

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบบริหารงานพัสดุตามระเบียบสำนักนายกรัฐมนตรี
Inventory Control System Complied with the Regulation of
the Prime Minister's Office

โดย

นางสาวกัลยลักษณ์ โพธิพันธ์

รหัส 42067115



H001878

อาจารย์ที่ปรึกษา

รศ.ดร.วิเชียร เปรมชัยสวัสดิ์

วัน เดือน ปี..... 5 ต.ค. 2550

เลขทะเบียน..... 01878

เลขเรียกหนังสือ..... จพ ก 3978 2544

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2544
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวเรื่อง	ระบบบริหารงานพัสดุตามระเบียบสำนักนายกรัฐมนตรี
นักศึกษา	นางสาว กัลยลักษณ์ โพรพันธ์
อาจารย์ที่ปรึกษา	รศ.ดร. วิเชียร เปรมชัยสวัสดิ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

ระบบบริหารงานพัสดุตามระเบียบสำนักนายกรัฐมนตรี เป็นระบบงานที่ใช้ควบคุมการเบิกจ่ายพัสดุ ซึ่งได้แก่ วัสดุ ครุภัณฑ์ ที่ดินและสิ่งปลูกสร้าง ของส่วนราชการ โดยได้มีการนำวิธีการวิเคราะห์และออกแบบระบบเชิงวัตถุ(Object Oriented Analysis and design Method) มาใช้เนื่องด้วยเทคโนโลยี และแนวคิดเชิงวัตถุมีจุดเด่นในด้านต่าง ๆ คือ การนำกลับมาใช้ใหม่ ในส่วนของโปรแกรม ทำให้การพัฒนาระบบงานทำได้อย่างรวดเร็วและมีคุณภาพที่ดีขึ้น และมีความง่ายต่อการบำรุงรักษา

ในการวิเคราะห์และออกแบบของระบบบริหารงานพัสดุตามระเบียบสำนักนายกรัฐมนตรี นี้ได้นำ UML (Unified Modeling Language) มาใช้เป็นเครื่องมือ ในการสร้างแบบจำลองเชิงวัตถุ (Object-Oriented Model) ของระบบงาน

Title	Inventory Control System Complied with the Regulations of the Prime Minister 's Office
Student	Miss Kunyaluk Phothiphun
Advisor	Assoc. Prof. Dr. Wichian Premchaiswadi
Level of study	Master of Science in Information Technology
Major	Information Science
Academic Year	2001

Abstract

Inventory Control System Complied with the Regulations of the Prime Minister's Office is the system designed to control the inventories of government sector such as supply, durable articles, land and building. Because of many advantages such as reusability, easier maintenance, high quality, Object Oriented Analysis and Design Method is choosed to applied for the Inventory Control System.

In this project, UML (Unified Modeling Language) is selected as a tool for analysis and design the Object-Oriented Model of the Inventory Control System.

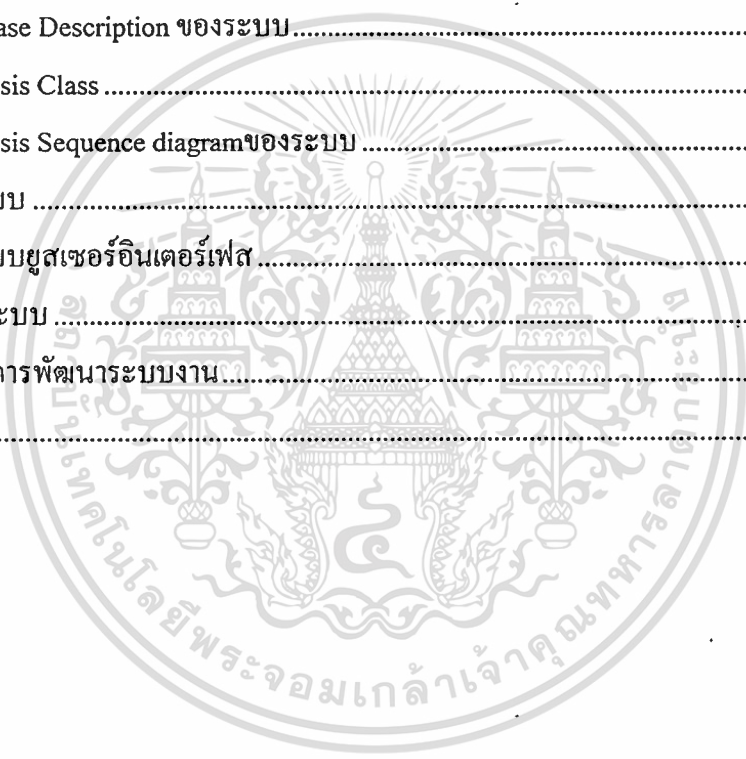
กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้สำเร็จลงได้ด้วยดี ต้องขอขอบพระคุณอาจารย์ที่ปรึกษาโครงการ คือ รองศาสตราจารย์ ดร.วิเชียร เปรมชัยสวัสดิ์ ที่ให้คำปรึกษาที่ดีเสมอมา และขอขอบพระคุณท่านอาจารย์คณะเทคโนโลยีสารสนเทศทุกท่านที่ได้กรุณาให้คำตอบเมื่อมีปัญหา ขอขอบคุณเพื่อนๆ ที่คอยเป็นกำลังใจ และสุดท้ายขอขอบคุณ คุณพ่อ และคุณแม่ ที่คอยเป็นกำลังใจ ทำให้โครงการสำเร็จลงได้ด้วยดี



บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	V
สารบัญภาพ.....	VI
บทที่	
1. บทนำ.....	1
1.1 ความสำคัญและที่มา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขั้นตอนการศึกษา.....	1
1.4 เครื่องมือที่ใช้ในการพัฒนา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 รายละเอียดในแต่ละบท.....	3
2. ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 ความแตกต่างของ OO-approach และ approach แรก ๆ.....	4
2.2 หลักการของ Object-Oriented.....	4
2.3 โมเดล.....	6
2.4 UML.....	6
2.4.1 Use case diagram.....	7
2.4.2 Sequence diagram.....	9
2.4.3 Collaboration diagram.....	10
2.4.4 Class diagram.....	10
2.4.5 State Transition diagram.....	13
2.4.6 Activity diagram.....	14
2.4.7 Component diagram.....	15

2.4.8	Deployment diagram.....	16
3.	การวิเคราะห์.....	17
3.1	ความต้องการของระบบ.....	17
3.2	Use case diagram ของระบบ	17
3.3	Use case Description ของระบบ.....	21
3.4	Analysis Class	30
3.5	Analysis Sequence diagramของระบบ.....	31
4.	การออกแบบ	39
5.	การออกแบบยูสเซอร์อินเตอร์เฟซ.....	57
6.	การสร้างระบบ	78
7.	สรุปโครงการพัฒนาระบบงาน.....	83
	บรรณานุกรม.....	95



สารบัญตาราง

ตารางที่	หน้า
5.1 คำอธิบายเมนูย่อยของเมนูเพิ่มข้อมูลหลัก	58
5.2 คำอธิบายเมนูย่อยของเมนูวัสดุ.....	58
5.3 คำอธิบายเมนูย่อยของเมนูครุภัณฑ์.....	59
5.4 คำอธิบายเมนูย่อยของเมนูที่ดิน/สิ่งปลูกสร้าง.....	60



สารบัญภาพ

รูปที่	หน้า
2.1 แสดงตัวอย่างของ Use Case Diagram	7
2.2 แสดงสัญลักษณ์ของ Actor.....	8
2.3 แสดงสัญลักษณ์ของ Use Case	8
2.4 แสดงตัวอย่างของ Sequence Diagram	9
2.5 แสดงตัวอย่างของ Collaboration Diagram	10
2.6 แสดงส่วนประกอบของ Class	11
2.7 แสดงคลาสไดอะแกรมในความสัมพันธ์แบบ aggregation	12
2.8 แสดงคลาสไดอะแกรมในความสัมพันธ์แบบ Composition aggregation	12
2.9 แสดงคลาสไดอะแกรมในความสัมพันธ์แบบ Dependency	12
2.10 แสดงคลาสไดอะแกรมในความสัมพันธ์แบบ Generalization	13
2.11 แสดงตัวอย่างของ State Transition Diagram	13
2.12 แสดงตัวอย่างของ Activity Diagram.....	14
2.13 แสดงสัญลักษณ์ Component Package	15
2.14 แสดงสัญลักษณ์ Component	15
2.15 แสดงสัญลักษณ์ Interface	15
2.16 แสดงสัญลักษณ์ Dependency Relationship	16
2.17 แสดงตัวอย่างของ Deployment Diagram	16
3.1 Use Case Diagram งานรับ-จ่ายวัสดุ	18
3.2 Use Case Diagram งานรับ-จ่ายครุภัณฑ์	18
3.3 Use Case Diagram งานบันทึกการซ่อมครุภัณฑ์	19
3.4 Use Case Diagram จำหน่ายครุภัณฑ์	19

สารบัญภาพ(ต่อ)

รูปที่	หน้า
3.5 Use Case Diagram ตรวจสอบพัสดุประจำปี	20
3.6 Use Case Diagram ขอขึ้นทะเบียนที่ราชพัสดุ	20
3.7 Use Case Diagram งานกำหนดข้อมูลหลัก	21
3.8 Boundary Class	30
3.9 Entity Class	31
3.10 Control Class	31
3.11 Sequence Diagram รับวัสดุ	32
3.12 Sequence Diagram เบิกวัสดุ	33
3.13 Sequence Diagram รับครุภัณฑ์	34
3.14 Sequence Diagram เบิกครุภัณฑ์	35
3.15 Sequence Diagram ยืมครุภัณฑ์	36
3.16 Sequence Diagram คืนครุภัณฑ์	37
3.17 Sequence Diagram บันทึกผลตรวจพัสดุประจำปี	38
4.1 Class Diagram ของระบบงานบริหารงานพัสดุตามระเบียบสำนักงานนายกรัฐมนตรี ระดับที่ 1	40
4.2 Class Diagram ของระบบงานบริหารงานพัสดุตามระเบียบสำนักงานนายกรัฐมนตรี ระดับที่ 2	41
4.3 Class Diagram ของระบบงานบริหารงานพัสดุส่วนที่ 1	42
4.4 Class Diagram ของระบบงานบริหารงานพัสดุส่วนที่ 2	43
4.5 Class Diagram ของระบบงานบริหารงานพัสดุส่วนที่ 3	43
4.6 Class Diagram ของระบบงานบริหารงานพัสดุส่วนที่ 4	44
4.7 Class Diagram ของระบบงานบริหารงานพัสดุส่วนที่ 5	44
4.8 แสดงการสร้างคอมโพเนนต์	45

สารบัญภาพ(ต่อ)

รูปที่	หน้า
4.9 แสดงการกำหนดสเตอร์ไอโปปี้ให้กับคลาส Employee	45
4.10 แสดงการกำหนดไอโปปี้ให้คลาส Employee อยู่ในคอมโพเนนต์ project	46
4.11 แสดงการสร้างคลาสที่มีสเตอร์ไอโปปี้เป็น <<Relational Table>>	46
4.12 แสดงคลาสรีเลชันของ หน่วยงานและเจ้าหน้าที่	47
4.13 แสดงคลาสรีเลชันใบรับพัสดุ ส่วนหลัก ส่วนรายละเอียด และกอง หน่วยงาน และเจ้าหน้าที่	47
4.14 แสดงคลาสรีเลชันของรหัสกลุ่มวัสดุ ประเภทวัสดุ และวัสดุ	48
4.15 แสดงคลาสรีเลชันของใบรับวัสดุ(รายการหลัก) บริษัท และเจ้าหน้าที่	48
4.16 แสดงคลาสรีเลชันของใบรับวัสดุ(รายการหลัก, รายการย่อย) และวัสดุ	49
4.17 แสดงคลาสรีเลชันของใบเบิกวัสดุ(รายการหลัก, รายการย่อย) และวัสดุ	49
4.18 แสดงการสร้าง Schema Generation	50
4.19 แสดงการอ้างถึงคอมโพเนนต์	51
4.20 แสดงคำสั่งนิยามฐานข้อมูลที่ได้ในไฟล์ Schema.sql	52
5.1 แสดงเมนูหลักของระบบบริหารงานพัสดุ	57
5.2 แสดงเมนูย่อยของเมนูเพิ่มข้อมูลหลัก	57
5.3 แสดงเมนูย่อยของเมนูวัสดุ	58
5.4 แสดงเมนูย่อยของเมนูครุภัณฑ์	59
5.5 แสดงเมนูย่อยของเมนูที่ดิน/สิ่งปลูกสร้าง	60
5.6 แสดงตัวอย่างของแถบสถานะ	60
5.7 แสดงลักษณะของปุ่มคำสั่ง	60
5.8 แสดงรายการค่าข้อมูล	61
5.9 แสดงไดอะล็อกแสดงข้อความเตือน	61
5.10 แสดงไดอะล็อกเข้าสู่ระบบ	62

สารบัญภาพ(ต่อ)

รูปที่	หน้า
5.11 แสดงฟอร์มข้อมูลหน่วยงาน	62
5.12 แสดงฟอร์มข้อมูลกลุ่มวัสดุ	63
5.13 แสดงฟอร์มข้อมูลประเภทวัสดุ	64
5.14 แสดงฟอร์มข้อมูลวัสดุ	65
5.15 แสดงฟอร์มข้อมูลรับวัสดุ	66
5.16 แสดงฟอร์มข้อมูลเบิกวัสดุ	67
5.17 แสดงฟอร์มข้อมูลรับครุภัณฑ์	68
5.18 แสดงฟอร์มข้อมูลเบิกครุภัณฑ์	69
5.19 แสดงฟอร์มข้อมูลยืมครุภัณฑ์	70
5.20 แสดงฟอร์มข้อมูลคืนครุภัณฑ์	71
5.21 แสดงฟอร์มข้อมูลรับแจ้งซ่อมครุภัณฑ์	72
6.1 ตัวอย่าง Source Code ส่วน Package Specification-ของ Inventory Package ...	79
6.2 ตัวอย่าง Source Code ส่วน Package Body-ของ Inventory Package	80
6.3 Component Diagram ของระบบ	82
6.4 Deployment Diagram ของระบบ	83

บทที่ 1

บทนำ

1.1. ความสำคัญและที่มา

การพัฒนาระบบเชิงวัตถุ เป็นวิธีการที่กำลังได้รับความนิยมกันอย่างมากเนื่องจากสามารถพัฒนาระบบได้ในเวลารวดเร็ว ด้วยต้นทุนที่ต่ำเพราะสามารถนำเอาโปรแกรมที่เคยสร้างไว้แล้วกลับมาใช้ใหม่ได้ ทำให้ได้ระบบที่มีคุณภาพและมีประสิทธิภาพดียิ่งขึ้น นอกจากนี้ยังสามารถบำรุงรักษาได้ง่ายโดยไม่กระทบกับส่วนอื่น ๆ ที่ไม่เกี่ยวข้องกัน

Unified approach (UA) เป็น methodology หนึ่งที่น่ามาใช้ในการพัฒนาแบบเชิงวัตถุ โดยได้รวบรวมส่วนดี ๆ ของแต่ละ methodology ไว้ และใช้ Unified Modeling Language (UML) ซึ่งเป็นกลุ่มของ notation ที่ใช้ในการจำลองรูปแบบของระบบงาน โดยรูปแบบนั้น สามารถแสดงได้ทั้ง 2 แบบ คือ เริงสถติก โมเดล (Static model) และเริงไดนามิก โมเดล (Dynamic model)

1.2. วัตถุประสงค์ของโครงการ

เพื่อเป็นแนวทางในการพัฒนาโปรแกรมเพื่อช่วยในการแก้ปัญหาที่มีอยู่ในระบบงานเดิม ซึ่งพบว่ายังมีปัญหา และต้องการการปรับปรุงอีกมาก

1.3. ขั้นตอนการศึกษา

1. ศึกษาถึงความต้องการในปัจจุบัน พร้อมทั้งปัญหาและกำหนดขอบเขตและจุดประสงค์ต่าง ๆ ของโครงการ
2. วิเคราะห์และออกแบบระบบงานและระบบฐานข้อมูลพร้อมทั้งเลือกใช้เครื่องมือที่จะนำมาพัฒนาต่อไป
3. ทำการพัฒนาแบบงานตามที่ออกแบบไว้ข้างต้น
4. ทำการทดสอบระบบว่าสามารถทำงานได้ตามจุดประสงค์ที่กำหนดไว้หรือไม่
5. ทำการปรับปรุงแก้ไขเพื่อให้ได้ระบบงานที่สมบูรณ์ตรงตามเป้าหมายที่กำหนดไว้
6. สรุปผลการศึกษา และเขียนรายงานรวมทั้งจัดทำคู่มือการใช้งาน

1.4. เครื่องมือที่ใช้ในการพัฒนา

1.4.1. ฮาร์ดแวร์ (Hardware)

- CPU AMD K6-2 400 MHz
- RAM 64 MB
- Harddisk 10 GB

1.4.2. ซอฟต์แวร์ (Software)

- ระบบปฏิบัติการ – Microsoft Windows 98
- ฐานข้อมูล Oracle
- Rational Rose

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1. เป็นการพัฒนาระบบเชิงวัตถุ และสามารถนำเอาโปรแกรมที่เคยสร้างไว้แล้วกลับมาใช้ใหม่ได้ ทำให้ได้ระบบที่ได้มีคุณภาพและมีประสิทธิภาพดียิ่งขึ้น
2. ระบบที่ได้สามารถบำรุงรักษาได้ง่าย โดยไม่กระทบกับส่วนอื่นๆ ที่ไม่เกี่ยวข้องกัน
3. ผู้ใช้ได้โปรแกรมที่ง่ายต่อการใช้งาน

1.6 รายละเอียดของแต่ละบท

- บทที่ 1 เป็นบทนำกล่าวถึง ความสำคัญและที่มาของการนำเอาหลักการพัฒนาโปรแกรมเชิงวัตถุมาใช้ งาน วัตถุประสงค์ของระบบ ขอบเขตของระบบ และรายละเอียดของแต่ละบทที่นำเสนอในรายงานฉบับนี้
- บทที่ 2 อธิบายทฤษฎีที่เกี่ยวข้องที่นำมาใช้ประกอบในการวิเคราะห์และออกแบบระบบเชิงวัตถุ กล่าวถึงขั้นตอนวิธี และการนำโคออร์เดเนทของยูเอ็มแอลมาใช้ในการวิเคราะห์และออกแบบระบบ
- บทที่ 3 ทำการวิเคราะห์ระบบด้วยวิธีเชิงวัตถุ แสดงภาพรวมของระบบด้วยการ วิเคราะห์ยูสเคสโคออร์เดเนทและสร้างซีเควิน โคออร์เดเนทเพื่อหาคลัสของระบบ
- บทที่ 4 ทำการออกแบบระบบด้วยวิธีเชิงวัตถุ โดยทำการออกแบบคลัสที่วิเคราะห์ได้ในบทที่ 3 ในรายละเอียดของแอตทริบิวต์ และสร้างซีเควิน โคออร์เดเนทเพื่อหาเมธอดของแต่ละคลัส และอธิบายวิธีในการแปลงคลัสโคออร์เดเนทมาเป็นฐานข้อมูลเชิงสัมพันธ์โดยอาศัย ความสามารถของ Rational Rose ในทำ Forward Engineering มาเป็นคำสั่งในการนิยามฐานข้อมูลของ Oracle 8
- บทที่ 5 กล่าวถึงการออกแบบยูสเซอร์อินเตอร์เฟซ ได้แก่ เมนู แบบฟอร์ม และรายงาน
- บทที่ 6 การพัฒนาระบบ โดยนำสิ่งที่ได้จากการวิเคราะห์และออกแบบระบบมาพัฒนาระบบ แสดงการแปลงเมธอดจากคลัสมาเป็นสตอ โปรซีเจอร์ใน Oracle แสดงองค์ประกอบของซอฟต์แวร์ด้วยคอม โปเนนท์โคออร์เดเนท และแสดงสถาปัตยกรรมของระบบด้วยดีพลอยเมนต์โคออร์เดเนท
- บทที่ 7 สรุปโครงการพัฒนาซอฟต์แวร์ปัญหาและอุปสรรคในการดำเนินงาน ประโยชน์ที่ได้รับ และข้อเสนอแนะต่างๆ

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

Object-Oriented approach เป็นการมอง application ที่แตกต่างไป ด้วยหลักการของ Object-Oriented approach จะทำการแบ่ง application เป็น ส่วน ๆ ที่ค่อนข้างจะเป็นอิสระต่อกัน การสร้าง application ก็จะเป็นการนำแต่ละส่วนมารวมกันนั่นเอง

ประโยชน์ของ Object-Oriented approach ก็คือ ความสามารถที่จะสร้าง component ขึ้นมา และสามารถใช้ในครั้งต่อ ๆ ไป ได้อีก

2.1 ความแตกต่างของ OO-approach และ approach แรก ๆ คือ

approach แรก ๆ จะมีลักษณะที่เรียกว่า data-centric คือ จะมุ่งไปที่ information ของระบบ ว่ามีอะไรบ้าง โดยไม่ได้สนใจเกี่ยวกับพฤติกรรมของระบบ ซึ่งหาก requirement ของระบบเปลี่ยนแปลงไป หากเกี่ยวกับในส่วนของ การเปลี่ยนแปลง database จะกระทำได้ง่าย แต่หากต้องมีการเปลี่ยนแปลงในส่วนของ business rule หรือ behavior ของระบบ การ implement ก็จะมี ความยากขึ้น

OO-approach จะสนใจทั้ง information และ behavior ของระบบ ทำให้การพัฒนา ระบบมีความยืดหยุ่นสามารถรองรับต่อการเปลี่ยนแปลง behavior ของระบบได้

2.2 หลักการของ Object-Oriented

ลักษณะที่สำคัญบางประการของ Object-Oriented ได้แก่ encapsulation inheritance และ polymorphism ซึ่งคุณสมบัติเหล่านี้เองที่ทำให้การพัฒนา ระบบมีความยืดหยุ่น

• Encapsulation

เป็นการซ่อน information และ behavior เข้าไว้ใน object นั้น ๆ เช่น ในบัญชีธนาคารของ ลูกค้า information ที่สนใจได้แก่ หมายเลขบัญชี ชื่อลูกค้า ที่อยู่ ประเภทบัญชี เป็นต้น และมี behavior ของ บัญชีธนาคาร ได้แก่ เปิด-ปิดบัญชี ฝาก ถอน เปลี่ยนที่อยู่ เป็นต้น เราก็จะนำ information และ behavior รวมกันไว้ที่ object หนึ่ง หากต้องมีการเปลี่ยนแปลงภายหลัง ก็จะสามารถทำกับ object นั้น ๆ โดยไม่ต้องกระทบกับ object อื่น ๆ

- **Inheritance**

เป็นวิธีการที่ทำให้การสร้าง object ใหม่ (อาจเรียกว่า child object) ที่มีพื้นฐานจากอีก object หนึ่ง(อาจเรียกว่า parent object) โดย child object จะถ่ายทอดคุณสมบัติมาจาก parent object

ประโยชน์ คือ ทำให้ง่ายต่อการ maintenance ยกตัวอย่าง เช่น แมว สุนัข คน จะ inherit คุณสมบัติต่าง ๆ มาจาก สัตว์เลี้ยงลูกด้วยนม หากมีการเปลี่ยนแปลง ก็เปลี่ยนเฉพาะคลาสของสัตว์เลี้ยงลูกด้วยนม ส่วนคลาสของแมว สุนัข คน ก็จะ inherit การเปลี่ยนแปลงนั้นอย่างอัตโนมัติ

- **Polymorphism**

หมายถึงฟังก์ชันหนึ่งมีลักษณะการ implement ได้หลาย ๆ แบบ ตัวอย่างเช่น

- กรณีไม่มีลักษณะของ *polymorphism*

Function Shape.drawMe()

{

CASE Shape.Type

Case "Circle"

Shape.drawCircle();

Case "Rectangle"

Shape.drawRectangle();

