบบโมเดล ส่วนประกอบภายใน รวมไปถึง ประวัติการเช่า ว่าหน่วยพักอาศัยนี้มีผู้ใดบ้างมาทำสัญญา เมื่อไร และเป็นระยะเวลาานเท่าใด

คลาสไดอะแกรมในรูป 3.20 ก็จะแสดงความสัมพันธ์ของระบบย่อยซึ่งค่อนข้างมีความซับซ้อน เพราะมีการเกี่ยวข้องกับข้อมูลหลายออบเจกต์ และมีการสื่อสารหลายขั้นตอน รายละเอียดการสื่อสารที่ดี สามารถดูได้ที่ซีเคเวนซ์ไดอะแกรมในระดับการออกแบบ ซึ่งจะนำเสนอในโอกาสต่อไป ต่อไปจะเป็นระบบย่อย การรับเงินประจำวัน ซึ่งแสดงไว้ในรูปที่ 3.21

การชำระเงินประจำวัน

วันที่: 0 มีนาคม 1999 เฉพาะค่าเช่า

เขตควบคุม: bangkai region

เลขที่ชำระเงิน	วันที่จ่าย	รหัสสัญญา	รหัสหน่วยอาศัย	รายการชำระเงิน	จำนวน	งวดที่จ่าย
1	01 มี.ค. 1999	1001000	100001	rent	1000	มี.ค. 1999
1	01 มี.ค. 1999	1001000	100001	insurance	21	มี.ค. 1999
1	01 มี.ค. 1999	1001000	100001	water	10	มี.ค. 1999
121	01 มี.ค. 1999	100202	100002	rent	1000	มี.ค. 1999
121	01 มี.ค. 1999	100202	100002	insurance	100	มี.ค. 1999

ยอดรวม 3431

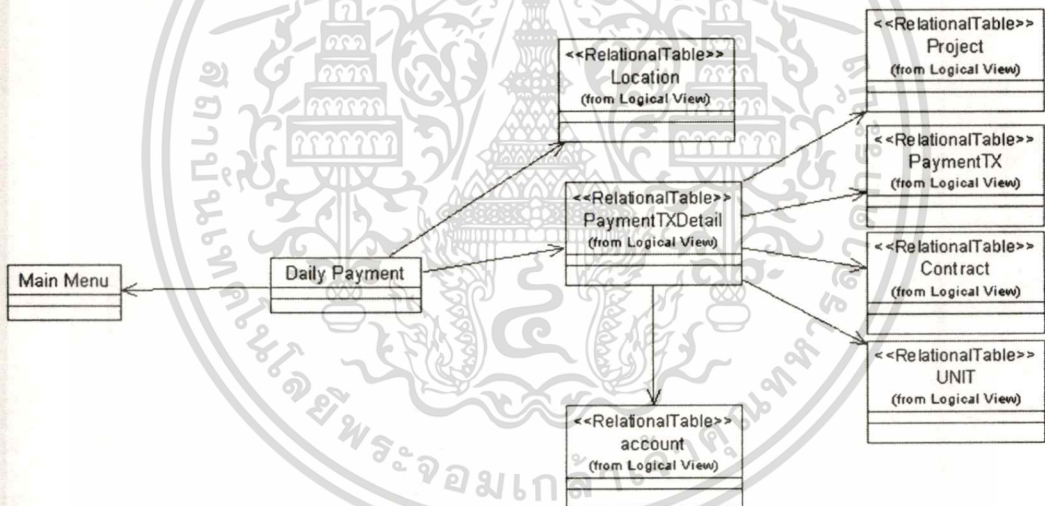
พิมพ์รายงาน พิมพ์ทุกเขตควบคุม แสดงข้อมูล กลับสู่รายการหลัก

รูปที่ 3.21 อินเทอร์เน็ตของระบบย่อย การชำระเงินประจำวัน
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.21 จะแสดงให้เห็นอินเตอร์เฟซของระบบย่อยการชำระเงินประจำวัน ซึ่งจะเป็นบริการให้ข้อมูลเกี่ยวกับการชำระเงินประจำวัน และสามารถเรียกข้อมูลสรุปทั้งเดือนได้ โดยสิ่งที่ผู้ใช้ต้องทำก็คือ ทำการเลือกเขตควบคุมที่ต้องการ และเลือกวันที่ที่จะดูข้อมูล ซึ่งการให้ข้อมูลของระบบย่อยนี้จะเป็นการให้ข้อมูลเป็นระดับวัน เพราะเป็นระบบการชำระเงินประจำวัน แต่ก็มีทางเลือกพิเศษคือ ถ้ากำหนดให้วันที่ที่ต้องการดูนั้นเป็นศูนย์ ก็จะสามารถเลือกดูข้อมูลการชำระเงินของทั้งเดือนนั้นได้

ต่อมาผู้ใช้สามารถเลือกดูเฉพาะการชำระเงินที่เป็นค่าเช่าได้ด้วยการเลือกที่ที่checkbox ออกซ์เฉพาะค่าเช่า ซึ่งข้อมูลที่แสดงที่ด้านล่างก็จะเลือกแสดงเฉพาะข้อมูลที่เป็นการชำระค่าเช่าเท่านั้น ที่ด้านล่างจะมีการสรุปยอดเงินทั้งหมดที่ปรากฏอยู่ในหน้าต่างนั้น

ต่อไปจะเป็นคลาสไดอะแกรมของระบบย่อยการชำระเงินประจำวัน ซึ่งจะแสดงความสัมพันธ์ของคลาสในระบบย่อยนี้ ในรูปที่ 3.22



รูปที่ 3.22 คลาสไดอะแกรมของระบบย่อยการชำระเงินประจำวัน

ต่อไปจะเป็นระบบหนี้เสีย ซึ่งเป็นระบบย่อยที่มีอินเตอร์เฟซ ดังแสดงในรูปที่ 3.23 และแสดงคลาสไดอะแกรมของระบบย่อยเดียวในรูปที่ 3.24

จากรูปที่ 3.23 จะเห็นว่าระบบย่อยนี้ซึ่งประกอบด้วยหน้าต่าง 1 หน้าต่างนั้น จะมีวิธีการทำงานคือให้ผู้ใช้ ใส่เขตควบคุม และเลือกโครงการก่อนที่ด้านบน ซึ่งระบบก็จะแสดงหนี้เสียทั้งหมดที่ปรากฏอยู่ในโครงการที่ผู้ใช้เลือก และในขณะเดียวกันผู้ใช้ก็จำเป็นต้องเลือกว่าต้องการให้ระบบประมวลผลหนี้เสียย้อนหลังไปเป็นระยะเวลาเท่าใด โดยสามารถใส่งวดเริ่มต้นประมวลผลได้ที่ด้านบนของอินเตอร์เฟซ

หน้าต่าง WINDOW1

หน้าเสีย

ตั้งแต่งวด: มกราคม 1999

เขตควบคุม รหัสเขต 1 ชื่อเขต bangkake region

ก่อนหน้า ต่อไป

โครงการ รหัสโครงการ ชื่อโครงการ Happy Village

ก่อนหน้า ต่อไป

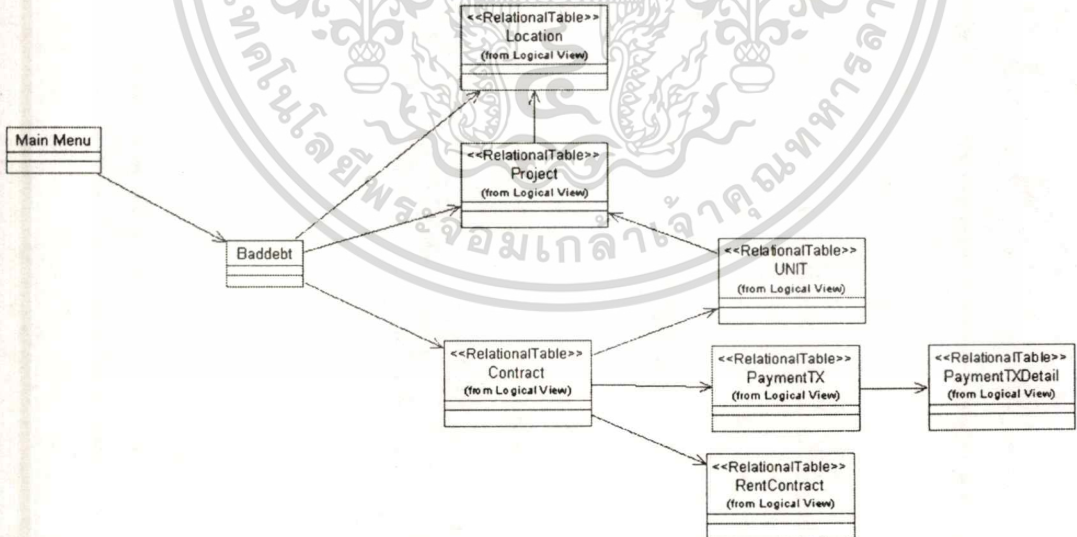
พนักงานชำระ

รหัสหน่วยอาศัย	เลขที่สัญญา	ค่าเช่าค้าง	ค่าประกันค้าง	จำนวนงวด			สถานะ	ค่านำเข้า	ชื่อ	นามสกุล
				จ่าย	ค้าง	รวม				
100001	1001000	9000	0	4	9	50	1	Mr. Frijob	Capra	
100003	102001	3000	60	0	3	60	1	Mrs. Michael	Crichton	
100005	1020009	13000	400	0	14	15	1	Mr. Frijob	Capra	

รวม 25000 460 ยอดรวมหนี้ 25460

พิมพ์รายงาน พิมพ์ทุกโครงการ กลับสู่รายการหลัก

รูปที่ 3.23 อินเทอร์เน็ตของระบบย่อยหนี้เสีย



รูปที่ 3.24 คลาสไดอะแกรมของระบบย่อยหนี้เสีย

จากนั้นระบบก็จะประมวลผลให้ โดยแยกแต่ละรายการเป็นหน่วยอาศัยและสัญญาที่มี ปัญหาค้างชำระ ระบบจะคำนวณยอดหนี้ให้ โดยแบ่งแยกเป็นหนี้ที่เกิดจากค่าเช่า และหนี้ที่เกิดจาก ค่าประกันการเช่า จากนั้นระบบก็จะสามารถรวมยอดให้ได้ และคลาสไดอะแกรมของระบบย่อยที่ เอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกมัดให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงไว้ในรูปที่ 3.24 นั้นก็จะแสดงให้เห็นความเกี่ยวเนื่อง และสายการสื่อสารของระบบหนึ่งเสียซึ่งเกี่ยวโยงไปหลายคลาส แต่ถ้าต้องการดูข้อความ คำร้องที่สื่อสารกันระหว่างคลาสอย่างมีลำดับสามารถดูได้ที่ซีควেনซ์ไดอะแกรมในระดับการออกแบบ ซึ่งจะแสดงในโอกาสต่อไป

ต่อไปจะเป็นระบบย่อยหนึ่งถึงกำหนดชำระจะมีอินเตอร์เฟซดังรูปที่ 3.25 และรูป 3.26 เป็นหน้าต่างย่อยที่แสดงรายละเอียดของหนี้ที่ถึงหนดชำระ และแสดงคลาสไดอะแกรมในรูป 3.27

รูป WINDOW1

ลูกหนี้ถึงกำหนดชำระ

เขตควบคุม

ชื่อเขต	bangkae region	ถนน	petkasame
หมู่	5	ซอย	thainamtip
ตำบล	bangwa	อำเภอ	pasrichaoroen
จังหวัด	bangkok	รหัสไปรษณีย์	10160

สรุปข้อมูลกำหนดชำระของโครงการ

รหัสโครงการ	ชื่อโครงการ	จำนวนหน่วยรวม	หน่วยที่เช่า	รวมยอดค่าเช่า
1	Happy Village	11	5	5200
3	Delaware Project	8	0	0
4	Minor Hill	0	0	0

พิมพ์ทุกหน่วยควบคุม

รูปที่ 3.25 อินเตอร์เฟซของระบบย่อยลูกหนี้ถึงกำหนดชำระ

ระบบย่อยนี้จะประกอบด้วยหน้าต่าง 2 หน้าต่างด้วยกัน คือหน้าต่างแรกนั้นจะแสดงสรุปลูกหนี้ถึงกำหนดชำระ โดยจะให้ผู้ใช้งานสามารถเลือกได้ว่าต้องการจะดูข้อมูลของเขตไหน และระบบก็จะสรุปข้อมูลโครงการภายในเขตควบคุมนั้นออกมา ว่าแต่ละโครงการที่สังกัดอยู่ในเขตปกครองนั้นมีทั้งหมดกี่หน่วยพักอาศัย และมีผู้เช่าไปเป็นจำนวนเท่าใด รวมเป็นยอดหนี้เท่าใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ต้องการดูข้อมูลรายละเอียดของแต่ละโครงการ ก็จะเลือกที่โครงการที่ต้องการ และ กดปุ่ม รายละเอียด ซึ่งก็จะเป็นการเปิด ไปดูหน้าต่างต่อไปที่รูป 3.26 ซึ่งที่หน้าต่างนี้ จะเป็นหน้าต่าง รายละเอียดของลูกหนี้ถึงกำหนดชำระ โดยจะแสดงรายละเอียดเป็นรายหน่วยอาศัย และจะแยกตาม โครงการที่สามารถเลือกได้ ดังนั้น ด้านบนก็จะแสดงข้อมูลของโครงการนี้โดยสรุปว่ามีหน่วยอาศัย ทั้งหมดเท่าใด และมีผู้เช่าเป็นจำนวนเท่าใด รวมยอดค่าเช่าเป็นเท่าใด ผู้ใช้สามารถเลือกที่จะเปลี่ยน ไปดูโครงการอื่นได้ ที่ด้านล่างของหน้าต่างก็จะแสดงข้อมูลลูกหนี้แต่ละรายว่า ที่โครงการนี้มีใคร บ้างที่เป็นลูกหนี้ โดยจะแสดงชื่อ นามสกุล และอยู่ที่หน่วยพักอาศัยใด แสดงเป็นรหัสหน่วยอาศัย มี ยอดหนี้ที่ต้องชำระเป็นเงินเท่าใด ชำระเงินด้วยวิธีการใด

รายละเอียดลูกหนี้ถึงกำหนดชำระของโครงการ

โครงการ: หน่วยเช่า:

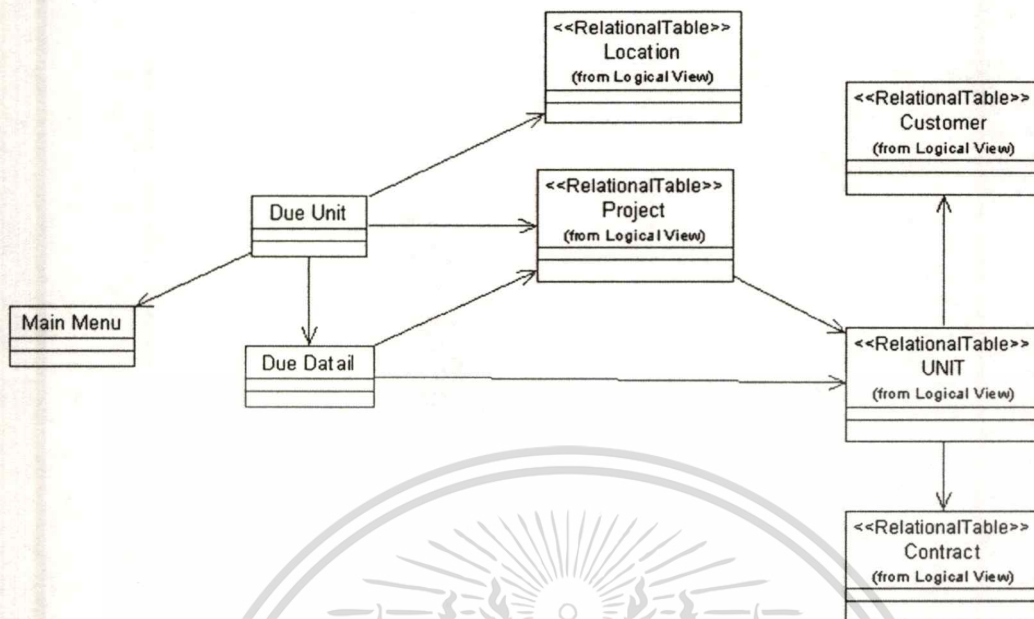
รหัสโครงการ: ยอดค่าเช่ารวม:

จำนวนหน่วยรวม:

รหัสหน่วยอาศัย	คำนำชื่อ	ชื่อลูกค้า	นามสกุล	ค่าเช่า	วิธีชำระค่าเช่า
100001	Mr.	Frijob	Capra	1000	cash
100002	Mr.	Alexandre	Dumas	1000	cash
100003	Mrs.	Michael	Crichton	1000	cash
100005	Mr.	Frijob	Capra	1000	bank
101001	Mr.	Victor	Hugo	1200	bank

รูปที่ 3.26 อินเทอร์เฟซของระบบย่อยหนี้ถึงกำหนดชำระตามโครงการ

ที่หน้าต่างแรก คือรูป 3.25 นั้นผู้ใช้สามารถพิมพ์ข้อมูลที่แสดงออกเป็นรายงานได้ โดยกดที่ปุ่ม พิมพ์รายงาน โดยสามารถเลือกได้ว่าจะพิมพ์เฉพาะหน่วยควบคุมที่แสดงผลอยู่ หรือต้องการจะพิมพ์ทุกหน่วยควบคุมออกในรายงานเดียวกัน



รูปที่ 3.27 คลาสไดอะแกรมของระบบย่อยหนึ่งถึงกำหนดชำระ

ต่อไปรูปที่ 3.28 จะแสดงหน้าต่างอินเทอร์เน็ตเฟสของระบบย่อยลูกค้าหนี้หมดอายุสัญญา และรูปที่ 3.29 จะแสดงคลาสดิอะแกรมของระบบย่อยหนี้

WINDOW1

ลูกค้าหนี้หมดอายุสัญญา

เขตควบคุม
รหัสเขต
ชื่อเขต Bangkok region

ก่อนหน้า ต่อไป

โครงการ
รหัสโครงการ
ชื่อโครงการ Happy Village

ก่อนหน้า ต่อไป

หน่วยอาศัยหมดสัญญา

รหัสหน่วย	เลขที่สัญญา	งวดแรก	ค่าเช่า	สถานะ	ค่าน้ำ	ชื่อ	นามสกุล	ค่าเช่าค้าง	ค่าประกันค้าง
100005	1020009	01 ม.ค. 1999	1000	1	Mr. Frijob	Capra	15000	500	-
101001	902001	01 ก.พ. 1999	1200	1	Mr. Victor	Hugo	1200	0	-

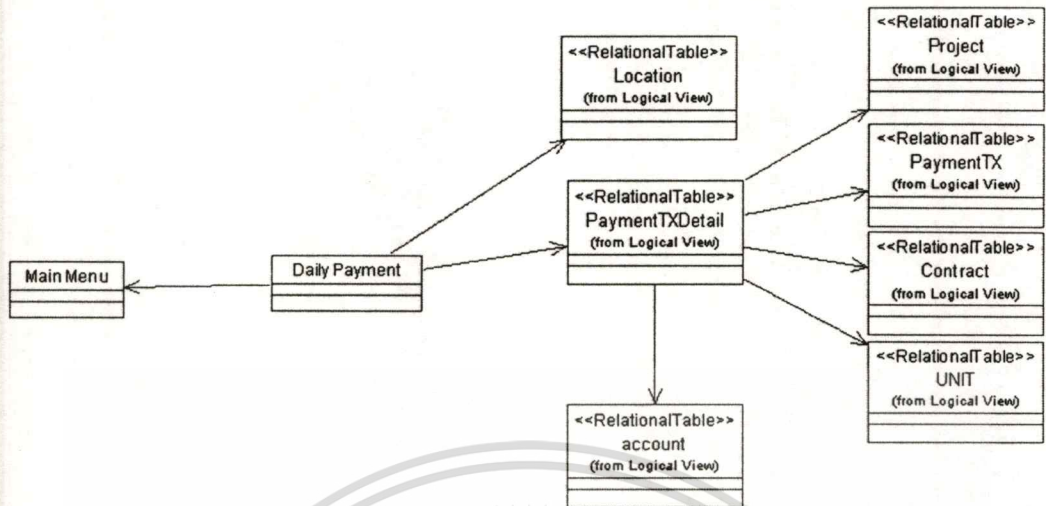
พิมพ์รายงาน พิมพ์ทุกโครงการ

กลับรายการหลัก

รูปที่ 3.28 อินเทอร์เน็ตเฟสของระบบย่อยลูกค้าหนี้หมดอายุสัญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 คลาสไดอะแกรมของระบบย่อยหนึ่งถึงกำหนดชำระ

จากรูปที่ 3.28 จะเห็นว่าผู้ใช้สามารถเลือกเขตควบคุมที่ต้องการได้ และเลือกโครงการที่อยู่ในเขตควบคุมนั้น หลังจากนั้นระบบก็จะแสดงรายการหน่วยพักอาศัยที่จะหมดสัญญาในงวดเดือนนั้น พร้อมทั้งรายละเอียดสัญญาว่ามีงวดแรกของการชำระเงินเมื่อใด อัตราค่าเช่าเป็นเท่าใด สถานะสัญญาเป็นอย่างไร แสดงข้อมูลชื่อ นามสกุลลูกค้าที่เป็นเจ้าของสัญญานั้น และข้อมูลการหนี้ค้างการชำระทั้งค่าเช่า และค่าประกันการเช่า ข้อมูลทั้งหมดสามารถส่งพิมพ์ได้ โดยกดปุ่มพิมพ์รายงานเพื่อสร้างรายงาน และสั่งพิมพ์ต่อไป โดยสามารถเลือกจะพิมพ์เฉพาะโครงการ หรือพิมพ์ทุกโครงการที่อยู่ในเขตควบคุมนั้นได้

โครงการนี้มีเป้าหมายของการออกแบบอยู่ที่ การพัฒนาระบบสารสนเทศที่จะพัฒนาบนเครื่องมือคือโอราเคิล เดเวลอปเปอร์ ซึ่งเป็นที่ทราบดีว่าโครงสร้างเดิมของโอราเคิลนั้นเป็นระบบฐานข้อมูลเชิงสัมพันธ์ และมีส่วนติดต่อกับผู้ใช้ผ่านทางฟอร์ม ซึ่งสามารถกำหนดเป็นออปเจกต์ได้อย่างที่ได้ออกแบบมาแล้ว สิ่งที่เป็นประเด็นก็คือ การออกแบบที่จะทำให้ระบบฐานข้อมูลเชิงสัมพันธ์สามารถรองรับ หรือมีโครงสร้างพื้นฐานสำหรับระบบเชิงวัตถุได้อย่างไร

จะเห็นได้ว่าโครงสร้างพื้นฐานที่ต้องการนั้นก็คือ ระบบการครอบคลุมทั้งข้อมูลและโปรแกรมให้เป็นส่วนเดียวกัน (Encapsulation) ซึ่งจากการศึกษาโครงสร้างของโอราเคิลนั้น ทำให้ทราบว่ามีการจัดชั้นบางอย่างของโอราเคิลที่รองรับการทำงานเช่นนั้น ซึ่งก็คือการสร้างหน่วยของโปรแกรมที่รวมกันเป็นแพ็คเกจนั่นเอง

ดังนั้นในการออกแบบนี้ จะเป็นการออกแบบให้ตัวคลาสที่ออกแบบไว้ มีแอทริบิวต์ซึ่งเป็นตัวข้อมูลเป็นตารางที่อยู่ในฐานข้อมูล แต่ส่วนที่เป็นบริการที่คลาสเหล่านั้นมีให้ หรือเมธอดนั้นจะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นฟังก์ชันที่สามารถสร้างได้ด้วย สตอร์โพรซีเจอร์ (Store Procedure) ซึ่งจะรวมกันไว้ที่ฝั่งเซิร์ฟเวอร์ และจะทำงานอยู่ที่ด้านเซิร์ฟเวอร์เช่นกัน โดยจะสร้างเป็นกลุ่มรวมกันไว้ในโครงสร้างที่เรียกว่า แพ็กเกจ ดังนั้นโดยสถาปัตยกรรมแล้ว ระบบนี้จะเป็นระบบที่มีออปเจกต์ของส่วนติดต่อกับผู้ใช้ทำงานอยู่เครื่องไคลเอนท์ หรือเครื่องด้านผู้ใช้ และเมื่อผู้ใช้กระทำการกิจกรรมขึ้น เมธอร์คของฟอร์มก็จะถูกเรียกใช้ และไปเรียกใช้บริการของเมธอร์คของออปเจกต์ที่อยู่ด้านเซิร์ฟเวอร์ ซึ่งก็เป็นการเรียก สตอร์โพรซีเจอร์ที่ทำงานอยู่ด้านเซิร์ฟเวอร์นั่นเอง

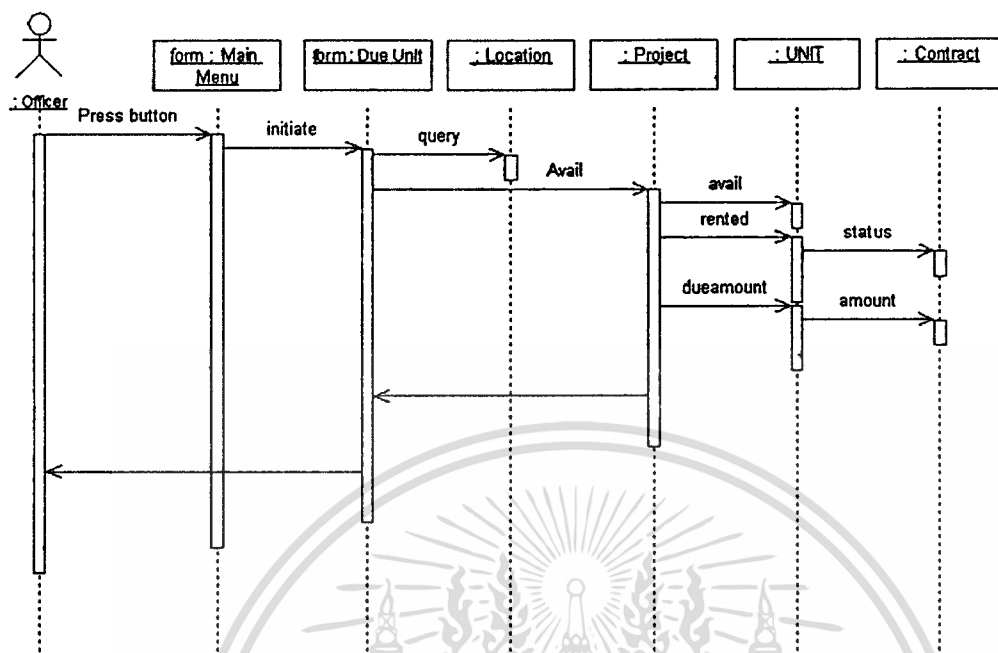
ดังนั้นขั้นตอนต่อไปของการออกแบบก็คือ การออกแบบซีเควนซ์ไคอะแกรมของระบบ เพื่อจะแสดงให้เห็นถึงการสื่อสารที่เกิดขึ้นระหว่างออปเจกต์ต่าง ๆ และการสื่อสารต่าง ๆ นั่นก็จะสามารถระบุถึงบริการที่แต่ละคลาสควรจะต้องมีให้ได้ และท้ายที่สุดก็จะได้โครงสร้างของแอปพลิเคชันที่สมบูรณ์

3.5 การออกแบบซีเควนซ์ไคอะแกรม

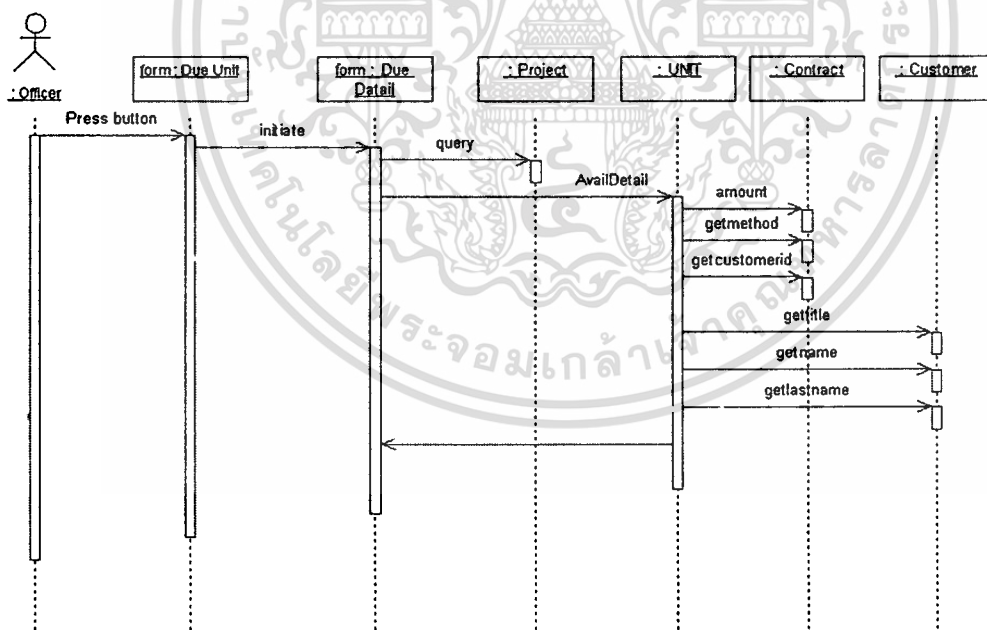
อย่างที่กล่าวไปแล้วว่าซีเควนซ์ไคอะแกรมนั้นจะเป็นลำดับต่อไปของการออกแบบ ซึ่งจะเป็นการออกแบบโดยอาศัยพื้นฐานจากคลาสไคอะแกรม และซีเควนซ์ไคอะแกรมที่ได้จากการวิเคราะห์ระบบก่อนหน้านี้ ลักษณะก็คือ หนึ่งอินเตอร์เฟสก็จะได้หนึ่งซีเควนซ์ไคอะแกรม การออกแบบในระดับนี้จะเป็นการออกแบบที่ลงรายละเอียดถึงกิจกรรมที่จะเกิดขึ้นจริงในระบบ ของแต่ละระบบย่อย ซึ่งลงรายละเอียดไปถึงกิจกรรมของแต่ละหน้าต่างที่อยู่ในระบบนั้น และแต่ละหน้าต่างก็จะลงรายละเอียดไปถึง การสื่อสารที่เกิดขึ้น ว่าบริการหนึ่ง ๆ ที่จะมีให้ นั้น จำเป็นต้องอาศัยความร่วมมือจากคลาสใดบ้าง และข้อมูลที่ส่งถึงกันนั้น มีอะไรบ้าง รวมไปถึงชนิดของข้อมูลที่สื่อสารถึงกันระหว่างออปเจกต์

สิ่งที่ต้องพิจารณาในการออกแบบซีเควนซ์ไคอะแกรมก็คือ การออกแบบบริการที่แต่ละคลาสต้องมีให้ และต้องออกแบบโครงสร้างข้อมูลที่จะใช้สื่อสารระหว่างกันด้วย โดยในโครงงานนี้ การออกแบบบริการนั้นจะกำหนดให้กับคลาสที่มีข้อมูลมากที่สุด และอยู่ใกล้กับคลาสอื่นมากที่สุด เพื่อที่จะทำให้ลดการสื่อสารให้น้อยที่สุด และทำให้คลาสที่เป็นผู้ให้บริการนั้นทำหน้าที่เป็นผู้ประสานงานระหว่างออปเจกต์ต่าง ๆ ที่จะต้องให้ข้อมูลประกอบทั้งหมดเพื่อจะให้บริการที่เป็นไปดังประสงค์ที่กำหนดไว้ในแต่ละบริการ

ผลจากการออกแบบซีเควนซ์ไคอะแกรมของระบบย่อยรายละเอียดสัญญา เป็นไปดังรูปที่ 3.30 และ 3.31 เพราะระบบย่อยนี้มีสองหน้าต่างที่ติดต่อกับผู้ใช้



รูปที่ 3.30 ซีเควนซ์ไดอะแกรมของหน้าต่างระบบถึงกำหนดชำระตามเขตปกครอง



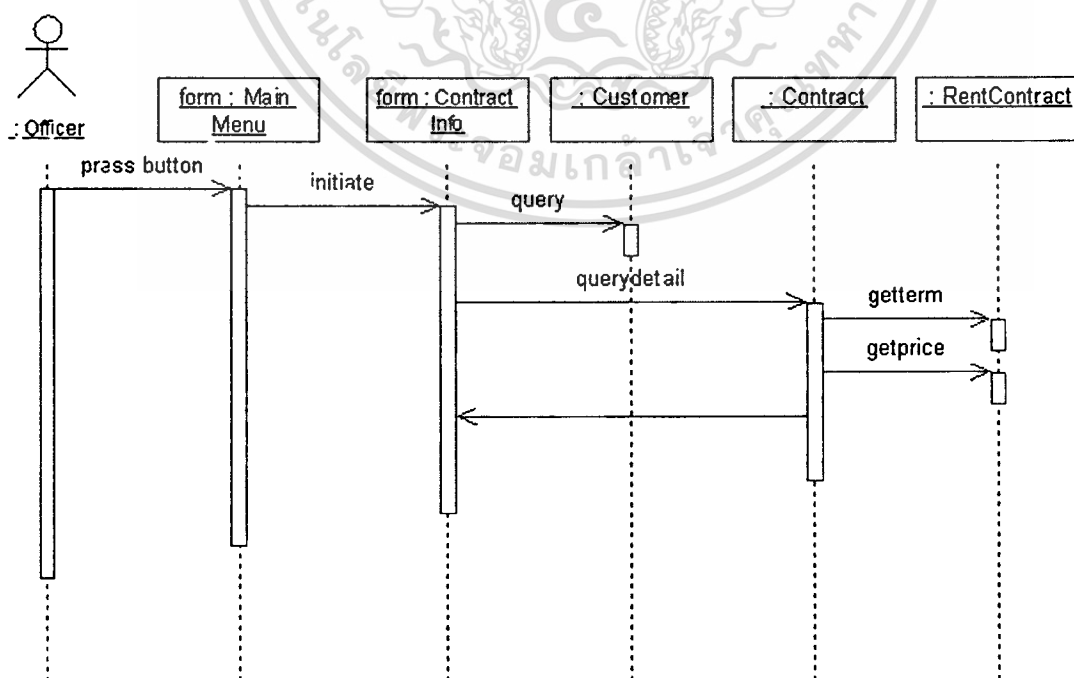
รูปที่ 3.31 ซีเควนซ์ไดอะแกรมของหน้าต่างนี้ถึงกำหนดชำระตามโครงการ

ระบบย่อยนี้ถึงกำหนดชำระนั้น จะเริ่มต้นเมื่อผู้ใช้คลิกปุ่มเลือกรายการที่เมนูหลัก ซึ่งออกเจดเมนูหลักก็จะไปเรียกออกเจดหน้าต่างแรกคือ หน้าถึงกำหนดชำระตามเขตควบคุมขึ้นมา ซึ่งเมื่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกเลือกขึ้นมาแล้ว ก็จะไปเรียกออปเจกโกลเคชั่น เพื่อสอบถามรายชื่อโกลเคชั่นทั้งหมดมาให้ผู้ใช้เลือก และก็จะส่งคำร้องไปหาออปเจกโปรเจกเพื่อให้ รายงานว่าตามเขตที่เลือกไปนั้น มีจำนวนหน่วยอาศัยที่ถึงกำหนดชำระเป็นจำนวนเท่าใด และเป็นมูลค่าเท่าใด เมื่อออปเจกโปรเจกได้รับคำร้องแล้ว ก็จะส่งคำร้องไปที่ออปเจกยูนิตเพื่อถามว่าหน่วยอาศัยนั้นว่างอยู่หรือไม่ ออปเจกยูนิตไม่ได้เก็บข้อมูลสถานะการเช่าไว้ แต่เก็บไว้ที่ออปเจกคอนแทรก หรือสัญญา ก็จะตอบกลับสถานะ และค่าเช่า ซึ่งก็สามารถสร้างเป็นก้อนข้อมูลขึ้นมาให้ออปเจกโปรเจก ส่งกลับไปให้ออปเจกฟอร์มอีกที

ส่วนต่อมาก็คือ อินเตอร์เฟสที่ย่อยลงมาจากหน้าต่างแรก ก็อันนี้ถึงกำหนดชำระตามโครงการ คือเมื่อจากหน้าต่างแรก เมื่อผู้ใช้เลือกโครงการที่ต้องการดูแล้ว กดปุ่มรายละเอียด ออปเจกฟอร์มที่แสดงตามเขตก็จะเรียกออปเจกฟอร์มที่แสดงตามโครงการขึ้นมา เมื่อเริ่มทำงานแล้ว ก็จะไปเรียกใช้ออปเจกโปรเจกเพื่อขอรายชื่อโครงการทั้งหมดเพื่อให้ผู้ใช้เลือก จากนั้นฟอร์มก็จะเรียกไปที่ออปเจกยูนิตเพื่อให้แสดงรายการหน่วยที่ถึงกำหนดชำระ ซึ่งออปเจกยูนิตก็จะเรียกใช้ออปเจกคอนแทรก เพื่อถามอัตราค่าเช่า สอบถามวิธีการจ่ายเงิน และสอบถามรหัสลูกค้า เพื่อนำไปใช้เรียกข้อมูลลูกค้าจากออปเจกคัสตอเมอร์ต่อไป ข้อมูลที่จะสอบถามมาจากออปเจกคัสตอเมอร์ก็จะมี คำนานี้ นาม ชื่อ และนามสกุล

ต่อไปจะเป็นซีควเอนซ์ไดอะแกรมของระบบย่อยรายละเอียดสัญญาซึ่งมีหนึ่งหน้าต่างดังนั้น จึงมีหนึ่งยูสเคสดังแสดงในรูปที่ 3.32



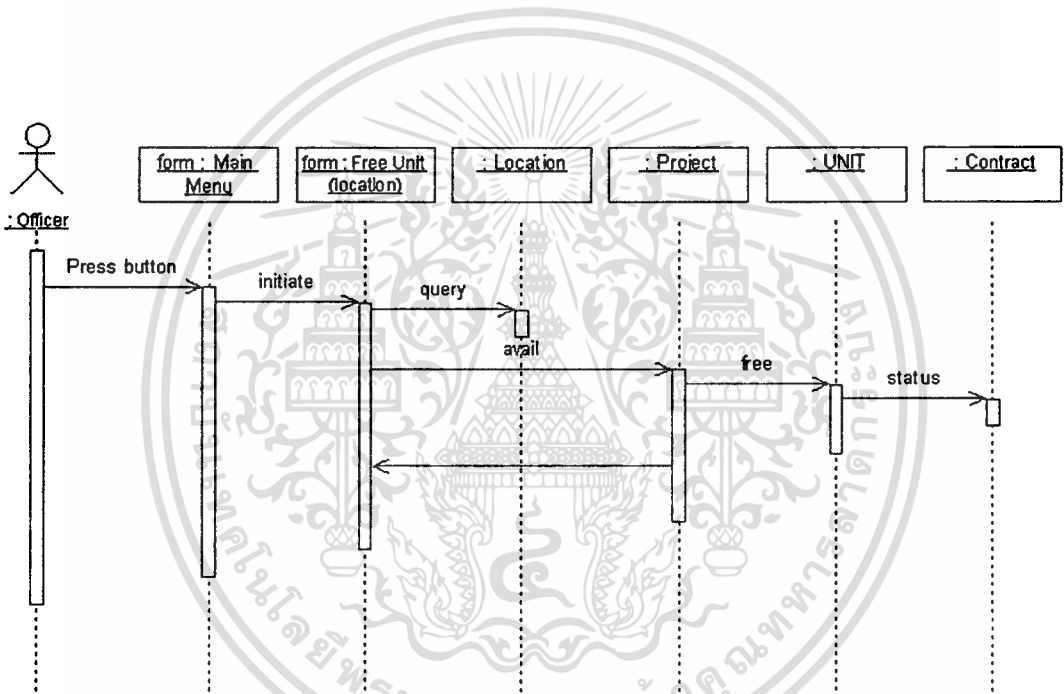
รูปที่ 3.32 ซีควเอนซ์ไดอะแกรมของหน้าต่างรายละเอียดสัญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างรายละเอียดสัญญาจะให้ผู้ใช้สามารถเรียกข้อมูลสัญญาได้ตามรายชื่อลูกค้า ดังนั้นจึงต้องส่งคำร้องไปหาแอปเจคต์สต่อเมอร์เพื่อขอรายชื่อลูกค้าทั้งหมด เมื่อได้แล้วก็ส่งคำร้องไปที่แอปเจคต์คอนแทรกเพื่อขอข้อมูลของลูกค้ารายนั้น แอปเจคต์คอนแทรกก็จะส่งคำร้องไปขอข้อมูลเพิ่มเติมจากแอปเจคต์เรนท์คอนแทรก เพราะเก็บข้อมูลเพิ่มเติมเกี่ยวกับค่าประกันการเช่าไว้ เมื่อได้ข้อมูลทั้งหมดแล้วก็ส่งข้อมูลกลับไปหาฟอร์ม

