การทำซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ

SINGLE SIGN-ON IMPLEMENTATION FOR WEB - BASED APPLICATION

โดย

กำพล ภักดีลิขิต

GUMPOL PHUKDEELIKIT

อาจารย์ที่ปรึกษา

รศ.ดร. โชติพัชร์ ภรณวลัย

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2550

SINGLE SIGN-ON IMPLEMENTATION
FOR WEB - BASED APPLICATION

GUMPOL PHUKDEELIKIT

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/ 2007

หัวข้อ	การทำชิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ
นักศึกษา	นายกำพล ภักดีลิขิต
รหัสนักศึกษา	49066526
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	รศ.ดร.โชติพัชร์ ภรณวลัย

บทคัดย่อ

โครงงานพัฒนาระบบงานฉบับนี้เสนอวิธีการในการจัดการเพื่อให้สามารถเข้าถึงระบบหรือโปรแกรมประยุกต์ต่างๆ บนเว็บ ได้ด้วยการพิสูจน์ตัวจริงเพียงครั้งเดียว โดยลักษณะเด่นของวิธีการที่นำเสนอในโครงงานนี้คือ การใช้ศูนย์กลางในการพิสูจน์ตัวจริง ซึ่งระบบศูนย์กลางดังกล่าว จะเก็บหลักฐาน (Identity) ของผู้ใช้งานระบบต่างๆเอาไว้เป็นส่วนกลาง และเปิดบริการให้ระบบต่างๆใช้ในการตรวจพิสูจน์ตัวจริงผู้เข้าใช้งาน ซึ่งระบบศูนย์กลางจะทำหน้าที่พิสูจน์ตัวจริงและยืนยันกลับไปยังระบบต้นทางโดยมีการลงลายมือชื่อดิจิตอล (Digital Signature) แนบไปด้วย ในโครงงานเล่มนี้จะใช้อัลกอริธึม SHA1 และ RSA ในการสร้างเป็นลายมือชื่อดิจิตอล (Digital Signature) ซึ่งจะใช้ Lotus Domino Server เพื่อพัฒนาเป็น Centralized Authentication Server

Title	Single sign-on Implementation for Web - Based Application
Student	Mr. Gumpol Phukdeelikit
Student ID.	49066526
Degree	Master of Science
Programme	Information Science
Academic Year	2007
Advisor	Assoc.Prof. Dr.Chotipat Pornavalai

ABSTRACT

This project proposes a Single sign-on (SSO), specialized form of software authentication management that enable user to authenticate once and gain access to the resources of multiple software systems. In this project, the dominant features of this system are centralized system which collects the identities of all system and centralize authentication. Then send the digital signature with RSA and SHA algorithm to certify user who has been authenticated. The design of Single sign-on architecture for web application, we are using IBM Lotus Domino Server to run as a Centralized Authentication Server.

กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จได้ด้วยดี ด้วยคำแนะนำและคำปรึกษาจาก รศ.ดร.โชติพัชร์ ภรณวลัย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานนี้ ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากท่านอาจารย์และขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทั้งที่เคยเรียนด้วยกันในระดับต่างๆ และในระดับปริญญาโท IS21.2 สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง แห่งนี้ด้วย ทุกคนที่ให้คำแนะนำและคอยให้กำลังใจเป็นอย่างดีเสมอมา

ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจเสมอมา และเพื่อนๆ พี่ๆ ที่บริษัท M.B System Automation จำกัด ที่ให้ความรู้ในด้านต่างๆ ทำให้ข้าพเจ้าสามารถทำโครงงานฉบับนี้สำเร็จลุล่วงไปด้วยดี

กำพล ภักดีลิขิต

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการพัฒนาระบบหรือโปรแกรมประยุกต์ต่าง ๆ บนเว็บ เป็นไปอย่างแพร่หลายมากขึ้น ซึ่งในองค์กรต่างๆ แต่ละองค์กรส่วนใหญ่แล้วมีการพัฒนาโปรแกรมประยุกต์บนเว็บมากกว่าหนึ่งระบบ ซึ่งอาจทำงานอยู่บนแพลตฟอร์มที่แตกต่างกัน และพัฒนาด้วยภาษาที่แตกต่างกันอีกด้วย ซึ่งระบบดังกล่าวมีการเปิดให้เข้าถึงข้อมูลที่สำคัญต่างๆ ขององค์กร จึงจำเป็นจะต้องมีการปกป้องแหล่งข้อมูลดังกล่าวให้ปลอดภัยและให้สามารถเข้าถึงได้ตามผู้ที่มีสิทธิ์ในแหล่งข้อมูลนั้นๆ ซึ่งการที่จะทำให้ข้อมูลปลอดภัยนั้น จึงต้องมีการตรวจพิสูจน์ตัวจริง ก่อนการเข้าใช้งานระบบหรือเข้าถึงข้อมูล โดยวิธีการส่วนใหญ่แล้วก็คือการตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน ว่ามีตัวตนจริง และสามารถเข้าใช้งานระบบได้หรือไม่ ซึ่งแต่ละระบบจะมีการจัดการดูแลในเรื่องของบัญชีผู้ใช้งาน (Account) และการตรวจพิสูจน์ตัวจริง (Authentication) เหล่านี้ด้วยกลไกของแต่ละระบบเอง

ดังนั้นหากในองค์กรมีการใช้งานระบบต่างๆ หลากหลายระบบ ที่มีความแตกต่างกันของแพลตฟอร์มและภาษาที่ใช้ในการพัฒนาแล้ว ปัญหาที่เกิดขึ้นคือ ผู้ดูแลระบบ (Admin) ที่ทำหน้าที่บริหารจัดการในเรื่องของการจัดการบัญชีผู้ใช้งานและการตรวจสอบพิสูจน์ตัวจริง ก็อาจเกิดความสับสนและยุ่งยากขึ้นได้ อีกทั้งในส่วนของบัญชีผู้ใช้ก็อาจเกิดความซ้ำซ้อนขึ้น เนื่องจากผู้ใช้งานหนึ่งคน (User) สามารถใช้งานได้หลายๆระบบ ก็อาจทำให้ผู้ใช้งานเกิดความสับสนขึ้นได้ในการที่จะเลือกบัญชีผู้ใช้งานให้ถูกต้องเพื่อเข้าสู่ระบบ รวมถึงผู้พัฒนาระบบ (Developer) ก็จะต้องพัฒนาขั้นตอนหรือส่วนของโปรแกรมที่ใช้ในการตรวจสอบพิสูจน์ตัวจริง แตกต่างกันไปตามแพลตฟอร์มที่ระบบนั้นๆ ทำงาน และหากมีการใช้งานระบบร่วมกันหรือผู้ใช้งานเข้าใช้งานระบบต่างๆพร้อมกัน ก็จะต้องทำการพิสูจน์ตัวจริงทุกครั้งที่มีการเข้าใช้งานหรือเข้าถึงข้อมูลของอีกระบบ ทำให้เกิดความไม่สะดวกในการทำงาน การนำเทคนิคการจัดการเพื่อให้ผู้ใช้งานเข้าใช้งานระบบต่างๆ ด้วยการตรวจพิสูจน์ตัวจริงเพียงครั้งเดียว (Single Sign-on) มาใช้ จึงเป็นรูปแบบหนึ่งที่จะช่วยแก้ปัญหาดังกล่าวข้างต้นได้ เนื่องจากไม่ต้องมีการตรวจพิสูจน์ตัวจริงใหม่ทุกครั้งที่เข้าใช้งานอีกระบบ ทำให้ผู้ใช้งานจดจำแค่บัญชีผู้ใช้งานเดียว ทำให้ผู้ดูแลระบบสามารถตรวจสอบบริหารจัดการบัญชีผู้ใช้งานได้ง่ายขึ้น และผู้พัฒนาระบบสามารถพัฒนาโปรแกรมในส่วนของการพิสูจน์ตัวจริงไปในทิศทางเดียวกันได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

โครงงานฉบับนี้มุ่งหวังเพื่อศึกษาและพัฒนาเทคนิค การทำซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ (Single sign-on For Web-Based Application) เพื่อให้ระบบต่างๆ สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพสูงสุด ทำให้การดูแลบริหารจัดการบัญชีผู้ใช้งานและการตรวจพิสูจน์ตัวจริงก่อนเข้าใช้งานระบบมีความสะดวก และไม่ก่อให้เกิดความสับสนได้ต่อผู้ดูแลระบบ ผู้ใช้งาน อีกทั้งผู้พัฒนาระบบด้วย ทำให้ลดความซ้ำซ้อนในการจัดการและจัดเก็บบัญชีผู้ใช้งาน และเป็นการลดการส่งข้อมูลชื่อผู้ใช้งานและรหัสผ่านเข้าสู่เครือข่ายอินเตอร์เน็ตด้วย ดังนั้นในโครงงานฉบับนี้จึงเสนอวิธีการพัฒนาระบบจัดการกลางเพื่อดูแลบัญชีผู้ใช้งาน และการพิสูจน์ตัวจริงผู้ใช้งาน (Centralized Authentication System) ซึ่งจะทำให้การทำงานร่วมกันของระบบต่างๆ ภายในองค์กรดีขึ้น และมีประสิทธิภาพมากยิ่งขึ้น

1.3 สมมติฐานของการศึกษา

ข้อด้อยในการกระจายหน้าที่ การพิสูจน์ตัวจริงของผู้ใช้งานไปยังโปรแกรมประยุกต์บนเว็บต่างๆ คือ อาจเกิดความซ้ำซ้อนในบัญชีชื่อผู้ใช้งานได้ อาจเกิดความไม่สะดวกและก่อให้เกิดความสับสนต่อผู้ดูแลระบบ ผู้ใช้งานระบบได้ ทำให้การแสดงผลข้อมูลที่ถูกปกป้องจากหลายระบบพร้อมกัน เป็นไปได้ยากเนื่องจากผู้ใช้งานอาจต้องมีการพิสูจน์ตัวจริงอีกครั้ง ทำให้เป็นข้อจำกัดของผู้พัฒนาระบบที่จะต้องเลือก แพลตฟอร์มหรือภาษาเดียวกัน เพื่อใช้ในการพัฒนาระบบเพื่อให้สามารถทำงานร่วมกันได้ง่าย

ดังนั้นเพื่อการแก้ไขปัญหาข้างต้นนี้ จึงได้เสนอวิธีการใช้ระบบจัดการกลางเพื่อดูแลบัญชีผู้ใช้งาน และการพิสูจน์ตัวจริงผู้ใช้งาน (Centralized Authentication System) คือระบบกลางที่ทำหน้าที่จัดเก็บและตรวจสอบพิสูจน์ตัวจริง เมื่อมีการร้องขอจากผู้ใช้งานผ่านทางระบบที่ผู้ใช้งานต้องการที่จะเข้าถึง เพื่อลดความซ้ำซ้อนของบัญชีผู้ใช้ และทำให้การทำงานร่วมกันของระบบต่างๆภายในองค์กรมีประสิทธิภาพเพิ่มมากขึ้น

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการพัฒนา

การทำงานของระบบการจัดเก็บบัญชีรายชื่อผู้ใช้งานและการตรวจพิสูจน์ตัวจริง จะทำที่ศูนย์กลาง โดยในส่วนของบริการด้านการจัดการบัญชีรายชื่อผู้ใช้งาน (Directory Service) จะเปิดช่องทางให้จัดการโดยใช้ Lightweight Directory Access Protocol (LDAP) หรือใช้ Web Service เพื่อให้โปรแกรมประยุกต์บนเว็บที่เข้าร่วมสามารถเพิ่ม ลบ หรือเปลี่ยนแปลงข้อมูลของบัญชีรายชื่อที่ระบบนั้นๆ ดูแลอยู่ได้ ในส่วนของการตรวจพิสูจน์ตัวจริง (Authentication) นั้น จะใช้กลไกการทำงานของระบบศูนย์กลางนั้นๆ เป็นตัวจัดการ ซึ่งในที่นี้จะใช้ Lotus Domino Server

1.5 ขอบเขตการพัฒนา

ในโครงงานฉบับนี้ได้นำเสนอวิธีการในการออกแบบและพัฒนาการทำซิงเกิ้ลไซออน สำหรับโปรแกรมประยุกต์บนเว็บ โดยจัดทำในแบบการให้บริการเป็นศูนย์กลางในด้านการจัดการบัญชีรายชื่อผู้ใช้งาน (Directory Service) และการตรวจพิสูจน์ตัวจริง (Authentication) เพื่อให้ระบบต่างๆ สามารถเรียกใช้บริการได้ ซึ่งมีการยืนยันความถูกต้องของการพิสูจน์ตัวจริงส่งผ่านไปยังระบบต่างๆ โดยใช้ลายมือชื่อดิจิตอล (Digital Signature) ในโครงงานนี้จะใช้อัลกอริธึม SHA-1 และ RSA ในการสร้างลายมือชื่อดิจิตอล โดยใช้โครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure) ในการเข้ารหัส

1.6 ขั้นตอนของการศึกษา

โครงงานฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหาในโครงงาน ความมุ่งหมายและวัตถุประสงค์ สมมติฐานของการศึกษา ทฤษฎีที่ใช้ ขอบเขตของการพัฒนา และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่ใช้ในการพัฒนาระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ

บทที่ 3 กล่าวถึงการวิเคราะห์และออกแบบระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ

บทที่ 4 กล่าวถึงโครงสร้างและขั้นตอนการทำงานของระบบ

บทที่ 5 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในการพัฒนา ระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการออกแบบและพัฒนา ระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ ซึ่งเนื้อหาในบทนี้จะกล่าวถึงเทคนิคการตรวจพิสูจน์ตัวจริงแบบต่างๆ การบริการและการจัดการบัญชีรายชื่อผู้ใช้งาน การศึกษาเกี่ยวกับรหัสสัญญาณ (Cryptography) แบบต่างๆ ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษา เพื่อออกแบบและพัฒนาระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ และเพื่อการรักษาความปลอดภัยของข้อมูลในการทำงานร่วมกันของระบบต่างๆ

2.1 เทคนิคการตรวจพิสูจน์ตัวจริง (Authentication)

ในการเข้าถึงแหล่งข้อมูลที่มีความสำคัญหรือการเข้าใช้ระบบนั้น มีความจำเป็นที่จะต้องทำการตรวจสอบและพิสูจน์ตัวตนที่แท้จริงของผู้ที่จะเข้าถึงหรือเข้าใช้งาน เพื่อให้แน่ใจได้ว่าเป็นตัวจริงและมีสิทธิ์ในการเข้าถึงข้อมูลหรือเข้าใช้ระบบนั้นๆ โดยสามารถแบ่งได้ออกเป็น 2 ส่วน คือ การตรวจพิสูจน์ตัวจริง (Authentication) และการอนุญาตให้เข้าถึงตามสิทธิ์ของผู้ใช้งาน (Authorization) ซึ่งโดยส่วนใหญ่แล้วทั้ง 2 ส่วนนี้จะทำงานควบคู่กันไปโดยที่ระบบจะตรวจพิสูจน์ตัวจริงก่อน หากถูกต้องจึงจะตรวจสอบว่าบัญชีผู้ใช้งานนั้นมีสิทธิ์ในระบบอย่างไร สามารถเข้าถึงข้อมูลได้มากน้อยเพียงใด จึงอนุญาตให้เข้าถึงตามสิทธิ์ของผู้ใช้งานนั้นๆ

ในระบบของการทำซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ ที่จะนำเสนอในโครงงานนี้ จะจัดการในส่วนของการตรวจพิสูจน์ตัวจริงเท่านั้น ในส่วนของการอนุญาตให้เข้าถึงตามสิทธิ์ของผู้ใช้งาน จะเป็นหน้าที่ของระบบต่างๆ ที่จะตรวจสอบเอง

2.1.1 รูปแบบการตรวจพิสูจน์ตัวจริง

การตรวจพิสูจน์ตัวจริงในระบบนี้จะเป็นลักษณะของ Web Server Authentication โดยรูปแบบที่ Web Server ทั่วไปส่วนใหญ่สนับสนุนการทำงานคือ Basic HTTP Authentication นอกจากนี้บาง Web Server ยังสนับสนุนการตรวจพิสูจน์ตัวจริงในรูปแบบ Digest HTTP Authentication ซึ่งการนำรูปแบบใดมาใช้นั้น ก็ขึ้นอยู่กับ Web Server ที่เลือกใช้

โดย Basic HTTP Authentication จะเป็นรูปแบบที่เก่าแก่และแพร่หลายมากที่สุดเนื่องจาก Browser ส่วนใหญ่สนับสนุนการทำงานแบบ Basic HTTP Authentication โดยวิธีการคือ Browser จะส่งข้อมูลสำคัญ 2 ส่วนไปยัง Web Server คือ ชื่อผู้ใช้งาน (User Name) และ รหัสผ่าน (Password) โดยจะส่งไปในรูปแบบของข้อความปกติ (Plaintext) ซึ่งหากมีการดักจับข้อมูลก็จะสามารถอ่านข้อมูลดังกล่าวออกได้

ส่วน Digest HTTP Authentication นั้นจะมีการทำ Digest ของข้อมูลชื่อผู้ใช้ และรหัสผ่าน ซึ่งจะช่วยให้การส่งข้อมูลในระหว่าง Authenticate นั้น ไม่ได้อยู่ในรูปแบบ Plain Text

ซึ่งกลไกการพิสูจน์ตัวจริงของโปรแกรมประยุกต์บนเว็บนั้น จะมีการส่งข้อมูลสำคัญดังกล่าวคือ ชื่อผู้ใช้งานและรหัสผ่าน ออกไปยัง Web Server ทุกครั้งที่มีการร้องขอ เนื่องจากเป็นการทำงานตามลักษณะของโปรโตคอล HTTP คือไม่ได้เป็นการเชื่อมต่อกับ Web Server แบบถาวร ดังนั้นหากมีการใช้งานแบบ Basic HTTP Authentication จึงควรมีการเพิ่มความปลอดภัยในการส่งข้อมูลให้มากขึ้น โดย Secure Socket Layer (SSL) จึงถูกนำมาใช้เพื่อเพิ่มความปลอดภัยในการรับส่งข้อมูลในขั้นตอน Authentication นี้ด้วย

2.1.2 ขั้นตอนการตรวจพิสูจน์ตัวจริง

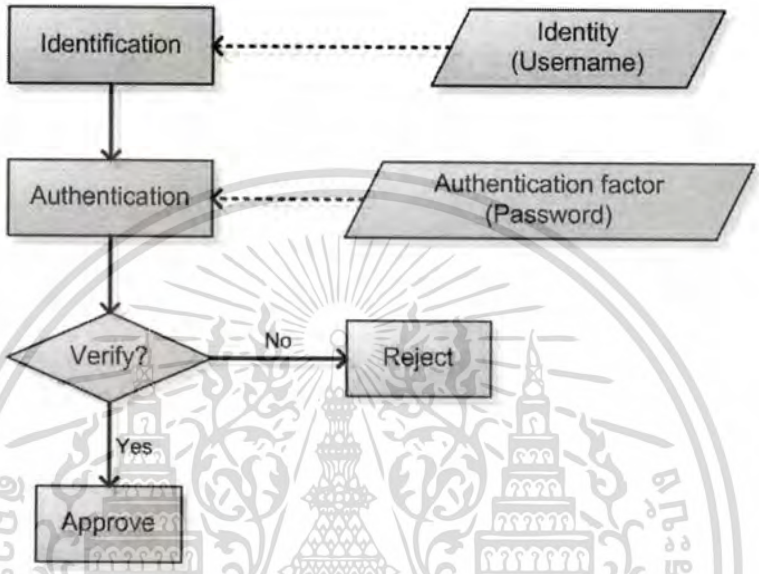
ขั้นตอนการทำงานของ Web Server Authentication โดยทั่วไปเป็นดังนี้

1. เมื่อผู้ใช้งานมีการร้องขอการเข้าถึงแหล่งข้อมูลที่ถูกปกป้อง Web Server จะแสดงหน้าจอเพื่อลงบันทึกเข้าใช้งาน (Login)
2. ผู้ใช้งานกรอกข้อมูลแล้วยืนยันการส่งข้อมูลมายัง Web Server
3. Web Server ตรวจสอบข้อมูล ที่ถูกส่งมาด้วยกลไกการทำงานของแต่ละ Web Server
4. ถ้าหากการลงบันทึกเข้าใช้งาน (Login) สำเร็จ Browser จะเก็บข้อมูลในการพิสูจน์ตัวจริงไว้ชั่วคราว และมีการส่งข้อมูลดังกล่าวไป เมื่อมีการร้องขอการเข้าใช้งานหลังจากนั้น
5. Web Server จะดำเนินการกับการร้องขอเหล่านั้นจาก Browser ถ้าหากมีสิทธิ์ถูกต้องก็จะส่ง Page ที่ได้มีการร้องขอนั้นกลับไป

ในทางปฏิบัติจะแบ่งออกเป็น 2 ขั้นตอน คือ

การระบุตัวตน (Identification) คือขั้นตอนที่ผู้ใช้แสดงหลักฐานว่าตนเองคือใครเช่น ชื่อผู้ใช้ (username)

การพิสูจน์ตัวตน (Authentication) คือขั้นตอนที่ตรวจสอบหลักฐานเพื่อแสดงว่าเป็นบุคคลที่กล่าวอ้างจริง



รูปที่ 2.1 ขั้นตอนการพิสูจน์ตัวจริง

การกำหนดนโยบายการรักษาความปลอดภัยในการพิสูจน์ตัวจริงก็มีความสำคัญด้วยเช่นกัน โดยส่วนใหญ่ เช่น

- การระงับบัญชีชื่อผู้ใช้งานชั่วคราวหากมีการพยายามลงชื่อเข้าใช้งานแต่ไม่สำเร็จภายใน 3 ครั้งติดต่อกัน
- การกำหนดความยาวขั้นต่ำของรหัสผ่านให้มีความยาว 8 ตัวอักษร
- การกำหนดให้มีการเปลี่ยนรหัสผ่านทุกๆ 60 วัน เป็นต้น

2.2 การศึกษาเกี่ยวกับวิทยาการการเข้ารหัสลับ (Cryptography)

ในเรื่องการรับส่งข้อมูลผ่านทางเครือข่ายอินเทอร์เน็ตนั้นจำเป็นจะต้องคำนึงถึงความปลอดภัยของข้อมูลที่มีการแลกเปลี่ยนกัน ดังนั้นเพื่อเป็นการหลีกเลี่ยงไม่ให้ผู้ที่ไม่ได้รับอนุญาตล่วงรู้ข้อมูล จึงต้องมีการเข้ารหัสเพื่อป้องกันข้อมูลเหล่านี้ นอกเหนือจากการป้องกันไม่ให้ล่วงรู้แล้ว ยังต้องมีการป้องกันไม่ให้ผู้ไม่มีสิทธิ์แก้ไขหรือปลอมแปลงข้อมูลให้ผิดเพี้ยนไปจากเดิม ซึ่งในเรื่องของซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บนี้ จะนำวิธีการนี้เพื่อมาช่วยในการลงลายมือชื่อดิจิตอลเพื่อยืนยันตัวผู้ใช้งานกับระบบต่างๆ หลังจากที่ได้มีการพิสูจน์ตัวจริงไปแล้ว

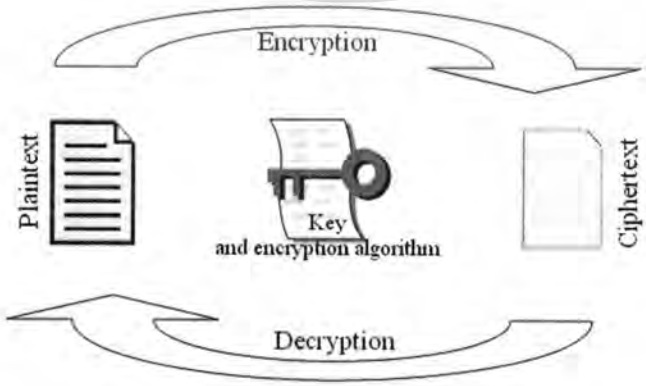
2.2.1 ขั้นตอนวิธีในการเข้ารหัส (Encryption Algorithm)

การเข้ารหัสข้อมูลโดยพื้นฐานแล้วจะเกี่ยวข้องกับวิธีการทางคณิตศาสตร์เพื่อ ใช้ในการป้องกันข้อมูลหรือข้อความตั้งต้นที่ต้องการส่งไปถึงผู้รับ ข้อมูลตั้งต้นจะถูกแปรเปลี่ยนไปสู่ข้อมูลหรือข้อความอีกรูปแบบหนึ่งที่ไม่สามารถอ่านเข้าใจได้โดยใครก็ตาม ที่ไม่มีกุญแจสำหรับเปิดดูข้อมูลนั้น เราเรียกกระบวนการในการแปรรูปของข้อมูลตั้งต้นว่า การเข้ารหัสข้อมูล (Encryption) และกระบวนการในการแปลงข้อความที่ไม่สามารถอ่าน และทำความเข้าใจให้กลับไปสู่ข้อความดั้งเดิมว่า การถอดรหัสข้อมูล (Decryption)

อัลกอริธึมในการเข้ารหัสข้อมูลมี 2 ประเภทหลัก คือ

1. อัลกอริธึมแบบสมมาตร (Symmetric key algorithms)

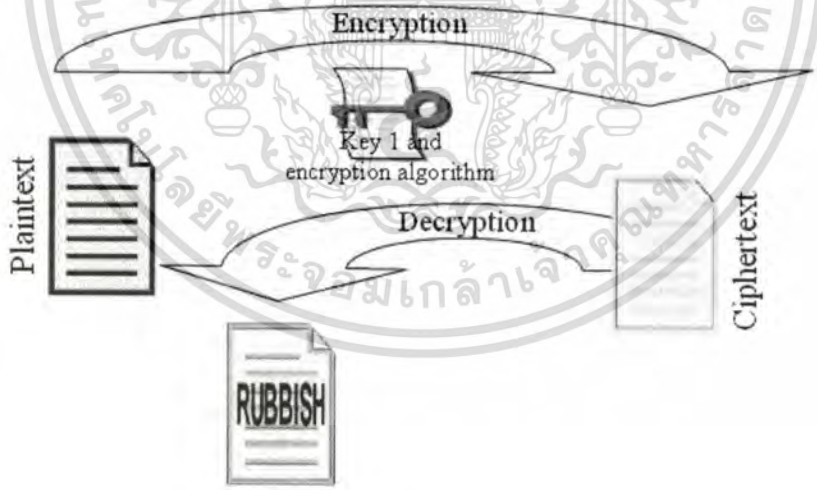
อัลกอริธึมแบบนี้จะใช้กุญแจที่เรียกว่า กุญแจลับ (Secret key) ซึ่งมีเพียงหนึ่งเดียวเพื่อใช้ในการเข้าและถอดรหัสข้อความที่ส่งไป อัลกอริธึมยังสามารถแบ่งย่อยออกเป็น 2 ประเภท ได้แก่ แบบบล็อก (Block Algorithms) ซึ่งจะทำการเข้ารหัสทีละบล็อก (1 บล็อกประกอบด้วยหลายไบต์ เช่น 64 ไบต์ เป็นต้น) และแบบสตรีม (Stream Algorithms) ซึ่งจะทำการเข้ารหัสทีละไบต์ อัลกอริธึมแบบนี้จะใช้กุญแจที่เรียกว่า กุญแจลับ (Secret key) ซึ่งมีเพียงหนึ่งเดียวเพื่อใช้ในการเข้าและถอดรหัสข้อความที่ส่งไป



รูปที่ 2.2 การเข้ารหัสโดยใช้อัลกอริธึมแบบสมมาตร

2. อัลกอริธึมแบบอสมมาตร (Asymmetric key algorithms)

อัลกอริธึมนี้จะใช้กุญแจสองตัวเพื่อทำงาน ตัวหนึ่งใช้ในการเข้ารหัสและอีกตัวหนึ่งใช้ในการถอดรหัสข้อมูลที่เข้ารหัส มาโดยกุญแจตัวแรก อัลกอริธึมกลุ่มสำคัญในแบบอสมมาตรนี้คือ อัลกอริธึมแบบกุญแจสาธารณะ (Public keys Algorithms) ซึ่งใช้กุญแจที่เรียกกันว่า กุญแจสาธารณะ (Public keys) ในการเข้ารหัสและใช้กุญแจที่เรียกกันว่า กุญแจส่วนตัว (Private keys) ในการถอดรหัสข้อมูลนั้น กุญแจสาธารณะนี้สามารถส่งมอบให้กับผู้อื่นได้ เช่น เพื่อนร่วมงานที่เราต้องการติดต่อด้วย หรือแม้กระทั่งวางไว้บนเว็บไซต์เพื่อให้ผู้อื่นสามารถดาวน์โหลดไปใช้งานได้ สำหรับกุญแจส่วนตัวนั้นต้องเก็บไว้กับผู้เป็นเจ้าของกุญแจส่วนตัวเท่านั้น และห้ามเปิดเผยให้ผู้อื่นทราบโดยเด็ดขาด อัลกอริธึมแบบกุญแจสาธารณะยังสามารถประยุกต์ใช้ได้กับการลงลายมือชื่อดิจิตอล (ซึ่งเปรียบเสมือนการลงลายมือชื่อของเราที่ใช้กับเอกสารสำนักงานทั่วไป) การลงลายมือชื่อนี้จะเป็นการพิสูจน์ความเป็นเจ้าของและสามารถใช้ได้กับการ ทำธุรกรรมต่างๆ บนอินเทอร์เน็ต เช่น การซื้อสินค้า เป็นต้น วิธีการใช้งานคือ ผู้เป็นเจ้าของกุญแจส่วนตัวลงลายมือชื่อของตนกับข้อความที่ต้องการส่งไป ด้วยกุญแจส่วนตัว แล้วจึงส่งข้อความนั้นไปให้กับผู้รับ เมื่อได้รับข้อความที่ลงลายมือชื่อมา ผู้รับสามารถใช้กุญแจสาธารณะ (ที่เป็นคู่ของกุญแจส่วนตัวนั้น) เพื่อตรวจสอบว่าเป็นข้อความที่มาจากผู้ส่งนั้นหรือไม่



