ระบบการจัดการผู้ใช้งานบัญชีผู้มีสิทธิ์สูง
Privileged Account Management

อิษณัย วงศ์สิทธิกร
Isanai Wongsittigorn

สารนิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครือข่ายและความมั่นคง
ปลอดภัยสารสนเทศ แขนงความมั่นคงปลอดภัยไซเบอร์ (Cyber Security)
คณะวิทยาการและเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีมหานคร
ปีการศึกษา 2562

หัวข้อ	ระบบการจัดการผู้ใช้งานบัญชีผู้มีสิทธิ์สูง Privileged Account Management
ชื่อนักศึกษา	อิษณัย วงศ์สิทธิกร
รหัสนักศึกษา	6017810024
หลักสูตร	วิทยาศาสตรมหาบัณฑิต สาขาวิศกรรมเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
ปีการศึกษา	2562
อาจารย์ที่ปรึกษา	ผศ.ดร.เอกรัฐ รัฐกาญจน์

บทคัดย่อ

ระบบบริหารจัดการบัญชีผู้มีสิทธิ์สูง (Privileged Account Management: PAM) พัฒนาขึ้นมาเพื่อเพิ่มความปลอดภัยให้กับองค์กรที่มีการใช้งาน Privileged Account ร่วมกันในแต่ละบัญชี ทำให้ผู้ดูแลระบบสามารถใช้บัญชีเดียวเข้าถึงได้ในหลายส่วน อาจทำให้เกิดช่องโหว่ด้านความปลอดภัย ประกอบกับระบบบริหารจัดการบัญชีผู้มีสิทธิ์สูงที่มีขายในท้องตลาดนั้นมีราคาสูง เมื่อเทียบกับขนาดขององค์กรและจำนวนระบบที่ไม่มากนัก

PAM จึงเป็นระบบที่ถูกออกแบบมาเพื่อลดข้อจำกัดดังกล่าว และเพิ่มความปลอดภัยให้กับองค์กร โดยจำกัดจำนวนผู้ดูแลระบบ (Admin) ป้องกันปัญหาเรื่องรหัสผ่านที่อ่อนแอ (Weak Passwords) ที่มีการตั้งรหัสผ่านเดิมซ้ำ ๆ ซึ่งง่ายต่อการคาดเดา รวมไปถึงป้องกันการใช้รหัสผ่านเป็นชุดเดียวกันในทุกระบบ โดย PAM จะเป็นระบบที่ควบคุมการเบิกจ่าย Privileged Account ในลักษณะของเว็บแอปพลิเคชัน เรียกใช้งานผ่านทางหน้าจอเว็บเบราว์เซอร์ซึ่งใช้ภาษา PHP ในการพัฒนาโปรแกรม และใช้ MySQL ในการจัดการฐานข้อมูล และนอกจากจะช่วยบริหารจัดการเรื่องบัญชีผู้มีสิทธิ์สูงแล้ว ยังช่วยลดขั้นตอนของการขออนุมัติสิทธิ์ เพื่อใช้งาน Privileged Account ให้รวดเร็วขึ้น และสามารถตรวจสอบย้อนหลังได้

กิตติกรรมประกาศ

สารนิพนธ์ฉบับนี้เกิดขึ้นได้ เพราะการแนะนำ ชี้แนะแนวทางให้แง่คิด มุมมองที่เป็นประโยชน์จากท่าน ผศ.ดร.เอกรัฐ รัฐกาญจน์ ผู้เป็นอาจารย์ที่ปรึกษา และ ดร.นันทา จันทร์พิทักษ์ ซึ่งได้เสนอแนะแนวทางในการดำเนินโครงการ รวบรวม แก้ไข และตรวจสอบข้อบกพร่องในระหว่างการจัดทำสารนิพนธ์

ขอขอบพระคุณคณาจารย์ทุกท่านในคณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร ที่ได้ประสิทธิ์ประสาทวิชาความรู้ ทักษะ เทคนิคต่าง ๆ เพื่อใช้ประกอบในการจัดทำสารนิพนธ์ฉบับนี้

สุดท้ายนี้ข้าพเจ้าขอขอบพระคุณกำลังใจจากครอบครัว และเพื่อน ๆ ที่คอยช่วยเหลือสนับสนุน และให้กำลังใจในการทำสารนิพนธ์อยู่เสมอ ทำให้ข้าพเจ้ามีกำลังใจที่จะพัฒนาโครงงานจนสำเร็จลุล่วงได้

อิษณัย วงศ์สิทธิกร

มีนาคม 2562

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

หน้า

สารบัญรูป (ต่อ)

หน้า

สารบัญตาราง

หน้า

บทที่ 1
บทนำ

1.1 ปัญหาและแรงจูงใจ

เนื่องจากรหัสผ่านเป็นส่วนสำคัญในการเข้าใช้งานระบบงานต่าง ๆ เป็นอย่างมาก เพราะรหัสผ่านถือเป็นสิ่งที่ใช้สำหรับยืนยันตัวตนในการเข้าใช้งานระบบ เพื่อให้ User (ผู้ใช้งาน) ที่มีสิทธิ์สามารถเข้าไปใช้งานระบบ หรือเข้าไปจัดการระบบ องค์กรมีผู้ดูแลระบบจำนวนมากซึ่งมักจะมีการแชร์ Privileged Account เพื่อใช้งานร่วมกัน เช่น บัญชี Admin ของเซิร์ฟเวอร์หรืออุปกรณ์ IT ทางฝั่งเน็ตเวิร์ครวมไปถึงแอพพลิเคชันต่างๆ ซึ่งทำให้ผู้ดูแลระบบสามารถใช้บัญชีเดียวเข้าถึงทุกระบบได้ในหลายส่วน อาจทำให้เกิดช่องโหว่ด้านความปลอดภัยขึ้นกับองค์กร

ซึ่ง Privileged Account เหล่านี้ไม่ได้ถูกบริหารจัดการอย่างถูกต้อง อาจก่อให้เกิดผลเสียต่อองค์กรได้ในกรณีที่ผู้ไม่ประสงค์ดีนำบัญชีดังกล่าวไปใช้งานในทางที่ผิด ยิ่งในระบบงานที่มีความสำคัญมาก ๆ และหากระบบนั้นอยู่ในระดับชั้นความเสี่ยงที่มีผลกระทบสูง หากมีการรั่วไหลของรหัสผ่านหรือบริหารจัดการการเบิกจ่ายรหัสผ่านไม่ดี ทำให้เกิดการเข้าถึงระบบงานที่สำคัญจากผู้ไม่ประสงค์ดีที่ต้องการนำข้อมูลที่มีความสำคัญและเป็นความลับออกไปเผยแพร่ หรือกระทำการใด ๆ ที่ไม่ถูกต้องกับระบบงาน ก่อให้เกิดผลเสียทั้งในส่วนของตัวบุคคลและชื่อเสียงขององค์กร และไม่สามารถตรวจสอบได้ว่าใครเป็นผู้ไม่ประสงค์ดีดังกล่าว เนื่องจากมีการแชร์ Privileged Account ประกอบกับระบบบริหารจัดการบัญชีผู้มีสิทธิ์สูงที่มีขายในท้องตลาดนั้นมีราคาสูง เมื่อเทียบกับขนาดขององค์กรและจำนวนระบบที่ไม่มากนัก รวมไปถึงการร้องขอการใช้งาน Privileged Account เพื่อเข้าไปใช้งานระบบที่เกี่ยวข้อง การเบิกสิทธิ์ Privileged Account นั้น จะถูกสร้าง (Generate) จากผู้บริหารระบบที่ทางผู้ใช้งานร้องขอจากนั้นนำรหัสผ่านที่ถูกสร้างขึ้นมาใหม่ใส่ซองจดหมาย และมอบให้กับผู้ที่ร้องขอการใช้งานต่อไป

จากกระบวนการข้างต้น ทางผู้ศึกษามองเห็นปัญหาในเรื่องการเบิกสิทธิ์ที่มีความล่าช้า และกระบวนการทำงานในการรับรหัสผ่านจากผู้บริหารระบบ ดังนั้นการบริหารจัดการ และการควบคุมการเบิกสิทธิ์ (Privileged Account Management) เพื่อนำไปใช้งานจึงเป็นเรื่องสำคัญเป็นอย่างยิ่งที่จะต้องดำเนินการควบคุม เพื่อป้องกันการเข้าถึงโดยผู้ที่ไม่มีสิทธิ์ และสามารถตรวจสอบได้ว่าใครเป็นผู้ใช้งาน Privileged Account นั้น ผู้ศึกษาได้มีแนวคิดในการพัฒนาระบบที่สามารถช่วยบริหารจัดการ Privileged Account ในการเข้าใช้งานระบบงานที่มีความสำคัญให้ปลอดภัย และสามารถตรวจสอบการเบิกจ่ายการนำไปใช้งานโดยอาศัยหลักการ บริหารจัดการรหัสผ่าน (Password Management) ในการควบคุมรหัสผ่าน และสิทธิ์ของชื่อผู้ใช้ที่มีสิทธิ์ในการจัดการระบบ เพื่อเพิ่มความปลอดภัยให้กับองค์กร

1.2 แนวทางการแก้ปัญหา

นำเสนอระบบการบริหารจัดการ Privileged Account เพื่อกำหนดหนดรหัสผ่านของ Privileged Account ของระบบที่ถูกร้องขอจากผู้ใช้งาน และสามารถควบคุมตรวจสอบการเข้าใช้งานระบบต่างๆ ที่สำคัญภายในหน่วยงานหรือองค์กร

1.3 วัตถุประสงค์

1.3.1 เพื่อป้องกันการใช้ Static Password หรือรหัสผ่านเดียวโดยไม่มีการเปลี่ยนแปลง

1.3.2 เพื่อป้องกันการใช้รหัสผ่านของ Privileged Account ที่คาดเดาง่าย

1.3.3 เพื่อป้องกันการใช้รหัสผ่านเดียวกันในหลายๆ ระบบ

1.3.4 เพื่อป้องกันการจัดเก็บรหัสผ่านในรูปแบบการบันทึกไว้ตามที่ต่างๆ

1.3.5 เพื่อป้องกันผู้รับจ้างงานที่หมดวาระ หรือพนักงานที่พ้นสภาพ มิให้สามารถเข้าถึงการใช้งานระบบของหน่วยงานหรือองค์กรได้จากรหัสผ่านที่มีอยู่

1.3.6 เพื่อบันทึกว่าผู้ใดเป็นผู้ใช้งาน Privileged Account ในช่วงเวลานั้น ๆ

1.4 ขอบเขตของโครงงาน

การพัฒนาระบบแบ่งออกเป็น 2 ส่วนคือ

1) ส่วน Front-End ใช้สำหรับการติดต่อเพื่อรับข้อมูลจากผู้ใช้งาน มีขั้นตอนการดำเนินงานดังนี้

   1.1) ขอรหัสผ่าน ของ Privileged Account เช่น Root หรือ Administrator ในการตั้งค่าระบบต่าง ๆ โดยจะต้องได้รับการอนุมัติจากผู้บริหารระบบก่อน

   1.2) การเพิ่ม Script ชุดคำสั่งสำหรับส่งข้อมูลไปยังระบบบริหารจัดการรหัสผ่าน

   1.3) ระบบสามารถตรวจสอบ ผู้ใช้งานที่เข้ามาทำการล็อกอิน ณ เวลาที่ต้องการตรวจสอบ

2) ส่วน Back-End

   2.1) ระบบบริหารจัดการรหัสผ่าน ดำเนินโดยมีขั้นตอน การสร้างรหัสผ่านใหม่ เปลี่ยนรหัส และนำส่งรหัสดังกล่าวนี้ให้กับผู้ใช้งานที่มีการร้องขอ

   2.2) ระบบฐานข้อมูลในการจัดเก็บรหัสผ่านที่มีการสร้างขึ้นให้กับผู้ใช้งานที่ได้ทำการร้องขอการเข้าถึงระบบงานที่สำคัญภายในหน่วยงานหรือองค์กร

บทที่ 2
พื้นฐานและทฤษฎีที่เกี่ยวข้อง

2.1 การบริหารจัดการบัญชีผู้มีสิทธิ์สูง (Privileged Account Management)

การจัดการสิทธิ์สามารถดำเนินการได้หลากหลายวิธี ซึ่งทุกวิธีมีเป้าหมายที่เหมือนกันคือการพยายามจำกัดสิทธิ์ในการใช้งานให้น้อยที่สุด โดยจะระบุในรูปแบบของข้อจำกัดสิทธิ์ในการเข้าถึงและการได้รับอนุญาตของผู้ใช้งาน บัญชีผู้ใช้ แอปพลิเคชัน ระบบ อุปกรณ์ (เช่น IoT) และโปรแกรมประมวลผล เพื่อระบุความจำเป็นทั้งหมดในการปฏิบัติงาน และตามที่ได้รับอนุญาต ซึ่งในสารนิพนธ์ที่จัดทำขึ้นนี้ได้ใช้วิธีการบริหารจัดการบัญชีผู้มีสิทธิ์สูง (Privileged Account Management)

2.1.1 Privileged คืออะไร

Privilege ในบริบทของเทคโนโลยีสารสนเทศ คือการให้การอนุญาตกับบัญชีผู้ใช้ หรือโปรแกรมประมวลผลในระบบคอมพิวเตอร์ หรือระบบเครือข่าย โดยทำการอนุญาตให้ไม่ต้องตรวจสอบข้อจำกัดของระบบความปลอดภัยนั้นๆ และอาจรวมถึงอำนาจในการดำเนินการ เช่น การปิดระบบ นำเข้าโปรแกรมสนับสนุนการทำงานของอุปกรณ์ ตั้งค่าระบบหรือระบบเครือข่าย จัดการ/ตั้งค่าบัญชีผู้ใช้ เป็นต้น

การกำหนดสิทธิ์เป็นขั้นตอนสำคัญเพื่อทำให้ผู้ใช้งาน แอปพลิเคชัน และระบบประมวลผลอื่นๆ มีสิทธิ์ที่สูงขึ้นในการเข้าใช้ทรัพยากร และปฏิบัติงานได้สำเร็จ ในขณะเดียวกันความพยายามที่จะใช้งานสิทธิ์ในทางที่ผิดทั้งโดยบุคคลภายในหรือผู้โจมตีภายนอก ทำให้องค์กรต้องพบกับความเสี่ยงทางความมั่นคงปลอดภัยที่ยากต่อการจัดการ

2.1.2 Privileged Account คืออะไร

Privileged Account คือบัญชีผู้ใช้งานที่ได้รับสิทธิ์เหนือกว่าบัญชีผู้ใช้งานทั่วไป (Non-Privileged Account) ซึ่งผู้ใช้งานที่ได้รับสิทธิ์ (Privileged User) คือผู้ใช้งานที่สามารถใช้งานผ่านบัญชีผู้ใช้ที่ได้รับสิทธิ์ เนื่องจากขีดความสามารถ และการเข้าถึงที่ได้รับการยกระดับให้สูงขึ้น ผู้ใช้งานบัญชีที่ได้รับสิทธิ์ จะแสดงให้เห็นได้ชัดว่าบัญชีเหล่านี้มีความเสี่ยงสูงกว่าผู้ใช้งานบัญชีที่ไม่ได้รับสิทธิ์ ดังนั้นการบริหารจัดการบัญชีผู้มีสิทธิ์สูง ถูกพัฒนาขึ้นมาเพื่อเพิ่มความปลอดภัยให้กับองค์กรที่มีการใช้งาน Privilege Account

บัญชีผู้มีสิทธิ์สูง หรือเรียกว่า Superuser จะถูกใช้งานโดยผู้ที่ได้รับมอบหมาย เพื่อใช้บริหารและจัดการในการสั่งปฏิบัติการชุดคำสั่ง หรือแก้ไขค่าต่างๆในระบบ บัญชีผู้มีสิทธิ์สูง Superuser โดยทั่วไปจะถูกเรียกว่า “Root” ในระบบปฏิบัติการ Unix และ Linux และเรียกว่า “Administrator” ในระบบปฏิบัติการ Windows

บัญชีผู้มีสิทธิ์สูง นั้นสามารถเข้าถึงไฟล์ แฟ้ม และทรัพยากร ต่างๆบนเครื่องรวมทั้งสามารถอ่าน เขียน ดำเนินการ และสามารถแสดงการเปลี่ยนแปลงที่เกิดขึ้นในระบบเครือข่าย เช่น การสร้างไฟล์ ติดตั้งซอฟต์แวร์ แก้ไขไฟล์ การตั้งค่า การลบผู้ใช้งานและข้อมูล บัญชีผู้มีสิทธิ์สูงยังสามารถกำหนดสิทธิ์ และเพิกถอนสิทธิ์สำหรับผู้ใช้งานอื่นๆ หากบัญชีผู้มีสิทธิ์สูง ถูกนำไปใช้งานในทางที่ผิดอาจส่งผลให้เกิดความเสียหายเนื่องจากความผิดพลาด (เช่น ทำการลบไฟล์ที่สำคัญโดยบังเอิญ หรือพิมพ์คำสั่งผิดพลาดซึ่งส่งผลกระทบต่อระบบ) หรือมีเจตนาที่ไม่ดีต่อระบบ บัญชีผู้ใช้ที่ได้รับสิทธิ์สูงนี้สามารถก่อให้เกิดความเสียหายทั้งระบบ

เนื่องจากบัญชีผู้มีสิทธิ์สูงใช้สำหรับบริหารจัดการได้รับการกำหนดสิทธิ์ที่มากกว่าบัญชีใช้งานทั่วๆไป ตามแนวทางปฏิบัติของ Privileged Account Management คือการใช้งานบัญชีผู้ใช้สำหรับบริหารจัดการต่อเมื่อมีความจำเป็นต้องใช้งานจริงๆเท่านั้น และควรใช้งานในระยะเวลาที่เหมาะสม

2.1.3 ความเสี่ยงและภัยคุกคามจากการกำหนดสิทธิ์

1) ความไม่ตระหนักถึงบัญชีผู้ใช้ บัญชีผู้ใช้ที่ถูกสร้างไว้เป็นระยะเวลานาน และถูกแชร์ให้สามารถใช้งานอย่างแพร่หลายภายในองค์กรโดยไม่ได้รับการจัดการอย่างเหมาะสม บัญชีเหล่านี้อาจมีจำนวนมากและเป็นช่องทางให้กับผู้โจมตีระบบ ตัวอย่างที่เกิดขึ้นบ่อยครั้ง เช่น พนักงานที่ลาออกไปแล้ว แต่ผู้ดูแลระบบไม่ได้ทำการลบหรือปิดบัญชีของพนักงานผู้นั้น
2) การให้สิทธิ์มากเกินความจำเป็น หากสิทธิ์ที่ได้รับนั้นจำกัดมากเกินไป อาจทำให้ปฏิบัติงานได้ไม่ราบรื่น เกิดความยากลำบากในการปฏิบัติงาน ลดทอนประสิทธิภาพในการทำงาน ซึ่งอาจเกิดจากพนักงานมีการปรับเปลี่ยนตำแหน่งหน้าที่ซึ่งมีความรับผิดชอบที่แตกต่างออกไป และได้รับสิทธิ์ที่สอดคล้องกับงานใหม่นั้น ในขณะที่สิทธิ์เดิมที่เคยได้รับก่อนหน้าก็ยังคงอยู่ สิทธิ์ที่มากเกินความจำเป็นจะเพิ่มความเสี่ยงที่ผู้เจาะระบบจะขโมยรหัสผ่าน หรือติดตั้งโปรแกรมที่อันตรายต่อระบบ เพื่อที่จะเข้าถึงข้อมูลในเครื่องเป้าหมาย หรือแม้กระทั่งโจมตีเซิร์ฟเวอร์ หรือคอมพิวเตอร์เครื่องอื่นๆในระบบเครือข่าย
3) การใช้บัญชีผู้ใช้และรหัสผ่านร่วมกัน ผู้ดูแลระบบมักจะใช้งานบัญชี Root หรือบัญชี Administrator ร่วมกัน รวมไปถึงรหัสผ่านเพื่อความสะดวก ทำให้สามารถทำงานได้ราบรื่นมากขึ้น อย่างไรก็ตาม เมื่อผู้ใช้งานหลายคนใช้บัญชีและรหัสผ่านร่วมกัน ทำให้ไม่สามารถตรวจสอบได้ว่าการดำเนินการต่างๆที่เกิดขึ้นบนบัญชีผู้ใช้นั้น ถูกกระทำโดยผู้ใช้งานคนใด ทำให้เกิดปัญหาในด้านความมั่นคงปลอดภัยความเสี่ยงและภัยคุกคามจากการกำหนดสิทธิ์

2.1.4 ประโยชน์ของการจัดการบัญชีผู้มีสิทธิ์สูง

ยิ่งผู้ใช้งานได้รับสิทธิ์ในการใช้งานระบบมากเท่าไร โอกาสที่จะถูกภัยคุกคาม และทำให้เกิดข้อผิดพลาด ก็มีมากขึ้นเช่นกัน การบริหารจัดการบัญชีผู้มีสิทธิ์สูงจึงไม่เพียงแต่ลดความเสี่ยงในการเกิดช่องโหว่ แต่ยังช่วยในเรื่องของการจำกัดขอบเขตของการเกิดช่องโหว่ได้ ซึ่งการบริหารจัดการบัญชีผู้มีสิทธิ์สูงมีประโยชน์ดังนี้

1) การจัดการบัญชีผู้มีสิทธิ์สูงสามารถลดขอบเขตของการโจมตีทั้งจากภายในและภายนอกได้
2) ทำให้การปฏิบัติและพิสูจน์ผลตามข้อบังคับง่ายขึ้น ด้วยการควบคุมการเข้าถึงตามที่ได้รับสิทธิ์ การจัดการบัญชีผู้มีสิทธิ์สูงช่วยทำให้การใช้งานของระบบซับซ้อนน้อยลง และสามารถตรวจสอบได้ง่าย
3) ป้องกันผู้ไม่ประสงค์ดีนำบัญชีผู้มีสิทธิ์สูงไปใช้งานในทางที่ผิด ยิ่งในระบบงานที่มีความสำคัญมาก ๆ และหากระบบนั้นอยู่ในระดับชั้นความเสี่ยงที่มีผลกระทบสูง หากมีการรั่วไหลของรหัสผ่าน หรือบริหารจัดการการเบิกจ่ายรหัสผ่านไม่ดี ทำให้เกิดการเข้าถึงระบบงานที่สำคัญจากผู้ไม่ประสงค์ดี ที่ต้องการนำข้อมูลที่มีความสำคัญและเป็นความลับออกไปเผยแพร่ หรือกระทำการใด ๆ ที่ไม่ถูกต้องกับระบบงาน

2.1.5 แนวทางปฏิบัติในการจัดการบัญชีผู้มีสิทธิ์สูง

เนื่องจากบัญชีผู้มีสิทธิ์สูงมีผลกระทบต่อระบบมาก จึงต้องมีการควบคุมบางอย่างเพื่อจำกัดหรือตรวจสอบการเข้าใจงานบัญชีเหล่านี้ ดังนั้นแนวทางปฏิบัติเบื้องต้นที่ใช้ควบคุมบัญชีผู้มีสิทธิ์สูง มีดังนี้

1) ไม่อนุญาตให้ผู้ดูแลระบบทำการแชร์บัญชีผู้มีสิทธิ์สูง โดยใช้บัญชีของผู้ที่ได้รับมอบหมายโดยให้สิทธิ์สูง เพื่อทำให้ผู้ใช้งานไม่สามารถปฏิเสธความรับผิดชอบต่อการกระทำของผู้ใช้นั้นๆ
2) จำกัดจำนวนบัญชีผู้มีสิทธิ์สูงให้น้อยที่สุด ผู้ดูแลระบบแต่ละคนควรมีบัญชีที่มีสิทธ์สูงบัญชีเดียวสำหรับการทำงานทุกระบบ
3) กำหนดนโยบายการตั้งรหัสผ่านและบังคับใช้อย่างเคร่งครัด ตามวิธีปฏิบัติในการเพิ่มความปลอดภัยในการตั้งรหัสผ่าน ซึ่งจะกล่าวในเนื้อหาถัดไป
4) หากผู้ใช้งานต้องการเข้าถึงบัญชีผู้มีสิทธิ์สูง ต้องทำตามขั้นตอนการร้องขอและการอนุมัติเอกสารไม่ว่าจะเป็นทางกระดาษ หรือระบบที่องค์กรได้จัดทำขึ้น และเข้าใช้งานเฉพาะในช่วงเวลาที่ได้ระบุไว้

5) ตรวจสอบและเก็บบันทึกกิจกรรมการใช้งานบัญชีผู้มีสิทธิ์สูง เช่นการเข้าสู่ระบบ การออกจากระบบ การดำเนินการอื่นๆของผู้ใช้งานที่ได้รับสิทธิ์

2.2 การเพิ่มความปลอดภัยในการตั้งรหัสผ่าน (Password)

รหัสผ่าน เป็นด่านแรกในการป้องกันการเข้าถึงข้อมูลบัญชีผู้ใช้งานหรือ ข้อมูลภายในองค์กรโดยไม่ได้รับอนุญาต ดังนั้น การตั้งรหัสผ่านที่ไม่ดี จะทำให้ผู้ที่โจมตี สามารถเข้าสู่ระบบคอมพิวเตอร์และเครือข่ายได้โดยง่าย ในขณะที่การตั้งรหัสผ่าน ที่ดีพอ จะยากต่อการแกะรอยแม้จะใช้ซอฟต์แวร์เพื่อการแกะหรือ ถอดรหัสผ่าน ที่มีการพัฒนาอย่างต่อเนื่องและมีประสิทธิ์ภาพสูงขึ้นก็ตาม หรืออาจต้องใช้เวลานานมาก

รหัสผ่านเป็นส่วนหนึ่งที่มีความสำคัญในการรักษาความปลอดภัยของบัญชีผู้ใช้งานหรือ ในระบบที่ต้องการความปลอดภัย ซึ่งรหัสผ่านถือเป็นสิ่งที่ใช้สำหรับยืนยันความถูกต้องของตัวบุคคลนั้นๆ การใช้งานรหัสผ่านจึงช่วยป้องกันความปลอดภัย การเข้าถึงข้อมูลโดยมิชอบนั้นได้ หากผู้ใช้งานไม่ให้ความสำคัญในการตั้งค่ารหัสผ่านก็จะทำให้ผู้ไม่หวังดีสามารถคาดเดารหัสผ่านและเข้าถึงข้อมูลของท่านได้อย่างง่ายดาย

ลักษณะของรหัสผ่านที่ปลอดภัย

1) ใช้อักขระมีความยาวไม่น้อยกว่า 8 ตัวอักษร
2) ไม่ใช้คำใดๆ ก็ตามที่เข้าข่ายสิ่งที่ไม่ควรนำมาใช้เป็นรหัสผ่าน
3) ไม่ใช้รหัสผ่านซ้ำกับ รหัสผ่านที่เคยใช้ไปแล้วในระบบงานอื่นๆ
4) ควรประกอบด้วยอักขระอื่นๆ หลายๆ ตัว ผสมกันอยู่ในรหัสผ่าน ได้แก่ ตัวอักษรพิมพ์เล็ก (a-z), ตัวอักษรพิมพ์ใหญ่ (A-Z), ตัวเลข (0-9) และตัวอักขระพิเศษ (!@#$%^&*()_+|~-=\`{}[]:";'<>?,./)
5) ใช้คำที่ไม่มีในพจนานุกรม

สิ่งที่ไม่ควรนำมาใช้เป็นรหัสผ่าน

1) ข้อมูลที่ใช้ในการระบุตัวตนทั่วไป อย่างเช่น ชื่อ นามสกุล เลขบัตรประจำตัวต่างๆ หรือวันเดือนปีเกิด
2) ข้อมูลการติดต่อ อย่างเช่น เบอร์โทรศัพท์
3) ชื่อบุคคลรอบข้างหรือ สัตว์เลี้ยง
4) ใช้คำที่มีความหมายและหาได้ในพจนานุกรม
5) คำทั่วๆไปที่มีการสะกดจากหลังไปหน้า อย่างเช่น admin -> nimda, root -> toor
6) ใช้รูปแบบตัวอักษรหรือตัวเลขที่เป็นที่นิยม อย่างเช่น qwerty, 12345, 123321

7) ใช้รูปแบบการตั้งรหัสผ่านที่คล้ายคลึงกันในแต่ละบัญชี อย่างเช่น secret1, 1secret, secret?

