โครงงานเลขที่ วศ.คพ. **P005-2/2568**

เรื่อง

SCMC AssetManager : A Web Application for Centralized Digital Asset Management at Smart Campus Management Center

โดย

นายณัฐนันทน์ ตาตะนันทน์ รหัส **650610760**
นายบวรภัค ดวงจันทร์ รหัส **650610779**
นายพชร จารุรักษ์สกุล รหัส **650610789**
นายศิรวิชญ์ ศิรบรรจงกราน รหัส **650610810**

โครงงานนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา **2568**

PROJECT No. CPE P005-2/2568

เอสซีเอ็มซี แอสเซตแมนเนเจอร์ : เว็บแอปพลิเคชันสำหรับระบบบริหารจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ภายในศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน

Natanan Tatanan	**650610760**
Borwonpak Duangjun	**650610779**
Patchara Jaruraxsakul	**650610789**
Sirawit Sirabanchongkran	**650610810**

A Project Submitted in Partial Fulfillment of Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering
Faculty of Engineering
Chiang Mai University
2025

หัวข้อโครงงาน	:	SCMC AssetManager : A Web Application for Centralized Digital Asset Management at Smart Campus Management Center
	:	เอสซีเอ็มซี แอสเซตแมนเนเจอร์ : เว็บแอปพลิเคชันสำหรับระบบบริหารจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ภายในศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน
โดย	:	นายณัฐนันทน์ ตาตะนันทน์ รหัส 650610760
		นายบวรภัค ดวงจันทร์ รหัส 650610779
		นายพชร จารุรักษ์สกุล รหัส 650610789
		นายศิรวิชญ์ ศิรบรรจงกราน รหัส 650610810
ภาควิชา	:	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	:	อ.ดร. พนธกร สอนจันทร์
ปริญญา	:	วิศวกรรมศาสตรบัณฑิต
สาขา	:	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	:	2568

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ผศ.ดร. นวดนย์ คุณเลิศกิจ)

คณะกรรมการสอบโครงงาน

..... ประธานกรรมการ
(อ.ดร. พนธกร สอนจันทร์)

..... กรรมการ
(ผศ.ดร. ภาสกร แช่มประเสริฐ)

..... กรรมการ
(รศ.ดร. ศักดิ์กษิต ระมิงค์วงศ์)

หัวข้อโครงงาน	: SCMC AssetManager : A Web Application for Centralized Digital Asset Management at Smart Campus Management Center	
	: เอสซีเอ็มซี แอสเซตแมนเนเจอร์ : เว็บแอปพลิเคชันสำหรับระบบบริหารจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ภายในศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน	
โดย	: นายณัฐนันทน์ ตาตะนันทน์	รหัส 650610760
	นายบวรภัค ดวงจันทร์	รหัส 650610779
	นายพชร จารุรักษ์สกุล	รหัส 650610789
	นายศิรวิชญ์ ศิรบรรจงกราน	รหัส 650610810
ภาควิชา	: วิศวกรรมคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	: อ.ดร. พนธกร สอนจันทร์	
ปริญญา	: วิศวกรรมศาสตรบัณฑิต	
สาขา	: วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	: 2568	

บทคัดย่อ

ศูนย์บริหารจัดการเมืองเพื่อความยั่งยืนมีครุภัณฑ์ และทรัพย์สินจำนวนมากกระจายอยู่หลายหน่วยงาน ปัจจุบันยังใช้การตรวจนับ และจัดการผ่านไฟล์ Microsoft Excel หลายเวอร์ชัน ส่งต่อกันไปมาระหว่างหน่วยงาน ทำให้เกิดภาระงานซ้ำซ้อน ใช้เวลามาก เสี่ยงต่อข้อมูลผิดพลาด ขาดความเป็นปัจจุบัน ไม่สามารถตรวจสอบย้อนหลังได้อย่างเป็นระบบ และยังไม่ได้ใช้ประโยชน์จากเทคโนโลยี เช่น QR Code อย่างเต็มศักยภาพ จึงส่งผลต่อความน่าเชื่อถือของข้อมูล การจัดทำงบประมาณประจำปี และการบริหารทรัพย์สินของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืนโดยรวม

จากปัญหาการจัดการครุภัณฑ์ที่ยังพึ่งพาไฟล์ Microsoft Excel มีความเสี่ยงต่อความผิดพลาดและข้อมูลไม่เป็นปัจจุบัน จึงมีความจำเป็นในการพัฒนาระบบการจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ เพื่อให้กระบวนการทำงานมีความทันสมัย มีประสิทธิภาพ และลดควาพลาดจาดผู้ใช้งาน โดยโครงงานนี้ยึดหลักการสำคัญ 4 ประการ คือ

1. การรวมศูนย์ข้อมูล (Centralization) สร้างฐานข้อมูลครุภัณฑ์เพียงแหล่งเดียว ให้ผู้ที่มีสิทธิ์สามารถเข้าถึงข้อมูลล่าสุดผ่านเว็บแอปพลิเคชัน ลดปัญหาข้อมูลซ้ำซ้อนและความสับสนจากการใช้ไฟล์ Microsoft Excel หลายเวอร์ชัน
2. การทำงานแบบอัตโนมัติ (Automation) นำเทคโนโลยี QR Code มาใช้ให้เจ้าหน้าที่และผู้ครอบครองครุภัณฑ์สแกนเพื่อดูข้อมูล ยืนยันการตรวจนับ แจ้งซ่อม หรือขอโอนย้ายได้ทันที ช่วยลดงานเอกสารและการใช้กระดาษ
3. ความโปร่งใสและตรวจสอบได้ (Transparency & Auditability) ระบบบันทึกประวัติทุกการดำเนินการกับครุภัณฑ์ (Audit Trail) ตั้งแต่การลงทะเบียน การโอนย้าย การแจ้งซ่อม จนถึงการจำหน่าย ทำให้ตรวจสอบย้อนหลังได้อย่างโปร่งใส

4. การมีส่วนร่วมของผู้ใช้งาน (User Engagement) ออกแบบระบบให้ใช้งานง่ายสำหรับผู้ใช้งานทุกระดับ ตั้งแต่ผู้บริหาร เจ้าหน้าที่ครุภัณฑ์ ไปจนถึงผู้ครอบครองครุภัณฑ์ เพื่อส่งเสริมให้ทุกคนมีส่วนร่วมในการดูแลทรัพย์สินของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืนอย่างถูกต้องและสะดวกสบาย

โครงงานนี้พัฒนาระบบโดยอิงข้อมูลครุภัณฑ์จากระบบเดิมและไฟล์ Microsoft Excel ต้องปฏิบัติตามระเบียบการเงิน การบัญชี และการครุภัณฑ์ของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน ใช้โครงสร้างพื้นฐานไอทีที่มีอยู่ และเน้นฟังก์ชันพื้นฐานสำหรับการตรวจนับครุภัณฑ์ในระยะแรก การใช้งานจะเริ่มจากหน่วยงานนำร่อง ขึ้นกับคุณภาพข้อมูลเดิมและวินัยในการอัปเดตของผู้ใช้ ระบบต้องใช้อุปกรณ์ที่มีกล้องและอินเทอร์เน็ต ไม่มีโหมดออฟไลน์ และรายงาน/แดชบอร์ดอยู่ในระดับพื้นฐานที่สามารถต่อยอดได้ในอนาคต

Project Title	: เอสซีเอ็มซี แอสเซตแมนเนเจอร์ : เว็บแอปพลิเคชันสำหรับระบบบริหารจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ภายในศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน	
Name	: Natanan Tatanan	650610760
	Borwonpak Duangjun	650610779
	Patchara Jaruraxsakul	650610789
	Sirawit Sirabanchongkran	650610810
Department	: Computer Engineering	
Project Advisor	: Pontakorn Sonchan, Ph.D.	
Degree	: Bachelor of Engineering	
Program	: Computer Engineering	
Academic Year	: 2025	

ABSTRACT

Background and Significance

Smart Campus Management Center possesses a large number of assets and equipment distributed across multiple departments. Currently, asset verification and management rely on multiple versions of Excel files exchanged between departments. This approach leads to redundant workloads, time consumption, risks of data inconsistency, outdated information, and the lack of systematic auditability. Moreover, existing technologies such as QR Code have not been fully utilized. These limitations affect data reliability, annual budgeting processes, and overall asset management efficiency within the Smart Campus Management Center.

Principles and Rationale

Due to the limitations of Excel-based asset management, there is a necessity to develop a centralized digital asset management system to modernize workflows, improve efficiency, and reduce human error. This project is based on four key principles: (1) Centralization — establishing a single, centralized asset database accessible through a web application to eliminate data redundancy and version conflicts; (2) Automation — integrating QR Code technology to allow staff and asset holders to scan for asset information, confirm inspections, report maintenance, or request transfers instantly; (3) Transparency and Auditability — implementing an audit trail mechanism that records all asset-related activities from registration to disposal; and (4) User Engagement — designing a user-friendly system suitable for administrators, asset officers, and asset holders to encourage active participation in asset management.

Conditions and Limitations

The system is developed based on existing asset data and Excel files and must comply with the Smart Campus Management Center's financial and asset management regulations.

It operates within the current IT infrastructure and initially focuses on core asset verification functions. The implementation begins with pilot departments and depends on data quality and user discipline in updating information. The system requires devices with cameras and internet connectivity, does not support offline mode, and provides basic reporting and dashboard functionalities that can be extended in the future.

กิตติกรรมประกาศ

การพัฒนาระบบ SCMC AssetManager ในครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้จัดทำขอกราบขอบพระคุณ อ.ดร. พนธกร สอนจันทร์ อาจารย์ที่ปรึกษาโครงงาน ที่ได้กรุณาให้คำปรึกษา ชี้แนะแนวทาง ตรวจสอบความถูกต้อง และให้ความอนุเคราะห์ในด้านต่าง ๆ ตลอดระยะเวลาในการดำเนินงาน รวมถึงขอกราบขอบพระคุณ ผศ.ดร. ภาสกร แซ่มประเสริฐ และ รศ.ดร. ศักดิ์กษิต ระมิงค์วงศ์ คณะกรรมการสอบโครงงาน ที่ได้เสียสละเวลาในการประเมินและให้ข้อเสนอแนะอันเป็นประโยชน์อย่างยิ่งต่อการพัฒนาระบบให้มีความสมบูรณ์มากยิ่งขึ้น

ขอกราบขอบพระคุณพี่ ๆ นักวิชาการคอมพิวเตอร์ นายพร้อมสิน ศีลอุดม(เจ้าหน้าที่ฝ่ายพัสดุ) ตลอดจนบุคลากรของมหาวิทยาลัยเชียงใหม่ทุกท่าน ที่ได้ให้ความอนุเคราะห์ข้อมูล เอื้อเฟื้อประสบการณ์การทำงานจริง และให้ความช่วยเหลือในการทดสอบระบบ ซึ่งมีส่วนสำคัญอย่างยิ่งในการทำให้ระบบสามารถใช้งานได้จริงและตรงตามวัตถุประสงค์ที่ตั้งไว้

นอกจากนี้ ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่คอยให้ความรัก ความอบอุ่น และการสนับสนุนในทุก ๆ ด้านเสมอมา ตลอดจนขอขอบคุณเพื่อน ๆ นักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ ที่คอยช่วยเหลือ แลกเปลี่ยนความรู้ และเป็นกำลังใจให้กันตลอดการศึกษา

ท้ายที่สุดนี้ คุณค่าและประโยชน์อันใดที่เกิดจากโครงงานฉบับนี้ ผู้จัดทำขอมอบเป็นกตัญญูกตเวทิตาแด่ผู้มีพระคุณทุกท่าน และหวังเป็นอย่างยิ่งว่าประสบการณ์และความรู้ที่ได้รับจากการดำเนินโครงการครั้งนี้ จะเป็นรากฐานสำคัญในการประกอบวิชาชีพทางด้านวิศวกรรมคอมพิวเตอร์ของผู้จัดทำต่อไปในอนาคต

นายณัฐนันทน์ ตาตะนันทน์
นายบวรภัค ดวงจันทร์
นายพชร จารุรักษ์สกุล
นายศิรวิชญ์ ศิรบรรจงกราน
5 มีนาคม 2569

สารบัญ

บทคัดย่อ . . . . . ข
Abstract . . . . . ง
กิตติกรรมประกาศ . . . . . ฉ
สารบัญ . . . . . ช
สารบัญรูป . . . . . ญ
สารบัญตาราง . . . . . ฑ

1 บทนำ **1**
1.1 ที่มาของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 1
1.3.1 ขอบเขตด้านฮาร์ดแวร์ 1
1.3.2 ขอบเขตด้านซอฟต์แวร์ 2
1.4 ประโยชน์ที่ได้รับ 2
1.5 เทคโนโลยีและเครื่องมือที่ใช้ 3
1.5.1 เทคโนโลยีด้านซอฟต์แวร์ 3
1.6 แผนการดำเนินงาน 4
1.7 บทบาทและความรับผิดชอบ 4
1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม 5
1.9 แผนการใช้งบประมาณ 6

2 ทฤษฎีที่เกี่ยวข้อง **7**
2.1 สถาปัตยกรรมระบบแบบ Client–Server 7
2.2 หลักการออกแบบฐานข้อมูลเชิงสัมพันธ์ 7
2.3 แนวคิด Role-Based Access Control (RBAC) 7
2.4 สถาปัตยกรรม RESTful API 7
2.5 แนวคิด Audit Trail และความโปร่งใสของข้อมูล 8
2.6 เทคโนโลยี QR Code ในการจัดการทรัพย์สิน 8
2.7 มาตรฐาน OAuth 2.0 และระบบ CMU Authorization 8
2.7.1 แนวคิดพื้นฐานของ OAuth 2.0 8
2.7.2 บริการ Microsoft Entra ID 9
2.7.3 ระบบ CMU Authorization 9
2.7.4 รูปแบบการให้สิทธิ์ (OAuth 2.0 Grant Types) 11
2.7.5 ขอบเขตและสิทธิ์การเข้าถึง (Scope and Permission) 16
2.7.6 ประเภทของโทเค็นในระบบ OAuth 2.0 18
2.7.7 การเปลี่ยนผ่านระบบจาก CMU OAuth สู่ Microsoft Entra ID 19
2.8 โครงสร้างกลุ่มภารกิจศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน (SCMC) 19
2.9 โครงสร้างข้อมูลแบบต้นไม้ (Tree Data Structure) 21
2.10 ทฤษฎีการทดสอบระบบ (System Testing) 22
2.10.1 การทดสอบแบบกล่องดำ (Black-box Testing) 22
2.10.2 การทดสอบแบบกล่องขาว (White-box Testing) 22
2.11 การประเมินความสามารถในการใช้งานด้วย System Usability Scale (SUS) 22
2.12 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงงาน 23
2.13 ความรู้นอกหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงงาน 24

3 โครงสร้างและขั้นตอนการทำงาน **25**
3.1 โครงสร้างและการทำงานของระบบ . . . 25
3.1.1 ระบบส่วนติดต่อผู้ใช้งาน (Frontend) . . . 25
3.1.2 ระบบจัดการหลังบ้าน (Backend) . . . 25
3.1.3 ระบบฐานข้อมูล (Database) . . . 25
3.1.4 การนำระบบขึ้นใช้งาน (Deployment) . . . 26
3.2 การกำหนดบทบาทผู้ใช้งานในระบบ . . . 26
3.2.1 ผู้ใช้งานทั่วไป (User) . . . 26
3.2.2 ผู้ดูแลระบบ (Admin) . . . 26
3.2.3 ผู้ดูแลระบบระดับสูง (Superadmin) . . . 27
3.3 ฟีเจอร์ของระบบ . . . 27
3.3.1 ระบบการแสดงข้อมูลครุภัณฑ์ในรูปแบบ Dashboard . . . 27
3.3.2 ระบบจัดการข้อมูลครุภัณฑ์ . . . 28
3.3.3 ระบบรายงานครุภัณฑ์ประจำปี . . . 28
3.3.4 ระบบโอนย้ายครุภัณฑ์ . . . 29
3.3.5 ระบบการแจ้งซ่อมครุภัณฑ์ . . . 29
3.3.6 ระบบแจ้งเตือน . . . 29
3.3.7 ระบบนับรอบการตรวจครุภัณฑ์ . . . 30
3.3.8 ระบบจัดการข้อมูลหน่วยงาน . . . 31
3.3.9 ระบบบันทึกประวัติย้อนหลัง . . . 31
3.3.10 ระบบสร้างและใช้งาน QR Code สำหรับครุภัณฑ์ . . . 32
3.4 ขั้นตอนการทำงานของระบบ . . . 32
3.5 การออกแบบส่วนติดต่อผู้ใช้งาน . . . 32
3.5.1 การออกแบบ User Flow . . . 32
3.5.2 Use Case Diagram . . . 40
3.5.3 การออกแบบหน้าจอของระบบ . . . 41
3.6 การออกแบบฐานข้อมูล . . . 44
3.6.1 โครงสร้างฐานข้อมูล . . . 44

4 การทดลองและผลลัพธ์ **46**
4.1 การทดสอบการทำงานของระบบ (System Testing) . . . 46
4.1.1 การทดสอบระดับหน่วย (Unit Testing / White-Box Testing) . . . 46
4.1.2 การทดสอบระดับระบบ (End-to-End Testing / Black-Box Testing) . . . 47
4.2 จุดประสงค์ของการประเมินระบบ . . . 47
4.3 วิธีการประเมินระบบ . . . 48
4.4 ผลการประเมินระบบ . . . 48
4.4.1 ผลประเมินเชิงปริมาณ . . . 49
4.4.2 ผลประเมินเชิงคุณภาพ . . . 49
4.4.3 ผลประเมินด้านความสามารถในการใช้งานระบบ (System Usability Scale: SUS) 50

5 บทสรุปและข้อเสนอแนะ **51**
5.1 สรุปผล . . . 51
5.2 ปัญหาที่พบและแนวทางการแก้ไข . . . 51
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ . . . 51

บรรณานุกรม **52**

ก คู่มือการใช้งานระบบ **55**
ก.1 การเข้าสู่ระบบและการตั้งค่าเริ่มต้น 55
ก.1.1 การเข้าสู่ระบบ 55
ก.1.2 การเลือกหน่วยงาน 55
ก.1.3 การตั้งค่าต่าง ๆ ในระบบ 60
ก.2 การจัดการครุภัณฑ์ (Assets) 63
ก.2.1 การดูข้อมูลครุภัณฑ์ (Assets) 63
ก.2.2 การค้นหาครุภัณฑ์ (Assets) 64
ก.2.3 การพิมพ์ QR Code ครุภัณฑ์ (Assets) 65
ก.3 การแก้ไขและแจ้งซ่อม 67
ก.3.1 คำขอย้ายตำแหน่ง 67
ก.3.2 การแจ้งซ่อม 71
ก.4 การโอนย้ายครุภัณฑ์ 74
ก.4.1 การส่งการโอนย้ายครุภัณฑ์ 74
ก.4.2 การรับการโอนย้ายครุภัณฑ์ 78
ก.4.3 การปฏิเสธการโอนย้ายครุภัณฑ์ 79
ก.5 ระบบสแกน QR Code 80
ก.5.1 การสแกน 80
ก.5.2 ผลลัพธ์การสแกน 81
ก.6 รอบการตรวจนับครุภัณฑ์ 83
ก.6.1 การตรวจสอบสถานะ 83
ก.6.2 การรายงานสภาพ 84
ก.6.3 กรณีชำรุด 85
ก.7 การแจ้งเตือน 86
ก.7.1 การรับข่าวสารและการจัดการข้อความ 86
ก.8 คู่มือสำหรับผู้ดูแล (ADMIN) 87
ก.8.1 จัดการครุภัณฑ์ (Admin Assets) 88
ก.8.2 จัดการการโอนย้ายและคำร้อง (Admin Transfers & Requests) 89
ก.8.3 รอบตรวจนับ (Check Cycles) 92
ก.8.4 รายงาน (Reports) 93
ก.9 คู่มือสำหรับผู้ดูแลระบบสูงสุด (Superadmin) 97
ก.9.1 แดชบอร์ดและสถิติ (Dashboard & Analytics) 97
ก.9.2 การบริหารจัดการข้อมูลครุภัณฑ์แบบเจาะลึก 98
ก.9.3 นำเข้าข้อมูลจำนวนมาก (Bulk Import) 100
ก.9.4 การจัดการองค์กร (Organization / Institution) 103
ก.9.5 จัดการผู้ใช้งาน (User Management) 106
ก.9.6 การแจ้งเตือนและการเตือนความจำรอบตรวจนับ 107

ข การทดสอบระบบกับศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน **108**

ประวัติผู้เขียน **111**

สารบัญรูป

2.1 ภาพรวมการทำงานของสถาปัตยกรรม RESTful API ระหว่าง Client และ Server ผ่านโปรโตคอล HTTP 8
2.2 ภาพรวมการทำงานของ CMU Authorization Code Flow แสดงความสัมพันธ์ระหว่าง Client App, Microsoft Entra ID, Kong API Gateway, Resource Owner และ ITSC 10
2.3 ภาพรวมการทำงานของ CMU Client Credential Flow สำหรับการสื่อสารระหว่างเซิร์ฟเวอร์ (Server-to-Server) โดยไม่มีกระบวนการของผู้ใช้งานเข้ามาเกี่ยวข้อง 11
2.4 แผนภาพลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Authorization Code Flow ตั้งแต่การร้องขอ Authorization Code จนถึงการใช้งาน Refresh Token 12
2.5 แผนภาพลำดับแสดงขั้นตอนการทำงานของ Authorization Code Flow with PKCE สำหรับ Public Client ที่ไม่สามารถจัดเก็บ Client Secret ได้ 13
2.6 แผนภาพลำดับแสดงขั้นตอนการทำงานของ Client Credentials Flow เริ่มตั้งแต่การขอ Admin Consent ไปจนถึงการเรียกใช้งาน Protected API 14
2.7 แผนภาพลำดับแสดงขั้นตอนการทำงานของ Device Code Flow โดยแสดงการทำงานที่ขนานกันระหว่าง Device Channel และ User Channel 15
2.8 การตั้งค่า Implicit grant and hybrid flows ใน Microsoft Entra ID เพื่อเปิดใช้งาน ID tokens สำหรับใช้งาน Hybrid Flow 16
2.9 หน้าจอ CMU API Portal แสดงการกำหนด Scope `Mis.Account.Read.Me.Basicinfo` สำหรับ Grant Type ประเภท AUTHORIZATION_CODE 17
2.10 หน้าจอ CMU API Portal แสดง Scope สำหรับ Grant Type ประเภท `CLIENT_CREDENTIAL` ซึ่งต้องได้รับการอนุมัติระดับผู้ดูแลระบบ (Admin Consent) ก่อนใช้งาน 17
2.11 หน้าจอ Request API permissions ใน Microsoft Entra ID แสดงการแบ่งกลุ่ม Delegated permissions และ Application permissions ของระบบ CMU API 18
2.12 แผนภาพแสดงลักษณะของโครงสร้างข้อมูลแบบต้นไม้ (Tree Data Structure) ประกอบด้วย Root Node, Internal Node และ Leaf Node 21

3.1 Login User Flow 33
3.2 Asset Query User Flow 33
3.3 Add Asset User Flow 34
3.4 Edit Asset User Flow 34
3.5 Delete Asset User Flow 35
3.6 Creating Asset Checking Cycle Per Year User Flow 35
3.7 Monitoring Asset Checking Cycle User Flow 36
3.8 Creating Asset Report User Flow 36
3.9 Notification System User Flow 37
3.10 QR Code Printing User Flow 37
3.11 QR Scanning User Flow 38
3.12 Transfer Ownership of Asset User Flow 39
3.13 Use Case Diagram ของระบบ 40
3.14 ตัวอย่างหน้าการแสดงข้อมูลครุภัณฑ์ในรูปแบบ Dashboard 41
3.15 ตัวอย่างหน้า Annual Report 41
3.16 ตัวอย่างหน้า Transfer 42
3.17 ตัวอย่างหน้า Notification 42
3.18 ตัวอย่างหน้า Check Cycle 43

3.19 ตัวอย่างหน้า Institution Management 43
3.20 ตัวอย่างหน้า Audit Log 44
3.21 ตัวอย่างหน้า QR Code And Smart QR 44
3.22 Database Schema ของระบบ 45

ก.1 หน้าต่าง Login 55
ก.2 เลือกหรือสร้างหน่วยงาน 56
ก.3 เลือกหน่วยงานหลักและหน่วยงานย่อย (ไม่บังคับสำหรับหน่วยงานย่อย) 57
ก.4 ยืนยันการเลือกหน่วยงานหลักและหน่วยงานย่อย 58
ก.5 หน้าต่างหลังจากเข้าหน่วยงานสำเร็จ 58
ก.6 สร้างหน่วยงานหลักใหม่ 59
ก.7 ยืนยันการสร้างหน่วยงานหลักใหม่ 59
ก.8 หน้าต่างหลังจากสร้างหน่วยงานหลักใหม่ ผู้สร้างเป็น Superadmin ของหน่วยงานที่สร้างใหม่ 60
ก.9 ทางลัดไปหน้าต่าง ๆ ที่เกี่ยวกับผู้ใช้ 61
ก.10 หน้าโปรไฟล์ของผู้ใช้ 62
ก.11 หน้าการแก้ไขเบอร์โทรศัพท์ในโปรไฟล์ของผู้ใช้ 62
ก.12 หน้าการตั้งค่า 62
ก.13 หน้าหน่วยงาน/องค์กร 62
ก.14 หน้าต่างการออกจากหน่วยงาน ผู้ใช้ต้องพิมพ์คำว่า LEAVE จึงจะสามารถออกจากหน่วยงานได้ 63
ก.15 หน้าต่างการดูครุภัณฑ์ทั้งหมดที่ผู้ใช้เป็นผู้ดูแล 63
ก.16 หน้าต่างการดูรายละเอียดครุภัณฑ์แต่ละชิ้น 64
ก.17 การพิมพ์ค้นหาครุภัณฑ์ที่ตรงตามต้องการ 65
ก.18 หน้าต่างการตั้งค่า QR Code ที่จะพิมพ์ 66
ก.19 หน้าต่างแสดง QR Code ก่อนพิมพ์ 66
ก.20 หน้าต่างการพิมพ์ QR Code 67
ก.21 ปุ่มขอย้ายตำแหน่งในหน้ารายละเอียดครุภัณฑ์ 68
ก.22 หน้าต่างการกรอกข้อมูลตำแหน่งสถานที่ใหม่ของครุภัณฑ์ 69
ก.23 ผู้ใช้สามารถกดปุ่ม '+' ที่อาคารเพื่อเพิ่มอาคารใหม่ได้ 70
ก.24 ผู้ใช้สามารถกดปุ่ม '+' ที่ห้องเพื่อเพิ่มห้องใหม่ได้ 70
ก.25 คำขอย้ายตำแหน่งครุภัณฑ์จะขึ้นในหน้าคำขอย้ายตำแหน่ง 70
ก.26 หน้าต่างดูรายละเอียดของคำขอย้ายตำแหน่ง ผู้ใช้สามารถยกเลิกคำขอย้ายตำแหน่งได้ . . . 71
ก.27 ปุ่มแจ้งซ่อมในหน้ารายละเอียดครุภัณฑ์ 72
ก.28 หน้าต่างการส่งคำขอซ่อม 73
ก.29 คำขอซ่อมครุภัณฑ์จะขึ้นในหน้าคำขอซ่อม 73
ก.30 หน้าต่างดูรายละเอียดของคำขอซ่อม 74
ก.31 มาที่หน้าการโอนย้ายครุภัณฑ์ เพื่อจะเริ่มการโอนย้ายครุภัณฑ์โดยกดที่ปุ่ม 'สร้างคำขอโอนย้าย' ที่มุมขวาบน 75
ก.32 หน้าต่างการกรอกข้อมูลครุภัณฑ์และผู้ดูแลคนใหม่ และสามารถเลือกให้หน่วยงานของครุภัณฑ์เปลี่ยนตามหน่วยงานของผู้ดูแลคนใหม่ได้ 75
ก.33 หลังจากสร้างคำขอโอนย้ายแล้ว สามารถดูคำขอโอนย้ายที่ผู้ใช้ได้ทำการสร้างในแท็บ 'ส่งออก' 76
ก.34 ผู้ใช้สามารถดูรายละเอียดตำแหน่งสถานที่ใหม่ของครุภัณฑ์ที่โอนย้ายที่ผู้รับได้ทำการกรอกเข้ามาได้ 77
ก.35 ผู้รับสามารถดูคำขอโอนย้ายที่จะทำการรับไปดูแลได้ในแท็บ 'รับเข้า' 78

ก.36 ผู้รับต้องเข้ามากรอกข้อมูลสถานที่ใหม่ของครุภัณฑ์ 79
ก.37 ทั้งผู้ส่งและผู้รับสามารถกดปุ่ม 'ปฏิเสธ' เพื่อเข้ามาหน้าต่างการยืนยันการปฏิเสธการโอนย้ายได้ 80
ก.38 หน้าต่างเปิดกล้องการสแกน QR Code บนระบบของเรา 81
ก.39 หน้าต่างการดูรายละเอียดครุภัณฑ์จะแสดงขึ้นมาหลังสแกน 82
ก.40 ในกรณีที่ QR ที่สแกนเป็นครุภัณฑ์ที่อยู่ระหว่างการตรวจนับ จะแสดงหน้าต่างการตรวจสภาพครุภัณฑ์ (จะลงรายละเอียดในส่วนการตรวจนับอีกที) 83
ก.41 เมื่อมีรอบตรวจนับครุภัณฑ์ที่เกี่ยวข้องกับผู้ใช้เกิดขึ้น จะมีแท็บ 'รอบตรวจนับ' ที่ผู้ใช้สามารถกดเข้ามาดูหน้ารอบตรวจนับได้ 83
ก.42 ผู้ใช้สามารถกดขยายดูรายละเอียดต่าง ๆ ได้ว่าต้องตรวจนับครุภัณฑ์ชิ้นไหนบ้าง ตรวจครบหรือยัง 84
ก.43 ผู้ใช้สามารถกดปุ่ม 'ตรวจนับ' เพื่อเข้ามาหน้าการตรวจสภาพครุภัณฑ์ได้ 84
ก.44 ผู้ใช้ต้องเลือกสภาพของครุภัณฑ์หลังจากทำการตรวจสอบแล้ว 85
ก.45 เมื่อตรวจนับเสร็จแล้วกลับมาที่หน้ารอบตรวจนับ จะเห็นความคืบหน้าขอการตรวจนับ . . 85
ก.46 หากผู้ใช้เลือกสภาพชำรุดตอนตรวจนับ ผู้ใช้จะต้องอัพโหลดรูปภาพครุภัณฑ์ปัจจุบันและกรอกรายละเอียดการชำรุด 86
ก.47 หน้าต่างการแจ้งเตือนทางด้านบน 87
ก.48 หน้าการจัดการการแจ้งเตือน 87
ก.49 หน้ารายการครุภัณฑ์สำหรับผู้ดูแลและการแก้ไขข้อมูล 88
ก.50 ปุ่มสำหรับแก้ไขข้อมูลครุภัณฑ์สำหรับผู้ดูแล 88
ก.51 หน้าต่างตรวจสอบการโอนย้ายสำหรับผู้ดูแล 89
ก.52 หน้าต่างอนุมัติการโอนย้ายสำหรับผู้ดูแล 90
ก.53 หน้าต่างดูคำขอย้ายตำแหน่งสำหรับ Admin/Superadmin 90
ก.54 หน้าต่างอนุมัติคำขอย้ายตำแหน่งสำหรับ Admin/Superadmin 91
ก.55 หน้าต่างดูคำร้องแจ้งซ่อมสำหรับ Admin/Superadmin 91
ก.56 หน้าต่างอนุมัติคำร้องแจ้งซ่อมสำหรับ Admin/Superadmin 91
ก.57 หน้าต่างตรวจสอบรอบตรวจนับสำหรับ Admin/Superadmin 92
ก.58 หน้าต่างสร้างรอบตรวจนับสำหรับ Admin/Superadmin 92
ก.59 หน้าต่างแก้ไขรอบตรวจนับสำหรับ Admin/Superadmin 93
ก.60 หน้าต่างติดตามผลรอบตรวจนับสำหรับ Admin/Superadmin 93
ก.61 หน้าต่างสร้างรายงานสำหรับ Admin/Superadmin 94
ก.62 หน้าต่างสร้างรายงานสำหรับ Admin/Superadmin 94
ก.63 หน้าต่างปรับแต่งรายงานสำหรับ Admin/Superadmin 95
ก.64 หน้าต่างส่งออกรายงานเป็นไฟล์ Excel สำหรับ Admin/Superadmin 95
ก.65 หน้าต่างลบรายงานสำหรับ Admin/Superadmin 96
ก.66 หน้าต่างแดชบอร์ดส่วนบนสำหรับ Superadmin 97
ก.67 หน้าต่างแดชบอร์ดส่วนล่างสำหรับ Superadmin 97
ก.68 หน้าต่างแบบฟอร์มการเพิ่มครุภัณฑ์สำหรับ Superadmin 98
ก.69 ปุ่มการลบครุภัณฑ์ขแบบรายชิ้นสำหรับ Superadmin 99
ก.70 ปุ่มการลบครุภัณฑ์แบบพร้อมกันหลายรายการสำหรับ Superadmin 99
ก.71 หน้าต่างการเรียกดูประวัติทำรายการ (Audit Logs) ภายในรายละเอียดครุภัณฑ์สำหรับ Superadmin 100
ก.72 หน้าต่างการนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin 100
ก.73 หน้าต่างเริ่มนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin 101
ก.74 หน้าต่างเลือกชีต, หน่วยงาน, และแถวหัวข้อสำหรับ Superadmin 101

ก.75 หน้าต่างจับคู่คอลัมน์สำหรับ Superadmin 102
ก.76 หน้าต่างตรวจสอบข้อมูลก่อนนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin 102
ก.77 หน้าต่างเริ่มนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin 103
ก.78 หน้าต่างแผนผังหน่วยงานสำหรับ Superadmin 103
ก.79 หน้าต่างเพิ่มหน่วยงานสำหรับ Superadmin 103
ก.80 หน้าต่างแก้ไขหน่วยงานสำหรับ Superadmin 104
ก.81 หน้าต่างลบหน่วยงานสำหรับ Superadmin 104
ก.82 หน้าต่างลบอาคารสำหรับ Superadmin 105
ก.83 หน้าต่างลบห้องสำหรับ Superadmin 105
ก.84 หน้าต่างจัดการผู้ใช้งานสำหรับ Superadmin 106
ก.85 หน้าต่างแก้ไขสังกัดหน่วยงานสำหรับ Superadmin 106
ก.86 หน้าต่างแก้ไขบทบาทผู้ใช้งานสำหรับ Superadmin 107
ก.87 หน้าต่างเตะผู้ใช้งานออกจากหน่วยงานสำหรับ Superadmin 107
ก.88 หน้าต่างส่งการแจ้งเตือนเพื่อเร่งรัดผู้ใช้งานสำหรับ Superadmin 107

ข.1 ภาพบรรยากาศการทดสอบระบบกับศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน 108
ข.2 ภาพระหว่างที่นายบวรภัค ดวงจันทร์ กำลังอธิบายการใช้งานระบบให้กับเจ้าหน้าของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน 108
ข.3 ภาพระหว่างที่สมาชิกทีมกำลังชมการทดสอบการใช้งานระบบร่วมกับเจ้าหน้าที่ฝ่ายพัสดุของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน 109
ข.4 ภาพหน้าจอที่แสดงการทดสอบการใช้งานระบบร่วมกับเจ้าหน้าที่ฝ่ายพัสดุของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน 109
ข.5 ภาพระหว่างที่สมาชิกทีมกำลังทดสอบการใช้งานระบบร่วมกับเจ้าหน้าที่ฝ่ายพัสดุของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน 110

สารบัญตาราง

2.1 ประเภทของ Token และวัตถุประสงค์การใช้งานในระบบ CMU Authorization 18
2.2 เปรียบเทียบการเปลี่ยนแปลงค่าพารามิเตอร์ระหว่างระบบ CMU OAuth เดิมและ Microsoft Entra ID 19

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

ปัจจุบันศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน มีครุภัณฑ์และทรัพย์สินจำนวนมากโดยกระจาย อยู่ในหลายหน่วยงาน ซึ่งการจัดการข้อมูล เช่นการตรวจนับ ยังพึ่งพาไฟล์ Microsoft Excel หลายเวอร์ชันที่ส่งต่อกันระหว่างหน่วยงาน ก่อให้เกิดปัญหางานซ้ำซ้อน ข้อมูลไม่ เป็นปัจจุบัน ความผิดพลาดจากมนุษย์ อีกทั้งยังไม่สามารถตรวจสอบย้อนหลังได้ อย่างเป็นระบบและมีประสิทธิภาพ ซึ่งส่งผลต่อความน่าเชื่อถือของข้อมูลและการ บริหารงบประมาณและทรัพย์สินของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน ดังนั้น โครงงานนี้มีเป้าหมายใน การพัฒนาระบบจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ เพื่อเพิ่มประสิทธิภาพ ความถูกต้อง และความโปร่งใสในการบริหารทรัพย์สินของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน

1.2 วัตถุประสงค์ของโครงงาน

โครงงานการพัฒนาระบบบริหารจัดการครุภัณฑ์แบบดิจิทัล มีวัตถุประสงค์ เพื่อปรับปรุงกระบวนการจัดการทรัพย์สินของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืนให้มีความเป็นระบบ ลดความซ้ำซ้อน และเพิ่มประสิทธิภาพในการดำเนินงาน โดยมีวัตถุประสงค์ดังต่อไปนี้

1. เพื่อพัฒนาระบบฐานข้อมูลครุภัณฑ์แบบรวมศูนย์ ที่สามารถจัดเก็บและจัดการข้อมูลจากหลายหน่วยงานไว้ในระบบเดียว ลดปัญหาการใช้ไฟล์หลายเวอร์ชัน และลดความคลาดเคลื่อนของข้อมูล
2. เพื่อประยุกต์ใช้เทคโนโลยี QR Code ในกระบวนการตรวจนับและติดตามครุภัณฑ์ ให้สามารถสแกนเพื่อเข้าถึงข้อมูลได้อย่างรวดเร็ว ช่วยลดขั้นตอนการทำงานและลดข้อผิดพลาดจากการบันทึกข้อมูลด้วยมือ
3. เพื่อพัฒนาระบบบันทึกประวัติการดำเนินการ (Audit Log) สำหรับติดตามการเปลี่ยนแปลงข้อมูล เช่น การเพิ่ม แก้ไข โอนย้าย หรือแจ้งซ่อมครุภัณฑ์ เพื่อให้สามารถตรวจสอบย้อนหลังได้อย่างโปร่งใส
4. เพื่อจัดทำระบบ Dashboard และรายงานสรุปข้อมูลในรูปแบบที่เข้าใจง่าย เพื่อสนับสนุนการวางแผน การติดตาม และการตัดสินใจของผู้บริหาร
5. เพื่อพัฒนาระบบแจ้งเตือนการตรวจนับและการบำรุงรักษาครุภัณฑ์ เพื่อลดการตกหล่นของงาน และช่วยยืดอายุการใช้งานของทรัพย์สิน

1.3 ขอบเขตของโครงงาน

1.3.1 ขอบเขตด้านฮาร์ดแวร์

ในด้านฮาร์ดแวร์ ระบบถูกออกแบบให้สามารถใช้งานผ่านอุปกรณ์ทั่วไปที่มีอยู่แล้วภายในศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน โดยในฝั่งผู้ใช้งาน (Client Side) รองรับการใช้งานผ่านคอมพิวเตอร์แบบตั้งโต๊ะหรือคอมพิวเตอร์พกพา รวมถึงสมาร์ตโฟนหรือแท็บเล็ตที่มีกล้องสำหรับสแกน QR Code ทั้งนี้อุปกรณ์ดังกล่าวต้องสามารถเชื่อมต่ออินเทอร์เน็ตและใช้งานผ่านเว็บเบราว์เซอร์ได้ โดยระบบไม่รองรับการทำงานแบบออฟ

ไลน์ ในส่วนของเครื่องแม่ข่าย (Server Side) ระบบใช้โครงสร้างพื้นฐานด้านเทคโนโลยีสารสนเทศที่มีอยู่เดิมของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน ไม่ได้มีการจัดซื้อหรือเพิ่มฮาร์ดแวร์เฉพาะทางเพิ่มเติม โดยใช้เครื่อง Server หรือ Cloud Server สำหรับประมวลผลระบบจัดการหลังบ้าน จัดเก็บฐานข้อมูล และจัดเก็บไฟล์รูปภาพหรือเอกสารที่เกี่ยวข้องกับครุภัณฑ์ ทั้งนี้โครงงานยังไม่ครอบคลุมการออกแบบระบบโครงสร้างพื้นฐานระดับ High Availability หรือระบบสำรองข้อมูลขั้นสูงระดับศูนย์ข้อมูล

1.3.2 ขอบเขตด้านซอฟต์แวร์

ในด้านซอฟต์แวร์ โครงงานพัฒนาเป็น Web-Based Application ภายใต้สถาปัตยกรรมแบบ Client–Server โดยครอบคลุมการทำงานหลัก ได้แก่ การจัดการข้อมูลครุภัณฑ์ การกำหนดสิทธิ์การเข้าถึงตามบทบาทของผู้ใช้งาน การสร้างและสแกน QR Code การโอนย้ายครุภัณฑ์ การกำหนดรอบการตรวจนับประจำปี การแสดงผลข้อมูลผ่าน Dashboard และการจัดทำรายงานสรุป รวมถึงการบันทึกประวัติการเปลี่ยนแปลงข้อมูลในรูปแบบ Audit Log และระบบแจ้งเตือนที่เกี่ยวข้องกับกระบวนการจัดการครุภัณฑ์ นอกจากนี้ ระบบถูกออกแบบให้สามารถรองรับผู้ใช้งานหลายคนพร้อมกัน มีการสื่อสารข้อมูลผ่านโปรโตคอลที่มีความปลอดภัย และสามารถส่งออกข้อมูลในรูปแบบไฟล์รายงาน เช่น PDF หรือ Excel ได้ อย่างไรก็ตาม ระบบยังไม่ครอบคลุมการเชื่อมต่อกับระบบ ERP หรือระบบบัญชีส่วนกลางของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน และไม่รองรับการทำงานแบบออฟไลน์ ทั้งนี้โครงสร้างของระบบได้รับการออกแบบให้สามารถพัฒนาและขยายความสามารถเพิ่มเติมได้ในอนาคต

1.4 ประโยชน์ที่ได้รับ

การพัฒนาระบบ SCMC AssetManager ก่อให้เกิดประโยชน์ทั้งต่อผู้ใช้งานและต่อศูนย์บริหารจัดการเมืองเพื่อความยั่งยืนในภาพรวม ดังนี้

ประโยชน์ต่อผู้ใช้งาน

ระบบช่วยลดภาระงานเอกสารและการจัดการข้อมูลซ้ำซ้อน โดยผู้ใช้งานสามารถเข้าถึงข้อมูลครุภัณฑ์ที่ตนรับผิดชอบได้แบบเรียลไทม์ผ่านระบบออนไลน์ ลดความสับสนเกี่ยวกับสถานะ ความเป็นเจ้าของ และสถานที่ตั้งของครุภัณฑ์ นอกจากนี้ ยังเพิ่มความสะดวกในการแจ้งซ่อม การยืนยันการตรวจนับประจำปี และการติดตามสถานะการดำเนินการต่าง ๆ ได้อย่างรวดเร็วและเป็นระบบ

ประโยชน์ต่อศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน

ระบบช่วยให้ข้อมูลครุภัณฑ์มีความถูกต้อง เป็นปัจจุบัน และรวมศูนย์อยู่ในฐานข้อมูลเดียว เพิ่มความโปร่งใสและสามารถตรวจสอบย้อนหลังได้ผ่านระบบ Audit Log สนับสนุนการจัดทำรายงานและการวางแผนงบประมาณประจำปีได้อย่างมีประสิทธิภาพ ลดความเสี่ยงจากการสูญหายของทรัพย์สิน และส่งเสริมการบริหารจัดการองค์กรในรูปแบบดิจิทัล (Digital Transformation)

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านซอฟต์แวร์

โครงการนี้พัฒนาในรูปแบบ Web Application โดยใช้สถาปัตยกรรมแบบ Client–Server Architecture ซึ่งแบ่งออกเป็นส่วนของ ระบบส่วนติดต่อผู้ใช้งาน, ระบบจัดการหลังบ้าน, ระบบฐานข้อมูล, การทดสอบระบบ และ โครงสร้างพื้นฐานของระบบ เพื่อให้ระบบมีความยืดหยุ่น เสถียร และสามารถขยายต่อได้ในอนาคต

Frontend Technology

- **TypeScript** [1] เป็นภาษาที่พัฒนาต่อยอดจาก JavaScript โดยเพิ่มระบบกำหนดชนิดข้อมูล (Static Typing) ช่วยลดข้อผิดพลาดของโปรแกรม และเพิ่มความน่าเชื่อถือของโค้ด
- **React** [2] เป็น JavaScript Library สำหรับพัฒนา User Interface แบบ Component-based ช่วยให้สามารถจัดการโครงสร้างของหน้าเว็บได้อย่างเป็นระบบ และรองรับการอัปเดตข้อมูลแบบ Dynamic
- **Tailwind CSS** [3] เป็น CSS Framework ที่ช่วยในการออกแบบและจัดรูปแบบหน้าเว็บ ทำให้พัฒนา UI ได้รวดเร็วและมีความสม่ำเสมอ

Backend Technology

- **Node.js** [4] เป็น Runtime Environment สำหรับรัน JavaScript ฝั่งเซิร์ฟเวอร์ รองรับการทำงานแบบ Asynchronous ซึ่งเหมาะสมกับระบบที่มีผู้ใช้งานหลายคน
- **Fastify** [5] เป็น Web Framework สำหรับสร้าง RESTful API มีประสิทธิภาพสูงและรองรับการพัฒนาแบบ Modular
- **Drizzle ORM** [6] เป็นเครื่องมือสำหรับจัดการฐานข้อมูลในรูปแบบ Object Relational Mapping (ORM) ช่วยลดความซับซ้อนของคำสั่ง SQL และเพิ่มความปลอดภัยของข้อมูล
- **ExcelJS** [7] เป็น Library สำหรับอ่านและสร้างไฟล์ Excel (.xlsx) ใช้สำหรับการนำเข้าหรือส่งออกข้อมูลในรูปแบบตารางภายในระบบ
- **FastCSV** [5] เป็น Library สำหรับอ่านและเขียนไฟล์ CSV ใช้สำหรับประมวลผลข้อมูลในรูปแบบ Comma-Separated Values อย่างมีประสิทธิภาพ
- **NestJS** [8] เป็น Framework สำหรับพัฒนา Backend ด้วยภาษา TypeScript และ Node.js โดยใช้แนวคิดแบบ Modular Architecture และ Dependency Injection ช่วยให้สามารถพัฒนา RESTful API ได้อย่างเป็นระบบ และรองรับการขยายระบบในอนาคต

Database and Storage

- **PostgreSQL** [9] เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) ที่มีความเสถียร รองรับข้อมูลจำนวนมาก และมีความปลอดภัยสูง

- **MinIO** [10] เป็นระบบจัดเก็บข้อมูลแบบ Object Storage ใช้สำหรับจัดเก็บไฟล์เอกสารหรือไฟล์แนบภายในระบบ

Testing Tools

- **Postman** [11] ใช้สำหรับทดสอบและตรวจสอบการทำงานของ API
- **Jest** [12] เป็น Framework สำหรับเขียน Unit Test เพื่อทดสอบความถูกต้องของฟังก์ชันในระบบ
- **Cypress** [13] เป็นเครื่องมือสำหรับทดสอบระบบฝั่ง Frontend แบบ End-to-End Testing โดยจำลองพฤติกรรมของผู้ใช้งานจริงในการใช้งานเว็บไซต์
- **Supertest** เป็น Library สำหรับทดสอบการทำงานของ REST API โดยใช้ในการตรวจสอบการตอบสนองของ Server และ Endpoint ต่าง ๆ

Infrastructure

- **Docker** [14] ใช้สำหรับสร้าง Container เพื่อให้ระบบสามารถทำงานได้ในทุกสภาพแวดล้อม และลดปัญหาความแตกต่างของระบบปฏิบัติการ
- **Nginx** [15] ทำหน้าที่เป็น Web Server และ Reverse Proxy เพื่อเพิ่มประสิทธิภาพและความปลอดภัยของระบบ

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	พ.ย. 2568	ธ.ค. 2568	ม.ค. 2569	ก.พ. 2569	มี.ค. 2569
ศึกษาค้นคว้าและวิเคราะห์ความต้องการของระบบจัดการครุภัณฑ์	■	■			
ออกแบบสถาปัตยกรรมระบบและโครงสร้างฐานข้อมูล		■	■		
ออกแบบส่วนติดต่อผู้ใช้งาน (User Interface Design)		■	■		
พัฒนา Frontend ของระบบสำหรับจัดการและแสดงข้อมูลครุภัณฑ์			■	■	
พัฒนา Backend และ REST API สำหรับจัดการข้อมูล			■	■	
ทดสอบระบบและปรับปรุงข้อผิดพลาด				■	■
จัดทำรายงานและสรุปผลโครงงาน					■

1.7 บทบาทและความรับผิดชอบ

1. ณัฐนันทน์ ตาตะนันทน์ รหัส 650610760

มีบทบาทเป็น Project Manager รับผิดชอบในการวางแผนและประสานงานการดำเนินงานของโครงงานให้เป็นไปตามแผนที่กำหนด รวมถึงติดตามความคืบหน้าของการพัฒนาระบบ นอกจากนี้ยังทำหน้าที่เป็น Back-

end Developer ในการพัฒนาและออกแบบการทำงานของระบบฝั่งเซิร์ฟเวอร์ และทำหน้าที่เป็น Tester ในการทดสอบระบบเพื่อตรวจสอบความถูกต้องของการทำงานและแก้ไขข้อผิดพลาดที่เกิดขึ้น

2. บวรภัค ดวงจันทร์ รหัส 650610779

มีบทบาทเป็น Designer รับผิดชอบในการออกแบบส่วนติดต่อผู้ใช้งานของระบบให้มีความเหมาะสมและใช้งานได้สะดวก รวมถึงทำหน้าที่เป็น Frontend Developer ในการพัฒนาและปรับปรุงส่วนการแสดงผลของระบบบนเว็บไซต์ให้สามารถทำงานร่วมกับระบบ Backend ได้อย่างถูกต้อง

3. พชร จารุรักษ์สกุล รหัส 650610789

มีบทบาทเป็น Designer โดยรับผิดชอบในการออกแบบรูปแบบและโครงสร้างของหน้าจอภายในระบบให้มีความสอดคล้องและเหมาะสมกับการใช้งานของผู้ใช้ นอกจากนี้ยังทำหน้าที่เป็น Frontend Developer ในการพัฒนาส่วนติดต่อผู้ใช้งาน

4. ศิรวิชญ์ ศิรบรรจงกราน รหัส 650610810

มีบทบาทเป็น Database Designer รับผิดชอบในการออกแบบโครงสร้างฐานข้อมูลของระบบให้สามารถจัดเก็บข้อมูลครุภัณฑ์และข้อมูลที่เกี่ยวข้องได้อย่างมีประสิทธิภาพ นอกจากนี้ยังทำหน้าที่เป็น Backend Developer ในการพัฒนาการทำงานของระบบฝั่งเซิร์ฟเวอร์ และทำหน้าที่เป็น Tester ในการทดสอบระบบเพื่อให้ระบบสามารถทำงานได้อย่างถูกต้อง

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

ผลกระทบด้านสังคม

- ช่วยเพิ่มความโปร่งใสในการบริหารจัดการครุภัณฑ์ของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน
- ข้อมูลถูกจัดเก็บในฐานข้อมูลกลาง สามารถตรวจสอบย้อนหลังได้
- ลดความซ้ำซ้อนและความคลาดเคลื่อนของข้อมูลระหว่างหน่วยงาน
- สนับสนุนการบริหารงานบนพื้นฐานของข้อมูล (Data-Driven Management)

ผลกระทบด้านสุขภาพ

- ลดภาระงานจากการจัดการเอกสารจำนวนมาก
- ลดความเครียดจากการทำงานที่ต้องจัดการข้อมูลจากหลายแหล่ง
- ระบบแจ้งซ่อมช่วยลดความเสี่ยงจากการใช้งานอุปกรณ์ที่ชำรุด
- ส่งเสริมสภาพแวดล้อมการทำงานที่ปลอดภัยมากขึ้น

ผลกระทบด้านความปลอดภัย

- มีระบบกำหนดสิทธิ์การเข้าถึงข้อมูลตามบทบาทของผู้ใช้งาน

- มีการบันทึกประวัติการดำเนินการ (Audit Log)
- ลดความเสี่ยงจากการเข้าถึงหรือแก้ไขข้อมูลโดยไม่ได้รับอนุญาต
- เพิ่มความมั่นคงปลอดภัยของข้อมูลในระดับระบบ

