**เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่**

**ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO**

**นาย ศุภวิชญ์ แซ่ลิ่ม**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต**

**สาขาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

**ปีการศึกษา 2567**

**สิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่

ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO

นาย ศุภวิชญ์ แซ่ลิ่ม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2567

สิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

A Web Application for Configuring and Assessing Configuration

Vulnerabilities of Cisco Network Devices

MR. SUPAWIT SAELIM

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE BACHELOR’S DEGREE OF ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

FACULTY OF INDUSDTRIAL TECHNOLOGY AND MANAGEMENT

KING MONGKUT’S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

2024

COPYRIGHT OF KING MONGKUT’S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

|  |
| --- |
| **ใบรับรองปริญญานิพนธ์**  **คณะเทคโนโลยีและการจัดการอุตสาหกรรม**  **มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**  เรื่อง เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO  โดย นายศุภวิชญ์ แซ่ลิ่ม  ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมบัณฑิต  สาขาวิชาวิศวกรรมสารสนเทศและเครือข่าย  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**คณบดี  (ผู้ช่วยศาสตราจารย์ ดร.กฤษฎากร บุดดาจันทร์)  คณะกรรมการสอบปริญญานิพนธ์  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**ประธานกรรมการ  (ผู้ช่วยศาสตราจารย์ ดร.สุพาภรณ์ ซิ้มเจริญ)  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**กรรมการ  (อาจารย์ ดร.ศรายุทธ ธเนศสกุลวัฒนา)  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**กรรมการ  (อาจารย์ ดร.วัชรชัย คงศิริวัฒนา) |

ชื่อ : นายศุภวิชญ์ แซ่ลิ่ม

ชื่อปริญญานิพนธ์ : เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO

สาขาวิชา : วิศวกรรมสารสนเทศและเครือข่าย

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

อาจารย์ที่ปรึกษาปริญญานิพนธ์ : อาจารย์ ดร.วัชรชัย คงศิริวัฒนา

ปีการศึกษา : 2567

**บทคัดย่อ**

ปัจจุบันเทคโนโลยีเครือข่ายที่รวดเร็ว และมีประสิทธิภาพมีความสำคัญต่อการดำเนินธุรกิจ และการสื่อสารในชีวิตประจำวันของมนุษย์ การจัดการอุปกรณ์เครือข่ายจึงกลายเป็นสิ่งสำคัญสำหรับองค์กรต่าง ๆ อุปกรณ์เครือข่ายที่สำคัญ เช่น Router และ Switch เป็นส่วนประกอบหลัก ในการทำให้ระบบเครือข่ายสื่อสารกันได้อย่างมีประสิทธิภาพ นอกจากนี้การจัดการและตรวจสอบการตั้งค่าของอุปกรณ์ดังกล่าวค่อนข้างมีความซับซ้อน และใช้เวลานาน ก่อให้เกิดปัญหา เช่น การตั้งค่าผิดพลาด ไม่ถูกต้องตามหลักการกำหนดค่า รวมถึงการตรวจสอบสถานะของอุปกรณ์ที่ไม่เป็นไปตามที่ควร นำไปสู่ความเสี่ยงต่อการเกิดช่องโหว่ในระบบเครือข่าย ดังนั้นการมีเครื่องมือที่สามารถช่วยตรวจสอบ และจัดการการตั้งค่าของอุปกรณ์เครือข่ายจึงเป็นสิ่งจำเป็น ส่งผลให้ผู้ดูแลระบบสามารถควบคุม ตรวจสอบสถานะของอุปกรณ์ได้อย่างมีประสิทธิภาพ นอกจากนี้ การสามารถเข้าถึงและจัดการอุปกรณ์เครือข่ายได้ในแบบเรียลไทม์ ยังเป็นสิ่งสำคัญที่ช่วยให้ผู้ดูแลระบบสามารถแก้ไขปัญหาได้อย่างรวดเร็ว ลดระยะเวลาที่อุปกรณ์จะไม่พร้อมทำงาน ดังนั้นการพัฒนาโปรแกรมที่สามารถรองรับการเชื่อมต่อกับอุปกรณ์เครือข่ายใหม่ การตั้งค่า และการตรวจสอบช่องโหว่ในการกำหนดค่าของอุปกรณ์จึงมีความสำคัญมากในการปรับปรุงประสิทธิภาพ ความปลอดภัยของเครือข่ายในองค์กรในการใช้งานในปัจจุบัน

(ปริญญานิพนธ์มีจำนวนทั้งสิ้น ... หน้า)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**อาจารย์ที่ปรึกษาปริญญานิพนธ์

Name : MR.Supawit Saelim

Project Title : A Web Application for Configuring and Assessing Configuration

Vulnerabilities of Cisco Network Devices

Major Field : Information and Network Engineering

: King Mongkut’s University of Technology North Bangkok

Project Advisor : DR.Watcharachai Kongsiriwattana

Academic Year : 2024

**Abstract**

In today's world, fast and efficient network technology is essential for business operations and daily human communication. Managing network devices has become crucial for various organizations. Key network devices, such as routers and switches, serve as the main components for enabling effective network communication. Moreover, managing and verifying the configuration of these devices is often complex and time-consuming, leading to issues such as misconfigurations, non-compliance with configuration standards, and improper device status monitoring. These issues can result in vulnerabilities within the network system. Therefore, having tools to assist in verifying and managing the configuration of network devices is essential. Such tools allow system administrators to effectively control and monitor the status of these devices. Additionally, real-time access to and management of network devices is vital, enabling administrators to quickly resolve issues and minimize device downtime. As a result, developing software capable of supporting new network device connections, configurations, and vulnerability assessments is critically important for enhancing the efficiency and security of organizational networks in today's usage scenarios.

(Total …. Page)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Project Advisor

**กิตติกรรมประกาศ**

โครงงานเรื่อง "เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO**"** นี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือและการสนับสนุนจากหลายท่าน ข้าพเจ้าขอกราบขอบพระคุณ ดร.วัชรชัย คงศิริวัฒนา อาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำ องค์ความรู้ และข้อเสนอแนะที่มีประโยชน์อย่างยิ่งตลอดการดำเนินโครงงาน

ข้าพเจ้าขอขอบคุณ ผศ. สุพาภรณ์ ซิ้มเจริญ ในฐานะประธานสอบโครงงาน และ อ. ดร. ศรายุทธ ธเนศสกุลวัฒนา ในฐานะกรรมการสอบโครงงาน ที่ได้กรุณาให้คำแนะนำและข้อเสนอแนะอันทรงคุณค่าที่ช่วยปรับปรุงโครงงานให้มีความสมบูรณ์มากยิ่งขึ้น

นอกจากนี้ ข้าพเจ้าขอขอบคุณครอบครัว เพื่อน ๆ และผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่ได้ให้กำลังใจและการสนับสนุนในทุกด้านจนสามารถดำเนินโครงงานนี้ได้สำเร็จ

สุดท้ายนี้ ข้าพเจ้าหวังเป็นอย่างยิ่งว่าโครงงานนี้จะเป็นประโยชน์แก่การศึกษาและการพัฒนาด้านเทคโนโลยีเครือข่ายต่อไป

ศุภวิชญ์ แซ่ลิ่ม

**สารบัญ**

หน้า

บทคัดย่อภาษาไทย ข

บทคัดย่อภาษาอังกฤษ ค

กิตติกรรมประกาศ ง

สารบัญภาพ ช

สารบัญตาราง ฌ

บทที่ 1 บทนำ

* 1. ความเป็นมาและความสำคัญ 1
  2. วัตถุประสงค์ของโครงงาน 2
  3. ขอบเขตของการจัดทำโครงงานพิเศษ 3
  4. วิธีการดำเนินงาน 13
  5. ทรัพยากรที่ใช้ ...
  6. ประโยชน์ที่คาดว่าจะได้รับ ...

บทที่ 2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง ...

* 1. โปรแกรม Visual Studio Code ...
  2. MongoDB
  3. ...

**บทที่ 1**

**บทนำ**

* 1. **ความเป็นมาและความสำคัญของปัญหา**

ปัจจุบันเทคโนโลยีเครือข่ายที่รวดเร็ว และมีประสิทธิภาพมีความสำคัญต่อการดำเนินธุรกิจ และการสื่อสารในชีวิตประจำวันของมนุษย์ การจัดการอุปกรณ์เครือข่ายจึงกลายเป็นสิ่งสำคัญสำหรับองค์กรต่าง ๆ อุปกรณ์เครือข่ายที่สำคัญ เช่น Router และ Switch เป็นส่วนประกอบหลัก ในการทำให้ระบบเครือข่ายสื่อสารกันได้อย่างมีประสิทธิภาพ นอกจากนี้การจัดการและตรวจสอบการตั้งค่าของอุปกรณ์ดังกล่าวค่อนข้างมีความซับซ้อน และใช้เวลานาน ก่อให้เกิดปัญหา เช่น การตั้งค่าผิดพลาด ไม่ถูกต้องตามหลักการกำหนดค่า รวมถึงการตรวจสอบสถานะของอุปกรณ์ที่ไม่เป็นไปตามที่ควร นำไปสู่ความเสี่ยงต่อการเกิดช่องโหว่ในระบบเครือข่าย และการกำหนดค่านั้น จำเป็นต้องกำหนดค่าทีละอุปกรณ์ ทำให้ผู้ดูแลระบบเสียเวลาในการกำหนดค่าของตัวอุปกรณ์ ดังนั้นการมีเครื่องมือที่สามารถช่วยตรวจสอบ และจัดการการตั้งค่าของอุปกรณ์เครือข่ายจึงเป็นสิ่งจำเป็น ส่งผลให้ผู้ดูแลระบบสามารถควบคุม ตรวจสอบสถานะของอุปกรณ์ได้อย่างมีประสิทธิภาพ นอกจากนี้ การสามารถเข้าถึงและจัดการอุปกรณ์เครือข่ายได้ในแบบเรียลไทม์ และช่วยลดเวลากำหนดค่าให้กับผู้ดูระบบ ยังเป็นสิ่งสำคัญที่ช่วยให้ผู้ดูแลระบบสามารถแก้ไขปัญหาได้อย่างรวดเร็ว ลดระยะเวลาที่อุปกรณ์จะไม่พร้อมทำงาน

ดังนั้นการพัฒนาโปรแกรมที่สามารถรองรับการเชื่อมต่อกับอุปกรณ์เครือข่ายใหม่ การตั้งค่า และการตรวจสอบช่องโหว่ในการกำหนดค่าของอุปกรณ์จึงมีความสำคัญมากในการปรับปรุงประสิทธิภาพ ความปลอดภัยของเครือข่ายในองค์กรในการใช้งานในปัจจุบัน

* 1. **วัตถุประสงค์ของโครงงาน**
     1. เพิ่มความสะดวกให้ผู้ดูระบบสามารถตั้งค่าเริ่มต้นของอุปกรณ์ได้ผ่านเว็บแอปพลิเคชัน โดยไม่ต้องใช้ command line ในการตั้งค่าเริ่มต้นของอุปกรณ์เครือข่าย
     2. พัฒนาโปรแกรมที่สามารถกำหนดค่าอุปกรณ์เครือข่ายหลายตัวพร้อมกันเพื่อประหยัดเวลา และสามารถตรวจสอบการตั้งค่าของอุปกรณ์ในเครือข่ายนั้น ๆ ได้
     3. เพื่อเพิ่มความปลอดภัยให้กับอุปกรณ์เครือข่ายด้วยความสามารถในการตรวจสอบการตั้งค่าของอุปกรณ์ และแสดงจุดที่ควรปรับปรุง
  2. **ขอบเขตของการทำโครงงานพิเศษ**
     1. โปรแกรมรองรับการเชื่อมต่อกับอุปกรณ์เครือข่ายใหม่ที่ยังไม่ได้ตั้งค่า โดยใช้การเชื่อมต่อผ่าน serial port และสามารถทำการตั้งค่า ค่าต่าง ๆ โดยผู้ใช้สามารถกรอกค่าเพื่อให้ตรงกับความต้องการได้ เพื่อให้พร้อมสำหรับการควบคุมผ่านโปรแกรม

1.3.1.1 กำหนด Console port ที่เชื่อมต่อกับตัวอุปกรณ์

1.3.1.2 กำหนด Hostname

1.3.1.3 กำหนด Privilege Password

1.3.1.4 กำหนด Username และ password สำหรับการ SSH

1.3.1.5 กำหนด Domain-name

1.3.1.6 ระบุ Interface ที่ใช้ในการตั้งค่า IP Address เพื่อการเชื่อมต่อในเครือข่าย สามารถเลือกได้ ดังนี้

1.3.1.6.1 ระบุ IP Address

1.3.1.6.2 DHCP

* + 1. โปรแกรมรองรับการตั้งค่าอุปกรณ์เครือข่ายหลายตัวในเวลาเดียวกันโดยใช้เทคนิค threading ซึ่งช่วยลดเวลาที่ใช้ในการกำหนดค่าอุปกรณ์หลายเครื่องพร้อมกัน ดังนี้

1.3.2.1 การสร้างหรือลบ VLANs ในหลายเครื่องพร้อมกัน

1.3.2.2 กำหนดค่า SNMP location, contact ในหลายเครื่องพร้อมกัน

* + 1. ผู้ใช้สามารถเพิ่มรายละเอียดของอุปกรณ์ที่จะตั้งค่าผ่านโปรแกรมและบันทึกลงระบบฐานข้อมูล โดยมีรายละเอียดดังนี้

1.3.3.1 ชื่ออุปกรณ์ (Name)

1.3.3.2 ที่อยู่ IP Address

1.3.3.3 ชื่อผู้ใช้ (Username)

1.3.3.4 รหัสผ่านของผู้ใช้ (Password)

1.3.3.5 รหัสผ่านเข้าสู่โหมด Privilege ของอุปกรณ์ (Secret Password)

* + 1. ผู้ใช้สามารถดูรายละเอียดของอุปกรณ์ที่เพิ่มเข้าสู่ระบบฐานข้อมูลทั้งหมดได้และสามารถลบได้
    2. โปรแกรมรองรับการตั้งค่าอุปกรณ์โดยมีความสามารถในการตั้งค่า ค่าต่าง ๆ ดังนี้
       1. หัวข้อการตั้งค่าพื้นฐาน
          1. กำหนด hostname
          2. กำหนด Secret Password
          3. กำหนด Banner
       2. หัวข้อการตั้งค่า Interface เครือข่าย
          1. สามารถกำหนดค่าได้ทั้ง IPv4, IPv6
          2. Interface ที่ต้องการ
          3. Interface สามารถเลือกได้ ดังนี้

Manual

DHCP

* + - * 1. สถานะของ Interface (Up/Down)
        2. สามารถลบที่อยู่ของ Interface ได้ (IP Address)
        3. สามารถเลือก Duplex ได้ (half/full)
      1. หัวข้อการตั้งค่า VLAN สามารถกำหนดค่าต่าง ๆ ได้ ดังนี้
         1. กำหนดค่าสร้าง VLAN
         2. สามารถเปลี่ยนชื่อของ VLAN นั้น ๆ ได้
         3. สามารถกำหนดสถานะของ VLAN นั้น ๆ ได้ (Enable/Disable)
         4. สามารถลบ VLAN ได้
         5. สามารถกำหนด interface ให้เป็นสถานะ Access หรือ Trunk ได้
      2. หัวข้อการตั้งค่าการจัดการ มีรายละเอียด ดังนี้
         1. การตั้งค่า Console line

รหัสผ่าน

ระยะเวลาหมดการเชื่อมต่อ (Session)

เปิดปิดใช้งานการบันทึกข้อมูลแบบซิงโครนัส

สามารถกำหนดวิธีการตรวจสอบความถูกต้องของรหัสผ่านได้

Local User

Console Password

* + - * 1. การตั้งค่า VTY line

รหัสผ่าน

ระยะเวลาหมดการเชื่อมต่อ (Session)

โปรโตคอลที่ยอมรับ ดังนี้

SSH

TELNET

ALL

NONE

เปิดปิดใช้งานการบันทึกข้อมูลแบบซิงโครนัส

สามารถกำหนดวิธีการตรวจสอบความถูกต้องของรหัสผ่านได้

Local User

VTY Password

* + - * 1. การตั้งค่าเทคโนโลยี protocol DHCP

ชื่อ Pool

ขนาดเครือข่าย

IP ที่ไม่ต้องการให้บริการ (IP Exclude)

ค่าเราเตอร์ค่าเริ่มต้น (Default Gateway)

ค่าที่อยู่ DNS เซิร์ฟเวอร์

ชื่อโดเมน

* + - * 1. การตั้งค่าเทคโนโลยี protocol NTP

ที่อยู่ของ NTP เซิร์ฟเวอร์

เขตเวลา (Time zone)

* + - * 1. การตั้งค่าเทคโนโลยี protocol SNMP

กำหนดค่า Community string แบบ read-only

กำหนดค่า Community string แบบ read-write

กำหนดค่า Location

กำหนดค่า Contact

* + - * 1. เปิดปิดการตั้งค่าเทคโนโลยี protocol CDP
        2. เปิดปิดการตั้งค่าเทคโนโลยี protocol LLDP
      1. หัวข้อการตั้งค่าเทคโนโลยี Spanning Tree Protocols
         1. การเลือกเวอร์ชันและโหมดของ protocol ได้ดังนี้

Per-Vlan spanning tree mode (pvst)

Per-Vlan rapid spanning tree mode (rapid-pvst)

* + - * 1. กำหนดให้อุปกรณ์ตัวนั้นเป็น Root (primary/secondary) ใน VLAN หมายเลข ID อะไร
        2. สามารถระบุ interface ในการเปิดปิดฟังก์ชัน Postfast
        3. เทคโนโลยี BPDU-Guard จะเปิดใช้อัตโนมัติกับ interface ที่ระบุในฟังก์ชัน Postfast
      1. หัวข้อการตั้งค่าการรวมกลุ่ม Aggregation Protocols
         1. เทคโนโลยี Port Aggregation Protocol (PAgP)

สามารถกำหนด interface

สามารถกำหนดหมายเลขของการรวมกลุ่มของ interface นั้น (Channel-group Number)

สามารถกำหนดโหมดได้ทั้งแบบ Desirable และ Auto

* + - * 1. เทคโนโลยี Link Aggregation Protocol (LAcP)

สามารถกำหนด interface

สามารถกำหนดหมายเลขของการรวมกลุ่มของ interface นั้น (Channel-group Number)

สามารถกำหนดโหมดได้ทั้งแบบ Active และ Passive

* + - * 1. สามารถลบ Port group นั้น ๆ ได้
    1. หัวข้อการกำหนด Routing protocols
       1. การกำหนด routing แบบ Static Route
          1. ระบุเครือข่ายปลายทางได้สูงสุด 4 เครือข่าย
          2. สามารถกำหนด routing แบบ Default Route ได้
          3. สามารถลบ routing ที่กำหนดไว้ได้
       2. การกำหนด routing โปรโตคอล OSPF
          1. สามารถกำหนด Process ID ได้
          2. สามารถกำหนด Router ID ของอุปกรณ์นั้น ๆ ได้
          3. กำหนดเครือข่ายได้สูงสุด 3 เครือข่ายต่อการตั้งค่าหนึ่งครั้ง
          4. สามารถลบ Process ID ได้
          5. สามารถลบ routing ที่กำหนดไว้ได้
       3. การกำหนด routing โปรโตคอล RIPv2
          1. สามารถกำหนดเครือข่ายได้
          2. สามารถเปิดปิดฟังก์ชัน Auto-Summary
          3. สามารถลบ routing ที่กำหนดไว้ได้
          4. สามารถเปิดปิดการทำงานของตัว Routing Protocols ได้
       4. การกำหนดเส้นทางโปรโตคอล EIGRP
          1. สามารถกำหนด AS Number ได้
          2. สามารถกำหนด Router ID ของอุปกรณ์นั้น ๆ ได้
          3. กำหนดเครือข่ายได้สูงสุด 3 เครือข่ายต่อการตั้งค่าหนึ่งครั้ง
          4. สามารถลบ AS Number ได้
          5. สามารถลบ routing ที่กำหนดไว้ได้
    2. โปรแกรมสามารถคืนค่าโรงงานการตั้งค่าของตัวอุปกรณ์นั้นได้ และจะถูกลบออกจากฐานข้อมูล
    3. โปรแกรมสามารถแสดงการกำหนดค่าของอุปกรณ์นั้น ๆ ได้ โดยผู้ใช้สามารถเลือกคำสั่งที่ต้องการให้แสดงได้
    4. โปรแกรมมีการตรวจสอบการตั้งค่า (configuration) ของอุปกรณ์เครือข่ายในด้านความปลอดภัย ดังนี้
       1. สามารถตรวจสอบหาช่องโหว่ของการตั้งค่าอุปกรณ์โดยอัตโนมัติ ดังนี้
          1. ช่องโหว่ด้านรหัสผ่าน
          2. การเข้ารหัสรหัสผ่าน
          3. สถานะพอร์ตเครือข่ายที่ไม่ได้ใช้งาน
          4. การตั้งค่าการหมดเวลาการเชื่อมต่อ
          5. การตรวจสอบโปรโตคอลการเข้าถึงระยะไกล
    5. โปรแกรมสามารถตรวจสอบรายละเอียดของอุปกรณ์นั้น ๆ มีรายละเอียดต่าง ๆ ดังนี้
       1. ประเภทของอุปกรณ์ Router หรือ Switch
       2. ชื่อของอุปกรณ์
       3. รายละเอียดคำอธิบาย (Description) ของอุปกรณ์
       4. ระยะเวลาการทำงาน (Uptime) ของอุปกรณ์
       5. ข้อมูลการติดต่อ (Contact) ของอุปกรณ์
       6. สถานที่ตั้ง (Location) ของอุปกรณ์
       7. ชื่อ Interfaces ทั้งหมดที่มีของอุปกรณ์นั้น ๆ
  1. **วิธีการดำเนินงาน**
     1. ศึกษาโครงสร้าง Flask และการพัฒนา API เพื่อรองรับการเชื่อมต่อกับอุปกรณ์เครือข่าย
     2. ศึกษาการตั้งค่าอุปกรณ์สวิตช์ Layer 2, Layer 3 และเราเตอร์ด้วย Netmiko
     3. ศึกษาการใช้งาน MongoDB เพื่อจัดเก็บข้อมูลการตั้งค่าอุปกรณ์
     4. พัฒนาฟังก์ชันการเชื่อมต่อและตั้งค่าอุปกรณ์เครือข่าย เช่น VLAN, Routing, และ Spanning Tree Protocol
     5. พัฒนาระบบตรวจสอบความปลอดภัยการตั้งค่าด้วย ai\_password\_with\_re
     6. ออกแบบ User Interface โดยใช้ render\_template (Flask) และ HTML/CSS
     7. สร้าง API เพื่อส่งและรับข้อมูลจาก Front-end ในรูปแบบ JSON
     8. ทดสอบการตั้งค่าและจัดการเครือข่ายด้วยฟังก์ชันใน device\_config
     9. พัฒนาส่วน Back-end เพื่อเชื่อมต่อกับ MongoDB และรองรับการ Query ข้อมูล
     10. ใช้ Threading และ Scheduler เพื่อจัดการ Task ที่ต้องทำงานพร้อมกันหรือทำงานตามเวลา
     11. สร้างและปรับปรุงฟังก์ชันที่เกี่ยวข้องกับ Routing เช่น Static Route, OSPF, และ EIGRP
     12. ทดสอบระบบ Front-end และ Back-end ให้สามารถทำงานร่วมกันได้อย่างสมบูรณ์
     13. ตรวจสอบและแก้ไขปัญหาในโค้ดที่เกี่ยวข้องกับการเชื่อมต่อและตั้งค่าอุปกรณ์
     14. ปรับปรุงหน้าตาและฟังก์ชันการทำงานของ User Interface
     15. นำระบบไปทดลองใช้งานจริงในสถานการณ์จำลอง
     16. รวบรวมผลการทดสอบ แก้ไขข้อผิดพลาด และจัดทำเอกสารโครงงาน
  2. **ทรัพยากรที่ใช้**
     1. ด้านฮาร์ดแวร์
        1. คอมพิวเตอร์ส่วนบุคคล (PC) หรือ Laptop
        2. อุปกรณ์เครือข่าย เช่น Router, Switch
        3. สาย Serial
        4. สาย UTP
     2. ด้านซอฟต์แวร์
        1. Python ภาษาโปรแกรมหลักในการพัฒนาโปรแกรม
        2. Flask Framework สำหรับพัฒนาเว็บแอปพลิเคชัน
        3. EVE-NG ซอฟต์แวร์จำลองการทำงานของอุปกรณ์เครือข่ายในสภาพแวดล้อม เสมือนจริง
        4. Netmiko ไลบรารี Python สำหรับเชื่อมต่อและตั้งค่าอุปกรณ์เครือข่ายผ่าน SSH
        5. Pymongo ไลบรารี Python สำหรับการจัดการฐานข้อมูล MongoDB
        6. Scheduler โมดูลสำหรับจัดการการทำงานของ Task ที่ต้องทำงานตามเวลา
        7. MongoDB ระบบฐานข้อมูล NoSQL ที่ใช้เก็บข้อมูลการตั้งค่าอุปกรณ์
  3. **ประโยชน์ที่คาดว่าจะได้รับ**
     1. ทำให้ผู้ดูแลระบบสามารถตั้งค่าเริ่มต้นอุปกรณ์ได้จริง
     2. ทำให้ผู้ดูแลระบบสามารถทำการตั้งค่าอุปกรณ์เครือข่ายได้หลายตัวพร้อมกัน และสามารถตรวจสอบการตั้งค่าของอุปกรณ์ได้จริง
     3. ทำให้ผู้ดูแลระบบสามารถตรวจสอบความปลอดภัยเกี่ยวกับการตั้งค่าของอุปกรณ์ได้จริง

**บทที่ 2**

**แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง**

* 1. **โปรแกรม Visual Studio Code**

Visual Studio Code (VS Code) เป็นโปรแกรม Code Editor ที่พัฒนาโดย Microsoft ซึ่งออกแบบมาเพื่อช่วยในการเขียน แก้ไข และตรวจสอบความผิดพลาดของโค้ดอย่างมีประสิทธิภาพ โปรแกรมนี้มีจุดเด่นในด้านความเบา รองรับการทำงานหลายแพลตฟอร์มทั้ง Windows, macOS และ Linux และรองรับภาษาคอมพิวเตอร์หลายภาษา เช่น JavaScript, Python, TypeScript, HTML, CSS, C++, Java, PHP, YAML และอื่น ๆ อีกมากมาย พร้อมทั้งมี ส่วนขยาย (Extensions) ที่สามารถเพิ่มศักยภาพการทำงาน เช่น GitHub Copilot ซึ่งเป็น AI Pair Programmer ช่วยแนะนำและเขียนโค้ดอย่างชาญฉลาด ช่วยให้ผู้ใช้งานสามารถพัฒนาโค้ดได้รวดเร็วและมีประสิทธิภาพมากขึ้น, Python Extension ที่รองรับฟีเจอร์ต่าง ๆ ของการพัฒนา Python เช่น IntelliSense, Linting, Refactoring และการจัดรูปแบบโค้ด, Prettier Formatter สำหรับจัดระเบียบโค้ดให้อ่านง่าย รองรับหลายภาษา รวมถึง Live Preview ที่ช่วยให้ผู้พัฒนาเว็บไซต์สามารถดูผลลัพธ์แบบเรียลไทม์ขณะเขียน HTML ได้ VS Code มีลิขสิทธิ์แบบ MIT License ซึ่งให้ใช้งานฟรีโดยไม่มีข้อจำกัด และยังสามารถใช้งานบนแพลตฟอร์มออนไลน์ที่เรียกว่า VS Code for Web ช่วยให้สามารถเขียนโค้ดได้ทุกที่โดยไม่ต้องติดตั้งโปรแกรมบนเครื่องคอมพิวเตอร์ ด้วยคุณสมบัติทั้งหมดนี้ทำให้ VS Code เป็นหนึ่งในโปรแกรม Code Editor ที่ได้รับความนิยมสูงสุดในปัจจุบันและเป็นเครื่องมือสำคัญสำหรับนักพัฒนาทั่วโลก (*Visual Studio Code หรือ VS Code คืออะไร ทำอะไรได้บ้าง - WebDoDee*, 2024)

* 1. **โปรแกรม Flask – เครื่องมือพัฒนาเว็บ**

การพัฒนาเว็บไซต์ด้วย Python โดยใช้ Flask ถือเป็นจุดเริ่มต้นที่ดีสำหรับผู้ที่สนใจด้าน Web Development ซึ่งเป็นสายงานที่ได้รับความนิยมในตลาดแรงงาน เนื่องจาก Flask เป็นเฟรมเวิร์คที่มีความเรียบง่าย ไม่ซับซ้อน และให้อิสระในการเขียนโค้ดอย่างมาก เหมาะสำหรับการสร้างเว็บไซต์, เว็บแอปพลิเคชัน และ API นอกจากนี้ การพัฒนาเว็บด้วย Flask ยังเน้นการใช้โครงสร้างโฟลเดอร์มาตรฐาน เช่น โฟลเดอร์ `templates` สำหรับเก็บไฟล์ HTML, โฟลเดอร์ `static` สำหรับจัดเก็บไฟล์ CSS, JavaScript, และไฟล์มีเดียต่าง ๆ รวมถึงไฟล์ `app.py` ซึ่งเป็นไฟล์หลักในการรันโปรแกรม โดย Flask รองรับการแสดงผล HTML และการปรับแต่งหน้าเว็บด้วย CSS อย่างยืดหยุ่น ช่วยให้นักพัฒนาสามารถออกแบบเว็บไซต์ที่มีทั้งความสวยงามและการใช้งานที่มีประสิทธิภาพได้อย่างง่ายดาย ทำให้ Flask เป็นเฟรมเวิร์คที่เหมาะสำหรับผู้เริ่มต้นและผู้ที่ต้องการเรียนรู้การพัฒนาเว็บไซต์ด้วยภาษา Python ในเชิงลึกอย่างแท้จริง (*พัฒนาเว็บด้วยภาษา Python (Flask) สำหรับผู้เริ่มต้น ฉบับเต็มปี 2024*, n.d.)

* 1. **ไลบรารี Netmiko**

Netmiko เป็นไลบรารี Python ที่ถูกออกแบบมาเพื่อช่วยเหลือในงาน Network Automation หรือระบบเครือข่ายอัตโนมัติ โดยเน้นการลดความซับซ้อนในการจัดการเครือข่าย เช่น การเชื่อมต่ออุปกรณ์ การส่งคำสั่ง และการดึงข้อมูลจากอุปกรณ์เครือข่าย Netmiko ถูกพัฒนาบนพื้นฐานของไลบรารี Paramiko ซึ่งเป็นไลบรารีสำหรับการเชื่อมต่อแบบ SSH (Secure Shell) ที่ให้ความปลอดภัยสูง โดย Netmiko ได้เพิ่มฟังก์ชันที่ช่วยให้การเชื่อมต่อและการจัดการอุปกรณ์เครือข่ายสะดวกขึ้น เช่น การเข้าสู่โหมดการกำหนดค่า (Configuration Mode) การส่งคำสั่งที่มีความซับซ้อน และการจัดการคำสั่งที่ต้องรอผลลัพธ์ นอกจากนี้ Netmiko ยังรองรับการเชื่อมต่อกับอุปกรณ์หลากหลายแพลตฟอร์ม เช่น Cisco, Juniper, HP และอื่น ๆ โดยมีการทดสอบความสามารถกับแพลตฟอร์มที่หลากหลายอยู่เป็นประจำ จุดเด่นของ Netmiko คือช่วยลดเวลาที่ใช้ในการพัฒนาสคริปต์สำหรับการเชื่อมต่อและจัดการอุปกรณ์เครือข่ายผ่าน SSH ได้อย่างมีประสิทธิภาพ และยังช่วยลดความซับซ้อนในงานจัดการเครือข่ายที่เคยใช้เวลานาน เช่น การตั้งค่าอุปกรณ์หรือการตรวจสอบข้อมูล ด้วยคุณสมบัติที่ออกแบบมาเพื่อการใช้งานในระบบเครือข่ายอัตโนมัติ Netmiko จึงเป็นหนึ่งในไลบรารียอดนิยมสำหรับนักพัฒนาที่ต้องการสร้างระบบที่สามารถเชื่อมต่อและจัดการอุปกรณ์เครือข่ายได้อย่างมีประสิทธิภาพและปลอดภัยมากยิ่งขึ้น (MaCoM, 2020)

* 1. **EVE-NG – เครื่องมือจำลองอุปกรณ์เครือข่าย**

EVE-NG (Emulated Virtual Environment Next Generation) เป็น Emulator ที่ใช้สำหรับการจำลองอุปกรณ์เครือข่ายได้หลากหลายชนิด เช่น switch, router, firewall และอุปกรณ์อื่น ๆ ที่เกี่ยวข้องในระบบเครือข่าย EVE-NG ช่วยให้ผู้ใช้งานสามารถทดสอบและฝึกฝนการตั้งค่าอุปกรณ์เครือข่ายในสภาพแวดล้อมเสมือนจริง (Virtual Environment) โดยไม่จำเป็นต้องใช้ฮาร์ดแวร์จริง ซึ่งทำให้สะดวกและประหยัดค่าใช้จ่ายในการทดลองและเรียนรู้การทำงานของอุปกรณ์เครือข่ายต่าง ๆ ได้อย่างครบถ้วน อย่างไรก็ตาม การใช้งานอุปกรณ์ใน EVE-NG จำเป็นต้องดาวน์โหลด firmware ของอุปกรณ์ที่ต้องการจำลองแล้วนำมาบันทึกไว้ใน EVE-NG เพื่อให้สามารถใช้งานได้ ซึ่งการดาวน์โหลดและติดตั้ง firmware เหล่านี้เป็นขั้นตอนที่สำคัญในการใช้งาน EVE-NG ให้ได้ผลลัพธ์ที่ถูกต้องและมีประสิทธิภาพสูงสุด ในการจำลองการตั้งค่าระบบเครือข่ายต่าง ๆ นอกจากนี้ EVE-NG ยังรองรับการจำลองการเชื่อมต่อหลายอุปกรณ์พร้อมกันในเวลาเดียวกัน ช่วยให้ผู้ใช้งานสามารถสร้างโครงข่ายเครือข่ายที่มีความซับซ้อนได้โดยไม่จำเป็นต้องใช้ฮาร์ดแวร์จริง ทำให้เป็นเครื่องมือที่สำคัญสำหรับการฝึกอบรมและการทดสอบอุปกรณ์เครือข่ายในหลาย ๆ สถานการณ์ (*ทำความรู้จัก EVE Next Generation พร้อมวิธีการ Deploy – AbleNet*, n.d.)

* 1. **การใช้งาน Threading เพื่อการประมวลผลแบบพร้อมกันใน Python**

การใช้ Threading ใน Python เป็นเทคนิคที่ช่วยให้สามารถทำงานหลายกระบวนการพร้อมกันในเวลาเดียวกันได้ ซึ่งเหมาะสำหรับกรณีที่ต้องการประมวลผลที่ใช้เวลานาน หรือกรณีที่มีการทำงานที่เป็น I/O-bound เช่น การดึงข้อมูลจากฐานข้อมูลหรือการดาวน์โหลดไฟล์จากอินเทอร์เน็ต โดยการใช้งาน threading module ใน Python จะช่วยให้สามารถสร้างและควบคุมการทำงานของแต่ละ Thread ได้อย่างมีประสิทธิภาพ การทำงานของ Thread จะเป็นการแบ่งเวลาในการทำงานให้กับแต่ละส่วนเล็ก ๆ ที่เรียกว่า Thread ซึ่งแต่ละ Thread จะทำงานในเส้นทางของตัวเองและสามารถทำงานพร้อมกันได้โดยการแบ่งทรัพยากรร่วมกัน การใช้งาน Thread จะช่วยเพิ่มประสิทธิภาพในกรณีที่โปรแกรมต้องการทำงานหลาย ๆ อย่างในเวลาเดียวกัน เช่น การพิมพ์ตัวเลข 1-5 และตัวอักษร A-E ในเวลาเดียวกัน โดยการเรียกใช้ start() ของแต่ละ Thread ฟังก์ชันในแต่ละ Thread จะเริ่มทำงานพร้อมกัน เมื่อโปรแกรมเรียกใช้ join() ของแต่ละ Thread โปรแกรมจะรอให้ทั้งสอง Thread เสร็จสิ้นการทำงานก่อนที่จะพิมพ์ข้อความ "โปรแกรมจบการทำงาน" ซึ่งแสดงถึงการเสร็จสิ้นของการทำงานทั้งหมด การใช้งาน Threading จึงเป็นเครื่องมือที่สะดวกและมีประสิทธิภาพสำหรับการประมวลผลหลายๆ งานพร้อมกันใน Python โดยเฉพาะในกรณีที่ต้องจัดการกับงานที่ใช้เวลาในการประมวลผลนานหรือมีลักษณะ I/O-bound. (*การใช้งาน Threading เพื่อการประมวลผลแบบพร้อมกันใน Python*, 2023)

* 1. **NLP (Natural Language Processing)**

NLP (Natural Language Processing) หรือ การประมวลผลภาษาธรรมชาติ เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่มุ่งเน้นให้คอมพิวเตอร์สามารถเข้าใจและประมวลผลภาษาที่มนุษย์ใช้ในการสื่อสารประจำวัน เช่น การพูด การเขียน และแม้กระทั่งภาษามือ โดยกระบวนการ Processing หมายถึงการที่คอมพิวเตอร์ทำความเข้าใจภาษาของมนุษย์ ซึ่งเป็นสิ่งที่ท้าทายเนื่องจากภาษามีความหลากหลายและซับซ้อน การพัฒนา NLP จึงมีความสำคัญในการเชื่อมช่องว่างระหว่างมนุษย์และคอมพิวเตอร์ เพื่อทำให้การสื่อสารเป็นไปได้อย่างราบรื่นและมีประสิทธิภาพมากขึ้น ตัวอย่างการใช้งาน NLP ที่เราคุ้นเคยกันดีในชีวิตประจำวัน ได้แก่ การค้นหาคำใน Google ที่มักมีคำแนะนำหรือการเติมคำอัตโนมัติเมื่อพิมพ์ข้อความในอีเมล และยังรวมถึงเทคโนโลยีการสั่งงานด้วยเสียง เช่น Siri บน iPhone หรือ Google Maps ที่สามารถบอกทางได้ด้วยเสียง ซึ่งทั้งหมดนี้ล้วนใช้เทคโนโลยี NLP ในการประมวลผลและเข้าใจคำพูดของผู้ใช้

ในขณะที่คอมพิวเตอร์มักทำงานบนพื้นฐานของ Logic หรือเหตุและผล การใช้ NLP ช่วยให้คอมพิวเตอร์สามารถเข้าใจภาษามนุษย์ที่มีความหลากหลายและซับซ้อนมากขึ้น ซึ่งการทำให้คอมพิวเตอร์สามารถเข้าใจภาษามนุษย์ได้ จะช่วยอำนวยความสะดวกในด้านต่าง ๆ เช่น การทำงานของอุปกรณ์ที่สามารถรับคำสั่งจากผู้ใช้และตอบสนองได้อย่างถูกต้อง *(“มารู้จักกับ ‘NLP’ ที่ไม่ใช่ Neuro Linguistic Programming แต่คือ Natural Language Processing…,” 2021)*

* 1. **ฐานข้อมูล MongoDB**

MongoDB เป็นระบบจัดการฐานข้อมูลโอเพ่นซอร์สแบบ ข้ามแพลตฟอร์ม ซึ่งไม่ใช้ SQL และได้รับการออกแบบมาเพื่อจัดเก็บและจัดการข้อมูลจำนวนมากได้อย่างยืดหยุ่นและสามารถปรับขนาดได้ตามต้องการ โดย MongoDB ใช้ โมเดลข้อมูลเชิงเอกสาร (Document-based Model) ซึ่งเก็บข้อมูลในรูปแบบเอกสารที่คล้ายกับ JSON พร้อมสคีมาเสริม (Flexible Schema) ที่ช่วยให้การจัดการข้อมูลมีความยืดหยุ่นมากขึ้น และสามารถพัฒนาได้เร็วขึ้น เนื่องจากสคีมาสามารถปรับเปลี่ยนได้ง่ายโดยไม่จำเป็นต้องย้ายข้อมูลที่มีค่า ซึ่งเป็นข้อได้เปรียบที่สำคัญเมื่อเทียบกับฐานข้อมูลที่ใช้ SQL ซึ่งต้องมีการย้ายข้อมูลทุกครั้งที่มีการเปลี่ยนแปลงโครงสร้างฐานข้อมูล MongoDB ใช้ สถาปัตยกรรมแบบกระจาย (Distributed Architecture) ที่สามารถแบ่งข้อมูลเป็น พาร์ติชัน (Shards) ข้ามเซิร์ฟเวอร์หลายตัว เพื่อรองรับการขยายขนาดในแนวนอนได้ (Horizontal Scaling) ซึ่งหมายความว่า MongoDB สามารถจัดการข้อมูลจำนวนมากและรองรับการอ่าน-เขียนข้อมูลในเวิร์กโหลดที่สูงได้ โดยไม่จำเป็นต้องอัปเกรดฮาร์ดแวร์ราคาแพง นอกจากนี้ยังมีฟีเจอร์การ เฟลโอเวอร์อัตโนมัติ และ ชุดเรพลิกา (Replica Sets) เพื่อความพร้อมใช้งานสูง (High Availability) โดยช่วยให้มั่นใจได้ว่าแม้ในกรณีที่เกิดความล้มเหลวของฮาร์ดแวร์หรือการหยุดชะงักอื่นๆ ข้อมูลในฐานข้อมูลยังคงสามารถเข้าถึงและใช้งานได้ (*MongoDB คืออะไร | AppMaster*, n.d.)

เปรียบเทียบข้อมูลในรูปแบบต่างๆ เช่น MongoDB, JSON

**MongoDB (ฐานข้อมูล NoSQL)**

|  |
| --- |
| {  "\_id": "12345",  "name": "ข้าวโอ๊ต",  "age": 20,  "address": {  "street": "123 ซอยท่าเรือ",  "city": "กรุงเทพฯ",  "zipcode": "10110"  },  "courses": [  { "course\_id": "ENCOR", "course\_name": "CCNP ENCOR" },  { "course\_id": "NLP001", "course\_name": "NLP" }  ]  } |

**JSON (JavaScript Object Notation)**

|  |
| --- |
| {  "id": "12345",  "name": "ข้าวโอ๊ต",  "age": 20,  "address": {  "street": "123 ซอยท่าเรือ",  "city": "กรุงเทพฯ",  "zipcode": "10110"  },  "courses": [  { "course\_id": "ENCOR", "course\_name": "CCNP ENCOR" },  { "course\_id": "NLP001", "course\_name": "NLP" }  ]  } |

* MongoDB เป็นฐานข้อมูล ซึ่งสามารถเก็บข้อมูลในรูปแบบ JSON-like ได้ แต่ไม่ได้เก็บเป็นไฟล์ JSON แต่เป็นฐานข้อมูลที่สามารถจัดการข้อมูลขนาดใหญ่ได้
* MongoDB สามารถเก็บข้อมูลที่มีโครงสร้างที่แตกต่างกันในแต่ละ document ได้ (เช่น ฟิลด์บางตัวอาจมีค่าบางตัวที่ไม่มีใน document อื่นๆ)
  1. ไลบรารี Net-SNMP

Simple Network Management Protocol (SNMP) เป็นโปรโตคอลที่ใช้กันอย่างแพร่หลายสำหรับการตรวจสอบสุขภาพและสภาพการทำงานของอุปกรณ์เครือข่าย เช่น เราเตอร์, อุปกรณ์คอมพิวเตอร์, และอุปกรณ์อื่นๆ เช่น UPS (Uninterruptible Power Supply) โดยใช้โปรโตคอล SNMP เพื่อเข้าถึงข้อมูลการจัดการและสถานะของอุปกรณ์ต่างๆ Net-SNMP เป็นชุดเครื่องมือที่ใช้ในการทำงานกับ SNMP ซึ่งรองรับ SNMP v1, SNMP v2c และ SNMP v3 ทั้งใน IPv4 และ IPv6 โดยชุดเครื่องมือนี้ประกอบด้วยแอปพลิเคชันหลายตัวที่ช่วยในการตรวจสอบและจัดการอุปกรณ์ที่รองรับ SNMP เช่น การดึงข้อมูลจากอุปกรณ์ การตั้งค่าคอนฟิก การจัดการกับข้อมูล MIB (Management Information Base), และการส่งการแจ้งเตือน SNMP Net-SNMP รองรับหลายระบบปฏิบัติการ รวมถึง Unix, Unix-like และ Windows โดยมีฟังก์ชันที่สามารถปรับแต่งได้ตามระบบปฏิบัติการที่ใช้ (*Net-SNMP*, n.d.)

