ระบบการพิสูจน์ตัวตนเพียงครั้งเดียว

SINGLE SIGN-ON SYSTEM

โดย

วิมลรัตน์ โชติไพศาลกุล

WIMOLRAT CHOTIPAISARNGUL

อาจารย์ที่ปรึกษา

ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2550

SINGLE SIGN-ON SYSTEM

WIMOLRAT CHOTIPAISARNGUL

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2008

หัวข้อ	ระบบการพิสูจน์ตัวตนเพียงครั้งเดียว
นักศึกษา	นางสาววิมลรัตน์ โชติไพศาลกุล
รหัสนักศึกษา	49066513
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

บทคัดย่อ

เมื่อระบบสารสนเทศเข้ามามีบทบาทสำคัญในองค์กรทางธุรกิจ ผู้ใช้งานระบบหรือพนักงานขององค์กรนั้นๆ จำเป็นที่จะต้องมีรหัสผู้ใช้งานเพื่อเข้าใช้งานทรัพยากรตามสิทธิ หน้าที่รับผิดชอบของตน ดังนั้นขั้นตอนการตรวจสอบผู้ใช้งาน การให้สิทธิแก่ผู้ใช้งานของแต่ละระบบสารสนเทศ จึงเป็นขั้นตอนที่สำคัญ เพื่อเป็นการรักษาความปลอดภัยของทรัพยากรและข้อมูลของระบบ รวมถึงในการพัฒนาระบบสารสนเทศ มีความหลากหลายของเทคโนโลยี ต่างแพลตฟอร์ม ทำให้ผู้ใช้หนึ่งคนที่มีสิทธิเข้าใช้งานระบบมากว่าหนึ่งระบบ จำเป็นต้องมีรหัสผ่านมากกว่าหนึ่งชุด ทำให้การจัดการรหัสผู้ใช้งาน/รหัสผ่านมีความยุ่งยากและเกิดความซ้ำซ้อนในการเก็บข้อมูล ดังนั้นจึงมีแนวความคิดในการพัฒนาระบบสารสนเทศในการจัดการรหัสผู้ใช้งาน ตามแนวเทคโนโลยี Single Sign-on ซึ่งเป็นเทคโนโลยีหนึ่งที่ได้รับความสนใจในปัจจุบัน เพื่อลดปัญหาในการจัดการรหัสผู้ใช้งาน/รหัสผ่านให้น้อยลง แต่ยังคงรักษาสมดุลที่ดีระหว่างการรักษาความปลอดภัยและการพัฒนาระบบสารสนเทศ ระบบนี้พัฒนาด้วยภาษาจาวา ใช้ LDAP และ Oracle ในการเก็บข้อมูล สร้างเซอร์วิส เพื่อให้ระบบสารสนเทศอื่นๆ เรียกใช้งานได้ผ่านเว็บเซอร์วิสทำให้เกิดความอิสระของเทคโนโลยีในการพัฒนาระบบสารสนเทศ

Title	Single Sign-on System
Student	Miss. Wimolrat Chotipaisarngul
Student ID.	49066513
Degree	Master of Science
Programme	Information Science
Academic Year	2007
Advisor	Asst.Prof. Dr.Chanboon Sathitwiriyawong

ABSTRACT

Today, the term information technology is more recognizable regards with many aspects of computing and technology. **Single Sign-on** (SSO) is one of the method that has been accepted as the authenticate method of access control solution way out that enables user's authorize once and gain access to the resources of multiple software systems.

In a different IT infrastructure, authentication scheme has been in a complex where user database is not centralized. Single Sign-on suddenly becomes a visible benefit to achieve such an organization wide. To start up with a homogeneous IT infrastructure that authentication scheme has been existed, all users in this infrastructure would have the authentication credentials. It is a solution way out that balancing between how to secure with still improve the infrastructure.

Single Sign-on has been developed by Java, store its user database in a LDAP and Oracle database for the authentication and authorization and been much achieved in the organization wide.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำ และปรึกษา ตลอดจนการตรวจสอบแก้ไข เพื่อให้โครงงานนี้เสร็จสมบูรณ์ จาก ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน

คณาจารย์คณะเทคโนโลยีสารสนเทศทุกๆ ท่าน ที่ได้ให้ความรู้มาโดยตลอด

ขอบคุณเจ้าหน้าที่ประจำคณะเทคโนโลยีสารสนเทศทุกท่าน ที่อำนวยความสะดวกในด้านต่างๆ

ขอบคุณพี่ๆ ที่ทำงานที่ให้ โอกาสและสนับสนุนการเรียนมาตลอดจน คำแนะนำต่างๆ

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดา และพี่ๆ ที่ให้กำลังใจมาโดยตลอด

วิมลรัตน์ โชติไพศาลกุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เมื่อระบบสารสนเทศได้ขยายเข้ามามีบทบาทในระบบธุรกิจ ทำให้มีความต้องการในการพัฒนาระบบสารสนเทศในรูปแบบต่างๆ ทั้งระบบสารสนเทศภายในองค์กรระบบสารสนเทศเพื่อใช้งานทั่วไป ระบบสารสนเทศที่พัฒนาขึ้นมีความแตกต่างกันทั้งเทคโนโลยีและแพลตฟอร์มที่ใช้ในการพัฒนาบางองค์กรมีระบบสารสนเทศทั้งภายใน ภายนอกอยู่มากกว่าหนึ่งระบบ และแต่ละระบบมีระบบการพิสูจน์ตัวตนแยกกันไปตามระบบงานนั้นๆ มีความยากง่ายและความซับซ้อนที่แตกต่างกัน ตามความสำคัญมากน้อยของทรัพยากรภายในระบบ ทำให้ผู้ใช้งานระบบจำเป็นต้องมีรหัสผู้ใช้งาน/รหัสผ่านที่แตกต่างกัน ตามข้อกำหนดด้านความปลอดภัยของแต่ละระบบและเมื่อมีผู้ใช้งานระบบจำนวนเพิ่มมากขึ้น ปัญหาที่ตามมาก็คือการจัดการรหัสผู้ใช้งาน/รหัสผ่าน ทำให้เกิดความซ้ำซ้อนและกระจัดกระจายของการทำงานรวมถึงการเก็บรวบรวมข้อมูล การแก้ไข เปลี่ยนแปลงข้อมูลทำได้ยาก ค่าใช้จ่ายในการบำรุงรักษาที่สูงขึ้น หลายองค์กรเริ่มศึกษาถึงกระบวนวิธีการต่างๆ ในการลดความซ้ำซ้อนในเรื่องดังกล่าว รวมถึงการเพิ่มประสิทธิภาพในการจัดเก็บข้อมูล การบำรุงรักษาข้อมูล และการรักษาความปลอดภัยของทรัพยากรที่สอดคล้องกับแนวทางในการพัฒนาระบบสารสนเทศต่างๆ ทำให้เทคโนโลยี Single Sign-On (SSO) ซึ่งเป็นเทคโนโลยีการเก็บข้อมูลสิทธิ์ของผู้ใช้ระบบที่ได้รับอนุญาตสำหรับเข้าใช้งานระบบสารสนเทศและจะทำการตรวจสอบพิสูจน์ตัวตนเพียงครั้งเดียว ได้รับความนิยมในการพัฒนาร่วมกับระบบงานต่างๆ อย่างมากในปัจจุบัน โดยมีการปรับปรุงรูปแบบในการพัฒนาระบบ Single Sign-On (SSO) ให้เหมาะสมกับแนวทาง ทางด้านการรักษาความปลอดภัยของแต่ละองค์กรหรือแต่ละระบบสารสนเทศ

รายงานฉบับนี้จึงเสนอรูปแบบในการพัฒนาระบบ Single Sign-On (SSO) ซึ่งเป็นระบบพิสูจน์ตัวตนและการกำหนดสิทธิให้แก่ผู้ใช้งานสำหรับองค์กร คำนึงถึงความปลอดภัยในการเข้าถึงข้อมูลและการใช้งานระบบสารสนเทศอย่างสมบูรณ์นำเอาเทคโนโลยี ทางด้านความปลอดภัยโพรโตคอล LDAP และการเก็บข้อมูลใน LDAP Directory ร่วมกับเทคโนโลยีเว็บเซอร์วิสโดยใช้ภาษา XML ในการแลกเปลี่ยนข้อมูลกับระบบต่างๆ ทำให้เกิดความอิสระทางด้านเทคโนโลยีและแพลตฟอร์ม ในการติดต่อสื่อสารกับระบบ Single Sign-On (SSO) ที่พัฒนาขึ้น

การพัฒนาระบบ Single Sign-On (SSO) เป็นพัฒนาโมดูลของระบบสารสนเทศ เพื่อให้ระบบสารสนเทศต่างๆ เรียกใช้บริการในการพิสูจน์ตัวตนและการกำหนดสิทธิให้แก่ผู้ใช้งานระบบ ดังนั้นนอกจากความปลอดภัยในการเก็บข้อมูลต่างๆ ของผู้ใช้งาน ระบบ Single Sign-On ต้องคำนึง

ถึงการสื่อสาร การแลกเปลี่ยนข้อมูลระบบ ระหว่างระบบ Single Sign-On (SSO) กับระบบสารสนเทศต่างๆ โดยเป็นปัญหาหลักที่ทำให้ระบบสารสนเทศหลายๆ ระบบที่มีผู้ใช้งานระบบกลุ่มเดียวกันระบบมีความซ้ำซ้อน ในการเก็บข้อมูลและความยุ่งยากในการบำรุงรักษาให้ข้อมูลถูกต้องอยู่เสมอ จึงได้นำเทคโนโลยีเว็บเซอร์วิสเข้ามาใช้งานในการแลกเปลี่ยนข้อมูลระหว่างระบบสารสนเทศ การติดต่อของใช้บริการ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

โครงงานฉบับนี้มุ่งหวังเพื่อการศึกษาและพัฒนาเว็บเซอร์วิสสำหรับการพิสูจน์ตัวตนและการกำหนดสิทธิให้แก่ผู้ใช้งาน เพื่อให้การพัฒนาโมดูลในการพิสูจน์ตัวตนและการกำหนดสิทธิให้แก่ผู้ใช้งานมีความเป็นอิสระ เพิ่มความเป็นอิสระต่อเทคโนโลยี (Loosely Coupled) สามารถขอใช้บริการได้จากระบบสารสนเทศที่พัฒนาขึ้นด้วยเทคโนโลยีภาษาที่แตกต่างกัน ซึ่งนักพัฒนาระบบสารสนเทศขององค์กร ไม่จำเป็นต้องพัฒนาโมดูลการพิสูจน์ตัวตนและการกำหนดสิทธิให้แก่ผู้ใช้งาน ลดความซ้ำซ้อนของการพัฒนาระบบ ความซ้ำซ้อนในการเก็บข้อมูล สะดวกในการบำรุงรักษาข้อมูลให้ถูกต้อง ความมั่นคงและความน่าเชื่อถือได้ในการรักษาข้อมูลทรัพยากรของระบบสารสนเทศขององค์กรให้เหมาะสมกับแนวทาง ทางด้านการรักษาความปลอดภัยของแต่ละองค์กร

1.3 สมมติฐานของการศึกษา

ข้อดีของระบบ Single Sign-On (SSO) คือทำการพิสูจน์ตัวตนและการกำหนดสิทธิให้แก่ผู้ใช้งานระบบ โดยผู้งานระบบมีบัญชีรายชื่อผู้ใช้งานเพียงหนึ่งชุด สามารถใช้รายชื่อนั้นเข้าใช้งานระบบของระบบสารสนเทศต่างๆ ภายในองค์กรเข้าถึงทรัพยากรระบบได้ตามสิทธิหน้าที่งานที่องค์กรกำหนด เพื่อความปลอดภัยของทรัพยากรและองค์กรสามารถจัดการการกำหนดสิทธิการเข้าถึงทรัพยากรระบบ หรือการเข้าใช้งานระบบสารสนเทศไว้ในระบบ Single Sign-On (SSO) เพียงที่เดียว

ข้อดีของเทคโนโลยีเว็บเซอร์วิส คือมีความเป็นอิสระต่อแพลตฟอร์ม เพื่อใช้ในการแลกเปลี่ยนข้อมูลกันระหว่างระบบสารสนเทศ ซึ่งระบบสารสนเทศที่เป็นผู้ร้องขอไม่จำเป็นต้องรู้การทำงาน และข้อมูลภายใน ก็สามารถเรียกใช้บริการได้ ร่วมถึงในการแก้ไข ปรับปรุงเปลี่ยนแปลงการทำงานภายในไม่มีผลกระทบกับระบบสารสนเทศที่มาขอใช้บริการอีกด้วย

จากข้อดีของระบบ Single Sign-On (SSO) และเทคโนโลยีเว็บเซอร์วิส จึงมีแนวคิดในการพัฒนาระบบ Single Sign-On (SSO) เป็นบริการที่พัฒนาบนเทคโนโลยีเว็บเซอร์วิส ให้สามารถเรียกใช้บริการได้จากระบบสารสนเทศต่างๆ ได้สะดวกและยังคงรักษาความปลอดภัยของทรัพยากรระบบของระบบสารสนเทศต่างๆ ได้อย่างเหมาะสม

1.4 ขอบเขตของการพัฒนาระบบการเข้าระบบเพียงครั้งเดียว

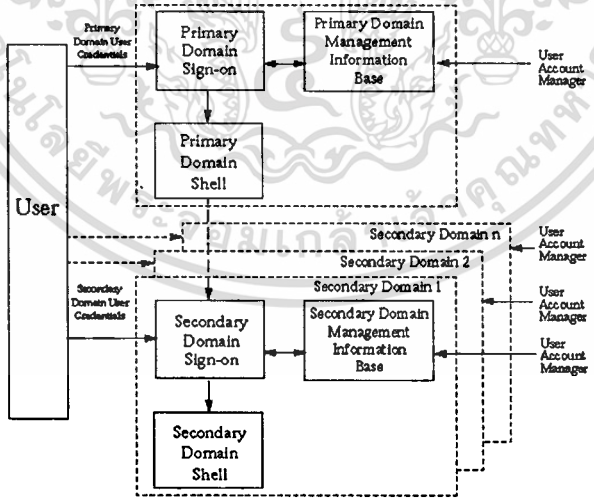
ระบบการพิสูจน์ตัวตนครั้งเดียว พัฒนาโดยใช้ภาษาจาวาเป็นหลัก แบ่งการพัฒนาออกเป็นส่วนประกอบด้วยเว็บเซอร์วิสเป็นเซอร์วิสในการพิสูจน์ตัวตนและกำหนดสิทธิให้กับรายชื่อผู้ใช้งานระบบคนนั้นๆ เซอร์วิสจะทำการติดต่อกับ LDAP เซิร์ฟเวอร์ เพื่อทำการตรวจสอบข้อมูลรายชื่อผู้ใช้งาน/รหัสผู้ใช้งานว่ามีความถูกต้องและทำการติดต่อกับฐานข้อมูล Oracle ที่ใช้เก็บข้อมูลสิทธิของผู้ใช้งาน การร้องขอใช้บริการหรือระบบสารสนเทศต่างๆ ที่พัฒนาด้วยภาษาอื่นๆ ที่รองรับการทำงานของเว็บเซอร์วิส สามารถร้องขอใช้บริการได้ ส่วนที่สองคือเว็บแอพพลิเคชัน เซิร์ฟเวอร์ เป็นส่วนเชื่อมประสานกันผู้ใช้งานเพื่อกำหนดสิทธิการเข้าถึงให้แต่ละบัญชีรายชื่อ โดยทำงานต่างๆจะเป็นลักษณะของการเรียกใช้ เซอร์วิส ในส่วนของเว็บเซอร์วิส เพื่อจัดการและจัดเก็บข้อมูลระบบของผู้ใช้งานระบบทั้งหมด มีการเชื่อมต่อ LDAP เซิร์ฟเวอร์ ฐานข้อมูลแบบรีเลชันนอลผ่าน เซอร์วิสของเว็บเซอร์วิส เท่านั้น เพื่อความปลอดภัยในการจัดเก็บข้อมูล ในระบบส่วนนี้จะเป็นการเพิ่มข้อมูลและแก้ไขจัดการข้อมูลผู้ใช้งาน การตั้งค่าระบบต่าง ๆ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่ใช้ในการพัฒนาระบบ ได้แก่ Single Sign-on, LDAP, Ajax และ เทคโนโลยีเว็บเซอร์วิส ซึ่งเนื้อหาทั้งหมดนี้มีความสำคัญต่อการพัฒนาระบบการเข้าระบบเพียงครั้งเดียวผ่านเว็บเซอร์วิส

2.1 Single Sign-On (SSO)

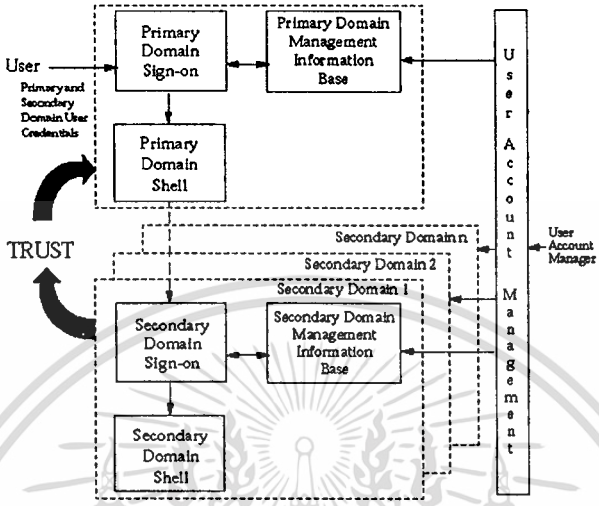
เทคโนโลยีการเข้าระบบเพียงครั้งเดียวหรือการตรวจ สอบตัวตนเพียงครั้งเดียว (Single Sign-on : SSO) จัดอยู่ในกลุ่มของเทคโนโลยีเซกเคียวรีตี้อินฟาสตรักเจอร์ (Security infrastructure คือ ส่วนของการรักษาความปลอดภัยที่แอพพลิเคชันไม่จำเป็นที่จะต้องเปลี่ยนแปลงตรรกะของแอพพลิเคชัน เมื่อมีการเปลี่ยนแปลงหรือเพิ่มเติมวิธีการตรวจสอบสิทธิ์ของผู้ใช้งาน สามารถอิมพลีเมนต์เพิ่มความสามารถในส่วนนี้ได้อย่างเป็นอิสระ) เป็นเทคโนโลยีที่มีประโยชน์มากเมื่อนำมาประยุกต์ใช้ ร่วมกับระบบสารสนเทศหรือแอพพลิเคชันต่างๆ ปกติหากไม่มีการนำเทคโนโลยี Single Sign-on มาใช้ เมื่อใช้งานระบบสารสนเทศที่มีหลายๆ แอพพลิเคชันอยู่ด้วยกัน ผู้ใช้งานจะต้องทำการพิสูจน์ตัวตน การเข้าใช้งานระบบหลายๆ ครั้งดังแสดงในรูปที่ 2.1



รูปที่ 2. 1 การเข้าใช้งานระบบของผู้ใช้ต่อหลายๆ ระบบ

แต่เมื่อมีการพัฒนาเทคโนโลยี Single Sign-on ขึ้นมาจะช่วยให้ผู้ใช้งานไม่จำเป็นต้องทำการพิสูจน์ตัวตนหลายๆ ครั้งอีกต่อไป เนื่องจากระบบจะเก็บข้อมูลสิทธิ์ของผู้ใช้ที่ได้รับอนุญาตสำหรับผู้ใช้แต่ละคนและจะทำการตรวจสอบพิสูจน์ตัวตนเพียงครั้งเดียว ดังแสดงตามรูปที่ 2.2 จาก

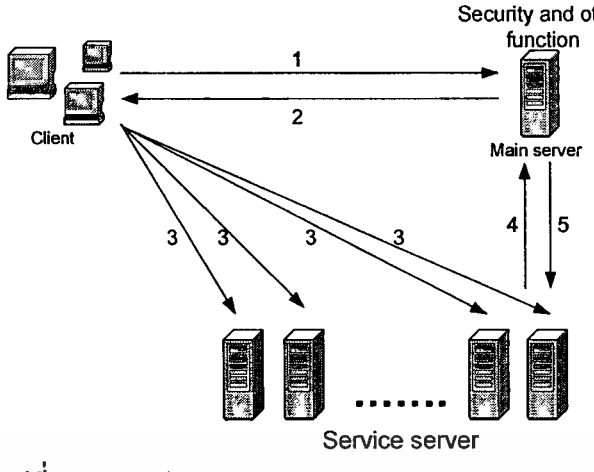
นั้นผู้ใช้สามารถเข้าถึงทรัพยากรระบบต่างๆ ได้ตามสิทธิ์เท่าที่ได้รับอนุญาตไว้และนอกจากจะเป็นการช่วยให้ผู้ใช้ ระบบสามารถเข้าใช้งานระบบสารสนเทศได้ง่ายและรวดเร็วขึ้นแล้ว เทคโนโลยี Single Sign-on ยังจะช่วยให้ผู้ดูแลระบบสามารถจัดการกับรหัสผู้ใช้งาน/รหัสผ่านได้ง่ายและเป็นระบบมากขึ้น



รูปที่ 2. 2 การเข้าใช้งานระบบของผู้ใช้ต่อหลายๆ ระบบที่มีการนำเทคโนโลยี SSO มาใช้งาน

การออกแบบระบบ Single Sign-On (SSO) สามารถแบ่งระบบได้เป็นส่วนหลักๆ 3 ส่วน คือ เมนเซิร์ฟเวอร์, เซอร์วิสเซิร์ฟเวอร์ และไคลเอนท์

เมนเซิร์ฟเวอร์เป็นส่วนของการควบคุมการเข้าใช้งาน บริการต่างๆ ของเซอร์วิสเซิร์ฟเวอร์ เช่น การเข้าใช้งานไฟล์เซิร์ฟเวอร์ ระบบเมล์เซิร์ฟเวอร์เป็นต้น โดยในส่วนนี้จะมีการเพิ่มความสามารถพิเศษต่างๆ ให้เหมาะสมกับการใช้งานส่วนในการทำงานของเซอร์วิสเซิร์ฟเวอร์ จะให้ความไว้วางใจในข้อมูลที่บ่งบอกถึงการระบุตัวตนของผู้ใช้งานระบบและข้อมูลในการเข้าถึงทรัพยากรระบบจากเมนเซิร์ฟเวอร์ โดยทั้งสองส่วนจะมีการร่วมใช้งานคีย์ ประกอบด้วย Public key (เป็นคีย์ที่ใช้งานร่วมกันทั้งสามส่วน) และ Private key สุดท้ายในส่วนของไคลเอนท์ จะเป็นการร้องขอการเข้าใช้งานบริการหรือทรัพยากรต่างๆ ที่สามารถเข้าถึงได้จะทำการตรวจสอบสิทธิ์และระบุตัวตนจากเมนเซิร์ฟเวอร์ก่อนเสมอ ดังแสดงตามรูปที่ 2.3 โดยจะเห็นว่า เมื่อไคลเอนท์ต้องการเข้าใช้งานระบบจะต้องการทำติดต่อกับเมนเซิร์ฟเวอร์เป็นอันดับแรก ดังนั้นในการพัฒนาการทำงานของเมนเซิร์ฟเวอร์จะมีความแตกต่างกับไปในการพัฒนาของแต่ละองค์กร



รูปที่ 2. 3 สถาปัตยกรรมของระบบ Single Sign-On (SSO)

ในปัจจุบันเราจะเห็นการนำเทคโนโลยี Single Sign-on มาใช้งานร่วมกับระบบงานต่างๆ อย่างกว้างขวางและมีความหลายหลาก ขึ้นอยู่กันสิ่งแวดล้อมขององค์กรหรือระบบงานที่ต้องการ โดยยังคงรักษาสมดุลที่ดีระหว่างการรักษาความปลอดภัยและประสิทธิภาพในการทำงาน ไม่ว่าจะเป็นระบบสารสนเทศที่ใช้ภายในองค์กรขนาดใหญ่ ซึ่งประกอบไปด้วยแอพพลิเคชันหลายแอพลิเคชัน ที่มีความแตกต่างกันในเรื่องของแฟลตฟอร์ม ระบบสารสนเทศออนไลน์ที่มีการเข้าถึงข้อมูลจากต่างพื้นที่หรือไม่ว่าจะเป็นระบบปฏิบัติการหลักทั้งไมโครซอฟต์และซัน เช่น Microsoft Passport, Sun Liberty Alliance, Windows 2000 และ 2003 (Kerberos authentication)

2.2 Lightweight Directory Access Protocol: LDAP

โพรโทคอลบนระบบเครือข่าย ที่ทำงานร่วมกับ Directory Service เป็นตัวกลางในการติดต่อระหว่าง ไดเร็คทอรี่ไคลเอนท์และ ไดเร็คทอรี่เซิร์ฟเวอร์ให้มีการทำงานอย่างสะดวก รวดเร็วและลดปัญหาการติดต่อกันในหลายๆ แพลตฟอร์มที่มีระบบการทำงานที่แตกต่างอยู่ในระดับแอพพลิเคชันจะใช้โพรโทคอล OSI Network ในการติดต่อสื่อสารและมีการทำงานอยู่ในโพรโทคอล TCP/IP เพื่อการเข้ารหัสของ ไดเร็คทอรี่ไคลเอนท์ ให้สามารถการนำข้อมูล เข้า –ออก จากไดเร็คทอรี่ ได้ง่ายและมีความปลอดภัยสูง ด้วยมาตรฐาน LDAP โดย LDAP ถูกออกแบบและพัฒนาขึ้นโดยบริษัท Netscape ร่วมกับทีมวิจัยจากมหาวิทยาลัยมิชิแกน วัตถุประสงค์เพื่อนำมาตรฐาน X.500 มาใช้งานจริงได้บนโพรโทคอล TCP/IP โดย LDAP ได้ถูกปรับปรุงขึ้นจากการนำเอามาตรฐาน X.500 มาพัฒนา โดยยึดถือหลักการเดิมของ X.500 ถึง 90% และปัจจุบัน LDAP ได้เข้ามา กลายเป็นที่รู้จัก ซึ่งถูกสร้างมาตรฐานโดย IETP (RFC 4510) โดยในการพัฒนาระบบ Single sign-on (SSO) จะนำ LDAP มาใช้ในส่วนของเมนเซิร์ฟเวอร์

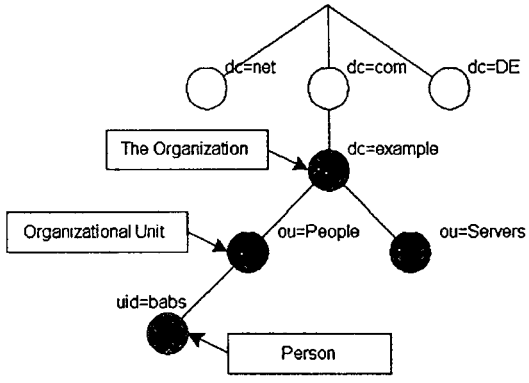
ไดเร็คทอรี่เซิร์ฟเวอร์ในโพรโทคอล LDAP มีระบบการทำงานที่เป็นการบริหารงานจากส่วนกลางและในส่วนกลางนี้จะมีการแบ่งหน้าที่การทำงานออกเป็นส่วนๆ เมื่อผู้ใช้ต้องการเข้ามาใช้งานระบบ ระบบจะทำการส่งความต้องการไปตรวจสอบและการควบคุม (Bind) ไปที่ LDAP เซิร์ฟเวอร์ จากนั้น ค้นหาข้อมูลของผู้ใช้คนนั้นๆ ส่งผลลัพธ์กลับไปยัง LDAP ไคลเอนท์ เพื่อทำการติดต่อและส่งผลที่ได้กลับสู่แอพพลิเคชันระบบ