นอกเหนือจากหลักการตั้งรหัสผ่านที่ดีพอแล้ว พฤติกรรมของผู้ใช้ยังเป็นปัจจัยในการเสริมความเข้มแข็งของรหัสผ่าน ดังนั้น การสร้างลักษณะนิสัยในการใช้งานรหัสผ่านที่ดี มีดังนี้

1) เมื่อใช้งานผ่านคอมพิวเตอร์ ไม่ควรเลือกฟังก์ชั่นจำรหัสผ่านอัตโนมัติ ต้องกรอกรหัสผ่านเองทุกครั้ง
2) เปลี่ยนรหัสผ่านทุกๆ 30 – 45 วัน
3) ในแต่ละบัญชีควรมีการตั้งรหัสผ่านที่แตกต่างกัน ไม่ควรใช้รหัสผ่านเดิม
4) ตรวจสอบการเข้าถึงบัญชีเป็นประจำ
5) ออกจากระบบทุกครั้งหลังใช้งาน
6) ไม่เขียนรหัสผ่านไว้บนกระดาษและ แปะไว้ตามที่ต่างๆ เพื่อเตือนความจำ รวมถึงไม่เก็บรหัสผ่านไว้ในรูปแบบของไฟล์ในคอมพิวเตอร์
7) เมื่อจำเป็นต้องทำธุรกรรมออนไลน์ผ่านคอมพิวเตอร์สาธารณะ ให้เปลี่ยนรหัสผ่านทันทีเมื่อมีโอกาส

2.3 Secure Shell (SSH)

Secure Shell คือ โพรโทคอล (Protocol) เครือข่ายที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์บนระบบเครือข่าย โดยอาศัยช่องทางที่มีความปลอดภัย (Secure Channel) ที่ทำงานผ่าน TCP Port 22 ซึ่งโพรโทคอล Secure Shell มีวัตถุประสงค์หลักเพื่อให้ผู้ใช้งานสามารถเข้าควบคุมหรือสั่งการเครื่องคอมพิวเตอร์ที่ให้บริการ Secure Shell ตามสิทธิของผู้ใช้งานซึ่งได้มาจากการพิสูจน์ตัวตน (Authentication) ด้วยการล็อกอิน (Login) โดยผ่านช่องทางการสื่อสารที่มีการรักษาความมั่นคงปลอดภัยด้วยการเข้ารหัสลับของข้อมูล (Encryption) ซึ่งถูกออกแบบมาเพื่อใช้แทนที่การสื่อสารข้อมูลบนระบบเครือข่ายที่ส่งข้อมูล แบบไม่ได้เข้ารหัสลับ (Plaintext) เช่น Telnet, Rlogin หรือ FTP ปัจจุบันโพรโทคอล SSH มีสองเวอร์ชั่นคือ SSH-1 และ SSH-2 (ถูกพัฒนาจาก SSH-1 เพื่อแก้ไขช่องโหว่หรือข้อผิดพลาดที่ทำให้ผู้โจมตีสามารถโจมตีเข้ามายัง เครื่องคอมพิวเตอร์ที่ให้บริการ SSH ได้)

การทำงานของโพรโทคอล Secure Shell ทำงานในลักษณะไคลเอนต์และเซิร์ฟเวอร์ (Client-Server) โดยมีโปรแกรมใช้งาน 2 ส่วนคือ โปรแกรมส่วนที่ทำหน้าที่เป็นเครื่องที่ให้บริการ (Server) จะถูกติดตั้งลงที่เครื่องคอมพิวเตอร์ที่ต้องการให้บริการ Secure Shell เช่น โปรแกรม OpenSSH-Server บนระบบปฏิบัติการ Linux และโปรแกรมอีกส่วนจะทำหน้าที่เป็นผู้เชื่อมต่อ (Client) ไปยังเครื่องคอมพิวเตอร์ที่ให้บริการ Secure Shell เช่น โปรแกรม PuTTY บนระบบปฏิบัติการ Windows หรือ โปรแกรม OpenSSH-Client บนระบบปฏิบัติการ Linux

ถึงแม้โพรโทคอล Secure Shell จะมีข้อดีในเรื่องของการรักษาความมั่นคงปลอดภัยโดยมีการเข้ารหัสลับข้อมูล และมีการล็อกอินก่อนการเข้าใช้งาน แต่มีโอกาสสูงที่จะถูกโจมตีจากผู้ไม่หวังดี โดยลักษณะการโจมตีที่เกิดขึ้นมักจะมาจากการใช้เทคนิคในการเข้าโจมตีที่เครื่องคอมพิวเตอร์ที่ให้บริการโดยตรง เช่น การโจมตีด้วยวิธีการสุ่มรหัสผ่าน (Brute-force) เพื่อพยายามเข้าสู่ระบบเครื่องคอมพิวเตอร์ที่ให้บริการ Secure Shell ซึ่งหากผู้ใช้งานหรือผู้ดูแลระบบตั้งค่ารหัสผ่านในการล็อกอินง่ายเกินไปจะทำให้โอกาสในการโจมตีสำเร็จได้ง่ายมากยิ่งขึ้น ซึ่งวิธีการป้องกันที่ผู้ดูแลระบบส่วนใหญ่นิยมใช้ คือ การเปลี่ยนวิธีการล็อกอินจากวิธีการปกติที่ใช้รหัสผ่าน เป็นการใช้ Key Authentication ซึ่งเป็นรูปแบบการเข้ารหัสแบบอสมมาตร (Asymmetric-key cryptography) โดยมีการสร้างคู่กุญแจ ซึ่งจะประกอบไปด้วยกุญแจสาธารณะ (Public Key) และ กุญแจส่วนตัว (Private Key) มีหลักการทำงานคือ หากใช้กุญแจ A ในการเข้ารหัสลับ จะต้องใช้กุญแจ B ในการถอดรหัสลับ โดยการเข้ารหัสและถอดรหัสดังกล่าวจะใช้ฟังก์ชันทางคณิตศาสตร์เข้ามาช่วยซึ่งการใช้หลักการดังกล่าวในการพิสูจน์ตัวตนของผู้ใช้งานกับเครื่อง คอมพิวเตอร์ที่ให้บริการ Secure Shell จะช่วยป้องกันการโจมตีด้วยวิธีการสุ่มรหัสผ่านจากผู้ที่ไม่หวังดีได้

2.4 Authentication

การพิสูจน์ตัวตน (Authentication) คือ การยืนยันตัวตนสามารถระบุผู้ที่เข้ามาในระบบนั้นคือใคร และเป็นคนๆ นั้นจริงหรือไม่ เช่น การเข้าสู่ระบบอีเมล (Email), การเข้าสู่ระบบ Internet Banking (อินเตอร์เน็ตแบงค์กิ้ง) เป็นต้น ซึ่งล้วนแล้วต้องมีการยืนยันตัวตนก่อนที่จะเข้าใช้งานได้ ไม่ว่าจะเป็นการใช้ Username และ Password ที่เราเป็นคนกำหนดเองหรือจะเป็นการใช้หมายเลขบัตรประจำตัวประชานชนของเราเอง จุดประสงค์หลักของการ Authentication คือพิสูจน์ตัวบุคคลพร้อมทั้งทำการตรวจสอบสิทธิ์ว่าผู้ใช้งานระบบนั้นมีสิทธิ์ใช้ได้และเป็นเจ้าของข้อมูลเหล่านั้นจริง ๆ

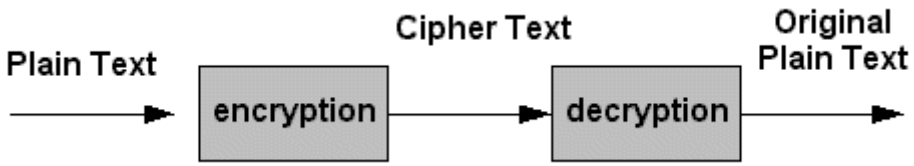
การพิสูจน์ตัวตน ในทางปฏิบัติสามารถแบ่งออกเป็น 2 ขั้นตอน ได้แก่

1) การระบุตัวตน (Identification) คือการตรวจสอบหลักฐานเพื่อให้สามารถระบุได้ว่าตนเองคือใคร เช่น Username
2) การพิสูจน์ตัวตน (Authentication) คือการตรวจสอบหลักฐานเพื่อพิสูจน์ว่าเป็นบุคคลที่พูดถึงจริง ซึ่งคุณลักษณะของการพิสูจน์ตัวตนนั้นสามารถแบ่งออกได้เป็น 3 ประเภท คือ
    2.1) สิ่งที่คุณมี (Something you have) เช่น บัตรประชาชน บัตรเครดิต เป็นต้น
    2.2) สิ่งที่คุณรู้ (Something you know) เช่น รหัสผ่าน PIN เป็นต้น
    2.3) สิ่งที่คุณเป็น (Something you are) เช่น ลายนิ้วมือ ม่านตา เป็นต้น

ในปัจจุบันได้มีการนำแต่ละคุณลักษณะมาใช้ร่วมกันเป็น Multi-factor authentication เพื่อเพิ่มประสิทธิภาพในการรักษาความปลอดภัยของข้อมูล

2.5 การเข้ารหัสลับ (Cryptography)

เทคโนโลยีการเข้ารหัสลับนั้น เป็นศาสตร์ที่มีมาตั้งแต่สมัยโบราณถูกพัฒนาจากแนวคิดเกี่ยวกับ พื้นฐานในการรักษาความปลอดภัยของข้อมูลอิเล็กทรอนิกส์ และพัฒนาต่อเนื่องมาจนถึงปัจจุบัน ซึ่งเป็นการประยุกต์โดยการนำหลักการทางคณิตศาสตร์มาใช้ในการเข้ารหัสลับ (Cryptographic Algorithms) โดยการสร้างสิ่งที่อยู่ในรูปตัวอักษร อักขระ ตัวเลข หรือสัญลักษณ์ใดๆ ขึ้นมา และเรียกสิ่งนั้นว่า กุญแจ (Key) และใช้กุญแจเป็นกลไกสำคัญในการแปลงข้อมูลอิเล็กทรอนิกส์ที่อ่านได้ (Plain Text) เป็นข้อมูลอิเล็กทรอนิกส์ที่อ่านไม่ได้ (Cipher Text) จะเรียกกระบวนการนี้ว่า การเข้ารหัสลับ (Encryption) และการแปลงข้อมูลอิเล็กทรอนิกส์กลับให้อยู่ในรูปของข้อมูลอิเล็กทรอนิกส์ที่อ่านได้เรียกว่า การถอดรหัสลับ (Decryption)



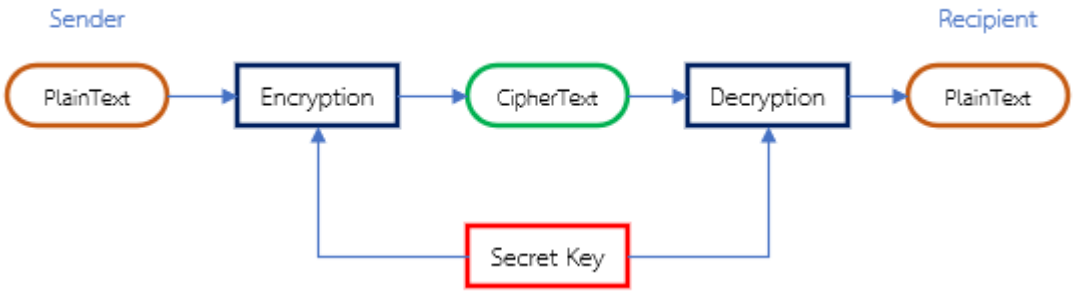
รูปที่ 2.1 กระบวนการเข้ารหัส และถอดรหัส

จุดมุ่งหมายของความปลอดภัยในระบบคอมพิวเตอร์ และเป็นคุณสมบัติที่ดีของวิทยาการเข้ารหัสลับมี 4 คุณสมบัติ ดังนี้

1) การพิสูจน์ตัวตน (Authentication) หมายถึงการพิสูจน์ตัวตนของผู้ที่ทำการเข้ามาถึงข้อมูลภายในระบบได้
2) การรักษาความลับ (Confidentiality) คือการที่ข้อมูลจะไม่ถูกเปิดเผยโดยผู้ที่ไม่ได้รับอนุญาต
3) ความถูกต้องของข้อมูล (Integrity) คือการทำให้มั่นใจว่าข้อมูลที่อยู่ในระบบหรือข้อมูลที่ถูกส่งออกไปยังเครือข่าย มีความถูกต้องและสมบูรณ์ ไม่ถูกแก้ไขเปลี่ยนแปลงโดยผู้ที่ไม่ได้รับอนุญาต
4) การป้องกันการปฏิเสธความรับผิดชอบ (Non-repudiation) คือการที่ผู้ส่งข้อมูลไม่สามารถปฏิเสธได้ว่า ข้อมูลที่ได้ทำการส่งไป ตนเองไม่ได้ส่ง

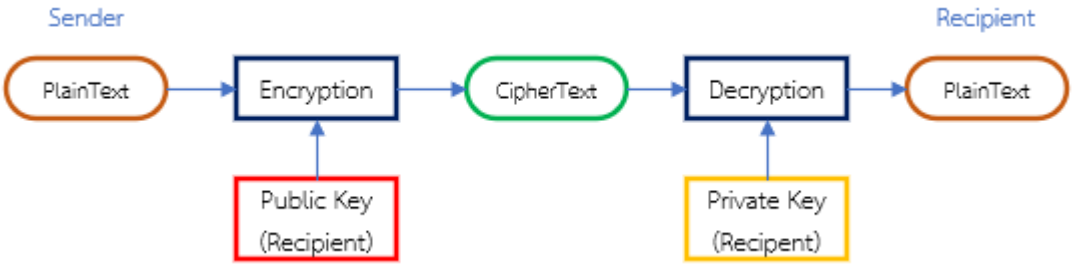
การเข้ารหัสลับแบ่งออกเป็น 2 ประเภท คือ การเข้ารหัสลับแบบสมมาตร (Symmetric Cryptography) และการเข้ารหัสลับแบบอสมมาตร (Asymmetric Cryptography)

1) การเข้ารหัสลับแบบสมมาตร เป็นการเข้ารหัสลับและถอดรหัสลับโดยใช้กุญแจชุดเดียวกัน (Symmetric key) คือทั้งผู้รับและผู้ส่งข้อความจะมีกุญแจชุดที่เหมือนกันเพื่อใช้ในการเข้ารหัสลับและถอดรหัสลับข้อความนั้น โดยกุญแจดังกล่าวจะถูกเก็บเป็นความลับระหว่างผู้รับและผู้ส่งข้อความหรือคู่สื่อสาร วิธีนี้จะใช้งานได้ดีเมื่อมีคู่สื่อสารจำนวนไม่มาก ทำให้การแจกจ่ายกุญแจสามารถทำได้ง่าย แต่ในกรณีที่มีคู่สื่อสารจำนวนมากจะมีผลทำให้ผู้ใช้งานต้องมีกุญแจลับในปริมาณมากขึ้นเนื่องจากกุญแจหนึ่งดอกจะใช้สำหรับในแต่ละคู่สื่อสาร นับว่าเป็นความยุ่งยากในการจัดการควบคุมและการแจกจ่ายกุญแจอย่างมาก



รูปที่ 2.2 การเข้ารหัสลับแบบสมมาตร

2) การเข้ารหัสลับแบบอสมมาตร เป็นการเข้ารหัสลับแบบกุญแจอสมมาตร (Asymmetric Key Cryptography) หรือ แบบกุญแจสาธารณะ (Public Key Cryptography) เป็นเทคโนโลยีที่ประกอบด้วยกุญแจ 2 ดอก หรือคู่ของกุญแจ คือ กุญแจส่วนตัว (Private Key) และกุญแจสาธารณะ (Public Key) ตัวหนึ่งใช้สำหรับเข้ารหัสลับและอีกตัวหนึ่งใช้สำหรับถอดรหัสลับที่ถูกเข้ารหัสลับมาโดยกุญแจตัวแรก กุญแจสาธารณะสามารถเผยแพร่ออกสู่สาธารณะเพื่อให้ผู้อื่นสามารถนำไปใช้งานเมื่อต้องการติดต่อสื่อสารกับเราในลักษณะที่ต้องการส่งข้อมูลที่เป็นความลับให้ สำหรับกุญแจส่วนตัว ผู้ที่เป็นเจ้าของจะต้องเก็บรักษาไว้เป็นความลับ และห้ามเปิดเผยสู่สาธารณะโดยเด็ดขาด แต่ถ้ามีเหตุสุดวิสัย เช่น กุญแจลับสูญหาย หรือถูกขโมย เจ้าของก็ควรที่จะสร้างกุญแจคู่ใหม่แล้วจึงเผยแพร่กุญแจสาธรณะชุดใหม่ออกไป



รูปที่ 2.3 การเข้ารหัสลับแบบอสมมาตร

2.6 PHP

PHP ย่อมาจาก PHP Hypertext Preprocessor แต่เดิมย่อมาจาก Personal Home Page Tools เป็นภาษาจำพวก scripting language คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่าสคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปเช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถ สอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

ดังนั้น PHP จึงมีการพัฒนาไปอย่างรวดเร็ว และแพร่หลายโดยเฉพาะอย่างยิ่งเมื่อใช้ร่วมกับ Apache Webserver ระบบปฏิบัติอย่างเช่น Linux หรือ FreeBSD เป็นต้น ในปัจจุบัน PHP สามารถใช้ร่วมกับ Web Server หลาย ๆ ตัว บนระบบปฏิบัติการ อย่างเช่น Windows 95/98/NT/2000/XP เป็นต้น

2.7 MySQL

2.6.1 MySQL คืออะไร

MySQL เป็นโปรแกรมจัดการฐานข้อมูลแบบ Relational Database Management System (RDBMS) ซึ่งได้รับการพัฒนาขึ้นมาจากชาวสวีเดน 2 คน ชื่อ David Axmark, Allan Larsson และชาวฟินแลนด์ 1 คน Michael “Monty” Widenius ซึ่งได้จัดตั้งบริษัทที่ชื่อว่า MySQL ซึ่งโปรแกรมจัดการฐานข้อมูลนี้ได้ถูกพัฒนามาตั้งแต่ปี 1979 แต่ได้เปิดให้ใช้งานจริงเมื่อปี 1996 และ MySQL ยังเป็นโปรแกรมที่ได้รับรางวัล Linux Journal Reader ‘s Choice Award 3 ปีซ้อน ซึ่งเป็นเครื่องการันตีความสามารถของโปรแกรมนี้ได้อย่างยอดเยี่ยม ในปัจจุบันได้ถูดซื้อไปอยู่กับเจ้าของคนใหม่ที่บริษัทว่า ซันไมโครซิสเต็มส์ (Sun Microsystems, Inc.) ถึงแม้ว่าจะมีการขาย MySQL ให้กับ

Sun แล้วแต่โปรแกรมนี้ก็ยังมีการพัฒนาอย่างต่อเนื่องทำให้กลายเป็นโปรแกรมที่ทุกคนเลือกใช้งาน ความสามารถที่ทำให้ MySQL กลายเป็นโปรแกรมจัดการฐานข้อมูลที่ทุกคนไว้ว่างใจก็คือการสนับสนุนการทำงานได้เกือบทุกระบบปฏิบัติการณ์ อาทิเช่น Windows และ Linux เป็นต้น นอกจากนั้น MySQL ยังเป็นที่นิยมในการนำไปใช้งานกับ Web Application เป็นอย่างมาก ซึ่งในปัจจุบันเกือบทุกเว็บไซต์ได้ใช้งานโปรแกรม MySQL ทั้งสิ้น

นอกจากความสามารถในการรองรับระบบปฏิบัติการหลากหลายรุ่นแล้ว ความสามารถในการจัดการต่างๆก็ทำได้ดีอีกด้วย ซึ่งจุดเด่นของ MySQL นั้นก็คือความเร็ว เมนูการใช้งานที่สะดวกสบาย พร้อมกับความเอาใจใส่ในการดูแลด้วยการอัพเดทความสามารถของโปรแกรม MySQL จากผู้ผลิตอยู่เรื่อย ๆทำให้ MySQL มีความสามารถใหม่และแก้ไขข้อผิดพลาดที่เกิดขึ้นอยู่เสมอ

MySQL เป็นฐานข้อมูลที่มีการจัดการฐานข้อมูลแบบโครงสร้าง ซึ่งข้อมูลที่ได้รวบรวมมาจะอยู่ในรูปแบบของตารางเพื่อช่วยให้สามารถเข้าหาและสืบค้นข้อมูลได้ง่ายกว่าการเก็บข้อมูลเป็นไฟล์ ซึ่งการเก็บข้อมูลแบบตารางนั้นส่งผลให้การทำงานของ MySQL นั้นทำงานได้รวดเร็วและยืดหยุ่น และข้อมูลทุกๆตารางจะเชื่อมโยงกันทำให้สามารถจัดการข้อมูลต่างๆได้ตามต้องการ

2.6.2 ประโยชน์ของฐานข้อมูล MySQL

โปรแกรม MySQL นั้นเป็นโปรแกรมจัดการฐานข้อมูลที่มีด้วยกัน 2 แบบคือ Open Source License แบบใช้งานได้ฟรีและแบบ Commercial License แบบธุรกิจ ซึ่งเราสามารถเลือกใช้งานได้ตามลักษณะการใช้งาน โดยประโยชน์และความสามารถของ MySQL ส่งผลให้สามารถใช้งานได้หลายด้านด้วยกันเริ่มจาก

การใช้ร่วมกับเครื่องบริการเว็บ (Web Server) ซึ่ง MySQL ถูกออกแบบให้สามารถทำงานร่วมกับฮาร์ดแวร์ตัวอื่น ๆได้ พร้อมกันนั้นยังรองรับภาษาคอมพิวเตอร์ได้อย่างหลากหลาย อีกทั้ง MySQL ยังสามารถจัดการข้อมูลที่มีขนาดใหญ่ได้เป็นอย่างดีจึงเป็นส่วนหนึ่งที่ทำให้ทุกคนเลือกใช้ MySQL เป็นโปรแกรมจัดการฐานของข้อมูลภายในเครื่องเซิร์ฟเวอร์

การใช้งานด้านกราฟฟิก(Graphical) เป็นอีกหนึ่งในความสามารถของ MySQL ที่รองรับการทำงานด้านกราฟฟิก(GUI)โดยมีโปรแกรมต่าง ๆรองรับมากมายอาทิเช่น phpMyAdmin, Navicat, OpenOffice.org, SQLBuddy, Sequel Pro, SQLYog, Toad for MySQL, Adminer, DaDaBIK นอกจากโปรแกรมที่กล่าวมาแล้วนั้นยังมีอีกหลายโปรแกรมที่ให้การสนับสนุนการทำงานของ MySQL

การใช้งาน MySQL ในด้านการเขียนโปรแกรมนั้น MySQL สามารถรองรับระบบการทำงานได้หลายหลายระบบ อาทิเช่น AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS และ อื่น ๆ

2.8 Crontab

Cron มีไว้เพื่อ schedule tasks ที่เราต้องการเช่น กำหนด ณ เวลาเท่านี้ จะต้องทำ task นี้ มันก็จะไปทำ task ตาม script ที่เราเขียนเพื่อไปรันเซิร์ฟเวอร์หรือเรียกอีกอย่างว่าเป็น Job scheduler ซึ่งใช้สำหรับ Unix-like operating systems นั่นเอง และแต่ละ task หรือ job จะถูกเรียกง่ายๆว่า Cron Jobs

คำสั่งและ Option ของ Crontab มีดังนี้

Code:

#crontab filename	การนำเอาคำสั่ง crontab เข้ามาจาก ไฟล์อื่น
#crontab -e	แก้ไข crontab ปัจจุบัน
#crontab -l	ดูคำสั่ง crontab ทั้งหมดที่มีอยู่
#crontab -r	ลบคำสั่ง crontab ที่มีทั้งหมด
#crontab -u user	เป็นคำสั่งของผู้ดูแลระบบเท่านั้น เพื่อใช้ดู แก้ไข ลบ crontab ของ user แต่ละคน

