

เว็บไคเร็กทอรีส่วนตัว

Personal Web Directory

โดย

นายสมรัฐ เขตนุช

รหัส 41067115



H001695

อาจารย์ที่ปรึกษา

ดร. จันทรบุรณ์ สติทวีรวงศ์

วัน เดือน ปี.....	25 S.A. 2549
เลขทะเบียน.....	01695
เลขเรียกหนังสือ.....	จท. ศ 275 จ 2543
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจธ."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	เว็บไคเร็กทอรีส่วนตัว
นักศึกษา	นายสมรัฐ เขตนุช
อาจารย์ที่ปรึกษา	ดร. จันทรบุรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

จากสถิติการใช้งานเครือข่ายอินเทอร์เน็ตมีจำนวนมากขึ้นทุกวัน ทั้งทางด้านปริมาณ Traffic ของข้อมูล และจำนวนผู้ใช้ที่เพิ่มขึ้น มีแนวโน้มที่การใช้งานจะแพร่ไปสู่ นักเรียน นักศึกษา และผู้ใช้ที่อยู่ตามบ้านเรือน (Home User) มากขึ้นและปัจจุบันบริการหนึ่งของอินเทอร์เน็ต ได้แก่ เครือข่ายเวิลด์ไวด์เว็บ (WWW) ได้รับความนิยมสูงมากขึ้นเนื่องจากมี ภาพและเสียง ประกอบที่สวยงาม เสนอข้อมูลแบบหลายสื่อ (Multi Media) ได้สามารถติดต่อกับผู้ใช้ได้หลากหลายภาษา ในแบบ GUI ในประเทศไทยผู้ใช้งานในแบบ End User มีแนวโน้มที่จะพัฒนาโฮมเพจเป็นของตนเองไปใส่ตาม Free Home Page ต่างๆ ได้มากขึ้นเพราะเครื่องมือที่ใช้ในการพัฒนามีความง่ายมากขึ้นบางรายทำได้ ตีจนเกิดเป็นธุรกิจแนวทาง E-Commerce ที่น่าสนใจอย่างมาก เนื่องจากสามารถสร้างการค้าที่หน้าร้านเปิดตลอด 24 ชั่วโมงโดยไม่ต้องมีคนเฝ้าร้าน และผู้ที่มาเลือกชมสินค้าก็เป็นผู้คนที่ Online ใช้งาน Internet อยู่ทั่วโลกแต่ Website ที่เกิดขึ้นนี้มีจำนวนมากจนบางครั้งผู้สนใจข้อมูลบางอย่างใน WWW ไม่สามารถค้นหาข้อมูลที่ต้องการได้ การพัฒนา Web Directory สำหรับ Web ต่างๆ ที่มีอยู่มากมายให้ผู้ใช้สามารถค้นหาได้เป็นหมวดหมู่และมีประสิทธิภาพ จึงอาจช่วยให้การเติบโตของการใช้งาน Internet เป็นไปอย่างมีรูปแบบมากขึ้น

Title	Personal Web Directory
Student	Mr. Somrut Chetnuch
Advisor	Dr. Chanboon Sathitwiriawong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

From Internet use statistics that much more every day. Both Network Traffic and Number of Internet Users. Recent Internet to be widespread to student and Home User. Internet Service that have high amount of User is World Wide Web (WWW) cause it have Picture and Sound , can present Multimedia and can Display Multi Language in Graphic User Interface. In Thailand End User can use web page developer tools to create own Home Page and send to store in Free Home Page Server. Some People can create own business call E-Commerce, that every interesting to create the shop that can open 24 hour with no man to watch the shop and your customer is people Online WWW around the World. But many user can not find Information they need on the Web. To develop Web Directory may help grouping and searching on the Web.

กิตติกรรมประกาศ

ในการจัดทำโครงการพัฒนาระบบเว็บไคลเอนต์ส่วนตัว นี้ ได้รับความสนับสนุนเป็นอย่างดี จากหลายฝ่าย ที่คอยให้คำแนะนำปรึกษา และเสียสละเวลาอันมีค่า จนทำให้การศึกษาโครงการนี้บรรลุผลตามเป้าหมายที่วางไว้ ผู้จัดทำจึงใคร่ขอขอบพระคุณ

1. บิดา, มารดา ที่คอยให้กำลังใจในการทำงานแก่ข้าพเจ้าตลอดมา
2. ดร. จันทร์บุรณ สติฉวีริยวงศ์ อาจารย์ที่ปรึกษาโครงการที่ได้ให้ความรู้ คำปรึกษา ในการจัดทำโครงการ
3. อาจารย์ทุกๆ ท่าน ที่เคยสอนให้ข้าพเจ้ามีความรู้ความสามารถ ด้าน IT จากความรู้พื้นฐานที่ได้ศึกษามา ในทุกรายวิชาตลอดระยะเวลา 2 ปี
4. เครื่องคอมพิวเตอร์ ซอฟต์แวร์ และระบบเครือข่ายอินเทอร์เน็ตของ KMITL และการสื่อสารแห่งประเทศไทย ที่ใช้เป็นเครื่องมือในการสืบค้นข้อมูลที่เกี่ยวข้อง
5. เพื่อนๆ IS6 ภาคสมทบรุ่นแรก ที่เป็นกำลังใจ และฝ่าฟันอุปสรรคไปพร้อมกัน รวมทั้ง is6.smiletrip.com ที่เป็นสื่อกลางในการแลกเปลี่ยนความรู้ และกระจายข่าวสารต่างๆ

สมรัฐ เขตनुช

ผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนการศึกษา.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. วรรณกรรมและทฤษฎีที่เกี่ยวข้อง.....	4
2.1 การพัฒนาซอฟต์แวร์เชิงวัตถุ.....	4
2.2 วิธีการเชิงวัตถุ.....	5
2.3 UML.....	7
2.3.1 ความเป็นมาของ UML.....	7
2.3.2 แผนผังใน UML.....	11
2.4 สถาปัตยกรรม 3 Tiers และ ASP.....	25
2.4.1 ข้อดีของสถาปัตยกรรมแบบ 3 Tiers.....	25
2.5 ASP.....	26
3. ศึกษาระบบ Search Engine และ Web Directory.....	27
3.1 การศึกษาโครงสร้างของเสิร์จเอนจิน.....	27
3.1.1 โปรแกรมโรบอต.....	27
3.1.2 Indexer.....	29

3.1.3 Search Engine	31
3.2 โครงสร้างของเว็บไคลเอนต์.....	31
4. การออกแบบระบบ.....	32
4.1 การวิเคราะห์ห้ออกแบบระบบ	32
4.1.1 กำหนดวัตถุประสงค์.....	32
4.1.2 Hardware and Software Specification.....	32
4.1.3 Use Case Diagram.....	33
4.1.4 Component Diagram.....	33
4.1.5 Class Diagram.....	36
4.1.6 Activity Diagram.....	37
4.1.7 Deployment Diagram.....	43
4.1.8 Relational Table	44
4.2 พัฒนาโปรแกรมด้วย ASP.....	46
5. บทสรุปและข้อเสนอแนะ.....	47
5.1 ผลการศึกษา.....	47
5.2 สรุป.....	48
5.3 ข้อเสนอแนะ.....	48
บรรณานุกรม.....	49
ภาคผนวก ก. Data Dictionary.....	50
ภาคผนวก ข. ตัวอย่างการใช้งาน เว็บไคลเอนต์ส่วนตัว.....	56
ประวัติผู้เขียน	61

สารบัญตาราง

หน้า

ตารางที่

1.1	ขั้นตอนการศึกษา	3
3.1	การเข้าถึงข้อมูลในคลาส	15



สารบัญภาพ

หน้า

ภาพที่		
2.1	แสดงการจำแนก Methodology	8
2.2	ความเป็นมาของ UML	9
2.3	สัญลักษณ์ของ Actor	12
2.4	สัญลักษณ์ของยูสเคส.....	12
2.5	สัญลักษณ์ของชิตเต็ม	12
2.6	สัญลักษณ์ของคอมมิวนิเคชัน.....	13
2.7	สัญลักษณ์ของรีเลชันชิพ	13
2.8	รูปแบบของยูสเคสไคอะแกรม	13
2.9	ตัวอย่างยูสเคสไคอะแกรมของระบบจองห้องพัก	14
2.10	ส่วนประกอบพื้นฐานของคลาส	14
2.11	ความสัมพันธ์แบบทางเดียว.....	15
2.12	ความสัมพันธ์แบบสองทาง.....	15
2.13	ความสัมพันธ์แบบ by value	16
2.14	ความสัมพันธ์แบบ by reference	16
2.15	ความสัมพันธ์แบบ Whole Part.....	16
2.16	ความสัมพันธ์แบบ Depends on.....	16
2.17	ความสัมพันธ์แบบ Generalization	17
2.18	สัญลักษณ์ที่ใช้ในสเตทไคอะแกรม	17
2.19	สเตทไคอะแกรมของออบเจกต์จองห้องพัก	18
2.20	Action State.....	19
2.21	สัญลักษณ์แสดงการตัดสินใจ.....	20
2.22	การตัดสินใจ	20
2.23	Swimlanes	20
2.24	สัญลักษณ์ Sync State.....	21
2.25	Sync State.....	21

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

2.23 Swimlanes.....	20
2.24 สัญลักษณ์ Sync State.....	21
2.25 Sync State.....	21
2.26 สัญลักษณ์ที่ใช้ในไดอะแกรมแสดงลำดับ.....	22
2.27 ตัวอย่างการใช้คอลลาบอเรชันไดอะแกรมในระบบลงทะเบียน.....	23
2.28 สัญลักษณ์แทนคอมโพเนนต์และอินเทอร์เฟซ.....	23
2.29 ระบบลงทะเบียนที่ใช้คอมโพเนนต์ไดอะแกรม.....	24
2.30 การใช้คีย์พลอยเมนต์ไดอะแกรมในระบบลงทะเบียน.....	24
2.31 สถาปัตยกรรมแบบ 3 Tiers.....	25
3.1 แสดงการทำงานของโปรแกรมโรบอต.....	27
3.2 การทำงานของอินเด็กเซอร์.....	29
3.3 รายละเอียดการทำงานของอินเด็กเซอร์.....	30
4.1 รายละเอียดยูสเคสไดอะแกรม.....	33
4.2 รายละเอียดคอมโพเนนต์ไดอะแกรม.....	34
4.3 รายละเอียดคอมโพเนนต์ไดอะแกรมใน Client Display.....	34
4.4 รายละเอียดคอมโพเนนต์ไดอะแกรมใน Application YuDir.....	35
4.5รายละเอียดคอมโพเนนต์ไดอะแกรมใน Application YuDir (1).....	35
4.6 รายละเอียดคอมโพเนนต์ไดอะแกรมใน Application YuDir (2).....	36
4.7 รายละเอียดคอมโพเนนต์ไดอะแกรมใน Database.....	36
4.8 รายละเอียด Class Diagram ใน Database Engine.....	36
4.9 Activity Diagram ส่วนของการ Login.....	37
4.10 Activity Diagram ส่วนของการ เป็น Admin (1/7).....	37
4.11 Activity Diagram ส่วนของการ Admin (2/7).....	38
4.12 Activity Diagram ส่วนของการ Admin (3/7).....	38
4.13 Activity Diagram ส่วนของการ Admin (4/7).....	39

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้ใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

4.14 Activity Diagram ส่วนของการ Admin (5/7).....	39
4.15 Activity Diagram ส่วนของการ Admin (6/7).....	40
4.16 Activity Diagram ส่วนของการ Admin (7/7).....	40
4.17 Activity Diagram ส่วนของการ เป็น User (1/4).....	41
4.18 Activity Diagram ส่วนของการ เป็น User (2/4).....	41
4.19 Activity Diagram ส่วนของการ เป็น User (3/4).....	42
4.20 Activity Diagram ส่วนของการ เป็น User (4/4).....	42
4.21 ดิพลอยเมนต์ไดอะแกรม	43
4.22 รายละเอียดของดิพลอยเมนต์ไดอะแกรม	43
4.23 การสร้าง Relational Table จาก Database Class Diagram	44
4.24 การสร้าง Relational Table จาก Database Class Diagram (2)	45
4.25 การสร้าง Relational Table จาก Database Class Diagram (3)	45

บทที่ 1

บทนำ

1.1 ความเป็นมา

อินเทอร์เน็ตในปัจจุบันมีข้อมูลข่าวสารอยู่มากมาย โดยเฉพาะข้อมูลไฮเปอร์เทกซ์ (Hyper Text) ที่ให้บริการอยู่ในระบบ Web ซึ่งเป็นที่นิยมอย่างแพร่หลายด้วยข้อดีที่ว่าสามารถมีภาพและเสียง ประกอบที่สวยงาม เสนอข้อมูลแบบสื่อผสม (MultiMedia) ได้สามารถติดต่อกับผู้ใช้ได้หลากหลายภาษา ในแบบกราฟฟิก (Graphic User Interface) การค้นหาข้อมูลที่มีเนื้อหาตรงกับความต้องการจากเว็บที่มีหลายล้านเพจในอินเทอร์เน็ตจึงเป็นเรื่องยากหากไม่มีเครื่องมือช่วยค้นหาซึ่งรู้จักกันดีในชื่อของเสิร์จเอนจิน (Search Engine)

บริการค้นหาเว็บเพจที่มีอยู่ในปัจจุบันเช่น AltaVista, Lycos, Yahoo, Hotbot, Infoseek ซึ่งมักเรียกเว็บทั้งหมดนี้โดยรวมว่าเป็นเสิร์จเอนจินนั้น จริงๆแล้วมีเพียงบางเว็บที่ทำงานเป็นระบบเสิร์จเอนจินอย่างแท้จริงและเว็บบางแห่งทำงานด้วยวิธีเก็บเว็บเพจไว้เป็น “ไครเอทอรี” ซึ่งมีความแตกต่างกันโดยหลักพื้นฐานการสร้างครรชนีเว็บเพจไครเอทอรีมีการทำงานที่แตกต่างจากเสิร์จเอนจิน คือการใช้การปรับเพิ่มข้อมูลโดยผู้ดูแลระบบเองโดยไม่ได้ทำอย่างอัตโนมัติ เว็บไซด์ใดที่ต้องการมีรายชื่อในไครเอทอรีต้องติดต่อไปยังผู้ดูแลไครเอทอรี เพื่อให้ผู้ดูแลจำแนกและจัดเก็บลงฐานข้อมูลด้วยตัวเอง ระบบการเก็บแบบไครเอทอรีอาจจะมีการทำงานที่ไม่อัตโนมัติเท่าที่ควร แต่การค้นหาข้อมูลจากเว็บที่ทำงานลักษณะเว็บไครเอทอรีอาจให้ผลลัพธ์ของการค้นหาข้อมูลที่ตรงประเด็นมากกว่าเสิร์จเอนจินเพราะผ่านการตรวจแยกหมวดหมู่ของข้อมูลจากผู้ดูแลไครเอทอรีมาแล้ว ตัวอย่างของบริการค้นหาที่ใช้ระบบไครเอทอรีได้แก่ Yahoo (ที่ URL <http://www.yahoo.com>) แต่ละส่วนของไครเอทอรีของ Yahoo จะมีข้อมูลที่แบ่งย่อยลงไปเป็นหมวดหมู่อีกหลายระดับ

โครงการนี้นำเสนอการออกแบบและการพัฒนาโปรแกรมสำหรับใช้ในการทำเว็บไครเอทอรีส่วนตัว สามารถเก็บข้อมูลผู้ใช้งาน ได้แก่ เพศ, อายุ, การศึกษา และ สี่ที่ชอบ เป็นต้น เพื่อเป็นข้อมูลเบื้องต้นที่ใช้ในการค้นหาข้อมูลจากเว็บไครเอทอรีที่จะสร้างขึ้น และเรียงลำดับข้อมูลที่ค้นหาได้ หรือแสดงออกมาเป็นแบนเนอร์ (Banner) ให้มีแนวโน้มที่จะตรงกับความต้องการของผู้ใช้มากขึ้น และสร้างความประทับใจให้ผู้ใช้งาน

การพัฒนาโปรแกรมในปัจจุบัน ประสบปัญหาที่สำคัญปัญหาหนึ่งคือ ปัญหาการบำรุงรักษา ซึ่งเป็นปัญหาที่รุนแรงเพิ่มมากขึ้นเรื่อยๆ และสิ้นเปลืองค่าใช้จ่าย ตลอดจนเสียทรัพยากร โดยรวมในการพัฒนาซอฟต์แวร์มากกว่าครึ่งหนึ่งของการพัฒนาทั้งหมด ปัจจุบันเริ่มมีการนำวิธีการแบบ Object-Oriented มาใช้เพื่อแก้ไขปัญหาดังกล่าวข้างต้น โดยเป็นวิธีการที่มุ่งเน้นไปที่ การนำเอาส่วนประกอบของซอฟต์แวร์กลับมาใช้ใหม่ได้ (Reusability of Software Component) เป็นโปรแกรมที่สามารถปรับปรุงแก้ไขได้ง่าย

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาหลักการของระบบเว็บไคลเอนทอริ
2. ศึกษาการวิเคราะห์และออกแบบระบบงานด้วย UML
3. ศึกษาการการนำผลลัพธ์จากการออกแบบด้วยแนวคิดเชิงวัตถุมาใช้งานบนฐานข้อมูลเชิงสัมพันธ์ (Relational Database)
4. สามารถออกแบบฐานข้อมูล และพัฒนาฐานข้อมูลเชิงสัมพันธ์ โดยใช้ CASE Tool

1.3 ขอบเขตของโครงการ

1. เพื่อวิเคราะห์และออกแบบระบบเว็บไคลเอนทอริส่วนตัว โดยใช้พื้นฐานการออกแบบของ UML (Unified Modeling Language)
2. เพื่อจัดทำโปรแกรมเว็บไคลเอนทอริส่วนตัว
3. ออกแบบ และสร้างฐานข้อมูลเชิงสัมพันธ์ โดยการสร้าง Database Schema ด้วย CASE Tool Rational Rose 2000 Edition

1.4 ขั้นตอนการศึกษา

ลำดับที่	รายละเอียด	ก.ค.	ส.ค.	ก.ย.	ต.ค.
1	ศึกษา Web Directory ในปัจจุบัน และข้อมูลที่จัดเก็บ	■			
2	ศึกษา UML และ Object-Oriented Method	■			
3	ออกแบบระบบ และซอฟต์แวร์ที่ใช้	■			
4	ออกแบบฐานข้อมูล		■		
5	ศึกษาภาษา ASP		■		
6	พัฒนาโปรแกรม		■		
7	ทดสอบและปรับปรุงแก้ไข			■	
8	สรุปผลการพัฒนา และข้อเสนอแนะ				■
9	เขียนรายงาน และแก้ไขเอกสาร	■	■	■	■

ตารางที่ 1.1 ขั้นตอนการศึกษา

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถนำแนวคิดและทฤษฎีต่างๆ ที่ได้ศึกษาในการทำโครงการศึกษากรณีพิเศษไปประยุกต์ใช้กับการพัฒนาระบบงานจริงได้ ซึ่งได้แก่
 - การวิเคราะห์และออกแบบระบบด้วยหลักการ Object-Oriented
 - การพัฒนาโปรแกรมโดยใช้ UML ในการออกแบบ
- ได้โปรแกรม Personal Web Directory
- ศึกษาการใช้งานภาษา ASP ร่วมกับฐานข้อมูล Oracle บน Windows NT

บทที่ 2

วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

2.1 การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software Development)

การทำธุรกิจในปัจจุบันต้องการความรวดเร็วและมีประสิทธิภาพในการทำงานเพื่อให้ธุรกิจสามารถดำเนินไปได้อย่างมีประสิทธิภาพจากความต้องการดังกล่าวจึงได้มีการนำระบบคอมพิวเตอร์เข้ามาใช้ในธุรกิจเพื่อให้การทำงานมีประสิทธิภาพและได้ข้อมูลทางธุรกิจที่ถูกต้องรวดเร็ว มีการพัฒนาซอฟต์แวร์ต่างๆขึ้นมาช่วยในการทำธุรกิจประเภทต่างๆ ไม่ว่าจะเป็นซอฟต์แวร์สำหรับการทำงานต่างๆไป หรือแม้แต่ซอฟต์แวร์เสนอข้อมูลสำหรับผู้บริหาร (EIS)

ในการพัฒนาซอฟต์แวร์ (Software Development) มีรูปแบบการพัฒนาซอฟต์แวร์อยู่ด้วยกันหลายรูปแบบ โดยแต่ละรูปแบบมีข้อดีข้อเสียแตกต่างกัน โดยรูปแบบที่ได้รับความนิยมในปัจจุบันคือการพัฒนาซอฟต์แวร์แบบโครงสร้าง (Structure Software Development) เป็นรูปแบบการพัฒนาซอฟต์แวร์ภายใต้หลักการที่มีการแยกส่วนของฟังก์ชัน (function) การทำงานและข้อมูล (data) ออกจากกัน การพัฒนาจะให้ความสำคัญกับฟังก์ชันการทำงานเป็นหลัก มีการแตกฟังก์ชันการทำงานออกเป็นฟังก์ชันย่อยๆเพื่อลดความซับซ้อนของระบบ เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์รูปแบบนี้ได้แก่ data flow diagram , structure chart และ entity relationship diagram รูปแบบการพัฒนาแบบนี้ค่อนข้างดี แต่ก็ยังมีข้อเสียที่พบคือการบำรุงรักษาซอฟต์แวร์ทำได้ลำบากเมื่อมีการเปลี่ยนแปลงความต้องการของลูกค้าซึ่งเป็นปัญหาใหญ่ในการพัฒนาซอฟต์แวร์ นอกจากนี้ซอฟต์แวร์ที่มีการพัฒนาขึ้นมาแต่ละครั้งมักจะไม่ถูกนำมาใช้ใหม่ในการพัฒนาซอฟต์แวร์อื่นๆถัดไปซึ่งเป็นการลงทุนที่ไม่คุ้มค่ากล่าวคือขาดความสามารถในการนำซอฟต์แวร์ที่พัฒนาแล้วกลับมาใช้ใหม่ (Reusability)

จากปัญหาที่เกิดขึ้นในเรื่องการพัฒนาซอฟต์แวร์แบบโครงสร้าง จึงได้มีผู้เสนอแนวคิดในการพัฒนาซอฟต์แวร์รูปแบบใหม่ที่ช่วยแก้ปัญหาดังกล่าวข้างต้นคือทำให้การบำรุงรักษาซอฟต์แวร์ทำได้ง่ายและเกิดความคุ้มค่าในการพัฒนาซอฟต์แวร์ สามารถนำซอฟต์แวร์ที่มีอยู่เดิมกลับมาใช้ใหม่ได้รูปแบบการพัฒนาซอฟต์แวร์แบบนี้เรียกว่า “การพัฒนาซอฟต์แวร์เชิงวัตถุ” (Object-Oriented Software Development) แนวคิดในการพัฒนา ซอฟต์แวร์รูปแบบนี้เป็นการการผนึกเอาข้อมูลและฟังก์ชันการทำงานเข้าด้วยกันเรียกว่าวัตถุ (Object) และกำหนดตัวประสาน (interface)

ระหว่างวัตถุต่างๆ โดยถ้าวัตถุหนึ่งต้องการจะติดต่อกับอีกวัตถุหนึ่งต้องติดต่อผ่านส่วนติดต่อกันเท่านั้น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่น จากหลักการนี้จะพบว่า การเปลี่ยนแปลงแก้ไขฟังก์ชันงานและข้อมูลใดๆ ในวัตถุหนึ่งจะมีผลกระทบต่อวัตถุอื่นน้อยมาก ซึ่งจะทำให้การแก้ไขโปรแกรมทำได้สะดวกและรวดเร็วยิ่งขึ้น เป็นการลดต้นทุนในการบำรุงรักษาโปรแกรม นอกจากนี้ยังมีความสามารถในการถ่ายทอดคุณสมบัติ (inheritance) จากวัตถุหนึ่งไปอีกรายหนึ่งได้ ทำให้สามารถนำซอฟต์แวร์บางส่วนที่มีอยู่เดิมกลับมาใช้ใหม่ได้ โดยผู้พัฒนาไม่จำเป็นต้องทราบว่าซอฟต์แวร์ส่วนที่นำกลับมาใช้นั้นเขียนขึ้นมาอย่างไร (How does it do?) เพียงแต่ทราบว่าซอฟต์แวร์ส่วนนั้นทำอะไร (What does it do?) และทำการเพิ่มคุณสมบัติอื่นๆ ที่ผู้พัฒนาต้องการเข้าไป ทำให้การพัฒนาซอฟต์แวร์ใหม่ๆ ทำได้รวดเร็วขึ้น

2.2 วิธีการเชิงวัตถุ (Object-Oriented Methodology)

เป็น Paradigm ที่ใหม่ล่าสุดในการจัดการกับปัญหาโดยมองเป็นออบเจกต์ ซึ่งกำลังเป็นที่นิยมในปัจจุบัน และมี Tool ที่สำคัญคือ Object Model (รวม data กับ process เข้าด้วยกัน) ซึ่งมีภาษาที่สามารถใช้ร่วมกับ Technology นี้คือ C++, Smalltalk, Eiffel, Object C, Object Pascal และ Java ในขณะที่ Delphi และ VB กำลังพยายามทำอยู่ คำว่า "Object" (Real World Object) โดยทั่วไปจะหมายถึง วัตถุใดๆ จะมีสิ่งสำคัญคือ ลักษณะ, สถานะ และ พฤติกรรม (Attribute, State, Behavior) ส่วน Software Object จะพยายามเลียนแบบ Real World Object โดยแบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็นสถานะ (Data and variable) และ พฤติกรรม (Method, Process, Operation)

