

การพัฒนาระบบสารสนเทศเพื่อการบริหารงานห้องสมุด  
ด้วยวิธีคอมพิวเตอร์เบส

Component-Based System Development  
of a Library Automation System



โดย

นาย ไพศาล กาญจนสุทธิรักษ์

รหัส 42067128



\*H001869\*

อาจารย์ที่ปรึกษา

กร. กัทรชัย อธิติโรจน์วงศ์
วัน เดือน ปี..... 15 ต.ค. 2530
เลขทะเบียน..... 01869
เลขเรียกหนังสือ...ดพ. พ.๑๕๖ ก. 2544
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 2 ปีการศึกษา 2544  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบสารสนเทศเพื่อการบริหารงานห้องสมุด ด้วยวิธีคอมโพเนนต์เบส
นักศึกษา	นาย ไพศาล กาญจนสุทธิรักษ์
อาจารย์ที่ปรึกษา	ดร. กัทธชัย สถิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

### บทคัดย่อ

โครงการพัฒนาระบบงานนี้ มีวัตถุประสงค์เพื่อที่จะพัฒนาระบบระบบสารสนเทศที่มีความยืดหยุ่นต่อการขยายขนาดหรือการเปลี่ยนแปลงแก้ไขในอนาคตและสนับสนุนความสามารถในการนำกลับมาใช้ใหม่ (Reusability) ได้ในทางปฏิบัติ โดยเลือกระบบสารสนเทศห้องสมุดเป็นกรณีศึกษา และใช้วิธีการพัฒนาระบบในรูปแบบของคอมโพเนนต์ (Component-Based System Development) และการประยุกต์ใช้สถาปัตยกรรมแบบหลายชั้น (Multi-Tier) ในการดำเนินการพัฒนาร่วมกับการใช้งานแบบจำลองของระบบ ซึ่งทั้งหมดนี้จะใช้กรอบงาน (Framework) ของ Microsoft Component Object Model (COM) ในการดำเนินการพัฒนาร่วมกับการใช้งาน UML ในการสร้างแบบจำลองของระบบ โดยใช้เครื่องมือ Visual Basic 6.0 Enterprise Edition ในการพัฒนาคอมโพเนนต์และส่วนของแอปพลิเคชัน และใช้ Rational Rose 2000 Enterprise Edition เป็น CASE Tool ในการสร้างแบบจำลอง UML ระบบทั้งหมดจะทำงานอยู่บน Microsoft Windows Platform โดยทำงานร่วมกับฐานข้อมูล Microsoft SQL Server 7.0 และ Microsoft IIS 5.0 Web Server

<b>Title</b>	Component-Based System Development of a Library Automation System
<b>Student</b>	Mr. Phaisan Kanchanasutiruk
<b>Advisor</b>	Dr. Pattarachai Lalitrojwong
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2001

## ABSTRACT

The objective of this System Development Project is developed information system that scalable and encourage reusability by choose the Library Automation System to be case study. The Development applies the component-base software development Process and multi-tier architecture together with system modeling to develop the case study system. The Implementation of the system use Visual Basic 6.0 Enterprise Edition to developing component and application Along with Rational Rose 2000 Enterprise Edition as UML CASE Tool. The entire system is based on Microsoft Windows Platform. In addition the system utilize SQL Server 7.0 as system database and Microsoft IIS 5.0 as web server.

## กิตติกรรมประกาศ

ในการดำเนินการพัฒนาระบบสารสนเทศเพื่อการบริหารงานห้องสมุดโดยใช้การออกแบบเชิงวัตุดิบี้ได้รับความสนับสนุนเป็นอย่างดี จากหลายๆฝ่าย ที่คอยให้คำแนะนำปรึกษา และเสียสละเวลาอันมีค่า จนทำให้การศึกษาโครงการนี้บรรลุผลตามเป้าหมายที่วางไว้ ผู้จัดทำจึงใคร่ขอขอบพระคุณ

1. บิดา มารดา ผู้ให้กำเนิด เลี้ยงดูเอาใจใส่และดูแล การอบรมให้ประพฤติในสิ่งที่ดีและถูกต้อง ตลอดจนการส่งเสริมในด้านการศึกษา ได้อย่างดีที่สุด
2. ดร. กัทรชัย ถลิตโรจน์วงศ์ อาจารย์ที่ปรึกษาโครงการที่ได้ให้ความรู้ คำปรึกษา ในการจัดทำโครงการ
3. อาจารย์ทุกๆ ท่านในคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้อบรมให้ข้าพเจ้าสามารถนำเอาความรู้พื้นฐานที่ได้ศึกษามาใช้ในการแก้ปัญหาต่างๆ
4. เพื่อนๆ IS 8 ภาคปกติ ที่ช่วยเหลือและให้กำลังใจในการทำโครงการ

นาย ไพศาล กาญจนสุทธิรักษ์  
ผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	IX
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและปัญหาของระบบปัจจุบัน.....	1
1.2 ขั้นตอนการทำงานและปัญหาของระบบปัจจุบัน.....	2
1.3 วัตถุประสงค์ของการพัฒนาระบบ.....	3
1.4 ขอบเขตของโครงการศึกษา.....	3
1.5 ทฤษฎีที่ใช้ในการวิเคราะห์และออกแบบระบบ.....	3
1.6 เทคโนโลยีที่ใช้ในการพัฒนา.....	4
1.7 ผลที่คาดว่าจะได้รับ.....	5
1.8 โครงสร้างเนื้อหาของเอกสาร.....	5
2. การพัฒนาระบบด้วยวิธีคอมโพเนนต์เบส.....	7
2.1 UML (Unified Modeling Language).....	7
2.2 การพัฒนาซอฟต์แวร์ในรูปแบบของคอมโพเนนต์.....	11
2.3 กระบวนการพัฒนา (Development Process).....	13
3. การวิเคราะห์ระบบ.....	26
3.1 ความต้องการของระบบ.....	26
3.2 การวิเคราะห์ความต้องการของระบบ.....	28
3.3 การวางแผนในการแบ่งการพัฒนาแบบเพิ่มทีละขั้น (Incremental Plan).....	36

## สารบัญ (ต่อ)

หน้า

4. การดำเนินการพัฒนาระบบ .....	38
4.1 รายละเอียดของ Use Case .....	38
4.2 การออกแบบคลาส .....	55
4.3 การออกแบบส่วนติดต่อผู้ใช้ .....	61
4.4 Design Sequence Diagram.....	61
4.5 State Diagram .....	67
4.6 Defining the Physical Architecture.....	69
5. การออกแบบฐานข้อมูล.....	72
5.1 แบบจำลองแสดงความสัมพันธ์ระหว่างเอนทิตี .....	72
5.2 การสร้างฐานข้อมูล .....	75
6. การสร้างระบบ .....	82
6.1 Round-Trip Engineering.....	82
6.2 ตัวอย่าง Source Code .....	84
6.3 การสร้าง COM+ Application จากคอม โปเนนท์.....	89
6.4 COM+ Application Deployment .....	91
6.5 ผลการทดสอบระบบ .....	92
7. สรุปผลการพัฒนา.....	93
บรรณานุกรม .....	95
ภาคผนวก	
ก. การติดตั้งระบบ .....	96
ก.1 การย้ายระบบจากคอมพิวเตอร์ที่ใช้พัฒนาไปยังคอมพิวเตอร์ที่ใช้งาน .....	96
ก.2 โครงสร้างเฉพาะทางซอฟต์แวร์ที่เหมาะสมต่อการใช้งานระบบ .....	100
ก.3 การติดตั้งส่วนประกอบต่างๆในเครื่องคอมพิวเตอร์ที่จะใช้งาน.....	100

## สารบัญ (ต่อ)

หน้า

ภาคผนวก

ข. การใช้งานระบบ.....	103
ข.1 ระบบงานในส่วนที่เป็น Windows Application.....	103
ข.2 ระบบงานในส่วนที่เป็น Web Application .....	117
ประวัติผู้เขียน.....	121



# สารบัญตาราง

	หน้า
ตารางที่	
2.1 ตัวอย่าง Decision Table สำหรับ Use Case ยืมสารนิเทศ.....	16
2.2 ตัวอย่างการแสดงทุกกรณีของเงื่อนไขไว้ในตารางตัดสินใจ .....	16
2.3 ตัวอย่าง Decision Table ที่มีเหตุการณ์ซ้ำซ้อนกันอยู่.....	17
2.4 ตัวอย่างการสร้าง Decision Table ของ Use Case ครอบคลุมในขั้นแรก.....	17
2.5 ตัวอย่าง Decision Table ชั้นสมบูรณ์ของ Use Case ครอบคลุม.....	18
2.6 ตัวอย่าง Decision Table สำหรับ Use Case รับชำระค่าปรับ.....	19
4.1 Decision Table สำหรับ Use Case - Log In .....	39
4.2 Decision Table สำหรับ Use Case - Add Member.....	40
4.3 Decision Table สำหรับ Use Case - Add Title.....	41
4.4 Decision Table สำหรับ Use Case - Add Item.....	42
4.5 Decision Table สำหรับ Use Case - Extended Expire Date.....	44
4.6 Decision Table สำหรับ Use Case - Pay Fine.....	45
4.7 Decision Table สำหรับ Use Case - Lend Item.....	46
4.8 Decision Table สำหรับ Use Case - Return Item.....	48
4.9 Decision Table สำหรับ Use Case - Make Reservation.....	49
4.10 Decision Table สำหรับ Use Case - Remove Reservation .....	51
4.11 Decision Table สำหรับ Use Case - Remove Member .....	52
4.12 Decision Table สำหรับ Use Case - Remove Title .....	53
4.13 Decision Table สำหรับ Use Case - Remove Item .....	54
5.1 รายละเอียดตารางผู้แต่ง.....	75
5.2 รายละเอียดตารางรายชื่อผู้แต่ง .....	75
5.3 รายละเอียดตารางประเภทหนังสือ .....	76
5.4 รายละเอียดตารางทะเบียนหนังสือ.....	76

## สารบัญตาราง (ต่อ)

	หน้า
ตารางที่	
5.5 รายละเอียดตารางคณะ .....	76
5.6 รายละเอียดตารางค่านำหน้าชื่อ .....	77
5.7 รายละเอียดตารางรายการยืมหนังสือ.....	77
5.8 รายละเอียดตารางสมาชิก .....	77
5.9 รายละเอียดตารางประเภทสมาชิก.....	78
5.10 รายละเอียดตารางข้อกำหนดการยืม.....	78
5.11 รายละเอียดตารางสำนักพิมพ์ .....	79
5.12 รายละเอียดตารางการจองหนังสือ.....	79
5.13 รายละเอียดตารางเจ้าหน้าที่.....	79
5.14 รายละเอียดตารางข้อมูลการทำงานของเจ้าหน้าที่ .....	80
5.15 รายละเอียดตารางหัวเรื่อง.....	80
5.16 รายละเอียดตารางรายการหัวเรื่องของหนังสือ.....	80
5.17 รายละเอียดตารางหนังสือ .....	81

# สารบัญภาพ

รูปที่	หน้า
2.1 ภาพรวมของขบวนการในการพัฒนาที่ใช้ในโครงการนี้.....	14
2.2 สถาปัตยกรรมของระบบ.....	22
3.1 Use Case Package ของระบบที่จะทำการพัฒนา.....	30
3.2 Use Case ของแพ็คเกจส่วนงานอำนวยความสะดวกทั่วไป .....	31
3.3 Use Case ของแพ็คเกจส่วนงานจัดการวัสดุสารนิเทศ .....	31
3.4 Use Case ของแพ็คเกจส่วนงานจัดการการยืม-คืนสารนิเทศ .....	32
3.5 Use Case ของแพ็คเกจส่วนงานจัดการสมาชิก .....	32
3.6 การขึ้นต่อกันของ Use Case ในการพัฒนาระบบ .....	36
4.1 Facade Class ในระบบ .....	55
4.2 ส่วนติดต่อ IMaint.....	56
4.3 ส่วนติดต่อ IViewer.....	56
4.4 Main Business Class ในระบบ .....	57
4.5 Data Access Class ที่ไม่รองรับทรานแซกชันในระบบ.....	58
4.6 Data Access Class ที่รองรับทรานแซกชันในระบบ .....	58
4.7 การดึงข้อมูลสถานะของวัตถุจากฐานข้อมูล .....	60
4.8 การบันทึกข้อมูลสถานะของวัตถุจากฐานข้อมูล .....	61
4.9 Sequence Diagram ของ Use Case Log In.....	62
4.10 Sequence Diagram ของ Use Case Add Member .....	62
4.11 Sequence Diagram ของ Use Case Add Title .....	62
4.12 Sequence Diagram ของ Use Case Add Item .....	63
4.13 Sequence Diagram ของ Use Case Browse Catalog.....	63
4.14 Sequence Diagram ของ Use Case Browse Member.....	64
4.15 Sequence Diagram ของ Use Case Extended Expire Date .....	64

## สารบัญภาพ (ต่อ)

รูปที่	หน้า
4.16 Sequence Diagram ของ Use Case Pay Fine .....	64
4.17 Sequence Diagram ของ Use Case Lend Item .....	65
4.18 Sequence Diagram ของ Use Case Return Item .....	65
4.19 Sequence Diagram ของ Use Case Make Reservation .....	66
4.20 Sequence Diagram ของ Use Case Remove Reservation .....	66
4.21 Sequence Diagram ของ Use Case Remove Member .....	66
4.22 Sequence Diagram ของ Use Case Remove Title .....	67
4.23 Sequence Diagram ของ Use Case Remove Item .....	67
4.24 State Diagram ของวัตถุสมาชิก .....	68
4.25 State Diagram ของวัตถุเลขทะเบียนหนังสือ .....	68
4.26 State Diagram ของวัตถุการขืม .....	69
4.27 State Diagram ของวัตถุการจอง .....	69
4.28 Component Diagram ของระบบ .....	70
4.29 การ Assign คลาสให้กับคอมโพเนนท์ .....	71
5.1 Context Data Model .....	73
5.2 Key-Base Data Model .....	74
6.1 ขบวนการ Round-Trip Engineering .....	82
6.2 การทำ Forward Engineering ของ LASFacades Component .....	83
6.3 โครงของโค้ดที่ถูกสร้างขึ้นจากคลาส fcdLoanMaint .....	83
6.4 ตัวอย่างโปรแกรมในส่วนของ การสร้าง Object และการเรียกใช้งาน Method .....	84
6.5 การกำหนดการอ้างอิงของคอมโพเนนท์ .....	85
6.6 ตัวอย่างโปรแกรมที่รองรับการทำงานในสภาพแวดล้อมของ Component Service .....	86
6.7 ตัวอย่างโปรแกรมที่แสดงการใช้งานอ็อบเจ็ค ObjectContext .....	88

## สารบัญภาพ (ต่อ)

รูปที่	หน้า
6.8 Component Services Management Console.....	89
6.9 รูปแบบของ COM+ Application ของระบบ .....	90
6.10 โครงสร้างของ COM+ Application ของระบบ.....	91
6.11 Deployment Diagram ของระบบ.....	92
ก.1 หลักการในการทำ Deployment.....	97
ก.2 การ Export COM+ Application.....	98
ก.3 Export Wizard ขั้นตอนที่ 1.....	98
ก.4 Export Wizard ขั้นตอนที่ 2.....	99
ก.5 เสร็จสิ้นการทำ Export Wizard .....	99
ก.6 ไฟล์ที่ได้หลังจากการทำ Export Wizard.....	99
ก.7 Registration Entry ของระบบ .....	101
ก.8 โครงสร้างของ Web Application.....	102
ข.1 หน้าจอ Log In เพื่อเข้าใช้งานระบบห้องสมุด.....	103
ข.2 ข้อความเตือนในกรณีไม่พบเจ้าหน้าที่ที่มีเลขประจำตัวนี้ในระบบ .....	104
ข.3 ข้อความเตือนในกรณีรหัสผ่าน ไม่ถูกต้อง.....	104
ข.4 หน้าจอหลักของระบบห้องสมุด .....	104
ข.5 รายละเอียดของเมนูส่วนการเข้าใช้งานระบบ .....	105
ข.6 รายละเอียดของเมนูส่วนการจัดการข้อมูลหลัก.....	105
ข.7 รายละเอียดของเมนูส่วนการให้บริการ .....	105
ข.8 หน้าจอแสดงข้อมูลการเข้าใช้งานระบบของเจ้าหน้าที่.....	106
ข.9 หน้าจอหลักของงานจัดการข้อมูลสมาชิก .....	107
ข.10 หน้าจอข้อมูลสมาชิก.....	107
ข.11 หน้าจอหลักข้อมูลประเภทสมาชิก.....	108

## สารบัญญภาพ (ต่อ)

รูปที่	หน้า
ข.12 หน้าจอหลักข้อมูลคณะ.....	108
ข.13 หน้าจอการชำระค่าปรับ.....	108
ข.14 หน้าจอข้อมูลประเภทสมาชิก .....	109
ข.15 หน้าจอหลักของงานจัดการข้อมูลหนังสือ .....	110
ข.16 หน้าจอข้อมูลหนังสือ .....	111
ข.17 หน้าจอหลักของข้อมูลประเภทหนังสือ .....	111
ข.18 หน้าจอหลักของข้อมูลสำนักพิมพ์ .....	112
ข.19 หน้าจอหลักของข้อมูลผู้แต่ง .....	112
ข.20 หน้าจอหลักของข้อมูลหัวเรื่อง.....	112
ข.21 หน้าจอข้อมูลเลขทะเบียนหนังสือ.....	113
ข.22 หน้าจอหลักของข้อมูลข้อกำหนดในการยืม .....	113
ข.23 หน้าจอการยืม คืนหนังสือ.....	114
ข.24 หน้าจอการยืม คืนหนังสือในกรณีที่สมาชิก ไม่มีสิทธิทำการยืม .....	115
ข.25 หน้าจอการยืม คืนหนังสือในกรณีที่สมาชิกมีสิทธิทำการยืม.....	115
ข.26 หน้าจอการจองหนังสือ .....	116
ข.27 หน้าจอการจองหนังสือในกรณีที่ทำการจองสำเร็จ .....	117
ข.28 เว็บเพจหน้าแรกของสำนักหอสมุด.....	118
ข.29 เว็บเพจผลการค้นหาหนังสือจากชื่อเรื่อง .....	118
ข.30 เว็บเพจผลการค้นหาหนังสือจากหัวเรื่อง .....	118
ข.31 เว็บเพจผลการค้นหาหนังสือจากชื่อผู้แต่ง.....	119
ข.32 เว็บเพจผลการค้นหาหนังสือจากเลขเรียกหนังสือ.....	119
ข.33 เว็บเพจผลตรวจสอบรายการยืมหนังสือของสมาชิก.....	120

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในการพัฒนาระบบสารสนเทศที่ผ่านมามีลักษณะเป็นแบบรวมศูนย์ (Centralize) เป็นส่วนใหญ่ และมีรูปแบบเป็นดั่งแบบจำลองที่เรียกว่า แบบจำลองแบบ 2 ชั้น (2-Tier) ที่แบ่งองค์ประกอบของระบบเป็น 2 ส่วนคือ ส่วนของผู้ใช้ (Client) และส่วนที่ให้บริการ (Server) ซึ่งในขณะที่ยังดำเนินงานนั้น ส่วนของผู้ใช้จะเชื่อมต่ออยู่กับส่วนที่ให้บริการซึ่งโดยมากจะเป็นเซิร์ฟเวอร์ฐานข้อมูลอยู่ตลอดเวลา เพื่อให้ทรัพยากรของเซิร์ฟเวอร์ อย่างเช่น หน่วยความจำและการประมวลผล ซึ่งในปัจจุบันพบว่า ระบบที่มีรูปแบบเช่นนี้มีแนวโน้มที่จะทำให้เกิดการสูญเสียในด้านทรัพยากร เนื่องจากปล่อยให้ส่วนของผู้ใช้ทำการครอบครองทรัพยากรแม้แต่ในขณะที่ยังไม่ได้ใช้งาน ดังนั้น หากเราทำการพิจารณาในประเด็นของการใช้งานทรัพยากรให้ได้คุ้มค่าที่สุด เราจึงต้องให้ความสนใจกับการพัฒนาระบบในรูปแบบที่ต่างออกไป ซึ่งก็คือรูปแบบของแบบจำลองแบบ 3 ชั้น (3-Tier) ซึ่งได้รับการยอมรับว่ามีประสิทธิภาพในด้านต่างๆ (Sundblad, S. and Sundblad, P. 2000) เช่น

1. ความคุ้มค่าในการควบคุมการใช้ทรัพยากร
2. ความเหมาะสมในการควบคุมดูแลรักษากฎเกณฑ์ทางธุรกิจ (Business Rule)
3. เหมาะสมกับแอปพลิเคชันที่ทำงานบนระบบเครือข่ายอินเทอร์เน็ตที่เติบโตอย่างรวดเร็ว

ในปัจจุบัน

นอกจากประเด็นในด้านของรูปแบบของระบบ ซึ่งอาจจะเรียกว่าเป็นสถาปัตยกรรมของระบบ แล้วสิ่งที่สำคัญอีกประการหนึ่งในการพัฒนาแอปพลิเคชันคือ โครงสร้างพื้นฐาน (Infrastructure) ของระบบ ซึ่งสนับสนุนการทำงานในสถาปัตยกรรมนั้น ซึ่งสิ่งนี้เองที่ทำให้การพัฒนาแอปพลิเคชันบนพื้นฐานของการใช้งานคอมพิวเตอร์เข้ามามีบทบาทในการพัฒนาระบบด้วยปัจจัยต่างๆดังที่กล่าวมาทั้งหมดนี้ ทำให้เกิดแรงกระตุ้นในการพัฒนาระบบสารสนเทศในรูปแบบที่กล่าวมา โดยในโครงการพัฒนาระบบสารสนเทศนี้จะได้ยกรณีสึกษาที่เป็นการพัฒนาแบบสารสนเทศสำหรับการบริหารงานห้องสมุดของมหาวิทยาลัยเอริกซ์ขึ้นมาเป็นกรณีศึกษา เนื่องจากมีปัญหาของการทำงานของระบบในปัจจุบันที่สามารถทำการแก้ไขโดยการประยุกต์ใช้แนวทางที่ได้กล่าวมาได้ ทั้งนี้ในการวิเคราะห์และออกแบบระบบสำหรับโครงการนี้ จะใช้แนวทางการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และออกแบบระบบเชิงวัตถุซึ่งมีวิธีการดังที่จะได้กล่าวต่อไปในหัวข้อกรรมวิธีในการพัฒนาระบบ เนื่องจากเป็นรูปแบบการพัฒนาที่สนับสนุนการใช้แนวทางที่กล่าวมาในการพัฒนาระบบ โดยในการพัฒนาจะใช้ UML (Unified Modeling Language) ในการสร้างแบบจำลองของระบบ

## 1.2 ขั้นตอนการทำงานและปัญหาของระบบปัจจุบัน

ปัจจุบันการทำงานของระบบงานห้องสมุดของห้องสมุดแต่ละแห่งนั้นจะกระทำโดยใช้แอปพลิเคชันที่ห้องสมุดแต่ละแห่งพัฒนาขึ้นมาเอง โดยมีฐานข้อมูลเป็นของตนเอง และแต่ละห้องสมุดไม่สามารถเข้าถึงข้อมูลของกันและกันได้

หน้าที่การทำงานของระบบงานโดยรวม สามารถจำแนกได้ตามส่วนงาน ดังนี้คือ

- 1). ส่วนงานการยืม-คืนหนังสือ ข้อมูลของการยืมหนังสือหรือสื่อจะถูกเก็บบันทึกรายการการยืม-คืน ลงในแอปพลิเคชัน
- 2). ส่วนงานสมาชิก ข้อมูลสมาชิกที่มีในระบบจะมีเพียงประเภทเดียว คือ สมาชิกภายในมหาวิทยาลัย ซึ่งจะทำกรจัดการเก็บข้อมูลโดยใช้ข้อมูลนักศึกษาที่มีการลงทะเบียนในแต่ละเทอม ซึ่งได้รับข้อมูลดังกล่าวจากฝ่ายทะเบียน นักศึกษาของมหาวิทยาลัย และข้อมูลบุคลากรของมหาวิทยาลัย แล้วทำการบันทึกลงในแอปพลิเคชัน
- 3). ส่วนงานการลงทะเบียนสื่อ จะทำการบันทึกข้อมูลของหนังสือลงในแอปพลิเคชัน
- 4). ส่วนงานการจอง ในปัจจุบันยังไม่มีบริการในส่วนนี้
- 5). ส่วนงานการค้นหาสื่อ จะทำการค้นหาข้อมูลของหนังสือจากฐานข้อมูลของแอปพลิเคชัน

จากการศึกษาถึงปัญหาต่างๆที่เกิดขึ้นในระบบงานปัจจุบันพบว่าขั้นตอนการทำงานของระบบงานห้องสมุดนั้น ไม่มีประสิทธิภาพเท่าที่ควรแม้ว่าจะมีการนำเอาระบบคอมพิวเตอร์เข้ามาช่วยงาน เนื่องจากลักษณะของแอปพลิเคชันที่พัฒนาขึ้นมา จะเป็นการทำงานแบบบนเครื่องคอมพิวเตอร์เครื่องเดียว (Stand Alone) และจากการศึกษาสามารถสรุปปัญหาของระบบการทำงานในปัจจุบัน ได้ดังนี้คือ ระบบในปัจจุบันไม่รองรับการทำงานในลักษณะของระบบเครือข่าย กล่าวคือในกรณีที่มีห้องสมุดมีหลายๆแห่ง(สาขา) และผู้ใช้งานต้องการเรียกดูข้อมูลรายการสารนิเทศของห้องสมุดแต่ละแห่งได้โดยไม่ต้องเดินทางไปใช้ค้นหาที่ห้องสมุดสาขานั้นๆ ทำให้เกิดความต้องการให้ระบบสามารถให้บริการในรูปแบบออนไลน์ (Online System) นั่นคือ สามารถทำงานผ่านทางระบบเครือข่าย ซึ่งระบบงานในปัจจุบันยังไม่รองรับการทำงานดังที่กล่าวมาแล้ว ข้างต้น

### 1.3 วัตถุประสงค์ของการพัฒนาระบบ

ระบบงานห้องสมุด ถูกพัฒนาขึ้นเพื่อให้สามารถทำงานบนระบบเครือข่ายได้อย่างมีประสิทธิภาพ โดยประยุกต์ใช้แนวทางการออกแบบในลักษณะที่แบ่งการบริการของระบบออกเป็นชั้น (Layer) และการพัฒนาระบบบนพื้นฐานของการใช้คอมพิวเตอร์ ทั้งนี้เพื่อให้ระบบมีความยืดหยุ่นต่อการเปลี่ยนแปลงส่วนใด ๆ ของระบบในอนาคต ซึ่งถือเป็นวัตถุประสงค์หลักในการพัฒนาโครงการในครั้งนี้

### 1.4 ขอบเขตของโครงการศึกษา

ในการพัฒนาระบบงานห้องสมุดในส่วนที่ได้ทำการศึกษาไว้ จะเน้นในด้านประเด็นของการนำเอาสถาปัตยกรรมของระบบและการใช้งานซอฟต์แวร์คอมพิวเตอร์เพื่อช่วยในการพัฒนาระบบที่มีความยืดหยุ่นต่อการเปลี่ยนแปลง และสามารถจัดการการใช้ทรัพยากรได้อย่างมีประสิทธิภาพ ดังนั้นเพื่อให้เหมาะสมกับเวลาในการพัฒนาระบบ จึงจำเป็นต้องพัฒนาในส่วนที่มีความสำคัญมากที่สุดก่อน แล้วขยายขอบเพิ่มเติมภายหลัง โดยจะทำการกำหนดขอบเขตในการศึกษาแต่ละส่วนงาน ไว้ดังต่อไปนี้ คือ

- 1). งานการยืม-คืนหนังสือ บันทึกรายการยืม-คืนหนังสือ
- 2). งานสมาชิก จัดเก็บข้อมูลของสมาชิกลงในฐานข้อมูล
- 3). งานการลงทะเบียนสื่อ บันทึกและจัดการข้อมูลของหนังสือที่มีอยู่ในห้องสมุด
- 4). งานการจอง สมาชิกสามารถทำรายการจองสารนิเทศได้โดยมีข้อกำหนดคือ สมาชิก 1 คน สามารถทำรายการจองสารนิเทศได้ไม่จำกัดจำนวน เมื่อสมาชิกผู้จองได้รับสิทธิ์ในสื่อนั้นๆ ระบบจะแจ้ง ไปยังสมาชิกผ่านทางระบบอีเมลล์หรือทางโทรศัพท์ มีการกำหนดระยะเวลาในการรับสื่อ ซึ่งถ้าพ้นกำหนดการรับสื่อแล้ว สิทธิ์นั้นจะตกสู่ผู้รับสิทธิ์ในลำดับถัดไป
- 5). งานการค้นหาสื่อ ผู้ใช้สามารถทำการสืบค้นสารนิเทศหรือข้อมูลอื่นๆ ได้โดยการเรียกผ่านเว็บเบราว์เซอร์

### 1.5 ทฤษฎีที่ใช้ในการวิเคราะห์และออกแบบระบบ

ในการวิเคราะห์และออกแบบระบบงานห้องสมุด เป็นขั้นตอนหนึ่งในวงจรของการพัฒนาระบบ หรือเรียกว่า SDLC (System Development Life Cycle) โดยในการพัฒนาจะใช้ Object-Oriented Development Process เป็นแบบจำลองของวิธีการในการพัฒนาระบบ (Process Model) และใช้ UML (Unified Modeling Language) เป็นภาษาในการสร้างแบบจำลองในระหว่างการ

ดำเนินการพัฒนาระบบ รวมไปถึงการประยุกต์ใช้การพัฒนาระบบที่มีพื้นฐานบนการใช้งานคอมพิวเตอร์และการใช้สถาปัตยกรรมแบบ 3 ชั้นในระหว่างการออกแบบระบบด้วย

## 1.6 เทคโนโลยีที่ใช้ในการพัฒนา

หลังจากที่ได้ทำการศึกษาถึงปัญหา เหตุผล และทฤษฎีต่างๆ รวมทั้งสภาวะแวดล้อมการทำงานของระบบงานปัจจุบันและการศึกษาความเป็นไปได้ของระบบใหม่ ทำให้สามารถหว่านการสรุปรายละเอียดเกี่ยวกับเครื่องมือที่จะใช้ในการพัฒนาแอปพลิเคชัน รวมไปถึงเทคโนโลยีที่ใช้ร่วมกันในการพัฒนาได้ดังนี้คือ

- 1) ในการวิเคราะห์และออกแบบระบบ ได้เลือก CASE (Computer Aided Software Engineering) Tool ที่เป็นของบริษัท Rational, Inc. คือ Rational Rose 2000 ซึ่งสามารถรายละเอียดและข้อมูลเพิ่มเติมได้ที่ <http://www.rational.com>
- 2) ระบบฐานข้อมูล que เลือกใช้ คือ ระบบฐานข้อมูล Microsoft SQL Server ของไมโครซอฟท์
- 3) เครื่องมือที่เลือกใช้ในการพัฒนาส่วนของคอมพิวเตอร์เน็ตและในส่วนติดต่อผู้ใช้ (User Interface) คือ Microsoft Visual Basic 6.0
- 4) ในส่วนของกาให้บริการต่างๆที่ฝั่ง Server นั้น การให้บริการกับส่วนติดต่อผู้ใช้ที่เป็น Web Browser นั้น ได้เลือก ใช้ Microsoft Internet Information Server (IIS 5.0) ที่ทำงานบน Microsoft Windows 2000 และใช้ Active Server Pages ซึ่งเป็น Server Side Script ในการเป็นตัวกลางติดต่อระหว่างส่วนติดต่อผู้ใช้ที่เป็น Web Browser กับคอมพิวเตอร์เน็ตต่างๆในระบบ โดยในส่วนของการจัดการคอมพิวเตอร์เน็ตเลือกใช้เทคโนโลยี COM+ (Component Object Model) ของไมโครซอฟท์เป็นเทคโนโลยีที่รองรับการทำงานของระบบ
- 5) สำหรับแพลตฟอร์ม (Platform) ทั้งหมดที่ใช้ในระบบจะเป็นของไมโครซอฟท์ เนื่องจากเป็นแพลตฟอร์มที่มีอยู่ในสภาพแวดล้อมที่จะดำเนินการพัฒนาอยู่แล้ว นอกจากนั้นยังสนับสนุนการพัฒนาในระบบในรูปแบบหลายชั้น ซึ่งเป็นสถาปัตยกรรมของระบบที่เราเลือกใช้ด้วย

## 1.7 ผลที่คาดว่าจะได้รับ

หลังจากที่ได้พัฒนาระบบขึ้นมา เพื่อแก้ไขปัญหาที่เกิดขึ้นและประยุกต์ใช้ข้อดีต่างๆของเทคโนโลยีที่นำมาใช้ในการพัฒนาระบบ ทำให้สามารถสรุปถึงสิ่งที่คาดว่าจะจะเป็นประโยชน์ดังต่อไปนี้ คือ

- 1) เนื่องจากระบบที่ใช้ในห้องสมุดแต่ละแห่งเป็นระบบเดียวกัน ดังนั้นจึงสามารถใช้ข้อมูลร่วมกันได้ เป็นการใช้ทรัพยากรให้เกิดประโยชน์มากที่สุด
- 2) สามารถปรับปรุงส่วนใดๆของระบบ (โดยเฉพาะส่วนติดต่อผู้ใช้ที่ค่อนข้างจะมีการเปลี่ยนแปลงบ่อยครั้ง) ได้โดยที่ไม่กระทบกระเทือนต่อส่วนอื่นๆของระบบ ซึ่งข้อดีนี้สามารถทำให้ระบบสามารถให้บริการแก่ผู้ใช้ในรูปแบบใหม่ๆ เช่น ให้บริการกับผู้ใช้ผ่านทางระบบเครือข่ายอินเทอร์เน็ตโดยที่ใช้เพียงโครงสร้างพื้นฐานเดิมของระบบ ไม่ต้องเพิ่มเติมส่วนใหม่ๆเข้าไปหรือแก้ไขระบบเก่าเลย
- 3) รองรับการทำงานในลักษณะของระบบเครือข่าย ทำให้การประมวลผลในบางขั้นตอนสามารถทำได้แบบออนไลน์
- 4) เนื่องจากการพัฒนาได้นำเอาหลักการของการพัฒนาซอฟต์แวร์ในรูปแบบของคอมโพเนนท์ ทำให้เป็นการสนับสนุนการนำเอาสิ่งที่ได้ทำการพัฒนาไปใช้ใหม่ (Reusability)

## 1.8 โครงสร้างเนื้อหาของเอกสาร

สำหรับเนื้อหาในโครงการพัฒนาระบบงานจะประกอบด้วยเนื้อหาต่างๆดังต่อไปนี้ ในบทแรกจะกล่าวถึงความเป็นมา ความสำคัญและวัตถุประสงค์ในการพัฒนาระบบงาน บทที่ 2 จะกล่าวถึงแนวความคิดและหลักการต่างๆที่เกี่ยวข้องในการพัฒนาระบบงาน บทที่ 3 กล่าวถึงการวิเคราะห์ระบบโดยใช้รูปแบบเชิงวัตถุ โดยเฉพาะอย่างยิ่งการใช้งาน use case ในการรวบรวมความต้องการของระบบ บทที่ 4 จะเป็นเนื้อหาในส่วนที่เกี่ยวกับขั้นตอนในการดำเนินการพัฒนาระบบ โดยจะเป็นการประยุกต์ใช้แนวทางการพัฒนาในรูปแบบของคอมโพเนนท์เบสในการดำเนินการพัฒนาร่วมกับการใช้สิ่งที่ได้จากการวิเคราะห์ในบทที่ 3 เพื่อทำการออกแบบส่วนประกอบพื้นฐานที่สำคัญเพื่อจะนำไปใช้ในส่วนต่างๆในสถาปัตยกรรมของระบบ ซึ่งการพัฒนาจะถูกดำเนินการในรูปแบบของการทำซ้ำและเพิ่มทีละขั้น (Iteration and Increment) ในบทที่ 5 จะกล่าวถึงการออกแบบส่วนที่ใช้ในการจัดเก็บข้อมูลของระบบ ซึ่งจะดำเนินการออกแบบโดยใช้หลักการของแบบจำลอง Entity Relationship บทที่ 6 จะกล่าวถึงการดำเนินการสร้างระบบโดยการนำสิ่งที่ได้จากการออกแบบมาทำการสร้างโครงสร้างของชุดคำสั่ง เพื่อที่จะทำการเพิ่มเติมชุดคำสั่งดังกล่าวจนกระทั่งแอปพลิเคชันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาสามารถทำงานได้ตามที่ทำการออกแบบไว้แล้วทำการแก้ไขแบบจำลองเพื่อเพิ่มเติมส่วน  
 ที่ได้ทำการเพิ่มเติมชุดคำสั่งต่างๆลงไป ซึ่งขบวนการต่างๆที่กล่าวมานี้เรียกรวมๆว่า Round-Trip  
 Engineering ซึ่งเป็นขบวนการที่สนับสนุนการพัฒนาในรูปแบบของการทำซ้ำและเพิ่มทีละขั้น  
 (Iteration and Increment) และทำให้การพัฒนาสามารถดำเนินไปอย่างต่อเนื่องและสามารถติดตาม  
 ขั้นตอนต่างๆในการพัฒนาได้ง่าย และสุดท้ายจะเป็นการกล่าวถึงผลของการทดสอบระบบและ  
 ความคิดเห็นต่างๆ ในบทที่ 7 จะเป็นการสรุปในส่วน of โครงการพัฒนาระบบนี้ โดยจะสรุปเนื้อหา  
 โดยรวมของการพัฒนาโครงการและความคิดเห็นต่างๆที่มีต่อการพัฒนาโครงการ และในส่วน  
 ของภาคผนวกจะแบ่งเป็น 2 ส่วนคือ ภาคผนวก ก ซึ่งจะกล่าวถึงการติดตั้งระบบ และภาคผนวก ข  
 จะกล่าวถึงคู่มือในการใช้งานระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### การพัฒนาระบบด้วยวิธีคอมโพเนนท์เบส

ในการดำเนินการพัฒนาระบบสารสนเทศเพื่อการบริหารห้องสมุดนี้ ได้ใช้หลักการและเครื่องมือหลายๆอย่างร่วมกันในการพัฒนา โดยหลักการที่สำคัญที่ถูกนำมาใช้ได้แก่ การพัฒนาซอฟต์แวร์ในรูปแบบคอมโพเนนท์ รวมไปถึงการใช้แบบจำลองของระบบเป็นพื้นฐานในการดำเนินการพัฒนาซึ่ง โดยใช้ UML เป็นเครื่องมือที่นำมาใช้สร้างแบบจำลองของระบบ ซึ่งรายละเอียดของเครื่องมือและกระบวนการต่างๆที่ถูกนำมาใช้ในการดำเนินการพัฒนาระบบนั้นมีดังต่อไปนี้

#### 2.1 UML (Unified Modeling Language)

##### 2.1.1) ความสำคัญของภาษาที่ใช้ในการสร้างแบบจำลอง (Eriksson, H. and Penker, M. 1998)

ในงานทางวิศวกรรม โดยทั่วไปนั้น เมื่อมีความต้องการที่จะสร้างสิ่งใดขึ้นมาแบบ (Drawing) จะมีบทบาทสำคัญต่อการสร้างสิ่งนั้น ซึ่งแบบนี้จะถูกใช้โดยผู้ที่มีบทบาทเกี่ยวข้องกับการพัฒนาสิ่งของนั้นที่อาจมีได้หลายบทบาทในโครงการหนึ่ง ทำให้แบบนี้สามารถถูกแบ่งย่อยออกไปได้ตามรายละเอียดที่เพิ่มขึ้นในแต่ละระดับที่ถูกแบ่งออกมา เพื่อให้เหมาะสมกับผู้ที่จะนำมันไปใช้งาน นอกจากนี้ แบบจำลองยังมีบทบาทที่สำคัญในงานด้านอื่นๆอีกเช่น การวางแผนในด้านเงินทุน เวลา การกระจายงาน และการกระจายทรัพยากร ซึ่งล้วนแต่เป็นกิจกรรมที่ดำเนินการอยู่บนพื้นฐานของข้อมูลที่มาจากแบบ ทั้งสิ้น

จากตัวอย่างที่ได้กล่าวมาทั้งหมดนี้ จะเห็นได้ว่าที่จริงแล้วแบบ ก็เปรียบได้กับแบบจำลอง (Model) ของสิ่งที่จะทำการสร้างนั่นเอง แบบจำลองจะอธิบายถึงภาพพจน์และพฤติกรรมของสิ่งนั้น โดยสิ่งที่กล่าวถึงอาจจะมืออยู่แล้ว กำลังได้รับการพัฒนา หรือเป็นแค่แผนการที่อยู่ระหว่างการพิจารณา

ในระหว่างกระบวนการสร้างแบบจำลอง (Modeling) นั้น ผู้ออกแบบจะต้องทำการเก็บรวบรวมข้อมูลและความต้องการ เพื่อที่จะทำการสร้างผลิตภัณฑ์หรือระบบนั้น โดยข้อมูลดังกล่าวอาจจะอยู่ในมุมมองของหน้าที่ (Functionality) ลักษณะที่ปรากฏ (Appearance) ประสิทธิภาพ (Performance) และความน่าเชื่อถือ (Reliability) ดังนั้น จึงเป็นหน้าที่ของผู้ออกแบบที่จะต้องสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองที่สามารถอธิบายในมุมมองเหล่านี้ได้ ดังนั้นแบบจำลองที่ใช้จึงจำเป็นต้องสามารถที่จะแยกย่อยลงไปได้เพื่อที่จะสามารถอธิบายในมุมมองที่แตกต่างกันของการพัฒนาผลิตภัณฑ์หรือระบบที่พิจารณาอยู่ได้

โดยทั่วไปแล้วแบบจำลองนิยามที่จะถูกแสดงออกในรูปของภาษาที่แสดงด้วยภาพ (Visual Language) ในที่นี้หมายถึงภาษาที่เราเห็นอยู่ในรูปของสัญลักษณ์และความสัมพันธ์ระหว่างแต่ละสัญลักษณ์ นั่นเอง การที่แบบจำลองสามารถถูกอธิบายได้ในลักษณะนี้ทำให้การสื่อสารระหว่างผู้ที่เกี่ยวข้องกับแบบจำลองนั้นสามารถทำได้ง่ายขึ้น แต่อย่างไรก็ตาม ไม่จำเป็นที่ทุกสิ่งทุกอย่างที่ใช้ในการอธิบายแบบจำลองจะต้องอยู่ในรูปสัญลักษณ์ทั้งหมด ในบางครั้งข้อมูลบางอย่างในแบบจำลองก็สามารถทำการอธิบายได้โดยใช้เพียงข้อความธรรมดาเท่านั้น โดยสิ่งที่สำคัญคือแบบจำลองที่ดีนั้นจะต้องมีลักษณะดังต่อไปนี้