รูปที่ 2.3 การเข้ารหัสโดยใช้อัลกอริธึมแบบอสมมาตร

2.2.2 อัลกอริธึม RSA

อัลกอริธึมที่จะถูกนำมาใช้ในโครงงานนี้คือ อัลกอริธึม RSA ได้รับการพัฒนาขึ้นที่มหาวิทยาลัย MIT ในปี 1977 โดยศาสตราจารย์ 3 คน ซึ่งประกอบด้วย Ronald Rivest, Adi Shamir และ Leonard Adleman ชื่อของอัลกอริธึม ได้รับการตั้งชื่อตามตัวอักษรตัวแรกของนามสกุลของ ศาสตราจารย์ทั้งสามคน การเข้ารหัสแบบ RSA นั้น ถือได้ว่าเป็นการเข้ารหัสแบบอสมมาตรที่นิยมใช้มากที่สุด ทั้งนี้เนื่องจาก RSA เป็นการเข้ารหัสแบบอสมมาตรตัวแรกที่เกิดขึ้นหลัง การเข้ารหัสแบบ Diffie-Hellman โดยมีความสามารถในการทำงานมากกว่า และ RSA เป็นลิขสิทธิ์ของบริษัท RSA Security ซึ่งอัลกอริธึมนี้สามารถใช้ในการเข้ารหัสข้อมูลรวมทั้งการลงลายมือชื่อดิจิตอลด้วย

สำหรับความปลอดภัยของ RSA นั้นถือได้ว่ามีมากเพียงพอ หากความยาวของกุญแจมีมากพอ โดยมีเหตุการณ์ในปี 1977 โดยผู้คิดค้น RSA ทั้งสามคน ได้ท้าทายลงในหนังสือ Scientific American โดยมีรางวัล 100 ดอลลาร์สำหรับผู้ที่สามารถหา Plain Text จาก Cipher Text ที่ลงไว้ในหนังสือได้ โดยเขาคาดว่าจะใช้เวลาถึง 40 พันล้านล้านปี แต่ในปี 1994 ได้มีทีมที่ใช้คอมพิวเตอร์จำนวน 1600 เครื่องร่วมกันถอดรหัส โดยใช้เวลาทั้งสิ้น 8 เดือนเท่านั้น โดยขณะนั้นใช้ความยาวกุญแจเท่ากับ 428 บิต แต่สำหรับ RSA ที่มีความยาวกุญแจมากพอ คือ 1024 บิตขึ้นไป ถือว่ามีความปลอดภัยมากพอในงานทุกๆ ด้าน โดยส่วนใหญ่ระบบต่างๆ มักจะให้เลือกความยาวกุญแจระหว่าง 512, 1024 และ 2048 บิต

การทำงานของอัลกอริธึม RSA นั้น จะเริ่มจากการหาค่าจำนวนเฉพาะ (Prime) มา 2 จำนวน คือ p และ q ที่แตกต่างกัน และจะต้องมีขนาดของความยาว (ในหน่วยเป็นบิต) เท่ากัน จากนั้นนำ p และ q มาคูณกันได้เป็น n และนำ (p -1) คูณกับ (q -1) จะได้เป็น Φ (n) จากนั้นสุ่มค่าของ e ขึ้นมาโดย e จะต้องมากกว่า 1 และน้อยว่า Φ และตัวหารร่วมมากของ Φ และ e จะต้องเป็น 1 จากนั้นใช้หลักการทางคณิตศาสตร์ Extended Euclidean Algorithm คำนวณหาค่า d (โดยที่ d จะต้องมากกว่า 1 และน้อยว่า Φ) ที่ทำให้ ed mod Φ แล้วได้เท่ากับ 1 และ e จะทำหน้าที่เป็น Public Key และ d จะทำหน้าที่เป็น Private Key

Key Generation	
Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

รูปที่ 2.4 การสร้างกุญแจเพื่อใช้ในอัลกอริธึม RSA

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

รูปที่ 2.5 การเข้ารหัสด้วยอัลกอริธึม RSA

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \pmod n$

รูปที่ 2.6 การถอดรหัสด้วยอัลกอริธึม RSA

2.2.3 อัลกอริธึม SHA-1

อัลกอริธึม SHA-1 (Secure Hash Algorithm) เป็นอัลกอริธึมที่ใช้สำหรับสร้างเมสเซสไดเจสต์ (Message Digest) หรือไดเจสต์ หรือข้อความสรุปจากเนื้อหาข้อความตั้งต้น โดยปกติข้อความสรุปจะมีความยาวน้อยกว่าความยาวของข้อความตั้งต้นมาก จุดประสงค์สำคัญของอัลกอริธึมนี้คือ การสร้างข้อความสรุปที่สามารถใช้เป็นตัวแทนของข้อความตั้งต้นได้ โดยทั่วไปข้อความสรุปจะมีความยาวอยู่ระหว่าง 128 ถึง 256 บิต และจะไม่ขึ้นกับขนาดความยาวของข้อความตั้งต้น

คุณสมบัติที่สำคัญของอัลกอริธึมสำหรับสร้างไดเจสต์มีดังนี้

- ทุกๆ บิตของไดเจสต์จะขึ้นอยู่กับทุกบิตของข้อความตั้งต้น
- ถ้าบิตใดบิตหนึ่งของข้อความตั้งต้นเกิดการเปลี่ยนแปลง เช่น ถูกแก้ไข ทุกๆ บิตของไดเจสต์จะมีโอกาสร้อยละ 50 ที่จะแปรเปลี่ยนค่าไปด้วย ซึ่งหมายถึงว่า 0 เปลี่ยนค่าเป็น 1 และ 1 เปลี่ยนเป็น 0 คุณสมบัติข้อนี้สามารถอธิบายได้ว่าการเปลี่ยนแปลงแก้ไขข้อความตั้งต้นโดย ผู้ไม่ประสงค์ดีแม้ว่าอาจแก้ไขเพียงเล็กน้อยก็ตาม เช่น เพียง 1 บิตเท่านั้น ก็จะส่งผลให้ผู้รับข้อความทราบว่าข้อความที่ตนได้รับไม่ใช่ข้อความตั้งต้น (โดยการนำข้อความที่ตนได้รับเข้าอัลกอริธึมเพื่อทำการคำนวณหาไดเจสต์ออกมา แล้วจึงเปรียบเทียบไดเจสต์ที่คำนวณได้กับไดเจสต์ที่ส่งมาให้ด้วย ถ้าต่างกัน แสดงว่าข้อความที่ได้รับนั้นถูกเปลี่ยนแปลงแก้ไข)
- โอกาสที่ข้อความตั้งต้น 2 ข้อความใดๆ ที่มีความแตกต่างกัน จะสามารถคำนวณได้ค่าไดเจสต์เดียวกันมีโอกาสน้อยมาก คุณสมบัติข้อนี้ทำให้แน่ใจได้ว่า เมื่อผู้ไม่ประสงค์ดีทำการแก้ไขข้อความตั้งต้น ผู้รับข้อความที่ถูกแก้ไขไปแล้วนั้นจะสามารถตรวจพบได้ถึงความผิดปกติที่เกิดขึ้นอย่างแน่นอนอย่างไรก็ตามในทางทฤษฎีแล้ว มีโอกาสที่ข้อความ 2 ข้อความที่แตกต่างกันจะสามารถคำนวณแล้วได้ค่าไดเจสต์เดียวกัน ปัญหานี้เรียกกันว่าการชนกันของไดเจสต์ (Collision) อัลกอริธึมสำหรับสร้างไดเจสต์ที่ดีควรจะมีโอกาสน้อยมากๆ ที่จะก่อให้เกิดปัญหาการชนกันของไดเจสต์

SHA-1 เป็นอัลกอริธึมที่แก้ไขเพิ่มเติมเล็กน้อยจาก SHA การแก้ไขเพิ่มเติมนี้เป็นที่เชื่อกันว่าทำให้อัลกอริธึม SHA-1 มีความปลอดภัยที่สูงขึ้นและ SHA-1 สร้างไดเจสต์ที่มีขนาด 160 บิต

ตารางที่ 2.1 รายละเอียดของอัลกอริธึม SHA ในรูปแบบต่างๆ

Algorithm	Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)
SHA-0	160	160	512	$2^{64} - 1$	32
SHA-1	160	160	512	$2^{64} - 1$	32
SHA-256/224	256/224	256	512	$2^{64} - 1$	32
SHA-512/384	512/384	512	1024	$2^{128} - 1$	64

2.2.4 โครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure)

เทคโนโลยีโครงสร้างพื้นฐานระบบกุญแจสาธารณะ (Public Key Infrastructure) เป็นเทคโนโลยีที่ใช้ในการรักษาความปลอดภัยของข้อมูลอิเล็กทรอนิกส์ ที่ทำการรับ-ส่งผ่านระบบเครือข่ายอินเทอร์เน็ต ซึ่งกำลังได้รับความนิยมอย่างแพร่หลาย โดยสามารถปกปิดข้อมูล ไม่ให้ผู้ไม่ได้รับอนุญาตหรือไม่มีสิทธินำข้อมูลดังกล่าวไปใช้ได้ (Confidentiality) สามารถตรวจสอบความถูกต้องครบถ้วนของข้อมูล (Data Integrity) สามารถระบุตัวตนที่แท้จริงของผู้ส่งข้อมูลอิเล็กทรอนิกส์ (Authentication) รวมทั้งสามารถป้องกันไม่ให้บุคคลผู้ส่งปฏิเสธว่าตนไม่ได้ส่งข้อมูลอิเล็กทรอนิกส์นั้นได้ (Non-repudiation)

เทคโนโลยี PKI สามารถก่อให้เกิดความน่าเชื่อถือในการระบุตัวตนระหว่างโลกแห่งความจริง (Real World) และโลกอิเล็กทรอนิกส์ (Cyber World) ได้โดยใช้เทคโนโลยีระบบรหัสแบบกุญแจสาธารณะ (Public Key Cryptography) ซึ่งประกอบด้วยกุญแจ (Key) 2 ดอก ได้แก่ กุญแจส่วนตัว (Private Key) และกุญแจสาธารณะ (Public Key) โดยที่บุคคลหนึ่งๆ จะถือกุญแจคนละ 2 ดอกดังกล่าวนี้ กุญแจส่วนตัวจะถูกเก็บอยู่กับเจ้าของกุญแจไว้อย่างปลอดภัย เพื่อใช้ในการยืนยันตัวตน และกุญแจสาธารณะจะถูกนำไปเผยแพร่ เพื่อให้บุคคลอื่นสามารถติดต่อสื่อสารกับเจ้าของกุญแจได้

ด้วยเหตุที่เทคโนโลยีระบบรหัสแบบกุญแจสาธารณะ เป็นเพียงกลไกในการทำให้เกิดความปลอดภัยในตัวข้อมูล แต่สิ่งที่ก่อให้เกิดความน่าเชื่อถือว่ากุญแจสาธารณะเป็นของบุคคลนั้นจริง จำเป็นต้องมีหน่วยงานที่มีความน่าเชื่อถือ ทำการรับรองกุญแจสาธารณะว่าเป็นของบุคคลดังกล่าว โดยการออกใบรับรองอิเล็กทรอนิกส์ (Certificate) เพื่อยืนยันความมีตัวตนที่แท้จริงในโลกอิเล็กทรอนิกส์ อันจะทำให้คู่ติดต่อสื่อสารมั่นใจว่ากำลังติดต่ออยู่กับเจ้าของกุญแจนั้น ซึ่งหน่วยงานนี้จะเรียกว่า ผู้ให้บริการออกใบรับรอง (Certification Authority: CA) ซึ่งเป็นองค์ประกอบหลักที่สำคัญของเทคโนโลยี PKI

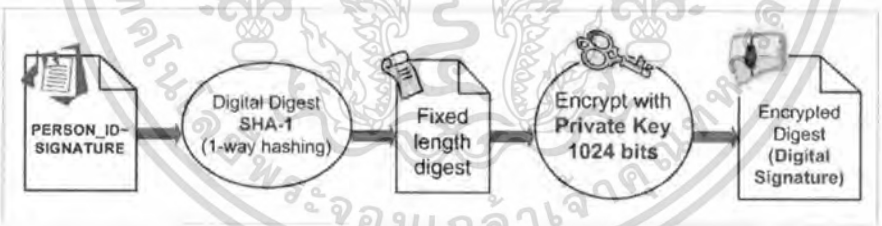
การประยุกต์ใช้งานเทคโนโลยี PKI สามารถทำได้ 2 วิธีคือ การเข้ารหัสลับ/การถอดรหัสลับ (Encryption/Decryption) เพื่อรักษาความลับของข้อมูลอิเล็กทรอนิกส์ และการลงลายมือชื่อดิจิตอล/ตรวจสอบลายมือชื่อดิจิตอล (Digital Signing/Verifying) เพื่อให้สามารถตรวจสอบตัวตนของผู้ส่งข้อมูลอิเล็กทรอนิกส์ และยังสามารถยืนยันได้ว่าข้อมูลนั้นไม่ถูกเปลี่ยนแปลงแก้ไขในระหว่างการส่ง โดยอาศัยกุญแจส่วนตัวของผู้ส่งซึ่งจะถูกเก็บไว้ที่ผู้ส่งเพียงผู้เดียวเป็นปัจจัยสำคัญ ดังนั้นผู้ส่งจึงจำเป็นต้องเก็บรักษาและป้องกันกุญแจส่วนตัวของตนไว้อย่างเป็นความลับและปลอดภัย

2.2.5 ลายมือชื่อดิจิตอล (Digital Signature)

ลายมือชื่อดิจิตอล คือข้อมูลอิเล็กทรอนิกส์ที่ได้จากการเข้ารหัสข้อมูลด้วยกุญแจส่วนตัวของผู้ส่งซึ่งเปรียบเสมือนเป็นลายมือชื่อของผู้ส่ง คุณสมบัติของลายมือชื่อดิจิตอลนั้น นอกจากจะสามารถใช้ในการระบุตัวบุคคล และเป็นกลไกการป้องกันการปฏิเสธความรับผิดชอบแล้ว ยังสามารถป้องกันข้อมูลที่ส่งไปไม่ให้ถูกแก้ไข หรือ หากถูกแก้ไขไปจากเดิมก็สามารถล่วงรู้ได้

2.2.5.1 การสร้างลายมือชื่อดิจิตอล (Digital Signature)

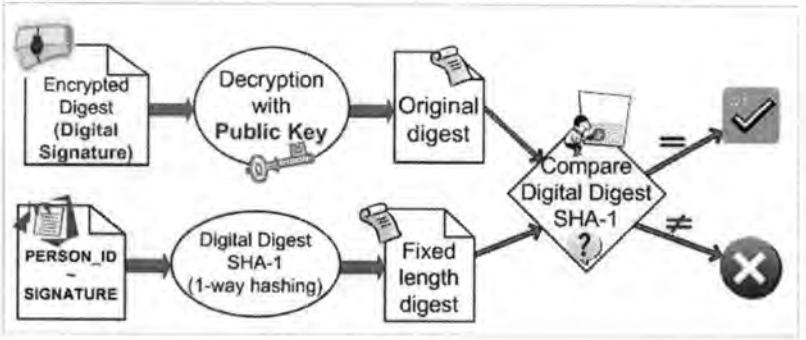
กระบวนการในการสร้าง Digital Signature นั้น มีขั้นตอนคือ จะนำเอาข้อมูลตั้งต้นทำการย่อย (Digest) ด้วย Algorithm หนึ่ง เพื่อให้ได้ข้อมูลที่มีขนาดเท่ากันทุกครั้งก่อนที่จะทำการ Sign ด้วย Algorithm หนึ่ง เช่น RSA ซึ่ง Key ที่ใช้ในการ Sign คือ Private Key ที่มีขนาด 1024-bits ผลลัพธ์ที่ได้จะออกมาเป็น Digital Signature



รูปที่ 2.7 แสดงขั้นตอนการสร้าง Digital Signature

2.2.5.2 การตรวจสอบลายมือชื่อดิจิตอล (Digital Signature)

การตรวจสอบ Digital Signature หรือ Verify นั้น Key ที่ใช้ในคือ Public Key ซึ่งเป็นส่วนหนึ่งของ Private Key และจะต้องใช้งานเป็นคู่กันเสมอ โดยจะต้องนำเอา Digital Signature ดังกล่าวมาถอดรหัส เพื่อให้ได้เป็น Message Digest หลังจากนั้นจึงนำเอา ข้อมูลต้นฉบับที่ได้มาย่อยด้วย Algorithm เดียวกัน ก็จะได้เป็น Message Digest อีกหนึ่งชุด จึงนำมาเปรียบเทียบกันกับชุดแรก หากตรงกันแสดงว่า ข้อมูลดังกล่าว ไม่ได้ถูกเปลี่ยนหรือแก้ไขไปจากเดิม



รูปที่ 2.8 แสดงขั้นตอนการตรวจสอบ Digital Signature

2.3 บริการบัญชีรายชื่อผู้ใช้งาน (Directory Service)

Directory Service เป็นกลุ่มของโปรแกรมที่ให้บริการ ในการจัดเก็บและจัดการกับข้อมูลเกี่ยวกับผู้ใช้งานคอมพิวเตอร์ในระบบเครือข่ายและทรัพยากรต่างๆ โดยมีผู้ดูแลระบบเป็นผู้บริหารจัดการ ในเรื่องของสิทธิในการเข้าถึงทรัพยากรของผู้ใช้งานดังกล่าว โดย Directory Service ทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้งานระบบกับทรัพยากรที่ต้องมีการใช้งานร่วมกัน

2.3.1 ลักษณะการให้บริการ

การให้บริการของ Directory Service นั้น จะมีการจัดเก็บบัญชีรายชื่อของผู้ใช้งานที่มีรูปแบบเป็นโครงสร้าง มีการสืบค้นได้ และสามารถใช้ในเรื่องของการพิสูจน์ตัวจริงด้วย เนื่องจากจะมีการเก็บข้อมูลของผู้ใช้งานนั้นๆเอาไว้ เช่น ชื่อ นามสกุล ชื่อผู้ใช้งาน รหัสผ่าน และ Email เป็นต้น

2.3.2 มาตรฐานที่เกี่ยวกับบริการบัญชีรายชื่อผู้ใช้งาน

มาตรฐานที่เกี่ยวกับ Directory Service คือ X.500 ซึ่งเป็นมาตรฐานที่มีประสิทธิภาพสูง สามารถจัดเก็บข้อมูลได้หลากหลาย และมีการจัดเก็บข้อมูลแบบมีโครงสร้าง โดยการติดต่อจะใช้โปรโตคอลในการเข้าถึงข้อมูล คือ Directory Access Protocol (DAP) ในการเชื่อมต่อเข้าถึงข้อมูล X.500 Directory และในปัจจุบันได้มี โปรโตคอลที่ใช้สำหรับเชื่อมต่อเข้าถึงข้อมูล X.500 แทนโปรโตคอล DAP ที่มีความซับซ้อนมาก ทำให้สิ้นเปลืองทรัพยากร และระบบล่าช้า นั่นก็คือ Lightweight Directory Access Protocol (LDAP) และจะต้องมีเครื่องแม่ข่าย (Server) ในการจัดเก็บข้อมูลเหล่านี้ ตัวอย่างของผู้ให้บริการหรือเครื่องแม่ข่าย ในลักษณะของ Directory Service ในปัจจุบันได้แก่ Active Directory (Microsoft) Sun Java System Directory Server (SUN), Fedora Directory Server, IBM Tivoli Directory Server (IBM) และ Lotus Domino Directory (IBM) เป็นต้น

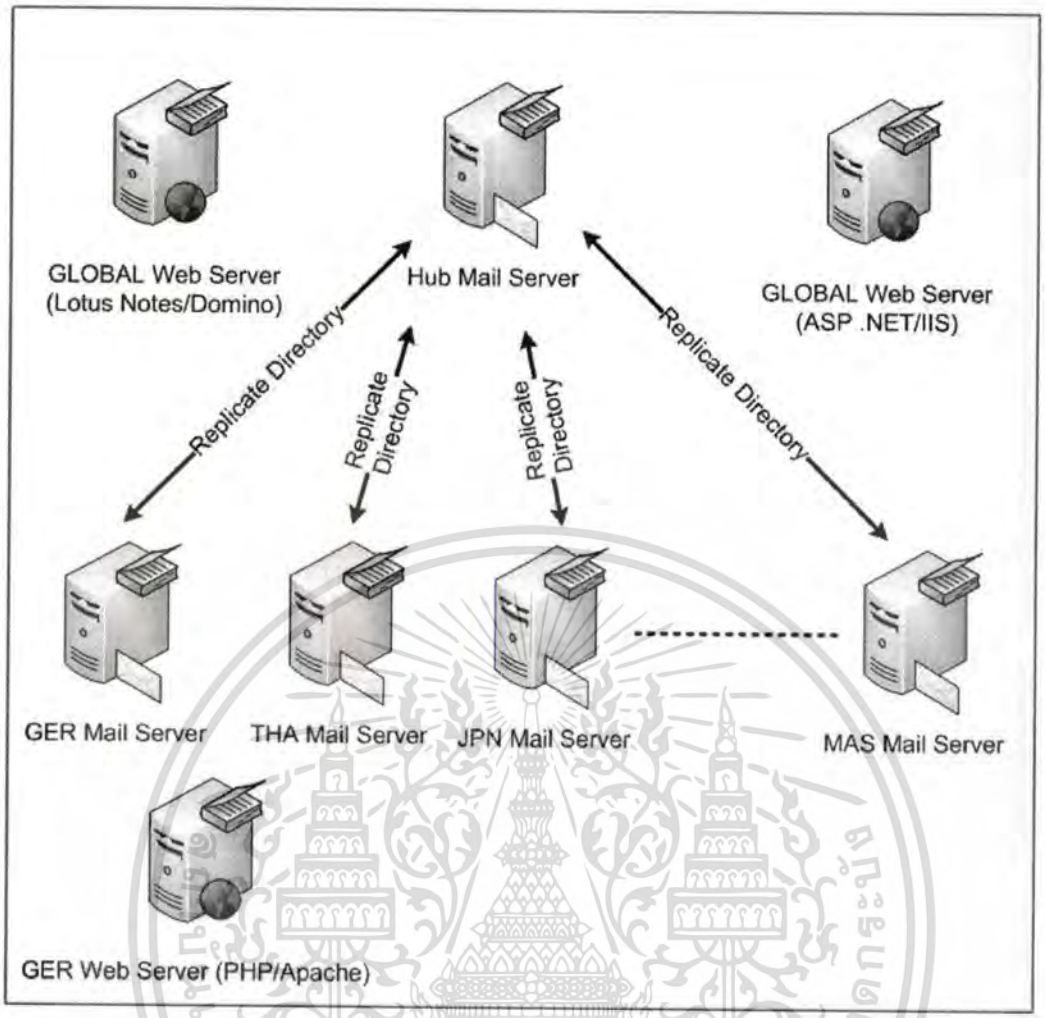
บทที่ 3
การวิเคราะห์และออกแบบ
ระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ

ในหัวข้อนี้จะกล่าวถึงการวิเคราะห์แนวทาง หน้าที่ และบริการที่ระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ จะให้บริการกับระบบอื่นๆ ที่มาขอใช้บริการ ซึ่งเนื้อหาในบทนี้จะกล่าวถึงหน้าที่หลักคือ ให้บริการเรื่องบัญชีรายชื่อผู้ใช้งาน และบริการพิสูจน์ตัวจริง องค์ประกอบของระบบ และการออกแบบ ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการออกแบบเพื่อให้ระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ เหมาะสมกับที่จะนำไปใช้งานในองค์กร และเพื่อเพิ่มประสิทธิภาพในการทำงานร่วมกันของระบบต่างๆ

3.1 วิเคราะห์สภาพแวดล้อมและสมมติฐานเบื้องต้นของระบบ

ในโครงงานนี้ จะนำเสนอโดยอ้างอิงตามสภาพแวดล้อมขององค์กรหนึ่ง ซึ่งมีลักษณะคือ องค์กรนี้เป็นองค์กรขนาดกลาง ซึ่งมีผู้ร่วมองค์กร (Partner) กระจายอยู่ตามประเทศต่างๆ ทั่วโลก โดยมีการใช้งานระบบไปรษณีย์อิเลคทรอนิกส์ (E-Mail) ในการติดต่อสื่อสารกันในทุกผู้ร่วมองค์กร ของทุกประเทศ โดยใช้งานระบบ E-Mail ของ Lotus Domino Server เป็นเครื่องแม่ข่าย ซึ่งในทุก Partner จะมี เครื่องแม่ข่ายที่ติดตั้งระบบดังกล่าวไว้ และมีสมุดรายชื่อ (Directory) เพื่อใช้ในการสืบค้นและระบุที่อยู่ของผู้ใช้งาน ซึ่ง Directory ดังกล่าวจะมีการทำสำเนา (Replicate) กัน เพื่อแลกเปลี่ยนข้อมูลให้มีความทันสมัยอยู่เสมอ ซึ่งการใช้งาน E-Mail สามารถเข้าใช้งานผ่านทางเว็บไซต์ ของเครื่อง Server ของประเทศตนเองได้

ในส่วนของการใช้งานโปรแกรมประยุกต์ ในปัจจุบันมีการตั้งเครื่อง Server เพื่อให้บริการโปรแกรมประยุกต์ไว้เป็นส่วนกลางเพื่อให้สามารถใช้งานร่วมกันได้ ซึ่งโปรแกรมประยุกต์ดังกล่าว เป็นโปรแกรมประยุกต์บนเว็บ ซึ่งผู้ร่วมองค์กรในประเทศต่างๆต้องใช้งาน และในการเข้าใช้งานนั้นจะต้องเป็นผู้ที่มีสิทธิ์เท่านั้น ระบบดังกล่าวได้มีการจัดการในเรื่องผู้ใช้งานภายในระบบเอง ซึ่งกลุ่มของผู้ใช้งานระบบนี้ บางส่วนจะซ้ำซ้อนกับผู้ใช้งานบนระบบ E-Mail ซึ่งมีบัญชีรายชื่ออยู่แล้ว หากแต่มีการจัดเก็บไว้ต่าง Directory กัน และระบบดังกล่าวพัฒนาด้วย Microsoft ASP.NET ซึ่งเป็นอีกแพลตฟอร์ม และมีบางส่วนที่ถูกพัฒนาโดย Lotus Notes ซึ่งเป็นระบบกลางที่ต้องใช้งานร่วมกัน รวมไปถึงระบบอื่นๆ ที่พัฒนาขึ้นโดยผู้ร่วมองค์กรบางประเทศซึ่งมีการพัฒนาขึ้นเอง โดยมีการใช้ PHP, Ruby on Rails หรือภาษาในกลุ่มของ Open Source อีกด้วย ซึ่งผู้ใช้งานระบบบางส่วนก็เป็นกลุ่มเดียวกับที่มีในส่วนกลาง โดยแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดงแผนผังของระบบที่มีอยู่ในองค์กรโดยย่อ

เนื่องมาจากองค์กรดังกล่าว ผู้ร่วมองค์กรไม่มีการขึ้นต่อกันแต่จะมี ผู้ร่วมองค์กรกลางที่มีหน้าที่ประสานงานดูแลระบบในส่วนกลางและบัญชีรายชื่อส่วนกลางให้ จึงเกิดความหลากหลายของแพลตฟอร์มที่ใช้พัฒนาระบบ จากรูปแบบดังกล่าวจึงทำให้ผู้ใช้งานไม่ได้รับความสะดวกหากผู้ใช้งานดังกล่าวมีสิทธิและต้องใช้งานหลากหลายระบบ รวมทั้งเกิดความซ้ำซ้อนของชื่อบัญชีผู้ใช้งานด้วย

ด้วยเหตุนี้ จึงได้มีการนำเอาระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บมาใช้เพื่อให้เกิดการทำงานร่วมกันที่มีประสิทธิภาพมากยิ่งขึ้น โดยผู้ร่วมองค์กรกลางจะเป็นผู้จัดทำประสานงานรวมทั้งดูแลการทำงานของระบบ ให้สามารถทำงานได้ถูกต้องและต่อเนื่อง โดยระบบซิงเกิ้ลไซออนนั้น จะต้องมีกลไกในการจัดทำ ทั้งในส่วนของการปรับแต่งโครงสร้างพื้นฐาน (Infrastructure) และการเขียนโปรแกรม (Programming) ด้วย

3.2 วิเคราะห์หน้าที่และออกแบบบริการของระบบ

เมื่อวิเคราะห์จากวัตถุประสงค์ในการจัดทำระบบแล้ว จะเห็นว่ามีผู้ที่เกี่ยวข้องกับระบบแบ่งออกเป็นกลุ่มต่างๆ คือ ผู้ใช้งาน ผู้ดูแลระบบ และผู้พัฒนาระบบ ซึ่งแต่ละกลุ่มจะได้รับประโยชน์แตกต่างกันไปจากบริการที่มีอยู่ โดยหน้าที่และบริการหลักของระบบจะเกี่ยวข้องกับบัญชีรายชื่อผู้ใช้งาน และการตรวจพิสูจน์ตัวจริง

หน้าที่ของระบบซิงเกิ้ลไซออนที่มีต่อผู้ใช้งานคือ ให้ผู้ใช้งานสามารถเข้าใช้งานระบบต่างๆ ที่มีแพลตฟอร์มที่แตกต่างกัน เสมือนกับว่าใช้งานอยู่บนระบบเดียว เพื่อให้เกิดคุณสมบัติ Transparency ในการเข้าใช้งานระบบขององค์กร ทำให้ผู้ใช้งานสามารถทำงานระบบต่างๆ ร่วมกันได้อย่างต่อเนื่องและมีประสิทธิภาพ นอกจากนั้นยังอำนวยความสะดวกในเรื่องชื่อบัญชีผู้ใช้งาน เนื่องจากผู้ใช้งานหนึ่งคน อาจมีชื่อบัญชีผู้ใช้งานได้หลายๆชื่อ เนื่องจากต้องใช้งานหลายระบบทำให้อาจเกิดความสับสนได้ หน้าที่ของซิงเกิ้ลไซออนจึงทำหน้าที่รวมชื่อบัญชีเหล่านั้นให้สามารถใช้งานชื่อบัญชีเดียว เข้าใช้งานได้ทุกระบบที่ร่วมบริการ ทำให้ผู้ใช้งานไม่ต้องจดจำหลายๆบัญชี และหากมีการแก้ไข เปลี่ยนแปลงข้อมูลก็สามารถแก้ไขได้ง่าย เนื่องจากแก้ไขที่เดียว