Format ของคำสั่ง crontab มีทั้งหมด 6 fields ดังนี้

1 = minute มีค่า 0 - 59 เวลาเป็นนาที จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงนาทีที่กำหนด

2 = hour มีค่า 0 - 23 เวลาเป็นชั่วโมง จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงชั่วโมงที่กำหนด

3 = day มีค่า 1 - 31 เวลาเป็นวัน จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงวันที่กำหนด

4 = month มีค่า 1 - 12 เวลาเป็นเดือน จะสั่งให้คำสั่งที่กำหนดทำงานทันทีเมื่อถึงเดือนที่กำหนด

5 = weekday มีค่า 0 - 6 วันของแต่ละสัปดาห์

6 = command ผู้ใช้สามารถกำหนดคำสั่งได้มากมาย รวมทั้ง script ต่าง ๆ ตามที่ผู้ใช้งานต้องการ

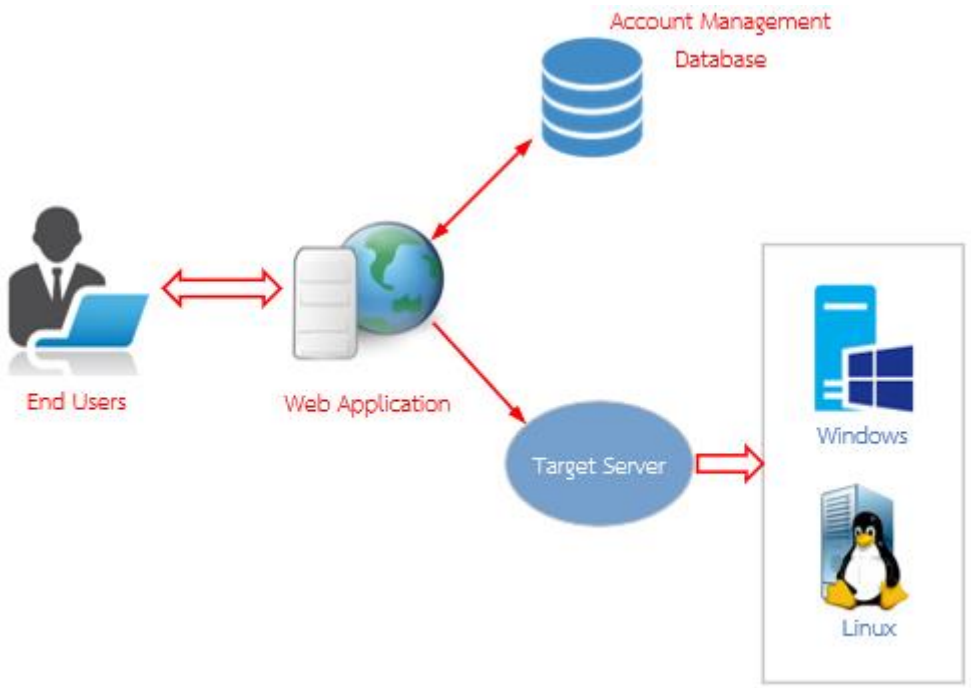
รูปที่ 2.4 รูปแบบของ Crontab

บทที่ 3
โครงสร้างและการออกแบบระบบ

3.1 แนวคิดการออกแบบระบบงาน

ระบบการบริหารจัดการ Privileged Account สำหรับการเข้าถึงการใช้งานระบบต่าง ๆ ที่มีความสำคัญ เป็นระบบที่ทำให้ผู้ใช้งานที่มีความประสงค์ที่จะใช้งาน Privileged Account สามารถร้องขอ และได้รับรหัสผ่าน ผ่านทางเว็บไซต์ เมื่อผู้ใช้งานร้องขอในการเข้าใช้งาน Privileged Account บนหน้าเว็บไซต์ และผู้บริหารระบบได้ดำเนินการอนุมัติเรียบร้อยแล้ว จากนั้นระบบจะทำการ Generate รหัสผ่าน โดยผู้ใช้งานสามารถนำรหัสผ่านที่ได้รับนั้นไปใช้งานทันที ณ เวลาที่ดำเนินการร้องขอ

3.2 โครงสร้างของระบบ

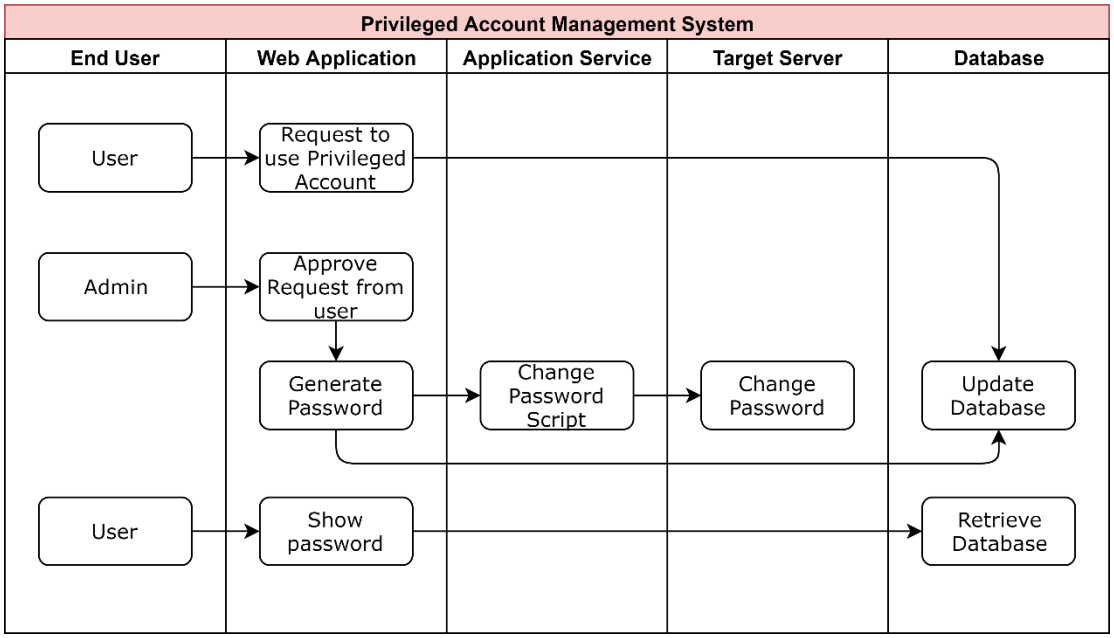


รูปที่ 3.1 แสดงโครงสร้างของระบบ Privileged Account Management

3.3 โครงสร้างของระบบประกอบด้วย

1) ผู้ใช้งาน (User) คือ ผู้ที่ต้องการร้องขอ Privileged Account ในการเข้าถึงการใช้งานระบบ
2) ผู้ดูแลระบบ (Admin) คือ ผู้ดูแลและบริหารจัดการระบบจัดการ Privileged Account
3) Web Application คือ การให้บริการเว็บไซต์สำหรับให้ผู้ใช้งาน ทำการบันทึกข้อมูลคำร้องขอ Privileged Account และทำหน้าที่ Generate รหัสผ่านส่งไปให้ระบบที่ผู้ใช้งานต้องการเข้าถึงการใช้งาน
4) Database Server คือ การให้บริการเก็บข้อมูลผู้ใช้งาน, เครื่องเซิร์ฟเวอร์ที่สามารถให้บริการได้ในระบบ, คำร้องขอของผู้ใช้งาน, รหัสผ่านที่ถูกสร้างขึ้น และเวลาเริ่มต้น-สิ้นสุดในการเข้าระบบต่างๆ ที่ผู้ใช้งานได้ทำการร้องขอ
5) Target Server คือ เซิร์ฟเวอร์ที่ User ร้องขอรหัสผ่านในการเข้ามาทำงาน

3.4 การทำงานของระบบ



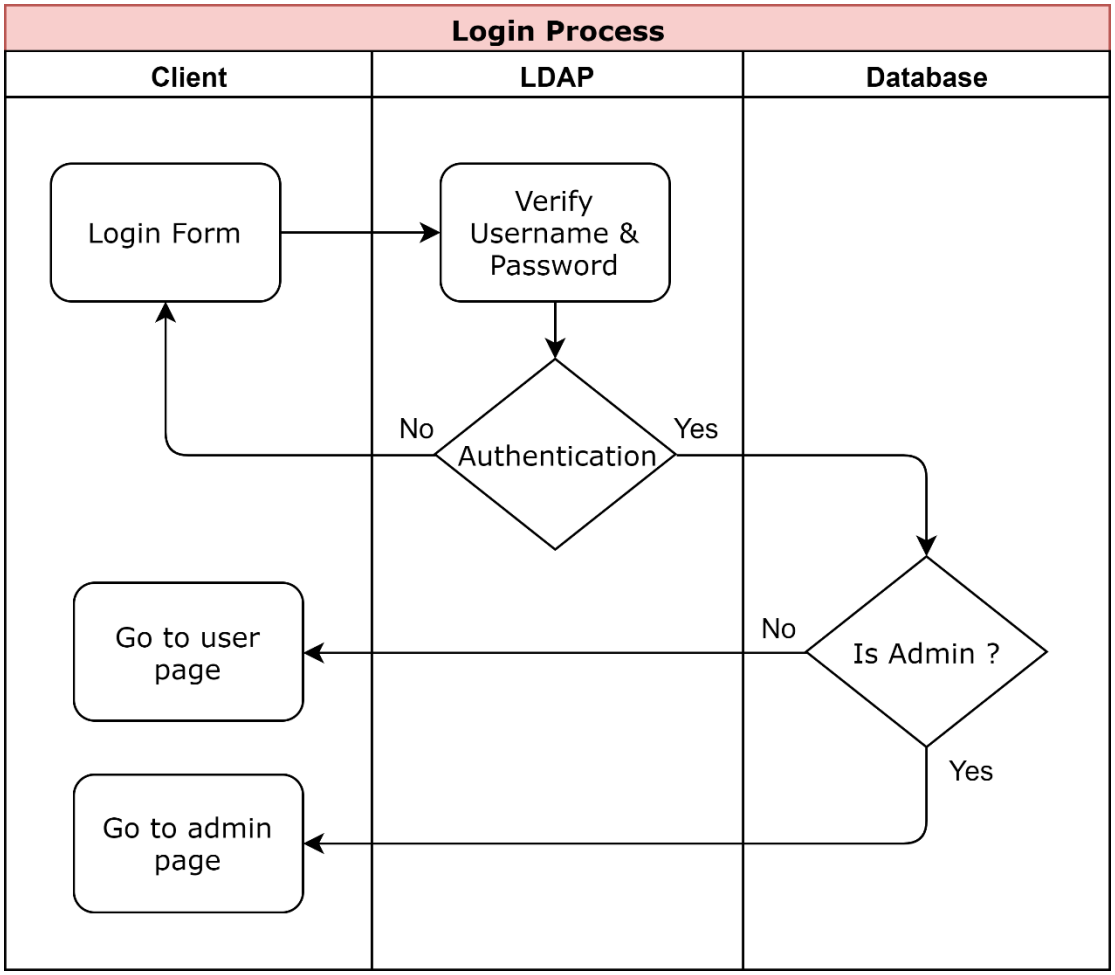
รูปที่ 3.2 ภาพรวมการทำงานของระบบ Privileged Account Management

จากรูปที่ 3.2 แสดงการทำงานของระบบโดยรวมซึ่งสามารถอธิบายได้ ดังนี้

1) ผู้ดูแลระบบหรือผู้ที่ต้องการร้องขอ Privileged Account ทำการล็อกอินเข้าสู่ระบบโดยการระบุชื่อผู้ใช้งานและรหัสผ่านบนหน้าเว็บไซต์ การตรวจสอบชื่อผู้ใช้งาน และรหัสผ่านจะถูกนำไปเปรียบเทียบกับข้อมูลในฐานข้อมูลของ AD (Active Directory)
2) การเข้าถึงข้อมูลของผู้ใช้งานที่ร้องขอ Privileged Account และ ผู้ดูแลระบบสามารถแบ่งแยกได้เป็น 2 ส่วน
    2.1) ผู้ใช้งานที่ร้องขอ Privileged Account
        - สามารถร้องขอรหัสผ่านสำหรับใช้งานระบบที่เกี่ยวข้องกับงาน โดยระบุถึงระบบที่ต้องการเข้าถึงการใช้งาน ระบุเวลาเริ่มต้น และเวลาสิ้นสุด
        - ผู้ใช้งานสามารถเรียกดูข้อมูลรหัสผ่านที่ได้รับการอนุมัติจากผู้ดูแลระบบผ่านทางเว็บไซต์
    2.2) ผู้ดูแลระบบ
        - สามารถเพิ่ม และถอดถอนสิทธิ์ผู้ดูแลระบบได้
        - สามารถเพิ่ม แก้ไข และลบ เซิร์ฟเวอร์ที่มีในระบบได้
        - สามารถร้องขอรหัสผ่านสำหรับใช้งานระบบที่เกี่ยวข้องกับงาน โดยระบุถึงระบบที่ต้องการเข้าถึงการใช้งาน ระบุเวลาเริ่มต้น และเวลาสิ้นสุด
        - สามารถตรวจสอบการร้องขอ การบันทึก และการอนุมัติการร้องขอจากผู้ใช้งาน แต่จะไม่สามารถอนุมัติขอคำของตนเองได้
        - สามารถเรียกดูประวัติคำร้องขอทั้งหมดได้
        - สามารถตรวจสอบประวัติการเข้าระบบเครื่องเซิร์ฟเวอร์ปลายทางของผู้ใช้งานได้
3) เมื่อคำขอร้องของผู้ใช้งานได้รับการอนุมัติจากผู้ดูแลระบบแล้ว มีขั้นตอนการทำงานดังนี้
    3.1) ระบบจะทำการสร้างรหัสผ่าน และจัดเก็บข้อมูลรายละเอียดการร้องขอ บันทึกไว้ในฐานข้อมูล
    3.2) ระบบจะส่งคำสั่งเปิดบัญชี และเปลี่ยนรหัสผ่านไปยังระบบที่เกี่ยวข้องกับงานตามที่ผู้ใช้งานร้องขอ
4) เมื่อครบกำหนดเวลาที่ผู้ใช้งานร้องขอ ระบบจะทำการปิดบัญชีผู้ใช้นั้น เพื่อป้องกันการนำรหัสผ่านมาใช้งานในภายหลัง

3.5 การออกแบบ Process Flow การทำงานของระบบ

3.5.1 การออกแบบ Process Flow ขั้นตอนการตรวจสอบชื่อผู้ใช้

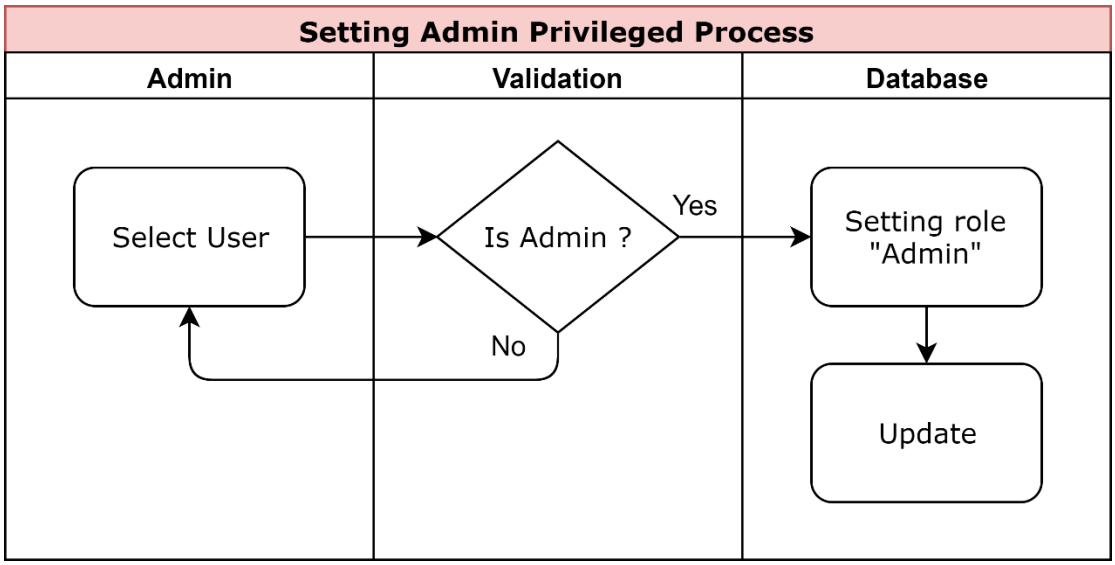


รูปที่ 3.3 การออกแบบ Process Flow การตรวจสอบชื่อผู้ใช้

จากรูปที่ 3.3 แสดงขั้นตอนการตรวจสอบชื่อผู้ใช้มีขั้นตอน ดังต่อไปนี้

1) ระบบจะแสดงหน้าล็อกอินสำหรับระบุชื่อผู้ใช้งานและรหัสผ่าน
2) หลังจากผู้ใช้ทำการกรอกชื่อผู้ใช้งาน และรหัสผ่านแล้ว ระบบจะนำชื่อผู้ใช้งานไปตรวจสอบกับข้อมูลที่มีอยู่ในฐานข้อมูลใน LDAP ว่ามีอยู่หรือไม่ หากไม่มีชื่อผู้ใช้งานก็จะกลับสู่หน้าล็อกอินอีกครั้ง
3) ระบบจะทำการตรวจสอบสิทธิ์ หากเป็นผู้ใช้งานทั่วไป จะเข้าสู่หน้าเว็บไซต์ของผู้ใช้งาน และหากเป็นผู้ดูแลระบบจะเข้าสู่หน้าเว็บไซต์ของผู้ดูแลระบบ

3.5.2 การออกแบบ Process Flow ขั้นตอนการเพิ่มสิทธิ์ผู้ดูแลระบบ

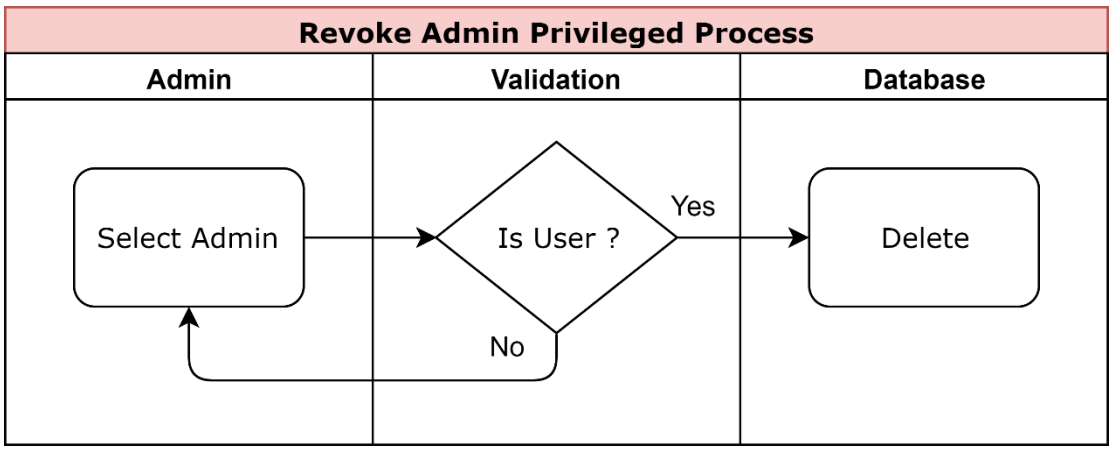


รูปที่ 3.4 การออกแบบ Process Flow ขั้นตอนการเพิ่มสิทธิ์ผู้ดูแลระบบ

จากรูปที่ 3.4 แสดงขั้นตอนการเพิ่มสิทธิ์ผู้ดูแลระบบมีขั้นตอน ดังต่อไปนี้

ผู้ดูแลระบบเลือกผู้ใช้งานที่ได้รับสิทธิ์ในการเป็นผู้ดูแลระบบ จากนั้นระบบจะตรวจสอบกับฐานข้อมูลว่าผู้ใช้งานดังกล่าวมีสิทธิ์เป็นผู้ดูแลระบบหรือไม่ ถ้าหากยังไม่มีสิทธิ์จะเพิ่มข้อมูลลงในฐานข้อมูล

3.5.3 การออกแบบ Process Flow ขั้นตอนการเพิกถอนสิทธิ์ผู้ดูแลระบบ

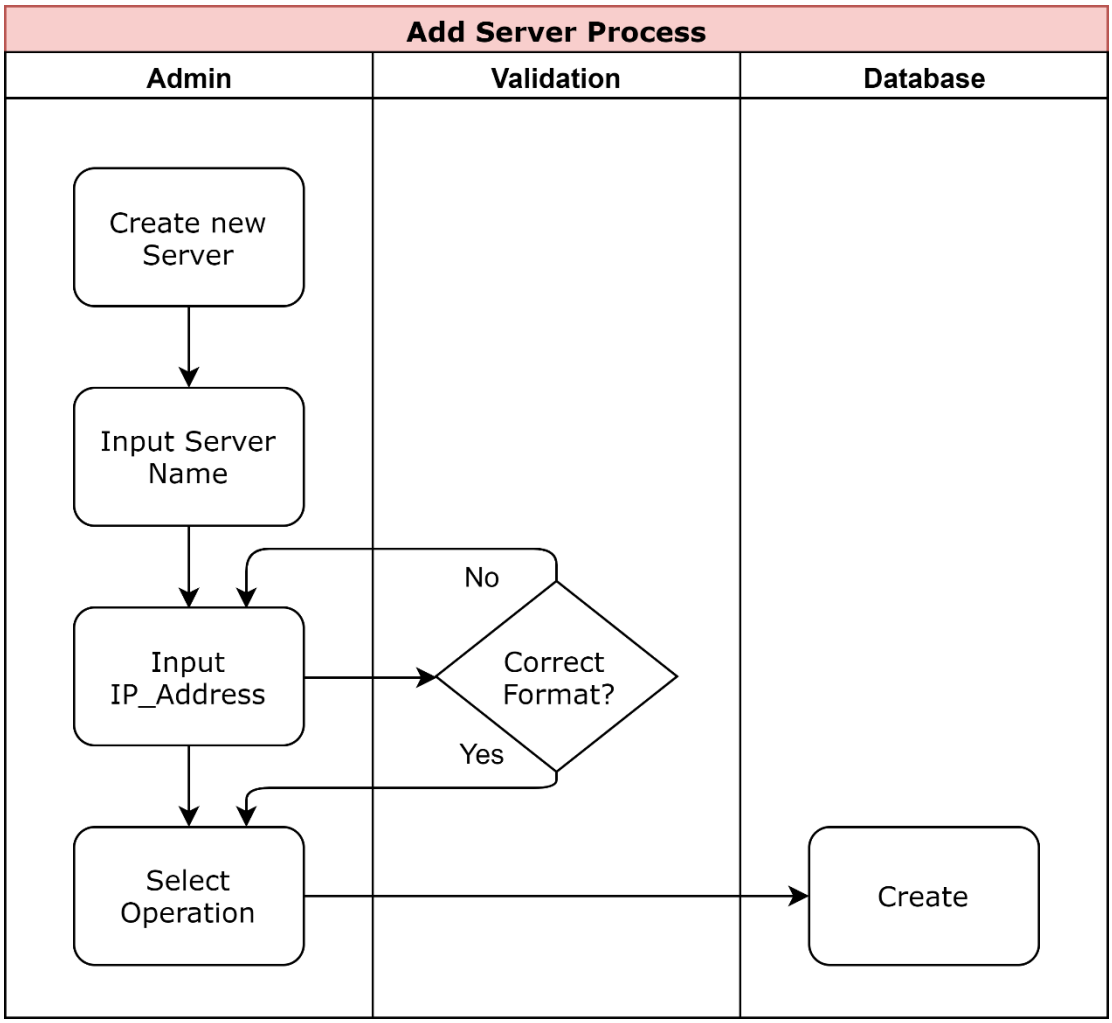


รูปที่ 3.5 การออกแบบ Process Flow ขั้นตอนการเพิกถอนสิทธิ์ผู้ดูแลระบบ

จากรูปที่ 3.5 แสดงขั้นตอนการเพิกถอนสิทธิ์ผู้ดูแลระบบมีขั้นตอน ดังต่อไปนี้

ผู้ดูแลระบบเลือกผู้ใช้งานที่ต้องการจะเพิกถอนสิทธิ์ ระบบจะตรวจสอบว่าเป็นสิทธิ์ผู้ใช้งานหรือไม่ จากนั้นระบบจะทำการลบผู้ใช้งานดังกล่าวออกจากฐานข้อมูล