1. มีความถูกต้อง (Accuracy) คือสามารถที่จะอธิบายสิ่งที่จะทำการพัฒนาได้ถูกต้องในทุกๆด้าน
2. มีความสอดคล้องกัน (Consistency) ในกรณีที่แบบจำลองนั้นมีหลายรูปแบบตามมุมมองที่แตกต่างกันออกไป แบบจำลองเหล่านั้นจะต้องถูกอธิบายไปในทิศทางเดียวกันโดยไม่มี ความขัดแย้งซึ่งกันและกัน
3. ง่ายต่อการสื่อสาร
4. สามารถได้รับการปรับเปลี่ยนได้ง่าย
5. เข้าใจได้ง่าย

เมื่อได้เห็นถึงความสำคัญของแบบจำลองแล้ว มาพิจารณาถึงปัญหาที่เกิดขึ้นในวงการอุตสาหกรรมซอฟต์แวร์ เช่น การพัฒนาซอฟต์แวร์หรือระบบที่ไม่ตรงตามความต้องการของผู้ใช้งานอย่างแท้จริง หรือการที่ไม่สามารถที่จะทำการควบคุมการพัฒนาซอฟต์แวร์หรือระบบให้อยู่ในกำหนดการที่ได้ทำการกำหนดไว้ตั้งแต่แรกได้ ซึ่งปัญหาต่างๆเหล่านี้เอง ที่ได้ผลักดันให้วงการอุตสาหกรรมซอฟต์แวร์มีการสร้างจำลองก่อนที่จะดำเนินการพัฒนาซอฟต์แวร์ เช่นที่ในวงการวิศวกรรมอื่นๆได้ทำกันมาเป็นเวลานานแล้ว ซึ่งสิ่งนี้ทำให้เกิดความต้องการภาษาที่ใช้ในการสร้างแบบจำลอง (Modeling Language) ขึ้น

### 2.1.2) ภาพรวมของ UML

สิ่งที่สำคัญของแบบจำลองคือมันจะต้องมีจุดมุ่งหมายที่ชัดเจนในการแสดงถึงสิ่งที่มันเป็นแบบจำลองให้ ถ้าแบบจำลองขาดจุดนี้ไป มันจะไม่สามารถทำให้ผู้ที่ใช้งานมันสามารถรับรู้ถึงเหตุ

ผลและวิธีการที่จะใช้ เราจะสามารถแสดงแบบจำลองออกมาได้โดยการใช้ ภาษาในการสร้างแบบจำลอง ซึ่งประกอบด้วย

1. สัญลักษณ์ (Notation) ใช้ในการอธิบายส่วนต่างๆในสิ่งที่แบบจำลองได้อธิบาย

2. กฎเกณฑ์ต่างๆ (Rule) เป็นข้อกำหนดถึงวิธีการใช้งาน โดยจะแบ่งออกเป็น

2.1. กฎเกณฑ์ในทางโครงสร้าง (Syntax Rules) จะบอกเราถึงการใช้สัญลักษณ์ต่างๆในการแสดงสิ่งที่เราจะสร้างแบบจำลองได้อย่างไร และเราจะทำการรวบรวมสัญลักษณ์ต่างๆเหล่านั้นเพื่อทำการแสดงถึงสิ่งที่เราจะทำการจำลองได้อย่างไร สิ่งนี้เปรียบเทียบกับคำในภาษาธรรมชาติที่มนุษย์ใช้กัน ซึ่งเราจะต้องรู้ว่าจะต้องใช้งานคำแต่ละคำอย่างไรจึงจะถูกต้อง และจะสร้างประโยคขึ้นมาโดยใช้คำแต่ละคำมารวมกันได้อย่างไร

2.2. กฎเกณฑ์ในทางความหมาย (Semantic Rules) จะบอกเราถึงความหมายของสัญลักษณ์แต่ละตัวที่เราใช้งานว่า มันสามารถถูกแปลความหมายโดยตัวมันเองได้อย่างไรและถูกแปลความหมายเมื่อพิจารณาจากมุมมองของสัญลักษณ์อื่นๆได้อย่างไร สิ่งนี้เปรียบเทียบกับความหมายของคำแต่ละคำในภาษาธรรมชาติที่มนุษย์ใช้กัน

2.3. กฎเกณฑ์ในทางการใช้งาน (Pragmatic Rules) จะบอกเราถึงการใช้สร้างแบบจำลองเพื่อให้ตรงกับเป้าหมายและทำให้ผู้ที่ดูมันสามารถเข้าใจมันได้ สิ่งนี้เปรียบเทียบกับวิธีการในการสร้างประโยคให้ผู้อื่นเข้าใจได้ง่าย หรือรูปแบบในการเขียนหนังสือในภาษาธรรมชาติที่มนุษย์ใช้กัน

เนื่องจากภาษาในการสร้างแบบจำลองหนึ่งๆจะต้องมีกฎเกณฑ์ต่างๆที่ชัดเจน ทำให้ผู้ที่ต้องการสร้างแบบจำลองจำเป็นต้องศึกษาถึงกฎเกณฑ์ต่างๆเหล่านี้ให้เข้าใจก่อนที่จะทำการพัฒนาแบบจำลองขึ้นมา ซึ่งโดยส่วนมากแล้วภาษาที่ใช้ในการสร้างแบบจำลองจะมีกฎเกณฑ์ในทางโครงสร้างและความหมายค่อนข้างชัดเจน ส่วนกฎเกณฑ์ในทางการใช้งานนั้นค่อนข้างยากที่จะทำให้อยู่ในรูปแบบที่สามารถนำไปใช้งานได้ง่าย ดังนั้นส่วนมากจะเห็นกันในรูปของ คำแนะนำ (Guideline) มากกว่า สิ่งสำคัญอีกประการที่ต้องคำนึงถึงในการใช้ภาษาในการสร้างแบบจำลองคือ ภาษาในการสร้างแบบจำลองก็เหมือนกับภาษาในธรรมชาติทั่วไป การที่จะสร้างแบบจำลองได้ดีหรือไม่ ขึ้นอยู่กับตัวผู้เขียนหรือผู้สร้างแบบจำลองด้วย เช่นในการเขียนงานเขียนขึ้นมาโดยใช้ภาษาธรรมชาติซึ่งเป็นเพียงเครื่องมือที่ใช้ในการสร้างงานเขียนเท่านั้น ส่วนงานที่เขียนจะดีหรือไม่ ขึ้นอยู่กับผู้แต่ง

UML (Unified Modeling Language) เป็นภาษาในการสร้างแบบจำลองที่ถูกสร้างขึ้นมาให้สามารถรองรับการใช้งานที่หลากหลาย มันอาจถูกใช้ในการสร้างแบบจำลองทางธุรกิจ (Business Modeling) หรือแบบจำลองของซอฟต์แวร์ในทุกๆระยะของการพัฒนา หรือในระบบอื่นๆ ซึ่งกล่าว

โดยรวมๆ ได้ว่ามันสามารถถูกใช้ในการสร้างแบบจำลองใดๆก็ตามที่ต้องการแสดงถึง โครงสร้างที่มีลักษณะคงที่ (Static Structure) และพฤติกรรมที่มีลักษณะที่ไม่คงที่ (Dynamic Behavior) ของสิ่งนั้น

UML มีส่วนประกอบย่อยๆ คือ มุมมอง แผนภาพ องค์ประกอบย่อยของแบบจำลองและส่วนเพิ่มเติม (Eriksson, H. and Penker, M. 1998) ซึ่งแต่ละส่วนมีรายละเอียดดังต่อไปนี้

1. มุมมอง (View) เป็นสิ่งที่ใช้แสดงให้เห็นถึงระบบที่จะทำการสร้างแบบจำลองในด้านต่างๆ โดยในแต่ละมุมมองจะถูกอธิบายโดยใช้แผนภาพ ซึ่งจะได้กล่าวถึงต่อไป นอกจากนี้ มุมมองยังถูกใช้เป็นส่วนที่เชื่อมระหว่างภาษาที่ใช้ในการสร้างแบบจำลอง (คือตัว UML เอง) เข้ากับวิธีที่ใช้ในการพัฒนา การสร้างแบบจำลองที่สามารถแสดงทุกอย่างเอาไว้ในรูปแบบเพียงหนึ่งเดียวของแบบจำลองเป็นสิ่งที่เราต้องการ แต่ในความเป็นจริงเราไม่สามารถทำเช่นนั้นได้ เนื่องจากระบบสามารถถูกมองดูได้ในหลายๆด้าน ไม่ว่าจะเป็น ด้านที่เกี่ยวกับหน้าที่ของระบบ (Functionality) ซึ่งได้แก่ สิ่งที่เป็นโครงสร้างของระบบซึ่งมีลักษณะที่ไม่มีการเปลี่ยนแปลง (Static) และ พฤติกรรม (Behavior) ของระบบที่เป็นสิ่งที่มีเปลี่ยนแปลงตลอดเวลา (Dynamic) หรือในด้านการเกี่ยวกับสิ่งที่อาจไม่ใช่หน้าที่ของระบบโดยตรง (Non Functionality) เช่น ประสิทธิภาพหรือความน่าเชื่อถือและด้านการเกี่ยวกับโครงสร้างการทำงานต่างๆ (Organization) เช่นการแปลงรูปแบบจากแบบจำลองไปเป็นส่วนของโปรแกรม (Mapping to code module) สิ่งเหล่านี้ทำให้ระบบจำเป็นที่จะต้องสามารถถูกอธิบายในมุมมองหลายๆมุมมอง ใน UML แต่ละมุมมอง สามารถถูกแสดงออกมาได้โดยใช้แผนภาพ 1 หรือมากกว่า 1 แผนภาพ และเช่นเดียวกัน แผนภาพ 1 แผนภาพอาจจะแสดงถึงมุมมองได้มากกว่า 1 มุมมอง โดยภาพรวมของระบบสามารถมองได้โดยใช้การมองระบบในแต่ละมุมมองร่วมกัน ใน UML จะมีประกอบด้วยมุมมองทั้งสิ้น 5 มุมมอง โดยสำหรับแต่ละมุมมองมีความหมายดังต่อไปนี้

1.1. Use Case View เป็นมุมมองที่แสดงถึงหน้าที่ของระบบที่ถูกมองโดยผู้ที่ใช้ระบบซึ่งอยู่ภายนอกขอบเขตของระบบ

1.2. Logical View เป็นมุมมองที่บอกเราว่าหน้าที่ของระบบนั้นถูกออกแบบอย่างไรเมื่อพิจารณาภายในตัวระบบ โดยจะถูกแสดงในรูปของโครงสร้างและ พฤติกรรมของระบบ

1.3. Component View เป็นมุมมองที่แสดงถึงโครงสร้างในส่วนขององค์ประกอบที่เป็นโปรแกรม (Code Component)

1.4. Concurrency View เป็นมุมมองที่แสดงการพิจารณาระบบในส่วนที่ละเอียดลงไปถึงระดับของ Process และการทำงานของ Processors ซึ่งมุมมองนี้ถือว่าเป็นมุมมองในด้านการเกี่ยวกับสิ่งที่อาจไม่ใช่หน้าที่ของระบบโดยตรง (Non Functionality)

**1.5. Deployment View** เป็นมุมมองที่แสดงถึงหน้าที่ของระบบที่ถูกมองโดยผู้ที่ใช้ระบบ ซึ่งอยู่ภายนอกขอบเขตของระบบ

2. **แผนภาพ (Diagram)** เป็นสิ่งที่ใช้แสดงให้เห็นถึงเนื้อหาที่อยู่ภายในแต่ละมุมมอง UML มีแผนภาพถึง 9 รูปแบบให้ใช้ และสามารถใช่เพียงแผนภาพเดี่ยวๆหรือใช้ร่วมกันในการอธิบายถึงมุมมองในแต่ละมุมมองได้

3. **องค์ประกอบย่อยของแบบจำลอง (Model Element)** เป็นสิ่งที่ใช้แสดงให้เห็นถึงส่วนที่เป็นแนวความคิดหลักๆที่แสดงถึง แนวความคิดในทาง Object-Oriented ที่เรียกว่า “Concept” เช่น Class, Object, Message และความสัมพันธ์ระหว่าง Concept เหล่านั้น (ประกอบด้วย Association, Dependency และ Generalization) ซึ่งเราสามารถใช้นั้นได้ในหลายๆแผนภาพโดยที่ความหมายไม่มีการเปลี่ยนแปลง

4. **ส่วนเพิ่มเติม (General Mechanism)** เราสามารถทำการเพิ่มเติมคำแนะนำ ข้อมูลอื่นๆที่เกี่ยวข้องหรือความหมายขององค์ประกอบย่อยๆของแบบจำลอง และนอกจากนั้น เรายังสามารถเพิ่มเติมรายละเอียดในส่วนต่างๆเข้าไปใน UML เพื่อให้เหมาะสมกับระเบียบวิธีที่เราใช้

### 2.1.3) UML กับ Object-Oriented Development Process

เมื่อเราพูดถึง “วิธี (Method)” เราจะหมายถึง สิ่งที่จะบอกเราถึงสิ่งที่จะต้องทำ ต้องทำอะไร เมื่อใด และทำไม (จุดประสงค์ในการทำ) (Eriksson, H. and Penker, M. 1998) วิธีจะประกอบไปด้วยแบบจำลอง (Model) ซึ่งจะถูกใช้ในการอธิบายและการสื่อสารให้เห็นถึงผลที่จะเกิดจากการปฏิบัติตามวิธี ความแตกต่างที่สำคัญของวิธีและภาษาที่ใช้ในการสร้างแบบจำลองคือ ภาษาที่ใช้ในการสร้างแบบจำลองนั้น ไม่ได้ระบุถึงขบวนการหรือคำแนะนำที่จะบอกว่า จะต้องทำอะไร อย่างไร เมื่อใดและทำไม นอกจากนั้นการเลือกใช้วิธีการในการพัฒนาให้เหมาะสมกับภาษาที่ใช้ในการสร้างแบบจำลองยังเป็นสิ่งที่สำคัญซึ่งจะขอยกไปกล่าวในหัวข้อ 2.3 กระบวนการพัฒนา

## 2.2 การพัฒนาซอฟต์แวร์ในรูปแบบของคอมโพเนนท์ (Component-Based Software-Development)

ในการพัฒนาซอฟต์แวร์ด้วยแนวทางแบบวัตถุ (Object-Oriented Approach) นั้นมีจุดเด่นในแง่ของการที่มันทำให้เราสามารถนำชุดคำสั่ง (Code) เดียวกันใน แอปพลิเคชันที่แตกต่างกันได้ ซึ่งถือเป็นแนวความคิดในเบื้องต้นของการนำไปใช้ใหม่ (Reusability) ที่เป็นแนวความคิดที่ดีมาก แต่ในทางปฏิบัติแล้วเรากลับพบสิ่งนี้น้อยมาก เนื่องจากการที่จะใช้การนำมาใช้ใหม่ได้นั้นจะต้องมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบที่ดีมาก เพราะการนำมาใช้ใหม่ที่เรากำลังพูดถึงอยู่นี้เป็น การนำมาใช้ใหม่ในระดับของชุดคำสั่ง (Code Reuse) ตัวอย่างง่ายๆของการนำมาใช้ใหม่ในระดับของชุดคำสั่งได้แก่การอ้างอิงถึง Code Library ในแอปพลิเคชัน ซึ่งการนำมาใช้ใหม่ในระดับของชุดคำสั่งนี้ทำให้เกิดปัญหาตามมาได้มากมายเนื่องจากการให้ผู้ที่จะนำเอาชุดคำสั่งไปใช้ใหม่โดยให้ชุดคำสั่งไปด้วย หากผู้ใช้นั้นไม่ชอบชุดคำสั่งนั้นขึ้นมาแล้วทำการแก้ไข มันก็จะส่งผลกระทบต่อแอปพลิเคชันอื่นที่จะนำเอาชุดคำสั่งนั้นไปใช้งานต่อเพราะข้อกำหนดของชุดคำสั่งนั้นได้ถูกเปลี่ยนไปแล้ว ถ้ามีผู้ที่ทำการแก้ไขชุดคำสั่งในรูปแบบนี้หลายๆคน ชุดคำสั่งนี้ก็จะมียุค (Version) ทำให้การควบคุมการนำไปใช้ใหม่เป็นไปได้ยากมากขึ้นเช่นกัน เราจึงไม่ค่อยพบเห็นการนำไปใช้ใหม่มากนักในทางปฏิบัติ นอกจากนี้การทำเช่นนี้ยังเป็นการเปิดเผยซอร์สโค้ด (Source Code) ของเราให้ผู้อื่นได้รับรู้ด้วย ด้วยปัญหาเหล่านี้เอง การพัฒนาซอฟต์แวร์ในรูปแบบของคอมโพเนนต์ซอฟต์แวร์จึงเข้ามามีบทบาทสำคัญ เนื่องจากมันมีหลักการโดยพื้นฐานเหมือนกับการวิเคราะห์และออกแบบเชิงวัตถุทั่วไปคือ การพยายามแบ่งปัญหาออกเป็นองค์ประกอบย่อยๆในรูปแบบของวัตถุต่างๆ แนวความคิดของซอฟต์แวร์คอมโพเนนต์ก็เช่นเดียวกัน มันจะประกอบไปด้วยคอมโพเนนต์ย่อยๆซึ่งสามารถนำไปใช้ใหม่ได้ ที่อยู่ในรูปของไบนารี (Binary) จึงไม่สามารถแก้ไขหรือดูซอร์สโค้ดได้ โดยคอมโพเนนต์เหล่านี้จะสามารถทำการเสมือนว่าเชื่อมต่อเข้าด้วยกันเพื่อสร้างเป็นแอปพลิเคชันขึ้นมา โดยไม่คำนึงว่าคอมโพเนนต์นั้นจะมาจากผู้พัฒนาคนเดียวกันหรือไม่ ด้วยแบบจำลองแบบนี้ทำให้คอมโพเนนต์นั้นไม่ขึ้นอยู่กับแอปพลิเคชันที่ใช้งานมัน หรือกล่าวได้ว่ามันสามารถถูกใช้งานในหลายๆแอปพลิเคชันได้ และยังไม่ขึ้นอยู่กับภาษาในการโปรแกรมที่ใช้พัฒนามันขึ้นมาอีกด้วย

ซอฟต์แวร์คอมโพเนนต์สามารถถูกแบ่งออกได้เป็นหลายประเภท เช่น ซอฟต์แวร์คอมโพเนนต์ประเภทที่สามารถเห็นรูปร่างของมันได้ (Visual Component) เช่น ปุ่ม หรือ กล่องแสดงข้อความ (Text Box) ที่เราพบในส่วนติดต่อผู้ใช้แบบกราฟิก โดยทั่วไป หรือซอฟต์แวร์คอมโพเนนต์ที่ให้บริการต่างๆเช่น ส่วนที่ทำหน้าที่ในการตรวจสอบคำในแอปพลิเคชันประเภท Word Processing ในแอปพลิเคชันที่ถูกสร้างขึ้นมาจากคอมโพเนนต์นั้น เราสามารถเลือกคอมโพเนนต์ที่จะนำมาใช้งานในแอปพลิเคชันของเราได้ สิ่งนี้ทำให้เกิดประโยชน์ต่อผู้ใช้แอปพลิเคชันมาก ตัวอย่างเช่นในแอปพลิเคชันประเภท Word Processing ผู้ใช้อาจไม่ชอบส่วนที่ทำหน้าที่ตรวจสอบคำที่มากับแอปพลิเคชัน ถ้าแอปพลิเคชัน Word Processing นั้นถูกออกแบบมาให้สามารถทำการเปลี่ยนคอมโพเนนต์ที่เป็นส่วนที่ทำหน้าที่ตรวจสอบคำได้ ผู้ใช้ก็สามารถที่จะหาซื้อคอมโพเนนต์ที่ทำหน้าที่ตรวจสอบคำที่ผู้ใช้ชอบมาเปลี่ยนใส่แทนคอมโพเนนต์เดิมได้ โดยที่ไม่ต้องทำการคอมไพล์ตัวแอปพลิเคชันใหม่ ทำให้ไม่มีความจำเป็นที่เราต้องให้ซอร์สโค้ดของแอปพลิเคชันไปกับตัวแอปพลิเคชันด้วย และผู้พัฒนา ก็ไม่ต้องยึดติดอยู่กับภาษาในการพัฒนาภาษาใดภาษาหนึ่งด้วย แนวทางนำเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาใช้ใหม่ในรูปแบบนี้เรียกว่าการนำชุดคำสั่งที่ได้รับการคอมไพล์แล้วมาใช้ใหม่ (Binary reuse) เนื่องจากมันมีพื้นฐานบนส่วนที่ใช้ติดต่อ (Interface) ไม่ใช่บนชุดคำสั่งแบบในแนวทางเชิงวัตถุ โดยการใช้หลักการของส่วนที่ใช้ติดต่อนี้ ครอบคลุมที่ซอฟต์แวร์คอมไพเลอร์นั้นทำตามสิ่งที่กำหนดเอาไว้ในส่วนที่ใช้ติดต่อนี้ มันก็สามารถที่จะทำงานร่วมกับส่วนอื่นๆของแอปพลิเคชันได้โดยไม่ต้องมีการคอมไพล์แอปพลิเคชันใหม่

หลักการที่สำคัญอีกประการหนึ่งในการพัฒนาซอฟต์แวร์จากคอมไพเลอร์คือการสนับสนุนการทำงานร่วมกันของคอมไพเลอร์ (Interoperability) นั่นคือเวลาที่เรากำลังทำการวิเคราะห์ระบบที่จะทำการพัฒนา เราควรที่จะค้นหาว่ามีบางส่วนในระบบที่ไม่ควรจะทำงานร่วมกันหรือไม่ ส่วนดังกล่าวนี้เองที่ควรแยกออกเป็นคอมไพเลอร์ นอกจากหลักการนี้แล้วยังมีอีกหลักการหนึ่งของซอฟต์แวร์คอมไพเลอร์ที่สอดคล้องกับแนวทางเชิงวัตถุ นั่นคือการซ่อนรายละเอียด (Encapsulation) ซึ่งเป็นคุณสมบัติของวัตถุที่จะซ่อนรายละเอียดของตัวเองเอาไว้จากผู้ใช้งานมัน ผู้ที่ใช้งานมันจะทำงานกับมันผ่านส่วนติดต่อที่วัตถุที่กำหนดไว้เท่านั้น ซอฟต์แวร์คอมไพเลอร์ก็เช่นเดียวกัน วัตถุแต่ละตัวที่อยู่ในซอฟต์แวร์คอมไพเลอร์นั้นจะประกาศสิ่งที่มันสามารถให้บริการได้ผ่านทางส่วนติดต่อที่แน่นอน ดังนั้นทางเดียวที่จะสามารถเข้าถึงบริการของวัตถุในซอฟต์แวร์คอมไพเลอร์ได้จะต้องทำผ่านส่วนติดต่อที่มันรองรับเท่านั้น และส่วนติดต่อนี้เองที่เป็นพื้นฐานของการนำเอาหลักการของการสนับสนุนการทำงานร่วมกันของคอมไพเลอร์ (Interoperability) มาปฏิบัติ

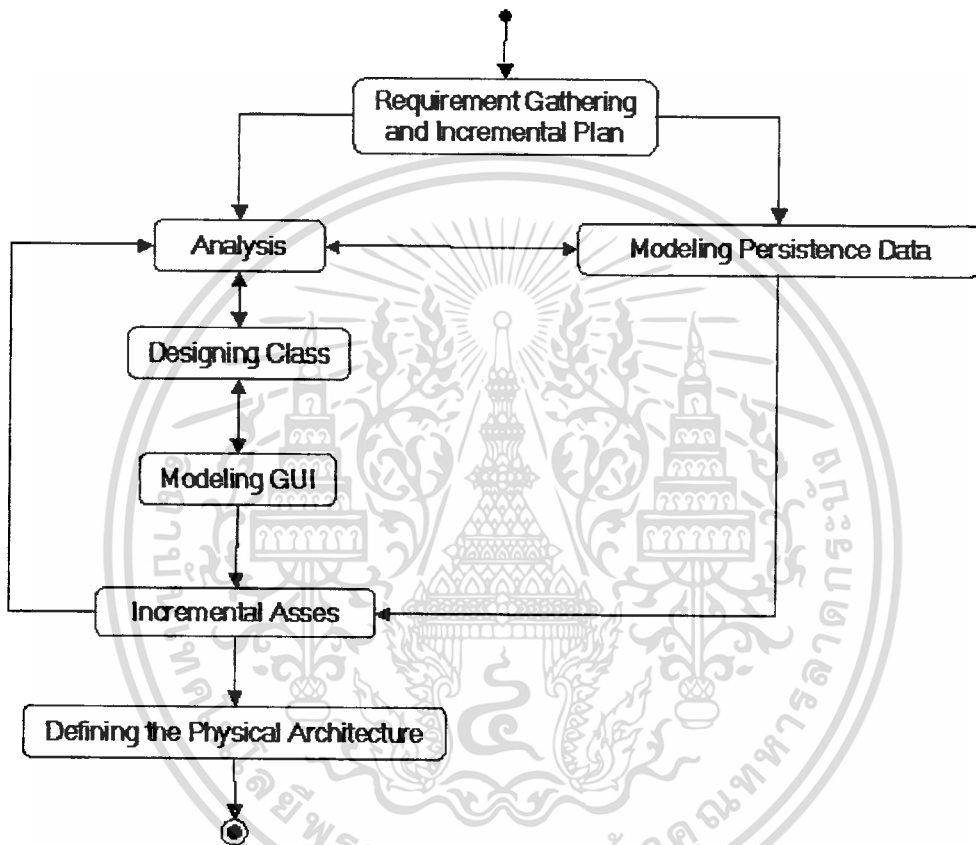
### 2.3 กระบวนการพัฒนา (Development Process)

ในการพัฒนาระบบสำหรับกรณีศึกษาที่ได้อ้างอิงไว้ กระบวนการพัฒนาเชิงวัตถุในการพัฒนาซึ่งในปัจจุบันมีกระบวนการพัฒนาเชิงวัตถุเกิดขึ้นมากมายแต่สิ่งที่สำคัญในการที่จะเลือกขบวนการใดๆในการพัฒนาเชิงวัตถุเพื่อใช้งานร่วมกับ UML นั้น (Eriksson, H. and Penker, M. 1998) แนะนำว่าควรจะมีลักษณะดังต่อไปนี้ คือ

- 1). เป็นขบวนการที่งานส่วนต่างๆในขบวนการล้วนมี Use Case เป็นพื้นฐานในการดำเนินงาน (Use Case-Driven)
- 2). เป็นขบวนการที่ให้ความสำคัญในเรื่องของสถาปัตยกรรมของระบบ (Architecture Centric)
- 3). เป็นขบวนการที่มีลักษณะที่ใช้การทำซ้ำ (Iteration)
- 4). เป็นขบวนการที่มีลักษณะของค่อยๆทำการเพิ่มการพัฒนาทีละขั้น (Incremental)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับขบวนการในการพัฒนาระบบสารสนเทศเพื่อการบริหารงานห้องสมุดนี้จะเป็นการประยุกต์ใช้ขบวนการพัฒนาเชิงวัตถุ (Object-Oriented Development Process) ที่ถูกพัฒนาขึ้นโดย Rowlett (2001) และขบวนการในการพัฒนาซอฟต์แวร์คอมพิวเตอร์ (Kirtland 1998) ซึ่งขบวนการโดยรวมแสดงในรูปที่ 2.1



รูปที่ 2.1 แผนภาพแสดงภาพรวมของขบวนการในการพัฒนาที่ใช้ในโครงการนี้

### 2.3.1) การรวบรวมความต้องการของระบบ (Requirement Gathering)

ในขั้นตอนนี้เราจะใช้ข้อความระบุถึงความต้องการของระบบ (Requirement Statement) มาใช้ในการดำเนินการโดยสำหรับคำถามที่ค้นพบในข้อความระบุความต้องการของระบบให้ทำการแยกประเภทออกเป็นวัตถุ คุณสมบัติของวัตถุ ผู้กระทำ หรืออื่นๆ โดยสำหรับผู้กระทำแต่ละรายให้ทำการแสดงรายชื่อของ Use Case ที่เกี่ยวข้อง และบันทึกเป็นข้อกำหนดของ Use Case (Use Case Definition) โดยทำการบันทึกสิ่งที่ต้องนำเข้าและผลลัพธ์ของแต่ละ Use Case จากนั้นให้ทำการแยก Use Case ทั้งหมดเพื่อวางแผนในการดำเนินการทีละขั้น (Increments)

### 2.3.2) การวิเคราะห์ (Analysis and Application Specification)

ในขั้นตอนนี้จะเป็นการอธิบายเงื่อนไขก่อนเกิดเหตุการณ์ (ในเทอมของคุณสมบัติของระบบและสิ่งที่นำไปใน Use Case) สำหรับแต่ละ Use Case โดยใช้ข้อกำหนดของ Use Case (สำหรับการพัฒนาในขั้นตอนปัจจุบัน) รายชื่อวัตถุและ คุณสมบัติของวัตถุเบื้องต้นที่ได้จากขั้นตอนที่ผ่านมา โดยในการดำเนินการให้ทำการอธิบายค่าสุดท้ายของคุณสมบัติที่มีการเปลี่ยนแปลงในเหตุการณ์นั้นๆ และสำหรับวัตถุทุกๆตัวและคุณสมบัติของมันที่ถูกระบุใน Use Case ให้สร้างพจนานุกรมข้อมูล และนำคุณสมบัติของวัตถุนั้นไปใส่เอาไว้ในแผนภาพคลาส

เมื่อเสร็จสิ้นกิจกรรมทั้งหมดเราจะได้รายละเอียดของแต่ละ Use Case (Use Case Description) ในรูปของตารางตัดสินใจ พจนานุกรมข้อมูล (Data Dictionary) แผนภาพคลาส (Class Diagram) สำหรับในขั้นตอนนี้มีข้อสังเกตบางประการสำหรับการอธิบายเหตุการณ์ในแต่ละ Use Case คือในการบันทึกรายละเอียดของแต่ละ Use Case สิ่งที่จะใช้ในการอธิบายควรมีคุณสมบัติที่เหมาะสมในการอธิบาย ในที่นี้เราจะใช้ Decision Table ในการบันทึกรายละเอียดของแต่ละ Use Case โดยในแต่ละสดมภ์จะแสดงถึงเหตุการณ์ที่สามารถเกิดขึ้นใน Use Case นั้นได้ คำ T และ F ในครั้งแรกของตารางจะหมายถึงค่าของเงื่อนไขในแต่ละเหตุการณ์ที่เกิดขึ้น ส่วนช่องที่ว่างจะหมายถึงในเหตุการณ์นั้นจะไม่ทำการพิจารณาเงื่อนไขในแถวที่ว่าง หรือเงื่อนไขเป็นสิ่งที่ไม่สำคัญต่อเหตุการณ์นั้น ส่วนในตารางครึ่งด้านล่างจะหมายถึงการตอบสนองของระบบต่อเหตุการณ์นั้น เราสามารถใส่รายละเอียดให้กับเงื่อนไขหรือผลตอบสนองโดยอ้างอิงออกมาอธิบายต่างหาก ดังตัวอย่างที่แสดงในตารางที่ 2.1

เราจะใช้ Decision Table ในการบันทึกรายละเอียดของแต่ละ Use Case เนื่องจากมันมีคุณสมบัติที่เหมาะสมในการอธิบาย Use Case ดังมีรายละเอียดดังต่อไปนี้

#### 1). เงื่อนไขก่อนที่จะเกิดเหตุการณ์ (Pre Condition) จะต้องสมบูรณ์

ในเหตุการณ์หนึ่งๆจะต้องมีทุกรูปแบบการรวมกันของเงื่อนไขทั้งหมด แต่ถ้าเราทำเช่นนี้ เราต้องสร้างตารางตัดสินใจให้ครอบคลุมทุกๆเงื่อนไข โดยการสร้างสดมภ์ให้ครอบคลุมทุกๆกรณีของรูปแบบเงื่อนไข (T และ F) ซึ่งทำให้จำนวนสดมภ์ที่ต้องสร้างต่อหนึ่งเหตุการณ์นั้นมีจำนวนมาก ( $2^n$ ,  $n$  = จำนวนเงื่อนไขก่อนเกิดเหตุการณ์) วิธีอื่นที่สามารถแสดงผลได้ครอบคลุมเช่นเดียวกัน ดังแสดงในตัวอย่างในตารางที่ 2.1 เป็นการใชสดมภ์เพียง 5 สดมภ์ในการแสดงผลเงื่อนไข ซึ่งถ้าจะแสดงทุกเงื่อนไขสำหรับกรณีจะต้องใช้สดมภ์ถึง 16 สดมภ์ ( $2^4 = 16$ ) ดังตารางที่ 2.2 ซึ่งสามารถพิสูจน์ได้ดังนี้ จะเห็นว่าเหตุการณ์ที่ 3 เพียงเหตุการณ์เดียวสามารถแสดงผลแทนเหตุการณ์ที่พิจารณาจากทุกเงื่อนไข ได้ 2 เหตุการณ์ คือเหตุการณ์ที่ 3 และ 4 เช่นเดียวกันสำหรับเหตุการณ์ที่ 2 เพียงเหตุการณ์เดียวสามารถแสดงผลแทนเหตุการณ์ที่พิจารณาจากทุกเงื่อนไข ได้ 4 เหตุการณ์ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์ที่ 5 6 7 และ 8 และเหตุการณ์ที่ 1 เพียงเหตุการณ์เดียวสามารถแสดงผลแทนเหตุการณ์ที่พิจารณาจากทุกเงื่อนไข ได้ 8 เหตุการณ์ คือเหตุการณ์ที่ 9-16 ด้วยเหตุผลเหล่านี้จึงทำให้การใช้เพียง 4 เหตุการณ์จึงสามารถแทนเงื่อนไขได้ครอบคลุมทุกกรณีเช่นเดียวกัน

ตารางที่ 2.1 ตัวอย่าง Decision Table สำหรับ Use Case ยืมสารนิเทศ

Scenarios	1	2	3	4	5			
<b>Conditions</b>								
พบสารนิเทศในระบบ	F	T	T	T	T			
สารนิเทศอยู่ในสถานะที่ว่าง		F	T	T	T			
พบสมาชิกในระบบ			F	T	T			
สมาชิกอยู่ในสถานะที่สามารถยืมได้*				F	T			
<b>Actions</b>								
ยกเลิกการยืม	X	X	X	X				
ให้ยืม**					X			

\*สมาชิกจะสามารถทำการยืมสารนิเทศได้ไม่เกินที่กำหนด และไม่มีสารนิเทศที่สมาชิกยืมไปคืนใดที่เกินกำหนดส่ง และสมาชิกต้องไม่ค้างค่าปรับกับทางห้องสมุด

\*\*ขั้นตอนการให้ยืมจะประกอบด้วยการกำหนดสถานะของสารนิเทศนั้นเป็น “ถูกยืม” และทำการสร้างข้อมูลการยืมของสมาชิกรายนั้น

ตารางที่ 2.2 ตัวอย่างการแสดงทุกกรณีของเงื่อนไขไว้ในตารางตัดสินใจ

เหตุการณ์	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
เงื่อนไขที่ 1	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
เงื่อนไขที่ 2	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
เงื่อนไขที่ 3	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
เงื่อนไขที่ 4	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2). เงื่อนไขก่อนที่จะเกิดเหตุการณ์จะต้องแยกจากกันโดยสิ้นเชิง

ไม่มีภาวะของระบบใดๆที่จะสามารถที่จะทำให้เงื่อนไขก่อนเกิดเหตุการณ์มากกว่า 1 เงื่อนไขเป็นจริงได้ ยกตัวอย่างในตารางที่ 2.3 ซึ่งมี 3 เงื่อนไข จะสังเกตเห็นว่า ในเหตุการณ์ที่ 2 และ 3 นั้นมีการซ้ำซ้อนกันอยู่ สังเกตที่เงื่อนไข “สมาชิกติดค้างค่าปรับกับห้องสมุด” นั้นสามารถเป็นได้ทั้ง T และ F ในเหตุการณ์ที่ 2 และนอกจากนั้นเงื่อนไข ”สมาชิกมีสารนิเทศที่เกินกำหนดคืนอยู่” นั้นสามารถเป็นได้ทั้ง T และ F ในเหตุการณ์ที่ 3 นั้นหมายถึงว่าเงื่อนไขที่จะทำการลบสมาชิกได้นั้นคือ “สมาชิกติดค้างค่าปรับกับห้องสมุด” และ”สมาชิกมีสารนิเทศที่เกินกำหนดคืนอยู่” ซึ่งอยู่ในเหตุการณ์มากกว่า 1 เหตุการณ์ การหลีกเลี่ยงเหตุการณ์นี้ทำได้โดยการพยายามพิจารณาจากเงื่อนไขที่มีอยู่ โดยเริ่มจากจำนวนเงื่อนไขที่น้อยที่สุดแล้วค่อยๆเพิ่มขึ้นเมื่อพิจารณาในเหตุการณ์ถัดไปทางขวามือของตาราง โดยจะเริ่มต้นจากการพิจารณาเงื่อนไขที่มีอิทธิพลมากที่สุดก่อนแล้วค่อยๆพิจารณาเงื่อนไขที่มีอิทธิพลรองลงมา ซึ่งได้ทำการแสดงเป็นตัวอย่างในตารางที่ 2.4 และตารางที่ 2.5 ตามลำดับ

ตารางที่ 2.3 ตัวอย่าง Decision Table ที่มีเหตุการณ์ซ้ำซ้อนกันอยู่

Scenarios	1	2	3
Conditions			
พบสมาชิกในระบบ	F	T	T
สมาชิกมีสารนิเทศที่เกินกำหนดคืนอยู่		T	
สมาชิกติดค้างค่าปรับกับห้องสมุด			T

ตารางที่ 2.4 ตัวอย่างการสร้าง Decision Table ของ Use Case ลบสมาชิกในขั้นแรก

Scenarios	1	2	3
Conditions			
พบสมาชิกในระบบ	F	T	T
สมาชิกมีสารนิเทศที่เกินกำหนดคืนอยู่		T	F
สมาชิกติดค้างค่าปรับกับห้องสมุด			

ตารางที่ 2.5 ตัวอย่าง Decision Table ชั้นสมบูรณ์ของ Use Case อบรมสมาชิก

Scenarios	1	2	3	4
Conditions				
พบสมาชิกในระบบ	F	T	T	T
สมาชิกมีสารนิเทศที่เกินกำหนดคืนอยู่		T	F	F
สมาชิกติดค้างค่าปรับกับห้องสมุด			T	F

ข้อสังเกตอีกประการหนึ่งคือ จริงๆแล้วเราสามารถระบุเงื่อนไข “สมาชิกติดค้างค่าปรับกับห้องสมุด” และ “สมาชิกมีสารนิเทศที่เกินกำหนดคืนอยู่” อย่างหนึ่งอย่างใดเพียงอย่างเดียวก็ได้ แต่ในทางปฏิบัติแอปพลิเคชันที่ทำการพัฒนา อาจจะมีความต้องการให้มีการแจ้งเตือนอย่างชัดเจนถึงเงื่อนไขต่างๆ

### 3). เงื่อนไขหลังเกิดเหตุการณ์จะต้องไม่มีการซ้ำเกิดขึ้น

เมื่อทำการพัฒนาในส่วนของเงื่อนไขก่อนเกิดเหตุการณ์เรียบร้อยแล้ว ขั้นตอนต่อมาคือการพิจารณาว่ามีสองเหตุการณ์ใดๆหรือไม่ที่มีการตอบสนองของระบบเหมือนกัน สิ่งนี้ไม่ได้บอกถึงความผิดพลาดในการกำหนดข้อกำหนดของความต้องการของระบบ แต่มันจะบอกเราถึงสิ่งที่ยังขาด ตัวอย่างในตารางที่ 2.6 ซึ่งแสดงถึงตารางตัดสินใจของ Use Case รับชำระค่าปรับ จะสังเกตได้ว่าการตอบสนองในเหตุการณ์ที่ 3 และ 5 นั้นเหมือนกัน ทั้งๆที่ตารางตัดสินใจนี้ก็ต้อง ครอบคลุมที่กฎทางธุรกิจกำหนดว่าให้ปรับค่าค้างชำระลงเป็นศูนย์

การที่มีการตอบสนองที่เหมือนกันในเหตุการณ์ที่ต่างกันั้นนั้น จะบอกเราว่าเราควรจะทำ การพิจารณากฎเกณฑ์ทางธุรกิจนี้ใหม่ ซึ่งอาจจะมีข้อสรุปออกมาได้หลายแบบเช่น

1. เหตุการณ์ที่ให้ผลตอบสนองเหมือนกันนั้นควรจะนำมารวมกันเป็นเหตุการณ์เดียว
2. กฎเกณฑ์ทางธุรกิจนั้นอาจไม่ถูกต้องหรือไม่สมบูรณ์ อาจจะเป็นไปได้ที่มีเงื่อนไขหรือการตอบสนองบางอย่างที่หายไป
3. ลูกค้าน่ามีความต้องการให้มีการตอบสนองที่แตกต่างกันเมื่อมีการชำระเงินพอดีกับที่ค้างชำระกับในกรณีที่มีการชำระเงินมากกว่าที่ค้างชำระ

ตารางที่ 2.6 ตารางแสดงตัวอย่าง Decision Table สำหรับ Use Case รับชำระค่าปรับ

Scenarios	1	2	3	4	5
<b>Conditions</b>					
พบสมาชิกในระบบ	F	T	T	T	N
สมาชิกไม่ติดค้างค่าปรับ		T	F	F	F
จำนวนเงินที่ชำระมากกว่าที่ค้างอยู่			T		
จำนวนเงินที่ชำระน้อยกว่าที่ค้างอยู่				T	
จำนวนเงินที่ชำระเท่ากับที่ค้างอยู่					T
<b>Actions</b>					
ยกเลิกทรานแซคชั่น	X				
แจ้งสมาชิกว่าไม่ได้ค้างชำระ		X			
กำหนดค่าการค้างชำระเป็นศูนย์			X		X
ลดจำนวนหนี้ที่ค้างเท่ากับจำนวนที่ชำระ				X	
แจ้งสมาชิกว่าเขายังคงค้างชำระอยู่				X	
ทอนเงิน			X		

#### 4). เงื่อนไขทางธุรกิจจะต้องไม่คลุมเครือ

จะเห็นว่ากฎเกณฑ์ทางธุรกิจ (Business Rule) ที่ได้เห็นจากตารางตัดสินใจนี้ค่อนข้างชัดเจนกว่าการเขียนในลักษณะที่เป็นการบรรยาย แต่ก็ยังมีคำแนะนำบางส่วนที่จำเป็นในการบันทึกรายละเอียดของกฎเกณฑ์ทางธุรกิจนั้นคือ