ออบเจกต์ เป็น ชุดของ Data (Attribute, properties) และ function logic ซึ่ง data จะบอกถึงคุณสมบัติหรือสถานะของ Object และ Method จะบอกถึงพฤติกรรมต่างๆ ของออบเจกต์นั้นๆ โดย Attribute คือข้อมูลที่เราสสนใจเกี่ยวกับออบเจกต์ นั้น Method จะแบ่งเป็น 2 ประเภทคือ Interface method (เป็น method ที่ถูกใช้ได้จากออบเจกต์อื่น) และ Internal method (เป็น method ที่จะถูกเรียกใช้ได้เฉพาะภายในออบเจกต์ที่เป็นเจ้าของเท่านั้น)

นอกจากนี้แล้ว State of Object สถานะของออบเจกต์ ณ ระยะเวลาใดเวลาหนึ่ง ซึ่งจะมีการเปลี่ยนแปลงสถานะเมื่อมีการกระทำจากภายนอกแต่มีออบเจกต์บางตัวสามารถเปลี่ยนสถานะด้วยตัวเองได้ เรียกว่า "Object with life" หรือ "Actor" เช่น นาฬิกา ลูกค้า และยังมีอีกคำหนึ่งที่ควรทราบคือ "Class" เป็นพิมพ์เขียวของออบเจกต์ที่ไม่สามารถนำมาใช้ได้โดยตรง โดยจะมีการบอกถึง Method ที่ใช้ได้โดยออบเจกต์ และมีการแสดง Data type ที่บอกถึงสถานะของออบเจกต์ โดยยังไม่ระบุค่าใน Attribute แต่ละตัวซึ่งถ้าเป็นออบเจกต์ จะมีการระบุค่าของ Attribute ที่แน่นอน ถ้าออบเจกต์ เป็น instance ของ Class ใดออบเจกต์ นั้นจะต้องมี Attribute และ Method เหมือน Class ของมัน การที่จะเป็นออบเจกต์ ได้ต้องมีคุณสมบัติดังนี้

1. Encapsulation / Information hiding ทฤษฎีนี้ถูกคิดค้นโดย James Rumbaugh ซึ่งหมายถึง การแยกลักษณะภายนอก (Interface) ซึ่งสามารถติดต่อกับออบเจกต์ อื่นๆ ออกจากส่วนที่ implement ภายในของออบเจกต์ ซึ่งจะถูกใช้ได้เฉพาะตัวออบเจกต์ นั้นๆ มีข้อดีคือ เพิ่มความสามารถในการปกป้องข้อมูลโดยไม่ให้ออบเจกต์ อื่นเข้ามาทำการเปลี่ยนแปลงข้อมูลก่อนได้รับอนุญาตทั้งนี้ตัวออบเจกต์ควรรู้และทำการเปลี่ยนค่าด้วยตนเองซึ่งเป็นผลให้ maintain object ง่ายขึ้น
2. Inheritance คือหลักการในการที่ออบเจกต์ ใดๆในตัวใน generalized collection ได้มีการใช้ข้อมูล (structure) และพฤติกรรม (behavior) ร่วมกันซึ่งจะมีความสัมพันธ์แบบ is-a relationship โดยเรียก class ที่อยู่เหนือกว่าว่า parent class ซึ่งจะถ่ายทอดคุณสมบัติทั้ง Attribute และ method มายัง class ที่ต่ำกว่าเรียกว่า subclass ซึ่งจะมี Attribute และ method เพิ่มเติมจาก parent class มีข้อดีคือ เพิ่ม consistency เพียงแค่เปลี่ยน Method หรือ Attribute ที่ Super class ซึ่งเป็นผลให้ค่าที่ Subclass เปลี่ยนไป ด้วย เป็นการส่งเสริมการนำออบเจกต์ กลับมาใช้ใหม่
3. Polymorphism อาจหมายถึง “having many forms” โดยการส่ง message เดียวกันให้กับออบเจกต์ ที่ต่างกันเป็นผลทำให้ออบเจกต์ แสดงพฤติกรรมที่แตกต่างกัน มีข้อดีคือเป็นการสนับสนุนการนำโปรแกรมกลับมาใช้ใหม่และสามารถเปลี่ยนแปลงซอฟต์แวร์ได้ในระหว่างการพัฒนา ความสัมพันธ์ระหว่างออบเจกต์ มี 4 แบบคือ
 - 3.1 Association เป็นการอธิบายถึงความสัมพันธ์ระหว่างโครงสร้างโดยทั่วไปกับ semantic ทั่วไป โดยอ้างถึงความสัมพันธ์ทั้งสองด้านและมีการระบุชื่อความสัมพันธ์อย่างชัดเจน ความสัมพันธ์แบบนี้ประกอบด้วย
 - Constraint คือการแสดงข้อมูลที่เกี่ยวข้องกับความสัมพันธ์ระหว่างออบเจกต์
 - Link คือ การเชื่อมต่อระหว่างออบเจกต์ ที่มีความสัมพันธ์แบบ ถาวร ซึ่งเป็น instance ของ Association
 - 3.2 Aggregation เป็นรูปแบบความสัมพันธ์อย่างหนึ่งของ Association ในการกำหนด parts ใน Component นั้นโดยมีการแยกระหว่าง Whole กับ parts ซึ่งจะได้ความสัมพันธ์แบบ Whole - Part หรือ a-part-of หรืออาจเรียกอีกแบบหนึ่งว่ามีความสัมพันธ์แบบ Has-a relationship เป็นผลให้ Superclass มีอิทธิพลต่อการดำรงอยู่ของ Subclass คือถ้า Superclass ถูกทำลาย Subclass จะถูกทำลายไปด้วยโดยอัตโนมัติ นอกจากนี้การพิจารณาถึงการแบ่งระดับของ Aggregation นั้นขึ้นอยู่กับความต้องการของระบบ

3.3 Generalization เป็นความสัมพันธ์ในการจัดกลุ่มสิ่งๆ ที่เหมือนกันระหว่าง ออบเจกต์ หรือ Class ขึ้นมาเป็นอีกออบเจกต์ หนึ่งโดยพิจารณาจากข้อมูล (structure) หรือ พฤติกรรม (behavior) ในกลุ่มของออบเจกต์ นั้น จะมีลักษณะเป็น ลักษณะทั่วไปไม่ใช่เฉพาะเจาะจง

3.4 Depends On เป็นความสัมพันธ์ที่ชี้ให้เห็นถึง source หรือ client ขึ้นอยู่กับ Destination หรือ supplier เช่นใน Class หนึ่งสามารถมี Method ที่ขึ้นอยู่กั Class อื่น เวลาทำงานจะมีการอ้างถึง Class นั้นด้วย

การติดต่อสื่อสารระหว่างออบเจกต์ใช้ Message Passing คือการที่ออบเจกต์ ติดต่อกันด้วยการส่งข้อความ (Message) ถึงกันและกันซึ่งข้อความจะประกอบด้วยจุดหมายปลายทาง (Destination) ของข้อความนั้นและข้อมูลที่สำคัญ (Argument หรือ Parameter) Message Passing เปรียบได้กับ Function call หรือ Procedure call ที่มีใน Structured programming โดยผ่าน Interface method ของออบเจกต์ นั้นๆ มีผลทำให้ออบเจกต์ ที่เป็นผู้รับข้อความ (Received Object) นั้นกระทำการอย่างใดอย่างหนึ่ง

2.3 UML (Unified Modeling Language)

นับตั้งแต่เริ่มมีการพัฒนาเทคโนโลยีซอฟต์แวร์มาจนถึงปัจจุบัน ได้มีวิวัฒนาการด้านเครื่องมือ (Tools) ตลอดจนภาษา (Language) ที่ใช้ในการพัฒนาโปรแกรม (Programming) มาเป็นลำดับ และเทคโนโลยีด้านการพัฒนาโปรแกรมที่น่าสนใจที่สุดในขณะนี้ก็คือ การพัฒนาการโปรแกรมแบบวัตถุ หรือ OOP ซึ่งมีประสิทธิภาพ และพัฒนาได้อย่างรวดเร็ว ในขณะที่มีต้นทุนที่ต่ำลง เนื่องจากสามารถนำเอาโปรแกรมที่เคยสร้างไว้กลับมาใช้ใหม่ได้ ทำให้โปรแกรมที่ได้พัฒนาขึ้นจากเทคโนโลยีนี้เป็นที่น่าสนใจอย่างยิ่ง อย่างไรก็ตามการวิเคราะห์ระบบงานที่สนับสนุนการพัฒนาโปรแกรมด้วยเทคโนโลยีนี้ยังไม่เป็นที่แพร่หลาย และยังใหม่อยู่มากโดยเฉพาะกับประเทศไทย แต่สิ่งที่กำลังจะเป็นมาตรฐานใหม่ในการวิเคราะห์ระบบงานก็คือ UML กล่าวแบบสรุป UML (Unified Modeling Language) ก็คือ วิธีการ (Methodology) ที่ใช้ในการจำลองรูปแบบของระบบงานเชิงวัตถุ ซึ่งเราจะได้อธิบายในรายละเอียดกันต่อไป

2.3.1 ความเป็นมาของ UML

การโปรแกรมใน 3GL (3rd Generation Language) ยุคแรก เริ่มต้นยุคจากภาษา BASIC (Beginner's All Purpose Symbolic Instruction Code) มองจากรากศัพท์ของคำว่าภาษา BASIC

ความหมายที่เห็นได้ชัดคือ เป็นภาษาที่เน้นให้ใช้งานได้เอนกประสงค์ และเหมาะสำหรับผู้เริ่มใช้

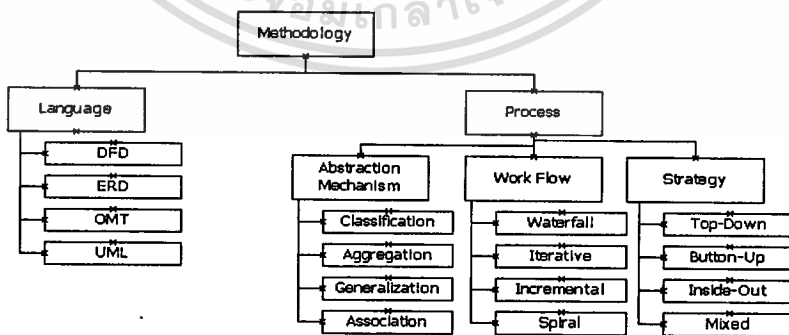
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาดเห็นาไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือสรุปได้ว่าเขียนไม่ยากนั่นเอง ภาษา BASIC ได้รับความนิยมอย่างแพร่หลายเมื่อไมโครซอฟท์ ได้ออกตัวแปลภาษา BASIC มาในนาม ควิกเบสิก (Quick Basic) ซึ่งก็ยังคงข้อเสียของการโปรแกรมด้วยภาษาเบสิกไว้คือ การโปรแกรมไม่เป็นแบบโครงสร้าง (Structure) เพราะจะมีการใช้ลาเบล (Label) ร่วมกับคำสั่ง GOTO เพื่อกระโดดไปมายังจุดต่างๆ ในโปรแกรม

แต่เมื่อมีผู้เชี่ยวชาญมาเสนอให้เห็นถึงผลเสียดังกล่าว จึงเริ่มมีการพัฒนาภาษาสำหรับเขียนโปรแกรมให้มีโครงสร้างที่ชัดเจนยิ่งขึ้น จากนั้นจึงเริ่มเข้าสู่ยุคปลายของ 3GL อย่างเช่น C หรือ Pascal ซึ่งใช้งานแพร่หลายมากกว่า QBASIC โดยมีการนำมาใช้ร่วมกับฐานข้อมูลแบบสัมพันธ์ (RDBMS) คือมีการสร้าง Context Diagram, DataFlow Diagram, ER (Entity Relation) จนถึงการวิเคราะห์และแมปไปเป็นโครงสร้างแบบสัมพันธ์ หรือตารางแบบ 2 มิติ (Table) นั่นเอง แต่ความต้องการของมนุษย์ก็มีไม่สิ้นสุด ดังนั้นจึงเริ่มหาวิธีการสร้างโปรแกรม และการเก็บข้อมูลที่ใช้งานง่าย ปรับปรุงได้ง่าย รวมทั้งเห็นการสื่อสารระหว่างกันได้ชัดเจนขึ้น มีการสร้างแนวคิดในการโปรแกรมแบบใหม่คือ การโปรแกรมเชิงวัตถุ หรือ OOP (Object-Oriented Programming) ซึ่งมีพื้นฐานอยู่บนคุณสมบัติในการทำงานร่วมกับออบเจกต์ (Object) เช่น พร็อพเพอร์ตี้ (Properties), เมธอด (Method), เอ็นแคปซูลชัน (Encapsulation), โอเวอร์โหลด (Overload), การถ่ายทอดคุณสมบัติ (Inheritance) และ Polymorphism เป็นต้น ซึ่งในช่วงแรกก็มีผู้เสนอวิธีการเชิงออบเจกต์เหล่านี้ขึ้นมามากมาย เช่น OMT (Object Modeling Technique), Object-Oriented Method, OOSE (Object-Oriented Software Engineering) และ OOSC (Object-Oriented Software Construction) เป็นต้น

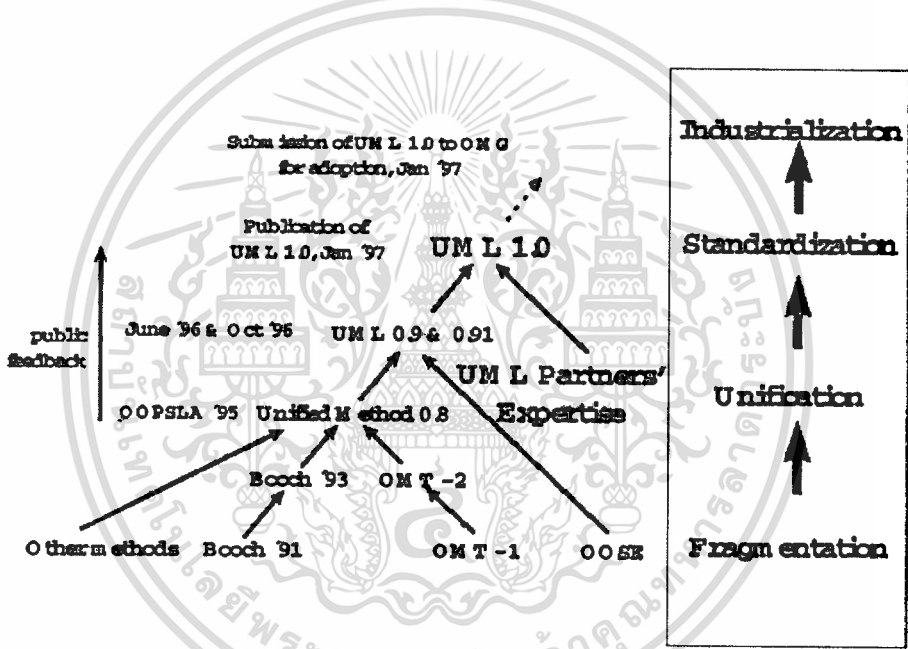
โดยทั่วไปวิธีการ (Methodology) หมายถึงระบบ ระเบียบ กฎ และวิธีปฏิบัติของศาสตร์ต่างๆ แต่ในศาสตร์คอมพิวเตอร์ เราสามารถจำแนกออกเป็น 2 ส่วนหลักได้ดังรูปที่ 2.1



รูปที่ 2.1 แสดงการจำแนก Methodology

จะเห็นได้ว่า UML ถูกจำแนกออกเป็นส่วนหนึ่งในรูปแบบของภาษา บางครั้ง UML จึงถูกเรียกว่า “ภาษาพิมพ์เขียวของซอฟต์แวร์” หรือ “ภาษาแบบจำลองซอฟต์แวร์” กระบวนการพัฒนา

ซอฟต์แวร์หนึ่งๆ จะประกอบด้วย Methodology ทั้ง 2 ส่วนคือตัวภาษาที่ใช้ในการบรรยายแบบจำลอง (Modeling Language) และกระบวนการ (Process) UML เป็นภาษาสำหรับบรรยายแบบจำลองเพื่อใช้สื่อสารกับผู้ที่ใช้วิธีการต่างๆ ซึ่งถึงแม้ว่าความคิดในเชิงออบเจกต์ของแต่ละวิธีการจะมีความคล้ายคลึงกัน แต่ก็จำเป็นต้องมีแผนผังที่ใช้บรรยายความคิดออกมาให้เป็นรูปแบบเดียวกัน UML เปรียบเสมือนขั้นแรกในการพัฒนาโปรแกรมแบบวัตถุ โดยเป็นผลมาจากการพัฒนาของ Jim Rumbaugh (ผู้เขียนหนังสือ OMT: Object Modeling Technique ในปี 1991), Gredy Booch (จากบริษัท Rational Software) และ Ivar Jacobson (ผู้เขียนหนังสือ OOSE: Object-Oriented Software Engineering ในปี 1994) แผนผังความเป็นมาของ UML แสดงในรูปที่ 2.2



รูปที่ 2.2 ความเป็นมาของ UML

UML เป็นภาษาที่ใช้ในการบรรยายแบบจำลองทางซอฟต์แวร์ จากแผนผังขอ Methodology ในรูปที่ 2.1 จะเห็นได้ว่าการใช้ UML ไม่ได้มีการเกี่ยวข้องใดๆ กับกระบวนการในการตัดสินใจและวิเคราะห์ออกแบบพัฒนาซอฟต์แวร์แต่อย่างใด และที่สำคัญ UML ไม่ใช่ภาษาที่ใช้ในการเขียนโปรแกรม (Programming Language) แต่อาจจะมามีเครื่องมือประเภท CASE (Computer Aided Software Engineering) บางตัวที่สามารถสร้างโมเดลแบบ UML ให้ออกมาเป็นออบเจกต์ที่นำไปใช้ในภาษาที่ใช้เขียนโปรแกรมได้ (ตัวอย่างเช่น Rational Rose ที่สร้างออบเจกต์ให้กับ C++ หรือ VB ได้เป็นต้น) UML เป็นรูปแบบที่รวมข้อดีของรูปแบบอื่นๆ เข้าไว้ด้วยกันคือ Data Modeling Concept (Entity Relation Concept), Business Modeling (Work Flow), Object Modeling และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้ใช้ประโยชน์ในการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Component Modeling นอกจากนี้ UML ยังสามารถใช้ได้ตลอดทั้งโครงการหรือการทำงาน เช่นเดียวกับวงจรพัฒนาซอฟต์แวร์ SDLC (Software Development Life Cycle) ปัจจุบัน UML ได้รับการรับรองจาก OMG (Object Management Group) ซึ่งเป็นองค์กรกำหนดมาตรฐานทางเทคโนโลยี ออบเจกต์ ให้ UML เป็นมาตรฐานหนึ่งที่ไม่มีการเป็นเจ้าของ และเป็นรูปแบบเปิด กฎเกณฑ์ที่ทำให้ UML ประสบความสำเร็จคือ UML มีข้อดีในเรื่องรูปแบบของภาษา เช่นเดียวกับภาษาอื่นที่เคยเสนอมาในอดีต รวมทั้งเป็นการรวมรูปแบบที่ใช้ได้ในหลายๆ ระบบ (Business & Software) และขั้นตอนการพัฒนา (Development Phase) เข้าด้วยกันชนิดของแผนผังใน UML แบ่งได้โดยแอมของ UML ออกเป็นโมเดลต่างๆ ได้ดังนี้คือ ฟังก์ชันนอลโมเดล (Functional Model), ออบเจกต์โมเดล (Object Model) และไดนามิกโมเดล (Dynamic Model)

ฟังก์ชันนอลโมเดล ฟังก์ชันนอลโมเดลเป็นโมเดลที่ใช้แสดงความต้องการของระบบทั้งหมด, ช่วยในการอธิบายรายละเอียดหลักๆ ภายในออบเจกต์ และแสดงให้เห็นการไหลของข้อมูลในแต่ละการทำงาน โดยจะสนใจเพียงแค่ว่ามีงานอะไรบ้างที่ต้องทำ แต่จะยังไม่สนใจว่างานนั้นๆ ทำอย่างไร เช่น ในการพิมพ์รายงานสรุปประจำเดือน จะต้องรู้ว่าใครเป็นผู้ที่รับผิดชอบรายงานนี้ รูปร่างหน้าตาของรายงานเป็นอย่างไร มีข้อมูลอะไรบ้างที่สนใจ และเมื่อรายงานเสร็จแล้ว ใครเป็นผู้ที่รับรายงานนั้นเป็นต้น โดยเครื่องมือที่ใช้ในการแสดงความต้องการของระบบคือ ยูสเคสไดอะแกรม (Use Case Diagram) ซึ่งเป็นไดอะแกรมที่ใช้แสดงความต้องการของระบบทั้งหมดในลักษณะที่ผู้ใช้งานเข้าใจได้ง่าย และจะถูกนำไปใช้ต่อไปในเฟสของการวิเคราะห์และออกแบบระบบ

ออบเจกต์โมเดล เป็นโมเดลที่ใช้แสดงโครงสร้างของระบบ โดยจะแสดงในรูปของคลาส (Class) ต่างๆ ที่ได้จากการพิจารณาความต้องการของระบบในฟังก์ชันนอลโมเดล เครื่องมือที่ใช้ในการแสดงโครงสร้างของระบบจะประกอบด้วย คอมโพเนนต์ไดอะแกรม (Component Diagram) และคลาสไดอะแกรม (Class Diagram) ซึ่งคอมโพเนนต์ไดอะแกรมจะแสดงถึงความสัมพันธ์ระหว่างคอมโพเนนต์ต่างๆ ภายในระบบ ส่วนคลาสไดอะแกรมก็ใช้แสดงถึงส่วนต่างๆ ดังนี้คือ ลักษณะของแอตทริบิวต์ว่าเป็นข้อมูล (Data) หรือตัวแปร (Variable), หน้าที่การทำงาน (Operation) ซึ่งหมายถึงเมธอด (Method) ภายในคลาส รวมไปถึงความสัมพันธ์ระหว่างคลาสต่างๆ ภายในระบบ

ไดนามิกโมเดล ไดนามิกโมเดล เป็นโมเดลที่ใช้แสดงถึงการทำงานระหว่างออบเจกต์ต่างๆ ตามการส่งข้อความ (Message) หรือเมื่อเหตุการณ์ (Event) ต่างๆ ได้เกิดขึ้น ออบเจกต์ในที่นี้หมายถึงอินสแตนซ์ (Instance) ที่สร้างขึ้นจากคลาสที่ได้ออกแบบไว้ในออบเจกต์โมเดล โดยที่แต่ละออบเจกต์จะมีคุณสมบัติและพฤติกรรมเช่นเดียวกับคลาสดั้งเดิม ซึ่งการทำงานของระบบจะประกอบขึ้นจากการส่งข้อความไปมาระหว่างออบเจกต์เหล่านั้น และเมื่อระบบทำงานต่อไปเรื่อยๆ

แล้วออบเจกต์อาจจะเปลี่ยนสถานะ (State) ไปตามเหตุการณ์ที่เกิดขึ้นได้ เพื่อให้เป็นไปตามที่กำหนดไว้ในฟังก์ชันนอลโมเดล

เครื่องมือที่ใช้ในการสร้างโมเดลนี้คือ ซีควেনซ์ไดอะแกรม (Sequence Diagram) และสเตตไดอะแกรม (State Diagram) โดยซีควেনซ์ไดอะแกรมจะใช้แสดงลำดับขั้นตอนในการทำงาน (เมื่อเหตุการณ์หนึ่งเกิดขึ้นแล้วออบเจกต์ต่างๆ จะทำงานต่อไปอย่างไร) ซึ่งวัตถุประสงค์หลักของซีควেনซ์ไดอะแกรมก็คือ เพื่อให้ผู้เขียนโปรแกรมสามารถพัฒนาระบบได้ตามที่ออกแบบไว้ ดังนั้นจึงนับได้ว่าซีควেনซ์ไดอะแกรมเป็นเครื่องมือที่สามารถนำไปศึกษา เพื่อพัฒนาหรือขยายขีดความสามารถของโปรแกรมต่อไป ส่วนสเตตไดอะแกรมก็เป็นส่วนที่แสดงให้เห็นถึงสถานะทั้งหมดที่เป็นไปได้ และเหตุการณ์ที่ทำให้เกิดการเปลี่ยนสถานะของออบเจกต์แต่ละตัว

2.3.2 แผนผังใน UML

UML ใช้วิธีอธิบายแบบจำลองซอฟต์แวร์ด้วยองค์ประกอบ 2 ส่วนคือเมตะโมเดล (Meta Model) และ Notation โดยเมตะโมเดลคือแผนผังที่ใช้บรรยายโครงสร้างและเครื่องหมายต่างๆ ซึ่งจะมีโครงสร้างที่เป็นระเบียบตายตัว ส่วน Notation ก็คือไดอะแกรมต่างๆ ที่ใช้ในการบรรยายแบบจำลอง โดยสามารถแบ่งไดอะแกรมใน UML ออกเป็น 4 หมวด รวม 8 ชนิดได้ดังนี้

1. Use Case Diagram <Functional Model>
2. Class Diagram <Object Model>
3. Behavior Diagrams:
 - 3.1 Statechart Diagram <Dynamic Model>
 - 3.2 Activity Diagram <Dynamic Model>
 - 3.3 Interaction Diagrams
 - 3.3.1 Sequence Diagram <Dynamic Model>
 - 3.3.2 Collaboration Diagram <Dynamic Model>
4. Implementation Diagrams:
 - 4.1 Component Diagram <Object Model>
 - 4.2 Deployment Diagram <Dynamic Model>

2.3.2.1 ยูสเคสไดอะแกรม

ยูสเคสไดอะแกรมเป็นแนวคิดของไอวาร์ จาคอบสัน (Ivar Jacobson) ซึ่ง OMG ได้รวมไว้ในมาตรฐานของ UML ด้วย และแนวคิดนี้ก็ถูกนำไปใช้อย่างกว้างขวาง ยูสเคสไดอะแกรมเป็นสิ่งที่