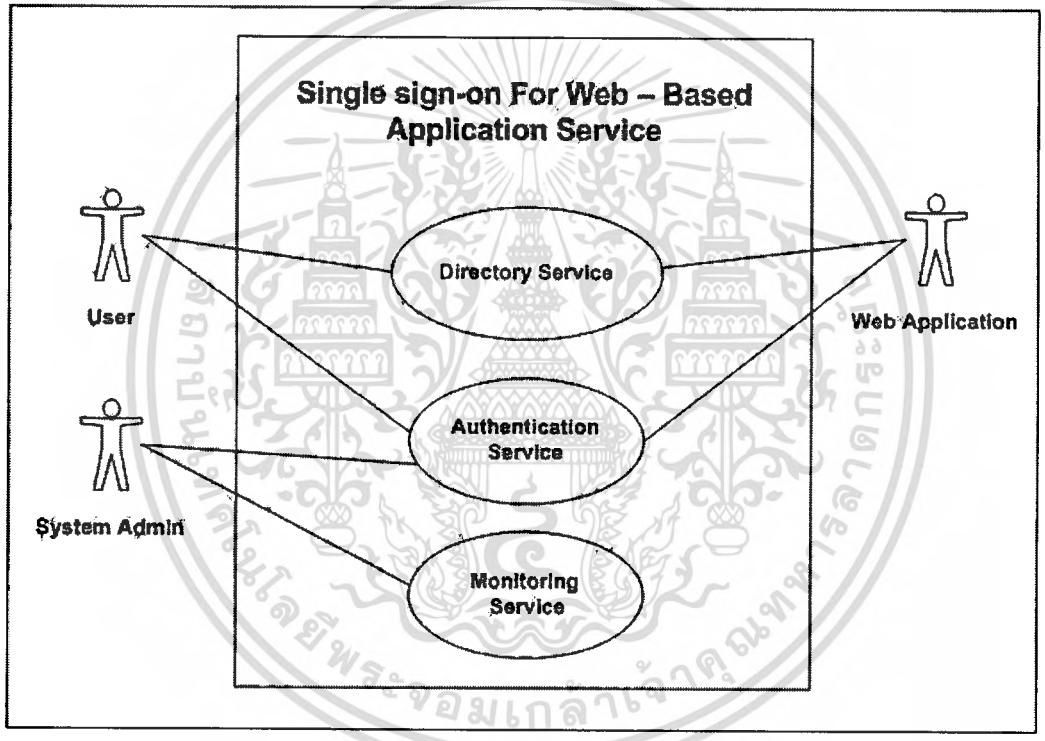
หน้าที่ของระบบซิงเกิ้ลไซออนที่มีต่อผู้ดูแลระบบคือ ให้ผู้ดูแลระบบสามารถจัดการกับบัญชีผู้ใช้งานได้สะดวกและสามารถตรวจสอบได้ง่ายเมื่อเกิดข้อขัดข้อง เนื่องจากบริการบัญชีรายชื่อผู้ใช้งาน (Directory Service) จะรวมไว้ที่เดียวกัน

หน้าที่ของซิงเกิ้ลไซออนที่มีต่อผู้พัฒนาระบบคือ เป็นการวางแนวทางในการพัฒนาระบบในส่วนที่จะต้องมีการพิสูจน์ตัวจริง ให้เป็นรูปแบบ มาตรฐานเดียวกัน เพื่อให้สามารถปรับแต่งระบบได้ง่าย ไม่ขึ้นกับภาษาที่ใช้พัฒนา ทำให้ผู้พัฒนาระบบเข้าใจตรงกัน สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ

เมื่อพิจารณาจากหน้าที่ของระบบแล้วสามารถแบ่งบริการของระบบ Single sign-on for Web –Based Application ออกเป็น 3 บริการหลัก โดยแสดงได้ดังรูปที่ 3.2 คือ

1) บริการบัญชีรายชื่อผู้ใช้งาน (Directory Service) ผู้ที่เกี่ยวข้องคือ Web Application ที่เข้าร่วมบริการ โดยจะมีการติดต่อขอใช้บริการในส่วนทั้งการเรียกดูข้อมูลบัญชีรายชื่อผู้ใช้งาน และการจัดการแก้ไข ปรับปรุงข้อมูลบัญชีชื่อผู้ใช้งาน และ User ผู้ใช้งานระบบ ก็อาจจะมีการเรียกใช้งานบริการในส่วนนี้ทั้งทางตรง เช่น การเปลี่ยนแปลงรหัสผ่านของตน หรือว่าทางอ้อมโดยการเรียกใช้งานผ่านทาง Web Application

2) บริการพิสูจน์ตัวจริง (Authentication Service) ผู้ที่เกี่ยวข้องคือ Web Application ที่เข้าร่วมบริการ และUser ผู้ใช้งานระบบ ซึ่งเมื่อผู้ใช้งานระบบจะทำการขอเข้าใช้งาน Web Application โดยตรงแต่จะถูก Redirect มาเพื่อขอใช้บริการพิสูจน์ตัวจริงของระบบ Single sign-on นี้ ในส่วนของผู้ดูแลระบบที่จะเข้ามาตรวจสอบดูแลการทำงานก็ต้องผ่านการพิสูจน์ตัวจริงด้วย
3) บริการตรวจสอบและดูแลการทำงาน (Monitoring Service) ผู้ที่เกี่ยวข้องคือ System Admin ผู้ดูแลระบบ ซึ่งจะสามารถดูแลและตรวจสอบการทำงานของระบบหากเกิดข้อขัดข้อง และสามารถดู Log หรือสถิติบางอย่างได้ เช่น เวลาหรือข้อมูลของการเข้าใช้งาน เป็นต้น



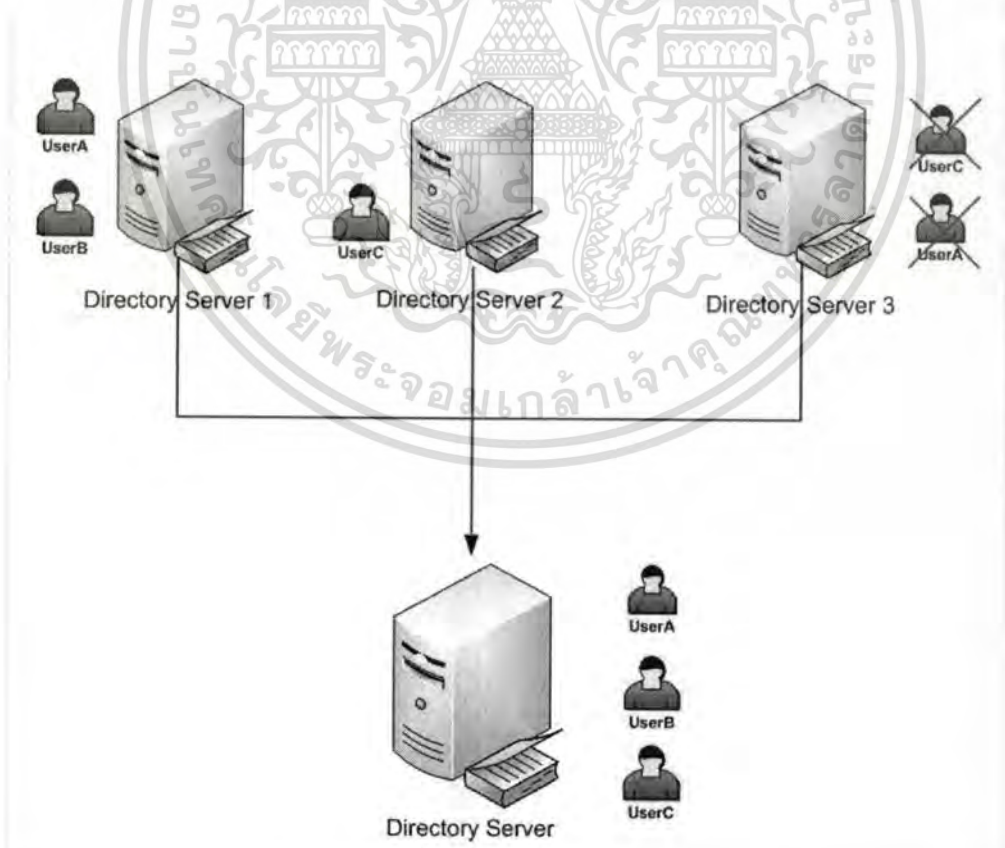
รูปที่ 3.2 บริการหลักของระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ

3.2.1 บริการบัญชีรายชื่อผู้ใช้งาน (Directory Service)

เป็นบริการที่เปรียบเสมือนฐานข้อมูลของระบบสำหรับ โครงงานระบบชิงเกิ้ลไซออน สำหรับโปรแกรมประยุกต์บนเว็บ โดยบริการบัญชีรายชื่อผู้ใช้งานจะเก็บข้อมูลรายละเอียดโดยย่อ (Profile) ของผู้ใช้งานทั้งหมด ซึ่งบริการนี้จะต่างจากฐานข้อมูลของระบบที่มีการดำเนินการ (Transaction) โดยทั่วตรงที่ว่าจะมีการอ่านข้อมูลมากกว่าการเขียนบันทึกข้อมูล เนื่องจากจะใช้บริการบัญชีรายชื่อผู้ใช้งานนี้ ในเรื่องของการพิสูจน์ตัวจริงเป็นส่วนใหญ่

3.2.1.1 ลักษณะของบริการบัญชีรายชื่อผู้ใช้งาน

บริการบัญชีรายชื่อผู้ใช้งานที่ใช้ในระบบชิงเกิ้ลไซออนนี้ จะเป็นลักษณะการรวมข้อมูลเข้าสู่ส่วนกลาง (Centralization) เพื่อง่ายต่อการบริหารจัดการ ลดความซ้ำซ้อนของข้อมูล (Redundancy) โดยยุบรวม Directory ของระบบต่างๆ ให้เหลือเพียงแค่ Directory เดียว เพื่อให้ผู้ใช้งานมีแค่บัญชีเดียวแต่สามารถใช้งานได้ทุกระบบ (Single User) หากอนาคตมีการเพิ่มระบบก็ไม่จำเป็นจะต้องเพิ่มบัญชีรายชื่อผู้ใช้ ที่ระบบนั้นๆ เพียงจัดการให้ระบบนั้นเข้าร่วมใช้บริการของระบบชิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ ระบบต่างๆ เหล่านั้นก็จะสามารถรู้จักบัญชีรายชื่อของระบบทั้งหมดเอง โดยแสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 บริการบัญชีรายชื่อผู้ใช้งานที่ถูกยุบรวมกัน

ในส่วนของ Directory หลักนี้จะ มีลักษณะเป็นโดเมน โดยเป็นเสมือนขอบเขตของระบบรักษาความปลอดภัย (Security Boundary) และขอบเขตการบริหาร (Administrative Boundary) ด้วย ซึ่งสามารถอธิบายได้ดังนี้

- ขอบเขตของระบบรักษาความปลอดภัย (Security Boundary) เมื่อผู้ใช้ล็อกอินเข้าสู่โดเมนแล้ว ผู้ใช้จะสามารถใช้ทรัพยากร ในระบบต่างๆได้ตามสิทธิที่ตัวเองมีในระบบนั้นๆ ด้วย Identity ที่ผู้ใช้งาน ใช้ในการล็อกอิน เปรียบได้กับคนที่ถือบัตรประชาชนเมื่อถูกตรวจสอบก่อนเข้าบัตรและมีการยืนยันตัวตนแล้ว ก็สามารถเข้าใช้งานทรัพยากรต่างๆ โดยไม่ต้องถูกตรวจสอบอีกครั้ง แต่จะใช้ได้ในขอบเขตที่ได้มีการกำหนดไว้เท่านั้น

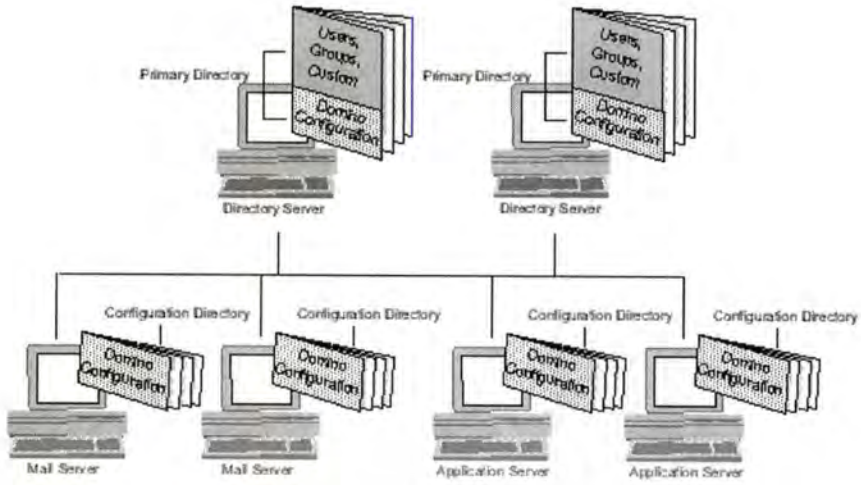
- ขอบเขตของการบริหาร (Administrative Boundary) ผู้ดูแลระบบโดเมนหรือผู้บริหารระบบซิงเกิ้ลไซออนจะมีสิทธิ์เต็มที่ในการจัดการบัญชีรายชื่อผู้ใช้ทั้งหมด ส่วนผู้ดูแลระบบของระบบที่มาใช้บริการจะสามารถบริหารจัดการบัญชีรายชื่อผู้ใช้ได้เฉพาะภายในส่วนของตนเท่านั้น ไม่มีสิทธ์เข้าไปจัดการกับบัญชีรายชื่อผู้ใช้ของระบบอื่น ยกเว้นแต่ผู้ดูแลระบบนั้นๆ จะอนุญาตให้สิทธิ์เท่านั้นหรือกล่าวได้ว่า มี Authorization ใน Directory Service นี้ด้วยนั่นเอง

3.2.1.2 ลักษณะของเครื่องผู้ให้บริการบัญชีรายชื่อผู้ใช้งาน

เนื่องจากบริการบัญชีรายชื่อผู้ใช้งาน ของระบบซิงเกิ้ลไซออนเป็นการรวมบัญชีรายชื่อของระบบต่างๆ เข้าด้วยกันดังนั้น จะต้องใช้ระบบที่สามารถรองรับบัญชีรายชื่อจำนวนมากได้ แต่อย่างไรก็ตาม ก็ขึ้นอยู่กับจำนวนผู้ใช้งานของทุกระบบในองค์กรว่ามากน้อยเพียงใด

การใช้งานบัญชีรายชื่อของระบบซิงเกิ้ลไซออน จะใช้ในเรื่องการพิสูจน์ตัวจริงเป็นส่วนหลัก ส่วนในลักษณะอื่นๆ นอกเหนือจากการพิสูจน์ตัวจริง เช่น การสืบค้นเพื่อขอดูรายละเอียดแบบย่อ การค้นหาเพื่อส่งจดหมายอีเล็กโทรนิกส์ เป็นต้น สำหรับในโครงงานนี้จึงเลือก IBM Lotus Domino Server เพื่อใช้เป็น Directory Server เนื่องจากตามสมมติฐานเบื้องต้นองค์กรมีการใช้งาน IBM Lotus Domino Server ในเรื่องของระบบไปรษณีย์อิเลคทรอนิกส์อยู่แล้ว และ Domino Server ยังมีลักษณะอื่นๆ ที่สามารถนำมาใช้เพื่อให้เกิดระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ ได้ เช่น ความสามารถในการทำ Replication, การทำ Domino LDAP Server, การทำ Domino Web Server เป็นต้น ด้วยความสามารถเหล่านี้จึงเป็นปัจจัยที่ทำให้เลือก Domino Server โดยการทำ Replication เพื่อใช้ในการทำ Fault Tolerant และ การทำ Fail Over ก็สามารถทำได้ง่ายด้วย

โดยลักษณะของ Domino Directory Server สามารถแบ่ง Directory ออกเป็นส่วนๆ เพื่อควบคุมสิทธิ์การเข้าใช้งานของทรัพยากรต่างๆ ในระบบได้ โดยที่ Domino Directory นี้จะเก็บเอกสารควบคุมการปรับแต่งระบบ (Configuration Document) เอาไว้อีกด้วย โดยจะเก็บเอกสารเหล่านี้อยู่ในรูปแบบ Document ของ IBM Lotus Notes Database ซึ่งสามารถกำหนดสิทธิ์การเข้าถึงได้เป็นระดับต่างๆ ด้วยโปรโตคอลของตัว Domino Server เอง โดยแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 สถาปัตยกรรมของ Central directory ใน Domino domain

โดยบัญชีผู้ใช้งานจะถูกเก็บเป็น Document ภายใน Lotus Notes Database ที่ทำหน้าที่เป็น Directory และเรียก Document นี้กว่า Person Document ซึ่ง 1 Person Document จะเก็บรายละเอียดของ 1 บัญชีผู้ใช้งาน โดยสามารถเก็บรายละเอียดต่างๆได้มากมาย แต่ในการจัดทำระบบชิงเกิ้ลไซออนนี้ จะเก็บเฉพาะข้อมูลโดยทั่วไปและข้อมูลที่สามารถระบุตัวตนของผู้ใช้งานได้ เท่านั้น โดยแสดงได้ดังรูปที่ 3.5 ส่วนรายละเอียดเฉพาะและข้อมูลเชื่อมโยงต่างๆของผู้ใช้งานจะเก็บที่ระบบนั้น โดยจะมีค่าของข้อมูลหนึ่งที่ทำหน้าที่เชื่อมโยงไปยังระบบดังกล่าวเพื่อระบุชื่อบัญชีผู้ใช้งานนั้น

รูปที่ 3.5 ลักษณะของบัญชีผู้ใช้งาน (Person Document) ที่เก็บใน Domino Directory

ข้อมูลของบัญชีผู้ใช้งานที่จะถูกเก็บใน Domino Directory นั้นจะเป็นข้อมูลกลางที่สามารถเข้าถึงได้โดยระบบต่างๆ ซึ่งเป็นข้อมูลที่สามรถระบุตัวตนไปยังระบบนั้นได้ โดยแสดงรายละเอียดดังตารางที่ 3.1

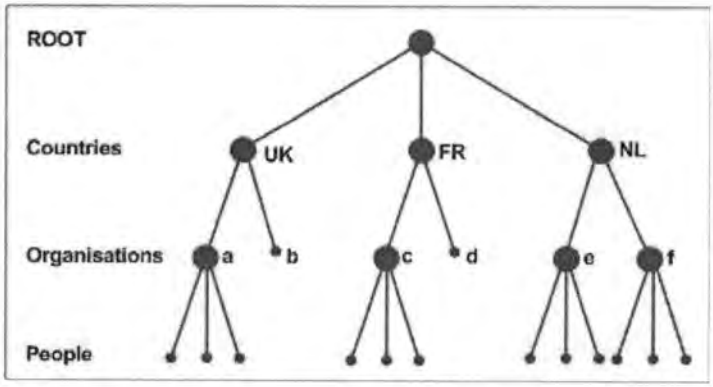
ตารางที่ 3.1 รายละเอียดทั่วไปของบัญชีผู้ใช้งานที่สามารถใช้งานร่วมกันกับระบบอื่นๆ

No	Type	Field Name	Uniqueness	Remark
1	M	First Name	No	ชื่อเจ้าของบัญชีผู้ใช้งาน
2	M	Last Name	No	นามสกุลเจ้าของบัญชีผู้ใช้งาน
3	M	OU	No	รหัสตัวย่อของหน่วยงานโดยในที่นี้ใช้เป็นตัวย่อแทนประเทศของ Partner มี 3 ตัวอักษร
4	M	UserID	Yes	ชื่อผู้ใช้งานที่ใช้ในการ Log in
5	M	PersonID	Yes	รหัสประจำชื่อบัญชีผู้ใช้งานเพื่อใช้เชื่อโยงกับระบบต่างๆ
6	M	Password	No	รหัสผ่านถูกเก็บโดยมีการเข้ารหัสด้วยอัลกอริธึมของ Domino Directory
7	O	Email	No	ชื่อที่อยู่ของไปรษณีย์อิเลคทรอนิกส์
8	O	Secret Question	No	คำถามลับเพื่อใช้ในการ Reset Password
9	O	Secret Answer	No	คำตอบลับเพื่อใช้ในการ Reset Password

จากตารางที่ 3.1 ค่าของ Type ที่เป็น M (Mandatory) แทนชนิดข้อมูลที่จำเป็นสำหรับชื่อบัญชีผู้ใช้งาน ส่วนค่าที่เป็น O (Optional) แทนชนิดของข้อมูลที่ไม่จำเป็น มีหรือไม่ ก็ได้ ซึ่งข้อมูลเหล่านี้จะเป็นข้อมูลที่ระบบต่างๆเข้าใจตรงกันโดยเก็บไว้ส่วนกลาง และจะถูกเก็บใน Person Document ของ Domino Directory แต่จะมีข้อมูลอีกบางส่วนที่เกิดจากการคำนวณหรือนำมาประกอบกัน เพื่อใช้ในระบบของ Domino Directory ด้วย

3.2.1.3 การการตั้งชื่อบัญชีชื่อผู้ใช้งาน

การตั้งชื่อบัญชีผู้ใช้งานสำหรับ Domino Directory นั้นจะใช้การตั้งชื่อแบบตามลำดับชั้น (Hierarchical Naming) เพื่อให้สามารถแบ่งผู้ใช้งานตามโครงสร้างองค์กรได้อย่างชัดเจนและมีความเป็นหนึ่งเดียวให้มากที่สุด โดยแสดงได้ดังรูปที่ 3.6



รูปที่ 3.6 แสดงโครงสร้างอย่างง่ายของชื่อบัญชีใน Directory

การตั้งชื่อแบบมีลำดับชั้น (Hierarchical Naming) ของ Domino Directory นั้นจะยึดมาตรฐาน X.500 เป็นหลัก ซึ่งสามารถช่วยในเรื่องขององค์ประกอบของความปลอดภัย เพื่อให้สามารถกำหนดสิทธิ์ในการเข้าถึงทรัพยากรที่มี ของระบบทั้งหมด และสามารถป้องกันความซ้ำซ้อนของบัญชีชื่อผู้ใช้งานที่อาจเกิดขึ้นได้ องค์ประกอบของการตั้งชื่อแบบมีลำดับชั้นแสดงรายละเอียดได้ดังตารางที่ 3.2

ตารางที่ 3.2 องค์ประกอบของการตั้งชื่อแบบมีลำดับชั้น

Term	Description	Characters	Required
Common Name (CN)	The person's full first and last names, or the server name	80 maximum	Yes
Organizational Unit Name (OU)	Typically, a department or location name	Up to 32 per OU	No
Organization Name (O)	Typically, a company name	3 to 64	Yes
Country (C)	ISO standard two-letter abbreviation for the country and top-level location	0 or 2	No

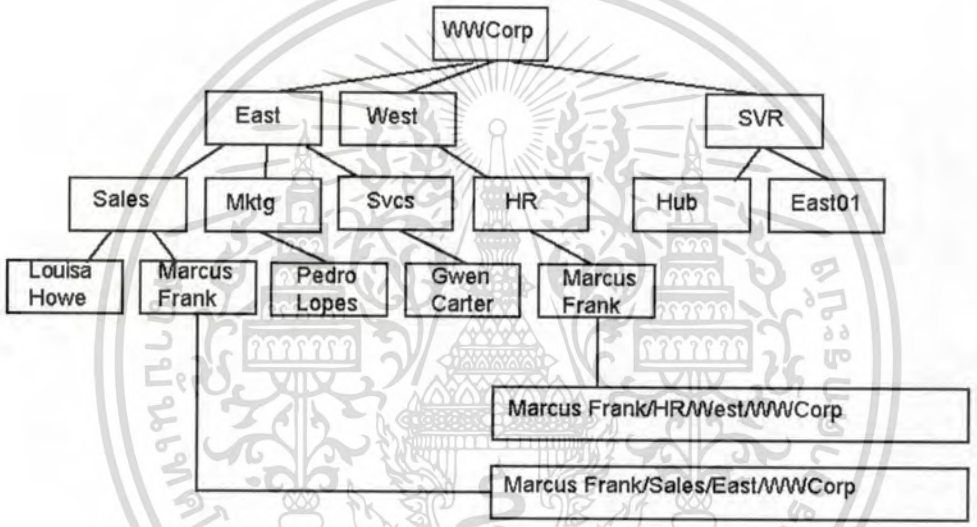
รูปแบบของชื่อที่มีลำดับชั้น CN/OU1/OU2/OU3/OU4/O/C

ตัวอย่างเช่น Sarah Forbes/Toronto/Acme/CA

โดยจากองค์กรตามสมมติฐานเบื้องต้นของโครงงานนี้ ผู้ร่วมองค์กรจะอยู่ตามประเทศต่างๆ ดังนั้นจึงใช้ OU เป็นชื่ออักษรย่อ 3 ตัวอักษรแทนชื่อประเทศ ที่ผู้ร่วมองค์กรนั้นตั้งอยู่ ดังนั้นรูปแบบของชื่อบัญชีผู้ใช้งานจึงเป็นดังนี้

FirstName LastName (UserID) /OU/CUSTOMER

ซึ่งรูปแบบดังกล่าวจะทำให้สามารถป้องกันความซ้ำซ้อนของผู้ที่มีชื่อและนามสกุล ซ้ำกันได้ โดยรูปแบบนี้ Common Name (CN) ก็คือค่าที่ได้จากการนำมารวมกันของ FirstName, LastName และ UserID ส่วน Organization Name (O) ในที่นี้ระบุเป็น "Customer" ไว้ และลดขั้น Country (C) ออกไปเนื่องจากโครงสร้างดังกล่าวพอเพียงต่อการใช้งานแล้ว



รูปที่ 3.7 แสดงการป้องกันชื่อผู้ใช้งานซ้ำของ Hierarchical Naming

จากรูปที่ 3.7 แสดงให้เห็นถึงการป้องกันปัญหาชื่อซ้ำกัน อันอาจจะเกิดขึ้นได้เนื่องจากชื่อและนามสกุลซ้ำกัน แต่เนื่องจากมี O และ OU เป็นตัวแบ่งแยกทำให้เห็นความแตกต่างชัดเจนขึ้นว่าเป็นคนละคนกัน

3.2.1.4 การติดต่อกับบริการบัญชีรายชื่อผู้ใช้งาน

การติดต่อกับบริการบัญชีรายชื่อผู้ใช้งานใน โครงงานระบบซิงเกิ้ลไซออน สำหรับโปรแกรมประยุกต์บนเว็บนี้ สามารถออกแบบการติดต่อได้ 2 วิธี คือ 1) ใช้ Lightweight Directory Access Protocol (LDAP) และ 2) ใช้ Web Service ซึ่งทั้ง 2 วิธีมีข้อดีข้อเสียแตกต่างกันไปขึ้นอยู่กับความถนัดในการใช้งานและองค์ประกอบของระบบอื่นๆ

1. ใช้ Lightweight Directory Access Protocol (LDAP)

เนื่องจากบริการ Directory Service ของ IBM Lotus Domino Server จัดเก็บอยู่ในรูปแบบมาตรฐาน X.500 และ Domino Server เอง สามารถเปิดการทำงานของ LDAP Service เพื่อทำให้สามารถเข้าถึง Domino Directory ผ่านทาง Lightweight Directory Access Protocol (LDAP) ได้และสามารถกำหนดสิทธิ์ในการเข้าถึงได้ด้วย ซึ่งหากมีการเปิดใช้งาน LDAP ควรเปลี่ยนพอร์ทเริ่มต้น (Default Port) จาก 389 ให้เป็น Port อื่น เช่น 1389 เป็นต้น เพื่อเพิ่มความปลอดภัยให้มากยิ่งขึ้น ในการใช้งาน LDAP บน Domino Directory นั้นจะต้องมีการปรับแต่งค่าต่างๆ เพื่อให้ระบบอื่นๆ สามารถติดต่อเข้ามาได้ตามเงื่อนไขที่ได้กำหนดไว้ สำหรับค่าเริ่มต้นในส่วนของการจับคู่เขตข้อมูลของ LDAP กับ Fields ของ Domino นั้นมีการกำหนดไว้อยู่แล้ว

2. ใช้ Web Service

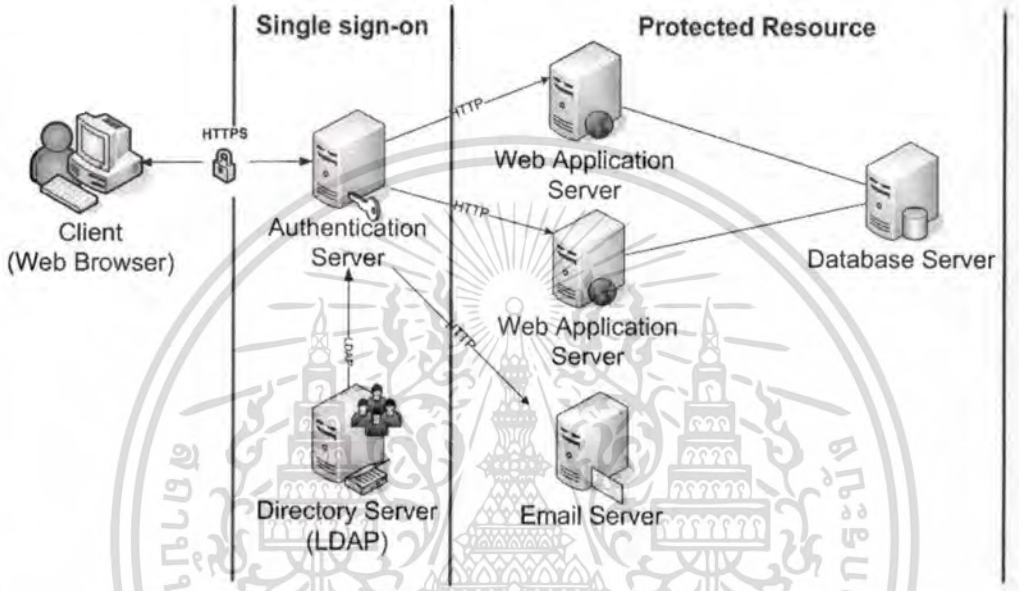
เนื่องจากในปัจจุบัน ความหลากหลายทางด้านภาษาที่ใช้ในการพัฒนาโปรแกรมประยุกต์บนเว็บ มีมากขึ้นเรื่อยๆ การนำเอา Web Service มาเป็นตัวกลางในการเรียกใช้ส่วนหนึ่งของโปรแกรมหรือโมดูล จากโมดูลที่ถูกพัฒนาด้วยอีกภาษาหนึ่ง ทำให้การพัฒนาระบบด้วย Web Service เป็นการเพิ่มขีดความสามารถโดยไม่ขึ้นกับภาษาได้ การใช้ Web service กับระบบซิงเกิ้ลไซออน จะเป็นการสร้างบริการที่เป็นช่องทางในการติดต่อกับ Directory Service ในด้านของการเรียกดู หรือสืบค้นข้อมูล และการแก้ไขหรือปรับปรุงข้อมูล ซึ่ง IBM Lotus Domino Server สามารถสร้าง Web Service ในลักษณะของ Provider นี้เพื่อให้บริการดังกล่าวได้

