

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจธ.

การพัฒนาห้องสมุดดิจิทัลโดยใช้ UML

Implementation of Digital Library using UML



วัน เดือน ปี.....	15 ส.ค. 2550
เลขทะเบียน.....	01891
เลขเรียกหนังสือ.....	๗๗ ปี ๗๘๖๓ 2543
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจธ."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2543
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

H001891

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาห้องสมุดดิจิทัลโดยใช้ UML
นักศึกษา	นางสาวเบญจรงค์ นิตยพัฒน์
อาจารย์ที่ปรึกษา	รศ. ดร. วิเชียร เปรมชัยสวัสดิ์ และ ดร. ประจวบ วานิชชัชวาล
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

โครงการนี้ได้จัดทำขึ้นโดยการขีดยกยืมแนวคิดของห้องสมุดดิจิทัล กล่าวคือพัฒนาระบบงานจัดเก็บและรวบรวมเอกสารชนิดต่างๆ ซึ่งผลิตและเผยแพร่ โดยกรมวิทยาศาสตร์บริการ (Department of Science Service หรือใช้ตัวย่อว่า DSS) โดยจะเรียกโครงการนี้ว่า “DSS Digital Library Project” (DSSDL) ซึ่งจะสามารทำให้บริการเนื้อหาเต็มของเอกสารในลักษณะเพิ่มข้อมูลอิเล็กทรอนิกส์สำหรับผู้สนใจ และให้บริการค้นคืนเอกสาร ผ่านทางระบบเครือข่ายอินเทอร์เน็ต อันเป็นการขยายขีดความสามารถในการให้บริการข้อมูล ของหน่วยงานให้กว้างขวางและตรงใจผู้ใช้บริการมากยิ่งขึ้น

ด้วยปัจจุบันแนวคิดเชิงวัตถุได้เข้ามามีบทบาทในการพัฒนาระบบงานสารสนเทศมากขึ้น โดยจะสังเกตได้จากเครื่องมือในเชิงวัตถุ ปัจจุบันกำลังได้รับความนิยมมากขึ้นเรื่อยๆ ได้แก่ ภาษาเชิงวัตถุ ฐานข้อมูลเชิงวัตถุสัมพันธ์ ดังนั้นเมื่อเล็งเห็นคุณประโยชน์และความก้าวไกลของแนวคิดนี้ ในอนาคต จึงได้ริเริ่มความคิดที่จะพัฒนาระบบงานห้องสมุดดิจิทัล ด้วยการนำเอา UML (Unified Modeling Language) ซึ่งสนับสนุนแนวคิดเชิงวัตถุมาใช้ในขั้นตอนการวิเคราะห์และออกแบบระบบงาน

Title Implementation of Digital Library using UML
Student Miss Benjarong Nittayapat
Advisor Assoc. Prof. Wichian Premchaiswadi, Ph.D.
Prachuab Vanitchatchavan, Ph. D.
Level of Study Master of Science in Information Technology
Major Information Science
Academic Year 2000

ABSTRACT

Implementation of Digital Library using UML develop the library system to collect and manage documents that is provided by Department of Science Service (DSS) called "DSS Digital Library" (DSSDL). The full text paper in electronic file format is served to customer. The DSSDL project also provides retrieval system through Internet. We hope that it increases the performance of organization in wide area as well as covers more customer requirements. At the present time, Object-oriented paradigm (OO) plays an important role in the information technology. It is very flavor and is widely used, for example, Structural programming language is replaced by Object-oriented programming language as well as the traditional Relational Database Management System (RDBMS) is replaced by Object-relational Database Management System (ORDBMS). Thus the change in the recent technology makes us use UML (Unified Modeling Language) that support the Object-oriented analysis and design (OOAD) process.

กิตติกรรมประกาศ

ในการจัดทำโครงการพัฒนาห้องสมุดดิจิทัลด้วย UML นี้ ได้รับความสนับสนุนเป็นอย่างดี จากหลายฝ่าย ที่คอยให้คำแนะนำปรึกษา และเสียสละเวลาอันมีค่า จนทำให้การศึกษาโครงการนี้ บรรลุผลตามเป้าหมายที่วางไว้ ผู้จัดทำจึงใคร่ขอขอบพระคุณ

1. บิดา, มารดา และน้องสาว ที่คอยให้ความช่วยเหลือ และเป็นกำลังใจในการทำงาน
2. รศ. ดร. วิเชียร เปรมชัยสวัสดิ์ อาจารย์ที่ปรึกษาโครงการที่ได้ให้ความรู้ คำปรึกษา ในการจัดทำโครงการ
3. ดร. ประจวบ วาณิชชวล อาจารย์ที่ปรึกษาร่วม ที่ได้ให้คำแนะนำ และความช่วยเหลือ ต่างๆ ด้วยดีตลอดมา
4. รศ. ดร. เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษาระหว่างการเรียนวิชา สัมมนา 1 และ สัมมนา 2 ซึ่งให้คำแนะนำ และแนวคิดสำหรับหัวข้อโครงการฯ
5. อ. รุ่งโรจน์ โพนคำ ที่ได้ให้คำแนะนำ ในการแก้ไขปัญหาต่างๆ ของการเขียน โปรแกรม
6. อาจารย์ทุกๆ ท่าน ที่สั่งสอนให้ข้าพเจ้าคิดเป็น และปฏิบัติเป็น รวมทั้ง ตัดสินใจในการ แก้ปัญหาต่างๆ จากความรู้พื้นฐานที่ได้ศึกษามา
7. คุณสัณห์พล ธาดาประดิษฐ์ (AAJC Thailand specialist) ที่คอยให้คำปรึกษาทางด้าน เทคนิค และให้ความรู้ เรื่อง Java, Servlet API โดยคุณผ่านทางระบบเครือข่าย ตลอด ระยะเวลาในการทำงาน รวมทั้ง เป็นกำลังใจ และเสนอแนวคิดในการแก้ปัญหาต่างๆ
8. คุณเอกสิทธิ์ เถกิงศักดิ์กุล (Java programmer) ที่ช่วยแก้ปัญหาทางด้านเทคนิคการ เขียนโปรแกรม เสนอความคิดเห็น และตอบคำถามต่างๆ ที่เป็นประโยชน์ในการทำงาน
9. เพื่อนๆ IS6 ที่เป็นกำลังใจ และฝ่าฟันอุปสรรคไปพร้อมกัน รวมทั้ง webboard ของรุ่น ที่เป็นสื่อกลางในการแลกเปลี่ยนความรู้ และระบายความเครียดจากการทำงาน
10. พี่ๆ บรรณารักษ์ และเพื่อนร่วมงาน กรมวิทยาศาสตร์บริการ ที่ให้ข้อมูล คำแนะนำ และข้อคิดเห็นเกี่ยวกับโครงการฯ
11. เครื่องคอมพิวเตอร์ ซอฟต์แวร์ และระบบเครือข่าย Internet ของสถาบันฯ และกรม วิทยาศาสตร์บริการ ที่ใช้เป็นเครื่องมือในการทำงาน

เบญจรงค์ นิตยพัฒน์

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VI
สารบัญตาราง.....	VII
บทที่	
1. บทนำ	
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	3
1.4 เนื้อหาในโครงการ.....	3
2. ห้องสมุดดิจิทัล	
2.1 หลักการของห้องสมุดดิจิทัล.....	4
2.2 คุณสมบัติที่สำคัญ.....	4
2.3 ส่วนประกอบ.....	5
3. Object-oriented Methodology	
3.1 Object-oriented คืออะไร.....	6
3.2 คำศัพท์และความหมาย.....	6
3.3 คุณสมบัติที่สำคัญของ Object-oriented.....	9
3.4 Object-oriented Methodology.....	10
4. UML และ Waterfall Process Model	
4.1 UML คืออะไร.....	12
4.2 ประวัติศาสตร์ของ UML.....	12
4.3 ประเภทของ UML Diagram.....	14
4.4 ความหมายของแต่ละ Diagram.....	15
4.5 Waterfall Process Model.....	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5. Object-oriented Case Tool: Rational Rose	
5.1 Rational Rose 2000 Enterprise Edition.....	24
5.2 Rational Rose J.....	24
5.3 Rational Rose Oracle8.....	30
6. DSS Digital Library System Modeling	
6.1 Use Case Diagram.....	35
6.2 Component Diagram.....	35
6.3 Class Diagram.....	38
6.4 Sequence Diagram.....	50
7. สถาปัตยกรรมระบบและเครื่องมือที่ใช้พัฒนาโปรแกรม	
7.1 สถาปัตยกรรมของระบบ.....	56
7.2 เครื่องมือที่ใช้พัฒนาโปรแกรม.....	56
8. บทสรุปและแนวทางการพัฒนา.....	57
บรรณานุกรม.....	58
ภาคผนวก.....	60
ประวัติผู้เขียน.....	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการพัฒนาระบบงาน

โครงการพัฒนาระบบห้องสมุดดิจิทัลโดยใช้ UML เป็นโครงการพัฒนาระบบงานสารสนเทศ ของกองสนเทศวิทยาศาสตร์และเทคโนโลยี (Scientific and Information Division โดยมีตัวย่อว่า “SI”) ซึ่งเป็นหน่วยงานภายใต้กรมวิทยาศาสตร์บริการ (Department of Science Service ใช้ตัวย่อว่า “DSS”) มีหน้าที่ เผยแพร่ผลงาน และความรู้ทางด้านวิทยาศาสตร์และเทคโนโลยี ของกรมวิทยาศาสตร์บริการ ได้แก่

1. วารสารกรมวิทยาศาสตร์บริการ (DSS Journal) อยู่ในรูปของตัวเล่มเท่านั้น
2. รายงานกิจกรรมกรมวิทยาศาสตร์บริการ (DSS Report) อยู่ในรูปของตัวเล่มเท่านั้น
3. บทความเผยแพร่ทางวิทยุ และสื่อต่างๆ (DSS Topic) อยู่ในรูปของเอกสารเท่านั้น
4. เอกสารงานวิจัย (DSS Research Paper) อยู่ในรูปของเอกสารเท่านั้น
5. ข่าวสารกรมวิทยาศาสตร์บริการ (DSS News) อยู่ในรูปของ WWW Page และตัวเล่ม
6. สารความรู้ทางด้านวิทยาศาสตร์ และเทคโนโลยี (DSS Knowledge) มีอยู่ในรูปแบบ WWW Page เท่านั้น และเผยแพร่ผ่าน WWW Site ของกรมวิทยาศาสตร์บริการ (<http://www.dss.go.th>)
7. บัญชีรายชื่อเอกสารใหม่ประจำเดือน (SI New Resource List) มีอยู่ทั้งในรูปตัวเล่ม และ WWW Page อยู่ภายใน WWW Site กองสนเทศวิทยาศาสตร์และเทคโนโลยี (<http://www.dss.go.th/siwebth/index.htm>)
8. หน้าสารบัญของวารสารที่บอกรับ (SI Table of Content) อยู่ในรูปของ WWW Page และรูปของเอกสาร
9. เอกสารอื่นๆ ที่ผลิตขึ้น และเป็นเจ้าของโดย กรมวิทยาศาสตร์บริการ

ในปัจจุบันเอกสารเหล่านี้ ถูกเผยแพร่แก่ประชาชนทั่วไป ในลักษณะรูปเล่มที่เป็นกระดาษ ซึ่งมีข้อจำกัดในการให้บริการ อาทิ อายุการใช้งานของเอกสารที่ค่อนข้างต่ำ หรือ มีการจัดเก็บที่กระจัดกระจาย ทำให้เกิดความยากในการเข้าถึงเอกสาร จึงได้เกิดความต้องการจาก นักวิทยาศาสตร์ ผู้ผลิตผลงานและข้อมูลความรู้ รวมทั้งบรรณารักษ์ ผู้มีหน้าที่ให้บริการ ที่ต้องการจะเพิ่มขีดความ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถในการให้บริการข้อมูล โดยมีความต้องการให้มีการบริการเอกสารฉบับเต็มดังกล่าวในรูปแบบของแฟ้มอิเล็กทรอนิกส์ ผ่านระบบเครือข่ายอินเทอร์เน็ต ชนิด World Wide Web ซึ่งกำลังเป็นสื่อที่ใช้กันอย่างแพร่หลาย และเป็นที่ยอมรับสำหรับการค้นคว้าข้อมูลในขณะนี้ โดยผู้ใช้บริการไม่ต้องเดินทางมาใช้บริการตัวเล่ม เหมือนแต่ก่อน จึงเป็นจุดกำเนิดของการพัฒนาโครงการนี้ขึ้น

สำหรับแนวคิดที่จะใช้ UML ในการพัฒนาระบบงาน ก็เพราะเหตุผลที่ว่า ในโลกของความเป็นจริง สิ่งต่างๆ ที่เรามองเห็นล้วนแล้วก็คือวัตถุ ทำให้คนเราค้นเคยกกับการมองสิ่งต่างๆ ที่เห็นเป็นวัตถุ ฉะนั้นเมื่อมีการนำแนวคิดนี้มาประยุกต์ใช้กับการพัฒนาระบบงานสารสนเทศ ทำให้ประมาณปี ค.ศ. 1960 จึงเกิดมีผู้นำเสนอแนวคิดเชิงวัตถุขึ้น เพื่อใช้ในการวิศวกรรมซอฟต์แวร์ (System Engineering) เนื่องด้วยเทคโนโลยีเชิงวัตถุ (Object Technology) มีจุดเด่นที่สำคัญคือ การนำกลับมาใช้ใหม่ (Reusability) ของส่วนของโปรแกรม (Program Components) ทำให้การพัฒนาระบบงานสำเร็จได้อย่างรวดเร็ว และมีคุณภาพที่สูงขึ้น โดยเฉพาะการพัฒนาซอฟต์แวร์ในเชิงวัตถุ มีความง่ายในการบำรุงรักษา (Maintenance) เนื่องจากมีคุณสมบัติการสืบทอด (Inheritance) [14]

หลายปีมาแล้วที่คำว่า “Object-oriented” (OO) ถูกนำมาใช้ในการพัฒนาระบบงาน ได้แก่ ภาษาในเชิงวัตถุ (Object-oriented Programming Languages) ตัวอย่างเช่น ภาษา C++, Smalltalk และ Java นอกจากนั้นแล้ว การวิศวกรรมซอฟต์แวร์ (System Development Process) ก็ได้นำเอาแนวคิดนี้มาใช้ด้วยเช่นกัน โดยจะเห็นว่าเกิดคำศัพท์ต่างๆ อาทิ Object-oriented Programming (OOP), Object-oriented Design (OOD), Object-oriented Analysis (OOA), Object-oriented Database Management Systems (OODBMS) และ Object-oriented Computer Aided Software Engineering (OOCASE) ซึ่งล้วนแต่เป็นการนำแนวคิดเชิงวัตถุมาใช้ในการพัฒนาระบบทั้งสิ้น [14]

ในปัจจุบันการพัฒนาระบบงานสารสนเทศมีแนวโน้มจะใช้แนวคิดในเชิงวัตถุมากขึ้น โดยสังเกตได้จากเทคโนโลยีในเชิงวัตถุที่ได้รับความนิยมสูงขึ้นเรื่อยๆ ซึ่งมีสาเหตุมาจากข้อดีของแนวคิดเชิงวัตถุที่ได้กล่าวแล้วข้างต้น ดังนั้นเมื่อเล็งเห็นคุณประโยชน์และความก้าวไกลของแนวคิดนี้ในอนาคต จึงได้ริเริ่มความคิดที่จะพัฒนาระบบงานห้องสมุดดิจิทัล โดยการใช้ UML (Unified Modeling Language) เป็นเครื่องมือสำหรับการจำลองระบบงาน โดยสนับสนุนการวิเคราะห์ และออกแบบระบบเชิงวัตถุ ซึ่งจะกล่าวรายละเอียดของ UML ในบทที่ 4

1.2 วัตถุประสงค์ของโครงการพัฒนาระบบงาน

1. ศึกษาหลักการของห้องสมุดดิจิทัล
2. ศึกษาในหัวข้อ การวิเคราะห์และออกแบบระบบงานด้วย UML
3. ศึกษาการผสมผสานการพัฒนาระบบงาน ด้วยแนวคิดเชิงวัตถุ (Object-oriented Application) กับฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ศึกษาวิธีการเทียบเคียง (Mapping) ระหว่างการออกแบบฐานข้อมูลโดยใช้แนวคิดเชิงวัตถุด้วย UML กับฐานข้อมูลเชิงสัมพันธ์
5. สามารถออกแบบฐานข้อมูล และพัฒนาฐานข้อมูลเชิงสัมพันธ์ โดยใช้ CASE Tool
6. สามารถพัฒนาโปรแกรมสำหรับระบบงาน โดยใช้ภาษาเชิงวัตถุ ด้วยแนวคิดการเขียนโปรแกรมในเชิงวัตถุ

1.3 ขอบเขตของโครงการพัฒนาระบบงาน

1. วิเคราะห์และออกแบบระบบงาน โดยใช้แนวคิดเชิงวัตถุ (สร้าง UML Diagram ต่างๆ โดยการใช้โปรแกรม Rational Rose 2000 Enterprise Edition)
2. พัฒนาระบบงานด้วยการ Mapping จากแบบจำลอง (UML Diagram) ไปสู่โปรแกรมในลักษณะ WWW-based Application ด้วยภาษา Java โดยเลือกใช้ Java API (Java Application Programming Interface) ที่เหมาะสม
3. ออกแบบ และสร้างฐานข้อมูลเชิงสัมพันธ์ โดยการสร้าง Database Schema ด้วยการใช้ CASE Tool (Rational Rose 2000 Edition Oracle8 Add-in)

1.4 เนื้อหาในโครงการพัฒนาระบบงาน

เนื้อหาสำหรับโครงการพัฒนาระบบงานในบทแรกนี้จะกล่าวถึงความเป็นมา ความสำคัญ และวัตถุประสงค์ในการทำโครงการพัฒนาระบบงาน บทที่ 2 จะกล่าวถึงหลักการของห้องสมุดดิจิทัล ได้แก่ ความหมายและหลักการ คุณสมบัติ และส่วนประกอบของห้องสมุดดิจิทัล บทที่ 3 กล่าวถึงแนวคิดเชิงวัตถุ ส่วนประกอบที่สำคัญของ Object-oriented Methodology บทที่ 4 อธิบายถึงภาษาหรือแบบจำลองที่ใช้ ได้แก่ UML รวมทั้ง การใช้งาน UML ในขั้นต่างๆ ของการวิเคราะห์และออกแบบระบบด้วย Waterfall Process Model บทที่ 5 แนะนำและกล่าวถึงคุณสมบัติ พร้อมตัวอย่างวิธีการใช้งาน Rational Rose 2000 Enterprise Edition ในการสร้าง UML Diagram และอธิบายคุณสมบัติ และตัวอย่างการทำ Forward Engineering และการทำ Reverse Engineering ของโปรแกรมภาษา Java และ UML ด้วย Rational Rose J รวมทั้งการสร้าง Database Schema ด้วย Rational Rose Oracle8 วิธีการ Mapping จาก Class Diagram ไปเป็น Relational Database Schema บทที่ 6 จะกล่าวถึงแบบจำลองห้องสมุดดิจิทัลกรมวิทยาศาสตร์บริการ (DSS Digital Library) ที่ได้จากการวิเคราะห์ และออกแบบด้วย UML บทที่ 8 สถาปัตยกรรมของระบบ และเครื่องมือที่ใช้ในการพัฒนาระบบงาน (Java, Java Servlet API, JDBC, Oracle8) บทที่ 9 บทสรุปของโครงการพัฒนาระบบงาน และภาคผนวก ได้แก่ ตัวอย่างการจัดเก็บและสืบค้นข้อมูลของห้องสมุดดิจิทัลที่พัฒนาขึ้น และบางส่วนของ source program

บทที่ 2

ห้องสมุดดิจิทัล

2.1 หลักการของห้องสมุดดิจิทัล

“ห้องสมุดดิจิทัล” (Digital Library) บางครั้งอาจจะใช้คำว่า “ห้องสมุดเสมือน” (Virtual Library) หรือ “ห้องสมุดอิเล็กทรอนิกส์” (Electronic Library) และอื่นๆ แต่ทั้งหมดนี้ถือได้ว่าเป็นสิ่งเดียวกัน [1]

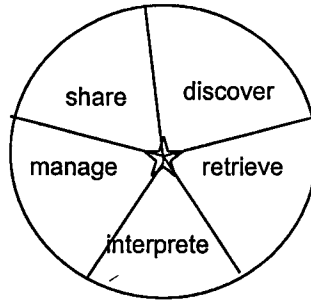
“ในปัจจุบันนี้ ยังไม่มีคำจำกัดความที่แน่ชัดของห้องสมุดดิจิทัล แต่นักห้องสมุด หลายท่าน ได้ให้คำจำกัดความไว้ อาทิ ห้องสมุดดิจิทัล ไม่ใช่สิ่งใดสิ่งหนึ่ง จำเป็นต้องใช้เทคโนโลยี หลายๆ ส่วนเพื่อการเชื่อมโยงทรัพยากรที่มีอยู่ เป็นการรวบรวมเอกสารข้อมูลโดยไม่ต้องมีรูปเล่มอยู่ด้วย นอกจากนี้เป้าหมาย คือการเชื่อมโยงข้อมูลจากทั่วมุมโลก” [1]

2.2 คุณสมบัติที่สำคัญ

ห้องสมุดดิจิทัลไม่เพียงจะต้องสามารถสืบค้นผ่านระบบเครือข่ายได้เท่านั้น แต่ควรมีคุณสมบัติที่สำคัญ 5 ข้อ [18] ดังนี้ (ภาพประกอบในรูปที่ 2-1)

1. สามารถค้นหา (Discover) สารนิเทศพบ นั่นคือสามารถช่วยให้ผู้ใช้ค้นหาและเลือกสารนิเทศจากแหล่งต่างๆ เช่น ฐานข้อมูล หรือแหล่งอ้างอิงที่เหมาะสม
2. สามารถค้นคืน (Retrieve) สารนิเทศที่ต้องการได้ โดยห้องสมุดดิจิทัลจะต้องช่วยให้ผู้ใช้ค้นคืนตัวเอกสาร ทั้งชนิดเป็นตัวอักษร และไม่ใช่อักษร
3. สามารถตีความ (Interpret) สารนิเทศที่ค้นคืนมาได้ โดยควรมีสิ่งที่จะช่วยในการตีความสารนิเทศได้ เช่น คำบรรยาย หรือ คำจำกัดความ หรือ ประเภทของเอกสาร
4. สามารถจัดการ (Manage) โดยควรมีที่เก็บข้อมูลที่ค้นคืนได้ ในระยะสั้นและระยะยาว และจัดการกับข้อมูลเหล่านั้น
5. สามารถแลกเปลี่ยน (Share) ห้องสมุดดิจิทัล ควรเก็บข้อมูลในรูปแบบที่แตกต่างกันได้ และสามารถใช้อินเทอร์เน็ตร่วมกันตามวัตถุประสงค์

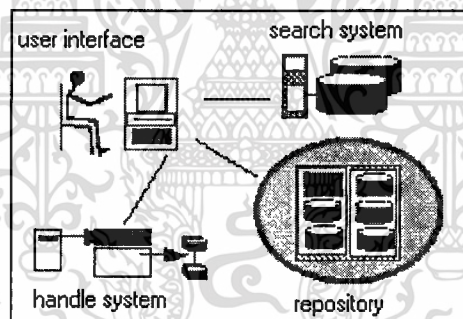
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 คุณสมบัติสำคัญของห้องสมุดดิจิทัล [18]

2.3 ส่วนประกอบของห้องสมุดดิจิทัล

ตามหลักการของห้องสมุดดิจิทัลทำให้สามารถแบ่งองค์ประกอบที่สำคัญ 4 ส่วน ได้แก่ ส่วนติดต่อกับผู้ใช้ (User Interface) ระบบค้นหา (Search System) ระบบควบคุมจัดการ (Handle System) และ รีพอสิตอรี (Repository) [6] ดังแสดงในรูปที่ 2-2



รูปที่ 2-2 ส่วนประกอบสำคัญของห้องสมุดดิจิทัล [6]

โดยแต่ละส่วนมีหน้าที่ดังนี้

1. ส่วนติดต่อกับผู้ใช้ (User Interface) มีหน้าที่สื่อสารกับผู้ใช้ โดยผู้ใช้สามารถเลือกต้องการ โดยส่ง Query String มา จากนั้นระบบแปลงให้อยู่ในรูปแบบที่สามารถใช้ได้
2. ระบบค้นหา (Search System) มีหน้าที่ช่วยให้ผู้ใช้ระบุได้ถึง Handle ซึ่งหมายถึง Unique Identifier ที่ชี้ไปยัง Digital Object (วัตถุที่เก็บในรีพอสิตอรี)
3. ระบบควบคุมจัดการ (Handle System) มีหน้าที่แสดงที่อยู่ของ Object หลังจากที่ได้รับ Handle มาจากระบบค้นหา
4. รีพอสิตอรี (Repository) มีหน้าที่จัดเก็บ Digital Object ชนิดต่างๆ โดยจัดกลุ่มเป็น