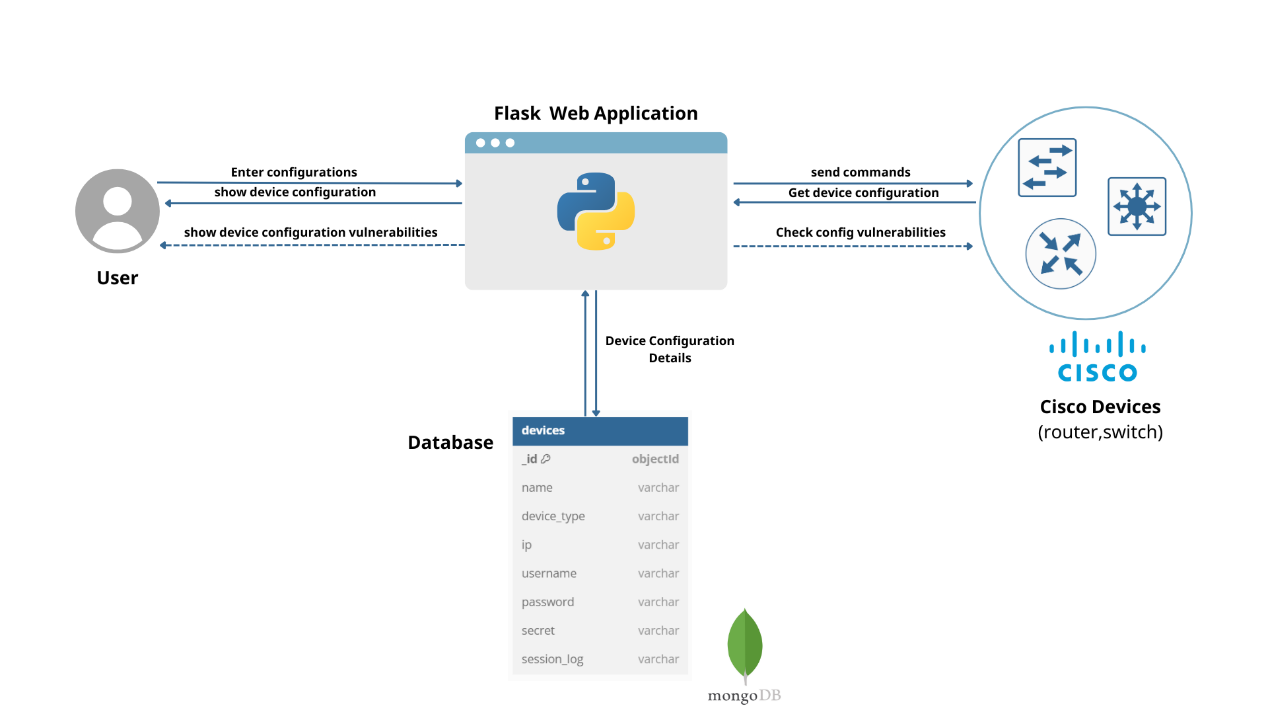
**บทที่ 3**

**วิธีการดำเนินงาน**

* 1. **ขั้นตอนการศึกษา**

การพัฒนาเว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO จำเป็นต้องศึกษาขั้นตอนและกระบวนการที่เป็นระบบ เพื่อให้สามารถนำไปใช้งานจริงได้อย่างมีประสิทธิภาพ ขั้นตอนในการศึกษามีดังนี้

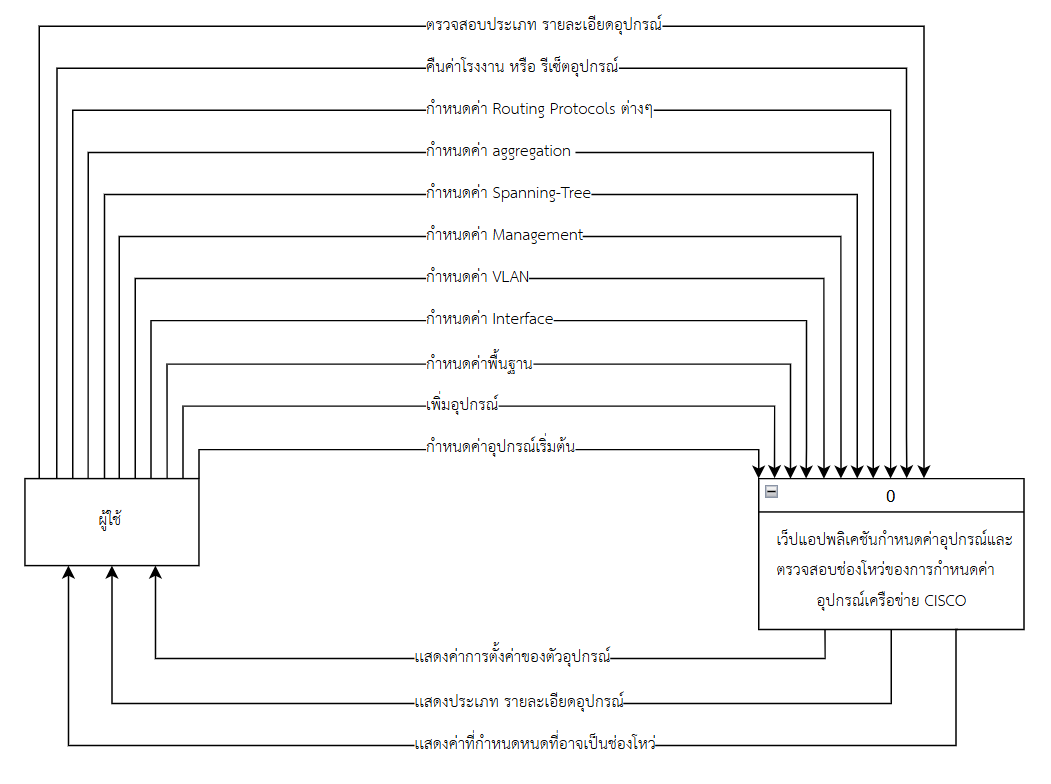
* + 1. ศึกษาวิธีการเขียนโปรแกรมด้วยภาษา Python เพื่อเชื่อมต่อและเข้าถึงอุปกรณ์เครือข่าย
    2. ศึกษาการตั้งค่าอุปกรณ์ประเภท Switch (Layer 2, Layer 3) และ Router ในระบบเครือข่าย
    3. ศึกษาการพัฒนาเว็บแอปพลิเคชันด้วย Flask สำหรับสร้างส่วนติดต่อผู้ใช้ (UI) เพื่อให้สามารถใช้งานระบบผ่านเว็บ
    4. ศึกษาการใช้ MongoDB ในการจัดเก็บรายละเอียดของอุปกรณ์และการวิเคราะห์ความปลอดภัย
    5. ศึกษาการใช้ Netmiko สำหรับการเชื่อมต่อและตั้งค่าอุปกรณ์เครือข่าย เช่น Switch และ Router รวมถึงการดำเนินการคำสั่งต่างๆ บนอุปกรณ์เพื่อปรับแต่งการตั้งค่า
    6. ศึกษาโพรโทคอล SNMP เพื่อนำมาต่อยอดในโครงงาน
    7. ศึกษาการใช้งาน Net-SNMP เพื่อช่วยในการตรวจสอบรายละเอียดของอุปกรณ์
    8. ศึกษาการใช้ Machine Learning ในการตรวจสอบความปลอดภัยของการตั้งค่าอุปกรณ์เครือข่าย เช่น การตรวจสอบความแข็งแกร่งของรหัสผ่านและการวิเคราะห์ความเสี่ยงในการกำหนดค่าของอุปกรณ์เครือข่าย
    9. ศึกษาการใช้ Threading และ Scheduler ในการจัดการ Task ที่ต้องทำงานพร้อมกันหรือทำงานตามกำหนดเวลา เช่น การใช้ ThreadPoolExecutor และ BackgroundScheduler เพื่อประมวลผลการตั้งค่าอุปกรณ์เครือข่ายแบบขนาน
    10. ศึกษาการใช้งาน API เพื่อเชื่อมต่อระบบต่าง ๆ และส่งข้อมูลระหว่าง Front-end และ Back-end ในรูปแบบ JSON
    11. ศึกษาการใช้ Subprocess สำหรับการเรียกใช้งานคำสั่งจากระบบปฏิบัติการและการจัดการข้อผิดพลาดในกระบวนการประมวลผล
    12. ศึกษาวิธีการจัดการการเข้าถึงและการกำหนดค่าอุปกรณ์ผ่าน Serial Ports โดยใช้ไลบรารี serial ของ python
  1. **Overview Diagram**

****

**ภาพที่ 3-1** แสดงการทำงานและลำดับขั้นตอนโดยรวมของโปรแกรม

จากภาพที่ 3-1 จะแสดงภาพรวมของการทำงานของโปรแกรม ทำการเชื่อมต่อกับอุปกรณ์เครือข่ายที่ยังไม่ได้กำหนดค่า โดยผ่านสาย Serial เมื่อทำการเชื่อมต่อผ่านสาย Serial แล้ว ผู้ใช้ทำการกำหนดค่าต่างๆของอุปกรณ์ สำหรับการกำหนดค่าเริ่มต้น เพื่อให้พร้อมใช้งานกับเว็บแอปพลิเคชัน และอุปกรณ์นั้นจะถูกบันทึกลงฐานข้อมูล หรือในกรณีที่อุปกรณ์มีที่อยู่ IP Address แล้ว และรองรับการกำหนดค่าผ่าน SSH แล้ว ก็สามารถเพิ่มรายละเอียดเช่น IP Address, Username, Password เพื่อให้รองรับการกำหนดค่าผ่านเว็บแอปพลิเคชัน โดยที่ไม่ต้องใช้ฟังก์ชันการกำหนดค่าเริ่มต้นอุปกรณ์

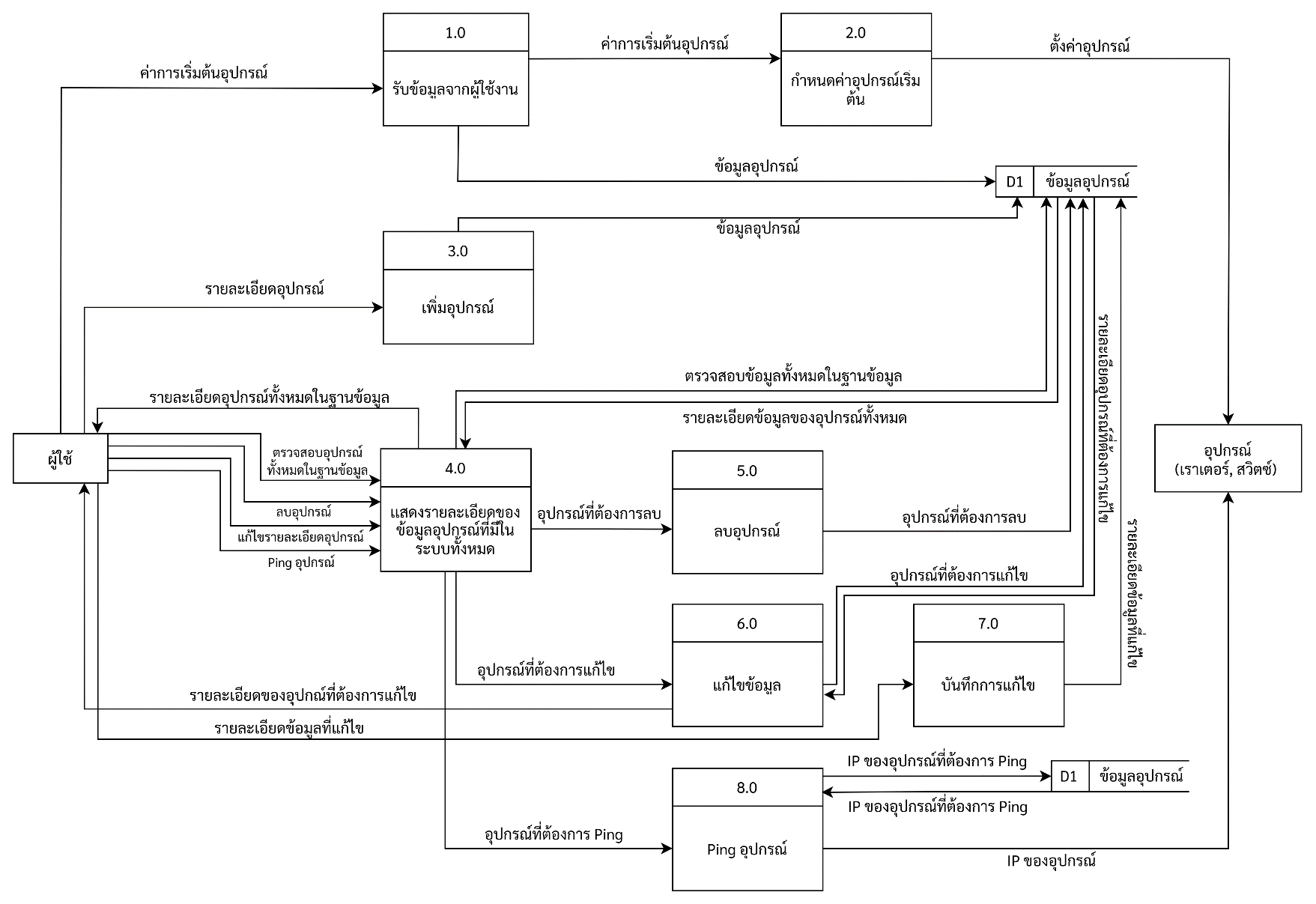
* 1. **Context Diagram**



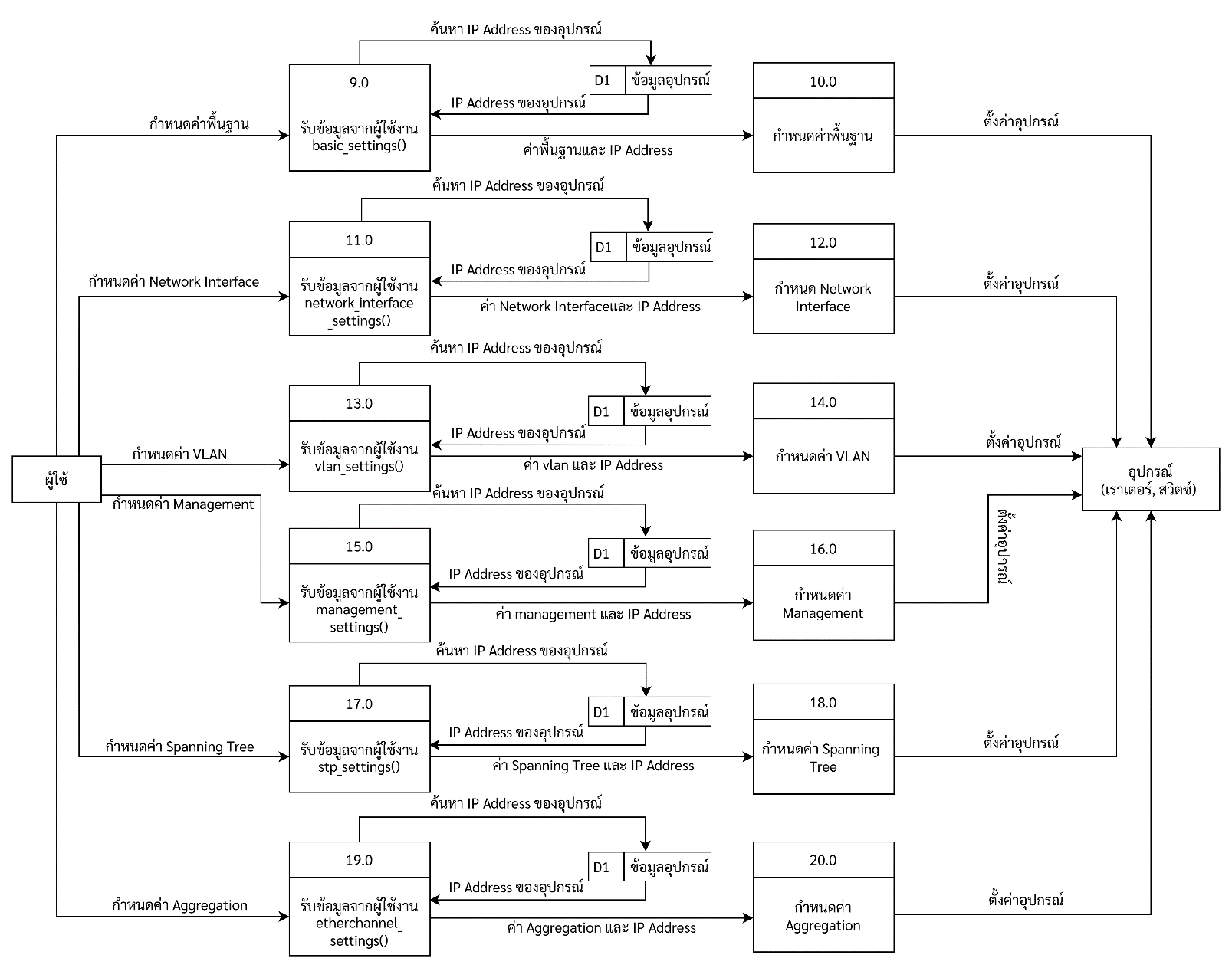
**ภาพที่ 3-2** Context Diagram เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO

จากภาพที่ 3-2 เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO เป็นระบบที่ช่วยให้ผู้ใช้งานสามารถกำหนดค่าเกี่ยวกับอุปกรณ์เครือข่าย CISCO ได้ครอบคลุม โดยเริ่มจากการตั้งค่าการใช้งานเริ่มต้น การตรวจสอบประเภทและรายละเอียดของอุปกรณ์ การคืนค่าโรงงาน หรือรีเซ็ตอุปกรณ์ การกำหนดค่าฟังก์ชันสำคัญต่าง ๆ เช่น Routing Protocols, Aggregation, Spanning-Tree, Management, VLAN, Interface และค่าพื้นฐานเช่น Banner, Hostname ตลอดจนการเพิ่มอุปกรณ์ใหม่ ระบบยังสามารถตรวจสอบช่องโหว่ของการกำหนดค่าเพื่อเพิ่มความปลอดภัยของเครือข่าย โดยผู้ใช้งานสามารถโต้ตอบกับระบบผ่านเว็บไซต์หรือแอปพลิเคชันเพื่อรับข้อมูลผลลัพธ์ เช่น การแสดงค่าที่ถูกกำหนดไว้ (Configured) ของอุปกรณ์ รายละเอียดต่างๆของอุปกรณ์ และข้อมูลการตั้งค่าที่อาจเป็นช่องโหว่ ทั้งนี้ ระบบถูกออกแบบมาเพื่อสนับสนุนการจัดการและเสริมสร้างความปลอดภัยให้กับเครือข่าย CISCO และเพิ่มความสะดวกสบายแก่ผู้ใช้งาน

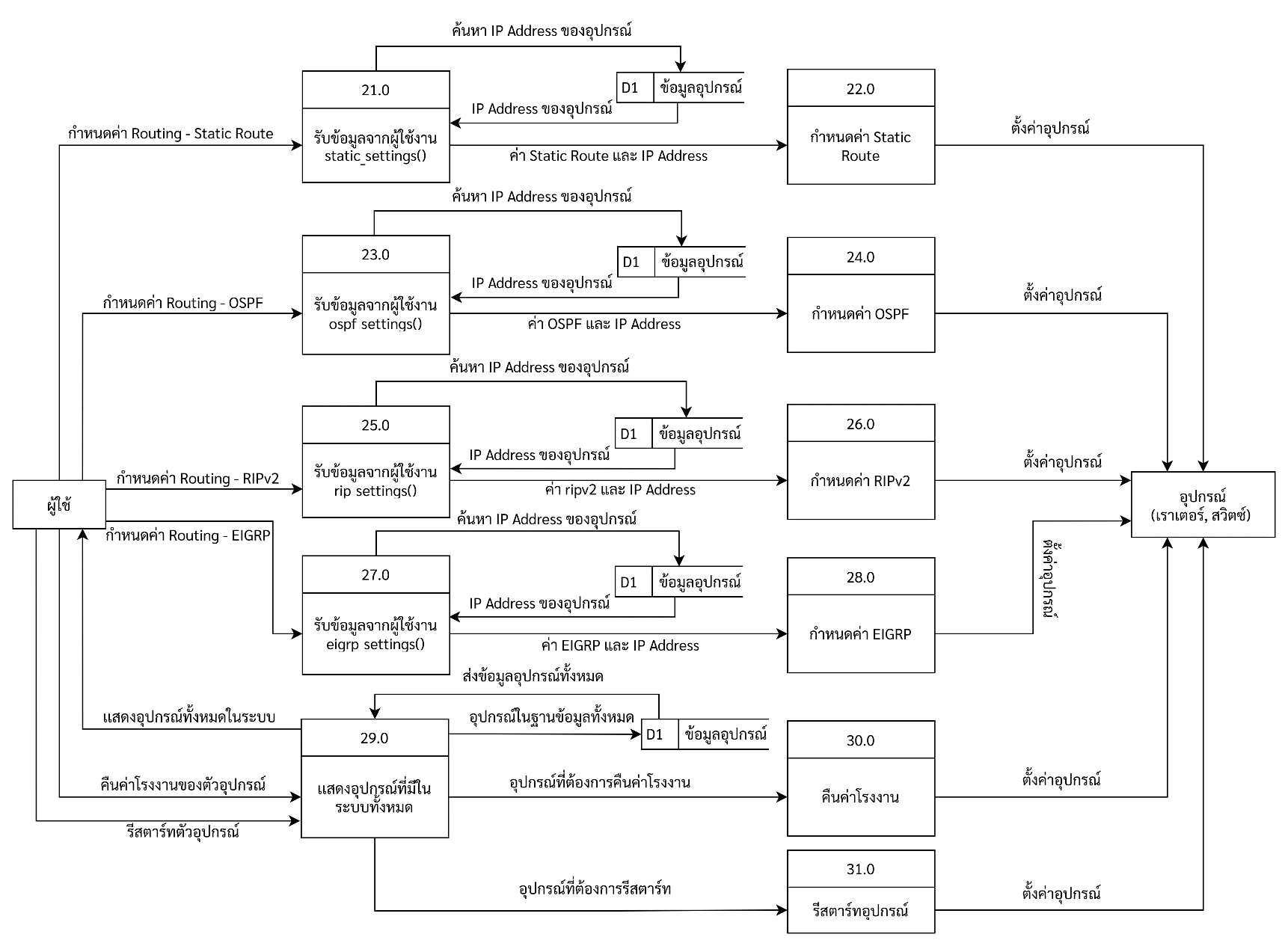
* 1. **Data Flow Diagram (DFD)**

****

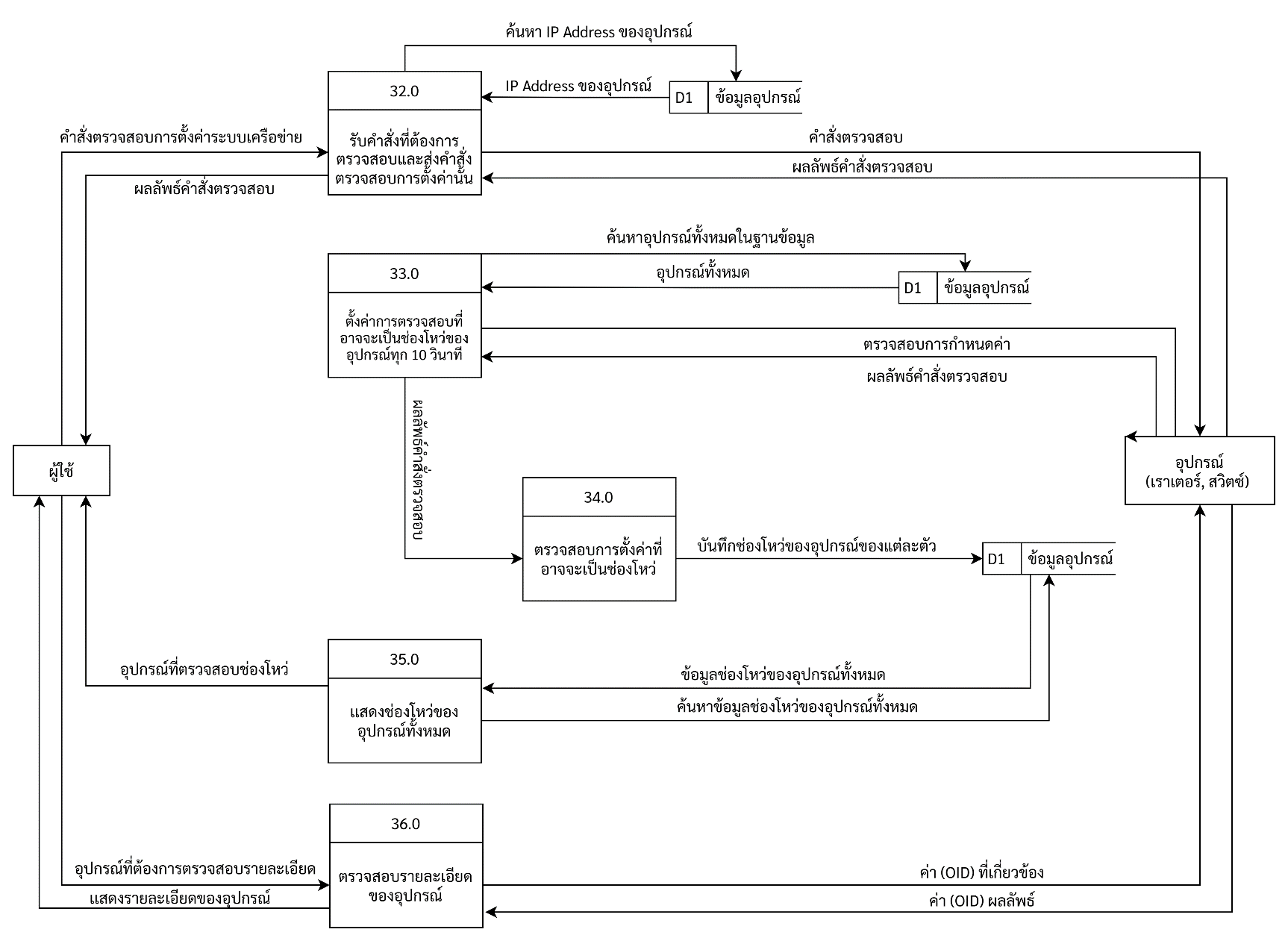
**ภาพที่ 3-3** Data Flow Diagram Level 0 เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO

****

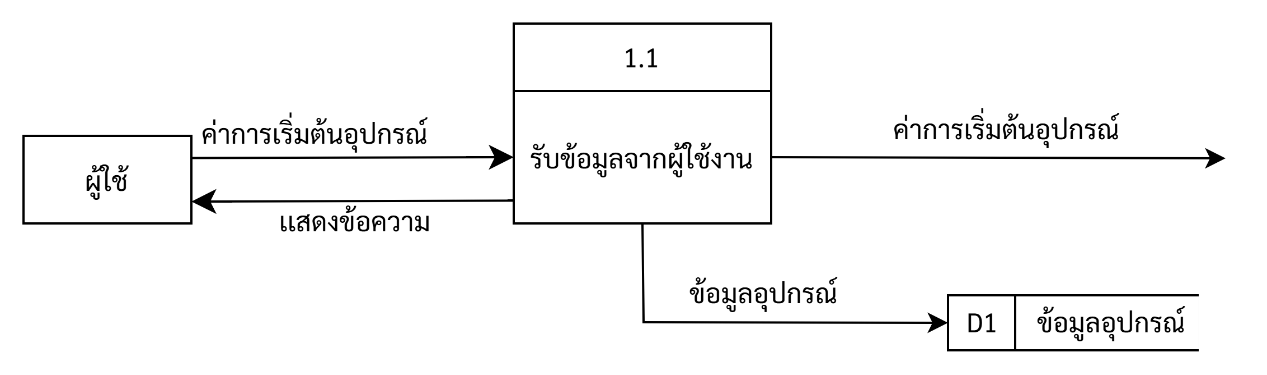
**ภาพที่ 3-4** Data Flow Diagram Level 0 เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO (ต่อ)

****

**ภาพที่ 3-5** Data Flow Diagram Level 0 เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO (ต่อ)

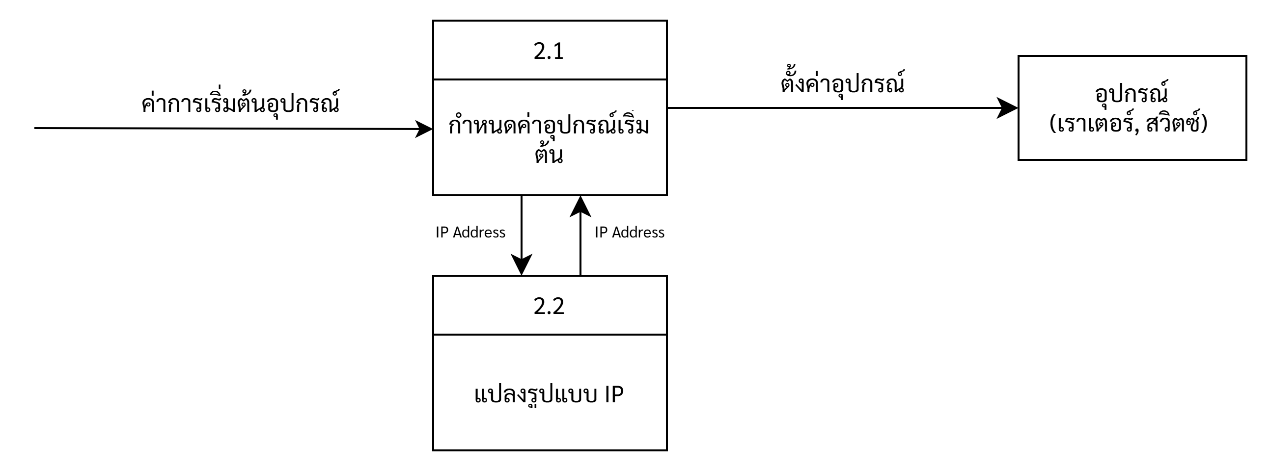
****

**ภาพที่ 3-6** Data Flow Diagram Level 0 เว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO (ต่อ)

****

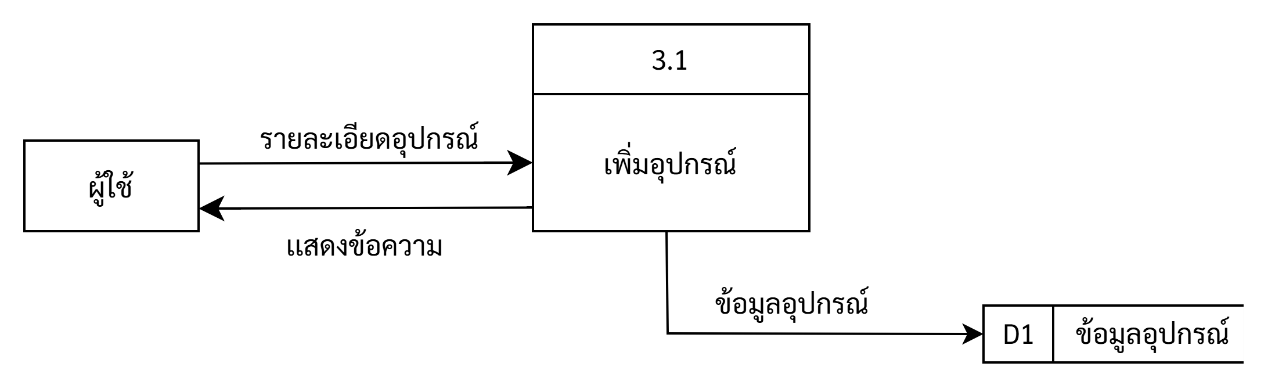
**ภาพที่ 3-7** Data Flow Diagram Level 1 การรับค่าเริ่มต้นของอุปกรณ์

จากภาพ 3-7DFD Level 1 แสดงถึงการทำงานของโมดูลย่อย 1.1 ที่เน้นการรับค่าการตั้งต้น DFD Level 1 แสดงถึงการทำงานของโมดูลย่อย 1.1 โดยมีหน้าที่รับค่าข้อมูลเริ่มต้นจากผู้ใช้ผ่านอินเทอร์เฟซ เช่น พอร์ต Console, ชื่อโฮสต์, โดเมน, รหัสผ่าน, ข้อมูล SSH, ชนิดการตั้งค่าที่อยู่ IP และที่อยู่ IP พร้อมตรวจสอบความถูกต้องและความซ้ำซ้อนของข้อมูลในฐานข้อมูล D1 ก่อนบันทึกข้อมูลใหม่ และส่งต่อค่าที่ได้รับไปยัง Process 2 (serial\_script) เพื่อดำเนินการตั้งค่าการเชื่อมต่ออุปกรณ์ โดยผลลัพธ์จะแสดงข้อความยืนยันความสำเร็จหรือแจ้งข้อผิดพลาดกลับไปยังผู้ใช้ผ่านหน้าเว็บ



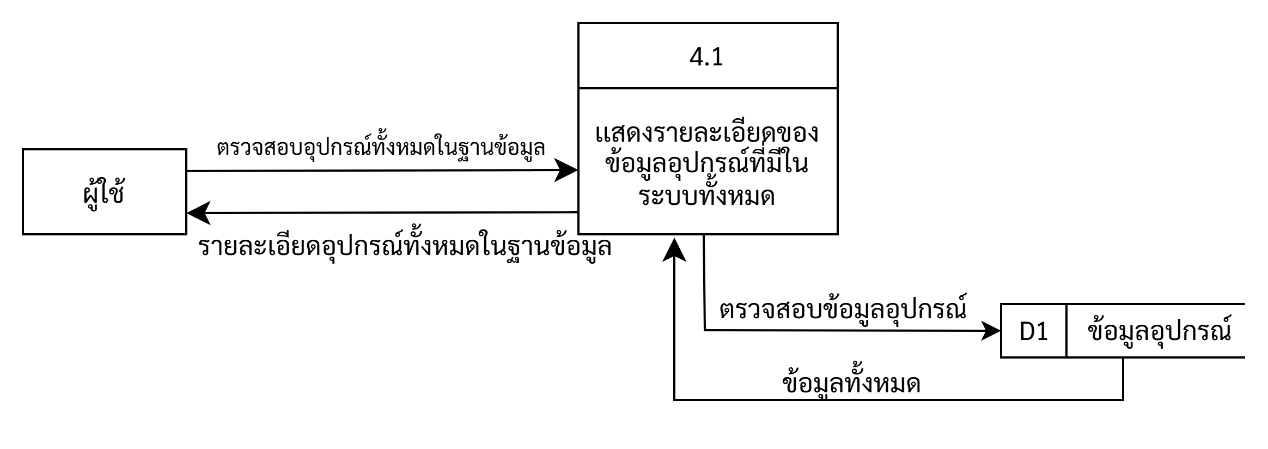
**ภาพที่ 3-8** Data Flow Diagram Level 1 การส่งการตั้งค่าเริ่มต้นไปยังอุปกรณ์

จากภาพ 3-8 DFD Level 1 แสดงถึงการทำงานของโมดูลย่อย 2.1 ซึ่งเกี่ยวข้องกับกระบวนการส่งคำสั่งไปยังอุปกรณ์ผ่านการเชื่อมต่อพอร์ตอนุกรม (Serial Port) โดยมีการเปิดพอร์ต การแปลงรูปแบบ CIDR เป็น Subnet Mask โดยโมดูลย่อย 2.2 การรับ-ส่งคำสั่ง และการตรวจสอบผลลัพธ์ พร้อมดำเนินการตั้งค่าระบบเครือข่าย เช่น การตั้งค่า Hostname, Domain Name, Password, การกำหนดค่า IP แบบ DHCP หรือ Static, การเปิดใช้งานอินเทอร์เฟซ, การกำหนดค่า SSH และการบันทึกค่าการตั้งต้นของระบบ ทั้งหมดนี้ดำเนินการผ่านฟังก์ชันในโมดูล serial\_script.commands ซึ่งส่งคำสั่งไปยังอุปกรณ์และอ่านค่าผลลัพธ์เพื่อตรวจสอบสถานะความสำเร็จในการดำเนินการ



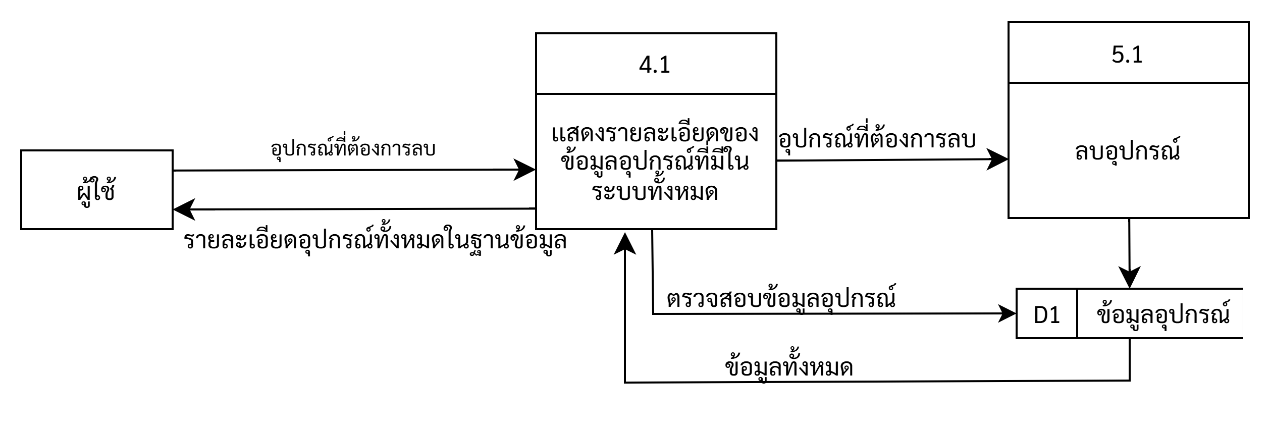
**ภาพที่ 3-9** Data Flow Diagram Level 1 การจัดการข้อมูลบันทึกของอุปกรณ์

จากภาพ 3-9 DFD Level 1 แสดงถึงกระบวนการจัดการข้อมูลบันทึกของอุปกรณ์ (Device Record Management) ซึ่งเป็นส่วนหนึ่งของ โมดูล 3.1 โดยโมดูลนี้มีหน้าที่หลักในการรับข้อมูลของอุปกรณ์จากผู้ใช้ผ่านแบบฟอร์ม เช่น ชื่ออุปกรณ์ (Hostname), ที่อยู่ IP, รหัสผ่านระดับ Privilege, ชื่อผู้ใช้ และรหัสผ่าน SSH ก่อนที่จะดำเนินการตรวจสอบความซ้ำซ้อนของข้อมูลในฐานข้อมูล D1 ทั้งในส่วนของชื่ออุปกรณ์และที่อยู่ IP เพื่อหลีกเลี่ยงความขัดแย้ง หากไม่มีข้อมูลซ้ำ ระบบจะเพิ่มข้อมูลใหม่ลงในฐานข้อมูล D1 และแสดงข้อความยืนยันความสำเร็จ หรือแจ้งเตือนหากเกิดข้อผิดพลาด พร้อมนำผู้ใช้กลับไปยังหน้าจัดการข้อมูลอีกครั้ง



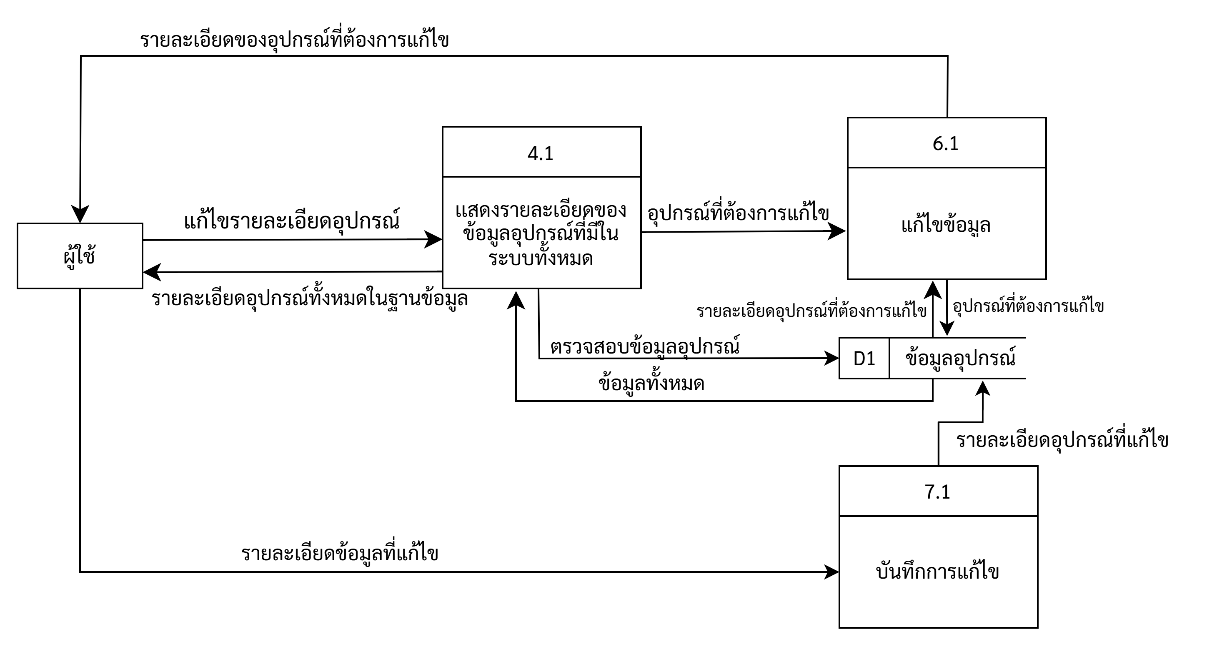
**ภาพที่ 3-10** Data Flow Diagram Level 1 การแสดงรายละเอียดของอุปกรณ์

จากภาพ 3-10 DFD Level 1 แสดงถึงกระบวนการจัดการข้อมูลและการแสดงผลข้อมูลอุปกรณ์ (Devices Information) ซึ่งเป็นส่วนหนึ่งของ โมดูล 4.1 โดยโมดูลนี้ทำหน้าที่ดึงข้อมูลทั้งหมดของอุปกรณ์ที่บันทึกไว้ในฐานข้อมูล D1 เช่น ชื่ออุปกรณ์ (Hostname), ที่อยู่ IP, และรายละเอียดการตั้งค่าต่างๆ เพื่อรวบรวมเป็นรายการและส่งต่อให้แสดงผลในหน้าเว็บ Devices Information หากเกิดข้อผิดพลาดในการเชื่อมต่อกับฐานข้อมูล ระบบจะส่งค่าข้อมูลว่างและแสดงหน้าเว็บพร้อมแจ้งสถานะความผิดพลาดในการแสดงผลข้อมูล