3.5.4 การออกแบบ Process Flow ขั้นตอนการเพิ่ม เซิร์ฟเวอร์ในระบบ



รูปที่ 3.6 การออกแบบ Process Flow ขั้นตอนการเพิ่ม เซิร์ฟเวอร์ในระบบ

จากรูปที่ 3.6 แสดงขั้นตอนการเพิ่ม เซิร์ฟเวอร์ในระบบมีขั้นตอน ดังต่อไปนี้

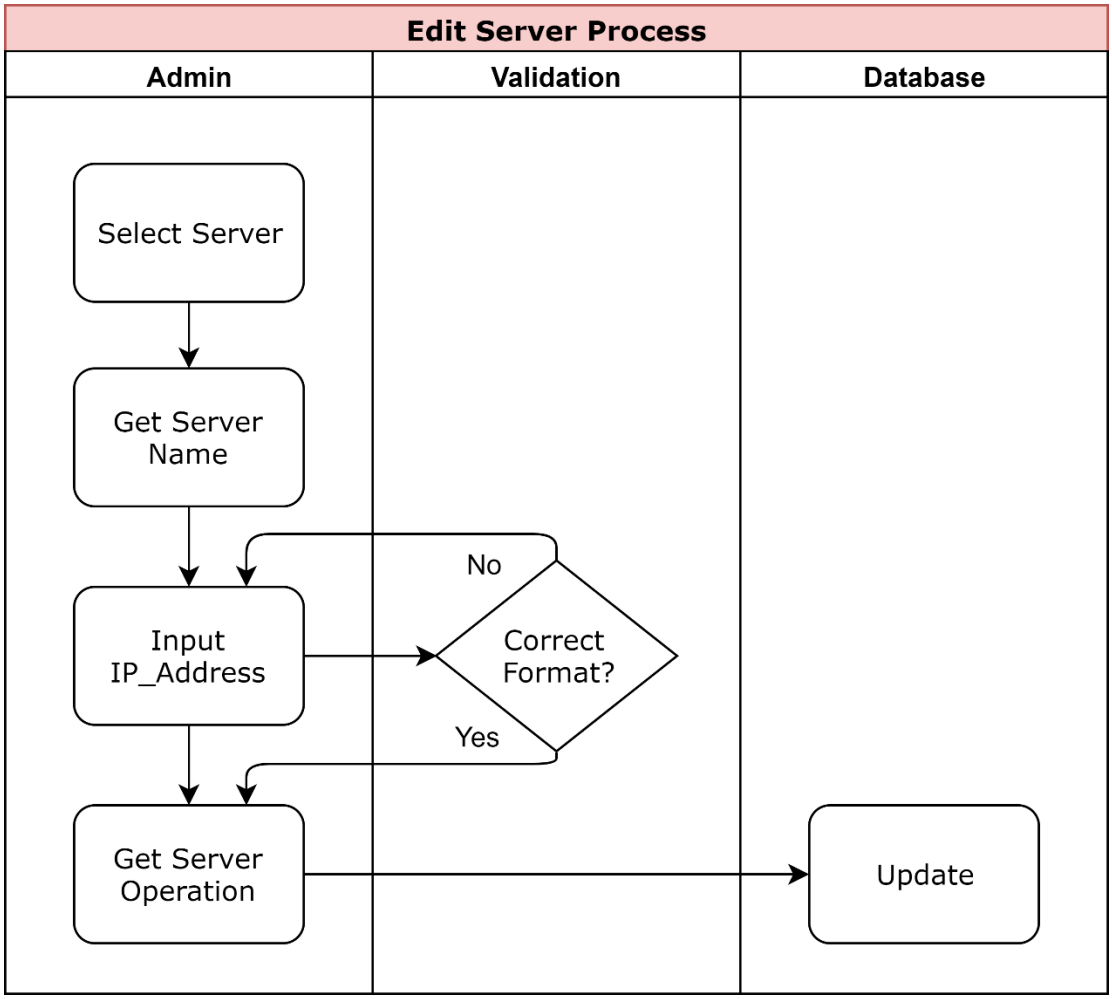
ผู้ดูแลระบบดำเนินการบันทึกข้อมูลของ เซิร์ฟเวอร์ที่จะเพิ่มเข้ามาในระบบ โดยทำการระบุข้อมูลดังต่อไปนี้

- Server Name คือ ชื่อของเครื่องเซิร์ฟเวอร์
- IP Address คือ IP ของเครื่องเซิร์ฟเวอร์

- Select Operation คือ เลือกระบบปฏิบัติการของเครื่องเซิร์ฟเวอร์ เช่น ระบบ Windows หรือ ระบบ Linux

เมื่อดำเนินการบันทึกข้อมูลของเซิร์ฟเวอร์แล้ว ข้อมูลจะถูกจัดเก็บข้อมูลลงในฐานข้อมูล

3.5.5 การออกแบบ Process Flow ขั้นตอนการแก้ไขเซิร์ฟเวอร์ที่มีอยู่ในระบบ

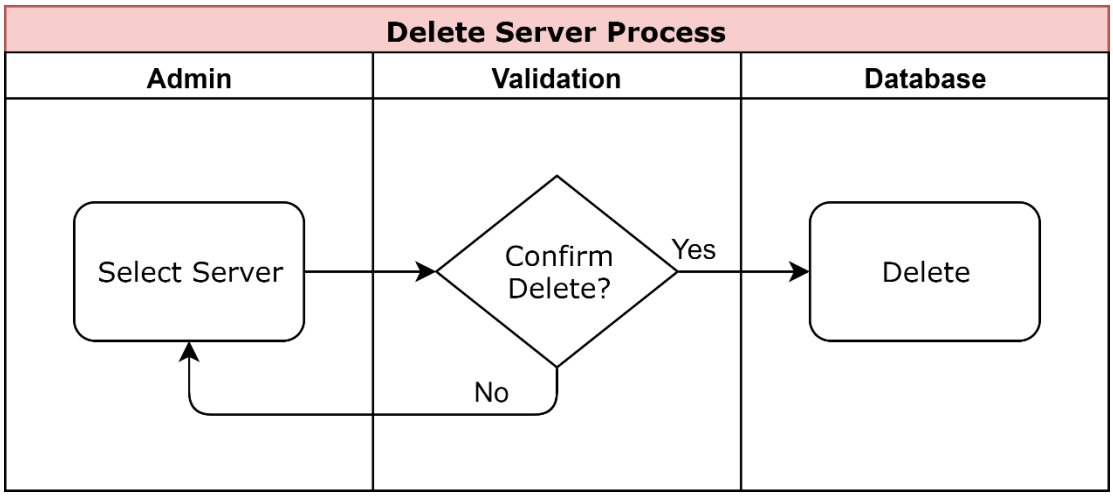


รูปที่ 3.7 การออกแบบ Process Flow ขั้นตอนการแก้ไขเซิร์ฟเวอร์ที่มีอยู่ในระบบ

จากรูปที่ 3.7 แสดงขั้นตอนการแก้ไขเซิร์ฟเวอร์ที่มีอยู่ในระบบมีขั้นตอน ดังต่อไปนี้

ระบบจะไปดึงข้อมูลชื่อเซิร์ฟเวอร์และ ระบบปฏิบัติการของเซิร์ฟเวอร์นั้นมาแสดง ทางผู้ดูแลระบบจะสามารถแก้ไขได้เฉพาะ IP Address เท่านั้น หลังจากแก้ไขเสร็จเรียบร้อยแล้ว ข้อมูลจะถูกจัดเก็บข้อมูลลงในฐานข้อมูล

3.5.6 การออกแบบ Process Flow ขั้นตอนการลบเซิร์ฟเวอร์ที่มีอยู่ในระบบ

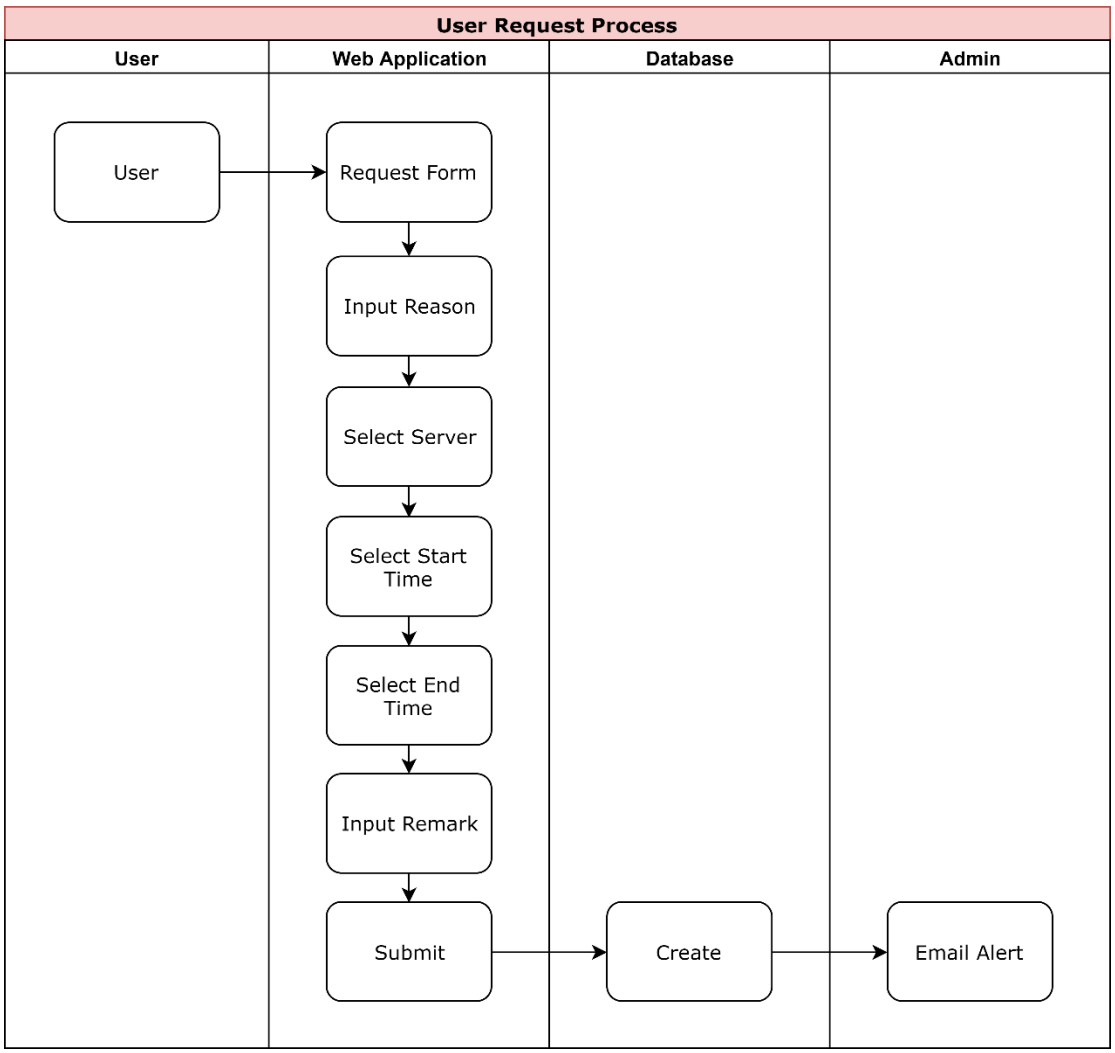


รูปที่ 3.8 การออกแบบ Process Flow ขั้นตอนการลบเซิร์ฟเวอร์ที่มีอยู่ในระบบ

จากรูปที่ 3.8 แสดงขั้นตอนการลบเซิร์ฟเวอร์ที่มีอยู่ในระบบมีขั้นตอน ดังต่อไปนี้

ผู้ดูแลระบบสามารถเลือกเซิร์ฟเวอร์ ที่ต้องการจะลบที่มีอยู่ในระบบได้ หลังจากยืนยันคำสั่งลบแล้ว ข้อมูลจะถูกลบออกจากฐานข้อมูล

3.5.7 การออกแบบ Process Flow ขั้นตอนการร้องขอใช้งานรหัสผ่านของ Privileged Account



รูปที่ 3.9 การออกแบบ Process Flow การร้องขอใช้งานรหัสผ่านของ Privileged Account

จากรูปที่ 3.9 แสดงขั้นตอนการร้องขอใช้งานรหัสผ่านของ Privileged Account มีขั้นตอนดังต่อไปนี้

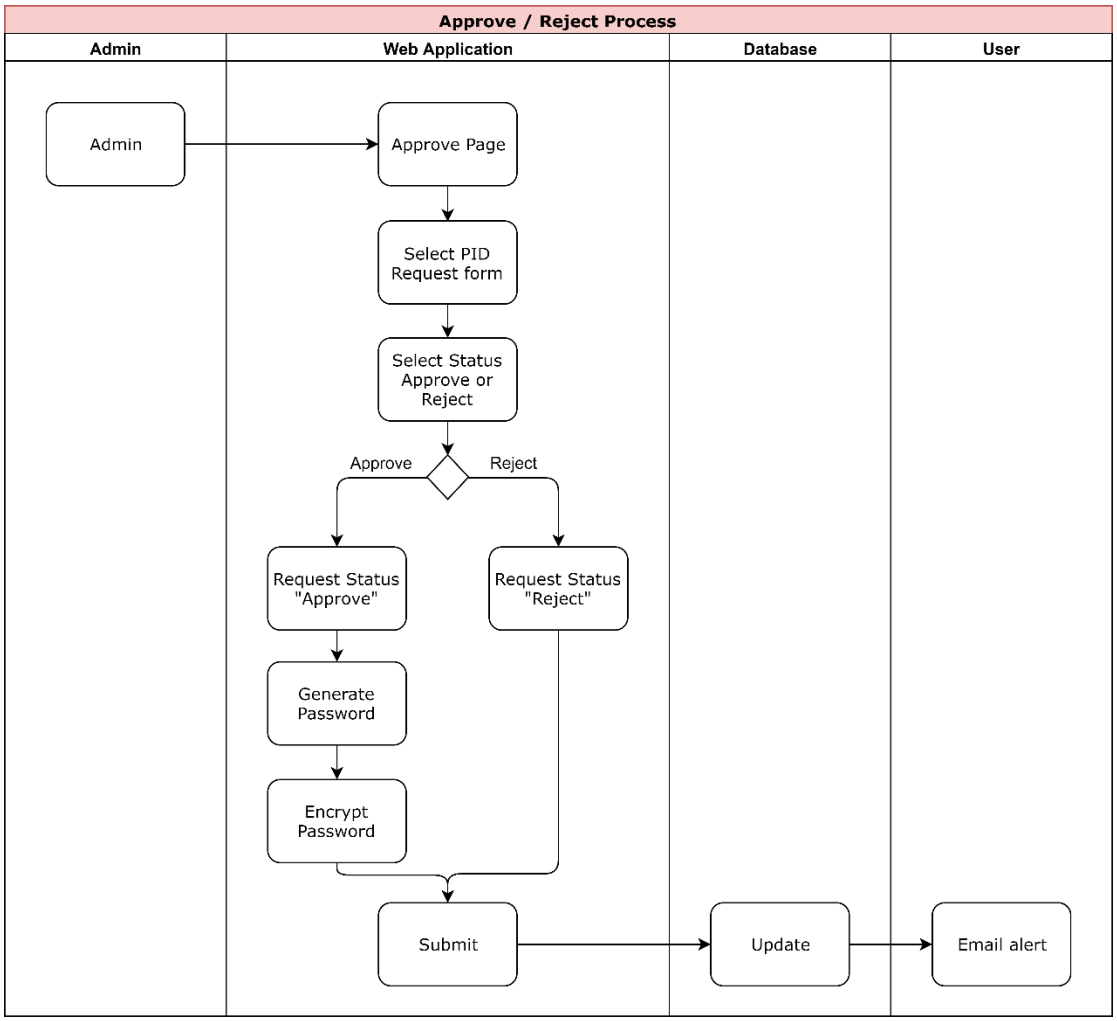
ผู้ใช้งานดำเนินการบันทึกข้อมูลใน Request Form เพื่อร้องขอการใช้งาน Privileged Account โดยทำการระบุข้อมูลดังต่อไปนี้

- Reason คือ เหตุผลในการขอใช้งาน Privileged Account
- Server คือ เซิร์ฟเวอร์ที่ผู้ใช้มีความประสงค์จะขอเข้าใช้งาน โดยระบบจะทำการดึงข้อมูลของเซิร์ฟเวอร์จากฐานข้อมูล และนำมาแสดงผล ดังนี้ IP Address, ชื่อเซิร์ฟเวอร์ และ ชื่อระบบปฏิบัติการ

- Start Time คือ เวลาเริ่มต้นที่ผู้ใช้งานมีความประสงค์ขอใช้งาน
- End Time คือ เวลาสิ้นสุดที่ผู้ใช้งานมีความประสงค์สิ้นสุดการใช้งาน
- Remark คือ ข้อมูลเพิ่มเติมในการขอใช้งาน Privileged Account

เมื่อดำเนินการบันทึกคำขอใช้งานแล้ว ข้อมูลจะถูกจัดเก็บข้อมูลลงในฐานข้อมูล และระบบจะทำการแจ้งเตือนอัตโนมัติผ่านทาง email ให้กับผู้ดูแลระบบได้รับทราบ

3.5.8 การออกแบบ Process Flow ขั้นตอนการอนุมัติ หรือ ไม่อนุมัติคำร้องขอ



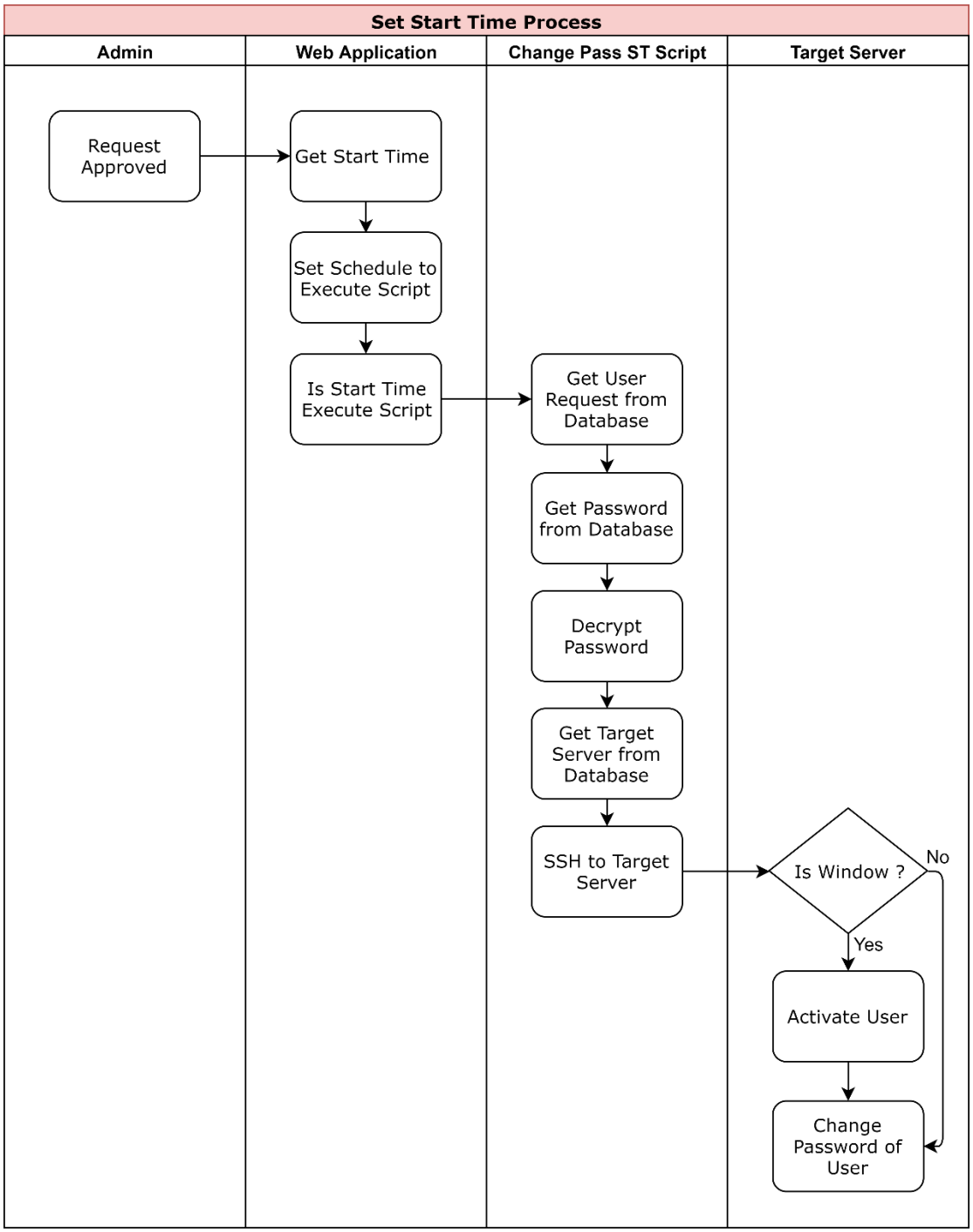
รูปที่ 3.10 การออกแบบ Process Flow ขั้นตอนการอนุมัติ หรือ ไม่อนุมัติคำร้องขอ

จากรูปที่ 3.10 แสดงขั้นตอนการอนุมัติ หรือ ไม่อนุมัติคำร้องขอมีขั้นตอน ดังต่อไปนี้

ในหน้าเพจของผู้ดูแลระบบ จะแสดงคำร้องขอของผู้ใช้งาน ซึ่งทางผู้ดูแลระบบจะดำเนินการตรวจสอบว่าจะอนุมัติหรือไม่ โดยกำหนดสถานะ ดังนี้

- กรณีที่ผู้ดูแลระบบอนุมัติ ระบบจะทำการ Generate รหัสผ่าน รหัสผ่านที่ถูก Generate ขึ้นมาจะถูกเข้ารหัส หลังจากนั้นระบบจะดำเนินการอัพเดทรหัสผ่านและสถานะ Approve ลงในฐานข้อมูล
- กรณีที่ผู้ดูแลระบบไม่อนุมัติ ระบบจะดำเนินการอัพเดท สถานะ Reject ลงในฐานข้อมูล
- ระบบจะดำเนินการแจ้งเตือนอัตโนมัติผ่านทาง email ให้กับผู้ใช้ที่ได้ทำการร้องขอทราบ โดยจะส่งข้อมูลเกี่ยวกับสถานะของการร้องขอว่าได้รับการอนุมัติหรือไม่

3.5.9 การออกแบบ Process Flow ขั้นตอนการตั้งเวลาเริ่มต้น



รูปที่ 3.11 การออกแบบ Process Flow การตั้งเวลาเริ่มต้น

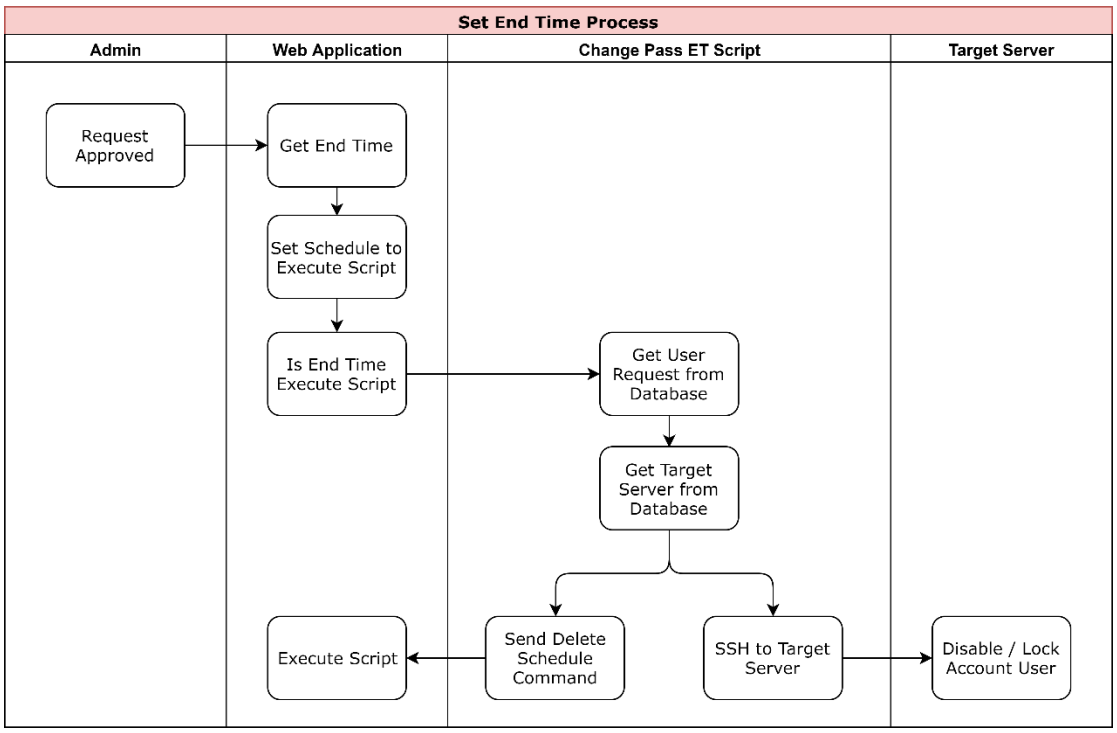
จากรูปที่ 3.11 แสดงขั้นตอนการตั้งเวลาเริ่มต้นมีขั้นตอน ดังต่อไปนี้

หลังจากที่ผู้ดูแลระบบอนุมัติคำร้องขอของผู้ใช้งานแล้ว ระบบจะดึงเวลาเริ่มต้นที่ผู้ใช้งานร้องขอนำมาตั้งเวลาในการส่งคำสั่งให้ Script ทำงาน

เมื่อถึงเวลาที่ผู้ใช้งานร้องขอ Script จะเริ่มทำงานโดยมีขั้นตอนดังต่อไปนี้

- Script จะทำการดึงข้อมูลของผู้ใช้งาน, รหัสผ่านที่ถูก Generate ขึ้นมา และเครื่องเซิร์ฟเวอร์ปลายทางที่ผู้ใช้งานร้องขอ
- รหัสผ่านที่ดึงมานั้น จะถูกถอดรหัสก่อนนำมาใช้งาน
- Script จะดำเนินการ SSH ไปยัง เครื่องเซิร์ฟเวอร์ปลายทางที่ผู้ใช้งานร้องขอไว้
- ตรวจสอบว่าเครื่องเซิร์ฟเวอร์ปลายทางเป็นระบบปฏิบัติการอะไร ถ้าหากเป็นระบบปฏิบัติการ Windows จะส่งคำสั่งเพื่อ Activate Account ของผู้ใช้งานก่อน
- ส่งคำสั่งในการเปลี่ยนรหัสผ่านไปยัง Account ของผู้ใช้งาน