1. พยายามใช้เฉพาะคำที่มีการอธิบายไว้ในพจนานุกรมข้อมูล Use Case ควรจะต้องถูกอธิบายโดยใช้คำที่อยู่ในรูปของวัตถุทางธุรกิจ ค่า Attribute ของวัตถุทางธุรกิจ และ Input Parameter ของ Use Case นั้น
2. พยายามอธิบายความหมายของสิ่งหนึ่งๆ โดยใช้คำเพียงคำเดียว
3. พยายามหลีกเลี่ยงการใช้งานคำคุณศัพท์ เพราะมันจะทำให้เกิดความคลุมเครือขึ้นได้ง่าย
4. ในกรณีที่มีการคำนวณหรืออะไรที่เกี่ยวข้องพยายามแสดงออกโดยใช้สูตรทางคณิตศาสตร์ดีกว่าใช้การเขียนอธิบาย

ในการอธิบายรายละเอียดของ Use Case นั้น ไม่จำเป็นที่ทุก Use Case จะต้องใช้ตารางตัดสินใจในการอธิบาย แต่ถ้าใช้ก็จะช่วยให้เข้าใจรายละเอียดต่างๆ ได้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.3) การออกแบบรูปแบบการเก็บข้อมูล (Modeling Persistence Data)

ในขั้นตอนนี้จะเป็นการนำเอาข้อความระบุถึงความต้องการของระบบ (Requirement Statement) มาดำเนินการค้นหาข้อมูลที่จะต้องถูกจัดเก็บภายในระบบของเรา ไม่ว่าฐานข้อมูลที่เราจะใช้งานอาจจะอยู่ในรูปแบบใดก็ตาม ซึ่งในที่นี้เราจะใช้ฐานข้อมูลเชิงสัมพันธ์ในการพัฒนา เนื่องจากเป็นฐานข้อมูลที่มีการใช้งานกันมาเป็นระยะเวลานาน ดังนั้นเทคโนโลยีต่างๆที่เกี่ยวข้องจึงได้รับการพัฒนาจนได้รับการยอมรับอย่างกว้างขวาง ดังนั้น เราจึงต้องดำเนินการพัฒนาฐานข้อมูลตามหลักการของการพัฒนาฐานข้อมูลเชิงสัมพันธ์ ในโครงการนี้จะไม่ทำการพัฒนาแบบจำลองของคลาสแล้วทำการแปลงให้อยู่ในรูปแบบของแบบจำลองของตารางในฐานข้อมูลเชิงสัมพันธ์ เนื่องจากในแนวทางการออกแบบแอปพลิเคชันที่สร้างขึ้นมาจากคอมพิวเตอร์นั้น มีมุมมองต่อแบบจำลองคลาสเป็นเพียงตัวแทนในทางความคิด (Abstraction) ของชุดของวัตถุที่ทำหน้าที่ในการให้บริการต่อสิ่งที่มาใช้บริการจากมันเท่านั้น ไม่ได้มองว่าวัตถุเป็นผู้เก็บข้อมูลไม่ว่าจะเป็นการเก็บข้อมูลแบบชั่วคราวหรือถาวรก็ตาม ดังนั้นการจัดเก็บข้อมูลจึงควรเป็นหน้าที่ของฐานข้อมูล สำหรับการพัฒนาในขั้นตอนนี้ของกรณีศึกษาจะอยู่ในบทที่ 5 การออกแบบระบบฐานข้อมูล

### 2.5.4) การออกแบบคลาส (Designing Class)

ในขั้นตอนการออกแบบคลาสนี้เราต้องทำความเข้าใจกับสถาปัตยกรรมที่เราใช้ เพื่อที่จะได้เข้าใจการแบ่งแยกประเภทของคลาสได้ดียิ่งขึ้น

สำหรับสถาปัตยกรรมที่จะถูกนำมาประยุกต์ใช้กับระบบของเรานั้นจะอยู่ในรูปของการแบ่งแยกบริการออกเป็นชั้น (Layer) และทำการพัฒนาระบบโดยใช้คอมโพเนนต์ (Component) เป็นองค์ประกอบของระบบ โดยที่แบบจำลองที่เราใช้คือ แบบจำลองแบบ 3 ชั้น (3-Tier) ซึ่งจะมองทุกๆ แอปพลิเคชันที่เกี่ยวข้องในฐานะที่เป็นชุดของบริการซึ่งแบ่งออกเป็น 3 หมวดหมู่ ดังที่แสดงไว้ในรูปที่ 2.2 คือ

1. ชั้นบริการผู้ใช้ (User Service Layer) ประกอบด้วยส่วนติดต่อกับผู้ใช้แบบกราฟฟิก (Graphical User Interface-GUI) และตรรกะที่เกี่ยวข้องในการนำเสนอข้อมูลให้ผู้ใช้
2. ชั้นบริการธุรกิจ (Business Service Layer) เป็นชั้นที่รวมเอากฎเกณฑ์และตรรกะทางด้านธุรกิจทั้งหมดที่จำเป็นที่ต้องใช้ในการควบคุมกฎเกณฑ์ทางด้านธุรกิจ การออกแบบวัตถุที่ทำงานในชั้นนี้นั้นมีจุดประสงค์เพื่อทำการกำหนดวัตถุที่ให้บริการต่างๆ ที่แสดงออกมาในรูปของสิ่งที่ระบบทำ วัตถุธุรกิจควบคุมการดำเนินงานและกฎเกณฑ์ในทางธุรกิจต่างๆ ของระบบ เช่นเกี่ยวกับการควบคุมบูรณาภาพในการดำเนินการ (Transaction Integrity) ของกิจกรรมทุกอย่างที่มันกระทำ วัตถุธุรกิจที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้รับการออกแบบเป็นอย่างดีนั้น สามารถนำไปใช้ใหม่ได้ในหลายแอปพลิเคชัน เราสามารถแบ่งบริการต่างๆที่ชั้นนี้ให้บริการออกเป็น 3 ประเภท คือ

(2.1) บริการในส่วนหน้า (Façade Service) เป็นบริการที่ถูกสร้างขึ้นจากฟาซาดออบเจกต์ (Façade Object) ซึ่งหน้าที่สำคัญของชั้นย่อยนี้คือ การแยกส่วนติดต่อผู้ใช้ออกจากความยุ่งยากซับซ้อนของชั้นที่ให้บริการทางธุรกิจ ทำให้การพัฒนาส่วนติดต่อผู้ใช้นั้นสามารถทำได้ง่ายขึ้น

(2.2) บริการต่อส่วนหลักของธุรกิจ (Main Business Service) เป็นบริการที่ถูกสร้างขึ้นจากวัตถุธุรกิจ (Business Object หรือ Entity Object) ซึ่งเป็นส่วนหลักของแอปพลิเคชันทั้งหมด วัตถุที่ใช้ในบริการนี้จะมีโอกาสที่จะถูกนำไปใช้ใหม่ (Resuability) สูงมาก

(2.3) บริการเข้าถึงข้อมูล (Data Access Service) เป็นบริการที่ถูกสร้างขึ้นจาก Data Access Object ซึ่งทำหน้าที่ในการแยกแอปพลิเคชันทั้งหมดออกจากหน้าที่ทั้งหมดที่เกี่ยวข้องกับฐานข้อมูล จุดประสงค์หลักของมันคือการให้บริการข้อมูลแก่วัตถุธุรกิจ การแก้ไข ปรับปรุงฐานข้อมูลตามความต้องการของบริการต่อส่วนหลักของธุรกิจ รวมถึงความรับผิดชอบต่อการตรวจสอบข้อมูลว่ามีความถูกต้องตามเงื่อนไขที่กำหนดหรือไม่ โดย Data Access Object เองก็ยังสามารถแบ่งออกได้อีกเป็น 2 ประเภทคือ

Data Access Object ที่ทำหน้าที่ในการดึง (Fetch) ข้อมูลออกมาจากฐานข้อมูลเป็นหลัก ซึ่งไม่เป็นส่วนหนึ่งของทรานแซกชัน (Transaction) ใดๆ

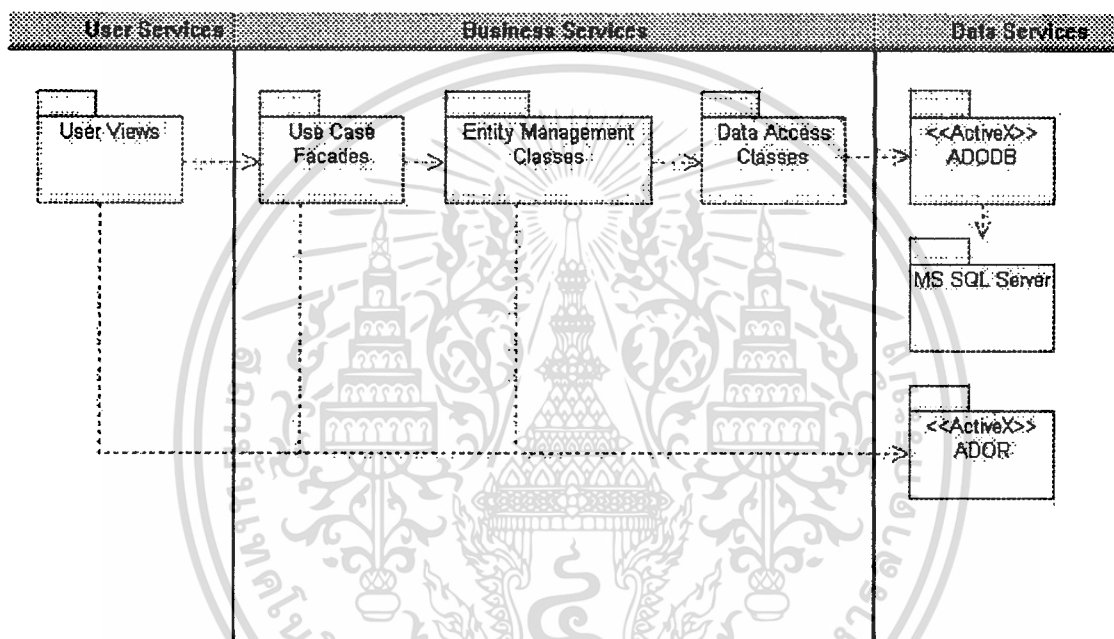
Data Access Object ที่ทำหน้าที่ในด้านที่เกี่ยวข้องกับการปรับปรุง แก้ไขฐานข้อมูล ซึ่งถือเป็นส่วนหนึ่งหรือทั้งหมดของทรานแซกชันใดๆ

3. ชั้นบริการข้อมูล (Data Service Layer) ประกอบด้วยฐานข้อมูลและส่วนที่ใช้ในการติดต่อเพื่อเข้าถึงฐานข้อมูล (Data Access Interface) ซึ่งในที่นี้คือ ADO และ OLE DB

อนึ่ง ในการดำเนินการพัฒนาคลาสประเภทต่างๆที่กล่าวมานั้น เนื่องจากคลาสเหล่านี้มีความสัมพันธ์ในแบบพึ่งพากันอยู่ และเนื่องจากขบวนการที่เราใช้ในการพัฒนานั้นมีรูปแบบของการทำซ้ำได้ ดังนั้นการออกแบบคลาสประเภทหนึ่งอาจจะช่วยให้เราค้นพบคลาสในอีกประเภทหนึ่งได้ ยกตัวอย่างเช่น เมื่อเราออกแบบคลาส Data Access เรียบร้อยแล้ว และมาออกแบบคลาสธุรกิจซึ่งต้องพึ่งพาบริการของคลาส Data Access แต่พบว่าในกลุ่มของคลาส Data Access ที่ได้ออกแบบไว้ไม่มีคลาสที่ต้องการ เราสามารถกลับไปทำการเพิ่มคลาสที่เราต้องการเข้าไปในกลุ่มของคลาส Data Access ได้ นี่เป็นข้อดีประการหนึ่งของการใช้ขบวนการในการพัฒนาที่มีลักษณะที่ย้อนกลับ (ทำซ้ำใหม่) ได้ คือ เราสามารถย้อนกลับมาปรับปรุงแก้ไขรายละเอียดที่ได้ทำการออกแบบไปแล้วได้ ทำให้เราสามารถปรับปรุงแบบจำลองที่เราทำการออกแบบได้ตลอดเวลา ซึ่งเป็นการแก้

ปัญหาสำคัญที่พบในการออกแบบระบบที่มีความไม่แน่นอน หรือมีข้อมูลที่ยังไม่เป็นที่ทราบแน่ชัด ในขณะที่เริ่มทำการออกแบบ

สำหรับการแบ่งบริการของระบบออกเป็นชั้น (Layer) ในการพัฒนาระบบสารสนเทศเพื่อการบริหารงานห้องสมุดนี้ ในแต่ละชั้นจะมีคลาสที่ถูกพัฒนาในรูปของ COM (Component Object Model) Object ซึ่งอยู่ภายใน ActiveX Component โดยใช้ Visual Basic 6.0 เป็นเครื่องมือในการพัฒนา



รูปที่ 2.2 แผนภาพแสดงสถาปัตยกรรมของระบบ

สำหรับชั้นตอนนี้ มีประเด็นสำคัญที่ต้องกล่าวถึงคือ มุมมองที่แตกต่างกันในการพัฒนาระบบสารสนเทศโดยการใช้แนวทางการวิเคราะห์และออกแบบเชิงวัตถุกับการออกแบบคอมโพเนนต์ กล่าวคือ แม้ว่าการใช้การวิเคราะห์และออกแบบเชิงวัตถุในการสร้างแบบจำลองเชิงตรรก (Logical Model) ของระบบที่จะทำการพัฒนานั้น จะเป็นที่ยอมรับโดยทั่วไปในปัจจุบันก็จริง แต่ก็ไม่สามารถนำมาประยุกต์ใช้กับการออกแบบคอมโพเนนต์ได้อย่างเต็มที่ เนื่องจากมุมมองของแนวคิดเชิงวัตถุนั้นจะเป็นมุมมองที่เน้นในเรื่องของสิ่งที่นำเสนอถึงระบบที่พิจารณา (Concept) องค์ประกอบย่อยๆ ในระบบ และการทำงานร่วมกันของสิ่งเหล่านี้ แต่ในมุมมองของการออกแบบคอมโพเนนต์แล้ว จะเป็นการเน้นไปที่การให้บริการมากกว่า ซึ่งในการออกแบบนั้น เราไม่สามารถทำการตัดสินใจได้โดยง่ายว่าหน้าที่ให้บริการควรจะเป็นของวัตถุใดวัตถุหนึ่ง หรือเป็นหน้าที่ของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะผิดใจทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุหลายๆตัวที่มาทำงานร่วมกัน จุดนี้เองที่ทำให้เมื่อเราใช้แนวคิดเชิงวัตถุทั้งหมดกับการออกแบบคอมโพเนนท์ จะทำให้เราอาจมองข้ามมุมมองในเรื่องของการจัดกลุ่มวัตถุที่ควรจะต้องทำงานร่วมกันเอาไว้ในกลุ่มเดียวกัน ซึ่งภายหลังในขั้นตอนการพัฒนาวัตถุเหล่านั้น จะถูกจัดให้อยู่ในคอมโพเนนท์เดียวกัน ซึ่งแนวความคิดนี้จะสนับสนุนต่อการใช้งานบนสถาปัตยกรรมแบบหลายชั้นของเราอีกด้วย นอกจากนี้ การออกแบบเชิงวัตถุมีแนวโน้มที่วัตถุที่ถูกออกแบบจะมีพฤติกรรมแบบรักษาสถานะตลอดเวลา (Stateful) ซึ่งในมุมมองของการออกแบบของเราที่ต้องการพัฒนาระบบที่มีความยืดหยุ่นและมีประสิทธิภาพในการใช้งานทรัพยากรแล้ว สิ่งนี้ไม่ใช่เรื่องที่ดี ดังนั้น การที่เราออกแบบวัตถุให้มีพฤติกรรมแบบที่ไม่รักษาสถานะ (Stateless) จึงค่อนข้างแตกต่างจากการออกแบบในทางเชิงวัตถุ สำหรับการดำเนินการในขั้นตอนนี้จะรวมอยู่ในบทที่ 3 การวิเคราะห์และออกแบบระบบ โดยจะรวมอยู่ในแต่ละขั้นของการพัฒนา

เราสามารถสรุปความแตกต่างของแนวความคิดในการออกแบบคลาสระหว่างแนวคิดเชิงวัตถุกับแนวคิดเชิงคอมโพเนนท์ได้ดังนี้ กล่าวคือ ดังที่กล่าวไว้ในหัวข้อ 2.5.3 ว่าในแนวทางการออกแบบแอปพลิเคชันที่สร้างขึ้นมาจากคอมโพเนนท์นั้น มีมุมมองต่อแบบจำลองคลาสเป็นเพียงตัวแทนในทางความคิด (Abstraction) ของชุดของวัตถุที่ทำหน้าที่ในการให้บริการต่อสิ่งที่มาใช้บริการจากมันเท่านั้น ไม่ได้มองว่าวัตถุเป็นผู้เก็บข้อมูล ไม่ว่าจะเป็นการเก็บข้อมูลแบบชั่วคราวหรือถาวรก็ตาม ดังนั้น แนวทางการออกแบบแอปพลิเคชันจึง ไม่ได้ให้ความสำคัญกับแบบจำลองของคลาส ทั้งในด้านที่แสดงถึง โครงสร้างของระบบ หรือในด้านการทำงานร่วมกันของวัตถุเพื่อแสดงออกถึงหน้าที่ของระบบ โดยสำหรับประเด็นแรกในแบบจำลองคลาส คือการแสดงถึงโครงสร้างในลักษณะเดียวกับที่ถูกแสดงในแบบจำลองของข้อมูล (Data Model) คือส่วนของข้อมูลจะถูกซ่อนเอาไว้ในวัตถุ และมีการกำหนดการเข้าถึงข้อมูลและบริการของแต่ละวัตถุ ในขณะที่สิ่งเหล่านี้ในแนวทางการออกแบบแอปพลิเคชันที่สร้างขึ้นมาจากคอมโพเนนท์นั้น ไม่ได้ถูกทำในลักษณะเดียวกันนี้ แต่จะต้องคำนึงถึงประเด็นของการจัดการสถานะของวัตถุและ การจัดการการใช้ทรัพยากร ซึ่งพิจารณาบนพื้นฐานของขอบเขตการดำเนินการ (Transaction Boundary) เช่นเดียวกับการจัดการในส่วนของการรักษาความปลอดภัยในการเข้าถึงวัตถุซึ่งพิจารณาในระดับของคอมโพเนนท์แอปพลิเคชัน คอมโพเนนท์ และขอบเขตการติดต่อ (Interface Boundary) การพิจารณาในส่วนเหล่านี้ เราสามารถใช้บริการของส่วนบริการคอมโพเนนท์ได้ สิ่งนี้ทำให้เรามีความจำเป็นที่จะต้องทำการออกแบบวัตถุของเราโดยคำนึงถึงแบบจำลองของการเขียนโปรแกรมของส่วนบริการคอมโพเนนท์ที่เราจะใช้งาน (ในที่นี้คือ COM+) นั่นคือเราจะต้องพิจารณาในประเด็นเกี่ยวกับลักษณะของวัตถุต่อไปนี้เพิ่มเติม คือ

- 1). สถานะของวัตถุ (object's state) เป็นการพิจารณาว่าข้อมูลอะไรที่วัตถุนี้ต้องการและจะจัดการข้อมูลเหล่านั้นอย่างไร
- 2). พฤติกรรมของวัตถุ (object's behavior) เป็นการพิจารณาบริการที่วัตถุจะมีให้
- 3). การใช้งานทรัพยากรของวัตถุ (object's resource usage) ประเด็นเรื่องการใช้งานทรัพยากรเช่น การติดต่อฐานข้อมูล การถือครองข้อมูล
- 4). พฤติกรรมเกี่ยวกับทรานแซกชันของวัตถุ (object's transactional behavior)
- 5). ประเด็นเรื่องความปลอดภัยของตัววัตถุ (object's security requirements) คือการพิจารณาถึงข้อจำกัดเกี่ยวกับการอนุญาตให้ใครบ้างที่สามารถเข้าถึงบริการของวัตถุได้
- 6). สิ่งที่มาเรียกใช้วัตถุ (object's clients)

#### 2.5.5) การออกแบบส่วนติดต่อผู้ใช้ (Modeling Presentation Layer)

ในการพิจารณาถึงส่วนที่ติดต่อกับผู้ใช้งาน เราสามารถพิจารณาจากแผนภาพ Use Case และควรสร้างส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้ 1 ส่วนจากแต่ละความสัมพันธ์ระหว่างผู้กระทำกับ Use Case ในแผนภาพ Use Case โดยส่วนติดต่อผู้ใช้นี้จะพึงพาการทำงานของคลาสส่วนหน้า ซึ่งจะควบคุมการดำเนินการทั้งหมดสำหรับ Use Case หนึ่งๆเอาไว้ การออกแบบส่วนติดต่อผู้ใช้ที่ดีควรจะทำให้มีความเรียบง่าย ผู้ใช้สามารถทำความเข้าใจได้ง่าย และควรจะมีการตอบสนองต่อผู้ใช้เมื่อผู้ใช้งานการดำเนินการอย่างหนึ่งอย่างใดลงไป ทุกๆบริการที่มีให้ผู้ใช้ ควรจะทำให้ผู้ใช้สามารถควบคุมทุกอย่างได้ เช่น เมื่อผู้ใช้สั่งดำเนินการใดๆก็ควรที่จะสามารถยกเลิกกิจกรรมนั้นได้ โดยประเด็นที่สำคัญที่เกี่ยวข้องกับสถาปัตยกรรมของระบบที่เราใช้งานคือ ไม่ว่าส่วนติดต่อผู้ใช้จะอยู่ในรูปแบบใด (อาจจะเป็นรูปแบบวินโดว์มาตรฐานหรืออยู่ในรูป Web Page ก็ตาม) ส่วนติดต่อผู้ใช้งานแต่ใช้บริการจากส่วนบริการธุรกิจส่วนเดียวกันทั้งสิ้น สิ่งนี้ทำให้การออกแบบส่วนติดต่อผู้ใช้มีความยืดหยุ่นในการเลือกที่จะพัฒนารูปแบบในการติดต่อผู้ใช้ได้หลากหลายขึ้น

#### 2.5.6) การประเมินผลก่อนที่จะไปยังการพัฒนาขั้นต่อไป (Incremental Assess)

เมื่อดำเนินการครบทุกขั้นตอนในการพัฒนาขั้นปัจจุบันแล้ว ให้ทำการตรวจสอบและประเมินผลการพัฒนาว่า ส่วนที่ดำเนินการพัฒนาไปแล้วมีข้อบกพร่องอะไรเกิดขึ้นบ้างหรือไม่ ก่อนที่จะดำเนินการพัฒนาในขั้นตอนต่อไปตามที่ได้วางแผนเอาไว้

### 2.5.7) การกำหนดสถาปัตยกรรมทางกายภาพที่จะนำเอาระบบไปใช้งาน (Defining the Physical Architecture)

เมื่อเราทำการออกแบบคลาสต่างๆเสร็จ ถือว่าเราทำการออกแบบในระดับแนวความคิด (Conceptual Design) เป็นที่เรียบร้อยแล้ว ขั้นตอนถัดไปคือการออกแบบในทางกายภาพ (Physical Design) ซึ่งมีรายละเอียดดังนี้คือ

- (1). จัดกลุ่มคลาส (ในที่นี้คือ COM+ Class) ที่ทำการออกแบบไว้ไปใส่ไว้ในคอมโพเนนท์
- (2). จัดกลุ่มคอมโพเนนท์เป็นแอปพลิเคชัน (ในที่นี้คือ COM+ Application หมายถึงกลุ่มของคอมโพเนนท์ที่ทำหน้าที่ให้บริการซึ่งเป็นของแอปพลิเคชันนั้นๆ)
- (3). กำหนดว่าจะติดตั้งแอปพลิเคชันลงในเครื่องคอมพิวเตอร์ที่จะใช้งานซอฟต์แวร์นี้อย่างไร



## บทที่ 3

### การวิเคราะห์ระบบ

สำหรับการวิเคราะห์และออกแบบระบบบริหารงานห้องสมุดนี้ ได้ใช้หลักการในส่วนของ การวิเคราะห์ส่วนใหญ่จากหลักการของการวิเคราะห์และออกแบบเชิงวัตถุที่ถูกพัฒนาขึ้น โดย (Rowlett, T.W. 2000) และหลักการในส่วนของ การออกแบบที่ถูกกล่าวเอาไว้ใน (Kirtland, M. 1999) และ (Sundblad, S. and Sundblad, P. 2000) โดยมีขั้นตอนต่างๆดังต่อไปนี้

#### 3.1 ความต้องการของระบบ

ผลจากการเก็บรวบรวมข้อมูลและศึกษาการทำงานของระบบงานเดิม เพื่อนำมาวิเคราะห์ หาปัญหา ข้อจำกัด ของระบบงานเดิม สามารถสรุปความต้องการของระบบที่จะทำการพัฒนาขึ้น ใหม่ได้ดังต่อไปนี้

##### 3.1.1. ภาพรวม

ระบบจะต้องสามารถให้บริการแก่ผู้มาใช้บริการ (Borrower) ของห้องสมุดที่อาจจะเข้ามา ทำการค้นหาหรือเรียกดูรายการหนังสือ (Title) ในรูปแบบต่างๆ เช่น อาจจะเรียกดูตามชื่อเรื่อง ชื่อผู้แต่ง และอื่นๆ โดยระบบจะต้องสามารถแสดงรายละเอียดทั้งหมดเกี่ยวกับรายการหนังสือนั้น และ หนังสือแต่ละเล่ม (Copy) ทั้งหมดในรายการหนังสือนั้น รวมถึงบอกได้ว่าหนังสือที่ต้องการนั้นอยู่ในสถานะใด (ว่างหรือถูกยืม) และหากหนังสือเล่มนั้นมีสถานะเป็นถูกยืม แล้วหนังสือนั้นมีกำหนด ที่จะต้องนำมาคืนเมื่อใด นอกจากนี้ระบบจะต้องสามารถที่จะจัดการข้อมูลการให้ยืมหนังสือ (Loan) แก่ผู้มาใช้บริการ โดยในการให้บริการยืม-คืนหนังสือแก่สมาชิกของห้องสมุด โดยที่การยืม หนังสือ นั้นจะต้องดำเนินการตามข้อตกลงในการยืมหนังสือของห้องสมุด (Lend Agreement) ซึ่งจะ จำกัดจำนวนการหนังสือที่สมาชิกแต่ละประเภทสามารถยืมได้ นอกจากนี้การให้บริการยืมหนังสือ อาจจะรวมถึงกรณีที่ผู้ใช้บริการต้องการที่จะทำการจองหนังสือ (Reservation) นั้นไม่ว่าหนังสือ นั้นจะอยู่ในสถานะที่ถูกยืม (Lend) หรือว่าง (Available) อยู่ กล่าวคือหากสารนิเทศนั้นอยู่ในสถานะ ที่ถูกยืม ระบบจะต้องทำการจองสารนิเทศนั้นจนกระทั่งสารนิเทศนั้นถูกนำมาคืน ระบบจะทำการ แจ้งบรรณารักษ์ให้ทำการเรียกผู้ใช้บริการตามลำดับในคิวในการมารับหนังสือคืน หรือในกรณีที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสือนั้นอยู่ในสถานะที่ว่างอยู่ ระบบจะต้องเปลี่ยนสถานะของหนังสือนั้นเป็นถูกยืมและบรรณารักษ์ (Librarian) หรือเจ้าหน้าที่ให้บริการยืมคืนหนังสือ (Circulation Clerk) จะนำหนังสือนั้นมาเก็บไว้รอให้ผู้ที่ยืมไว้มารับต่อไป ระบบจะต้องมีการกำหนดสิทธิในการติดต่อกับระบบและการเข้าถึงข้อมูลของผู้ใช้ โดยผู้ดูแลระบบ (Administrator) จะเป็นคนกำหนดสิทธิของผู้ใช้งานระบบซึ่งประกอบด้วย บรรณารักษ์ และเจ้าหน้าที่ให้บริการยืมคืนหนังสือ ส่วนผู้ใช้งานที่เป็นสมาชิกห้องสมุดนั้นจะไม่ได้รับอนุญาตให้ทำการติดต่อกับระบบโดยตรง แต่จะสามารถใช้บริการที่ห้องสมุดเปิดบริการสาธารณะเอาไว้ได้ เช่นการค้นหารายการสารนิเทศ เป็นต้น

### 3.1.2). สมาชิกห้องสมุด

ระบบจะต้องสามารถเพิ่มและ ลบ ข้อมูลของสมาชิกของห้องสมุด (Member) ได้ (โดยที่สมาชิกของห้องสมุดนั้นมีหลายประเภทตั้งแต่ นักศึกษาปริญญาตรี นักศึกษาปริญญาโท อาจารย์ อาจารย์พิเศษ ข้าราชการ และลูกจ้างประจำ โดยที่สมาชิกประเภทต่างก็จะมีสิทธิในการยืมหนังสือที่ต่างกันออกไป (อ่านรายละเอียดในส่วนของสารนิเทศ) สมาชิกแต่ละประเภทจะมีอายุการเป็นสมาชิกต่างกัน ไป สมาชิกประเภทนักศึกษาปริญญาตรีและนักศึกษาปริญญาโท จะเป็นสมาชิกห้องสมุดหากยังเป็นนักศึกษาของมหาวิทยาลัยและได้ทำการลงทะเบียนเรียนในภาคการศึกษานั้นเรียบร้อยแล้วดังนั้นสมาชิกประเภทนักศึกษาปริญญาตรีและนักศึกษาปริญญาโทจะต้องทำการต่ออายุการเป็นสมาชิกทุกภาคการศึกษา ส่วนสมาชิกประเภทอาจารย์ อาจารย์พิเศษ ข้าราชการ และลูกจ้างประจำจะมีอายุการเป็นสมาชิกตลอดการการเป็นพนักงานของมหาวิทยาลัย สมาชิกจะไม่มีสิทธิยืมหนังสือหากสมาชิกมีหนังสือที่ยังไม่ได้ทำการส่งคืนห้องสมุดเมื่อถึงกำหนดเวลาจนกว่าสมาชิกจะทำการคืนหนังสือดังกล่าวและชำระค่าปรับเป็นที่เรียบร้อยแล้ว จึงจะมีสิทธิในการยืมตามปกติ และสมาชิกจะติดต่อกับห้องสมุด โดยใช้บัตรห้องสมุด ที่ห้องสมุดออกให้

### 3.1.3). สารนิเทศ

ปัจจุบันห้องสมุดมีสารนิเทศสำหรับให้บริการยืม เฉพาะสิ่งพิมพ์ประเภทหนังสือ(Book Title) และสื่อประกอบอื่นๆ (เช่น โสตทัศนศึกษาและโปรแกรม ที่มาที่หนังสือ) โดยการยืมหนังสือนั้น สมาชิกประเภทที่ต่างกันจะมีกำหนดเวลาในการยืม และค่าปรับในการที่ถูกส่งคืนช้ากว่าที่กำหนดที่แตกต่างกัน ระบบจะต้องให้บริการยืมเจ้าหน้าที่ให้บริการยืมคืนหนังสือ และคืนหนังสือตามกฎเกณฑ์ ต่างๆที่ได้กำหนดไว้ และระบบจะต้องสามารถเพิ่ม ลบและแก้ไขข้อมูลหนังสือในห้องสมุดได้ โดยหนังสือแต่ละเล่มจะมีแถบรหัสเป็นของตัวเอง โดยสำหรับรายการหนังสือหนึ่งๆ อาจจะมีหนังสือหลายเล่มก็ได้ และหนังสือแต่ละเล่มอาจจะมีสถานะเป็น"สูญหาย"ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การวิเคราะห์ความต้องการของระบบ

หลังจากที่ได้ทำการพิจารณาถึงข้อความแสดงความต้องการของระบบ (Requirement Statement) เรียบร้อยแล้วเราจะสามารถจำแนกสิ่งต่างๆออกไปนี้ออกมาได้ เพื่อนำไปใช้ในการสร้างข้อกำหนดของ Use Case (Use Case Definitions) ต่อไป

#### 3.2.1). คำนาม

เป็นการนำคำนามที่ได้จากถึงข้อความแสดงความต้องการของระบบ (Requirement Statement) ข้างต้น มาทำการจำแนกประเภทของคำนามนั้น ซึ่งประกอบด้วย Actor คำที่มีความหมายใกล้เคียงกัน (Alias) วัตถุ หรือ ข้อมูลของตัววัตถุ (Object, Attribute) เหตุการณ์ (Event) และไม่สามารถระบุได้ (N/A)

ซึ่งจากแนวทางข้างต้นทำให้พบวัตถุในระบบในขั้นตอนนี้คือ สมาชิกห้องสมุด รายการหนังสือ หนังสือแต่ละเล่ม รายการการยืมหนังสือ ข้อกำหนดในการยืมหนังสือ ผู้แต่ง การจองหนังสือ และ เจ้าหน้าที่

#### 3.2.2). Actor และ Use Case

สำหรับ Actor ที่ได้รับการพิจารณาในการวิเคราะห์ความต้องการของระบบ ได้แก่ สมาชิกห้องสมุด (Member) บรรณารักษ์ (Librarian) และ เจ้าหน้าที่ให้บริการยืมคืน (Circulation Clerk) ส่วน use case ที่จำแนกตาม actor ที่ดำเนินการ use case นั้นๆได้แก่

##### สมาชิกห้องสมุด (Borrower)

- (1) เรียกดูรายการหนังสือ (Browse Catalog) – เรียกดูรายการหนังสือทั้งหมดในฐานข้อมูลสารสนเทศของห้องสมุดตามเงื่อนไขที่กำหนด
- (2) ตรวจสอบรายการยืมหนังสือ (Check Circulation Record) – เรียกดูรายการหนังสือทั้งหมดที่สมาชิกทำการยืม

##### บรรณารักษ์ (Librarian)

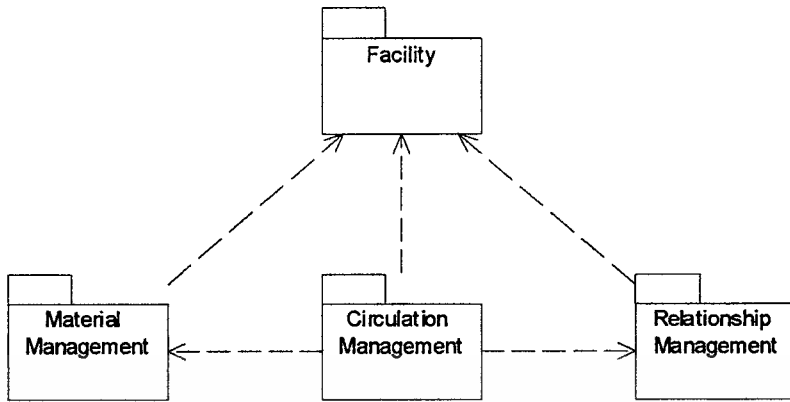
- (1) เข้าสู่ระบบ (Log In)
- (2) เพิ่มรายการหนังสือ (Add Title) -- เพิ่มรายการหนังสือเข้ามายังห้องสมุด
- (3) ลบรายการหนังสือ (Remove Title) -- ลบรายการหนังสือออกจากห้องสมุด
- (4) เพิ่มหนังสือ (Add Item) -- เพิ่มหนังสือเข้ามายังห้องสมุด

- (5) ลบหนังสือ (Remove Item) -- ลบหนังสือออกจากห้องสมุด
- (6) เพิ่มสมาชิกห้องสมุด (Add Member) -- เพิ่มสมาชิกห้องสมุดเข้าไปในระบบ
- (7) ลบสมาชิกห้องสมุด (Remove Member) -- ลบสมาชิกห้องสมุดออกจากระบบ
- (8) ต่ออายุการเป็นสมาชิก (Extended Expire Date) -- ต่ออายุของสมาชิกแต่ละประเภทในช่วงเริ่มต้นภาคการศึกษาใหม่
- (7) เรียกดูรายการหนังสือ (Browse Catalog) – เรียกดูรายการหนังสือทั้งหมดในฐานข้อมูลสารสนเทศของห้องสมุดตามเงื่อนไขที่กำหนด
- (8) เรียกดูสมาชิก (Browse Member) – เรียกดูสมาชิกทั้งหมดในฐานข้อมูลสารสนเทศของห้องสมุดตามเงื่อนไขที่กำหนด

#### เจ้าหน้าที่ให้บริการยืม-คืน (Circulation Clerk)

- (1) เข้าสู่ระบบ (Log In)
- (2) ยืมหนังสือ (Lend Item)--ทำรายการยืมหนังสือสำหรับสมาชิกที่มาทำการยืมหนังสือ
- (3) คืนหนังสือ (Return Item) -- ทำรายการคืนหนังสือสำหรับสมาชิกที่นำหนังสือมาคืน
- (4) จองหนังสือ (Make Reservation) -- ทำรายการจองหนังสือที่สมาชิกต้องการยืมแต่หนังสือนั้นถูกยืมไป
- (5) ยกเลิกการจองหนังสือ (Remove Reservation) -- ยกเลิกรายการจองหนังสือที่สมาชิกทำการจองเอาไว้
- (6) รับชำระค่าปรับ (Pay Fine) – รับชำระค่าปรับสำหรับสมาชิกที่มีค่าปรับในการยืมหนังสือ
- (7) เรียกดูรายการหนังสือ (Browse Catalog) – เรียกดูรายการหนังสือทั้งหมดในฐานข้อมูลสารสนเทศของห้องสมุดตามเงื่อนไขที่กำหนด
- (8) เรียกดูสมาชิก (Browse Member) – เรียกดูสมาชิกทั้งหมดในฐานข้อมูลสารสนเทศของห้องสมุดตามเงื่อนไขที่กำหนด

จากรายละเอียดของ Use Case ที่กล่าวมาทั้งหมดจะทำการอธิบายโดยใช้แบบจำลองของ UML คือแผนภาพ Use Case โดยจะเป็นการมองจากระดับมุมมองที่กว้างลงมายังรายละเอียดโดยใช้สัญลักษณ์ Use Case Package มาช่วยในการสร้างแบบจำลองดังรูปที่ 3.1



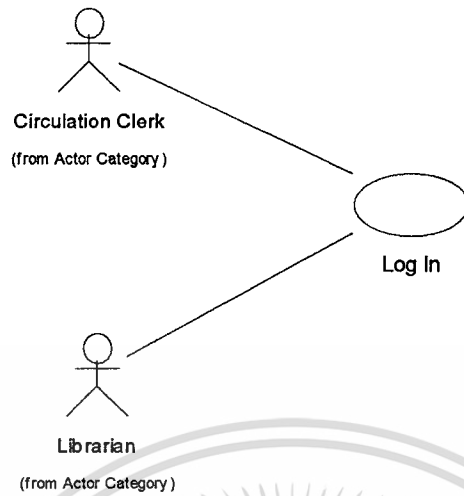
รูปที่ 3.1 แผนภาพ Use Case Package ของระบบที่จะทำการพัฒนา

รูปที่ 3.1 แสดงให้เห็นถึงแผนภาพ Use Case ในระดับสูง (High Abstraction) ของระบบที่จะทำการพัฒนา โดยแสดงในรูปของ UML Use Case Package ซึ่งจะมีความหมายแทนระบบงานย่อยๆที่อยู่ในระบบงานที่กำลังพิจารณา โดยเส้นประที่แสดงในรูปจะแทนความสัมพันธ์ในรูปแบบที่ขึ้นต่อกัน (Dependency Relationship) ของแต่ละ Use Case Package เพื่อที่จะทำให้ง่ายต่อความเข้าใจและสามารถที่จะแบ่งรายละเอียดของหน้าที่ในส่วนที่มีความสัมพันธ์กันไว้ในแพ็คเกจเดียวกัน (หลังจากขั้นตอนการออกแบบ Use Case ที่อยู่ใน Package เดียวกันมีแนวโน้มที่จะถูกบรรจุเอาไว้ในคอมโพเนนต์เดียวกันด้วย) โดยที่รายละเอียดของแต่ละแพ็คเกจนั้นมีดังต่อไปนี้

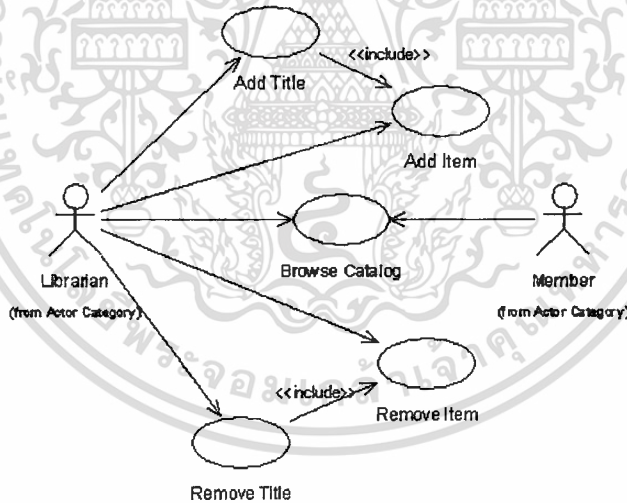
(1). แพ็คเกจส่วนงานอำนวยความสะดวกทั่วไป (Facility Package) เป็นส่วนงานย่อยๆที่มีหน้าที่ในการให้บริการทั่วไปของระบบ สำหรับแผนภาพ Use Case ของแพ็คเกจส่วนนี้ในระบบที่จะทำการพัฒนา จะประกอบด้วยหน้าที่ในการตรวจสอบสิทธิในการใช้ระบบของผู้ใช้ซึ่งได้แก่ บรรณารักษ์ และ เจ้าหน้าที่ให้บริการยืม-คืนหนังสือ ดังมีรายละเอียดตามแผนภาพ Use Case ดังรูปที่ 3.2

(2). แพ็คเกจส่วนงานจัดการวัสดุสารนิเทศ (Material Management Package) เป็นส่วนงานย่อยๆที่มีหน้าที่ในการจัดการหนังสือที่มีในห้องสมุด รวมไปถึงการให้บริการค้นหารายการหนังสือต่างๆ (Catalog) ในห้องสมุดแก่สมาชิกที่เข้ามาใช้บริการอีกด้วย ดังมีรายละเอียดตามแผนภาพ Use Case ดังรูปที่ 3.3

(3). แพ็คเกจส่วนงานจัดการการยืม-คืนหนังสือ (Circulation Management Package) เป็นส่วนงานย่อยๆที่มีหน้าที่ในการจัดการการยืม-คืนหนังสือของสมาชิก รวมไปถึงการให้บริการของหนังสือ ดังมีรายละเอียดตามแผนภาพ Use Case ดังรูปที่ 3.4