3.2.1.5 การเปลี่ยนแปลงข้อมูลในบัญชีชื่อผู้ใช้งาน

สำหรับการเปลี่ยนแปลงข้อมูลบัญชีผู้ใช้งานนั้นรูปแบบและขั้นตอน จะขึ้นอยู่กับวิธีที่ใช้ในการติดต่อเข้ามายังบริการบัญชีรายชื่อ ทั้ง 2 วิธีคือ LDAP และ Web Service ซึ่งแต่ละวิธีที่ใช้ในการติดต่อก็มีขั้นตอนที่แตกต่างกันด้วย

3.2.2 บริการพิสูจน์ตัวจริง (Authentication Service)

การพิสูจน์ตัวจริงนั้นจะขึ้นอยู่กับกลไกการทำงานของ Web Server ที่นำมาให้บริการนั่นก็คือ IBM Lotus Domino Server โดยกลไกของ Web Server นี้จะมีลักษณะเป็นแบบ Basic HTTP Authentication โดยระบบที่จะเข้าร่วมบริการซิงเกิ้ลไซออนนั้น จะต้องใช้บริการพิสูจน์ตัวจริงจากเครื่องให้บริการดังกล่าวภายใต้ Directory เดียวกันทุกในทุกระบบที่เข้าร่วม โดยเครื่องให้บริการดังกล่าวจะทำหน้าเป็น Authentication Server โดยแสดงได้ดังรูปที่ 3.8

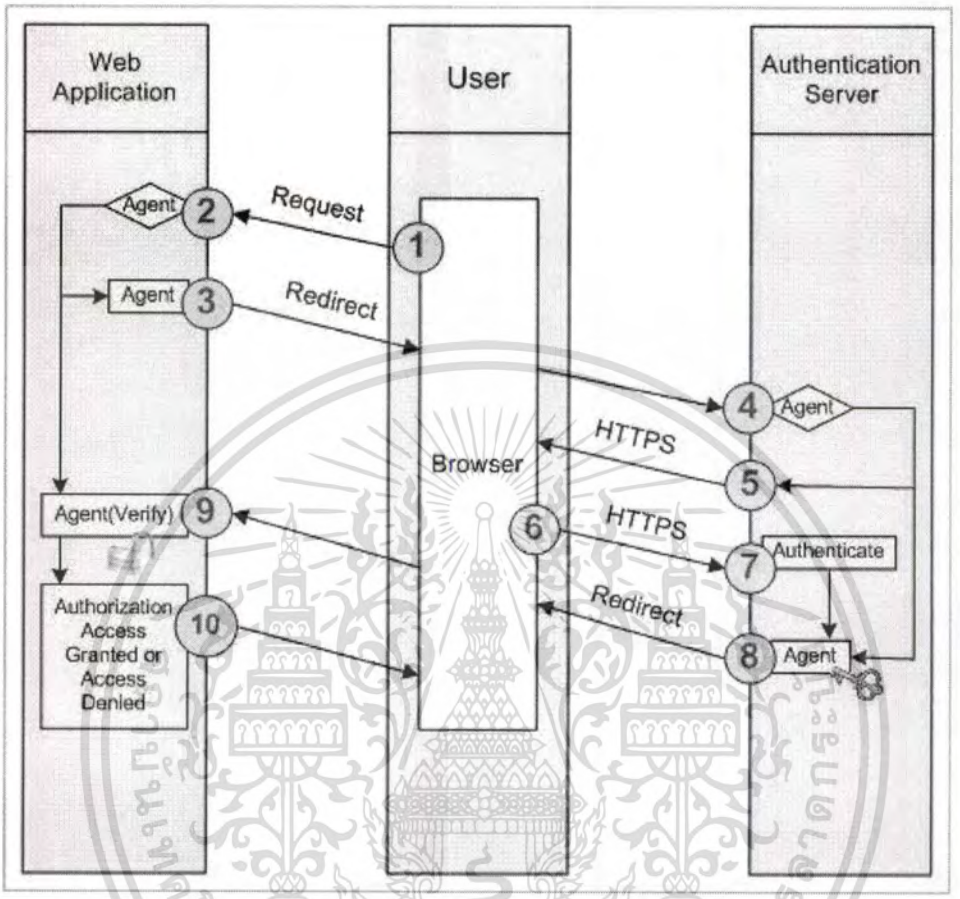


รูปที่ 3.8 แสดงการพิสูจน์ตัวจริงในระบบซิงเกิ้ลไซออน

สำหรับ บริการพิสูจน์ตัวจริงนี้ถือเป็นหัวใจหลักของบริการของระบบซิงเกิ้ลไซออน การออกแบบนอกจากจะต้องคำนึงถึงความถูกต้องแล้ว ยังต้องคำนึงถึงความปลอดภัยอีกด้วย โดยแนวคิดในการทำซิงเกิ้ลไซออนในรูปแบบนี้ จะทำให้ Authentication อยู่ที่ศูนย์กลาง โดยตั้งเป็น Authentication Server และให้ Web Application ต่างๆ ที่ต้องการเข้าร่วมบริการทำการ Redirect Request มาพิสูจน์ตัวจริงที่เดียว หลังจากที่ได้มีการพิสูจน์ตัวจริงโดย Authentication Server แล้วนั้นจึงจะ Redirect Request ดังกล่าวกลับไปยัง Page เดิมที่ผู้ใช้งานทำการ Request มา พร้อมกับมีการส่ง Signature กลับไปด้วย เพื่อเป็นการรับรองและยืนยันว่ามาจากแหล่งที่ถูกต้อง หลังจาก Web Application ดังกล่าวได้รับ Request กลับมาพร้อม Signature จะต้องมีการ Verify Signature นั้น เพื่อตรวจสอบ Request ถูกต้องหรือเชื่อถือได้หรือไม่

3.2.2.1 กลไกในการพิสูจน์ตัวจริง

กลไกในการพิสูจน์ตัวจริงนั้นสามารถแสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 แสดงขั้นตอนและลำดับในการ Authentication

ขั้นตอนต่างๆ เมื่อผู้ใช้งานต้องการเข้าใช้งานโปรแกรมประยุกต์บนเว็บ ที่มีการเข้าร่วมบริการซิงเกิ้ลไซออนเป็นดังนี้

1. ผู้ใช้งานทำการร้องขอการเข้าใช้งานระบบ ไปยัง Web Application
2. Web Application ทำการตรวจสอบว่าผู้ใช้งานได้ผ่านการพิสูจน์ตัวจริงแล้วหรือยัง ถ้าหากผ่านการพิสูจน์ตัวจริงแล้วจะไปทำงานที่ขั้นตอนที่ 9) แต่ถ้าหากยังไม่ผ่านการพิสูจน์ตัวจริงจะทำงานในขั้นตอนที่ 3) ต่อไป
3. Web Application ตรวจสอบ พบว่ายังไม่ได้ผ่านการพิสูจน์ตัวจริง จึง Redirectผู้ใช้งานไปยัง Authentication Server เพื่อทำการตรวจพิสูจน์ตัวจริง
4. Authentication Server ทำการตรวจสอบว่าผู้ใช้งานได้ผ่านการพิสูจน์ตัวจริงแล้วหรือยัง ถ้าหากผ่านการพิสูจน์ตัวจริงแล้วจะไปทำงานในขั้นตอนที่ 8) แต่ถ้าหากยังไม่ผ่านการพิสูจน์ตัวจริงจะทำงานในขั้นตอนที่ 5) ต่อไป
5. Authentication Server ตรวจสอบ พบว่ายังไม่ได้ผ่านการพิสูจน์ตัวจริง จึงส่งหน้าเว็บเพื่อใช้ในการ Authenticate ให้ผู้ใช้งานใส่ Username และ Password
6. ผู้ใช้งาน ป้อน Username และ Password ส่งไปยัง Authentication Server เพื่อตรวจพิสูจน์ตัวจริง
7. Authentication Server ทำการตรวจพิสูจน์ตัวจริงกับ Directory ในระบบ หากไม่ผ่านจะกระทำซ้ำในขั้นตอนที่ 5) แต่ถ้าหากพิสูจน์ตัวจริงผ่านแล้วจึงดำเนินการในขั้นตอนที่ 8) ต่อไป
8. Authentication Server ทำการ Sign ข้อมูล เพื่อสร้างเป็น Signature และทำการ Redirect ผู้ใช้งานกลับไปยัง Web Application ที่ได้มีการร้องขอมาตั้งแต่ต้น พร้อมกับส่ง Signature ไปด้วย
9. Web Application ทำการ Verify Signature ที่ส่งมาจาก Authentication Server
10. Web Application ตรวจสอบสิทธิ์ในการเข้าถึง (Authorization) ของผู้ใช้งานว่ามีสิทธิ์เข้าใช้งานหรือไม่

หากมีร้องขอเพื่อเข้าใช้งานไปยังอีก Web Application หนึ่ง ก็จะทำการตามขั้นตอนดังกล่าวข้างต้น เช่นกัน

3.2.2.2 การรักษาความปลอดภัยในการพิสูจน์ตัวจริง

การรักษาความปลอดภัยในการพิสูจน์ตัวจริงนั้นจะให้ความสำคัญในการรับ-ส่งข้อมูลที่เป็น Username และ Password โดยจะใช้การรับ-ส่งข้อมูลผ่าน Protocol HTTPS เพื่อให้เกิดความปลอดภัยมากยิ่งขึ้น นอกจากนั้นจะกำหนดนโยบายในการพิสูจน์ตัวจริงอีกด้วย เช่น

- การระงับบัญชีชื่อผู้ใช้งานชั่วคราวหากมีการพยายามลงชื่อเข้าใช้งานแต่ไม่สำเร็จภายใน 3 ครั้งติดต่อกัน
- การกำหนดความยาวขั้นต่ำของรหัสผ่านให้มีความยาว 8 ตัวอักษร
- กำหนดให้มีการเปลี่ยนรหัสผ่านทุกๆ 60 วัน เป็นต้น

3.2.3 บริการตรวจสอบและดูแลการทำงาน (Monitoring Service)

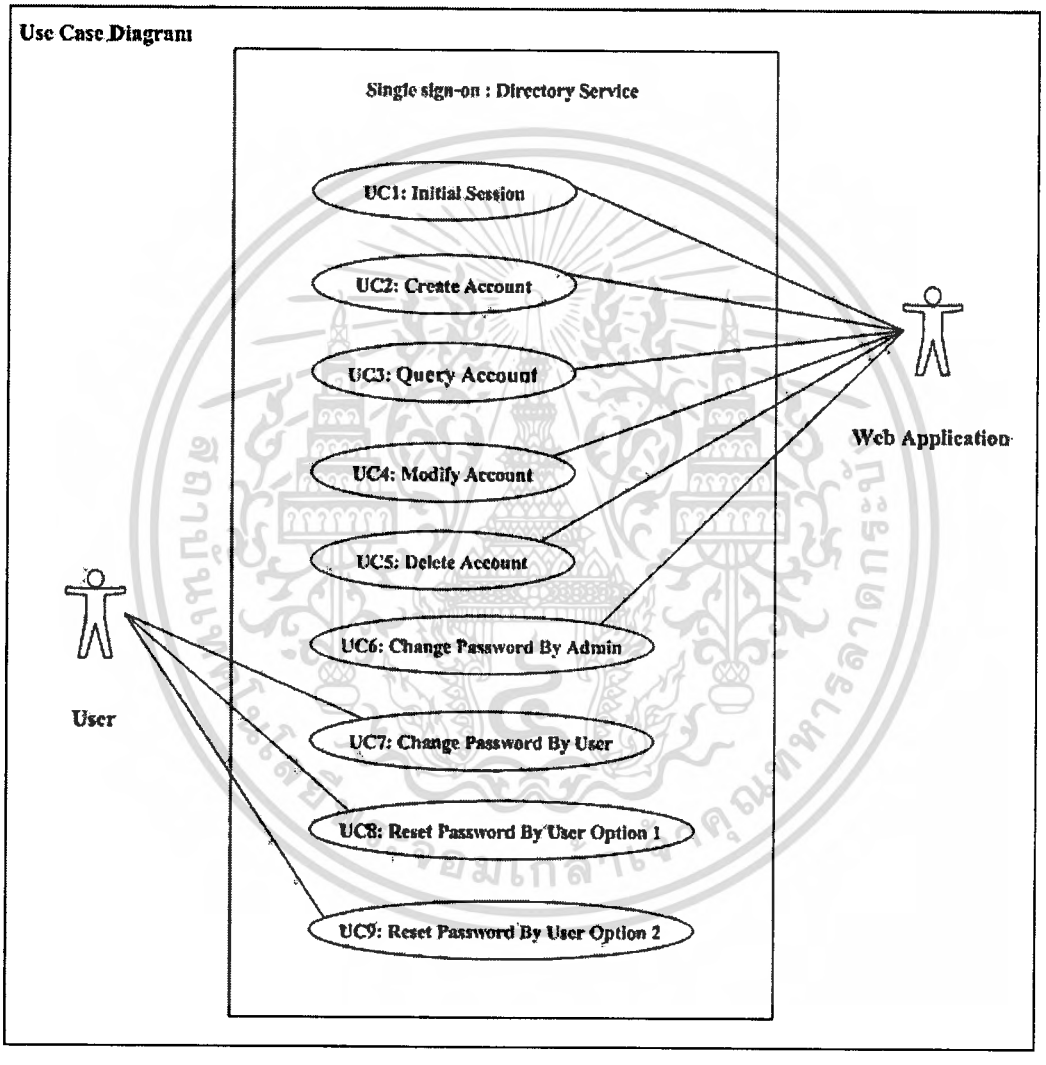
เป็นบริการสำหรับการตรวจสอบการเข้าใช้งานระบบ โดยระบบจะเก็บบันทึกข้อมูลการเข้าใช้งานต่างๆ เช่น ชื่อผู้ใช้งาน วันเวลา หมายเลขไอพี เป็นต้น ซึ่งข้อมูลเหล่านี้จะสามารถนำมาใช้ในการติดตาม ย้อนรอย เมื่อเกิดปัญหาขึ้นได้ โดยการเข้าถึงข้อมูลเหล่านี้จะต้องเป็นผู้ที่มีสิทธิ์เท่านั้น สำหรับ IBM Lotus Domino Server สามารถทำการเปิดให้ใช้งาน การเก็บบันทึกข้อมูลการใช้งาน (Web Server Log) ดังกล่าวได้ และยังมีเครื่องมือเพื่อใช้ในการดูข้อมูลในมุมมองต่างๆ ได้อีกด้วย

รูปที่ 3.10 แสดงเครื่องมือที่ใช้ในการดู Web Server Log ในมุมมองต่างๆ

3.3 แผนภาพแสดง Use Case

ในหัวข้อนี้จะเป็นการแสดงแผนภาพการทำงานของระบบ ที่ได้พัฒนาขึ้น และบุคคลที่เกี่ยวของในการทำงานของระบบทั้งหมด โดยจะมี 2 ส่วนที่สำคัญคือ Directory Service และ Authentication Service

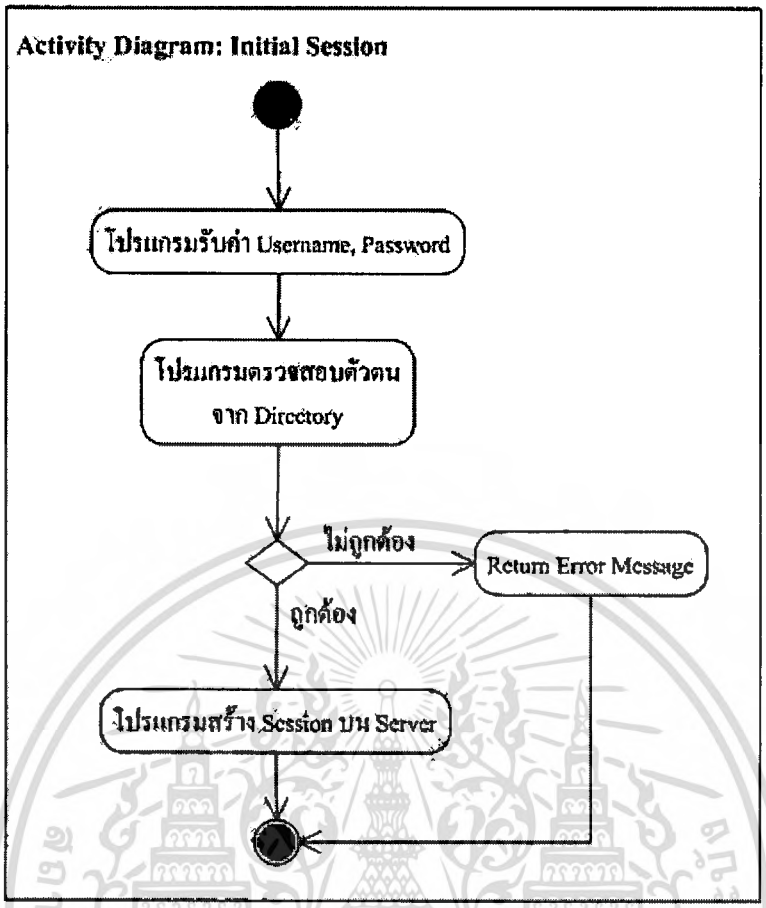
3.3.1 แผนภาพแสดง Use Case ของ Directory Service



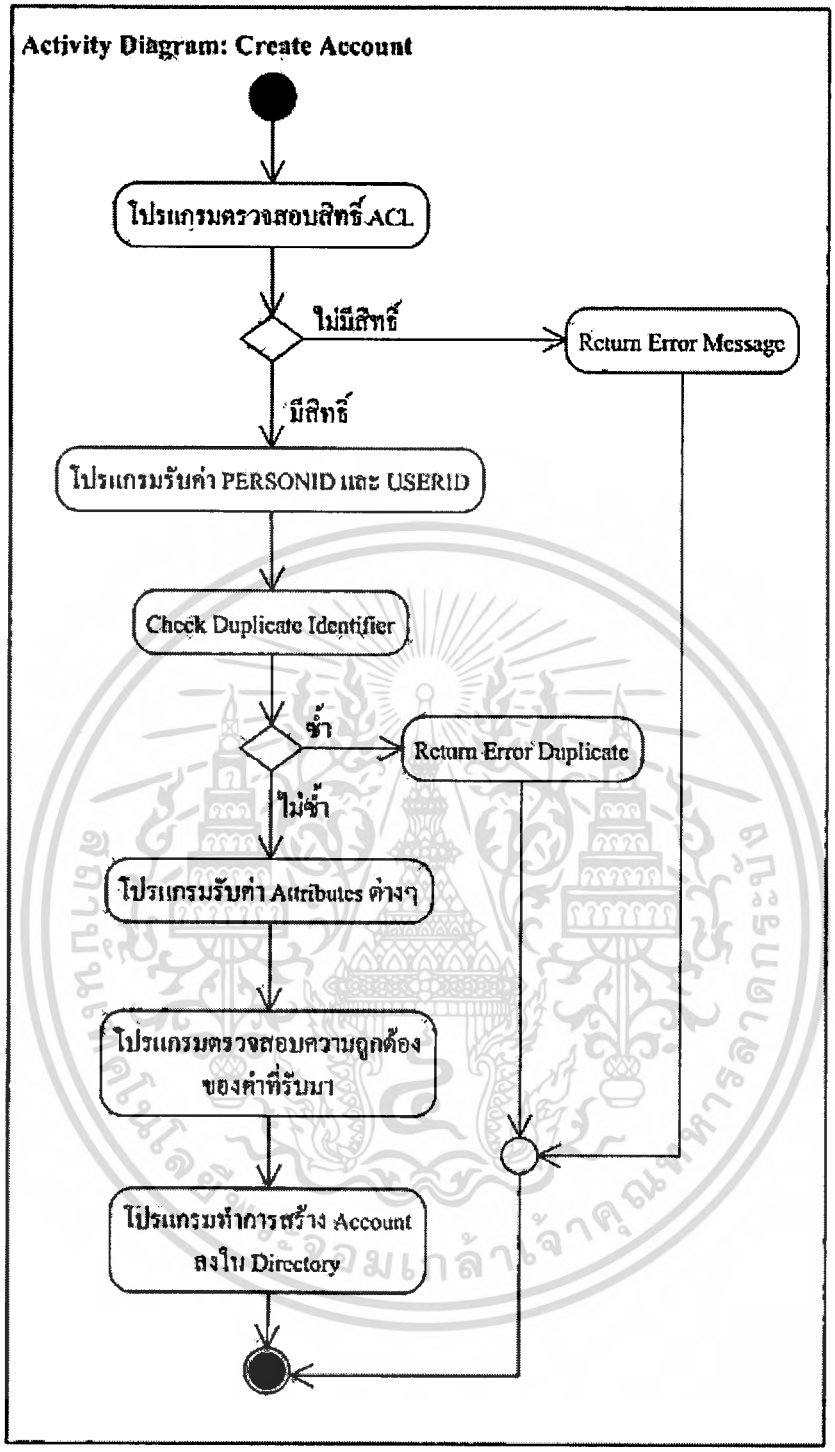
รูปที่ 3.11 แสดง Use Case ของระบบส่วน Directory Service

1. Actor แสดงถึงผู้ที่เกี่ยวข้องกับระบบ ทั้งบุคคลและ Application
   - User ผู้ใช้งานที่ต้องการใช้งาน Web Application และใช้งาน Directory Service
   - Web Application ที่ทำการติดต่อกับ Directory Service
2. Process แสดงถึงกระบวนการทำงานต่างๆ ของ Service
   - UC1: Initial Session เป็นกระบวนการที่ Web Application ทำการร้องขอ Session การเข้าทำงานไปยัง Directory Service
   - UC2: Create Account เป็นกระบวนการสร้างบัญชีผู้ใช้งานที่ Directory
   - UC3: Query Account เป็นกระบวนการสืบค้นข้อมูลบัญชีผู้ใช้งาน
   - UC4: Modify Account เป็นกระบวนการแก้ไขบัญชีผู้ใช้งาน
   - UC5: Delete Account เป็นกระบวนการลบบัญชีผู้ใช้งาน
   - UC6: Change Password By Admin เป็นกระบวนการเปลี่ยนรหัสผ่านของบัญชีผู้ใช้งานโดยผู้ดูแลระบบ
   - UC7: Change Password By User เป็นกระบวนการเปลี่ยนรหัสผ่านของบัญชีผู้ใช้งานโดยผู้ใช้งาน
   - UC8: Reset Password By User Option 1 เป็นกระบวนการตั้งค่ารหัสผ่านใหม่ทางเลือกที่ 1 ใช้ UserID
   - UC9: Reset Password By User Option 2 เป็นกระบวนการตั้งค่ารหัสผ่านใหม่ทางเลือกที่ 2 ใช้ Email ของผู้ใช้งาน

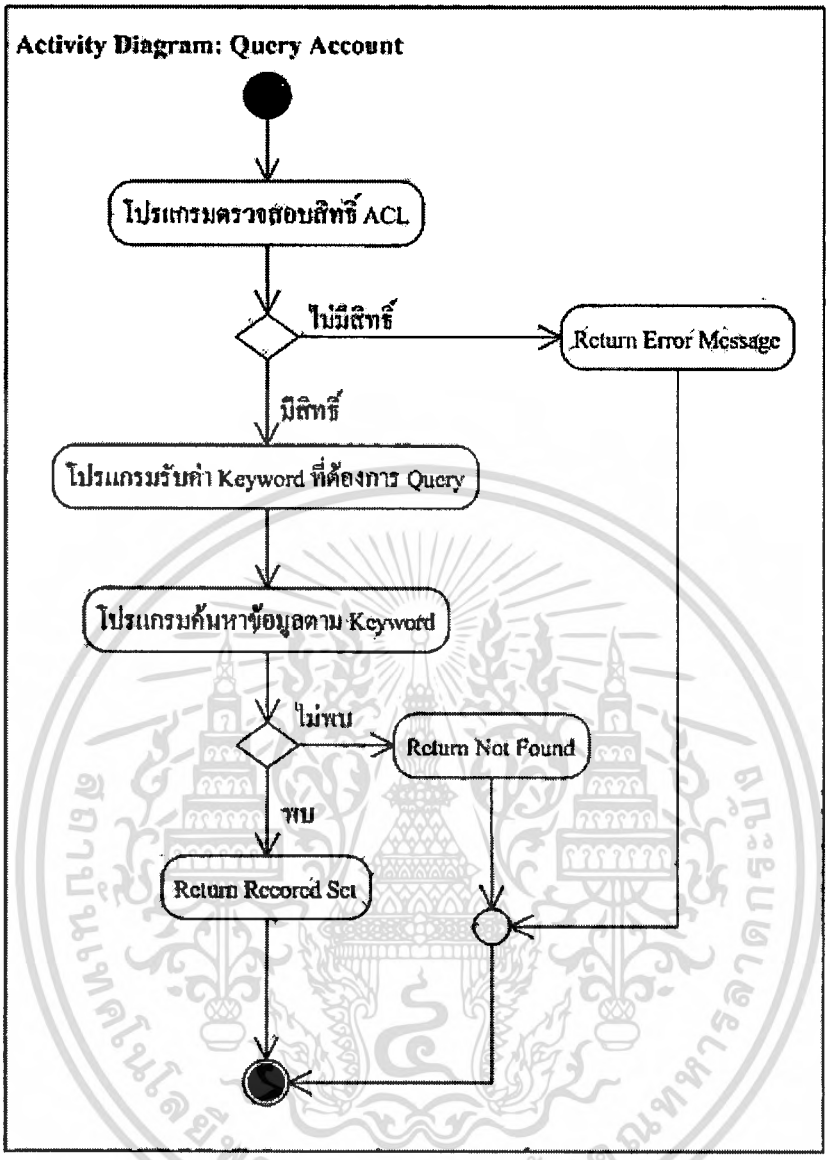
โดยการทำงาน UC1 – UC6 จะเรียกใช้งานโดย Web Application ผ่าน Web Service ส่วน UC7-UC9 เป็นการเรียกใช้งานโดยตรงที่ Directory Service จาก User



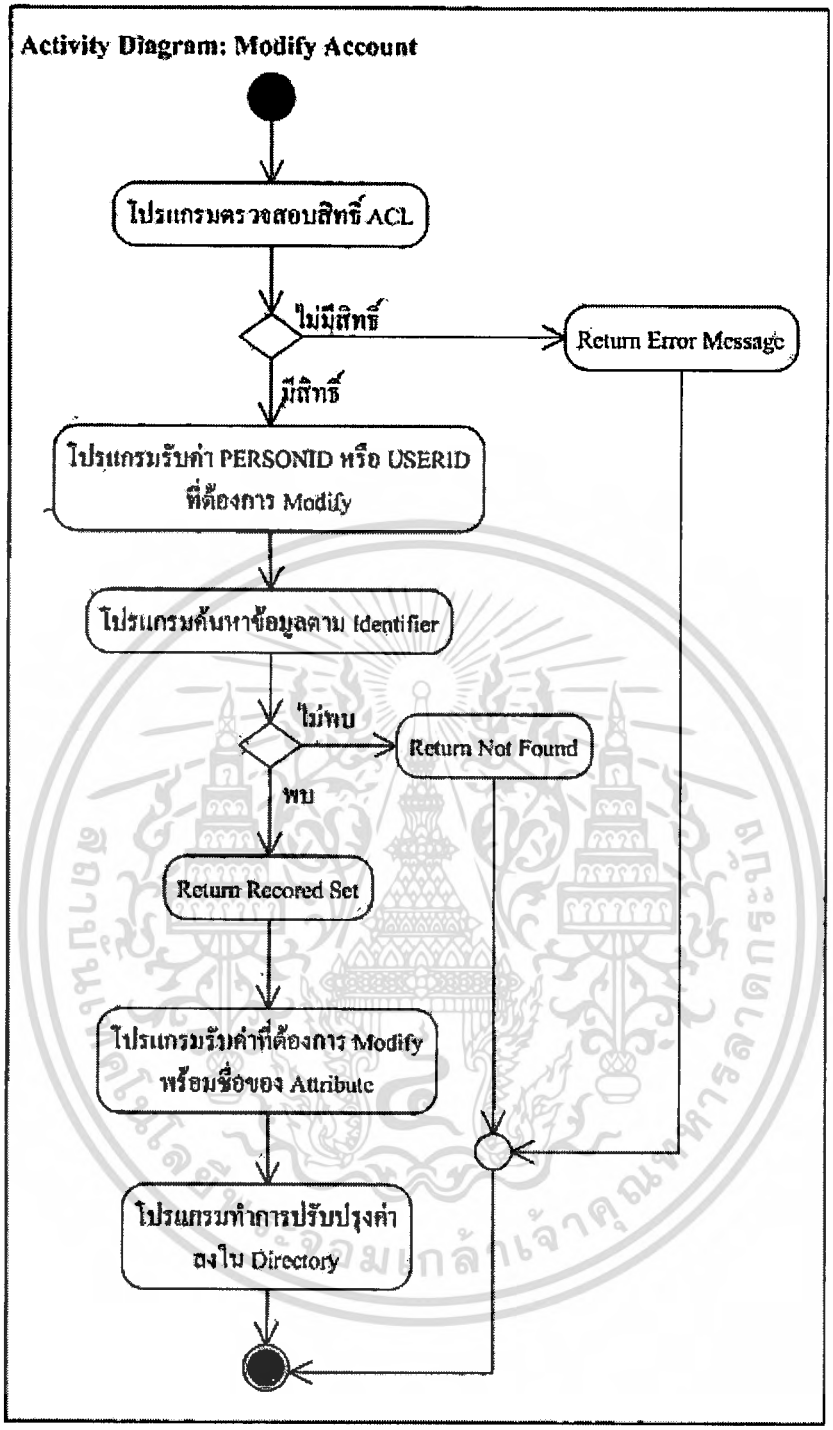
รูปที่ 3.12 แผนภาพแสดง Activity Diagram Initial Session