หมวดหมู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Object-oriented Methodology

3.1 Object-oriented คืออะไร

“Object-oriented” เป็นแนวคิด หรือ ระเบียบวิธีคิด (Paradigm) ของการสร้างหรือพัฒนาระบบงานหนึ่งๆ ดังคำพูดที่ว่า “OO is a technique for system modeling” [17] โดยจะมองระบบเป็นกลุ่มของวัตถุ (Object) ที่มีปฏิริยาต่อกัน หรือ “a number of object that interact” [17] โดยการรวมเอาข้อมูล (Information) และฟังก์ชันการทำงาน (Operation) เข้าไว้ด้วยกันในวัตถุ และกำหนด “วิธีการติดต่อกันระหว่างวัตถุ” (Interface) โดยถ้าวัตถุหนึ่งต้องการจะติดต่อกับอีกวัตถุหนึ่ง จะต้องติดต่อผ่าน Interface ที่กำหนดไว้เท่านั้น

จากหลักการนี้จะพบว่าการเปลี่ยนแปลงแก้ไข หน้าที่การทำงาน และข้อมูลใดๆ ในวัตถุหนึ่ง จะมีผลกระทบต่อวัตถุอื่นน้อยมาก ซึ่งจะทำให้การแก้ไขโปรแกรมทำได้สะดวกและรวดเร็วยิ่งขึ้น เป็นการลดต้นทุนในการบำรุงรักษาโปรแกรม นอกจากนี้ยังมีความสามารถในการสืบทอดคุณสมบัติ (Inheritance) จากวัตถุหนึ่งไปอีกวัตถุหนึ่งได้ ทำให้สามารถนำซอฟต์แวร์บางส่วนที่มีอยู่เดิมกลับมาใช้ใหม่ได้ (Reusability) โดยผู้พัฒนาไม่จำเป็นต้อง ทราบว่าซอฟต์แวร์บางส่วนที่น่ากลับมาใช้นั้นเขียนขึ้นมาอย่างไร (How does it work?) เพียงแต่ทราบว่าซอฟต์แวร์ส่วนนั้นทำอะไร (What does it work?) และทำการเพิ่มคุณสมบัติอื่นๆ ที่ผู้พัฒนา ต้องการเข้าไป ทำให้การพัฒนาซอฟต์แวร์ใหม่ๆ ทำได้รวดเร็วขึ้น [8]

3.2 คำศัพท์ และความหมาย

- **Object** คือ วัตถุ หรือสิ่งที่ประกอบด้วย สถานะ (State) หรือข้อมูล (Information) และฟังก์ชันการทำงาน (Operation) หรือพฤติกรรม (Behavior) [17]

An Object is characterized by
a number of operations
and a state which remembers the effect of these operations

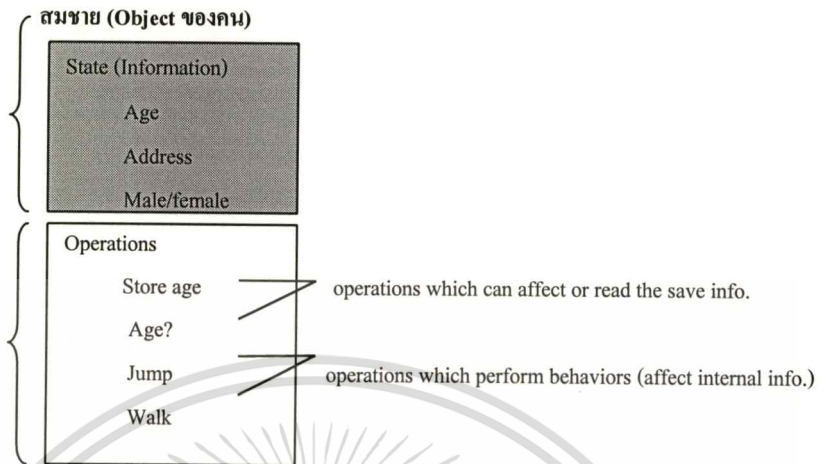
รูปที่ 3-1 ฟังแสดงคำนิยามของ Object [17]

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง

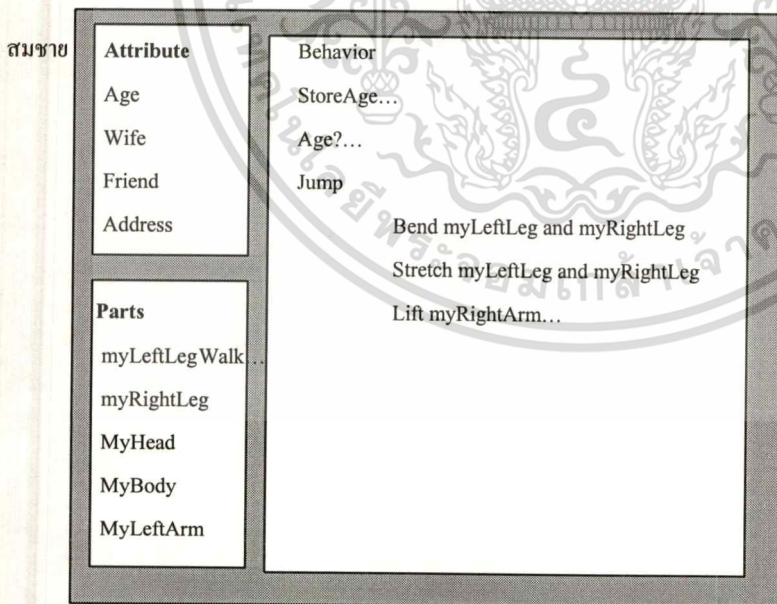
This part is **hidden** to us.

The only part of the object that we can see is **its operation**.



รูปที่ 3-2 ตัวอย่าง Object ของคนที่ชื่อ “สมชาย” [17]

ดังนั้นเมื่อเรามองเข้าไปใน Object จะประกอบด้วย Attribute, Part และ Behavior ดังรูป

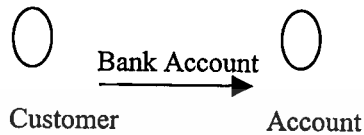


รูปที่ 3-2 ตัวอย่างส่วนประกอบภายใน Object ของคนที่ชื่อสมชาย [17]

- **Message passing**

ความเกี่ยวข้องกันระหว่าง Object แบ่งเป็น 2 ลักษณะคือ

1. Static Relation เช่น ลูกค้ายเป็นเจ้าของบัญชี



รูปที่ 3-3 ตัวอย่าง Static Relation ของ Object

2. Dynamic Relation เช่น ผู้ฝากเงินฝากเงินเข้าไปในบัญชี



รูปที่ 3-4 ตัวอย่าง Dynamic Relation ของ Object

Dynamic Relation โดยความหมายของ OO จะหมายถึง Object หนึ่งจะกระตุ้น (Stimuli) ไปยังอีก Object หนึ่ง ส่วนในทางการเขียนโปรแกรมจะเรียกเป็นการส่งข้อความ (Message Passing)

ดังตัวอย่าง หากต้องการให้สมชาย Jump ก็ต้องส่ง Stimulus ไป เมื่อสมชายได้รับก็จะกระทำต่างๆ ได้แก่ ส่ง Stimulus ไปยัง Arm และ Leg

- **Class และ Instance**

Class คือ เซตของ Object

Instance เป็น Object ที่สร้างจาก Class

ตัวอย่าง Class ของคน ประกอบด้วย คนแต่ละคน โดยมี สมชายเป็น Instance ของ Class คน หรือ คนชื่อ “ประดิษฐ์” เป็น Instance ของ Class คน เช่นกัน โดยแต่ละ Instance จะมีความแตกต่างกันระหว่าง Object อื่นๆ (มีเอกลักษณ์ หรือ Identity)

3.3 คุณสมบัติที่สำคัญของ Object-oriented

- **Encapsulation และ Information Hiding**

ข้อมูล (Information) และ หน้าที่การทำงาน หรือ พฤติกรรม (Operation/Behavior) จะถูกจับมัดรวมเข้าด้วยกัน (Encapsulation) ซึ่งจะสนับสนุนให้เกิด “การซ่อนข้อมูล” (Information Hiding) และสิ่งที่เราจะเห็นได้ของ Object คือ Instance ของมัน

ทำให้การเข้าถึงข้อมูลจะกระทำโดยเรียกใช้ Operation ของ Object ขึ้นมาทำงานดังกล่าวที่ว่า “The only way to reaching the information in an object is by using its operation.” [17]

“Abstract Data Type (ADT)” คือ Model ที่มี Operation ของมันที่จะกระทบต่อ Model [17]

ข้อดี ของ Encapsulation คือ สามารถลดความซับซ้อนลง เนื่องจากสามารถใช้งานโดยรู้เพียงคุณลักษณะต่างๆ เท่านั้น โดยไม่ต้องรู้ถึงโครงสร้างภายใน

- **Inheritance**

การสืบทอดคุณสมบัติต่างๆ ไปที่เหมือนกัน ระหว่าง Class

ตัวอย่าง Class ของผู้ชาย inherits Class ของคน

ข้อดี ของ Inheritance คือ

1. สามารถนำนิยามโดยทั่วๆ ไปมาใช้ใหม่ (Reuse common description)
2. การปรับปรุง แก้ไข ทำเพียงที่เดียวเท่านั้น (Modification can be made in only one place)
3. ลดความซ้ำซ้อนสำหรับ โปรแกรมย่อยเพื่อให้ง่ายต่อการทำความเข้าใจ (Redundancy can be reduced leading to smaller models that are easier to understand)

“Overriding” คือคุณสมบัติที่สามารถนิยาม Behavior หรือ Information ใหม่ได้สำหรับ Class ที่สืบทอด มาจาก Super Class

- **Polymorphism**

โดยความหมายคือ “Having many forms” หมายถึงการส่งข้อความเดียวกันให้กับ Object ที่ต่างกันจะแสดงพฤติกรรมที่แตกต่างกัน

ตัวอย่าง การบวกของ String และ Int จะมีสัญลักษณ์เดียวคือ + แต่ทำกับข้อมูลต่างชนิดกัน

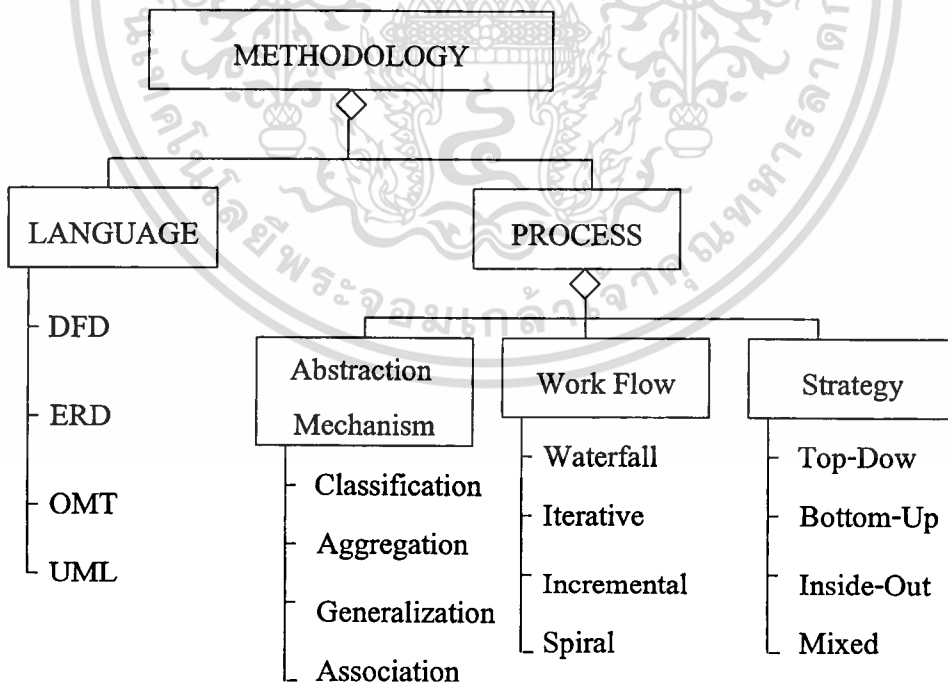
ข้อดี ของ Polymorphism คือ สนับสนุนการนำกลับมาใช้ใหม่ (Reuse) และมีการเปลี่ยนแปลงแก้ไขได้

3.4 Object-oriented Methodology

การพัฒนาารระบบด้วย Object-oriented Methodology นักวิเคราะห์ระบบจะต้องมีแนวคิดหรือระเบียบวิธีคิดในลักษณะเชิงวัตถุที่ได้กล่าวมาแล้ว ตั้งแต่กระบวนการวิเคราะห์ ออกแบบ การเขียนโปรแกรม ไปจนถึงการทดสอบการทำงานของระบบ หรือโปรแกรม (OO Analysis, OO Design, OO Programming และ OO Testing) ยกตัวอย่างเช่น การวิเคราะห์ระบบงาน จะต้องวิเคราะห์ให้ได้ว่าควรมี Class อะไรบ้าง แต่ละ Class มี พฤติกรรมการสื่อสารระหว่างกันอย่างไร เกิดเหตุการณ์อะไรบ้าง เป็นต้น จึงจะได้ออกมาเป็นระบบที่ต้องการ ต่างจากการวิเคราะห์และออกแบบในระบบแบบดั้งเดิม ที่มองแยกกันระหว่างการออกแบบในส่วนของคุณสมบัติ กับส่วนของกระบวนการทำงาน

3.5 ส่วนประกอบที่สำคัญของ Methodology

System Modeling Methodology โดยทั่วไปประกอบด้วย 2 ส่วน คือ Language และ Process Model โดย Language คือสัญลักษณ์ (Notation) ที่ใช้ในการอธิบาย แสดงความหมายของระบบงาน ส่วน Process หมายถึง การบอกกล่าวว่าจะทำอะไรในขั้นตอนใดของการวิเคราะห์ และออกแบบระบบงาน โดยสามารถแสดงได้ดังรูป



รูปที่ 3-5 โครงสร้างส่วนประกอบของ System Development Methodology [17]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น Object-oriented Methodology จึงประกอบด้วย 2 ส่วน เช่นกัน โดย Object-oriented Methodology ที่ศึกษาและใช้ในการจัดทำโครงการก็จะเลือก UML เป็น Language และใช้ Waterfall Process Model เป็นส่วนของการไหลเวียนของงาน (Work Flow) โดยจะกล่าวถึงรายละเอียดของ UML ที่สัมพันธ์กับ Waterfall Process Model ในบทถัดไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

Unified Modeling Language (UML)

และ Waterfall Process Model

1. UML คืออะไร

UML (Unified Modeling Language) เป็นภาษา หรือ รูปแบบจำลอง มาตรฐานที่ใช้ในการจำลองแบบในรายละเอียด (Blue Print) สำหรับการผลิตสร้าง (Construct) ซอฟต์แวร์ เช่นเดียวกับแบบจำลองทางธุรกิจ (Business Modeling) โดย UML จะเป็นการเสนอรูปแบบการปฏิบัติจริงในทางวิศวกรรมที่ดี (Best engineering practices) อีกทั้งยังเหมาะกับระบบที่มีขนาดใหญ่และยุ่งยากซับซ้อนอีกด้วย [15]

UML เป็นรูปแบบที่รวมข้อดีของรูปแบบอื่น ๆ เข้าไว้ด้วยกัน คือ

- Data Modeling Concept (Entity Relation Concept)
- Business Modeling (Work Flow)
- Object Modeling
- Component Modeling

นอกจากนี้ UML ยังสามารถใช้ได้ตลอดทั้งโครงการหรือการทำงาน เช่นเดียวกับ SDLC (Software Development Life Cycle)

2. ประวัติศาสตร์ของ UML

การกำหนดรูปแบบภาษาระดับวิเคราะห์และออกแบบเชิงวัตถุ เริ่มปรากฏในระหว่างกลางทศวรรษที่ 1970 ถึงปลายทศวรรษที่ 1980 มีการทดลองด้วยแนวความคิดที่แตกต่างกันอย่างแพร่หลายเกี่ยวกับการวิเคราะห์ และออกแบบเชิงวัตถุ และในช่วงระหว่างปี 1989 – 1994 ได้มีการเพิ่มจำนวนของรูปแบบภาษาจากเดิมซึ่งมีน้อยกว่า 10 รูปแบบไปเป็นมากกว่า 50 รูปแบบ และผู้ใช้จำนวนมากที่ใช้วิธีการทางเชิงวัตถุ (Object-oriented) ก็ยังไม่พบกับวิธีที่ตนเองพอใจได้

จนกระทั่งกลางทศวรรษที่ 1990 ได้เริ่มมีการพัฒนา UML ขึ้นในเดือนตุลาคม 1994 โดย Grady Booch และ Jim Rumbaugh ของ Rational Software Corporation ได้มีการรวมรูปแบบของ Booch และ OMT (Object Modeling Technique) เพื่อศึกษาวิธีการในเชิงวัตถุ หลังจากนั้นในเดือนตุลาคม 1995 ก็มีรูปแบบของ Unified Method 0.8 แต่ก็ไม่ประสบความสำเร็จไปในปี 1995 นั้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

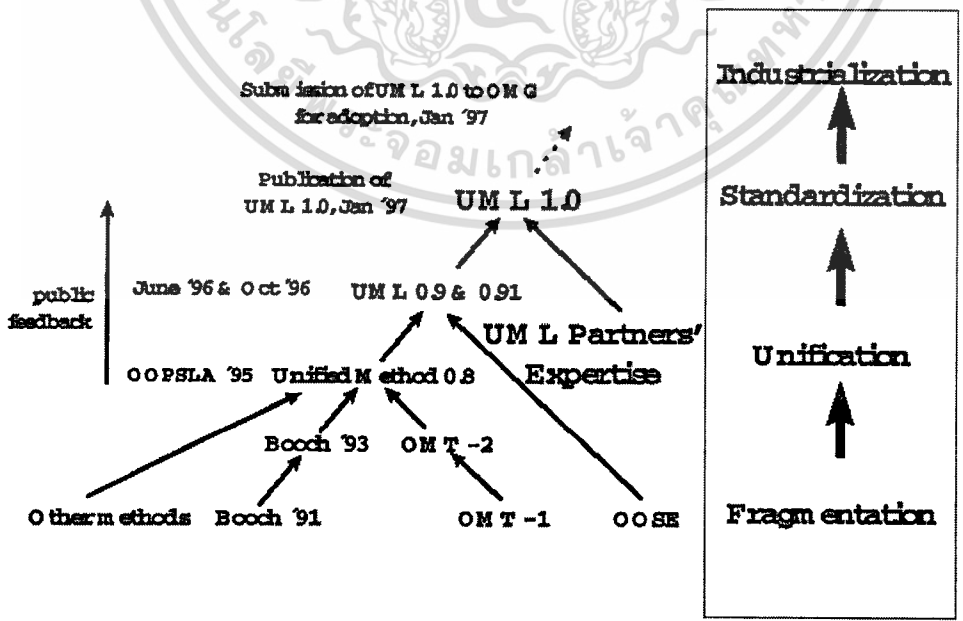
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ivar Jacobson ก็ได้นำรูปแบบนั้นมารวมกับวิธี OOSE (Object-oriented Software Engineering) และ 'สุดท้ายในปี 1996 จึงมี UML 1.0 มาเป็นมาตรฐานโดยการร่วมมือของหลาย ๆ บริษัท เช่น Digital Equipment Corp., i-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, TI และ Unisys

ปัจจุบัน UML ได้รับการรองรับจาก OMG (Object Management Group) ให้เป็นมาตรฐานหนึ่ง และไม่มีใครเป็นเจ้าของ เป็นรูปแบบเปิด กฎเกณฑ์ที่ทำให้ UML ประสบความสำเร็จ คือ

1. UML เป็นสิ่งที่ทำแล้วได้ผลในหลาย ๆ วิธีที่เป็นลักษณะรูปแบบทางภาษาที่เคยเสนอมาในอดีต
2. UML เป็นการรวมรูปแบบที่ใช้ได้ในหลาย ๆ ระบบ , ขั้นตอนการพัฒนาระบบ (Development Phase)

หลาย ๆ องค์กรได้มีการสนับสนุนมาตรฐานของ UML ตั้งแต่พื้นฐานรูปแบบทางภาษา (Modeling Language) ซึ่งเป็นวิธีการเชิงวัตถุ UML เป็นที่ยอมรับกันอย่างกว้างขวาง โดย UML 1.0 เป็นเวอร์ชันที่สามารถใช้ได้และมีเสถียรภาพ ในเดือนมกราคม 1997 เอกสารเกี่ยวกับ UML Version 1.0 ก็ได้รับการตอบสนองโดย Object Management Group (OMG) Analysis & Design Task Force's RFP-1 และในกลางปี 1997 OMG ได้มีการยอมรับมาตรฐานของ UML จึงทำให้หลาย ๆ องค์กรและผู้ผลิตได้ยอมรับมาตรฐาน UML จึงมีการใช้วิธีการสนับสนุนอื่น ๆ ทางด้าน UML มีเครื่องมือที่ช่วยในการพัฒนา ตลอดจนองค์กรที่ช่วยในการฝึกอบรม

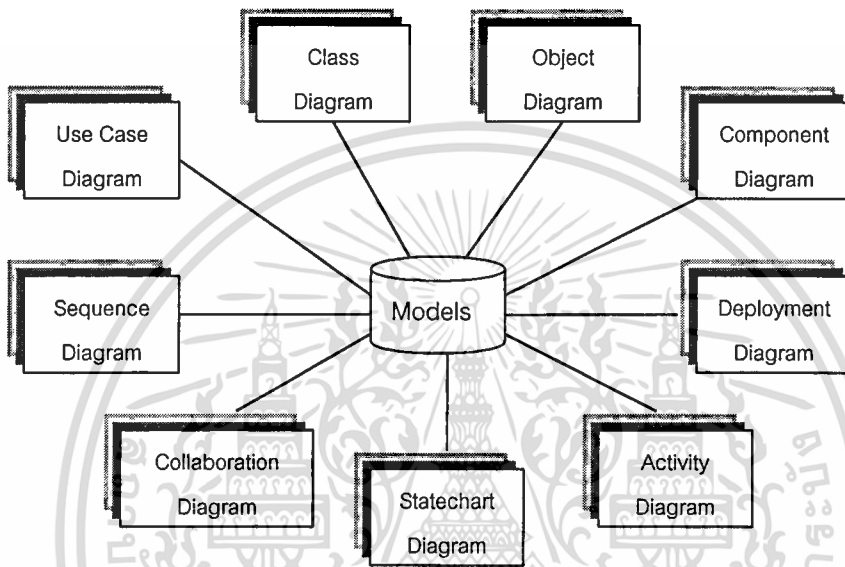


รูปที่ 4-1 แผนภาพแสดงประวัติศาสตร์ของ UML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ประเภทของ UML Diagram

UML ได้กำหนด Diagram ทั้งหมด 9 Diagram ดังแสดงในรูปที่ 4-2 เพื่อใช้ในการจำลองแบบระบบงาน โดยการมองในแง่มุมต่างๆ (Diagram แต่ละชนิด) เพื่อให้สามารถเข้าใจระบบงานให้มากที่สุด โดยผู้จำลองแบบไม่จำเป็นต้องใช้ทุก Diagram โดยสามารถเลือกใช้ Diagram ที่เหมาะสม



รูปที่ 4-2 แผนภาพแสดง UML Diagram ทั้งหมด 9 Diagram

UML ทั้ง 9 Diagram สามารถจัดแบ่งออกเป็นกลุ่มต่างๆ ได้ดังนี้

3.1 Functional Model

Functional Model เป็นแบบจำลองที่ใช้ในการแสดงความต้องการของระบบทั้งหมด ช่วยในการอธิบายรายละเอียดหลักๆ ภายในวัตถุ แสดงให้เห็นการไหลของข้อมูลในแต่ละการทำงาน โดยจะสนใจเพียงแค่ว่ามีงานอะไรบ้างที่ต้องทำ เครื่องมือที่ใช้ในการแสดงความต้องการของระบบคือ Use Case Diagram ใช้ในการแสดงความต้องการของระบบทั้งหมดในลักษณะที่ผู้ใช้งานสามารถเข้าใจได้ง่าย โดยจะถูกนำไปใช้ต่อไปใน Phase ต่างๆ ของการวิเคราะห์และออกแบบระบบ