ไดเร็คทอรี่ทำงานคล้ายกับฐานข้อมูลหรือกล่าวได้ว่าไดเร็คทอรี่เป็นฐานข้อมูลชนิดพิเศษเหมาะแก่การเก็บข้อมูล ที่เป็นวัตถุมีการจัดลำดับขั้นของข้อมูล เน้นการเข้าถึงและการค้นหาง่ายต่อการเพิ่มคุณลักษณะของข้อมูล (เปรียบได้กับการเพิ่มคอลัมน์ในฐานข้อมูลซึ่งสามารถทำได้ง่ายกว่า) แต่ไม่เหมาะสำหรับการเก็บข้อมูลที่มีการแก้ไขหรือเปลี่ยนแปลงข้อมูลบ่อย (Dynamic Data) ไม่สนับสนุนภาษาเอสคิวแอล ในการเข้าถึงข้อมูลของไดเร็คทอรี่ ใช้ภาษาฉพาะที่ช่วยให้เข้าถึงข้อมูลได้อย่างเร็วขึ้น ดังนั้นผู้ออกแบบระบบต้องคำนึงถึงและเข้าใจในข้อจำกัดของความแตกต่างทั้งสองเพื่อที่จะได้แบ่งแยกได้ถูกว่าข้อมูลชนิดไหนควรจะเก็บลงฐานข้อมูลหรือไดเร็คทอรี่ ข้อมูลที่จะทำการเก็บลงจะต้องมีการตรวจสอบความถูกต้องด้วย และด้วยคุณสมบัติพิเศษของไดเร็คทอรี่ ทำให้ไดเร็คทอรี่มีการใช้งานอย่างไม่จำกัดเตรียมรับกับสถานการณ์ที่ผู้ใช้งานเข้ามาใช้งานมากๆ เหมาะสำหรับการดึงข้อมูลมากกว่าการแก้ไขปรับปรุงข้อมูล เครื่องเซิร์ฟเวอร์ที่ให้การบริการอยู่ก็สามารถรองรับการทำงานได้และสามารถรองรับการทำงานจากไคลเอนท์ได้หลายๆ แพลตฟอร์ม ได้พร้อมๆ กัน

การพัฒนาโมดูลที่ใช้ในการติอต่อกับ LDAP ซึ่งเป็นมาตรฐานกลางนั้นได้มีการพัฒนา APIs เพื่อติดต่อกับไดเร็คทอรี่เซอร์วิส โดยไม่ต้องทราบวิธีการเข้าถึงโดยละเอียด เช่น โครงสร้างของไดเร็คทอรี่หรือชนิดของข้อมูล (Data Type) ภายในหรือไม่จำเป็นต้องปรับแก้ตรรกะของแอพพลิเคชันใหม่ เป็นต้น เช่น ภาษาซีใช้ได้กับ Netscape's java SDK, Sunsoft's JNDI และ Microsoft Active Directory Service interface (ADSI) , จาวา และ พีเอชพี เป็นต้น

2.2.1 LDAP Naming

การอ้างถึงข้อมูลของ LDAP จะมีโครงสร้างคล้ายกับระบบแฟ้มข้อมูลแบบลำดับขั้นของ UNIX ทุกโหนดในลำดับขั้นจะมีข้อมูลรวมไปถึงโหนดที่เป็นโหนดราก (root) ด้วย โหนดหนึ่งโหนดใน LDAP เปรียบได้กับข้อมูลหนึ่งข้อมูล โดยแต่ละ เอ็นทรีสามารถระบุความเป็น uniquely โดยใช้ distinguished name (DN) DN จะบอกได้แน่ชัดว่าเอ็นทรีอยู่ตำแหน่งไหนในไดเร็คทอรี่ ข้อมูลแบบลำดับขั้น (hierarchical) ถูกนำเสนอโดย directory information tree (DIT) ดังรูปที่ 2.4



รูปที่ 2. 4 การเก็บข้อมูลของ LDAP

2.2.2 LDAP Schema

เป็นกลุ่มของข้อมูลในแต่ละโหนดในฐานข้อมูลประกอบ ด้วยวัตถุหนึ่งประเภทหรือมากกว่า แต่ละประเภทเรียกว่า objectClass แต่ละ objectClass สามารถขยายมาจาก schema ที่มีอยู่แล้วได้

ตัวอย่าง LDAP Schema

objectclass (2.5.6.2 NAME 'country'
DESC 'RFC2256: a country'
SUP top STRUCTURAL
MUST c
MAY (searchGuide $ description))

objectClass ที่ชื่อว่า country โดยได้ขยายมาจาก scheme top จะต้องมีข้อมูล c และอาจจะมีข้อมูล searchGuides หรือ description ก็ได้

2.2.3 LDAP Function

การตรวจสอบและการควบคุม (Authentication and Control)

- Bind การเชื่อมต่อกับ LDAP directory โดยไคลเอนท์จะแสดงข้อมูลของตัวเอง เพื่อขอรหัสผู้ใช้งาน/รหัสผ่านโดยจะส่งค่า TRUE เมื่อทำการเชื่อมต่อสำเร็จและ FALSE เมื่อมีข้อผิดพลาด
- Unbind การยกเลิกการติดต่อกับโพรโทคอล ซึ่งเป็นการสิ้นสุดการทำงานของ LDAP โดยจะส่งค่า TRUE เมื่อทำการยกเลิกการเชื่อมต่อสำเร็จและ FALSE เมื่อมีข้อผิดพลาด
- Abandon เป็นการแสดงว่าจะไม่มีการทำงานนั้น ๆ ต่อโดยจะส่ง Message ID ไปยัง LDAP เพื่อทำการยกเลิกการทำงาน

การค้นหา (Query)

Search การค้นหาหรือการอ่านข้อมูล โดยการระบุค่าของ attribute ของข้อมูลถ้าไม่พบจะแสดงข้อมูลที่ใกล้เคียง Compare entry เป็นการเปรียบเทียบข้อมูลที่ต้องการค้นหากับคำที่มีความหมายใกล้เคียงกัน

การปรับปรุง (Update)

- Add การเพิ่มข้อมูลเอ็นทรีลงในไดเรกทอรี่
- Delete การลบข้อมูลเอ็นทรีจากไดเรกทอรี่
- Modify การปรับปรุงข้อมูลเอ็นทรีที่มีอยู่

LDAP เป็นมาตรฐานที่ได้รับการยอมรับอย่างกว้างขวางมี Application Vendor อยู่หลายราย อาทิ OpenLDAP, IBM, Oracle, Microsoft, ไดเร็คทอรี่เซิร์ฟเวอร์ ของ Netscape, Active Directory (AD) ของ Microsoft, Novell Directory Services (NDS) ของ Novell, Sun Directory Services (SDS) ของ Sun และ Internet Directory เซิร์ฟเวอร์ (IDS) ของ Lucent ซึ่งมี OpenLDAP เป็น LDAP อิมพรีเมนเตชั่นแบบโอเพนซอร์สที่ได้รับความนิยมสูง มีชุดซอฟต์แวร์ที่แจกจ่ายให้ดาวน์โหลดพร้อมเอกสารการติดตั้ง (www.openldap.org) ดังนั้นรายงานฉบับนี้ จึงเลือกอธิบายรายละเอียดของ OpenLDAP องค์ประกอบ และหน้าที่การทำงานขององค์ประกอบนั้น

2.2.4 OpenLDAP

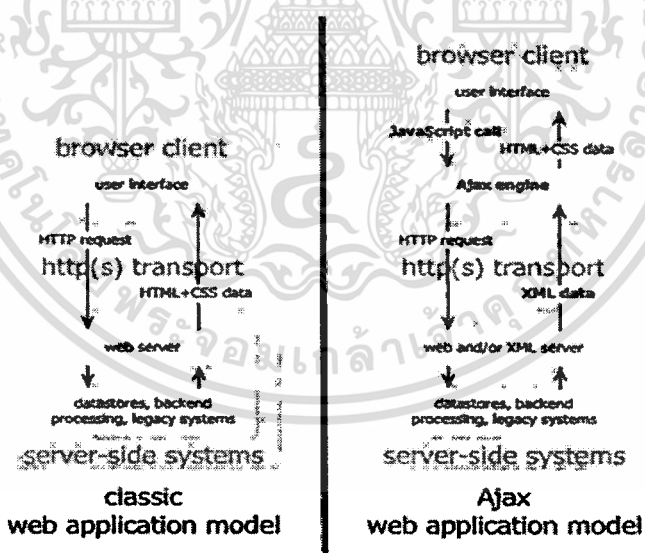
ซอฟต์แวร์โอเพนซอร์สที่พัฒนาจากพื้นฐาน Lightweight Directory Access Protocol (LDAP) ประกอบด้วย lsapd คือ stand-alone LDAP เซิร์ฟเวอร์, slurpd คือ stand-alone LDAP Replication เซิร์ฟเวอร์, Library

- Slapd เป็นเดมอน stand-alone LDAP รับการเชื่อมต่อผ่านโพรโทคอล LDAP ผ่านพอร์ตที่ได้ติดตั้ง (พอร์ตมาตรฐาน คือ 389) และการตอบสนองต่อ LDAP operation ที่ได้รับ
- slurpd เป็นเดมอน stand-alone LDAP Replication เซิร์ฟเวอร์ ทำงานในเรื่องของการทำซ้ำของข้อมูลระหว่าง LDAP เซิร์ฟเวอร์ เช่น เมื่อมีการติดตั้งให้ LDAP เซิร์ฟเวอร์ 2 เซิร์ฟเวอร์ ขึ้นไป แล้วตั้งให้ เซิร์ฟเวอร์ 2 เซิร์ฟเวอร์ นั้น มีการจัดการทำซ้ำข้อมูลระหว่างกันตลอดเวลา เมื่อทำการแก้ไข/ปรับเปลี่ยนข้อมูลที่เครื่องแม่เครื่องลูกก็จะการแก้ไข/ปรับเปลี่ยนข้อมูลนั้นๆ ตามไปด้วย
- Library เป็นกลุ่มคำสั่งที่ใช้ในการติดต่อผ่านโพรโทคอล LDAPโดยการนำเอา openLDAPไปพัฒนาร่วมกันแอพพลิเคชันระบบต่างๆ นั้นก็มี APIs เพื่อนำไปใช้งาน เช่น

- JLDAP-LDAP ไลบรารีเพื่อใช้ในการอิมพลีเมนต์แอพพลิเคชันในการเข้าถึง จัดการ แก้ไข/ปรับปรุง และค้นหาข้อมูลในไดเรคทอรี่
- JDBC-LDAP – สำหรับจาวาแอพพลิเคชันใช้ในการเข้าถึงข้อมูลในไดเร็คทอรี่

2.3 Ajax (Asynchronous JavaScript And XML)

AJAX (Asynchronous JavaScript And XML) ซึ่งหมายถึงการทำงานร่วมกันของ JavaScript และ XML แบบ Asynchronous มีหลักการทำงาน 2 ประเด็น คือ การปรับปรุงหน้าจอแบบบางส่วน และการติดต่อสื่อสารกับ เซิร์ฟเวอร์ โดยใช้หลักการ Asynchronous ทำให้ผู้ใช้ไม่ต้องหยุดการทำงาน เพื่อรอการประมวลผลจาก เซิร์ฟเวอร์ รวมถึงการโหลดและการรีเฟรชหน้าจอของบราวเซอร์ทางฝั่ง ไคลเอ็นท์มีการใช้ Ajax โดยการเพิ่มเลเยอร์ระหว่าง บราวเซอร์ผู้ใช้งานกับเซิร์ฟเวอร์แสดงดังรูป 2.5 ทำให้ผู้ใช้สามารถทำงานได้โดยไม่ต้องรอไคลเอ็นท์ติดต่อไปยังเซิร์ฟเวอร์ รวมถึงการโหลดและการรีเฟรชหน้าจอทั้งหมดด้วย ดังนั้นผู้ใช้สามารถใช้งานเว็บแอพพลิเคชันหน้านั้นๆ ได้อย่างมีประสิทธิภาพมากขึ้น AJAX จึงไม่ใช่เทคโนโลยีใหม่ในตัวของมันเอง แต่เป็นการผสมผสานเทคโนโลยีในปัจจุบันหลายๆ ตัวเข้ามารวมกัน JavaScript, DHTML, XML, Css, Dom และ XMLHTTPRequest



รูปที่ 2. 5 โครงสร้างการทำงานเว็บแอพพลิเคชันแบบใช้ Ajax เทียบกับแบบเดิม

Ajax engine ทำหน้าที่เป็นตัวกลางระหว่าง ไคลเอนท์และเซิร์ฟเวอร์ ฉะนั้นเมื่อ ไคลเอนท์มีคำร้องขอใช้บริการ(request) แทนที่จะส่ง HTTP request ไปยังเซิร์ฟเวอร์ โดยตรงไคลเอ็นท์จะส่ง JavaScript ไปยัง Ajax engine เพื่อโหลดข้อมูลที่ผู้ใช้งานต้องการและหาก Ajax engine ต้องการข้อมูลเพิ่มเติมในการตอบสนองต่อผู้ใช้งาน Ajax engine จะส่ง request ไปยังเซิร์ฟเวอร์โดยใช้ XML

เทคนิค AJAX เป็นการผสมผสานกันของ

- XHTML (หรือ HTML), CSS สำหรับตกแต่ง และจัดระเบียบข้อมูล ในส่วนแสดงผล
- DOM และ JavaScript หรือ JScript ซึ่งเป็น client-side scripting language เอาไว้แสดงผลแบบไดนามิก และจัดการตอบสนองกับการแสดงผล
- XMLHttpRequest เป็น object ที่ใช้สำหรับแลกเปลี่ยนข้อมูลกับเว็บเซิร์ฟเวอร์ แบบไม่ต่อเนื่องกัน (Asynchronous) ในบางสถานการณ์ object ประเภท IFrame จะถูกใช้แทนการใช้ XMLHttpRequest
- XML ที่จะใช้เป็นสื่อกลางในการรับข้อมูลมาจากเซิร์ฟเวอร์ (ใช้แบบไหนก้อได้ เช่น HTML, Text, JSON หรือแม้กระทั่ง EBML)
- JavaScript คือเครื่องมือที่รวมทุกสิ่งทุกอย่างเข้าด้วยกัน

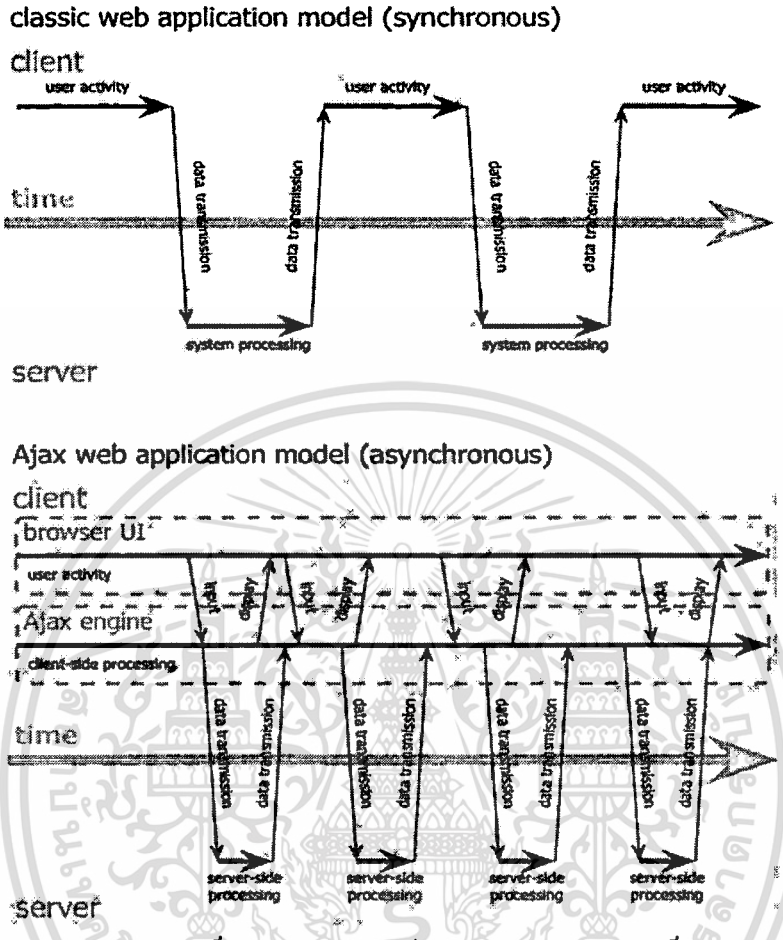
การทำงานของ Ajax

Ajax เป็นการใช้จาวาสคริปต์ที่ฝังอยู่ในหน้าเว็บไปทำการดึงข้อมูล XML จากฝั่งเซิร์ฟเวอร์ กระบวนการที่เกิดขึ้นจะเป็นแบบ Asynchronous จากนั้นจะใช้ข้อมูลที่ได้รับมาทำการแก้ไข Document Object Model (DOM) ของหน้าเว็บนั้นโดยใช้เทคนิคของจาวาสคริปต์หัวใจของ Ajax อยู่ที่ตัวออปเจ็ค XMLHttpRequest ที่มีอยู่ในบราวเซอร์รุ่นใหม่ๆ ทุกตัว สามารถจัดการได้โดยการออกแบบหน้าเว็บให้เหมาะสมและยังต้องรองรับในกรณีที่บราวเซอร์ถูกปิดการทำงานของ JavaScript ด้วย

Ajax จะช่วยลดการติดต่อระหว่างไคลเอ็นท์กับเซิร์ฟเวอร์ โดยในการโหลดหน้าเว็บนั่น บราวเซอร์จะโหลดข้อมูลจาก Ajax engine แทนการร้องขอข้อมูลจากเซิร์ฟเวอร์โดยตรง ดังนั้น Ajax จะทำหน้าที่ทั้งการแสดงข้อมูล ส่วนติดต่อกับผู้ใช้และติดต่อไปยังเซิร์ฟเวอร์แล้ว Ajax engine อนุญาตให้การกระทำต่างๆ ในเว็บแอพพลิเคชันเป็นแบบ Asynchronous คือความเป็นอิสระในการติดต่อไปยังเซิร์ฟเวอร์นั่นเอง ดังนั้นผู้ใช้จะไม่พบกับบราวเซอร์หน้าขาวอีกต่อไปและไม่ต้องรอการโหลดข้อมูลต่างๆ จากเซิร์ฟเวอร์

การตอบโต้กับผู้ใช้งานส่วนใหญ่ของเว็บแอพพลิเคชันที่เขียนด้วยเทคนิค Ajax จะทำงานในเครื่องลูกข่ายการตีความในแต่ละหน้านั้น จะทำโดยใช้ DOM (document object model) ของบราวเซอร์ Ajax นั้นยังสามารถทำงานหลายๆ งานได้ อย่างเช่น ปรับปรุงหรือลบแถวในฐานข้อมูล หด หรือขยายหน้าฟอร์ม แสดงผลการค้นหาอย่างง่ายหรือแม้กระทั่งแก้ไขระบบหมวดหมู่ โดยไม่ต้องโหลดหน้าทั้งหน้าทุกครั้ง เพียงการส่งคำร้อง (Request) เล็กๆไปที่แม่ข่าย (เซิร์ฟเวอร์) แล้วผลลัพธ์ก็จะได้กลับมาอย่างรวดเร็ว ด้วยเทคนิคจะช่วยให้การใช้ DHTML ได้ผสมผสานไปในการ

พัฒนาส่วนที่ตอบโต้กับผู้ใช้ และการพัฒนาหน้าเว็บให้มีลูกเล่นมากขึ้นด้วย Ajax เป็นรูปแบบเอกสารที่ดีและรับรองการแสดงผลในบราวเซอร์ทุกชนิด และในหลาย OS ด้วย อธิบายดังรูปที่ 2.6



รูปที่ 2. 6 การทำงานของ Ajax เว็บแอพพลิเคชันเทียบกับการทำงานของเว็บแอพพลิเคชันแบบเดิม

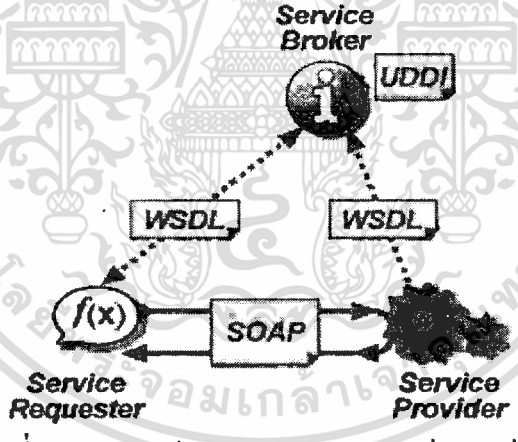
ข้อดีของ Ajax

- ตอบสนองต่อผู้ใช้ได้อย่างรวดเร็วเนื่องจากการปรับปรุงแบบบางส่วน
- ผู้ใช้ไม่ต้องหยุดรอคอยการประมวลของเซิร์ฟเวอร์ เนื่องจากการติดต่อแบบ Asynchronous
- รองรับกับบราวเซอร์ที่สามารถใช้ JavaScript ได้
- ทำให้การประมวลผลที่เซิร์ฟเวอร์มีความรวดเร็วขึ้นเนื่องจากการประมวลผลที่เซิร์ฟเวอร์ลดลง
- ไม่ต้องทำการติดตั้ง หรือใช้ Plugs-in
- ไม่ยึดติดกับแพลตฟอร์มหรือภาษาที่ใช้ในการเขียนโปรแกรม
- เป็นเทคโนโลยีใหม่ที่ไม่ได้เป็นของนักพัฒนาเว็บแอพลิเคชันคนใด นั่นคือทุกคนมีสิทธิ์เข้ามาพัฒนาแอพลิเคชันตัวนี้

2.4 เว็บเซอร์วิส (Web Service)

เทคโนโลยีที่ออกแบบมา เพื่อสนับสนุนการแลกเปลี่ยนข้อมูลกันระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย โดยมีภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ คือ XML เว็บเซอร์วิสมีอินเทอร์เฟสที่ใช้อธิบายรูปแบบข้อมูล การเรียกใช้งาน คือ WSDL ระบบคอมพิวเตอร์ใช้งานสื่อสารโต้ตอบกับเว็บเซอร์วิสตามรูปแบบที่ได้กำหนดไว้แล้ว โดยการส่งข้อมูลตามอินเตอร์เฟสของเว็บเซอร์วิสนั้น โดยที่ข้อมูลดังกล่าวอาจแนบไว้ในซอง SOAP ข้อมูลเหล่านี้ปกติแล้วถูกส่งโดยอาศัย HTTP และใช้ XML ร่วมกับมาตรฐานเกี่ยวกับเว็บอื่นๆ ทำให้โปรแกรมประยุกต์ที่เขียนโดยภาษาต่างๆ และทำงานบนแพลตฟอร์มต่างๆกัน สามารถใช้เว็บเซอร์วิสเพื่อแลกเปลี่ยนข้อมูลผ่านทางเครือข่ายคอมพิวเตอร์ เช่น อินเทอร์เน็ต ในลักษณะเดียวกับการสื่อสารระหว่างโปรเซส (Inter-process communication) บนเครื่องเดียวกัน ความสามารถในการแลกเปลี่ยนข้อมูลระหว่างระบบที่ต่างกันนี้ เกิดขึ้นได้เนื่องจากการใช้มาตรฐานเปิด โดย OASIS และ W3C เป็นคณะกรรมการหลักในการรับผิดชอบมาตรฐานและสถาปัตยกรรมของเว็บเซอร์วิส

สถาปัตยกรรมของเว็บเซอร์วิส (Web Service Architecture)



รูปที่ 2. 7 โครงสร้างสถาปัตยกรรมของเว็บเซอร์วิส

โครงสร้างสถาปัตยกรรมระบบที่เน้นการให้บริการเป็นหลัก เรียกว่า Service-Oriented Architecture: SOA ประกอบด้วย 3 ส่วนหลัก แสดงดังรูปที่ 2.7

1. ผู้ให้บริการ (Service Provider) ที่ทำการประกาศ (Publish) บริการขององค์กรไปยัง ไดเรคทรอรี่ที่เก็บทะเบียนของการบริการ
2. ตัวแทนผู้ให้บริการ (Service Broker/Repository) โดยผู้ให้บริการต้องลงทะเบียน เพื่อระบุชื่อบริการและพารามิเตอร์ หรือเงื่อนไขที่จะสามารถเรียกใช้บริการนั้นได้ ด้วยการใช้มาตรฐาน WSDL:Web Services Description Language ในการอธิบายว่าโปรแกรมนั้น จะถูกเรียกใช้งานได้อย่างไร ทั้งนี้ตัวแทนผู้ให้บริการเองจะบันทึกและจัดเก็บบริการต่างนั้น โดย

ใช้มาตรฐานของระบบไดเรคทรอรี่ ที่เรียกว่า UDDI :Universal Description, Discovery and Integration ซึ่งจะช่วยกำหนดการลงทะเบียนการค้นหา และเข้าถึงบริการที่อยู่ในไดเรคทรอรี่

3. ผู้ขอใช้บริการ (Service Requester) เป็นเครื่องคอมพิวเตอร์ปลายทาง ที่ต้องการเข้าไปสืบค้น หารายการ บริการ จากตัวแทนผู้ให้บริการ เมื่อพบว่าบริการนั้น อยู่ที่ผู้ให้บริการใด ก็จะเรียกใช้ (bind) ไปยังผู้ให้บริการนั้นๆ ได้โดยตรงต่อไป

XML (The Extensible Markup Language)

XML ย่อมาจากคำว่า e**X**tensible **M**arkup **L**anguage เป็นภาษาที่ใช้กำหนดรูปแบบของคำสั่งภาษา HTML หรือที่เรียกว่า Meta Data ซึ่งจะใช้สำหรับกำหนดรูปแบบของคำสั่ง Markup ต่าง ๆ แต่มีข้อแตกต่างกับ HTML ที่เป็น Markup Language ซึ่ง XML ได้รับการพัฒนามาจาก SGML (Standard Generalized Markup Language) ที่เป็นข้อกำหนดในการสร้างหรือจัดทำเอกสารในรูปแบบอิเล็กทรอนิกส์ที่กำหนดโดย W3C หรือ World Wide Web Consortium ซึ่งเป็นภาษาที่นิยมใช้และได้รับการพัฒนาให้มีประสิทธิภาพสูงสุดในการทำงานบนเว็บ โดย XML จะประกอบด้วย 3 ส่วนพื้นฐานด้วยกัน คือ เอกสารข้อมูล (Data document) เอกสารนิยามความหมาย (definition document) และ นิยามภาษา (definition language)

การใช้งาน XML จำเป็นต้องใช้ร่วมกับ Style Sheet หรือมาตรฐานอื่นๆ เพราะ XML เพียงแต่กำหนดรูปแบบของ Tag เท่านั้น ไม่ได้กำหนดว่า Tag จะแสดงผลแบบใด ดังนั้นหากเอาข้อมูลในรูปแบบ XML ไปแสดงผลในอุปกรณ์ชนิดใดก็ตามจะต้องกำหนดวิธีแสดงผลของอุปกรณ์นั้นด้วย นอกจากนี้ XML ยังสนับสนุนตัวอักษรภาษานานาชาติ โดยใช้มาตรฐาน ISO 10646 จุดมุ่งหมายของภาษา XML คือ ภาษาต้องเรียบง่าย มีคำสั่งน้อยที่สุด สามารถเขียนด้วยโปรแกรมแก้ไขข้อความ (Text Editor) และสนับสนุนการทำงานร่วมกับแอพพลิเคชันได้หลายชนิด ซึ่งในปัจจุบันนี้ได้มีการพัฒนาภาษา Markup ตามข้อกำหนดของ XML แล้ว เช่น SMIL สำหรับควบคุมข้อมูลมัลติมีเดีย

XML เป็นส่วนหนึ่งของ HTML แต่ XML จะให้รายละเอียดเกี่ยวกับข้อมูลต่างๆ เช่น ชื่อเมือง อุณหภูมิ ความกดอากาศ เป็นต้น ส่วน HTML เป็นการกำหนด tag ต่างๆ ว่าจะให้ข้อมูลแสดงผลในรูปแบบใด ซึ่งข้อมูลสามารถแสดงผลได้หลายรูปแบบไม่ว่าจะเป็นตารางหรือ text ธรรมดา ขึ้นอยู่กับการกำหนดของ HTML และ XML ยังสามารถให้รายละเอียดของเนื้อหาเอกสาร เรียกว่า Document Type Definition (DTD) ว่าจะแสดงหรือซ่อนส่วนใดของเอกสาร

XML เป็นภาษาที่ใช้เน้นส่วนที่เป็นข้อมูล โดยสามารถกำหนดชื่อแท็ก (Element) และชื่อแอตทิบิวต์ ได้ตามความต้องการของผู้สร้างเอกสาร XML โดยเอกสารนั้นจะต้องมีความเป็น Well-formed ส่วน DTD และ Schema จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับว่ามีผู้ใช้เอกสารนั้นมากน้อยแค่ไหน

เอกสาร XML จึงเป็นแค่เท็กซ์ไฟล์ชนิดหนึ่งที่มีแท็กเปิดและเท็กปิดครอบข้อมูลไว้ตรงกลางเท่านั้น ทำให้เอกสาร XML ถูกใช้ในการติดต่อกับระบบที่ต่างกัน เนื่องจากความง่ายในการสร้างเอกสาร การนำเอกสาร XML ไปใช้งาน จะสนใจแต่ข้อมูลที่ถูกเน้นด้วยแท็กมากกว่า

