

การพัฒนาโปรแกรมบริหารฐานข้อมูลผู้ใช้โดยใช้โพรโทคอล LDAP
The Development of a User Management Program Using LDAP Protocol

โดย

วัชรสิทธิ์ สิงห์โตทอง

รหัส 45066024

อาจารย์ที่ปรึกษา

ผศ. ดร. ภัทรชัย ลลิตโรจน์วงศ์

วัน เดือน ปี.....	18 ก.ย. 2550
เลขทะเบียน.....	02296
เลขเรียกหนังสือ.....	ดง. ๑38ก 2547
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ จอ.ล."	

b11704858
112847027

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2547
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาโปรแกรมบริหารฐานข้อมูลผู้ใช้โดยใช้โปรโตคอล LDAP
นักศึกษา	นาย วัชรสิทธิ์ สิงโตทอง
อาจารย์ที่ปรึกษา	ผศ.ดร. กัทรชัย ลลิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ในปัจจุบันเทคโนโลยีอินเทอร์เน็ตมีการพัฒนาไปอย่างรวดเร็ว และเทคโนโลยีที่พัฒนานั้นรวมไปถึงไคลเอนต์เซิร์ฟเวอร์วิสซึ่งมีส่วนสำคัญในการพัฒนาการให้บริการเว็บเซิร์ฟเวอร์ โดยไคลเอนต์เซิร์ฟเวอร์วิสจะช่วยในการบริหารทรัพยากรของระบบ ควบคุมการเข้าใช้งานของระบบต่างๆ และช่วยสนับสนุนให้ผู้บริหารระบบทำการบริหารระบบรายชื่อผู้ใช้งานเพียงที่เดียว ทำให้ง่ายต่อการออกแบบรักษาความปลอดภัย ส่วนในการติดต่อกับฐานข้อมูลไคลเอนต์เซิร์ฟเวอร์นั้นมีการกำหนดมาตรฐานการติดต่อโดยใช้โปรโตคอล LDAP ในการกำหนดมาตรฐานของข้อความที่ใช้ในการจัดการบริหารไคลเอนต์เซิร์ฟเวอร์ การศึกษาและพัฒนาระบบในโครงการนี้จะจัดทำระบบจำลองเพื่อจัดการข้อมูลรายชื่อผู้ใช้งานโดยใช้เทคโนโลยีไคลเอนต์เซิร์ฟเวอร์วิสตามมาตรฐานโปรโตคอล LDAP

Title	The Development of a User Management Program Using LDAP Protocol
Student	Mr. Vatcharasit Singtothong
Advisor	Asst. Prof. Dr. Pattarachai Lalitrojwong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2004

Abstract

Recently, Internet technology has been developed rapidly. Including a directory services that is important to develop Web services. A directory service helps an administrator to manages system resources, control the authentication to access the system, and manage user data in the same place. This makes security management more efficiently. The database connection uses LDAP protocol to define a communication standard to manage directories. This project report describes the development of a prototype model using a directory service with LDAP protocol.

กิตติกรรมประกาศ

ในการทำโครงการเรื่องการพัฒนาโปรแกรมบริหารฐานข้อมูลผู้ใช้โดยใช้โปรโตคอล LDAP สามารถสำเร็จลุล่วงไปด้วยดี ข้าพเจ้าขอขอบพระคุณ ดร. ภัทรชัย ลลิตโรจน์วงศ์ อาจารย์ที่ปรึกษาวิชาโครงการพัฒนาระบบงานที่กรุณาให้คำแนะนำและเป็นที่ปรึกษา อันเป็นประโยชน์ต่อการพัฒนาระบบ และได้สละเวลาในการตรวจสอบแก้ไขข้อบกพร่องของโครงการฉบับนี้

นอกจากนี้ข้าพเจ้าต้องขอกราบขอบพระคุณ บิดา มารดา และบุคคลในครอบครัว ที่ได้ให้ความสนับสนุนทางด้านกำลังใจในการเรียนจนการทำโครงการพัฒนาระบบนี้สำเร็จด้วยดี รวมทั้งขอขอบคุณเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ ที่เกี่ยวกับโครงการไว้ ณ ที่นี้ด้วย

ผู้จัดทำ

มีนาคม 2548

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	IX
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของการพัฒนาระบบงาน.....	1
1.3 ขอบเขตของการพัฒนาระบบ.....	2
1.4 ขั้นตอนการดำเนินการพัฒนาระบบงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากการพัฒนาระบบงาน.....	3
2. ความรู้เบื้องต้นเกี่ยวกับไคเร็คทอรีเซอรัวิส และทฤษฎีที่เกี่ยวข้อง.....	4
2.1 ไคเร็คทอรีเซอรัวิส.....	4
2.2 โพรโทคอล LDAP.....	5
2.3 การทำงานของโพรโทคอล LDAP.....	6
2.4 การจัดการฐานข้อมูลไคเร็คทอรี.....	8
2.5 สถาปัตยกรรม LDAP.....	8
2.6 JNDI Application Programming Interface.....	18
2.7 สถาปัตยกรรม JNDI.....	19
2.8 กระบวนการทำงานของ JNDI ที่มาใช้ร่วมกับโพรโทคอล LDAP.....	20
2.9 การจัดการกับอ็อบเจกต์ที่อยู่ใน LDAP Server.....	22
2.10 การแก้ไขกับอ็อบเจกต์ที่อยู่ใน LDAP เซิร์ฟเวอร์.....	24
2.11 การลบอ็อบเจกต์จากไคเร็คทอรี.....	25

สารบัญ (ต่อ)

	หน้า
2.12 การค้นหาอีอบเจกต์จากไคลเอนต์	25
2.13 นโยบายในการจัดการเพื่อเพิ่มประสิทธิภาพของ JNDI API	26
3. การวิเคราะห์และออกแบบระบบ	28
3.1 ปัญหาและข้อจำกัดและระบบงานเดิม	28
3.2 ความต้องการของระบบ	29
3.3 การออกแบบระบบการทำงานโดยใช้แผนภาพแสดงการทำงานของระบบ	29
3.4 แผนภาพแสดงการทำงานของระบบ ของกระบวนการจัดการข้อมูลรายชื่อผู้ใช้	30
3.5 แผนภาพแสดงกิจกรรมในการทำงานของระบบและการโต้ตอบในแต่ละอีอบเจกต์ หรือแต่ละคลาส	37
3.6 แผนภาพแสดงคลาสและความสัมพันธ์	43
4. การพัฒนาระบบงาน	50
4.1 กำหนดวัตถุประสงค์	50
4.2 อุปกรณ์และ โปรแกรมที่ใช้ในการพัฒนาระบบฐานข้อมูลผู้ใช้	51
4.3 ขั้นตอนในการพัฒนาระบบฐานข้อมูลผู้ใช้	51
4.4 หน้าจอการทำงานของระบบ	55
5. บทสรุปการออกแบบและพัฒนาระบบงาน	62
5.1 สรุปผลการพัฒนา	62
5.2 อุปสรรคในการพัฒนา	63
5.3 ข้อเสนอแนะ	63
บรรณานุกรม	64
ประวัติผู้เขียน	65

สารบัญตาราง

หน้า

ตารางที่

2.1 รูปแบบของข้อมูลในแอตทริบิวต์.....	10
2.2 การประกาศประเภทของข้อมูลในแอตทริบิวต์	10
2.3 เงื่อนไขต่างๆที่สามารถกำหนดได้ในการค้นหา	16
2.4 เงื่อนไขประเภทบูลีน	16
2.5 การเปรียบเทียบระหว่างการทำงานกับคำสั่งของ JNDI.....	20
3.1 รายละเอียดประกอบการทำงานของ Login	31
3.2 รายละเอียดประกอบการทำงานของ Config Environment.....	31
3.3 รายละเอียดประกอบการทำงานของ Connect	32
3.4 รายละเอียดประกอบการทำงานของ Manipulate Directory	33
3.5 รายละเอียดประกอบการทำงานของ Authenticate	34
3.6 รายละเอียดประกอบการทำงานของ Search Directory.....	34
3.7 รายละเอียดประกอบการทำงานของ Export LDIF	35
3.8 การออกแบบแอตทริบิวต์และ โอเปอเรชั่นของคลาส LDAP Connection	44
3.9 การออกแบบแอตทริบิวต์และ โอเปอเรชั่นของคลาส LDAP ConnectionPool	46
3.10 การออกแบบแอตทริบิวต์และ โอเปอเรชั่นของคลาส LDAPMessage.....	46
3.11 การออกแบบแอตทริบิวต์และ โอเปอเรชั่นของคลาส LDAPException.....	47
3.12 การออกแบบแอตทริบิวต์และ โอเปอเรชั่นของคลาส LDAPEntryWrapper.....	48
3.13 การออกแบบแอตทริบิวต์และ โอเปอเรชั่นของคลาส ConnectionManager.....	48
3.14 การออกแบบแอตทริบิวต์และ โอเปอเรชั่นของคลาส ConfigurationLDAP	49
4.1 แสดงรายละเอียดของเครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนา	51

สารบัญรูป

หน้า

รูปที่

2.1	หลักการทํางานของ LDAP Server	5
2.2	รูปแบบการทํางานของ Stand Alone LDAP Server	6
2.3	การโต้ตอบกันของโปรแกรมประยุกต์กับไคลเอนต์เซิร์ฟเวอร์	7
2.4	ความสัมพันธ์ระหว่างเอนทรีกับแอตทริบิวต์	9
2.5	รายละเอียดต่างๆ ของ Directory Information Tree	11
2.6	การอ้างถึงเอนทรีเซิร์ฟเวอร์	12
2.7	การทํางานของไคลเอนต์เซิร์ฟเวอร์โดยให้เครื่องไคลเอนต์ทํางาน	13
2.8	การทํางานของไคลเอนต์เซิร์ฟเวอร์ในการอ้างถึงไคลเอนต์เซิร์ฟเวอร์	13
2.9	แสดงถึงโครงสร้างในการเชื่อมการติดต่อของ JNDI	19
2.10	โครงสร้างและการทํางานของคลาส refAddr	23
3.1	แผนภาพแสดงการทํางานของระบบจัดการข้อมูลรายชื่อผู้ใช้	30
3.2	แผนภาพจำลองกิจกรรมของกระบวนการ Login	36
3.3	แผนภาพจำลองกิจกรรมของกระบวนการ Connect	37
3.4	แผนภาพจำลองกิจกรรมของกระบวนการ Manipulate Directory	38
3.6	แผนภาพจำลองกิจกรรมของกระบวนการ Search Directory	39
3.7	แผนภาพจำลองกิจกรรมของกระบวนการ Export LDIF	40
3.8	แผนภาพความสัมพันธ์ระหว่างคลาสของระบบจัดการข้อมูลรายชื่อผู้ใช้	41
4.1	หน้าจอการติดตั้ง OpenLDAP	50
4.2	หน้าจอการกำหนดค่าเริ่มต้นของ OpenLDAP	51
4.3	หน้าจอการทดสอบการใช้งาน OpenLDAP	52
4.4	หน้าจอสำหรับเลือกไฟล์คอนฟิกเพื่อติดต่อกับไคลเอนต์เซิร์ฟเวอร์	53
4.5	หน้าจอการระบุข้อมูลสำหรับการติดต่อไคลเอนต์เซิร์ฟเวอร์	54
4.6	หน้าจอการทํางานของเมนู File	55
4.7	หน้าจอการเพิ่มข้อมูล groupOfNames	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.8 หน้าจอกำหนดเงื่อนไขการค้นหา.....	57
4.9 หน้าจอแสดงรายละเอียดของเอนทรี	58
4.10 หน้าจอการนำข้อมูลออกมาในรูปแบบไฟล์ข้อความ	59



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในปัจจุบันในการจัดการเกี่ยวกับระบบรายชื่อผู้ใช้ในองค์กรนั้น ได้แยกการจัดการระบบรายชื่อออกจากกันในแต่ละส่วน แต่ละแผนก ทำให้เกิดความซับซ้อนของระบบข้อมูลรายชื่อผู้ใช้ในองค์กร ทำให้ยุ่งยากในการกำหนดมาตรฐานของข้อมูล และการกำหนดมาตรการในการรักษาความปลอดภัย

จากปัญหาที่เกิดขึ้นจึงมีการนำเอาเทคโนโลยีไคลเอด์ทอรีเซอร์วิส มาใช้ในการพัฒนาระบบฐานข้อมูล ซึ่งระบบฐานข้อมูลแบบไคลเอด์ทอรีเซอร์วิสนั้นเป็นฐานข้อมูลที่จะอำนวยความสะดวกในการนำไปประยุกต์กับงานต่างๆ ได้อย่างหลากหลาย โดยนำเอาความสามารถในการสืบค้นข้อมูลอย่างรวดเร็ว และความสามารถในการจัดการเกี่ยวกับการรักษาความปลอดภัยให้กับฐานข้อมูล โดยใช้การกำหนดสิทธิในการเข้าถึงข้อมูลในฐานข้อมูล เมื่อนำเอาเทคโนโลยีไคลเอด์ทอรีเซอร์วิสมาใช้ในการจัดการเกี่ยวกับฐานข้อมูลผู้ใช้งานในองค์กรแล้ว ทำให้เพิ่มประสิทธิภาพให้กับองค์กรมากยิ่งขึ้น ในรายงานฉบับนี้จะกล่าวถึงเทคโนโลยีไคลเอด์ทอรีเซอร์วิส การวิเคราะห์ การออกแบบ และการพัฒนาโปรแกรมสำหรับบริหารฐานข้อมูลผู้ใช้งานในองค์กรรวมทั้งเครื่องมือที่ใช้ในการพัฒนา

โดยที่ในโครงการพัฒนาโปรแกรมบริหารฐานข้อมูลผู้ใช้โดยใช้โพรโทคอล LDAP นี้ได้ทำการศึกษาและพัฒนาโปรแกรมที่จัดการฐานข้อมูลผู้ใช้ในองค์กรขึ้นมา โดยการจำลองการทำงานของฐานข้อมูลรายชื่อผู้ใช้งานในองค์กร และทำการสร้างไคลเอด์ทอรีที่ใช้ในการเก็บข้อมูล และสร้างโปรแกรมที่ช่วยในการจัดการข้อมูลในไคลเอด์ทอรี และนำเอาความสามารถต่างๆ ของไคลเอด์ทอรีเซอร์วิส มาเพื่อเพิ่มประสิทธิภาพให้กับการจัดการกับฐานข้อมูลรายชื่อผู้ใช้ในองค์กร

1.2 วัตถุประสงค์ของการพัฒนาระบบงาน

การศึกษาโครงการพัฒนาโปรแกรมบริหารฐานข้อมูลผู้ใช้นี้มีวัตถุประสงค์ ดังนี้

1. เพื่อสร้างระบบรายชื่อผู้ใช้งานในองค์กร และจัดการกับข้อมูลผู้ใช้ เพื่อเพิ่มประสิทธิภาพในการจัดการ
2. เพื่อลดความซ้ำซ้อนของข้อมูลรายชื่อผู้ใช้ในองค์กร โดยพัฒนาระบบให้มีการจัดการกับ

เอกสารนี้เป็นเอกสารร่วมกันของศูนย์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพื่อให้การจัดการกับข้อมูลรายชื่อผู้ใช้ให้เป็นมาตรฐานเดียวกัน ทำให้ง่ายในการพัฒนาระบบจัดการรายชื่อผู้ใช้ในอนาคต
4. เพื่อสร้างระบบจัดการรายชื่อผู้ใช้ให้มีประสิทธิภาพ สามารถจัดการได้อย่างรวดเร็ว ปลอดภัย มีมาตรฐาน ความถูกต้อง และยังสามารถรองรับปริมาณงานที่จะเพิ่มขึ้นในอนาคตได้
5. เรียนรู้เกี่ยวกับเทคโนโลยีไคลเอนท์เซิร์ฟเวอร์ที่จัดการตามมาตรฐานโปรโตคอล LDAP
6. เรียนรู้การออกแบบระบบจัดการ และการบริหารข้อมูลรายชื่อผู้ใช้ในองค์กร

1.3 ขอบเขตของการพัฒนาระบบงาน

ในการศึกษาโครงการพัฒนาระบบจัดการข้อมูลรายชื่อผู้ใช้นี้ได้กำหนดขอบเขตของการศึกษาไว้ ดังนี้

1. การวิเคราะห์และออกแบบโครงสร้างของไคลเอนท์เซิร์ฟเวอร์ที่จะนำมาใช้องค์กร
2. การศึกษากระบวนการทำงานของไคลเอนท์เซิร์ฟเวอร์ และรูปแบบของคำสั่งที่ใช้จัดการตามมาตรฐานโปรโตคอล LDAP
3. พัฒนาโปรแกรมที่ทำการติดต่อเพื่อร้องขอใช้บริการไคลเอนท์เซิร์ฟเวอร์ในการจัดการเกี่ยวกับข้อมูลรายชื่อผู้ใช้ในองค์กร ตัวอย่างเช่น การเพิ่มบัญชีผู้ใช้งาน ลบ และค้นหารายชื่อผู้ใช้จากบัญชีรายชื่อผู้ใช้งาน
4. พัฒนาโปรแกรมในด้านการรักษาความปลอดภัย โดยสามารถกำหนดสิทธิการใช้งานของผู้ใช้งาน และสามารถกำหนดนโยบายรักษาความปลอดภัยต่างๆ ให้กับโปรแกรมได้

1.4 ขั้นตอนการดำเนินการพัฒนาระบบงาน

การทำการศึกษาระบบจัดการข้อมูลรายชื่อผู้ใช้สำหรับองค์กรนี้ มีขั้นตอนในการดำเนินการ ดังนี้

1. ศึกษาหลักการทำงานไคลเอนท์เซิร์ฟเวอร์ รูปแบบการจัดเก็บข้อมูล และกระบวนการทำงานต่างๆ
2. ศึกษารูปแบบของคำสั่งในการทำงานกับไคลเอนท์เซิร์ฟเวอร์ตามมาตรฐานโปรโตคอล LDAP
3. ศึกษาภาษา และเครื่องมือที่ใช้ในการพัฒนาระบบ
4. ทำการออกแบบฐานข้อมูลผู้ใช้ที่จะนำมาใช้กับองค์กร โดยอ้างอิงกับเทคโนโลยีไคลเอนท์เซิร์ฟเวอร์
5. ออกแบบการทำงานของระบบโดยใช้ภาษา UML ร่วมในการออกแบบและอธิบาย

6. ออกแบบหน้าจอบริบทเว็บแอปพลิเคชันในส่วนติดต่อกับผู้ใช้
7. ทำการพัฒนาระบบ โดยสามารถแยกเป็น 2 ส่วนคือ
 - 7.1 ส่วนผู้ให้บริการไคลเอนต์เซิร์ฟเวอร์เป็นส่วนที่ทำการจัดเก็บข้อมูลรายชื่อผู้ใช้ในรูปแบบไคลเอนต์ โดยทำหน้าที่เป็นไคลเอนต์เซิร์ฟเวอร์
 - 7.2 ส่วนของโปรแกรมประยุกต์ที่ใช้เป็นส่วนของผู้ใช้ ทำหน้าที่ในการติดต่อกับ ไคลเอนต์เซิร์ฟเวอร์ และส่งข้อความร้องขอในการจัดการกับข้อมูลรายชื่อผู้ใช้
8. ทดสอบการทำงานโดยทำการติดต่อสื่อสารระหว่างสองส่วน โดยทั้งสองส่วนนั้นต้องมีความเข้าใจในข้อความของกันและกันและสามารถทำงานจัดการข้อมูลผู้ใช้ได้ตรงตามข้อความที่ร้องขอ
9. ตรวจสอบระบบ และแก้ไขข้อผิดพลาดของระบบทั้งสองส่วน
10. ประเมินผลและวิเคราะห์ผลที่ได้รับ
11. สรุปผลจากการศึกษา

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการพัฒนาระบบงาน

ประโยชน์ที่คาดว่าจะได้รับจากการศึกษาโครงการระบบจัดการข้อมูลรายชื่อผู้ใช้สำหรับองค์กร มีดังนี้

1. สามารถเข้าใจหลักการทำงานของเทคโนโลยีไคลเอนต์เซิร์ฟเวอร์ และการจัดการกับไคลเอนต์ตามมาตรฐาน โพรโทคอล LDAP ได้
2. สามารถนำเอาความสามารถของไคลเอนต์มาใช้ในการพัฒนาโปรแกรม และนำโปรแกรมที่พัฒนามาประยุกต์ใช้กับฐานข้อมูลรายชื่อผู้ใช้ในองค์กร ทำให้การทำงานขององค์กรมีประสิทธิภาพมากยิ่งขึ้น
3. สามารถนำเอาแนวคิดและทฤษฎีต่าง ๆ ที่ได้จากการศึกษาในการทำโครงการไปประยุกต์ใช้เพื่อเป็นประโยชน์ในการพัฒนาระบบงานอื่นๆ ได้อย่างมีประสิทธิภาพ
4. ได้เรียนรู้และแนวทางในการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นจากการพัฒนาโครงการ เพื่อเป็นประโยชน์ในการทำงานและการจัดการงานต่าง ๆ ในอนาคตต่อไป

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับไคลเอนต์เซิร์ฟเวอร์ และทฤษฎีที่เกี่ยวข้อง

ในปัจจุบันระบบธุรกิจได้อาศัยการติดต่อธุรกิจผ่านระบบเครือข่าย ซึ่งมีการทำงานแบบกระจายที่มีจำนวนมากขึ้นเรื่อยๆ แต่จะประสบปัญหาความซ้ำซ้อนของข้อมูลและความปลอดภัยของข้อมูล ต่อมาจึงมีการพัฒนาระบบธุรกิจให้มีการทำงานแบบรวมศูนย์เพื่อลดความซ้ำซ้อนของข้อมูล และง่ายต่อการรักษาความปลอดภัย ดังนั้น จึงมีการพัฒนาระบบมาเป็นแบบรวมศูนย์ทรัพยากรต่างๆ ได้ถูกนำใช้ร่วมกัน โดยโปรแกรมประยุกต์จากผู้ใช้หลายคน ทำให้ลดความซ้ำซ้อนที่เกิดจากการจัดเก็บข้อมูลไว้หลายที่ แต่จะเกิดปัญหาในการรักษาความปลอดภัย จะป้องกันอย่างไรไม่ให้ผู้ที่ไม่ได้รับอนุญาตเข้ามาใช้ทรัพยากรในองค์กรได้

ตัวอย่างทรัพยากรที่ใช้ร่วมกันในระบบเครือข่าวนั้นอาจจะเป็น ข้อมูล ไฟล์ หรือ เครื่องพิมพ์ เป็นต้น ดังนั้น จึงมีการนำเอาทรัพยากรเหล่านี้มาเก็บในรูปแบบของฐานข้อมูลแบบพิเศษ ที่เรียกว่าไคลเอนต์เซิร์ฟเวอร์ และฐานข้อมูลแบบไคลเอนต์เซิร์ฟเวอร์นี้มีคุณสมบัติพิเศษในการเข้าถึงข้อมูลสามารถทำได้อย่างรวดเร็ว รองรับการทำงานกับผู้ใช้จำนวนมาก และมีการรักษาความปลอดภัยกับข้อมูลที่เก็บไว้อีกด้วย ฐานข้อมูลแบบไคลเอนต์เซิร์ฟเวอร์จึงเหมาะกับระบบที่ทำงานแบบรวมศูนย์ และมีผู้ใช้จำนวนมาก