**ภาพที่ 3-11** Data Flow Diagram Level 1 การลบอุปกรณ์

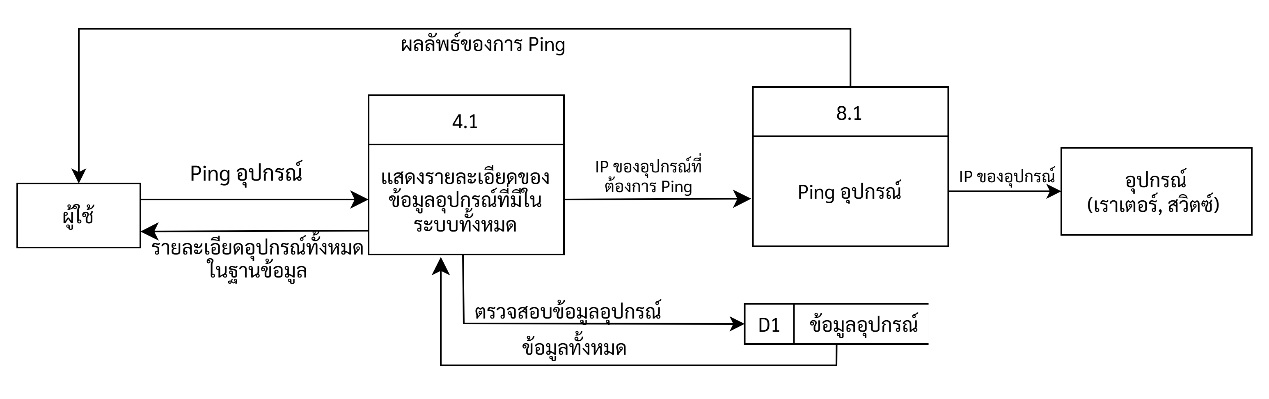
จากภาพ 3-11 DFD Level 1 แสดงถึงกระบวนการลบข้อมูลอุปกรณ์ (Delete Device) ซึ่งเป็นส่วนหนึ่งของ โมดูล 5.1 โดยโมดูลนี้ทำงานร่วมกับหน้าเว็บใน โมดูล 4.1 ที่แสดงรายการอุปกรณ์ทั้งหมดให้ผู้ใช้สามารถลบอุปกรณ์ที่ต้องการได้ ผู้ใช้สามารถกดปุ่มลบ (Delete) บนอุปกรณ์ที่เลือก โดยระบบจะรับค่าที่อยู่ IP ของอุปกรณ์จากแบบฟอร์ม Frontend และดำเนินการลบข้อมูลออกจากฐานข้อมูล D1 หลังจากการลบสำเร็จ ระบบจะนำผู้ใช้กลับไปยังหน้าแสดงรายการอุปกรณ์ พร้อมปรับปรุงรายการให้แสดงเฉพาะข้อมูลที่เหลืออยู่



**ภาพที่ 3-12** Data Flow Diagram Level 1 การแก้ไขอุปกรณ์

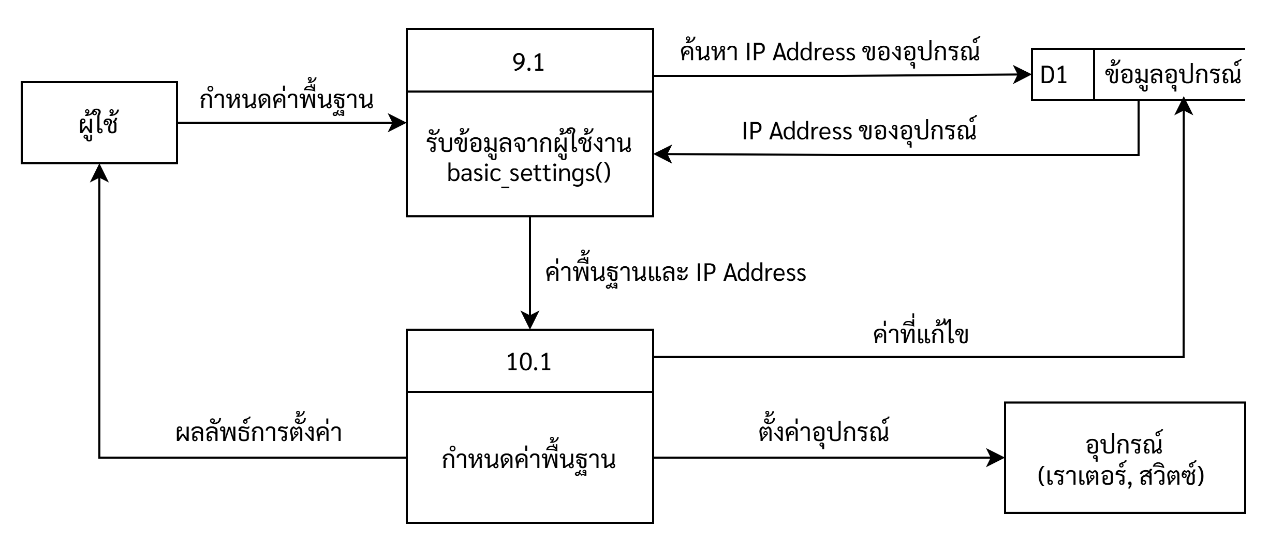
จากภาพ 3-12 DFD Level 1 แสดงถึงกระบวนการแก้ไขและอัปเดตข้อมูลอุปกรณ์ (Edit and Update Device) ซึ่งเกี่ยวข้องกับ โมดูล 6.1 และ 7.1 โดยกระบวนการเริ่มต้นจาก โมดูล 4.1 ที่แสดงรายการอุปกรณ์ทั้งหมดให้ผู้ใช้สามารถเลือกอุปกรณ์ที่ต้องการแก้ไขได้ เมื่อผู้ใช้กดปุ่มแก้ไข (Edit) ระบบจะส่งค่าที่อยู่ IP ของอุปกรณ์ผ่าน Route /edit/<ip\_address> เพื่อดึงข้อมูลของอุปกรณ์จากฐานข้อมูล D1 และแสดงในแบบฟอร์มแก้ไขข้อมูลบนหน้าเว็บ edit\_device.html ซึ่งเป็นส่วนของ โมดูล 6.1 ผู้ใช้สามารถปรับปรุงข้อมูล เช่น ชื่ออุปกรณ์, IP Address, SSH Username, SSH Password และ Secret Password ได้ตามต้องการ หลังจากแก้ไขข้อมูลเสร็จและกดบันทึก ระบบจะส่งข้อมูลที่ได้รับผ่าน Route /update ซึ่งเป็นส่วนของ โมดูล 7.1 เพื่อทำการตรวจสอบความถูกต้องและความซ้ำซ้อนของข้อมูล เช่น ชื่ออุปกรณ์หรือ IP Address หากไม่มีปัญหา ระบบจะดำเนินการอัปเดตข้อมูลในฐานข้อมูล D1 หรือเพิ่มข้อมูลใหม่ในกรณีที่ IP Address มีการเปลี่ยนแปลง

หลังจากอัปเดตข้อมูลสำเร็จ ระบบจะนำผู้ใช้กลับไปยังหน้าแสดงรายการอุปกรณ์ใน โมดูล 4.1 พร้อมแสดงข้อมูลที่อัปเดตแล้ว เพื่อให้มั่นใจว่ารายการแสดงข้อมูลที่ถูกต้องและเป็นปัจจุบัน



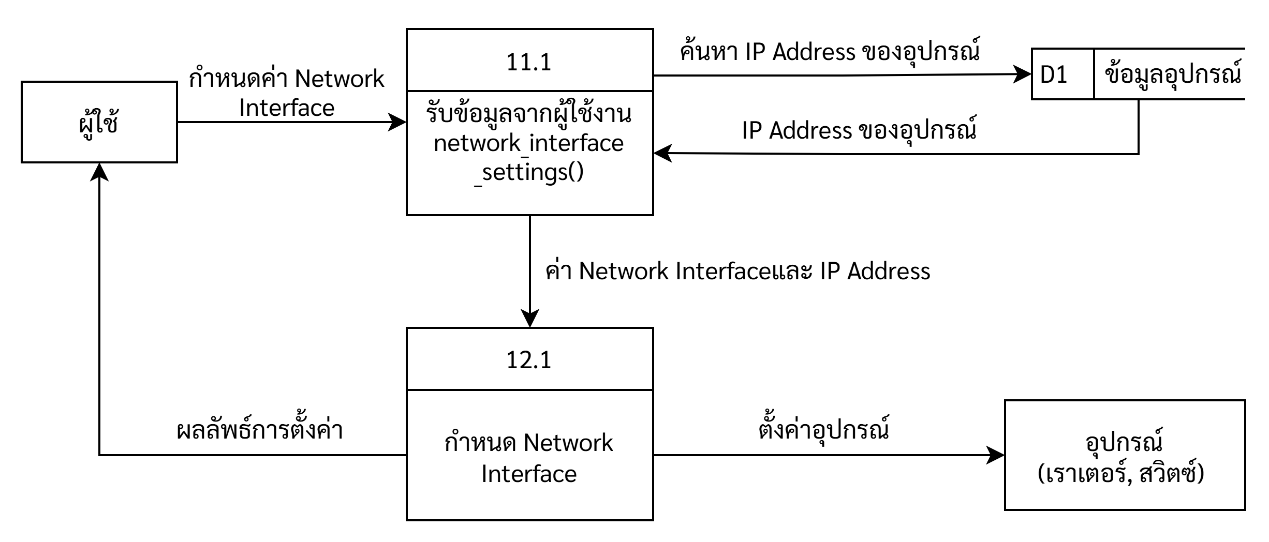
**ภาพที่ 3-13** Data Flow Diagram Level 1 การ Ping อุปกรณ์

จากภาพ 3-13 DFD Level 1 แสดงถึงกระบวนการทดสอบการเชื่อมต่ออุปกรณ์ (Ping Device) ซึ่งเป็นส่วนหนึ่งของกระบวนการในโมดูล 8.1 โดยระบบจะได้รับค่าที่อยู่ IP Address ของอุปกรณ์โดยตรงจากโมดูล 4.1 ซึ่งเป็นหน้าจอที่แสดงรายการอุปกรณ์ทั้งหมดให้ผู้ใช้เลือกดำเนินการ และเมื่อผู้ใช้กดปุ่ม Ping บนอุปกรณ์ที่ต้องการ ระบบจะรับค่าที่อยู่ IP Address จากรายการที่แสดงในหน้าจอนั้นโดยตรง โดยไม่จำเป็นต้องดึงข้อมูลจากฐานข้อมูลอีกครั้ง จากนั้นระบบจะส่งคำขอแบบ POST พร้อมกับ IP Address ไปยัง Backend เพื่อดำเนินการ Ping อุปกรณ์ปลายทาง เมื่อ Backend ได้รับคำขอ จะใช้คำสั่ง Ping เพื่อตรวจสอบสถานะการเชื่อมต่อกับอุปกรณ์ และส่งผลลัพธ์กลับไปยัง Frontend เพื่อแสดงข้อความที่บ่งบอกถึงความสำเร็จหรือความล้มเหลวในการ Ping อุปกรณ์ให้ผู้ใช้ทราบ ทั้งนี้การออกแบบให้ระบบรับค่าที่อยู่ IP Address จากโมดูล 4.1 โดยตรงช่วยลดความซับซ้อนของกระบวนการและเพิ่มความรวดเร็วในการทำงาน โดยเฉพาะในกรณีที่ผู้ใช้ต้องการดำเนินการตรวจสอบสถานะของอุปกรณ์หลายครั้งในเวลาอันสั้น



**ภาพที่ 3-14** Data Flow Diagram Level 1 การกำหนดค่าพื้นฐาน

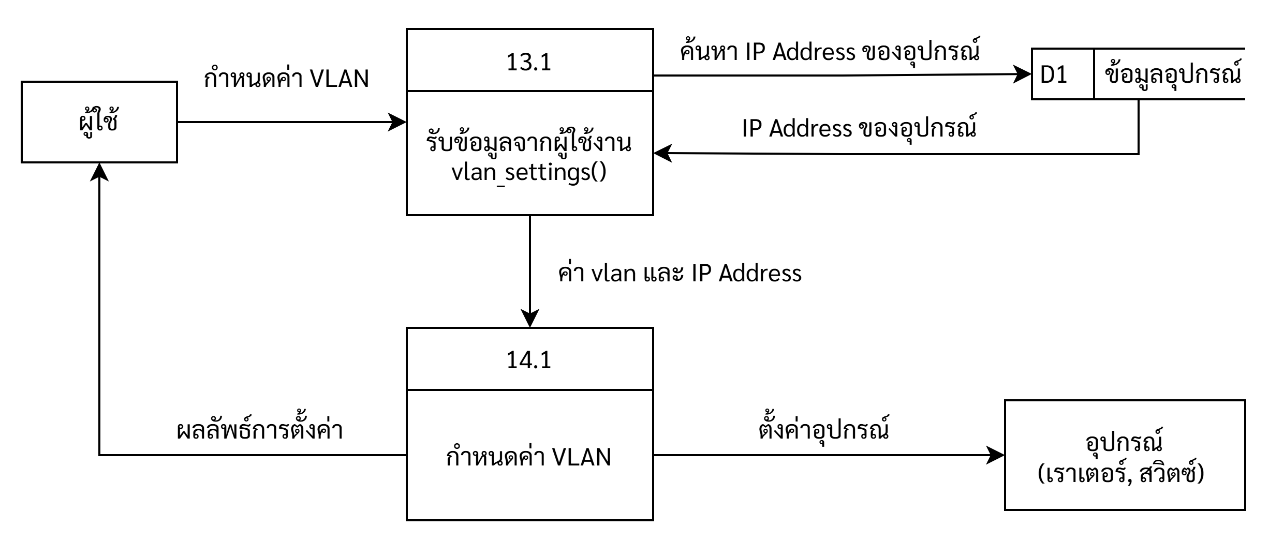
จากภาพ 3-14 DFD Level 1 แสดงถึงกระบวนการตั้งค่าพื้นฐานของอุปกรณ์เครือข่าย ซึ่งเป็นการทำงานของกระบวนการ 9.1 ที่รับข้อมูลจากหน้าเว็บของโมดูล Basic Settings โดยหน้าเว็บดังกล่าวจะแสดงรายการอุปกรณ์จากฐานข้อมูล D1 เพื่อให้ผู้ใช้เลือกอุปกรณ์หรือกรอกค่าการตั้งค่า เช่น ชื่อโฮสต์ (Hostname) รหัสผ่านลับ (Secret Password) ข้อความต้อนรับ (Banner) และการเปิดหรือปิดบริการเข้ารหัสรหัสผ่าน (Password Encryption) หลังจากที่ผู้ใช้ส่งข้อมูลผ่านฟอร์ม กระบวนการ 9.1 จะตรวจสอบค่าที่ป้อนว่าไม่มีความซ้ำซ้อนในระบบ และส่งคำสั่งไปยังกระบวนการ 10.1 ซึ่งเชื่อมต่อกับอุปกรณ์ผ่าน Netmiko เพื่อดำเนินการตั้งค่าตามคำสั่งที่ได้รับ เช่น การเปลี่ยนชื่อโฮสต์ การตั้งค่ารหัสผ่านลับ การเพิ่มบัญชีผู้ใช้งาน และการกำหนดข้อความ Banner เมื่อดำเนินการเสร็จสิ้น ระบบจะบันทึกข้อมูลที่อัปเดตกลับไปยังฐานข้อมูล D1 และส่งผลลัพธ์กลับไปยังหน้าเว็บเพื่อแจ้งให้ผู้ใช้ทราบสถานะการดำเนินการ



**ภาพที่ 3-15** Data Flow Diagram Level 1 การกำหนดค่า Network Interface

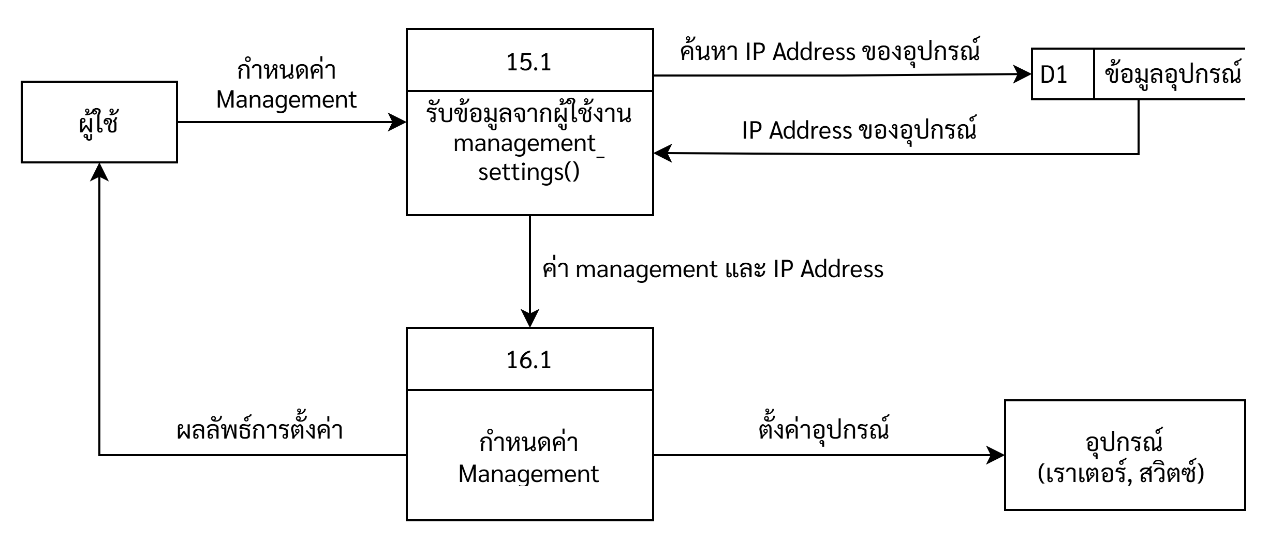
จากภาพ 3-15 DFD Level 1 แสดงถึงกระบวนการตั้งค่าการเชื่อมต่อเครือข่าย (Network Interface Settings) ซึ่งเป็นการทำงานของกระบวนการ 11.1 ที่รับข้อมูลจากหน้าเว็บของโมดูลการตั้งค่าพื้นฐานของอุปกรณ์เครือข่าย กระบวนการ 11.1 จะรับค่าการตั้งค่าจากฟอร์มที่ผู้ใช้กรอก เช่น ชื่ออุปกรณ์ (Device Name), การตั้งค่า IPv4, IPv6, การตั้งค่า DHCP, ที่อยู่ IP, มาสก์เครือข่าย (Subnet Mask), การตั้งค่าความเร็ว Duplex และพอร์ตต่างๆ

หลังจากนั้น กระบวนการ 11.1 จะทำการตรวจสอบข้อมูล เช่น การตรวจสอบการซ้ำซ้อนของ IP และส่งข้อมูลไปยังกระบวนการ 12.1 ซึ่งจะใช้ Netmiko ในการเชื่อมต่อกับอุปกรณ์และดำเนินการตั้งค่าตามคำสั่งที่ได้รับ เช่น การตั้งค่า IPv4, IPv6, การตั้งค่า DHCP, การเปิดหรือปิดพอร์ต, และการตั้งค่าความเร็ว Duplex ของพอร์ตที่เลือก โดยหลังจากกระบวนการ 12.1 ดำเนินการเสร็จสิ้น ระบบจะส่งผลลัพธ์กลับไปยังหน้าเว็บเพื่อให้ผู้ใช้ทราบสถานะของการดำเนินการ



**ภาพที่ 3-16** Data Flow Diagram Level 1 การกำหนดค่า VLAN

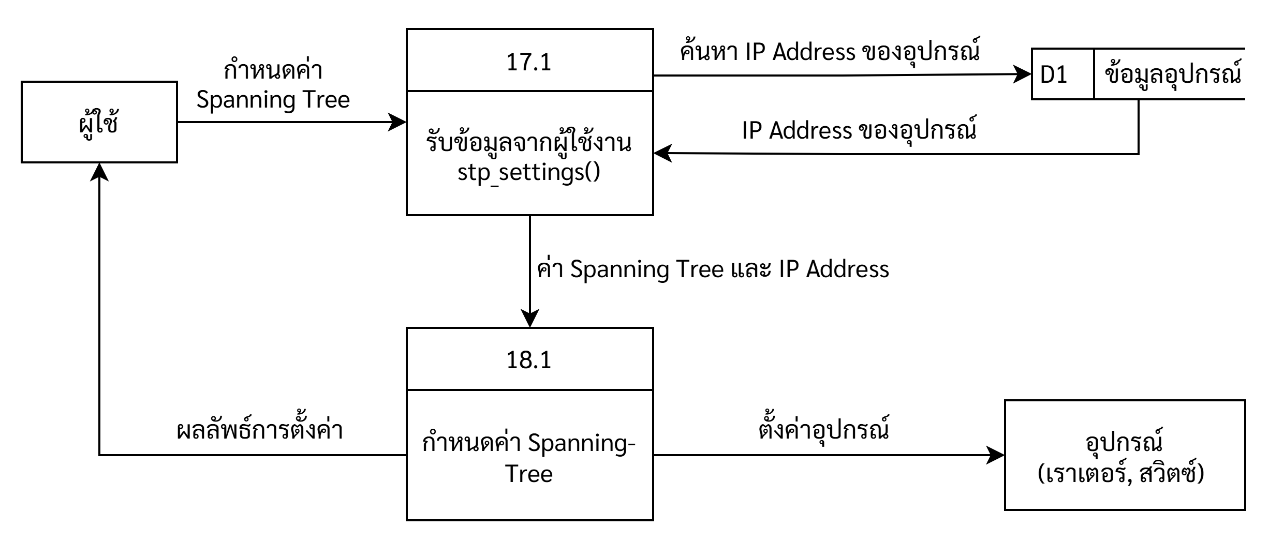
จากภาพ 3-16 DFD Level 1 แสดงถึงกระบวนการตั้งค่าการจัดการ VLAN (VLAN Management Settings) ซึ่งเป็นการทำงานของกระบวนการ 13.1 ที่รับข้อมูลจากหน้าเว็บของโมดูลการตั้งค่า VLAN ของอุปกรณ์เครือข่าย กระบวนการ 13.1 จะรับค่าต่างๆ จากฟอร์มที่ผู้ใช้กรอกผ่านหน้าเว็บ เช่น ชื่ออุปกรณ์ (Device Name), VLAN ID ที่ต้องการเพิ่มหรือลบ, การตั้งค่า VLAN ที่มีอยู่, การเปิดหรือปิด VLAN, การตั้งค่า Access VLAN, Trunk Ports, การตั้งค่า DTP (Dynamic Trunking Protocol), และการตั้งค่า VLAN สำหรับ Trunk Ports เมื่อได้รับข้อมูลแล้ว กระบวนการ 13.1 จะทำการตรวจสอบข้อมูล เช่น การตรวจสอบว่า IP ของอุปกรณ์ไม่ซ้ำซ้อนในฐานข้อมูล จากนั้นกระบวนการจะส่งคำสั่งไปยังกระบวนการ 14.1 ซึ่งเป็นการส่งคำสั่งไปยังอุปกรณ์จริงโดยใช้ Netmiko ในการเชื่อมต่อและดำเนินการตั้งค่าตามคำสั่งที่ได้รับ เช่น การสร้าง VLAN, การลบ VLAN, การเปลี่ยนชื่อ VLAN, การเปิดหรือปิด VLAN, การตั้งค่า Access VLAN, การตั้งค่า Trunk Ports, การตั้งค่า DTP, และการลบไฟล์ vlan.dat จากนั้น ระบบจะให้ผลลัพธ์กลับไปยังหน้าเว็บเพื่อแจ้งสถานะการตั้งค่าที่เสร็จสมบูรณ์



**ภาพที่ 3-17** Data Flow Diagram Level 1 การกำหนดค่า Management

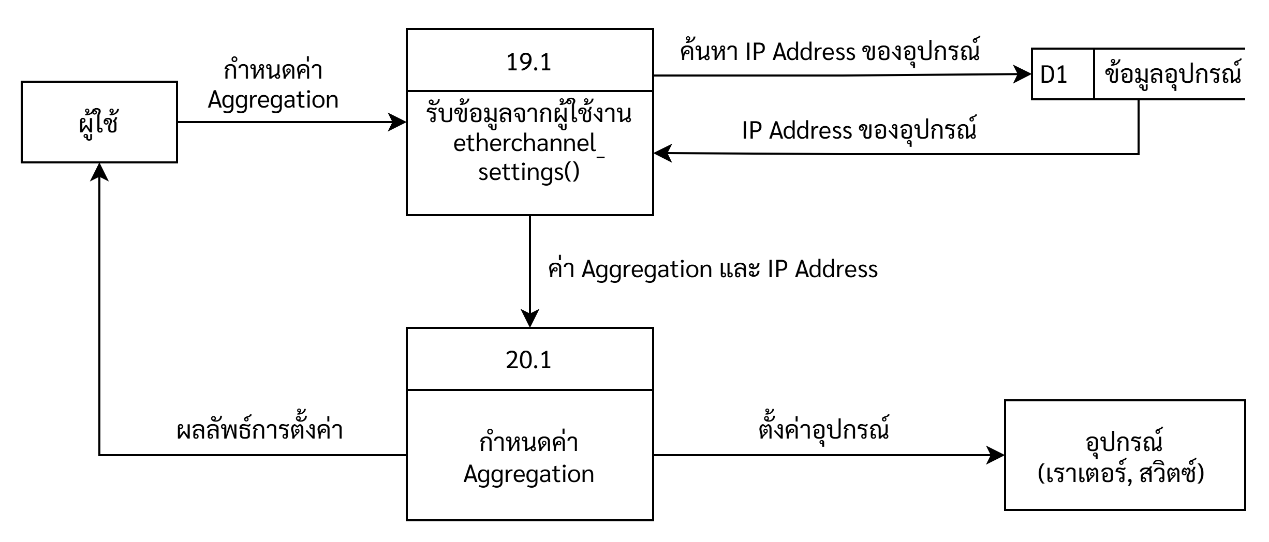
จากภาพ 3-17 และกระบวนการ 15.1 จะรับค่าจากฟอร์มที่ผู้ใช้กรอกผ่านหน้าเว็บ เช่น ชื่ออุปกรณ์ (Device Name), การตั้งค่า VTY และ Console (เช่น รหัสผ่าน, เวลา Timeout, การตั้งค่า DHCP, SNMP, NTP, Time Zone, และโปรโตคอล CDP/LLDP). เมื่อได้รับข้อมูลจากผู้ใช้แล้ว ระบบจะทำการตรวจสอบข้อมูลอุปกรณ์ เช่น การตรวจสอบ IP ของอุปกรณ์ในฐานข้อมูล จากนั้นกระบวนการ 15.1 จะดำเนินการเชื่อมต่อไปยังอุปกรณ์ที่เลือกโดยใช้ Netmiko เพื่อดำเนินการตั้งค่าตามคำสั่งที่ได้รับจากฟอร์ม เช่น การตั้งค่ารหัสผ่าน VTY และ Console, การตั้งค่า DHCP, การกำหนดค่าต่างๆ เช่น NTP, Time Zone, SNMP, และการตั้งค่าโปรโตคอล CDP/LLDP

กระบวนการ 16.1 คือการกำหนดค่าอุปกรณ์จริง โดยใช้คำสั่งผ่าน Netmiko เพื่อส่งคำสั่งไปยังอุปกรณ์ เช่น การตั้งค่ารหัสผ่าน VTY, การตั้งค่า DHCP, การตั้งค่า SNMP, การตั้งค่า NTP, Time Zone, CDP และ LLDP. การตั้งค่าจะถูกดำเนินการแบบขนานสำหรับอุปกรณ์หลายๆ ตัว โดยใช้เธรด (Thread). เมื่อการตั้งค่าทุกอย่างเสร็จสมบูรณ์ ผลลัพธ์จะถูกส่งกลับไปยังหน้าเว็บเพื่อแจ้งสถานะการตั้งค่าที่เสร็จสมบูรณ์ให้ผู้ใช้ทราบ



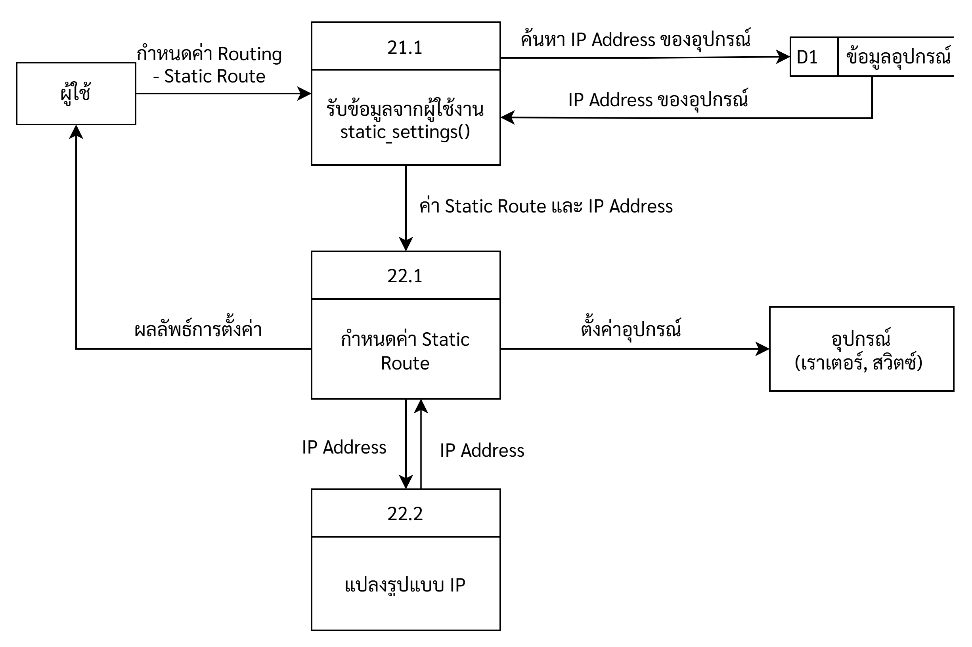
**ภาพที่ 3-18** Data Flow Diagram Level 1 การกำหนดค่า Spanning-Tree

จากภาพ 3-18 และกระบวนการ 17.1 คือการรับค่าจากฟอร์มที่ผู้ใช้กรอกในหน้าเว็บ เช่น การตั้งค่า STP mode, Root Primary, Root VLAN, Root Secondary, การเปิดหรือปิด PortFast, การตั้งค่า PortFast สำหรับแต่ละอินเตอร์เฟซ จากนั้นกระบวนการจะทำการตรวจสอบข้อมูลอุปกรณ์ที่ต้องการจากฐานข้อมูล เช่น การตรวจสอบชื่ออุปกรณ์หรือ IP อุปกรณ์ และหากพบอุปกรณ์ที่ตรงกัน กระบวนการจะเริ่มต้นการตั้งค่าผ่านเธรดหลายตัวเพื่อส่งคำสั่งไปยังอุปกรณ์หลายตัวพร้อมกัน, กระบวนการ 18.1 คือการส่งคำสั่ง STP ไปยังอุปกรณ์ที่เชื่อมต่ออยู่จริงโดยใช้ Netmiko ในการเชื่อมต่อและดำเนินการตามการตั้งค่าที่ผู้ใช้กรอก เช่น การตั้งค่า spanning-tree mode, การตั้งค่า Root Primary/Secondary, การเปิดหรือปิด PortFast รวมถึงการตั้งค่า BPDU Guard ผ่านคำสั่ง CLI ตามที่ได้รับจากฟอร์ม และหลังจากดำเนินการเสร็จสิ้นระบบจะให้ผลลัพธ์กลับไปยังหน้าเว็บเพื่อแสดงสถานะของการตั้งค่าที่เสร็จสมบูรณ์



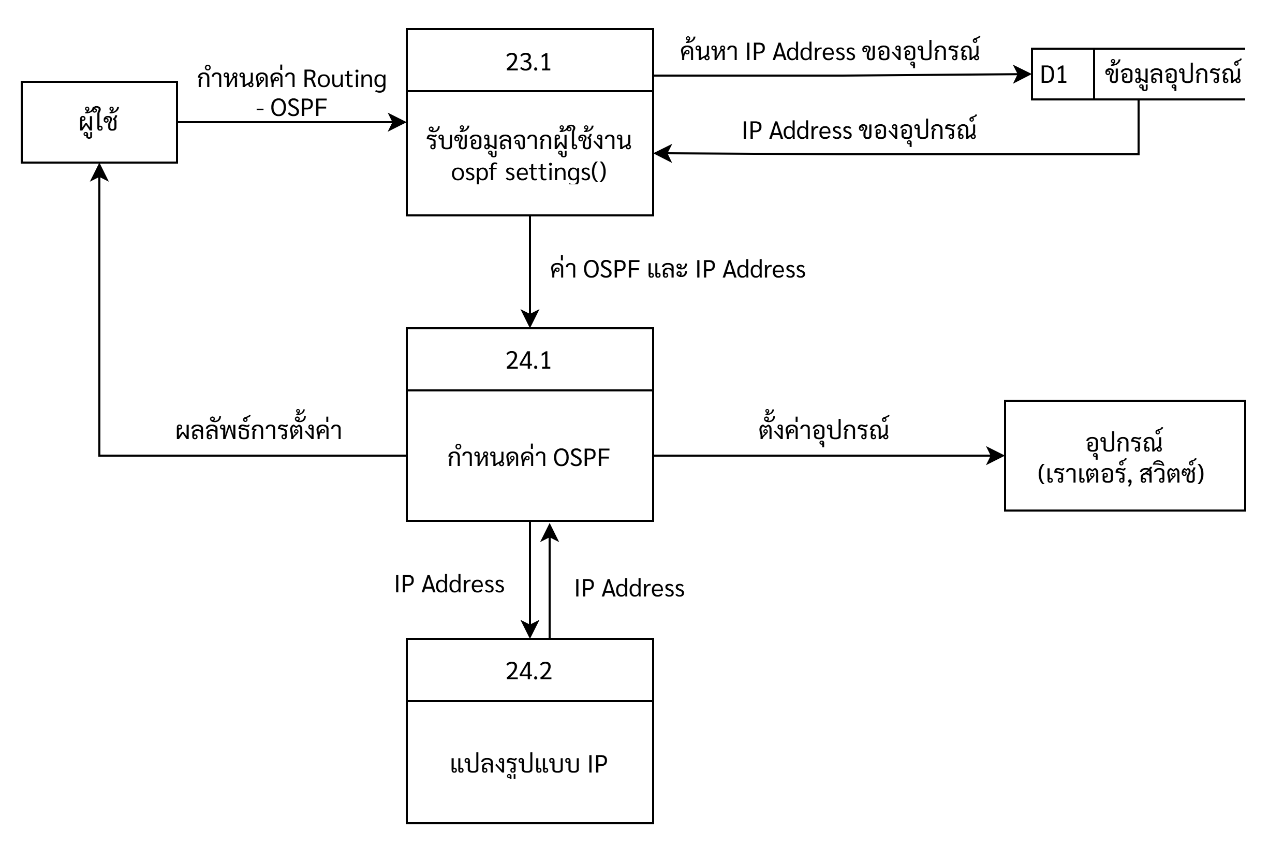
**ภาพที่ 3-19** Data Flow Diagram Level 1 การกำหนดค่า Aggregation

จากภาพ 3-19 และกระบวนการ 19.1 คือการรับค่าจากฟอร์มที่ผู้ใช้กรอกในหน้าเว็บ เช่น การตั้งค่าการใช้งาน EtherChannel ผ่าน PAgP หรือ LACP โดยการเลือกพอร์ตที่ต้องการใช้งาน (EtherChannel Interfaces), หมายเลขกลุ่มช่องสัญญาณ (Channel Group Number), โหมด PAgP (Desirable, Auto), โหมด LACP (Active, Passive) และการลบการตั้งค่ากลุ่มพอร์ตที่เลือก จากนั้นกระบวนการจะตรวจสอบข้อมูลอุปกรณ์ที่ต้องการจากฐานข้อมูล เช่น การตรวจสอบชื่ออุปกรณ์หรือ IP อุปกรณ์ กระบวนการ 20.1 คือการส่งคำสั่ง EtherChannel ไปยังอุปกรณ์จริงโดยใช้ Netmiko ในการเชื่อมต่อและดำเนินการตามการตั้งค่าที่ผู้ใช้กรอก เช่น การตั้งค่าการใช้งาน PAgP หรือ LACP รวมถึงการลบพอร์ตกลุ่มที่เลือก และหลังจากดำเนินการเสร็จสิ้นระบบจะให้ผลลัพธ์กลับไปยังหน้าเว็บเพื่อแสดงสถานะของการตั้งค่า



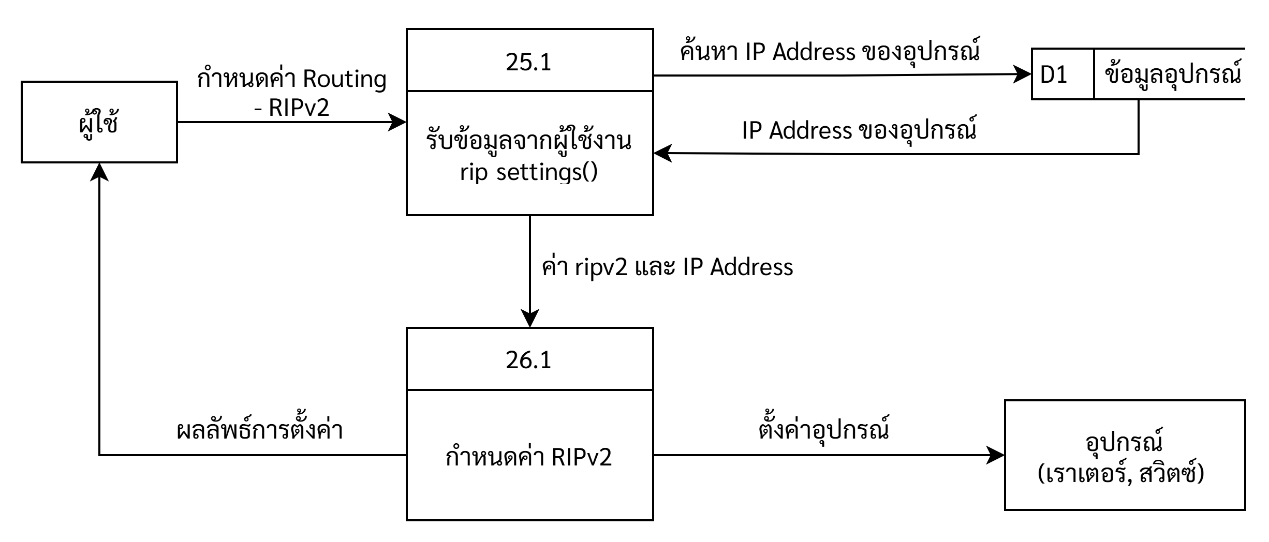
**ภาพที่ 3-20** Data Flow Diagram Level 1 การกำหนดค่า Static Route

จากรูป 3-20 และกระบวนการ 21.1 คือการรับค่าจากฟอร์มที่ผู้ใช้กรอกในหน้าเว็บ เช่น การตั้งค่าเส้นทางแบบคงที่ (Static Route) ที่ประกอบไปด้วยค่าต่างๆ เช่น เส้นทางปลายทาง (Destination Networks), Next Hop หรือ Exit Interfaces, การตั้งค่าเส้นทางเริ่มต้น (Default Route), และตัวเลือกในการลบเส้นทางที่มีอยู่ โดยกระบวนการนี้จะตรวจสอบข้อมูลที่ได้รับจากฟอร์มและหากข้อมูลถูกต้อง จะเริ่มต้นกระบวนการ 22.1 ซึ่งคือการส่งคำสั่งการตั้งค่าไปยังอุปกรณ์ที่เลือกผ่านการเชื่อมต่อ CLI โดยใช้ Netmiko ในกระบวนการ 22.1 จะมีการแปลง IP ที่ได้รับจากฟอร์มเป็น Subnet Mask โดยการใช้ CIDR (Classless Inter-Domain Routing) ซึ่งจะคำนวณเป็น Subnet Mask ที่เหมาะสมก่อน ที่ Process 22.2 จากนั้นจะนำคำสั่งการตั้งค่าเส้นทางคงที่ (Static Route) ที่ได้มา เช่น ip route หรือ no ip route พร้อมกับ Subnet Mask ไปส่งให้กับอุปกรณ์ที่ต้องการเพื่อดำเนินการตั้งค่าต่อไป หลังจากคำสั่งการตั้งค่าเส้นทางสำเร็จ ระบบจะทำการแจ้งเตือนผลลัพธ์กลับไปยังผู้ใช้ผ่านหน้าเว็บ เพื่อให้ผู้ใช้สามารถตรวจสอบสถานะของการตั้งค่าเส้นทางที่ได้ดำเนินการเสร็จสมบูรณ์แล้ว



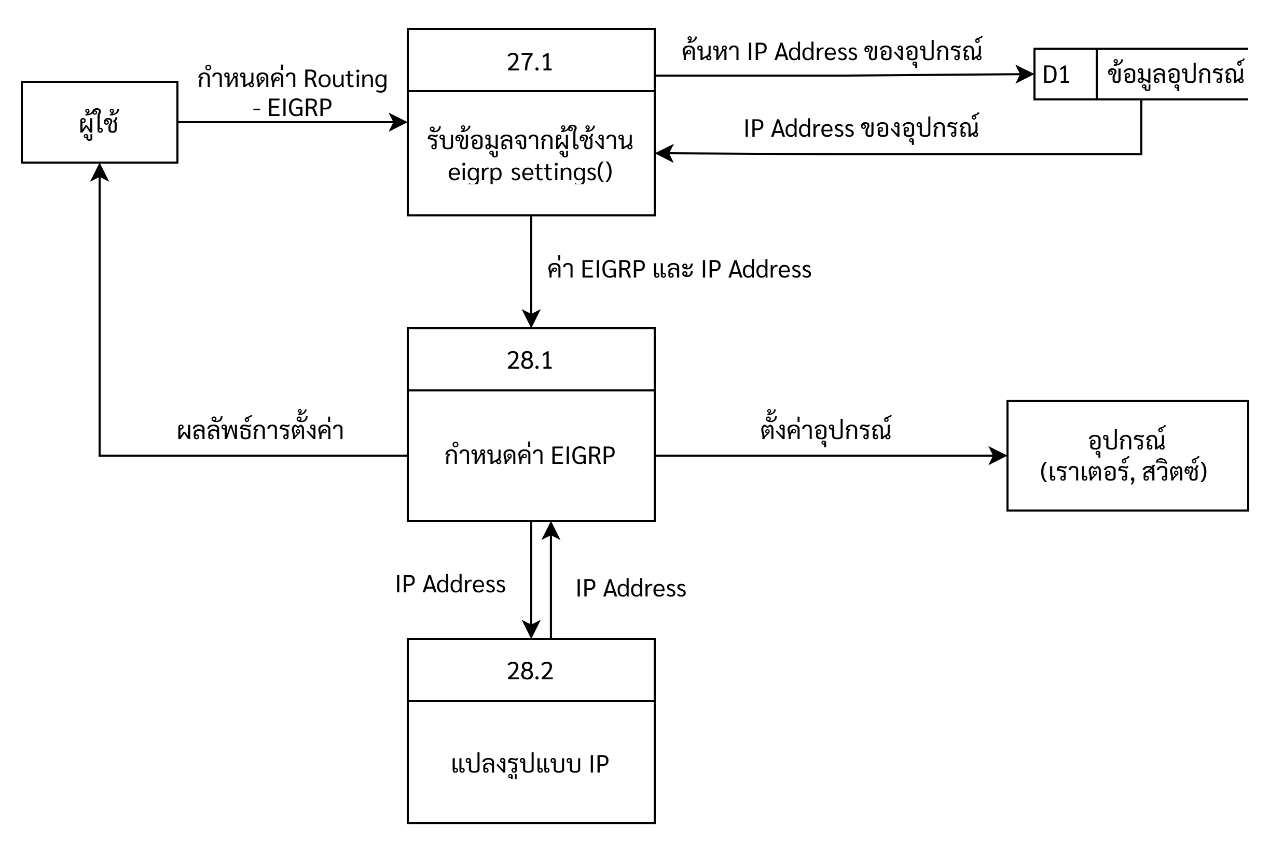
**ภาพที่ 3-21** Data Flow Diagram Level 1 การกำหนดค่า OSPF

จากภาพ 3-21 DFD Level 1 แสดงถึงการทำงานของกระบวนการตั้งค่า OSPF Routing โดยเริ่มต้นที่กระบวนการ 23.1 ซึ่งเป็นการรับค่าผ่านฟอร์มจากหน้าเว็บ เช่น Process ID, Router ID, เครือข่ายปลายทาง (Destination Networks), พื้นที่ OSPF (OSPF Areas) รวมถึงค่าที่ต้องการลบหรือแก้ไข หลังจากได้รับข้อมูล ระบบจะตรวจสอบข้อมูลอุปกรณ์ในฐานข้อมูลเพื่อค้นหาอุปกรณ์ตาม IP หรือชื่ออุปกรณ์ และจัดเตรียมรายการอุปกรณ์ที่ตรงกับข้อมูลดังกล่าว เมื่อข้อมูลถูกส่งต่อไปยังกระบวนการ 24.1 ระบบจะสร้างคำสั่งที่จำเป็นสำหรับการตั้งค่า OSPF เช่น การเพิ่มเครือข่าย การลบเครือข่ายที่ไม่ต้องการ หรือการกำหนดค่าเริ่มต้นใหม่ ขณะเดียวกัน กระบวนการ 24.1 จะเรียกใช้งานกระบวนการย่อย 24.2 เพื่อแปลงค่าที่อยู่ในรูปแบบ CIDR ให้เป็น Subnet Mask ที่เหมาะสมสำหรับการใช้งานในคำสั่ง OSPF Routing หลังจากดำเนินการตั้งค่าหรือปรับเปลี่ยนเสร็จสิ้น ระบบจะตัดการเชื่อมต่อจากอุปกรณ์และส่งผลลัพธ์กลับไปยังผู้ใช้ผ่านหน้าเว็บเพื่อแสดงสถานะการตั้งค่าที่เสร็จสมบูรณ์หรือแจ้งข้อผิดพลาดที่อาจเกิดขึ้น