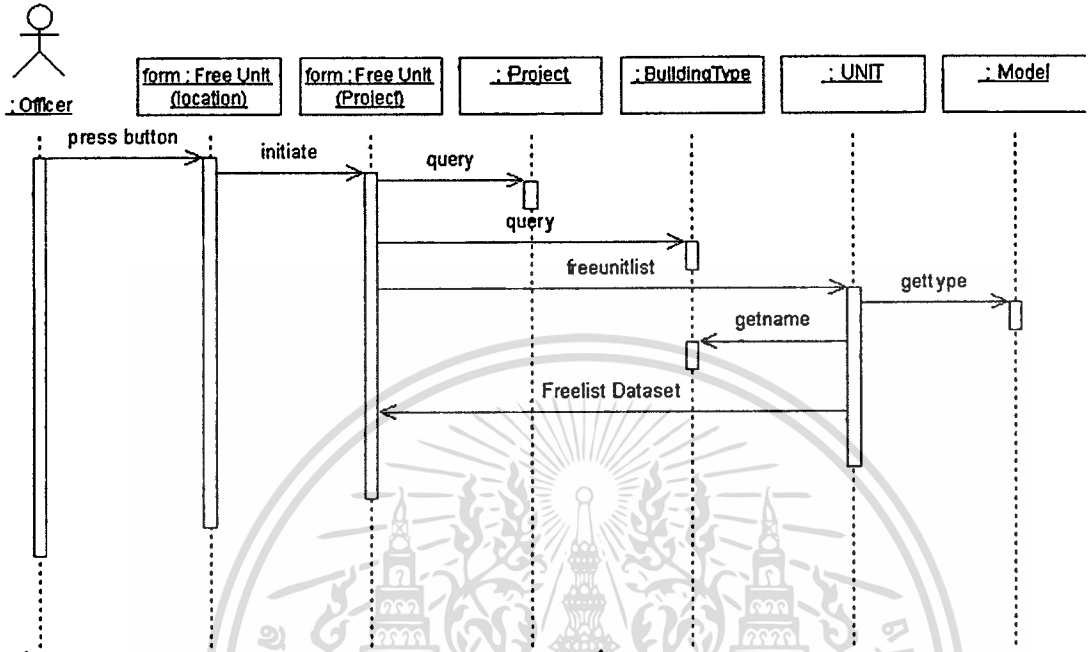
ต่อไปจะเป็นหน้าต่างหน่วยพักอาศัยที่ว่าง ซึ่งประกอบด้วย 3 หน้าต่าง คือตามเขต ตามโครงการ และรายละเอียดของแต่ละหน่วย ซึ่งมีทั้งหมด 3 ซีควนซ์ไดอะแกรมดังแสดงในรูปที่ 3.33 ถึง 3.35



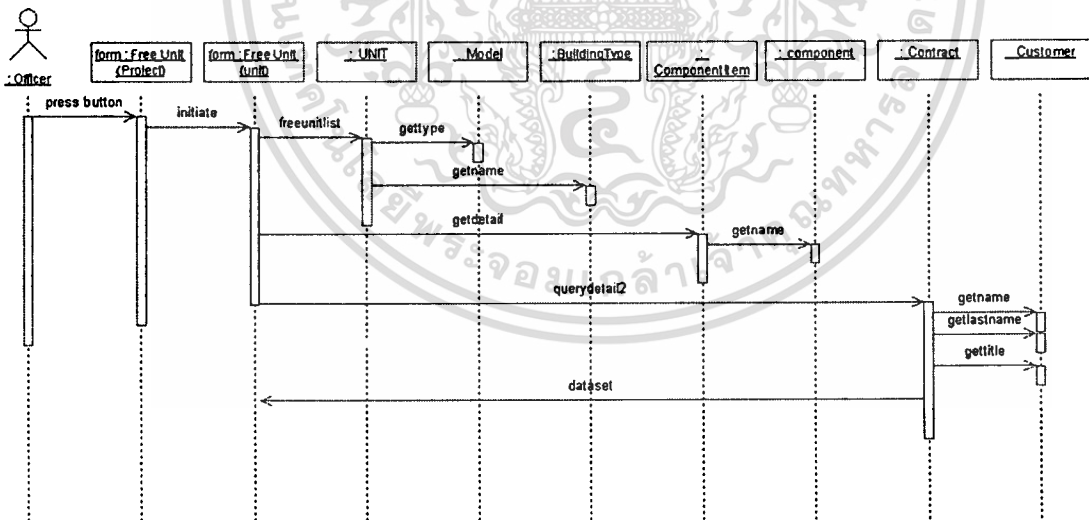
รูปที่ 3.33 ซีควนซ์ไดอะแกรมของหน้าต่างหน่วยพักอาศัยที่ว่างตามเขต

จากรูปที่ 3.33 จะเป็นซีควนซ์ไดอะแกรมของหน่วยพักอาศัยว่างตามเขต นั่นคือผู้ใช้สามารถเลือกเขตที่ต้องการแสดงข้อมูลได้ ดังนั้นฟอร์มก็จะร้องขอข้อมูลเขตทั้งหมดไปที่แอปเจคต์โลเคชัน และทำคำร้องขอข้อมูลหน่วยพักอาศัยที่ว่างตามเขตที่ต้องการไปที่แอปเจคต์โปรเจค ซึ่งแอปเจคต์โปรเจค จะสอบถามต่อไปยังแอปเจคต์ยูนิตว่ามีหน่วยพักอาศัยใดบ้างที่สังกัดโครงการนั้น ว่างอยู่หรือไม่ ซึ่งแอปเจคต์ยูนิตจะไม่ทราบว่ สถานะของหน่วยพักอาศัยนั้นเป็นอย่างไร ทำให้สอบถามไปที่แอปเจคต์สัญญาอีกทีว่าสถานะสัญญาเป็นอย่างไร ถ้าไม่มีการเช่าก็แสดงว่าสถานะยังว่างอยู่ ก็จะ

รวบรวมไว้สร้างเป็นรายชื่อหน่วยพักอาศัยที่ยังว่าง และก็จะสรุปเป็นข้อมูลส่งไปให้ออปเจก โปรเจคต่อไป ส่วนออปเจกโปรเจคเมื่อได้บทสรุปแล้วก็จะส่งชุดข้อมูลคืนไปให้ฟอร์มแสดงผลต่อไป



รูปที่ 3.34 ซีเควนซ์ไดอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตามโครงการ



รูปที่ 3.35 ซีเควนซ์ไดอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตามหน่วยพักอาศัย

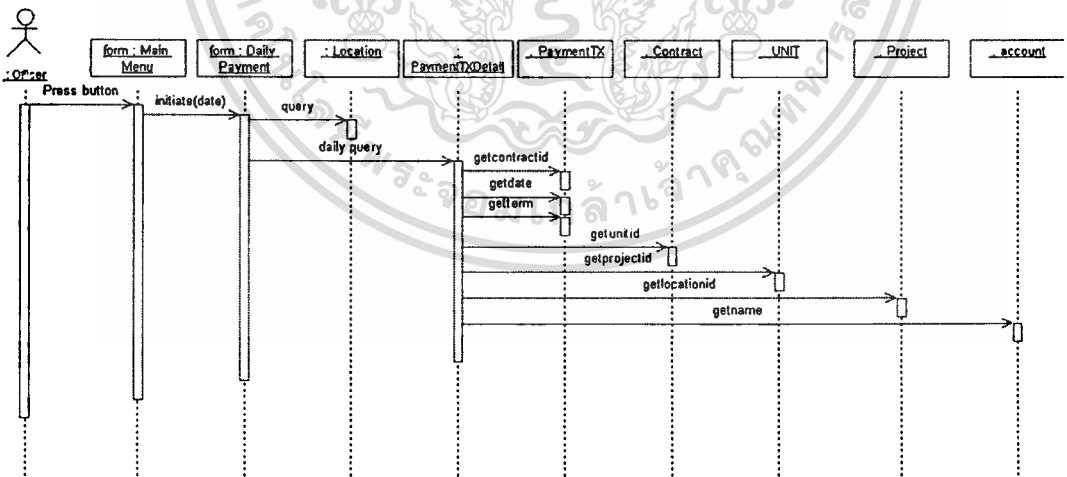
รูปที่ 3.34 จะแสดงซีเควนซ์ไดอะแกรมของหน้าต่างหน่วยพักอาศัยที่ว่างตามโครงการ ที่หน้าต่างนี้ผู้ใช้จะสามารถเลือกดูได้ตามโครงการ และเลือกชนิดของอาคารได้ด้วย ดังนั้นขั้นตอนแรก ฟอร์มก็จะต้องติดต่อไปที่ออปเจกโปรเจคและบิวคิงไทป์เพื่อขอข้อมูลทั้งหมด โดยจะวิ่งหาข้อมูลไปที่ข้อมูลที่ผู้ใช้เลือกจากหน้าต่างก่อนหน้า จากนั้นก็จะส่งค่าที่ต้องที่ไปออปเจกยูนิทเพื่อขอรายเอ็กสแตร์นเป็นเอ็กสแตร์นที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหน่วยพักอาศัยที่ว่างตามโครงการนั้น และตามชนิดหน่วยพักอาศัยนั้น ออปเจกต์ยูนิทก็จะไปถามที่ออปเจกต์โมเดลเพื่อถามชนิดของอาคารว่า โมเดลนั้นเป็นอาคารชนิดใด และก็ไปสอบถามชื่อวชนิดอาคารนั้นชื่ออะไร ก่อนจะแสดงให้ผู้ผู้เห็นต่อไปโดยส่งคืนไปให้ฟอร์ม

ซีควเอนซ์ไดอะแกรมต่อมาคืออยู่ในรูปที่ 3.35 จะเป็นข้อมูลของหน่วยอาศัยที่ว่างว่ามีรายละเอียดเช่นใด และมีประวัติการเช่าเป็นอย่างไร ทำให้จะต้องติดต่อไปหาออปเจกต์ยูนิทก่อนเพื่อเอารายชื่อหน่วยพักอาศัยมา และติดต่อไปหาออปเจกต์โมเดลเพื่อพยายามหาชนิดของอาคาร เมื่อได้แล้ว ก็ไปสอบถามชื่อชนิดอาคารจากออปเจกต์บิวคิงไทป์ ต่อไปจะเป็นพยายามให้ได้ซึ่งคุณสมบัติของหน่วยพักอาศัยว่ามีลักษณะอย่างไร โดยไปสอบถามมาจากออปเจกต์คอมโพเนนท์ไอเทม ซึ่งออปเจกต์นี้จะทราบว่าแต่ละคุณสมบัติมีปริมาณเท่าใด และออปเจกต์นี้จะไปสอบถามชื่อคุณสมบัติจากออปเจกต์คอมโพเนนท์อ็อกทอด ทำให้ท้ายที่สุดแล้วออปเจกต์ฟอร์มก็จะได้รับรายชื่อคุณสมบัติทั้งหมดพร้อมปริมาณ ต่อไปก็จะเป็นการสอบถามประวัติการเช่า โดยติดต่อไปที่ออปเจกต์คอนแทรกต์ เพื่อสอบถามสัญญาที่ตรงกับหน่วยพักอาศัยที่ต้องการ ออปเจกต์คอนแทรกต์ก็จะประสานงานไปที่ออปเจกต์คัสทอเมอร์เพื่อติดต่อขอชื่อและนามสกุลผู้ทำสัญญานั้น ทั้งหมดส่งคืนไปให้ออปเจกต์ฟอร์มเพื่อแสดงผลให้ผู้ผู้ต่อไป

ต่อไปจะเป็นซีควเอนซ์ไดอะแกรมของหน้าต่างการชำระเงินประจำวัน ซึ่งแสดงไว้ในรูปที่ 3.36 ซึ่งมีเพียงหน้าต่างดังนั้นก็จะมีหนึ่งซีควเอนซ์ไดอะแกรม



รูปที่ 3.36 ซีควเอนซ์ไดอะแกรมของหน้าต่างการชำระเงินประจำวัน

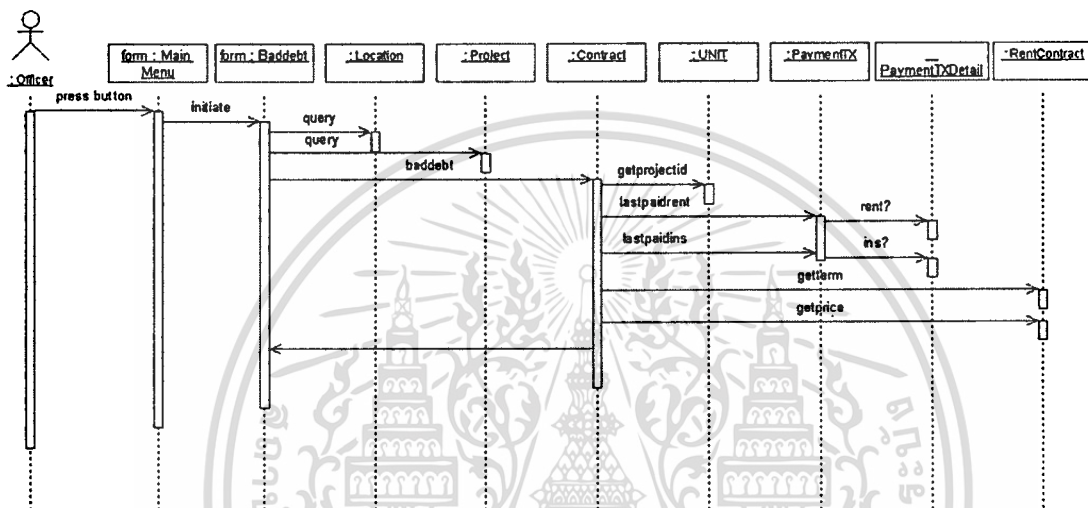
จากรูปที่ 3.36 เป็นซีควเอนซ์ไดอะแกรมของหน้าต่างการชำระเงินประจำวัน ที่มีลักษณะคือ ให้ผู้ผู้สามารถเลือกได้ว่าต้องการจะดูข้อมูลของหน่วยควบคุมใด และเลือกวันที่ที่ต้องการดูข้อมูล ดังนั้นในขั้นแรกจึงเป็นการเรียกขอข้อมูลเขตควบคุมทั้งหมดจากออปเจกต์โลเคชัน จากนั้นก็ส่งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร้องขอข้อมูลประจำวันจากออปเจกเพเมนต์ทีเลท ออปเจกนี้ก็จะส่งคำร้องไปหาออปเจกเพเมนต์เพื่อขอข้อมูลรหัสสัญญา วันที่จ่ายเงิน และงวดที่จ่าย จากนั้นก็จะติดต่อสอบถามไปยังออปเจกต่อไปนี้ตามลำดับเพื่อให้ได้มาซึ่งเจตควบคุมที่การจ่ายเงินนั้นสังกัดอยู่ ซึ่งก็คือออปเจกคอนแทรก ออปเจกยูนิต ออปเจกโปรเจก และติดต่อไปที่ออปเจกแอกเคาท์เพื่อสอบถามข้อมูลซึ่งนั่นว่าเป็นการจ่ายค่าอะไร ก่อนจะส่งข้อมูลทั้งหมดคืนไปให้ออปเจกฟอร์ม

ต่อไปจะเป็นซีควเอนซ์ไดอะแกรมสุดท้าย คือหน้าต่างระบบนี้เสีย ซึ่งแสดงไว้ในรูปที่ 3.37

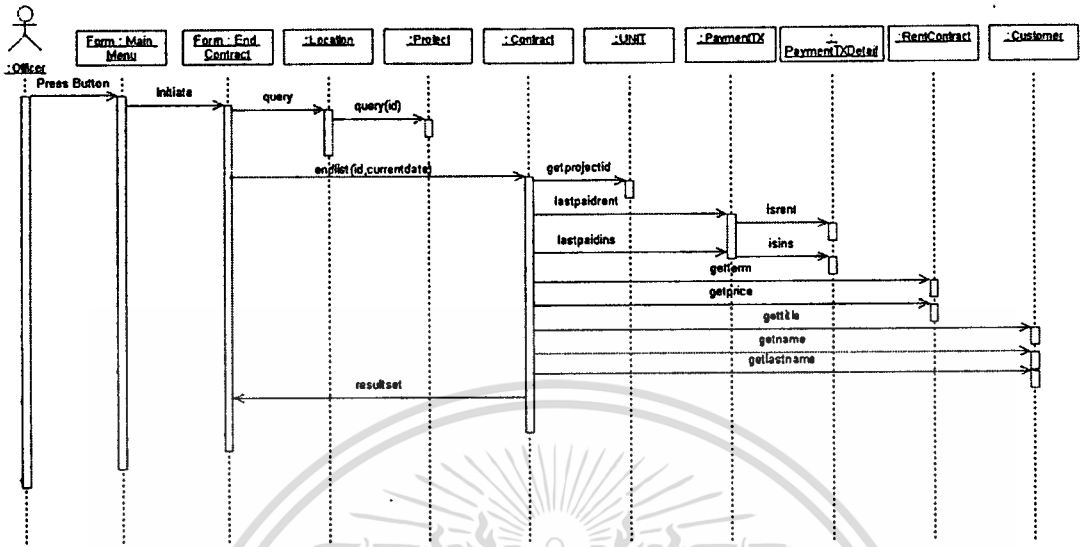


รูปที่ 3.37 ซีควเอนซ์ไดอะแกรมของหน้าต่างหนี้เสีย

ซีควเอนซ์ไดอะแกรมต่อไปจะเป็นของหน้าต่างหนี้เสีย ซึ่งมีลักษณะการทำงานคือให้ผู้ใช้สามารถเลือกดูข้อมูลได้ตามเขตควบคุมและตามโครงการ ดังนั้นกิจกรรมแรกที่ทำการก็คือให้ฟอร์มติดต่อไปยังออปเจกโลเคชัน และออปเจกโปรเจก เพื่อขอรายชื่อทั้งหมดให้ผู้ใช้เลือก จากนั้นก็จะติดต่อไปที่ออปเจกคอนแทรกเพื่อขอให้ทำรายการข้อมูลหนี้เสีย ดังนั้นออปเจกคอนแทรกก็จะติดต่อไปที่ออปเจกยูนิตเพื่อสอบถามว่าหน่วยอาศัยนั้นอยู่ในโครงการเป้าหมายหรือไม่ และก็จะติดต่อไปที่ออปเจกเพเมนต์เพื่อให้รายงานมาว่าการชำระค่าเช่า และค่าประกันครั้งสุดท้ายนั้นเป็นของงวดเท่าไร ซึ่งออปเจกเพเมนต์ก็จะต้องติดต่อไปที่ออปเจกเพเมนต์ทีเลทเพื่อสอบถามว่าแต่ละการจ่ายเงินนั้นเป็นการชำระค่าอะไรบ้าง และสุดท้ายออปเจกคอนแทรกก็จะติดต่อไปที่ออปเจกเรนท์คอนแทรกเพื่อสอบถามข้อมูลการประกันว่าจะต้องทำกี่งวด และทำงวดละเท่าใด เพื่อจะสามารถตรวจสอบได้ว่าเป็นหนี้เสียหรือไม่ และถ้าเสียจะมียอดหนี้เท่าใด หลังจากนั้นก็จะสร้างเป็นรายงานส่งคืนไปให้ฟอร์ม

ต่อไปจะเป็นซีควเอนซ์ไดอะแกรมสุดท้าย คือซีควเอนซ์ไดอะแกรมของหน้าต่างลูกหนี้หมดอายุสัญญา ซึ่งแสดงอยู่ในรูปที่ 3.38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.38 ซีควเอนซ์ไดอะแกรมของหน้าต่างลูกหนี้หมดอายุสัญญา

จากรูปที่ 3.38 ซึ่งแสดงซีควเอนซ์ไดอะแกรมสุดท้ายของหน้าต่างลูกหนี้หมดอายุสัญญานั้น จะเป็นการทำงานที่ผู้ใช้จะสามารถเลือกได้ก่อนว่าจะดูข้อมูลที่เขตควบคุมใด และโครงการใด ดังนั้นสิ่งแรกที่ต้องทำก็คือ ส่งคำร้องไปยังออปเจกต์โลเคชัน เพื่อให้แสดงรายการเขตควบคุมทั้งหมด จากนั้นเมื่อผู้ใช้เลือกแล้ว ก็จะส่งคำร้องไปยังออปเจกต์โปรเจกต์ เพื่อให้แสดงรายการโครงการทั้งหมดที่อยู่ในเขตควบคุมนั้น ผู้ใช้ก็จะสามารถเลือกโครงการที่ต้องการได้ โดยจะขอคำร้องไปยังออปเจกต์คอนแทรกต์ โดยส่งข้อมูลไปให้ค็อร์หัสโครงการที่ต้องการดูข้อมูล และวันที่งวดปัจจุบัน ออปเจกต์คอนแทรกต์ในที่นี้ก็จะทำหน้าที่เป็นผู้ประสานงาน โดยจะส่งคำร้องไปยังออปเจกต์ยูนิต เพื่อตรวจสอบว่าสัญญาใดบ้างที่อยู่ในโครงการเป้าหมาย และเลือกเฉพาะสัญญาที่จะหมดอายุในงวดปัจจุบัน เมื่อได้สัญญาที่อยู่ในโครงการนั้นแล้ว ก็จะส่งคำร้องไปยังออปเจกต์เพเมนต์ เพื่อสอบถามว่าสัญญานั้นจ่ายค่าเช่า และค่าประกันครั้งสุดท้ายเมื่อใด ออปเจกต์เพเมนต์ทำการจ่ายเงินครั้งล่าสุดของสัญญานั้น และถามไปยังออปเจกต์เพเมนต์ดีเทลว่าค่าการจ่ายนั้นเป็นค่าเช่า และค่าประกันหรือไม่ เมื่อได้วันที่จ่ายครั้งสุดท้ายแล้ว ก็จะสอบถามต่อไปยังออปเจกต์เรนท์คอนแทรกต์เพื่อสอบถามรายละเอียดสัญญาเช่าว่าต้องจ่ายค่าประกันกี่งวดและงวดละเท่าใด เพื่อนำมาประกอบการคำนวณหนี้ค้างชำระ เมื่อคำนวณเสร็จแล้ว ก็จะสอบถามไปยังออปเจกต์คัสทอเมอร์เพื่อสอบถามข้อมูลของลูกค้าคือ ชื่อและนามสกุล เพื่อนำมาแสดงผลต่อไป โดยคืนข้อมูลที่สร้างเสร็จแล้วไปให้ออปเจกต์ฟอร์มที่ทำหน้าที่เป็นอินเตอร์เฟซต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับส่วนสุดท้ายที่จำเป็นต้องมีการพิจารณานั้นก็คือ การส่งผลการค้นหาที่ปรากฏบนหน้าต่างจอ ไปออกสู่เครื่องพิมพ์ ซึ่งตามสถาปัตยกรรมนั้นจะเป็นการส่งชุดข้อมูลที่มีในฟอร์ม นั้นเป็นพารามิเตอร์ไปให้ออปเจกทีฟพอร์ต เพื่อให้สร้างออกแบบเป็นรูปแบบรายงานก่อนที่จะปล่อยออกสู่เครื่องพิมพ์ต่อไป ดังนั้นลักษณะของรายงานก็คือจะมีเอทริบิวเหมือนกับสิ่งที่ปรากฏเป็นเอทริบิวของหน้าต่าง และจัดรวมตามกลุ่มที่ปกติผู้ใช้จะเลือกได้เช่นกัน ซึ่งมีโครงสร้างตามรูปที่ 3.39 ถึงรูปที่ 3.45

รหัสเขต ชื่อเขตควบคุม รหัสโครงการ ชื่อโครงการ รหัสหน่วยอาศัย รหัสสัญญา วิธีชำระ สถานะ วันที่ทำสัญญา จำนวนงวด ค่าเช่า รวมยอดหนี้ของโครงการ รวมยอดหนี้ของเขต
--

รูปที่ 3.39 รายงานหนี้ถึงกำหนดชำระ

รหัสลูกค้า ชื่อลูกค้า ที่อยู่ เบอร์โทรศัพท์ เลขที่สัญญา รหัสหน่วยอาศัย วันที่ทำสัญญา วิธีชำระ ค่าเช่า งวดแรก จำนวนงวดเช่า จำนวนงวดประกัน ค่าประกัน สถานะสัญญา รวมจำนวนสัญญา
--

รูปที่ 3.40 รายงานรายละเอียดสัญญา

รหัสเขต ชื่อเขตควบคุม รหัสโครงการ ชื่อโครงการ รหัสหน่วยอาศัยที่ว่าง ชนิดอาคาร ชั้นที่ อาคารที่ รวมจำนวนต่อโครงการ รวมจำนวนต่อเขต
--

รูปที่ 3.41 รายงานหน่วยอาศัยที่ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสหน่วยอาศัย ชั้นที่ อาคารที่
 รหัสสัญญา วันที่ทำสัญญา รหัสลูกค้า ชื่อลูกค้า สถานะสัญญา
 รวมจำนวนต่อหน่วยอาศัย

รูปที่ 3.42 รายงานหน่วยอาศัยที่ว่างประวัติการทำสัญญา

รหัสหน่วยควบคุม ชื่อหน่วยควบคุม
 วันที่การชำระเงิน
 รหัสการชำระเงิน งวดที่จ่าย จำนวน รายการค่าใช้จ่าย เลขที่สัญญา
 รวมยอดประจำวัน
 รวมยอดของเขต

รูปที่ 3.43 รายงานการชำระเงินประจำวัน

รหัสโครงการ ชื่อโครงการ
 รหัสหน่วยอาศัย รหัสสัญญา ยอดหนี้เช่า ยอดหนี้ประกัน จำนวนงวด จำนวนงวดที่จ่าย
 จำนวนงวดค้างจ่าย สถานะสัญญา ชื่อลูกค้า รวมยอดหนี้
 รวมยอดหนี้โครงการ

รูปที่ 3.44 รายงานหนี้เสีย

รหัสโครงการ ชื่อโครงการ
 รหัสหน่วยอาศัย รหัสสัญญา ค่าเช่า งวดแรก ชื่อลูกค้า สถานะสัญญา ยอดหนี้เช่า ยอดหนี้
 ประกัน
 รวมยอดหนี้โครงการ

รูปที่ 3.45 รายงานลูกหนี้หมดอายุสัญญา

3.6 การสร้างระบบ

ระบบนี้จะเป็นการแปลงจากระบบเชิงวัตถุไปเป็นระบบเชิงฐานข้อมูลเชิงสัมพันธ์ และโปรแกรมที่เป็นโปรแกรมโครงสร้างธรรมดา สำหรับระบบการแปลงนั้นได้กล่าวไปแล้วในหัวข้อ ทฤษฎีที่เกี่ยวข้อง และส่วนเมธอร์คของออปเจคนั้นจะกลายเป็นสโคร์โพรซีเจอร์ที่อยู่กับเซอเวออร์ ซึ่งจะถูกรวมกันเป็นกลุ่มแพ็คเกจ

เพื่อเป็นการขยายที่มาของการออกแบบฐานข้อมูล จากคลาสไดอะแกรมที่มีการระบุจำนวนที่มีในความสัมพันธ์ด้วย (Cardinality) ดังในรูปที่ 3.10 ซึ่งจะสังเกตว่าเป็นคลาสไดอะแกรมที่ไม่มีเมธอร์ค มีแต่แอทริบิว ซึ่งทำให้คลาสไดอะแกรมทำหน้าที่เหมือนเอ็นทีดี รีเลชันไดอะแกรม (ER-Diagram) ซึ่งเป็นการบอกโดยอ้อมว่า การออกแบบคลาส หรือเอ็นทีดี ซึ่งในที่นี้จะมีลักษณะเหมือนกัน คือหนึ่งคลาสหรือหนึ่งเอ็นทีดีจะมีความหมายเดียว มีหน้าที่เดียว และไม่มีความสัมพันธ์แบบ หลายข้อมูลต่อหลายข้อมูล เพราะถูกแยกออกเป็น คู่ของความสัมพันธ์แบบ หนึ่งต่อหลายข้อมูลหมดสิ้นแล้ว

ผลก็คือทำให้คลาสไดอะแกรมที่ได้สามารถสร้างเป็นโครงสร้างฐานข้อมูล ซึ่งสามารถวิเคราะห์การปลดออกจากข้อผิดพลาดได้ หรือสามารถทำการนอร์มอลไลซ์ต่อไปได้ ซึ่งจากการวิเคราะห์ทำให้ทราบได้ว่า ทุกตารางในฐานข้อมูลนั้น ไม่มีความสัมพันธ์ระหว่างแอทริบิวแบบ หลายค่าขึ้นต่อกัน (MVD - Multi Value Dependent) เพราะมีแต่ความสัมพันธ์แบบฟังก์ชันการขึ้นต่อกัน (FD - Function Dependency) ไม่มีความสัมพันธ์แบบถ่ายทอด (Transitive Relation) และไม่มีการขึ้นต่อบางส่วนของคีย์ (Partial Dependent) ทำให้เป็นนอร์มอลฟอร์มในรูปแบบที่สี่ (Fourth Normal Form) โดยทันที

สำหรับการผลการสร้างระบบนั้นจะแสดงโดยเรียงตามคลาส โดยส่วนแอทริบิวนั้นจะแสดงด้วยโครงสร้างฐานข้อมูล หรือ Database Schema ส่วนเมธอร์คนั้นจะถูกแสดงด้วยชื่อฟังก์ชัน และโพรซีเจอร์ที่รวมกันในแพ็คเกจของคลาสนั้น ซึ่งจะแสดงถึงพารามิเตอร์ และข้อมูลที่ส่งคืนด้วย ดังรูป 3.46 ถึงรูปที่ 3.59

```
Location (location_id, locationname, road, moo, soi, district, city, province, postalcode)
- query(location_id, locationname, road, moo, soi, district, city, province, postalcode)
```

รูปที่ 3.46 โครงสร้างคลาสโลเคชัน

Project(project_id, projectname, location_id)

- query(project_id, projectname, location_id)
- avail(project_id, projectname, location_id, unitcount, rentcount, freeunit, dueamount)
- getlocationid(project_id) return number;

รูปที่ 3.47 โครงสร้างคลาสโปรเจกต์

Model(model_id, modelname, buildingtype_id)

- gettype(model_id) return number
- getname(model_id) return varchar;

รูปที่ 3.48 โครงสร้างคลาสโมเดล

Buildingtype (buildingtype_id, buildingtypename)

- query(buildingtype_id, buildingtypename)
- getname(buildingtype_id) return varchar

รูปที่ 3.49 โครงสร้างคลาสบิวดิ้งไทป์

Componentitem(model_id, component_id, componentquantity)

- getdetail(model_id, component_id, componentquantity, componentname, id model_id)

รูปที่ 3.50 โครงสร้างคลาสคอมโพเนนท์ไอเทม

Component(component_id, component)

- getname(component id)

รูปที่ 3.51 โครงสร้างคลาสคอมโพเนนท์

Quotation(model_id, project_id, unitamount, space, cost, stdinterest)

- query(model_id, project_id, unitamount, space, cost, stdinterest)

รูปที่ 3.52 โครงสร้างคลาสโควเทชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unit(unit_id, debtorcode, floorno, buildingno, *model_id*, *project_id*)

- query(unit_id, debtorcode, floorno, buildingno, *model_id*, *project_id*)
- avail(id *project_id*)return unit number;
- rent(id *project_id*) return unit number;
- dueamount(id *project_id*) return amount number;
- availdetail(unit_id, title, firstname, lastname, rentrate, method)
- freeunit(id number)return number;
- freeunitlist(unit_id, buildingtype_id, buildingtypename, *project_id*, floorno, buildingno, *model_id*, modelname)
- getprojectid(id *unit_id*)return number;

รูปที่ 3.53 โครงสร้างคลาสยูนิต

Rentcontract(contract_id, insuranceterm, insuranceprice)

- getterm(id number) return number;
- getprice(id number) return number;

รูปที่ 3.54 โครงสร้างคลาสเรนท์คอนแทรค

Customer(customer_id, title, firstname, lastname, regisaddr, currentaddr, telephone)

- gettitle(cusid number) return varchar
- getname(cusid number) return varchar
- getlastname(cusid number)return varchar
- query(customer_id, title,firstname,lastname, regisaddr,currentaddr, telephone)

รูปที่ 3.55 โครงสร้างคลาสคัสตอมเมอร์

Contract(contract_id, contractdate, unit_id, contractstatus, paymenttype, priceperterm, datefirstterm, term, customer_id)

- amount(unitid number)return number;
- status(unitid number)return number;
- getcusid(unitid number)return number;
- getmethod(unitid number)return varchar;
- querydetail(contract_id, contractdate,contractstatus, paymenttype, priceperterm, datefirstterm, term, lastterm, insuranceterm, insuranceprice, unit_id, title,firstname,lastname id unit_id);
- getunitid(contract_id)return number;
- querybaddebt(contract_id, unit_id, rentdebt, insdebt, title, name, lastname, totalterm, paidterm, badrentterm, status, id project_id, startmonth, startyear, nowdate)
- endquery(unit_id, contract_id, datefirstterm, priceperterm, status, rentdebt, insdebt, id title, name, lastname, projectid, current date)

รูปที่ 3.56 โครงสร้างคลาสคอนแทรก

Paymenttx(paymenttx_id, contract_id, paymentdate, paymentterm)

- getdate(txid)return date;
- getcontractid(txid)return number;
- getterm(txid)return date;
- lastpaidterm(contract_id)return date;
- lastpaidins(contract_id)return date;

รูปที่ 3.57 โครงสร้างคลาสเพเมนต์

Paymenttxdetail(paymenttx_id, account_id, paymentprice)

- dailyquery(paymenttx_id, paymentdate, contract_id, unit_id, accountname, amount, term, location_id, querydate, querymonth, queryyear, id location_id, rentfilter)
- isrent(txid) return number;
- isins(txid)return number;

รูปที่ 3.58 โครงสร้างคลาสเพเมนต์ดีเทล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Account(account_id, accountname, accountdesc)

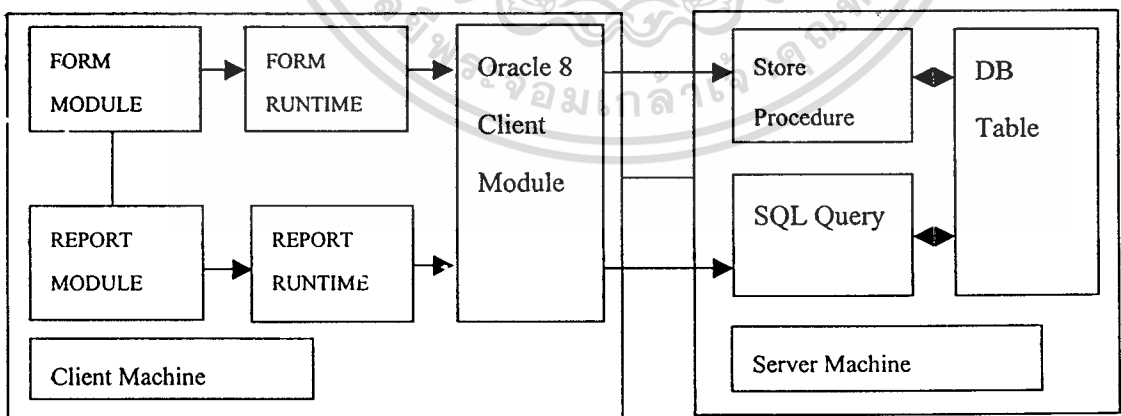
-getname(account_id)return varchar

รูปที่ 3.59 โครงสร้างคลาสแอคเคาท์

ซึ่งการออกแบบโครงสร้างแบบนี้คือให้แอทริบิวต์เป็นตารางในฐานข้อมูลและกำหนดให้เมธอด หรือบริการที่คลาสจะมีให้กลายเป็นแพคเกจของสโตร์โพรซีเจอร์ ทำให้ระบบการออกแบบและพัฒนาแบบเชิงวัตถุยังคงดำรงตามกฎอยู่ และสามารถเป็นไปได้

ซึ่งลงไปรายละเอียดก็คือ การที่คลาสยังคงดำรงกฎความถูกต้องคือ ข้อมูลนั้นเป็นสมบัติเฉพาะตัวของคลาสและผู้อื่นจะไม่สามารถเข้าไปจัดการได้โดยตรง หากแต่ต้องจัดการผ่านทางบริการที่คลาสมีให้ หรือมองอีกมุมหนึ่งก็คือข้อมูลของคลาสนั้นจะถูกห่อหุ้มไว้ด้วยโพรซีเจอร์ของคลาสนั้น ทำให้สามารถรักษาความเป็นอะตอมเอาไว้ได้ คือการเปลี่ยนแปลงใด ๆ ที่เกิดขึ้นภายในจะไม่มีผลกระทบต่อส่วนอื่น ๆ ควบคุมใดที่บริการที่ทำหน้าที่เป็นส่วนติดต่อกันไม่มีการเปลี่ยนแปลง

ส่วนต่อมาจะเป็นการอธิบายโครงการการทำงานของระบบ โดยลักษณะการทำงานแล้วจะเป็นสถาปัตยกรรม แบบไคลเอนท์ เซอร์เวอร์ (Client / Server) แบบ 2 ระดับ (2 Tiers) โดยมีลักษณะเป็นไคลเอนต์เล็ก (Thin Client) คือระบบงานจะถูกเก็บไว้ที่ด้านเซอร์เวอร์ มีเพียงส่วนน้อยและส่วนติดต่อกับผู้ใช้เท่านั้นที่อยู่ด้านผู้ใช้



รูปที่ 3.60 โครงสร้างสถาปัตยกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.60 จะสังเกตว่าทางด้านไคลเอนท์ นั้นจำเป็นต้องมีส่วนประกอบคือส่วนรันไทม์ของทั้งฟอร์ม และรายงาน และส่วนโอราเคิลไคลเอนต์ เพื่อทำหน้าที่เป็นไคลเอนท์ของฐานข้อมูลติดต่อกับส่วนเซิร์ฟเวอร์ และทางด้านเซิร์ฟเวอร์ก็จะมีสโตร์โพรซีเจอร์ ทำหน้าที่รับการติดต่อ

ซึ่งสโตร์โพรซีเจอร์ ที่อยู่ทางด้านเซิร์ฟเวอร์นั้น จะถูกรวบรวมไว้เป็นแพ็คเกจ โดยแต่ละแพ็คเกจจะรวบรวมโพรซีเจอร์สำหรับคลาสหนึ่งคลาส ทำให้มีจำนวนแพ็คเกจเท่ากับจำนวนคลาสพอดี ทำหน้าที่เป็นอินเตอร์เฟซให้กับเซิร์ฟเวอร์

ในการสร้างระบบงานจริงนั้นยังมีปัญหาอีกปัญหาหนึ่ง คือเครื่องมือที่ใช้ในการพัฒนานั้นเกิดไม่รองรับโครงสร้างที่ออกแบบไว้ นั่นคือปัญหาในการสร้างรายงานเพื่อพิมพ์ออกทางเครื่องพิมพ์ โดยการออกแบบนั้น ออกแบบไว้ฟอร์มซึ่งมีข้อมูลที่ต้องการส่งชุดข้อมูลไปให้รายงานเพื่อสร้างเป็นรายงาน แต่ว่าโอราเคิลและเดเวลลอปเปอร์นั้น มิใช่ระบบเชิงวัตถุที่แท้ และไม่ใช่มัธยาระบบเชิงโครงสร้างที่ดี เพราะไม่รองรับการสื่อสารพารามเตอร์แบบที่สามารถให้โปรแกรมเมอร์สามารถกำหนดโครงสร้างข้อมูลเอง ดังนั้นปัญหาที่เกิดขึ้นก็คือ ไม่สามารถส่งชุดข้อมูลไปให้รายงานได้ ดังนั้นสิ่งที่ต้องทำก็คือ ต้องให้รายงานสร้างชุดข้อมูลของตนเองขึ้นมา และก็เกิดปัญหาที่สองคือ รายงานนั้นไม่มีความสามารถในการติดต่อกับสโตร์โพรซีเจอร์ได้ สามารถทำได้เพียงติดต่อกับตารางข้อมูลโดยผ่านคำสั่งเรียกค้นเอสคิวแอลเท่านั้น ดังนั้นโครงสร้างการทำงานของรายงานจึงเป็นโครงสร้างการทำงานที่แตกต่างกับฟอร์มโดยสิ้นเชิง แต่ให้ผลลัพธ์ที่เหมือนกัน

แต่อย่างไรก็ดี การที่รายงานติดต่อเข้าที่ข้อมูลโดยตรงนับเป็นการละเมิดกฎการเข้าถึงข้อมูลของระบบเชิงวัตถุทีเดียว แต่ก็เป็นการละเมิดที่ไม่สร้างความเสียหายแต่ความถูกต้องของระบบ เพราะรายงานเป็นการทำงานที่อ่านข้อมูลอย่างเดียวเท่านั้น มิได้ทำการแก้ไขหรือปรับแต่งข้อมูลแต่อย่างใด ทำให้สามารถคงความถูกต้องเอาไว้ได้ แต่ถ้ามีความเปลี่ยนแปลงเกิดขึ้นภายในแอปพลิเคชัน ข้อมูลความเปลี่ยนแปลงนั้นก็จะแพร่ไปสู่รายงานอย่างไม่สามารถหลีกเลี่ยงได้

บทที่ 4

สรุปผลการพัฒนา

โครงการนี้เป็นโครงการที่ครบสมบูรณ์ในตัวเอง คือเป็นโครงการที่วิเคราะห์ ออกแบบ และพัฒนาระบบจนจบสิ้นกระบวนการ วัตถุประสงค์หลักของโครงการนี้ก็คือ ศึกษาความเป็นไปได้ และทดลองการพัฒนาระบบเชิงวัตถุ โดยใช้ทั้งเครื่องมือในการวิเคราะห์และออกแบบที่เป็นเชิงวัตถุ ในขณะที่การพัฒนานั้นจะเป็นเชิงโครงสร้าง และเป็นฐานข้อมูลเชิงสัมพันธ์

จากการทดลอง ซึ่งเป็นการวิเคราะห์ ออกแบบ และพัฒนานั้น ทำให้ได้ทราบว่าพัฒนาระบบในแนวทางนี้ มีความเป็นไปได้ และสามารถทำได้ โดยมีเงื่อนไขว่าระบบที่จะเป็นเครื่องมือในการพัฒนานั้นจำเป็นต้องมีโครงสร้างรองรับ เช่นถ้าเป็นระบบฐานข้อมูล ก็จะต้องจำเป็นที่จะยอมให้ผู้ออกแบบสามารถสร้างโปรแกรมรองรับการเข้าใช้งานกับข้อมูลได้ ซึ่งมีได้เป็นเช่นนั้นในทุก ๆ ระบบ สิ่งนี้เป็นสิ่งที่สำคัญมาก เพราะถ้าไม่มีโครงสร้างดังนี้แล้ว การพัฒนาระบบในเชิงวัตถุก็จะเป็นไปไม่ได้เลย เพราะถึงแม้จะมีการออกแบบในเชิงวัตถุแล้ว แต่ถ้าเครื่องมือในการพัฒนาไม่สามารถรองรับได้ ก็จะไม่สามารถรักษาความถูกต้องของโครงสร้างเอาไว้ได้