3.2 Object Model

Object Model เป็นแบบจำลองที่ใช้ในการแสดงโครงสร้างของระบบ โดยจะแสดงในรูปแบบของ Class ต่างๆ พิจารณาจากความต้องการของระบบ ที่แสดงอยู่ใน Functional model เครื่องมือที่ใช้ในการแสดงโครงสร้างของระบบ คือ Component Diagram แสดงถึงโครงสร้างและความสัมพันธ์ขององค์ประกอบต่างๆ ของซอฟต์แวร์ ส่วน Class Diagram แสดงถึงโครงสร้างและความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ Class Object Diagram แสดงถึงโครงสร้างและความสัมพันธ์ของ Instance และ Deployment Diagram แสดงถึงความสัมพันธ์ทางกายภาพระหว่างส่วนประกอบของฮาร์ดแวร์และซอฟต์แวร์

3.3 Dynamic Model

Dynamic Model เป็นแบบจำลองที่ใช้ในการแสดงถึงการทำงานระหว่าง Object ต่างๆ ตามการส่งข้อความ (Message) หรือเมื่อเหตุการณ์ (Event) ต่างๆ ได้เกิดขึ้น Object ในที่นี้หมายถึง Instance ที่สร้างขึ้นจาก Class ที่ได้ออกแบบไว้ใน Object Model มีคุณสมบัติ และพฤติกรรมเช่นเดียวกับ Class ต้นแบบ ในการทำงานของระบบจะประกอบขึ้นจากการส่งข้อความไปมาระหว่าง Object เหล่านั้น เมื่อมีการทำงานไปเรื่อยๆ แล้ว Object อาจจะมีการเปลี่ยนสถานะ(State) ไปตามเหตุการณ์ที่เกิดขึ้นได้ เพื่อให้เป็นตามที่กำหนดไว้ใน Functional Model เครื่องมือที่ใช้คือ Sequence Diagram แสดงให้เห็นถึงการแลกเปลี่ยนข่าวสารระหว่าง Object ต่างๆ, Collaboration Diagram สามารถแสดงถึงความสัมพันธ์ระหว่าง Object ภายใน, Activity Diagram แสดงถึงสถานะการทำงานในแต่ละ Action และ State Diagram แสดงวงจรชีวิตของ Object บอกถึงว่าเหตุการณ์ต่างๆ มีผลกระทบอะไรเกิดขึ้นบ้าง

4. ความหมายของแต่ละ UML Diagram

จะอธิบายความหมายและสัญลักษณ์ต่างๆ ที่ใช้ใน UML Diagram โดยจะอธิบายเฉพาะ diagram ที่เลือกใช้ในการพัฒนาในโครงการพัฒนาระบบงานนี้ 4 Diagram ได้แก่ Use Case Diagram, Class Diagram, Component Diagram และ Sequence Diagram ดังนี้

4.1 Use Case Diagram

Use Case Diagram เป็นแนวความคิดของไอวาร์ จาคอบสัน (Ivar Jacobson) ซึ่ง OMG ได้รวมไว้ในมาตรฐานของ UML ด้วย และมีวิธีการอื่นๆ นำแนวคิดนี้ไปใช้อย่างกว้างขวาง Use Case Diagram เป็นสิ่งที่ใช้ในการแสดงความต้องการของระบบทั้งหมดในลักษณะที่ผู้ใช้งานสามารถเข้าใจได้ง่าย โดยแสดงความสัมพันธ์ระหว่างผู้ใช้กับระบบต่างๆ สัญลักษณ์ได้ดังนี้

- **Actor** คือสิ่งที่อยู่ภายนอกระบบที่จะพัฒนาขึ้นทั้งหมด ที่ต้องการแลกเปลี่ยนหรือส่งข้อมูลให้กับระบบ โดยที่ Actor อาจจะเป็นคน ระบบ หรือโปรแกรมอื่นๆ ก็ได้ โดยมากจะเป็นผู้เริ่มทำงานกับ Use Case เช่นผู้ใช้ระบบ เป็นต้น



Human

รูปที่ 4-3 สัญลักษณ์ของ Actor

- **Use Case** จะเป็นตัวแทนงานที่เกิดขึ้นในขั้นตอนต่างๆ โดยจะใช้สัญลักษณ์เป็นรูปวงรี หรือวงกลม ถ้า Use Case ใดมีกรอบสี่เหลี่ยมสีเทาล้อมรอบแสดงว่ามี Diagram ย่อยที่ใช้อธิบายรายละเอียดของ Use Case ต่อไป



รูปที่ 4-4 สัญลักษณ์ของ Use Case

- **Communication** เป็นการแสดงความสัมพันธ์หรือการติดต่อสื่อสารกัน (การรับและให้ข้อมูลข่าวสารแก่กันและกัน) ระหว่าง Actor และ Use Case ซึ่งอาจจะเป็นการสื่อสารแบบทางเดียว หรือ 2 ทางก็ได้ แสดงด้วยสัญลักษณ์เส้นที่มีหัวลูกศรสีดำ



รูปที่ 4-5 สัญลักษณ์ของ Communication

- **Relationship** เป็นการแสดงความสัมพันธ์ระหว่าง Use Case กับ Use Case โดยใช้เส้นที่มีหัวลูกศรสีขาว ความสัมพันธ์ระหว่าง Use Case มีได้ 2 แบบ คือ Extends และ Uses



รูปที่ 4-6 สัญลักษณ์ของ Relationship

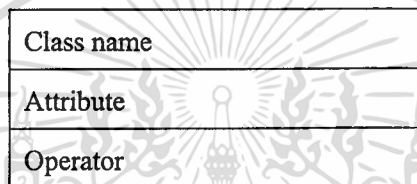
Extends เป็นการเพิ่มการทำงานให้กับ Use Case โดยการเรียกใช้จากอีก Use Case หนึ่ง ลูกศรจะออกจาก Use Case ที่ถูกเรียกใช้งานไปยัง Use Case ที่ต้องการเพิ่มเติมการทำงาน และจะมีข้อความ <<extends>> ระบุอยู่ข้างเส้น

Uses เป็นการแสดงให้เห็นถึงการ Inherit หน้าที่การทำงานจาก Use Case หนึ่ง ไปอีก Use Case หนึ่ง ปลายลูกศรจะอยู่ที่ Use Case หลักที่จะถูกถ่ายทอดความสามารถออกไป โดยจะมีข้อความ <<uses>> ปรากฏอยู่ข้างเส้น

4.2 Class Diagram

Class Diagram เป็น Model ชนิดหนึ่งที่เป็น Static Model จะเป็นการอธิบายถึง Class และความสัมพันธ์ระหว่าง Class ที่มีโครงสร้างของข้อมูลรวมถึงพฤติกรรมของข้อมูลที่แตกต่างกัน ซึ่ง Class 1 Class สามารถกำหนดทิศทางของการนำไปสร้างโปรแกรมและการสร้าง Class ใน Object-oriented ได้

Class ใช้สัญลักษณ์เป็นรูปสี่เหลี่ยม จะอธิบายถึงกลุ่มของ Object ต่างๆ ที่มีคุณลักษณะ (Attribute) หน้าที่การทำงาน (Operation) และความสัมพันธ์ (Relationship) ขึ้นพื้นฐาน ซึ่งในแต่ละ Class ประกอบไปด้วยรายละเอียดพื้นฐาน 3 ส่วนดังรูป



รูปที่ 4-7 สัญลักษณ์ของ Class

- Class Name แสดงชื่อของ Class ที่กำหนดในระบบ
- Attribute เป็นการกำหนดคุณลักษณะทั้งหมดที่มีภายใน Class บอกถึงรายละเอียดของข้อมูล (Attribute) ภายใน Class เดียวกันจะไม่ซ้ำกัน แต่ชื่อข้อมูลใน Class อาจจะไปซ้ำกับชื่อข้อมูลใน Class อื่นได้
- Operation เป็นส่วนที่ใช้อธิบายใน Class นั้นมี Method อะไรบ้าง มีการรับค่าชุดของตัวแปร (Argument) อะไรบ้าง หรือมีการส่งค่าออกไปหรือไม่
- ความสัมพันธ์ระหว่าง Class (Relationship)

UML ได้เตรียมความสัมพันธ์ระหว่าง Class ไว้ จะมีความสัมพันธ์อยู่ 4 แบบ ดังนี้

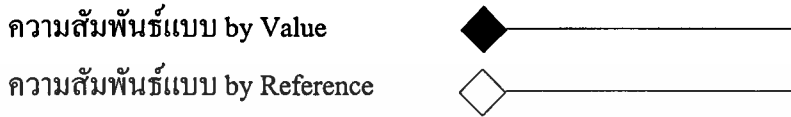
1. Association จะแสดงความสัมพันธ์ระหว่าง Class จะมีได้ทั้งทางเดียว และ 2 ทาง มีสัญลักษณ์ดังนี้

ความสัมพันธ์แบบทางเดียว

ความสัมพันธ์แบบสองทาง

รูปที่ 4-8 สัญลักษณ์ของ Association Relationship

- 2. Aggregation เป็นรูปแบบพิเศษของ association คือเป็นความสัมพันธ์ระหว่าง Whole และ Parts ของมัน คือ Whole จะประกอบไปด้วย Parts ต่างๆ ของมัน ดังนั้นการคงอยู่ของ Parts จะต้องขึ้นกับ Whole หรือจะเรียกความสัมพันธ์แบบนี้ว่า “Whole-Part” ก็ได้ มีสัญลักษณ์ดังนี้



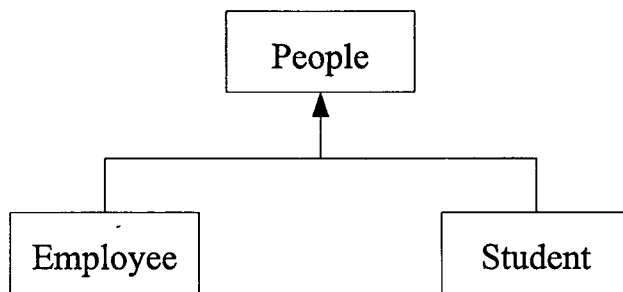
รูปที่ 4-9 สัญลักษณ์ของ Aggregation Relationship

- 3. Depends on เป็นรูปแบบความสัมพันธ์แบบหนึ่งที่ใช้แสดงความสัมพันธ์กันระหว่าง คลาส 2 คลาส ในแง่ที่คลาส หนึ่งเรียกใช้บริการของอีกคลาสหนึ่ง กล่าวคือ Class ของ ผู้ขอบริการขึ้นอยู่กับบริการของ Class ของผู้ให้บริการ แต่ไม่มีการขึ้นต่อกันภายใน โครงสร้างของ Class มีสัญลักษณ์ดังนี้



รูปที่ 4-10 สัญลักษณ์ของ Depend on Relationship

- 4. Generalization ความสัมพันธ์รูปแบบนี้ใช้แสดงความสัมพันธ์ระหว่าง Class กับ Class ในแง่ที่ Class หนึ่งถ่ายทอดคุณสมบัติและ โครงสร้างจากอีก Class หนึ่ง โดยเรียก Class ที่ถูกถ่ายทอดว่า Super Class และเรียกคลาสที่ทำการถ่ายทอดว่า Sub Class มีสัญลักษณ์ ดังตัวอย่างในรูปที่ 2-1

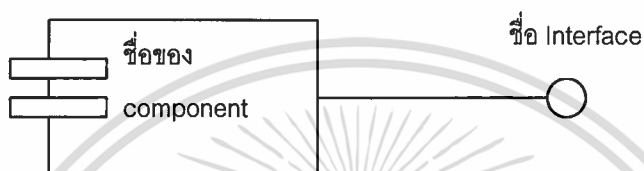


รูปที่ 4-11 ตัวอย่างความสัมพันธ์แบบ Generalization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 Component Diagram

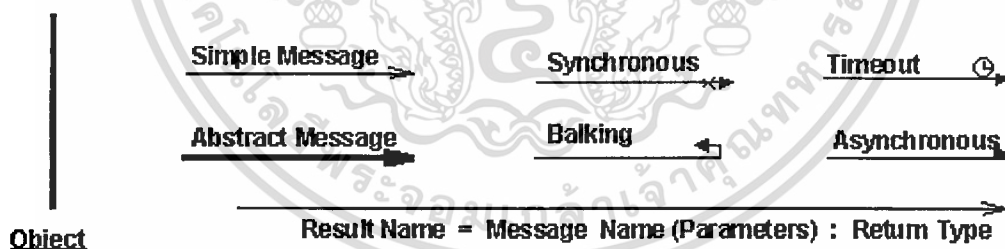
เป็น Diagram ที่แสดงถึงโครงสร้างและความสัมพันธ์ระหว่างองค์ประกอบต่างๆ ของซอฟต์แวร์ ซึ่งองค์ประกอบดังกล่าวอาจเป็น Source Program หรือ Library หรือ Executable Program ก็ได้ โดย Component Diagram จะเป็นแบบจำลองที่ประกอบด้วยองค์ประกอบต่างๆ ของระบบเชื่อมโยงกันโดยใช้ความสัมพันธ์ในลักษณะที่ขึ้นต่อกันในรูปของเส้นประที่มีหัวลูกศรชี้จาก Component ถูก ไปยัง Component หลัก ดังรูป



รูปที่ 4-12 สัญลักษณ์ที่ใช้ภายใน Component Diagram

4.4 Sequence Diagram

เป็น Diagram ที่แสดงให้เห็นถึงการทำงานระหว่าง Object ต่างๆ ตามการส่งข้อความ และเมื่อเหตุการณ์ต่างๆ ได้เกิดขึ้น ซึ่งผู้เขียนโปรแกรมจะใช้ Diagram ตัวนี้เป็นตัวช่วยเพื่อที่จะได้เขียนโปรแกรมให้ได้ตามที่ได้ออกแบบไว้ สัญลักษณ์ที่ใช้ใน Sequence Diagram มีดังนี้



รูปที่ 4-13 สัญลักษณ์ต่างๆ ที่ใช้ใน Sequence Diagram

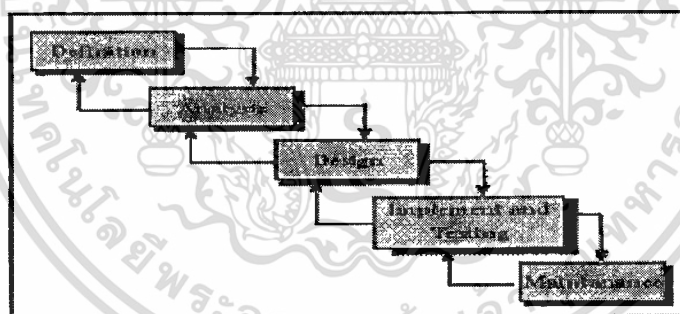
- Object คือ Instance ของ Class ที่ทำหน้าที่รับ-ส่งข้อความ (Message) จาก Object อื่นๆและทำการตอบสนองตามข้อความนั้นๆเพื่อให้เกิดการทำงานในขั้นตอนต่างๆของระบบโดยจะใช้สัญลักษณ์เป็นเส้นตรงแนวตั้งและมีชื่อของ Object และ Class กำหนดอยู่ด้านล่าง
- Message เป็นข้อความที่ส่งไปมาระหว่าง Object ที่ถูกเรียก ซึ่ง Message จะประกอบไปด้วยส่วนประกอบต่างๆดังต่อไปนี้
 - Result Name คือตัวแปรที่จะจัดเก็บค่าที่ได้จากการส่งข้อความ (Message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Parameters คือ Argument ต่างๆที่ส่งให้กับ Object เพื่อทำงานตาม Method ที่กำหนดด้วย Message นั้นๆ
- Return Type คือประเภทของข้อมูลที่ส่งกลับมากหลังจากการทำงานของการส่ง Method นั้นๆ ถ้า Method นั้นกำหนดว่ามีการส่งค่ากลับมา
- Constraint คือเงื่อนไขการส่ง Message

5. Waterfall Process Model

ในการพัฒนาซอฟต์แวร์เป็นเรื่องที่ค่อนข้างซับซ้อนและมีรายละเอียดค่อนข้างมาก จึงได้มีการแบ่งกระบวนการพัฒนาซอฟต์แวร์ออกเป็น Phase ต่างๆ เรียกว่า Software Development Life Cycle model ซึ่งการแบ่งขั้นตอนการพัฒนาออกเป็น Phase ต่าง ๆ ทำให้การปฏิบัติและติดตามผลของการพัฒนาง่ายขึ้น ซึ่งได้มีการกำหนดมาตรฐาน Software Development Life Cycle ไว้หลายโมเดล เช่น Waterfall Model, Rapid Prototyping Model, Spiral Model, Concurrent Development Model และ Formal Methods Model เป็นต้น ในที่นี้จะนำ Waterfall Model มาเป็นหลักในการประยุกต์ใช้กับการวิเคราะห์และออกแบบเชิงวัตถุ เพราะ Waterfall Model นั้นเป็น Model ที่นิยมและเข้าใจง่าย



รูปที่ 4-14 แผนภาพแสดง Waterfall Model

Waterfall Model จะแบ่ง Phase การทำงานออกเป็น 5 Phase คือ

1. Phase ของการให้คำนิยามต่างๆ ของซอฟต์แวร์ (Definition)
2. Phase ของการวิเคราะห์ (Analysis)
3. Phase ของการออกแบบ (Design)
4. Phase ของการปฏิบัติและทดสอบ (Implement and Testing)
5. Phase ของการบำรุงรักษา (Maintenance)

โดยสามารถอธิบายแต่ละ Phase ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Phase 1: การให้คำนิยามของซอฟต์แวร์ (Definition)**

Phase นี้เป็น Phase ที่สำคัญที่สุด เนื่องจากการศึกษาถึงความต้องการต่างๆ ของระบบ ปัญหาหลักของระบบคืออะไร ขอบเขตของระบบงานที่สนใจมีแค่ไหน ความต้องการของผู้ใช้มีอะไรเพิ่มเติมบ้าง และมีคำศัพท์อะไรบ้างจะต้องมีการให้คำนิยามของศัพท์เหล่านั้นเพื่อความเข้าใจที่ตรงกัน ถ้าใน Phase นี้วิเคราะห์ความต้องการของระบบไม่ชัดเจน ก็จะมีผลถึง Phase ต่อไปของการดำเนินงานด้วย ผลลัพธ์ที่ได้จาก Phase นี้เป็น Functional Model คือ Use Case Diagram และพจนานุกรมของข้อมูล (Data Dictionary)

- **Phase 2: การวิเคราะห์ (Analysis)**

ใน Phase การวิเคราะห์จะเป็นหลักสำคัญของการที่จะศึกษาและออกแบบถึงโครงสร้างของระบบงานตามขอบเขตของปัญหาที่สนใจมีอะไรบ้างตาม Use Case Diagram ที่ได้จาก Phase ที่ 1 ใน Phase นี้จะได้ผลลัพธ์เป็น Object Model ผู้วิเคราะห์จะทำการรวบรวมคลาสที่สัมพันธ์กันขึ้นมาเป็นหมวดหมู่ เรียกว่า Component เพื่อเป็นการลดความซ้ำซ้อนของระบบ

- **Phase 3: การออกแบบ (Design)**

การออกแบบระบบจะเริ่มเมื่อข้อมูลในช่วงของการวิเคราะห์เพียงพอที่จะเริ่มทำการปฏิบัติกับระบบแล้ว ในการออกแบบควรจะพิจารณาถึง Hardware ว่า ระบบเป็น Centralized หรือ Distributed ถ้าเป็นระบบแบบ Distributed มีการจัดการกับระบบความปลอดภัยอย่างไร จะใช้อะไรเป็นตัวติดต่อสื่อสาร ต้องการ Object เพิ่มเติมเพื่อช่วยอำนวยความสะดวกในการติดต่อสื่อสารหรือไม่ เป็นต้น แล้วพิจารณาว่า Software ว่ามี Class พิเศษอะไรบ้างเพื่อสร้างหน้าต่างส่วนที่ติดต่อกับผู้ใช้ (User Interface) ต้องการคลาสเพิ่มเติมเพื่อช่วยอำนวยความสะดวก ในเรื่องของความสัมพันธ์ระหว่าง Class หรือไม่ ถ้าในระบบต้องการใช้ระบบฐานข้อมูล มีคลาสพิเศษที่ต้องใช้ในการติดต่อระหว่าง DBMS หรือไม่ ต้องมีการลดความซ้ำซ้อนของข้อมูลหรือไม่ ต้องทำการ Normalization หรือไม่ มีการใช้ Class ของบุคคลภายนอกหรือไม่ การติดต่อระหว่าง Object เป็นแบบ พร้อมกัน (Synchronously) หรือ เกิดขึ้นไม่พร้อมกัน (Asynchronously) เป็นต้น

ผลลัพธ์ที่ได้จาก Phase นี้ควรจะได้ Class เพิ่มเติม เช่น Class เกี่ยวกับการติดต่อกับผู้ใช้งาน คือหน้าต่าง ต่างๆ (GUI Classes), Class ที่จัดการและติดต่อกับฐานข้อมูล (Database Classes), การสนทนาระหว่าง Class (Dialogue) และประเภทของ Object ที่ต้องการเป็นแบบถาวร (Persistent Object), ความสัมพันธ์ระหว่าง Object , Transaction ต่างๆ ที่สนใจ Dynamic Model จะมี Sequence Diagram และ/หรือ State Diagram

● **Phase 4: Implement และ Testing**

Implement และ Testing เป็น Phase ของการเขียนโปรแกรมและทดสอบให้ได้เป็นไปตามการวิเคราะห์ โดยจะใช้การโปรแกรมเชิงวัตถุ (Object-oriented Programming) ซึ่งการแปลงจากการวิเคราะห์และออกแบบไปเป็นการโปรแกรมเชิงวัตถุ ส่วนการ Testing ก็เหมือนเป็นการตรวจสอบคุณภาพของโปรแกรมที่เราทำขึ้นมาที่มีวิธีการตรวจสอบ Object รองรับ (Object Testing) การแปลงจากการวิเคราะห์และออกแบบไปเป็นการโปรแกรมสามารถเชิงวัตถุทำได้ไม่ยุ่งยาก ดังตารางที่ 4-1 เนื่องจาก ทั้งการวิเคราะห์และออกแบบเชิงวัตถุ กับการโปรแกรมเชิงวัตถุ นั้นสนับสนุนคุณสมบัติของ Object-oriented ได้แก่ Encapsulation, Inheritance และ Polymorphism

การวิเคราะห์และออกแบบระบบเชิงวัตถุ (UML notation)	การโปรแกรมเชิงวัตถุ (ภาษา Java)
Component	เป็น package ในภาษา Java
Class	Class
Attribute	Attribute
Method	Method
Attribute ต่างๆ ถ้าเป็น Read only	เป็น Set Method
Attribute ต่างๆ ถ้าเป็น Read-write	เป็น Get, Set method
Inherit	Sub Class จะต้อง Extends จาก Super Class
ถ้า Component มีการอ้างอิงถึง Module หรือ Component อื่นๆ	จะต้อง Import Package นั้นๆ
Public (+)	เป็น Public ใน Java
Private (-)	เป็น Package ใน Java
Protected (#)	เป็น Protected ใน Java
Implementation (?)	คือการกำหนด ณ ขณะ Implement เราอาจจะใช้ Private ในภาษาจาวาก็ได้แล้วแต่ขอบเขตกรณีที่เราศึกษา

ตารางที่ 4-1 การเทียบจาก UML ไปเป็นการโปรแกรมเชิงวัตถุ (ภาษา Java)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Phase 5: Maintenance**

ในการ Maintenance สำหรับการพัฒนาด้วยแนวคิดเชิงวัตถุจะทำได้สะดวกกว่า ยกตัวอย่าง เช่น ถ้ามีการเปลี่ยนแปลงโครงสร้างของ Class ก็จะมีการเปลี่ยนแปลงและ Implement คลาสนี้ใหม่เท่านั้น คลาสอื่นๆ ที่เกี่ยวข้องไม่จำเป็นต้องเปลี่ยนแปลง

จะเห็นว่าตั้งแต่ Phase ที่ 1 ถึง Phase ที่ 5 การวิเคราะห์และออกแบบเชิงวัตถุจะใช้เครื่องมือที่มีลักษณะเดียวกันคืออ้างอิงถึง Object มองทุกอย่างเป็น Object ทำให้เมื่อมีการเปลี่ยนแปลงแก้ไขง่าย สะดวกต่อการจัดการ การส่งทอดระหว่าง Phase หนึ่งถึงอีก Phase หนึ่งจะไม่มีช่องว่างเพราะอ้างอิงถึง Object เหมือนกัน และมี Object-oriented Case Tool ที่สนับสนุนการวิเคราะห์และออกแบบเชิงวัตถุออกหลายบริษัท แต่ที่นี่จะเลือกใช้ เช่น Rose จากบริษัท Rational ดังนั้นวิธีการวิเคราะห์และออกแบบระบบเชิงวัตถุจึงเป็นวิธีการที่น่าสนใจและมีแนวโน้มที่ดีในอนาคต