ใช้แสดงความต้องการของระบบทั้งหมดในลักษณะที่ผู้ใช้งานเข้าใจได้ง่าย โดยจะแสดงความสัมพันธ์ของการโต้ตอบกันระหว่างผู้ใช้กับตัวระบบ หรืออาจกล่าวได้ว่า ยูสเคสไดอะแกรมนั้นใช้แสดงการทำงานเด่นๆ ของซอฟต์แวร์ที่จะสร้างขึ้นนั่นเอง และยูสเคสไดอะแกรมก็เป็นจุดเริ่มต้นในการรวบรวมความต้องการของผู้ใช้ในขั้นแรก ซึ่งข้อดีของวิธีนี้ก็คือ สามารถกำหนดขอบเขตของฟังก์ชันได้อย่างชัดเจน และเข้าถึงเป้าหมายที่แตกต่างกันของผู้ใช้ได้ สัญลักษณ์ที่ใช้ในยูสเคสไดอะแกรมมีดังนี้

แอ็กเตอร์ (Actor) คือสิ่งที่อยู่ภายนอกระบบที่จะพัฒนาขึ้นทั้งหมด ที่ต้องการแลกเปลี่ยนหรือส่งข้อมูลให้กับระบบ (สร้าง Role) โดยที่แอ็กเตอร์อาจจะเป็นคน ระบบ หรือโปรแกรมอื่นๆ ก็ได้ (ส่วนใหญ่จะเป็นผู้เริ่มทำงานกับยูสเคส) สัญลักษณ์ที่ใช้แทน Actor ดังรูปที่ 2.3 แอ็กเตอร์ไม่เหมือนกับผู้ใช้ระบบ เพราะผู้ใช้ระบบหนึ่งคนอาจจะแสดงได้หลายบทบาท เช่นผู้ที่ล็อกอินเข้ามาทำงานอย่างหนึ่ง อาจจะทำงานของผู้ใช้ในอีกฐานะหนึ่งไปพร้อมๆ กันก็ได้

รูปที่ 2.3 สัญลักษณ์ของแอ็กเตอร์

ยูสเคส (Use Case) จะเป็นตัวแทนกิจกรรมที่เกิดขึ้นในขั้นตอนต่างๆ โดยจะใช้สัญลักษณ์เป็นรูปวงรี หรือวงกลม ดังรูปที่ 2.4 ถ้ายูสเคสใดมีกรอบสี่เหลี่ยมสีเทาล้อมรอบแสดงว่ามียูสเคสไดอะแกรมย่อยที่ใช้อธิบายรายละเอียดต่อไป



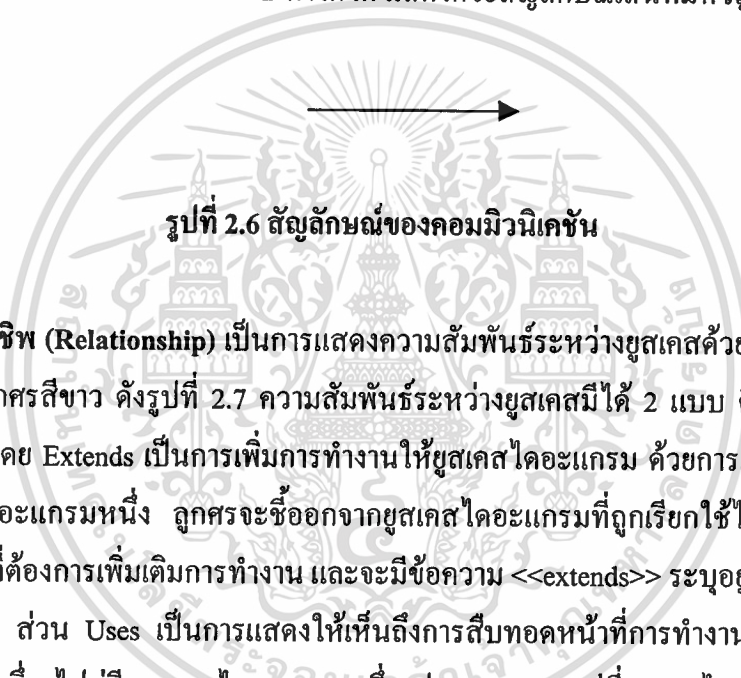
รูปที่ 2.4 สัญลักษณ์ของยูสเคส

ซิสเต็ม (System) เป็นตัวแทนของระบบที่ถูกกระทำโดยแอ็กเตอร์ ซึ่งใช้สัญลักษณ์เป็นรูปสี่เหลี่ยม ดังรูปที่ 2.5 ภายในจะประกอบไปด้วยยูสเคสต่างๆ ตัวอย่างเช่น ระบบเงินเดือน (Payroll System) เป็นต้น



รูปที่ 2.5 สัญลักษณ์ของซิสเต็ม

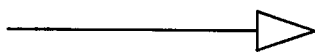
คอมมิวนิเคชัน (Communication) เป็นการแสดงความสัมพันธ์หรือการติดต่อสื่อสารกัน (การรับและให้ข้อมูลข่าวสารแก่กันและกัน) ระหว่างแอ็กเตอร์และยูสเคส ซึ่งอาจจะเป็น การสื่อสารแบบทางเดียวหรือ 2 ทางก็ได้ แสดงด้วยสัญลักษณ์เส้นที่มีหัวลูกศรสีดำ ดังรูปที่ 2.6



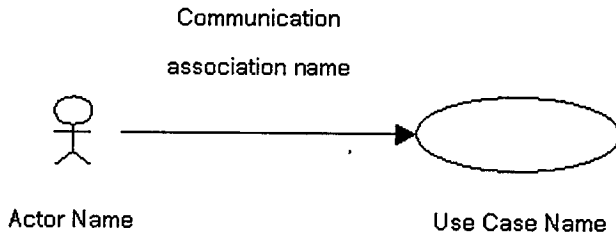
รูปที่ 2.6 สัญลักษณ์ของคอมมิวนิเคชัน

รีเลชันชิพ (Relationship) เป็นการแสดงความสัมพันธ์ระหว่างยูสเคสด้วยกัน โดยใช้เส้นที่มีหัวลูกศรสีขาว ดังรูปที่ 2.7 ความสัมพันธ์ระหว่างยูสเคสมีได้ 2 แบบ คือ Extends และ Uses โดย Extends เป็นการเพิ่มการทำงานให้ยูสเคสไดอะแกรม ด้วยการเรียกใช้จากอียูสเคสไดอะแกรมหนึ่ง ลูกศรจะชี้ออกจากยูสเคสไดอะแกรมที่ถูกเรียกใช้ไปยังยูสเคสไดอะแกรมที่ต้องการเพิ่มเติมการทำงาน และจะมีข้อความ <<extends>> ระบุอยู่ข้างๆ เส้น

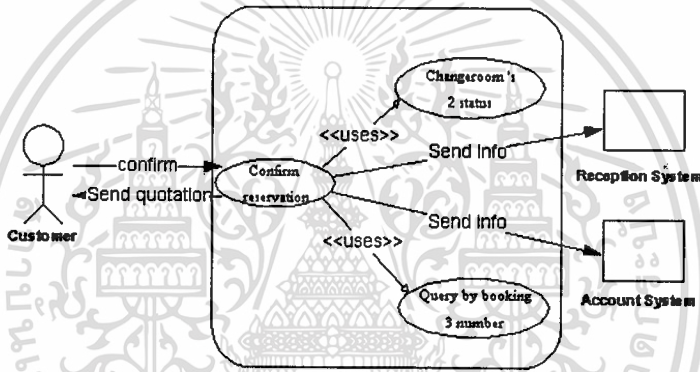
ส่วน Uses เป็นการแสดงให้เห็นถึงการสืบทอดหน้าที่การทำงานจากยูสเคสไดอะแกรมหนึ่ง ไปสู่อียูสเคสไดอะแกรมหนึ่ง ปลายลูกศรจะอยู่ที่ยูสเคสไดอะแกรมหลักที่จะถูกถ่ายทอดความสามารถออกไป โดยจะมีข้อความ <<uses>> ปรากฏอยู่ข้างๆ เส้น ดังตัวอย่างการใช้งานในรูปที่ 2.9



รูปที่ 2.7 สัญลักษณ์ของรีเลชันชิพ



รูปที่ 2.8 รูปแบบของยูสเคสไดอะแกรม



รูปที่ 2.9 ตัวอย่างยูสเคสไดอะแกรมของระบบจองห้องพัก

2.3.2.2 คลาสไดอะแกรม

คลาสไดอะแกรมเป็นสแตติกโมเดลชนิดหนึ่งที่ใช้อธิบายถึงคลาส และความสัมพันธ์ระหว่างคลาส ที่มีโครงสร้างของข้อมูลรวมถึงพฤติกรรมของข้อมูลที่แตกต่างกัน ซึ่งคลาสหนึ่งๆ นั้นสามารถกำหนดทิศทางการนำไปสร้างโปรแกรม และการสร้างคลาสอื่นๆ ใน OOP ได้ ตัวคลาสใช้สัญลักษณ์เป็นรูปสี่เหลี่ยม และจะอธิบายถึงกลุ่มของออบเจกต์ต่างๆ ที่มีแอตทริบิวต์, หน้า ที่การทำงาน (Operation) และความสัมพันธ์ (Relationship) ขึ้นพื้นฐาน ซึ่งในแต่ละคลาสประกอบไปด้วยรายละเอียดพื้นฐาน 3 ส่วนคือ ชื่อคลาส (Name Compartment), แอตทริบิวต์ (Attribute Compartment) และโอเปอเรชัน (Operation Compartment) ดังแสดงในรูปที่ 2.10

ส่วนที่เป็นชื่อคลาสจะใช้แสดงชื่อของคลาสที่กำหนดไว้ในระบบ และแอตทริบิวต์ก็เป็นการกำหนดคุณลักษณะทั้งหมดภายในคลาส ซึ่งแอตทริบิวต์ภายในคลาสเดียวกันจะมีชื่อไม่ซ้ำกัน

แต่ชื่อของข้อมูลในคลาสอาจจะซ้ำกับชื่อของข้อมูลในคลาสอื่นได้ อย่างสุดท้ายคือโอเปอเรชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นส่วนที่ใช้อธิบายว่าในคลาสนั้นมีเมธอดอะไรบ้าง มีการรับค่าชุดของตัวแปร (Argument) อย่างไร หรือมีการส่งค่าออกไปหรือไม่

ชื่อคลาส
แอดทริบิวต์
โอเปอเรชัน

รูปที่ 2.10 ส่วนประกอบพื้นฐานของคลาส

จากรูปที่ 2.10 แอดทริบิวต์ และ โอเปอเรชัน จะมีการแสดงสิทธิการเข้าถึงข้อมูลและโอเปอเรชันของคลาส ดังรายละเอียดต่อไปนี้

สัญลักษณ์	ความหมาย	คำอธิบาย
-	Private	เข้าถึงได้เฉพาะตัวเอง, ชับคลาส และ Friend
+	Public	เข้าถึงได้ทุกคลาส
#	Protected	เข้าถึงได้เฉพาะตัวเอง และชับคลาส
?	Implementation	ขอบเขตถูกตัดตัดสินใจในขณะที่ใช้โปรแกรม

ตารางที่ 2.1 การเข้าถึงข้อมูลในคลาส

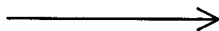
ความสัมพันธ์ระหว่างคลาส

UML ได้เตรียมความสัมพันธ์ระหว่างคลาสไว้ (Relationship) ซึ่งมีความสัมพันธ์อยู่ 4 แบบ ใช้สัญลักษณ์ดังนี้

- Association : จะแสดงความสัมพันธ์ระหว่างคลาสในลักษณะของการรู้จักกัน และมีบทบาทของคลาส (Role) ใน 2 ทิศทางคือ ถ้ามี Association ระหว่าง A ไป B จะมีทั้งโรลจาก A ไป B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ B ไป A เกิดขึ้น ดังนั้นสัญลักษณ์จะมีได้ทั้งทางเดียวหรือ 2 ทาง และอาจมีการกำกับชื่อโรลไว้ที่เส้นด้วยก็ได้ ดังรูปที่ 2.11 และ 2.12



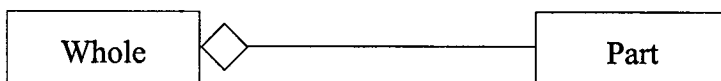
รูปที่ 2.11 ความสัมพันธ์แบบทางเดียว

รูปที่ 2.12 ความสัมพันธ์แบบสองทาง (ไม่นิยมเขียนหัวลูกศรกำกับ)

- Aggregation : เป็นรูปแบบพิเศษของ Association คือเป็นความสัมพันธ์ระหว่าง Whole และ Part ของมัน คือ Whole จะประกอบไปด้วย Part ต่างๆ ดังนั้นการคงอยู่ของ Part จะต้องขึ้นกับ Whole มีสัญลักษณ์ดังรูปที่ 2.13 , 2.14 และ 2.15

รูปที่ 2.13 ความสัมพันธ์แบบ by Value

รูปที่ 2.14 ความสัมพันธ์แบบ by Reference



รูปที่ 2.15 ความสัมพันธ์ Whole – Part และมีความสัมพันธ์แบบ by Reference

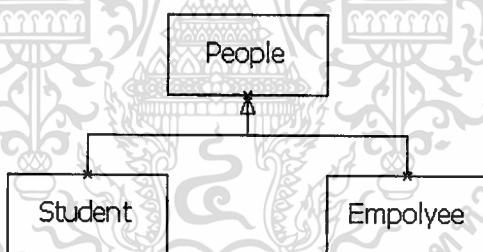
- Depends on : หรือ Coupling เป็นรูปแบบของความสัมพันธ์ประเภทหนึ่งที่ใช้แสดงความสัมพันธ์กันระหว่างคลาส 2 คลาส ในแง่ที่คลาสหนึ่งเรียกใช้บริการของอีกคลาสหนึ่ง กล่าวคือ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสของผู้ขอบริการขึ้นอยู่กับบริการในคลาสของผู้ให้บริการ แต่ไม่มีการขึ้นต่อกันภายในโครงสร้างของคลาส คือไม่ได้มี Association กันโดยตรงแต่มีการใช้งานที่ขึ้นตรงต่อกัน เมื่อมีการแก้ไขคลาสหนึ่งจะส่งผลกระทบต่อคลาสที่ Depends on ด้วย ใช้สัญลักษณ์ดังรูปที่ 2.16

รูปที่ 2.16 ความสัมพันธ์ Depends on

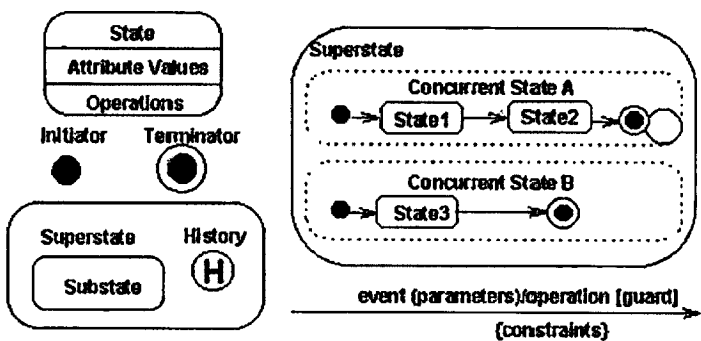
- Generalization : หรือการสืบทอดความสัมพันธ์ (Inheritance) ความสัมพันธ์รูปแบบนี้ใช้แสดงความสัมพันธ์ระหว่างคลาสกับคลาส ในแง่ที่คลาสหนึ่งถ่ายทอดคุณสมบัติและโครงสร้างจากอีกคลาสหนึ่ง โดยเรียกคลาสที่ถูกถ่ายทอดว่าซูเปอร์คลาส (Super Class) และเรียกคลาสต้นทางว่าซับคลาส (Sub Class) มีสัญลักษณ์ดังนี้ (หัวของลูกศรจะอยู่ในทิศชี้จากซับคลาสไปที่ซูเปอร์คลาส) ดังแสดงในรูปที่ 2.17



รูปที่ 2.17 ความสัมพันธ์ Generalization

2.3.2.3 สเตทไดอะแกรม

เป็นวงจรชีวิตของออบเจกต์, ซับซิสเต็ม และระบบต่างๆ ซึ่งสเตทจะเป็นตัวบ่งบอกถึงเหตุการณ์ต่างๆ ว่ามีผลกระทบอะไรเกิดขึ้นบ้าง สเตทไดอะแกรม (State Diagram) จะทำหน้าที่เชื่อมต่อกับทุกคลาสที่มีพฤติกรรมอันซับซ้อนให้ชัดเจนขึ้น และเป็นการอธิบายถึงพฤติกรรมของระบบ ซึ่งในแต่ละสเตทจะมีความแตกต่างกันขึ้นอยู่กับสถานะในปัจจุบัน รวมทั้งมีเหตุการณ์ไคบังที่มีผลกับการเปลี่ยนแปลงทั้งภายในและภายนอก โดยอาจจะมีจุดเริ่มต้นและจุดจบได้ในหลายๆ จุด ซึ่งมีสัญลักษณ์ดังรูปที่ 2.18



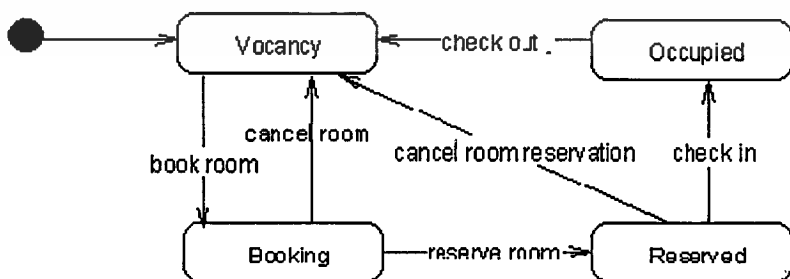
รูปที่ 2.18 สัญลักษณ์ที่ใช้ในสเตทไดอะแกรม (State Diagram)

จากรูปที่ 2.18 ภายใน State Diagram จะมีส่วนประกอบอยู่ 3 ส่วนคือ

1. State 's Name
2. Optional State เป็นคุณสมบัติของคลาสใน State Diagram
3. Optional Activity State เป็นรายละเอียดของเหตุการณ์ และแอ็กชัน (Action) ต่างๆ

State ที่ใช้เป็นมาตรฐาน ในการทำกิจกรรมต่างๆ มี 3 สถานะคือ

- Initiator คือการเข้าไปทำกิจกรรมต่างๆ ภายในสเตท หรือการเริ่มต้น (สัญลักษณ์แทนด้วยวงกลมทึบ)
- Terminator จะใช้ในกรณีที่ต้องการออกจากสเตท หรือสิ้นสุดการทำงาน (สัญลักษณ์แทนด้วยวงกลมทึบและมีอีก 1 วงล้อมรอบ)
- Do ใช้ในการปฏิบัติงานต่างๆ ในสเตทเช่น การส่งเมสเซจ, การรอ หรือการคำนวณ



รูปที่ 2.19 สเตทไดอะแกรมของออบเจ็กต์จองห้องพัก

จากรูปที่ 2.19 เป็นตัวอย่างสเตตไดอะแกรมของออบเจกต์ห้องพัก โดยที่ห้องพักจะมีสถานะที่เป็นไปทั้งหมด 4 สถานะคือ ว่าง (Vacancy), สำรอง (Booking), จอง (Reserved) และมีลูกค้าพักอยู่ (Occupied) โดยห้องพักจะเริ่มจากว่าง เมื่อมีการจองห้องพักแล้วห้องพักก็就会被สำรอง และเมื่อมีการยกเลิกการสำรองก็กลับเป็นห้องว่างได้ หรือเมื่อมีการยืนยันการจองแล้ว ห้องพักที่ถูกสำรองไว้ก็จะเปลี่ยนเป็นห้องพักที่จองแล้ว (ห้องพักที่มีการจองแล้วก็สามารถยกเลิกแล้วเปลี่ยนเป็นห้องว่างได้) หรือถ้าลูกค้าเข้าพักก็จะกลายเป็นห้องที่มีลูกค้าพักอยู่ สุดท้ายก็คือ ห้องที่มีลูกค้าพักอยู่จะกลับไปเป็นห้องว่างเมื่อลูกค้า Check out

2.3.2.4 แอ็กทिवิตีไดอะแกรม

ไดอะแกรมนี้ (Activity Diagram) ใช้แสดงถึงลำดับขั้นตอนการทำงานของกิจกรรมในระบบ ซึ่งจะสามารถใช้เป็นแบบในการปฏิบัติงานในหน้าที่ต่างๆ รวมทั้งยังอธิบายถึงขั้นตอนการทำงานในขั้นต่อไปของกิจกรรมอื่นด้วย แอ็กทिवิตีไดอะแกรมเป็นการพิจารณาถึงสถานะของการทำงานในแต่ละแอ็กชัน (Action) ซึ่งเมื่อเริ่มต้นทำแล้วก็ต้องทำให้เสร็จก่อนที่จะเริ่มทำงานในสถานะต่อไปได้ ดังนั้นสิ่งที่จะใช้ควบคุมการทำงานในขั้นตอนต่างๆ ก็คือ การติดต่อสื่อสารกันหรือการเชื่อมโยงกันของสเตตต่างๆ ซึ่งสามารถกระทำได้ทั้งการส่งและรับข้อมูลข่าวสารได้พร้อมๆ กัน โดยมีการนำแนวคิดมาจากหลายๆ เทคนิคเช่น Event Diagram, เทคนิค SDL State Modeling และ Petri Nets Diagram โดยนำมาใช้ประโยชน์ในการเชื่อมต่อกับ Flow และรายละเอียดของพฤติกรรมที่มีการประมวลผลแบบขนาน (Parallel) คือสามารถทำงานได้หลายอย่างในเวลาเดียวกันได้เป็นจำนวนมาก จุดประสงค์หลักของไดอะแกรมนี้ก็เพื่อแสดงถึงความเปลี่ยนแปลงอันเกิดจากการประมวลผลภายใน (ใช้แสดงเหตุการณ์ต่างๆ อันเป็นผลมาจากการกระทำภายในของระบบเอง) ในขณะที่สเตตไดอะแกรมจะแสดงการกระทำที่เกิดขึ้นในลักษณะที่ไม่ต่อเนื่องกัน

สัญลักษณ์ที่ใช้ในแอ็กทिवิตีไดอะแกรม

● Action State

ใน Activity Diagram จะประกอบด้วย Action State ซึ่งเป็นสถานะซึ่งมีกระบวนการทำงานภายในและมี transition อย่างน้อยหนึ่ง transition ออกจาก Action state เมื่อสิ้นสุดการกระทำภายใน state นั้น ซึ่งใน Action State จะไม่มีการเกิด transition ขึ้นภายใน หรือมี transition ออกจาก state อันเป็นผลมาจากเหตุการณ์ภายนอก การประยุกต์ใช้โดยทั่วไปของ Action State ก็คือเพื่อแสดงขั้นตอนในการทำงานของอัลกอริทึม (Algorithm) สัญลักษณ์ของ Action State มีลักษณะดังรูปที่ 2.20

action

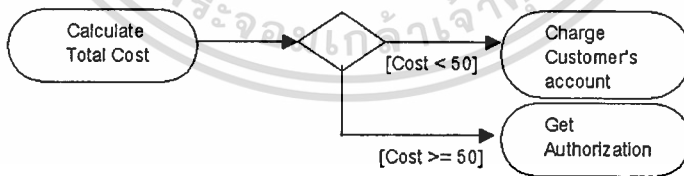
รูปที่ 2.20 Action State

ซึ่ง Action ที่แสดงไว้ภายใน Action State อาจใช้ภาษาทั่วไป, pseudocode หรือ ภาษาโปรแกรมก็ได้ โดยอาจมีการใช้ attribute หรือ link ของออบเจกต์ ที่เป็นเจ้าของ action ดังกล่าวได้

แอกทิวิตี้โคอะแกรมจะแสดงลักษณะของการตัดสินใจตามเงื่อนไขต่างๆ เพื่อกำหนดเส้นทางของการทำงานตามเงื่อนไขเหล่านั้น สำหรับสัญลักษณ์ที่ใช้แสดงการตัดสินใจแทนด้วยรูปข้าวหลามตัดดังรูปที่ 2.21

รูปที่ 2.21 สัญลักษณ์แสดงการตัดสินใจ

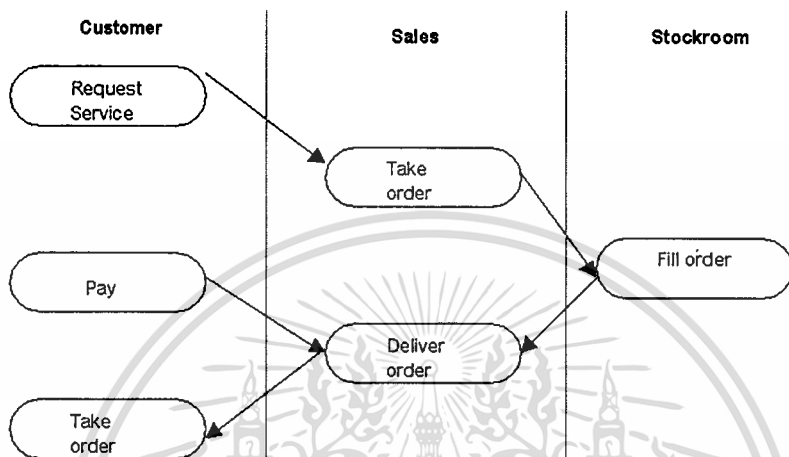
สำหรับ Input ของการตัดสินใจจะใช้รูปเส้นหัวลูกศรเข้าสู่สัญลักษณ์ และ output จะใช้เส้นหัวลูกศรออกจากสัญลักษณ์ โดยจะมีกำกับไว้ด้วยเงื่อนไขของแต่ละเส้นทางออกจากสัญลักษณ์ สำหรับทุกเงื่อนไขที่เป็นไปได้ ดังตัวอย่างรูปที่ 2.22