2.1 ไคลเอนต์เซิร์ฟเวอร์ (Sarwar, 2002)

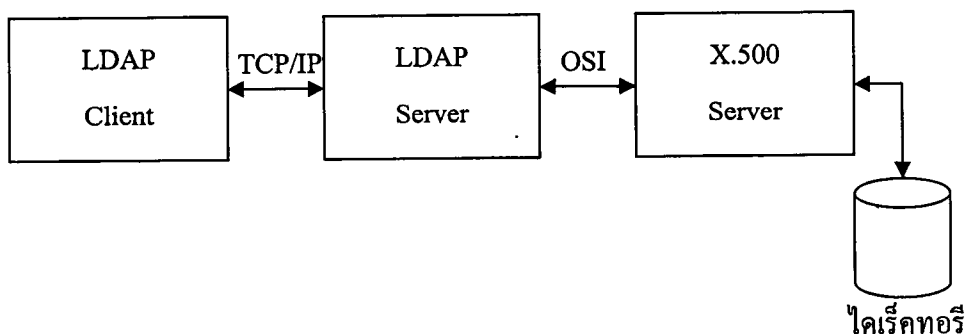
ไคลเอนต์เซิร์ฟเวอร์เปรียบได้กับสมุดโทรศัพท์ สมุดโทรศัพท์เป็นไคลเอนต์ประเภทหนึ่งที่ทำหน้าที่ในการให้ข้อมูลเกี่ยวกับชื่อ นามสกุล ที่อยู่ และหมายเลขโทรศัพท์ ข้อมูลที่เก็บอยู่ในสมุดโทรศัพท์นั้นจะมีการเก็บข้อมูลต่างๆของผู้ใช้โทรศัพท์ แบ่งออกเป็นหมวดหมู่เพื่อให้ง่ายต่อการใช้งาน และสะดวกรวดเร็ว เช่นเดียวกับการทำงานไคลเอนต์เซิร์ฟเวอร์ที่มีการแบ่งประเภทของข้อมูลต่างๆเก็บลงไปไคลเอนต์เซิร์ฟเวอร์ ไคลเอนต์เซิร์ฟเวอร์ทำการเก็บข้อมูลโดยแบ่งข้อมูลออกเป็น โหนดๆ เรียกว่า เอนทรี ไคลเอนต์เซิร์ฟเวอร์ทำการเก็บข้อมูลในรูปแบบของโครงสร้างต้นไม้โดยสร้างเป็นกลุ่มของความสัมพันธ์ในแต่ละเอนทรี โดยที่เอนทรีชั้นบนมีคุณสมบัติเหมือนกับเอนทรีชั้นล่าง แต่เอนทรีชั้นล่างมีคุณลักษณะพิเศษที่เพิ่มเติมขึ้นมาจากเอนทรีบน รูปแบบการอ้างอิงแต่ละเอนทรีของไคลเอนต์

ทอร์นั้นมึลักษณะเดียวกับการอ้างอิงของชื่อ โดเมน โดยการบ่งชี้ไปยังเอนทรีที่ต้องการนั้นจะใช้ชื่อเฉพาะของเอนทรีที่อยู่เหนือเอนทรีที่ต้องการมาเป็นตัวอ้างอิง

จากที่กล่าวมาข้างต้นจะเห็นว่าไคเร็คทอรีสามารถที่จะจัดเก็บข้อมูลต่างๆ ของบุคคล อุปกรณ์ต่างๆ ในองค์กร ค่าเริ่มต้นของ โปรแกรมต่างๆ ที่ทำงานร่วมกับไคเร็คทอรี ทำให้เราสามารถที่จะทำการออกแบบระบบรักษาความปลอดภัยให้กับระบบได้ง่าย และทำให้การควบคุมการใช้งานคอมพิวเตอร์ในระบบเครือข่ายให้มีประสิทธิภาพสูงสุด นอกจากนี้โครงสร้างของไคเร็คทอรีถูกออกแบบให้มีความเร็วในการทำงานค้นหาข้อมูลในฐานะข้อมูลมากกว่าฐานข้อมูลแบบ Relational Database และรองรับการทำงานในระบบเครือข่ายที่มีผู้ใช้จำนวนมาก ข้อเสียของระบบการเก็บข้อมูลแบบไคเร็คทอรีนั้น ไม่เหมาะกับงานที่มีการบันทึกข้อมูลบ่อยๆ เพราะฐานข้อมูลแบบไคเร็คทอรีนั้นไม่มีระบบตรวจสอบทรานแซกชัน ในการบันทึกข้อมูลเมื่อมีปัญหาเราจึงไม่สามารถตรวจสอบการบันทึกข้อมูลได้

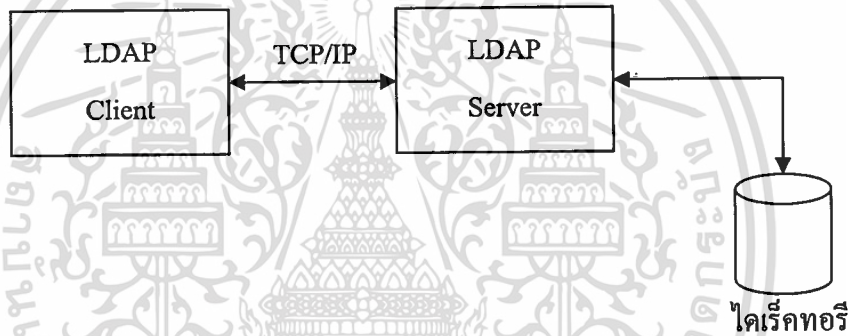
2.2 โพรโทคอล LDAP (Sarwar. 2002)

LDAP หรือ Lightweight Directory Service Protocol คือโพรโทคอลสำหรับการสื่อสาร โดยมีรูปแบบการส่งและรูปแบบข้อความที่ส่งเป็นมาตรฐานสำหรับติดต่อระหว่างเครื่องรับบริการกับเครื่องบริการ เพื่อจัดการกับฐานข้อมูลแบบไคเร็คทอรีตามมาตรฐานของ X.500 ซึ่ง X.500 เป็นมาตรฐานของฐานข้อมูลที่มีการเก็บข้อมูลแบบไคเร็คทอรี กำหนดมาตรฐานโดย คณะกรรมการที่ปรึกษาการโทรศัพท์และโทรเลขระหว่างประเทศ หรือ CCITT ซึ่งกระบวนการทำงานจะแบ่งออกเป็นสองส่วน เริ่มต้นจาก โปรแกรมประยุกต์ทางฝั่งเครื่องรับบริการจะทำงานโดยเรียก LDAP API ให้ส่งข้อความตามมาตรฐานของโพรโทคอล LDAP เพื่อทำการสร้างการเชื่อมต่อระหว่างโปรแกรมประยุกต์ และไคเร็คทอรีเครื่องให้บริการ เนื่องจากในการติดต่อในระบบเครือข่าย อินเทอร์เน็ตจะทำงานบนชั้น TCP/IP แต่การจัดการกับไคเร็คทอรีนั้นจะทำงานบนชั้น OSI



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.1 หลักการทำงานของ LDAP เครื่องให้บริการ ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตาม การเชื่อมต่อโดยอาศัยโพรโทคอล LDAP ยังสามารถเข้าถึงไคลเอนต์โดยไม่ต้องอาศัย X.500 Server ก็สามารถจัดการกับไคลเอนต์ได้ด้วยตัวเองเรียกว่า Stand Alone LDAP Server โดย LDAP Server ชนิดนี้จะทำการจัดการกับไคลเอนต์ได้โดยตรง ซึ่งสามารถทำงานได้ทุกความสามารถที่ LDAP Server ทั่วไป แต่อาจจะไม่สามารถทำงานบางประเภทได้เท่ากับความสามารถที่ X.500 Server ทำให้ การทำงานของ Stand Alone LDAP Server สามารถแสดงได้ดังรูปที่ 2.2



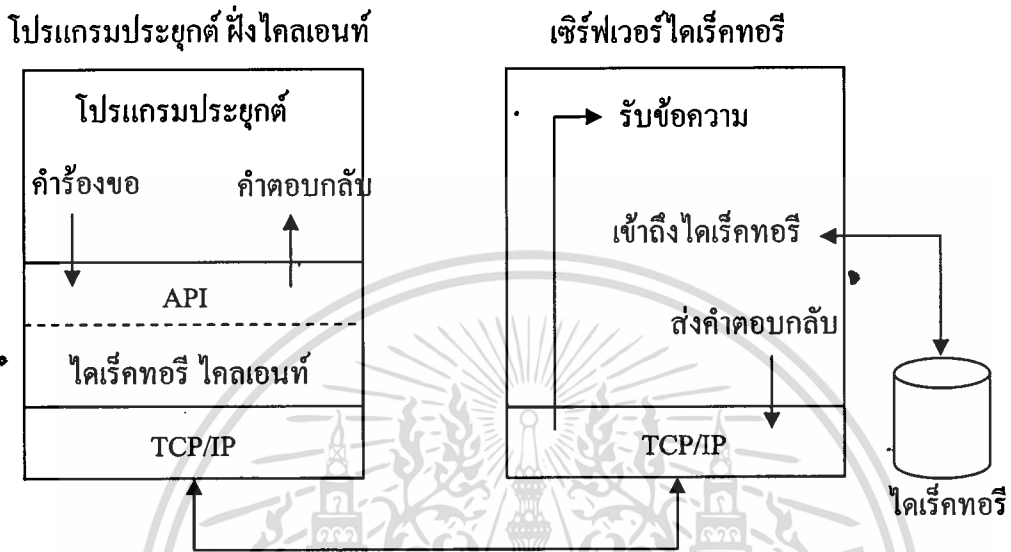
รูปที่ 2.2 รูปแบบการทำงานของ Stand Alone LDAP Server

2.3 การทำงานของโพรโทคอล LDAP (Han, 2001)

การเข้าถึงฐานข้อมูลแบบไคลเอนต์ของโพรโทคอล LDAP นั้นจะใช้รูปแบบการติดต่อแบบเครื่องรับบริการ/เครื่องให้บริการ สำหรับระบบที่มีการประมวลผลแบบกระจาย โดยโปรแกรมประยุกต์นั้นจะไม่เข้าถึงฐานข้อมูลแบบไคลเอนต์โดยตรง แต่โปรแกรมประยุกต์จะทำการเรียกฟังก์ชันในเครื่อง หรือส่วนต่อประสานโปรแกรมประยุกต์ API เป็นตัวแทนในการทำงานส่งข้อความร้องขอถึงเครื่องให้บริการไคลเอนต์ หลังจากนั้นเครื่องให้บริการไคลเอนต์ก็จะทำการส่งข้อความตอบโต้กลับไปยังโปรแกรมประยุกต์ของเครื่องรับบริการโดยผ่าน LDAP API เป็นตัวกลางอีกทีหนึ่ง โดยข้อความที่ติดต่อกันนั้นเป็นข้อความตามมาตรฐานของโพรโทคอล LDAP ซึ่งการส่งข้อความนั้นจะทำบนชั้น TCP/IP โดยการทำงานหลังจากนั้น ไม่ว่าจะเป็นการสร้างการเชื่อมต่อยกเลิกการเชื่อมต่อ หรือจัดการกับฐานข้อมูลทางโปรแกรมประยุกต์จะทำงานผ่าน LDAP API ตลอด ซึ่ง LDAP API นั้นจะทำการเปลี่ยนความต้องการให้อยู่ในรูปของข้อความตามมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LDAP ซึ่งสามารถแสดงให้เห็นถึงเส้นทางการตอบโต้กันของโปรแกรมประยุกต์ของเครื่องเครื่องรับบริการกับเครื่องให้บริการไคลเอนท์ได้ดังรูปที่ 2.3



รูปที่ 2.3 การโต้ตอบกันของโปรแกรมประยุกต์กับเครื่องให้บริการไคลเอนท์

ซึ่งการทำงานระหว่างโปรแกรมประยุกต์ทางฝั่งเครื่องรับบริการ กับเครื่องให้บริการไคลเอนท์มีกระบวนการทำงานหลักๆดังนี้ (Ham, 2001)

- เครื่องรับบริการทำการสร้างการติดต่อกับเครื่องให้บริการไคลเอนท์ โดยจะเรียกขั้นตอนนี้ว่า Binding โดยเครื่องรับบริการจะทำการส่งเลขที่อยู่ไอพี และเลขที่พอร์ตให้กับไคลเอนท์เครื่องให้บริการทำการตรวจจาก ACL เพื่อตรวจสอบเพื่อรับรองสิทธิของผู้ใช้และข้อมูลที่ส่งเพื่อจัดเตรียมในการสร้างการติดต่อนี้จะมีการรักษาความปลอดภัยในการเข้ารหัสลับข้อมูล
- เครื่องรับบริการทำงานกับข้อมูลในไคลเอนท์ โดยการทำงานของทางเครื่องเครื่องรับบริการนั้นจะทำงานตามลำดับ โดยการทำงานนั้นจะทำการค้นหาลักษณะเฉพาะของข้อมูลที่ต้องการได้โดยเครื่องให้บริการไคลเอนท์นั้นจะทำการค้นหาจากเอนทรีโดยดูจากชื่อเฉพาะที่อ้างอิงไปยังยังเอนทรีที่ต้องการ
- เมื่อเครื่องรับบริการทำงานเสร็จแล้วจะทำการปิด หรือยกเลิกการติดต่อโดยส่งข้อความไปยังเครื่องให้บริการไคลเอนท์เพื่อยกเลิกการติดต่อ เรียกขั้นตอนนี้ว่า

Unbinding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกระบวนการต่างๆ ข้างต้นนั้นจะเป็นหน้าที่การทำงานของ LDAP API ในการใช้ LDAP API จะช่วยให้โปรแกรมประยุกต์จัดการกับไคลเอนต์ทำได้ง่ายขึ้น และลดการทำงานของโปรแกรมประยุกต์ที่ต้องจัดการ

2.4 การจัดการฐานข้อมูลไคลเอนต์

ข้อมูลที่เกี่ยวข้องในไคลเอนต์จะถูกอธิบายโดยเอ็นทรี โดยเอ็นทรินั้นจะอธิบายถึงอ็อบเจกต์ต่างๆ ที่เก็บในไคลเอนต์ ชื่อของเอ็นทรินั้นจะเรียกว่า Distinguished Name (DN) ซึ่งประกอบไปด้วย Relative Distinguished Names (RDNS) โดยมีการจัดเรียงในรูปแบบต้นไม้เชิงลำดับชั้น ที่เรียกว่า Directory Information Tree

การจัดการในการเข้าถึง และการจัดการในไคลเอนต์ นั้นจะประกอบไปด้วยการจัดการต่างๆ ดังนี้ (Weltman. 2003)

- การค้นหาเอ็นทรีที่ผู้ใช้ต้องการจากลักษณะพิเศษที่ผู้ใช้กำหนด
- การเพิ่มเอ็นทรีเข้าไปในไคลเอนต์
- การลบเอ็นทรีจากไคลเอนต์
- การเปลี่ยนแปลงเอ็นทรี
- การเปลี่ยนแปลง Distinguished Name หรือ Relative Distinguished Names ของเอ็นทรี เรียกว่าเป็นการเคลื่อนย้ายเอ็นทรินั้นเอง
- การเปรียบเทียบเอ็นทรี

2.5 สถาปัตยกรรม LDAP

LDAP ได้ถูกอธิบายว่าเป็นกลุ่มของข้อความที่มีความหมาย ที่แลกเปลี่ยนระหว่างเครื่องรับบริการ LDAP และ เครื่องให้บริการLDAP โดยข้อความเหล่านี้อาจจะประกอบไปด้วย กระบวนการที่เครื่องเครื่องรับบริการร้องขอ ตัวอย่างเช่น ค้นหา เปลี่ยนแปลง ลบ เป็นต้น หรือข้อความตอบรับการร้องขอจากเครื่องให้บริการ หรือข้อมูลที่เครื่องเครื่องรับบริการต้องการ ซึ่งข้อความต่างๆ นั้นจะอิงตามมาตรฐานของโพรโทคอล LDAP สามารถอธิบายสถาปัตยกรรมโพรโทคอล LDAP โดยใช้รูปแบบ 4 รูปแบบหลักดังนี้ (Weltman. 2003)

1. Information อธิบายโครงสร้างของข้อมูลที่ถูกเก็บไว้ในไคลเอนต์LDAP
2. Naming อธิบายถึงการจัดระบบ และตัวบ่งชี้ข้อมูลในไคลเอนต์ LDAP
3. Functional อธิบายถึงกระบวนการที่กระทำกับข้อมูลที่ถูกเก็บอยู่ในไคลเอนต์ LDAP
4. Security อธิบายการรักษาความปลอดภัยของข้อมูลที่ถูกเก็บอยู่ในไคลเอนต์ LDAP

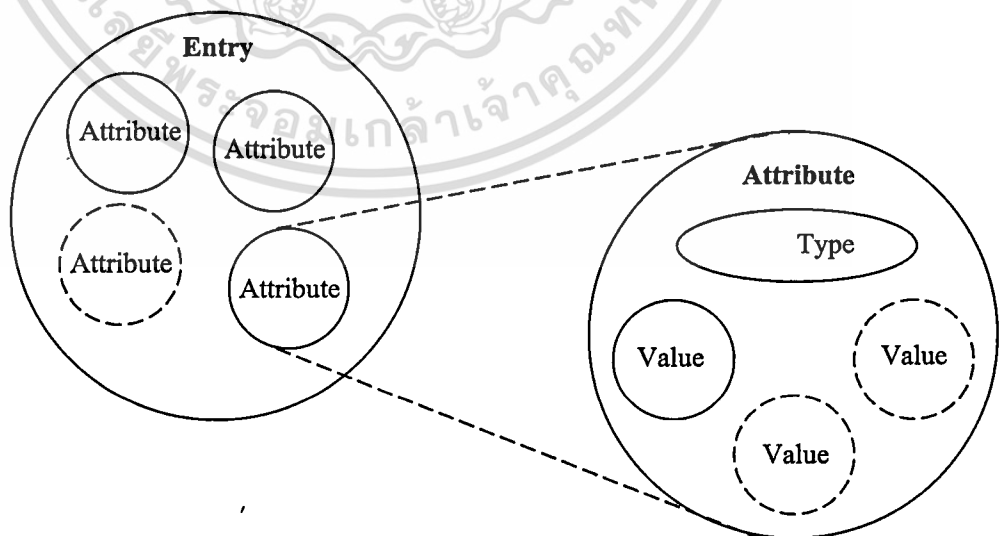
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบทั้ง 4 รูปแบบนั้นจะสามารถอธิบายถึงรูปแบบการทำงานของโปรโตคอล LDAP โดยแยกเป็นส่วนต่างๆ เพื่อให้่ายในการอธิบาย ซึ่งรายละเอียดในแต่ละส่วนจะมีรายละเอียดดังนี้

2.5.1 แบบจำลองข้อมูล (Information model) (Weltman, 2003)

หน่วยย่อยที่สุดของข้อมูลที่ถูกเก็บไว้ในไดเรกทอรีจะถูกเรียกว่า เอนทรี โดยเอนทรีนั้นจะอธิบายถึงกลุ่มของอ็อบเจกต์ที่เราสนใจ ตัวอย่างเช่น บุคคล องค์กร เป็นต้น แลเอนทรีจะประกอบไปด้วยกลุ่มของแอตทริบิวต์ (Attribute) ที่เก็บรายละเอียดของอ็อบเจกต์นั้นไว้ โดยที่ทุกๆแอตทริบิวต์นั้นจะมีประเภท (Type) และค่า (Value) ตั้งแต่ 1 ค่าขึ้นไป ซึ่งประเภทของแอตทริบิวต์นั้นจะกำหนดชนิดของข้อมูลที่สามารถเก็บในแอตทริบิวต์ด้วยรูปแบบโครงสร้างประโยคหรือไวยากรณ์ (Syntax)

ตัวอย่างเช่น เอนทรีของไดเรกทอรีบุคลากรในองค์กร อาจมีการเก็บแอตทริบิวต์หมายเลขโทรสารไว้ แต่บางคนอาจมีหลายหมายเลขก็ได้ ดังนั้น ในแอตทริบิวต์หมายเลขโทรสารอาจมีหลายค่าก็ได้ และประเภทของแอตทริบิวต์นั้นจะถูกกำหนดโดยรูปแบบโครงสร้าง ตัวอย่างเช่น ข้อมูลในแอตทริบิวต์หมายเลขโทรสาร มีการเก็บข้อมูลประเภทหมายเลขโทรศัพท์ เป็นต้น ความสัมพันธ์ของเอนทรีกับแอตทริบิวต์และค่าในแอตทริบิวต์ สามารถแสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 การโต้ตอบกันของโปรแกรมประยุกต์กับผู้ใช้บริการไดเรกทอรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นได้ว่า ในแต่ละเอนทรีสามารถมีได้หลายแอตทริบิวต์ และในแต่ละแอตทริบิวต์นั้น จะประกอบไปด้วยประเภทของข้อมูลและค่า ซึ่งในแต่ละแอตทริบิวต์นั้นอาจจะมีตั้งแต่ 1 ค่าขึ้นไป ซึ่งรูปแบบโครงสร้างข้อมูลของค่าที่ใช้ในแอตทริบิวต์ในโพรโทคอล LDAP สามารถแสดงได้ดังตารางที่ 2.1 ดังนี้

ตารางที่ 2.1 รูปแบบของข้อมูลในแอตทริบิวต์ (R.Colley. 1997)

ไวยากรณ์	รายละเอียด
bin	ข้อมูลไบนารี
ces	อักขระที่มีความหมาย
cis	อักขระที่ไม่มี ความหมาย
tel	หมายเลขโทรศัพท์
dn	ชื่อที่บ่งบอกลักษณะเฉพาะ
Generalized Time	วัน เดือน ปี.
Postal Address	ที่อยู่ที่จะกันแต่ละบรรทัดด้วยอักขระ “\$”

บางแอตทริบิวต์อาจจะสามารถกำหนดสมนาม (Alias) ได้ ตัวอย่างเช่น cn สามารถใช้เป็นชื่อแทนแอตทริบิวต์ CommonName ได้ ตัวอย่างการประกาศสามารถแสดงได้ดังตารางที่ 2.2 ดังนี้

ตารางที่ 2.2 การประกาศประเภทของข้อมูลในแอตทริบิวต์ (R.Colley. 1997)

แอตทริบิวต์,สมนาม	ไวยากรณ์	คำอธิบาย	ตัวอย่าง
CommonName,cn	cis	ชื่อของเอนทรี	John Smith
Surname,sn	cis	นามสกุล	Smith
TelephoneNumber	tel	หมายเลขโทรศัพท์	512-838-6008
owner	dn	ชื่อที่บ่งบอกลักษณะเฉพาะ	cn = John Smith o = IBM
Organization,o	cis	ชื่อองค์กร	IBM
JpegPhoto	bin	ไฟล์รูปถ่าย	รูปของ John