3.5.10 การออกแบบ Process Flow ขั้นตอนการตั้งเวลาสิ้นสุด



รูปที่ 3.12 การออกแบบ Process Flow การตั้งเวลาสิ้นสุด

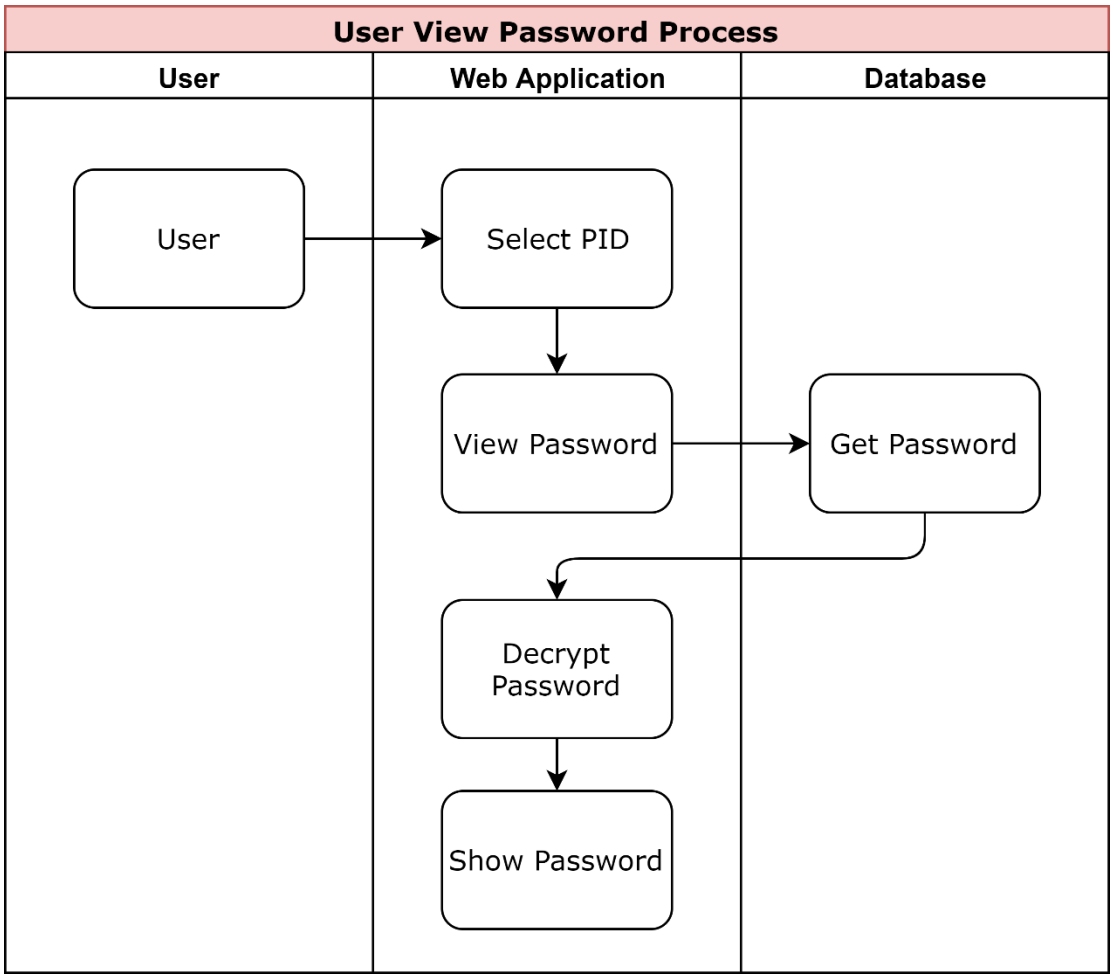
จากรูปที่ 3.12 แสดงขั้นตอนการตั้งเวลาสิ้นสุดมีขั้นตอน ดังต่อไปนี้

หลังจากที่ผู้ดูแลระบบอนุมัติคำร้องขอของผู้ใช้งานแล้ว ระบบจะดึงเวลาสิ้นสุดที่ผู้ใช้งานร้องขอนำมาตั้งเวลาในการส่งคำสั่งให้ Script ทำงาน

เมื่อถึงเวลาสิ้นสุด Script จะเริ่มทำงานดังต่อไปนี้

- Script จะทำการดึงข้อมูลของผู้ใช้งาน และ เครื่องเซิร์ฟเวอร์ปลายทางที่ผู้ใช้งานร้องขอ
- ส่งคำสั่งไปยัง Web Application เพื่อลบเวลาเริ่มต้น และเวลาสิ้นสุดที่ถูกตั้งไว้ของคำร้องขอนี้
- Script จะดำเนินการ SSH ไปยัง เครื่องเซิร์ฟเวอร์ปลายทาง
- ส่งคำสั่งในการ Disable หรือ Lock Account ของผู้ใช้งาน

3.5.11 การออกแบบ Process Flow ขั้นตอนการดูรหัสผ่าน

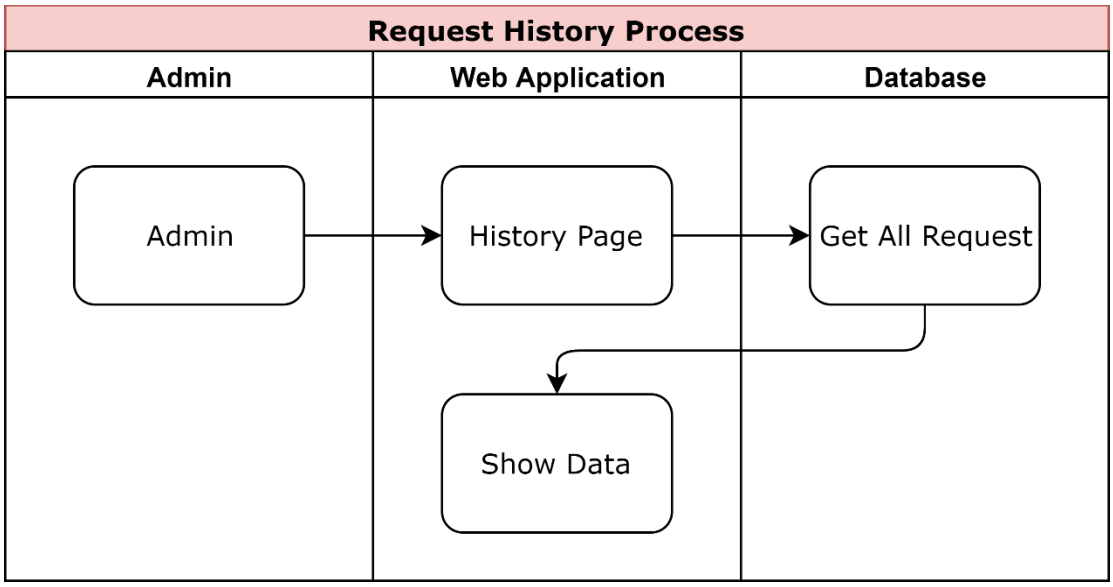


รูปที่ 3.13 การออกแบบ Process Flow การดูรหัสผ่าน

จากรูปที่ 3.13 แสดงขั้นตอนการการดูรหัสผ่านมีขั้นตอน ดังต่อไปนี้

ผู้ใช้งานเลือกคำร้องขอที่ได้รับการอนุมัติ เพื่อดูรหัสผ่านที่ระบบสร้างขึ้นมา ระบบจะดำเนินการดึงรหัสผ่านจากฐานข้อมูลออกมา ซึ่งรหัสผ่านที่ได้มีการเข้ารหัสไว้ ระบบจะทำการถอดรหัสก่อน และแสดงผลให้กับผู้ใช้งาน

3.5.12 การออกแบบ Process Flow ขั้นตอนดูประวัติของการร้องขอใช้งาน Privileged Account



รูปที่ 3.14 การออกแบบ Process Flow ขั้นตอนดูประวัติของการร้องขอใช้งาน Privileged Account

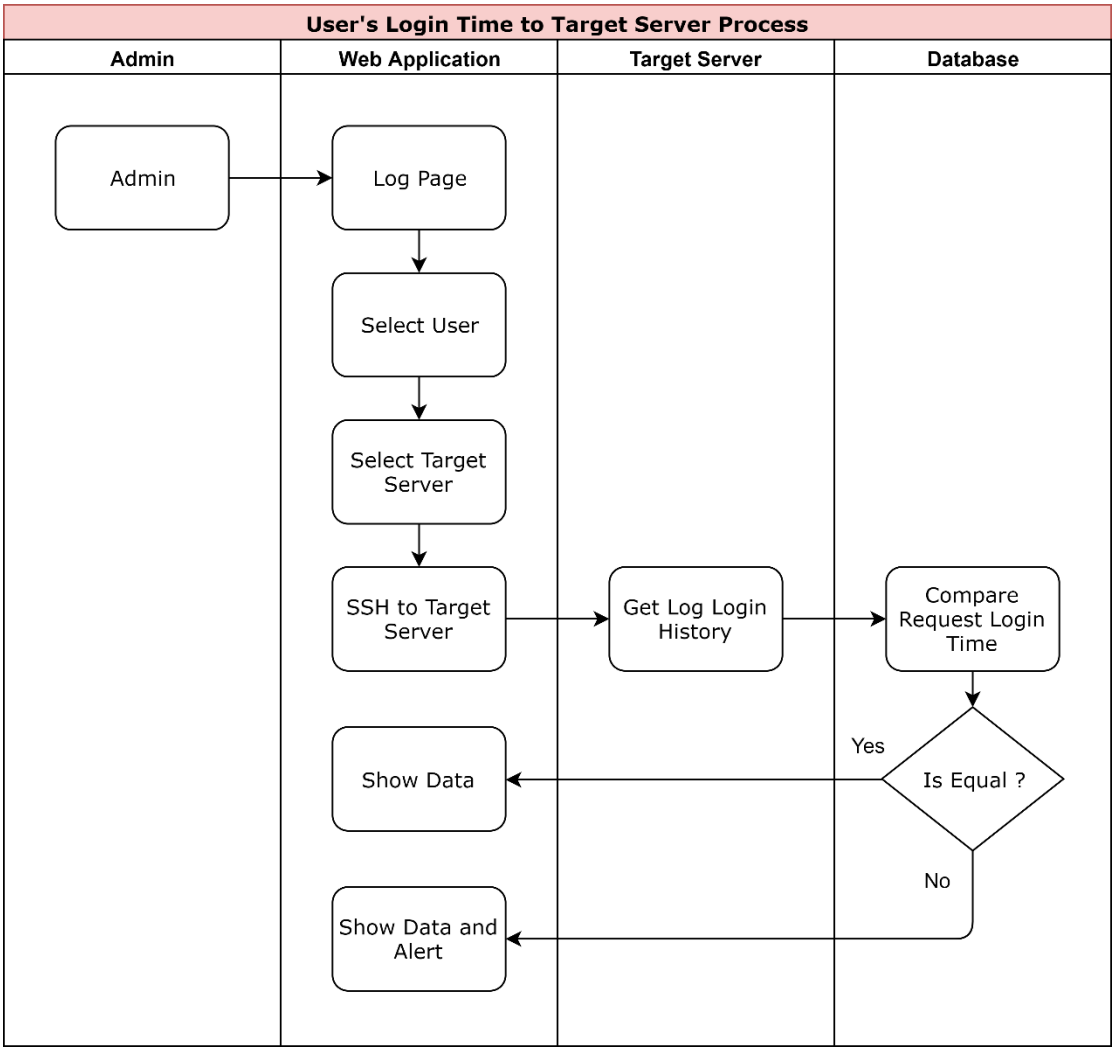
จากรูปที่ 3.14 แสดงขั้นตอนการดูประวัติของการร้องขอใช้งาน Privileged Account ดังต่อไปนี้

ผู้ดูแลระบบสามารถเรียกดูข้อมูลประวัติการร้องขอใช้งาน Privileged Account ทั้งหมดได้ โดยระบบจะทำการดึงข้อมูลจากฐานข้อมูลมาแสดง โดยจะนำข้อมูลมาแสดงดังนี้

- Request By คือ ชื่อของผู้ใช้งานที่ทำการร้องขอใช้งาน Privileged Account
- Server Detail คือ รายละเอียดของเครื่องเซิร์ฟเวอร์ที่ผู้ใช้งานทำการร้องขอ โดยจะแสดงข้อมูล IP Address, ชื่อของเครื่องเซิร์ฟเวอร์ และระบบปฏิบัติการ
- Start Time คือ เวลาเริ่มต้นที่ผู้ใช้งานมีความประสงค์ขอใช้งาน
- End Time คือ เวลาสิ้นสุดที่ผู้ใช้งานมีความประสงค์สิ้นสุดการใช้งาน

- Status คือ สถานะของคำขอใช้งาน Privileged Account
- Approve/Reject By คือ ชื่อของผู้ดูแลระบบที่เข้ามาอนุมัติหรือไม่อนุมัติคำขอนั้น

3.5.13 การออกแบบ Process Flow ขั้นตอนการตรวจสอบประวัติการเข้าระบบเครื่องเซิร์ฟเวอร์ปลายทาง



รูปที่ 3.15 การออกแบบ Process Flow ขั้นตอนการตรวจสอบประวัติการเข้าระบบเครื่องเซิร์ฟเวอร์ปลายทาง

จากรูปที่ 3.15 แสดงขั้นตอนการตรวจสอบประวัติการเข้าระบบเครื่องเซิร์ฟเวอร์ปลายทางมีขั้นตอน ดังต่อไปนี้

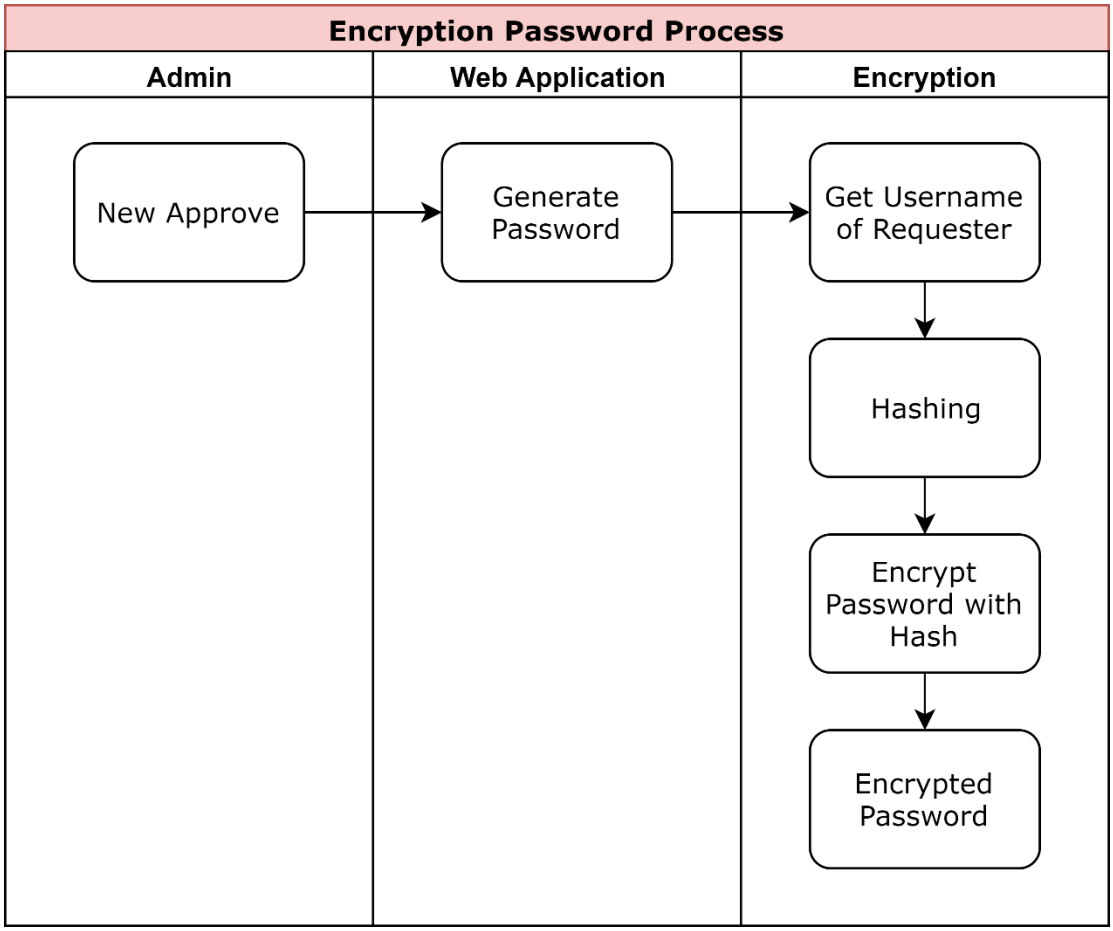
ผู้ดูแลระบบสามารถเรียกดูข้อมูลการเข้าสู่ระบบของเครื่องเซิร์ฟเวอร์ปลายทางที่ผู้ใช้งานทำการร้องขอเพื่อตรวจสอบการเข้าสู่ระบบของผู้ใช้งานได้ โดยระบุข้อมูลดังต่อไปนี้

- User คือ ชื่อของผู้ใช้งานที่ต้องการตรวจสอบประวัติการเข้าสู่ระบบ
- Server คือ เครื่องเซิร์ฟเวอร์ที่ต้องการตรวจสอบประวัติการเข้าสู่ระบบ

เมื่อทำการกรอกข้อมูลเรียบร้อยแล้ว ระบบจะทำการส่งคำสั่งเพื่อไปดึงประวัติของเวลาที่ผู้ใช้งานเข้าสู่ระบบไปยังเซิร์ฟเวอร์ที่ระบุข้างต้น และจะนำเวลาที่ได้รับมาไปตรวจสอบกับเวลาที่ได้รับการอนุมัติของผู้ใช้งาน หากเวลาที่เข้าสู่ระบบไม่ใช่เวลาที่ได้รับการอนุมัติจะแสดง Alert ให้ผู้ดูแลระบบทราบ

3.5.14 การออกแบบ Process Flow ขั้นตอนการเข้ารหัสลับของรหัสผ่าน

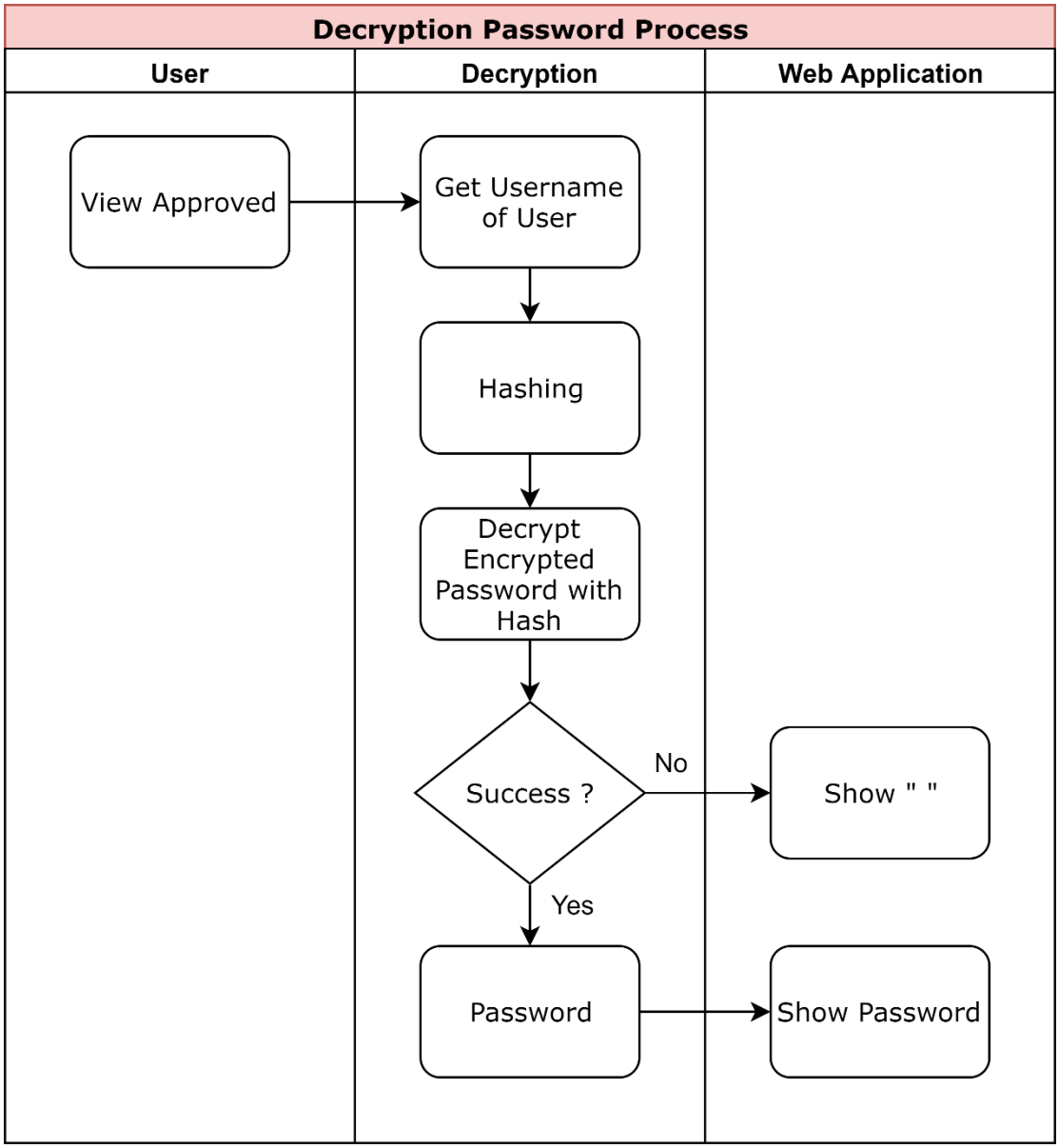
หลังจากที่ผู้ดูแลระบบอนุมัติคำร้องขอใช้งาน Privileged Account เรียบร้อยแล้ว ระบบจะทำการสร้างรหัสผ่านขึ้นมา ระบบจะทำการดึงข้อมูล Username ของผู้ร้องขอจากฐานข้อมูลจากนั้นจะทำการ Hashing เพื่อนำค่า Hash มาเป็น Secret Key เพื่อนำมาใช้เข้ารหัสลับด้วยอัลกอริทึม Symmetric-key จะได้รหัสผ่านที่ถูกเข้ารหัสลับไว้เรียกว่า Encrypted Password จากนั้นระบบจะทำการบันทึกข้อมูลลงในฐานข้อมูล



รูปที่ 3.16 การออกแบบ Process Flow ขั้นตอนการเข้ารหัสลับของรหัสผ่าน

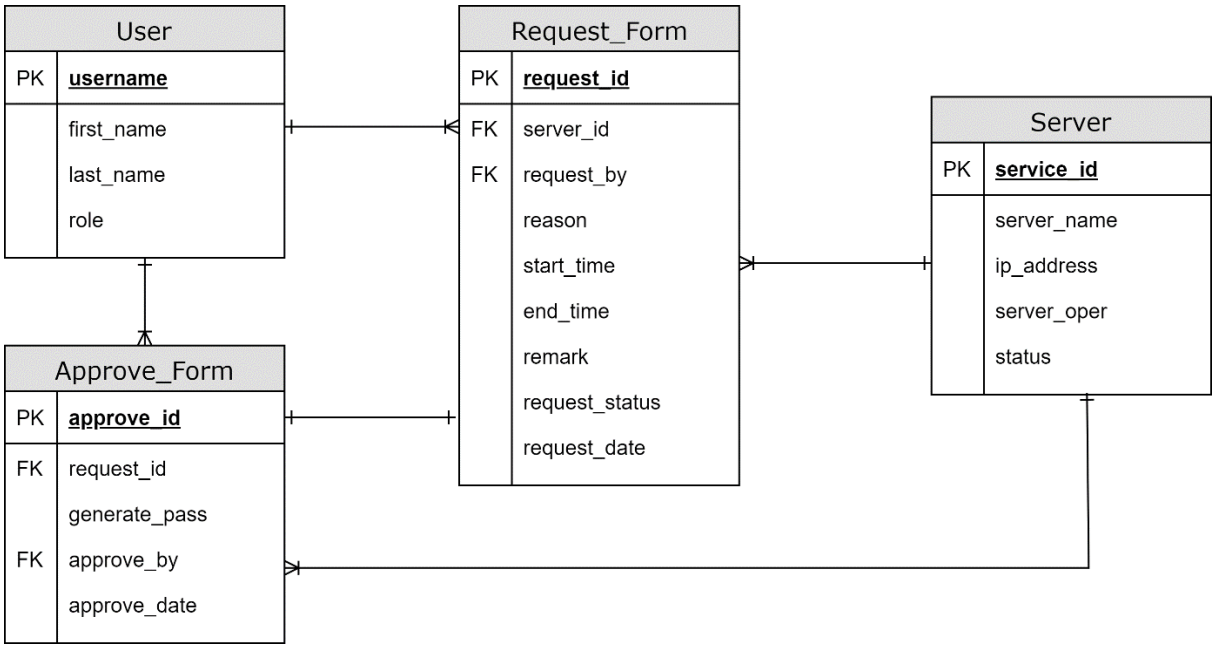
3.5.15 การออกแบบ Process Flow ขั้นตอนการถอดรหัสลับของรหัสผ่าน

เมื่อผู้ใช้งานต้องการดูรหัสผ่านที่ได้รับการอนุมัติแล้วนั้น ระบบจะเข้าสู่กระบวนการถอดรหัสลับโดยการ ดึงข้อมูล Username ของผู้ใช้งานจากฐานข้อมูลจากนั้นจะทำการ hashing เพื่อนำค่า hash มาเป็น Secret Key เพื่อนำมาใช้ถอดรหัสลับด้วยอัลกอริทึม Symmetric-key จะได้รหัสผ่านที่ถูกถอดรหัสลับ หลังจากนั้นระบบจะแสดงข้อมูลรหัสผ่านให้ผู้ใช้งานเพื่อนำไปใช้ในการเข้าระบบต่อไป



รูปที่ 3.17 การออกแบบ Process Flow ขั้นตอนการถอดรหัสลับของรหัสผ่าน

3.6 การออกแบบฐานข้อมูล



รูปที่ 3.17 การออกแบบโครงสร้างฐานข้อมูล

จากการวิเคราะห์ขั้นตอนการทำงาน และความสัมพันธ์ของข้อมูลทำให้สามารถออกแบบระบบฐานข้อมูลเชิงสัมพันธ์ของระบบงาน ซึ่งจะทำให้ได้ตารางของระบบฐานข้อมูลของโปรแกรม โดยสามารถอธิบายถึงรายละเอียดต่างๆ ของข้อมูล ดังแสดงตามตารางที่ 3.1 - 3.4

สำหรับการนำเสนอรายละเอียดของแต่ละตาราง จะนำเสนอในรูปแบบของตาราง ซึ่งมีหัวข้อแต่ละส่วนดังนี้

1) Field หมายถึง ชื่อของ Column ในตาราง
2) PK หมายถึง Primary Key ของตาราง
3) FK หมายถึง Foreign Key ของตาราง
4) Type คือ ชนิดของข้อมูลในแต่ละ Column