ผลกระทบด้านกฎหมาย

- สนับสนุนการปฏิบัติตามระเบียบการบริหารจัดการครุภัณฑ์ของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน
- รองรับกระบวนการตรวจสอบทั้งภายในและภายนอก
- ข้อมูลมีโครงสร้างและสามารถตรวจสอบหลักฐานการดำเนินการได้
- ลดความเสี่ยงจากข้อพิพาทหรือความผิดพลาดในการจัดการข้อมูล

ผลกระทบด้านวัฒนธรรมองค์กร

- ส่งเสริมค่านิยมด้านความรับผิดชอบและความโปร่งใส
- สนับสนุนการใช้ทรัพยากรขององค์กรอย่างคุ้มค่า
- เปิดโอกาสให้ผู้ใช้งานมีส่วนร่วมในการดูแลครุภัณฑ์
- สะท้อนการปรับตัวขององค์กรสู่การใช้เทคโนโลยีดิจิทัล

1.9 แผนการใช้งบประมาณ

1. ค่าใช้จ่ายในการใช้งาน GitHub Copilot จำนวน 1 บัญชี ราคา 300 บาทต่อเดือน เป็นระยะเวลา 3 เดือน รวมเป็นเงินทั้งสิ้น 900 บาท
2. ค่าใช้จ่ายในการจัดพิมพ์โปสเตอร์ เป็นจำนวนเงิน 300 บาท
3. ค่าใช้จ่ายในการใช้งาน Claude Code จำนวน 2 บัญชี ราคา 650 บาทต่อเดือน เป็นระยะเวลา 3 เดือน รวมเป็นเงินทั้งสิ้น 3,900 บาท

งบประมาณรวมทั้งสิ้น 5100 บาท

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 สถาปัตยกรรมระบบแบบ Client–Server

ระบบบริหารจัดการครุภัณฑ์ถูกพัฒนาในรูปแบบ Web Application โดยอาศัยสถาปัตยกรรมแบบ Client–Server [16] ซึ่งเป็นรูปแบบที่แยกส่วนการทำงานออกเป็นส่วนผู้ใช้งาน (Client) และส่วนประมวลผลหลักของระบบ (Server) แนวคิดนี้ช่วยให้สามารถควบคุมความปลอดภัยของข้อมูลจากฝั่งเซิร์ฟเวอร์ ลดภาระการประมวลผลของอุปกรณ์ผู้ใช้ และสามารถบริหารจัดการฐานข้อมูลแบบรวมศูนย์ได้อย่างมีประสิทธิภาพ ซึ่งเหมาะสมกับระบบที่มีผู้ใช้งานหลายหน่วยงาน

2.2 หลักการออกแบบฐานข้อมูลเชิงสัมพันธ์

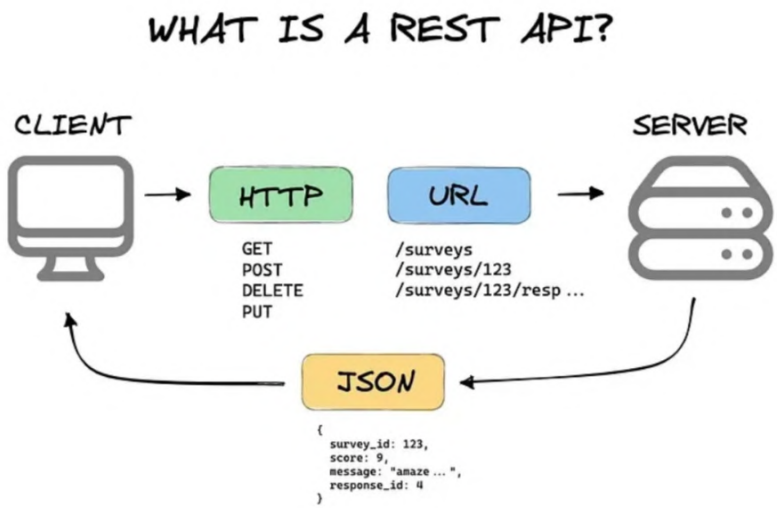
ระบบจัดเก็บข้อมูลครุภัณฑ์อาศัยแนวคิดของระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) [17] โดยข้อมูลถูกจัดเก็บในรูปแบบตารางที่มีความสัมพันธ์กันผ่าน Primary Key และ Foreign Key ในการออกแบบฐานข้อมูล โครงงานได้นำหลักการทำ Normalization มาใช้เพื่อลดความซ้ำซ้อนของข้อมูล และป้องกันปัญหาความไม่สอดคล้องของข้อมูล (Data Inconsistency) ซึ่งมีความสำคัญอย่างยิ่งในระบบที่ต้องจัดเก็บข้อมูลจากหลายหน่วยงาน

2.3 แนวคิด Role-Based Access Control (RBAC)

แนวคิด Role-Based Access Control [18] เป็นแนวคิดเพื่อรักษาความมั่นคงปลอดภัยของข้อมูล ระบบได้นำแนวคิด Role-Based Access Control มาใช้ในการกำหนดสิทธิ์การเข้าถึงข้อมูล โดยกำหนดสิทธิ์ตามบทบาทของผู้ใช้งาน เช่น ผู้ดูแลระบบ เจ้าหน้าที่ และผู้ใช้งานครุภัณฑ์ หลักการนี้ช่วยลดความเสี่ยงจากการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต และเพิ่มความชัดเจนในการกำหนดความรับผิดชอบของแต่ละบทบาทภายในระบบ

2.4 สถาปัตยกรรม RESTful API

REST (Representational State Transfer) เป็นรูปแบบสถาปัตยกรรมซอฟต์แวร์สำหรับการพัฒนาระบบให้บริการบนเครือข่าย (Web Services) โดยใช้โปรโตคอล HTTP ในการสื่อสาร [19] ในโครงงานนี้ได้ประยุกต์ใช้ RESTful API เป็นตัวกลางในการรับส่งข้อมูลระหว่างส่วนติดต่อผู้ใช้งาน (Frontend) และส่วนประมวลผล (Backend) โดยอาศัยเมธอดมาตรฐานของ HTTP เช่น GET สำหรับดึงข้อมูล POST สำหรับเพิ่มข้อมูล PUT สำหรับแก้ไขข้อมูล และ DELETE สำหรับลบข้อมูลครุภัณฑ์ ซึ่งช่วยให้ระบบมีความยืดหยุ่นและรองรับการขยายตัวในอนาคต



รูปที่ 2.1: ภาพรวมการทำงานของสถาปัตยกรรม RESTful API ระหว่าง Client และ Server ผ่านโปรโตคอล HTTP

2.5 แนวคิด Audit Trail และความโปร่งใสของข้อมูล

Audit Trail [20] เป็นแนวคิดด้านการบันทึกประวัติการดำเนินการของผู้ใช้งานในระบบ เช่น การเพิ่ม แก้ไข โอนย้าย หรือจำหน่ายครุภัณฑ์ ข้อมูลดังกล่าวสามารถใช้ตรวจสอบย้อนหลังได้ การบันทึกกิจกรรมอย่างเป็นระบบช่วยเสริมสร้างความโปร่งใส (Transparency) และสนับสนุนหลักธรรมาภิบาล (Good Governance) ในการบริหารจัดการทรัพย์สินขององค์กร

2.6 เทคโนโลยี QR Code ในการจัดการทรัพย์สิน

QR Code [21] เป็นรหัสสองมิติที่สามารถจัดเก็บข้อมูลและสแกนผ่านกล้องของอุปกรณ์พกพาได้อย่างรวดเร็ว การประยุกต์ใช้ QR Code ในโครงงานนี้ช่วยให้ผู้ใช้งานสามารถเข้าถึงข้อมูลครุภัณฑ์ได้ทันที ลดขั้นตอนการค้นหา และลดข้อผิดพลาดจากการกรอกข้อมูลด้วยตนเอง แนวคิดดังกล่าวสนับสนุนการเพิ่มประสิทธิภาพกระบวนการตรวจนับครุภัณฑ์และลดระยะเวลาในการดำเนินงาน

2.7 มาตรฐาน OAuth 2.0 และระบบ CMU Authorization

2.7.1 แนวคิดพื้นฐานของ OAuth 2.0

OAuth 2.0 เป็นมาตรฐานกรอบการทำงานสำหรับการให้สิทธิ์ (Authorization Framework) ที่กำหนดโดย RFC 6749 ซึ่งช่วยให้แอปพลิเคชันสามารถร้องขอสิทธิ์ในการเข้าถึงข้อมูลของผู้ใช้งานจากเซิร์ฟเวอร์ทรัพยากร (Resource Server) ได้อย่างปลอดภัย โดยที่ผู้ใช้งานไม่จำเป็นต้องเปิดเผยรหัสผ่านให้กับแอปพลิเคชันโดยตรง กระบวนการทำงานอาศัยการออก **Access Token** เพื่อใช้เป็นตัวแทนสิทธิ์การเข้าถึง ซึ่งโทเค็นดังกล่าวจะมีอายุการใช้งานจำกัดและสามารถกำหนดขอบเขต (Scope) ของการเข้าถึงได้อย่างชัดเจน

องค์ประกอบหลักของระบบ OAuth 2.0 ประกอบด้วย 4 ส่วน ได้แก่

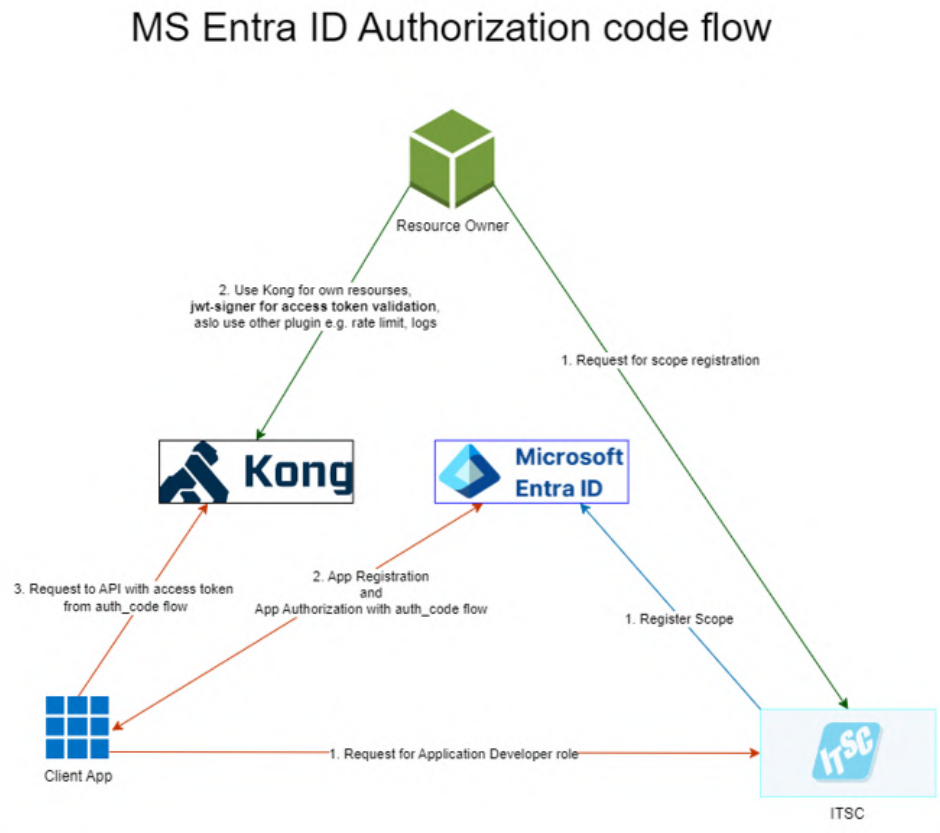
1. **Resource Owner** — ผู้ใช้งานซึ่งเป็นเจ้าของข้อมูล
2. **Client** — แอปพลิเคชันที่ต้องการร้องขอสิทธิ์เข้าถึงข้อมูล
3. **Authorization Server** — เซิร์ฟเวอร์ที่ทำหน้าที่ตรวจสอบและออกโทเค็น (เช่น Microsoft Entra ID)
4. **Resource Server** — เซิร์ฟเวอร์ที่จัดเก็บข้อมูลและให้บริการทรัพยากร (เช่น CMU API)

2.7.2 บริการ Microsoft Entra ID

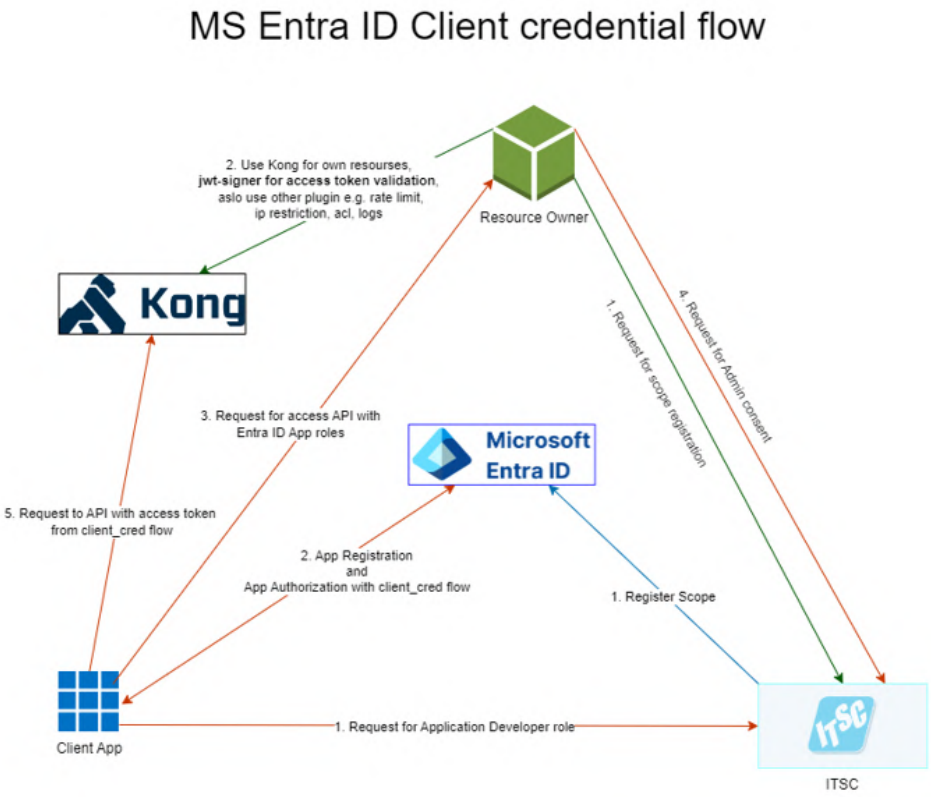
Microsoft Entra ID (เดิมชื่อ Azure Active Directory) เป็นบริการผู้ให้บริการยืนยันตัวตน (Identity Provider: IdP) บนระบบคลาวด์ (Cloud) ของบริษัทไมโครซอฟท์ ที่รองรับมาตรฐาน OAuth 2.0 และ OpenID Connect มหาวิทยาลัยเชียงใหม่ได้เลือกใช้ Microsoft Entra ID เป็นระบบจัดการข้อมูลประจำตัวและพิสูจน์ตัวตนหลัก โดยมีรหัสผู้เช่า (Tenant ID) คือ `cf81f1df-de59-4c29-91da-a2dfd04aa751`

2.7.3 ระบบ CMU Authorization

ระบบ CMU Authorization เป็นระบบบริหารจัดการสิทธิ์การเข้าถึงข้อมูลของมหาวิทยาลัยเชียงใหม่ [22] ปัจจุบันได้ดำเนินการเปลี่ยนผ่าน (Migrate) จากระบบ CMU OAuth เดิม (`oauth.cmu.ac.th`) มาใช้ Microsoft Entra ID เป็นเซิร์ฟเวอร์ให้สิทธิ์ (Authorization Server) หลัก และทำงานร่วมกับ Kong API Gateway ในการตรวจสอบ Access Token ก่อนอนุญาตให้ระบบต่าง ๆ เข้าถึงข้อมูลผ่าน CMU API



รูปที่ 2.2: ภาพรวมการทำงานของ CMU Authorization Code Flow แสดงความสัมพันธ์ระหว่าง Client App, Microsoft Entra ID, Kong API Gateway, Resource Owner และ ITSC



รูปที่ 2.3: ภาพรวมการทำงานของ CMU Client Credential Flow สำหรับการสื่อสารระหว่างเซิร์ฟเวอร์ (Server-to-Server) โดยไม่มีกระบวนการของผู้ใช้งานเข้ามาเกี่ยวข้อง

2.7.4 รูปแบบการให้สิทธิ์ (OAuth 2.0 Grant Types)

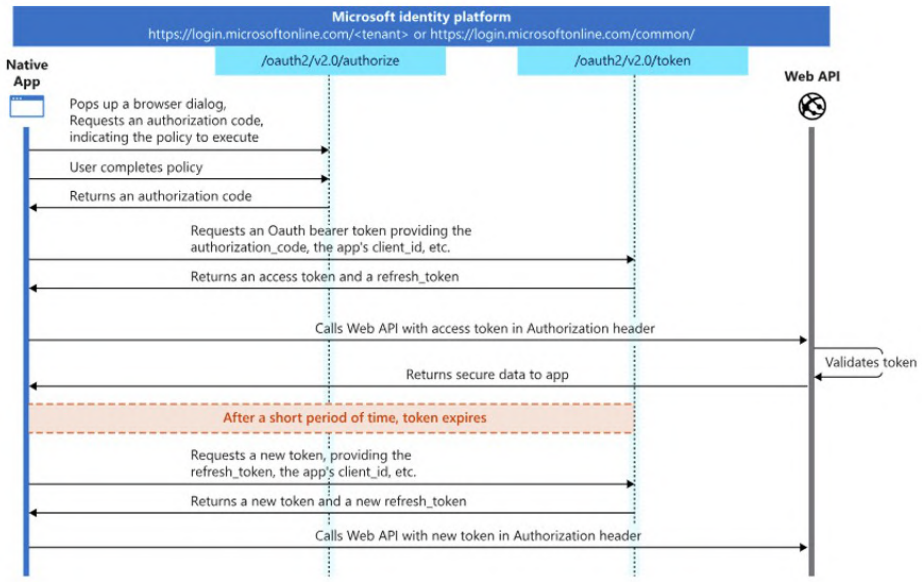
มาตรฐาน OAuth 2.0 ได้กำหนดรูปแบบการให้สิทธิ์ (Grant Type) ไว้หลายรูปแบบเพื่อรองรับสถาปัตยกรรมของแอปพลิเคชันที่แตกต่างกัน โดยระบบ CMU Authorization รองรับ Grant Type ทั้งหมด 5 รูปแบบ ดังนี้

รูปแบบ Authorization Code Grant (Client Secret)

Authorization Code Grant เป็นรูปแบบที่ได้รับความนิยมสูงสุด เหมาะสำหรับเว็บแอปพลิเคชันที่ทำงานฝั่งเซิร์ฟเวอร์ (Server-side Web Application) ซึ่งสามารถจัดเก็บ `client_secret` ได้อย่างปลอดภัย ขั้นตอนการทำงานประกอบด้วย:

1. แอปพลิเคชันนำทาง (Redirect) ผู้ใช้งานไปยัง Authorization Endpoint พร้อมกับส่งค่า `client_id`, `redirect_uri` และ `scope`
2. ผู้ใช้งานดำเนินการเข้าสู่ระบบ (Login) และให้ความยินยอม (Consent) ตามสิทธิ์ที่แอปพลิเคชันร้องขอ
3. Entra ID ส่งคืน **Authorization Code** กลับมายังแอปพลิเคชันผ่าน Redirect URI

4. แอปพลิเคชันนำ Authorization Code ไปแลกเปลี่ยนเป็น **Access Token** โดยใช้ `client_secret` เป็นข้อมูลยืนยันตัวตนผ่าน Token Endpoint
5. แอปพลิเคชันนำ Access Token ไปใช้ในการเรียกใช้งาน CMU API
6. กรณีที่ Access Token หมดอายุการใช้งาน สามารถนำ **Refresh Token** ไปร้องขอ Access Token ชุดใหม่ได้ (จำเป็นต้องระบุ scope `offline_access` ในขั้นตอนแรก)

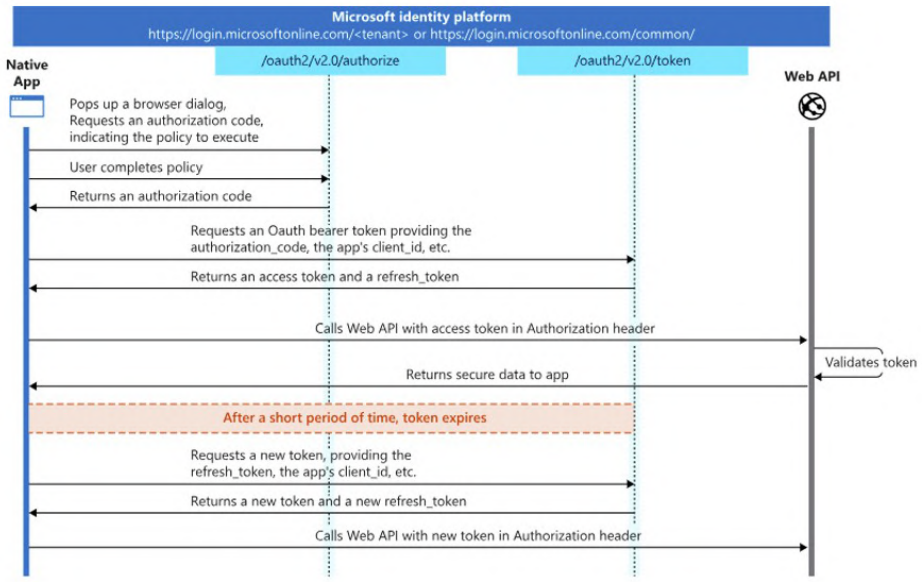


รูปที่ 2.4: แผนภาพลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Authorization Code Flow ตั้งแต่การร้องขอ Authorization Code จนถึงการใช้งาน Refresh Token

รูปแบบ Authorization Code Grant with PKCE

PKCE (Proof Key for Code Exchange, RFC 7636) เป็นกลไกความปลอดภัยเพิ่มเติมที่ออกแบบมาสำหรับไคลเอนต์สาธารณะ (Public Client) เช่น แอปพลิเคชันแบบหน้าเดียว (Single-Page Application: SPA) และแอปพลิเคชันบนสมาร์ทโฟน (Mobile Application) ที่ไม่สามารถจัดเก็บ `client_secret` ได้อย่างปลอดภัย กลไกนี้ช่วยป้องกันการโจมตีแบบดักจับรหัสสิทธิ์ (Authorization Code Interception Attack) โดยมีขั้นตอนคือ:

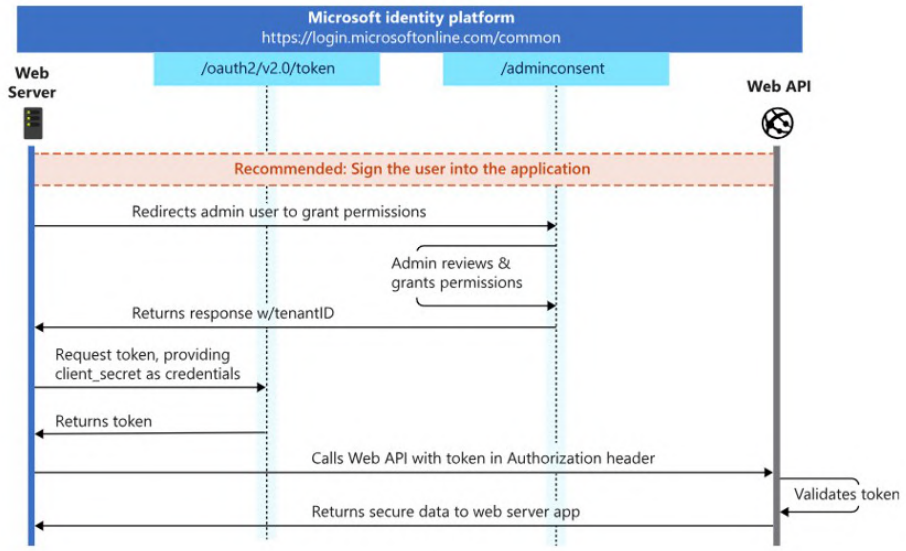
1. แอปพลิเคชันสร้างค่าสุ่ม `code_verifier` และคำนวณค่า `code_challenge` = SHA256(`code_verifier`)
2. แอปพลิเคชันแนบ `code_challenge` ไปพร้อมกับการร้องขอสิทธิ์ (Authorization Request)
3. ในขั้นตอนการแลกเปลี่ยนโทเค็น แอปพลิเคชันจะต้องส่ง `code_verifier` ไปให้ Authorization Server เพื่อตรวจสอบความถูกต้องและยืนยันตัวตน



รูปที่ 2.5: แผนภาพลำดับแสดงขั้นตอนการทำงานของ Authorization Code Flow with PKCE สำหรับ Public Client ที่ไม่สามารถจัดเก็บ Client Secret ได้

รูปแบบ Client Credentials Grant

Client Credentials Grant ถูกออกแบบมาสำหรับการสื่อสารระหว่างระบบ (Machine-to-Machine หรือ Server-to-Server) ที่ไม่มีบริบทของผู้ใช้งานเข้ามาเกี่ยวข้อง แอปพลิเคชันจะใช้ `client_id` และ `client_secret` ของตนเองในการร้องขอ Access Token จาก Token Endpoint โดยตรง การใช้งาน Grant Type รูปแบบนี้จำเป็นต้องใช้ **Application permissions** ซึ่งต้องได้รับการอนุมัติสิทธิ์ระดับผู้ดูแลระบบ (**Admin Consent**) จากทางสำนักบริการเทคโนโลยีสารสนเทศ (ITSC) ก่อนจึงจะสามารถใช้งานได้

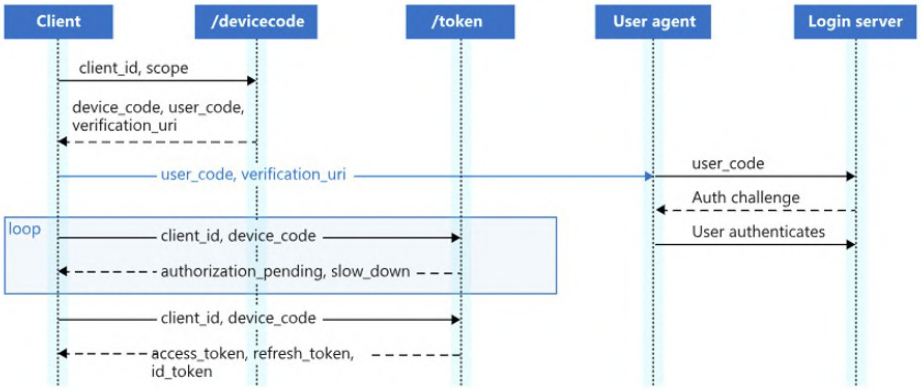


รูปที่ 2.6: แผนภาพลำดับแสดงขั้นตอนการทำงานของ Client Credentials Flow เริ่มตั้งแต่การขอ Admin Consent ไปจนถึงการเรียกใช้งาน Protected API

รูปแบบ Device Code Flow

Device Code Flow ออกแบบมาเพื่อรองรับอุปกรณ์ที่ไม่มีเว็บเบราว์เซอร์ หรือมีข้อจำกัดด้านการป้อนข้อมูลนำเข้า (Input) เช่น สมาร์ททีวี (Smart TV), เครื่องมือคอมมานด์ไลน์ (CLI Tools) หรืออุปกรณ์ IoT กระบวนการทำงานแบ่งออกเป็น 2 ส่วนที่ทำงานควบคู่กัน ได้แก่:

1. **ช่องทางอุปกรณ์ (Device Channel):** แอปพลิเคชันจะร้องขอ `device_code` และ `user_code` จาก Device Authorization Endpoint จากนั้นจะทำการตรวจสอบ (Poll) ไปยัง Token Endpoint เป็นระยะอย่างต่อเนื่องจนกว่าจะได้รับ Access Token
2. **ช่องทางผู้ใช้งาน (User Channel):** ผู้ใช้งานใช้เว็บเบราว์เซอร์บนอุปกรณ์เครื่องอื่น นำทางไปยัง `microsoft.com/devicelogin` พร้อมทั้งกรอกรหัส `user_code` เพื่อดำเนินการเข้าสู่ระบบและให้ความยินยอม



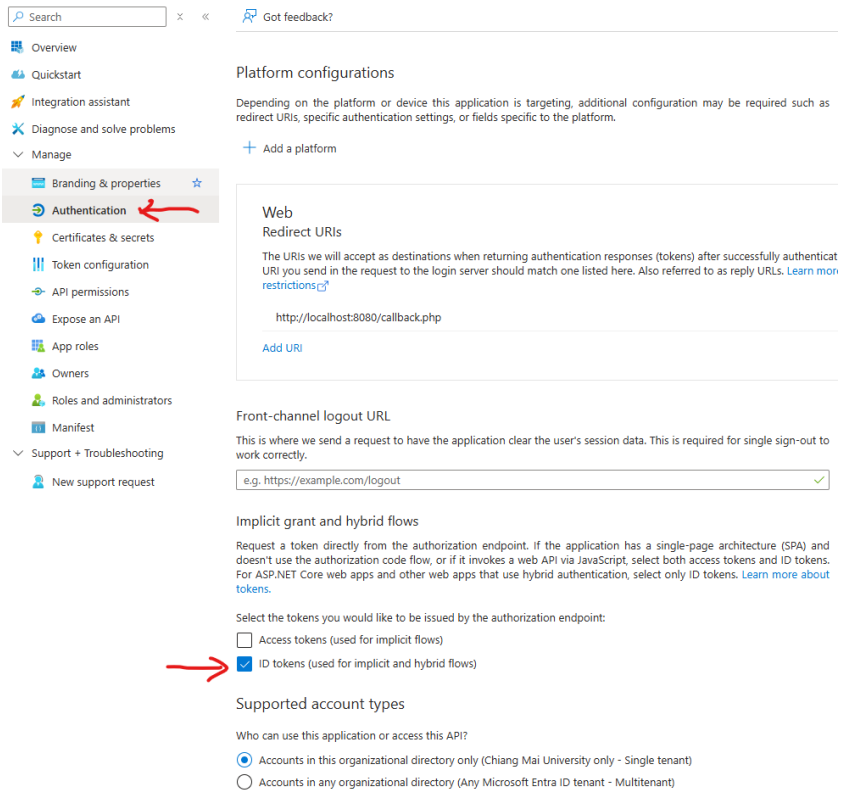
รูปที่ 2.7: แผนภาพลำดับแสดงขั้นตอนการทำงานของ Device Code Flow โดยแสดงการทำงานที่ขนานกันระหว่าง Device Channel และ User Channel

รูปแบบ Hybrid Flow

Hybrid Flow เป็นการผสมผสานการทำงานระหว่าง Authorization Code Flow และ OpenID Connect โดยแอปพลิเคชันจะได้รับทั้ง **Authorization Code** และ **ID Token** กลับมาพร้อมกันในขั้นตอน Authorization Response ซึ่งช่วยให้แอปพลิเคชันสามารถระบุตัวตนของผู้ใช้งานได้ทันทีโดยไม่ต้องรอแลกโทเค็น การใช้งานรูปแบบนี้จำเป็นต้องตั้งค่าเปิดใช้งาน **ID tokens** ใน Platform configurations ของ Entra ID และเพิ่ม `openid` ลงใน Scope

พารามิเตอร์ที่แตกต่างจาก Authorization Code Flow แบบปกติ ประกอบด้วย:

- `response_type=code id_token`
- `response_mode=form_post`
- `nonce={ค่าสุ่ม}` — เพื่อใช้ป้องกันการโจมตีแบบ Replay Attack

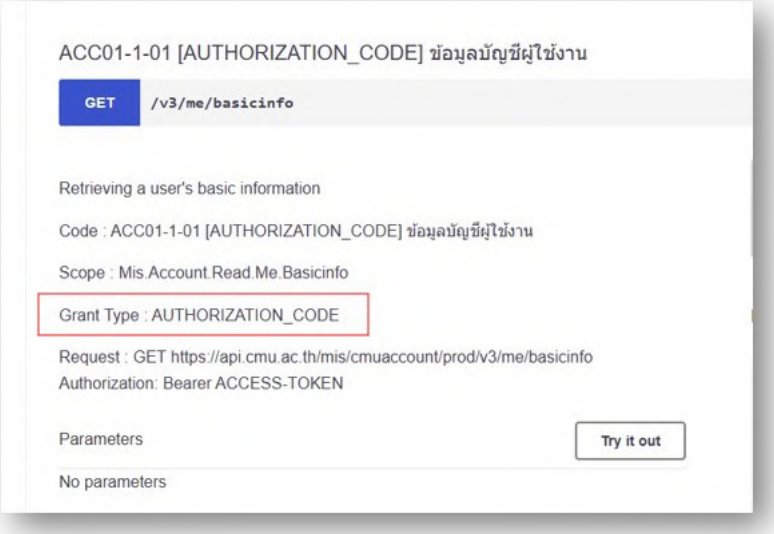


รูปที่ 2.8: การตั้งค่า Implicit grant and hybrid flows ใน Microsoft Entra ID เพื่อเปิดใช้งาน ID tokens สำหรับใช้งาน Hybrid Flow

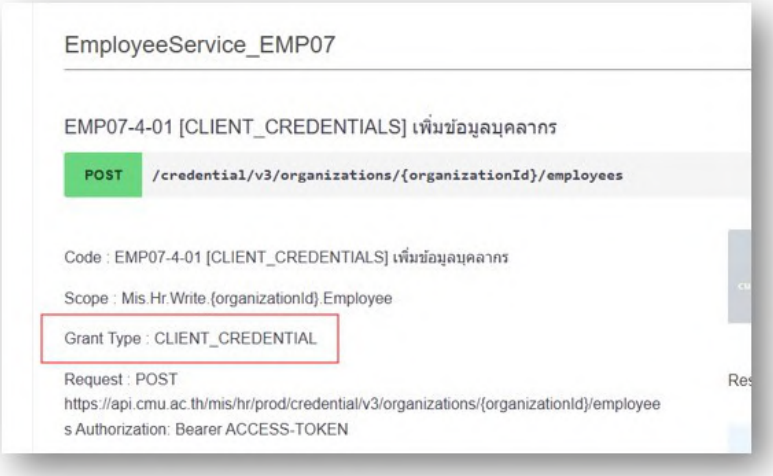
2.7.5 ขอบเขตและสิทธิ์การเข้าถึง (Scope and Permission)

Scope ในบริบทของ OAuth 2.0 คือกลไกการกำหนดขอบเขตสิทธิ์การเข้าถึงข้อมูลของแอปพลิเคชัน ระบบ CMU Authorization แบ่งประเภทของ Permission ออกเป็น 2 กลุ่มหลัก ได้แก่:

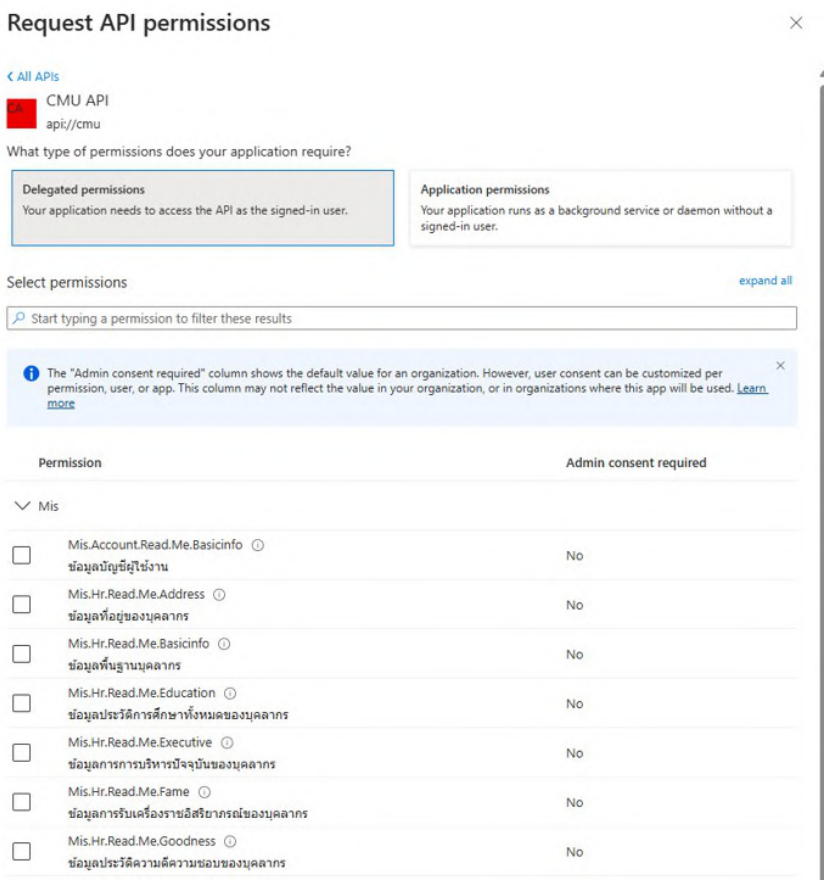
1. **Delegated permissions** — ใช้งานร่วมกับ Authorization Code Grant โดย Access Token จะถูกนำไปใช้ดึงข้อมูลในนามของผู้ใช้งาน (User) ที่เข้าสู่ระบบ และผู้ใช้งานจะต้องเป็นผู้กดยินยอมสิทธิ์ (Consent) ด้วยตนเอง
   ตัวอย่าง Scope: `api://cmu/Mis.Account.Read.Me.Basicinfo`
2. **Application permissions** — ใช้งานร่วมกับ Client Credentials Grant โดย Access Token จะเข้าถึงข้อมูลในระดับแอปพลิเคชันโดยไม่มีบริบทของผู้ใช้งาน (User Context) ซึ่งจำเป็นต้องได้รับการอนุมัติ (Admin Consent) จากระบบส่วนกลางก่อน
   ตัวอย่าง Scope: `api://cmu/.default`



รูปที่ 2.9: หน้าจอ CMU API Portal แสดงการกำหนด Scope `Mis.Account.Read.Me.Basicinfo` สำหรับ Grant Type ประเภท AUTHORIZATION_CODE



รูปที่ 2.10: หน้าจอ CMU API Portal แสดง Scope สำหรับ Grant Type ประเภท `CLIENT_CREDENTIAL` ซึ่งต้องได้รับการอนุมัติระดับผู้ดูแลระบบ (Admin Consent) ก่อนใช้งาน



รูปที่ 2.11: หน้าจอ Request API permissions ใน Microsoft Entra ID แสดงการแบ่งกลุ่ม Delegated permissions และ Application permissions ของระบบ CMU API

หลักการสำคัญในการร้องขอ Scope คือ **หลักการให้สิทธิ์เท่าที่จำเป็น (Least Privilege)** โดยนักพัฒนาควรระบุ Scope ให้น้อยที่สุดเท่าที่แอปพลิเคชันจำเป็นต้องใช้งาน เพื่อลดความเสี่ยงและผลกระทบด้านความมั่นคงปลอดภัยในกรณีที่โทเค็นเกิดการรั่วไหล

2.7.6 ประเภทของโทเค็นในระบบ OAuth 2.0

ระบบ CMU Authorization มีการออกและบริหารจัดการโทเค็นหลัก 3 ประเภท ดังแสดงรายละเอียดในตารางที่ 2.1

ตารางที่ 2.1: ประเภทของ Token และวัตถุประสงค์การใช้งานในระบบ CMU Authorization

ประเภท Token	วัตถุประสงค์การใช้งาน	เงื่อนไขการขอรับ
Access Token	นำไปใช้แนบเพื่อเรียก Protected API (มีอายุการใช้งานสั้น)	ได้รับจากทุก Grant Type
Refresh Token	ใช้สำหรับยื่นขอ Access Token ชุดใหม่เมื่อชุดเดิมหมดอายุ	ต้องระบุ scope `offline_access`
ID Token	โทเค็น JWT สำหรับระบุข้อมูลพื้นฐานของผู้ใช้งาน (OpenID)	ได้รับผ่านกลไก Hybrid Flow

2.7.7 การเปลี่ยนผ่านระบบจาก CMU OAuth สู่ Microsoft Entra ID

มหาวิทยาลัยเชียงใหม่ได้ดำเนินการปรับปรุงสถาปัตยกรรมระบบพิสูจน์ตัวตน จากระบบ CMU OAuth เดิม มาสู่แพลตฟอร์ม Microsoft Entra ID ส่งผลให้มีการเปลี่ยนแปลงค่าพารามิเตอร์หลักสำหรับการตั้งค่าแอปพลิเคชัน ดังรายละเอียดในตารางที่ 2.2

ตารางที่ 2.2: เปรียบเทียบการเปลี่ยนแปลงค่าพารามิเตอร์ระหว่างระบบ CMU OAuth เดิมและ Microsoft Entra ID

พารามิเตอร์	**ระบบ CMU OAuth เดิม**	**ระบบ Microsoft Entra ID (ปัจจุบัน)**
Authorization Endpoint	`oauth.cmu.ac.th/v2/Authorize.aspx`	`login.microsoftonline.com/cf81f1df.../oauth2/v2.0/authorize`
Token Endpoint	`oauth.cmu.ac.th/v2/GetToken.aspx`	`login.microsoftonline.com/cf81f1df.../oauth2/v2.0/token`
Scope	`cmuitaccount.basicinfo`	`api://cmu/Mis.Account.Read.Me.Basicinfo`
Basic Info API	`misapi.cmu.ac.th/.../basicinfo`	`api.cmu.ac.th/mis/cmuaccount/prod/v3/me/basicinfo`

2.8 โครงสร้างกลุ่มภารกิจศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน (SCMC)

ในการพัฒนาระบบบริหารจัดการครุภัณฑ์ (AssetManager) โครงงานได้นำร่องศึกษาและอ้างอิงโครงสร้างข้อมูลจาก ศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน (Smart City Management Center: SCMC) ของมหาวิทยาลัยเชียงใหม่ เพื่อนำมาออกแบบระบบจัดการหน่วยงานแบบลำดับชั้น (Institution Hierarchy) ภายในแอปพลิเคชัน เนื่องจาก SCMC มีการแบ่งโครงสร้างกลุ่มภารกิจและหน่วยงานย่อยที่รับผิดชอบดูแลครุภัณฑ์ที่หลากหลายและมีความซับซ้อน

โครงสร้างกลุ่มภารกิจของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน (SCMC) ประกอบด้วย 6 กลุ่มภารกิจหลัก ดังนี้:

1. **กลุ่มภารกิจงานจราจรและความปลอดภัย (Traffic and Security Management)** ประกอบด้วยหน่วยงานและกลุ่มจัดเก็บครุภัณฑ์ย่อย ได้แก่
    - ครุภัณฑ์ของ SCMC (ขสมช.)
    - ครุภัณฑ์ของ SCMC (รปภ.)
    - ครุภัณฑ์จัดเป็นวัสดุ SCMC (ขสมช.)
    - ครุภัณฑ์จัดเป็นวัสดุ SCMC (รปภ.)
    - ครุภัณฑ์จากกองพัฒนา
    - งานการบริหารจัดการคมนาคมขนส่ง

- งานการรักษาความปลอดภัยและจราจร

2. **กลุ่มภารกิจงานบริการ (Service Management)**

    - งานอาคารสถานที่ (Facilities Management)

3. **กลุ่มภารกิจงานสารสนเทศ (Information Technology Management)**

    - ครุภัณฑ์ของ SCMC (งานสารสนเทศ)
    - ครุภัณฑ์จัดเป็นวัสดุ SCMC (งานสารสนเทศ)
    - งานบริหารจัดการข้อมูลสารสนเทศ
    - งานบริหารจัดการข้อมูลสารสนเทศ 2

4. **กลุ่มภารกิจสถาปัตยกรรมและโครงสร้างวิศวกรรม (Architecture and Engineering)**

    - งานออกแบบ (Design)

5. **กลุ่มภารกิจสนับสนุนและจัดการงานทั่วไป (General Administration and Support)**

    - การเงิน
    - การเจ้าหน้าที่
    - พัสดุ
    - สารบรรณ
    - ส่วนกลาง

6. **กลุ่มภารกิจโครงสร้างพื้นฐาน (Infrastructure Management)**

    - ช่างยนต์ (Mechanical)
    - ไฟฟ้า (Electrical)
    - โทรศัพท์ (Telephone)
    - ประปาผลิต (Water Production)
    - ประปาซ่อมบำรุง (Plumbing Maintenance)
    - น้ำเสีย (Waste Water)
    - ธุรการงานสาธารณูปโภค (Utilities Administrative Officer)

ความหลากหลายของกลุ่มภารกิจและหน่วยงานย่อยเหล่านี้ สะท้อนให้เห็นถึงความจำเป็นในการมีระบบฐานข้อมูลส่วนกลางที่สามารถกำหนดสิทธิ์ผู้ดูแล (Admin) ประจำแต่ละหน่วยงานย่อย และสามารถยุบรวม (Roll-up) ข้อมูลเพื่อดูภาพรวมในระดับศูนย์ฯ (SuperAdmin) ได้อย่างมีประสิทธิภาพ

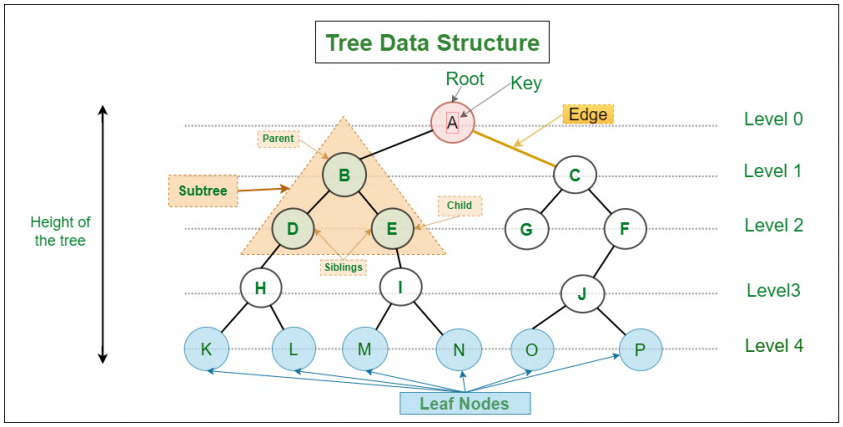
2.9 โครงสร้างข้อมูลแบบต้นไม้ (Tree Data Structure)

เนื่องจากระบบบริหารจัดการครุภัณฑ์ต้องรองรับการจัดเก็บข้อมูลหน่วยงานที่มีความซับซ้อนและมีลำดับชั้น (Hierarchy) ดังเช่นโครงสร้างของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน (SCMC) โครงงานจึงได้นำทฤษฎีโครงสร้างข้อมูลแบบต้นไม้ (Tree Data Structure) [23] มาเป็นแนวทางในการออกแบบสถาปัตยกรรมข้อมูล

โครงสร้างข้อมูลแบบต้นไม้ เป็นโครงสร้างข้อมูลแบบไม่เป็นเชิงเส้น (Non-linear Data Structure) ที่ใช้จำลองความสัมพันธ์ของข้อมูลแบบลำดับชั้น โดยข้อมูลแต่ละตัวจะถูกเก็บไว้ในสิ่งที่เรียกว่า ”โหนด (Node)” และมีความสัมพันธ์กันในลักษณะของ ”พ่อ-ลูก (Parent-Child)” องค์ประกอบที่สำคัญของโครงสร้างต้นไม้ประกอบด้วย:

- **โหนดราก (Root Node):** โหนดแรกสุดที่อยู่บนสุดของโครงสร้าง ซึ่งไม่มีโหนดพ่อ (Parent) ในบริบทของระบบนี้คือหน่วยงานระดับสูงสุด เช่น ศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน (SCMC)
- **โหนดภายใน (Internal Node):** โหนดที่มีทั้งโหนดพ่อและโหนดลูก เปรียบเสมือนกลุ่มภารกิจหลักที่มีหน่วยงานย่อยอยู่ภายใต้การดูแล
- **โหนดใบ (Leaf Node):** โหนดที่อยู่ปลายสุดและไม่มีโหนดลูก (Child) เปรียบเสมือนหน่วยงานย่อยระดับล่างสุดที่ดูแลจัดการครุภัณฑ์โดยตรง
- **เส้นเชื่อม (Edge):** เส้นที่แสดงความสัมพันธ์ระหว่างโหนด แสดงถึงสายการบังคับบัญชาหรือการสืบทอดสิทธิ์ (Permission Inheritance)

การนำโครงสร้างข้อมูลแบบต้นไม้มาประยุกต์ใช้ในการออกแบบฐานข้อมูล (เช่น รูปแบบ Adjacency List ที่แต่ละแถวในตารางจะเก็บรหัสของหน่วยงานแม่) ช่วยให้ระบบสามารถจัดหมวดหมู่หน่วยงานได้อย่างเป็นระเบียบ รองรับการขยายตัวขององค์กรในอนาคต (Scalability) และทำให้การค้นหา หรือการสรุปผลข้อมูลครุภัณฑ์แบบรวบยอด (Data Roll-up) จากหน่วยงานย่อยขึ้นไปยังหน่วยงานระดับบน สามารถทำได้อย่างถูกต้องและมีประสิทธิภาพ



รูปที่ 2.12: แผนภาพแสดงลักษณะของโครงสร้างข้อมูลแบบต้นไม้ (Tree Data Structure) ประกอบด้วย Root Node, Internal Node และ Leaf Node

2.10 ทฤษฎีการทดสอบระบบ (System Testing)

การทดสอบระบบเป็นขั้นตอนสำคัญในกระบวนการวิศวกรรมซอฟต์แวร์ [24] เพื่อตรวจสอบและยืนยันว่าระบบที่พัฒนาขึ้นสามารถทำงานได้ถูกต้องตามความต้องการและปราศจากข้อผิดพลาด โดยทฤษฎีการทดสอบระบบที่เกี่ยวข้องและนำมาประยุกต์ใช้ในโครงงานนี้ แบ่งออกเป็น 2 แนวทางหลัก ได้แก่

2.10.1 การทดสอบแบบกล่องดำ (Black-box Testing)

การทดสอบแบบกล่องดำ (Black-box Testing) หรือการทดสอบเชิงฟังก์ชัน (Functional Testing) เป็นการทดสอบซอฟต์แวร์โดยพิจารณาจากพฤติกรรมการทำงานภายนอกของระบบเป็นหลัก โดยผู้ทดสอบไม่จำเป็นต้องทราบถึงโครงสร้างซอร์สโค้ดหรือตรรกะการประมวลผลภายใน การทดสอบจะเน้นไปที่การจำลองการใช้งานจริง การป้อนข้อมูลนำเข้า (Input) และตรวจสอบผลลัพธ์ (Output) ว่าเป็นไปตามข้อกำหนดความต้องการ (Requirements) หรือไม่ สำหรับในโครงงานนี้ได้ประยุกต์ใช้การทดสอบแบบกล่องดำในการทดสอบส่วนติดต่อผู้ใช้งาน (User Interface) และการทดสอบการรับ-ส่งข้อมูลฝั่งเซิร์ฟเวอร์ โดยใช้เครื่องมือ Postman [11] ซึ่งใช้สำหรับทดสอบและตรวจสอบการทำงานของ API เพื่อให้มั่นใจว่าระบบสามารถแลกเปลี่ยนข้อมูลระหว่าง Frontend และ Backend ได้อย่างถูกต้องและตรงตามข้อกำหนด

2.10.2 การทดสอบแบบกล่องขาว (White-box Testing)

การทดสอบแบบกล่องขาว (White-box Testing) หรือการทดสอบเชิงโครงสร้าง (Structural Testing) เป็นกระบวนการทดสอบที่ผู้ทดสอบต้องมีความเข้าใจในโครงสร้างภายใน โค้ด และขั้นตอนวิธี (Algorithm) ของซอฟต์แวร์ การทดสอบรูปแบบนี้มีจุดมุ่งหมายเพื่อตรวจสอบความถูกต้องของตรรกะการทำงาน เส้นทางการประมวลผล (Control Flow) การทำงานของเงื่อนไข (Condition) และการวนลำ้ (Loop) ภายในระบบ ในโครงงานนี้ได้นำแนวคิดการทดสอบแบบกล่องขาวมาประยุกต์ใช้ในรูปแบบของการทำ Unit Test โดยเลือกใช้เครื่องมือ Jest [12] ซึ่งเป็น Framework สำหรับเขียน Unit Test เพื่อทดสอบความถูกต้องของฟังก์ชันในระบบ การทดสอบในระดับนี้ช่วยให้สามารถค้นพบและแก้ไขข้อผิดพลาดในระดับย่อยของระบบได้อย่างรวดเร็ว ทำให้โค้ดมีความน่าเชื่อถือและง่ายต่อการบำรุงรักษาในระยะยาว

2.11 การประเมินความสามารถในการใช้งานด้วย System Usability Scale (SUS)

นอกเหนือจากการทดสอบระบบเชิงฟังก์ชันและเชิงโครงสร้างแล้ว โครงงานนี้ยังได้นำแนวคิดแบบประเมิน System Usability Scale (SUS) [25] มาประยุกต์ใช้ในการวัดระดับความสามารถในการใช้งาน (Usability) และประเมินความพึงพอใจของผู้ใช้งานที่มีต่อระบบบริหารจัดการครุภัณฑ์ (AssetManager)