**ภาพที่ 3-22** Data Flow Diagram Level 1 การกำหนดค่า RIPv2

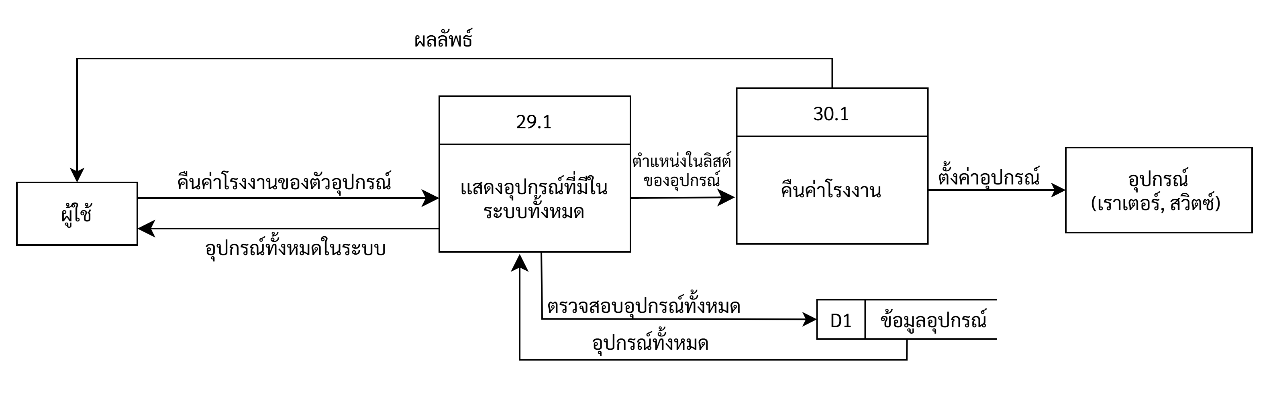
จากภาพ 3-22 DFD Level 1 แสดงถึงการทำงานของกระบวนการตั้งค่า RIP Routing โดยเริ่มต้นที่กระบวนการ 25.1 ซึ่งเป็นการรับค่าผ่านฟอร์มจากหน้าเว็บ เช่น รายการเครือข่ายที่ต้องการเพิ่ม (Destination Networks), การตั้งค่า Auto Summary, รายการเครือข่ายที่ต้องการลบ (Remove Destination Networks) และการปิดการใช้งาน RIP (Disable RIP) หลังจากที่ได้รับข้อมูล ระบบจะตรวจสอบอุปกรณ์ในฐานข้อมูลตามชื่อหรือ IP Address ที่ผู้ใช้กำหนด และรวบรวม IP Address ของอุปกรณ์ที่พบไว้สำหรับดำเนินการตั้งค่า ข้อมูลที่รวบรวมได้จะถูกส่งต่อไปยังกระบวนการ 26.1 ซึ่งมีหน้าที่สร้างและส่งคำสั่งการตั้งค่า RIP Routing ไปยังอุปกรณ์ เช่น การเปิดใช้งาน RIP Version 2, การเพิ่มเครือข่ายใหม่เข้าสู่กระบวนการ Routing, การเปิดหรือปิดฟังก์ชัน Auto Summary, การลบเครือข่ายที่ไม่ต้องการ และการปิดการใช้งาน RIP หากระบุในคำสั่ง คำสั่งเหล่านี้จะถูกส่งผ่านไปยังอุปกรณ์จริงเพื่อดำเนินการตั้งค่าผ่าน CLI และตรวจสอบผลลัพธ์ที่ได้ หลังจากการดำเนินการเสร็จสิ้น ระบบจะตัดการเชื่อมต่อจากอุปกรณ์ และแจ้งผลลัพธ์กลับไปยังผู้ใช้ผ่านหน้าเว็บเพื่อแสดงสถานะการตั้งค่าที่เสร็จสมบูรณ์หรือข้อผิดพลาดที่อาจเกิดขึ้น



**ภาพที่ 3-23** Data Flow Diagram Level 1 การกำหนดค่า EIGRP

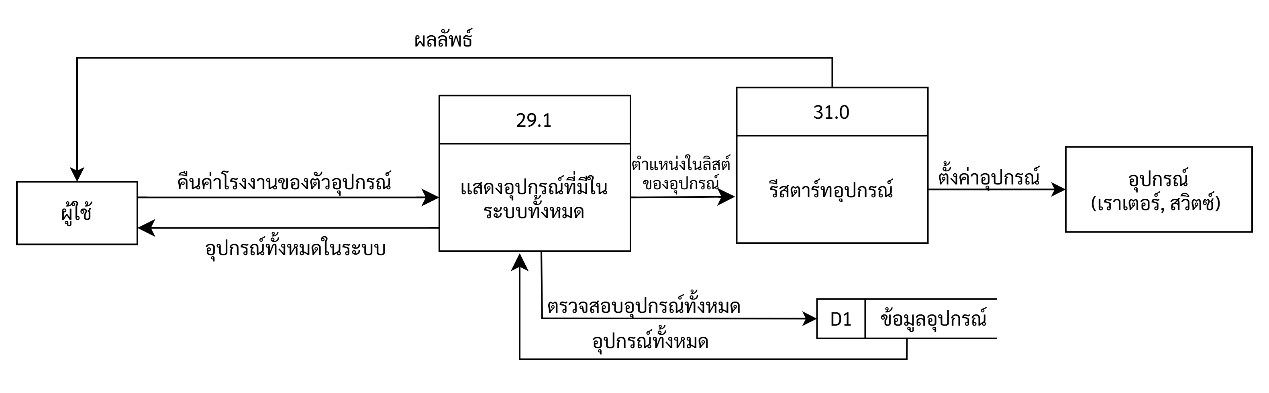
จากภาพ 3-23 DFD Level 1 แสดงถึงกระบวนการตั้งค่า EIGRP Routing โดยเริ่มต้นที่กระบวนการ 27.1 ซึ่งเป็นการรับค่าจากผู้ใช้ผ่านฟอร์ม เช่น หมายเลขกระบวนการ (Process ID), หมายเลข Router ID, รายการเครือข่ายที่ต้องการเพิ่ม (Destination Networks), รายการเครือข่ายที่ต้องการลบ (Remove Destination Networks) และการลบ Process ID ทั้งหมด จากนั้นระบบจะตรวจสอบข้อมูลอุปกรณ์ในฐานข้อมูลตาม IP Address หรือชื่ออุปกรณ์ที่ผู้ใช้ระบุ และรวบรวม IP Address ของอุปกรณ์ที่พบไว้สำหรับดำเนินการตั้งค่า กระบวนการตั้งค่าจะดำเนินการในขั้นตอน 28.1 โดยระบบจะสร้างคำสั่งการตั้งค่า EIGRP Routing เช่น การเปิดใช้งานฟีเจอร์ Routing, การกำหนดค่า Router EIGRP ด้วยหมายเลข Process ID, การเพิ่มเครือข่ายใหม่เข้าสู่กระบวนการ EIGRP Routing โดยมีการแปลงรูปแบบ CIDR เป็น Subnet Mask ในกระบวนการ 28.2, การตั้งค่า Router ID และการลบเครือข่ายที่ไม่ต้องการออกจาก EIGRP Routing นอกจากนี้ หากมีการระบุให้ลบ Process ID ระบบจะส่งคำสั่งเพื่อปิดการใช้งานกระบวนการ EIGRP Routing ที่ระบุไว้

เมื่อดำเนินการเสร็จสิ้น ระบบจะตัดการเชื่อมต่อจากอุปกรณ์ และรายงานผลลัพธ์กลับไปยังผู้ใช้เพื่อแสดงสถานะการตั้งค่าที่สำเร็จหรือข้อผิดพลาดที่เกิดขึ้นระหว่างกระบวนการ



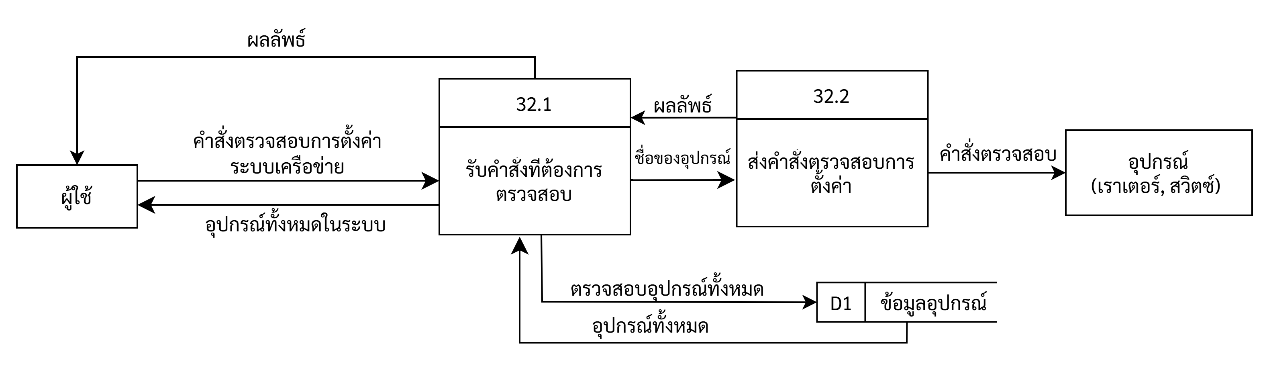
**ภาพที่ 3-24** Data Flow Diagram Level 1 การคืนค่าโรงงาน

จากภาพ 3-24 DFD Level 1 แสดงถึงกระบวนการลบค่าการตั้งค่าของอุปกรณ์เครือข่าย โดยในกระบวนการ 29.1 ระบบจะแสดงรายการอุปกรณ์ทั้งหมดที่มีอยู่ในฐานข้อมูล ให้ผู้ใช้เลือกอุปกรณ์ที่ต้องการดำเนินการลบค่าการตั้งค่า จากนั้นกระบวนการ 30.1 จะรับข้อมูลอุปกรณ์ที่ผู้ใช้เลือกไว้ และดำเนินการคืนค่าโรงงานโดยเชื่อมต่อกับอุปกรณ์ผ่าน SSH ในขั้นตอนนี้ ระบบจะเข้าสู่โหมด Enable ลบค่าการตั้งค่าทั้งหมดใน Startup Configuration ด้วยคำสั่ง erase startup-config เพื่อคืนค่าโรงงาน และดำเนินการสั่งรีโหลดอุปกรณ์เพื่อบูตกลับเข้าสู่สถานะเริ่มต้น เมื่อกระบวนการเสร็จสิ้น ระบบจะลบข้อมูลของอุปกรณ์นั้นออกจากฐานข้อมูลเพื่อให้สอดคล้องกับการคืนค่าโรงงาน และแจ้งสถานะความสำเร็จให้ผู้ใช้ทราบ หากเกิดข้อผิดพลาด เช่น การเชื่อมต่อ SSH ล้มเหลวหรือคำสั่งดำเนินการไม่สำเร็จ ระบบจะแจ้งเตือนผู้ใช้ให้ลองดำเนินการใหม่



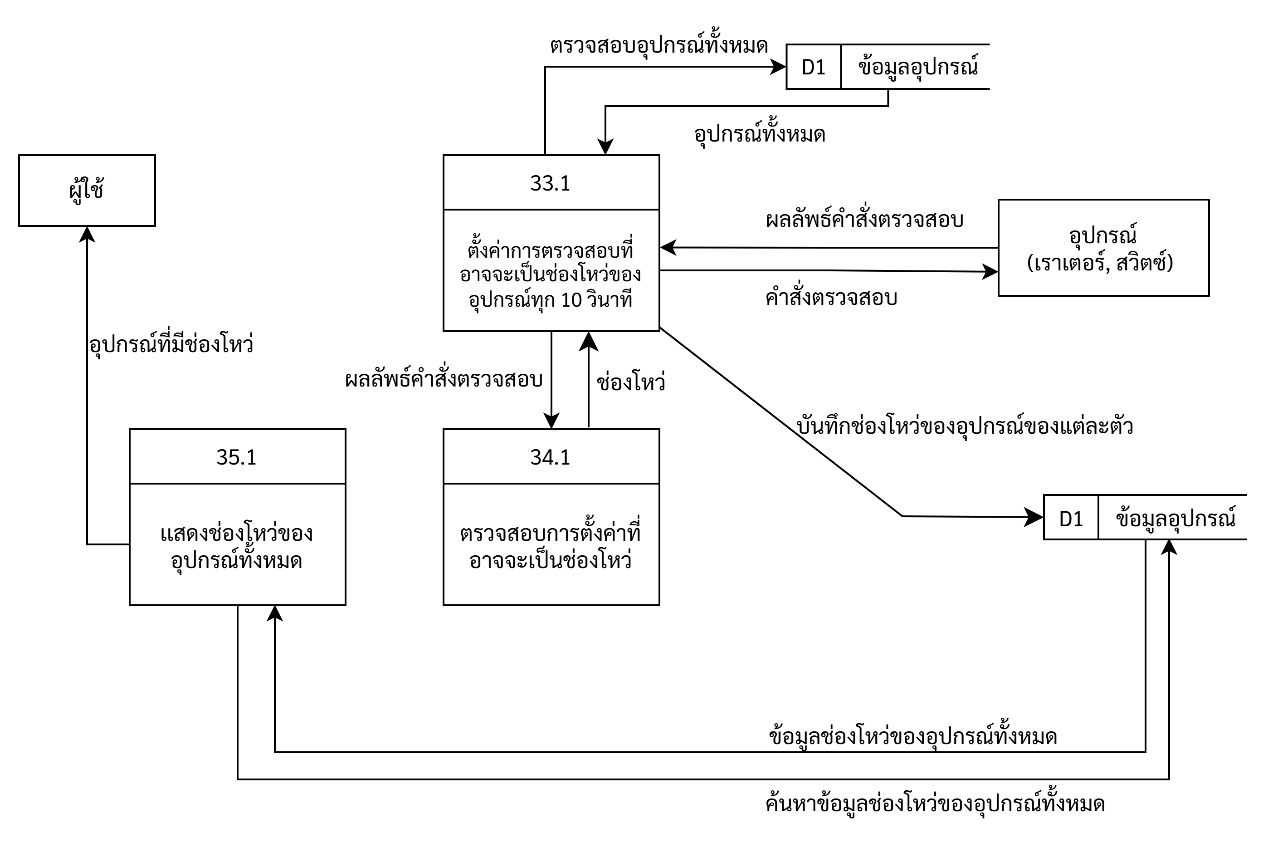
**ภาพที่ 3-25** Data Flow Diagram Level 1 การรีสตาร์ท

จากภาพ 3-25 DFD Level 1 แสดงถึงกระบวนการรีโหลดอุปกรณ์ ซึ่งในกระบวนการ 29.1 ระบบจะแสดงรายการอุปกรณ์ทั้งหมดที่มีอยู่ในฐานข้อมูลเพื่อให้ผู้ใช้สามารถเลือกอุปกรณ์ที่ต้องการดำเนินการรีโหลดได้ เมื่อผู้ใช้เลือกอุปกรณ์ ระบบจะเริ่มต้นกระบวนการ 31.1 ซึ่งจะเชื่อมต่อกับอุปกรณ์นั้นผ่าน SSH โดยใช้ข้อมูลจากฐานข้อมูล เช่น IP, username และ password ในขั้นตอนนี้ ระบบจะเข้าสู่โหมด Enable และส่งคำสั่ง reload เพื่อรีสตาร์ทอุปกรณ์ หากการรีโหลดอุปกรณ์มีการเปลี่ยนแปลงค่าการตั้งค่าบางประการ ระบบจะทำการแสดง modal เพื่อถามผู้ใช้ว่าจะบันทึกการเปลี่ยนแปลงหรือไม่ หากผู้ใช้เลือกที่จะบันทึกการเปลี่ยนแปลง ระบบจะทำการส่งข้อมูลให้กับ backend และทำการรีโหลดอุปกรณ์อีกครั้ง ถ้าผู้ใช้เลือกไม่บันทึก ระบบจะทำการรีโหลดอุปกรณ์โดยไม่บันทึกการเปลี่ยนแปลงใดๆ และแจ้งสถานะให้ผู้ใช้ทราบ



**ภาพที่ 3-26** Data Flow Diagram Level 1 การตรวจสอบการตั้งค่าอุปกรณ์

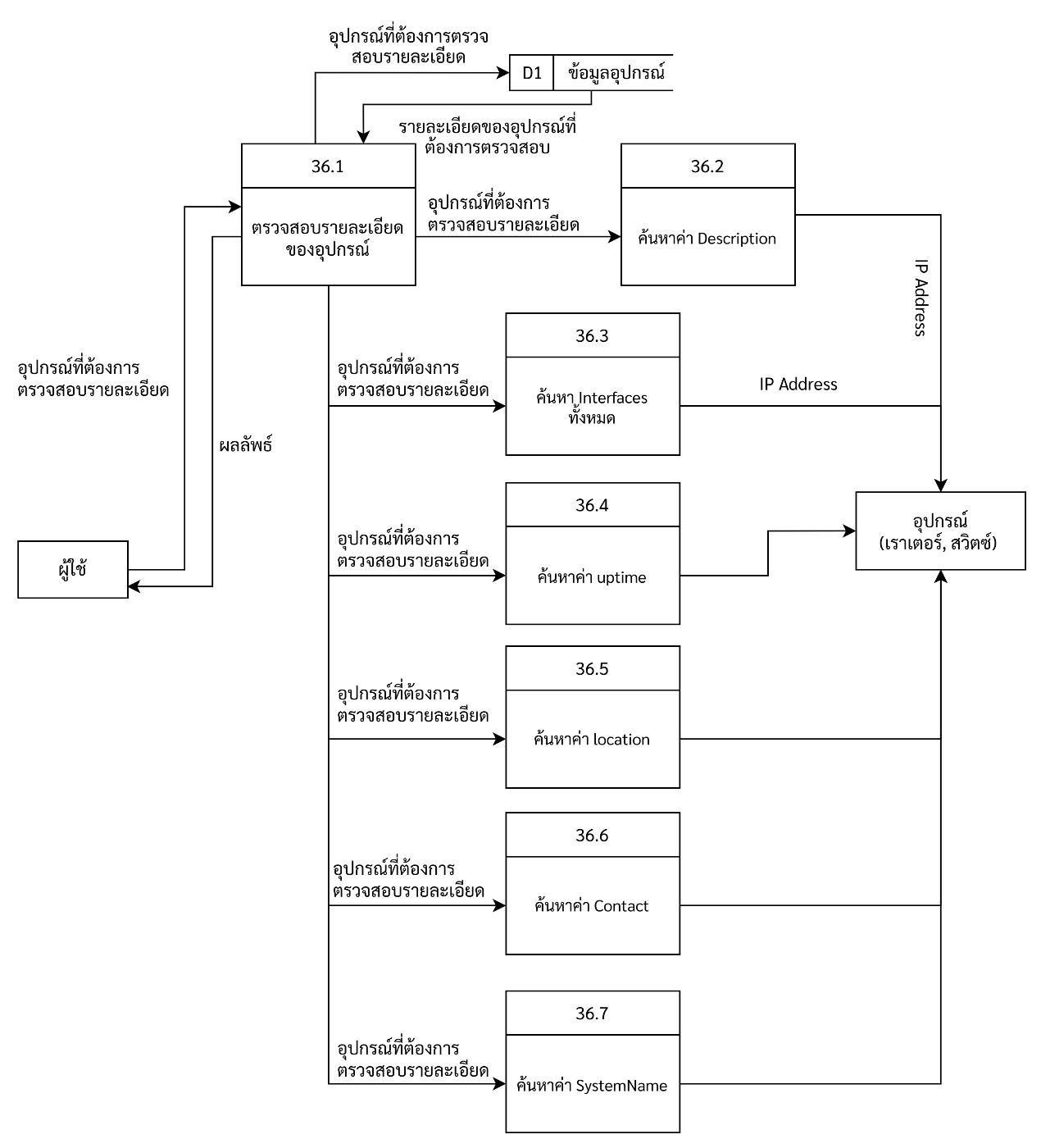
จากภาพ 3-26 DFD Level 1 แสดงถึงกระบวนการแสดงการตั้งค่าของอุปกรณ์ ซึ่งในกระบวนการ 32.1 ระบบจะแสดงรายการอุปกรณ์ทั้งหมดที่มีอยู่ในฐานข้อมูลเพื่อให้ผู้ใช้สามารถเลือกอุปกรณ์ที่ต้องการดูการตั้งค่าได้ เมื่อผู้ใช้เลือกอุปกรณ์ ระบบจะเริ่มต้นกระบวนการ 32.2 ซึ่งจะเชื่อมต่อกับอุปกรณ์นั้นผ่าน SSH โดยใช้ข้อมูลจากฐานข้อมูล เช่น IP, username และ password ในขั้นตอนนี้ ระบบจะเข้าสู่โหมด Enable และส่งคำสั่งต่างๆ ที่ผู้ใช้เลือกให้กับอุปกรณ์นั้น เช่น show running-config, show version, หรือ show ip route หลังจากนั้น ระบบจะดึงข้อมูลที่ได้จากคำสั่งและแสดงผลให้ผู้ใช้เห็นในรูปแบบที่อ่านง่ายบนหน้าจอ หากเกิดข้อผิดพลาดในการเชื่อมต่อหรือดึงข้อมูลจากอุปกรณ์ ระบบจะทำการแสดงข้อความข้อผิดพลาดให้ผู้ใช้ทราบ



**ภาพที่ 3-27** Data Flow Diagram Level 1 การตรวจสอบช่องโหว่ของอุปกรณ์

จากภาพ 3-27 DFD Level 1 ในกระบวนการ 33.1 ระบบจะตั้งค่าการตรวจสอบช่องโหว่ของการตั้งค่าของอุปกรณ์ทุก 10 วินาที โดยจะใช้ฟังก์ชัน fetch\_and\_analyze() ซึ่งทำหน้าที่ดึงข้อมูลการตั้งค่าของอุปกรณ์ทั้งหมดจากฐานข้อมูล แล้วทำการวิเคราะห์ช่องโหว่ด้านความปลอดภัยอย่างต่อเนื่องผ่านการเชื่อมต่อกับอุปกรณ์แต่ละตัว และใช้เวลาในการทำงานซ้ำทุก 10 วินาทีเพื่อให้ข้อมูลการตรวจสอบเป็นปัจจุบัน

ในกระบวนการ 34.1 ระบบจะทำการใช้งานNetworkConfigSecurityChecker ซึ่งเป็นคลาสที่มีหน้าที่ในการตรวจสอบช่องโหว่ของการตั้งค่าของอุปกรณ์ โดยจะทำการตรวจสอบด้านความปลอดภัย เช่น การใช้รหัสผ่านที่อ่อนแอ หรือการตั้งค่าที่ไม่ปลอดภัย เช่น enable password, exec-timeout และ telnet ที่ไม่ควรใช้แทน ssh ในการเชื่อมต่อเมื่อทำการตรวจสอบเสร็จสิ้น ระบบจะบันทึกผลการวิเคราะห์ในฐานข้อมูล รวมถึงคำเตือนเกี่ยวกับความปลอดภัยและเวลาในการอัปเดตข้อมูล จากนั้นในกระบวนการ 35.1 ระบบจะแสดงช่องโหว่เหล่านั้นในหน้าเว็บ frontend เพื่อให้ผู้ใช้สามารถรับข้อมูลการตรวจสอบได้ และสามารถดำเนินการแก้ไขได้ตามความจำเป็น



**ภาพที่ 3-28** Data Flow Diagram Level 1 การตรวจสอบรายละเอียดของอุปกรณ์

จากภาพ 3-28 DFD Level 1 แสดงถึงกระบวนการดึงข้อมูล SNMP จากอุปกรณ์เครือข่าย โดยในกระบวนการ 36.1 ระบบจะรับข้อมูล IP ของอุปกรณ์จากผู้ใช้และค้นหาอุปกรณ์ในฐานข้อมูล หากพบอุปกรณ์ ระบบจะเริ่มต้นกระบวนการ 36.2 ซึ่งจะดึงข้อมูล description ของอุปกรณ์จาก SNMP และตรวจสอบสถานะการตั้งค่า SNMP บนอุปกรณ์ หากอุปกรณ์ไม่สามารถเชื่อมต่อได้ ระบบจะแจ้งข้อผิดพลาดให้ผู้ใช้ทราบ จากนั้นในกระบวนการ 36.3 ระบบจะดึงข้อมูล uptime ของอุปกรณ์ผ่าน SNMP เพื่อแสดงระยะเวลาการทำงานของอุปกรณ์ และในกระบวนการ 36.4 ระบบจะดึงข้อมูล location ของอุปกรณ์จาก SNMP เพื่อแสดงตำแหน่งที่ตั้งของอุปกรณ์ กระบวนการ 36.5 จะดึงข้อมูล contact ของอุปกรณ์เพื่อให้รายละเอียดเกี่ยวกับผู้ดูแลระบบ และในกระบวนการ 36.6 ระบบจะดึงข้อมูล sysname ของอุปกรณ์เพื่อแสดงชื่อของอุปกรณ์ในเครือข่าย ทั้งหมดนี้จะนำข้อมูลที่ได้ไปแสดงบนเว็บให้ผู้ใช้เห็น

* 1. **พจนานุกรมข้อมูล (Data Dictionary)**

**ตารางที่ 3-1** แสดง Data flow ข้อมูลเริ่มต้นของอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0001 |
| Data Flow Name : | ข้อมูลเริ่มต้นของอุปกรณ์ |
| Description : | ค่าต่างๆของข้อมูลเริ่มต้นของอุปกรณ์ |
| Source : | ผู้ใช้ |
| Destination : | Process 2.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ข้อมูลเริ่มต้นของอุปกรณ์ = {  "consoleport": "ค่าของพอร์ตคอนโซล",  "hostname": "ค่าของชื่อโฮสต์",  "domainname": "ค่าของชื่อโดเมน",  "privilege\_password": "ค่าของรหัสผ่านสิทธิพิเศษ",  "ssh\_username": "ค่าของชื่อผู้ใช้งาน SSH",  "ssh\_password": "ค่าของรหัสผ่าน SSH",  "interface": "ค่าของชื่ออินเทอร์เฟซ",  "interface\_type": "ค่าของประเภทอินเทอร์เฟซ",  "ip\_address": "ค่าของที่อยู่ IP",  "save\_startup": "ค่าของการบันทึกการตั้งค่าเริ่มต้น"  } |

|  |  |
| --- | --- |
| Data Flow ID : | 0002 |
| Data Flow Name : | ข้อมูลเริ่มต้นของอุปกรณ์ |
| Description : | ค่าต่างๆของข้อมูลเริ่มต้นของอุปกรณ์ส่งไปยังอุปกรณ์โดย Process 2.1 |
| Source : | Process 1.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่าเริ่มต้นของอุปกรณ์ = {  "consoleport": "ค่าของพอร์ตคอนโซล",  "hostname": "ค่าของชื่อโฮสต์",  "domainname": "ค่าของชื่อโดเมน",  "privilege\_password": "ค่าของรหัสผ่านสิทธิพิเศษ",  "ssh\_username": "ค่าของชื่อผู้ใช้งาน SSH",  "ssh\_password": "ค่าของรหัสผ่าน SSH",  "interface": "ค่าของชื่ออินเทอร์เฟซ",  "interface\_type": "ค่าของประเภทอินเทอร์เฟซ",  "ip\_address": "ค่าของที่อยู่ IP",  "save\_startup": "ค่าของการบันทึกการตั้งค่าเริ่มต้น"  } |

**ตารางที่ 3-2** แสดง Data flow ข้อมูลตั้งค่าเริ่มต้นของอุปกรณ์

**ตารางที่ 3-3** แสดง Data flow ข้อมูลการเพิ่มอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0003 |
| Data Flow Name : | รายละเอียดอุปกรณ์ |
| Description : | ข้อมูลต่างๆของอุปกกรณ์ |
| Source : | ผู้ใช้ |
| Destination : | ฐานข้อมูล D1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ = {  "name": "ชื่ออุปกรณ์",  "ip\_address": "ที่อยู่ IP ของอุปกรณ์",  "privilege\_password": "รหัสผ่านสิทธิพิเศษของอุปกรณ์",  "ssh\_username": "ชื่อผู้ใช้งาน SSH ของอุปกรณ์",  "ssh\_password": "รหัสผ่าน SSH ของอุปกรณ์",  "device\_info": {  "device\_type": "ประเภทของอุปกรณ์ (เช่น 'cisco\_ios')",  "host": "ที่อยู่ IP ของอุปกรณ์ (ถ้าใช้ในการเชื่อมต่อ)",  "username": "ชื่อผู้ใช้สำหรับการเชื่อมต่อ SSH",  "password": "รหัสผ่านสำหรับการเชื่อมต่อ SSH",  "secret": "รหัสผ่าน enable สำหรับเข้าสู่โหมดพิเศษ",  "timeout": "ระยะเวลาเชื่อมต่อ"  }  } |

**ตารางที่ 3-4** แสดง Data flow รายละเอียดของข้อมูลอุปกรณ์ที่มีในระบบทั้งหมด

|  |  |
| --- | --- |
| Data Flow ID : | 0004 |
| Data Flow Name : | ข้อมูลทั้งหมดของอุปกรณ์ |
| Description : | แสดงข้อมูลทั้งหมดที่มีในฐานข้อมูลให้แก่ผู้ใช้ |
| Source : | ฐานข้อมูล D1 |
| Destination : | ผู้ใช้ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ทั้งหมด = {  "devices": [  {  "name": "ชื่ออุปกรณ์",  "ip\_address": "ที่อยู่ IP ของอุปกรณ์",  "ssh\_username": "ชื่อผู้ใช้งาน SSH ของอุปกรณ์",  "ssh\_password": "รหัสผ่าน SSH ของอุปกรณ์",  "secret\_password": "รหัสผ่านสิทธิพิเศษของอุปกรณ์",  "actions": "การจัดการ เช่น การรีสตาร์ท, การตรวจสอบสถานะ"  },  {  "name": "ชื่ออุปกรณ์ที่สอง",  "ip\_address": "ที่อยู่ IP ของอุปกรณ์ที่สอง",  "ssh\_username": "ชื่อผู้ใช้งาน SSH ของอุปกรณ์ที่สอง",  "ssh\_password": "รหัสผ่าน SSH ของอุปกรณ์ที่สอง",  "secret\_password": "รหัสผ่านสิทธิพิเศษของอุปกรณ์ที่สอง",  "actions": "การจัดการ เช่น การรีสตาร์ท, การตรวจสอบสถานะ"  }  ]  } |

**ตารางที่ 3-5** แสดง Data flow การลบอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0005 |
| Data Flow Name : | ข้อมูลของอุปกรณ์ |
| Description : | ข้อมูลของอุปกรณ์สำหรับการลบออกจากฐานข้อมูล D1 |
| Source : | Process 4.1 |
| Destination : | ฐานข้อมูล D1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ที่ลบ = {  "deleted\_device": {  "ip\_address": "ที่อยู่ IP ของอุปกรณ์ที่ถูกลบ"  }  } |

**ตารางที่ 3-6** แสดง Data flow การแก้ไขอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0006 |
| Data Flow Name : | ข้อมูลของอุปกรณ์ที่ต้องการแก้ไข |
| Description : | ข้อมูลของอุปกรณ์ที่ต้องการแก้ไข |
| Source : | Process 4.1 |
| Destination : | Process 7.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ต้องแก้ไข = {  "device\_to\_edit": {  "name": "ชื่ออุปกรณ์",  "ip\_address": "ที่อยู่ IP ของอุปกรณ์",  "ssh\_username": "ชื่อผู้ใช้งาน SSH",  "ssh\_password": "รหัสผ่าน SSH",  "privilege\_password": "รหัสผ่านสิทธิพิเศษ"  }  } |

**ตารางที่ 3-7** แสดง Data flow การอัปเดตข้อมูลที่แก้ไข

|  |  |
| --- | --- |
| Data Flow ID : | 0007 |
| Data Flow Name : | ข้อมูลของอุปกรณ์ที่แก้ไข |
| Description : | ข้อมูลของอุปกรณ์ที่แก้ไขจาก Process 6.1 |
| Source : | Process 4.1 |
| Destination : | ฐานข้อมูล D1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์แก้ไข = {  "device\_to\_update": {  "name": "ชื่ออุปกรณ์",  "current\_ip": "ที่อยู่ IP เดิม",  "new\_ip": "ที่อยู่ IP ใหม่",  "username": "ชื่อผู้ใช้งาน SSH",  "password": "รหัสผ่าน SSH",  "secret": "รหัสผ่านสิทธิพิเศษ"  }  } |

**ตารางที่ 3-8** แสดง Data flow การ Ping อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0008 |
| Data Flow Name : | ข้อมูลของอุปกรณ์ที่ต้องการ Ping |
| Description : | ข้อมูลของอุปกรณ์ที่ต้องการ Ping |
| Source : | Process 4.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ที่ต้องการ Ping = {  "device\_to\_ping": {  "ip\_address": "ที่อยู่ IP ของอุปกรณ์"  }  } |

**ตารางที่ 3-9** แสดง Data flow การกำหนดค่าพื้นฐาน

|  |  |
| --- | --- |
| Data Flow ID : | 0009 |
| Data Flow Name : | ค่าต่างๆกำหนดค่าพื้นฐานจากผู้ใช้ |
| Description : | ค่าต่างๆกำหนดค่าพื้นฐานในเมนู Basic settings |
| Source : | ผู้ใช้ |
| Destination : | Process 10.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนดพื้นฐานของอุปกรณ์ = {  "device\_name": ชื่ออุปกรณ์ (string),  "hostname": ชื่อโฮสต์ (string),  "secret\_password": รหัสผ่านลับ (string),  "banner": ข้อความ Banner (string),  "many\_hostname": รายชื่อโฮสต์ (string, แยกด้วย ","),  "enable\_password\_encryp": เปิดใช้งานการเข้ารหัส (boolean),  "disable\_password\_encryp": ปิดการเข้ารหัส (boolean),  "username": ชื่อผู้ใช้งาน SSH (string),  "password": รหัสผ่าน SSH (string)  } |

**ตารางที่ 3-10** แสดง Data flow กำหนดค่าพื้นฐานไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0010 |
| Data Flow Name : | ค่าต่างๆกำหนดค่าพื้นฐาน |
| Description : | ค่าต่างๆที่รับมาจาก Process 9.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 9.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนดพื้นฐานของอุปกรณ์ = {  "device\_name": ชื่ออุปกรณ์ (string),  "hostname": ชื่อโฮสต์ (string),  "secret\_password": รหัสผ่านลับ (string),  "banner": ข้อความ Banner (string),  "many\_hostname": รายชื่อโฮสต์ (string, แยกด้วย ","),  "enable\_password\_encryp": เปิดใช้งานการเข้ารหัส (boolean),  "disable\_password\_encryp": ปิดการเข้ารหัส (boolean),  "username": ชื่อผู้ใช้งาน SSH (string),  "password": รหัสผ่าน SSH (string)  } |

**ตารางที่ 3-11** แสดง Data flow การกำหนดค่า Network Interfaces

|  |  |
| --- | --- |
| Data Flow ID : | 0011 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Network Interfaces จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า Network Interfaces |
| Source : | ผู้ใช้ |
| Destination : | Process 12.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Network Interfaces ของอุปกรณ์ = {  "device\_name": ชื่ออุปกรณ์ (string),  "many\_hostname": รายชื่อโฮสต์ (string, แยกด้วย ","),  "interfaces\_ipv4": ชื่ออินเทอร์เฟซ IPv4 (string),  "config\_type": ประเภทการกำหนดค่า IPv4 ("dhcp\_ipv4" หรือ "static\_ipv4") (string),  "dhcp\_ipv4": กำหนดค่า DHCP สำหรับ IPv4 (boolean),  "ip\_address\_ipv4": ที่อยู่ IP ของ IPv4 (string),  "subnet\_mask\_ipv4": มาส์กซับเน็ต IPv4 (string),  "enable\_ipv4": เปิดใช้งาน IPv4 (boolean),  "disable\_ipv4": ปิดการใช้งาน IPv4 (boolean),  "delete\_ipv4": ลบการกำหนดค่า IPv4 (boolean),  "interfaces\_ipv6": ชื่ออินเทอร์เฟซ IPv6 (string),  "dhcp\_ipv6": กำหนดค่า DHCP สำหรับ IPv6 (boolean),  "ip\_address\_ipv6": ที่อยู่ IP ของ IPv6 (string),  "enable\_ipv6": เปิดใช้งาน IPv6 (boolean),  "disable\_ipv6": ปิดการใช้งาน IPv6 (boolean),  "delete\_ipv6": ลบการกำหนดค่า IPv6 (boolean),  "interfaces\_du": ชื่ออินเทอร์เฟซดูเพล็กซ์ (string),  "speed\_duplex": ความเร็วและโหมดดูเพล็กซ์ (string)  } |

**ตารางที่ 3-12** แสดง Data flow การกำหนดค่า Network Interfaces ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0012 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Network Interfaces |
| Description : | ค่าต่างๆที่รับมาจาก Process 11.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 11.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Network Interfaces ของอุปกรณ์ = {  "device\_name": ชื่ออุปกรณ์ (string),  "many\_hostname": รายชื่อโฮสต์ (string, แยกด้วย ","),  "interfaces\_ipv4": ชื่ออินเทอร์เฟซ IPv4 (string),  "config\_type": ประเภทการกำหนดค่า IPv4 ("dhcp\_ipv4" หรือ "static\_ipv4") (string),  "dhcp\_ipv4": กำหนดค่า DHCP สำหรับ IPv4 (boolean),  "ip\_address\_ipv4": ที่อยู่ IP ของ IPv4 (string),  "subnet\_mask\_ipv4": มาส์กซับเน็ต IPv4 (string),  "enable\_ipv4": เปิดใช้งาน IPv4 (boolean),  "disable\_ipv4": ปิดการใช้งาน IPv4 (boolean),  "delete\_ipv4": ลบการกำหนดค่า IPv4 (boolean),  "interfaces\_ipv6": ชื่ออินเทอร์เฟซ IPv6 (string),  "dhcp\_ipv6": กำหนดค่า DHCP สำหรับ IPv6 (boolean),  "ip\_address\_ipv6": ที่อยู่ IP ของ IPv6 (string),  "enable\_ipv6": เปิดใช้งาน IPv6 (boolean),  "disable\_ipv6": ปิดการใช้งาน IPv6 (boolean),  "delete\_ipv6": ลบการกำหนดค่า IPv6 (boolean),  "interfaces\_du": ชื่ออินเทอร์เฟซดูเพล็กซ์ (string),  "speed\_duplex": ความเร็วและโหมดดูเพล็กซ์ (string)  } |

**ตารางที่ 3-13** แสดง Data flow การกำหนดค่า VLAN

|  |  |
| --- | --- |
| Data Flow ID : | 0013 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า VLAN จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า VLAN |
| Source : | ผู้ใช้ |
| Destination : | Process 14.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด VLAN = {  "device\_name": ชื่ออุปกรณ์ (string),  "many\_hostname": รายชื่อโฮสต์ (string),  "vlan\_id": รายการ VLAN ID (string),  "vlan\_id\_del": รายการ VLAN ID ที่จะลบ (string),  "vlan\_ids\_to\_change": รายการ VLAN ID ที่จะเปลี่ยน (array of strings),  "vlan\_names\_to\_change": รายการชื่อ VLAN ที่จะเปลี่ยน (array of strings),  "enable\_vlans": เปิดการใช้งาน VLAN (boolean),  "disable\_vlans": ปิดการใช้งาน VLAN (boolean),  "vlan\_id\_enable": VLAN ID ที่จะเปิดใช้งาน (string),  "vlan\_id\_disable": VLAN ID ที่จะปิดใช้งาน (string),  "del\_vlan\_dat": ลบข้อมูล VLAN (boolean),  "access\_vlans": VLAN สำหรับ Access (string),  "access\_interface": ชื่ออินเทอร์เฟซสำหรับ Access (string),  "access\_vlan\_id": VLAN ID สำหรับ Access (string),  "disable\_dtp": ปิดการใช้งาน DTP (boolean),  "trunk\_ports": พอร์ตที่ตั้งเป็น Trunk (string),  "trunk\_mode\_select": โหมด Trunk ที่เลือก (string),  "trunk\_interface": อินเทอร์เฟซสำหรับ Trunk (string),  "trunk\_native": VLAN Native ของ Trunk (string),  "allow\_vlan": VLAN ที่อนุญาตบน Trunk (string)  } |

**ตารางที่ 3-14** แสดง Data flow การกำหนดค่า VLAN ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0014 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า VLAN |
| Description : | ค่าต่างๆที่รับมาจาก Process 13.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 13.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด VLAN = {  "device\_name": ชื่ออุปกรณ์ (string),  "vlan\_range": ช่วง VLAN ที่ต้องการสร้าง (array of integers),  "vlan\_range\_del": ช่วง VLAN ที่ต้องการลบ (array of integers),  "vlan\_changes": การเปลี่ยนแปลง VLAN เช่น (vlan\_id, new\_name) (array of tuples),  "vlan\_range\_enable": ช่วง VLAN ที่ต้องการเปิดใช้งาน (array of integers),  "vlan\_range\_disable": ช่วง VLAN ที่ต้องการปิดใช้งาน (array of integers),  "access\_vlans": VLAN สำหรับ Access (string),  "access\_interface": ชื่ออินเทอร์เฟซสำหรับ Access (string),  "access\_vlan\_id": VLAN ID สำหรับ Access (string),  "disable\_dtp": ปิดการใช้งาน DTP (boolean),  "trunk\_ports": พอร์ตที่ตั้งเป็น Trunk (string),  "trunk\_mode\_select": โหมด Trunk ที่เลือก (string),  "trunk\_interface": อินเทอร์เฟซสำหรับ Trunk (string),  "trunk\_native": VLAN Native ของ Trunk (string),  "allow\_vlan": VLAN ที่อนุญาตบน Trunk (string),  "del\_vlan\_dat": ลบข้อมูล VLAN.dat (boolean)  } |

**ตารางที่ 3-15** แสดง Data flow การกำหนดค่า Management

|  |  |
| --- | --- |
| Data Flow ID : | 0015 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Management จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า Management |
| Source : | ผู้ใช้ |
| Destination : | Process 16.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Management = {  "device\_name": ชื่ออุปกรณ์ (string),  "many\_hostname": รายชื่อโฮสต์ที่ต้องการตั้งค่า (string, แยกด้วย ","),  "password\_vty": รหัสผ่านสำหรับ VTY (string),  "authen\_method": วิธีการตรวจสอบสิทธิ์สำหรับ VTY (string),  "exec\_timeout\_vty": เวลาหมดเวลาในการใช้งาน VTY (integer),  "login\_method": วิธีการเข้าสู่ระบบสำหรับ VTY (string),  "logging\_sync\_vty": การเปิดหรือปิดการซิงค์การบันทึกใน VTY (boolean),  "password\_console": รหัสผ่านสำหรับ Console (string),  "exec\_timeout\_console": เวลาหมดเวลาในการใช้งาน Console (integer),  "logging\_sync\_console": การเปิดหรือปิดการซิงค์การบันทึกใน Console (boolean),  "authen\_method\_con": วิธีการตรวจสอบสิทธิ์สำหรับ Console (string),  "pool\_name": ชื่อของ DHCP Pool (string),  "network": เครือข่ายสำหรับ DHCP (string),  "dhcp\_subnet": Subnet สำหรับ DHCP (string),  "dhcp\_exclude": การยกเว้น IP สำหรับ DHCP (string),  "default\_router": Router ค่าเริ่มต้นสำหรับ DHCP (string),  "dns\_server": DNS Server สำหรับ DHCP (string),  "domain\_name": ชื่อโดเมนสำหรับ DHCP (string),  "pool\_name\_del": ชื่อ DHCP Pool ที่ต้องการลบ (string),  "ntp\_server": NTP Server สำหรับการตั้งเวลา (string),  "time\_zone\_name": โซนเวลา (string),  "hour\_offset": การปรับเวลา (integer),  "snmp\_ro": รหัส SNMP Read-Only (string),  "snmp\_rw": รหัส SNMP Read-Write (string),  "snmp\_contact": ข้อมูลการติดต่อสำหรับ SNMP (string),  "snmp\_location": ตำแหน่งของอุปกรณ์ใน SNMP (string),  "enable\_cdp": การเปิดใช้งาน CDP (boolean),  "disable\_cdp": การปิดใช้งาน CDP (boolean),  "enable\_lldp": การเปิดใช้งาน LLDP (boolean),  "disable\_lldp": การปิดใช้งาน LLDP (boolean),  "device\_ips": รายชื่อ IP ของอุปกรณ์ที่ต้องการตั้งค่า (array of strings)  } |