ตารางที่ 3.1 แสดงรายละเอียดของตาราง Users

Field	Type	Key	ข้อมูลที่จัดเก็บ	ตัวอย่างข้อมูล
username	varchar(50)	PK	ชื่อ Account ของผู้ใช้งาน	isanai.w
first_name	varchar(255)		ชื่อผู้ใช้งาน	Isanai
last_name	varchar(255)		นามสกุลผู้ใช้งาน	Wongsittigorn
role	int		สิทธิ์ของผู้ใช้งาน (1=ADMIN)	1

ตารางที่ 3.2 แสดงรายละเอียดของตาราง Servers

Field	Type	Key	ข้อมูลที่จัดเก็บ	ตัวอย่างข้อมูล
server_id	int	PK	ลำดับของเครื่อง Server	1
server_name	varchar(255)		ชื่อเครื่อง Server	LinuxCentOS01
ip_address	varchar(255)		IP Address ของเครื่อง Server	192.168.182.129
server_oper	varchar(255)		ชื่อ Operation ของเครื่อง Server (Linux, Window)	Linux
status	int		สถานะของเครื่อง server (0=inactive, 1=active)	1

ตารางที่ 3.3 แสดงรายละเอียดของตาราง Approve Form

Field	Type	Key	ข้อมูลที่จัดเก็บ	ตัวอย่างข้อมูล
request_id	int	PK	ลำดับของการร้องขอ	1
server_id	int	FK	ลำดับของเครื่อง server	1
request_by	varchar(50)	FK	ลำดับของผู้ร้องขอ	isanai.w
reason	varchar(50)		เหตุผลในของการร้องขอ	Allow port 25 for SMTP
start_time	datetime		วันเวลาเริ่มต้นที่ผู้ใช้ขอใช้งาน	15/03/2019 13:00
end_time	datetime		วันเวลาสิ้นสุดที่ผู้ใช้ขอใช้งาน	15/03/2019 16:00
remark	text		รายละเอียดของการร้องขอ	CHG001 config rule iptables
request_status	int		สถานะของการร้องขอ (1=Waiting, 2=Approve, 3=Reject)	2
request_date	timestamp		วันเวลาที่ผู้ใช้ทำการกรอกข้อมูลร้องขอ	12/03/2019 15:00

ตารางที่ 3.4 แสดงรายละเอียดของตาราง Request Form

Field	Type	Key	ข้อมูลที่จัดเก็บ	ตัวอย่างข้อมูล
approved_id	int	PK	ลำดับของการอนุมัติ	1
request_id	int	FK	ลำดับของการร้องขอ	1
generate_pass	varchar(32)		รหัสผ่านที่ถูกสร้างขึ้นมา	aCtBVXNxaWhnVytqVFc5T1BSWU9Zdz09
approve_by	varchar(50)	FK	ลำดับของ คนอนุมัติการร้องขอ	kamol.k
approve_date	timestamp		วันเวลาที่อนุมัติการร้องขอ	14/03/2019 10:00

บทที่ 4
ผลการทดลองและการดำเนินงาน

4.1 อุปกรณ์และซอฟต์แวร์ที่ใช้ในการทำโครงงาน

4.1.1 ซอฟต์แวร์ (Software)

1) Xampp เวอร์ชั่น 3.2.3
2) PHP เวอร์ชั่น 5.4.16
3) MySQL
4) Windows 10
5) Visual Studio Code
6) VMware Workstation เวอร์ชั่น 15
7) Putty

4.1.2 ฮาร์ดแวร์ (Hardware)

1) เครื่องคอมพิวเตอร์โน๊ตบุ๊ก CPU 2.3 GHz
2) RAM 8 Gb
3) Hard disk 500 Gb

เครื่องมือและอุปกรณ์ที่ใช้ในการพัฒนาระบบทั้งหมด รวมถึงฐานข้อมูล ถูกพัฒนาขึ้นด้วยภาษา PHP โดยใช้ Bootstrap ซึ่งเป็น Frontend Framework และ MySQL เป็นฐานข้อมูล โดยในบทนี้จะแสดงการทำงานของระบบที่สอดคล้องกับการออกแบบในบทก่อนหน้านี้ โดยจำลองระบบทั้งหมดบนโปรแกรม VMware โดยมีองค์ประกอบดังนี้

1) เครื่อง Web Application ทำหน้าที่เป็น Web Server และ Database Server ใช้ระบบปฏิบัติการ Linux CentOS 7 โดยติดตั้งซอฟต์แวร์ PHP เวอร์ชั่น 5.4.16 และ MySQL
2) เครื่อง LDAP ทำหน้าที่สำหรับ User Authentication ในการเข้าใช้งาน Web Application ใช้ระบบปฏิบัติการ Linux CentOS 7
3) เครื่อง Linux Server ทำหน้าที่เป็นเครื่องทดสอบของระบบปฏิบัติการ Linux ใช้ระบบปฏิบัติการ Linux CentOS 7
4) เครื่อง Windows Server ทำหน้าที่เป็นเครื่องทดสอบของระบบปฏิบัติการ Windows ใช้ระบบปฏิบัติการ Windows Server 2016

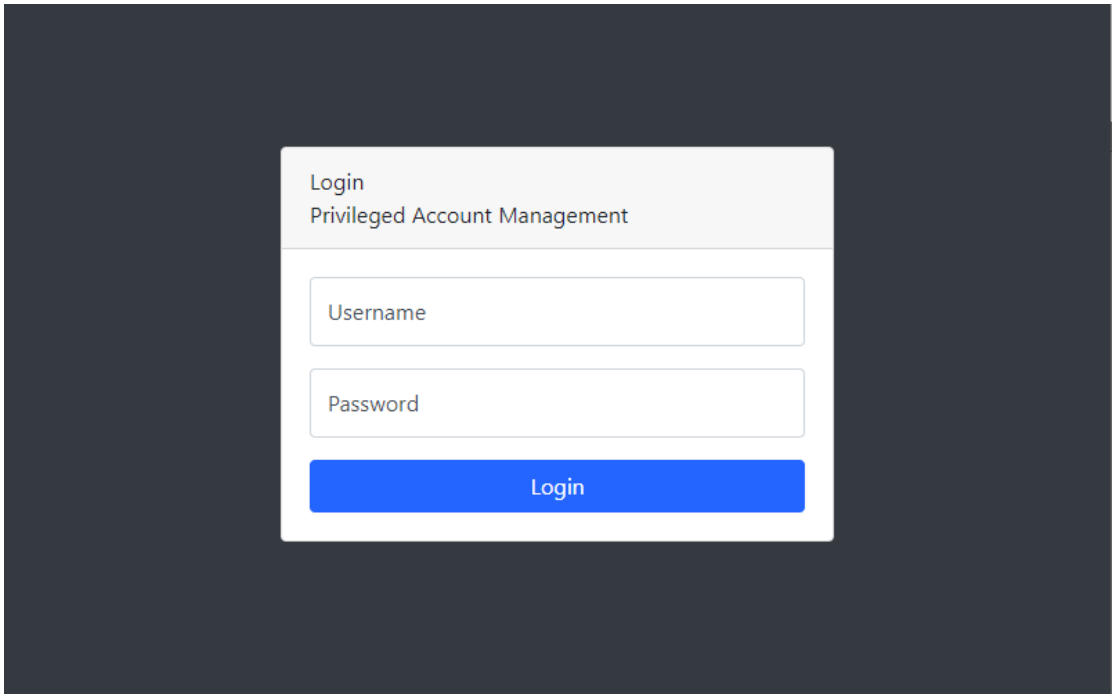
4.2 ผลการดำเนินงาน

4.2.1 หน้าจอเข้าระบบ มีขั้นตอนดังนี้

1) กรอกชื่อผู้ใช้งานในช่อง “Username”

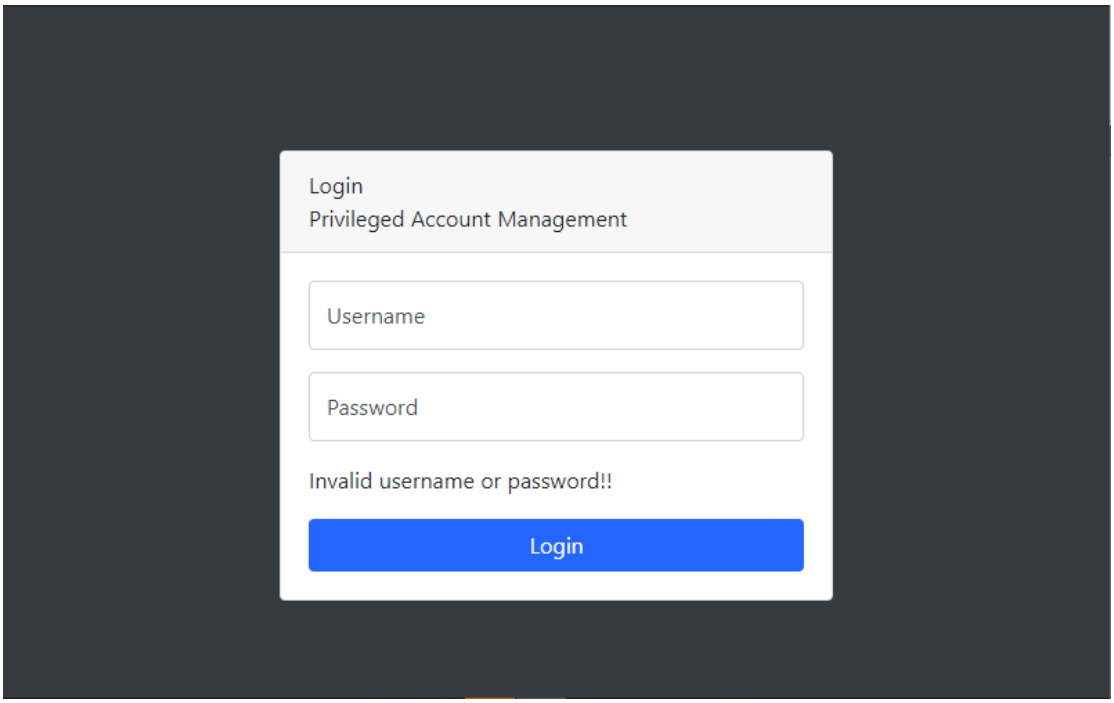
2) กรอกรหัสผ่านในช่อง “Password”

3) กดปุ่ม Login เพื่อใช้งานระบบ ดังรูปที่ 4.1



รูปที่ 4.1 ภาพแสดงหน้าจอการเข้าสู่ระบบ

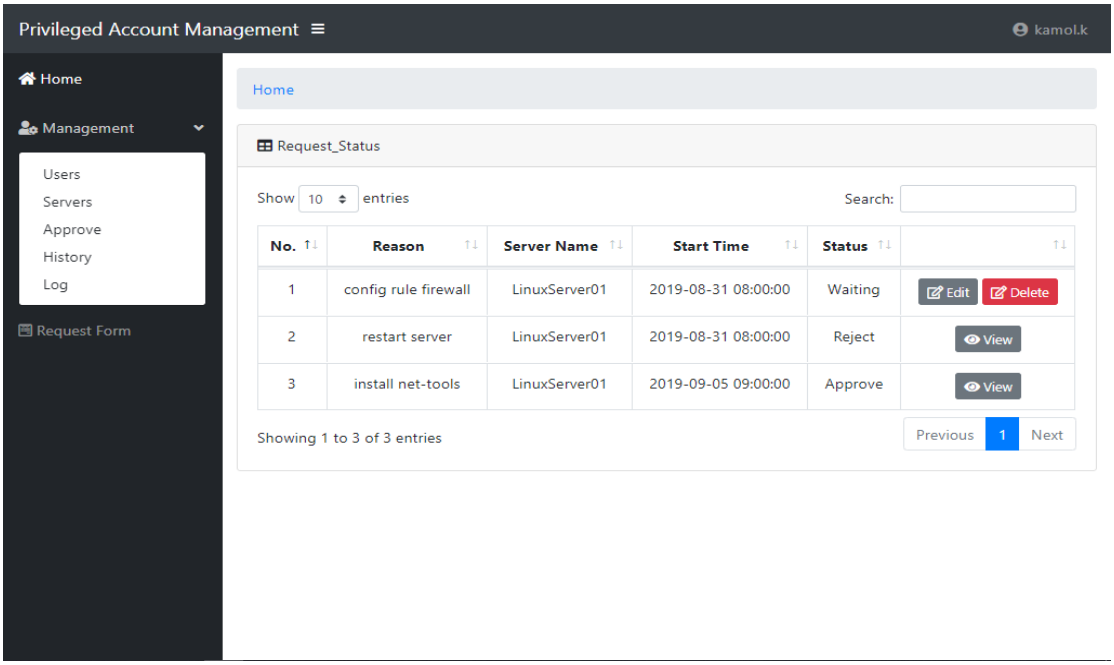
4) ในกรณีที่ผู้ใช้งานกรอกชื่อหรือรหัสผ่านไม่ถูกต้อง ระบบจะทำการแจ้งเตือน ดังรูป 4.2



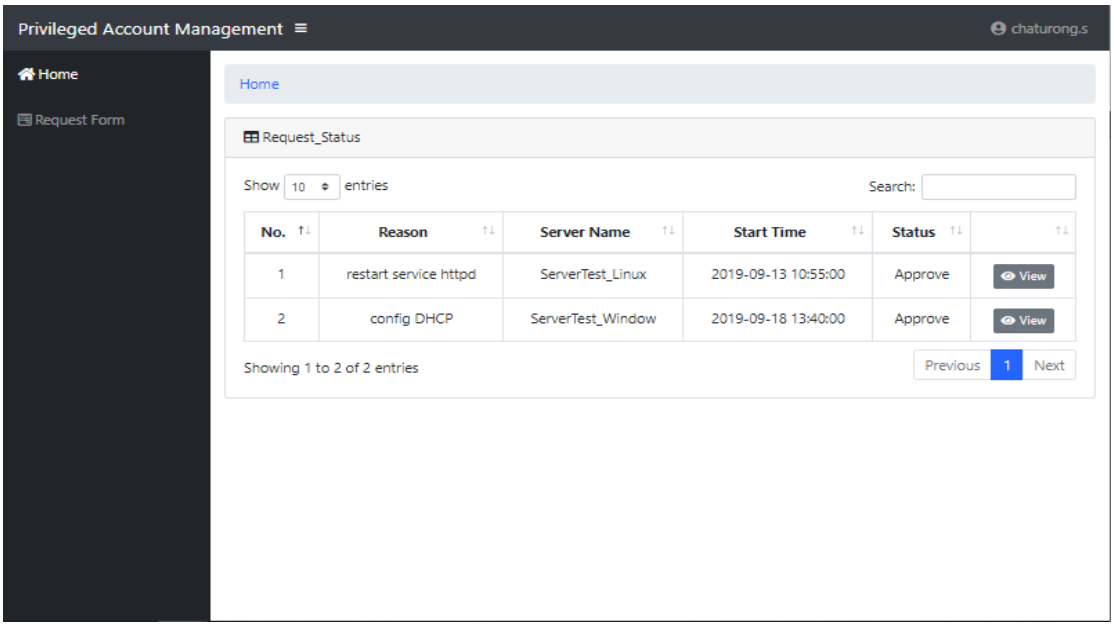
รูปที่ 4.2 ภาพแสดงหน้าจอการเข้าสู่ระบบไม่ถูกต้อง

4.2.2 หน้าจอแรกของระบบ

หลังจากการเข้าสู่ระบบอย่างถูกต้อง ระบบจะดำเนินการตรวจสอบว่าผู้ใช้งานที่ทำการเข้าสู่ระบบเป็นผู้ใช้งาน หรือผู้ดูแลระบบ และจะแสดงรายการตามสิทธิ์ของผู้ใช้งานนั้น หน้าจอแรกจะพบหน้ารายการของผู้ใช้งานนั้นๆ ที่ได้ทำการร้องขอรหัสผ่าน



รูปที่ 4.3 ภาพแสดงหน้าจอแรกหลังเข้าสู่ระบบสำเร็จในส่วนของผู้ดูแลระบบ



รูปที่ 4.4 ภาพแสดงหน้าจอแรกหลังเข้าสู่ระบบสำเร็จในส่วนของผู้ใช้งาน

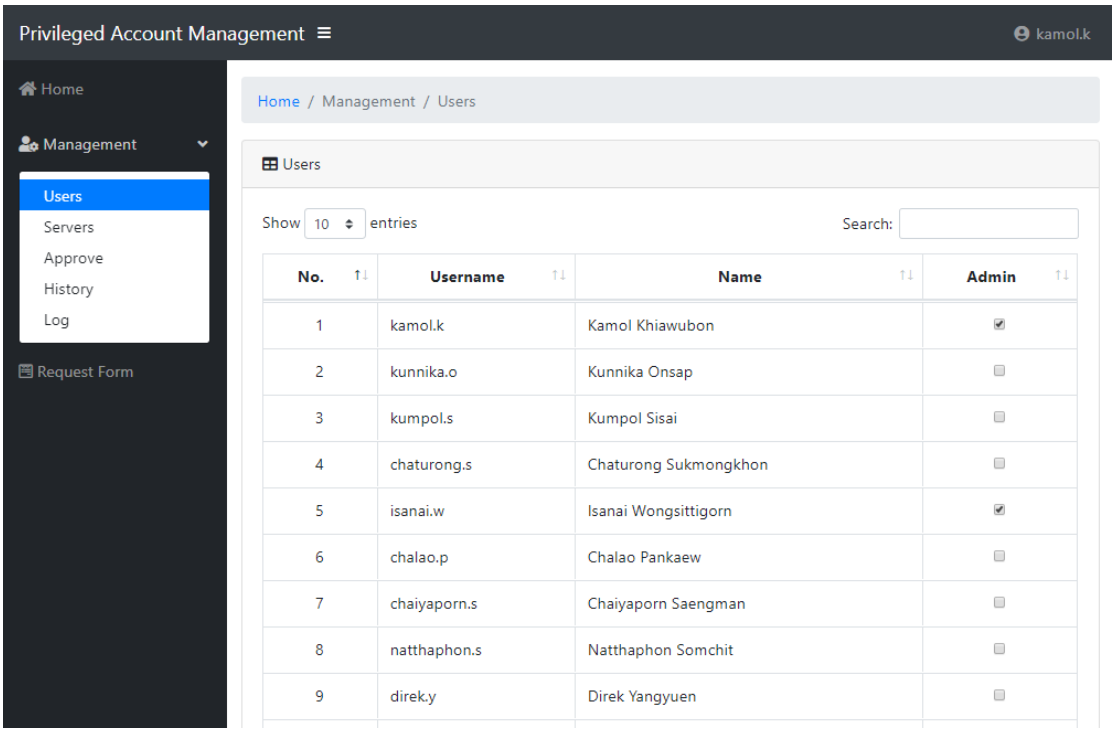
จากรูปที่ 4.3 และ 4.4 สำหรับหน้าแสดงผลหลักจะมีเมนูสำหรับผู้ใช้งาน ซึ่งผู้ใช้งานแต่ละคนสามารถมองเห็นเมนูที่สามารถใช้งานได้ไม่เหมือนกัน ตามแต่สิทธิ์การใช้งานของแต่ละคนที่ได้รับ ดังนี้

1) เมนู Management คือ เมนูสำหรับผู้ที่ได้รับสิทธิ์ในการจัดการระบบ เพื่อเข้ามาจัดการข้อมูลต่างๆ

- Users คือเมนูที่ใช้สำหรับจัดการสิทธิ์ในการเป็นผู้ดูแลระบบ
- Servers คือเมนูที่ใช้สำหรับเพิ่ม แก้ไข หรือลบเซิร์ฟเวอร์ที่มีอยู่ในระบบ
- Approve คือเมนูที่ใช้สำหรับอนุมัติ หรือไม่อนุมัติ คำร้องขอของผู้ใช้งาน
- History คือเมนูที่ใช้ดูประวัติการร้องขอใช้งาน Prevailed Account ทั้งหมด
- Log คือเมนูที่ใช้ตรวจสอบการเข้าสู่ระบบบนเซิร์ฟเวอร์ที่ทำการร้องขอ

2) เมนู Request Form คือเมนูสำหรับผู้ใช้งานสามารถกรอกคำร้องขอรหัสผ่านเพื่อเข้าไปยังเซิร์ฟเวอร์ที่ต้องการ

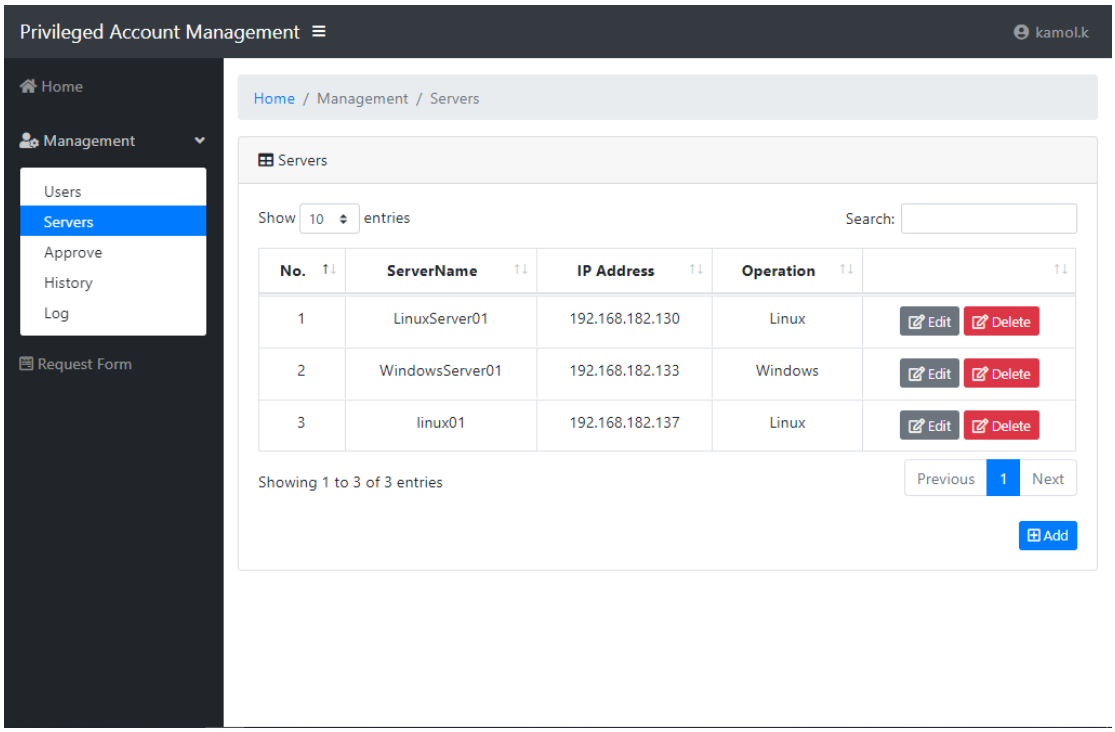
4.2.3 รายการจัดการ Users



รูปที่ 4.5 ภาพหน้าจอแสดงรายการจัดการ Users

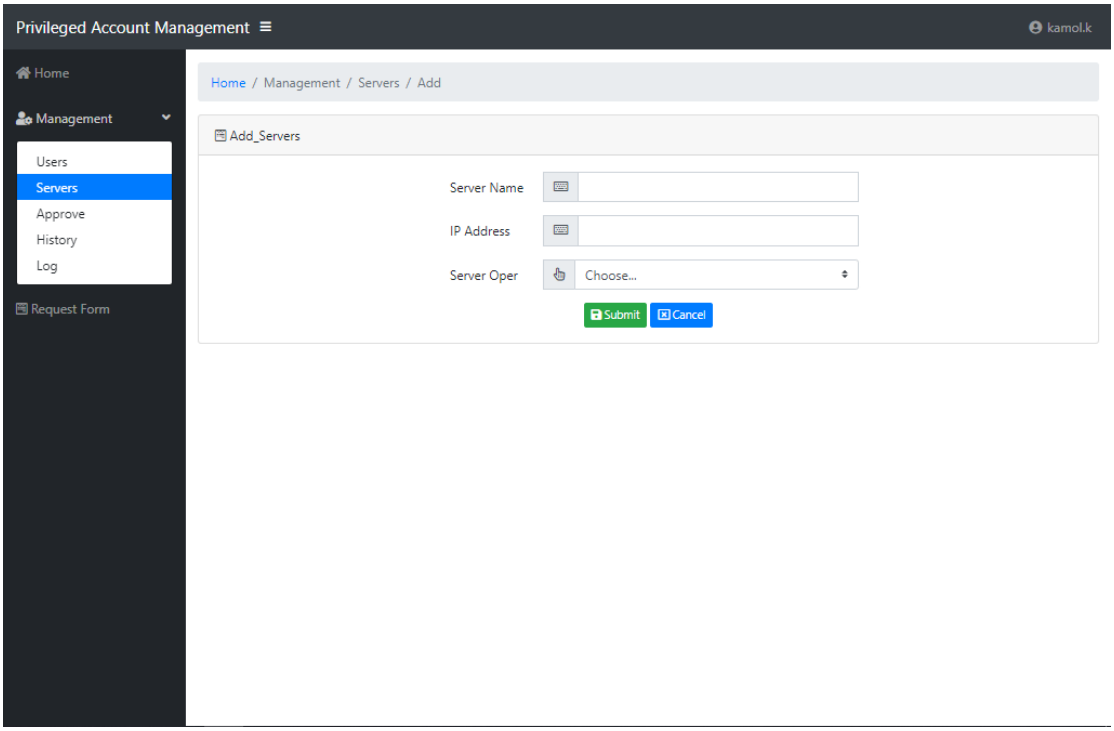
จากรูปที่ 4.5 แสดงรายชื่อผู้ใช้งานของระบบ ที่จะมีเพียงผู้ที่ได้รับสิทธิ์ในการเป็นผู้ดูแลระบบเท่านั้น โดยที่ระบบจะไปดึงข้อมูลของผู้ใช้จาก LDAP มาแสดง ซึ่งผู้ดูแลระบบสามารถกำหนดสิทธิ์ในการจัดการระบบให้กับผู้ใช้งานที่ได้รับมอบหมายได้

4.2.4 รายการจัดการ Servers



รูปที่ 4.6 ภาพหน้าจอแสดงรายการจัดการ Servers

จากรูปที่ 4.6 แสดงหน้าจอการจัดการ Servers ที่จะมีเพียงผู้ที่ได้รับสิทธิ์ในการเป็นผู้ดูแลระบบเท่านั้น ที่สามารถสร้าง แก้ไข และลบ เซิร์ฟเวอร์ในระบบได้ เมื่อคลิกที่ปุ่ม Add จะปรากฏหน้าจอดังรูปที่ 4.7