System Usability Scale (SUS) เป็นเครื่องมือมาตรฐานที่ถูกพัฒนาขึ้นโดย John Brooke ในปี ค.ศ. 1996 ซึ่งได้รับการยอมรับอย่างกว้างขวางในด้านวิศวกรรมซอฟต์แวร์และการออกแบบส่วนติดต่อผู้ใช้งาน (UX/UI) ว่าเป็นแบบประเมินที่สามารถนำไปใช้งานได้ง่ายและมีความน่าเชื่อถือสูง แบบประเมิน SUS ประกอบด้วยข้อคำถามจำนวน 10 ข้อ ที่มีลักษณะสลับข้อคำถามเชิงบวก (ข้อคี่) และเชิงลบ (ข้อคู่) เพื่อป้องกันความลำเอียงในการตอบคำถาม โดยผู้ประเมินจะต้องให้คะแนนตามระดับความเห็นด้วย 5 ระดับ (Likert Scale) ตั้งแต่คะแนน 1 (ไม่เห็นด้วยอย่างยิ่ง) ไปจนถึง 5 (เห็นด้วยอย่างยิ่ง)

การคำนวณคะแนนของ SUS จะมีขั้นตอนและวิธีการที่เป็นมาตรฐานเฉพาะ ดังนี้:

1. **ข้อคำถามเชิงบวก (ข้อคี่):** นำคะแนนที่ผู้ใช้งานประเมินได้ ลบด้วย 1 (คะแนน - 1)
2. **ข้อคำถามเชิงลบ (ข้อคู่):** นำค่า 5 ลบด้วยคะแนนที่ผู้ใช้งานประเมินได้ (5 - คะแนน)
3. **การหาผลลัพธ์รวม:** นำผลลัพธ์ที่ได้จากข้อ 1 และ 2 ของทั้ง 10 ข้อมาบวกกัน จากนั้นคูณด้วย 2.5 เพื่อปรับฐานคะแนนให้อยู่ในรูปแบบร้อยละ (คะแนนเต็ม 100 คะแนน)

ตามเกณฑ์มาตรฐานสากลของการประเมิน SUS คะแนนเฉลี่ยที่บ่งบอกว่าระบบมีความสามารถในการใช้งานในระดับที่ยอมรับได้ (Acceptable) จะต้องมีค่าตั้งแต่ 68 คะแนนขึ้นไป การนำทฤษฎีนี้มาใช้ช่วยให้โครงงานมีตัวชี้วัดที่เป็นรูปธรรมและเป็นมาตรฐานสากลในการประเมินความสำเร็จของระบบในมุมมองของผู้ใช้งานจริง

2.12 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงงาน

ในการพัฒนาโครงงานระบบบริหารจัดการครุภัณฑ์ผ่านเว็บแอปพลิเคชัน ได้มีการบูรณาการองค์ความรู้จากรายวิชาของหลักสูตรวิศวกรรมคอมพิวเตอร์อย่างเป็นระบบ เพื่อให้การออกแบบและพัฒนาระบบเป็นไปตามหลักวิชาการและสามารถนำไปใช้งานได้จริง

1. ในด้านการเขียนโปรแกรม ได้นำความรู้จากรายวิชา **261102 Computer Programming** และ **261200 Object-Oriented Programming** มาใช้ในการออกแบบโครงสร้างโปรแกรมเชิงวัตถุ การแบ่งโมดูลการทำงาน และการจัดการลำดับขั้นตอนของระบบ เพื่อให้โค้ดมีความชัดเจน เข้าใจง่าย และสามารถบำรุงรักษาได้สะดวก
2. ในด้านโครงสร้างข้อมูลและขั้นตอนวิธี จากรายวิชา **261217 Data Structures for Computer Engineers** และ **261218 Algorithms for Computer Engineers** ได้ถูกนำมาใช้ในการพัฒนากลไกการค้นหา การกรอง และการจัดเรียงข้อมูลครุภัณฑ์ รวมถึงการจัดการข้อมูลในรูปแบบรายการภายในระบบให้มีประสิทธิภาพ
3. ในด้านเครือข่ายคอมพิวเตอร์ จากรายวิชา **261332 Data and Computer Communications** และ **261335 Computer Networks** ได้ประยุกต์ใช้แนวคิดสถาปัตยกรรมแบบ Client–Server การรับ–ส่งข้อมูลผ่านโปรโตคอล HTTP และการออกแบบ RESTful API เพื่อเชื่อมต่อส่วนติดต่อผู้ใช้กับเซิร์ฟเวอร์ผ่านเครือข่ายอินเทอร์เน็ต
4. ในด้านวิศวกรรมซอฟต์แวร์ จากรายวิชา **261361 Software Engineering** ได้ถูกนำมาใช้ในการวิเคราะห์ความต้องการของผู้ใช้ การออกแบบสถาปัตยกรรมระบบ การจัดทำเอกสาร และการทดสอบระบบ เพื่อให้ซอฟต์แวร์มีคุณภาพและรองรับการขยายในอนาคต
5. ในด้านการจัดเตรียมและปรับปรุงข้อมูล ได้นำความรู้จากรายวิชา **261448 Data Mining for Computer Engineering** มาประยุกต์ใช้ในกระบวนการทำความสะอาดข้อมูล (Data Cleaning) และการเตรียมข้อมูล (Data Preprocessing) จากไฟล์ Excel เดิมของหน่วยงาน เพื่อปรับปรุงคุณภาพข้อมูล ลดความซ้ำซ้อน และเตรียมข้อมูลให้เหมาะสมก่อนนำเข้าสู่ฐานข้อมูลของระบบใหม่

6. ในด้านการพัฒนาระบบแบบ Full-Stack Development จากรายวิชา **261497 Selected Topics in Computer Software** ได้ถูกนำมาใช้ในการพัฒนาระบบเว็บแอปพลิเคชันทั้งฝั่งผู้ใช้และฝั่งเซิร์ฟเวอร์ ครอบคลุมทั้งส่วนติดต่อผู้ใช้และการจัดการข้อมูลฝั่งเซิร์ฟเวอร์ เพื่อให้ระบบสามารถทำงานได้อย่างครบถ้วนและมีประสิทธิภาพ

2.13 ความรู้นอกหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงงาน

ในการพัฒนาโครงงานระบบบริหารจัดการครุภัณฑ์ผ่านเว็บแอปพลิเคชัน คณะผู้จัดทำได้ศึกษาความรู้เพิ่มเติมนอกเหนือจากเนื้อหาที่ได้เรียนในหลักสูตร เพื่อให้สามารถพัฒนาระบบให้สอดคล้องกับเทคโนโลยีที่ใช้งานจริงในปัจจุบัน ความรู้ดังกล่าวได้มาจากการศึกษาด้วยตนเองผ่านแหล่งเรียนรู้ออนไลน์ เช่น วิดีโอการสอนบนแพลตฟอร์ม YouTube คอร์สเรียนออนไลน์ และเอกสารประกอบการใช้งานอย่างเป็นทางการ (Official Documentation) ของเทคโนโลยีและเครื่องมือที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ องค์ความรู้ที่ได้จากการศึกษาด้วยตนเองถูกนำมาประยุกต์ใช้ในการออกแบบและพัฒนาระบบเว็บแอปพลิเคชัน ทั้งในด้านการพัฒนาส่วนติดต่อผู้ใช้ การพัฒนาระบบฝั่งเซิร์ฟเวอร์ การจัดการข้อมูล และการแก้ไขปัญหาที่เกิดขึ้นระหว่างกระบวนการพัฒนา การเรียนรู้จากแหล่งความรู้ภายนอกดังกล่าวช่วยเสริมสร้างความเข้าใจในการใช้เทคโนโลยีสมัยใหม่ และทำให้คณะผู้จัดทำสามารถพัฒนาระบบให้สามารถทำงานได้อย่างมีประสิทธิภาพและเหมาะสมกับการนำไปใช้งานจริง

บทที่ 3
โครงสร้างและขั้นตอนการทำงาน

3.1 โครงสร้างและการทำงานของระบบ

3.1.1 ระบบส่วนติดต่อผู้ใช้งาน (Frontend)

ส่วน Frontend ทำหน้าที่เป็นส่วนติดต่อผู้ใช้งาน (User Interface) ของระบบ โดยพัฒนาในรูปแบบ Web Application เพื่อให้สามารถเข้าถึงผ่านเว็บเบราว์เซอร์ได้อย่างสะดวก

ระบบถูกพัฒนาโดยใช้ React ร่วมกับ TypeScript ซึ่งช่วยให้สามารถจัดการโครงสร้างของ User Interface ในรูปแบบ Component-based และลดข้อผิดพลาดในการพัฒนา นอกจากนี้ยังใช้ Tailwind CSS สำหรับการออกแบบหน้าตาและจัดรูปแบบการแสดงผลให้มีความสวยงามและรองรับการใช้งานบนอุปกรณ์ที่มีขนาดหน้าจอแตกต่างกัน (Responsive Design)

หน้าที่หลักของ Frontend ได้แก่ การแสดงข้อมูลครุภัณฑ์ในรูปแบบ Dashboard การแสดงรายงาน เช่น รายงานประจำปี การจัดการข้อมูลครุภัณฑ์ (เพิ่ม แก้ไข ลบ) รวมถึงการรองรับการสแกน QR Code เพื่อเข้าถึงข้อมูลครุภัณฑ์ และการอัปเดตสถานะการตรวจนับหรือการโอนย้าย

Frontend จะทำหน้าที่ส่งคำร้องขอ (Request) ไปยัง Backend และรับข้อมูลตอบกลับ (Response) ในรูปแบบ JSON เพื่อนำมาประมวลผลและแสดงผลให้ผู้ใช้งาน

3.1.2 ระบบจัดการหลังบ้าน (Backend)

ส่วน Backend ทำหน้าที่เป็นศูนย์กลางในการประมวลผลข้อมูล และควบคุมตรรกะการทำงานของระบบ (Business Logic)

ระบบพัฒนาโดยใช้ Node.js และ Fastify สำหรับสร้าง RESTful API ที่ใช้ในการสื่อสารกับ Frontend โดย Backend มีหน้าที่หลักในการจัดการข้อมูลต่าง ๆ ภายในระบบ เช่น การเพิ่ม แก้ไข และลบข้อมูลครุภัณฑ์ การจัดการข้อมูลผู้ใช้งานและหน่วยงาน การโอนย้ายครุภัณฑ์ และการจัดการรอบการตรวจนับครุภัณฑ์ประจำปี

นอกจากนี้ Backend ยังมีการจัดการสิทธิ์การเข้าถึงข้อมูลตามบทบาทของผู้ใช้งาน (Role-Based Access Control) เช่น User, Admin และ Superadmin เพื่อควบคุมการเข้าถึงข้อมูลและฟังก์ชันต่าง ๆ ของระบบ

ระบบยังมีการบันทึกประวัติการทำงาน (Audit Log) เพื่อให้สามารถตรวจสอบการเปลี่ยนแปลงข้อมูลย้อนหลังได้ และช่วยเพิ่มความโปร่งใสของระบบ

3.1.3 ระบบฐานข้อมูล (Database)

ระบบใช้ฐานข้อมูล PostgreSQL ซึ่งเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) สำหรับจัดเก็บข้อมูลทั้งหมดของระบบ

โครงสร้างฐานข้อมูลถูกออกแบบให้รองรับความสัมพันธ์ระหว่างข้อมูล โดยมีตารางหลัก เช่น ตารางผู้ใช้งาน (User) ตารางครุภัณฑ์ (Asset) ตารางหน่วยงาน (Institution) และตารางตำแหน่งจัดเก็บ (Object) เช่น อาคารและห้อง

นอกจากนี้ยังมีตารางที่ใช้สำหรับการดำเนินการภายในระบบ เช่น ตาราง TransferAssets สำหรับบันทึกการโอนย้ายครุภัณฑ์ ตาราง Asset_Checking สำหรับการตรวจนับครุภัณฑ์ และตาราง Logs สำหรับบันทึกประวัติการทำงานของระบบ

การออกแบบฐานข้อมูลในลักษณะเชิงสัมพันธ์ช่วยลดความซ้ำซ้อนของข้อมูล และเพิ่มความสามารถในการตรวจสอบข้อมูลย้อนหลัง (Auditability)

3.1.4 การนำระบบขึ้นใช้งาน (Deployment)

ในส่วนของ Deployment ระบบถูกออกแบบให้สามารถติดตั้งและทำงานได้ในสภาพแวดล้อมที่หลากหลาย โดยใช้เทคโนโลยี Containerization และ Web Server เพื่อเพิ่มความสะดวกในการติดตั้งและเพิ่มประสิทธิภาพของระบบ

ระบบใช้ Docker ในการสร้าง Container ของแต่ละส่วนประกอบ เช่น Frontend, Backend และ Database เพื่อให้สามารถ Deploy ได้อย่างรวดเร็วและลดปัญหาความแตกต่างของสภาพแวดล้อม

นอกจากนี้ยังใช้ Nginx ทำหน้าที่เป็น Web Server และ Reverse Proxy เพื่อจัดการการรับส่งข้อมูลระหว่าง Client และ Server และช่วยเพิ่มความปลอดภัยของระบบ

โครงสร้าง Deployment ดังกล่าวช่วยให้ระบบสามารถขยายขนาด (Scalability) ได้ง่าย รองรับผู้ใช้งานจำนวนมาก และเพิ่มความเสถียรและความปลอดภัยในการใช้งานระบบ

3.2 การกำหนดบทบาทผู้ใช้งานในระบบ

ระบบ SCMC AssetManager ใช้แนวคิดการควบคุมสิทธิ์การเข้าถึงตามบทบาทของผู้ใช้งาน (Role-Based Access Control: RBAC) เพื่อกำหนดขอบเขตการใช้งานฟังก์ชันต่าง ๆ ภายในระบบ โดยผู้ใช้งานแต่ละประเภทจะมีสิทธิ์การเข้าถึงข้อมูลและการดำเนินการที่แตกต่างกันตามหน้าที่รับผิดชอบ ในระบบนี้ ผู้ใช้งานที่มีบทบาทระดับสูงกว่าจะสามารถใช้งานฟังก์ชันทั้งหมดของบทบาทที่ต่ำกว่าได้ เช่น ผู้ดูแลระบบ (Admin) สามารถใช้งานฟังก์ชันทั้งหมดที่ผู้ใช้งานทั่วไป (User) สามารถทำได้ และผู้ดูแลระบบระดับสูง (Superadmin) สามารถใช้งานฟังก์ชันทั้งหมดของทั้ง User และ Admin รวมถึงฟังก์ชันเพิ่มเติมที่เกี่ยวข้องกับการจัดการระบบในภาพรวม

3.2.1 ผู้ใช้งานทั่วไป (User)

เป็นบุคลากรทุกคนที่มี CMU Account และเป็นผู้ใช้งานครุภัณฑ์ ต้องการเห็นรายการครุภัณฑ์ที่ตนรับผิดชอบ เข้าถึงข้อมูลผ่านการสแกน QR Code แจ้งซ่อม ยืนยันการตรวจนับ และอัปเดตข้อมูลที่เกี่ยวข้องกับครุภัณฑ์ของตนได้สะดวก

3.2.2 ผู้ดูแลระบบ (Admin)

เป็นเจ้าหน้าที่ที่รับผิดชอบดูแลครุภัณฑ์ของหน่วยงานหรือคณะ ต้องการเครื่องมือในการจัดการข้อมูลครุภัณฑ์ของหน่วยงานตนเอง เช่น อนุมัติการเปลี่ยนสถานที่/รูปภาพ ตรวจสอบคำขอโอนย้าย ติดตามสถานะการตรวจนับ และสรุปรายงานในระดับหน่วยงาน รวมถึงสามารถกำหนดหรือกระจายความเป็นเจ้าของครุภัณฑ์ให้กับผู้ใช้งานภายในหน่วยงาน เพื่อให้ผู้รับผิดชอบสามารถดูแลและอัปเดตข้อมูลครุภัณฑ์ที่เกี่ยวข้องได้

3.2.3 ผู้ดูแลระบบระดับสูง (Superadmin)

เป็นหัวหน้าฝ่ายหรือผู้ประสานงานที่ดูแลครุภัณฑ์ในระดับศูนย์/คณะ ต้องการมองภาพรวมครุภัณฑ์ทั้งหมด สร้างรอบการตรวจนับประจำปี ติดตามความคืบหน้าของแต่ละหน่วยงาน และใช้ข้อมูลจากระบบในการวางแผนงบประมาณและการตัดสินใจเชิงนโยบาย นอกจากนี้ยังมีหน้าที่ในการกำหนดสิทธิ์การใช้งานของระบบ เช่น การมอบสิทธิ์ผู้ดูแลระบบ (Admin) ให้กับผู้ที่รับผิดชอบครุภัณฑ์ในหน่วยงานย่อย สามารถนำเข้าและแก้ไขข้อมูลครุภัณฑ์ในระบบ รวมถึงกระจายความเป็นเจ้าของครุภัณฑ์ให้กับผู้ใช้งานที่เกี่ยวข้องในแต่ละหน่วยงานได้

3.3 ฟีเจอร์ของระบบ

3.3.1 ระบบการแสดงข้อมูลครุภัณฑ์ในรูปแบบ Dashboard

ระบบแสดงข้อมูลสรุปภาพรวมของครุภัณฑ์ทั้งหมดในรูปแบบแดชบอร์ด สำหรับผู้ดูแลระบบระดับสูง (Superadmin) โดยนำเสนอข้อมูลในหลายมิติดังนี้

- **การ์ดสรุปข้อมูลภาพรวม** ได้แก่ จำนวนครุภัณฑ์ทั้งหมด, มูลค่ารวม, มูลค่าเฉลี่ย, จำนวนผู้ใช้งาน, จำนวนหน่วยงาน และจำนวนอาคาร
- **กราฟสถิติจำแนกตามสถานะ** แสดงสัดส่วนครุภัณฑ์ในสถานะ
  - ใช้งานอยู่ (INUSE)
  - ส่งซ่อม (FIXING)
  - ชำรุด/เสื่อมสภาพ (USELESS)
  - รอจำหน่าย (SELLING)
- **กราฟสถิติจำแนกตามสภาพ** แสดงสัดส่วนครุภัณฑ์ที่มีสภาพ
  - ดี (GOOD)
  - พอใช้ (MEDIUM)
  - ชำรุด (BROKEN)
- **สถิติจำแนกตามประเภท** แสดงจำนวนครุภัณฑ์ประเภท
  - รายชิ้น (SINGLE)
  - ประเภทหลายรายการ (MULTI)
- **แผนผังต้นไม้หน่วยงาน** แสดงจำนวนครุภัณฑ์และมูลค่า กำกับในแต่ละหน่วยงาน
- **สถิติจำแนกตามอาคาร** แสดงจำนวนครุภัณฑ์จัดกลุ่มตาม อาคารที่จัดเก็บ
- **การวิเคราะห์เชิงมูลค่า** แสดงยอดรวม, ค่าเฉลี่ย, มูลค่าต่ำสุด และมูลค่าสูงสุด
- **กิจกรรมล่าสุด** นำเสนอรายการ Audit Log ล่าสุด จำนวน 20 รายการ

- **สถิติการโอนย้าย** แสดงยอดรวม, รอดำเนินการ, เสร็จสมบูรณ์ และยกเลิก
- **ตารางผู้ครอบครอง** แสดงรายชื่อผู้ใช้งานพร้อมจำนวน ครุภัณฑ์และมูลค่าในความดูแล

นอกจากนี้ ระบบยังรองรับการแสดงรายการครุภัณฑ์สำหรับทุกบทบาท โดย User จะเห็นเฉพาะครุภัณฑ์ที่ตนเองรับผิดชอบ, Admin จะเห็นครุภัณฑ์ทั้งหมดภายใต้โครงสร้างหน่วยงานของตน และ Superadmin จะเห็นครุภัณฑ์ทั้งหมดภายใต้หน่วยงานหลักที่ดูแล พร้อมฟังก์ชันค้นหาจากชื่อหรือรายละเอียด, กรองข้อมูล (Filter) ตามสภาพ/สถานะ/หน่วยงาน/ผู้ครอบครอง, จัดเรียง (Sort) ตามชื่อ, รหัส, มูลค่า หรือวันที่ลงทะเบียน รวมถึงระบบแบ่งหน้า (Pagination) และการแสดงคะแนนความสมบูรณ์ของข้อมูล (0–100%) ในแต่ละรายการ

3.3.2 ระบบจัดการข้อมูลครุภัณฑ์

ระบบรองรับการบริหารจัดการข้อมูลครุภัณฑ์อย่างครบวงจร ครอบคลุมการเพิ่ม, แก้ไข และลบข้อมูล โดยกำหนดสิทธิ์ตามบทบาทดังนี้

การเพิ่มครุภัณฑ์ (Create) Superadmin สามารถเพิ่มครุภัณฑ์พร้อมข้อมูลครบถ้วน ได้แก่ ชื่อ, ยี่ห้อ, รหัสครุภัณฑ์, หมายเลข, สภาพ, สถานะ, รายละเอียด, รุ่น, Serial Number, ระยะเวลารับประกัน, วันที่ลงทะเบียน, เปอร์เซ็นต์ความสมบูรณ์, ผู้ครอบครอง และหน่วยงาน สำหรับครุภัณฑ์ประเภทหลายรายการ (Multi) สามารถสร้างรายการย่อย ได้ไม่จำกัดจำนวน โดยแต่ละรายการย่อยต้องระบุอาคาร, ชั้น, ห้อง, รายละเอียดสถานที่ตั้ง และรูปภาพ ระบบจะสร้าง QR Code โดยอัตโนมัติเมื่อมีการเพิ่มข้อมูลครุภัณฑ์ใหม่ และหากกรอกชื่อหน่วยงานที่ยังไม่มีในระบบ ระบบจะเพิ่มหน่วยงาน นั้นเข้าสู่ฐานข้อมูลโดยอัตโนมัติ

การแก้ไขครุภัณฑ์ (Edit) Admin และ Superadmin สามารถแก้ไขข้อมูลครุภัณฑ์ได้ทุกฟิลด์ ส่วน User (เจ้าของ) สามารถยื่นคำขอแก้ไขสถานที่ตั้ง (อาคาร/ชั้น/ห้อง) และรูปภาพ โดยต้องรอการอนุมัติจาก Admin ซึ่ง Admin สามารถอนุมัติหรือปฏิเสธคำขอดังกล่าวได้

การลบครุภัณฑ์ (Delete) Superadmin สามารถลบข้อมูลครุภัณฑ์แบบรายชิ้นหรือพร้อมกันหลาย รายการ (Bulk Delete) ทั้งนี้การลบจะไม่ถูกบันทึกใน Audit Log

3.3.3 ระบบรายงานครุภัณฑ์ประจำปี

ระบบรองรับการสร้างรายงานสรุปครุภัณฑ์ประจำปีในรูปแบบ Snapshot สำหรับ Admin และ Superadmin โดยมีฟังก์ชันดังต่อไปนี้

- **การกรองรายงาน** สามารถกรองตามปี, หน่วยงาน และสภาพครุภัณฑ์ได้
- **การเลือกฟิลด์** ผู้ใช้สามารถเลือกได้ว่าต้องการให้ รวมแอตทริบิวต์ใดบ้างในรายงาน
- **การส่งออกข้อมูล** รองรับการส่งออกเป็นไฟล์ Excel
- **การไฮไลต์ครุภัณฑ์สำคัญ** ระบบจะไฮไลต์ครุภัณฑ์ ที่ใกล้หมดอายุรับประกัน และครุภัณฑ์ที่จัดซื้อภายในปีปัจจุบัน

- **การบันทึกและเรียกดูรายงาน** ระบบบันทึกรายงานที่เคย สร้างไว้ให้เรียกดูได้ภายหลัง และสามารถลบรายงานทิ้งได้
- **การมองเห็นตามบทบาท** Superadmin มองเห็นรายงาน ทั้งหมด ส่วน Admin จะเห็นเฉพาะรายงานของหน่วยงานตนเอง

3.3.4 ระบบโอนย้ายครุภัณฑ์

ระบบรองรับกระบวนการโอนย้ายความครอบครองครุภัณฑ์ระหว่างผู้ใช้งาน โดยมีขั้นตอนที่กำหนดให้ทุกฝ่ายที่เกี่ยวข้องต้องดำเนินการครบถ้วน จึงจะถือว่าการโอนย้ายสำเร็จ ดังนี้

1. **ผู้ส่ง (เจ้าของเดิม)** ริเริ่มการโอนย้ายโดยเลือก ครุภัณฑ์และระบุตัวผู้รับโอน
2. **ผู้รับโอน** กดยอมรับการโอนย้ายพร้อมอัปเดตสถานที่ตั้ง ใหม่ (อาคาร, ชั้น, ห้อง, รายละเอียด) และอัปโหลดภาพถ่ายปัจจุบัน
3. **Admin** กดยืนยันการโอนย้ายเป็นลำดับสุดท้าย

กรรมสิทธิ์ผู้ครอบครองจะเปลี่ยนแปลงก็ต่อเมื่อทั้ง 3 ฝ่ายดำเนินการ ครบถ้วนเท่านั้น ทั้งนี้ฝ่ายใดฝ่ายหนึ่งสามารถปฏิเสธ (Deny) การโอนย้ายได้ตลอดกระบวนการ ระบบยังรองรับการเรียกดูรายการโอนย้ายโดย User สามารถดูรายการ ที่ส่งไปและรับมา ส่วน Admin สามารถดูรายการโอนย้ายทั้งหมดภายใน โครงสร้างหน่วยงานของตน

3.3.5 ระบบการแจ้งซ่อมครุภัณฑ์

ระบบรองรับการยื่นคำร้องแจ้งซ่อมครุภัณฑ์ผ่านช่องทางออนไลน์ โดยมีฟังก์ชันดังต่อไปนี้

- User สามารถยื่นคำร้องแจ้งซ่อมสำหรับครุภัณฑ์ของตนเอง พร้อมระบุรายละเอียดอาการชำรุด
- เมื่อยื่นคำร้อง ระบบจะสร้างเอกสารแจ้งซ่อมในรูปแบบ PDF ให้โดยอัตโนมัติ
- Admin สามารถดูคำร้องแจ้งซ่อมที่รอดำเนินการทั้งหมด ภายในหน่วยงาน และกดรับเรื่อง (Accept) หรือปฏิเสธ (Deny) คำร้องได้
- User สามารถติดตามสถานะใบแจ้งซ่อมของตนเองได้
- ปุ่มแจ้งซ่อมยังปรากฏบนหน้าข้อมูลครุภัณฑ์หลังการสแกน QR Code ด้วย

3.3.6 ระบบแจ้งเตือน

ระบบมีการแจ้งเตือนแบบเรียลไทม์ผ่าน Server-Sent Events (SSE) ครอบคลุมเหตุการณ์สำคัญต่าง ๆ ดังนี้

- แจ้งเตือนไปยังผู้ครอบครองเมื่อครุภัณฑ์ถูกเพิ่ม, แก้ไข หรือลบ
- แจ้งเตือนผู้โอน, ผู้รับโอน และ Admin เมื่อมีกระบวนการ โอนย้ายเกิดขึ้น

- แจ้งเตือน Admin เมื่อผู้ใช้งานยื่นคำขอแก้ไขรูปภาพหรือ สถานที่ตั้ง
- แจ้งเตือนผู้ใช้ทุกคนเมื่อมีการสร้างหรือยืนยันเปิดรอบ การตรวจนับครุภัณฑ์
- แจ้งเตือนผู้สร้างรอบการตรวจนับเมื่อความคืบหน้าถึง 100%
- แจ้งเตือนผู้สร้างรอบการตรวจนับเมื่อรอบหมดเวลา (ทำงานผ่าน Cronjob)
- แจ้งเตือนผู้ใช้เมื่อถูกเชิญออกจากหน่วยงาน

ผู้ใช้งานสามารถจัดการการแจ้งเตือนได้ด้วยตนเอง ได้แก่ กดทำเครื่องหมายว่าอ่านแล้ว (Mark as Read), ลบแต่ละรายการ และลบทั้งหมดพร้อมกัน (Clear All) นอกจากนี้ยังมีไอคอนตัวเลข แสดงจำนวนข้อความที่ยังไม่ได้อ่านบนแถบเมนูด้านบน รวมถึงรองรับการส่งแจ้งเตือนผ่านอีเมลสำหรับฟีเจอร์เร่งรัด (Reminder) ผู้ที่ยังไม่ตรวจนับครุภัณฑ์โดย Superadmin

3.3.7 ระบบนับรอบการตรวจครุภัณฑ์

ระบบนับรอบการตรวจครุภัณฑ์ (Asset Checking Cycle) เป็นระบบที่ถูกออกแบบมา เพื่ออำนวยความสะดวกและทดแทนกระบวนการตรวจนับครุภัณฑ์ประจำปีแบบเดิมที่พึ่งพาไฟล์เอกสาร โดยเปลี่ยนมาใช้การดำเนินการผ่านระบบดิจิทัลแบบรวมศูนย์ ซึ่งช่วยลดข้อผิดพลาด ลดความซ้ำซ้อน และเพิ่มความสามารถในการติดตามผลได้อย่างมีประสิทธิภาพ โดยมีฟังก์ชันการทำงานหลักดังนี้

- **การสร้างและจัดการรอบตรวจนับ (Cycle Creation & Management):** ผู้ดูแลระบบระดับสูง (Superadmin) และผู้ดูแลระบบ (Admin) สามารถสร้างรอบการตรวจนับประจำปีของหน่วยงานตนเองและหน่วยงานย่อยภายใต้สังกัดได้ โดยสามารถกำหนดช่วงเวลาเริ่มต้นและสิ้นสุดของการตรวจนับ ระบบมีกลไกตรวจสอบความถูกต้อง (Validation) เช่น การป้องกันการสร้างรอบการตรวจนับที่มีช่วงเวลาทับซ้อนกัน (Overlap) ภายในหน่วยงานเดียวกัน
- **การจัดการลำดับขั้นและการควบรวมรอบ (Hierarchy & Absorption):** ระบบรองรับการจัด-การรอบตรวจนับตามโครงสร้างของหน่วยงาน หากหน่วยงานระดับบน (Parent Institution) ทำการยืนยันเพื่อเริ่มรอบการตรวจนับ (Start Cycle) ระบบจะทำการควบรวม (Absorb) รอบการตรวจนับของหน่วยงานระดับล่าง (Child Institution) ที่มีช่วงเวลาทับซ้อนกันเข้ามาโดยอัตโนมัติ เพื่อให้สามารถบริหารจัดการและดูภาพรวมจากส่วนกลางได้
- **การดำเนินการตรวจนับครุภัณฑ์ (Checking Execution):** เมื่อรอบการตรวจนับอยู่ในสถานะเปิดใช้งาน (Active) ผู้ใช้งาน (User) จะสามารถดำเนินการตรวจนับครุภัณฑ์ที่ตนเองเป็นเจ้าของได้ทั้งผ่านหน้าเว็บไซต์และการสแกนคิวอาร์โค้ด (QR Code) ประจำครุภัณฑ์ โดยผู้ใช้สามารถอัปเดตสภาพการใช้งานและสถานะปัจจุบันของครุภัณฑ์ชิ้นนั้น ๆ ได้ทันที ซึ่งทุกการดำเนินการจะถูกบันทึกไว้ในระบบประวัติการทำรายการ (Audit Log)
- **การติดตามความคืบหน้า (Progress Monitoring):** ผู้ดูแลระบบสามารถติดตามความคืบหน้าของการตรวจนับได้แบบเรียลไทม์ (Real-time Progress) ผ่านแดชบอร์ด โดยระบบจะคำนวณและแสดงสัดส่วนเปอร์เซ็นต์ จำนวนครุภัณฑ์ที่ตรวจนับแล้ว และจำนวนที่ยังไม่ได้ตรวจนับ ซึ่งข้อมูลเหล่านี้จะถูกคำนวณใหม่โดยอัตโนมัติหากมีการเพิ่ม ลบ หรือโอนย้ายครุภัณฑ์ระหว่างรอบการตรวจนับ

- **การติดตามผู้ใช้และการแจ้งเตือน (Unchecked Users & Reminders):** ผู้ดูแลระบบระดับสูงสามารถเรียกดูรายชื่อผู้ใช้งานที่ยังดำเนินการตรวจนับครุภัณฑ์ในความรับผิดชอบไม่ครบถ้วนได้ และสามารถส่งข้อความแจ้งเตือน (Send Reminder) ไปยังกลุ่มผู้ใช้งานดังกล่าวผ่านระบบแจ้งเตือนภายใน (Notification) และทางอีเมล เพื่อเร่งรัดให้การตรวจนับแล้วเสร็จตามกำหนดเวลา

3.3.8 ระบบจัดการข้อมูลหน่วยงาน

ระบบรองรับการบริหารจัดการโครงสร้างหน่วยงานในรูปแบบแผนผังต้นไม้ (Tree Structure) สำหรับ Superadmin โดยมีฟังก์ชันดังนี้

- **การเพิ่มหน่วยงาน** สร้างหน่วยงานระดับบนสุด (Root Institution) ได้เฉพาะเมื่อยังไม่มีหน่วยงานใดในระบบ; และเพิ่มหน่วยงานย่อย (Sub-institution) ภายใต้หน่วยงานหลักได้
- **การแก้ไขหน่วยงาน** แก้ไขชื่อหน่วยงานได้ (รองรับทั้งภาษาไทยและภาษาอังกฤษ)
- **การลบหน่วยงานย่อย** ครุภัณฑ์ทั้งหมดจะถูกโอนกลับ ไปยังหน่วยงานระดับบน (Parent) โดยอัตโนมัติ
- **การลบหน่วยงานระดับบนสุด** จะลบข้อมูลหน่วยงานย่อย ทั้งหมดด้วย (Cascade)
- **การป้องกันการลบ** ระบบจะไม่อนุญาตให้ลบหน่วยงาน หากยังมีครุภัณฑ์ผูกมัดอยู่
- **การแสดงผล** แสดงแผนผังโครงสร้างหน่วยงานในรูปแบบ Tree พร้อมจำนวนครุภัณฑ์สะสมของแต่ละหน่วยงาน (รวมตัวเลขจากหน่วยงานย่อยด้วย) สำหรับทุกบทบาท

3.3.9 ระบบบันทึกประวัติย้อนหลัง

ระบบบันทึก Audit Log สำหรับติดตามความเปลี่ยนแปลงของครุภัณฑ์ ครอบคลุมกิจกรรมดังต่อไปนี้

- การเพิ่มและแก้ไขข้อมูลครุภัณฑ์
- การมอบหมายและเปลี่ยนเจ้าของครุภัณฑ์
- การตรวจนับครุภัณฑ์ประจำปี
- การแจ้งซ่อมครุภัณฑ์
- การโอนย้ายครุภัณฑ์

ทั้งนี้การลบครุภัณฑ์จะไม่ถูกบันทึกใน Audit Log ของ Superadmin สามารถเรียกดูประวัติแบบเจาะจงรายครุภัณฑ์ได้ ผ่านหน้ารายละเอียดครุภัณฑ์ โดยระบบออกแบบให้ดูประวัติ ทีละรายการเท่านั้น ไม่มีหน้ารวมประวัติทั้งหมด

3.3.10 ระบบสร้างและใช้งาน QR Code สำหรับครุภัณฑ์

ระบบสร้าง QR Code โดยอัตโนมัติสำหรับครุภัณฑ์ทุกชิ้นเมื่อมีการ เพิ่มข้อมูลเข้าสู่ระบบ โดยมีฟังก์ชันการใช้งานดังนี้

- **การพิมพ์ QR Code** User สามารถพิมพ์ QR Code ของครุภัณฑ์แต่ละชิ้นได้ผ่านหน้าต่าง Modal บน Frontend
- **การสแกน QR Code** ระบบรองรับการสแกนผ่านกล้อง ในหน้าสแกนเนอร์ของระบบ
- **การแสดงข้อมูลหลังสแกน** ระบบจะนำทางไปยังหน้า รายละเอียดครุภัณฑ์ที่ถูกสแกน
- **ปุ่มแจ้งซ่อม** หน้าข้อมูลหลังสแกนมีปุ่ม “แจ้งซ่อม” เชื่อมไปยังฟอร์มขอส่งซ่อม
- **อินเทอร์เฟซการตรวจนับ** หากมีรอบการตรวจนับ เปิดอยู่ (Active) หน้า QR จะแสดงอินเทอร์เฟซสำหรับการ ตรวจนับให้ทันที

3.4 ขั้นตอนการทำงานของระบบ

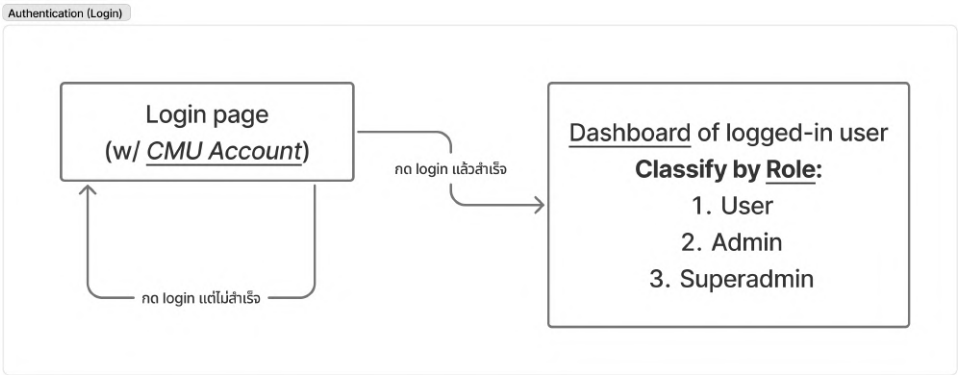
การทำงานของระบบ SCMC AssetManager เริ่มต้นจากผู้ใช้งานเข้าสู่ระบบผ่านเว็บแอปพลิเคชัน โดยผู้ใช้งานจะต้องทำการยืนยันตัวตนก่อนเข้าสู่ระบบ เมื่อระบบตรวจสอบข้อมูลผู้ใช้งานถูกต้องแล้ว ผู้ใช้งานจะสามารถเข้าถึงฟังก์ชันต่าง ๆ ของระบบตามสิทธิ์ที่ได้รับ หลังจากเข้าสู่ระบบ ผู้ใช้งานสามารถดำเนินการต่าง ๆ ภายในระบบ เช่น การดูข้อมูลครุภัณฑ์ การเพิ่มหรือแก้ไขข้อมูลครุภัณฑ์ และการตรวจสอบข้อมูล โดยคำร้องขอจากผู้ใช้งานจะถูกส่งจากส่วน Frontend ไปยัง Backend ผ่าน REST API เพื่อทำการประมวลผล ในส่วนของ Backend จะทำหน้าที่ประมวลผลคำสั่งจากผู้ใช้งาน และติดต่อกับฐานข้อมูลเพื่อดึงข้อมูล บันทึกข้อมูล หรือปรับปรุงข้อมูลตามคำสั่งที่ได้รับ จากนั้นผลลัพธ์จะถูกส่งกลับไปยัง Frontend เพื่อแสดงผลให้ผู้ใช้งานผ่านหน้าเว็บแอปพลิเคชัน ระบบมีการควบคุมการเข้าถึงฟังก์ชันต่าง ๆ โดยใช้แนวคิด Role-Based Access Control (RBAC) ซึ่งผู้ดูแลระบบสามารถกำหนดบทบาทให้กับผู้ใช้งานแต่ละคน เพื่อควบคุมสิทธิ์ในการใช้งานระบบ

3.5 การออกแบบส่วนติดต่อผู้ใช้งาน

3.5.1 การออกแบบ User Flow

Authentication (Login) User Flow

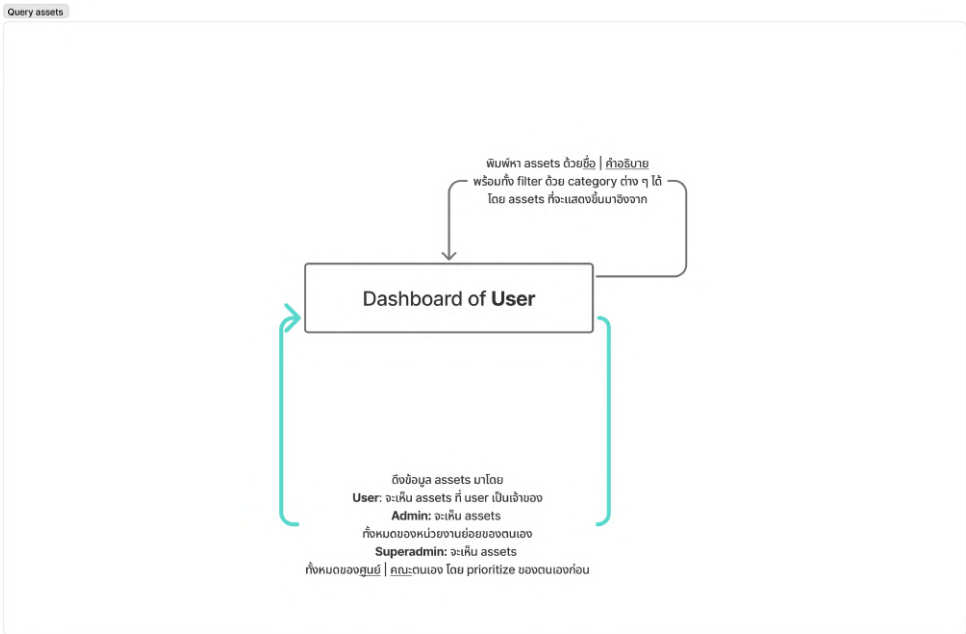
แสดงขั้นตอนการเข้าสู่ระบบของผู้ใช้งาน โดยผู้ใช้จะต้องมีบัญชีผู้ใช้ของมหาวิทยาลัยก่อนจึงจะสามารถเข้าถึงระบบได้ เมื่อการยืนยันตัวตนสำเร็จ ระบบจะกำหนดสิทธิ์การใช้งานตามบทบาทของผู้ใช้ เช่น User, Admin หรือ Superadmin และนำผู้ใช้เข้าสู่หน้าหลักของระบบ



รูปที่ 3.1: Login User Flow

Asset Query User Flow

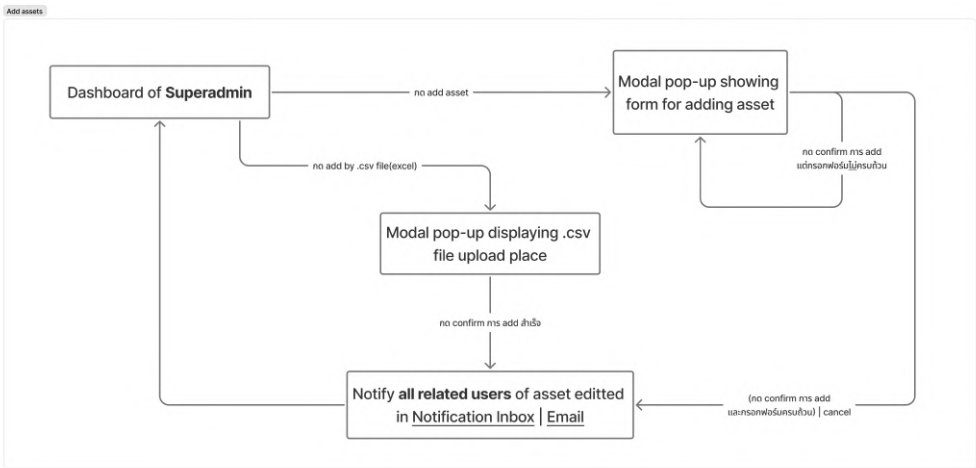
กระบวนการค้นหาและเรียกดูข้อมูลครุภัณฑ์ภายในระบบ ผู้ใช้งานสามารถค้นหาครุภัณฑ์ผ่านช่องค้นหา หรือใช้ตัวกรองตามหมวดหมู่ต่าง ๆ ได้ โดยข้อมูลที่แสดงจะแตกต่างกันตามสิทธิ์ของผู้ใช้งานแต่ละประเภท



รูปที่ 3.2: Asset Query User Flow

Add Asset User Flow

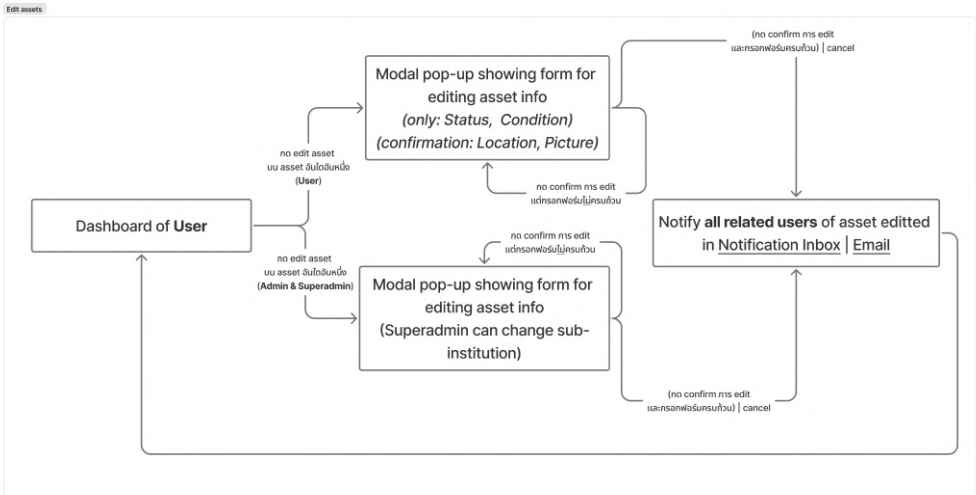
การเพิ่มข้อมูลครุภัณฑ์เข้าสู่ระบบ โดยผู้ดูแลระบบสามารถกรอกข้อมูลครุภัณฑ์ผ่านแบบฟอร์มภายในระบบ หรือทำการนำเข้าข้อมูลผ่านไฟล์ Excel เมื่อข้อมูลถูกบันทึกสำเร็จ ระบบจะทำการจัดเก็บข้อมูลและแจ้งเตือนผู้ใช้งานที่เกี่ยวข้อง



รูปที่ 3.3: Add Asset User Flow

Edit Asset User Flow

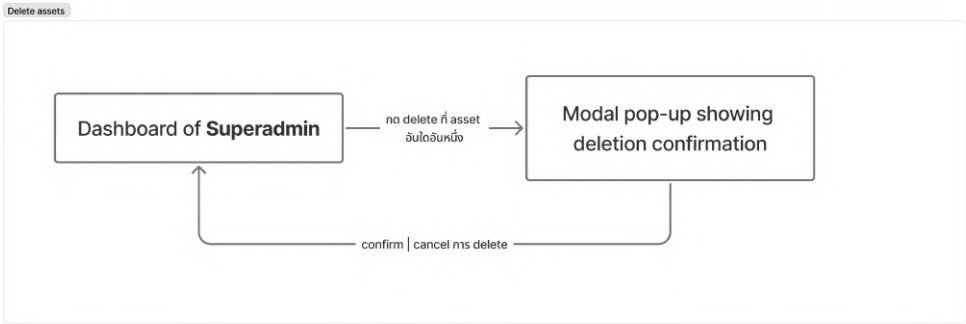
การแก้ไขข้อมูลครุภัณฑ์ในระบบ ผู้ใช้งานที่มีสิทธิ์สามารถเลือกครุภัณฑ์ที่ต้องการแก้ไข และบันทึกการเปลี่ยน-แปลง เมื่อการแก้ไขสำเร็จ ระบบจะอัปเดตข้อมูลและแจ้งเตือนผู้ใช้งานที่เกี่ยวข้อง



รูปที่ 3.4: Edit Asset User Flow

Delete Asset User Flow

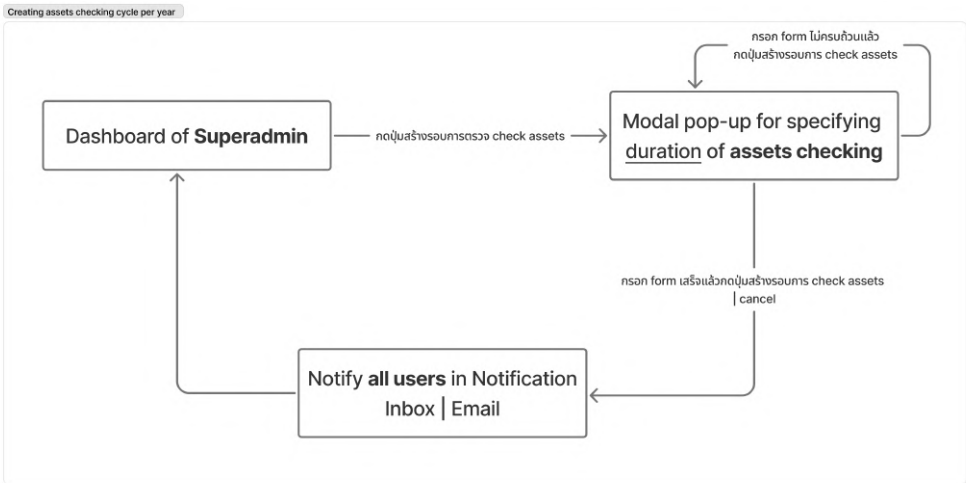
การลบข้อมูลครุภัณฑ์ออกจากระบบ โดยผู้ใช้ที่มีสิทธิ์สามารถเลือกครุภัณฑ์ที่ต้องการลบ จากนั้นระบบจะแสดงหน้าต่างยืนยันการลบเพื่อป้องกันความผิดพลาด หากผู้ใช้ยืนยัน ระบบจะทำการลบข้อมูลออกจากฐานข้อมูล



รูปที่ 3.5: Delete Asset User Flow

Creating Asset Checking Cycle Per Year User Flow

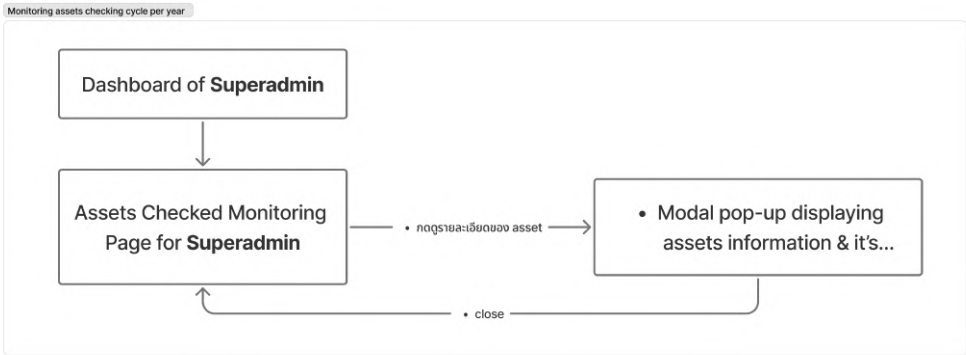
การสร้างรอบการตรวจสอบครุภัณฑ์ประจำปี โดยผู้ดูแลระบบสามารถกำหนดช่วงเวลาของการตรวจสอบครุ-ภัณฑ์ เมื่อสร้างรอบการตรวจสอบสำเร็จ ระบบจะแจ้งเตือนผู้ใช้งานให้ทำการตรวจสอบครุภัณฑ์ของตน



รูปที่ 3.6: Creating Asset Checking Cycle Per Year User Flow

Monitoring Asset Checking Cycle User Flow

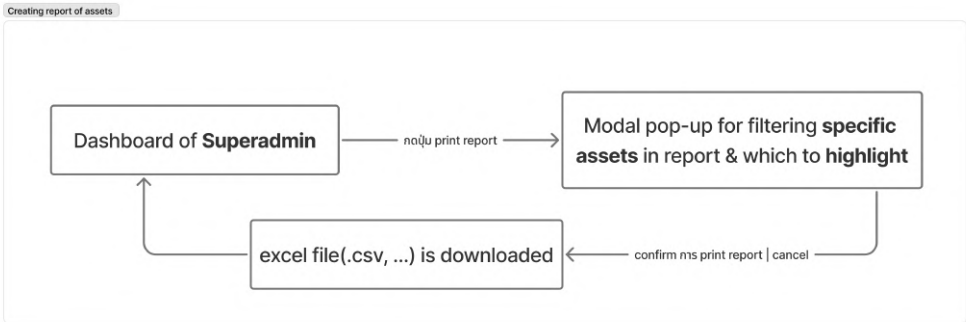
การติดตามสถานะการตรวจสอบครุภัณฑ์ ผู้ดูแลระบบสามารถดูความคืบหน้าของการตรวจสอบครุภัณฑ์ของผู้ใช้งานแต่ละคน รวมถึงสามารถดูรายละเอียดของครุภัณฑ์แต่ละรายการได้