Case "Line"

Shape.drawLine();

END CASE

}

- กรณีมีลักษณะของ *polymorphism*

Function draw()

{

Shape.drawMe();

}

ในแต่ละ shape (ซึ่ง ได้แก่ circle line rectangle หรือ อื่น ๆ) จะมีฟังก์ชัน drawMe() ที่ใช้ในการวาดรูปเฉพาะของแต่ละ shape เอง

ประโยชน์ คือ ทำให้ง่ายต่อการ maintenance เช่น จากตัวอย่างข้างบนดังกล่าว หากต้องการวาดรูปสามเหลี่ยม(triangle) ฟังก์ชัน draw() ก็ไม่ต้องเปลี่ยนแปลง เพียงแต่สร้างฟังก์ชัน drawMe() ของรูปสามเหลี่ยมเพิ่มขึ้นมา

2.3 โมเดล (Model)

เป็นส่วนที่สร้างขึ้นมาเพื่อให้สามารถเข้าใจระบบงาน ก่อนที่จะทำการสร้างหรือแก้ไขระบบงานหนึ่งๆ โดยเราสามารถแบ่งโมเดลออกเป็น ออกเป็น 2 แบบคือ เชิงสถติก โมเดล (Static Model) และ เชิงไดนามิก โมเดล (Dynamic Model)

- *Static Model* เป็น โมเดลที่ใช้แสดงมุมมองจากผู้ใช้งานนอกต่อพฤติกรรมของระบบ และเห็นถึงภาพรวมของระบบ ตัวอย่างหนึ่งของ สถติกโมเดล ได้แก่ โครงสร้างของคลาสและความสัมพันธ์กันระหว่างคลาส

- *Dynamic Model* เป็น โมเดลที่รวบรวมของการปฏิบัติ หรือพฤติกรรมที่เกิดขึ้นภายในระบบ เช่น ใบสั่งซื้อ สินค้าจะมีความสัมพันธ์กับสต็อกสินค้าในระบบโดยนำไปใช้ในการวางแผนการผลิตให้มีจำนวนเพียงพอสำหรับการสั่งซื้อในครั้งต่อไป

การสร้างโมเดล จะมีประโยชน์ดังนี้คือ

- ทำให้ความคิดที่ซับซ้อนสามารถแสดงออกมาได้ง่ายขึ้น
- สามารถลดความซับซ้อนของระบบได้โดยทำการแยกลักษณะที่ไม่สำคัญออกไปทำให้สิ่งที่มีความซับซ้อนเข้าใจได้ง่ายขึ้น
- โมเดลต่าง ๆ สามารถเรียนรู้และฝึกฝนได้ง่าย
- การแก้ไขโมเดลเมื่อมีการเปลี่ยนแปลงเกิดขึ้นก็สามารถทำได้ง่ายกว่าทำกับระบบจริง

2.4 UML (Unified Modeling Language)

UML (Unified Modeling Language) คือ ภาษาที่ใช้แทน method ของ Object Oriented Analysis and Design (OOA&D) และสามารถนำมาใช้เป็น โมเดลในการออกแบบระบบที่ครอบคลุมได้กว้าง ถึงแม้ว่าระบบต่าง ๆ นั้นจะมีความแตกต่างกันก็ตาม UML ก็สามารถบอกถึงรายละเอียดได้อย่างชัดเจน รวมทั้งการพัฒนาซอฟต์แวร์ในเฟสต่าง ๆ UML ก็สามารถบอกถึงรายละเอียดของแต่ละเฟสได้เช่นกัน ไม่ว่าจะเป็ความค้องการของระบบที่ซับซ้อนหรือ

การทดสอบการทำงานของระบบในขั้นตอนต่าง ๆ โดยมี diagram ที่ใช้ในการพัฒนาระบบต่าง ๆ ดังนี้ คือ

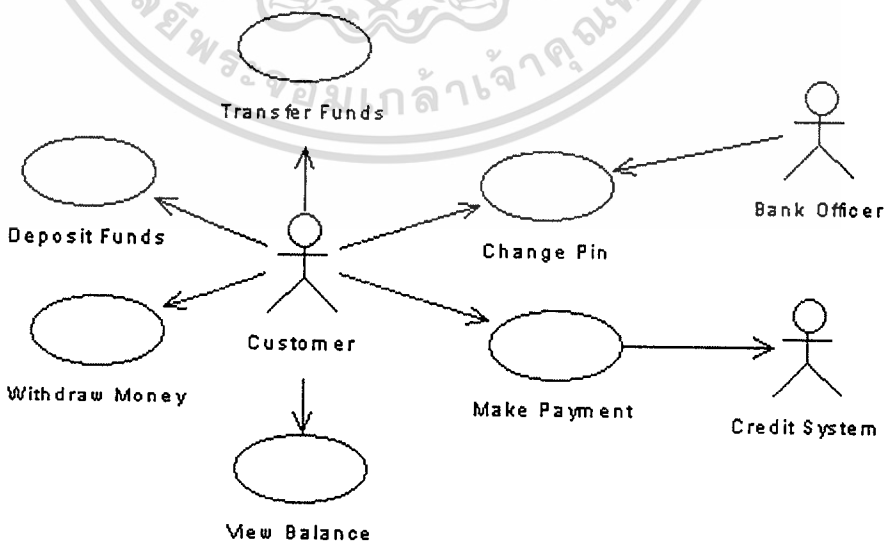
- Use case diagram
- Class diagram
- Sequence diagram
- Collaboration diagram
- State Transition diagram
- Activity diagram
- Component diagram
- Deployment diagram

ส่วนที่เป็นสถิตโมเดลที่นำมาใช้ในส่วนของการวิเคราะห์และออกแบบระบบ คือ *Use Case Diagram* และ *Class Diagram*

และส่วนที่เป็นไดนามิกโมเดลที่นำมาใช้ในส่วนของการวิเคราะห์และออกแบบระบบ คือ *Sequence Diagram* *Collaboration Diagram* *Statechart Diagram* และ *Activity Diagram*

2.4.1 Use Case Diagram

Use Case Diagram เป็นการแสดงความต้องการของระบบทั้งหมดในลักษณะที่ผู้ใช้สามารถเข้าใจได้ง่าย และเป็นการแสดงความสัมพันธ์ระหว่างผู้ใช้กับระบบต่าง ๆ ตัวอย่างของ Use case diagram แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 แสดงตัวอย่างของ Use Case diagram

โดยสามารถอธิบายสัญลักษณ์ที่ใช้ในไดอะแกรมได้ดังนี้

Actor คือ ผู้กระทำซึ่งมีหน้าที่แสดงถึงความสัมพันธ์ของระบบ และเป็นสิ่งที่อยู่นอกระบบ ในการพัฒนาระบบขึ้นมาทั้งหมด ซึ่งจะทำการแลกเปลี่ยนหรือส่งข้อมูลให้กับระบบ โดยอาจจะ เป็นคน หรือ ระบบงาน โดยจะใช้ รูปคน แทนสัญลักษณ์ของ Actor

สัญลักษณ์ ของ Actor



รูปที่ 2.2 แสดงสัญลักษณ์ของ Actor

Use Case เป็นตัวแทนงานที่เกิดขึ้นในขั้นตอนต่าง ๆ โดยใช้รูปวงรี หรือ วงกลม เป็น สัญลักษณ์

สัญลักษณ์ ของ Use Case



รูปที่ 2.3 แสดงสัญลักษณ์ของ Use Case

ความสัมพันธ์กันภายใน Use Case Diagram มี 3 แบบ คือ

- **Communication** เป็นการแสดงความสัมพันธ์ หรือการติดต่อสื่อสารกัน ระหว่าง Actor และ Use case ปลายเส้นของลูกศรมีลักษณะเป็นหัวเปิด
- **Extends** เป็นการแสดงความสัมพันธ์ ระหว่าง Use Case กับ Use Case ปลายเส้นของ ลูกศรมีลักษณะเป็นหัวปิดและมีข้อความ <<extends>> ระบุอยู่ข้างเส้น ใช้เมื่อมี use case หนึ่งมี ลักษณะการทำงานที่คล้ายคลึงกันแต่มีบางส่วนที่ต่างกันเพื่อทำงานเฉพาะงานนั้น ซึ่งมีลักษณะ เหมือนกับ subclass
- **Uses** เป็นการแสดงความสัมพันธ์ ระหว่าง Use Case กับ Use Case ปลายเส้นของลูก ศรมีลักษณะเป็นหัวเปิดและมีข้อความ <<uses>> ระบุอยู่ข้างเส้น ใช้เมื่อมีการกระทำที่ซ้ำ ๆ กัน ในหลาย use case และเราต้องการกระทำนั้นมาสร้างเป็น use case ใหม่

คุณสมบัติของ Use Case

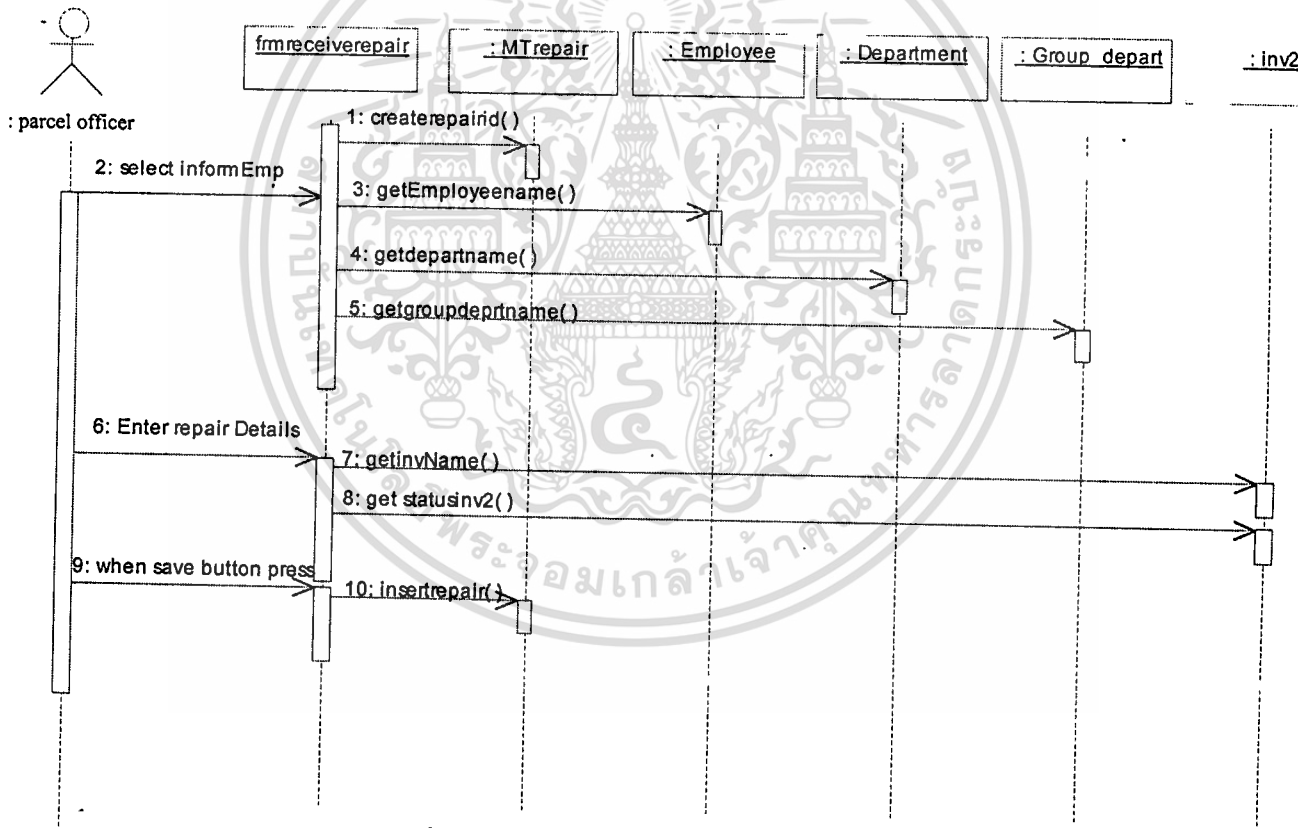
- สามารถกำหนดขอบเขตของฟังก์ชันให้กับ Use Case ได้อย่างชัดเจน
- อาจมีขนาดเล็ก หรือ ใหญ่ก็ได้

- สามารถเข้าถึงเป้าหมายที่แตกต่างกันของผู้ใช้ได้

Actor จะสนับสนุนการทำ Use Case ซึ่งใน 1 Actor สามารถทำงานได้หลาย ๆ Use Case โดยอาจจะมีการเจาะจง Actor ที่มันกระทำด้วย ซึ่งจะมีประโยชน์สำหรับการทำงานในระบบใหญ่ ๆ ทำให้เข้าใจได้ง่าย โดยไม่จำเป็นต้องเป็นคน อาจจะเป็นเหตุการณ์ต่าง ๆ ที่แสดงด้วยรูปภาพได้

2.4.2 Sequence Diagram

ใช้แสดงการทำงานของออบเจกต์ที่มีการติดต่อกันในระบบตามลำดับของเวลา การอ่านไดอะแกรมจะอ่านจากบนลงล่าง และจะมีหมายเลขแสดงลำดับของการทำงาน แต่ละซีเควินไดอะแกรมจะแสดงภาพของการทำงานในแต่ละ ยูสเคส นั้นเอง ตัวอย่างของซีเควินไดอะแกรมแสดงได้ดังรูปที่ 2.4

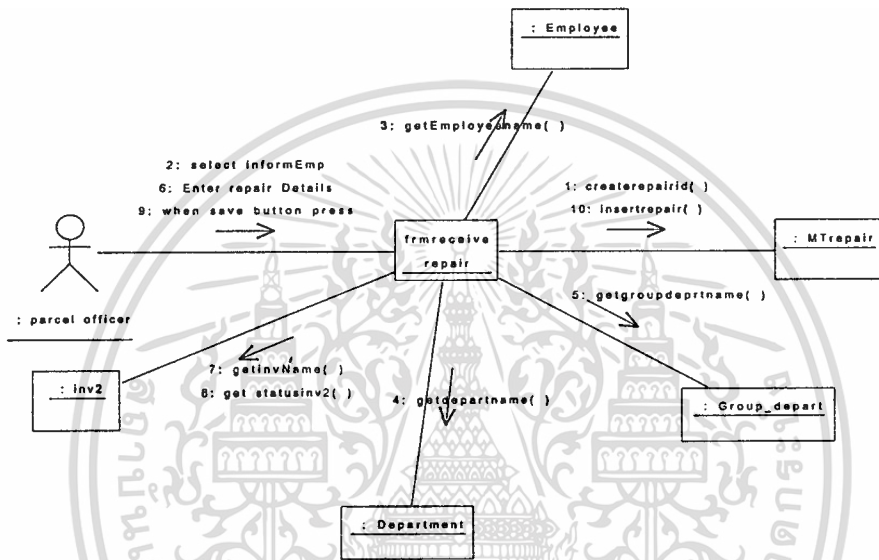


รูปที่ 2.4 แสดงตัวอย่างซีเควินไดอะแกรม

จากรูปที่ 2.4 แต่ละออบเจกต์จะมีเส้นตามแนวตั้ง ซึ่งเรียกว่า *lifeline* และจะมีข้อความที่ใช้ในการติดต่อกันระหว่าง ออบเจกต์ และข้อความต่าง ๆ ดังกล่าวนี้อาจจะเป็น operation ใน Class Diagram ต่อไป

2.4.3 Collaboration Diagram

ใช้แสดงการทำงานของออบเจกต์ที่มีการติดต่อกันเช่นเดียวกับซีเควิน โคอะแกรม แต่ต่างกันตรงที่ คอลแลบอเรนซ์ โคอะแกรมแสดงการไหลของข้อมูลในขณะที่ ซีเควิน โคอะแกรมไม่ได้แสดงในส่วนนี้ ตัวอย่างของคอลแลบอเรนซ์ โคอะแกรมแสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 แสดงตัวอย่างคอลแลบอเรนซ์ โคอะแกรม

2.4.4 Class Diagram

เป็นสเตคโมเดลหลักที่ใช้ในการวิเคราะห์ระบบ จะเป็นการอธิบายถึง Class และความสัมพันธ์ระหว่าง Class ที่มีโครงสร้างของข้อมูลรวมถึงพฤติกรรมของข้อมูลที่แตกต่างกัน ซึ่ง Class 1 Class นั้นสามารถกำหนดทิศทางของการนำไปสร้างโปรแกรมและการสร้าง Class ใน Object-oriented ได้

ในการสร้าง Class Diagram จะมีการบ่งบอกถึงรายละเอียดและลำดับของ Class ซึ่งมีความสัมพันธ์กันในแต่ละ Class ได้แบ่งออกเป็น 3 ส่วน คือ

Name Compartment

Attribute Compartment

Operational Compartment

โดยสามารถแสดงส่วนประกอบของ Class ได้ดังรูปที่ 2.6

Name
Attribute
Operation

รูปที่ 2.6 แสดงส่วนประกอบของ Class

จากรูปที่ 2.3 อธิบายได้ดังนี้ คือ

Name : จะแสดงชื่อของ Class ที่กำหนดในระบบ

Attribute : เป็นการกำหนดคุณลักษณะทั้งหมดที่มีภายใน Class และบอกถึงรายละเอียดของข้อมูล(attribute) ภายใน Class เดียวกันจะไม่ซ้ำกัน แต่ชื่อข้อมูลใน Class อาจจะไปซ้ำกับชื่อข้อมูลใน Class อื่น ๆ ก็ได้

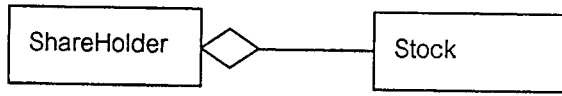
Operation : แสดงคุณลักษณะและหน้าที่ของ Class ซึ่งอาจจะเป็นค่าตัวแปรของ Object ใน Class และบางครั้งอาจเป็นการอธิบายถึงการกระทำร่วมกับ Class อื่นด้วย โดยอาจเรียกว่าเป็น “Function”

ในคลาสไดอะแกรมจะมีการกำหนดจำนวนความสัมพันธ์ (multiplicity) โดยระบุเป็นตัวเลข หรือช่วงของตัวเลข ในรูปแบบ ค่าต่ำสุด..ค่าสูงสุด ไว้ที่เส้นของเส้นแสดงความสัมพันธ์ ถ้าเป็นเลขจำนวนเดียวหมายถึงค่าที่แน่นอน ถ้าเป็นช่วงหมายถึงค่าที่เป็นไปได้ ถ้าหมายถึงจำนวนใด ๆ ให้แทนด้วยเครื่องหมายดอกจัน(*)

ความสัมพันธ์ระหว่าง Class ได้แก่

- **Association** จะเป็นการแสดงความสัมพันธ์ระหว่าง Class จะมีได้ทั้งทางเดียวและสองทาง โดยเขียนคำกำกับบอกบทบาทของ class นั้น ๆ
- **Aggregation** เป็นรูปแบบพิเศษของความสัมพันธ์ คือเป็นความสัมพันธ์ระหว่าง Whole และ Parts ของมัน คือ Whole จะประกอบไปด้วย Parts ต่าง ๆ ของมัน ดังนั้น การคงอยู่ของ Parts จะต้องขึ้นอยู่กับ Whole หรือเรียกความสัมพันธ์แบบนี้ว่า “Whole-Parts” โดยมีสัญลักษณ์ดังนี้

- สัญลักษณ์ลูกศรที่มีหัวรูปสี่เหลี่ยมขนมเปียกปูน โปร่งอยู่ทางด้าน Whole ดังรูปที่ 2.7



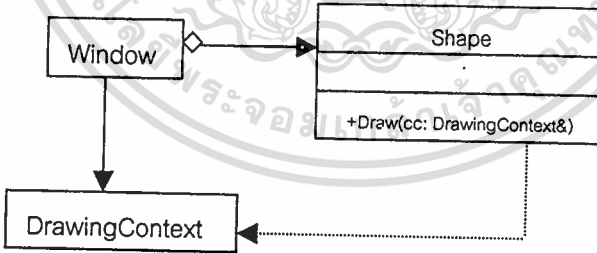
รูปที่ 2.7 แสดงคลาสไดอะแกรม ในความสัมพันธ์แบบ aggregation

- สัญลักษณ์ลูกศรที่มีหัวรูปสี่เหลี่ยมขนมเปียกปูนทึบอยู่ทางด้าน Whole จะหมายถึง ถ้าไม่มี Class ทางด้าน Whole Class ทางด้าน Part ก็จะไม่เกิด โดยเรียกความสัมพันธ์แบบ Composition aggregation ดังรูปที่ 2.8



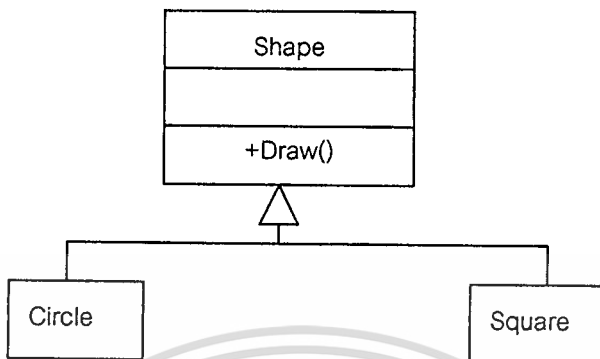
รูปที่ 2.8 แสดงคลาสไดอะแกรม ในความสัมพันธ์แบบ Composition aggregation

- **Dependency** เป็นความสัมพันธ์ที่ขึ้นตรงต่อกัน เมื่อมีการแก้ไข Class หนึ่งจะมีผลกระทบต่ออีก Class หนึ่งขึ้นต่อกัน เส้นที่เชื่อมโยงกันเป็นเส้นประและลูกศรหัวเปิด ดังรูปที่ 2.9



รูปที่ 2.9 แสดงคลาสไดอะแกรม ในความสัมพันธ์แบบ Dependency

- **Generalization** เป็นความสัมพันธ์แบบ SuperClass/Subclass ใช้สัญลักษณ์ลูกศรหัวสามเหลี่ยม โปร่งชี้จาก Subclass ไปยัง SuperClass หรือเรียกความสัมพันธ์เป็น “IS A” และมีการถ่ายทอด attribute และการดำเนินการผ่านจาก SuperClass มายัง SubClass ดังรูปที่ 2.10

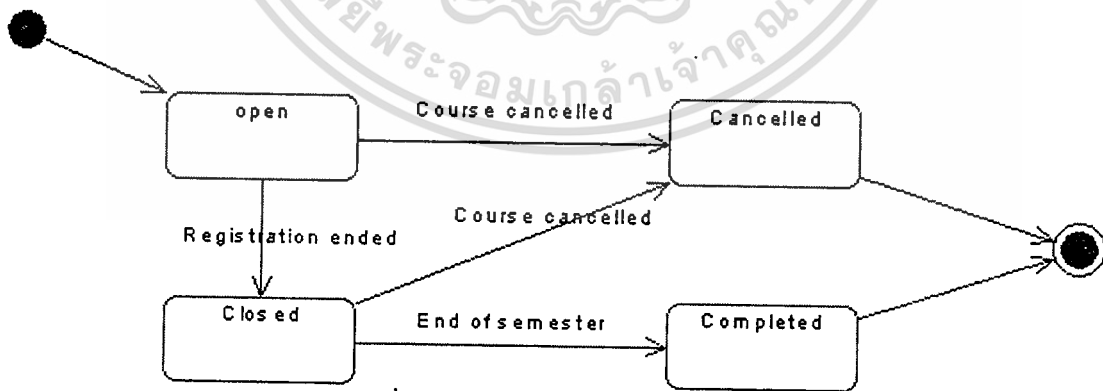


รูปที่ 2.10 แสดงคลาสไลอะแกรม ในความสัมพันธ์แบบ Generalization

2.4.5 State Transition Diagram

ใช้อธิบายวงจรชีวิต(life cycle) ของออบเจ็กต์จากช่วงเวลาหนึ่งที่ออบเจ็กต์นั้น ๆ ถูกสร้างขึ้นจนกระทั่งหมดหน้าที่ลง ซึ่งไม่จำเป็นต้องเขียนสเตทชาร์ทไลอะแกรมของทุกคลาส การเขียนไลอะแกรมนี้จะเขียนขึ้นเมื่อออบเจ็กต์นั้น ๆ มีหลายสถานะที่แตกต่างกันไป ออบเจ็กต์นั้นเปลี่ยนจากสถานะหนึ่งไปยังอีกสถานะหนึ่งได้อย่างไร มีเงื่อนไขใดบ้าง และพฤติกรรมของออบเจ็กต์ในแต่ละสถานะหนึ่ง ๆ แตกต่างกันอย่างใด ๆ ตัวอย่างของ State Transition diagram แสดงได้ดังรูปที่