Well-formed เป็นไวยากรณ์พื้นฐานของเอกสาร XML อย่างเช่น เอกสาร XML ต้องเริ่มต้นด้วย <?xml version="1.0" ?> เอกสาร XML 1 เอกสารจะต้องมีแท็กรูทเพียงแท็กเดียว หมายความว่าแท็กและข้อมูลต่างๆ จะต้องอยู่ภายในแท็กแรกสุดเพียงแท็กเดียว การเปิดและปิดแท็กจะต้องไม่มีการครอมกัน เช่น ตัวหนาและเอียง จะไม่ Well-formed

เนื่องจากเอกสาร XML สามารถกำหนดชื่อแท็ก และชื่อแอตทิบิวต์ได้ตามความต้องการของผู้สร้างเอกสาร ทำให้ในการเน้นข้อมูลใดข้อมูลหนึ่งสามารถมีเอกสาร XML หลายรูปแบบ (ผู้เขียนอาจใช้ชื่อแท็กต่างกัน ทั้งที่สื่อความหมายไปที่สิ่งเดียวกัน) หากว่าเอกสาร XML นั้น ถูกนำไปใช้ติดต่อกับระบบอื่นๆ อาจทำให้สื่อความหมายไม่ตรงกัน ดังนั้นจึงต้องมีการกำหนดรูปแบบที่เป็นมาตรฐานขึ้น โดย DTD และ Schema จะเป็นตัวกำหนดว่าเอกสาร XML นั้น จะต้องมีแท็กอะไรบ้าง ภายในแท็กนั้นจะมีแท็กแอตทิบิวต์ หรือข้อมูลอะไรได้บ้าง โดย DTD จะต่างกับ Schema ตรงที่ Schema เป็นเอกสาร XML ด้วย ตัวอย่างการเขียน XML แสดงตามรูปที่ 2.8

ตัวอย่าง XML

รูปที่ 2. 8 ตัวอย่างการเขียน XML และ โครงสร้างแบบ Tree ของ XML

ความถูกต้องของ XML แบ่งเป็น 2 ระดับ

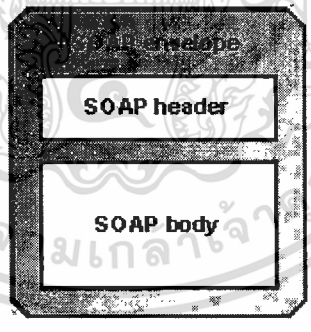
- **Well-formed** เอกสารที่ well-formed คือใช้ syntax ของ XML ถูกต้องตามมาตรฐานทุกอย่าง เอกสารที่ไม่ well-formed ถือว่าไม่เป็น XML
- **Valid** นอกจาก well-formed แล้ว เอกสารที่ valid ยังต้องใช้แท็ก XML ที่กำหนดเฉพาะใน schema ที่ตกลงกันไว้เท่านั้น ปัจจุบันมี schema ที่นิยม 3 ตัว คือ Document Type Definition (DTD), XML Schema (WXS) และ RELAX NG

SOAP (Simple Object Access Protocal)

เป็นมาตรฐานที่เขียนขึ้นในรูป XML (lightweight protocol) สำหรับ การแลกเปลี่ยนข้อมูลในสภาพแวดล้อมแบบกระจายศูนย์ (decentralized, distributed environment) SOAP ได้ กำหนดเมเสจจิ้งโปรโตคอล (Messaging Protocol) ระหว่างผู้ขอบริการ (requestor) กับผู้ให้บริการ (provider) การพัฒนา SOAแม้ว่า SOA จะไม่ได้กำหนดเมจเสจจิ้งโปรโตคอล (Messaging Protocol) ไว้ แต่ SOAP ได้ถูกกำหนด ให้เป็น Services-Oriented Architecture Protocol เรียบร้อยแล้ว เนื่องจากมันได้ถูกใช้ในการพัฒนา SOA อย่างแพร่หลายแล้วนั่นเอง จุดเด่นของ SOAP ก็คือเป็นโปรโตคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของและเป็นโปรโตคอลที่ทำงานกับโปรโตคอลอื่นหลายชนิด การพัฒนาก็อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการ แบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา

SOAP ประกอบด้วย SOAP envelope แบ่งข้อมูลเป็น 2 ส่วน แสดงดังรูปที่ 2.9 ได้แก่

- The SOAP header and the SOAP body
- ข้อมูลเกี่ยวกับชื่อในการระบุตัวผู้รับส่วนหัว (header) หากมีการแสดงอยู่จะส่งข้อมูลข่าวสารที่บรรจุอยู่ภายใน ตัวอย่างเช่น รายละเอียดการดำเนินการ วิธีการรักษาความปลอดภัยของข้อมูลคำอธิบายหรือข้อมูลรายละเอียดบุคคล ส่วน body ประกอบด้วยการร้องขอหรือการตอบกลับของเว็บเซอร์วิส ซึ่งจะอยู่ในรูปแบบของภาษา XML โครงสร้างระดับสูงของข้อความ SOAP

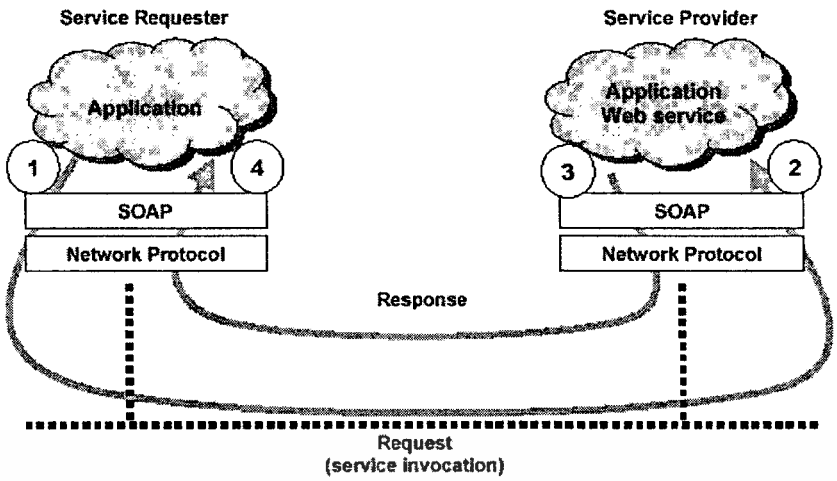


รูปที่ 2. 9 โครงสร้างส่วนประกอบของ SOAP

SOAP มีลักษณะเฉพาะดังต่อไปนี้

- ไม่ขึ้นกับโปรโตคอล (Protocol independence)
- ไม่ขึ้นกับภาษาที่เขียน (Language independence)
- ไม่ขึ้นกับรูปแบบและระบบปฏิบัติการ (Platform and operating system independence)

สนับสนุนข้อความ SOAP ที่มีโครงสร้างในรูปแบบ XML (ใช้โครงสร้าง MIME หลายส่วน)



รูปที่ 2. 10 การเรียกใช้งานเว็บเซอร์วิส

แอพพลิเคชันของผู้ร้องขอบริการสร้าง SOAP message เพื่อเรียกใช้บริการจากเว็บเซอร์วิส เว็บเซอร์วิสของผู้ให้บริการได้รับ SOAP message จากผู้ร้องขอ ซึ่งอยู่ในรูปแบบ XML เว็บเซอร์วิสประมวลผลตาม Component ที่ให้บริการ เว็บเซอร์วิสส่งผลลัพธ์มา แล้วผู้ให้บริการก็จะสร้าง SOAP Message ที่มีผลลัพธ์นั้นส่งกลับมายังผู้ร้องขอบริการ

แอพพลิเคชันของผู้ร้องขอบริการได้ผลลัพธ์ที่เป็น SOAP Message แล้วทำการแปลงให้อยู่ในรูปแบบที่ต้องการเพื่อนำไปประมวลผลต่อ

WSDL (Web Service Description Language)

WSDL นั้นเป็น XML-base language ซึ่งใช้ในการบรรยาย เว็บเซอร์วิสหรือ network endpoint เว็บเซอร์วิสดีแอล (WEB SERVICEDL) นั้นยังสามารถที่จะบรรยายการส่ง messaging ระหว่างเว็บเซอร์วิส ระบุตำแหน่งที่อยู่ของเว็บเซอร์วิสและร่วมทั้งโพรโทคอลที่ใช้ในการติดต่อสื่อสารกันของเว็บเซอร์วิส WSDL นั้นจะทำงานรวมกันกับ SOAP และ UDDI เพื่อที่จะทำให้เว็บเซอร์วิสติดต่อกับเว็บเซอร์วิสอื่นได้บนระบบอินเทอร์เน็ต UDDI นั้นเป็นตัวระบุตำแหน่งของเว็บเซอร์วิส เว็บเซอร์วิสดีแอล นั้นเป็นตัวที่บรรยายเว็บเซอร์วิส ส่วน SOAP เป็นตัวที่ทำหน้าที่เกี่ยวกับการส่งข้อมูลบนระบบเว็บเซอร์วิส ซึ่ง WSDL นั้นสามารถที่จะบรรยายเว็บเซอร์วิสหรือ network endpoint เพื่อเป็นการเผยแพร่ข้อมูลการให้บริการ (Services) แก่ระบบภายนอกโดยที่ทำงานผ่านระบบ network

ประโยชน์ของ WSDL นอกเหนือไปจากการเป็นตัวนิยามรูปแบบรายละเอียดเว็บเซอร์วิส ไฟล์ WSDL ซึ่งนำเสนอผ่านรูปแบบเอกสาร XML ยังสามารถใช้เป็นเอกสารอ้างอิงสำหรับผู้พัฒนาเว็ปแอปพลิเคชันทั้งผู้สร้างเว็ปเซอร์วิสและผู้ใช้งานเว็ปเซอร์วิส เพราะคำนิยามต่างๆภายในสามารถเข้าใจได้ง่าย

ในมุมมองของโครงสร้าง WSDL ซึ่งนิยามโครงสร้างการเข้าถึงใช้งานเว็ปเซอร์วิส นิยามเซอร์วิส (services) เป็นกลุ่มของจุดเชื่อมต่อปลายทางหรือ URL ปลายทางตัวหลักที่จะเรียกใช้ โดยจุดเชื่อมต่อแต่ละตัวถูกเรียกว่า พอร์ต (port) ส่วนฟังก์ชันที่ใช้ในการทำงานซึ่งอาจมีได้หลายตัวนั้น ถูกเรียกว่าโอเปอเรชัน (Operations) แต่ละโอเปอเรชันจะมีพารามิเตอร์และค่าส่งกลับซึ่งพารามิเตอร์หรือค่าส่งกลับจากฟังก์ชันแต่ละตัวนั้น จะถูกนำมานิยามแยกต่างหากเรียกว่า แมสเสจ (messages) เหตุผลที่มีการแยกส่วนนิยามพารามิเตอร์และค่าส่งกลับออกมาก็เพื่อหากผู้นิยาม WSDL มีการนิยามโอเปอเรชันหลายตัว ซึ่งพารามิเตอร์มีจุดมุ่งหมายเหมือนกัน ก็จะสามารถใช้แมสเสจร่วมกันได้โดยสะดวกนั่นเอง

โอเปอเรชันจะถูกนิยามรวมกันภายใต้พอร์ตไทป์ (port types) และกลไกการเข้ารหัสการส่ง การจัดการกระบวนการเชื่อมต่อของเว็บเซอร์วิส ซึ่งทำหน้าที่เชื่อมโอเปอเรชันที่นิยามในพอร์ตไทป์ เข้ากับจุดเชื่อมต่อที่นิยามในเซอร์วิส ถูกกระทำโดยการไบน์ดิง (bindings)

กล่าวโดยสรุป โครงสร้างเอกสารอธิบายเว็บเซอร์วิสตามมาตรฐาน WSDL จึงประกอบไปด้วยอิลิเมนต์หลักๆ ดังนี้

Element	Defines
<portType>	รวมโอเปอเรชันที่ใช้ในเว็บเซอร์วิส
<message>	การนิยามพารามิเตอร์ และค่าส่งกลับของโอเปอเรชัน (ชนิดของข้อมูลต่างๆ ที่จะส่งไปและกลับระหว่างเว็บเซอร์วิสกับโปรแกรมที่เรียกใช้งานเว็บเซอร์วิส)
<types>	การนิยามชนิดข้อมูลเป็นการเฉพาะเพื่อใช้ในบริการนี้เท่านั้น (สำหรับกรณีแบบข้อมูลที่นอกเหนือจากมาตรฐานที่มีการกำหนดโดยปกติ)
<operation>	การนิยามโอเปอเรชัน หรืออาจกล่าวเทียบเคียงว่าเป็นโปรโตไทป์ของฟังก์ชันแต่ละตัวของเว็บเซอร์วิส
<port>	จุดหมายปลายทางหรือที่ตั้งของเว็บเซอร์วิสที่จะเรียกใช้งาน
<binding>	การนิยามโพรโทคอล และการเข้ารหัสข้อมูลเพื่อใช้ในการเรียกเว็บเซอร์วิส
<service>	รวมการนิยามพอร์ตที่ใช้ในเว็บเซอร์วิสนี้

เพื่อความสะดวกในการนิยามเอกสารแสดงรายละเอียดเว็บเซอร์วิสด้วย WSDL จึงได้มีการนิยามเนมสเปซ URI ที่เป็นมาตรฐานไว้ให้ใช้งาน ซึ่งหากเราพบเนมสเปซที่มีชื่อดังต่อไปนี้ ผู้พัฒนาก็จะได้เข้าใจได้ว่ากำลังใช้มาตรฐานกลางอยู่

prefix	namespace URI	definition
wsdl	http://schemas.xmlsoap.org/wsdl/	WSDL namespace for WSDL framework.
soap	http://schemas.xmlsoap.org/wsdl/soap/	WSDL namespace for WSDL SOAP binding.
http	http://schemas.xmlsoap.org/wsdl/http/	WSDL namespace for WSDL HTTP GET & POST binding.
mime	http://schemas.xmlsoap.org/wsdl/mime/	WSDL namespace for WSDL MIME binding.
soapenc	http://schemas.xmlsoap.org/soap/encoding/	Encoding namespace as defined by SOAP 1.1
soapenv	http://schemas.xmlsoap.org/soap/envelope/	Envelope namespace as defined by SOAP 1.1
xsi	http://www.w3.org/2000/10/XMLSchema-instance	Instance namespace as defined by XSD.
xsd	http://www.w3.org/2000/10/XMLSchema	Schema namespace as defined by XSD.
tns	แล้วแต่ผู้ใช้จะกำหนด	tns ย่อมาจาก this namespage ซึ่งก็คือเนมสเปซที่อ้างถึงเอกสาร WSDL ปัจจุบัน
อื่นๆ	แล้วแต่ผู้ใช้จะกำหนด	เป็น URI เฉพาะที่ผู้ใช้กำหนดเอง

UDDI (Universal Description, Discovery and Integration)

ไดเรกทอรี่ที่เก็บหรือลงทะเบียนเว็บเซอร์วิส UDDI เป็นข้อกำหนดอันเกี่ยวกับระบบบริการลงทะเบียน (registry service) สำหรับเว็บเซอร์วิสและสำหรับบริการอื่นๆทั้งที่ไม่ใช่แบบอิเล็กทรอนิกส์ และแบบอิเล็กทรอนิกส์ ตัวบริการลงทะเบียน UDDI คือ เว็บเซอร์วิสซึ่งจัดการข้อมูลเกี่ยวกับผู้ให้ บริการด้านต่างๆ หรือแม้แต่กระทั่งให้บริการข้อมูล บรรดาผู้ให้บริการสามารถใช้ UDDI ในการประกาศว่า บริการใดบ้างที่ให้บริการ และลูกค้าสามารถใช้บริการของ UDDI ในการ

ค้นหาบริการที่ตนต้องการได้ตรงตามความต้องการของตนได้โดยการลงทะเบียนของ UDDI สามารถแบ่งได้เป็น 2 แบบ

1. การลงทะเบียนแบบ public UDDI ที่ให้บริการแก่ผู้ค้นหาโดยทั่วไป
2. การลงทะเบียนแบบ private UDDI ที่บริการภายในองค์กร

ข้อกำหนด UDDI ได้มีการนิยามตามนี้

- SOAP API (Simple Object Application Programming Application Programming Interface) ซึ่งตัวแอพพลิเคชันจะใช้ในการสอบถามและประกาศข้อมูลไปยังระบบลงทะเบียน UDDI
- XML Schema คือโครงสร้างรูปแบบของระบบลงทะเบียนและรูปแบบของข้อความ SOAP (SOAP Message format)
- WSDL คือข้อกำหนดของ SOAP APIs
- ข้อกำหนด UDDI รีจิสทรี(รูปแบบทางเทคนิค t-models) ของข้อกำหนดหลายๆ อย่างและหมวดหมู่ของระบบซึ่งอาจใช้ในการกำหนดและจัดแบ่งหมวดหมู่การลงทะเบียน UDDI

บทที่ 3
การวิเคราะห์และออกแบบ

บทนี้การศึกษาความต้องการของระบบการวิเคราะห์ระบบ โครงสร้างการทำงานโดยรวม รวม ถึงการออกแบบการเก็บข้อมูลทั้งในรีเลชันนอลดาต้าเบสและไดเร็คทอรี่ การออกแบบส่วนติดต่อกับผู้ใช้งาน

3.1 ปัญหาของระบบการพิสูจน์ตัวตน

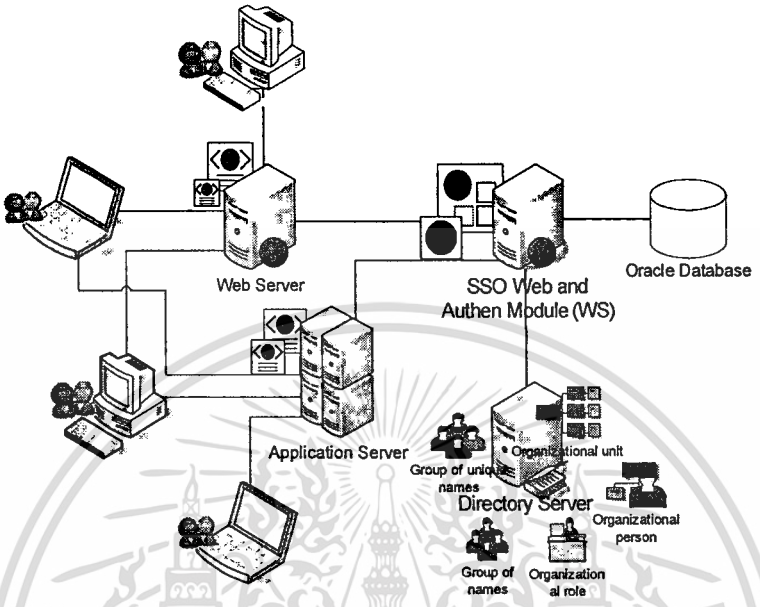
เมื่อระบบสารสนเทศเข้ามามีบทบาทในระบบธุรกิจ องค์กรทั้งขนาดเล็กไปจนถึงองค์กรขนาดใหญ่ นำระบบสารสนเทศเข้ามา เพื่ออำนวยความสะดวกให้แก่ระบบงานในภายองค์กร ซึ่งจำนวน ขนาดของระบบสารสนเทศนั้นขึ้นอยู่กับความเหมาะสม แนวทางการดำเนินธุรกิจขององค์กรและการมีระบบสารสนเทศหลายระบบภายในองค์กรหนึ่งๆ นั้นทำให้ผู้ใช้งานภายในองค์กรจำเป็นต้อมีชุดรหัสเพื่อเข้าใช้งานระบบสารสนเทศนั้นๆ ตามจำนวนของระบบสารสนเทศ ซึ่งโดยทั่วไปเมื่อมีการพัฒนาระบบสารสนเทศกจะมีโมดูลในการพิสูจน์ตัวตนและการให้สิทธิการเข้าถึงข้อมูลไว้ในแต่ละระบบ ทำให้เก็บการจัดเก็บข้อมูลซ้ำซ้อนและด้วยความแตกต่างกันเทคโนโลยีในการพัฒนาระบบสารสนเทศ มีผล ให้การจัดการหรือควบคุมข้อมูลการเข้าถึงระบบสารสนเทศต่างๆ ทำได้ไม่สะดวกและเกิดความผิด พลาด ได้ง่าย

3.2 ความต้องการของระบบ

- จัดทำระบบการพิสูจน์ตัวตนและการกำหนด สิทธิการเข้าถึงทรัพยากรด้วยรหัสผ่านชุดเดียว
- เพิ่มความเป็นอิสระต่อเทคโนโลยี (loosely Coupled) ด้วยการพัฒนาในรูปแบบเว็บเซอร์วิส
- ลดปัญหาความซ้ำซ้อนในการพิสูจน์ตัวตน และการกำหนดสิทธิเข้าระบบ
- เพิ่มประสิทธิภาพในการรักษาความปลอดภัยของข้อมูลสำคัญด้วยฐานข้อมูลไดเรคทอรี่ (Directory)
- เน้นการกำหนดสิทธิการเข้าถึงแอพพลิเคชันในแพลตฟอร์มที่ต่างกันโดยการใช้เว็บเซอร์วิส

3.3 การวิเคราะห์ระบบและการออกแบบการทำงานของระบบ

การวิเคราะห์และออกแบบระบบการพิสูจน์ตัวตนครั้งเดียว จะแบ่งการพัฒนาออกเป็น 3 ส่วนหลักๆ แสดงตามรูปที่ 3.1



รูปที่ 3. 1 ภาพรวมการพัฒนาระบบ

ประกอบด้วย

- ไคลเอนท์ ผู้ใช้งานระบบจะส่ง request ผ่านแอพพลิเคชันเซิร์ฟเวอร์ไปเรียกใช้งานเซอร์วิสของเว็บเซอร์วิสเซิร์ฟเวอร์หรือผู้ใช้งานที่ส่ง request มาใช้งานเซอร์วิสของเว็บเซอร์วิสโดยตรง
- เว็บเซอร์วิสจะประกอบไปด้วยเซอร์วิสในการพิสูจน์ตัวตน โดยในการค้นหาข้อมูลในไดเรคทอรี่เซิร์ฟเวอร์จะมีการติดต่อกันผ่าน LDAP Access Protocol เพื่อให้เกิดความปลอดภัยในการสื่อสารไดเรคทอรี่เซิร์ฟเวอร์ ใช้เก็บข้อมูลตัวตนของผู้ใช้งานส่วนข้อมูลรายละเอียดอื่นจะเก็บอยู่ในฐานข้อมูลแบบรีเลชันนอล เช่น ข้อมูลการกำหนดสิทธิอำนาจการเข้าถึง รูปแบบการเข้าถึง ข้อมูลอายุการใช้งานรหัสผู้ใช้ เพื่อใช้กำหนดสิทธิของผู้ใช้งานระบบคนนั้นๆ ระบบในส่วนของเว็บเซอร์วิสจะทำหน้าที่หลักในการตรวจสอบเงื่อนไขที่กำหนดไว้
- เว็บแอพพลิเคชันเซิร์ฟเวอร์ เป็นส่วนเชื่อมประสานกันผู้ใช้งานเพื่อกำหนดสิทธิการเข้าถึงให้แต่ละบัญชีรายชื่อ โดยทำงานต่างๆจะเป็นลักษณะของการเรียกใช้เซอร์วิสในส่วนของ เว็บเซอร์วิส เพื่อจัดการและจัดเก็บข้อมูลระบบของผู้ใช้งานระบบทั้งหมดมีการเชื่อมต่อกันไดเรคทอรี่และฐานข้อมูลแบบรีเลชันนอลผ่านเซอร์วิสของเว็บเซอร์วิส

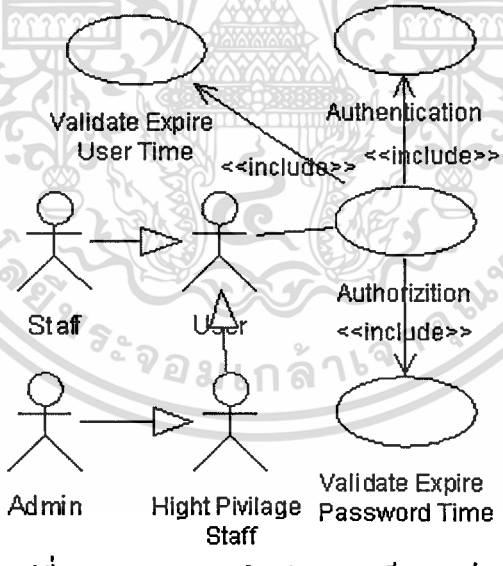
เท่านั้น เพื่อความปลอดภัยในการจัดเก็บข้อมูลในระบบส่วนนี้ เป็นการเพิ่มข้อมูลและแก้ไขจัดการข้อมูลผู้ใช้งาน การตั้งค่าระบบต่างๆ ซึ่งในการพัฒนา

3.4 ยูสเคสไดอะแกรม

ยูสเคสไดอะแกรมใช้แสดงว่าระบบประกอบด้วยฟังก์ชันงานหลัก และมีความสัมพันธ์กันอย่าง ไร ซึ่งสำหรับระบบการพิสูจน์ตัวตนเพียงครั้งเดียว เป็นการออกแบบเป็น 2 ส่วน คือ การทำงานของเว็บเซอร์วิส และการทำงานของเว็บแอพพลิเคชันในการจัดการข้อมูล

3.4.1 ยูสเคสการทำงานของเว็บเซอร์วิส

การทำงานของเซอร์วิสของเว็บเซอร์วิสจะเป็นในส่วนของการ Authorization โดยเมื่อผู้ใช้งานต้องการที่จะล็อกอินเข้าระบบสารสนเทศที่ต้องการใช้งาน ระบบสารสนเทศนั้นๆ จะทำตัวเป็นเครื่องไคลเอนท์ ของระบบพิสูจน์ตัวตนเพียงครั้งเดียว (Single Sign-On) ส่ง request ที่ประกอบไปด้วยชื่อระบบ รหัสผู้ใช้งาน และรหัสผ่าน มาที่เซอร์วิส Authorization เพื่อทำการตรวจสอบว่ารหัสผู้ใช้งานนี้ มีสิทธิการเข้าใช้งานระบบนั้นๆ หรือไม่ แล้วส่ง response กลับไปยังระบบสารสนเทศที่ request เข้ามาแสดงตามยูสเคสรูปที่ 3.2



รูปที่ 3. 2 ยูสเคสหลักในส่วนของเว็บเซอร์วิส

คำอธิบายยูสเคสไดอะแกรม

จากยูสเคสไดอะแกรมมีคำอธิบายยูสเคสอธิบายรายละเอียดของแต่ละยูสเคส ดังตารางที่ 3.1 ถึง 3.3 ตามลำดับ

ตารางที่ 3. 1 คำอธิบายยูสเคส Authentication

ยูสเคส	**Authentication**
วัตถุประสงค์	ใช้ในการตรวจสอบรหัสผู้ใช้งานและรหัสผ่าน
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	-
สิ่งที่กระตุ้นการทำงาน	เรียกใช้งานจากโมดูล authorization
อินพุต	รหัสผู้ใช้งานและรหัสผ่าน
เอาต์พุต	ข้อมูลบ่งบอกว่ารหัสผู้ใช้และรหัสผ่านถูกต้องหรือไม่
รายละเอียด	ทุกครั้งที่มี Request จากไคลเอนท์เพื่อทำการ Authorization โมดูลนั้นจะทำการเรียกใช้งาน authentication ก่อนเพื่อตรวจสอบว่ารหัสผู้ใช้งานนั้นมีในระบบและรหัสผ่านถูกต้องหรือไม่