รูปที่ 3.7: Monitoring Asset Checking Cycle User Flow

Creating Asset Report User Flow

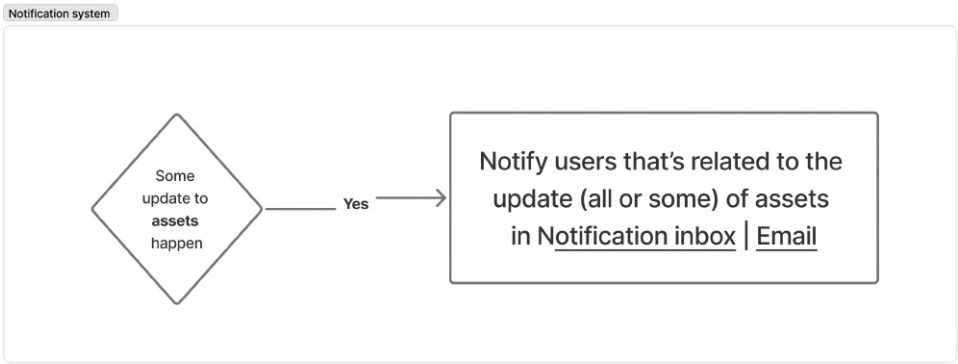
การสร้างรายงานครุภัณฑ์จากข้อมูลในระบบ ผู้ดูแลระบบสามารถเลือกตัวกรองข้อมูลที่ต้องการ จากนั้นระบบจะทำการสร้างรายงานในรูปแบบไฟล์ Excel เพื่อให้สามารถนำไปใช้งานต่อได้



รูปที่ 3.8: Creating Asset Report User Flow

Notification System User Flow

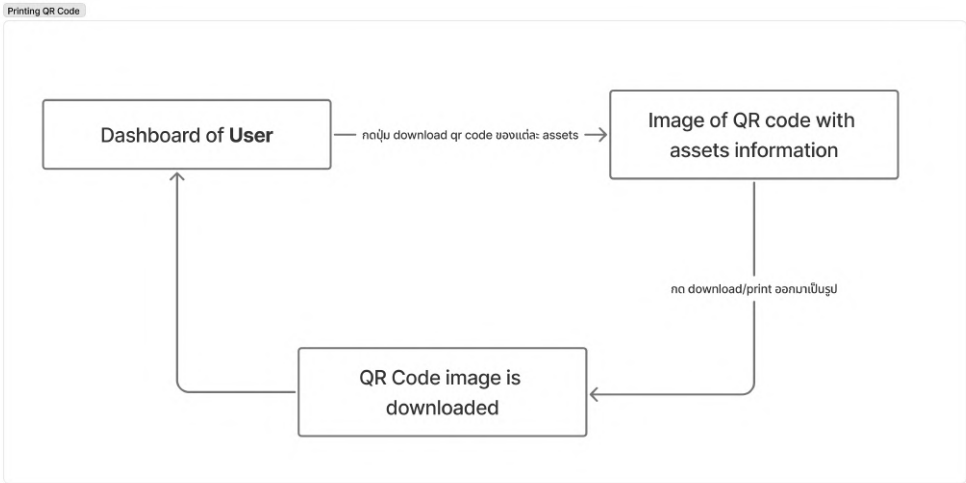
การทำงานของระบบแจ้งเตือน เมื่อมีการเปลี่ยนแปลงข้อมูลครุภัณฑ์หรือมีเหตุการณ์สำคัญเกิดขึ้น ระบบจะส่งการแจ้งเตือนไปยังผู้ใช้งานที่เกี่ยวข้องผ่าน Notification ภายในระบบและทาง Email



รูปที่ 3.9: Notification System User Flow

QR Code Printing User Flow

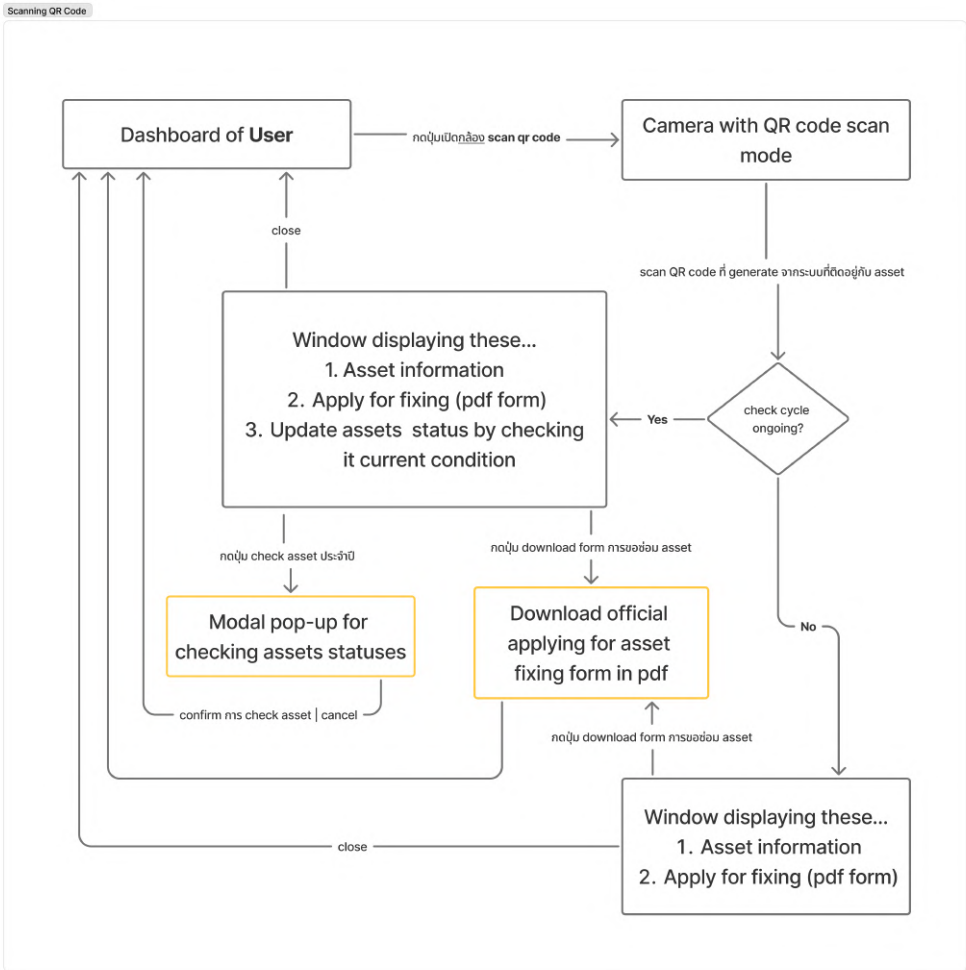
การสร้างและดาวน์โหลด QR Code สำหรับครุภัณฑ์แต่ละรายการ โดยผู้ใช้งานสามารถดาวน์โหลด QR Code จากระบบเพื่อนำไปพิมพ์และติดบนครุภัณฑ์ เพื่อใช้ในการสแกนตรวจสอบข้อมูลได้สะดวก



รูปที่ 3.10: QR Code Printing User Flow

QR Scanning User Flow

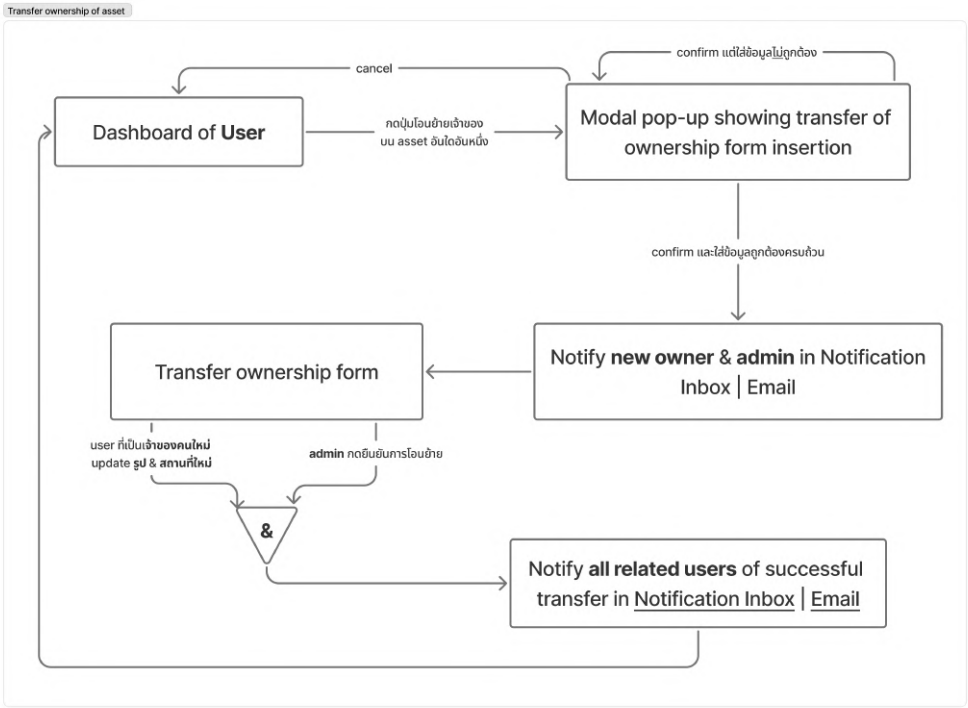
การเข้าถึงข้อมูลครุภัณฑ์ผ่านการสแกน QR Code ที่ติดอยู่บนครุภัณฑ์ ผู้ใช้งานสามารถใช้กล้องของอุปกรณ์มือถือหรือแอปพลิเคชันสแกน QR Code เพื่อเปิดลิงก์ไปยังหน้าข้อมูลครุภัณฑ์ในระบบได้โดยตรง หลังจากเข้าสู่หน้าข้อมูลครุภัณฑ์ ผู้ใช้งานสามารถตรวจสอบรายละเอียดของครุภัณฑ์ รวมถึงดำเนินการต่าง ๆ ตามสิทธิ์ที่ได้รับ เช่น การอัปเดตสถานะหรือการตรวจสอบครุภัณฑ์



รูปที่ 3.11: QR Scanning User Flow

Transfer Ownership of Asset User Flow

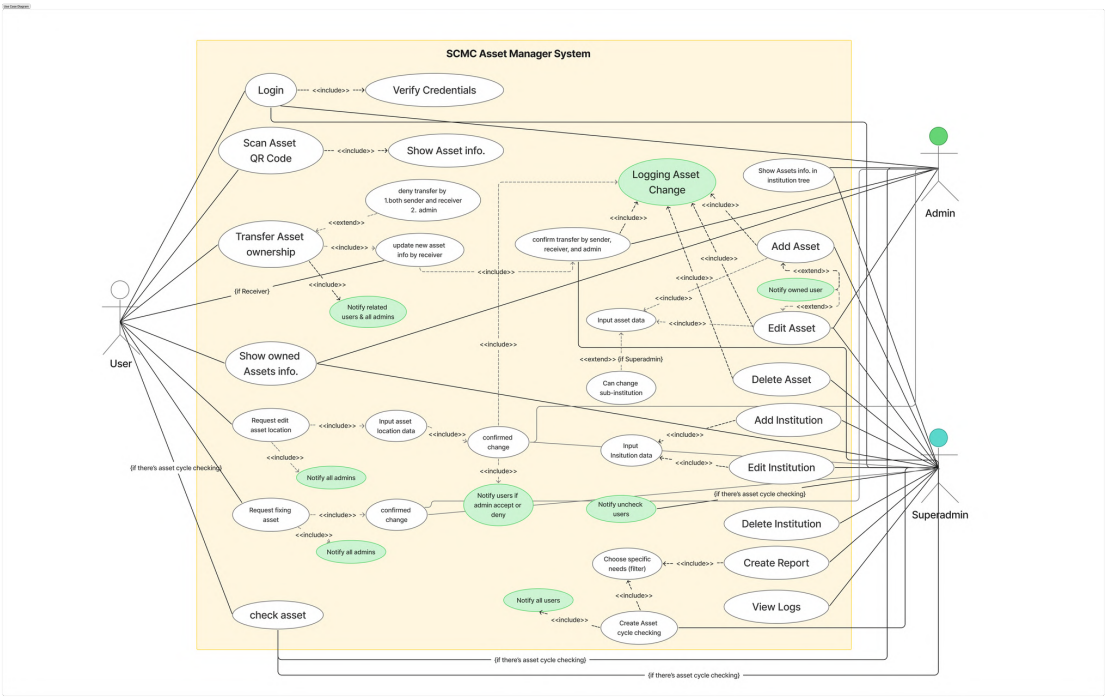
การโอนความเป็นเจ้าของครุภัณฑ์จากผู้ใช้งานคนหนึ่งไปยังผู้ใช้งานอีกคนหนึ่งภายในระบบ ผู้ใช้ที่มีสิทธิ์สามารถเลือกครุภัณฑ์ที่ต้องการโอน จากนั้นกำหนดผู้รับโอนใหม่ผ่านแบบฟอร์มภายในระบบ เมื่อยืนยันการโอนแล้ว ระบบจะทำการอัปเดตข้อมูลเจ้าของครุภัณฑ์ในฐานข้อมูล และส่งการแจ้งเตือนไปยังผู้ใช้งานที่เกี่ยวข้องเพื่อรับทราบการเปลี่ยนแปลงดังกล่าว



รูปที่ 3.12: Transfer Ownership of Asset User Flow

3.5.2 Use Case Diagram

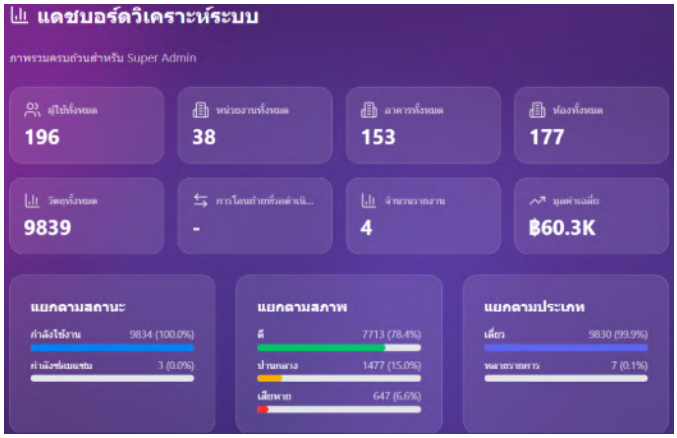
Use Case Diagram นี้จะเชื่อมโยงความสัมพันธ์ระหว่าง Feature กับ Stakeholders



รูปที่ 3.13: Use Case Diagram ของระบบ

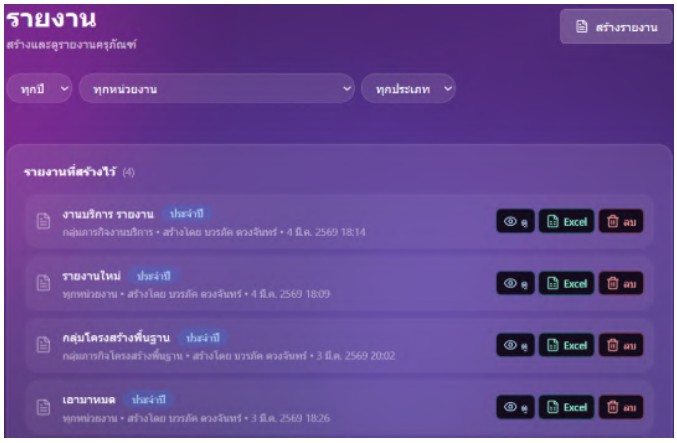
3.5.3 การออกแบบหน้าจอของระบบ

หน้าการแสดงข้อมูลครุภัณฑ์ในรูปแบบ Dashboard



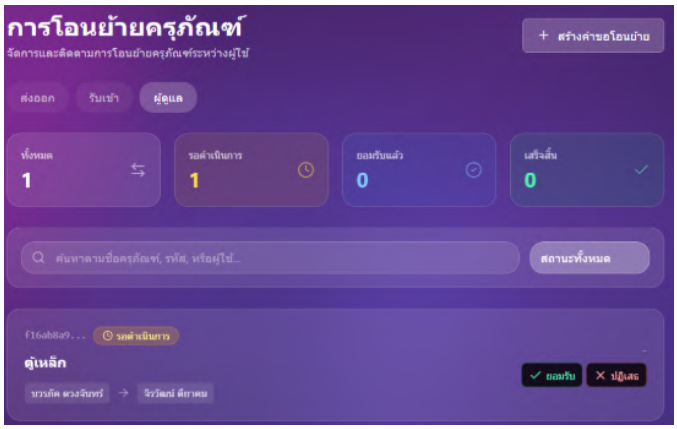
รูปที่ 3.14: ตัวอย่างหน้าการแสดงข้อมูลครุภัณฑ์ในรูปแบบ Dashboard

หน้ารายงานครุภัณฑ์ประจำปี (Annual Report)



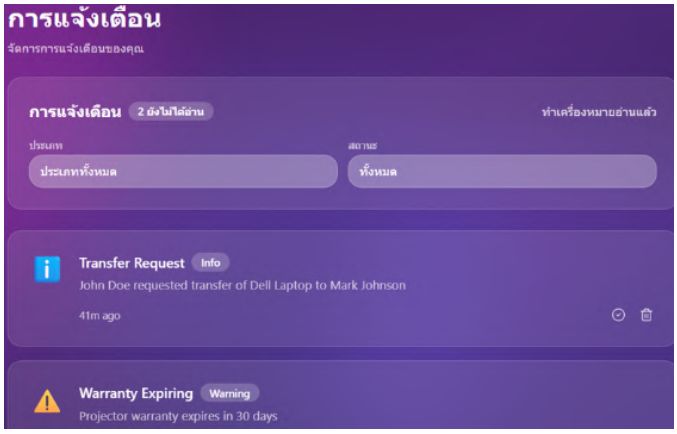
รูปที่ 3.15: ตัวอย่างหน้า Annual Report

หน้าโอนย้ายครุภัณฑ์ (Transfer)



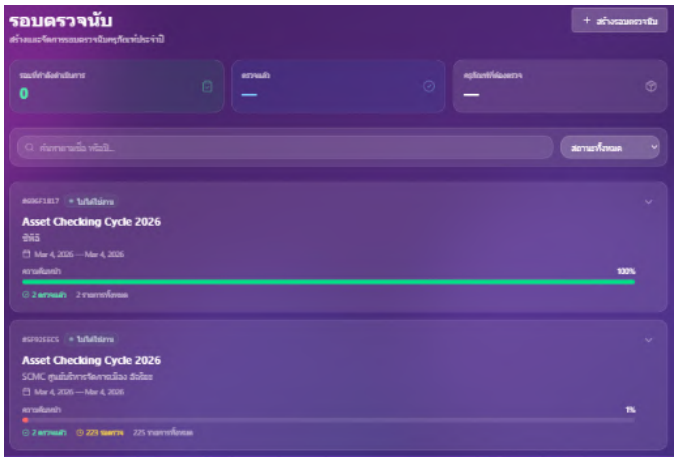
รูปที่ 3.16: ตัวอย่างหน้า Transfer

หน้าแจ้งเตือน (Notification)



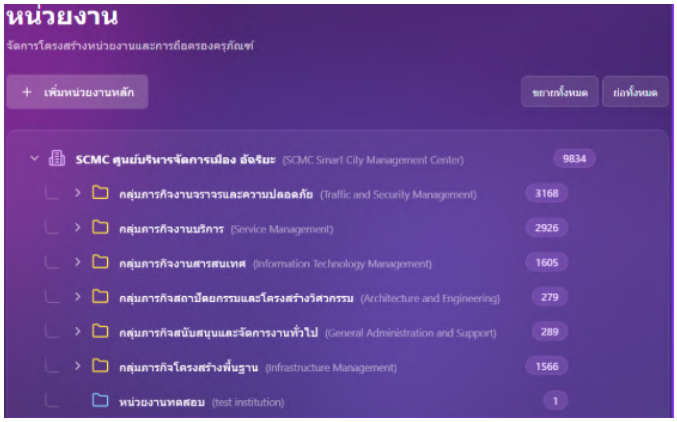
รูปที่ 3.17: ตัวอย่างหน้า Notification

หน้านับรอบการตรวจครุภัณฑ์ Check Cycle



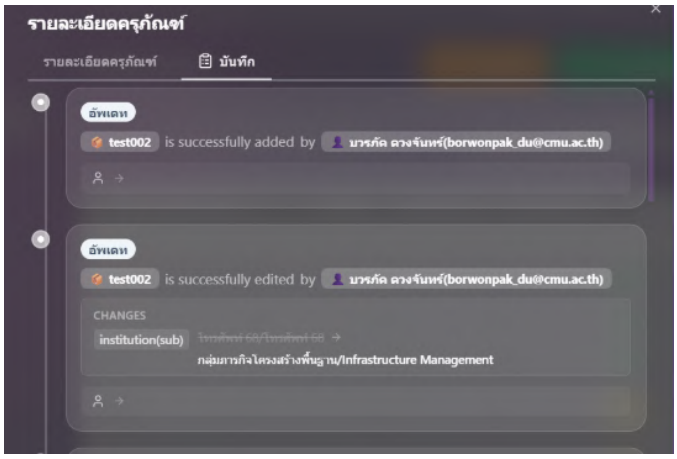
รูปที่ 3.18: ตัวอย่างหน้า Check Cycle

หน้าจัดการข้อมูลหน่วยงาน (Institution Management)



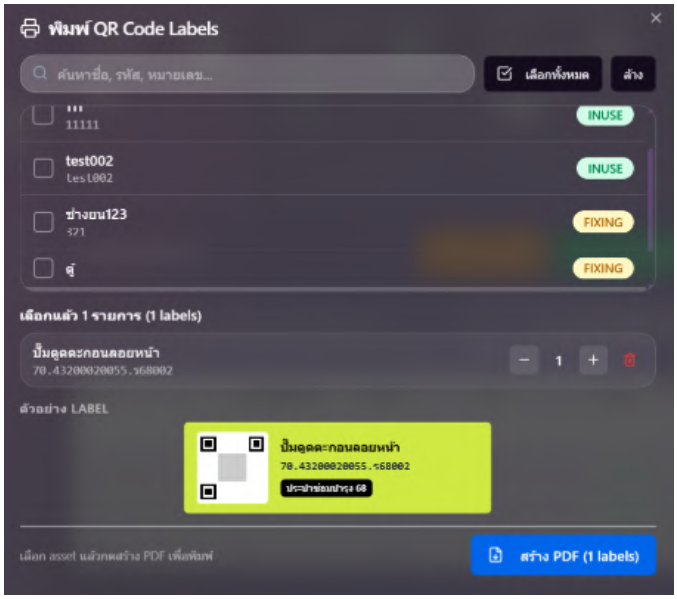
รูปที่ 3.19: ตัวอย่างหน้า Institution Management

หน้าบันทึกประวัติย้อนหลัง Audit Log



รูปที่ 3.20: ตัวอย่างหน้า Audit Log

หน้าการสร้างและใช้งาน QR Code สำหรับครุภัณฑ์ (QR Code And Smart QR)

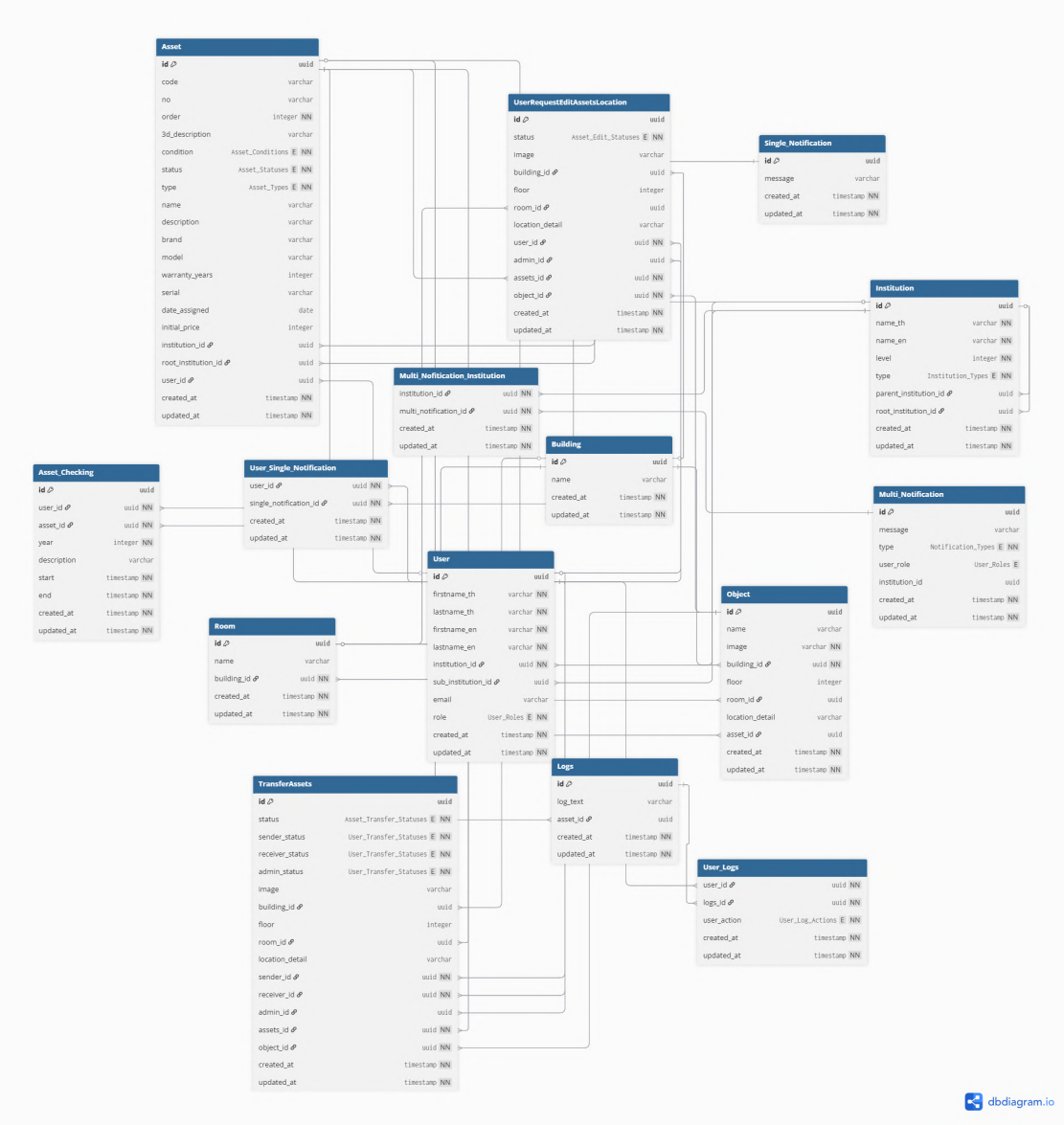


รูปที่ 3.21: ตัวอย่างหน้า QR Code And Smart QR

3.6 การออกแบบฐานข้อมูล

3.6.1 โครงสร้างฐานข้อมูล

แผนภาพฐานข้อมูลของระบบแสดงโครงสร้างของตารางต่าง ๆ และความสัมพันธ์ระหว่างตารางภายในระบบ SCMC AssetManager



รูปที่ 3.22: Database Schema ของระบบ

บทที่ 4
การทดลองและผลลัพธ์

4.1 การทดสอบการทำงานของระบบ (System Testing)

ในการพัฒนาระบบบริหารจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ (SCMC AssetManager) ได้มีการทดสอบความถูกต้องของการทำงานในส่วนของระบบหลังบ้าน (Backend) อย่างเป็นระบบ เพื่อยืนยันความพร้อมของฟังก์ชันการทำงานก่อนเปิดใช้งานจริง โดยแบ่งระดับการทดสอบออกเป็น 2 ส่วนหลัก ได้แก่ การทดสอบแบบกล่องขาว (White-Box Testing) และการทดสอบแบบกล่องดำ (Black-Box Testing)

4.1.1 การทดสอบระดับหน่วย (Unit Testing / White-Box Testing)

การทดสอบระดับหน่วยดำเนินการผ่านเฟรมเวิร์ก Jest ซึ่งเป็นเครื่องมือมาตรฐานสำหรับทดสอบระบบที่พัฒนาด้วย NestJS โดยใช้วิธีการทดสอบแบบกล่องขาว (White-Box Testing) จำนวนทั้งสิ้นประมาณ 801 กรณีทดสอบ (Test cases) ครอบคลุมไฟล์ทดสอบ (Spec files) 31 ไฟล์ การทดสอบนี้ช่วยยืนยันความถูกต้องของตรรกะและฟังก์ชันการทำงานย่อยภายในระบบ สามารถจำแนกกลุ่มการทดสอบได้ดังนี้

- **ระบบจัดการผู้ใช้งานและการยืนยันตัวตน (Auth & Users):** ทดสอบการทำงานของ auth และ users รวม 75 กรณีทดสอบ ครอบคลุมการเข้าสู่ระบบผ่าน CMU OAuth2 SSO, การจัดการเซสชัน และการแก้ไขข้อมูลบทบาท/หน่วยงาน
- **ระบบจัดการครุภัณฑ์และสถานที่ (Assets, Objects, Buildings & Rooms):** ทดสอบฟังก์ชัน assets, objects, buildings และ rooms รวม 225 กรณีทดสอบ ครอบคลุมการสร้าง แก้ไข ลบ และค้นหาข้อมูล ตลอดจนการจัดการสถานที่ตั้งแบบลำดับชั้น
- **ระบบโอนย้ายและตรวจนับครุภัณฑ์ (Transfer Assets & Assets Checking):** ครอบคลุม transfer-assets และ assets_checking รวม 160 กรณีทดสอบ เพื่อยืนยันความถูกต้องของกระบวนการอนุมัติ 3 ฝ่าย และการสร้าง/ตรวจสอบรอบตรวจนับ
- **ระบบรายงาน สถิติ และการนำเข้าข้อมูล (Reports, Analytics & Bulk Import):** รวม 113 กรณีทดสอบ ครอบคลุมการคำนวณสถิติเพื่อแสดงผลบน Dashboard การสร้างรายงาน และระบบนำเข้าข้อมูลแบบกลุ่ม (Unified Bulk Import)
- **ระบบจัดการหน่วยงานและไฟล์ (Institutions & MinIO):** ทดสอบ institutions และ minio รวม 90 กรณีทดสอบ ครอบคลุมการจัดการแผนผังหน่วยงานและการเชื่อมต่อระบบจัดเก็บรูปภาพแบบ Object Storage
- **ระบบแจ้งเตือนอัตโนมัติ (Notifications & Cron):** ทดสอบ notifications และ cron รวม 80 กรณีทดสอบ ตรวจสอบความถูกต้องของการส่ง Event แจ้งเตือน และการทำงานตามรอบเวลาที่กำหนด
- **ระบบบันทึกประวัติ (Asset Logs):** ทดสอบ assets_logs รวม 22 กรณีทดสอบ ยืนยันความสามารถในการบันทึก Audit Trail สำหรับทุกธุรกรรมที่เกิดขึ้นในระบบ

4.1.2 การทดสอบระดับระบบ (End-to-End Testing / Black-Box Testing)

เพื่อจำลองการใช้งานเสมือนจริงในฝั่งผู้ใช้งาน จึงได้ทำการทดสอบระบบแบบ End-to-End (E2E) ด้วยวิธีการทดสอบแบบกล่องดำ (Black-Box Testing) ผ่านเฟรมเวิร์ก Jest ร่วมกับ Supertest จำนวนทั้งสิ้นประมาณ 257 กรณีทดสอบ โดยเชื่อมต่อกับฐานข้อมูล PostgreSQL และพื้นที่จัดเก็บไฟล์ MinIO จริงผ่าน TestContainers ผลการทดสอบครอบคลุมโมดูลต่างๆ ดังนี้

- **การจัดการข้อมูลหลัก (Master Data Management):** ทดสอบ API ส่วน Institutions (29 กรณี), Buildings (12 กรณี) และ Rooms (14 กรณี) ยืนยันว่าระบบสามารถสร้าง แก้ไข และลบข้อมูลได้ตามสิทธิ์ที่กำหนด
- **การจัดการครุภัณฑ์และสถานที่ (Asset & Object Management):** ทดสอบ Assets (60 กรณี) และ Objects (50 กรณี) ตรวจสอบการเข้าถึงข้อมูลครุภัณฑ์ตามลำดับขั้นของหน่วยงาน (Institution Tree) และการจัดการรูปภาพสถานที่
- **กระบวนการทำงานหลัก (Core Workflows):** ทดสอบระบบการโอนย้าย Transfer Assets (32 กรณี) ยืนยันตรรกะการอนุมัติ/ปฏิเสธจากผู้ส่ง ผู้รับ และผู้ดูแลระบบ และทดสอบ Assets Checking (21 กรณี) ในการควบคุมรอบการตรวจนับ
- **ระบบสนับสนุน (Support Systems):** ทดสอบ Users (42 กรณี), Reports (8 กรณี), Analytics (3 กรณี), Bulk Import (3 กรณี), Notifications (9 กรณี) และ Asset Logs (4 กรณี) เพื่อยืนยันประสิทธิภาพของฟังก์ชันเสริมในการสนับสนุนผู้ดูแลระบบ

การทดสอบทั้ง 2 ระดับช่วยยืนยันความเสถียรและความพร้อมของระบบหลังบ้านในการประมวลผล ควบคุมสิทธิ์การเข้าถึง และการทำงานแบบอัตโนมัติ ก่อนที่จะนำระบบไปทดสอบความพึงพอใจการใช้งานจริง (User Evaluation) กับกลุ่มเป้าหมาย

4.2 จุดประสงค์ของการประเมินระบบ

การประเมินระบบบริหารจัดการครุภัณฑ์ดิจิทัลแบบรวมศูนย์ (SCMC AssetManager) มีจุดประสงค์หลักเพื่อตรวจสอบและวัดผลสัมฤทธิ์ของระบบที่พัฒนาขึ้น ว่าสามารถแก้ไขปัญหาการจัดการครุภัณฑ์แบบเดิมที่พึ่งพาไฟล์ Excel ได้อย่างมีประสิทธิภาพหรือไม่ โดยอ้างอิงจากหลักการสำคัญ 4 ประการของโครงงาน ดังนี้

1. **เพื่อประเมินประสิทธิภาพของการรวมศูนย์ข้อมูล (Centralization):** ตรวจสอบความสามารถของระบบในการทำหน้าที่เป็นฐานข้อมูลครุภัณฑ์แหล่งเดียว ว่าสามารถลดความซ้ำซ้อน ลดข้อผิดพลาด (Human Error) และเพิ่มความรวดเร็วในการสืบค้นข้อมูล เมื่อเทียบกับการจัดการผ่านไฟล์ Excel หลายเวอร์ชัน
2. **เพื่อทดสอบความสามารถด้านการทำงานแบบอัตโนมัติ (Automation):** ประเมินความสะดวกและรวดเร็วในการประยุกต์ใช้เทคโนโลยี QR Code เพื่อให้ผู้ใช้งานสามารถเข้าถึงข้อมูล แจ้งซ่อม และอัปเดตสถานะการตรวจนับครุภัณฑ์ประจำปีได้อย่างมีประสิทธิภาพ ลดงานเอกสารและการใช้กระดาษ

3. **เพื่อประเมินระดับความโปร่งใสและตรวจสอบได้ (Transparency & Auditability):** วัดผลความพึงพอใจต่อระบบการยืนยันการโอนย้าย 3 ฝ่าย (ผู้โอน-ผู้รับ-ผู้ดูแลระบบ) และระบบบันทึกประวัติการทำรายการ (Audit Logs) ว่าช่วยให้การตรวจสอบย้อนหลังมีความน่าเชื่อถือเพียงใด
4. **เพื่อศึกษาความพึงพอใจและการมีส่วนร่วมของผู้ใช้งาน (User Engagement):** รวบรวมข้อเสนอแนะและปัญหาจากการใช้งานจริงในทุกระดับสิทธิ์ (User, Admin, Superadmin) สำหรับนำไปพัฒนาต่อยอด ภายใต้ข้อจำกัดของโครงงานที่ต้องใช้อุปกรณ์ที่มีอินเทอร์เน็ต (ไม่มีโหมดออฟไลน์) และมีหน้าต่างแสดงผล (Dashboard) ในระดับพื้นฐาน

4.3 วิธีการประเมินระบบ

การประเมินระบบใช้วิธีการให้ผู้ใช้งานกลุ่มเป้าหมายจากหน่วยงานนำร่องได้ทดลองใช้งานระบบจริงตามสิทธิ์การเข้าถึง (Role-Based Access Control) และเก็บรวบรวมข้อมูลผ่านแบบประเมินผลการใช้งาน โดยมีรายละเอียดกระบวนการประเมินดังนี้

1. **กลุ่มเป้าหมาย (Target Group):** ตัวแทนผู้ใช้งานจากหน่วยงานนำร่องจำนวน 5 ท่าน ซึ่งครอบคลุมบทบาทการใช้งานภายในระบบครบทั้ง 3 ระดับ ประกอบด้วย
   - ผู้ดูแลระบบระดับสูง (Superadmin) จำนวน 2 ท่าน
   - ผู้ดูแลระบบ (Admin) จำนวน 1 ท่าน
   - ผู้ใช้งานทั่วไป (User) จำนวน 2 ท่าน
2. **เครื่องมือที่ใช้ในการประเมิน (Evaluation Tool):** แบบสอบถามประเมินความพึงพอใจและประสิทธิภาพการใช้งานระบบ ซึ่งแบ่งออกเป็น 2 ส่วนหลัก ได้แก่
   - **การประเมินเชิงปริมาณ (Quantitative Assessment):** การให้คะแนนระดับความพึงพอใจและความสะดวกในการใช้งานฟีเจอร์ต่าง ๆ (คะแนนเต็ม 10) เช่น ความรวดเร็วในการสแกน QR Code, การนำเข้าข้อมูลแบบกลุ่ม (Bulk Import), กระบวนการโอนย้ายครุภัณฑ์ และการแสดงผลสรุปบนหน้า Dashboard
   - **การประเมินเชิงคุณภาพ (Qualitative Assessment):** การตอบคำถามปลายเปิดเพื่อรวบรวมข้อคิดเห็นเชิงลึก แบ่งเป็น จุดเด่นที่ชื่นชอบ (LIKE), จุดที่ควรปรับปรุง (WISH) และแนวคิดสำหรับต่อยอด (IDEA)
3. **กระบวนการทดสอบ (Testing Process):** ผู้ทดสอบทำการเข้าสู่ระบบผ่านระบบยืนยันตัวตนของมหาวิทยาลัย (CMU OAuth2 SSO) และทดลองปฏิบัติงานตามขั้นตอนการทำงานจริง (User Flow) เช่น การนำเข้าข้อมูล การสแกน QR Code การดำเนินการโอนย้าย และการตรวจสอบภาพรวมผ่าน Dashboard ก่อนที่จะทำการตอบแบบประเมิน

4.4 ผลการประเมินระบบ

จากการทดสอบใช้งานจริง พบว่า **ผู้ใช้งาน 100% ยินดีแนะนำระบบ** โดยมีความพึงพอใจรวม (Overall Satisfaction) เท่ากับ **7.60 / 10**

4.4.1 ผลประเมินเชิงปริมาณ

1) ฟีเจอร์สำหรับผู้ใช้งานทั่วไป

QR Code Scan	ความสะดวกและรวดเร็วในการเข้าถึงข้อมูลและแจ้งซ่อม	**9.20**
Transfer Verification	ระบบยืนยัน 3 ฝ่าย เพิ่มความโปร่งใสในการโอนย้าย	**9.00**
Annual Update	ความสะดวกในการอัปเดตสถานะครุภัณฑ์	**8.60**
Search & Filter	การค้นหาและกรองข้อมูล	**7.40**
Location Update	การอัปเดตสถานที่และรูปภาพ	**7.40**

2) ฟีเจอร์สำหรับผู้ดูแลระบบ

Bulk Import / Export	นำเข้าข้อมูลและส่งออกรายงาน	**8.75**
Progress Panel	การติดตามสถานะการตรวจนับ	**8.00**
Audit Logs	ระบบบันทึกประวัติย้อนหลัง	**7.25**
Dashboard	ภาพรวมสถิติสำหรับการตัดสินใจ	**6.50**

4.4.2 ผลประเมินเชิงคุณภาพ

1) จุดเด่น (LIKE)

- **QR Code System:** ลดขั้นตอนการทำงานและเพิ่มความรวดเร็ว
- **Bulk Import:** รองรับข้อมูลจำนวนมากได้อย่างมีประสิทธิภาพ
- **Search System:** ค้นหาข้อมูลได้รวดเร็วและเป็นระบบ

2) จุดที่ควรปรับปรุง (WISH)

- **UI/UX:** ปรับให้ใช้งานง่ายขึ้น และเพิ่มความชัดเจนขององค์ประกอบ
- **Performance:** ลดปัญหา Delay ในการโหลดข้อมูล
- **Logs Display:** ปรับปรุงการแสดงผล History และ Audit Logs

3) แนวทางพัฒนาในอนาคต (IDEA)

- เพิ่มรายละเอียดตำแหน่งครุภัณฑ์ และรองรับหลายรูปภาพ
- เพิ่มรายการ “ครุภัณฑ์รอซ่อม” บน Dashboard
- รองรับการแจ้งเตือน (LINE / Email)
- เพิ่มระบบ Disposal สำหรับครุภัณฑ์ที่เสื่อมสภาพ

4.4.3 ผลประเมินด้านความสามารถในการใช้งานระบบ (System Usability Scale: SUS)

นอกจากการประเมินความพึงพอใจและฟีเจอร์การทำงานแล้ว โครงงานได้นำแบบสำรวจความสามารถในการใช้งานระบบ (System Usability Scale - Thai Version) มาใช้ในการประเมินด้านประสบการณ์ผู้ใช้งาน (User Experience) และความง่ายในการใช้งาน (Usability) โดยมีผู้เข้าร่วมประเมินจำนวน 5 ท่าน ผลการประเมินจำแนกตามบทบาทผู้ใช้งานได้ดังนี้

ผู้ใช้งานทั่วไป (User) คะแนนเฉลี่ยจากผู้ประเมิน 2 ท่าน **47.50**

ผู้ดูแลระบบ (Admin) คะแนนจากผู้ประเมิน 1 ท่าน **60.00**

ผู้ดูแลระบบระดับสูง (Superadmin) คะแนนเฉลี่ยจากผู้ประเมิน 2 ท่าน **71.25**

ภาพรวมและการวิเคราะห์ข้อมูล SUS

เมื่อนำคะแนนจากผู้ประเมินทั้ง 5 ท่านมาคำนวณ พบว่าระบบ SCMC AssetManager มีคะแนนเฉลี่ยรวมอยู่ที่ **59.50 คะแนน** จากคะแนนเต็ม 100 คะแนน ซึ่งสามารถแปรผลและวิเคราะห์ข้อมูลได้ดังนี้

- **การแปรผลตามเกณฑ์มาตรฐาน:** คะแนน 59.50 จัดอยู่ในเกณฑ์ **Marginal (ระดับปานกลางค่อนข้างต่ำ)** บ่งชี้ว่าระบบยังมีความท้าทายในการใช้งาน (Usability Issues) ที่ต้องได้รับการปรับปรุง แม้ว่าระบบจะสามารถทำงานและบรรลุวัตถุประสงค์หลักได้ครบถ้วน
- **ความแตกต่างตามบทบาท (Role-Based Discrepancy):** กลุ่มผู้ดูแลระบบ (Admin/Superadmin) ให้คะแนนสูงกว่ากลุ่มผู้ใช้งานทั่วไป (User) อย่างมีนัยสำคัญ สะท้อนให้เห็นว่าผู้ใช้ทั่วไปอาจยังรู้สึกว่าระบบมีขั้นตอนหรือส่วนติดต่อผู้ใช้งานที่ซับซ้อนเกินความจำเป็นสำหรับการปฏิบัติงานพื้นฐาน
- **ความสอดคล้องกับข้อมูลเชิงคุณภาพ:** ผลคะแนน SUS สอดคล้องไปในทิศทางเดียวกันกับข้อเสนอแนะเชิงคุณภาพในหัวข้อ WISH ที่ผู้ใช้งานระบุว่าต้องการให้ปรับปรุง UI/UX ให้มีความเป็นมิตรและเข้าใจง่ายยิ่งขึ้น
- **แนวทางพัฒนาต่อยอด:** ผลการประเมินนี้เป็นตัวชี้วัดสำคัญสำหรับการพัฒนาระยะถัดไป โดยควรเน้นที่การออกแบบส่วนติดต่อผู้ใช้งานใหม่ (UI/UX Redesign) เพื่อลดความซับซ้อน (Cognitive Load) และอาจพิจารณาเพิ่มระบบแนะนำการใช้งานเบื้องต้น (Onboarding/Tutorials) เพื่อยกระดับประสบการณ์ของผู้ใช้งานทั่วไป

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผล

โครงงานนี้ได้พัฒนาระบบ SCMC AssetManager เพื่อสนับสนุนการจัดการและการตรวจนับครุภัณฑ์ของหน่วยงานภายในศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน โดยระบบถูกออกแบบให้ผู้ใช้งานสามารถตรวจสอบข้อมูลครุภัณฑ์ ยืนยันสถานะของครุภัณฑ์ และติดตามความคืบหน้าของการตรวจนับผ่านระบบเว็บแอปพลิเคชันได้อย่างสะดวก ระบบประกอบด้วยฟีเจอร์หลัก เช่น การแสดงข้อมูลครุภัณฑ์ในรูปแบบ Dashboard การจัดการข้อมูลครุภัณฑ์ การสร้างรายงานประจำปี การโอนย้ายครุภัณฑ์ ระบบแจ้งเตือน การนับรอบการตรวจครุภัณฑ์ รวมถึงการใช้งาน QR Code สำหรับการเข้าถึงข้อมูลครุภัณฑ์อย่างรวดเร็ว นอกจากนี้ยังมีการกำหนดสิทธิ์ผู้ใช้งานตามบทบาท เพื่อให้การเข้าถึงข้อมูลและการดำเนินการต่าง ๆ เป็นไปอย่างเหมาะสม จากการพัฒนาและทดสอบระบบ พบว่าระบบสามารถช่วยลดขั้นตอนการจัดการข้อมูลครุภัณฑ์ที่เดิมใช้ไฟล์ Excel หลายเวอร์ชัน ทำให้ข้อมูลมีความเป็นปัจจุบันมากขึ้น ลดความผิดพลาดจากการทำงานแบบการทำด้วยตนเอง และเพิ่มความสะดวกในการตรวจสอบข้อมูลย้อนหลัง อย่างไรก็ตาม ระบบที่พัฒนาขึ้นยังมีข้อจำกัดบางประการ เช่น การพึ่งพาข้อมูลจากระบบเดิมและไฟล์ Excel รวมถึงการใช้งานที่ยังอยู่ในขอบเขตของหน่วยงานนำร่อง ซึ่งสามารถนำไปพัฒนาและขยายการใช้งานในอนาคตได้ต่อไป

5.2 ปัญหาที่พบและแนวทางการแก้ไข

ในการพัฒนาโครงงานนี้ พบปัญหาหลักที่เกิดขึ้นในหลายด้าน เช่น ความไม่สอดคล้องของข้อมูลครุภัณฑ์จากแหล่งข้อมูลเดิม การจัดรูปแบบข้อมูลจากไฟล์ Excel ให้สามารถนำมาใช้ในระบบฐานข้อมูลได้อย่างถูกต้อง รวมถึงความท้าทายในการออกแบบระบบให้สามารถรองรับบทบาทของผู้ใช้งานที่แตกต่างกัน นอกจากนี้ยังพบข้อจำกัดด้านเวลาในการพัฒนาและการทดสอบระบบ แนวทางในการแก้ไขปัญหาดังกล่าว คือการปรับปรุงโครงสร้างข้อมูลให้มีความเป็นระบบมากขึ้น ตรวจสอบและทำความสะอาดข้อมูลก่อนนำเข้าสู่ฐานข้อมูล รวมถึงแบ่งหน้าที่การพัฒนาระบบภายในทีมเพื่อให้สามารถดำเนินงานได้อย่างมีประสิทธิภาพ อีกทั้งยังมีการทดสอบระบบอย่างต่อเนื่องเพื่อค้นหาและแก้ไขข้อผิดพลาดที่เกิดขึ้นในระหว่างการพัฒนา

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

ข้อเสนอแนะสำหรับการพัฒนาระบบในอนาคต คือการขยายขอบเขตการใช้งานของระบบให้ครอบคลุมหน่วยงานต่าง ๆ ภายในมหาวิทยาลัยมากยิ่งขึ้น โดยเริ่มต้นจากการนำระบบไปทดลองใช้งานในศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน มหาวิทยาลัยเชียงใหม่ ในฐานะหน่วยงานนำร่อง เพื่อประเมินประสิทธิภาพของระบบและปรับปรุงให้เหมาะสมกับการใช้งานจริง ก่อนขยายการนำระบบไปใช้งานในหน่วยงานอื่น ๆ ภายในมหาวิทยาลัยในระดับที่ใหญ่ขึ้นต่อไป

บรรณานุกรม

[1] Microsoft, "Typescript: A typed superset of javascript." https://www.typescriptlang.org/, 2012. Accessed: 2026-03-24.

[2] Meta Platforms, Inc., "React: A javascript library for building user interfaces." https://react.dev/, 2013. Accessed: 2026-03-24.

[3] Tailwind Labs, "Tailwind css: A utility-first css framework." https://tailwindcss.com/, 2017. Accessed: 2026-03-24.

[4] OpenJS Foundation, "Node.js: A javascript runtime built on chrome v8." https://nodejs.org/, 2009. Accessed: 2026-03-24.

[5] Fastify Team, "Fastify: A fast and low overhead web framework for node.js." https://fastify.dev/, 2016. Accessed: 2026-03-24.

[6] Drizzle Team, "Drizzle orm: A lightweight typescript orm for sql databases." https://orm.drizzle.team/, 2023. Accessed: 2026-03-24.

[7] npm, Inc., "Exceljs: A library for reading and writing excel files in node.js." https://www.npmjs.com/package/exceljs, 2017. Accessed: 2026-03-24.

[8] NestJS, "Nestjs: A progressive node.js framework." https://nestjs.com/, 2017. Accessed: 2026-03-24.

[9] PostgreSQL Global Development Group, "Postgresql: An advanced open-source relational database." https://www.postgresql.org/, 1996. Accessed: 2026-03-24.

[10] MinIO, Inc., "Minio: A high-performance object storage system." https://www.min.io/, 2014. Accessed: 2026-03-24.

[11] Postman, Inc., "Postman: An api platform for building and testing apis." https://www.postman.com/, 2012. Accessed: 2026-03-24.

[12] Meta Platforms, Inc., "Jest: A javascript testing framework." https://jestjs.io/, 2014. Accessed: 2026-03-24.

[13] Cypress.io, "Cypress: A javascript end-to-end testing framework." https://www.cypress.io/, 2017. Accessed: 2026-03-24.

[14] Docker, Inc., "Docker: A platform for containerized applications." https://www.docker.com/, 2013. Accessed: 2026-03-24.

[15] NGINX, Inc., "Nginx: A high-performance http server and reverse proxy." https://nginx.org/, 2004. Accessed: 2026-03-24.

[16] GeeksforGeeks, "Client server model." https://www.geeksforgeeks.org/system-design/client-server-model//. Accessed: 2026-03-24.

[17] Oracle Corporation, "Relational database: A structured data model based on relations." https://www.oracle.com/database/what-is-a-relational-database/. Accessed: 2026-03-24.

[18] National Institute of Standards and Technology (NIST), "Role-based access control (rbac): Access control based on roles." https://csrc.nist.gov/projects/role-based-access-control. Accessed: 2026-03-24.

[19] R. T. Fielding, "Architectural styles and the design of network-based software architectures." Doctoral Dissertation, University of California, Irvine, 2000. Chapter 5: Representational State Transfer (REST).

[20] IBM Corporation, "Audit trail: A record of system activities." https://www.ibm.com/topics/audit-trail. Accessed: 2026-03-24.

[21] DENSO WAVE Incorporated, "Qr code: A two-dimensional barcode system." https://www.qrcode.com/en/about/. Accessed: 2026-03-24.

[22] S. W., "Cmu authorization consumer." https://gitlab.mis.cmu.ac.th/supawit.w/cmu-authorization-consumer. Accessed: 2026-03-27.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 3rd ed., 2009.

[24] I. Sommerville, *Software Engineering*. Pearson, 10th ed., 2015.

[25] J. Brooke, "Sus: A "quick and dirty" usability scale," in *Usability Evaluation in Industry* (P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, eds.), pp. 189–194, Taylor & Francis, 1996.