อย่างไรก็ดี การใช้เครื่องมือในการพัฒนาที่ไม่ได้เป็นเชิงวัตถุอย่างแท้จริง ทำให้ต้องเกิดการคิดแปลง และปรับเปลี่ยนโครงสร้าง หรือยืดความสามารถอื่นมาใช้โดยอ้อมนั้น ทำให้เกิดความไม่สะดวกในการพัฒนาอย่างยิ่ง เพราะเครื่องมือแต่ละชนิดก็ยอมออกแบบมาให้มีความสะดวกในวิธีการหลักที่ตนสนับสนุนอยู่

ประสิทธิภาพในการทำงานก็เป็นอีกประเด็นที่ต้องพิจารณา อย่างไรก็ตามข้อบ่งชี้การทำงาน ของโครงการนี้มีได้คำนึงถึงประสิทธิภาพในการทำงานเป็นหลัก ซึ่งจะเห็นได้ว่าหลักการทำงานหลักของระบบนี้จะเป็นการสื่อสารกันระหว่างแอปพลิเคชันที่เกี่ยวข้อง ซึ่งจำเป็นต้องอาศัยการสื่อสารด้วยชุดข้อมูลขนาดใหญ่ ๆ เช่นกลุ่มของเรคอร์ดเป็นพารามิเตอร์ และขึ้นอยู่กับวิธีการเรียกค้นซ้อนกันเป็นหลาย ๆ ชั้น ซึ่งเป็นสิ่งที่กระทำโดยผู้เขียนหรือพัฒนาระบบ ทำให้กลไกปกติในการปรับปรุงประสิทธิภาพของฐานข้อมูล (Optimizer) ไม่สามารถทำงานได้ เช่นการปรับปรุงประสิทธิภาพที่เกิดจากการจอยน์หลาย ๆ ตารางร่วมกันโดยใช้ฟังก์ชันแฮทช์เป็นต้น ซึ่งประสิทธิภาพนี้เป็นประเด็นที่จะต้องศึกษาต่อไปว่า การพัฒนาระบบเชิงวัตถุ ซึ่งมีแนวคิดเชิงห่อหุ้มข้อมูลให้มีเฉพาะคลาสตนเองเท่านั้นที่จะสามารถเข้าถึงข้อมูลตนเองนั้น จะสามารถปรับปรุงประสิทธิภาพได้อย่างไร จึงจะทำให้เกิดผลลัพธ์ที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ดี การพัฒนาระบบเชิงวัตุนั้นก็มีประโยชน์หลาย ๆ อย่างที่คิดว่าพัฒนาระบบในกรอบเดิม อย่างเช่น การที่มองข้อมูลเป็นกล่องคำที่มีทั้งข้อมูลและบริการไว้ในที่เดียวกันนั้น ทำให้สามารถจับผลกระทบที่เกิดจากการเปลี่ยนแปลงภายในเอาไว้ได้ สิ่งนี้เป็นสิ่งที่เกิดขึ้นตลอดระยะเวลาพัฒนาโครงการนี้ ที่การเปลี่ยนแปลงใด เช่นการเพิ่มความสามารถใหม่ ๆ นั้นสามารถกระทำได้ทันที ในขณะที่ส่วนอื่น ๆ ของระบบยังคงทำงานโดยไม่รู้สึกลถึงความเปลี่ยนแปลงที่เกิดขึ้น ข้อดีต่อมาก็คือ การนำกลับมาใช้ใหม่ ซึ่งถ้าการออกแบบยึดถือตามแนวทางการออกแบบคือให้แต่ละส่วนงานนั้นมีหน้าที่เดียว และเป็นหน้าที่ที่ค่อนข้างเป็นการทั่วไป มิใช่เป็นการเฉพาะ ก็จะทำให้บริการนั้นสามารถถูกนำกลับมาใช้ใหม่ได้

ซึ่งในการพัฒนาระบบนี้ก็เกิดสิ่งนี้ตลอดเวลาเช่นกัน มีหลายครั้งที่บริการเก่าสามารถถูกนำมาเรียกใช้ใหม่ได้ทันที และมีอีกหลายกรณีที่บริการเก่าถูกนำมาเพิ่มความสามารถเพื่อให้มีบริการเพิ่มขึ้น โดยเพิ่มเติมจากสิ่งที่มีอยู่แล้ว ก็ทำได้ทันทีเช่นกัน

ข้อแตกต่างระหว่างการพัฒนากระบวนเชิงวัตุนั้น กับการพัฒนากระบวนเชิงโครงสร้างนั้น มีมากมาย เช่นการระบบออกเป็นระบบย่อยนั้น ระบบเชิงโครงสร้างจะแบ่งระบบงานออกโดยโครงสร้างต้นไม้ของฟังก์ชัน แต่ระบบเชิงวัตุนั้นจะแบ่งโดยการกำหนดบริการของแต่ละแอปเจต นอกจากนั้นการรักษาความถูกต้องของข้อมูล นั้นระบบเชิงวัตุนั้นจะดูแลความถูกต้องของข้อมูลตนเอง แต่ระบบเชิงโครงสร้างต้องอาศัยความร่วมมือระหว่างส่วนต่าง ๆ และสุดท้ายคือการนำโปรแกรมเก่ากลับมาใช้ใหม่ ระบบเชิงโครงสร้างนั้นจะต้องสร้างเป็นชุดฟังก์ชันหรือบริการมาตรฐานเพื่อให้ส่วนต่าง ๆ สามารถนำกลับมาใช้ได้ ไม่สามารถข้ามไปเรียกใช้โมดูลอื่นได้ เพราะจะเกิดความซับซ้อนในการปรับแต่งแก้ไข และจะทำให้โครงสร้างการไหลของข้อมูลซับซ้อนโดยไม่จำเป็น แต่ในระบบเชิงวัตุนั้นสามารถเรียกใช้บริการเก่าที่คลาสต่าง ๆ มีได้ทันที

ดังนั้นโดยสรุปก็คือ ระบบงานนี้ซึ่งอ้างอิงจากสถานการณ์ของระบบบัญชีลูกหนี้เช่าของการเคหะแห่งชาตินั้น เป็นระบบที่สามารถพัฒนาได้ และการพัฒนาเชิงวัตุนั้นให้ข้อดีมากมาย และมีอีกหลายสิ่งที่ต้องการได้รับการปรับปรุงวิธีการเพื่อประสิทธิภาพที่ดีขึ้น

บรรณานุกรม

- Bahrami, Ali. 1999. **Object-Oriented Systems Development:Using The Unified Modeling Language**. Singapore : McGraw-Hill.
- Date, C.J. 1995. **An introduction to database systems**. 6th ed. Massachusetts : Addison-Wesley.
- Eriksson, Hans-Erik. and Penker, Magnus. 1998. **UML Toolkit**. New York : John Wiley & Sons.
- Hipsley, Paul. 1996. **Developing Client/ Server Applications with Oracle Developer/2000**. Indiana : Sams.
- Koch, George. and Loney, Kevin. 1997. **Oracle8: The complete Reference**. Singapore : Osborne McGraw-Hill
- Kroenke, David M. 1998. **Database Processing, Fundamentals, Design and Implementation**. 6th ed. New Jersey : Prentice-Hall.
- Lulushi, Albert. 1996. **Developing Oracle Forms Applications**. New Jersey : Prentice-Hall.
- Pressman, Roger S. 1997. **Software Engineering: A Practitioner's Approach**. Singapore : McGraw-Hill.
- Whitten, Jeffery L. 1997. **System Analysis and Design Methods**. 4th ed. Boston : Irwin.

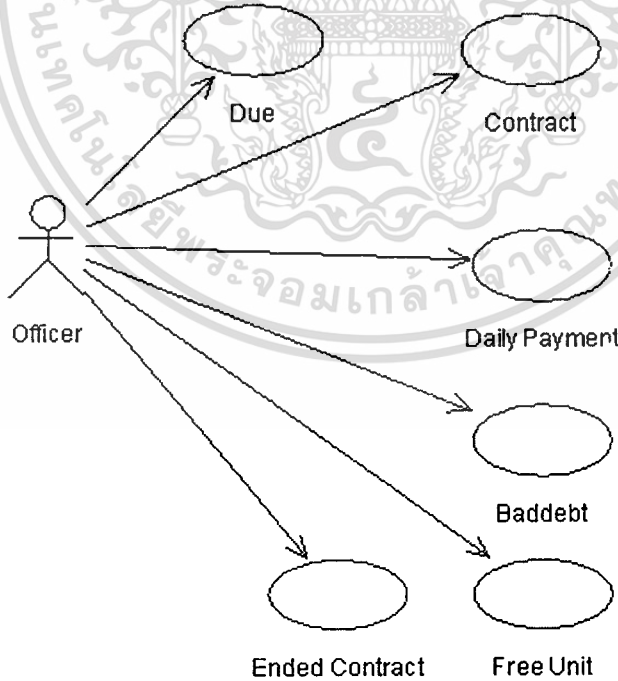
ภาคผนวก ก.

คู่มือพัฒนาระบบ

เนื่องจากการพัฒนาระบบงานระบบนี้เป็นการพัฒนาระบบงานแบบเชิงวัตถุ ดังนั้นสิ่งที่
เป็นคู่มือการพัฒนาระบบจึงจำเป็นต้องเป็นเนื้อหาที่เป็นเชิงวัตถุด้วย และเนื้อหาที่อยู่ในส่วนนี้จะเป็
เนื้อหาที่เป็นผลการออกแบบ ซึ่งจะประกอบด้วย ยูสเคสไดอะแกรมแสดงขอบเขตและบริการของ
ระบบงาน คลาสไดอะแกรม และส่วนการติดต่อกับผู้ใช้ ซึ่งอธิบายการทำงานภายในด้วยซีเควนซ์
ไดอะแกรม และสุดท้ายคือแอกทิวิตีไดอะแกรมแสดงการทำงานของเมอร์ธอร์ภายในแต่ละคลาส

ก.1 ยูสเคสไดอะแกรม

ยูสเคสไดอะแกรมจะเป็นไดอะแกรมหลักที่บอกถึงขอบเขตและบริการที่ระบบบัญชีลูกหนี้
เช่าทั้งหมดมีให้ซึ่งแสดงในรูปที่ ก.1

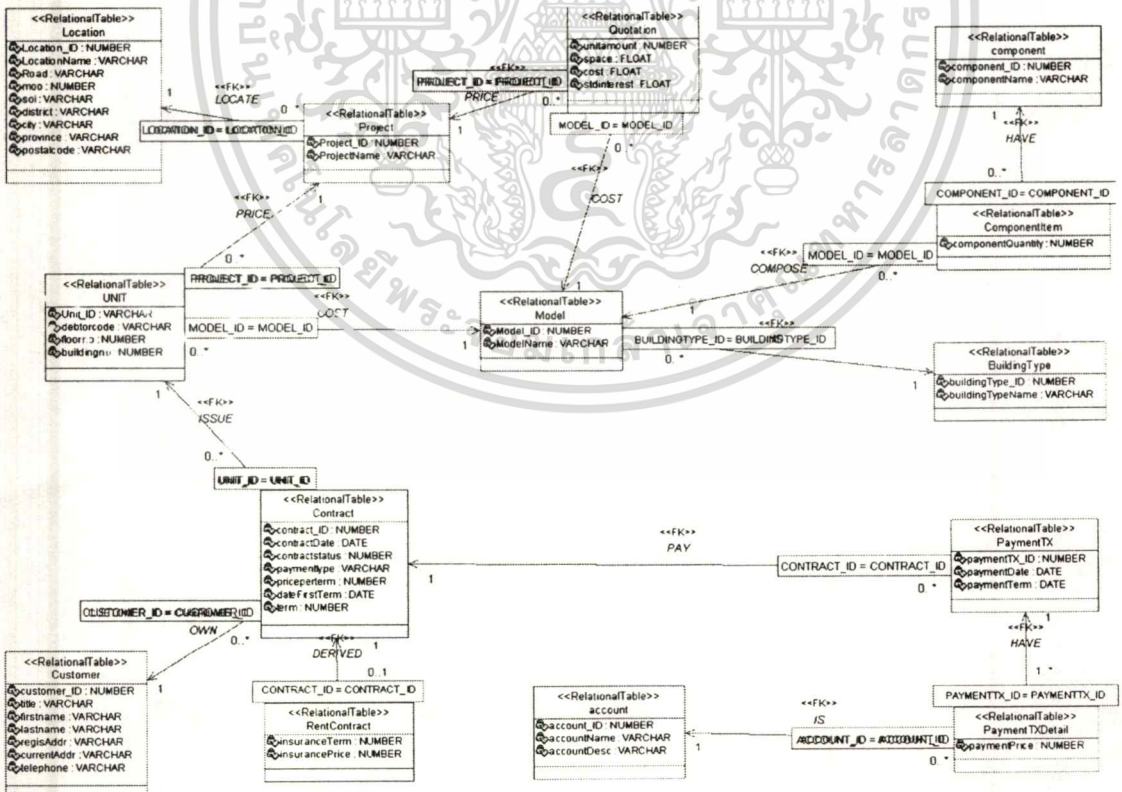


รูปที่ ก.1 ยูสเคสไดอะแกรมระบบบัญชีลูกหนี้

จากรูปจะเห็นไว้ระบบงานนี้มีบริการให้ผู้ใช้ ซึ่งเป็นผู้ใช้เพียงคนเดียวคือเจ้าหน้าที่ และมีบริการให้ 6 บริการด้วยกัน คือ บริการหนึ่งถึงกำหนดชำระ เป็นบริการให้ข้อมูลลูกหนี้ที่ถึงกำหนดจะต้องชำระในงวดเดือนปัจจุบัน บริการที่สองคือรายละเอียดสัญญา ให้ข้อมูลรายละเอียดสัญญาที่ลูกค้ากระทำการเช่าเหมาเช่าติ บริการที่สามคือ หน่วยอาศัยว่าง เป็นบริการที่ให้ข้อมูลหน่วยอาศัยที่ว่างของแต่ละโครงการเพื่อให้ลูกค้าสามารถเข้าเลือกเพื่อทำสัญญาได้ บริการที่สี่คือบริการการชำระเงินประจำวัน เป็นบริการเพื่อแสดงรายงานการชำระเงินค่าเช่าและค่าใช้จ่ายอื่น ๆ ของแต่ละเขตควบคุม บริการที่ห้าคือบริการหนี้เสีย เป็นบริการที่ให้ข้อมูลลูกหนี้ที่ค้างชำระ และคำนวณยอดหนี้ ตามโครงการ และบริการสุดท้ายคือบริการหนี้หมดอายุ เป็นบริการที่ให้ข้อมูลลูกหนี้ที่สัญญาจะสิ้นสุดในงวดถัดไป เพื่อให้เจ้าหน้าที่สามารถดำเนินการได้ และรายงานหนี้ที่ยังค้างชำระของลูกค้ารายนั้น ๆ ด้วย

ก.2 คลาสไดอะแกรม

คลาสไดอะแกรมจะเป็นหัวใจของการออกแบบในระบบเชิงวัตถุ เพราะเป็นโครงสร้างพื้นฐานของระบบบัญชีลูกหนี้ โดยมีโครงสร้างดังรูป ก.2



รูปที่ ก.2 คลาสไดอะแกรมระบบบัญชีลูกหนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป ก.2 จะเป็นคลาสไดอะแกรมแสดงเฉพาะแอททริบิวต์ของคลาสเท่านั้น โดยจะอธิบายถึงคลาสที่มีอยู่ทั้งหมด โดยมีคลาสแต่ตัวมีรายละเอียดคือ คลาสโลเคชัน เป็นคลาสที่เป็นตัวแทนของเขตควบคุม หรือสช.(สำนักงานเคหะชุมชน) ภายในคลาสก็จะมีแอททริบิวต์ที่บอกถึงรหัสของเขต ชื่อเขต และตำแหน่งที่ตั้งของเขตนั้น โดยรูป ก.3 จะแสดงแอททริบิวต์ และเมธอดของคลาสโลเคชัน โดยบริการเมธอดที่คลาสนี้มีให้ ก็เป็นเพียงบริการเรียกข้อมูลของคนออกมาแสดงเท่านั้น

```
Location (location_id, locationname, road, moo, soi, district, city, province, postalcode)
```

```
- query(location_id, locationname, road, moo, soi, district, city, province, postalcode)
```

รูปที่ ก.3 โครงสร้างคลาสโลเคชัน

คลาสต่อมาคือคลาสโปรเจกต์ จะเป็นคลาสที่เป็นตัวแทนของโครงการก่อสร้างของกรมเคหะแห่งชาติ โดยแต่ละโครงการจะต้องสังกัดเขตควบคุมใดเขตหนึ่ง ดังนั้นความสัมพันธ์ระหว่างคลาสโลเคชัน และคลาสโปรเจกต์จึงเป็นแบบ หนึ่งต่อหลายข้อมูล เพราะเขตควบคุมหนึ่งสามารถมีหลายโครงการ ส่วนแอททริบิวต์ภายในก็จะเป็นเพียงชื่อโครงการเท่านั้น ซึ่งมีโครงสร้างดังรูป ก.4 โดยบริการที่คลาสนี้มีให้ก็คือ แสดงข้อมูลภายในคลาสตนเป็นบริการแรก บริการต่อมาคือสร้างรายงานหน่วยพักอาศัยว่าง และบริการสุดท้ายคือบอกว่าโครงการที่สอบถามมานั้นอยู่ในเขตใด

```
Project(project_id, projectname, location_id)
```

```
- query(project_id, projectname, location_id)
```

```
- avail( project_id, projectname, location_id, unitcount, rentcount, freeunit, dueamount)
```

```
- getLocationid( project_id) return number;
```

รูปที่ ก.4 โครงสร้างคลาสโปรเจกต์

คลาสชุดต่อมาจะเป็นคลาสชุดที่ใช้อธิบายแบบบ้านที่การเคหะมีในสังกัด โดยจะเริ่มที่คลาสโมเดล จะเป็นคลาสที่บอกถึงแบบต่าง ๆ ที่การเคหะมีให้ลูกค้าเลือก โดยแอททริบิวต์ภายในก็จะบอกถึงชื่อโมเดลนั้นเท่านั้น ดังมีโครงสร้างในรูป ก.5

```
Model( model_id, modelname, buildingtype_id)
```

```
- gettype(model_id) return number
```

```
- getname(model_id) return varchar;
```

รูปที่ ก.5 โครงสร้างคลาสโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยบริการที่คลาสนี้มีให้ก็คือ บริการสอบถามว่าโมเดลที่ถามมานั้นเป็นชนิดอาคารชนิดใด เช่น บ้านทรงสเปนก็จะมีชนิดเป็นบ้านเดี่ยวเป็นต้น และบริการที่สองก็คือบริการสอบถามชนิดอาคารเช่นกันแต่เป็นการถามชื่อ ในขณะที่บริการแรกจะตอบเป็นรหัสของชนิดอาคาร

คลาสต่อมาจะเป็นคลาสบิวด์ิงไทป์ คลาสนี้จะป็นคลาสที่กำหนดว่าชนิดอาคารที่การเคหะ มีให้บริการนั้นมีชนิดใดบ้าง โดยแต่ละแบบบ้านก็จะมีชนิดเป็นของตัวเอง ดังนั้นจะเห็นว่าความสัมพันธ์ระหว่างโมเดลกับบิวด์ิงไทป์จะเป็นแบบ หลายต่อหนึ่ง แอทริบิวทภายในบิวด์ิงไทป์ก็เพียงบอกชื่อของชนิดอาคารเท่านั้น ดังแสดงในโครงสร้างรูปที่ ก.6 ส่วนบริการที่มีให้นั้นก็ป็นบริการสอบถามข้อมูลทั้งหมด และบริการที่สองก็คือบริการถามชื่อชนิดอาคารเท่านั้น โดยให้ข้อมูลมาเป็นรหัสชนิดอาคาร และตอบเป็นชื่อ

```
Buildingtype ( buildingtype_id, buildingtypename)
-query(buildingtype_id, buildingtypename)
-getname(buildingtype_id) return varchar
```

รูปที่ ก.6 โครงสร้างคลาสบิวด์ิงไทป์

คลาสชุดต่อมาจะเป็นคลาสที่อธิบายลักษณะของแต่ละแบบบ้านว่ามีส่วนประกอบอย่างไรบ้าง เช่นมีกี่ห้องนอน กี่ห้องน้ำเป็นต้น โดยจะใช้คลาสคอมโพเนนท์ในการกำหนดส่วนประกอบ เช่นห้องนอน หรือห้องน้ำ และใช้คลาสคอมโพเนนท์ไอเทมส์ในการระบุจำนวนว่ามีจำนวนเท่าใด และบอกด้วยว่าจำนวนนั้น ๆ เป็นของบ้านแบบใด

ดังนั้นโดยภาพแล้ว คลาสคอมโพเนนท์ไอเทมจะทำหน้าที่เป็นแอตโซซิเอตคلاسระหว่างคลาสโมเดลกับคลาสคอมโพเนนท์ โดยภายในคลาสคอมโพเนนท์ก็จะมีแอทริบิวท์ที่บอกว่าส่วนประกอบนั้นมีรหัสอะไร และมีชื่ออะไร ซึ่งมีโครงสร้างดังรูป ก.7 สำหรับบริการที่คลาสคอมโพเนนท์มีให้ ก็มีเพียงบริการสอบถามชื่อรหัสที่ให้นั้นมีชื่อเป็นส่วนประกอบว่าอะไร

```
Component(component_id, component)
- getname(component_id)
```

รูปที่ ก.7 โครงสร้างคลาสคอมโพเนนท์

ส่วนโครงสร้างของคลาสคอมโพเนนท์ไอเทมซึ่งเป็นแอตโซซิเอตคلاسระหว่างคลาสโมเดลกับคลาสคอมโพเนนท์ นั้นก็จะมีส่วนประกอบคือเอกลักษณ์ของทั้งสองคลาสมาเป็นคอมไบน์คีย์ของคน และมีส่วนข้อมูลที่เป็นส่วนอธิบายความสัมพันธ์คือจำนวนส่วนประกอบนั่นเอง ดังมีโครง

สร้างในรูป ก.8 และบริการที่มีให้ก็นี้ก็เป็นบริการที่สร้างรายการส่วนประกอบของแต่ละโมเดล โดยจะต้องอาศัยความร่วมมือกับคลาสคอมโพเนนท์ด้วย

```
Componentitem(model_id, component_id, componentquantity)
- getdetail(model_id, component_id, componentquantity, componentname, id model_id)
```

รูปที่ ก.8 โครงสร้างคลาสคอมโพเนนท์ไอเทม

คลาสต่อไปจะเป็นคลาสที่เป็นแอตทริบิวต์เหมือนกัน คือคลาสโควเทชัน ออกแบบมาเพื่อให้เป็นคลาสที่บอกราคาของหน่วยอาศัย โดยลักษณะคือจะบอกราคาของแต่ละแบบบ้าน แต่แต่ละแบบบ้านนั้นก็จะมีราคาเปลี่ยนไปตามโครงการ ดังนั้น คลาสนี้จึงต้องเป็นแอตทริบิวต์คลาสระหว่างคลาสโมเดล และคลาสโปรเจกต์ แอททริบิวต์ที่มีเพิ่มเติมจากการเอาคีย์ของคลาสต้นทางมาเป็นคอมโพเนนต์ของคลาสตนเองก็คือ แอททริบิวต์ที่อธิบายลักษณะของแบบก็คือ พื้นที่, ราคา, อัตราดอกเบี้ย และจำนวนหน่วย ดังในรูปที่ ก.9 ส่วนบริการที่มีให้ก็นี้ก็เป็นบริการสอบถามข้อมูลทั้งหมดนั่นเอง

```
Quotation(model_id, project_id, unitamount, space, cost, stdinterest)
- query(model_id, project_id, unitamount, space, cost, stdinterest)
```

รูปที่ ก.9 โครงสร้างคลาสโควเทชัน

จบจากคลาสชุดที่แล้ว ซึ่งเป็นชุดของคลาสที่อธิบายลักษณะของแบบบ้านแล้ว คลาสต่อไปจะเป็นคลาสที่เป็นตัวหลักในการทำธุรกรรมจริง ๆ ซึ่งคลาสแรกในชุดนี้ก็คือยูนิต ดังรูป ก.10

```
Unit(unit_id, debtorcode, floorno, buildingno, model_id, project_id)
-query(unit_id, debtorcode, floorno, buildingno, model_id, project_id)
-avail(id project_id) return unit number;
-rent(id project_id) return unit number;
-dueamount(id project_id) return amount number;
-availdetail(unit_id, title, firstname, lastname, rentrate, method)
-freeunit(id number) return number;
-freeunitlist(unit_id, buildingtype_id, buildingtypename, project_id, floorno, buildingno, model_id, modelname)
- getprojectid(id unit_id) return number;
```

รูปที่ ก.10 โครงสร้างคลาesyunit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสชนิดเป็นคลาสที่แต่ละข้อมูลหลายถึงหน่วยอาศัยแต่ละหน่วย ดังนั้น จึงสัมพันธ์กับคลาสโปรเจกต์ แบบ หลายต่อหนึ่ง เพราะแต่ละโครงการก็ย่อมต้องมีหลายหน่วยอาศัย และแอทริบิวต์ภายในก็จะอธิบายว่าหน่วยอาศัยนั้นมีรหัสอะไร มีรหัสการเป็นหนี้ซึ่งเป็นรหัสภายในของการเคหะแห่งชาติว่าจะไร หน่วยอาศัยนั้นอยู่ที่ชั้นใด อาคารใด เป็นแบบใด และอยู่ในโครงการใด ดังมีโครงสร้างในรูปที่ ก.10 ส่วนบริการที่มีให้นั้นก็มีความมากมาย เพราะเป็นคลาสที่อยู่ระหว่างหลายคลาส คือมีบริการ รายงานข้อมูลหน่วยอาศัยทั้งหมด บริการรายงานจำนวนหน่วยอาศัยที่มีทั้งหมด หน่วยอาศัยที่ถูกเช่า จำนวนยอดค่าเช่า สร้างรายงานหน่วยอาศัยถึงกำหนดชำระ รายงานจำนวนหน่วยอาศัยว่าง สร้างรายงานหน่วยอาศัยว่าง และบริการสอบถามว่าหน่วยอาศัยนั้น ๆ อยู่ในโครงการใด

คลาสต่อมาก็คือคลาสคอนแทรก หรือสัญญา ซึ่งทำหน้าที่เป็นแอตโซซิเอตคลาสระหว่างคลาสหน่วยอาศัย กับคลาสลูกค้า เพราะทั้งลูกค้าและหน่วยอาศัยมีความสัมพันธ์แบบ หลายต่อหลาย เพราะ ลูกค้าแต่ละรายก็สามารถมีได้หลายห้อง และ แต่ละห้องก็สามารถมีคนเช่าหลาย ๆ คนได้ แต่คนละช่วงเวลา ดังนั้นแอทริบิวต์ของคลาสสัญญาก็จะมีข้อมูลเป็น รายละเอียดการเช่า นั่นคือวันที่ทำสัญญา สถานะสัญญา วิธีการชำระเงิน ค่าเช่า งวดแรก และจำนวนงวด ดังแสดงในรูป ก.11

```
Contract(contract_id, contractdate, unit_id, contractstatus, paymenttype, priceperterm,
datefirstterm, term, customer_id)
-amount(unitid number)return number;
-status(unitid number)return number;
-getcusid(unitid number)return number;
-getmethod(unitid number)return varchar;
-querydetail(contract_id, contractdate,contractstatus, paymenttype, priceperterm, datefirstterm,
term, lastterm, insuranceterm, insuranceprice, unit_id, title,firstname,lastname id unit_id);
-getunitid(contract_id)return number;
-querybaddebt(contract_id, unit_id, rentdebt, insdebt, title, name, lastname, totalterm,
paidterm, badrentterm, status, id project_id, startmonth, startyear, nowdate)
-endquery(unit_id, contract_id, datefirstterm, priceperterm, status, rentdebt, insdebt, id title,
name, lastname, projectid, current date)
```

รูปที่ ก.11 โครงสร้างคลาสคอนแทรก

สำหรับบริการที่มีให้นั้นก็มีความมาย ตั้งแต่สอบถามค่าเช่า สอบถามสถานะสัญญา สอบถามลูกค้าที่เป็นเจ้าของสัญญา สอบถามวิธีการชำระเงิน สร้างรายงานรายละเอียดสัญญา สอบถามหน่วยอาศัยที่อยู่ในสัญญานั้น สร้างรายงานหนี้เสีย และสร้างรายงานหนี้หมดอายุสัญญา

คลาสต่อมาที่มีความสัมพันธ์กับคลาสคอนแทรคอย่างใกล้ชิด ก็คือคลาสเรนท์คอนแทรค ที่สัมพันธ์อย่างใกล้ชิดก็เพราะว่าในการออก แบบเชิงวัตถุ นั้นสองคลาสนี้มีความสัมพันธ์แบบคลาสแม่ และคลาสย่อย เมื่อผ่านมาถึงการออกแบบนั้นก็กำหนดให้สองคลาสนี้เป็นสองคลาสที่แยกจากกัน และมีความสัมพันธ์แบบหนึ่งต่อหนึ่ง โดยคลาสคอนแทรคจะอธิบายลักษณะทั่วไปของสัญญา ส่วนคลาสเรนท์คอนแทรค จะเก็บข้อมูลที่เป็นลักษณะพิเศษของสัญญาเช่า ซึ่งในกรณีนี้ก็คือรายละเอียดการประกันการเช่านั่นเอง ดังนั้นรายละเอียดแอทริบิวภายในคลาสนี้ก็จะมี จำนวนงวดการจ่ายค่าประกัน และค่าประกันแต่ละงวด ดังแสดงในรูป ก.12 ส่วนบริการที่มีให้นั้นก็จะเป็นบริการสอบถามค่าประกัน และบริการสอบถามจำนวนงวดประกันเท่านั้น

```
Rentcontract(contract_id, insuranceterm, insuranceprice)
```

```
-getterm(id number) return number;
```

```
-getprice(id number) return number;
```

รูปที่ ก.12 โครงสร้างคลาสเรนท์คอนแทรค

คลาสต่อมาก็คือคลาสคัสทอเมอร์ เป็นคลาสที่เป็นตัวแทนของลูกค้า เพราะแต่ละข้อมูลของคลาสนี้ก็หมายถึงลูกค้าแต่ละรายนั่นเอง ดังนั้นความสัมพันธ์กับคลาสคอนแทรค ก็จะเป็น หนึ่งต่อหลายดังที่กล่าวมาแล้ว และแอทริบิวของคลาสนี้ก็จะเป็นรายละเอียดของลูกค้า เช่นชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ ดังรูป ก.13

```
Customer(customer_id, title, firstname, lastname, regisaddr, currentaddr, telephone)
```

```
-gettitle(cusid number) return varchar
```

```
-getname(cusid number) return varchar
```

```
-getlastname(cusid number) return varchar
```

```
-query(customer_id, title,firstname,lastname, regisaddr,currentaddr, telephone)
```

รูปที่ ก.13 โครงสร้างคลาสคัสทอเมอร์

บริการที่คลาสนี้มีให้ก็จะเป็นบริการสอบถามข้อมูลภายในคลาสตนเอง นั่นคือสอบถามค่านำหน้าชื่อ สอบถามชื่อ นามสกุล และสร้างรายงานข้อมูลลูกค้าทั้งหมดที่มี เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสต่อมา จะเป็นคลาสตระกูลที่พยายามอธิบายการชำระเงินของการเช่า โดยจะมีคลาสที่เป็นส่วนประกอบคือ คลาสเพเมนต์ แต่ละข้อมูลของคลาสนี้จะหมายถึงการชำระเงินแต่ละงวด ดังนั้นจึงสัมพันธ์กับคลาสคอนแทรกแบบ หลายต่อหนึ่ง เพราะสัญญาหนึ่ง ๆ จะมีการชำระเงินหลายครั้งนั่นเอง และแอทริบิวทภายในคลาสนี้ก็จะเข้าไปเพื่ออธิบายการชำระเงินแต่ละครั้ง ดังนั้นจึงมีข้อมูลคือ รหัสการชำระเงิน วันที่จ่าย และงวดที่จ่าย และมีโครงสร้างดังรูปที่ ก.14

```
Paymenttx(paymenttx_id, contract_id, paymentdate, paymentterm)
```

```
-getdate(txid) return date;
```

```
-getcontractid(txid) return number;
```

```
-getterm(txid) return date;
```

```
-lastpaidterm(contract_id) return date;
```

```
-lastpaidins(contract_id) return date;
```

รูปที่ ก.14 โครงสร้างคลาสเพเมนต์

สำหรับบริการที่คลาสนี้มีให้ก็เป็นบริการที่สอบถามข้อมูลเกี่ยวกับการชำระเงิน คือสอบถามวันที่ชำระเงิน สอบถามรหัสสัญญาที่ชำระเงิน สอบถามงวดที่จ่าย สอบถามว่าสัญญานั้น ๆ จ่ายค่าเช่า และค่าประกันครั้งสุดท้ายเมื่อใด

คลาสต่อมาก็คือคลาสเพเมนต์ดีเทล ซึ่งเป็นแต่รายละเอียดการชำระเงิน ว่าการจ่ายเงินแต่ละครั้งนั้นเป็นค่าอะไรบ้าง แต่ละข้อมูลของคลาสนี้ก็คือ รายการแต่ละรายการนั่นเอง ดังนั้น คลาสนี้จึงสัมพันธ์กับคลาสเพเมนต์แบบ หลายต่อหนึ่ง เพราะการจ่ายเงินแต่ละครั้งสามารถมีได้หลายค่าใช้จ่าย และจริง ๆ แล้วคลาสนี้เป็นแอต โซซิเอตระหว่างคลาสเพเมนต์ และคลาสเอคเคาท์ ดังนั้นแอทริบิวทก็จะมามีคอมไบน์คีย์ที่ได้จากคลาสนี้ และจำนวนเงินที่จ่ายแต่ละรายการ ดังมีโครงสร้างแอทริบิว และบริการดังในรูป ก.15

```
Paymenttxdetail(paymenttx_id, account_id, paymentprice)
```

```
-dailyquery(paymenttx_id, paymentdate, contract_id, unit_id, accountname, amount, term, location_id, querydate, querymonth, queryyear, id location_id, rentfilter)
```

```
-isrent(txid) return number;
```

```
-isins(txid) return number;
```

รูปที่ ก.15 โครงสร้างคลาสเพเมนต์ดีเทล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับบริการที่มีให้ก็นั้นก็เป็นบริการสอบถามข้อมูลในรายการว่า แต่ละการจ่ายเงินนั้น มีค่าเช่า อยู่หรือไม่ และมีค่าประกันอยู่หรือไม่

และคลาสสุดท้ายในชุดการจ่ายเงิน และคลาสสุดท้ายของระบบก็คือคลาสแอดเคาท์ โดยแนวคิดแล้ว คลาสนี้เป็นคลาสที่ออกแบบมาเพื่อเก็บรหัสบัญชีการชำระเงินแต่ละรายการ หรืออีกนัยหนึ่งก็คือคำอธิบายรายการค่าใช้จ่ายนั่นเอง ว่าเป็นค่าใช้จ่ายค่าอะไร เช่นค่าเช่า ค่าน้ำ เป็นต้น ดังนั้นแอทริบิวต์ ก็จะเป็นข้อมูลสำหรับการอธิบายธรรมดา นั่นคือรหัสบัญชี ชื่อบัญชี และคำอธิบาย ดังมีโครงสร้างในรูป ก.16

```
Account(account_id, accountname, accountdesc)
-getname(account_id)return varchar
```

รูปที่ ก.16 โครงสร้างคลาสแอดเคาท์

สำหรับบริการที่มีให้ก็นั้นก็เป็นบริการสอบถามชื่อบัญชี ว่ารหัสบัญชีที่ให้นั้นมีชื่อบัญชีว่าอะไร โดยตอบกลับเป็นชื่อบัญชื่อนั้น

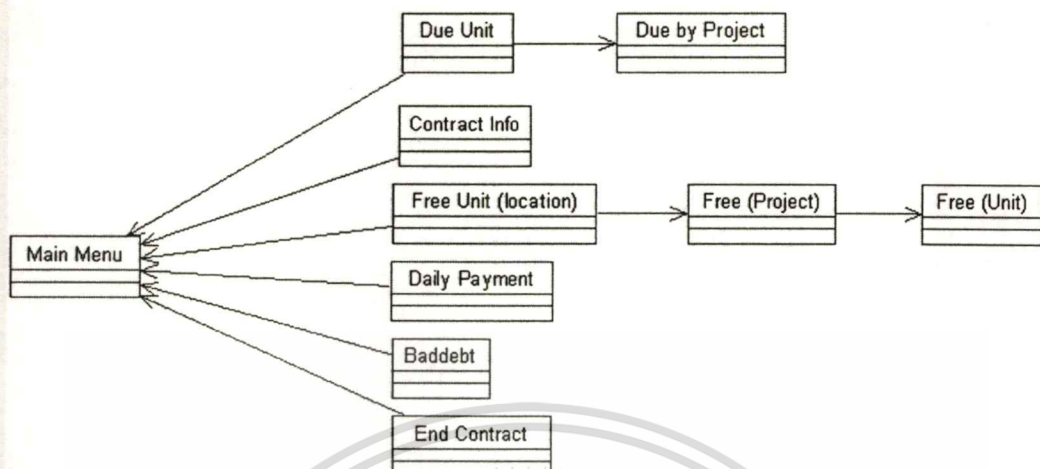
ก.3 ซีควেনซ์ไดอะแกรม

ซีควেনซ์ไดอะแกรมจะทำหน้าที่บอกถึงการสื่อสารที่เกิดขึ้นในแต่ละระบบย่อย ดังนั้นหน้าหนึ่งหน้าต่างก็จะมีหนึ่งซีควেনซ์ไดอะแกรม ที่แสดงการทำงานร่วมมือกันของคลาสต่าง ๆ โดยรูปที่ ก.17 จะแสดงคลาสไดอะแกรมของหน้าต่างที่มีทั้งหมดในระบบ โดยแต่ละคลาสจะเป็นตัวแทนของหน้าต่างของโอราเคิล เดเวลลอปเปอร์ฟอรัม ซึ่งนับเป็นออบเจกต์อย่างชนิดหนึ่งที่มีคุณสมบัติเป็นแอทริบิวต์ และมีเมธอดเป็นกิจกรรมที่ฟอรัมนั้นรองรับ เช่นการกดปุ่ม เป็นต้น โดยการให้บริการแต่ละบริการก็จะต้องไปประสานงานกับคลาสที่เป็นคลาสกิจกรรมทางธุรกิจจริง ๆ

จากรูป จะเห็นว่าระบบหน้าต่างนั้นมีหน้าต่างหลักเป็นตัวเชื่อมโยงทั้งหมด โดยหน้าต่างระบบย่อยหนึ่งถึงกำหนดชำระ จะมีสองระดับ ระบบย่อยหน่วยพักอาศัยว่างจะมีสามระดับ ส่วนระบบย่อยอื่น ๆ จะมีเพียงระดับเดียว ดังนั้นรวมทั้งหมดแล้วจะมี 10 หน้าต่าง (นับรวมหน้าต่างหลักด้วย) ทำให้จะมีซีควেনซ์ไดอะแกรมทั้งหมด 9 ซีควেনซ์ไดอะแกรม เพราะไม่นับหน้าต่างหลักซึ่งไม่มีกิจกรรมทางธุรกิจเกิดขึ้นจริง ๆ

โดยส่วนต่อไปนี้จะแสดงให้เห็นหน้าต่างที่ใช้งานจริง เทียบกับซีควেনซ์ไดอะแกรมของหน้าต่างนั้น เพื่อให้เห็นกิจกรรมที่เกิดขึ้นภายใน และถ้าต้องการทราบส่งที่เกิดขึ้นในแต่ละคำร้องที่สื่อสารถึงกันนั้น ก็สามารถดูได้ในส่วนต่อไปซึ่งจะให้รายละเอียดภายในแต่ละเมธอดของคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.17 คลาสไดอะแกรมของหน้าต่างอินเทอร์เน็ตเฟส

หน้าต่างแรกที่จะแสดงให้เห็นก็คือหน้าต่างหนึ่งถึงกำหนดชำระตามเขต ซึ่งแสดงในรูป

ก.18 และ ซีเควนซ์ไดอะแกรมในรูป ก.19

WINDOW1

ลูกหนี้ถึงกำหนดชำระ

เขตควบคุม

ชื่อเขต	bangkae region	ถนน	petkasame
หมู่	5	ซอย	thainamtip
ตำบล	bangwa	อำเภอ	pasrichaoroen
จังหวัด	bangkok	รหัสไปรษณีย์	10160