2.11

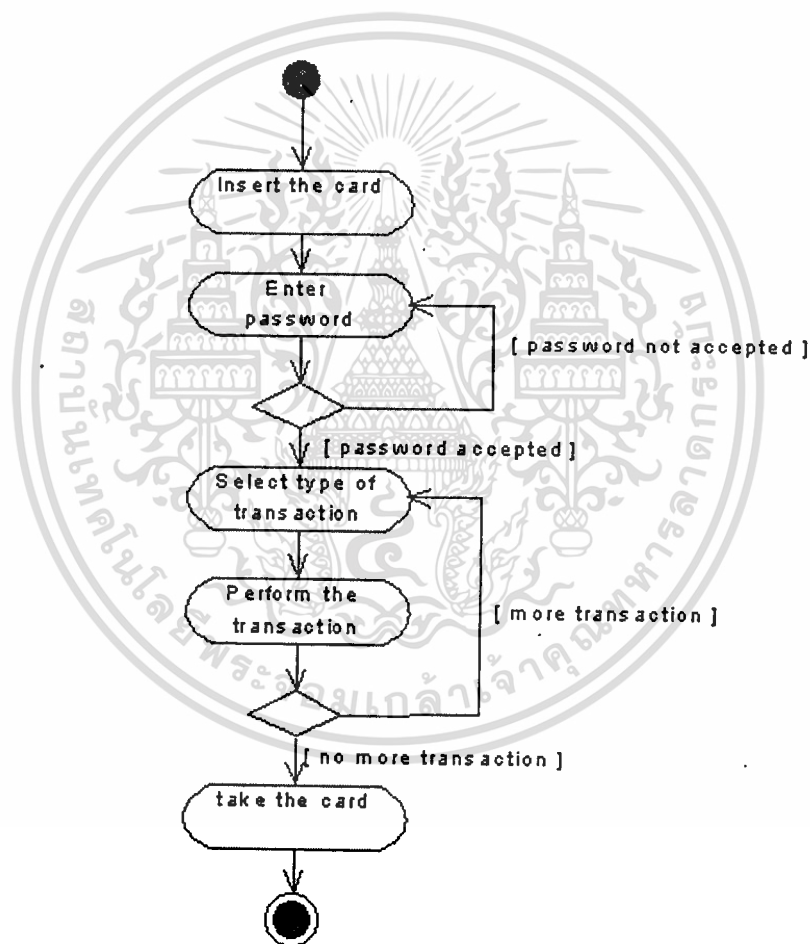


รูปที่ 2.11 แสดงสเตททรานสิชันไลอะแกรม

จากรูปที่ 2.11 คลาสคอร์สการเรียนการสอนอาจจะมีสถานะที่แตกต่างกันได้ คือ เปิด(Opened) ปิด(Closed) ยกเลิก (Cancelled) หรือจบคอร์สการเรียนการสอนแล้ว(Completed)

2.4.6 Activity Diagram

ใช้อธิบายกิจกรรม (activity) ต่าง ๆ ในการทำงานหนึ่ง ๆ มีลักษณะคล้ายกับการเขียน Flowchart และต่างกับ สดตทธานลีชันไคอะแกรมในส่วนที่สดตทธานลีชันไคอะแกรมจะพุดถึงสถานนะของออบเจ็กต์หนึ่ง ที่มีหลายสถานะแตกต่างกัน สัญลักษณ์ที่ใช้เหมือนกับสดตทธานลีชันไคอะแกรมแต่มีสัญลักษณ์ที่ใช้ในการตัดสินใจเพิ่มมา ตัวอย่างของ activity diagram แสดงได้ดังรูปที่ 2.12

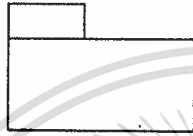


รูปที่ 2.12 แสดงตัวอย่างแอคตีวิตไคอะแกรม

2.4.7 Component Diagram

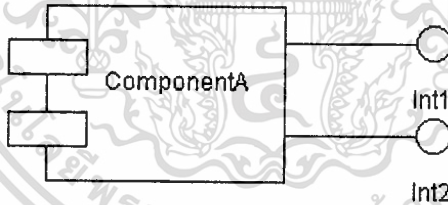
ใช้อธิบายองค์ประกอบทางกายภาพของการออกแบบระบบ แสดงทั้งการจัดการและการขึ้นต่อกันของ software component รวมทั้ง ส่วนของ source code ส่วนของ binary code และ ส่วนของ executable คอมโพเนนต์ที่โคแอมประกอบไปด้วย ส่วนต่าง ๆ ดังนี้คือ

- *Component Packages* ใช้แสดงถึงส่วนประกอบที่เกี่ยวข้องกันทางตรรกะ มีสัญลักษณ์ดังรูปที่ 2.13



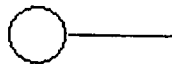
รูปที่ 2.13 แสดงสัญลักษณ์ Component Package

- *Components* ใช้แสดง software module กับ interface ซึ่งได้แก่ source code binary code executable DLL มีสัญลักษณ์ดังรูปที่ 2.14



รูปที่ 2.14 แสดงสัญลักษณ์ Components

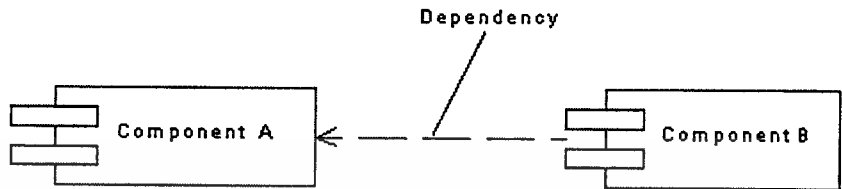
- *Interfaces* เป็นส่วนที่ใช้ระบุ operation ภายนอกของคลาส หรือของ component มีสัญลักษณ์ดังรูปที่ 2.15



Interface 1

รูปที่ 2.15 แสดงสัญลักษณ์ interfaces

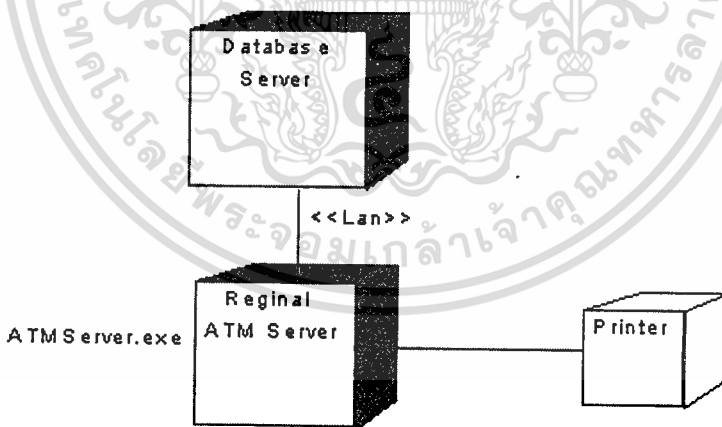
- *Dependency relationships* เป็นความสัมพันธ์ระหว่าง 2 ส่วนประกอบ คือเมื่อมีการเปลี่ยนแปลงที่ส่วนใด อีกส่วนหนึ่งก็ย่อมมีผลกระทบไปด้วย โดยมีสัญลักษณ์ คือเป็นเส้นประ ดังรูปที่ 2.16



รูปที่ 2.16 แสดงสัญลักษณ์ dependency relationship

2.4.8 Deployment Diagram

ใช้อธิบายโหนดการทำงานทั้งหมดบนเครือข่าย การเชื่อมต่อกันระหว่างโหนด ซึ่งได้แก่ โปรเซสเซอร์ อุปกรณ์ต่าง ๆ (devices) และซอฟต์แวร์ต่าง ๆ ไว้ด้วยกัน ซึ่งแสดงดังรูปที่ 2.17



รูปที่ 2.17 แสดงตัวอย่าง Deployment Diagram

จากรูปที่ 2.17 ก่อตั้งสี่เหลี่ยม ใช้แทน โหนดต่าง ๆ ของการเชื่อมต่อ โดยก่อกองที่มีการระบายสีจะหมายถึง โปรเซสเซอร์ ส่วนก่อกองที่ไม่ได้ระบายสีจะหมายถึงอุปกรณ์ต่าง ๆ (devices) และเส้นตรงที่เชื่อมกันระหว่างก่อกอง ใช้แทนการเชื่อมต่อกันของโหนด

บทที่ 3

การวิเคราะห์ระบบ

3.1 ความต้องการของระบบ

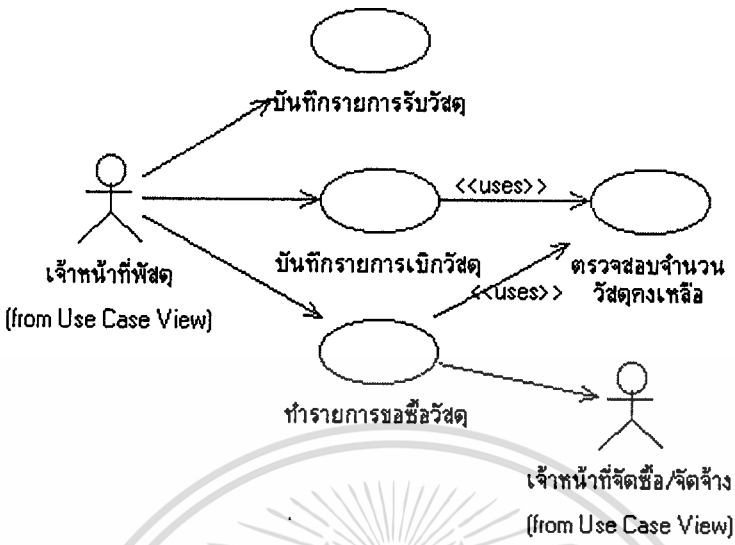
ระบบงานบริหารงานพัสดุของหน่วยงานราชการทั่วไป มีการควบคุมและจัดเก็บข้อมูลในรูปแบบเอกสารเป็นส่วนใหญ่ทำให้การค้นหาข้อมูลพัสดุทำได้ล่าช้า อีกทั้งในระบบยังประกอบไปด้วยงานหลาย ๆ ส่วนที่ต้องมีการประสานงาน ติดต่อกัน จึงทำให้เกิดความต้องการให้มีการพัฒนาระบบงานบริหารงานพัสดุนั้นเพื่อให้สามารถรองรับการทำงานในส่วนต่าง ๆ ให้เกิดประสิทธิภาพและเกิดความรวดเร็วในการทำงาน ระบบงานใหม่นี้สามารถรองรับการทำงานของระบบในส่วนต่าง ๆ ได้

3.2 Use Case Diagram

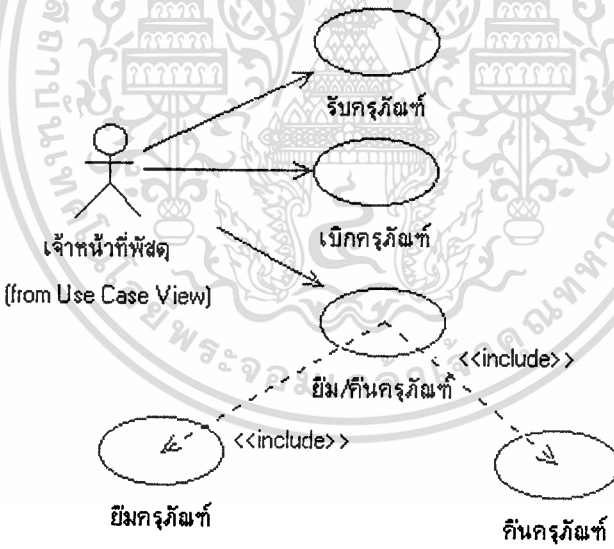
เราจะทำการสร้างโมเดลของระบบบริหารงานพัสดุ โดยใช้ Use Case Diagram ซึ่งจะทำให้การแบ่งออกเป็นงานย่อย ๆ ได้ดังนี้ คือ

- งานรับ-จ่ายวัสดุ
- งานรับ-จ่ายครุภัณฑ์
- งานบันทึกรายการซ่อมครุภัณฑ์
- งานจำหน่ายครุภัณฑ์
- งานตรวจพัสดุประจำปี
- งานขอขึ้นทะเบียนที่ราชพัสดุ
- งานกำหนดข้อมูลหลัก

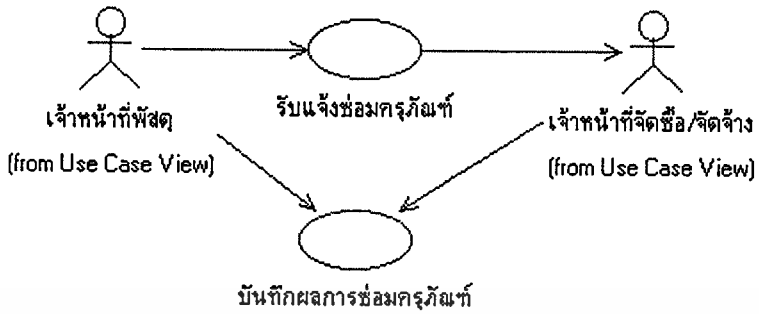
งานย่อย ๆ ดังกล่าวสามารถแสดงได้ดัง Use Case Diagram ต่าง ๆ ดังรูปที่ 3.1 ถึงรูปที่ 3.7



รูปที่ 3.1 Use Case Diagram ของ งานรับ-จ่ายวัสดุ



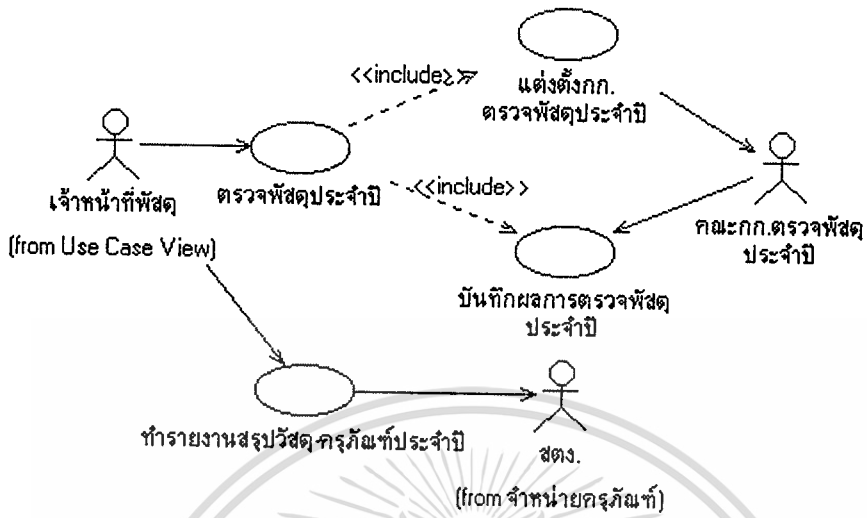
รูปที่ 3.2 Use Case Diagram ของ งานรับ-จ่ายครุภัณฑ์



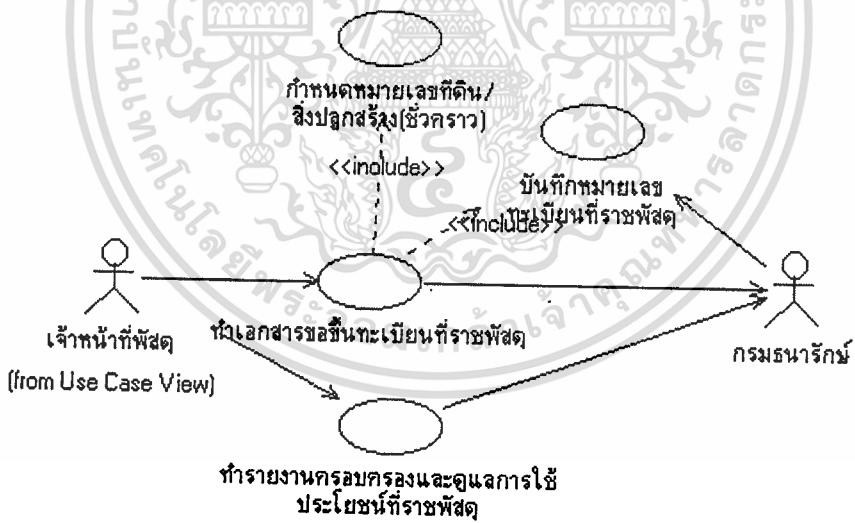
รูปที่ 3.3 Use Case Diagram ของ งานบันทึกการซ่อมครุภัณฑ์



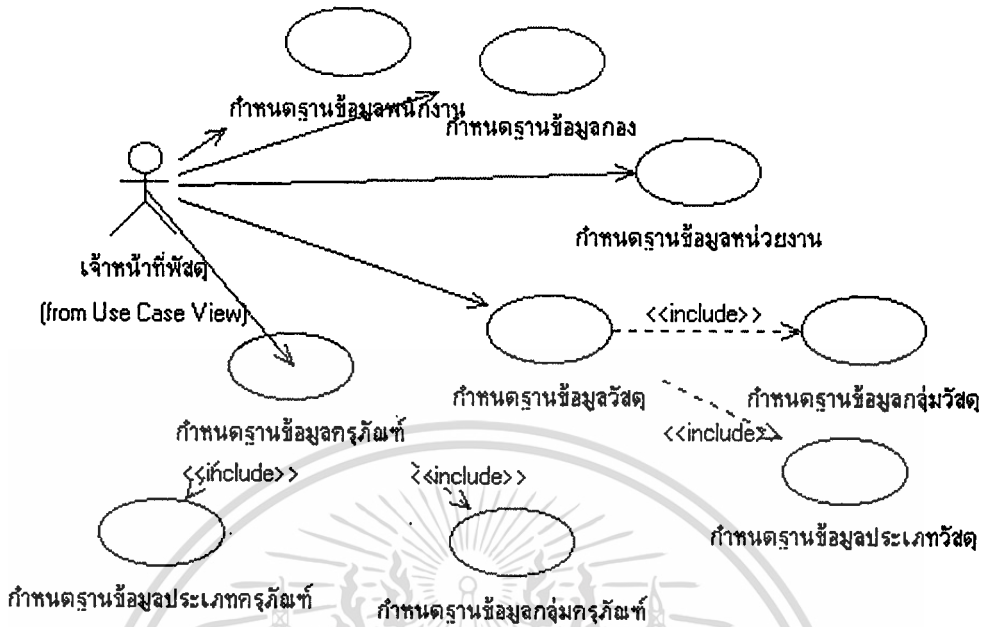
รูปที่ 3.4 Use Case Diagram ของ งานจำหน่ายครุภัณฑ์



รูปที่ 3.5 Use Case Diagram ของ งานตรวจพัสดุประจำปี



รูปที่ 3.6 Use Case Diagram ของ งานขอขึ้นทะเบียนที่ราชพัสดุ



รูปที่ 3.7 Use Case Diagram ของงานกำหนดข้อมูลหลัก

3.3 Use Case Description

จาก Use Case Diagram ที่เราวิเคราะห์ได้จากการทำงานในระบบงานเราสามารถอธิบายรายละเอียดขั้นตอนการทำงาน ได้ดังนี้

3.3.1 งานรับ-จ่ายวัสดุ

3.3.1.1 Use case รับวัสดุ

Actor : เจ้าหน้าที่วัสดุ

Description : ใช้ในการบันทึกข้อมูลรายการรับวัสดุ ลงทะเบียนวัสดุ

Basic Flow :

1. เริ่มเมื่อ เจ้าหน้าที่พัสดุ ทำการ Login เข้าสู่ระบบ และเลือกรายการรับวัสดุเพื่อบันทึก รายการรับวัสดุ
2. ระบบจะแสดงเลขที่รับวัสดุให้อัตโนมัติ (โดยจะ running จากเลขที่รับวัสดุล่าสุด) และ ข้อมูลที่ระบบให้ทำการบันทึกประกอบด้วยข้อมูล
 - รายการวัสดุ ซึ่งต้องเป็นรายการที่ได้กำหนดหมายเลขวัสดุแล้ว
 - วันที่รับวัสดุ

- จำนวนวัสดุที่รับเข้า
 - ราคาของวัสดุ
3. เมื่อเจ้าหน้าที่บันทึกข้อมูลรายการรับวัสดุ ระบบจะตรวจสอบความถูกต้องของข้อมูล และเมื่อข้อมูลถูกต้องระบบจะบันทึกข้อมูลนั้นลงในฐานข้อมูล ทะเบียนวัสดุ
 4. เจ้าหน้าที่ที่สามารถตรวจสอบความถูกต้อง โดยสามารถพิมพ์รายงานบัญชีวัสดุตามรหัส วัสดุ ซึ่งจะเป็นการแสดงรายละเอียดการรับวัสดุแต่ละรายการ

3.3.1.2 Use case เบิกวัสดุ

Actor : เจ้าหน้าที่วัสดุ

Description : ใช้บันทึกข้อมูลรายการเบิกวัสดุออกจากทะเบียน

Basic Flow :

1. เริ่มเมื่อ เจ้าหน้าที่พัสดุ ทำการ Login เข้าสู่ระบบ และเลือกรายการเบิกวัสดุเพื่อบันทึก รายการเบิกวัสดุ
2. ระบบจะแสดงเลขที่จ่ายวัสดุให้อัตโนมัติ (โดยจะ running จากเลขที่จ่ายวัสดุล่าสุด) และบันทึกข้อมูลตามใบเบิกวัสดุ ซึ่งประกอบด้วยข้อมูล
 - รายการวัสดุที่ขอเบิก
 - หน่วยงานที่ขอเบิก
 - รายการวัสดุที่ขอเบิก
 - จำนวนเบิก
 - จำนวนวัสดุคงเหลือ
3. เมื่อเจ้าหน้าที่บันทึกข้อมูลรายการเบิก ระบบจะแสดงยอดจำนวนวัสดุคงเหลือแต่ละ รายการ และทำการตรวจสอบรายการเบิก
4. รายการใดที่มีจำนวนเบิกมากกว่าจำนวนวัสดุคงเหลือ ระบบก็จะไม่สามารถให้เบิกได้ เจ้าหน้าที่สามารถแก้ไขในส่วนของ จำนวนวัสดุที่จ่ายได้ และเมื่อตรวจสอบข้อมูลถูกต้องระบบจะทำการบันทึกข้อมูลจ่ายวัสดุและปรับปรุงจำนวนวัสดุเหลือลงฐานข้อมูล ทะเบียนวัสดุ
5. เจ้าหน้าที่สามารถตรวจสอบความถูกต้อง โดยสามารถพิมพ์รายงานต่าง ๆ ได้ คือ
 - รายงานการเบิกวัสดุแยกตามหน่วยงาน/กองที่ขอเบิก

3.3.1.3 Use case ทำรายการขอซื้อวัสดุ

Actor : เจ้าหน้าที่วัสดุ, เจ้าหน้าที่จัดซื้อ/จัดจ้าง

Description : ใช้สำหรับให้เจ้าหน้าที่นำไปเป็นข้อมูลในการจัดทำรายการขอซื้อวัสดุ

Basic Flow :