โครงสร้างของอ็อบเจกต์ หรือเค้าร่าง จะทำหน้าที่รวบรวมแอตทริบิวต์ที่เกี่ยวข้องและจำเป็นของแต่ละเอนทรี ตัวอย่างเช่น โครงสร้างของเอนทรี Person จะต้องการแอตทริบิวต์

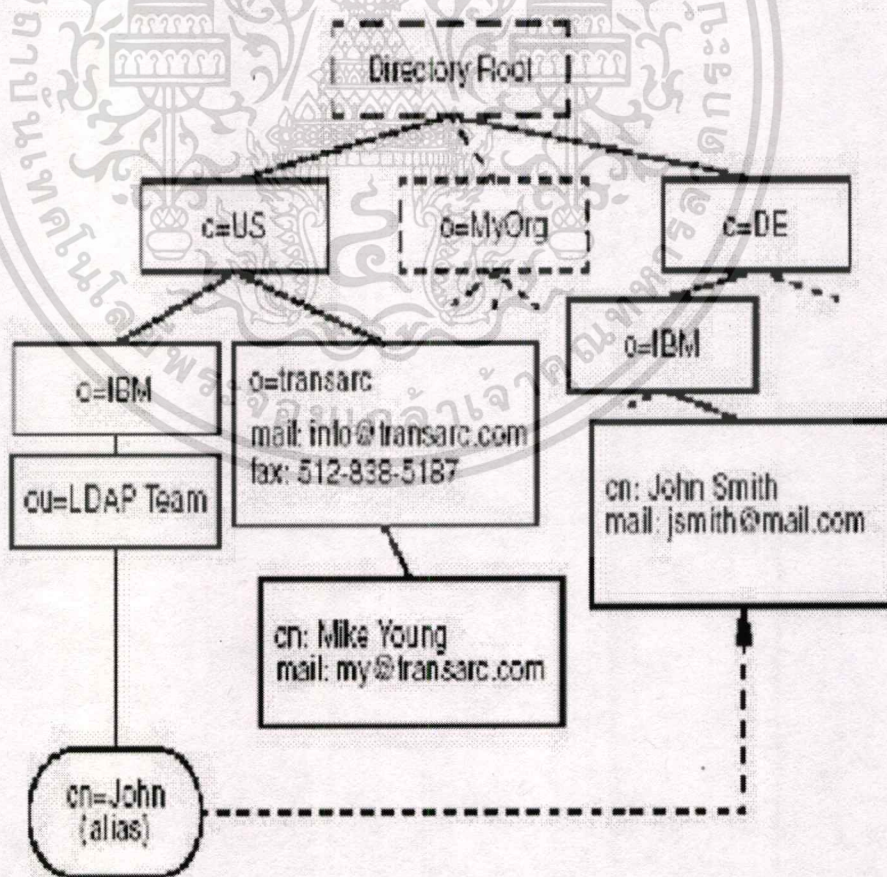
Surname (sn) และแอตทริบิวต์ firstname (fn) เป็นต้น โดยแอตทริบิวต์สามารถถูกเลือกมาใช้เมื่อไร

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็ได้ตามที่ต้องการ โดยโครงสร้างสามารถใช้คุณสมบัติการถ่ายทอด (Inheritance) และคลาสย่อย (Subclass) ของอ็อบเจกต์ได้ และสามารถแสดงโครงสร้างของฐานข้อมูลโดย DIT (Directory Information Tree)

2.5.2 แบบจำลองการตั้งชื่อ (Naming Model)

LDAP จะทำการจัดการตัวบ่งชี้โดยระบุเอนทรีและรวบรวมเอนทรีเป็นโครงสร้างต้นไม้ เรียกว่า Directory Information Tree (DIT) ทุกๆเอนทรีจะถูกจัดเรียงในโครงสร้าง DIT โดยจัดเรียงตามชื่อเฉพาะ (Distinguish Name, dn) โดย dn นั้นจะเป็นชื่อที่มีค่าเฉพาะ (Unique) และไม่มีความคลุมเครือ (Unambiguous) ในการระบุเอนทรีตัวนั้นๆ โดย dn นั้นจะถูกนำมาสร้างเป็น Relative Distinguish Name (RDN) ซึ่ง RDN จะมีรูปแบบในการเขียนดังนี้ <ชื่อแอดทริบิวต์> = <ค่า> นอกจากนี้ในโครงสร้าง DIT ยังสามารถใช้การอ้างอิงข้อมูลโดยอาศัยสมนาม ในการอ้างอิงข้อมูลต่างเอนทรี จากรูปที่ 2.5 จะแสดงถึงรายละเอียดของ Directory Information Tree และความสัมพันธ์ระหว่าง DN และ RDN พร้อมทั้งแสดงการอ้างอิงเอนทรี



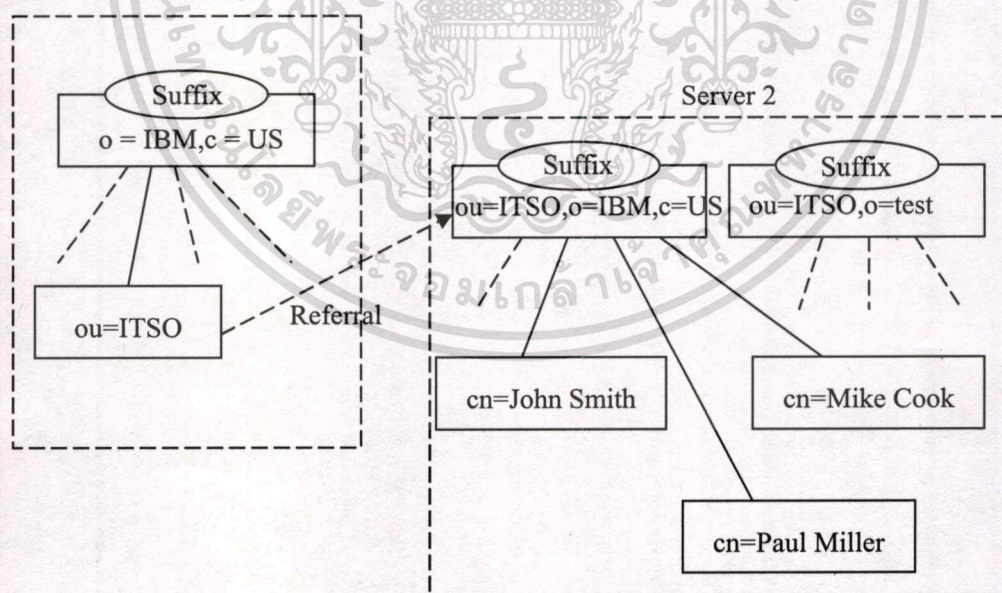
รูปที่ 2.5 รายละเอียดต่างๆ ของ Directory Information Tree

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.5 ช่องสี่เหลี่ยมจะแสดงถึงเอ็นทรีในไคเร็คทอรี และแต่ละแอดทรีบิวต์จะถูกแสดงในแต่ละเอ็นทรี บางแอดทรีบิวต์อาจจะมีการแสดงค่าเป็นสมนาม เพื่อไปอ้างอิงกับค่าในแอดทรีบิวต์อื่นในอีกเอ็นทรีหนึ่ง ตัวอย่างเช่น เอ็นทรีของประเทศ DE (c=DE) นั้นจะอ้างถึงแอดทรีบิวต์ description ที่มีค่าเป็น Germany ในอีกเอ็นทรีหนึ่ง อีกตัวอย่างหนึ่งของเอ็นทรี Name (cn=John) นั้นค่า John นั้นไม่ได้เป็นค่าจริงๆ ของแอดทรีบิวต์ commonName แต่เป็นชื่อย่อเพื่อที่จะอ้างถึงค่าในอีกเอ็นทรีหนึ่งเพื่อที่จะดึงเอาค่าที่สมบูรณ์มาใช้

ในการอ้างถึงเอ็นทรีที่ต้องการนั้น ต้องทำความเข้าใจกับ Suffix โดยเอ็นทรีที่เป็น Suffix ของอีกเอ็นทรีหนึ่งนั้น เอ็นทรีนั้นต้องเป็นเอ็นทรีที่ถ่ายทอดคุณสมบัติมา หรือเรียกว่าเป็นเอ็นทรีที่เป็น Ancestors

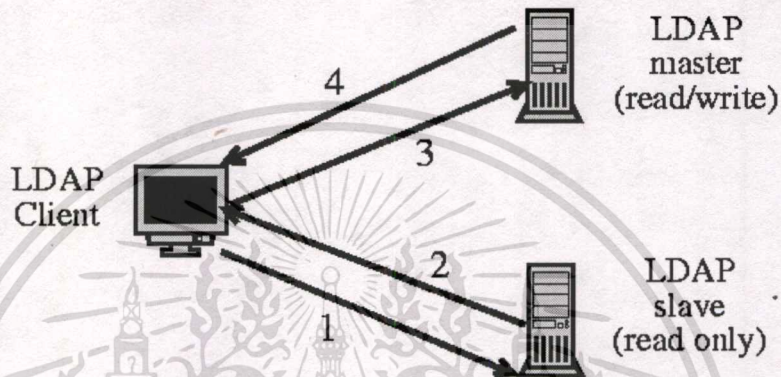
นอกจากนี้ ฐานข้อมูลแบบไคเร็คทอรียังมีการอ้างอิงถึงฐานข้อมูลไคเร็คทอรีที่อยู่ในเครื่องเครื่องให้บริการเครื่องอื่น โดยยึดหลักการอ้างอิง เอ็นทรีที่เป็นพ่อแม่ หรือเอ็นทรีที่อยู่เหนือขึ้นไป ในการค้นหาไคเร็คทอรีที่จะอ้างอิง ไปถึงนั้น เครื่องเครื่องให้บริการนั้นต้องทราบที่อยู่ของเครื่องให้บริการไคเร็คทอรีที่เก็บไคเร็คทอรีที่อ้างอิงถึงนั้น ซึ่งค่านั้นจะเรียกว่า LDAP URLs



รูปที่ 2.6 การอ้างถึงเอ็นทรีข้ามเครื่องให้บริการ

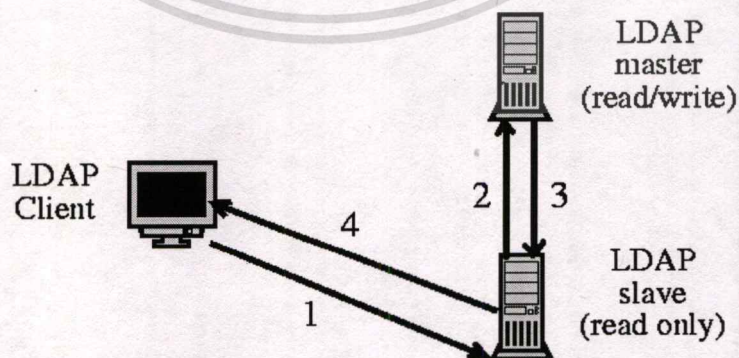
จากรูปที่ 2.6 จะเห็นได้ว่าในการที่จะค้นหารายละเอียดของ OrganizationUnitName(ou) ที่ ITSO นั้นต้องอ้างถึงไคเร็คทอรีในเครื่องให้บริการอีกเครื่องหนึ่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการอ้างอิงถึงข้อมูลเอ็นทรีในเครื่องให้บริการไคลเอนต์หรืออีกเครื่องหนึ่งจะสามารถทำได้สองแบบ แบบแรกคือให้เครื่องให้บริการไคลเอนต์ เครื่องแรกส่งค่า LDAP URLs ไปให้กับเครื่องรับบริการ จัดการเข้าถึงเครื่องให้บริการไคลเอนต์ที่อ้างอิงเอง วิธีนี้จะทำให้เครื่องให้บริการไคลเอนต์นั้นไม่ต้องทำงานหนักในการทำการติดต่อเครื่องให้บริการไคลเอนต์ให้กับเครื่องรับบริการ



รูปที่ 2.7 การทำงานของเครื่องให้บริการไคลเอนต์โดยให้เครื่องรับบริการทำงาน

จากรูปที่ 2.7 เมื่อเครื่องรับบริการได้ส่งข้อความร้องขอไปยังเครื่องให้บริการไคลเอนต์แล้วเครื่องรับบริการได้รับข้อความตอบกลับว่าข้อมูลที่ต้องการมีการอ้างอิงถึงเครื่องให้บริการไคลเอนต์หรืออีกเครื่อง เครื่องรับบริการจะทำการส่งข้อความติดต่อกับเครื่องให้บริการไคลเอนต์ที่อ้างอิง เพื่อร้องขอข้อมูล จากนั้นเครื่องให้บริการไคลเอนต์ที่อ้างอิงจะจัดหาข้อมูลที่เครื่องรับบริการต้องการให้กับเครื่องรับบริการอีกเครื่อง



รูปที่ 2.8 การทำงานของเครื่องให้บริการไคลเอนต์ในการอ้างอิงเครื่องให้บริการไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.8 เมื่อเครื่องให้บริการได้ส่งข้อความร้องขอไปยังเครื่องให้บริการไคลเอนต์แล้วเครื่องให้บริการไคลเอนต์จะทำการค้นหาข้อมูลที่เกี่ยวข้องไปยังอีกเครื่องให้บริการไคลเอนต์ด้วยตัวเอง เมื่อได้ข้อมูลครบถ้วนแล้วจึงทำการส่งให้กับเครื่องรับบริการที่เดียว

ในแต่ละเครื่องให้บริการไคลเอนต์จะทำการสร้างข้อมูลเกี่ยวกับรายละเอียดตัวเองเป็นเอนทรีชนิดพิเศษ ซึ่งเอนทรีนั้นจะมีแอตทริบิวต์ที่อธิบายรายละเอียดต่างๆของเครื่องให้บริการ ตัวอย่างเช่น รายชื่ออ็อบเจกต์ และแอตทริบิวต์ที่เก็บในไคลเอนต์ เวอร์ชันของ LDAP สนับสนุน เป็นต้น

2.5.3 แบบจำลองการทำงาน (Function Model)

โพรโทคอล LDAP ได้กำหนดกระบวนการการทำงานในการเข้าถึง และเปลี่ยนแปลงแก้ไขเอนทรีในไคลเอนต์ ซึ่งกระบวนการทำงานเหล่านี้จะไม่ขึ้นอยู่กับภาษาที่เขียน หรือโปรแกรมประยุกต์ ประเภทของกระบวนการทำงานสามารถแบ่งออกได้เป็น รูปแบบใหญ่ 3 ประเภทดังนี้

- Query การทำงานประเภทค้นหาและเปรียบเทียบสำหรับดึงข้อมูลออกมาจากไคลเอนต์
- Update การทำงานประเภทเพิ่ม ลบ และแก้ไข การทำงานประเภทนี้จะทำการแก้ไขข้อมูล และเก็บข้อมูลที่แก้ไขไว้ในไคลเอนต์
- Authentication การทำงานประเภทสร้างการเชื่อมต่อ และยกเลิกการติดต่อกับไคลเอนต์เครื่องให้บริการ รวมทั้งรักษาสภาพความถูกต้องของข้อมูลในไคลเอนต์อีกด้วย

กระบวนการทำงานทั้ง 3 รูปแบบ นั้นเป็นลักษณะการทำงานหลักๆที่ฐานข้อมูลทั่วไปสามารถทำได้ เราสามารถอธิบายรูปแบบการทำงานต่างๆ ตามมาตรฐานโพรโทคอล LDAP ได้ดังนี้

2.6.3.1 การค้นหา

กระบวนการในการค้นหานั้นเครื่องให้บริการไคลเอนต์ยินยอมให้เครื่องรับบริการทำการร้องขอเพื่อที่จะทำการค้นหาข้อมูลในบางส่วนของ DIT เพื่อที่จะหาข้อมูลที่ต้องการ และส่งผลการค้นหากลับมา โดยการค้นหามีการทำงานพิเศษที่สามารถกำหนดได้ ตัวอย่างเช่น สามารถกำหนดตำแหน่งเริ่มต้นที่ค้นหา ความลึกของ DIT ที่ค้นหา และเมื่อพบเอนทรีที่ต้องการก็สามารถส่งค่าแอตทริบิวต์ของเอนทรีนั้นกลับไปได้ โดยการค้นหาจะมีพารามิเตอร์ต่างๆ ดังนี้

- Base เป็นพารามิเตอร์ที่บอกจุดเริ่มต้นในการค้นหาข้อมูลใน DIT
- Scope เป็นพารามิเตอร์ที่บอกความลึกของ DIT ในการค้นหาโดยอาจจะกำหนดให้ค้นหาเฉพาะเอนทรีลูกก็ได้

- Search Filter เป็นพารามิเตอร์ที่ใช้ในการเปรียบเทียบค่าของแอตทริบิวต์ กับแอตทริบิวต์ของเอนทรีที่ค้นหา โดยการเปรียบเทียบนั้นอาจเปรียบเทียบค่าได้หลายแบบ เช่น เท่ากับ มากกว่า น้อยกว่า เป็นต้น
- Attribute to Return เป็นพารามิเตอร์ที่จะบอกว่าดึงเอาค่าแอตทริบิวต์ไหนบ้างจากเอนทรี เมื่อค้นหาเจอเอนทรีที่ต้องการ เพราะบางทีเอนทรีหนึ่งอาจมีหลายแอตทริบิวต์ แต่แอตทริบิวต์ที่ต้องการใช้จริงๆ นั้นจะมีอยู่ไม่กี่แอตทริบิวต์ ดังนั้นจึงต้องกำหนดแอตทริบิวต์ที่ต้องการในพารามิเตอร์นี้
- Alias Dereferencing เป็นพารามิเตอร์ที่กำหนดความต้องการในการค้นหาค่าที่อ้างถึง หรือไม่เมื่อเอนทรีที่ต้องการมีการใช้ สมนาม แทน พารามิเตอร์นี้จะกำหนดว่าจะให้ไปทำการค้นหาค่าแอตทริบิวต์จากเอนทรีที่แท้จริง หรือนำเอาค่าแอตทริบิวต์จากเอนทรีที่ใช้สมนามเลย
- Limits เป็นพารามิเตอร์ที่กำหนดปริมาณของเอนทรี หรือระยะเวลาที่ค้นหา ในการค้นหาเอนทรีที่ต้องการนั้น อาจมีการค้นหาเจอเอนทรีที่ตรงกับค่าที่ต้องการมาก เพื่อป้องกันการค้นหามากเกินไป จึงมีการกำหนดพารามิเตอร์นี้ โดยสามารถกำหนดจำนวนเอนทรีที่ส่งค่ากลับ หรือกำหนดเวลาที่ใช้ในการค้นหา

เมื่อทำการค้นหาสามารถกำหนดค่าพารามิเตอร์ต่างๆ ข้างต้นเพื่อให้สามารถค้นหาเป็นกรณีพิเศษได้ ทำให้อำนวยความสะดวกแก่ผู้ใช้ในการค้นหาเอนทรีที่ต้องการ

2.6.3.2 การอ้างอิง

ถ้าใครเรียกเครื่องให้บริการนั้นไม่มีอบเจกต์ที่ต้องการ ใครเรียกเครื่องให้บริการนั้นจะส่งค่าอ้างอิงถึงใคร่คทอริเครื่องให้บริการที่น่าจะมีเอนทรีที่ต้องการเหมือนเป็นการบ่งชี้อ้างอิงให้เครื่องเครื่องให้บริการทำการค้นหาข้อมูลที่ต้องการจากเครื่องใคร่คทอริเครื่องให้บริการนั้นๆ

ตัวอย่างเช่น ต้องการหาข้อมูลของคนชื่อ Smith เมื่อใคร่คทอริเครื่องให้บริการ ที่เครื่องให้บริการส่งข้อความร้องขอข้อมูล แต่ใคร่คทอริเครื่องให้บริการไม่มีข้อมูล มันจะทำการส่งข้อความบอกให้ไปหาข้อมูลนั้นจากเครื่องใคร่คทอริเครื่องให้บริการ ที่คิดว่าจะมีข้อมูล แต่ก็ไม่รับประกันว่าใคร่คทอริเครื่องให้บริการ ที่ไปหานั้นจะมีข้อมูลที่ต้องการหรือไม่

2.6.3.3 การค้นหาแบบมีเงื่อนไข

ในการค้นหาอาจจะมีการกำหนดเงื่อนไขในการค้นหา ในการค้นหาแบบมีเงื่อนไขนี้จะส่งค่าของเอนทรีที่ตรงเงื่อนไขที่ต้องการกลับ โดยรูปแบบในการอ้างเงื่อนไขจะมีรูปแบบ

<attribute><operator><value>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ในการค้นหาบุคคลชื่อ John Smith เงื่อนไขในการค้นหาจะเป็น $cn = \text{John Smith}$ ในกรณีนี้ cn เป็นแอตทริบิวต์ = เป็นเงื่อนไขในการค้นหา และ John Smith เป็นค่า ผลที่ได้จากการค้นหาจะเป็นเอนทรีที่มี $cn = \text{John Smith}$

ตารางที่ 2.3 เงื่อนไขต่างๆที่สามารถกำหนดได้ในการค้นหา

เงื่อนไข	รายละเอียด	ตัวอย่าง
=	จะส่งค่าเอนทรีที่มีแอตทริบิวต์ ที่มีค่าเท่ากับค่าที่กำหนด	$cn = \text{John Smith}$ จะค้นหาเอนทรีที่มี $cn = \text{John Smith}$
>=	จะส่งค่าเอนทรีที่มีแอตทริบิวต์ ที่มีค่ามากกว่าหรือเท่ากับค่าที่กำหนด	$sn \geq \text{Smith}$ จะค้นหาเอนทรีที่มีค่าตั้งแต่ Smith ถึง z^*
<=	จะส่งค่าเอนทรีที่มีแอตทริบิวต์ ที่มีค่าน้อยกว่าหรือเท่ากับค่าที่กำหนด	$sn \leq \text{Smith}$ จะค้นหาเอนทรีที่มีค่า ตั้งแต่ a^* ถึง Smith
=*	จะส่งค่าเอนทรีที่มีแอตทริบิวต์ ที่มีค่าในแอตทริบิวต์ที่กำหนด	$sn = *$ จะค้นหาเอนทรีที่มีแอตทริบิวต์ sn
~=	จะส่งค่าเอนทรีที่มีแอตทริบิวต์ ที่มีค่าใกล้เคียงกับค่าที่กำหนด	$sn \sim \text{Smith}$ จะค้นหาเอนทรีที่มีค่าแอตทริบิวต์ sn ใกล้เคียง Smith

จะเห็นได้ว่าการนำอักขระ “*” มาใช้แทนอักขระตัวใดก็ได้ตัวอย่างเช่น $cn = J*Smi*$ จะตรงกับ John Smith และ Jan Smitty เป็นต้น นอกจากนี้ยังมีการเอาเงื่อนไขประเภทบูลมาประกอบเงื่อนไขปรกติ เพื่อให้สามารถทำการค้นหาข้อมูลได้ง่ายขึ้น

ตารางที่ 2.4 เงื่อนไขประเภทบูล

เงื่อนไข	รายละเอียด
&	จะส่งค่าเอนทรีที่ตรงกับเงื่อนไขทั้งหมดกลับ
	จะส่งค่าเอนทรีที่ตรงกับเงื่อนไขใดเงื่อนไขหนึ่งกลับ
!	จะส่งค่าเอนทรีที่ไม่ตรงกับเงื่อนไขกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3.4 การเปรียบเทียบ