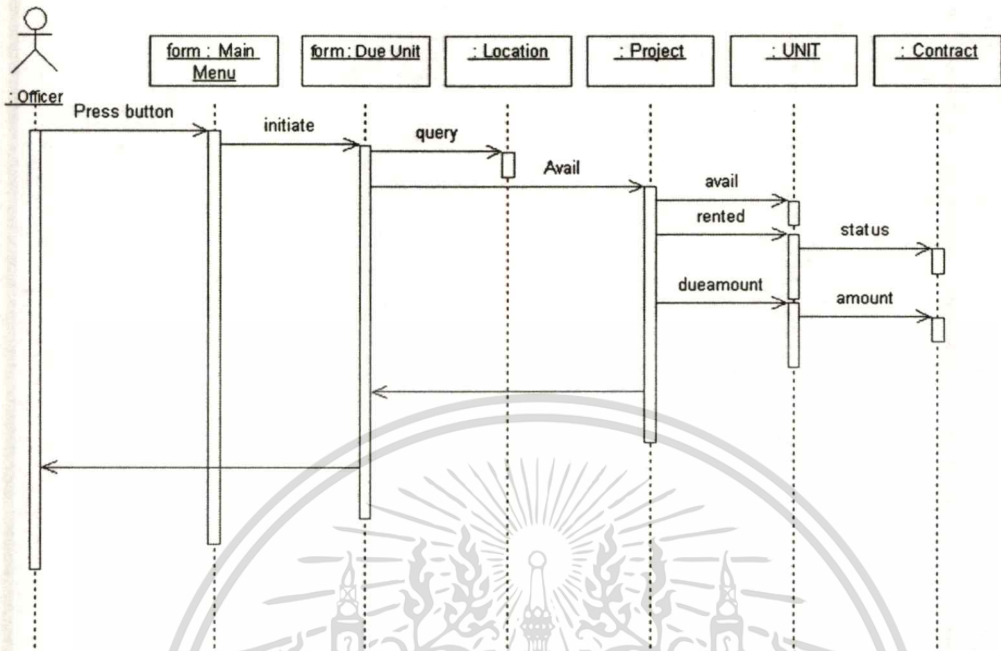
สรุปข้อมูลกำหนดชำระของโครงการ

รหัสโครงการ	ชื่อโครงการ	จำนวนหน่วยรวม	หน่วยที่เช่า	รวมยอดค่าเช่า
1	Happy Village	11	5	5200
3	Delaware Project	8	0	0
4	Minor Hill	0	0	0

พิมพ์ทุกหน่วยควบคุม

รูปที่ ก.18 หน้าต่างระบยย่อหนึ่งถึงกำหนดชำระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.19 ซีควেনซ์ไดอะแกรมของหน้าต่างระบบถึงกำหนดชำระตามเขตปกครอง

จากรูปที่ ก.18 จะเห็นว่าผู้ใช้สามารถเลือกเขตที่ต้องการได้ และหน้าต่างจึงแสดงผลตามโครงการ ดังนั้นในรูปที่ ก.19 จะเห็นได้ว่าแอปเจคฟอร์มจะติดต่อไปที่แอปเจค โลกชั้นเพื่อให้แสดงข้อมูลที่มีทั้งหมดเพื่อให้ผู้ใช้สามารถเลือกได้ และเมื่อผู้ใช้เลือกแล้ว ก็ส่งค่าที่ผู้ใช้เลือกไปที่แอปเจคโปรเจคเพื่อให้เห็นรายการ โดยร้องขอบริการแสดงรายการ จากนั้นแอปเจคโปรเจค ก็จะสอบถามไปยังแอปเจคยูนิตเพื่อให้เห็นว่าหน่วยอาศัยทั้งหมดของโครงการนั้นมีกี่หน่วยโดยเรียกใช้บริการ “Avail” และถามไปอีกว่าโครงการนี้มีหน่วยขอเช่าถูกเช่าไปกี่หน่วยโดยเรียกใช้บริการ “Rented” ซึ่งแอปเจคยูนิตก็ต้องสอบถามไปยังแอปเจคคอนแทรคต่อว่าหน่วยอาศัยหนึ่ง ๆ นั้นมีสถานะสัญญาเป็นอย่างไร ถูกเช่าอยู่หรือไม่ หลังจากนั้นแอปเจคโปรเจคก็จะถามไปยังแอปเจคยูนิตว่าค่าเช่ารวมกันของโครงการนั้นเป็นเท่าใด ซึ่งแอปเจคยูนิตก็ต้องไปถามแอปเจคคอนแทรคว่าแต่ละหน่วยอาศัยที่ถูกเช่านั้นมีค่าเช่าเป็นเท่าใด ซึ่งเมื่อแอปเจคโปรเจคได้ข้อมูลหมดแล้วก็สามารถสร้างเป็นชุดข้อมูลคืนไปให้ฟอร์มได้

หน้าต่างต่อไปจะเป็นหน้าต่างรายละเอียดของระบบย่อยหนึ่งถึงกำหนดชำระ ซึ่งก็คือการดูข้อมูลแบ่งตามโครงการนั่นเอง โดยที่ที่หน้าต่างก่อนหน้านี้ เมื่อผู้ใช้สนใจจะดูข้อมูลโครงการใดเป็นการเฉพาะเจาะจง ผู้ใช้จะต้องเลือกโครงการที่ต้องการดูรายละเอียดก่อน และกดปุ่ม “รายละเอียด” ซึ่งหน้าต่างก่อนหน้านี้ก็จะส่งข้อมูลเป็นรหัสโครงการที่ต้องการมาให้หน้าต่างนี้ ดังจะมีรูปหน้าต่างอยู่ที่รูป ก.20 และซีควেনซ์ไดอะแกรมในรูป ก.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WINDOW1

รายละเอียดลูกค้าที่ถึงกำหนดชำระของโครงการ

โครงการ

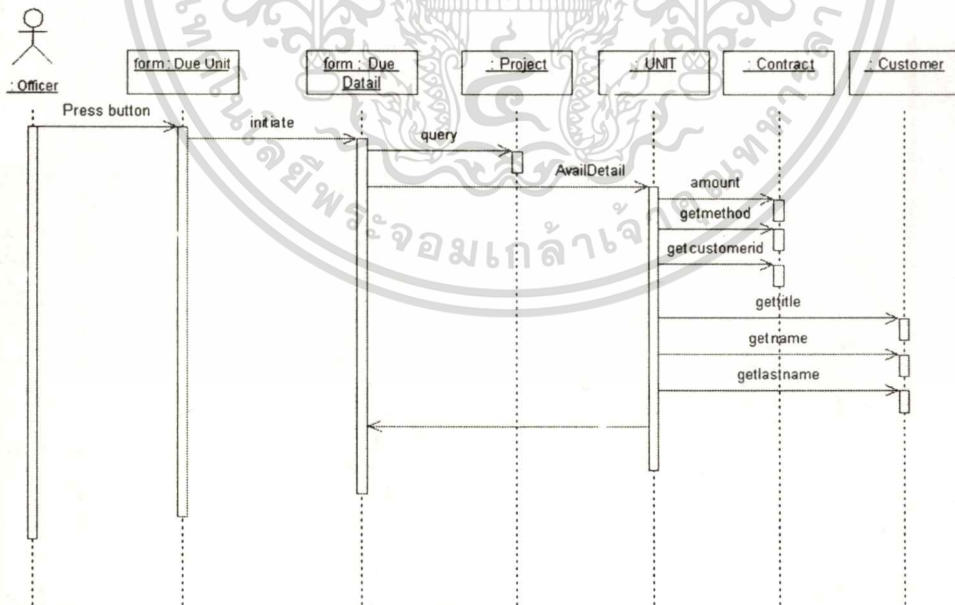
รหัสโครงการ: หน่วยเช่า: 5

ชื่อโครงการ: Happy Village ยอดค่าเช่ารวม: 5200

จำนวนหน่วยรวม: 11

รหัสหน่วยอาศัย	ท่าน่าชื่อ	ชื่อลูกค้า	นามสกุล	ค่าเช่า	วิธีชำระค่าเช่า
100001	Mr.	Frijob	Capra	1000	cash
100002	Mr.	Alexandre	Dumas	1000	cash
100003	Mrs.	Michael	Crichton	1000	cash
100005	Mr.	Frijob	Capra	1000	bank
101001	Mr.	Victor	Hugo	1200	bank

รูปที่ ก.20 อินเทอร์เฟซของระบบย่อยนี้ถึงกำหนดชำระตามโครงการ



รูปที่ ก.21 ซีควেনซ์ไดอะแกรมของหน้าต่างนี้ถึงกำหนดชำระตามโครงการ

จากรูปที่ ก.20 จะเห็นได้ว่าผู้ใช้จะต้องสามารถเปลี่ยนโครงการที่ต้องการดูได้ และข้อมูลที่ด้านล่างจะเปลี่ยนตาม ดังนั้นสิ่งแรกที่ต้องทำ ซึ่งแสดงในรูป ก.21 ก็คือฟอร์มจะต้องเรียกไปที่ออปเจกต์อื่นเป็นเอกสารที่ส่งงานไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่นับญาติให้ไปใช้ประโยชน์การค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แจกโปรเจก เพื่อให้แสดงข้อมูลโครงการทั้งหมดที่ตรงกับเขตควบคุมที่เลือกไว้ในหน้าต่างแรก จากนั้นก็ร้องขอไปยังออปเจกยูนิคเพื่อขอให้สร้างชุดข้อมูลรายละเอียดหน่วยอาศัยว่างให้ ดังนั้นออปเจกยูนิคซึ่งเป็นผู้ประสานงานต้องทำก็คือ ไปสอบถามออปเจกคอนแทรก เพื่อสอบถามสถานะของสัญญาว่าเป็นการเช่าหรือไม่ ถ้าใช่ก็สอบถามว่าชำระเงินอย่างไร และค่าเช่าเป็นเท่าใด เพื่อนำมาสร้างเป็นรายงาน จากนั้นก็สอบถามรหัสลูกค้าที่เป็นผู้ทำสัญญาเพื่อนำรหัสไปถามออปเจกคัสทอเมอร์ต่อไป เพื่อจะได้ข้อมูลรายละเอียดของลูกค้า ซึ่งก็คือชื่อและนามสกุลนั่นเอง เพียงเท่านี้ก็ได้ข้อมูลครบตามต้องการ ดังนั้นออปเจกยูนิคก็สามารถส่งชุดข้อมูลที่สร้างได้คืนไปให้ฟอร์มเพื่อแสดงผลต่อไป

หน้าต่างต่อไปจะเป็นหน้าต่างของระบบย่อยรายละเอียดสัญญา ซึ่งมีหน้าตาของหน้าต่างดังรูป ก.22 และซีเควนซ์โคอะแกรมดังรูป ก.23

รายละเอียดสัญญา

ลูกค้า

รหัสลูกค้า	100	ที่อยู่ปัจจุบัน	47/1 Petkasame
ชื่อนามสกุล	Mr. Frijob Capra	ที่อยู่ตามทะเบียน	64/2 Sukhumwit
		โทรศัพท์	3488383

ก่อนหน้า ต่อไป

รายละเอียดสัญญา

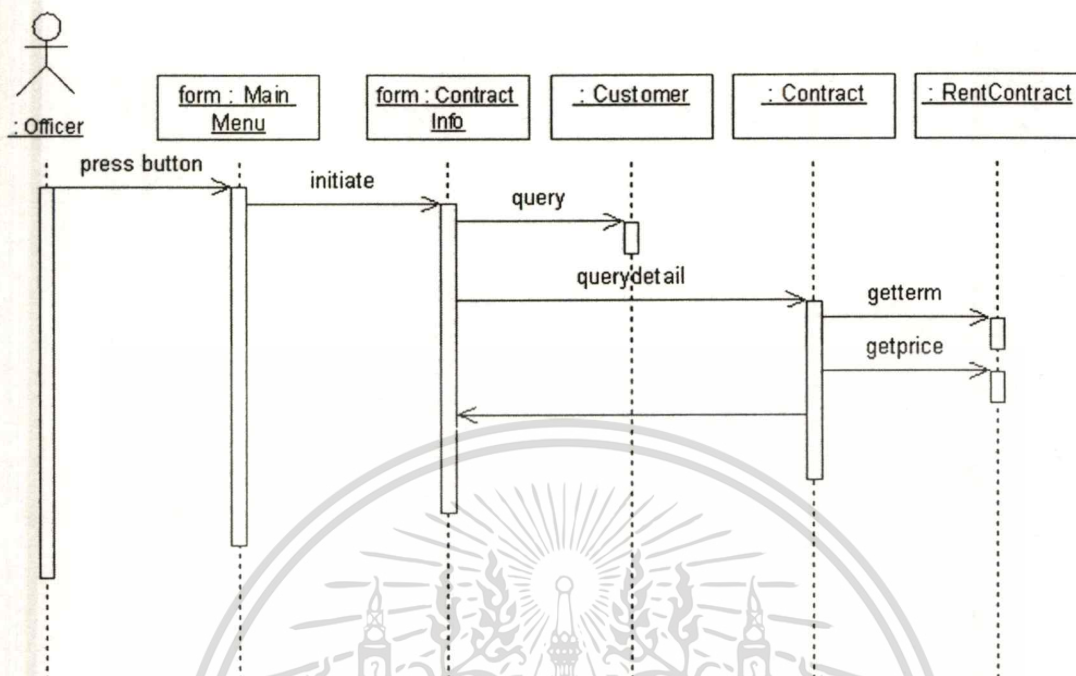
รหัสสัญญา	1001000	รหัสหน่วยอาศัย	100001
วันที่ทำสัญญา	01 ม.ค. 1999	ค่าเช่า	1000
สถานะสัญญา	1	จำนวนงวดประกัน	3
วิธีชำระเงิน	cash	ค่าประกัน	21
งวดแรก	01 ก.พ. 1999		
งวดสุดท้าย	01 มี.ค. 2003		
จำนวนงวด	50		

สัญญาก่อนหน้า สัญญาต่อไป

พิมพ์รายงาน พิมพ์ลูกค้าทั้งหมด กลับรายการหลัก

รูปที่ ก.22 ส่วนติดต่อกับผู้ใช้ของระบบย่อยรายละเอียดสัญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.23 ซีควেনซ์ไดอะแกรมของหน้าต่างรายละเอียดสัญญา

จากรูปที่ ก.22 จะเห็นได้ว่าหน้าต่างนี้จะให้ผู้ใช้สามารถเลือกได้ก่อนว่าจะดูข้อมูลลูกค้ารายใด ดังนั้นในซีควেনซ์ไดอะแกรมในรูป ก.23 ก็นั้น ฟอรัมก็จะต้องเรียกไปที่ออปเจกต์สทอเมอร์เพื่อให้ขอสร้างข้อมูลลูกค้าที่มีทั้งหมดขึ้นมาให้ก่อน จากนั้นออปเจกต์ฟอรัมก็จะเรียกไปที่ออปเจกต์คอนแทรก เพื่อขอให้สร้างชุดข้อมูลรายละเอียดสัญญาให้ โดยเนื้อแท้แล้ว ออปเจกต์สัญญาก็จะมีข้อมูลของสัญญาอยู่เกือบทั้งหมด แต่จะต้องการข้อมูลเพิ่มเติมคือข้อมูลที่สัญญาเช่าโดยเฉพาะ ซึ่งก็หมายถึงค่าประกัน และจำนวนงวดของการผ่อนประกันนั่นเอง ดังนั้น ออปเจกต์คอนแทรกก็จะส่งรหัสสัญญาไปสอบถามข้อมูลทั้งสองที่ต้องการนั้นมา เมื่อได้ข้อมูลครบแล้วก็ส่งชุดข้อมูล คือสัญญาที่ลูกค้ารายที่ผู้ใช้เลือกมีทั้งหมด คืน ไปให้กับ ออปเจกต์ฟอรัม เพื่อแสดงผล เป็นอันเสร็จกระบวนการ

หน้าต่างต่อมาจะเป็นหน้าต่างของระบบย่อยหน่วยพักอาศัยว่าง ซึ่งระบบย่อยนี้ประกอบด้วยหน้าต่างทั้งหมด 3 หน้าต่าง คือหน่วยพักอาศัยว่างตามเขต ตามโครงการ และตามหน่วยอาศัย ซึ่งในขั้นแรกก็จะพุดถึงหน้าต่างหน่วยพักอาศัยว่างตามเขตก่อน ซึ่งมีหน้าตาของหน้าต่างดังแสดงในรูป ก.24 และมีซีควেনซ์ไดอะแกรมดังรูป ก.25

WINDOW1

หน่วยอาศัยว่าง

เขตควบคุม

รหัสเขต ตำบล

ชื่อเขต อำเภอ

หมู่ จังหวัด

ถนน รหัสไปรษณีย์

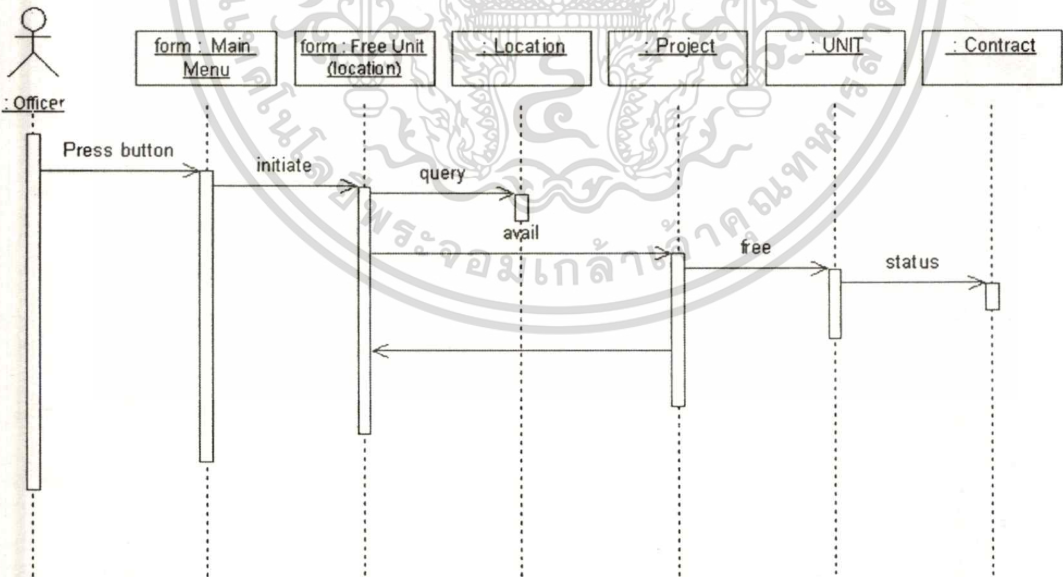
ซอย

หน่วยอาศัยว่างตามโครงการ

รหัสโครงการ	ชื่อโครงการ	รหัสเขต	หน่วยอาศัยว่าง
2	More Happy	2	9

พิมพ์ทุกหน่วยควบคุม

รูปที่ ก.24 อินเทอร์เน็ตหน่วยพักอาศัยว่าง ตามเขตควบคุม



รูปที่ ก.25 ซีควেনซ์ไดอะแกรมของหน้าต่างหน่วยพักอาศัยที่ว่างตามเขต

จากรูปหน้าต่างจะเห็นได้ว่าระบบย่อยนี้ จะให้ผู้ใช้สามารถเลือกได้ว่าจะดูข้อมูลของเขตควบคุมใดก่อน ดังนั้นในซีควেনซ์ไดอะแกรม สิ่งแรกที่ต้องทำก็คือออกปจเจตฟอร์มจะต้องส่งคำร้องเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปที่แอปเจคโบลเช่นกันเพื่อขอให้สร้างชุดข้อมูลรายการเขตควบคุมที่มีมาทั้งหมดให้ผู้ใช้เลือกก่อน เมื่อผู้ใช้เลือกแล้ว แอปเจคฟอร์มก็จะส่งรหัสเขตที่ผู้ใช้เลือกไปเป็นพารามิเตอร์ให้ออปเจคโปรเจคเพื่อให้ออปเจคโปรเจคสร้างรายงานหน่วยพักอาศัยว่างให้ และสิ่งที่ออปเจคโปรเจคต้องทำเพื่อให้ได้ข้อมูลครบตามที่ต้องการก็คือการส่งคำร้องไปยังออปเจคยูนิตเพื่อให้ออปเจคยูนิตนับว่า มีหน่วยอาศัยใดบ้างที่สังกัดโครงการที่ต้องการและยังว่างอยู่ โดยจะนับและส่งจำนวนกลับมาให้ แต่ถ้าฟังก์ชันออปเจคยูนิตเองก็ยังไม่สามารถทราบได้ว่าแต่ละหน่วยอาศัยที่อยู่ในโครงการเป้าหมายนั้นว่างหรือไม่ ทำให้จำเป็นต้องไปสอบถามสถานะของสัญญาจะออปเจคคอนแทรกก่อนว่ามีการเช่าหน่วยอาศัยนั้นหรือไม่ ซึ่งเมื่อออปเจคโปรเจคได้ข้อมูลครบแล้วก็จะส่งชุดข้อมูลคืนไปให้ฟอร์มเพื่อแสดงผลต่อไป

หน้าต่างต่อไปจะเป็นหน้าต่างที่สองของระบบย่อยหน่วยพักอาศัยว่าง ซึ่งก็คือหน้าต่างหน่วยพักอาศัยว่างตามโครงการนั่นเอง โดยหน้าต่างนี้จะแสดงเฉพาะรายละเอียดของโครงการที่ผู้ใช้เลือกมาจากหน้าต่างก่อนหน้านี้ และจะแสดงรายการหน่วยอาศัยทั้งหมดที่ว่างอยู่ของโครงการนั้น ออกมาเป็นรายหน่วยอาศัย พร้อมให้ข้อมูลเบื้องต้นของหน่วยอาศัยนั้น ทั้งยังให้ผู้ใช้สามารถเลือกเฉพาะชนิดของหน่วยอาศัยที่ต้องการอีกด้วย ดังมีโครงร่างหน้าต่างดังแสดงในรูป ก.26 และมีซีเควนซ์ไคอะแกรมดังแสดงในรูป ก.27

WINDOW1

หน่วยอาศัยว่างตามโครงการ

โครงการ

รหัสโครงการ ชื่อโครงการ

ชนิดอาคาร

รหัสชนิดอาคาร ชื่อชนิดอาคาร

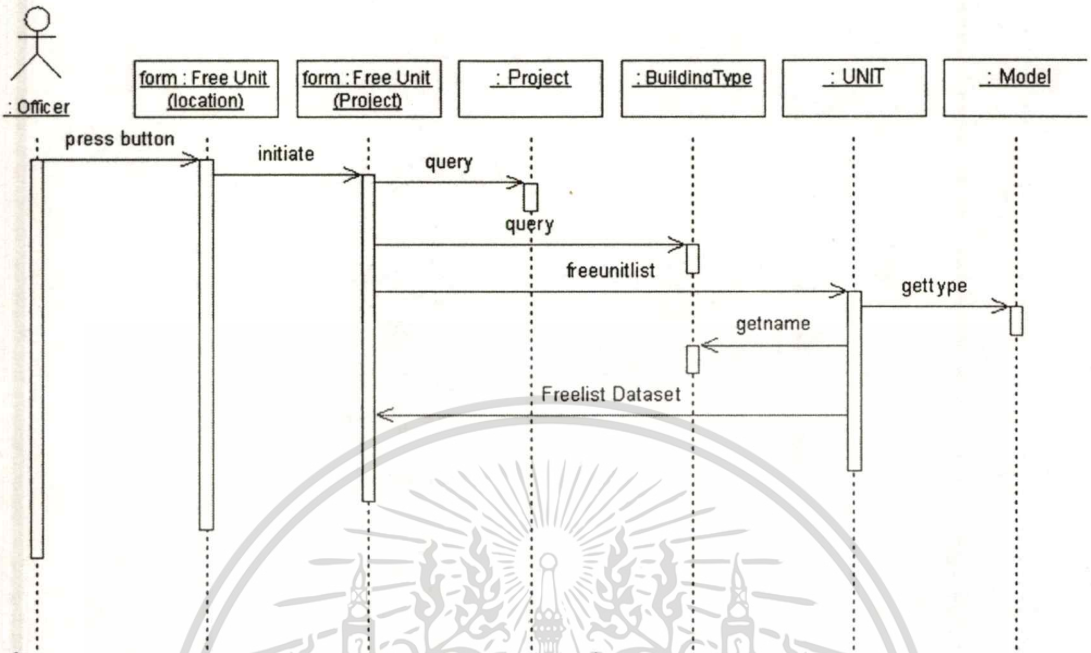
หน่วยอาศัย

รหัสหน่วยอาศัย	ชนิดอาคาร	ชั้นที่	อาคารที่
100004	standard house	0	0
101002	standard house	0	0
101003	standard house	0	0

พิมพ์ทุกโครงการ

รูปที่ ก.26 หน่วยพักอาศัยว่างตามโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.27 ซีเควนซ์ไคอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตามโครงการ

จากรูปหน้าต่างส่วนติดต่อนี้จะเห็นว่าจะอนุญาตให้ผู้ใช้เลือกได้ก่อนว่าต้องการจะดูข้อมูลของโครงการใด โดยโครงการแรกที่จะแสดงให้ดูจะเป็น โครงการที่ผู้ใช้เลือกมาก่อนแล้วจากหน้าต่างก่อนหน้านี้ ดังนั้นในซีเควนซ์ไคอะแกรม สิ่งแรกที่ต้องทำก็คือฟอร์มจะส่งคำร้องไปที่ออปเจคโปรเจคเพื่อขอให้ แสดงข้อมูล โครงการทั้งหมดที่อยู่ในเขตควบคุมของหน้าต่างก่อนหน้านี้มาให้เลือก

นอกจากนั้นแล้วที่ส่วนกลางของหน้าต่างผู้ใช้ยังสามารถเลือกแบบบ้านที่อยู่ในความสนใจได้ด้วย ดังนั้นออปเจคฟอร์มก็ต้องส่งคำร้อง ไปที่ออปเจคบิวคิงไทป์เพื่อให้สร้างชุดข้อมูลชนิดของอาคารที่มีมาทั้งหมดเพื่อให้ผู้ใช้เลือกอีกเช่นกัน

เมื่อผู้ใช้เลือกทั้งโครงการ และชนิดอาคารที่ต้องการแล้ว ออปเจคฟอร์มก็จะส่งคำร้องพร้อมการเลือกของผู้ใช้ไปที่ออปเจคยูนิตเพื่อขอให้สร้างรายงานหน่วยพักอาศัยว่างให้ โดยออปเจคยูนิตก็ต้องอาศัยความร่วมมือจากออปเจคคอนแทรกเพื่อสอบถามสถานะการเช่าก่อน และโดยออปเจคยูนิตเองทราบอยู่แล้วว่า มีหน่วยอาศัยใดบ้างอยู่ในโครงการเป้าหมาย ดังนั้นสิ่งสุดท้ายที่ออปเจคยูนิตต้องทำ ก็คือเลือกเอาเฉพาะหน่วยอาศัยที่ตรงกับชนิดอาคารที่ผู้ใช้เลือกมาเท่านั้น การจะตรวจสอบได้ว่าหน่วยอาศัยนั้นเป็นชนิดใด ก็ต้องเอาแบบบ้านที่ตนเองมีนั้น ไปถามยังออปเจคแบบบ้านว่าแบบบ้านนั้นเป็นชนิดอาคารใด และหลังจากนั้นก็ไปสอบถามยังออปเจคบิวคิงไทป์เพื่อขอชื่อแบบบ้านนั้นว่าชื่ออะไรมาสร้างเป็นรายงาน และส่งคืนไปให้ออปเจคฟอร์มเพื่อแสดงผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างต่อไปจะเป็นหน้าต่างสุดท้ายของระบบย่อยหน่วยพักอาศัยว่าง ซึ่งจะเป็นหน้าต่างที่ให้รายละเอียดแต่ละหน่วยพักอาศัยว่ามีคุณสมบัติเป็นอย่างไร และมีประวัติการเช่าเป็นอย่างไร ดังมีหน้าต่างแสดงไว้ในรูป ก.28 และมีซีเควนซ์ไดอะแกรมแสดงในรูป ก.29

WINDOW1

หน่วยอาศัยว่าง - รายละเอียดหน่วยอาศัย

หน่วยอาศัย

รหัสหน่วยอาศัย: 102002

ชื่อรุ่นอาคาร: american house

ชนิดอาคาร: standard house

ชั้นที่: []

อาคารที่: []

กลับหน้า ต่อไป

คุณสมบัติของโมเดล

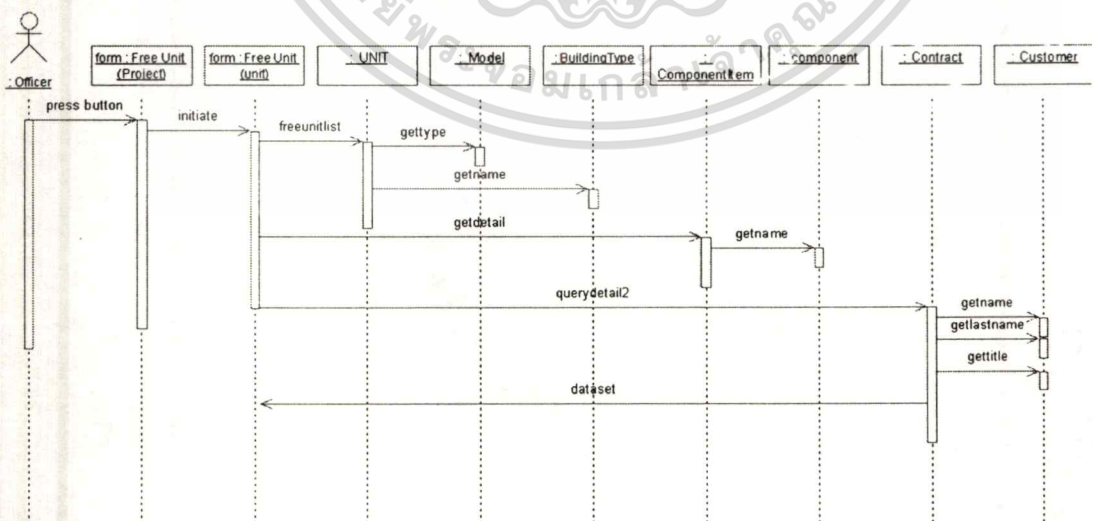
จำนวน	ส่วนประกอบ
4	bedroom
3	bathroom
3	floor
1	restroom

ประวัติสัญญา

เลขที่สัญญา	สถานะ	งวดแรก	งวดสุดท้าย	คำนำชื่อ	ชื่อลูกค้า	นามสกุล
920100	2	01 มี.ค. 1995	01 เม.ย. 1999	Mr.	Fijob	Capra
901001	2	01 พ.ค. 1996	01 มิ.ย. 2000	Mr.	Alexandre	Dumas

พิมพ์รายงาน กลับรายการก่อนหน้า

รูปที่ ก.28 หน้าต่างระบบย่อยรายละเอียดหน่วยอาศัยที่ว่าง



รูปที่ ก.29 ซีเควนซ์ไดอะแกรมหน้าต่างหน่วยพักอาศัยที่ว่างตามหน่วยพักอาศัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปหน้าต่างจะเห็นได้ว่าส่วนติดต่อกับผู้ใช้นี้จะออกแบบมาให้ผู้ใช้สามารถเลือกได้ว่าต้องการจะดูข้อมูลของหน่วยอาศัยใด ดังนั้นสิ่งที่ต้องทำในซีเควนซ์โคอะแกรมก่อนก็คือให้ฟอร์มส่งคำร้องไปยังออปเจกต์ยูนิคเพื่อให้แสดงรายงานหน่วยอาศัยทั้งหมดที่อยู่ในโครงการเป้าหมาย และต้องเป็นหน่วยพักอาศัยที่มีลักษณะชนิดอาคารตรงกับที่ผู้ใช้เลือกไว้ในหน้าต่างก่อนด้วย เมื่อได้รับรายชื่อหน่วยอาศัยมาแล้ว ผู้ใช้ก็จะสามารถเลือกได้ว่าต้องการดูข้อมูลของหน่วยอาศัยใด เมื่อผู้ใช้เลือกแล้ว ก็จะเป็นการพยายามแสดงรายละเอียดของหน่วยอาศัยนั้น โดยต้องส่งคำร้องไปยังออปเจกคอมโพเนนท์ไอเทมเพื่อให้ออปเจกต์สร้างรายการส่วน ประกอบของแบบบ้านชนิดนั้นโดยส่งรหัสแบบบ้านไปให้เป็นพารามิเตอร์โดยออปเจกคอมโพเนนท์ไอเทมก็จะทราบว่าบ้านแบบนั้นมีส่วนประกอบอะไรบ้าง โดยต้องไปเอาข้อมูลส่วนประกอบมาจากออปเจกคอมโพเนนท์ เพียงเท่านี้ก็จะสามารถสร้างรายการส่วนประกอบได้

จากนั้นก็จะเป็นการสร้างรายงานประวัติการเช่า โดยฟอร์มจะส่งคำร้องไปที่ออปเจกคอนแทรกเพื่อให้สร้างรายงานประวัติการเช่าให้ ออปเจกคอนแทรกซึ่งมีข้อมูลประวัติการเช่าครบในตัวก็ติดต่อไปยังออปเจกต์สทอเมอร์ เพื่อขอชื่อ และนามสกุลลูกค้ามาประกอบเป็นรายงานที่สมบูรณ์ เมื่อได้ข้อมูลครบทั้งหมดแล้วก็ส่งชุดข้อมูลคืนไปให้ออปเจกฟอร์มเพื่อทำการแสดงผลต่อไป

หน้าต่างต่อไปจะเป็นหน้าต่างของระบบย่อยการชำระเงินประจำวัน ซึ่งประกอบด้วยหน้าต่างเพียงหน้าต่างเดียวเท่านั้นดังแสดงในรูป ก.30 และมีซีเควนซ์โคอะแกรมดังรูป ก.31

หน้าต่าง WINDOW1

การชำระเงินประจำวัน

วันที่ 01 มีนาคม 1999 เฉพาะค่าเช่า

เขตควบคุม รหัสเขต 1 ชื่อเขต bangkae region

รายการชำระเงิน

เลขที่ชำระเงิน	วันที่จ่าย	รหัสสัญญา	รหัสหน่วยอาศัย	รายการชำระเงิน	จำนวน	งวดที่จ่าย
1	01 มี.ค. 1999	1001000	100001	rent	1000	มี.ค. 1999
1	01 มี.ค. 1999	1001000	100001	insurance	21	มี.ค. 1999
1	01 มี.ค. 1999	1001000	100001	water	10	มี.ค. 1999
121	01 มี.ค. 1999	100202	100002	rent	1000	มี.ค. 1999
121	01 มี.ค. 1999	100202	100002	insurance	100	มี.ค. 1999

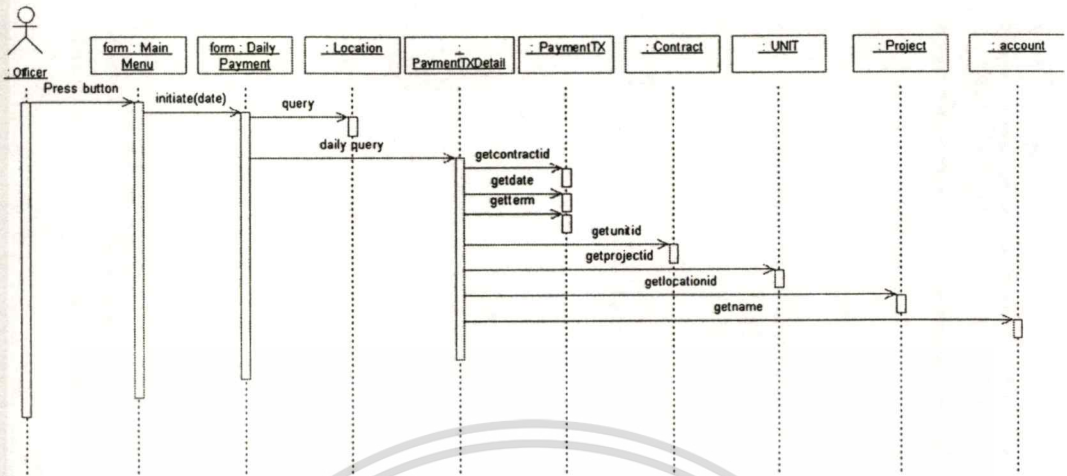
ยอดรวม 3431

พิมพ์ทุกเขตควบคุม

รูปที่ ก.30 อินเทอร์เน็ตของระบบย่อย การชำระเงินประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.31 ซีควেনซ์ไดอะแกรมของหน้าต่างการชำระเงินประจำวัน

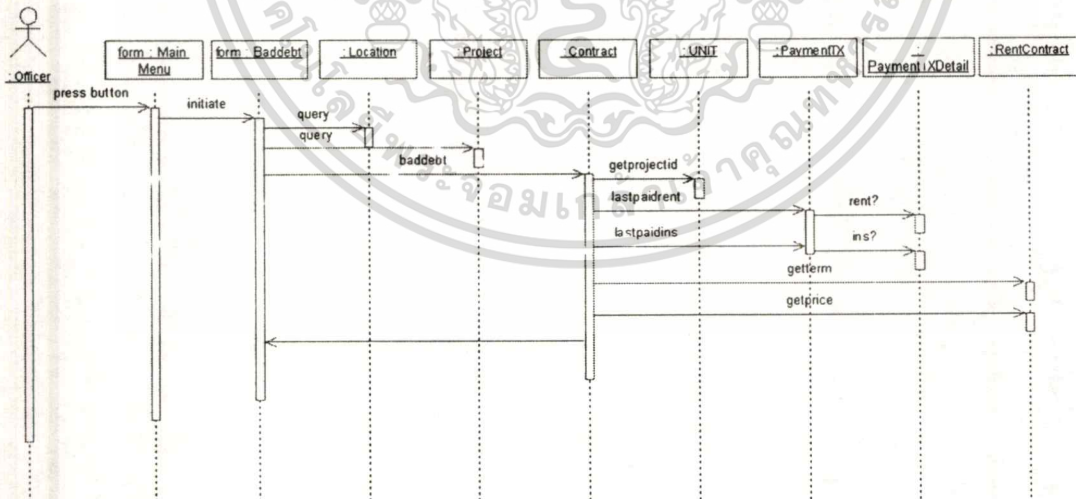
จากหน้าต่างส่วนติดต่อนี้ จะให้ผู้ใช้เลือกวันที่ที่ต้องการแสดงข้อมูล โดยการเลือกวันที่นี้ไม่จำเป็นต้องติดต่อกับใคร จึงไม่มีกิจกรรมเกิดขึ้นในซีควেনซ์ไดอะแกรม และต่อไปก็ให้ผู้ใช้เลือกเขตควบคุมที่ต้องการแสดงข้อมูล ดังนั้นในซีควেনซ์ไดอะแกรม ฟอรัมจึงติดต่อกับคลาสโลเคชัน เพื่อให้สร้างชุดข้อมูลแสดงรายการหน่วยควบคุมทั้งหมดมาให้ผู้ใช้เลือก เมื่อผู้ใช้เลือกแล้วก็จะส่งคำร้องไปยังออปเจกต์ดีเทลเพื่อขอให้สร้างรายงานการชำระเงินให้ โดยส่งเขตควบคุม และวันที่ไปเป็นพารามิเตอร์

สิ่งแรกที่ออปเจกต์ดีเทลต้องทำก็คือ การตรวจสอบว่าการชำระเงินแต่ละรายการที่ตนเองมีข้อมูลอยู่นั้น เป็นของลูกค้ำท่านใด เป็นการชำระเงินของงวดเดือนไหน และชำระเงินเมื่อวันที่เท่าใด โดยทั้งหมดนี้จะต้องสอบถามไปที่ออปเจกต์คอนแทรกต์ เมื่อได้ข้อมูลแล้ว ก็จะเลือกเอาเฉพาะวันที่ที่ต้องการเท่านั้น จากนั้นก็ติดต่อไปยังออปเจกต์คอนแทรกต์ ซึ่งจะติดต่ออีกทอดไปยังออปเจกต์ยูนิท ซึ่งก็จะติดต่ออีกทอดไปยังออปเจกต์โปรเจกต์ โดยที่ทำไมทั้งหมดนี้ ก็เพื่อจะตรวจสอบว่าการชำระเงินครั้งนั้น ๆ เป็นการชำระเงินของเขตควบคุมที่ต้องการหรือไม่ เพราะออปเจกต์การชำระเงินก็จะทราบว่าเป็นของสัญญาใด ออปเจกต์สัญญาก็จะทราบว่าเป็นหน่วยอาศัยใด ออปเจกต์หน่วยอาศัยก็จะทราบว่าเป็นโครงการใด และออปเจกต์โครงการก็จะทราบว่าเป็นหน่วยควบคุมใด โดยจะเลือกเฉพาะหน่วยควบคุมที่ต้องการ ดังนั้นก็จะสามารถสร้างเป็นรายงานได้แล้ว แต่รายงานก็ยังขาดความสมบูรณ์ นั่นคือการอธิบายว่าค่าใช้จ่ายแต่ละรายการนั้นเป็นค่าใช้จ่ายรายการอะไร ซึ่งก็ต้องสอบถามไปยังออปเจกต์แอคเคาท์ เพื่อให้ได้ข้อมูลนี้ เมื่อได้ข้อมูลครบทุกอย่างแล้ว ก็สามารถสร้างเป็นชุดข้อมูลรายงานที่สมบูรณ์ คินไปให้ออปเจกต์ฟอรัม เพื่อแสดงผลต่อไป

หน้าต่างต่อไปจะเป็นหน้าต่างของระบบย่อยหนี้เสีย ซึ่งเป็นระบบย่อยที่มีหน้าต่างเพียงหน้าต่างเดียวเช่นกัน โดยระบบย่อยนี้จะให้ข้อมูลลูกค้าที่ค้างชำระหนี้ตามโครงการ ดังนั้นหน้าต่างของส่วนติดต่อกับผู้ใช้ก็จะมีหน้าต่างดังรูป ก.32 และจะมีชีควอนซ์โคอะแกรมดังรูป ก.33

รหัสหน่วยอาศัย	เลขที่สัญญา	ค่าเช่าค้าง	ค่าประกันค้ำ	จ่าย	ค้าง	รวม	สถานะ	ทำนาชื่อ	ชื่อ	นามสกุล
100001	1001000	9000	0	4	9	50	1	Mr. Frijob	Capra	
100003	102001	3000	60	0	3	60	1	Mrs. Michael	Crichton	
100005	1020009	13000	400	0	14	15	1	Mr. Frijob	Capra	