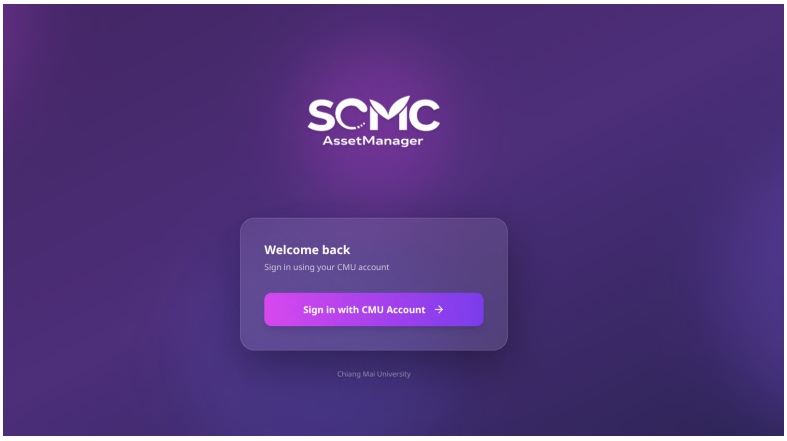
ภาคผนวก

ภาคผนวก ก
คู่มือการใช้งานระบบ

ก.1 การเข้าสู่ระบบและการตั้งค่าเริ่มต้น

ก.1.1 การเข้าสู่ระบบ

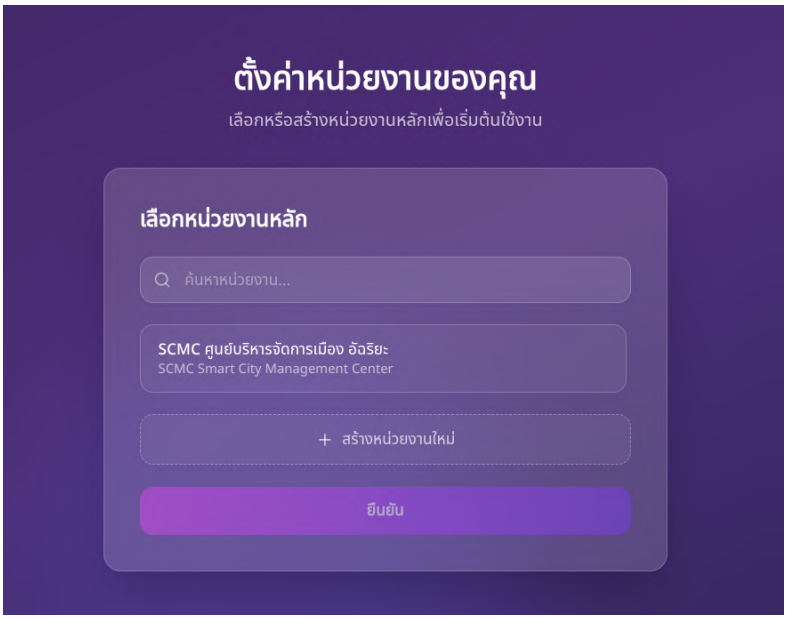
ทุกหน้าของระบบบังคับให้ต้องเข้าสู่ระบบผ่าน CMU SSO OAuth2 ระบบจะสร้างบัญชีผู้ใช้งานใหม่โดยอัตโนมัติเมื่อล็อกอินครั้งแรกด้วยข้อมูลจาก CMU SSO



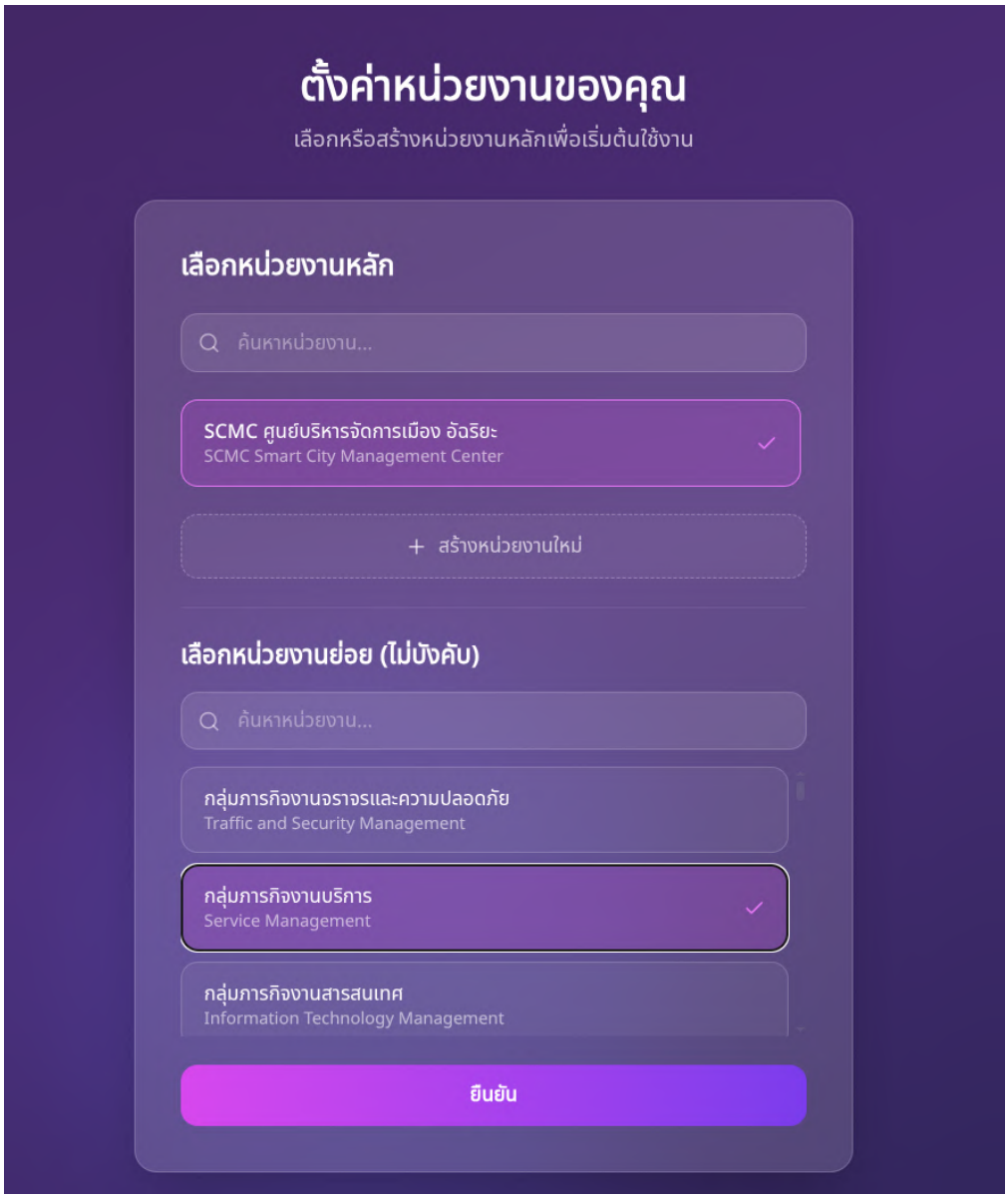
รูปที่ ก.1: หน้าต่าง Login

ก.1.2 การเลือกหน่วยงาน

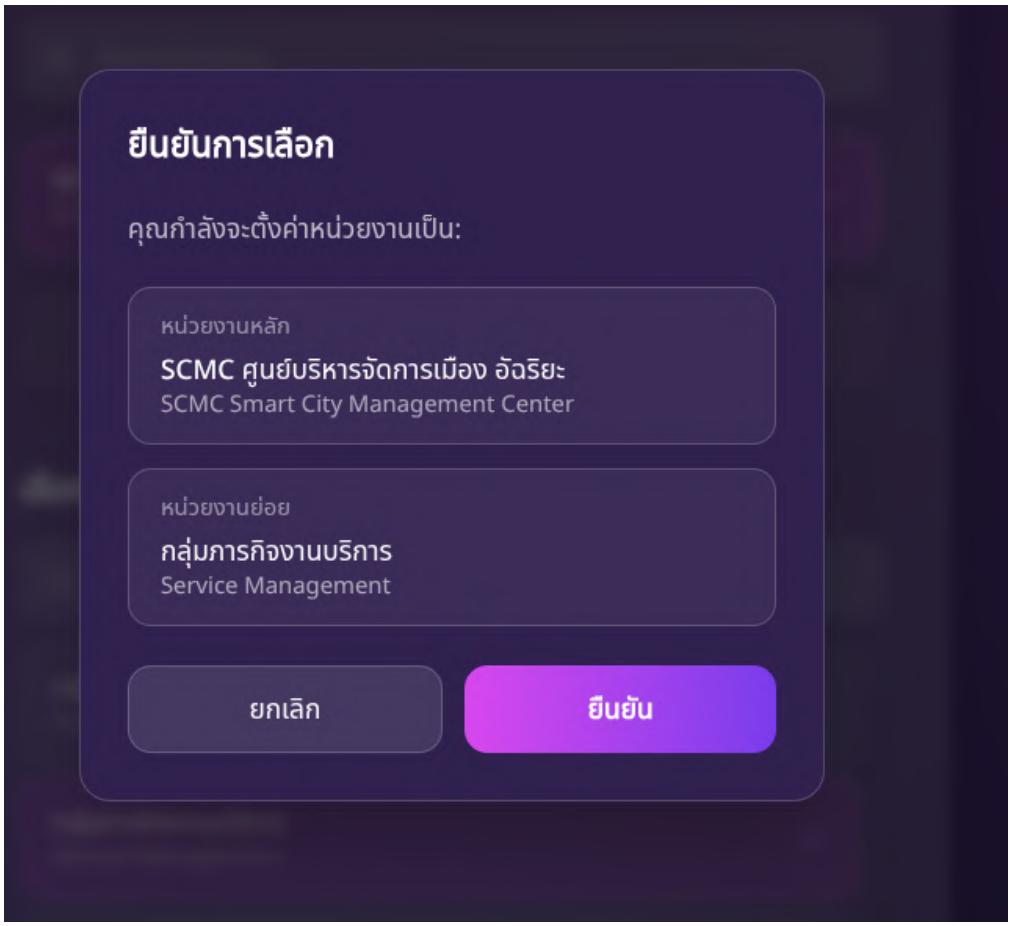
ผู้ใช้งานที่ยังไม่ได้เลือกหน่วยงานหลักจะถูกเปลี่ยนหน้าไปยังหน้าจอเลือกหน่วยงานหลังจากเข้าสู่ระบบสำเร็จ ผู้ใช้สามารถค้นหาและเลือกหน่วยงานหลักได้เพียงครั้งเดียวซึ่งจะเป็นการตั้งค่าถาวร หลังจากนั้นผู้ใช้สามารถเลือกหน่วยงานย่อยได้ตามความเหมาะสม หากไม่มีข้อมูลหน่วยงานในระบบเลย ระบบจะแสดงฟอร์มสำหรับสร้างหน่วยงานใหม่ การสร้างหน่วยงานหลักระดับบนสุดคนแรกจะได้รับสิทธิ์ Superadmin ทันที



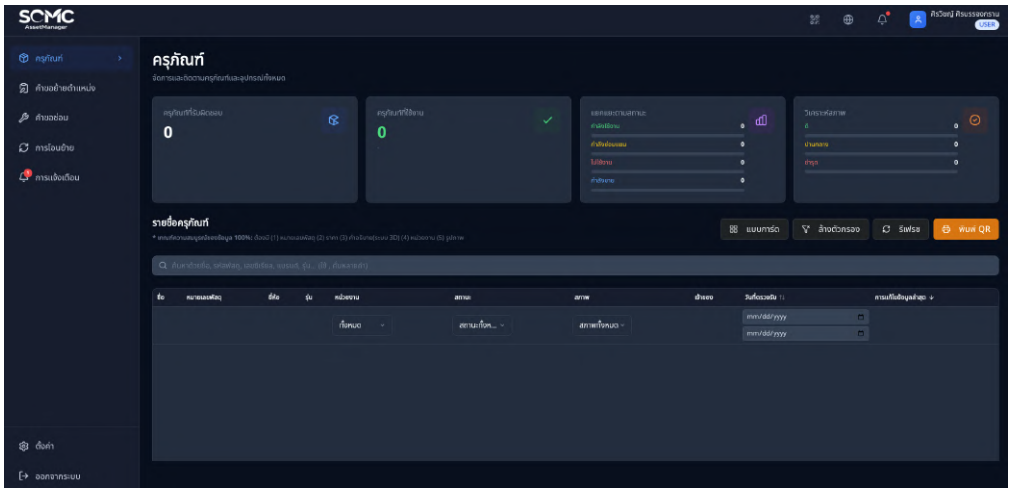
รูปที่ ก.2: เลือกหรือสร้างหน่วยงาน



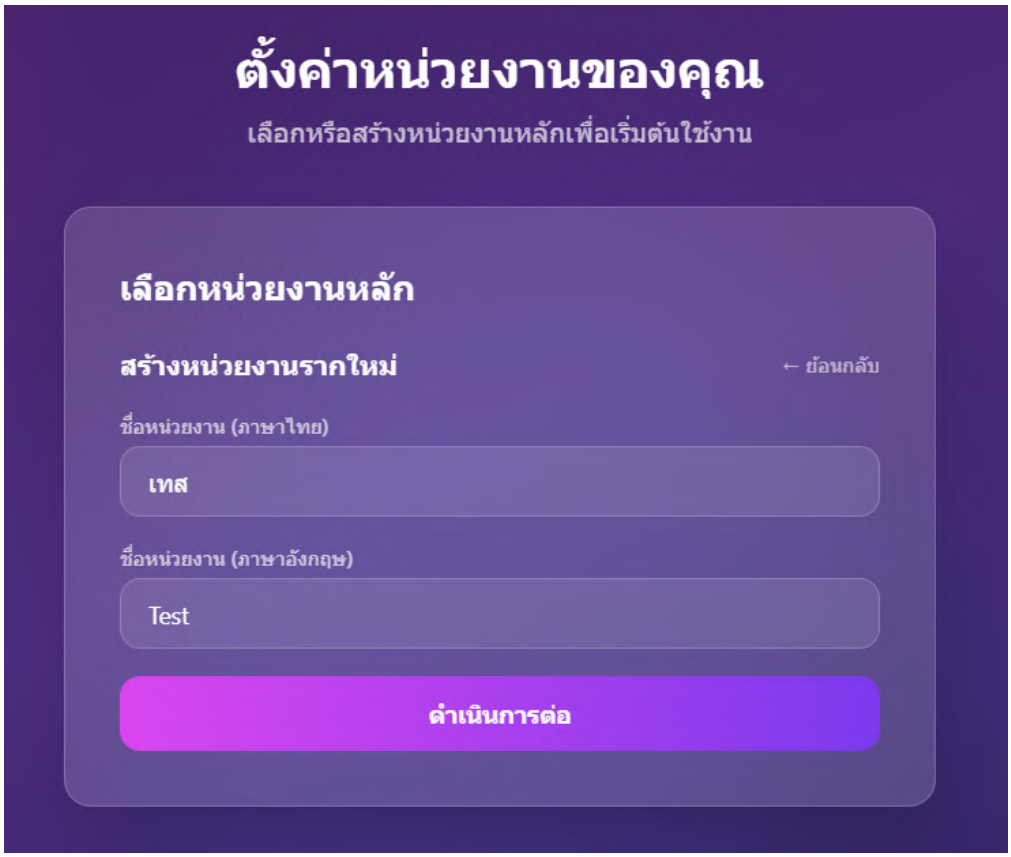
รูปที่ ก.3: เลือกหน่วยงานหลักและหน่วยงานย่อย (ไม่บังคับสำหรับหน่วยงานย่อย)



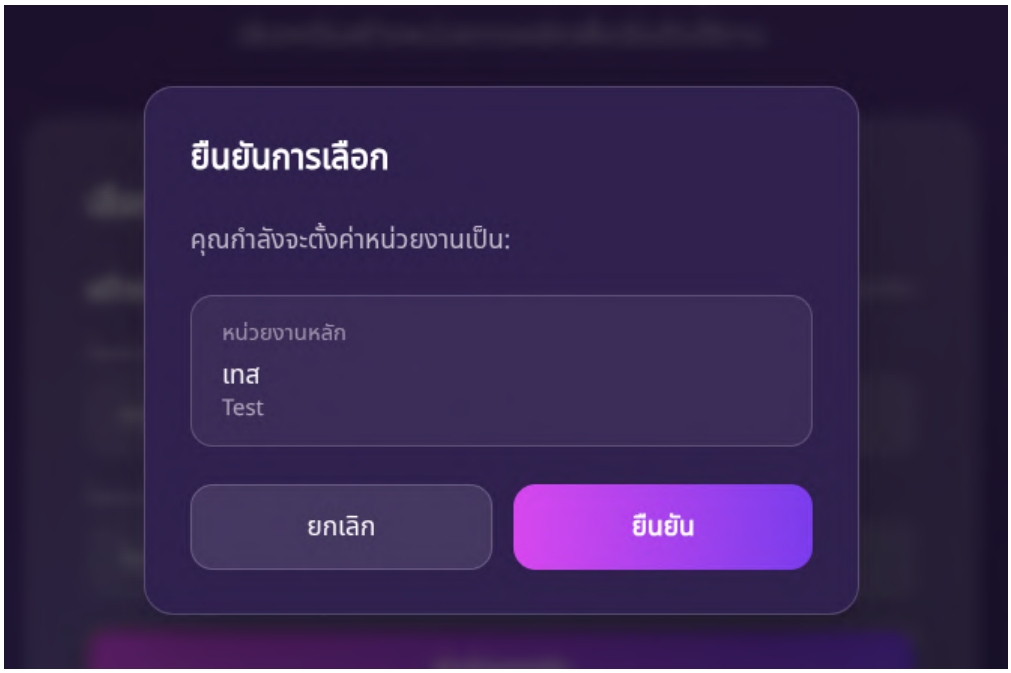
รูปที่ ก.4: ยืนยันการเลือกหน่วยงานหลักและหน่วยงานย่อย



รูปที่ ก.5: หน้าต่างหลังจากเข้าหน่วยงานสำเร็จ



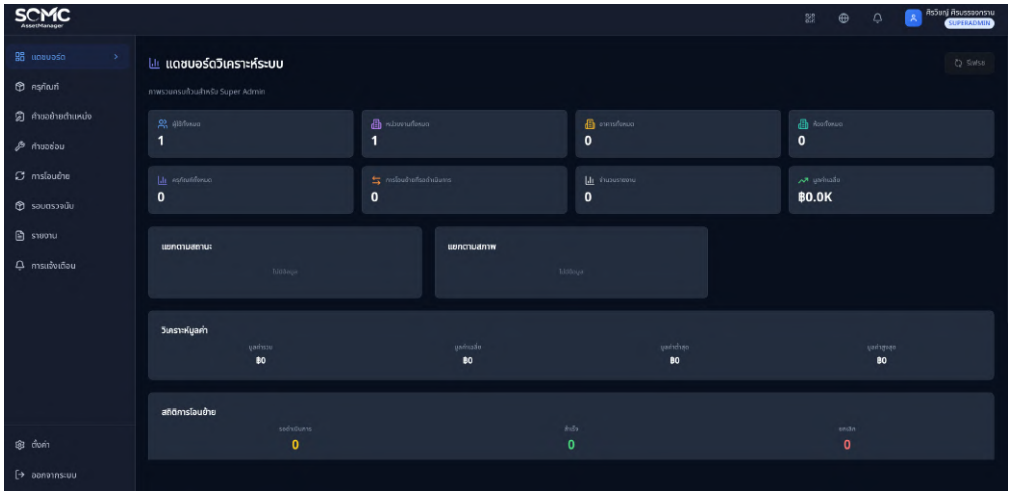
รูปที่ ก.6: สร้างหน่วยงานหลักใหม่



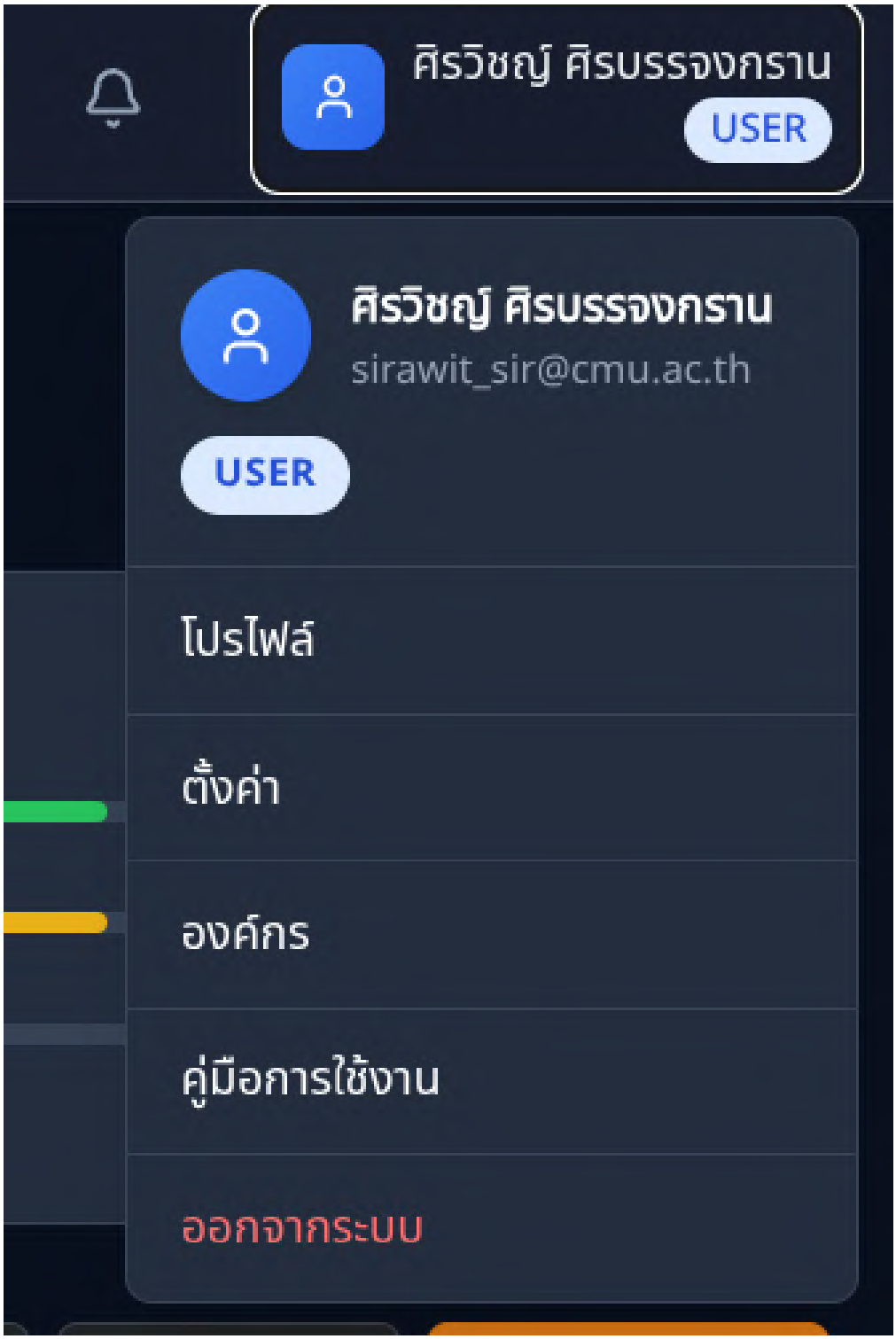
รูปที่ ก.7: ยืนยันการสร้างหน่วยงานหลักใหม่

ก.1.3 การตั้งค่าต่าง ๆ ในระบบ

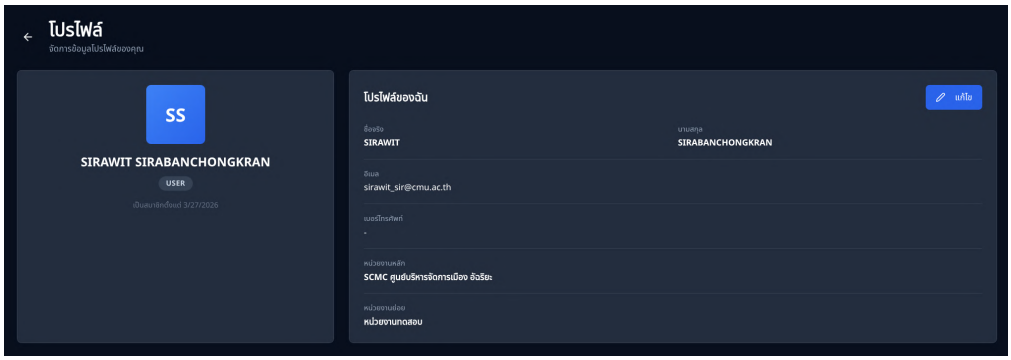
ผู้ใช้งานสามารถเปลี่ยนธีม (Theme) เป็นโหมดสว่าง (Light), โหมดมืด (Dark) หรือธีม CMU (ม่วงเข้ม). ระบบมีตัวเลือกภาษา (Language) รองรับทั้งภาษาไทยและภาษาอังกฤษ. ค่าที่ตั้งไว้จะถูกบันทึกเพื่อจำค่าในครั้งถัดไป. ผู้ใช้งานสามารถอัปเดตหมายเลขโทรศัพท์ของตนเองได้ในหน้าโปรไฟล์. ใช้งานสามารถกดออกจากหน่วยงานเองได้โดยพิมพ์คำว่า LEAVE แล้วกดยืนยัน. ครุภัณฑ์ที่ผู้ใช้นั้นดูแลอยู่จะถูกโอนกรรมสิทธิ์ให้กับ Superadmin โดยอัตโนมัติ.



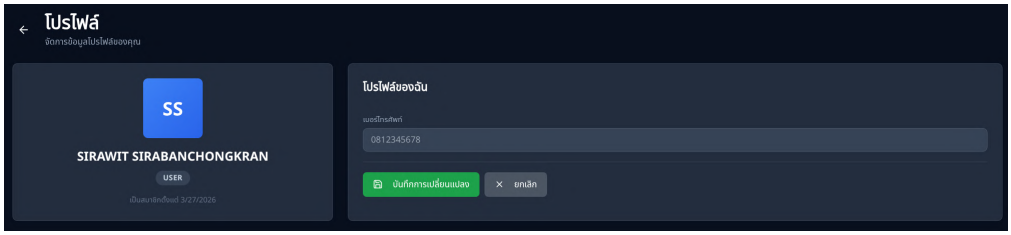
รูปที่ ก.8: หน้าต่างหลังจากสร้างหน่วยงานหลักใหม่ ผู้สร้างเป็น Superadmin ของหน่วยงานที่สร้างใหม่



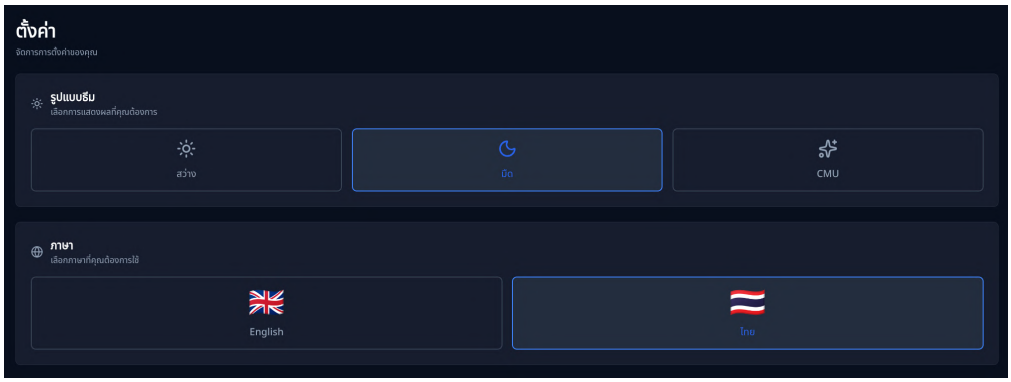
รูปที่ ก.9: ทางลัดไปหน้าต่าง ๆ ที่เกี่ยวกับผู้ใช้



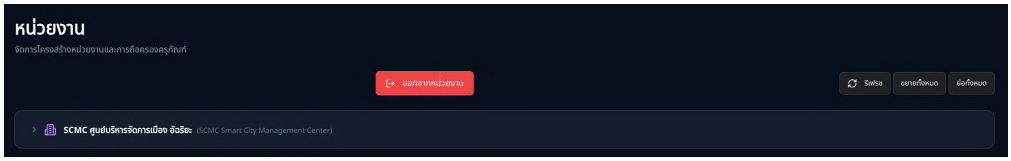
รูปที่ ก.10: หน้าโปรไฟล์ของผู้ใช้



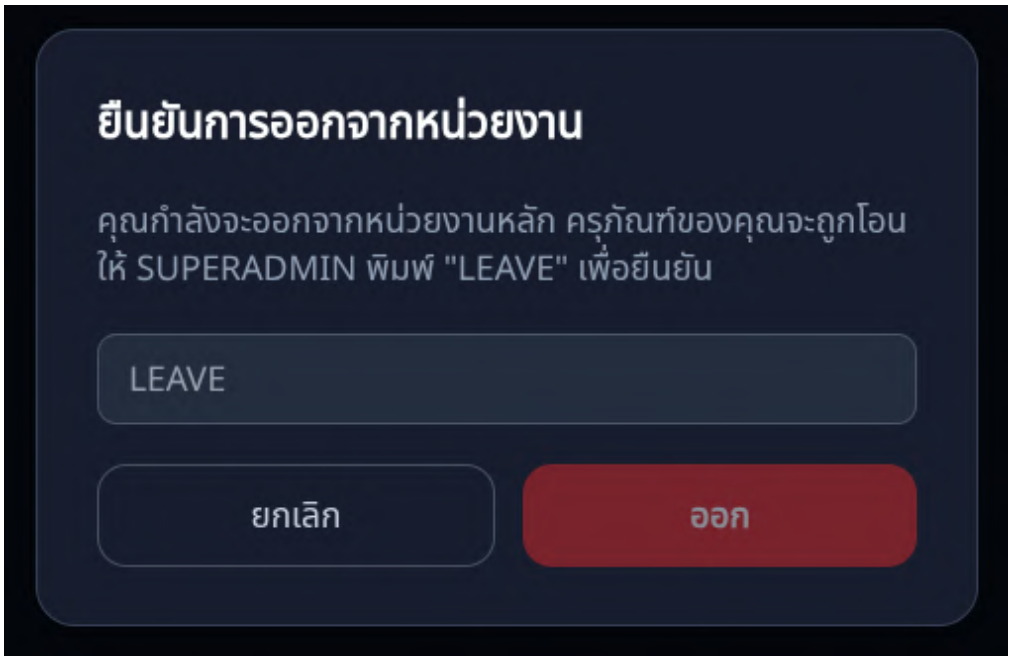
รูปที่ ก.11: หน้าการแก้ไขเบอร์โทรศัพท์ในโปรไฟล์ของผู้ใช้



รูปที่ ก.12: หน้าการตั้งค่า



รูปที่ ก.13: หน้าหน่วยงาน/องค์กร

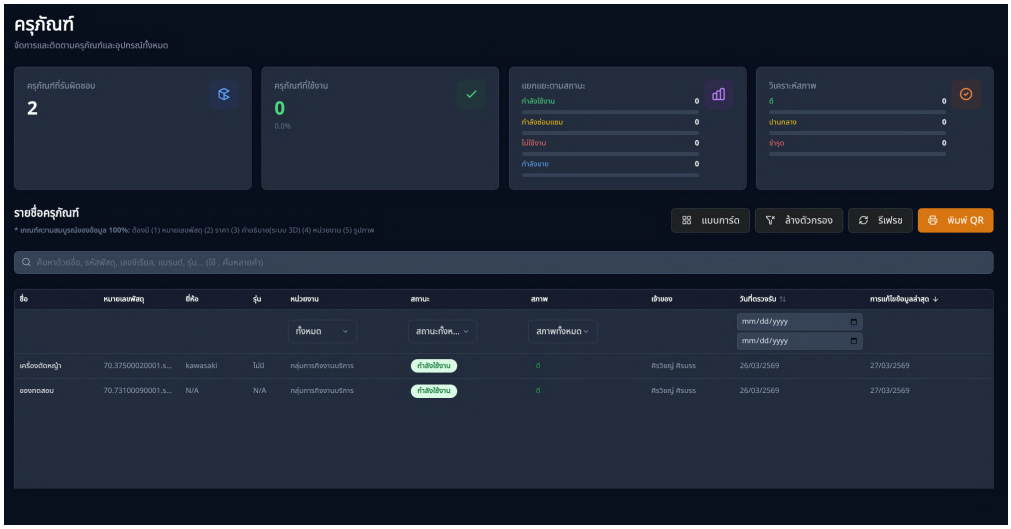


รูปที่ ก.14: หน้าต่างการออกจากหน่วยงาน ผู้ใช้ต้องพิมพ์คำว่า LEAVE จึงจะสามารถออกจากหน่วยงานได้

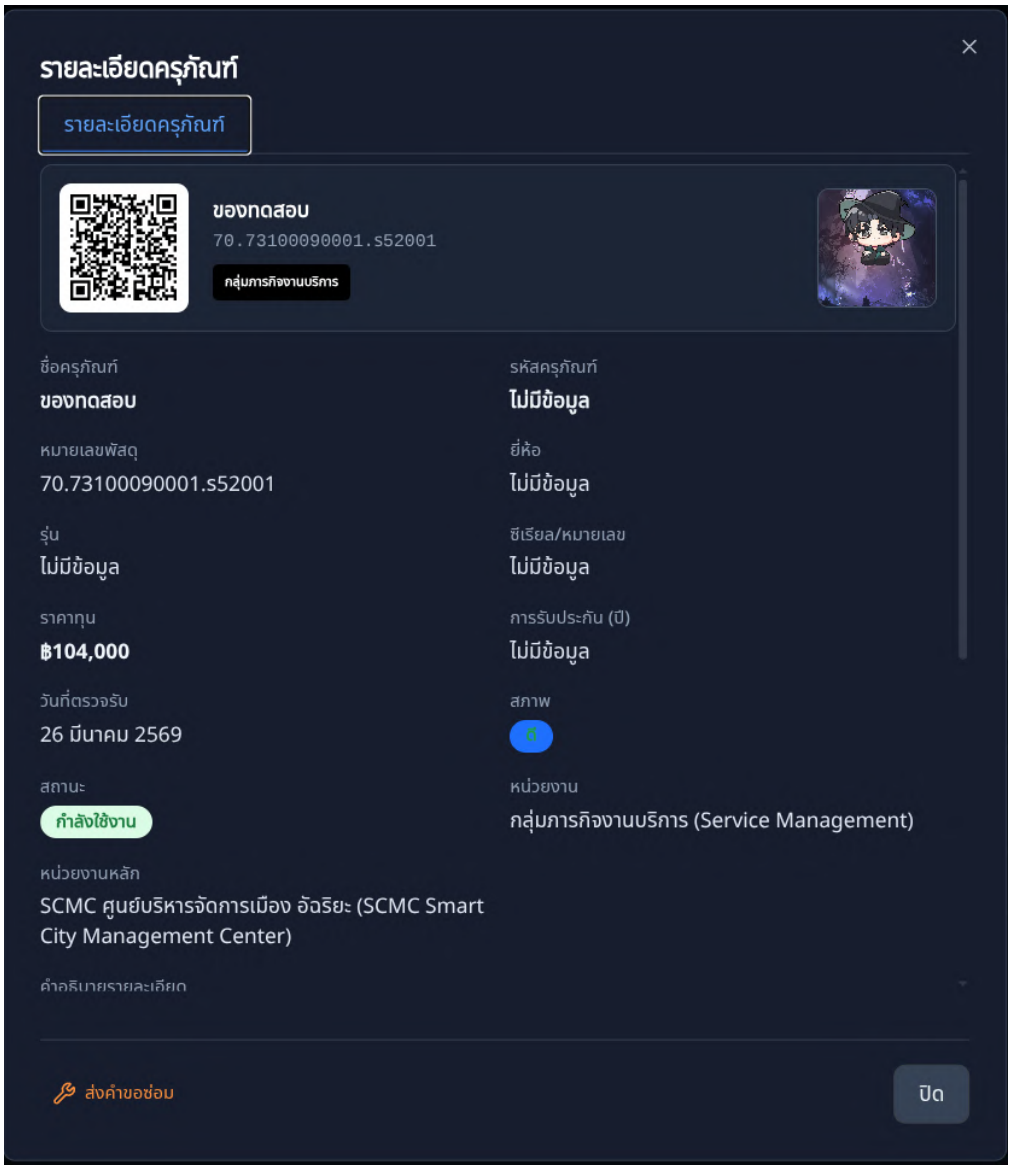
ก.2 การจัดการครุภัณฑ์ (Assets)

ก.2.1 การดูข้อมูลครุภัณฑ์ (Assets)

เมื่อเข้าสู่ระบบแล้ว ระบบจะพาผู้ใช้ไปยังหน้าข้อมูลครุภัณฑ์. ผู้ใช้งานทั่วไปจะมองเห็นและดูรายละเอียดครุภัณฑ์เฉพาะรายการที่ตนเองเป็นผู้รับผิดชอบเท่านั้น.



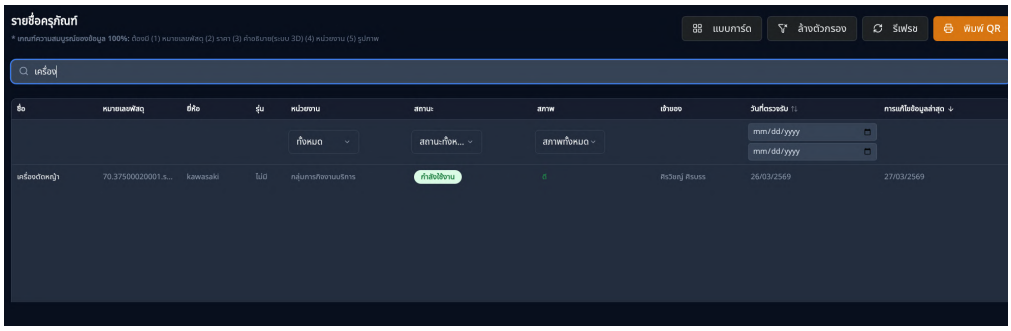
รูปที่ ก.15: หน้าต่างการดูครุภัณฑ์ทั้งหมดที่ผู้ใช้เป็นผู้ดูแล



รูปที่ ก.16: หน้าต่างการดูรายละเอียดครุภัณฑ์แต่ละชิ้น

ก.2.2 การค้นหาครุภัณฑ์ (Assets)

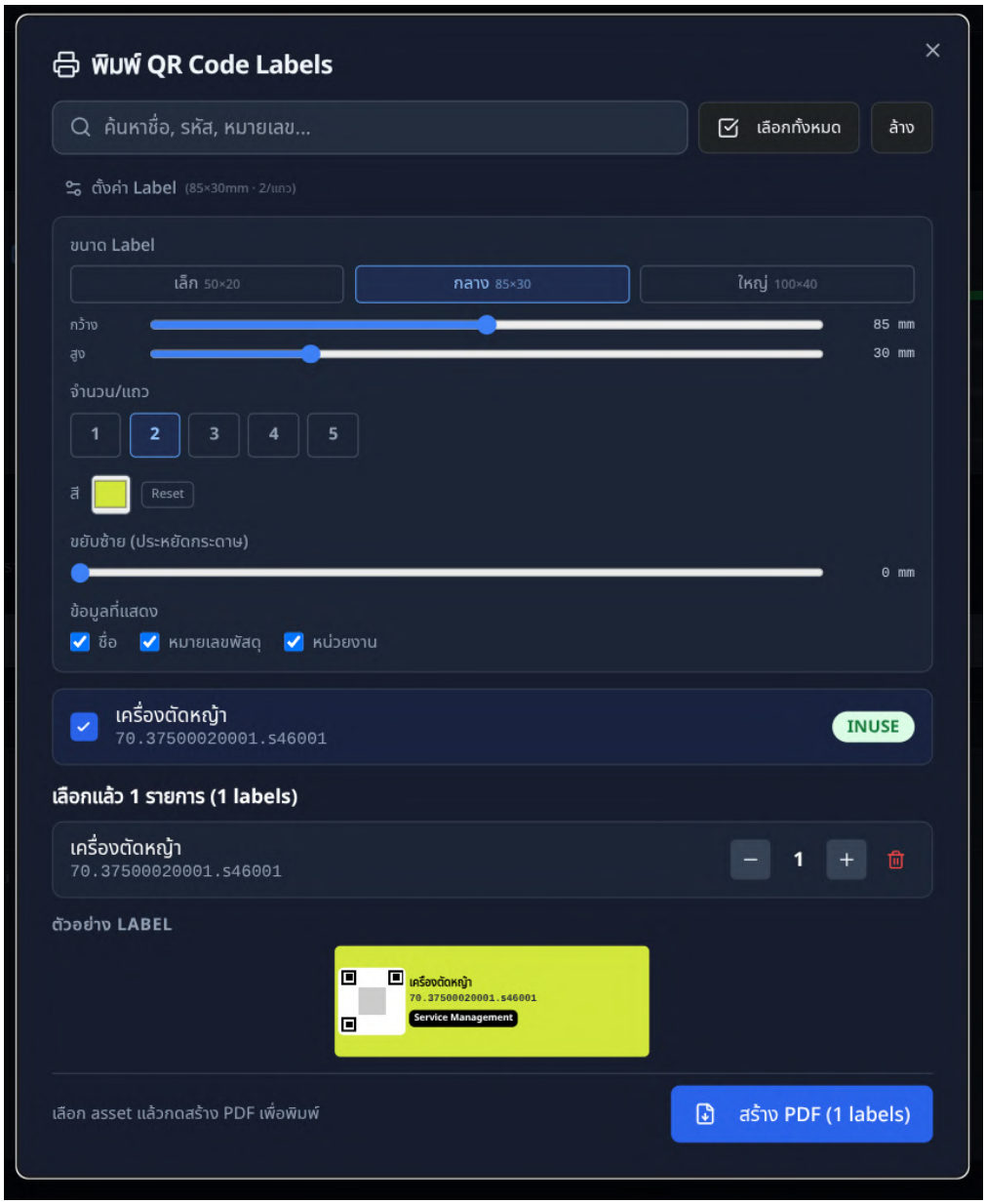
สามารถใช้ช่องค้นหาเพื่อค้นหาครุภัณฑ์ของคุณตามชื่อ, รหัส หรือสถานะ รวมถึงการกรองข้อมูล (Filter) ต่างๆ.



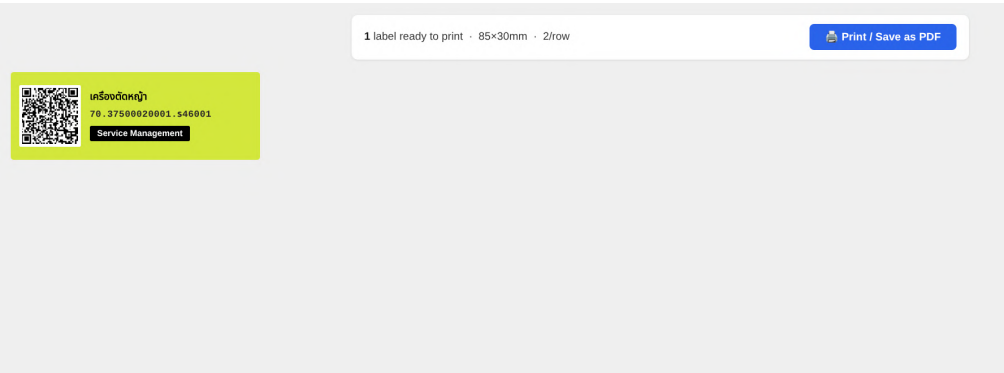
รูปที่ ก.17: การพิมพ์ค้นหาครุภัณฑ์ที่ตรงตามต้องการ

ก.2.3 การพิมพ์ QR Code ครุภัณฑ์ (Assets)

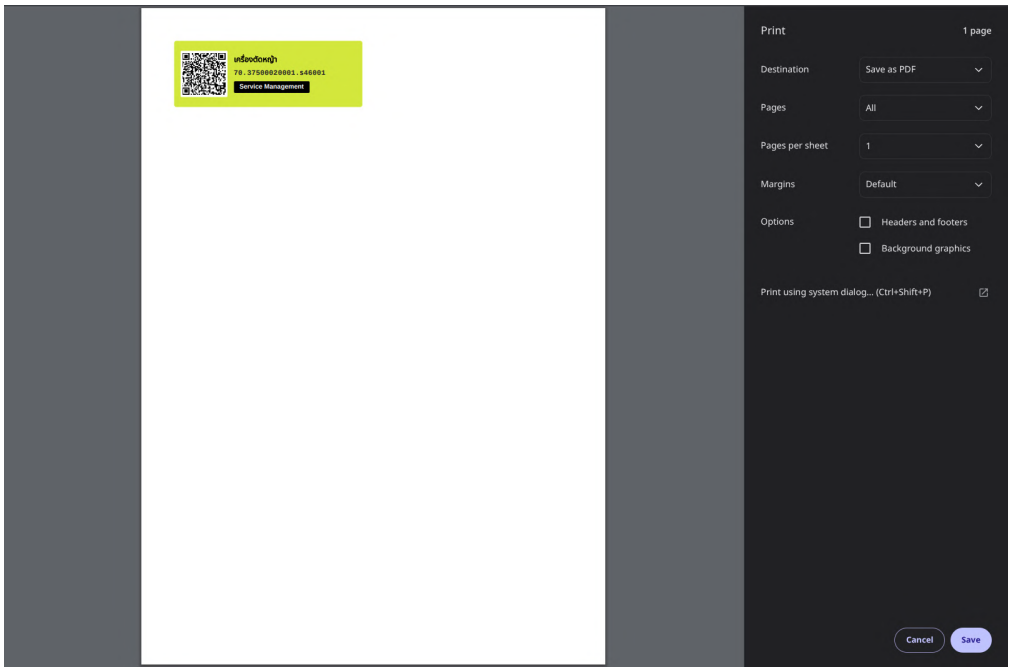
ผู้ใช้สามารถเลือกครุภัณฑ์แล้วกดพิมพ์ QR เพื่อสร้างไฟล์ PDF ของ QR Code สำหรับนำไปพิมพ์และติดที่ตัวครุภัณฑ์ได้.



รูปที่ ก.18: หน้าต่างการตั้งค่า QR Code ที่จะพิมพ์



รูปที่ ก.19: หน้าต่างแสดง QR Code ก่อนพิมพ์

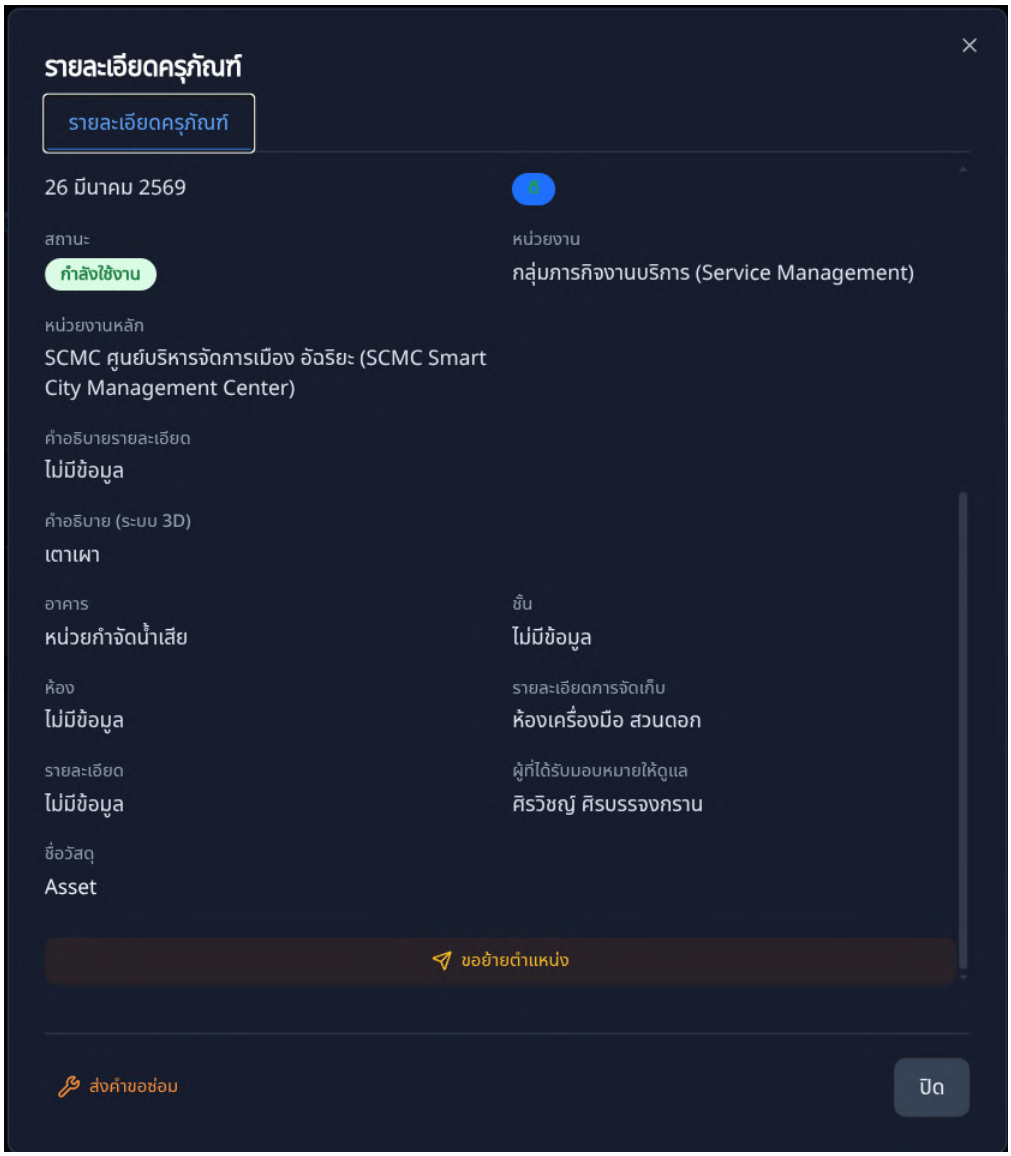


รูปที่ ก.20: หน้าต่างการพิมพ์ QR Code

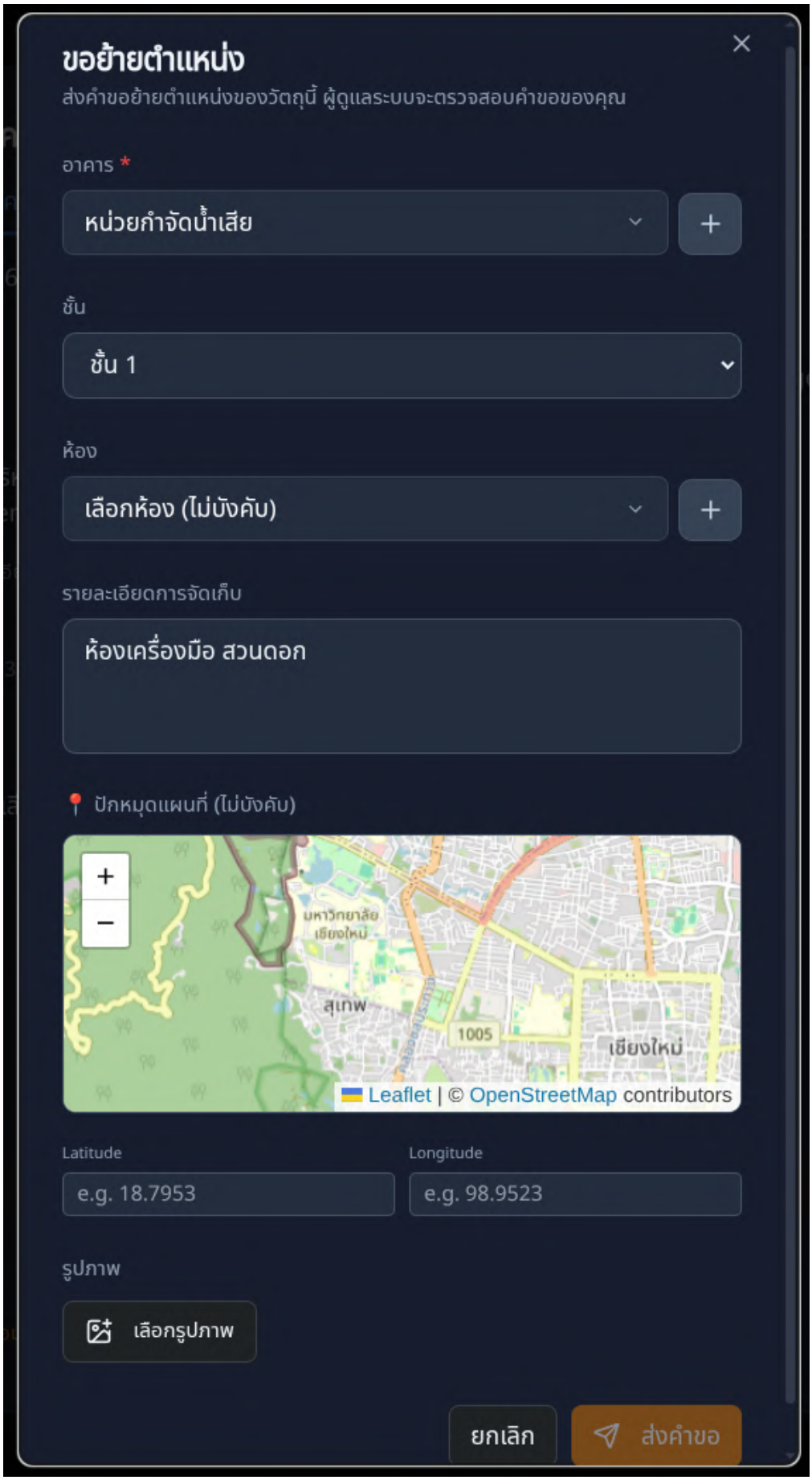
ก.3 การแก้ไขและแจ้งซ่อม

ก.3.1 คำขอย้ายตำแหน่ง

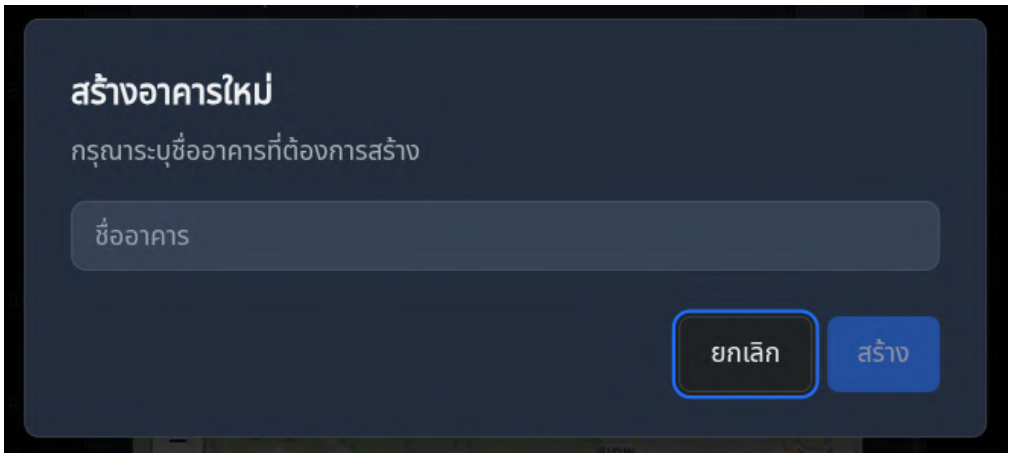
หากมีการย้ายสถานที่ ผู้ใช้ไม่สามารถแก้ไขข้อมูลสถานที่ได้เอง แต่สามารถส่งคำขอย้ายตำแหน่งสถานที่ตั้งและรูปภาพได้ โดยระบุอาคาร, ชั้น, ห้อง พร้อมอัปโหลดรูปภาพใหม่ และต้องรอให้ Admin อนุมัติคำขอ.