ในการทำการกำหนดเงื่อนไขในการเปรียบเทียบในการค้นหา จะเห็นได้ว่าการเปรียบเทียบแบบมีเงื่อนไข นั้นจะทำการเปรียบเทียบจากค่าของแอตทริบิวต์ แต่เมื่อเอนทรีที่ทำการเปรียบเทียบนั้นไม่มีแอตทริบิวต์ที่เปรียบเทียบ มันจะทำการส่งค่ากลับมาว่าไม่พบ แต่การเปรียบเทียบแบบ Compare จะทำการเปรียบเทียบ ถ้าค่าในแอตทริบิวต์นั้นตรงกันจะทำการส่งค่าเป็น จริง กลับมา แต่ ถ้าค่าของแอตทริบิวต์นั้นไม่เท่ากัน หรือไม่มีแอตทริบิวต์ที่เปรียบเทียบมันจะทำการส่งค่าที่เป็นเท็จกลับมา

2.6.3.5 การพิสูจน์ตัวจริง

การพิสูจน์ตัวจริงเป็นกระบวนการในการสร้าง และจบการติดต่อระหว่าง LDAP Client กับ LDAP Server โดยอาศัยโปรโตคอล LDAP ในการติดต่อ ในกระบวนการนี้จะมีการให้เครื่องเครื่องให้บริการ ระบุสิทธิ โดยอาจจะใช้รหัสในการระบุผู้ใช้ โดยจะมีการเข้ารหัสข้อมูลกับข้อความที่ส่งติดต่อกันระหว่างที่ติดต่อกัน เพื่อเป็นการรักษาความปลอดภัยของข้อมูล การติดต่อจะมีกระบวนการที่สำคัญ 3 อย่างคือ

- Bind สร้างการติดต่อระหว่างเครื่องเครื่องให้บริการกับเครื่องเครื่องให้บริการ โดยจะมีการระบุผู้ใช้โดยใช้รหัสผ่านเพื่อระบุผู้ใช้ในกระบวนการนี้ด้วย
- Unbind ขกเลิกการติดต่อระหว่างเครื่องเครื่องให้บริการกับเครื่องเครื่องให้บริการ
- Abandon อนุญาตให้เครื่องเครื่องให้บริการละทิ้งกระบวนการที่ไคลเอนทรีร้องขอแล้วยังทำไม่เสร็จ

2.6.4 แบบจำลองความมั่นคง (Security Model)

จากแบบจำลองการทำงานนั้นจะเห็นได้ว่ามีการรักษาความมั่นคงของข้อมูล ในขั้นตอนในการสร้างการยึดเหนี่ยว (Binding) โดยอาจจะมีการกำหนดสิทธิในการเข้าถึงเอนทรีได้ ลักษณะการรักษาความมั่นคงจะแบ่งหน้าที่ออกเป็น 4 ลักษณะดังนี้

- การพิสูจน์ตัวจริง (Authentication) การรับรองสิทธิต่างๆ ที่ผู้ใช้พึงมี รับผิดชอบ สิทธิในการจัดการข้อมูลในไดเรกทอรี
- บุรณภาพ (Integrity) รับผิดชอบต่อความถูกต้องของข้อมูล ว่าข้อมูลที่ส่งกับที่รับเป็นข้อมูลที่ไม่มี ความผิดพลาดเกิดขึ้น
- การรักษาความลับ (Confidentiality) การรับประกันว่าข้อมูลจะไม่มี การรั่วไหลไปยังบุคคลภายนอกที่ยังไม่ยินยอมให้ใช้ข้อมูล โดยใช้การเข้ารหัสลับข้อมูลในการป้องกัน

- การให้อำนาจ (Authorization) กำหนดสิทธิ์ต่างๆ ในการจัดการข้อมูลให้กับผู้ใช้ในแต่ละราย ว่ามีสิทธิในการเข้าถึงแตกต่างกัน

2.7 ส่วนต่อประสานโปรแกรมประยุกต์ JNDI

LDAP (Lightweight Directory Access Protocol) เป็นโพรโทคอลที่ใช้ในการติดต่อกับผู้ให้บริการไดเรกทอรีในระบบเครือข่าย การให้บริการไดเรกทอรีเป็นสิ่งที่จำเป็นในระบบเครือข่ายอินเทอร์เน็ต โดยผู้ให้บริการไดเรกทอรีจะมีกระบวนการมาตรฐานในการจัดการฐานข้อมูลแบบไดเรกทอรี เช่น การเก็บข้อมูล ค้างค่าข้อมูล และค้นหาข้อมูล มาตรฐานในการจัดการฐานข้อมูลแบบไดเรกทอรีนั้นจะถูกกำหนดมาตรฐานโดยโพรโทคอล LDAP และเพื่อความสะดวกสำหรับการโปรแกรม จึงมีการคิดค้นส่วนติดต่อระหว่างโปรแกรมประยุกต์กับส่วนติดต่อผู้ให้บริการตามมาตรฐานโพรโทคอล LDAP หรือ Application Programming Interface (API) เรียกว่า Java Naming and Directory Interface (JNDI) โดย JNDI จะเป็นผู้เตรียมการจัดการทำงานต่างๆกับผู้ให้บริการไดเรกทอรีทำให้โปรแกรมประยุกต์สามารถติดต่อกับผู้ให้บริการไดเรกทอรีหรือผู้ให้บริการอ้างอิงชื่อได้หลากหลายประเภท โดยใช้ JNDI เพียงตัวเดียวในการเชื่อมต่อ เพราะ JNDI ไม่ขึ้นอยู่กับผู้ให้บริการไดเรกทอรีประเภทใดประเภทหนึ่ง

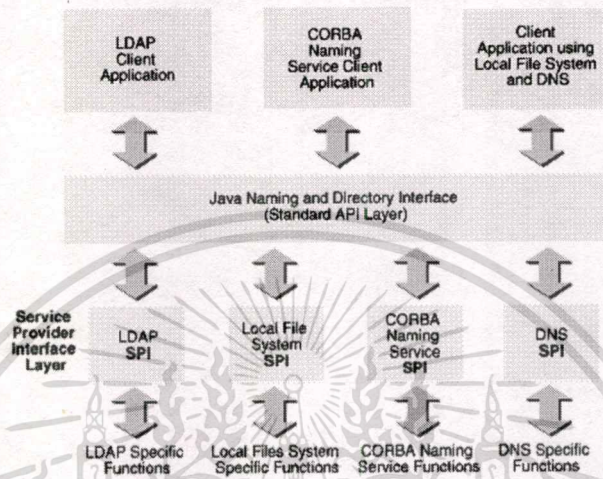
JNDI เป็นส่วนขยายมาจากโครงสร้างภาษาจาวา ซึ่งประกอบไปด้วยกลุ่มของส่วนในการติดต่อกับโปรแกรมประยุกต์ ทำให้ JNDI สามารถใช้รองรับเทคโนโลยีต่างๆของภาษาจาวา ได้ JNDI เป็นเทคโนโลยีที่นำมาใช้กับส่วนติดต่อโปรแกรมประยุกต์ และ ส่วนติดต่อกับผู้ให้บริการ Service-Provider Interface (SPI) ทำให้เทคโนโลยี JNDI นั้นสามารถรองรับกับเทคโนโลยีในการอ้างอิงชื่อแบบต่างๆได้ ในปัจจุบันนี้ JNDI นั้นสามารถใช้ร่วมกับเทคโนโลยีต่างๆตัวอย่างเช่น LDAP, CORBA, RMI, NDS, DNS ซึ่งส่วนใหญ่เป็นเทคโนโลยีในการจัดการการอ้างอิงชื่อและการจัดการไดเรกทอรี

JNDI นั้นถูกใช้ในเรื่องของระบบการอ้างอิงโดยชื่อ ตัวอย่างเช่น ในเรื่องของระบบไฟล์ในการใช้ชื่อไฟล์อ้างอิงไปถึงไฟล์ ในเรื่องของ DNS โดย URL นั้นจะอ้างอิงไปถึงหมายเลข IP และในเรื่องของการให้บริการไดเรกทอรี จะใช้ชื่อของอ็อบเจกต์ในการอ้างอิงไปถึงเอนทรีของอ็อบเจกต์

2.8 สถาปัตยกรรม JNDI

Java Naming and Directory Interface (JNDI) เป็น API ตัวหนึ่งที่น่ามาเป็นส่วนเชื่อมระหว่างโปรแกรมประยุกต์ที่พัฒนาโดยภาษาจาวา กับผู้ให้บริการชนิดต่างๆ โดยจะเรียกส่วนติดต่อกับผู้ให้บริการว่า Service Provider Interface (SPI) JNDI มักจะใช้ติดต่อกับผู้ให้บริการสองประเภท เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ ผู้ให้บริการอ้างอิงชื่อ (Naming Servers) และผู้ให้บริการไดเรกทอรี (Directory Servers) สามารถแสดงรายละเอียดของโครงสร้างในการติดต่อได้ดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างในการเชื่อมการติดต่อของ JNDI (Cooley, 1997: page 560)

จากรูปจะแสดงการเชื่อมต่อระหว่างผู้ให้บริการกับ โปรแกรมประยุกต์ จะเห็นได้ว่า JNDI นั้นสามารถทำการเชื่อมต่อกับผู้ให้บริการได้หลากหลายทำให้สะดวกในการพัฒนาโปรแกรมประยุกต์

JNDI API จะประกอบไปด้วยแพ็คเกจหลัก 4 แพ็คเกจดังนี้ (Weltman, 2003)

- Javax.naming จะประกอบไปด้วยคลาสและส่วนการ ติดต่อสำหรับการเข้าถึงผู้ให้บริการอ้างอิงชื่อ
- Javax.naming.directory ได้เพิ่มขึ้นมาจากแพ็คเกจ Javax.naming และจะมีหน้าที่จัดการการเข้าถึงไดเรกทอรี
- Javax.naming.event ประกอบไปด้วยคลาสและส่วนการติดต่อสำหรับสนับสนุนการ ประกาศข้อความจากผู้ให้บริการอ้างอิงชื่อและไดเรกทอรี
- Javax.naming ldap ประกอบไปด้วยคลาสและส่วนการติดต่อสำหรับสนับสนุน

LDAP v3

2.9 กระบวนการทำงานของ JNDI ที่มาใช้ร่วมกับโปรโตคอล LDAP (Weltman, 2003)

ผู้พัฒนาโปรแกรมประยุกต์ที่เขียนด้วยภาษาจาวา นั้นจะใช้ JNDI ในการติดต่อกับไคลเอนต์ทอริและมาตรฐานในการจัดการไคลเอนต์ทอรินั้น ได้ถูกกำหนดโดยโปรโตคอล LDAP ดังนั้นจึงมีความจำเป็นที่จะต้องทำการพัฒนาให้ JNDI นั้นสามารถจัดการไคลเอนต์ทอริตามมาตรฐานของโปรโตคอล LDAP และสามารถจัดการได้อย่างมีประสิทธิภาพ

ข้อมูลที่ถูกเก็บไว้ในไคลเอนต์ทอรินั้นจะเป็นข้อมูลเกี่ยวกับแอตทริบิวต์และความสัมพันธ์ระหว่างอ็อบเจกต์ได้ถูกกำหนดโดยโครงสร้างของไคลเอนต์ทอริ และแต่ละแอตทริบิวต์สามารถมีค่าได้มากกว่า 1 ค่า แต่ก็อาจจะขึ้นอยู่กับข้อกำหนดโครงสร้างของไคลเอนต์ทอริ ซึ่งจะกำหนดโดยผู้บริหารไคลเอนต์ทอริ ตัวอย่างเช่น ข้อมูลเกี่ยวกับบุคคลอาจจะมีความหมายเลขโทรศัพท์มากกว่า 1 ก็ได้

นอกจากนั้น ยังสามารถนำไปประยุกต์ใช้ในการให้บริการอ้างอิงชื่อ (Naming Service) เป็นการสร้างความสัมพันธ์ระหว่างชื่อกับอ็อบเจกต์ ซึ่งจะทำการค้นหาอ็อบเจกต์จากการอ้างอิงจากชื่อ การให้บริการอ้างอิงชื่อส่วนใหญ่ได้เพิ่มเติมมาจากมาจากการให้บริการไคลเอนต์ทอริ การให้บริการอ้างอิงชื่อจะมองหาอ็อบเจกต์ที่ต้องการจากชื่อ เปรียบเทียบได้กับการให้บริการไคลเอนต์ทอรินั้นจะมองหาอ็อบเจกต์จากแอตทริบิวต์ของอ็อบเจกต์นั้นๆ

JNDI ได้ถูกสร้างมาเป็น API มาตรฐานในการติดต่อกับผู้ให้บริการชื่อและไคลเอนต์ทอริ โดยจะอาศัย Service Provider Interface (SPI) ซึ่งจะทำหน้าที่ในการติดต่อกับผู้ให้บริการชื่อ หรือผู้ให้บริการไคลเอนต์ทอริอีกทีหนึ่ง LDAP เป็นมาตรฐานในการเข้าถึงข้อมูลแบบไคลเอนต์ทอริ JNDI นั้น จะช่วยให้โปรแกรมประยุกต์ที่เขียนด้วยภาษาจาวา สามารถใช้ความสามารถในการใช้บริการไคลเอนต์ทอริได้อย่างมีประสิทธิภาพเหมือนกับที่จัดการไคลเอนต์ทอริด้วยโปรโตคอล LDAP เราสามารถแสดงกระบวนการทำงานของ โปรโตคอล LDAP เปรียบเทียบกับคำสั่งของ JNDI ได้ดังตารางที่ 2.5

ตารางที่ 2.5 การเปรียบเทียบระหว่างการทำงานกับคำสั่งของ JNDI

กระบวนการทำงาน	การทำงาน	คำสั่ง JNDI
Search	ค้นหาเอนทรีที่ต้องการจากไคลเอนต์ทอริ	DirContext.Search()
Compare	เปรียบเทียบเอนทรีในไคลเอนต์ทอริโดยอาศัยแอตทริบิวต์ของเอนทรี	DirContext.Search()
Add	เพิ่มเอนทรีเข้าไปในไคลเอนต์ทอริ	DirContext.bind() DirContext.createSubcontext()
Modify	แก้ไขข้อมูลในไคลเอนต์ทอริ	DirContext.modifyAttributes()
Delete	ลบข้อมูลในไคลเอนต์ทอริ	Context.unbind()

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ห้ามเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 การเปรียบเทียบระหว่างการทำงานกับคำสั่งของ JNDI (ต่อ)

กระบวนการทำงาน	การทำงาน	คำสั่ง JNDI
Rename	เปลี่ยนชื่อในไดเรกทอรี	Context.rename()
Bind	สร้างการเชื่อมต่อกับ LDAP Server	new InitialDirContext()
Unbind	ปิดการเชื่อมต่อกับ LDAP Server	Context.close()
Abandon	ยกเลิกกระบวนการทำงานที่ส่งไปยัง LDAP Server ก่อนหน้านี้	Context.close() NamingEnumeration.close()
Extended	เพิ่มคำสั่งในการจัดการ	LdapContext.extendedOperation()

2.10 การจัดการกับอ็อบเจกต์ที่อยู่ใน LDAP Server (Cooley, 1997)

2.10.1 การติดต่อกับผู้ให้บริการ

ในการติดต่อกับผู้ให้บริการจะต้องมีการอ้างอิงไปยังไดเรกทอรีที่ต้องการ โดยจะสร้าง DirContext เก็บรายละเอียดต่างๆของไดเรกทอรี โปรแกรมประยุกต์ต่างๆจะใช้ฟังก์ชันการทำงาน InitialDirContext และนำ DirContext ไปเก็บในตาราง Hashtable ซึ่งข้อมูลเหล่านั้นจะประกอบไปด้วย ชื่อของผู้ให้บริการ และที่อยู่ของผู้ให้บริการ เป็นต้น

2.10.2 การรับรองสิทธิการเข้าถึงข้อมูลกับผู้ให้บริการ

ในการติดต่อผู้ร้องขอต้องการการรับรองสิทธิในการใช้ข้อมูลกระบวนการนี้เรียกว่า Binding LDAP v3 ใช้ Access Control Lists (ACLs) ในการตัดสินใจว่าผู้ร้องขอมีสติธิในการเข้าถึงข้อมูลในระดับไหนซึ่งจะถูกกำหนดโดย โปรแกรมประยุกต์ LDAP ได้แบ่งระดับในการรักษาความปลอดภัยของข้อมูลออกเป็น 3 ระดับดังนี้

- Simple: รับรองสิทธิอย่างรวดเร็วโดยตรวจสอบจากชื่อผู้ใช้ และรหัสผ่านธรรมดา
- SSL: รับรองสิทธิโดย SSL ที่เข้ารหัสบนเครือข่ายการสื่อสาร
- SASL: ใช้กระบวนการ MD5/Kerberos ในการเข้ารหัสข้อมูลซึ่งเป็นการรักษาความปลอดภัยของข้อมูล

2.10.3 การเพิ่มเอนทรีใน LDAP Server

LDAP Server นั้นสามารถทำงานรองรับอ็อบเจกต์เชิงจาวาได้ และ JNDI นั้นจะมองไดเรกทอรีในเชิงวัตถุ ทำให้สามารถเพิ่มอ็อบเจกต์และดึงค่าอ็อบเจกต์จากไดเรกทอรีได้โดยที่อ็อบเจกต์สามารถถูกเก็บได้ 3 กรณี

- เก็บอ็อบเจกต์เชิงจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เก็บค่าที่อ้างอิงไปยังอ็อบเจกต์นั้นๆ
- เก็บข้อมูลต่างๆในรูปของแอตทริบิวต์

2.10.3.1 การเก็บอ็อบเจกต์เชิงจาวา

การเพิ่มเอนทรีแบบนี้จะทำการเรียกใช้โดยใช้คำสั่ง `Context.bind()` ในการเพิ่มข้อมูลที่เป็นอ็อบเจกต์ลงในข้อมูล และยังสามารถทำการใส่ค่าแอตทริบิวต์ลงไปในตอนที่เราเพิ่มเอนทรีได้อีกด้วย และเมื่อเราทำการ `DirContext.bind()` ระบบจะทำการจัดเรียงตำแหน่งอ็อบเจกต์ลงในหน่วยความจำโดยจะเรียกคลาส `java.io.Serializable` โดยอัตโนมัติดังเช่นตัวอย่าง

```
MyObject obj = new MyObject();
```

```
Ctx.bind("cn=anobject",obj);
```

เมื่อทำการเก็บอ็อบเจกต์แล้วเราสามารถดึงข้อมูลจากการค้นหาจากชื่อของไคลเอนท์

```
MyObject obj = (MyObject)ctx.lookup("cn=anobject");
```

เมื่อโปรแกรมประยุกต์เรียงตำแหน่งและบันทึกลงในหน่วยความจำ มันจะบันทึกข้อมูลที่ระบุตัวบ่งชี้ของอ็อบเจกต์นั้นๆลงไปด้วย และจะเก็บรายละเอียดของคลาสลงในคลาสไฟล์ พร้อมทั้งจัดตำแหน่งอ็อบเจกต์ใหม่โดยจะตัดสินใจว่าจะจัดตำแหน่งอย่างไร และจะทำการดึงคลาสที่จำเป็นมาเก็บอีกด้วย

นอกจากนั้น โปรแกรมประยุกต์สามารถเรียงตำแหน่งของอ็อบเจกต์ซึ่งกำหนดโดยผู้ใช้ระบุค่าลงไปแอตทริบิวต์พื้นฐานที่ชื่อว่า `Javacodebase` เพื่อที่จะระบุโครงสร้างของการจัดตำแหน่งอ็อบเจกต์ที่เพิ่มขึ้นมา และ LDAP จะทำการตรวจสอบโครงสร้างของไคลเอนท์และทำการเพิ่มอ็อบเจกต์ลงในไคลเอนท์เมื่อโครงสร้างที่กำหนดมาสอดคล้องกับโครงสร้างที่มีอยู่แล้วดังเช่นตัวอย่าง

```
MyObject obj = New MyObject();
```

```
BasicAttribytes battr = new
```

```
BasicAttribytes("javacodebase", ".http://myserver.com/classes" )
```

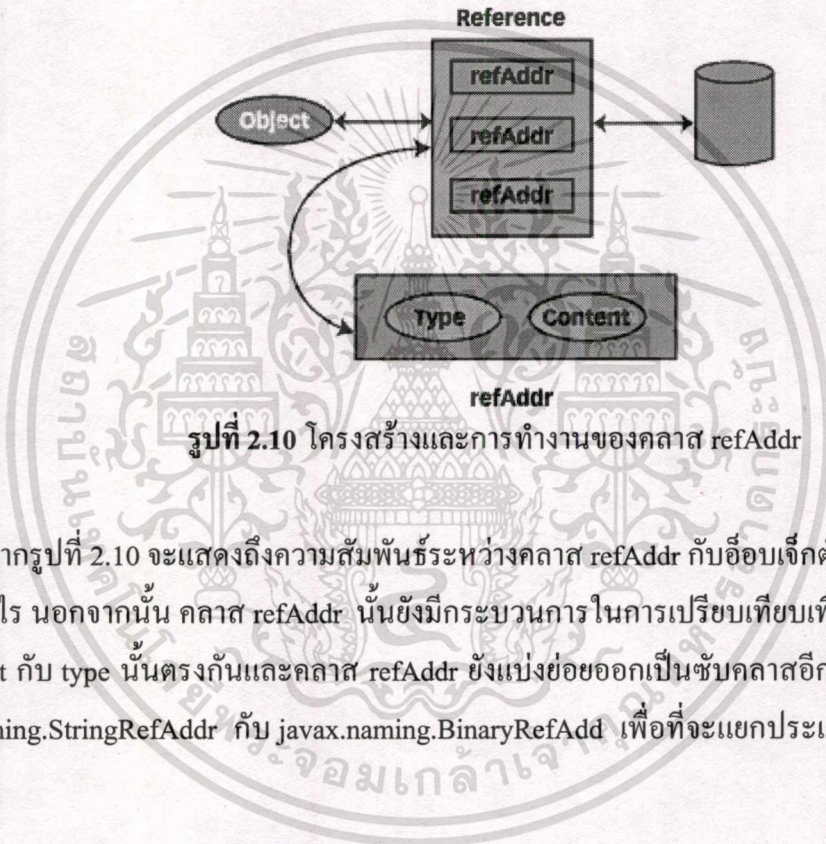
```
Ctx.bind ("cn=anobject",obj,battr);
```

2.10.3.2 การเก็บค่าอ้างอิงไปยังอ็อบเจกต์นั้นๆ

นอกจากนั้น LDAP ยังสามารถเก็บข้อมูลที่อ้างอิงไปยังอ็อบเจกต์โดยอาศัยคลาส `javax.naming.RefAddr` ในการเก็บค่าที่อ้างอิงไปยังอ็อบเจกต์โดยข้อมูลที่เก็บจะประกอบไปด้วย