รูปที่ ก.32 อินเทอร์เฟซของระบบย่อยหนี้เสีย



รูปที่ ก.33 ชีควอนซ์โคอะแกรมของหน้าต่างหนี้เสีย

จากรูปหน้าต่างจะเห็นว่าสิ่งที่ผู้ใช้ต้องทำก่อนคือเลือกวันที่ที่จะเริ่มนับหนี้เสีย ซึ่งการเลือกวันนี้ไม่จำเป็นต้องติดต่อกับแอปเจกอื่น จึงสามารถทำได้ภายในแอปเจกฟอร์ม ทำให้ไม่จำเป็นต้องมีการสื่อสารในชีควอนซ์โคอะแกรม สิ่งต่อมาที่ผู้ใช้ต้องทำก็คือการเลือกหน่วยควบคุมที่เอกสารนี้เป็นเอกสารที่ส่งวันเวสสำหรับการเรียนเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการข้อมูล และเลือกโครงการที่ต้องการข้อมูล ดังนั้นสิ่งที่ต้องทำในซีเควนซ์ไคอะแกรมก็คือให้แอปเจคโลเคชันสร้างรายการหน่วยควบคุมที่มีทั้งหมดขึ้นมาให้ เมื่อผู้ใช้เลือกแล้ว ก็ส่งค่าหน่วยควบคุมที่ผู้ใช้เลือกไปให้แอปเจคโปรเจคสร้างรายการโครงการที่อยู่ในหน่วยควบคุมนั้น ให้ผู้ใช้เลือกอีกเช่นกัน เมื่อผู้ใช้เลือกโครงการที่ต้องการแล้ว ก็ส่งรหัสโครงการนั้น กับวันที่ผู้ใช้เลือกไปให้แอปเจคคอนแทรกเพื่อสร้างรายงานหนี้เสีย ส่งแรกที่แอปเจคคอนแทรกต้องทำก็คือเลือกเฉพาะสัญญาที่อยู่ในโครงการที่ผู้ใช้เลือก โดยสอบถามชื่อโครงการไปยังแอปเจคยูนิต และเลือกเอาเฉพาะสัญญาที่ตรงเงื่อนไขเท่านั้น โดยรวมแล้วแอปเจคคอนแทรกจะทราบข้อมูลเกี่ยวกับสัญญาเกือบทั้งหมด ดังนั้นสิ่งที่ต้องการทราบเพื่อจะระบุได้ว่าสัญญาใดค้างชำระบ้างก็คือ การชำระเงินครั้งสุดท้ายนั่นเอง ดังนั้นแอปเจคคอนแทรกจึงต้องส่งคำร้องไปยังแอปเจคเพเมนต์เพื่อสอบถามว่าสัญญานั้นชำระค่าเช่า และค่าประกันครั้งสุดท้ายเมื่อใด โดยแอปเจคเพเมนต์ก็ต้องสอบถามไปยังแอปเจคเพเมนต์ดีเทลเพื่อให้ทราบว่าชำระเงินแต่ละครั้งมีค่าเช่า และค่าประกันหรือไม่ เมื่อทราบวันที่ชำระค่าเช่า และค่าประกันครั้งสุดท้ายแล้ว ก็สามารถทราบได้ว่าสัญญาใดบ้างที่มีหนี้ค้างชำระ โดยสามารถคำนวณค่าเช่าค้างชำระได้ทันที แต่ค่าประกันค้างชำระจะต้องสอบถามข้อมูลไปยังแอปเจคเรนต์คอนแทรกเพิ่มเติม เพื่อจะทราบได้ว่าสัญญานั้นมีค่าประกันเท่าใด และถึงงวด จากนั้นก็ทำการคำนวณค่า และสร้างเป็นชุดข้อมูลส่งคืนไปให้แอปเจคฟอร์มทำการแสดงผลต่อไป

หน้าต่างต่อไปซึ่งเป็นหน้าต่างสุดท้าย ก็คือหน้าต่างของระบบย่อยลูกหนี้หมดอายุสัญญา โดยมีหน้าต่างดังแสดงในรูปที่ ก.34 และมีซีเควนซ์ไคอะแกรมดังแสดงในรูปที่ ก.35

ลูกหนี้หมดอายุสัญญา

เขตควบคุม: รทสเขต [] ชื่อเขต: bangkae region []

โครงการ: รหัสโครงการ: 1 ชื่อโครงการ: Happy Village []

หน่วยอาศัยหมดสัญญา

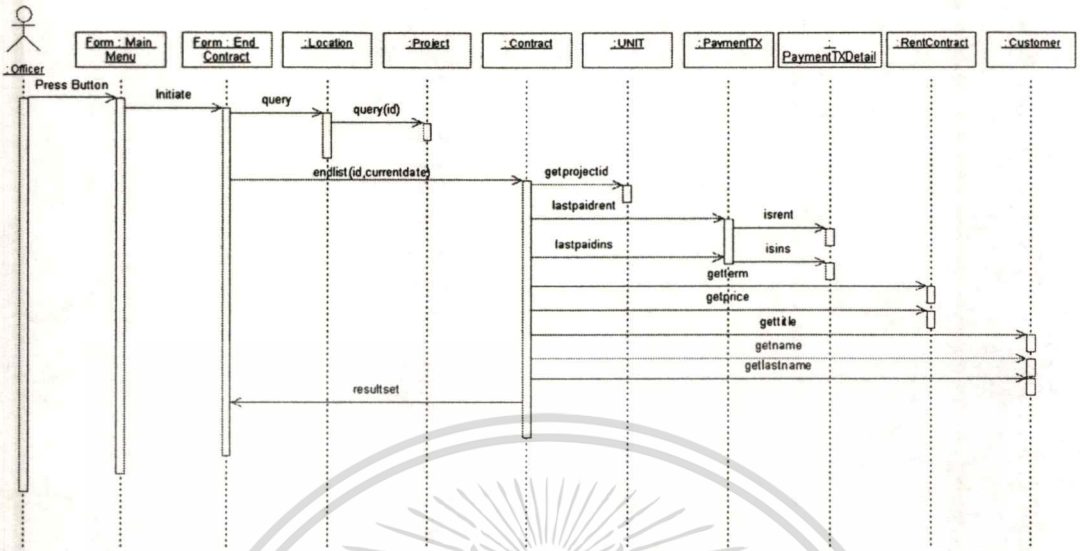
รหัสหน่วย	เลขที่สัญญา	งวดแรก	ค่าเช่า	สถานะ	สำเนา	ชื่อ	นามสกุล	ค่าเช่าค้าง	ค่าประกันค้าง
100005	1020009	01 ม.ค. 1999	1000	1	Mr.	Frijob	Capra	15000	500
101001	902001	01 ก.พ. 1999	1200	1	Mr.	Victor	Hugo	1200	0

พิมพ์ทุกโครงการ

รูปที่ ก.34 อินเทอร์เฟซของระบบย่อยลูกหนี้หมดอายุสัญญา

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.35 ซีควเอนซ์ไดอะแกรมของหน้าต่างลูกหนี้หมดอายุสัญญา

หน้าต่างสุดท้ายนี้จะเป็นหน้าต่างที่แสดงข้อมูลลูกหนี้ที่หมดอายุสัญญา จากรูปส่วนติดต่อกับผู้ใช้ จะแสดงให้เห็นว่าผู้ใช้จะต้องเลือกเขตควบคุม และโครงการที่ต้องการดูข้อมูลก่อน ดังนั้น สิ่งแรกที่ต้องทำในซีควเอนซ์ไดอะแกรมก็คือ การส่งคำร้องไปยังออปเจกต์โลเคชันเพื่อให้สร้างชุดข้อมูลรายการเขตควบคุมทั้งหมดที่มีเพื่อให้ผู้ใช้สามารถเลือกได้

เมื่อผู้ใช้เลือกแล้วก็จะส่งรหัสเขตที่ผู้ใช้เลือกไปให้ออปเจกต์โปรเจกต์เพื่อสร้างชุดข้อมูลรายการโครงการที่อยู่ในเขตควบคุมนั้นมาให้ผู้ใช้เลือกอีกเช่นกัน เมื่อผู้ใช้เลือกโครงการที่ต้องการแล้ว ออปเจกต์ฟอร์มก็จะส่งคำร้องไปยังออปเจกต์คอนแทรกต์เพื่อขอให้สร้างรายงานลูกหนี้หมดอายุสัญญาให้ โดยจะต้องส่งรหัสโครงการมาให้เป็นพารามิเตอร์ ออปเจกต์คอนแทรกต์ก็ต้องเลือกเฉพาะสัญญาที่อยู่ในโครงการที่ผู้ใช้เลือกเท่านั้น

ดังนั้นสิ่งที่ต้องทำก็คือการสอบถามไปยังออปเจกต์ยูนิตเพื่อดูว่าแต่ละสัญญานั้นอยู่ในโครงการใด เมื่อได้ข้อมูลแล้ว ก็เลือกเฉพาะสัญญาที่อยู่ในโครงการเป้าหมาย จากนั้นก็จะเลือกเฉพาะสัญญาที่หมดอายุเท่านั้น โดยข้อมูลการหมดอายุอยู่ในออปเจกต์คอนแทรกต์แล้วจึงไม่จำเป็นต้องสอบถามไปที่อื่นอีก

เมื่อได้ข้อมูลแล้ว สิ่งที่ต้องทำต่อไปก็คือการคำนวณหนี้ค้างชำระ โดยต้องคำนวณทั้งค่าเช่าค้างชำระ และค่าประกันค้างชำระ โดยวิธีการก็คือสอบถามไปยังออปเจกต์เพเมนต์เพื่อสอบถามว่าแต่ละสัญญาที่จะหมดอายุนั้น ชำระค่าเช่า และค่าประกันครั้งสุดท้ายเมื่อใด โดยออปเจกต์เพเมนต์ก็ต้องสอบถามไปยังออปเจกต์เพเมนต์ดีเทลอีกทอด ว่าแต่ละการชำระเงินนั้นมีค่าเช่า และค่าประกันอยู่ด้วยหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

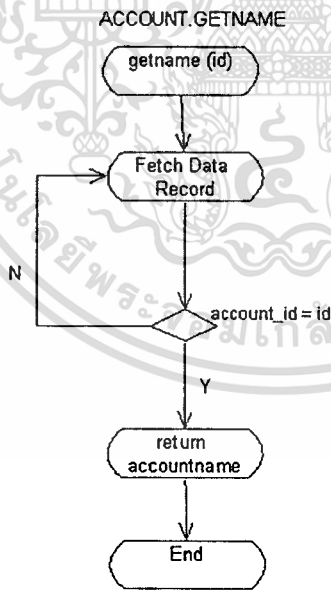
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ข้อมูลการชำระหนี้ครั้งสุดท้ายแล้ว ก็จะคำนวณค่าเช่าค้างชำระ แต่การคำนวณค่าประกันค้างชำระจำเป็นต้องสอบถามข้อมูลเพิ่มเติมไปยังแอปเจเนอเรทเพเมนต์ เพื่อสอบถามว่าอัตราค่าประกันต่อเดือนเป็นเท่าใด และต้องจ่ายค่าประกันกี่งวด เมื่อได้ข้อมูลแล้วก็สามารถคำนวณค่าประกันค้างชำระได้ และสุดท้ายก็สอบถามไปยังแอปเจเนอเรทเพเมนต์เพื่อสอบถามชื่อ และนามสกุลของเจ้าของสัญญา เมื่อได้ข้อมูลทุกอย่างครบแล้วก็สร้างเป็นชุดข้อมูลส่งคืนไปให้ออปเจเนอเรทแสดงผลต่อไป

ก.4 แอคทีวิตี้ไดอะแกรมของบริการของคลาส

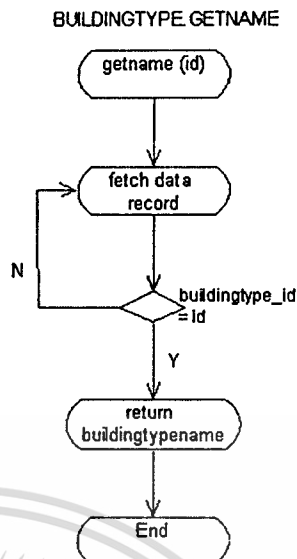
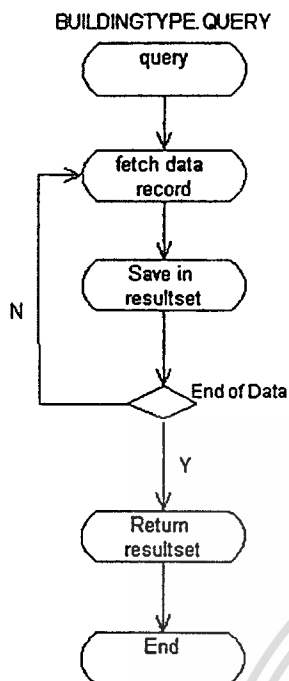
ต่อไปนี้จะเป็นส่วนที่แสดงการทำงานภายในแต่ละเมธอดของคลาส ซึ่งก็หมายถึงบริการแต่ละรายการที่แต่ละคลาสมีให้นั่นเอง โดยบางบริการสามารถทำเสร็จลุล่วงภายในคลาสตนเอง เพราะข้อมูลทุกอย่างมีพร้อมแล้ว ในขณะที่บางบริการจำเป็นต้องอาศัยความร่วมมือจากคลาสอื่นด้วย เพราะต้องการข้อมูลเพิ่มเติมนั่นเอง

โดยการแสดงการทำงานภายในนั้นจะเป็นการแสดงให้เห็นอยู่ในรูปของแอคทีวิตี้ไดอะแกรม ซึ่งมีลักษณะคล้ายโฟลว์ชาร์ตนั่นเอง โดยจะแสดงไปที่แต่ละคลาส ทีละบริการจนครบ เริ่มจากคลาสแอคเคาท์ก่อนซึ่งมีบริการเดียว และแสดงในรูป ก.36



รูปที่ ก.36 แอคทีวิตี้ไดอะแกรมของเมธอดของคลาสแอคเคาท์

ต่อไปรูป ก.37 จะเป็นแอคทีวิตี้ของเมธอดของคลาสบิลดิ้งไทป์ ซึ่งมีเมธอด 2 เมธอดด้วยกัน คือ “query” และ “getname”



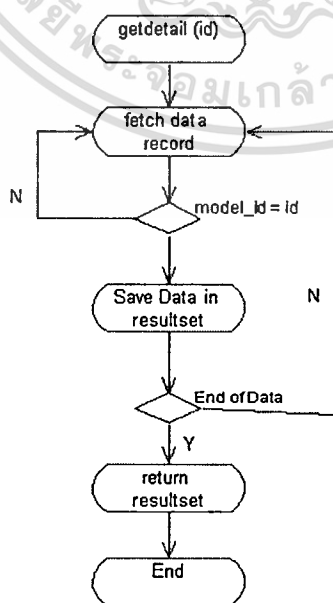
(ก) เมธอร์ค “Query”

(ข) เมธอร์ค “Getname”

รูปที่ ก.37 แอคทิวิตีไดอะแกรมของเมธอร์คของคลาสบิวติงไทป์

ต่อไปจะเป็นชุดแอคทิวิตีไดอะแกรมของเมธอร์คของคลาสคอมโพเนนท์ไอเทม ซึ่งมีเพียงเมธอร์คเดียวเท่านั้น ดังแสดงในรูป ก.38

COMPONENTITEM.GETDETAIL

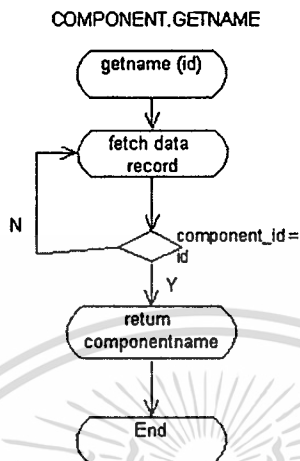


รูปที่ ก.38 แอคทิวิตีไดอะแกรมของเมธอร์คของคลาสคอมโพเนนท์ไอเทม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

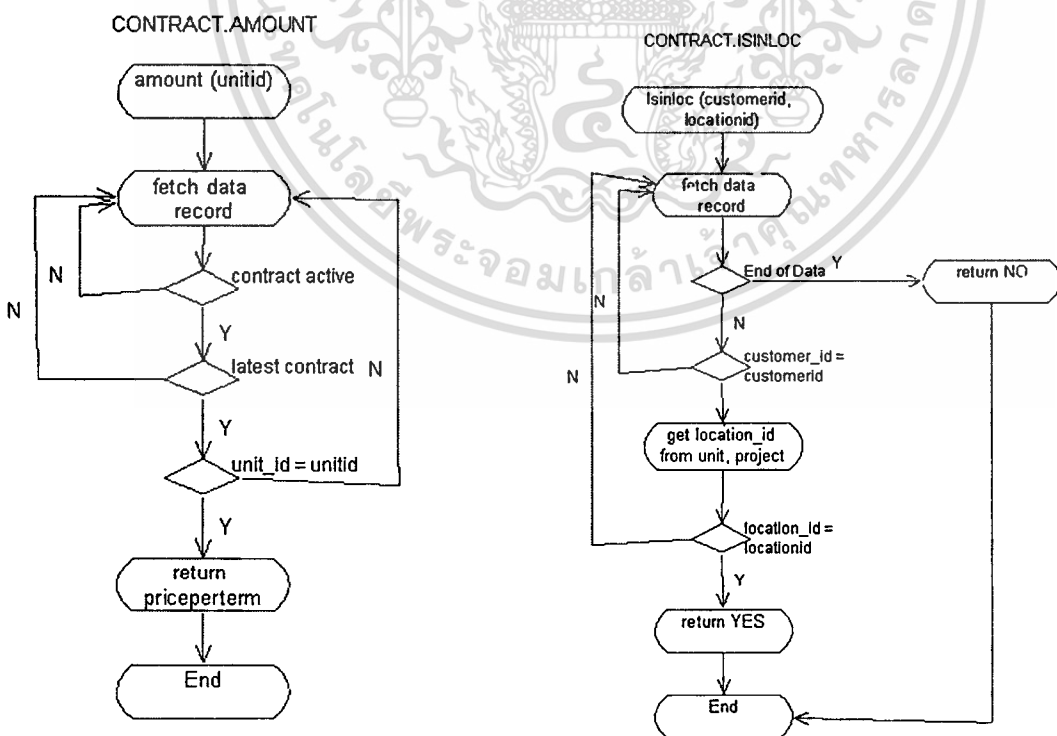
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปจะเป็นคลาสไดอะแกรมของคลาสคอมพิวเตอร์ ที่มีบริการเพียงบริการเดียว ดังแสดงในรูป ก.39



รูปที่ ก.39 แอคทिवิตีไดอะแกรมของเมธอร์ดของคลาสคอมพิวเตอร์

ต่อไปจะเป็นแอคทिवิตีไดอะแกรมของเมธอร์ดของคลาสคอนแทรกต์ ซึ่งเป็นคลาสขนาดใหญ่ที่มีเมธอร์ดถึง 10 เมธอร์ด ดังแสดงในรูปที่ ก.40



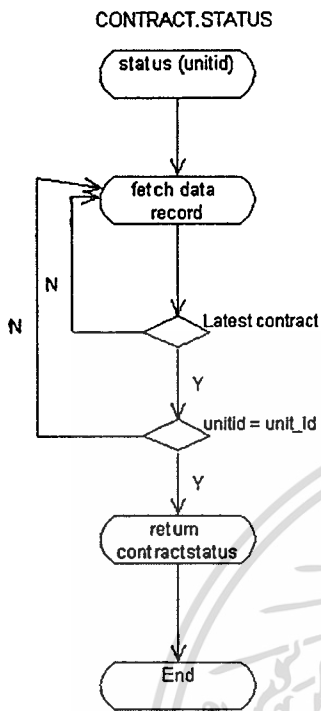
(ก) เมธอร์ด "Amount"

(ข) เมธอร์ด "Isinloc"

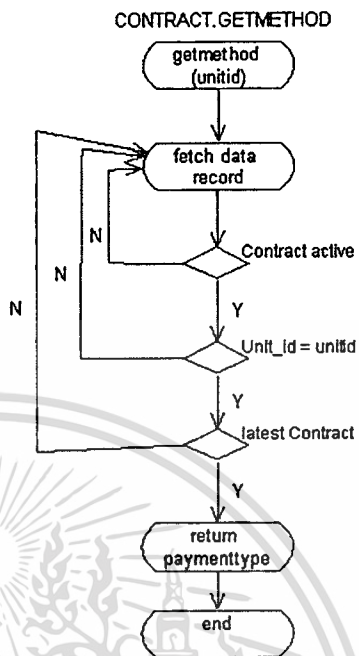
รูปที่ ก.40 แอคทिवิตีไดอะแกรมของเมธอร์ดของคลาสคอนแทรกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

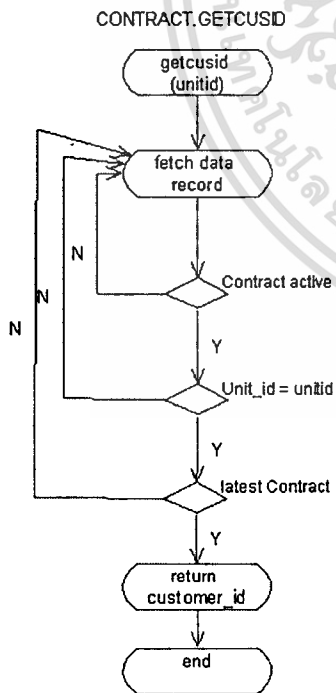
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



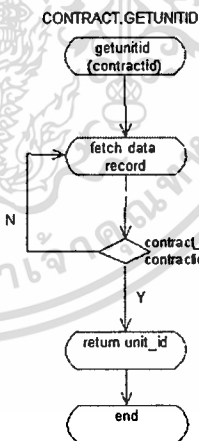
(ค) เมธอร์ด “Status”



(ง) เมธอร์ด “Getmethod”



(จ) เมธอร์ด “Getcusid”

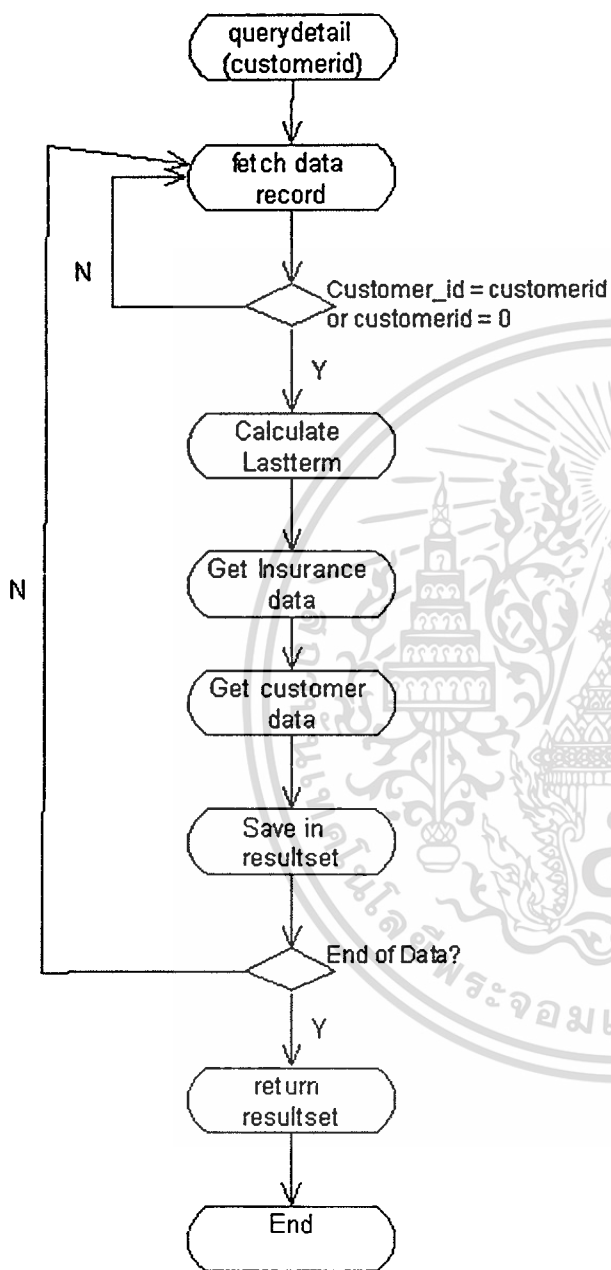


(ฉ) เมธอร์ด “Getunitid”

รูปที่ ก.40 (ต่อ) แอคทีวิตีไดอะแกรมของเมธอร์ดของคลาสคอนแทรก

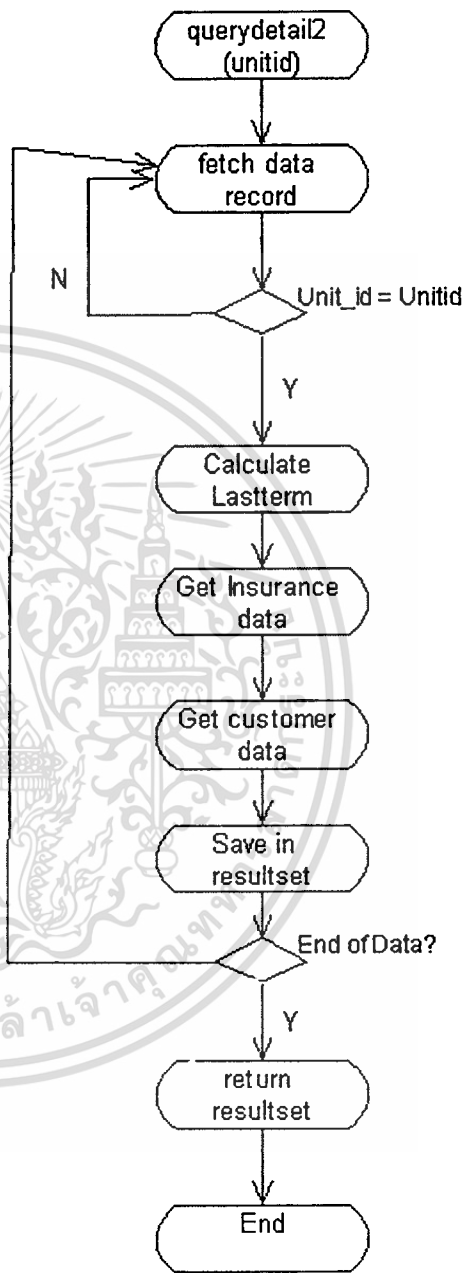
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTRACT.QUERYDETAIL



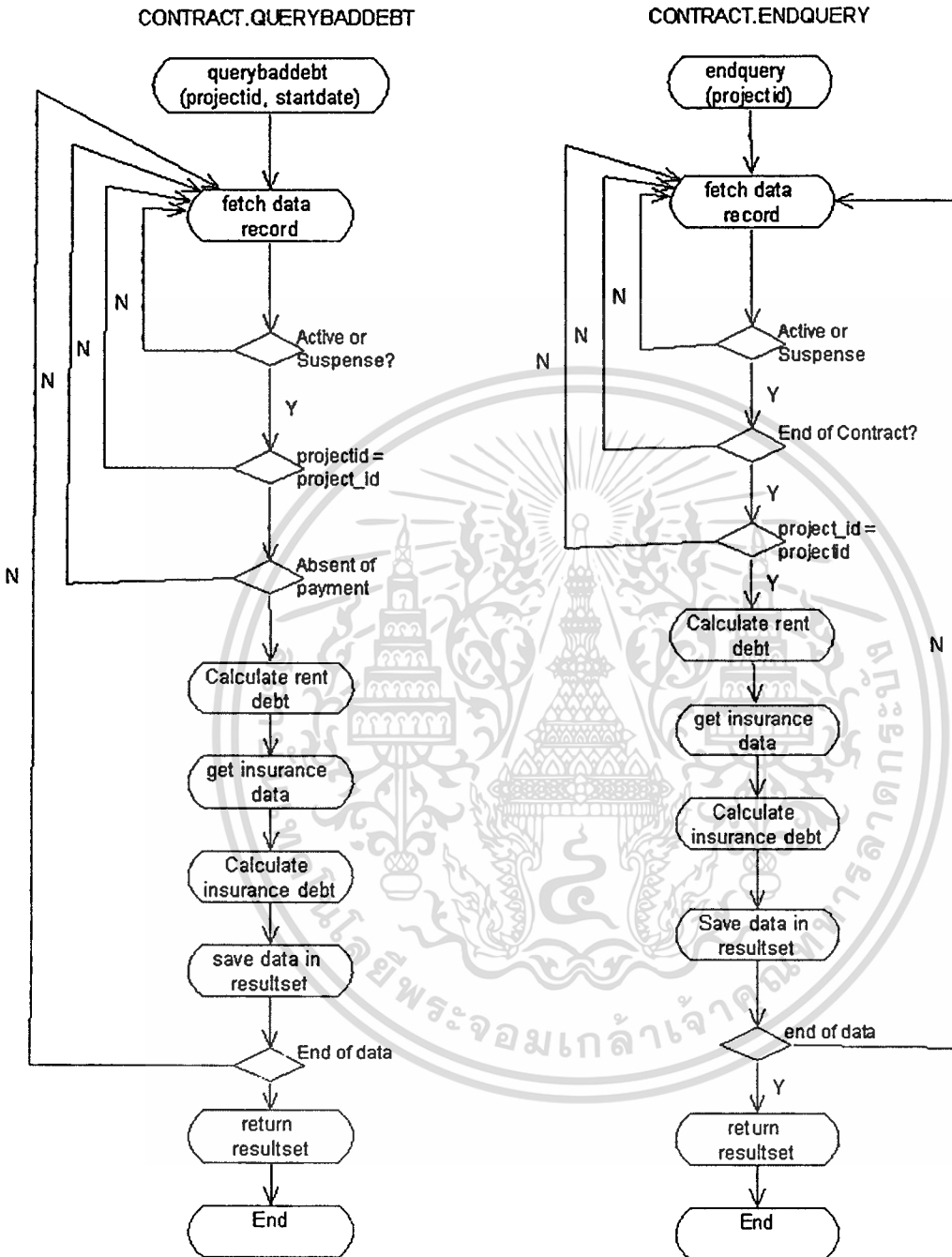
(ข) เมธอร์ด “Getdetail”

CONTRACT.QUERYDETAIL2



(ข) เมธอร์ด “Getdetail2”

รูปที่ ก.40 (ต่อ) แอคทีวิตีไดอะแกรมของเมธอร์ดของคลาสคอนแทรค



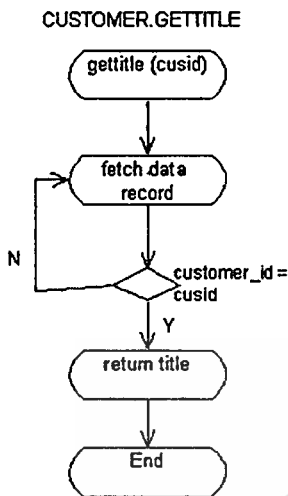
(ณ) เมธอร์ด "Querybaddebt"

(ญ) เมธอร์ด "Endquery"

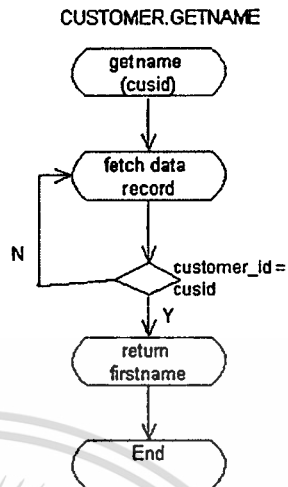
รูปที่ ก.40 (ต่อ) แอคทีวิตีไดอะแกรมของเมธอร์ดของคลาสคอนแทรก

ต่อไปแอคทีวิตีไดอะแกรมของเมธอร์ดของคลาสคัสทอเมอร์ ซึ่งประกอบด้วย 5 เมธอร์ด
 ดังแสดงในรูป ก.41

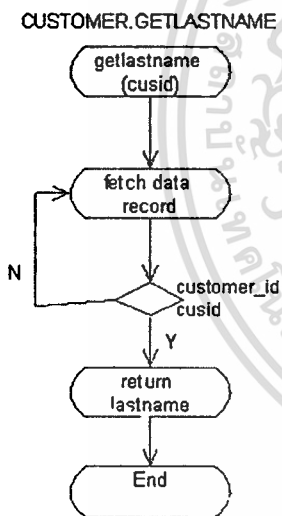
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



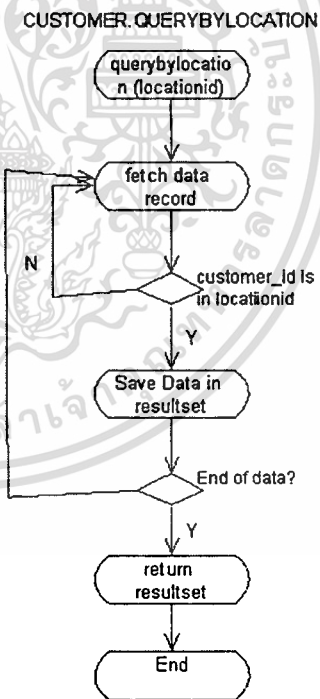
(ก) เมธอร์ค “Gettitle”



(ข) เมธอร์ค “Getname”

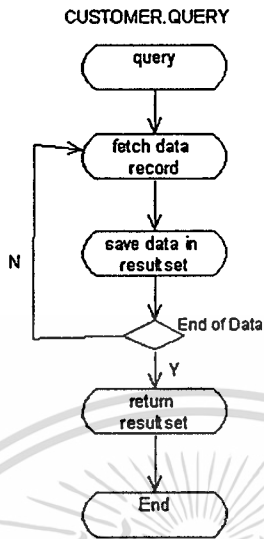


(ค) เมธอร์ค “Getlastname”



(ง) เมธอร์ค “Querybylocation”

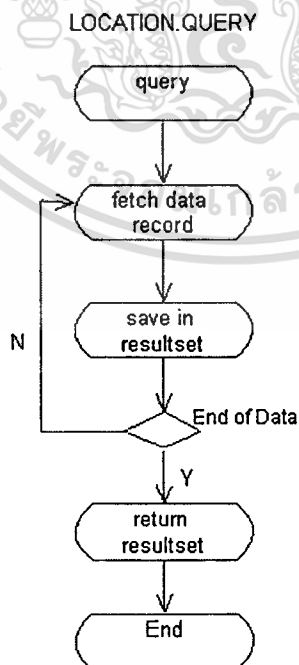
รูปที่ ก.41 แดทวิทดีโอะแกรมของเมธอร์คของคลาสคัสทอเมอร์



(จ) เมธอร์ด “Query”

รูปที่ ก.41 (ต่อ) แอคทิวิตีไดอะแกรมของเมธอร์ดของคลาสคัสตอมเมอร์

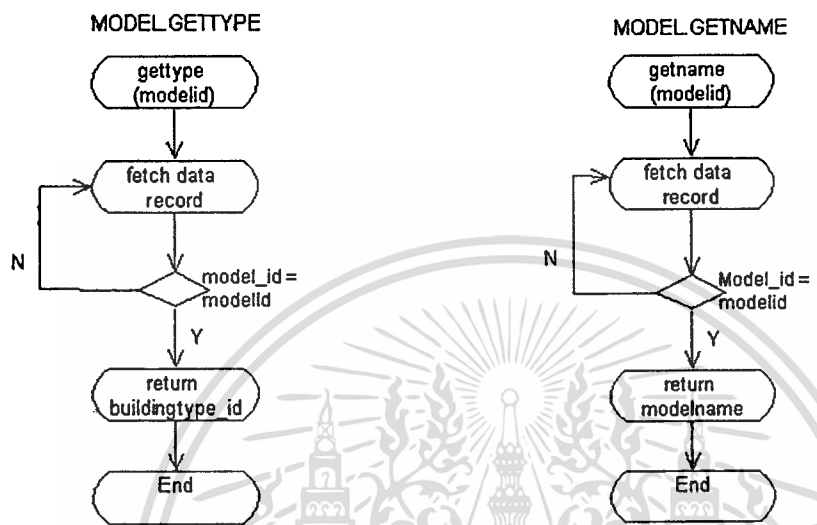
ต่อไปจะเป็นแอคทิวิตีไดอะแกรมของเมธอร์ดของคลาสโลเคชัน ซึ่งประกอบด้วยบริการเพียงบริการเดียวเท่านั้น ดังแสดงในรูปที่ ก.42



รูปที่ ก.42 แอคทิวิตีไดอะแกรมของเมธอร์ดของคลาสโลเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปจะเป็นแอกทิวิตีไดอะแกรมของเมธอร์ดของคลาส โมเดล ซึ่งประกอบด้วยบริการเพียง 2 เมธอร์ดเท่านั้น ดังแสดงในรูปที่ ก.43

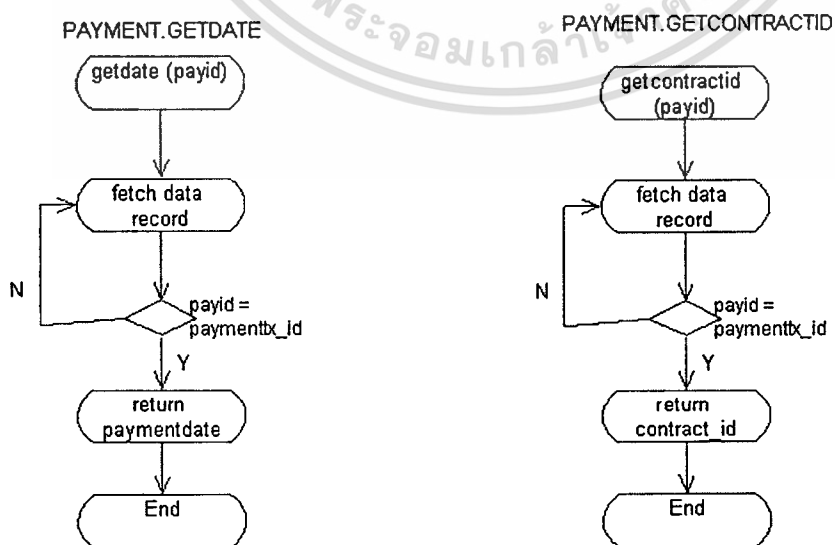


(ก) เมธอร์ด “Gettype”

(ข) เมธอร์ด “Getname”

รูปที่ ก.43 แอกทิวิตีไดอะแกรมของเมธอร์ดของคลาสโมเดล

ต่อไปจะเป็นแอกทิวิตีไดอะแกรมของเมธอร์ดของคลาสเพเมนต์ ซึ่งประกอบด้วยบริการจำนวน 5 บริการด้วยกัน ดังแสดงในรูปที่ ก.44



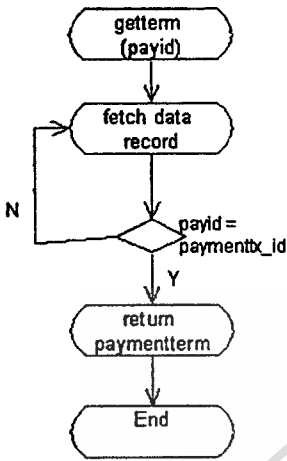
(ก) เมธอร์ด “Getdate”

(ข) เมธอร์ด “Getcontractid”

รูปที่ ก.44 แอกทิวิตีไดอะแกรมของเมธอร์ดของคลาสเพเมนต์

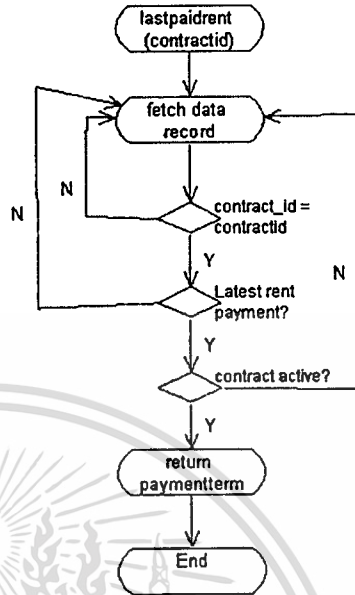
นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PAYMENT.GETTERM



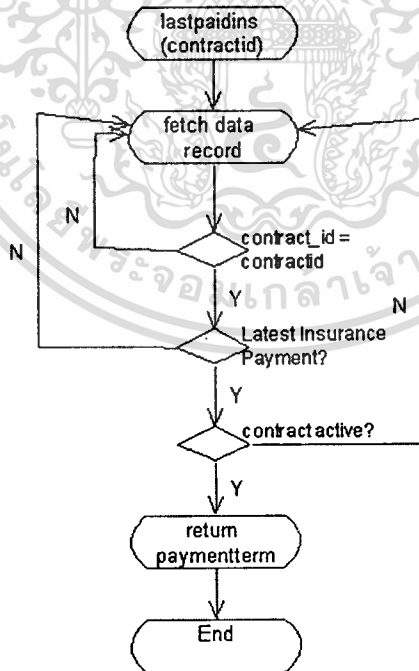
(ค) เมธอร์ด “Getterm”

PAYMENT.LASTPAIDRENT



(ง) เมธอร์ด “Lastpaidrent”