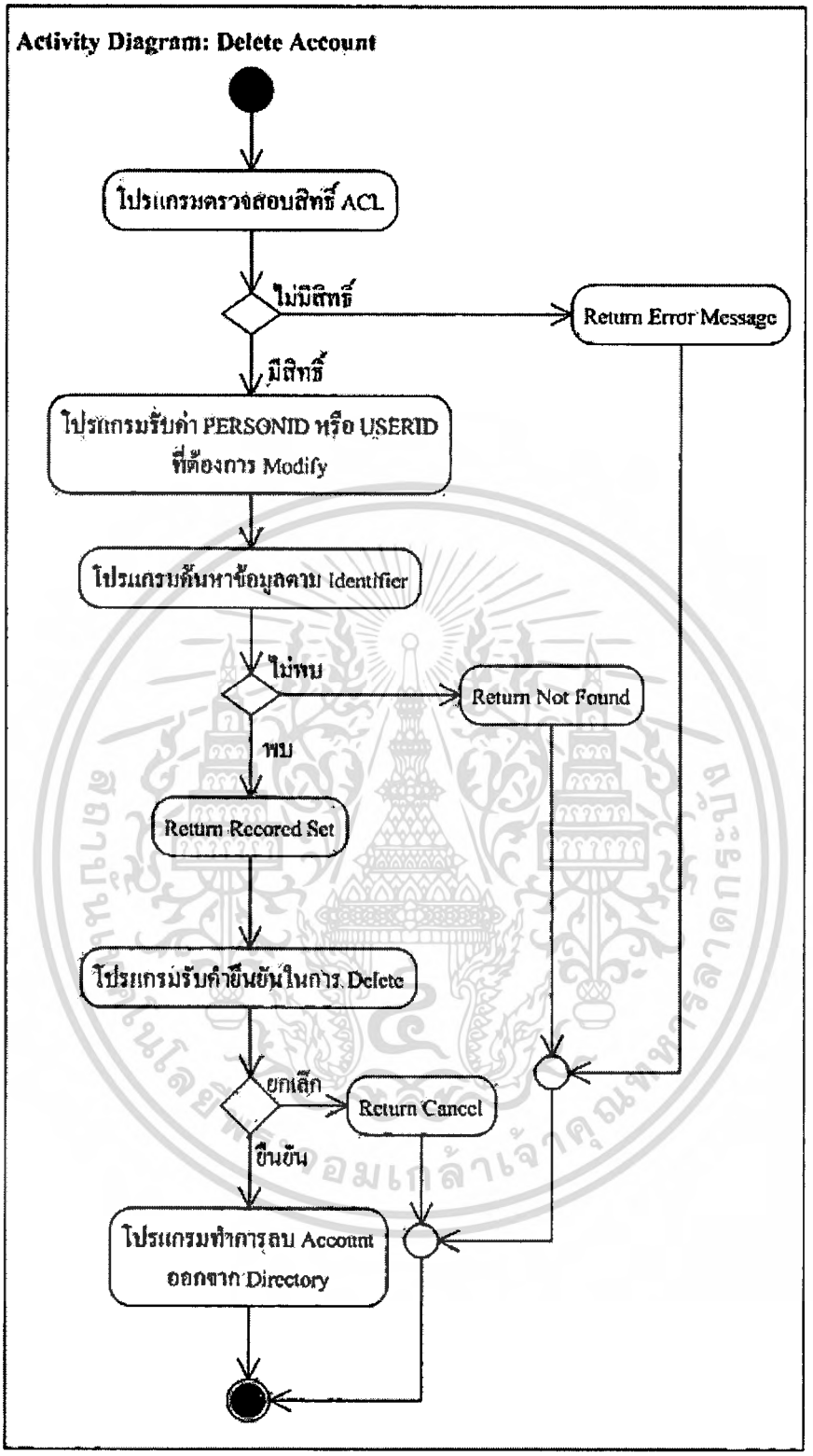
รูปที่ 3.13 แผนภาพแสดง Activity Diagram Create Account



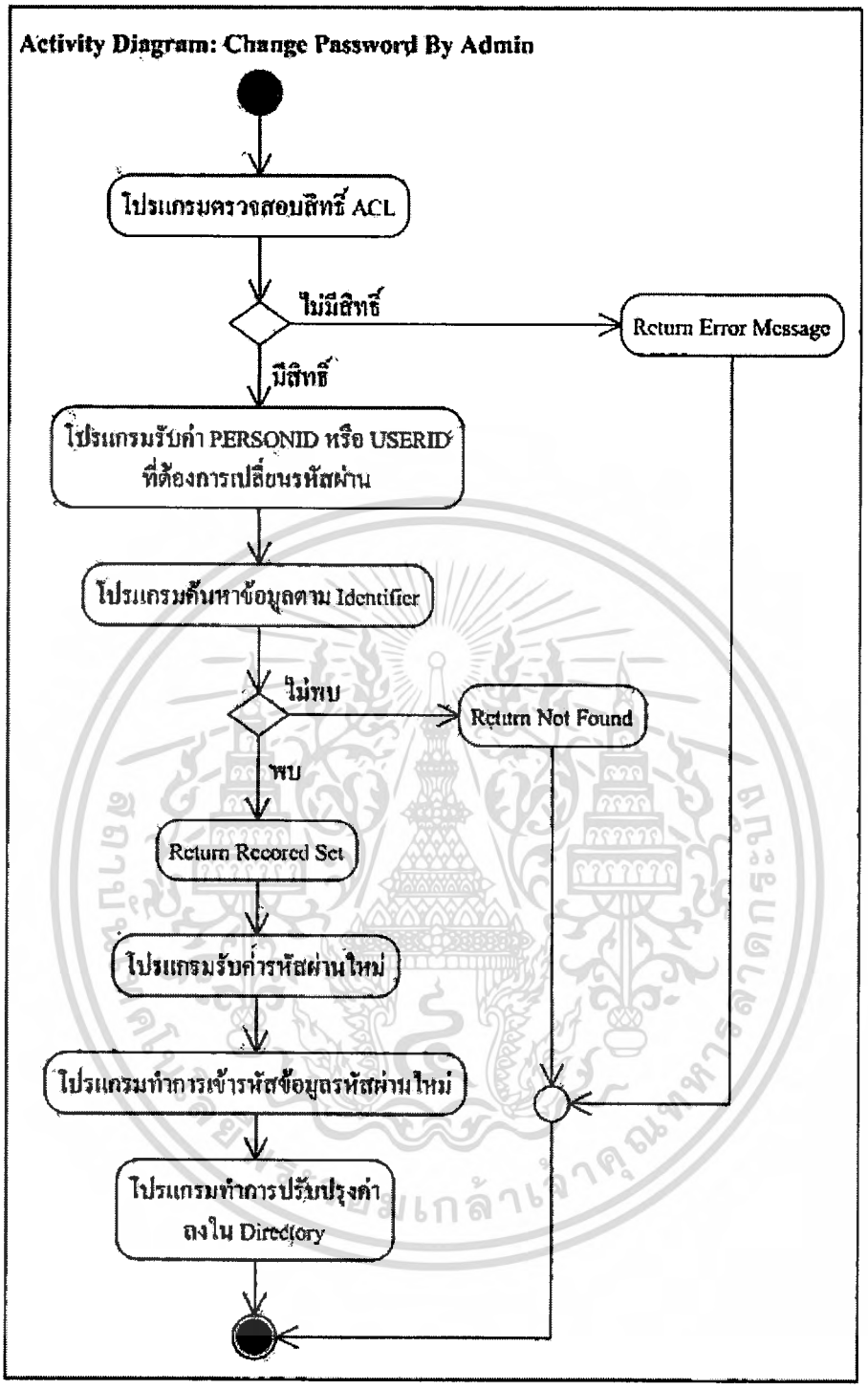
รูปที่ 3.14 แผนภาพแสดง Activity Diagram Query Account



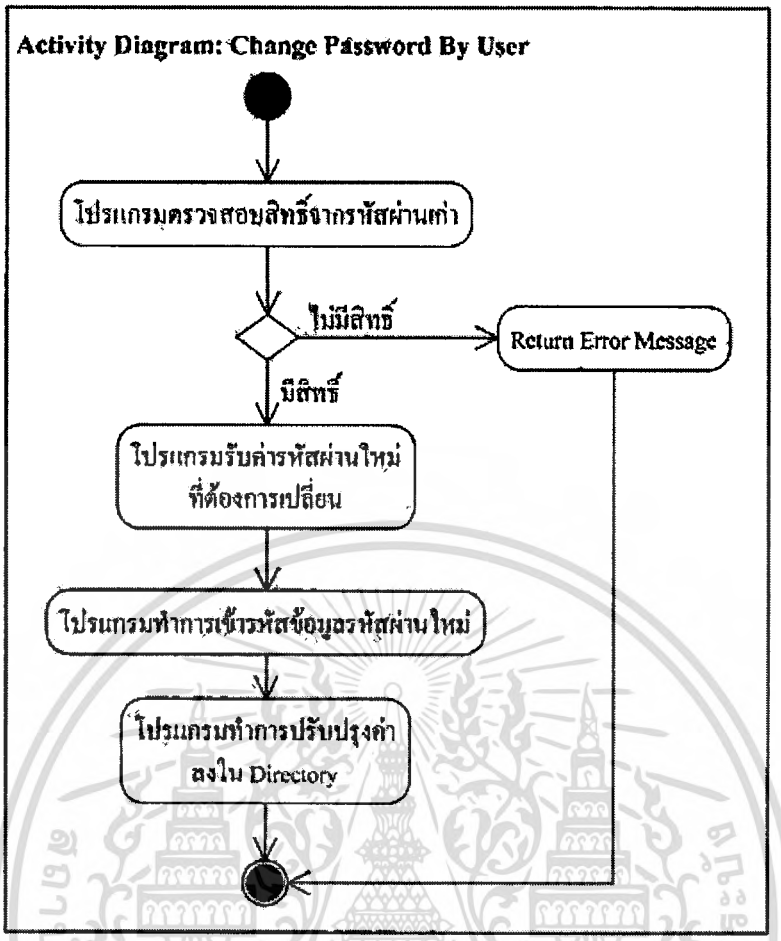
รูปที่ 3.15 แผนภาพแสดง Activity Diagram Modify Account



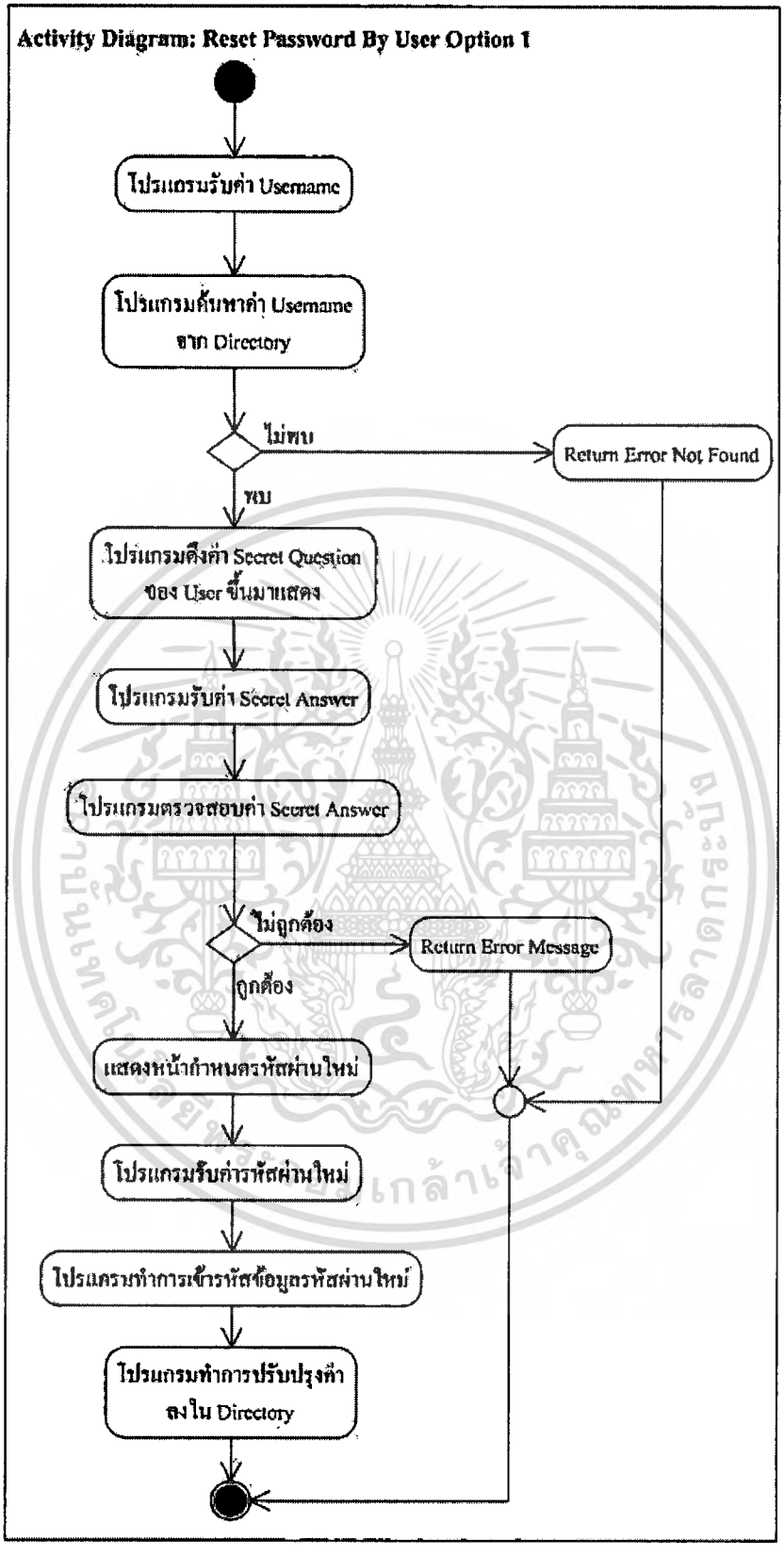
รูปที่ 3.16 แผนภาพแสดง Activity Diagram Delete Account



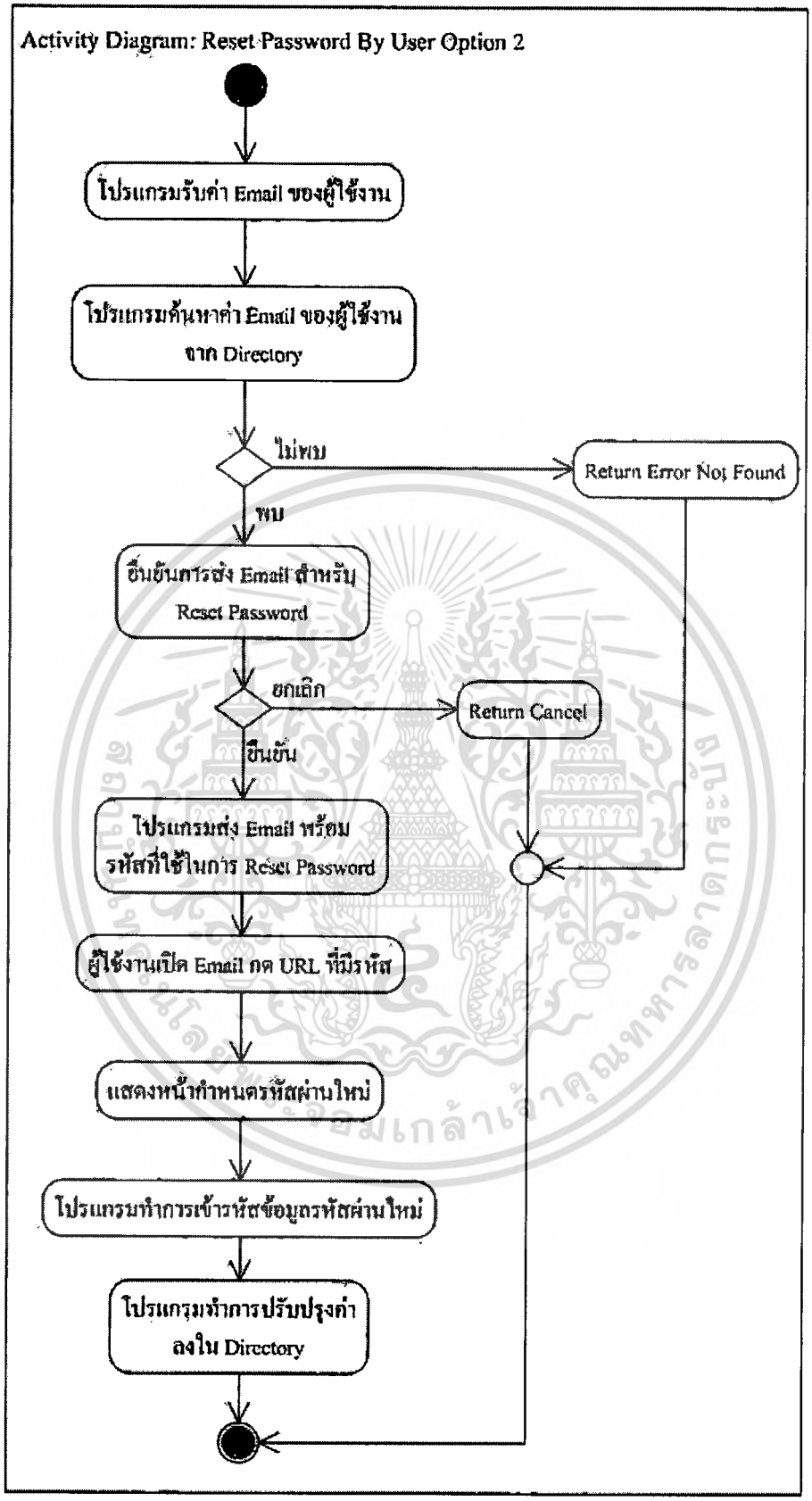
รูปที่ 3.17 แผนภาพแสดง Activity Diagram Change Password By Admin



รูปที่ 3.18 แผนภาพแสดง Activity Diagram Change Password By User

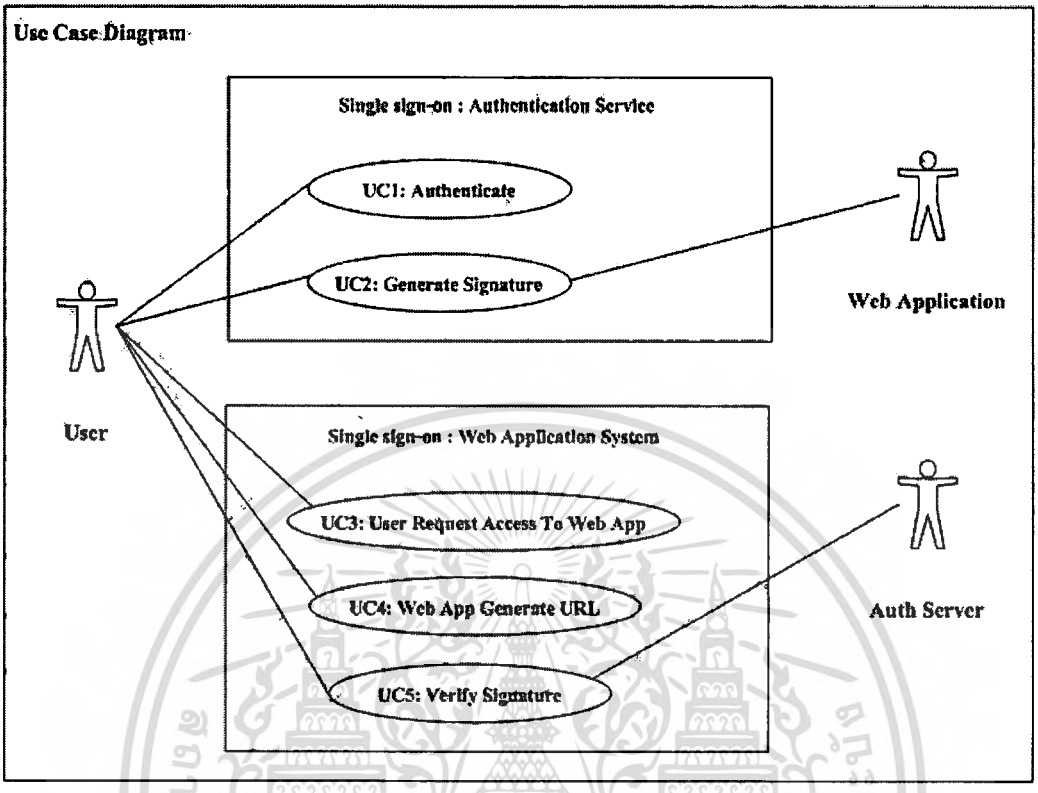


รูปที่ 3.19 แผนภาพแสดง Activity Diagram Reset Password By User Option 1



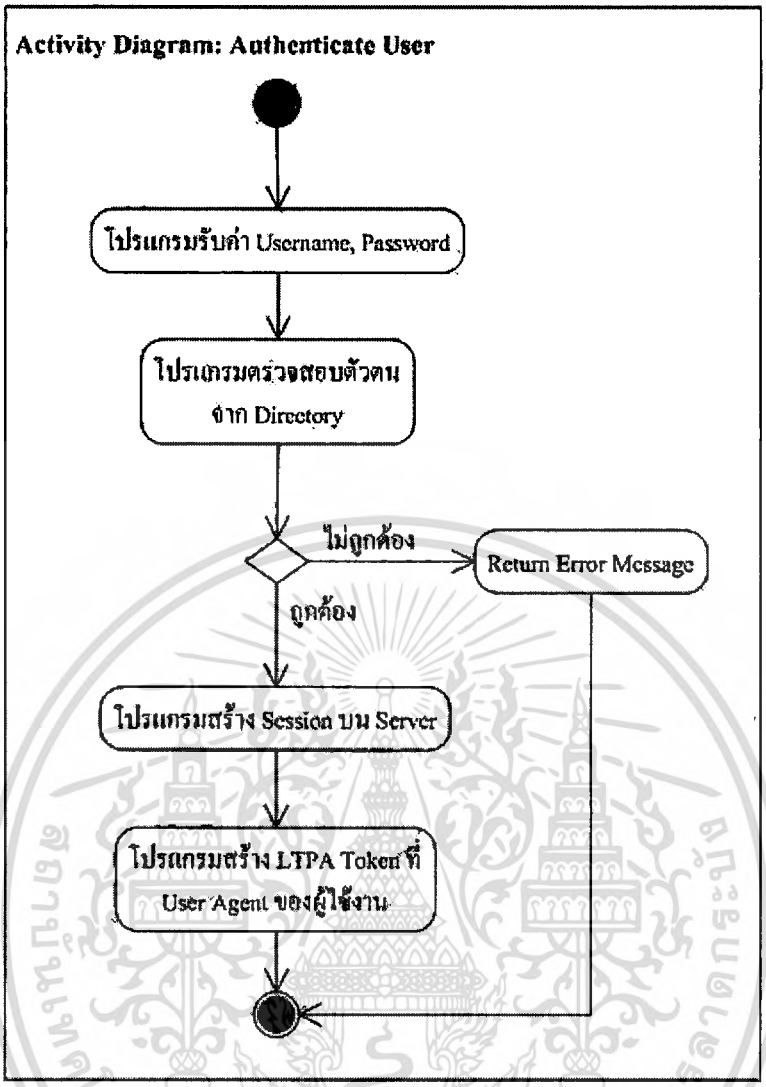
รูปที่ 3.20 แผนภาพแสดง Activity Diagram Reset Password By User Option 2

3.3.2 แผนภาพแสดง Use Case ของ Authentication Service

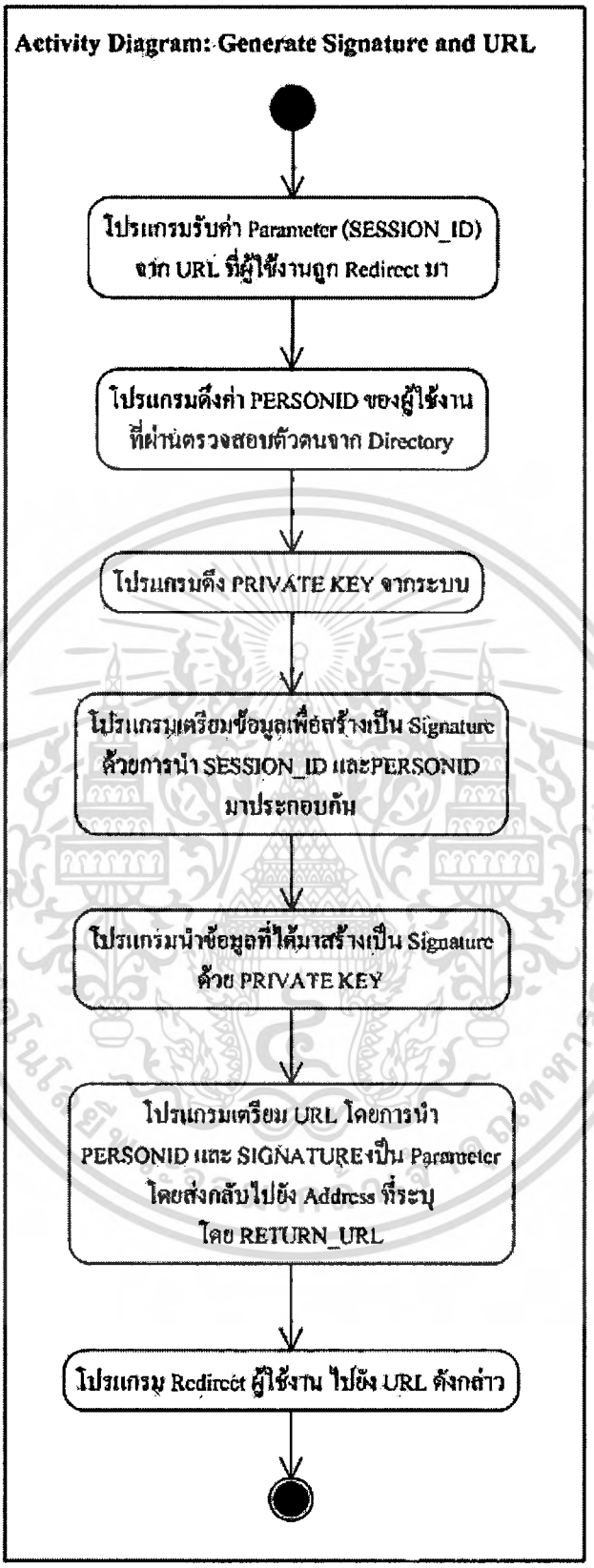


รูปที่ 3.21 แสดง Use Case ของระบบส่วน Authentication Service

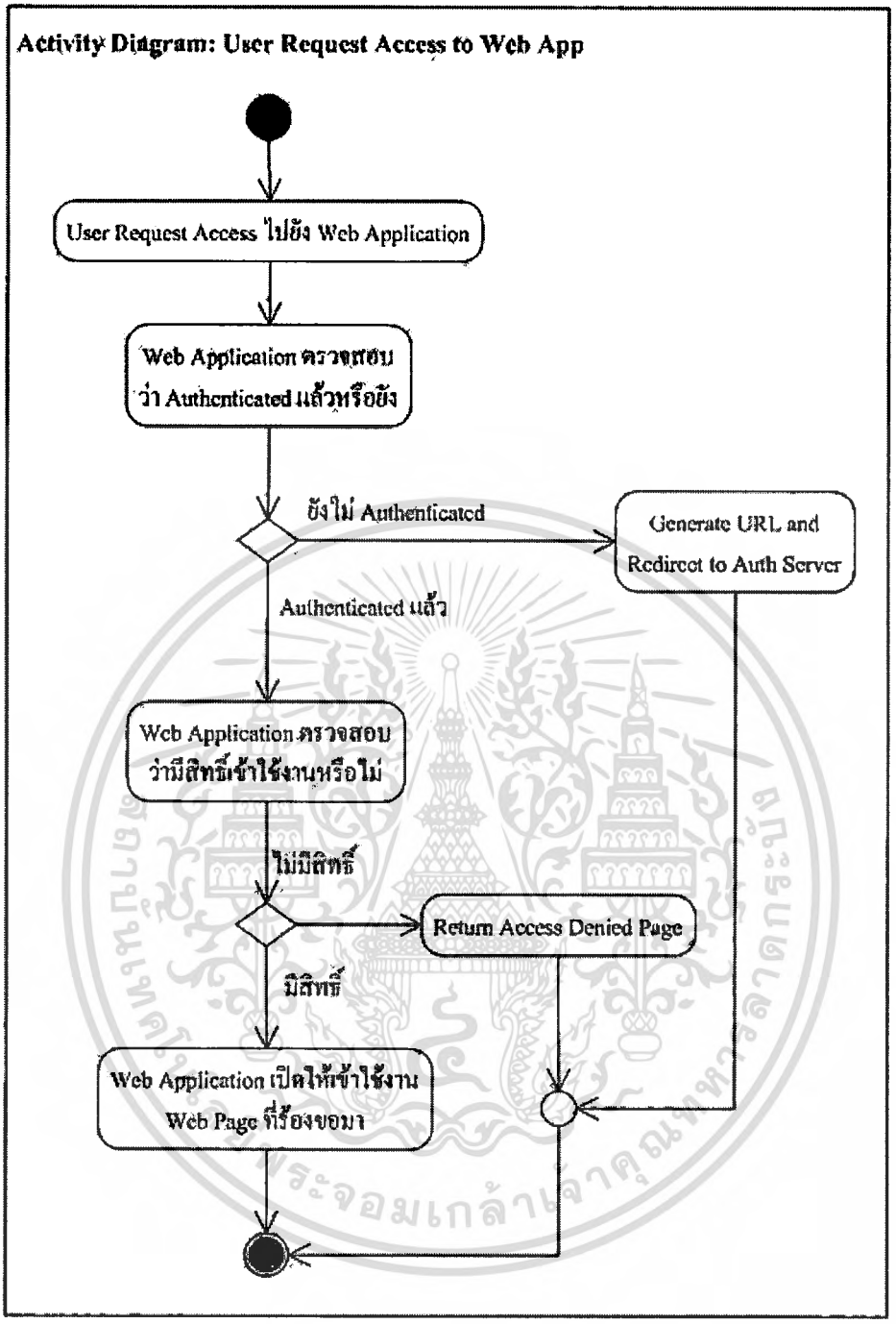
1. Actor แสดงถึงผู้ที่เกี่ยวข้องกับระบบ ทั้งบุคคลและ Application
   - User ผู้ใช้งานที่ต้องการใช้งาน Web Application และใช้งาน Authentication Service
   - Web Application ที่ทำการติดต่อกับ Authentication Service
   - Authentication Server ที่ติดต่อกับ Web Application ที่ผู้ใช้งานร้องขอเข้าใช้
2. Process แสดงถึงกระบวนการทำงานต่างๆ ของ Service
   - UC1: Authenticate เป็นกระบวนการที่ User ทำการร้องขอการพิสูจน์ตัวจริงกับ Authentication Service
   - UC2: Generate Signature เป็นกระบวนการสร้างลายมือชื่อดิจิตอล
   - UC3: User Request Access To Web App เป็นกระบวนการที่ User ร้องขอเพื่อเข้าใช้งาน Web Application
   - UC4: Web App Generate URL เป็นกระบวนการสร้าง URL เพื่อ Redirect User ไปร้องการพิสูจน์ตัวจริงจาก Authentication Server
   - UC5: Verify Signature เป็นกระบวนการตรวจสอบลายมือชื่อดิจิตอล