- ชื่อของคลาสที่อ้างอิงไปยังอ็อบเจกต์
- ตำแหน่งของคลาส `RefAddr` ที่แสดงที่อยู่ของอ็อบเจกต์
- ชื่อ และตำแหน่งของอ็อบเจกต์ factory ที่ใช้ระหว่างทำการสร้าง

คลาส Javax.naming.RefAddr จะประกอบไปด้วยข้อมูลที่แสดงทางที่จะสามารถติดต่อกับอ็อบเจกต์ ตัวอย่างเช่น ตำแหน่งกายภาพของหน่วยความจำของข้อมูลบนเครื่องอื่นที่เราติดต่อกับเป็นต้น ในคลาสนี้จะอธิบายถึงความสัมพันธ์ระหว่างเนื้อหาและชนิด โดยเนื้อหาหรืออ็อบเจกต์นั้นจะเก็บข้อมูลที่ต้องการสำหรับการสร้างอ็อบเจกต์ และ type หรือสตริงนั้นจะระบุวัตถุประสงค์ของ content ซึ่งจะสามารถแสดงโครงสร้างได้ดังรูปที่ 2.10



รูปที่ 2.10 โครงสร้างและการทำงานของคลาส refAddr

จากรูปที่ 2.10 จะแสดงถึงความสัมพันธ์ระหว่างคลาส `refAddr` กับอ็อบเจกต์ว่ามีโครงสร้างเป็นอย่างไร นอกจากนั้น คลาส `refAddr` นั้นยังมีกระบวนการในการเปรียบเทียบเพื่อที่จะให้แน่ใจว่า `content` กับ `type` นั้นตรงกันและคลาส `refAddr` ยังแบ่งย่อยออกเป็น subclass อีก 2 subclass คือ `javax.naming.StringRefAddr` กับ `javax.naming.BinaryRefAddr` เพื่อที่จะแยกประเภทของข้อมูลที่ใช้อย่างอิง

2.10.4 การเก็บข้อมูลต่างๆ ในรูปของแอตทริบิวต์

เทคนิคสุดท้ายในการเก็บอ็อบเจกต์โดยจะทำการรวบรวมแอตทริบิวต์ และทำการอ้างอิงไปยังอ็อบเจกต์ เทคนิคนี้จะไม่ได้ทำการเพิ่มอ็อบเจกต์เข้าไปโดยตรง แต่ทำการเพิ่มแอตทริบิวต์เข้าไปในอ็อบเจกต์ทำให้ดูเหมือนกับว่าเพิ่มอ็อบเจกต์เข้าไปในโครงสร้างจริงๆ

2.11 การแก้ไขกับอ็อบเจกต์ที่อยู่ใน เครื่องให้บริการLDAP

เมื่อเราทำการเก็บอ็อบเจกต์ เราสามารถทำการจัดการแอตทริบิวต์ที่มีอยู่ในเอนทรี โดยจะกระทำโดยอาศัยคลาส `modifyAttributes()` ในการจัดการ เพิ่ม แทนที่ และลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแอตทริบิวต์ของอ็อบเจกต์นั้นมีหลายค่า เมื่อค่าเก่ากับค่าใหม่ที่เป็นค่าที่ต้องการจะแก้ไขไม่ได้ถูกส่งมา มันจะทำการลบค่าอื่นๆที่มีอยู่ทิ้ง โดยการเก็บค่าของแอตทริบิวต์ต่างๆจะถูกเก็บในรูปแบบโครงสร้างของแถวลำดับโดยในการจัดการจะจัดการโดยอ้างถึงค่าจากแถวลำดับ

2.12 การลบอ็อบเจกต์จากไคลเรคทอรี

Context ประกอบไปด้วยกลุ่มของชื่อของอ็อบเจกต์ ส่วนซัพคอนเท็กซ์นั้นจะเป็น context ที่อยู่ด้านล่างของ LDAP Tree เราสามารถลบข้อมูลโดยการทำลายซัพคอนเท็กซ์แต่ในความเป็นจริงจะทำขั้นตอนที่ตรงกันข้าม ซึ่งจะทำการสร้างซัพคอนเท็กซ์โดยใช้ `DirContext.createSubcontext()` ในความเป็นจริงแล้วไคลเรคทอรียังไม่ทำการสร้างอ็อบเจกต์ ให้ใช้ `Context.unbind` โดยใช้รายละเอียดข้อมูลที่จะทำการลบ ระบบอ้างอิงชื่อจะทำการลบ Context ที่มีรายละเอียดเหมือนกันซึ่งจะเรียกว่า Foreign Context แต่ถ้าไม่พบอ็อบเจกต์ที่มีค่าตรงกับที่ต้องการลบ JNDI จะทำการส่งค่า `ContextNotEmptyException` ออกมา

2.13 การค้นหาอ็อบเจกต์จากไคลเรคทอรี

LDAP มีลักษณะพิเศษในการค้นหาอย่างรวดเร็ว ทำให้เป็นการง่ายในการใช้ JNDI Naming Service ในการดำเนินการค้นหาชนิดต่างๆดังนี้

- `Context.lookup()` จะส่งค่าอ็อบเจกต์กลับเมื่อพบอ็อบเจกต์ที่มีค่าตรงกับที่ต้องการในไคลเรคทอรี
- `Context.list()` จะส่งค่าของขอบเขต และชื่อของคลาสของอ็อบเจกต์ที่อยู่ในขอบเขตของ context
- `Context.listBindings()` จะส่งค่าของขอบเขต และชื่อของคลาสของอ็อบเจกต์ที่อยู่ในขอบเขตของ context ที่อยู่ในช่วง context

LDAP นั้นจะเก็บข้อมูลในโครงสร้างต้นไม้ และเก็บลักษณะพิเศษในโหนดที่อยู่ลึกลงมา โดยในการค้นหาเราต้องทำการกำหนดโหนดที่เริ่มต้นในการค้นหา และช่วงในการค้นหา ซึ่งจะแบ่งช่วงในการค้นหาออกเป็น 3 ประเภท

- `SUBTREE_SCOPE` จะทำการเริ่มต้นที่เอนทรีที่กำหนดและทำการค้นหาทุกโหนดที่อยู่ต่ำลงไป
- `ONELEVEL_SCOPE` จะทำการเริ่มต้นที่เอนทรีที่กำหนดและทำการค้นหาทุกโหนดที่อยู่ต่ำลงไปหนึ่งลำดับชั้น
- `OBJECT_SCOPE` จะทำการค้นหาเอนทรีที่มีค่าตรงกับที่ต้องการเลยไม่มีการกำหนด

เอนทรีเริ่มต้นในการค้นหา

ในการกำหนดช่วงในการค้นหา นั้น จะถูกกำหนดโดยทำการสร้างคลาส SearchControl เพื่อทำการกำหนดเงื่อนไขต่างๆในการค้นหา นอกจากนั้น รูปแบบในการค้นหายังมีหลายวิธีในการค้นหาซึ่งมีรูปแบบที่นิยมใช้ดังนี้

- การค้นหาอ็อบเจกต์จากแอตทริบิวต์ การค้นหานี้สามารถค้นหาจากค่าของแอตทริบิวต์ซึ่งเป็นรูปแบบที่ง่ายที่สุดในการค้นหา ซึ่งการค้นหาแบบนี้จะต้องการค่าของกลุ่มของแอตทริบิวต์ที่ต้องการค้นหาและชื่อของ context ที่ต้องการจะค้นหาเป็นเงื่อนไขในการค้นหา
- การค้นหาอ็อบเจกต์โดยอาศัยเงื่อนไขทางคณิตศาสตร์ LDAP สามารถทำการค้นหาจากการใช้เงื่อนไขในการค้นหา ชื่อเงื่อนไขที่ใช้ในการค้นหาจะกำหนดเปรียบเทียบระหว่างค่าเริ่มต้นกับค่าที่จะต้องการหา โดยจะใช้เงื่อนไขทางคณิตศาสตร์ในการเปรียบเทียบ เช่น มากกว่า น้อยกว่า เท่ากับ เป็นต้น

นอกจากตัวอย่างข้างต้นนี้แล้ว ยังมีการค้นหาข้อมูลในรูปแบบต่างๆอีก เช่น การค้นหาการอ้างอิงถึงข้อมูลที่ต้องการว่ามีอ้างอิงไปยังผู้ใช้บริการใดเรืกทอดอื่นอีกหรือไม่ โดยค่าที่ส่งกลับนั้นจะประเภทบูล

2.14 นโยบายในการจัดการเพื่อเพิ่มประสิทธิภาพของ JNDI API

เราสามารถกำหนดนโยบายต่างๆในการควบคุมผลที่ได้จากการค้นหาเพื่อที่จะทำให้การค้นหาและผลที่ได้จากการค้นหามีประสิทธิภาพมากยิ่งขึ้น

- การควบคุมแอตทริบิวต์จะเพิ่มประสิทธิภาพในการส่งค่าที่ได้จากการค้นหาโดยจะกำหนดเงื่อนไขของข้อมูลที่จะส่งค่ากลับคืนว่าผู้ใช้ต้องการค่าอะไรบ้าง โดยกำหนดใน setReturningAttribute ที่จะให้ผลที่ได้จากการค้นหาส่งค่าแอตทริบิวต์อะไรบ้าง
- การควบคุมเวลาจะใช้ setTimeLimit ในการกำหนดระยะเวลาในการค้นหา เนื่องจากโปรแกรมประยุกต์อาจจะทำการค้นหาข้อมูลที่ไม่มีเป็นระยะเวลาที่นาน ทำให้เสียเวลาที่ใช้ในการติดต่อกับผู้ให้บริการ
- การคืนค่าผลลัพธ์ในรูปแบบจาวาอ็อบเจกต์ จะเป็นการเพิ่มประสิทธิภาพความเร็วในการส่งผลที่ได้รับจากการค้นหา โดยจะกำหนดค่าของ setReturningObjFlag() ถ้าค่าเป็นจริง จะส่งผลที่ได้จากการค้นหาเป็นรูปแบบของอ็อบเจกต์ ส่วนถ้าค่าเป็นเท็จ จะส่งผลที่ได้กลับมาเป็นเพียงชื่อของอ็อบเจกต์หรือคลาสเท่านั้น ไม่ส่งข้อมูลของอ็อบเจกต์นั้นมาทำให้การส่งผลที่ได้จากการค้นหาเร็วยิ่งขึ้น

- การควบคุมลิงค์ในแต่ละเอนทรีใน context นั้นจะมีการอ้างอิงถึงเอนทรีอื่นๆ โดยมีลักษณะคล้ายคลึงกับระบบจัดการไฟล์ เพื่อเพิ่มประสิทธิภาพจึงมีการกำหนดค่าของ setDerefLingFlag เพื่อที่จะกำหนดว่าจะทำการค้นหาถึงเอนทรีที่อ้างอิงไปหรือไม่ หัวข้อที่กล่าวข้างต้นเป็นเทคนิคที่ผู้พัฒนาโปรแกรมประยุกต์สามารถกำหนดเพื่อที่จะทำให้ผลที่ได้จากการค้นหาตรงกับความต้องการ และเพิ่มประสิทธิภาพในการทำงานของโปรแกรมประยุกต์ที่พัฒนาขึ้น



บทที่ 3

การวิเคราะห์และออกแบบระบบ

ในบทนี้จะกล่าวถึง การศึกษารูปแบบระบบการทำงานของระบบจัดการข้อมูลรายชื่อผู้ใช้ ปัจจุบัน ซึ่งมีความยุ่งยาก และซับซ้อน รวมถึงไม่มีประสิทธิภาพในการให้บริการแก่องค์กร ทั้งยังทำให้องค์กรสามารถจัดการกับข้อมูลรายชื่อผู้ใช้ที่ไม่มีประสิทธิภาพ จึงมีการนำเอาเทคโนโลยีไคลเอนต์เซิร์ฟเวอร์วิสมาใช้ในการจัดการกับข้อมูลรายชื่อผู้ใช้ในองค์กร

ในปัจจุบันระบบบัญชีรายชื่อผู้ใช้ในองค์กรมีการเก็บที่ซ้ำซ้อนในแต่ละแผนก ยกตัวอย่างเช่น บุคลากรในองค์กรอาจมีรายละเอียดส่วนตัวเก็บไว้กับฐานข้อมูลฝ่ายบุคคล แต่ในการใช้งานในแผนกบางแผนก เช่น แผนกการเงิน ก็ต้องอาศัยข้อมูลของบุคลากรเช่นเดียวกัน ทำให้มีการเก็บข้อมูลที่ซ้ำซ้อนกัน และเมื่อข้อมูลบุคลากรในองค์กรมีจำนวนมาก จะทำให้ยากในการค้นหาข้อมูล จึงมีการเก็บฐานข้อมูลผู้ใช้ให้อยู่ในรูปแบบไคลเอนต์เซิร์ฟเวอร์ เนื่องจากสามารถเก็บข้อมูลแบบรวมศูนย์ และยังสามารถให้บริการได้หลายแผนก เนื่องจากมีการกำหนดสิทธิผู้ใช้ทำให้ผู้ใช้ที่ไม่มีสิทธิสามารถดูรายละเอียดสำคัญได้ เช่น แผนกการเงินจะไม่สามารถดูรายละเอียดของบุคลากรได้ทุกอย่าง อาจดูได้แต่ชื่อ และเลขบัญชีของบุคลากรนั้นๆ เป็นต้น นอกจากนี้ ฐานข้อมูลแบบไคลเอนต์เซิร์ฟเวอร์สามารถเข้าถึงโดยผู้ใช้จำนวนมากได้ ทำให้สะดวกในการค้นหาข้อมูล

3.1 ปัญหาและข้อจำกัดของระบบงานเดิม

ในหัวข้อนี้จะกล่าวถึงการศึกษาการทำงานของระบบการจัดการข้อมูลรายชื่อผู้ใช้ในอดีต ปัจจุบันว่ามีปัญหาและข้อจำกัดอย่างไร ซึ่งสามารถจำแนกออกเป็นข้อๆ ได้ดังนี้

1. ข้อมูลรายชื่อผู้ใช้ในแต่ละแผนกในองค์กรนั้น มีการแยกเก็บกระจัดกระจายในแต่ละแผนกเมื่อมีการจัดการกับข้อมูลในแผนกใดแผนกหนึ่ง อาจจะทำให้ข้อมูลอาจมีความขัดแย้งกัน ทำให้ข้อมูลไม่น่าเชื่อถือ

2. ไม่สามารถจัดการเกี่ยวกับสิทธิในการเข้าใช้ได้อย่างมีประสิทธิภาพ เนื่องจากในการกำหนดสิทธิเมื่อข้อมูลไม่ได้จัดเก็บอยู่ในที่เดียวกัน จะต้องดำเนินการร่วมกันในหลายๆแผนก

3. เนื่องจากข้อมูลได้ถูกจัดเก็บอย่างกระจัดกระจาย ทำให้ไม่สะดวกในการกำหนด

เอกสารนี้จัดทำขึ้นเพื่อการศึกษาความปลอดภัยของข้อมูลรายชื่อผู้ใช้ในองค์กร ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ระบบไม่สามารถรองรับการใช้งานที่อาจเพิ่มขึ้นมาในอนาคตได้ เนื่องจากระบบฐานข้อมูลแบบ DBMS นั้นจะมีข้อจำกัดในการเข้าถึงของผู้ใช้จำนวนมาก

3.2 ความต้องการของระบบ

เนื่องจากข้อจำกัดของระบบงานที่ผ่านมา และเมื่อนำเอาเทคโนโลยีของไคลเอนต์เซิร์ฟเวอร์มาประกอบ จึงทำให้สามารถกำหนดความต้องการของผู้ใช้ระบบที่พัฒนาขึ้นเพื่อแก้ปัญหาที่เกิดขึ้นได้ดังนี้

1. สามารถจัดเก็บข้อมูลรายชื่อผู้ใช้ในองค์กรไว้แบบรวมศูนย์ เพื่อสะดวกในการจัดการเกี่ยวกับข้อมูลรายชื่อผู้ใช้ในด้านต่างๆ
2. สามารถกำหนดนโยบายในการรักษาความปลอดภัยของข้อมูลได้สะดวกเนื่องจากข้อมูลรายชื่อผู้ใช้ได้ถูกจัดเก็บไว้ที่เดียวกัน
3. ลดความซ้ำซ้อนของข้อมูล และข้อมูลได้ถูกจัดเก็บโดยมีมาตรฐานเดียวกัน ทำให้ข้อมูลที่จัดเก็บไว้มีความน่าเชื่อถือ

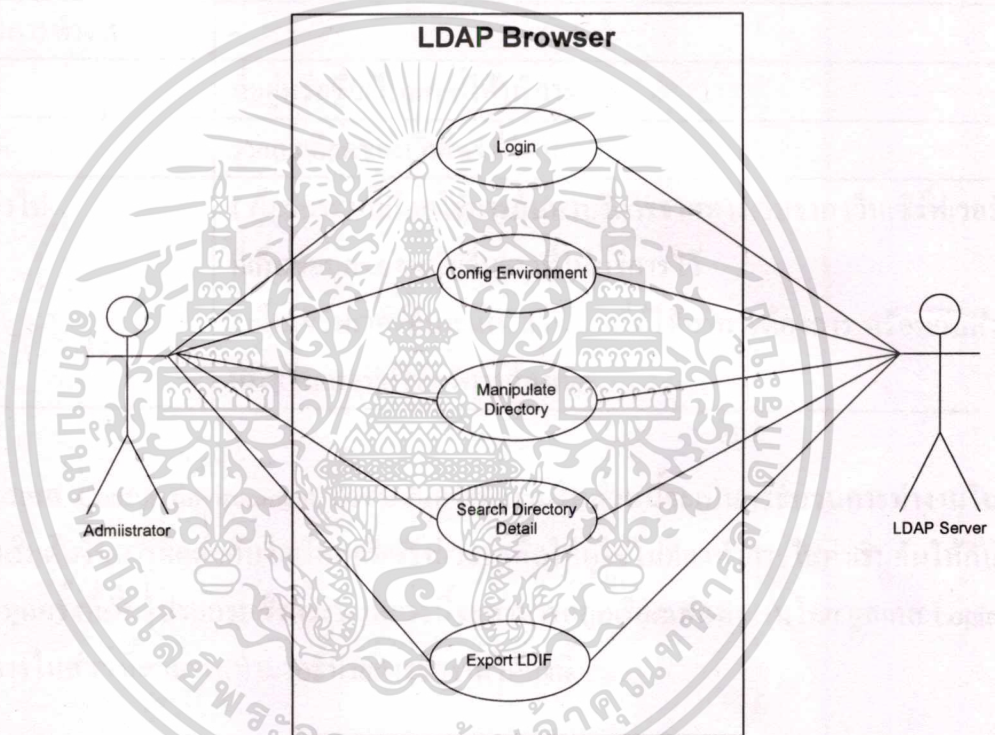
3.3 การออกแบบระบบการทำงานโดยใช้ แผนภาพแสดงการทำงานของระบบ

การออกแบบระบบโดยใช้แผนภาพแสดงการทำงานของระบบ เป็นการแสดงถึงกระบวนการทำงานของระบบ ซึ่งระบบการจัดการข้อมูลรายชื่อผู้ใช้จะมีผู้ที่เกี่ยวข้องดังต่อไปนี้ คือ ผู้ดูแลระบบ (System Administrator) พนักงานในแผนกต่างๆ ในองค์กร (User) และใน ส่วนของ LDAP เซิร์ฟเวอร์ ซึ่งจะมีกระบวนการทำงานต่างๆ ดังต่อไปนี้

1. Login เป็นกระบวนการจัดการเกี่ยวกับรายละเอียดเบื้องต้นในการติดต่อกับเซิร์ฟเวอร์ ขึ้นันตัวเองเพื่อเข้าสู่ระบบ และรับรองสิทธิ์ในการจัดการกับไคลเอนต์
2. Config environment เป็นกระบวนการในการเก็บรายละเอียดเกี่ยวกับค่าเริ่มต้นเพื่อให้ผู้ใช้ไม่ต้องทำการใส่รายละเอียดทุกครั้งที่ต้องการ
3. Connect เป็นกระบวนการในการส่งข้อความติดต่อกับ LDAP Server เพื่อสร้างการติดต่อ
4. Manipulate Data เป็นกระบวนการทำงานในการจัดการกับไคลเอนต์ และรายละเอียดของไคลเอนต์ เช่นการเพิ่มหรือแก้ไขรายละเอียดของไคลเอนต์เป็นต้น
5. Search Directory Detail เป็นกระบวนการทำงานเพื่อค้นหาไคลเอนต์ที่ต้องการจากรายละเอียดเฉพาะของไคลเอนต์ที่ต้องการ
6. Export LDIP เป็นการรวบรวมรายละเอียดของไคลเอนต์เพื่อทำออกมาเป็นข้อมูลตัวอักษร

3.4 แผนภาพแสดงการทำงานของระบบของกระบวนการจัดการข้อมูลรายชื่อผู้ใช้

โดยที่ แผนภาพแสดงการทำงานของระบบ ของกระบวนการจัดการข้อมูลรายชื่อผู้ใช้จะแสดงไว้ดังรูปที่ 3.1 โดยในระบบจะประกอบไปด้วยกระบวนการทำงานต่างๆดังต่อไปนี้



รูปที่ 3.1 แผนภาพการทำงานของระบบจัดการข้อมูลรายชื่อผู้ใช้

จากแผนภาพการทำงานของระบบ สามารถอธิบายหน้าที่และการทำงานของแต่ละกระบวนการได้ดังต่อไปนี้

ยูสเคส Login กระบวนการทำงานนี้จะเป็กระบวนการในการติดต่อเพื่อยืนยันตัวตนของผู้จัดการระบบ นอกจากนั้นในส่วนนี้ยังมีการให้ผู้ใช้ทำการใส่ค่าเริ่มต้นในการติดต่อกับไคลเอนต์เซิร์ฟเวอร์ เช่น หมายเลขพอร์ตเป็นต้น ซึ่งกระบวนการในส่วนนี้ผู้จัดการระบบจะใช้ร่วมกับผู้ใช้รายอื่นๆ ซึ่ง LDAP นั้นจะทำการกำหนดสิทธิ์ให้กับผู้ใช้เอง กระบวนการในส่วนนี้สามารถนำมาอธิบายการทำงาน (Use Case Description) ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 รายละเอียดประกอบการทำงานของ Manipulate Directory (ต่อ)

ยูสเคส	Manipulate Directory
ข้อมูลที่ได้	เว็บเพจที่แสดงข้อมูลที่ผ่านการจัดการแล้ว
ขั้นตอนทั่วไป	<ol style="list-style-type: none"> 1. ผู้ดูแลระบบเรียกใช้การค้นหบริการจากทางเว็บของ Register Server ที่เก็บข้อมูลบริการไว้ 2. ผู้ดูแลระบบทำร้องขอบริการจากไคลเอนต์เซิร์ฟเวอร์ผ่านเว็บเซิร์ฟเวอร์ 3. ไคลเอนต์เซิร์ฟเวอร์ได้ทำงานกับฐานข้อมูลผู้ใช้ตามที่ผู้ดูแลระบบต้องการ
ขั้นตอนทางเลือก	<ol style="list-style-type: none"> 2a. ผู้ดูแลระบบเลือกการทำงานในการเพิ่มเอนทรีในไคลเอนต์เซิร์ฟเวอร์ 2b. ผู้ดูแลระบบเลือกการทำงานในการลบเอนทรีในไคลเอนต์เซิร์ฟเวอร์ 2c. ผู้ดูแลระบบเลือกการทำงานในการแก้ไขรายละเอียดของเอนทรีในไคลเอนต์เซิร์ฟเวอร์ 2d. ระบบทำการค้นหาเอนทรีทั้งหมดในไคลเอนต์เซิร์ฟเวอร์เพื่อนำมาแสดงเป็นโครงสร้างต้นไม้