PAYMENT.LASTPAIDINS

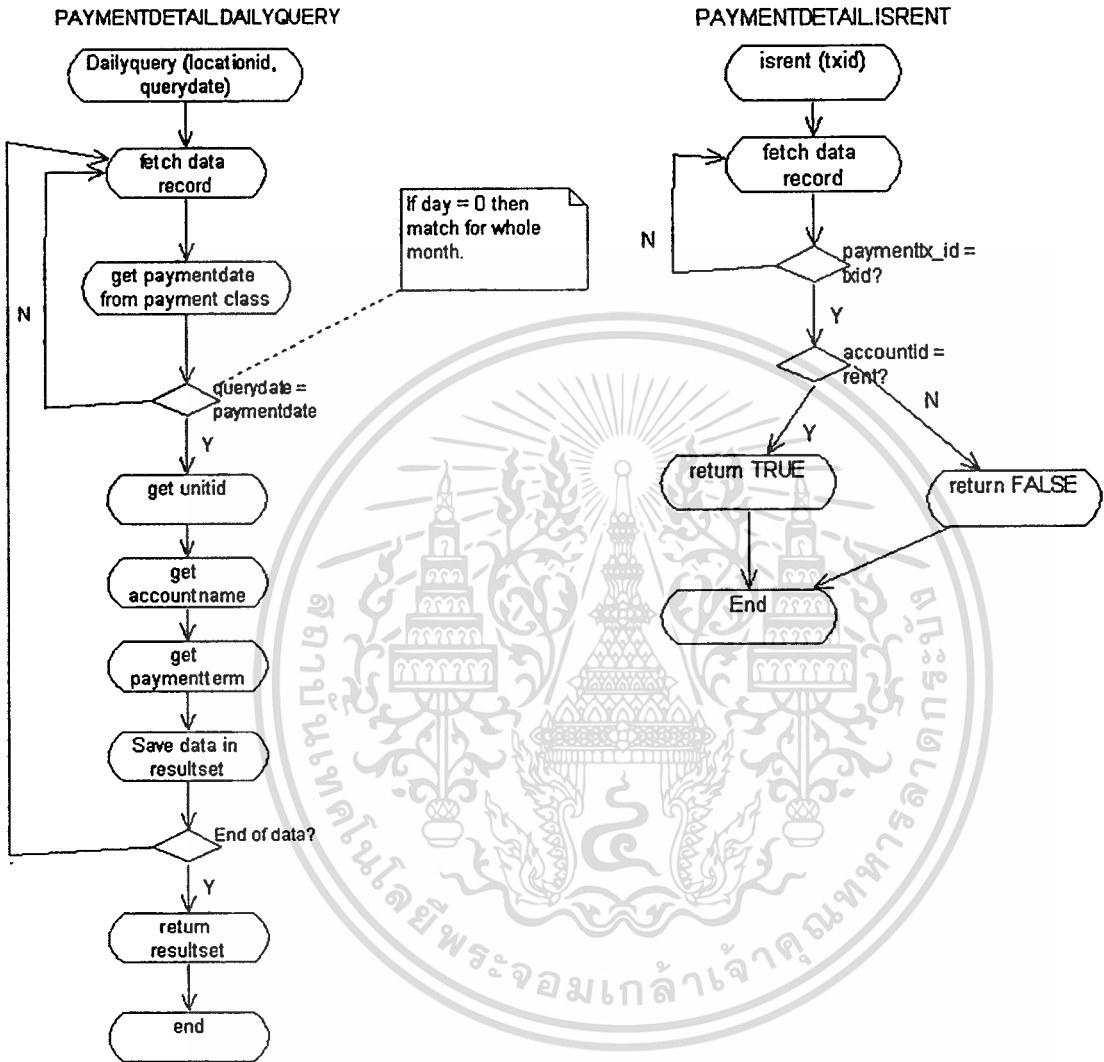


(จ) เมธอร์ด “Lastpaidins”

รูปที่ ก.44 (ต่อ) แอคทิวิตีไดอะแกรมของเมธอร์ดของคลาสเพเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

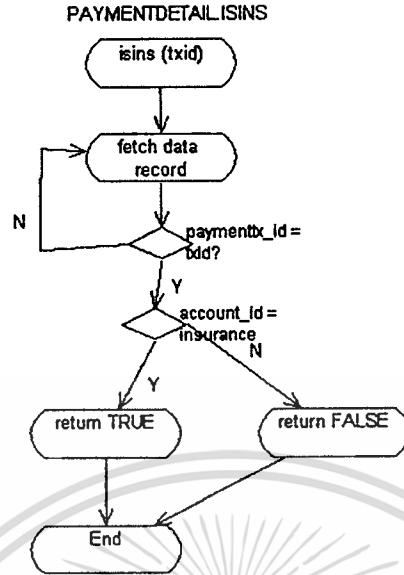
ต่อไปจะเป็นแอกทวิตีโคอะแกรมของเมธอร์คของคลาสเพเมนต์ดีเทล ซึ่งประกอบด้วย 3 เมธอร์ค ทำให้มี 3 แอกทวิตีโคอะแกรมดังแสดงในรูป ก.45



(ก) เมธอร์ค “Dailyquery”

(ข) เมธอร์ค “Isrent”

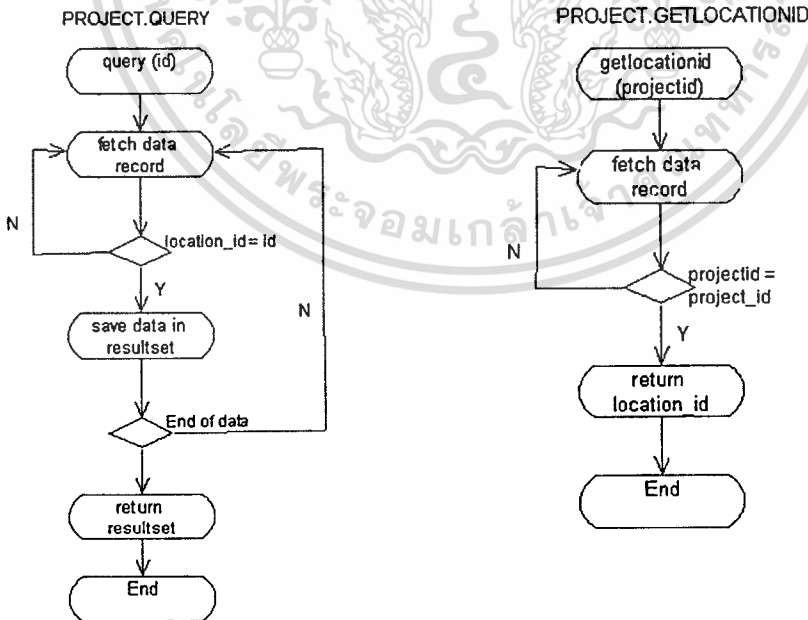
รูปที่ ก.45 แอกทวิตีโคอะแกรมของเมธอร์คของคลาสเพเมนต์ดีเทล



(ค) เมธอร์ค “Isins”

รูปที่ ก.45 (ต่อ) แอคทิวิตีไดอะแกรมของเมธอร์คของคลาสเพเมนต์ดีเทล

ต่อไปจะเป็นแอคทิวิตีไดอะแกรม ของเมธอร์คของคลาส โปรเจค ซึ่งมีบริการประกอบด้วย 3 เมธอร์ค และแสดงออกด้วย 3 แอคทิวิตีไดอะแกรมดังแสดงในรูป ก.46

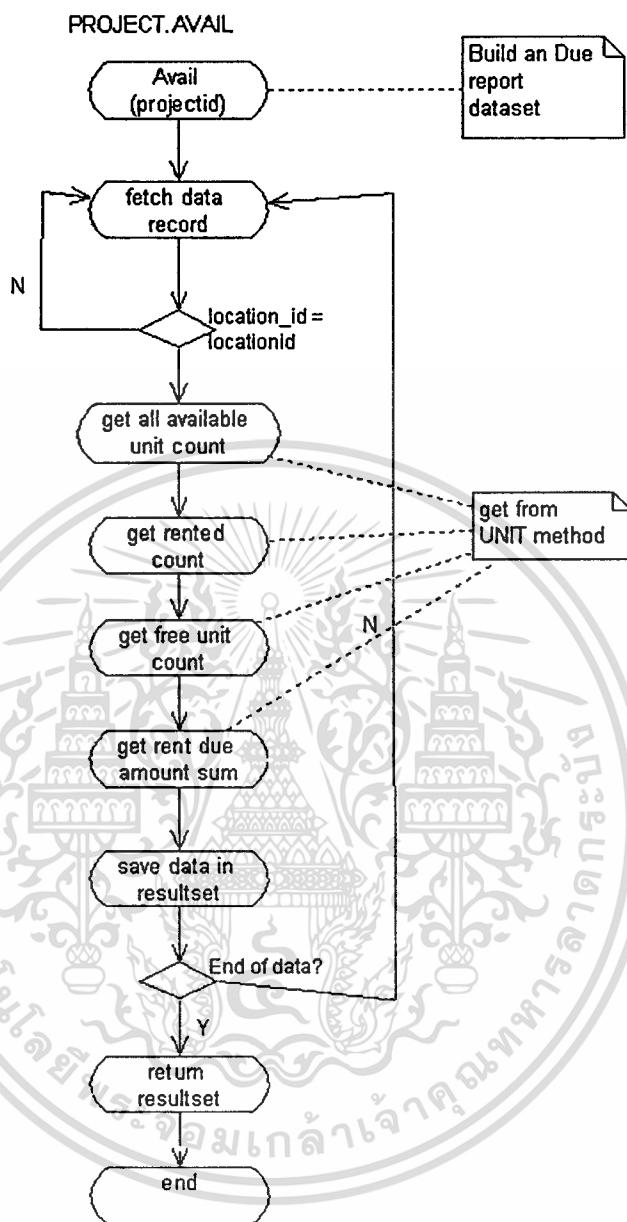


(ก) เมธอร์ค “Query”

(ข) เมธอร์ค “Getlocationid”

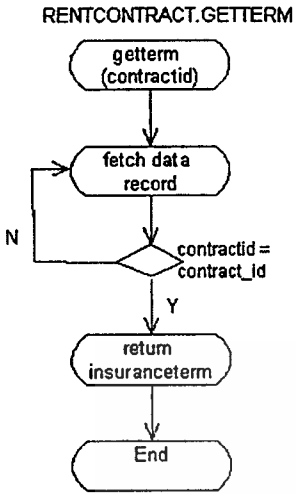
รูปที่ ก.46 แอคทิวิตีไดอะแกรมของเมธอร์คของคลาสโปรเจค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

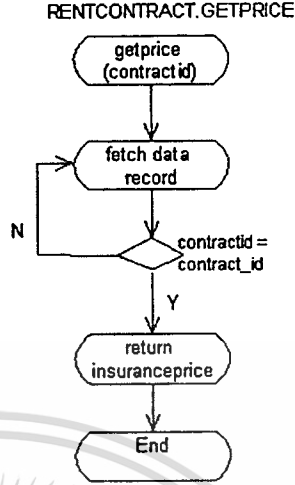


รูปที่ ก.46 (ต่อ) แอคทิวิตีไดอะแกรมของเมธอร์ดของคลาสโปรเจค

ต่อไปจะเป็นแอคทิวิตีไดอะแกรมของเมธอร์ดของคลาสเรนท์คอนแทรค ซึ่งประกอบด้วย เมธอร์ดทั้งสิ้น 2 เมธอร์ด ดังนั้นจึงสามารถอธิบายด้วย 2 แอคทิวิตีไดอะแกรม ดังแสดงในรูป ก.47



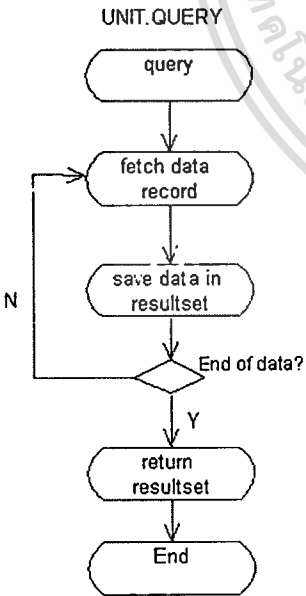
(ก) เมธอร์ค “Getterm”



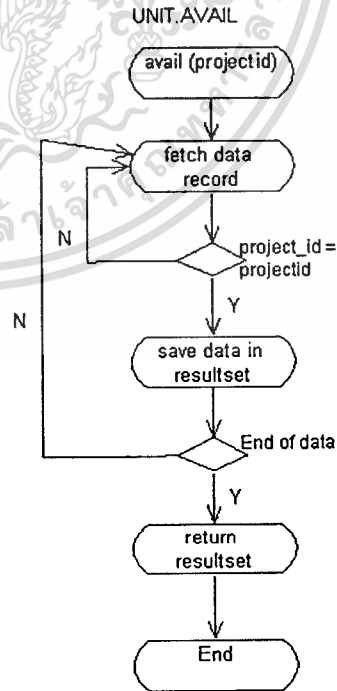
(ข) เมธอร์ค “Getprice”

รูปที่ ก.47 แอคทิวิตีไดอะแกรมของเมธอร์คของคลาสเรนท์คอนแทรค

ต่อไปจะเป็นแอคทิวิตีไดอะแกรมของเมธอร์คของคลาสสุดท้ายคือคลาสยูนิต ซึ่งเป็นคลาสขนาดใหญ่เช่นกัน โดยมีจำนวนเมธอร์คถึง 8 เมธอร์ค และมีแอคทิวิตีไดอะแกรม 8 ไดอะแกรม ดังแสดงในรูปที่ ก.48



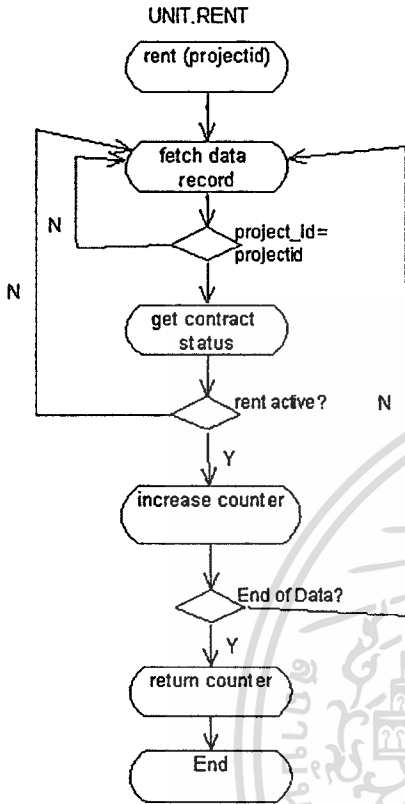
(ก) เมธอร์ค “Query”



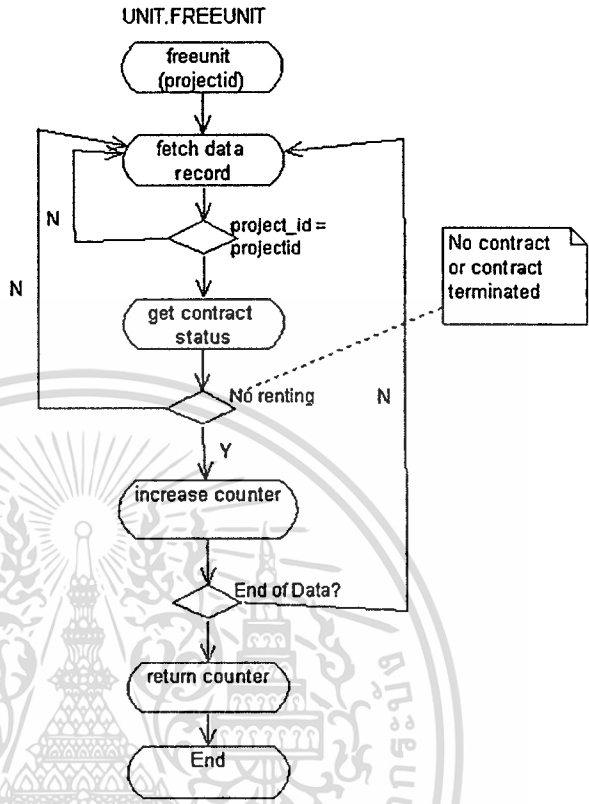
(ข) เมธอร์ค “Avail”

รูปที่ ก.48 แอคทิวิตีไดอะแกรมของเมธอร์คของคลาสยูนิต

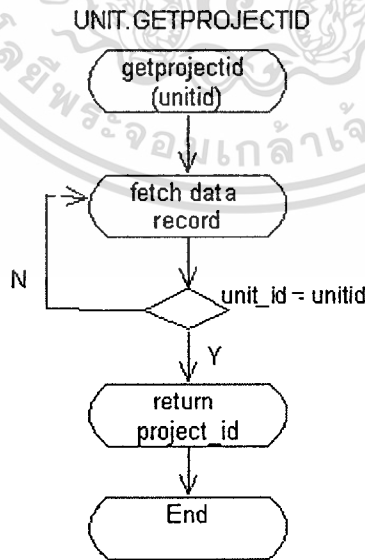
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค) เมธอร์ค “Rent”



(ง) เมธอร์ค “Freeunit”



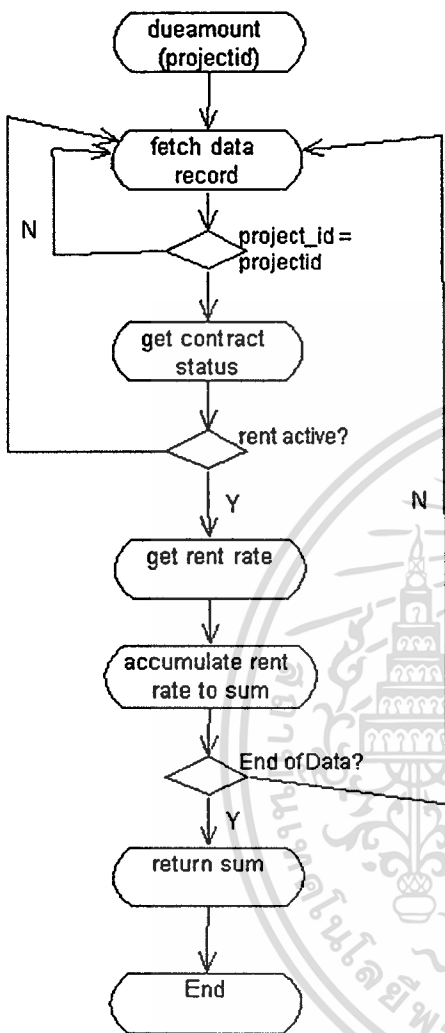
(จ) เมธอร์ค “Getprojectid”

รูปที่ ก.48 (ต่อ) แอคทีวิตี้ไดอะแกรมของเมธอร์คของคลาสยูนิต

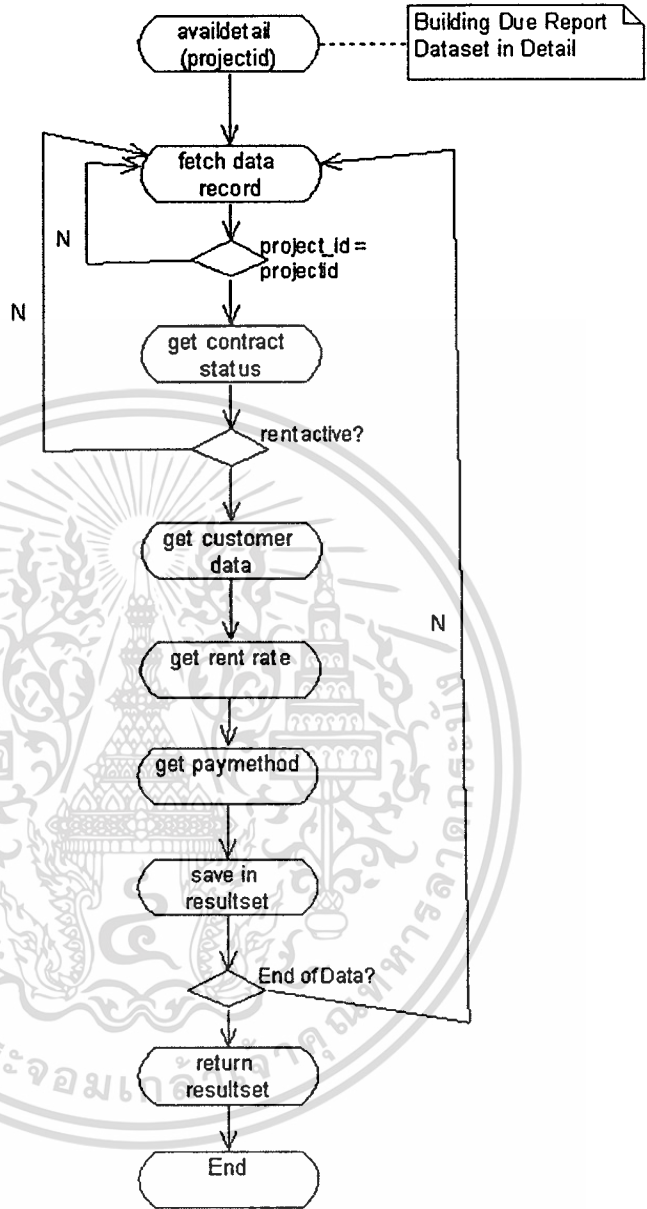
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UNIT.DUEAMOUNT



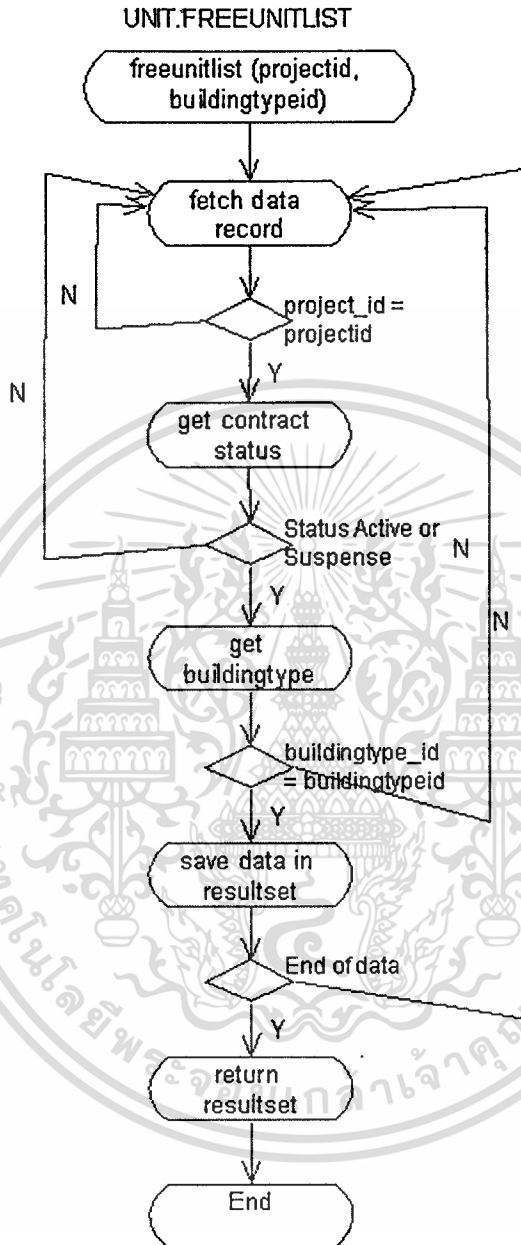
UNIT.AVAILEDDETAIL



(ก) เมธอร์ด “Dueamount”

(ข) เมธอร์ด “Availedetail”

รูปที่ ก.48 (ต่อ) แอคทิวิตีไดอะแกรมของเมธอร์ดของคลาสยูนิต



(ข) เมธอร์ด “Freeunitlist”

รูปที่ ก.48 (ต่อ) แดททีวี่ตีไดอะแกรมของเมธอร์ดของคลาสยูนิต

ภาคผนวก ข.

คู่มือการใช้ระบบ

ระบบบัญชีลูกหนี้เช่า การเคหะแห่งชาติ นี้เป็นระบบโปรแกรมที่ทำงานบนระบบปฏิบัติการ ไมโครซอฟต์ วินโดวส์ 95 / 98 / เอ็นที โดยเป็นระบบติดต่อกับผู้ใช้แบบวินโดวส์ที่ทำงานโดยอาศัยการใช้งานเมาส์และคีย์บอร์ดร่วมกัน ประกอบด้วยชุดของหน้าต่างจำนวนหนึ่ง และชุดของรายงานอีกจำนวนหนึ่ง

ข.1 การติดตั้งทางด้านเซิร์ฟเวอร์

ระบบฐานข้อมูลโอราเคิลนั้น จะต้องทำงานบนระบบปฏิบัติการวินโดวส์เอ็นทีเท่านั้น ดังนั้น เครื่องที่จะติดตั้งนั้นจะต้องมีคุณสมบัติที่จำเป็นสำหรับการติดตั้งระบบปฏิบัติการก่อน และเพิ่มเติมสำหรับการติดตั้งโอราเคิลคือหน่วยความจำ ที่ต้องการเพิ่มเป็นพิเศษ อย่างน้อย 64 เมกะไบต์ และจะทำงานได้ดีที่หน่วยความจำ 80 เมกะไบต์ ต้องการฮาร์ดดิสก์ประมาณ 400 เมกะไบต์เป็นอย่างน้อย และหน่วยประมวลผลระดับเพนเทียม 150 เมกะเฮิร์ตซ์ขึ้นไป

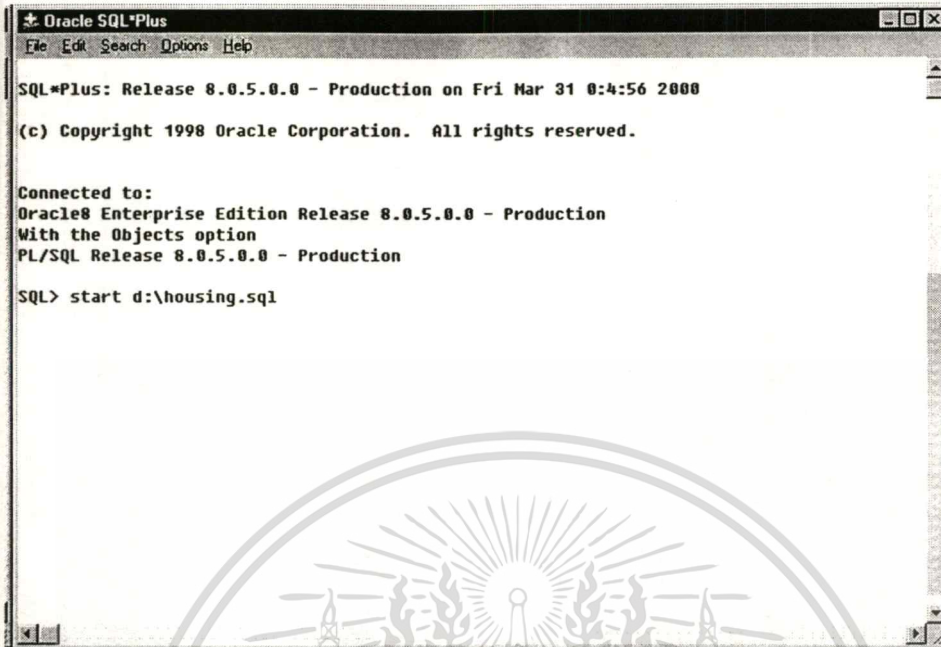
การติดตั้งระบบฐานข้อมูลโอราเคิลนั้นก็สามารถกระทำได้ตามคู่มือการติดตั้ง เมื่อมีระบบฐานข้อมูลที่สามารถใช้งานได้แล้ว หลังจากนั้นก็สามารถนำโปรแกรมสร้างโครงสร้างข้อมูลดังที่แสดงไว้ในส่วนโปรแกรมต้นฉบับสคริปต์ “HOUSING.SQL” เพื่อให้ระบบฐานข้อมูลสร้างโครงสร้างข้อมูลขึ้นมา โดยจะเรียกใช้โปรแกรม “SQL Plus 8.0” ซึ่งเป็นส่วนหนึ่งของระบบฐานข้อมูล โดยเรียกผ่านคำสั่ง start [drive:][path]\HOUSING.SQL ดังรูปที่ ข.1

ที่จุดนี้ก็จะมีชุดฐานข้อมูลที่พร้อมทำงานทันที สิ่งที่ต้องทำต่อไปก็คือการสร้างจุดเชื่อมจากภายนอก ซึ่งสามารถทำได้ที่โปรแกรม “Net8 Easy Config” โดยมีขั้นตอนการติดตั้งตามคู่มือการติดตั้งเครือข่ายของโอราเคิล โดยวัตถุประสงค์ก็เพื่อสร้างจุดที่ให้โปรแกรมภายนอกสามารถเข้ามาใช้บริการได้ โดยต้องใช้ชุดโปรโตคอล “TCP/IP” ดังแสดงในรูปที่ ข.2

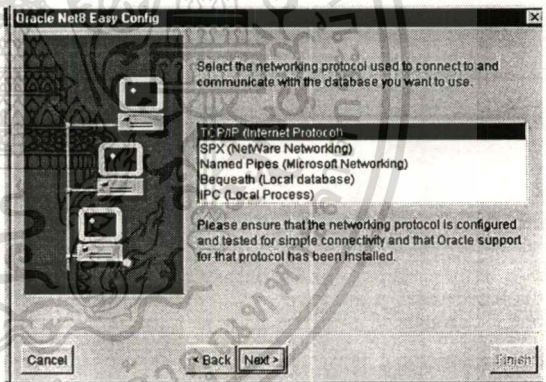
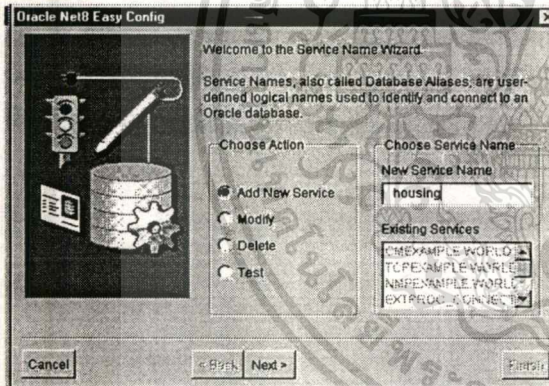
จากนั้นส่วนสุดท้ายที่ต้องติดตั้งทางด้านเซิร์ฟเวอร์ก็คือการติดตั้งสโตร์โพรซีเจอร์ โดยวิธีการที่แนะนำก็คือ ควรจะติดตั้งผ่านโปรแกรมชุดเดเวลลอปเปอร์ ก็คือต้องติดตั้งโปรแกรมชุดเดเวลลอปเปอร์ก่อน และเปิดเรียกโปรแกรมที่เก็บชุดคำสั่งสโตร์โพรซีเจอร์ไว้ และเคลื่อนย้ายไปที่ด้านเซิร์ฟเวอร์ โดยลากโปรแกรมทั้งหมดจากหัวข้อ “Program Unit” ไปไว้ที่ “Store Procedure” ของด้านเซิร์ฟเวอร์ ก็จะสามารมีชุดสโตร์โพรซีเจอร์ที่สามารถทำงานได้จริง พร้อมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

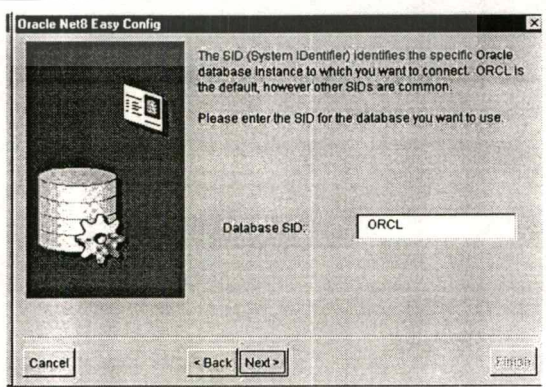
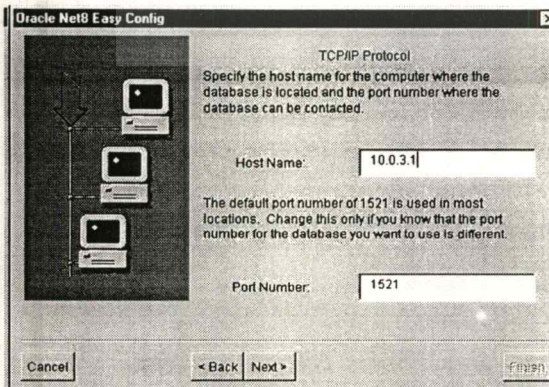


รูปที่ ข.1 การใช้โปรแกรม “SQL Plus 8.0” สร้างโครงสร้างข้อมูล



(ก) เพิ่มบริการใหม่

(ข) เลือกชนิดโปรโตคอล



(ค) กำหนดชื่อเซิร์ฟเวอร์

(ง) เลือกฐานข้อมูลต้นทาง

รูปที่ ข.2 การใช้โปรแกรม “Net8 Easy Config” สร้างการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 การติดตั้งทางด้านไคลเอ็นท์

สิ่งที่จำเป็นสำหรับการติดตั้งด้านไคลเอ็นท์นั้นก็จะต้องติดตั้ง โปรแกรมโอราเคิลไคลเอ็นท์ และโปรแกรมรันไทม์ของโอราเคิล เดเวลลอปเปอร์ ฟอรัมและรีพอร์ท ซึ่งทั้งหมดสามารถติดตั้งได้จากโปรแกรมโอราเคิลเดเวลลอปเปอร์ โดยเพียงเลือกทางเลือกในการติดตั้งเป็น "Deployment" ระบบก็จะติดตั้งส่วนประกอบที่จำเป็นทั้งหมดเข้าสู่เครื่อง

สิ่งที่ต้องทำต่อไปก็คือการสร้างจุดเชื่อมต่อเข้ากับด้านเซิร์ฟเวอร์ โดยการคัดลอกไฟล์ชื่อ "TNSNAMES.ORA" จากไดเรกทอรี [oracle directory]\NET80\ADMIN ของเครื่องเซิร์ฟเวอร์ไปไว้ที่ ไดเรกทอรีเดียวกันของเครื่องไคลเอ็นท์ เพื่อเป็นการสร้างการอ้างอิงเดียวกัน ก็จะสามารถเชื่อมต่อจากเครื่องไคลเอ็นท์เข้าสู่เครื่องเซิร์ฟเวอร์ได้ โดยสามารถทดสอบการเชื่อมต่อได้ที่โปรแกรม "SQL Plus" ของเครื่องไคลเอ็นท์

หลังจากนั้นก็คัดลอกไฟล์ที่เป็น โมดูลของโปรแกรมมาเก็บไว้ที่ไดเรกทอรีที่จะใช้งานของเครื่องไคลเอ็นท์ ก็พร้อมจะใช้งานทันที

ข.3 การใช้งานระบบงาน

เริ่มต้นการทำงานให้ไปที่ไดเรกทอรีที่เก็บ โปรแกรมโมดูลชุดระบบบัญชีลูกหนี้เข้าเอาไว้แล้วเลือกที่โปรแกรม "MAIN.FMX" ก็จะมีหน้าต่างให้ทำการล็อกเข้าสู่ระบบ ก็ให้พิมพ์ชื่อและรหัสผ่านเข้าไป ก็จะปรากฏหน้าต่างดังรูปที่ ข.3

รูปที่ ข.3 หน้าต่างโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหน้าต่างโปรแกรมหลักทางด้านซ้าย จะเป็นส่วนที่ให้ผู้ใช้งานสามารถดูสถานะการเชื่อมต่อกับเซิร์ฟเวอร์ได้ว่า ผู้ใช้คนใดกำลังเชื่อมต่อกับด้านเซิร์ฟเวอร์ และผู้ใช้งานสามารถเลือกที่จะเปลี่ยนการเชื่อมต่อได้โดยพิมพ์ชื่อและรหัสผ่านที่ช่องทางด้านล่างลงมา และกดปุ่ม “เข้าสู่ระบบ” เพื่อเปลี่ยนการเชื่อมต่อ นอกจากนี้ผู้ใช้งานยังสามารถออกจากระบบได้โดยกดปุ่ม “ออกจากระบบ” เพื่อจะปิดการทำงาน และออกจากระบบ

การทำงานของหน้าต่างหลักนี้ จะเป็นส่วนเชื่อมต่อเข้าสู่ระบบย่อยต่าง ๆ โดยมีทั้งหมด 7 ระบบย่อย และสามารถเข้าสู่ระบบย่อยต่าง ๆ ได้โดยการกดปุ่มตามชื่อระบบย่อยที่ต้องการจะทำงาน โดยจะเริ่มจากระบบย่อยแรก คือระบบย่อย “ใส่ข้อมูลการชำระเงิน” จะปรากฏหน้าต่างดังรูปที่ ข.4 ซึ่งเป็นหน้าต่างของระบบย่อยใส่ข้อมูลการชำระเงิน

หน้าต่าง WINDOW1

นำข้อมูลการรับชำระเข้า

เขตควบคุม

รหัสเขต 1

ชื่อเขต bangkai region

ก่อนหน้า ต่อไป

ลูกค้า

ชื่อ Frijob

นามสกุล Capra

รายละเอียดสัญญา

ค่าเช่า 1000

เลขที่สัญญา 1001000

ก่อนหน้า ต่อไป

ข้อมูลการชำระเงิน

วันที่ชำระเงิน 01 03 1999

เลขที่ชำระเงิน 1

งวดที่จ่าย 03 1999

ก่อนหน้า ต่อไป

รายการ

จำนวนเงิน	เลขที่ชำระเงิน	รายการที่ชำระ
1000	1	rent
21	1	insurance
10	1	water

จัดเก็บข้อมูล

กลับสู่รายการหลัก

รูปที่ ข.4 หน้าต่างระบบย่อยใส่ข้อมูลการชำระเงิน

โดยที่ด้านบนสุดผู้ใช้งานต้องเลือกก่อนว่าต้องการจะชำระเงินให้กับเขตควบคุมใด คือการเช่าของตุนั้นอยู่ในเขตควบคุมใด ก็สามารถเลือกเขตควบคุมที่ต้องการได้โดยการกดปุ่มเลือก “ก่อนหน้า” และ “ต่อไป” เพื่อเลือกเขตควบคุมที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นก็เลือกซื้อลูกค้ารายที่จะเข้ามาชำระเงิน โดยสามารถเลือกซื้อลูกค้ารายที่ต้องการ ได้ด้วยการกดปุ่ม “ก่อนหน้า” และ “ต่อไป” อีกเช่นกัน จนกว่าจะเจอลูกค้ารายที่ต้องการ เมื่อได้ลูกค้ารายที่ต้องการแล้ว ก็จะเลือกสัญญาที่ต้องการจะชำระเงิน เพราะมีความเป็นไปได้ที่ลูกค้ารายเดียวกันจะมีสัญญามากกว่าหนึ่งสัญญา ซึ่งก็หมายความว่าลูกค้านั้นเข้ามามากกว่าหนึ่งหน่วยอาศัย

เมื่อได้สัญญาที่ต้องการแล้ว ก็มาป้อนที่ช่อง “ข้อมูลการชำระเงิน” โดยป้อนวันที่ทำการชำระเงินก่อน เป็นวันที่ เดือน และปี ทั้งหมดเป็นตัวเลขทั้งสิ้นเพื่อความสะดวก และป้อน เลขที่การชำระเงิน จากนั้นก็ป้อนงวดที่จ่าย ซึ่งจะมีเฉพาะเดือนและปีเท่านั้น ทั้งหมดเป็นตัวเลขเช่นกัน

จากนั้นก็จะเป็นลงรายการการชำระเงินว่ามีค่าอะไรบ้าง โดยจะทำงานที่ส่วน “รายการ” ส่วนแรกที่จะต้องใส่ก็คือจำนวนเงินที่เป็นค่าใช้จ่าย ใส่ในช่องจำนวนเงิน จากนั้นก็ใส่หมายเลขการชำระเงิน ซึ่งต้องตรงกับค่านับน จากนั้นก็เลือกรายการการชำระเงินว่าเป็นค่าใช้จ่ายอะไร โดยเลือกจากคอมโบบ็อกซ์ และวนทำซ้ำจนครบทุกรายการ เมื่อเสร็จสิ้นงานแล้ว ก็กดปุ่ม “จัดเก็บข้อมูล” เพื่อจัดเก็บข้อมูลลงฐานข้อมูล กดปุ่ม “กลับสู่รายการหลัก” เพื่อกลับไปหน้าจอต่างรูป ข.3

ข.4 การใช้ระบบย่อยหนึ่งถึงกำหนดชำระ

จากหน้าต่างรายการหลัก ถ้าต้องการเข้าสู่ระบบย่อยลูกหนี้ถึงกำหนดชำระ ก็สามารถเข้าได้ โดยการกดปุ่ม “ลูกหนี้ถึงกำหนดชำระ” ซึ่งก็จะปรากฏหน้าต่างดังรูป ข.5

ลูกหนี้ถึงกำหนดชำระ

เขตควบคุม

ชื่อเขต	bangkae region	ถนน	petkasame
หมู่	5	ซอย	thainantip
ตำบล	bangwa	อำเภอ	pasrichaoroen
จังหวัด	bangkok	รหัสไปรษณีย์	10160

ก่อนหน้า ต่อไป

สรุปข้อมูลกำหนดชำระของโครงการ

รหัสโครงการ	ชื่อโครงการ	จำนวนหน่วยรวม	หน่วยที่เข้า	รวมยอดค่าเช่า
1	Happy Village	11	5	5200
3	Delaware Project	8	0	0
4	Minor Hill	0	0	0

พิมพ์รายงาน พิมพ์ทุกหน่วยควบคุม รายละเอียด กลับรายการหลัก

รูปที่ ข.5 หน้าต่างระบบย่อยหนึ่งถึงกำหนดชำระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ ข.5 จะเห็นได้ว่า ระบบจะให้ข้อมูลลูกหนี้ที่ถึงกำหนดชำระโดยให้ผู้ใช้สามารถเลือกได้ว่าต้องการจะดูข้อมูลของเขตควบคุมใด ดังนั้นสิ่งแรกที่ต้องทำก็คือการเลือกเขตควบคุมที่ต้องการจะดูข้อมูล โดยกดปุ่ม “ก่อนหน้า” และ “ต่อไป” เพื่อเลือกเขตควบคุมที่ต้องการดูข้อมูล

เมื่อเลือกได้เขตควบคุมที่ต้องการแล้วที่ด้านล่าง ก็จะแสดงข้อมูลเป็นรายโครงการที่อยู่ในเขตควบคุมที่เลือก โดยจะแสดงชื่อโครงการ จำนวนหน่วยอาศัยที่มีทั้งหมด จำนวนหน่วยที่ถูกเช่า และยอดค่าเช่ารวมที่จะต้องได้รับในงวดปัจจุบัน ที่หน้าค่างนี้ผู้ใช้สามารถพิมพ์ข้อมูลออกเป็นรายงานได้โดยการกดปุ่ม “พิมพ์รายงาน” ถ้าต้องการให้สร้างรายงานทุกหน่วยควบคุมก็ให้เลือกที่เช็คบ็อกซ์ “พิมพ์ทุกหน่วยควบคุม” แต่ถ้าต้องการพิมพ์เพียงหน่วยควบคุมเดียว ก็ไม่ต้องเลือก หลักจากนั้นระบบก็จะสร้างรายงานให้ดังรูป ข.6