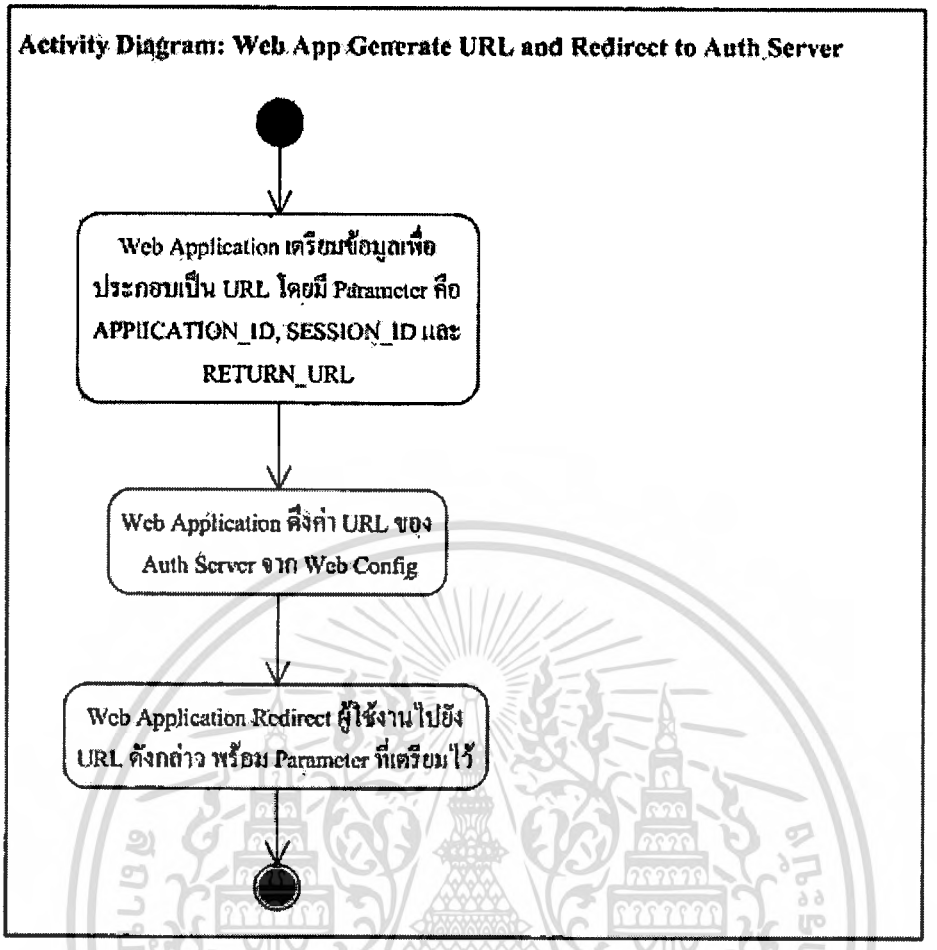
รูปที่ 3.22 แผนภาพแสดง Activity Diagram Authenticate



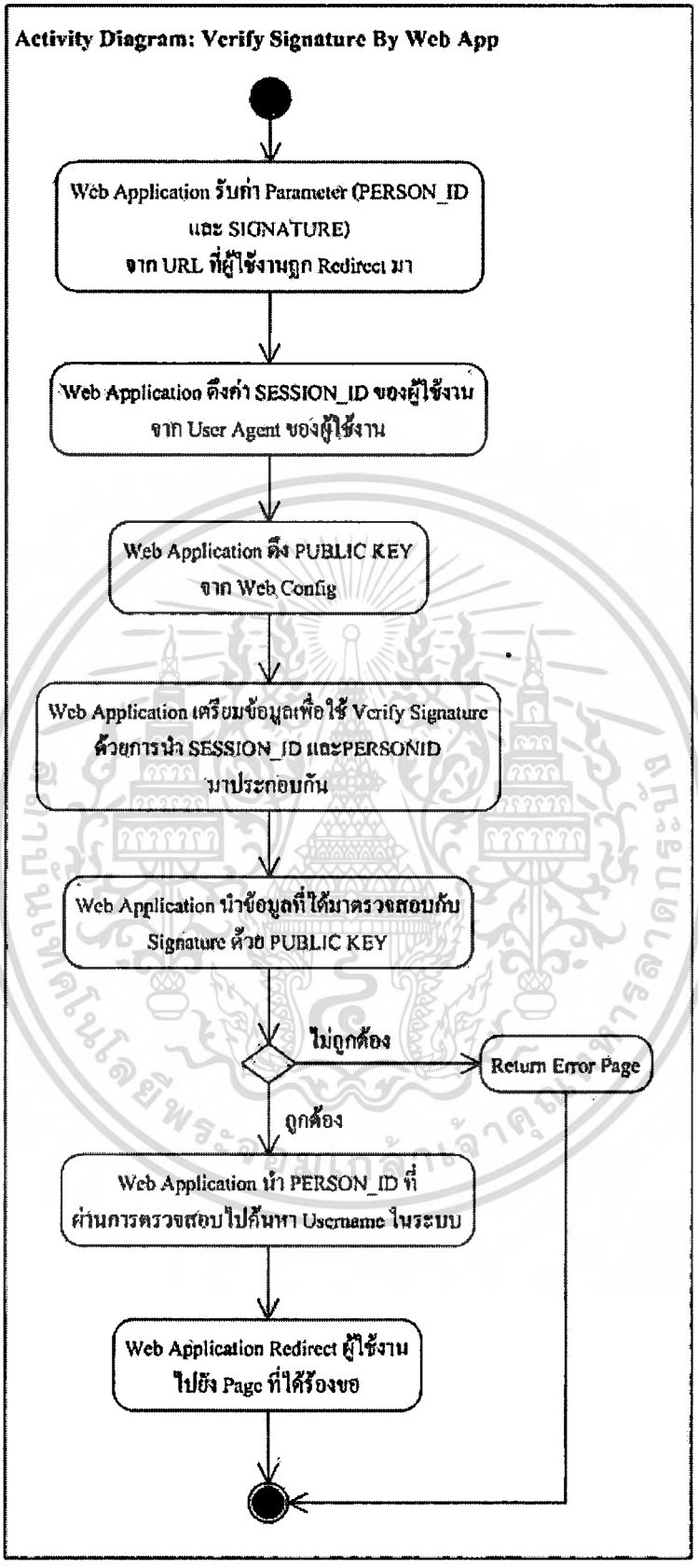
รูปที่ 3.23 แผนภาพแสดง Activity Diagram Generate Signature



รูปที่ 3.24 แผนภาพแสดง Activity Diagram User Request Access To Web App



รูปที่ 3.25 แผนภาพแสดง Activity Diagram Web App Generate URL



รูปที่ 3.26 แผนภาพแสดง Activity Diagram Verify Signature

3.4 คุณสมบัติของระบบ

จากที่ได้วิเคราะห์และออกแบบระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บเพื่อช่วยในเรื่องของการพิสูจน์ตัวจริงและการดูแลบัญชีรายชื่อผู้ใช้งานไปแล้วนั้น ยังมีคุณสมบัติอื่นของระบบดังนี้

3.4.1 การอนุญาตให้สิทธิ์ทำในระดับโปรแกรมประยุกต์ (Authorization on the Application Level)

การอนุญาต ให้สิทธิ์ (Authorization) ให้สามารถใช้งานระบบสำหรับผู้ใช้งานที่ผ่านการพิสูจน์ตัวจริงแล้วนั้น จะกระทำแยกกันในแต่ละระบบเนื่องจากสิทธิ์ในการเข้าถึงข้อมูลอาจไม่เท่ากัน Authentication Server จะทำหน้าที่เพียงแค่ให้บริการพิสูจน์ตัวจริง และให้ข้อมูลกับระบบหลังจากผ่านการพิสูจน์ตัวจริงแล้วเท่านั้น จากนั้นผู้ใช้งานจะติดต่อกับระบบที่ร้องขอข้อมูล ตามสิทธิ์ ของระบบนั้นๆ ต่อไป

3.4.2 การทำงานร่วมกันแบบข้ามโดเมน (Cross Domain)

คุณสมบัติอย่างหนึ่ง ที่จะทำให้ระบบสามารถขยายขอบเขตการบริการได้เพิ่มมากขึ้น นั่นก็คือความสามารถในการทำงานร่วมกันได้แบบข้ามโดเมน เช่น Web1.abc.org สามารถใช้งานระบบซิงเกิ้ลไซออน ร่วมกับ Web2.xyz.org ได้ เป็นต้น

3.4.3 บริการศูนย์กลางบัญชีรายชื่อ (Centralized Directory Service / Single User)

เนื่องจากในองค์กรมีข้อมูลที่สำคัญหลายอย่างดังนั้น คุณสมบัติที่สำคัญอีกอย่างหนึ่งของระบบซิงเกิ้ลไซออนที่จะนำเสนอนี้คือ การรวมบัญชีรายชื่อของทุกระบบเข้าด้วยกันให้มีแค่เพียงศูนย์กลางเท่านั้น และทำการรวมบัญชีรายชื่อที่ซ้ำซ้อนรวมเข้าด้วยกัน โดยลักษณะดังกล่าวระบบส่วนกลางจะเปิดเป็นบริการบัญชีรายชื่อ (Directory Service) เพื่อให้บริการแก่ระบบอื่นๆ สามารถทำการเพิ่ม ลด หรือปรับเปลี่ยนข้อมูลในส่วนกลางได้ ตามสิทธิ์การอนุญาตและเงื่อนไขของระบบซิงเกิ้ลไซออนที่ตั้งไว้ ในส่วนนี้ถือเป็นการทำ Authorization ให้กับ Directory Service นั่นเอง

3.4.4 บริการศูนย์กลางการพิสูจน์ตัวจริง (Centralized Authentication)

เพื่อให้สามารถตรวจสอบการเข้าใช้งานระบบ และสามารถระบุตัวตนที่แท้จริงได้จากศูนย์กลางบัญชีรายชื่อ ดังนั้นระบบซิงเกิ้ลไซออนนี้จึงทำหน้าที่เป็นศูนย์กลางในการพิสูจน์ตัวจริงอีกด้วย ซึ่งเปรียบเสมือนคนกลาง ที่ได้รับความไว้วางใจจากระบบต่างๆ ให้ตรวจสอบและระบุตัวตนของผู้ใช้งาน แล้วทำหน้าที่รับรองผู้ใช้งานนั้นว่าเป็นตัวจริง

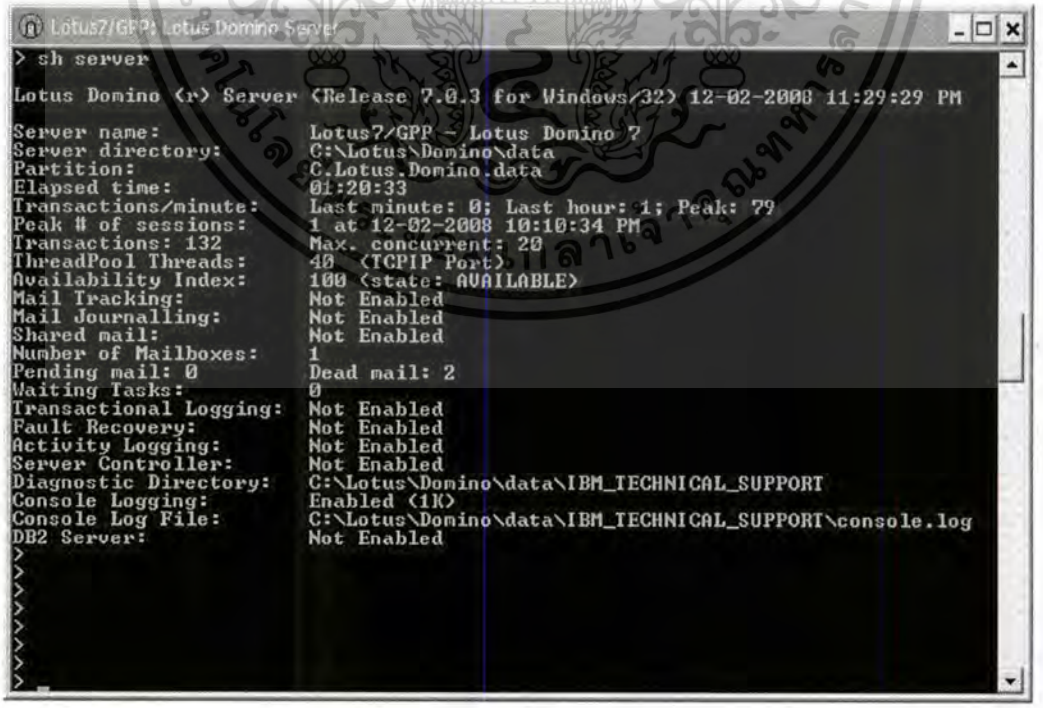
บทที่ 4
การพัฒนาระบบซิงเกิ้ลไซออน

ในหัวข้อนี้จะกล่าวถึงการพัฒนาทั้งในเรื่องของการปรับแต่งโครงสร้างระบบพื้นฐานที่มีอยู่และการเขียนโปรแกรม โดยจะอธิบายทั้งในส่วนของระบบที่เป็นผู้ให้บริการและระบบที่ต้องการเข้าร่วมใช้งานบริการ

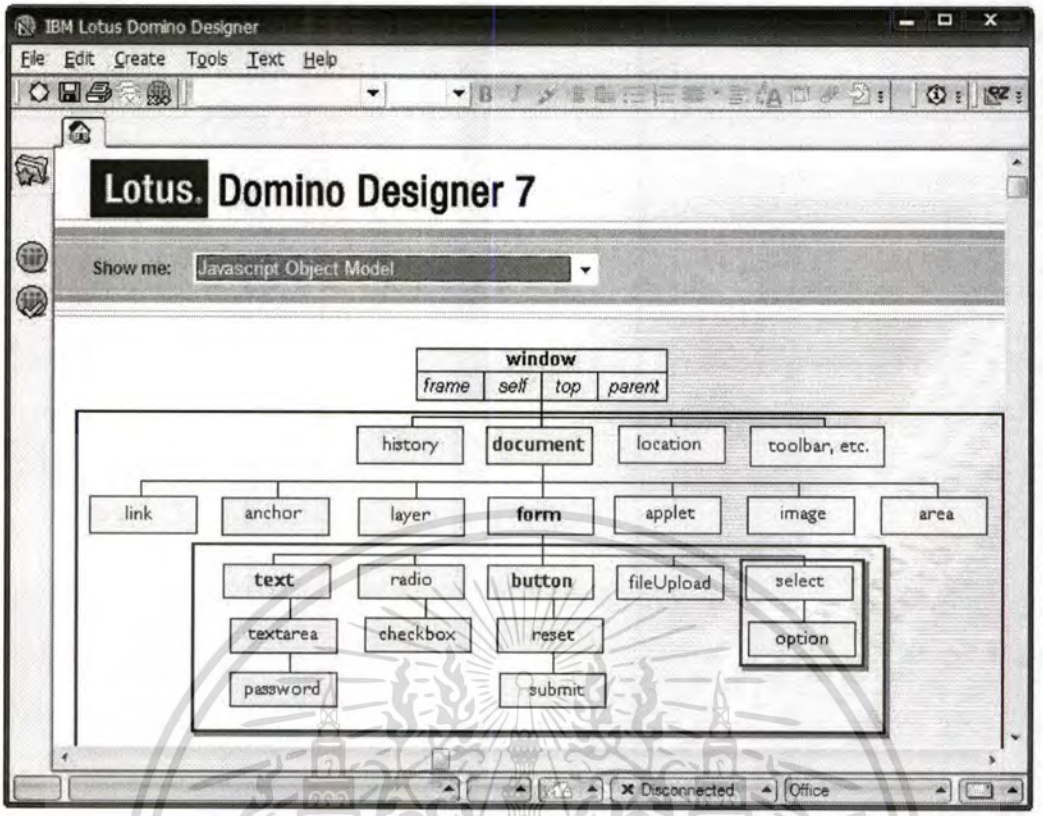
4.1 เครื่องมือที่ใช้ในการพัฒนา

ในโครงงานนี้ จะพัฒนาโดยใช้

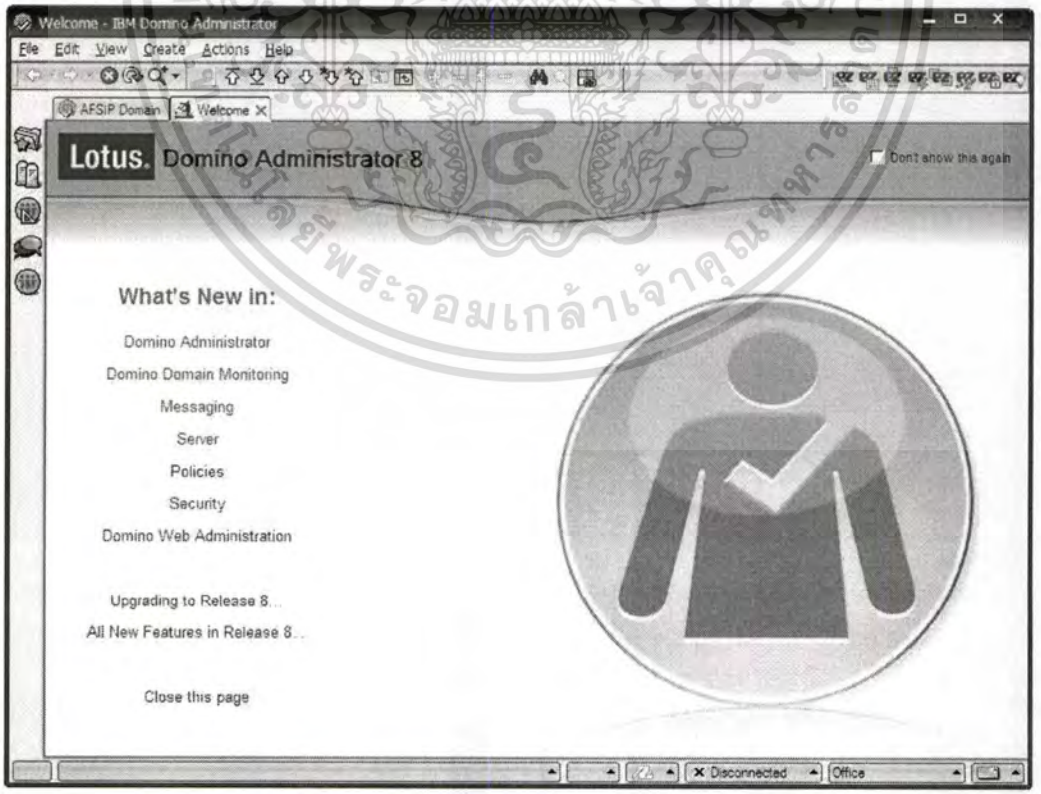
1. IBM Lotus Domino Server เพื่อใช้เป็น Directory Server/LDAP Server และเป็น Web Server ในส่วนของ Authentication Server ด้วย
2. IBM Lotus Designer เพื่อใช้ในการออกแบบ พัฒนาระบบและเขียนโปรแกรม
3. IBM Lotus Administrator เพื่อใช้ในการตรวจสอบและการควบคุมระบบ
4. Microsoft Internet Information Services (IIS) เพื่อใช้เป็น Web Server ในส่วนของ Web Application ที่จะมาใช้บริการ Single sign-on
5. Microsoft Visual Studio เพื่อใช้ในการออกแบบ พัฒนาระบบและเขียนโปรแกรม



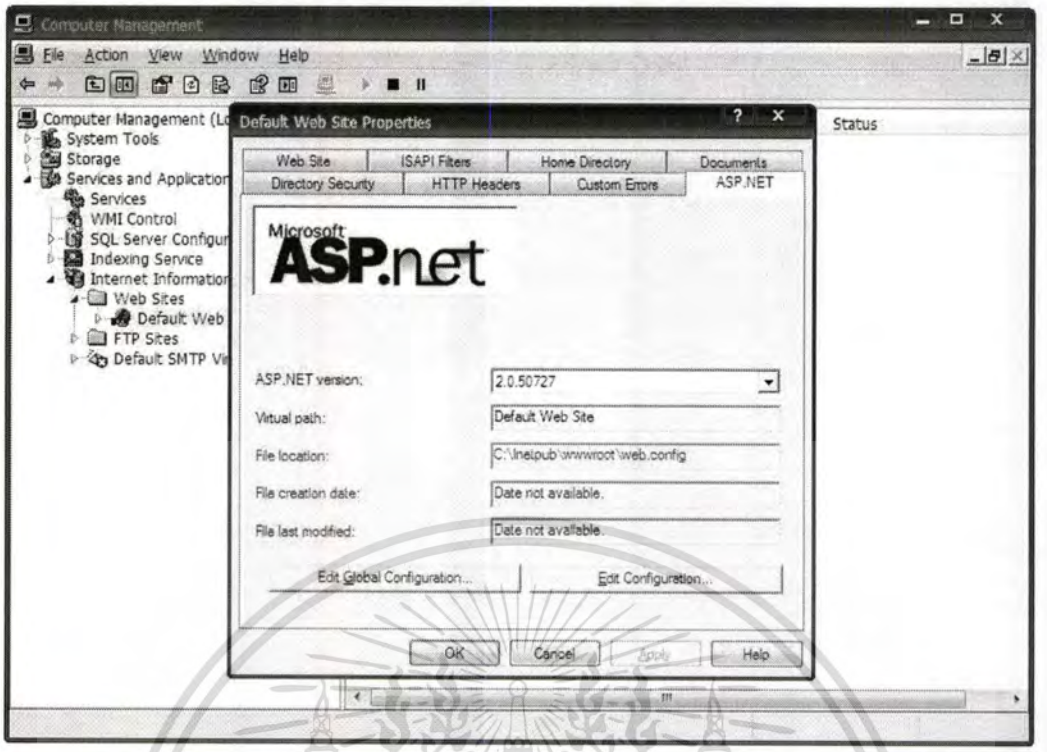
รูปที่ 4.1 แสดงโปรแกรม IBM Lotus Domino Server



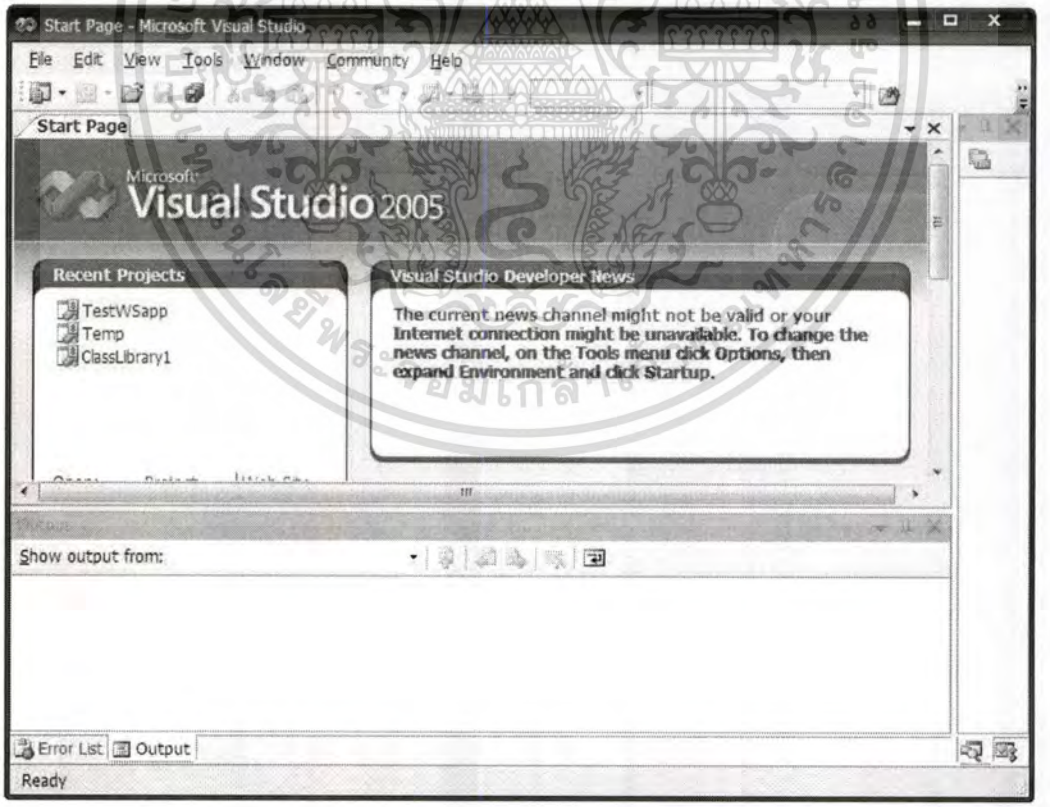
รูปที่ 4.2 แสดงโปรแกรม IBM Lotus Domino Designer



รูปที่ 4.3 แสดงโปรแกรม IBM Lotus Domino Designer

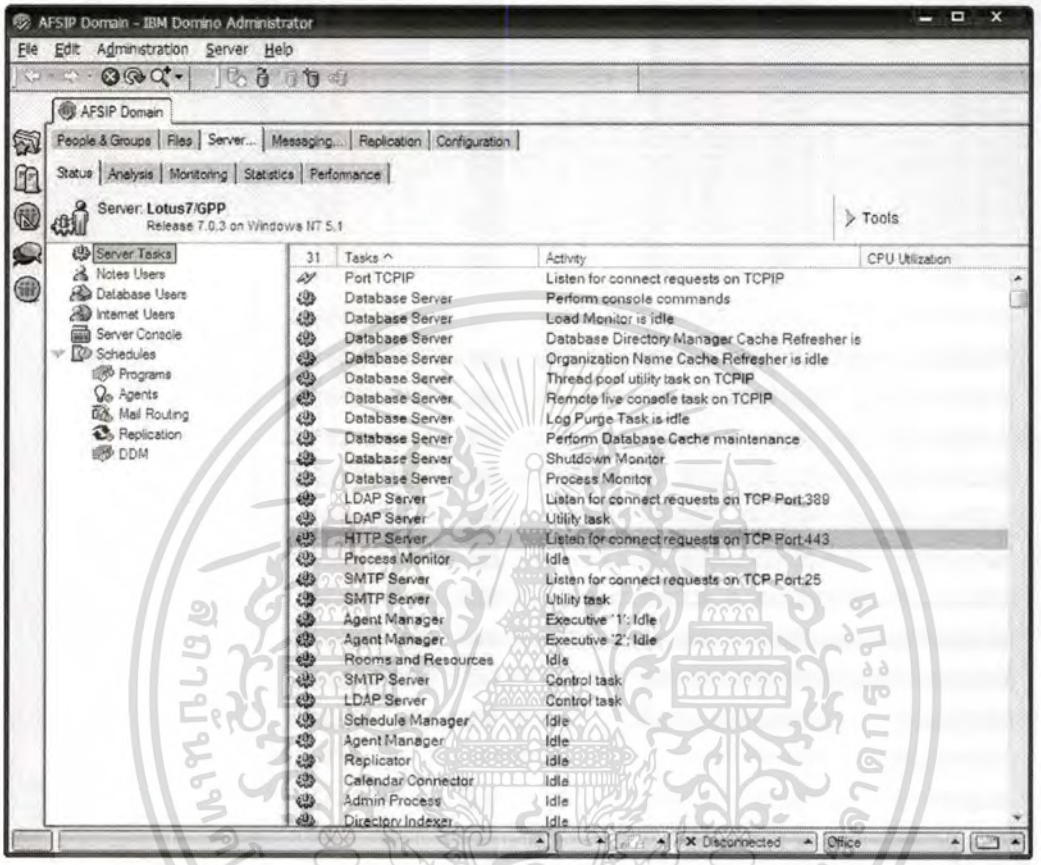


รูปที่ 4.4 แสดงโปรแกรม Microsoft Internet Information Services (IIS)