รูปที่ 2.22 การตัดสินใจ

- **Swimlanes**

Action ที่แสดงใน Diagram อาจมีการจัดให้อยู่ในลักษณะของถู่ (Swimlanes) เพื่อแสดงขอบเขตความรับผิดชอบของกิจกรรมต่างๆ โดยแต่ละ Class ที่เกี่ยวข้อง โดยมีการใช้ transition เพื่อเชื่อมโยงแต่ละกิจกรรมเหล่านั้นเข้าด้วยกันดังรูปที่ 2.23



รูปที่ 2.23 Swimlanes

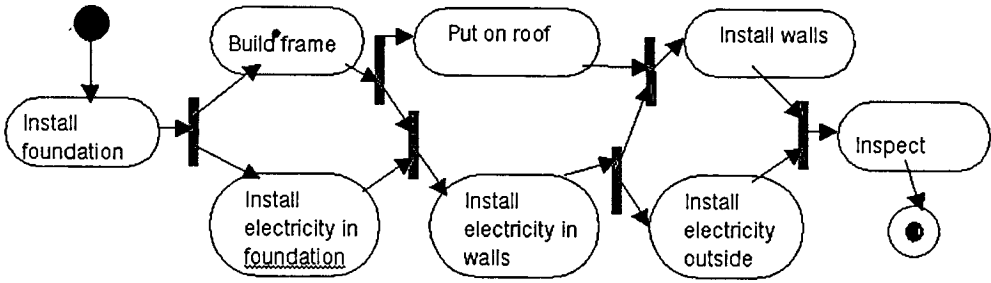
ลูกค้าส่งการสั่งซื้อสินค้ามาให้กับ Sales รับใบสั่งซื้อส่งให้กับคลังสินค้า และให้สินค้ามาเตรียมจัดส่งมาที่ Sales เมื่อลูกค้าจ่ายเงินก็จะส่งสินค้า

- **Sync States**

Sync States จะแสดงถึงการเชื่อมต่อที่ใช้เพื่อแสดงถึงการแยกของ transition หรือการกระทำที่ต้องมีการรอให้กิจกรรมใดกิจกรรมหนึ่งทำงานเสร็จลงก่อนซึ่งแสดงโดยใช้สัญลักษณ์ดังรูปที่ 2.24

รูปที่ 2.24 สัญลักษณ์ Sync States

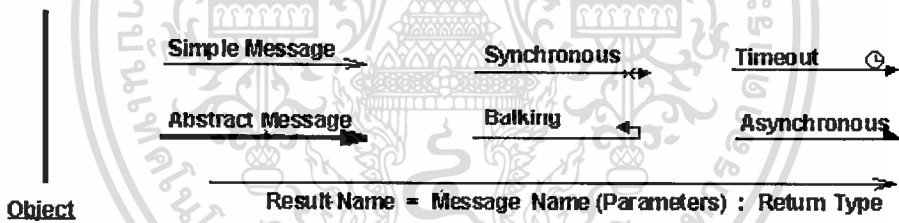
สำหรับตัวอย่างการใช้งาน Sync States สามารถแสดงได้จากตัวอย่างการสร้างบ้านซึ่งกิจกรรมบางอย่างสามารถทำไปพร้อมๆ กันได้ และบางกิจกรรมจะสามารถทำได้ก็ต่อเมื่อกิจกรรมก่อนหน้านี้ได้เสร็จสิ้นแล้วเท่านั้น ซึ่งสามารถแสดงโดยใช้เอกทิวตีโคอะแกรมได้ดังรูปที่ 2.25



รูปที่ 2.25 Sync States

2.3.2.5 ไคอะแกรมแสดงลำดับ

หรือเรียกว่าซีควเอนซ์ไคอะแกรม (Sequence Diagram) เป็นไคอะแกรมที่แสดงการทำงานระหว่างออบเจกต์ต่างๆ ตามการส่งข้อความ และเหตุการณ์ที่เกิดขึ้น ซึ่งผู้เขียนโปรแกรมจะใช้ไคอะแกรมนี้ เพื่อช่วยให้เขียนโปรแกรมได้ตามที่ออกแบบไว้



รูปที่ 2.26 สัญลักษณ์ที่ใช้ในไคอะแกรมแสดงลำดับ

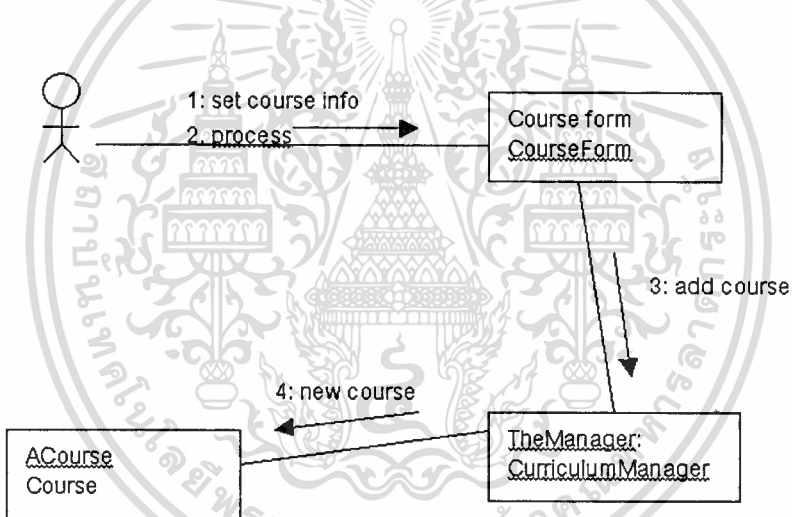
จากสัญลักษณ์ที่แสดงในรูปที่ 2.26

- ออบเจกต์ : คือ อินสแตนซ์ของคลาสที่ทำหน้าที่รับส่งเมสเซจ (Message) จากออบเจกต์อื่นๆ และตอบสนองตามข้อความนั้นๆ เพื่อให้เกิดการทำงานตามขั้นตอนของระบบ โดยจะใช้สัญลักษณ์เป็นเส้นตรงแนวตั้ง และมีชื่อของออบเจกต์กับคลาสอยู่ที่ด้านล่าง
- เมสเซจ : เป็นข้อความที่ส่งไปมาระหว่างออบเจกต์ที่ถูกเรียก ซึ่งจะประกอบไปด้วยส่วนต่างๆ ดังต่อไปนี้
- Result Name : คือตัวแปรที่จะจัดเก็บค่าที่ได้จากการส่งข้อความ (Message)
- Parameters : คือข้อตกลงต่างๆ ที่ส่งให้กับออบเจกต์ เพื่อทำงานตามเมธอดที่กำหนดด้วยข้อความนั้นๆ

- Return Type : คือประเภทของข้อมูลที่เมธอดส่งกลับมา
- Constraint : คือเงื่อนไขในการส่งข้อความ

2.3.2.6 คอลลาบอเรชันไดอะแกรม

คอลลาบอเรชันไดอะแกรม (Collaboration Diagram) นั้นใช้แสดงการติดต่อสื่อสารระหว่างออบเจกต์ที่สนใจ รวมทั้งความสัมพันธ์ระหว่างแต่ละออบเจกต์ด้วย ซึ่งจะมีลักษณะที่ใกล้เคียงกับไดอะแกรมแสดงลำดับมาก โดยในบางครั้งคุณอาจเลือกทำเพียงไดอะแกรมเดียวก็ได้ หลักเกณฑ์ในการพิจารณาก็คือ ถ้ามีการกำหนดช่วงของเวลาที่แน่นอน (เวลาเป็นสิ่งสำคัญที่สุด) ก็ควรเลือกใช้ไดอะแกรมแสดงลำดับ แต่ถ้าให้ความสำคัญกับรายละเอียดและเนื้อหาแล้ว ก็ควรเลือกใช้คอลลาบอเรชันไดอะแกรม ตัวอย่างคอลลาบอเรชันไดอะแกรมดังแสดง ในรูปที่ 2.27

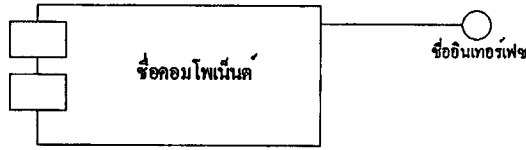


รูปที่ 2.27 ตัวอย่างการใช้คอลลาบอเรชันไดอะแกรมในระบบลงทะเบียน

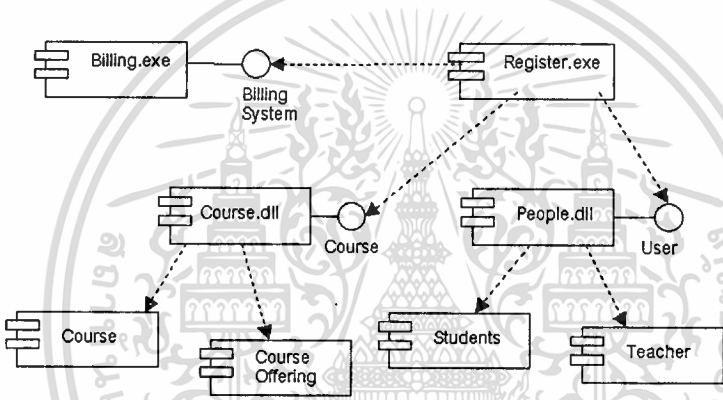
2.3.2.7 ไดอะแกรมแสดงองค์ประกอบ

หรือคอมโพเนนต์ไดอะแกรม (Component Diagram) มีหน้าที่แสดงโครงสร้าง และความสัมพันธ์กันระหว่างองค์ประกอบต่างๆ ของซอฟต์แวร์ ซึ่งองค์ประกอบดังกล่าวอาจเป็นโปรแกรมต้นฉบับ (Source Program), ส่วนของโปรแกรมที่ใช้เมื่อโปรแกรมทำงานเช่น ไลบรารี (Library) ต่างๆ หรืออาจเป็นโปรแกรมสำหรับทำงาน (Executable Program) ก็ได้ และในการประยุกต์ใช้กับธุรกิจนั้น คุณอาจผนวกเอาเอกสาร และกระบวนการทำงานของธุรกิจ เข้าเป็นส่วนหนึ่งของซอฟต์แวร์ก็ได้ ไดอะแกรมแสดงองค์ประกอบคือกราฟที่แสดงองค์ประกอบต่างๆ ของระบบที่

เชื่อมโยงกัน โดยใช้ความสัมพันธ์ในลักษณะ Dependency โดยจะแสดงในรูปของเส้นประที่มีหัว ลูกศรชี้จากคอมโพเนนต์ลูกไปยังคอมโพเนนต์หลัก ใน UML ดังแสดงในรูปที่ 2.28



รูปที่ 2.28 สัญลักษณ์แทนคอมโพเนนต์ และอินเทอร์เฟซ



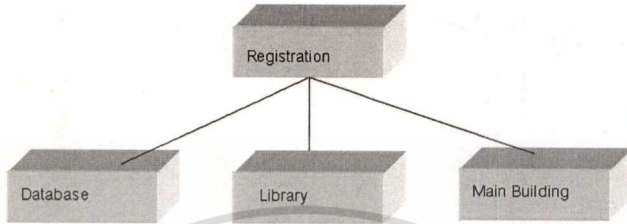
รูปที่ 2.29 ระบบลงทะเบียนที่ใช้คอมโพเนนต์ไคอะแกรม

จากรูปที่ 2.29 แสดงระบบลงทะเบียน ส่วนหลักคือระบบลงทะเบียน (Register.exe) ซึ่งมีความเกี่ยวข้องกับโปรแกรมอีก 3 ส่วนคือระบบเก็บเงิน (Billing.exe) ระบบรายวิชา (Course.dll) และบุคคลที่เกี่ยวข้อง (People.dll)

2.3.2.8 ดีพลอยเมนต์ไคอะแกรม

4ดีพลอยเมนต์ไคอะแกรมแสดงถึงส่วนประกอบต่างๆ และลักษณะของระบบในขณะที่ทำงาน รวมทั้งองค์ประกอบของซอฟต์แวร์, โพรเซส (Process) และออบเจกต์ (Object) ซึ่งไคอะแกรมนี้จะแสดงการจัดวางคอมโพเนนต์ได้อย่างชัดเจน โดยองค์ประกอบของซอฟต์แวร์ที่แสดงไว้จะมีเฉพาะที่ใช้ในขณะที่ระบบทำงานเท่านั้น และส่วนที่เหลือจะแสดงไว้ในคอมโพเนนต์ไคอะแกรม

ดีพลอยเมนต์ไดอะแกรมคือกราฟที่ประกอบด้วยโหนด (Node) ที่เชื่อมโยงกัน และติดต่อสื่อสารกัน แต่ละโหนดจะแสดงถึงหน่วยประมวลผล อันได้แก่อุปกรณ์คำนวณต่างๆ หรือมนุษย์ ดังตัวอย่างแสดงในรูปที่ 2.30



รูปที่ 2.30 การใช้ดีพลอยเมนต์ไดอะแกรมในระบบลงทะเบียน

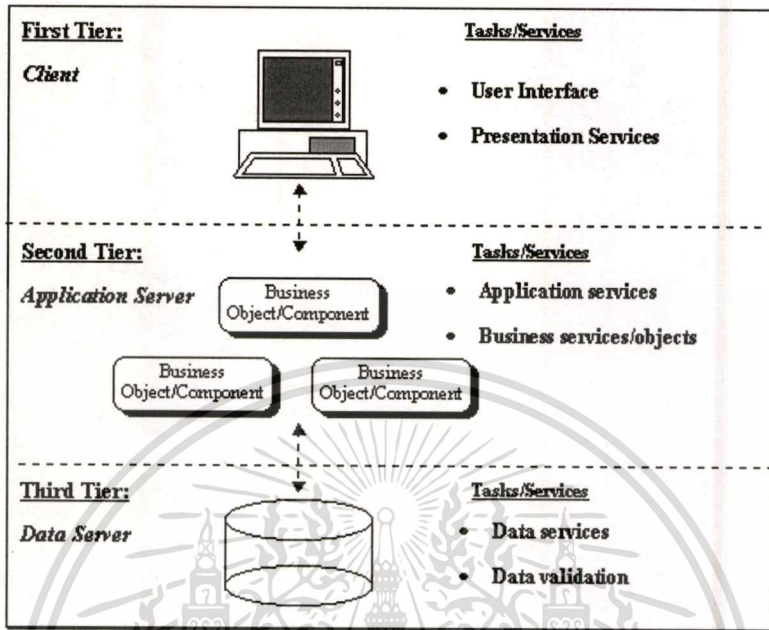
2.4 สถาปัตยกรรม 3 Tiers

ในที่นี้ 3 Tiers ไม่ได้หมายความว่าต้องมีเครื่องที่ประมวลผล Application 3 เครื่อง แต่หมายถึงการแยกส่วนประกอบของ Application ออกเป็น 3 ส่วน ดังรูปที่ 2.31 การทำงานของสถาปัตยกรรม 3 Tier มีความยืดหยุ่นกว่าแบบ 1 Tier และ 2 Tier โดย 3 Tier จะกระจายส่วนของการทำงานของ Application เป็น 3 ส่วนคือ Presentation Logic, Business Logic และ Data Access Logic แต่ละ Application Component สามารถแบ่งกันใช้งานระหว่าง Application ที่สอดคล้องกันได้จากรูป Application Server ซึ่งอยู่ใน Tier ที่ 2 เป็นส่วนที่เป็น Application Logic สามารถปรับใช้กับเครื่องมือต่างๆ ได้โดยอิสระ ไม่จำเป็นต้องพัฒนาบนเครื่องที่มี platform เดียวกับฐานข้อมูลหรือส่วน presentation logic

2.4.1 ข้อดีของสถาปัตยกรรม 3 Tiers

1. Object Reuse เนื่องจาก 3 Tier แบ่ง Application เป็น Module ที่ชัดเจนในการทำงานของ Application ที่สามารถทำงานร่วมกันได้ สามารถนำ Object ที่มีอยู่แล้วมาใช้ร่วมกันได้
2. สามารถดูแลรักษาระบบได้ง่าย
3. ใช้เครือข่ายอย่างมีประสิทธิภาพ ไม่เปลือง Traffic มากเกินไป

4. มีความน่าเชื่อถือของระบบสูง



รูปที่ 2.31 สถาปัตยกรรม 3 Tiers

โครงการพัฒนาระบบเว็บไคเร็กทอรีส่วนตัวนี้ ในส่วนของ Tier ที่ 1 คือส่วนที่ทำหน้าที่เป็น Client ได้แก่โปรแกรม Browser ที่อยู่ที่ฝั่ง Client เช่น Internet Explorer และ Netscape เป็นต้น มีหน้าที่ในการแสดงผลแอปพลิเคชันที่เป็นภาษา HTML และส่วนที่เป็น Client Script ต่างๆ เช่น VB Script และ Java Script เป็นต้น ส่วน Tier ที่ 2 คือโปรแกรมที่ทำหน้าที่เป็น Application Server ในที่นี้คือ Internet Information Server 4.0 ทำงานบนระบบปฏิบัติการ WindowsNT 4.0 มีหน้าที่ประมวลผล ASP ที่เป็น Server Side Script และส่งผลจากการทำงานกลับไปแสดงผลที่โปรแกรม Browser ที่ Tier ที่ 1 และ Tier ที่ 3 คือ

2.5 ASP

ASP ย่อมาจาก Active Server Pages เจ้าของหรือผู้คิดค้นคือบริษัท Microsoft (<http://www.microsoft.com>) ASP เป็นโปรแกรมคอมพิวเตอร์ชนิดหนึ่งที่เขาของได้ให้นิยามไว้คือ "Server side scripting" ซึ่งหมายถึงภาษาการโปรแกรมที่ทำงานในฝั่งของ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Web server หรือ Application Server ที่ให้บริการเอกสารหรือสื่อต่างๆ ในอินเทอร์เน็ตหรืออินทราเน็ต ASP สามารถใช้งานร่วมกับ ADO (ActiveX Data Object) ซึ่งเป็นเทคโนโลยีที่ใช้

ในการเข้าถึงฐานข้อมูล และข้อมูลจากแหล่งเก็บอื่นๆ ได้อย่างมีประสิทธิภาพและง่ายต่อการใช้งานการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

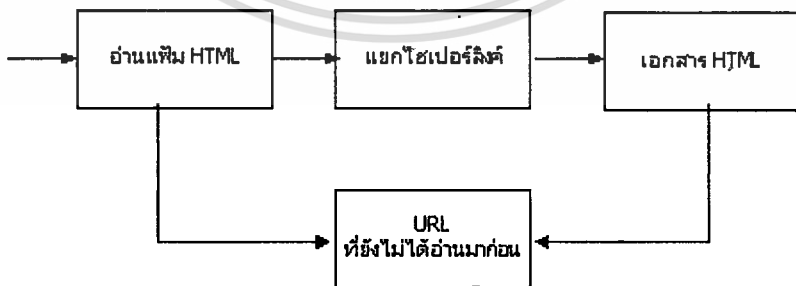
ศึกษาระบบ Search Engine และ Web Directory

3.1 การศึกษาโครงสร้างของเสิร์จ เอ็นจิน

ประกอบด้วย 3 ส่วนใหญ่ๆคือ โปรแกรมโรบอต (Robot), อินเด็กเซอร์ (Indexer) และ โปรแกรมเสิร์จเอ็นจิน (Search Engine) ซึ่งจะอธิบายแยกแต่ละส่วนดังนี้

3.1.1 โปรแกรมโรบอต (Robot)

โปรแกรมโรบอตถือว่าเป็นองค์ประกอบที่สำคัญมากของเสิร์จเอ็นจิน เพราะเป็นตัวแทนของเสิร์จเอ็นจินที่คอยเข้าเยี่ยมชมเว็บเพจต่างๆที่อยู่ในอินเทอร์เน็ตคอยเก็บข้อมูลของลิงค์ที่เราต้องการกลับมาเก็บในฐานข้อมูลของเสิร์จเอ็นจิน การท่องไปในอินเทอร์เน็ตของโปรแกรมโรบอตนี้จะอาศัยเว็บเพจตั้งต้น โดยในเว็บเพจนั้นจะมีลิงค์เชื่อมโยงไปยังเว็บเพจอื่นๆ ซึ่งแต่ละเว็บเพจที่เชื่อมต่อกับเว็บเพจเริ่มต้น ก็มักมีลิงค์ไปยังเว็บเพจอื่นๆอีกเช่นกัน โปรแกรมโรบอตจะค่อยๆเข้าไปยังเว็บเพจเหล่านั้นและกระจายการอ่านข้อมูลออกไปเรื่อยๆ ลึกขึ้นๆ และเราต้องกำหนดด้วยว่าจะให้โปรแกรมโรบอตสำรวจลงไปลึกกี่ชั้น เพราะเป็นไปได้ที่จะให้โปรแกรมโรบอตสำรวจโดยไม่มีขอบเขตที่จำกัดภายในเครือข่ายอินเทอร์เน็ต



รูปที่ 3.1 แสดง การทำงานของโปรแกรมโรบอต

การเดินทางไปยังเว็บต่างๆของโปรแกรมโรบอตไม่ได้เข้าไปเฉยๆ แต่จะคลุดเนื้อหากจากเว็บเพจนั้นกลับมาเพื่อให้เสิร์จเอนจินคัดสรรแล้วเก็บข้อมูลลงในฐานข้อมูลของเสิร์จเอนจิน เราใช้ความสามารถของโปรโตคอล HTTP (Hyper Text Transfer Protocol) ในการเข้าไปสำรวจเว็บเพจต่างๆ แล้วดึงเอาเว็บเพจที่เป็นเอกสารแทรกด้วยแท็ก HTML โดยจะต้องมีวิธีการตัดแท็กของ HTML ออกคัดลอกเฉพาะส่วนของเนื้อหาของเว็บเพจจริงๆ ซึ่งต้องอาศัยโปรแกรมในการตัดสตริง ฟังก์ชันที่เกี่ยวกับสตริงจะทำหน้าที่หลักดังต่อไปนี้ (ดูได้จากรูปที่ 3.1)

1. ฟังก์ชันที่ทำหน้าที่คัดแยกส่วนที่เป็นลิงค์ ออกมาจากแต่ละเว็บเพจ (โดยนำมาเก็บในหน่วยความจำส่วนหนึ่ง) เพื่อเป็นจุดมุ่งหมายต่อไปในการสำรวจของโปรแกรมโรบอต ในการแยกลิงค์เราใช้การหาสตริงแท็ก <A HREF> กับ ก่อนเพื่อเป็นจุดเริ่มต้นและสิ้นสุด
2. จากนั้นมาเก็บข้อมูลสิ่งที่อยู่ตรงกลางซึ่งก็คือ URL ของเว็บเพจหรือลิงค์ที่เราต้องการ
3. ฟังก์ชันที่ทำหน้าที่กำจัดแท็กของ HTML ออกไปเพื่อให้เหลือเฉพาะเนื้อหาล้วนๆ ในการกำจัดแท็กเราจะหาตัวอักษร '<' และ '>' ของ Tag ต่างๆ และตัดส่วนตรงกลางออกไปพร้อมกับตัวอักษร '<' และ '>' ก็จะเหลือเฉพาะส่วนที่เป็นเนื้อหา
4. ฟังก์ชันที่ทำหน้าที่นับคีย์เวิร์ดในเนื้อหาของเว็บเพจนั้นๆ ว่ามีอยู่มากน้อยเพียงไรถ้าพบว่ามีคีย์เวิร์ดมากก็มีแนวโน้มว่าเว็บเพจนั้นมีเนื้อหาตรงกับที่ผู้ใช้ต้องการค้นหา

ส่วนโรบอตนี้บางครั้งเรียกว่า 'Crawler' หรือ 'Bot' ก็ได้ และเนื่องจากการทำงานของ Spider นั้นใช้ทรัพยากรของระบบเป็นจำนวนมาก ทั้งเน็ตเวิร์ก และ Host ที่ทำงาน เพราะฉะนั้น Spider ที่ดีควรมีลักษณะการทำงานบางอย่างที่จะลดการใช้ทรัพยากรของระบบลง เช่น

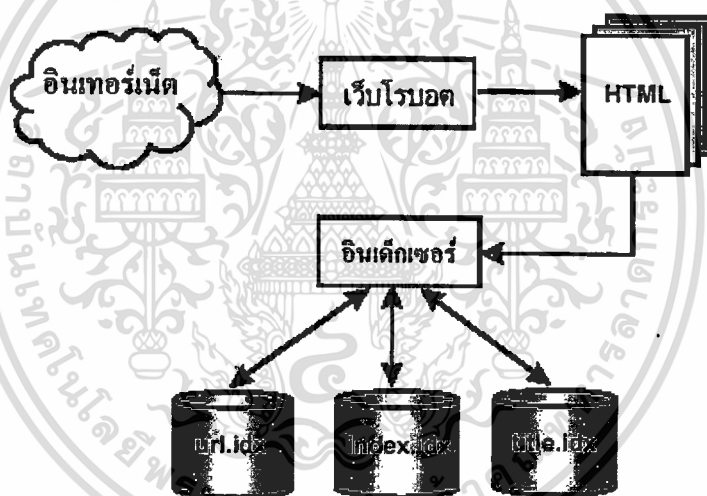
- ไม่อ่านเอกสาร HTML มากเกินไป แม้ว่า spider จะมีความสามารถจัดการเอกสารได้ทันที เพราะว่าอาจจะทำให้ Network ของทั้งระบบช้าลง
- อ่านเฉพาะส่วนที่จำเป็นต้องใช้ เช่น อาจจะอ่านมาเฉพาะส่วนที่เป็นตัวอักษร ซึ่งในโปรโตคอล HTTP มีฟิลด์ ที่ชื่อว่า Accept ซึ่งใช้สำหรับบอกชนิดของข้อมูลที่เราต้องการ หากที่การระบุชนิดของข้อมูลลงในฟิลด์นี้เว็บเซิร์ฟเวอร์ (web server) จะส่งข้อมูลมาเฉพาะชนิดที่ถูกระบุในไฟล์
- ตรวจสอบไม่ให้อ่าน เว็บที่เคยอ่านไป แล้วข้อควรระวังของข้อนี้ก็คือ เซิร์ฟเวอร์ บางเซิร์ฟเวอร์อาจจะมีชื่อได้หลายชื่อ เช่น web.nexor.co.uk , nercules.nexor.co.uk และ 128.243.219.1 เป็นเซิร์ฟเวอร์ เดียวกัน