1. เจ้าหน้าที่เลือกรายการจัดทำรายการขอซื้อวัสดุ
2. ระบบจะแสดงรายการวัสดุที่ควรสั่งซื้อ โดยจะตรวจสอบข้อมูลวัสดุที่มีจำนวนวัสดุคงเหลือน้อยกว่าจำนวนวัสดุอย่างต่ำ (ซึ่งได้กำหนดไว้ในขั้นตอนกำหนดหมายเลขวัสดุ) จากฐานข้อมูลทะเบียนวัสดุ
3. เจ้าหน้าที่สามารถเลือกพิมพ์รายการดังกล่าวออกทางเครื่องพิมพ์ เพื่อนำไปใช้ในการพิจารณาจัดทำรายงานการขอซื้อวัสดุ ต่อไป

3.3.1.4 Use case ตรวจสอบวัสดุคงเหลือ

Actor :

Description : ใช้สำหรับให้เจ้าหน้าที่ตรวจสอบจำนวนวัสดุคงเหลือ

Basic Flow :

1. เลือกรายการตรวจสอบวัสดุคงเหลือ เพื่อต้องการตรวจสอบจำนวนวัสดุคงเหลือ
2. ระบบจะมีรายการย่อยให้เลือก คือ
 - แสดงรายการจำนวนวัสดุคงเหลือทางหน้าจอ
 - แสดงรายการจำนวนวัสดุคงเหลือทางรายงาน
3. เมื่อเจ้าหน้าที่เลือกรายการ ระบบจะแสดงจำนวนวัสดุคงเหลือแต่ละรายการ โดยจะดึงข้อมูลจำนวนวัสดุคงเหลือจากฐานข้อมูลทะเบียนวัสดุ

3.3.2 งานรับ-จ่ายครุภัณฑ์

3.3.2.1 Use Case รับครุภัณฑ์

Actor : เจ้าหน้าที่วัสดุ

Description : ใช้ในการบันทึกข้อมูล รายการรับครุภัณฑ์ลงทะเบียนครุภัณฑ์

Basic Flow :

1. เลือกรายการรับครุภัณฑ์เพื่อบันทึกรายการรับครุภัณฑ์
2. ระบบจะแสดงเลขที่รับครุภัณฑ์ให้อัตโนมัติ (โดยจะ running จากเลขที่รับครุภัณฑ์ล่าสุด) และบันทึกข้อมูลตามเอกสารรับครุภัณฑ์ ซึ่งประกอบด้วยข้อมูล

- รายการครุภัณฑ์ ซึ่งต้องเป็นรายการที่ได้กำหนดหมายเลขครุภัณฑ์แล้ว
 - วันที่รับ
 - ราคาต่อหน่วย
 - ปีงบประมาณ
 - ประจําหน่วยงาน
 - วิธีการได้มา ระบบจะแสดงเป็นรายการต่าง ๆ ให้สามารถเลือกได้ ซึ่งได้แก่ วิธีตกลงราคา วิธีสอบราคา วิธีประกวดราคา วิธีพิเศษ วิธีกรณีพิเศษ รับบริจาค
 - ชื่อมาจาก
 - สถานภาพครุภัณฑ์ ระบบจะแสดงเป็นรายการต่าง ๆ ให้สามารถเลือกได้ ซึ่งได้แก่ สถานภาพ ปกติ ยืม ช่อม ชำรุด หาย ส่งคืน โอน จำหน่ายเป็นสูญ ขาย แปรสภาพ บริจาค
 - ชื่อผู้เบิก
 - รับจากใบส่งของ/ใบส่งชื่อเลขที่
3. เมื่อเจ้าหน้าที่บันทึกข้อมูลรายการรับครุภัณฑ์ ระบบจะตรวจสอบความถูกต้องของข้อมูลและเมื่อข้อมูลถูกต้องระบบจะบันทึกข้อมูลนั้นลงในฐานข้อมูลทะเบียนครุภัณฑ์
 4. เจ้าหน้าที่สามารถตรวจสอบความถูกต้อง โดยสามารถพิมพ์รายงานต่าง ๆ ได้คือ
 - รายงานบัญชีครุภัณฑ์ ตามกลุ่ม/ประเภท หรือรหัสครุภัณฑ์
 - รายงานครุภัณฑ์ ประจำกอง/หน่วยงาน

3.3.2.2 Use case เบิกครุภัณฑ์

Actor : เจ้าหน้าที่วัสดุ

Description : ใช้บันทึกข้อมูลรายการเบิกครุภัณฑ์ออกจากทะเบียน

Basic Flow :

1. เลือกรายการเบิก-จ่ายวัสดุ/ครุภัณฑ์
2. เลือกรายการเบิกครุภัณฑ์เพื่อบันทึกรายการจ่ายครุภัณฑ์ออกจากทะเบียน
3. ระบบจะแสดงเลขที่จ่ายครุภัณฑ์ให้อัตโนมัติ (โดยจะ running จากเลขที่จ่ายครุภัณฑ์ล่าสุด) และบันทึกข้อมูลตามใบเบิกครุภัณฑ์ ซึ่งประกอบด้วยข้อมูล
 - หน่วยงานที่ขอเบิก
 - รายการครุภัณฑ์ที่ขอเบิก

- จำนวนเบิก
4. ระบบจะตรวจสอบข้อมูลรายการเบิก เมื่อสามารถเบิก-จ่ายครุภัณฑ์ได้ตามรายการเบิก ระบบจะทำการบันทึกข้อมูลเบิกครุภัณฑ์และปรับปรุงลงฐานข้อมูลทะเบียนครุภัณฑ์ในแต่ละรายการที่มีการเบิก
 5. เจ้าหน้าที่สามารถตรวจสอบความถูกต้อง โดยสามารถพิมพ์รายงานต่าง ๆ ได้ คือ
 - รายงานการเบิกครุภัณฑ์แยกตามหน่วยงาน/กอง

3.3.2.3 Use case ยืม/คืนครุภัณฑ์

Actor : เจ้าหน้าที่วัสดุ

Description : ใช้บันทึกข้อมูลรายการยืม-คืนครุภัณฑ์

Basic Flow :

1. เลือกรายการยืม-คืนครุภัณฑ์
2. ระบบจะแสดงเลขที่ยืมครุภัณฑ์ให้อัตโนมัติ (โดยจะ running จากเลขที่ยืมครุภัณฑ์ ล่าสุด) และบันทึกข้อมูลรายการยืมตามใบยืมครุภัณฑ์ ซึ่งประกอบด้วยข้อมูล
 - หน่วยงานที่ขอยืม
 - รายการครุภัณฑ์ที่ขอยืม
3. เมื่อเจ้าหน้าที่บันทึกข้อมูลรายการยืมครุภัณฑ์ ระบบจะตรวจสอบข้อมูลจากฐานข้อมูลทะเบียนครุภัณฑ์ โดยแสดงรายการดังกล่าวที่สามารถให้ยืมได้
4. ถ้าไม่พบรายการครุภัณฑ์ที่ต้องการ ระบบจะแสดงข้อความ “ไม่พบรายการที่ต้องการยืม”
5. ถ้าพบรายการครุภัณฑ์ที่สามารถให้ยืมได้ ระบบจะบันทึกลงฐานข้อมูลรายการยืม ครุภัณฑ์ และทำการปรับปรุงฐานข้อมูลทะเบียนครุภัณฑ์
6. เมื่อเจ้าหน้าที่ได้รับรายการคืนครุภัณฑ์ ให้เจ้าหน้าที่ระบุเลขที่รายการยืมครุภัณฑ์รายการนั้น ๆ ระบบก็จะแสดงรายการยืมที่ได้บันทึกไว้มาแสดง
7. เจ้าหน้าที่ทำการบันทึกครุภัณฑ์ที่คืน เมื่อข้อมูลถูกต้องจะทำการปรับปรุงฐานข้อมูลรายการยืม และปรับปรุงฐานข้อมูลทะเบียนครุภัณฑ์

3.3.3 งานบันทึกรายการซ่อมครุภัณฑ์

3.3.3.1 Use case รับแจ้งซ่อมครุภัณฑ์

Actor : เจ้าหน้าที่วัสดุ

Description : ใช้ในการบันทึกข้อมูลรายการแจ้งซ่อมครุภัณฑ์

Basic Flow :

การทำงาน :

1. เมื่อเจ้าหน้าที่ได้รับการแจ้งซ่อมอาจจะทางโทรศัพท์หรือจากใบแจ้งซ่อม ให้เลือกรายการรับแจ้งซ่อมครุภัณฑ์
2. ระบบจะแสดงเลขที่แจ้งซ่อมให้อัตโนมัติ (โดยจะ running จากเลขที่แจ้งซ่อมล่าสุด) และให้บันทึกรายละเอียดการแจ้งซ่อม ซึ่งประกอบด้วย
 - วันที่แจ้งซ่อม ซึ่งระบบจะแสดงวันที่ปัจจุบันให้
 - หน่วยงานที่แจ้งซ่อม
 - รายการครุภัณฑ์ที่ขอซ่อม
 - รายละเอียดการแจ้งซ่อม
3. เมื่อข้อมูลถูกต้อง ระบบจะทำการปรับปรุงข้อมูลลงฐานข้อมูลแจ้งซ่อมครุภัณฑ์
4. เจ้าหน้าที่เลือกรายการพิมพ์ รายงานการแจ้งซ่อม ส่งให้กับเจ้าหน้าที่จัดซื้อ/จัดจ้าง

3.3.3.2 Use case บันทึกรายการซ่อมครุภัณฑ์

Actor : เจ้าหน้าที่วัสดุ, เจ้าหน้าที่จัดซื้อ/จัดจ้าง

Description : ใช้ในการบันทึกข้อมูลรายละเอียดการซ่อมครุภัณฑ์

Basic Flow :

1. เมื่อดำเนินการซ่อมเสร็จแล้ว ให้เจ้าหน้าที่บันทึกประวัติการซ่อม ตามเอกสารใบสั่งจ้างซ่อมครุภัณฑ์
2. เจ้าหน้าที่บันทึกประวัติการซ่อม ซึ่งประกอบด้วย
 - เลขที่ใบสั่งซื้อ/ใบสั่งจ้าง
 - วันที่สั่งซื้อ/สั่งซ่อม
 - รายการครุภัณฑ์ที่ซ่อม
 - จำนวนเงินในการซ่อม
 - เลขที่ใบแจ้งซ่อม

3. เมื่อข้อมูลถูกต้อง ระบบจะทำการปรับปรุงข้อมูลลงฐานข้อมูลประวัติการซ่อมครุภัณฑ์

4. เจ้าหน้าที่สามารถเลือกรายการพิมพ์รายงาน ต่าง ๆ
 - ประวัติการซ่อมครุภัณฑ์ โดยระบุรายการครุภัณฑ์ที่ต้องการพิมพ์ ซึ่งระบบจะดึงข้อมูลจากฐานข้อมูลประวัติการซ่อมครุภัณฑ์
 - รายงานสรุปค่าใช้จ่ายในการซ่อมครุภัณฑ์ ตามปีงบประมาณ
5. ระบบจะมีหน้าจอให้เจ้าหน้าที่สามารถสอบถาม ราคาในซ่อมครั้งล่าสุด เพื่อใช้ในการพิจารณาการซ่อมครั้งต่อไป โดยจะต้องระบุเลขที่ใบแจ้งซ่อม

3.3.4 งานจำหน่ายครุภัณฑ์

3.3.4.1 Use Case รับเรื่องขอจำหน่ายครุภัณฑ์

Actor : เจ้าหน้าที่วัสดุ

Description : ทำการบันทึกรายการขอจำหน่ายครุภัณฑ์จากหน่วยงานต่าง ๆ

Basic Flow :

1. เมื่อเจ้าหน้าที่เลือกรายการขอจำหน่ายครุภัณฑ์ ระบบจะแสดงเลขที่ขอจำหน่ายให้อัตโนมัติ (โดยจะ running จากเลขที่ขอจำหน่ายล่าสุด)
2. ให้เจ้าหน้าที่บันทึกรายละเอียด
 - วันที่ขอจำหน่าย
 - วิธีการจำหน่าย
 - รายชื่อคณะกรรมการที่รับผิดชอบเรื่องจำหน่ายครุภัณฑ์
3. เลือกรายการครุภัณฑ์ ที่ขอจำหน่าย
4. เมื่อข้อมูลถูกต้อง ระบบจะทำการปรับปรุงข้อมูลลงฐานข้อมูลขอจำหน่ายครุภัณฑ์
5. พิมพ์รายการขอจำหน่ายเพื่อเสนอต่อหัวหน้าส่วนราชการต่อไป

3.3.4.2 Use Case บันทึกผลการจำหน่ายครุภัณฑ์

Actor : เจ้าหน้าที่วัสดุ, คณะกรรมการดำเนินการจำหน่ายครุภัณฑ์

Description : ทำการบันทึกผลรายการจำหน่ายครุภัณฑ์

Basic Flow :

1. เจ้าหน้าที่เลือกรายการบันทึกผลการจำหน่ายครุภัณฑ์
2. ให้เจ้าหน้าที่เลือกเลขที่ขอจำหน่ายหน้า ระบบจะแสดงเฉพาะรายการขอจำหน่ายครุภัณฑ์ที่ยังไม่ได้รับผลมาแสดง
3. ให้เจ้าหน้าที่บันทึกผลการจำหน่ายครุภัณฑ์

3.3.5 งานตรวจสอบพัสดุประจำปี

3.3.5.1 Use Case บันทึกผลการตรวจพัสดุประจำปี

Actor : เจ้าหน้าที่วัสดุ

Description : ทำการปรับปรุงฐานข้อมูลทะเบียนวัสดุ-ครุภัณฑ์ เมื่อมีการตรวจสอบพัสดุประจำปี

Basic Flow :

1. เจ้าหน้าที่พัสดุเลือกรายการพิมพ์รายงานต่าง ๆ เช่น
 - รายงานแบบสำรวจครุภัณฑ์แยกตามกอง/หน่วยงาน นำส่งให้กับคณะกรรมการตรวจสอบพัสดุประจำปีของแต่ละกอง/หน่วย ตรวจสอบว่ารายการครุภัณฑ์ที่อยู่ในทะเบียนครุภัณฑ์กับรายการที่ประจำอยู่ตามกอง/หน่วยงานถูกต้องตรงกัน หรือมีรายการใดเปลี่ยนแปลงไป มีสภาพปกติเช่นเดิม หรือมีสภาพชำรุดไม่สมควรใช้อีกต่อไป
 - รายงานแบบสำรวจวัสดุ นำส่งให้กับคณะกรรมการตรวจสอบพัสดุประจำปีตรวจสอบว่าจำนวนวัสดุคงเหลือในทะเบียนวัสดุ กับ จำนวนที่เจ้าหน้าที่พัสดुकควบคุมอยู่มีจำนวน ตรงกันหรือไม่
2. หลังจากได้รับผลการตรวจสอบจากคณะกรรมการตรวจสอบพัสดุประจำปี หากมีรายการครุภัณฑ์ชำรุดให้เจ้าหน้าที่เลือกรายการรับครุภัณฑ์เพื่อทำการปรับปรุงสถานภาพครุภัณฑ์
3. เมื่อเจ้าหน้าที่ระบุเลขที่รับครุภัณฑ์ ระบบจะแสดงรายละเอียดทั้งหมดของครุภัณฑ์รายการดังกล่าว
4. ให้เจ้าหน้าที่บันทึกสถานภาพครุภัณฑ์ ซึ่งระบบจะแสดงรายการให้เลือก โดยเลือกรายการชำรุด
5. เมื่อข้อมูลถูกต้อง ระบบจะทำการปรับปรุงข้อมูลลงฐานข้อมูลทะเบียน ครุภัณฑ์

3.3.6 งานขอขึ้นทะเบียนที่ราชพัสดุ

3.3.6.1 Use Case ทำเอกสารขอขึ้นทะเบียนที่ราชพัสดุ

Actor : เจ้าหน้าที่พัสดุ

Description : ระบบจะบันทึกรายการขอขึ้นทะเบียนที่ราชพัสดุ ซึ่งประกอบด้วย 2 ส่วน คือ ที่ดิน/สิ่งปลูกสร้าง

Basic Flow :

1. เลือกรายการรับที่ราชพัสดุเพื่อบันทึกรายการรับที่ราชพัสดุ
2. ระบบจะแสดงเลขที่รับที่ดินราชพัสดุให้อัตโนมัติ (โดยจะ running จากเลขที่รับที่ราชพัสดุล่าสุด) และบันทึกข้อมูลตามเอกสารรับที่ราชพัสดุ ซึ่งประกอบด้วยข้อมูล 2 ส่วน คือ

ส่วนที่ 1 : ข้อมูลที่ดิน ซึ่งประกอบด้วยข้อมูลของ

- หมายเลขทะเบียนที่ดิน (บันทึกตอนได้รับการขึ้นทะเบียนจากกรมธนารักษ์แล้ว)
- เนื้อที่ ซึ่งก็จะให้ป้อนข้อมูลของ จำนวนไร่ งาน ตารางวา
- สถานที่ตั้ง ซึ่งก็จะให้ป้อนข้อมูลของ เลขที่ตั้ง ตำบล อำเภอ จังหวัด
- สถานที่ใกล้เคียง ซึ่งก็จะให้ป้อนข้อมูลของ แต่ละทิศจดสถานที่ใดบ้าง
- ราคาที่ดิน
- การได้มา
- การใช้ประโยชน์

ส่วนที่ 2 : ข้อมูลสิ่งปลูกสร้าง ซึ่งประกอบด้วยข้อมูลของ

- หมายเลขทะเบียนสิ่งปลูกสร้าง (บันทึกตอนได้รับการขึ้นทะเบียนจากกรมธนารักษ์แล้ว)
- ลักษณะรูปทรง
- ราคาสิ่งปลูกสร้าง
- การได้มา
- การใช้ประโยชน์

3. เมื่อเจ้าหน้าที่บันทึกข้อมูลรายการรับที่ราชพัสดุ ระบบจะตรวจสอบความถูกต้องของข้อมูลและเมื่อข้อมูลถูกต้องระบบจะบันทึกข้อมูลนั้นลงในฐานข้อมูลทะเบียนที่ราชพัสดุ

4. เมื่อเจ้าหน้าที่ต้องนำรายการรับที่ราชพัสดุขึ้นทะเบียนต่อกรมธนารักษ์ ให้เลือกรายการพิมพ์รายงาน รายการรับ-ส่งที่ราชพัสดุขึ้นทะเบียน โดยระบบจะไปดึงข้อมูลเฉพาะข้อมูลที่ยังไม่มีการนำขึ้นทะเบียนต่อกรมธนารักษ์จากฐานข้อมูลทะเบียนที่ราชพัสดุ เมื่อเจ้าหน้าที่ได้รับหมายเลขทะเบียนที่ดิน/สิ่งปลูกสร้าง จากการรับขึ้นทะเบียนที่ราชพัสดุของกรมธนารักษ์ ให้เจ้าหน้าที่เรียกเลขที่รับของรายการดังกล่าว ระบบจะแสดงข้อมูลทั้งหมดขึ้นมา เจ้าหน้าที่ทำการบันทึกข้อมูลในส่วนของหมายเลขที่ดิน/ สิ่งปลูกสร้าง เมื่อข้อมูลถูกต้อง ระบบจะปรับปรุงข้อมูลลงในฐานข้อมูลทะเบียนที่ราชพัสดุ

3.3 Analysis Class

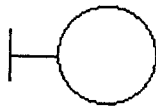
หลังจากที่ได้ภาพรวมการทำงานของระบบจาก Use Case Diagram แล้ว ขั้นตอนต่อไปคือทำการหา Class ของระบบ ซึ่งเราจะสามารถจำแนก Class ออกเป็น 3 ประเภท คือ

- Boundary Class
- Entity Class
- Control Class

3.3.1 Boundary Class

Boundary Class เป็นส่วนของระบบงานกับผู้ใช้ ซึ่ง ได้แก่

- User interface Classes คือ Class ซึ่งใช้ติดต่อกับผู้ใช้ระบบ
- System interface Classes คือ Class ซึ่งใช้ติดต่อกับระบบภายนอก
- Device interface Classes คือ Class ซึ่งใช้เชื่อมโยงกับอุปกรณ์ภายนอก
- มีสัญลักษณ์ ดังรูป



Boundary Class

รูปที่ 3.8 Boundary Class

3.3.2 Entity Class

- Entity Class เป็นส่วนของโครงสร้างข้อมูลที่จะทำการจัดเก็บในระบบซึ่ง
- มีสัญลักษณ์ ดังรูป



Entity Class

รูปที่ 3.9 Entity Class

3.3.3 Control Class

- Control Class ใช้ในการควบคุมกระบวนการทำงาน โดยจะมี หนึ่ง Control Class ในหนึ่ง Use Case
- มีสัญลักษณ์ ดังรูป