ตารางที่ 3. 2 คำอธิบายยูสเคส Authorization

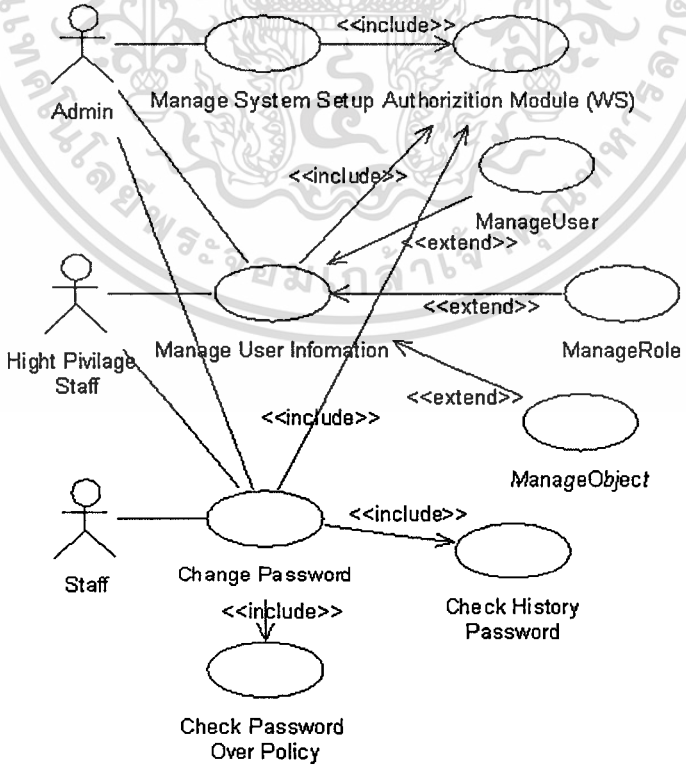
ยูสเคส	**Authorization**
วัตถุประสงค์	โมดูลหลักในการรับ Request และเรียกใช้โมดูลอื่น เพื่อการพิสูจน์ตัวตน
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	เครื่องไคลเอนท์(ระบบสารสนเทศอื่นๆ)
สิ่งที่กระตุ้นการทำงาน	Request จากไคลเอนท์เพื่อการพิสูจน์ตัวตนของรหัสผู้ใช้งานนั้นๆ
อินพุต	รหัสระบบสารสนเทศ รหัสผู้ใช้งานและรหัสผ่าน
เอาต์พุต	ข้อมูลของรหัสผู้ใช้งานนั้นๆ
รายละเอียด	1. ทุกครั้งที่มี Request จากไคลเอนท์เพื่อทำการ Authorization โมดูล จะทำการเรียกใช้งานAuthentication เพื่อตรวจ สอบว่ารหัสผู้ใช้งานนั้นมีในระบบและรหัสผ่านถูกต้องหรือไม่ 2. นำรหัสผู้ใช้งานเพื่อไปตรวจสอบว่า รหัสผู้ใช้งานนั้น ยังไม่หมดอายุ และรหัสผ่านสามารถใช้งานได้ 3. นำรหัสผู้ใช้งานและรหัสของระบบสารสนเทศนั้นๆ ไปดึงข้อมูลสิทธิในการเข้าระบบแล้วส่ง Response ให้ ไคลเอนท์

ตารางที่ 3. 3 คำอธิบายยูสเคส Validate Expire Password Time

ยูสเคส	Validate Expire Password Time
วัตถุประสงค์	ใช้ในการตรวจสอบข้อมูลการหมดอายุของรหัสผู้ใช้งานแต่รหัสผ่าน
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	-
สิ่งที่กระตุ้นการทำงาน	การเรียกใช้งานจากโมดูล Authorization
อินพุต	รหัสผู้ใช้งาน
เอาต์พุต	ข้อมูลของรหัสผู้ใช้งานนั้นๆ
รายละเอียด	ทำการตรวจสอบวันที่หมดอายุของรหัสผู้ใช้งานและรหัสผ่านว่ารหัสผ่านที่ส่งเข้ามานี้สามารถใช้ได้ ณ วันที่เรียกใช้

3.4.2 ยูสเคสการทำงานของเว็บแอพพลิเคชัน

การทำงานในส่วนของเว็บแอพพลิเคชันจะเป็นการจัดการกับข้อมูลต่างๆ ของผู้ใช้งานการเพิ่มข้อมูล แก้ไข ปรับปรุงข้อมูลต่างๆ รวมถึงการเปลี่ยนแปลงพารามิเตอร์ระบบ โดยแสดงยูสเคสหลักๆ ของเว็บแอพพลิเคชันดังรูปที่ 3.3



รูปที่ 3. 3 ยูสเคสหลักในส่วนของการจัดการข้อมูล (Web Application)

คำอธิบายยูสเคสไดอะแกรม

จากยูสเคสไดอะแกรมมีคำอธิบายยูสเคสอธิบายรายละเอียดของแต่ละยูสเคส ดังตารางที่ 3.4 ถึง 3.11 ตามลำดับ

ตารางที่ 3. 4 คำอธิบายยูสเคส Manage System Setup

ยูสเคส	**Manage System Setup**
วัตถุประสงค์	การจัดการเพิ่มและแก้ไขข้อมูลในส่วนของพารามิเตอร์ต่าง ๆที่ใช้ในระบบ เช่น ระยะเวลาในการหมดอายุของรหัสผ่าน เมื่อมีการสร้างผู้ใช้งานคนใหม่ การเพิ่มระบบงานใหม่ เป็นต้น
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	System Administrator
สิ่งที่กระตุ้นการทำงาน	เมื่อผู้ใช้คลิกที่ลิง Setup parameter
อินพุต	ค่าของพารามิเตอร์ที่ต้องการกำหนด ปรับแปลง แก้ไข
เอาต์พุต	ข้อมูลถูกจัดเก็บลงฐานข้อมูลเรียบร้อย
รายละเอียด	เมื่อผู้ใช้งานล็อกอินเข้ามาที่ระบบ ระบบทำการพิสูจน์ตัวตนของรหัสผู้ใช้งานมีสิทธิในการตั้งค่า แก้ไข เปลี่ยนแปลง
	ระบบได้ ผู้ใช้งานคลิกลิงค์เพื่อเข้าสู้หน้าจอ Manage system setup ทำการกำหนดค่าพารามิเตอร์แล้วกดปุ่ม save เพื่อบันทึกข้อมูลระบบทำการ confirm ว่าจัดเก็บข้อมูลเรียบร้อยแล้ว

หมายเหตุ เนื่องจากระบบจะทำการโหลดข้อมูลพารามิเตอร์เพียงครั้งแรกครั้งเดียวหลังจากที่ระบเริ่มต้นทำงานดังนั้นเมื่อแก้ไขพารามิเตอร์แล้วจำเป็นต้อง Restart ระบบ

ตารางที่ 3. 5 คำอธิบายยูสเคส Manage User Information

ยูสเคส	**Manage User Information**
วัตถุประสงค์	การเพิ่มและแก้ไขข้อมูลของผู้ใช้งานระบบ เช่น การเพิ่ม/แก้ไขหน้าที่การทำงาน (Role) การเพิ่มแก้ไขลักษณะการเข้าถึง (Access Type) เป็นต้น
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	High Privilege Staff
สิ่งที่กระตุ้นการทำงาน	เมื่อผู้ใช้เข้าสู่ระบบการพิสูจน์ตัวตนเพียงครั้งเดียว
อินพุต	-

ตารางที่ 3.5 คำอธิบายยูสเคส Manage User Information (ต่อ)

เอาต์พุต	-
รายละเอียด	เมื่อผู้ใช้งานล็อกอินเข้ามาที่ระบบ ระบบทำการพิสูจน์ตัวตนของรหัสผู้ใช้งานมีสิทธิในการจัดการข้อมูลผู้ใช้งาน ข้อมูลหน้าที่ (Role) ข้อมูลสิทธิ (Object) หน้าจอจะแสดงลิงค์เพื่อเข้าสู่การทำงานตามสิทธิที่ได้รับ เช่น การจัดการกับข้อมูลผู้ใช้งาน ค้นหาผู้ใช้งาน แก้ไขข้อมูลประวัติผู้ใช้งาน การจัดการกำหนดหน้าที่ให้กับผู้ใช้งาน การเพิ่ม หรือลบหน้าที่ของผู้ใช้งาน เป็นต้น

ตารางที่ 3. 6 คำอธิบายยูสเคส Change Password

ยูสเคส	**Change Password**
วัตถุประสงค์	การเปลี่ยนแปลงรหัสผ่านของผู้ใช้งานระบบ
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	Administrator, High Privilege Staff, Staff
สิ่งที่กระตุ้นการทำงาน	ผู้ใช้งานเลือกเมนู Change Password
อินพุต	รหัสผู้ใช้งาน รหัสผ่านชุดเก่า และ รหัสผ่านชุดใหม่
เอาต์พุต	ข้อมูลถูกจัดเก็บลงฐานข้อมูลเรียบร้อย
รายละเอียด	เมื่อผู้ใช้งานล็อกอินเข้ามาที่ระบบ ระบบทำการพิสูจน์ตัวตนของรหัสผู้ใช้งานมีสิทธิในการเปลี่ยนรหัสผ่าน เมื่อผู้ใช้ใส่รหัสผ่านชุดใหม่แล้วระบบจะทำการตรวจสอบรหัสผ่านว่าถูกต้องตามเงื่อนไขที่กำหนดไว้ เช่น รหัส ผ่านจะต้องมีความยาวน้อยที่สุด 8 ตัวอักษรและมากที่สุด 12 ตัวอักษรจะต้องประกอบด้วยอักขระพิเศษ 2 ตัว รหัสผ่านชุดใหม่จะต้องไม่ซ้ำกับรหัสผ่านชุดเก่าที่เคยเปลี่ยนมา 10 ครั้ง เป็นต้น เมื่อตรวจสอบแล้วจะทำการบันทึก รหัสผ่านชุดเก่าลงที่ฐานข้อมูลรีเลชันนอลและรหัสผ่านชุดใหม่จะถูกเก็บรักษาไว้ที่ไดเรคทอรี่เซอร์วิส

ตารางที่ 3. 7 คำอธิบายยูสเคส ManageUser

ยูสเคส	**Manage User Profile**
วัตถุประสงค์	ใช้ในการค้นหา แก้ไขข้อมูลรายละเอียดของผู้ใช้งาน
เงื่อนไขเมื่อเริ่มต้น	-

ตารางที่ 3.7 คำอธิบายยูสเคส ManageUser (ต่อ)

แอคเตอร์ที่เกี่ยวข้อง	Administrator, High Privilege Staff
สิ่งที่กระตุ้นการทำงาน	ผู้ใช้งานเลือกเมนู ในกลุ่ม Manage User Profile
อินพุต	รหัสผู้ใช้งาน ข้อมูลที่ต้องการแก้ไข
เอาต์พุต	ข้อมูลถูกจัดเก็บลงฐานข้อมูลเรียบร้อย
รายละเอียด	เมื่อผู้ใช้งานล็อกอินเข้ามาที่ระบบ ระบบทำการพิสูจน์ตัวตนของรหัสผู้ใช้งานมีสิทธิในการเปลี่ยนรหัสผ่านรหัสผ่านชุดเก่าจะถูกจัดเก็บไว้ที่ฐานข้อมูลรีเลชันนอล และรหัสผ่านชุดใหม่จะถูกเก็บรักษาไว้ที่ไดเรคทอรี่เซอร์วิส

ตารางที่ 3. 8 คำอธิบายยูสเคส Manage Role

ยูสเคส	**Manage Role Information**
วัตถุประสงค์	ใช้ในการค้นหา แก้ไขข้อมูลรายละเอียด-ของหน้าที่
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	Administrator, High Privilege Staff
สิ่งที่กระตุ้นการทำงาน	ผู้ใช้งานเลือกเมนู ในกลุ่ม Manage Role Information
อินพุต	รหัสหน้าที่ ข้อมูลที่ต้องการแก้ไข
เอาต์พุต	ข้อมูลถูกจัดเก็บลงฐานข้อมูลเรียบร้อย
รายละเอียด	เมื่อผู้ใช้งานล็อกอินเข้ามาที่ระบบ ระบบทำการพิสูจน์ตัวตนของรหัสผู้ใช้งานมีสิทธิในการจัดการกับหน้าที่ต่างๆ ในองค์กร โดยหน้าที่จะเป็นการกำหนดว่าผู้ใช้งานระบบมีหน้าที่ อะไรบ้างในระบบสารสนเทศนั้นๆ

ตารางที่ 3. 9 คำอธิบายยูสเคส Manage Object

ยูสเคส	**Manage Object Information**
วัตถุประสงค์	ใช้ในการค้นหา แก้ไขข้อมูลรายละเอียด-ของสิทธิ
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	Administrator, High Privilege Staff
สิ่งที่กระตุ้นการทำงาน	ผู้ใช้งานเลือกเมนู ในกลุ่ม Manage Object Information
อินพุต	รหัสสิทธิ ข้อมูลที่ต้องการแก้ไข
เอาต์พุต	ข้อมูลถูกจัดเก็บลงฐานข้อมูลเรียบร้อย

ตารางที่ 3.9 คำอธิบายยูสเคส Manage Object (ต่อ)

รายละเอียด	เมื่อผู้ใช้งานล็อกอินเข้ามาที่ระบบ ระบบทำการพิสูจน์ตัวตนของรหัสผู้ใช้งานมีสิทธิในการจัดการกับสิทธิการเข้าถึงทรัยพยากรหรือข้อมูลของระบบต่างๆ ในองค์กร โดยสิทธิจะเป็นการกำหนดว่าผู้ใช้งานระบบสามารถเข้าถึงส่วนต่างๆ ของแต่ละระบบได้หรือไม่ เช่น มีสิทธิในการดูข้อมูลลูกค้าในระบบ Customer Management ได้ ระบบ Customer Management ก็จะแสดงหน้าจอข้อมูลลูกค้าให้แก่รหัสผู้ใช้งานคนนั้นๆ

ตารางที่ 3. 10 คำอธิบายยูสเคส Check History Password

ยูสเคส	**Check History Password**
วัตถุประสงค์	การตรวจสอบว่ารหัสผ่านที่ใส่เข้ามาซ้ำกับรหัสผ่านในครั้งที่แล้วหรือไม่
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	Administrator, High Privilege Staff, Staff
สิ่งที่กระตุ้นการทำงาน	ผู้ใช้งานเลือกเมนู Check History Password
อินพุต	รหัสผู้ใช้ รหัสผู้ใช้งานชุดใหม่ จำนวนครั้งของ History password (ถูกกำหนดโดยผู้ดูแลระบบ)
เอาต์พุต	ข้อมูลบ่งบอกว่ารหัสผ่านชุดใหม่ซ้ำหรือไม่กับ History password ที่กำหนดไว้
รายละเอียด	ระบบจะทำการดึงข้อมูลจำนวนครั้งที่ระบบไม่ยอมให้ตั้งรหัสผ่านซ้ำจากพารามิเตอร์ทำการส่งผลลัพธ์ข้อมูลที่ตรวจสอบไปยังโมดูลที่เรียกใช้ต่อไป

ตารางที่ 3. 11 คำอธิบายยูสเคส Check Password Over Policy

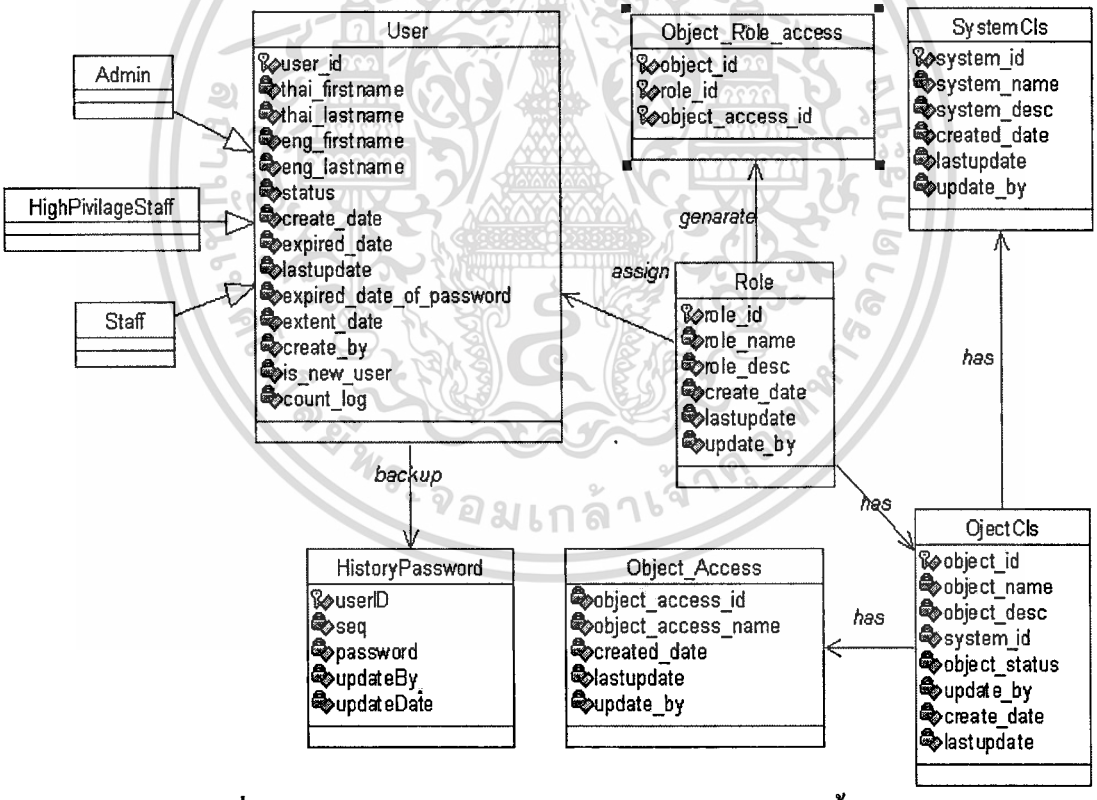
ยูสเคส	**Check Password Over Policy**
วัตถุประสงค์	การตรวจสอบว่า รหัสผ่านที่ผู้ใช้งานตั้งถูกต้องตาม กฎขององค์กรหรือไม่
เงื่อนไขเมื่อเริ่มต้น	-
แอคเตอร์ที่เกี่ยวข้อง	Administrator, High Privilege Staff, Staff
สิ่งที่กระตุ้นการทำงาน	ผู้ใช้งานเลือกเมนู Check History Password

ตารางที่ 3.11 คำอธิบายยูสเคส Check Password Over Policy (ต่อ)

อินพุต	รหัสผู้ใช้ รหัสผู้ใช้งานชุดใหม่ Policy
เอาต์พุต	ข้อมูลบ่งบอกว่ารหัสผ่านชุดใหม่ถูกต้องตาม Policy หรือไม่
รายละเอียด	ระบบจะทำการดึงข้อมูล Policy ที่ถูกกำหนดไว้ แล้วทำการตรวจสอบว่าข้อมูลรหัสผ่านชุดใหม่ที่ส่งเข้ามาถูกตาม Policy หรือไม่ แล้วส่งผลลัพธ์ข้อมูลที่ตรวจสอบไปยังโมดูลที่เรียกใช้ต่อไป

3.5 คลาสไดอะแกรม

คลาสไดอะแกรมใช้แสดงคลาสของระบบและความสัมพันธ์ระหว่างคลาส ซึ่งเป็นความสัมพันธ์เชิงสถิต หมายถึงความสัมพันธ์ที่มีอยู่แล้วเป็นปกติระหว่างคลาสต่างๆ คลาสไดอะแกรมของระบบการพิสูจน์ตัวตนเพียงครั้งเดียว ดังแสดงในรูปที่ 3.4



รูปที่ 3. 4 คลาสไดอะแกรมของระบบพิสูจน์ตัวตนเพียงครั้งเดียว

จากรูปที่ 3.4 มีคลาสต่างๆ ดังต่างไปนี้

- User หมายถึงคลาสของผู้ใช้ระบบหรือพนักงานในองค์กร
- Role หมายถึงคลาสของหน้าที่
- Object หมายถึงคลาสของสิทธิ

- System หมายถึงคลาสของระบบสารสนเทศที่ใช้การพิสูจน์ตัวตนของระบบพิสูจน์ตัวตน เพียงครั้งเดียว
- Object_Access หมายถึงคลาสของประเภทในการเข้าถึงตามสิทธิที่มี
- Object_Role_Access หมายถึงคลาสองค์ประกอบที่ประกอบด้วยสิทธิ หน้าที่ และประเภทการเข้าถึงทรัพยากร
- History_pasword หมายถึงคลาสที่เก็บข้อมูลรหัสผ่านของผู้ใช้งานแต่และคนย้อนหลัง

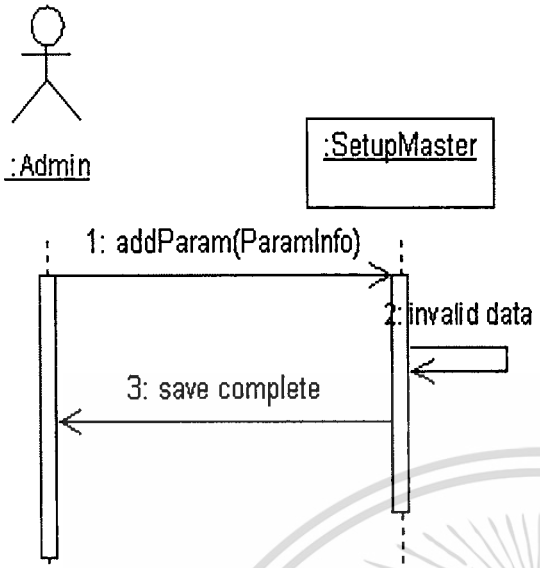
โดยแต่ละคลาสมีความสัมพันธ์กันดังนี้

- คลาส Admin, High Privilege Staff และ Staff เป็นคลาสลูกขงของคลาส User
- คลาส User จะมีความสัมพันธ์กับ History_Password แบบ 1 to many ผู้ใช้ระบบหนึ่งคนสามารถมี History_password ได้หลาย History_password
- คลาส Object มีความสัมพันธ์กับคลาส Object_Access แบบ many to many สิทธิหนึ่งสิทธิสามารถมีประเภทการเข้าถึงทรัพยากรได้มากกว่าหนึ่งประเภทและประเภทการเข้า ถึง ทรัพยากรหนึ่งๆสามารถกำหนดให้หลายๆ สิทธิก็ได้
- คลาส Role มีความสัมพันธ์กับคลาส Object แบบ many to many หน้าที่การทำงานหนึ่งๆ สามารถมีสิทธิในการเข้าถึงข้อมูลหรือทรัพยากรระบบได้มากกว่าหนึ่งสิทธิ และ สิทธิสามารถกำหนดให้สิทธิเดียวกันแก่หน้าที่ได้หลายหน้าที่
- ด้วยความสัมพันธ์ของคลาส Object กับคลาส Object_Access และ คลาส Role กับคลาส Object เป็นแบบ many to many และทั้งสามคลาสก็มีความสัมพันธ์กัน จึงมีคลาสองค์ประกอบเพื่อระบุ สิทธิ หน้าที่และประเภทการเข้าถึงทรัพยากร
- คลาส System มีความสัมพันธ์กับคลาส Object แบบ one to many โดยหนึ่งระบบสารสนเทศสามารถมีได้หลายๆ สิทธิในการเข้าใช้ทรัพยากร แต่หนึ่งสิทธิจะอยู่ในระบบหนึ่งระบบเท่านั้น

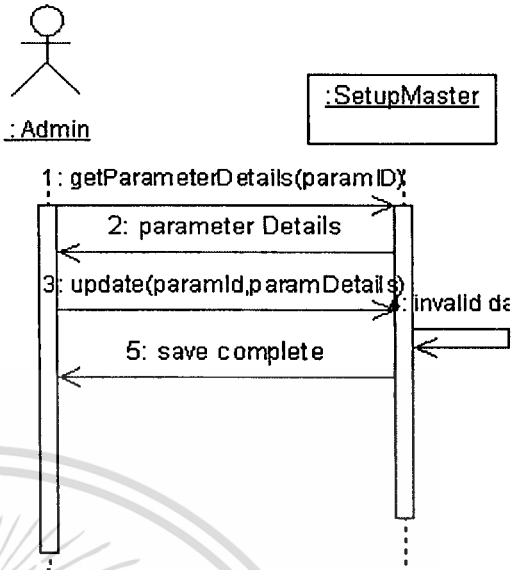
3.6 ซีเควนซ์ไดอะแกรม

แผนภาพซีเควนซ์ไดอะแกรมแสดงถึงการส่งผ่านหรือโต้ตอบข้อความ (Message) กันระหว่าง อ๊อบเจกต์ (Object) โดยในระบบการพิสูจน์ตัวตนครั้งเดียว

- ซีเควนซ์ไดอะแกรมการตั้งค่าพารามิเตอร์ต่าง ๆ ของระบบแสดงตามรูปที่ 3.5 และ 3.6

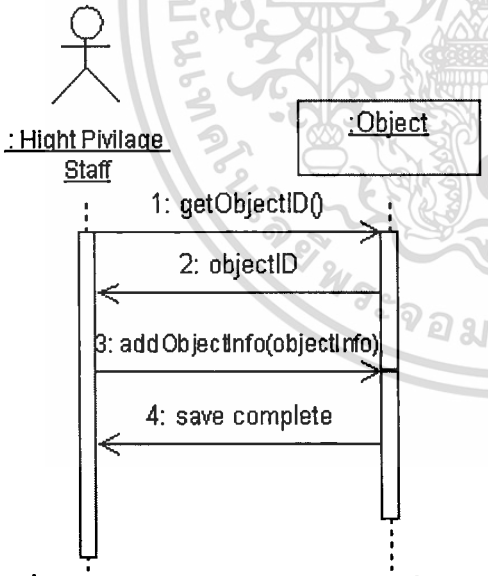


รูปที่ 3. 5 ซีเควนซ์ไดอะแกรมกำหนดพารามิเตอร์ระบบ

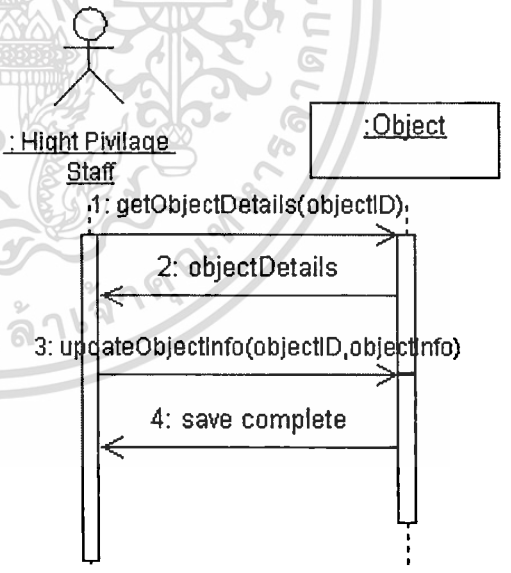


รูปที่ 3.6 ซีเควนซ์ไดอะแกรมปรับเปลี่ยนพารามิเตอร์ระบบ

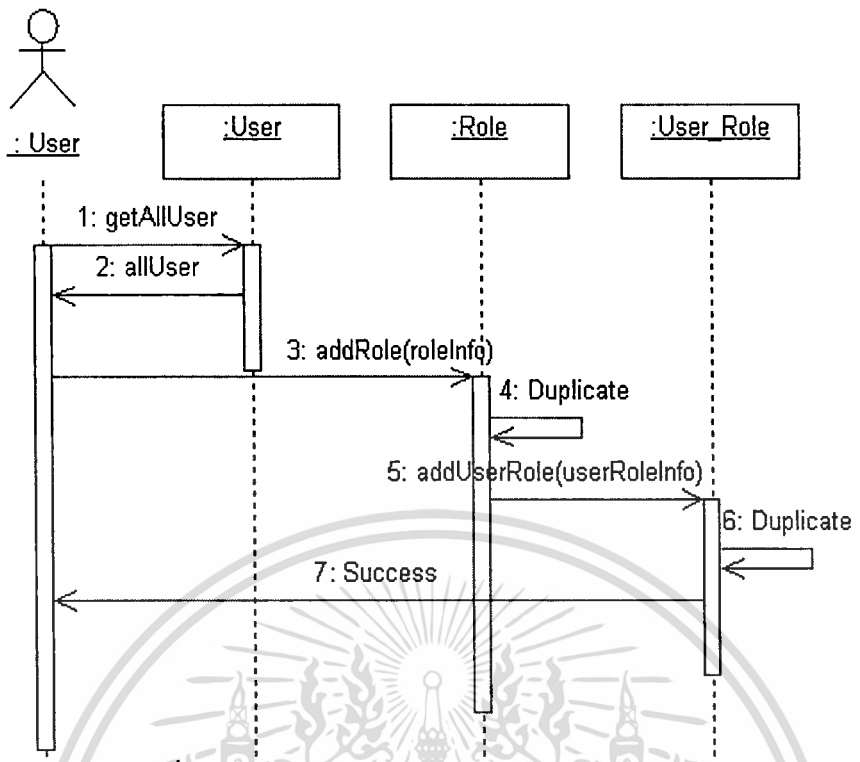
- ซีเควนซ์ไดอะแกรมการกำหนดสิทธิ หน้าที่ และการเปลี่ยนแปลงข้อมูลดังกล่าว ของผู้ใช้งานระบบแสดงตามรูปที่ 3.7 - 3.10