เราสามารถสรุปหน้าที่ของโรบอตได้ดังนี้

- Indexing คือลิงก์และทำอินเด็กซ์เก็บลงฐานข้อมูล
- HTML Validation ตรวจสอบว่าเป็นเอกสาร HTML ที่สามารถแปลความหมายได้หรือไม่
- Link Validation ตรวจสอบลิงก์ว่ายังสามารถไปได้หรือไม่ และอัปเดตฐานข้อมูล
- "What's New" monitoring ตรวจสอบการเปลี่ยนแปลงข้อมูลของโฮมเพจ และอัปเดตเฉพาะส่วนที่เปลี่ยนแปลง

3.1.2 อินเด็กเซอร์ (Indexer)

ทำหน้าที่เป็นตัวสร้างดัชนีผลลัพธ์ จากอินเด็กเซอร์ประกอบด้วยฐานข้อมูล 3 ฐานข้อมูลดังรูปที่ 3.2 นี้

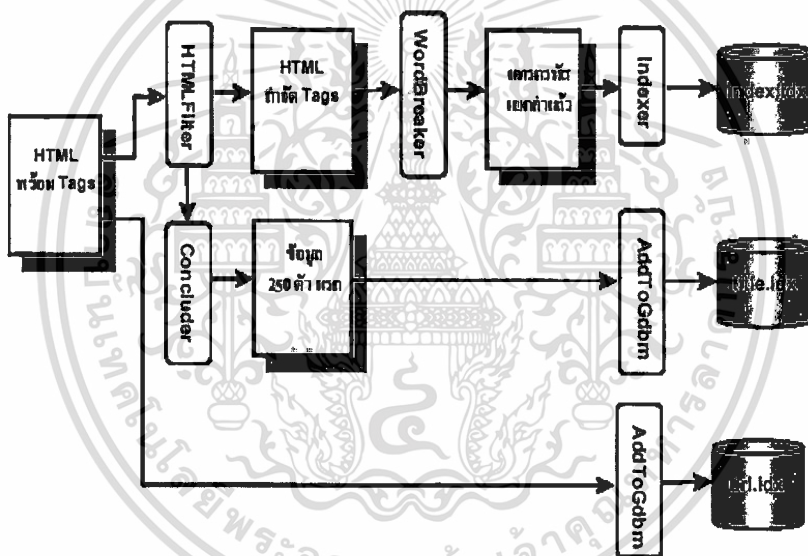


รูปที่ 3.2 การทำงานของอินเด็กเซอร์

- ฐานข้อมูล index เป็นฐานข้อมูลที่บรรจุคำศัพท์ทั้งหมด และหมายเลขกำกับ URL เป็นตัวชี้ไปยังเว็บเพจที่มีคำศัพท์นั้นๆ อยู่
- ฐานข้อมูล url เป็นฐานข้อมูลที่เก็บหมายเลขกำกับ URL และชื่อ URL นั้น
- ฐานข้อมูล Title เป็นฐานข้อมูลที่เก็บหมายเลขกำกับ URL และเก็บอักษรหัวเรื่องของเว็บเพจจากแท็ก <TITLE> และอักษรระ 250-300 ตัวแรก เพื่อเป็นข้อมูลสำหรับการค้นหา

โครงสร้างการทำงานของอินเด็กเซอร์มีองค์ประกอบหลักดังรูปที่ 3.3 อธิบายได้ดังนี้

- HtmlFilter ทำหน้าที่กรอง HTML Tag ออกและหากพบรหัสแฮชแท็กที่ใช้แทนอักขระ จะดำเนินการแปลงไปเป็นรหัสแอสกี
- WordBreaker ทำหน้าที่แบ่งคำ โดยตัดคำที่ยาวที่สุดที่เป็นไปได้ โดยอาศัยดัชนีการช่วย
- Indexer ทำหน้าที่นำคำศัพท์แต่ละคำไปเก็บในฐานข้อมูล index
- Concluder ทำหน้าที่หาคำบรรยายเว็บเพจโดยนำข้อความที่กำกับด้วย <Title> ... </Title> มาเป็นหัวข้อ และนำข้อมูล 250-300 ตัวอักษรถัดไปมาเป็นคำบรรยาย
- AddtoGdbm ทำหน้าที่เพิ่มข้อมูลในฐานข้อมูล title และฐานข้อมูล url



รูปที่ 3.3 รายละเอียดการทำงานของอินเด็กเซอร์

3.1.3 เสิร์จเอนจิน (Search Engine)

เป็นส่วนที่รับคำศัพท์ที่ต้องการค้นหาจากผู้ใช้งานผ่านทาง CGI (Common Gateway Interface) มาค้นในฐานข้อมูลและส่งผลการค้นหาคลับไปแสดงที่หน้าจอ โดยนำคำศัพท์มาเปิดหาในฐานข้อมูล Index และนำผลลัพธ์ที่ได้จากการค้นหาของแต่ละครั้งมาหาเอกสารที่มีคำศัพท์ครบทุกคำ แล้วค่อยเปิดฐานข้อมูล url และฐานข้อมูล title เพื่อแสดงชื่อ URL และคำบรรยายออกมาทางหน้าจอ

3.2 โครงสร้างของเว็บไคเร็กทอรี

โดยหลักพื้นฐานการสร้างสรรค์ระบบเว็บเพจไคเร็กทอรีมีการทำงานที่แตกต่างจากเสิร์จเอนจิน คือการใช้การปรับเปลี่ยนข้อมูลโดยผู้ดูแลระบบเองโดยไม่ได้ทำอย่างอัตโนมัติ เว็บไซต์ไคที่ต้องการมีรายชื่อในไคเร็กทอรีต้องติดต่อไปยังผู้ดูแลไคเร็กทอรี เพื่อให้ผู้ดูแลจำแนกและจัดเก็บลงฐานข้อมูลด้วยตัวเอง ระบบการเก็บแบบไคเร็กทอรีอาจจะมีการทำงานที่ไม่อัตโนมัติเท่าที่ควร แต่การค้นหาข้อมูลจากเว็บที่ทำงานลักษณะเว็บไคเร็กทอรีอาจให้ผลลัพธ์ของการค้นหาข้อมูลที่ตรงประเด็นมากกว่าเสิร์จเอนจินเพราะผ่านการตรวจแยกหมวดหมู่ของข้อมูลจากผู้ดูแลไคเร็กทอรีมาแล้ว

ดังนั้นเว็บไคเร็กทอรีประกอบด้วยส่วนที่เป็นแกนหลักคือ โปรแกรมเสิร์จเอนจิน (Search Engine) ส่วนโปรแกรมโรบอต (Robot) และอินเด็กเซอร์ (Indexer) จะมีผู้ที่ทำหน้าที่เป็นผู้จัดการไคเร็กทอรีแทน ดังนั้นโปรแกรมเสิร์จเอนจิน ที่มีความสามารถเก็บและดึงข้อมูล Personal Database ของผู้ใช้แต่ละคนจึงเป็นหัวใจหลักในการพัฒนาโครงการครั้งนี้



บทที่ 4

การออกแบบระบบ

4.1 การวิเคราะห์ออกแบบระบบ

ในการออกแบบระบบเว็บไคเร็กทอรีครั้งนี้เพื่อสร้างระบบเว็บไคเร็กทอรีที่แสดงผลของการค้นหาที่เรียงสอดคล้องกับข้อมูลส่วนตัว

4.1.1 กำหนดวัตถุประสงค์

- เก็บข้อมูล URL, คำอธิบาย และข้อมูลที่เกี่ยวข้องได้
- เก็บข้อมูลและรายละเอียดของผู้ใช้งานได้ (Personal Information)
- มีระบบ Counter ตรวจสอบจำนวนการใช้งานเว็บไคเร็กทอรี
- ผู้ใช้สามารถ Login หรือ ไม่ต้อง Login ก็ได้ในการใช้งานเว็บไคเร็กทอรี
- ถ้าผู้ใช้ Login เว็บไคเร็กทอรีจะแสดงผลลิ้งค์ขอการ Search สอดคล้องกับข้อมูลที่เก็บใน Personal Information
- ผู้ใช้สามารถแจ้ง URL ใหม่เข้ามายังเว็บไคเร็กทอรีได้
- ผู้ใช้สามารถติดต่อกับผู้ดูแลไคเร็กทอรีผ่านเว็บได้
- ผู้ดูแลไคเร็กทอรีสามารถบริหารฐานข้อมูลไคเร็กทอรี และฐานข้อมูลผู้ใช้ผ่านเว็บได้

4.1.2 Hardware and Software Specification

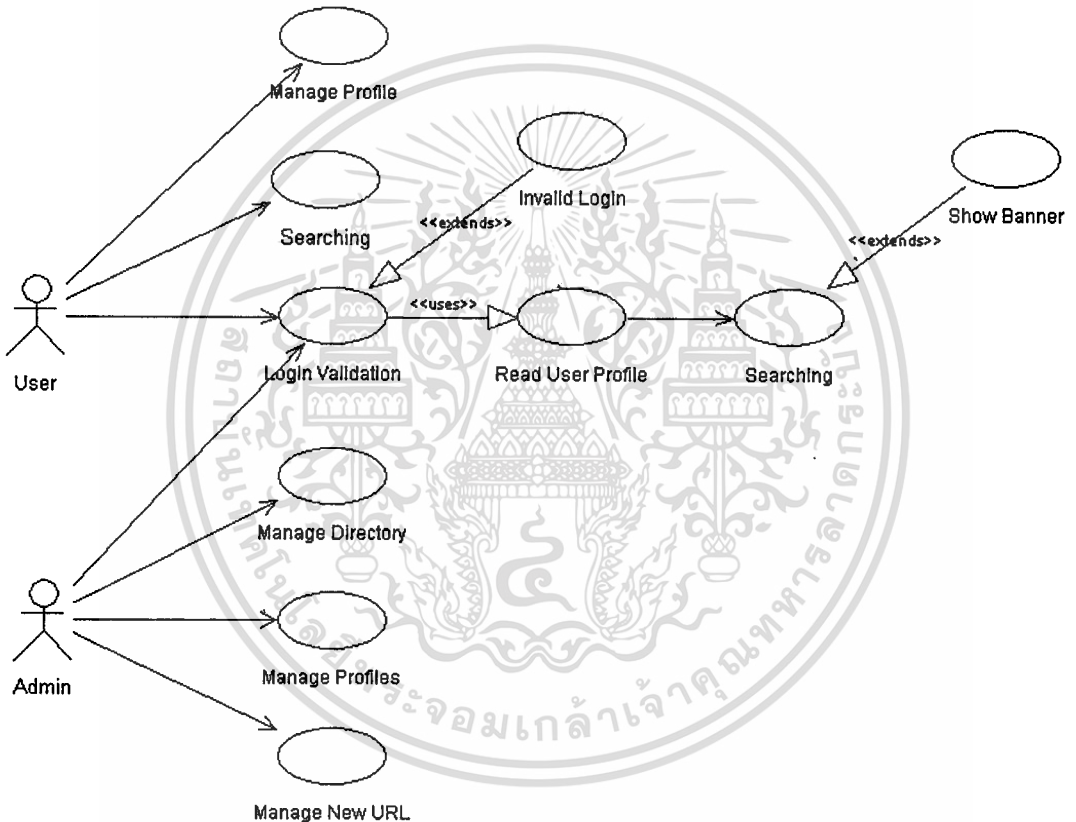
4.1.2.1 เครื่องคอมพิวเตอร์ที่ใช้ในโครงการ : ใช้เครื่องคอมพิวเตอร์ IBM Compatible ที่ใช้ CPU Intel Pentium II 400 Mhz เป็นหน่วยประมวลผลการ มีหน่วยความจำ RAM 128 MB ฮาร์ดดิสก์ 22 GB

4.1.2.2 ซอฟต์แวร์ที่ใช้ในโครงการ : ใช้ระบบปฏิบัติการ WindowsNT เวอร์ชัน 4.0 โดยติดตั้ง Service Pack 4 ,Internet Explorer 5.0 และ WindowsNT Option Packs (IIS4) ระบบจัดการฐานข้อมูล Oracle 8.0.5 for Windows NT

ผลของการออกแบบระบบเว็บไคลเอนต์ส่วนตัว จะแสดงเป็นไดอะแกรมของ UML ดังแสดงในหัวข้อที่ 4.1.3 ถึง 4.1.7

4.1.3 ยูสเคสไดอะแกรม (Use Case Diagram)

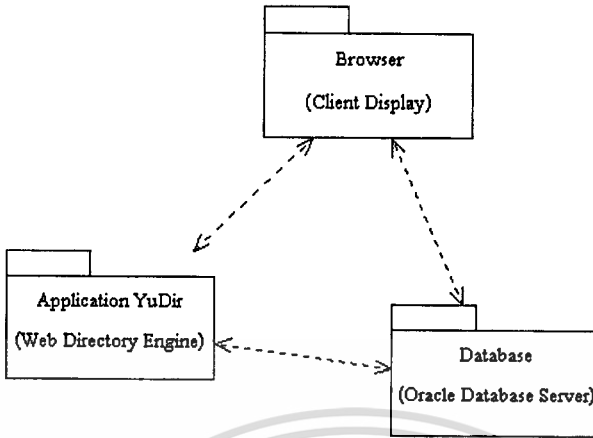
เมื่อผ่าน Phase ของการให้คำนิยามต่างๆ ของซอฟต์แวร์ (Software Definition) แล้วจึงได้ผลลัพธ์การจำลองระบบเว็บไคลเอนต์ส่วนตัว ดังรูปที่ 4.1



รูปที่ 4.1 รายละเอียดยูสเคสไดอะแกรม (Use Case Diagram)

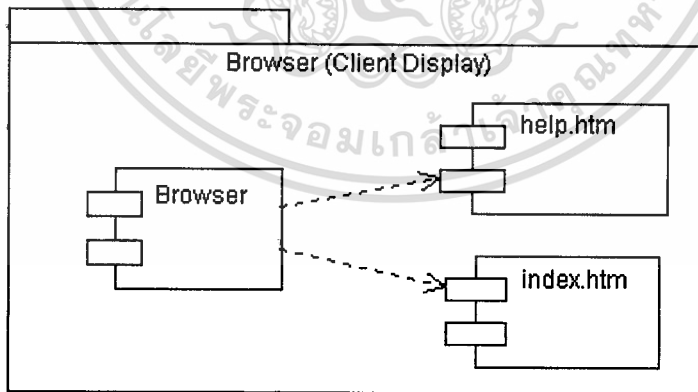
4.1.4 คอมโพเนนต์ไดอะแกรม (Component Diagram)

เมื่อผ่าน Phase ของการวิเคราะห์ (Analysis) ทำให้สามารถออกแบบ Component Diagram โดย Packages ภายใน มีด้วยกัน 3 Packages ได้แก่ Interface, Digital Library และ Database ดังรูปที่ 4.2 ถึงรูปที่ 4.7

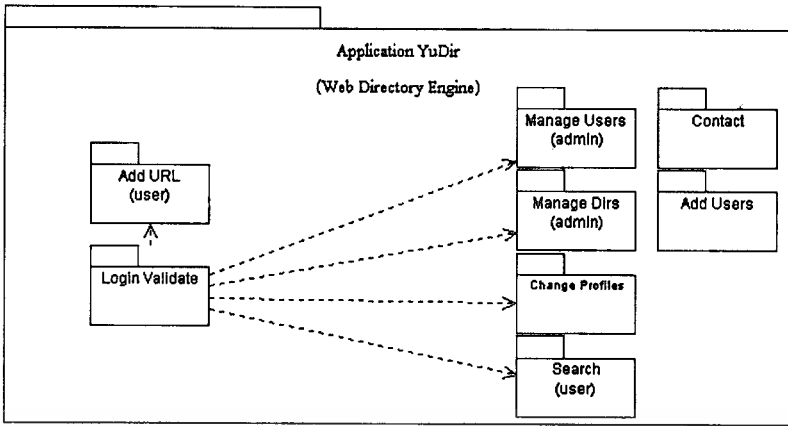


รูปที่ 4.2 รายละเอียดคอมโพเนนต์ไคอะแกรมหลัก (Main Component Diagram)

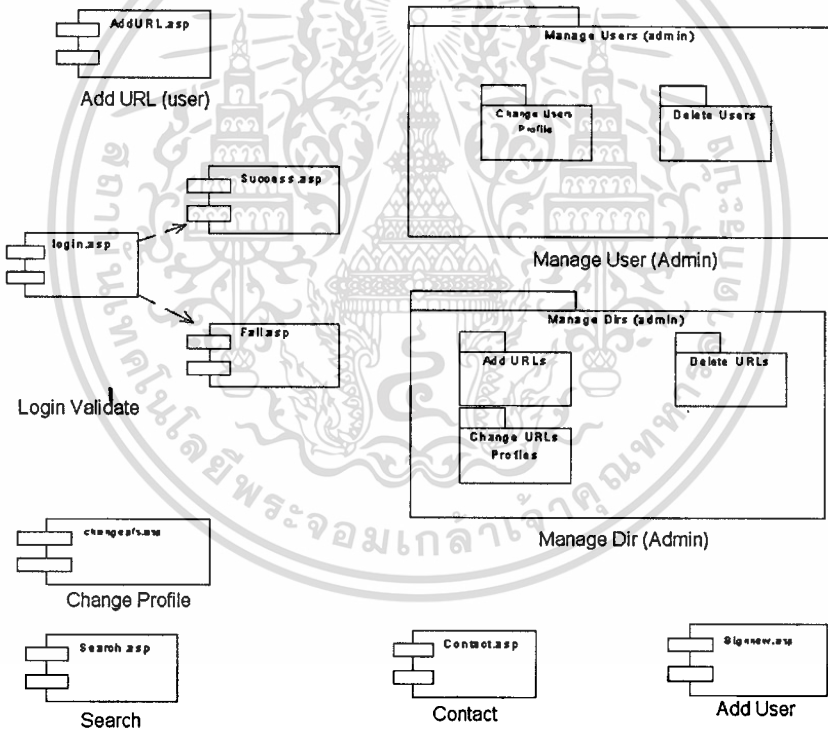
ซึ่งจากคอมโพเนนต์ไคอะแกรม ดังรูปที่ 4.2 สามารถอธิบายย่อยออกเป็นองค์ประกอบของ 3 Component หลัก ดังรูปที่ 4.3 เป็นคอมโพเนนต์ในส่วนของ Client Display, รูปที่ 4.4 ถึงรูปที่ 4.6 เป็นคอมโพเนนต์ในส่วนของ Application YuDir และ รูปที่ 4.7 เป็นคอมโพเนนต์ในส่วนของ Database ตามรายละเอียดต่อไปนี้



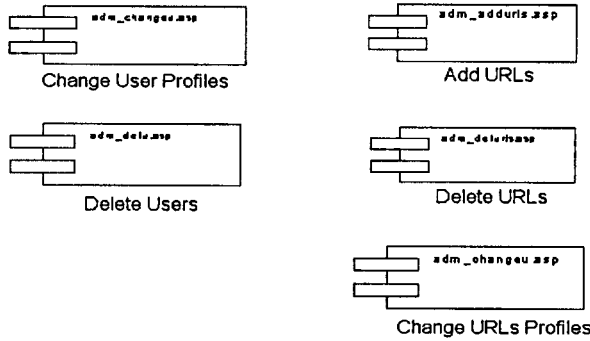
รูปที่ 4.3 รายละเอียดคอมโพเนนต์ไคอะแกรม (Component Diagram) ใน Client Display



รูปที่ 4.4 คอมโพเนนต์โคโอะแกรม (Component Diagram) ใน Application Yu Dir (1/3)



รูปที่ 4.5 คอมโพเนนต์โคโอะแกรม (Component Diagram) ใน Application Yu Dir (2/3)



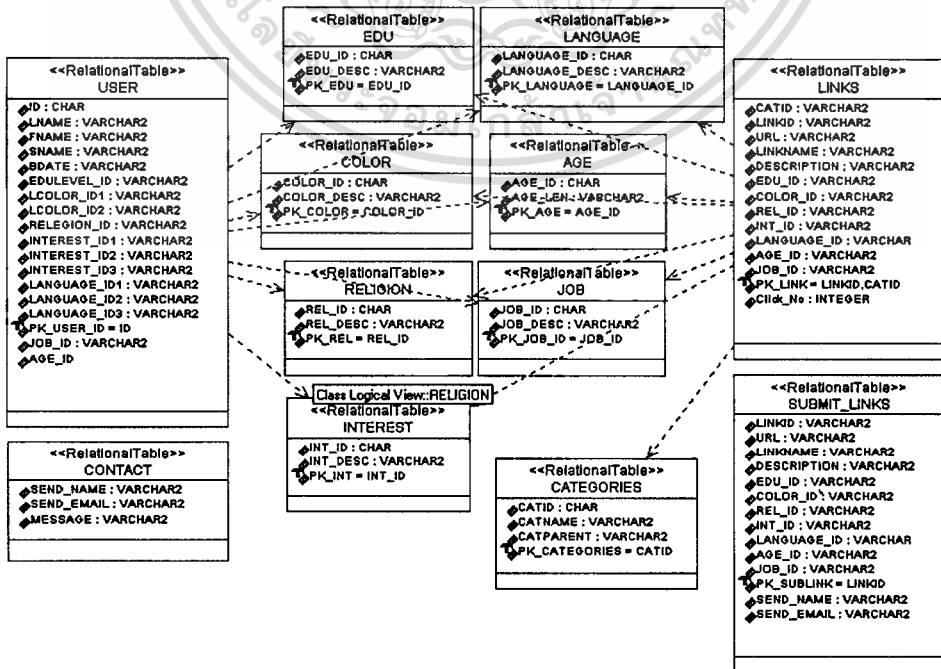
รูปที่ 4.6 คอมโพเนนต์โคโอะแกรม (Component Diagram) ใน Application Yu Dir (3/3)

หมายเหตุ : รูปที่ 4.5 เป็น Component ที่แสดงรายละเอียดภายใน Package ของ รูปที่ 4.4 Application YuDir (1/3) และรูปที่ 4.6 เป็น Component ที่แสดงรายละเอียดภายใน Package ชื่อ Manage User และ Manage Dir ตามลำดับ ต่อจาก รูปที่ 4.5 Application YuDir (2/3)



รูปที่ 4.7 รายละเอียดคอมโพเนนต์โคโอะแกรม (Component Diagram) ใน Database

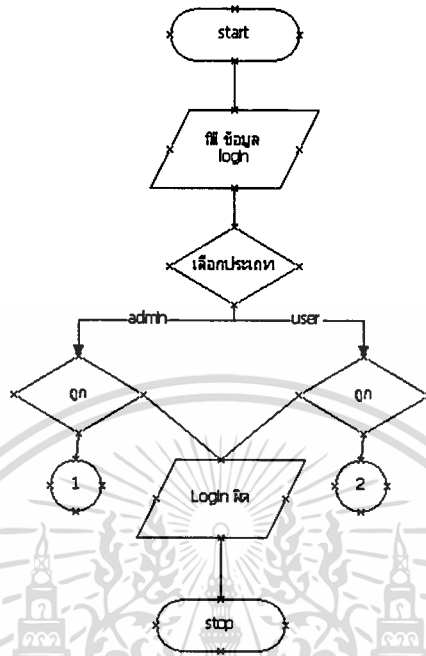
4.1.5 คลาสโคโอะแกรม (Class Diagram)



รูปที่ 4.8 รายละเอียดClass Diagram ใน Database Engine

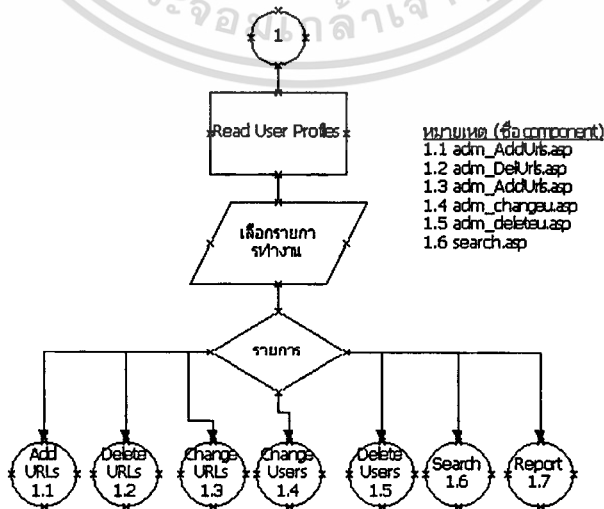
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 แอ็กทิวิตีไดอะแกรม (Activity Diagram)