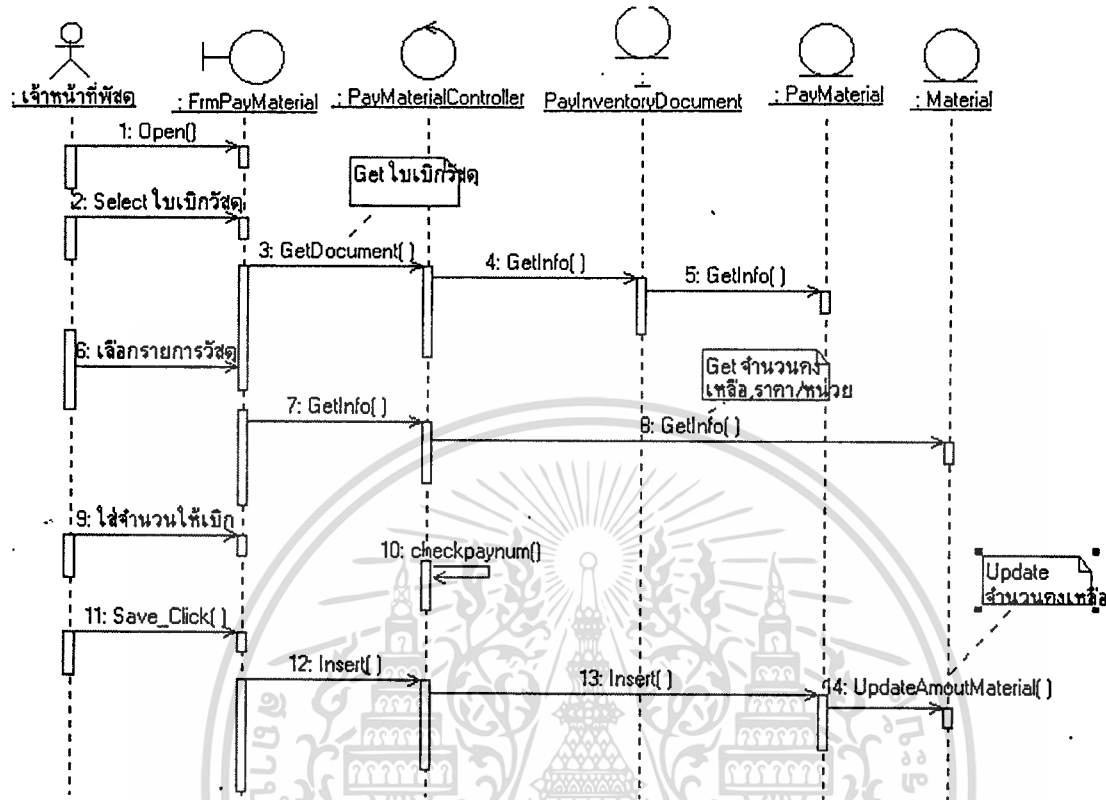
Control Class

รูปที่ 3.10 Control Class

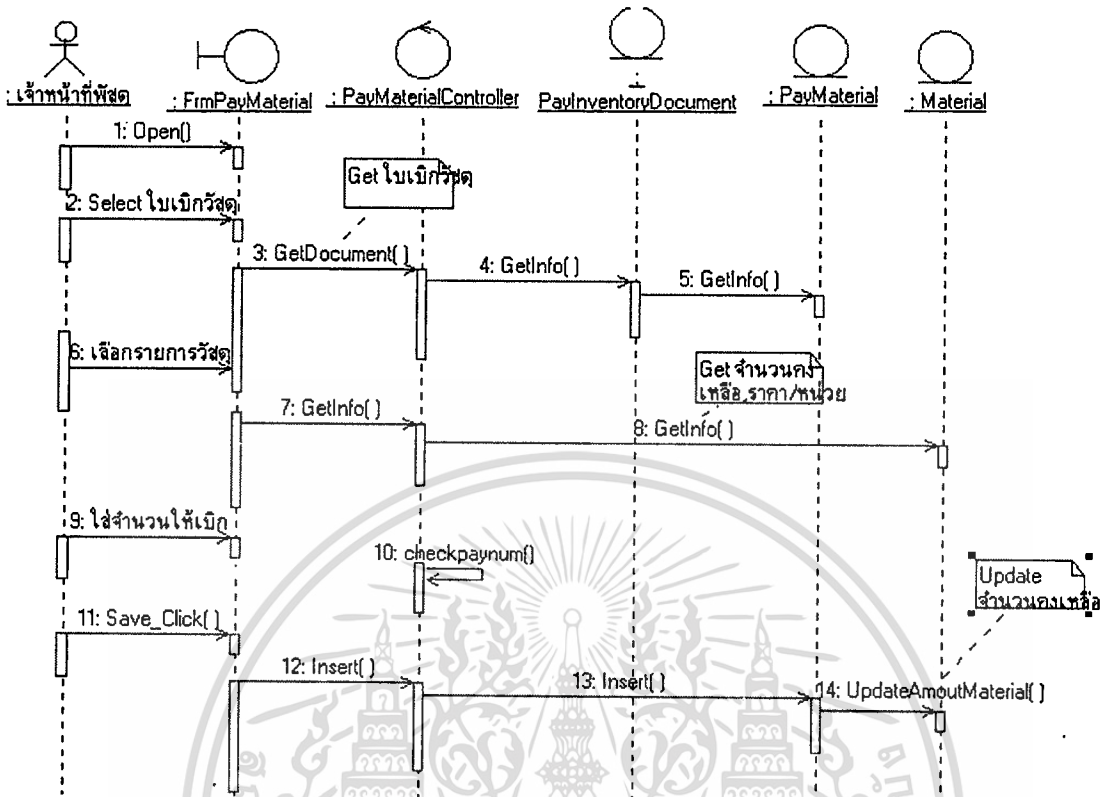
3.4 Analysis Sequence Diagram

เป็นขั้นตอนในการแสดงความสัมพันธ์กันระหว่าง Class โดยจะทำการสร้าง Sequence

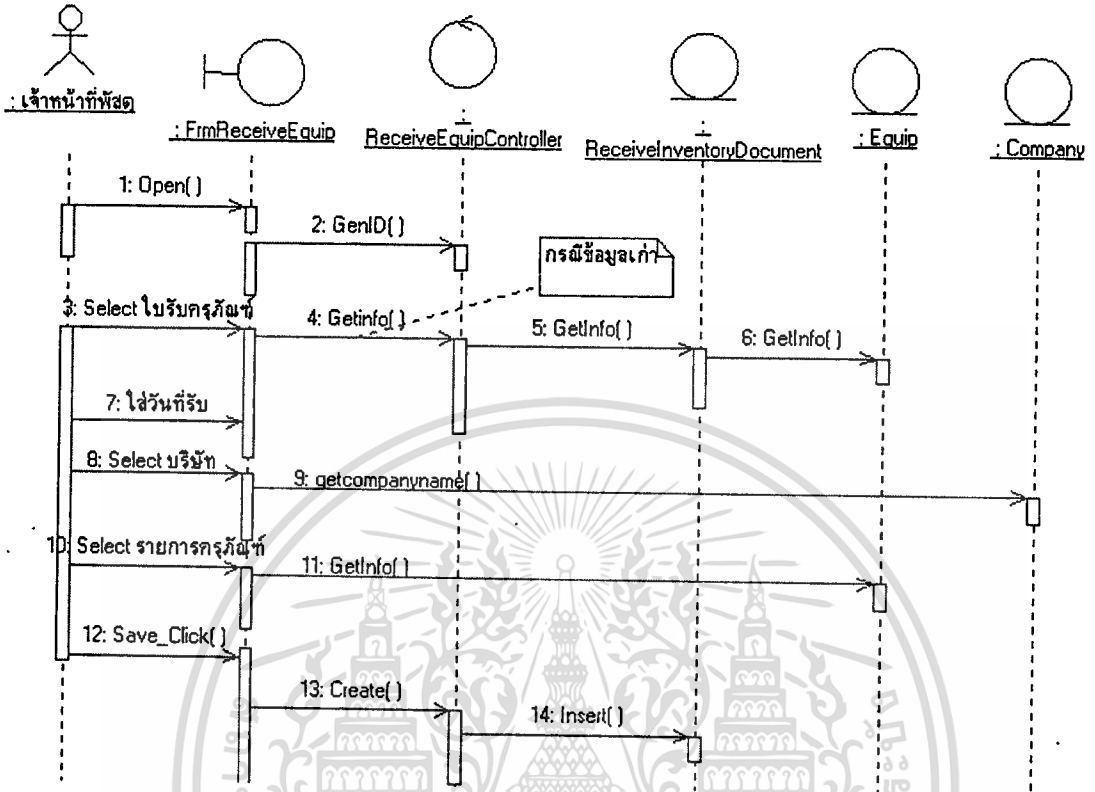
Diagram –ของแต่ละ Use Case ดังรูปที่3.10-3.16



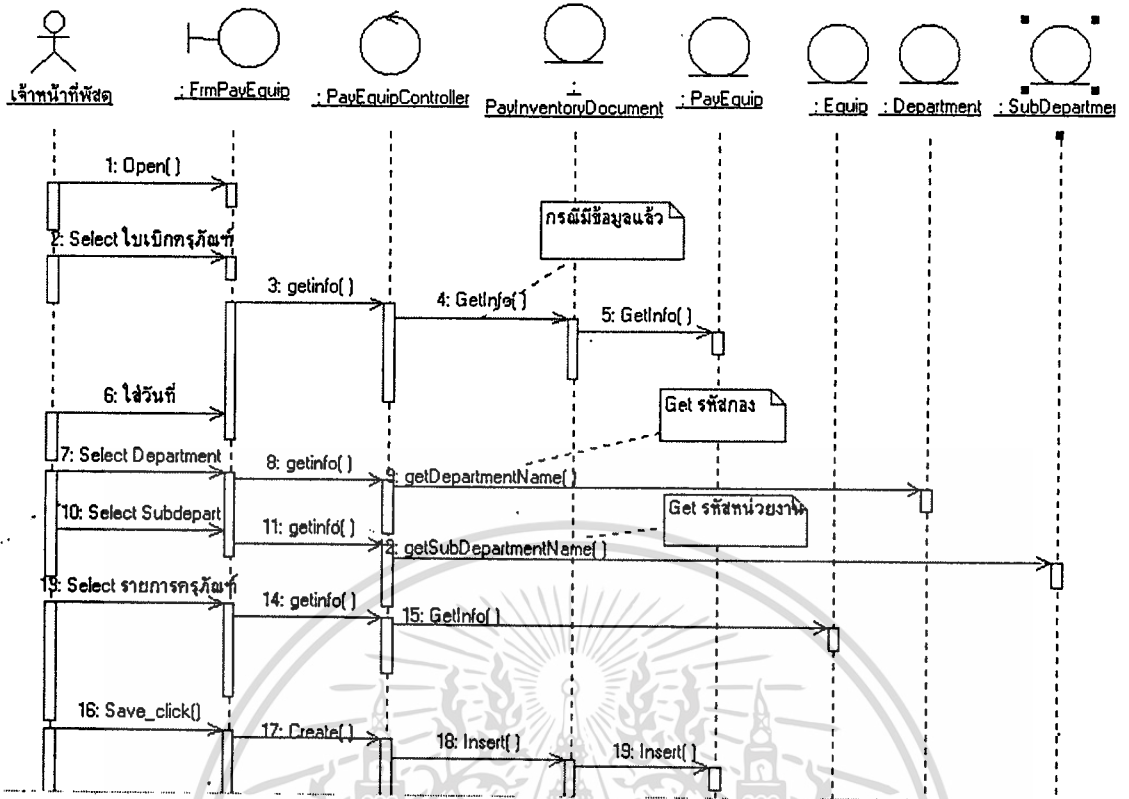
รูปที่ 3.10 Sequence Diagram ของ Use Case รับวัสดุ



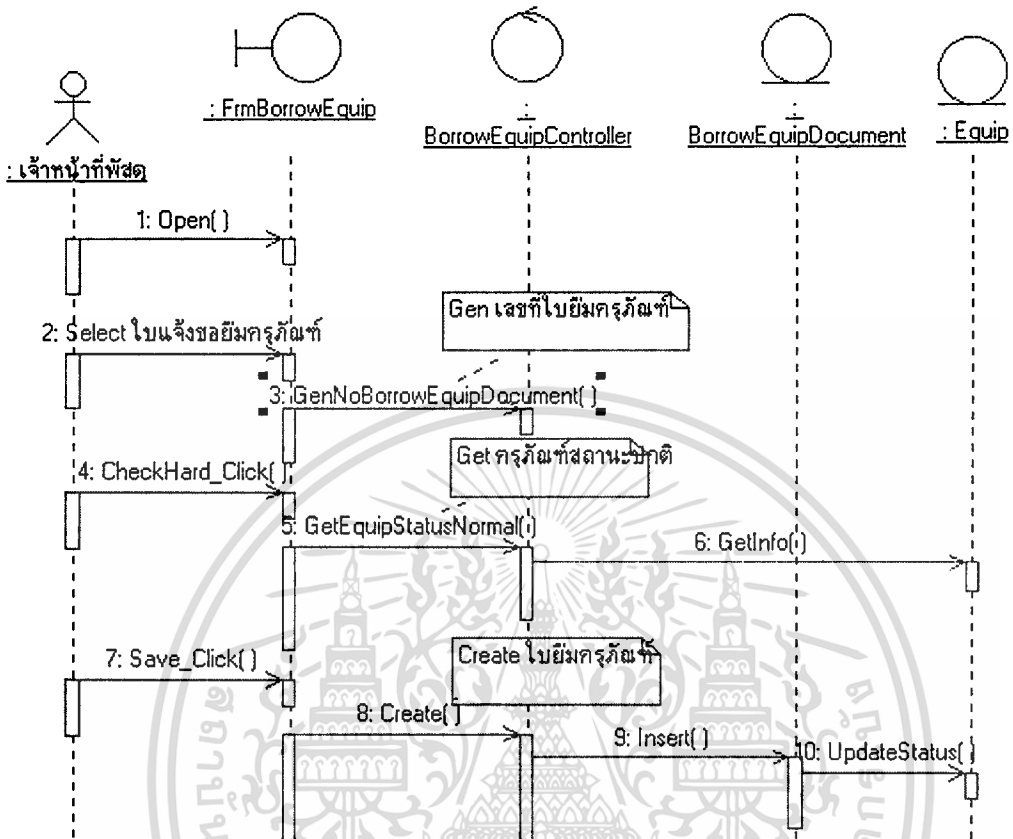
รูปที่ 3.11 Sequence Diagram ของ Use Case เบิกวัสดุ



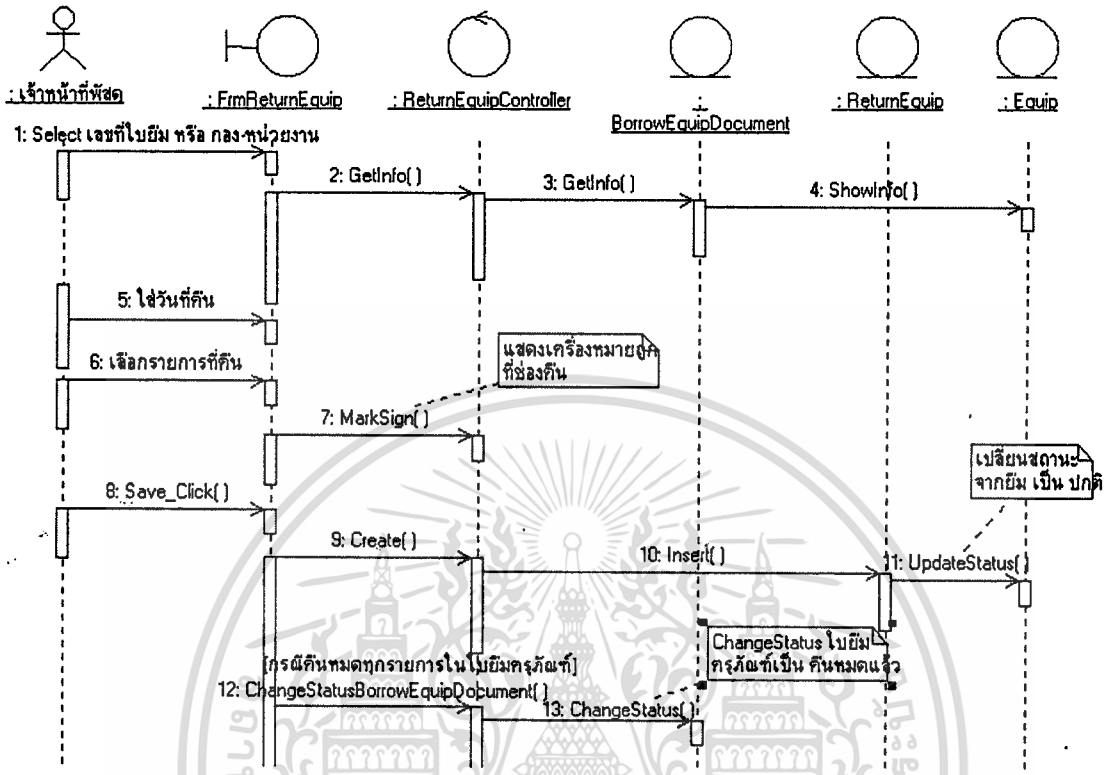
รูปที่ 3.12 Sequence Diagram ของ Use Case รับครุภัณฑ์



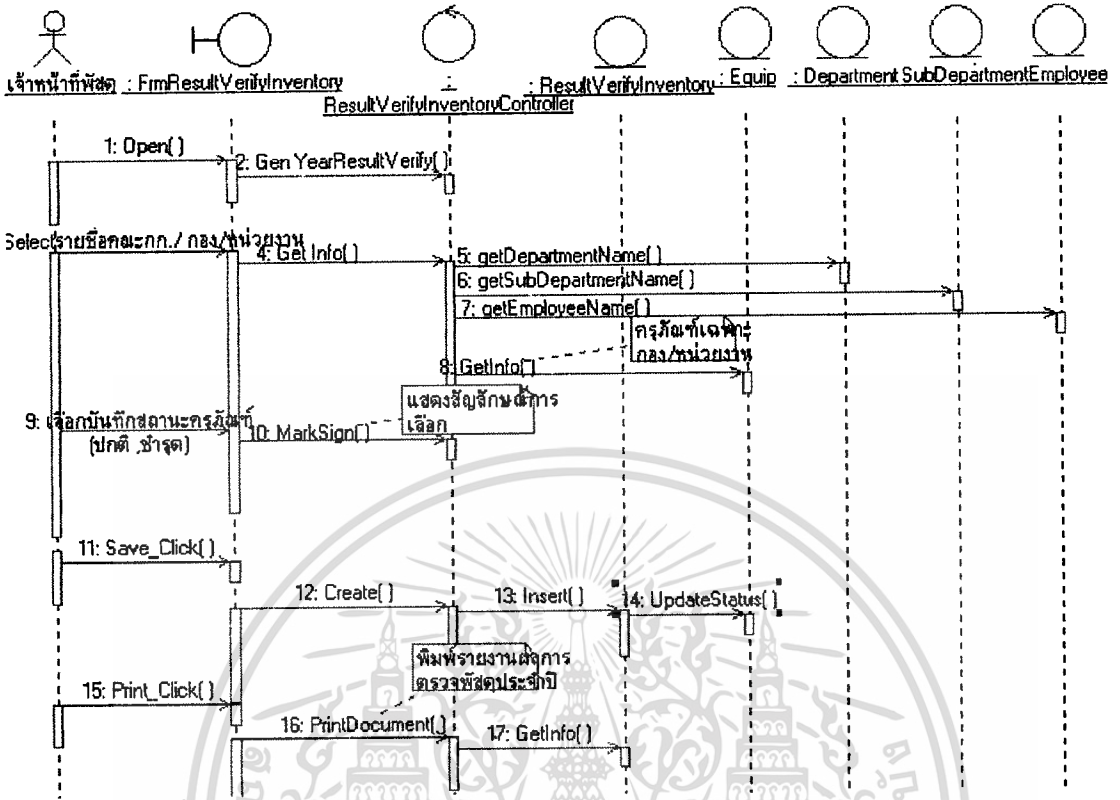
รูปที่ 3.13 Sequence Diagram ของ Use Case เปิดกรงกัญชา



รูปที่ 3.14 Sequence Diagram ของ Use Case ยืมครุภัณฑ์



รูปที่ 3.15 Sequence Diagram ของ Use Case คืนครุภัณฑ์



รูปที่ 3.16 Sequence Diagram ของ Use Case บันทึกผลตรวจพัสดุประจำปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

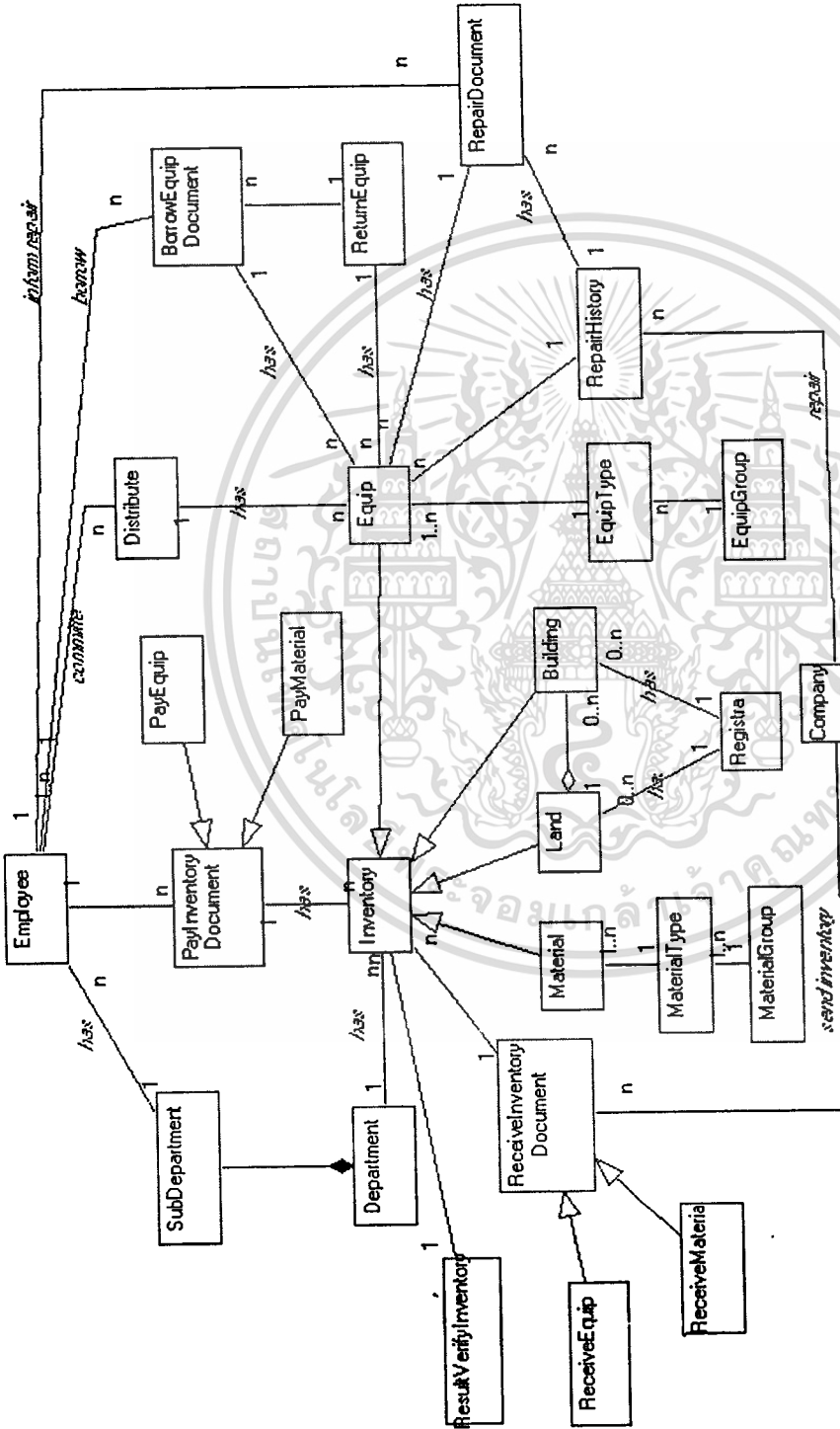
บทที่ 4

การออกแบบระบบ

การออกแบบระบบบริหารงานพัสดุของหน่วยงานราชการ ได้แก่การปรับปรุงคลาส โดยทำการกำหนดแอมพริบิวต์และการดำเนินการของคลาสที่ได้จากการวิเคราะห์ระบบโดยอาศัย Sequence Diagram ในบทที่ 3

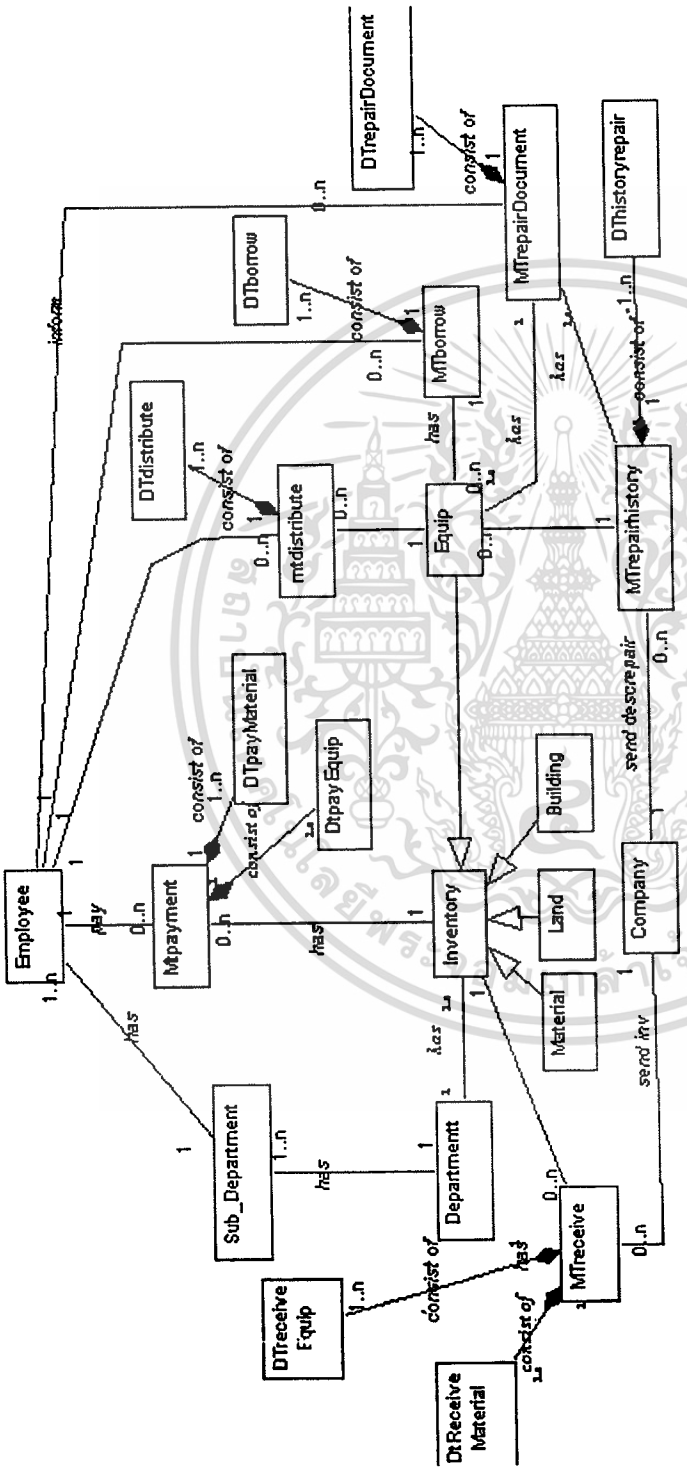
4.1 การออกแบบ Class ของระบบ

1. จากขั้นตอนการวิเคราะห์ระบบโดยใช้ Sequence Diagram ในบทที่ 3 เราต้องเพิ่มเติมและทำการปรับปรุงรายละเอียดในเรื่องของ attribute และ ชื่อของแต่ละ message ที่ object ใช้ในการติดต่อสื่อสารให้ชัดเจนลงไปซึ่ง message เหล่านี้ จะกลายเป็น method หรือ operation ของ Object ใน Class ต่อไป
2. หลังจากนั้นเราก็จะได้ Class โดยแสดงเป็น Class Diagram ของระบบได้ดังรูปที่ 4.1
3. ทำการนอมนัดไลซ์ Class เนื่องจาก Class เหล่านี้จะต้องถูกนำไปใช้ในขั้นตอนของการออกแบบโครงสร้างฐานข้อมูล หรือ Database Schema ต่อไป โดยสามารถแสดงได้ดังรูปที่ 4.2
4. ทำการสร้างคลาสโคออดิเนตสำหรับการแปลงคลาสโคออดิเนตไปเป็นโครงสร้างฐานข้อมูลหรือ Database Schema ซึ่งได้อธิบายรายละเอียดไว้ในหัวข้อ 4.2