**ตารางที่ 3-16** แสดง Data flow การกำหนดค่า Management ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0016 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Management |
| Description : | ค่าต่างๆที่รับมาจาก Process 15.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 15.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Management = {  "device\_name": ชื่ออุปกรณ์ (string),  "password\_vty": รหัสผ่านสำหรับ VTY (string),  "authen\_method": วิธีการตรวจสอบสิทธิ์สำหรับ VTY (string),  "exec\_timeout\_vty": เวลาหมดเวลาในการใช้งาน VTY (integer),  "login\_method": วิธีการเข้าสู่ระบบสำหรับ VTY (string),  "logging\_sync\_vty": การเปิดหรือปิดการซิงค์การบันทึกใน VTY (boolean),  "password\_console": รหัสผ่านสำหรับ Console (string),  "exec\_timeout\_console": เวลาหมดเวลาในการใช้งาน Console (integer),  "logging\_sync\_console": การเปิดหรือปิดการซิงค์การบันทึกใน Console (boolean),  "authen\_method\_con": วิธีการตรวจสอบสิทธิ์สำหรับ Console (string),  "pool\_name": ชื่อของ DHCP Pool (string),  "network": เครือข่ายสำหรับ DHCP (string),  "dhcp\_subnet": Subnet สำหรับ DHCP (string),  "dhcp\_exclude": การยกเว้น IP สำหรับ DHCP (string),  "default\_router": Router ค่าเริ่มต้นสำหรับ DHCP (string),  "dns\_server": DNS Server สำหรับ DHCP (string),  "domain\_name": ชื่อโดเมนสำหรับ DHCP (string),  "pool\_name\_del": ชื่อ DHCP Pool ที่ต้องการลบ (string),  "ntp\_server": NTP Server สำหรับการตั้งเวลา (string),  "time\_zone\_name": โซนเวลา (string),  "hour\_offset": การปรับเวลา (integer),  "snmp\_ro": รหัส SNMP Read-Only (string),  "snmp\_rw": รหัส SNMP Read-Write (string),  "snmp\_contact": ข้อมูลการติดต่อสำหรับ SNMP (string),  "snmp\_location": ตำแหน่งของอุปกรณ์ใน SNMP (string),  "enable\_cdp": การเปิดใช้งาน CDP (boolean),  "disable\_cdp": การปิดใช้งาน CDP (boolean),  "enable\_lldp": การเปิดใช้งาน LLDP (boolean),  "disable\_lldp": การปิดใช้งาน LLDP (boolean)} |

**ตารางที่ 3-17** แสดง Data flow การกำหนดค่า Spanning Tree

|  |  |
| --- | --- |
| Data Flow ID : | 0017 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Spanning Tree จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า Spanning Tree |
| Source : | ผู้ใช้ |
| Destination : | Process 18.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Spanning Tree = {  "device\_name": ชื่ออุปกรณ์ (string),  "many\_hostname": รายชื่อโฮสต์หลายตัว (string),  "stp\_mode": โหมด Spanning Tree (string),  "root\_primary": การตั้งค่า Root Primary (boolean),  "root\_vlan\_id": VLAN ID สำหรับ Root Primary (string, เฉพาะเมื่อ root\_primary เป็น True),  "root\_secondary": การตั้งค่า Root Secondary (boolean),  "root\_secondary\_vlan\_id": VLAN ID สำหรับ Root Secondary (string, เฉพาะเมื่อ root\_secondary เป็น True),  "portfast\_enable": การเปิดใช้งาน Portfast (boolean),  "portfast\_disable": การปิดใช้งาน Portfast (boolean),  "portfast\_int\_enable": การเปิดใช้งาน Portfast สำหรับพอร์ต (string, เฉพาะเมื่อ portfast\_enable เป็น True),  "portfast\_int\_disable": การปิดใช้งาน Portfast สำหรับพอร์ต (string, เฉพาะเมื่อ portfast\_disable เป็น True)  } |

**ตารางที่ 3-18** แสดง Data flow การกำหนดค่า Spanning Tree ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0018 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Spanning Tree |
| Description : | ค่าต่างๆที่รับมาจาก Process 17.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 17.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Spanning Tree = {  "stp\_mode": โหมด Spanning Tree (string),  "root\_primary": การตั้งค่า Root Primary (boolean),  "root\_vlan\_id": VLAN ID สำหรับ Root Primary (string, เฉพาะเมื่อ root\_primary เป็น True),  "root\_secondary": การตั้งค่า Root Secondary (boolean),  "root\_secondary\_vlan\_id": VLAN ID สำหรับ Root Secondary (string, เฉพาะเมื่อ root\_secondary เป็น True),  "portfast\_enable": การเปิดใช้งาน Portfast (boolean),  "portfast\_disable": การปิดใช้งาน Portfast (boolean),  "portfast\_int\_enable": การเปิดใช้งาน Portfast สำหรับพอร์ต (string, เฉพาะเมื่อ portfast\_enable เป็น True),  "portfast\_int\_disable": การปิดใช้งาน Portfast สำหรับพอร์ต (string, เฉพาะเมื่อ portfast\_disable เป็น True)  } |

**ตารางที่ 3-19** แสดง Data flow การกำหนดค่า Aggregation

|  |  |
| --- | --- |
| Data Flow ID : | 0019 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Aggregation จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า Aggregation |
| Source : | ผู้ใช้ |
| Destination : | Process 20.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Aggregation = {  "etherchannel\_interfaces": ค่าการตั้งค่า Interfaces สำหรับ EtherChannel (string),  "channel\_group\_number": ค่าหมายเลข Channel Group สำหรับ EtherChannel (string),  "pagp\_mode": ค่าโหมด PAgP ที่เลือก (list of strings, เช่น "auto", "desirable"),  "etherchannel\_interfaces\_lacp": ค่าการตั้งค่า Interfaces สำหรับ LACP (string),  "channel\_group\_number\_lacp": ค่าหมายเลข Channel Group สำหรับ LACP (string),  "lacp\_mode": โหมด LACP ที่ผู้ใช้เลือก (list of strings, เช่น "active", "passive"),  "etherchannel\_interfaces\_lacp\_delete": การลบการตั้งค่า EtherChannel สำหรับ LACP (string)  } |

**ตารางที่ 3-20** แสดง Data flow การกำหนดค่า Aggregation ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0020 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Aggregation |
| Description : | ค่าต่างๆที่รับมาจาก Process 19.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 19.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Aggregation = {  "etherchannel\_interfaces": ค่าการตั้งค่า Interfaces สำหรับ EtherChannel (string),  "channel\_group\_number": ค่าหมายเลข Channel Group สำหรับ EtherChannel (string),  "pagp\_mode": โหมด PAgP ที่ผู้ใช้เลือก (list of strings, เช่น "desirable", "auto"),  "etherchannel\_interfaces\_lacp": ค่าการตั้งค่า Interfaces สำหรับ LACP (string),  "channel\_group\_number\_lacp": ค่าหมายเลข Channel Group สำหรับ LACP (string),  "lacp\_mode": โหมด LACP ที่ผู้ใช้เลือก (list of strings, เช่น "active", "passive"),  "etherchannel\_interfaces\_lacp\_delete": การลบการตั้งค่า EtherChannel สำหรับ LACP (string)  } |

**ตารางที่ 3-21** แสดง Data flow การกำหนดค่า Routing – Static Route

|  |  |
| --- | --- |
| Data Flow ID : | 0021 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Static Route จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า Static Route |
| Source : | ผู้ใช้ |
| Destination : | Process 22.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Static Route = {  "destination\_networks": รายการของเครือข่ายปลายทาง (list of strings),  "exit\_interfaces\_or\_next\_hops": รายการของ interface หรือ next hop ที่ใช้ในการกำหนดเส้นทาง (list of strings),  "default\_route": ค่าการตั้งค่า default route (string),  "remove\_default": ค่าที่ผู้ใช้ตั้งค่าลบ default route (boolean),  "remove\_destination\_networks": รายการของเครือข่ายที่ต้องการลบ (list of strings),  "remove\_exit\_interfaces\_or\_next\_hops": รายการของ next hop หรือ interfaces ที่ต้องการลบ (list of strings)  } |

**ตารางที่ 3-22** แสดง Data flow การกำหนดค่า Routing – Static Route ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0022 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า Static Route |
| Description : | ค่าต่างๆที่รับมาจาก Process 21.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 21.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด Static Route = {  "destination\_networks": "รายการของเครือข่ายปลายทาง (list of strings)",  "exit\_interfaces\_or\_next\_hops": "รายการของ interface หรือ next hop ที่ใช้ในการกำหนดเส้นทาง (list of strings)",  "default\_route": "ค่าการตั้งค่า default route (string)",  "remove\_default": "ค่าที่ผู้ใช้ตั้งค่าลบ default route (boolean)",  "remove\_destination\_networks": "รายการของเครือข่ายที่ต้องการลบ (list of strings)",  "remove\_exit\_interfaces\_or\_next\_hops": "รายการของ next hop หรือ interfaces ที่ต้องการลบ (list of strings)"  } |

**ตารางที่ 3-23** แสดง Data flow การกำหนดค่า Routing – OSPF

|  |  |
| --- | --- |
| Data Flow ID : | 0023 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า OSPF จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า OSPF |
| Source : | ผู้ใช้ |
| Destination : | Process 24.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด OSPF = {  "destination\_networks": "รายการของเครือข่ายปลายทางที่ต้องการเพิ่มใน OSPF (list of strings)",  "ospf\_areas": "รายการของ OSPF areas ที่ใช้ในการกำหนดเส้นทาง (list of strings)",  "process\_id": "หมายเลขของ OSPF process (string)",  "router\_id": "ค่า router ID ที่กำหนดสำหรับ OSPF (string)",  "remove\_destination\_networks": "รายการของเครือข่ายที่ต้องการลบจาก OSPF (list of strings)",  "remove\_ospf\_areas": "รายการของ OSPF areas ที่ต้องการลบ (list of strings)",  "delete\_process\_id": "ค่าที่ผู้ใช้ตั้งค่าลบ OSPF process (boolean)",  "process\_id\_input": "ค่า input สำหรับการกำหนด OSPF process ID (string)"  } |

**ตารางที่ 3-24** แสดง Data flow การกำหนดค่า Routing – OSPF ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0024 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า OSPF |
| Description : | ค่าต่างๆที่รับมาจาก Process 23.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 23.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด OSPF = {  "process\_id": "หมายเลขของ OSPF process ที่ใช้ในการกำหนดค่า (string)",  "destination\_networks": "รายการของเครือข่ายปลายทางที่ต้องการเพิ่มใน OSPF (list of strings)",  "ospf\_areas": "รายการของ OSPF areas ที่ใช้ในการกำหนดเส้นทาง (list of strings)",  "router\_id": "ค่า router ID ที่กำหนดสำหรับ OSPF (string)",  "remove\_destination\_networks": "รายการของเครือข่ายที่ต้องการลบจาก OSPF (list of strings)",  "remove\_ospf\_areas": "รายการของ OSPF areas ที่ต้องการลบ (list of strings)",  "delete\_process\_id": "ค่าที่ผู้ใช้ตั้งค่าลบ OSPF process (boolean)",  "process\_id\_input": "ค่า input สำหรับการกำหนด OSPF process ID (string)"  } |

**ตารางที่ 3-25** แสดง Data flow การกำหนดค่า Routing – RIPv2

|  |  |
| --- | --- |
| Data Flow ID : | 0025 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า RIPv2 จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า RIPv2 |
| Source : | ผู้ใช้ |
| Destination : | Process 26.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด RIPv2 = {  "destination\_networks": "รายการของเครือข่ายปลายทางที่ต้องการเพิ่มใน RIPv2 (list of strings)",  "auto\_summary": "ค่าการตั้งค่า auto-summary ของ RIPv2 (boolean)",  "remove\_destination\_networks": "รายการของเครือข่ายที่ต้องการลบจาก RIPv2 (list of strings)",  "disable\_rip": "ค่าการตั้งค่า disable สำหรับ RIP (boolean)"  } |

**ตารางที่ 3-26** แสดง Data flow การกำหนดค่า Routing – RIPv2 ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0026 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า RIPv2 |
| Description : | ค่าต่างๆที่รับมาจาก Process 25.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 25.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด RIPv2 = {  "destination\_networks": "รายการของเครือข่ายที่ต้องการเพิ่มใน RIPv2 (list of strings)",  "auto\_summary": "ค่าการตั้งค่า auto-summary ของ RIPv2 (string, 'Enable' หรือ 'Disable')",  "remove\_destination\_networks": "รายการของเครือข่ายที่ต้องการลบจาก RIPv2 (list of strings)",  "disable\_rip": "ค่าการตั้งค่า disable สำหรับ RIP (boolean)"  } |

**ตารางที่ 3-27** แสดง Data flow การกำหนดค่า Routing – EIGRP

|  |  |
| --- | --- |
| Data Flow ID : | 0027 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า EIGRP จากผู้ใช้ |
| Description : | ค่าต่าง ๆ ที่รับมาจากผู้ใช้กำหนดค่า EIGRP |
| Source : | ผู้ใช้ |
| Destination : | Process 28.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด EIGRP = {  "process\_id": "หมายเลขของ EIGRP process ID (string)",  "router\_id": "ค่าของ Router ID สำหรับ EIGRP (string)",  "destination\_networks": "รายการของเครือข่ายที่ต้องการกำหนดให้ EIGRP (list of strings)",  "remove\_destination\_networks": "รายการของเครือข่ายที่ต้องการลบออกจาก EIGRP (list of strings)",  "delete\_process\_id": "ค่าที่ใช้ในการลบ EIGRP process (boolean)",  "process\_id\_input": "หมายเลขของ EIGRP process ID ที่ต้องการลบ (list of strings)"  } |

**ตารางที่ 3-28** แสดง Data flow การกำหนดค่า Routing – EIGRP ไปที่อุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0028 |
| Data Flow Name : | ค่าต่าง ๆ กำหนดค่า EIGRP |
| Description : | ค่าต่างๆที่รับมาจาก Process 27.1 และส่งไปตั้งค่าอุปกรณ์ |
| Source : | Process 27.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ค่ากำหนด EIGRP = {  "process\_id": "หมายเลขของ EIGRP process ID (string)",  "router\_id": "ค่าของ Router ID สำหรับ EIGRP (string)",  "destination\_networks": "รายการของเครือข่ายที่ต้องการกำหนดให้ EIGRP (list of strings)",  "remove\_destination\_networks": "รายการของเครือข่ายที่ต้องการลบออกจาก EIGRP (list of strings)",  "delete\_process\_id": "ค่าที่ใช้ในการลบ EIGRP process (boolean)",  "process\_id\_input": "หมายเลขของ EIGRP process ID ที่ต้องการลบ (list of strings)"  } |

**ตารางที่ 3-29** แสดง Data flow แสดงอุปกรณ์ทั้งหมดในฐานข้อมูล D1

|  |  |
| --- | --- |
| Data Flow ID : | 0029 |
| Data Flow Name : | อุปกรณ์ทั้งหมด |
| Description : | อุปกรณ์ทั้งหมดในฐานข้อมูล D1 |
| Source : | ฐานข้อมูล D1 |
| Destination : | ผู้ใช้ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ทั้งหมด = {  "device\_name": "ชื่อของอุปกรณ์ (string)", "device\_ip": "IP Address ของอุปกรณ์ (string)"  } |

**ตารางที่ 3-30** แสดง Data flow การคืนค่าโรงงาน

|  |  |
| --- | --- |
| Data Flow ID : | 0030 |
| Data Flow Name : | อุปกรณ์ที่ต้องการคืนค่าโรงงาน |
| Description : | อุปกรณ์ที่ต้องการคืนค่าโรงงานโดยรับมาจาก Process 29.1 |
| Source : | Process 29.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ที่ต้องการคืนค่าโรงงาน = {  "device\_index": ดัชนีของอุปกรณ์ในฐานข้อมูล (integer),  "device\_name": ชื่อของอุปกรณ์ (string),  "device\_ip": ที่อยู่ IP ของอุปกรณ์ (string),  "username": ชื่อผู้ใช้งาน SSH สำหรับการเชื่อมต่อ (string),  "password": รหัสผ่าน SSH สำหรับการเชื่อมต่อ (string),  "secret": รหัสผ่าน enable สำหรับการเข้าสู่โหมดพิเศษ (string),  } |

**ตารางที่ 3-31** แสดง Data flow การรีสตาร์ทอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0031 |
| Data Flow Name : | อุปกรณ์ที่ต้องการรีสตาร์ท |
| Description : | อุปกรณ์ที่ต้องการรีสตาร์ทโดยรับมาจาก Process 29.1 |
| Source : | Process 29.1 |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ที่ต้องการรีสตาร์ท = {  "device\_index": ดัชนีของอุปกรณ์ในฐานข้อมูล (integer),  "device\_name": ชื่อของอุปกรณ์ (string),  "device\_ip": ที่อยู่ IP ของอุปกรณ์ (string),  "username": ชื่อผู้ใช้งาน SSH สำหรับการเชื่อมต่อ (string),  "password": รหัสผ่าน SSH สำหรับการเชื่อมต่อ (string),  "secret": รหัสผ่าน enable สำหรับการเข้าสู่โหมดพิเศษ (string),  } |

**ตารางที่ 3-32** แสดง Data flow การตรวจสอบรับคำสั่งในการตรวจสอบรายละเอียดของอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0032 |
| Data Flow Name : | อุปกรณ์ที่ต้องการตรวจสอบการตั้งค่า |
| Description : | อุปกรณ์ที่ต้องการตรวจสอบการตั้งค่า |
| Source : | ผู้ใช้ |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | อุปกรณ์ที่ต้องการตรวจสอบ = {  "device\_name": ชื่อของอุปกรณ์ที่ผู้ใช้เลือก (string),  "selected\_commands": รายการคำสั่งที่ผู้ใช้เลือกเพื่อตรวจสอบ (list of strings),  "device\_info": {  "device\_type": ประเภทของอุปกรณ์ (เช่น "cisco\_ios") (string),  "host": ที่อยู่ IP ของอุปกรณ์ (string),  "username": ชื่อผู้ใช้ในการเข้าถึงอุปกรณ์ (string),  "password": รหัสผ่านในการเข้าถึงอุปกรณ์ (string),  "secret": รหัสผ่าน enable ของอุปกรณ์ (string),  "timeout": ระยะเวลาเชื่อมต่อ (integer)  },  "commands\_to\_execute": {  "show\_running\_config": คำสั่ง "show running-config" (string),  "show\_version": คำสั่ง "show version" (string),  "show\_interfaces": คำสั่ง "show interfaces" (string),  "show\_ip\_interface\_brief": คำสั่ง "show ip interface brief" (string),  "show\_ip\_route": คำสั่ง "show ip route" (string),  "show\_vlan": คำสั่ง "show vlan" (string),  "show\_cdp\_neighbors": คำสั่ง "show cdp neighbors" (string),  "show\_ip\_protocols": คำสั่ง "show ip protocols" (string),  "show\_mac\_address\_table": คำสั่ง "show mac address-table dynamic" (string),  "show\_clock": คำสั่ง "show clock" (string),  "show\_logging": คำสั่ง "show logging" (string),  "show\_interfaces\_trunk": คำสั่ง "show interfaces trunk" (string),  "show\_etherch\_sum": คำสั่ง "show etherch sum" (string),  "show\_lldp\_neighbors": คำสั่ง "show lldp neighbors" (string),  "show\_startup": คำสั่ง "show startup-config" (string),  "show\_interfaces\_status": คำสั่ง "show int status" (string),  "show\_ipv6\_interface\_brief": คำสั่ง "show ipv6 int br" (string),  "show\_flash": คำสั่ง "show flash:" (string),  "show\_dhcp\_pool": คำสั่ง "show ip dhcp pool" (string),  "show\_dhcp\_dinding": คำสั่ง "show ip dhcp binding" (string),  "show\_ntp\_status": คำสั่ง "show ntp status" (string),  "show\_spanning\_tree": คำสั่ง "show spanning-tree" (string),  "show\_spanning\_tree\_sum": คำสั่ง "show spanning-tree sum" (string)  },  "config\_data": ข้อมูลการตั้งค่าที่ได้รับจากการรันคำสั่ง (string),  "error\_message": ข้อความแสดงข้อผิดพลาด (string) [ถ้ามี]  } |

**ตารางที่ 3-33** แสดง Data flow การตั้งค่าการตรวจสอบและบันทึกที่อาจเป็นช่องโหว่ของอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0033 |
| Data Flow Name : | อุปกรณ์ที่ต้องการตรวจสอบช่องโหว่ |
| Description : | อุปกรณ์ที่ต้องการตรวจสอบช่องโหว่ |
| Source : | ระบบอัติโนมัติ |
| Destination : | อุปกรณ์, Process 34.1, ฐานข้อมูล D1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ช่องโหว่ = {  "device\_name": "ชื่อของอุปกรณ์",  "device\_info": {  "device\_type": "ประเภทของอุปกรณ์ (เช่น 'cisco\_ios')",  "host": "ที่อยู่ IP หรือชื่อโฮสต์ของอุปกรณ์",  "username": "ชื่อผู้ใช้สำหรับการเข้าถึงอุปกรณ์",  "password": "รหัสผ่านสำหรับการเข้าถึงอุปกรณ์",  "secret": "รหัสผ่าน enable เพื่อเข้าสู่โหมดพิเศษของอุปกรณ์",  "timeout": "ระยะเวลาเชื่อมต่อกับอุปกรณ์ (เป็นวินาที)"  },  "Process\_34\_1": {  "show\_run": "ผลลัพธ์จากคำสั่ง show running-config",  "show\_ip\_int\_br": "ผลลัพธ์จากคำสั่ง show ip interface brief"  },  "D1": {  "device\_name": "ชื่อของอุปกรณ์",  "analysis": {  "warnings": [  "คำเตือนที่พบจากการตรวจสอบช่องโหว่"  ],  "last\_updated": "เวลาที่ทำการอัปเดตผลการตรวจสอบ (รูปแบบ 'YYYY-MM-DD HH:MM:SS')"  }  }  } |

**ตารางที่ 3-34** แสดง Data flow การตรวจสอบการตั้งค่าที่อาจจะเป็นช่องโหว่

|  |  |
| --- | --- |
| Data Flow ID : | 0034 |
| Data Flow Name : | ผลลัพธ์อุปกรณ์ที่ต้องการตรวจสอบช่องโหว่ |
| Description : | ค่า configuration ของอุปกรณ์ที่ต้องการตรวจสอบ |
| Source : | Process 33.1 |
| Destination : | Process 33.1 |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ผลลัพธ์การตรวจสอบ = {  "device\_name": "ชื่อของอุปกรณ์",  "analysis": {  "warnings": [  "คำเตือนที่พบจากการตรวจสอบช่องโหว่"  ],  "last\_updated": "เวลาที่ทำการอัปเดตผลการตรวจสอบ (รูปแบบ 'YYYY-MM-DD HH:MM:SS')"  }  } |

|  |  |
| --- | --- |
| Data Flow ID : | 0035 |
| Data Flow Name : | ผลลัพธ์อุปกรณ์ที่มีช่องโหว่ |
| Description : | ผลลัพธ์อุปกรณ์ที่มีช่องโหว่ในฐานข้อมูล D1 |
| Source : | ฐานข้อมูล D1 |
| Destination : | ผู้ใช้ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | ช่องโหว่ของอุปกรณ์ = {  "device\_name": "ชื่อของอุปกรณ์", "analysis": { "warnings": [ "คำเตือนที่พบจากการตรวจสอบช่องโหว่" ], "last\_updated": "เวลาที่ทำการอัปเดตผลการตรวจสอบ (รูปแบบ 'YYYY-MM-DD HH:MM:SS')" }  } |

**ตารางที่ 3-35** แสดง Data flow แสดงการตรวจสอบที่อาจจะเป็นช่องโหว่ของอุปกรณ์

**ตารางที่ 3-36** แสดง Data flow การตรวจสอบรายละเอียดของอุปกรณ์

|  |  |
| --- | --- |
| Data Flow ID : | 0036 |
| Data Flow Name : | อุปกรณ์ที่ต้องการตรวจสอบรายละเอียด |
| Description : | อุปกรณ์ที่ต้องการตรวจสอบรายละเอียดผ่านโพรโทคอล SNMP |
| Source : | ผู้ใช้ |
| Destination : | อุปกรณ์ |
| Type Of Data Flow : | หน้าจอ |
| Data Structure : | รายละเอียดของอุปกรณ์ = {  "device\_ip": "ที่อยู่ IP ของอุปกรณ์",  "device\_info": {  "device\_name": "ชื่อของอุปกรณ์",  "device\_type": "ประเภทของอุปกรณ์ (เช่น 'Switch Layer3', 'Switch Layer2', 'Router')",  "uptime": "ข้อมูลเวลาทำงานของอุปกรณ์",  "location": "สถานที่ตั้งของอุปกรณ์",  "contact": "ข้อมูลติดต่อของผู้ดูแลอุปกรณ์",  "description": "คำอธิบายของระบบอุปกรณ์",  "sysname": "ชื่อของระบบอุปกรณ์",  "ports": [  {  "port\_id": "ชื่อพอร์ตหรือหมายเลขพอร์ต",  "port\_description": "คำอธิบายของพอร์ต"  }  ]  }  } |

* 1. **ส่วนการจัดเก็บข้อมูล (Data Store)**

**ตารางที่ 3-37** แสดง Data Store D1 ข้อมูลอุปกรณ์

|  |  |
| --- | --- |
| Data Store ID : | D1 |
| Data Store Name : | รายละเอียดอุปกรณ์ |
| Description : | เป็นแฟ้มที่เก็บรายละเอียดข้อมูลอุปกรณ์งานทั้งหมด |
| File Type : | แฟ้มข้อมูลทางคอมพิวเตอร์ |
| Number of Record : | จำนวนอุปกรณ์ที่มีอยู่ในฐานข้อมูล |
| Data Structrue : | {  "device\_name": "ชื่อของอุปกรณ์ (string)",  "device\_info": {  "device\_type": "ประเภทของอุปกรณ์ (เช่น 'cisco\_ios') (string)",  "ip": "ที่อยู่ IP ของอุปกรณ์ (string)",  "username": "ชื่อผู้ใช้งาน SSH (string)",  "password": "รหัสผ่าน SSH (string)",  "secret": "รหัสผ่าน enable (string)",  "session\_log": "ไฟล์บันทึกการเชื่อมต่อ (string)"  },  "analysis": {  "warnings": [  "คำเตือนที่พบจากการตรวจสอบช่องโหว่ (string)"  ],  "last\_updated": "เวลาที่ทำการอัปเดต (string)"  }  } |

* 1. **การกําหนดลักษณะประมวลผลข้อมูลในระบบ (Process Specification)**

**ตารางที่ 3-38** แสดง Process Specification 1.1 รับข้อมูลเริ่มต้นจากผู้ใช้งาน

|  |  |
| --- | --- |
| Process Number : | 1.1 |
| Process Name: | รับข้อมูลจากผู้ใช้งาน |
| Description : | กระบวนการนี้ทำหน้าที่รับข้อมูลเริ่มต้นจากผู้ใช้งานเกี่ยวกับอุปกรณ์ เช่น ชื่อโฮสต์, ที่อยู่ IP, รหัสผ่าน SSH, และข้อมูลที่เกี่ยวข้อง จากนั้นตรวจสอบความถูกต้องของข้อมูล (เช่น ตรวจสอบว่า IP หรือ Hostname ไม่ซ้ำกับอุปกรณ์ที่มีอยู่ในฐานข้อมูล) และบันทึกข้อมูลเหล่านี้ลงในฐานข้อมูล (D1) เพื่อใช้ในกระบวนการต่อไป |
| Input Data Flow : | ข้อมูลที่รับมาจากฟอร์มที่ผู้ใช้กรอก เช่น consoleport, hostname, domainname, privilege\_password, ssh\_username, ssh\_password, interface, interface\_type, ip\_address, save\_startup |
| Output Data Flow : | ข้อมูลที่ได้รับจากผู้ใช้งาน พร้อมรายละเอียดอุปกรณ์ที่บันทึกลงในฐานข้อมูล D1 |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะรับข้อมูลจากผู้ใช้งาน ตรวจสอบความถูกต้อง เช่น IP และ Hostname ไม่ซ้ำกับอุปกรณ์ที่มีในฐานข้อมูล และบันทึกข้อมูลลงในฐานข้อมูล D1 หากข้อมูลถูกต้อง |

**ตารางที่ 3-39** แสดง Process Specification 2.1 กำหนดค่าอุปกรณ์เริ่มต้น

|  |  |
| --- | --- |
| Process Number : | 2.1 |
| Process Name: | การตั้งค่าอุปกรณ์ผ่านการเชื่อมต่อ Serial |
| Description : | กระบวนการนี้จะทำหน้าที่ในการเชื่อมต่อกับอุปกรณ์ผ่านพอร์ตคอนโซล (console port) และตั้งค่าต่างๆ ตามที่ผู้ใช้งานระบุ เช่น ชื่อโฮสต์, ชื่อโดเมน, รหัสผ่าน, การตั้งค่าการเข้าถึง SSH และ IP Address รวมถึงการตั้งค่าการเชื่อมต่อผ่าน DHCP หรือแบบ Static พร้อมทั้งเปิดใช้งานอินเทอร์เฟซ (interface) และบันทึกการตั้งค่าไปยังหน่วยความจำ หากผู้ใช้งานเลือกตัวเลือกการบันทึกการตั้งค่า |
| Input Data Flow : | ข้อมูลการตั้งค่าอุปกรณ์ที่ได้รับจาก Process 1.1 |
| Output Data Flow : | การตั้งค่าที่ส่งไปยังอุปกรณ์ และผลลัพธ์จากการตั้งค่า |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะเริ่มต้นด้วยการเชื่อมต่ออุปกรณ์ผ่านพอร์ตคอนโซล จากนั้นจะตรวจสอบว่าค่าของ IP Address เป็นแบบ DHCP หรือ Static หากเป็นแบบ Static จะทำการแปลง CIDR เป็น Subnet Mask (กระบวนการ 2.2) และส่งคำสั่งการตั้งค่าไปยังอุปกรณ์ เช่น ตั้งชื่อโฮสต์, ตั้งชื่อโดเมน, ตั้งค่า SSH และอินเทอร์เฟซ หลังจากนั้นจะทำการบันทึกการตั้งค่าลงในหน่วยความจำของอุปกรณ์ หากผู้ใช้งานเลือกตัวเลือกบันทึกการตั้งค่า |

**ตารางที่ 3-40** แสดง Process Specification 2.2 กำหนดค่าอุปกรณ์เริ่มต้น

|  |  |
| --- | --- |
| Process Number : | 2.2 |
| Process Name: | การแปลง CIDR เป็น Subnet Mask |
| Description : | กระบวนการนี้จะทำการแปลงค่า CIDR ที่ได้รับจากผู้ใช้งาน (เช่น 192.168.1.0/24) ให้เป็น Subnet Mask (เช่น 255.255.255.0) เพื่อใช้ในการตั้งค่า IP Address บนอุปกรณ์ กระบวนการนี้จะคำนวณค่าของ Subnet Mask จากจำนวนบิต CIDR ที่ได้รับมา |
| Input Data Flow : | ข้อมูล CIDR ที่ได้รับจาก Process 2.1 เช่น 192.168.1.0/24 |
| Output Data Flow : | ค่า Subnet Mask ที่แปลงจาก CIDR (เช่น 255.255.255.0) |
| Process Type : | Calculation/Conversion |
| Process Logic : | กระบวนการนี้จะรับค่า CIDR ที่ผู้ใช้งานกรอกในฟอร์ม จากนั้นจะแยกส่วนของ IP และจำนวนบิต CIDR เพื่อคำนวณ Subnet Mask โดยใช้หลักการในการแปลง CIDR เช่น /24 จะได้ 255.255.255.0 เมื่อแปลงเสร็จแล้วค่า Subnet Mask จะถูกส่งกลับไปเพื่อใช้ในการตั้งค่า IP Address ในกระบวนการ 2.1 |

**ตารางที่ 3-41** แสดง Process Specification 3.1 เพิ่มอุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 3.1 |
| Process Name: | การจัดการข้อมูลอุปกรณ์ |
| Description : | กระบวนการนี้รับข้อมูลจากผู้ใช้งานเพื่อเพิ่มข้อมูลอุปกรณ์ใหม่ลงในฐานข้อมูล หากข้อมูลที่รับมาซ้ำกับข้อมูลที่มีอยู่ในฐานข้อมูล (เช่น ชื่อโฮสต์หรือที่อยู่ IP) จะทำการแจ้งเตือนให้ผู้ใช้งานเลือกข้อมูลใหม่ กระบวนการนี้ทำหน้าที่เพิ่มข้อมูลอุปกรณ์ลงในฐานข้อมูลอย่างถูกต้องและปลอดภัย |
| Input Data Flow : | ข้อมูลที่ผู้ใช้งานกรอกในฟอร์ม เช่น ชื่ออุปกรณ์ (name), ที่อยู่ IP (ip\_address), ชื่อผู้ใช้งาน SSH (ssh\_username), รหัสผ่าน SSH (ssh\_password), รหัสผ่านสิทธิพิเศษ (privilege\_password) |
| Output Data Flow : | ข้อมูลอุปกรณ์ที่ได้รับการบันทึกในฐานข้อมูล (D1) หลังจากการตรวจสอบข้อมูลแล้ว |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะรับข้อมูลจากผู้ใช้งานที่กรอกในฟอร์ม เช่น ชื่ออุปกรณ์, IP Address, รหัสผ่าน SSH และรหัสผ่านสิทธิพิเศษ จากนั้นจะทำการตรวจสอบว่าอุปกรณ์ที่มีชื่อโฮสต์หรือ IP Address ซ้ำกับข้อมูลที่มีอยู่ในฐานข้อมูลหรือไม่ หากพบข้อมูลซ้ำ จะมีการแจ้งเตือนผู้ใช้งานให้เลือกข้อมูลใหม่ หากไม่มีการซ้ำข้อมูล กระบวนการจะทำการบันทึกข้อมูลอุปกรณ์ลงในฐานข้อมูล และแสดงข้อความยืนยันความสำเร็จ |

**ตารางที่ 3-42** แสดง Process Specification 4.1 เเสดงรายละเอียดของข้อมูลอุปกรณ์ที่มีในระบบทั้งหมด

|  |  |
| --- | --- |
| Process Number : | 4.1 |
| Process Name: | การแสดงข้อมูลอุปกรณ์ทั้งหมด |
| Description : | กระบวนการนี้ทำหน้าที่ดึงข้อมูลอุปกรณ์ทั้งหมดจากฐานข้อมูลและแสดงผลให้ผู้ใช้งานเห็นในรูปแบบตาราง โดยแสดงชื่ออุปกรณ์, ที่อยู่ IP, ชื่อผู้ใช้งาน SSH, รหัสผ่าน SSH, และรหัสผ่านสิทธิพิเศษ พร้อมกับปุ่มสำหรับจัดการอุปกรณ์ เช่น การลบ, แก้ไข, หรือ Ping อุปกรณ์ |
| Input Data Flow : | ข้อมูลที่ดึงจากฐานข้อมูล เช่น ชื่ออุปกรณ์, ที่อยู่ IP, ชื่อผู้ใช้งาน SSH, รหัสผ่าน SSH, รหัสผ่านสิทธิพิเศษ |
| Output Data Flow : | รายละเอียดอุปกรณ์ทั้งหมดในฐานข้อมูล |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้ดึงข้อมูลทั้งหมดของอุปกรณ์จากฐานข้อมูล (D1) มาแสดงบนหน้าเว็บไซต์ในรูปแบบตาราง ซึ่งรวมถึงชื่ออุปกรณ์, IP Address, SSH Username, SSH Password, Secret Password และแสดงปุ่มการจัดการสำหรับการลบ, แก้ไข, และ Ping อุปกรณ์ โดยจะมีการแสดงข้อความเมื่อไม่มีข้อมูลในระบบ |

**ตารางที่ 3-43** แสดง Process Specification 5.1 ลบอุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 5.1 |
| Process Name: | การลบอุปกรณ์ |
| Description : | กระบวนการนี้ทำหน้าที่รับคำขอลบอุปกรณ์จากผู้ใช้งาน โดยตรวจสอบที่อยู่ IP ของอุปกรณ์ที่จะลบจากฟอร์มที่ผู้ใช้กรอก หากพบอุปกรณ์ในฐานข้อมูลที่ตรงกับที่อยู่ IP ที่ระบุ ระบบจะทำการลบอุปกรณ์นั้นออกจากฐานข้อมูล |
| Input Data Flow : | ที่อยู่ IP ของอุปกรณ์ที่ต้องการลบ |
| Output Data Flow : | ข้อมูลที่ได้รับการอัปเดตในฐานข้อมูลหลังจากการลบอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะรับคำขอลบจากฟอร์มที่ผู้ใช้กรอก ซึ่งจะระบุที่อยู่ IP ของอุปกรณ์ที่ต้องการลบ จากนั้นทำการค้นหาในฐานข้อมูลโดยใช้ที่อยู่ IP ที่ได้รับ หากพบข้อมูลอุปกรณ์นั้นในฐานข้อมูล ก็จะทำการลบข้อมูลออกและแสดงหน้าเว็บที่มีข้อมูลอุปกรณ์ที่อัปเดตแล้ว |

**ตารางที่ 3-44** แสดง Process Specification 6.1 แก้ไขข้อมูล

|  |  |
| --- | --- |
| Process Number : | 6.1 |
| Process Name: | การแสดงข้อมูลอุปกรณ์ที่ต้องการแก้ไข |
| Description : | กระบวนการนี้ทำหน้าที่ในการแสดงข้อมูลของอุปกรณ์ที่ผู้ใช้ต้องการแก้ไข โดยอ้างอิงจากที่อยู่ IP ของอุปกรณ์ เมื่อผู้ใช้ต้องการแก้ไขข้อมูล เช่น ชื่อโฮสต์ ที่อยู่ IP รหัสผ่าน SSH ข้อมูลเหล่านี้จะถูกดึงมาจากฐานข้อมูลและแสดงให้ผู้ใช้ดูในฟอร์มที่ให้กรอกข้อมูลใหม่ |
| Input Data Flow : | ข้อมูลอุปกรณ์ที่ผู้ใช้เลือกแก้ไข (โดยระบุจากที่อยู่ IP ของอุปกรณ์) |
| Out put Data Flow : | ฟอร์มที่แสดงข้อมูลของอุปกรณ์ให้ผู้ใช้กรอกข้อมูลใหม่ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะค้นหาอุปกรณ์ในฐานข้อมูลจากที่อยู่ IP หากพบอุปกรณ์ ระบบจะแสดงข้อมูลของอุปกรณ์ในฟอร์มแก้ไขให้ผู้ใช้สามารถแก้ไขข้อมูลใหม่ได้ |

**ตารางที่ 3-45** แสดง Process Specification 7.1 บันทึกการแก้ไข

|  |  |
| --- | --- |
| Process Number : | 7.1 |
| Process Name: | การอัปเดตข้อมูลอุปกรณ์ |
| Description : | กระบวนการนี้ทำหน้าที่ในการอัปเดตข้อมูลอุปกรณ์ในฐานข้อมูล โดยจะตรวจสอบข้อมูลที่ผู้ใช้กรอกในฟอร์มแก้ไข และทำการอัปเดตหรือแก้ไขข้อมูลในฐานข้อมูลให้ตรงกับที่ผู้ใช้ได้กรอกใหม่ |
| Input Data Flow : | ข้อมูลที่ผู้ใช้กรอกในฟอร์มแก้ไขจาก Process 6.1 |
| Output Data Flow : | ข้อมูลอุปกรณ์ที่ได้รับการอัปเดตในฐานข้อมูล |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะตรวจสอบว่า IP หรือชื่อโฮสต์ที่ผู้ใช้กรอกใหม่มีความซ้ำกับข้อมูลในฐานข้อมูลหรือไม่ หากพบการซ้ำ ระบบจะให้ผู้ใช้เลือกแก้ไขข้อมูลหรือให้คำเตือน จากนั้นจะทำการอัปเดตข้อมูลในฐานข้อมูล |

**ตารางที่ 3-46** แสดง Process Specification 8.1 Ping อุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 8.1 |
| Process Name: | การตรวจสอบสถานะอุปกรณ์ด้วย Ping |
| Description : | กระบวนการนี้ทำหน้าที่ในการตรวจสอบสถานะการเชื่อมต่อของอุปกรณ์ที่ระบุ โดยใช้คำสั่ง ping ไปยังที่อยู่ IP ของอุปกรณ์นั้น ๆ เพื่อตรวจสอบว่าการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์นั้นสำเร็จหรือไม่ หาก ping สำเร็จจะแสดงผลลัพธ์เป็นข้อความว่าการเชื่อมต่อสำเร็จและแสดงรายละเอียดผลการทดสอบ หาก ping ล้มเหลวจะแสดงผลลัพธ์ที่บ่งชี้ว่าไม่สามารถเชื่อมต่อได้ |
| Input Data Flow : | ที่อยู่ IP ของอุปกรณ์ที่ต้องการตรวจสอบสถานะการเชื่อมต่อ |
| Output Data Flow : | ข้อความผลลัพธ์จากการทำ ping ซึ่งประกอบด้วยข้อมูลว่าการเชื่อมต่อสำเร็จหรือไม่และรายละเอียดเพิ่มเติมจากการทดสอบ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้รับคำขอ ping จากผู้ใช้โดยส่งที่อยู่ IP ไปยังระบบ หากการเชื่อมต่อสำเร็จ ระบบจะแสดงข้อความที่บ่งบอกถึงผลสำเร็จ พร้อมรายละเอียดผลลัพธ์ของคำสั่ง ping หากล้มเหลวจะแสดงข้อความข้อผิดพลาดที่อธิบายว่าไม่สามารถเชื่อมต่อได้ |

**ตารางที่ 3-47** แสดง Process Specification 9.1 รับข้อมูลการตั้งค่าพื้นฐาน (Basic Settings)

|  |  |
| --- | --- |
| Process Number : | 9.1 |
| Process Name: | การตรวจสอบและเตรียมข้อมูลการตั้งค่าพื้นฐาน |
| Description : | กระบวนการนี้ทำหน้าที่รับข้อมูลการตั้งค่าจากผู้ใช้ เช่น hostname, รหัสผ่าน, banner และตัวเลือกการตั้งค่าความปลอดภัย จากนั้นตรวจสอบข้อมูลที่ได้รับกับฐานข้อมูลเพื่อป้องกันความซ้ำซ้อน เช่น ชื่อโฮสต์หรือ IP Address พร้อมตรวจสอบว่าข้อมูลอุปกรณ์ที่ระบุมีอยู่ในระบบหรือไม่ หากพบข้อผิดพลาด เช่น อุปกรณ์ไม่พบในฐานข้อมูลหรือ IP ซ้ำ ระบบจะแจ้งเตือนผู้ใช้ให้แก้ไขข้อมูลก่อนดำเนินการต่อ และหากข้อมูลถูกต้อง ระบบจะรวบรวมข้อมูลอุปกรณ์ที่ต้องการตั้งค่าเพื่อส่งต่อไปยังกระบวนการตั้งค่า (Process 10.1) |
| Input Data Flow : | ค่าการตั้งค่าที่ผู้ใช้ระบุ เช่น hostname, IP Address ของอุปกรณ์, รหัสผ่าน SSH, และตัวเลือกการตั้งค่าความปลอดภัย |
| Output Data Flow : | รายการ IP Address ของอุปกรณ์ที่ผ่านการตรวจสอบ พร้อมข้อมูลการตั้งค่าที่พร้อมส่งต่อไปยังกระบวนการตั้งค่า (Process 10.1) |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้ตรวจสอบข้อมูลการตั้งค่าจากผู้ใช้ โดยเปรียบเทียบกับฐานข้อมูลเพื่อป้องกันความซ้ำซ้อนหรือข้อผิดพลาด เช่น การใช้ hostname หรือ IP Address ซ้ำ ตรวจสอบการเชื่อมโยงระหว่างข้อมูลอุปกรณ์ที่ระบุและค่าการตั้งค่าที่ต้องการ พร้อมรวบรวมรายการอุปกรณ์ที่ต้องการตั้งค่าเพื่อนำส่งไปยังกระบวนการถัดไปสำหรับการดำเนินการตั้งค่าจริง |