การเคหะแห่งชาติ
รายงานลูกหนี้เช่าถึงกำหนดชำระ
วันที่ออกรายงาน: 30 มี.ค. 2000 / 02:39

รหัสหน่วยควบคุม 1 หน่วยควบคุม bangkae region
รหัสโครงการ 1 ชื่อโครงการ Happy Village

รหัสหน่วยอาศัย	รหัสสัญญา	วิธีการชำระเงิน	สถานะสัญญา	วันที่ทำสัญญา	จำนวนงวด	ยอดผ่อนคงขาด
100001	1001000	cash	1	01 ม.ค. 1999	50	1000
100002	100202	cash	1	12 ก.พ. 1999	60	1000
100003	102001	cash	1	11 พ.ธ. 1999	60	1000
100005	1020009	bank	1	10 ม.ค. 1999	15	1000
101001	902001	bank	1	01 ม.ค. 1999	14	1200
จำนวนหน่วยอาศัยต่อโครงการ 5					ยอดชำระต่อโครงการ 5200	
จำนวนหน่วยอาศัยทั้งหมด 5					ยอดชำระต่อหน่วยควบคุม 5200	
จำนวนหน่วยอาศัยรวม 5					ยอดชำระทั้งหมด 5200	

รูปที่ ข.6 รายงานหนี้ถึงกำหนดชำระ

จากรูปที่ ข.6 ก็จะแสดงรายงานเป็นรายโครงการ และแสดงรายละเอียดภายในแต่ละโครงการว่ามีหน่วยอาศัยใดบ้างที่ถึงกำหนดชำระค่าเช่าในงวดเดือนปัจจุบัน จากหน้าต่าง ข.5 ถ้าผู้ใช้ต้องการดูรายละเอียดในโครงการใด ก็ให้เอาเมาส์ไปเลือกที่บรรทัดที่ต้องการ ซึ่งเป็นบรรทัดของโครงการที่ต้องการดูข้อมูล จากนั้นก็กดปุ่ม “รายละเอียด” ซึ่งก็จะแสดงรายละเอียดภายในโครงการนั้นออกมาดังรูปที่ ข.7

WINDOW1

รายละเอียดลูกหนี้ถึงกำหนดชำระของโครงการ

โครงการ

รหัสโครงการ หน่วยเช่า 5

ชื่อโครงการ Happy Village ยอดค่าเช่ารวม 5200

จำนวนหน่วยรวม 11

ก่อนหน้า ต่อไป

รหัสหน่วยอาศัย	กาน้ำชื่อ	ชื่อลูกค้า	นามสกุล	ค่าเช่า	วิธีชำระค่าเช่า
100001	Mr.	Frijob	Capra	1000	cash
100002	Mr.	Alexandre	Dumas	1000	cash
100003	Mrs.	Michael	Crichton	1000	cash
100005	Mr.	Frijob	Capra	1000	bank
101001	Mr.	Victor	Hugo	1200	bank

กลับสู่รายการก่อนหน้า

รูปที่ ข.7 หน้าต่างรายละเอียดลูกหนี้ถึงกำหนดชำระของโครงการ

จากรูปที่ ข.7 จะเห็นได้ว่าที่ด้านบนจะให้ผู้ใช้สามารถเปลี่ยนโครงการที่ต้องการดูข้อมูลได้ โดยจะเปลี่ยนได้เฉพาะโครงการที่อยู่ในเขตควบคุมที่เลือกไว้ที่หน้าต่างก่อนหน้านี้ ที่ส่วนบนซึ่งเป็นส่วนรายละเอียดของโครงการ ก็จะให้ข้อมูลโดยรวมไว้ เช่น ชื่อโครงการ ยอดหนี้รวม จำนวนหน่วยอาศัย และจำนวนหน่วยอาศัยที่ถูกเช่า เป็นต้น

ที่ด้านล่างก็จะเป็นรายละเอียดเป็นรายหน่วยอาศัย ว่าแต่ละหน่วยอาศัยที่มีกำหนดจะต้องชำระค่าเช่านั้นมีรายละเอียดเป็นอย่างไร เช่น มีรหัสหน่วยอาศัย เป็นเท่าใด ลูกค้าที่เป็นเจ้าของสัญญาชื่ออะไร และมีค่าเช่าเท่าใด ชำระเงินด้วยวิธีใด ซึ่งจะให้ข้อมูลจนครบทุกหน่วยอาศัยของโครงการนั้น ผู้ใช้สามารถเลื่อนดูข้อมูลได้โดยเปลี่ยนที่แถบเลื่อนด้านข้าง

เมื่อเสร็จสิ้นกิจกรรมทั้งหมดแล้วสามารถกดปุ่ม “กลับสู่รายการก่อนหน้า” เพื่อกลับไปสู่นำหน้าต่างก่อนหน้านี้ได้ และกดปุ่ม “กลับสู่รายการหลัก” เพื่อกลับสู่หน้าต่างหลักที่ควบคุมระบบ

ข.5 การใช้ระบบย่อยรายละเอียดสัญญา

ระบบย่อยต่อมาก็คือระบบย่อยรายละเอียดสัญญา ซึ่งสามารถเข้าใช้งานได้โดยการกดปุ่ม “รายละเอียดสัญญา” ที่รายการหลัก ก็จะปรากฏหน้าต่างระบบย่อยรายละเอียดสัญญาดังรูปที่ ข.8

The screenshot shows a software window titled "รายละเอียดสัญญา" (Contract Details) with the following fields and values:

ลูกค้า (Customer)	
รหัสลูกค้า (Customer Code)	100
ที่อยูปัจจุบัน (Current Address)	47/1 Petkasame
ชื่อนามสกุล (Name and Surname)	Mr. Frijob Capra
ที่อยู่ตามทะเบียน (Registered Address)	64/2 Sukhumwit
ชื่อ (Name)	Frijob
โทรศัพท์ (Phone Number)	3488383
นามสกุล (Surname)	Capra
<input type="button" value="ก่อนหน้า"/> <input type="button" value="ต่อไป"/>	

รายละเอียดสัญญา (Contract Details)	
รหัสสัญญา (Contract Code)	1001000
รหัสหน่วยอาศัย (Apartment Code)	100001
วันที่ทำสัญญา (Contract Date)	01 ม.ค. 1999
สถานที่สัญญา (Contract Location)	1
วิธีชำระเงิน (Payment Method)	cash
งวดแรก (First Installment)	01 ก.พ. 1999
งวดสุดท้าย (Last Installment)	01 มี.ค. 2003
จำนวนงวด (Number of Installments)	50
ค่าเช่า (Rent)	1000
จำนวนงวดประกัน (Number of Guarantee Installments)	3
ค่าประกัน (Guarantee Fee)	21
<input type="button" value="สัญญาก่อนหน้า"/> <input type="button" value="สัญญาต่อไป"/>	

Buttons at the bottom of the window:

- (Print Report)
- พิมพ์ลูกค้าทั้งหมด (Print All Customers)
- (Return to Main List)

รูปที่ ข.8 หน้าต่างระบบย่อยรายละเอียดสัญญา

การทำงานของระบบย่อยนี้ก็คือ จะเป็นการดูรายละเอียดสัญญา โดยมองจากลูกค้าเป็นหลัก ดังนั้นสิ่งที่ทำก็คือการเลือกลูกค้าที่ต้องการจะดูรายละเอียดสัญญา โดยสามารถกดปุ่ม “ก่อนหน้า” และ “ต่อไป” เพื่อเลือกลูกค้ารายที่ต้องการ ที่ส่วนข้อมูลลูกค้าก็จะมีรายละเอียดของลูกค้ารายนั้นด้วย เช่นชื่อ นามสกุล ที่อยู่ติดต่อ และเบอร์โทรศัพท์ เป็นต้น

เมื่อได้ลูกค้ารายที่ต้องการแล้ว ก็จะมาเลือกสัญญาที่ด้านล่าง เพราะเนื่องจากลูกค้าหนึ่งท่านสามารถทำสัญญาได้มากกว่าหนึ่งฉบับ นั่นคือ ลูกค้าหนึ่งท่านสามารถเช่าหน่วยอาศัยได้มากกว่าหนึ่งหน่วย ดังนั้นที่ด้านล่างจึงต้องเลือกสัญญารับที่ต้องการโดยกดปุ่ม “สัญญาก่อนหน้า” และ “สัญญาต่อไป” เพื่อเลือกสัญญา เมื่อได้สัญญาที่ต้องการแล้ว ส่วนรายละเอียดสัญญาก็จะให้รายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WINDOW1

หน่วยอาสาสมัคร

เขตควบคุม

รหัสเขต ตำบล

ชื่อเขต อำเภอ

หมู่ จังหวัด

ถนน รหัสไปรษณีย์

ซอย

หน่วยอาสาสมัครตามโครงการ

รหัสโครงการ	ชื่อโครงการ	รหัสเขต	หน่วยอาสาสมัคร
2	More Happy	2	9

พิมพ์ทุกหน่วยควบคุม

รูปที่ ข.10 หน้าต่างระบบย่อยหน่วยอาสาสมัคร

ระบบย่อยนี้จะเริ่มทำงานโดยให้ข้อมูลรวมในระดับโครงการก่อน โดยผู้ใช้ต้องเลือกเขตควบคุมที่ต้องการดูข้อมูลโดยกดปุ่ม “ก่อนหน้า” และ “ต่อไป” เพื่อเลือกเขตควบคุมที่ต้องการ และที่ช่องข้อมูลเขตควบคุมก็จะให้ข้อมูลรายละเอียดของเขตควบคุมนั้น ๆ เช่น ที่ตั้ง เป็นต้น ที่ด้านล่างของหน้าต่าง ก็จะเป็นรายชื่อของโครงการที่อยู่ในเขตควบคุมนั้น โดยจะแสดงจำนวนหน่วยอาสาสมัครของโครงการนั้น ผู้ใช้สามารถร้องขอให้ระบบสร้างรายงานหน่วยอาสาสมัครของโครงการนั้นได้ โดยกดปุ่ม “พิมพ์รายงาน” และถ้าต้องการรายงานหน่วยอาสาสมัครของทุกหน่วยควบคุม ก็สามารถเลือกได้ที่เช็คบ็อกซ์ “พิมพ์ทุกหน่วยควบคุม” และกดปุ่ม “พิมพ์รายงาน” ระบบก็จะสร้างรายงานให้ดังรูป ข.11

freerep: Previewer
แฟ้ม มุมมอง วิดีโอ

รายงานหน่วยอาศัยว่าง
วันที่ออกรายงาน : 30 มี.ค. 2000 / 02:44

รหัสเขต 1 ชื่อเขต bangkae region

รหัสโครงการ 1 ชื่อโครงการ Happy Village

รหัสหน่วยอาศัย	ชั้นที่	อาคารที่	ชนิดอาคาร
100004	0	0	standard house
101002	0	0	standard house
101003	0	0	standard house
101004	0	0	standard house
102001			standard house
102002			standard house

จำนวนหน่วยอาศัยทั้งโครงการ 6

รหัสโครงการ 3 ชื่อโครงการ Delaware Project

รหัสหน่วยอาศัย	ชั้นที่	อาคารที่	ชนิดอาคาร
300100			standard house
300101			standard house
300102			standard house
300201	3	1	apartment
300202	3	1	apartment
300203	4	1	apartment
300204	4	2	apartment
300205	4	2	apartment

จำนวนหน่วยอาศัยทั้งโครงการ 8

จำนวนหน่วยอาศัยทั้งหมด 14

รูปที่ ข.11 รายงานหน่วยอาศัยว่าง

จากนั้นเมื่อต้องการดูรายละเอียดหน่วยอาศัยว่างของแต่ละโครงการก็สามารถทำได้ โดยการเลือกที่บรรทัดของโครงการที่ต้องการ โดยการเอาเมาส์ไปคลิกที่บรรทัดนั้น และกดปุ่ม “รายละเอียด” ซึ่งก็จะเปิดหน้าต่างรายละเอียดขึ้นมาดังรูปที่ ข.12

จากรูป ข.12 จะเห็นได้ว่าที่ด้านบนจะแสดงโครงการที่เลือกไว้จากหน้าต่างก่อนหน้านี้ และผู้ใช้สามารถเลือกที่จะเปลี่ยนไปดูโครงการอื่นในเขตควบคุมเดียวกันได้ โดยการกดปุ่ม “ก่อนหน้า” และ “ต่อไป” เมื่อได้โครงการตามที่ต้องการแล้ว ก็จะเลือกชนิดอาคารที่ส่วนล่างถัดลงมา โดยวัตถุประสงค์ของการเลือกก็เพื่อที่จะสามารถ เลือกแบบบ้านตามที่ถูกค้าต้องการ เช่นถ้าสนใจบ้านเดี่ยว ก็จะได้ดูเฉพาะหน่วยอาศัยว่างที่เป็นบ้านเดี่ยวเป็นต้น ผู้ใช้สามารถเลือกชนิดอาคารที่ต้องการได้โดยกดปุ่ม “ก่อนหน้า” และ “ต่อไป” อีกเช่นเดิม เมื่อได้ชนิดอาคารที่ต้องการแล้ว ที่ส่วนล่างสุด ก็จะเป็นรายการหน่วยอาศัยว่างที่ยังไม่มีเจ้าของ โดยจะแสดงข้อมูลคือ รหัสหน่วยอาศัย ชนิดอาคาร ชั้นที่ และ อาคารที่ ที่จุดนี้ผู้ใช้สามารถเลือกพิมพ์ออกเป็นรายงานได้โดยกดปุ่ม “พิมพ์รายงาน” และสามารถพิมพ์รายงาน ทุกโครงการ โดยเลือกที่เช็คบ็อกซ์ “พิมพ์ทุกโครงการ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง WINDDW1

หน่วยอาคารว่างตามโครงการ

โครงการ

รหัสโครงการ: ชื่อโครงการ: Happy Village

ชนิดอาคาร

รหัสชนิดอาคาร: 1 ชื่อชนิดอาคาร: standard house

หน่วยอาคาร

รหัสหน่วยอาคาร	ชนิดอาคาร	ชั้นที่	อาคารที่
100004	standard house	0	0
101002	standard house	0	0
101003	standard house	0	0

พิมพ์ทุกโครงการ

รูปที่ ข.12 หน้าต่างหน่วยอาคารว่างตามโครงการ

ซึ่งรายงานที่พิมพ์ออกมานี้ก็จะมีลักษณะเหมือนรายงานดังรูปที่ ข.11 แต่จะต่างกันก็เพียงจะแสดงเฉพาะโครงการที่สนใจเท่านั้น หรือพิมพ์เฉพาะเขตที่สนใจเท่านั้น ต่อจากนั้นเมื่อผู้ใช้สนใจในหน่วยอาคารใดเป็นพิเศษก็สามารถเลือกดูรายละเอียดหน่วยอาคารนั้นได้โดยการคลิกที่บรรทัดหน่วยอาคารที่ต้องการและกดปุ่ม “รายละเอียด” ซึ่งจะเปิดหน้าต่างดังรูป ข.13 ขึ้นมา

จากรูป ข.13 จะเห็นว่าเป็นรายละเอียดของหน่วยอาคารนั้น ๆ โดยที่ด้านบนสุดจะเป็นรายละเอียดของหน่วยอาคารนั้น ว่ามีรหัสเป็นเท่าใด อยู่ที่ชั้นใด อาคารใด เป็นแบบบ้านชนิดใด ผู้ใช้สามารถเลือกเปลี่ยนไปดูหน่วยอาคารอื่นได้โดยกดที่ปุ่ม “ก่อนหน้า” และ “ต่อไป” ถัดลงมาก็จะเป็นรายละเอียดของหน่วยอาคารนั้นว่ามีรายละเอียดเป็นอย่างไร เช่นมีกี่ห้องนอน กี่ห้องน้ำ เป็นต้น ซึ่งเป็นรายละเอียดที่ผูกติดอยู่กับแบบบ้านนั้นๆ และส่วนล่างสุดจะเป็นส่วนที่ให้ข้อมูลรายละเอียดของสัญญาเช่า หรือประวัติการเช่าก่อนหน้านี้ว่า หน่วยพักอาศัยนี้ถูกเช่ามากี่ครั้งแล้ว แต่ละครั้งเช่าเมื่อใด และสิ้นสุดเมื่อใด ใครเป็นผู้เช่า เป็นต้น

ผู้ใช้สามารถพิมพ์ประวัติการเช่าออกเป็นรายงานได้ถ้าต้องการ โดยการกดปุ่ม “พิมพ์รายงาน” ระบบก็จะสร้างเป็นรายงานขึ้นมาให้ โดยในรายงานจะให้รายละเอียดประวัติการเช่า เหมือนกับส่วนล่างสุดของหน้าต่าง ซึ่งรูปแบบรายงานจะเป็นไปดังรูป ข.14

หน้าต่าง WINDOW1

หน่วยอาศัยว่าง - รายละเอียดหน่วยอาศัย

หน่วยอาศัย

รหัสหน่วยอาศัย: 102002 ชั้นที่:

ชื่อรุ่นอาคาร: american house อาคารที่:

ชนิดอาคาร: standard house

คุณสมบัติของโมเดล

จำนวน	ส่วนประกอบ
4	bedroom
3	bathroom
3	floor
1	restroom

ประวัติสัญญา

เลขที่สัญญา	สถานะ	งวดแรก	งวดสุดท้าย	คำนำชื่อ	ชื่อลูกค้า	นามสกุล
920100	2	01 มี.ค. 1995	01 เม.ย. 1999	Mr. Frijob	Capra	
901001	2	01 พ.ค. 1996	01 มิ.ย. 2000	Mr. Alexandre	Dumas	

รูปที่ ข.13 หน้าต่างรายละเอียดหน่วยอาศัย

unitprofile: Previewer

แฟ้ม มุมมอง วิธียใช้

การลงทะเบียนชาติ
รายงานประวัติการทำสัญญา
วันที่ออกรายงาน 30 มี.ค. 2000 / 02:45

รหัสหน่วยอาศัย 102002 ชั้นที่ อาคารที่

รหัสสัญญา	วันที่ทำสัญญา	รหัสลูกค้า	ชื่อลูกค้า	สถานะสัญญา
920100	21 ก.พ. 1995	100	Mr. Frijob Capra	Terminated
901001	01 พ.ค. 1996	101	Mr. Alexandre Dumas	Terminated

จำนวนสัญญาต่อหน่วยอาศัย 2

รูปที่ ข.14 รายงานประวัติการทำสัญญา

เมื่อเสร็จสิ้นงานทั้งหมดแล้ว สามารถกลับสู่รายการหลักได้โดยการปิดหน้าต่างรายงานที่เครื่องหมายกากบาทที่มุมหน้าต่าง และสามารถปิดหน้าต่างระบบย่อยโดยกดปุ่ม “กลับรายการก่อนหน้า” และปุ่ม “กลับรายการหลัก” ที่มุมขวาล่างของหน้าต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.7 ระบบย่อยการชำระเงินประจำวัน

ระบบย่อยนี้ออกแบบมาเพื่อให้ข้อมูลการชำระเงินของแต่ละเขตควบคุม โดยสามารถเข้าสู่ระบบย่อยนี้ได้โดยการคลิกปุ่ม “การชำระเงินประจำวัน” ซึ่งจะแสดงหน้าต่างดังรูป ข.15

การชำระเงินประจำวัน

วันที่: 0 มีนาคม 1999 เฉพาะค่าเช่า

เขตควบคุม: รหัสเขต 1 ชื่อเขต bangkae region

เลขที่ชำระเงิน	วันที่จ่าย	รหัสสัญญา	รหัสหน่วยอาศัย	รายการชำระเงิน	จำนวน	งวดที่จ่าย
1	01 มี.ค. 1999	1001000	100001	rent	1000	มี.ค. 1999
1	01 มี.ค. 1999	1001000	100001	insurance	21	มี.ค. 1999
1	01 มี.ค. 1999	1001000	100001	water	10	มี.ค. 1999
121	01 มี.ค. 1999	100202	100002	rent	1000	มี.ค. 1999
121	01 มี.ค. 1999	100202	100002	insurance	100	มี.ค. 1999

ยอดรวม: 3431

พิมพ์ทุกเขตควบคุม

รูปที่ ข.15 หน้าต่างระบบย่อยการชำระเงินประจำวัน

ด้วยแนวคิดการออกแบบคือต้องการให้แสดงการชำระเงินของแต่ละเขตควบคุมดังนั้น สิ่งที่ใช้ใช้ต้องกระทำก่อน ก็คือเลือกเขตควบคุมที่ต้องการจะดูข้อมูล โดยคลิกปุ่ม “ก่อนหน้า” และ “ต่อไป” เพื่อเลือกเขตควบคุมที่ต้องการดูข้อมูล เมื่อได้เขตควบคุมที่ต้องการแล้ว ก็จะต้องเลือกวันที่ที่ต้องการดูรายการชำระเงิน โดยสามารถเลือกวันที่ได้ที่มุมซ้ายบน โดยจะใส่วันที่ในช่องแรก และเลือกเดือนจากรายการเดือนในช่องต่อมา และพิมพ์ปีที่ต้องการในช่องสุดท้าย สำหรับวันที่ที่ต้องการดูข้อมูลนั้นถ้าใส่เป็นเลขศูนย์ ก็จะเป็นการบอกระบบว่าต้องการดูข้อมูลการชำระเงินของทั้งเดือน แต่ถ้าระบุวันที่ลงไปเป็นวันอื่น ก็จะเป็นการขอข้อมูลเฉพาะวันเดียวเท่านั้น

จากนั้นก็สามารเลือกได้ว่าต้องการจะดูเฉพาะข้อมูลการชำระเงินค่าเช่าหน่วยพักอาศัยหรือไม่ เพราะการชำระเงินนั้นสามารถชำระได้หลายอย่าง เช่นค่าเช่า ค่าประกัน ค่าไฟ ค่าน้ำ และค่าใช้จ่ายอื่น ๆ ที่อาจเพิ่มเติมในภายหลัง ถ้าต้องการเลือกดูเฉพาะค่าเช่า ให้เลือกที่เช็คบ็อกซ์เอกสารเป็นเอกสารที่ส่งวงเวียนไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“เฉพาะค่าเช่า” หลังจากนั้นก็ให้กดปุ่ม “แสดงข้อมูล” ซึ่งระบบก็จะแสดงข้อมูลในส่วนล่างของหน้าต่างนั่นเอง

ซึ่งจากรูปจะเห็นว่าข้อมูลที่แสดงนั้นจะประกอบด้วย เลขที่การชำระเงิน วันที่ชำระเงิน รหัสสัญญาที่ทำการชำระเงินนั้น และรหัสหน่วยพักอาศัยที่เป็นการชำระเงินนั้น ต่อมาก็บอกว่ารายการชำระเงินนั้นเป็นค่าใช้จ่ายเป็นค่าอะไร ซึ่งถ้าเป็นกรณีเลือกเฉพาะค่าเช่าเท่านั้น ส่วนนี้ก็จะมีการเฉพาะค่าเช่าปรากฏขึ้นมา

ส่วนต่อมาก็บอกว่าจำนวนเงินที่จ่ายนั้นเป็นจำนวนเงินเท่าใด และส่วนสุดท้ายจะบอกว่าการชำระเงินนั้น เป็นการจ่ายงวดเดือนที่เท่าใด ที่ด้านล่างของหน้าต่างจะรวมยอดเงินให้ด้วยการชำระเงินทั้งหมดที่ปรากฏในหน้าต่างนั้นรวมเป็นตัวเงินเท่าใด

ผู้ใช้สามารถเลือกได้ว่าต้องการจะให้แสดงข้อมูลที่อยู่ในหน้าต่างนั้นออกเป็นรายงานเพื่อส่งออกเครื่องพิมพ์หรือไม่ ถ้าต้องการก็ให้กดปุ่ม “พิมพ์รายงาน” และสามารถเลือกได้อีกว่าจะให้พิมพ์รายงานการชำระเงินของทุกเขตควบคุมในรายงานเดียวกันหรือไม่ ถ้าไม่เลือกก็จะเป็นการสร้างรายงานเฉพาะเขตควบคุมเดียวเท่านั้น โดยสามารถเลือกได้ที่เช็คบ็อกซ์ “พิมพ์ทุกเขตควบคุม” ซึ่งรายงานที่ระบบสร้างให้ก็จะมีหน้าตา ดังแสดงในรูปที่ ข.16

The screenshot shows a software window titled 'dailyrep: Previewer' displaying a report for 'bangkai region'. The report is titled 'การเคหะแห่งชาติ รายงานการชำระเงิน' and is dated '30 มี.ค. 2000 / 02:46'. It lists two payment periods: one for '01 มี.ค. 1999' and another for '02 มี.ค. 1999'. Each period includes a table of payment details with columns for 'รหัสการชำระเงิน' (Payment Code), 'งวดที่จ่าย' (Payment Period), 'จำนวน' (Amount), 'ชนิดการชำระ' (Payment Type), 'รหัสหน่วยอาศัย' (Residence Code), and 'เลขที่สัญญา' (Contract Number). The total amount for the first period is 2131, and for the second period, it is 1300. The overall total for the region is 3431.

รหัสการชำระเงิน	งวดที่จ่าย	จำนวน	ชนิดการชำระ	รหัสหน่วยอาศัย	เลขที่สัญญา
1	01 มี.ค. 1999	1000	rent	100001	1001000
1	01 มี.ค. 1999	21	insurance	100001	1001000
1	01 มี.ค. 1999	10	water	100001	1001000
121	01 มี.ค. 1999	1000	rent	100002	100202
121	01 มี.ค. 1999	100	insurance	100002	100202
ยอดชำระของวันทั้งหมด:		2131			
Paymentdate 02 มี.ค. 1999					
รหัสการชำระเงิน	งวดที่จ่าย	จำนวน	ชนิดการชำระ	รหัสหน่วยอาศัย	เลขที่สัญญา
102	01 มี.ค. 1999	1200	rent	101001	902001
102	01 มี.ค. 1999	100	insurance	101001	902001
ยอดชำระของวันทั้งหมด:		1300			
ยอดชำระของเขต:		3431			
ยอดชำระทั้งหมด:		3431			

รูปที่ ข.16 รายงานการชำระเงิน

เมื่อเสร็จสิ้นการทำงานแล้วสามารถกลับสู่รายการหลักได้โดยการกดปุ่ม “กลับรายการหลัก” ซึ่งจะปิดหน้าต่าง และจบการทำงานของระบบย่อยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.8 ระบบย่อยหนี้เสีย

ระบบย่อยหนี้เสียเป็นระบบย่อยที่มีวัตถุประสงค์ให้เจ้าหน้าที่ที่เกี่ยวข้องสามารถตรวจสอบได้ว่าแต่ละโครงการนั้นมีลูกหนี้ค้างชำระเป็นจำนวนเท่าใด และเป็นจำนวนเงินเท่าใด ซึ่งสามารถเข้าระบบย่อยได้โดยการกดปุ่ม “หนี้เสีย” ที่หน้าต่างรายการหลัก ซึ่งจะแสดงหน้าต่างระบบย่อยหนี้เสียขึ้นมาดังรูป ข.17

WINDOW1

หนี้เสีย ตั้งแต่งวด: มกราคม 1999

เขตควบคุม รหัสเขต: 1 ชื่อเขต: bangkae region

โครงการ รหัสโครงการ: ชื่อโครงการ: Happy Village

หนี้ค้างชำระ

รหัสหน่วยอาศัย	เลขที่สัญญา	ค่าเช่าค้าง	ค่าประกันค้าง	จำนวนงวด			สถานะ	คำนำชื่อ	ชื่อ	นามสกุล	
				จ่าย	ค้าง	รวม					
100001	1001000	9000	0	4	9	50	1	Mr.	Frijob	Capra	
100003	102001	3000	60	0	3	60	1	Mrs.	Michael	Crichton	
100005	1020009	13000	400	0	14	15	1	Mr.	Frijob	Capra	
รวม		25000	460								ยอดรวมหนี้: 25460

พิมพ์รายงาน พิมพ์ทุกโครงการ กลับสู่รายการหลัก

รูปที่ ข.17 หน้าต่างระบบย่อยหนี้เสีย

ที่ระบบย่อยนี้ จะสามารถให้ผู้ใช้เลือกเขตควบคุมและโครงการที่อยู่ในความสนใจได้โดยการกดปุ่ม “ก่อนหน้า” และ “ต่อไป” โดยต้องเลือกเขตควบคุมที่ต้องการก่อน และจึงเลือกโครงการที่ต้องการดูข้อมูล จากนั้น ก็จะทำกรเลือกงวดแรกที่รับรู้หนี้เสีย คือระบบนี้จะสามารถกำหนดได้ว่าต้องการจะรับรู้หนี้เสียตั้งแต่งวดใด เช่นย้อนหลัง 1 ปี หรือ 3 เดือน เป็นต้น

โดยสามารถกระทำได้โดยเลือกงวดเริ่มต้นรับรู้หนี้เสียที่ช่องเดือน และ ปี ที่มุมขวาบนหลังข้อความ “ตั้งแต่งวด” เมื่อเสร็จแล้ว ที่ส่วนล่างของหน้าต่างก็จะแสดงรายการหน่วยอาศัยที่ค้างชำระขึ้นมา โดยจะให้ข้อมูลคือ รหัสหน่วยอาศัย เลขที่สัญญาที่กระทำ และแสดงยอดค่าเช่าค้างชำระ ค่าประกันการเช่าค้างชำระ และแสดงสถิติออกมาว่า จ่ายค่าเช่าไปเป็นจำนวนกี่งวด และค้างค่าเช่าเป็นจำนวนกี่งวด จากจำนวนงวดทั้งหมดในสัญญาเป็นจำนวนกี่งวด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนต่อมาก็จะแสดงสถานะของสัญญาว่าเป็นอย่างไร และสุดท้ายก็จะแสดงชื่อลูกค้าที่เป็นเจ้าของสัญญานั้น ด้านล่างก็จะเป็นการรวมยอดให้ว่ารวมทั้งโครงการนั้นมีค่าเช่าค้าง ค่าประกันค้าง เป็นจำนวนเท่าใด และรวมทั้งหมดเป็นจำนวนเงินเท่าใด

ผู้ใช้สามารถพิมพ์รายการออกเป็นรายงานได้โดยกดปุ่ม “พิมพ์รายงาน” และสามารถสั่งให้พิมพ์ทุกโครงการในเขตควบคุมนั้นได้โดยเลือกที่เช็คบ็อกซ์ “พิมพ์ทุกโครงการ” ซึ่งรายงานที่สร้างขึ้นมาก็คจะมีหน้าตาดังรูปที่ ข.18

baddebtrep: Previewer

พิมพ์ มุมมอง วิดีโอ

การลงทะเบียนแห่งชาติ
รายงานหนี้เสีย

วันที่ออกรายงาน: 30 มี.ค. 2000 / 02:47 หนัวย่อนถึงงวด: มกราคม 1999

รหัสโครงการ 1 ชื่อโครงการ Happy Village

เลขสัญญา	รหัสสัญญา	พื้นที่ควบคุม	จำนวนหนี้	พื้นที่	จำนวนงวด	จำนวนหนี้	สถานะสัญญา	ลูกค้า	ยอดหนี้
100001	1001000	9000	0	50	4	9	Active	Mr. Priyob Capra	9000
100002	1002002	0	0	60	12	0	Active	Mr. Alexandre Dumas	0
100003	1020001	3000	60	60	0	3	Active	Mrs. Michael Christian	3000
100005	1020005	13000	-400	15	0	14	Active	Mr. Priyob Capra	13000
101001	902001	0	0	14	13	0	Active	Mr. Victor Hugo	0
รวมยอดโครงการ									25460
รวมหนี้เงินเช่า									25460

รูปที่ ข.18 รายงานหนี้เสีย

เมื่อเสร็จกิจกรรมทั้งหมดแล้วสามารถกลับสู่รายการหลักได้โดยกดปุ่ม “กลับรายการหลัก”

ข.9 ระบบย่อยลูกค้าหมดอายุสัญญา

ระบบย่อยนี้ออกแบบมาเพื่อให้ผู้ใช้ที่เป็นเจ้าหน้าที่สามารถตรวจสอบได้ว่าที่หน่วยควบคุมของตน และที่โครงการของตน จะมีหน่วยอาศัยใดบ้างที่กำลังจะหมดอายุสัญญา เพื่อที่จะได้ดำเนินการที่เหมาะสมต่อไปกับลูกค้ารายนั้น เช่นอาจจะทำจดหมายแจ้งเตือนเพื่อให้มาต่อสัญญา หรืออาจจะจัดเก็บหนี้ค้าง หรือเชิญให้ออกจากหน่วยอาศัย ซึ่งก็แล้วแต่กรรมวิธีที่ต้องกระทำ

ระบบย่อยนี้สามารถเข้าใช้งานได้โดยการกดปุ่ม “หนีหมดสัญญา” ซึ่งระบบก็จะแสดงหน้าต่างของระบบย่อยนี้ขึ้นมาดังแสดงในรูป ข.19

จากรูป จะเห็นได้ว่าผู้ใช้สามารถเลือกที่จะดูข้อมูลของเขตควบคุม และโครงการที่ต้องการ ดังนั้นสิ่งแรกที่ต้องทำ ก็คือเลือกเขตควบคุม และโครงการที่ต้องการก่อน โดยการกดปุ่ม “ก่อนหน้า” และ “ต่อไป” จนกว่าจะได้เขตควบคุม และ โครงการที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WINDOW1

ลูกหนี้หมดอายุสัญญา

เขตควบคุม

รหัสเขต

ชื่อเขต

โครงการ

รหัสโครงการ

ชื่อโครงการ

หน่วยอาศัยหมดสัญญา

รหัสหน่วย	เลขที่สัญญา	งวดแรก	ค่าเช่า	สถานะ	สำเนา	ชื่อ	นามสกุล	ค่าเช่าค้าง	ค่าประกันค้าง
100005	1020009	01 ม.ค. 1999	1000	1		Mr. Fijob	Capra	15000	500
101001	902001	01 ก.พ. 1999	1200	1		Mr. Victor	Hugo	1200	0

พิมพ์ทุกโครงการ

รูปที่ ข.19 หน้าต่างระบบย่อยลูกหนี้หมดอายุสัญญา

เมื่อได้เขตควบคุมและโครงการที่ต้องการแล้ว ด้านล่างของหน้าต่างก็จะแสดงรายชื่อสัญญาทั้งหมด ที่จะหมดอายุในงวดเดือนถัดไปขึ้นมาให้ โดยจะมีรายละเอียดเช่น รหัสหน่วยอาศัย เลขที่สัญญา และบอกว่าสัญญานั้น ๆ เริ่มงวดแรกเมื่อใด เพื่อจะแสดงให้เห็นถึงอายุของสัญญานี้ ต่อไปก็จะบอกค่าเช่ารายเดือน บอกถึงสถานะของสัญญานั้น ๆ และบอกชื่อลูกค้าที่เป็นเจ้าของสัญญานั้น เพื่อจะสามารถติดต่อได้ถูกต้อง

และส่วนสุดท้ายก็จะบอกหนี้ค้างชำระของสัญญานั้น ซึ่งแยกออกเป็นหนี้ที่เกิดจากค่าเช่า และหนี้ที่เกิดจากค่าประกันการเช่า รวมกันออกเป็นยอดหนี้ดังรายการ ซึ่งจะเป็นประโยชน์ต่อการตามหนี้ ถ้าต้องการจะพิมพ์รายงานก็สามารถทำได้โดยกดปุ่ม “พิมพ์รายงาน” โดยระบบจะนำเอาชุดข้อมูลที่ปรากฏในหน้าต่างนี้ ไปสร้างเป็นรายงาน เพื่อจะพิมพ์ออกทางเครื่องพิมพ์ต่อไป และถ้าต้องการให้สร้างรายงานที่แสดงทุกโครงการที่อยู่ในเขตควบคุมก็สามารถทำได้ โดยเลือกที่เช็คบ็อกซ์ “พิมพ์ทุกโครงการ” แต่ถ้าไม่เลือกเช็คบ็อกซ์นี้ ระบบก็จะแสดงรายงานเฉพาะโครงการเดียวที่อยู่ในหน้าต่างนี้ เมื่อสร้างรายงานแล้วระบบก็จะแสดงรายงานออกมาให้ดังรูป ข.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

termrep: Previewer

แฟ้ม มุมมอง วิดีโอ

การทะเลาะแย่งชาติ
รายงานลูกหนี้หมดอายุสัญญา
วันที่ออกรายงาน: 30 มี.ค. 2000 / 02:48

ชื่อโครงการ Happy Village รหัสโครงการ 1

ประเภทบัญชี	บัญชีแยก	จำนวน	จำนวน	วันที่	สถานะ	วันที่ถึง	จำนวนเงิน
200005	200005	1000	02 มี.ค. 1999	Mr. Frisco Castro	Active	1500	500
101001	902001	1200	01 มี.ค. 1999	Mr. Victor Hugo	Active	1200	0

จำนวนใบหนี้รวม 2

รูปที่ ข.20 รายงานลูกหนี้หมดอายุสัญญา

เมื่อรายงานขึ้นมาดังรูป สามารถสั่งพิมพ์ออกเครื่องพิมพ์ได้โดยการกดปุ่มรูปเครื่องพิมพ์ที่มุมซ้ายบน เมื่อต้องการจบการทำงาน ก็สามารถปิดรายงานนี้ได้ด้วยการกดปุ่มกากบาทที่แถบของปุ่มเครื่องมือ หรือที่มุมขวาบนของหน้าต่างก็ได้ และถ้าต้องการกลับสู่รายการหลัก ก็สามารถทำได้โดยการกดปุ่ม “กลับรายการหลัก” ซึ่งเป็นอันจบกระบวนการทำงานของระบบย่อยนี้

ข.10 การทำงานกับรายงาน

ระบบรายงานของโอราเคิลนั้นจะเป็นมาตรฐานเดียวกันทั้งหมด และมีการทำงานที่ง่ายตรงไปตรงมา โดยควบคุมจากแถบปุ่มเครื่องมือเดียว คือ

สามารถสั่งพิมพ์รายงานออกทางเครื่องพิมพ์ได้โดยการกดปุ่มรูปเครื่องพิมพ์ สามารถย่อหรือขยาย เพื่อดูรายละเอียดของรายงานได้โดยการกดปุ่มรูปแว่นขยายอันเล็ก และอันใหญ่ เพื่อย่อและขยายตามลำดับ นอกจากนี้ถ้ารายงานมีมากกว่าหนึ่งหน้า สามารถใช้ปุ่มลูกศรทั้งสี่ปุ่มเพื่อเปลี่ยนหน้า โดยปุ่มแรก จะไปที่หน้าแรก ปุ่มต่อมาจะไปที่หน้าก่อนหน้านี้นี้ ปุ่มที่สามจะไปที่หน้าถัดไป และปุ่มสุดท้ายจะไปที่หน้าสุดท้าย

ผู้ใช้สามารถออกจากรายงานได้โดยการกดปุ่มรูปกากบาทสีแดงที่แถบเครื่องมือ หรือกดปุ่มกากบาทสีดำที่มุมขวาบนของหน้าต่างได้

ทั้งหมดนี้ก็จะเป็นการทำงานของโปรแกรมระบบบัญชีลูกหนี้เข้าทั้งหมด ซึ่งสามารถใช้งานได้ง่ายด้วยระบบควบคุมการทำงานมาตรฐานเดียวกันหมดทุกหน้าต่าง และรายงานทุกรายงานก็สามารถใช้งานได้ด้วยมาตรฐานเดียวกันทั้งหมด

ภาคผนวก ค.