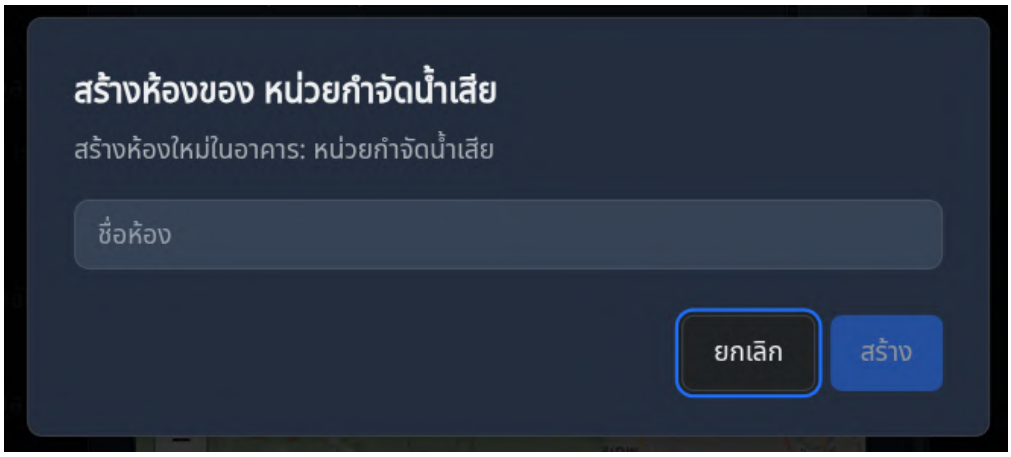
รูปที่ ก.21: ปุ่มขอย้ายตำแหน่งในหน้ารายละเอียดครุภัณฑ์



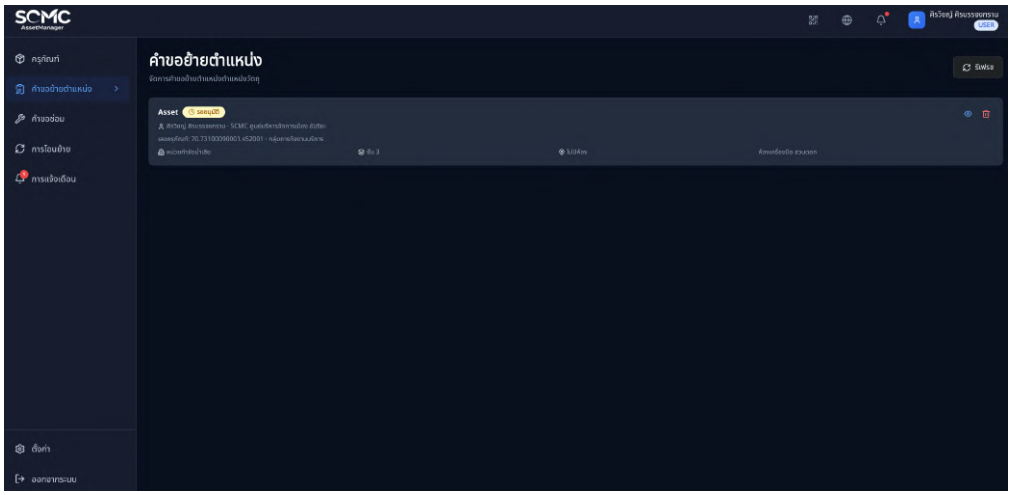
รูปที่ ก.22: หน้าต่างการกรอกข้อมูลตำแหน่งสถานที่ใหม่ของครุภัณฑ์



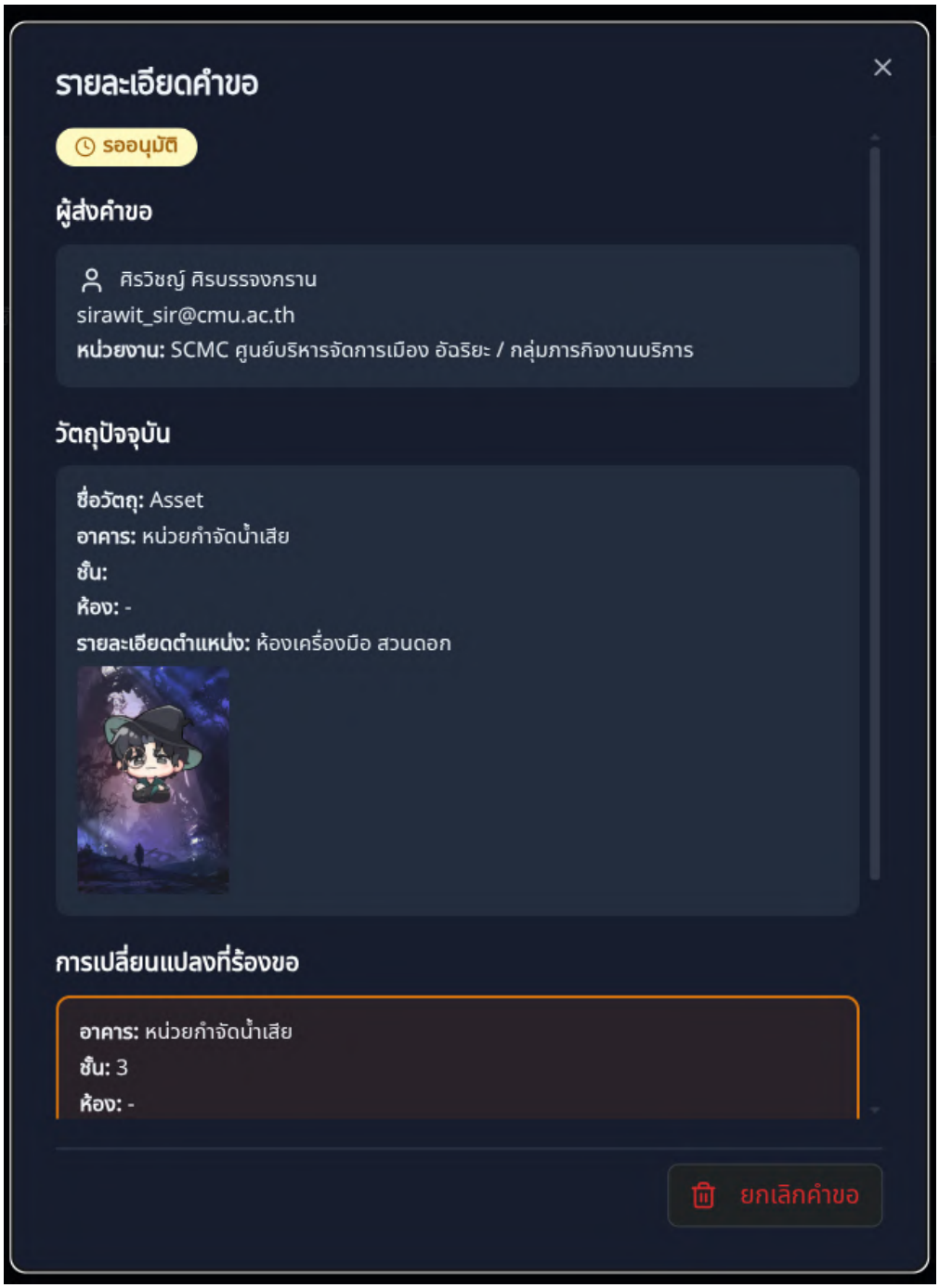
รูปที่ ก.23: ผู้ใช้สามารถกดปุ่ม '+' ที่อาคารเพื่อเพิ่มอาคารใหม่ได้



รูปที่ ก.24: ผู้ใช้สามารถกดปุ่ม '+' ที่ห้องเพื่อเพิ่มห้องใหม่ได้



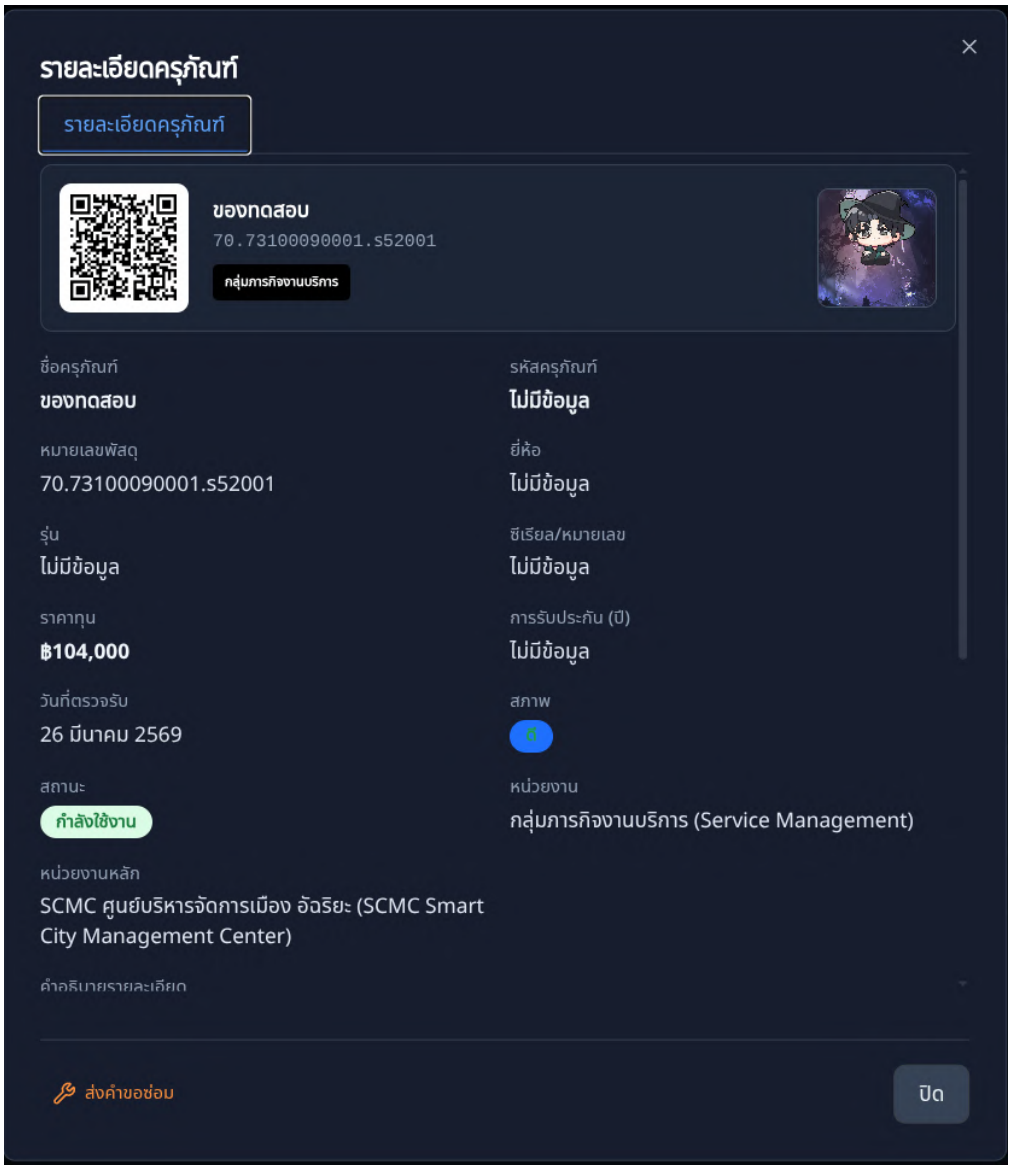
รูปที่ ก.25: คำขอย้ายตำแหน่งครุภัณฑ์จะขึ้นในหน้าคำขอย้ายตำแหน่ง



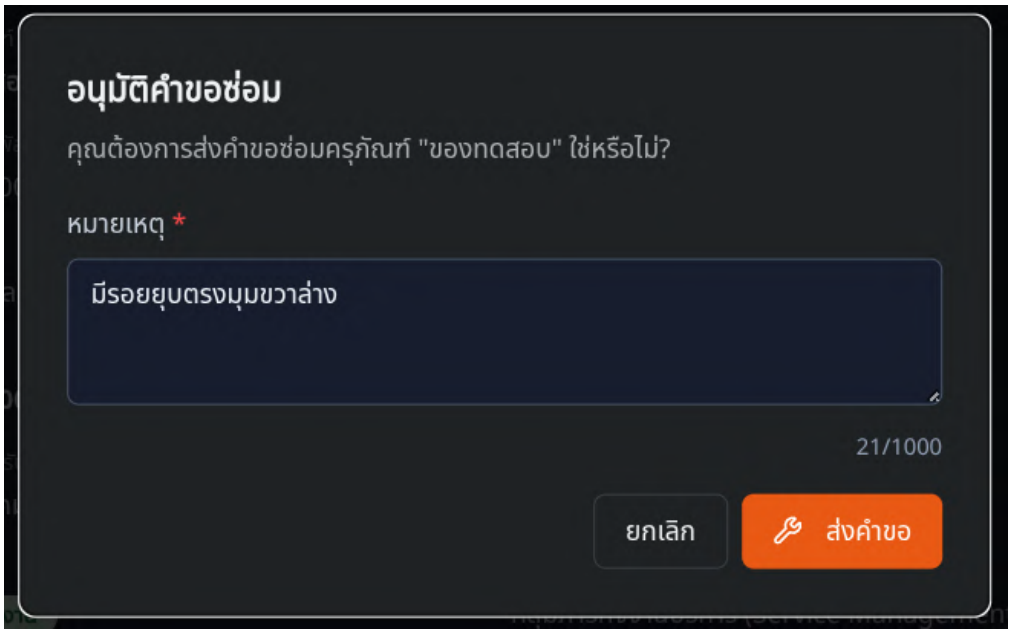
รูปที่ ก.26: หน้าต่างดูรายละเอียดของคำขอย้ายตำแหน่ง ผู้ใช้สามารถยกเลิกคำขอย้ายตำแหน่งได้

ก.3.2 การแจ้งซ่อม

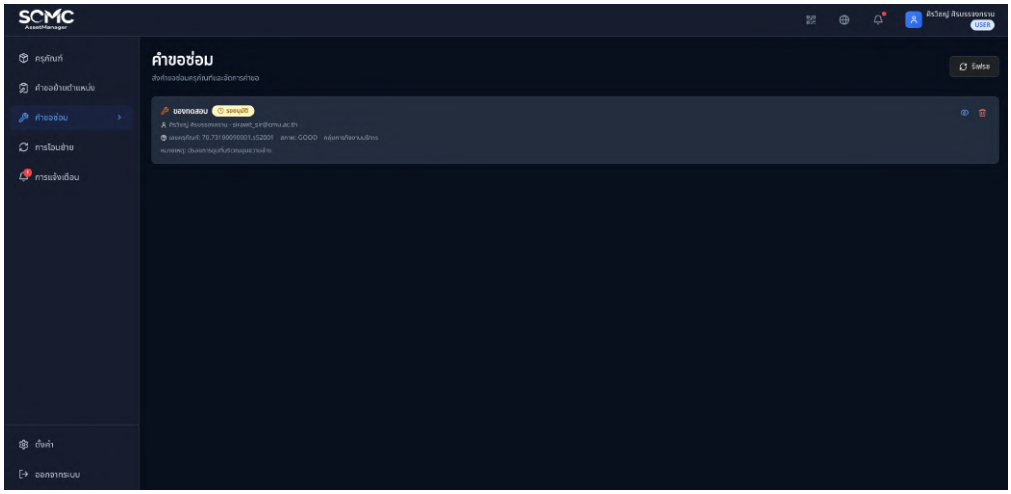
ผู้ใช้งานสามารถยื่นคำร้องแจ้งซ่อมสำหรับครุภัณฑ์ของตนเองโดยระบุรายละเอียดอาการชำรุด. ผู้ใช้สามารถติดตามสถานะใบแจ้งซ่อมของตนเองได้ผ่านระบบ.



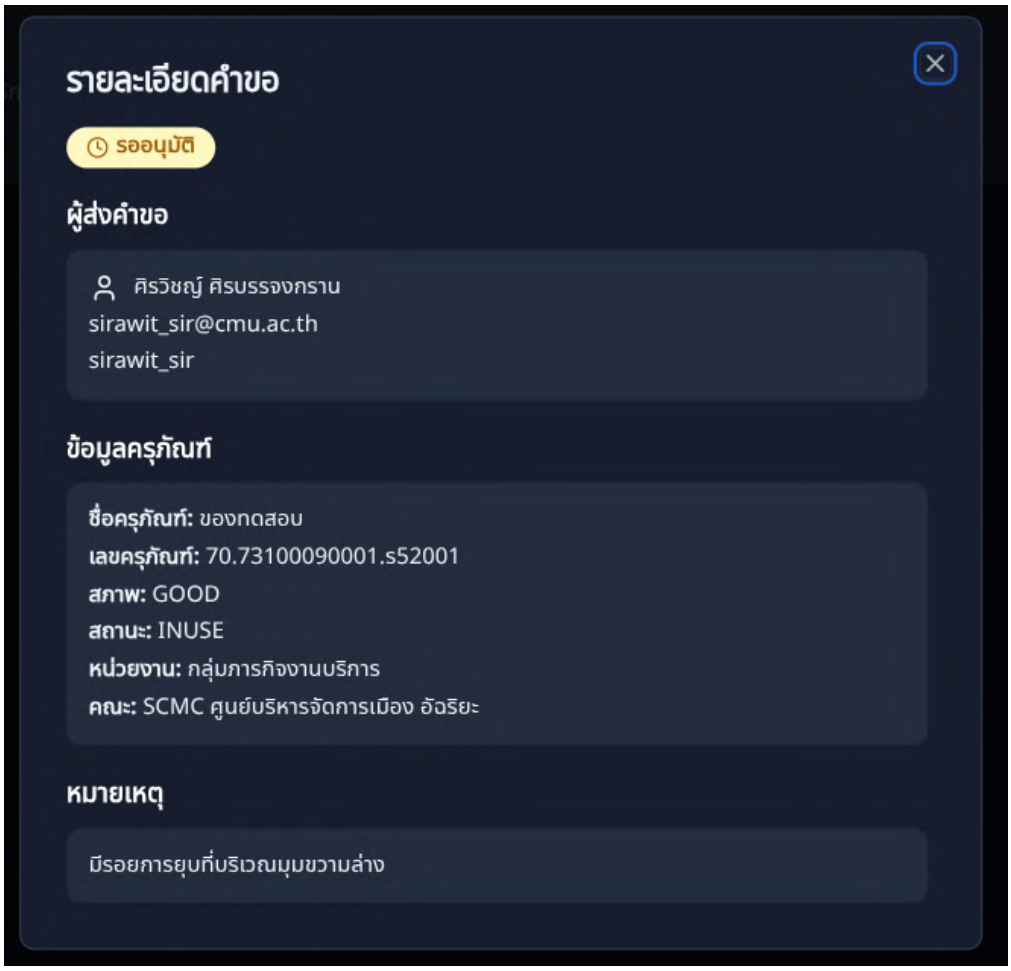
รูปที่ ก.27: ปุ่มแจ้งซ่อมในหน้ารายละเอียดครุภัณฑ์



รูปที่ ก.28: หน้าต่างการส่งคำขอซ่อม



รูปที่ ก.29: คำขอซ่อมครุภัณฑ์จะขึ้นในหน้าคำขอซ่อม

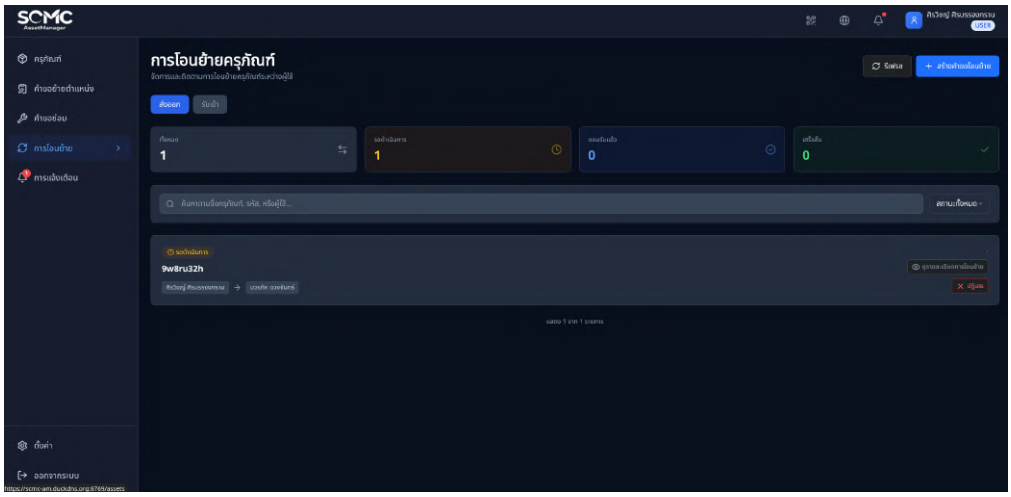


รูปที่ ก.30: หน้าต่างดูรายละเอียดของคำขอซ่อม

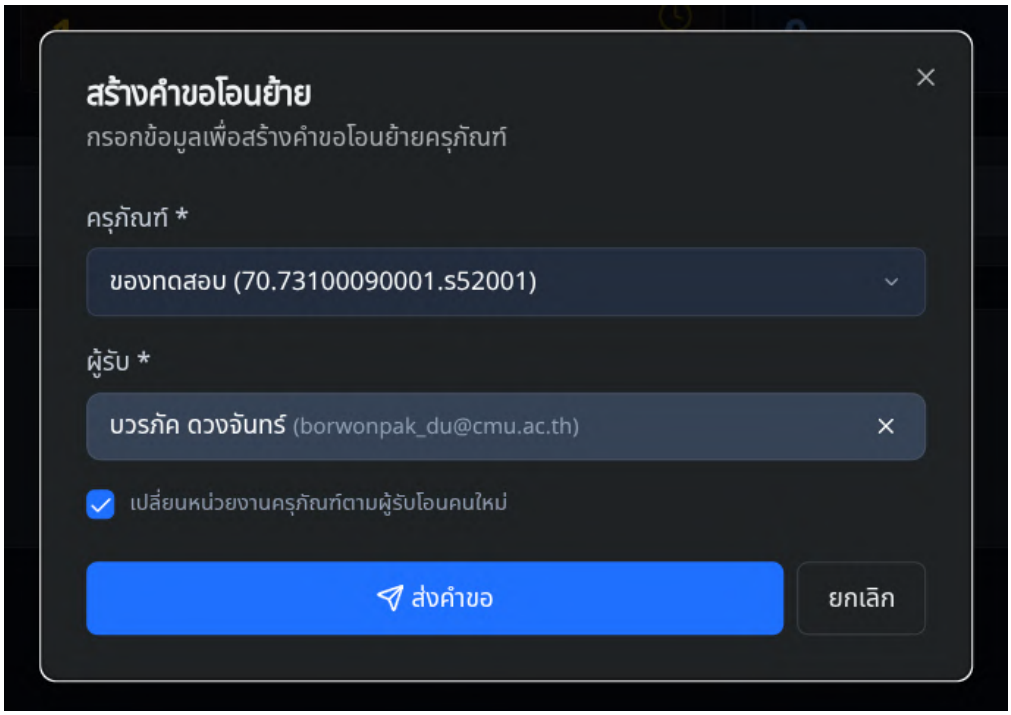
ก.4 การโอนย้ายครุภัณฑ์

ก.4.1 การส่งการโอนย้ายครุภัณฑ์

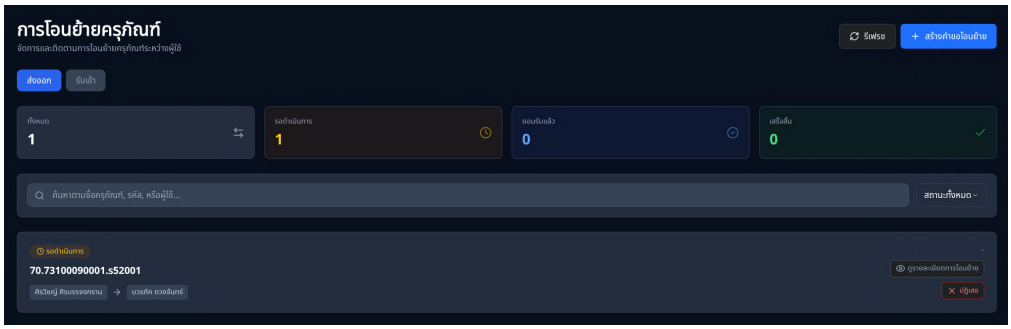
เจ้าของครุภัณฑ์เป็นผู้ริเริ่มการโอนย้ายโดยเลือกครุภัณฑ์และระบุตัวผู้รับโอน. ผู้ส่งสามารถติดตามสถานะคำขอและดูรายการที่ส่งไปแล้วได้.



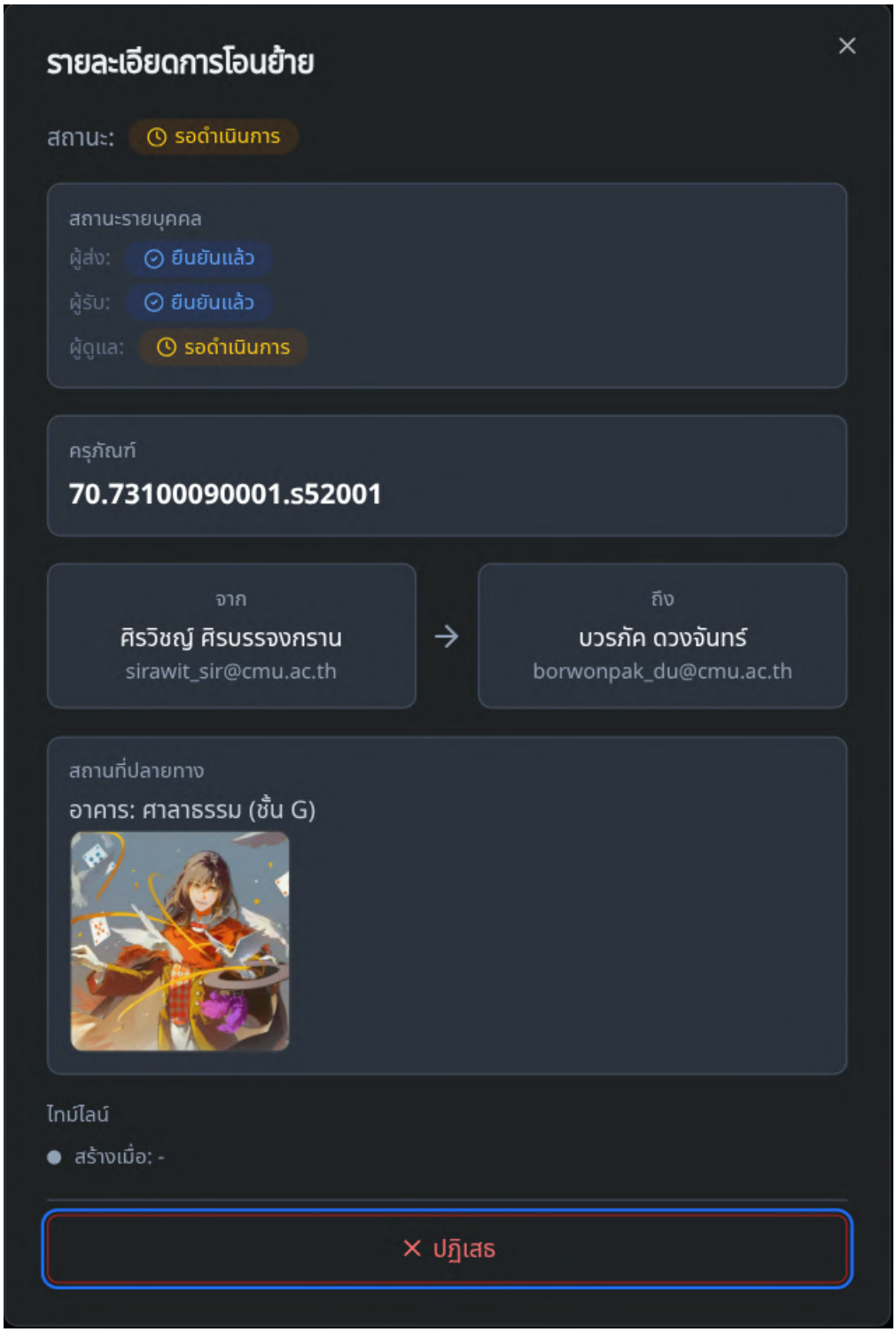
รูปที่ ก.31: มาที่หน้าการโอนย้ายครุภัณฑ์ เพื่อจะเริ่มการโอนย้ายครุภัณฑ์โดยกดที่ปุ่ม 'สร้างคำขอโอนย้าย' ที่มุมขวาบน



รูปที่ ก.32: หน้าต่างการกรอกข้อมูลครุภัณฑ์และผู้ดูแลคนใหม่ และสามารถเลือกให้หน่วยงานของครุภัณฑ์เปลี่ยนตามหน่วยงานของผู้ดูแลคนใหม่ได้



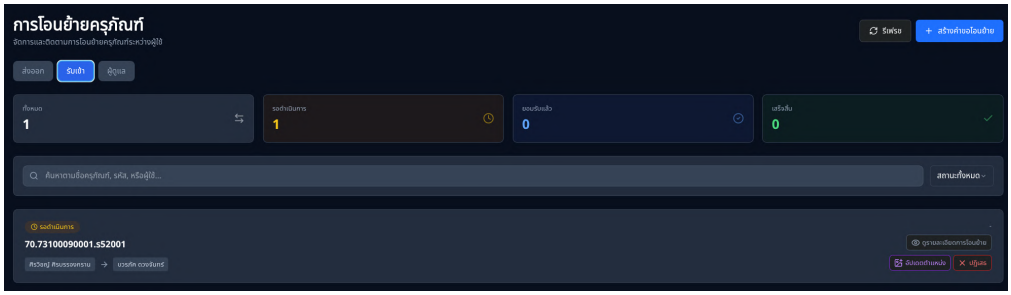
รูปที่ ก.33: หลังจากสร้างคำขอโอนย้ายแล้ว สามารถดูคำขอโอนย้ายที่ผู้ใช้ได้ทำการสร้างในแท็บ ’ส่งออก’



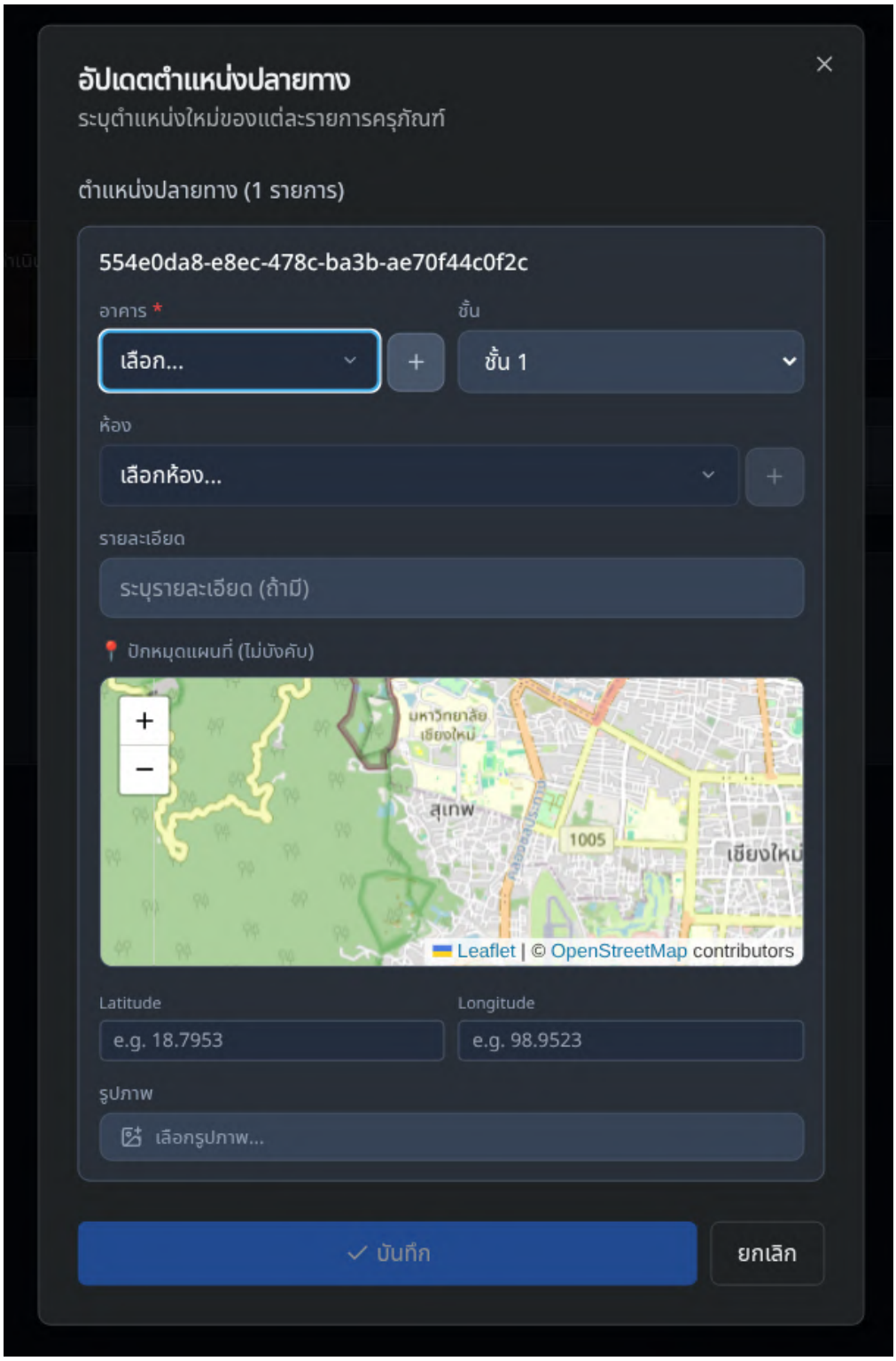
รูปที่ ก.34: ผู้ใช้สามารถดูรายละเอียดตำแหน่งสถานที่ใหม่ของครุภัณฑ์ที่โอนย้ายที่ผู้รับได้ทำการกรอกเข้ามาได้

ก.4.2 การรับการโอนย้ายครุภัณฑ์

ผู้รับโอนจะต้องกดยอมรับการโอนย้ายผ่านระบบ. ในขณะที่กดยอมรับ ผู้รับโอนจะต้องอัปเดตสถานที่ตั้งใหม่ (อาคาร, ชั้น, ห้อง) และอัปโหลดภาพถ่ายปัจจุบันของครุภัณฑ์. ผู้รับสามารถดูรายการที่ได้รับมาได้.



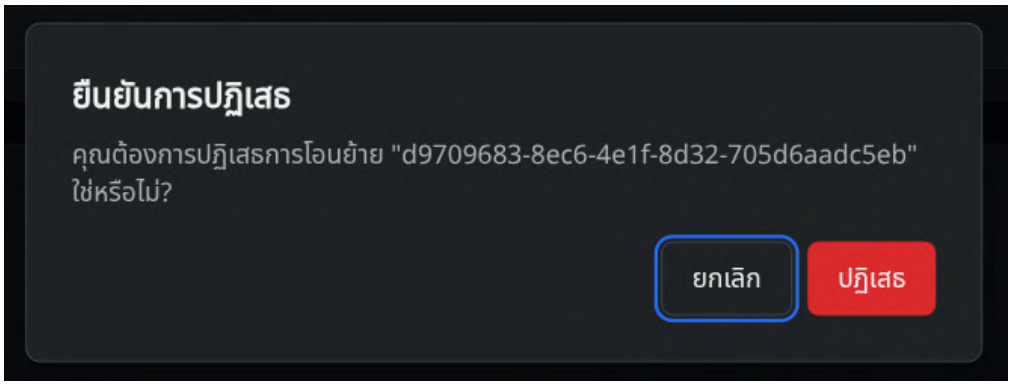
รูปที่ ก.35: ผู้รับสามารถดูคำขอโอนย้ายที่จะทำการรับไปดูแลได้ในแท็บ 'รับเข้า'



รูปที่ ก.36: ผู้รับต้องเข้ามากรอกข้อมูลสถานที่ใหม่ของครุภัณฑ์

ก.4.3 การปฏิเสธการโอนย้ายครุภัณฑ์

ฝ่ายใดฝ่ายหนึ่งสามารถกดปฏิเสธ (Deny) การโอนย้ายได้ตลอดเวลาก่อนที่กระบวนการจะเสร็จสิ้น. หากผู้ส่งและผู้รับปฏิเสธการโอนย้ายจะถูกปฏิเสธทันที

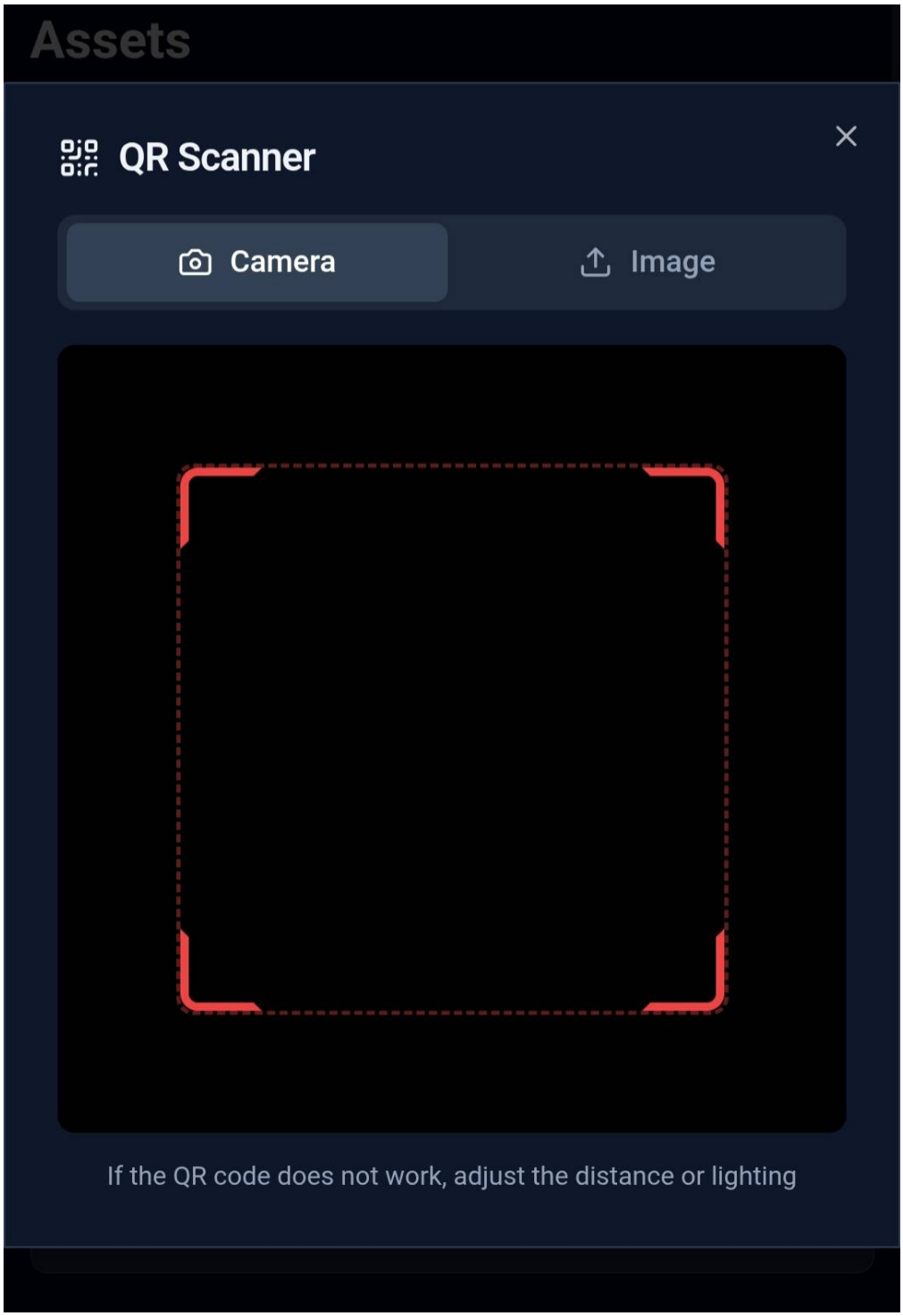


รูปที่ ก.37: ทั้งผู้ส่งและผู้รับสามารถกดปุ่ม 'ปฏิเสธ' เพื่อเข้ามาหน้าต่างการยืนยันการปฏิเสธการโอนย้ายได้

ก.5 ระบบสแกน QR Code

ก.5.1 การสแกน

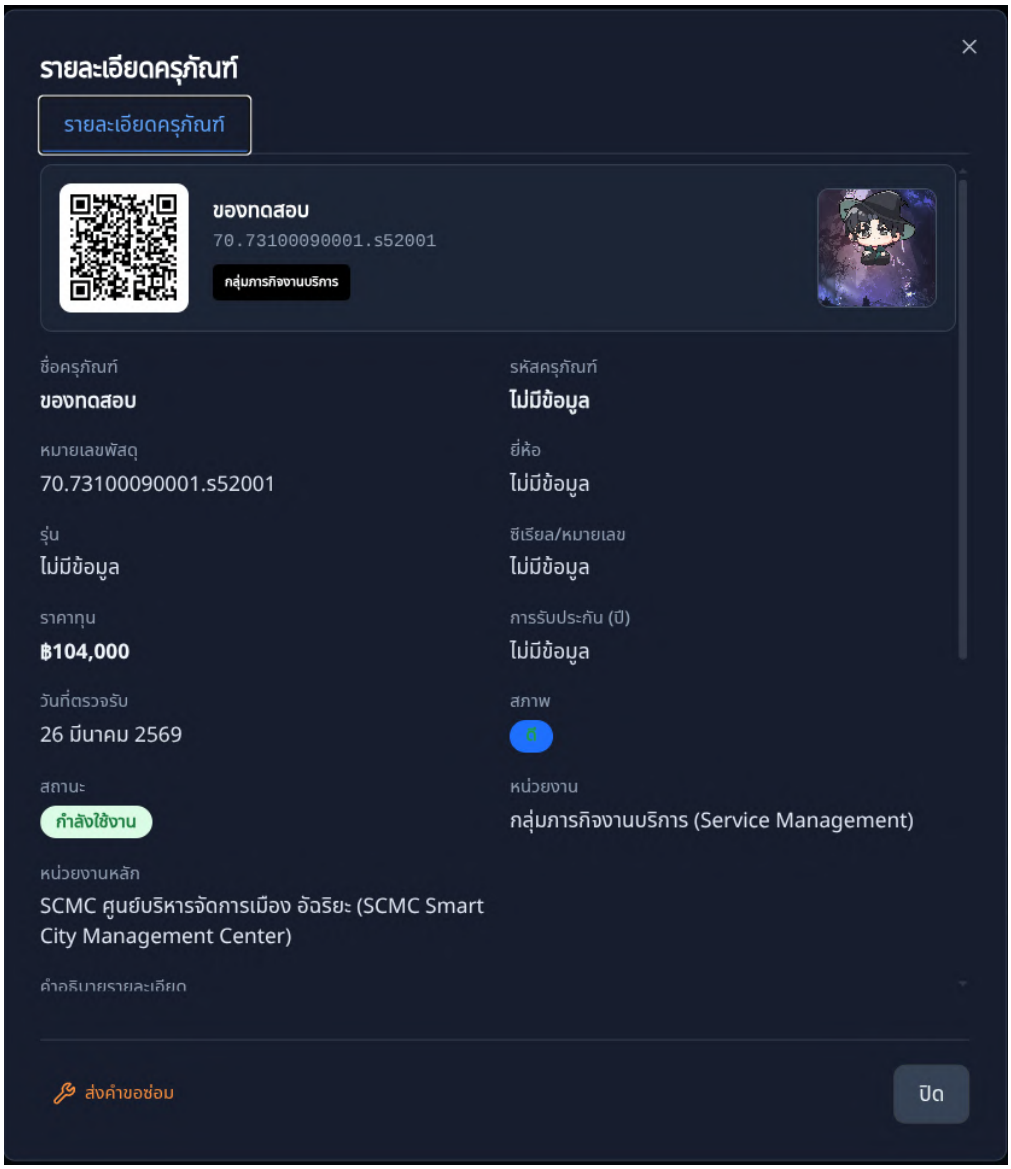
ผู้ใช้สามารถใช้ตัวสแกน QR ที่แถบเมนูด้านบนเพื่อเปิดกล้องและสแกน QR Code ของครุภัณฑ์. หรือจะใช้กล้องโทรศัพท์สแกนก็ได้



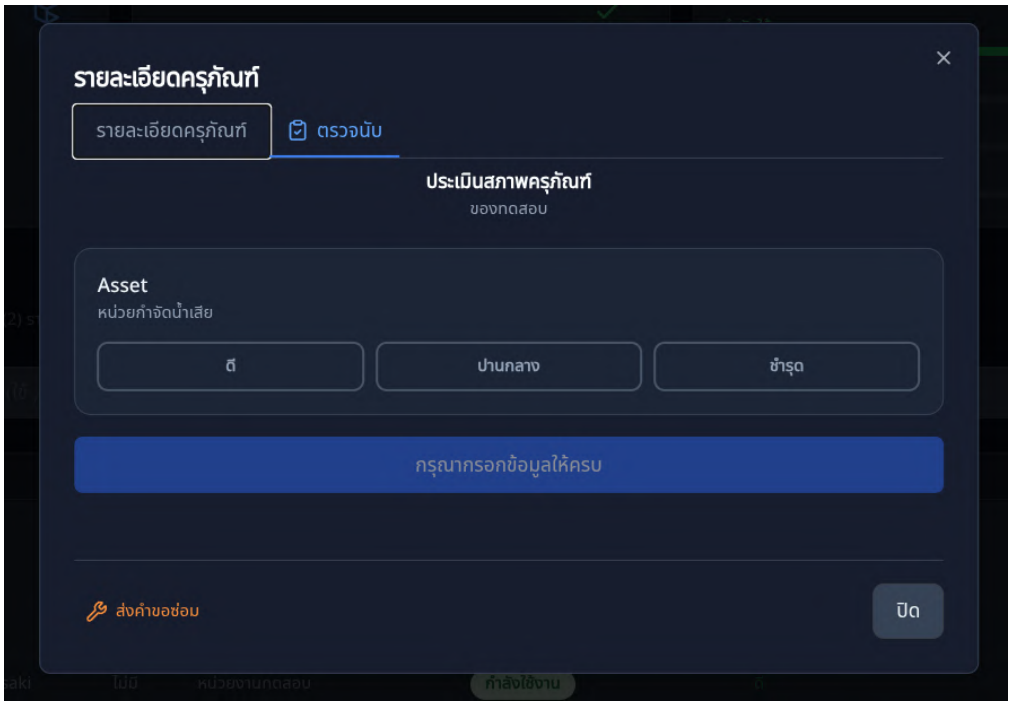
รูปที่ ก.38: หน้าต่างเปิดกล้องการสแกน QR Code บนระบบของเรา

ก.5.2 ผลลัพธ์การสแกน

เมื่อสแกนแล้ว ระบบจะแสดงข้อมูลรายละเอียดครุภัณฑ์นั้นทันที. ในหน้าข้อมูลหลังการสแกนจะมีปุ่ม ”แจ้งซ่อม” เพื่อเชื่อมไปยังฟอร์มขอส่งซ่อมได้ทันที. หากมีรอบการตรวจนับเปิดอยู่ ระบบจะแสดงอินเทอร์-เฟซสำหรับการตรวจนับให้ทันทีเมื่อสแกน.



รูปที่ ก.39: หน้าต่างการดูรายละเอียดครุภัณฑ์จะแสดงขึ้นมาหลังสแกน

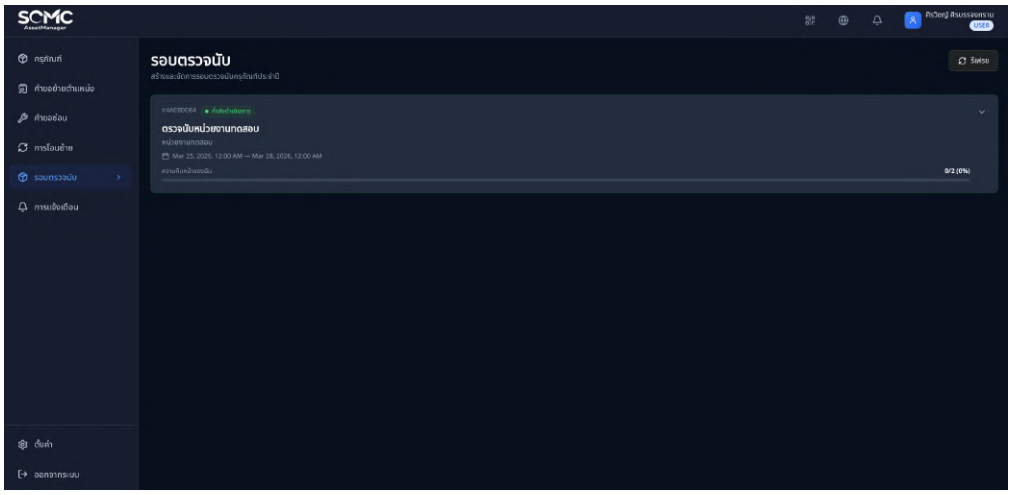


รูปที่ ก.40: ในกรณีที่ QR ที่สแกนเป็นครุภัณฑ์ที่อยู่ระหว่างการตรวจนับ จะแสดงหน้าต่างการตรวจสภาพครุภัณฑ์ (จะลงรายละเอียดในส่วนการตรวจนับอีกที)

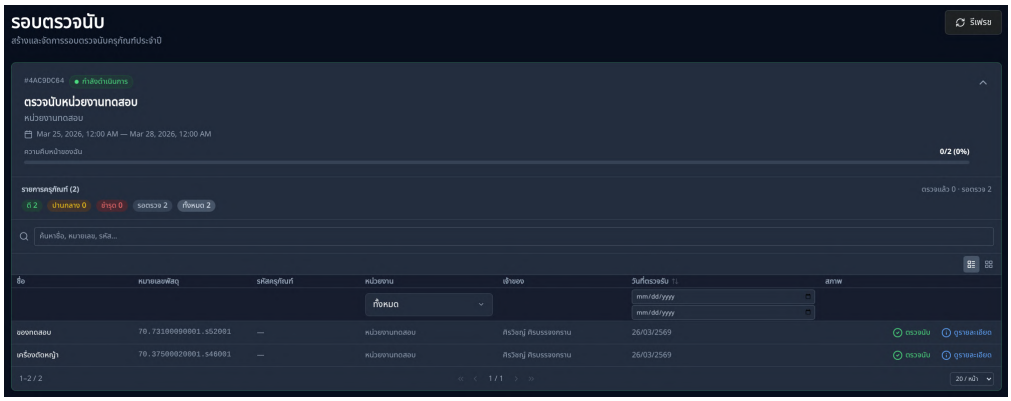
ก.6 รอบการตรวจนับครุภัณฑ์

ก.6.1 การตรวจสอบสถานะ

ผู้ใช้งานสามารถดูสถานะความคืบหน้าส่วนตัวในรอบการตรวจนับได้.