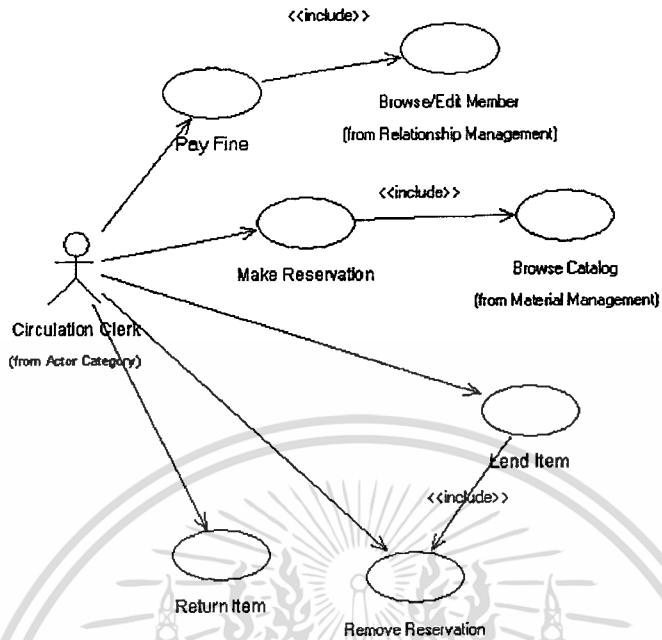
รูปที่ 3.2 แผนภาพ Use Case ของแพ็คเกจส่วนงานอำนวยความสะดวกทั่วไป



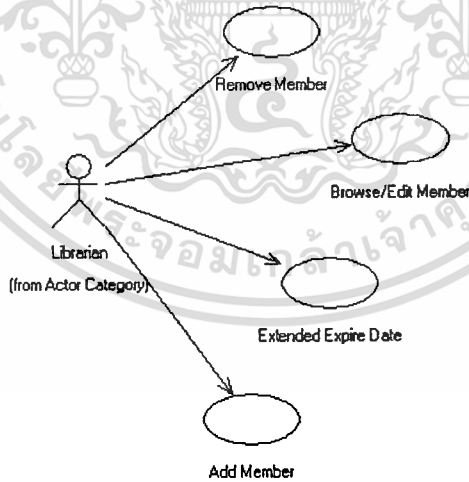
รูปที่ 3.3 แผนภาพ Use Case ของแพ็คเกจส่วนงานจัดการวัสดุสารนิเทศ

(4). แพ็คเกจส่วนงานจัดการสมาชิก (Relationship Management Package) เป็นส่วนงานย่อยๆที่มีหน้าที่ในการจัดการข้อมูลต่างๆของสมาชิก ดังมีรายละเอียดตามแผนภาพ Use Case ดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แผนภาพ Use Case ของแพ็คเกจส่วนงานจัดการการยืม-คืนสารนิเทศ



รูปที่ 3.5 แผนภาพ Use Case ของแพ็คเกจส่วนงานจัดการสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3). ข้อกำหนดของ Use Case (Use Case Definitions)

สำหรับ Use Case ที่ได้จากการวิเคราะห์ จะต้องแสดง Input และผลลัพธ์ที่ได้และข้อมูลที่คืนมาพร้อมผลลัพธ์ของแต่ละ Use Case โดยข้อกำหนดของ Use Case ที่ได้จากขั้นตอนนี้จะถูกนำไปใช้เป็นข้อความที่ระบุถึงความต้องการของระบบอย่างเป็นทางการ (Formal Statement of Requirement) ตลอดทั้งโครงการ รวมไปถึงในการบำรุงรักษาด้วย ผู้จัดการโครงการอาจจะใช้ข้อกำหนดของ Use Case นี้ในการวางแผนการพัฒนาโครงการ หรือทีมงานพัฒนาส่วนติดต่อผู้ใช้อาจจะใช้ข้อกำหนดของ Use Case นี้ในการสร้างส่วนที่ทำการติดต่อระหว่างส่วนติดต่อผู้ใช้กับตัว Application นอกจากนี้ ทีมงานทดสอบจะใช้ข้อกำหนดของ Use Case นี้ในการวางแผนในการทดสอบระบบ

Use Case	เข้าใช้ระบบ (Log In)
Inputs	หมายเลขประจำตัว, รหัสผ่าน
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบหมายเลขประจำตัวนี้ในระบบ, รหัสผ่านไม่ถูกต้อง
Use Case	เพิ่มสมาชิกห้องสมุด (Add Member)
Inputs	เลขประจำตัวสมาชิก, ประเภทสมาชิก, ชื่อสมาชิก
ผลลัพธ์ (Outcomes)	ตกลง, พบเลขประจำตัวสมาชิกนี้ในระบบอยู่แล้ว, ไม่พบประเภทสมาชิกนี้ในระบบ
Use Case	เพิ่มรายการหนังสือ (Add Title)
Inputs	เลขเรียกหนังสือ, ประเภทหนังสือ, สำนักพิมพ์, ISBN, ชื่อหนังสือ
ผลลัพธ์ (Outcomes)	ตกลง, มีเลขเรียกหนังสือในระบบอยู่แล้ว, มี ISBNในระบบอยู่แล้ว, ไม่พบประเภทหนังสือนี้ในระบบ, ไม่พบสำนักพิมพ์นี้ในระบบ, มีชื่อหนังสือนี้อยู่แล้วในระบบ

Use Case	เพิ่มหนังสือ (Add Item)
Inputs	รหัสบาร์โค้ด, เลขเรียกหนังสือ
ผลลัพธ์ (Outcomes)	ตกลง, พบเลขเรียกหนังสือในระบบ, พบเลขทะเบียนหนังสือในระบบ
Use Case	ยืมสารนิเทศ (Lend Item)
Inputs	รหัสบาร์โค้ด, รหัสประจำตัวเจ้าหน้าที่ที่ยืม, รหัสประจำตัวสมาชิก
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบหนังสือเล่มนี้ในระบบ, ไม่พบสมาชิกนี้ในระบบ, สมาชิกไม่มีสิทธิในการยืม, หนังสือเล่มนี้ถูกจอง
Use Case	คืนหนังสือ (Return Item)
Inputs	รหัสบาร์โค้ด
ผลลัพธ์ (Outcomes)	ตกลง, ส่งเกินกำหนด (Overdue) – (วัน), ถูกจองโดย (สมาชิก), ส่งเกินกำหนด (วัน) และ ถูกจอง (สมาชิก), ไม่พบหนังสือเล่มนี้ในระบบ, หนังสือเล่มนี้ไม่ได้ถูกยืม, ไม่พบสมาชิกนี้ในระบบ
Use Case	จองหนังสือ (Make Reservation)
Inputs	เลขประจำตัวสมาชิก, เลขประจำตัวเจ้าหน้าที่ที่ทำการจอง, รหัสบาร์โค้ด
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบหนังสือนี้ในระบบ, ไม่พบสมาชิกนี้ในระบบ, สมาชิกไม่มีสิทธิ
Use Case	ลบสมาชิก (Remove Member)
Inputs	รหัสประจำตัวสมาชิก
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบเลขประจำตัวสมาชิกนี้ในระบบอยู่แล้ว

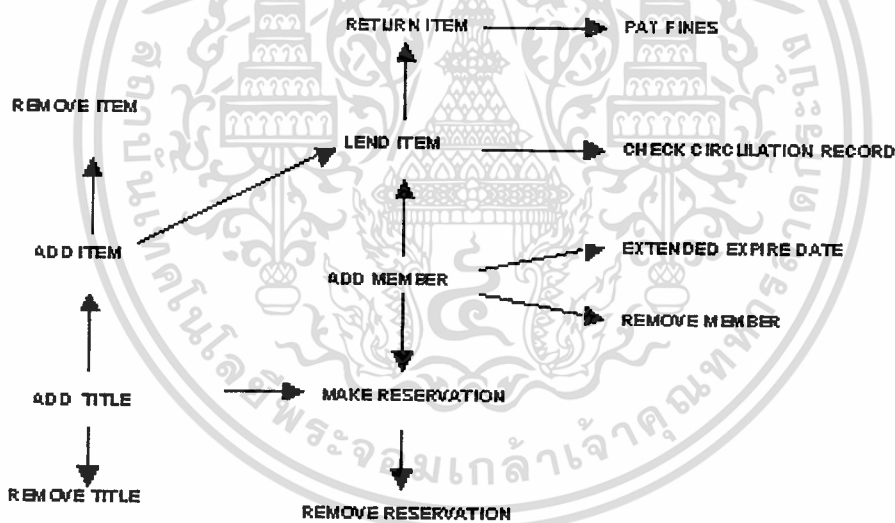
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Case	ลบรายการหนังสือ (Remove Title)
Inputs	เลขเรียกหนังสือ
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบเลขเรียกหนังสือในระบบ
Use Case	ลบหนังสือ (Remove Item)
Inputs	รหัสบาร์โค้ด
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบหนังสือที่มีบาร์โค้ดนี้ในระบบ
Use Case	ยกเลิกการจองหนังสือ (Remove Reservation)
Inputs	เลขประจำตัวสมาชิก, รหัสบาร์โค้ด
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบหนังสือนี้ในระบบ, ไม่พบสมาชิกนี้ในระบบ
Use Case	รับชำระค่าปรับ (Pay Fine)
Inputs	เลขประจำตัวสมาชิก
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบสมาชิกนี้ในระบบ, สมาชิกไม่ได้ค้างชำระค่าปรับ
Use Case	ต่ออายุการเป็นสมาชิก (Extended Expire Date)
Inputs	เลขประจำตัวสมาชิก
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบสมาชิกนี้ในระบบ, สมาชิกไม่มีสิทธิต่ออายุ
Use Case	เรียกดูรายการหนังสือ (Browse Catalog)
Inputs	ชื่อเรื่อง, ชื่อผู้แต่ง, เลขเรียกหนังสือ, หัวเรื่อง
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบหนังสือนี้ในระบบ

Use Case	เรียกดูสมาชิก (Browse Member)
Inputs	ชื่อสมาชิก, หมายเลขประจำตัว, ประเภทสมาชิก, คณะ
ผลลัพธ์ (Outcomes)	ตกลง, ไม่พบสมาชิกนี้ในระบบ

### 3.3 การวางแผนในการแบ่งการพัฒนาแบบเพิ่มทีละขั้น (Incremental Plan)

ในการวางแผนในการพัฒนาแบบเพิ่มทีละขั้นนั้น เราจะกำหนดขั้นในการพัฒนา โดยใช้ Use Case เป็นหลัก ทั้งนี้การพัฒนาในแต่ละขั้นจะทำการพัฒนา Use Case ที่ขึ้นอยู่กับ Use Case ที่อยู่ในการพัฒนาขั้นเดียวกันหรือ ขั้นตอนก่อนหน้าเท่านั้น ซึ่งการขึ้นต่อกันของ Use Case ในการพัฒนาระบบสารสนเทศเพื่อการบริหารห้องสมุดนี้ ได้ถูกแสดงในรูปที่ 3.3 ส่วนแผนในการพัฒนา จะได้แสดงไว้ดังต่อไปนี้พร้อมทั้งเหตุผล



รูปที่ 3.6 แผนภาพแสดงการขึ้นต่อกันของ Use Case ในการพัฒนาระบบ

#### การพัฒนาในขั้นที่ 1 (Increment 1)

(เหตุผล -- เลือกทำ Use Case ที่ง่ายและ มีการทำงานที่มีผลกับ use case อื่นๆ เพื่อที่จะ ได้ขั้นตอน การพัฒนาที่ค่อนข้างสั้นทำให้มีประโยชน์ในการเรียนรู้ในการพัฒนาระบบในช่วงแรกนี้) Use Case ที่ถูกเลือกมาทำการพัฒนาในขั้นที่ 1 นี้ได้แก่

- เข้าใช้ระบบ (Log In)
- เพิ่มสมาชิกห้องสมุด (Add Member)
- เพิ่มรายการหนังสือ (Add Title)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่มหนังสือ (Add Item)
- เรียกดูรายการหนังสือ (Browse Catalog)
- เรียกดูสมาชิก (Browse Member)

### การพัฒนาในขั้นที่ 2 (Increment 2)

(เหตุผล -- เลือก Use Case ที่ทำให้เห็นถึงความสามารถหลักๆของระบบเพื่อที่จะสามารถประเมินความสามารถและประสิทธิภาพของระบบในขั้นต้นได้) โดย Use Case ที่ถูกเลือกมาทำการพัฒนาในขั้นที่ 2 นี้ได้แก่

- ต่ออายุสมาชิก (Extended Expire Date)
- รับชำระค่าปรับ (Pay Fine)
- ยืมหนังสือ (Lend Item)
- คืนหนังสือ (Return Item)
- จองหนังสือ (Make Reservation)
- ยกเลิกการจองหนังสือ (Remove Reservation)
- ตรวจสอบการยืมหนังสือ (Check Circulation Record)

### การพัฒนาในขั้นที่ 3 (Increment 3)

(เหตุผล -- เลือก Use Case ที่ทำให้เห็นถึงความสามารถหลักๆของระบบแต่มีลำดับความสำคัญน้อยกว่า Use Case ที่เลือกในการพัฒนาขั้นที่ 2) โดย Use Case ที่ถูกเลือกมาทำการพัฒนาในขั้นที่ 3 นี้ได้แก่

- ลบสมาชิกห้องสมุด (Remove Member)
- ลบรายการหนังสือ (Remove Title)
- ลบหนังสือ (Remove Item)

## บทที่ 4

### การดำเนินการพัฒนาระบบ

#### 4.1 รายละเอียดของ Use Case (Use Case Description)

จากข้อกำหนดของ Use Case ที่ได้แสดงในหัวข้อ 3.2.3 เราจะนำ Use Case ที่จะทำการพัฒนาในแต่ละขั้นตอนของการพัฒนามาทำการบันทึกรายละเอียดของ Use Case (Use Case Description) ที่มีรายละเอียดของกฎทางธุรกิจซึ่งจะอธิบายเหตุการณ์ในรูปของเงื่อนไขก่อนเกิดเหตุการณ์นั้นๆ ขึ้น (Pre-Condition) และการตอบสนองของระบบ โดยสำหรับ use case ที่มีรายละเอียดที่ค่อนข้างซับซ้อนจะทำการแสดงกฎทางธุรกิจในรูปของตารางตัดสินใจ (Decision Table) โดยรายละเอียดของ use case ที่จะทำการพัฒนาตามรายชื่อ use case ต่อไปนี้จะถูกแสดงเอาไว้ในหัวข้อที่ 4.1.1 ถึง 4.1.16 ตามลำดับ

- เข้าสู่ระบบ (Log In)
- เพิ่มสมาชิกห้องสมุด (Add Member)
- เพิ่มรายการหนังสือ (Add Title)
- เพิ่มหนังสือ (Add Item)
- เรียกดูรายการหนังสือ (Browse Catalog)
- เรียกดูสมาชิก (Browse Member)
- ตรวจสอบการยืมหนังสือ (Check Circulation Record)
- ต่ออายุสมาชิก (Extended Expire Date)
- รับชำระค่าปรับ (Pay Fine)
- ยืมหนังสือ (Lend Item)
- คืนหนังสือ (Return Item)
- จองหนังสือ (Make Reservation)
- ยกเลิกการจองหนังสือ (Remove Reservation)
- ลบสมาชิกห้องสมุด (Remove Member)
- ลบรายการหนังสือ (Remove Title)
- ลบหนังสือ (Remove Item)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.1 Use Case : เข้าสู่ระบบ (Log In)

Actor : บรรณารักษ์ (Librarian), เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk)

Description : บรรณารักษ์และเจ้าหน้าที่รับยืมคืนหนังสือ เข้าสู่ระบบโดยใช้รหัสประจำตัวและรหัสผ่าน

Pre-Condition and System Action :

ตารางที่ 4.1 ตารางแสดง Decision Table สำหรับ Use Case – Log In

Scenarios	1	2	3				
<b>Conditions</b>							
พบเจ้าหน้าที่นี้ในระบบ	F	T	T				
รหัสผ่านถูกต้อง		F	F				
<b>Actions</b>							
ปฏิบัติการเข้าสู่ระบบ	X	X					
บันทึกการเข้าสู่ระบบ			X				

คำอธิบายเหตุการณ์ :

- 1) เจ้าหน้าที่ทำการขอเข้าสู่ระบบแต่ไม่พบรหัสเจ้าหน้าที่ดังกล่าวในระบบจึงทำการปฏิบัติการเข้าสู่ระบบ
- 2) เจ้าหน้าที่ทำการขอเข้าสู่ระบบ ระบบค้นพบเจ้าหน้าที่ดังกล่าวในระบบแต่พบว่ารหัสผ่านที่อยู่ในระบบไม่ตรงกับที่ป้อนจึงทำการปฏิบัติการเข้าสู่ระบบ
- 3) เจ้าหน้าที่ทำการขอเข้าสู่ระบบ ระบบค้นพบเจ้าหน้าที่ดังกล่าวในระบบและรหัสผ่านที่อยู่ในระบบตรงกับที่ป้อนจึงอนุญาตให้เข้าสู่ระบบ และทำการบันทึกการเข้าใช้งานระบบ

#### 4.1.2 Use Case : เพิ่มสมาชิกห้องสมุด (Add Member)

Actor : บรรณารักษ์ (Librarian)

Description : บรรณารักษ์ทำการเพิ่มข้อมูลสมาชิกใหม่เข้าสู่ระบบ

Pre-Condition and System Action :

ตารางที่ 4.2 ตารางแสดง Decision Table สำหรับ Use Case – Add Member

Scenarios	1	2	3						
<b>Conditions</b>									
พบเลขประจำตัวสมาชิกนี้ในระบบ	T	F	F						
พบประเภทสมาชิกนี้ในระบบ		F	T						
<b>Actions</b>									
ปฏิเสธการดำเนินการ	X	X							
เพิ่มข้อมูลสมาชิกนั้นลงในระบบ			X						

คำอธิบายเหตุการณ์ :

- 1) บรรณารักษ์ทำการเพิ่มข้อมูลสมาชิกใหม่เข้าสู่ระบบแต่ระบบพบว่า มีสมาชิกที่มีเลขประจำตัวดังกล่าวในระบบอยู่แล้วจึงทำการปฏิเสธการดำเนินการ
- 2) บรรณารักษ์ทำการเพิ่มข้อมูลสมาชิกใหม่เข้าสู่ระบบ และระบบพบว่ายังไม่มีสมาชิกที่มีเลขประจำตัวดังกล่าวในระบบ แต่ไม่พบประเภทสมาชิกในระบบจึงทำการปฏิเสธการดำเนินการ
- 3) บรรณารักษ์ทำการเพิ่มข้อมูลสมาชิกใหม่เข้าสู่ระบบ และระบบพบว่ายังไม่มีสมาชิกที่มีเลขประจำตัวดังกล่าวในระบบ และพบประเภทสมาชิกในระบบ จึงทำการเพิ่มข้อมูลสมาชิกนั้นเข้าสู่ระบบ

### 4.1.3 Use Case : เพิ่มรายการหนังสือ (Add Title)

Actor : บรรณารักษ์ (Librarian)

Description : บรรณารักษ์ทำการเพิ่มข้อมูลรายการหนังสือใหม่เข้าสู่ระบบ

Pre-Condition and System Action :

ตารางที่ 4.3 ตารางแสดง Decision Table สำหรับ Use Case – Add Title

Scenarios	1	2	3	4				
<b>Conditions</b>								
พบเลขเรียกหนังสือในระบบ	T	F	F	F				
พบประเภทหนังสือในระบบ		F	T	T				
พบสำนักพิมพ์ในระบบ			F	T				
<b>Actions</b>								
ปฏิบัติการดำเนินการ	X	X	X					
เพิ่มข้อมูลรายการหนังสือลงในระบบ				X				

คำอธิบายเหตุการณ์ :

- 1) บรรณารักษ์ทำการเพิ่มข้อมูลรายการหนังสือใหม่เข้าสู่ระบบแต่ระบบพบว่ามีการหนังสือที่มีเลขเรียกหนังสือดังกล่าวในระบบอยู่แล้วจึงทำการปฏิบัติการดำเนินการ
- 2) บรรณารักษ์ทำการเพิ่มข้อมูลรายการหนังสือใหม่เข้าสู่ระบบ และระบบพบว่ายังไม่มีรายการหนังสือที่มีเลขเรียกหนังสือดังกล่าวในระบบ แต่ไม่พบประเภทหนังสือในระบบจึงทำการปฏิบัติการดำเนินการ
- 3) บรรณารักษ์ทำการเพิ่มข้อมูลรายการหนังสือใหม่เข้าสู่ระบบ และระบบพบว่ายังไม่มีรายการหนังสือที่มีเลขเรียกหนังสือดังกล่าวในระบบ และพบประเภทหนังสือในระบบ แต่ไม่พบสำนักพิมพ์ในระบบจึงทำการปฏิบัติการดำเนินการ
- 4) บรรณารักษ์ทำการเพิ่มข้อมูลรายการหนังสือใหม่เข้าสู่ระบบ ระบบพบประเภทหนังสือและสำนักพิมพ์ในระบบ และพบว่ายังไม่มีรายการหนังสือที่มีเลขเรียกหนังสือดังกล่าวในระบบ จึงทำการเพิ่มข้อมูลรายการหนังสือเข้าสู่ระบบ

#### 4.1.4 Use Case : เพิ่มหนังสือ (Add Item)

Actor : บรรณารักษ์ (Librarian)

Description : บรรณารักษ์ทำการเพิ่มข้อมูลหนังสือใหม่เข้าสู่ระบบ

Pre-Condition and System Action :

ตารางที่ 4.4 ตารางแสดง Decision Table สำหรับ Use Case – Add Item

Scenarios	1	2	3					
<b>Conditions</b>								
พบหมายเลขบาร์โค้ดในระบบ	T	F	F					
พบรายการหนังสือในระบบ		F	T					
<b>Actions</b>								
ปฏิเสธการดำเนินการ	X	X						
เพิ่มข้อมูลหนังสือลงในระบบ			X					

คำอธิบายเหตุการณ์ :

- 1) บรรณารักษ์ทำการเพิ่มข้อมูลหนังสือใหม่เข้าสู่ระบบแต่ระบบพบว่าไม่มีหนังสือที่มีหมายเลขบาร์โค้ดดังกล่าวในระบบอยู่แล้วจึงทำการปฏิเสธการดำเนินการ
- 2) บรรณารักษ์ทำการเพิ่มข้อมูลรายการหนังสือใหม่เข้าสู่ระบบ และระบบพบว่ายังไม่มีหนังสือที่มีหมายเลขบาร์โค้ดดังกล่าวในระบบ แต่ไม่พบรายการหนังสือในระบบจึงทำการปฏิเสธการดำเนินการ
- 3) บรรณารักษ์ทำการเพิ่มข้อมูลรายการหนังสือใหม่เข้าสู่ระบบ และพบรายการหนังสือในระบบ และระบบพบว่ายังไม่มีหนังสือที่มีหมายเลขบาร์โค้ดดังกล่าวในระบบ จึงทำการเพิ่มข้อมูลรายการหนังสือเข้าสู่ระบบ

#### 4.1.5 Use Case : เรียกดูรายการหนังสือ (Browse Catalog)

**Actor :** บรรณารักษ์ (Librarian), เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk), สมาชิกห้องสมุด

**Description :** Actor เรียกดูรายการหนังสือที่มีในระบบตามเงื่อนไขต่างๆ

**Basic Flow :**

- 1) Use case นี้เริ่มต้นเมื่อ Actor ทำการเรียกดูรายการหนังสือจากเงื่อนไข เช่น ชื่อเรื่อง, ชื่อผู้แต่ง, หัวเรื่อง, เลขเรียกหนังสือ
- 2) ระบบทำการแสดงรายการหนังสือทั้งหมดที่มีเงื่อนไขตรงตามที่กำหนด
- 3) Actor ทำการเลือกรายการหนังสือที่ต้องการทราบรายละเอียด
- 4) ระบบแสดงรายละเอียดของหนังสือที่สอดคล้องกับรายการหนังสือที่ Actor ทำการเลือก

#### 4.1.6 Use Case : เรียกดูสมาชิก (Browse Member)

**Actor :** บรรณารักษ์ (Librarian), เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk)

**Description :** Actor เรียกดูสมาชิกที่มีในระบบตามเงื่อนไขต่างๆ

**Basic Flow :**

- 1) Use case นี้เริ่มต้นเมื่อ Actor ทำการเรียกดูสมาชิกจากเงื่อนไข เช่น ชื่อสมาชิก, ประเภทสมาชิก
- 2) ระบบทำการแสดงสมาชิกทั้งหมดที่มีเงื่อนไขตรงตามที่กำหนด
- 3) Actor ทำการเลือกสมาชิกที่ต้องการทราบรายละเอียด
- 4) ระบบแสดงรายละเอียดของสมาชิกที่สอดคล้องกับสมาชิกที่ Actor ทำการเลือก

#### 4.1.7 Use Case : ตรวจสอบการยืมหนังสือ (Check Circulation Record)

**Actor :** สมาชิกห้องสมุด

**Description :** สมาชิกห้องสมุดเรียกดูรายการยืมหนังสือของตน

**Basic Flow :**

- 1) Use case นี้เริ่มต้นเมื่อสมาชิกห้องสมุดทำการเรียกดูรายการยืมหนังสือของตน
- 2) ระบบทำการแสดงรายการยืมทั้งหมดที่ยังไม่ได้ทำการคืนที่เป็นของสมาชิกคนนั้น

#### 4.1.8 Use Case : ต่ออายุสมาชิก (Extended Expire Date)

Actor : บรรณารักษ์ (Librarian)

Description : บรรณารักษ์ทำการปรับสถานะของสมาชิกที่หมดอายุในแต่ละปีการศึกษาเพื่อให้สามารถใช้บริการห้องสมุดได้ตามปกติ

Pre-Condition and System Action :

ตารางที่ 4.5 ตารางแสดง Decision Table สำหรับ Use Case – Extended Expire Date

Scenarios	1	2	3				
<b>Conditions</b>							
พบสมาชิกในระบบ	F	T	T				
สมาชิกหมดอายุ		F	T				
<b>Actions</b>							
ปฏิเสธการดำเนินการ	X	X					
ปรับสถานะการต่ออายุของสมาชิก			X				

คำอธิบายเหตุการณ์ :

- 1) บรรณารักษ์ทำการปรับสถานะการต่ออายุของสมาชิก แต่ระบบไม่พบสมาชิกในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) บรรณารักษ์ทำการปรับสถานะการต่ออายุของสมาชิก ระบบพบสมาชิกในระบบ แต่พบว่าสมาชิกได้ทำการปรับสถานะการต่ออายุเรียบร้อยแล้วจึงทำการปฏิเสธการดำเนินการ
- 3) บรรณารักษ์ทำการปรับสถานะการต่ออายุของสมาชิก ระบบพบสมาชิกในระบบ และพบว่าสมาชิกยังไม่ได้ทำการปรับสถานะการต่ออายุจึงทำการปรับสถานะการต่ออายุของสมาชิก

#### 4.1.9 Use Case : รับชำระค่าปรับ (Pay Fine)

**Actor :** เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk)

**Description :** เจ้าหน้าที่รับยืมคืนหนังสือทำการปรับยอดค่าปรับของสมาชิก

**Pre-Condition and System Action :**

ตารางที่ 4.6 ตารางแสดง Decision Table สำหรับ Use Case – Pay Fine

Scenarios	1	2	3					
<b>Conditions</b>								
พบสมาชิกในระบบ	F	T	T					
สมาชิกมีค่าปรับ		F	T					
<b>Actions</b>								
ปฏิเสธการดำเนินการ	X	X						
ปรับยอดค่าปรับของสมาชิก			X					

คำอธิบายเหตุการณ์ :

- 1) เจ้าหน้าที่รับยืมคืนหนังสือทำการปรับยอดค่าปรับของสมาชิก แต่ระบบ ไม่พบสมาชิกในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) เจ้าหน้าที่รับยืมคืนหนังสือทำการปรับยอดค่าปรับของสมาชิก ระบบพบสมาชิกในระบบ แต่พบว่าสมาชิกไม่มีค่าปรับจึงทำการปฏิเสธการดำเนินการ
- 3) เจ้าหน้าที่รับยืมคืนหนังสือทำการปรับยอดค่าปรับของสมาชิก ระบบพบสมาชิกในระบบ และพบว่าสมาชิกมีค่าปรับจึงทำการปรับยอดค่าปรับของสมาชิก

#### 4.1.10 Use Case : ยืมหนังสือ (Lend Item)

Actor : เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk)

Description : เจ้าหน้าที่รับยืมคืนหนังสือทำการเพิ่มรายการยืมหนังสือของสมาชิกเข้าสู่ระบบ

Pre-Condition and System Action :

ตารางที่ 4.7 ตารางแสดง Decision Table สำหรับ Use Case – Lend Item

Scenarios	1	2	3	4	5
<b>Conditions</b>					
พบสมาชิกในระบบ	F	T	T	T	T
สมาชิกมีสิทธิในการยืม		F	T	T	T
หนังสือเล่มนี้มีรายการจอง			F	T	T
สมาชิกนี้มีชื่ออยู่ในรายการจองอันดับแรก				F	T
<b>Actions</b>					
ปฏิเสธการดำเนินการ	X	X		X	
กำหนดสถานะหนังสือเล่มนี้เป็นถูกยืม			X	X	
เพิ่มจำนวนการยืมสำหรับสมาชิกรายนี้			X	X	
เพิ่มรายการยืมหนังสือของสมาชิกเข้าสู่ระบบ			X	X	
ลบสมาชิกรายนี้ออกจากรายการจองของหนังสือเล่มนี้				X	

คำอธิบายเหตุการณ์ :

- 1) เจ้าหน้าที่รับยืมคืนหนังสือทำการเพิ่มรายการยืมหนังสือของสมาชิกเข้าสู่ระบบ แต่ระบบไม่พบสมาชิกในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) เจ้าหน้าที่รับยืมคืนหนังสือทำการเพิ่มรายการยืมหนังสือของสมาชิกเข้าสู่ระบบ ระบบพบสมาชิกในระบบ แต่พบว่าสมาชิกไม่มีสิทธิในการยืม(เนื่องจากสมาชิกทำการยืมสารนิเทศไปเป็นจำนวนเท่าที่กำหนด หรือมีหนังสือที่สมาชิกยืมไปเกินใดที่เกินกำหนดส่ง หรือสมาชิกมีค่าปรับที่ค้างชำระกับทางห้องสมุด)ระบบจึงทำการปฏิเสธการดำเนินการ
- 3) เจ้าหน้าที่รับยืมคืนหนังสือทำการเพิ่มรายการยืมหนังสือของสมาชิกเข้าสู่ระบบ ระบบพบสมาชิกในระบบ และสมาชิกมีสิทธิในการยืม และหนังสือเล่มนี้ไม่ได้ถูกจอง ระบบจึงทำ

การกำหนดสถานะของหนังสือเล่มนี้เป็นถูกยืม และเพิ่มจำนวนหนังสือที่สมาชิกทำการยืม และบันทึกรายการยืมลงในระบบ

- 4) เจ้าหน้าที่รับยืมคืนหนังสือทำการเพิ่มรายการยืมหนังสือของสมาชิกเข้าสู่ระบบ ระบบพบสมาชิกในระบบ และสมาชิกมีสิทธิในการยืม แต่พบว่าหนังสือเล่มนี้ถูกจอง แต่สมาชิกไม่ได้ อยู่ในคิวการจองอันดับแรก ดังนั้นระบบจึงทำการปฏิเสธการดำเนินการ
- 5) เจ้าหน้าที่รับยืมคืนหนังสือทำการเพิ่มรายการยืมหนังสือของสมาชิกเข้าสู่ระบบ ระบบพบสมาชิกในระบบ และสมาชิกมีสิทธิในการยืม แต่พบว่าหนังสือเล่มนี้ถูกจอง และสมาชิกได้ อยู่ในคิวการจองอันดับแรก ดังนั้นระบบจึงทำการกำหนดสถานะของหนังสือเล่มนี้เป็นถูกยืม เพิ่มจำนวนหนังสือที่สมาชิกทำการยืม บันทึกรายการยืมลงในระบบ และลบสมาชิกออกจากคิวการจองของหนังสือเล่มนี้



#### 4.1.11 Use Case : คืนหนังสือ (Return Item)

Actor : เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk)

Description : เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกวันที่ทำการคืนหนังสือในรายการยืมหนังสือของสมาชิก

Pre-Condition and System Action :

ตารางที่ 4.8 ตารางแสดง Decision Table สำหรับ Use Case – Return Item

Scenarios	1	2	3	4				
<b>Conditions</b>								
พบหนังสือเล่มนี้ในระบบ	F	T	T	T				
หนังสือเล่มนี้ถูกยืมไป		F	T	T				
หนังสือเล่มนี้เกินกำหนดคืน			F	T				
<b>Actions</b>								
ปฏิเสธการดำเนินการ	X	X						
กำหนดสถานะหนังสือเล่มนี้เป็นว่าง			X	X				
ลดจำนวนการยืมสำหรับสมาชิกรายนี้			X	X				
เพิ่มข้อมูลค่าปรับสำหรับสมาชิกรายนี้				X				

คำอธิบายเหตุการณ์ :

- 1) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการคืนหนังสือ แต่ระบบไม่พบหนังสือเล่มนี้ในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการคืนหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ แต่พบว่าหนังสือเล่มนี้ไม่ได้ถูกยืมจึงทำการปฏิเสธการดำเนินการ
- 3) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการคืนหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ และหนังสือเล่มนี้ได้ถูกยืม และไม่ได้เป็นการคืนที่เกินกำหนด จึงทำการบันทึกข้อมูลสถานะหนังสือเล่มนี้เป็นว่าง และลดจำนวนหนังสือที่สมาชิกทำการยืม
- 4) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการคืนหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ และหนังสือเล่มนี้ได้ถูกยืม แต่พบว่าเป็นการคืนที่เกินกำหนด จึงทำการบันทึกข้อมูลสถานะหนังสือเล่มนี้เป็นว่าง และลดจำนวนหนังสือที่สมาชิกทำการยืมและทำการบันทึกค่าปรับสำหรับสมาชิก

#### 4.1.12 Use Case : จองหนังสือ (Make Reservation)

Actor : เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk)

Description : เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกรายการจองสำหรับหนังสือเล่มที่สมาชิกต้องการ

Pre-Condition and System Action :

ตารางที่ 4.9 ตารางแสดง Decision Table สำหรับ Use Case – Make Reservation

Scenarios	1	2	3	4	5	6
<b>Conditions</b>						
พบหนังสือเล่มนี้ในระบบ	F	T	T	T	T	T
พบสมาชิกในระบบ		F	T	T	T	T
สมาชิกมีสิทธิในการจอง			F	T	T	T
หนังสือเล่มนี้มีรายการจอง				F	T	T
สมาชิกนี้มีชื่ออยู่ในรายการจอง					T	F
<b>Actions</b>						
ปฏิเสธการดำเนินการ	X	X	X		X	
สร้างรายการจองขึ้นมาสำหรับหนังสือเล่มนี้				X		
เพิ่มข้อมูลการจองสำหรับสมาชิกรายนี้ในรายการจองของหนังสือ						X

คำอธิบายเหตุการณ์ :

- 1) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการจองหนังสือ แต่ระบบไม่พบหนังสือเล่มนี้ในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ แต่ไม่พบสมาชิกในระบบจึงทำการปฏิเสธการดำเนินการ
- 3) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ แต่พบว่าสมาชิกไม่มีสิทธิในการจอง(เนื่องจากสมาชิกทำการยืมสารนิเทศไปเป็นจำนวนเท่าที่กำหนด หรือมีหนังสือที่สมาชิกยืมไปเกินใดที่เกินกำหนดส่ง หรือสมาชิกมีค่าปรับที่ค้างชำระกับทางห้องสมุด)ระบบจึงทำการปฏิเสธการดำเนินการ

- 4) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ และสมาชิกมีสิทธิในการจอง และระบบไม่พบรายการจองสำหรับหนังสือเล่มนี้จึงทำการสร้างรายการจองสำหรับหนังสือเล่มนี้ขึ้น แล้วทำการเพิ่มสมาชิกเข้าไปยังรายการจองดังกล่าว
- 5) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ และสมาชิกมีสิทธิในการจอง ระบบพบรายการจองสำหรับหนังสือเล่มนี้ แต่พบว่าไม่มีสมาชิกนี้อยู่ในรายการจองนั้นแล้ว ระบบจึงทำการปฏิเสธการดำเนินการ
- 6) เจ้าหน้าที่รับยืมคืนหนังสือทำการบันทึกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ และสมาชิกมีสิทธิในการจอง ระบบพบรายการจองสำหรับหนังสือเล่มนี้ และไม่พบว่าไม่มีสมาชิกนี้อยู่ในรายการจองนั้น ระบบจึงทำการเพิ่มสมาชิกเข้าไปยังรายการจองดังกล่าว



#### 4.1.13 Use Case : ยกเลิกการจองหนังสือ (Remove Reservation)

Actor : เจ้าหน้าที่รับยืมคืนหนังสือ (Circulation Clerk)

Description : เจ้าหน้าที่รับยืมคืนหนังสือทำการลบรายการจองของสมาชิกออกจากรายการจองของหนังสือเล่มที่สมาชิกต้องการ

Pre-Condition and System Action :

ตารางที่ 4.10 ตารางแสดง Decision Table สำหรับ Use Case – Remove Reservation

Scenarios	1	2	3	4	5	6
<b>Conditions</b>						
พบหนังสือเล่มนี้ในระบบ	F	T	T	T	T	
พบสมาชิกในระบบ		F	T	T	T	
หนังสือเล่มนี้มีรายการจอง			F	T	T	
สมาชิกนี้มีชื่ออยู่ในรายการจอง				F	T	
<b>Actions</b>						
ปฏิเสธการดำเนินการ	X	X	X	X		
ลบสมาชิกออกจากรายการจองของหนังสือเล่มนี้					X	

คำอธิบายเหตุการณ์ :

- 1) เจ้าหน้าที่รับยืมคืนหนังสือทำการยกเลิกการจองหนังสือ แต่ระบบไม่พบหนังสือเล่มนี้ในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) เจ้าหน้าที่รับยืมคืนหนังสือทำการยกเลิกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ แต่ไม่พบสมาชิกในระบบจึงทำการปฏิเสธการดำเนินการ
- 3) เจ้าหน้าที่รับยืมคืนหนังสือทำการยกเลิกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ แต่ระบบไม่พบรายการจองสำหรับหนังสือเล่มนี้จึงทำการปฏิเสธการดำเนินการ
- 4) เจ้าหน้าที่รับยืมคืนหนังสือทำการยกเลิกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ ระบบพบรายการจองสำหรับหนังสือเล่มนี้ แต่ไม่พบสมาชิกในรายการจองดังกล่าวจึงทำการปฏิเสธการดำเนินการ

- 5) เจ้าหน้าที่รับซืมคืนหนังสือทำการยกเลิกการจองหนังสือ ระบบพบหนังสือเล่มนี้ในระบบ ระบบพบรายการจองสำหรับหนังสือเล่มนี้ และพบสมาชิกในรายการจองดังกล่าวจึงทำการลบสมาชิกออกจากรายการจองดังกล่าว

#### 4.1.14 Use Case : ลบสมาชิกห้องสมุด (Remove Member)

Actor : บรรณารักษ์ (Librarian)

Description : บรรณารักษ์ทำการลบสมาชิกออกจากระบบ

Pre-Condition and System Action :

ตารางที่ 4.11 ตารางแสดง Decision Table สำหรับ Use Case – Remove Member

Scenarios	1	2	3	4				
<b>Conditions</b>								
พบสมาชิกในระบบ	F	T	T	T				
สมาชิกมีหนังสือที่ยังไม่ได้คืน		T	F	F				
สมาชิกมีค่าปรับค้างชำระ			T	F				
<b>Actions</b>								
ปฏิเสธการดำเนินการ	X	X	X					
ลบสมาชิกออกจากระบบ				X				

คำอธิบายเหตุการณ์ :

- 1) บรรณารักษ์ทำการลบสมาชิกออกจากระบบ แต่ระบบไม่พบสมาชิกในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) บรรณารักษ์ทำการลบสมาชิกออกจากระบบ ระบบพบสมาชิกในระบบ แต่พบว่าสมาชิกรายนี้มีหนังสือที่ยังค้างส่งจึงทำการปฏิเสธการดำเนินการ
- 3) บรรณารักษ์ทำการลบสมาชิกออกจากระบบ ระบบพบสมาชิกในระบบ สมาชิกไม่มีหนังสือที่ยังค้างส่ง แต่มีค่าปรับค้างชำระกับห้องสมุด ระบบจึงทำการปฏิเสธการดำเนินการ
- 4) บรรณารักษ์ทำการลบสมาชิกออกจากระบบ ระบบพบสมาชิกในระบบ สมาชิกไม่มีหนังสือที่ยังค้างส่ง และไม่มีค่าปรับค้างชำระกับห้องสมุด ระบบจึงทำการลบสมาชิกออกจากระบบ

#### 4.1.15 Use Case : ลบรายการหนังสือ (Remove Title)

Actor : บรรณารักษ์ (Librarian)

Description : บรรณารักษ์ทำการลบรายการหนังสือออกจากระบบ

Pre-Condition and System Action :

ตารางที่ 4.12 ตารางแสดง Decision Table สำหรับ Use Case – Remove Title

Scenarios	1	2	3	4				
<b>Conditions</b>								
พบชื่อหนังสือในระบบ	F	T	T	T				
รายการหนังสือนี้มีหนังสือที่ถูกยืมไป		T	F	F				
รายการหนังสือนี้มีหนังสือที่มีรายการจอง			T	F				
<b>Actions</b>								
ปฏิเสธการดำเนินการ	X	X	X					
ลบหนังสือที่สอดคล้องกับรายการหนังสือนี้ ออกจากระบบ				X				
ลบรายการหนังสือออกจากระบบ				X				