ยูสเคส Search Directory กระบวนการทำงานนี้จะ เป็นกระบวนการที่ให้ผู้ใช้งานสามารถค้นหาไคลเอนต์เซิร์ฟเวอร์ที่ต้องการจากรายละเอียดของไคลเอนต์เซิร์ฟเวอร์นั้นๆ ซึ่งสามารถกำหนดความลึกในการค้นหาในโครงสร้างนั้นๆ ได้ กระบวนการในส่วนนี้สามารถนำมาอธิบายการทำงานได้ดังนี้

ตารางที่ 3.6 รายละเอียดประกอบการทำงานของ Search Directory

ยูสเคส	Search Directory
วัตถุประสงค์	การค้นหาข้อมูลที่ต้องการในไคลเอนต์เซิร์ฟเวอร์
เงื่อนไขเริ่มต้น	ผู้ดูแลระบบต้องทราบรายละเอียดของข้อมูลที่ต้องการค้นหา
เมื่อทำงานสำเร็จ	แสดงรายละเอียดของข้อมูลที่ทำการค้นหาผ่านทางหน้าเว็บเพจ
เมื่อทำงานไม่สำเร็จ	ผู้ดูแลระบบไม่พบข้อมูลที่ต้องการค้นหา
Actor ที่เกี่ยวข้อง	ผู้ดูแลระบบ, เว็บเซิร์ฟเวอร์, ไคลเอนต์เซิร์ฟเวอร์
สิ่งกระตุ้นการทำงาน	-
ข้อมูลเข้า	รายละเอียดของข้อมูลที่ต้องการค้นหา

ตารางที่ 3.6 รายละเอียดประกอบการทำงานของ Search Directory (ต่อ)

ยูสเคส	Search Directory
ข้อมูลที่ได้	เว็บเพจที่แสดงรายละเอียดของข้อมูลที่ต้องการ
ขั้นตอนทั่วไป	<ol style="list-style-type: none"> 1. ผู้ดูแลระบบร้องขอข้อมูลที่ต้องการค้นหาจากทางเว็บของเว็บเซิร์ฟเวอร์ 2. เว็บเซิร์ฟเวอร์ทำการร้องขอบริการ โดยการระบุรายละเอียดข้อมูลของผู้ดูแลระบบที่ต้องการ 3. ไคลเอนต์เซิร์ฟเวอร์ค้นหาข้อมูลของผู้ดูแลระบบที่ต้องการ พร้อมกับแสดงรายละเอียดผ่านทางเว็บเพจ

ยูสเคส Export LDIF กระบวนการทำงานในขั้นตอนนี้จะเป็นการนำโครงสร้างของไคลเอนต์ทั้งหมดมาสร้างเป็นไฟล์ข้อความเพื่อแสดงให้กับผู้ใช้สามารถดูรายละเอียดของโครงสร้างต้นไม้ของไคลเอนต์ กระบวนการในส่วนนี้สามารถนำมาอธิบายการทำงานได้ดังนี้

ตารางที่ 3.7 รายละเอียดประกอบการทำงานของ Export LDIF

ยูสเคส	Export LDIF
วัตถุประสงค์	การนำข้อมูลออกนอกระบบ
เงื่อนไขเริ่มต้น	ผู้ดูแลระบบต้องทราบเซิร์ฟเวอร์ที่เก็บข้อมูลที่จะนำออกนอกระบบ
เมื่อทำงานสำเร็จ	สามารถนำข้อมูลในไคลเอนต์มาแสดงในรูปของไฟล์ตัวอักษรนอกระบบได้
เมื่อทำงานไม่สำเร็จ	ไม่สามารถนำข้อมูลในไคลเอนต์มาแสดงในรูปของไฟล์ตัวอักษร
Actor ที่เกี่ยวข้อง	ผู้ดูแลระบบ, เว็บเซิร์ฟเวอร์, ไคลเอนต์เซิร์ฟเวอร์
สิ่งกระตุ้นการทำงาน	-
ข้อมูลเข้า	ตำแหน่งที่เก็บข้อมูลที่ต้องการจะนำออกนอกระบบ
ข้อมูลที่ได้	ไฟล์ตัวอักษรที่แสดงรายละเอียดของข้อมูลในไคลเอนต์
ขั้นตอนทั่วไป	<ol style="list-style-type: none"> 1. ผู้ดูแลระบบร้องขอข้อมูลที่ต้องการนำออกจากทางเว็บของเว็บเซิร์ฟเวอร์ 2. เว็บเซิร์ฟเวอร์ทำการร้องขอบริการ โดยการระบุรายละเอียดข้อมูลของผู้ดูแลระบบที่ต้องการจะนำออก 3. ไคลเอนต์เซิร์ฟเวอร์ค้นหาข้อมูลของผู้ดูแลระบบที่ต้องการ พร้อมกับส่งข้อมูลให้ผู้ดูแลระบบที่ต้องการ ในรูปแบบของไฟล์ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 การออกแบบแอตทริบิวต์และโอเปอเรชันของคลาส LDAP Connection

ชื่อคลาส : LDAP Connection		
คำอธิบาย : เป็นคลาสที่ทำหน้าที่ในการสร้างการติดต่อกับไดเร็กทอรี		
แอตทริบิวต์		
ชื่อ	ประเภท	คำอธิบาย
Name	String	ชื่อเอนทรี
Val	Object	ค่าในเอนทรี
Host	String	ชื่อเซิร์ฟเวอร์
Port	Integer	หมายเลข port
Dn	String	ค่า distingunish name ของเอนทรี
Rdn	String	ค่า Relationdistingunish nameของเอนทรี
Password	String	ค่ารหัสในการยืนยันตัวตน
Cons	LDAP Constraint	ค่าเริ่มต้นในการติดต่อ
Attr	String	แอตทริบิวต์
Ldapmsg	LDAP Message	ข้อความตามมาตรฐาน LDAP
Is	InputStream	ข้อความที่ส่งไปเซิร์ฟเวอร์
Os	OutputStream	ข้อความที่รับจากเซิร์ฟเวอร์
Entry	LDAP Entry	เอนทรี
Mods	LDAP Modification	การเปลี่ยนแปลงเอนทรี
โอเปอเรชัน		
ชื่อ	คำอธิบาย	
LDAPConnection()	กระบวนการในการสร้างการติดต่อ	
finalize()	กระบวนการกระบวนการในการสิ้นสุดในการทำงาน	
setCatch(catch)	กระบวนการกำหนดค่า catch	
setProductVersion()	กระบวนการกำหนด Version	
setOutputStream()	กระบวนการในการกำหนดข้อความที่ส่งไปเซิร์ฟเวอร์	
getOutputStream()	กระบวนการในการรับข้อความที่ส่งไปเซิร์ฟเวอร์	
getCatch()	กระบวนการรับค่า catch	
getProperty(name)	กระบวนการในการดึงแอตทริบิวต์ของเอนทรี	

ตารางที่ 3.8 การออกแบบแอตทริบิวต์และโอเปอเรชันของคลาส LDAP Connection (ต่อ)

getHost()	กระบวนการในการรับชื่อเซิร์ฟเวอร์
getPort()	กระบวนการในการรับหมายเลขพอร์ต
getAuthenticationDN()	กระบวนการในการรับDistingunishName ของผู้ใช้
getAuthenticationPsw()	กระบวนการในการรับรหัสยืนยันตัวตนของเซิร์ฟเวอร์
getSocketFactory()	กระบวนการในการรับค่าsocket ที่ใช้ในการติดต่อ
getResponseControl()	กระบวนการในการรับค่าที่ส่งกลับมาจากการติดต่อ
getConstriant()	กระบวนการในการรับค่าเริ่มต้น
getInputStream()	กระบวนการในการรับค่าที่ส่งกลับเซิร์ฟเวอร์
getOutputStream()	กระบวนการในการรับค่าที่ส่งจากเซิร์ฟเวอร์
isConnect()	กระบวนการในการตรวจสอบการเชื่อมต่อ
isAuthenticate()	กระบวนการในการตรวจสอบการยืนยันตัวตน
connect()	กระบวนการในการสร้างการติดต่อ
authenticate()	กระบวนการในการรับรองสิทธิ์
checkMsg()	กระบวนการในการตรวจสอบข้อความ
checkCommunicator()	กระบวนการในการตรวจสอบการเชื่อมต่อ
resultRetrieve()	กระบวนการดึงผลลัพธ์ที่ได้
reconnect()	กระบวนการในการสร้างการเชื่อมต่อ
Bind()	กระบวนการในการสร้างการเชื่อมต่อ
rename()	กระบวนการในการเปลี่ยนชื่อเอนทรี
compare()	กระบวนการในการเปรียบเทียบเอนทรี
Add()	กระบวนการในการเพิ่มเอนทรี
disconnect()	กระบวนการในการยกเลิกการติดต่อ
read()	กระบวนการในการอ่านค่าในเอนทรี
search()	กระบวนการในการค้นหาเอนทรี
mofidy()	กระบวนการในการเปลี่ยนแปลงค่าในเอนทรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 การออกแบบแอดทริบิวต์และโอเปอเรชั่นของคลาส LDAP ConnectionPool

ชื่อคลาส : LDAP ConnectionPool		
คำอธิบาย : เป็นคลาสที่จัดการกับการกำหนดจำนวนการเชื่อมต่อ		
แอดทริบิวต์		
ชื่อ	ประเภท	คำอธิบาย
connectionMsg	String	ข้อความในการสร้างการเชื่อมต่อ
connectionObject	Object	ออบเจ็คในการเชื่อมต่อ
name	String	ชื่อการเชื่อมต่อ
host	String	ชื่อในการเชื่อมต่อ
Port	Integer	หมายเลขพอร์ต
Lcon	LDAPConnection	การเชื่อมต่อ
โอเปอเรชั่น		
ชื่อ	คำอธิบาย	
getConnectionPool()	กระบวนการในการดึงค่า connectionpool	
setConnectionPool()	กระบวนการในการบันทึกค่า connectionpool	
addConnectionPool()	กระบวนการในการเพิ่ม connectionpool	
destroy()	กระบวนการในการลบ connectionpool	
disconnect()	กระบวนการยกเลิกการติดต่อ	
setUpPool()	กระบวนการกำหนดค่าใน connectionpool	
createPool()	กระบวนการในการสร้าง connectionpool	

ตารางที่ 3.10 การออกแบบแอดทริบิวต์และโอเปอเรชั่นของคลาส LDAPMessage

ชื่อคลาส : LDAPMessage		
คำอธิบาย : เป็นคลาสในการจัดการกับข้อความในการติดต่อตามมาตรฐานโพรโทคอล LDAP		
แอดทริบิวต์		
ชื่อ	ประเภท	คำอธิบาย
msgID	Integer	หมายเลขของข้อความ
Op	String	กระบวนการทำงาน
s	String	ข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 การออกแบบแอดทริบิวต์และ โอเปอเรชั่นของคลาส LDAPMessage (ต่อ)

โอเปอเรชั่น	
ชื่อ	คำอธิบาย
LDAPMessage()	กระบวนการในการสร้างข้อความตามมาตรฐาน LDAP
parseMessage()	กระบวนการในการจัดการกับข้อความ
getID()	กระบวนการในการรับหมายเลขข้อความ
getErrorMsg()	กระบวนการในการรับข้อความแสดงความผิดพลาด
getType()	กระบวนการในการรับประเภทของข้อความ
write()	กระบวนการในการเขียนข้อความ
getResultCode()	กระบวนการในการรับรหัสของข้อความแสดงผล
getResult()	กระบวนการในการรับข้อความแสดงผล

ตารางที่ 3.11 การออกแบบแอดทริบิวต์และ โอเปอเรชั่นของคลาส LDAPException

ชื่อคลาส : LDAPException		
คำอธิบาย : คลาสจัดการกับข้อความในการแสดงความผิดพลาด		
แอดทริบิวต์		
ชื่อ	ประเภท	คำอธิบาย
msg	String	ข้อความแสดงความผิดพลาด
resultCode	Integer	หมายเลขของความผิดพลาด
โอเปอเรชั่น		
ชื่อ	คำอธิบาย	
LDAPException()	กระบวนการดักจับความผิดพลาด	
getLDAPResultCode()	กระบวนการในการส่งค่าความผิดพลาด	
getLDAPErrorMsg()	กระบวนการในการส่งข้อความแสดงความผิดพลาด	
LDAPInterruptException()	กระบวนการในการตรวจสอบความผิดพลาด	
toString()	กระบวนการในการเปลี่ยนเป็นข้อความแสดงความผิดพลาด	

ตารางที่ 3.12 การออกแบบแอตทริบิวต์และโอเปอเรชันของคลาส LDAPEntryWrapper

ชื่อคลาส : LDAPEntryWrapper		
คำอธิบาย : คลาสในการจัดการกับเอนทรีในการหาความสัมพันธ์ระหว่างเอนทรี		
แอตทริบิวต์		
ชื่อ	ประเภท	คำอธิบาย
children	LDAPEntryWrapper	เอนทรีที่เป็น Node ลูก
conn	LDAPConnection	การเชื่อมต่อกับ LDAP
entry	LDAPEntry	เอนทรีที่อยู่ในไคลเร็คทอรี
Modified	Boolean	Flag ที่เก็บค่าในการเปลี่ยนแปลง
โอเปอเรชัน		
ชื่อ	คำอธิบาย	
addChild()	กระบวนการเพิ่มเอนทรีที่เป็น Node ลูก	
deleteChild()	กระบวนการลบเอนทรีที่เป็น Node ลูก	
getAllChildren()	กระบวนการดึงเอนทรีที่เป็นลูกทั้งหมด	
getAttributeSet()	กระบวนการดึงค่าแอตทริบิวต์ที่เป็นเอนทรีลูกทั้งหมด	
getDN()	กระบวนการดึงเอาค่า distinguish name ของเอนทรีนั้นๆขึ้นมา	
LDAPEntryWrapper()	กระบวนการในการห่อเอนทรีหลายอันเข้าไว้ด้วยกัน	
Modified()	กระบวนการเปลี่ยนแปลงกับเอนทรี	

ตารางที่ 3.13 การออกแบบแอตทริบิวต์และโอเปอเรชันของคลาส ConnectionManager

ชื่อคลาส : ConnectionManager		
คำอธิบาย : คลาสในการจัดการเกี่ยวกับการเชื่อมต่อกับไคลเร็คทอรี		
แอตทริบิวต์		
ชื่อ	ประเภท	คำอธิบาย
configDir	File	ที่อยู่ของ configuration
configurationMap	HashMap	รายละเอียดของ configuration
โอเปอเรชัน		
ชื่อ	คำอธิบาย	
connection()	กระบวนการในการสร้างการติดต่อ	

ตารางที่ 3.13 การออกแบบแอตทริบิวต์และโอเปอเรชั่นของคลาส ConnectionManager (ต่อ)

getAllConfiguration()	กระบวนการในการส่งค่าการติดต่อทั้งหมดที่มี
getConfiguration()	กระบวนการในการส่งค่าการติดต่อกลับ
getLDAPRootEntry()	กระบวนการในการส่งค่าเอนทรีที่เป็นเอนทรีเริ่มต้น
init()	กระบวนการทำงานเริ่มต้นครั้งแรก
main()	กระบวนการทำงานหลักที่ใช้ในการสร้างconfiguration
printNodeAndChildren()	กระบวนการในการดึงnodeลูก
removeConfiguration()	กระบวนการในการลบ configuration
saveConfiguration()	กระบวนการในการบันทึก configuration
connectionManager()	กระบวนการในการนำ configuration มาสร้างเป็น connection
loadConfiguration()	กระบวนการในการดึงค่า configuration มาใช้
initConfiguration()	กระบวนการในการทำงานกับค่า configuration เริ่มต้น

ตารางที่ 3.14 การออกแบบแอตทริบิวต์และโอเปอเรชั่นของคลาส ConfigurationLDAP

ชื่อคลาส : ConfigurationLDAP		
คำอธิบาย : คลาสในการจัดการกับไฟล์ configuration		
แอตทริบิวต์		
ชื่อ	ประเภท	คำอธิบาย
authType	String	ประเภทในการรับรองสิทธิ
baseDN	String	Distinguish Name เริ่มต้น
name	String	ชื่อของ configuration
password	String	รหัสที่ใช้ในการยืนยันตัวตน
port	Integer	หมายเลข Port ที่ใช้ในการติดต่อ
Server	String	ชื่อของเซิร์ฟเวอร์ที่ใช้ในการติดต่อ
Users	String	ชื่อของเอนทรีของผู้ใช้

บทที่ 4

การพัฒนาระบบงาน

ในบทที่ 4 นี้จะกล่าวถึงการพัฒนากระบวนการจัดการฐานข้อมูลผู้ใช้ โดยการนำโปรโตคอล LDAP มาประยุกต์ใช้ ซึ่งภายในบทนี้จะอธิบายถึงการจัดการสร้างไคลเอนต์ และการสร้างการติดต่อระหว่างไคลเอนต์ไคลเอนท์และไคลเอนต์เซิร์ฟเวอร์ การทดสอบการทำงานของระบบจัดการฐานข้อมูลผู้ใช้ในองค์กรผ่านระบบเครือข่ายอินเทอร์เน็ต และแสดงผลการทำงานของระบบจัดการฐานข้อมูลผู้ใช้

4.1 กำหนดวัตถุประสงค์

ในปัจจุบันในการจัดการเกี่ยวกับระบบรายชื่อผู้ใช้ในองค์กรนั้น ได้แยกการจัดการระบบรายชื่อออกจากกันในแต่ละส่วน แต่ละแผนก ทำให้เกิดความซับซ้อนของระบบรายชื่อผู้ใช้ในองค์กร ทำให้ยุ่งยากในการกำหนดมาตรฐานของข้อมูล และการกำหนดมาตรการในการรักษาความปลอดภัย

จากปัญหาที่เกิดขึ้นจึงมีการนำเอาเทคโนโลยีไคลเอนต์เซิร์ฟเวอร์ มาใช้ในการพัฒนาระบบฐานข้อมูล ซึ่งระบบฐานข้อมูลแบบไคลเอนต์เซิร์ฟเวอร์นั้นเป็นฐานข้อมูลที่จะอำนวยความสะดวกในการนำไปประยุกต์กับงานต่างๆ ได้อย่างหลากหลาย โดยนำเอาความสามารถในการสืบค้นข้อมูลอย่างรวดเร็ว และความสามารถในการจัดการเกี่ยวกับการรักษาความปลอดภัยให้กับฐานข้อมูล โดยใช้การกำหนดสิทธิในการเข้าถึงข้อมูลในฐานข้อมูล เมื่อนำเอาเทคโนโลยีไคลเอนต์เซิร์ฟเวอร์มาใช้ในการจัดการเกี่ยวกับฐานข้อมูลผู้ใช้ภายในองค์กรแล้ว ทำให้เพิ่มประสิทธิภาพให้กับองค์กรมากยิ่งขึ้น ในรายงานฉบับนี้จะกล่าวถึงเทคโนโลยีไคลเอนต์เซิร์ฟเวอร์ การวิเคราะห์ การออกแบบ และการพัฒนาโปรแกรมสำหรับบริหารฐานข้อมูลผู้ใช้ภายในองค์กรรวมทั้งเครื่องมือที่ใช้ในการพัฒนา

โดยที่ในโครงการพัฒนาโปรแกรมบริหารฐานข้อมูลผู้ใช้โดยใช้โปรโตคอล LDAP นี้ได้ทำการศึกษาและพัฒนาโปรแกรมที่จัดการฐานข้อมูลผู้ใช้ในองค์กรขึ้นมา โดยการจำลองการทำงานของฐานข้อมูลรายชื่อผู้ใช้งานในองค์กร และทำการสร้างไคลเอนต์ที่ใช้ในการเก็บข้อมูล และสร้างโปรแกรมที่ช่วยในการจัดการข้อมูลในไคลเอนต์ และนำเอาความสามารถต่างๆ ของไคลเอนต์เซิร์ฟเวอร์ มาเพื่อเพิ่มประสิทธิภาพให้กับการจัดการกับฐานข้อมูลรายชื่อผู้ใช้ในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 อุปกรณ์และโปรแกรมที่ใช้ในการพัฒนาระบบฐานข้อมูลผู้ใช้

ระบบฐานข้อมูลผู้ใช้ต้องประกอบไปด้วยเครื่องคอมพิวเตอร์อย่างน้อย 2 เครื่องขึ้นไป โดยที่เครื่องหนึ่งทำหน้าที่เป็นไคลเอนท์เซิร์ฟเวอร์ และอีกเครื่องหนึ่งทำหน้าที่เป็นเครื่องไคลเอนท์ และเชื่อมต่อกันเป็นระบบเครือข่าย แต่ในโครงการพัฒนาระบบนี้ จะทำการสร้างระบบเครือข่ายจำลองโดยนำเอาโปรแกรมมาช่วยในการจำลองระบบเครือข่าย ทำให้สามารถพัฒนาและทดลองโครงการนี้ได้โดยอาศัยเพียงคอมพิวเตอร์เครื่องเดียวได้ ทางผู้พัฒนาระบบได้ทำการจัดเตรียมเครื่องคอมพิวเตอร์ที่มีความเหมาะสมในการใช้งานมีรายละเอียดดังต่อไปนี้

ตารางที่ 4.1 รายละเอียดของเครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนา

CPU	Pentium 3 800 MHz
หน่วยความจำ	256 MB
ฮาร์ดดิสก์	20 GB
กราฟิกการ์ด	NVidia TNT 2-64M
OS	Window XP service pack 1
โปรแกรมประยุกต์	Jbuilder, VmWare

นอกจากนี้ ยังมีการติดตั้งระบบปฏิบัติการ FreeBSD ที่รูปแบบการทำงานและโครงสร้างคล้าย Unix ลงบน VmWare เพื่อทำหน้าที่เป็นไคลเอนท์เซิร์ฟเวอร์ โดย VmWare นั้นจะเป็นโปรแกรมที่ช่วยจำลองเครื่องคอมพิวเตอร์ โดยทำการแบ่งทรัพยากรให้กับเครื่องคอมพิวเตอร์เสมือน และการเชื่อมต่อระหว่างเครื่องไคลเอนท์กับเครื่องคอมพิวเตอร์เสมือนที่ทำหน้าที่เป็นเซิร์ฟเวอร์นั้นสามารถใช้โปรแกรม VmWare ในการสร้างการเชื่อมต่อจำลองเสมือนต่อด้วยอุปกรณ์เครือข่ายจริง ส่วนในการเก็บฐานข้อมูลรายชื่อผู้ใช้ได้ทำการติดตั้งโปรแกรม Openldap บนเครื่องคอมพิวเตอร์เสมือนเพื่อใช้ในการจัดการกับไคลเอนท์ ในการพัฒนาระบบใช้โปรแกรม Jbuilder ในการพัฒนาโปรแกรมที่ใช้จัดการกับฐานข้อมูลผู้ใช้ในองค์กรด้วยภาษา JAVA