รูปที่ 4.1 Class Diagram ของระบบงานบริการงานพัสดุตามระเบียบสำนักงานกษัตริย์ระดับที่ 1

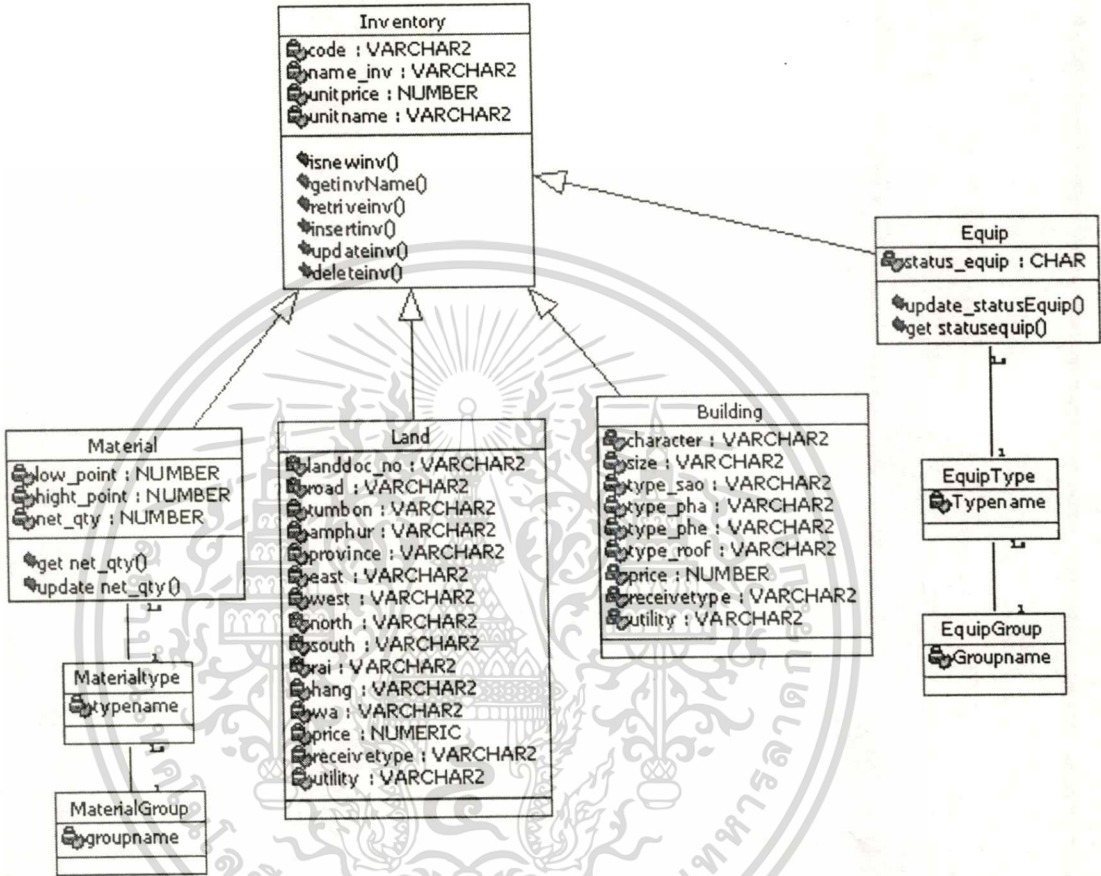
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 Class Diagram ของระบบงานบริหารงานพัสดุตามระเบียบสำนักนายกรัฐมนตรีระดับที่ 2

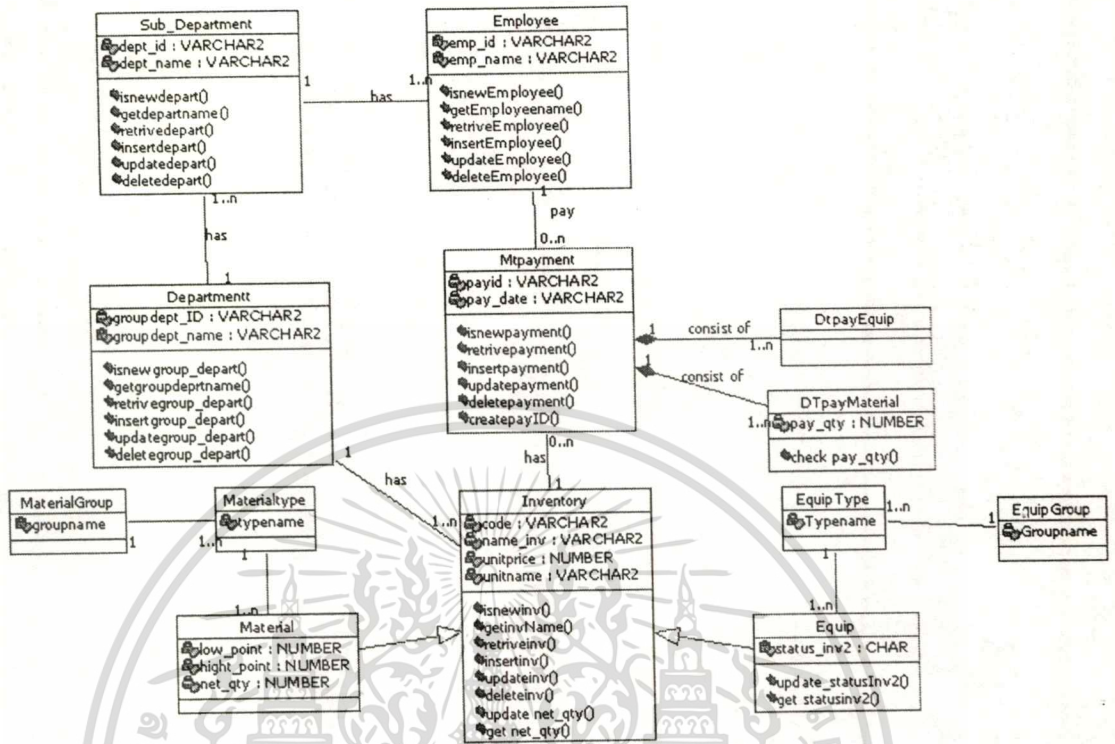
ต่อไปจะแสดงรายละเอียดของ Class ที่ประกอบไปด้วย attribute และ method ดังรูปที่ 4.3-

4.

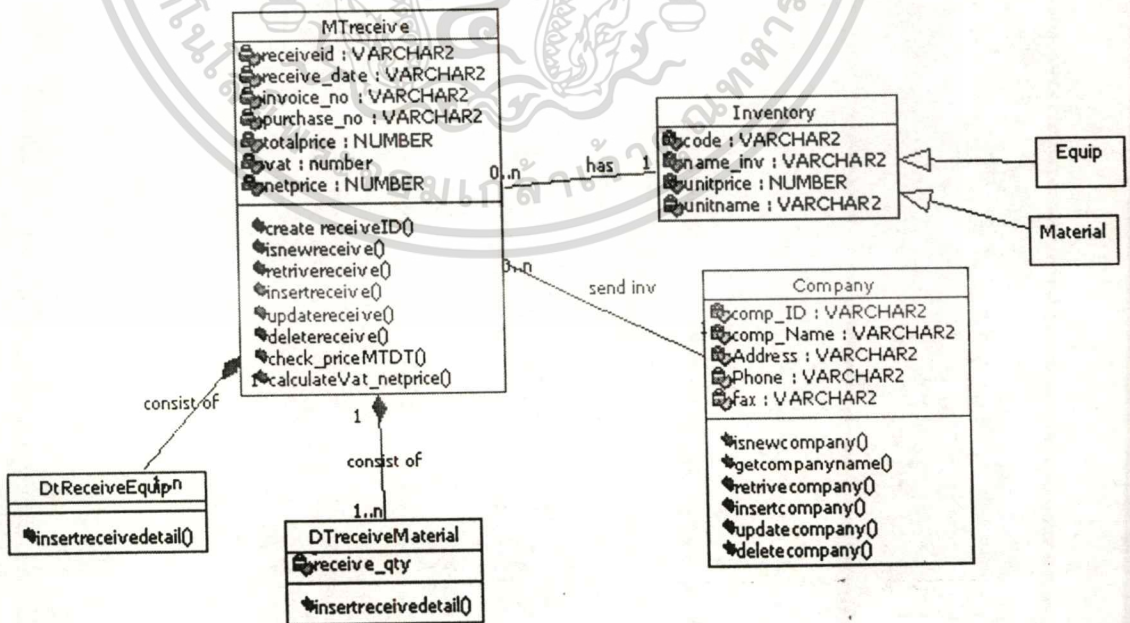


รูปที่ 4.3 Class Diagram ของระบบ ส่วนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

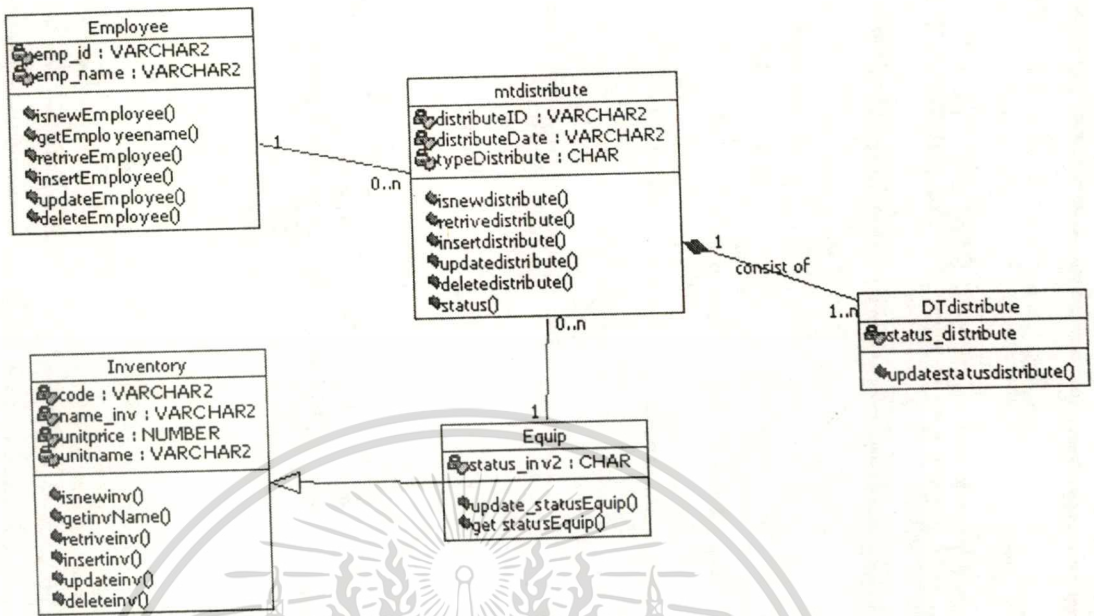


รูปที่ 4.4 Class Diagram ของระบบ ส่วนที่ 2

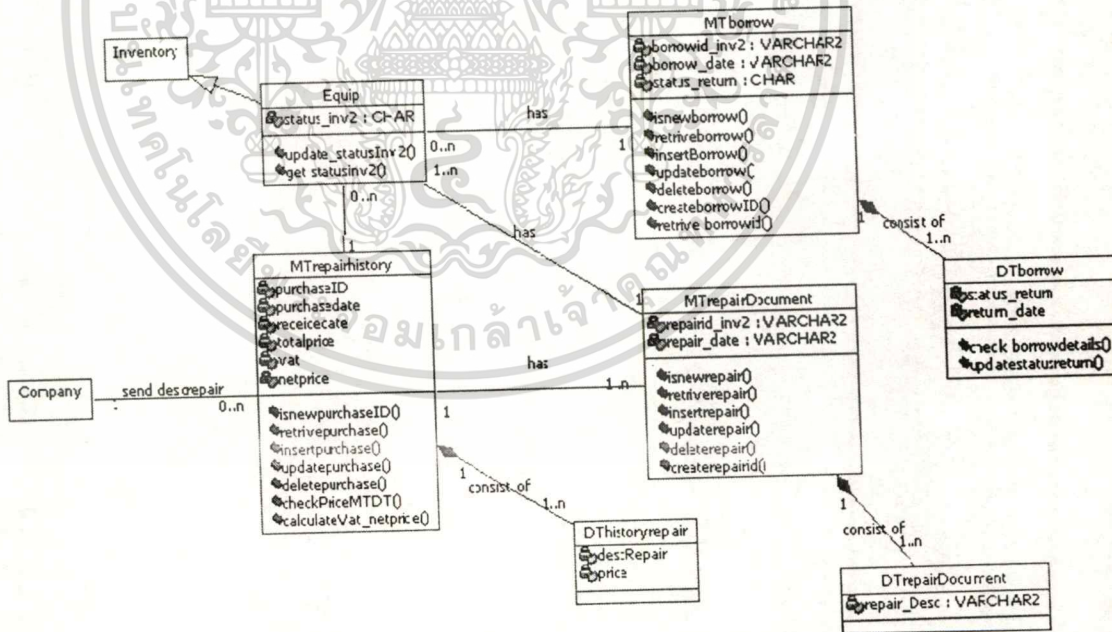


รูปที่ 4.5 Class Diagram ของระบบ ส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 Class Diagram ของระบบ ส่วนที่ 4



รูปที่ 4.7 Class Diagram ของระบบ ส่วนที่ 5

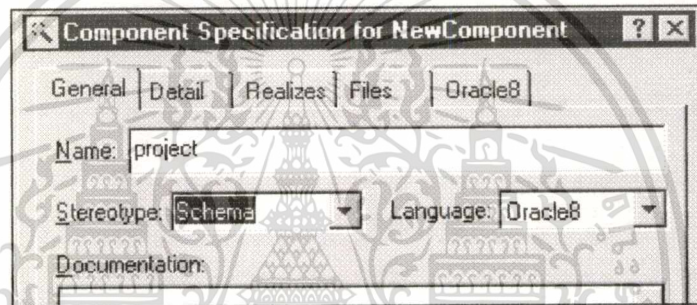
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การสร้างคลาสไดอะแกรมสำหรับการแปลงคลาสไดอะแกรมเป็นโครงสร้างฐานข้อมูล

การสร้างคลาสไดอะแกรมเพื่อการแปลงที่นำไปสู่การนิยามฐานข้อมูล จะทำการลอกเลียนมาจากคลาสที่สร้างไว้แล้ว และทำการปรับให้เหมาะสมกับระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งเราอาจจะแปลงได้หลายแบบ คือ การแปลงหนึ่งคลาสเป็นหนึ่งรีเลชัน หรือ บางคลาสอาจต้องแปลงเป็นหลาย ๆ รีเลชัน (หรือหลาย Table)

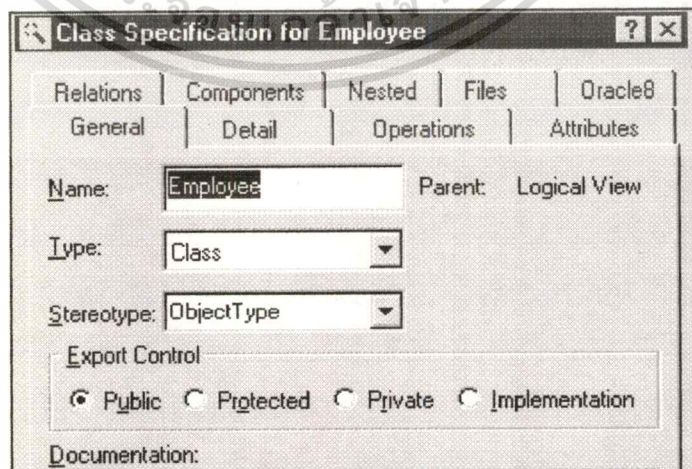
ในส่วนของ Relation Rose จะมีเครื่องมือที่ช่วยในการสนับสนุนการนิยามของออโรเคิล 8 ซึ่งจะมีขั้นตอนในการแปลงคลาสไปสู่การนิยามฐานข้อมูลดังนี้

1. ทำการเพิ่มคอมโพเนนท์ ที่มีสเตอริโอไทป์ เป็น schema ซึ่งใช้สำหรับ Oracle 8 ดังรูปที่ 4.8 และชื่อคอมโพเนนท์ ต้องสอดคล้องกับชื่อ schema ของฐานข้อมูลในออโรเคิล



รูปที่ 4.8 แสดงการสร้างคอมโพเนนท์

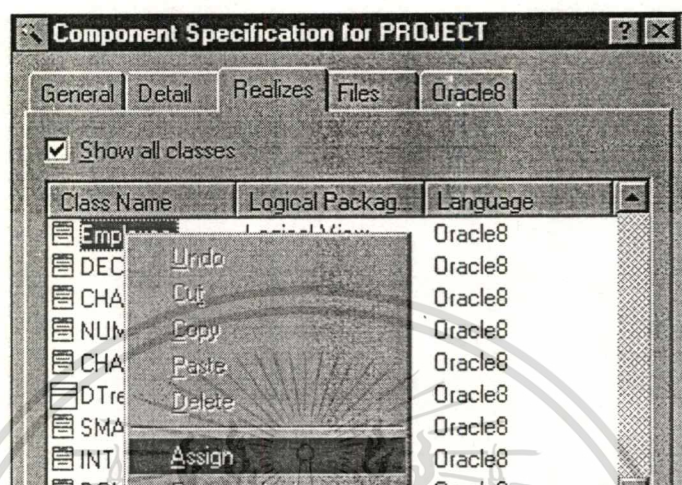
2. ทำการกำหนดสเตอริโอไทป์ ให้เป็น <<Object type>> ให้กับทุกคลาสที่จะใช้เป็นตัวแบบในการแปลงคลาส ซึ่งแสดงดังรูปที่ 4.2 เป็นการกำหนดสเตอริโอไทป์ ให้กับคลาส Employee



รูปที่ 4.9 การกำหนดสเตอริโอไทป์ ให้กับคลาส Employee

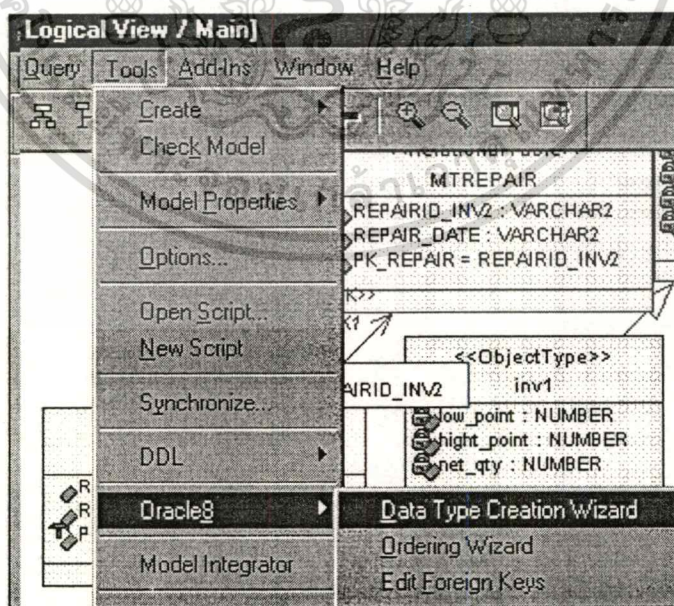
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการกำหนดให้ทุกคลาสอยู่ในคอมโพเนนท์ที่กำหนด โดยคลิกขวาที่ชื่อคลาสแล้วเลือก Assign ดังรูปที่ 4.10



รูปที่ 4.10 การกำหนดให้คลาส Employee อยู่ในคอมโพเนนท์ project

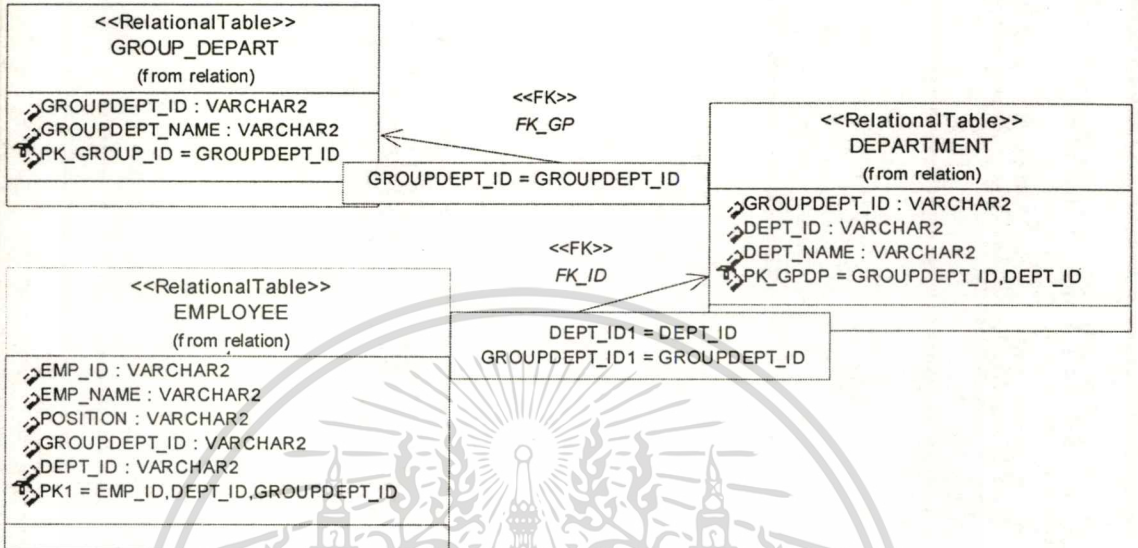
1. ใช้ Create Data Type Wizard สร้างคลาสที่มีสเตอริโอไทป์เป็น <<Relational Table>> โดยเลือก Tool → Oracle 8 → Create Data Type Wizard ดังรูปที่ 4.11