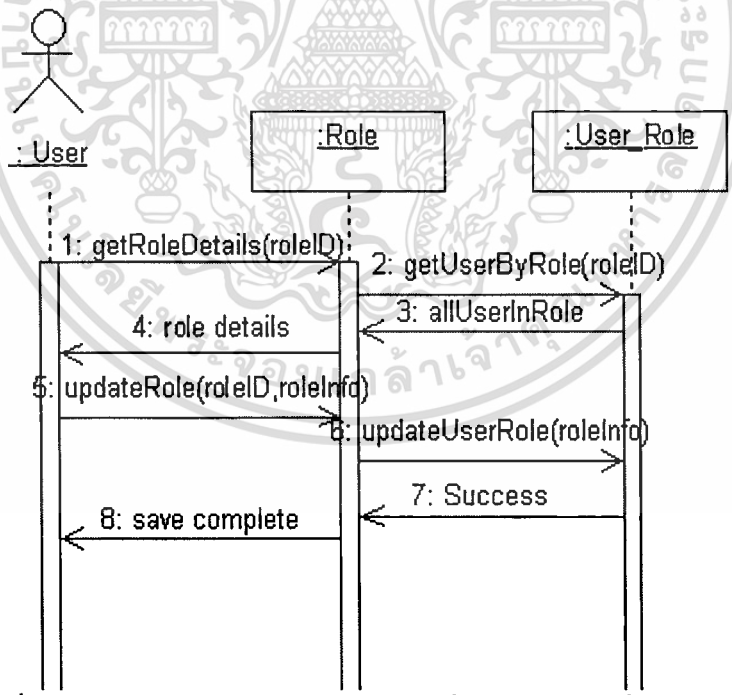
รูปที่ 3. 7 ซีเควนซ์ไดอะแกรมการเพิ่มข้อมูลสิทธิ



รูปที่ 3. 8 ซีเควนซ์ไดอะแกรมการปรับเปลี่ยนข้อมูลสิทธิ

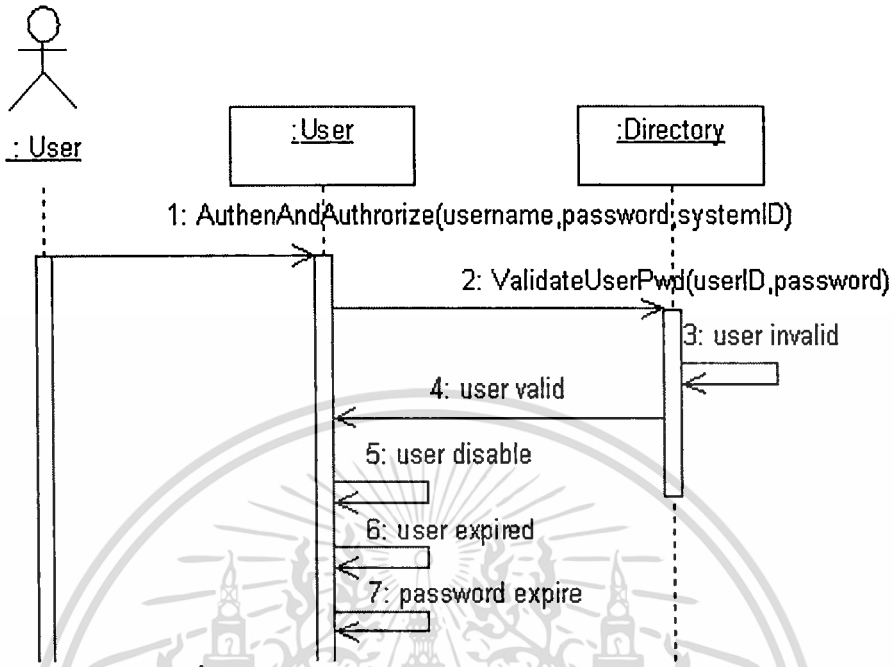


รูปที่ 3. 9 ซีเควนซ์ไดอะแกรมการเพิ่มหน้าที่ (Role)

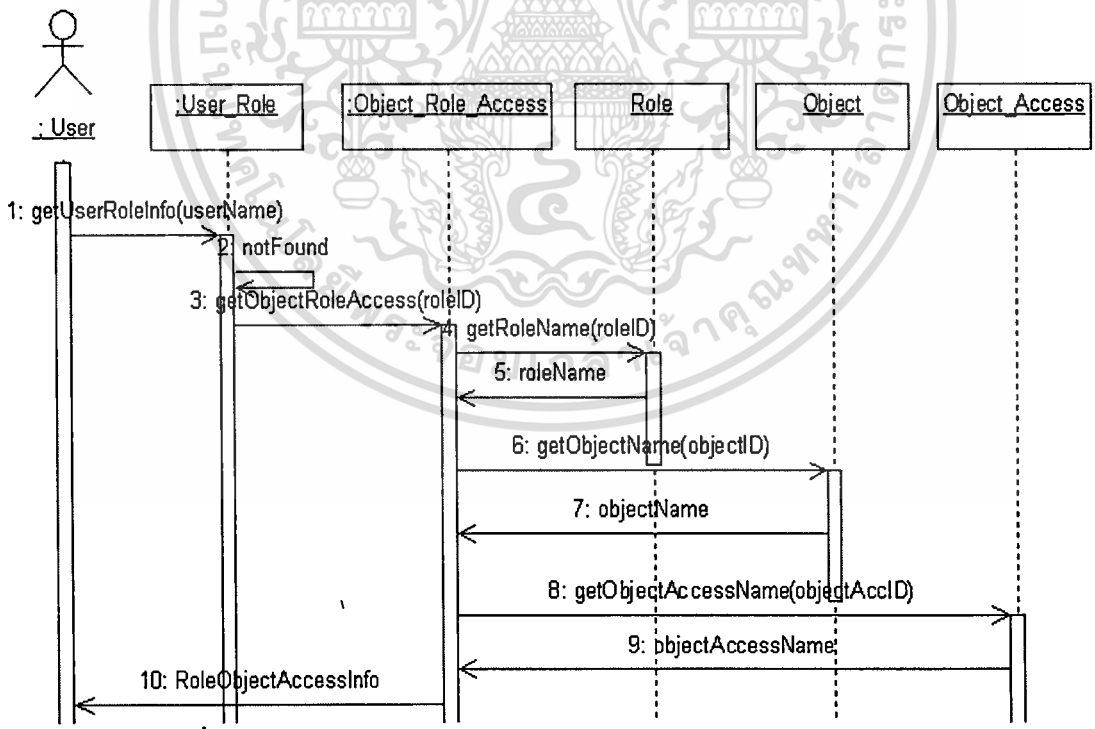


รูปที่ 3. 10 ซีเควนซ์ไดอะแกรมการปรับเปลี่ยนข้อมูลหน้าที่ (Role)

- ซีเควนซ์ไดอะแกรมการพิสูจน์ตัวตนและการตรวจสอบสิทธิการเข้าถึงทรัพยากรระบบแสดงตามรูปที่ 3.11 และรูปที่ 3.12

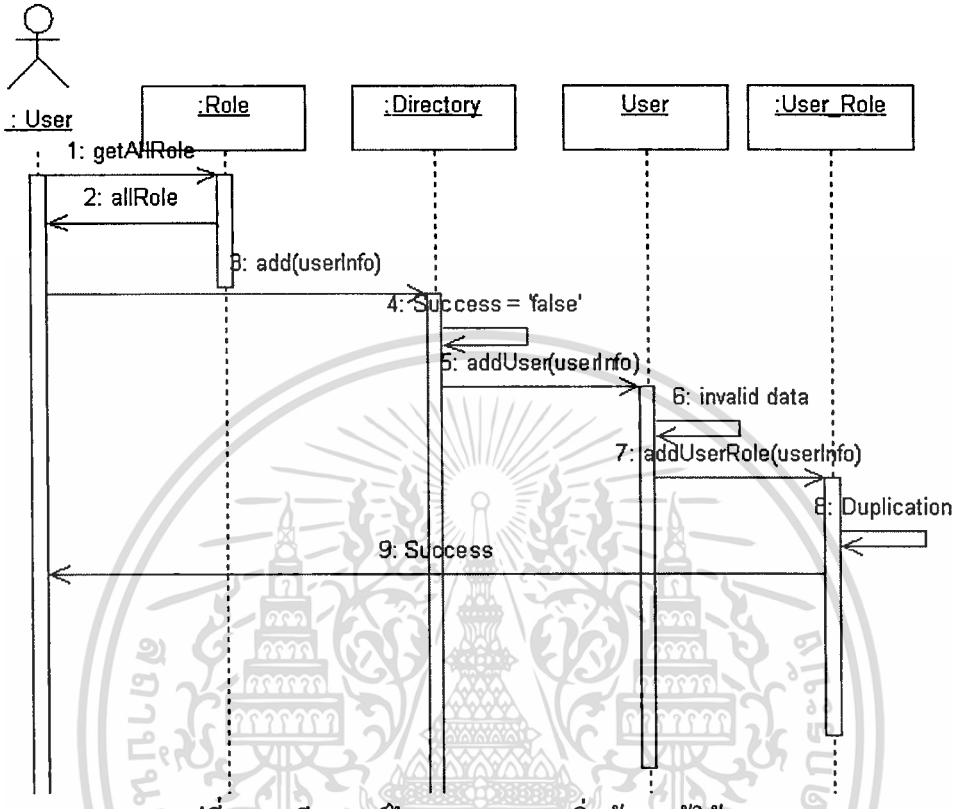


รูปที่ 3. 11 ซีเควนซ์ไดอะแกรมการพิสูจน์ตัวตน

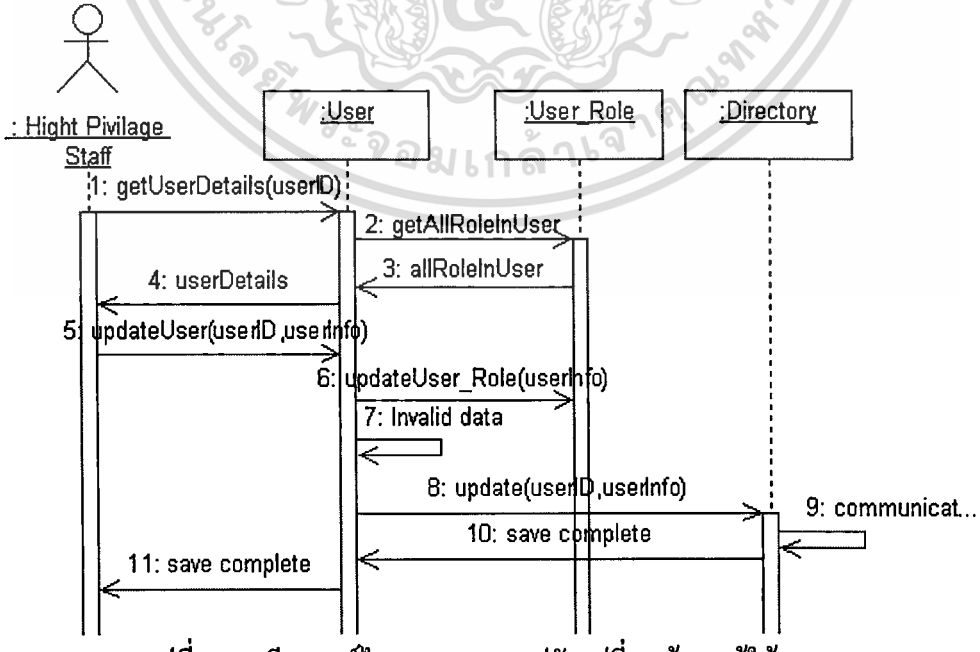


รูปที่ 3. 12 ซีเควนซ์ไดอะแกรมการตรวจสิทธิการเข้าถึงทรัพยากรระบบ

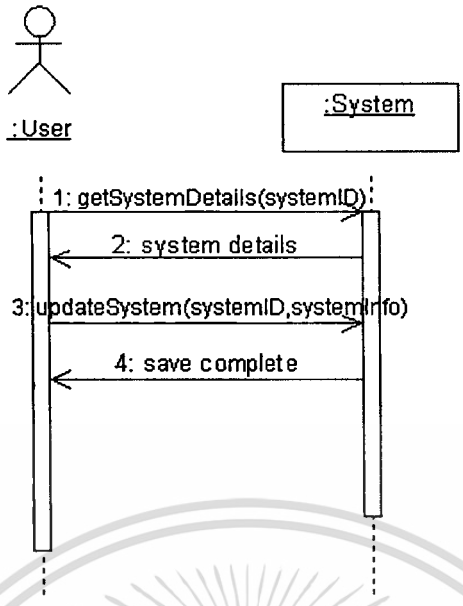
- ซีเควนซ์ไดอะแกรมการเพิ่มข้อมูลผู้ใช้งานระบบ เพิ่มระบบที่จะใช้บริการการพิสูจน์ตัวตน การสร้างความสัมพันธ์และการเปลี่ยนแปลงข้อมูลดังกล่าวแสดงตามรูปที่ 3.13 - 3.18



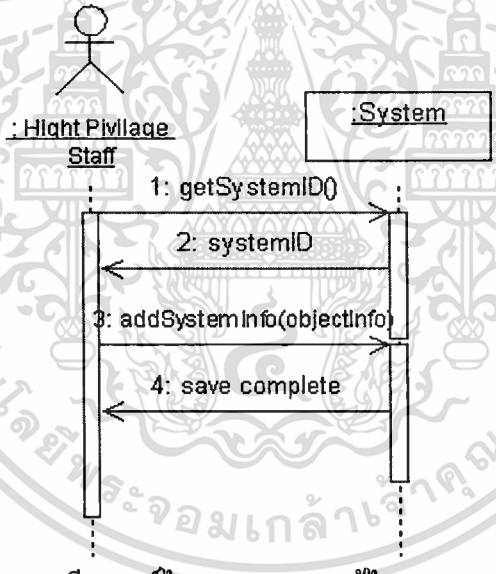
รูปที่ 3. 13 ซีเควนซ์ไดอะแกรมการเพิ่มข้อมูลผู้ใช้งาน



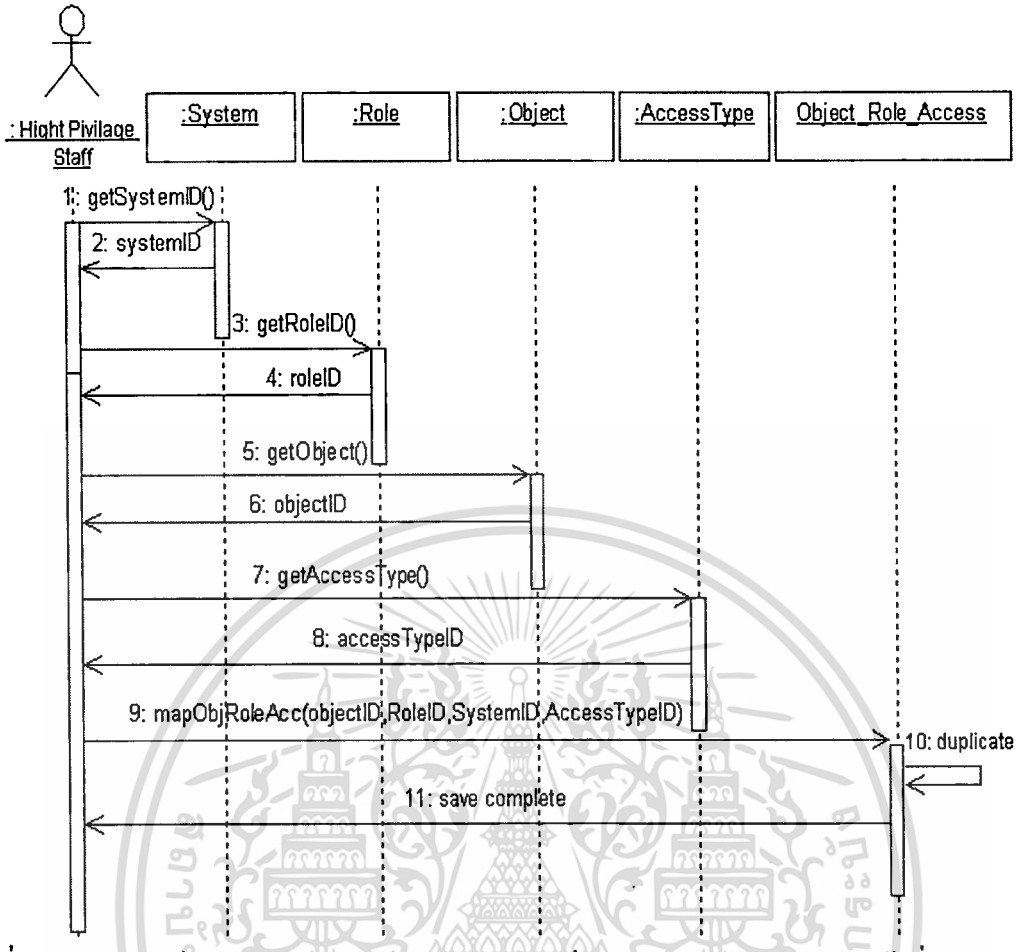
รูปที่ 3. 14 ซีเควนซ์ไดอะแกรมการปรับเปลี่ยนข้อมูลผู้ใช้งาน



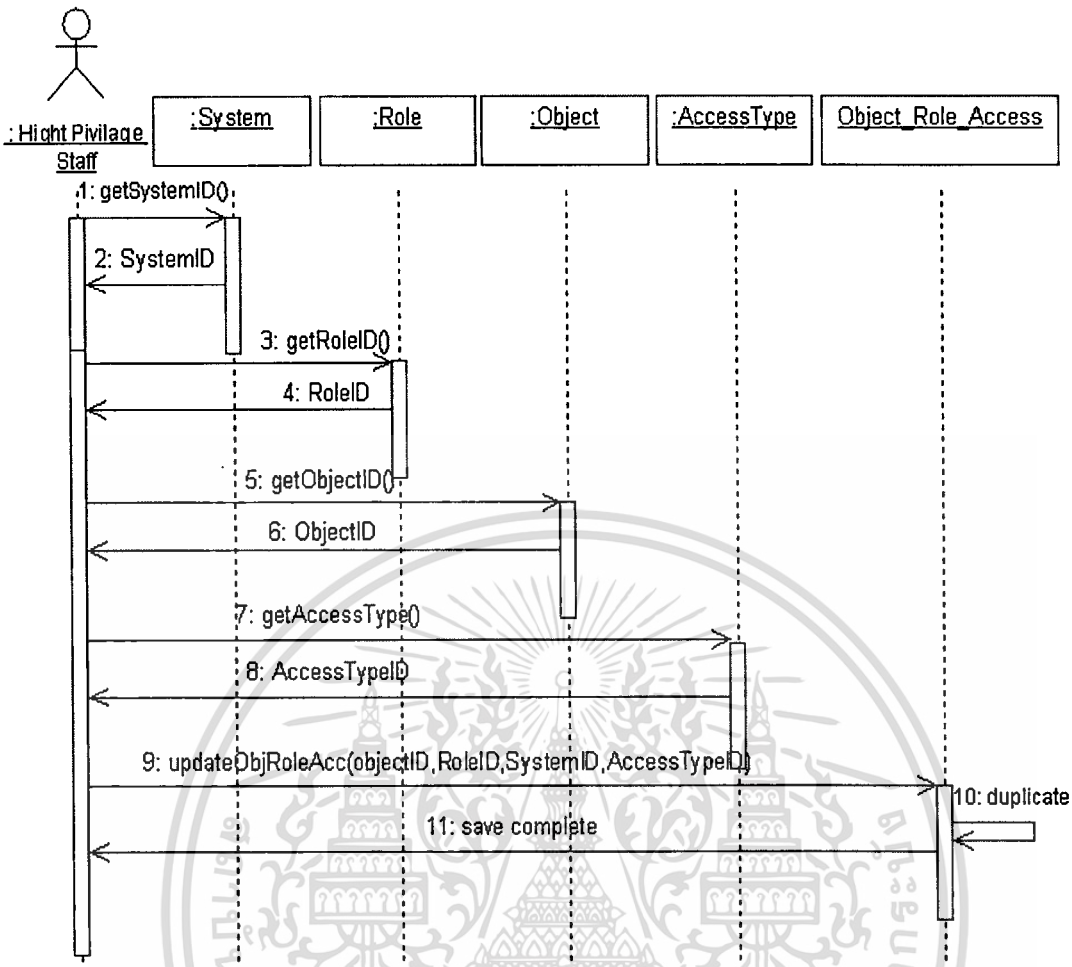
รูปที่ 3. 15 ซีเควนซ์ไดอะแกรมการสร้างระบบ (System)



รูปที่ 3. 16 ซีเควนซ์ไดอะแกรมการแก้ไขระบบ (System)



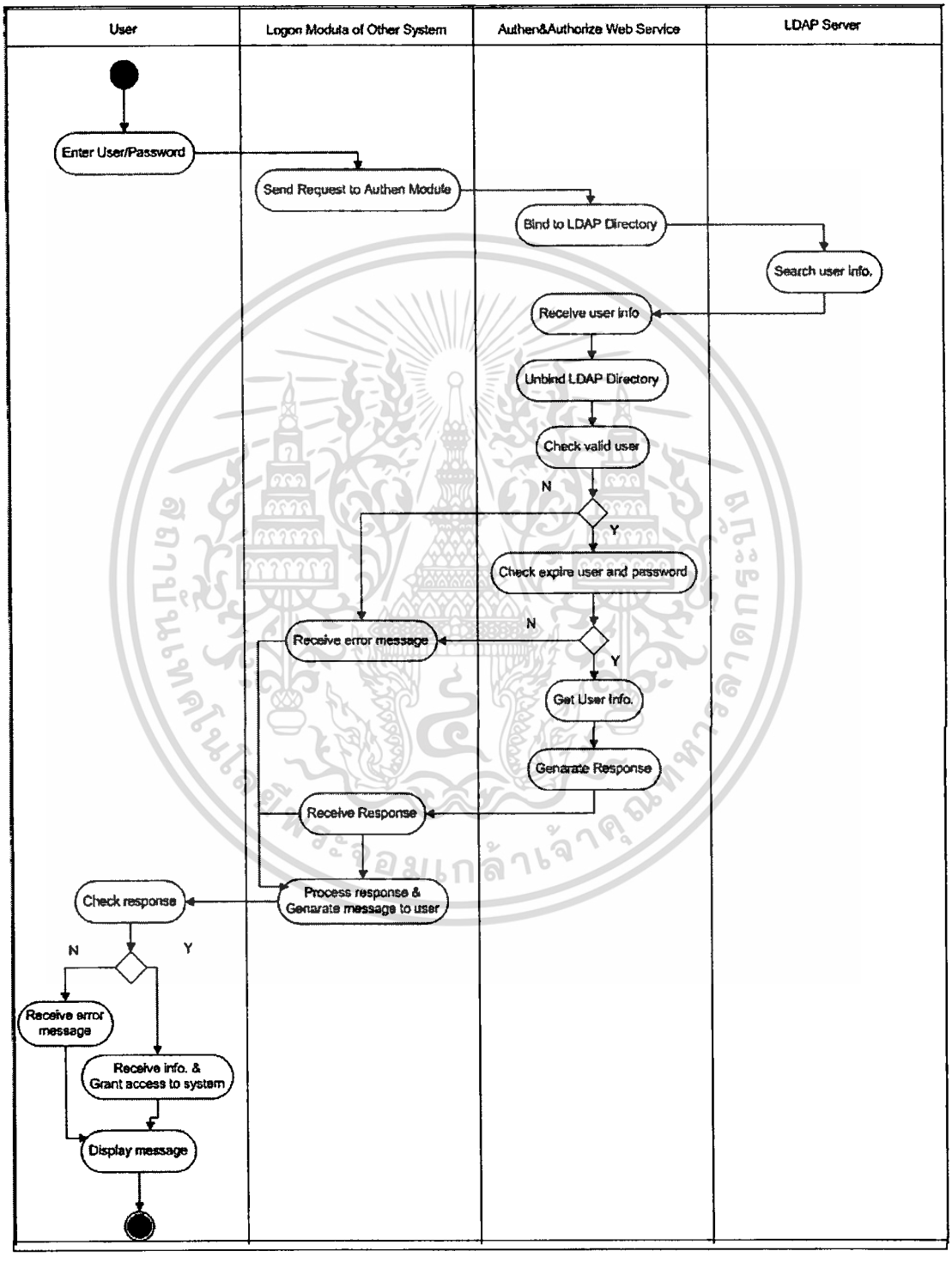
รูปที่ 3. 17 ซีเควนซ์ไดอะแกรมการสร้างความสัมพันธ์ระหว่างระบบ (System) หน้าที่ (Role) สิทธิ (Object) และรูปแบบการเข้าถึง



รูปที่ 3. 18 ซีเควนซ์ไดอะแกรมการแก้ไขความสัมพันธ์ระหว่างระบบ (System) หน้าที่ (Role) สิทธิ (Object) และรูปแบบการเข้าถึง

3.7 การออกแบบด้วยแอ็คทีวิตี้ไดอะแกรม

แอ็คทีวิตี้ไดอะแกรมแสดงการทำงานของการพิสูจน์ตัวตนและการกำหนดสิทธิการเข้าถึงระบบแสดงตามรูปที่ 3.19



รูปที่ 3. 19แอ็คทีวิตี้ไดอะแกรมของการพิสูจน์ตัวตนและการกำหนดสิทธิ

บทที่ 4

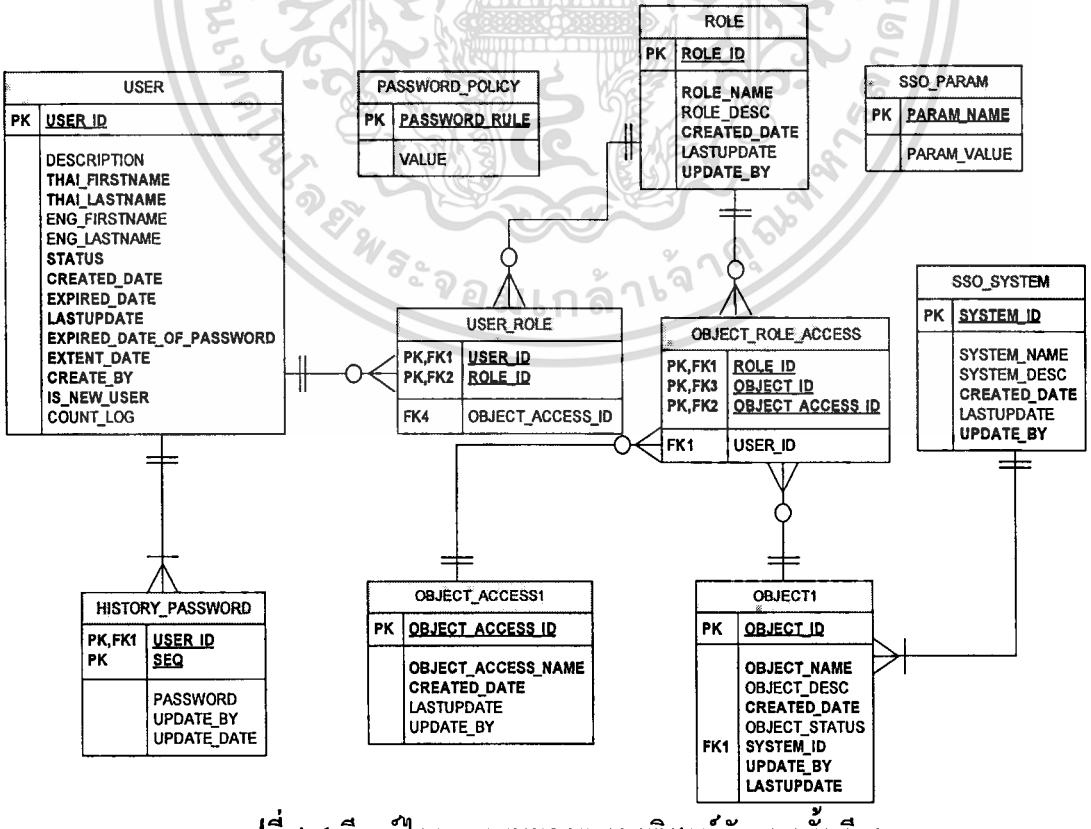
การออกแบบการเก็บข้อมูล

ระบบพิสูจน์ตัวตนเพียงครั้งเดียวมีการนำเอาเทคโนโลยี LDAP ซึ่งเป็นโพรโตคอล ในการติดต่อและมีรูปแบบการเก็บข้อมูลแบบเฉพาะคือการเก็บข้อมูลในรูปแบบต้นไม้ (Tree) ซึ่งใช้เก็บข้อมูลในการพิสูจน์ตัวตนข้อมูลที่เป็นความลับเพราะเป็นการป้องกันข้อมูลมีผลทำให้ระบบมีความปลอดภัยมากขึ้น ในการเก็บข้อมูลแบบต้นไม้หรือการเก็บข้อมูลในไดเรคทอรี่มีทั้งข้อดีและข้อเสียตามที่ได้กล่าวไว้ในบทของทฤษฎีและนั้น ระบบพิสูจน์ตัวตนเพียงครั้งเดียวจึงนำการเก็บข้อมูลแบบรีเลชันนอลดาต้าเบสเข้ามาช่วยให้การพัฒนาการและการปรับปรุงระบบทำได้ง่ายขึ้น ดังนั้นการเก็บข้อมูลในระบบการพิสูจน์ตัวตนเพียงครั้งเดียวนั้นจะมีแบ่งเป็น 2 แบบ