#### คำอธิบายเหตุการณ์ :

- 1) บรรณารักษ์ทำการลบรายการหนังสือออกจากระบบ แต่ระบบไม่พบชื่อหนังสือนี้ในระบบ จึงทำการปฏิเสธการดำเนินการ
- 2) บรรณารักษ์ทำการลบรายการหนังสือออกจากระบบ ระบบพบชื่อหนังสือในระบบ แต่พบว่ารายการหนังสือนี้มีหนังสือที่ถูกยืมไปจึงทำการปฏิเสธการดำเนินการ
- 3) บรรณารักษ์ทำการลบรายการหนังสือออกจากระบบ ระบบพบชื่อหนังสือในระบบ และรายการหนังสือนี้ไม่มีหนังสือที่ถูกยืมไป แต่พบว่ามีรายการจองกับหนังสือที่สอดคล้องกับรายการหนังสือนี้จึงทำการปฏิเสธการดำเนินการ
- 4) บรรณารักษ์ทำการลบรายการหนังสือออกจากระบบ ระบบพบชื่อหนังสือในระบบ รายการหนังสือนี้ไม่มีหนังสือที่ถูกยืมไป และไม่มีรายการจองกับหนังสือที่สอดคล้องกับรายการหนังสือนี้ ระบบจึงทำการลบข้อมูลหนังสือที่สอดคล้องกับรายการหนังสือนี้และข้อมูลรายการหนังสือนี้ออกจากระบบ

#### 4.1.16 Use Case : ลบหนังสือ (Remove Item)

Actor : บรรณารักษ์ (Librarian)

Description : บรรณารักษ์ทำการลบหนังสือออกจากระบบ

Pre-Condition and System Action :

ตารางที่ 4.13 ตารางแสดง Decision Table สำหรับ Use Case – Remove Item

Scenarios	1	2	3	4
<b>Conditions</b>				
พบเลขทะเบียนหนังสือในระบบ	F	T	T	T
เลขทะเบียนหนังสือนี้มีสถานะว่าง		F	T	T
เลขทะเบียนหนังสือนี้มีรายการจอง			T	F
<b>Actions</b>				
ปฏิเสธการดำเนินการ	X	X	X	
ลบเลขทะเบียนหนังสือนี้ออกจากระบบ				X

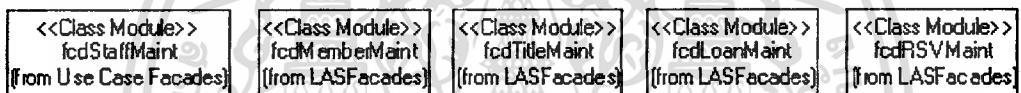
คำอธิบายเหตุการณ์ :

- 1) บรรณารักษ์ทำการลบหนังสือออกจากระบบ แต่ระบบไม่พบหนังสือที่มีเลขทะเบียนนี้ในระบบจึงทำการปฏิเสธการดำเนินการ
- 2) บรรณารักษ์ทำการลบหนังสือออกจากระบบ ระบบพบหนังสือที่มีเลขทะเบียนนี้ในระบบ แต่พบว่าหนังสือที่มีเลขทะเบียนนี้มีสถานะถูกขี้มจึงทำการปฏิเสธการดำเนินการ
- 3) บรรณารักษ์ทำการลบหนังสือออกจากระบบ ระบบพบหนังสือที่มีเลขทะเบียนนี้ในระบบ และพบว่าหนังสือที่มีเลขทะเบียนนี้มีสถานะว่าง แต่พบรายการจองสำหรับหนังสือที่มีเลขทะเบียนนี้ในระบบ จึงทำการปฏิเสธการดำเนินการ
- 4) บรรณารักษ์ทำการลบหนังสือออกจากระบบ ระบบพบหนังสือที่มีเลขทะเบียนนี้ในระบบ หนังสือที่มีเลขทะเบียนนี้มีสถานะว่าง และไม่พบรายการจองสำหรับหนังสือที่มีเลขทะเบียนนี้ในระบบ จึงทำการลบหนังสือที่มีเลขทะเบียนนี้ออกจากระบบ

## 4.2 การออกแบบคลาส (Designing Class)

### 4.2.1 การออกแบบคลาสส่วนหน้า (Designing Façade Class)

สำหรับคลาสส่วนหน้าเราจะพิจารณาจาก use case ที่เราจะดำเนินการพัฒนาในขั้นตอนนี้(ในบางครั้งเราอาจจะพบคลาสส่วนหน้าในชื่ออื่นๆ เช่น use case controller class (Larman. 1998) เนื่องจากโดยทั่วไปแล้วคลาสส่วนหน้ามักจะถูกใช้ในการควบคุมลำดับการทำงานที่ถูกระบุเอาไว้ใน use case หนึ่งๆหรือ กล่าวอีกนัยหนึ่งคือคลาสส่วนหน้าจะทำหน้าที่ในการห่อหุ้มพฤติกรรมของ use case หนึ่งๆเอาไว้) คลาสส่วนหน้าจะทำหน้าที่ป้องกันส่วนติดต่อผู้ใช้ออกจากวัตถุธุรกิจที่เป็นเซิร์ฟเวอร์แอปพลิเคชันซึ่งมีความยุ่งยากซับซ้อนค่อนข้างมาก โดยทั่วไปแล้วมักจะกำหนดให้มีคลาสส่วนหน้าหนึ่งคลาสต่อหนึ่ง use case แต่สำหรับโครงการพัฒนาของเราจะเลือกใช้คลาสในส่วนหน้าหนึ่งคลาสต่อ use case ได้หลาย use case ที่สามารถจัดให้อยู่ในกลุ่มเดียวกันได้ โดยสำหรับคลาสส่วนหน้าที่จะใช้ในระบบได้แสดงไว้ในรูปที่ 4.1



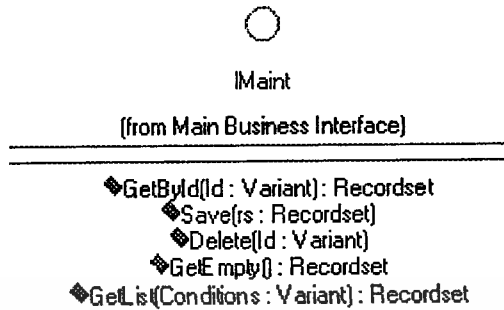
รูปที่ 4.1 Facade Class ในระบบ

### 4.2.2 การออกแบบคลาสธุรกิจหลัก (Designing Main Business Class)

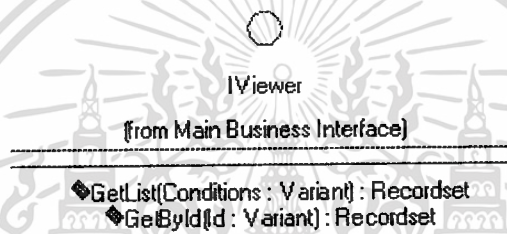
สำหรับคลาสในส่วนนี้จะพิจารณาจากวัตถุจริงๆที่มีความเกี่ยวข้องใน Use Case ที่เรากำลังดำเนินการพัฒนา สิ่งที่สำคัญในการออกแบบที่ต้องระลึกเอาไว้คือ คลาสที่เรากำลังออกแบบนั้นมีหน้าที่ในการให้บริการเท่านั้น ไม่ได้มีหน้าที่ในการเก็บข้อมูลไม่ว่าจะเป็นการเก็บข้อมูลแบบชั่วคราวหรือถาวรก็ตาม โดยส่วนใหญ่แล้วคลาสธุรกิจหลักจะให้บริการเกี่ยวกับการเรียกดู แก้ไข ลบ, และเพิ่มข้อมูล ดังนั้นจึงจะเป็นการดีกว่าที่เราจะรวมบริการในส่วนนี้ให้อยู่ในรูปของส่วนติดต่อ (Interface) เพื่อให้การเข้าถึงบริการในวัตถุธุรกิจหลักสามารถทำได้โดยการเรียกใช้ผ่านส่วนติดต่อนี้ โดยที่การดำเนินการให้บริการนั้นจะแตกต่างกันไปตามคลาสที่สนับสนุนส่วนติดต่อนี้ ซึ่งเป็นหลักการที่เรียกว่า Polymorphism ในแนวความคิดเชิงวัตถุ ดังนั้นเราจึงสร้างส่วนติดต่อเพิ่มขึ้นมาในคลาสธุรกิจหลักนี้ 2 คลาส คือ

1. **Imaint** เป็นส่วนติดต่อที่ให้บริการในการจัดการข้อมูลของวัตถุธุรกิจนั้น และ
  2. **Iviewer** เป็นส่วนติดต่อที่ให้บริการในการเรียกดูข้อมูลของวัตถุธุรกิจนั้นเท่านั้น
- โดย สำหรับสัญลักษณ์ที่ใช้แสดงถึงส่วนติดต่อจะใช้รูปวงกลม ดังที่แสดงไว้ในรูปที่ 4.2 และ 4.3

สำหรับ **Imaint** และ **Iviewer** ตามลำดับ



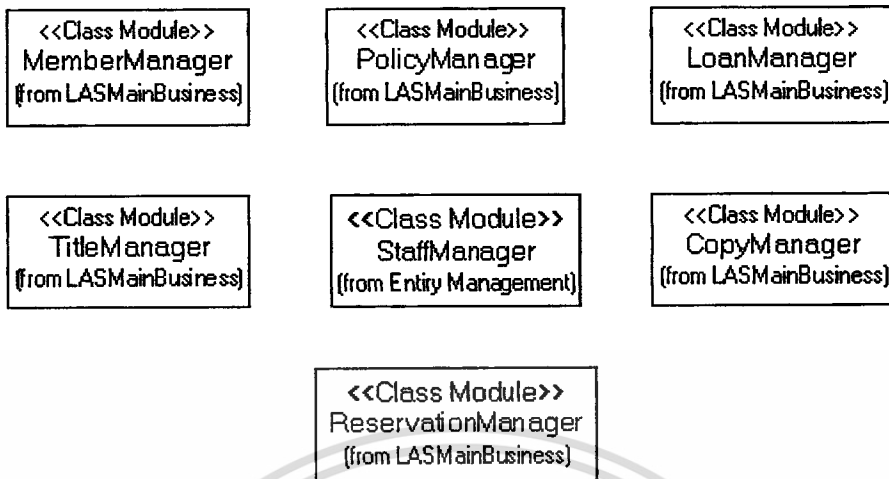
รูปที่ 4.2 แผนภาพส่วนติดต่อ IMaint



รูปที่ 4.3 แผนภาพส่วนติดต่อ IViewer

ในส่วนติดต่อนั้นเราจะทำการกำหนดบริการขึ้นมาแต่ไม่มีการกำหนดรายละเอียดการดำเนินการใดๆเอาไว้ คลาสธุรกิจที่จะใช้ส่วนติดต่อนี้เป็นช่องทางให้คลาสส่วนหน้าเรียกใช้งานมัน โดยคลาสธุรกิจต้องทำการกำหนดรายละเอียดของการบริการในรหัส (Code) โดยที่ชื่อบริการและพารามิเตอร์ต่างๆต้องใช้ตามที่กำหนดเอาไว้ในส่วนติดต่อเท่านั้น การกระทำเช่นนี้ของคลาสธุรกิจที่มีต่อส่วนติดต่อ เราจะเรียกว่า Interface Inheritance

ต่อไปเป็นการพิจารณาคลาสธุรกิจ ซึ่งจะเป็นตัวแทนของวัตถุที่อยู่ในระบบ และให้บริการต่างๆซึ่งทำให้เกิดเป็นบริการของระบบที่มีให้กับสิ่งที่มาขอใช้บริการจากระบบ สำหรับคลาสธุรกิจในระบบ LAS ทั้งหมดได้แสดงไว้ในรูปที่ 4.4



รูปที่ 4.4 Main Business Class ในระบบ

#### 4.2.3 การออกแบบคลาสเข้าถึงข้อมูล (Designing Data Access Class)

สำหรับคลาสส่วนที่ทำหน้าที่เข้าถึงข้อมูลนี้ มีประเด็นที่ต้องคำนึงถึงในการออกแบบคือ ถึงแม้ว่าสภาพแวดล้อมในการให้บริการคอม โพนেন্টของเรา (COM+) จะให้การสนับสนุนทรานเซคชันของคอม โพนেন্ট แต่อย่างไรก็ตามถึงเหล่านี้ก็ทำให้เกิด Overhead ในการสื่อสารระหว่างชั้นได้มาก ดังนั้นเราควรจะใช้ทรานเซคชันเท่าที่จำเป็นในการดำเนินงานกับฐานข้อมูล เพราะฉะนั้นเราจึงต้องทำการแบ่งประเภทคลาสเข้าถึงข้อมูลออกเป็น 2 ประเภทคือ ประเภทที่ไม่ต้องการทรานเซคชัน (เช่นกิจกรรมที่เป็นการเรียกดูข้อมูลเท่านั้น ซึ่งคลาสประเภทนี้จะมีค่าลงท้ายด้วย Fetcher) และประเภทที่ต้องการทรานเซคชัน (เช่นกิจกรรมที่มีการเปลี่ยนแปลงแก้ไขข้อมูลเกิดขึ้น ซึ่งคลาสประเภทนี้จะมีค่าลงท้ายด้วย Modifier) โดยคลาสเข้าถึงข้อมูลนี้จะจับคู่ 1 ต่อ 1 กับคลาสธุรกิจหลัก โดยอาจจะมีคลาสเพิ่มเติมที่ไม่ได้จับคู่กับคลาสธุรกิจหลักแต่มีเอาไว้เพื่อรองรับการควบคุมทรานเซคชันที่ซับซ้อนซึ่งจะได้กล่าวถึงในภายหลัง ดังนั้นในการพัฒนาชั้นตอนนี้จะได้คลาสเข้าถึงข้อมูลประเภทที่รองรับทรานเซคชันและไม่รองรับทรานเซคชัน ดังแสดงในรูปที่ 4.5 และ 4.6 ตามลำดับ

<<Class Module>> AuthorFetcher (from LASDataAccess)	<<Class Module>> BibliographyFetcher (from LASDataAccess)	<<Class Module>> CategoryFetcher (from LASDataAccess)	<<Class Module>> CopyFetcher (from LASDataAccess)
<<Class Module>> FacultyFetcher (from LASDataAccess)	<<Class Module>> InitialFetcher (from LASDataAccess)	<<Class Module>> LoanFetcher (from LASDataAccess)	<<Class Module>> MemberFetcher (from LASDataAccess)
<<Class Module>> MemberTypeFetcher (from LASDataAccess)	<<Class Module>> PolicyFetcher (from LASDataAccess)	<<Class Module>> PressFetcher (from LASDataAccess)	<<Class Module>> ReservationFetcher (from LASDataAccess)
<<Class Module>> StaffFetcher (from LASDataAccess)	<<Class Module>> StaffLogFetcher (from LASDataAccess)	<<Class Module>> SubjectFetcher (from LASDataAccess)	<<Class Module>> SubjectListFetcher (from LASDataAccess)

รูปที่ 4.5 Data Access Class ที่ไม่รองรับทรานแซกชันในระบบ

<<Class Module>> AuthorModifier (from LASDataAccess)	<<Class Module>> BibliographyModifier (from LASDataAccess)	<<Class Module>> CategoryModifier (from LASDataAccess)	<<Class Module>> CopyModifier (from LASDataAccess)
<<Class Module>> FacultyModifier (from LASDataAccess)	<<Class Module>> InitialModifier (from LASDataAccess)	<<Class Module>> LoanModifier (from LASDataAccess)	<<Class Module>> MemberModifier (from LASDataAccess)
<<Class Module>> MemberTypeModifier (from LASDataAccess)	<<Class Module>> PolicyModifier (from LASDataAccess)	<<Class Module>> PressModifier (from LASDataAccess)	<<Class Module>> ReservationModifier (from LASDataAccess)
<<Class Module>> StaffModifier (from LASDataAccess)	<<Class Module>> StaffLogModifier (from LASDataAccess)	<<Class Module>> SubjectModifier (from LASDataAccess)	<<Class Module>> SubjectListModifier (from LASDataAccess)
<<Class Module>> BookTrSvo (from LASDataAccess)	<<Class Module>> PolicyTrSvo (from LASDataAccess)	<<Class Module>> MemberTrSvo (from LASDataAccess)	<<Class Module>> LoanTrSvo (from LASDataAccess)
<<Class Module>> RSVTrSvo (from LASDataAccess)			

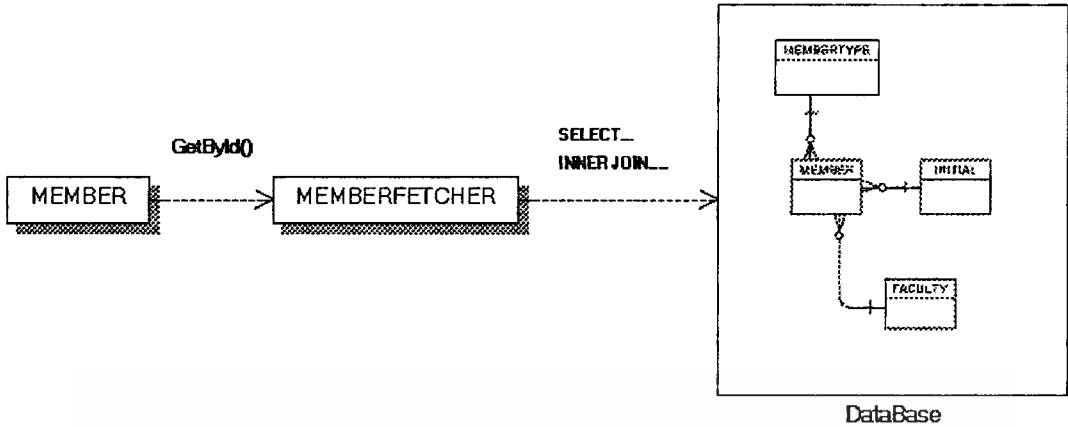
รูปที่ 4.6 Data Access Class ที่รองรับทรานแซกชันในระบบ

ประเด็นที่สำคัญอีกประการหนึ่งในการออกแบบคลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลคือการแปลงมุมมองที่แตกต่างกันระหว่างคลาสที่เรียกใช้งานมันและบริการข้อมูลหรือฐานข้อมูลให้ดูเสมือนว่าการทำงานในด้านของชั้นให้บริการทางธุรกิจและในฝั่งของการบริการข้อมูลนั้นไม่มีความแตกต่างกัน ดังที่กล่าวมาแล้วว่าการออกแบบชั้นให้บริการทางธุรกิจของเรานั้นจะไม่ทำการรักษาสถานะของวัตถุใดๆเอาไว้ในระหว่างการทำงาน ซึ่งจะแตกต่างจากระบบที่ถูกพัฒนาด้วยแนวคิดแบบเชิงวัตถุซึ่งจะมีการรักษาสถานะของวัตถุเอาไว้ตลอดเวลากว่าวัตถุจะถูกทำลายจึงจะทำการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บสถานะของวัตถุนั้นลงในฐานข้อมูลทำให้การดำเนินการต่างๆกับสถานะของวัตถุนั้นจะถูกทำโดยวัตถุโดยตรงในหน่วยความจำของคอมพิวเตอร์ ส่วนฐานข้อมูลเป็นเพียงที่เก็บรักษาสถานะของวัตถุเพียงอย่างเดียวเท่านั้น แต่การออกแบบในลักษณะที่ไม่ให้วัตถุในระบบรักษาสถานะใดๆเอาไว้ นั้นจะทำให้เราสามารถใช้งานสิ่งที่มีอยู่แล้วในฐานข้อมูลให้เกิดประโยชน์มากที่สุดไม่ว่าจะเป็นเรื่องของการกำหนดทางด้านทรานเซคชัน การล็อคเรคอร์ด และการควบคุมในด้านอื่นๆ ถึงแม้ว่ามันจะมีมุมมองที่แตกต่างจากมุมมองในเชิงวัตถุที่เราใช้ในการออกแบบในชั้นบริการทางธุรกิจก็ตาม

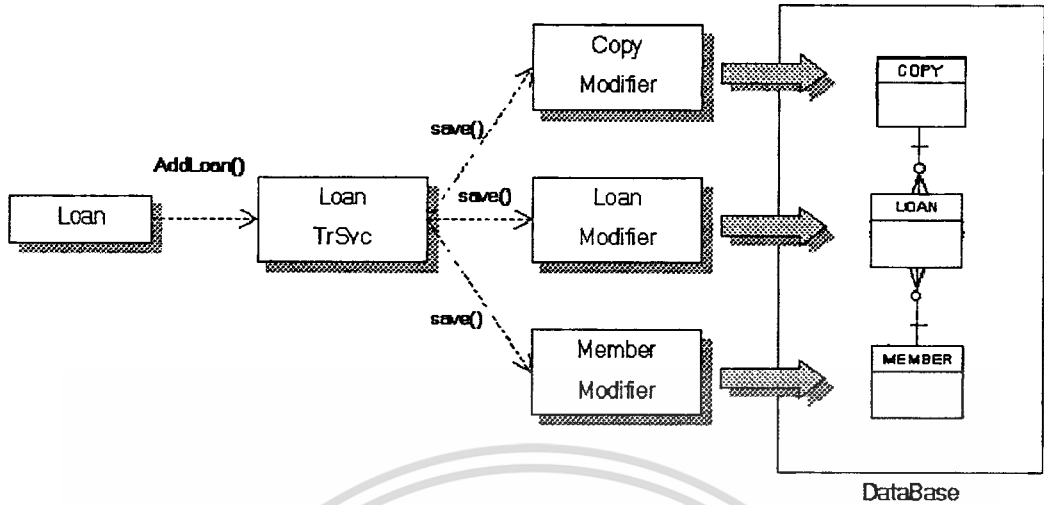
ดังนั้นในกรณีนี้คลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลจึงมีบทบาทที่สำคัญในการให้บริการต่อคลาสธุรกิจหลักซึ่งมีมุมมองในรูปแบบเชิงวัตถุ ให้สามารถเข้าถึงข้อมูลในชั้นบริการข้อมูลหรือฐานข้อมูลซึ่งข้อมูลนั้นไม่ได้เก็บข้อมูลในรูปแบบของวัตถุทางธุรกิจโดยตรง เนื่องจากเป็นฐานข้อมูลเชิงสัมพันธ์ ซึ่งรูปแบบของข้อมูลที่จะถูกจัดเก็บในฐานข้อมูลเชิงสัมพันธ์นี้จะต้องถูกลดความซ้ำซ้อนลง ทำให้ข้อมูลของวัตถุหนึ่งๆนั้นอาจจะต้องถูกแตกออกเป็นหลายๆตารางในฐานข้อมูลเชิงสัมพันธ์

การให้บริการของคลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลที่มีต่อคลาสธุรกิจหลักจะแบ่งออกเป็น 2 ส่วนหลักๆ คือ การค้นคืนสถานะของวัตถุ(การดึงข้อมูล) และการบันทึกสถานะของวัตถุ(การเพิ่ม ลบ เปลี่ยนแปลงข้อมูล) โดยในส่วนแรกนั้นคลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลสามารถทำได้โดยการใช้ความสามารถของ SQL Statement เช่นคำสั่ง JOIN ในการเชื่อมข้อมูลของวัตถุที่กระจายออกไปอยู่ในหลายๆตารางเพื่อรวมเป็นกลุ่มของข้อมูลที่เป็นสถานะของวัตถุหนึ่งๆได้ ยกตัวอย่างเช่น คลาส Member อาจเรียกใช้งานคลาส MemberFetcher ในการดึงข้อมูลสถานะของมันจากฐานข้อมูล ซึ่งในฐานข้อมูลนั้นข้อมูลสถานะของ Member อาจกระจายอยู่ในตารางสมาชิก ตารางคณะ ตารางประเภทสมาชิก และตารางค่านำหน้าชื่อ ดังนั้นหน้าที่ของคลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลคือการรวบรวมข้อมูลของวัตถุที่กระจายอยู่ในหลายๆตารางนี้เข้าด้วยกันเป็นข้อมูลสถานะของสมาชิกก่อนที่จะส่งไปยังคลาสธุรกิจหลัก ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 การดึงข้อมูลสถานะของวัตถุจากฐานข้อมูล

สำหรับในส่วนของการบันทึกสถานะของวัตถุซึ่งจะมีการเปลี่ยนแปลงข้อมูลในฐานข้อมูลนั้น เนื่องจากหลักการในการออกแบบคลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลนั้นจะออกแบบให้มี 1 คลาสต่อตาราง 1 ตารางในฐานข้อมูลเพื่อให้แต่ละคลาสนั้นรับรู้ในการที่จะดำเนินการใด ๆ กับตารางในฐานข้อมูลที่สอดคล้องกับมันเท่านั้น แต่ในการบันทึกสถานะของวัตถุนั้นอาจจะมีผลกระทบต่อข้อมูลในหลายตารางเนื่องจากสาเหตุดังที่กล่าวมาแล้วข้างต้นนั้นทำให้ในกรณีนี้จึงต้องออกแบบให้มีคลาสที่ทำหน้าที่ควบคุมการดำเนินการของคลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลหลายๆคลาสที่จะทำการเปลี่ยนแปลงข้อมูลในหลายๆตารางเพื่อให้การดำเนินการนั้นมีลักษณะที่เป็นทรานแซกชันคือถ้าสำเร็จต้องสำเร็จทั้งหมด ถ้ามีคลาสใดคลาสหนึ่งดำเนินการไม่สำเร็จทุกๆคลาสที่เกี่ยวข้องจะต้องยกเลิกการดำเนินการทั้งหมด เพื่อรักษาความถูกต้องของข้อมูลทั้งก่อนและหลังดำเนินการ ซึ่งเป็นหลักการโดยทั่วไปของทรานแซกชันนั่นเอง สำหรับตัวอย่างในกรณีนี้ จะเป็นตัวอย่างในการบันทึกรายการยืมหนังสือซึ่งจะต้องเกี่ยวข้องกับตารางในฐานข้อมูลทั้งสิ้น 3 ตารางคือ ตารางการยืม ตารางเลขทะเบียนหนังสือ ตารางสมาชิก ดังนั้นจึงต้องออกแบบให้มีคลาสที่ทำหน้าที่ในการควบคุมการดำเนินการของคลาสที่ทำหน้าที่ในการเข้าถึงข้อมูลที่จะดำเนินการใด ๆ กับตารางในฐานข้อมูลทั้ง 3 ตารางดังแสดงในรูปที่ 4.8



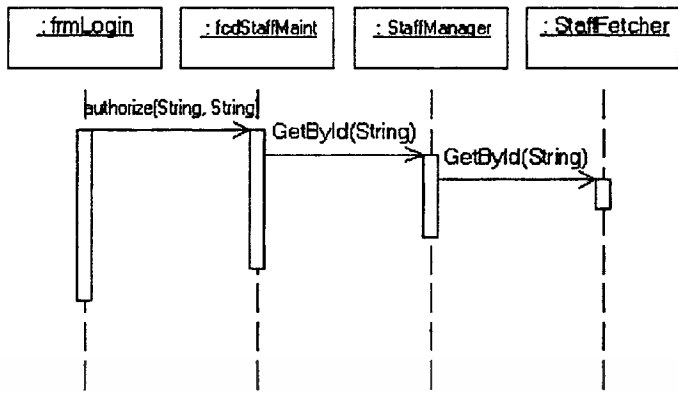
รูปที่ 4.8 การบันทึกข้อมูลสถานะของวัตถุจากฐานข้อมูล

#### 4.3 การออกแบบส่วนติดต่อผู้ใช้ (Designing Presentation Layer)

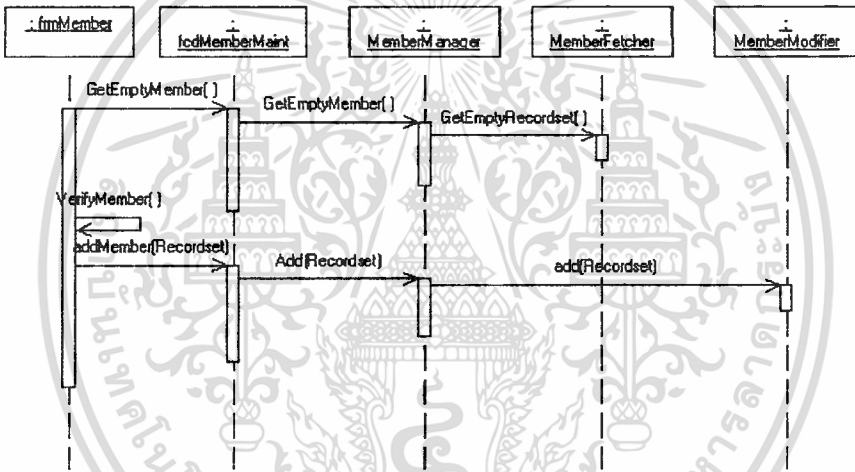
สำหรับส่วนติดต่อผู้ใช้ในขั้นตอนการพัฒนาจะทำการพัฒนาส่วนติดต่อผู้ใช้ให้อยู่ในลักษณะที่เป็นแอปพลิเคชัน Windows มาตรฐาน (windows 32 bit) ซึ่งจะต้องทำการติดตั้งแอปพลิเคชันก่อนจึงจะสามารถใช้งานระบบได้ และมีงานบางส่วนที่เปิดให้บริการในรูปแบบสาธารณะ ผู้ใช้งานระบบสามารถเข้าถึงระบบผ่านทาง Web Browser ได้

#### 4.4 Design Sequence Diagram

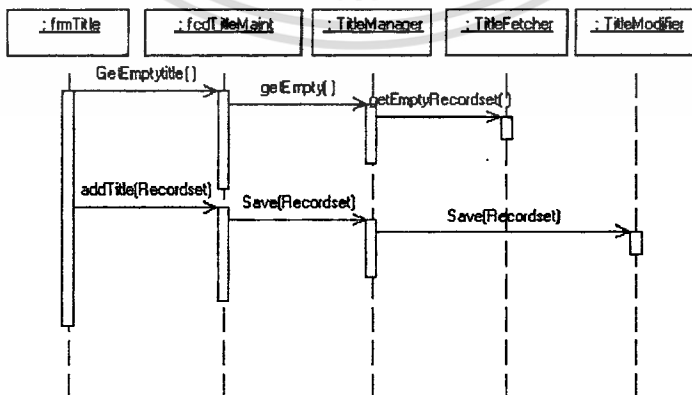
เมื่อได้ทำการพัฒนาส่วนของคลาสที่เกี่ยวข้องในสถาปัตยกรรมของระบบทั้งหมดแล้ว เราจะสามารถแสดงพฤติกรรมของระบบผ่านทาง Interaction Diagram ของ UML ได้โดยที่ข้อความที่ถูกส่งระหว่างวัตถุต่างๆก็คือ operation หรือ method ของวัตถุนั้นเอง interaction diagram นี้จะเป็นแบบจำลองของระบบที่ทำให้เราได้ทราบว่าการทำงานที่ระบบจะสามารถดำเนินการ use case ต่างๆ โดยการทำงานร่วมกันระหว่างวัตถุต่างๆในระบบได้อย่างไร โดยจะแสดงในรูปที่ 4.9 ถึง 4.23



รูปที่ 4.9 Sequence Diagram ของ Use Case Log In

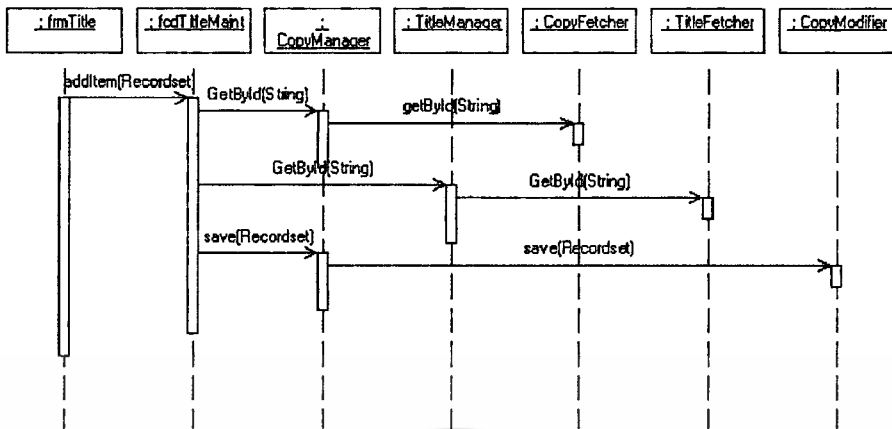


รูปที่ 4.10 Sequence Diagram ของ Use Case Add Member

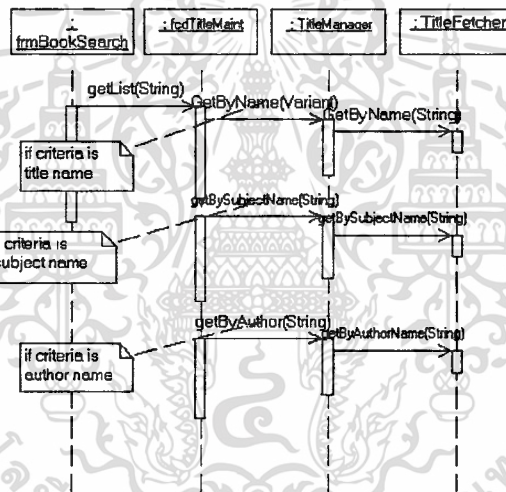


รูปที่ 4.11 Sequence Diagram ของ Use Case Add Title

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

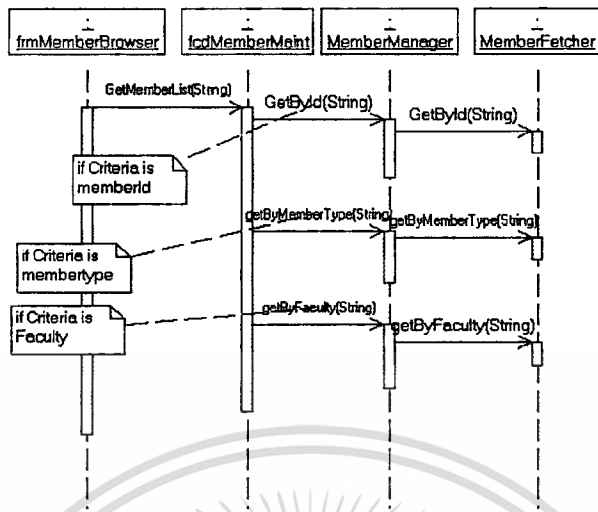


รูปที่ 4.12 Sequence Diagram ของ Use Case Add Item

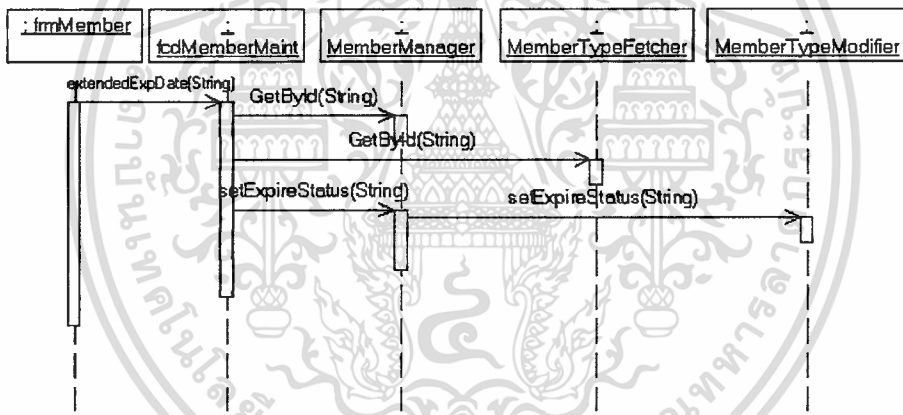


รูปที่ 4.13 Sequence Diagram ของ Use Case Browse Catalog

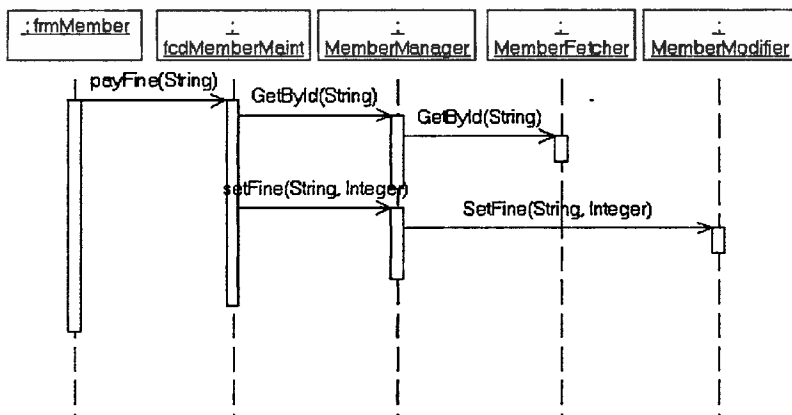
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 Sequence Diagram ของ Use Case Browse Member

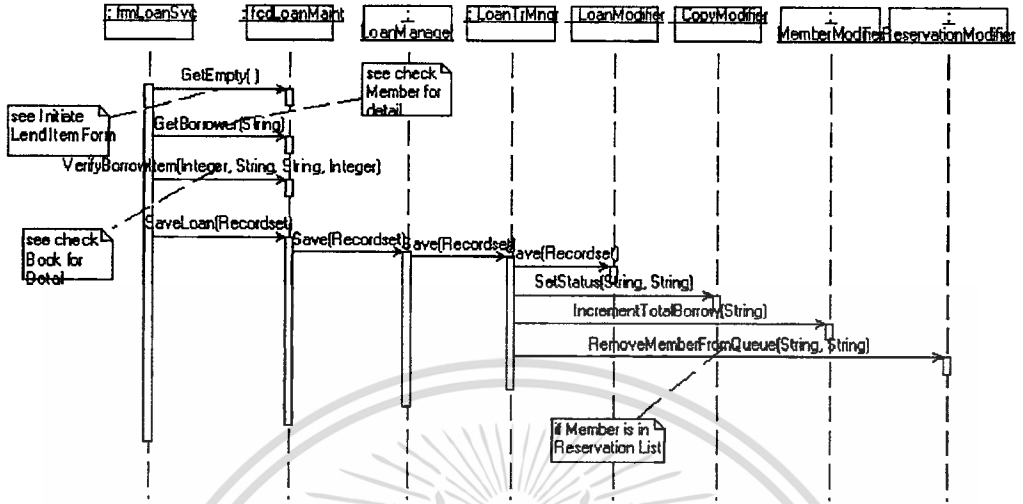


รูปที่ 4.15 Sequence Diagram ของ Use Case Extended Expire Date

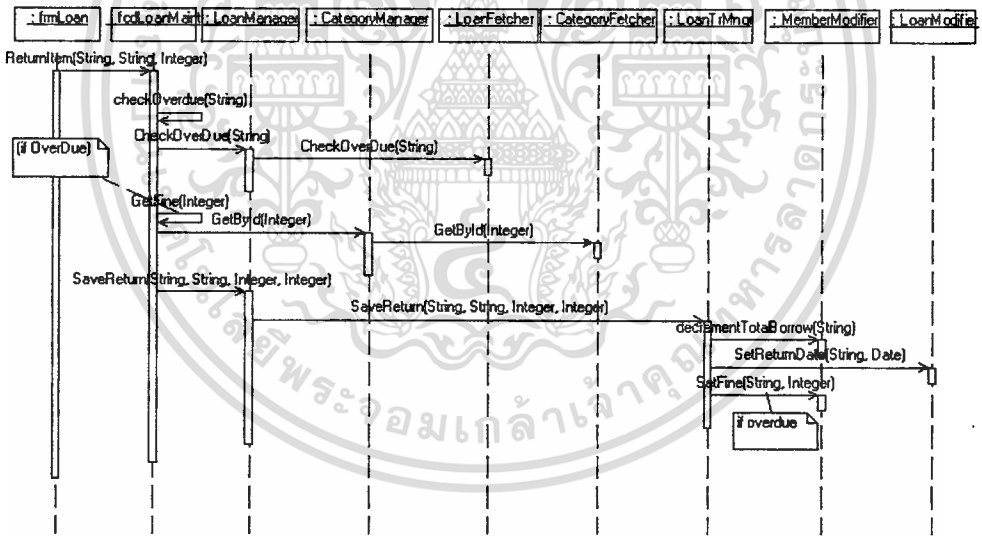


รูปที่ 4.16 Sequence Diagram ของ Use Case Pay Fine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

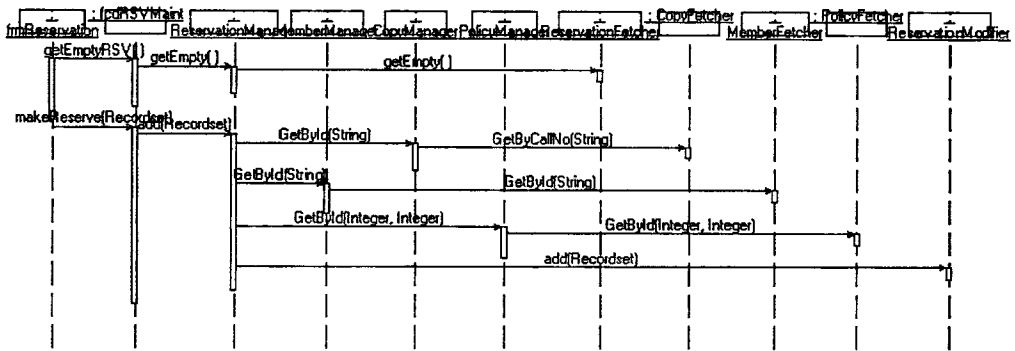


รูปที่ 4.17 Sequence Diagram ของ Use Case Lend Item

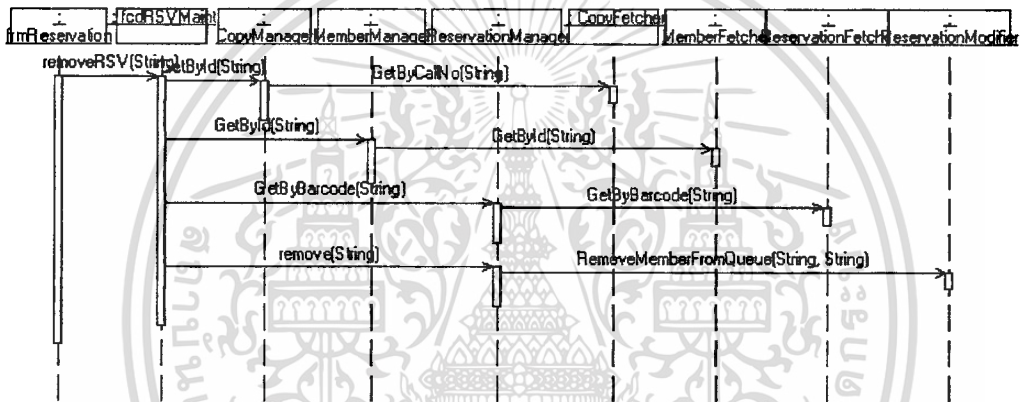


รูปที่ 4.18 Sequence Diagram ของ Use Case Return Item

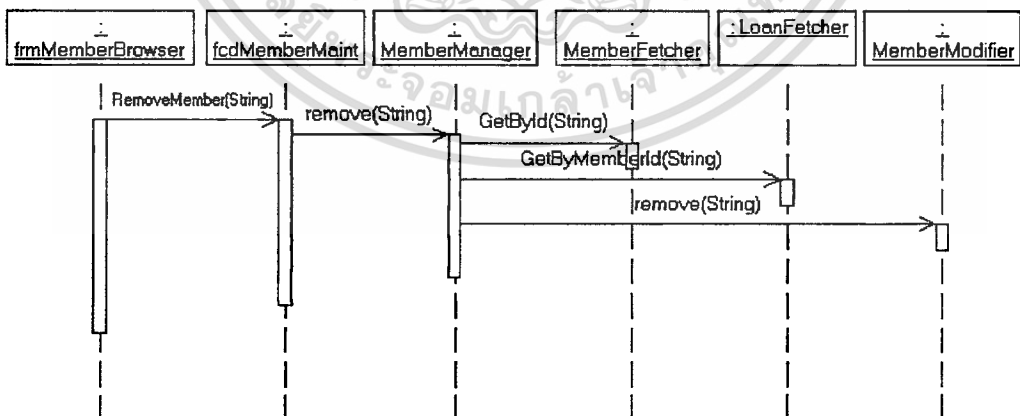
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 Sequence Diagram ของ Use Case Make Reservation