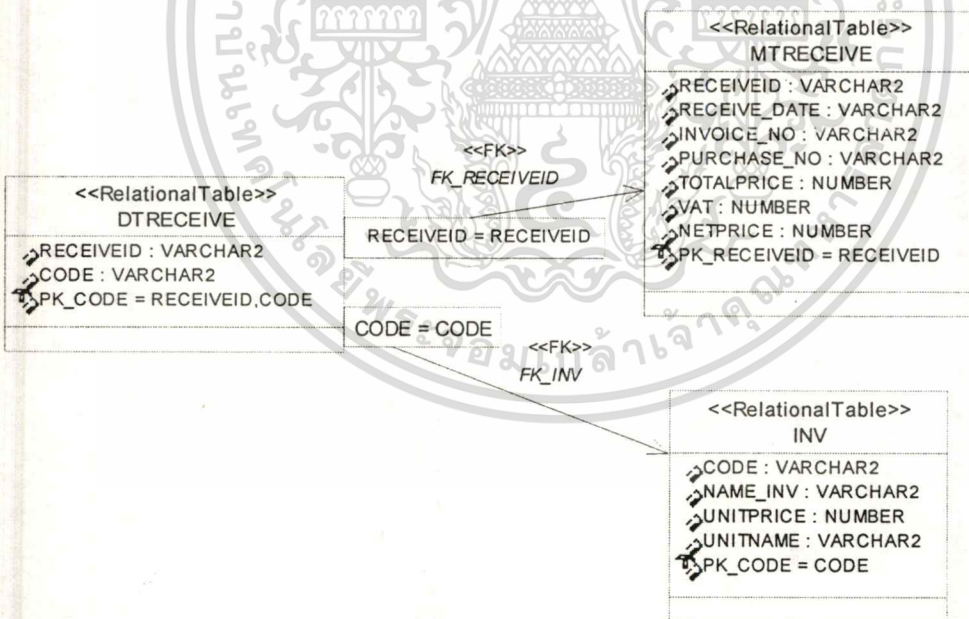
รูปที่ 4.11 สร้างคลาสที่มีสเตอริโอไทป์เป็น <<Relational Table>>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำตามลำดับที่เครื่องมือนี้แนะนำจะได้คลาสสำหรับฐานข้อมูลของระบบบริหารงานพัสดุ ดังรูปที่ 4.12-4.17

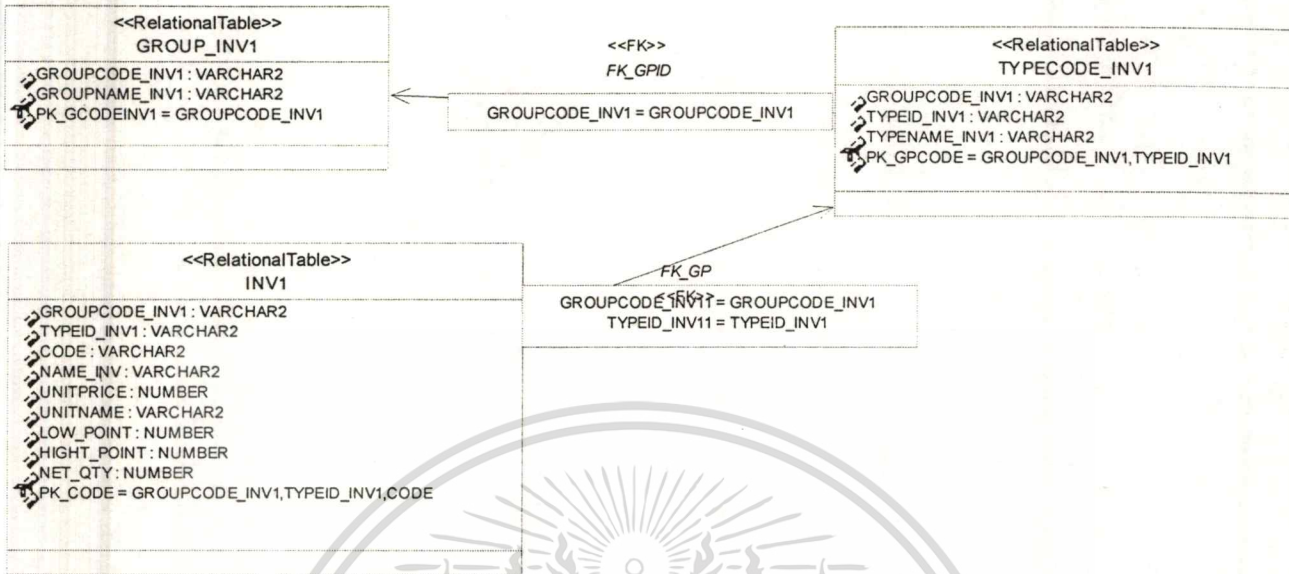


รูปที่ 4.12 คลาสรีเลชันกอง หน่วยงาน และเจ้าหน้าที่



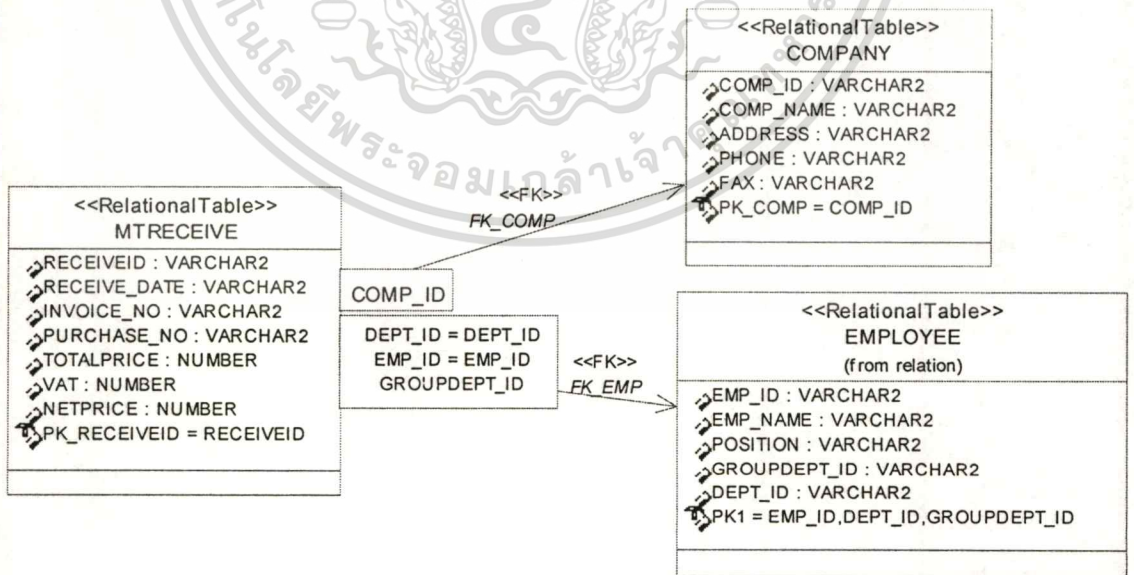
รูปที่ 4.13 คลาสรีเลชันใบรับพัสดุ ส่วนหลัก ส่วนรายละเอียด และเจ้าหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



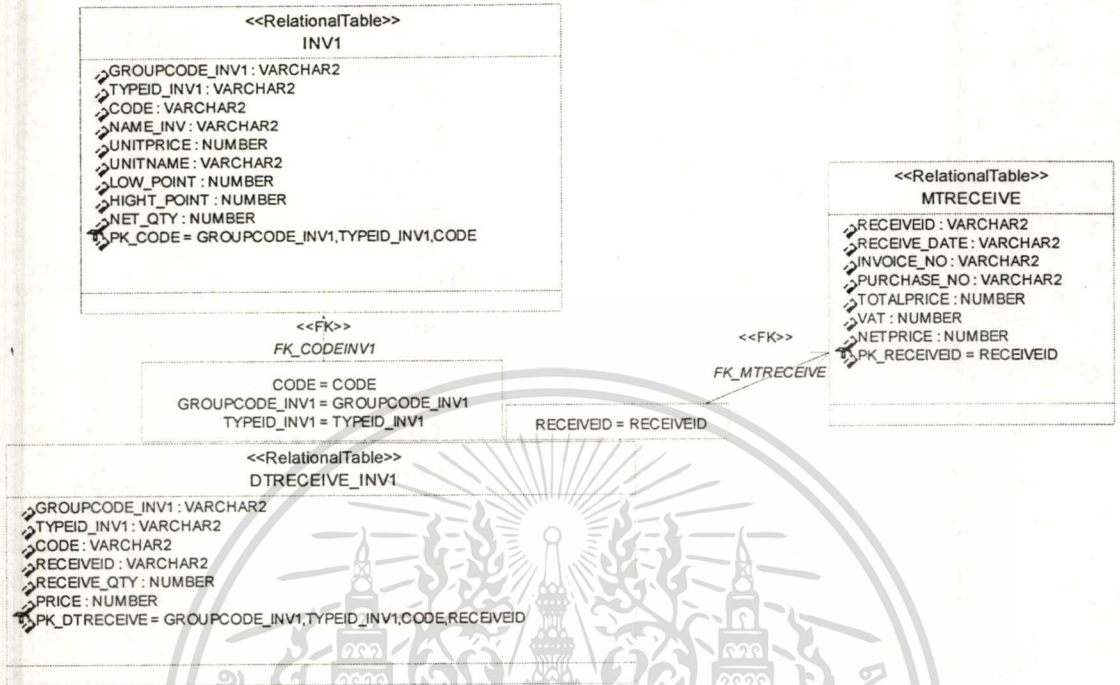
รูปที่ 4.14 คลาสรีเลขณ์ของรหัสกลุ่มวัสดุ ประเภทวัสดุ และ วัสดุ

หมายเหตุ คลาสรีเลขณ์กลุ่มวัสดุ และประเภทวัสดุเป็น คลาสรีเลขณ์ที่สร้างเพิ่มขึ้นมาเพื่อให้ตรงกับระเบียบการกำหนดรหัสของสำนักนายกรัฐมนตรี และการแปลงคลาสิรีเลขณ์ของครุภัณฑ์ก็ใช้วิธีเดียวกันนี้

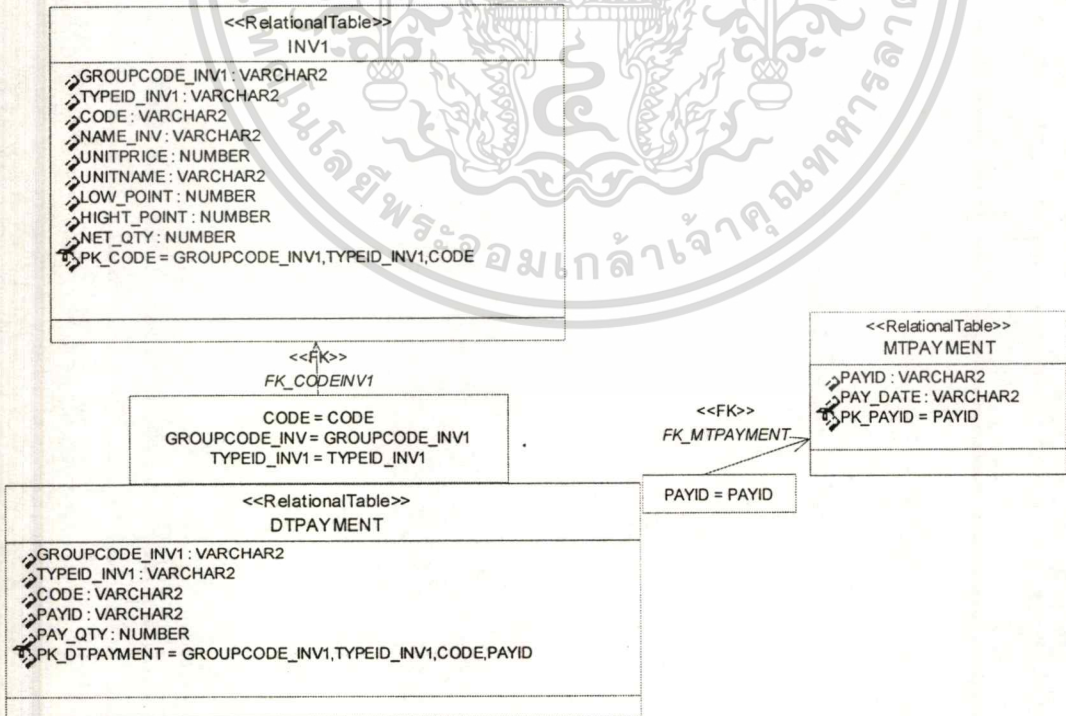


รูปที่ 4.15 คลาสรีเลขณ์ของใบรับวัสดุ(รายการหลัก) บริษัท และ เจ้าหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 คลาสรีเลย์ชันของใบรับวัสดุ(รายการหลัก ,รายการย่อย) และ วัสดุ



รูปที่ 4.17 คลาสรีเลย์ชันของใบเบิกวัสดุ(รายการหลัก ,รายการย่อย) และ วัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

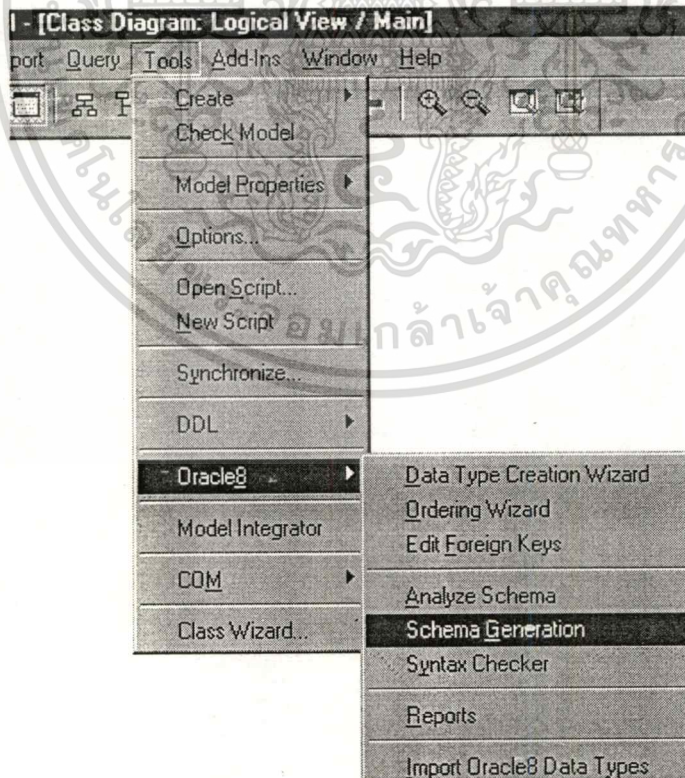
คลาสอื่น ๆ ก็มีลักษณะการแปลงเป็นคลาสรีเลชันในทำนองเดียวกันนี้ โดยพิจารณาถึงความสัมพันธ์กันระหว่างคลาส โดยมีข้อสังเกตในการแปลงคลาสได้ดังนี้

1. การกำหนดแอทริบิวต์ที่เป็นคีย์หลัก (Primary Key) มีสตอริโอไทป์เป็น <<PK>>
2. การกำหนดแอทริบิวต์ที่เป็นคีย์นอก (Foreign Key) มีสตอริโอไทป์เป็น <<FK>>และเป็นแอทริบิวต์ที่อ้างอิงถึงคีย์หลักของรีเลชัน
3. ความสัมพันธ์ระหว่างคลาสเป็นแบบทิศทางเดียวหัวลูกศรเปิดทางด้านคลาสที่คีย์นอกอ้างอิง

4.3 คำสั่งนิยามฐานข้อมูล

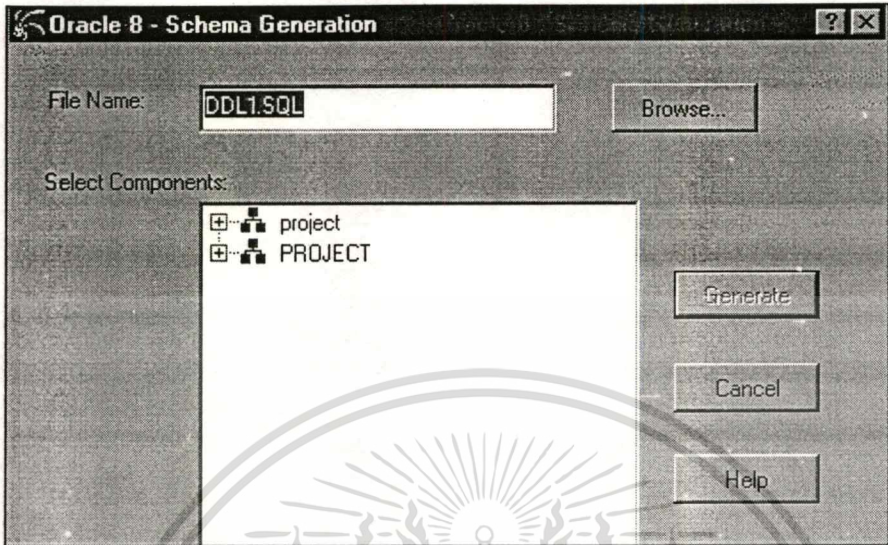
หลังจากที่ทำการแปลงคลาสเป็นคลาสรีเลชัน(หรืออาจสร้างคลาสรีเลชันเพิ่มเติม) แล้วให้เลือกที่เมนู Tools→Oracle8→Schema Generation และทำการอ้างอิงคอมโพเนนท์ ดังรูปที่ 4.18 –4.19 เพื่อทำการสร้างการนิยามฐานข้อมูล

สำหรับออราเคิลจะได้ไฟล์ นามสกุล .sql ซึ่งภายในจะบรรจุคำสั่งในการนิยามฐานข้อมูลสำหรับระบบนั่นเอง

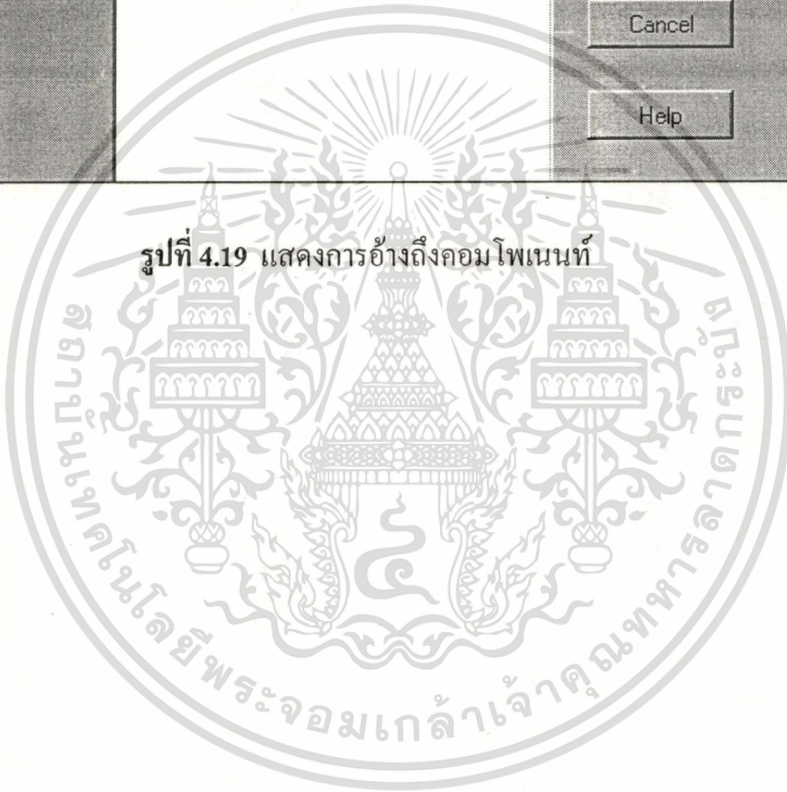


รูปที่ 4.18 แสดงการสร้าง Schema generation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงการอ้างถึงคอม โฟเนนท์



ไฟล์ที่ได้จะทำการบันทึกเก็บไว้ในชื่อไฟล์ DDL1.sql ซึ่งได้ยกตัวอย่างมาบางส่วน มีรายละเอียดดังรูปที่ 4.20

```

CREATE TABLE PROJECT.GROUP_DEPART (
  GROUPDEPT_ID VARCHAR2(2),
  GROUPDEPT_NAME VARCHAR2(30),
  CONSTRAINT PK_GROUP_ID PRIMARY KEY (GROUPDEPT_ID));

CREATE TABLE PROJECT.INV2 (
  CODE VARCHAR2(21),
  NAME_INV VARCHAR2(30),
  UNITPRICE NUMBER(2,6),
  UNITNAME VARCHAR2(30),
  STATUS_INV2 CHAR(1),
  CONSTRAINT PK_CODE PRIMARY KEY (CODE));

CREATE TABLE PROJECT.DEPARTMENT (
  DEPT_ID VARCHAR2(2),
  DEPT_NAME VARCHAR2(30),
  GROUPDEPT_ID VARCHAR2(2),

  CONSTRAINT PK_GDPD PRIMARY KEY (GROUPDEPT_ID,DEPT_ID),
  CONSTRAINT FK_GP FOREIGN KEY(GROUPDEPT_ID) REFERENCES
PROJECT.GROUP_DEPART(GROUPDEPT_ID));

CREATE TABLE PROJECT.COMPANY (
  COMP_ID VARCHAR2(4),
  COMP_NAME VARCHAR2(30),
  ADDRESS VARCHAR2(50),
  PHONE VARCHAR2(30),
  FAX VARCHAR2(30),
  CONSTRAINT PK_COMP PRIMARY KEY (COMP_ID));

```

รูปที่ 4.20 คำสั่งนิยามฐานข้อมูลที่ได้ ในไฟล์ Schema.sql(ส่วนที่ 1/5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE TABLE PROJECT.GROUP_INV1 (
  GROUPCODE_INV1 VARCHAR2(4),
  GROUPNAME_INV1 VARCHAR2(30),
  CONSTRAINT PK_GCODEINV1 PRIMARY KEY (GROUPCODE_INV1));

CREATE TABLE PROJECT.EMPLOYEE (
  EMP_ID VARCHAR2(4),
  EMP_NAME VARCHAR2(30),
  POSITION VARCHAR2(30),
  GROUPDEPT_ID VARCHAR2(2),
  DEPT_ID VARCHAR2(2),
  CONSTRAINT PK1 PRIMARY KEY (EMP_ID,DEPT_ID,GROUPDEPT_ID),
  CONSTRAINT FK_ID FOREIGN KEY(DEPT_ID,GROUPDEPT_ID) REFERENCES
PROJECT.DEPARTMENT(DEPT_ID,GROUPDEPT_ID));

CREATE TABLE PROJECT.TYPECODE_INV1 (
  GROUPCODE_INV1 VARCHAR2(4),
  TYPEID_INV1 VARCHAR2(3),
  TYPENAME_INV1 VARCHAR2(30),
  ,
  CONSTRAINT PK_GPCODE PRIMARY KEY (GROUPCODE_INV1,TYPEID_INV1),
  CONSTRAINT FK_GPID FOREIGN KEY(GROUPCODE_INV1) REFERENCES
PROJECT.GROUP_INV1(GROUPCODE_INV1));

CREATE TABLE PROJECT.MTRECEIVE (
  RECEIVED VARCHAR2(10),
  RECEIVE_DATE VARCHAR2(10),
  INVOICE_NO VARCHAR2(10),
  PURCHASE_NO VARCHAR2(10),

```

รูปที่ 4.20 คำสั่งนิยามฐานข้อมูลที่ได้ ในไฟล์ Schema.sql(ส่วนที่ 2/5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TOTALPRICE NUMBER(2,6),
VAT NUMBER(2,6),
NETPRICE NUMBER(2,6),
DEPT_ID VARCHAR2(2),
EMP_ID VARCHAR2(4),
GROUPDEPT_ID VARCHAR2(2),
COMP_ID VARCHAR2(4),
CONSTRAINT PK_RECEIVEID PRIMARY KEY (RECEIVEID),
CONSTRAINT FK_EMP FOREIGN KEY(DEPT_ID,EMP_ID,GROUPDEPT_ID) REFERENCES
PROJECT.EMPLOYEE(DEPT_ID,EMP_ID,GROUPDEPT_ID),
CONSTRAINT FK_COMP FOREIGN KEY(COMP_ID) REFERENCES
PROJECT.COMPANY(COMP_ID));

CREATE TABLE PROJECT.MTPAYMENT (
PAYID VARCHAR2(10),
PAY_DATE VARCHAR2(10),
DEPT_ID VARCHAR2(2),
EMP_ID VARCHAR2(4),
GROUPDEPT_ID VARCHAR2(2),
CONSTRAINT PK_PAYID PRIMARY KEY (PAYID),
CONSTRAINT FK_EMP FOREIGN KEY(DEPT_ID,EMP_ID,GROUPDEPT_ID) REFERENCES
PROJECT.EMPLOYEE(DEPT_ID,EMP_ID,GROUPDEPT_ID));

CREATE TABLE PROJECT.INV1 (
GROUPCODE_INV1 VARCHAR2(4),
TYPEID_INV1 VARCHAR2(3),
CODE VARCHAR2(21),
NAME_INV VARCHAR2(30),
UNITPRICE NUMBER(2,6),
UNITNAME VARCHAR2(30),
LOW_POINT NUMBER(2,6),