4.1 การเก็บข้อมูลแบบด้วยฐานข้อมูลรีเลชันนอลดาต้าเบส

4.1.1 อีอาร์ไดอะแกรม

อีอาร์ไดอะแกรมแสดงการเก็บข้อมูลและความสัมพันธ์กันของข้อมูลในระบบพิสูจน์ตัวตนครั้งเดียวแสดงตามรูปที่ 4.1



รูปที่ 4. 1 อีอาร์ไดอะแกรมของระบบพิสูจน์ตัวตนครั้งเดียว

จากรูปที่ 4.1 มีเอนทิตีทั้งหมด 10 เอนทิตี แต่ละเอนทิตีเก็บข้อมูลดังต่อไปนี้

- เอนทิตี USERS เก็บข้อมูลรายละเอียดของผู้ใช้งาน
- เอนทิตี ROLE เก็บข้อมูลหน้าที่ของระบบสามารสนเทศต่างๆ สำหรับการกำหนดสิทธิ
- เอนทิตี USER_ROLE เก็บข้อมูลความสัมพันธ์ระหว่างผู้ใช้งานกับหน้าที่เพื่อการกำหนดสิทธิ
- เอนทิตี OBJECT เก็บข้อมูลสิทธิในการเข้าถึงระบบสารสนเทศ สำหรับการกำหนดสิทธิ
- เอนทิตี OBJECT_ACCESS เก็บประเภทของสิทธิในการเข้าถึงทรัพยากรของระบบสารสนเทศ
- เอนทิตี OBJECT_ROLE_ACCESS เก็บข้อมูลความสัมพันธ์ของสิทธิ หน้าที่และประเภทในการเข้าถึงทรัพยากรระบบ
- เอนทิตี SSO_SYSTEM เก็บข้อมูลระบบสารสนเทศที่เข้าใช้ระบบการพิสูจน์ตัวตนเพียงครั้งเดียว
- เอนทิตี HISTORY_PASSWORD เก็บข้อมูลประวัติของชุดรหัสผ่านของผู้ใช้งานระบบ
- เอนทิตี PASSWORD_POLICY เก็บพารามิเตอร์ของระบบเพื่อใช้ในการกำหนดกฎเกณฑ์ของการตั้งชุดรหัสผ่าน
- เอนทิตี SSO_PARAM เก็บพารามิเตอร์ต่างๆ ของระบบพิสูจน์ตัวตนเพียงครั้งเดียว

แต่ละเอนทิตีมีความสัมพันธ์กันดังนี้

- เอนทิตี USER กับเอนทิตี ROLE มีความสัมพันธ์กัน โดย แต่ละผู้ใช้งานสามารถกำหนดหน้าที่ได้ หลายหน้าที่ สำหรับเข้าใช้ระบบสามารถสนเทศต่างๆ
- เอนทิตี OBJECT กับเอนทิตี OBJECT_ACCESS มีความสัมพันธ์กัน โดย แต่ละสิทธิสามารถมีประเภทในการเข้าถึงทรัพยากรได้มากกว่าหนึ่งสิทธิ เพื่อการกำหนดระดับการเข้าถึงทรัพยากรระบบ
- เอนทิตี ROLE, OBJECT และ OBJECT_ACCESS มีความสัมพันธ์กันหนึ่งหน้าที่สามารถกำหนดสิทธิและระดับการเข้าถึงทรัพยากรระบบ ได้เพียงหนึ่งความสัมพันธ์
- เอนทิตี SSO_SYSTEM กับเอนทิตี OBJECT มีความสัมพันธ์กัน โดยหนึ่ง OBJECT สามารถกำหนดให้กัน ระบบสารสนเทศได้หนึ่งระบบไม่ซ้ำกัน

4.1.2 พจนานุกรมข้อมูล

จากอีอาร์ไดอะแกรมที่มีทั้งหมด 10 เอนทิตี สามารถกำหนดคุณลักษณะของแต่ละเอนทิตีได้ ดังตารางที่ 4.1 ถึง 4.10 ดังนี้

ตารางที่ 4. 1 รายละเอียดตาราง USER

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
USER_ID	รหัสผู้ใช้งาน	VARCHAR2(50)	PK	
DESCRIPTION	รายละเอียดผู้ใช้งาน	VARCHAR2(100)		
THAI_FIRSTNAME	ชื่อผู้ใช้งาน(ไทย)	VARCHAR2(50)		
THAI_LASTNAME	นามสกุลผู้ใช้งาน(ไทย)	VARCHAR2(50)		
ENG_FIRSTNAME	ชื่อผู้ใช้งาน(อังกฤษ)	VARCHAR2(50)		
ENG_LASTNAME	นามสกุลผู้ใช้งาน(อังกฤษ)	VARCHAR2(50)		
STATUS	สถานะผู้ใช้งาน	VARCHAR2(5)		
CREATED_DATE	วันที่สร้างบัญชีผู้ใช้งาน	DATE		
EXPIRED_DATE	วันที่บัญชีผู้ใช้งานหมดอายุ	DATE		
LASTUPDATE	วันที่สุดท้ายที่มีการแก้ไข	DATE		
EXPIRED_DATE_OF_PASSWORD	วันที่หมดอายุของรหัสผ่าน	DATE		
EXTENT_DATE	วันที่ระบบต่อเวลาการใช้งาน	NUMBER		
CREATED_BY	ชื่อผู้สร้างบัญชีผู้ใช้นี้	VARCHAR2(50)		
IS_NEW_USER	บัญชีผู้ใช้งานนี้พึ่งสร้างใหม่	VARCHAR2(5)		
COUNT_LOG	จำนวนการล๊อกอินผิด	NUMBER		

ตารางที่ 4. 2 รายละเอียดตาราง ROLE

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
ROLE_ID	รหัสหน้าที่	VARCHAR2(10)	PK	
ROLE_NAME	รายชื่อหน้าที่	VARCHAR2(100)		
ROLE_DESC	รายละเอียดหน้าที่	VARCHAR2(300)		
CREATED_DATE	วันที่สร้างรายชื่อหน้าที่	DATE		
LASTUPDATE	วันสุดท้ายที่ทำการแก้ไข	DATE		
UPDATE_BY	รายชื่อผู้แก้ไขข้อมูล	VARCHAR2(50)		

ตารางที่ 4. 3 รายละเอียดตาราง OBJECT

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
OBJECT_ID	รหัสสิทธิ	VARCHAR2(10)	PK	
ROLE_NAME	รายชื่อสิทธิ	VARCHAR2(100)		
ROLE_DESC	รายละเอียดสิทธิ	VARCHAR2(300)		
CREATED_DATE	วันที่สร้างรายชื่อสิทธิ	DATE		
OBJECT_STATUS	สถานะของสิทธิ	VARCHAR2(1)		
SYSTEM_ID	ระบบที่เกี่ยวข้องกับสิทธิ	VARCHAR2(10)	FK	SSO_SYSTEM
LASTUPDATE	วันสุดท้ายที่ทำการแก้ไข	DATE		
UPDATE_BY	รายชื่อผู้แก้ไขข้อมูล	VARCHAR2(50)		

ตารางที่ 4. 4 รายละเอียดตาราง OBJECT_ACCESS

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
OBJECT_ACCESS_ID	รหัสระดับของสิทธิ	VARCHAR2(10)	PK	
OBJECT_ACCESS_NAME	รายชื่อระดับของสิทธิ	VARCHAR2(100)		
CREATED_DATE	วันที่สร้างระดับของสิทธิ	DATE		
LASTUPDATE	วันสุดท้ายที่ทำการแก้ไข	DATE		
UPDATE_BY	รายชื่อผู้แก้ไขข้อมูล	VARCHAR2(50)		

ตารางที่ 4. 5 รายละเอียดตาราง SSO_SYSTEM

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
SYSTEM_ID	รหัสระบบสารสนเทศ	VARCHAR2(10)	PK	
SYSTEM _NAME	รายชื่อระบบสารสนเทศ	VARCHAR2(50)		
SYSTEM_DESC	รายละเอียดระบบสารสนเทศ	VARCHAR2(300)		

ตารางที่ 4. 6 รายละเอียดตาราง SSO_SYSTEM (ต่อ)

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
CREATED_DATE	วันที่สร้างระบบสารสนเทศ	DATE		
LASTUPDATE	วันสุดท้ายที่ทำการแก้ไข	DATE		
UPDATE_BY	รายชื่อผู้แก้ไขข้อมูล	VARCHAR2(50)		

ตารางที่ 4. 7 รายละเอียดตาราง OBJECT_ROLE_ACCESS

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
ROLE_ID	รายชื่อหน้าที่	VARCHAR2(10)	PK,FK	ROLE
OBJECT_ID	รายชื่อสิทธิ	VARCHAR2(10)	PK,FK	OBJECT
OBJECT_ACCESS_ID	รายชื่อระดับของสิทธิ	VARCHAR2(10)	PK,FK	OBJECT_ACESS

ตารางที่ 4. 8 รายละเอียดตาราง HSITORY_PASSWORD

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
USER_ID	รหัสผู้ใช้งาน	VARCHAR2(10)	PK,FK	USER
PASSWORD	ประวัติรหัสผ่านชุดเก่า	VARCHAR2(100)	PK	
UPDATE_BY	รายชื่อผู้แก้ไข	DATE		
UPDATE_DATE	วันสุดท้ายที่ทำการแก้ไข	DATE		

ตารางที่ 4. 9 รายละเอียดตาราง USER_ROLE

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
USER_ID	รหัสผู้ใช้งาน	VARCHAR2(10)	PK,FK	USER
ROLE_ID	รหัสหน้าที่	VARCHAR2(10)	PK,FK	ROLE

ตารางที่ 4. 10 รายละเอียดตาราง PASSWORD_POLICY

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
PASSWORD_RULE	ข้อบังคับในการตั้งรหัสผ่าน	VARCHAR2(30)	PK	
VALUE	ค่าของข้อบังคับ	VARCHAR2(10)	PK,FK	

ตารางที่ 4. 11 รายละเอียดตาราง SSO_PARAM

ชื่อแอตทริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางที่อ้างถึง
PARAM_NAME	พารามิเตอร์ของระบบ SSO	VARCHAR2(50)	PK	
PARAM_VAULE	ค่าพารามิเตอร์	VARCHAR2(50)		

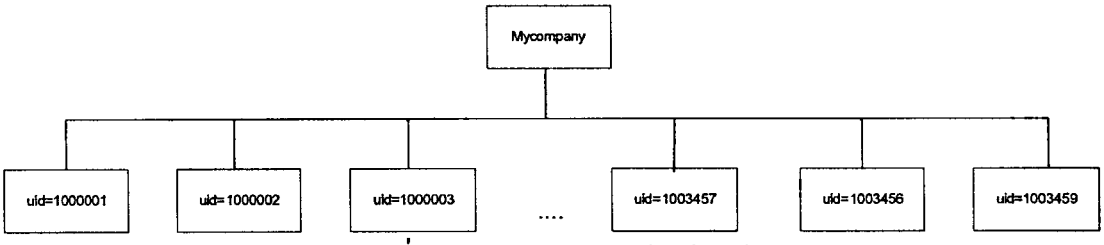
4.2 การเก็บข้อมูลแบบไดเรคทอรี่

4.2.1 โครงสร้างของ Schema ใช้ schema ที่เป็นพื้นฐานของ LDAP ประกอบด้วย

- objectclass (1.3.6.1.1.3.1 NAME 'uidObject'
  DESC 'RFC2377: uid object'
  SUP top AUXILIARY MUST uid)
- objectclass (2.5.6.5 NAME 'organizationalUnit'
  DESC 'RFC2256: an organizational unit'
  SUP top STRUCTURAL
  MUST ou
  MAY (userPassword $ searchGuide $ seeAlso $ businessCategory $
  x121Address $ registeredAddress $ destinationIndicator $
  preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $
  telephoneNumber $ internationaliSDNNumber $
  facsimileTelephoneNumber $ street $ postOfficeBox $ postalCode $
  postalAddress $ physicalDeliveryOfficeName $ st $ l $ description))

4.2.2 โครงสร้างต้นไม้ (Tree)

โครงสร้างต้นไม้ในระบบจะเก็บข้อมูลรหัสผู้ใช้งาน และรหัสผ่านชุดปัจจุบัน ภายใต้องค์กรหนึ่งๆ แสดงดังรูปที่ 4.2



รูปที่ 4. 2 ตัวอย่างโครงสร้างต้นไม้

บทที่ 5
การพัฒนาระบบ

5.1 เครื่องมือที่ใช้ในการพัฒนาระบบ

5.1.1 ฮาร์ดแวร์

ในการพัฒนาระบบงานใช้เครื่องคอมพิวเตอร์ที่มีคุณสมบัติดังนี้

- CPU: Intel Centrino Duo T2300 1.66GHz.
- Hard disk 80 GB.
- RAM 1.25 GB.

5.1.2 ซอฟต์แวร์

ในการพัฒนาระบบงานใช้ซอฟต์แวร์ดังนี้

- Windows XP
- Eclipse 3.3.1 Europa
- Apache Tomcat 5.5
- JDK 6
- Windows 2000 เซิร์ฟเวอร์
- Oracle 9i
- LDAPBrowser
- soapUI 2.0

5.2 รายละเอียดของการทำงานของระบบ

โครงงานพัฒนาระบบพิสูจน์ตัวตนเพียงครั้งเดียว โดยเว็บเซอร์วิสมีระบบเว็บแอพพลิเคชันเป็นระบบที่ใช้ติดต่อกับผู้ใช้งาน มีรายละเอียดหน้าจอการทำงาน ดังต่อไปนี้

หน้าจอหลักของระบบพิสูจน์ตัวตนเพียงครั้งเดียว

หน้าจอล็อกออนเข้าสู่ระบบพิสูจน์ตัวตน

เป็นหน้าจอในการเข้าสู่ระบบการจัดการการพิสูจน์ตัวตนเพียงรั้งเดียว ดังรูปที่ 5.1

รูปที่ 5. 1หน้าจอการเข้าสู่ระบบการจัดการการพิสูจน์ตัวตนเพียงครั้งเดียว

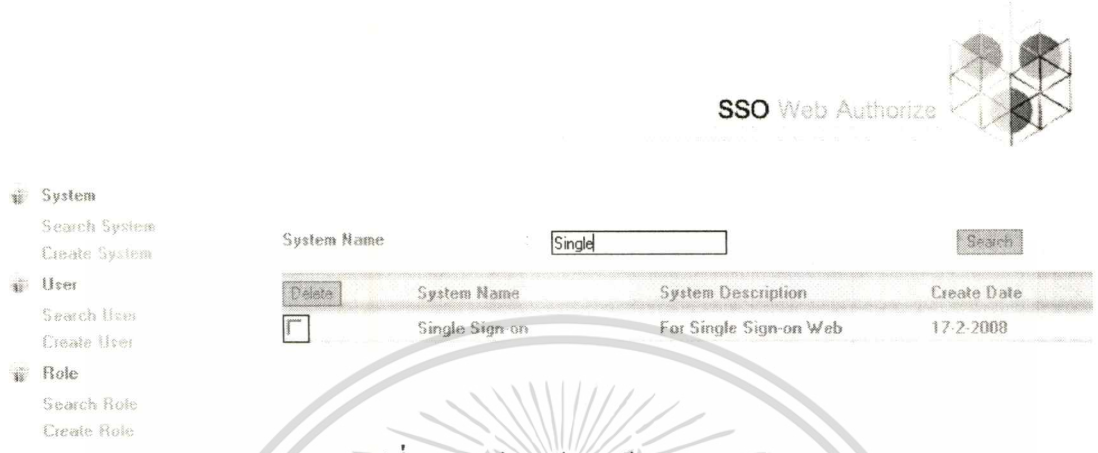
หน้าจอสร้างระบบ

เป็นหน้าจอที่ใช้ในการสร้างข้อมูลระบบที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.2

รูปที่ 5. 2 หน้าจอสร้างระบบ

หน้าจอค้นหาข้อมูลระบบ

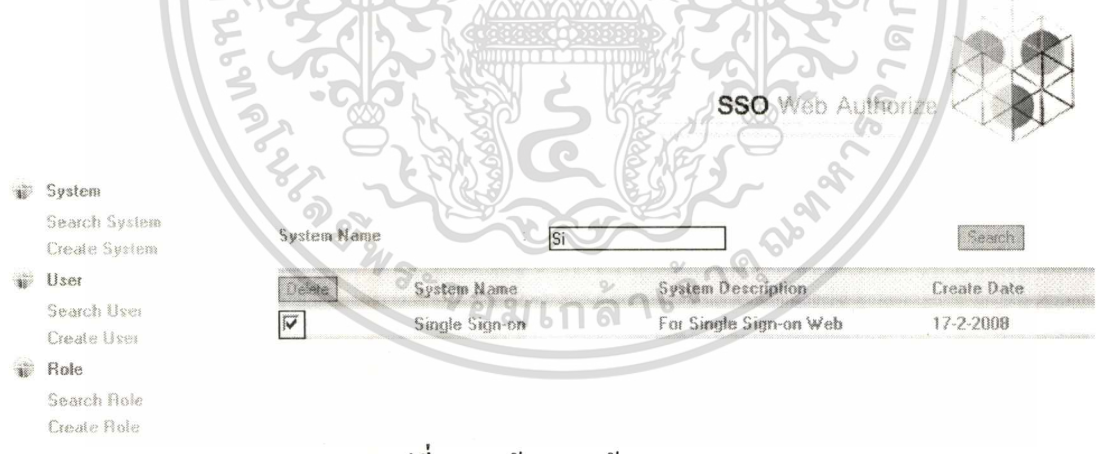
เป็นหน้าจอที่ใช้ในการค้นหาข้อมูลระบบที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.3



รูปที่ 5. 3 หน้าจอค้นหาข้อมูลระบบ

หน้าจอลบข้อมูลระบบ

เป็นหน้าจอที่ใช้ในการลบข้อมูลระบบที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.4



รูปที่ 5. 4หน้าจอลบข้อมูลระบบ

หน้าจอแก้ไขข้อมูลระบบ

เป็นหน้าจอที่ใช้ในการแก้ไขข้อมูลระบบที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.5

รูปที่ 5. 5 หน้าจอแก้ไขข้อมูลระบบ

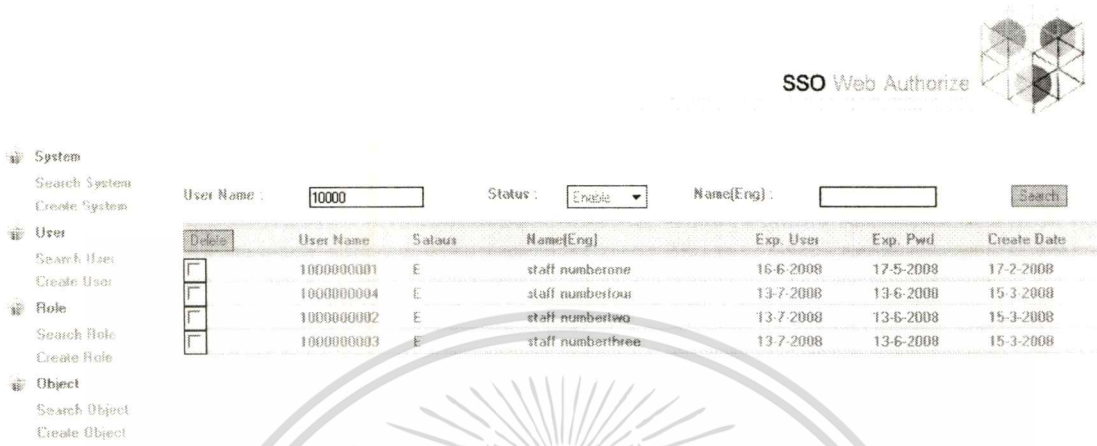
หน้าจอสร้างข้อมูลผู้ใช้งาน

เป็นหน้าจอที่ใช้ในการสร้างข้อมูลผู้ใช้งานที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.6

รูปที่ 5. 6 หน้าจอสร้างข้อมูลผู้ใช้งาน

หน้าจอค้นหาข้อมูลผู้ใช้งาน

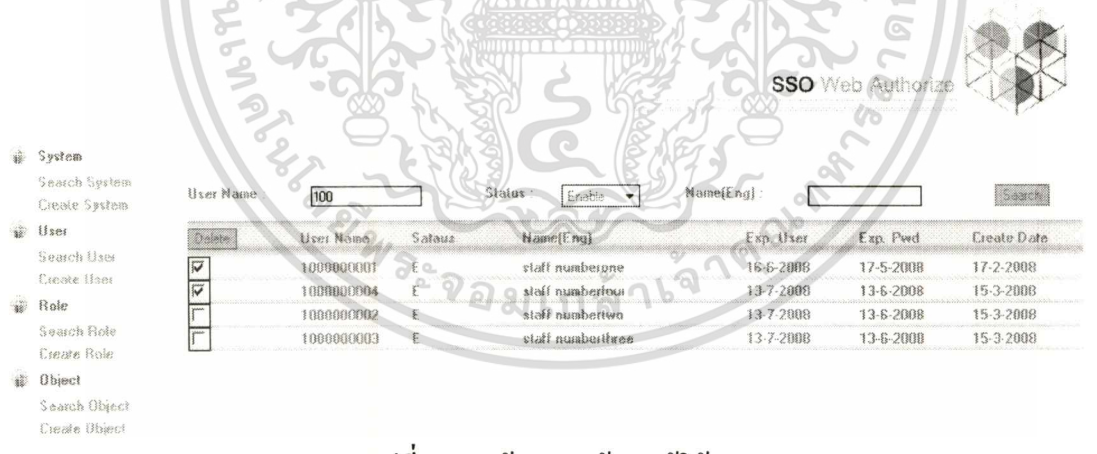
เป็นหน้าจอที่ใช้ในการค้นหาข้อมูลผู้ใช้งานที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.7



รูปที่ 5. 7 หน้าจอค้นหาข้อมูลผู้ใช้งาน

หน้าจอลบข้อมูลผู้ใช้งาน

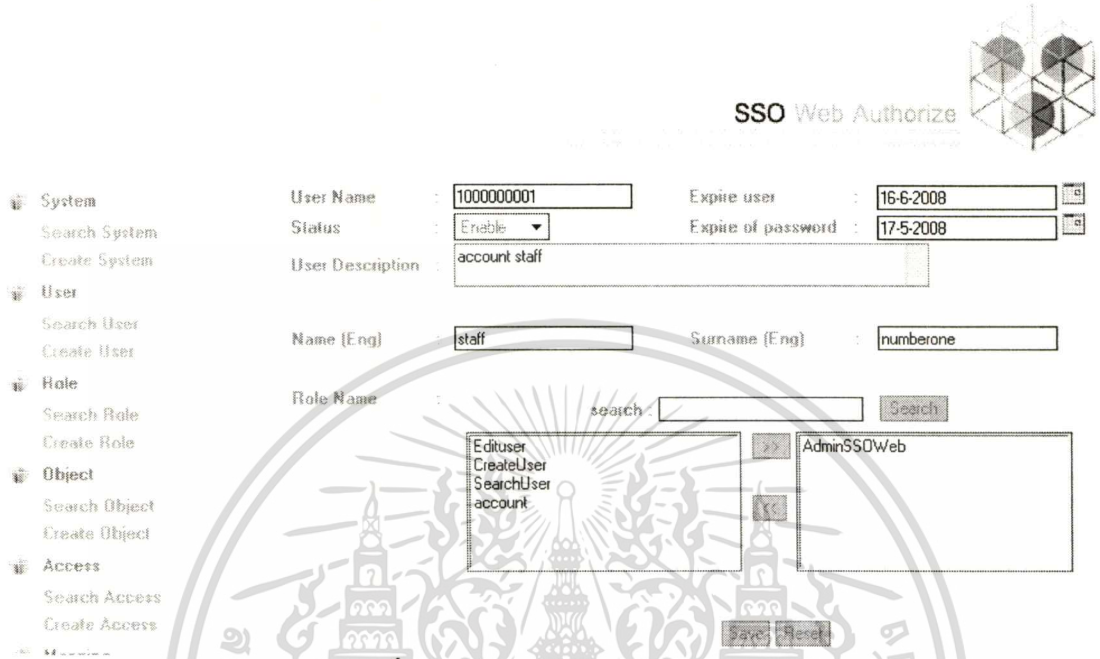
เป็นหน้าจอที่ใช้ในการลบข้อมูลผู้ใช้งานที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.8



รูปที่ 5. 8 หน้าจอลบข้อมูลผู้ใช้งาน

หน้าจอแก้ไขข้อมูลผู้ใช้งาน

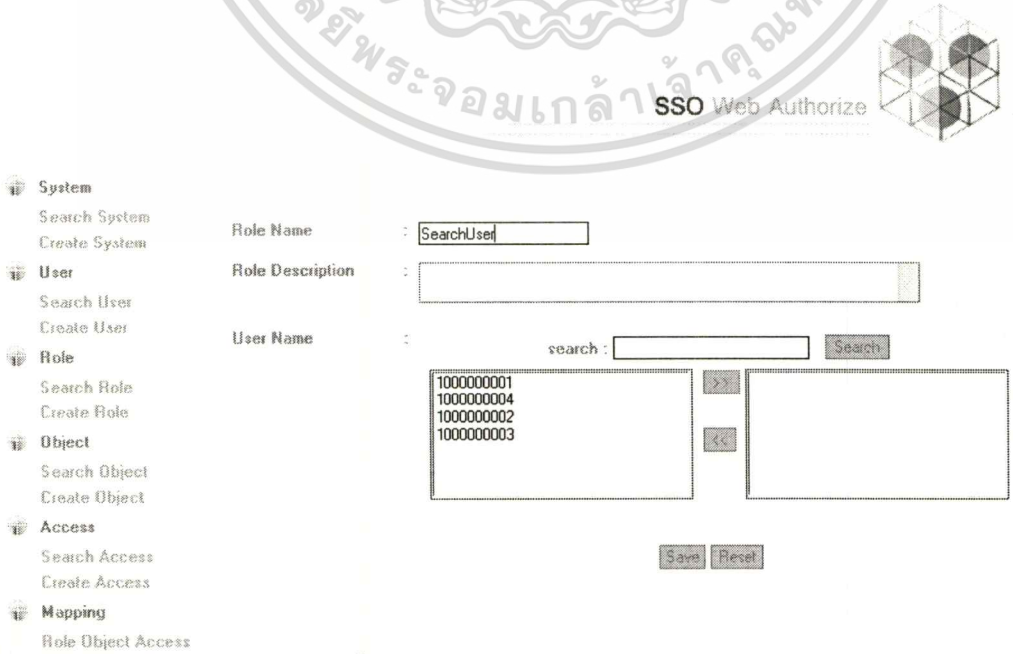
เป็นหน้าจอที่ใช้ในการแก้ไขข้อมูลผู้ใช้งานที่จะเข้ามารับบริการกับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.9



รูปที่ 5. 9 หน้าจอแก้ไขข้อมูลผู้ใช้งาน

หน้าจอสร้างข้อมูลหน้าที่ (Role)