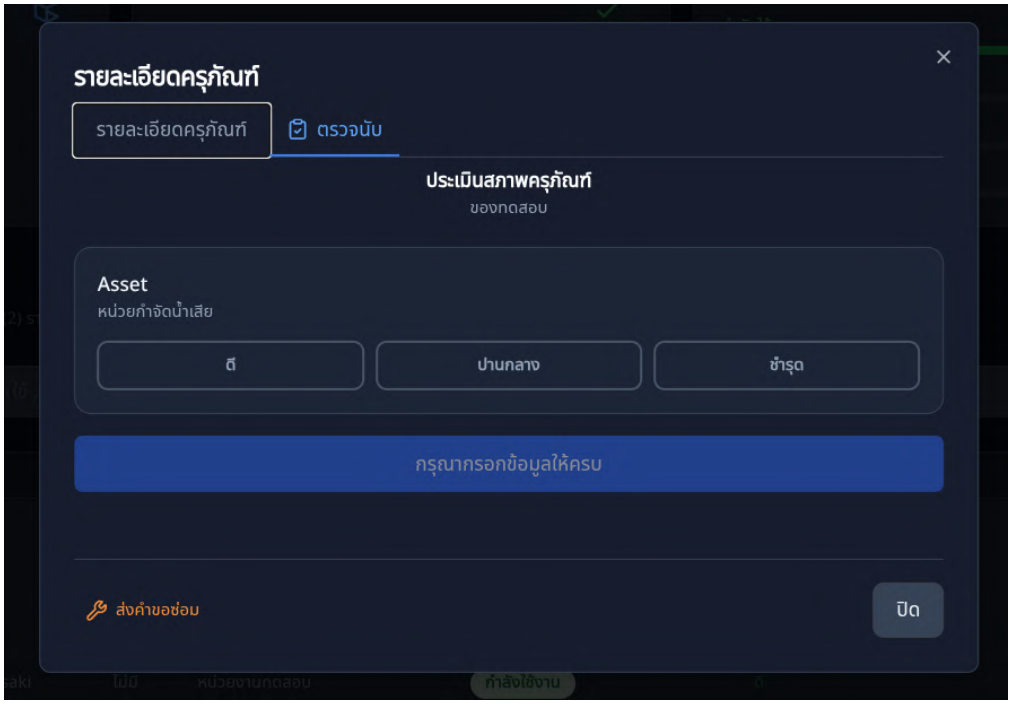
รูปที่ ก.41: เมื่อมีรอบตรวจนับครุภัณฑ์ที่เกี่ยวข้องกับผู้ใช้เกิดขึ้น จะมีแท็บ 'รอบตรวจนับ' ที่ผู้ใช้สามารถกดเข้ามาดูหน้ารอบตรวจนับได้



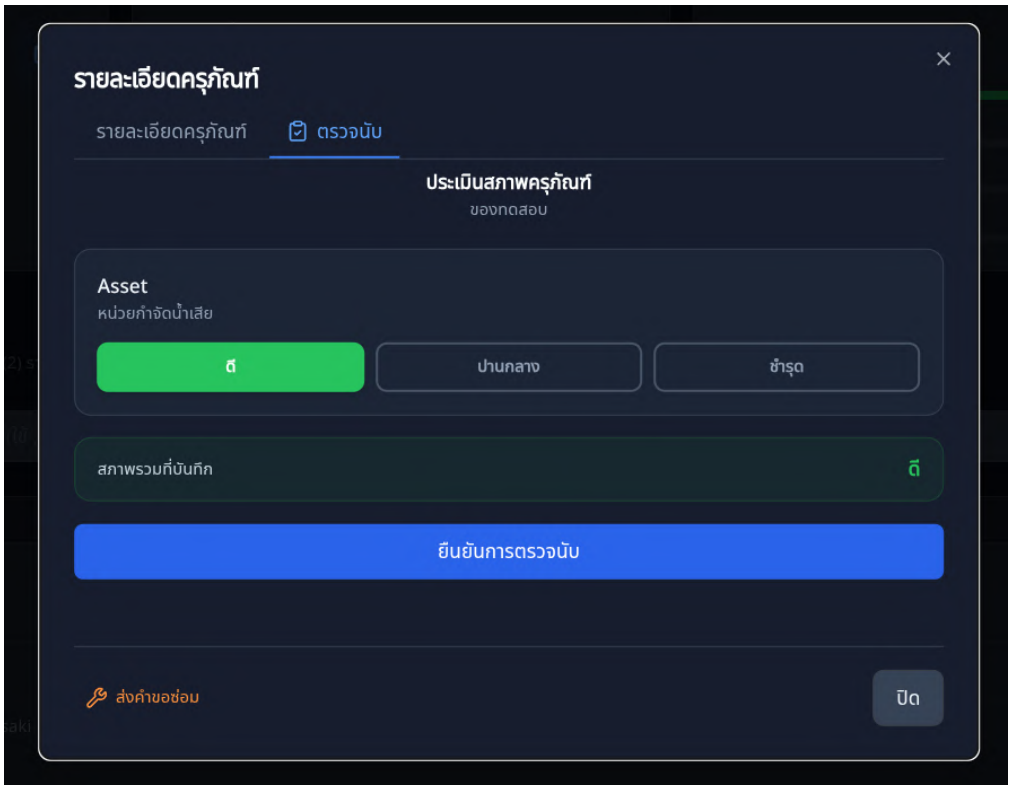
รูปที่ ก.42: ผู้ใช้สามารถกดขยายดูรายละเอียดต่าง ๆ ได้ว่าต้องตรวจนับครุภัณฑ์ชิ้นไหนบ้าง ตรวจครบหรือยัง

ก.6.2 การรายงานสภาพ

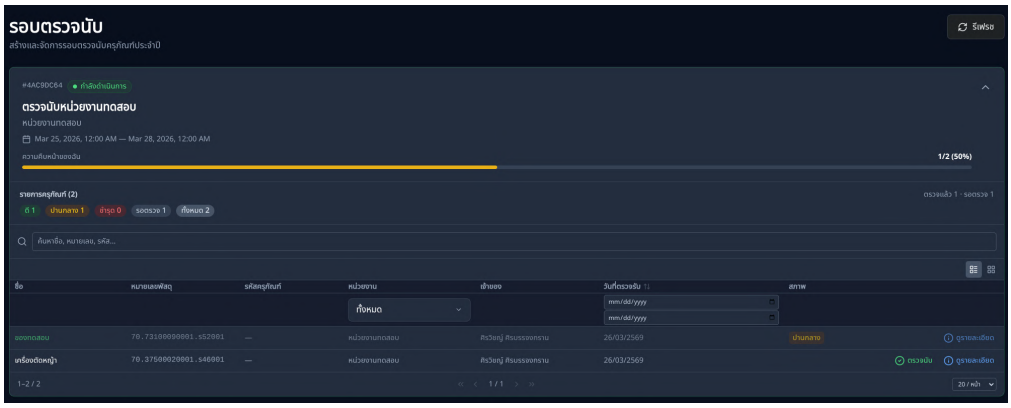
ผู้ใช้สามารถยืนยันการตรวจนับครุภัณฑ์ของตนในช่วงที่รอบกำลังเปิดใช้งาน โดยเลือกประเมินสภาพ (ดี / ปานกลาง / ชำรุด) พร้อมอัปเดตรายละเอียด



รูปที่ ก.43: ผู้ใช้สามารถกดปุ่ม 'ตรวจนับ' เพื่อเข้ามาหน้าการตรวจสภาพครุภัณฑ์ได้



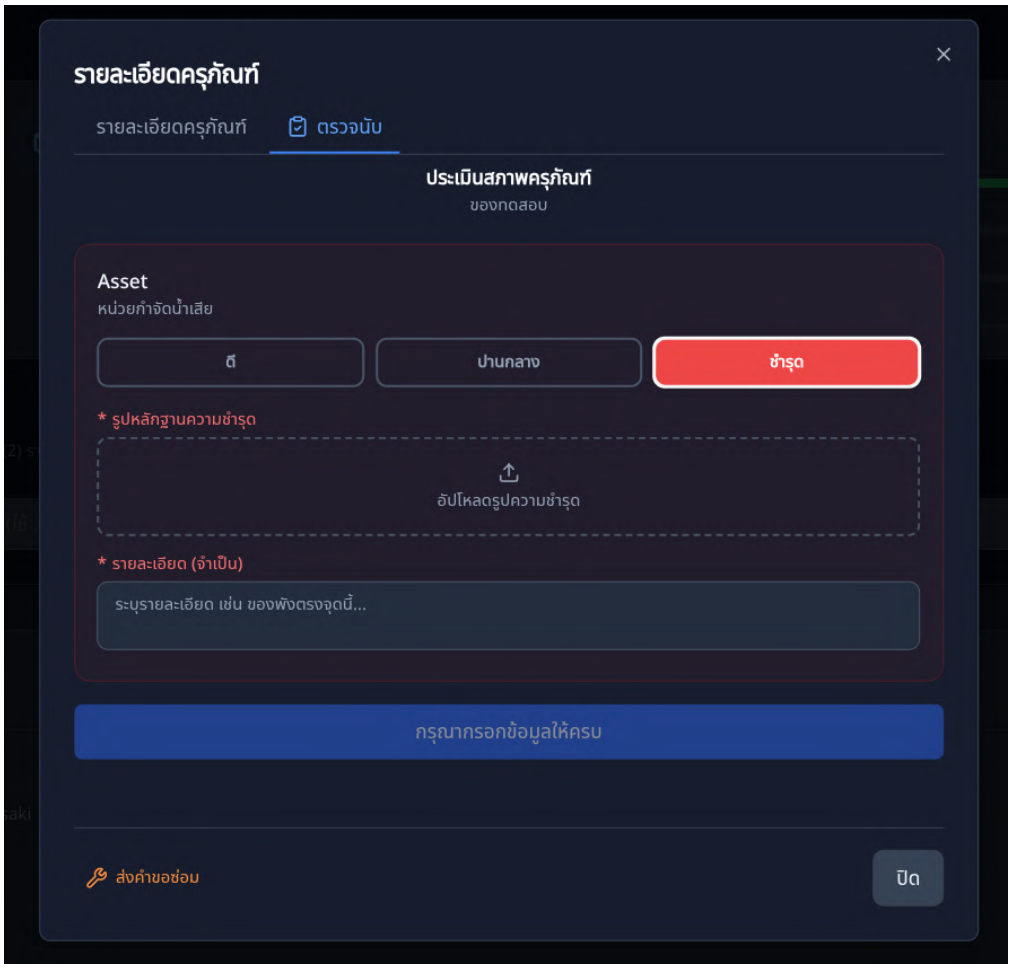
รูปที่ ก.44: ผู้ใช้ต้องเลือกสภาพของครุภัณฑ์หลังจากทำการตรวจสอบแล้ว



รูปที่ ก.45: เมื่อตรวจนับเสร็จแล้วกลับมาที่หน้ารอบตรวจนับ จะเห็นความคืบหน้าขอการตรวจนับ

ก.6.3 กรณีชำรุด

หากผู้ใช้ประเมินสภาพครุภัณฑ์หรือรายการย่อยเป็น ”ชำรุด” ระบบจะบังคับให้อัปโหลดรูปภาพความเสียหายและระบุรายละเอียดการจัดเก็บใหม่ทันที.

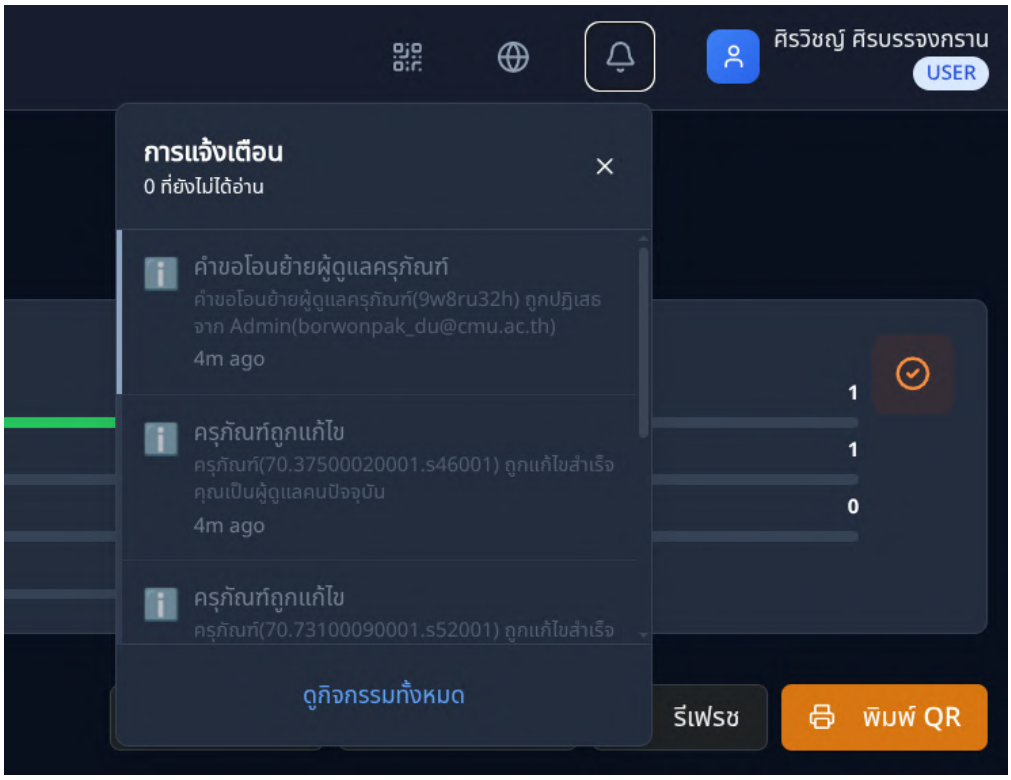


รูปที่ ก.46: หากผู้ใช้เลือกสภาพชำรุดตอนตรวจนับ ผู้ใช้จะต้องอัพโหลดรูปภาพครุภัณฑ์ปัจจุบันและกรอกรายละเอียดการชำรุด

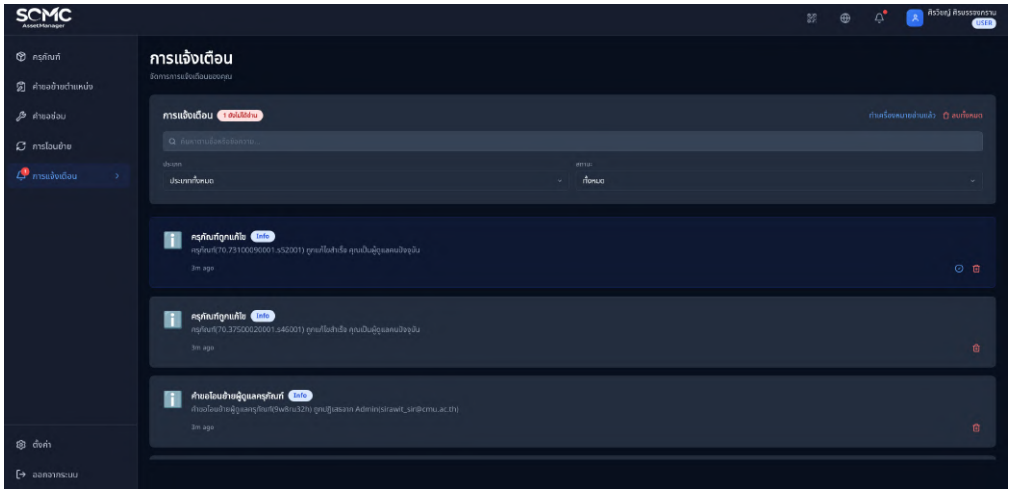
ก.7 การแจ้งเตือน

ก.7.1 การรับข่าวสารและการจัดการข้อความ

ระบบจะส่งการแจ้งเตือนเมื่อครุภัณฑ์ถูกเพิ่ม/แก้ไข/ลบ, เมื่อมีกระบวนการโอนย้าย, และเมื่อมีการสร้างรอบการตรวจนับครุภัณฑ์. การแจ้งเตือนทำงานแบบเรียลไทม์ผ่าน SSE. มีไอคอนตัวเลขแสดงจำนวนข้อความที่ยังไม่ได้อ่านบนแถบเมนูด้านบน. ผู้ใช้สามารถกดทำเครื่องหมายว่าอ่านแล้ว, ลบข้อความรายบุคคล, หรือกดลบข้อความแจ้งเตือนทั้งหมดพร้อมกันได้.



รูปที่ ก.47: หน้าต่างการแจ้งเตือนทางด้านบน



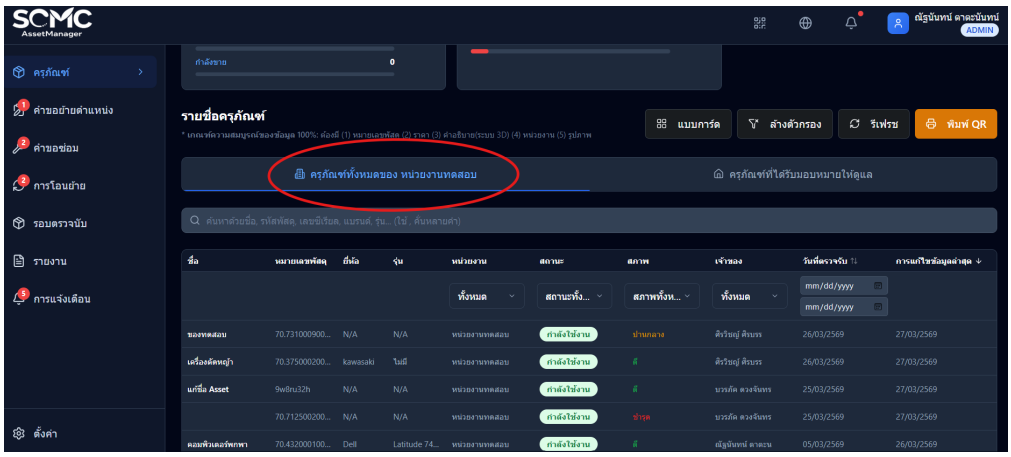
รูปที่ ก.48: หน้าการจัดการการแจ้งเตือน

ก.8 คู่มือสำหรับผู้ดูแล (ADMIN)

ผู้ดูแลสามารถใช้งานฟีเจอร์ทั้งหมดของผู้ใช้งานทั่วไปได้ และมีฟีเจอร์เพิ่มเติมดังนี้

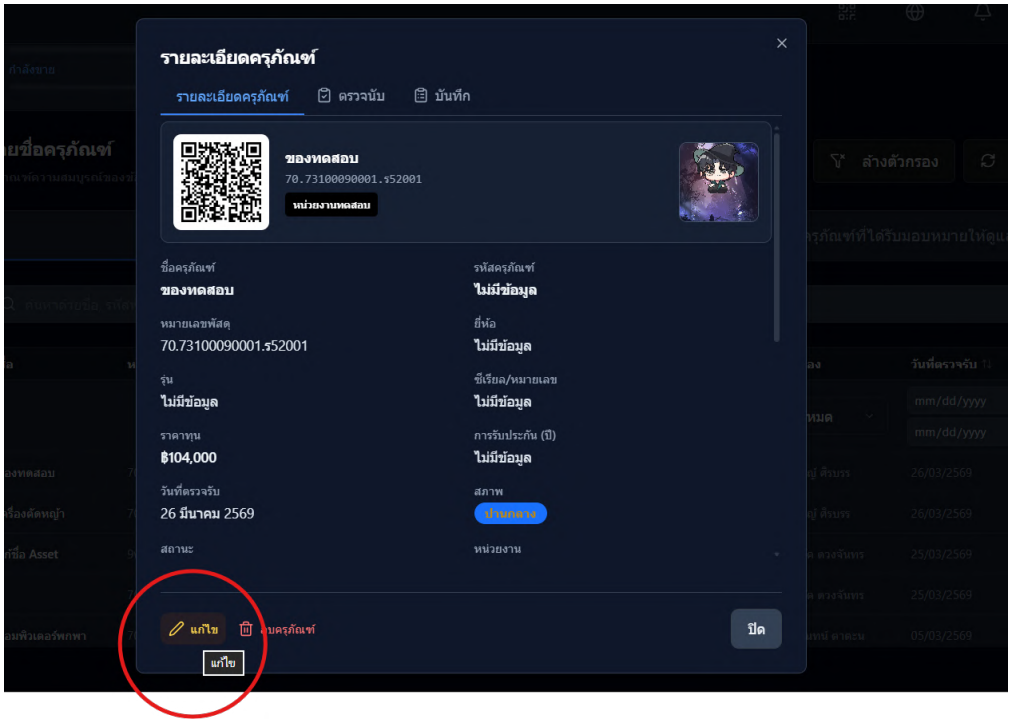
ก.8.1 จัดการครุภัณฑ์ (Admin Assets)

สิทธิ์การมองเห็น: ผู้ดูแลจะมองเห็นครุภัณฑ์ทั้งหมดที่อยู่ภายใต้โครงสร้างหน่วยงานของตนผ่านแท็บหน่วยงาน



รูปที่ ก.49: หน้ารายการครุภัณฑ์สำหรับผู้ดูแลและการแก้ไขข้อมูล

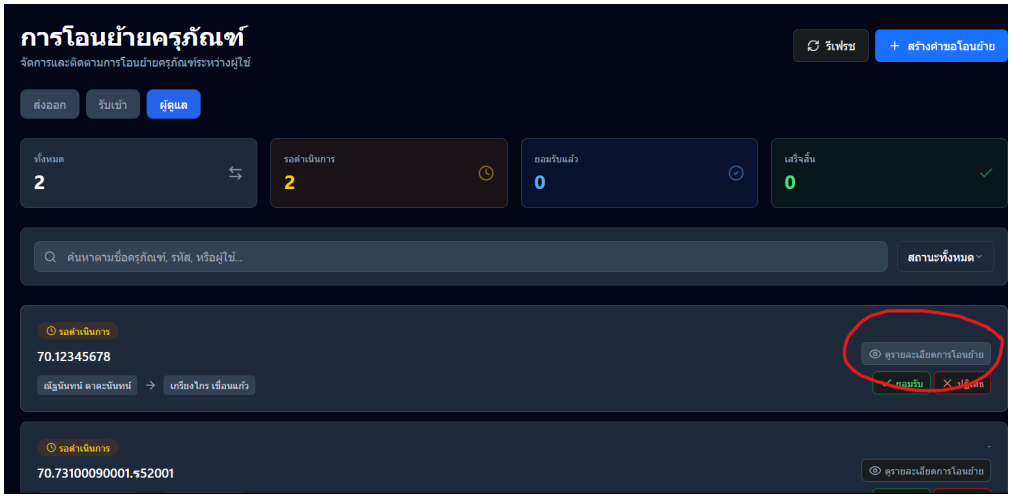
การแก้ไขข้อมูล: ผู้ดูแลสามารถแก้ไขข้อมูลครุภัณฑ์ที่อยู่ในความดูแลของหน่วยงานตนเองได้ทุกฟิลด์โดยตรง โดยไม่ต้องส่งคำขอ



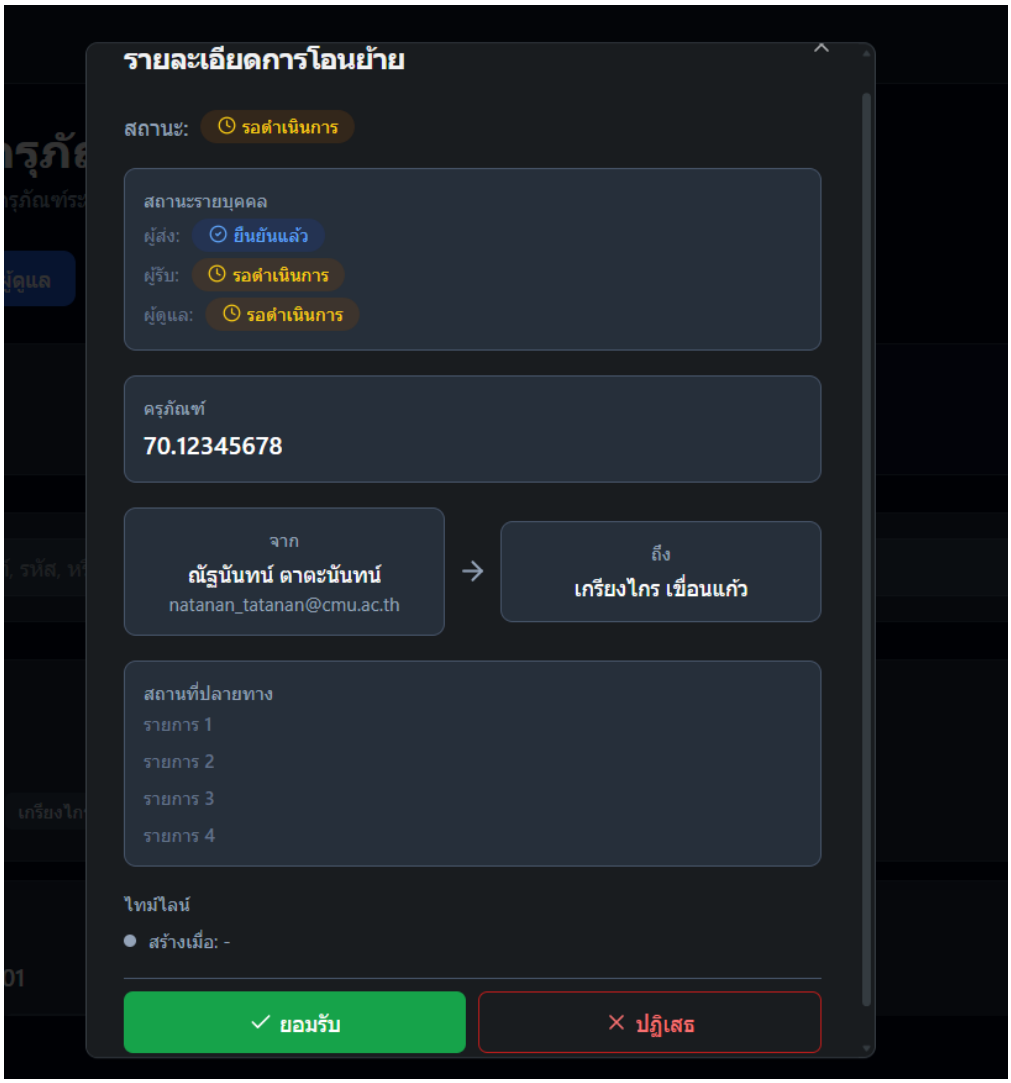
รูปที่ ก.50: ปุ่มสำหรับแก้ไขข้อมูลครุภัณฑ์สำหรับผู้ดูแล

ก.8.2 จัดการการโอนย้ายและคำร้อง (Admin Transfers & Requests)

การอนุมัติการโอนย้าย: กรรมสิทธิ์ผู้ครอบครองจะถูกเปลี่ยนก็ต่อเมื่อผู้ดูแล (Admin) ของหน่วยงานกดยืนยันการโอนย้ายเป็นลำดับสุดท้าย ผู้ดูแลสามารถดูรายการโอนย้ายทั้งหมดที่เกิดขึ้นภายในโครงสร้างหน่วยงานของตน และสามารถอนุมัติหรือปฏิเสธคำขอได้

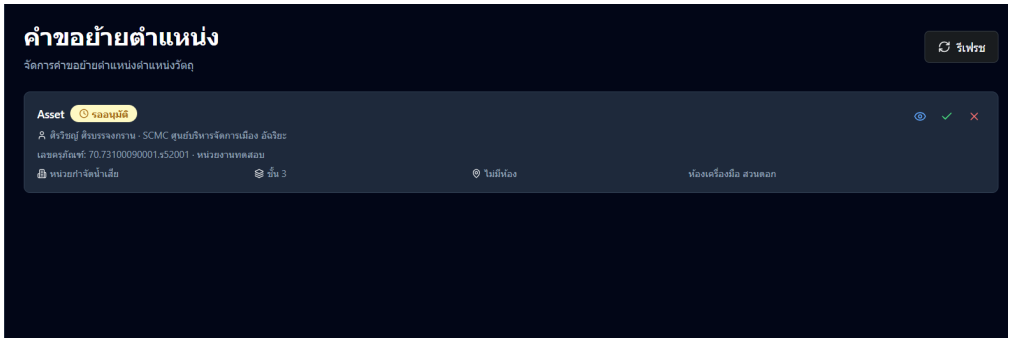


รูปที่ ก.51: หน้าต่างตรวจสอบการโอนย้ายสำหรับผู้ดูแล

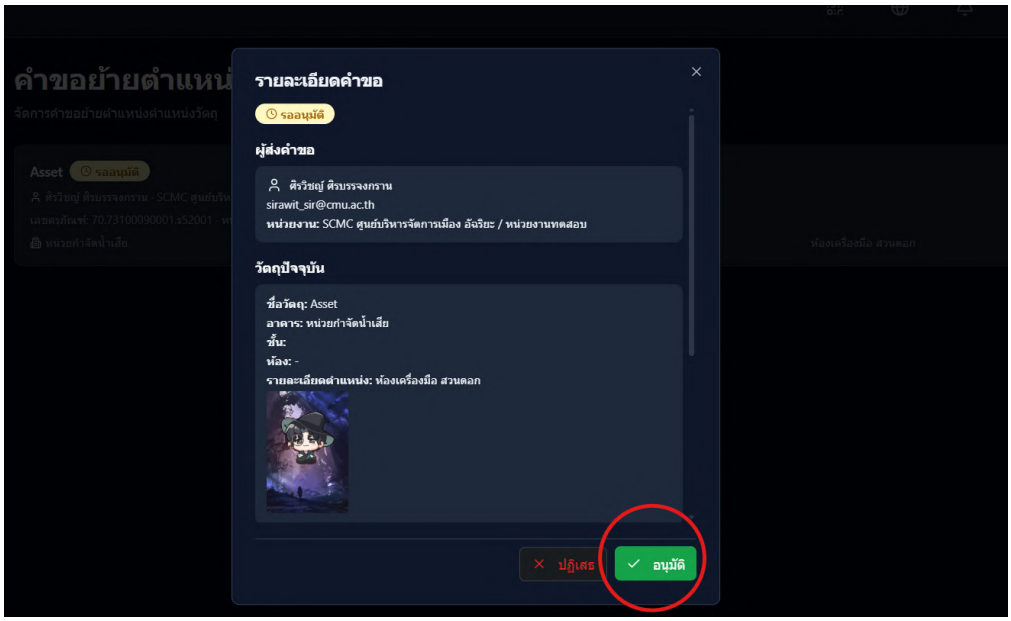


รูปที่ ก.52: หน้าต่างอนุมัติการโอนย้ายสำหรับผู้ดูแล

การอนุมัติคำขอย้ายตำแหน่ง: ผู้ดูแลมีหน้าที่ตรวจสอบและสามารถอนุมัติหรือปฏิเสธคำขอย้ายตำแหน่งสถานที่ตั้งและรูปภาพที่ส่งมาจากผู้ใช้งานทั่วไปได้

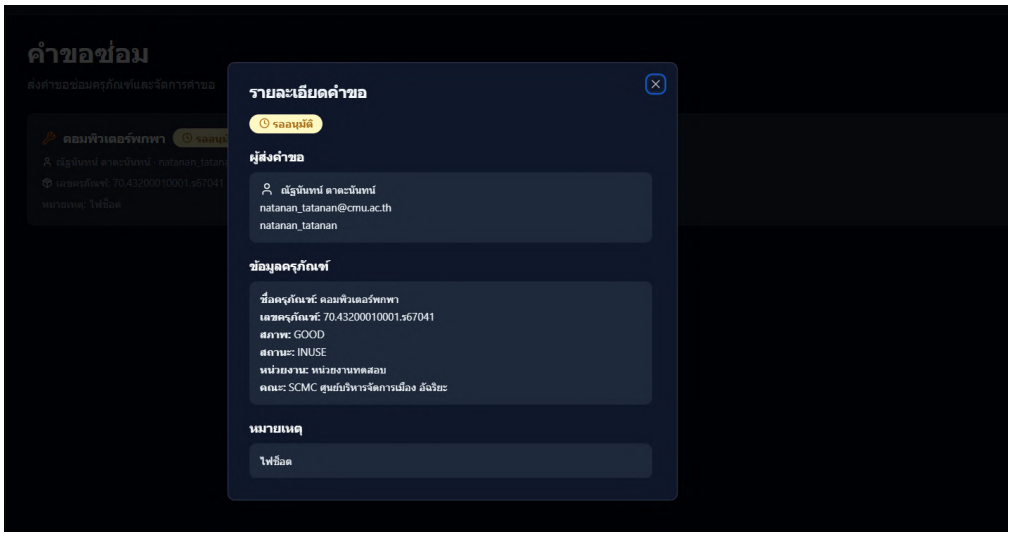


รูปที่ ก.53: หน้าต่างดูคำขอย้ายตำแหน่งสำหรับ Admin/Superadmin



รูปที่ ก.54: หน้าต่างอนุมัติคำขอย้ายตำแหน่งสำหรับ Admin/Superadmin

การอนุมัติแจ้งซ่อม: ผู้ดูแลสามารถดูคำร้องแจ้งซ่อมที่รอดำเนินการทั้งหมดภายในหน่วยงาน และสามารถกดรับเรื่อง (Accept) หรือปฏิเสธ (Deny) คำร้องแจ้งซ่อมได้

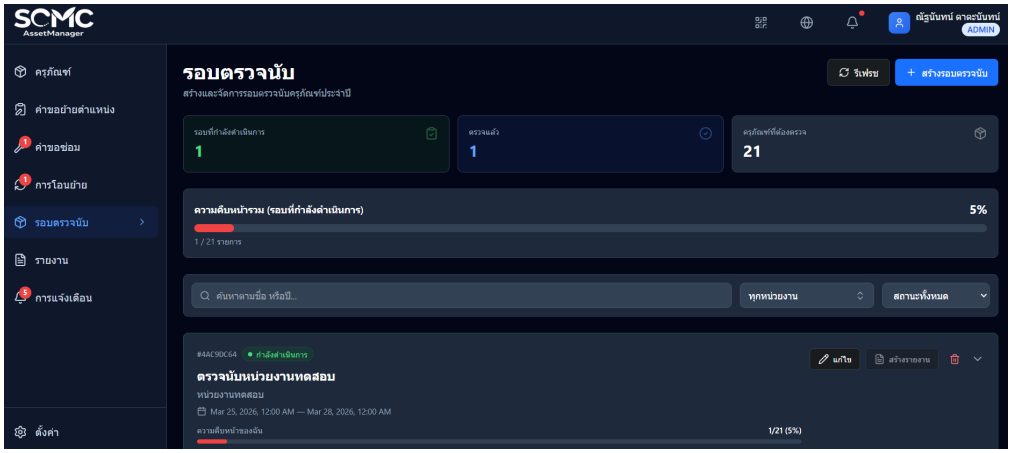


รูปที่ ก.55: หน้าต่างดูคำร้องแจ้งซ่อมสำหรับ Admin/Superadmin

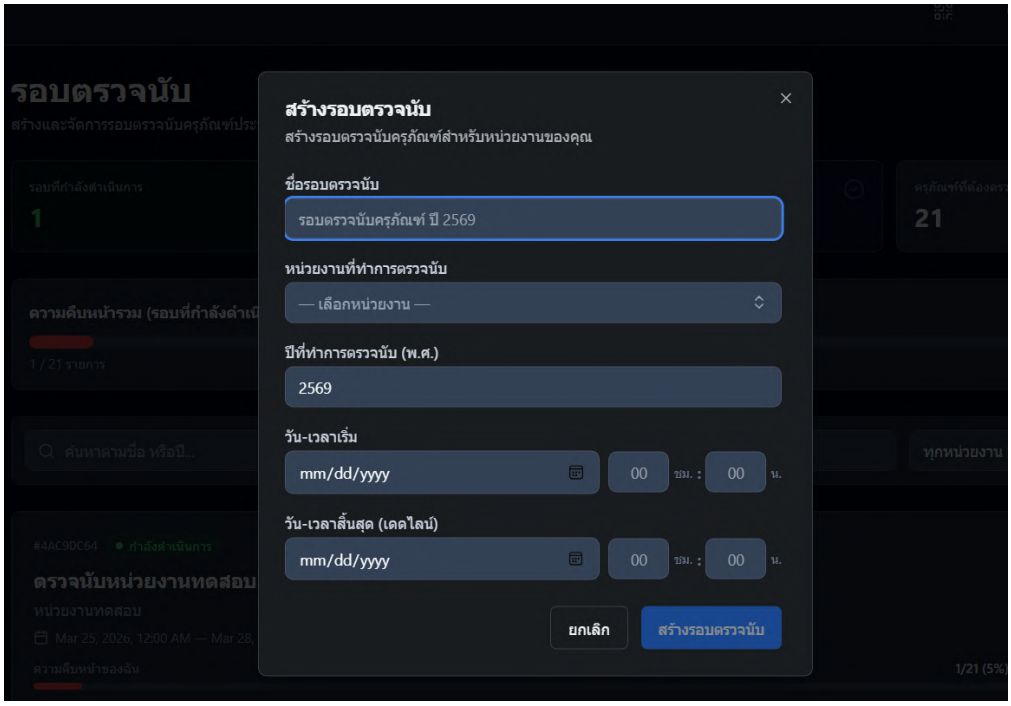
รูปที่ ก.56: หน้าต่างอนุมัติคำร้องแจ้งซ่อมสำหรับ Admin/Superadmin

ก.8.3 รอบตรวจนับ (Check Cycles)

การสร้างรอบ: ผู้ดูแลสามารถสร้างรอบการตรวจนับครุภัณฑ์ประจำปีของหน่วยงานตนเองได้ โดยระบุชื่อ, ปี, วันเริ่มต้น, วันสิ้นสุด และรายละเอียด แต่ไม่สามารถสร้างรอบตรวจนับที่อยู่สูงกว่าหน่วยงานที่ตัวเองสังกัดอยู่ได้

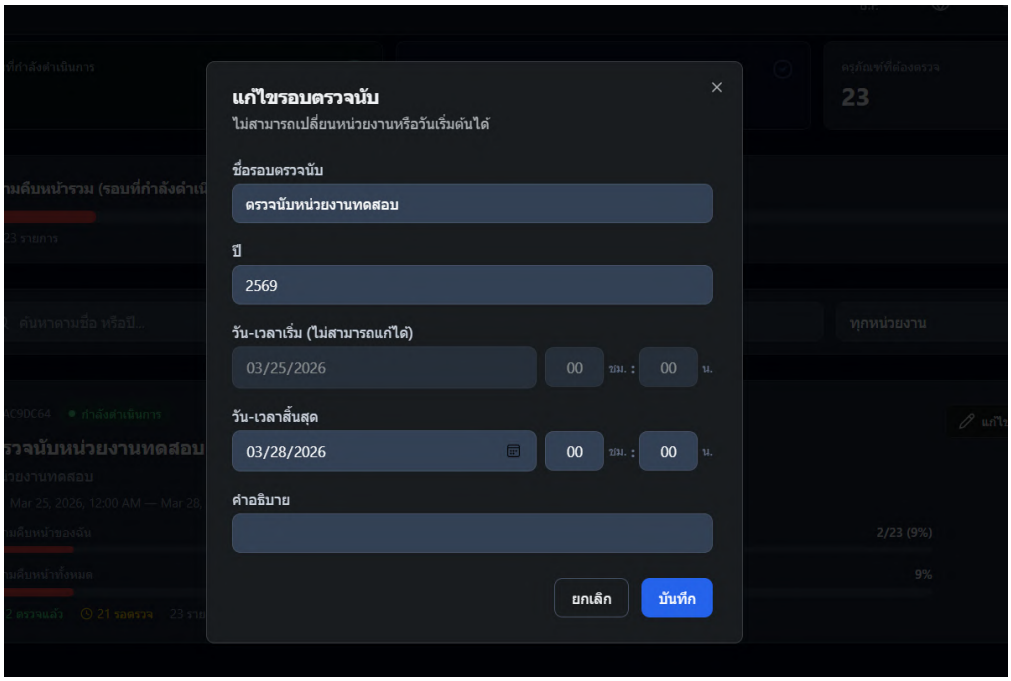


รูปที่ ก.57: หน้าต่างตรวจสอบรอบตรวจนับสำหรับ Admin/Superadmin



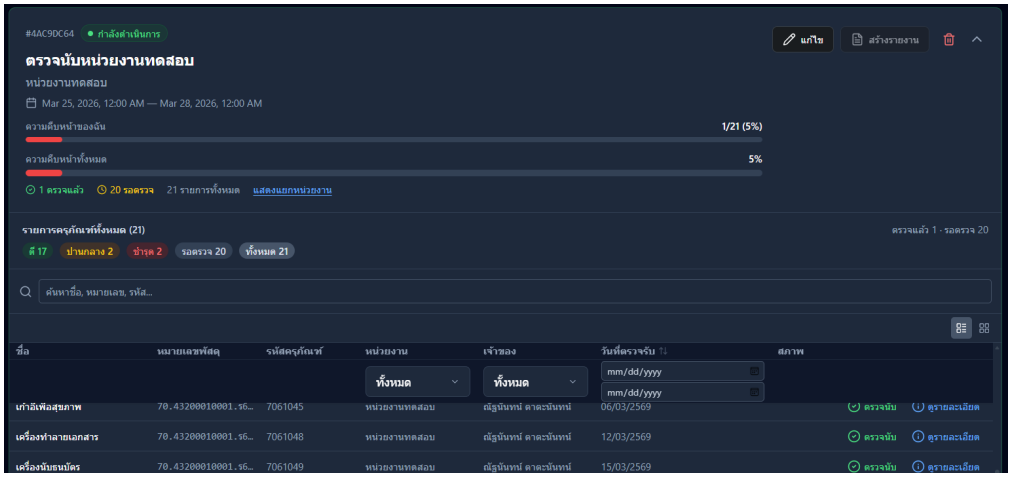
รูปที่ ก.58: หน้าต่างสร้างรอบตรวจนับสำหรับ Admin/Superadmin

การจัดการรอบ: สามารถแก้ไขรอบตรวจนับเพื่อขยายเวลาสิ้นสุดได้ และสามารถลบรอบการตรวจนับได้



รูปที่ ก.59: หน้าต่างแก้ไขรอบตรวจนับสำหรับ Admin/Superadmin

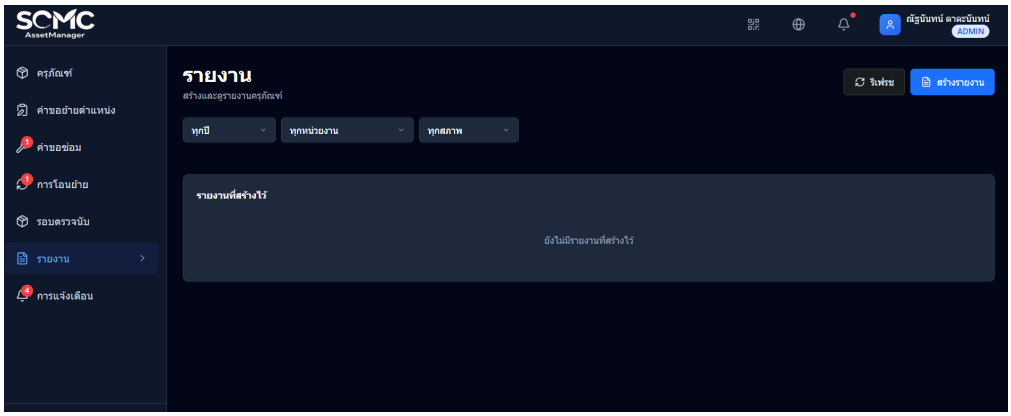
การติดตามผล: สามารถดูแผงแสดงความคืบหน้า (จำนวนทั้งหมด / ที่นับแล้ว / ที่ยังไม่นับ / เปอร์-เซ็นต์) ของหน่วยงานตนเอง และสามารถเรียกดูรายชื่อผู้ใช้งานที่ยังไม่ได้ทำการตรวจนับได้



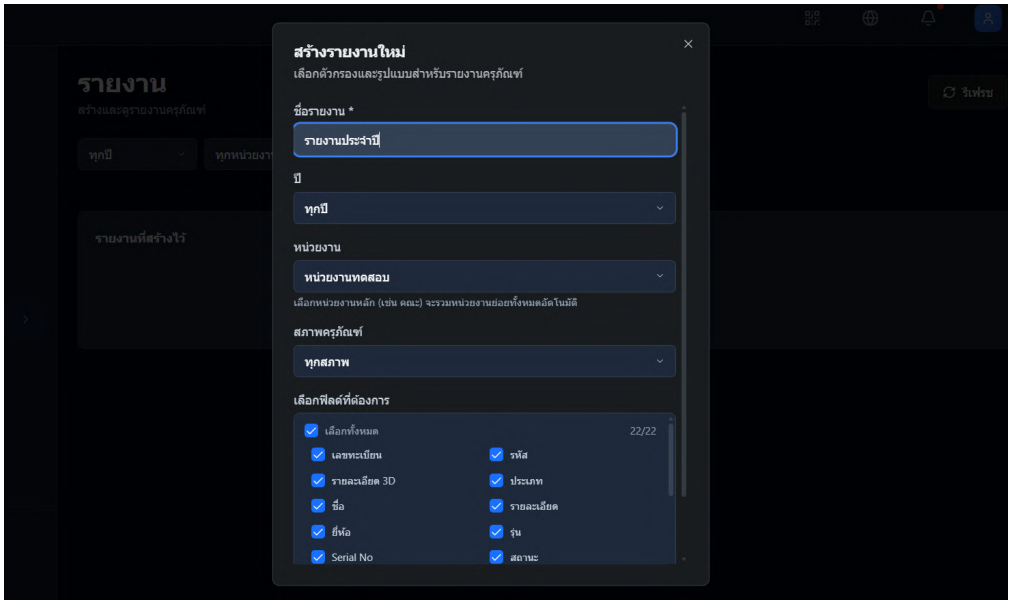
รูปที่ ก.60: หน้าต่างติดตามผลรอบตรวจนับสำหรับ Admin/Superadmin

ก.8.4 รายงาน (Reports)

การสร้างรายงาน: ผู้ดูแลสามารถสร้างรายงานสรุปครุภัณฑ์ประจำปี หรือกรองรายงานตามปี, หน่วยงาน, และสภาพครุภัณฑ์ได้

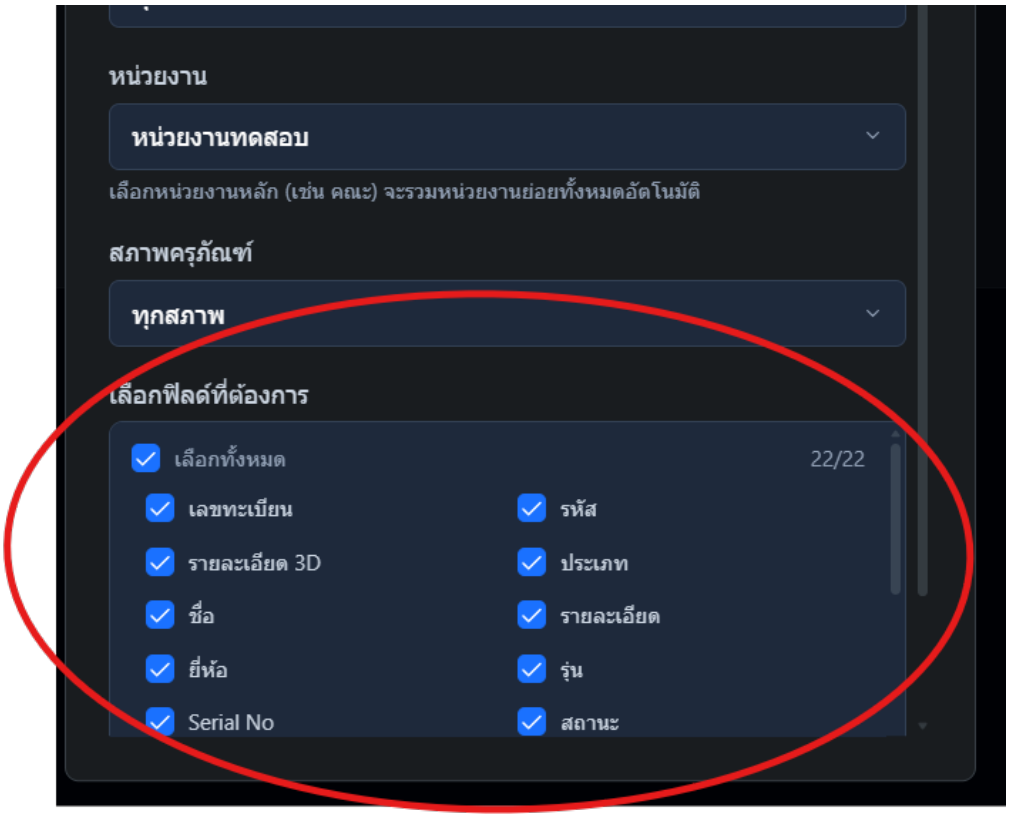


รูปที่ ก.61: หน้าต่างสร้างรายงานสำหรับ Admin/Superadmin



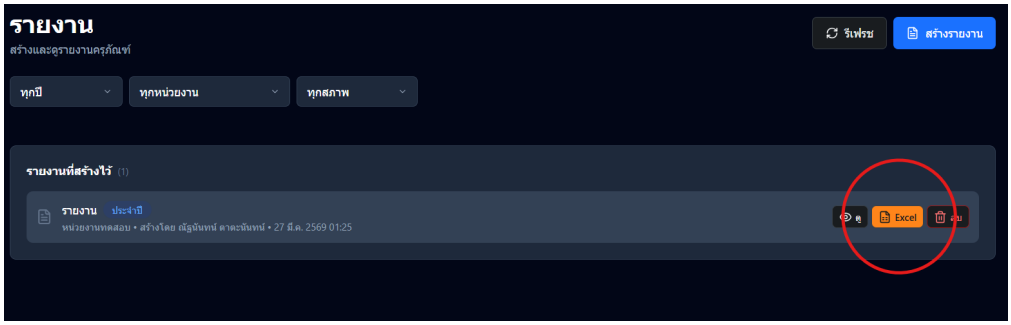
รูปที่ ก.62: หน้าต่างสร้างรายงานสำหรับ Admin/Superadmin

การปรับแต่ง: สามารถเลือกได้ว่าต้องการให้รวมฟิลด์หรือแอตทริบิวต์ข้อมูลใดบ้างในรายงาน

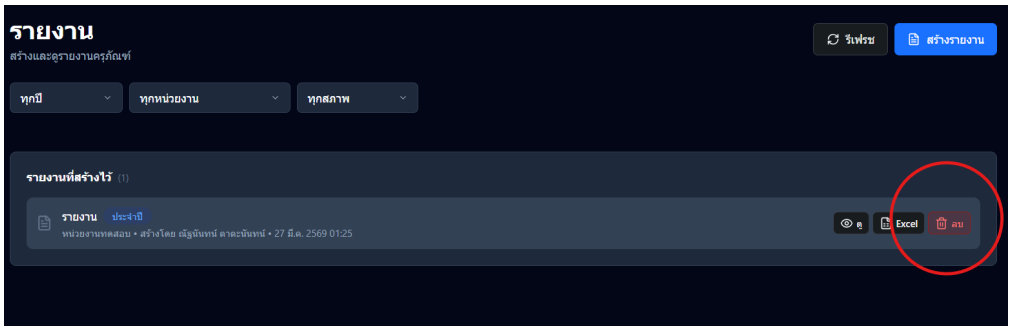


รูปที่ ก.63: หน้าต่างปรับแต่งรายงานสำหรับ Admin/Superadmin

การจัดการไฟล์: สามารถส่งออกข้อมูลรายงานเป็นไฟล์ Excel สำหรับการวิเคราะห์เพิ่มเติม สามารถบันทึก เรียกดู และลบรายงานที่เคยสร้างไว้ก่อนหน้าได้ ผู้ดูแลจะเห็นเฉพาะรายงานของโครงสร้างหน่วยงานตนเองเท่านั้น



รูปที่ ก.64: หน้าต่างส่งออกรายงานเป็นไฟล์ Excel สำหรับ Admin/Superadmin



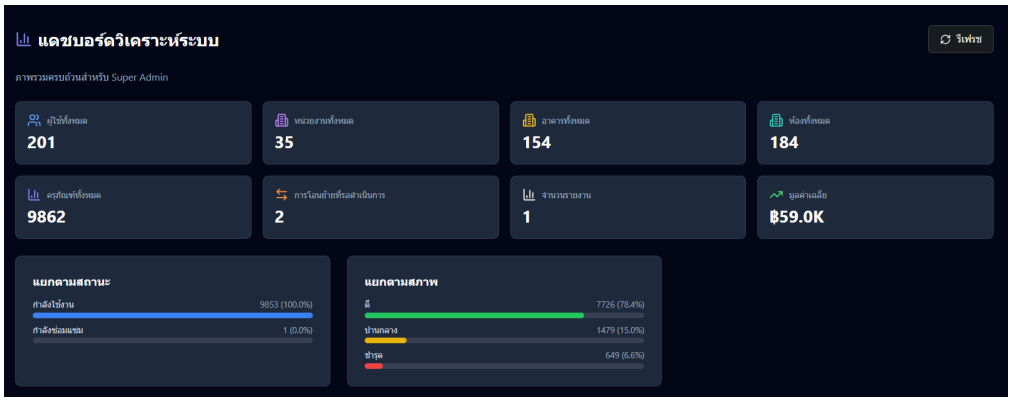
รูปที่ ก.65: หน้าต่างลบรายงานสำหรับ Admin/Superadmin

ก.9 คู่มือสำหรับผู้ดูแลระบบสูงสุด (Superadmin)