รูปที่ 4.5 แสดงโปรแกรม Microsoft Visual Studio

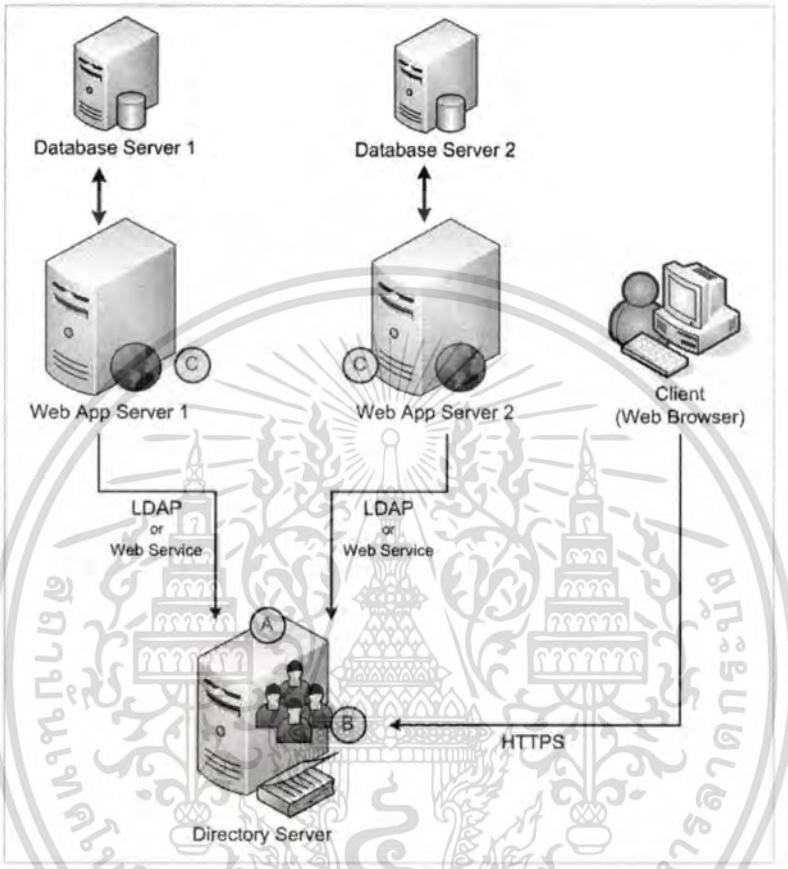
โดยในส่วนของ IBM Lotus Domino Server จะมีการเปิดใช้งาน Services ต่างๆ เช่น HTTPS, LDAP, SMTP เป็นต้น เพื่อให้รองรับกับการใช้งานในส่วนของการให้บริการ เป็น Authentication Server และ Directory Server



รูปที่ 4.6 แสดง Services ต่างๆที่เปิดใช้งานบน IBM Lotus Domino Server

4.2 บริการบัญชีรายชื่อผู้ใช้งาน (Directory Service)

ในโครงงานนี้ จะพัฒนาโดยอ้างอิงตามสภาพแวดล้อมขององค์กรที่ได้มีการวิเคราะห์ไว้ในบทที่แล้ว



รูปที่ 4.7 แสดงแผนผังของการติดต่อกับ Directory Server

ในส่วนของ Directory Service นี้จะประกอบด้วย 2 ส่วนสำคัญคือ Directory Server และ Application Server ที่เข้าร่วมใช้บริการของระบบซิงเกิ้ลไซออน

การพัฒนาระบบในส่วนของ Directory Server จะใช้ IBM Lotus Domino Server และพัฒนาโปรแกรมโดยใช้ Lotus Script ส่วนของ Application Server จะใช้ IIS Server และพัฒนาโปรแกรมโดยใช้ ASP .NET 2005

4.2.1 Directory Server

Directory Server จะทำหน้าที่ให้บริการโดยเปิด Service LDAP หรือ Web Service เพื่อขอใช้บริการ Creating, Retrieving, Updating และ Deleting User Accounts ตามสิทธิ์ที่กำหนดไว้ใน Directory โดยบริการดังกล่าวทั้งหมดจะอยู่ในส่วนของ (A) ดังรูปที่ 4.7 ซึ่งข้อมูลของบัญชีผู้ใช้งานจะประกอบที่ Application จะต้องส่งมาเมื่อใช้บริการ Creating คือ

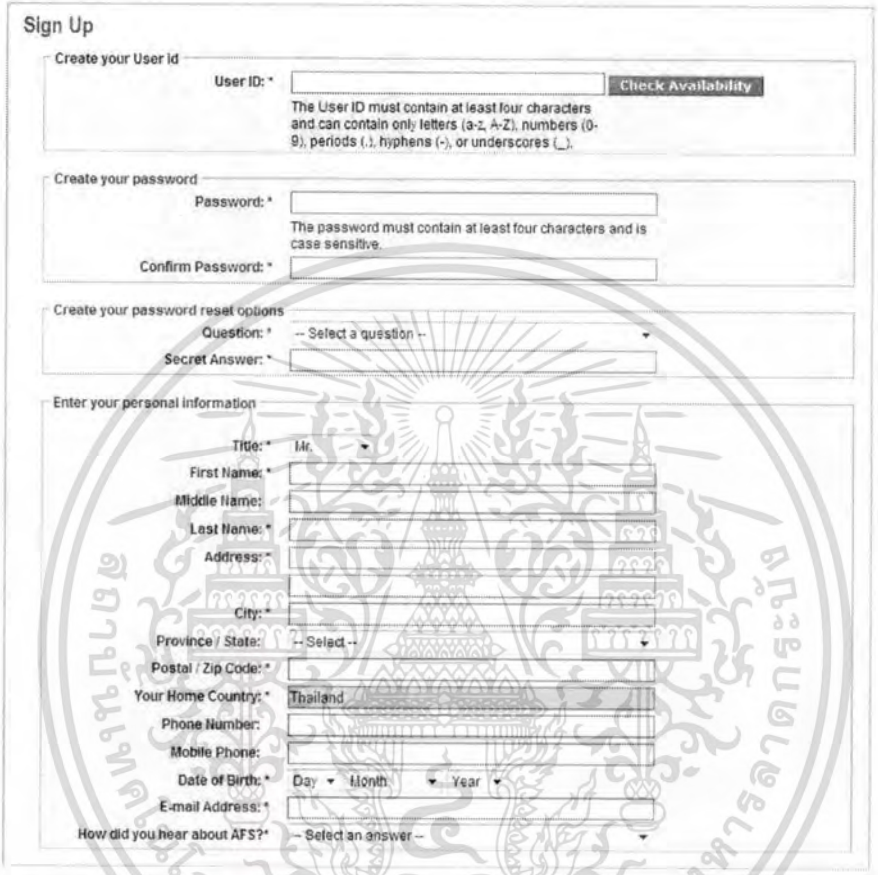
1. First Name - ชื่อเจ้าของบัญชีผู้ใช้งาน
2. Last Name - นามสกุลเจ้าของบัญชีผู้ใช้งาน
3. OU - รหัสตัวย่อของหน่วยงานโดยในที่นี้ใช้เป็นตัวย่อแทนประเทศของ Partner มี 3 ตัวอักษร
4. UserID - ชื่อผู้ใช้งานที่ใช้ในการ Log in
5. PersonID - รหัสประจำชื่อบัญชีผู้ใช้งานเพื่อใช้เชื่อโยงกับระบบต่างๆ
6. Password - รหัสผ่านถูกเก็บโดยมีการเข้ารหัสด้วยอัลกอริธึมของ Directory Server ในที่นี้คือ IBM Lotus Domino Directory
7. Email - ชื่อที่อยู่ของไปรษณีย์อิเลคทรอนิกส์
8. Secret Question - คำถามลับเพื่อใช้ในการ Reset Password
9. Secret Answer - คำตอบลับเพื่อใช้ในการ Reset Password

ในส่วนนี้จะถูกพัฒนาขึ้นเป็น Web Service ด้วย Lotus Script ซึ่งทำงานอยู่บน IBM Lotus Domino Server

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="urn:DefaultNamespace"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:apachesoap="http://xml.apache.org/xml-soap"
    xmlns:impl="urn:DefaultNamespace" xmlns:intf="urn:DefaultNamespace"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  + <wsdl:message name="UPDATEUSERResponse">
  + <wsdl:message name="DELETEUSERRequest">
  + <wsdl:message name="ADDUSERRequest">
  + <wsdl:message name="ADDUSERResponse">
  + <wsdl:message name="UPDATEUSERRequest">
  + <wsdl:message name="DELETEUSERResponse">
  + <wsdl:portType name="SSOuserMA">
  + <wsdl:binding name="DominoSoapBinding" type="impl:SSOuserMA">
  + <wsdl:service name="SSOuserMAService">
  </wsdl:definitions>
```

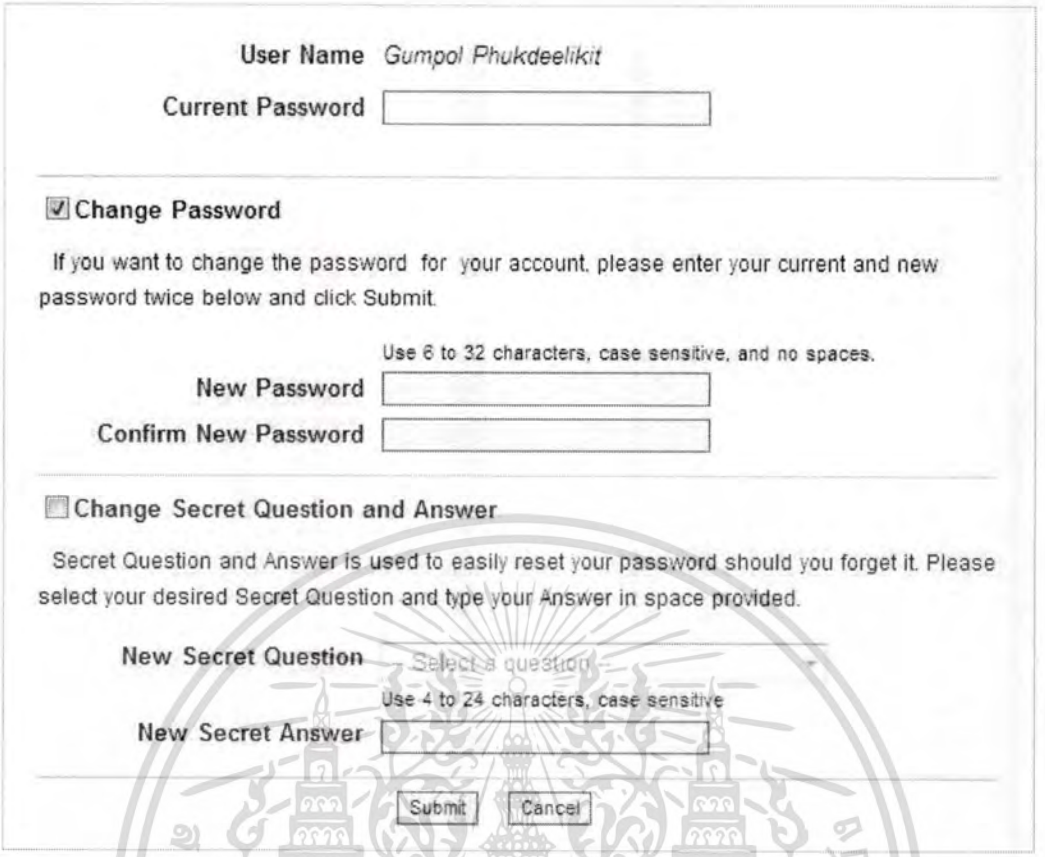
รูปที่ 4.8 แสดง Web Service ที่ใช้ในการติดต่อกับ Directory Server

ซึ่ง Application จะเป็นผู้สร้าง หน้าเว็บเพื่อเรียกใช้บริการเอง ซึ่งอยู่ในส่วนของ (C) ดังรูปที่ 4.7 โดยข้อมูลในส่วนอื่นๆ ของบัญชีผู้ใช้งานนอกเหนือจากที่กล่าวมาแล้วทั้ง 9 Fields นั้น จะอยู่ที่ Application เป็นผู้จัดการ ว่าจะเก็บข้อมูลเพิ่มอีกมากน้อยเพียงใด โดยจะเก็บไว้ที่ Database ของ Application นั้นๆ เอง ไม่ใช่ใน Directory

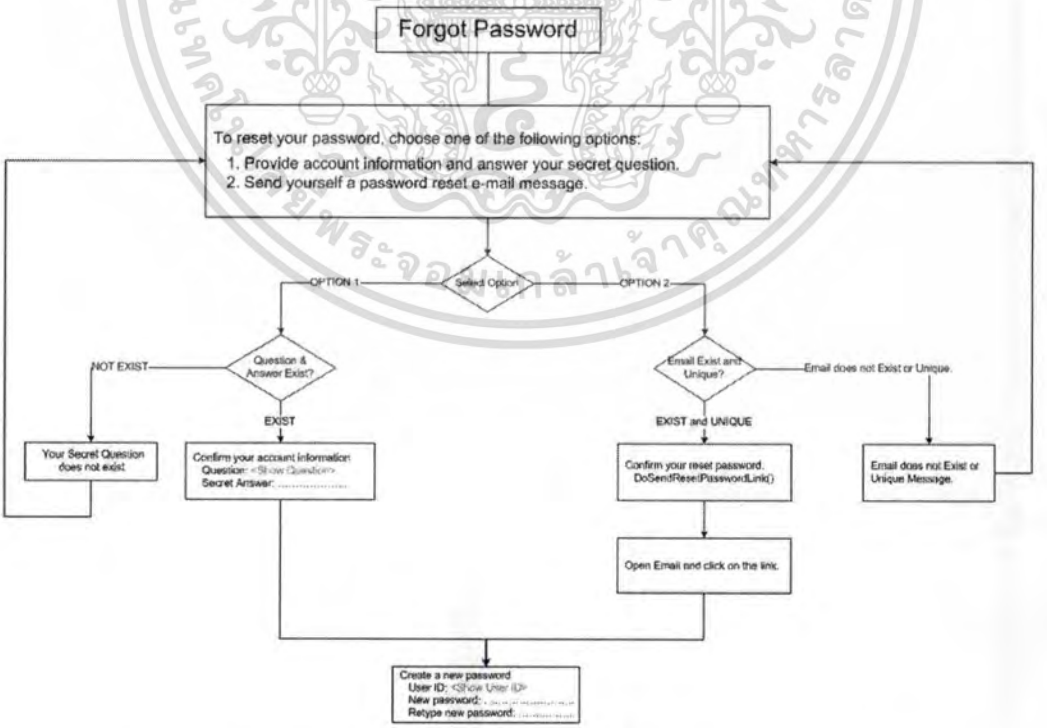


รูปที่ 4.9 แสดง Web Page ที่สร้างขึ้นโดย Web Application เพื่อเรียกใช้ Directory Service

ในส่วนของการเปลี่ยนรหัสผ่านหรือต้องการตั้งค่ารหัสผ่านใหม่นั้นจะอยู่ในส่วนของ (B) ดังรูปที่ 4.7 ซึ่งในเรื่องของการเปลี่ยนรหัสผ่านและการตั้งค่ารหัสผ่านใหม่นั้น จะมีเงื่อนไขและการไหลของข้อมูลตามนโยบายของ Directory Server เป็นผู้กำหนด ดังนั้นในส่วนนี้จะประกอบไปด้วยบริการเปลี่ยนรหัสผ่าน (Change Password) ซึ่งผู้ใช้งานต้องทำการเข้าสู่ระบบก่อนจึงจะใช้งานในส่วนนี้ได้ และบริการตั้งค่ารหัสผ่านใหม่ (Reset Password) ซึ่งจะเรียกใช้ในกรณีที่ผู้ใช้งานลืมรหัสผ่านและไม่สามารถเข้าสู่ระบบได้



รูปที่ 4.10 แสดง Web Page ที่ใช้ในการเปลี่ยนรหัสผ่านและข้อมูลลับ



รูปที่ 4.11 แสดง Work Flow ของบริการในส่วนการตั้งค่ารหัสผ่านใหม่

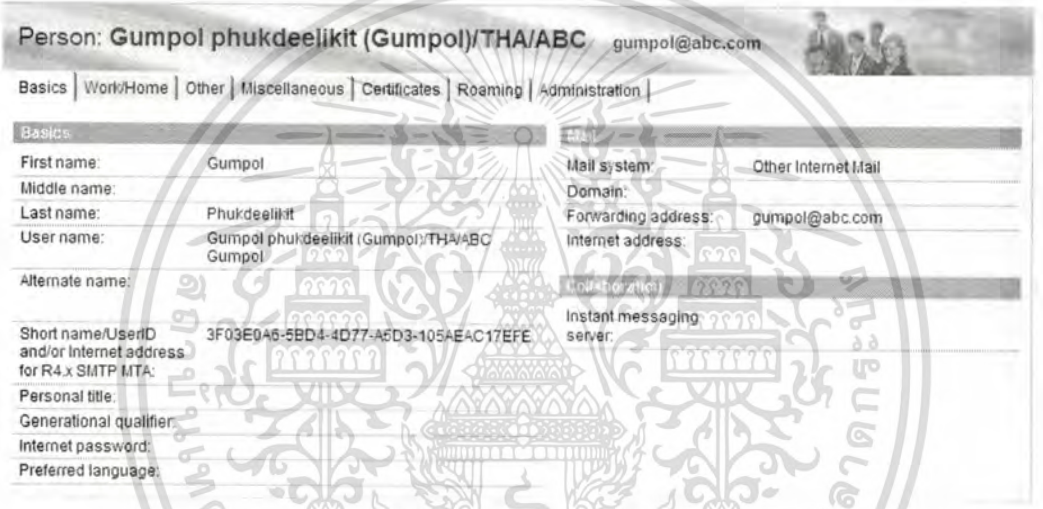
4.2.2 ระบบการตั้งชื่อบัญชีผู้ใช้งานและโครงสร้าง

จากข้อมูลของผู้ใช้งานที่จะเก็บไว้ใน Directory กลางและ Application จะต้องส่งมาให้ จะมีบางส่วนที่นำมาจัดรูปแบบเพื่อเป็นชื่อที่ไม่ซ้ำกัน โดยข้อมูลที่จำนำมาประกอบกันได้แก่ First Name, Last Name, OU และ UserID

โดยจะมีรูปแบบดังนี้ First Name Last Name (UserID)/OU/Domain

ตัวอย่างเช่น Gumpol Phukdeelikit (Gumpol)/THA/ABC

ซึ่งจะทำให้สามารถลดความซ้ำซ้อนของชื่อบัญชีผู้ใช้งานได้ การตั้งชื่อแบบตามลำดับชั้นนี้ (Hierarchical Naming) เพื่อให้สามารถแบ่งผู้ใช้งานตามโครงสร้างองค์กรได้อย่างชัดเจนและมีความเป็นหนึ่งเดียวได้มากที่สุดด้วย

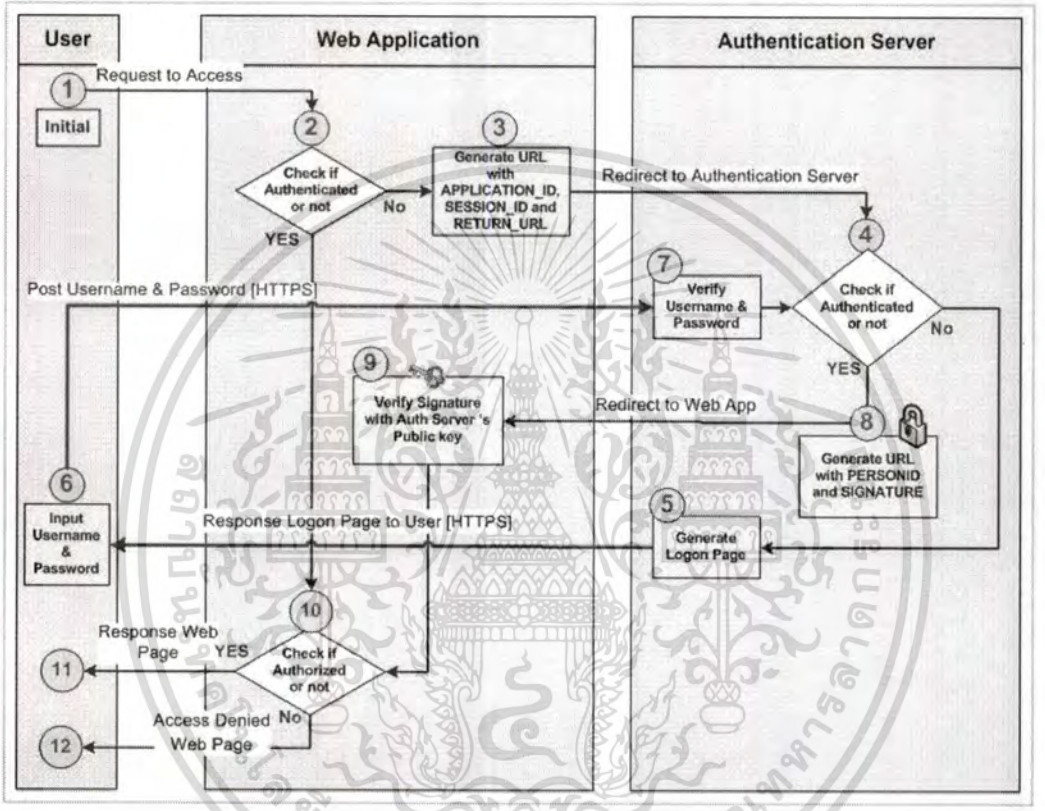


รูปที่ 4.12 แสดงข้อมูลของชื่อบัญชีผู้ใช้งานที่เก็บไว้ใน Lotus Domino Directory

4.3 บริการพิสูจน์ตัวจริง (Authentication Service)

ระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ ในส่วนของการพิสูจน์ตัวจริงนั้น ถือเป็นบริการหลัก ที่จะมีการใช้งานมากที่สุดและจะต้องมีความปลอดภัยมากที่สุดด้วย

4.3.1 ขั้นตอนในการตรวจสอบพิสูจน์ตัวจริง



รูปที่ 4.13 แสดงขั้นตอนในการตรวจสอบพิสูจน์ตัวจริง

สำหรับขั้นตอนในการพิสูจน์ตัวจริงนั้น มีเงื่อนไขและการทำงานดังต่อไปนี้

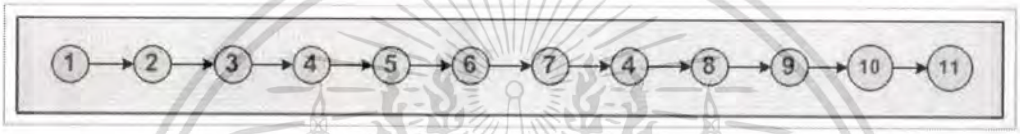
1. ผู้ใช้งานทำการร้องขอการใช้งานระบบไปยัง Web Application
2. Web Application ตรวจสอบว่าได้ผ่านการพิสูจน์ตัวจริงแล้วหรือไม่ หากใช่จะไปที่ข้อ 10) หากไม่ใช่จะไปที่ ข้อ 3)
3. กรณีไม่ที่ยังไม่ผ่านการพิสูจน์ตัวจริง Web Application จะทำการสร้าง URL พร้อมกับ Parameter ต่างๆ และ Redirect ผู้ใช้งานไปยัง Authentication Server เพื่อตรวจพิสูจน์ตัวจริง
4. Authentication Server ทำการตรวจสอบว่าผู้ใช้งานผ่านการพิสูจน์ตัวจริงแล้วหรือยัง หากใช้จะไปที่ข้อ 10) หากไม่ใช่จะไปที่ข้อ 5)
5. กรณีไม่ที่ยังไม่ผ่านการพิสูจน์ตัวจริง Authentication Server จะทำการ Response Logon Page ไปยังผู้ใช้งาน
6. ผู้ใช้งานป้อนข้อมูลชื่อบัญชีผู้ใช้งานและรหัสผ่านส่งกลับมาที่ Authentication Server
7. Authentication Server ตรวจสอบชื่อบัญชีผู้ใช้งานและรหัสผ่านจาก Directory แล้ว
8. กรณีที่ผ่านการพิสูจน์ตัวจริงแล้ว Authentication Server จะทำการสร้าง URL กับ Parameter ต่างๆ รวมทั้ง Signature และ Redirect ผู้ใช้งานไปยัง Web Application
9. Web Application ทำการตรวจสอบ Signature ที่ส่งมา โดยการใช้งาน Public Key ของ Authentication Server
10. กรณีที่ผ่านการพิสูจน์ตัวจริงแล้ว Web Application ทำการตรวจสอบสิทธิ์ในการเข้าใช้งาน หากมีสิทธิ์ในการเข้าใช้งาน ไปที่ข้อ 11) หากไม่มีสิทธิ์เข้าใช้งานไปข้อ 12)
11. กรณีที่มีสิทธิ์ในการเข้าใช้งาน Web Application ทำการ Response Web Page ที่ผู้ใช้งานได้ร้องขอไปตอนต้น
12. กรณีที่ไม่มีสิทธิ์ในการเข้าใช้งาน Web Application ทำการ Response Web Page ที่แจ้งเตือนว่าผู้ใช้งานไม่มีสิทธิ์ในการเข้าใช้งาน

ในกรณีที่ผู้ใช้งานมีการร้องขอเข้าใช้งานระบบจาก Web Application อื่นๆ ก็จะดำเนินการตามขั้นตอนต่างๆ เช่นกัน

จากขั้นตอนต่างๆ จะต้องมีการพัฒนาโปรแกรมเพื่อใช้ในระบบซิงเกิ้ลไซออน ซึ่งในส่วนของ Web Application ที่จะต้องพัฒนาโปรแกรมคือ ขั้นตอนที่ 2, 3, 9 และ 10 และขั้นตอนในส่วนของ Authentication Server ที่จะต้องพัฒนาโปรแกรมคือ ขั้นตอนที่ 5 และ 8 ซึ่งรายละเอียดจะกล่าวในหัวข้อถัดไปเรื่องการพัฒนาโปรแกรมของแต่ละขั้นตอน

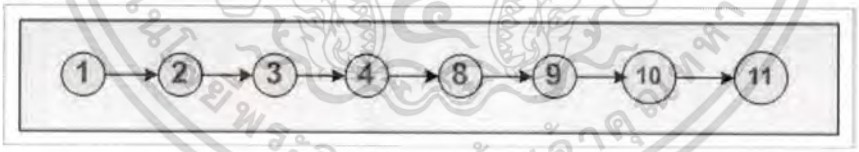
โดยนอกเหนือจากนี้ จะใช้ความสามารถของตัวเครื่องมือที่นำมาใช้งานร่วมกับระบบในการจัดการ แต่อย่างไรก็ตามจะต้องมีการปรับแต่งเพื่อค่าต่างๆ เพื่อให้สามารถทำงานได้อย่างถูกต้องและปลอดภัยด้วย

หากพิจารณาจากแผนการที่ผู้ใช้งานมีการเริ่มใช้งาน โดยมีการร้องขอเข้าใช้งานกับ Web Application ที่ใช้บริการระบบซิงเกิ้ลไซออน เป็นระบบแรกและสามารถเข้าใช้งานได้สำเร็จจะต้อง ผ่านขั้นตอนต่างๆ โดยแสดงได้ดังรูปที่ 4.14



รูปที่ 4.14 แสดงขั้นตอนในการเข้าใช้งานกับ Web Application ระบบแรกได้สำเร็จ

และพิจารณาแผนการที่ผู้ใช้งานร้องขอเข้าใช้งานกับ Web Application ที่ใช้บริการระบบซิงเกิ้ลไซออน เป็นระบบถัดไป และสามารถเข้าใช้งานได้สำเร็จจะต้อง ผ่านขั้นตอนต่างๆ โดยแสดงได้ดังรูปที่ 4.15



รูปที่ 4.15 แสดงขั้นตอนในการเข้าใช้งานกับ Web Application ระบบถัดไปได้สำเร็จ

4.3.2 การพัฒนาระบบในขั้นตอนต่างๆ

สำหรับรายละเอียดของขั้นตอนต่างๆ ที่เกี่ยวข้องสามารถอธิบายได้ดังนี้

4.3.2.1 การเตรียมกุญแจเพื่อใช้ในส่วนของการสร้างลายมือชื่อดิจิตอล

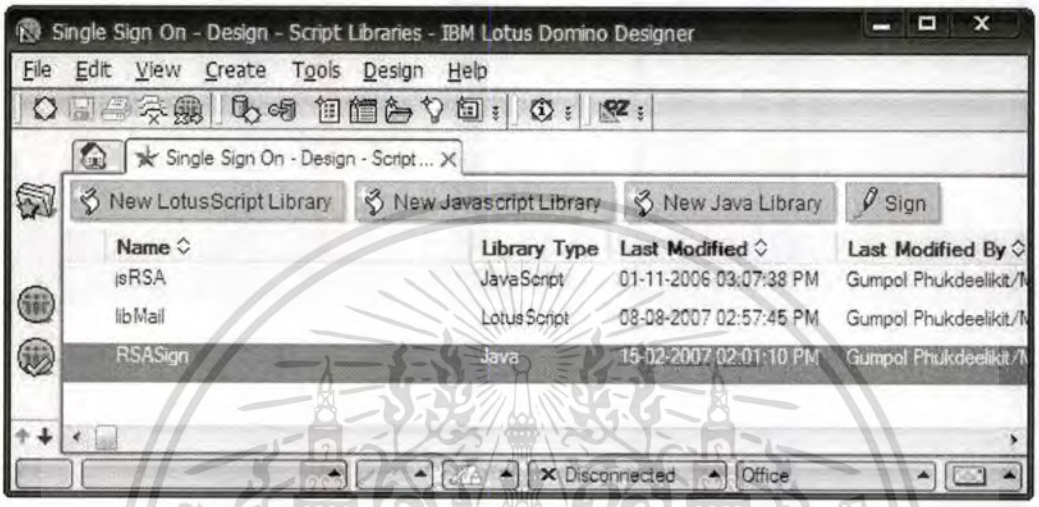
การเตรียมกุญแจเพื่อใช้ในส่วนของการสร้าง Signature นั้น ทาง Authentication Service จะทำหน้าที่ในการ Generate Key ดังกล่าวโดยใช้อัลกอริธึมแบบอสมมาตร ซึ่งในที่นี้คือ อัลกอริธึม RSA ซึ่งจะได้ Key 2 ส่วนคือ Private Key และ Public Key โดยที่ Private Key จะถูกเก็บและใช้โดย Authentication Server ในการสร้างลายมือชื่อดิจิตอล ส่วน Public Key จะถูกเก็บและใช้โดย Web Application ต่างๆที่ร่วมใช้บริการของระบบชิงเกิ้ลไซออน ในการตรวจสอบลายมือชื่อดิจิตอล ในส่วนนี้พัฒนาโดยภาษาจาวาและทำงานอยู่บน IBM Lotus Domino Server

Private Key	IBMJCA RSA Private Crt Key: modulus: 144498414296621339655173946243186068411136835815987352479104208253701229160519876808053039369091157634888871565388792765852728009368298720006913511228522485003633355762807875640530628015046903450589266808030621125703360226679160468325340504444872370381616160669187827444184458051181144244596868421477778663033 public exponent: 65537 private exponent: 123806017357887627818138264330796101643684569094394339325978599903288698289844089941953312886662597975407478464679669368267732474511325048840017203755191620423987358979711498448970682713230219544786293046658034780280873839491600640134126928566862596533040152663986676279196369161293900834251771714928031721473 prime P: 122391958312331008942787785211073935089525877425195527397485009115215643619125836270894220004123607089396252524363939921999793277515057156473680748404840001 prime Q: 118062016728155498280306117800630713867635190608428512318296303652740925548039168402757315470900022106032005870721424117258428371799775484392668015986910331 prime exponent P: 655501127113358623966896754643742484648726413724211210713295362916225810005236256329765952384872455076950190519122677458869454512836795427350381352367347 3 prime exponent Q: 282378215697062336625692112322029760269036057919513699221705991997911715209044351238723304393908455759655109636321211382581727074471135498703796504355527 3 crt coefficient: 806678649852318255058580672306960875161885710610735637718755573783171532345183729231119213684454323435263977497337218781057965870119568445029249958681341 6
Public Key	IBMJCA RSA Public Key: modulus: 144498414296621339655173946243186068411136835815987352479104208253701229160519876808053039369091157634888871565388792765852728009368298720006913511228522485003633355762807875640530628015046903450589266808030621125703360226679160468325340504444872370381616160669187827444184458051181144244596868421477778663033 public exponent: 65537

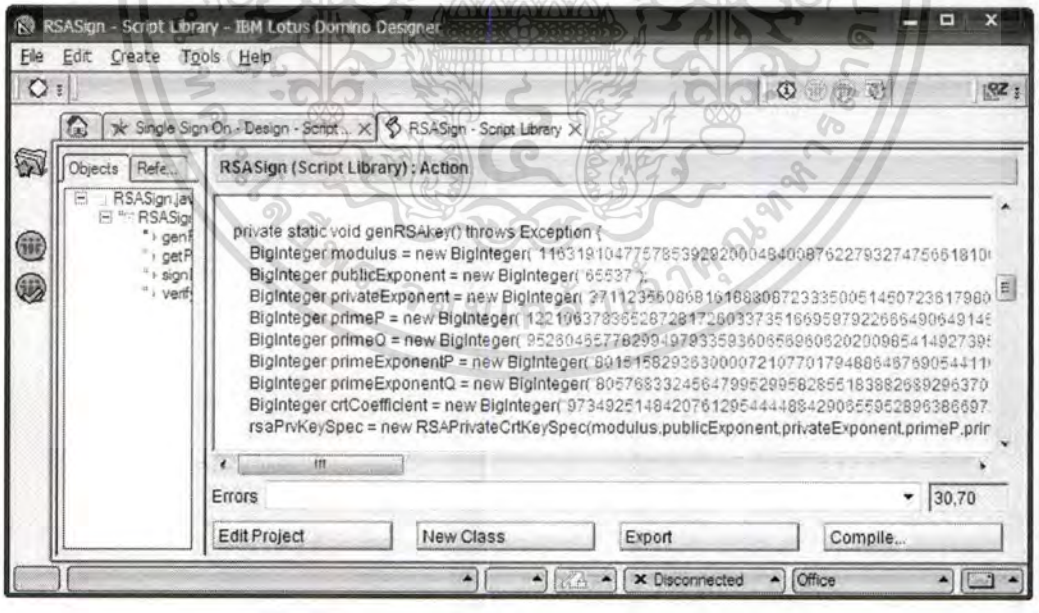
รูปที่ 4.16 แสดงกุญแจเพื่อใช้ในการสร้างและตรวจสอบลายมือชื่อดิจิตอล

4.3.2.2 การจัดเก็บกุญแจเพื่อใช้ในส่วนของการสร้างลายมือชื่อดิจิตอล

ในส่วนนี้จะเป็นกุญแจส่วนตัว (Private Key) ซึ่งถูกจัดเก็บและเรียกใช้งานโดย IBM Lotus Domino Server ซึ่งทำหน้าที่ให้บริการเป็น Authentication Server ภายใน Lotus Notes Database สำหรับภาษาที่ใช้ในการพัฒนาจะใช้ภาษาจาวาเป็น Library เรียกใช้โดย Lotus Script



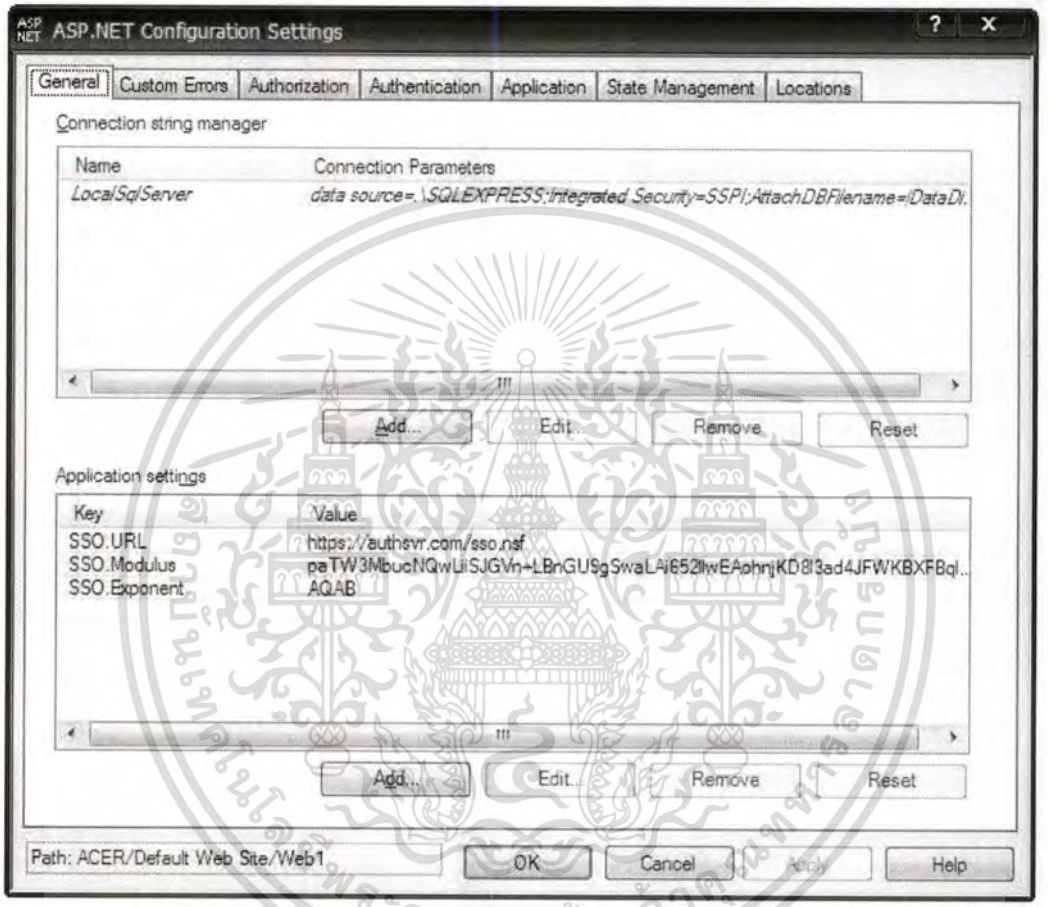
รูปที่ 4.17 แสดงที่เก็บกุญแจส่วนตัว (Private key)



รูปที่ 4.18 แสดงการเรียกใช้งานกุญแจส่วนตัว (Private key)

4.3.2.3 การจัดเก็บกุญแจเพื่อใช้ในส่วนของการตรวจสอบลายมือชื่อดิจิตอล

ในส่วนนี้จะเป็นกุญแจสาธารณะ (Public Key) ซึ่งถูกจัดเก็บและเรียกใช้งานโดย Web Application ที่เข้าร่วมใช้งานบริการซิงเกิ้ลไซออน ซึ่งในระบบนี้จะใช้ ASP .NET จัดทำเป็น Application และ Public Key ดังกล่าวจะถูกเก็บไว้ในส่วนของ ASP .NET Configuration Settings



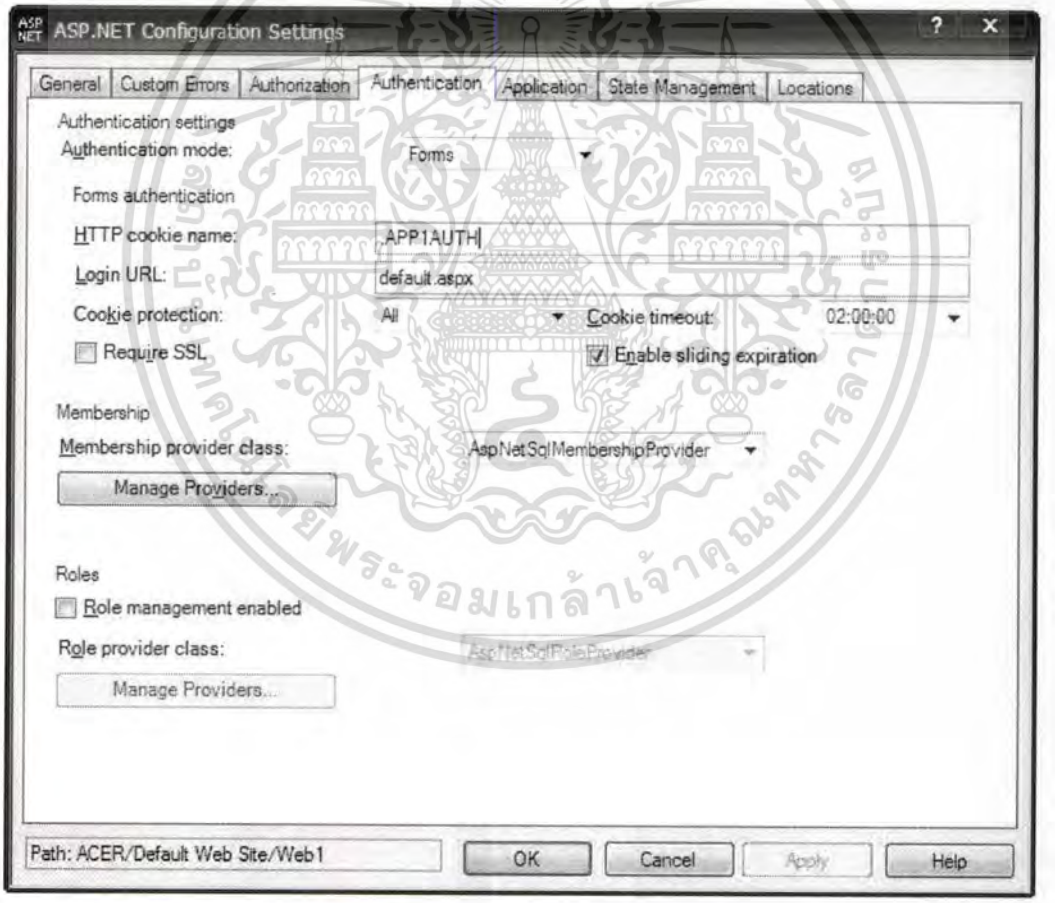
รูปที่ 4.19 แสดงที่เก็บกุญแจสาธารณะ (Public Key)

โดยกุญแจสาธารณะดังกล่าวจะถูกเรียกใช้งาน เมื่อมีการตรวจสอบลายมือชื่อดิจิตอล โดย ASP .NET ทำการตรวจสอบ Signature ที่ถูกส่งมาจาก Authentication Server

4.3.2.4 การพัฒนาโปรแกรมในส่วนของ Web Application

การพัฒนาโปรแกรมในส่วนของ Web Application เพื่อให้สามารถเข้าร่วมใช้งานระบบซิงเกิ้ลไซออนได้นั้น จะเกี่ยวข้องกับขั้นตอนในการพิสูจน์ตัวจริงในรูปที่ 4.13 ขั้นตอนที่ 2, 3, 9 และ 10 ซึ่งมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 2) ระบบที่มีการร้องขอการใช้งานจะต้องตรวจสอบข้อมูลจากผู้ใช้งานโดย Browser จะเก็บข้อมูลในส่วนนี้ไว้โดยใช้ Cookie และเพื่อให้เกิดความเข้าใจจะเรียกชุดข้อมูลเหล่านี้ว่า Authentication Ticket ซึ่งเป็นข้อมูลที่ใช้ในการตรวจสอบว่า ผู้ใช้งานตาม Session ดังกล่าว ได้ผ่านกระบวนการ Authenticate จาก Web Application แล้วหรือยัง โดยชื่อของ Cookie จะถูกระบุในส่วนของ HTTP cookie name และสามารถตั้งค่า Cookie timeout เพื่อใช้ในการกำหนดขอบเขตของเวลาในการเข้าใช้งานของผู้ใช้งานในแต่ละ Session ได้อีกด้วย



รูปที่ 4.20 แสดงการตั้งค่าในส่วนของ Authentication ของ Web Application

ขั้นตอนที่ 3) ระบบตรวจสอบแล้วพบว่ายังไม่ได้ผ่านการ Authenticate จึงทำการสร้าง URL โดยประกอบด้วย Parameter ต่างๆ คือ

- APPLICATION_ID เพื่อใช้ในการระบุว่าเป็น Request จากระบบใด
- SESSION_ID เพื่อใช้ในการระบุว่าผู้ใช้งานมาจาก Session ใดในการเข้าใช้ระบบ ซึ่งค่านี้จะถูกนำไปสร้างเป็นลายมือชื่อดิจิตอล (Digital Signature) ในภายหลังโดย Authentication Server
- RETURN_URL เพื่อใช้ในการระบุว่าให้ Redirect ไปที่ URL ใดหลังจากที่ผ่านการตรวจสอบพิสูจน์ตัวจริงจาก Authentication Server

โดยข้อมูลทั้ง 3 ส่วนนี้จะนำมาประกอบเป็น URL แล้ว Redirect ผู้ใช้งานไปยัง URL นี้ ซึ่ง Address ของ URL นี้จะถูกนำมาจากค่าที่ระบุไว้ใน ASP .NET Configuration Settings ด้วย