4.3 ขั้นตอนในการพัฒนาระบบฐานข้อมูลผู้ใช้

เนื่องจากระบบฐานข้อมูลผู้ใช้นั้นต้องใช้ในการติดต่อผ่านระบบเครือข่ายเพื่อเข้าถึงฐานข้อมูลส่วนกลาง ซึ่งเป็นการทำงานแบบไคลเอนท์/เซิร์ฟเวอร์ เพื่อจำลองการทำงานให้เหมือนกับการใช้งานจริง ผู้พัฒนาจึงนำเอาโปรแกรม VmWare มาใช้ในการจำลองระบบเครือข่ายเพื่อพัฒนาระบบฐานข้อมูลผู้ใช้ซึ่งมีขั้นตอนการพัฒนาค้างนี้

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ติดตั้งโปรแกรม VmWare ในเครื่องที่ใช้ในการจำลองว่าเป็นเครื่องไคลเอนท์ ซึ่งเครื่องที่ใช้ควรที่จะมีพื้นที่ไม่ต่ำกว่า 20 GB และหน่วยความจำไม่ต่ำกว่า 128 MB เนื่องจากต้องแบ่งพื้นที่ส่วนหนึ่งให้กับ VmWare ในการทำงานเช่น หน่วยความจำที่จะใช้ พื้นที่ใช้ในการเก็บข้อมูล เป็นต้น และทำการเลือกการเชื่อมต่อว่าจะจำลองการเชื่อมต่อระหว่าง VmWare กับเครื่องไคลเอนท์ ซึ่งการพัฒนาแบบนี้จะเลือกการเชื่อมต่อด้วย Bridge

2. ติดตั้งระบบปฏิบัติการใน VmWare ซึ่งในการพัฒนาแบบนี้ จะใช้ระบบปฏิบัติการ FreeBSD ในการทำงานเป็นเซิร์ฟเวอร์ เนื่องจากมีส่วนที่คล้ายกับระบบปฏิบัติการ Unix และทำการกำหนดหมายเลข IP และ Subnet Mask ของทั้งสองเครื่อง พร้อมทำการทดสอบการเชื่อมต่อระหว่าง VmWare กับภายนอกโดยทำการ ping ทดสอบ

3. ติดตั้ง Openldap บนระบบปฏิบัติการ FreeBSD โดยทำการคลายการบีบอัดโดยใช้คำสั่ง “gunzip -c openldap-VERSION.tgz | tar xvfB -“ เมื่อเสร็จแล้ว ทำการเข้าถึงไดเรกทอรีที่เกิดจากการคลายการบีบอัด แล้วทำการสร้างโปรแกรมโดยใช้คำสั่ง “make depend” และ “make” แล้วติดตั้งโปรแกรม openldap ซึ่งโปรแกรมจะถูกติดตั้งลงใน /usr/local

```

FreeBSD OpenLDAP - VMware Workstation
File Edit View VM Power Snapshot Windows Help
Snapshot Revert
Home FreeBSD OpenLDAP
localtime printcap wall_cmos_clock
openldap# cd /usr/local
openldap# ls
.cshrc boot entropy proc sys
.profile cdrom etc rescue syslogd.core
.snap compat lib root tmp
COPYRIGHT dev libexec sbin usr
bin dist mnt stand var
openldap# cd /usr/local
openldap# ls
.snap bin games lib libexec obj share sup
X11R6 compat include libdata local sbin src
openldap# cd /usr/local/etc
openldap# ls
bin include lib libexec openssl share
etc info libdata man sbin
openldap# cd /usr/local/etc/openldap
openldap# ls
pam.d rc.d
openldap# ls
Begin.ldif ldap.conf slapd.conf
a.ldif ldap.conf.default slapd.conf.default
b.ldif schema
openldap#

```

⚠ You do not have VMware Tools installed.

รูปที่ 4.1 หน้าจอการติดตั้ง OpenLDAP

4. ตั้งค่าเริ่มต้นของ Openldap โดยทำการแก้ไขไฟล์ slapd.conf ซึ่งปรกติจะติดตั้งที่

/usr/local/etc/openldap/slapd.conf โดยกำหนดค่าดังนี้

เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

database bdb

suffix "dc=<MY-DOMAIN>,dc=<COM>"

rootdn "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>"

rootpw secret

directory /usr/local/var/openldap-data

โดยทำการแทนที่ <MY-DOMAIN> และ <COM> ด้วยชื่อของโดเมนของระบบ ซึ่งจะเป็นการกำหนดค่าเริ่มต้นให้กับฐานข้อมูล

```

FreeBSD OpenLDAP - VMware Workstation
File Edit View VM Power Snapshot Windows Help
Snapshot Revert
Home FreeBSD OpenLDAP
# if no access controls are present, the default policy is:
# Allow read by all
# rootdn can always write!
#####
# ldbm database definitions
#####
database bdb
suffix "dc=kmitl,dc=ac,dc=th"
rootdn "cn=Manager,dc=kmitl,dc=ac,dc=th"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory /var/db/openldap-data
# Indices to maintain
index objectClass eq

You do not have VMware Tools installed.

```

รูปที่ 4.2 หน้าจอการกำหนดค่าเริ่มต้นของ OpenLDAP

5. รัน โพรเซส Slapd เพื่อให้ทำหน้าที่ในการจัดการกับฐานข้อมูลผู้ใช้ โดยใช้คำสั่ง "su root -c /usr/local/libexec/slapd"