ผู้ดูแลระบบสูงสุดสามารถใช้งานฟีเจอร์ทั้งหมดของ User และ Admin ได้ และมีสิทธิ์ควบคุมระบบอย่างสมบูรณ์แบบดังนี้

ก.9.1 แดชบอร์ดและสถิติ (Dashboard & Analytics)

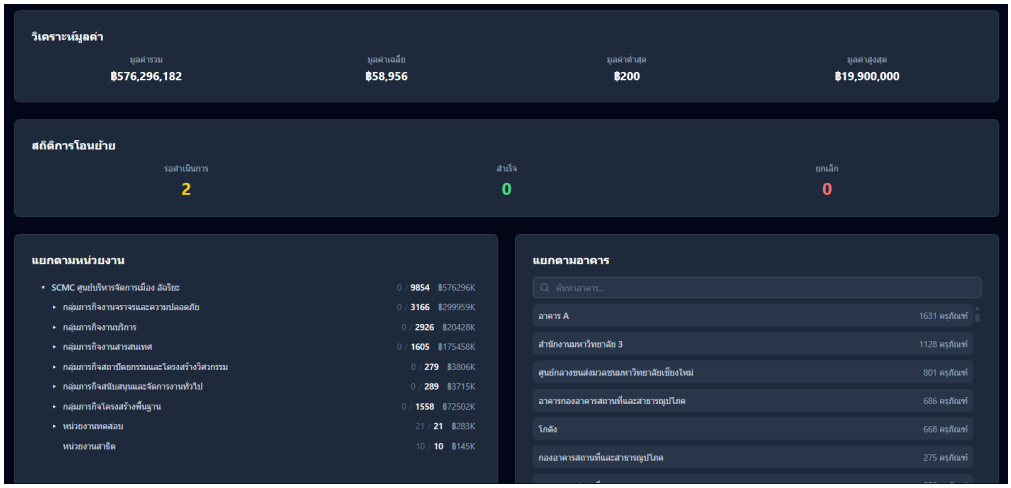
ภาพรวมระบบ: Superadmin จะสามารถดูหน้าแดชบอร์ดภาพรวมของระบบได้ทันทีที่เข้าสู่ระบบ หน้าต่างนี้จะแสดงการ์ดสรุปจำนวนครุภัณฑ์ทั้งหมด, มูลค่ารวม, จำนวนผู้ใช้งาน, หน่วยงาน, และอาคาร



รูปที่ ก.66: หน้าต่างแดชบอร์ดส่วนบนสำหรับ Superadmin

กราฟวิเคราะห์: มีกราฟแบบอินเทอร์แอคทีฟแสดงสถิติจำนวนครุภัณฑ์แยกตามสถานะ, สภาพ, ประเภท และจัดกลุ่มตามอาคารที่จัดเก็บ

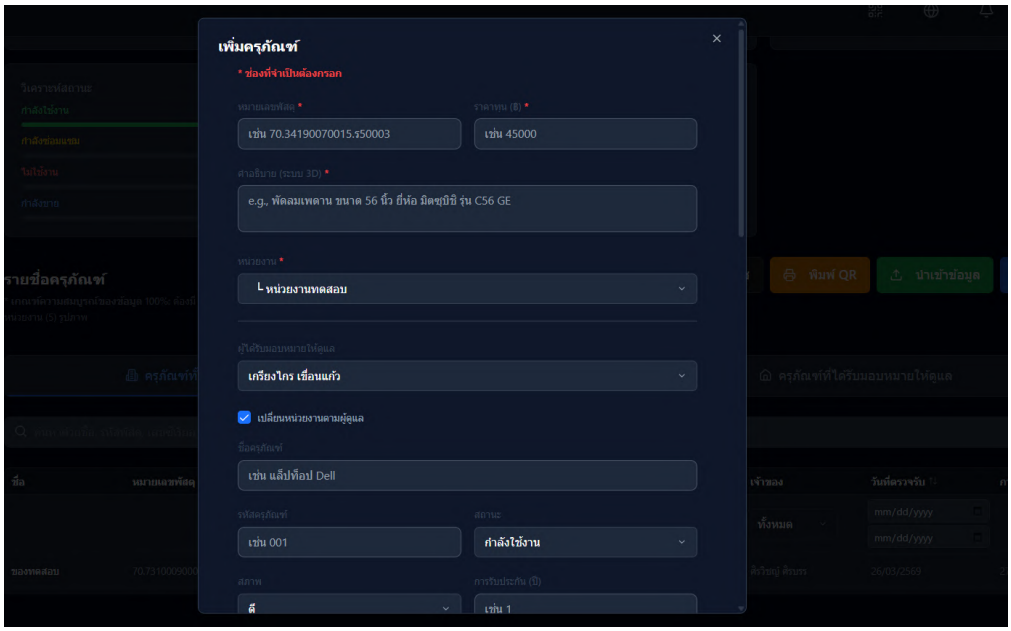
กิจกรรมและข้อมูลเชิงลึก: สามารถดูตารางรายชื่อผู้ครอบครองครุภัณฑ์พร้อมมูลค่าในความดูแล, สถิติการโอนย้าย



รูปที่ ก.67: หน้าต่างแดชบอร์ดส่วนล่างสำหรับ Superadmin

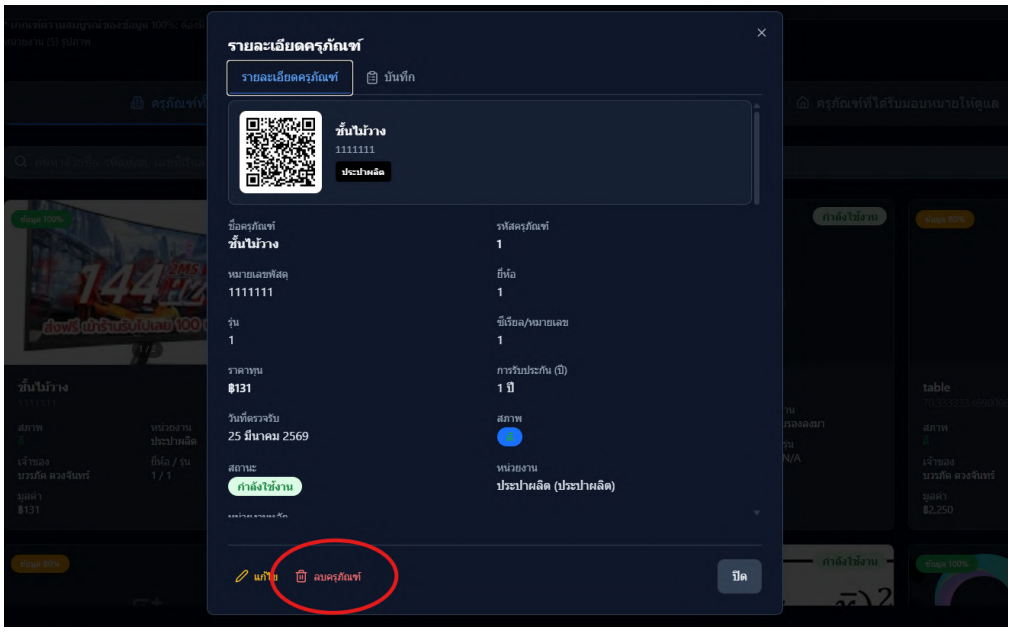
ก.9.2 การบริหารจัดการข้อมูลครุภัณฑ์แบบเจาะลึก

การเพิ่มครุภัณฑ์ใหม่แบบเดี่ยว: สามารถเพิ่มข้อมูลครุภัณฑ์ใหม่พร้อมรายละเอียดทั้งหมด รวมถึงสามารถสร้างรายการย่อยได้หลายรายการต่อหนึ่งครุภัณฑ์สำหรับประเภท MULTI (ระบุอาคาร, ชั้น, ห้อง, รูปภาพ แยกแต่ละชิ้นได้) ระบบจะสร้าง QR Code อัตโนมัติเมื่อเพิ่มข้อมูล

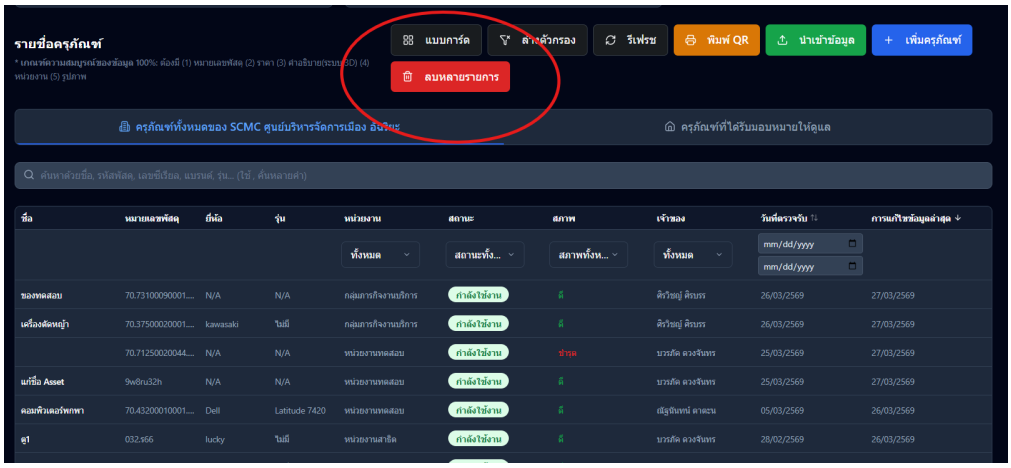


รูปที่ ก.68: หน้าต่างแบบฟอร์มการเพิ่มครุภัณฑ์สำหรับ Superadmin

การลบข้อมูล: สามารถลบข้อมูลครุภัณฑ์แบบรายชิ้น หรือลบข้อมูลพร้อมกันหลายรายการ (Bulk delete) ออกจากระบบได้

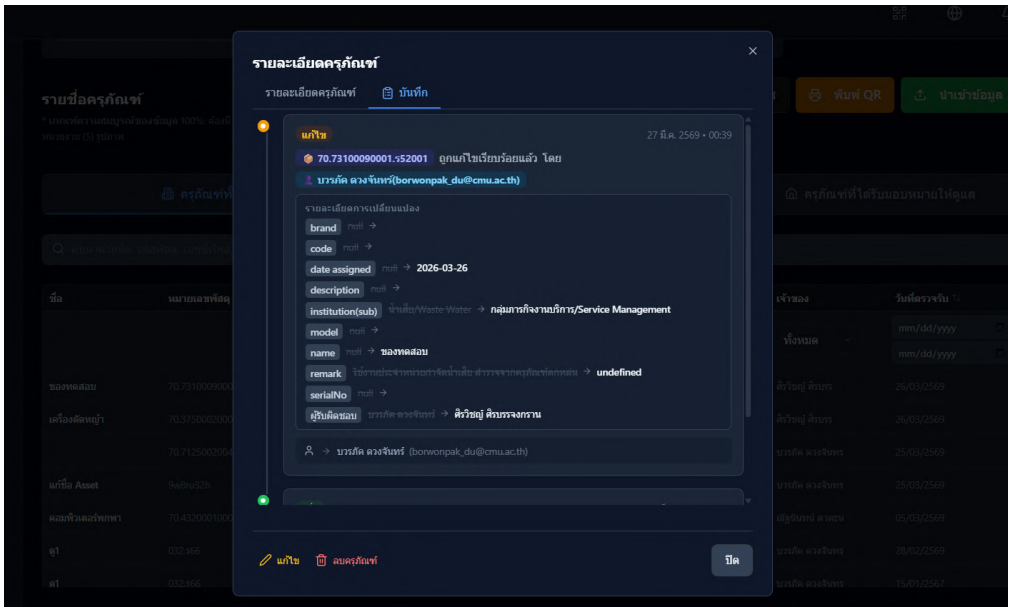


รูปที่ ก.69: ปุ่มการลบครุภัณฑ์ขแบบรายชิ้นสำหรับ Superadmin



รูปที่ ก.70: ปุ่มการลบครุภัณฑ์แบบพร้อมกันหลายรายการสำหรับ Superadmin

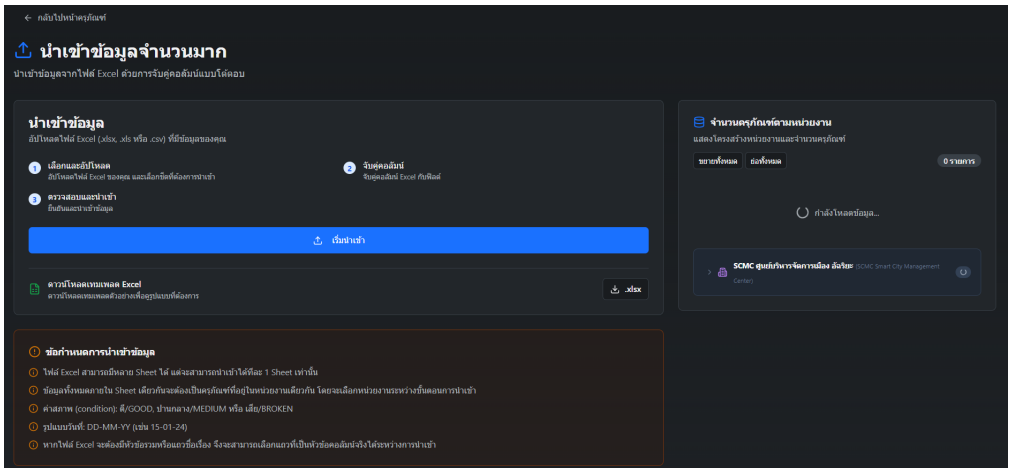
ประวัติทำรายการ (Audit Logs): สามารถเรียกดูประวัติการเปลี่ยนแปลงข้อมูลของครุภัณฑ์แต่ละชิ้นได้จากหน้ารายละเอียดครุภัณฑ์โดยตรง (ติดตามการเพิ่ม, แก้ไข, เปลี่ยนเจ้าของ, ตรวจนับ, โอนย้าย, แจ้งซ่อม)



รูปที่ ก.71: หน้าต่างการเรียกดูประวัติทำรายการ (Audit Logs) ภายในรายละเอียดครุภัณฑ์สำหรับ Superadmin

ก.9.3 นำเข้าข้อมูลจำนวนมาก (Bulk Import)

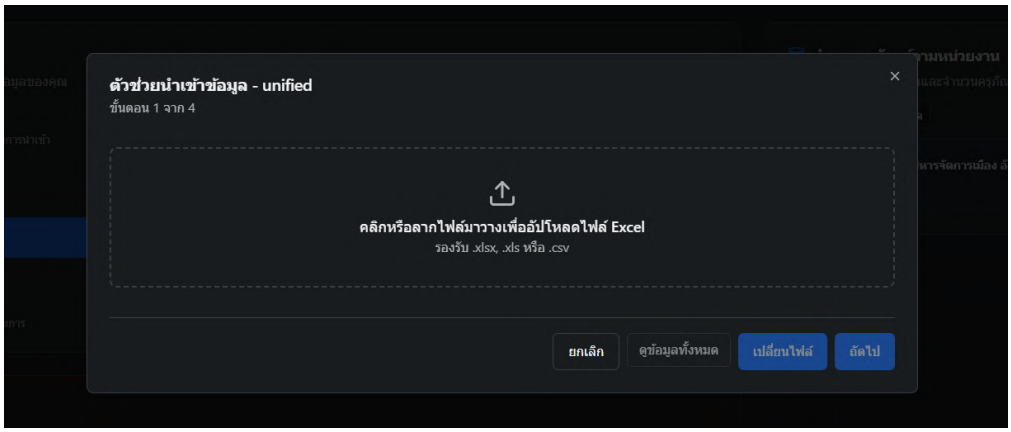
การเข้าใช้งาน: นำเข้าข้อมูลครุภัณฑ์จำนวนมากจากไฟล์ Excel ผ่านระบบ Import Wizard



รูปที่ ก.72: หน้าต่างการนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin

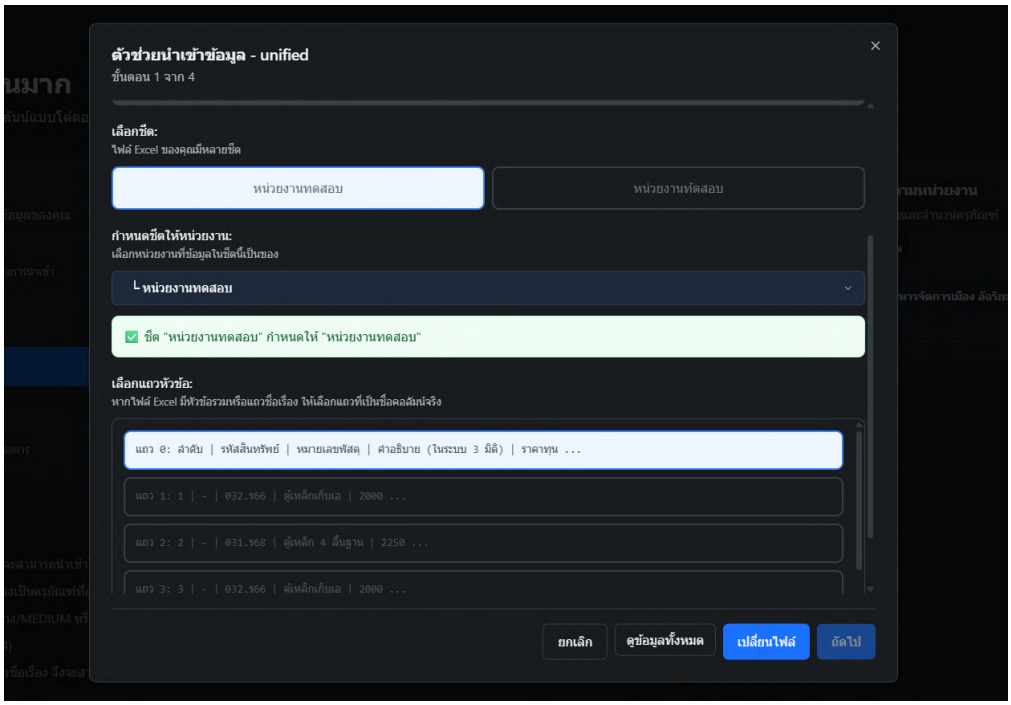
ขั้นตอนการนำเข้าข้อมูล:

1. **อัปโหลด:** เริ่มดาวน์โหลดเทมเพลตไฟล์จากระบบ กรอกข้อมูลให้ครบถ้วน จากนั้นเลือกไฟล์ Excel เพื่ออัปโหลด



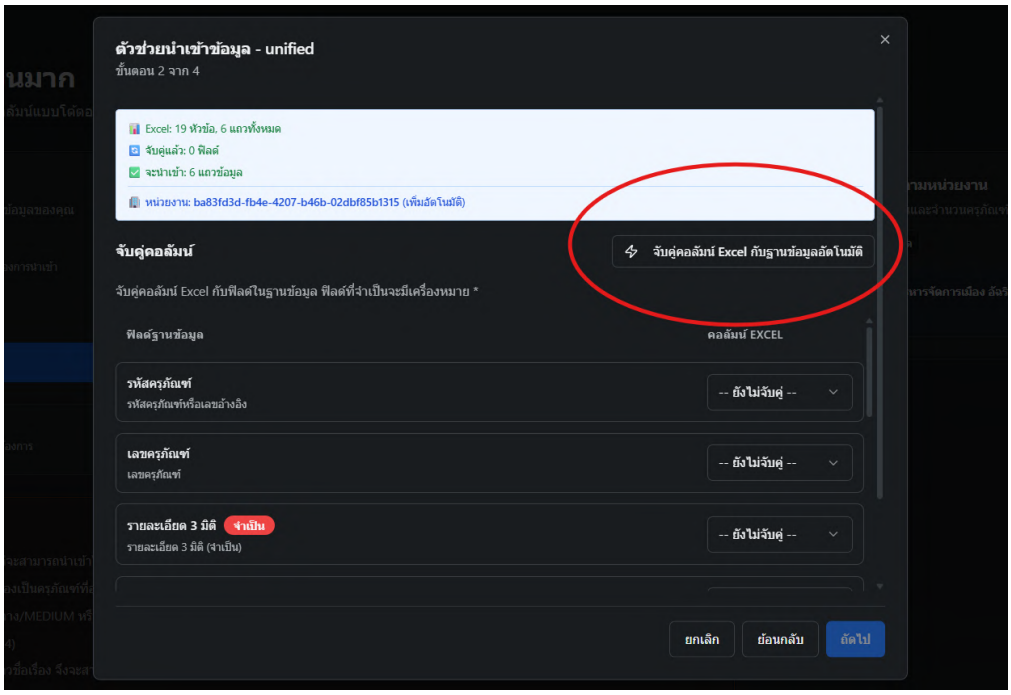
รูปที่ ก.73: หน้าต่างเริ่มนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin

2. **เลือกชีต:** เลือกระบุชีตที่ต้องการนำเข้า และเลือกหน่วยงานเป้าหมาย
3. **แถวหัวข้อ:** เลือกระบุแถวที่เป็นหัวข้อ (Header Row) ของตาราง Excel



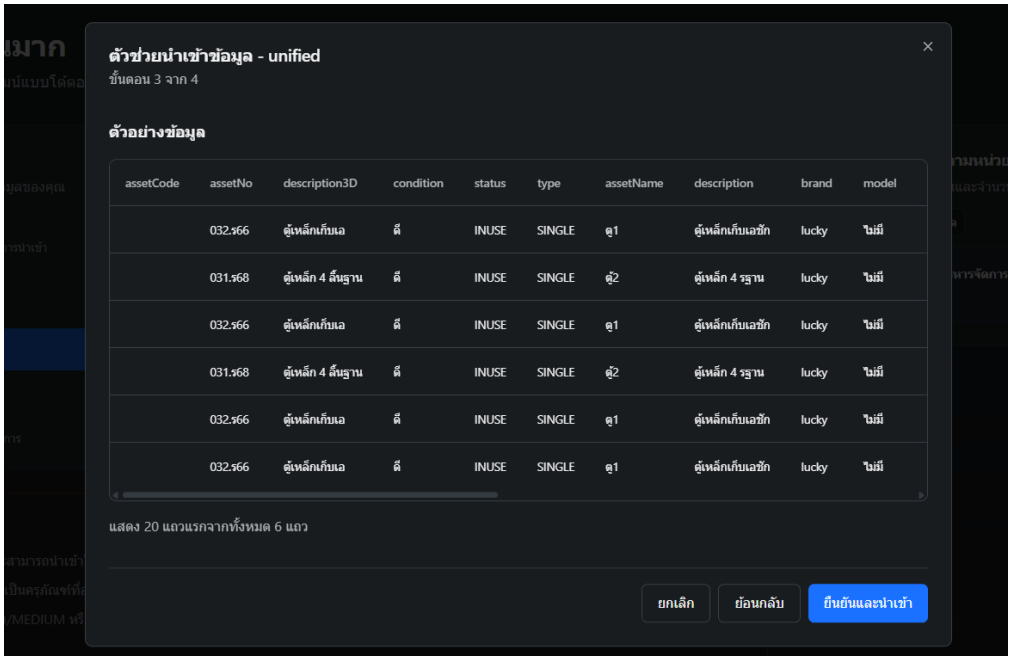
รูปที่ ก.74: หน้าต่างเลือกชีต, หน่วยงาน, และแถวหัวข้อสำหรับ Superadmin

4. **จับคู่คอลัมน์:** ทำการ Map ข้อมูลคอลัมน์จาก Excel เข้ากับ Database Attribute ของระบบ สามารถกดปุ่มจับคู่อัตโนมัติได้หากตั้งชื่อคอลัมน์ตรงกัน ฟิลด์ที่จำเป็นต้องมีคือ รายละเอียด 3 มิติ (description3D) [Manual prompt]



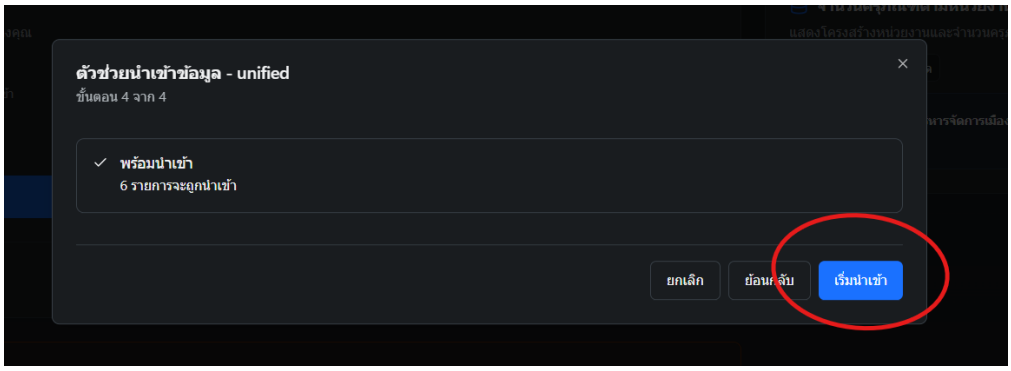
รูปที่ ก.75: หน้าต่างจับคู่คอลัมน์สำหรับ Superadmin

5. **ตรวจสอบ:** ตรวจสอบความถูกต้องของข้อมูลก่อนกดยืนยัน



รูปที่ ก.76: หน้าต่างตรวจสอบข้อมูลก่อนนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin

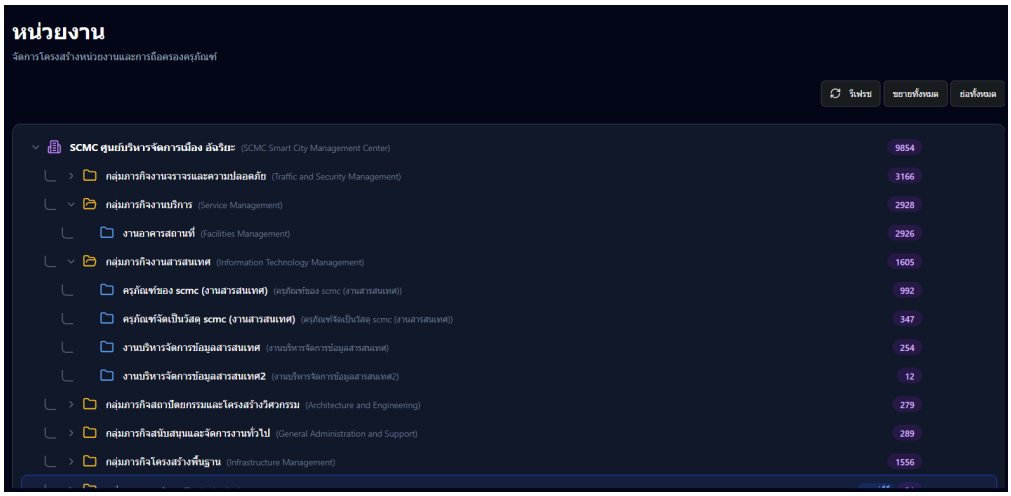
6. **นำเข้า:** กดยืนยันเพื่อนำเข้าข้อมูล หากระบบตรวจพบ อาคาร, ห้อง หรือผู้ใช้งานใหม่ที่ไม่มีในฐานข้อมูล ระบบจะสร้างให้โดยอัตโนมัติ ครุภัณฑ์ที่ไม่มีชื่อผู้ดูแลจะถูกโอนกรรมสิทธิ์อัตโนมัติให้ SUPER-ADMIN



รูปที่ ก.77: หน้าต่างเริ่มนำเข้าข้อมูลจำนวนมากสำหรับ Superadmin

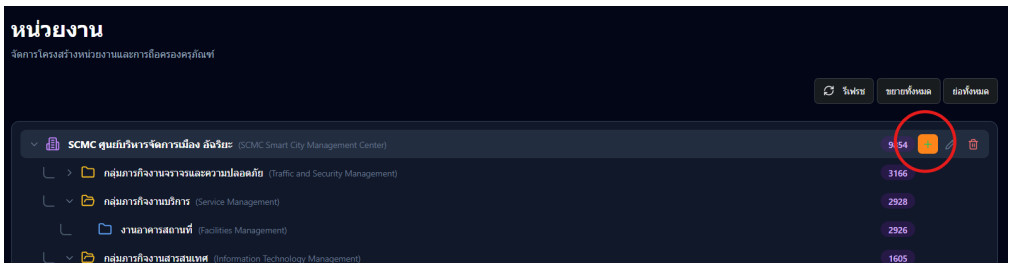
ก.9.4 การจัดการองค์กร (Organization / Institution)

แผนผังหน่วยงาน: สามารถดูแผนผังโครงสร้างหน่วยงานในรูปแบบ Tree ขยายโหนดเพื่อดูหน่วยงานย่อยและจำนวนครุภัณฑ์สะสมได้

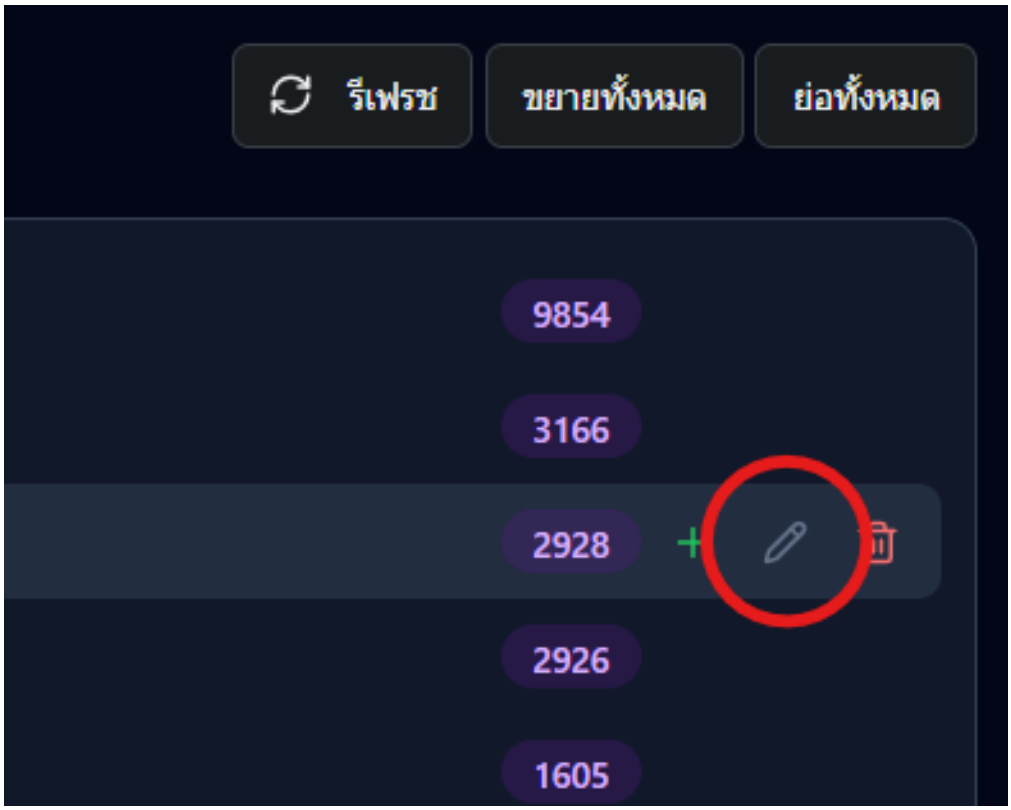


รูปที่ ก.78: หน้าต่างแผนผังหน่วยงานสำหรับ Superadmin

การปรับแต่งหน่วยงาน: สามารถเพิ่มหน่วยงานย่อยภายใต้หน่วยงานหลัก และสามารถแก้ไขชื่อหน่วยงาน (รองรับภาษาไทยและภาษาอังกฤษ) ได้

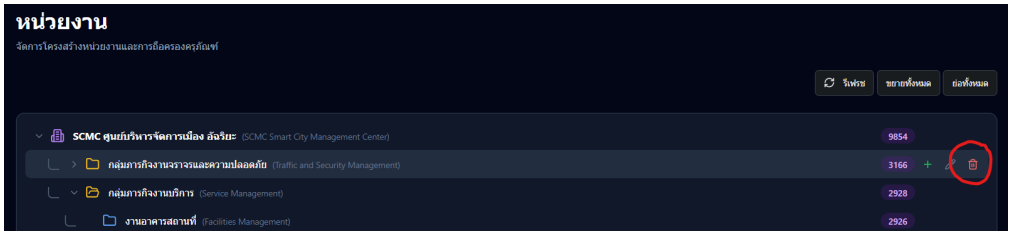


รูปที่ ก.79: หน้าต่างเพิ่มหน่วยงานสำหรับ Superadmin



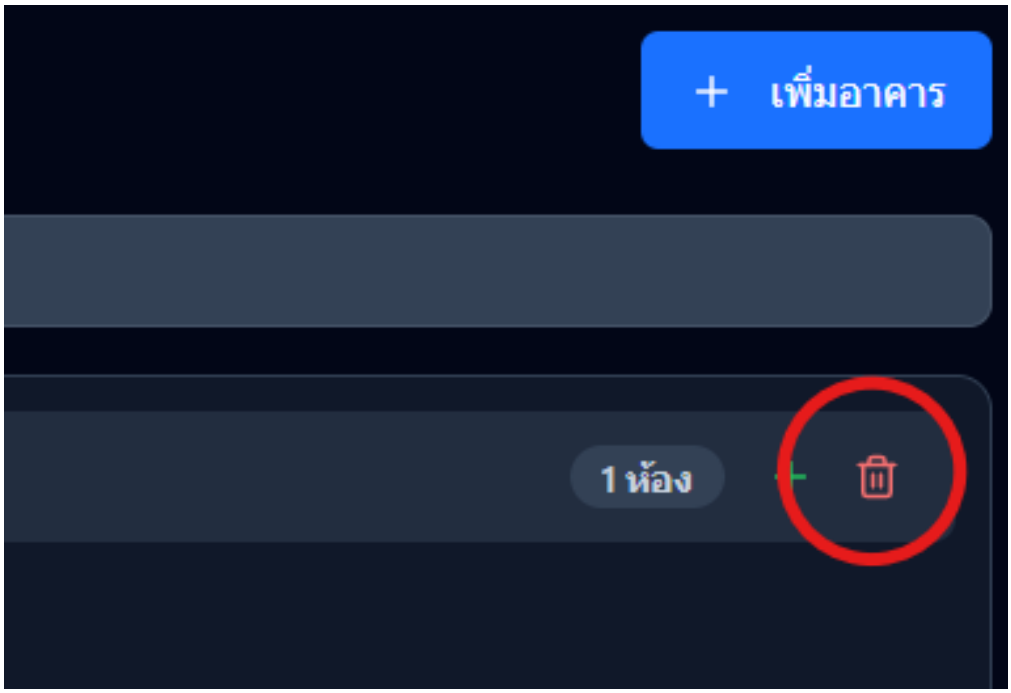
รูปที่ ก.80: หน้าต่างแก้ไขหน่วยงานสำหรับ Superadmin

การลบ/ยุบหน่วยงาน: สามารถลบหน่วยงานย่อยได้ (ครุภัณฑ์ทั้งหมดจะถูกโอนกลับไปยังหน่วยงานระดับบนอัตโนมัติ)ไม่สามารถลบหน่วยงานได้หากยังมีครุภัณฑ์ผูกมัดอยู่ SUPERADMIN สามารถกดลบ/ยุบหน่วยงานระดับสูงสุดของตนเองได้ หากตนเป็นสมาชิกเพียงคนเดียว ซึ่งจะเป็นการลบข้อมูลหน่วยงานลูกและครุภัณฑ์ทั้งหมดทิ้ง

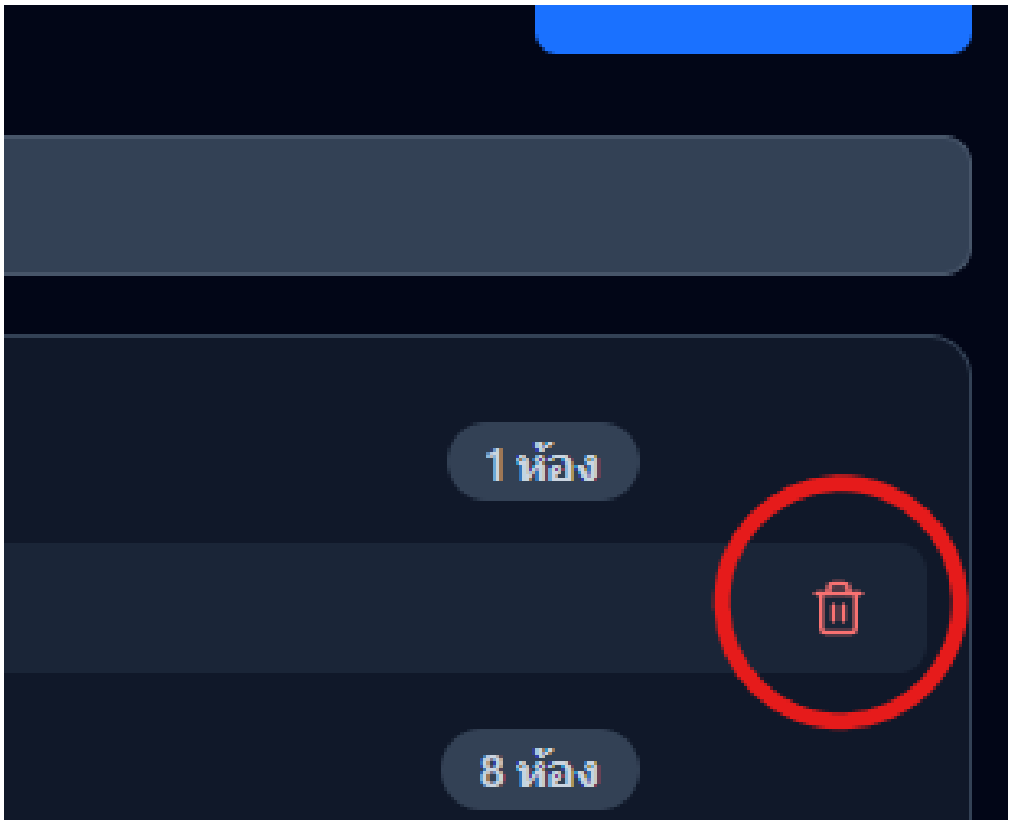


รูปที่ ก.81: หน้าต่างลบหน่วยงานสำหรับ Superadmin

การลบอาคารและห้อง: มีสิทธิ์ในการลบข้อมูลอาคาร (ซึ่งจะลบห้องที่ผูกอยู่ด้วยทั้งหมดแบบ Cascade) และลบข้อมูลห้องได้



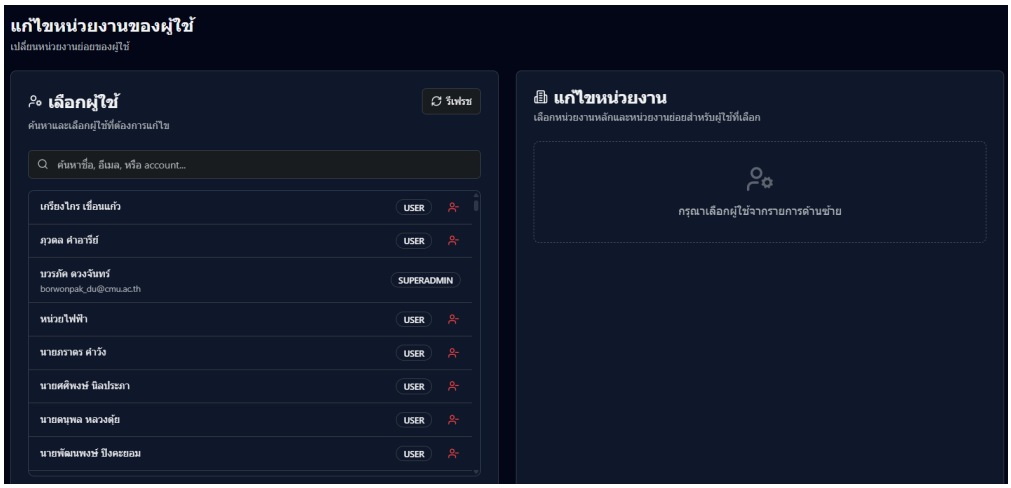
รูปที่ ก.82: หน้าต่างลบอาคารสำหรับ Superadmin



รูปที่ ก.83: หน้าต่างลบห้องสำหรับ Superadmin

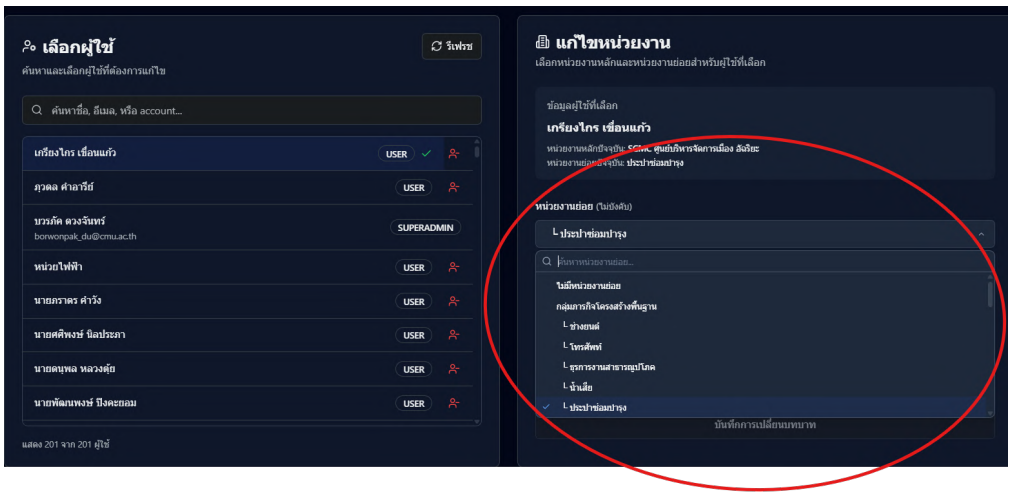
ก.9.5 จัดการผู้ใช้งาน (User Management)

การดูแลสมาชิก: สามารถเรียกดูรายชื่อผู้ใช้งานทั้งหมด ค้นหาผู้ใช้ และแก้ไขสังกัดหน่วยงานย่อยของผู้ใช้งานแต่ละรายได้

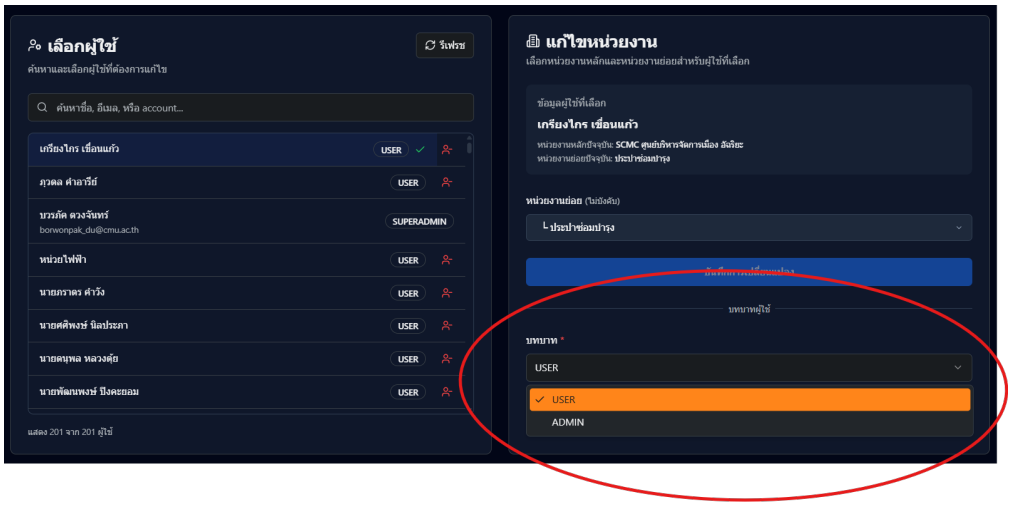


รูปที่ ก.84: หน้าต่างจัดการผู้ใช้งานสำหรับ Superadmin

การปรับเปลี่ยนสิทธิ์: สามารถแก้ไขเปลี่ยนบทบาทผู้ใช้ระหว่าง ผู้ใช้งาน (User) และผู้ดูแล (Admin) หรือสามารถส่งมอบสิทธิ์ Superadmin ให้กับผู้ใช้งานคนอื่นได้

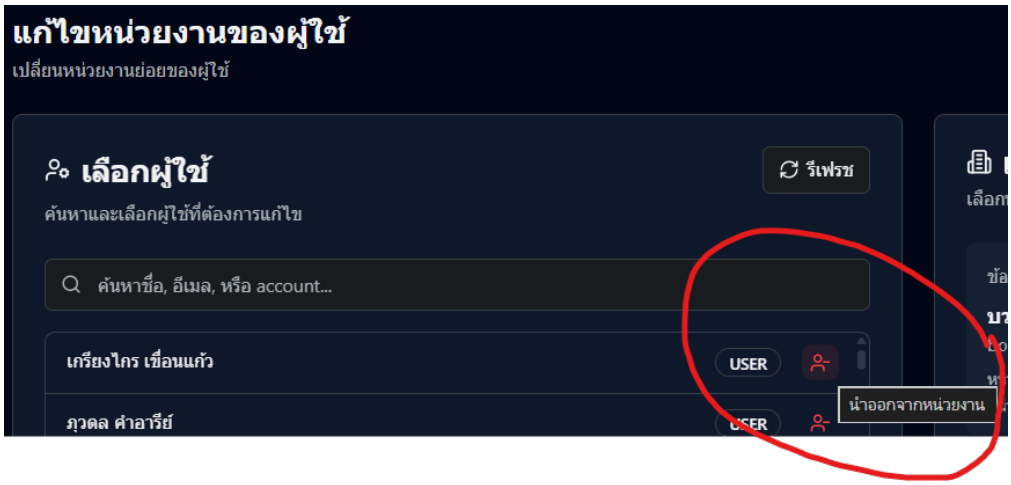


รูปที่ ก.85: หน้าต่างแก้ไขสังกัดหน่วยงานสำหรับ Superadmin



รูปที่ ก.86: หน้าต่างแก้ไขบทบาทผู้ใช้งานสำหรับ Superadmin

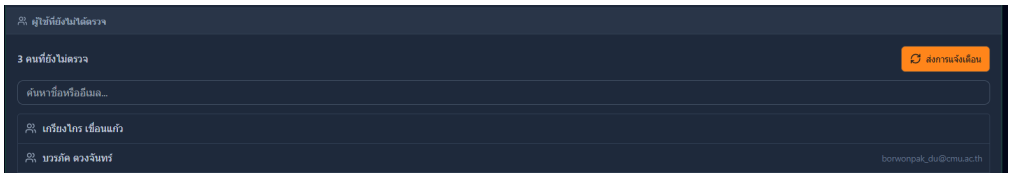
การนำผู้ใช้ออก: สามารถเตะ (Kick) ผู้ใช้งานออกจากหน่วยงานได้โดยพิมพ์คำว่า KICK ยืนยัน ครุภัณฑ์ที่ผู้ใช้นั้นถืออยู่จะถูกโอนกรรมสิทธิ์กลับมายัง Superadmin โดยอัตโนมัติ



รูปที่ ก.87: หน้าต่างเตะผู้ใช้งานออกจากหน่วยงานสำหรับ Superadmin

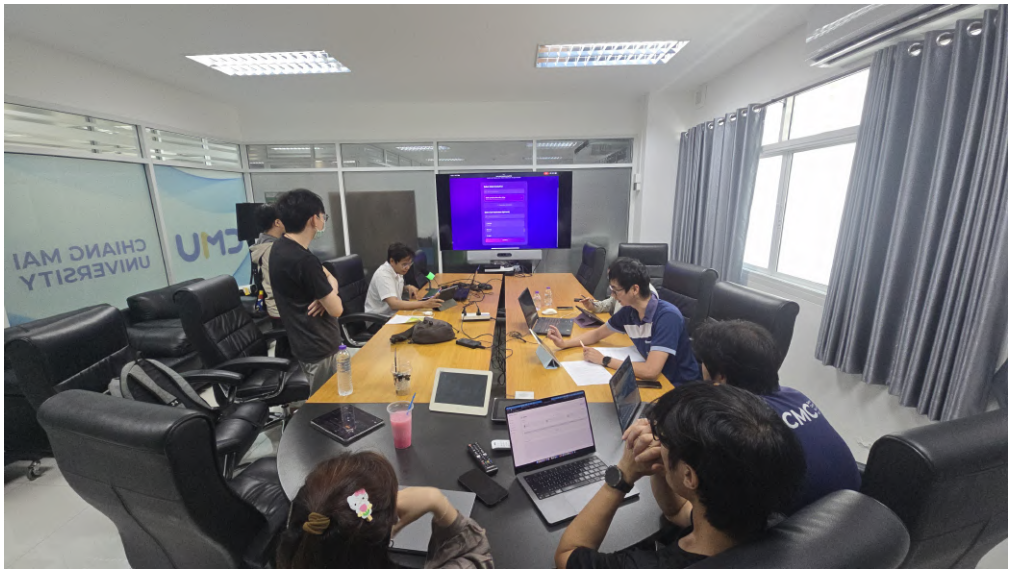
ก.9.6 การแจ้งเตือนและการเตือนความจำรอบตรวจนับ

สิทธิ์การควบคุมการแจ้งเตือน: ในระหว่างที่มีการตรวจนับครุภัณฑ์ Superadmin จะมีปุ่มทางเลือกพิเศษสำหรับกดส่งการแจ้งเตือนเพื่อเร่งรัด (Reminder) ผ่านทางอีเมลและระบบ ไปยังผู้ใช้งานรายบุคคลที่ยังไม่ได้ทำการตรวจนับครุภัณฑ์ของตนให้แล้วเสร็จ



รูปที่ ก.88: หน้าต่างส่งการแจ้งเตือนเพื่อเร่งรัดผู้ใช้งานสำหรับ Superadmin

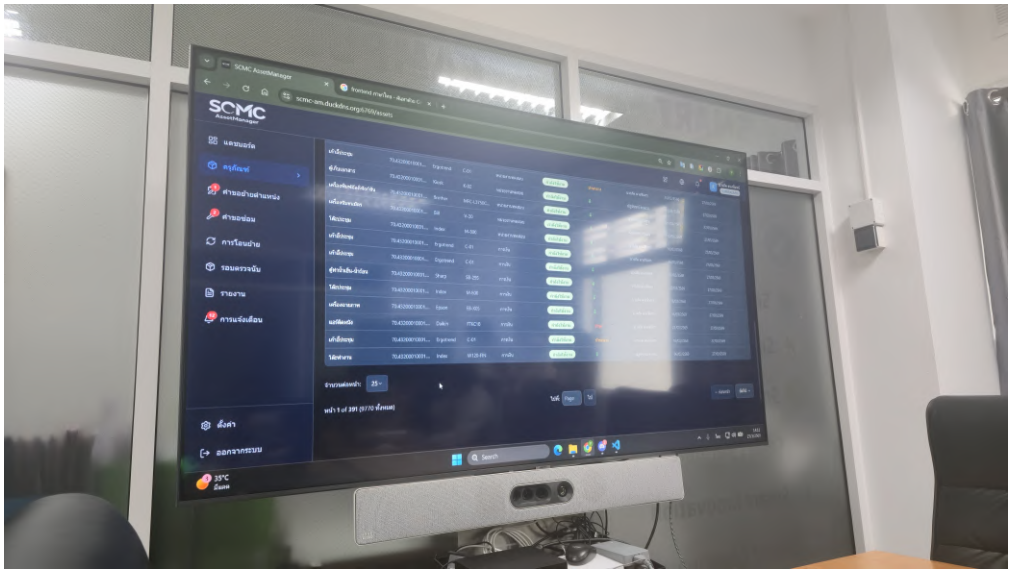
ภาคผนวก ข
การทดสอบระบบกับศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน



รูปที่ ข.1: ภาพบรรยากาศการทดสอบระบบกับศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน

รูปที่ ข.2: ภาพระหว่างที่นายบวรภัค ดวงจันทร์ กำลังอธิบายการใช้งานระบบให้กับเจ้าหน้าของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน

รูปที่ ข.3: ภาพระหว่างที่สมาชิกทีมกำลังชมการทดสอบการใช้งานระบบร่วมกับเจ้าหน้าที่ฝ่ายพัสดุของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน



รูปที่ ข.4: ภาพหน้าจอที่แสดงการทดสอบการใช้งานระบบร่วมกับเจ้าหน้าที่ฝ่ายพัสดุของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน

รูปที่ ข.5: ภาพระหว่างที่สมาชิกทีมกำลังทดสอบการใช้งานระบบร่วมกับเจ้าหน้าที่ฝ่ายพัสดุของศูนย์บริหารจัดการเมืองเพื่อความยั่งยืน

ประวัติผู้เขียน

ชื่อ: นายณัฐนันทน์ ตาตะนันทน์

รหัสนักศึกษา: 650610760

การศึกษา: วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ชื่อ: นายบวรภัค ดวงจันทร์

รหัสนักศึกษา: 650610779

การศึกษา: วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ชื่อ: นายพชร จารุรักษ์สกุล

รหัสนักศึกษา: 650610789

การศึกษา: วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ชื่อ: นายศิรวิชญ์ ศิรบรรจงกราน

รหัสนักศึกษา: 650610810

การศึกษา: วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่