โปรแกรมต้นฉบับ

โปรแกรมต้นฉบับ จะแบ่งเป็นสองส่วน โดยเลือกเฉพาะส่วนที่ตีพิมพ์ได้ คือส่วนแรก จะเป็นคำสั่งเอสคิวแอล สำหรับสร้างโครงสร้างของตารางของฐานข้อมูล และอีกส่วนคือ โปรแกรมต้นฉบับเป็นภาษา พีแอล / เอสคิวแอล ของโอราเคิล โดยจะเป็นส่วนที่ทำหน้าที่เป็น สโตรโปรซีเจอร์ ที่อยู่ทางด้านฝั่งเซิร์ฟเวอร์

ค.1 สกริปต์สร้างตาราง

```
CREATE TABLE pok.Customer (  
  customer_ID NUMBER ,  
  title VARCHAR(20),  
  firstname VARCHAR(20),  
  lastname VARCHAR(20),  
  regisAddr VARCHAR(20),  
  currentAddr VARCHAR(20),  
  telephone VARCHAR(20),  
  PRIMARY KEY (Customer_ID));
```

```
CREATE TABLE pok.account (  
  account_ID NUMBER ,  
  accountName VARCHAR(20),  
  accountDesc VARCHAR(20),  
  PRIMARY KEY (account_ID));
```

```
CREATE TABLE pok.BuildingType (  
  buildingType_ID NUMBER ,  
  buildingTypeName VARCHAR(20),
```

PRIMARY KEY (BuildingType_ID));

```
CREATE TABLE pok.component (
  component_ID NUMBER ,
  componentName VARCHAR(20),
  PRIMARY KEY (component_ID));
```

```
CREATE TABLE pok.Location (
  Location_ID NUMBER ,
  LocationName VARCHAR(20),
  Road VARCHAR(20),
  moo NUMBER,
  soi VARCHAR(20),
  district VARCHAR(20),
  city VARCHAR(20),
  province VARCHAR(20),
  postalcode VARCHAR(20),
  PRIMARY KEY (Location_ID));
```

```
CREATE TABLE pok.Model (
  Model_ID NUMBER ,
  ModelName VARCHAR(20),
  buildingtype_ID NUMBER REFERENCES pok.BuildingType(buildingtype_ID),
  PRIMARY KEY (Model_ID));
```

```
CREATE TABLE pok.ComponentItem (
  componentQuantity NUMBER,
  component_ID NUMBER REFERENCES pok.component(component_ID),
  model_ID NUMBER REFERENCES pok.Model(model_id),
  PRIMARY KEY (Component_ID,model_ID));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CREATE TABLE pok.Project (
    Project_ID NUMBER ,
    ProjectName VARCHAR(20),
    location_ID NUMBER REFERENCES pok.Location(location_ID),
    PRIMARY KEY (Project_ID));
```

```
CREATE TABLE pok.Quotation (
    unitamount NUMBER,
    space FLOAT(20),
    cost FLOAT(20),
    stdinterest FLOAT(20),
    model_ID NUMBER REFERENCES pok.Model(model_ID),
    project_ID NUMBER REFERENCES pok.Project(project_ID),
    PRIMARY KEY (model_ID,project_ID));
```

```
CREATE TABLE pok.UNIT (
    Unit_ID VARCHAR(20) ,
    debtorcode VARCHAR(20),
    floorno NUMBER,
    buildingno NUMBER,
    model_ID NUMBER REFERENCES pok.Model(model_ID),
    project_ID NUMBER REFERENCES pok.Project(project_ID),
    PRIMARY KEY (UNIT_ID));
```

```
CREATE TABLE pok.Contract (
    contract_ID NUMBER ,
    contractDate DATE,
    contractstatus NUMBER,
```

```
    paymenttype VARCHAR(20),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

priceperterm NUMBER,
dateFirstTerm DATE,
term NUMBER,
unit_ID VARCHAR(20) REFERENCES pok.UNIT(unit_ID),
customer_ID NUMBER REFERENCES pok.Customer(customer_ID),
PRIMARY KEY (Contract_ID));

```

```

CREATE TABLE pok.RentContract (

```

```

insuranceTerm NUMBER,
insurancePrice NUMBER,
contract_ID NUMBER REFERENCES pok.Contract(contract_ID),
PRIMARY KEY (Contract_ID));

```

```

CREATE TABLE pok.PaymentTX (

```

```

paymentTX_ID NUMBER,
paymentDate DATE,
paymentTerm DATE,
contract_ID NUMBER REFERENCES pok.Contract(contract_ID),
PRIMARY KEY (PaymentTX_ID));

```

```

CREATE TABLE pok.PaymentTXDetail (

```

```

paymentPrice NUMBER,
paymentTX_ID NUMBER REFERENCES pok.PaymentTX(paymentTX_ID),
account_ID NUMBER REFERENCES pok.account(account_ID),
PRIMARY KEY (paymenttx_ID,account_ID));

```

ค.2 โปรแกรมต้นฉบับของสโตร์โพรซีเจอร์

```
PACKAGE accountp IS
```

```
function getname(id account.account_id%type) return account.accountname%type;
```

```
END;
```

```
PACKAGE BODY accountp IS
```

```
function getname(id account.account_id%type) return account.accountname%type is
```

```
temp account.accountname%type;
```

```
begin
```

```
select accountname
```

```
into temp
```

```
from account
```

```
where account_id = id;
```

```
return temp;
```

```
exception
```

```
when others then
```

```
return " ";
```

```
end;
```

```
END;
```

```
PACKAGE buildingp IS
```

```
type buildingrec is record (buildingtype_id buildingtype.buildingtype_id%type,
```

```
buildingtypename buildingtype.buildingtypename%type);
```

```
type buildingcur is ref cursor return buildingrec;
```

```
procedure query(resultset in out buildingcur);
```

```
function getname(id buildingtype.buildingtype_id%type) return
```

```
buildingtype.buildingtypename%type;
```

```
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PACKAGE BODY buildingp IS

procedure query(resultset in out buildingcur) is

```
begin
    open resultset for
    select *
    from buildingtype;
end;
```

function getname(id buildingtype.buildingtype_id%type) return

buildingtype.buildingtypename%type is

temp buildingtype.buildingtypename%type;

begin

```
select buildingtypename
into temp
from buildingtype
where buildingtype_id = id;
return temp;
```

exception

when others then

return 0;

end;

END;

PACKAGE componentitem IS

```
type comprec is record (model_id componentitem.model_id%type,
    component_id componentitem.component_id%type,
    componentquantity componentitem.componentquantity%type,
    componentname component.componentname%type);
```

type comptab is table of comprec

index by binary_integer;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure getdetail(resultset in out comptab, id componentitem.model_id%type);
END;

```

PACKAGE BODY componentitem IS

```

procedure getdetail(resultset in out comptab, id componentitem.model_id%type) is

```

```

counter integer;

```

```

temp number;

```

```

cursor compcur is

```

```

select *

```

```

from componentitem

```

```

where model_id = id;

```

```

currow componentitem%rowtype;

```

```

begin

```

```

open compcur;

```

```

counter := 0;

```

```

temp := 0;

```

```

loop

```

```

    fetch compcur into currow;

```

```

    exit when compcur%notfound or

```

```

    compcur%notfound is NULL;

```

```

    resultset(counter).model_id := currow.model_id;

```

```

    resultset(counter).component_id := currow.component_id;

```

```

    resultset(counter).componentquantity := currow.componentquantity;

```

```

    resultset(counter).componentname := componenttp.getname(currow.component_id);

```

```

    counter := counter + 1;

```

```

end loop;

```

```

end;

```

```

END;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PACKAGE componentp IS

```
function getname(id component.component_id%type)return component.componentname%type;
END;
```

PACKAGE BODY componentp IS

```
function getname(id component.component_id%type)return component.componentname%type is
temp component.componentname%type;
begin
select componentname
into temp
from component
where component_id = id;
return temp;
exception
when others then
return 0;
end;
END;
```

PACKAGE contractp IS

```
type contdetail is record (contract_id contract.contract_id%type,
customer_id contract.customer_id%type,
contractdate contract.contractdate%type,
contractstatus contract.contractstatus%type,
paymenttype contract.paymenttype%type,
priceperterm contract.priceperterm%type,
datefirstterm contract.datefirstterm%type,
term contract.term%type,
lastterm date,
insuranceterm rentcontract.insuranceterm%type,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

insuranceprice rentcontract.insuranceprice%type,
unit_id contract.unit_id%type,
title customer.title%type,
firstname customer.firstname%type,
lastname customer.lastname%type);

```

type contdetailtab is table of contdetail

index by binary_integer;

```

type baddebtrec is record (unit_id contract.unit_id%type,
    contract_id contract.contract_id%type,
    rentdebt number,
    insdebt number,
    totalterm number,
    paidterm number,
    badrentterm number,
    status contract.contractstatus%type,
    title customer.title%type,
    firstname customer.firstname%type,
    lastname customer.lastname%type);

```

type baddebttab is table of baddebtrec

index by binary_integer;

```

type endrec is record (unit_id contract.unit_id%type,
    contract_id contract.contract_id%type,
    datefirstterm contract.datefirstterm%type,
    datelastterm date,
    priceperterm contract.priceperterm%type,
    status contract.contractstatus%type,
    title customer.title%type,
    firstname customer.firstname%type,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lastname customer.lastname%type,
        rentdebt number,
        insdebt number);

```

```

type endtab is table of endrec

```

```

index by binary_integer;

```

```

procedure endquery(resultset in out endtab, id project.project_id%type,now date);

```

```

procedure querybaddebt(resultset in out baddebttab,id project.project_id%type,startmonth
number,startyear number,now date);

```

```

procedure querydetail(resultset in out contdetailtab,id contract.customer_id%type);

```

```

procedure querydetail2(resultset in out contdetailtab,id contract.unit_id%type);

```

```

function status(unitid number) return number;

```

```

function amount(unitid number) return number;

```

```

function getcusid(unitid number) return number;

```

```

function getmethod(unitid number) return varchar2;

```

```

function getunitid(contractid number) return number;

```

```

function isinloc(cusid customer.customer_id%type,locid location.location_id%type)return
integer;

```

```

END;

```

```

PACKAGE BODY contractp IS

```

```

function amount(unitid number) return number is

```

```

temp number;

```

```

begin

```

```

    select priceperterm

```

```

    into temp

```

```

    from contract

```

```

    where unit_id = unitid and

```

```

    contractstatus = 1 and

```

```

    contractdate = (select max(contractdate)

```

```

    from contract

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        where unit_id = unitid);
    return temp;
exception
    when others then
        return 0;
end;

function isinloc (cusid customer.customer_id%type,locid location.location_id%type) return
integer is
temp number;
cursor cont is
    select *
    from contract
    where contract.customer_id = cusid;
controw contract%rowtype;
begin
    temp :=0;
    open cont;
    loop
        fetch cont into controw;
        exit when cont%NOTFOUND or
            cont%NOTFOUND is NULL;
        if (projectp.getlocationid(unitp.getprojectid(controw.unit_id))=locid) then
            temp := temp+1;
        end if;
    end loop;
    return temp;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

function status (unitid number) return number is

temp number;

begin

temp := 0;

select contractstatus

into temp

from contract

where unitid=unit_id

and contractdate = (select max(contractdate)

from contract

where unit_id = unitid);

return temp;

exception

when others then

return 0;

end;

function getcusid(unitid number) return number is

temp number;

begin

select customer_id

into temp

from contract

where unit_id = unitid and

contractstatus = 1

and contractdate = (select max(contractdate)

from contract

where unit_id = unitid);

return temp;

exception

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
when others then
```

```
    return 0;
```

```
end;
```

```
function getmethod(unitid number) return varchar2 is
```

```
temp varchar2(20);
```

```
begin
```

```
    select paymenttype
```

```
    into temp
```

```
    from contract
```

```
    where unit_id = unitid and
```

```
          contractstatus = 1
```

```
          and contractdate = (select max(contractdate)
```

```
                                from contract
```

```
                                where unit_id = unitid);
```

```
    return temp;
```

```
exception
```

```
    when others then
```

```
        return 0;
```

```
end;
```

```
procedure querydetail(resultset in out contdetailtab,id contract.customer_id%type)is
```

```
counter integer;
```

```
cursor cont is
```

```
    select *
```

```
    from contract;
```

```
cursor cont1 is
```

```
    select *
```

```
    from contract
```

```
    where customer_id = id;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

controw contract%rowtype;
begin
  if (id = 0) or (id is NULL) then
    open cont;
    counter :=0;
    loop
      fetch cont into controw;
      exit when cont%notfound or
      cont%notfound is NULL;
      resultset(counter).contract_id := controw.contract_id;
      resultset(counter).customer_id := controw.customer_id;
      resultset(counter).contractdate := controw.contractdate;
      resultset(counter).contractstatus := controw.contractstatus;
      resultset(counter).paymenttype := controw.paymenttype;
      resultset(counter).priceperterm := controw.priceperterm;
      resultset(counter).datefirstterm := controw.datefirstterm;
      resultset(counter).term := controw.term;
      resultset(counter).lastterm :=
add_months(add_months(controw.datefirstterm,controw.term),-1);
      resultset(counter).insuranceterm := rentp.getterm(controw.contract_id);
      resultset(counter).insuranceprice := rentp.getprice(controw.contract_id);
      resultset(counter).unit_id := controw.unit_id;
      resultset(counter).title := customerp.gettitle(controw.customer_id);
      resultset(counter).firstname := customerp.getname(controw.customer_id);
      resultset(counter).lastname := customerp.getlastname(controw.customer_id);
      counter := counter +1;
    end loop;
  else
    open cont1;
    counter :=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

loop

```

fetch cont1 into controw;
exit when cont1%notfound or
        cont1%notfound is NULL;
resultset(counter).contract_id := controw.contract_id;
resultset(counter).customer_id := controw.customer_id;
resultset(counter).contractdate := controw.contractdate;
resultset(counter).contractstatus := controw.contractstatus;
resultset(counter).paymenttype := controw.paymenttype;
resultset(counter).priceperterm := controw.priceperterm;
resultset(counter).datefirstterm := controw.datefirstterm;
resultset(counter).term := controw.term;
resultset(counter).lastterm :=
add_months(add_months(controw.datefirstterm,controw.term),-1);
resultset(counter).insuranceterm := rentp.getterm(controw.contract_id);
resultset(counter).insuranceprice := rentp.getprice(controw.contract_id);
resultset(counter).unit_id := controw.unit_id;
resultset(counter).title := customerp.gettitle(controw.customer_id);
resultset(counter).firstname := customerp.getname(controw.customer_id);
resultset(counter).lastname := customerp.getlastname(controw.customer_id);
counter := counter + 1;

```

end loop;

end if;

if (counter = 0) then

```
resultset(0).title :=";
```

end if;

end;

```

procedure querydetail2(resultset in out contdetailtab,id contract.unit_id%type)is
counter integer;
cursor cont is
    select *
    from contract;
cursor cont1 is
    select *
    from contract
    where unit_id = id;
controw contract%rowtype;
begin
if (id = 0) or (id is NULL) then
    open cont;
    counter :=0;
    loop
        fetch cont into controw;
        exit when cont%notfound or
            cont%notfound is NULL;
        resultset(counter).contract_id := controw.contract_id;
        resultset(counter).customer_id := controw.customer_id;
        resultset(counter).contractdate := controw.contractdate;
        resultset(counter).contractstatus := controw.contractstatus;
        resultset(counter).paymenttype := controw.paymenttype;
        resultset(counter).priceperterm := controw.priceperterm;
        resultset(counter).datefirstterm := controw.datefirstterm;
        resultset(counter).term := controw.term;
        resultset(counter).lastterm :=
add_months(add_months(controw.datefirstterm,controw.term),-1);
        resultset(counter).insuranceterm := rentp.getterm(controw.contract_id);
        resultset(counter).insuranceprice := rentp.getprice(controw.contract_id);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

resultset(counter).unit_id := controw.unit_id;
resultset(counter).title := customerp.gettitle(controw.customer_id);
resultset(counter).firstname := customerp.getname(controw.customer_id);
resultset(counter).lastname := customerp.getlastname(controw.customer_id);
counter := counter +1;
end loop;
else
open cont1;
counter :=0;
loop
    fetch cont1 into controw;
    exit when cont1%notfound or
        cont1%notfound is NULL;
    resultset(counter).contract_id := controw.contract_id;
    resultset(counter).customer_id := controw.customer_id;
    resultset(counter).contractdate := controw.contractdate;
    resultset(counter).contractstatus := controw.contractstatus;
    resultset(counter).paymenttype := controw.paymenttype;
    resultset(counter).priceperterm := controw.priceperterm;
    resultset(counter).datefirstterm := controw.datefirstterm;
    resultset(counter).term := controw.term;
    resultset(counter).lastterm :=
add_months(add_months(controw.datefirstterm,controw.term),-1);
    resultset(counter).insuranceterm := rentp.getterm(controw.contract_id);
    resultset(counter).insuranceprice := rentp.getprice(controw.contract_id);
    resultset(counter).unit_id := controw.unit_id;
    resultset(counter).title := customerp.gettitle(controw.customer_id);
    resultset(counter).firstname := customerp.getname(controw.customer_id);
    resultset(counter).lastname := customerp.getlastname(controw.customer_id);
    counter := counter +1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end loop;
    end if;
    if (counter = 0) then
        resultset(0).title :=";
    end if;
end;

```

function getunitid(contractid number) return number is

```

temp number;
begin
    select unit_id
    into temp
    from contract
    where contract_id = contractid;
    return temp;
exception
    when others then
        return 0;
end;

```

procedure querybaddebt(resultset in out baddebttab,id project.projct_id%type,startmonth
number,startyear number,now date)is

```

counter integer;
cursor cont is
    select *
    from contract
    where contractstatus = 1 or contractstatus = 3;
lastrent date;
lastins date;
curterm date;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

thesdate date;
finalins date;
finalrent date;
ubound date;
lbound date;
insterm number;
insprice number;
tempcurterm varchar(20);
controw contract%rowtype;
tempid number;
begin
  open cont;
  curterm := now;
  tempcurterm := to_char(curterm,'mm yyyy');
  curterm := to_date(tempcurterm,'mm yyyy');
  curterm := add_months(curterm,-1);
  counter :=0;
  thesdate := to_date(to_char(startmonth)||' '||to_char(startyear),'mm yyyy');
  -- boundary check for current term and theshold term
  if (months_between(thesdate,curterm) >0)then
    thesdate := curterm;
  end-if;
  loop
    fetch cont into controw;
    exit when cont%notfound or
      cont%notfound is NULL;
    tempid := unitp.getprojectid(controw.unit_id);
    lastrent := paymentp.lastpaidrent(controw.contract_id);
    lastins := paymentp.lastpaidins(controw.contract_id);
    if (lastrent is NULL)then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lastrent := add_months(controw.datefirstterm,-1);
    end if;
    if (lastins is NULL)then
        lastins := add_months(controw.datefirstterm,-1);
    end if;
    insterm := rentp.getterm(controw.contract_id);
    insprice := rentp.getprice(controw.contract_id);
    finalrent := add_months(controw.datefirstterm,controw.term-1);
    finalins := add_months(controw.datefirstterm,insterm-1);
    if (tempid = id) and(
        ((curterm - lastrent > 0) and (finalrent-lastrent >0)) or -- condition for bad rent
        ((curterm - lastins >0) and(finalins - lastins >0))) then -- condition for bad

```

insurance

```

        resultset(counter).contract_id := controw.contract_id;
        resultset(counter).unit_id := controw.unit_id;
        resultset(counter).status := controw.contractstatus;
        resultset(counter).title := customerp.gettitle(controw.customer_id);
        resultset(counter).firstname := customerp.getname(controw.customer_id);
        resultset(counter).lastname := customerp.getlastname(controw.customer_id);
        resultset(counter).totalterm := controw.term;
        resultset(counter).paidterm := floor(months_between(lastrent,

```

controw.datefirstterm)+1);

```

        ubound := least(finalrent,curterm);

```

```

        lbound := greatest(thesdate,lastrent);

```

```

        resultset(counter).rentdebt :=

```

floor(months_between(ubound,lbound))*controw.priceperterm;

```

        resultset(counter).badrentterm := floor(months_between(ubound,lastrent));

```

```

        ubound := least(finalins,curterm);

```

```

        lbound := greatest(thesdate,lastins);

```

```

        resultset(counter).insdebt := floor(months_between(ubound,lbound))*insprice;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (resultset(counter).rentdebt <0) and (months_between(thesdate,finalrent)>0)
then
    resultset(counter).rentdebt :=0;
end if;
if (resultset(counter).insdebt <0) and (months_between(thesdate,finalins)>0)
then
    resultset(counter).insdebt :=0;
end if;
counter := counter +1;
end if;
end loop;
if (counter = 0) then
    resultset(0).title :="";
end if;
end;

procedure endquery(resultset in out endtab, id project.project_id%type,now date) is
counter integer;
cursor cont is
select *
    from contract
    where contractstatus = 1 or contractstatus = 3;

lastrent date;
lastins date;
curterm date;
finalins date;
finalrent date;
ubound date;
lbound date;
insterm number;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

insprice number;
controw contract%rowtype;
tempcurterm varchar(20);
tempid number;
begin
    open cont;
    curterm := now;
    tempcurterm := to_char(curterm,'mm yyyy');
    curterm := to_date(tempcurterm,'mm yyyy');
    curterm := add_months(curterm,-1);
    counter :=0;
    -- boundary check for current term and theshold term
loop
    fetch cont into controw;
    exit when cont%notfound or
        cont%notfound is NULL;
    tempid := unitp.getprojectid(controw.unit_id);
    lastrent := paymentp.lastpaidrent(controw.contract_id);
    lastins := paymentp.lastpaidins(controw.contract_id);

    if (lastrent is NULL)then
        lastrent := add_months(controw.datefirstterm,-1);
    end if;

    if (lastins is NULL)then
        lastins := add_months(controw.datefirstterm,-1);
    end if;

    insterm := rentp.getterm(controw.contract_id);
    insprice := rentp.getprice(controw.contract_id);
    finalrent := add_months(controw.datefirstterm,controw.term-1);
    finalins := add_months(controw.datefirstterm,insterm-1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (tempid = id) and (months_between(curterm,finalrent) >= -1) then
    resultset(counter).contract_id := controw.contract_id;
    resultset(counter).unit_id := controw.unit_id;
    resultset(counter).status := controw.contractstatus;
    resultset(counter).title := customerp.gettitle(controw.customer_id);
    resultset(counter).firstname := customerp.getname(controw.customer_id);
    resultset(counter).lastname := customerp.getlastname(controw.customer_id);
    resultset(counter).datefirstterm := controw.datefirstterm;
    resultset(counter).datelastterm := finalrent;
    resultset(counter).priceperterm := controw.priceperterm;
    ubound := finalrent;
    lbound := lastrent;
    resultset(counter).rentdebt :=
floor(months_between(ubound,lbound))*controw.priceperterm;
    ubound := finalins;
    lbound := lastins;
    resultset(counter).insdebt := floor(months_between(ubound,lbound))*insprice;
    counter := counter +1;
end if;
end loop;
if (counter = 0) then
    resultset(0).title := "";
end if;
end;
END;

```

```
PACKAGE customerp IS
```

```

type cusrec is record (customer_id customer.customer_id%type,
    title customer.title%type,
    firstname customer.firstname%type,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lastname customer.lastname%type,
        regisaddr customer.regisaddr%type,
        currentaddr customer.currentaddr%type,
        telephone customer.telephone%type);

type cuscur is ref cursor return cusrec;

type custab is table of cusrec
    index by binary_integer;

procedure query(resultset in out cuscur);

procedure querybylocation(resultset in out custab, id location.location_id%type);

    function gettitle(cusid number) return varchar2;
    function getname(cusid number) return varchar2;
    function getlastname(cusid number) return varchar2;

END;

PACKAGE BODY customerp IS
    function gettitle(cusid number) return varchar2 is
        temp varchar2(20);
    begin
        select title
        into temp
        from customer
        where customer_id = cusid;
        return temp;
    exception
        when others then
            return 0;
    end;

    function getname(cusid number) return varchar2 is
        temp varchar2(20);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    select firstname
    into temp
    from customer
    where customer_id = cusid;
    return temp;
exception
    when others then
        return 0;
end;

```

```

function getlastname(cusid number) return varchar2 is
temp varchar2(20);
begin
    select lastname
    into temp
    from customer
    where customer_id = cusid;
    return temp;
exception
    when others then
        return 0;
end;

```

```

procedure query(resultset in out cuscur) is
begin
    open resultset for
    select *
    from customer;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure querybylocation(resultset in out custab, id location.location_id%type) is
counter integer;
cursor temp is
select *
from customer;

temprow customer%rowtype;
begin
    open temp;
    counter := 0;
    loop
        fetch temp into temprow;
        exit when temp%notfound or
temp%notfound is NULL;
        if (contractp.isinloc(temprow.customer_id,id)> 0) then
            resultset(counter).customer_id := temprow.customer_id;
            resultset(counter).title := temprow.title;
            resultset(counter).firstname := temprow.firstname;
            resultset(counter).lastname := temprow.lastname;
            counter :=counter+1;
        end if;
    end loop;
    if (counter = 0) then
        resultset(0).title :="";
    end if;
end;
END;

```

PACKAGE locationp IS

```

type locationrec is record (location_id location.location_id%type,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

locationname location.locationname%type,
road    location.road%type,
moo     location.moo%type,
soi     location soi%type,
district location.district%type,
city    location.city%type,
province location.province%type,
postalcode location.postalcode%type);

```

```

type locationcur is ref cursor return locationrec;

```

```

procedure query(resultset in out locationcur);

```

```

END;

```

```

PACKAGE BODY locationp IS

```

```

procedure query(resultset in out locationcur) is

```

```

begin

```

```

open resultset for

```

```

select location_id, locationname,road, moo,soi,district, city,province,postalcode

```

```

from location;

```

```

end;

```

```

END;

```

```

PACKAGE modelp IS

```

```

function gettype(id model.model_id%type) return number;

```

```

function getname(id model.model_id%type) return model.modelname%type;

```

```

END;

```

```

PACKAGE BODY modelp IS

```

```

function gettype(id model.model_id%type) return number is

```

```

temp number;

```

```

begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

select buildingtype_id
into temp
from model
where model_id = id;
return temp;

exception
when others then
return 0;

end;

function getname(id model.model_id%type) return model.modelname%type is
temp model.modelname%type;
begin
select modelname
into temp
from model
where model_id = id;
return temp;

exception
when others then
return 0;

end;
END;

```

PACKAGE paymentp IS

```

function getdate(id paymenttx.paymenttx_id%type) return paymenttx.paymentdate%type;
function getcontractid(id paymenttx.paymenttx_id%type) return
paymenttx.contract_id%type;
function getterm(id paymenttx.paymenttx_id%type) return
paymenttx.paymentterm%type;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function lastpaidrent(id contract.contract_id%type)return date;
```

```
function lastpaidins(id contract.contract_id%type)return date;
```

```
END;
```

```
PACKAGE BODY paymentp IS
```

```
function getdate(id paymenttx.paymenttx_id%type)return paymenttx.paymentdate%type is
```

```
temp date;
```

```
begin
```

```
    select paymentdate
```

```
    into temp
```

```
    from paymenttx
```

```
    where paymenttx_id = id;
```

```
    return temp;
```

```
exception
```

```
    when others then
```

```
        return to_date('00 00 0000','dd mm yyyy');
```

```
end;
```

```
function getcontractid(id paymenttx.paymenttx_id%type)return paymenttx.contract_id%type is
```

```
temp number;
```

```
begin
```

```
    select contract_id
```

```
    into temp
```

```
    from paymenttx
```

```
    where paymenttx_id = id;
```

```
    return temp;
```

```
exception
```

```
    when others then
```

```
        return 0;
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function getterm(id paymenttx.paymenttx_id%type)return paymenttx.paymentterm%type is
temp date;
begin
    select paymentterm
    into temp
    from paymenttx
    where paymenttx_id = id;
    return temp;
exception
    when others then
        return to_date('00 0000','mm yyyy');
end;

function lastpaidrent(id contract.contract_id%type)return date is
temp date;
cursor hunt is
    select *
    from paymenttx
    where contract_id = id
    order by paymentterm desc;
huntrow paymenttx%rowtype;
begin
    open hunt;
    loop
        fetch hunt into huntrow;
        exit when hunt%notfound or
            hunt%notfound is NULL;
        if (paymentdetailp.isrent(huntrow.paymenttx_id)=1)then
            temp := huntrow.paymentterm;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
exit;
```

```
end if;
```

```
end loop;
```

```
return temp;
```

```
exception
```

```
when others then
```

```
return to_date('00 0000','mm yyyy');
```

```
end;
```

```
function lastpaidins(id contract.contract_id%type) return date is
```

```
temp date;
```

```
cursor hunt is
```

```
select *
```

```
from paymenttx
```

```
where contract_id = id
```

```
order by paymentterm desc;
```

```
huntrow paymenttx%rowtype;
```

```
begin
```

```
open hunt;
```

```
loop
```

```
fetch hunt into huntrow;
```

```
exit when hunt%notfound or
```

```
hunt%notfound is NULL;
```

```
if (paymentdetailp.isins(huntrow.paymenttx_id)=1) then
```

```
temp := huntrow.paymentterm;
```

```
exit;
```

```
end if;
```

```
end loop;
```

```
return temp;
```

```
exception
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

when others then

return to_date('00 0000','mm yyyy');

end;

END;

PACKAGE paymentdetailp IS

type dailyrec is record (paymenttx_id paymenttx.paymenttx_id%type,
 paymentdate paymenttx.paymentdate%type,
 location_id location.location_id%type,
 contract_id paymenttx.contract_id%type,
 unit_id unit.unit_id%type,
 accountname account.accountname%type,
 amount paymenttxdetail.paymentprice%type,
 term paymenttx.paymentterm%type);

type dailytab is table of dailyrec

index by binary_integer;

procedure dailyquery(resultset in out dailytab,querydate integer,querymonth integer, queryyear
 integer,id location.location_id%type,rentfilter number);

function isrent(id paymenttx.paymenttx_id%type)return number;

function isins(id paymenttx.paymenttx_id%type)return number;

END;

PACKAGE BODY paymentdetailp IS

procedure dailyquery(resultset in out dailytab,querydate integer,querymonth integer, queryyear
 integer,id location.location_id%type,rentfilter number)is

counter integer;

txdate integer;

txmonth integer;

txyear integer;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

txlocate integer;
tempdate date;
tempcid number;
cursor cur is
    select *
    from paymenttxdetail;
currow paymenttxdetail%rowtype;
matched integer;
begin
    open cur;
    counter :=0;
    txdate :=0;
    txmonth :=0;
    txyear :=0;
    txlocate :=0;
    tempcid :=0;
loop
    fetch cur into currow;
    exit when cur%notfound or
        cur%notfound is NULL;
    tempcid := paymentp.getcontractid(currow.paymenttx_id);
    tempdate := paymentp.getdate(currow.paymenttx_id);
    txdate := to_number(to_char(tempdate,'dd'));
    txmonth := to_number(to_char(tempdate,'mm'));
    txyear := to_number(to_char(tempdate,'yyyy'));
    txlocate := projectp.getlocationid(unitp.getprojectid(contractp.getunitid
(tempcid)));
    matched :=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (querydate = 0) then -- Query for whole month..
    if (rentfilter = 1) then
        if (querymonth = txmonth) and (queryyear = txyear) and (txlocate = id)
and (currow.account_id = 1) then
            matched :=1;
        end if;
    else
        if (querymonth = txmonth) and (queryyear = txyear) and (txlocate = id)
then
            matched :=1;
        end if;
    end if;
else -- Query only one day...
    if (rentfilter = 1) then
        if (querydate = txdate) and(querymonth = txmonth) and (queryyear =
txyear) and (txlocate = id) and (currow.account_id=1) then
            matched :=1;
        end if;
    else
        if (querydate = txdate) and(querymonth = txmonth) and (queryyear =
txyear) and (txlocate = id) then
            matched :=1;
        end if;
    end if;
end if;
if matched = 1 then
    resultset(counter).paymenttx_id := currow.paymenttx_id;
    resultset(counter).paymentdate := tempdate;
    resultset(counter).location_id := txlocate;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        resultset(counter).contract_id := tempcid;
        resultset(counter).unit_id := contractp.getunitid(tempcid);
        resultset(counter).accountname :=
accountnp.getname(currow.account_id);
        resultset(counter).amount := currow.paymentprice;
        resultset(counter).term := paymentp.getterm(currow.paymenttx_id);
        counter := counter+1;
    end if;
end loop;
if (counter = 0) then
    resultset(0).accountname := "";
end if;
end;

function isrent(id paymenttx.paymenttx_id%type) return number is
temp number;
begin
    temp := 0;
    select account_id
    into temp
    from paymenttxdetail
    where paymenttx_id = id
    and account_id = 1;
    return temp;
exception
    when others then
        return 0;
end;

function isins(id paymenttx.paymenttx_id%type) return number is
temp number;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
```

```

temp:=0;
select account_id
into temp
from paymenttxdetail
where paymenttx_id = id
and account_id = 2;
if temp != 0 then
    temp:= 1;
end if;
return temp;

```

```
exception
```

```

when others then
return 0;

```

```
end;
```

```
END;
```

```
PACKAGE projectp IS
```

```

type projectrec is record( project_id project.project_id%type,
    projectname project.projectname%type,
    location_id location.location_id%type);

```

```
type projectcur is ref cursor return projectrec;
```

```

type projectsumrec is record( project_id project.project_id%type,
    projectname project.projectname%type,
    location_id project.location_id%type,
    unitcount number,
    rentcount number,
    freeunit number,
    dueamount number);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

type projectsumtab is table of projectsumrec

index by binary_integer;

procedure query(resultset in out projectcur,id location.location_id%type);

procedure avail(resultset in out projectsumtab,id project.location_id%type);

function getlocationid(id project.project_id%type)return number;

END;

PACKAGE BODY projectp IS

procedure query(resultset in out projectcur,id location.location_id%type) is

begin

open resultset for

select *

from project

where project.location_id = id;

end;

procedure avail(resultset in out projectsumtab,id project.location_id%type) is

counter integer;

cursor temp is

select *

from project

where location_id=id;

temprow project%rowtype;

begin

open temp;

counter := 0;

loop

fetch temp into temprow;

exit when temp%notfound or

temp%notfound is NULL;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

resultset(counter).project_id := temprow.project_id;
resultset(counter).projectname := temprow.projectname;
resultset(counter).location_id := temprow.location_id;
resultset(counter).unitcount := unitp.avail(temprow.project_id);
resultset(counter).rentcount :=unitp.rent(temprow.project_id);
resultset(counter).freeunit := unitp.freeunit(temprow.project_id);
resultset(counter).dueamount :=unitp.dueamount(temprow.project_id);
counter :=counter+1;

end loop;
if (counter = 0) then
    resultset(0).projectname :=";
end if;
end;

function getlocationid(id project.project_id%type)return number is
temp number;
begin
    select location_id
    into temp
    from project
    where project_id = id;
    return temp;

exception
    when others then
        return 0;

end;
END;

```

PACKAGE rentp IS

```
function getterm(id rentcontract.contract_id%type)return number;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function getprice(id rentcontract.contract_id%type) return number;
```

```
END;
```

```
PACKAGE BODY rentp IS
```

```
function getterm(id rentcontract.contract_id%type) return number is
```

```
temp number;
```

```
begin
```

```
    select insuranceterm
```

```
    into temp
```

```
    from rentcontract
```

```
    where contract_id = id;
```

```
    return temp;
```

```
exception
```

```
    when others then
```

```
        return 0;
```

```
end;
```

```
function getprice(id rentcontract.contract_id%type) return number is
```

```
temp number;
```

```
begin
```

```
    select insuranceprice
```

```
    into temp
```

```
    from rentcontract
```

```
    where contract_id = id;
```

```
    return temp;
```

```
exception
```

```
    when others then
```

```
        return 0;
```

```
end;
```

```
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PACKAGE unitp IS

```

type unitrec is record (unit_id unit.unit_id%type,
    debtorcode unit.debtorcode%type,
    floorno unit.floorno%type,
    buildingno unit.buildingno%type,
    model_id unit.model_id%type,
    project_id unit.project_id%type);
type unitcur is ref cursor return unitrec;

```

```

type unitavaiildetail is record (unit_id unit.unit_id%type,
    project_id unit.project_id%type,
    title customer.title%type,
    firstname customer.firstname%type,
    lastname customer.lastname%type,
    rentrate contract.priceperterm%type,
    method contract.paymenttype%type);

```

Type unitavaiildetailtab is table of unitavaiildetail

Index by binary_integer;

```

type unitfreerec is record (unit_id unit.unit_id%type,
    buildingtype_id buildingtype.buildingtype_id%type,
    buildingtypename buildingtype.buildingtypename%type,
    project_id unit.project_id%type,
    floorno unit.floorno%type,
    buildingno unit.buildingno%type,
    model_id unit.model_id%type,
    modelname model.modelname%type);

```

type unitfreetab is table of unitfreerec

index by binary_integer;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure freeunitlist(resultset in out unitfreetab,pid unit.project_id%type,bid
buildingtype.buildingtype_id%type);
procedure query(resultset in out unitcur);
procedure availdetail(resultset in out unitavaildetailtab,id unit.project_id%type);

function avail(id project.project_id%type)return number;
function rent(id project.project_id%type) return number;
function dueamount(id project.project_id%type)return number;
function freeunit(id project.project_id%type) return number;
function getprojectid(id unit.unit_id%type)return number;
END;
```

PACKAGE BODY unitp IS

```

procedure query(resultset in out unitcur) is
begin
    open resultset for
    select *
    from unit;
end;

function avail(id project.project_id%type)return number is
temp number;
temp1 number;
begin
    select max(project_id)
    into temp1
    from unit;
    if id <= temp1 then
        select count(*)
        into temp
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        from unit
        where unit.project_id = id
        group by project_id;
    else
        temp :=0;
    end if;
return (temp);
end;

```

```

function rent(id project.project_id%type) return number is
rentnum number;
temp1 number;
cursor unitcur is
    select *
    from unit
    where project_id = id;
unitrow unitcur%rowtype;
begin
    select max(project_id)
    into temp1
    from unit;
    if id <= temp1 then
        open unitcur;
        rentnum := 0;
        loop fetch unitcur into unitrow;
            exit when unitcur%notfound or
                unitcur%notfound is NULL;
            if contractp.status(unitrow.unit_id) = 1 then
                rentnum := rentnum +1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end if;

    end loop;

else
    rentnum :=0;

end if;

return rentnum;

end;

function freeunit(id project.project_id%type) return number is
frenum number;
temp1 number;

cursor unitcur is
select *
from unit
where project_id = id;
unitrow unitcur%rowtype;

begin
select max(project_id)
into temp1
from unit;

if id <= temp1 then
    open unitcur;
    frenum := 0;
    loop fetch unitcur into unitrow;
        exit when unitcur%notfound or
            unitcur%notfound is NULL;
            if (contractp.status(unitrow.unit_id) = 2) or
                (contractp.status(unitrow.unit_id) = 0) then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
freenum := freenum +1;
```

```
end if;
```

```
end loop;
```

```
else
```

```
freenum :=0;
```

```
end if;
```

```
return freenum;
```

```
end;
```

```
function dueamount(id project.project_id%type) return number is
```

```
dueamount number;
```

```
temp1 number;
```

```
cursor unitcur is
```

```
select unit_id,project_id
```

```
from unit
```

```
where project_id = id;
```

```
unitrow unitcur%rowtype;
```

```
begin
```

```
dueamount :=0;
```

```
select max(project_id)
```

```
into temp1
```

```
from unit;
```

```
if id <= temp1 then
```

```
open unitcur;
```

```
loop fetch unitcur into unitrow;
```

```
exit when unitcur%notfound or
```

```
unitcur%notfound is NULL;
```

```
dueamount :=dueamount + contractp.amount(unitrow.unit_id);
```

```
end loop;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    dueamount :=0;
end if;
return dueamount;
end;

```

```

procedure availdetail(resultset in out unitavaildetailtab,id unit.project_id%type)is

```

```

counter integer;

```

```

cursor unitcur is

```

```

select *

```

```

from unit

```

```

where project_id = id;

```

```

tempcus number;

```

```

unitrow unit%rowtype;

```

```

begin.

```

```

open unitcur;

```

```

counter := 0;

```

```

tempcus := 0;

```

```

loop

```

```

    fetch unitcur into unitrow;

```

```

    exit when unitcur%notfound or

```

```

        unitcur%notfound is NULL;

```

```

    if (contractp.status(unitrow.unit_id) = 1) then

```

```

        resultset(counter).unit_id := unitrow.unit_id;

```

```

        resultset(counter).project_id := unitrow.project_id;

```

```

        tempcus := contractp.getcusid(unitrow.unit_id);

```

```

        resultset(counter).rentrate := contractp.amount(unitrow.unit_id);

```

```

        resultset(counter).method := contractp.getmethod(unitrow.unit_id);

```

```

        resultset(counter).title := customerp.gettitle(tempcus);

```

```

        resultset(counter).firstname := customerp.getname(tempcus);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        resultset(counter).lastname := customerp.getlastname(tempcus);
        counter := counter + 1;
    end if;
end loop;
if (counter = 0) then
    resultset(0).title := "";
end if;
end;

procedure freeunitlist(resultset in out unitfreetab,pid unit.project_id%type,bid
buildingtype.buildingtype_id%type) is
counter integer;
temp number;
cursor unitcur is
select *
from unit
where project_id = pid;
unitrow unit%rowtype;
begin
open unitcur;
counter := 0;
temp := 0;
loop
    fetch unitcur into unitrow;
    exit when unitcur%notfound or
unitcur%notfound is NULL;
    if (contractp.status(unitrow.unit_id) = 0) or
(contractp.status(unitrow.unit_id) = 2) then
temp := modelp.gettype(unitrow.model_id);
if temp = bid then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

resultset(counter).unit_id := unitrow.unit_id;
resultset(counter).project_id := unitrow.project_id;
resultset(counter).floorno := unitrow.floorno;
resultset(counter).buildingno := unitrow.buildingno;
resultset(counter).buildingtype_id := temp;
resultset(counter).buildingtypename := buildingp.getname(temp);
resultset(counter).model_id := unitrow.model_id;
resultset(counter).modelname := modelp.getname(unitrow.model_id);
counter := counter + 1;

```

```

end if;

```

```

end if;

```

```

end loop;

```

```

if counter = 0 then

```

```

    resultset(0).modelname := "";

```

```

end if;

```

```

end;

```

Function getprojectid(id unit.unit_id%type) return number is

```

temp number;

```

```

begin

```

```

    select project_id

```

```

    into temp

```

```

    from unit

```

```

    where unit_id = id;

```

```

    return temp;

```

```

exception

```

```

    when others then

```

```

        return 0;

```

```

end;

```

```

END;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

นาย ภูสิทธิ์ รัตนปิยะสุนทร เกิดเมื่อวันพฤหัสบดีที่ 17 เมษายน พุทธศักราช 2518 ที่อำเภอ ภาษีเจริญ กรุงเทพมหานคร สำเร็จการศึกษาระดับประถมศึกษา มัธยมศึกษาตอนต้น และมัธยมศึกษาตอนปลายจาก โรงเรียนอัสสัมชัญธนบุรี จากนั้นจึงเข้าศึกษาต่อในระดับอุดมศึกษาที่ภาควิชา วิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และสำเร็จการศึกษาในปีพุทธศักราช 2539 โดยได้รับวุฒิปริญญาเป็น วิศวกรรมศาสตรบัณฑิต

หลังจากสำเร็จการศึกษาในระดับปริญญาตรีแล้ว ก็เข้าทำงานที่บริษัท แอดวานซ์ รีเสิร์ช อินฟอร์เมชัน แอนด์พับลิเคชัน ในตำแหน่งบรรณาธิการฝ่ายเทคนิค และเข้าศึกษาต่อในระดับปริญญาโทที่คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในสาขาวิชา วิทยาการสารสนเทศ