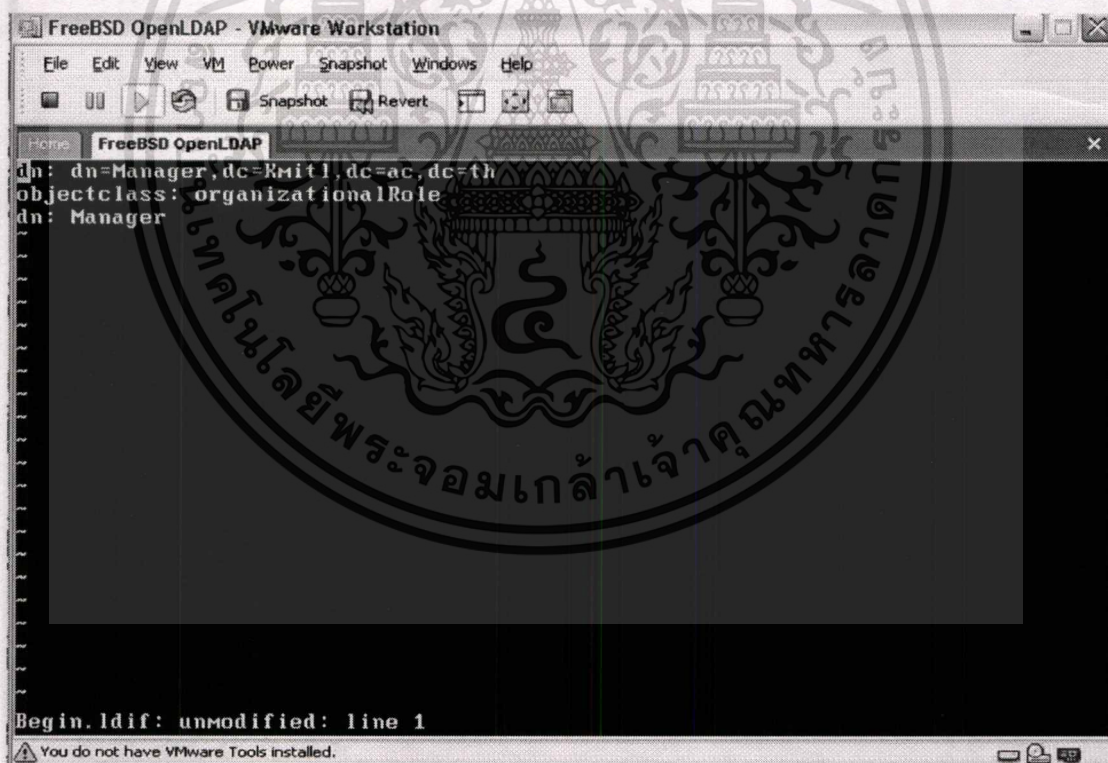
6. ทดสอบโดยเพิ่มเอนทรีโดยสร้างไฟล์ที่เป็นรายละเอียดของเอนทรีขึ้นมาด้วยโปรแกรม vi ที่มากับระบบปฏิบัติการ FreeBSD ในการสร้างไฟล์ที่มีนามสกุล LDIF ซึ่งมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dn: dc=<MY-DOMAIN>,dc=<COM>
objectclass: dcObject
objectclass: organization
o: <MY ORGANIZATION>
dc: <MY-DOMAIN>
dn: cn=Manager,dc=<MY-DOMAIN>,dc=<COM>
objectclass: organizationalRole
cn: Manager
```

แล้วบันทึกเป็นไฟล์ sample.ldif เพื่อทดลองในการเพิ่มเอนทรีเข้าไปในฐานข้อมูลโดยใช้คำสั่ง ldapadd -x -D "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>" -W -f example.ldif ในการเพิ่มเอนทรี



รูปที่ 4.3 หน้าจอการทดสอบการใช้งาน OpenLDAP

ทำการทดสอบผลการทดลองโดยทำการค้นหาเอนทรีที่เพิ่มเข้าไปโดยใช้คำสั่งในการค้นหาเอนทรีดังนี้ ldapsearch -x -b 'dc=example,dc=com' '(objectclass=*)' คำสั่งนี้จะค้นหาและคืนค่าเป็นเอนทรีทั้งหมดที่มีในฐานข้อมูล

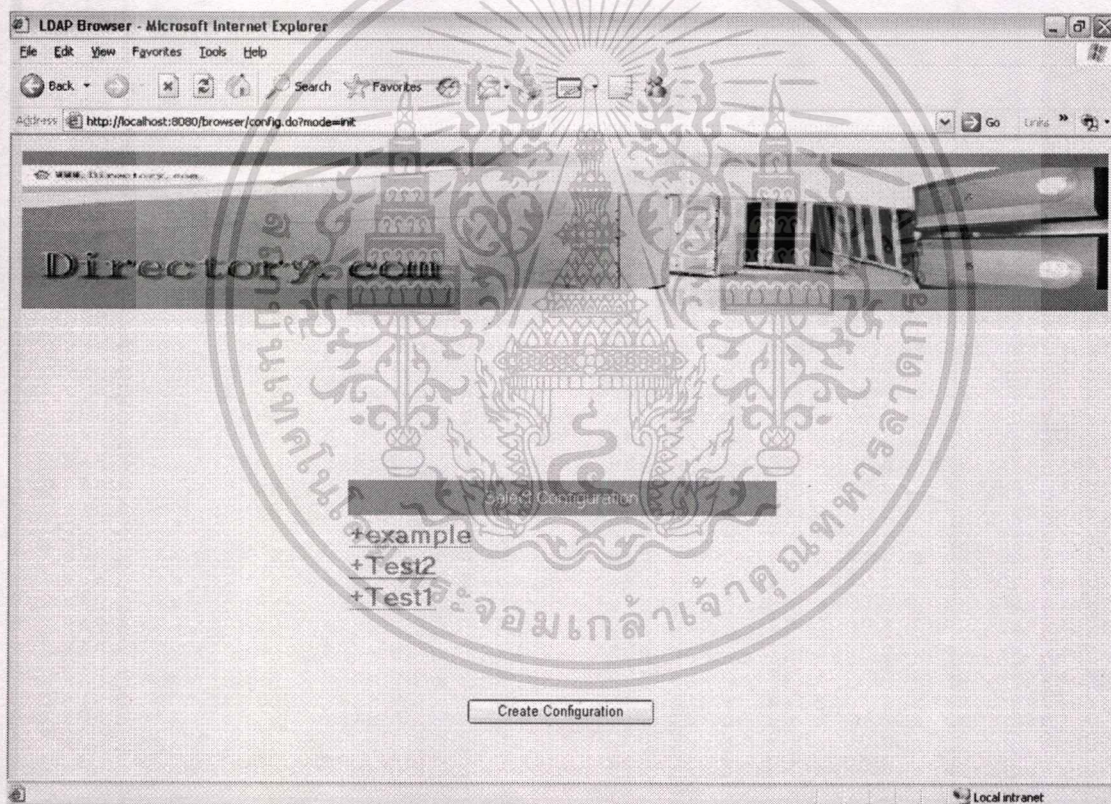
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 หน้าจอการทำงานของระบบ

หน้าจอการทำงานของระบบสามารถแบ่งการทำงานออกเป็น 6 ส่วนคือ

4.4.1 การสร้างการติดต่อกับไดเรกทอรีเซิร์ฟเวอร์

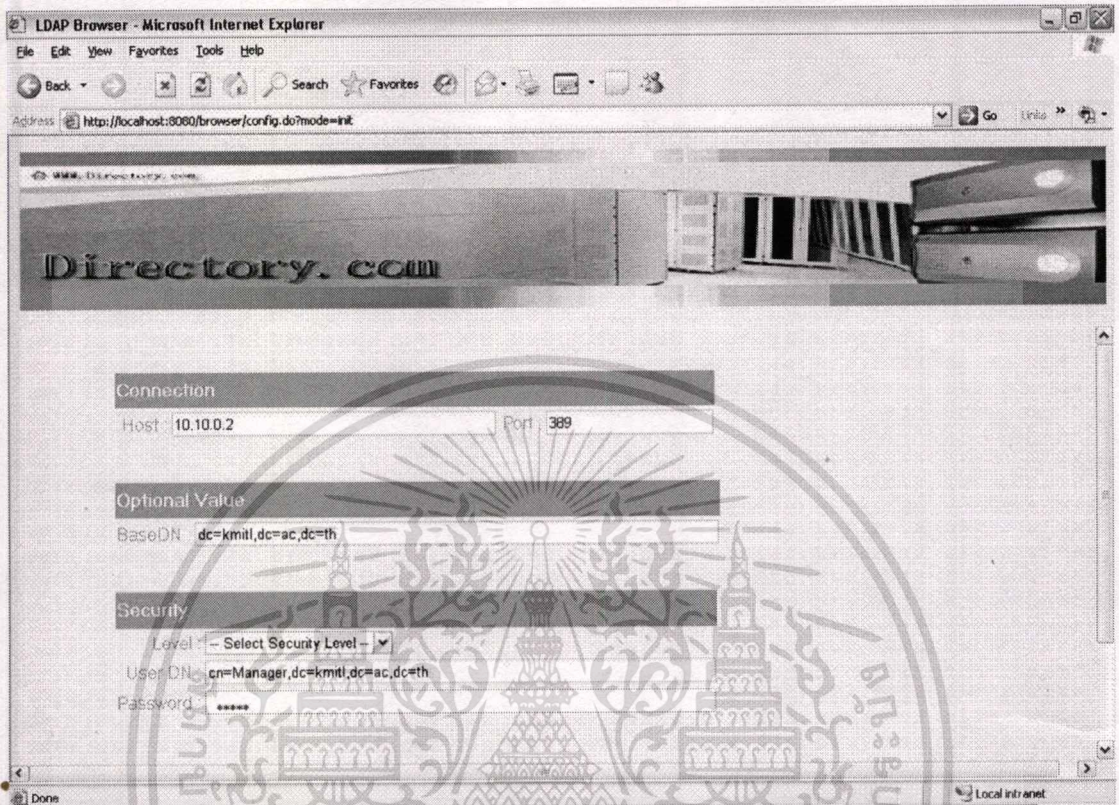
ผู้ใช้ที่ต้องการเข้ามาใช้งานระบบจะต้องมีการกำหนดค่าเริ่มต้นในการเชื่อมต่อกับไดเรกทอรีเซิร์ฟเวอร์ โดยอาจจะกำหนดผ่านทางไฟล์คอนฟิก (Configuration File) ซึ่งผู้ใช้อาจจะเรียกใช้ไฟล์คอนฟิกที่มีอยู่แล้วในระบบหรืออาจสร้างไฟล์ขึ้นมาใหม่ก็ได้ ดังรูป



รูปที่ 4.4 หน้าจอสำหรับเลือกไฟล์คอนฟิกเพื่อติดต่อกับไดเรกทอรีเซิร์ฟเวอร์

เมื่อผู้ใช้ได้เลือกไฟล์คอนฟิกแล้วข้อมูลทั้งหมดที่ใช้ในการเชื่อมต่อที่อยู่ในไฟล์คอนฟิกจะถูกดึงขึ้นมาแสดงที่หน้าจอเพื่อให้ผู้ใช้ทำการแก้ไขข้อมูล (หรือสร้างไฟล์ขึ้นมาใหม่) พร้อมทั้งระบุรหัสผ่านเพื่อให้ระบบนำข้อมูลที่เป็นในการเชื่อมต่อมาใช้ในทำการติดต่อกับไดเรกทอรีเซิร์ฟเวอร์ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 หน้าจอการระบุข้อมูลสำหรับการติดต่อไคลเอนต์ไคลเอนต์เซิร์ฟเวอร์

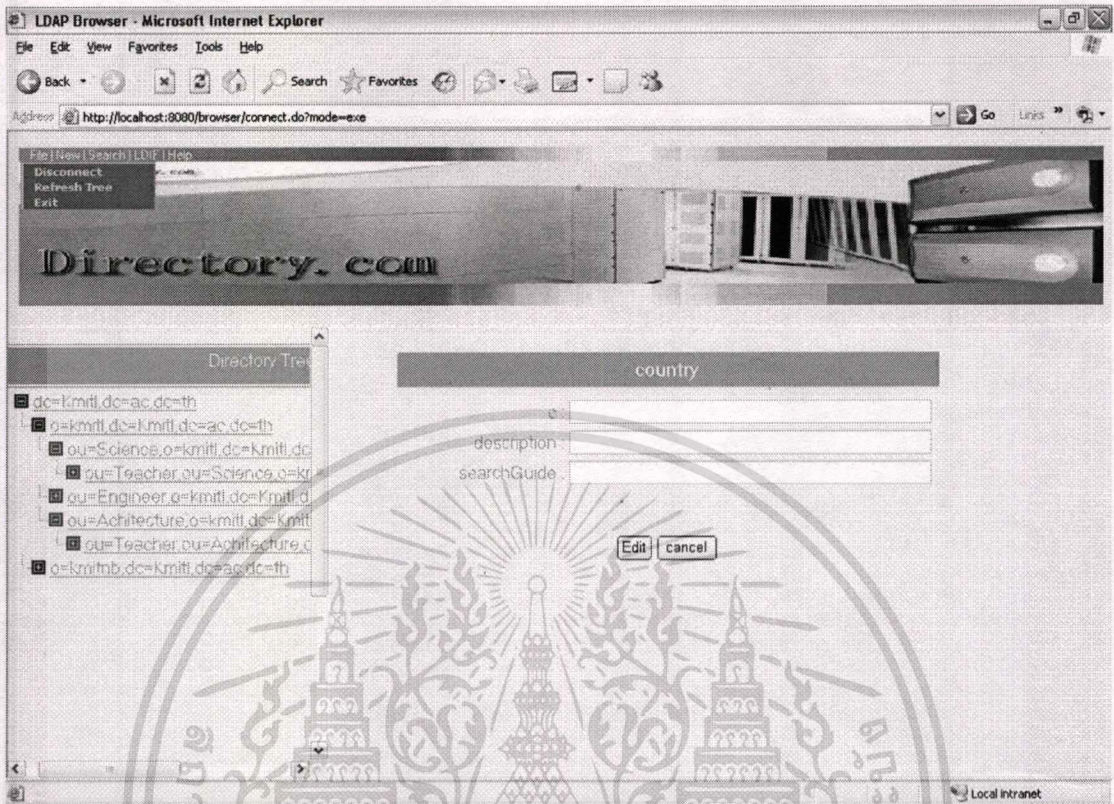
ในกรณีที่ข้อมูลและรหัสผ่านที่ผู้ใช้ระบุนั้นถูกต้อง ผู้ใช้ก็จะสามารถเข้ามาใช้งานในระบบ
ได้

4.4.2 การทำงานของเมนู File

ในเมนู File ผู้ใช้สามารถทำงานได้ดังนี้

- Disconnect การยกเลิกการติดต่อ
- Refresh Tree การรีเฟรชข้อมูลในเอ็นทรี
- Exit การออกจากระบบ

หน้าจอการทำงานของส่วนนี้แสดงได้ดังรูปต่อไปนี้



รูปที่ 4.6 หน้าจอการทำงานของเมนู File

4.4.3 การเพิ่มเอนทรีภายในไดเรกทอรี (เมนู New)

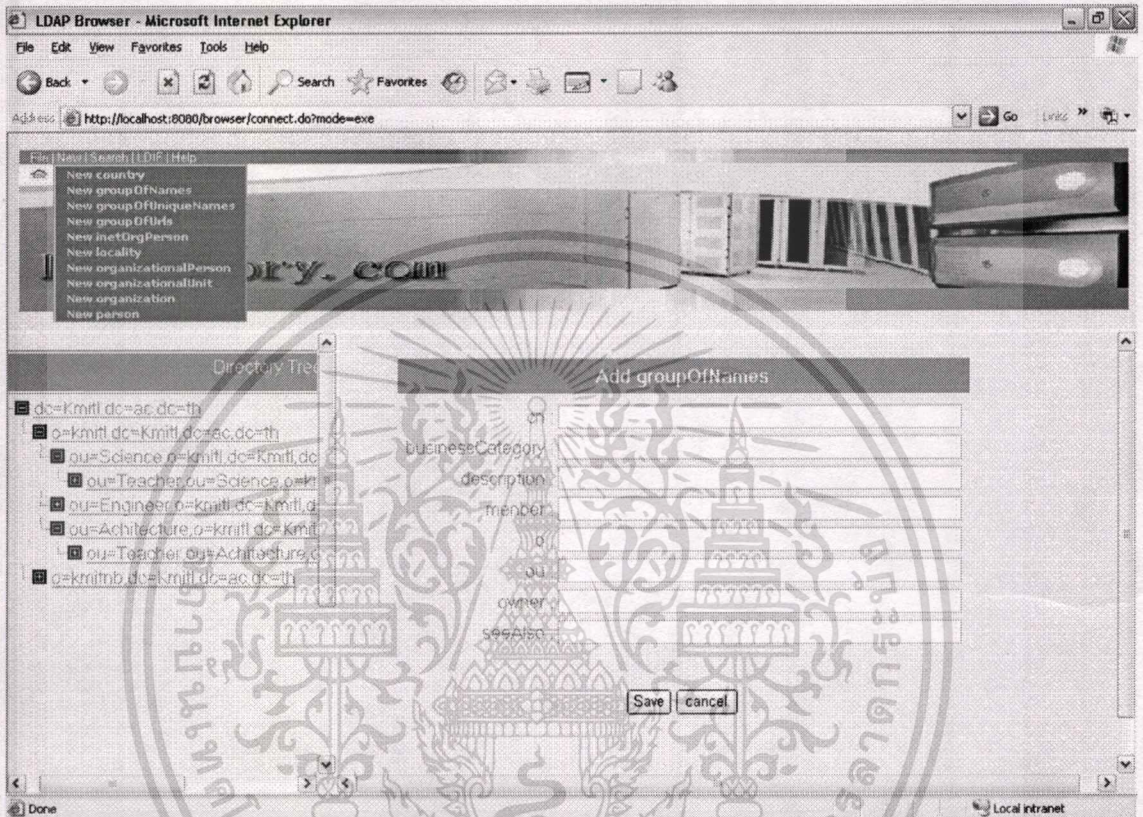
การทำงานในส่วนนี้ผู้ใช้สามารถเพิ่มข้อมูลในเอนทรีได้ตามต้องการ สามารถเพิ่มเข้ามาในเอนทรีจะประกอบด้วย

ซึ่งข้อมูลที่ผู้ใช้

- ข้อมูล Country
- ข้อมูล groupOfNames
- ข้อมูล groupOfUniqueNames
- ข้อมูล groupOfUrls
- ข้อมูล inetOrgPerson
- ข้อมูล locality
- ข้อมูล organizationalPerson
- ข้อมูล organizationalUnit
- ข้อมูล organization
- ข้อมูล person

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.7 เป็นการแสดงตัวอย่างของการเพิ่มข้อมูลเข้าไปในเอนทรี ซึ่งตัวอย่างนี้เป็น การเพิ่ม groupOfNames เข้าไปในเอนทรี



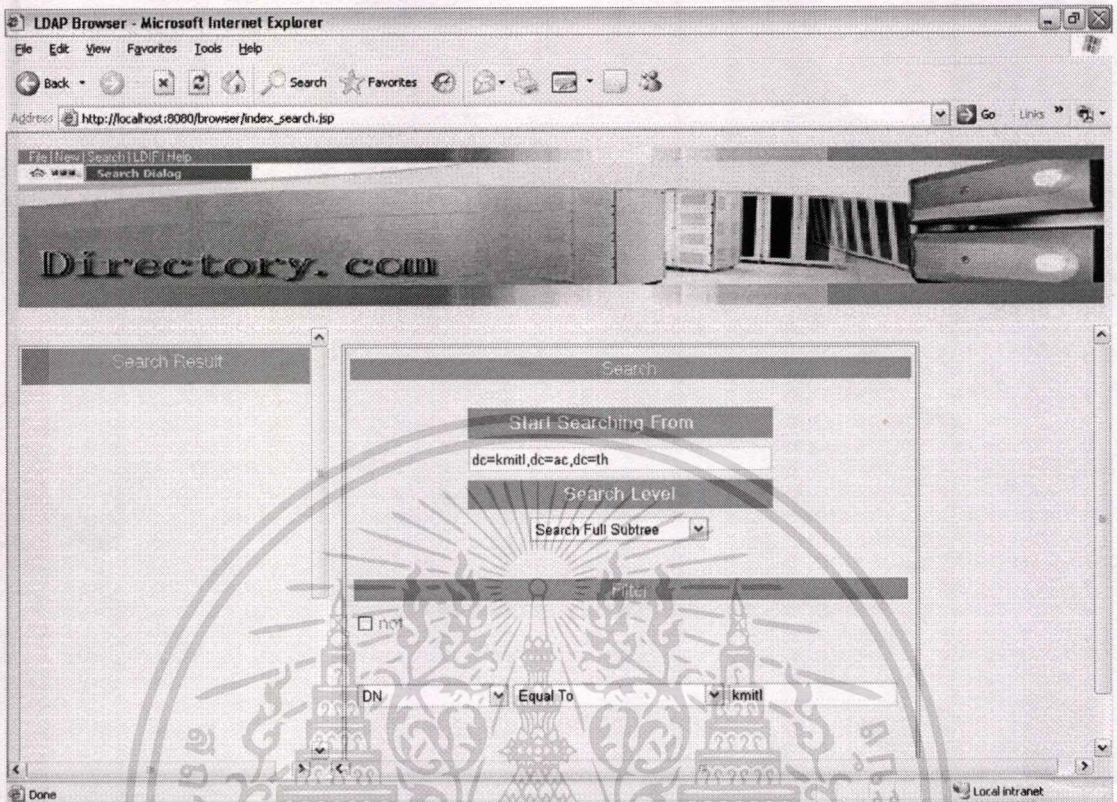
รูปที่ 4.7 หน้าจอการเพิ่มข้อมูล groupOfNames

4.4.4 การค้นหาข้อมูลภายในเอนทรี (เมนู Search)

ผู้ใช้งานสามารถค้นหาข้อมูลที่อยู่ในเอนทรี กำหนดเงื่อนไขต่างๆในการค้นหาคือ ข้อมูลเอนทรีเริ่มต้นที่ผู้ใช้งานต้องการค้นหา ระดับของการค้นหา แอดทริบิวต์ที่ต้องการค้นหา ค่าที่ใช้ในการค้นหา เป็นต้น นอกจากนี้ผู้ใช้งานยังสามารถระบุเงื่อนไขในการกรองข้อมูลที่จะแสดงได้

ซึ่งผลที่ได้จากการค้นหาสามารถมีได้มากกว่าหนึ่งเอนทรีซึ่งจะแสดงในส่วนทางด้านซ้ายของเว็บเพจซึ่งจะแสดงเป็นชื่อของเอนทรีที่ตรงกับเงื่อนไขในการค้นหา และเมื่อคลิกถึงจะแสดงรายละเอียดของเอนทรีที่ค้นหามาได้ สามารถแสดงได้ดังรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

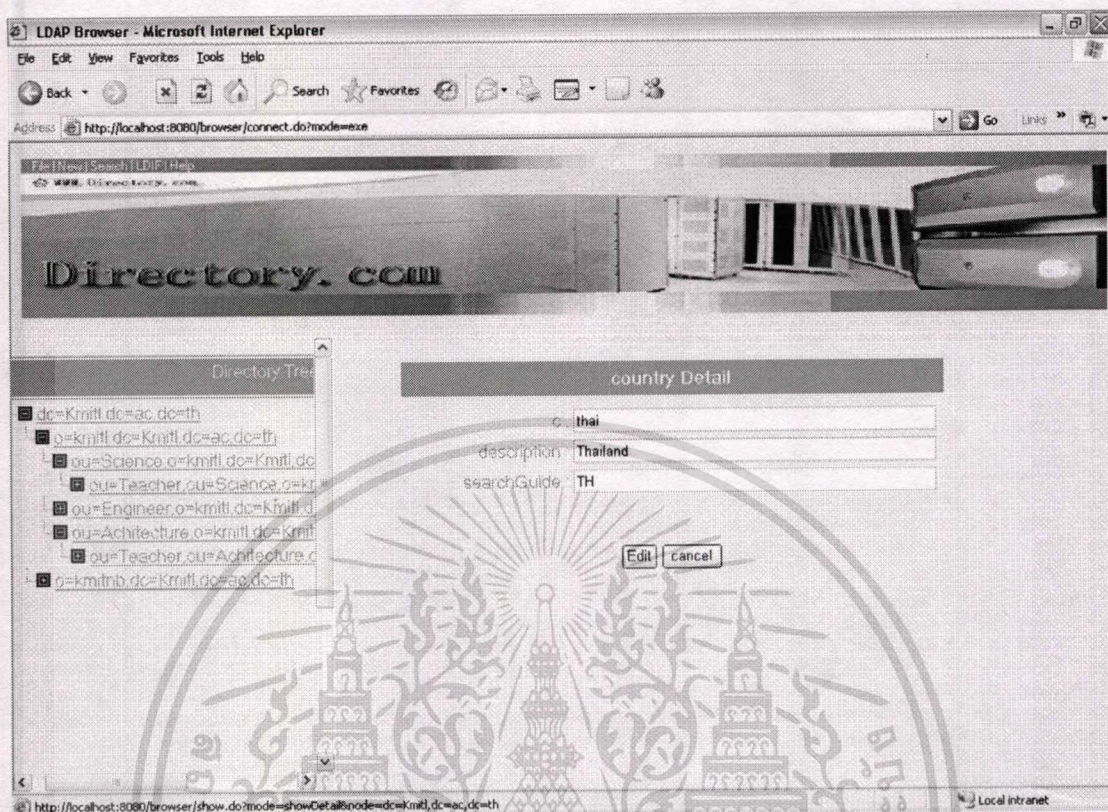


รูปที่ 4.8 หน้าจอการกำหนดเงื่อนไขการค้นหา

4.4.5 การแสดงรายละเอียดของเอนทรีที่ต้องการ

เมื่อผู้ใช้ได้ระบุเงื่อนไขในการค้นหาแล้ว ระบบจะทำการค้นหาข้อมูลในเอนทรีตามที่ต้องการ พร้อมกับนำข้อมูลที่ค้นหาได้มาแสดงในรูปแบบของแผนภาพต้นไม้ที่ด้านซ้ายของหน้าจอ เมื่อผู้ใช้เลือกที่เอนทรีที่ต้องการ รายละเอียดของเอนทรีนี้ก็จะแสดงที่ด้านขวาของหน้าจอ

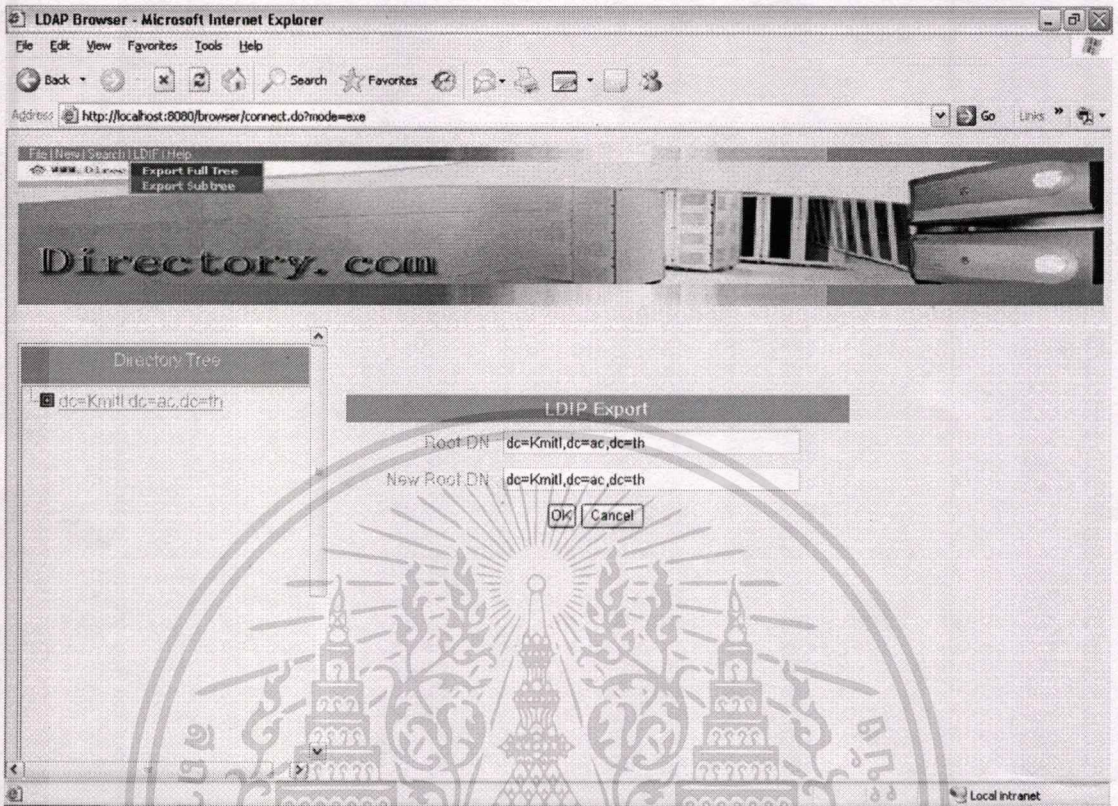
จากหน้าจอนี้ผู้ใช้สามารถทำการแก้ไขข้อมูลในเอนทรี และลบเอนทรีที่เลือกไว้แล้วออกจากไคลเอนต์ได้โดยการคลิกที่ปุ่ม Delete ที่ด้านล่างของหน้าจอ ดังรูปที่ 4.9



รูปที่ 4.9 หน้าจอแสดงรายละเอียดของเอนทรี

4.4.6 การนำข้อมูลในไดเรกทอรีแสดงออกมาในรูปแบบของไฟล์ข้อความ (เมนู LDIF)

ผู้ใช้สามารถนำข้อมูลในไดเรกทอรี (Import) แสดงออกมาในรูปแบบไฟล์ข้อความได้ ซึ่งผู้ใช้สามารถเลือกได้ว่า จะแสดงข้อมูลออกมาทั้งไดเรกทอรีหรือแค่บางส่วนของไดเรกทอรี โดยที่ผู้ใช้จะต้องกำหนดข้อมูล คือ Root DN และ New Root DN ดังรูปที่ 4.10



รูปที่ 4.10 หน้าจอการนำข้อมูลออกมาในรูปแบบไฟล์ข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

5.1 สรุปผลการพัฒนา

โครงการพัฒนาระบบงานนี้เป็นการพัฒนาระบบในการจัดการกับฐานข้อมูลผู้ใช้ตามมาตรฐานโพรโตคอล LDAP โดยใช้ Jbuilder ในการพัฒนาโดยใช้ภาษา java และโดยมีการนำ VmWare มาใช้ในการจำลองเป็นไคลเอนท์เซิร์ฟเวอร์เพื่อใช้ในการติดต่อ ซึ่งโครงสร้างของระบบในการพัฒนานั้นมีการสร้างเป็นหน้าเว็บเพจโดยใช้ jsp เพื่อติดต่อกับโครงสร้างของภาษา java เพื่อสร้างการติดต่อกับไคลเอนท์เซิร์ฟเวอร์โดยโครงสร้างของระบบที่พัฒนา ประกอบด้วยส่วนต่างๆ เพื่อสร้างระบบรายชื่อผู้ใช้ภายในองค์กร และจัดการกับข้อมูลผู้ใช้ เพื่อเพิ่มประสิทธิภาพในการจัดการกับข้อมูลในไคลเอนท์โดยโครงสร้างของระบบที่พัฒนาขึ้นประกอบไปด้วยส่วนต่างๆ ดังต่อไปนี้

1. ส่วนการแสดงผลการทำงานออกทางหน้าเว็บเพจ

คือส่วนที่ทำการแสดงผลเพื่อให้กับผู้ใช้สามารถทำงานได้สะดวกมากขึ้น โดยส่วนนี้ถูกพัฒนาโดยใช้ภาษา Jsp ในการสร้างหน้าเว็บเพจและใช้จาวาสคริปในการจัดการกับความถูกต้องของข้อมูลและมีการนำเอา Struts Framework มาใช้ในการจัดการในการทำงานในแต่ละเว็บเพจและใช้ JavaBean ในการจัดการส่งและรับข้อมูลในฟอร์ม

2. ส่วนการทำงานเพื่อสร้างการติดต่อกับไคลเอนท์เซิร์ฟเวอร์

คือส่วนที่ทำการสร้างการติดต่อกับเซิร์ฟเวอร์ตามมาตรฐานโพรโตคอล LDAP เพื่อให้สามารถจัดการกับข้อมูลในไคลเอนท์ หรือดึงรายละเอียดของเอนทรีได้ โดยใช้ภาษาจาวาในการพัฒนา ติดต่อกับ LDAP API

3. ส่วนในการสร้างรายละเอียดของโครงสร้างต้นไม้

คือส่วนที่ทำการนำข้อมูลของฐานข้อมูลผู้ใช้งานมาสร้างเป็นโครงสร้างต้นไม้ โดยใช้จาวาในการสร้างภาษา Jsp เพื่อแสดงผลออกมาเป็นโครงสร้างต้นไม้

4. ส่วนในการจัดการกับข้อมูลในไคลเอนท์

คือส่วนในการทำงานเพื่อจัดการกับข้อมูลเอนทรีในไคลเอนท์ไม่ว่าจะเป็นการเพิ่ม ลบ หรือแก้ไขข้อมูลในเอนทรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ส่วนในการนำข้อมูลในไคลเรคทอรีออกมาเป็น text ไฟล์

คือส่วนในการทำงานเพื่อที่จะนำข้อมูลรายละเอียดของเอนทรีออกมาเก็บออกเป็น text ไฟล์นามสกุล LDIF และสามารถนำเอาไฟล์ LDIF มาสร้างเป็นโครงสร้างไคลเรคทอรี ในไคลเรคทอรีเซิร์ฟเวอร์อื่นๆได้

6. ส่วนของไคลเรคทอรีเซิร์ฟเวอร์

คือส่วนในการทำงานเป็นไคลเรคทอรีเซิร์ฟเวอร์เพื่อให้ระบบจัดการฐานข้อมูลผู้ใช้สามารถติดต่อเพื่อดึงข้อมูลออกมาใช้ได้ซึ่งสร้างโดยใช้โปรแกรม Vmware ลงระบบปฏิบัติการ FreeBSD ลงโปรแกรม OpenLDAP เพื่อเป็นตัวจัดการกับฐานข้อมูลแบบไคลเรคทอรีพร้อมทั้งสร้างค่าเริ่มต้นในการจัดการกับฐานข้อมูล

ซึ่งผลในการพัฒนาระบบในการจัดการกับฐานข้อมูลผู้ใช้ตามมาตรฐานโปรโตคอล LDAP ได้ดำเนินการบรรลุจุดประสงค์ โดยเป็นระบบที่สามารถอำนวยความสะดวกในการจัดการกับฐานข้อมูลแบบไคลเรคทอรี ผู้ใช้สามารถมองเห็นภาพโครงสร้างของระบบ สะดวกในการค้นหาและจัดการกับรายละเอียดโครงสร้างในระบบ และสะดวกในการสร้างไฟล์ LDIF สามารถสร้างได้อย่างง่ายดาย ซึ่งสามารถพัฒนาระบบโครงสร้างของฐานข้อมูลผู้ใช้ได้มีประสิทธิภาพในอนาคต

5.2 อุปสรรคในการพัฒนา

การพัฒนากระบวนการบริหารฐานข้อมูลผู้ใช้นี้ประสบปัญหาในช่วงแรกในการทำความเข้าใจกับฐานข้อมูลแบบไคลเรคทอรีซึ่งมีโครงสร้างแตกต่างจากฐานข้อมูลทั่วไปและต้องใช้ความรู้ทางด้านเทคโนโลยีหลายๆด้านมาช่วยในการพัฒนา ทำให้การพัฒนาบางครั้งดำเนินแนวทางที่ผิดซึ่งมีการลองผิดลองถูกทำให้ล่าช้าในการพัฒนา และปัญหาที่ประสบในช่วงหลังคือเนื่องจากโครงสร้างของฐานข้อมูลแบบไคลเรคทอรีตามมาตรฐาน LDAP นั้นมีข้อจำกัดและเงื่อนไขที่ซับซ้อนจึงต้องมีการพัฒนาที่ซับซ้อนตามไปด้วย ผู้พัฒนาจึงต้องหาทางออกในการพัฒนาที่ดีที่สุดเพื่อให้เกิดข้อผิดพลาดกับระบบให้น้อยที่สุด

1.3 ข้อเสนอแนะ

ในการพัฒนาระบบจัดการข้อมูลรายชื่อผู้ใช้ตามมาตรฐานโปรโตคอล LDAP นี้เป็นแนวทางในการพัฒนาเทคโนโลยีอื่นในการทำงานกับโปรโตคอล LDAP เพื่อให้สามารถสร้างระบบในการจัดการกับโปรโตคอล LDAP ให้มีประสิทธิภาพมากยิ่งขึ้น และสามารถประยุกต์ใช้ระบบอื่นที่นำโปรโตคอล LDAP มาใช้ในการพัฒนา และระบบที่พัฒนานั้นยังมีข้อจำกัดในอีกหลายด้านที่ผู้พัฒนายังไม่สามารถพัฒนาระบบให้รองรับได้ดังต่อไปนี้คือ

1. การนำเข้าข้อมูลในรูปแบบของ LDIF ไฟล์ซึ่งควรที่ระบบจะสามารถนำเอาไฟล์ LDIF มาสร้างเป็นโครงสร้างของไคลเรคทอรีได้

2. พัฒนาโปรแกรมในด้านการรักษาความปลอดภัย โดยสามารถกำหนดคสิทธิการใช้งานของผู้ใช้งาน และสามารถกำหนดนโยบายรักษาความปลอดภัยต่างๆ ให้กับโปรแกรมได้
3. พัฒนาระบบให้สามารถแสดงรายละเอียดให้สามารถทำความเข้าใจได้ง่ายมากยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

Thuraisingham, Bhavani. 1999. **LDAP Technologies, Techniques, Tools, and Trends.**

Michigan : CRC Press.

Han, Jiawei and Kamber, Micheline. 2001. **LDAP : Concepts and Techniques.**

San Francisco : Morgan Kaufmann.

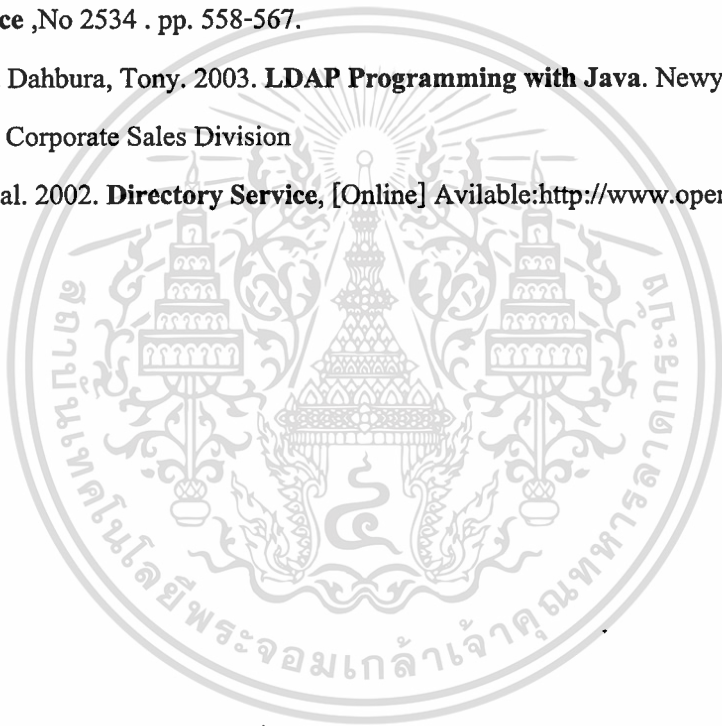
Cooley, R. et al. 1997. "Develop LDAP java implement," **IEEE Tools with Artificial**

Intelligence ,No 2534 . pp. 558-567.

Weltman, Rob and Dahbura, Tony. 2003. **LDAP Programming with Java.** Newyork :Pearson

Education Corporate Sales Division

Sarwar, Badrul. et al. 2002. **Directory Service**, [Online] Available:<http://www.openldap.org/doc.pdf>



ประวัติผู้เขียน

ชื่อผู้เขียน	นาย วัชรสิทธิ์ สิงห์โตทอง
วันเกิด	2 ตุลาคม 2523
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (วิทยาการคอมพิวเตอร์) คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้