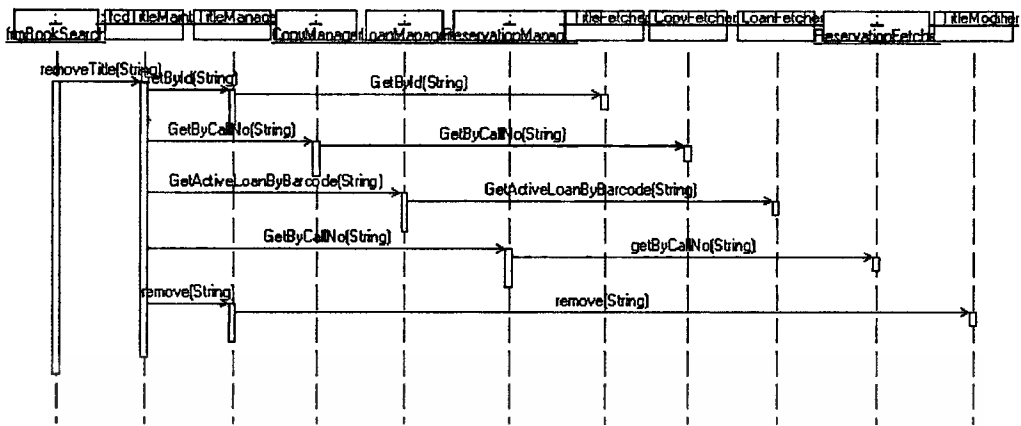


รูปที่ 4.20 Sequence Diagram ของ Use Case Remove Reservation

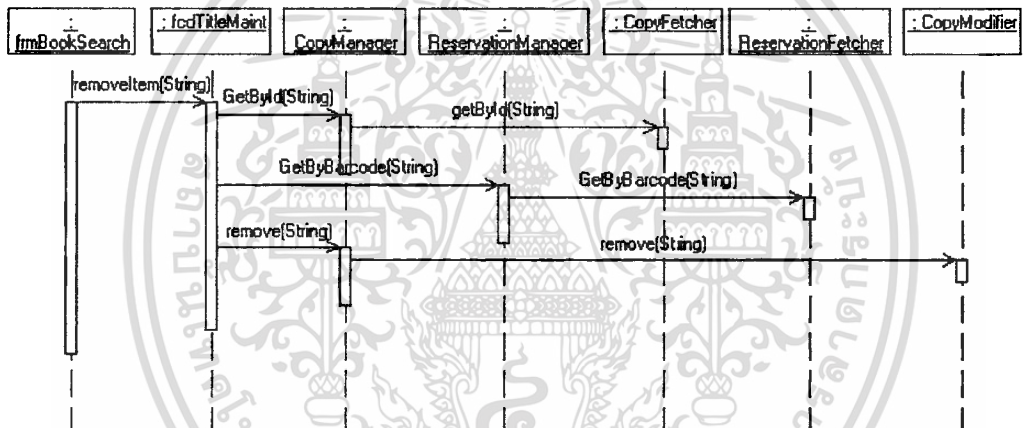


รูปที่ 4.21 Sequence Diagram ของ Use Case Remove Member

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



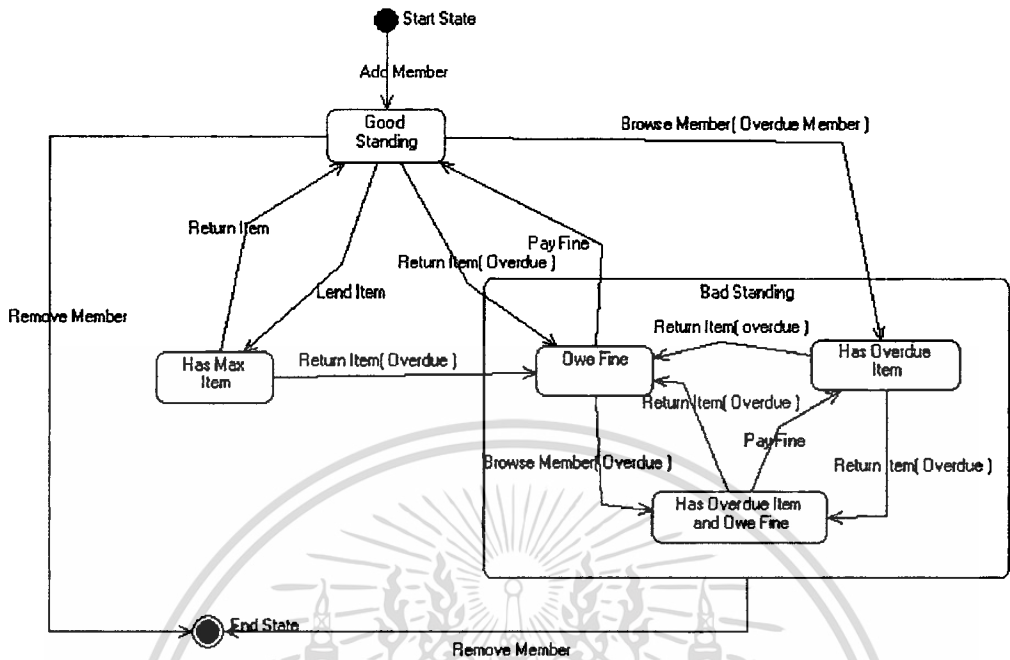
รูปที่ 4.22 Sequence Diagram ของ Use Case Remove Title



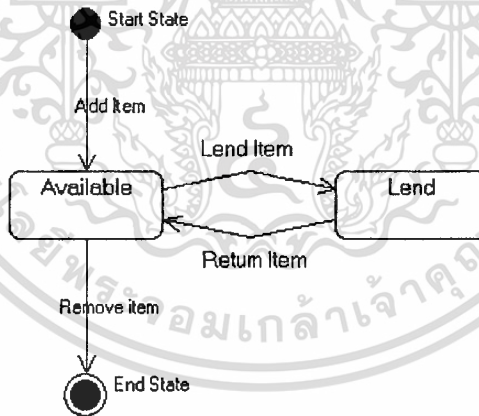
รูปที่ 4.23 Sequence Diagram ของ Use Case Remove Item

#### 4.5 State Diagram

เมื่อทำการพัฒนาแบบจำลอง Interaction Diagram ซึ่งใช้อธิบายพฤติกรรมของวัตถุหลายตัวที่มีการสื่อสารกันใน use case หนึ่งๆ แล้วเราจะสามารถใช้แผนภาพ State ในการอธิบายพฤติกรรมของวัตถุหนึ่งในหลายๆ use case ซึ่งวัตถุนั้นมีส่วนร่วมในกิจกรรมของ use case นั้นได้ สำหรับแผนภาพ State ของระบบที่ทำการพัฒนานี้ได้แสดงในรูปที่ 4.24 ถึง 4.27

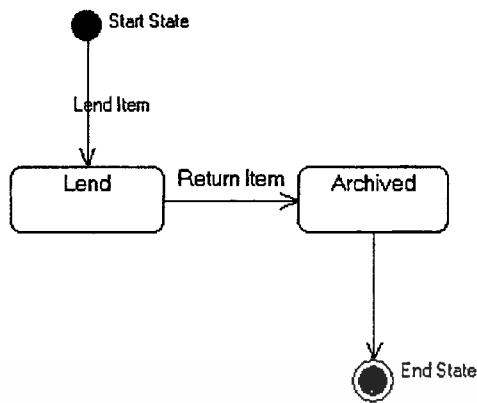


รูปที่ 4.24 State Diagram ของวัดตุตตะมาจิก

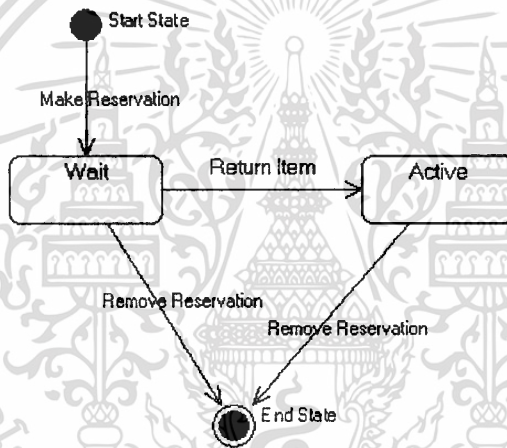


รูปที่ 4.25 State Diagram ของวัดฤๅษะเขียนหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 State Diagram ของวัตถุการยืม



รูปที่ 4.27 State Diagram ของวัตถุการจอง

#### 4.6 Defining the Physical Architecture

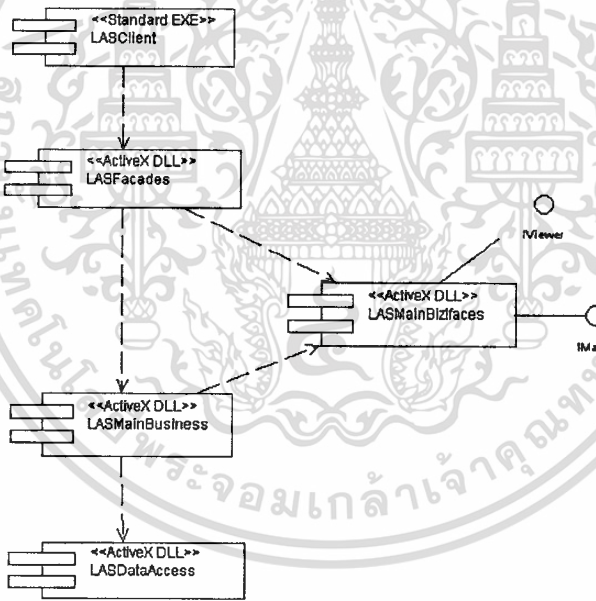
เมื่อเราทำการสร้างแบบจำลองที่เป็นพื้นฐานที่สำคัญต่อระบบแล้ว ขั้นตอนต่อไปเราจะทำการร่างโครงสร้างทางสถาปัตยกรรมพื้นฐานของระบบขึ้นมา โดยจะมีขั้นตอนย่อยๆดังนี้

1. ทำการรวบรวมคลาสที่ออกแบบเพื่อนำไปไว้ในคอมโพเนนท์
2. ทำการรวบรวมคอมโพเนนท์ไปไว้ใน Package หรือในอีกชื่อหนึ่งคือ COM+ Application
3. นำ Package หรือ COM+ Application ที่ได้ไปใช้งานในเครื่องคอมพิวเตอร์ต่างๆที่ได้จัดรูปแบบในการ Deployment เอาไว้

ในขั้นตอนนี้เราจะให้ความสนใจเฉพาะในขั้นตอนที่ 1 เท่านั้น ส่วนในขั้นตอนที่ 2 และ 3 นั้นจะกล่าวถึงอีกครั้งในบทที่ 6 การสร้างระบบ โดยรายละเอียดของการรวบรวมคลาสที่ออกแบบเพื่อนำไปไว้ในคอมโพเนนท์มีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำคลาสที่ออกแบบ (COM+ Class) ไปไว้ในคอมโพเนนท์นั้นหากเราทำจากเครื่องมือที่ใช้ในการพัฒนา(ในที่นี้คือ Visual Basic) เมื่อเราทำการสร้าง Project ประเภท ActiveX DLL นั้นจะเป็นการสร้างคอมโพเนนท์ขึ้นมาโดยที่คลาสที่อยู่ใน Project นั้นจะเป็นคลาสที่อยู่ในคอมโพเนนท์นั้นไปโดยปริยายเมื่อเราทำการคอมไพล์ Project แต่ในที่นี้เราทำการสร้างแบบจำลองขึ้นมาโดยใช้ CASE Tool ดังนั้นเราจะต้องทำการสร้างคอมโพเนนท์ขึ้นมาใน CASE Tool ตามรูปแบบของแผนภาพคอมโพเนนท์ของ UML โดยหลักในการพิจารณาว่าจะทำการสร้างคอมโพเนนท์ขึ้นมาเป็นจำนวนเท่าใดนั้นจะพิจารณาจากคลาสที่ออกแบบได้เป็นหลักโดยอาจจะพิจารณาจากการทำงานร่วมกันของคลาส, คลาสที่มีการทำงานร่วมกันอาจจะนำไปไว้ในคอมโพเนนท์เดียวกัน แต่ในที่นี้เราจะสร้างคอมโพเนนท์ขึ้นตามประเภทของคลาสที่ได้ทำการออกแบบไว้ดังนั้นทำให้เราสามารถสร้างคอมโพเนนท์ขึ้นได้ 5 คอมโพเนนท์ ดังรูปที่ 4.28



รูปที่ 4.28 Component Diagram ของระบบ

เมื่อเราทำการสร้างคอมโพเนนท์ขึ้นมาแล้วเราจะต้องทำการ Assign คลาสที่ออกแบบไว้ไปยังคอมโพเนนท์ต่างๆ ในที่นี้เราจะทำการ Assign แต่ละคลาสไปยังคอมโพเนนท์ที่มีสอดคล้องกับประเภทของมัน ใน CASE Tool (ในที่นี้ใช้ Rational Rose 2000) การ Assign คลาสไปยังคอมโพเนนท์สามารถทำได้ไม่ยากโดยการ Double-Click ที่ตัวคอมโพเนนท์ จากนั้นจะมีวินโดว์ให้ทำการกำหนดรายละเอียดของคอมโพเนนท์ซึ่งเราสามารถเลือกคลาสที่จะทำการ Assign ให้กับคอมโพเนนท์นั้นได้ ดังแสดงในรูปที่ 4.29



## บทที่ 5

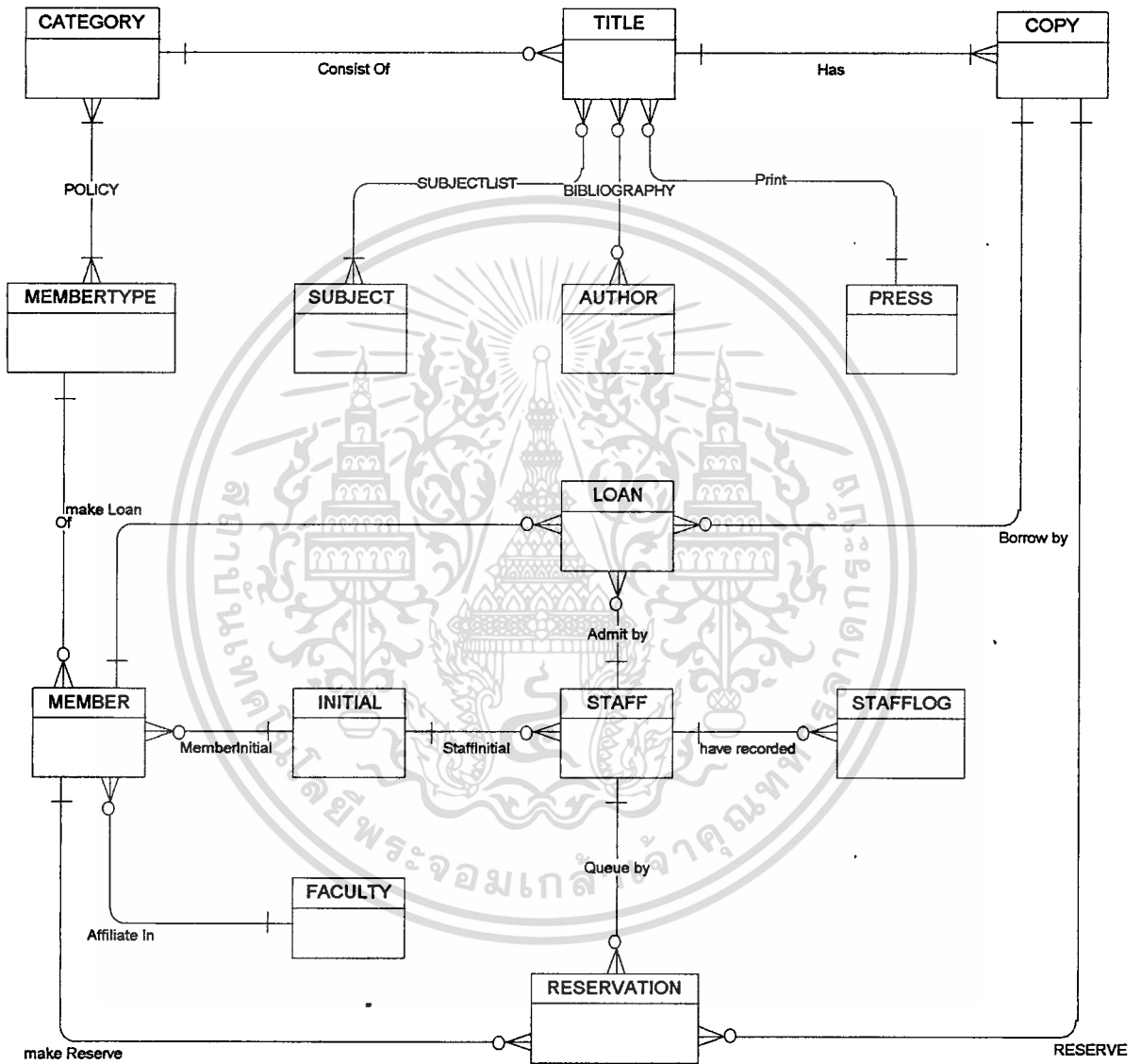
### การออกแบบฐานข้อมูล

ถึงแม้ว่าการออกแบบระบบบริหารห้องสมุดจะใช้แนวคิดเชิงวัตถุส่วนหนึ่งในการออกแบบระบบซึ่งโดยส่วนใหญ่แล้วจะการออกแบบโดยใช้แนวคิดเชิงวัตถุจะใช้แบบจำลองวัตถุ (object model) เป็นตัวแทนของระบบทั้งในแง่ของโครงสร้างซึ่งรวมถึงข้อมูลต่างๆและพฤติกรรมของระบบ แต่โดยแนวความคิดของการออกแบบในรูปแบบของคอมพิวเตอร์เบสที่ใช้กันจะมองแบบจำลองวัตถุเป็นเพียงแบบจำลองทางแนวความคิด(abstraction) ที่มีหน้าที่ในการนำบริการไปสู่ผู้ใช้งานมัน ไม่ได้มีหน้าที่ในการเก็บรักษาข้อมูล ไม่ว่าจะเป็นการเก็บข้อมูลแบบชั่วคราวหรือถาวร (Sundblad, S. 2000) ดังนั้นหน้าที่ในส่วนนี้ควรเป็นหน้าที่โดยตรงของฐานข้อมูล ซึ่งในที่นี้เราจะใช้ฐานข้อมูลเชิงสัมพันธ์ในการเก็บข้อมูลของวัตถุในระบบ ดังนั้นการออกแบบส่วนนี้จึงใช้หลักการของแบบจำลองข้อมูลที่ได้จากการศึกษาระบบทั้งหมด แล้วนำมาพิจารณาว่าควรมีข้อมูลใดเกี่ยวข้องกับระบบ และควรมีการจัดเก็บลงฐานข้อมูลอย่างไร โดยเริ่มจากการออกแบบจำลองความสัมพันธ์ระหว่างเอนติตี กำหนดความซ้ำซ้อนของข้อมูล กำหนดโครงสร้างของตารางให้เป็นไปตามข้อกำหนดของระบบจัดการฐานข้อมูลที่ต้องการใช้ แล้วจึงจัดการแม็ป (Map) แบบจำลองที่ได้สร้างขึ้นนี้เป็นโครงสร้างฐานข้อมูล(Physical Database) ลงบนฐานข้อมูล

#### 5.1 แบบจำลองแสดงความสัมพันธ์ระหว่างเอนติตี (Entity Relationship Model)

เป็นการสร้างแบบจำลองแสดงความสัมพันธ์ระหว่างข้อมูล โดยนำเอนติตีของระบบทั้งหมดมาหาความสัมพันธ์ระหว่างกันว่ามีความเกี่ยวข้องกันอย่างไร โดยที่ยังไม่ได้ผ่านขั้นตอนการกำจัดความซ้ำซ้อนของข้อมูล (Normalization) แล้วจึงกำหนดรายละเอียดและคุณสมบัติของแต่ละเอนติตีว่าข้อมูลภายในอะไรบ้าง และข้อมูลใดที่ใช้เป็นคีย์ของแต่ละเอนติตี

สำหรับ ER Model ของระบบบริหารงานห้องสมุด ได้แสดงให้เห็นดังภาพที่ 5.1 ซึ่งเป็นภาพ Context Data Model แสดงให้เห็นถึงความสัมพันธ์ของแต่ละเอนติตีและภาพที่ 5.2 เป็นภาพ Key-based Data Model แสดงให้เห็นถึงความสัมพันธ์ระหว่างเอนติตีและ Primary key ของแต่ละเอนติตี



ภาพที่ 5.1 Context Data Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## 5.2 การสร้างฐานข้อมูล

หลังจากที่ได้ออกแบบแบบจำลองแสดงความสัมพันธ์ระหว่างเอนติตีแล้วจะต้องมีการจัดเตรียมโครงสร้างข้อมูลให้เหมาะสมกับระบบจัดการฐานข้อมูลที่ต้องการสร้างฐานข้อมูลเพื่อจัดเก็บข้อมูล โดยนำแบบจำลองที่ได้ออกแบบไว้มาลดความซ้ำซ้อนของข้อมูลและกำหนดชื่อแอตทริบิวต์ชนิดของข้อมูลรวมทั้งข้อกำหนดต่างๆให้เป็นไปตามลักษณะของระบบจัดการฐานข้อมูลดังแสดงในตารางที่ 5.1 ถึงตารางที่ 5.17

ตารางที่ 5.1 รายละเอียดตารางผู้แต่ง

Table name: AUTHOR (ผู้แต่ง)				
No.	Attribute name	Description	Type	Key
1	AUTHOR_ID	รหัสผู้แต่ง	INTEGER NOT NULL	PK
2	AUTHOR_FIRSTNAME	ชื่อต้นผู้แต่ง	CHAR(7) NOT NULL	
3	AUTHOR_MIDDLENAME	ชื่อกลางผู้แต่ง	CHAR(13) NULL	
4	AUTHOR_LASTNAME	นามสกุลผู้แต่ง	CHAR(7) NOT NULL	

ตารางที่ 5.2 รายละเอียดตารางรายชื่อผู้แต่ง

Table name: BIBLIOGRAPHY				
No.	Attribute name	Description	Type	Key
1	CALL_NO	เลขเรียกหนังสือ	CHAR(25) NOT NULL	PK, FK
2	AUTHOR_ID	รหัสผู้แต่ง	INT	PK, FK

ตารางที่ 5.3 รายละเอียดตารางประเภทหนังสือ

Table name: CATEGORY				
No.	Attribute name	Description	Type	Key
1	CATEGORY_ID	รหัสประเภทหนังสือ	SMALLINT NOT NULL	PK
2	CATEGORY_DESC	รายละเอียดประเภทหนังสือ	CHAR(50) NOT NULL	
3	CATEGORY_FINERATE	อัตราค่าปรับ	SMALLINT NOT NULL	

ตารางที่ 5.4 รายละเอียดตารางทะเบียนหนังสือ

Table name: COPY				
No.	Attribute name	Description	Type	Key
1	BARCODE	รหัสบาร์โค้ด	CHAR(10) NOT NULL	PK
2	CALL_NO	เลขเรียกหนังสือ	CHAR(25) NOT NULL	FK
3	COMPANIONMEDIA	แฟล็กบอกว่ามีสื่อ อื่นๆแนบมาด้วยหรือไม่	CHAR(1) NULL	
4	REGISTER_DATE	วันที่นำเข้าห้องสมุด	DATETIME NULL	
5	STATUS	สถานะ	CHAR(1) NULL	

ตารางที่ 5.5 รายละเอียดตารางคณะ

Table name: FACULTY				
No.	Attribute name	Description	Type	Key
1	FAC_ID	รหัสคณะ	SMALLINT NOT NULL	PK
2	FAC_NAME	ชื่อคณะ	CHAR(50) NOT NULL	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.6 รายละเอียดตารางค่านำหน้าชื่อ

Table name: INITIAL				
No.	Attribute name	Description	Type	Key
1	INITIAL_ID	รหัสค่านำหน้าชื่อ	SMALLINT NOT NULL	PK
2	INITIAL_NAME	รายละเอียดค่านำหน้าชื่อ	CHAR(10) NOT NULL	

ตารางที่ 5.7 รายละเอียดตารางรายการยืมหนังสือ

Table name: LOAN				
No.	Attribute name	Description	Type	Key
1	LOAN_ID	หมายเลขการยืม	INTEGER NOT NULL	PK
2	BARCODE	รหัสบาร์โค้ด	CHAR(10) NOT NULL	FK
3	STAFF_ID	รหัสประจำตัวเจ้าหน้าที่ที่ให้ยืม	CHAR(10) NOT NULL	FK
4	MEMBER_ID	รหัสประจำตัวสมาชิก	CHAR(10) NOT NULL	FK
5	LOAN_DATE	วันที่ยืม	DATETIME NOT NULL	
6	LOAN_DUEDATE	กำหนดคืน	DATETIME NOT NULL	
7	LOAN_RETURNDATE	วันที่ยังนำมาคืน	DATETIME NULL	

ตารางที่ 5.8 รายละเอียดตารางสมาชิก

Table name: MEMBER				
No.	Attribute name	Description	Type	Key
1	MEMBER_ID	เลขประจำตัวสมาชิก	CHAR(10) NOT NULL	PK
2	MEMBERTYPE_ID	รหัสประเภทสมาชิก	SMALLINT NOT NULL	FK
3	FAC_ID	รหัสคณะ	SMALLINT NOT NULL	FK
4	INITIAL_ID	รหัสค่านำหน้าชื่อ	SMALLINT NOT NULL	FK
5	MEMBER_FIRSTNAME	ชื่อสมาชิก	CHAR(40) NOT NULL	

6	MEMBER_LASTNAME	นามสกุลสมาชิก	CHAR(50) NOT NULL	
7	MEMBER_EMAIL	อีเมลสมาชิก	CHAR(30) NULL	
8	MEMBER_EXPIREDATE	วันหมดอายุของสมาชิก	DATETIME NOT NULL	
9	MEMBER_FINEOWE	ค่าปรับค้างชำระ	SMALLINT NULL	
10	MEMBER_TOTALBORROW	จำนวนหนังสือที่มีในครอบครองทั้งหมด	SMALLINT NULL	

ตารางที่ 5.9 รายละเอียดตารางประเภทสมาชิก

Table name: MEMBERTYPE				
No.	Attribute name	Description	Type	Key
1	MEMBERTYPE_ID	รหัสประเภทสมาชิก	SMALLINT NOT NULL	PK
2	MEMBERTYPE_NAME	ชื่อประเภทสมาชิก	CHAR(50) NOT NULL	

ตารางที่ 5.10 รายละเอียดตารางข้อกำหนดการยืม

Table name: POLICY				
No.	Attribute name	Description	Type	Key
1	CATEGORY_ID	รหัสประเภทหนังสือ	SMALLINT NOT NULL	PK, FK
2	MEMBERTYPE_ID	รหัสประเภทสมาชิก	SMALLINT NOT NULL	PK, FK
3	DURATIONLIMIT	ระยะเวลาที่ยืมได้ต่อหนังสือ 1 เล่ม	SMALLINT NOT NULL	
4	TOTALCOPYLIMIT	จำนวนหนังสือที่ยืมได้ทั้งหมด	SMALLINT NOT NULL	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.11 รายละเอียดตารางสำนักพิมพ์

Table name: PRESS				
No.	Attribute name	Description	Type	Key
1	PUBLISHER_ID	รหัสสำนักพิมพ์	INT	PK
2	PUBLISHER_NAME	ชื่อสำนักพิมพ์	CHAR(50) NOT NULL	

ตารางที่ 5.12 รายละเอียดตารางการจองหนังสือ

Table name: RESERVATION				
No.	Attribute name	Description	Type	Key
1	RSV_ID	หมายเลขการจอง	INTEGER NOT NULL	PK
2	MEMBER_ID	เลขประจำตัวสมาชิก	CHAR(10) NOT NULL	FK
3	STAFF_ID	เลขประจำตัวเจ้าหน้าที่ ที่ทำการจอง	CHAR(10) NOT NULL	FK
4	BARCODE	รหัสบาร์โค้ด	CHAR(10) NOT NULL	FK
5	RSV_DATE	วันที่ทำการจอง	DATETIME NOT NULL	
6	RSV_RECIEVEDATE	วันกำหนดรับหนังสือ	DATETIME NULL	

ตารางที่ 5.13 รายละเอียดตารางเจ้าหน้าที่

Table name: STAFF				
No.	Attribute name	Description	Type	Key
1	STAFF_ID	เลขประจำตัวเจ้าหน้าที่	CHAR(10) NOT NULL	PK
2	INITIAL_ID	รหัสคำนำหน้าชื่อ	SMALLINT NOT NULL	FK
3	STAFF_FIRSTNAME	ชื่อเจ้าหน้าที่	CHAR(40) NOT NULL	
4	STAFF_LASTNAME	นามสกุลเจ้าหน้าที่	CHAR(50) NOT NULL	
5	STAFF_POSITION	ตำแหน่งเจ้าหน้าที่	CHAR(40) NOT NULL	
6	STAFF_PASSWORD	รหัสผ่านเข้าสู่ระบบ	CHAR(10) NOT NULL	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.14 รายละเอียดตารางข้อมูลการทำงานของเจ้าหน้าที่

Table name: STAFFLOG				
No.	Attribute name	Description	Type	Key
1	LOG_ID	หมายเลขบันทึก	INTEGER NOT NULL	PK
2	STAFF_ID	เลขประจำตัวเจ้าหน้าที่	CHAR(10) NOT NULL	FK
3	LOG_DATE	วันที่เข้าใช้ระบบ	DATETIME NOT NULL	
4	LOGIN_TIME	เวลาที่เข้าใช้ระบบ	DATETIME NOT NULL	
5	LOGOUT_TIME	เวลาที่ออกจากระบบ	DATETIME NOT NULL	

ตารางที่ 5.15 รายละเอียดตารางหัวเรื่อง

Table name: SUBJECT				
No.	Attribute name	Description	Type	Key
1	SUBJECT_ID	รหัสหัวเรื่อง	INTEGER NOT NULL	PK
2	SUBJECT_NAME	ชื่อหัวเรื่อง	CHAR(50) NOT NULL	

ตารางที่ 5.16 รายละเอียดตารางรายการหัวเรื่องของหนังสือ

Table name: SUBJECTLIST				
No.	Attribute name	Description	Type	Key
1	CALL_NO	เลขเรียกหนังสือ	CHAR(25) NOT NULL	PK, FK
2	SUBJECT_ID	รหัสหัวเรื่อง	INTEGER NOT NULL	PK, FK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.17 รายละเอียดตารางหนังสือ

Table name: TITLE				
No.	Attribute name	Description	Type	Key
1	CALL_NO	เลขเรียกหนังสือ	CHAR(25) NOT NULL	PK
2	CATEGORY_ID	รหัสประเภทหนังสือ	SMALLINT NOT NULL	FK
3	PUBLISHER_ID	รหัสสำนักพิมพ์	INTEGER NOT NULL	FK
4	ISBN	รหัส ISBN	CHAR(10) NULL	
5	TITLE_NAME	ชื่อหนังสือ	VARCHAR(100) NOT NULL	
6	EDITION	ครั้งที่พิมพ์	SMALLINT NULL	
7	IMPRINT	พิมพ์ลักษณะ	VARCHAR(100) NULL	
8	DESCRIPT	รายละเอียดหนังสือ	VARCHAR(100) NULL	
9	NOTE	หมายเหตุอื่นๆ	VARCHAR(200) NULL	

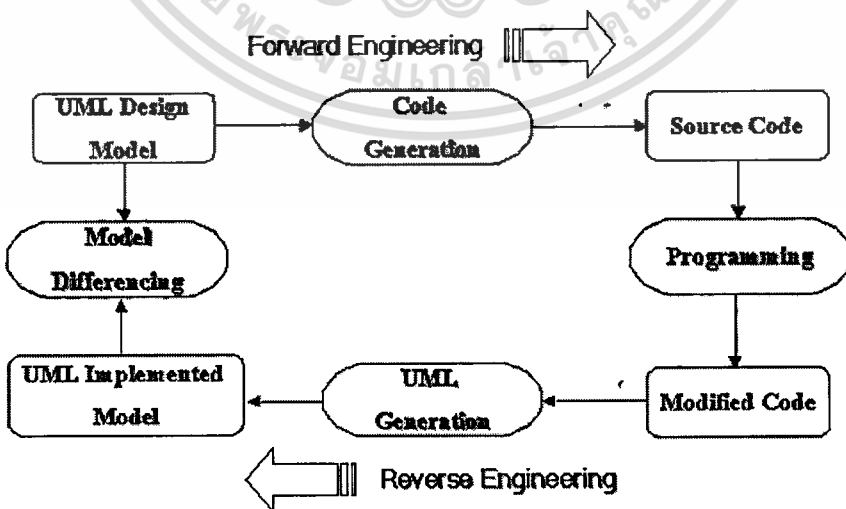
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### การสร้างระบบ

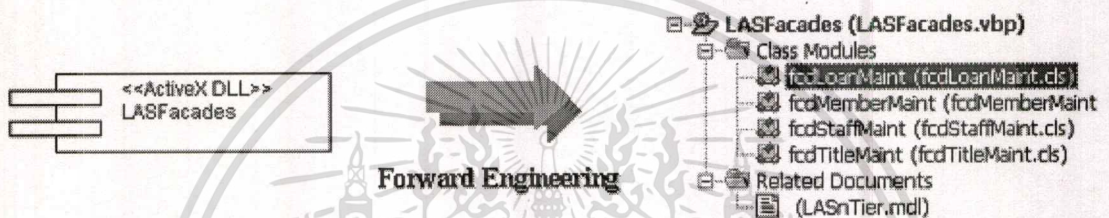
#### 6.1 Round-Trip Engineering

ภายหลังจากที่ทำการสร้างแบบจำลองของระบบที่จะดำเนินการพัฒนาเรียบร้อยแล้วขั้นตอนต่อไปจะเป็นการเริ่มเขียนโปรแกรม โดยเราจะอาศัยแบบจำลองที่เราได้ทำการพัฒนาขึ้นมาทำการสร้างโครงของโปรแกรม (Code Skeleton) โดยใช้ Case Tool ในการสร้างโครงของโปรแกรม ซึ่งเรียกว่าเป็นการทำ Forward Engineering หลังจากนั้นจะทำการเขียนโปรแกรมลงไปในโครงของโปรแกรมเหล่านั้นเพื่อให้แอปพลิเคชันสามารถทำงานได้ตามที่ได้ออกแบบเอาไว้ แล้วจึงทำการแปลงโค้ดที่เขียนเสร็จแล้วกลับไปเป็นแบบจำลองอีกครั้งซึ่งขบวนการนี้เรียกว่า Reverse Engineering โดยที่ CASE Tool จะทำการสร้างแบบจำลองที่เพิ่มเติมขึ้นมาจากการที่เราจะทำการเพิ่มเติมโค้ดบางส่วนที่ไม่ได้มีอยู่ในแบบจำลองที่ได้สร้างไว้ในตอนแรก โดยในแต่ละรอบของการพัฒนานั้นขอบเขตของการพัฒนานั้นจะถูกกำหนดโดย use case ที่เราเลือกมาทำการพัฒนาดังนั้นการพัฒนาโครงการทั้งหมดจะอยู่ในรูปแบบที่เรียกว่าการพัฒนาในลักษณะการทำซ้ำและเพิ่มทีละขั้น (Iteration and Incremental) ขบวนการทั้งหมดนี้สามารถแสดงในรูปของแผนภาพในรูปที่ 6.1



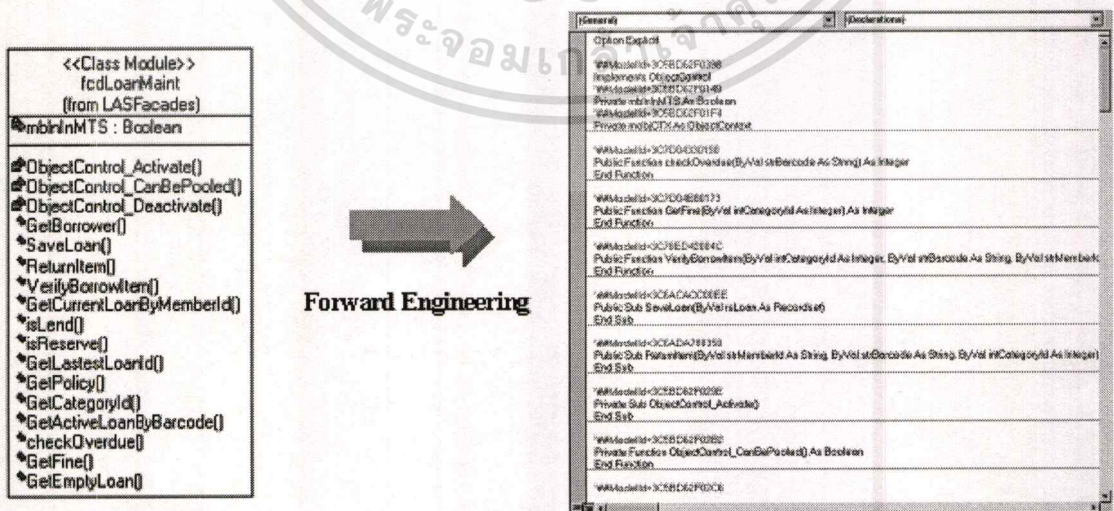
รูปที่ 6.1 ขบวนการ Round-Trip Engineering

สำหรับในกรอบงานของ Rational Rose Visual Basic แล้ว คลาสที่เราออกแบบไว้จะถูกนำไปกำหนดให้กับคอมโพเนนต์ดังรายละเอียดที่กล่าวไว้ในหัวข้อที่ 4.5) Defining the Physical Architecture ซึ่งคอมโพเนนต์เหล่านี้เมื่อนำไปทำ Forward Engineering ก็จะกลายเป็น Project หนึ่งของ Visual Basic ที่อยู่ในรูปของ ActiveX DLL Project และคลาสทั้งหมดที่อยู่ในคอมโพเนนต์นั้นก็จะกลายเป็น Class Module ใน ActiveX DLL Project นั้น ดังแสดงในรูปที่ 6.2 ซึ่งเป็นตัวอย่างที่เกิดจากการทำ Forward Engineering คอมโพเนนต์ LASFacades ไปเป็น LASFacades ActiveX DLL Project ใน Visual Basic



รูปที่ 6.2 การทำ Forward Engineering ของ LASFacades Component

เมื่อพิจารณาในแต่ละ Class Module ใน ActiveX DLL Project ที่ถูกสร้างขึ้นมาจาก CASE Tool นั้น จะพบว่ามีโครงของโปรแกรมที่ถูกสร้างขึ้นมาจาก Attributes และ Methods ของคลาสในคอมโพเนนต์ดังแสดงในรูปที่ 6.3 ซึ่งแสดงให้เห็นถึงโครงของโค้ดที่ถูกสร้างขึ้นมาจากคลาส fcdLoanMaint ในคอมโพเนนต์ LASFacades



รูปที่ 6.3 โครงของโค้ดที่ถูกสร้างขึ้นจากคลาส fcdLoanMaint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ได้ทำการเติมโครงของโค้ดเหล่านี้ด้วยการเขียน โปรแกรมและทำการทดสอบแอปพลิเคชัน จนกระทั่งสามารถทำงานในส่วนที่ออกแบบเอาไว้ได้ เราก็จะใช้การ Reverse Engineering เพื่อทำการสร้างแบบจำลองส่วนที่อาจจะเพิ่มขึ้นมาใน โปรแกรมที่ได้รับการเพิ่มเติมนี้ ซึ่งก็คือว่าจบ 1 รอบในการพัฒนาด้วย Round-Trip Engineering ในรอบต่อไปของการพัฒนาเราก็จะเลือก use case ที่จะทำให้การพัฒนาต่อไปมาดำเนินการในลักษณะเดียวกับที่กล่าวมา

## 6.2 ตัวอย่าง Source Code

ในขั้นตอนการทำการเขียน โปรแกรมนี้จะมีข้อสังเกตในการเรียกใช้งานคลาสที่อยู่ในคอม โปเนนท์ที่ต่างกันซึ่งจะต้องทำการประกาศเป็นตัวแปร โดยใช้การอ้างอิงในลักษณะดังรูปที่ 6.4

### การประกาศตัวแปร

```
Dim [ชื่อตัวแปร] As [ชื่อคอม โปเนนท์].[ชื่อคลาส โมดูลในคอม โปเนนท์]
(กรณี Client มีรูปแบบเป็น Win32, ในที่นี้คือ Visual Basic)
Set [ชื่อตัวแปร] = CreateObject("[ชื่อคอม โปเนนท์].[ชื่อคลาส โมดูลในคอม โปเนนท์]")
(กรณี Client มีรูปแบบเป็น Server-Side Script, ในที่นี้คือ ASP)
Set [ชื่อตัวแปร] = Server.CreateObject("[ชื่อคอม โปเนนท์].[ชื่อคลาส โมดูลในคอม โปเนนท์]")
```

### ตัวอย่างเช่น

```
Dim objMgr As LASMainBusiness.LoanManager
[Win32] Set objMgr = CreateObject("LASMainBusiness.LoanManager")
[ASP] Set objMgr = Server.CreateObject("LASMainBusiness.LoanManager")
```