**ตารางที่ 3-48** แสดง Process Specification 10.1 กำหนดค่าพื้นฐาน

|  |  |
| --- | --- |
| Process Number : | 10.1 |
| Process Name: | การตั้งค่าพื้นฐานบนอุปกรณ์เครือข่าย |
| Description : | กระบวนการนี้ทำหน้าที่รับข้อมูลการตั้งค่าพื้นฐานจากกระบวนการก่อนหน้า (Process 9.1) เช่น hostname, รหัสผ่าน, banner, และตัวเลือกการตั้งค่าความปลอดภัย แล้วดำเนินการเชื่อมต่อกับอุปกรณ์เครือข่ายโดยใช้โปรโตคอล SSH เพื่อส่งคำสั่งการตั้งค่าไปยังอุปกรณ์ โดยระบบจะตั้งค่าตามข้อมูลที่ได้รับ เช่น การเปลี่ยน hostname การตั้งค่า enable secret การเพิ่ม banner ข้อความ การตั้งค่าการเข้ารหัสรหัสผ่าน และการเพิ่มผู้ใช้ใหม่ พร้อมทั้งอัปเดตสถานะในฐานข้อมูลเมื่อการตั้งค่าสำเร็จ |
| Input Data Flow : | ข้อมูลการตั้งค่าจาก Process 9.1 ซึ่งรวมถึง hostname, รหัสผ่าน, banner, และรายการอุปกรณ์ที่ต้องตั้งค่า |
| Output Data Flow : | สถานะการตั้งค่าของอุปกรณ์แต่ละตัว เช่น สำเร็จหรือล้มเหลว |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เชื่อมต่อกับอุปกรณ์เครือข่ายที่ได้รับผ่าน SSH ตรวจสอบสิทธิ์การเข้าถึง จากนั้นส่งคำสั่งการตั้งค่าต่าง ๆ ไปยังอุปกรณ์ หากการตั้งค่าผ่านสำเร็จ ระบบจะแสดงผลลัพธ์เป็นข้อความยืนยันและอัปเดตข้อมูลในฐานข้อมูล หากล้มเหลว ระบบจะแสดงข้อความข้อผิดพลาดเพื่อให้ผู้ใช้ทราบถึงปัญหา เช่น การเชื่อมต่อล้มเหลวหรือคำสั่งผิดพลาด |

**ตารางที่ 3-49** แสดง Process Specification 11.1 รับข้อมูลการตั้งค่า Network Interface

|  |  |
| --- | --- |
| Process Number : | 11.1 |
| Process Name: | การตรวจสอบและเตรียมข้อมูลการตั้งค่า Network Interface |
| Description : | กระบวนการนี้ทำหน้าที่รับข้อมูลการตั้งค่าการเชื่อมต่อ เช่น รายชื่ออุปกรณ์, การตั้งค่า IPv4/IPv6, และ Duplex จากผู้ใช้ โดยทำการตรวจสอบข้อมูลกับฐานข้อมูลเพื่อป้องกันข้อผิดพลาด เช่น IP ซ้ำ หรืออุปกรณ์ที่ไม่พบในระบบ หากพบข้อผิดพลาด ระบบจะแจ้งเตือนให้ผู้ใช้แก้ไขข้อมูลก่อนดำเนินการต่อ และหากข้อมูลถูกต้อง จะส่งข้อมูลที่ผ่านการตรวจสอบไปยังกระบวนการตั้งค่าจริง (Process 12.1) |
| Input Data Flow : | รายชื่ออุปกรณ์ (device\_name หรือ many\_hostname), การตั้งค่า IPv4 เช่น Interface, IP Address, Subnet Mask, DHCP, การตั้งค่า IPv6 เช่น Interface, IP Address, DHCP, การตั้งค่า Duplex และความเร็ว |
| Output Data Flow : | รายการ IP Address ของอุปกรณ์ที่ผ่านการตรวจสอบพร้อมข้อมูลการตั้งค่าที่พร้อมส่งต่อไปยังกระบวนการตั้งค่า (Process 12.1) หรือข้อความแจ้งเตือนหากพบข้อผิดพลาด เช่น IP Address ซ้ำ หรืออุปกรณ์ไม่พบในระบบ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะรับค่าการตั้งค่าจากผู้ใช้ผ่านฟอร์ม จากนั้นทำการตรวจสอบความถูกต้องของข้อมูล เช่น การมีอยู่ของอุปกรณ์ในฐานข้อมูล และป้องกัน IP Address ซ้ำ หากพบข้อผิดพลาด เช่น "Duplicate IP detected" หรือ "Device not found" ระบบจะแจ้งเตือนผู้ใช้เพื่อแก้ไขข้อมูล หากข้อมูลถูกต้อง ระบบจะรวบรวมรายการอุปกรณ์ที่ต้องตั้งค่าพร้อมข้อมูลทั้งหมด แล้วส่งต่อไปยังกระบวนการตั้งค่าจริง (Process 12.1) |

**ตารางที่ 3-50** แสดง Process Specification 12.1 กำหนด Network Interface

|  |  |
| --- | --- |
| Process Number : | 12.1 |
| Process Name: | การตั้งค่า Network Interface บนอุปกรณ์เครือข่าย |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่าที่ผ่านการตรวจสอบจาก Process 11.1 เช่น การตั้งค่า IPv4/IPv6 และ Duplex จากนั้นดำเนินการตั้งค่าบนอุปกรณ์เครือข่ายผ่านคำสั่ง CLI โดยใช้ SSH หากการตั้งค่าสำเร็จ ระบบจะแสดงข้อความยืนยันและอัปเดตสถานะในฐานข้อมูล |
| Input Data Flow : | ข้อมูลการตั้งค่าที่ตรวจสอบแล้วจาก Process 11.1 เช่น IPv4/IPv6, Duplex และรายการอุปกรณ์ที่ต้องตั้งค่า |
| Output Data Flow : | สถานะการตั้งค่าของอุปกรณ์แต่ละตัว เช่น สำเร็จหรือล้มเหลว |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เชื่อมต่อกับอุปกรณ์ผ่าน SSH โดยใช้ข้อมูลที่ได้รับ จากนั้นดำเนินการตั้งค่าตามข้อมูล เช่น การกำหนด IP Address, การเปิด/ปิด Interface และการตั้งค่า Duplex หากการตั้งค่าสำเร็จ ระบบจะแสดงข้อความยืนยัน เช่น "IPv4 Config for [device\_name] completed" และอัปเดตสถานะในฐานข้อมูล หากการตั้งค่าล้มเหลว เช่น การเชื่อมต่อล้มเหลวหรือคำสั่งผิดพลาด ระบบจะแจ้งเตือนผู้ใช้พร้อมแสดงรายละเอียดปัญหา |

**ตารางที่ 3-51** แสดง Process Specification 13.1 รับข้อมูลการตั้งค่า VLAN

|  |  |
| --- | --- |
| Process Number : | 13.1 |
| Process Name: | การรับข้อมูลการตั้งค่า VLAN |
| Description : | กระบวนการนี้จะรับข้อมูลการตั้งค่าจากผู้ใช้ เช่น หมายเลข VLAN ที่ต้องการเพิ่ม ลบ หรือแก้ไข ชื่อ VLAN การตั้งค่า Access VLAN หรือ Trunk VLAN และตัวเลือกอื่น ๆ ที่เกี่ยวข้อง ข้อมูลที่ป้อนเข้ามาจะถูกตรวจสอบความถูกต้อง เช่น การตรวจสอบรูปแบบหมายเลข VLAN และการซ้ำซ้อนของข้อมูล หากมีข้อผิดพลาด ระบบจะแจ้งเตือนผู้ใช้ให้แก้ไขข้อมูลก่อนที่จะส่งต่อไปยังขั้นตอนถัดไป |
| Input Data Flow : | ข้อมูลการตั้งค่า VLAN ที่ผู้ใช้ระบุ เช่น หมายเลข VLAN, ชื่อ VLAN, รายชื่ออุปกรณ์, และการตั้งค่าที่เกี่ยวข้อง |
| Output Data Flow : | ข้อมูลการตั้งค่าที่ตรวจสอบแล้ว พร้อมส่งต่อให้กระบวนการ 14.1 เพื่อดำเนินการตั้งค่าจริงบนอุปกรณ์เครือข่าย |
| Process Type : | Online Processing |
| Process Logic : | ระบบจะตรวจสอบข้อมูลการตั้งค่าที่ป้อนเข้ามาโดยผู้ใช้ หากพบความผิดพลาด เช่น หมายเลข VLAN ไม่ถูกต้องหรือซ้ำซ้อน ระบบจะแจ้งเตือนผู้ใช้ด้วยข้อความ เช่น "Invalid VLAN ID: [vlan\_id]" หลังจากข้อมูลทั้งหมดถูกตรวจสอบแล้ว ระบบจะจัดเตรียมข้อมูลสำหรับการตั้งค่าจริงในกระบวนการถัดไป (process 14.1) |

**ตารางที่ 3-52** แสดง Process Specification 14.1 กำหนดค่า VLAN

|  |  |
| --- | --- |
| Process Number : | 14.1 |
| Process Name: | การตั้งค่า VLAN บนอุปกรณ์เครือข่าย |
| Description : | กระบวนการนี้จะนำข้อมูลที่ผ่านการตรวจสอบจาก Process 13.1 มาดำเนินการตั้งค่าบนอุปกรณ์เครือข่ายโดยใช้คำสั่ง CLI ผ่าน SSH ซึ่งครอบคลุมการเพิ่ม VLAN, ลบ VLAN, แก้ไขชื่อ VLAN การตั้งค่า Access VLAN และ Trunk VLAN รวมถึงการเปิดหรือปิด VLAN ตามคำสั่งที่กำหนด |
| Input Data Flow : | ข้อมูลการตั้งค่าที่ผ่านการตรวจสอบจาก Process 13.1 เช่น หมายเลข VLAN, ชื่อ VLAN, รายชื่ออุปกรณ์, และการตั้งค่าที่เกี่ยวข้อง |
| Output Data Flow : | สถานะการตั้งค่าของอุปกรณ์ เช่น สำเร็จหรือล้มเหลว |
| Process Type : | Online Processing |
| Process Logic : | ระบบจะเชื่อมต่อกับอุปกรณ์เครือข่ายผ่าน SSH และส่งคำสั่ง CLI สำหรับการตั้งค่าที่ได้รับ เช่น คำสั่งเพิ่ม VLAN, ลบ VLAN, หรือการตั้งค่า Access และ Trunk Mode หากการตั้งค่าสำเร็จ ระบบจะแสดงข้อความยืนยัน เช่น "VLAN [vlan\_id] created successfully on [device\_name]" พร้อมอัปเดตสถานะในฐานข้อมูล หากเกิดข้อผิดพลาด เช่น การเชื่อมต่อล้มเหลวหรือคำสั่งผิดพลาด ระบบจะแจ้งเตือนผู้ใช้และบันทึกรายละเอียดปัญหาในฐานข้อมูล |

**ตารางที่ 3-53** แสดง Process Specification 15.1 รับข้อมูลการตั้งค่า Management

|  |  |
| --- | --- |
| Process Number : | 15.1 |
| Process Name: | การรับข้อมูลการตั้งค่า Management |
| Description : | กระบวนการนี้รับข้อมูลที่ผู้ใช้ระบุสำหรับการตั้งค่า Management ของอุปกรณ์เครือข่าย ข้อมูลเหล่านี้รวมถึงการตั้งค่า VTY, Console, DHCP, NTP, Time Zone, SNMP, CDP และ LLDP โดยข้อมูลจะถูกตรวจสอบและเตรียมพร้อมสำหรับกระบวนการตั้งค่า |
| Input Data Flow : | ข้อมูลการตั้งค่าที่ได้รับจากผู้ใช้ เช่น ชื่ออุปกรณ์, รหัสผ่าน VTY และ Console, ข้อมูล DHCP, NTP Server, Time Zone, และการตั้งค่า SNMP |
| Output Data Flow : | ข้อมูลที่ตรวจสอบแล้วพร้อมสำหรับกระบวนการ 16.1 เพื่อดำเนินการตั้งค่าบนอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | ระบบรับข้อมูลจากฟอร์มที่ผู้ใช้กรอกบนหน้าเว็บและตรวจสอบข้อมูลดังกล่าว หากพบความผิดพลาด เช่น ไม่มีอุปกรณ์ที่ระบุอยู่ในฐานข้อมูล หรือค่าที่กรอกไม่ถูกต้อง ระบบจะแจ้งเตือนผู้ใช้ผ่านหน้าต่างข้อความ หลังจากตรวจสอบสำเร็จ ระบบจะจัดเตรียมข้อมูลสำหรับการตั้งค่าจริง |

**ตารางที่ 3-54** แสดง Process Specification 16.1 การกำหนดค่า Management

|  |  |
| --- | --- |
| Process Number : | 16.1 |
| Process Name: | การตั้งค่า Management บนอุปกรณ์เครือข่าย |
| Description : | กระบวนการนี้ใช้ข้อมูลที่ผ่านการตรวจสอบจาก Process 15.1 เพื่อดำเนินการตั้งค่าจริงบนอุปกรณ์เครือข่ายผ่าน SSH เช่น การกำหนดรหัสผ่าน VTY, Console, การตั้งค่า DHCP, NTP, Time Zone, SNMP, รวมถึง CDP และ LLDP |
| Input Data Flow : | ข้อมูลที่ตรวจสอบแล้วจาก Process 15.1 เช่น รหัสผ่าน, ข้อมูล DHCP, การตั้งค่า SNMP และการเปิด/ปิด CDP และ LLDP |
| Output Data Flow : | สถานะการตั้งค่าของอุปกรณ์ เช่น สำเร็จหรือล้มเหลว |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เริ่มต้นด้วยการเชื่อมต่ออุปกรณ์เครือข่ายผ่าน SSH จากนั้นดำเนินการตั้งค่าต่าง ๆ โดยใช้คำสั่ง CLI เช่น การตั้งรหัสผ่าน VTY และ Console, การตั้งค่าด้วย DHCP Pool, การตั้งค่า SNMP และ NTP หากการตั้งค่าประสบความสำเร็จ ระบบจะแสดงข้อความยืนยันพร้อมอัปเดตสถานะในฐานข้อมูล หากเกิดข้อผิดพลาด |

**ตารางที่ 3-55** แสดง Process Specification 17.1 รับข้อมูลการตั้งค่า Spanning Tree

|  |  |
| --- | --- |
| Process Number : | 17.1 |
| Process Name: | การรับข้อมูลการตั้งค่า Spanning Tree |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่าที่ผู้ใช้ระบุสำหรับการตั้งค่า Spanning Tree Protocol (STP) ของอุปกรณ์เครือข่าย ข้อมูลเหล่านี้รวมถึงโหมด STP, การตั้งค่า Root Primary และ Secondary, การเปิดหรือปิด PortFast และการตั้งค่าพอร์ตที่ใช้ PortFast |
| Input Data Flow : | ข้อมูลการตั้งค่า STP ที่ผู้ใช้กรอก เช่น โหมด STP, การตั้งค่า Root Primary, Secondary, การเปิด/ปิด PortFast และการเลือกพอร์ตที่ใช้ PortFast |
| Output Data Flow : | ข้อมูลการตั้งค่าที่พร้อมสำหรับการตั้งค่า Spanning Tree ไปยัง Process 18.1 |
| Process Type : | Online Processing |
| Process Logic : | ระบบรับข้อมูลจากฟอร์มที่ผู้ใช้กรอกบนหน้าเว็บและตรวจสอบข้อมูล หากพบความผิดพลาด เช่น อุปกรณ์ไม่พบในฐานข้อมูล หรือค่าที่กรอกไม่ถูกต้อง ระบบจะแจ้งเตือนผู้ใช้ผ่านหน้าต่างข้อความ หลังจากตรวจสอบข้อมูลเสร็จสิ้น ระบบจะจัดเตรียมข้อมูลสำหรับกระบวนการตั้งค่าจริงในขั้นตอนถัดไป |

**ตารางที่ 3-56** แสดง Process Specification 18.1 กำหนดค่า Spanning Tree

|  |  |
| --- | --- |
| Process Number : | 18.1 |
| Process Name: | การตั้งค่า Spanning Tree บนอุปกรณ์เครือข่าย |
| Description : | กระบวนการนี้ใช้ข้อมูลการตั้งค่า STP ที่รับจาก Process 17.1 เพื่อดำเนินการตั้งค่าจริงบนอุปกรณ์เครือข่าย เช่น การกำหนดโหมด STP, การตั้งค่า Root Primary และ Secondary, การเปิด/ปิด PortFast และการตั้งค่าพอร์ตที่ใช้ PortFast |
| Input Data Flow : | ข้อมูลการตั้งค่า STP ที่ได้รับจากผู้ใช้ เช่น โหมด STP, การตั้งค่า Root Primary และ Secondary, การเปิด/ปิด PortFast |
| Output Data Flow : | สถานะการตั้งค่าของอุปกรณ์ เช่น สำเร็จหรือล้มเหลว |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เริ่มต้นด้วยการเชื่อมต่อกับอุปกรณ์ผ่าน SSH โดยใช้ข้อมูลจากผู้ใช้ จากนั้นจะทำการตั้งค่า Spanning Tree ตามโหมดที่เลือก (เช่น pvst หรือ rapid-pvst) พร้อมทั้งตั้งค่าการทำงานของ Root Primary, Root Secondary และ PortFast บนพอร์ตที่ระบุ หากการตั้งค่าเสร็จสมบูรณ์จะมีการแสดงผลลัพธ์และการจัดการข้อผิดพลาด เช่น การเชื่อมต่อล้มเหลว หรือการตั้งค่าไม่ถูกต้อง โดยสุดท้ายระบบจะตัดการเชื่อมต่อจากอุปกรณ์ |

**ตารางที่ 3-57** แสดง Process Specification 19.1 รับข้อมูลการตั้งค่า Aggregation

|  |  |
| --- | --- |
| Process Number : | 19.1 |
| Process Name: | การรับข้อมูลการตั้งค่า Aggregation |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่าที่ผู้ใช้ระบุสำหรับการตั้งค่า Aggregation เช่น EtherChannel โดยข้อมูลเหล่านี้จะรวมถึงการตั้งค่าพอร์ตที่ใช้ในการเชื่อมต่อ, หมายเลข Channel Group, โหมด PAgP, LACP, การลบพอร์ตที่ใช้ใน LACP และการตรวจสอบสถานะต่าง ๆ ที่เกี่ยวข้องกับการตั้งค่า Aggregation |
| Input Data Flow : | ข้อมูลการตั้งค่า Aggregation เช่น ชื่ออุปกรณ์, พอร์ตที่ใช้สำหรับ EtherChannel, หมายเลข Channel Group, โหมด PAgP, LACP, และข้อมูลที่เกี่ยวข้องอื่น ๆ ที่ระบุโดยผู้ใช้ |
| Output Data Flow : | ข้อมูลการตั้งค่าที่พร้อมสำหรับการตั้งค่าการรวมพอร์ตไปยัง Process 20.1 |
| Process Type : | Online Processing |
| Process Logic : | ระบบรับข้อมูลจากฟอร์มที่ผู้ใช้กรอกบนหน้าเว็บและตรวจสอบข้อมูล หากพบความผิดพลาด เช่น อุปกรณ์ไม่พบในฐานข้อมูล หรือค่าที่กรอกไม่ถูกต้อง ระบบจะแจ้งเตือนผู้ใช้ผ่านหน้าต่างข้อความ หลังจากตรวจสอบข้อมูลเสร็จสิ้น ระบบจะจัดเตรียมข้อมูลสำหรับการตั้งค่าการรวมพอร์ตจริงในขั้นตอนถัดไป |

**ตารางที่ 3-58** แสดง Process Specification 20.1 กำหนดค่า Aggregation

|  |  |
| --- | --- |
| Process Number : | 20.1 |
| Process Name: | การตั้งค่าการรวมพอร์ต (Aggregation) |
| Description : | กระบวนการนี้จะทำการตั้งค่าการรวมพอร์ต (EtherChannel) บนอุปกรณ์ที่ได้รับการระบุในขั้นตอนก่อนหน้า โดยจะตั้งค่าด้วยการใช้โหมด PAgP หรือ LACP สำหรับการรวมพอร์ต รวมทั้งการตั้งค่าหมายเลข Channel Group สำหรับแต่ละโหมด การตั้งค่าเหล่านี้จะถูกนำไปใช้กับพอร์ตที่ระบุในขั้นตอนก่อนหน้า และสามารถลบพอร์ตที่ไม่ต้องการได้ตามคำขอ |
| Input Data Flow : | ข้อมูลการตั้งค่า EtherChannel ที่ได้รับจาก Process 19.1 เช่น พอร์ตที่ใช้สำหรับ EtherChannel, หมายเลข Channel Group, โหมด PAgP, โหมด LACP, การลบพอร์ต |
| Output Data Flow : | สถานะการตั้งค่าของ EtherChannel บนอุปกรณ์ เช่น สำเร็จหรือล้มเหลว โดยแสดงผลลัพธ์ผ่านข้อความหรือข้อผิดพลาดที่พิมพ์ออกมา |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เริ่มต้นด้วยการเชื่อมต่อกับอุปกรณ์ผ่าน SSH โดยใช้ข้อมูลที่ได้รับจากผู้ใช้ จากนั้นจะทำการตั้งค่าการรวมพอร์ต (EtherChannel) ตามโหมดที่เลือก (PAgP หรือ LACP) พร้อมทั้งตั้งค่าหมายเลข Channel Group ที่ระบุ หากการตั้งค่าการรวมพอร์ตเสร็จสมบูรณ์ ระบบจะทำการแสดงผลลัพธ์และจัดการข้อผิดพลาด เช่น การเชื่อมต่อล้มเหลว, การตั้งค่าผิดพลาด, หรือการลบพอร์ตที่ไม่ต้องการ จากนั้นจะตัดการเชื่อมต่อจากอุปกรณ์ |

**ตารางที่ 3-59** แสดง Process Specification 21.1 รับข้อมูลการตั้งค่า Static Route

|  |  |
| --- | --- |
| Process Number : | 21.1 |
| Process Name: | การรับข้อมูลการตั้งค่า Static Route |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่าที่ผู้ใช้กรอกสำหรับการตั้งค่า Static Route บนอุปกรณ์เครือข่าย ข้อมูลที่ได้รับรวมถึงเครือข่ายปลายทาง (Destination Networks), พอร์ตหรือ Next Hop (Exit Interfaces or Next Hops), การตั้งค่า Default Route, และการลบการตั้งค่า Static Route ที่มีอยู่ |
| Input Data Flow : | ข้อมูลที่กรอกโดยผู้ใช้ผ่านฟอร์มในหน้าเว็บ เช่น เครือข่ายปลายทาง (Destination Networks), พอร์ตหรือ Next Hop (Exit Interfaces or Next Hops), การตั้งค่า Default Route, การลบ Default Route และ Static Route ที่มีอยู่ |
| Output Data Flow : | ข้อมูลที่เตรียมไว้สำหรับการตั้งค่า Static Route ในขั้นตอนถัดไป ซึ่งจะถูกส่งไปยัง Process 22.1 |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เริ่มต้นด้วยการรับข้อมูลจากฟอร์มที่ผู้ใช้กรอกบนหน้าเว็บ เช่น เครือข่ายปลายทาง, พอร์ตหรือ Next Hop, การตั้งค่า Default Route และการลบ Static Route ที่มีอยู่ หากมีข้อผิดพลาด เช่น ไม่พบอุปกรณ์ในฐานข้อมูล หรือค่าที่กรอกไม่ถูกต้อง ระบบจะแจ้งเตือนผู้ใช้ หลังจากนั้นข้อมูลที่ได้รับจะถูกจัดเตรียมเพื่อส่งไปยังกระบวนการตั้งค่าจริงในขั้นตอนถัดไป (Process 22.1) |

**ตารางที่ 3-60** แสดง Process Specification 22.1 การกำหนดค่า Static Route

|  |  |
| --- | --- |
| Process Number : | 22.1 |
| Process Name: | การกำหนดค่า Static Route |
| Description : | กระบวนการนี้ใช้ข้อมูลการตั้งค่า Static Route ที่รับมาจาก Process 21.1 เพื่อตั้งค่า Static Route บนอุปกรณ์เครือข่าย โดยการกำหนดเครือข่ายปลายทาง, พอร์ตหรือ Next Hop, การตั้งค่า Default Route, และการลบ Static Route ที่มีอยู่ ข้อมูลที่ได้รับจาก Process 21.1 จะถูกแปลงและประมวลผลก่อนที่จะส่งคำสั่งการตั้งค่าไปยังอุปกรณ์ |
| Input Data Flow : | ข้อมูลการตั้งค่า Static Route ที่ได้รับจาก Process 21.1 เช่น เครือข่ายปลายทาง (Destination Networks), พอร์ตหรือ Next Hop (Exit Interfaces or Next Hops), การตั้งค่า Default Route, การลบ Default Route และ Static Route ที่มีอยู่ |
| Output Data Flow : | ข้อมูลที่ผ่านการแปลง subnet mask ซึ่งจะส่งไปยัง Process 22.2 ก่อนที่จะทำการส่งคำสั่งตั้งค่าจริงไปยังอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เริ่มต้นด้วยการรับข้อมูลการตั้งค่า Static Route จาก Process 21.1 จากนั้นจะทำการแปลง CIDR เป็น subnet mask โดยใช้ฟังก์ชัน convert\_cidr\_to\_netmask กระบวนการนี้จะจัดเตรียมคำสั่งการตั้งค่า Static Route พร้อม subnet mask ที่ถูกแปลงแล้ว เพื่อส่งไปยัง Process 22.2 สำหรับการประมวลผลเพิ่มเติม จากนั้นคำสั่งการตั้งค่าจะถูกส่งไปยังอุปกรณ์เครือข่ายเพื่อตั้งค่าตามคำสั่งที่เตรียมไว้ การแปลง subnet mask ใน Process 22.2 เป็นขั้นตอนสำคัญก่อนการส่งคำสั่งการตั้งค่าไปยังอุปกรณ์เพื่อให้การตั้งค่า Static Route ถูกต้องและสมบูรณ์ |

**ตารางที่ 3-61** แสดง Process Specification 22.2 การแปลงรูปแบบ IP

|  |  |
| --- | --- |
| Process Number : | 22.2 |
| Process Name: | การแปลงรูปแบบ IP (Subnet Mask) |
| Description : | กระบวนการนี้รับข้อมูลจาก Process 22.1 ซึ่งประกอบไปด้วย CIDR (Classless Inter-Domain Routing) ที่ใช้ในการตั้งค่า Static Route จากนั้นจะทำการแปลง CIDR เป็น Subnet Mask ในรูปแบบ IP ที่อุปกรณ์สามารถเข้าใจและใช้งานได้ ก่อนที่จะส่งข้อมูลที่แปลงแล้วกลับไปยัง Process 22.1 เพื่อทำการตั้งค่า Static Route บนอุปกรณ์ |
| Input Data Flow : | ข้อมูล CIDR ที่ได้รับจาก Process 22.1 ซึ่งประกอบไปด้วยค่าของเครือข่ายปลายทางในรูปแบบ CIDR เช่น 192.168.1.0/24 |
| Output Data Flow : | Subnet Mask ที่แปลงจาก CIDR เป็นรูปแบบ IP เช่น 255.255.255.0 ซึ่งจะส่งกลับไปยัง Process 22.1 เพื่อใช้ในการตั้งค่า Static Route |
| Process Type : | Online Processing  (Process 2.2 จัดเป็น Calculation/Conversion เพราะมันเป็นเพียงการคำนวณและแปลงข้อมูลจาก CIDR เป็น Subnet Mask ที่ใช้ในกระบวนการต่อไปโดยไม่มีการเชื่อมต่อหรือการตอบสนองจากอุปกรณ์เครือข่ายในทันที) |
| Process Logic : | กระบวนการนี้ทำการแปลง CIDR ที่รับมาจาก Process 22.1 ให้เป็น Subnet Mask โดยการแยกหมายเลขบิตจาก CIDR และคำนวณค่า Subnet Mask สำหรับแต่ละบิต โดยการเพิ่มค่า 255 สำหรับบิตที่ครบ 8 และคำนวณค่า Subnet Mask สำหรับบิตที่เหลือ จนกว่าทุกบิตจะได้รับการแปลงเป็น Subnet Mask ที่สมบูรณ์ จากนั้นจะส่งผลลัพธ์กลับไปยัง Process 22.1 เพื่อใช้ในการกำหนด Static Route ต่อไป |

**ตารางที่ 3-62** แสดง Process Specification 23.1 รับข้อมูลการตั้งค่า OSPF

|  |  |
| --- | --- |
| Process Number : | 23.1 |
| Process Name: | รับข้อมูลการตั้งค่า OSPF (Receive OSPF Configuration Data) |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่าจากผู้ใช้เพื่อการตั้งค่า OSPF บนอุปกรณ์ต่างๆ ผ่านทางเว็บแอปพลิเคชัน ซึ่งประกอบไปด้วยชื่ออุปกรณ์, การกำหนดเครือข่ายปลายทาง, โซนของ OSPF (OSPF Areas), Process ID, Router ID และการลบข้อมูลบางส่วนของ OSPF ที่มีอยู่ |
| Input Data Flow : | ข้อมูลการตั้งค่า OSPF เช่น ชื่ออุปกรณ์ (device\_name), เครือข่ายปลายทาง (destination\_networks), OSPF Areas, Process ID, Router ID, เครือข่ายที่จะลบ (remove\_destination\_networks), โซน OSPF ที่จะลบ (remove\_ospf\_areas) และ Process ID ที่ต้องการลบ (delete\_process\_id) |
| Output Data Flow : | ข้อมูลที่ได้รับการประมวลผลจาก Input Data Flow ซึ่งจะถูกส่งต่อไปยัง Process 24.1 (ตั้งค่า OSPF Routing) เพื่อทำการตั้งค่า OSPF บนอุปกรณ์ตามข้อมูลที่ได้รับ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้ทำการรับข้อมูลจากเว็บแอปพลิเคชันที่ผู้ใช้กรอก เช่น ชื่ออุปกรณ์, เครือข่ายปลายทาง, OSPF Areas, Process ID, และ Router ID โดยมีการดึงข้อมูลจากฐานข้อมูลเพื่อค้นหา IP ของอุปกรณ์ที่ตรงกับชื่อที่ระบุและดำเนินการกับข้อมูลที่ได้รับ รวมถึงการลบเครือข่ายปลายทางและโซน OSPF ที่ผู้ใช้ต้องการลบ หากมีการระบุ Process ID ที่ต้องการลบ ก็จะถูกจัดการด้วย ข้อมูลที่ประมวลผลแล้วจะถูกส่งต่อไปยัง Process 24.1 สำหรับการตั้งค่า OSPF Routing บนอุปกรณ์ต่อไป |

**ตารางที่ 3-63** แสดง Process Specification 24.1 กำหนดค่า OSPF

|  |  |
| --- | --- |
| Process Number : | 24.1 |
| Process Name: | ตั้งค่า OSPF อุปกรณ์ |
| Description : | กระบวนการนี้จะทำการตั้งค่า OSPF บนอุปกรณ์เครือข่ายที่กำหนด โดยรับข้อมูลการตั้งค่าจาก Process 23.1 รวมถึงการตั้งค่าเครือข่ายปลายทาง (destination\_networks), OSPF Areas, Process ID, Router ID และการลบข้อมูลที่มีอยู่ก่อนหน้า เช่น เครือข่ายหรือโซน OSPF ที่จะลบจากการตั้งค่าปัจจุบัน กระบวนการจะเชื่อมต่อไปยังอุปกรณ์ที่ได้รับการกำหนดในฐานข้อมูล และทำการส่งคำสั่งการตั้งค่าไปยังอุปกรณ์เหล่านั้น |
| Input Data Flow : | ข้อมูลการตั้งค่าจาก Process 23.1 เช่น ชื่ออุปกรณ์, Process ID |
| Output Data Flow : | ข้อมูลการตั้งค่า OSPF ที่ได้รับการประมวลผลแล้วจะถูกส่งต่อไปยัง Process 24.2 เพื่อทำการแปลง CIDR เป็น Subnet Mask ก่อนที่จะส่งคำสั่งไปยังอุปกรณ์สำหรับการตั้งค่า OSPF บนอุปกรณ์จริง |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะเริ่มต้นโดยการเชื่อมต่อกับอุปกรณ์ผ่านการตั้งค่า Device Information ที่มีการเชื่อมต่อจากฐานข้อมูล หลังจากนั้นจะสร้างคำสั่งสำหรับการตั้งค่า OSPF ซึ่งรวมถึงการกำหนดเครือข่ายปลายทางและ OSPF Areas ที่ได้รับจากผู้ใช้ ผ่านการใช้ Process ID และ Router ID ที่ผู้ใช้กำหนด คำสั่งการตั้งค่าจะถูกส่งไปยังอุปกรณ์ตามที่ระบุ และหากมีการลบข้อมูล OSPF ที่มีอยู่ก่อนหน้านี้ (destination\_networks, ospf\_areas) จะมีการส่งคำสั่งลบไปยังอุปกรณ์เช่นกัน กระบวนการนี้จะทำการแปลง CIDR เป็น Subnet Mask ด้วยฟังก์ชัน convert\_cidr\_to\_netmask ซึ่งเป็นขั้นตอนสำคัญก่อนการส่งคำสั่งการตั้งค่าไปยังอุปกรณ์ |

**ตารางที่ 3-64** แสดง Process Specification 24.2 การแปลงรูปแบบ IP

|  |  |
| --- | --- |
| Process Number : | 24.2 |
| Process Name: | การแปลงรูปแบบ IP (Subnet Mask) |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่า OSPF ที่ได้รับจาก Process 24.1 ซึ่งประกอบไปด้วย CIDR (Classless Inter-Domain Routing) ที่ใช้ในการตั้งค่า OSPF จากนั้นจะทำการแปลง CIDR เป็น Subnet Mask ในรูปแบบ IP ที่อุปกรณ์สามารถเข้าใจและใช้งานได้ ก่อนที่จะส่งข้อมูลที่แปลงแล้วกลับไปยัง Process 24.1 เพื่อทำการตั้งค่า OSPF บนอุปกรณ์ |
| Input Data Flow : | ข้อมูล CIDR ที่ได้รับจาก Process 24.1 ซึ่งประกอบไปด้วยค่าของเครือข่ายปลายทางในรูปแบบ CIDR เช่น 192.168.1.0/24 |
| Output Data Flow : | Subnet Mask ที่แปลงจาก CIDR เป็นรูปแบบ IP เช่น 255.255.255.0 ซึ่งจะส่งกลับไปยัง Process 24.1 เพื่อใช้ในการตั้งค่า OSPF |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้ทำการแปลง CIDR ที่รับมาจาก Process 22.1 ให้เป็น Subnet Mask โดยการแยกหมายเลขบิตจาก CIDR และคำนวณค่า Subnet Mask สำหรับแต่ละบิต โดยการเพิ่มค่า 255 สำหรับบิตที่ครบ 8 และคำนวณค่า Subnet Mask สำหรับบิตที่เหลือ จนกว่าทุกบิตจะได้รับการแปลงเป็น Subnet Mask ที่สมบูรณ์ จากนั้นจะส่งผลลัพธ์กลับไปยัง Process 24.1 เพื่อใช้ในการกำหนด OSPF บนอุปกรณ์ต่อไป |

**ตารางที่ 3-65** แสดง Process Specification 25.1 รับข้อมูลการตั้งค่า RIPv2

|  |  |
| --- | --- |
| Process Number : | 25.1 |
| Process Name: | รับข้อมูลการตั้งค่า RIPv2 |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่าจากผู้ใช้เพื่อการตั้งค่า RIPv2 บนอุปกรณ์ต่างๆ ผ่านทางเว็บแอปพลิเคชัน ซึ่งประกอบไปด้วยชื่ออุปกรณ์, การกำหนดเครือข่ายปลายทาง, การตั้งค่า Auto Summary, การลบเครือข่ายปลายทางที่มีอยู่, และการปิดการใช้งาน RIPv2 บนอุปกรณ์ |
| Input Data Flow : | ข้อมูลการตั้งค่า RIPv2 เช่น ชื่ออุปกรณ์ (device\_name), เครือข่ายปลายทาง (destination\_networks), การตั้งค่า Auto Summary (auto\_summary), เครือข่ายที่จะลบ  (remove\_destination\_networks), และการปิดการใช้งาน RIPv2 (disable\_rip) |
| Output Data Flow : | การตั้งค่า RIPv2 ที่ส่งไปยัง Process 26.1 เพื่อทำการกำหนดค่าเส้นทาง RIPv2 บนอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะรับข้อมูลจากผู้ใช้เกี่ยวกับการตั้งค่า RIPv2 และตรวจสอบความถูกต้องของข้อมูล เช่น ชื่ออุปกรณ์และเครือข่ายปลายทาง จากนั้นจะส่งข้อมูลเหล่านี้ไปยัง Process 26.1 เพื่อทำการตั้งค่าเส้นทาง RIPv2 บนอุปกรณ์ โดยการใช้ฟังก์ชัน configure\_rip\_route เพื่อดำเนินการตั้งค่าเหล่านี้บนอุปกรณ์ที่เลือก |

**ตารางที่ 3-66** แสดง Process Specification 26.1 กำหนดค่า RIPv2

|  |  |
| --- | --- |
| Process Number : | 26.1 |
| Process Name: | กำหนดค่า RIPv2 |
| Description : | กระบวนการนี้จะทำการกำหนดค่า RIPv2 บนอุปกรณ์ที่ได้รับข้อมูลการตั้งค่าจากกระบวนการ 25.1 โดยการตั้งค่าเครือข่ายปลายทาง (destination networks), การตั้งค่า Auto Summary, การลบเครือข่ายที่มีอยู่, และการปิดการใช้งาน RIPv2 (disable rip) |
| Input Data Flow : | ข้อมูลที่ได้รับจากกระบวนการ 25.1 |
| Output Data Flow : | การตั้งค่า RIPv2 ที่ได้ถูกส่งไปยังอุปกรณ์และแสดงผลลัพธ์ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะเชื่อมต่อกับอุปกรณ์ที่ได้รับข้อมูลจากกระบวนการ 25.1 ผ่านโปรโตคอล Netmiko และทำการกำหนดค่าต่างๆ บนอุปกรณ์ตามที่ได้รับข้อมูล เช่น การตั้งค่าเครือข่ายปลายทาง, การเปิด/ปิด Auto Summary, การลบเครือข่ายปลายทางที่มีอยู่ และการปิดการใช้งาน RIPv2 ถ้าผู้ใช้เลือก จากนั้นจะส่งคำสั่งการตั้งค่าไปยังอุปกรณ์ และปิดการเชื่อมต่อเมื่อการตั้งค่าเสร็จสิ้น |

**ตารางที่ 3-67** แสดง Process Specification 27.1 รับข้อมูลการตั้งค่า EIGRP

|  |  |
| --- | --- |
| Process Number : | 27.1 |
| Process Name: | รับข้อมูลการตั้งค่า EIGRP |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่า EIGRP จากผู้ใช้ผ่านเว็บแอปพลิเคชัน ซึ่งประกอบด้วยการตั้งค่า Process ID, Router ID, การกำหนดเครือข่ายปลายทาง (destination networks), การลบเครือข่ายปลายทางที่มีอยู่, การลบ Process ID และการตั้งค่าอื่นๆ ที่เกี่ยวข้องกับการกำหนดค่า EIGRP บนอุปกรณ์ต่างๆ |
| Input Data Flow : | ข้อมูลการตั้งค่า EIGRP ที่ได้รับจากผู้ใช้ เช่น Process ID, Router ID, เครือข่ายปลายทาง (destination\_networks), การลบเครือข่าย (remove\_destination\_networks), การลบ Process ID  (delete\_process\_id) และค่า Process ID ที่ต้องการลบ (process\_id\_input) |
| Output Data Flow : | ข้อมูลที่ส่งต่อไปยัง Process 28.1 สำหรับการดำเนินการตั้งค่า EIGRP บนอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะรวบรวมข้อมูลการตั้งค่าที่ผู้ใช้ป้อนเข้ามา เช่น Process ID, Router ID, destination networks และข้อมูลอื่นๆ ที่เกี่ยวข้อง และส่งต่อข้อมูลเหล่านี้ไปยัง Process 28.1 สำหรับการดำเนินการตั้งค่า EIGRP บนอุปกรณ์ |

**ตารางที่ 3-67** แสดง Process Specification 28.1 กำหนดค่า EIGRP

|  |  |
| --- | --- |
| Process Number : | 28.1 |
| Process Name: | กำหนดค่า EIGRP |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่าจาก Process 27.1 ซึ่งรวมถึงข้อมูลเกี่ยวกับ EIGRP เช่น Process ID, Router ID, เครือข่ายปลายทาง (destination networks), การลบเครือข่ายปลายทางที่มีอยู่, และการลบ Process ID นอกจากนี้ยังมีการส่งข้อมูลบางส่วนไปยัง Process 28.2 เพื่อทำการแปลง CIDR เป็น Subnet Mask ก่อนที่จะดำเนินการตั้งค่าบนอุปกรณ์ |
| Input Data Flow : | ข้อมูลจาก Process 27.1 ที่ประกอบด้วย Process ID, Router ID, destination networks, remove destination networks, delete process ID, และ process ID ที่ต้องการลบ (process\_id\_input) |
| Output Data Flow : | การตั้งค่า EIGRP ที่พร้อมส่งไปยังอุปกรณ์ที่ต้องการ |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้เริ่มต้นด้วยการรับข้อมูลจาก Process 27.1 และตรวจสอบค่า CIDR ใน destination networks หากมีการระบุ CIDR จะถูกส่งไปยัง Process 28.2 เพื่อทำการแปลงเป็น Subnet Mask โดยอัตโนมัติ หลังจากที่ได้ Subnet Mask แล้ว ข้อมูลทั้งหมดจะถูกใช้ในการสร้างคำสั่งการตั้งค่า EIGRP บนอุปกรณ์ผ่านการเชื่อมต่อกับอุปกรณ์นั้นๆ โดยการส่งคำสั่งเพื่อกำหนดเครือข่ายที่ใช้ใน EIGRP การกำหนด Router ID และการลบเครือข่ายที่ไม่ต้องการออกจากการตั้งค่าของ EIGRP นอกจากนี้ยังมีการลบ Process ID หากได้รับคำสั่งจากผู้ใช้ |

**ตารางที่ 3-68** แสดง Process Specification 28.2 การแปลงรูปแบบ IP

|  |  |
| --- | --- |
| Process Number : | 28.2 |
| Process Name: | การแปลงรูปแบบ IP (Subnet Mask) |
| Description : | กระบวนการนี้รับข้อมูลการตั้งค่า EIGRP ที่ได้รับจาก Process 28.1 ซึ่งประกอบไปด้วย CIDR (Classless Inter-Domain Routing) ที่ใช้ในการตั้งค่า EIGRP จากนั้นจะทำการแปลง CIDR เป็น Subnet Mask ในรูปแบบ IP ที่อุปกรณ์สามารถเข้าใจและใช้งานได้ ก่อนที่จะส่งข้อมูลที่แปลงแล้วกลับไปยัง Process 28.1 เพื่อทำการตั้งค่า EIGRP บนอุปกรณ์ |
| Input Data Flow : | ข้อมูล CIDR ที่ได้รับจาก Process 28.1 ซึ่งประกอบไปด้วยค่าของเครือข่ายปลายทางในรูปแบบ CIDR เช่น 192.168.1.0/24 |
| Output Data Flow : | Subnet Mask ที่แปลงจาก CIDR เป็นรูปแบบ IP เช่น 255.255.255.0 ซึ่งจะส่งกลับไปยัง Process 28.1 เพื่อใช้ในการตั้งค่า EIGRP |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้ทำการแปลง CIDR ที่รับมาจาก Process 28.1 ให้เป็น Subnet Mask โดยการแยกหมายเลขบิตจาก CIDR และคำนวณค่า Subnet Mask สำหรับแต่ละบิต โดยการเพิ่มค่า 255 สำหรับบิตที่ครบ 8 และคำนวณค่า Subnet Mask สำหรับบิตที่เหลือ จนกว่าทุกบิตจะได้รับการแปลงเป็น Subnet Mask ที่สมบูรณ์ จากนั้นจะส่งผลลัพธ์กลับไปยัง Process 28.1 เพื่อใช้ในการกำหนด EIGRP บนอุปกรณ์ต่อไป |