บทที่ 5

Object-oriented Case Tool: Rational Rose

5.1 Rational Rose 2000 Enterprise Edition

Rational Rose 2000 Enterprise Edition เป็น Visual Modeling Tool และ Object-oriented Case Tool ที่สามารถสร้าง UML Diagram ทั้งหมด เพื่อใช้ในการจำลองระบบงานได้ ด้วยการวาด Diagram ต่างๆ โดยมีตัวจัดการ Diagram และกำหนดคุณสมบัติต่างๆ ของ Diagram ที่จำเป็น การเชื่อมโยงกันระหว่าง Diagram และการจัดการเอกสารจากแบบจำลองที่ได้สร้างขึ้น รวมทั้งมีตัวช่วยอัตโนมัติในการสร้าง Class และ Add-in ต่างๆ เพิ่มเติม เช่น Rational Rose J และ Rational Rose Oracle8 ที่จะกล่าวถึงต่อไป

5.2 Rational Rose J

Rational Rose J เป็นส่วนเพิ่มเติมของ Rational Rose ที่มีคุณสมบัติดังต่อไปนี้

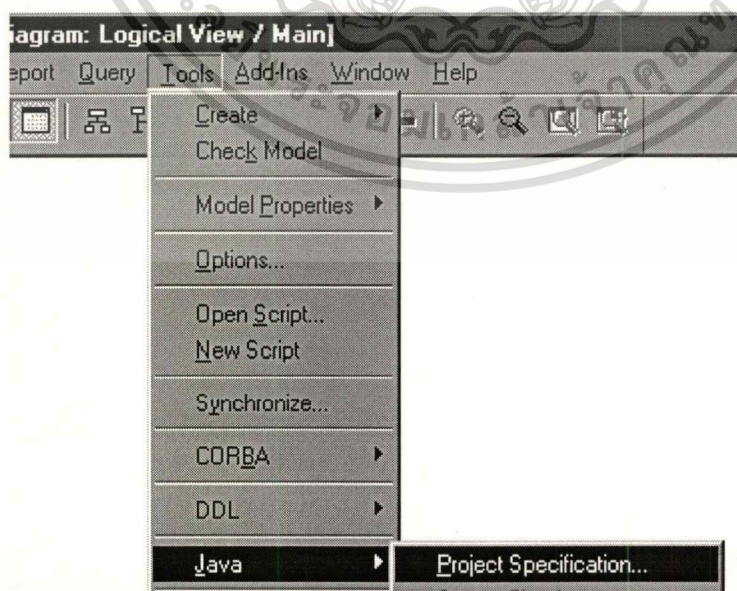
1. สนับสนุนวิวัฒนาการจากขั้นวิเคราะห์ไปจนถึงการออกแบบ
2. สนับสนุนการทำงานเป็นทีมโดยการทำ Configuration Management และ Version Control System
3. สามารถสร้าง Source Code ภาษา Java จาก Model
4. สามารถเพิ่มเติม Classpath ภายใน Model
5. สนับสนุนไฟล์ชนิด zip, jar และ cab ใน Classpath
6. สามารถลากและวางไฟล์ชนิด java, class, zip, jar และ cab เข้าไปใน Model โดยการ ทำ Reverse Engineering อย่างรวดเร็ว
7. ขยายไฟล์แบบอัตโนมัติสำหรับไฟล์ชนิด zip, jar และ cab เมื่อขณะทำการ Reverse Engineering
8. สามารถออกแบบจำลอง ทำให้เห็นภาพของการสร้างโปรแกรมของภาษา Java ได้แก่ Package, Class, Interfaces, Imports, Inheritance, Fields, Methods และ Modifiers
9. สามารถควบคุมคุณสมบัติต่างๆ อย่างอัตโนมัติ ได้แก่ Finalizers, และ Static Initializers, และสามารถกำหนด ชื่อ Field-name ได้เอง
10. สนับสนุนการ Application Partitioning และ Distribution

11. การ Mapping อย่างฉลาดจาก Model ไปสู่ Code
12. สนับสนุน Default Package ต่างๆ
13. ปรับปรุงการสนับสนุน Class ภายในและ Classes อื่นๆ ที่ไม่อยู่ในระบบ
14. การสร้าง Bean (Get/Set)
15. โปรแกรม Editor สามารถแสดงสีได้
16. สนับสนุน Framework ขนาดใหญ่ ประกอบไปด้วย Component ให้เลือกใช้งาน อาทิ
 - JDK Class Library
 - Java Database Connectivity (JDBC)
 - Java Generic Library (JGL)
 - Application Foundation Classes (AFC)
 - Windows Foundation Classes
 - Enterprise Java Frameworks (JNI, EJB, JTS)

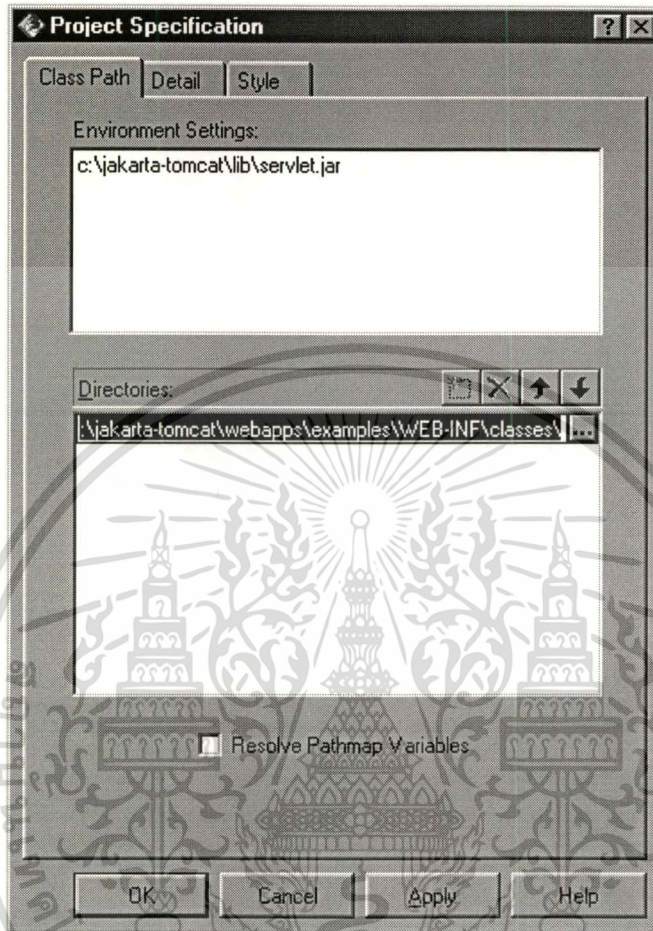
5.2.1 ตัวอย่างการทำ Forward Engineering ด้วย Rational Rose J

Forward Engineering คือ กระบวนการสร้างโปรแกรมภาษา Java จาก Class หรือ Package หรือ Component ใน Rational Rose Model

ขั้นแรกจะต้องทำการเซต Classpath ให้ชี้ไปยัง Directories หรือไฟล์ชนิด .jar, .cab, .zip ที่ต้องการใช้งาน โดยเพิ่มเติมในส่วนของ Directories ที่เพิ่มไปจาก Classpath Environment Variable ที่ถูกกำหนดไว้ก่อนแล้ว ดังรูปที่ 5-1 และ 5-2

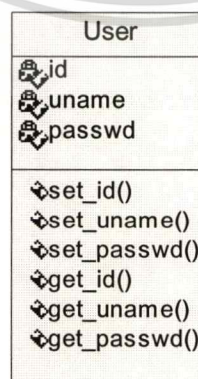


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 5-1 วิธีการเข้าสู่หน้าจอกำหนด Classpath ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-2 หน้าจอขณะทำการกำหนด Classpath ที่เก็บของไฟล์ .java, .class, .jar, .zip

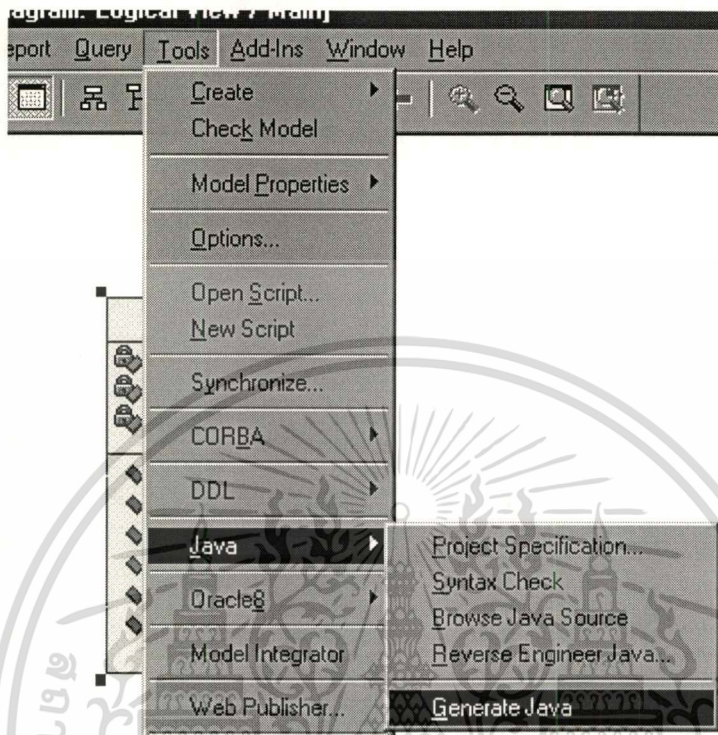
ขั้นที่ 2 ให้ทำการสร้าง Class Diagram ที่ต้องการ ดังเช่นตัวอย่างต่อไปนี้ สร้าง Class User



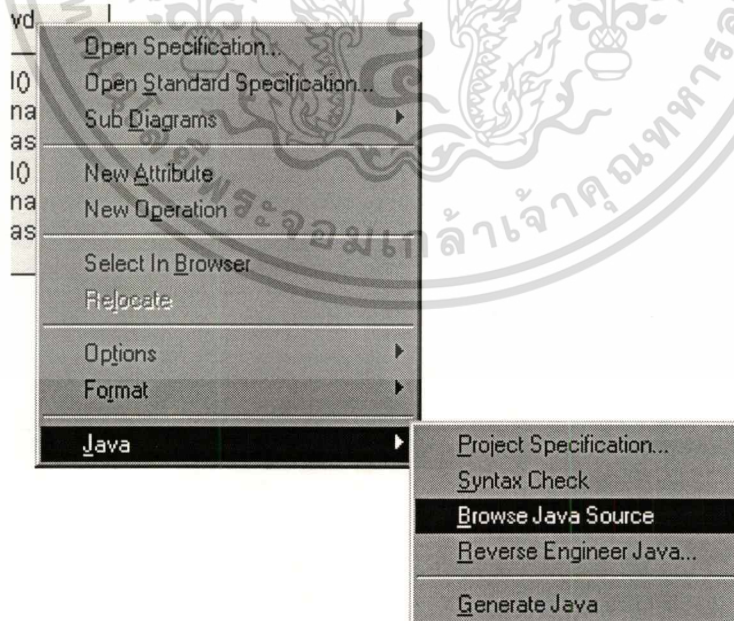
รูปที่ 5-3 Class User ใน Rational Rose Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3 เลือก Class ที่ต้องการ แล้วเลือก Tool > Java > Generate Java

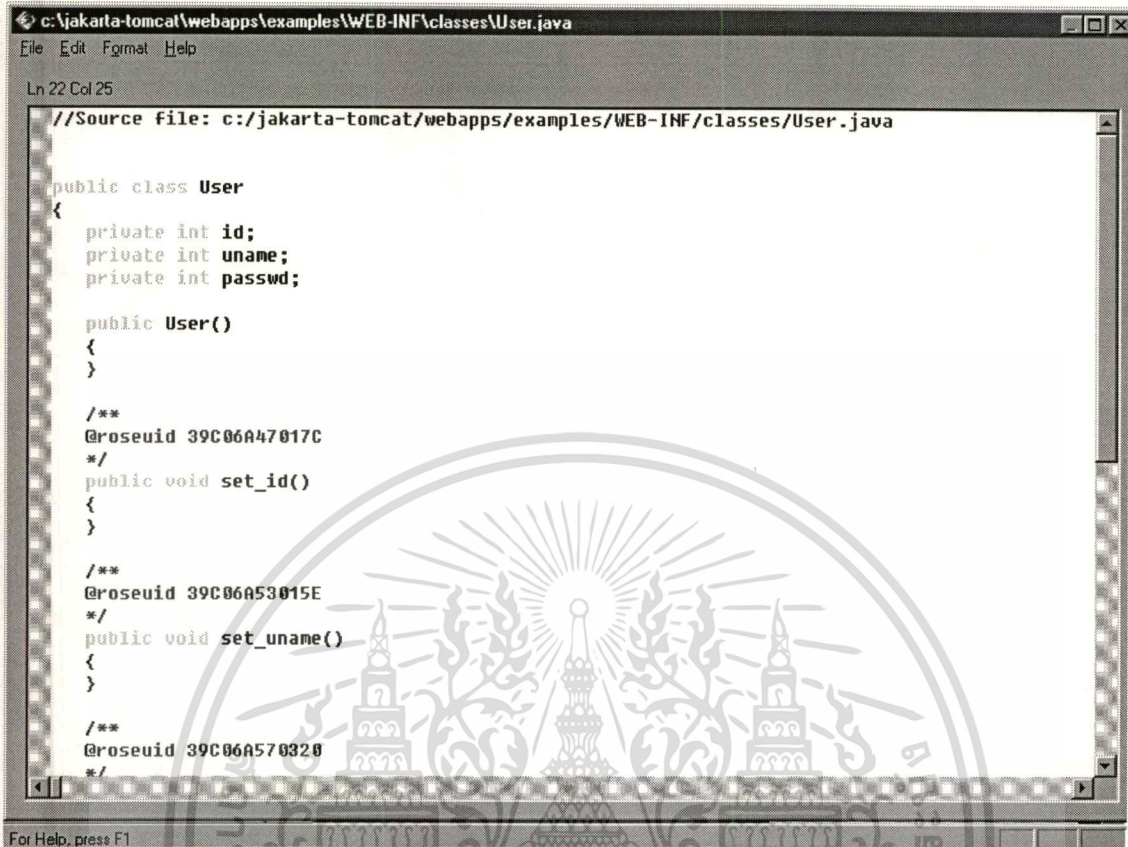


รูปที่ 5-4 วิธีการสั่งให้สร้าง Source Code จาก Class ที่ต้องการ



รูปที่ 5-5 วิธีการดูผลลัพธ์ Source Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

c:\jakarta-tomcat\webapps\examples\WEB-INF\classes\User.java
File Edit Format Help
Ln 22 Col 25

//Source file: c:/jakarta-tomcat/webapps/examples/WEB-INF/classes/User.java

public class User
{
    private int id;
    private int uname;
    private int passwd;

    public User()
    {
    }

    /**
     * @roseuid 39C06A47017C
     */
    public void set_id()
    {
    }

    /**
     * @roseuid 39C06A53015E
     */
    public void set_uname()
    {
    }

    /**
     * @roseuid 39C06A570320
     */
}

```

รูปที่ 5-6 ผลลัพธ์ Source Program

เมื่อทำตามขั้นตอนจะสามารถเข้าสู่ผลลัพธ์ได้ดังรูปที่ 5-5 และ 5-6 ไฟล์ .java ที่สร้างขึ้นจะบันทึกอยู่ใน Directory ที่กำหนดไว้ใน Classpath

5.2.2 ตัวอย่างการทำ Reverse Engineering ด้วย Rational Rose J

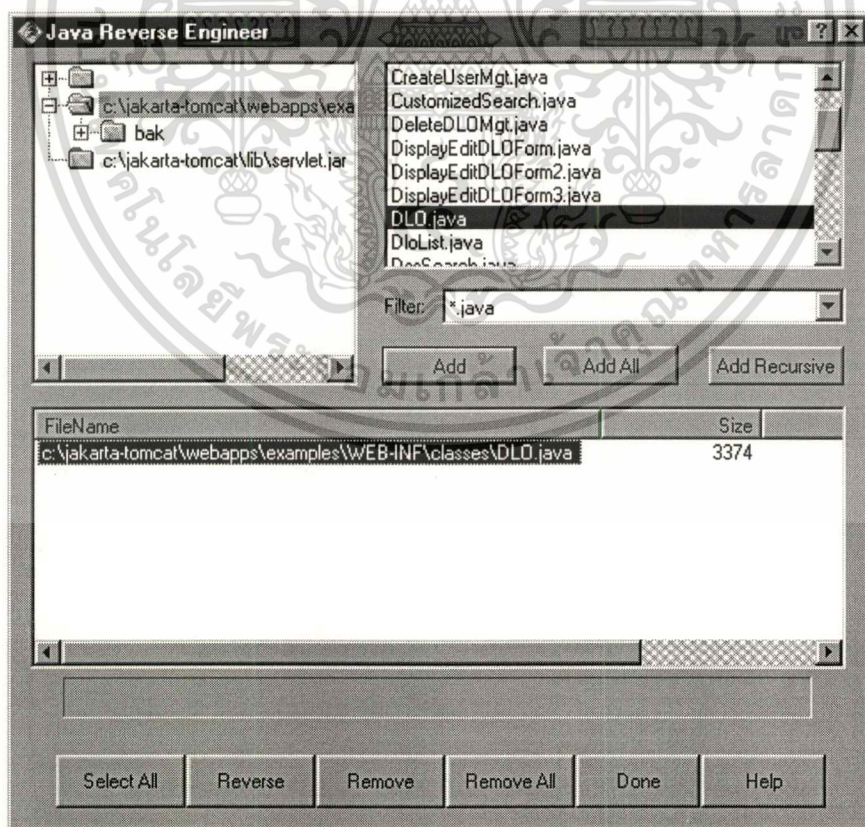
Reverse Engineering คือ กระบวนการวิเคราะห์ Source Code ภาษา Java แล้วแปลงกลับไปเป็น Class หรือ Component Model ใน Rational Rose

เช่นเดียวกับกัน ขั้นแรกจะต้องทำการเซต Classpath ให้ชี้ไปยัง Directories หรือไฟล์ชนิด .jar, .cab, .zip ที่ต้องการใช้งาน โดยเพิ่มเติมในส่วนของ Directories ที่เพิ่มไปจาก Classpath Environment Variable ที่ถูกกำหนดไว้ก่อนแล้ว ดังรูปที่ 5-1 และ 5-2

ขั้นที่ 2 เลือกไฟล์ .jar, .class ที่ต้องการ โดยเข้าที่เมนู Tool > Java > Reverse Engineering Java ดังรูปที่ 5-7 และ 5-8



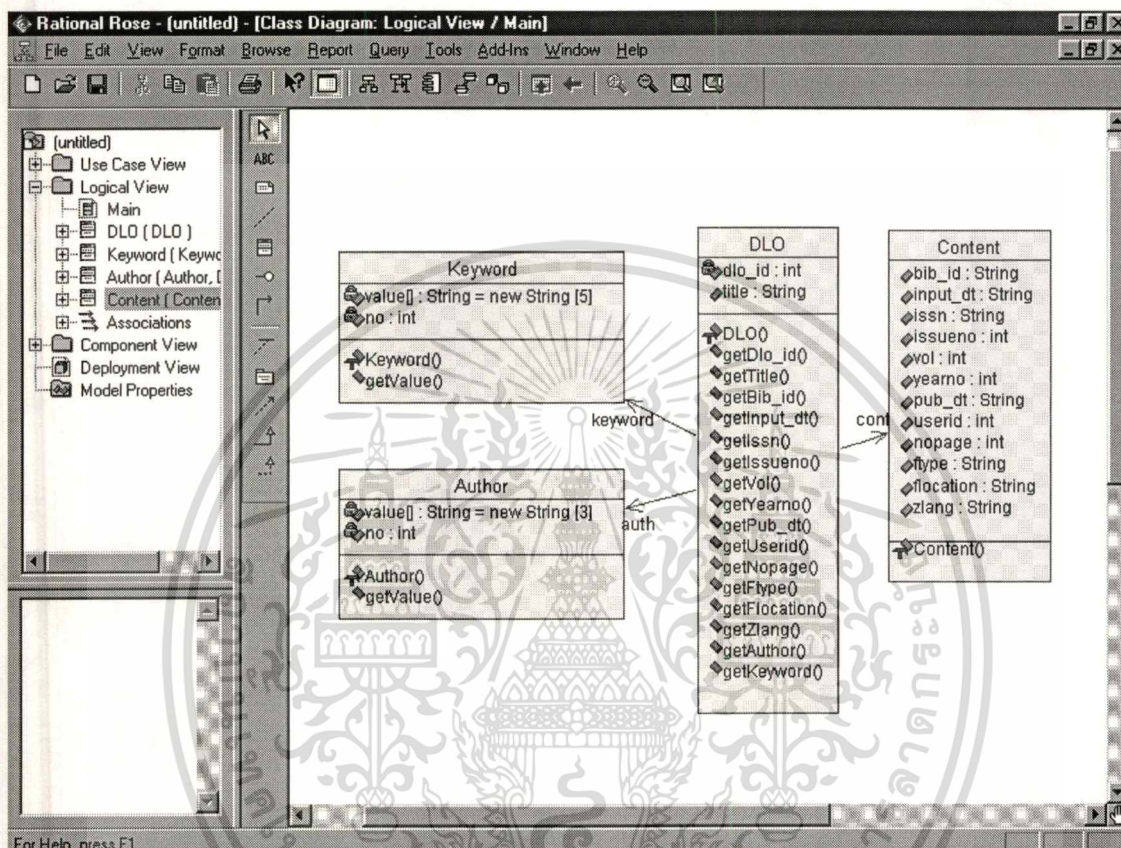
รูปที่ 5-7 วิธีการเลือกเมนูเพื่อทำการ Reverse Engineering



รูปที่ 5-8 ตัวอย่างการเลือกไฟล์จาก Directory มาทำ Reverse Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นสุดท้าย หลังจากได้เลือกไฟล์เรียบร้อยแล้ว ให้กดปุ่ม Reverse โปรแกรมจะทำการ Reverse ออกมาเป็น Class Diagram ที่ต้องการ ดังตัวอย่าง เลือก DLO.java ประกอบไปด้วย 3 Class และ 3 Component เพิ่มมาโดยสังเกตรอง Browser Windows ดังรูปที่ 5-9



รูปที่ 5-9 ตัวอย่าง Model ที่ได้จากการทำ Reverse Engineering

5.3 Rational Rose Oracle8

ในการทำงานเดียวกัน Rational Rose Oracle8 คือส่วนเพิ่มเติมของ Rational Rose ที่มีคุณสมบัติดังต่อไปนี้

1. สามารถสร้าง Object Models จาก Oracle8 Relational Schema และขยายความสามารถไปสู่มุมมองเชิงวัตถุของ Oracle8
2. สามารถทำให้มองเห็นภาพ หรือจำลอง Relational Database ที่มีอยู่เดิม และอำนวยความสะดวกในการค้นหาและประกอบเข้าด้วยกันกับ Business Object ซึ่งจะช่วยลดการลงทุนสำหรับการปรับปรุง Relational Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

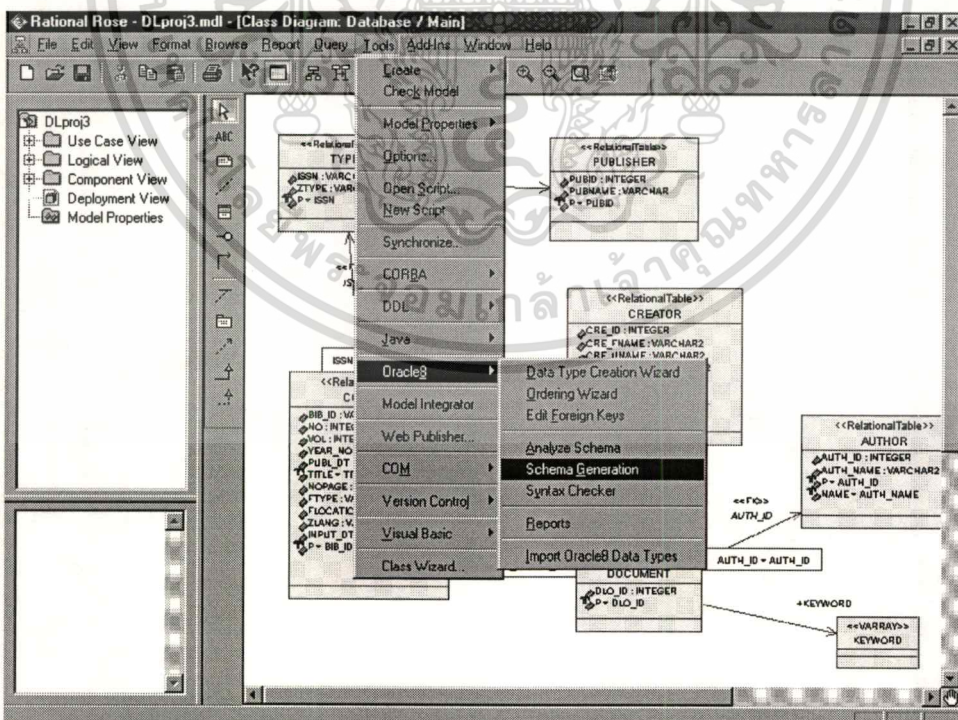
3. สามารถทำให้ผู้พัฒนาระบบสร้างแบบจำลองทางธุรกิจในรูปแบบของภาษาที่ใช้ในการพัฒนาโปรแกรมและฐานข้อมูล รวมทั้งตัวช่วยอัตโนมัติที่นำทางในการสร้าง Object Model ที่พัฒนาจาก Relational ไปสู่ Object-relational โดยสามารถสร้าง Relational Tables, Triggers, Object Types, Object Views, VARRAYs, Nested Tables, และคุณสมบัติเชิงวัตถุต่างๆ ใน Oracle8

5.3.1 การทำ database normalization

การจำลองแบบของข้อมูลสำหรับ Rational Rose Oracle8 ยังจำเป็นต้องผ่านกระบวนการออกแบบฐานข้อมูล และการทำ Normalize ข้อมูลจริงเสียก่อน ในระบบงานนี้ ได้ทำการ Normalize ข้อมูลจริง ดังแสดงในภาคผนวก จากนั้นจึงจะนำตารางที่ได้มาจากการ Normalize มาสร้างเป็น Class Diagram รวมไว้ใน Database Package แล้วจึงจะนำไปสร้างเป็น Database Schema ด้วย Rational Rose Oracle8 ซึ่งจะกล่าวถึงในหัวข้อถัดไป

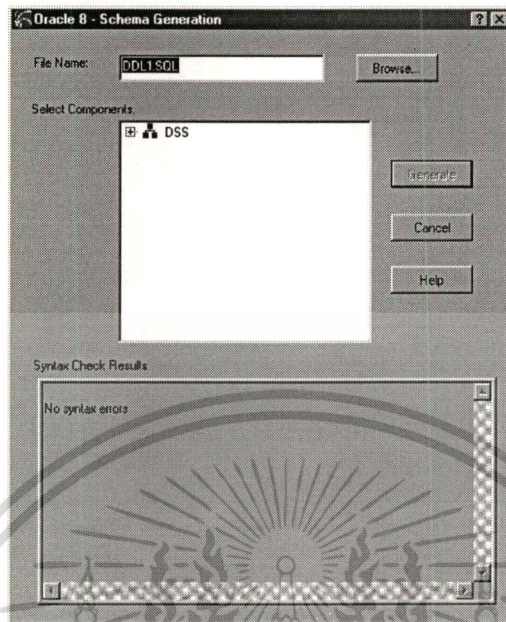
5.3.2 ตัวอย่างการสร้าง database schema ของระบบด้วย Rational Rose Oracle8

1. จาก Class Diagram ใน Database Package เลือก เมนู Tools > Oracle8 > Schema Generation Rational Rose ดังรูปที่ 5-10
2. ได้ผลลัพธ์ดังรูปที่ 5-11 กด Generate จะได้ Source Code (DDL) ดังรูปที่ 5-12

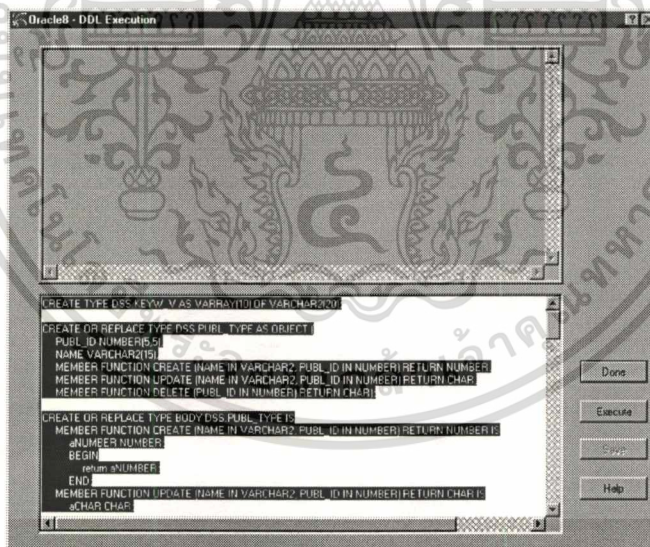


รูปที่ 5-10 Class Diagram ที่จะนำมาสร้าง Database Schema

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-11 หน้าจอของโปรแกรม Rational Rose ขณะทำการ Generate Schema สำหรับ Oracle8



รูปที่ 5-12 หน้าจอผลลัพธ์หลังจากขั้นตอนที่ 1

หมายเหตุ หากต้องการสั่ง Create Database ด้วย Rational Rose ให้กดปุ่ม Execute โดยระบบจะต้องติดตั้ง DBMS เสร็จสมบูรณ์แล้ว แต่ในกรณีที่ไม่ต้องการใช้วิธี Copy Source Code เก็บไว้เพื่อนำไปใช้ Create Database ในภายหลัง ผลลัพธ์ที่ได้ คือ Source Code (Database Definition Language) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE TABLE CREATOR (
    CRE_ID INTEGER NOT NULL UNIQUE,
    CRE_FNAME VARCHAR2(50) NOT NULL UNIQUE,
    CRE_UNAME VARCHAR2(10) NOT NULL UNIQUE,
    CRE_PASSWD VARCHAR2(10),
    CRE_LEVEL INTEGER,
    CRE_EMAIL VARCHAR(30),
    CONSTRAINT P PRIMARY KEY (CRE_ID));