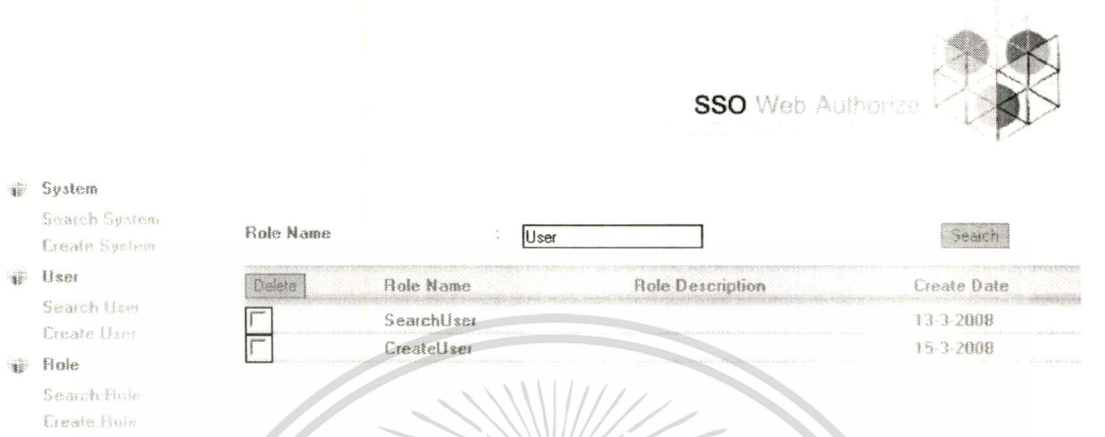
เป็นหน้าจอที่ใช้ในการสร้างข้อมูลหน้าที่ (Role) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.10



รูปที่ 5. 10 หน้าจอสร้างข้อมูลหน้าที่ (Role)

หน้าจอค้นหาข้อมูลหน้าที่ (Role)

เป็นหน้าจอที่ใช้ในการค้นหาข้อมูลหน้าที่ (Role) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.11



รูปที่ 5. 11 หน้าจอค้นหาข้อมูลหน้าที่ (Role)

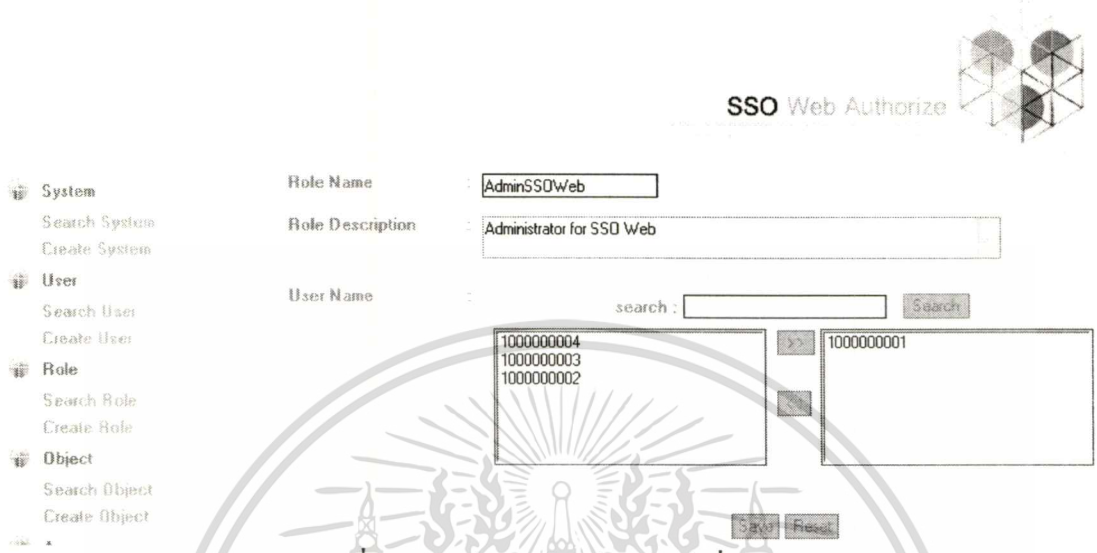
หน้าจอลบข้อมูลหน้าที่ (Role)

เป็นหน้าจอที่ใช้ในการลบข้อมูลหน้าที่ (Role) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.12

รูปที่ 5. 12 หน้าจอลบข้อมูลหน้าที่ (Role)

หน้าจอแก้ไขข้อมูลหน้าที่ (Role)

เป็นหน้าจอที่ใช้ในการแก้ไขข้อมูลหน้าที่ (Role) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.13



รูปที่ 5. 13 หน้าจอแก้ไขข้อมูลหน้าที่ (Role)

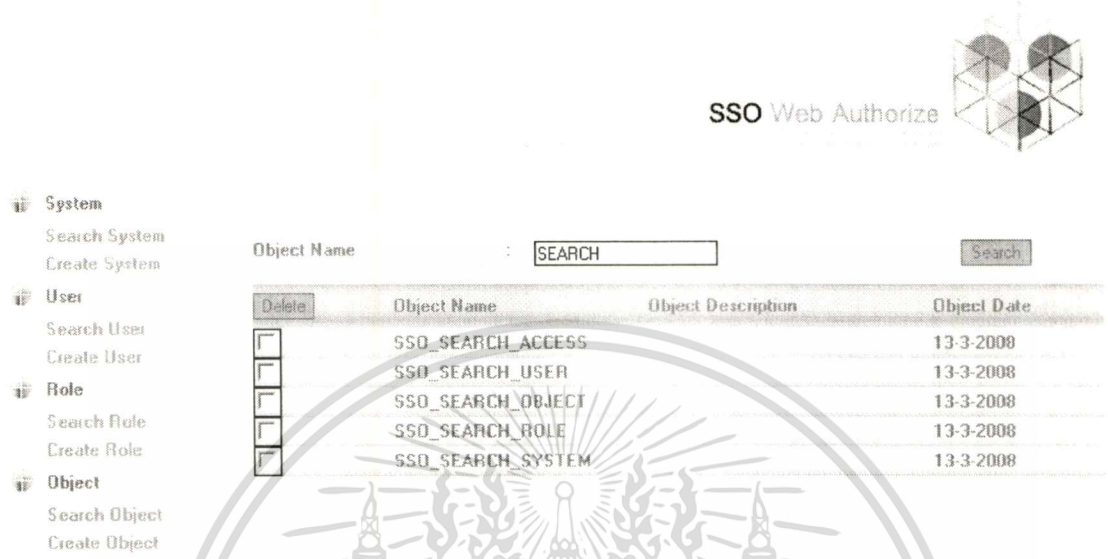
หน้าจอสร้างข้อมูลสิทธิ (Object)

เป็นหน้าจอที่ใช้ในการสร้างข้อมูลสิทธิ (Object) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.14

รูปที่ 5. 14 หน้าจอสร้างข้อมูลสิทธิ (Object)

หน้าจอค้นหาข้อมูลสิทธิ (Object)

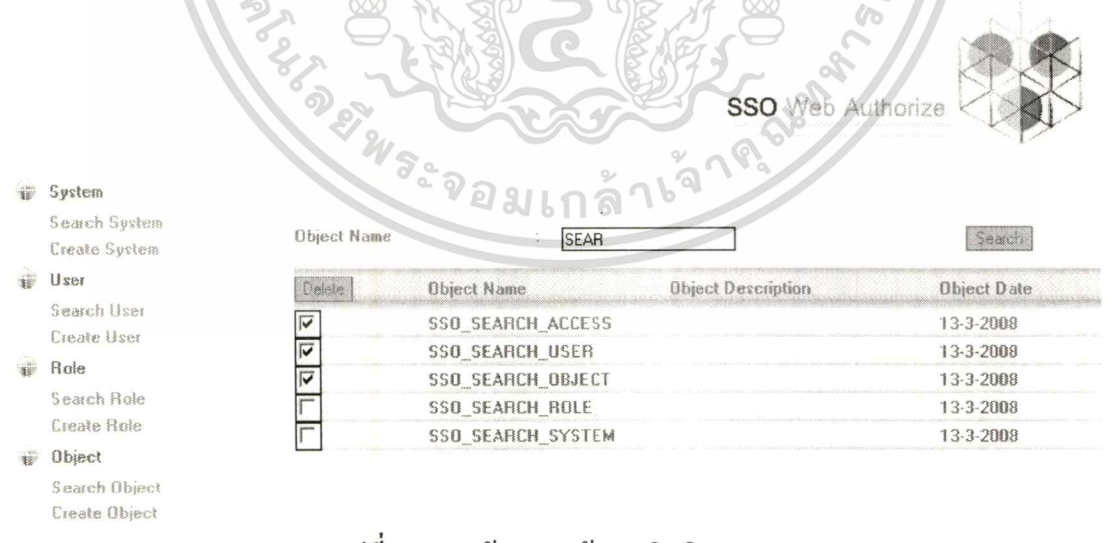
เป็นหน้าจอที่ใช้ในการค้นหาข้อมูลสิทธิ (Object) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.15



รูปที่ 5. 15 หน้าจอค้นหาข้อมูลสิทธิ (Object)

หน้าจอลบข้อมูลสิทธิ (Object)

เป็นหน้าจอที่ใช้ในการลบข้อมูลสิทธิ (Object) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.16



รูปที่ 5. 16 หน้าจอลบข้อมูลสิทธิ (Object)

หน้าจอแก้ไขข้อมูลสิทธิ (Object)

เป็นหน้าจอที่ใช้ในการแก้ไขข้อมูลสิทธิ (Object) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.17

รูปที่ 5. 17 หน้าจอแก้ไขข้อมูลสิทธิ (Object)

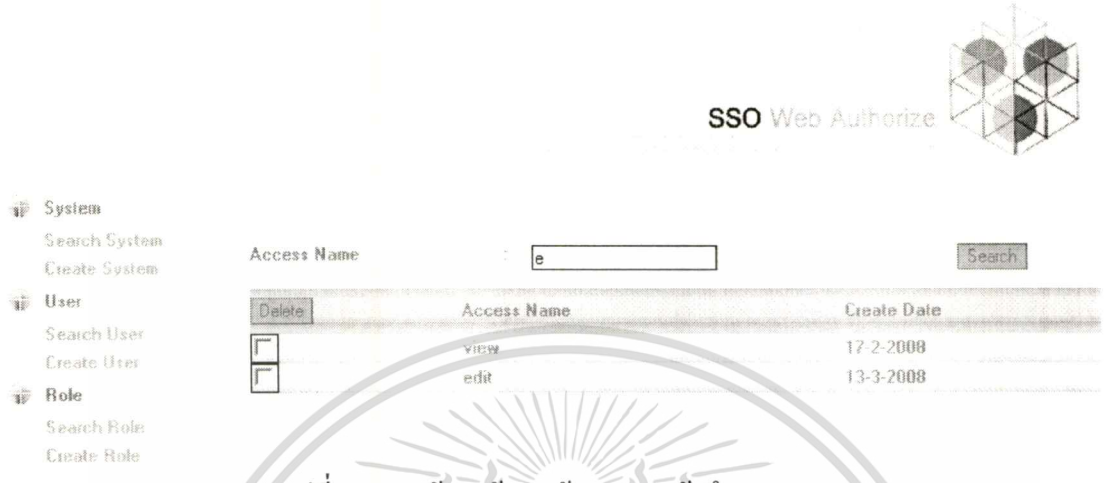
หน้าจอสร้างข้อมูลการเข้าถึง (Access)

เป็นหน้าจอที่ใช้ในการสร้างข้อมูลการเข้าถึง (Access) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.18

รูปที่ 5. 18 หน้าจอสร้างข้อมูลการเข้าถึง (Access)

หน้าจอค้นหาข้อมูลการเข้าถึง (Access)

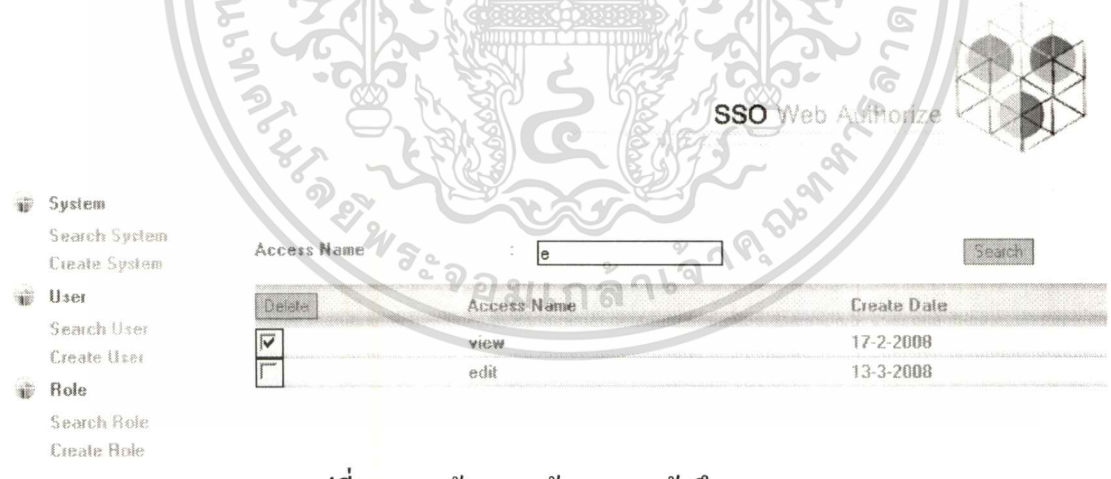
เป็นหน้าจอที่ใช้ในการค้นหาข้อมูลการเข้าถึง (Access) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.19



รูปที่ 5. 19 หน้าจอค้นหาข้อมูลการเข้าถึง (Access)

หน้าจอลบข้อมูลการเข้าถึง (Access)

เป็นหน้าจอที่ใช้ในการลบข้อมูลการเข้าถึง (Access) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.20



รูปที่ 5. 20 หน้าจอลบข้อมูลการเข้าถึง (Access)

หน้าจอแก้ไขข้อมูลการเข้าถึง (Access)

เป็นหน้าจอที่ใช้ในการแก้ไขข้อมูลการเข้าถึง (Access) ที่จะใช้กับโมดูลการพิสูจน์ตัวตน แสดงดังรูปที่ 5.21

รูปที่ 5. 21 หน้าจอแก้ไขข้อมูลการเข้าถึง (Access)

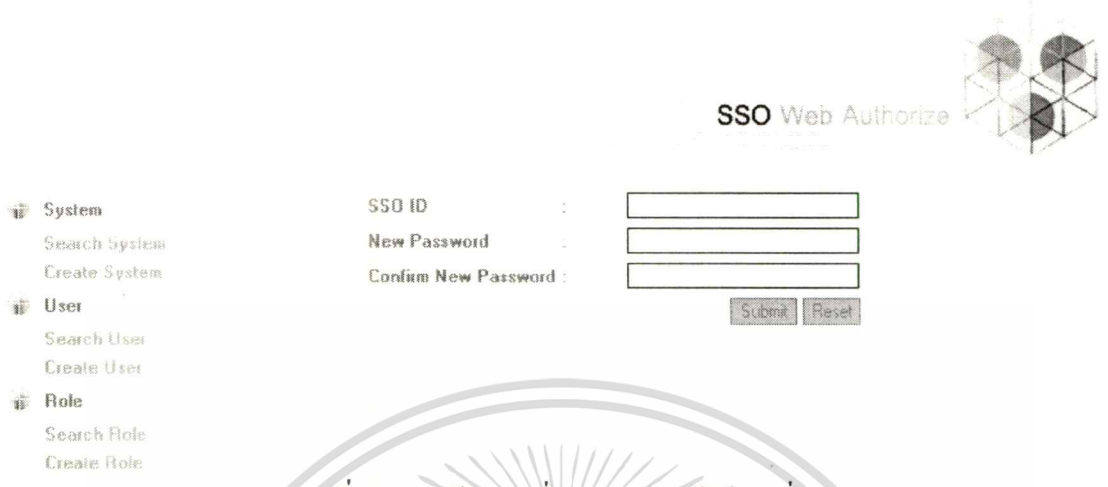
หน้าจอเปลี่ยนรหัสผ่านของรหัสผู้ใช้งานนี้ (Change Password This User)

เป็นหน้าจอที่ใช้ในการเปลี่ยนรหัสผ่านของผู้ใช้งานคนที่ทำการล็อกออนใช้งานระบบอยู่ ณ ขณะนี้ ดังรูปที่ 5.22

รูปที่ 5. 22 หน้าจอเปลี่ยนรหัสผ่านผู้ใช้งานนี้

หน้าจอเปลี่ยนรหัสผ่านของรหัสผู้ใช้งานอื่น(Change Password Other User)

เป็นหน้าจอที่ใช้ในการเปลี่ยนรหัสผ่านของผู้ใช้งานของบัญชีรายชื่ออื่นดังรูปที่ 5.23



รูปที่ 5. 23 หน้าจอเปลี่ยนรหัสผ่านผู้ใช้งานอื่น

บทที่ 6
สรุปผลการค้นคว้าและพัฒนาระบบ

6.1 ผลการพัฒนาระบบ

การพัฒนาระบบการพิสูจน์ตัวตนเพียงครั้งเดียว ได้วิเคราะห์และออกแบบในเชิงวัตถุโดยนำเสนอด้วย UML ในรูปแบบของไดอะแกรมต่างๆ ประกอบด้วยยูสเคส คลาสไดอะแกรม ซีเควนซ์ไดอะแกรม แอ็คทีวิตี้ไดอะแกรมและอีอาร์ไดอะแกรม ซึ่งทำให้การออกแบบพัฒนาง่ายยิ่งขึ้นระบบงานนี้ประกอบด้วย 2 ส่วน คือ เว็บแอพพลิเคชันและเว็บเซอร์วิสในการตรวจสอบ กำหนดสิทธิแก่ผู้ใช้งาน ผู้ใช้งานที่เป็นผู้ดูแลระบบสามารถเข้ามากำหนดสิทธิต่างๆ โดยแบ่งตามหน้าที่งานและสิทธิที่มีให้แก่ผู้ใช้งานระดับปฏิบัติการ เพิ่มความสะดวก รวดเร็วและง่ายต่อการจัดการเป็นการควบคุมสิทธิของผู้ใช้งานระดับปฏิบัติการเข้าไว้ด้วยกัน ลดความซ้ำซ้อนในการพัฒนาโมดูลการพิสูจน์ตัวตนและให้สิทธิขององค์กร

ระบบการพิสูจน์ตัวตนเพียงครั้งเดียว ได้พัฒนาขึ้นเพื่อรวมรวมการเข้าถึงทรัพยากรระบบของผู้ใช้งานระดับปฏิบัติการเข้าไว้ที่ระบบเดียว สามารถแก้ไขปัญหาการดำเนินงานในปัจจุบันดังนี้

6.1.1 ความซ้ำซ้อนของการพัฒนาระบบการพิสูจน์ตัวตนและการให้สิทธิการเข้าถึงข้อมูลของเว็บแอพพลิเคชันภายในองค์กร

6.1.2 การจัดการควบคุมการเข้าถึงข้อมูลรวมอยู่ที่เดียวเพิ่มความสะดวก และประสิทธิภาพในการทำงานของผู้ดูแลระบบ

6.1.3 ลดความซ้ำซ้อนในการเก็บข้อมูล

6.1.4 เพิ่มความอิสระในเทคโนโลยีของการพัฒนาระบบเว็บแอพพลิเคชันและแอพพลิเคชันขององค์กร

6.2 ผลจากการทดสอบโปรแกรม

ตารางที่ 6. 1 ผลการทดสอบการทำงานต่างๆ ของระบบ

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ	ผลที่ได้รับ
ทดสอบการตรวจสอบสิทธิและการให้สิทธิ	เว็บเซอร์วิสส่งข้อมูลสิทธิของรหัสผู้ใช้นั้นๆ กลับมาและเก็บข้อมูลการตรวจสอบตัวตนของรหัสผู้ใช้นั้นๆ เพื่อติดต่อกับเว็บเซิร์ฟเวอร์	ทำงานถูกต้อง

ตารางที่ 6.1 ผลการทดสอบการทำงานต่างๆ ของระบบ (ต่อ)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ	ผลที่ได้รับ
ทดสอบการเพิ่มข้อมูลระบบ	ผู้ดูแลระบบสามารถเพิ่มข้อมูลระบบได้อย่างเรียบร้อย	ทำงานถูกต้อง
ทดสอบการเพิ่มข้อมูลการเข้าถึง (Access)	ผู้ดูแลระบบสามารถเพิ่มข้อมูลการเข้าถึง (Access) ได้อย่างเรียบร้อย	ทำงานถูกต้อง
ทดสอบการเพิ่มข้อมูลสิทธิ (Object)	ผู้ดูแลระบบสามารถเพิ่มข้อมูลสิทธิได้อย่างเรียบร้อย	ทำงานถูกต้อง
ทดสอบการเพิ่มข้อมูลหน้าที่ (Role)	ผู้ดูแลระบบสามารถเพิ่มข้อมูลการเข้าถึง (Access) ได้อย่างเรียบร้อย	ทำงานถูกต้อง
ทดสอบการเพิ่มข้อมูลการเข้าถึง (Access)	ผู้ดูแลระบบสามารถเพิ่มข้อมูลการเข้าถึงได้อย่างเรียบร้อย	ทำงานถูกต้อง
ทดสอบการเพิ่มข้อมูลความสัมพันธ์ระหว่าง สิทธิ หน้าที่ และการเข้าถึง	ผู้ดูแลระบบสามารถเพิ่มข้อมูลความสัมพันธ์ระหว่าง สิทธิ หน้าที่ และการเข้าถึงได้อย่างเรียบร้อย	ทำงานถูกต้อง
ทดสอบการค้นหาระบบ	ผู้มีสิทธิในการค้นหาสามารถค้นหาข้อมูลระบบได้อย่างถูกต้อง	ทำงานถูกต้อง
ทดสอบการค้นหาผู้ใช้งาน	ผู้มีสิทธิในการค้นหาสามารถค้นหาข้อมูลการเข้าถึง (Access) ได้อย่างถูกต้อง	ทำงานถูกต้อง
ทดสอบการค้นหาสิทธิ (Object)	ผู้มีสิทธิในการค้นหาสามารถค้นหาข้อมูลสิทธิ ได้อย่างถูกต้อง	ทำงานถูกต้อง
ทดสอบการค้นหาหน้าที่ (Role)	ผู้มีสิทธิในการค้นหาสามารถค้นหาข้อมูลหน้าที่ได้อย่างถูกต้อง	ทำงานถูกต้อง
ทดสอบการค้นหาการเข้าถึง(Access)	ผู้มีสิทธิในการค้นหาสามารถค้นหาข้อมูลการเข้าถึงได้อย่างถูกต้อง	ทำงานถูกต้อง
ทดสอบการแก้ไขระบบ	ผู้มีสิทธิในการแก้ไขสามารถแก้ไขระบบได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการแก้ไขข้อมูลการเข้าถึง (Access)	ผู้มีสิทธิในการแก้ไขสามารถแก้ไขข้อมูลการเข้าถึง (Access) ได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการแก้ไขข้อมูลสิทธิ (Object)	ผู้มีสิทธิในการแก้ไขสามารถแก้ไขข้อมูลสิทธิ (Object)ได้เรียบร้อย	ทำงานถูกต้อง

ตารางที่ 6.1 ผลการทดสอบการทำงานต่างๆ ของระบบ (ต่อ)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ	ผลที่ได้รับ
ทดสอบการแก้ไขข้อมูลหน้าที่ (Role)	ผู้มีสิทธิในการแก้ไขสามารถแก้ไขข้อมูลหน้าที่ (Role)ได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการแก้ไขข้อมูลเข้าถึง(Access)	ผู้มีสิทธิในการแก้ไขสามารถแก้ไขข้อมูลเข้าถึง(Access)ได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการลบข้อมูลระบบ	ผู้มีสิทธิในการลบข้อมูลระบบได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการลบข้อมูลการเข้าถึง (Access)	ผู้มีสิทธิในการลบสามารถลบข้อมูลการเข้าถึง (Access) ได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการลบข้อมูลสิทธิ (Object)	ผู้มีสิทธิในการลบสามารถลบข้อมูลสิทธิ (Object)ได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการลบข้อมูลหน้าที่ (Role)	ผู้มีสิทธิในการลบสามารถลบข้อมูลหน้าที่ (Role)ได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการลบข้อมูลเข้าถึง(Access)	ผู้มีสิทธิในการลบสามารถลบข้อมูลเข้าถึง (Access)ได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการลบความสัมพันธ์ระหว่างสิทธิ หน้าที่ และการเข้าถึง	ผู้มีสิทธิในการลบสามารถลบความสัมพันธ์ระหว่างสิทธิ หน้าที่ และการเข้าถึงได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการเปลี่ยนรหัสผ่านของรหัสผู้ใช้งาน ณ ขณะนั้น	ผู้ใช้งานสามารถเปลี่ยนรหัสผ่านของรหัสผู้ใช้งาน ณ ขณะนั้นได้เรียบร้อย	ทำงานถูกต้อง
ทดสอบการเปลี่ยนรหัสผ่านของรหัสผู้ใช้งานอื่นอื่น	ผู้ดูแลระบบหรือผู้ที่มีสิทธิสามารถเปลี่ยนรหัสผ่านของรหัสผู้ใช้งานอื่นได้เรียบร้อย	ทำงานถูกต้อง

6.3 อุปสรรค์ในการพัฒนาโปรแกรม

6.3.1 การพัฒนามีการนำโปรแกรมหลายชนิดมาใช้งานร่วมกันจึงเกิดความยุ่งยากในการติดตั้งให้โปรแกรมต่างๆ สามารถทำงานร่วมกันได้อย่างดี

6.3.2 ในการพัฒนาระบบมีการนำเอา LDAP เข้ามาใช้งานบนระบบปฏิบัติการ Windows ดังนั้นจึงต้องใช้ LDAP เซิร์ฟเวอร์ ที่สามารถทำงานได้บนระบบปฏิบัติการ Windows โดยเฉพาะ

6.3.3 เนื่องจากระบบ Directory เป็นเทคโนโลยีที่ยังไม่แพร่หลาย ดังนั้นเมื่อเจอปัญหาระหว่างการพัฒนาจึงหาข้อมูลหรือข้อแนะนำในการแก้ไขปัญหาค่อนข้างยาก

6.4 ข้อเสนอแนะ

6.4.1 ในการนำเอาระบบพิสูจน์ตัวตนเพียงครั้งเดียวมาพัฒนาต่อ

6.4.1.1 น่าจะมีการเพิ่มในส่วนของความปลอดภัยมากขึ้น เช่น การส่งผ่านข้อมูลจาก http ไปเป็น https ,การสร้างข้อมูลผู้ใช้งานควรมีการสร้างเป็นรหัสผ่านอัตโนมัติโดยผู้สร้างข้อมูลผู้ใช้งานไม่ต้องกำหนดให้

6.4.1.2 การส่งข้อมูลรายละเอียดของผู้ใช้งานที่สร้างขึ้น เช่น ข้อมูล password อาจจะทำเป็นการส่งข้อมูลดังกล่าวอัตโนมัติไปที่ e-mail ของผู้ใช้งาน

6.4.1.3 ในการนำเข้าข้อมูลผู้ใช้งานและสิทธิของผู้ใช้งานในปริมาณเยอะๆ เช่น ในการนำเข้าข้อมูลครั้งแรกเผื่อเป็นข้อมูลตั้งต้นของระบบ น่าจะมีการทำ script ของการโหลดข้อมูลหรือการ import ข้อมูล

6.4.2 ในการติดตั้งโปรแกรมหรือทำการแก้ไขค่า Configuration ต่างๆ ของโปรแกรมควรทำการสำรองข้อมูลเอาไว้ก่อนเพื่อต้องกันความผิดพลาดที่อาจจะเกิดขึ้น

6.4.3 ในกรณีที่ต้องการเปลี่ยนแปลงค่า Configuration ของ Directory ควรทำการแก้ไขในช่วงที่ไม่มีการใช้งาน

6.4.4 ในการติดตั้งโปรแกรมที่ใช้ในการทำงานต่างๆ ควรทำการเก็บค่าพารามิเตอร์ในการติดตั้งทุกครั้ง เพื่อใช้ในการกำหนดค่าการติดตั้งครั้งต่อไปถ้าจำเป็น

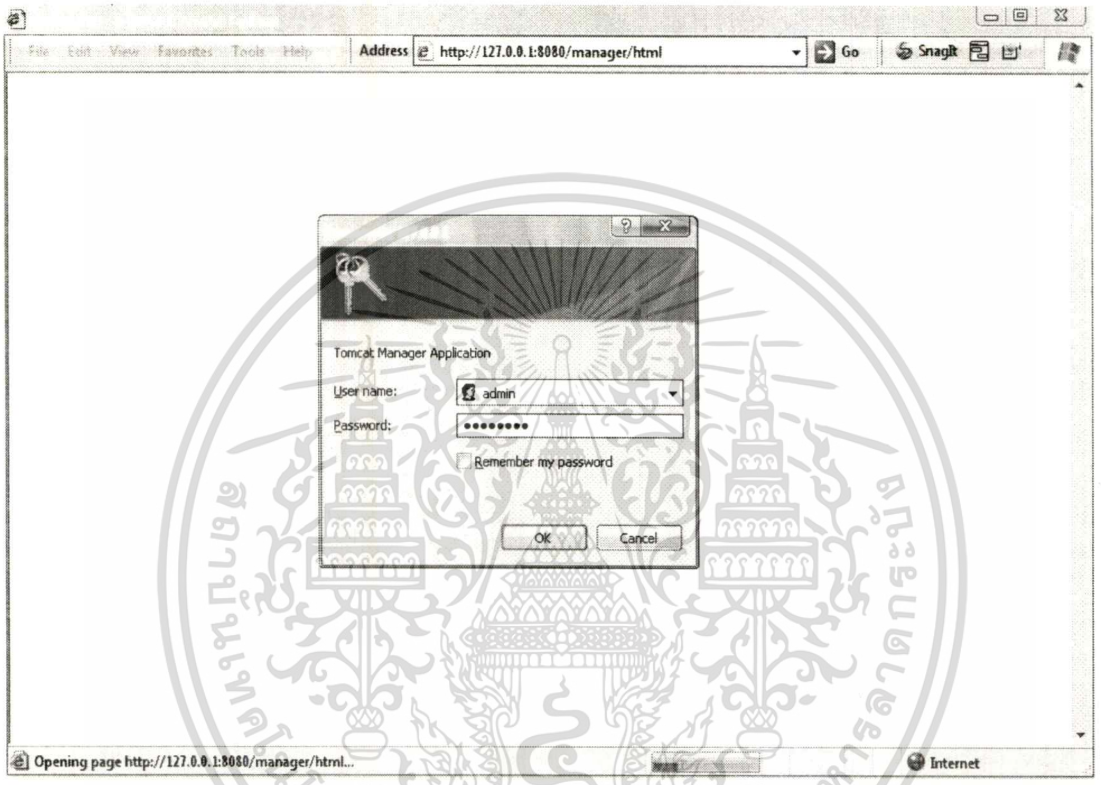
บรรณานุกรม

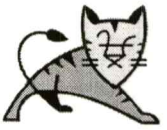
Howes, T. and Smith, M. 1997. **LDAP Programming Directory-Enabled Applications with Lightweight Directory Access Protocal**. Indianapolis. Macmillan Technical.