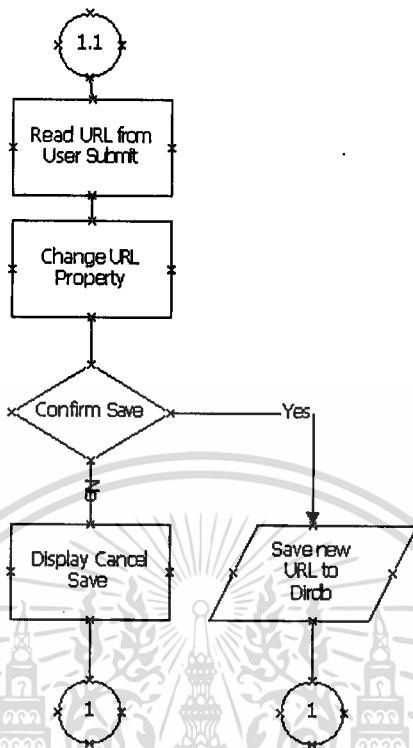


รูปที่ 4.9 Activity Diagram ส่วนของการ Login

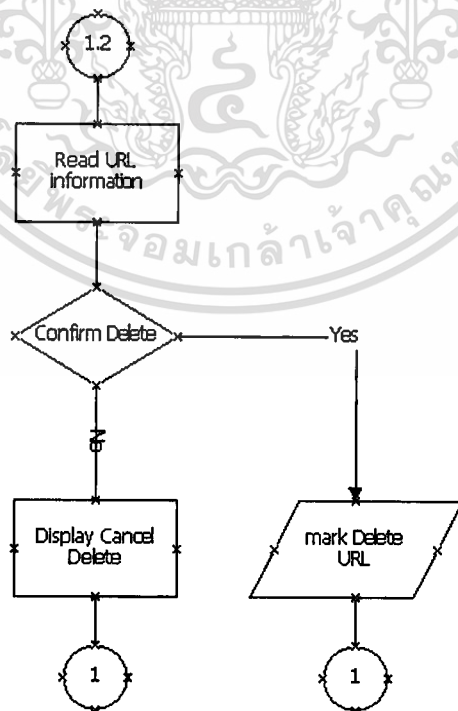
การ Login ดัง Activity Diagram ดังรูปที่ 4.9 สามารถแบ่งการทำงานออกเป็น 2 หมวดผู้ใช้งาน คือ Admin จะแสดง Activity Diagram ดังรูปที่ 4.10 ถึงรูปที่ 4.16 และ User จะแสดง Activity Diagram ดังรูปที่ 4.17 ถึงรูปที่ 4.20 ตามลำดับ



รูปที่ 4.10 Activity Diagram ส่วนของการ เป็น Admin (1/7)

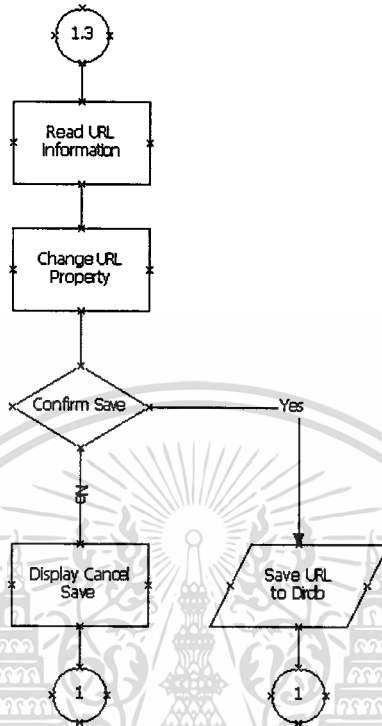


รูปที่ 4.11 Activity Diagram ส่วนของการ เป็น Admin (2/7) : เพิ่ม URLs

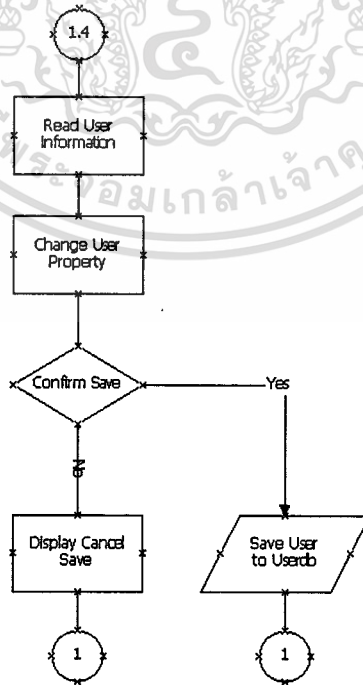


รูปที่ 4.12 Activity Diagram ส่วนของการ เป็น Admin (3/7) : ลบ URLs

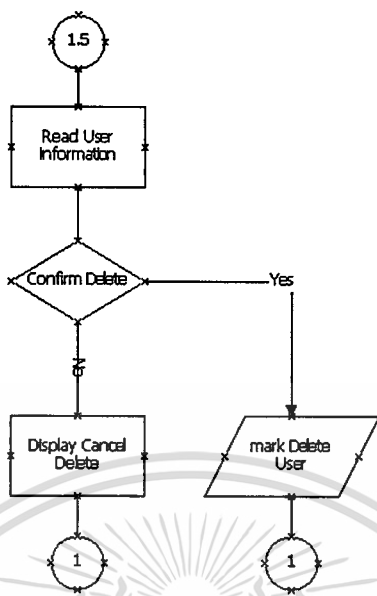
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



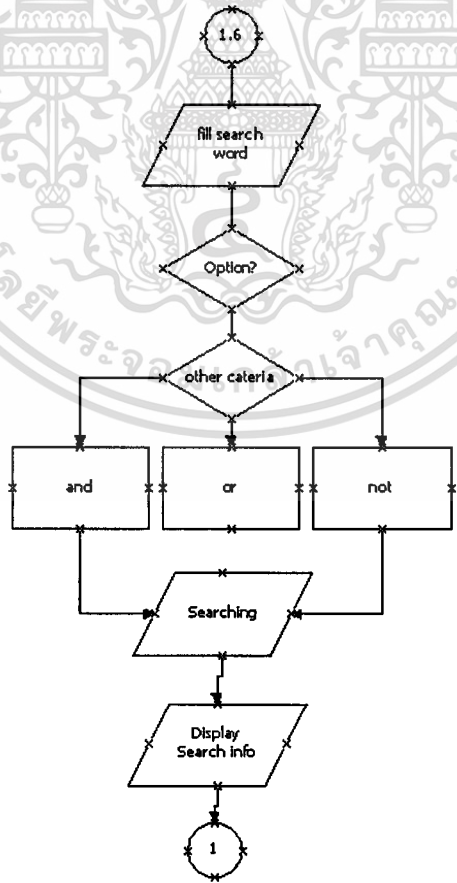
รูปที่ 4.13 Activity Diagram ส่วนของการ เป็น Admin (4/7) : เปลี่ยนคุณสมบัติ URLs



รูปที่ 4.14 Activity Diagram ส่วนของการ เป็น Admin (5/7) : เปลี่ยนคุณสมบัติ USERS

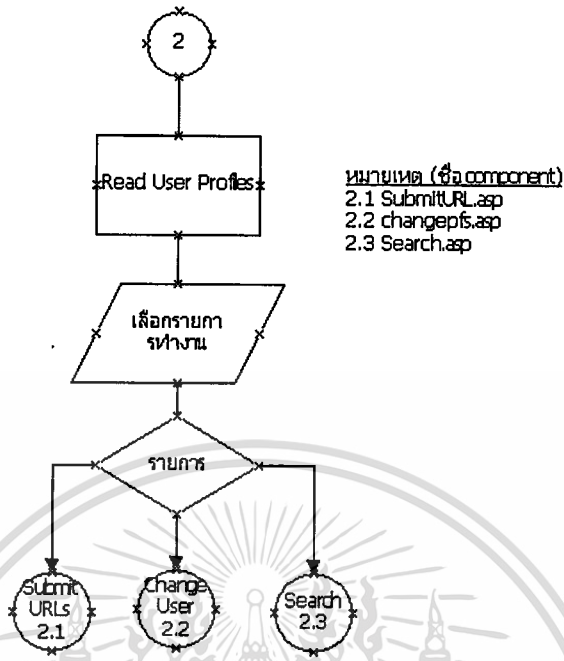


รูปที่ 4.15 Activity Diagram ส่วนของการ เป็น Admin (6/7) : ลบ USERS

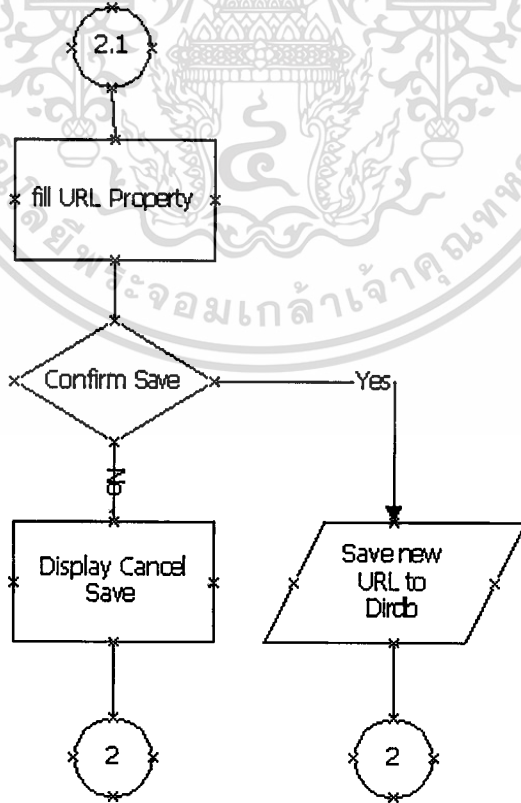


รูปที่ 4.16 Activity Diagram ส่วนของการ เป็น Admin (7/7) : Search

เอกสารนี้เป็นเอกสาร
 นโยบายด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

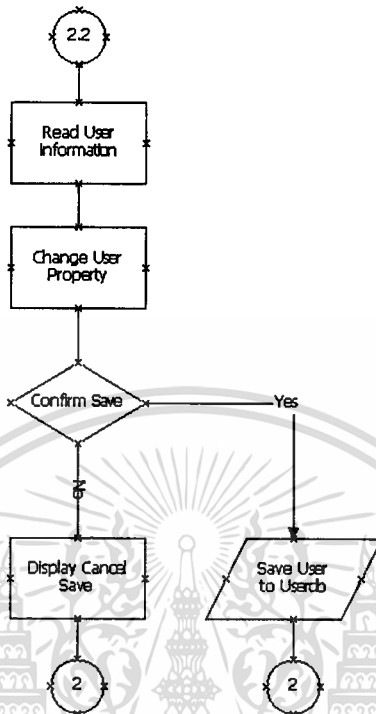


รูปที่ 4.17 Activity Diagram ส่วนของการ เป็น User (1/4)

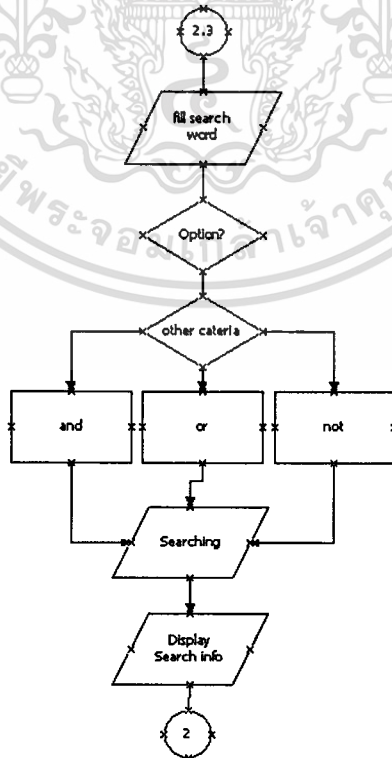


รูปที่ 4.18 Activity Diagram ส่วนของการ เป็น User (2/4) : Submit URLs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

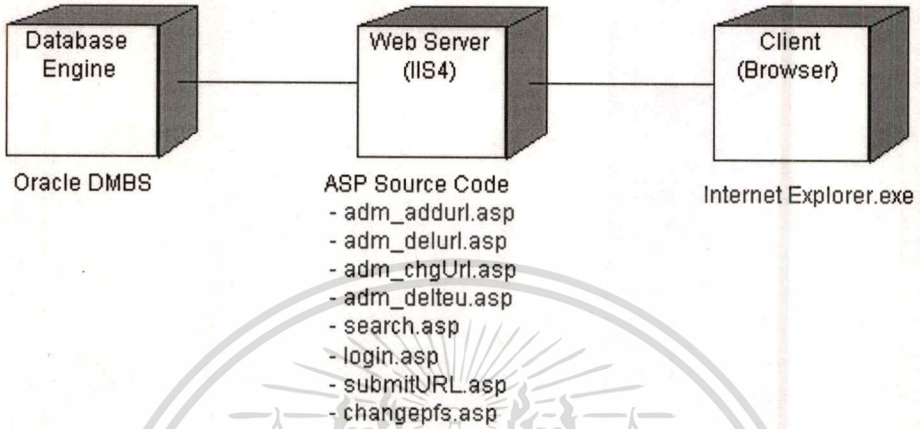


รูปที่ 4.19 Activity Diagram ส่วนของการ เป็น User (3/4) : Change USER Profile

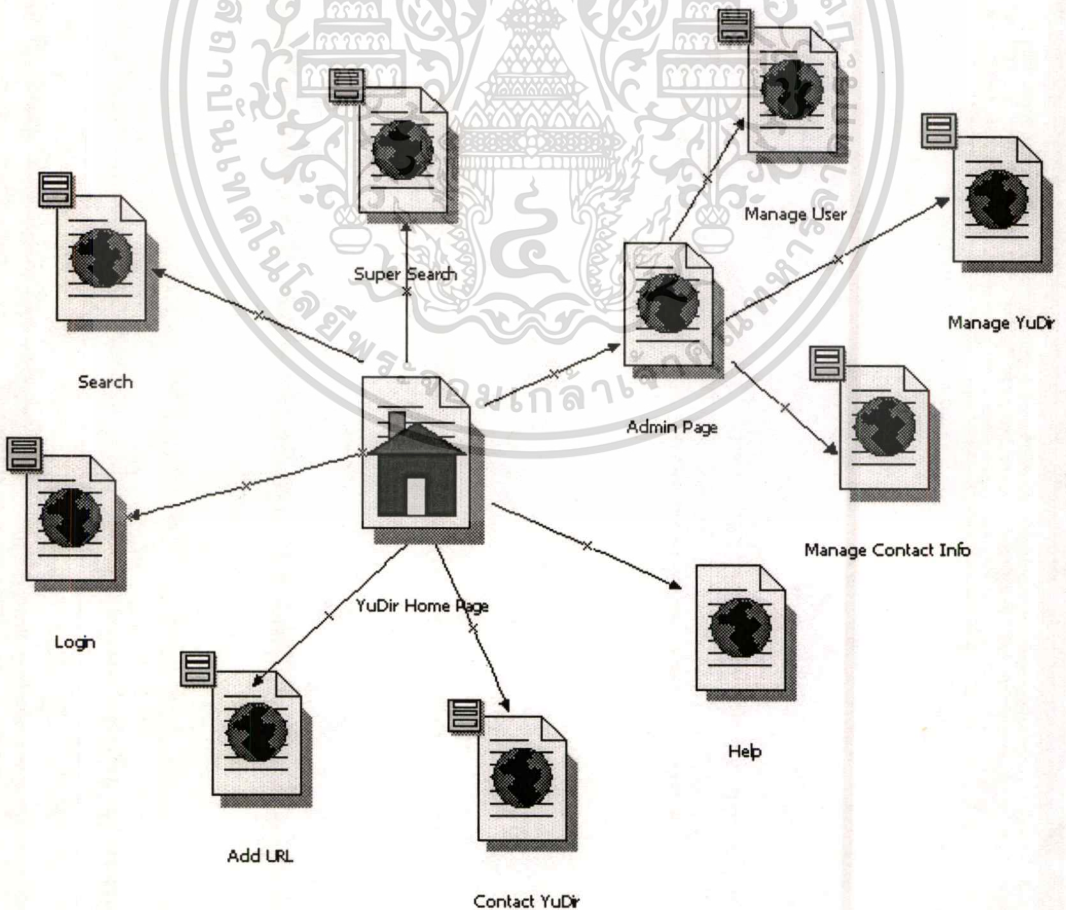


รูปที่ 4.20 Activity Diagram ส่วนของการ เป็น User (4/4) : Search

4.1.7 ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)



รูปที่ 4.21 ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram : 1/2)



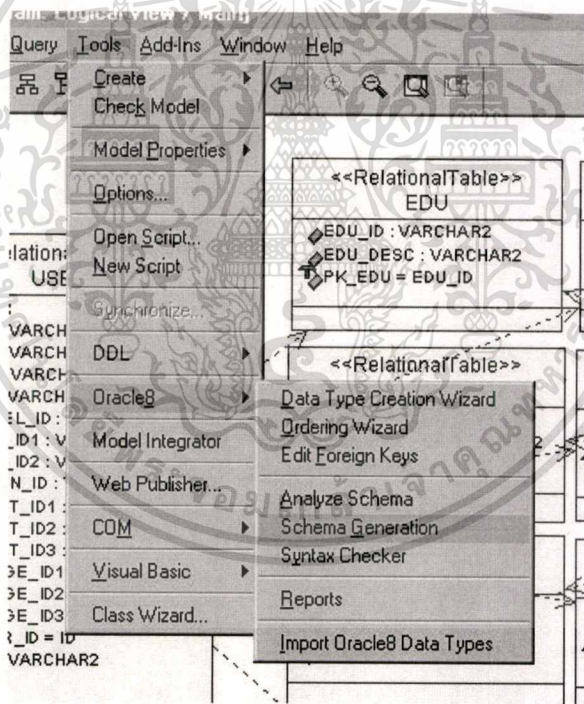
รูปที่ 4.22 รายละเอียดของดีพลอยเมนต์ไดอะแกรม (Deployment Diagram : 2/2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.8 การสร้างตารางในฐานข้อมูลแบบสัมพันธ์ (Relational Table)

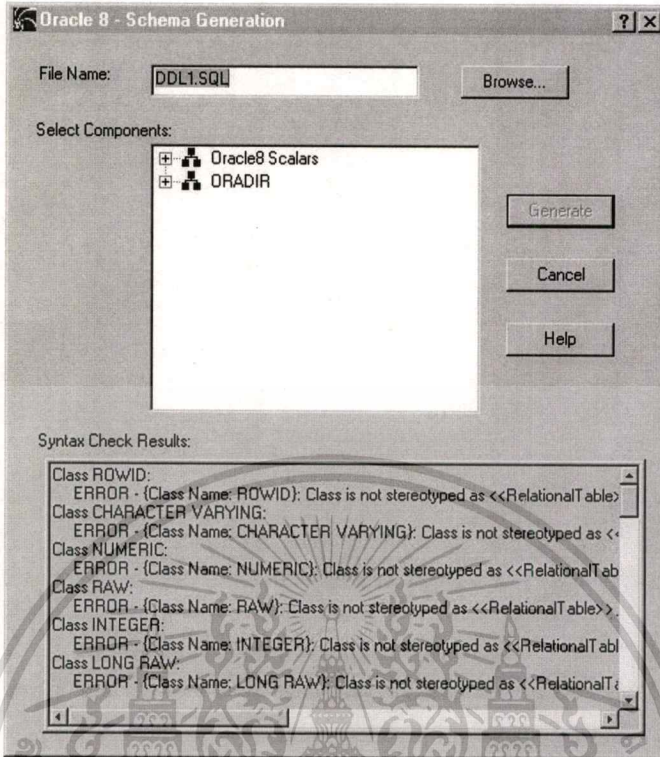
การสร้าง Relational Table นี้ใช้ CASE Tools ชื่อ Rational Rose 2000 Enterprise Edition (<http://www.rational.com>) เป็น Visual Modeling tool และ Object-oriented Case Tool ที่สามารถสร้าง UML diagram ได้ เพื่อใช้ในการจำลองระบบงานได้ ด้วยการวาด Diagram ต่างๆ โดยมีตัวจัดการ Diagram และกำหนดคุณสมบัติต่างๆ ของ Diagram ที่จำเป็น การเชื่อมโยงกันระหว่าง Diagram และการจัดการเอกสารจากแบบจำลองที่ได้สร้างขึ้น รวมทั้งมีตัวช่วยอัตโนมัติในการสร้าง Class Rational Rose Oracle8 ที่สามารถเอา Data Type และ Class ใน Oracle มาใช้ได้ การสร้าง Relational Table นี้เราใช้ รายละเอียด Class Diagram ใน Database Engine จากรูปที่ 4.8 จากนั้นที่โปรแกรม Rational Rose2000

1. เลือก Menu Tools -> Oracle8 -> Schema Generation ดังรูปที่ 4.23

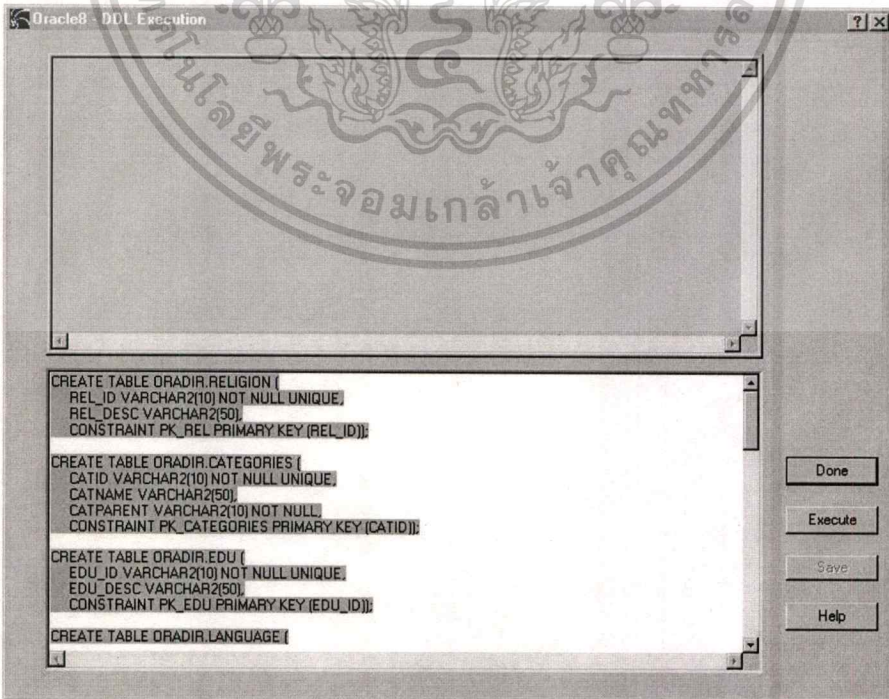


รูปที่ 4.23 การสร้าง Relational Table จาก Database Class Diagram (1/3)

2. จะมีข้อความ Confirm และให้เลือก Instance ที่จะสร้าง DDL ดังรูปที่ 4.24



รูปที่ 4.24 การสร้าง Relational Table จาก Database Class Diagram (2/3)



รูปที่ 4.25 การสร้าง Relational Table จาก Database Class Diagram (3/3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อคุณดูเพิ่มเติมเกี่ยวกับเงื่อนไขการใช้งานการคุ้มครองลิขสิทธิ์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จะได้คำสั่ง DML ดังรูปที่ 4.25 หากต้องการสั่ง Create Database ด้วย Rational Rose ให้กดปุ่ม Execute โดยระบบจะต้องติดตั้ง Engine ของ Oracle สมบูรณ์แล้ว แต่ในกรณีต้องนำไป Execute ด้วยตัวเองต้อง Copy Source Code เก็บไว้ หรือไปเปิดจาก SQL File ที่ Rational Rose Save ไว้ เพื่อไปนำไปใช้ Create Database ในภายหลัง

4.2 พัฒนาโปรแกรมด้วย ASP

หลังจากที่ได้ Relational Table แล้ว จึงทำการพัฒนาระบบเว็บไคลเอนต์ส่วนตัว ร่วมกับ Hardware และ Software ดังที่ได้อธิบายไปแล้วในข้อที่ 4.1.2 โดยนำ DML Script ที่ได้จากขั้นตอนการ Create Relational Table (ขั้นตอนที่ 4.1.8) มา Run ใน Oracle SQL Plus เพื่อสร้าง Table และติดตั้งและ Config IIS Server ให้รู้จักกับฐานข้อมูล Oracle ที่สร้างขึ้น และใช้ Macromedia DreamWeaver 3.0, Microsoft Front Page 2000 และ Microsoft Visual Interdev 6.0 ในการช่วยพัฒนาระบบเว็บไคลเอนต์ส่วนตัวครั้งนี้ และได้ผลลัพธ์สุดท้ายดังที่แสดงในภาคผนวก ข.



บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 ผลการศึกษา

ผลการดำเนินการศึกษาครั้งนี้ได้เป็นระบบเว็บไคลเอนทรีส่วนตัว โดยใช้หลักการออกแบบโดยใช้ Object-Oriented Paradigm โดยใช้เทคนิคของ UML (Unified Modeling Language) ซึ่งเป็นภาษาโมเดลแบบหนึ่ง เป็นพื้นฐานในการวิเคราะห์และออกแบบระบบ ส่วนภาษาที่ใช้ในการพัฒนาระบบเว็บไคลเอนทรีส่วนตัวใช้ภาษา ASP ในการแสดงผลของข้อมูลที่เก็บอยู่ในฐานข้อมูล Oracle เวอร์ชัน 8.0.5 ที่ทำงานบน WindowsNT 4.0 SP5 เป็นสถาปัตยกรรมแบบ 3 Tiers และโปรแกรมทุกส่วนทำงานบนเครื่องคอมพิวเตอร์ Pentium II 400 Mhz การดำเนินการได้เริ่มเก็บรวบรวมข้อมูลจากเว็บไคลเอนทรีที่มีอยู่ในอินเทอร์เน็ต เพื่อเป็นแนวทางในการพัฒนาเว็บไคลเอนทรีส่วนตัว จากนั้นนำข้อมูลมาวิเคราะห์ออกแบบระบบเว็บไคลเอนทรีส่วนตัวใหม่ได้เป็น ยูสเคสไดอะแกรม, คลาสไดอะแกรม, แอ็กทิวิตีไดอะแกรม, คอลลาบอเรชันไดอะแกรม และ ดีพลอยเมนต์ไดอะแกรม จากนั้น Map ข้อมูลที่ได้จากคลาสไดอะแกรมไปเป็นตารางในฐานข้อมูลแบบสัมพันธ์ ดังที่ได้แสดงไว้แล้วในบทที่ 4

โดยผลการพัฒนาโปรแกรมระบบเว็บไคลเอนทรีส่วนตัวสามารถทำงานได้ดังต่อไปนี้

1. ฐานข้อมูลไคลเอนทรีเก็บข้อมูล URL, คำอธิบาย และข้อมูลที่เกี่ยวข้องได้
2. ฐานข้อมูลไคลเอนทรีเก็บข้อมูลและรายละเอียดของผู้ใช้งานได้ (Personal Information)
3. มีระบบ Counter ตรวจสอบจำนวนการใช้งานเว็บไคลเอนทรี
4. มีระบบที่ผู้ใช้สามารถ Login หรือ ไม่ต้อง Login ก็ได้ในการใช้งานเว็บไคลเอนทรี
5. ถ้าผู้ใช้ Login เว็บไคลเอนทรีจะแสดงผลลัพธ์ของการ Search สอดคล้องกับข้อมูลที่เก็บใน Personal Information ของผู้ใช้นั้น
6. ฐานข้อมูลไคลเอนทรีสามารถรับ URL ใหม่ที่ผู้ใช้แจ้งเข้ามาให้ผู้ดูแลเว็บไคลเอนทรีเพิ่มข้อมูลใหม่ๆเข้าไปในไคลเอนทรีได้
7. ฐานข้อมูลไคลเอนทรีมีส่วนที่ให้ผู้ใช้งานส่งข้อความติดต่อกับผู้ดูแลไคลเอนทรีผ่านเว็บ
8. ผู้ดูแลไคลเอนทรีสามารถบริหารฐานข้อมูลไคลเอนทรี และฐานข้อมูลผู้ใช้ผ่านเว็บได้

ซึ่งระบบเว็บไคลเร็กทอรีที่ได้พัฒนาขึ้นสามารถทำงานได้ครอบคลุมตามวัตถุประสงค์เป็นอย่างดี รายละเอียดการใช้งานโปรแกรมระบบเว็บไคลเร็กทอรีสามารถดูได้จากภาคผนวก ก.

5.2 สรุป

จากผลการศึกษาพบว่า การนำเอาวิธีการของ Object-Oriented มาใช้ในการวิเคราะห์และออกแบบนั้น เป็นแนวคิดที่สามารถทำความเข้าใจได้ง่าย เพราะการวิเคราะห์และออกแบบทุกขั้นตอนจะพิจารณา จากสิ่งที่อยู่ในระบบเหมือนกับเป็นนาม หรือ Object ตัวหนึ่ง ขั้นตอนต่อไปในการวิเคราะห์ก็จะนำสิ่งที่ เป็น Object เหล่านั้นมาวิเคราะห์ว่าประกอบด้วยคุณสมบัติ (Property) อะไรบ้าง และสุดท้ายนำมาวิเคราะห์ว่าคุณสมบัติเหล่านั้นมีความสัมพันธ์กันอย่างไรบ้าง ซึ่งการวิเคราะห์แบบนี้เปรียบเสมือนการวิเคราะห์จากสิ่งที่เราพบเห็นอยู่ในชีวิตประจำวัน จึงเป็นการง่ายสำหรับผู้พัฒนาระบบ นอกจากนี้ยังสามารถนำผลจากการวิเคราะห์ออกแบบข้างต้นไปพัฒนาต่อ โดยใช้ภาษาการ โปรแกรมแบบ OOP ได้ทันที และสามารถทำงานได้ถูกต้องสอดคล้องกันเป็นอย่างดี เพราะมีแนวคิดของ OOP เหมือนกันเช่น Object Reuse, Inheritance, Polymorphism และ Encapsulation เป็นต้น

นอกจากประโยชน์ในขั้นตอนการออกแบบและวิเคราะห์ระบบงานแล้ว รวมถึงขั้นตอนการพัฒนาโปรแกรมแล้ว การใช้แนวคิด Object-Oriented ยังมีประโยชน์ในการบำรุงรักษาโปรแกรม และการนำ Object ที่พัฒนาแล้วไปใช้กับระบบงานอื่นๆต่อไปได้อีกด้วย เพราะแนวคิดเชิงวัตถุสามารถแก้ไขปรับปรุงโปรแกรมได้สะดวก รวดเร็ว และไม่มีผลกระทบกับส่วนอื่นของระบบได้

5.3 ข้อเสนอแนะ

1. การพัฒนาระบบเว็บไคลเร็กทอรีส่วนตัวนี้ใช้ภาษา ASP ซึ่งในเวอร์ชัน 4.0 นี้ ยังไม่สนับสนุนการใช้งานแบบ Object-Oriented Programming ในการทำงานร่วมกับการออกแบบของ UML ทำให้ต้องมีการ MAP Class Diagram ไปเป็น Relational Table ก่อน ทำให้เสียความยืดหยุ่นของหลักการ OOP ไปบ้าง ในการศึกษาและพัฒนาระบบแบบ Object Paradigm ต่อไปควรได้ศึกษาการใช้ Object Database เช่น CACHE ในการเก็บข้อมูล และใช้ภาษาโปรแกรมที่สนับสนุนเทคนิค OOP เช่น JAVA, Delphi หรือ C++ เป็นต้น
2. ในการศึกษาครั้งนี้ใช้ระบบงานจำลองที่ค่อนข้างเล็ก และไม่ซับซ้อน จิวอาจทำให้มองเห็นประโยชน์ของการพัฒนาระบบด้วยแนวคิดเชิงวัตถุ โดยใช้ UML ไม่ชัดเจน

บรรณานุกรม

จุฬารัตน์ ตันประเสริฐ. 2541. “UML ภาษาพิมพ์เขียวซอฟต์แวร์.” *ไมโครคอมพิวเตอร์*. 16(160) : 106-112.

โชคชัย เตชพรรุ่ง, ผู้แปล. 2538. *นําทงสู่ระบบฐานข้อมูลแบบไคลเอนต์/เซิร์ฟเวอร์*. กรุงเทพฯ : บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน).

Bennett Simon, McRobb Steve and Farmer Ray. 2000. **Object-Oriented System Analyst and Design using UML**. England : McGraw-Hill Publishing Company.

CDG System. 2000. **Object-oriented Knowledge**. [Online]. Available : <http://www.cdg.co.th/gcc/html/oo-knowledge.html>.

Conallen Jim. 1999. **Building Web Application with UML**. London : Addison-Wesley Publishing Company.

Fowler Martin. and Scott Kendall. 1997. **UML Distilled : Applying the Standard Object Modeling Language**. London : Addison-Wesley Publishing Company.

Quartrani Terry. 1998. **Visual Modeling with Rational Rose and UML**. London : Addison Wesley Publishing Company.

Rational Software Corp. **UML Resource Center**. [Online]. Available: <http://www.rational.com/uml>.



ภาคผนวก ก.

Data Dictionary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **Table : AGE** : เก็บข้อมูลเกี่ยวกับช่วงอายุ

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
AGE_ID *	รหัสอายุ	CHAR	10
AGE_LEN	ช่วงอายุ	VARCHAR2	50

2. **Table ; CATEGORIES** : เก็บข้อมูลหมวดหมู่ของลิงก์

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
CATID *	รหัสหมวดหมู่	CHAR	10
CATNAME	ชื่อหมวดหมู่	VARCHAR2	50
CATPARENT	รหัสหมวดหมู่ก่อนหน้า	VARCHAR2	10

3. **Table : COLOR** : เก็บข้อมูลสีที่ชอบ

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
COLOR_ID *	รหัสสี	CHAR	10
COLOR_DESC	สี	VARCHAR2	50

4. **Table : CONTACT** : เก็บข้อมูลที่มีผู้ใช้ส่งติดต่อเข้ามา

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
SEND_NAME	ชื่อผู้ส่ง	VARCHAR2	50
SEND_EMAIL	e-mail ผู้ส่ง	VARCHAR2	50
MESSAGE	สิ่งที่ส่งมา	VARCHAR2	100
CONTACT_ID *	รหัสการติดต่อ	CHAR	10

5. **Table : EDU** : เก็บข้อมูลระดับการศึกษา

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
EDU_ID *	รหัสระดับการศึกษา	CHAR	10
EDU_DESC	สี	VARCHAR2	50

6. **Table : INTEREST** : เก็บข้อมูลเกี่ยวกับความสนใจ

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
INT_ID *	รหัสความสนใจ	CHAR	10
INT_DESC	ความสนใจ	VARCHAR2	50

7. **Table : JOB** : เก็บข้อมูลเกี่ยวกับตำแหน่งงาน

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
JOB_ID *	รหัสงาน	CHAR	10
JOB_DESC	ชื่องาน	VARCHAR2	50

8. **Table : LANGUAGE** : เก็บข้อมูลเกี่ยวกับภาษาที่ใช้

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
LANGUAGE_ID *	รหัสภาษา	CHAR	10
LANGUAGE_DESC	ภาษา	VARCHAR2	50

9. **Table : LINKS** : เก็บข้อมูล Links ที่มีใน YuDir

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
CLICK_NO	จำนวนครั้งที่ถูกเลือก	NUMBER	
CATID *	รหัสหมวดหมู่	CHAR	10
LINKID *	รหัสลิงก์	CHAR	10
URL	URL	VARCHAR2	255
LINKNAME	ชื่อลิงก์	VARCHAR2	64
DESCRIPTION	คำอธิบาย	VARCHAR2	255
EDU_ID	รหัสการศึกษา	CHAR	10
COLOR_ID	รหัสสี	CHAR	10
REL_ID	รหัสศาสนา	CHAR	10
INT_ID	รหัสความสนใจ	CHAR	10
LANGUAGE_ID	รหัสภาษา	CHAR	10
AGE_ID	รหัสอายุ	CHAR	10
JOB_ID	รหัสงาน	CHAR	10

10. **Table : RELIGION** : เก็บข้อมูลเกี่ยวกับศาสนา

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
REL_ID *	รหัสศาสนา	CHAR	10
REL_DESC	ศาสนา	VARCHAR2	50

11. **Table : SUBMIT_LINKS** : เก็บข้อมูล Links ที่มีผู้ใช้ส่งเข้ามาเป็น URL ใหม่

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
SEND_NAME	ชื่อผู้ส่ง	VARCHAR2	50
SEND_EMAIL	e-mail ผู้ส่ง	VARCHAR2	50
LINKID *	รหัสลิงก์	CHAR	10
URL	URL	VARCHAR2	255
LINKNAME	ชื่อลิงก์	VARCHAR2	64
DESCRIPTION	คำอธิบาย	VARCHAR2	255
EDU_ID	รหัสการศึกษา	CHAR	10
COLOR_ID	รหัสสี	CHAR	10
REL_ID	รหัสศาสนา	CHAR	10
INT_ID	รหัสความสนใจ	CHAR	10
LANGUAGE_ID	รหัสภาษา	CHAR	10
AGE_ID	รหัสอายุ	CHAR	10
JOB_ID	รหัสงาน	CHAR	10

12. Table : USERS : เก็บข้อมูลเกี่ยวกับผู้ใช้งาน YuDir

ชื่อ Field	คำอธิบาย	ประเภท	ความยาว
ID *	รหัสผู้ใช้	VARCHAR2	50
LNAME	ชื่อ Login	VARCHAR2	50
FNAME	ชื่อหน้า	VARCHAR2	100
SNAME	ชื่อสกุล	VARCHAR2	100
BDATE	วันที่เกิด	DATE	
EDULEVEL_ID	รหัสระดับการศึกษา	CHAR	10
LCOLOR_ID1	สีที่ชอบ 1	CHAR	10
LCOLOR_ID2	สีที่ชอบ 2	CHAR	10
RELIGION_ID	รหัสศาสนา	CHAR	10
INTEREST_ID1	รหัสความสนใจ 1	CHAR	10
INTEREST_ID2	รหัสความสนใจ 2	CHAR	10
INTEREST_ID3	รหัสความสนใจ 3	CHAR	10
LANGUAGE_ID1	รหัสภาษาที่ใช้ได้ 1	CHAR	10
LANGUAGE_ID2	รหัสภาษาที่ใช้ได้ 2	CHAR	10
LANGUAGE_ID3	รหัสภาษาที่ใช้ได้ 3	CHAR	10
JOB_ID	รหัสงานที่ทำ	CHAR	10
AGE_ID	รหัสช่วงอายุ	CHAR	10

หมายเหตุ : Field ที่มีเครื่องหมาย * คือ Field ที่เป็น Primary Key

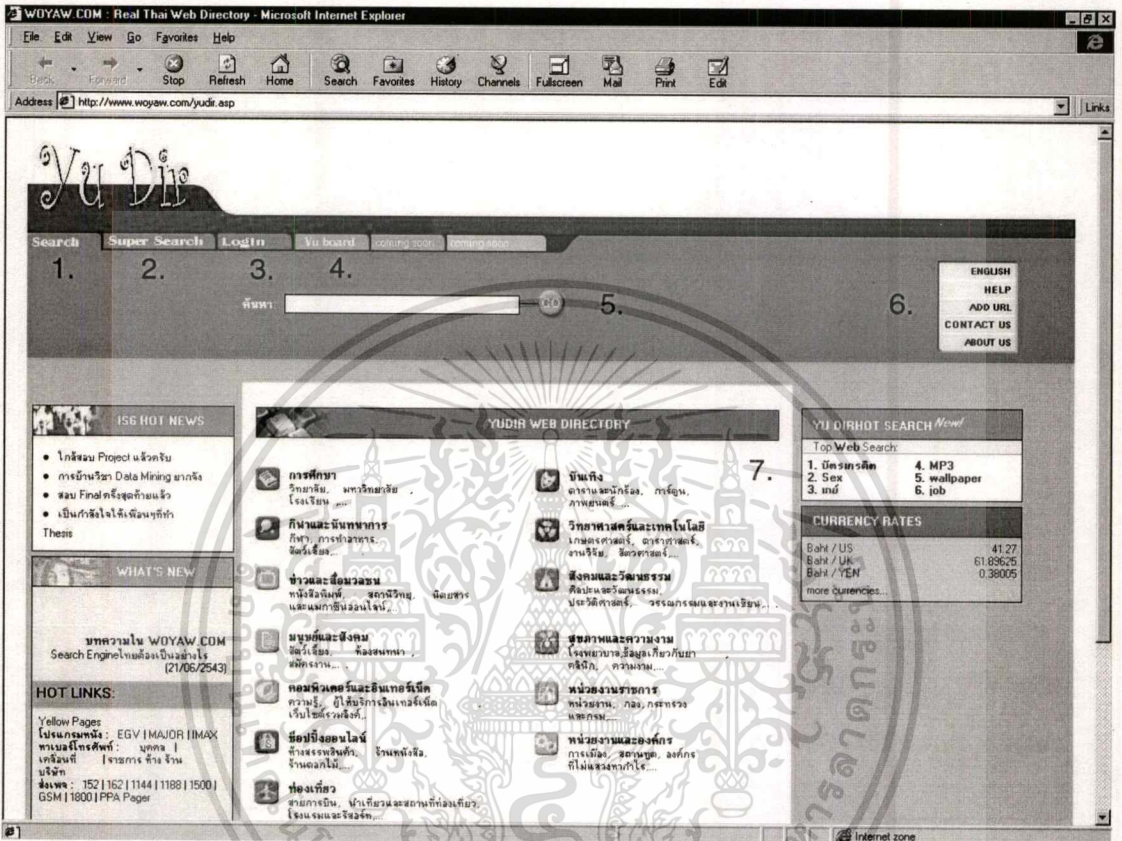


ภาคผนวก ข
ตัวอย่างการใช้งาน เว็บไซต์เรีกทอริส่วนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 : หน้าจอ User Interface

จากรูปที่ ข1. แสดง หน้าจอ Home Page (หน้าจอแรก) ของ YuDir Personal Web Directory



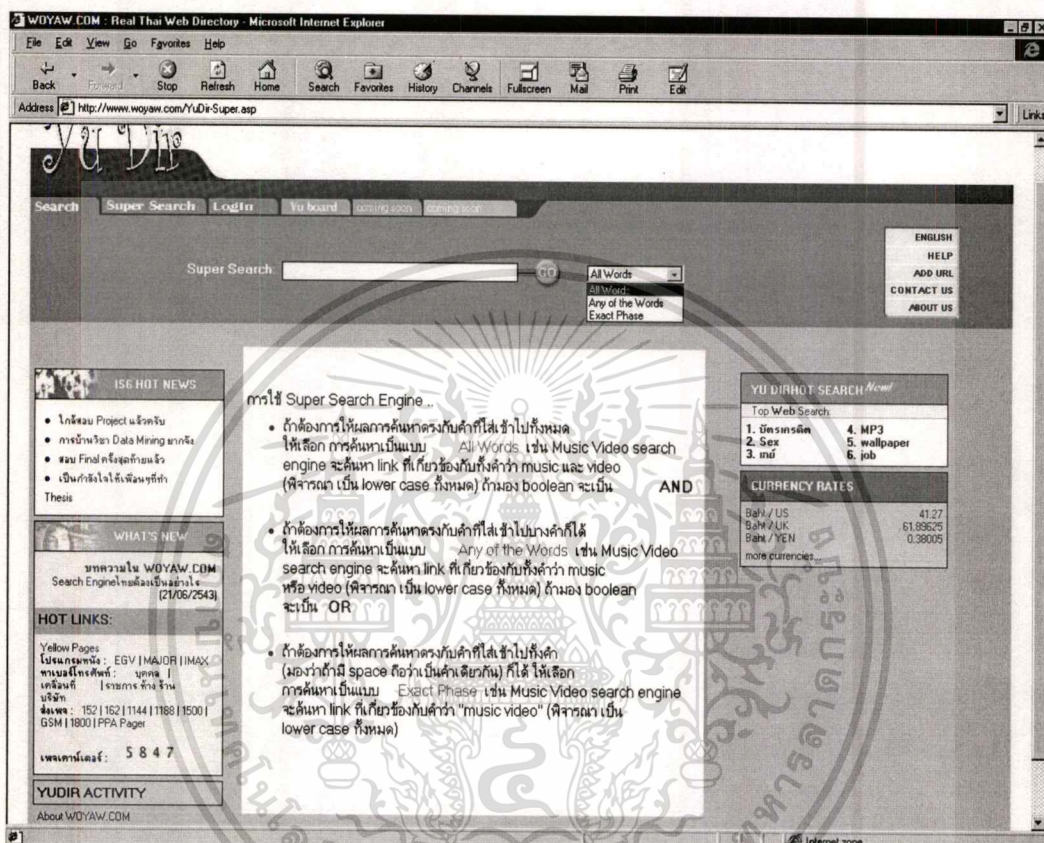
รูปที่ ข1. แสดง หน้าจอ Home Page ของ YuDir Personal Web Directory

ซึ่งประกอบด้วย 7 ส่วนหลักดังนี้

1. จากรูป ข1. เป็นส่วน Search ที่รวมไปกับหน้าจอแรก (Home Page) ผู้ใช้สามารถค้นหาคำที่ต้องการในแบบ Simple Search ได้ โดยใส่คำที่ต้องการค้นหาเข้าไปใน Query String เบอร์ 5. จากรูปที่ ข1. ตัวอย่างผลของการ Search ดูได้จากรูป ข15.
2. จากรูป ข1. เป็นส่วน Super Search เพิ่มความสามารถในการ Search จาก Simple Search ในข้อที่ 1 ดังแสดงหน้าจอในรูปที่ ข2. จะเห็นว่ามีส่วนของการค้นหา 3 แบบคือ
 - All Words เป็นการค้นหาที่ต้องเจอทุกคำที่ผู้ใช้ใส่เข้ามาใน Query String (เป็นลักษณะของ AND)
 - Any of the Words เป็นการค้นหาที่ต้องเจอคำใดคำหนึ่งที่ใช้ใส่เข้ามาใน Query String (เป็นลักษณะของ OR)

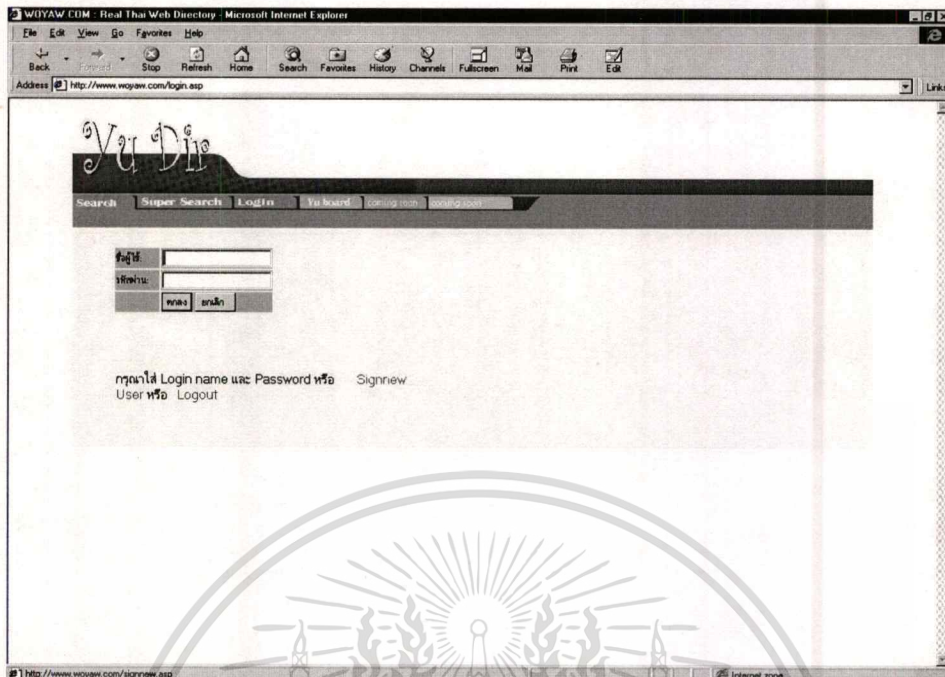
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนสิทธิ์ในเนื้อหา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Exact Phase เป็นการค้นหาที่ต้องเจอทุกคำเรียงตามที่ใช้ใส่เข้ามาใน Query String เท่านั้นจึงจะตรงเงื่อนไข (เป็นลักษณะของ LIKE %.)