```

รูปที่ 4.20 คำสั่งนิยามฐานข้อมูลที่ได้ ในไฟล์ Schema.sql(ส่วนที่ 3/5)

```

HIGHT_POINT NUMBER(2,6).
NET_QTY NUMBER(2,6).
GROUPCODE_INV11 VARCHAR2(4),
TYPEID_INV11 VARCHAR2(3),
CONSTRAINT PK_CODE PRIMARY KEY (GROUPCODE_INV1,TYPEID_INV1,CODE),
CONSTRAINT FK_GP FOREIGN KEY(GROUPCODE_INV11,TYPEID_INV11) REFERENCES
PROJECT.TYPECODE_INV1(GROUPCODE_INV1,TYPEID_INV1));

```

```

CREATE TABLE PROJECT.DTPAYMENT (
GROUPCODE_INV1 VARCHAR2(4),
TYPEID_INV1 VARCHAR2(3),
CODE VARCHAR2(21),
PAYID VARCHAR2(10),
PAY_QTY NUMBER(2,6),
CONSTRAINT PK_DTPAYMENT PRIMARY KEY
(GROUPCODE_INV1,TYPEID_INV1,CODE,PAYID),
CONSTRAINT FK_CODEINV1 FOREIGN KEY(CODE,GROUPCODE_INV1,TYPEID_INV1)
REFERENCES PROJECT.INV1(CODE,GROUPCODE_INV1,TYPEID_INV1),
CONSTRAINT FK_MTPAYMENT FOREIGN KEY(PAYID) REFERENCES
PROJECT.MTPAYMENT(PAYID));

```

```

CREATE TABLE PROJECT.DTRECEIVE_INV1 (
GROUPCODE_INV1 VARCHAR2(4),
TYPEID_INV1 VARCHAR2(3),
CODE VARCHAR2(21),
RECEIVEID VARCHAR2(10),
RECEIVE_QTY NUMBER(2,6),
PRICE NUMBER(2,6),

```

รูปที่ 4.20 คำสั่งนิยามฐานข้อมูลที่ได้ ในไฟล์ Schema.sql(ส่วนที่ 4/5)

```
CONSTRAINT PK_DTRECEIVE PRIMARY KEY
(GROUPCODE_INV1,TYPEID_INV1,CODE,RECEIVEID),
  CONSTRAINT FK_CODEINV1 FOREIGN KEY(CODE,GROUPCODE_INV1,TYPEID_INV1)
REFERENCES PROJECT.INV1(CODE,GROUPCODE_INV1,TYPEID_INV1).
  CONSTRAINT FK_MTRECEIVE FOREIGN KEY(RECEIVEID) REFERENCES
PROJECT.MTRECEIVE(RECEIVEID));
```

รูปที่ 4.20 คำสั่งนิยามฐานข้อมูลที่ได้ ในไฟล์ Schema.sql(ส่วนที่ 5/5)



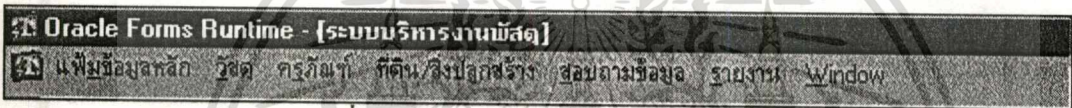
บทที่ 5

การออกแบบยูสเซอร์อินเตอร์เฟซ

การออกแบบยูสเซอร์อินเตอร์เฟซระบบบริหารงานพัสดุของหน่วยงานราชการ ซึ่งจะขึ้นอยู่กับฐานข้อมูลของระบบ และผู้พัฒนาได้เลือกใช้ Developer 2000 ของออราเคิล ซึ่งมีรายละเอียดการออกแบบดังนี้

5.1 การออกแบบเมนู

การออกแบบยูสเซอร์อินเตอร์เฟซ จะเริ่มจากเมนูหลัก ซึ่งต้องเรียกใช้ผ่านฟอร์มหลัก แสดงได้ดังรูปที่ 5.1

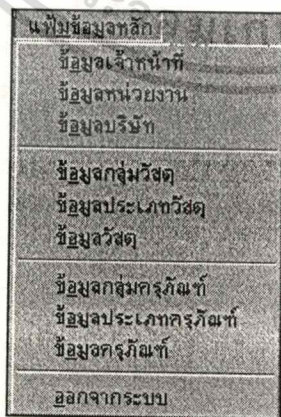


รูปที่ 5.1 แสดง เมนูหลักของระบบบริหารงานพัสดุ

เมนูหลักของระบบแบ่งเป็นหมวด ๆ ได้แก่ เมนูเพิ่มข้อมูลหลัก เมนูวิจด เมนูครุภัณฑ์ เมนูที่ดิน/สิ่งปลูกสร้าง เมนูสอบถามข้อมูล และเมนูรายงาน ซึ่งมีรายละเอียดของแต่ละเมนูดังนี้

1. เมนูเพิ่มข้อมูลหลัก

เมนูเพิ่มข้อมูลหลัก ประกอบด้วยเมนูย่อยที่ทำหน้าที่เกี่ยวกับข้อมูลหลัก ๆ ที่ต้องให้ผู้ใช้ทำการป้อนข้อมูลหลัก ๆ ก่อนเพื่อนำไปใช้ทำงานในส่วนอื่น ๆ ซึ่งแสดงดังรูปที่ 5.2



รูปที่ 5.2 แสดง เมนูย่อยของเมนูเพิ่มข้อมูลหลัก

และสามารถอธิบายรายละเอียดได้ดังตารางที่ 5.1

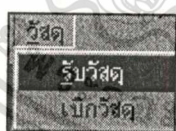
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 คำอธิบายเมนูย่อยของเมนูเพิ่มข้อมูลหลัก

ชื่อเมนูย่อย	คำอธิบาย
1. ข้อมูลเจ้าหน้าที่	เปิดฟอร์มข้อมูลเจ้าหน้าที่
2. ข้อมูลหน่วยงาน	เปิดฟอร์มข้อมูลหน่วยงาน
3. ข้อมูลบริษัท	เปิดฟอร์มข้อมูลบริษัท
4. ข้อมูลกลุ่มวัสดุ	เปิดฟอร์มข้อมูลกลุ่มวัสดุ
5. ข้อมูลประเภทวัสดุ	เปิดฟอร์มข้อมูลประเภทวัสดุ
5. ข้อมูลวัสดุ	เปิดฟอร์มข้อมูลวัสดุ
7. ข้อมูลกลุ่มครุภัณฑ์	เปิดฟอร์มข้อมูลกลุ่มครุภัณฑ์
8. ข้อมูลประเภทครุภัณฑ์	เปิดฟอร์มข้อมูลประเภทครุภัณฑ์
9. ข้อมูลครุภัณฑ์	เปิดฟอร์มข้อมูลครุภัณฑ์
10. ออกจากระบบ	สำหรับออกจากระบบงาน

2. เมนูวัสดุ

เมนูวัสดุ ประกอบด้วยเมนูย่อยของรายการวัสดุ ประกอบด้วยเมนูย่อย วัสดุ และเบิกวัสดุ ซึ่งแสดงดังรูปที่ 5.3



รูปที่ 5.3 แสดง เมนูย่อยของเมนูวัสดุ

และสามารถอธิบายรายละเอียดได้ดังตารางที่ 5.2

ตารางที่ 5.2 คำอธิบายเมนูย่อยของเมนูวัสดุ

ชื่อเมนูย่อย	คำอธิบาย
1. รับวัสดุ	เปิดฟอร์มรับวัสดุ
2. เบิกวัสดุ	เปิดฟอร์มเบิก/จ่ายวัสดุ

3. เมนูครุภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูครุภัณฑ์ ประกอบด้วยเมนูย่อยของรายการครุภัณฑ์ ประกอบด้วยเมนูย่อย รับประทาน และเบิกครุภัณฑ์ ซึ่งแสดงดังรูปที่ 5.4

ครุภัณฑ์
รับประทาน
เบิกครุภัณฑ์
ยืมครุภัณฑ์
คืนครุภัณฑ์
แจ้งซ่อมครุภัณฑ์
ประวัติการซ่อม

รูปที่ 5.4 แสดง เมนูย่อยของเมนูครุภัณฑ์

และสามารถอธิบายรายละเอียดได้ดังตารางที่ 5.2

ตารางที่ 5.3 คำอธิบายเมนูย่อยของเมนูครุภัณฑ์

ชื่อเมนูย่อย	คำอธิบาย
1. รับประทาน	เปิดฟอร์มรับประทาน
2. เบิกครุภัณฑ์	เปิดฟอร์มเบิก/จ่ายครุภัณฑ์
3. ยืมครุภัณฑ์	เปิดฟอร์มยืมครุภัณฑ์
4. คืนครุภัณฑ์	เปิดฟอร์มคืนครุภัณฑ์
5. แจ้งซ่อมครุภัณฑ์	เปิดฟอร์มรับแจ้งซ่อมครุภัณฑ์
5. ประวัติการซ่อมครุภัณฑ์	เปิดฟอร์มประวัติการซ่อมครุภัณฑ์

4. เมนูที่ดิน/สิ่งปลูกสร้าง

เมนูที่ดิน/สิ่งปลูกสร้าง ประกอบด้วยเมนูย่อย ที่ดิน และสิ่งปลูกสร้างซึ่งแสดงดังรูปที่ 5.5

ที่ดิน/สิ่งปลูกสร้าง
ที่ดิน
สิ่งปลูกสร้าง

รูปที่ 5.5 แสดง เมนูย่อยของเมนูที่ดิน/สิ่งปลูกสร้าง

และสามารถอธิบายรายละเอียดได้ดังตารางที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.4 คำอธิบายเมนูย่อยของเมนูที่คืน/สิ่งปลูกสร้าง

ชื่อเมนูย่อย	คำอธิบาย
1.ที่คืน	เปิดฟอร์มที่คืน
2.สิ่งปลูกสร้าง	เปิดฟอร์มสิ่งปลูกสร้าง

5.2 การออกแบบฟอร์มยูสเซอร์อินเตอร์เฟซ

การออกแบบฟอร์มยูสเซอร์อินเตอร์เฟซ จะมีรายละเอียดในแต่ละฟอร์ม ดังรายละเอียดต่อไปนี้

1. แถบแสดงสถานะ(Status Bar)

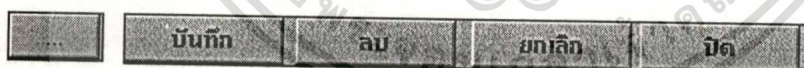
ส่วนประกอบที่ทุกฟอร์มมีคือ แถบแสดงสถานะการทำงาน แสดงดังรูปที่ 5.5 ซึ่งประกอบด้วย

- ชื่อผู้ใช้งานระบบ
- วันที่/เวลาปัจจุบัน



รูปที่ 5.5 ตัวอย่างแถบสถานะ

2. ปุ่มคำสั่ง (Command Button)



รูปที่ 5.7 ลักษณะของปุ่มคำสั่ง

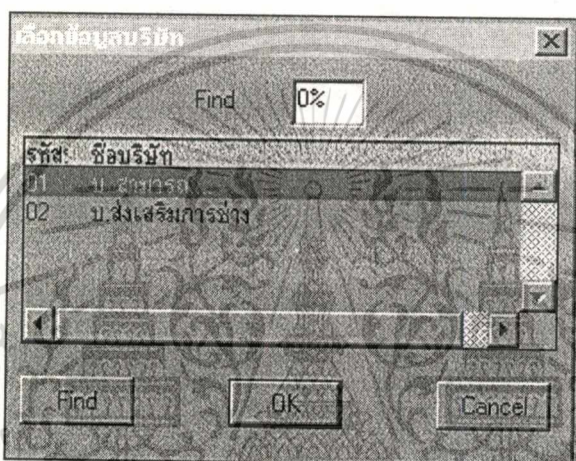
จากรูปที่ 5.7 แต่ละปุ่มจะมีลักษณะการทำงานต่าง ๆ ดังนี้คือ

- ปุ่มคำสั่ง “...” ใช้เมื่อต้องการเลือกรายการข้อมูลที่มีการบันทึกไว้ในระบบแล้ว จะวางอยู่ระดับเดียวกับรายการข้อมูลนั้น ๆ เช่น วางอยู่ระดับเดียวกับข้อมูลหน่วยงาน เมื่อผู้ใช้กดปุ่มดังกล่าว โปรแกรมก็จะแสดงข้อมูลหน่วยงานให้ผู้ใช้เลือก
- ปุ่มคำสั่ง “บันทึก” ใช้เมื่อต้องการบันทึกข้อมูล
- ปุ่มคำสั่ง “ลบ” ใช้เมื่อต้องการลบข้อมูล

- ปุ่มคำสั่ง “ยกเลิก” ใช้เมื่อต้องการยกเลิกการป้อนข้อมูลรายการนั้น ๆ โดยไม่ต้องการบันทึกข้อมูล
- ปุ่มคำสั่ง “ปิด” ใช้เมื่อต้องการปิดฟอร์มนั้น ๆ
-

3.รายการค่าข้อมูล

รายการค่าข้อมูลใช้แสดงข้อมูลต่างๆ ภายในระบบ และสามารถค้นหา และเลือกรายการที่ต้องการได้ ซึ่งแสดงดังรูปที่ 5.8



รูปที่ 5.8 รายการค่าข้อมูล

3.1 ไดอะล็อกแสดงข้อความเตือน

แสดงขึ้นเพื่อเตือนผู้ใช้งาน เช่น ในกรณีที่ต้องการลบข้อมูล หรือผู้ใช้ป้อนข้อมูลไม่ถูกต้อง ซึ่งกรณีโดยทั่วไปจะมีให้เลือก 2 รายการ คือ ตกลง และยกเลิก แต่กรณีที่ผู้ใช้ป้อนข้อมูลผิดจะมีเพียงรายการตกลง เพื่อให้ผู้ใช้ตรวจสอบข้อมูลต่างๆ อีกครั้ง แสดงดังรูปที่ 5.9



รูปที่ 5.9 ไดอะล็อกแสดงข้อความเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ไอ้ะลอกการเข้าสู่อระบบ

จะแสดงขึ้นเมื่อผู้ใช้งานเริ่มเข้าสู่ระบบ ผู้ใช้ต้องป้อนชื่อผู้ใช้ (Login Name) และ รหัสผ่าน (Password) ดังรูปที่ 5.10

รูปที่ 5.10 ไอ้ะลอกการเข้าสู่อระบบ

5. ฟอร์มข้อมูลหน่วยงาน

ฟอร์มข้อมูลหน่วยงาน เมื่อผู้ใช้ป้อนข้อมูลกอง ถ้าเป็นรายการที่มีการบันทึกไว้แล้วจะแสดงข้อมูลของหน่วยงาน ที่อยู่ในกองนั้น ๆ ผู้ใช้สามารถเพิ่มข้อมูลได้ หรือ ผู้ใช้สามารถกดปุ่ม"..." เพื่อดูข้อมูลกองที่บันทึกข้อมูลไว้ได้

รหัสหน่วยงาน	ชื่อหน่วยงาน	กองรหัสกอง
01	หน่วยพิจด	225
02	หน่วยการเงินและบัญชี	226

รูปที่ 5.11 ฟอร์มข้อมูลหน่วยงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ฟอรัมข้อมูลกลุ่มวัสดุ

ฟอรัมข้อมูลกลุ่มวัสดุ เมื่อผู้ใช้ป้อนข้อมูลกลุ่มวัสดุ ถ้าเป็นรายการที่มีการบันทึกไว้แล้วจะแสดงชื่อกลุ่มวัสดุ ผู้ใช้สามารถแก้ไขเพิ่มข้อมูล ได้ หรือ ผู้ใช้สามารถกดปุ่ม"...” เพื่อดูข้อมูลกลุ่มวัสดุที่บันทึกข้อมูลไว้ได้

รูปที่ 5.12 ฟอรัมข้อมูลกลุ่มวัสดุ

8. φόร้มข้อมูลประเภทวัสดุ

ฟอร์้มข้อมูลประเภทวัสดุ เมื่อผู้ใ้ป้อนข้อมูลกลุ่มวัสดุ ถ้าเป็นรายการที่มีการบันทึกไว้แล้วจะแสดงข้อมูลของประเภทวัสดุ ที่อยู่ในกลุ่มวัสดุนั้น ๆ ผู้ใ้สามารถเพิ่มข้อมูลได้ หรือ ผู้ใ้สามารถกดปุ่ม"..." เพื่อดูข้อมูลประเภทวัสดุที่บันทึกข้อมูลไว้ได้

ข้อมูลประเภทวัสดุ

กลุ่มวัสดุ: 7520

ชื่อกลุ่มวัสดุ: เครื่องใช้และเครื่องประกอบสำนักงาน

รหัสประเภทวัสดุ	ชื่อประเภทวัสดุ
001	7520-001
002	7520-002
003	7520-003

บันทึก ลบ ยกเลิก ปิด

FOR 13/03/2001 08:09:43

รูปที่ 5.13 ฟอร์้มข้อมูลประเภทวัสดุ

9. ฟอรัมข้อมูลวัสดุ

ฟอรัมข้อมูลวัสดุ เมื่อผู้ใช้ป้อนข้อมูลวัสดุ ถ้าเป็นรายการที่มีการบันทึกไว้แล้วจะแสดงข้อมูลของวัสดุ นั้น ๆ ผู้ใช้สามารถเพิ่มข้อมูลได้ หรือ ผู้ใช้สามารถกดปุ่ม"... " เพื่อดูข้อมูลวัสดุที่บันทึกข้อมูลไว้ได้

ข้อมูลวัสดุ	
รหัสกลุ่มวัสดุ	7520 <input type="text" value="แก้ไข"/>
ประเภทวัสดุ	001 <input type="text" value="7520-001"/>
รหัสวัสดุ	0001 <input type="text"/>
ชื่อวัสดุ	<input type="text" value="แก้ไข"/>
จุดต่ำสุด	50 <input type="text"/>
จุดสูงสุด	100 <input type="text"/>
หน่วย	ชิ้น <input type="text"/>
ราคาหน่วย	56 <input type="text"/>
จำนวนคงเหลือ	20 <input type="text"/>
<input type="button" value="บันทึก"/> <input type="button" value="ลบ"/> <input type="button" value="ยกเลิก"/> <input type="button" value="ปิด"/>	
POB	13/09/2021 08:09:09

รูปที่ 5.14 ฟอรัมข้อมูลวัสดุ

และฟอรัมข้อมูลกลุ่มครุภัณฑ์ ประเภทครุภัณฑ์ และครุภัณฑ์ ก็มีลักษณะเดียวกับของวัสดุ

11. ฟอรั่มเบิกวัสตุ

- ระบบจะทำการสร้างเลขที่ใบเบิกวัสตุอัตโนมัติ โดยมีรูปแบบ คือ "xxxxx/yy" โดยที่ xxxxx คือ เลขที่รัน yy คือปีพ.ศ. ที่มีการเบิกวัสตุรายการนั้น ๆ ผู้ใช้ไม่สามารถป้อนเลขที่เบิกวัสตุ ที่มีหมายเลขมากกว่าที่ระบบสร้างให้ได้
- เมื่อผู้ใช้ป้อนเลขที่เบิกวัสตุ ที่มีการบันทึกไว้แล้วจะแสดงข้อมูลของรายการเบิกคั่ง กล่าว ผู้ใช้สามารถเพิ่มข้อมูล ได้ หรือ ผู้ใช้สามารถกดปุ่ม..." เพื่อดูข้อมูลรายการเบิกวัสตุที่มีการบันทึกข้อมูลไว้ได้

รายการเบิก	จำนวนคงเหลือ	จำนวนจ่าย
แฟ้ม	20	20
พืชล้มลุก	10	10

รูปที่ 5.15 ฟอรั่มข้อมูลเบิกวัสตุ

12. ฟอรั่มรับครุภัณฑ์

- ระบบจะทำการสร้างเลขที่รับครุภัณฑ์อัตโนมัติ โดยมีรูปแบบ คือ "xxxxx/yy" โดยที่ xxxxx คือ เลขที่รับ yy คือปีพ.ศ. ที่มีการรับครุภัณฑ์รายการนั้น ๆ ผู้ใช้ไม่สามารถป้อนเลขที่รับครุภัณฑ์ ที่มีหมายเลขมากกว่าที่ระบบสร้างให้ได้
- เมื่อผู้ใช้ป้อนเลขที่รับครุภัณฑ์ ที่มีการบันทึกไว้แล้วจะแสดงข้อมูลของรายการรับดังกล่าว ผู้ใช้สามารถเพิ่มข้อมูล ได้ หรือ ผู้ใช้สามารถกดปุ่ม..." เพื่อดูข้อมูลรายการรับครุภัณฑ์ที่มีการบันทึกข้อมูลไว้ได้

รับครุภัณฑ์							Price
7420	001	0001	0002/44	/n=7420-001-0001-0001/44 .lang=64			2,500.00
จำนวนเงินรวม							2,500.00

PDH

13/09/2001 08:09:45

รูปที่ 5.17 ฟอรั่มข้อมูลรับครุภัณฑ์

13. ฟอรั่มเบิกครุภัณฑ์

- ระบบจะทำการสร้างเลขที่ใบเบิกครุภัณฑ์อัตโนมัติ โดยมีรูปแบบ คือ "xxxxx/yy" โดยที่ xxxxx คือ เลขที่รัน yy คือปีพ.ศ. ที่มีการเบิกครุภัณฑ์รายการนั้น ๆ ผู้ใช้ไม่สามารถป้อนเลขที่เบิกครุภัณฑ์ ที่มีหมายเลขมากกว่าที่ระบบสร้างให้ได้
- เมื่อผู้ใช้ป้อนเลขที่เบิกครุภัณฑ์ ที่มีการบันทึกไว้แล้วจะแสดงข้อมูลของรายการเบิกดังกล่าว ผู้ใช้สามารถเพิ่มข้อมูล ได้ หรือ ผู้ใช้สามารถกดปุ่ม"..." เพื่อดูข้อมูลรายการเบิกครุภัณฑ์ที่มีการบันทึกข้อมูลไว้ได้

รายการครุภัณฑ์				
7420	001	0001	0001/44	s/n=7420-001-0001-0001/44 ran=120
7420	001	0001	0002/44	s/n=7420-001-0001-0001/44 ran=64

รูปที่ 5.18 ฟอรั่มข้อมูลเบิกครุภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. ฟอรั่มยืมครุภัณฑ์

- ระบบจะทำการสร้างเลขที่ใบยืมครุภัณฑ์อัตโนมัติ โดยมีรูปแบบ คือ "xxxxx/yy" โดยที่ xxxxx คือ เลขที่รับ yy คือปีพ.ศ. ที่มีการยืมครุภัณฑ์รายการนั้น ๆ ผู้ใช้ไม่สามารถป้อนเลขที่ยืมครุภัณฑ์ ที่มีหมายเลขมากกว่าที่ระบบสร้างให้ได้
- เมื่อผู้ใช้ป้อนเลขที่ยืมครุภัณฑ์ ที่มีการบันทึกไว้แล้วจะแสดงข้อมูลของรายการยืมดังกล่าว ผู้ใช้สามารถเพิ่มข้อมูล ได้ หรือ ผู้ใช้สามารถกดปุ่ม..." เพื่อดูข้อมูลรายการยืมครุภัณฑ์ที่มีการบันทึกข้อมูลไว้ได้

รูปที่ 5.19 ฟอรั่มข้อมูลยืมครุภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้