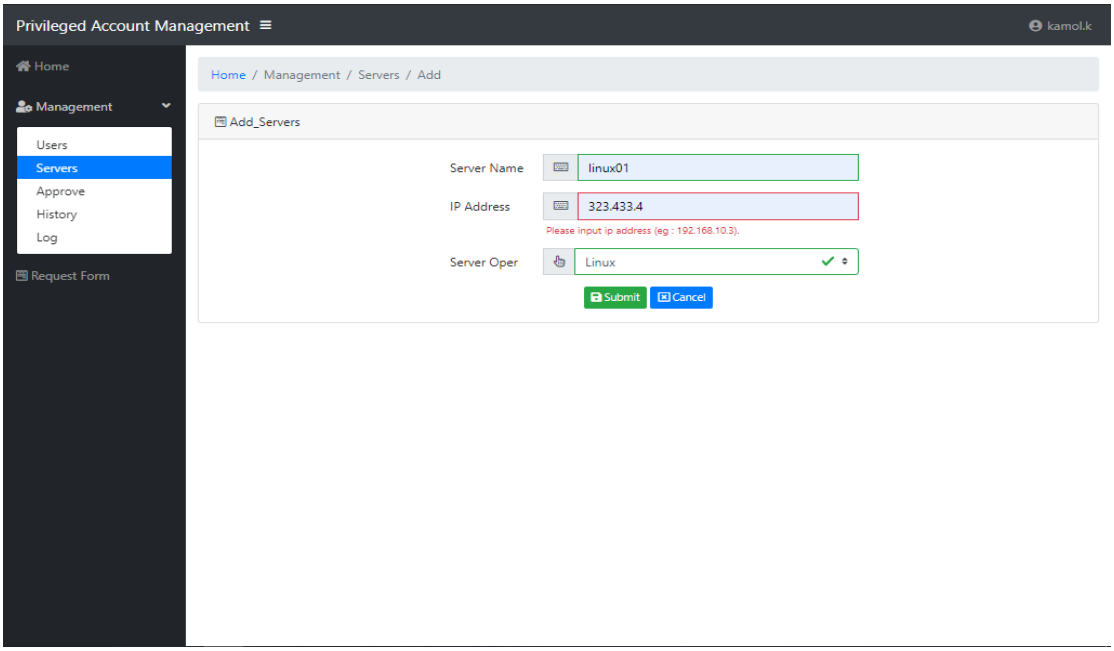


รูปที่ 4.7 ภาพหน้าจอแบบฟอร์มการสร้างเซิร์ฟเวอร์

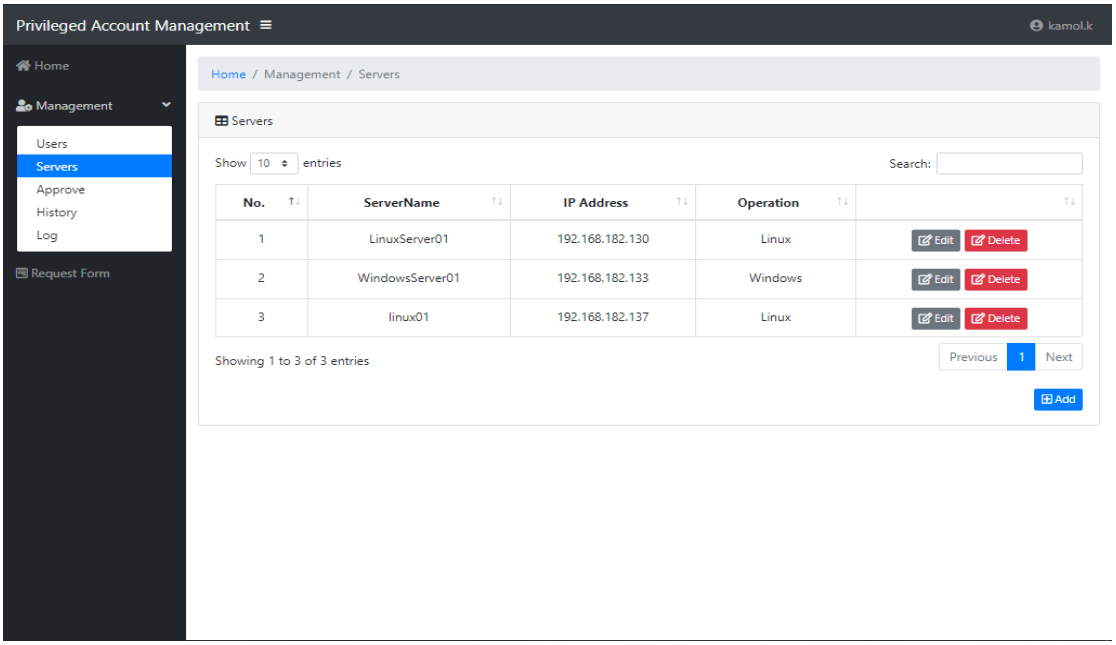
จากรูปที่ 4.7 คือแบบฟอร์มสำหรับสร้างเซิร์ฟเวอร์ใหม่ โดยผู้ดูแลระบบต้องกรอกข้อมูลให้ครบถ้วนทั้งหมด ดังนี้

1) ชื่อของเครื่องเซิร์ฟเวอร์
2) IP Address จะต้องใส่ IP Address ที่สามารถใช้งานได้ และระบบจะมีการตรวจสอบรูปแบบของ IP Address ที่กรอกเข้ามาด้วย
3) ระบบปฏิบัติการของเครื่องเซิร์ฟเวอร์

หากกรอกข้อมูลผิดพลาด จะปรากฏข้อความแสดงข้อผิดพลาดดังรูปที่ 4.8 ถ้าหากกรอกข้อมูลครบถ้วนจะแสดงผลดังรูปที่ 4.9

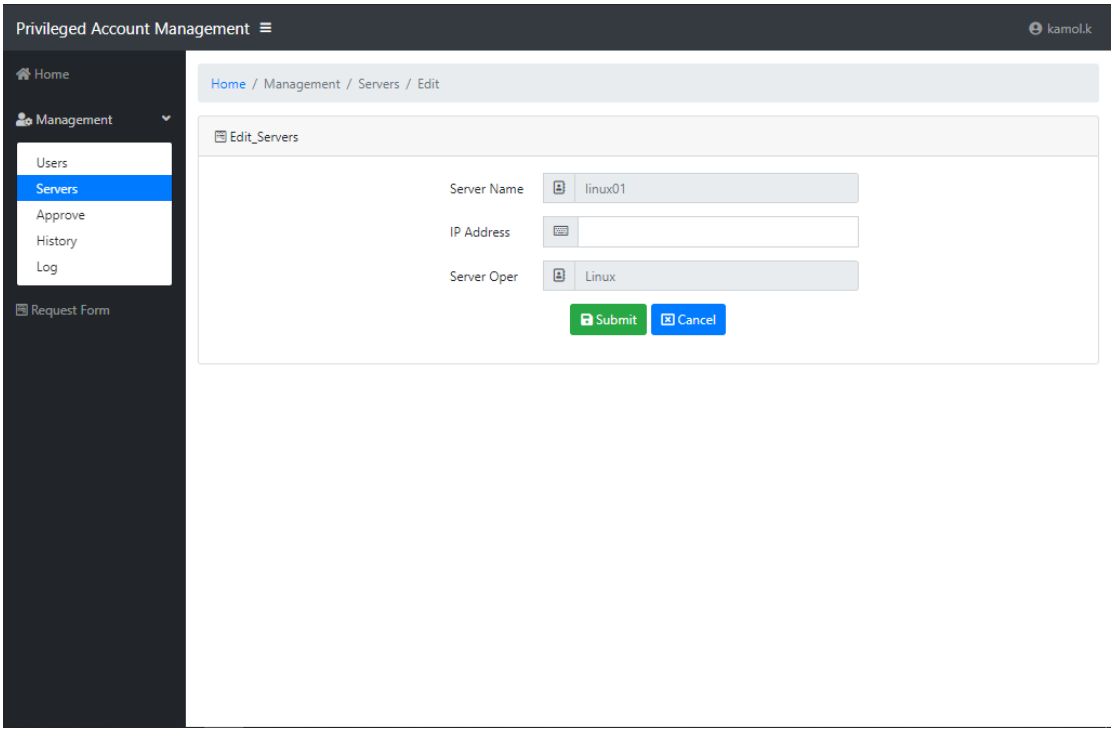


รูปที่ 4.8 ภาพหน้าจอแสดงการป้อนข้อมูลเซิร์ฟเวอร์ผิดพลาด



รูปที่ 4.9 ภาพหน้าจอแสดงหลังการสร้างข้อมูลเซิร์ฟเวอร์

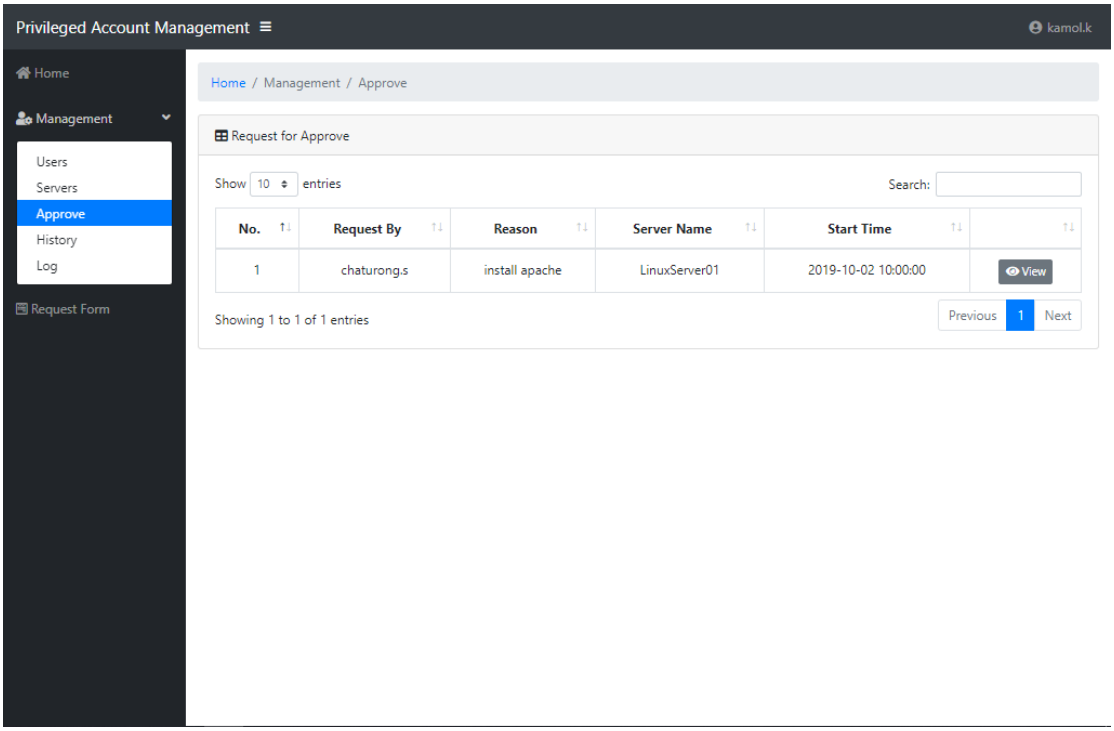
จากรูปที่ 4.9 เมื่อทำการป้อนข้อมูลเซิร์ฟเวอร์เรียบร้อยแล้ว ชื่อเซิร์ฟเวอร์จะปรากฏอยู่ในรายการจัดการ Servers และเมื่อคลิกเข้าปุ่ม Edit จะปรากฏหน้าจอสำหรับแก้ไขข้อมูลเซิร์ฟเวอร์ ดังรูปที่ 4.10



รูปที่ 4.10 ภาพหน้าจอแก้ไขข้อมูลเซิร์ฟเวอร์

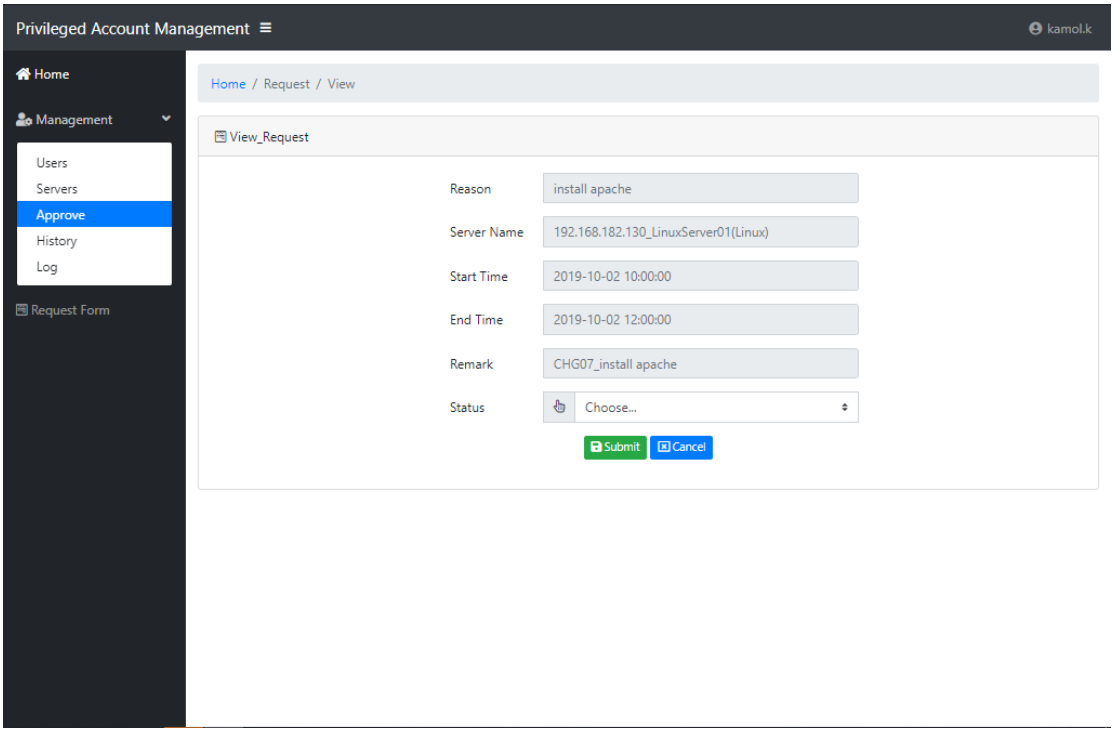
4.2.5 รายการจัดการ Approve

ในหน้านี้ผู้ดูแลระบบสามารถดูคำร้องขอของผู้ใช้งานที่ได้ทำการยื่นคำร้องขอเข้ามา และสามารถที่จะอนุมัติหรือไม่อนุมัติได้



รูปที่ 4.11 ภาพหน้าจอแสดงคำขอใช้งาน Privileged Account จากผู้ใช้

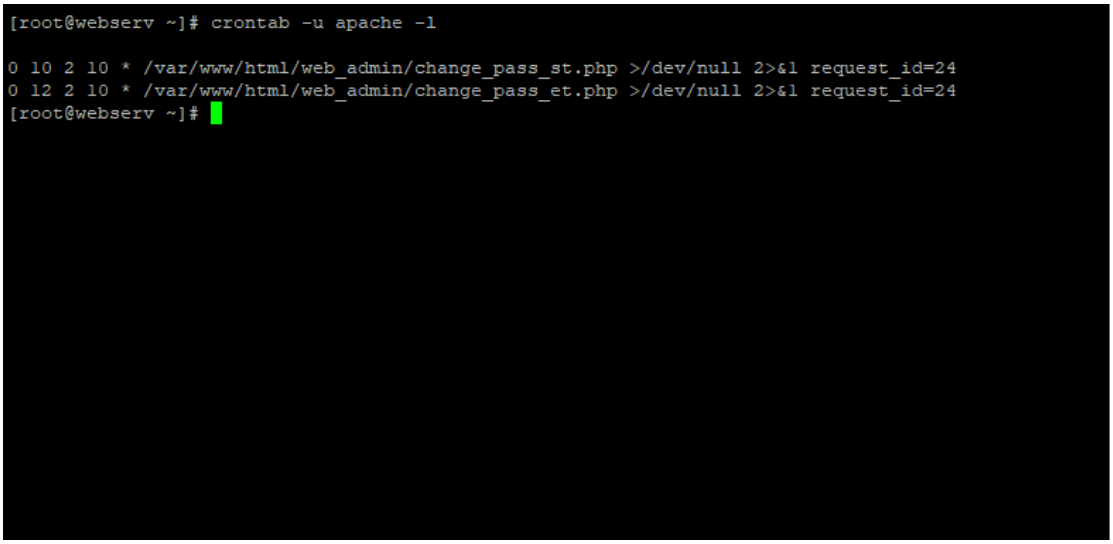
ผู้ดูแลระบบสามารถเลือกดูคำขอใช้งาน Privileged Account โดยสามารถดูรายละเอียดเพิ่มเติมได้ เมื่อคลิกเข้าปุ่ม View จะปรากฏหน้าจอแสดงรายละเอียดคำร้องขอ พร้อมทั้งสามารถอนุมัติหรือไม่อนุมัติ คำขอใช้งาน Privileged Account ได้ ดังรูปที่ 4.12



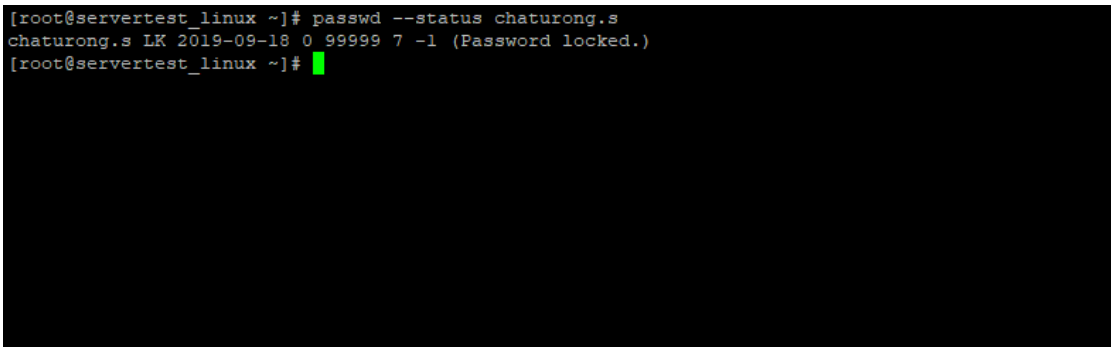
รูปที่ 4.12 ภาพหน้าจอแสดงรายละเอียดคำขอใช้งาน Privileged Account

เมื่อผู้ดูแลระบบทำการเลือกอนุมัติคำขอใช้งาน Privileged Account แล้ว ระบบจะทำงานดังนี้

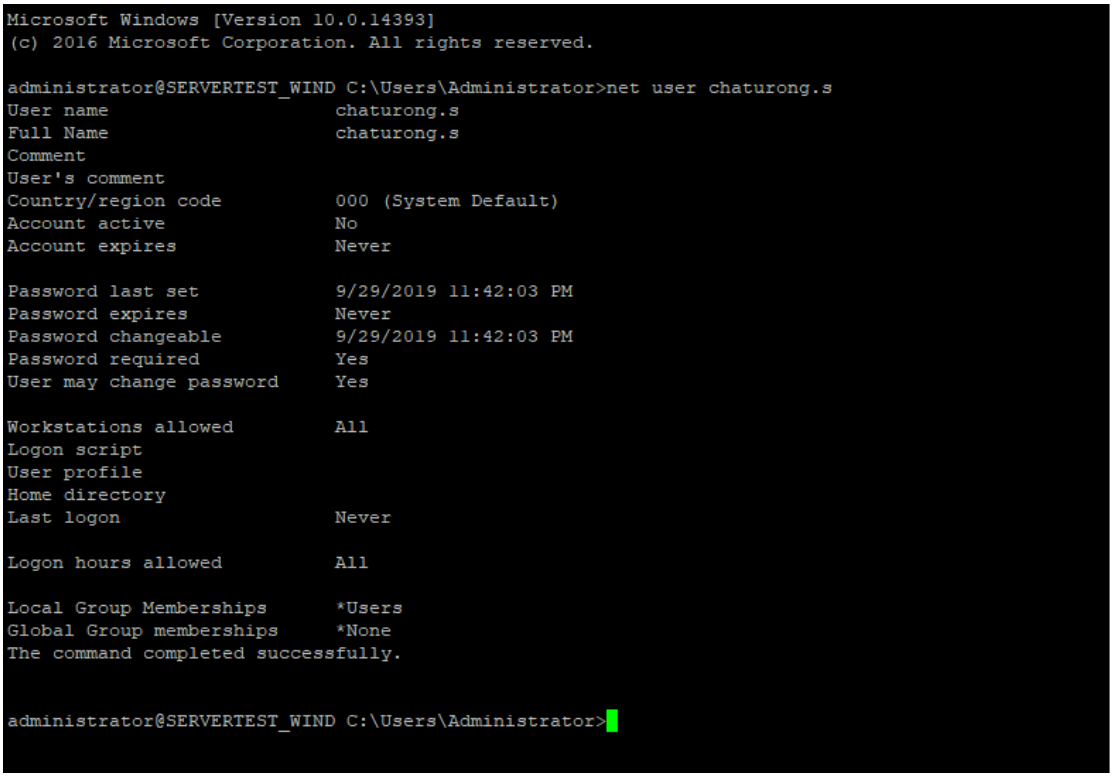
1) ระบบจะทำการสร้างรหัสผ่านเก็บไว้ในฐานข้อมูล
2) ตั้งเวลาเริ่มต้นในการส่งคำสั่งเปลี่ยนรหัสผ่าน และ Enable Account ของผู้ที่ได้รับการอนุมัติ
3) ตั้งเวลาสิ้นสุดในการใช้งานรหัสผ่าน เมื่อถึงเวลาสิ้นสุดระบบจะทำการ Disable หรือ Lock Account ของผู้ที่ได้รับการอนุมัติ และลบข้อมูลการตั้งเวลาของคำร้องขอนั้น
4) ส่ง Alert ไปให้ผู้ที่ทำการร้องขอทราบถึงผลการอนุมัติหรือไม่อนุมัติ ผ่านทาง Email



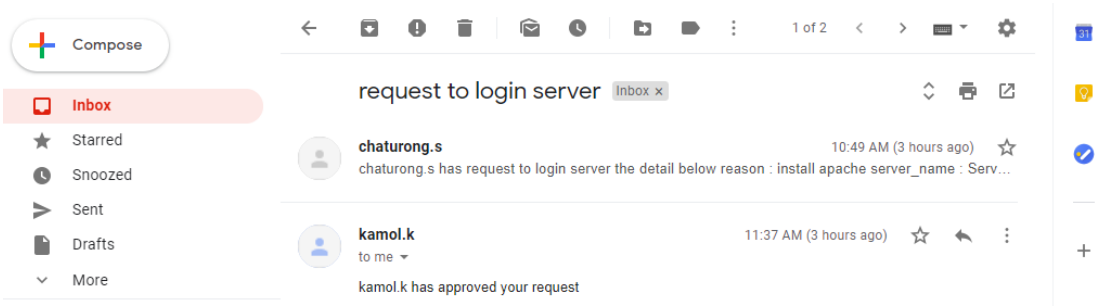
รูปที่ 4.13 ภาพหน้าจอแสดงการตั้งเวลาเริ่มต้น และเวลาสิ้นสุด



รูปที่ 4.14 ภาพหน้าจอแสดงสถานะ Account ของผู้ใช้หลังหมดเวลา ของระบบปฏิบัติการ Linux



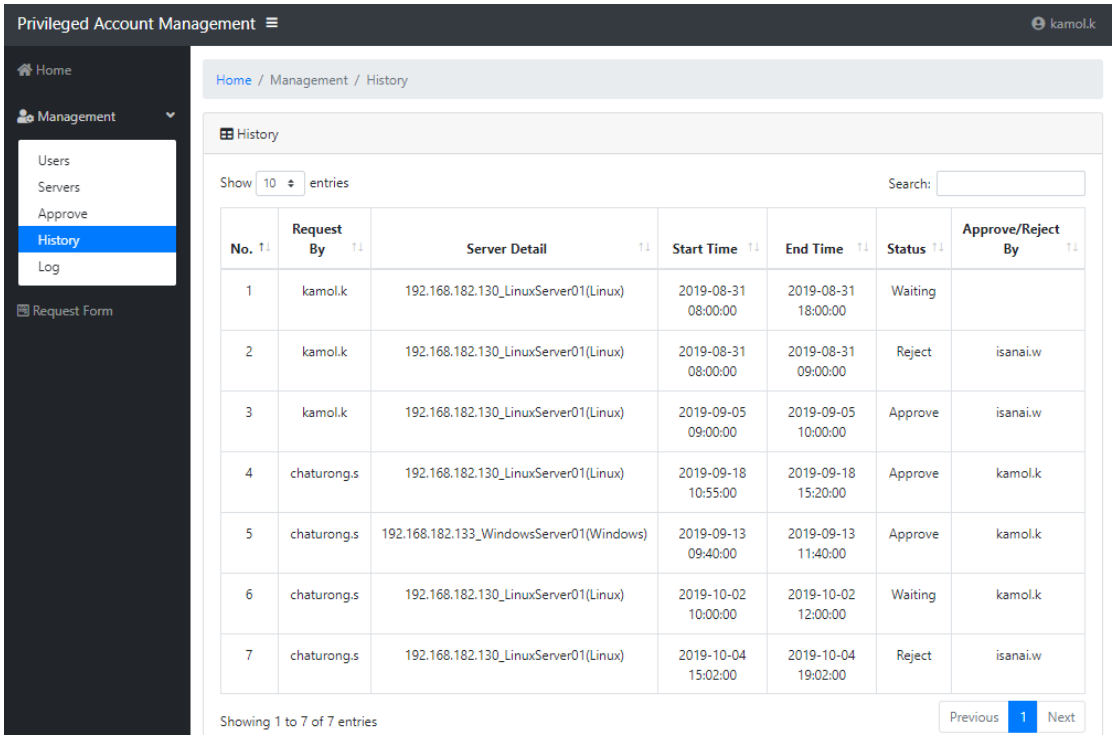
รูปที่ 4.15 ภาพหน้าจอแสดงสถานะ Account ของผู้ใช้หลังหมดเวลา ของระบบปฏิบัติการ Windows