```

ตารางที่ 5-1 DDL ของตาราง Creator ที่ได้จากการสร้างด้วย Rational Rose Oracle8

```

CREATE TABLE PUBLISHER (
    PUB_ID INTEGER NOT NULL UNIQUE,
    PUB_NAME VARCHAR2(50) UNIQUE,
    CONSTRAINT P2 PRIMARY KEY (PUB_ID));

```

ตารางที่ 5-2 DDL ของตาราง Publisher ที่ได้จากการสร้างด้วย Rational Rose Oracle8

```

CREATE TABLE TYPE (
    ISSN VARCHAR2(15) NOT NULL UNIQUE,
    PUB_ID INTEGER,
    ZTYPE VARCHAR2(15),
    CONSTRAINT P3 PRIMARY KEY (ISSN),
    CONSTRAINT PUB_ID FOREIGN KEY(PUB_ID)
REFERENCES PUBLISHER(PUB_ID));

```

ตารางที่ 5-3 DDL ของตาราง Type ที่ได้จากการสร้างด้วย Rational Rose Oracle8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE TABLE CONTENT (
  BIB_ID VARCHAR2(20) NOT NULL UNIQUE,
  ISSN VARCHAR2(15),
  ISSUE_NO INTEGER,
  VOL INTEGER,
  YEAR_NO INTEGER,
  PUB_DATE DATE,
  CRE_ID INTEGER,
  NO_PAGE INTEGER,
  FTYPE VARCHAR2(15),
  FLOCATION VARCHAR2(30),
  ZLANG VARCHAR2(2),
  INPUT_DT DATE,
  CONSTRAINT P4 PRIMARY KEY (BIB_ID),
  CONSTRAINT ISSN FOREIGN KEY (ISSN) REFERENCES TYPE (ISSN),
  CONSTRAINT CRE_ID FOREIGN KEY (CRE_ID) REFERENCES CREATOR (CRE_ID));

```

ตารางที่ 5-4 DDL ของตาราง Content ที่ได้จากการสร้างด้วย Rational Rose Oracle8