### การเรียกใช้งาน

```
[ชื่อตัวแปร]เรียกใช้ที่สร้างไว้. [ชื่อ Method]
```

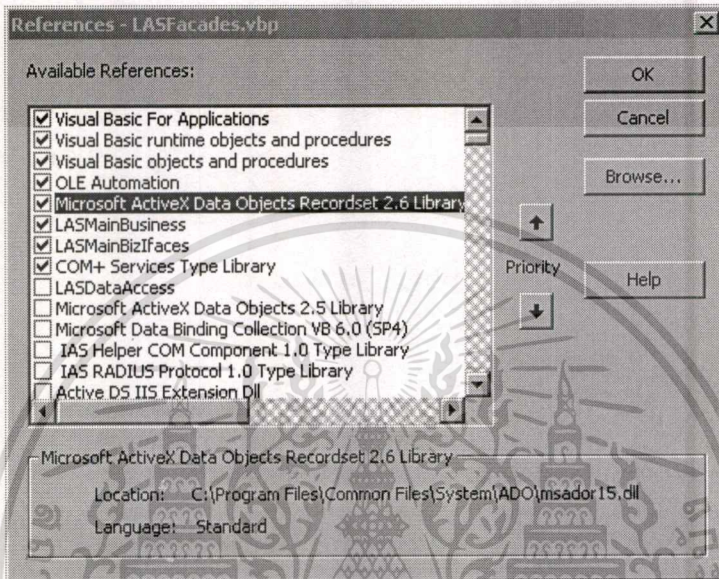
### ตัวอย่างเช่น

```
objMgr.checkOverdue(strBarcode)
```

รูปที่ 6.4 ตัวอย่าง โปรแกรมในส่วนของการสร้าง Object และการเรียกใช้งาน Method

นอกจากนี้การที่เราจะสามารถอ้างอิงถึงคอม โปเนนท์อื่นจาก Project ที่กำลังทำงานอยู่ได้นั้นเราจะต้องทำการกำหนดการอ้างอิงไปยังคอม โปเนนท์นั้นก่อน โดยการกำหนดที่ Project Reference ดังตัวอย่างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างในรูปที่ 6.5 ซึ่งเป็นการอ้างอิงไปยังคอมโพเนนต์ LASMainBusiness และ LASMainBizIFaces ของคอมโพเนนต์ LASFacades ทำให้คลาสโมดูลใดๆในคอมโพเนนต์ LASFacades นั้นสามารถเรียกใช้งานคลาสโมดูลใดๆที่อยู่ในคอมโพเนนต์ LASMainBusiness และ LASMainBizIFaces ได้



รูปที่ 6.5 การกำหนดการอ้างอิงของคอมโพเนนต์

เนื่องจากในที่นี่เรามีการใช้งาน Component Service (สำหรับใน Windows NT จะเรียกว่า MTS) ซึ่งจะต้องมีการเพิ่มเติมส่วนของโปรแกรมขึ้นมาบางส่วนและต้องมีการกำหนดการอ้างอิงไปยัง COM+ Services Type Library ด้วยดังที่แสดงในรูปที่ 6.5 ส่วนรายละเอียดของโปรแกรมในส่วนที่เพิ่มขึ้นมาจากการใช้งานคอมโพเนนต์ที่ไม่ได้ใช้งาน Component Service เพื่อรองรับการทำงานในสภาพแวดล้อมของ Component Service นั้นจะได้ทำการอธิบายเพิ่มเติมดังโศกในรูปที่ 6.6

```

Option Explicit
Implements ObjectControl
Private mblnInMTS As Boolean
Private mobjCTX As ObjectContext

Private Sub ObjectControl_Activate()
    Set mobjCTX = GetObjectContext()
    mblnInMTS = Not (mobjCTX Is Nothing)
End Sub

Private Function ObjectControl_CanBePooled() As Boolean
    ObjectControl_CanBePooled = False
End Function

Private Sub ObjectControl_Deactivate()
End Sub

```

### รูปที่ 6.6 ตัวอย่างโปรแกรมที่รองรับการทำงานในสภาพแวดล้อมของ Component Service

จากตัวอย่างโปรแกรมในรูปที่ 6.6 จะพบว่ามีการใช้งาน Object (ObjectContext) และ Interface (ObjectControl) ซึ่งจะสามารถสรุปถึงความจำเป็นในการเรียกใช้งานทั้งสองส่วนได้ดังนี้

#### 1) ObjectControl Interface

Interface นี้จะประกาศ Method ที่สำคัญ 3 Method ในการที่จะรองรับการทำงานของคอมพิวเตอร์ในสภาพแวดล้อมของ Component Service นั่นคือ

**Method Activate** เป็น method ที่ Component Service จะเรียกใช้เมื่อมันทำการสร้างอ็อบเจกต์ขึ้นมา ซึ่งในที่นี้เราใช้งานมันในการกำหนดตัวแปรที่ใช้ตรวจสอบว่าอ็อบเจกต์กำลังทำงานอยู่ภายใต้สภาพแวดล้อม Component Service หรือไม่

**Method Deactivate** เป็น method ที่ Component Service จะเรียกใช้เมื่อมันทำการทำลายอ็อบเจกต์ปกติกจะใช้งาน Method นี้ร่วมกับ Method CanBePooled เพื่อทำ Object Pooling แต่เนื่องจากคุณสมบัติอันนี้ไม่สามารถใช้งานกับคอมพิวเตอร์ที่สร้างขึ้นโดย Visual Basic ได้ดังนั้นจึงไม่ทำการ Implement Method นี้

**Method CanBePooled** เป็น method ที่ Component Service จะเรียกใช้หลังจากที่มันเรียกใช้ Method Deactivate เล็กน้อยเพื่อทำ Object Pooling แต่คุณสมบัติอันนี้ไม่สามารถใช้งานกับคอมโพเนนต์ที่สร้างขึ้นโดย Visual Basic ได้ ดังนั้นจึงทำการกำหนดให้มันเป็น False

## 2) Object ObjectContext

อ็อบเจกต์นี้จะถูกสร้างขึ้นมาเพื่อติดตามการทำงานของอ็อบเจกต์ทุกตัวที่ทำงานภายใต้สภาพแวดล้อมของ Component Service โดยอ็อบเจกต์ที่ทำงานภายใต้การควบคุมของ Component Service จะทำงานในบริบทเป็นของตัวเองหรือใช้บริบทร่วมกับอ็อบเจกต์อื่น ขึ้นอยู่กับการกำหนดคุณสมบัติด้านทรานแซกชันของอ็อบเจกต์นั้น สำหรับ method ที่สำคัญของอ็อบเจกต์นี้ ประกอบด้วย

**Method CreateInstance** เป็น method ที่อ็อบเจกต์หนึ่งจะเรียกใช้เพื่อทำการสร้างอ็อบเจกต์อีกตัวหนึ่งขึ้นมาภายใต้สภาพแวดล้อม Component Service

**Method SetComplete** เป็น method ที่อ็อบเจกต์จะใช้ในการแจ้งให้ Component Service ทราบว่าสิ่งที่มันดำเนินการนั้นสำเร็จ(ไม่มีข้อผิดพลาดเกิดขึ้น)เพื่อให้ Component Service สามารถนำทรัพยากรส่วนที่อ็อบเจกต์นั้นใช้กลับคืนไปจัดการต่อได้

**Method SetAbort** เป็น method ที่อ็อบเจกต์จะใช้ในการแจ้งให้ Component Service ทราบว่าสิ่งที่มันดำเนินการนั้นล้มเหลว(มีข้อผิดพลาดเกิดขึ้น)เพื่อให้ Component Service ยกเลิกกิจกรรมใดๆที่ถูกดำเนินการโดยอ็อบเจกต์ที่ทำงานภายใต้บริบทเดียวกันกับอ็อบเจกต์นั้นเพื่อเป็นการรักษาสภาพความถูกต้องทั้งก่อนและหลังดำเนินการตามหลักการของทรานแซกชัน

โดยสำหรับตัวอย่างการใช้งาน method ที่กล่าวมาทั้งหมดนี้ได้แสดงเอาไว้ในรูปที่ 6.7

```

Public Function GetActiveLoanByBarcode(ByVal strBarcode As String) As Recordset
Dim objMngr As LASMainBusiness.LoanManager
On Error GoTo ErrHandler

If mblnInMTS Then
    Set objMngr = mobjCTX.CreateInstance("LASMainBusiness.LoanManager")
Else
    Set objMngr = CreateObject("LASMainBusiness.LoanManager")
End If
Set GetActiveLoanByBarcode = objMngr.GetActiveLoanByBarcode(strBarcode)
If mblnInMTS Then mobjCTX.SetComplete
Exit Function

ErrHandler:
If mblnInMTS Then mobjCTX.SetAbort
Err.Raise Number:=Err.Number, Description:=Err.Description, _
    Source:=Err.Source
End Function

```

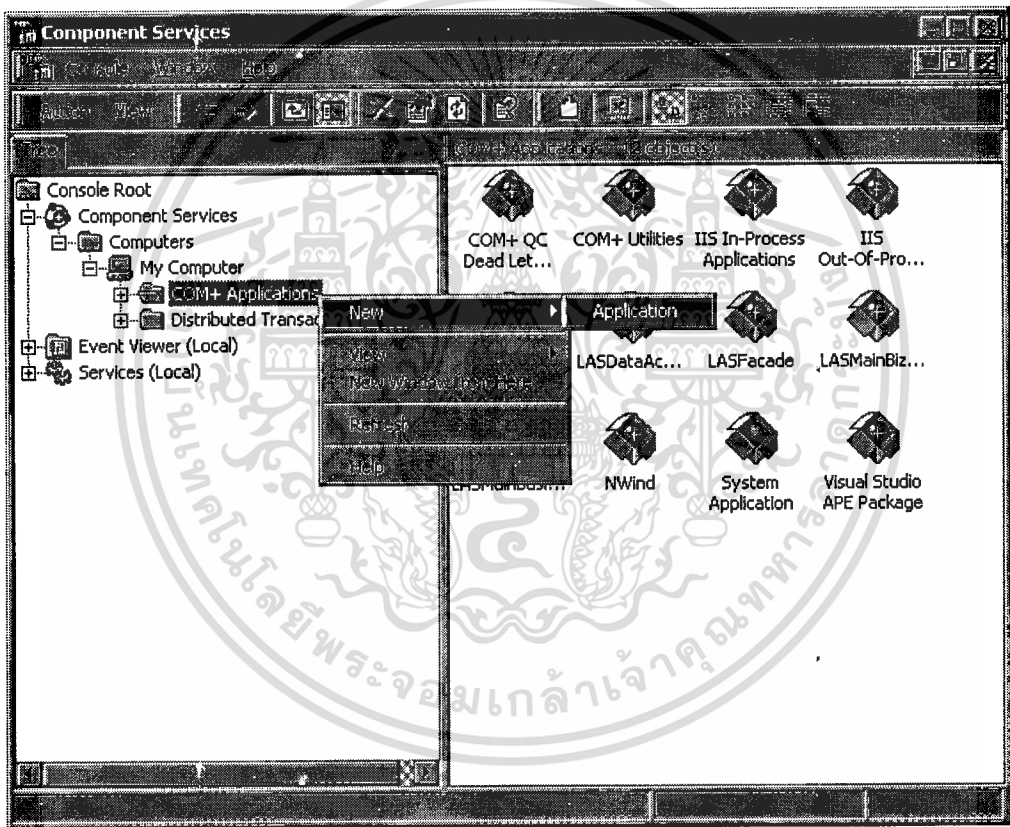
รูปที่ 6.7 ตัวอย่างโปรแกรมที่แสดงการใช้งานอ็อบเจ็กต์ ObjectContext

จากโปรแกรมตัวอย่างจะพบว่าในการสร้างอ็อบเจ็กต์ใหม่จะมีการเรียกใช้ตัวแปร mblnInMTS เพื่อทำการตรวจสอบว่า อ็อบเจ็กต์นั้นทำงานอยู่ภายใต้ Component Service หรือไม่ ถ้าใช่ก็จะทำการสร้างอ็อบเจ็กต์ขึ้นมาใหม่โดยใช้ method CreateInstance ของอ็อบเจ็กต์ ObjectContext แต่หากอ็อบเจ็กต์นั้นไม่ได้ทำงานอยู่ภายใต้ Component Service อ็อบเจ็กต์ใหม่จะถูกสร้างขึ้นโดยใช้คำสั่ง CreateObject แทน และเมื่อการดำเนินการใดๆภายในอ็อบเจ็กต์นั้นเสร็จเรียบร้อยแล้วจะมีการเรียกใช้ Method SetComplete ของอ็อบเจ็กต์ ObjectContext หากว่าการดำเนินการนั้นไม่มีข้อผิดพลาดใดๆเกิดขึ้น แต่หากมีข้อผิดพลาดใดๆเกิดขึ้นจะมีการเรียกใช้งาน Method SetAbort ของอ็อบเจ็กต์ ObjectContext เพื่อรายงานให้ Component Service ทราบและดำเนินการจัดการกับความผิดพลาดดังกล่าวมาแล้วต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 การสร้าง COM+ Application จากคอมโพเนนท์

เมื่อเราดำเนินการสร้าง โค้ดจากแบบจำลองในรูปแบบของการดำเนินการแบบ Round-Trip Engineering มาแล้วจนกระทั่งครบทุกๆ use case แล้วเราจะทำการคอมไพล์ ActiveX DLL Project เหล่านั้น ทำให้ได้คอมโพเนนท์ออกมาในรูปของไฟล์นามสกุล (\*.dll) ที่มีชื่อเดียวกันกับ ActiveX DLL Project เหล่านั้นเช่น LASFacades.dll เพื่อที่จะนำเอาคอมโพเนนท์เหล่านี้ไปไว้ใน Package หรือในอีกชื่อหนึ่งคือ COM+ Application ซึ่งเราจะทำโดยอาศัย Component Services MMC (Microsoft Management Console) ดังรูปที่ 6.8



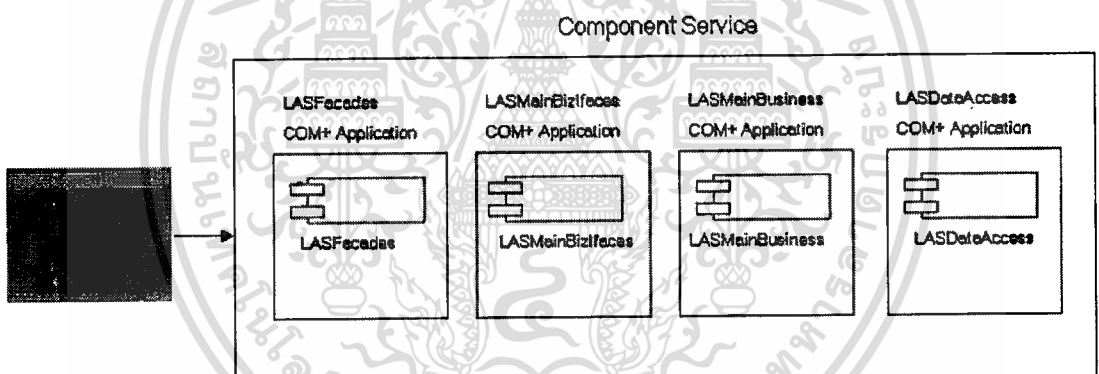
รูปที่ 6.8 Component Services Management Console

ซึ่งการสร้าง Package หรือ COM+ Application นี้จะทำได้โดยการคลิกขวาที่โฟลเดอร์ COM+ Application แล้วเลือก New -> Application จากนั้นก็จะมี Wizard ขึ้นมา เมื่อเราดำเนินการตาม Wizard ไปจนจบเราจะได้ COM+ Application ขึ้นมาอยู่ในลักษณะของโฟลเดอร์ย่อยใน COM+ Application หรือหากมองทางด้านขวาของ MMC จะพบ COM+ Application ในลักษณะของ Icon รูปร่างเป็นกล่องใส่ฮ้อยเบ็ค โดยประเด็นในการออกแบบ Package หรือ COM+ Application คือเราจะออกแบบให้มี Package ที่มีลักษณะอย่างไร เช่น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

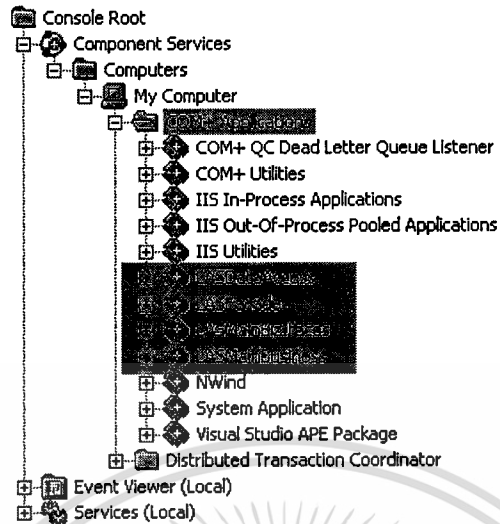
1. ออกแบบให้คอมโพเนนต์ทั้งหมดจะถูกติดตั้งใน Package เดียวกัน
2. ออกแบบให้คอมโพเนนต์ที่เป็นส่วนของ Facades Class ถูกติดตั้งใน Package ที่แยกกันตามบทบาทของ Actor ที่เกี่ยวข้องกับระบบ และติดตั้งคอมโพเนนต์ส่วนที่เป็น Main Business Class และ Data Access Class เอาไว้ด้วยกัน เดียวกัน
3. ออกแบบให้คอมโพเนนต์ที่เป็นส่วนของ Facades Class ถูกติดตั้งใน Package ที่แยกกันตามบทบาทของ Actor ที่เกี่ยวข้องกับระบบ และติดตั้งคอมโพเนนต์ส่วนที่เป็น Main Business Class และ Data Access Class ใน Package ที่แยกจากกัน เดียวกัน

สำหรับระบบที่ทำการพัฒนาจะทำการติดตั้งคอมโพเนนต์ไว้ใน Package ที่แยกจากกันตามประเภทของคอมโพเนนต์ ดังนั้นเราจะได้ Package หรือ COM+ Application ดังแสดงในรูปที่ 6.9 และโครงสร้างของ COM+ Application ในระบบดังรูปที่ 6.10



รูปที่ 6.9 รูปแบบของ COM+ Application ของระบบ

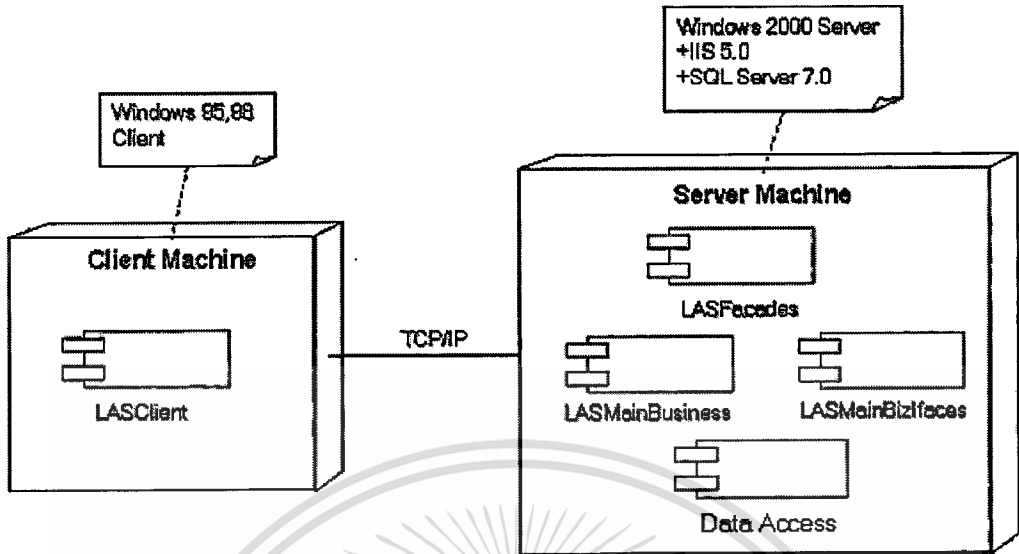
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.10 โครงสร้างของ COM+ Application ของระบบ

#### 6.4 COM+ Application Deployment

หลังจากที่ได้นำเอาคอมโพเนนท์ไปไว้ใน Package หรือในอีกชื่อหนึ่งคือ COM+ Application เรียบร้อยแล้ว งานในส่วนต่อไปคือการนำ Package หรือ COM+ Application ที่ได้ไปใช้งานในเครื่องคอมพิวเตอร์ต่างๆที่ได้จัดรูปแบบในการ Deployment เอาไว้ ซึ่งในการพัฒนาระบบสารสนเทศห้องสมุดนี้ได้ทำการออกแบบบนพื้นฐานของสถาปัตยกรรมแบบ Logical 3-Tier ซึ่งแต่ละชั้นนั้นจะทำงานแยกจากกันแต่อาจจะถูกติดตั้งอยู่ในเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรืออาจจะทำการติดตั้งแต่ละชั้นเอาไว้ในเครื่องคอมพิวเตอร์คนละตัวก็ได้ซึ่งจะถือว่าเป็นรูปแบบ Physical 3-Tier ซึ่งจะมีรายละเอียดในการจัดรูปแบบเพิ่มขึ้นไปอีก แต่เพื่อให้เนื้อหาของโครงการมีความมีความกระชับจึงทำการจัดรูปแบบสถาปัตยกรรมแบบ Logical 3-Tier โดยการจัดรูปแบบการ Deployment นั้นจะอธิบายโดยใช้แผนภาพ Deployment Diagram ของ UML ดังแสดงในรูปที่ 6.11



รูปที่ 6.11 Deployment Diagram ของระบบ

จากรูปที่ 6.11 นั้นเราจะทำการติดตั้งคอมพิวเตอร์ทางธุรกิจทั้งหมดเอาไว้ที่ฝั่ง Server และติดตั้งส่วนของแอปพลิเคชันที่ทำหน้าที่ติดต่อกับผู้ใช้ที่เครื่องคอมพิวเตอร์ที่เป็น Client และจะมีส่วนที่เป็น Application Proxy ที่จะทำให้ Client Application สามารถติดต่อกับคอมพิวเตอร์ที่ฝั่ง Server ได้ ซึ่งการสร้าง Application Proxy นี้จะมีขั้นตอนที่จะได้กล่าวถึงภายหลังในภาคผนวก ก การติดตั้งระบบ

## 6.5 ผลการทดสอบระบบ

หลังจากที่ได้ทำการสร้างระบบเป็นที่เรียบร้อยแล้ว และได้ทำการทดสอบการทำงานของระบบโดยการทดลองใช้งานหน้าที่ต่างๆของระบบพบว่าระบบสามารถทำหน้าที่ต่างๆได้ตามที่ออกแบบไว้ ในขั้นตอนการออกแบบระบบ แต่มีข้อสังเกตบางอย่างที่สังเกตได้คือ การเริ่มต้นระบบในช่วงแรกจะใช้เวลาในการโหลดนานมาก ซึ่งสามารถอธิบายเหตุผลของการที่การทำงานในช่วงแรกที่ยังไม่มีอ็อบเจกต์อยู่ในหน่วยความจำ แต่หลังจากที่ Client Application ใช้งานอ็อบเจกต์นั้นเสร็จ Component Service จะยังไม่ทำลายอ็อบเจกต์นั้นทันที แต่จะรอช่วงระยะเวลาหนึ่ง หากมีการร้องขออ็อบเจกต์นั้นมาอีก Component Service จะใช้การจัดการนำอ็อบเจกต์ในหน่วยความจำเหล่านี้กลับมาใช้งานได้ใหม่ ทำให้การทำงานในช่วงหลังนั้นทำได้เร็วขึ้น ดังนั้นการที่ Server มีปริมาณหน่วยความจำที่มากจะยังเป็นผลดีต่อการทำงานของระบบ

## บทที่ 7

### สรุปผลการพัฒนา

โครงการพัฒนาระบบสารสนเทศเพื่อการบริหารงานห้องสมุดด้วยวิธียูเอ็มแอลเบส มีจุดประสงค์ในการที่จะประยุกต์แนวความคิดทางด้านการออกแบบระขอฟท์แวร์ในรูปแบบคอมโพเนนท์ ซึ่งเป็นรูปแบบที่ค่อนข้างจะทำให้เกิดการนำมาใช้ใหม่ในทางปฏิบัติได้ค่อนข้างง่ายกว่าการใช้แนวทางออกแบบเชิงวัตถุ แต่อย่างไรก็ตามหลักการเชิงวัตถุก็ยังสามารถประยุกต์มาใช้ในการพัฒนาในรูปแบบคอมโพเนนท์เบสได้เป็นอย่างดี ถึงแม้ว่าจะมีหลักการในบางส่วนที่ค่อนข้างจะแตกต่างกัน เช่นแนวโน้มในการรักษาสภาพของวัตถุภายในระบบของระบบที่ถูกพัฒนาจากแนวคิดแบบวัตถุกับระบบที่ถูกพัฒนาจากแนวคิดแบบคอมโพเนนท์เบส ซึ่งต่างก็มีข้อดีข้อเสียที่ต้องนำไปเปรียบเทียบตามเงื่อนไขของสภาพแวดล้อมต่างๆ ในการพัฒนา แต่ข้อดีที่เห็นได้ชัดจากการดำเนินการพัฒนาโครงการคือทำให้เห็นความเป็นอิสระในการเปลี่ยนแปลงส่วนต่างๆ ของระบบ โดยที่ไม่กระทบกระเทือนต่อส่วนอื่นๆ ที่ทำงานอยู่ ซึ่งเป็นสิ่งที่จำเป็นในการพัฒนาระบบที่สามารถยืดขยายได้ในอนาคต

สิ่งสำคัญประการหนึ่งที่ได้จากการดำเนินการพัฒนาโครงการนี้คือ แนวความคิดในการประยุกต์ใช้ ภาษาในการสร้างแบบจำลองในการออกแบบและพัฒนาระบบซึ่งในที่นี้ใช้ UML กับแบบจำลองกระบวนการในการพัฒนาระบบ และ การใช้งาน CASE Tool ในการพัฒนาระบบ ซึ่งทั้งสามส่วนนี้มีส่วนช่วยให้การออกแบบระบบนั้นสามารถทำได้อย่างเป็นระเบียบมากยิ่งขึ้น ไม่ว่าจะเป็นการใช้คุณสมบัติในการทำ Round-Trip Engineer ของตัว CASE Tool เอง และการใช้แนวคิดในการออกแบบที่ชื่อ use case และสถาปัตยกรรมของระบบเป็นหลักในการดำเนินการพัฒนา รวมไปถึงการสร้างเอกสารจากแบบจำลองขบวนการในการพัฒนา, สิ่งต่างๆ เหล่านี้ทำให้ข้อมูลที่ได้จากการพัฒนาทั้งหมดสามารถถูกติดตาม (tracking) ได้ตลอด ทำให้ช่วยลดปัญหาที่เกิดจากการพัฒนาในแบบเดิมๆ ที่ค่อนข้างจะขาดความสัมพันธ์ของแบบจำลองในการพัฒนาระบบและตัวโปรแกรม แต่อย่างไรก็ตามเนื่องจากเครื่องมือและสภาพแวดล้อมที่ใช้ในการพัฒนานั้นไม่ได้สนับสนุนต่อแนวคิดเชิงวัตถุอย่างแท้จริง จึงมีความจำเป็นที่เราจะต้องปฏิบัติตามกรอบงาน (Framework) ของสภาพแวดล้อมและเครื่องมือที่เราใช้งาน เพื่อที่จะประยุกต์ใช้เครื่องมือที่มีกับแนวคิดที่ใช้ในการพัฒนาให้เกิดประสิทธิภาพมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการดำเนินการพัฒนาระบบเพื่อทำงานในสภาพแวดล้อมจริงนั้น ประสิทธิภาพก็เป็นอีกประเด็นหนึ่งที่ต้องทำการพิจารณา ซึ่งขอบเขตของการพัฒนาโครงการนี้ไม่ได้ครอบคลุมไปถึงประสิทธิภาพในการทำงานเป็นหลัก แต่ในความเป็นจริงแล้วจะเห็นว่าในการสื่อสารต่างๆระหว่างวัตถุในระบบล้วนเป็นการสื่อสารที่ต้องมีชุดข้อมูลจำนวนมากที่ถูกส่งผ่านไปตามวัตถุที่อยู่ในชั้นต่างๆ ซึ่งจะเป็นประเด็นที่ต้องศึกษาต่อไปว่าหากจะพัฒนาระบบโดยใช้เทคนิคคอมพิวเตอร์เน็ตเวิร์กที่มีประสิทธิภาพน่าเชื่อถือ นั้นจะต้องทำอย่างไร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- Booch, G. 1994. **Object-Oriented Analysis and Design with Applications**. 2<sup>nd</sup>.ed.  
New York : Addison-Wesley.
- Eriksson, H. and Penker, M. 1998. **UML Toolkit**. New York : John Wiley & Sons.
- Fowler, M. and Scott, K. 1999. **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. 2<sup>nd</sup>.ed. New York : Addison-Wesley.
- Jacobson, I. *et.al.* 1999. **The Unified Software Development Process**. New York : Addison-Wesley.
- Kirtland, M. 1999. **Designing Component-Based Applications**. Washington : Microsoft Press.
- Larman, C. 1998. **Applying UML and Patterns : an introduction to object-oriented analysis and design** : Prentice Hall.
- Lewis, T. 1999. **VB COM**. Birmingham : Wrox Press.
- Lhotka, R. 1998. **Visual Basic 6.0 Business Objects**. Birmingham : Wrox Press.
- Reed, P.R. 1999. **Developing Applications with Visual Basic and UML**. New York : Addison-Wesley.
- Rowlett, T.W. 2000. **Object-Oriented Development Process, The: Developing and Managing A Robust Process for Object-Oriented Development**. New York : Prentice Hall PTR.
- Stevens, P and Pooley, R. J. 2000. **Using UML: Software Engineering with Objects and Components Revised Edition**. New York : Addison-Wesley.
- Sundblad, S. and Sundblad, P. 2000. **Designing for Scalability with Microsoft Windows DNA**. Washington : Microsoft Press.

## ภาคผนวก ก การติดตั้งระบบ

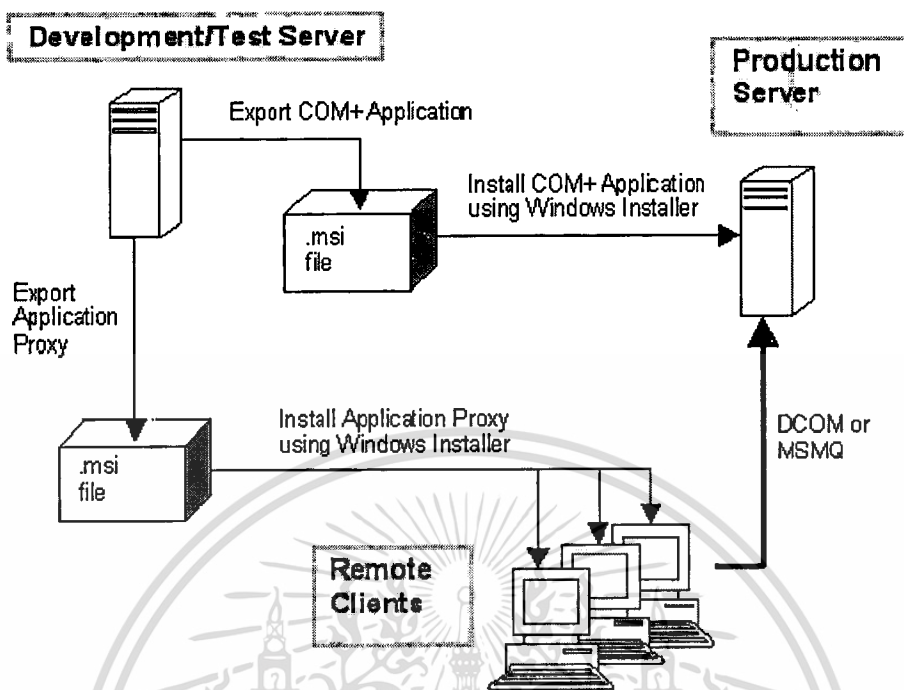
เมื่อทำการพัฒนาระบบและดำเนินการทดสอบระบบเป็นที่เรียบร้อยแล้ว ในการที่เราจะทำการโยกย้ายระบบจากสภาพแวดล้อมที่ใช้ในการพัฒนาไปใช้ในสภาพแวดล้อมในการทำงานจริงนั้น มีสิ่งที่จะต้องดำเนินการที่สำคัญอยู่ 2 ประการคือ การย้ายระบบจากคอมพิวเตอร์ที่ใช้ทำการพัฒนาไปยังคอมพิวเตอร์ที่จะใช้งานจริง และการติดตั้งส่วนประกอบต่างๆในเครื่องคอมพิวเตอร์ที่จะใช้งานจริง ซึ่งจะมีรายละเอียดต่างๆดังต่อไปนี้

### ก.1 การย้ายระบบจากคอมพิวเตอร์ที่ใช้พัฒนาไปยังคอมพิวเตอร์ที่ใช้งาน

การทำการโยกย้ายระบบที่ทำการพัฒนาจากเครื่องที่ใช้พัฒนาไปยังเครื่องที่ใช้งานหรือที่เรียกอีกอย่างว่า Deployment สำหรับระบบที่ทำการพัฒนาขึ้นมาแล้วมีสิ่งที่จะต้องดำเนินการ 2 อย่างคือ

1. การ Export COM+ Application
2. การสร้าง Application Proxy สำหรับ Client Application

สำหรับในส่วนแรกนั้นจะเป็นการ Export Package หรือ COM+ Application ที่ได้ทำการพัฒนาไว้ดังที่กล่าวเอาไว้ในบทที่ 6 การสร้างระบบ ออกมาในรูปแบบของชุดติดตั้งซึ่งในที่นี้คือ Microsoft Installer (MSI) File เพื่อนำไปใช้ติดตั้งในเครื่องที่จะใช้เป็น Server และอีกส่วนหนึ่งคือการสร้าง Application Proxy นั้นจะเป็นส่วนสำคัญที่ทำให้ Client Application นั้นสามารถทำการสื่อสารกับคอมโพเนนท์ที่มันอ้างอิงได้ไม่ว่าคอมโพเนนท์นั้นจะอยู่ในเครื่องคอมพิวเตอร์เดียวกันกับ Client Application นั้นหรืออยู่ต่างเครื่องกันก็ตาม โดย Application Proxy นั้นจะถูกสร้างออกมาในรูปแบบของชุดติดตั้ง Microsoft Installer (MSI) File เช่นเดียวกับในกรณีของการ Export Package หรือ COM+ Application ซึ่งโดยสรุปแล้วหลักการของการ Deployment นั้นจะต้องอาศัยการดำเนินการทั้ง 2 ส่วนนี้ดังแสดงในรูปแบบที่ ก.1

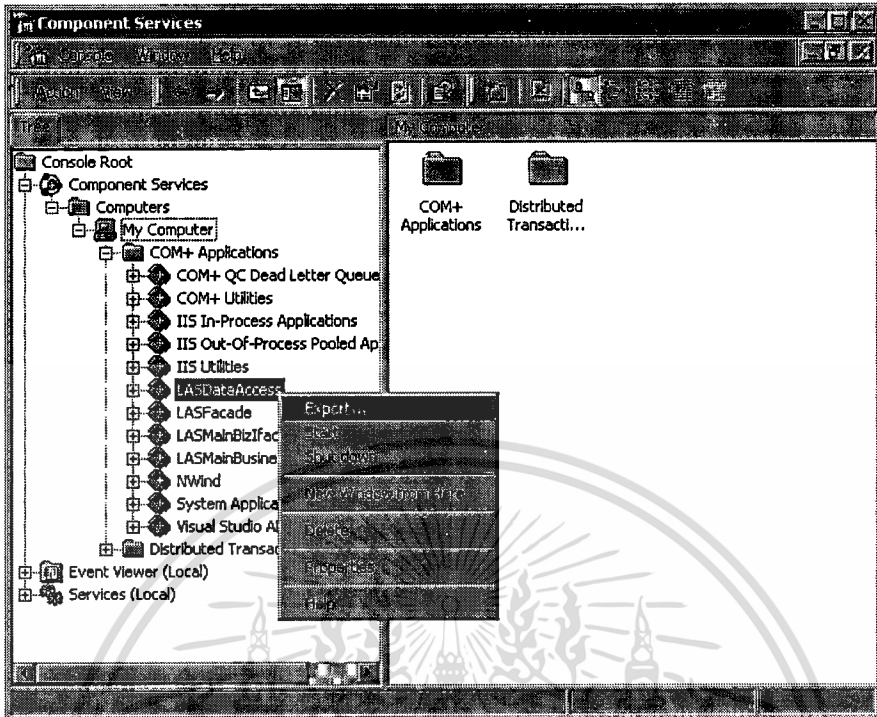


รูปที่ ก.1 หลักการในการทำ Deployment

สำหรับการดำเนินการในส่วนนี้จะใช้ Component Service Management Console ดังที่กล่าวไว้ในบทที่ 6 การสร้างระบบ โดยจะมี Wizard ที่ช่วยในการดำเนินการทั้ง 2 ส่วนซึ่งขั้นตอนในการดำเนินการมีดังต่อไปนี้

1. เปิด Component Service Management Console ขึ้นมาแล้วเลือก COM+ Application ที่ต้องการ Export จากนั้นให้คลิกเมาส์ที่ปุ่มขวาแล้วเลือก Export ดังรูปที่ ก.2 จะปรากฏหน้าต่างของ COM Application Export Wizard ขึ้นมาดังรูปที่ ก.3
2. เลือกโฟลเดอร์ที่ต้องการเก็บ Exported Package พร้อมทั้งตั้งชื่อของ Exported Package และเลือกรูปแบบในการ Export (COM+ Application Export หรือ Application Proxy Export) ดังรูปที่ ก.4
3. Windows จะทำการสร้าง Exported Package ในรูปของ Windows Installer File (\*.MSI) และแสดงข้อความบอกว่าการ Export เสร็จเรียบร้อยดังรูปที่ ก.5
4. เมื่อไปดูที่โฟลเดอร์ที่กำหนดไว้ในขั้นตอนที่ 2 จะพบไฟล์ 2 ไฟล์ดังรูปที่ ก.6 คือ MSI File และ CAP (\*.cap) File ซึ่งมีลักษณะเดียวกับ MSI File ที่สามารถให้ Browser ทำการดาวน์โหลดไปติดตั้งได้ในลักษณะเดียวกันกับ JAVA Applet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

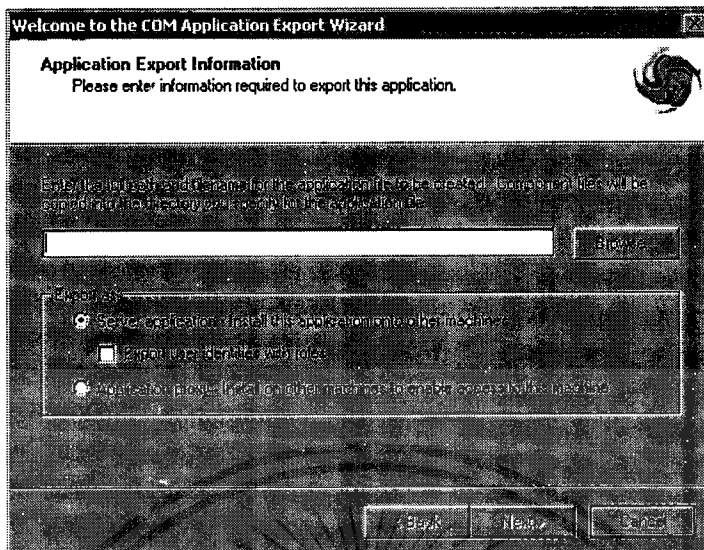


รูปที่ ก.2 การ Export COM+ Application



รูปที่ ก.3 Export Wizard ขั้นตอนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 Export Wizard ขั้นตอนที่ 2



รูปที่ ก.5 เสร็จสิ้นการทำ Export Wizard

Name	Size	Type
LASDataAccess.MSI	96 KB	Windows Installer Package
LASDataAccess.MSI.cab	58 KB	WinZip File

รูปที่ ก.6 ไฟล์ที่ได้หลังจากการทำ Export Wizard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ก.2 โครงสร้างเฉพาะทางซอฟต์แวร์ที่เหมาะสมต่อการใช้งานระบบ (Software Specification)

เนื่องจากระบบที่ทำการพัฒนานี้มีพื้นฐานการทำงานอยู่บน Microsoft Windows Platform ดังนั้นจึงมีความจำเป็นในการเลือกใช้ซอฟต์แวร์ที่เหมาะสมกับการทำงานของระบบ โดยจะทำการกำหนดรายละเอียดของลักษณะเฉพาะต่างๆของซอฟต์แวร์ที่จะใช้งานในระบบโดยจะแบ่งออกเป็น 2 ส่วนคือ ลักษณะของซอฟต์แวร์ที่มีความจำเป็นต้องใช้งานที่ฝั่ง Client และ Server โดยมีรายละเอียดดังนี้

### 1. ลักษณะของซอฟต์แวร์ที่มีความจำเป็นต้องใช้งานที่ฝั่ง Client

เนื่องจากระบบมีพื้นฐานการทำงานอยู่บน Microsoft Windows Platform ดังนั้นระบบปฏิบัติการที่เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client ใช้งานนั้นควรจะเป็นระบบปฏิบัติการตระกูล Windows

### 2. ลักษณะของซอฟต์แวร์ที่มีความจำเป็นต้องใช้งานที่ฝั่ง Server

สำหรับในส่วนของ Server นี้ระบบปฏิบัติการที่กำหนดให้ใช้คือ Windows 2000 Server เนื่องจาก ระบบนี้ถูกพัฒนาขึ้นมาโดยใช้โครงสร้างพื้นฐานของการทำงานที่เป็น Component Service หรือเรียกว่า COM+ ซึ่งมีในระบบปฏิบัติการ Windows 2000 หรือในระบบปฏิบัติการที่ใหม่กว่าเท่านั้น นอกจากนี้ Server จะต้องทำการติดตั้ง Web Server ซึ่งในที่นี้เราใช้ IIS 5.0 เป็น Web Server เพื่อให้บริการในส่วนของ Web Application และฐานข้อมูลที่ระบบถูกพัฒนาขึ้นมาให้ทำงานด้วยนั้นจะใช้ Microsoft SQL Server 7.0 หรือใหม่กว่า

## ก.3 การติดตั้งส่วนประกอบต่างๆในเครื่องคอมพิวเตอร์ที่จะใช้งาน

การติดตั้งส่วนประกอบต่างๆในเครื่องคอมพิวเตอร์ที่จะใช้งานนั้นแบ่งออกได้เป็น 2 ส่วนคือการติดตั้งด้าน Client และการติดตั้งส่วนประกอบทางด้าน Server โดยสำหรับรายละเอียดในแต่ละส่วนนั้นมีดังต่อไปนี้

### 1. การติดตั้งส่วนประกอบทางด้าน Client สำหรับส่วนประกอบที่จำเป็นในการติดตั้งที่ฝั่ง Client นี้ประกอบด้วย

- ชุดติดตั้ง Client Application
- Application Proxy ที่ได้สร้างขึ้นมาจากในหัวข้อ ก.1