```
https://authsvr.com/sso.nsf/sso?OpenAgent&APPLICATION_ID=WebApp1&
SESSION_ID=507DD76D-C315-4F63-962E-B98078723BAC&
RETURN_URL=http%3A%2F%2Fwebapp1.net%2Fweb1%2FDefault.aspx
```

รูปที่ 4.21 แสดงตัวอย่างของ URL ที่ถูกสร้างขึ้นโดย Web Application

ขั้นตอนที่ 9) ระบบจะนำ URL ที่ถูก Redirect จาก Authentication Server ซึ่งประกอบด้วย Parameter และ Signature มาทำการตรวจสอบความถูก เพื่อดำเนินการต่อไป โดย Parameter ดังกล่าวคือ PERSON_ID ซึ่งถูกนำไปรวมกับ SESSION_ID สร้างเป็น SIGNATURE ซึ่งหน้าที่ของโปรแกรมในส่วนนี้ก็คือ นำข้อมูลเหล่านี้มาเข้าสู่กระบวนการ Verify Signature ซึ่งจะมีการเรียกใช้งาน Public Key ที่เก็บไว้ด้วย ซึ่งการทำงานเหล่านี้จะเป็นโปรแกรมที่ถูกทำโดย Server ดังนั้น จะไม่มีหน้าจอการทำงานแสดงให้เห็นแต่อย่างใด

ขั้นตอนที่ 10) ระบบจะทำการตรวจสอบสิทธิ์ในการเข้าถึง (Authorization) ซึ่งจะทำหลังจากระบบได้ผ่านการตรวจพิสูจน์ตัวจริงผู้ใช้งานสำเร็จ และตรวจสอบลายมือชื่อดิจิตอลว่าถูกต้อง เป็นที่เรียบร้อยแล้ว โดยในส่วนนี้จะขึ้นอยู่กับเงื่อนไขการเข้างานของแต่ละระบบว่าต้องตรวจสอบจากเงื่อนไขใด ดังที่ได้กล่าวมาแล้วว่า ซิงเกิ้ลไซออนจะนำมาช่วยในเรื่องการพิสูจน์ตัวจริง (Authentication) เท่านั้น ส่วนการตรวจสอบสิทธิ์ในการเข้าถึงระบบ นั้นยังคงเป็นหน้าที่ของระบบนั้นๆต่อไป

4.3.2.5 การพัฒนาโปรแกรมในส่วนของ Authentication Server

การพัฒนาโปรแกรมในส่วนของ Authentication Server นั้นจะจัดการในเรื่องการพิสูจน์ตัวจริง โดยจะเกี่ยวข้องกับขั้นตอนในการพิสูจน์ตัวจริงในรูปที่ 4.13 ขั้นตอนที่ 4, 5, 7 และ 8 สำหรับที่จะต้องพัฒนาโปรแกรมคือ ขั้นตอนที่ 5 และ 8 เท่านั้น ในขั้นตอน ที่ 4 และ 7 จะใช้กลไกการทำงานของ IBM Lotus Domino Server ซึ่งทำหน้าที่เป็น Authentication Server มาช่วยโดยมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 4) Authentication Server โดยกลไกของ IBM Lotus Domino Server จะทำการ ตรวจสอบว่าผู้ใช้งานว่าได้ผ่านการพิสูจน์ตัวจริงแล้วหรือยัง โดยมีการตรวจสอบจากจากผู้ใช้งานโดย Browser จะเก็บข้อมูลในส่วนนี้ไว้โดยใช้ Cookie เพื่อให้เกิดความเข้าใจจะเรียกชุดข้อมูลเหล่านี้ว่า LtpaToken ซึ่งเป็นกลไกสำคัญที่ใช้ในการตรวจสอบพิสูจน์ตัวจริง ของ Authentication Server ซึ่งสามารถตั้งค่า Timeout เพื่อกำหนดขอบเขตของเวลาในการเข้าใช้งานของผู้ใช้งานแต่ละ Session ได้ และในส่วนนี้ยังมีความสำคัญในเรื่องของกลไกการตรวจสอบพิสูจน์ตัวจริงของ Server ต่างๆ ในกลุ่มของ IBM Lotus Domino Server ที่นำมาใช้เป็น Web Application Server อีกด้วย

รูปที่ 4.22 แสดงตั้งค่าในส่วนของ Authentication ของ Authentication Server

ขั้นตอนที่ 5) Authentication Server จะทำการ Response Logon Page ไปยังผู้ใช้งาน โดยในส่วนนี้จะต้องพัฒนา Logon Page โดยใช้ IBM Lotus Designer เพื่อให้ผู้ใช้งานกรอกข้อมูลชื่อผู้ใช้งานและรหัสผ่านส่งมายัง Authentication Server ได้

รูปที่ 4.23 แสดง Logon Page ของ Authentication Server

ขั้นตอนที่ 7) Authentication Server โดยกลไกของ IBM Lotus Domino Server จะทำการตรวจสอบข้อมูลชื่อผู้ใช้งานและรหัสผ่านที่ถูกส่งเข้ามาจากข้อมูลที่อยู่ใน Lotus Domino Directory

ขั้นตอนที่ 8) Authentication Server จะทำการสร้าง URL หลังจากที่ผ่านการตรวจพิสูจน์ตัวจริงสำเร็จแล้ว โดย URL ประกอบด้วย Parameter และ Signature เพื่อใช้ในการยืนยัน และตรวจสอบความถูกต้องของข้อมูล โดยที่

- PERSON_ID จะได้จากข้อมูลที่เก็บไว้ ในบัญชีรายชื่อของผู้ใช้งานที่ผ่านการตรวจสอบพิสูจน์ตัวจริงสำเร็จแล้ว โดยจะต้องพัฒนาโปรแกรมเพื่อให้ได้ค่าในส่วนนี้มา
- SIGNATURE จะถูกสร้างด้วยข้อมูล PERSON_ID และข้อมูล SESSION_ID ที่ได้มาจาก Web Application ในขั้นตอนที่ 3) โดยนำมาต่อกันแล้วสร้างเป็น SIGNATURE ด้วยอัลกอริธึม RSA และ SHA-1 ด้วย Private Key ของ Authentication Server ที่เก็บไว้

หลังจากที่ได้ข้อมูล 2 ส่วนนี้ จึงนำมาประกอบเป็น URL แล้ว Redirect ผู้ใช้งานไปยัง URL นี้ ซึ่ง Address ของ URL นี้นำมาจากค่า RETURN_URL ที่ถูกส่งมาในขั้นตอนที่ 3) เช่นกัน

http://webapp1.net/web1/Default.aspx?**PERSON_ID**=3F03E0A6-5BD4-4D77-A5D3-105AEAC17EFE&**SIGNATURE**=XTYpBceX%2BChCiz0%2FBBEEwXmM6y%2BUCSKA3ppPUc2FRr%2FNhZvU8Dq%2FmFLcZG2YmND9%2FI2xAY0ZkIMm%00w%2Fv77dCKSJ0O7ZRa6EOHRZajY4w4CqX2P%2FnV%2B45O2K9VvmE2gVZ1AD%2FsM%2F1vLuAlbrlblVNV8u2R%00%2FaNb93%2BnkCmq1cEg%2Fyg%3D

รูปที่ 4.24 แสดงตัวอย่างของ URL ที่ถูกสร้างขึ้นโดย Authentication Server

4.4 บริการตรวจสอบและดูแลการทำงานของระบบ (Monitoring Service)

ในส่วนของการตรวจสอบและดูแลระบบนั้นจะอาศัยกลไกการทำงานของ IBM Lotus Domino Server เพื่อมาช่วยในการตรวจสอบหากเกิดข้อผิดพลาดขึ้น

รูปที่ 4.25 แสดงเครื่องมือที่ใช้ในการตรวจสอบของ IBM Lotus Domino Server

บทที่ 5
สรุปผลของโครงงานและข้อเสนอแนะ

5.1 บทสรุป

ในโครงงานนี้ ได้มีการศึกษาเทคโนโลยีต่างๆ และนำเอาความรู้ที่ได้ มาประยุกต์เพื่อทำให้เกิดระบบซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ ในการที่จะทำให้ผู้ใช้งานได้รับประโยชน์มากที่สุดในเรื่องนี้ โดยได้มีการใช้ลายมือชื่อดิจิตอลมาช่วยในเรื่องของการยืนยันและรับรองข้อมูล หลังจากที่ได้มีการพิสูจน์ตัวจริง โดยรูปแบบของระบบจะเป็นการรวมสู่ศูนย์กลางทั้งในเรื่องของบัญชีรายชื่อผู้ใช้งานพร้อมหลักฐานในการยืนยันตัวตน และการตรวจพิสูจน์ตัวจริง

จากที่ได้มีการวิเคราะห์และออกแบบระบบนี้ขึ้นมา ทำให้ช่วยในเรื่องของการลดความซ้ำซ้อนของบัญชีรายชื่อผู้ใช้งาน และเป็นการลดการส่งข้อมูลที่เป็นความลับ เช่น รหัสผ่าน ออกสู่เครือข่ายอินเตอร์เน็ต เนื่องจากปัจจุบันจำนวนโปรแกรมประยุกต์บนเว็บในองค์กรต่างๆ มีจำนวนเพิ่มมากขึ้น และพัฒนาด้วยภาษาที่แตกต่างกันอีกด้วย ระบบนี้จึงทำให้การทำงานร่วมกันเป็นไปอย่างราบรื่นและมีประสิทธิภาพมากยิ่งขึ้น

5.2 ข้อดีและข้อเสียของระบบ

โครงงานฉบับนี้เป็นการทำซิงเกิ้ลไซออนสำหรับโปรแกรมประยุกต์บนเว็บ (Single sign-on For Web-Based Application) แบบรวมการจัดการเข้าสู่ศูนย์กลาง ซึ่งมีข้อดีและข้อเสียดังนี้

5.2.1 ข้อดีของระบบ

- การรวมบริการบัญชีรายชื่อผู้ใช้งาน (Directory Service) เข้าสู่ศูนย์กลางทำให้การปรับปรุงข้อมูลสามารถทำได้สะดวกและง่ายมากขึ้น โดยทำแค่ที่เดียวไม่ต้องทำการปรับปรุงข้อมูลในทุกที่ที่มีชื่อบัญชีผู้ใช้งานในหลายๆระบบ อีกทั้งยังลดความซ้ำซ้อนในการจัดเก็บและจัดการกับรายชื่อบัญชีผู้ใช้งาน และผู้ใช้งานก็ไม่ต้องจำบัญชีผู้ใช้งานหลายๆบัญชีอีกด้วย
- การรวมบริการพิสูจน์ตัวจริง (Authentication Service) เข้าสู่ศูนย์กลาง ทำให้การควบคุมและรูปแบบการพิสูจน์ตัวจริงเป็นมาตรฐานเดียวกัน และเป็นการเพิ่มความปลอดภัยในการส่งข้อมูลเพื่อทำการแสดงตนและพิสูจน์ตัวตน เนื่องจากลดการส่งข้อมูลเหล่านี้ในระบบเครือข่าย

5.2.2 ข้อเสียของระบบ

- การรวมการจัดการเรื่องบัญชีรายชื่อไว้ด้วยกันทำให้ภาระหน้าที่ในการจัดเก็บและจัดการของส่วนกลางมีเพิ่มมากขึ้น และจำเป็นจะต้องสามารถให้บริการได้ตลอดเวลา เนื่องจากทุกระบบต้องมาเรียกใช้งาน
- การทำให้การพิสูจน์ตัวจริงอยู่ที่ศูนย์กลาง ทำให้ระบบต่างๆ จะต้องปรับการทำงานของโปรแกรมใหม่ และระบบศูนย์กลางจะต้องสามารถให้บริการได้ตลอดเวลา (High Availability) เนื่องจากหากระบบศูนย์กลางการพิสูจน์ตัวจริง ไม่สามารถใช้งานได้ ระบบอื่นๆก็จะไม่สามารถได้ด้วยเช่นกัน

5.3 ปัญหาและอุปสรรคระหว่างการพัฒนา

ปัญหาสำหรับการพัฒนาระบบนี้คือ จะต้องทำให้สามารถใช้มาตรฐานเดียวกันได้ ในการพัฒนาบนแพลตฟอร์ม และภาษาที่แตกต่างกันไปตามระบบต่างๆที่เข้าร่วม ซึ่งในบางครั้งการส่งข้อมูลแลกเปลี่ยนกัน ก็มีปัญหาในเรื่องของรูปแบบของข้อมูล เช่น กุญแจที่ใช้ในการทำลายมือชื่อดิจิตอล (Private Key and Public Key) เมื่อมีการสร้างกุญแจจากแพลตฟอร์มหนึ่ง และมีการนำไปใช้งานในอีกแพลตฟอร์มหนึ่ง อาจจะต้องมีการแปลงข้อมูลดังกล่าวให้สามารถทำงานได้ เป็นต้น และการพัฒนาให้สามารถใช้งานได้หลากหลายแพลตฟอร์ม และโครงสร้างพื้นฐานที่แตกต่างกัน ทำให้จำเป็นต้องมีความรู้ในภาษาต่างๆ และเข้าใจการทำงานของระบบพื้นฐานอย่างมากด้วย

5.4 ข้อเสนอแนะ

- ปรับปรุงการทำงานของระบบ ให้สามารถใช้งานร่วมกับแพลตฟอร์มและภาษาอื่นๆที่แตกต่างกันให้ได้มากขึ้น
- เพิ่มประสิทธิภาพในการจัดเก็บและจัดการกับบัญชีรายชื่อผู้ใช้งานให้มากขึ้น

บรรณานุกรม

บรรจง หะรังษี. **"ความรู้เบื้องต้นของการเข้ารหัสข้อมูล (Introduction to Cryptography)".** [Online]. Available: http://www.thaicert.nectec.or.th/paper/encryption/intro_crypt.php

AuthenticationWorld.com, **Single Sign-On.** [Online]. Available: http://www.authenticationworld.com/Single-Sign-On-Authentication/

Gang Zhao, Dong Zheng, Kefei Chen. "Design of single sign-on" **E-Commerce Technology for Dynamic E-Business**, 2004. IEEE International Conference on 13-15 Sept. 2004 Page(s): 253- 256.

Google. **SAML Single Sign-On (SSO) Service for Google Apps.** [Online]. Available: http://code.google.com/apis/apps/sso/saml_reference_implementation.html

Google. **Web-based Reference Implementation of SAML-based SSO for Google Apps.** [Online]. Available: http://code.google.com/apis/apps/sso/saml_reference_implementation_web.html

IBM Web Sites. **Portal composite pattern::Runtime pattern and Access Integration::Web Single Sign-On application pattern.** [Online]. Available: http://www-128.ibm.com/developerworks/patterns/portal/access-sso-runtime.html

Sun Microsystems, Inc.. **Java XML Digital Signature.** [Online]. Available: http://java.sun.com/developer/technicalArticles/xml/dig_signatures/

Sun Microsystems, Inc.. **Single Sign-On And Sessions**. [Online]. Available: http://docs.sun.com/source/816-6774-10/prog_sso.html

Web Services Interoperability Organization (WS-I). **Basic Security Profile Version 1.0.** [Online]. Available: http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html

Wikipedia. **Digital signature.** [Online]. Available: http://en.wikipedia.org/wiki/Digital_signature

Wikipedia. **Encryption.** [Online]. Available: http://en.wikipedia.org/wiki/Encryption

Wikipedia. **Http Cookie.** [Online]. Available: http://en.wikipedia.org/wiki/HTTP_cookie

Wikipedia. **Lightweight Directory Access Protocol**. [Online]. Available: http://en.wikipedia.org/wiki/Ldap

Wikipedia. **Single sign-on.** [Online]. Available: http://en.wikipedia.org/wiki/Single_sign-on

Wikipedia. **URL redirection.** [Online]. Available: http://en.wikipedia.org/wiki/HTTP_redirect

Wikipedia. **X.500**. [Online]. Available: http://en.wikipedia.org/wiki/X.500

ประวัติผู้เขียน

ชื่อ	นายกำพล ภักดีลิขิต
วันเกิด	6 เมษายน 2524
สถานที่เกิด	กรุงเทพมหานคร
ปริญญาตรี	คณิตศาสตร์
	คณะวิทยาศาสตร์
	มหาวิทยาลัย ศรีนครินทรวิโรฒ
การทำงาน	Lotus Notes Application Developer
	บริษัท เอ็ม.บี. ซิสเต็มส์ ออโตเมชั่น จำกัด
	Lotus Notes Programmer
	บริษัท โพรเกรส ซอฟท์แวร์ จำกัด