**ตารางที่ 3-69** แสดง Process Specification 29.1 เเสดงอุปกรณ์ที่มีในระบบทั้งหมด

|  |  |
| --- | --- |
| Process Number : | 29.1 |
| Process Name: | แสดงอุปกรณ์ที่มีในระบบทั้งหมด |
| Description : | กระบวนการนี้จะทำการแสดงรายการอุปกรณ์ทั้งหมดที่มีในระบบ โดยจะดึงข้อมูลอุปกรณ์จากฐานข้อมูล และแสดงรายการอุปกรณ์ที่มีในระบบให้ผู้ใช้งานเลือกดำเนินการ เช่น การลบการตั้งค่าหรือรีโหลดการตั้งค่าของอุปกรณ์ |
| Input Data Flow : | ข้อมูลอุปกรณ์ที่จัดเก็บในฐานข้อมูล D1 |
| Output Data Flow : | แสดงรายการอุปกรณ์ทั้งหมดในระบบในรูปแบบตาราง |
| Process Type : | Online Processing |
| Process Logic : | ในกระบวนการนี้ เมื่อผู้ใช้งานเข้าถึงหน้าการลบการตั้งค่า (erase\_config\_page) ระบบจะทำการดึงข้อมูลอุปกรณ์ทั้งหมดจากฐานข้อมูล และแสดงข้อมูลอุปกรณ์ในรูปแบบตารางที่สามารถคลิกเพื่อทำการลบ รีโหลด หรือบันทึกการตั้งค่าของอุปกรณ์ โดยใช้ HTML สำหรับการแสดงผล พร้อมกับปุ่มให้ผู้ใช้สามารถทำการลบ, รีโหลดได้ |

**ตารางที่ 3-70** แสดง Process Specification 30.1 คืนค่าโรงงาน

|  |  |
| --- | --- |
| Process Number : | 30.1 |
| Process Name: | ลบการตั้งค่าโรงงาน |
| Description : | กระบวนการนี้จะทำการลบการตั้งค่าของอุปกรณ์และคืนค่าการตั้งค่าโรงงาน โดยการส่งคำสั่งไปยังอุปกรณ์ผ่าน SSH เพื่อทำการลบการตั้งค่า startup-config และรีโหลดอุปกรณ์ |
| Input Data Flow : | ข้อมูลอุปกรณ์ที่ต้องการลบการตั้งค่า ซึ่งได้แก่ที่อยู่ IP, ชื่อผู้ใช้, รหัสผ่าน และรหัสผ่าน enable ที่เก็บไว้ในฐานข้อมูลจากกระบวนการ 29.1 |
| Output Data Flow : | ข้อความผลลัพธ์ที่แจ้งว่าได้ลบการตั้งค่าเรียบร้อยแล้ว และอุปกรณ์จะรีโหลด |
| Process Type : | Online Processing |
| Process Logic : | เมื่อผู้ใช้งานเลือกอุปกรณ์ที่จะลบการตั้งค่าโรงงานผ่านหน้าเว็บ (erase\_config\_page) ระบบจะทำการเชื่อมต่อไปยังอุปกรณ์โดยใช้ SSH และส่งคำสั่งในการลบการตั้งค่า startup-config การลบการตั้งค่าสำเร็จจะมีการลบอุปกรณ์จากฐานข้อมูล และแสดงข้อความแจ้งเตือนผู้ใช้งาน |

**ตารางที่ 3-71** แสดง Process Specification 31.1 รีสตาร์ทอุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 31.1 |
| Process Name: | รีสตาร์ทอุปกรณ์ |
| Description : | กระบวนการนี้จะทำการรีสตาร์ทอุปกรณ์ที่เลือกจากกระบวนการ 29.1 หลังจากที่แสดงรายการอุปกรณ์ทั้งหมด โดยจะเชื่อมต่อผ่าน SSH และส่งคำสั่งเพื่อรีสตาร์ทอุปกรณ์ |
| Input Data Flow : | ข้อมูลของอุปกรณ์ที่ได้รับจากกระบวนการ 29.1 ซึ่งประกอบไปด้วยที่อยู่ IP, ชื่อผู้ใช้, รหัสผ่าน และรหัสผ่าน enable ของอุปกรณ์ที่เลือก |
| Output Data Flow : | ข้อความแจ้งเตือนว่าการรีสตาร์ทอุปกรณ์เสร็จสมบูรณ์หรือหากมีข้อผิดพลาด |
| Process Type : | Online Processing |
| Process Logic : | หลังจากที่ผู้ใช้งานเลือกอุปกรณ์จากการแสดงผลในกระบวนการ 29.1, ระบบจะทำการเชื่อมต่อ SSH ไปยังอุปกรณ์ที่เลือกและส่งคำสั่ง reload เพื่อรีสตาร์ทอุปกรณ์. หากมีการปรับการตั้งค่าใดๆ ระบบจะถามผู้ใช้ว่าให้บันทึกการตั้งค่าก่อนรีสตาร์ทหรือไม่ |

**ตารางที่ 3-72** แสดง Process Specification 32.1 ตรวจสอบการตั้งค่าของอุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 32.1 |
| Process Name: | ตรวจสอบการตั้งค่าของอุปกรณ์ |
| Description : | กระบวนการนี้จะให้ผู้ใช้สามารถเลือกอุปกรณ์จากรายการในระบบและส่งคำสั่งที่ต้องการเพื่อดึงข้อมูลการตั้งค่าจากอุปกรณ์ โดยการเชื่อมต่อผ่าน SSH เพื่อดึงข้อมูลการตั้งค่าและแสดงผลให้ผู้ใช้เห็น |
| Input Data Flow : | ข้อมูลอุปกรณ์ที่มีในระบบ เช่น ชื่ออุปกรณ์, ที่อยู่ IP, ชื่อผู้ใช้, รหัสผ่าน, และรหัสผ่าน enable ซึ่งเก็บไว้ในฐานข้อมูลจากกระบวนการก่อนหน้านี้ พร้อมกับคำสั่งที่ผู้ใช้เลือก เช่น show running-config, show version, show interfaces |
| Output Data Flow : | ข้อมูลการตั้งค่าที่ได้รับจากอุปกรณ์ เช่น ผลลัพธ์จากคำสั่งที่เลือก รวมถึงข้อความแสดงข้อผิดพลาดหากไม่สามารถดึงข้อมูลได้ |
| Process Type : | Online Processing |
| Process Logic : | เมื่อผู้ใช้เลือกอุปกรณ์และคำสั่งที่ต้องการดูการตั้งค่า ระบบจะเชื่อมต่อไปยังอุปกรณ์ผ่าน SSH และส่งคำสั่งไปยังอุปกรณ์เพื่อดึงข้อมูลการตั้งค่า หลังจากนั้นจะแสดงผลการตั้งค่าบนเว็บให้ผู้ใช้เห็น หากเกิดข้อผิดพลาดจะมีข้อความแสดงข้อผิดพลาด |

**ตารางที่ 3-73** แสดง Process Specification 33.1 การตรวจสอบที่อาจจะเป็นช่องโหว่ของอุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 33.1 |
| Process Name: | การตรวจสอบที่อาจจะเป็นช่องโหว่ของอุปกรณ์ |
| Description : | กระบวนการนี้ทำการดึงข้อมูลการตั้งค่าของอุปกรณ์ทั้งหมดจาก D1 ผ่านการเชื่อมต่อ SSH และส่งคำสั่ง show running-config และ show ip interface brief ไปยังอุปกรณ์แต่ละตัว เพื่อให้ 34.1 ทำการวิเคราะห์ช่องโหว่ จากนั้นจะส่งผลลัพธ์จาก 34.1 กลับมาบันทึกลงใน D1 |
| Input Data Flow : | ข้อมูลการตั้งค่าของอุปกรณ์จาก D1 ที่ถูกดึงออกมาโดยอัตโนมัติ (ไม่มี input จากผู้ใช้) |
| Output Data Flow : | ผลลัพธ์จากคำสั่ง show running-config และ show ip interface brief ที่ถูกส่งไปให้ 34.1 และผลการวิเคราะห์จาก 34.1 ที่ส่งกลับมาบันทึกใน D1 |
| Process Type : | ระบบอัตโนมัติ (Automated Process) |
| Process Logic : | กระบวนการจะดึงข้อมูลการตั้งค่าของอุปกรณ์จาก D1 โดยการเชื่อมต่อ SSH ไปยังแต่ละอุปกรณ์ และส่งคำสั่ง show running-config และ show ip interface brief ไปยังอุปกรณ์เพื่อให้ 34.1 ทำการวิเคราะห์ช่องโหว่ จากนั้นจะบันทึกผลการวิเคราะห์ที่ได้รับจาก 34.1 ลงใน D1 |

**ตารางที่ 3-74** แสดง Process Specification 34.1 ตรวจสอบการตั้งค่าที่อาจจะเป็นช่องโหว่

|  |  |
| --- | --- |
| Process Number : | 34.1 |
| Process Name: | ตรวจสอบการตั้งค่าที่อาจจะเป็นช่องโหว่ |
| Description : | กระบวนการนี้ทำการวิเคราะห์ข้อมูลการตั้งค่าของอุปกรณ์ที่ได้รับจาก 33.1 โดยจะรับข้อมูลการตั้งค่า (show running-config) และข้อมูลจากคำสั่ง show ip interface brief เพื่อตรวจสอบช่องโหว่ด้านความปลอดภัย เช่น รหัสผ่านที่อ่อนแอ การตั้งค่าการเข้ารหัสรหัสผ่านที่ไม่ได้เปิดใช้งาน หรือการตั้งค่าพอร์ตที่ไม่ได้ใช้งาน เป็นต้น |
| Input Data Flow : | ข้อมูลการตั้งค่าที่ได้รับจาก 33.1 |
| Output Data Flow : | ผลการวิเคราะห์ช่องโหว่ที่เกิดขึ้นจากการตั้งค่าของอุปกรณ์ ซึ่งจะประกอบด้วยคำเตือนเกี่ยวกับช่องโหว่ที่พบ เช่น การตั้งค่ารหัสผ่านที่อ่อนแอ การเปิดใช้งานช่องทางการเข้าถึงที่ไม่ปลอดภัย หรือพอร์ตที่ไม่ได้ใช้งาน |
| Process Type : | ระบบอัตโนมัติ (Automated Process) |
| Process Logic : | กระบวนการนี้จะรับข้อมูลจาก 33.1 ซึ่งเป็นผลลัพธ์ของคำสั่ง show running-config และ show ip interface brief ที่ดึงมาจากอุปกรณ์ต่างๆ และจะใช้เครื่องมือที่ถูกฝึกไว้ล่วงหน้าในการวิเคราะห์การตั้งค่าดังกล่าวเพื่อหาช่องโหว่ที่อาจเกิดขึ้น เช่น รหัสผ่านที่อ่อนแอ หรือการตั้งค่าความปลอดภัยที่ไม่ได้เปิดใช้งาน จากนั้นจะส่งผลการวิเคราะห์กลับไปยัง 33.1 เพื่อนำไปบันทึกและเก็บข้อมูลในระบบ |

**ตารางที่ 3-75** แสดง Process Specification 35.1 เเสดงช่องโหว่ของอุปกรณ์ทั้งหมด

|  |  |
| --- | --- |
| Process Number : | 35.1 |
| Process Name: | แสดงช่องโหว่ของอุปกรณ์ทั้งหมด |
| Description : | กระบวนการนี้จะรับข้อมูลจากการตรวจสอบความปลอดภัยของอุปกรณ์ทั้งหมดที่ได้รับจากกระบวนการก่อนหน้านี้ (เช่น 33.1 และ 34.1) โดยจะแสดงผลช่องโหว่ที่พบในแต่ละอุปกรณ์ ผ่านหน้าต่างของเว็บเพจ เพื่อให้ผู้ใช้สามารถดูคำเตือนและข้อมูลล่าสุดเกี่ยวกับช่องโหว่ของอุปกรณ์แต่ละตัว |
| Input Data Flow : | ข้อมูลการวิเคราะห์จากอุปกรณ์ทั้งหมดที่ได้จากฐานข้อมูล D1 |
| Output Data Flow : | ข้อมูลที่แสดงบนหน้าต่างเว็บเพจ โดยมีการแสดงชื่ออุปกรณ์, คำเตือนด้านความปลอดภัยที่พบ, และวันที่อัปเดตล่าสุด |
| Process Type : | Online Processing |
| Process Logic : | กระบวนการนี้จะดึงข้อมูลจากฐานข้อมูล (MongoDB) ซึ่งประกอบด้วยชื่ออุปกรณ์และผลการวิเคราะห์ช่องโหว่ที่ทำโดยกระบวนการก่อนหน้า จากนั้นจะใช้ข้อมูลเหล่านี้ในการแสดงผลบนหน้าเว็บ โดยแสดงรายชื่ออุปกรณ์, คำเตือนด้านความปลอดภัยที่พบ (หากมี) และวันที่อัปเดตล่าสุด |

**ตารางที่ 3-76** แสดง Process Specification 36.1 ตรวจสอบรายละเอียดอุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 36.1 |
| Process Name: | ตรวจสอบรายละเอียดอุปกรณ์ |
| Description : | กระบวนการนี้เป็นโมดูลหลักสำหรับดึงข้อมูลรายละเอียดของอุปกรณ์ผ่าน SNMP โดยมีการเรียกใช้โมดูลย่อย 36.2 ถึง 36.7 เพื่อดึงข้อมูลที่เกี่ยวข้อง เช่น ประเภทอุปกรณ์, รายละเอียดพอร์ต, ชื่อระบบ, สถานที่ตั้ง, ข้อมูลผู้ดูแล และระยะเวลาการทำงาน (Uptime) |
| Input Data Flow : | ค่าที่ได้รับจากผู้ใช้ เช่น IP Address ของอุปกรณ์ที่ต้องการตรวจสอบ |
| Output Data Flow : | ข้อมูล SNMP ที่ดึงมาได้ เช่น ประเภทอุปกรณ์, รายละเอียดพอร์ต, ชื่อระบบ, สถานที่ตั้ง, ข้อมูลผู้ดูแล และระยะเวลาการทำงานของอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | Process 36.1 จะเรียกใช้โมดูลย่อยทั้งหมด ได้แก่ การตรวจสอบ Description ของอุปกรณ์, ดึงข้อมูล Interfaces ทั้งหมด, ตรวจสอบ Uptime, Location, Contact และ System Name (SysName) เพื่อนำข้อมูลเหล่านี้มาประมวลผลและแสดงผลรายละเอียดของอุปกรณ์ในหน้าเว็บ device\_details\_snmp.html |

**ตารางที่ 3-77** แสดง Process Specification 36.2 ตรวจสอบ System Description ของอุปกรณ์

|  |  |
| --- | --- |
| Process Number : | 36.2 |
| Process Name: | ตรวจสอบ System Description ของอุปกรณ์ |
| Description : | รับคำขอจาก Process 36.1 เพื่อดึงข้อมูลรายละเอียดอุปกรณ์ผ่าน SNMP และส่งผลลัพธ์กลับไปยัง Process 36.1 |
| Input Data Flow : | คำขอ SNMP จาก Process 36.1 พร้อมข้อมูล IP Address ของอุปกรณ์ |
| Output Data Flow : | ข้อมูล System Description ของอุปกรณ์ที่ได้จาก SNMP |
| Process Type : | Online Processing |
| Process Logic : | ดึงค่า OID สำหรับ System Description (1.3.6.1.2.1.1.1.0) จากอุปกรณ์เป้าหมายโดยใช้ SNMP |

**ตารางที่ 3-78** แสดง Process Specification 36.3 ค้นหา Interfaces ทั้งหมด

|  |  |
| --- | --- |
| Process Number : | 36.3 |
| Process Name: | ค้นหา Interfaces ทั้งหมด |
| Description : | รับคำขอจาก Process 36.1 เพื่อดึงข้อมูลรายละเอียดอุปกรณ์ผ่าน SNMP และส่งผลลัพธ์กลับไปยัง Process 36.1 |
| Input Data Flow : | คำขอ SNMP จาก Process 36.1 พร้อมข้อมูล IP Address ของอุปกรณ์และประเภทของอุปกรณ์ (Layer 2, Layer 3, หรือ Router) |
| Output Data Flow : | ข้อมูลพอร์ตทั้งหมดของอุปกรณ์ที่ได้จาก SNMP |
| Process Type : | Online Processing |
| Process Logic : | ตรวจสอบประเภทของอุปกรณ์ที่ได้รับ จากนั้นดำเนินการสแกนพอร์ตผ่าน SNMP เพื่อดึงข้อมูลรายละเอียดทั้งหมดของพอร์ตที่มีในอุปกรณ์ และส่งผลลัพธ์กลับไปยัง Process 36.1 เพื่อดำเนินการต่อ |

**ตารางที่ 3-79** แสดง Process Specification 36.4 ค้นหาค่า uptime

|  |  |
| --- | --- |
| Process Number : | 36.4 |
| Process Name: | ค้นหาค่า Uptime ของอุปกรณ์ |
| Description : | รับคำขอจาก Process 36.1 เพื่อดึงข้อมูลรายละเอียดอุปกรณ์ผ่าน SNMP และส่งผลลัพธ์กลับไปยัง Process 36.1 |
| Input Data Flow : | คำขอพร้อมข้อมูล IP Address ของอุปกรณ์ที่ส่งมาจาก Process 36.1 |
| Output Data Flow : | ข้อมูลค่า Uptime ของอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | ทำการเชื่อมต่อกับอุปกรณ์ผ่าน SNMP โดยใช้ข้อมูลที่ได้รับ เพื่อค้นหาค่า Uptime ของระบบ และส่งค่าที่ค้นหาได้กลับไปยัง Process 36.1 เพื่อใช้งานต่อ |

**ตารางที่ 3-80** แสดง Process Specification 36.5 ค้นหาค่า location

|  |  |
| --- | --- |
| Process Number : | 36.5 |
| Process Name: | ค้นหาค่า Location ของอุปกรณ์ |
| Description : | รับคำขอจาก Process 36.1 เพื่อดึงข้อมูลรายละเอียดอุปกรณ์ผ่าน SNMP และส่งผลลัพธ์กลับไปยัง Process 36.1 |
| Input Data Flow : | คำขอพร้อมข้อมูล IP Address ของอุปกรณ์ที่ส่งมาจาก Process 36.1 |
| Output Data Flow : | ข้อมูล Location ของอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | ทำการเชื่อมต่อกับอุปกรณ์ผ่าน SNMP โดยใช้ข้อมูล IP Address ที่ได้รับมาเพื่อค้นหาค่า Location ผ่าน Object Identifier (OID) ที่กำหนด หากไม่พบค่า Location หรืออุปกรณ์ยังไม่ได้ตั้งค่า ระบบจะแสดงข้อความแจ้งเตือนพร้อมส่งผลลัพธ์กลับไปยัง Process 36.1 |

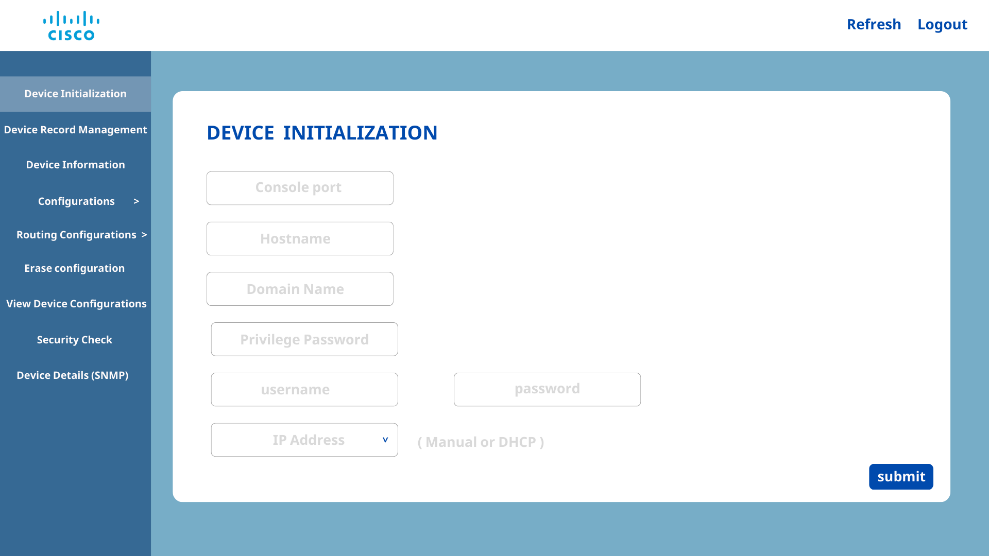
**ตารางที่ 3-81** แสดง Process Specification 36.6 ค้นหาค่า Contact

|  |  |
| --- | --- |
| Process Number : | 36.6 |
| Process Name: | ค้นหาค่า Contact ของอุปกรณ์ |
| Description : | รับคำขอจาก Process 36.1 เพื่อดึงข้อมูลรายละเอียดอุปกรณ์ผ่าน SNMP และส่งผลลัพธ์กลับไปยัง Process 36.1 |
| Input Data Flow : | คำขอพร้อมข้อมูล IP Address ของอุปกรณ์ที่ส่งมาจาก Process 36.1 |
| Output Data Flow : | ข้อมูล Contact ของอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | เชื่อมต่อกับอุปกรณ์ผ่าน SNMP โดยใช้ข้อมูล IP Address ที่ได้รับมาเพื่อค้นหาค่า Contact ผ่าน Object Identifier (OID) ที่กำหนด หากไม่พบค่า Contact หรืออุปกรณ์ยังไม่ได้ตั้งค่า ระบบจะแสดงข้อความแจ้งเตือนพร้อมส่งผลลัพธ์กลับไปยัง Process 36.1 |

**ตารางที่ 3-82** แสดง Process Specification 36.7 ค้นหาค่า System Name

|  |  |
| --- | --- |
| Process Number : | 36.7 |
| Process Name: | ค้นหาค่า System Name ของอุปกรณ์ |
| Description : | รับคำขอจาก Process 36.1 เพื่อดึงข้อมูลรายละเอียดอุปกรณ์ผ่าน SNMP และส่งผลลัพธ์กลับไปยัง Process 36.1 |
| Input Data Flow : | คำขอพร้อมข้อมูล IP Address ของอุปกรณ์ที่ส่งมาจาก Process 36.1 |
| Output Data Flow : | ข้อมูล System Name ของอุปกรณ์ |
| Process Type : | Online Processing |
| Process Logic : | เชื่อมต่อกับอุปกรณ์ผ่าน SNMP โดยใช้ข้อมูล IP Address ที่ได้รับมาเพื่อค้นหาค่า System Name ผ่าน Object Identifier (OID) ที่กำหนด หากไม่พบค่า System Name หรืออุปกรณ์ยังไม่ได้ตั้งค่า ระบบจะแสดงข้อความแจ้งเตือนพร้อมส่งผลลัพธ์กลับไปยัง Process 36.1 |

* 1. **การออกแบบอินเทอร์เฟซผู้ใช้ของเว็บแอปพลิเคชัน**



**ภาพที่ 3-29** หน้าการเริ่มต้นอุปกรณ์

A screenshot of a computer

Description automatically generated

**ภาพที่ 3-30** หน้าแสดงรายละเอียดอุปกรณ์

A screenshot of a computer

Description automatically generated

**ภาพที่ 3-31** หน้าเพิ่มข้อมูลเข้าสู่ฐานข้อมูล

A screenshot of a computer

Description automatically generated

**ภาพที่ 3-32** แสดงหัวข้อย่อยการตั้งค่าอุปกรณ์

A screenshot of a login screen

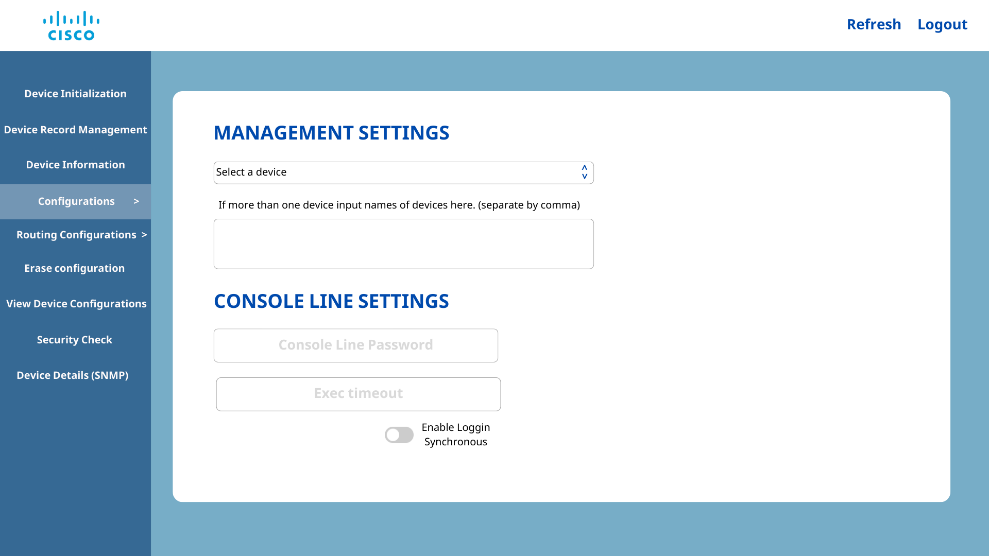
Description automatically generated

**ภาพที่ 3-33** หน้าการตั้งค่าพื้นฐาน

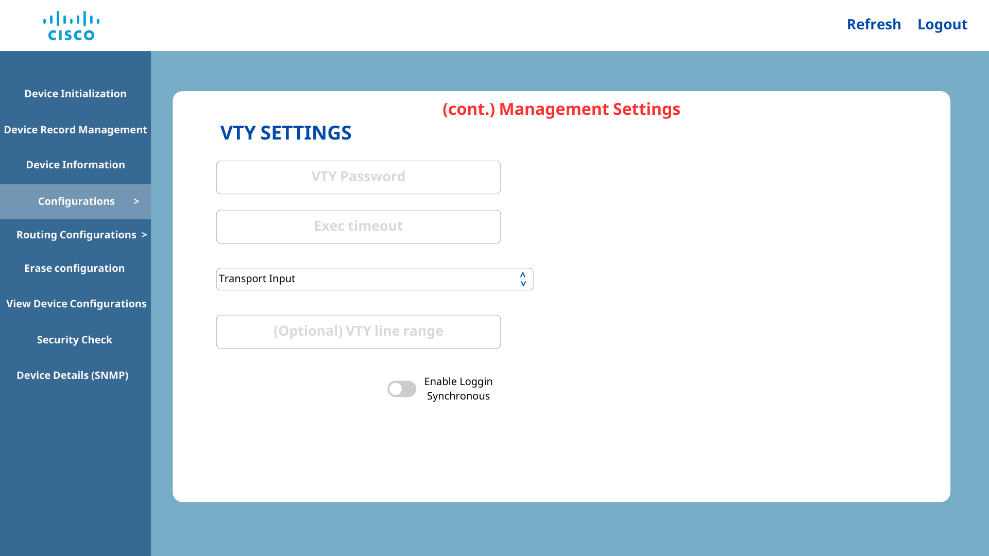
A screen shot of a computer

Description automatically generated

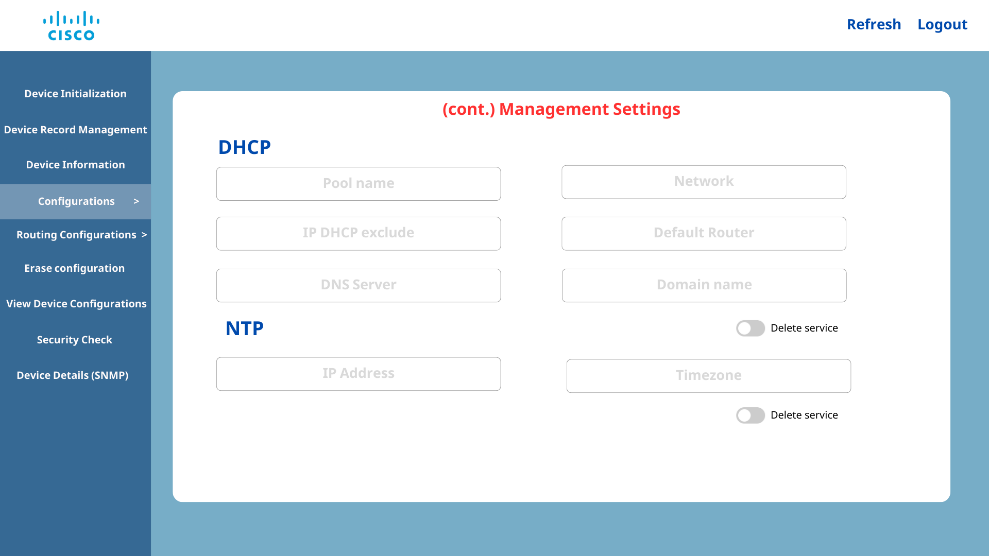
**ภาพที่ 3-34** หน้าการตั้งค่า Network Interface



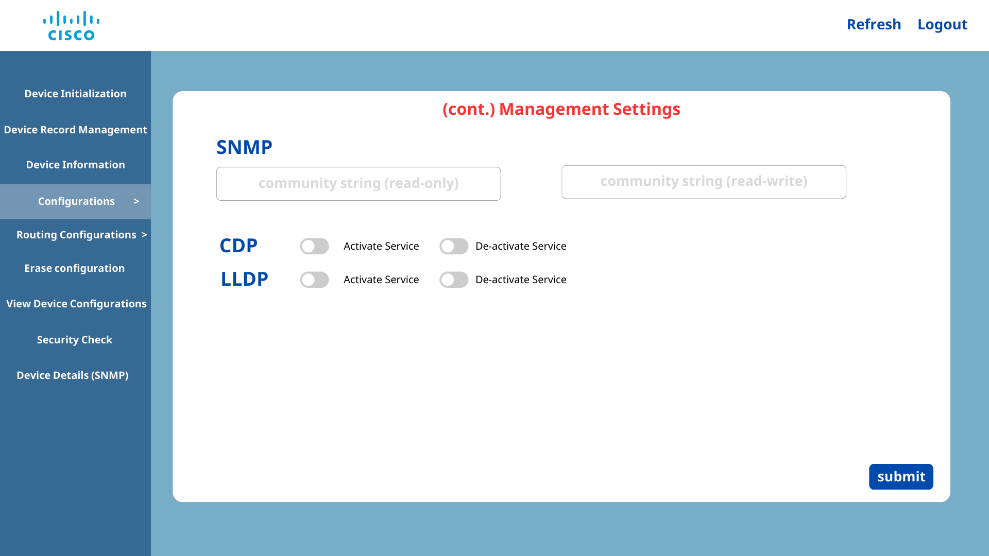
**ภาพที่ 3-35** หน้าการตั้งค่า Management



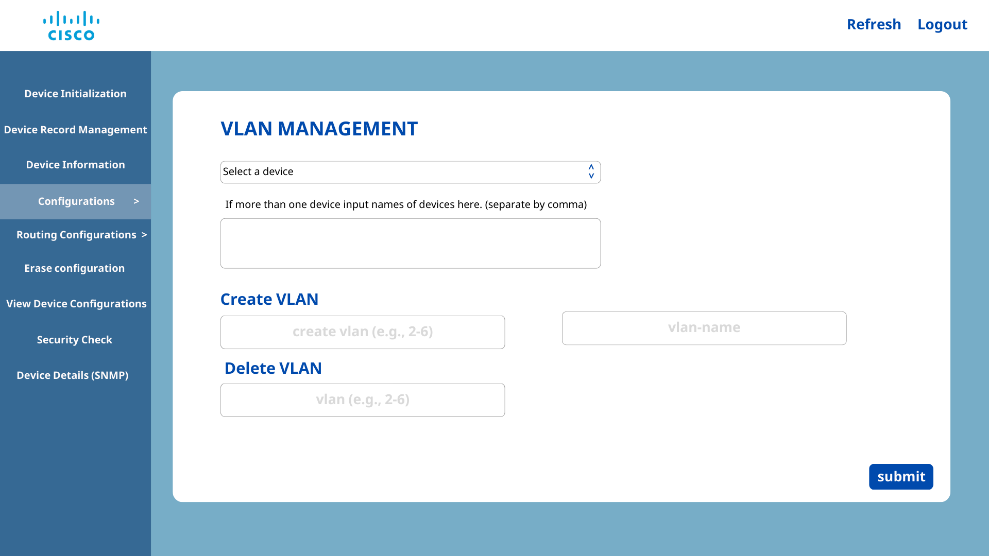
**ภาพที่ 3-36** หน้าการตั้งค่า VTY



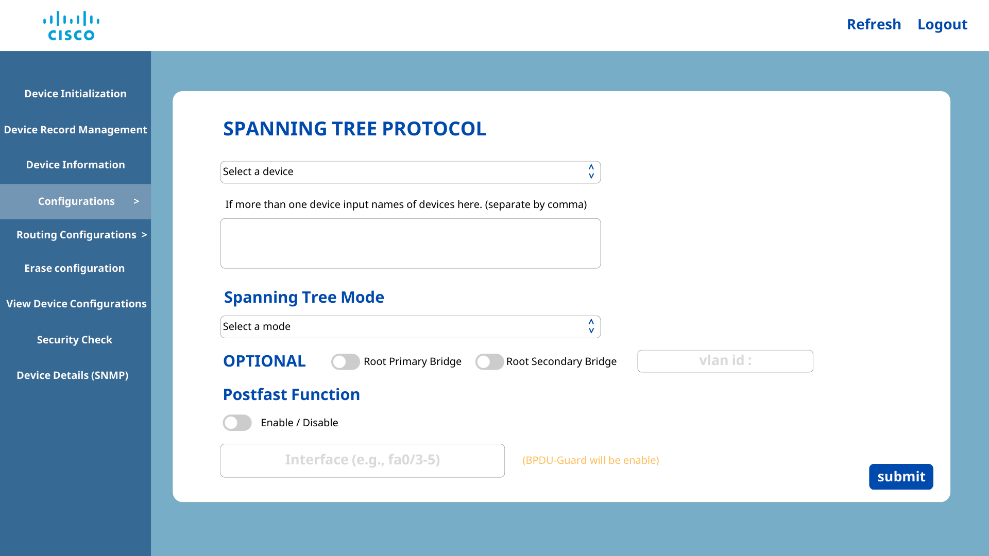
**ภาพที่ 3-37** หน้าการตั้งค่า DHCP และ NTP



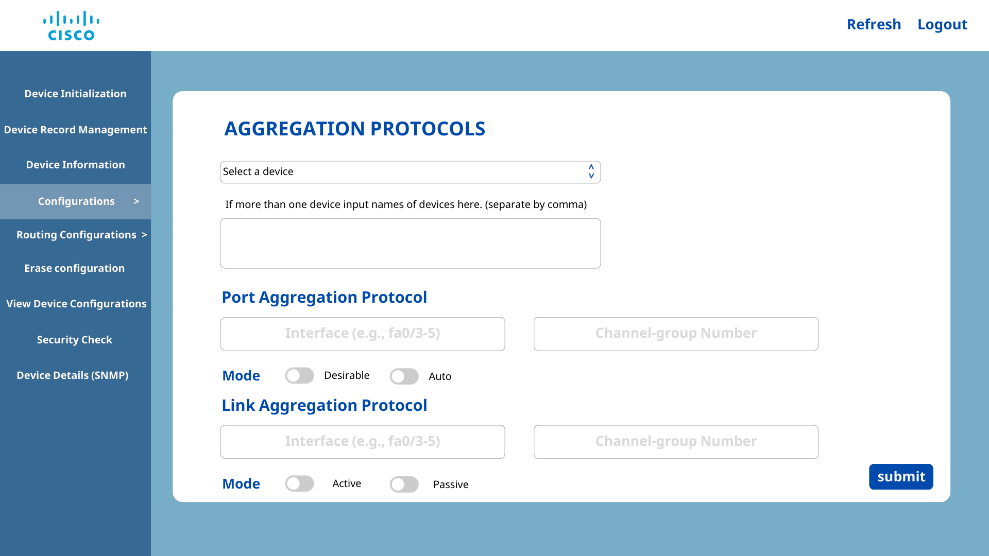
**ภาพที่ 3-38** หน้าการตั้งค่า SNMP, CDP, และ LLDP



**ภาพที่ 3-39** หน้าการตั้งค่า VLAN



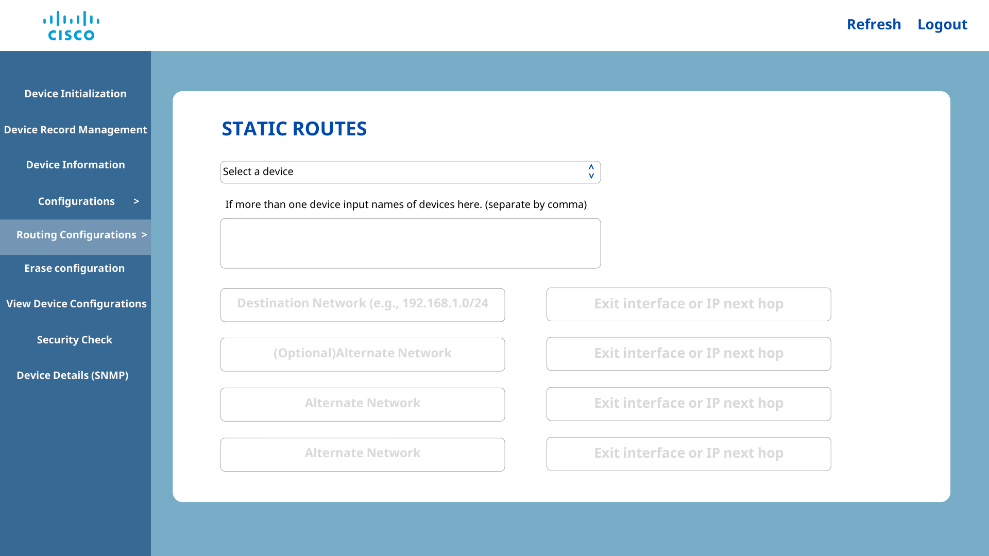
**ภาพที่ 3-40** หน้าการตั้งค่า Spanning Tree



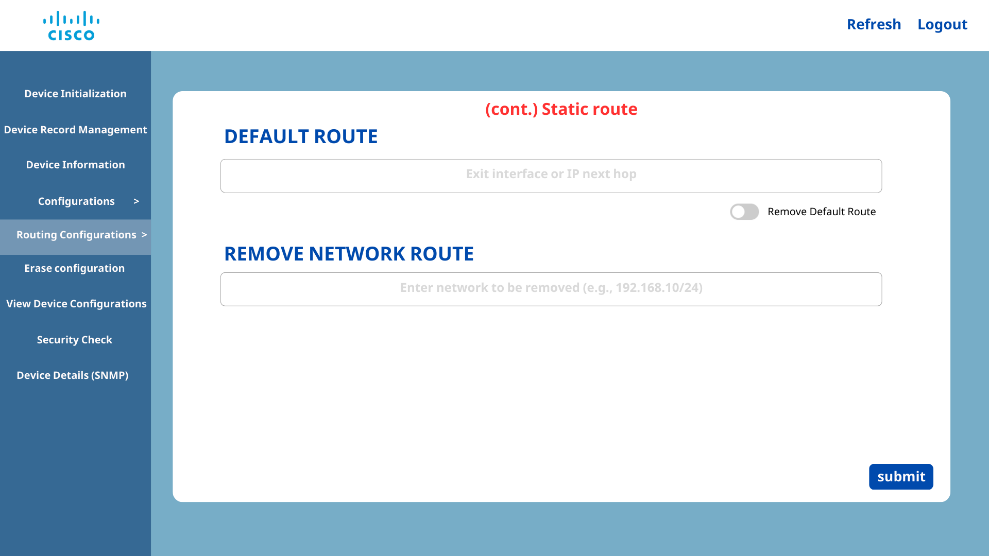
**ภาพที่ 3-41** หน้าการตั้งค่า Aggregation



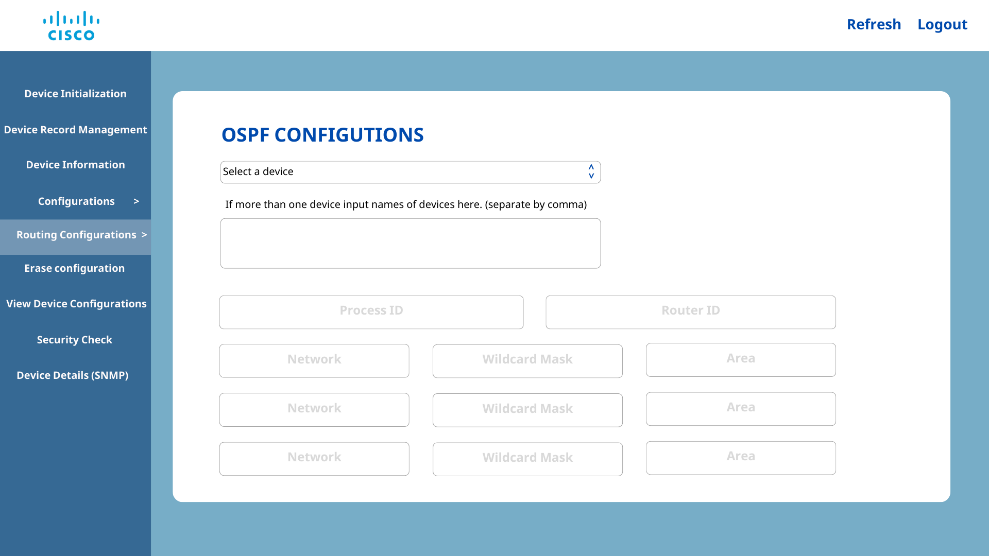
**ภาพที่ 3-42** หน้าแสดงหัวข้อย่อยการตั้งค่า Routing



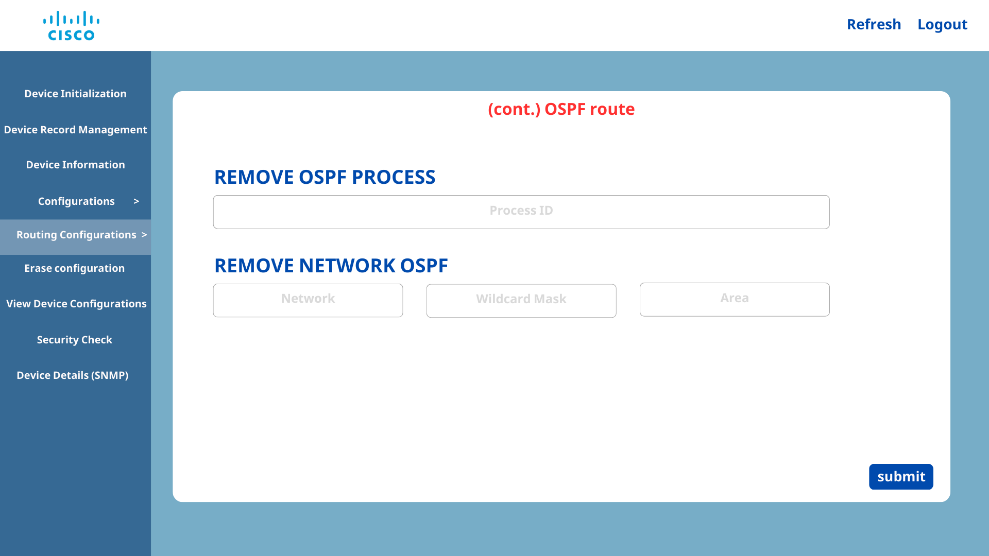
**ภาพที่ 3-43** หน้ากำหนดเส้นทางแบบ Static Route