โดยสำหรับการติดตั้งทั้ง 2 ส่วนนี้สามารถทำได้ง่ายเนื่องจากมีลักษณะการติดตั้งในรูปแบบที่เป็นการใช้ Wizard ดังนั้นจะไม่ขอกกล่าวถึงรายละเอียดไว้ ณ ที่นี้

2. การติดตั้งส่วนประกอบทางด้าน Server ส่วนประกอบที่จำเป็นในการติดตั้งที่ฝั่ง Client นี้ ประกอบด้วย

- ชุดติดตั้ง COM+ Application ซึ่งในที่นี้มีทั้งหมด 4 Package คือ LASFacades.MSI, LASMainBizIfaces.MSI, LASMainBusiness.MSI, LASDataAccess.MSI
- DDL Script ที่ใช้ในการสร้างฐานข้อมูลสำหรับระบบ ได้แก่ไฟล์ LASProject.SQL
- Registration Entry File เป็นไฟล์ Registry ของ Windows ที่ใช้ในการกำหนด Connection String ในการติดต่อฐานข้อมูล เนื่องจากระบบที่ทำการพัฒนาขึ้นนี้จะใช้การติดต่อกับฐานข้อมูลที่เป็น OLE DB ทำให้สามารถติดต่อไปยังฐานข้อมูลโดยใช้ Connection String ไม่ต้องทำการสร้าง DSN เช่นในกรณีที่ใช้ ODBC เป็นตัวกลางในการติดต่อฐานข้อมูล โดยรายละเอียดของไฟล์นี้ได้แสดงไว้ในรูปที่ ก.7
- Web Site Files เป็นชุดของไฟล์ที่เป็น Web Application ของระบบ โดยสำหรับโครงสร้างของ Web Site นั้นจะแสดงไว้ในรูปที่ ก.8 โดยการติดตั้งนั้นเพียงแค่ทำการคัดลอกไฟล์เตอร์ของ Web Site ซึ่งในที่นี้คือ LASWebCatalog ไปไว้ที่ไฟล์เตอร์ที่เป็นที่เก็บเอกสาร HTML ของ Web Server ซึ่งในที่นี้ใช้ IIS 5.0 โดยไฟล์เตอร์ดังกล่าวหากทำการติดตั้ง IIS 5.0 โดยใช้การกำหนดแบบปกติแล้วจะอยู่ที่ [Drive ที่ติดตั้ง IIS]/InetPub/wwwroot/ โดยที่ไฟล์เตอร์รากของ Web Site นั้นจะเก็บไฟล์ที่เป็น Server-Side Script ซึ่งทำหน้าที่ในการติดต่อกับ COM+ Application และส่งผลลัพธ์ในรูปแบบของ HTML ไปยัง Browser ส่วนในไฟล์เตอร์ /Include นั้นจะเก็บไฟล์ที่เป็น Server-Side Include ไฟล์ต่างๆ ส่วนในไฟล์เตอร์ /Images นั้นจะเก็บไฟล์ภาพต่างๆที่ใช้ใน Web Site รวมไปถึงไฟล์ Style Sheet ด้วย

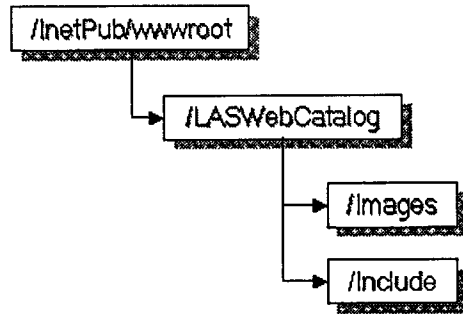
Windows Registry Editor Version 5.00

[HKEY\_CURRENT\_USER\Software\VB and VBA Program  
Settings\LASnTierProject]

[HKEY\_CURRENT\_USER\Software\VB and VBA Program  
Settings\LASnTierProject\ConnectionString]

"ConnectSQL"="Provider=SQLOLEDB.1;User ID=sa;Initial Catalog=lasProject;Data  
Source=[ชื่อ Database Server]"

รูปที่ ก.7 Registration Entry ของระบบ



รูปที่ ก.8 โครงสร้างของ Web Application



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข การใช้งานระบบ

สำหรับการใช้งานระบบสารสนเทศเพื่อการบริหารงานห้องสมุดนั้นจะแบ่งออกเป็น 2 ส่วนคือส่วนที่เป็นของเจ้าหน้าที่ ซึ่งจะเป็ Windows Application และส่วนที่เปิดให้บริการสาธารณะผ่านทางอินเทอร์เน็ต ซึ่งแต่ละส่วนมีรายละเอียดในการใช้งานดังต่อไปนี้

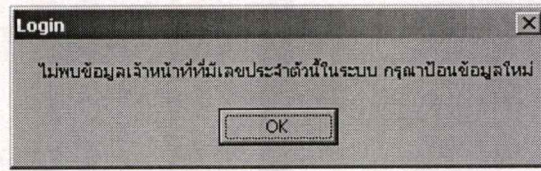
### ข.1 ระบบงานในส่วนที่เป็น Windows Application

หลังจากที่ได้ทำการติดตั้งส่วนประกอบต่างๆของระบบเป็นที่เรียบร้อยแล้ว เราสามารถใช้งานระบบโดยทำการ Double-Click ที่ Icon ของโปรแกรมที่ได้ทำการติดตั้งไว้ จากนั้นจะมีหน้าจอหลักของระบบปรากฏขึ้น โดยที่ผู้ใช้จะยังไม่สามารถจะดำเนินการใดๆกับระบบได้จนกว่าจะทำการเข้าใช้งานระบบโดยการเลือกเมนูเข้าสู่ระบบ จากนั้นจะปรากฏหน้าจอให้ป้อนชื่อผู้ใช้และรหัสผ่านดังรูปที่ ข.1 ถ้าระบบตรวจสอบแล้วพบว่าชื่อผู้ใช้และรหัสผ่านถูกต้อง ผู้ใช้จะสามารถใช้งานเมนูต่างๆของระบบจากหน้าจอหลักดังรูปที่ ข.4 ได้

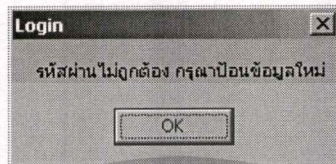


รูปที่ ข.1 หน้าจอ Log In เพื่อเข้าใช้งานระบบห้องสมุด

ในการป้อนเลขประจำตัวและรหัสผ่านเพื่อขอเข้าใช้งานระบบนี้หากว่าระบบไม่พบเลขประจำตัวของเจ้าหน้าที่นั้นในระบบ ระบบจะมีข้อความแจ้งดังรูปที่ ข.2 และหากผู้ใช้ทำการป้อนรหัสผ่านไม่ถูกต้องระบบจะมีข้อความแจ้งเตือนดังรูปที่ ข.2 โดยผู้ใช้จะสามารถป้อนรหัสผ่านใหม่เพื่อเข้าสู่ระบบได้ทั้งหมด 4 ครั้ง หากยังไม่ถูกต้องระบบจะยกเลิกการ Log In



รูปที่ ข.2 ข้อความเตือนในกรณีไม่พบเจ้าหน้าที่ที่มีเลขประจำตัวนี้ในระบบ



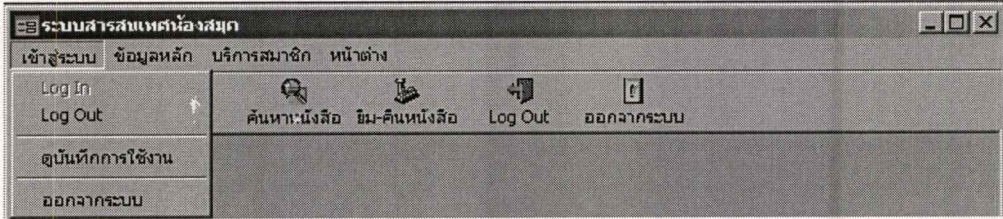
รูปที่ ข.3 ข้อความเตือนในกรณีรหัสผ่าน ไม่ถูกต้อง



รูปที่ ข.4 หน้าจอหลักของระบบห้องสมุด

สำหรับในส่วนหน้าจอหลักจะพบว่ามีเมนูซึ่งแบ่งออกได้เป็น 3 ส่วนคือ ส่วนของการเข้าใช้งานระบบ ส่วนของการจัดการข้อมูลของระบบ และส่วนของการให้บริการ ซึ่งหากทำการเลือกดูในแต่ละส่วนจะพบเมนูย่อยๆ ดังรูปที่ ข.5 ถึง ข.7 ตามลำดับ

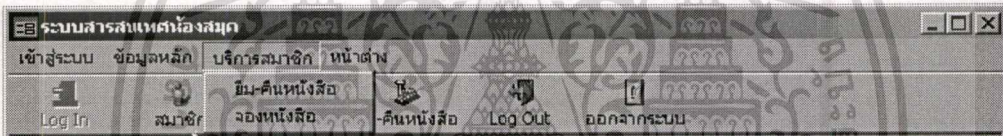
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.5 รายละเอียดของเมนูส่วนการเข้าใช้งานระบบ



รูปที่ ข.6 รายละเอียดของเมนูส่วนการจัดการข้อมูลหลัก



รูปที่ ข.7 รายละเอียดของเมนูส่วนการให้บริการ

สำหรับรายละเอียดในแต่ละส่วนมีดังต่อไปนี้

- 1) ส่วนการเข้าใช้งานระบบ เมนูส่วนนี้จะเกี่ยวกับการเข้าใช้งานระบบ ประกอบด้วยเมนูย่อยๆ ได้แก่
  - 1.1)เมนู Log In การทำงานในส่วนของเมนูนี้จะเป็นการตรวจสอบการสิทธิในการเข้าใช้งานระบบของผู้ใช้ดังที่ได้กล่าวไปแล้วในตอนต้น
  - 1.2)เมนู Log Out การทำงานในส่วนของเมนูนี้จะเป็นการบันทึกการเลิกใช้งานระบบของผู้ใช้และทำให้ระบบกลับไปสู่สถานะที่ต้องทำการ Log In เข้าใช้งานระบบอีกครั้งจึงจะใช้งานระบบได้
  - 1.3)เมนูดูบันทึกการใช้งาน การทำงานในส่วนของเมนูนี้จะเป็นการเรียกดูประวัติการเข้าใช้งานระบบของผู้ใช้ระบบขณะนั้น โดยจะแสดงหน้าจอข้อมูลการเข้าใช้งานระบบของผู้ใช้ระบบดังรูปที่ ข.8

บันทึกการเข้าใช้งานระบบ		
รหัสเจ้าหน้าที่	0	
ชื่อ-นามสกุล	นาย	ไพศาล กาญจนสุกรีรักษ์
ตำแหน่ง	admin	
วันที่เข้าใช้ระบบ	เวลาที่เข้าใช้ระบบ	เวลาที่ออกจากระบบ
3/3/2545	3:22:24	3:23:57
3/3/2545	3:24:03	3:24:27
3/3/2545	3:26:27	3:26:58
4/3/2545	9:27:39	9:28:23
4/3/2545	11:07:41	11:08:35
4/3/2545	11:23:29	11:23:48
4/3/2545	14:06:39	14:07:32
4/3/2545	14:07:38	14:07:45
4/3/2545	14:38:49	14:39:19
4/3/2545	17:32:34	
12/3/2545	6:11:11	6:12:04
15/3/2545	6:57:14	6:59:17

15 รายการ

รูปที่ ข.8 หน้าจอแสดงข้อมูลการเข้าใช้งานระบบของเจ้าหน้าที่

1.4) เมื่อนอกจากระบบ การทำงานในส่วนของเมนูนี้จะเป็นการการบันทึกการเลิกใช้งานระบบของผู้ใช้หากผู้ใช้นั้นยังไม่ทำการ Log Out จากระบบและทำการปิดการทำงานจากระบบ

2. ส่วนการจัดการข้อมูลของระบบ เมนูส่วนนี้จะเกี่ยวกับการดูแลจัดการข้อมูลหลักของระบบ เช่น ข้อมูลสมาชิก ข้อมูลหนังสือ เป็นต้น โดยจะประกอบด้วยเมนูย่อยๆ ได้แก่

2.1) เมนูสมาชิก การทำงานในส่วนของเมนูนี้จะเป็นการจัดการกับข้อมูลของสมาชิกในระบบ เช่น การเพิ่ม ลบ แก้ไข ข้อมูลสมาชิกในระบบ โดยเมื่อเลือกเมนูนี้จะปรากฏหน้าจอให้ทำการค้นหาสมาชิกที่ต้องการจากเงื่อนไขต่างๆ เช่น ค้นหาจากชื่อสมาชิก ประเภทสมาชิก เป็นต้น ดังรูปที่ ข.9 โดยผู้ใช้สามารถทำการลบหรือทำการเพิ่มสมาชิกใหม่เข้าสู่ระบบ และเมื่อผู้ใช้เลือกที่จะทำการแก้ไขข้อมูลสมาชิกที่ปรากฏในรายการที่ค้นหาได้ ระบบจะแสดงหน้าจอข้อมูลสมาชิกนั้นขึ้นมาดังรูปที่ ข.10

**ข้อมูลสมาชิก**

ค้นหาจาก **สมาชิกที่ไม่มีสิทธิทำกรอขึ้น**

หมายเลขสมาชิก   
 ชื่อสมาชิก   
 ประเภทสมาชิก   
 คณะ

เลขประจำตัว	ชื่อ	นามสกุล
401400168	วิมลพลา	สกุลโศตพานิชย์
401400160	สิริสิริศักดิ์	ไมตรี
401400158	ภกพร	ไมตรี
401400165	ไพพรรณวดี	ธรรมลังกา

4 : objects

รูปที่ ข.9 หน้าจอหลักของงานจัดการข้อมูลสมาชิก

**สมาชิก**

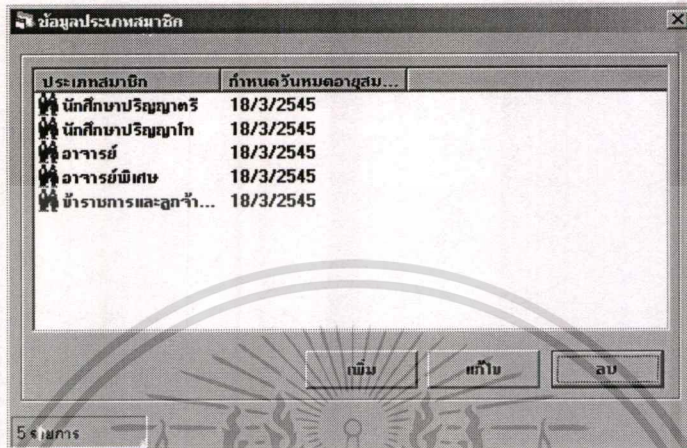
ข้อมูลสมาชิก

หมายเลขประจำตัว   
 ชื่อ    
 นามสกุล   
 ประเภทสมาชิก   
 คณะ   
 Email   
 จำนวนหนังสือที่ยืมทั้งหมด   
 ถ้าปรับล้างชำระ    
 วันหมดอายุการเป็นสมาชิก

รูปที่ ข.10 หน้าจอข้อมูลสมาชิก

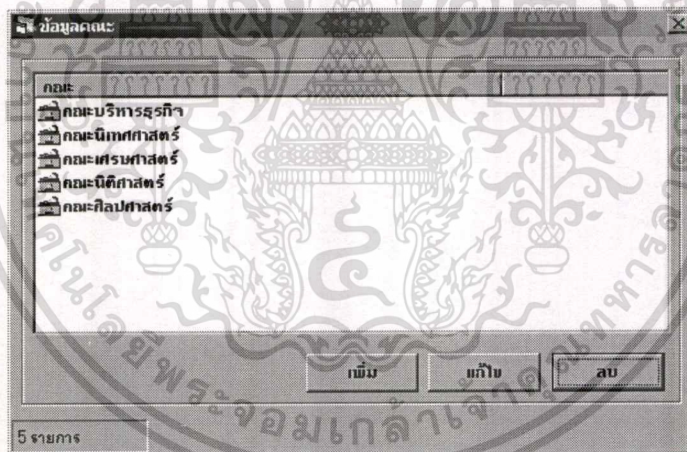
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับหน้าจอข้อมูลสมาชิกนี้จะมีหน้าจอย่อยในส่วนของ ประเภทสมาชิก คณะ และการชำระค่าปรับ ซึ่งได้แสดงในรูป ข.11 ถึง ข.13 ตามลำดับ

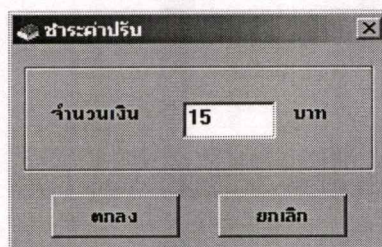


ประเภทสมาชิก	กำหนดรับทบทวนค่าสมาชิก...
นักศึกษาปริญญาตรี	18/3/2545
นักศึกษาปริญญาโท	18/3/2545
อาจารย์	18/3/2545
อาจารย์พิเศษ	18/3/2545
ข้าราชการและลูกจ้าง...	18/3/2545

รูปที่ ข.11 หน้าจอหลักข้อมูลประเภทสมาชิก



รูปที่ ข.12 หน้าจอหลักข้อมูลคณะ



รูปที่ ข.13 หน้าจอการชำระค่าปรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสำหรับหน้าจอประเภทสมาชิกและข้อมูลขณะนั้นผู้ใช้สามารถเรียกขึ้นมาได้โดยการกดปุ่มที่อยู่ด้านข้างของ Combo Box ของข้อมูลประเภทสมาชิกและข้อมูลขณะในกรณีที่ผู้ใช้ต้องการเพิ่มข้อมูลนอกเหนือไปจากที่มีอยู่ในระบบขณะนั้น โดยที่หน้าจอทั้งสองนั้นผู้ใช้สามารถทำการเพิ่ม ลบ หรือแก้ไขข้อมูลได้ โดยในส่วนของ การแก้ไขข้อมูลนั้นเมื่อผู้ใช้ทำการเลือกที่จะทำการแก้ไขข้อมูล ระบบจะทำการแสดงหน้าจอข้อมูลของรายการนั้นๆขึ้นมาให้ผู้ใช้สามารถทำการแก้ไขข้อมูลได้ เช่นในกรณีของการแก้ไขข้อมูลประเภทสมาชิกดังแสดงในรูปที่ ข.14 ส่วนหน้าจอการชำระค่าปรับนั้นผู้ใช้จะทำการเรียกใช้ได้ในกรณีที่สมาชิกคนนั้นมีค่าปรับค้างชำระกับทางห้องสมุด เพื่อทำการหักค่าปรับของสมาชิกที่ถูกบันทึกในระบบ

รูปที่ ข.14 หน้าจอข้อมูลประเภทสมาชิก

2.2) เมนูหนังสือ การทำงานในส่วน of เมนูนี้จะเป็นการจัดการกับข้อมูลหนังสือในระบบ เช่น การเพิ่ม ลบ แก้ไข ข้อมูลหนังสือในระบบ โดยเมื่อเลือกเมนูนี้จะปรากฏหน้าจอให้ทำการค้นหาหนังสือที่ต้องการจากเงื่อนไขต่างๆ เช่น ค้นหาจากชื่อเรื่อง หัวเรื่อง ชื่อผู้แต่ง เป็นต้นดังรูปที่ ข.15 โดยผู้ใช้สามารถทำการลบหรือแก้ไขข้อมูลหนังสือที่ปรากฏในรายการที่ค้นหาได้ หรือทำการเพิ่มหนังสือใหม่เข้าสู่ระบบ โดยระบบจะแสดงหน้าจอข้อมูลหนังสือนั้นขึ้นมาดังรูปที่ ข.16

ค้นหาหนังสือ

กำหนดเงื่อนไข

ค้นหา

เริ่มใหม่

ออก

ค้นหาจาก คำค้น

ชื่อเรื่อง

เลขเรียกหนังสือ	ISBN	ชื่อ
QA76.73.J38 b638ก	9745538582	การเขียนโปรแกรมเชิงวัตถุภาษาจาวา
QA76.73.J38 s636ก	9748580199	กระแทะเปลือจาวา = Object-orient

2 : objects

เพิ่ม แก้ไข ลบ

รูปที่ ข.15 หน้าจอหลักของงานจัดการข้อมูลหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสือ

เลขเรียกหนังสือ: QA76.73.J38 s636n

ชื่อเรื่อง: กระแทะเปลือกาวา = Object-oriented programming with Java Language

ประเภทหนังสือ: หนังสือทั่วไป

พิมพ์ที่: กรุงเทพฯ, จีรวัดณ์เอ็กแมทส

ISBN: 9748580199

ปีที่พิมพ์: 2543

ครั้งที่พิมพ์: 1

จำนวนหน้า: 209

หมายเหตุ:

ผู้แต่ง: รุ่งโรจน์ โพนคำ

หัวเรื่อง: การเขียนโปรแกรมเชิงวัตถุ

Barcode	ชื่อประเภทย่อย	วันที่ทำการลง...	สถานะ
K025430042	ไม้ซี	18/8/2543	ว่าง

ตกลง    ยกเลิก    บันทึก

### รูปที่ ข.16 หน้าจอข้อมูลหนังสือ

สำหรับหน้าจอข้อมูลหนังสือนั้น จะมีหน้าจอย่อยในส่วนของข้อมูลประเภทหนังสือ สำนักพิมพ์ ผู้แต่ง หัวเรื่อง และเลขทะเบียนหนังสือ ดังแสดงในรูปที่ ข.17 ถึง ข.21 ตามลำดับ

ข้อมูลประเภทหนังสือ

ประเภทหนังสือ	อัตราค่าปรับ
หนังสือทั่วไป	5
หนังสือสำรอง	100

เพิ่ม    แก้ไข    ลบ

2 รายการ

### รูปที่ ข.17 หน้าจอหลักของข้อมูลประเภทหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทสมาชิก	กำหนดวันหมดอายุสม...
<input checked="" type="checkbox"/> นักศึกษาปริญญาตรี	18/3/2545
<input checked="" type="checkbox"/> นักศึกษาปริญญาโท	18/3/2545
<input checked="" type="checkbox"/> อาจารย์	18/3/2545
<input checked="" type="checkbox"/> อาจารย์พิเศษ	18/3/2545
<input checked="" type="checkbox"/> ข้าราชการและลูกจ้...	18/3/2545

5 รายการ

เพิ่ม แก้ไข ลบ

รูปที่ ข.18 หน้าจอหลักของข้อมูลสำนักพิมพ์

ผู้แต่ง
<input checked="" type="checkbox"/> คาร์ล ไลว์
<input checked="" type="checkbox"/> รามส์ ดับบลิว เบกาส
<input checked="" type="checkbox"/> อำนาจชัย ปฏิบัติกรเผ่างค์
<input checked="" type="checkbox"/> สุวิทย์ วิบุลผลประเสริฐ
<input checked="" type="checkbox"/> สิมพันธ์ อุดมรักษ์
<input checked="" type="checkbox"/> ทฤษฎี แสงอุทัย
<input checked="" type="checkbox"/> นพวรรณ เข็มบริรัตน์
<input checked="" type="checkbox"/> วีรศักดิ์ ชิงถาวร
<input checked="" type="checkbox"/> กิตติ ภักดีวิชชนะกุล
<input checked="" type="checkbox"/> รุ่งโรจน์ โยชนก

12 รายการ

เพิ่ม แก้ไข ลบ

รูปที่ ข.19 หน้าจอหลักของข้อมูลผู้แต่ง

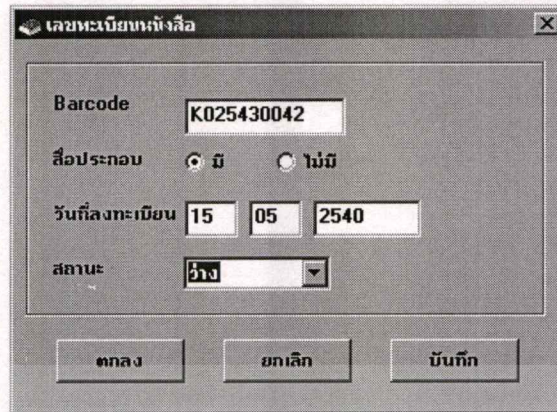
หัวเรื่อง
<input checked="" type="checkbox"/> จิตบำบัด -- สหรัฐอเมริกา
<input checked="" type="checkbox"/> ไรศ -- รวมเรื่อง
<input checked="" type="checkbox"/> สุขภาพ
<input checked="" type="checkbox"/> กฎหมาย
<input checked="" type="checkbox"/> ไทย -- กฎหมายและระเบียบข้อบังคับ
<input checked="" type="checkbox"/> ชาว (ภาษาคอมพิวเตอร์)
<input checked="" type="checkbox"/> การเขียนโปรแกรม (คอมพิวเตอร์)
<input checked="" type="checkbox"/> การเขียนโปรแกรมเชิงวัตถุ

8 รายการ

เพิ่ม แก้ไข ลบ

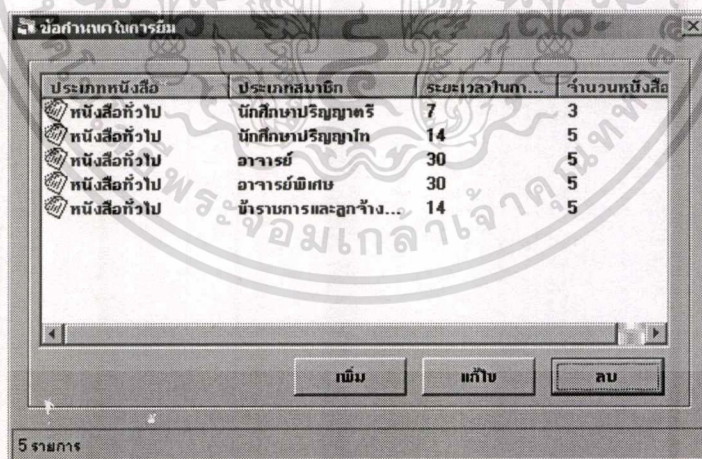
รูปที่ ข.20 หน้าจอหลักของข้อมูลหัวเรื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.21 หน้าจอข้อมูลเลขทะเบียนหนังสือ

2.3) เมฆนโยบายห้องสมุด การทำงานในส่วนของเมนูนี้จะเป็นการจัดการข้อมูลในส่วนที่เป็นข้อกำหนดต่างๆของห้องสมุดเช่น ประเภทสมาชิก ประเภทหนังสือ และข้อกำหนดในการยืมซึ่งข้อมูลเหล่านี้จะเป็นเมนูย่อยในส่วนของเมนูนี้ เมื่อทำการเลือกเมนูย่อยเหล่านั้น ระบบจะแสดงหน้าจอหลักในการจัดการข้อมูลเหล่านี้โดยในส่วนของแต่ละประเภทสมาชิกและประเภทหนังสือนั้นได้แสดงไปแล้วในรูปที่ ข.11 และ ข.17 ตามลำดับ ส่วนหน้าจอข้อกำหนดในการยืมจะแสดงในรูปที่ ข.22



ประเภทหนังสือ	ประเภทสมาชิก	ระยะเวลาในท...	จำนวนหนังสือ
หนังสือทั่วไป	นักศึกษาปริญญาตรี	7	3
หนังสือทั่วไป	นักศึกษาปริญญาโท	14	5
หนังสือทั่วไป	อาจารย์	30	5
หนังสือทั่วไป	อาจารย์พิเศษ	30	5
หนังสือทั่วไป	ข้าราชการและลูกจ้าง...	14	5

รูปที่ ข.22 หน้าจอหลักของข้อมูลข้อกำหนดในการยืม

3. ส่วนการให้บริการ เมนูส่วนนี้จะเกี่ยวกับการให้บริการต่างซึ่งเป็นหน้าที่หลักของระบบ เช่น การยืม-คืนหนังสือ การจองหนังสือเป็นต้น โดยจะประกอบด้วยเมนูย่อยๆ ได้แก่

3.1) เมนูยืม-คืนหนังสือ การทำงานในส่วนของเมนูนี้จะเป็นการให้บริการยืม คืนหนังสือแก่สมาชิกโดยเมื่อผู้ใช้ทำการเลือกเมนูนี้จะปรากฏหน้าจอการยืม คืนหนังสือขึ้น ดังรูปที่ ข.23 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

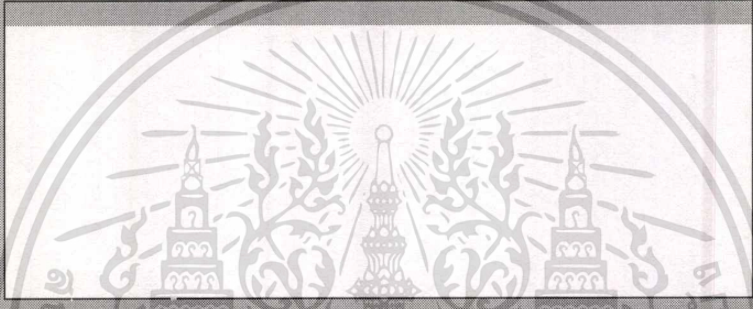
บริการยืม-คืนหนังสือ

เลขประจำตัวสมาชิก    บาร์โค้ด

รหัสสมาชิก	ค่าปรับค้างชำระ	บาท
ชื่อ-นามสกุล	หนังสือทั่วไป	เล่ม
คณะ	หนังสือสำรอง	เล่ม
ประเภทสมาชิก	รวม	เล่ม
วันหมดอายุ		

รายการ ยืม-คืนหนังสือ



รูปที่ ข.23 หน้าจอการยืม คืนหนังสือ

เมื่อผู้ใช้ป้อนเลขประจำตัวสมาชิกลงไปแล้วกด Enter ระบบจะทำการแสดงข้อมูลในการยืมหนังสือของสมาชิกคนนั้นและตรวจสอบสิทธิในการยืมของสมาชิก หากสมาชิกมีสิ่งที่ยึดต่อกฎเกณฑ์ใดๆที่ห้องสมุดกำหนดไว้ระบบจะไม่อนุญาตให้ทำการยืม ดังตัวอย่างในรูปที่ ข.24 ซึ่งสมาชิกรายนี้มีหนังสือที่เกินกำหนดคืนอยู่จึงไม่สามารถทำการยืมหนังสือได้ แต่หากสมาชิกไม่มีสิ่งที่ยึดต่อกฎเกณฑ์ใดๆที่ห้องสมุดกำหนดไว้ระบบจะอนุญาตให้ทำการยืม โดยผู้ใช้จะต้องทำการป้อนหมายเลขบาร์โค้ดเพื่อทำการบันทึกการยืมหนังสือเข้าสู่ระบบ โดยหลังจากที่ดำเนินการเสร็จสิ้นแล้วจะปรากฏรายการของหนังสือเล่มนั้นในรายการยืม-คืนหนังสือที่หน้าจอการยืม คืนหนังสือดังรูปที่ ข.25

บริการยืม-คืนหนังสือ

เลขประจำตัวสมาชิก บาร์โค้ด

401200153

ลบ

รหัสสมาชิก 401200153 ค่าปรับค้างชำระ 0 บาท

ชื่อ-นามสกุล นาย ชรรณจักร จันทระจิรโกวิท

คณะ คณะเศรษฐศาสตร์ หนังสือทั่วไป 2 เล่ม

ประเภทสมาชิก นักศึกษาปริญญาตรี

วันหมดอายุ 18/3/2545

LASClient

สมาชิกไม่สามารถทำการยืมหนังสือได้ เนื่องจากสมาชิกมีหนังสือที่เกินกำหนดคืน

OK

รายการ ยืม-คืนหนังสือ

บาร์โค้ด	ชื่อเรื่อง	วันที่ยืม	กำหนดคืน
B025300153	ความรู้เบื้องต้นเกี่ยวกับกฎหมาย = Introd...	22/1/2545	1/2/2545
K025430023	Fundamental of Java Programming	22/3/2545	1/3/2545
F025380129	ความรู้เบื้องต้นเกี่ยวกับกฎหมายทั่วไป	22/3/2545	1/3/2545

รูปที่ ข.24 หน้าจอการยืม-คืนหนังสือในกรณีที่สมาชิกไม่มีสิทธิทำการยืม

บริการยืม-คืนหนังสือ

เลขประจำตัวสมาชิก บาร์โค้ด

401400158

ลบ

รหัสสมาชิก 401400158 ค่าปรับค้างชำระ 0 บาท

ชื่อ-นามสกุล นางสาว ภคพร โมศิริ

คณะ คณะศิลปศาสตร์ หนังสือทั่วไป 1 เล่ม

ประเภทสมาชิก นักศึกษาปริญญาตรี หนังสือสำรอง 0 เล่ม

วันหมดอายุ 18/3/2545 รวม 1 เล่ม

รายการ ยืม-คืนหนังสือ

บาร์โค้ด	ชื่อเรื่อง	วันที่ยืม	กำหนดคืน
K025430042	กระบวนวิธีลวกจากา = Object-oriented ...	16/3/2545	23/3/2545

รูปที่ ข.25 หน้าจอการยืม-คืนหนังสือในกรณีที่สมาชิกมีสิทธิทำการยืม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2) เมนูจองหนังสือ การทำงานในส่วนของเมนูนี้จะเป็นการให้บริการจองหนังสือแก่สมาชิกโดยเมื่อผู้ใช้ทำการเลือกเมนูนี้จะปรากฏหน้าจอการจองหนังสือขึ้น ดังรูปที่ ข.26

รูปที่ ข.26 หน้าจอการจองหนังสือ

เมื่อผู้ใช้ป้อนหมายเลขบาร์โค้ดลงไปแล้วกด Enter ระบบจะทำการแสดงข้อมูลของหนังสือเล่มนั้นและทำการตรวจสอบว่าหนังสือเล่มนั้นถูกยืมไปจริงหรือไม่ และหากถูกยืมไปแล้วมีรายการจองหนังสือเล่มนี้โดยสมาชิกรายอื่นๆอยู่ในระบบหรือไม่ หากมีก็จะทำการแสดงรายชื่อของสมาชิกที่ทำการจองตามลำดับการจอง เมื่อผู้ใช้ทำการป้อนหมายเลขประจำตัวสมาชิกลงไปแล้วกด Enter ระบบจะทำการบันทึกรายการจองหนังสือเล่มนั้นไว้ในระบบและแสดงรายชื่อของสมาชิกรายการลำดับสมาชิกที่ทำการจองหนังสือ ดังแสดงในรูปที่ ข.27

บริการจองหนังสือ

บาร์โค้ด: k025430042      เลขประจำตัวสมาชิก: [ ]      ออก

หนังสือที่ต้องการยืม

บาร์โค้ด	ชื่อเรื่อง
K025430042	กระแทะเปลี่ยนภาษา = Object-oriented programming with Java Language

ลำดับสมาชิกที่ทำรายการหนังสือ

หมายเลขสมาชิก	ชื่อสมาชิก	วันที่ทำการจอง	วันกำหนดรับ
401400168	วิรัชพล สกุลโศภณินันท์	16/3/2545	
411050128	ศุภชัย สมพานิช	16/3/2545	


ยกเลิกการจอง

รูปที่ ข.27 หน้าจอการจองหนังสือในกรณีที่ทำการจองสำเร็จ

## ข.2 ระบบงานในส่วนที่เป็น Web Application

ในส่วนของ Web Application นี้ผู้ใช้งานจะสามารถเรียกใช้งานผ่านทางเว็บเพจของสำนักหอสมุดเพื่อทำการใช้งานส่วนของ Online Catalog ที่ใช้ในการค้นหาหนังสือที่ต้องการ และส่วนของ Check Circulation Record เพื่อทำการเรียกดูรายการยืมหนังสือของตนเองได้โดยสมาชิกต้องทำการป้อนเลขประจำตัวในการเรียกดูข้อมูลรายการยืมของตน สำหรับรายละเอียดของการใช้งานในแต่ละส่วนนั้นมีดังต่อไปนี้

1. ส่วน Online Catalog เป็น Application ส่วนที่เปิดให้บริการแบบสาธารณะสำหรับทุกคนที่สามารถเข้าถึงเครือข่ายอินเทอร์เน็ตได้ โดยงานส่วนนี้จะให้บริการค้นหาหนังสือที่มีอยู่ในห้องสมุดโดยใช้เงื่อนไขในการค้นหาต่างๆ ซึ่งเมื่อผู้ใช้งานเรียกดูเว็บเพจหน้าแรกของสำนักหอสมุดจะพบฟอร์มที่ใช้ในการค้นหาหนังสือทางด้านซ้ายของเว็บเพจดังแสดงในรูปที่ ข.28

 มหาวิทยาลัยศรีนครินทรวิโรฒ  
Krik University

ค้นหาคำที่ต้องการค้นหา  
ค้นหาจาก

ชื่อเรื่อง  
 หัวเรื่อง  
 ผู้แต่ง  
 เลขเรียกหนังสือ

Check Member Circulation

เลขประจำตัว

ติดต่อขอสงวนสิทธิ์ระบบสืบค้นสารสนเทศห้องสมุด มหาวิทยาลัยศรีนครินทรวิโรฒ  
หนังสือใหม่  
ขอสงวนสิทธิ์ในการค้นหาฐานข้อมูล โปรดติดต่อ lib@krik.ac.th

### รูปที่ ข.28 เว็บเพจหน้าแรกของสำนักหอสมุด

เมื่อผู้ใช้ทำการค้นหาหนังสือจากเงื่อนไขต่างๆที่มีให้ เช่น ค้นหาจากชื่อเรื่อง หัวเรื่อง ผู้แต่ง เลขเรียกหนังสือ จะปรากฏหน้าจอแสดงผลการค้นหาดังแสดงในรูปที่ ข.29 ถึง ข.32 ตามลำดับ

ชื่อเรื่อง

พบผลการค้นหาทั้งสิ้น 2 รายการ

ลำดับที่	Barcode	เลขเรียกหนังสือ	ชื่อหนังสือ	สถานะ
1	K025430042	QA76.73.J38 ๕636ก	กระบวนวิธีจาวา = Object-oriented programming with Java Language	ถูกยืม
2	K025430045	QA76.73.J38 ๕636ก	การเขียนโปรแกรมเชิงวัตถุภาษาจาวา = Object-oriented programming in java	ถูกยืม

| Home | Library Home |

### รูปที่ ข.29 เว็บเพจผลการค้นหาหนังสือจากชื่อเรื่อง

ชื่อเรื่อง

พบผลการค้นหาทั้งสิ้น 2 รายการ

ลำดับที่	Barcode	เลขเรียกหนังสือ	ชื่อหนังสือ	สถานะ
1	K025430042	QA76.73.J38 ๕636ก	กระบวนวิธีจาวา = Object-oriented programming with Java Language	ถูกยืม
2	K025430045	QA76.73.J38 ๕636ก	การเขียนโปรแกรมเชิงวัตถุภาษาจาวา = Object-oriented programming in java	ถูกยืม

| Home | Library Home |

### รูปที่ ข.30 เว็บเพจผลการค้นหาหนังสือจากหัวเรื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

ชื่อเรื่อง  ikit ค้นหาข้อมูล

---

พบผลการค้นหาลำดับที่ 1 รายการ

ลำดับที่	ชื่อผู้แต่ง
1	กิตติ กิตติวิริยะกุล

| Home | Library Home |

รูปที่ ข.31 เว็บเพจผลการค้นหาหนังสือจากชื่อผู้แต่ง

ชื่อเรื่อง  QA76.73 ค้นหาข้อมูล

---

พบผลการค้นหาลำดับที่ 5 รายการ

ลำดับที่	Barcode	เลขเรียกหนังสือ	ชื่อหนังสือ	สถานะ
1	K025420020	QA76.73.J38 ก672ง	Java ฉบับโปรแกรมเมอร์	ดูยืม
2	K025430023	QA76.73.J38 ว837น	Fundamental of Java Programming	ดูยืม
3	K025430024	QA76.73.J38 ว837น	Fundamental of Java Programming	ดูยืม
4	K025430042	QA76.73.J38 ข636ก	กระบวนทัศน์จาก = Object-oriented programming with Java Language	ดูยืม
5	K025430045	QA76.73.J38 ข638ก	ทฤษฎีโปรแกรมเชิงวัตถุภาษาจาวา = Object-oriented programming in java	ดูยืม

| Home | Library Home |

รูปที่ ข.32 เว็บเพจผลการค้นหาหนังสือจากเลขเรียกหนังสือ

2. ส่วนการตรวจสอบรายการยืมหนังสือของสมาชิก เป็น Application ส่วนที่ให้บริการกับสมาชิก ห้องสมุดที่ต้องการตรวจสอบรายการหนังสือที่ตนทำการยืม โดยสำหรับฟอร์มที่ใช้ในป้อนหมายเลขสมาชิกเพื่อทำการค้นหารายการหนังสือที่ทำการยืมนั้นจะอยู่ด้านล่างของแบบฟอร์มที่ใช้ในการค้นหาหนังสือในห้องสมุดที่หน้าแรกของเว็บเพจสำนักหอสมุดดังแสดงในรูปที่ ข.28 เมื่อสมาชิกทำการ Submit หมายเลขประจำตัวของตน ระบบจะทำการแสดงรายการหนังสือที่สมาชิกรายนั้นทำการยืมจนถึงปัจจุบัน โดยจะแสดงเฉพาะรายการที่สมาชิกยังไม่ได้ทำการคืนเท่านั้น ดังแสดงในรูปที่ ข.33

## Member Circulation Check

## รายละเอียดรายการยืมหนังสือของสมาชิก

เลขประจำตัว 401200153  
ชื่อ นายธรรมพิภพ หันกรธโรกรักษ์  
ประเภทสมาชิก นักศึกษามหาวิทยาลัยสุโขทัย  
คณะ คณะเศรษฐศาสตร์

## รายการยืม

บาร์โค้ด	ชื่อเรื่อง	วันที่เริ่ม	การหมดเกิน
B025300153	ความรู้เบื้องต้นเกี่ยวกับกฎหมาย = Introduction to Law	2002-01-22T00:00:00	2002-02-01T00:00:00
F025360129	ความรู้เบื้องต้นเกี่ยวกับกฎหมายมหาชน	2002-03-22T00:00:00	2002-03-01T00:00:00
K025430023	Fundamental of Java Programming	2002-03-22T00:00:00	2002-03-01T00:00:00

| home | library home | |

## รูปที่ ข.33 เว็บไซต์แสดงผลตรวจสอบรายการยืมหนังสือของสมาชิก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อผู้เขียน	นาย ไพศาล กาญจนสุทธิรักษ์
วันเดือนปีเกิด	11 มิถุนายน 2518
สถานที่เกิด	จังหวัดกาญจนบุรี
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตรบัณฑิต (วิศวกรรมเครื่องกล)
สถาบันที่สำเร็จการศึกษา	คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2541



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้