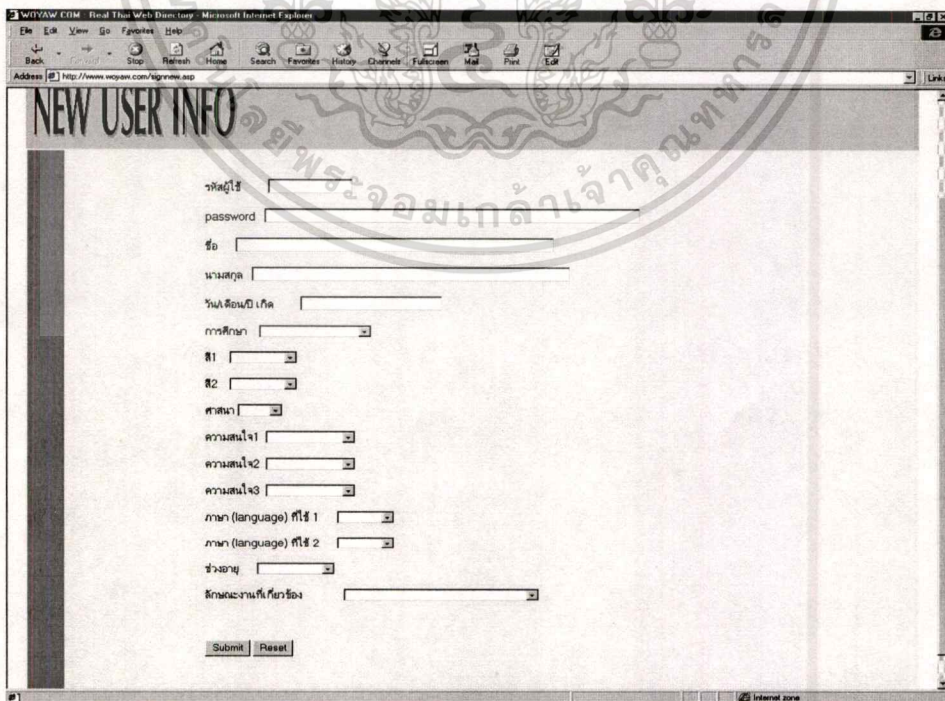
รูปที่ ข2. แสดง หน้าจอ Super Search

- จากรูป ข1. ปุ่ม "LOGIN" ปุ่มที่ 3 นี้ทำหน้าที่เป็นปุ่มเพื่อแสดงหน้าจอ Login เพื่อให้ผู้ใช้ที่สมัครไว้แล้วสามารถใช้ YuDir Personal Web Directory ให้สอดคล้องกับ Personal Information ที่ User ผู้ใช้นั้นได้สมัครไว้กับ YuDir ได้ ดังแสดงในฟอร์ม Login ในรูป ข3. ตรงนี้ผู้ใช้สามารถเลือก Signnew User เพื่อเรียก Form เพิ่ม User ดังแสดงในรูป ข4. หรือ เลือก Logout ออกจากระบบในกรณีที่ Login ไว้แล้วได้



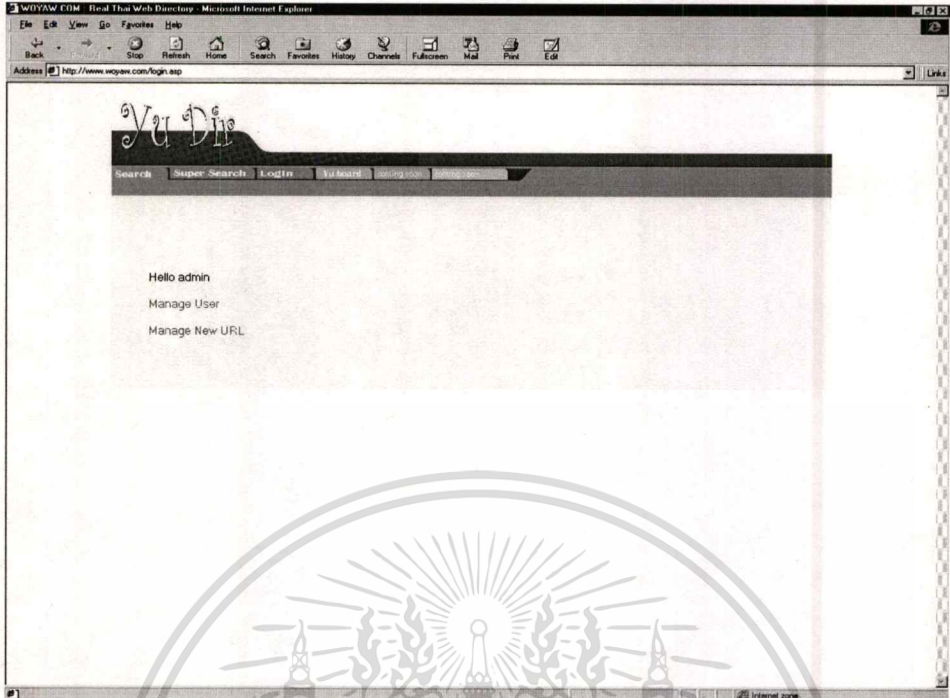
รูปที่ ข3. แสดง หน้าจอ Login

กรณีที่ใช้ Login เป็น Admin จะแสดงหน้าจอ ดังรูป ข5. ให้ผู้ที่ เป็น Admin สามารถเรียก หน้าจอในการบริหารจัดการ USER และ บริหารจัดการ YuDir ได้ดังภาพที่ ข6. และ ข7. ตามลำดับ

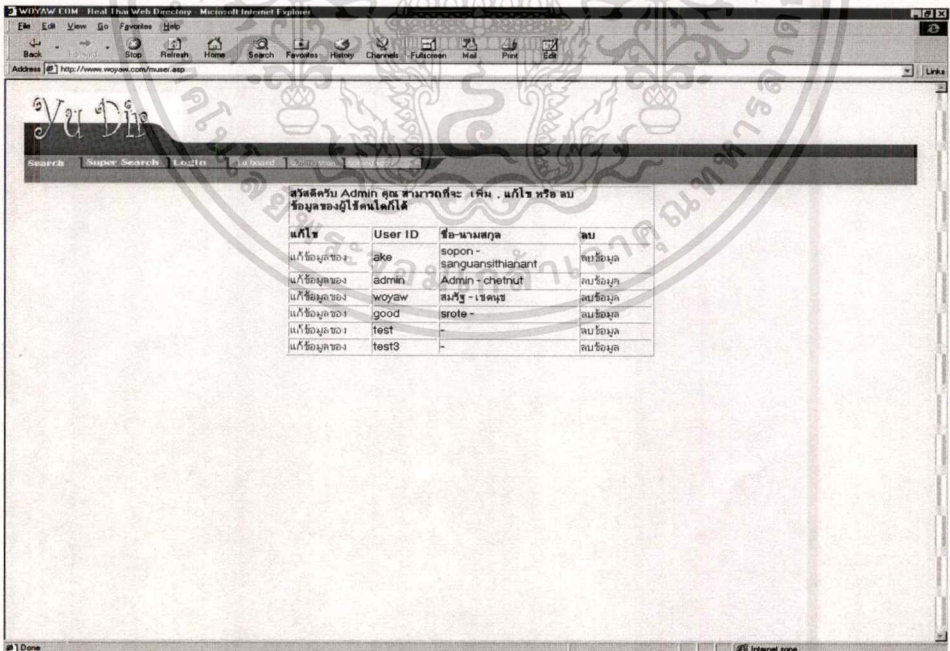


รูปที่ ข4. แสดง หน้าจอ Signnew User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

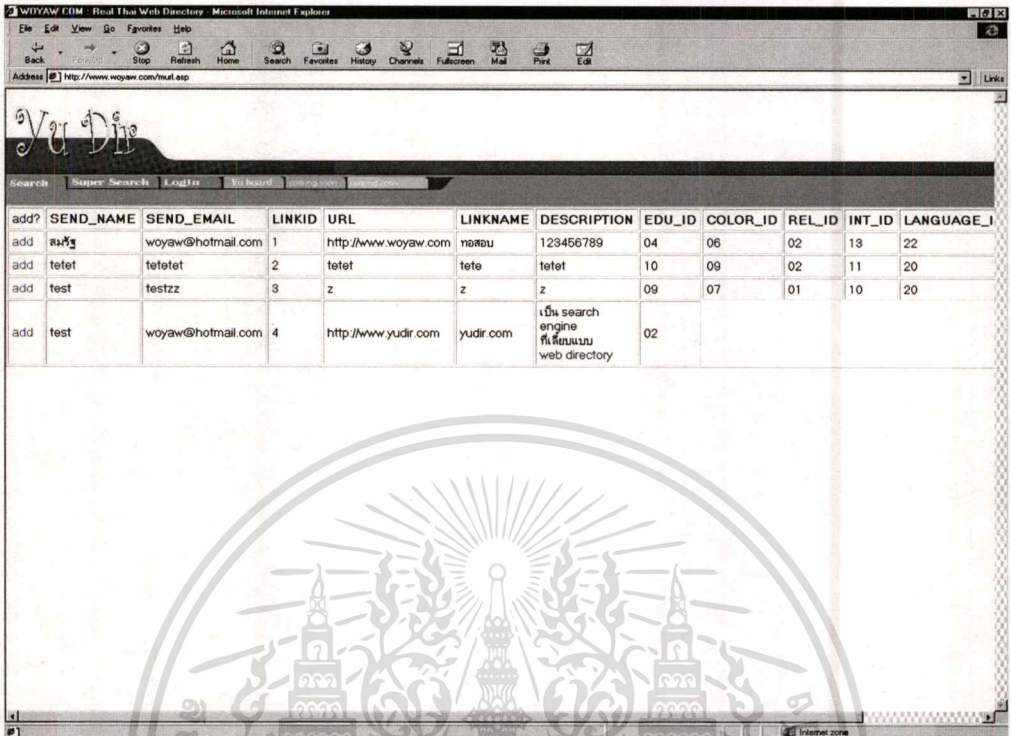


รูปที่ ข5. หน้าจอแสดงเมื่อผู้ใช้ที่เป็น Admin Login สำเร็จ



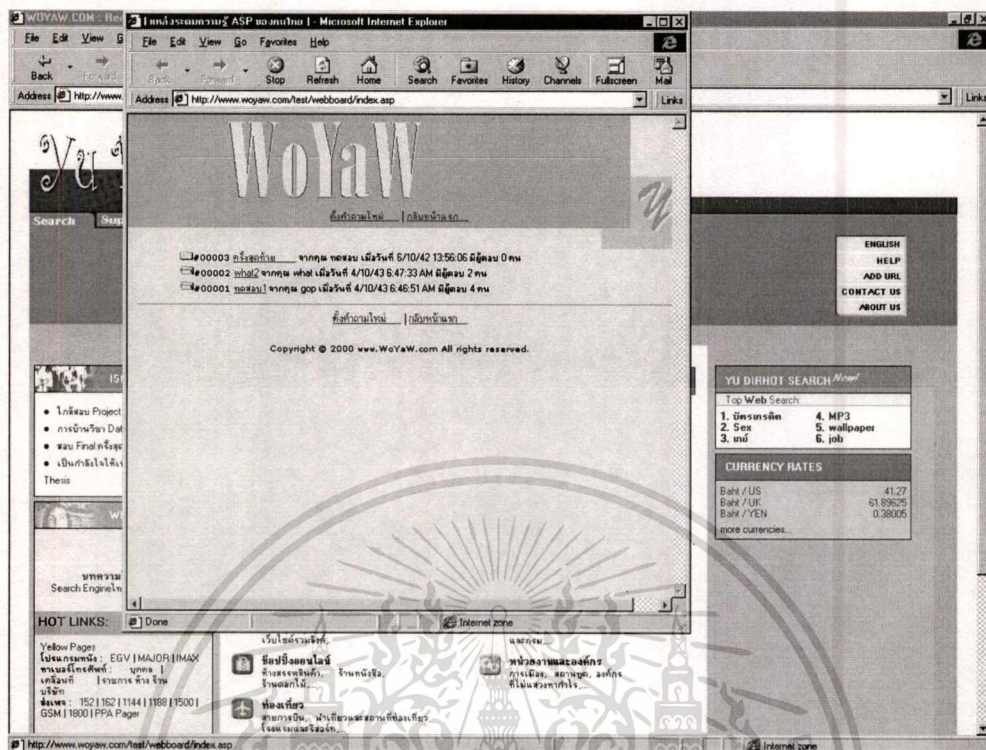
รูปที่ ข6. หน้าจอแสดงเมื่อผู้ใช้ที่เป็น Admin เลือก Manage User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข7. หน้าจอแสดงเมื่อผู้ใช้ที่เป็น Admin เลือก Manage New URL

4. จากรูป ข1. เป็นส่วนของ Webboard ที่มีใน YuDir ดังแสดงในรูป ข8. ผู้ใช้ (ทั้งที่เป็นสมาชิกและไม่ได้เป็นสมาชิก) สามารถตั้งหัวข้อ และตอบหัวข้อในการสนทนาได้



รูปที่ ข8. แสดง หน้าจอ Webboard

5. จากรูป ข1. เป็นส่วนรับ Query String ใน Simple Search
6. จากรูป ข1. ส่วนที่ทำหน้าที่เป็นเมนูประกอบไปด้วยเมนูย่อยดังนี้
 - 6.1 ENGLISH ปรับเปลี่ยน YuDir Personal Web Directory ไปทำงานในโหมดภาษาอังกฤษ
 - 6.2 HELP แสดงข้อความหน้าช่วยเหลือ และวิธีการใช้งาน YuDir Personal Web Directory
 - 6.3 ADD URL แสดงหน้าจอเพิ่ม URL เข้าไปสู่ฐานข้อมูล ดังแสดงหน้าจอในรูปที่ ข-9.
 - 6.4 CONTACT US แสดงหน้าจอเพื่อส่งข้อความติดต่อเข้ามายัง YuDir Personal Web Directory ดังแสดงหน้าจอในรูปที่ ข10.
 - 6.5 ABOUT US แสดงข้อความเกี่ยวกับ YuDir Personal Web Directory

WUYAW.COM Real Thai Web Directory Microsoft Internet Explorer

Address http://www.wuyaw.com/YuDir/url.asp

ADD NEW URL

link id <ไม่ต้องใส่>

ชื่อ (link name)

URL

คำอธิบาย (description)

การศึกษาชั้นคำทียมากให้ดู

สี

สถานที่

ความสนใจ

ภาษา (language) ที่ใช้

อายุชั้นคำทียมากให้ดู

ลักษณะงานที่เกี่ยวข้อง

ชื่อผู้ส่ง

e-mail ผู้ส่ง

รูปที่ ข9. หน้าจอส่วนของการ Add ข้อมูล URL เข้ามาใน YuDir Personal Web Directory

WUYAW.COM Real Thai Web Directory Microsoft Internet Explorer

Address http://www.wuyaw.com/YuDir/Contact.htm

CONTACT US

CONTACT FORM

ชื่อ

อีเมล

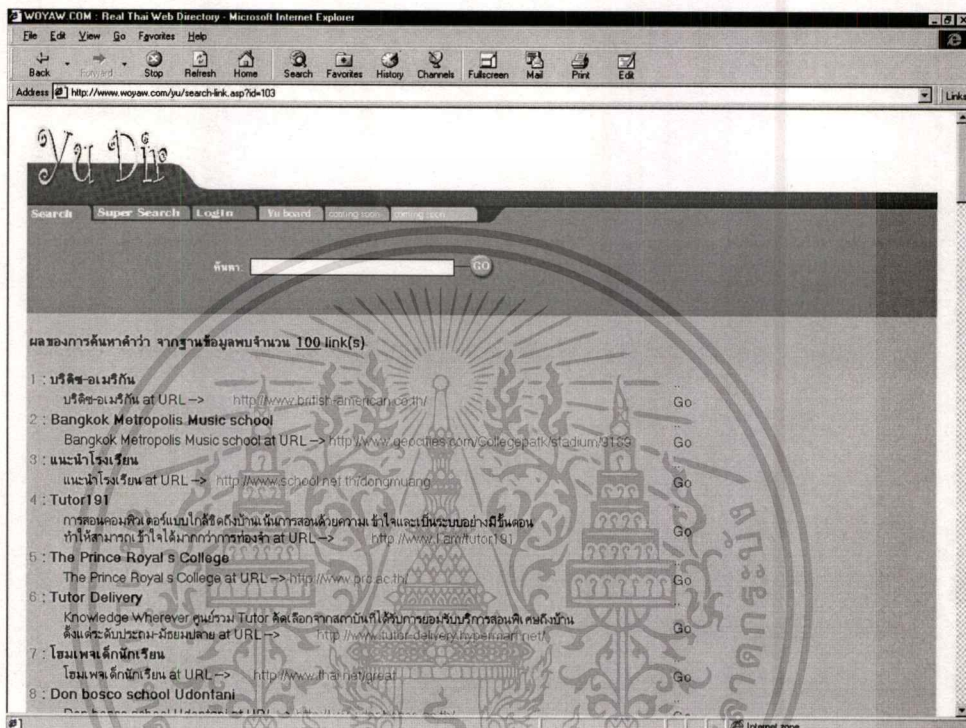
ข้อความ

Copyright © 1999 YuDir Project System. All rights reserved.

รูปที่ ข10. แสดงหน้าจอ Add Contact

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ ข10. แสดงหน้าจอ Add Contact เท่านั้น ไม่ควรนำข้อมูลไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

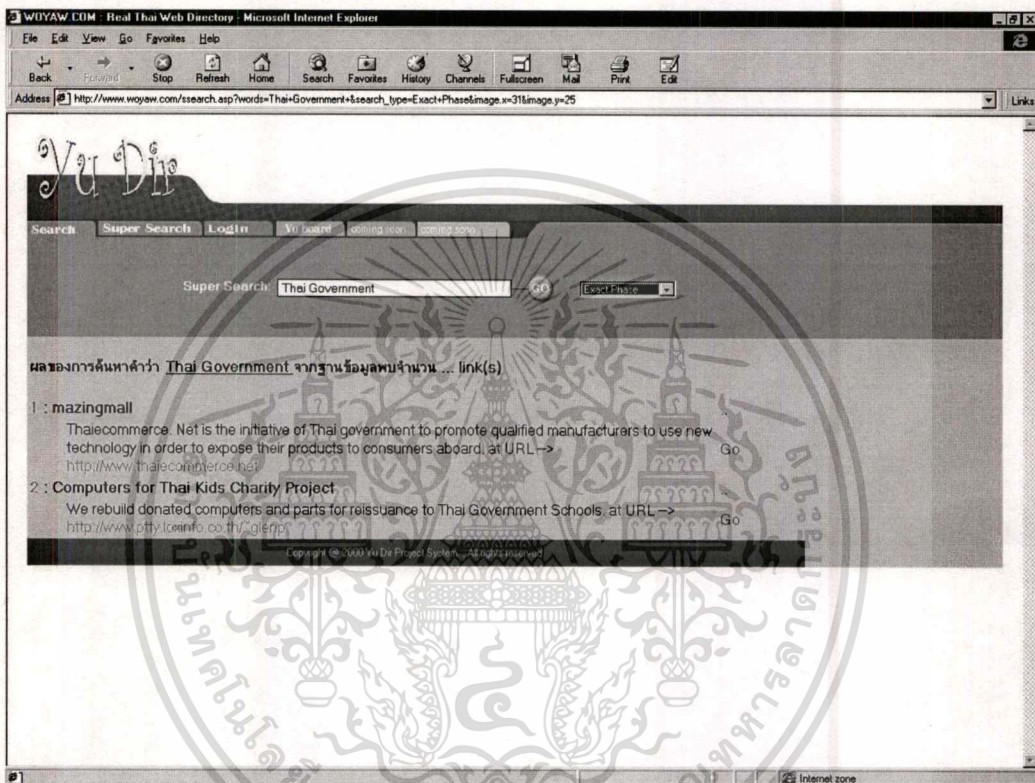
7. จากรูป ข1. แสดงส่วนที่เป็น Web Directory ให้ผู้ใช้สามารถ Follow ไปดูตามหมวดหมู่ที่สนใจ ดังแสดงในรูป ข11. เป็นการเข้าไปในหมวดการศึกษา – โรงเรียน ของ Web Directory



รูปที่ ข11. แสดงหน้าจอ ผลการเดินทางลงไปใน Directory หมวด การศึกษา - โรงเรียน

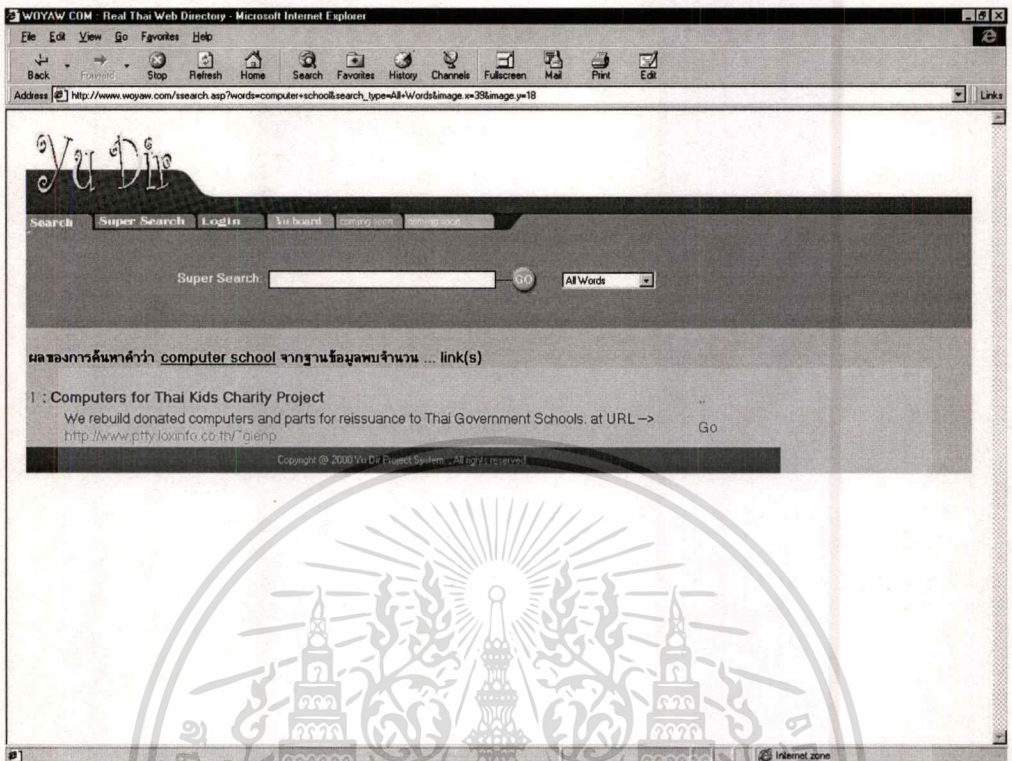
ส่วนที่ 2 : หน้าจอ ตัวอย่างผลลัพธ์ของการ Search แบบต่างๆ

จากรูปที่ ข12. ผลของการ Search จาก Personal Web Directory คำว่า “Thai Government” แบบ Exact Phase



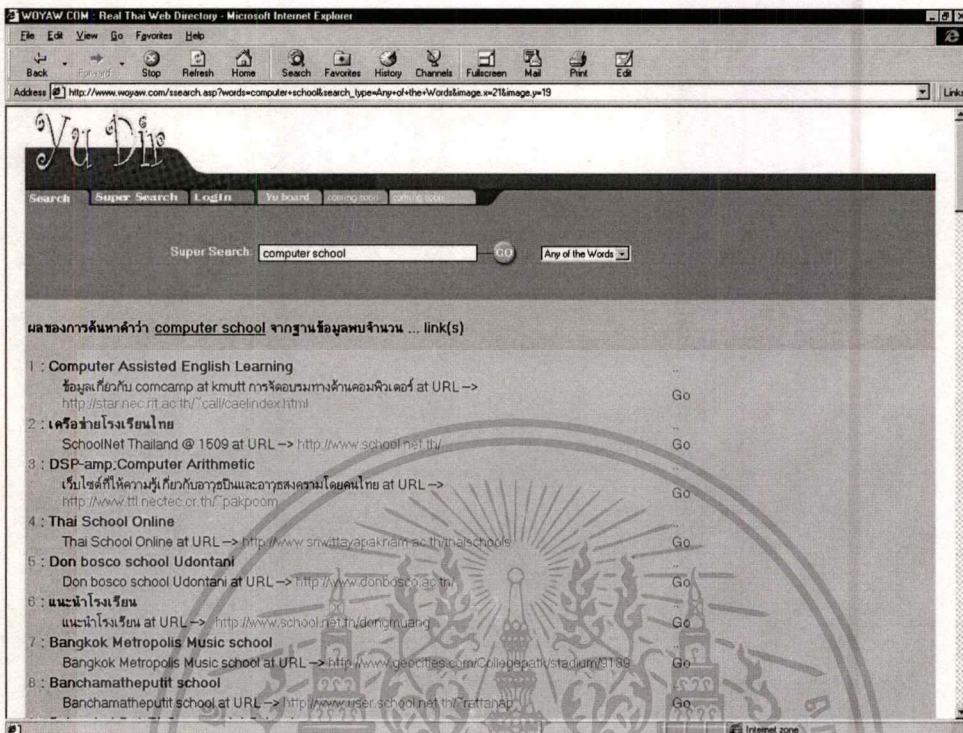
รูปที่ ข12. แสดงหน้าจอ ผลการใช้ Super Search “Thai Government” แบบ Exact Phase

จากรูปที่ ข13. แสดงผลของการ Search จาก Personal Web Directory คำว่า “Computer School” แบบ All Words

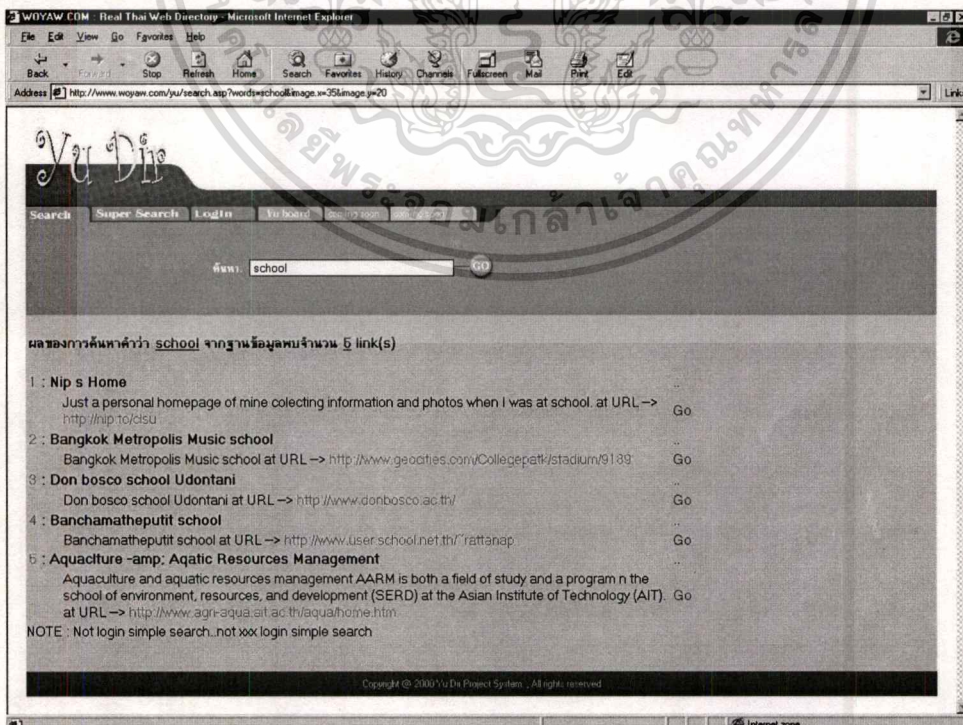


รูปที่ ข13. แสดงหน้าจอ ผลการใช้ Super Search “Computer School” แบบ All Words

จากรูปที่ ข14. แสดงผลของการ Search จาก Personal Web Directory คำว่า “Computer School” แบบ Any of the Words



รูปที่ ข14. แสดงหน้าจอ ผลการใช้ Super Search “Computer School” แบบ Any of the Words



เอกสารนี้เป็นเอกสารที่สง รูปที่ ข15. แสดงหน้าจอ ผลการใช้ Simple Search “School” ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสมรัฐ เชนุช
วันเดือนปีเกิด	28 เมษายน 2520
สถานที่เกิด	อ.เมืองฯ จังหวัดนครสวรรค์
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (คอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	คณะวิศวกรรมและเทคโนโลยีการเกษตร ศูนย์กลางสถาบันเทคโนโลยีราชมงคล คลอง 6. ปีการศึกษา 2540
ปีที่สำเร็จการศึกษา	
ที่ทำงานปัจจุบัน	กองระบบสารสนเทศ, การสื่อสารแห่งประเทศไทย