```

CREATE TABLE DOCUMENT (
  DLO_ID INTEGER NOT NULL UNIQUE,
  TITLE VARCHAR2(100) NOT NULL UNIQUE,
  KEYWORD1 VARCHAR(30),
  KEYWORD2 VARCHAR(30),
  KEYWORD3 VARCHAR(30),
  KEYWORD4 VARCHAR(30),
  KEYWORD5 VARCHAR(30),
  AUTH1 VARCHAR(50),
  AUTH2 VARCHAR(50),
  AUTH3 VARCHAR(50),
  BIB_ID VARCHAR2(20),
  CONSTRAINT P5 PRIMARY KEY (DLO_ID),
  CONSTRAINT BIB_ID FOREIGN KEY (BIB_ID) REFERENCES CONTENT (BIB_ID));

```

ตารางที่ 5-5 DDL ของตาราง Document ที่ได้จากการสร้างด้วย Rational Rose Oracle8

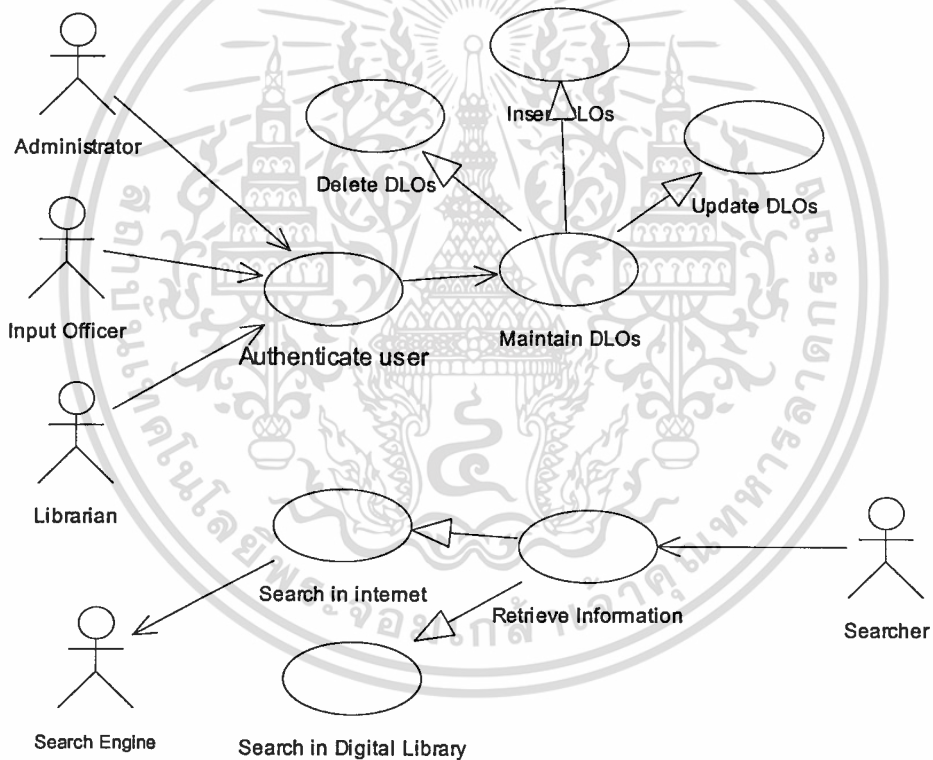
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

DSS Digital Library System Modeling

6.1 Use Case Diagram

เมื่อผ่าน Phase ของการให้คำนิยามต่างๆ ของซอฟต์แวร์ (Definition) แล้วจึงได้ผลลัพธ์การจำลองระบบงานห้องสมุดดิจิทัล ดังรูปที่ 6-1

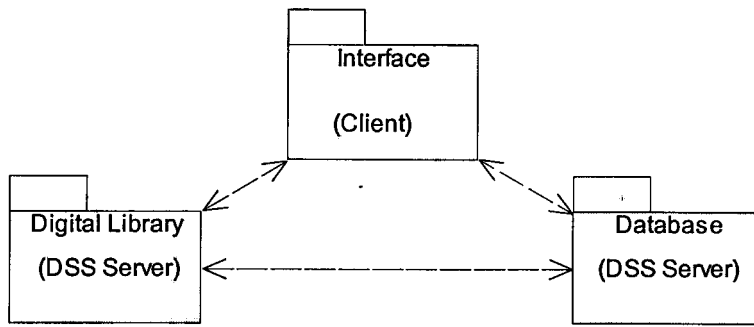


รูปที่ 6-1 Use Case Diagram สำหรับห้องสมุดดิจิทัล

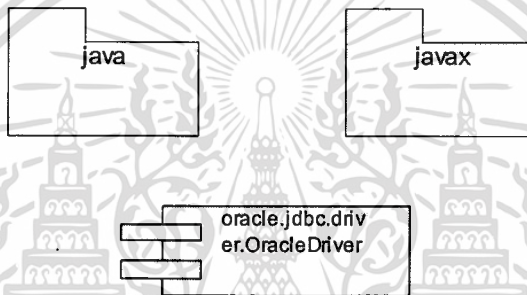
6.2 Component Diagram

เมื่อผ่าน Phase ของการวิเคราะห์ (Analysis) ทำให้สามารถออกแบบ Class Diagram โดย Packages ภายในที่จะต้องสร้างขึ้นเอง มีด้วยกัน 3 Packages ได้แก่ Interface, Digital Library และ Database ส่วน Package ข้างนอกที่นำมาใช้ได้แก่ Java และ Javax ซึ่งมี API ต่างๆ ไว้ให้ใช้งาน ดังรูปที่ 6-2 และ 6-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-2 Package ภายในทั้งหมด (Logical View)



รูปที่ 6-3 Package ภายนอกที่นำเข้ามาใช้ (Logical View)

6.2.1 Component Diagram ภายใน Interface package

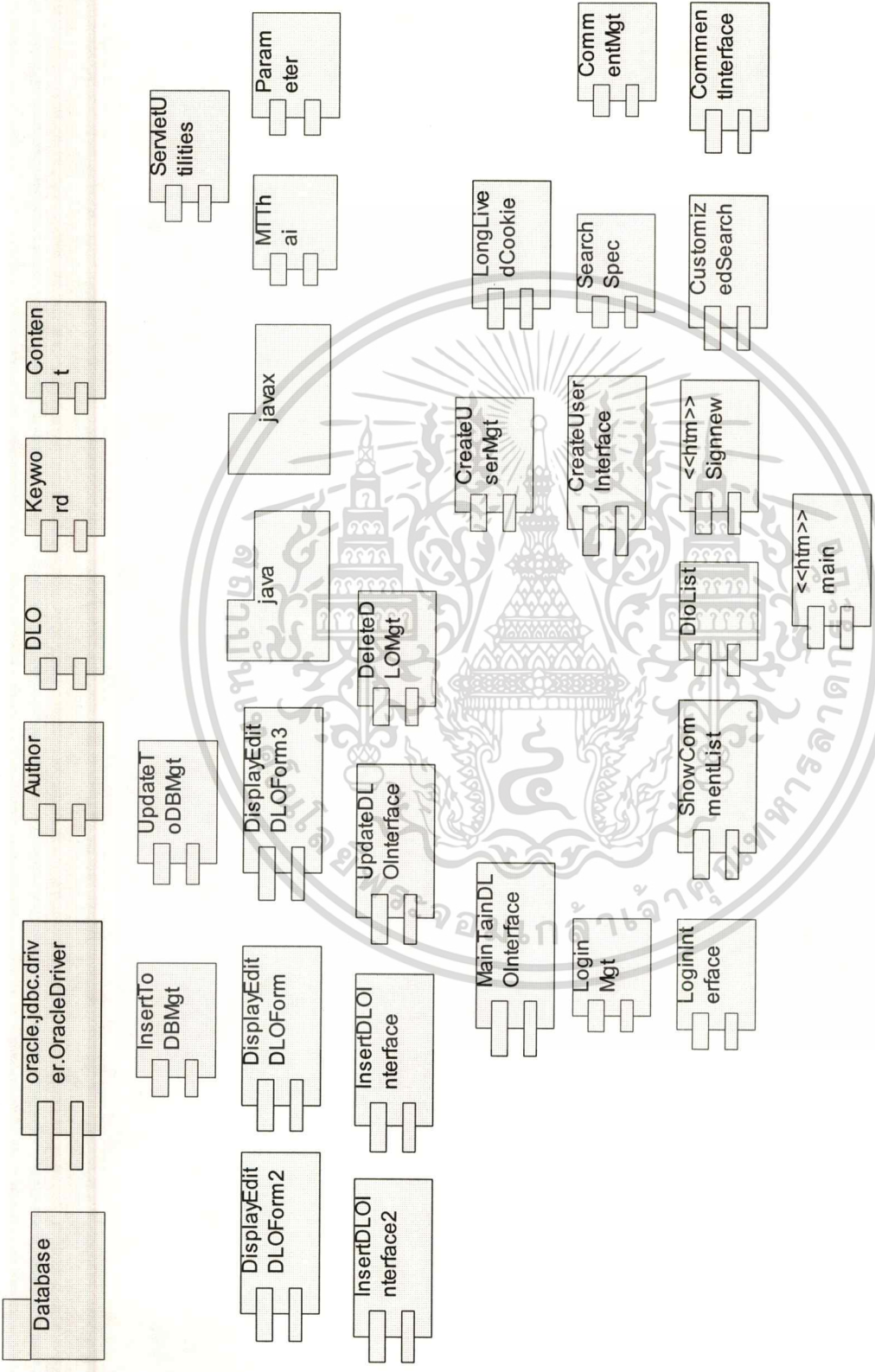
ในส่วน Interface จะมีด้วยกัน 2 Component คือ main.htm และ signnew.htm ซึ่งทำหน้าที่เป็น Interface อยู่ที่เครื่อง Client ดังรูปที่ 6-4



รูปที่ 6-4 Component Diagram ภายใน Interface package

6.2.2 Component Diagram ภายใน Digital Library package

ในส่วน Digital Library (ส่วนของ Business Logic ทั้งหมดจะรันที่ฝั่ง Server ด้วย Java Servlet API แล้วส่งผลลัพธ์ในรูปแบบ HTML ไปยัง Client ประกอบด้วย Component ต่างๆ ดังแสดงในรูปที่ 6-5



รูปที่ 6-5 แผนภาพ Component Diagram ภายใน Digital Library Package

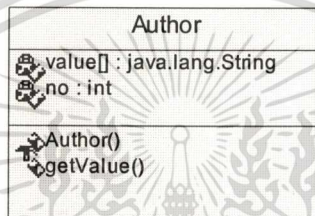
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 Class Diagram

เมื่อผ่านขั้นวิเคราะห์ (Analysis) เข้าสู่ขั้นออกแบบ (Design) ทำให้สามารถออกแบบ Class Diagram สำหรับอธิบายรายละเอียดของ Class ภายในโปรแกรมที่ต้องสร้างขึ้น ได้แก่ Digital Library Package และ Database Package ดังนี้

6.3.1 Class Diagram ภายใน Digital Library Package

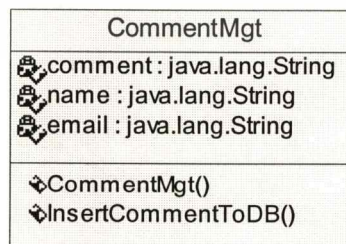
ประกอบด้วย 29 Class ดังรูปที่ 6-6 จนถึง รูปที่ 6-34 โดยการเรียงตามลำดับตัวอักษรของชื่อ Class



รูปที่ 6-6 Class Author



รูปที่ 6-7 Class Comment Interface



รูปที่ 6-8 Class CommentMgt

Content
↗ bib_id : java.lang.String ↗ input_dt : java.lang.String ↗ issn : java.lang.String ↗ issueno : int ↗ vol : int ↗ yeamo : int ↗ pub_dt : java.lang.String ↗ userid : int ↗ nopage : int ↗ ftype : java.lang.String ↗ flocation : java.lang.String ↗ zlang : java.lang.String
↗ Content()

รูปที่ 6-9 Class Content

CreateUserInterface
↗ \$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl" ↗ \$ DB_LOGIN : java.lang.String = "system" ↗ \$ DB_PASSWD : java.lang.String = "manager" ↗ \$ LEVEL : java.lang.String = "0" ↗ fname : java.lang.String ↗ lname : java.lang.String ↗ uname : java.lang.String ↗ passwd : java.lang.String ↗ email : java.lang.String ↗ level : java.lang.String
↗ CreateUserInterface() ↗ DisplayOutput() ↗ doGet() ↗ doPost()

รูปที่ 6-10 Class CreateUserInterface

CreateUserMgt
↗ fname : java.lang.String ↗ lname : java.lang.String ↗ passwd : java.lang.String ↗ email : java.lang.String ↗ level : java.lang.String
↗ CreateUserMgt() ↗ InsertUserToDB()

รูปที่ 6-11 Class CreateUserMgt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CustomizedSearch
↻CustomizedSearch() ↻doGet() ↻doPost()

รูปที่ 6-12 Class CustomizedSearch

DeleteDLOMgt
↻\$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl" ↻\$ DB_LOGIN : java.lang.String = "system" ↻\$ DB_PASSWD : java.lang.String = "manager" ↻\$ uname : java.lang.String ↻\$ level : java.lang.String ↻\$ id : java.lang.String ↻\$ dloid : java.lang.String ↻\$ bibid : java.lang.String ↻\$ out : java.io.PrintWriter ↻err : boolean ↻err1 : boolean ↻errtype : java.lang.String
↻DeleteDLOMgt() ↻DeleteFromDB() ↻doGet() ↻doPost()

รูปที่ 6-13 Class DeleteDLOMgt

DisplayEditDLOForm
↻\$ p[] : java.lang.String ↻\$ _YEAR : int = 2538
↻DisplayEditDLOForm() ↻DisplayContent() ↻DisplayDocument() ↻DisplayListBox() ↻DisplayListBoxLookup() ↻DocumentMetadata() ↻GetNewID() ↻InitParameter()

รูปที่ 6-14 Class DisplayEditDLOForm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

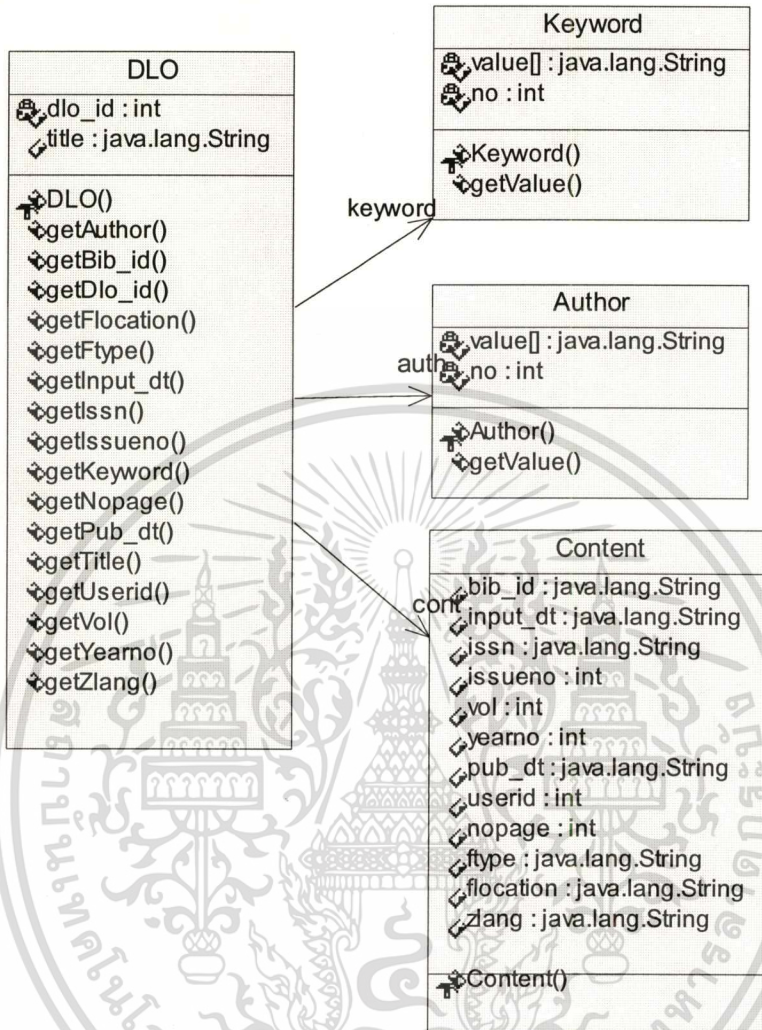
DisplayEditDLOForm2
☛ \$ _YEAR : int = 2538
<ul style="list-style-type: none"> ☛ DisplayEditDLOForm2() ☛ DisplayContent() ☛ DisplayDocument() ☛ DisplayListBox() ☛ DisplayListBoxLookup() ☛ DocumentMetadata() ☛ GetNewID() ☛ InitParameter()

รูปที่ 6-15 Class DisplayEditDLOForm2

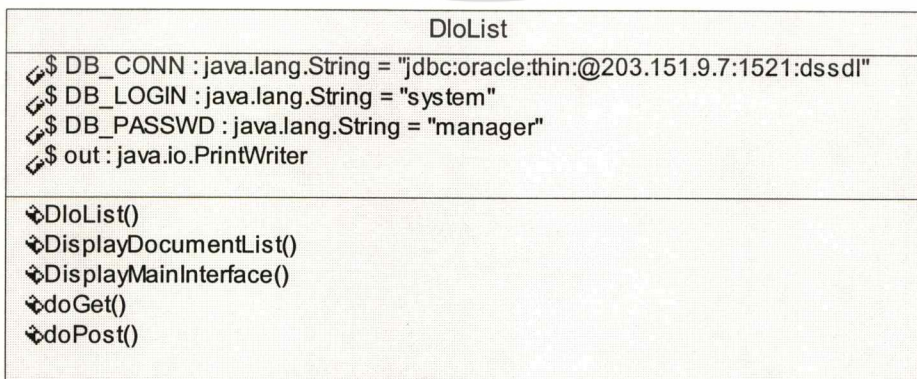
DisplayEditDLOForm3
☛ \$ _YEAR : int = 2538
☛ \$ a : java.lang.String
<ul style="list-style-type: none"> ☛ DisplayEditDLOForm3() ☛ DisplayContent() ☛ DisplayDocument() ☛ DisplayListBox() ☛ DisplayListBoxLookup() ☛ DocumentMetadata() ☛ GetAllDocInfo()

รูปที่ 6-16 Class DisplayEditDLOForm3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-17 Class DLO และความสัมพันธ์กับ Class อื่นๆ



รูปที่ 6-18 Class DloList

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DssSearch	
☞	\$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl"
☞	\$ DB_LOGIN : java.lang.String = "system"
☞	\$ DB_PASSWD : java.lang.String = "manager"
☞	query : java.lang.String
☞	out : java.io.PrintWriter
☞	DssSearch()
☞	DisplayDocumentList()
☞	DisplayMainInterface()
☞	doGet()
☞	doPost()

รูปที่ 6-19 Class DssSearch

InsertDLOInterface	
☞	\$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl"
☞	\$ DB_LOGIN : java.lang.String = "system"
☞	\$ DB_PASSWD : java.lang.String = "manager"
☞	uname : java.lang.String
☞	level : java.lang.String
☞	id : java.lang.String
☞	out : java.io.PrintWriter
☞	InsertDLOInterface()
☞	DisplayMainInterface()
☞	doGet()
☞	doPost()

รูปที่ 6-20 Class InsertDLOInterface

InsertDLOInterface2	
☞	\$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl"
☞	\$ DB_LOGIN : java.lang.String = "system"
☞	\$ DB_PASSWD : java.lang.String = "manager"
☞	uname : java.lang.String
☞	level : java.lang.String
☞	id : java.lang.String
☞	out : java.io.PrintWriter
☞	InsertDLOInterface2()
☞	DisplayMainInterface()
☞	doGet()
☞	doPost()

รูปที่ 6-21 Class InsertDLOInterface2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InsertToDBMgt
DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl" DB_LOGIN : java.lang.String = "system" DB_PASSWD : java.lang.String = "manager" uname : java.lang.String level : java.lang.String id : java.lang.String out : java.io.PrintWriter err : boolean err1 : boolean errtype : java.lang.String
InsertToDBMgt() InsertToDB() doGet() doPost()

รูปที่ 6-22 Class InsertToDBMgt

Keyword
value[] : java.lang.String no : int
Keyword() getValue()

รูปที่ 6-23 Class Keyword

LoginInterface
DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl" DB_LOGIN : java.lang.String = "system" DB_PASSWD : java.lang.String = "manager" uname : java.lang.String password : java.lang.String
LoginInterface() DisplayOutput() doGet() doPost()

รูปที่ 6-24 Class LoginInterface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LoginMgt
⚡\$ utype : java.lang.String ⚡\$ err : java.lang.String ⚡\$ id : java.lang.String
⚡LoginMgt() ⚡Verify()

รูปที่ 6-25 Class LoginMgt

LongLivedCookie
⚡\$ SECONDS_PER_YEAR : int = 31536000
⚡LongLivedCookie()

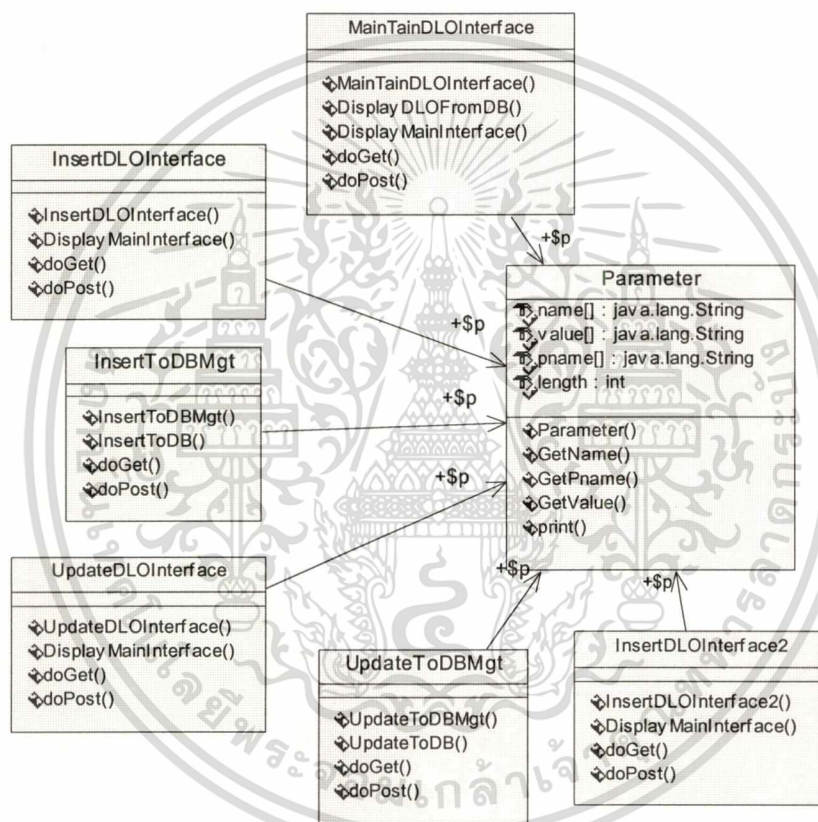
รูปที่ 6-26 Class LongLivedCookie

MainTainDLOInterface
⚡\$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl" ⚡\$ DB_LOGIN : java.lang.String = "system" ⚡\$ DB_PASSWD : java.lang.String = "manager" ⚡\$ uname : java.lang.String ⚡\$ level : java.lang.String ⚡\$ id : java.lang.String ⚡\$ out : java.io.PrintWriter
⚡MainTainDLOInterface() ⚡DisplayDLOFromDB() ⚡DisplayMainInterface() ⚡doGet() ⚡doPost()

รูปที่ 6-27 Class MainTainDLOInterface



รูปที่ 6-28 Class MTThai



รูปที่ 6-29 Class Parameter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SearchSpec
name : java.lang.String baseURL : java.lang.String numResultsSuffix : java.lang.String
SearchSpec() getCommonSpecs() getName() makeURL()

รูปที่ 6-30 Class SearchSpec

ServletUtilities
\$ DOCTYPE : String = "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">"
ServletUtilities() GetAllParameter() GetAllParameter2() GetAllParameter3() GetUserParameter() InputValidate() PrintError() endWithError() endWithLink() headWithTitle() headerWithUserInfo() printBackLink()

รูปที่ 6-31 Class ServletUtilities

ShowCommentList
ShowCommentList() doGet() doPost()

รูปที่ 6-32 Class ShowCommentList

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UpdateDLOInterface	
✖	\$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl"
✖	\$ DB_LOGIN : java.lang.String = "system"
✖	\$ DB_PASSWD : java.lang.String = "manager"
✖	\$ uname : java.lang.String
✖	\$ level : java.lang.String
✖	\$ id : java.lang.String
✖	\$ dloid : java.lang.String
✖	\$ bibid : java.lang.String
✖	\$ out : java.io.PrintWriter
✖	UpdateDLOInterface()
✖	DisplayMainInterface()
✖	doGet()
✖	doPost()

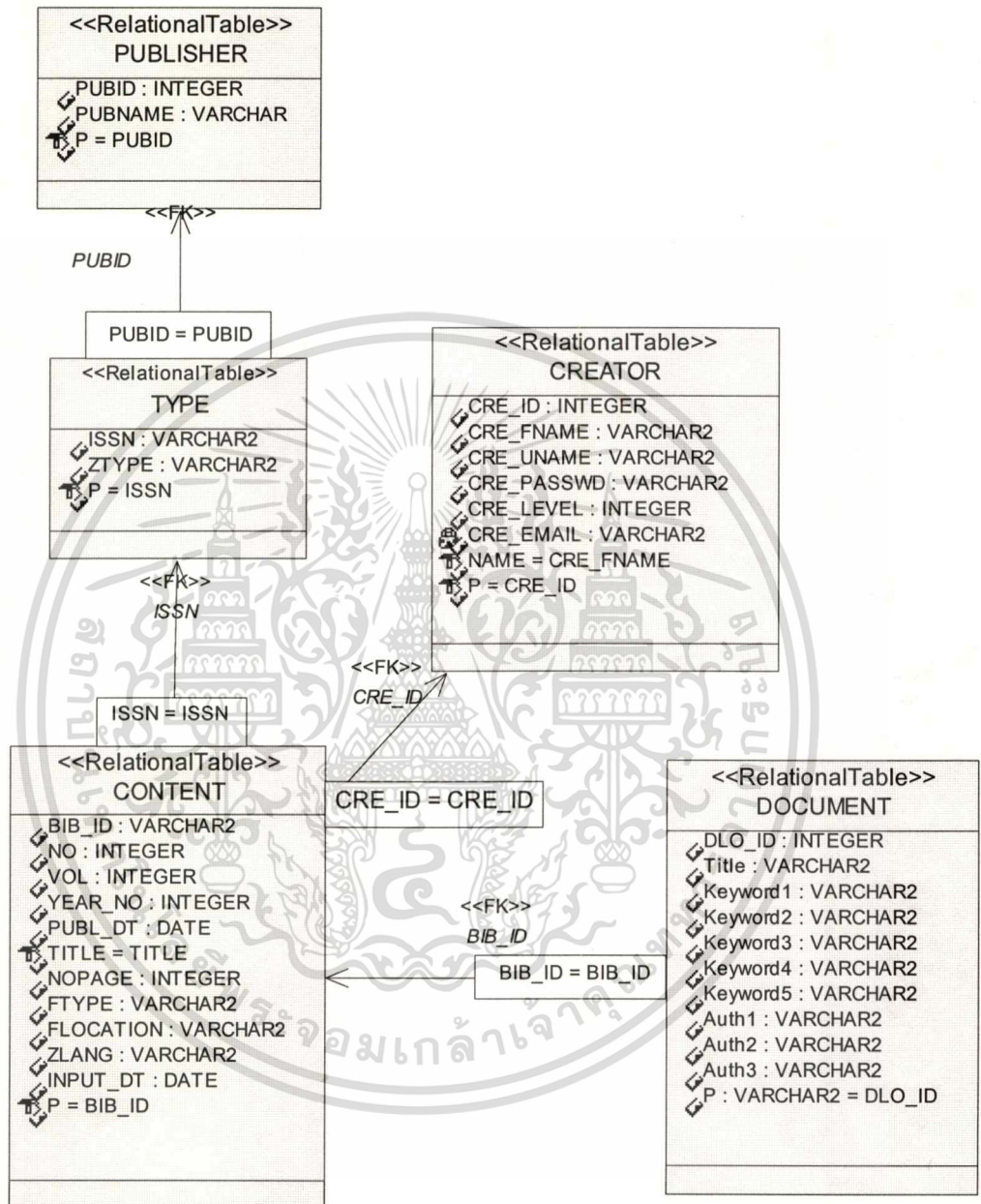
รูปที่ 6-33 Class UpdateDLOInterface

UpdateToDBMgt	
✖	\$ DB_CONN : java.lang.String = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl"
✖	\$ DB_LOGIN : java.lang.String = "system"
✖	\$ DB_PASSWD : java.lang.String = "manager"
✖	\$ uname : java.lang.String
✖	\$ level : java.lang.String
✖	\$ id : java.lang.String
✖	\$ out : java.io.PrintWriter
✖	err : boolean
✖	err1 : boolean
✖	errtype : java.lang.String
✖	UpdateToDBMgt()
✖	UpdateToDB()
✖	doGet()
✖	doPost()

รูปที่ 6-34 Class UpdateToDBMgt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.2 Class Diagram ภายใน Database package



รูปที่ 6-35 Class diagram ใน Database Package

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 Sequence Diagram

เมื่อผ่าน Phase ของการออกแบบ (Design) และมีความต้องการอธิบายความสัมพันธ์ระหว่าง Use Case และ Class Diagram ทำให้สามารถออกแบบ Sequence Diagram 5 Diagram ได้แก่ Authenticate User, Insert DLOs, Update DLOs, Delete DLOs และ Search (DLO ย่อมาจาก Document Like Object ใช้เรียกเอกสารอิเล็กทรอนิกส์)

6.4.1 Authenticate user

เป็นการยืนยันการเข้าใช้งานระบบ เมื่อ Admin, Input Officer, Librarian เข้าใช้งานระบบผ่าน main.htm รายละเอียดดังรูปที่ 6-36

6.4.2 Insert DLOs

เป็นการรับข้อมูล DLO มาจาก Admin , Input Officer, Librarian แล้วเรียกใช้ Object ต่างๆ ให้ทำงานเพิ่มเติมข้อมูลลงในฐานข้อมูล รายละเอียดดังรูปที่ 6-37

6.4.3 Update DLOs

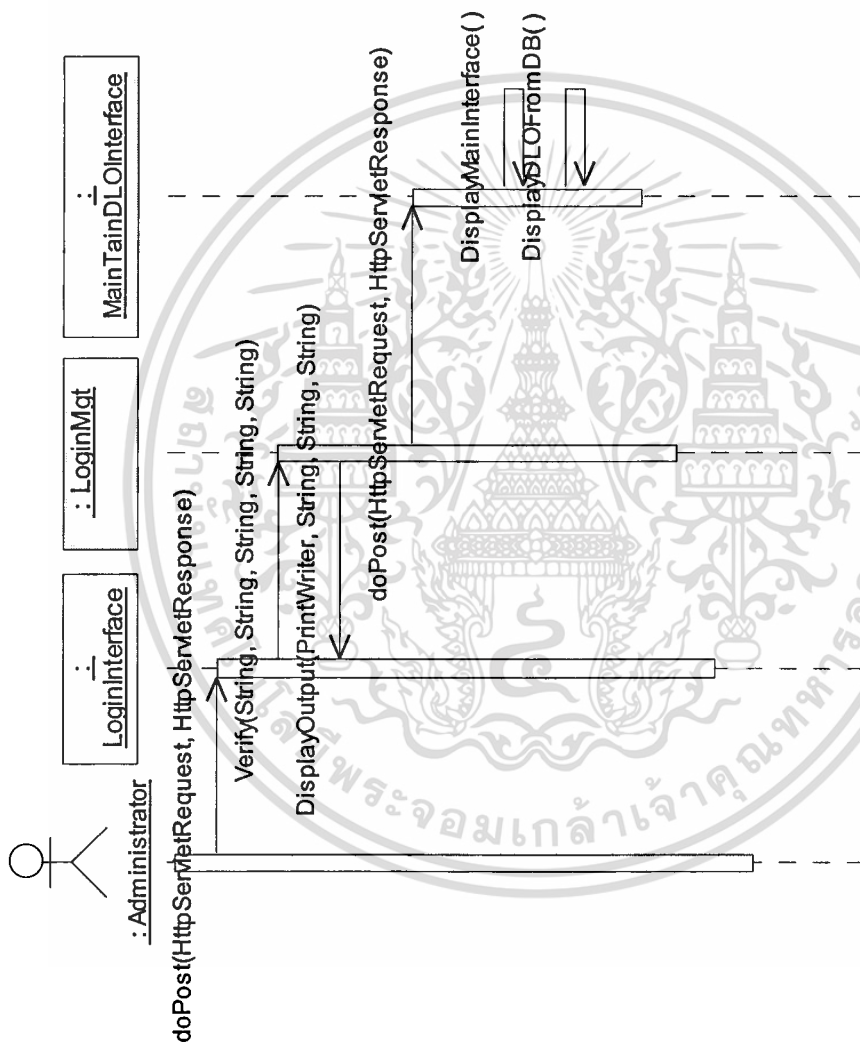
เป็นการรับข้อมูล DLO มาจาก Admin , Input Officer, Librarian แล้วแก้ไขข้อมูลที่มีอยู่แล้วเสียใหม่ และบันทึกลงฐานข้อมูล รายละเอียดดังรูปที่ 6-38

6.4.4 Delete DLOs

เป็นการรับความต้องการลบ DLO มาจาก Admin , Input Officer, Librarian แล้วเรียกใช้ Object ต่างๆ ให้ทำงานลบข้อมูลที่ต้องการ (ใช้ DLO เป็นตัวชี้) ออกจากฐานข้อมูล รายละเอียดดังรูปที่ 6-39

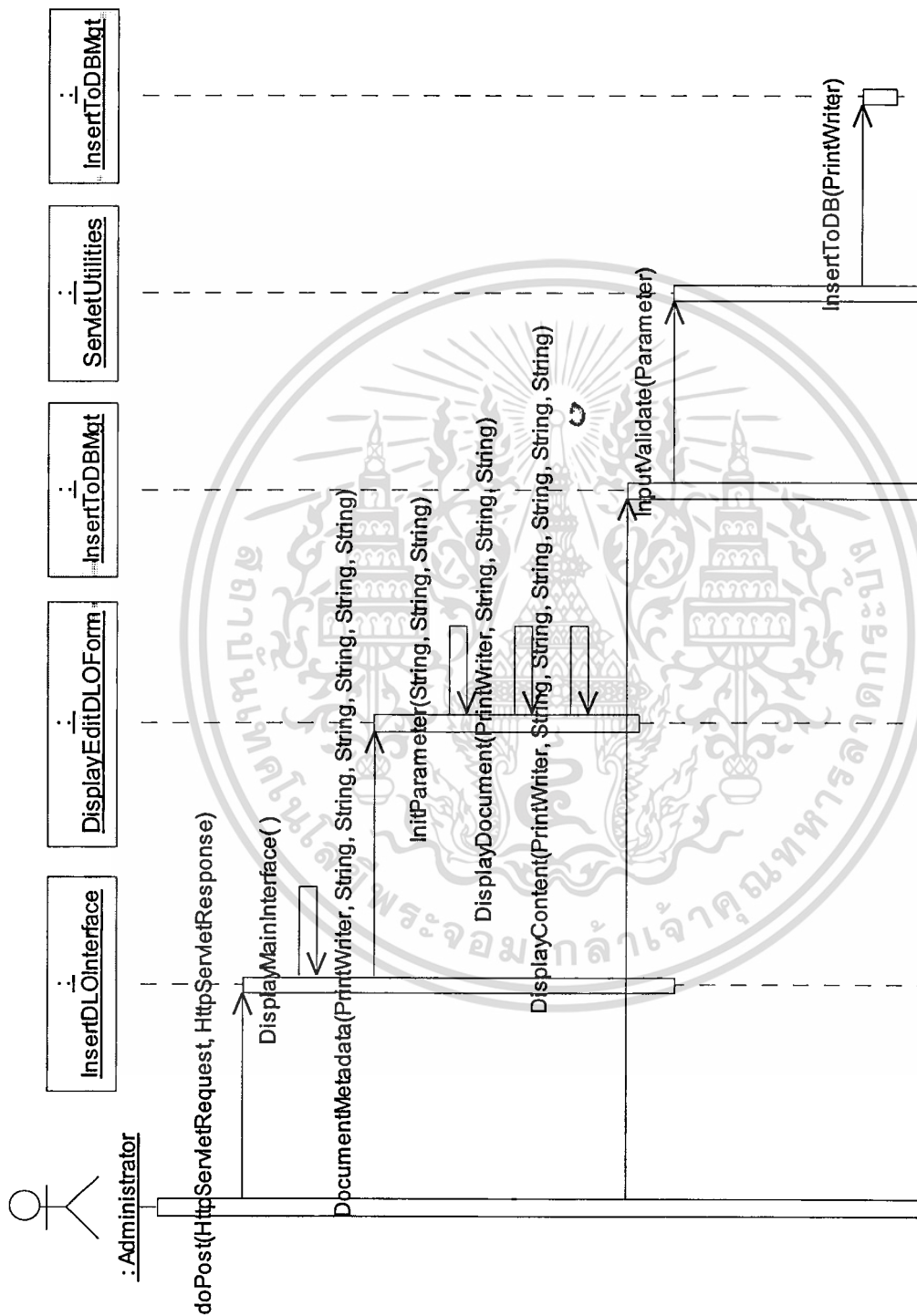
6.4.5 Search

เป็นการรับข้อมูล DLO มาจาก Searcher เข้ามาค้นข้อมูลโดยส่งคำค้น (Query String) เข้ามากระตุ้นให้ Object ไปค้นหาข้อมูลจากฐานข้อมูลแล้วแสดงผล รายละเอียดดังรูปที่ 6-40

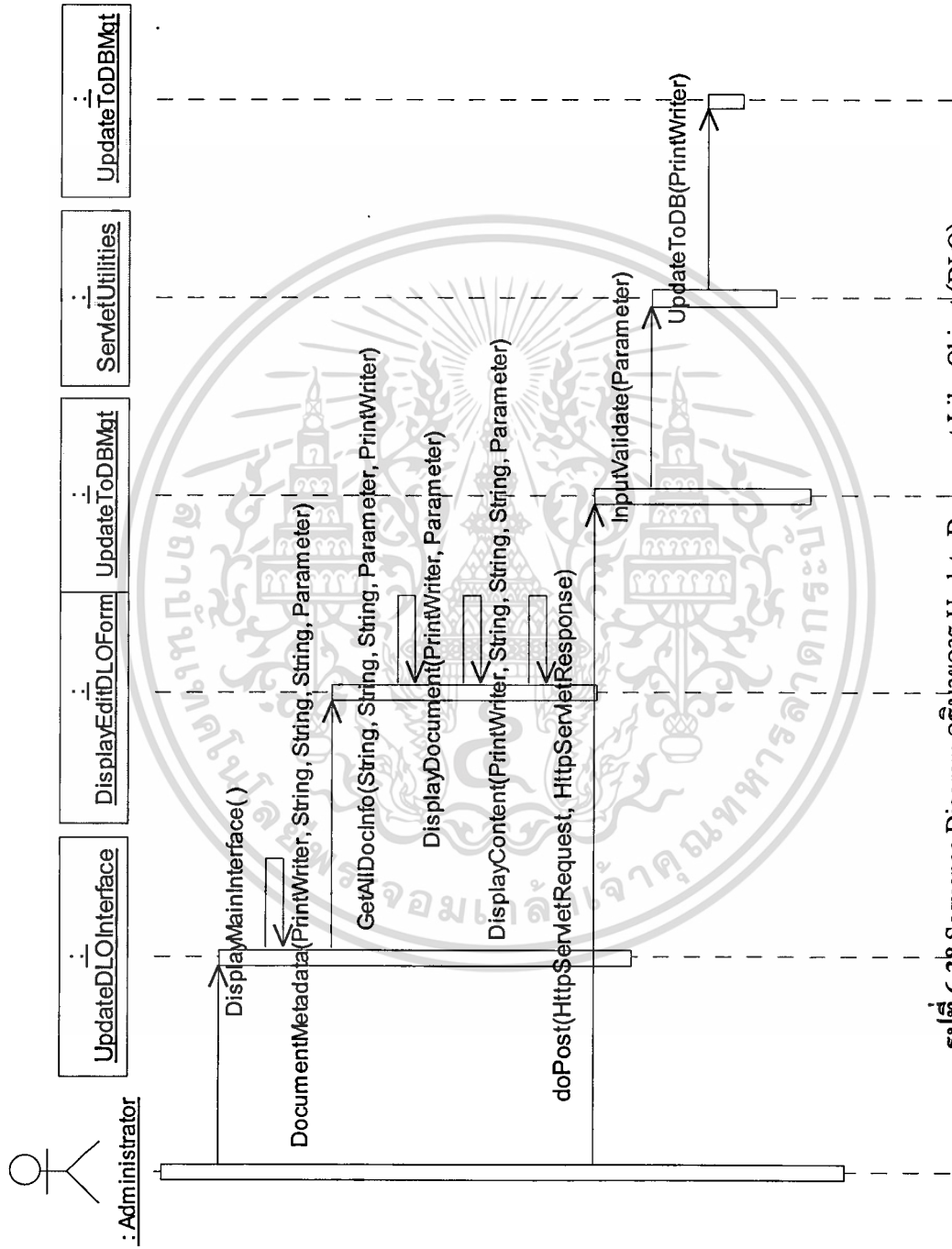


รูปที่ 6-36 Sequence Diagram อธิบายการ Authenticate User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

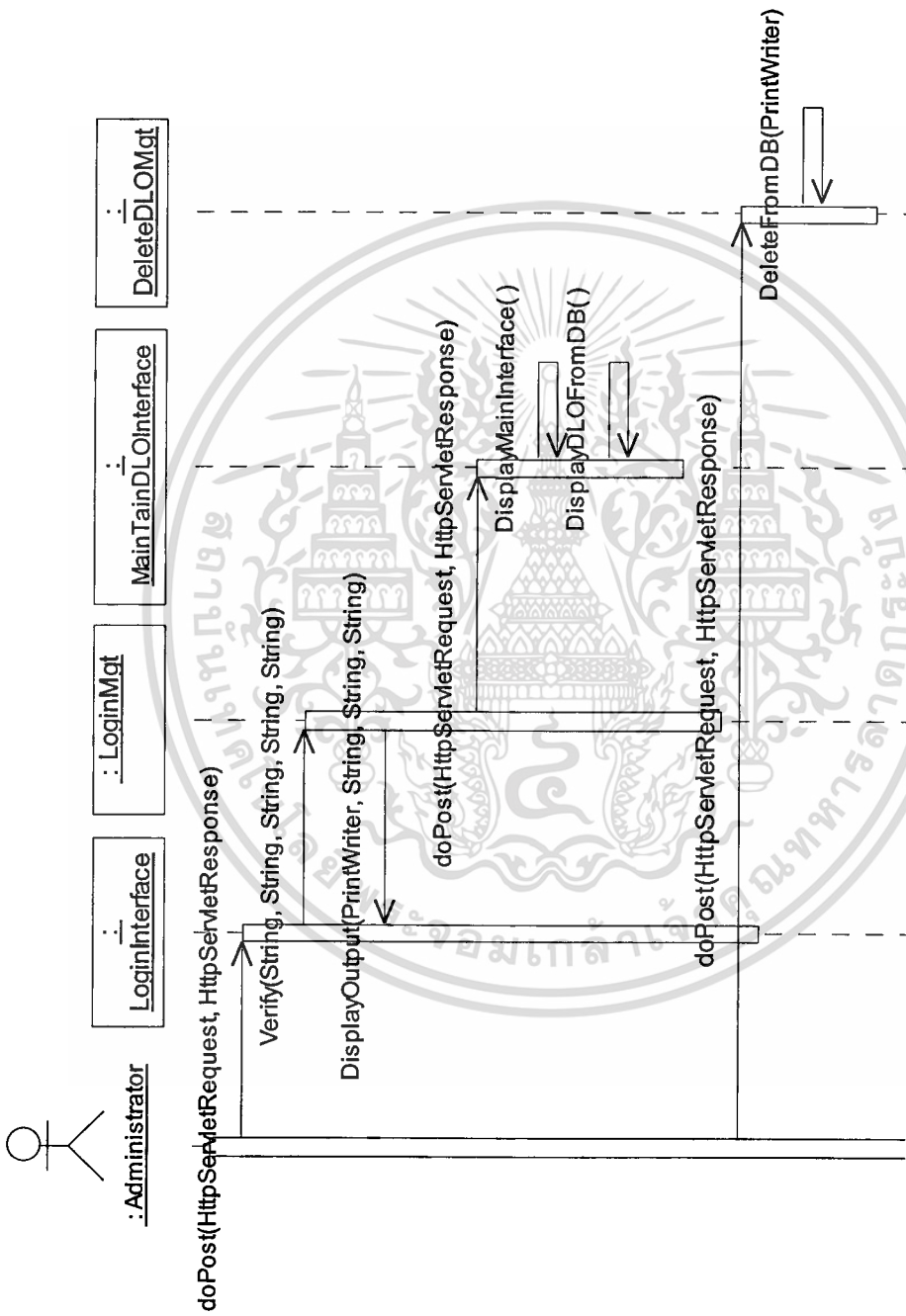


รูปที่ 6-37 Sequence Diagram ธิบายการ Insert Document Like Object (DLO)



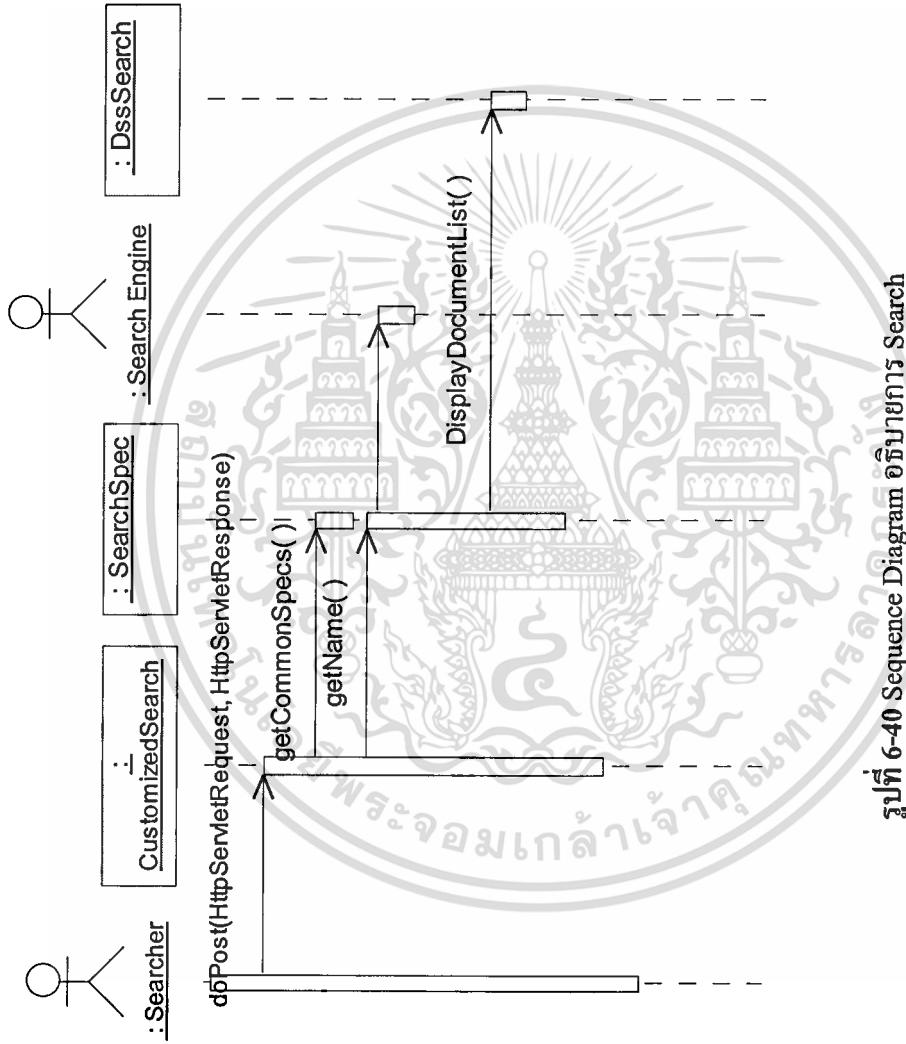
รูปที่ 6-38 Sequence Diagram อธิบายการ Update Document Like Object (DLO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-39 Sequence Diagram ธิบายการ Delete Document Like Object (DLO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-40 Sequence Diagram อธิบายการ Search

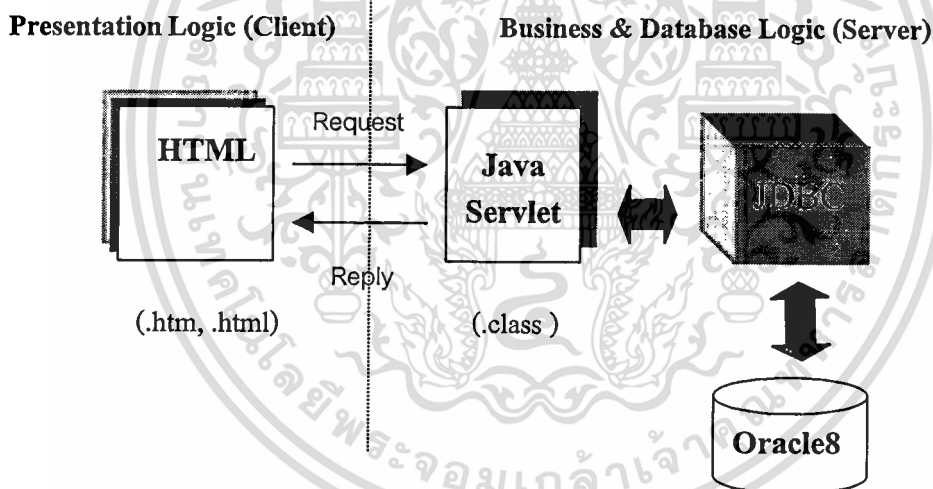
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สถาปัตยกรรม และเครื่องมือที่ใช้ในการพัฒนาโปรแกรม

7.1 สถาปัตยกรรมของระบบ

สำหรับ Architecture ของระบบจะเป็นแบบ 2-tier Client-Server นั่นคือส่วน Presentation จะอยู่ที่ Client ในขณะที่ Business Logic และ Database จะอยู่ที่ Server และเป็นลักษณะของ WWW-based Application ที่เรียกว่า Server Side โดยทำการประมวลผลจะทำที่ Server และส่งผลลัพธ์ไปที่ Client (WWW Browser) ดังแสดงได้ในรูปที่ 7-1



รูปที่ 7-1 สถาปัตยกรรมของระบบ

7.2 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม

ภาษาที่เลือกมาใช้ในการพัฒนาระบบงาน คือ ภาษา Java ซึ่งเป็น Object-oriented Programming Language โดยใช้ JDK (Java Development Kits) รุ่น 1.2 ที่ถูกสนับสนุนโดย UML และใช้ “Java Servlet” ซึ่งเป็น API อยู่ภายใน Javax Package สนับสนุนการพัฒนาแบบ WWW-based Application ชนิด Server Side และทำการติดต่อกับระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object Relational Database Management System) คือ Oracle8i Enterprise Edition for Microsoft WindowsNT 4.0 ผ่าน Oracle Java Database Connectivity (JDBC) รุ่น 12.01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

สรุปผลการศึกษา

โครงการพัฒนาระบบงานห้องสมุดดิจิทัลโดยใช้ UML มีผลสำเร็จดังแสดงในตัวอย่างการใช้งานระบบห้องสมุดดิจิทัลในภาคผนวก โดยโครงการที่จัดทำขึ้นนี้จะถูกนำไปใช้งานจริง โดยการนำเข้าข้อมูล ทดสอบการใช้งาน และใช้งานต่อไป หลังจากนั้นจะกลับเข้าสู่กระบวนการบำรุงรักษาระบบงานต่อไป โดยมีแนวทางจะพัฒนาระบบงานต่อ ดังนี้

1. เพิ่มเติมส่วนประวัติการค้นหาข้อมูลสำหรับผู้เข้าใช้ห้องสมุด
2. เพิ่มระบบรักษาความปลอดภัยให้ดียิ่งขึ้น
3. เพิ่มเติมส่วนเชื่อมโยงข้อมูลไปยังแหล่งข้อมูลอื่นที่มีประโยชน์
4. พัฒนาระบบค้นหาให้ฉลาดขึ้นด้วยการปรับปรุงวิธีค้นหาข้อมูลและการแสดงผล

ในการศึกษาเรื่องห้องสมุดดิจิทัล กับการพัฒนาด้วย UML ทำให้ ผู้จัดทำโครงการมีความเชี่ยวชาญในการใช้ UML และ Rational Rose เป็นเครื่องมือในการวิเคราะห์และออกแบบระบบงานในระดับหนึ่ง โดยมีความเห็นว่าน่าจะสามารนำมาใช้เป็นเครื่องมือช่วยในการพัฒนาระบบงานได้เป็นอย่างดี เพราะสามารถช่วยลดระยะเวลาการทำงานได้ ดังนั้น UML จึงเป็นมาตรฐานที่มีความน่าสนใจ และเหมาะสมกับการพัฒนาระบบงานสารสนเทศต่อไปในอนาคต

บรรณานุกรม

- [1] น้าทิพย์ วิชาวิน. 2542. ห้องสมุดยุคใหม่กับไอที : Library Automation System & Digital Library. กรุงเทพฯ. : โรงพิมพ์แอลเอส.
- [2] พงศธร สุกุทธธรรม. 2542. “Trend in Object-oriented Technology.” กรุงเทพฯ : Infowave Thailand. เอกสารอค์สำเนา
- [3] ศิริลักษณ์ วัฒนกิจอำนวย. 2537. SQL. กรุงเทพฯ : ดอกหญ้า.
- [4] AAJC Thailand. 2000 Fundamental of OOP. [Online] Available :
<http://aajc.au.ac.th/courses/oop/oop.htm>.
- [5] Apache. 2000 The Jakarta Project. [Online]. Available : <http://jakarta.apache.org>.
- [6] Arms, Y. 1995. Key Concepts in Architecture of the Digital Library. [Online]. Available :
<http://www.cnri.veston.va.us/home/dlib/July96107/arms.html>.
- [7] Booch G. et. Al. 1997. UML Distilled Applying the Standard and Object Modeling Language. : Addison-Wesley.
- [8] CDG System. 2000. Object-oriented Knowledge. [Online]. Available :
<http://www.cdg.co.th/gcc/html/oo-knowledge.html>
- [9] Date, C. 1995. An Introduction to Database Systems. 6th ed. : Addison-Wesley.
- [10] Dorsey, Paul and Hudicka, R. 1999. Oracle : Design using Object Modeling. California : McGraw-Hill.
- [11] Jacobson, I. 1996. “Applying UML in the Unified Process.” Bangkok : Rational Software. Presentation Document.
- [12] Maty Hall. 2000 A Tutorial on Java Servlets and Java Server Pages (JSP). [Online] Available : <http://www.apl.jhu.edu/%7Ehall/java/Servlet-Tutorial/>.
- [13] Oracle. 1999 “Oracle8 and Java.” Bangkok : Oracle Oracle Technical White Paper.
- [14] Pressman, S. 1997. Software Engineering : A Practitioner’s approach. 4th ed. Singapore : McGraw-Hill.

- [15] Quatrani, T. 1998. Visual Modeling with Rational Rose and UML. Massachusetts : Addison Wesley Longman.
- [16] Rational Software. 2000 Rational Rose 2000e Using Rose J. [Online]. Available : http://www.rational.com/support/document/manuals/doccet149/Rational%20Rose/Documentation/Rose_java.pdf.
- [17] Sibershatz, A. et. Al. 1997. Database System Concepts. 3rd ed. Singapore : McGraw-Hill.
- [18] Wiley, L. 2000 Digital Libraries : design, development and manangement. [Online]. Available : <http://www.digitallarningspace.com>.

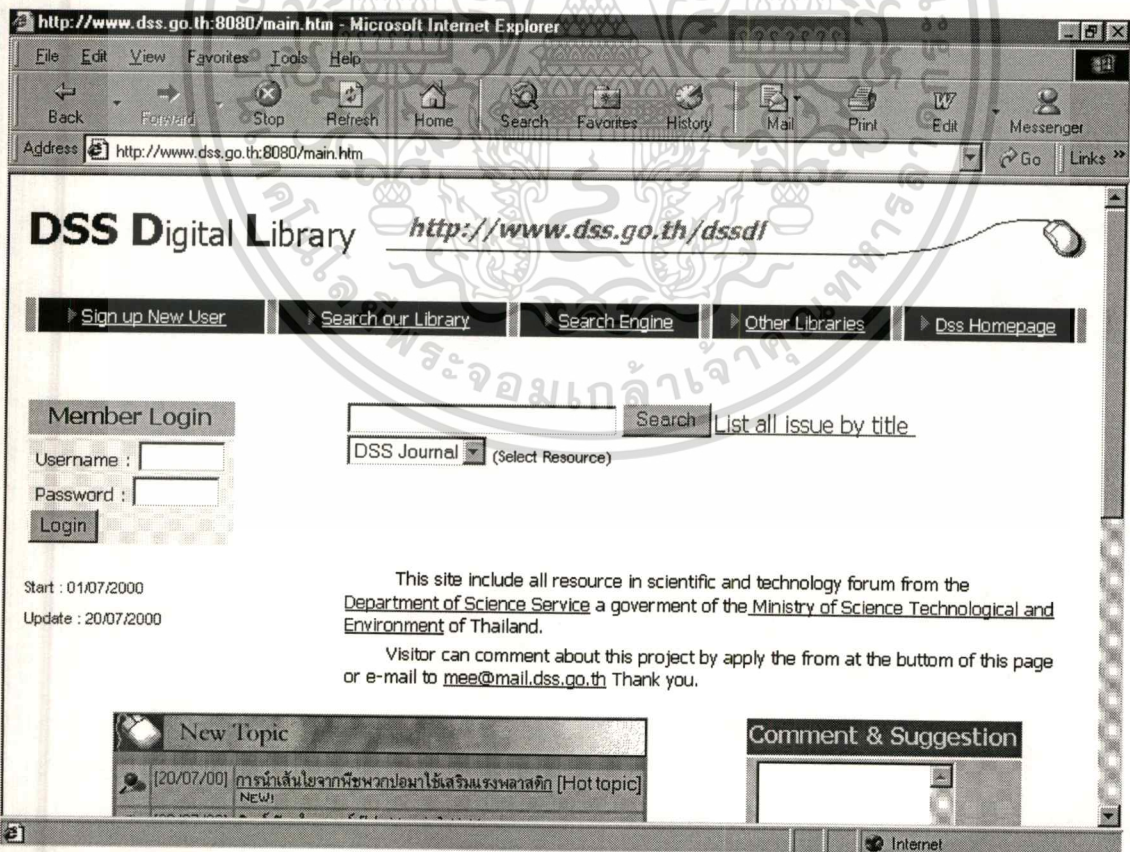


ภาคผนวก

ก. ตัวอย่างการใช้ห้องสมุดดิจิทัล

หน้าจอที่ 1: หน้าแรกห้องสมุดดิจิทัลกรมวิทยาศาสตร์บริการ ประกอบด้วย

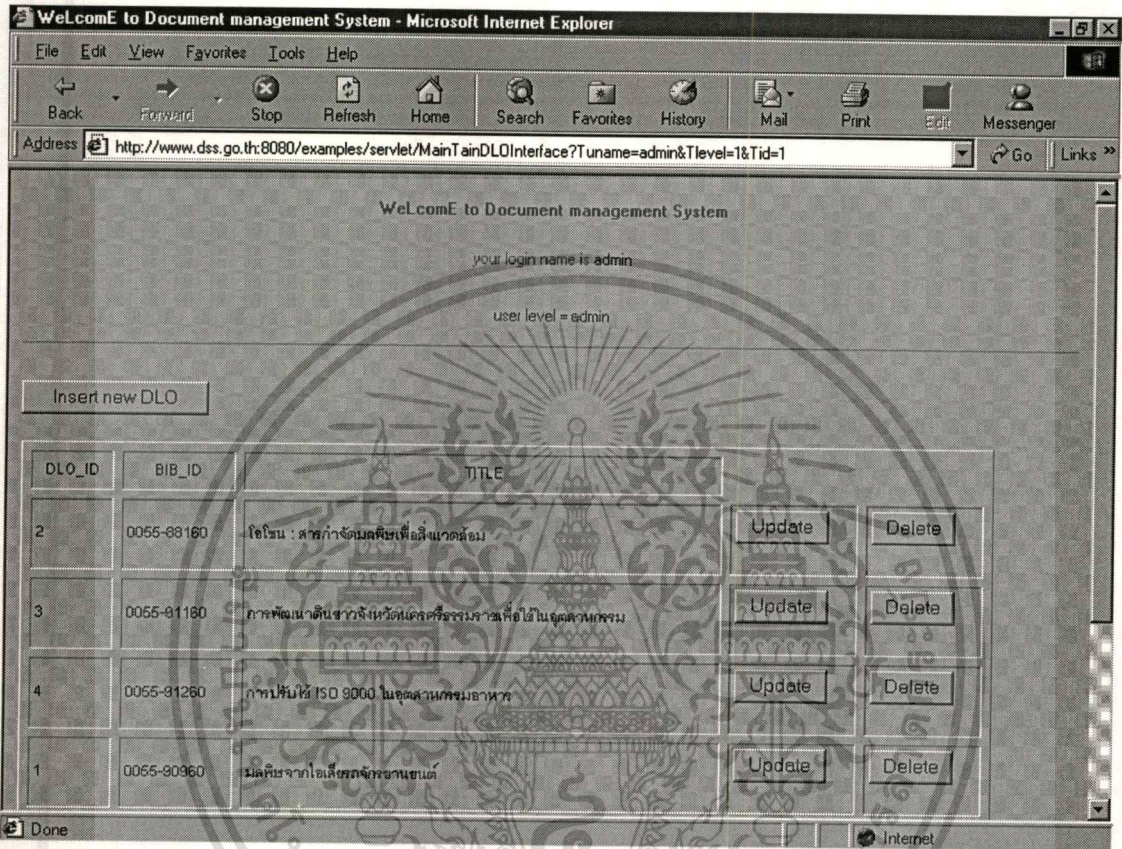
1. ฟอรัมสำหรับการเข้าใช้งานระบบ (Login)
2. ฟอรัมสำหรับการค้นหา (Search)
3. ฟอรัมสำหรับการส่งคำแนะนำ (Comment)
4. HyperLink ต่างๆ ได้แก่
 - รายการบทความในวารสารกรมวิทยาศาสตร์บริการทั้งหมด (List all issue by title)
 - รายการคำแนะนำทั้งหมด
 - รายการบทความทันสมัย (New Topic) และอื่นๆ



รูปที่ ก-1 หน้าจอแรก DSS DL Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ 2: หลังจากผ่านการยืนยัน เข้าสู่หน้าหลักการจัดการเก็บเอกสาร ประกอบด้วย 3 ฟังก์ชันการทำงานให้เลือก คือ การ Insert DLO , Update DLO และ Delete DLO



รูปที่ ก-2 หน้าจอหลักการจัดการเอกสาร

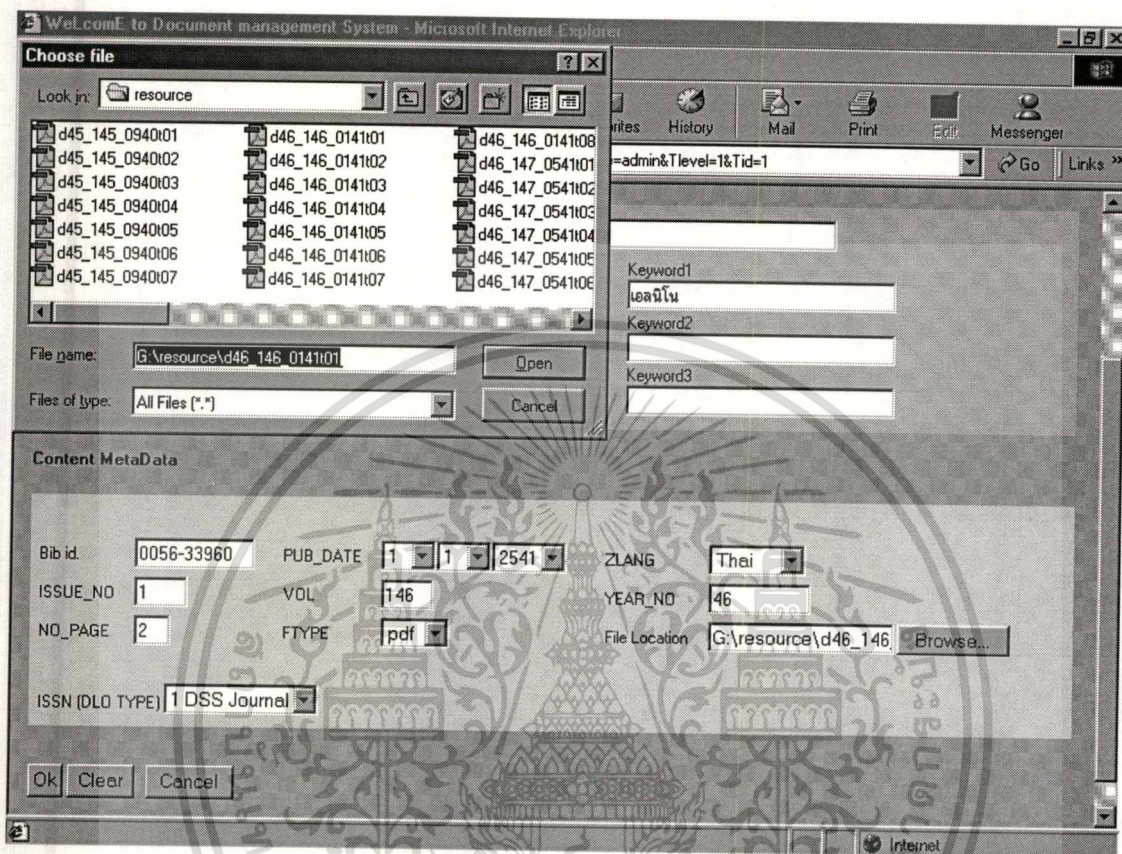
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ 3: หน้าจอเมื่อเลือก Insert DLO จะปรากฏแบบฟอร์มสำหรับการกรอกข้อมูลของเอกสาร แบ่งเป็น 2 ส่วนย่อย คือ Document Metadata และ Content Metadata ดังรูปที่ ก-3

รูปที่ ก-3 หน้าจอการจัดเก็บเอกสาร (Insert DLO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอกที่ 4: ตัวอย่างการเลือกเพิ่มข้อมูลเพื่อนำเข้าสู่ระบบ



รูปที่ ก-4 การเลือกเพิ่มข้อมูลเพื่อนำเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอที่ 5: หากต้องการแก้ไขจะต้องเลือกเอกสารที่ต้องการแก้ไขจากหน้าจอที่ 2 (รูปที่ ก-2) ในตัวอย่างเลือกชื่อเรื่อง “ไอโซน”

Wellcome to Document management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Messenger

Address <http://www.dss.go.th:8080/examples/servlet/UpdateDLOInterface?Tuname=admin&Tlevel=1&Tid=1&Tbid=0055-88160> Go Links >>

DLO_ID 2 Title ไอโซน : สารกำจัดแมลงพิษเพื่อสิ่งแวดล้อม

Author1 สุวิเศษ ศ.แสงจันทร์

Author2

Author3

Keyword1 ไอโซน

Keyword2 สิ่งแวดล้อม

Keyword3

Content MetaData

Bib id 0055-88160 PUB_DATE 1 9 2540 ZLANG Thai

ISSUE_NO 3 VOLUME 5 YEAR_NO 145

NO_PAGE 5 FTYPE pdf File Location d45_145_0940t03.pdf

ISSN (DLO TYPE) 1 DSS Journal

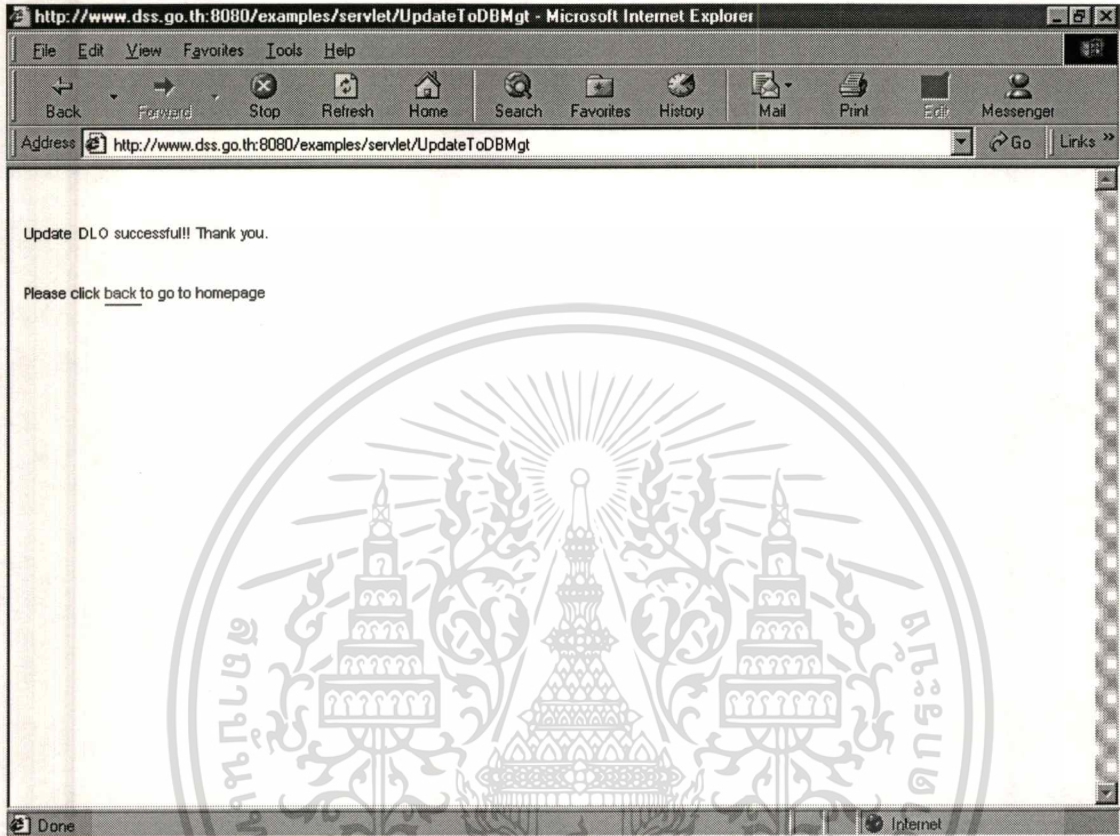
Update Cancel

Done Internet

รูปที่ ก-5 หน้าจอแก้ไขข้อมูลเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอที่ 6: หน้าจอเมื่อปรับปรุงข้อมูลเรียบร้อยแล้ว



รูปที่ ก-6 ผลลัพธ์หน้าจอหลังจากการปรับปรุงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอกที่ 7: การค้นคืนเอกสารจากห้องสมุดดิจิทัล ชนิควารสารกรมวิทยาศาสตร์บริการ (DSS journal) เมื่อป้อนคำค้น “ไอโซน” จากหน้าจอกที่ 2

Found 1 record(s)

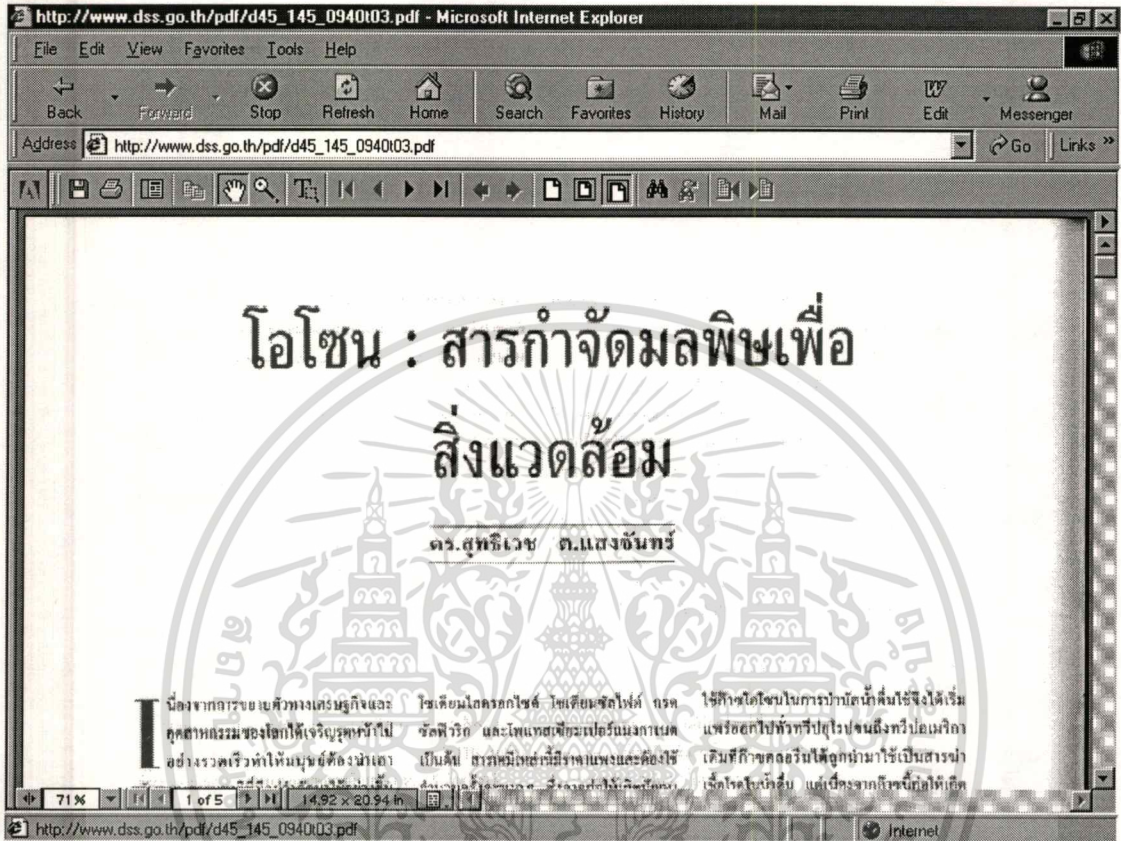
Doc. id	Title	Author	Type	
2	ไอโซน : สารกำจัดมลพิษเพื่อสิ่งแวดล้อม	สุทธิเวช ต.แสงจันทร์	DSS journal	Detail

Please click [back](#) to go to homepage

รูปที่ ก-7 ผลลัพธ์รายการที่สืบค้นพบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอที่ 8: เมื่อกดปุ่ม Detail เมื่อต้องการ เปิดไฟล์ หรือ ถ่ายโอนไฟล์เอกสาร



รูปที่ ก-8 หน้าจอเมื่อเปิดไฟล์ที่ค้นพบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. การทำ Database Normalization สำหรับระบบงานห้องสมุดดิจิทัล

1. สาเหตุที่ต้องทำ Normalization

- ความผิดปกติในการปรับปรุงข้อมูล (Update Anomalies)
- ความผิดปกติในการเพิ่มข้อมูล (Addition Anomalies)
- ความผิดปกติในการลบข้อมูล (Deletion Anomalies)

2. วิธีการทำ Normalization

⇒ ระดับที่ 1 First Normal Form (1NF)

จากตารางในรูปที่ ข-1 จะเห็นว่ายังไม่เป็น 1NF เพราะยังมี Repeating Group ได้แก่ รหัสผู้แต่ง (AUTH_ID) ชื่อผู้แต่ง (AUTH_NAME) และ คำค้น (KEYWORD)

ทำ First Normalization เพื่อขจัด Repeating Group เพื่อให้ค่าภายในแต่ละ Attribute มีค่าเพียงค่าเดียว (Atomic Value) โดยการเติม Attribute ทุก Record ให้สมบูรณ์ ดังรูปที่ ข-2

ตัวอย่างข้อมูลของบทความวารสารในรูปอิเล็กทรอนิกส์

ID	ISSN	NO	VOL	YEAR_NO	PUBL_DT	TITLE	AUT_H_ID	AUTH_NAME	PUB_ID	PUB_NAME	CRE_ID	CRE_NAME	NO_PAGE	FTYPE	FLOCATION	ZTYPE	LAN	KEYWORD	INPUT_DT
0023-53860	0857-7617	01	47	151	0942	การวัดคะแนน น่ายางธรรมชาติ โดยการใช้รังสีดี อย่างไร	1	ศรณี วัชรเวรวิทย์	01	กรมวิทยา ศาสตร์บริการ	1	เบญจรงค์ นิตยพัฒน์	4	pdf	d47_1 51_09 42101. pdf	DSSJ	TH	- ยางพารา - น่ายางพารา	08/1 2/99
0023-53560	0857-7617	02	47	151	0942	การจัดเก็บสาร เคมีสำหรับห้อง ปฏิบัติการ	5	อัญญา พุ่มจันทร์	01	กรมวิทยา ศาสตร์บริการ	1	เบญจรงค์ นิตยพัฒน์	3	pdf	d47_1 51_09 42102. pdf	DSSJ	TH	- ห้องปฏิบัติการ - สารเคมี - การจัดเก็บ	08/1 2/99
0061-16660	0857-7617	04	47	151	0942	กระดาษรีไซเคิล	2,3	รุ่งอรุณ วัฒน วงศ์, ธีรชัย รัตน โรจน์มงคล	01	กรมวิทยา ศาสตร์บริการ	1	เบญจรงค์ นิตยพัฒน์	5	pdf	d47_1 51_09 42104. pdf	DSSJ	TH	- กระดาษ - การนำกลับมาใช้ ใหม่.	08/1 2/99

รูปที่ ข-1 ตารางที่ยังไม่ได้ทำ Normalization (Unnormalization)

ID	ISSN	NO	VOL	YEAR_NO	PUBL_DT	TITLE	AUT_HL_ID	AUTH_NAME	PUB_ID	PUB_NAME	CRE_ID	CRE_NAME	NO_PAGE	FTYPE	FLOCA_TION	ZTYPE	LAN	KEYWORD	NPUT_DT
0023-53860	0857-7617	01	47	151	0942	การวัดคะแนนนโยบายธรรมาชาติ โดยการใช้รังสีดี อย่างไร	1	ศุภณี วัชรารองวิทย์	01	กรมวิทยาศาสตร์ บริการ	1	เบญจรงค์ นิตยพัฒน์	4	pdf	d47_1 51_094 2t01.p df	DSSJ	TH	ยางพารา	08/1 2/99
0023-53860	0857-7617	01	47	151	0942	การวัดคะแนนนโยบายธรรมาชาติ โดยการใช้รังสีดี อย่างไร	1	ศุภณี วัชรารองวิทย์	01	กรมวิทยาศาสตร์ บริการ	1	เบญจรงค์ นิตยพัฒน์	4	pdf	d47_1 51_094 2t01.p df	DSSJ	TH	นํายางพารา	08/1 2/99
0023-53560	0857-7617	02	47	151	0942	การจัดเก็บสารเคมีสำหรับห้องปฏิบัติการ	5	อัจฉรา พุ่มฉัตร	01	กรมวิทยาศาสตร์ บริการ	1	เบญจรงค์ นิตยพัฒน์	3	pdf	d47_1 51_094 2t02.p df	DSSJ	TH	ห้องปฏิบัติการ	08/1 2/99
0023-53560	0857-7617	02	47	151	0942	การจัดเก็บสารเคมีสำหรับห้องปฏิบัติการ	5	อัจฉรา พุ่มฉัตร	01	กรมวิทยาศาสตร์ บริการ	1	เบญจรงค์ นิตยพัฒน์	3	pdf	d47_1 51_094 2t02.p df	DSSJ	TH	สารเคมี	08/1 2/99
0023-53560	0857-7617	02	47	151	0942	การจัดเก็บสารเคมีสำหรับห้องปฏิบัติการ	5	อัจฉรา พุ่มฉัตร	01	กรมวิทยาศาสตร์ บริการ	1	เบญจรงค์ นิตยพัฒน์	3	pdf	d47_1 51_094 2t02.p df	DSSJ	TH	การจัดเก็บ	08/1 2/99

รูปที่ ข-2 ตารางจากรูปที่ 1-1 หลังจากที่ทำ First Normalization (เป็น 1NF)

4. ค้นหา Partial Dependency โดยดูจาก Non Key (attribute ที่ไม่ใช่ส่วนของ Primary Key) ว่าถูกระบุได้โดยเพียงบางส่วนของ Key ทำ Second Normalization เพื่อ "ขจัด Partial Dependency สำหรับทุกๆ Non Key " โดยการขจัด Partial Dependency ดังกล่าวออกมาเป็นตารางใหม่ โดยที่ยังคง Key ที่เป็นส่วนของ Primary Key ไว้ที่ตารางหลัก ในที่นี้มี AUTH_ID AUTH_NAME และ KEYWORD ที่เป็น Partial Dependency ฉะนั้นจึงแตกตารางหลักออกเป็นสอง ตารางย่อยดังนี้

DLO1 (BIB_ID,ISSN, NO, VOL, YEAR_NO, PUBL_DT, TITLE, PUB_ID, PUB_NAME, CRE_ID,CRE_NAME, NOPAGE, FTYPE, FLOCATION, ZTYPE, ZLANG, INPUT_DT)

DLO2 (DLO_ID, BIB_ID,. AUTHOR_ID, AUTHOR_NAME, KEYWORD)

รูปที่ ข-3 โครงสร้างตารางที่เป็น 2NF

⇒ ระดับที่ 3 Third Normal Form (3NF)

จากโครงสร้าง ข-3 จะเห็นว่ายังไม่เป็น 3NF เนื่องจากยังมี Transitive Dependency (การพึ่งพาแบบถ่ายทอด) ดังนี้

ISSN → PUB_ID, PUB_NAME, ZTYPE

PUB_ID → PUB_NAME

CRE_ID → CRE_NAME

AUTH_ID → AUTH_NAME

BIB_ID → ISSN, NO, VOL, YEAR_NO, PUBL_DT, TITLE, PUB_ID, PUB_NAME, CRE_ID,CRE_NAME, NOPAGE, FTYPE, FLOCATION, ZTYPE, ZLANG, INPUT_DT

DLO_ID → BIB_ID, AUTH_ID, AUTH_NAME, KEYWORD

ทำ Third Normalization เพื่อ "ขจัด Transitive Dependency " โดยการแยกตารางออก (แยก X,Y และ Y,Z ให้อยู่คนละตาราง) ในที่นี้คือ แยกตารางออกมา โดยยังคง key ที่ถ่ายทอดกันไว้ ดัง

รูปที่ ข-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

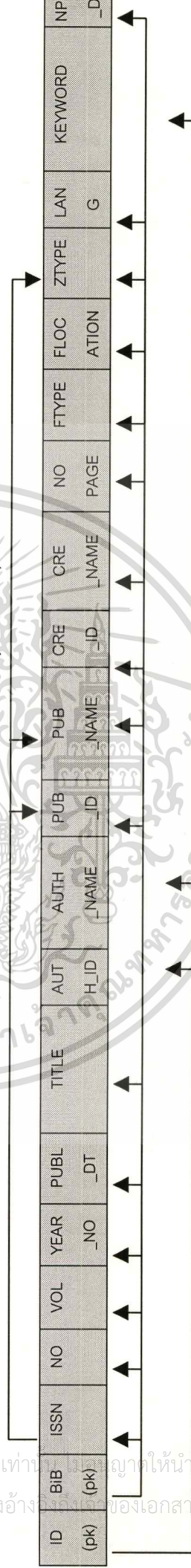
ID	ISSN	NO	VOL	YEAR	PUBL	TITLE	AUT	AUTH	PUB	PUB	CRE	NO	FTYPE	FLOC	ZTYPE	LAN	KEYWORD	INPUT
					_DT		H_ID	_NAME	_ID	_NAME	_ID	PAGE		ATION		G		_DT
0061-16660	0857-7617	04	47	151	0942	กระดาษรีไซเคิล	2,3	ธีรชัย รัตน์ โรจน์มงคล	01	กรมศึกษา ศาสตร์บริการ	1	5	pdf	d47_1 51_09 42104. pdf	DSSJ	TH	กระดาษรีไซเคิล	08/1 2/99
0061-16660	0857-7617	04	47	151	0942	กระดาษรีไซเคิล	2,3	รุ่งอรุณ วัฒน วงศ์	01	กรมศึกษา ศาสตร์บริการ	1	5	pdf	d47_1 51_09 42104. pdf	DSSJ	TH	การนำกลับมา ใช้ใหม่	08/1 2/99
0061-16660	0857-7617	04	47	151	0942	กระดาษรีไซเคิล	2,3	ธีรชัย รัตน์ โรจน์มงคล	01	กรมศึกษา ศาสตร์บริการ	1	5	pdf	d47_1 51_09 42104. pdf	DSSJ	TH	การนำกลับมา ใช้ใหม่	08/1 2/99

รูปที่ ข-2 ตารางจากรูปที่ ข-1 หลังจากทำ First Normalization (ต่อ)

⇒ ระดับที่ 2 Second Normal Form (2NF)

จากตารางในรูปที่ 1-2 จะเห็นว่ายังไม่เป็น 2NF เนื่องจากยังมี partial dependency (การพึ่งพาเพียงบางส่วน) โดยแสดงให้เห็นได้โดยทำตามขั้นตอนดังนี้

1. เขียนโครงสร้างของตาราง
2. ดูการพึ่งพากันของ Attribute ต่างๆ ในตาราง เช่น
 - ISSN จะทำให้ทราบข้อมูล PUB_ID PUB_NAME ZTYPE (แหล่งข้อมูลเดียวกัน)
 - AUTH_ID จะทำให้ทราบถึง AUTHO_NAME
 - CRE_ID จะทำให้ทราบถึง CRE_NAME, CRE_UNAME, CRE_PASSWD
 - PUB_ID จะทำให้ทราบถึง PUB_NAME
3. กำหนด Primary Key ในที่นี้จะเพิ่ม Field ชื่อว่า DLO_ID เข้าไปอีกหนึ่ง Field และกำหนดให้เป็น Primary Key ดังรูปที่ 1-3



รูปที่ ข-3 โครงสร้างของตาราง การพึ่งพากัน และ Primary Key

Type (ISSN , PUBID, ZTYPE)

Publisher(PUBID ,PUBNAME)

Creator(CRE_ID ,CRE_NAME)

Author(AUTH_ID, AUTH_NAME)

Content(BIB_ID, ISSN, NO, VOL, YEAR_NO, PUBL_DT, TITLE, CRE_ID, NOPAGE,

FTYPE, FLOCATION, ZLANG, INPUT_DT)

Document (DLO_ID, AUTH_ID KEYWORD,BIB_ID)

รูปที่ ข-5 โครงสร้างตาราง ต่างๆ หลังจากทำ Third Normalization (เป็น 3NF)

ก. บางส่วนของ source โปรแกรม