Timothy, A. et al. 1999. **Understanding and Deploying LDAP Directory Services**. USA.Macmillan Computer.

ภาคผนวก ก.
การติดตั้งระบบพิสูจน์ตัวตนเพียงครั้งเดียว

1. เข้าหน้าจอ Tomcat Manager





Tomcat Web Application Manager

Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
/SSOWeb		false	0	Start Stop Reload Undeploy
/host-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
/manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
/tomcat-docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy

Deploy

Deploy directory or WAR file located on server

Context Path (optional):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

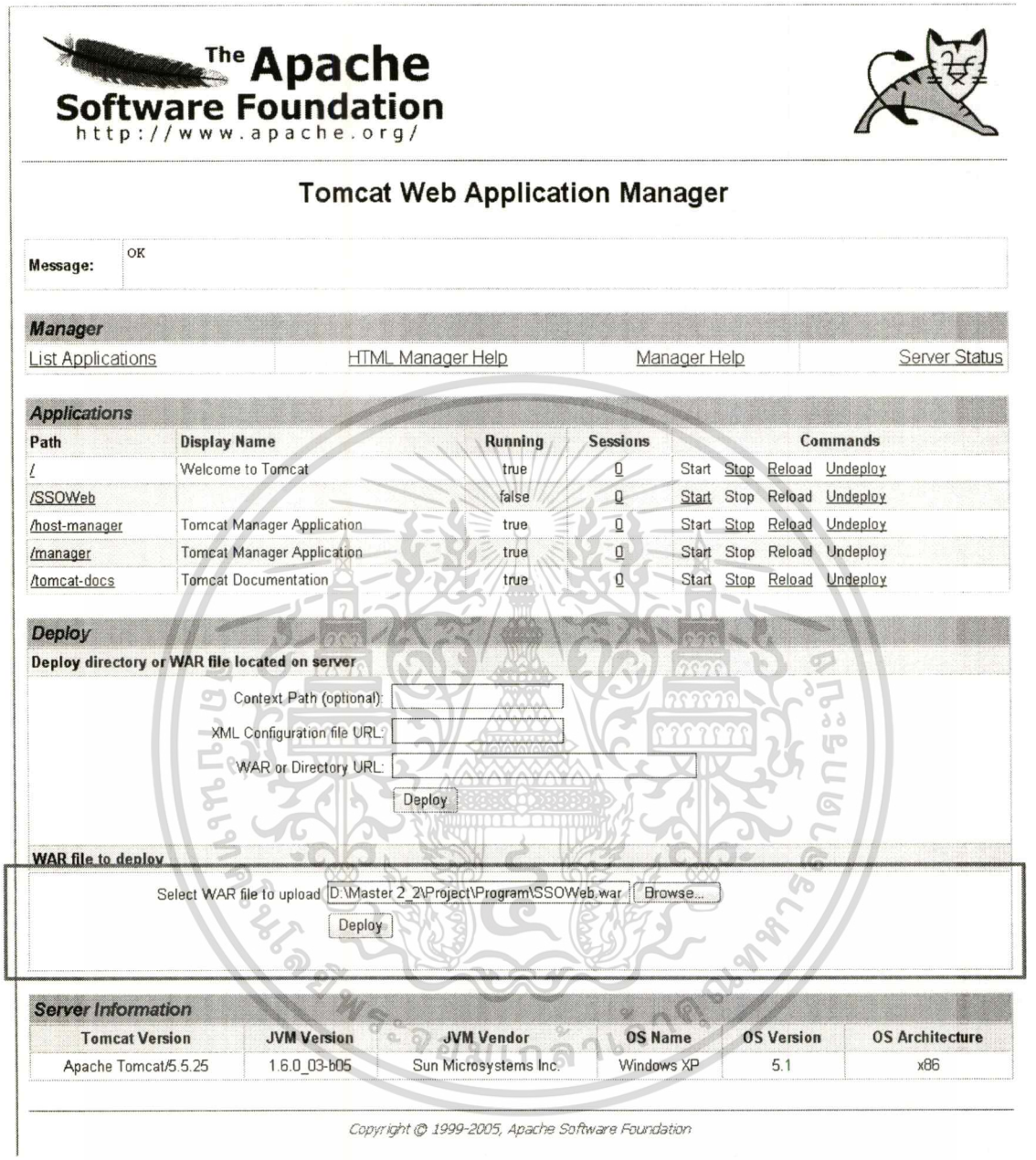
Select WAR file to upload Browse...

Deploy

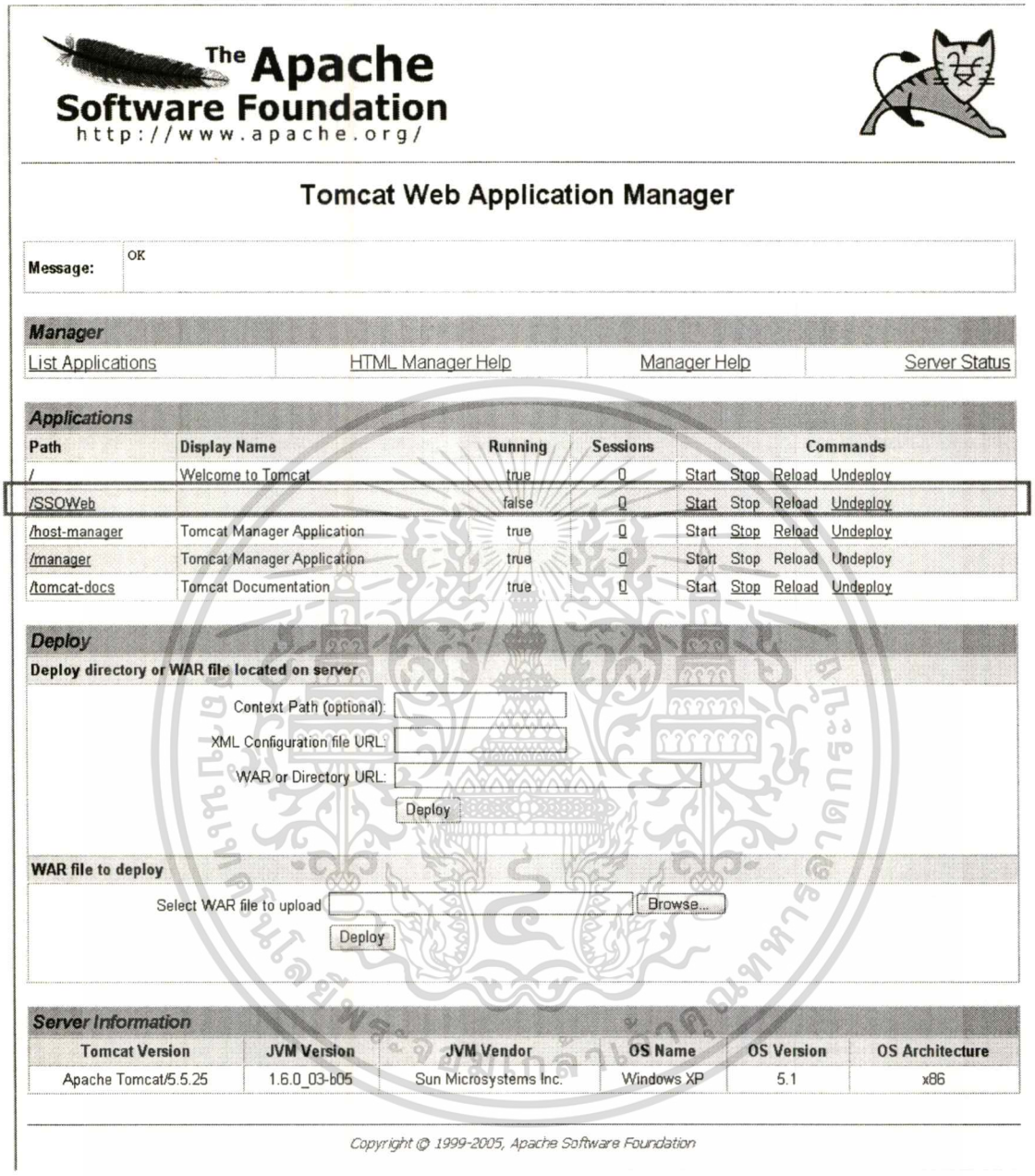
Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture
Apache Tomcat/5.5.25	1.6.0_03-b05	Sun Microsystems Inc.	Windows XP	5.1	x86

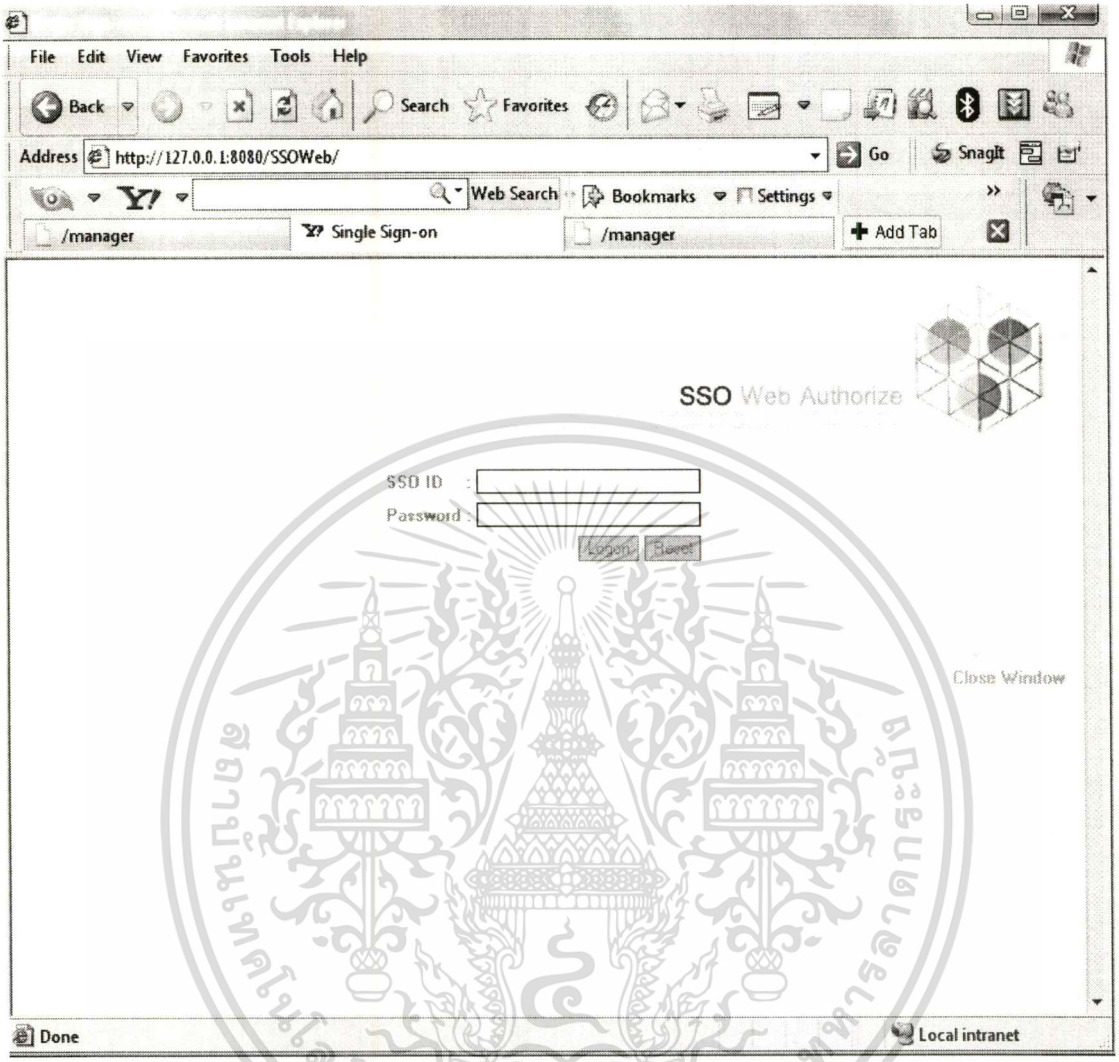
2. เลือก SSOWeb.war



3. ทำการ Start SSOWeb Application



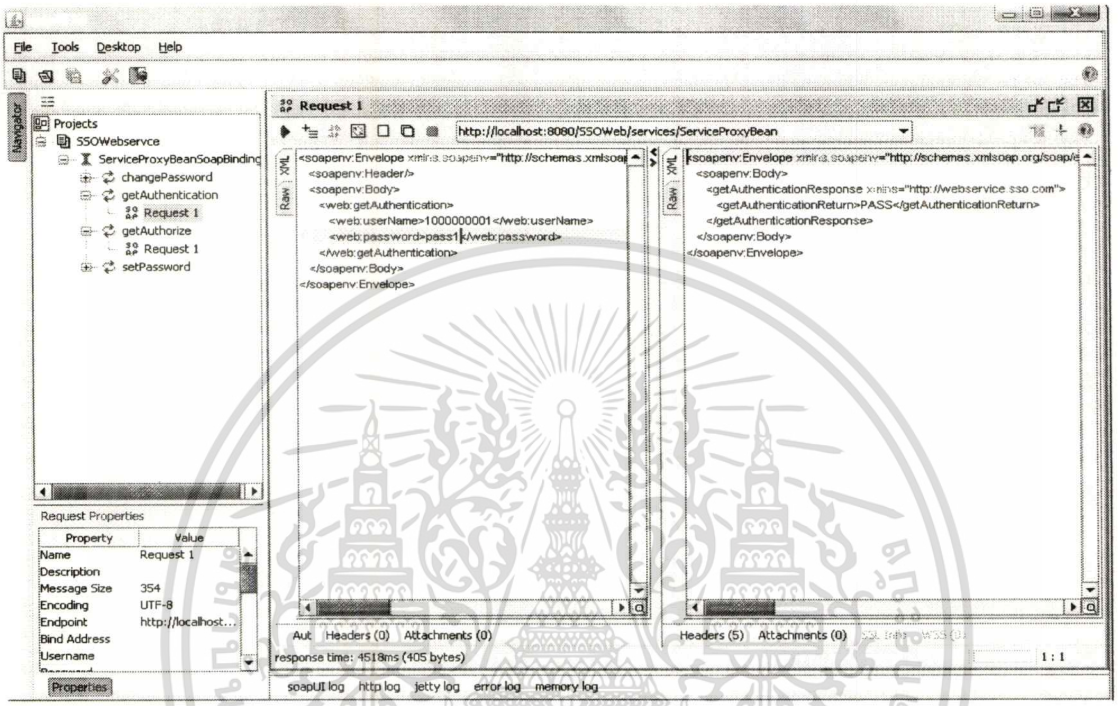
4. เข้าหน้า logon ของ SSOWeb



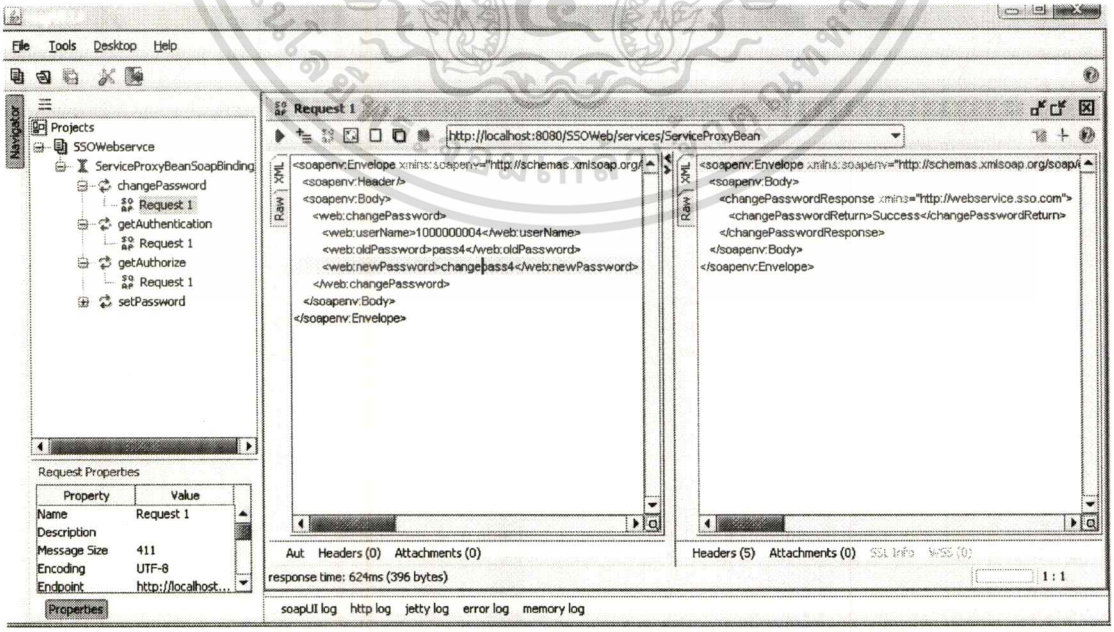
ภาคผนวก ข.

การทดสอบเว็บเซอร์วิส

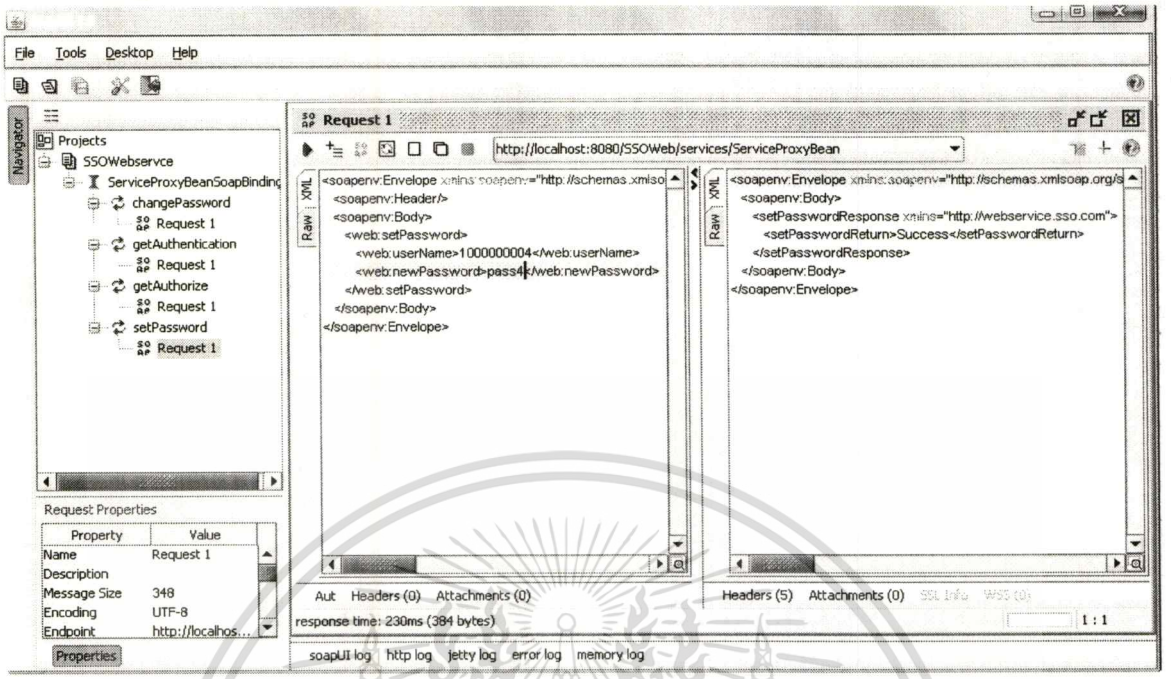
- โมดูลตรวจสอบตัวตน



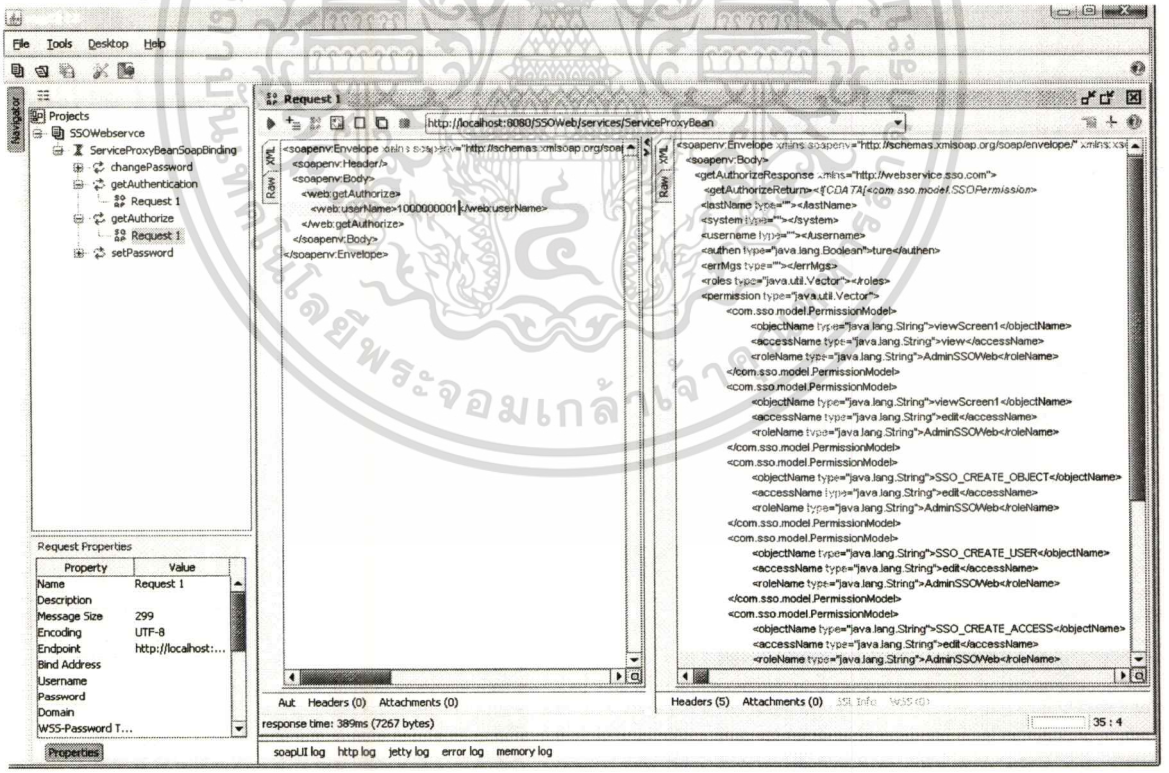
- โมดูลการเปลี่ยนรหัสผ่าน



- โมดูลการกำหนดรหัสผ่านใหม่

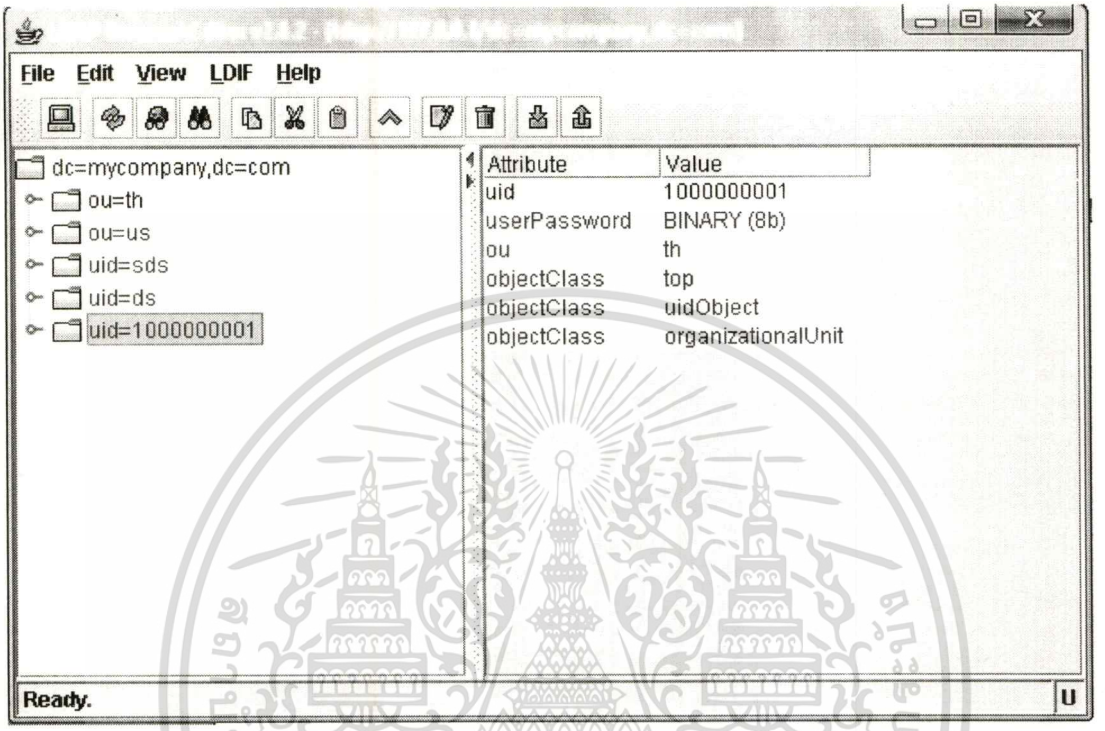


- โมดูลการให้สิทธิและหน้าที่แก่ผู้ใช้งานระบบ



ภาคผนวก ค.

โครงสร้างข้อมูลที่เก็บใน LDAP



ประวัติผู้เขียน

ชื่อผู้เขียน	วิมลรัตน์ โชติไพศาลกุล
วัน เดือน ปีเกิด	7 กรกฎาคม 2525
ที่อยู่	53/301 หมู่บ้านพูนศิริ 2 ซอยโชคชัย 4 ลาดพร้าว กรุงเทพมหานคร 10230
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
ปีการศึกษาที่สำเร็จการศึกษา	2547
ความชำนาญเฉพาะด้าน	1. ระบบสารสนเทศ e-Banking และ non-bank 2. การออกแบบและการพัฒนาระบบสารสนเทศด้วยภาษาเชิงอ็อบเจค 3. เขียน PL/SQL procedure