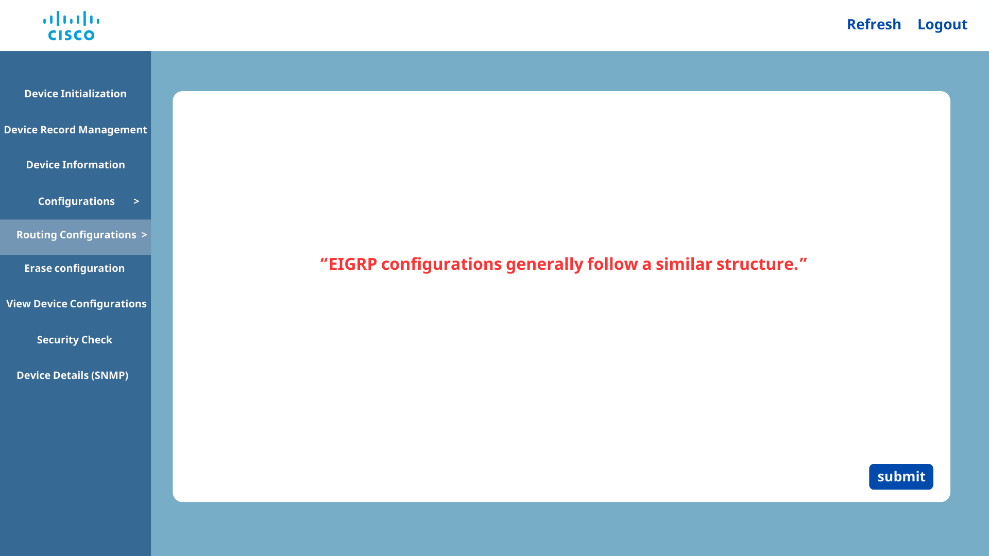
**ภาพที่ 3-44** หน้ากำหนดเส้นทางแบบ Static Route (ต่อ)



**ภาพที่ 3-45** หน้ากำหนดเส้นทางแบบ OSPF



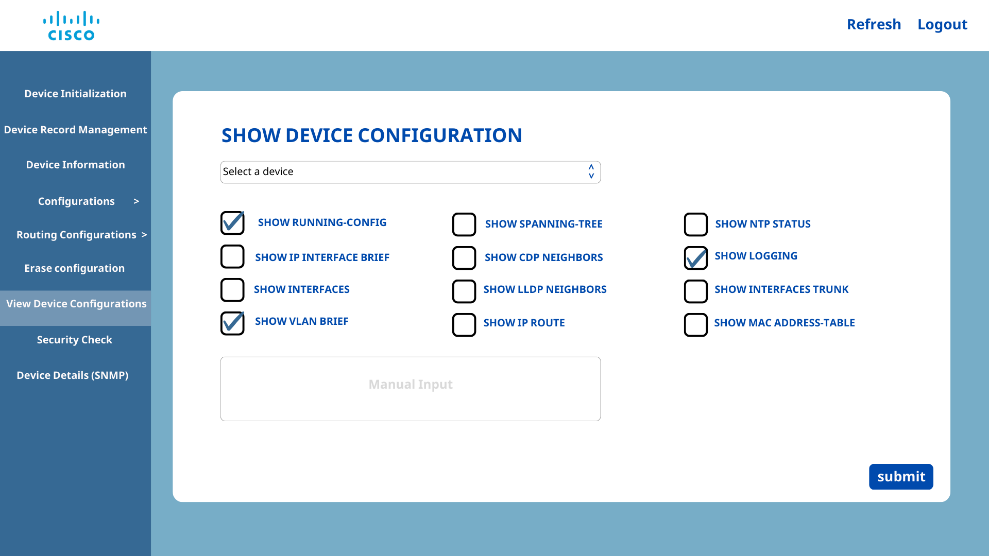
**ภาพที่ 3-46** หน้ากำหนดเส้นทางแบบ OSPF (ต่อ)



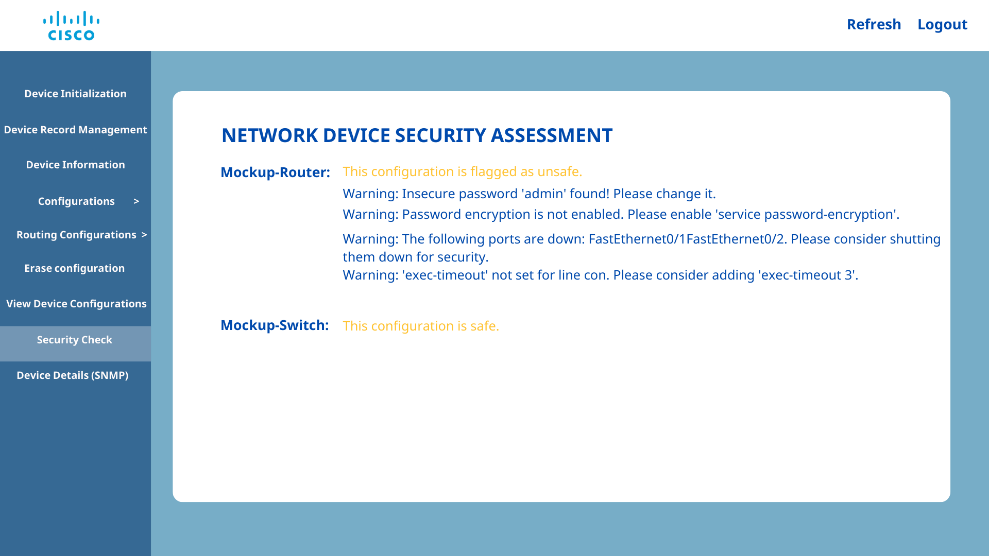
**ภาพที่ 3-46** หน้ากำหนดเส้นทางแบบ EIGRP



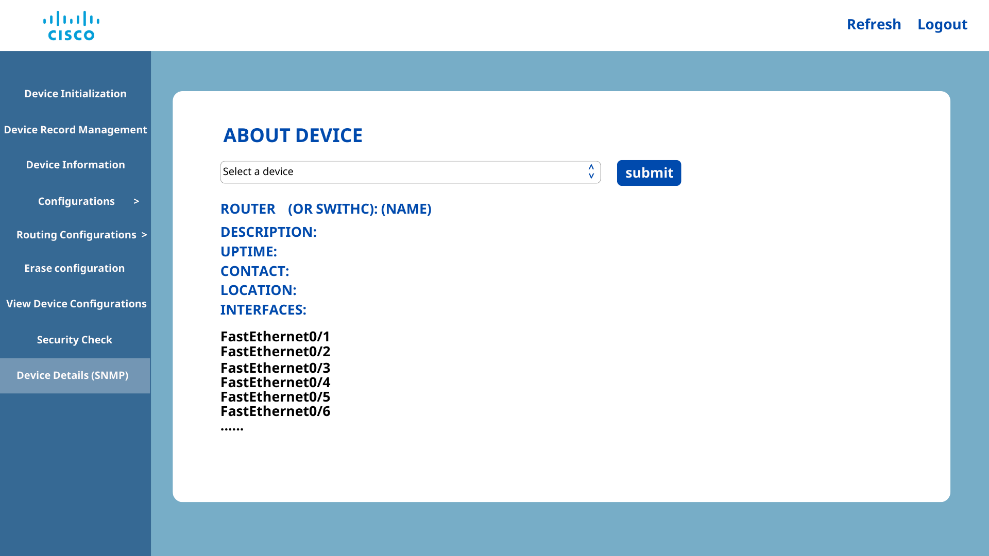
**ภาพที่ 3-47** หน้าการรีเซ็ตการตั้งค่าอุปกรณ์



**ภาพที่ 3-48** หน้าดึงข้อมูลการตั้งค่าของอุปกรณ์



**ภาพที่ 3-49** หน้าแสดงช่องโหว่อุปกรณ์



**ภาพที่ 3-50** หน้าแสดงข้อมูลเบื้องต้นนั้นๆของอุปกรณ์

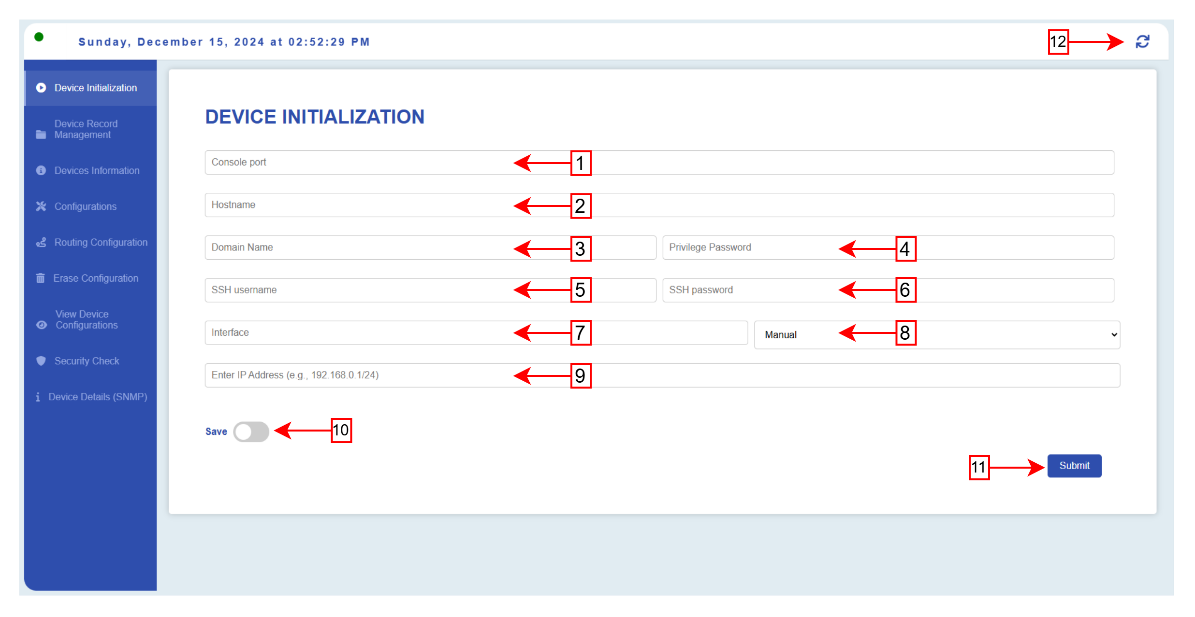
**บทที่ 4**

**ผลการดำเนินงาน**

* 1. **ผลการดำเนินงาน**

รายละเอียดการทำงานของโปรแกรม

ผลการดำเนินงานในการจัดทำโครงงานเว็บแอปพลิเคชันกำหนดค่าอุปกรณ์และตรวจสอบช่องโหว่ของการกำหนดค่าอุปกรณ์เครือข่าย CISCO มีรายละเอียดดังนี้



**ภาพที่ 4–1** หน้าจอการเริ่มต้นอุปกรณ์

จากภาพที่ 4-1 แสดงหน้าจอสำหรับการตั้งค่าเริ่มต้นของอุปกรณ์ในระบบ ซึ่งประกอบด้วยฟิลด์ต่างๆ สำหรับกรอกข้อมูลเริ่มต้น ดังนี้

หมายเลข 1 ช่องสำหรับกรอกพอร์ตคอนโซล (Console Port) ที่ใช้ในการเชื่อมต่อกับอุปกรณ์

หมายเลข 2 ช่องสำหรับกรอกชื่อโฮสต์ (Hostname) ของอุปกรณ์ที่จะตั้งค่า

หมายเลข 3 ช่องสำหรับกรอกชื่อโดเมน (Domain Name) ที่อุปกรณ์จะใช้งาน

หมายเลข 4 ช่องสำหรับกรอกรหัสผ่านที่ใช้ในการเข้าโหมดพิเศษ (Privilege Password)

หมายเลข 5 ช่องสำหรับกรอกรหัสผ่าน SSH (SSH Password) ที่ใช้ในการเชื่อมต่อ SSH

หมายเลข 6 ช่องสำหรับกรอกชื่อผู้ใช้ SSH (SSH Username)

หมายเลข 7 ช่องสำหรับเลือกประเภทของอินเตอร์เฟซที่อุปกรณ์จะใช้งาน

หมายเลข 8 ช่องสำหรับเลือกวิธีการตั้งค่า IP ที่จะใช้ ซึ่งสามารถเลือกได้ระหว่าง Manual หรือ DHCP (รับ IP แบบอัตโนมัติ

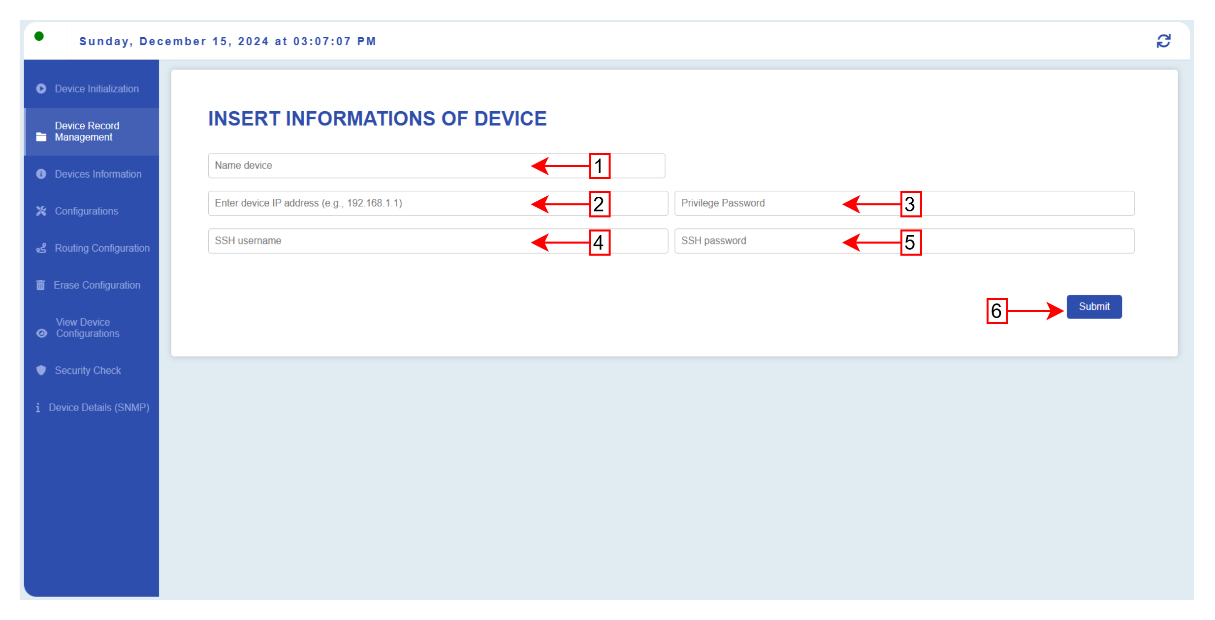
หมายเลข 9 ช่องสำหรับกรอกที่อยู่ IP ที่จะใช้กับอุปกรณ์

หมายเลข 10 สวิตช์เปิด/ปิดสำหรับการบันทึกการตั้งค่าที่ได้ทำไว้

หมายเลข 11 ปุ่ม "Submit" สำหรับบันทึกการตั้งค่าเริ่มต้นที่กรอกไว้ในแบบฟอร์ม

หมายเลข 12 ปุ่มรีเฟรช (Refresh) สำหรับโหลดหน้าจอใหม่

ส่วนโค้ดที่ใช้ในระบบเพื่อรับค่าจากฟอร์มการตั้งค่าเริ่มต้นนี้จะดำเนินการรับค่าและตรวจสอบการตั้งค่า โดยเฉพาะการตรวจสอบการใช้งาน IP และ Hostname ว่าไม่ซ้ำกับอุปกรณ์ที่มีอยู่ในฐานข้อมูล จากนั้นบันทึกข้อมูลเข้าสู่ฐานข้อมูลและทำการตั้งค่าอุปกรณ์ตามที่ระบุในฟอร์ม



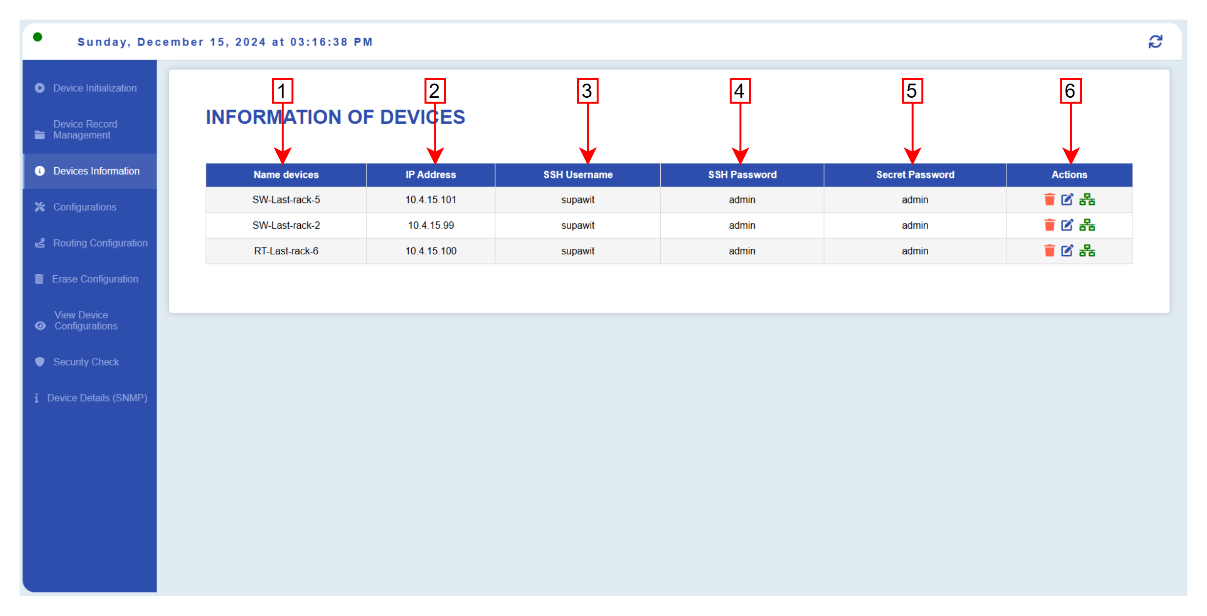
**ภาพที่ 4–2** หน้าจอการเพิ่มอุปกรณ์

จากภาพที่ 4-2 แสดงหน้าจอสำหรับจัดการข้อมูลบันทึกอุปกรณ์ในระบบ ซึ่งจะช่วยให้ผู้ใช้สามารถเพิ่มข้อมูลของอุปกรณ์ใหม่ได้ ผ่านการกรอกฟอร์มที่ประกอบด้วยฟิลด์ต่างๆ ดังนี้

หมายเลข 1 ช่องสำหรับกรอกชื่อของอุปกรณ์ (Device Name) ที่จะบันทึก

หมายเลข 2 ช่องสำหรับกรอกที่อยู่ IP ของอุปกรณ์ (IP Address)  
หมายเลข 3 ช่องสำหรับกรอกรหัสผ่านโหมดพิเศษ (Privilege Password) ของอุปกรณ์  
หมายเลข 4 ช่องสำหรับกรอกชื่อผู้ใช้ SSH (SSH Username) ที่จะใช้เชื่อมต่อกับอุปกรณ์  
หมายเลข 5 ช่องสำหรับกรอกรหัสผ่าน SSH (SSH Password) สำหรับการเชื่อมต่อ SSH  
หมายเลข 6 ปุ่ม "Submit" สำหรับบันทึกข้อมูลอุปกรณ์ที่กรอกไว้ในฟอร์ม

เมื่อกรอกข้อมูลเสร็จแล้ว ผู้ใช้สามารถกดปุ่ม "Submit" เพื่อบันทึกข้อมูลอุปกรณ์ใหม่ ระบบจะตรวจสอบว่าชื่อโฮสต์ (Hostname) หรือที่อยู่ IP ซ้ำกับข้อมูลในฐานข้อมูลหรือไม่ หากมีข้อมูลซ้ำ ระบบจะแสดงข้อความแจ้งเตือนให้ผู้ใช้เลือกชื่อหรือ IP ที่แตกต่างกัน หากไม่มีข้อผิดพลาด ข้อมูลจะถูกบันทึกลงในฐานข้อมูล และผู้ใช้จะได้รับข้อความยืนยันว่าอุปกรณ์ได้ถูกเพิ่มเรียบร้อยแล้ว



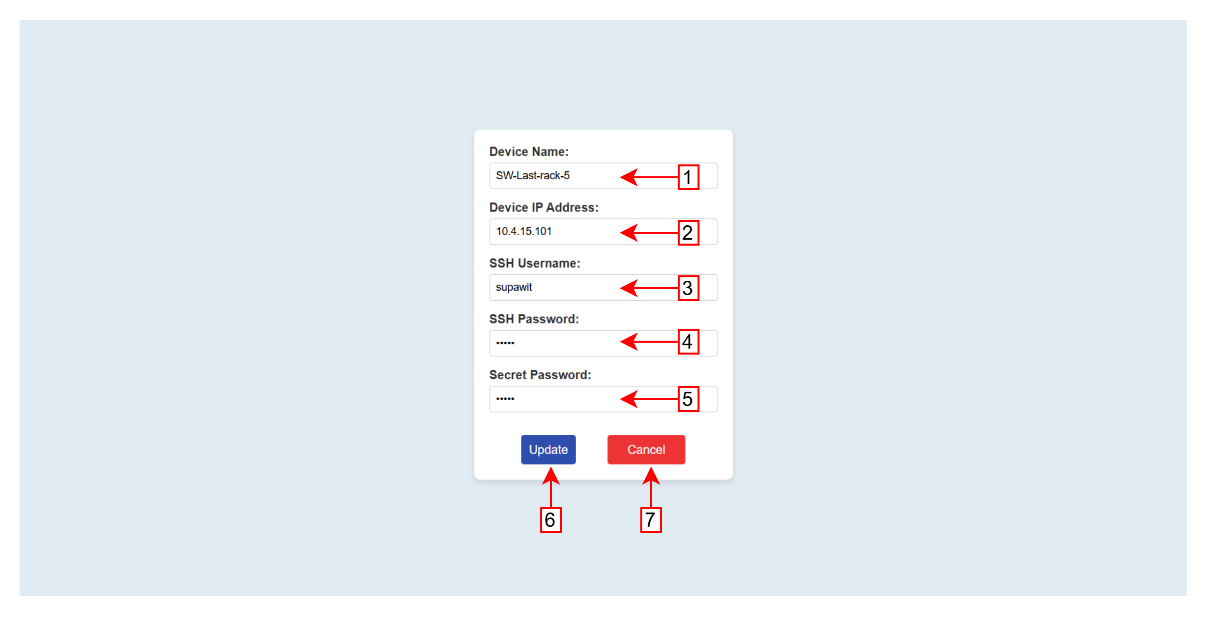
**ภาพที่ 4–3** หน้าจอแสดงรายละเอียดอุปกรณ์

จากภาพที่ 4-3 แสดงหน้าจอที่ใช้สำหรับแสดงรายละเอียดอุปกรณ์ในระบบ โดยประกอบด้วยข้อมูลของอุปกรณ์ต่างๆ พร้อมกับปุ่มการกระทำที่สามารถเลือกได้ดังนี้

หมายเลข 1 แสดงชื่ออุปกรณ์ (Name devices)  
หมายเลข 2 แสดงที่อยู่ IP ของอุปกรณ์ (IP Address)  
หมายเลข 3 แสดงชื่อผู้ใช้ SSH (SSH Username)  
หมายเลข 4 แสดงรหัสผ่าน SSH (SSH Password)  
หมายเลข 5 แสดงรหัสผ่านสำหรับเข้าสู่โหมดพิเศษ (Secret Password)  
หมายเลข 6 ปุ่มการกระทำ (Actions) ซึ่งรวมถึง:

* ปุ่มลบ (Delete) สำหรับลบอุปกรณ์ออกจากระบบ โดยจะมีการยืนยันก่อนลบ
* ปุ่มแก้ไข (Edit) สำหรับแก้ไขข้อมูลของอุปกรณ์ เช่น ชื่อ, IP, หรือข้อมูลการเข้าสู่ระบบ
* ปุ่มปิง (Ping) สำหรับทดสอบการเชื่อมต่ออุปกรณ์ผ่านคำสั่ง ping เพื่อตรวจสอบสถานะการเชื่อมต่อ

ฟอร์มจะตรวจสอบข้อมูลที่กรอกเพื่อทำการบันทึกหรือแก้ไขข้อมูลของอุปกรณ์ รวมถึงการทดสอบการเชื่อมต่อโดยใช้คำสั่ง ping หากไม่สามารถเชื่อมต่อได้จะมีการแจ้งเตือนถึงข้อผิดพลาด



**ภาพที่ 4–4** หน้าจอแสดงการแก้ไขข้อมูลอุปกรณ์

จากภาพที่ 4-4 แสดงหน้าจอสำหรับการแก้ไขข้อมูลบันทึกของอุปกรณ์เครือข่ายในระบบ ซึ่งช่วยให้ผู้ใช้สามารถปรับปรุงหรือแก้ไขข้อมูลที่เกี่ยวข้องกับอุปกรณ์ได้ ผ่านการกรอกฟอร์มที่ประกอบด้วยฟิลด์ต่างๆ ดังนี้

หมายเลข 1 ช่องสำหรับแก้ไขชื่อของอุปกรณ์ (Device Name) ที่จะบันทึกในระบบ  
หมายเลข 2 ช่องสำหรับแก้ไขที่อยู่ IP ของอุปกรณ์ (Device IP Address) ซึ่งระบบจะตรวจสอบไม่ให้ซ้ำกับอุปกรณ์อื่นที่มีอยู่

หมายเลข 3 ช่องสำหรับแก้ไขชื่อผู้ใช้ SSH (SSH Username) ที่ใช้สำหรับการเชื่อมต่อกับอุปกรณ์

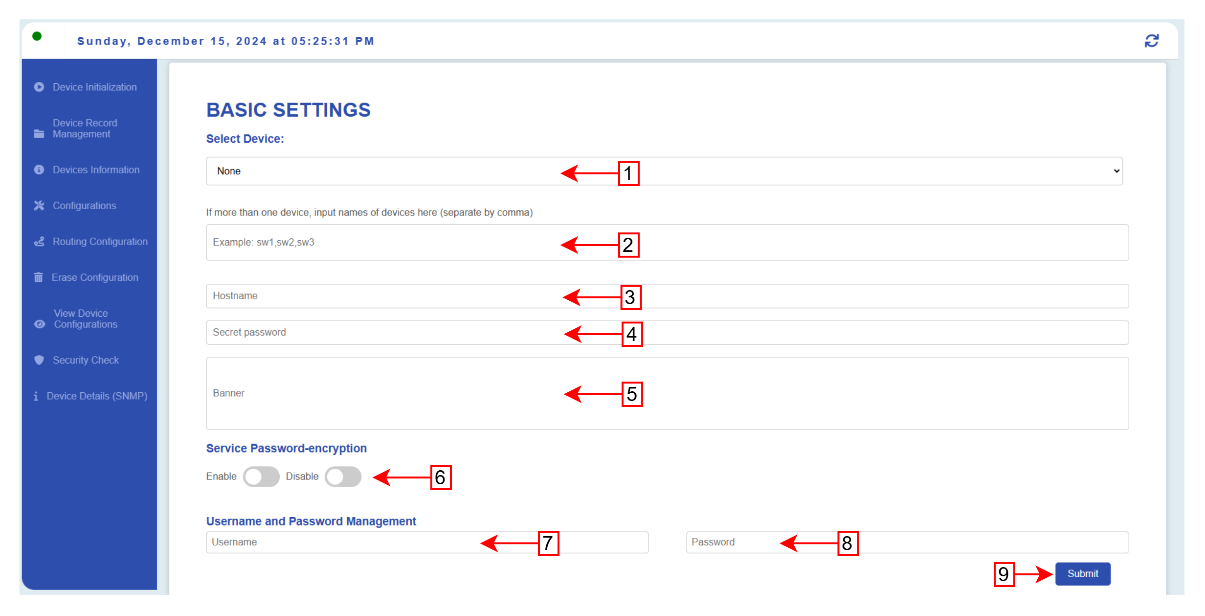
หมายเลข 4 ช่องสำหรับแก้ไขรหัสผ่าน SSH (SSH Password) ซึ่งใช้สำหรับการยืนยันตัวตนผ่านการเชื่อมต่อ SSH

หมายเลข 5 ช่องสำหรับแก้ไขรหัสผ่านโหมดพิเศษ (Secret Password) ที่ใช้สำหรับเข้าถึงโหมดผู้ดูแลระบบของอุปกรณ์

หมายเลข 6 ปุ่ม "Update" สำหรับบันทึกข้อมูลที่แก้ไขแล้วลงในฐานข้อมูล

หมายเลข 7 ปุ่ม "Cancel" สำหรับยกเลิกการแก้ไขข้อมูลและกลับไปยังหน้าจอแสดงข้อมูล

เมื่อผู้ใช้ทำการแก้ไขข้อมูลเสร็จสิ้นและกดปุ่ม "Update" ระบบจะตรวจสอบว่า ชื่ออุปกรณ์หรือที่อยู่ IP มีการซ้ำกับข้อมูลอื่นในฐานข้อมูลหรือไม่ หากพบข้อมูลซ้ำ ระบบจะแสดงข้อความแจ้งเตือนให้ผู้ใช้เลือกค่าที่เหมาะสมใหม่อีกครั้ง ในกรณีที่ไม่มีข้อผิดพลาด ข้อมูลที่แก้ไขจะถูกอัปเดตลงในฐานข้อมูลสำเร็จ และระบบจะนำผู้ใช้กลับไปยังหน้าจอแสดงข้อมูลอุปกรณ์เพื่อยืนยันการเปลี่ยนแปลงที่เกิดขึ้น



**ภาพที่ 4-5** แสดงหน้าจอสำหรับการตั้งค่าพื้นฐานของอุปกรณ์เครือข่าย

จากภาพที่ 4-5 แสดงหน้าจอสำหรับการตั้งค่าพื้นฐานของอุปกรณ์เครือข่าย ซึ่งช่วยให้ผู้ใช้สามารถกำหนดและบันทึกข้อมูลอุปกรณ์ต่าง ๆ ผ่านฟอร์มการตั้งค่าที่ประกอบด้วยฟิลด์ดังนี้

หมายเลข 1 ช่องสำหรับการเลือกอุปกรณ์ (Select Device) โดยผู้ใช้สามารถเลือกอุปกรณ์จากรายการที่มีอยู่ในฐานข้อมูล ระบบจะดึงข้อมูล IP และชื่ออุปกรณ์ขึ้นมาแสดงในตัวเลือก  
หมายเลข 2 ช่องสำหรับการป้อนชื่อหลายอุปกรณ์ (Many Hostname) โดยใช้เครื่องหมายจุลภาค (,) คั่นระหว่างชื่อ เช่น sw1,sw2,sw3

หมายเลข 3 ช่องสำหรับการป้อนชื่ออุปกรณ์ (Hostname) เพื่อกำหนดชื่อ (Hostname) อุปกรณ์ใหม่หรือเปลี่ยนชื่อที่มีอยู่

หมายเลข 4 ช่องสำหรับการป้อนรหัสผ่านลับ (Secret Password) ที่ใช้สำหรับการยืนยันการเข้าถึงโหมดผู้ดูแลระบบของอุปกรณ์

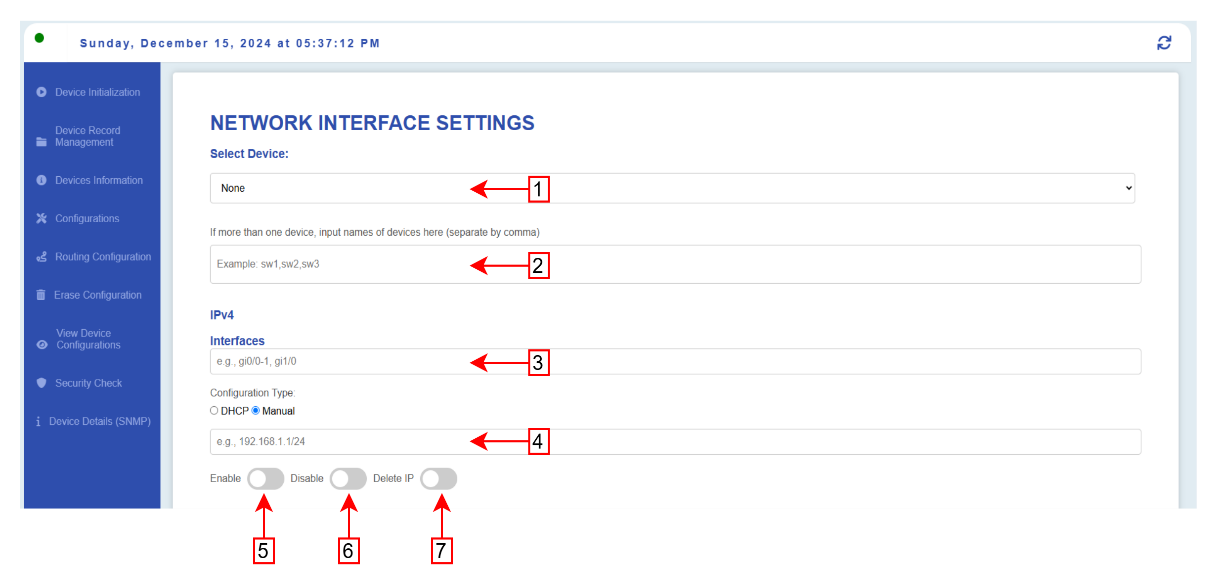
หมายเลข 5 ช่องสำหรับการป้อนข้อความแบนเนอร์ (Banner) ซึ่งจะแสดงข้อความต้อนรับหรือคำเตือนเมื่อมีการเข้าสู่ระบบ

หมายเลข 6 ตัวเลือกสำหรับการเปิดใช้งานหรือปิดใช้งานการเข้ารหัสรหัสผ่าน (Service Password-Encryption) ผู้ใช้สามารถเลือก Enable เพื่อเปิดใช้งาน หรือ Disable เพื่อปิดใช้งาน

หมายเลข 7 ช่องสำหรับป้อนชื่อผู้ใช้ (Username) ที่ใช้สำหรับการยืนยันตัวตนในกระบวนการเชื่อมต่อกับอุปกรณ์  
หมายเลข 8 ช่องสำหรับป้อนรหัสผ่าน (Password) ที่เชื่อมโยงกับชื่อผู้ใช้ที่กรอกไว้

หมายเลข 9 ปุ่ม "Submit" สำหรับยืนยันและบันทึกข้อมูลที่ป้อนไว้ลงในระบบ

เมื่อผู้ใช้กรอกข้อมูลในฟอร์มและกดปุ่ม "Submit" ระบบจะทำการตรวจสอบข้อมูล เช่น ชื่อโฮสต์ที่ซ้ำกันในฐานข้อมูล ที่อยู่ IP ของอุปกรณ์ และรหัสผ่านผู้ใช้ หากข้อมูลที่ป้อนไม่มีข้อผิดพลาด ระบบจะอัปเดตค่าที่กำหนดไปยังอุปกรณ์ที่เลือกไว้ และบันทึกข้อมูลลงในฐานข้อมูลพร้อมทั้งแจ้งผลลัพธ์ให้กับผู้ใช้ผ่านหน้าจอหรือข้อความแจ้งเตือน



**ภาพที่ 4-6** แสดงหน้าจอสำหรับการตั้งค่า Network Interfaces

จากโค้ดในภาพที่ 4-6 แสดงหน้าจอสำหรับการตั้งค่าพื้นฐานของเครือข่ายอินเทอร์เฟซ IPv4 ซึ่งช่วยให้ผู้ใช้สามารถกำหนดและบันทึกการตั้งค่าต่าง ๆ ผ่านฟอร์มที่มีองค์ประกอบดังนี้

หมายเลข 1 ช่องสำหรับการเลือกอุปกรณ์ (Select Device) โดยผู้ใช้สามารถเลือกอุปกรณ์จากรายการที่แสดง ซึ่งระบบจะดึงข้อมูลชื่ออุปกรณ์และ IP จากฐานข้อมูลขึ้นมาแสดงเป็นตัวเลือกให้เลือกใช้งานได้สะดวก

หมายเลข 2 ช่องสำหรับการป้อนชื่ออุปกรณ์หลายตัว (Many Hostname) เพื่อรองรับกรณีที่ต้องการกำหนดค่าพร้อมกันหลายอุปกรณ์ โดยใช้เครื่องหมายจุลภาค (,) คั่นระหว่างชื่อ เช่น sw1,sw2,sw3

หมายเลข 3 ส่วนการตั้งค่าอินเทอร์เฟซ IPv4 (Interfaces) ผู้ใช้สามารถป้อนชื่ออินเทอร์เฟซที่ต้องการ เช่น gi0/0-1, gi1/0 ซึ่งหมายถึงกลุ่มหรือช่วงของพอร์ตที่ต้องการกำหนดค่า

หมายเลข 4 ตัวเลือกสำหรับการเลือกประเภทการกำหนดค่า IP (Configuration Type) โดยมีตัวเลือกเป็น DHCP สำหรับการรับ IP โดยอัตโนมัติจากเซิร์ฟเวอร์ DHCP และ Manual สำหรับการกำหนด IP เอง ซึ่งจะต้องป้อนที่อยู่ IP ในรูปแบบ CIDR เช่น 192.168.1.1/24

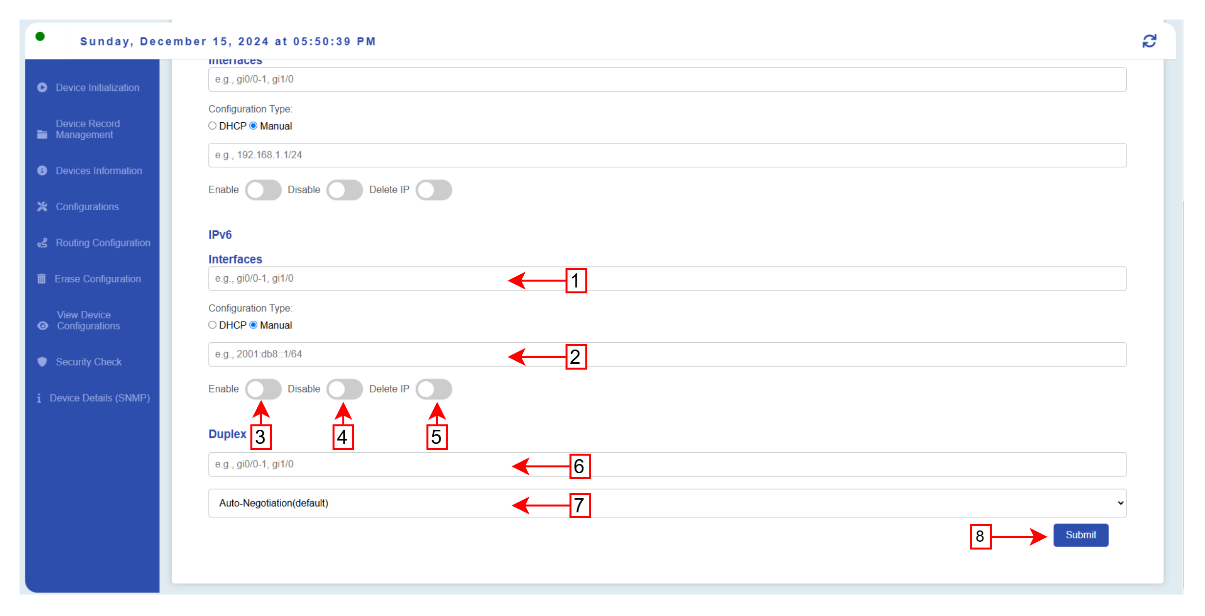
หมายเลข 5 ตัวเลือก Enable สำหรับเปิดใช้งานการเชื่อมต่อ IPv4 บนอุปกรณ์ที่กำหนดไว้

หมายเลข 6 ตัวเลือก Disable สำหรับปิดการใช้งาน IP ที่กำหนดไว้

หมายเลข 7 ตัวเลือก Delete IP สำหรับลบการกำหนด IP เดิมออกจากอินเทอร์เฟซ

หมายเลข 8 ปุ่ม "Submit" สำหรับยืนยันและบันทึกค่าที่ผู้ใช้ตั้งไว้ ระบบจะทำการประมวลผลข้อมูล ตรวจสอบความถูกต้องของ IP ที่ป้อน เช่น รูปแบบ CIDR หรือความซ้ำซ้อนของอินเทอร์เฟซในระบบ และดำเนินการอัปเดตการตั้งค่าไปยังอุปกรณ์เป้าหมาย

หลังจากกดปุ่ม "Submit" หากข้อมูลที่ป้อนถูกต้อง ระบบจะแจ้งผลลัพธ์การตั้งค่าให้ผู้ใช้ทราบ เช่น การเปิดใช้งาน IP สำเร็จหรือมีข้อผิดพลาดใด ๆ ผ่านข้อความแจ้งเตือนบนหน้าจอ



**ภาพที่ 4-7** แสดงหน้าจอสำหรับการตั้งค่า Network Interfaces (ต่อ)

จากภาพที่ 4-7 แสดงหน้าจอสำหรับการตั้งค่า Network Interfaces (ต่อ) ซึ่งในส่วนนี้เน้นการกำหนดค่าเครือข่าย IPv6 โดยผู้ใช้สามารถตั้งค่าอินเทอร์เฟซ IPv6 ได้ผ่านฟอร์มที่มีองค์ประกอบดังนี้

หมายเลข 1 ช่องสำหรับการป้อนชื่ออินเทอร์เฟซ (Interfaces) ที่ต้องการตั้งค่า เช่น gi0/0-1, gi1/0 ซึ่งหมายถึงกลุ่มหรือช่วงของพอร์ตที่ต้องการกำหนดค่า

หมายเลข 2 ตัวเลือกสำหรับการเลือกประเภทการกำหนดค่า IP (Configuration Type) โดยมีตัวเลือกเป็น DHCP สำหรับการรับ IP โดยอัตโนมัติจากเซิร์ฟเวอร์ DHCPv6 และ Manual สำหรับการกำหนด IP เอง ซึ่งจะต้องป้อนที่อยู่ IPv6 ในรูปแบบ CIDR เช่น 2001:db8::1/64

หมายเลข 3 ตัวเลือก Enable สำหรับเปิดใช้งานการเชื่อมต่อ IPv6 บนอุปกรณ์ที่กำหนดไว้

หมายเลข 4 ตัวเลือก Disable สำหรับปิดการใช้งาน IPv6 ที่กำหนดไว้

หมายเลข 5 ตัวเลือก Delete IP สำหรับลบการกำหนด IP เดิมออกจากอินเทอร์เฟซ

หมายเลข 6 ตัวเลือก Duplex สำหรับการตั้งค่าโหมด Duplex ของอินเทอร์เฟซ ซึ่งมีตัวเลือกเป็น Auto-Negotiation (ค่าเริ่มต้น), Full สำหรับการทำงานแบบเต็มความเร็วสองทิศทาง และ Half สำหรับการทำงานแบบความเร็วทิศทางเดียว

หมายเลข 7 ตัวเลือกสำหรับการตั้งค่า โหมดความเร็ว (Speed Mode) ซึ่งช่วยให้ผู้ใช้สามารถกำหนดการทำงานของอินเทอร์เฟซให้เหมาะสมกับความต้องการของเครือข่าย

หมายเลข 8 ปุ่ม "Submit" สำหรับยืนยันและบันทึกค่าที่ผู้ใช้ตั้งไว้ ระบบจะทำการตรวจสอบความถูกต้องของข้อมูล เช่น รูปแบบที่อยู่ IPv6 ความซ้ำซ้อนของอินเทอร์เฟซ หรือความขัดแย้งในการตั้งค่า ก่อนดำเนินการอัปเดตไปยังอุปกรณ์ที่เลือก

หลังจากผู้ใช้กดปุ่ม "Submit" ระบบจะแสดงผลการดำเนินการ เช่น การเปิดใช้งาน IPv6 สำเร็จ การปิดการใช้งาน การลบ IP หรือการปรับโหมด Duplex พร้อมแจ้งข้อความสถานะให้ผู้ใช้ทราบผ่านหน้าจอ