รูปที่ 4.16 ภาพหน้าจอแสดงผลการอนุมัติผ่านทาง Email

4.2.6 รายการจัดการ History

ในหน้านี้ผู้ดูแลระบบสามารถเรียกดูประวัติคำร้องขอใช้งาน Privileged Account ทั้งหมดของผู้ใช้งานที่ได้ทำการยื่นคำร้องเข้ามา



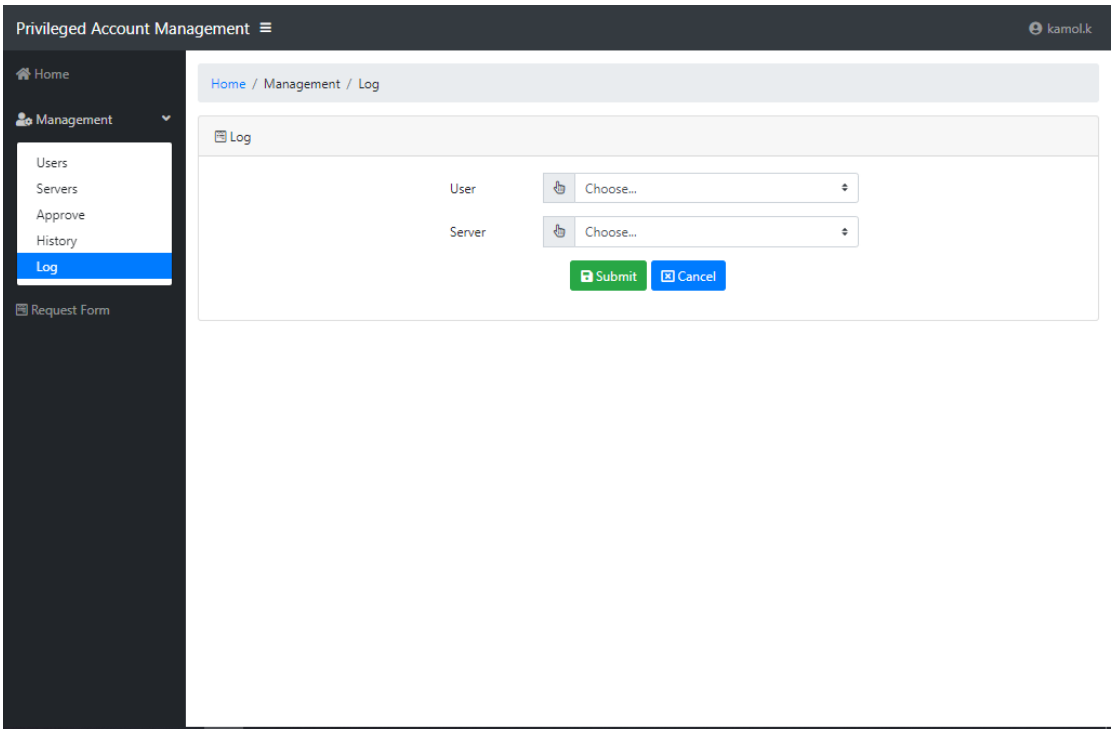
รูปที่ 4.17 ภาพหน้าจอแสดงประวัติคำร้องขอใช้งาน Privileged Account

จากรูปที่ 4.17 แสดงรายละเอียดประวัติคำร้องขอใช้งาน Privileged Account ประกอบด้วย

1) No. คือลำดับของการร้องขอ
2) By คือ Account ของผู้ใช้งานที่ทำการร้องขอใช้งาน Privileged Account
3) Server Detail คือ รายละเอียดของเครื่องเซิร์ฟเวอร์ที่ผู้ใช้งานร้องขอ Privileged Account ขอเข้าใช้โดยจะแสดงข้อมูล IP Address, ชื่อเครื่องเซิร์ฟเวอร์ และระบบปฏิบัติการ
4) Start Time คือ เวลาเริ่มต้นที่ผู้ใช้งานมีความประสงค์ขอใช้งาน
5) End Time คือ เวลาสิ้นสุดที่ผู้ใช้งานมีความประสงค์สิ้นสุดการใช้งาน
6) Status คือ สถานะของคำขอใช้งาน Privileged Account ประกอบด้วย
    - Waiting คือ คำขอใช้งาน Privileged Account นี้ อยู่ในขั้นตอนรอผู้ดูแลระบบตรวจสอบ และพิจารณาคำร้องขอว่าจะอนุมัติหรือไม่อนุมัติ
    - Approve คือ คำขอใช้งาน Privileged Account นี้ ได้รับการอนุมัติจากผู้ดูแลระบบ
    - Reject คือ คำขอใช้งาน Privileged Account นี้ ไม่ได้รับการอนุมัติจากผู้ดูแลระบบ

7) Approve/Reject By คือ Account ของผู้ดูแลระบบที่เป็นคนอนุมัติหรือไม่อนุมัติคำขอ

4.2.7 รายการจัดการ Log



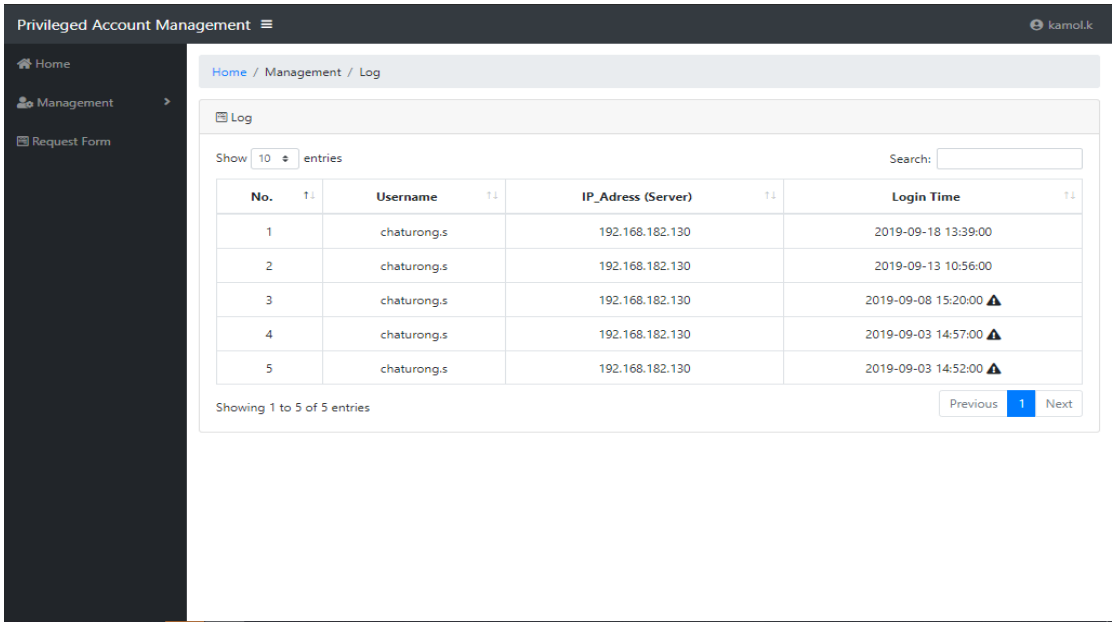
รูปที่ 4.18 ภาพหน้าจอแสดงแบบฟอร์มการกรอกข้อมูลเพื่อตรวจสอบ Log

จากรูปที่ 4.18 ผู้ดูแลระบบสามารถเรียกดูข้อมูลการเข้าสู่ระบบของเครื่องเซิร์ฟเวอร์ปลายทางที่ผู้ใช้งานทำการร้องขอเพื่อตรวจสอบการเข้าสู่ระบบของผู้ใช้งานได้ โดยระบุข้อมูลดังต่อไปนี้

1) User ระบุชื่อของผู้ใช้งานที่ต้องการตรวจสอบประวัติการเข้าสู่ระบบ

2) Server ระบุเครื่องเซิร์ฟเวอร์ที่ต้องการตรวจสอบประวัติการเข้าสู่ระบบ

เมื่อทำการกรอกข้อมูลเรียบร้อยแล้ว ระบบจะทำการส่งคำสั่งเพื่อไปดึงประวัติของเวลาที่ผู้ใช้งานเข้าสู่ระบบไปยังเซิร์ฟเวอร์ที่ระบุข้างต้น และจะนำเวลาที่ได้รับมาไปตรวจสอบกับเวลาที่ได้รับการอนุมัติของผู้ใช้งาน หากเวลาที่เข้าสู่ระบบไม่ใช่เวลาที่ได้รับการอนุมัติจะแสดง Alert ให้ผู้ดูแลระบบทราบ ดังรูปที่ 4.19

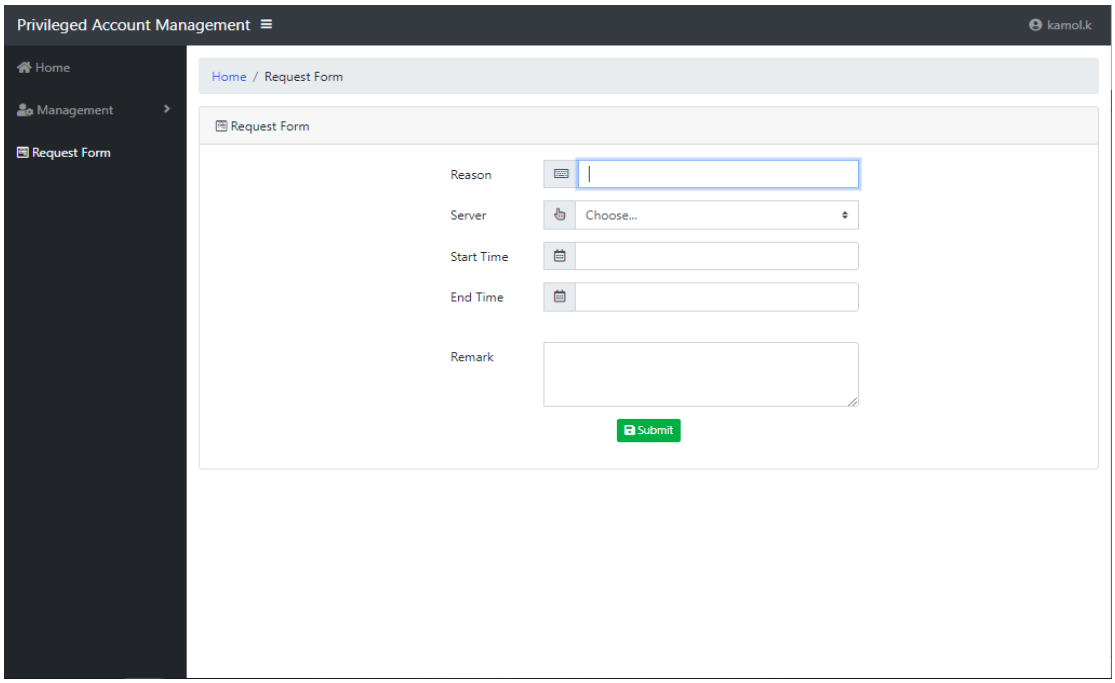


รูปที่ 4.19 ภาพหน้าจอแสดงข้อมูลการเข้าสู่ระบบของเครื่องเซิร์ฟเวอร์

จากรูปที่ 4.19 จะแสดงรายละเอียดแสดงข้อมูลการเข้าสู่ระบบของเครื่องเซิร์ฟเวอร์ ประกอบด้วย

1) No. คือลำดับเวลาในการเข้าเครื่องเซิร์ฟเวอร์
2) Username คือ Account ของผู้ใช้งานที่ทางผู้ดูแลระบบต้องการจะตรวจสอบ
3) IP Address (Server) คือ IP Address ของเครื่องเซิร์ฟเวอร์ที่ทางผู้ดูแลระบบต้องการจะตรวจสอบ
4) Login Time คือ เวลาในที่ผู้ใช้งานเชื่อมต่อเข้าเซิร์ฟเวอร์

4.2.8 แบบฟอร์มสร้างคำขอใช้งาน Privileged Account

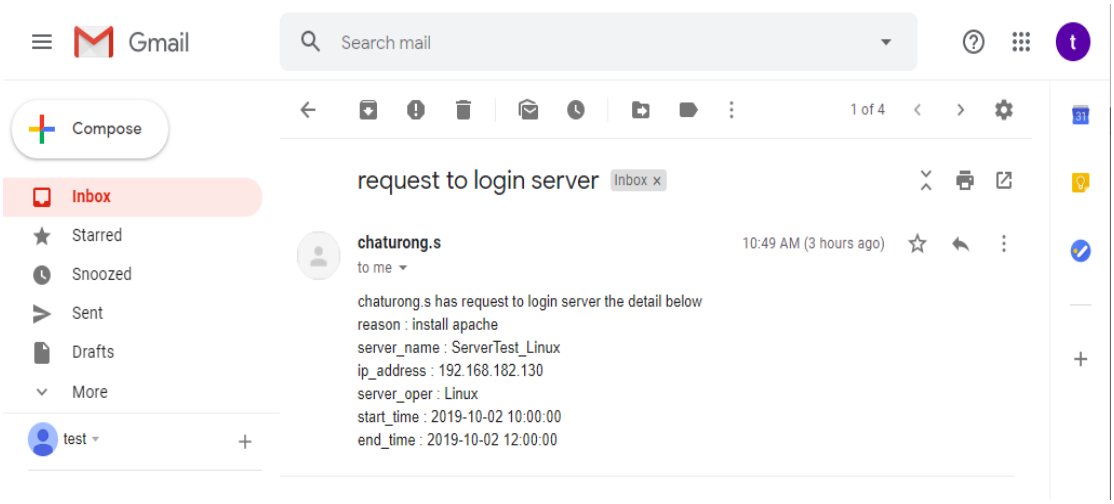


รูปที่ 4.20 ภาพหน้าจอแสดงแบบฟอร์มสร้างคำขอใช้งาน Privileged Account

จากรูปที่ 4.20 คือแบบฟอร์มสร้างคำขอใช้งาน Privileged Account ใหม่ โดยที่จะต้องกรอกข้อมูลให้ครบถ้วนทั้งหมด ดังนี้

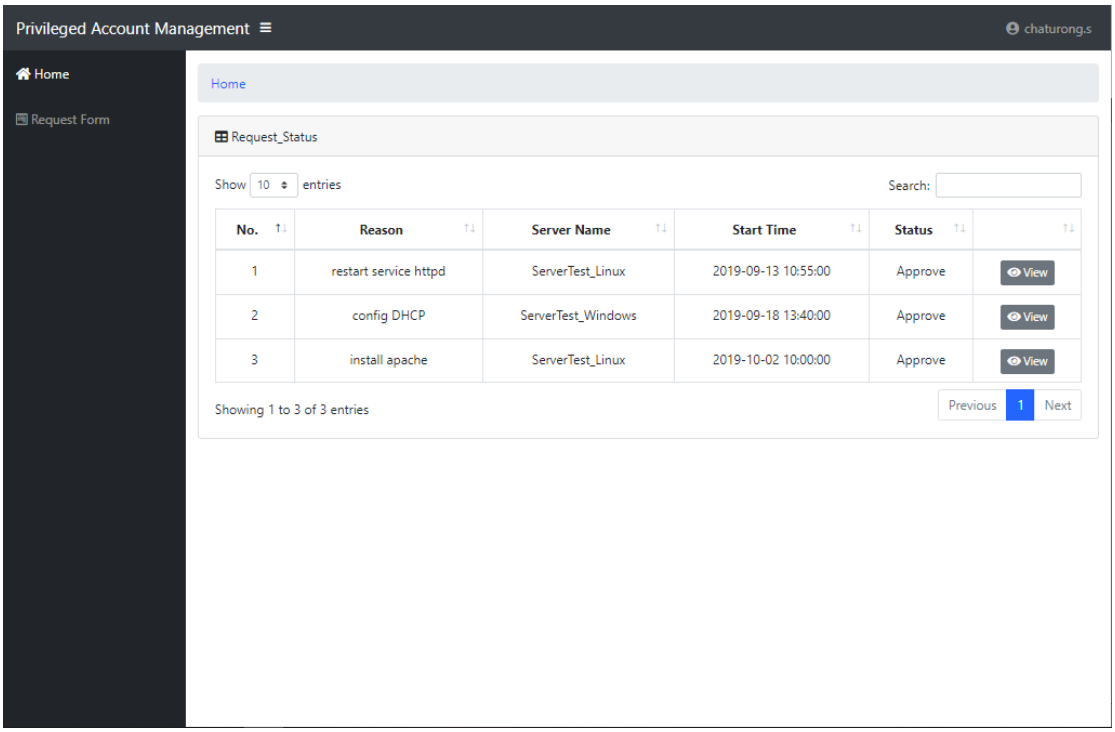
1) เหตุผลในการขอใช้งาน Privileged Account
2) เลือกเซิร์ฟเวอร์ที่ผู้ใช้มีความประสงค์จะขอเข้าใช้งาน โดยระบบจะทำการดึงข้อมูลของเซิร์ฟเวอร์จากฐานข้อมูล และนำมาแสดงผล ดังนี้ IP Address, ชื่อเซิร์ฟเวอร์ และ ชื่อระบบปฏิบัติการ
3) เลือกเวลาเริ่มต้นที่ผู้ใช้งานมีความประสงค์ขอใช้งาน
4) เลือกเวลาสิ้นสุดที่ผู้ใช้งานมีความประสงค์สิ้นสุดการใช้งาน
5) กรอกข้อมูลเพิ่มเติมในการขอใช้งาน Privileged Account

เมื่อกดปุ่ม Submit แล้วระบบจะส่ง Alert ไปยังผู้ดูแลระบบให้ทราบว่ามีการขอใช้งาน Privileged Account ผ่านทาง Email เพื่อให้ผู้ดูแลระบบเข้ามาอนุมัติต่อไป ดังรูปที่ 4.21



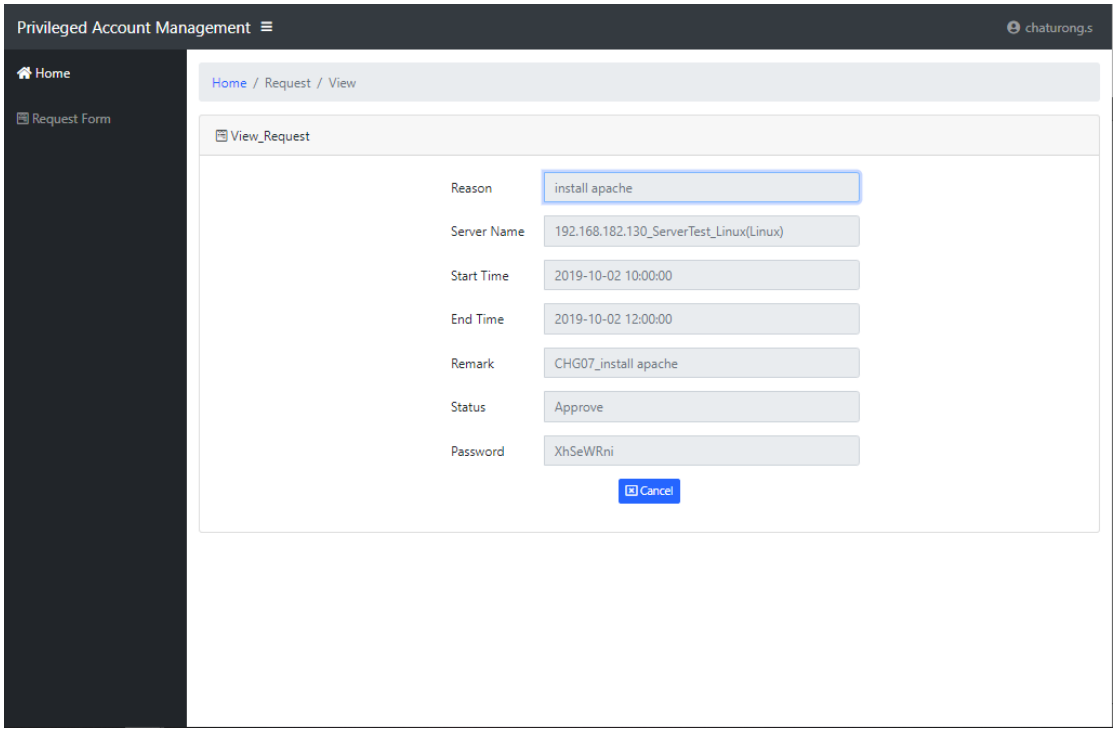
รูปที่ 4.21 ภาพหน้าจอแสดงผลการขอใช้งาน Privileged Account ทาง Email

4.2.9 การติดตามผลของคำขอใช้งาน Privileged Account



รูปที่ 4.22 ภาพหน้าจอแสดงผลของคำขอใช้งาน Privileged Account

จากรูปที่ 4.22 ผู้ใช้งานที่ได้ทำการร้องขอใช้งาน Privileged Account สามารถดูผลจากหน้า Home ซึ่งจะแสดงผลข้อมูลที่ผู้ใช้งานนั้นๆ ได้ทำการร้องขอผ่านทางระบบ ถ้าหากได้รับการอนุมัติคำขอแล้วสามารถกด View เพื่อดูรหัสผ่านในการเข้าใช้งานเซิร์ฟเวอร์ ดังรูปที่ 4.23



รูปที่ 4.23 ภาพหน้าจอแสดงรหัสผ่าน

จากรูปที่ 4.23 ระบบจะไปดึงข้อมูลต่างๆจากฐานข้อมูล ออกมาแสดงโดยรหัสผ่านที่ไปดึงมาได้จะถูกเข้ารหัสไว้ ระบบจะทำการถอดรหัสก่อนแล้วจึงนำรหัสที่ได้มาแสดงผล

บทที่ 5
สรุปผลการดำเนินการ

5.1 สรุปผลการดำเนินการ

การออกแบบและพัฒนาระบบการจัดการผู้ใช้งานบัญชีผู้มีสิทธิ์สูง (Privileged Account Management) จะเพิ่มประสิทธิภาพในเรื่องการเบิกจ่ายรหัสผ่านที่มีความล่าช้า, กระบวนการทำงานในการรับรหัสผ่านจากผู้บริหารระบบ และ เพิ่มความปลอดภัยในการได้รับรหัสผ่าน เนื่องจากกระบวนการในการเบิกสิทธิ์จะสามารถใช้บริการผ่านทาง Web Application ทำให้ทั้งผู้ใช้งานและผู้ดูแลระบบมีความสะดวกสบายมากขึ้น หลังจากผู้ใช้งานดำเนินการกรอกคำขอใช้ Privileged Account แล้ว ระบบจะส่ง Email เพื่อให้ผู้ดูแลระบบทราบและเข้ามาอนุมัติ หากผู้ดูแลระบบอนุมัติคำขอใช้งาน แล้วระบบจะสร้างรหัสผ่าน และเข้ารหัสลับเก็บไว้ในฐานข้อมูล อีกทั้งระบบยังจัดเก็บข้อมูลการเข้าใช้งานเซิร์ฟเวอร์ที่ผู้ใช้งานทำการร้องขอ ซึ่งทำให้ผู้ดูแลระบบสามารถตรวจสอบได้ว่าเวลาที่ผู้ใช้งานร้องขอเข้าเซิร์ฟเวอร์เครื่องที่ต้องการตรวจสอบ ผู้ใช้งานได้เข้าไปในช่วงเวลานั้นหรือไม่ เพื่อเพิ่มความปลอดภัยในการเข้าสู่ระบบภายในองค์กร

5.2 ปัญหาและอุปสรรคในการดำเนินการ

5.2.1 การพิสูจน์ว่าผู้ใช้งานที่ได้ทำการร้องขอสามารถดูรหัสผ่านที่ได้รับจากระบบ โดยที่รหัสผ่านที่ได้รับการอนุมัติจะแสดงให้เห็นได้เพียงแค่ผู้ใช้งานที่เป็นคนร้องขอมาเท่านั้น จึงได้นำค่า Hash มาช่วยในการพิสูจน์

5.2.2 การพัฒนาระบบมีการเชื่อมต่อไปยังระบบปฏิบัติการ Linux และ Windows มีความซับซ้อน และมีการเรียกใช้งานคำสั่งในการเชื่อมต่อที่แตกต่างกัน ภาษาที่ใช้ในการส่งคำสั่งหรือเก็บข้อมูลการเชื่อมต่อที่แตกต่างกัน ทำให้การพัฒนาระบบแต่ละหัวข้อใช้เวลานาน

5.2.3 เครื่องคอมพิวเตอร์ที่ใช้งานเป็บ SATA Disk และ ขนาดของ RAM ที่น้อย เมื่อต้องทำการทดสอบโดยเปิดระบบทุกเครื่องขึ้นมา เครื่องจะทำงานช้า

5.3 ข้อเสนอแนะ

5.3.1 ควรเพิ่มการบริหารจัดการบัญชีผู้มีสิทธิ์สูงในระบบต่างๆ เช่น ฐานข้อมูล ระบบเครือข่ายเป็นต้น

5.3.2 ควรปรับแต่งหน้าตาของระบบให้ดูน่าใช้งานมากยิ่งขึ้น

5.3.3 ควรเพิ่มการตรวจสอบการกระทำของผู้ใช้ที่ได้รับสิทธิ์ในเครื่องเซิร์ฟเวอร์ที่ผู้ใช้ร้องขอใช้งาน

เอกสารอ้างอิง

[1] จตุชัย แพงจันทร์, Master in Security 3rd Edition. นนทบุรี : ไอดีซีฯ, 2558.

[2] PHP Documentation [Online] Available: https://www.php.net/docs.php

[3] Privileged Account Management Best Practices [Online] Available: https://www.netwrix.com/privileged_account_management_best_practices.html

[4] How To Execute Shell Commands with PHP Exec and Examples [Online] Available : https://www.poftut.com/execute-shell-commands-php-exec-examples/

[5] PowerShell – Everything you wanted to know about Event Logs and then some [Online] Available: https://evotec.xyz/powershell-everything-you-wanted-to-know-about-event-logs/

[6] Linux Crontab Command [Online] Available: https://www.computerhope.com/unix/ucrontab.htm