```
import java.io.*;           // for IOException
import javax.servlet.http.*; // for HttpServlet
import javax.servlet.*;    // for HttpServletRequest & HttpServletResponse & ServletException
import java.util.*;        // for Enumeration
import java.sql.*;

/**
 * รับค่า 3 parameter คือ uname,level,id และแสดงฟอร์มเปล่า
 * กรณีปรกติ
 */

public class MainTainDLOInterface extends HttpServlet {

public static final String DB_CONN = "jdbc:oracle:thin:@203.151.9.7:1521:dssdl";
public static final String DB_LOGIN = "system";
public static final String DB_PASSWD = "manager";

public static Parameter p = new Parameter(4);
public static String uname,level,id;
public static PrintWriter out;

public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    p = ServletUtilities.GetUserParameter(request);
        uname=p.value[1];
        level=p.value[2];
        id=p.value[3];

        out = response.getWriter();
        DisplayMainInterface();
} // end doGet method

public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
doGet(request, response);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public static void DisplayMainInterface() {
    String title = "WeLcomE to Document management System";
    String bgcolor = "#6CECFD"; // เขียว
    ServletUtilities.headerWithUserInfo(out,title,bgcolor,uname,level);
    DisplayDLOFromDB();
}

public static void DisplayDLOFromDB() {
    // ปรึ้ม insert
    out.print("<form action=\"InsertDLOInterface?Tuname="+ uname +"&Tlevel="+ level + "&Tid="+
    id + "\"method=\"POST\">" +
    "<font size=\"1\" face=\"MS Sans Serif\">" +
    "<input type=\"submit\" name=\"B1\" value=\"Insert new DLO\"></font>" +
    "</form>");

try{
    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
    Connection conn = DriverManager.getConnection(DB_CONN, DB_LOGIN, DB_PASSWD);
    String sqlDloList = "select DLO_ID,bib_id,TITLE from document order by dlo_id desc";
    PreparedStatement stmt = conn.prepareStatement(sqlDloList);
    ResultSet rset = stmt.executeQuery();
    ResultSetMetaData rsmd = rset.getMetaData();
    int numberOfColumns = rsmd.getColumnCount();
    out.println("<TABLE width=700 CELLPADDING=3 CELLSPACING=6 BORDER=1>");
    out.println("<TR>");

    // print หัวตาราง
    for(int i=1;i<=rsmd.getColumnCount();i++){
        out.println("<TH>"+rsmd.getColumnName(i)+"</TH>");
    }

    out.println("</TR>");

    // print ตารางข้อมูล dlo_id, bib_id, title
    while(rset.next()) {
        out.println("<TR>");

        for(int j=1;j<=rsmd.getColumnCount();j++){
            // field title ข้อมูลเป็น ไทย ต้อง convert กลับเป็น windows874
            String title_out=MTThai.UnicodeToMS874(rset.getString(j));

            if(j==3) out.println("<TD>"+ MTThai.UnicodeToMS874(rset.getString(j))+"</TD></form>");
            else out.println("<TD>"+ MTThai.UnicodeToMS874(rset.getString(j))+ "</TD>
            </form>");

        } // end for
        out.println("<TD>" +
        "<FORM ACTION=\"UpdateDLOInterface?Tuname="+ uname +"&Tlevel="+ level + "&Tid="+
        id + "&Tbibid=" + rset.getString(2)+"\"method=\"POST\">" +
        "<input type=\"hidden\" name=\"Tdloid\" value=\"" + rset.getString(1) + "\">" +
        "<input type=\"submit\" name=\"B2\" value=\"Update\"></form></TD><TD>" +
        "<FORM ACTION=\"DeleteDLOMgt?Tuname="+ uname +"&Tlevel="+ level + "&Tid="+ id +
        "&Tbibid="+ rset.getString(2)+"\" method=\"POST\">" +
        "<input type=\"hidden\" name=\"Tdloid\" value=\"" + rset.getString(1) + "\">" +
        "<input type=\"submit\" name=\"B3\" value=\"Delete\"></form></TD>");
        out.println("</TR>");
    } // end while
    out.println("</TABLE>");
    rset.close();          stmt.close();          conn.close();
} catch(Exception e){
    System.err.println(e.toString());
    out.println(e.toString());
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวเบญจรงค์ นิตยพัฒน์
วันเดือนปีเกิด	13 มีนาคม 2520
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (คอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2540



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้